openHAB 3.x auf dem Raspberry Pi 2 B+ V1.1 oder Pi 4 (4GB RAM)

Die Basis - openHABian

Quellen:

- OpenHAB 3.x Installation - Option 1: Flash the openHABian image to a SD card (Recommended)
- OpenHab 3.x
- Add-ons via Konfigurationsdatei installieren
- Etcher

```
wget
https://github.com/openhab/openhabian/releases/download/v1.7.1/openhabian-
pi-raspios32-202112021757-gitbb960d8-crc8b6dfb1b.img.xz
# write *.img.xz to sd-card using etcher
./balenaEtcher-1.7.3-x64.AppImage
```

Basisinstallation

Basisinstallation wie hier beschrieben durchführen.

```
sudo su
apt-get install raspi-config python3 python3-pip python3-venv
pip install virtualenv
cd ~
virtualenv -p python3 venv
source venv/bin/activate
```

Basis Konfiguration

Grafana, MQTT Broker „Mosquitto“ und InfluxDB installieren:

```
sudo
openhabian-config
# Installation des MQTT Brokers Mosquitto:
#--> 20 Optional Components
#  --> 23 Mosquitto    (User:PW xxxxxxxx:xxxxxxxx)
#
# Installation von InflixDB & Grafana:
#--> 20 Optional Components
#  --> 24 InfluxDB & Grafana
# InfluxDB admin:xxxxxxxxxxx
# InfluxDB openhab:
```

```
# Openhab User:PW    openhab:xxxxxxxxxx
# Openhab Admin:    admin:xxxxxxxxxxxx
# Grafana User:PW    grafana:xxxxxxxxxxxx
# Influx Database Name:    openhab
# Grafana Admin Account PW: xxxxxxxxxxx
#

echo "listener 1883" >> /etc/mosquitto/mosquitto.conf
systemctl restart mosquitto

echo "binding = mqtt" >> /etc/openhab/services/addons.cfg
echo "transformation = jsonpath" >> /etc/openhab/services/addons.cfg
echo "persistence = influxdb" >> /etc/openhab/services/addons.cfg
```

Problembehebung

Bei Problemen mit der Konfiguration via Web-GUI:

```
# openhab-cli stop
systemctl stop openhab
openhab-cli clean-cache
# openhab-cli start
systemctl start openhab
```

OpenHAB Console

```
sudo su
openhab-cli console
# username/password -> openhab:habopen
feature:list | grep ^openhab
```

Erreichbarkeit

Die OpenHAB Hauptseite ist unter folgender URL erreichbar: http://<Rasb-Pi-PI>:8080/
Die Grafana Hauptseite unter folgender URL: http://<Rasberry-Pi-PI>:8080/

IKEA VINDRIKTNING

Seit dem Sommer 2021 bietet IKEA unter dem Produktnamen VINDRIKTNING einen Luftqualitätssensor an welcher mit einer kleine LED Ampel die Güte der Umgebungsluft grob quantisiert darstellen kann. Dazu misst er die Partikdichte (PM2,5) in µg/m³.
Bedeutung der Ampelfarben:
Grün: 0-35 - Gut/Niedrig
Gelb: 36-85 - OK/Mittel
Rot: 86-... - Nicht gut/Hoch
Pulsierend: Startmodus

Der Sensor kann über ein handelsübliches Smartphone Ladegerät mit USB-C Stecker betrieben werden. Er besitzt von Haus aus keinerlei Funkschnittstellen (weder Bluetooth, WLAN, RFID/NFC oder LoRaWAN) und ist somit kein smarter Sensor. Er kann also nicht ohne weiteres in eine bestehende Hausautomation oder automatische Messwerterfassung eingebunden werden.
Damit das doch gelingt und darüber hinaus die einzelnen Messwerte grafisch darstellbar werden gibt es bereits verschiedene Bastelprojekte im grenzenlosen Netz.
Inspiriert durch einen Artikel auf Heise.de in Kombination mit einem WEMOS D1 Mini ESP8266 läßt sich der IKEA VINDRIKTNING jedoch in eine Bestehende Hausautomation einbinden. Hier z.B: mittels des MQTT Broker Mosquitto und OpenHab 3.x.

