Assignment-Embedded

Create a New Project

Navigate to your ESP-IDF workspace, clone the provisioning example, and create a project:

```
cd ~/esp
mkdir ble_provisioning_project
cd ble_provisioning_project
git clone https://github.com/espressif/esp-idf.git
cd esp-idf/examples/provisioning/ble_prov
```

Modify the BLE Provisioning Example

Here's a simplified version of the code from the example. This code provisions Wi-Fi credentials over BLE and connects to the network.

```
#include "esp_log.h"
#include "nvs_flash.h"
#include "esp_wifi.h"
#include "esp_event.h"
#include "protocol_examples_common.h"
#include "esp_netif.h"
#include "wifi_provisioning/manager.h"
#include "wifi_provisioning/scheme_ble.h"

static const char *TAG = "app";

void app_main(void)
{
    // Initialize NVS
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret ==
ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK(ret);

    ESP_LOGI(TAG, "Initializing WiFi and BLE provisioning...");
```

```
    // Initialize networking stack
    ESP_ERROR_CHECK(esp_netif_init());
    ESP_ERROR_CHECK(esp_event_loop_create_default());

    // Start provisioning service over BLE
    wifi_prov_mgr_init(NULL);

    // Set provisioning scheme to BLE

wifi_prov_scheme_ble_set_service_uuid("000018FF-0000-1000-8000-00805
F9B34FB");

    // Start the provisioning service
    wifi_prov_mgr_start_provisioning(WIFI_PROV_SECURITY_1,
"prov123", NULL, NULL);

    ESP_LOGI(TAG, "BLE Provisioning started. Use the app to
provision Wi-Fi.");
}
```

Configure the Project

- Modify the `sdkconfig` file as needed for your Wi-Fi credentials, logging, and BLE options.

In the terminal, navigate to the project directory and run:
bash
Copy code
```
idf.py menuconfig
```

- Here, you can customize the BLE name, security settings, etc.

Build and Flash the Code

Once the code is in place, build and flash it to your ESP32 board:

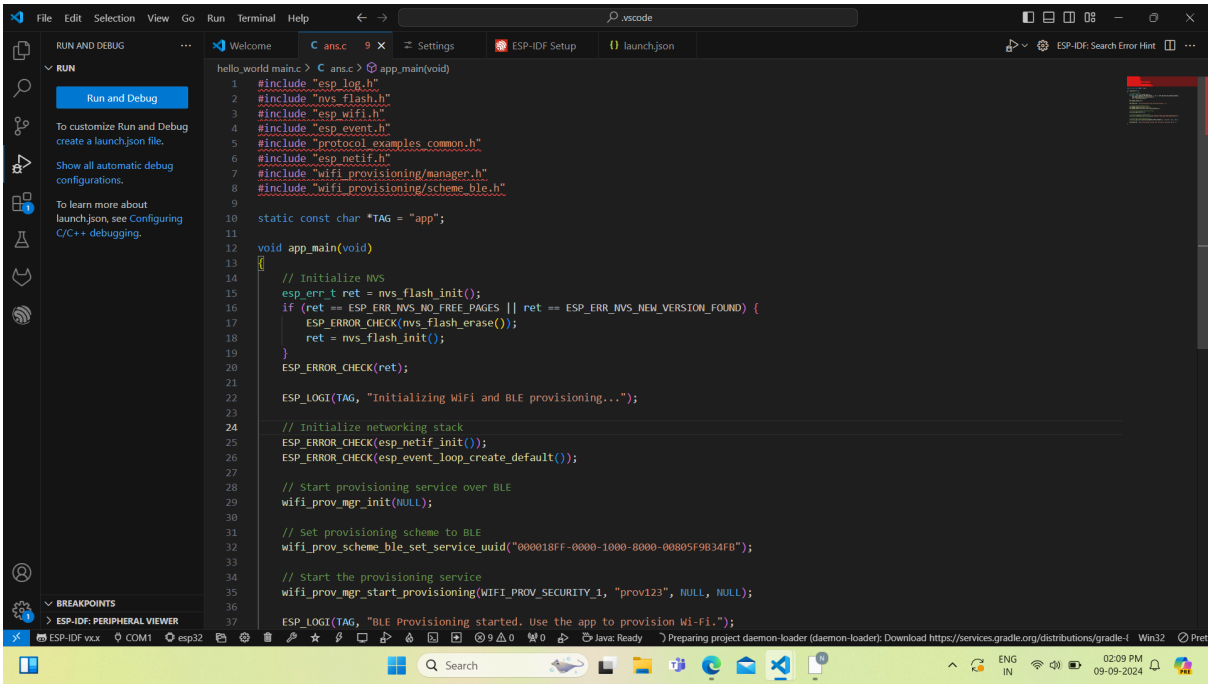
bash
Copy code
```
idf.py build
idf.py -p /dev/ttyUSB0 flash
```

