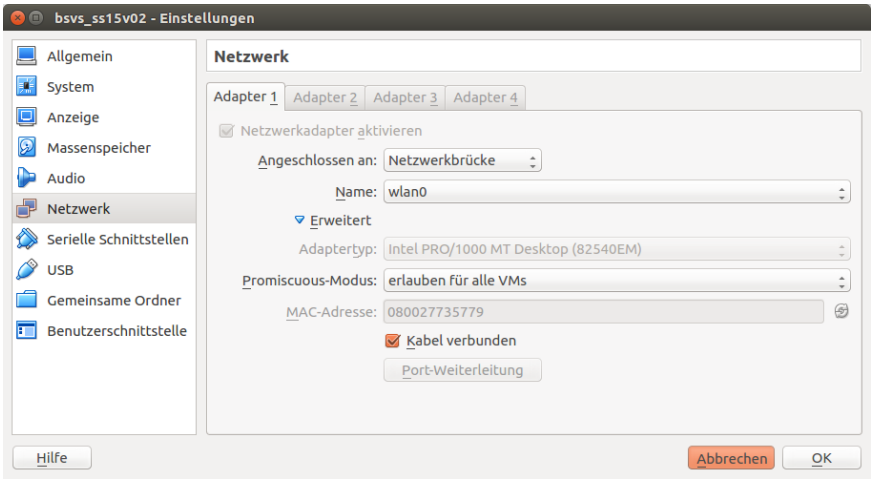
# Tutorial: MQTT

## Beschreibung der Funktionsweise

Message Queue Telemetry Transport (MQTT) ist ein offenes Protokoll, dass die Implementierung einer Message Oriented Middleware darstellt. MQTT wurde speziell für Embedded Systems im Bereich Internet of Things (IoT) entworfen und zeichnet sich durch eine hohe Fehlertoleranz aus. Es funktioniert nach dem publish / subscribe Prinzip und bietet somit eine effiziente Machine-to-Machine-Kommunikation. Hierbei verbinden sich die Subscriber zu einem MQTT-Broker und "hören" auf Topics. Publisher versenden Nachrichten mit einem Topic versehen. Passen diese zusammen erhält der Subcriber die Nachricht.

## Installation der Softwarekomponenten

Für die Installation der Softwarekomponenten in den Virtuellen Maschinen ist es erforderlich, die Netzwerkkonfiguration wie abgebildet zu ändern (getestet unter Ubuntu 16.04 als Hostsystem - Tests unter OS X folgen).




Die Installation der Softwarekomponenten erfolgt über das mitgeliefert Shellscript. Das Skript muss aus root ausgeführt werden. Die in der Shell angezeigten Fehler können ignoriert werden, sofern der Selbsttest am Ende folgendes Ergebnis hat:

```
Starting MQTT service
[ ok ] Starting network daemon:: mosquitto.
Subscribung to test/topic for one message
(in background to prevent blocking the shell)
Publishing on test/topic
test/topic helloWorld
Stopping MQTT service
[ ok ] Stopping network daemon:: mosquitto.
```

### MQTT-Test zwischen VMs

Um einen Test der MQTT-Kommunikation zwischen den VMs via Shellbefehlen durchzuführen muss die IP das Brokers beim Aufruf mit angegeben werden. Z.B:

```
mosquitto_sub -v -C 1 -t 'test/topic' -h '192.168.2.118'
```

bzw.

```
mosquitto_pub -t 'test/topic' -m 'helloWorld' -h '192.168.2.118'
```

## Codebasis in C++

Ein Minimalbeispiel basierend auf der "Asynchronous MQTT client library for C" ist im Anhang zu diesem Dokument zu finden. Die "Bibliothek" besteht lediglich aus Headern und C Dateien, welche ein Binding in die bereits installierte MQTT Umgebung darstellen. Diese stellt die einfachste Möglichkeit dar in MQTT einzusteigen. Wer tiefer in die Materie einsteigen möchte, findet in den Quellen weiter Hinweise für einen Embedded Ansatz.

Um das Beispiel auszuführen werden drei virtuelle Maschinen benötigt. Die erste um den Broker laufen zu lassen.

```
service mosquitto start
```

In der zweiten VM wird der Publisher gestartet und in der dritten der Client. Diese befinden sich als NetBeans Projekt im Anhang zu diesem Dokument und können direkt importiert, kompiliert und ausgeführt werden. Um den Code als Subscriber zu kompilieren muss das Define "Publisher" auf 0 gesetzt werden.

## Codebasis in Java

Für die Entwicklung in Java gibt es von dem Eclipse Paho Projekt ein Tutorial zur Installation und ein Codebeispiel (siehe Referenzen). Diese wurde im Rahmen der Vorbereitung nicht evaluiert. Bei Fragen und Problemen hierzu nutzen Sie bitte das Forum zur Veranstaltung.

## Referenzen

Installationsanweisungen

- Verwendete Beschreibung zu Installation von Mosquitto in der Debian VM http:// mosquitto.org/2013/01/mosquitto-debian-repository/

Codebeispiele

- Quelle zu dem verwendeten Codebeispiel inkl. Bibliothek zum Download: http:// www.eclipse.org/paho/files /mqttdoc/Casync/index.html
- C++ Beispiel für diejenigen, die etwas tiefer in die Materie einsteigen möchten: http://modelbasedtesting.co.uk/?p=181
- Beispiel Java: https://eclipse.org/paho/clients/java/

Android App

- Einfache App zum direkten Zugriff auf MQTT via Android Device: https://play. google.com/ store/apps/details?id=at.tripwire.mqtt.client&hl=de