Quellen:

- IKEA Hack
- VINDRIKTNING Bedienungsanleitung
- Heise Artikel
- ESP8266 Pinout

/etc/openhab/things/mqtt.things

`mcedit /etc/openhab/things/mqtt.things`

```
Bridge mqtt:broker:mosquitto "Mosquitto" @ "System" [ host="127.0.0.1",
secure="AUTO", username="openhabian", password="openhabian",
clientID="openHAB3" ]
{
  Thing topic IKEA-DD2B88 "IKEA Luftqualitätssensor DD2B88" @ "IKEA_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
stateTopic="esp8266-vindriktning-particle-sensor/VINDRIKTNING-DD2B88/state"
, transformationPattern="JSONPATH:$.pm25" ]
        }

  Thing topic IKEA-DD272B "IKEA Luftqualitätssensor DD272B" @ "IKEA_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
stateTopic="esp8266-vindriktning-particle-sensor/VINDRIKTNING-DD272B/state"
, transformationPattern="JSONPATH:$.pm25" ]
        }

  Thing topic IKEA-6D6D60 "IKEA Luftqualitätssensor 6D6D60" @ "IKEA_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
stateTopic="esp8266-vindriktning-particle-sensor/VINDRIKTNING-6D6D60/state"
, transformationPattern="JSONPATH:$.pm25" ]
        }

  Thing topic IKEA-1EE6E3 "IKEA Luftqualitätssensor 1EE6E3" @ "IKEA_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
```

```
stateTopic="esp8266-vindriktning-particle-sensor/VINDRIKTNING-1EE6E3/state"
, transformationPattern="JSONPATH:$.pm25" ]
        }

  Thing topic IKEA-1ED6BE "IKEA Luftqualitätssensor 1ED6BE" @ "IKEA_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
stateTopic="esp8266-vindriktning-particle-sensor/VINDRIKTNING-1ED6BE/state"
, transformationPattern="JSONPATH:$.pm25" ]
        }

  Thing topic PH-AC4236-6BB2 "Philips Luftreiniger AC4236/10 6BB2" @
"PHILIPS_Test" {
    Channels:
    Type number : particulate_matter "Feinstaubkonzentration" [
stateTopic="PH-AC4236-6BB2/state" , transformationPattern="JSONPATH:$.pm25"
]
        }
}
```

/etc/openhab/items/mqtt.items

mcedit /etc/openhab/items/mqtt.items

```
//Number : <Einheit> <Variablenname_in_Datenbank> "IKEA Sensor xxxxxxxx
[%.0f µg/m³]" <particulate_matter> [ "IKEA Luftsensor" ] {
channel="mqtt:topic:mosquitto:Ikea_AirQuality:<Sensor-ID>" }
Number IKEA_DD2B88 "IKEA Sensor DD2B88 [%.0f µg/m³]" <particulate_matter> [
"IKEA Sensor DD2B88" ] { channel="mqtt:topic:mosquitto:IKEA-
DD2B88:particulate_matter" }
Number IKEA_DD272B "IKEA Sensor DD272B [%.0f µg/m³]" <particulate_matter> [
"IKEA Sensor DD272B" ] { channel="mqtt:topic:mosquitto:IKEA-
DD272B:particulate_matter" }
Number IKEA_6D6D60 "IKEA Sensor 6D6D60 [%.0f µg/m³]" <particulate_matter> [
"IKEA Sensor 6D6D60" ] {
channel="mqtt:topic:mosquitto:IKEA-6D6D60:particulate_matter" }
Number IKEA_1EE6E3 "IKEA Sensor 1EE6E3 [%.0f µg/m³]" <particulate_matter> [
"IKEA Sensor 1EE6E3" ] {
channel="mqtt:topic:mosquitto:IKEA-1EE6E3:particulate_matter" }
Number IKEA_1ED6BE "IKEA Sensor 1ED6BE [%.0f µg/m³]" <particulate_matter> [
"IKEA Sensor 1ED6BE" ] {
channel="mqtt:topic:mosquitto:IKEA-1ED6BE:particulate_matter" }
//
Number PH_AC4236_6BB2 "Philips AC4236/10 [%.0f µg/m³]" <particulate_matter>
[ "Philips AC4236/10" ] { channel="mqtt:topic:mosquitto:PH-
AC4236-6BB2:particulate_matter" }
// particulate matter = Feinstaub, Einheit: µg/m³, particulate_matter
```

