MQTT on the Raspberry Pi

How to setup the Mosquitto Broker V2.1

1. Setup MQTT on a Pi

This docu details the **setup of MQTT** on a Raspberry Pi.

Before you attempt this make sure that the Raspberry Pi can connect you your local LAN (cable or wireless) and that you have SSH enabled on the Pi. You should also be able to login to your Pi from a terminal window on your Windows PC or Mac. Furthermore you need to be able to run a File Transfer program on your home computer - to transfer Python files to your Pi.
I always use the free FileZilla FTP to do this.
FileZilla on the Net

Show installed packages on the pi:

```
dpkg -l
```

Step 1

Check the size of the partition on the SD card. Should be using the whole card.

```
df -h
```

Step 2

Make sure you have the new list of packages

```
sudo apt-get update
```

Step 3

If you have a Pi with the **full Pi OS** installed, go to step4. If, however, your Pi runs an **OS Lite** you need to install a few packages first.

```
sudo apt install python3-pip
sudo apt install git
pip3 install Rpi.GPIO
pip3 install paho-mqtt
```

Step 4

Change to the Home-Dir of user pi, then clone the code from the GitHub Repository with this command:

```
git clone https://github.com/dakota127/python_stuff.git
```

After that you will find the code in the folder python_stuff.

Step 5

Mosquitto might already be on your pi. Go through these steps anyway.
Install the mosquitto MQTT broker on the pi, does not harm if already installed

```
sudo apt install -y mosquitto mosquitto-clients
```

or try this

```
sudo apt-get install mosquitto mosquitto-clients
```

(takes about 5 minutes)

Step 6

After installation the broker is configured (default config file) to allow only clients with known user-id's in the LAN to connect. - and only from the local machine. If you want to change any of this behaviour, you **need to have a user config file** and also acl file. See below.

Mosquittos default config file should **never be modified.**

How to add a user config file:

Create a user config file for mosquitto, description here
https://mosquitto.org/man/mosquitto-conf-5.html

Fortunately you already have one, check supplied folder **mosquitto_config -** there you will find four files

- my_mosquitto.conf — private config file for mosquitto
- my_passw.txt — password file for mosquitto
- my_aclfile.txt — ACL permissions on a per user basis
- setup_mqtt.ch — shellscript to copy these file to the right place on the pi, see below

See **end of document** for a schematic showing relationship between these files.

Note

Never change the **original** mosquitto config file which can be found here:

```
/etc/mosquitto/mosquitto.conf
```

This file looks like this (mosquitto v2.0.11)

```
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
```

```
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d
```

Mosquitto always looks into the folder /etc/mosquitto/conf.d/ for a additional user config file. Any file found there (no matter what the file name) is considered to be a user config file.
So we place our user config file into this folder.

Here is how to go about this:
FTP the folder **mosquitto_config** to the home dir of user pi, change to this folder and execute the shellscript using :

```
sudo ./setup_mqtt.sh
```

This shellscript copies three files to their correct place: **my_mosquitto.conf**, **my_passw.txt**, **my_acl.txt**

Now the three files are in the correct place on the pi. In addition the scripts calls a utility (supplied with mosquito) to **encrypt** the password file.

You can look at the encrypted password file using (do NOT change anything)

```
sudo nano /etc/mosquitto/my_passw.txt
```

Note

If you want to run mosquitto with **no** authorization simply change true to false in this line in my_mosquitto.conf

```
allow_anonymous false
```

The excute the shellscript again to replace the fils in /etc.....

