

Was als Demonstrator begann, entwickelt sich so langsam zum *Running Gag*: ich bin mit meiner Konferenzpflanze live auf Tour! Was es damit nun konkret auf sich hat? Lest einfach weiter...



Mit der [OSMC 2018](#) endet das Jahr sozusagen für mich – zumindest bezüglich großer Konferenzen und Vorträge. Die diesjährige OSMC hat es geschafft, die Erwartungen noch zu übertreffen – doch neben großartigem Essen, abwechslungsreichem Vortragsprogramm und perfekter Organisation ist es in erster Linie das Wiedersehen mit all den anderen Verrückten, das die OSMC für mich zu einem Highlight macht: sich mal nicht wie ein Freak fühlen, mittendrin statt nur dabei, Austausch und Ideen...

Okay, und jede Menge Dummschwatz wie dem wohlschmeckenden Cocktail „Sex On The Bällebad“, dem COC-Shop – ein eintägiger Workshop zum Thema „Code Of Conduct“, den der liebe [@n0braist](#) für uns vorbereiten wird, da ein einstündiger Talk mit dem Titel „Respect The COC“ der Sache nicht gerecht würde – schon während der Gin-Verkostung (und **DAS** war mal ein Gin!) beklagten die ersten akute Gesichtsmuskelschmerzen vom vielen Lachen ☐

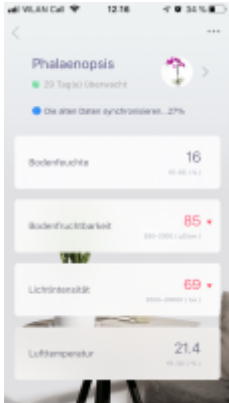
g

Widmen wir uns dem heimlichen Star meines diesjährigen OSMC-Talks „Smart Home Monitoring mit openHAB 2“, nämlich [Pflänzchen](#): Pflänzchen ist eine kleine Orchidee, die Freunde mir vor etwa vier Jahren schenkten. Seither vegetiert sie vor sich hin, denn einen grünen Daumen besitze ich nicht – der klare Vorteil ist, dass Pflanzen nicht schreien können... Wäre nicht ohnehin ein Wintergarten im Haus vorhanden, dann hätte ich auch keinen – da ich aber einen habe sehe ich mich unter Druck auch ein paar Pflanzen besitzen zu müssen, und es deprimiert mich, jeden Spätsommer mehrere davon zu entsorgen. Geht ja auch ganz schön ins Geld, ne? Was also wäre für jemanden wie mich naheliegender als einen „digitalen grünen Daumen“ zu etablieren – und smarte Unterstützung für die Pflanzenpflege zu holen?

Also besitzt *Pflänzchen* nun einen Pflanzensensor, namentlich einen [Xiaomi Mi Plant Flower Care](#); ich hab meinen von Amazon, hab mir von einem Twitter-Follower aber sagen lassen, dass das über [Direktbestellungen in China](#) deutlich günstiger geht. Der Sensor wird im Pflanztopf deponiert und greift Werte für Nährstoffgehalt, Lufttemperatur, Lichtintensität, Batteriestand und Bodenfeuchte ab. Ob er auf Dauer korrodiert und dann in seiner Funktionsfähigkeit eingeschränkt ist weiß ich noch nicht – aber ich werde berichten.

Einschub: Da sich die Konferenzpflanze großer Beliebtheit erfreut, ist sie nun online! Über [konferenzpflanze.unix.de](#) könnt ihr *Pflänzchen* nun life stalken und den Pflanzenschutzbund informieren, wenn ich mich nicht in angemessener Weise kümmerge.

Wenn jemand Lust hat, darf er gerne [lebtdiekonferenzpflanzeno.ch](#) sponsoren! ☐



Xiaomi stellt mit [Flower Care](#) eine App für die Kommunikation mit dem Sensor bereit; die ist so ein bisschen durchwachsen, aber auch ganz niedlich, und sie gibt die Möglichkeit die eigene Pflanze zu bestimmen - notfalls auch anhand von Blattform und -farbe. Und sie gibt Rahmenwerte vor: mein *Pflänzchen* fühlt sich beispielsweise angeblich bei einer Bodenfeuchte zwischen 15% und 65% am wohlsten. Ich kann nicht recht beurteilen, wie korrekt die Angaben hier sind - sollte man wohl zur Sicherheit nochmal gegenchecken - aber grundsätzlich mögen wohl alle Pflanzen Wasser, und das stellte bislang schon die erste Hürde dar.

