**ESP to VVVV: DB auslesen, sortieren, umschreiben und in einer neuen Tabelle für VVV bereitstellen**

Was wird gebraucht?

INSTALLATIONEN

- Texteditor wie zum Beispiel Atom: https://atom.io/

- Installation von node für Windows/macOS/Linux: https://nodejs.org/de/download/

- Installation von SQLite Browser für Windows/macOS/Linux: https://sqlitebrowser.org/

- Installation von einem Nodemon package

- Installation von MQTT; über das Terminal: **npm install mqtt --save**

DATA FILES

- Eine Datei, die eine Verbindung zum Server aufbaut und Topics abonniert (siehe FromMQTTToDAtabase01.js)

- Eine Datenbank mit einer Tabelle: Drei Spalten mit dem Datantyp „text“ und eine Spalte mit „REAL“

- Eine CodeSnippet, mit dem eine Verbindung zur Datenbank möglich ist (siehe connect.js)

- Einbindung des SQLite Moduls in den JS-Dateien: **const sqlite3 = require('sqlite3').verbose();**

- Eine CodeSnippet, der eine neue Tabelle in die DB schreibt (z.Bsp. TemplateWriteIntoDatabse.js)

Wie wird das ausgeführt?

1. *Windows:* Über cmd im Suchfeld „Node.js command prompt“ öffnen und mit „cd ../source“ Verzeichnis öffnen, wo Dateien abgelegt sind

*Mac:* Über das Suchfeld den Terminal öffnen du mit „cd ../source“ Verzeichnis öffnen, wo Dateien abgelegt sind

1. „FromMQTTToDAtabase01.js“ öffnen und IP-Adresse überprüfen
2. DB ausführen: Befehl „node Datenbankname.db“ eingeben
3. JS Datei ausführen/Server Connection/Messages von MQTT erhalten: nodemon --delay (Sekundenzahl für das Delay eintragen) Dataeiname.js (in diesem Fall newRead.js)

* Nodemon stellt Änderungen in newRead.js fest und führt, falls etwas geändert wurde, die Datei neu aus
  1. In der Praxis muss per Hand der Befehl „rs“ für Restart immer wieder eingegeben werden
  2. Mit Strg + C oder cmd + C wird der Vorgang abgebrochen

ALTERNATIVE

4. Mit „node FromMQTTToDAtabase01.js“ öffnen

4.1 In einem neuen Terminal parallel mit „node newRead.js“ newRead.ausführen

4.2 Bei jedem neuen Besucher, der an den Stand kommt, 4.1 erneut ausführen

Wie funktioniert der Code?

In diesem Teil der Prototypenumsetzung muss die Datenbank sortiert und eine neue Tabelle befüllt werden, die dann für das VVVV Programm bereitsteht.

Durch die Datei „FromMQTTToDAtabase01.js“ wird eine Verbindung zum Server hergestellt, sodass über MQTT Topics abonniert werden können. Die ankommenden „messages“ werden dann in die Datenbank gespeichert.

Mit der Dateien „newRead.js“ wird die Datenbank sortiert und ausgelesen. Um eine Person einer Zeit und einem Ort zuzuordnen

Quellen: [https://stackoverflow.com/questions/1972242/how-to-auto-reload-files-in-node-js#](https://stackoverflow.com/questions/1972242/how-to-auto-reload-files-in-node-js)

<https://stackoverflow.com/questions/37979489/how-to-watch-and-reload-ts-node-when-typescript-file-changes>

<https://docs.npmjs.com/files/package.json>

<https://github.com/remy/nodemon>

<https://www.shellbefehle.de/befehle/>

<https://de.wikipedia.org/wiki/NaN>

<https://www.mediaevent.de/javascript/math-zahlen-numbers.html>

<https://github.com/remy/nodemon>