MQTT

(Message Queuing Telemetry Transport)

Klaus Knopper (klaus.knopper@hs-kl.de)

Sommersemester 2019

Definition

- **offenes Nachrichtenprotokoll** für Machine-to-Machine-Kommunikation (**M2M**), das die **Übertragung von Telemetriedaten** in Form von **Nachrichten** zwischen Geräten ermöglicht, **trotz hoher Verzögerungen** oder **beschränkter Netzwerke**.
- Seit 2013 wird **MQTT** über die Organization for the Advancement of Structured Information Standards (**OASIS**) als **Protokoll** des **Internet der Dinge** standardisiert.

https://de.wikipedia.org/wiki/MQTT v. 3.5.2019

Eigenschaften von MQTT

- Techn: Offiziell benutzte TCP-Ports **1883** und 8883, optional mit TLS-Transportverschlüsselung.
- Ein MQTT-Server (sog. „**Broker**“) hält die gesamte Datenlage seiner Kommunikationspartner, die nur sporadisch (z.B. bei Änderungen) mit ihm kommunizieren, daher auch für kleine „unperformante“ Clients geeignet, die selbst keine Statusinformationen speichern müssen oder können.
- MQTT-Protokoll realisiert das Beobachter Entwurfsmuster (auch bekannt als „Publish-Subscribe“) und ermöglicht so Andocken oder Abmelden weiterer Clients nach Bedarf.

Nachrichten-Struktur und Sicherheit

- Meist „Klartext“-Nachrichten,
- Eine Nachricht besteht aus einem hierarchisch aufgebauten „Thema“ (**Topic**), z.B. **`Küche/Kühlschrank/Temperatur`** oder **`Auto/Rad/3/Luftdruck`** und dem Nachrichten**inhalt** (z.B. **ON**, **OFF**, oder **21**°C).
- Optional kann eine Autorisierung per Login/Passwort oder mit SSL Public/Private Keys zum Schutz vor unautorisiertem Zugriff (Senden oder Empfangen) von Nachrichten vom MQTT-Broker gefordert werden, bevor der Nachrichteninhalt transferiert wird.

MQTT-Broker (Server)
(noch mal zusammengefasst)

- Speichert (temporär oder persistent) den sporadisch von den Clients gemeldeten Systemzustand, der auf diese Weise jederzeit aktuell abgerufen werden kann, ohne dass die Clients (Sensoren) direkt kontaktiert werden müssten.
- Leitet Nachrichten über Änderungen auch direkt an die „Abonnenten“ eines Channels weiter.
- Realisiert eine „Datenbank des IST-Zustands“ von IoT-Geräten.

mosquitto

Open Source Implementation eines einfachen MQTT Brokers und Client Commandline-Tools

Installation (Debian):

Broker („Server“): **`sudo apt install mosquitto`**
Client: **`sudo apt install mosquitto-clients`**
Starten: **`sudo /etc/init.d/mosquitto start`**

Test: Channel (Topic) aufmachen mit:

```
mosquitto_sub -d -h localhost -t test
```

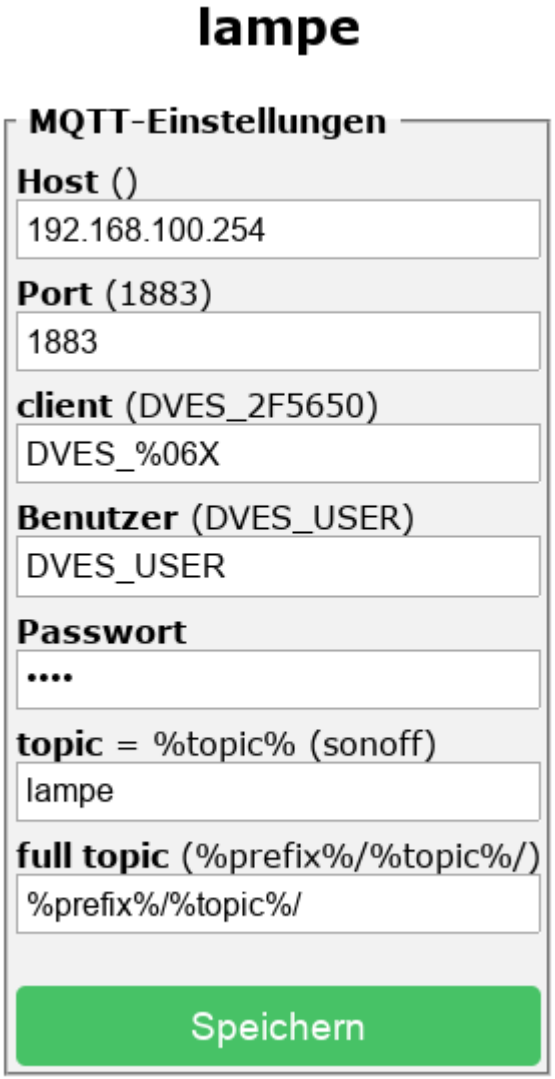
Nachricht schicken:

```
mosquitto_pub -d -h localhost -t test -m "Hello World"
```

mosquitto
Live-Test

- Test mit Sonoff-Tasmota-Steckdose

Konfiguration →



- Alle Channels abonnieren (# = Multi-Wildcard):
  `mosquitto_sub -d -h localhost -t \#`
  (Beobachten, was passiert, wenn die Steckdose ein/ausgeschaltet wird)
- Steckdose per MQTT-Nachricht schalten:
  `mosquitto_pub -d -h localhost \`
  `-t cmnd/lampe/POWER \`
  `-m "on"` (oder **`"off"`**)

Fragen?

-t /Cmnd/Küche/Kaffeemaschine -m „make Coffee“

Klaus.Knopper@hs-kl.de