

# Home Assistant

Von [jean-claude.feltes@education.lu](mailto:jean-claude.feltes@education.lu)

Dieses Dokument ist stark durch einen Artikel von Denis Bodor in Hackable No. 46 inspiriert.

Danke Denis für die ausführlichen Instruktionen!

Alle Beschreibungen beziehen sich auf einen Linux PC und einen Raspi3, auf dem der HomeAssiztant läuft.

## 1. Installation auf Raspi3

Raspi Imager – Choose OS – Other Specific Purpose OS – Home Assistant and Home Automation

(dieser Menüpunkt ist ziemlich unten in der Liste)

## 2. Ersteinrichtung

Mit der SD-Karte booten, headless ist OK

Mit einem (Linux-)PC im Heimnetz kontrollieren ob sich der Raspi angemeldet hat:

```
jcf@jclab:~$ sudo arp-scan --localnet
Interface: enp3s0, type: EN10MB, MAC: fc:34:97:14:05:bb, IPv4: 192.168.
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp
192.168.0.99    10:bf:48:8a:0f:6d    ASUSTek COMPUTER INC.
192.168.0.100   7c:ff:4d:3d:03:06    AVM Audiovisuelles Marketing un
192.168.0.103   00:e0:4c:36:4b:7e    REALTEK SEMICONDUCTOR CORP.
192.168.0.104   00:04:30:a2:33:b3    Netgem
192.168.0.154   b8:27:eb:04:ae:d1    Raspberry Pi Foundation
```

Alternativ dazu kann man im Router (Fritzbox o.ä.) nachschauen ob sich der Raspi angemeldet hat.

Ein ping sollte ebenfalls positiv ausfallen:

```
jcf@jclab:~$ ping 192.168.0.154
PING 192.168.0.154 (192.168.0.154) 56(84) bytes of data.
64 bytes from 192.168.0.154: icmp_seq=1 ttl=64 time=0.866 ms
64 bytes from 192.168.0.154: icmp_seq=2 ttl=64 time=0.623 ms
```

Im Browser meldet sich der Raspi unter <http://homeassistant.local:8123>

Nach der menügeführten Anmeldung zeigt sich zunächst ein relativ leerer Bildschirm, auf dem nur die Wettervorhersage zu sehen ist.

## 3. Den HA Jargon verstehen

**Device (appareil)** = Objekt das physikalisch mit HA kommuniziert, z.B. ein Smartphone, eine Kamera, ein ESP8266 mit einigen Sensoren.

**Entity** = Element von HA, nicht mit Gerät verwechseln!

Beispiel: ein ESP8266 mit BME280 misst Temperatur, Luftfeuchte, Luftdruck → es gibt 3 Entities in HA, eine für jede gemessene Variable

## 4. Settings

Um alle Möglichkeiten zu erlauben: erst einmal in der linken Toolbar im Profil des Users “Advanced mode” aktivieren.

Dann “Settings” aufrufen.

Im Settings-Menü:

