

ESP32 MQTT Connection

Jude's Code

```
#include <WiFi.h>
#include <PubSubClient.h>

// WiFi
const char *ssid = "iotlab"; // ssid
const char *password = "iotlab18"; //WiFi password

// MQTT Broker
const char *mqtt_broker = "broker.emqx.io";
const char *topic = "esp32/test";
const char *mqtt_username = "emqx";
const char *mqtt_password = "public";
const int mqtt_port = 1883;

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  // Set software serial baud to 115200;
  Serial.begin(115200);
  // connecting to a WiFi network
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
  //connecting to a mqtt broker
  client.setServer(mqtt_broker, mqtt_port);
  client.setCallback(callback);
  while (!client.connected()) {
    String client_id = "esp32-client-";
    client_id += String(WiFi.macAddress());
    Serial.printf("The client %s connects to the public mqtt broker\n",
client_id.c_str());
    if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
      Serial.println("Public emqx mqtt broker connected");
    } else {
      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(2000);
    }
  }
}
// publish and subscribe
```

```
client.publish(topic, "Hi EMQX I'm ESP32 ^^");
client.publish(topic, "Hi My name is Bambo Jude ^^");
client.publish(topic, "Hey Suleman!!!!");

client.subscribe(topic);
}

void callback(char *topic, byte *payload, unsigned int length) {
  Serial.println("Message arrived in topic: ");
  Serial.println(topic);
  Serial.println("Message:");
  for (int i = 0; i < length; i++) {
    Serial.print((char) payload[i]);
  }
  Serial.println();
  Serial.println("-----");
}

void loop() {
  client.loop();
}
```

From:
<https://student-wiki.eolab.de/> - **HSRW EOLab Students Wiki**

Permanent link:
https://student-wiki.eolab.de/doku.php?id=amc2022:esp32_mqtt:start&rev=1653391633

Last update: **2023/01/05 14:38**

