

# Sonoff 10A WiFi Power Switch

*Was tun mit einem neuen  
SONOFF 10A ?*

SONOFF 10A in openHAB  
funktionsfähig machen mittels MQTT

Als Thread (von RiotMode) zu finden hier - inkl. Links:

<https://community.openhab.org/t/sonoff-mqtt-and-openhab/33951/13>

## Was man dafür braucht:

- **Hardware:** USB zu TTL-Konverter-Modul mit eingebautem in CP2102  
[https://www.amazon.de/gp/product/B00AFRXKFU/ref=oh\\_aui\\_detailpage\\_o01\\_s00?ie=UTF8&psc=1](https://www.amazon.de/gp/product/B00AFRXKFU/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1)
- **Software:** Flash Download Tools V3.4.9.2 (ESP8266 & ESP32)  
<http://espressif.com/en/support/download/other-tools>
- **Software:** ESPEasy 2.0  
<https://github.com/letscontrolit/ESPEasy/releases>



USB zu TTL-Konverter-Modul mit eingebautem in CP2102

von WINGONEER®



123 Kundenrezensionen | 17 beantwortete Fragen

Preis: **EUR 7,99** ✓prime

Alle Preisangaben inkl. USt

**Auf Lager.**

Lieferung Freitag, 8. Sept.: Bestellen Sie innerhalb 12 Stunden und 22 Minuten per Pro Kasse. [Siehe Details.](#)

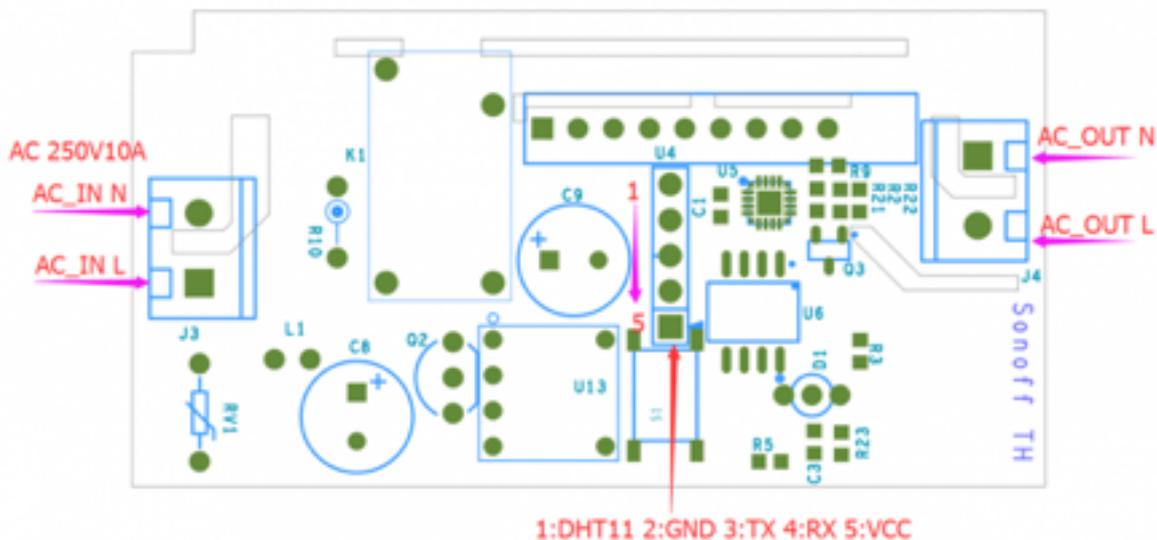
Verkauf durch 3C4U und Versand durch Amazon. Für weitere Informationen, Impressum, Widerrufsrecht klicken Sie bitte auf den Verkäufernamen. Geschenkverpackung verfügbar

1 neu ab EUR 7,99

[Falsche Produktinformationen melden](#)

## Anleitung:

1. Installiere mittels einem USB zu TTL-Konverter-Modul das ESPEasy auf das Sonoff
2. Verbinde den TTL-Konverter mit 4 PIN-Kabel mit den richtigen PINS zum SONOFF
3. GND=GND / TX=RX / RX=TX / VCC=3v3  
**VORSICHT: Hier unbedingt 3V und nicht 5V verwenden!!**  
**Ansonsten ist der Sonoff defekt. - Kontrolliere nochmals!**