Areas and zones: define rooms

Devices + Services – Add integration

Add-ons (zunächst leer) → head over to the store → sehr viele Addons

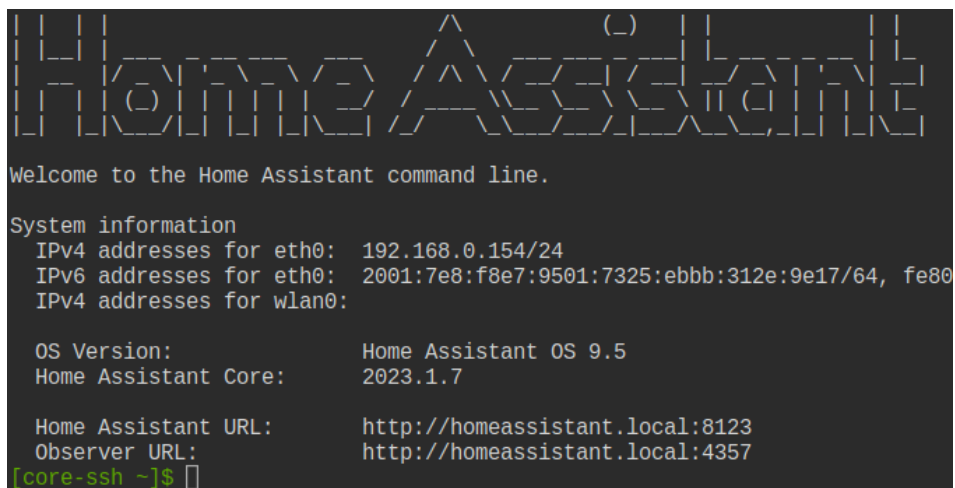
### Terminal

Terminal & SSH (OpenSSH) installieren.

Nach der Installation die Optionen “Start on boot” und “Show in sidebar” aktivieren.

Start klicken

Danach ist das Terminal-Icon in der Sidebar sichtbar und kann gestartet werden:



```

Home Assistant

Welcome to the Home Assistant command line.

System information
  IPv4 addresses for eth0: 192.168.0.154/24
  IPv6 addresses for eth0: 2001:7e8:f8e7:9501:7325:ebbb:312e:9e17/64, fe80
  IPv4 addresses for wlan0:

  OS Version:                Home Assistant OS 9.5
  Home Assistant Core:       2023.1.7

  Home Assistant URL:        http://homeassistant.local:8123
  Observer URL:              http://homeassistant.local:4357
[core-ssh ~]$
  
```

Sie wird mit root-Rechten in einem Docker-Container ausgeführt.

Test: ha help

## 5. SSH-Zugriff konfigurieren

Zunächst ein Schlüsselpaar auf dem externen PC, mit dem auf den Raspi zugegriffen wird, erzeugen:

```
ssh-keygen -t rsa -f ~/.ssh/Harsa
```

Der öffentliche Schlüssel wird angezeigt, er ist auch in der Datei ~/.ssh/Harsa.pub zu finden.

## 6. Mosquitto Broker installieren

<https://www.youtube.com/watch?v=8ol5pO7ZZhk>

Settings – Addons – Store – Search MQTT – Mosquitto broker – Install

Log shows that the broker has started:

```
[14:20:28] INFO: Starting mosquitto MQTT broker...
2023-01-31 14:20:28: Warning: Mosquitto should not be run as root/administrator.
[14:20:31] INFO: Successfully send discovery information to Home Assistant.
[14:20:32] INFO: Successfully send service information to the Supervisor.
```

Ein grüner Punkt oben rechts zeigt dass der Broker läuft.

Broker konfigurieren → Documentation

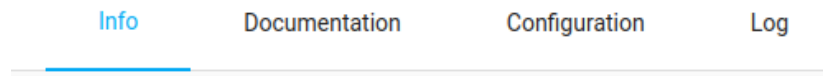
Settings - Integrations – Mosquitto broker – Configure

Später ist diese Konfiguration über Settings – Addons - Mosquitto broker zu erreichen

The screenshot shows the configuration interface for the Mosquitto broker add-on. At the top, there are tabs for 'Info', 'Documentation', 'Configuration', and 'Log'. The 'Info' tab is active, showing the add-on name 'Mosquitto broker', its current version '6.1.3', and a 'ChangeLog' link. Below this, there are buttons for 'Rating' (7), 'Auth', and 'Signed'. A description states it is an Open Source MQTT broker. The main configuration area includes three rows of settings: 'Start on boot' (checked), 'Watchdog' (unchecked), and 'Auto update' (unchecked). To the right, there are three sliders: 'Hostname' (set to 'core-mosquitto'), 'Add-on CPU Usage' (0%), and 'Add-on RAM Usage' (4.5%). At the bottom, there are three buttons: 'STOP', 'RESTART', and 'UNINSTALL'.

