**Nineti GmbH**

**Internet of Things Internship - Assignment**

**Name: ANISSH BABU**

**Email:** [**anisshbabuln@gmail.com**](mailto:anisshbabuln@gmail.com)

**Linkedin:** [**www.linkedin.com/in/anisshbabu**](http://www.linkedin.com/in/anisshbabu)

**TASK 2:**

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

To set up the ESP32 using Bluetooth Low Energy (BLE), we must employ a different BLE-capable gadget, typically a smartphone, to link the ESP32 to the BLE device and transmit the Wi-Fi login details. This can be accomplished through an Android or iOS application, or a Web Bluetooth application.

Espressif has developed applications for both Android and iOS that facilitate the setup of Wi-Fi on its devices, such as the ESP32 boards. Alternatively, you might consider developing your own BLE application.

Once the setup is complete, the ESP32 will be able to connect to the selected Wi-Fi network using the provided login information.

Here's a brief overview of the process:

You're setting up a new gadget that needs to be linked to an established network, in this scenario, an ESP32.

You link the ESP32 to the network through Bluetooth Low Energy (BLE). In the Wi-Fi setup application, you input and transmit the network's SSID and password to enable the ESP32 to connect.

Once the ESP32 is connected, it can perform any Wi-Fi-related activities it requires.

Wi-Fi Setup Application

For the purpose of linking and transmitting the login details to the ESP32 through Bluetooth Low Energy (BLE), we will employ the Wi-Fi Setup Application created by Espressif (or, you may opt to develop your own BLE application).

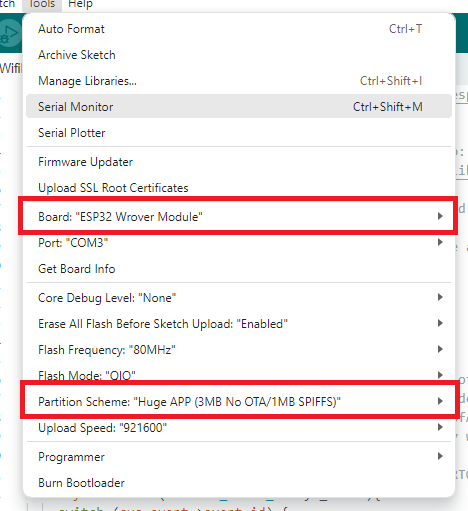
Downloading and Installing the Wi-Fi Setup Application

Prior to moving forward, ensure that the Wi-Fi Setup Application is installed on your mobile device:

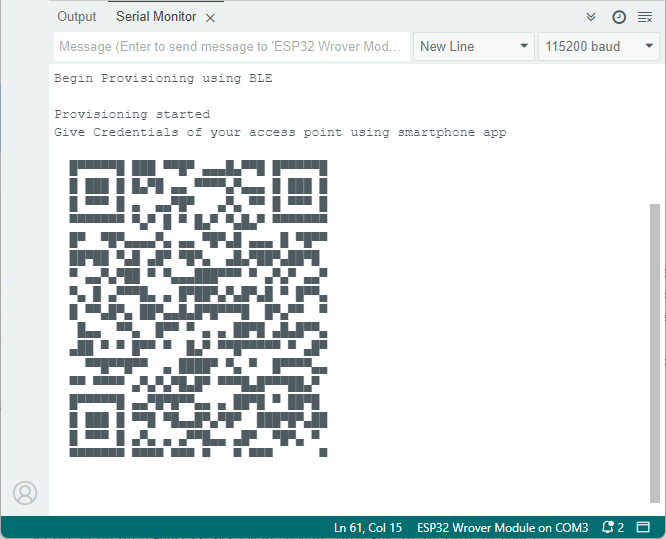
* [Wi-Fi Provisioning App for Android devices](https://play.google.com/store/apps/details?id=com.espressif.provble) – ([App source code](https://github.com/espressif/esp-idf-provisioning-android))
* [Wi-Fi Provisioning App for iOS devices](https://apps.apple.com/in/app/esp-ble-provisioning/id1473590141) – ([App source code](https://github.com/espressif/esp-idf-provisioning-ios))

**Instructions for Uploading the Code:**

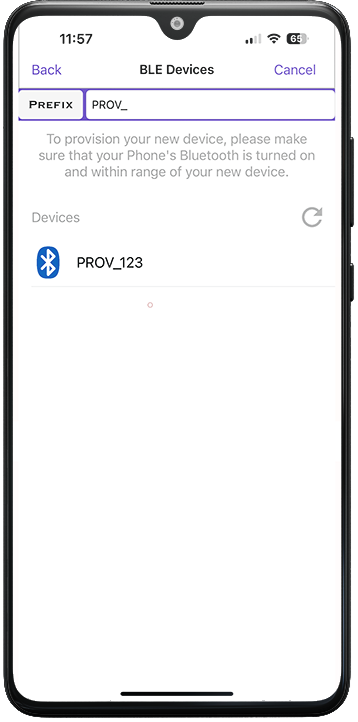
This code takes up a lot of space and may not be able to flash with the default settings on some chips. If you see an Error like this: “Sketch too big” during the uploading process, in Arduino IDE go to Tools > Partition scheme > choose anything that has more than 1.4MB APP, for example: “Huge APP (3MB No OTA/1MB SPIFFS“.



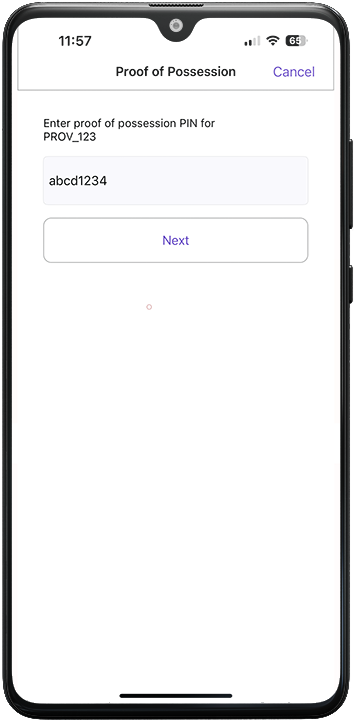
Testing the Code After uploading the code to the ESP32, open the Serial Monitor at a baud rate of 115200. Press the ESP32 RST button, so it starts running the code. Note: you may get a PSRAM error. You can ignore it. You’ll get something similar on your Serial Monitor.



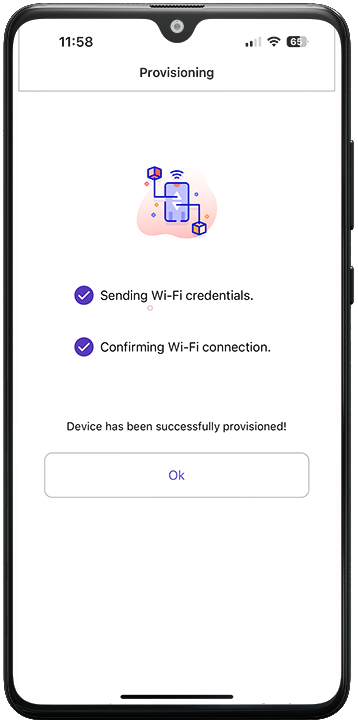
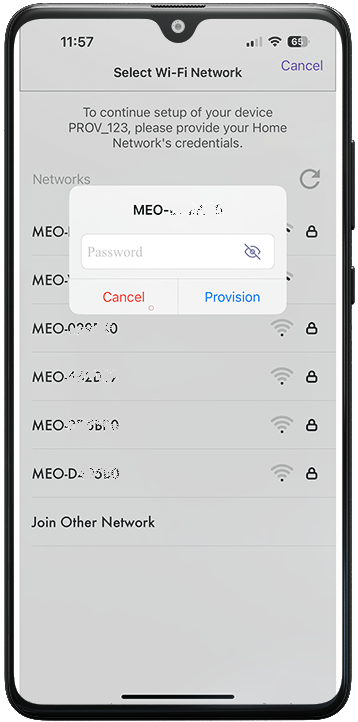
Now, open the Espressif WiFi Provisioning App on your smartphone and click on Provision Device. It has the option to scan the QR code, but I couldn’t make it work. Click on “I don’t have a QR code“. It will list all Bluetooth Devices within its range with the “PROV\_” prefix. Our code sets the ESP32 as a BLE Device called “PROV\_123“—click on that device.



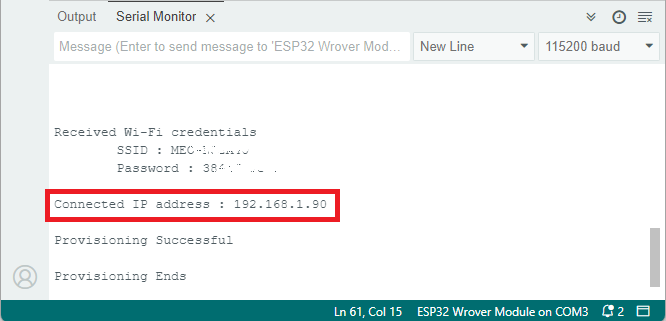
Then, you need to enter the proof of possession—it must be the same used in the code. We’re using the default values, so it will be *abcd1234*. After, it will list all Wi-Fi networks within its range (if it doesn’t list your network, you may need to enter it manually by clicking on “Join Other Network”).



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).



Thus we have connected the wifi with the credentials.