**ESP32 Wi-Fi Provisioning via BLE (Bluetooth Low Energy) – Arduino IDE**

The Wi-Fi provisioning service allows you to configure Wi-Fi credentials over Bluetooth Low Energy. This is a great option for IoT projects that require Wi-Fi credentials to connect to the internet, without the need to hard-code them on the Arduino sketch while developing.

**Wi-Fi Provisioning**

Wi-Fi provisioning is the process of connecting a new Wi-Fi device (station) to a Wi-Fi network (access point). In this case, we want to connect an ESP32 to a Wi-Fi network. The provisioning process involves loading the ESP32 with the name of the network (SSID) and password that we want to connect to.

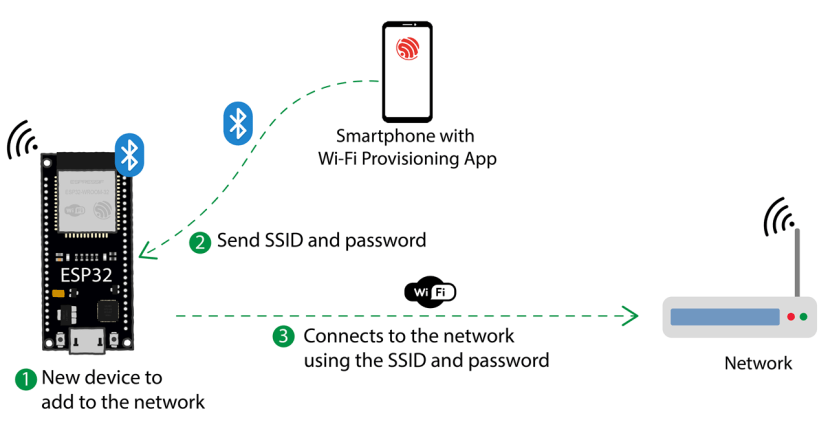
The ESP32 supports Wi-Fi provisioning over SoftAP (access point) or via Bluetooth Low Energy. In this tutorial, we’ll cover Wi-Fi provisioning via Bluetooth.

**Provisioning over BLE**

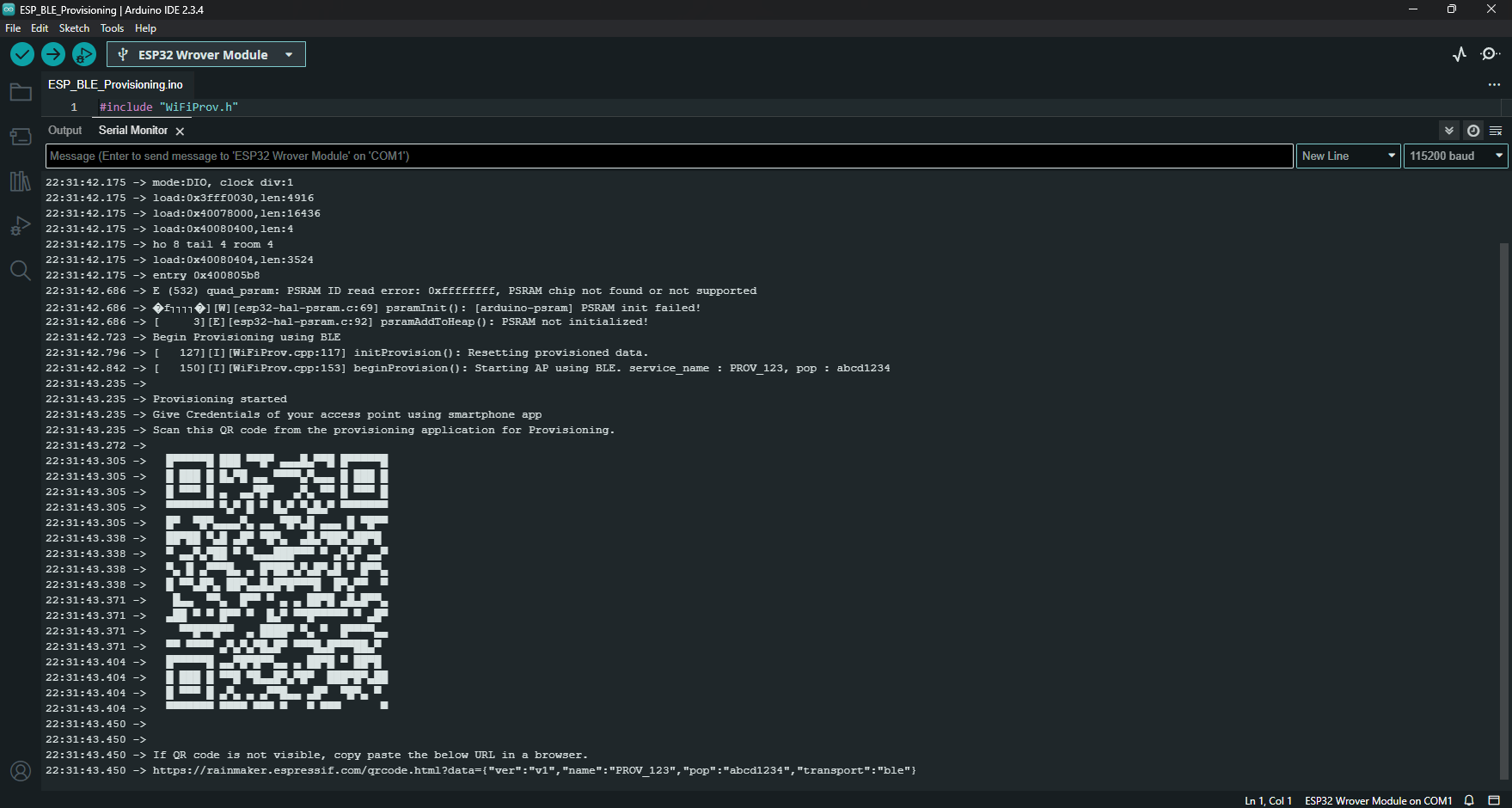
To provision the ESP32 via BLE, we need to use another BLE-enabled device, usually a smartphone to connect to the ESP32 via BLE and send the Wi-Fi credentials. We can use an Android or iOS app or a Web Bluetooth app.