Hier kann der Broker gestoppt oder neu gestartet werden.

Wichtig sind oben die Menüpunkte:



**Documentation** → wie der Name sagt: ausführliche Doku

### Configuration

Zum Lesen des Smartyreaders ist ein Login mit User und Passwort festzulegen (hierzu muss dieser natürlich umprogrammiert werden: User und Passwort in config.h festlegen, dann im Hauptprogramm mit dem dazugehörigen #define diese freischalten).

Dies geht am einfachsten durch editieren der YAML-Datei: auf die 3 Punkte rechts klicken, Datei editieren und speichern.

In dieser Datei wird statt des leeren logins: [] das neue Login eingetragen:

```

1 logins:
2   - username:
3     password:

```

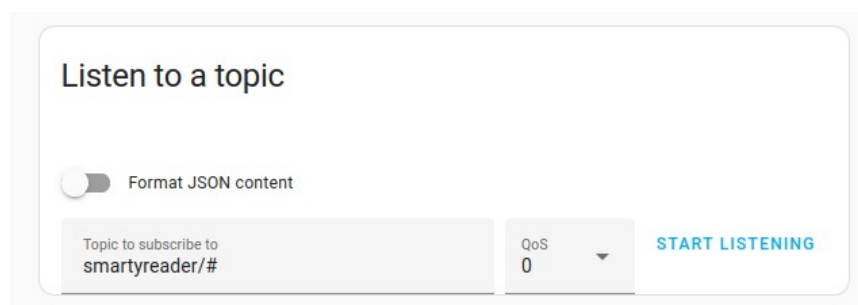
**Achtung auf die YAML – Syntax!**

Wenn man diese Änderungen abspeichert, werden die neuen

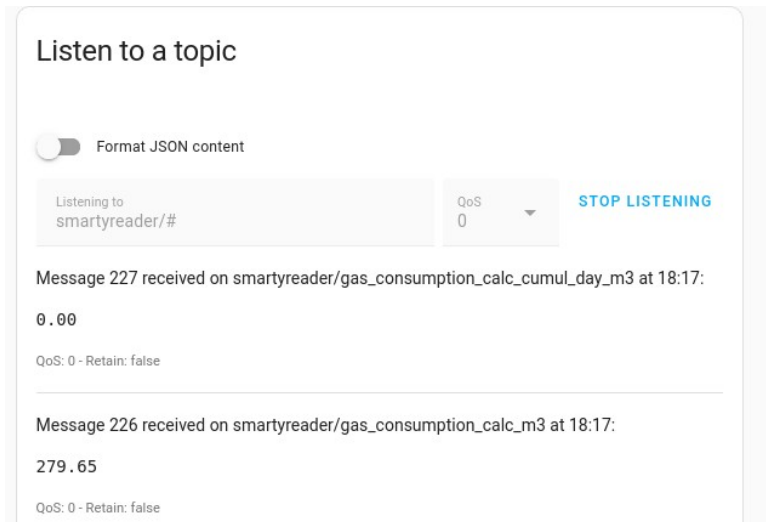
Der Punkt “**Log**” ist bei Problemen hilfreich. Hier habe ich z.B. bemerkt, dass mein Smartyreader wohl versucht hat, Verbindung mit dem Broker aufzunehmen, dies aber anonym nicht zulässig war. Das war dann die Motivation, User und Passwort im Arduino-Code zu aktivieren. Im Log war daraufhin kein Fehler mehr gemeldet.

## 7. MQTT subscriben

Menü links Settings – Devices & Services – core-mosquitto – configure



Mit etwas Geduld sieht man einkommende Meldungen, hier vom Smartyreader:



## 8. Raspi 2 mit Kamera integrieren

### Motioneye auf dem Raspi installieren

(Warum Raspi 2: Raspi 1 ist zu schmalbrüstig, 3 und 4 sind überdimensioniert. MotionEye gibt es aber auch für andere Plattformen wie z.B. Odroid usw.)

