

EX NO:	MINI PROJECT : Biometric Attendance System With GSM Module
DATE:	

### AIM :

The aim of this project is to create a biometric attendance system that enrolls and verifies fingerprints using an Adafruit fingerprint sensor and sends attendance notifications via a GSM module.

### HARDWARE REQUIREMENT :

- Arduino board
- Adafruit Fingerprint Sensor
- OLED Display (SSD1306)
- GSM Module
- SIM Card
- Power Supply
- Wires and Breadboard

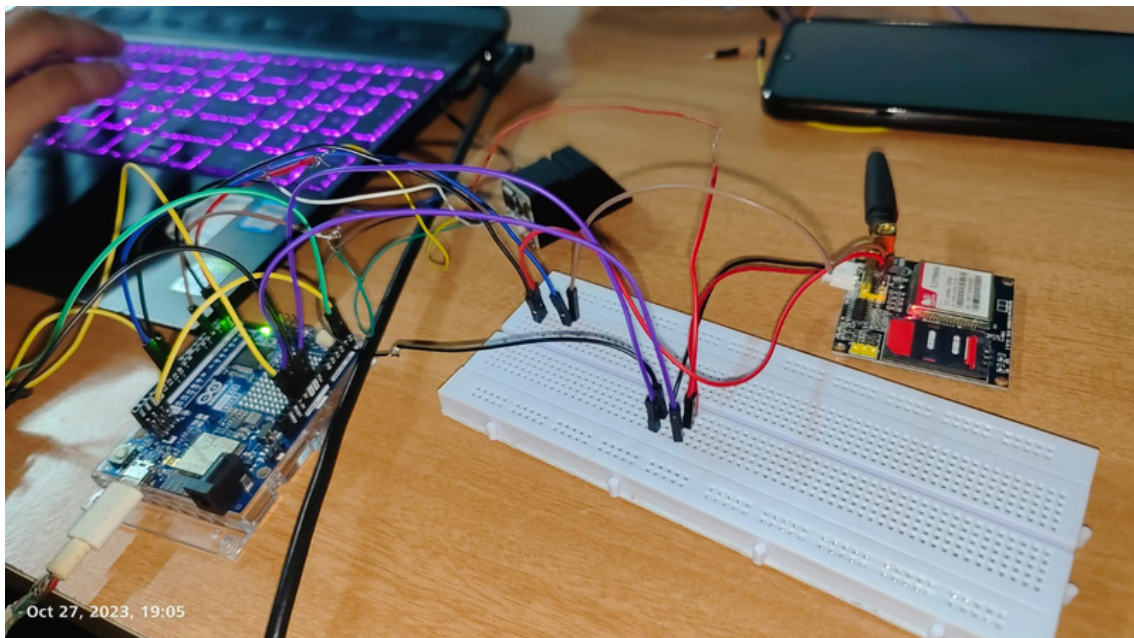
### SOFTWARE REQUIREMENT :

1. Arduino IDE
2. Necessary Libraries:
  - Wire.h
  - Adafruit\_GFX.h
  - Adafruit\_SSD1306.h
  - Adafruit\_Fingerprint.h
  - SoftwareSerial.h

## THEORY:

- This project combines biometric fingerprint recognition technology with a GSM module to create an attendance system.
- It enrolls multiple fingerprints with different IDs.
- SMS notifications are sent through the GSM module upon successful fingerprint enrollment.

## CIRCUIT :



## CODE :

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

// Define your GSM module's TX and RX pins
SoftwareSerial gsmModule(10, 11); // RX, TX

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

#define OLED_RESET -1 // Reset pin # (or -1 if not used)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#if defined(ARDUINO_SAMD_MKR1000) || defined(ESP8266)
SoftwareSerial mySerial(0, 1); // Define software serial for MKR1000 or ESP8266
#else
#define mySerial Serial1 // Use hardware serial for other boards (change to the appropriate
serial port)
#endif

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // Initialize the OLED display
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Update to use Wire for I2C
  display.display();
  delay(2000);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  Serial.println("Fingerprint Enrollment with GSM Notification");
  display.setCursor(0, 0);
  display.println("Fingerprint Enrollment");
  display.display();
```

```

finger.begin(57600);
delay(5);

// Initialize the GSM module
gsmModule.begin(9600);
delay(2000);
gsmModule.println("AT");
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read());
}
gsmModule.println("AT+CMGF=1"); // Set SMS mode to text
delay(1000);
while (gsmModule.available()) {
  Serial.write(gsmModule.read());
}

if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");
  display.setCursor(0, 20);
  display.println("Found fingerprint sensor!");
  display.display();
} else {
  Serial.println("Did not find fingerprint sensor :(");
  display.setCursor(0, 20);
  display.println("Did not find fingerprint sensor :(");
  display.display();
  while (1);
}

void loop() {
  // Enroll multiple fingerprints with different IDs
  enrollFingerprint(1);
  delay(10000); // Delay between enrolling fingerprints (10 seconds)
  enrollFingerprint(2);
  delay(10000); // Delay between enrolling fingerprints (10 seconds)
  // Add more enrollments as needed
}

```

```
void enrollFingerprint(uint16_t fingerprintID) {
    uint8_t p = -1;

    Serial.print("Place your finger on the sensor for ID #");
    Serial.println(fingerprintID);

    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("Place finger for ID #");
    display.println(fingerprintID);
    display.display();

    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                display.setCursor(0, 20);
                display.println("Image taken");
                display.display();
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println("No finger detected");
                display.setCursor(0, 20);
                display.println("No finger detected");
                display.display();
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                display.setCursor(0, 20);
                display.println("Communication error");
                display.display();
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                display.setCursor(0, 20);
                display.println("Imaging error");
                display.display();
                break;
        }
    }
}
```

default:

```
Serial.println("Unknown error");
display.setCursor(0, 20);
display.println("Unknown error");
display.display();
break;
}
}
```

```
p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    display.setCursor(0, 20);
    display.println("Image converted");
    display.display();
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    display.setCursor(0, 20);
    display.println("Image too messy");
    display.display();
    return;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    display.setCursor(0, 20);
    display.println("Communication error");
    display.display();
    return;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    display.setCursor(0, 20);
    display.println("Could not find fingerprint features");
    display.display();
    return;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    display.setCursor(0, 20);
    display.println("Could not find fingerprint features");
    display.display();
    return;
}
```

default:

```
Serial.println("Unknown error");
display.setCursor(0, 20);
display.println("Unknown error");
display.display();
return;
}
```

```
Serial.println("Remove your finger");
display.setCursor(0, 20);
display.println("Remove your finger");
display.display();
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
```

```
p = -1;
Serial.print("Place the same finger again for ID #");
Serial.println(fingerprintID);
```

```
display.clearDisplay();
display.setCursor(0, 0);
display.print("Place finger for ID #");
display.println(fingerprintID);
display.display();
```

```
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      display.setCursor(0, 20);
      display.println("Image taken");
      display.display();
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      display.setCursor(0, 20);
      display.println("No finger detected");
      display.display();
      break;
```

```
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    display.setCursor(0, 20);
    display.println("Communication error");
    display.display();
    break;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    display.setCursor(0, 20);
    display.println("Imaging error");
    display.display();
    break;
default:
    Serial.println("Unknown error");
    display.setCursor(0, 20);
    display.println("Unknown error");
    display.display();
    break;
}
}
```

```
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    display.setCursor(0, 20);
    display.println("Image converted");
    display.display();
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    display.setCursor(0, 20);
    display.println("Image too messy");
    display.display();
    return;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    display.setCursor(0, 20);
    display.println("Communication error");
    display.display();
    return;
}
```



```

case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    display.setCursor(0, 20);
    display.println("Could not find fingerprint features");
    display.display();
    return;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    display.setCursor(0, 20);
    display.println("Could not find fingerprint features");
    display.display();
    return;
}

// If the two images are successfully converted, create a fingerprint template
if (finger.createModel() == FINGERPRINT_OK) {
    if (finger.storeModel(fingerprintID) == FINGERPRINT_OK) {
        Serial.print("Fingerprint enrolled with ID #");
        Serial.println(fingerprintID);
        display.clearDisplay();
        display.setCursor(0, 20);
        display.print("Fingerprint enrolled");
        display.setCursor(0, 30);
        display.print("with ID #");
        display.println(fingerprintID);
        display.display();

        // Send a GSM message
        sendGSMMessage("+918870428899", "Fingerprint ID " + String(fingerprintID) + " enrolled.");

    } else {
        Serial.println("Enrollment failed. Please try again.");
        display.clearDisplay();
        display.setCursor(0, 20);
        display.println("Enrollment failed.");
        display.setCursor(0, 30);
        display.println("Please try again.");
        display.display();
    }
}

```

```

} else {
    Serial.println("Enrollment failed. Please try again.");
    display.clearDisplay();
    display.setCursor(0, 20);
    display.println("Enrollment failed.");
    display.setCursor(0, 30);
    display.println("Please try again.");
    display.display();
}
}

void sendGSMMessage(String phoneNumber, String message) {
    gsmModule.println("AT+CMGS=\"" + phoneNumber + "\"");
    delay(1000);
    gsmModule.println(message);
    delay(100);
    gsmModule.println((char)26); // Send Ctrl+Z (end of message)
    delay(1000);
    while (gsmModule.available()) {
        Serial.write(gsmModule.read());
    }
}

```

## EXPECTED OUTPUT :

- OLED display shows status messages during enrollment.
- SMS notifications are sent to the specified phone number after each successful enrollment.
- Confirmation messages indicating the status of fingerprint enrollment.

## RESULT :

The code, when correctly set up, allows for fingerprint enrollment and sends SMS notifications to the specified phone number upon successful enrollment.