**ESP32 BLE Wi-Fi Provisioning - Quick Guide**

**Overview:**

This code helps you **provision Wi-Fi credentials** to an ESP32 via **Bluetooth (BLE)**. Using a smartphone app, you can send Wi-Fi SSID and password to the ESP32, which then connects to the specified network.

**Key Components:**

* **Wi-Fi Provisioning Manager**: Manages provisioning over BLE.
* **BLE Scheme**: Used to send/receive credentials over Bluetooth.
* **NVS (Non-Volatile Storage)**: Stores Wi-Fi credentials for persistent connection.

**How It Works:**

1. **Provisioning Starts**: ESP32 advertises itself via BLE.
2. **App Connects**: A smartphone app connects to ESP32 over BLE and sends Wi-Fi credentials (SSID and password).
3. **Wi-Fi Connection**: ESP32 tries to connect to the provided network.
4. **Success/Failure Handling**: If successful, ESP32 connects and stores the credentials in NVS.

**Main Code Sections:**

**1. Event Handler:**

Handles various events (e.g., provisioning started, credentials received, Wi-Fi connection status).

Key Events:

* **Provisioning Start**: Starts BLE provisioning.
* **Credentials Received**: Logs SSID and password.
* **Connection Success/Failure**: Handles whether Wi-Fi credentials work or not.

void event\_handler(void\* arg, esp\_event\_base\_t event\_base, int32\_t event\_id, void\* event\_data) {

// Handle Wi-Fi and Provisioning events

}

**2. Start Wi-Fi Provisioning:**

Initializes everything (Wi-Fi, NVS, event handlers), checks if already provisioned, and if not, starts BLE provisioning.

Key Steps:

* Initializes **NVS** for storing credentials.
* Sets up **Wi-Fi** and **event handling**.
* **BLE** provisioning starts if no credentials are found.

void start\_wifi\_provisioning() { // Initialize NVS, Wi-Fi, and start BLE provisioning if needed

}

**3. Main Function:**

The app\_main() function simply calls start\_wifi\_provisioning() to begin the process.

void app\_main(void) {

start\_wifi\_provisioning();

}

**How to Use:**

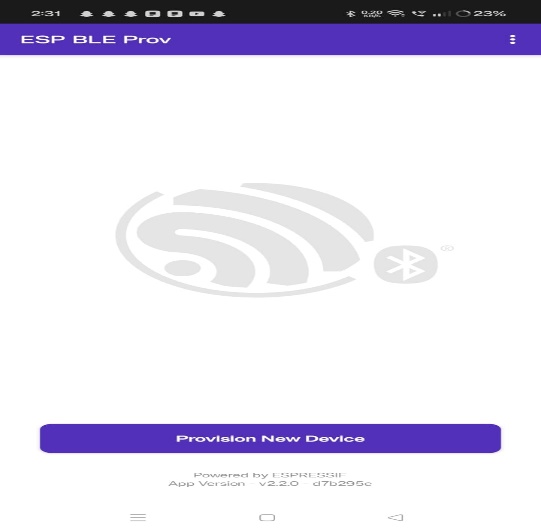
1. **Flash the Code**: Use ESP-IDF to flash the code to your ESP32.
2. **Open BLE Provisioning App**: Use the ESP BLE Provisioning App from the Play Store or App Store.
3. **Connect and Provision**:
   * The BLE service is advertised as **siva\_1**.
   * POP (Proof of Possession) is set as **abcd123**.
   * Enter Wi-Fi SSID and password when prompted by the app.
4. **ESP32 Connects**: Once credentials are provided, ESP32 attempts to connect to the Wi-Fi.