MotionEye ist ein Projekt das ein fertiges Betriebssystem für die Raspi-Kamera bietet.

Das Bild wird über Netzwerk übertragen, und so hat man ganz einfach eine fertige WeCam.

- Fertige Image-Datei finden:  
<https://github.com/motioneye-project/motioneyeos/releases>  
**Achtung: die Namen gleichen sich, man muss aufpassen die richtige Datei für Raspi 2 zu nehmen!**
- Mit dem Raspi Imager das OS installieren.
- Raspi mit Kamera booten und nach der IP-Adresse suchen

```
sudo arp-scan -localnet
```

- Im Browser diese Adresse eingeben  
Nun sollte ein Kamerabild angezeigt werden.  
Wenn ein login-Fenster angezeigt wird (nach dem ersten Booten!):  
user = admin, password leer
- Im Menü mit den 3 horizontalen Linien die Parameter einstellen.  
Vor allem kann ein zweiter User definiert werden, der nur sehen kann, aber keine Administrator-Rechte hat.
- Reboot

## 9. Die Kamera in HA integrieren

Settings – Integrations – Add Integration

Select brand

×

Suchen nach motioneye

Search for a brand name  
motioneye

×

Infos zu IP, Administrator und user eingeben.



motionEye

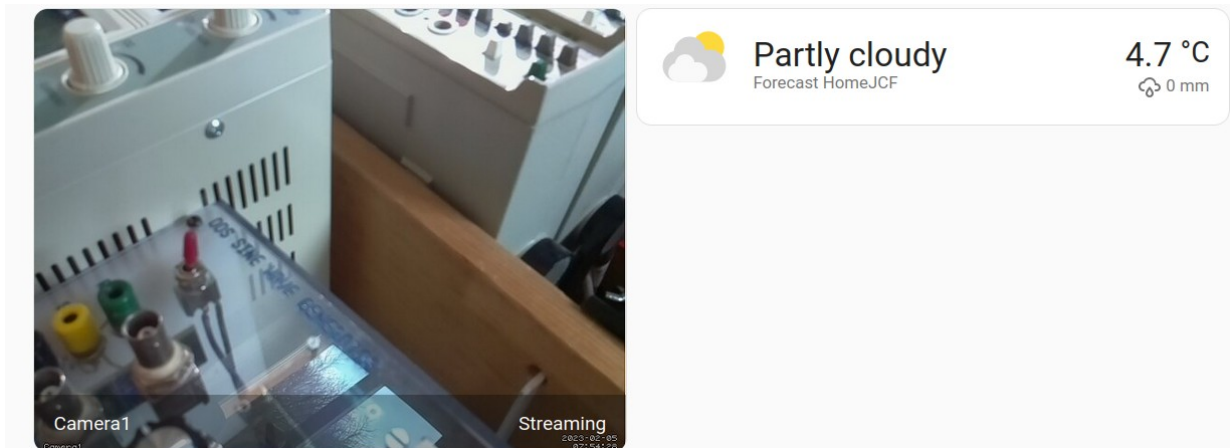
>

Korrekt laufend wird es gefunden.

Achtung, bei der IP-Adresse http:// davor setzen, sonst “Incorrect URL”

Area festlegen. Dies kann ein (eventuell fiktiver) Raum sein oder sonst was.

Nun wird im Overview-Fenster das Kamerabild angezeigt, allerdings defaultmässig nur alle 10s, um Bandbreite zu sparen.



Mit einem Klick auf das Bild erhält man streaming in Echt

## 10. ESPHome: Hello World mit ESP8266 und LED

Das Hello wird in diesem Beispiel von einer LED gemacht, die über das HA Dashboard HA geschaltet werden kann. Es können aber natürlich sehr viel komplexere Dinge gemacht werden, mit vielen unterschiedlichen Sensoren und Aktoren.