/etc/openhab/persistence/influxdb.persist

`mcedit /etc/openhab/persistence/influxdb.persist`

```
// persistence strategies have a name and a definition and are referred to
in the "Items" section
Strategies {
    // for rrd, we need a cron strategy
    everyMinute : "0 * * * * ?"
//    everyHour : "0 0 * * * ?"
//    everyDay : "0 0 0 * * ?"
    default = everyChange
}

Items {
              * : strategy = everyChange, everyDay, restoreOnStartup
    IKEA_* : strategy = everyMinute, restoreOnStartup
    PH_AC4236* : strategy = everyMinute, restoreOnStartup
    //IKEA_DD272B : strategy = everyMinute, restoreOnStartup
}
```

=== /etc/openhab/sitemaps/default.sitemap `mcedit /etc/openhab/sitemaps/default.sitemap`

```
sitemap default label="Christophs Hausautomation" {
    Frame label="IKEA Sensor DD2B88" {
    Text label="Feinstaub" item=IKEA_DD2B88 icon="line"
    Chart item=IKEA_DD2B88 period=D refresh=10000 legend=true
    Chart item=IKEA_DD2B88 period=2D refresh=10000 legend=true
    Chart item=IKEA_DD2B88 period=W refresh=10000 legend=true
    }
    Frame label="IKEA Sensor DD272B" {
    Text label="Feinstaub" item=IKEA_DD272B icon="line"
    Chart item=IKEA_DD272B period=D refresh=10000 legend=true
    Chart item=IKEA_DD272B period=2D refresh=10000 legend=true
    Chart item=IKEA_DD272B period=W refresh=10000 legend=true
    }
    Frame label="IKEA Sensor 6D6D60" {
    Text label="Feinstaub" item=IKEA_6D6D60 icon="line"
    Chart item=IKEA_6D6D60 period=D refresh=10000 legend=true
    Chart item=IKEA_6D6D60 period=2D refresh=10000 legend=true
    Chart item=IKEA_6D6D60 period=W refresh=10000 legend=true
    }
    Frame label="IKEA Sensor 1EE6E3" {
    Text label="Feinstaub" item=IKEA_1EE6E3 icon="line"
    Chart item=IKEA_1EE6E3 period=D refresh=10000 legend=true
    Chart item=IKEA_1EE6E3 period=2D refresh=10000 legend=true
    Chart item=IKEA_1EE6E3 period=W refresh=10000 legend=true
    }
    Frame label="IKEA Sensor 1ED6BE" {
```

```
    Text label="Feinstaub" item=IKEA_1ED6BE icon="line"
    Chart item=IKEA_1ED6BE period=D refresh=10000 legend=true
    Chart item=IKEA_1ED6BE period=2D refresh=10000 legend=true
    Chart item=IKEA_1ED6BE period=W refresh=10000 legend=true
    }
    Frame label="Luftreiniger Philips AC4236/10 6BB2" {
    Text label="Feinstaub" item=PH_AC4236_6BB2 icon="line"
    Chart item=PH_AC4236_6BB2 period=D refresh=10000 legend=true
    Chart item=PH_AC4236_6BB2 period=2D refresh=10000 legend=true
    Chart item=PH_AC4236_6BB2 period=W refresh=10000 legend=true
    }
}
```

Philips Air Purifier AC4236/10

Seit Ende Dezember 2021 gehärt nun auch ein Philips AC4236/10 Luftreiniger zur technischen Ausstattung unserer Wohnung. Da der Luftreiniger via Smartphone über die `Philips Clean Home+` App kontrollierbat ist lag der Wunsch nahe die Messwerte des integrierten Feinstaubsensors (PM2.5) in die eigene Messwerterfassung einzubinden und z.B. via OpenHAB, InfluxDB und Grafana zu darzustellen.

