#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// Wi-Fi credentials

const char\* ssid = "Cmf";

const char\* password = "03101992";

// MQTT broker details

const char\* mqtt\_server = "broker.hivemq.com"; // Public broker

const int mqtt\_port = 1883;

const char\* mqtt\_topic\_subscribe = "home/device/control"; // Topic to receive control commands

const char\* mqtt\_topic\_publish = "home/device/status"; // Topic to publish device status

// Define the pin for the device (e.g., LED on GPIO2 / D4 for NodeMCU)

const int devicePin = D4;

// Wi-Fi and MQTT clients

WiFiClient espClient;

PubSubClient client(espClient);

// Function to handle incoming MQTT messages

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

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

Serial.println(topic);

String message;

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

message += (char)payload[i];

}

//message.trim();

Serial.print("Message length: ");

Serial.println(message.length());

// Control relay

if (message == "ON") {

digitalWrite(devicePin, LOW); // Turn relay ON

Serial.println("Relay turned ON");

client.publish(mqtt\_topic\_publish, "Device is ON");

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

digitalWrite(devicePin, HIGH); // Turn relay OFF

Serial.println("Relay turned OFF");

client.publish(mqtt\_topic\_publish, "Device is OFF");

} else {

Serial.println("Unknown command");

client.publish(mqtt\_topic\_publish, "Unknown command received");

}

}

void setup() {

Serial.begin(115200);

// Initialize relay pin

pinMode(devicePin, OUTPUT);

digitalWrite(devicePin, HIGH); // Ensure relay is OFF initially

// Connect to Wi-Fi

Serial.print("Connecting to Wi-Fi");

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("\nWi-Fi connected!");

// Set MQTT server and callback

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

client.setCallback(callback);

// Connect to MQTT broker

while (!client.connected()) {

Serial.print("Connecting to MQTT broker...");

if (client.connect("ESP8266Client2075")) {

Serial.println("Connected to MQTT broker!");

if (client.subscribe(mqtt\_topic\_subscribe)) {

Serial.println("Subscribed to topic: home/device/control");

} else {

Serial.println("Failed to subscribe");

}

} else {

Serial.print("Failed to connect, rc=");

Serial.println(client.state());

Serial.println("Retrying in 5 seconds...");

delay(5000);

}

}

}

void loop() {

// Maintain MQTT connection

if (!client.connected()) {

while (!client.connected()) {

Serial.print("Reconnecting to MQTT broker...");

if (client.connect("ESP8266Client2075")) {

Serial.println("Reconnected!");

client.publish(mqtt\_topic\_subscribe,"Connected");

client.subscribe(mqtt\_topic\_subscribe);

} else {

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

Serial.println(client.state());

delay(5000);

}

}

}

client.loop();

}