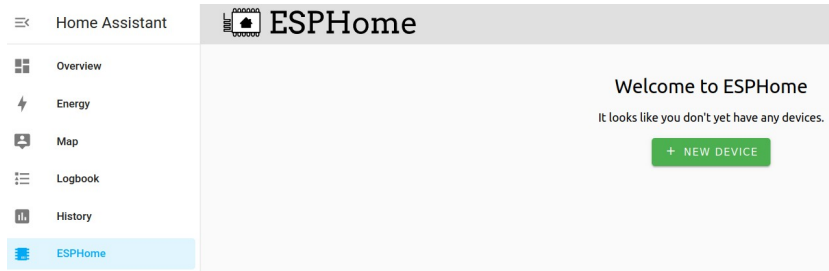
[https://esphome.io/guides/getting\\_started\\_hassio.html#](https://esphome.io/guides/getting_started_hassio.html#)

ESPHome erlaubt das Programmieren von ESP8266, ESP32, Raspi Pico und PicoW.

Installation:

Settings – Addons – nach ESPHome suchen – Installieren

Start – show in sidebar



ESPHome in sidebar starten

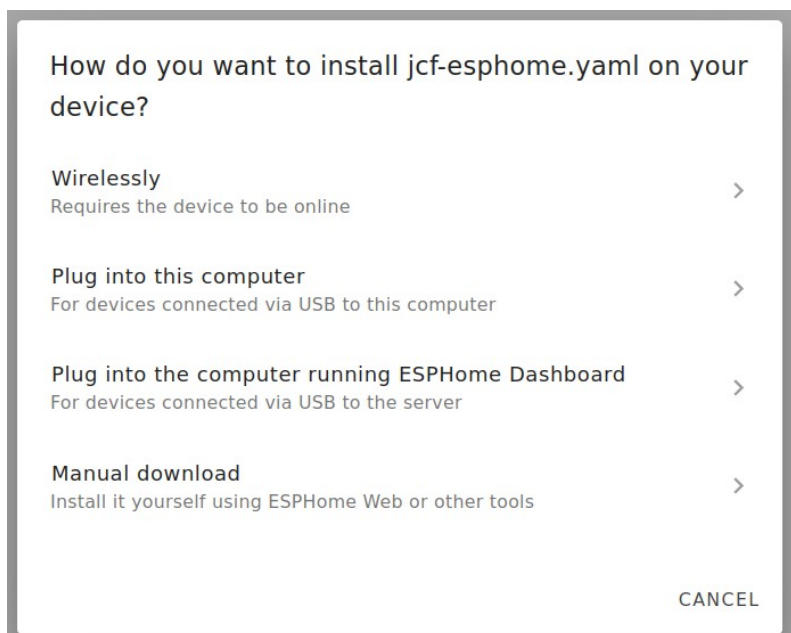
### + New Device button

Den Instruktionen folgen um SSID und Passwort des Netzwerks festzulegen, in das sich der ESP einloggen soll.

Es wird ein Encryption key generiert, den man später für HA braucht, z.B.

UjsV8C9xCzno6zV1/qQ5TMDH/S5zie+HcF7VQ7kUOfQ=

INSTALL



“Plug into the computer running ESPHome dashboard” ausgewählt.

Der ESP8266 muss dann natürlich auch mit dem Raspi auf dem HA läuft verbunden werden.

### Install esp8266-1.yaml

```
INFO Reading configuration /config/esp8266-1.yaml...
INFO Generating C++ source...
INFO Compiling app...
Processing esp8266-1 (board: esp01_1m; framework: arduino; platform: platformio/espressif8266 @ 3.2.0)
-----
Library Manager: Installing ottowinter/ESPAsyncTCP-esp8266 @ 1.2.3
INFO Installing ottowinter/ESPAsyncTCP-esp8266 @ 1.2.3
Unpacking [#####] 100%
Library Manager: ESPAsyncTCP-esp8266@1.2.3 has been installed!
```

Das dauert recht lange. Sehr lange. (Besser ist es gleich die benötigten Sensoren und Aktoren hinzuzufügen, siehe unten)

Nach einiger Zeit kommen schön farbig Meldungen über erfolgreiche Kompilation und Verbindung zum WiFi-Netz (mit IP-Adresse) usw.