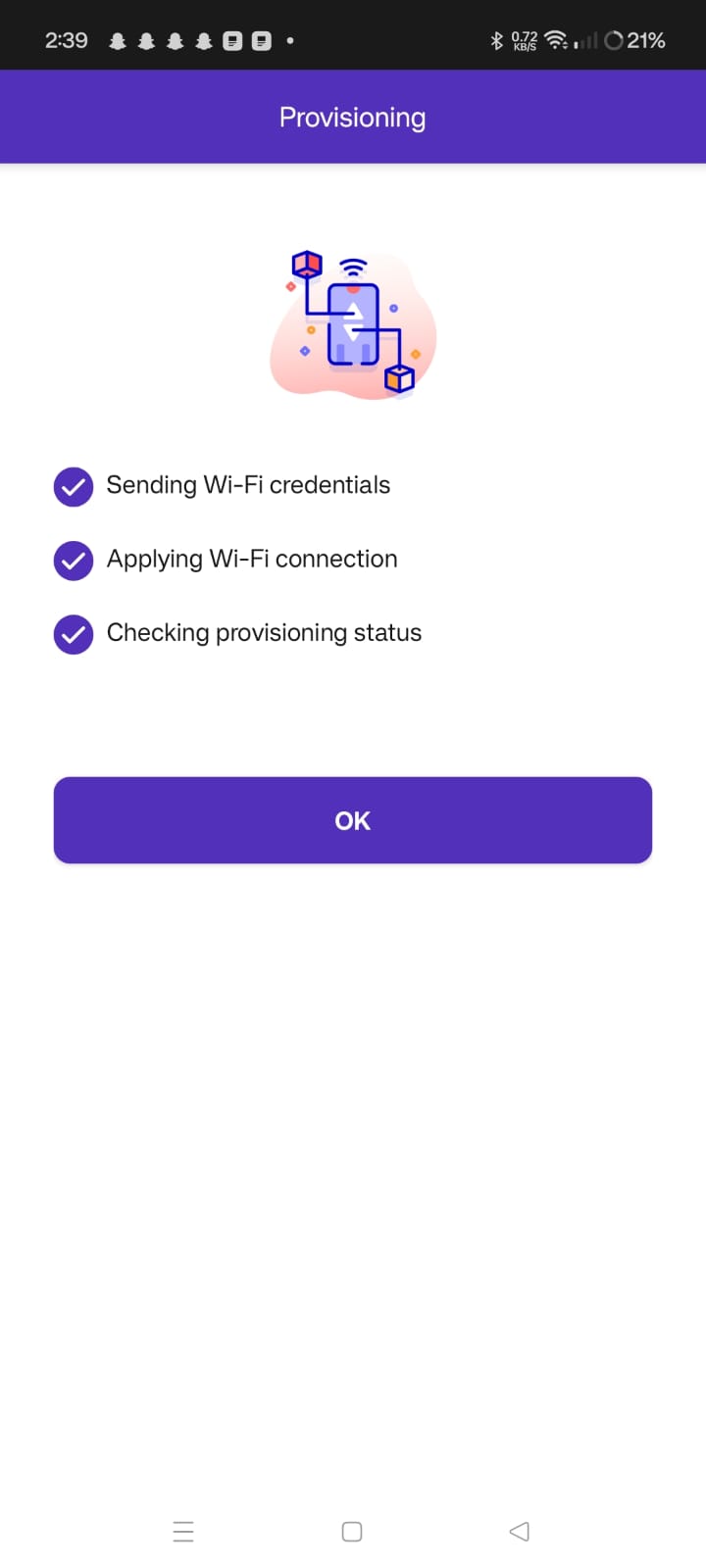
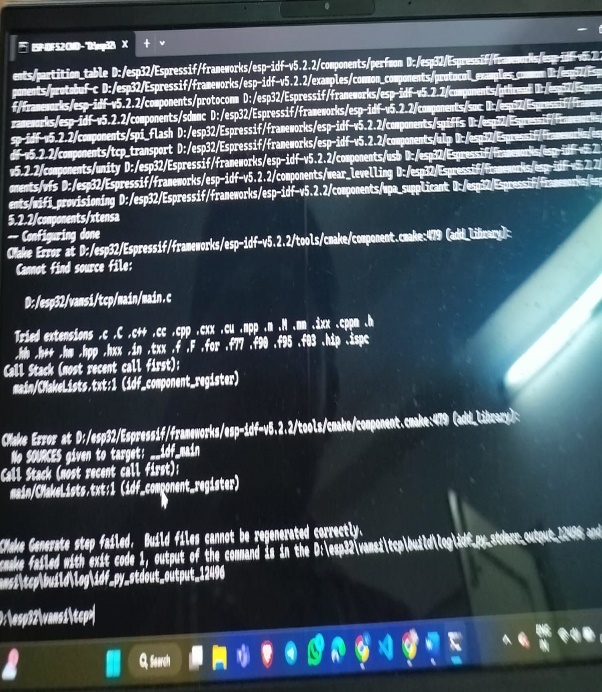
**Customization:**

* **Change Service Name**: Modify service\_name = "siva­\_r custom BLE service name.
* **Change POP**: Modify pop = "abcd1234" to set a different Proof of Possession.

That’s it! The code efficiently handles Wi-Fi provisioning over BLE, making it easy to connect your ESP32 to a network using a mobile app.

**Working:**

 ****

****