#include <WiFi.h>

#include <PubSubClient.h> //請先安裝PubSubClient程式庫

#include <SimpleDHT.h>

// ------ 以下修改成你自己的WiFi帳號密碼 ------

char ssid[] = "你的WiFi SSID";

char password[] = "你的WiFi密碼";

//------ 以下修改成你腳位 ------

int pinDHT11 = 23;//DHT11

SimpleDHT11 dht11(pinDHT11);

int pinGLED = 15;//綠色LED

// ------ 以下修改成你MQTT設定 ------

char\* MQTTServer = "mqttgo.io";//免註冊MQTT伺服器

int MQTTPort = 1883;//MQTT Port

char\* MQTTUser = "";//不須帳密

char\* MQTTPassword = "";//不須帳密

//推播主題1:推播溫度(記得改Topic)

char\* MQTTPubTopic1 = "YourTopic/class205/temp";

//推播主題2:推播濕度(記得改Topic)

char\* MQTTPubTopic2 = "YourTopic/class205/humi";

//訂閱主題1:改變LED燈號(記得改Topic)

char\* MQTTSubTopic1 = "YourTopic/class205/led";

long MQTTLastPublishTime;//此變數用來記錄推播時間

long MQTTPublishInterval = 10000;//每10秒推撥一次

WiFiClient WifiClient;

PubSubClient MQTTClient(WifiClient);

void setup() {

Serial.begin(115200);

pinMode(pinGLED, OUTPUT);//綠色LED燈

//開始WiFi連線

WifiConnecte();

//開始MQTT連線

MQTTConnecte();

}

void loop() {

//如果WiFi連線中斷，則重啟WiFi連線

if (WiFi.status() != WL\_CONNECTED) WifiConnecte();

//如果MQTT連線中斷，則重啟MQTT連線

if (!MQTTClient.connected()) MQTTConnecte();

//如果距離上次傳輸已經超過10秒，則Publish溫溼度

if ((millis() - MQTTLastPublishTime) >= MQTTPublishInterval ) {

//讀取溫濕度

byte temperature = 0;

byte humidity = 0;

ReadDHT(&temperature, &humidity);

// ------ 將DHT11溫度送到MQTT主題 ------

MQTTClient.publish(MQTTPubTopic1, String(temperature).c\_str());

MQTTClient.publish(MQTTPubTopic2, String(humidity).c\_str());

Serial.println("溫溼度已推播到MQTT Broker");

MQTTLastPublishTime = millis(); //更新最後傳輸時間

}

MQTTClient.loop();//更新訂閱狀態

delay(50);

}

//讀取DHT11溫濕度

void ReadDHT(byte \*temperature, byte \*humidity) {

int err = SimpleDHTErrSuccess;

if ((err = dht11.read(temperature, humidity, NULL)) !=

SimpleDHTErrSuccess) {

Serial.print("讀取失敗,錯誤訊息=");

Serial.print(SimpleDHTErrCode(err));

Serial.print(",");

Serial.println(SimpleDHTErrDuration(err));

delay(1000);

return;

}

Serial.print("DHT讀取成功：");

Serial.print((int)\*temperature);

Serial.print(" \*C, ");

Serial.print((int)\*humidity);

Serial.println(" H");

}

//開始WiFi連線

void WifiConnecte() {

//開始WiFi連線

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("WiFi連線成功");

Serial.print("IP Address:");

Serial.println(WiFi.localIP());

}

//開始MQTT連線

void MQTTConnecte() {

MQTTClient.setServer(MQTTServer, MQTTPort);

MQTTClient.setCallback(MQTTCallback);

while (!MQTTClient.connected()) {

//以亂數為ClietID

String MQTTClientid = "esp32-" + String(random(1000000, 9999999));

if (MQTTClient.connect(MQTTClientid.c\_str(), MQTTUser, MQTTPassword)) {

//連結成功，顯示「已連線」。

Serial.println("MQTT已連線");

//訂閱SubTopic1主題

MQTTClient.subscribe(MQTTSubTopic1);

} else {

//若連線不成功，則顯示錯誤訊息，並重新連線

Serial.print("MQTT連線失敗,狀態碼=");

Serial.println(MQTTClient.state());

Serial.println("五秒後重新連線");

delay(5000);

}

}

}

//接收到訂閱時

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

Serial.print(topic); Serial.print("訂閱通知:");

String payloadString;//將接收的payload轉成字串

//顯示訂閱內容

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

payloadString = payloadString + (char)payload[i];

}

Serial.println(payloadString);

//比對主題是否為訂閱主題1

if (strcmp(topic, MQTTSubTopic1) == 0) {

Serial.println("改變燈號：" + payloadString);

if (payloadString == "1") digitalWrite(pinGLED, HIGH);

if (payloadString == "0") digitalWrite(pinGLED, LOW);

}

}