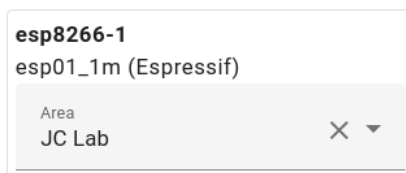
Nun muss dem ESP einer Integration zugewiesen werden:

Settings – Devices & Services – Anklicken ESP8266\_1

Success! ×

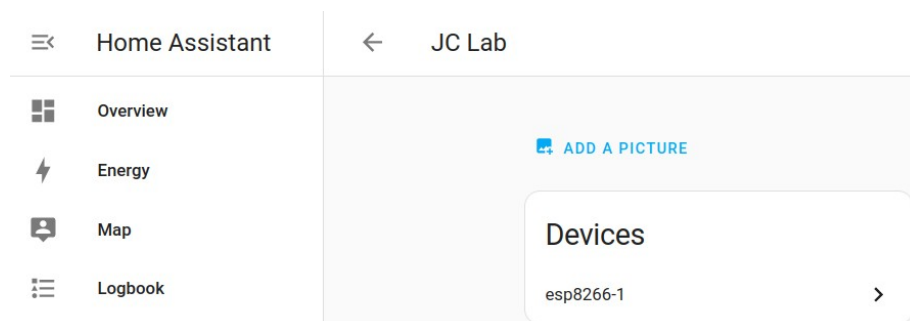
Created configuration for esp8266-1.

We found the following devices:



FINISH

Nach diesem Schritt sieht man in Settings – Areas & Zones dass der ESP zugeordnet wurde:



Jetzt kann die Konfigurationsdatei des ESP editiert werden, um die interne LED des ESP8266 anzusteuern.

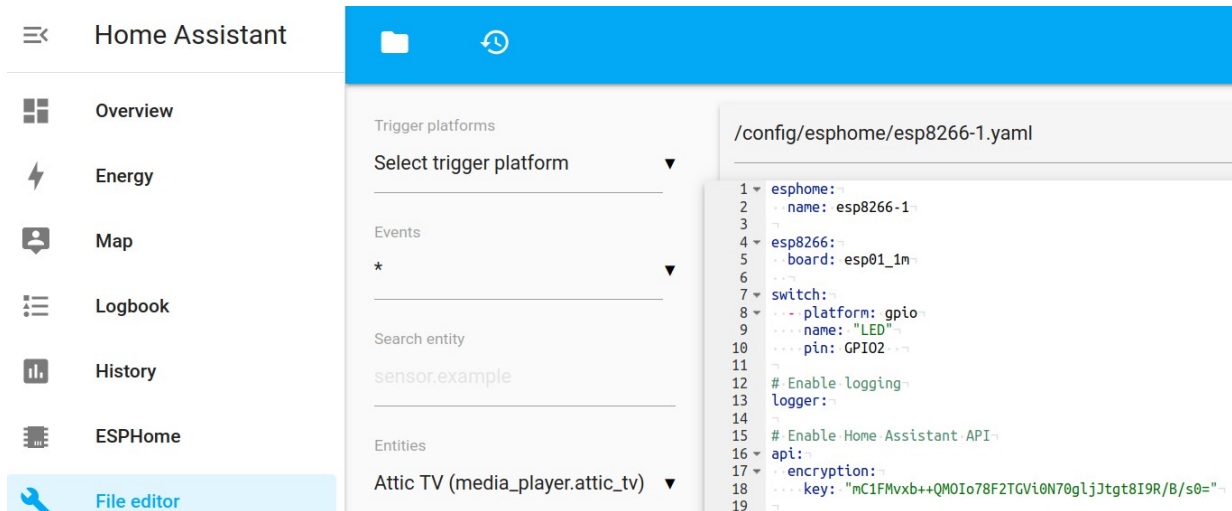
(Man hätte das natürlich auch gleich hinzufügen können)