Espressif developed Android and iOS apps that support Wi-Fi provisioning for its devices like the ESP32 boards. Alternatively, you can create your own BLE app.

After the provisioning process, the ESP32 can connect to the desired Wi-Fi network with the provided network credentials.

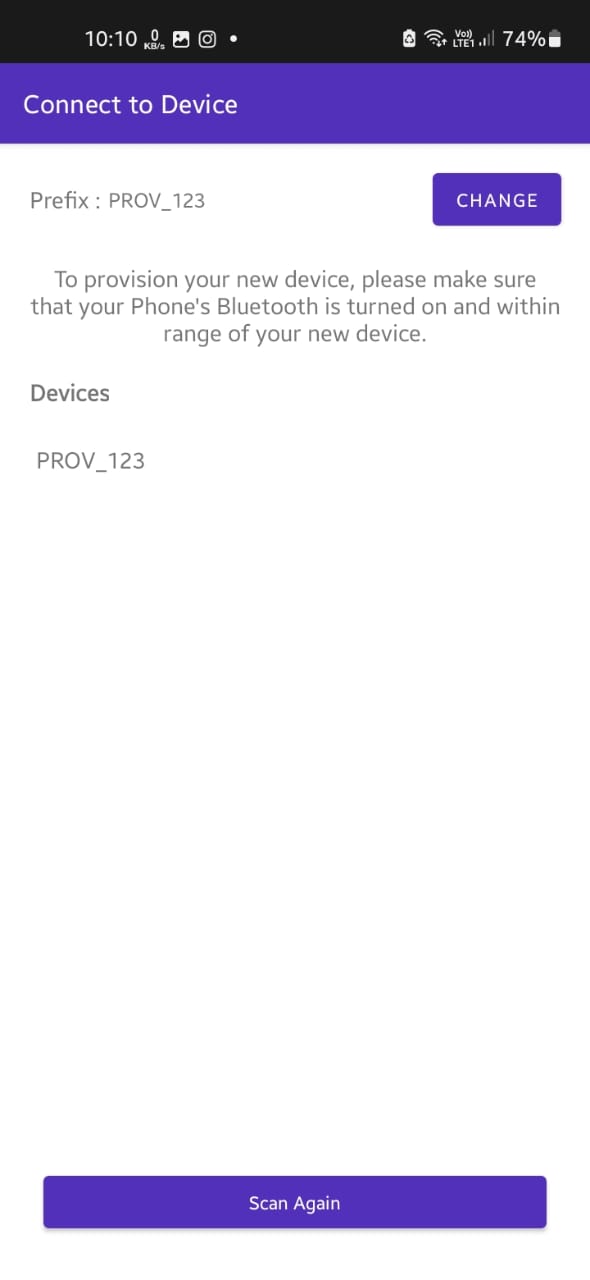


**Testing the Code:**

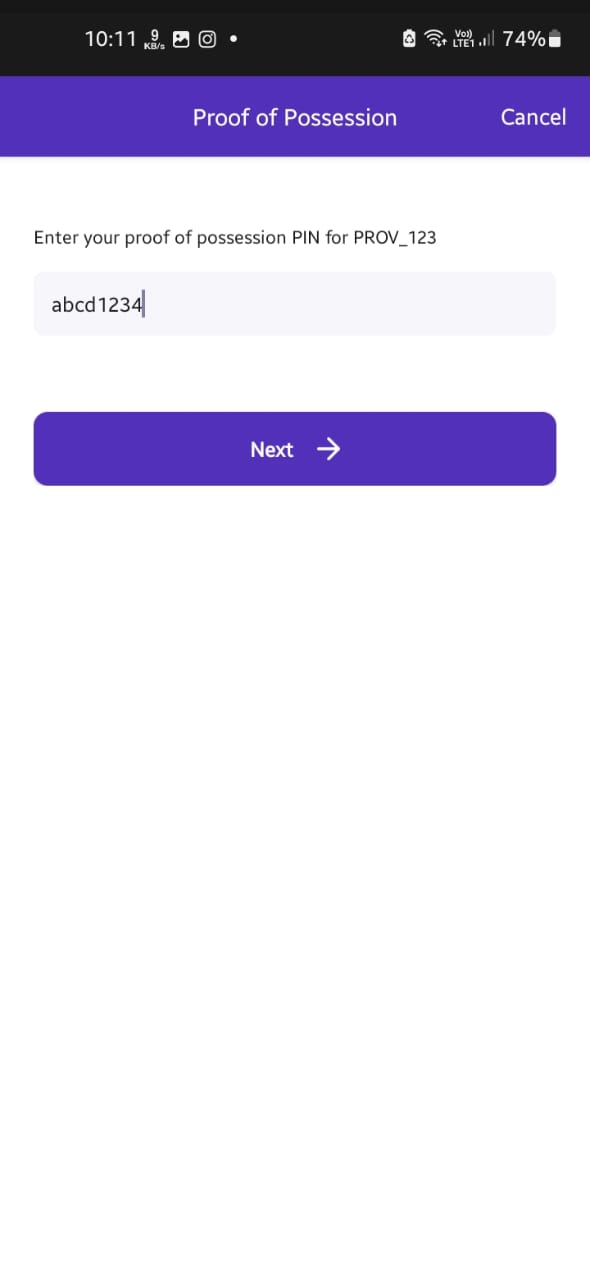
After uploading the code to the ESP32, open the Serial Monitor at a baud rate of 115200. Press the ESP32 RST button, so it starts running the code.

Now, open the Espressif WiFi Provisioning App on your smartphone and click on **Provision Device**. It has the option to scan the QR code, but I couldn’t make it work. Click on “**I don’t have a QR code**“.