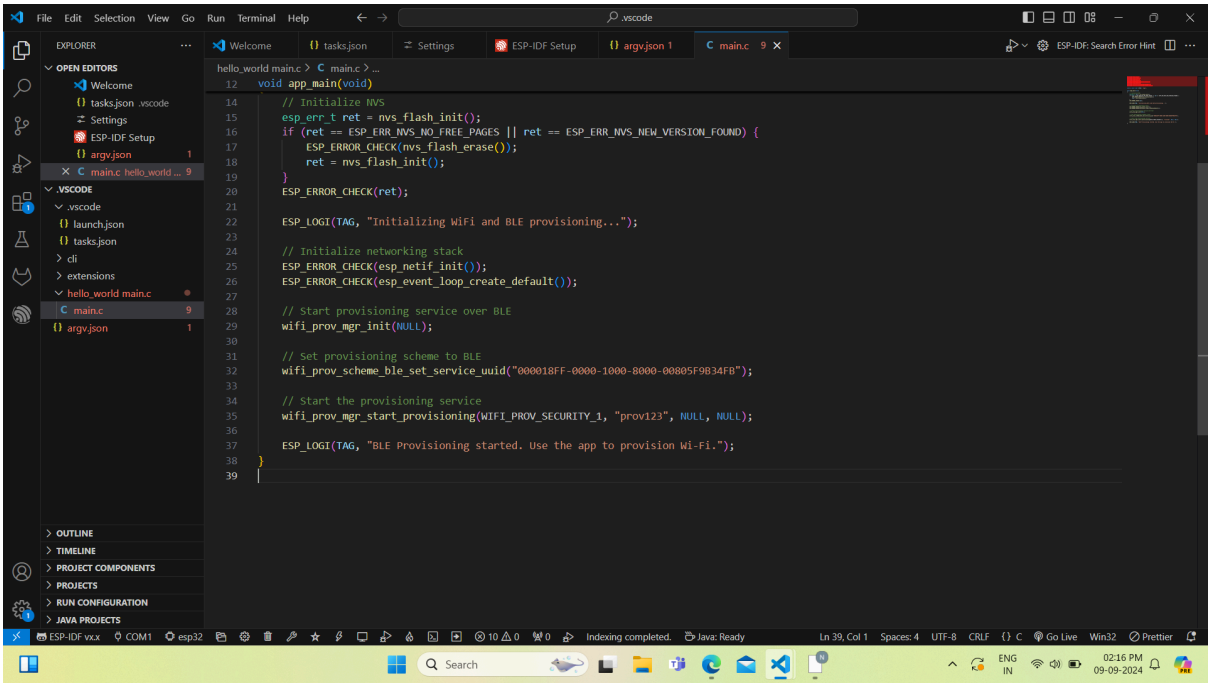
Replace `/dev/ttyUSB0` with the correct serial port for your device.