Das kann mit dem File Editor erledigt werden (diesen zuerst über Settings – Addons installieren). Die Datei befindet sich hier: </config/esphome/esp8266-1.yaml>

Oder einfacher über “EDIT”, siehe unten.

Mit dem File Editor:





Der Abschnitt beginnend mit “switch” wurde hinzugefügt (Die Perspektive ist von HA aus gesehen, von wo aus die LED mit einem (Software-) Schalter ein- und aus geschaltet werden soll.

Achtung auf die YAML-typische Einrückung!

Der File Editor zeigt an, wenn es einen Fehler gibt. Sehr praktisch!

Da die LED von +3.3V gegen den GPIO-Pin geschaltet ist, funktioniert sie invertierend. Dies kann man vor dem langwierigen Kompilieren gleich berücksichtigen:

```
switch:
  - platform: gpio
    name: "LED"
    pin:
      number: GPIO2
      inverted: True
```

Immer darauf achten dass der Editor “grün zeigt” (rechts oben):

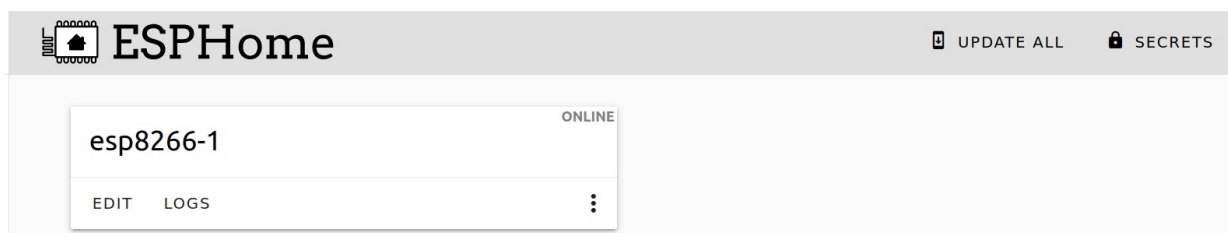
/config/esphome/esp8266-1.yaml



### **Mit “EDIT” von ESPHome:**

Dies geht ganz einfach (eigentlich einfacher als mit dem File Editor):

ESPHome in der linken Leiste anklicken, dann beim ESP8266-1 EDIT klicken:



Daraufhin öffnet sich der Konfigurationseditor:

```

X  esp8266-1.yaml
1  esphome:
2  |   name: esp8266-1
3
4  esp8266:
5  |   board: esp01_1m
6
7  # switch LED from HA:
8  switch:
9  |   - platform: gpio
10 |     name: "LED"
11 |     pin:
12 |       number: GPIO2
13 |       inverted: True

```

Der Editor zeigt hier gleich eventuelle Fehler an, z.B.:

```

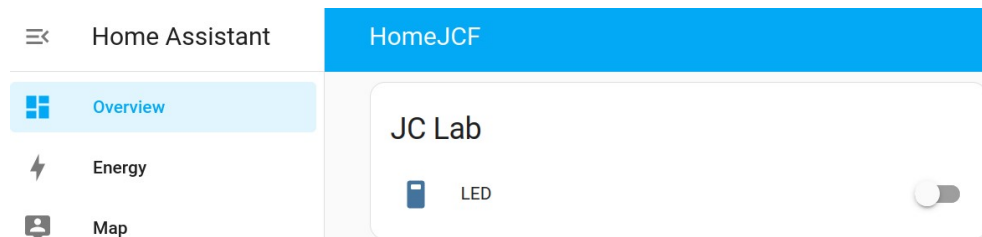
number: GPIO12
inverted: True
mode: INPUT_PULLUP

```

Nach dem Editieren: **“Update all”**

Und schon beginnt wieder ein langer Kompilervorgang...

