

Walther- Rathenau- Gewerbeschule Freiburg	<b>mqtt mit Webfrontend</b> mqtt-Meldungen auf einer Webseite anzeigen oder abschicken	Fach: ITV	Gruppe:
		Dat.:10. Mai 2019	Seite <b>1</b>
		Name:	
		Klasse: FTI3t	
		Punkte: /20	Note:

# 1 Apache für die Ausführung von CGI-Skripten konfigurieren

Möchte man mqtt-Meldungen auf einer Web-Seite anzeigen, lässt sich das mit einem CGI-Skript lösen.

## 1.1 Modul laden

Apache2 bietet dafür das Modul **cgi**. Um zu prüfen ob das Modul geladen ist, kann man sich den Inhalt von `/etc/apache2/mods-enabled` auflisten lassen:

```
ls /etc/apache2/mods-enabled/
...
lrwxrwxrwx 1 root root 27 May 10 10:04 cgid.conf -> ../mods-available/cgid.conf
lrwxrwxrwx 1 root root 27 May 10 10:04 cgid.load -> ../mods-available/cgid.load
...
```

Falls die beiden gezeigten Zeilen nicht zu sehen sind, das CGI-Modul aktivieren:

```
su ...
a2enmod cgi
```

## 1.2 Virtual Host konfigurieren

Das CGI-Programm kann in einer beliebigen Programmiersprache erstellt werden, es muss nur auf dem System auf dem Apache läuft, ausgeführt werden können.

Um nicht noch eine neue Programmiersprache lernen zu müssen, bleiben wir bei **C**.

Der Virtual-Host **mosquitto.wara.de** wird nun entsprechend konfiguriert.

Im Verzeichnis `/etc/apache2/sites-available` wird die Datei **mosquitto.wara.de.conf** mit folgendem Inhalt abgelegt:

```
<VirtualHost *:80>

    ServerName mosquitto.wara.de

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/mosquitto

    <Directory /var/www/mosquitto>
        Options +ExecCGI
        AddHandler cgi-script .bin
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Ähnlich wie oben das cgi-Modul, muss dieser Virtual-Host mit **a2ensite** aktiviert werden:

```
a2ensite mosquitto.wara.de
```

Im Verzeichnis **/etc/apache2/sites-enabled** muss anschliessend der Soft-Link auf die Datei **/etc/apache2/sites-available/mosquitto.wara.de.conf** zu sehen sein.

Das Document-Root dieses Virtual-Hosts ist **/var/www/mosquitto**. In diesem Verzeichnis wird später die Datei mit der Webseite und das CGI-Programm abgelegt.

## 2 mqtt-Clients selbst erstellen

Da wir die mqtt-Clients mit **C** selbst schreiben müssen, benötigen wir eine spezielle *Programm-bibliothek*.

### 2.1 Programmbibliothek erstellen

Damit wir sicher die neueste Version erhalten, wird die Programmbibliothek selbst compiliert (alles mit root-Rechten ausführen). Der angegebene Link gilt 10. Mai 2019 und muss evtl. angepasst werden.

```
cd /tmp
wget https://mosquitto.org/files/source/mosquitto-1.6.2.tar.gz
tar -xzf mosquitto-1.6.2.tar.gz
cd mosquitto-1.6.2
make
make install
```

Nach dem Installieren müssen die folgenden beiden Dateien bzw. Verknüpfungen existieren:

```
/usr/local/lib/libmosquitto.so
/usr/local/include/mosquitto.h
```

## 2.2 mqtt-Test-Subscriber erstellen

```
//compilieren mit
// gcc -L/usr/local/lib single.c -lmosquitto -o single.bin
#include <stdlib.h>
#include <stdio.h>
#include "mosquitto.h"

int main(int argc, char *argv[])
{

    printf("Content-Type: text/html\n\n");
    printf("<html>\n<body>\n");

    int rc;
    struct mosquitto_message *msg;

    mosquitto_lib_init();

    rc = mosquitto_subscribe_simple(
    &msg, 1, true,
    "test/#", 0,
    "localhost", 1883,
    NULL, 60, true,
    NULL, NULL,
    NULL, NULL);

    if(rc){
        printf("Error: %s\n", mosquitto_strerror(rc));
        mosquitto_lib_cleanup();
        return rc;
    }

    //printf("%s %s\n", msg->topic, (char *)msg->payload);
    printf("<h1>Testprogramm: libmosquitto + apache2</h1>");
    printf("<p>das cgi-programm holt sich eine meldung vom ");
    printf("mqtt-broker. das thema (topic) ist test/#</p>'");
    printf ("<p> damit es keinen http-timeout gibt, muss der ");
    printf("broker eine retained-message bereithalten.</p>");
    printf("<p>Topic: %s</p> <p>Message: %s</p>", msg->topic, (char *)msg->payload);
    mosquitto_message_free(&msg);

    mosquitto_lib_cleanup();

    printf("</body></html>");

    return 0;
}
```

Der Quelltext muss wie folgt übersetzt werden:

```
gcc -L/usr/local/lib single.c -lmosquitto -o single.bin
```

Die Option **-l** teilt dem Compiler (eigentlich dem Linker) den Namen der Bibliothek mit, während man mit der Option **-L** angibt, wo diese Bibliothek abgelegt ist.

Die nach dem Compilervorgang entstandene Datei **single.bin** muss dann ins Document-Root-Verzeichnis unseres VirtualHosts kopiert werden (vgl. Abschnitt 1.2).

## 3 Testen unseres CGI-Programms

### 3.1 Direktes Aufrufen des CGI-Programms

Ist alles richtig konfiguriert, kann man das CGI-Programm testen. Dazu sollte man vorher dem mqtt-Broker eine Nachricht mit gesetztem Retain-Flag und dem richtigen Topic (hier: `test/#`) schicken.

Das mqtt-CGI-Programm lässt sich wie folgt direkt von einem Web-Browser aus aufrufen:

```
http://mosquitto.wara.de/single.bin
```

Wenn alles richtig funktioniert, sieht man die im Broker mit Retain-Flag abgelegte Meldung.

### 3.2 Testen mit Webseite

Mit dem folgenden html-Quelltext kann man das CGI-Programm aus einer Seite (hier: `index.html`) heraus aufrufen:

```
<html>
das ist eine statische seite
<h1> Nun kommt ein Web-Formular </h1>
<p>
    Der Click auf den Knopf liefert die naechste
    mosquitto-veroeffentlichung mit dem topic test/# zurueck.
</p>
<form action="http://mosquitto.wara.de/single.bin">
<input type="submit" value="Hau wech!"/>
</form>
</html>
```