IoT Communication - MQTT

Welcome to Day 3 of our IoT Workshop Series! Building upon the foundation laid on Day 2, where you delved into the intricacies of sensors and their protocols, today is all about application. Get ready to harness that knowledge and dive into the world of data transmission as we explore how to gather and send sensor data using MQTT.

1. MQTT in detail

2. ESP8266 and WiFi

Next, we'll take a closer look at connecting the ESP8266 to WiFi. This step is crucial for enabling wireless communication, allowing your devices to seamlessly interact with the digital world.

wifi-http-api-example.cpp

```cpp
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>

// Replace with your WiFi credentials
const char* ssid = "wifi-ssid";
const char* password = "wifi-password";

// Replace with your API details
const char* apiHost = "weather.eolab.de";
const char* apiEndpoint = "/api";

// Create an instance of WiFiClientSecure
WiFiClientSecure client;

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);

  Serial.print("Connecting to ");
  Serial.print(ssid);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
```

```
  Serial.println("\nConnected to WiFi");

  // Set the client to verify the server's certificate
  client.setInsecure();

  // Give the client a chance to perform the handshake
  delay(1000);
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    // Make an HTTPS GET request
    client.connect(apiHost, 443); // Use port 443 for HTTPS
    client.print(String("GET ") + apiEndpoint + " HTTP/1.1\r\n" +
                 "Host: " + apiHost + "\r\n" +
                 "Connection: close\r\n\r\n");

    // Wait for the response
    while (client.connected()) {
      String line = client.readStringUntil('\n');
      if (line == "\r") {
        Serial.println("Headers received");
        break;
      }
    }

    // Print the response from the server
    while (client.available()) {
      String response = client.readStringUntil('\n');
      Serial.println(response);
    }

    client.stop();
  }

  // Wait for a while before making the next request
  delay(5000);
}
```

In this example code, you will first of all connect to a WiFi network and then try to access a predefined API. This API is the root of our public weather station API. You can find out more about the project right here: HSRW Weather Station at Campus Kamp-Lintfort

3. Our first MQTT Publish

Stepping ahead, you'll have the chance to implement your first MQTT publish from the ESP8266.

While the following example provides a static demonstration, we encourage you to take it a step further by incorporating one of the sensors available to you.

4. Subscribe

Now, let's move on to an essential aspect of our workshop. We'll guide you through the process of subscribing to an MQTT topic. This step is a practical gateway to interactive IoT applications. You'll learn how to exchange MQTT messages between devices, potentially triggering actions like LED reactions or serial outputs. While we'll provide initial guidance, feel free to explore further and experiment with this dynamic feature.

Recording

From:
https://wiki.eolab.de/ - **HSRW EOLab Wiki**

Permanent link:
https://wiki.eolab.de/doku.php?id=latinet:unicaes:workshops:communication-23&rev=1693336314

Last update: **2023/08/29 21:11**