Zum Zwecke der Integration ins Smart Home wird die App nicht benötigt - hier genügt ein RPi3 in Reichweite, denn der Pflanzensensor kommuniziert über LE Bluetooth. Bei der Gelegenheit hab ich mal wieder feststellen dürfen, dass ich Bluetooth irgendwie nicht so richtig leiden kann: der ältere Mac Mini beispielsweise *sieht* den Pflanzensensor einfach nicht, der Laptop auch nicht, der RPi3 ohne Probleme - Protokollwirrwarr at it's best. Also erstmal ein paar Pakete auf dem Raspberry installieren und dann mit `bluetoothctl` schauen: der Pflanzensensor wird als *Flower Care* aufgeführt, und seine MAC wird später für die Konfiguration benötigt.

```
$ apt-get install git python3 python3-pip bluetooth bluez
$ bluetoothctl
[NEW] Controller B8:27:EB:28:44:49 $HOSTNAME [default]
[NEW] Device A0:2C:36:83:3C:7C LM1180
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:28:44:49 Discovering: yes
[NEW] Device 67:C2:4C:86:4B:F4 67-C2-4C-86-4B-F4
[NEW] Device 98:D6:BB:18:F1:44 98-D6-BB-18-F1-44
[NEW] Device C4:7C:8D:67:4C:9E Flower care
[NEW] Device A0:2C:36:F3:05:0F LM4196
```

Thomas Dietrich sah in dem Sensor ein gewisses Potential und nahm sich der Sache an: mit dem [miflora-mqtt-daemon](#) hat er ein hübsches kleines Stück Software erschaffen, dass die ermittelten Werte über Bluetooth abgreift und beispielsweise an einen MQTT-Broker übergeben kann. Die Dokumentation des Projekts ist super - wie eigentlich bei allen von Thomas' Projekten, meinen aufrichtigen Dank an dieser Stelle - folgt zur Einrichtung also einfach den Anweisungen dort. In meinem Fall lauscht ein lokaler *mosquitto* auf dem RPi3, und deshalb sieht meine Version der `config.ini` folgendermaßen aus:

```
[General]
reporting_method = mqtt-json
adapter = hci0
```

```
[Daemon]
enabled = true
period = 300

[MQTT]
hostname = localhost
port = 1883
keepalive = 60
base_topic = miflora
username = openhabian
password = BuuNpd9Y5pen
tls = false
```

```
[Sensors]
Phaleanopsis      = C4:7C:8D:67:4C:9E
#JapaneseBonsai   = C4:7C:8D:44:55:66
#Petunia@Balcony = C4:7C:8D:77:88:99
```

Mit valider Konfiguration kann der Dienst nun in Betrieb genommen werden; im Moment läuft das auf meinem System in einem screen weil ich noch nicht dazu kam, das mal sauber zu machen - ihr wisst ja wie das ist mit dem Schuster und seinen Schuhen... Es tut jedenfalls und übermittelt gemäß Konfig alle fünf Minuten die aktuellen Werte, die dann im lokalen MQTT-Broker ankommen. Soweit so gut.

```
$ python3 /opt/miflora-mqtt-daemon/miflora-mqtt-daemon.py
Xiaomi Mi Flora Plant Sensor MQTT Client/Daemon
Source: https://github.com/ThomDietrich/miflora-mqtt-daemon
```

```
[2018-11-11 13:13:26] Connecting to MQTT broker ...
[2018-11-11 13:13:26] MQTT connection established
```

```
Adding sensor to device list and testing connection ...
```

```
Name:          "Phaleanopsis"
Internal name: "Phaleanopsis"
Device name:   "Flower care"
MAC address:   C4:7C:8D:67:4C:9E
Firmware:     3.1.9
```

```
[2018-11-11 13:13:29] Initial connection to Mi Flora sensor "Phaleanopsis"
(C4:7C:8D:67:4C:9E) successful
```

```
[2018-11-11 13:13:29] Announcing Mi Flora devices to MQTT broker for auto-
discovery ...
```

```
[2018-11-11 13:13:30] Retrieving data from sensor "Phaleanopsis" ...
```

```
[2018-11-11 13:13:30] Result: {"battery": 99, "temperature": 21.8,
"conductivity": 86, "moisture": 16, "light": 37}
```

```
[2018-11-11 13:13:30] Publishing to MQTT topic "miflora/Phaleanopsis"
```

```
[2018-11-11 13:13:31] Sleeping (300 seconds) ...
```

