**BÁO CÁO TUẦN BÀI TẬP 6 MQTT VỚI ESP32**

1. **Khởi tạo các thư viện và các giá trị tham số.**

#include <LiquidCrystal\_I2C.h>

#include "DHTesp.h"

#include <WiFi.h>

#include <HTTPClient.h>

#include <ArduinoJson.h>

#include <PubSubClient.h>

LiquidCrystal\_I2C lcd(0x27,20,4);

#define DHT\_PIN 15

#define LED\_PIN 26

#define PIR\_PIN 13

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;

1. **Hàm callback dung để điều khiển bóng đèn với các giá trị nhận được từ gói tin tương ứng.**
2. void callback(char\* topic, byte\* payload, unsigned int length) {
3. Serial.print("Message arrived [");
4. Serial.print(topic);
5. Serial.print("] ");
6. for (int i = 0; i < length; i++) {
7. Serial.print((char)payload[i]);
8. }
9. if ((char)payload[0] == '1') {
10. digitalWrite(LED\_PIN, LOW);
11. } else {
12. digitalWrite(LED\_PIN, HIGH);
13. }
14. }
15. **Hàm kết nối với MQTT.**

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/T6", "1");

      // ... and resubscribe

      client.subscribe("/IOT/T6");

    } else {

      Serial.print("failed, rc=");

      Serial.print(client.state());

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

      // Wait 5 seconds before retrying

      delay(5000);

    }

  }

}

1. **Hàm setup() chỉ chạy một lần trong quá trình.**

DHTesp dhtSensor;

void setup() {

  Serial.begin(11150);

  WiFi.begin("Wokwi-GUEST", "",6);

  // put your setup code here, to run once:

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

  pinMode(LED\_PIN, OUTPUT);

  pinMode(PIR\_PIN, INPUT);

  client.setServer(mqtt\_server, 1883);

  client.setCallback(callback);

  lcd.init();

  lcd.backlight();

}

1. **Vòng lặp để gửi dữ liệu liên tục nhiệt độ và độ ẩm về máy sever MQTT với 5s một lần.**

void loop() {

  if (!client.connected()) {

    reconnect();

  }

  client.loop();

  TempAndHumidity data = dhtSensor.getTempAndHumidity();

  int temp = data.temperature;

  int humid = data.humidity;

  lcd.setCursor(0,0);

  lcd.print("Temperature:" + String(temp) +"C");

  lcd.setCursor(0,1);

  lcd.print("Humidity:" +  String(humid) + "%");

  String mg = "Temperature: " + String(temp) + "C  Humidity: " + String(humid) + "%";

  unsigned long now = millis();

  ++value;

  // Convert 'value' to a C-style string

  char valueStr[10]; // Assuming a maximum length for the number

  snprintf(valueStr, sizeof(valueStr), "%d", value);

  // Concatenate 'valueStr' to 'mg'

  mg += " Value: ";

  mg += valueStr;

  // Convert 'mg' (String) to a C-style string (char array)

  char msg[MSG\_BUFFER\_SIZE];

  mg.toCharArray(msg, MSG\_BUFFER\_SIZE);

  Serial.print("Publish message: ");

  Serial.println(msg);

  client.publish("/IOT/T6", msg);

  delay(5000);

}

1. **Kết quả mô phỏng.**
   1. **Điều khiển bóng đèn sáng với payload = “0”.**

**Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Biểu tượng máy tính

Mô tả được tạo tự động**

* 1. **Điều khiển bóng đèn sáng với payload = “1”.**

**Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Phần mềm đa phương tiện

Mô tả được tạo tự động**

* 1. **Gửi nhiệt độ và độ ẩm liên tục cứ 5s một lần.**

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động**

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động**

1. **Link bài mô phỏng**

[BTT6 Lê Anh Vũ - Wokwi ESP32, STM32, Arduino Simulator](https://wokwi.com/projects/380981340459403265)