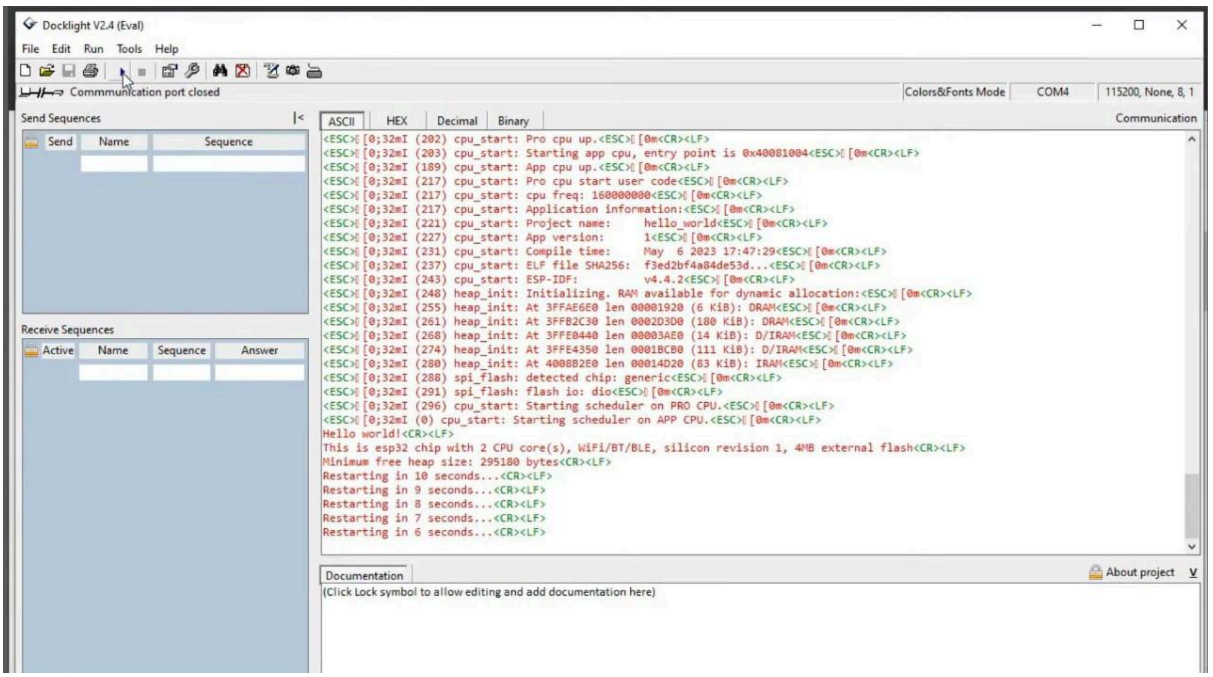
Test the Provisioning Process

- After flashing, open the **ESP BLE Provisioning** app on your smartphone.
- Connect to your ESP32 device via BLE.
- Enter the Wi-Fi credentials through the app.

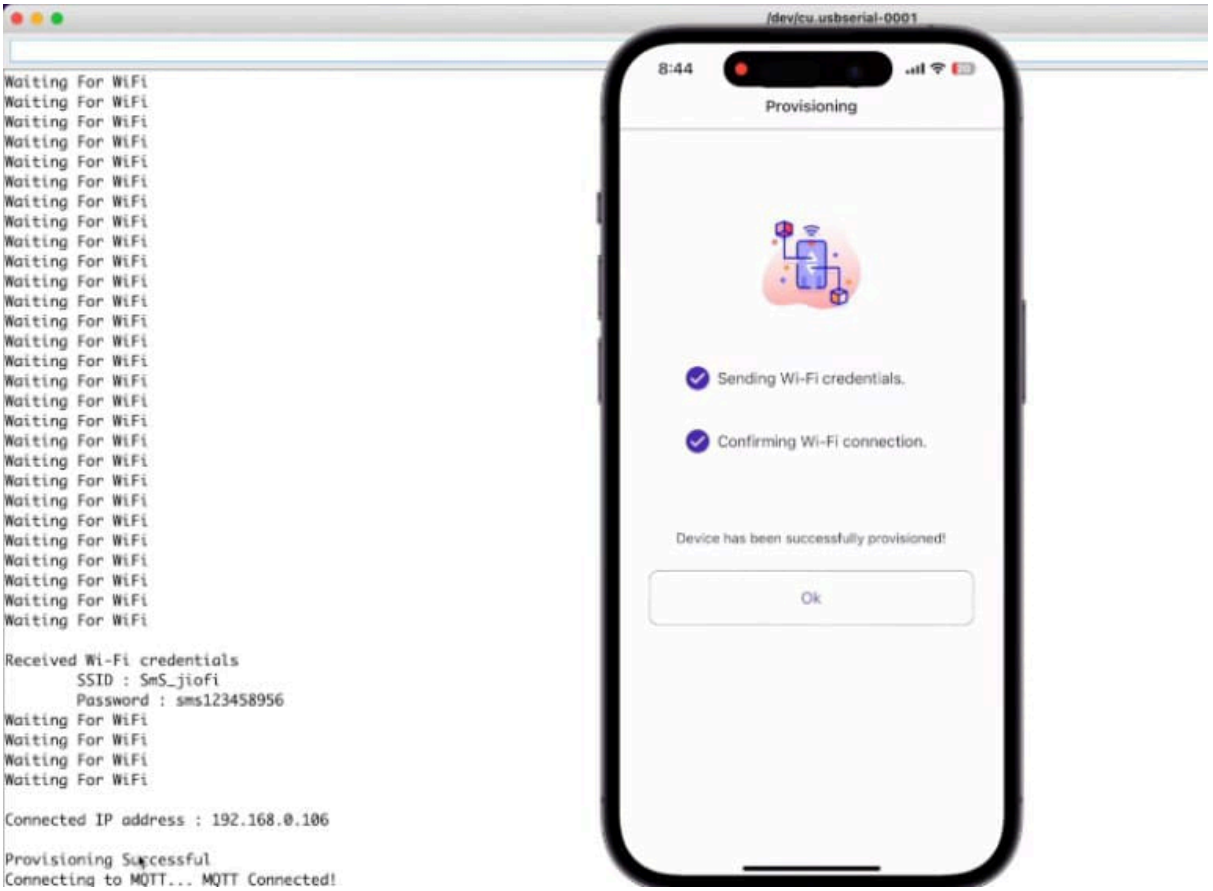
The output in the terminal should confirm that the provisioning process is successful, and your ESP32 will connect to the specified Wi-Fi network.







Scan the qr and connect to the mobile application called ESP BLE and the final out put is



```
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi

Received Wi-Fi credentials
        SSID : SmS_jiofi
        Password : sms123458956
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi
Waiting For WiFi

Connected IP address : 192.168.0.106

Provisioning Successful
Connecting to MQTT... MQTT Connected!
```

✓ Sending Wi-Fi credentials.

✓ Confirming Wi-Fi connection.

Device has been successfully provisioned!