Danach ist im **Overview-Fenster** ein Schalter hinzugekommen, mit dem man die LED schalten kann:



Das Kompilieren wird **sehr** abgekürzt, wenn man es von der Kommandozeile aus erledigt, nicht auf dem Raspi, sondern auf einem normalen (schnelleren) PC.

## 11. ESPHome: Reagieren auf einen Kontakt

Beispiel:

Ein Kontakt oder Schalter liegt zwischen D6 = GPIO12 eines WEMOS D1mini.

Es wird ein interner Pullup-Widerstand benutzt.

Die zusätzliche Konfiguration wird einfach hinzugefügt:

ESPHome – ESP8266-1 - **EDIT**

In der Konfigurationsdatei wird der Teil für den Schalter hinzugefügt:

```

esphome:
  name: esp8266-1

esp8266:
  board: esp01_1m

# switch LED from HA:
switch:
  - platform: gpio
    name: "LED"
    pin:
      number: GPIO2
      inverted: True

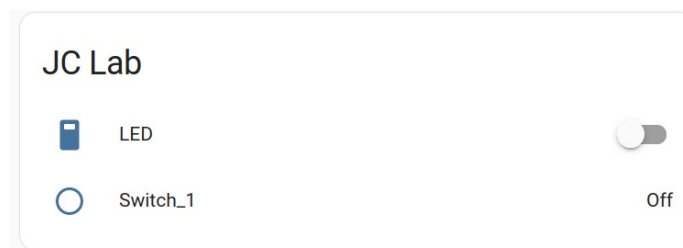
# Show state of a switch in HA
binary_sensor:
  - platform: gpio
    name: "Switch_1"
    pin:
      number: GPIO12
      inverted: True
      mode: INPUT_PULLUP
...

```

*Denis meint, die YAML-Syntax wäre unübersichtlich und schlecht zu lesen, weil Whitespaces eine Bedeutung haben. Dem kann ich nicht zustimmen, denn ein Code der mit {}; usw. verunstaltet ist, ist aus meiner Sicht schwieriger zu lesen. Es gibt aber eine schlechte Entscheidung der YAML-Entwickler: statt wie in Python 4 Spaces zum Einrücken zu nehmen, sind es in YAML nur 2. Und DAS macht den Code unübersichtlich, weil die Einrückungen so schlecht zu sehen sind.*

Danach: UPDATE ALL → Kompilieren und flashen

Anschließend ist im Overview – Fenster eine Anzeige für den Zustand des Schalters hinzugekommen:



Achtung: die Blickrichtung ist immer von HA auf die Devices:

- Der Software - Schalter rechts schaltet die Hardware -LED
- Die Anzeige Switch\_1 mit rechts der Zustandsanzeige gibt den Zustand des Hardware-Schalters an.

Unser System hat übrigens nun in dieser Area 1 device ESP8266-1 und zwei entities: LED und Switch\_1

## 12. Und nun?

HomeAssistant kann sehr viel mehr. So viel, dass man schon einige Zeit braucht, um nur ein wenig durchzublicken. Ausgehend von dem was ich bis jetzt ausprobiert habe, lohnt es sich auf jeden Fall.

## 13. Literatur

Denis Bodor in Hackable, No.46 / 2023

Tutorials:

[https://esphome.io/guides/getting\\_started\\_hassio.html#](https://esphome.io/guides/getting_started_hassio.html#)

<https://www.elektormagazine.com/labs/how-to-home-assistant-esphome>

<https://peyanski.com/complete-esphome-installation-guide/>

Config structure:

<https://savjee.be/blog/how-i-structure-my-esphome-config-files/>