

ESP32 MQTT

Use the ESP32's wifi module to communicate with a computer

 Difficulté Facile

 Durée 15 minute(s)

 Catégories Électronique, Robotique

 Coût 0 USD (\$)

Sommaire

Introduction

Étape 1 - Prepare an MQTT Broker

Étape 2 - Install ESP32 Development Board & other dependencies

Étape 3 - Full Code

Étape 4 - Running and Testing

Notes et références

Commentaires

Introduction

Introduction & Goals:

This project utilizes the Wifi module on an ESP32 along with the Publish/Subscribe (PubSub) library on the Arduino IDE to communicate with other Internet of Things (IoT) devices. In this specific tutorial, we learn how to use the ESP32 to communicate with your personal device in a bidirectional way.

GOALS:

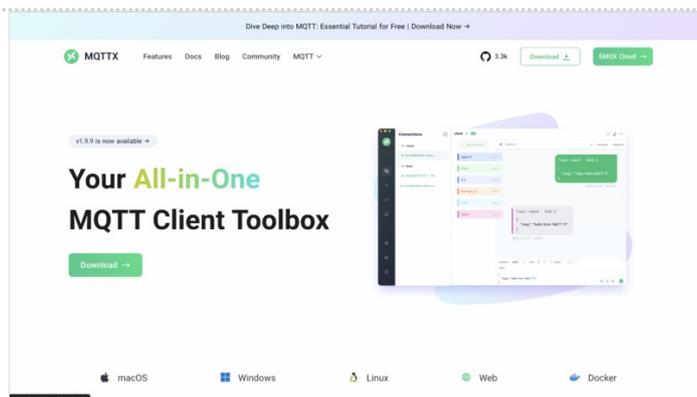
1. Learn what Message Queuing Telemetry Transport (MQTT) is
2. Set up esp32 dev library and pubSub along with MQTT to exchange messages

Matériaux

Étape 1 - Prepare an MQTT Broker

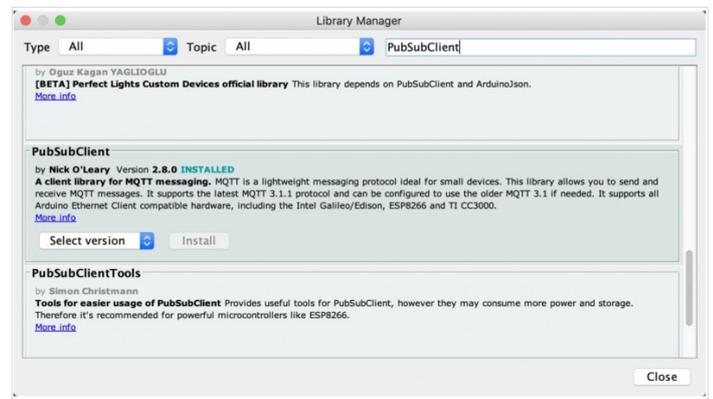
Before we begin, we need to ensure you have an MQTT broker to communicate and test with. We can download MQTXX (the front-end broker) to ensure communication. MQTT is a lightweight messaging protocol designed for efficient communication between devices in low-bandwidth, high-latency networks. It uses a publish-subscribe model, which we will go over later.

Outils



Étape 2 - Install ESP32 Development Board & other dependancies

Ensure that the ESP32 dev library is installed, then go to libraries in the Arduino IDE and download PubSubClient. PubSub, short for Publish/Subscribe, is a messaging pattern where senders (publishers) distribute messages to receivers (subscribers) without directly communicating with them. Publishers send messages to a central hub (broker), which then distributes them to all interested subscribers based on their topic subscriptions. This is useful to conserve data when trying to share information, because rather than multiple different connections between the publisher and each subscriber, the publisher only needs to send the information once.