Selbstverständlich wollte ich den Sensor auch in mein openHAB integrieren - nur so ist es möglich das System wütende „**GIEß MICH!**“-Emails, Selbstmorddrohungen per Telegram und Hilferufe per Twitter versenden zu lassen. Das ist gar nicht so schwierig - in erster Linie muss ich dafür sorgen, dass sich die openHAB-Instanz mit dem mosquitto verbindet. Ich verwende dazu noch die rustikale Variante binding-mqtt1 und hinterlege eine /etc/openhab2/services/mqtt.cfg mit folgendem Inhalt:

```
local.url=tcp://127.0.0.1:1883
local.clientId=openhab
local.user=openhabian
local.pwd=BuuNpd9Y5pen
local.qos=1
```

Mit „Publishing to MQTT topic „miflora/Phaleanopsis“ hat der Daemon mir ja schon mitgeteilt, wo die Daten landen; wenn ich nun also in openHAB das Addon JSONPath Transformation installiere kann ich die Werte auf diesem Wege genau dort einsammeln und in openHAB einbinden - ich habe die Items in /etc/openhab2/items/FlowerCare.items abgelegt, aber das sei euch überlassen.

```
Number Wohnzimmer_Orchidee_LightIntensity
  "Pflanze Phaleanopsis Sunlight Intensity [%d lux]"
  (gInfluxDB)
  {mqtt="<[local:miflora/Phaleanopsis:state:JSONPATH($.light)]"}
```

```
Number Wohnzimmer_Orchidee_AirTemperature
  "Pflanze Phaleanopsis Air Temperature [%.1f °C]"
  (gInfluxDB)
  {mqtt="<[local:miflora/Phaleanopsis:state:JSONPATH($.temperature)]"}
```

```
Number Wohnzimmer_Orchidee_SoilMoisture
  "Pflanze Phaleanopsis Soil Moisture [%d %%]"
  (gInfluxDB)
  {mqtt="<[local:miflora/Phaleanopsis:state:JSONPATH($.moisture)]"}
```

```
Number Wohnzimmer_Orchidee_SoilConductivity
  "Pflanze Phaleanopsis Soil Conductivity/Fertility [%d µS/cm]"
  (gInfluxDB)
  {mqtt="<[local:miflora/Phaleanopsis:state:JSONPATH($.conductivity)]"}
```

```
Number Wohnzimmer_Orchidee_Battery
  "Pflanze Phaleanopsis Sensor Battery Level [%d %%]"
  (gInfluxDB)
  {mqtt="<[local:miflora/Phaleanopsis:state:JSONPATH($.battery)]"}
```

Wir ihr vielleicht gesehen habt: in meinem System existiert eine Gruppe gInfluxDB, und jedes der Pflanzen-Items ist Mitglied dieser Gruppe. Was hat es damit auf sich? Nun - im System ist persistence-influxdb aktiv, so dass übermittelte Werte dauerhaft in einer InfluxDB gespeichert werden können; und in meinem Fall sorgt eine /etc/openhab2/persistence/influxdb.persist dafür, dass alle Items, die der Gruppe

gInfluxDB angehören, minütlich gespeichert werden.

```
Strategies {
  everyMinute : "0 * * * * ?"
  default = everyChange
}

Items {
  gInfluxDB* : strategy = everyMinute
}
```



Dauerhafte Speicherung in InfluxDB erlaubt grafische Aufbereitung in Grafana - in Bezug auf *Pflänzchen* ist das ein bisschen irre aber eben auch cool... *Pflänzchen* sollte nun also nicht nur leidlich überleben, sondern es sollte ihm mit technischem Feuer-aus-allen-Rohren nun auch so richtig gut gehen. Es wird mich weiter zu Live-Demos begleiten - Mitte Dezember steht die nächste an -, es wird hoffentlich nicht wieder kopfüber aus dem Auto fallen (aber hey, der Pflanzensensor hat auch das überlebt!), und idealerweise sollte es irgendwann blühen und sehr gesund aussehen. Challenge accepted! ☐ Es ist mein letzter Versuch in Sachen Grünzeug, und wenn auch das nichts bringt gibt es eben in Zukunft Plastikpflanzen von IKEA und dann komm mir keiner mit Luftwerten und Wohlfühlcharakter!

A handwritten signature in black ink that reads 'Maia'.

:wq!

So, dann legt mal los - und erzählt doch, ob und wie es für euch funktioniert.