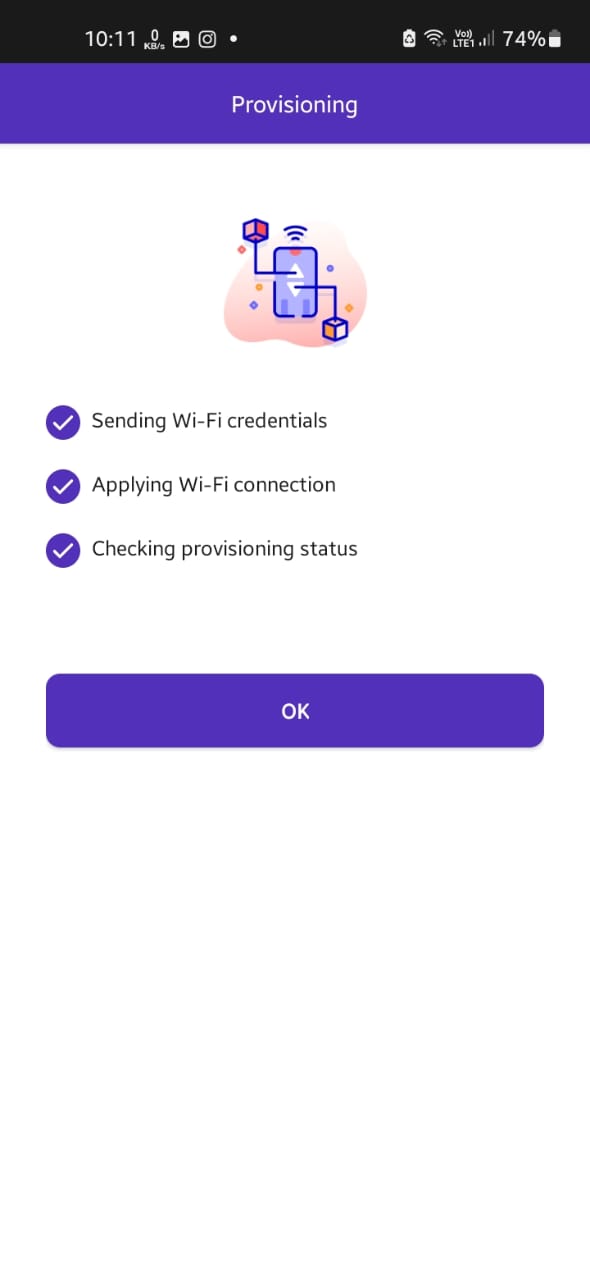
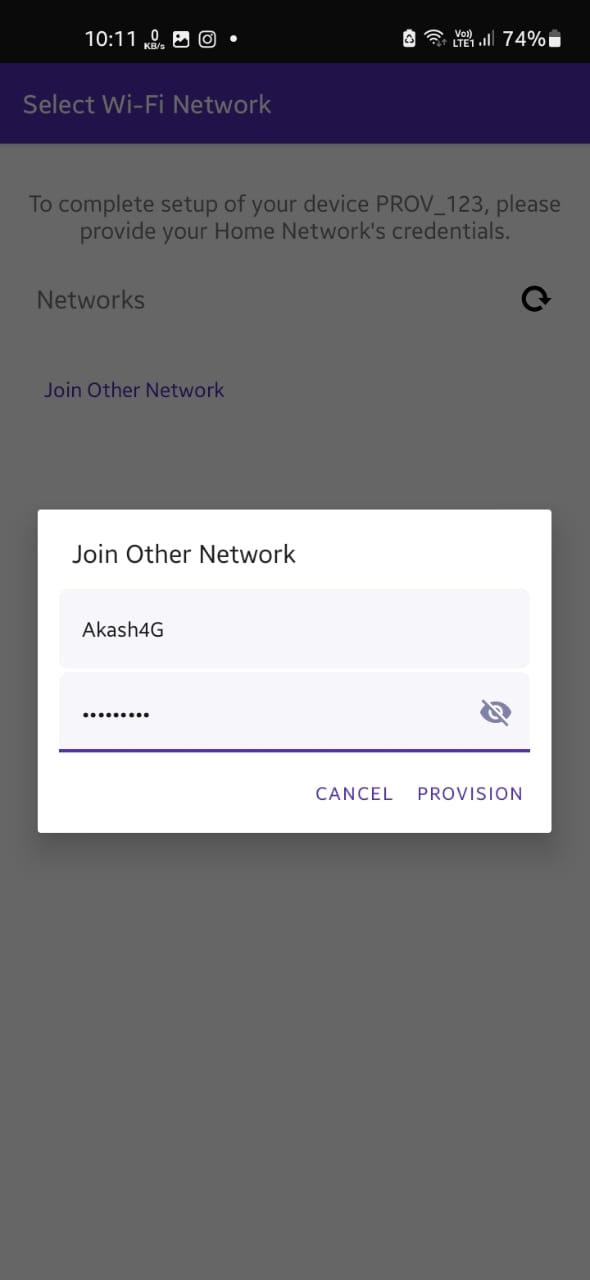
It will list all Bluetooth Devices within its range with the “**PROV\_**” prefix. Our code sets the ESP32 as a BLE Device called “**PROV\_123**“—click on that device.



Then, you need to enter the proof of possession—it must be the same used in the code. We’re using the default values, so it will be *abcd1234*. After, it will list all Wi-Fi networks within its range (if it doesn’t list your network, you may need to enter it manually by clicking on “Join Other Network”).



Finally, enter the password for your Wi-Fi network and click Connect. The Wi-Fi credentials should be sent to the ESP32 after a few seconds.



On the Serial monitor, you can see that the ESP32 received the Wi-Fi credentials and connected successfully to the network (it printed its IP address on the network it connected to).

