#define BLYNK\_TEMPLATE\_ID "TMPL3zjhMENXa" // Replace with your Template ID

#define BLYNK\_TEMPLATE\_NAME "Automation" // Replace with your Template Name

#define BLYNK\_AUTH\_TOKEN "1mEjcltvb\_83LoT0VZDrYUAyrqDrIL-f" // Replace with your Auth Token

#define BLYNK\_DEVICE\_NAME "Automation" // Replace with your Device Name

#define BLYNK\_PRINT Serial

#include <gpio.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

//#include <ESP8266WiFi.h>

#include <PubSubClient.h>

/\*// WiFi credentials

const char\* ssid = "Cmf";

//"Airtel\_anish\_2075"; // Replace with your WiFi SSID

const char\* password = "03101992";

//"Air@37616"; // Replace with your WiFi password\*/

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] ="Cmf"; // Your Wifi Name

//"Airtel\_anish\_2075\_5G","Air@37616")

char pass[] = "03101992"; // Your Wifi Password

/\*

// Variables to store the state of the relays

bool relay1State = LOW;

bool relay2State = LOW;\*/

// Relay pins

#define RELAY1\_PIN 12

#define RELAY2\_PIN 13

/\*

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

// Initialize Serial Monitor

Serial.begin(9600);

delay(10);

// Initialize relay pins

pinMode(RELAY1\_PIN, OUTPUT);

pinMode(RELAY2\_PIN, OUTPUT);

digitalWrite(RELAY1\_PIN, relay1State); // Relay 1 OFF by default

digitalWrite(RELAY2\_PIN, relay2State); // Relay 2 OFF by default

// Connect to WiFi

connectToWiFi();

// Initialize Blynk with Template ID and Auth Token

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

Serial.println("Setup complete!");

}

// Function to connect to WiFi

void connectToWiFi() {

Serial.print(ssid);

//("Connecting to WiFi");

WiFi.begin(ssid, pass);

//"Airtel\_anish\_2075\_5G","Air@37616");

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println("\nConnected to WiFi!");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

}

\*/

/\*void loop() {

Blynk.run(); // Run Blynk

}\*/

// Blynk Virtual Pin V1: Control Relay 1

BLYNK\_WRITE(V1)

{

int value = param.asInt();

Serial.println(value);

if(value == 1)

{

digitalWrite(RELAY1\_PIN, LOW);

Serial.println("LED ON"); //Setting Digital PIN as LOW to turn ON Device if relay module is "active low"

}

if(value == 0)

{

digitalWrite(RELAY1\_PIN, HIGH);

Serial.println("LED OFF");//Setting Digital PIN as HIGH to turn OFF Device if relay module is "active low"

}

}

/\*relay1State = param.asInt(); // Get the value from the Blynk app

digitalWrite(RELAY1\_PIN, relay1State ? HIGH : LOW);

Serial.println(relay1State ? "Relay 1 turned ON" : "Relay 1 turned OFF");\*/

// Blynk Virtual Pin V2: Control Relay 2

/\*BLYNK\_WRITE(V2) {

relay2State = param.asInt(); // Get the value from the Blynk app

digitalWrite(RELAY2\_PIN, relay2State ? HIGH : LOW);

Serial.println(relay2State ? "Relay 2 turned ON" : "Relay 2 turned OFF");

}\*/

void setup()

{

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

pinMode(RELAY1\_PIN,OUTPUT); //GPIO 16 (equivalent to PIN 16 of Arduino)

/\*pinMode(D1,OUTPUT); //GPIO 05 (equivalent to PIN 05 of Arduino)

pinMode(D2,OUTPUT);//GPIO 04 (equivalent to PIN 16 of Arduino)

pinMode(D3,OUTPUT);//GPIO 00 (equivalent to PIN 00 of Arduino)\*/

}

void loop()

{

Blynk.run();

}

MQTT Query

-------------

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// Wi-Fi credentials

const char\* ssid = "Cmf";

//"Airtel\_anish\_2075"; // Your WiFi Name

const char\* password = "03101992";

//"Air@37616"; // Your WiFi Password

// MQTT Broker details

const char\* mqtt\_server = "broker.hivemq.com"; // Public MQTT broker (replace with your broker if needed)

const int mqtt\_port = 1883;

const char\* mqtt\_user = ""; // Username (if required, else leave blank)

const char\* mqtt\_password = ""; // Password (if required, else leave blank)

// MQTT topics

const char\* topic\_d0 = "home/device1";

const char\* topic\_d1 = "home/device2";

const char\* topic\_d2 = "home/device3";

const char\* topic\_d3 = "home/device4";

WiFiClient espClient;

PubSubClient client(espClient);

// Function to handle incoming MQTT messages

void callback(char\* topic, byte\* payload, unsigned int length) {

String message;

for (int i = 0; i < length; i++) {

message += (char)payload[i];

}

Serial.print("Message received on topic ");

Serial.print(topic);

Serial.print(": ");

Serial.println(message);

// Control logic based on the topic

if (String(topic) == topic\_d0) {

if (message == "1") {

digitalWrite(D0, LOW);

Serial.println("Device 1 ON");

} else if (message == "0") {

digitalWrite(D0, HIGH);

Serial.println("Device 1 OFF");

}

} else if (String(topic) == topic\_d1) {

if (message == "1") {

digitalWrite(D1, LOW);

Serial.println("Device 2 ON");

} else if (message == "0") {

digitalWrite(D1, HIGH);

Serial.println("Device 2 OFF");

}

} else if (String(topic) == topic\_d2) {

if (message == "1") {

digitalWrite(D2, LOW);

Serial.println("Device 3 ON");

} else if (message == "0") {

digitalWrite(D2, HIGH);

Serial.println("Device 3 OFF");

}

} else if (String(topic) == topic\_d3) {

if (message == "1") {

digitalWrite(D3, LOW);

Serial.println("Device 4 ON");

} else if (message == "0") {

digitalWrite(D3, HIGH);

Serial.println("Device 4 OFF");

}

}

}

// Function to reconnect to the MQTT broker

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

if (client.connect("ESP8266Client", mqtt\_user, mqtt\_password)) {

Serial.println("connected");

// Subscribe to topics

client.subscribe(topic\_d0);

client.subscribe(topic\_d1);

client.subscribe(topic\_d2);

client.subscribe(topic\_d3);

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

void setup() {

Serial.begin(115200);

// Setup Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected");

// Setup MQTT client

client.setServer(mqtt\_server, mqtt\_port);

client.setCallback(callback);

// Set pin modes

pinMode(D0, OUTPUT);

pinMode(D1, OUTPUT);

pinMode(D2, OUTPUT);

pinMode(D3, OUTPUT);

// Ensure all devices are OFF initially

digitalWrite(D0, HIGH);

digitalWrite(D1, HIGH);

digitalWrite(D2, HIGH);

digitalWrite(D3, HIGH);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

}

Last query

--------------------

#define BLYNK\_TEMPLATE\_ID "TMPL3zjhMENXa" // Replace with your Template ID

#define BLYNK\_TEMPLATE\_NAME "Automation" // Replace with your Template Name

#define BLYNK\_AUTH\_TOKEN "1mEjcltvb\_83LoT0VZDrYUAyrqDrIL-f" // Replace with your Auth Token

#define BLYNK\_DEVICE\_NAME "Automation" // Replace with your Device Name

#define BLYNK\_PRINT Serial

#include <gpio.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

//#include <ESP8266WiFi.h>

#include <PubSubClient.h>

/\*// WiFi credentials

const char\* ssid = "Cmf";

//"Airtel\_anish\_2075"; // Replace with your WiFi SSID

const char\* password = "03101992";

//"Air@37616"; // Replace with your WiFi password\*/

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] ="Cmf"; // Your Wifi Name

//"Airtel\_anish\_2075\_5G","Air@37616")

char pass[] = "03101992"; // Your Wifi Password

/\*

// Variables to store the state of the relays

bool relay1State = LOW;

bool relay2State = LOW;\*/

// Relay pins

#define RELAY1\_PIN 2

#define RELAY2\_PIN 13

/\*

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

// Initialize Serial Monitor

Serial.begin(9600);

delay(10);

// Initialize relay pins

pinMode(RELAY1\_PIN, OUTPUT);

pinMode(RELAY2\_PIN, OUTPUT);

digitalWrite(RELAY1\_PIN, relay1State); // Relay 1 OFF by default

digitalWrite(RELAY2\_PIN, relay2State); // Relay 2 OFF by default

// Connect to WiFi

connectToWiFi();

// Initialize Blynk with Template ID and Auth Token

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

Serial.println("Setup complete!");

}

// Function to connect to WiFi

void connectToWiFi() {

Serial.print(ssid);

//("Connecting to WiFi");

WiFi.begin(ssid, pass);

//"Airtel\_anish\_2075\_5G","Air@37616");

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println("\nConnected to WiFi!");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

}

\*/

/\*void loop() {

Blynk.run(); // Run Blynk

}\*/

// Blynk Virtual Pin V1: Control Relay 1

BLYNK\_WRITE(V1)

{

int value = param.asInt();

Serial.println(value);

if(value == 1)

{

digitalWrite(RELAY1\_PIN, LOW);

Serial.println("LED ON"); //Setting Digital PIN as LOW to turn ON Device if relay module is "active low"

}

if(value == 0)

{

digitalWrite(RELAY1\_PIN, HIGH);

Serial.println("LED OFF");//Setting Digital PIN as HIGH to turn OFF Device if relay module is "active low"

}

}

/\*relay1State = param.asInt(); // Get the value from the Blynk app

digitalWrite(RELAY1\_PIN, relay1State ? HIGH : LOW);

Serial.println(relay1State ? "Relay 1 turned ON" : "Relay 1 turned OFF");\*/

// Blynk Virtual Pin V2: Control Relay 2

/\*BLYNK\_WRITE(V2) {

relay2State = param.asInt(); // Get the value from the Blynk app

digitalWrite(RELAY2\_PIN, relay2State ? HIGH : LOW);

Serial.println(relay2State ? "Relay 2 turned ON" : "Relay 2 turned OFF");

}\*/

void setup()

{

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

pinMode(RELAY1\_PIN,OUTPUT); //GPIO 16 (equivalent to PIN 16 of Arduino)

/\*pinMode(D1,OUTPUT); //GPIO 05 (equivalent to PIN 05 of Arduino)

pinMode(D2,OUTPUT);//GPIO 04 (equivalent to PIN 16 of Arduino)

pinMode(D3,OUTPUT);//GPIO 00 (equivalent to PIN 00 of Arduino)\*/

}

void loop()

{

Blynk.run();

}