Note 2

The following line makes sure that clients from remote machines can access the broker:

```
listener 1883 0.0.0.0
```

Step 7

Enable services for mosquitto (autostart after boot pi)

```
sudo systemctl enable mosquitto.service
```

Step 8

Check mosquitto with:

```
sudo mosquitto -v  -c /etc/mosquitto/mosquitto.conf
```

Step 9

Use these commands to Start/Stop/Restart mosquitto

```
sudo service mosquitto start
sudo service mosquitto stop
sudo service mosquitto restart
```

Check if the port is active

```
netstat -tln | grep 1883
```

Maybe check if the process is actually running

```
ps -ef | grep mosq
```

Query status with this

```
sudo service mosquitto status
```

Check the log file in case of problems

```
tail /var/log/mosquitto/mosquitto.log
```

Step 10

Get the IP-Adr of your Pi (needed if one client is on another machine)

```
hostname -I
```

Step 11

Test mosquitto with pub and sub **utilities** (provided with mosquitto install), **use your o**
Open **two** terminal windows, login to the pi and use these commands.

```
python3 demo_mqtt_sub.py            ( subscribe program)

python3 demo_mqtt_pub.py            ( publish program)
```

See available commandline options with `python3 demo_mqtt_pub.py -h`

Links:

https://iotbytes.wordpress.com/mosquitto-mqtt-broker-on-raspberry-pi/

http://www.steves-internet-guide.com/mosquitto-logging/

https://learn.adafruit.com/diy-esp8266-home-security-with-lua-and-mqtt/configuring-mqtt-on-the-raspberry-pi

here is another video explaining the setup

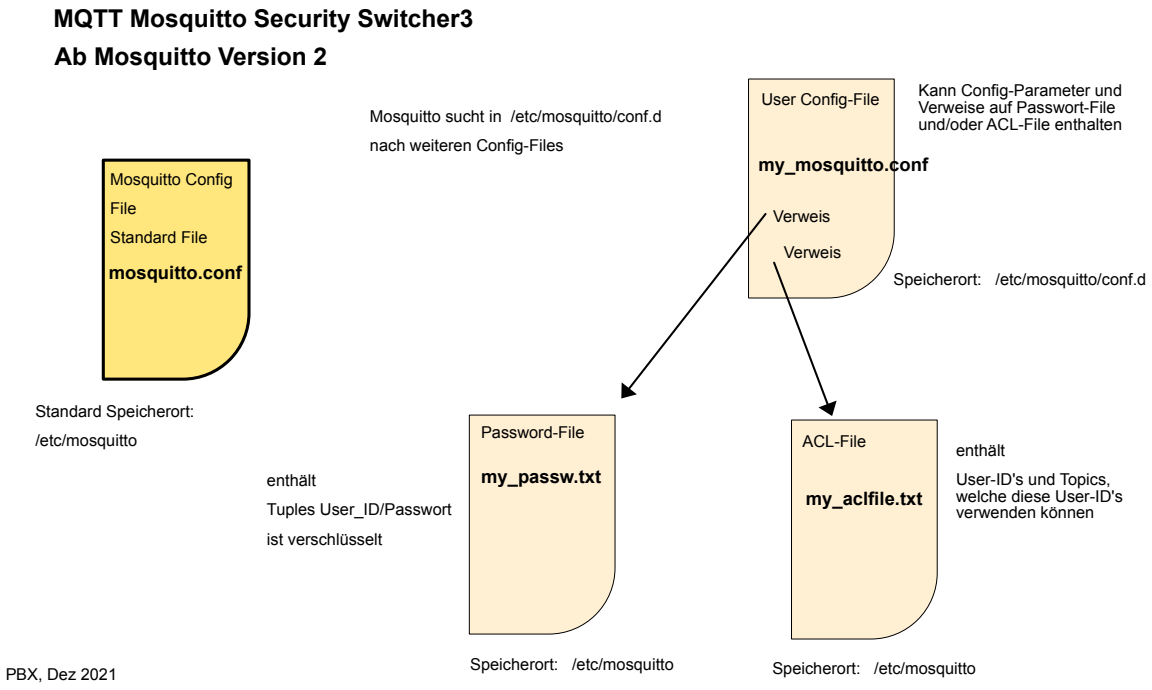
Video on YouTube

2. Running the demo programs

Read the docu mqtt_demo.pdf in this folder.

Also check out this video for a setup with MQTT and SQL Lite on a **Pi Zero**. It is amazing what this litte machine can do.
Andreas Spiess Pi Zero

3. Mosquitto Config Files



October 2021, Peter K. Boxler