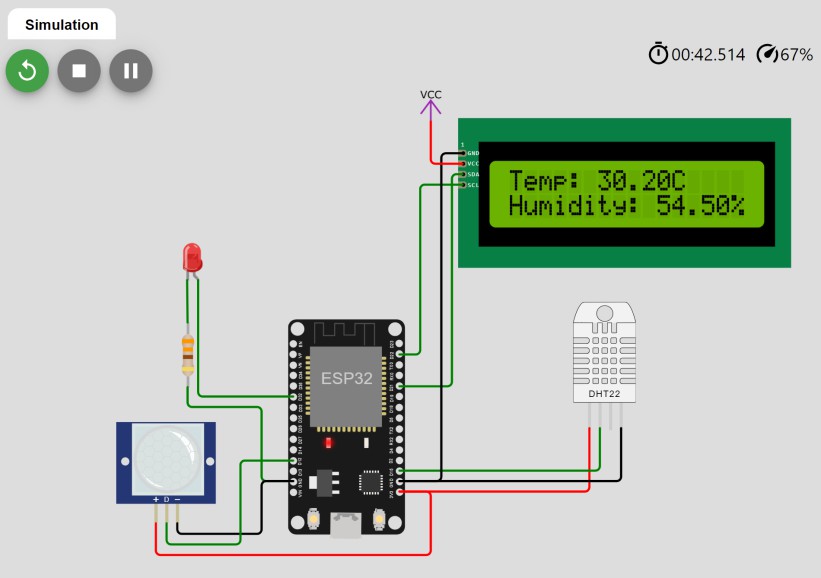
1

Bài 6. ESP32 gửi nhận dữ liệu qua MQTT

# Mục đích:

Lập trình cho ESP32 gửi nhận dữ liệu qua MQTT

Sử dụng công cụ mô phỏng wokwi <https://wokwi.com/>



# Hoạt động của chương trình:

1. Gửi dữ liệu thu thập từ sensor (nhiệt độ, độ ẩm) qua MQTT broker
2. Nhận dữ liệu qua MQTT broker, điều khiển LED on/off tương ứng Sử dụng MQTT broker có sẵn: broker.emqx.io
3. **Mã nguồn**

#include "DHTesp.h"

#define DHT\_PIN 15

#define LED\_PIN 32

#include <WiFi.h>

#include <PubSubClient.h>

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

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

int value = 0;

DHTesp dhtSensor;

void setup() {

//setup for serial communication

**Serial**.begin(9600);

  //setup for dht sensor

  dhtSensor.setup(DHT\_PIN, DHTesp::DHT22);

  pinMode(LED\_PIN, OUTPUT);

**Serial**.print("Connecting to WiFi");

  //setup for WiFi connection

  WiFi.begin("Wokwi-GUEST", "", 6);

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

  delay(100);

**Serial**.print(".");

  }

**Serial**.println("WiFi Connected!");

  client.setServer(mqtt\_server, 1883);

  client.setCallback(callback);

}

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

**Serial**.print("Message arrived [");

**Serial**.print(topic);

**Serial**.print("] ");

  String msg;

  for (int i = 0; i < length; i++) {

    msg+=((char)payload[i]);

  }

**Serial**.println();

  // Switch on the LED if an 1 was received as first character

  if (String(topic) == "iot\_collect") {

**Serial**.print("Changing output to ");

    if(msg == "on"){

**Serial**.println("on");

      digitalWrite(LED\_PIN, HIGH);

    }

    else if(msg == "off"){

**Serial**.println("off");

      digitalWrite(LED\_PIN, LOW);

    }

  }

}

void reconnect() {

  // Loop until we're reconnected

  while (!client.connected()) {

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

    // Create a random client ID

    String clientId = "ESP8266Client-";

    clientId += String(random(0xffff), HEX);

    // Attempt to connect

    if (client.connect(clientId.c\_str())) {

**Serial**.println("connected");

      // Once connected, publish an announcement...

      client.publish("iot\_collect", "record");

      // ... and resubscribe

      client.subscribe("iot\_collect");

    } else {

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

**Serial**.print(client.state());

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

      // Wait 5 seconds before retrying

      delay(5000);

    }

  }

}

void loop() {

   if (!client.connected()) {

    reconnect();

  }

  client.loop();

  TempAndHumidity data = dhtSensor.getTempAndHumidity();

  int temp = data.temperature;

  int humid = data.humidity;

  String stemp = String(temp) + "C";

  String shumid = String(humid) + "%";

  String msg = stemp+ "-" + shumid;

   unsigned long now = millis();

  if (now - lastMsg > 5000) {

    lastMsg = now;

    ++value;

**Serial**.print("Publish message: ");

**Serial**.println(msg);

    client.publish("iot\_collect", msg.c\_str());

  }

}

**Link wokwi:** https://wokwi.com/projects/383197246879171585

1. **Kết quả mô phỏng**
   * Subribe nhiệt độ và độ ẩm lên broker

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + Đọc dữ liệu từ broker để bật bóng đèn

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + Đọc dữ liệu từ broker để tắt bóng đèn

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated