M5Stack mit WiFi und Brokerverbindung

Wir brauchen die Anordnung vom letzten Projekt (M5Stack-Modul, ENV-Sensor und Motion-Sensor). Erstellt einen neuen Projektordner, wählt diesen für VSC aus und erstellt mit PlatformIO ein neues Projekt „M5Stack-Broker“.

Wählt wieder das M5Stack-Core-ESP32-Board aus und wählt die Arduino-Einstellung. Deselektiert den Haken bei dem Default-Ordner und wählt euer angelegtes Verzeichnis für das Projekt (den Ordner, der in VSC geöffnet wurde).

Zusätzlich sind die Libraries

- M5Stack
- ArduinoJson
- bsec2
- BME68x Sensor library
- M5Unit-ENV
- M5Unified
- M5UnitUnified
- M5Utility
- M5HAL
- PubSubClient

```
#include <Arduino.h>
#include <M5Stack.h>
#include <WiFi.h>
#include <Wire.h>
#include <ArduinoJson.h>
#include "M5UnitENV.h"
#include <PubSubClient.h>

...

WiFiClient wifiClient;
PubSubClient client(wifiClient);
...
```

notwendig. Bindet diese wie im vorherigen Projekt in das platformio.ini ein.

Die PubSubClient-Library stellt alle Informationen zur Verbindung mit einem Broker bereit. Wir brauchen zwei Objekte: den WifiClient, der auf der Wifi-Verbindung lauscht (Wifi trotzdem genauso initialisieren wie bei der früheren Übung) und den eigentlichen Client, der die Verbindung als Parameter übernimmt.

Dann benötigen wir zwei zusätzliche Funktionen, die wir wieder zunächst deklarieren und dann am Ende der Datei implementieren.

```
void callback(char *topic, byte *payload, unsigned int length);
void reConnect();
...

void setup() { ...
... // NACH dem connectWifi-Aufruf Client initialisieren

client.setServer(mqtt_url, 1883); // Sets the server parameter.
client.setCallback(callback);  // callback-Funktion
```

```
// in der loop()

  if (!client.connected())
  {
    reConnect();
  }
  client.loop();  // Aufruf auch zu einem späteren Zeitpunkt
...

client.publish("M5Stack", "hello world");
```

in Setup konfigurieren wir den Client mit der URL des Brokers und dem gewünschten Port. Zusätzlich müssen wir die Callback-Funktion setzen, um den Aufruf-Event verarbeiten zu können.

In der Hauptschleife prüfen wir regelmäßig, ob der Client noch verbunden ist und wenn nicht, wir die Verbindung neu initialisiert. Zudem seht ihr ein Beispiel einer einfachen Veröffentlichung (Topic, Paylod).

Die Callback-Funktion muss vor allem die erhaltene Information verarbeiten. Wie man an der Parameterliste gut sehen kann, kommt zuerst der Topic und dann die Payload zusammen mit der Angabe der Länge zur Kontrolle bzw. Verarbeitung.

```
void reConnect() {
  while (!client.connected()) {
    ...
    // Create a random client ID.
    String clientId = "M5Stack-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect.
    if (client.connect(clientId.c_str())) {
      // ... success and resubscribe.
      client.subscribe("M5Stack");
    } else {  // failed, print client.state())
      delay(5000);
    }
  }
}
```

In reconnect wird in einer Scheife versucht, die Verbindung wieder aufzubauen. Dafür brauchen wir eine eindeutige ID für unseren Client.

Dann versuchen wir, eine Verbindung aufzubauen. Schlägt der Versuch fehl, wird in 5sek ein neuer Versuch gestartet.

War der Versuch erfolgreich, subscriben wir uns probehalber zu einem Topic.

Aufgabe 1:

Stellt eine Verbindung zum hivemq-Broker her. Schickt eure Nachricht unter dem Topic „M5Stack“ und und subscribed euch zu diesem Topic.

Sorgt dafür, dass beim Verbinden und beim Empfangen Ausgaben auf dem Screen auftauchen, die über den Stand der Verbindung und der empfangenen Daten informieren.

Habt ihr alles richtig zusammengesetzt, solltet ihr die gesendeten Daten von euch und den anderen empfangen.

Nun müssen wir unsere Daten in eine JSON-Struktur verpacken und versenden.

Aufgabe 2:

Erstellt eine JSON-Struktur und verpackt die gemessenen Daten in diese Struktur. Die Daten des ENV-Sensors sollen alle 5 Sekunden erhoben werden und einmal pro Minute an den Broker verschickt werden. Die JSON-Struktur soll alle Informationen pro Sekunde sowie ein Mittelwert pro Messwert enthalten.

Schickt die Daten an den Broker.

```
// subscriptions and publishings
```