**Lab Exercise 20- ESP32 Http Wi-Fi using Blynk Web and Mobile App to ON OFF LED and Buzzer**

In this lab exercise, we will control an LED connected to an **ESP32** using **Blynk** and **Wokwi** (an online simulator). The ESP32 will connect to the **Blynk cloud** over Wi-Fi, allowing you to control the LED via the **Blynk app** on your smartphone.

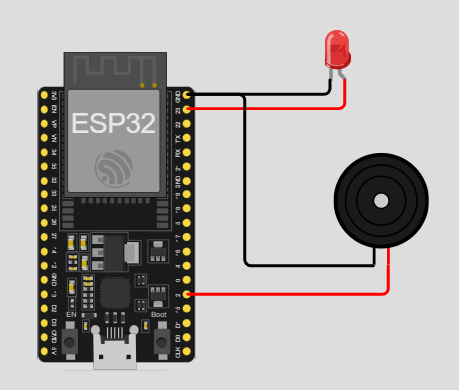
**Objective**

To control an LED connected to the ESP32 through the **Blynk** app using **Wi-Fi** in the **Wokwi** simulator.

**Materials Required**

* ESP32 Development Board (in Wokwi)
* LED (in Wokwi)
* Wokwi Account (<https://wokwi.com/>)
* Blynk App installed on your smartphone (available on Google Play or the App Store)
* Arduino IDE or Wokwi's built-in editor
* Blynk Auth Token (from the Blynk app)

**Circuit Diagram in Wokwi**

****

**2. Set Up the Blynk App**

1. Download and install the **Blynk app** from the **Google Play Store** or **Apple App Store**.
2. **Create a New Project** in the Blynk app:
   * Select **ESP32** as the device.
   * Choose **Wi-Fi** as the connection type.
   * You'll receive an **Auth Token** via email. Copy this token for use in the code.
3. **Add a Button Widget** to the Blynk app:
   * Click on the "+" icon to add a widget.
   * Add a **Button** widget to control the LED.
   * Set the button's **output pin** to **V1** (Virtual Pin 1).
   * Set the button mode to **Switch** (so it stays ON/OFF when you press it).

**3. Write the ESP32 Code in Wokwi**

1. In Wokwi, use the following code to set up the ESP32 with Blynk.

#define BLYNK\_TEMPLATE\_ID "TMPL6h8SUt2md"

#define BLYNK\_TEMPLATE\_NAME "LEDMOBWEB"

#define BLYNK\_AUTH\_TOKEN "tZ9GOrPIAZmBUtuXKi74hxHJNb-S9NWe"

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#define ssid "Wokwi-GUEST"

#define pass ""

#define pin\_led 23

#define pin\_buz 2

BLYNK\_WRITE(V0)

{

    int pinValue=param.asInt(); //assigning incoming value from pin1 to a variable

    digitalWrite(pin\_led,pinValue); //process received value

    tone(pin\_buz, 1000); // Play 1000 Hz for 1 second

    delay(1000);           // Wait for 1 second

}

BLYNK\_WRITE(V1)

{

    noTone(pin\_buz);

}

void setup() {

  // put your setup code here, to run once:

  Serial.begin(115200);

  Serial.println("Hello, ESP32!");

  pinMode(pin\_led, OUTPUT);

  Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

}

void loop() {

  Blynk.run();

}

**4. Simulate and Run the Project in Wokwi**

1. **Run the simulation** by clicking the green "Play" button in Wokwi.
2. The ESP32 will connect to the **Wokwi-GUEST** Wi-Fi network.
3. Open the **Serial Monitor** in Wokwi to see the ESP32 connecting to Wi-Fi and Blynk.

**5. Control the LED with the Blynk App**

1. Open the **Blynk app** on your smartphone.
2. Press the **Button** in the app to control the LED. The button will toggle the LED on and off.

**Explanation of the Code**

1. **Blynk Setup**:
   * The **Blynk Auth Token** is used to authenticate your ESP32 with the Blynk cloud.
   * The ESP32 connects to the **Wokwi-GUEST** Wi-Fi network (no password required).
2. **LED Control**:
   * The **BLYNK\_WRITE(V1)** function listens for changes on **Virtual Pin 1 (V1)**, which is linked to the button in the Blynk app.
   * If the button is pressed (value is 1), the LED is turned **ON**. When the button is released (value is 0), the LED is turned **OFF**.
3. **Serial Monitor**:

The Serial Monitor prints messages whenever the LED is turned on or off, helping to debug and visualize the status.

**Additional Features**

You can expand this project by:

* Adding multiple LEDs or other components (such as relays or motors) and controlling them through different virtual pins (e.g., V2, V3).
* Using other widgets like sliders or gauges to control brightness or monitor sensor data.

**Conclusion**

In this lab exercise, you've learned how to control an LED connected to an ESP32 using **Blynk** and **Wokwi**. You created a simple Wi-Fi-based system that allows you to control the LED from your smartphone using the Blynk app. This is a foundational project for building more complex IoT systems where you can remotely control devices and monitor sensors.