Étape 3 - Full Code

Copy the full code into your IDE. Feel free to change things like the text it prints, etc

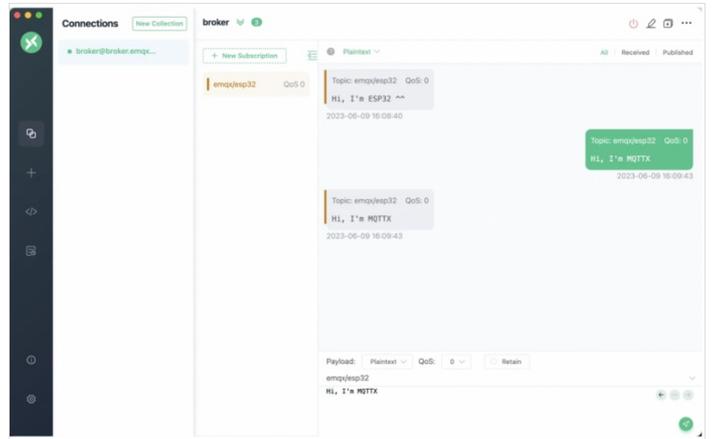
```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 // WiFi
5 const char *ssid = "xxxxx"; // Enter your Wi-Fi name
6 const char *password = "xxxxx"; // Enter Wi-Fi password
7
8 // MQTT Broker
9 const char *mqtt_broker = "broker.emqx.io";
10 const char *topic = "emqx/esp32";
11 const char *mqtt_username = "emqx";
12 const char *mqtt_password = "public";
13 const int mqtt_port = 1883;
14
15 WiFiClient espClient;
16 PubSubClient client(espClient);
17
18 void setup() {
19   // Set software serial baud to 115200;
20   Serial.begin(115200);
21   // Connecting to a WiFi network
22   WiFi.begin(ssid, password);
23   while (WiFi.status() != WL_CONNECTED) {
24     delay(500);
25     Serial.println("Connecting to WiFi..");
26   }
27   Serial.println("Connected to the Wi-Fi network");
28   //connecting to a mqtt broker
29   client.setServer(mqtt_broker, mqtt_port);
30   client.setCallback(callback);
31   while (!client.connected()) {
32     String client_id = "esp32-client-";
33     client_id += String(WiFi.macAddress());
34     Serial.printf("The client %s connects to the public MQTT broker\n", client_id.c_str());
35     if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
36       Serial.println("Public EMQX MQTT broker connected");
37     } else {
38       Serial.print("failed with state ");
39       Serial.print(client.state());
40       delay(2000);
41     }
42   }
43   // Publish and subscribe
44   client.publish(topic, "Hi, I'm ESP32 ^^");
45   client.subscribe(topic);
46 }
47
48 void callback(char *topic, byte *payload, unsigned int length) {
49   Serial.print("Message arrived in topic: ");
50   Serial.println(topic);
51   Serial.print("Message:");
52   for (int i = 0; i < length; i++) {
53     Serial.print((char) payload[i]);
54   }
55   Serial.println();
56   Serial.println("-----");
57 }
58
59 void loop() {
60   client.loop();
61 }
```

Étape 4 - Running and Testing

1. Please follow these steps to upload the complete code using Arduino and power on the ESP32:
 1. Connect the ESP32 to your computer using a USB cable.
 2. Open the Arduino IDE and select the appropriate board and port from the "Tools" menu.
 3. Copy and paste the complete code into the Arduino IDE.
 4. Click the "Upload" button (or use the shortcut Ctrl+U) to compile and upload the code to the ESP32.
 5. Wait for the upload process to finish, ensuring there are no errors.
 6. Once the code is uploaded, disconnect the ESP32 from the computer.
 7. Power on the ESP32 by connecting it to a suitable power source.
2. Open the serial monitor and set the baud rate to 115200. Then, check the connection status of the ESP32 by monitoring the output in the serial monitor.
3. Use the MQTTX client to establish a connection with the MQTT broker and publish messages such as to the ESP32.

Hi, I'm MQTTX

broker and publish messages such as to the ESP32.



Notes et références

Real World Use Cases:

1. <https://www.pubnub.com/blog/what-is-mqtt-use-cases/>
2. <https://www.influxdata.com/blog/mqtt-use-cases/>