Soham Thakur D10B/68

Experiment No. 5:Interface an LED with NodeMCU and Control via Blynk App

Aim: The aim of this experiment is to control an LED using the Blynk App and NodeMCU (ESP8266).

Apparatus Required:

- **NodeMCU (ESP8266) – 1 unit**
- **LED – 1 unit**
- **Resistor (220Ω) – 1 unit**
- **Jumper wires**
- **Breadboard**
- **Smartphone (Android/iOS) with Blynk app installed**
- **USB Cable for powering the NodeMCU and uploading the code**
- **Internet connection (for NodeMCU and smartphone to connect)**
- **Blynk app account (for accessing the data)**

MCU
B
Blynk Features
Device Provisioning
Blueprints & Templates
Network Co-Processor
Web & Mobile Apps

Theory: Blynk is an IoT (Internet of Things) platform that allows users to control and monitor devices remotely. It provides a user-friendly interface that enables interaction with hardware devices like Arduino, ESP8266, ESP32, and Raspberry Pi. Blynk allows you to create projects and control hardware through a mobile app.

Working:

- **Mobile App:** You create a project on the Blynk app by selecting a hardware device (e.g., NodeMCU, Arduino) and setting up widgets to control the LED.
- **Blynk Cloud:** The app communicates with Blynk Cloud, a remote server, which acts as a bridge between your mobile app and the hardware. The cloud sends and receives data between your app and NodeMCU via the internet.
- **Hardware Device:** The hardware (NodeMCU, ESP8266) runs a Blynk sketch, which connects to the Blynk Cloud using Wi-Fi. It receives control commands from the app.
- **LED:** The LED is connected to the NodeMCU and can be turned ON or OFF using the Blynk app.
- **NodeMCU:** It receives the ON/OFF command from the Blynk app and controls the LED accordingly.
- **Blynk App:** The app sends control commands to turn the LED ON or OFF remotely.

Key Features of Blynk:

- **Remote Control:** You can control the LED remotely using the app.
- **Widgets:** Blynk offers various widgets such as buttons, sliders, graphs, and more to control the hardware.
- **Easy to Use:** No advanced programming skills are required to set up the mobile app interface.
- **Real-Time Control:** It supports real-time device control.
- **Cloud Integration:** Data is sent to Blynk's cloud servers, which can be accessed from anywhere.

Stepwise Procedure:

Step 1: Setting Up the Blynk App

1. **Install Blynk App:**
    - Download and install the Blynk app on your smartphone (available for both Android and iOS).
2. **Create an Account and New Project:**
    - Open the app and sign up (if you don’t have an account).
    - Create a new project in the app. Choose ESP8266 as the device.
    - Set up a new project and note the Auth Token that Blynk sends to your email. You will need this token in your code.
3. **Add Widgets:**
    - Add a Button widget to control the LED.
    - Set the output pin for the button widget.
    - Link the button to a virtual pin (e.g., V1 for LED control).
4. **Get the Auth Token:**
    - You will receive an Auth Token via email. This token is used to authenticate the device to the Blynk server.

Step 2: Connecting the LED to NodeMCU

1. **Wiring the LED:**
    - Connect the longer leg (anode) of the LED to D2 (GPIO4) pin of NodeMCU via a 220Ω resistor.
    - Connect the shorter leg (cathode) of the LED to the GND pin of NodeMCU.

Step 3: Installing the Required Libraries

1. **Install Libraries in Arduino IDE:**
    - Open the Arduino IDE and go to Sketch > Include Library > Manage Libraries.
    - Install the Blynk library for NodeMCU.

Step 4: Writing the Code

```
#define BLYNK_TEMPLATE_ID "TMPL37IyLoKYG"
#define BLYNK_TEMPLATE_NAME "Soham"

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "IRr-WUcNvCkuERt3QMwWz8rjKA1IZ2M9";
char ssid[] = "Arjun Prabhu";
char pass[] = "Arjun0611";
#define LED_PIN 4
// Blynk Virtual Pin
#define VIRTUAL_PIN_SWITCH V3
bool buttonState = false;
BlynkTimer timer;
BLYNK_WRITE(VIRTUAL_PIN_SWITCH) {
  buttonState = param.asInt(); // param.asInt() will be 1 when the button is ON, 0
when OFF
  if (buttonState == 1) {
    digitalWrite(LED_PIN, HIGH); // Turn ON the LED
  } else {
    digitalWrite(LED_PIN, LOW);  // Turn OFF the LED
  }
}
void setup() {
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW); // Ensure the LED is OFF initially
  Blynk.begin(auth, ssid, pass);
}
void loop() {
  Blynk.run();
}
```

Step 5: Uploading the Code to NodeMCU

1. Select the correct Board (NodeMCU 1.0) and Port in the Tools menu of Arduino IDE.
2. Click on the Upload button to upload the code to the NodeMCU.

Step 6: Control LED on Blynk App

1. Open the Blynk app on your smartphone.
2. Use the button widget to turn the LED ON or OFF.
3. The LED should respond in real time based on your command in the app.

Output: