GETWAY PUB/SUB AVEC MQTT ET FIREE BASE

#include <ESP8266WiFi.h>  
#include <LittleFS.h>  
#include <PubSubClient.h>  
#include <FirebaseESP8266.h>  
  
// Wi-Fi credentials  
const char\* ssid = "Arduino";  
const char\* password = "12345678";  
  
// MQTT Broker details  
const char\* mqtt\_server = "5.196.78.28"; // Replace with your Mosquitto server's IP  
const int mqtt\_port = 1883;  
const char\* mqtt\_publish\_temp\_topic = "/device/home/kitchen/temperature";  
const char\* mqtt\_publish\_gaz\_topic = "/device/home/kitchen/gaz";  
const char\* mqtt\_publish\_flame\_topic = "/device/home/kitchen/flame";  
const char\* mqtt\_subscribe\_topic = "home/actuators/light";  
  
// Firebase details  
#define FIREBASE\_HOST "[connexionmqtt-default-rtdb.firebaseio.com](http://connexionmqtt-default-rtdb.firebaseio.com/)" // No trailing slash  
#define FIREBASE\_AUTH "AIzaSyB\_KcLIpWiPu2Lo1Qiy3VYg4ZT2yO-FMSU"  
  
// MQTT and Firebase clients  
WiFiClient espClient;  
PubSubClient mqttClient(espClient);  
FirebaseData firebaseData;  
FirebaseConfig firebaseConfig;  
  
// Function to connect to Wi-Fi  
void setupWiFi() {  
  Serial.print("Connecting to Wi-Fi");  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL\_CONNECTED) {  
    delay(1000);  
    Serial.print(".");  
  }  
  Serial.println("\nConnected to Wi-Fi!");  
}  
  
// MQTT message callback function  
void mqttCallback(char\* topic, byte\* payload, unsigned int length) {  
  Serial.print("Message arrived on topic: ");  
  Serial.println(topic);  
  
  String message = "";  
  for (int i = 0; i < length; i++) {  
    message += (char)payload[i];  
  }  
  Serial.println("Message: " + message);  
  
  // Write received data to Firebase  
  if (String(topic) == mqtt\_subscribe\_topic) {  
    if (Firebase.setString(firebaseData, "/actuators/light", message)) {  
      Serial.println("Command synced to Firebase.");  
    } else {  
      Serial.println("Failed to sync command: " + firebaseData.errorReason());  
    }  
  }  
}  
  
// Reconnect to MQTT broker  
void reconnectMQTT() {  
  while (!mqttClient.connected()) {  
    Serial.print("Connecting to MQTT broker...");  
    if (mqttClient.connect("ESP8266Client")) {  
      Serial.println("Connected!");  
      mqttClient.subscribe(mqtt\_subscribe\_topic);  
      Serial.println("Subscribed to topic: " + String(mqtt\_subscribe\_topic));  
    } else {  
      Serial.print("Failed, rc=");  
      Serial.print(mqttClient.state());  
      Serial.println(" - retrying in 5 seconds");  
      delay(5000);  
    }  
  }  
}  
  
// Publish sensor data  
template <typename T>  
void publishData(const char\* topic, const char\* firebasePath, T value) {  
  String valueString = String(value);  
  
  // Publish to MQTT  
  mqttClient.publish(topic, valueString.c\_str());  
  Serial.println("Published to MQTT (" + String(topic) + "): " + valueString);  
  
  // Push to Firebase  
  if (Firebase.setString(firebaseData, firebasePath, valueString)) {  
    Serial.println(String(firebasePath) + " synced to Firebase.");  
  } else {  
    Serial.println("Failed to sync " + String(firebasePath) + ": " + firebaseData.errorReason());  
  }  
}  
  
void publishSensorData() {  
  float temperature = 25.5; // Simulated temperature sensor data  
  publishData(mqtt\_publish\_temp\_topic, "/kitchen/temperature", temperature);  
  
  float gasLevel = 200.5; // Simulated gas sensor data  
  publishData(mqtt\_publish\_gaz\_topic, "/kitchen/gaz", gasLevel);  
  
  int flameDetected = 1; // Simulated flame sensor data (1 = detected, 0 = not detected)  
  publishData(mqtt\_publish\_flame\_topic, "/kitchen/flame", flameDetected);  
}  
  
void setup() {  
  Serial.begin(115200);  
  
  // Initialize LittleFS  
  if (!LittleFS.begin()) {  
    Serial.println("Failed to mount file system");  
    return;  
  }  
  
  // Connect to Wi-Fi  
  setupWiFi();  
  
  // Set up MQTT  
  mqttClient.setServer(mqtt\_server, mqtt\_port);  
  mqttClient.setCallback(mqttCallback);  
  
  // Configure Firebase  
  firebaseConfig.host = FIREBASE\_HOST;  
  firebaseConfig.signer.tokens.legacy\_token = FIREBASE\_AUTH;  
  
  Firebase.begin(&firebaseConfig, NULL);  
  
  if (Firebase.ready()) {  
    Serial.println("Connected to Firebase!");  
  } else {  
    Serial.println("Failed to connect to Firebase");  
  }  
}  
  
void loop() {  
  if (!mqttClient.connected()) {  
    reconnectMQTT();  
  }  
  mqttClient.loop();  
  
  // Publish sensor data every 10 seconds  
  static unsigned long lastPublish = 0;  
  if (millis() - lastPublish > 10000) {  
    publishSensorData();  
    lastPublish = millis();  
  }  
}