

MQTT is a messaging protocol used for machine to machine communications in Internet of Things (IoT) Applications. It uses a Pub/sub architecture, with a central broker.

Within the space we have a MQTT broker running on Acidburn, where events like door openings, sensors, and the doorbell are currently broadcast on the hacman network.

MQTT Topics

Topics within MQTT are a series of levels, separated by forward slashes /. To subscribe to multiple topics at once, wildcards can be used. + matches a single level, and # matches anything below the current level.

Topic hierarchies can be set up however the user wants, we use it in the following way:

- [door/inner/#](#) - Things involving the inner door
- [door/outer/#](#) - Things involving the outer door
 - door/outer/state gets an event every time the physical door is opened or closed (payload 'opened' or 'closed')
 - door/outer/opened/username gets an event every time an RFID card is used to open the door (payload is the display name set by the user opening the door)
 - door/outer/doorbell gets an event every time the doorbell is pushed
 - door/outer/opened/key gets an event every time the door is opened with a mechanical override key
- [door/shutter/#](#) - Things involving the shutter (Obsolete, for old space)
- [sensor/shutter/#](#) - Raw and averaged sensor readings for the shutter. Gets processed into door/shutter/opened, door/shutter/closed, and door/shutter/status messages. (Obsolete, for old space)
- sensor/temp/+/reading - Temperature readings in degrees Celsius.
- wifi/clients - Number of clients connected to the unifi wireless access point.
- lights/beacon - Controls the beacon (publish to this the number of milliseconds you would like the beacon to be on for)

Subscribing to messages on linux / raspberry pi

To install the mosquitto client

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

To view the inner door being opened & closed, and by who

```
mosquitto_sub -h acidburn -t 'door/inner/#' -v
```

Note that # is used as a wildcard, and will match anything after the point it is posted. To view everything on the network:

```
mosquitto_sub -h acidburn -t '#' -v
```

Subscribing to messages within python

First you'll need to install the python mqtt libraries:

```
sudo apt-get install python-mosquitto
```

Then the following code can be used to read messages on the MQTT network:

```
#!/usr/bin/python
import mosquitto

def on_message(mosq, obj, msg):
    if msg.topic == 'door/outer/opened/username':
        print("%s opened the outer door." % msg.payload)
    elif msg.topic == 'door/outer/buzzer':
        print('Buzzer')
    elif msg.topic == 'door/outer/invalidcard':
        print("Unknown card at outer door.")

mqttc = mosquitto.Mosquitto("mqtt2bot")    # The name of your client,
probably should be unique
mqttc.connect("acidburn")                  # the broker to connect to
mqttc.subscribe("door/outer/#")            # you can repeat this line
multiple times to subscribe to additional topics
mqttc.on_message = on_message              # function to run when message is
received

while mqttc.loop() == 0:
    pass
```

Category: [Hackspace projects](#)

From:

<http://testwiki.hecatron.com/> - **Hacman DEMO ONLY**

Permanent link:

<http://testwiki.hecatron.com/doku.php?id=old:hackspace:mqtt>

Last update: **2022/11/30 16:31**