Quelle(n):

1. https://github.com/rgerganov/py-air-control ← funktioniert nicht, liefert keine Werte zurück
2. https://giters.com/Oschangkai/philips-airpurifier-coap
3. https://mosquitto.org/man/mosquitto_pub-1.html
4. https://michlstechblog.info/blog/mqtt-publish-topics-from-command-line/

Tool [2] installieren

```
pip3 install py-air-control
pip3 install -U git+https://github.com/rgerganov/CoAPthon3
git clone https://github.com/betaboon/philips-airpurifier.git
cd philips-airpurifier
source aioairctrl-shell.sh
```

Erste Tests

Werte vom AC4236/10 abrufen:

```
# Pfad zum Tool: /usr/local/bin/aioairctrl
# Alle Infos mit je einer neuen Zeile ausgeben:
aioairctrl --host 192.168.10.103 status --json|  sed -e "s/,/\n/g"

# Gezielt den PM2.5 Wert herausfiltern:
aioairctrl --host 192.168.10.103 status --json|  sed -e "s/,/n/g" | grep -i
```

```
pm25

# oder etwas feiner:
aioairctrl --host 192.168.10.103 status --json|  sed -e "s/,/\n/g" |sed -e
"s/ //g" | grep -i pm25 | cut -d":" -f2

# Remote den Betriebsmodus umstellen:
# -> Automatik Modus:
aioairctrl --host 192.168.10.103 set mode=AG
# -> Sleep Modus:
aioairctrl --host 192.168.10.103 set mode=S
# -> Spees 1 Modus:
aioairctrl --host 192.168.10.103 set mode=M om=1
# -> Spees 2 Modus:
aioairctrl --host 192.168.10.103 set mode=M om=2
# -> Turbo Modus:
aioairctrl --host 192.168.10.103 set mode=T

# Exemplarisch - einen Messwert via ''mosquitto_pub'' publishen:
mosquitto_pub --topic Luftreiniger/pm25/value  -u xxxxxxxx -P 'xxxxxxxxxxx'
-h 192.168.10.11 -p 1883 -m "10"

# das ganze als Komandozeilen Befehl:
pm25=`aioairctrl --host 192.168.10.103 status --json|  sed -e "s/,/\n/g"
|sed -e "s/ //g" | grep -i pm25 | cut -d":" -f2` ; echo "PM25: $pm25";
mosquitto_pub --topic Luftreiniger/pm25/value  -u xxxxxxxx -P 'xxxxxxxxx' -h
192.168.10.11 -p 1883 -m "$pm25"

# Check, ob der Luftreiniger via WLAN erreichbar ist:
ping -c 1 luftreiniger; if [ $? == 0 ]; then echo "AC4236/10 ist online";
else echo "Mist - nix online"; fi
```

Cron Job einrichten

```
sudo su
crontab -e
*/1 * * * *  /home/openhabian/get_pm_value.sh
```

get_pm_value.sh

mcedit /home/openhabian/get_pm_value.sh chmod +x /home/openhabian/get_pm_value.sh

```
#!/bin/bash
#
air_purifier_ip=192.168.10.103
device=PH-AC4236
```

```
# Luftreiniger ID = last 4 digits of MAC address
air_purifier_id="6BB2"

mqtt_broker_ip="192.168.10.11"
mqtt_user="xxxxxxxx"
mqtt_pw="xxxxxxxxxxx"
app=/usr/local/bin/aioairctrl

TIMESTAMP=`date "+%Y%m%d_%H:%M"`
pm25=""
TIMEOUT="40s"

ping -c 1 $air_purifier_ip
if [ $? == 0 ]; then
#  echo -n "Get values from device..."
  pm25=`timeout &TIMEOUT $app --host ${air_purifier_ip} status --json |  sed -e "s/,/\n/g" |sed -e "s/ //g" | grep -i pm25 | cut -d":" -f2`
#  echo " done"
  state="{\"pm25\":$pm25}"
#  echo "$TIMESTAMP: STATE: $state" >> /tmp/luftreiniger.log
#  echo "$TIMESTAMP: STATE: $state"
  mosquitto_pub --topic ${device}-${air_purifier_id}/state  -u $mqtt_user -P $mqtt_pw -h $mqtt_broker_ip -p 1883 -m "$state"
else
  exit 0
# echo "$TIMESTAMP: Luftreiniger ($luftreiniger_ip) nicht erreichbar!" >> /tmp/luftreiniger.log
fi
```

From:
https://www.von-thuelen.de/ - **Christophs DokuWiki**

Permanent link:
https://www.von-thuelen.de/doku.php/wiki/projekte/openhab3/uebersicht

Last update: **2022/01/20 17:09**