4. Wir stecken nun den Konverter in den USB Slot und der Konverter wie das Sonoff bekommen Strom und das Sonoff leuchtet.
5. Nun bringen wir den Sonoff in den „Config-Mode“ indem wir das 3v3 Kabel am Konverter ausstecken, danach drücken wir die Config-Taste (schwarzer Stift) auf dem Sonoff, nun stecken wir den 3v3 (3Volt) PIN wieder ein und das Sonoff sollte nun nicht mehr leuchten.
6. Wir befinden uns jetzt im CONFIG-MODE auf dem Sonoff  
Nun kommen wir zur Firmware (ESPeasy) Installation

7. Download:

- Flash Download Tools V3.4.9.2
- ESPEasy 2.0

8. Unzip „Flash Download Tools" and run  
ESPFlashDownloadTool\_v3.4.9.2.exe

9. Select: ESPEasy\_v2.0.0-dev11\_normal\_1024.bin  
in the tool (pressing ... button)

10. Type adress after @ 0x00000

11. Checkbox before the pathname checked.

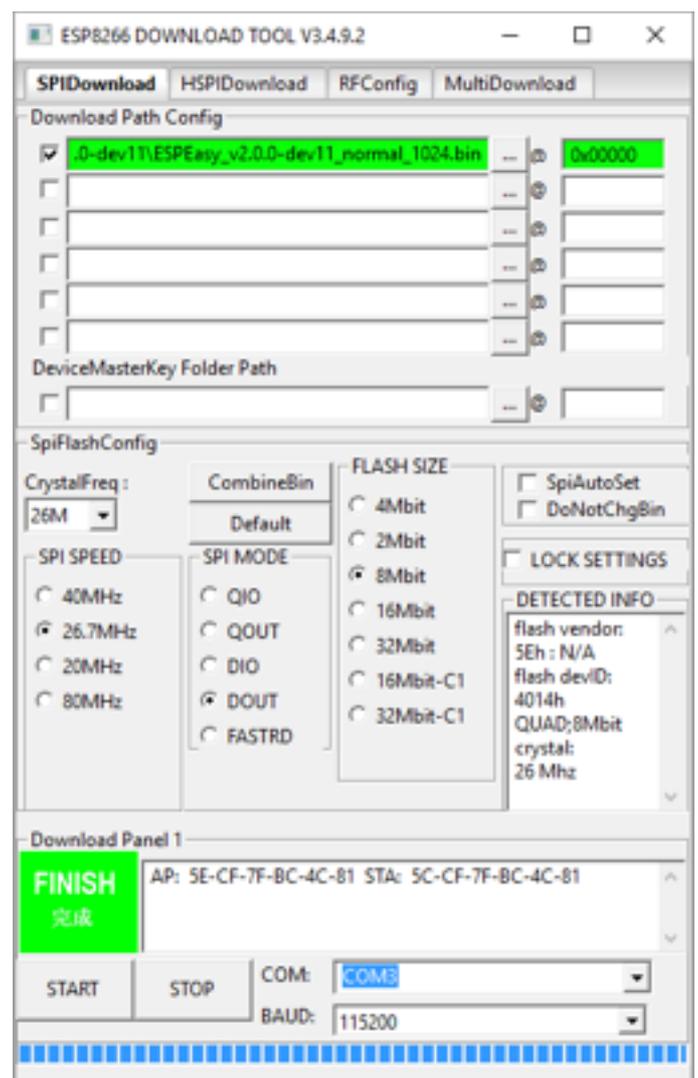
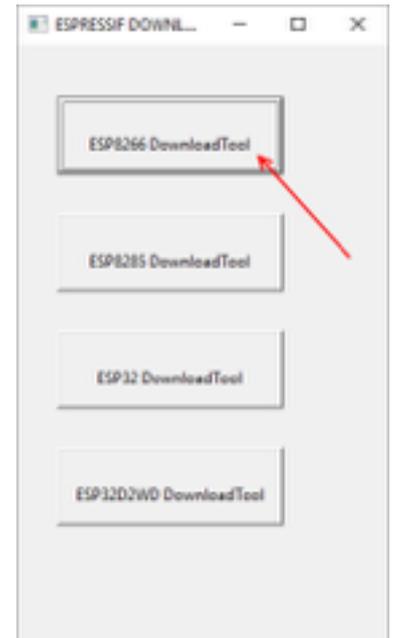
12. Settings:

- CrystalFreq: 26M
- SPI SPEED: 26.7 MHz
- SPI MODE: DOUT
- FLASH SIZE: 8MBit

13. Select

- COM PORT:** COM4  
(in my case COM4)
- BAUD:** 115200

14. Hit Start-Button



## Erfolgreicher Screen:

```
=====
EFUSE NORMAL MODE
=====
EFUSE CHECK PASS...
48bit mac
debug:
=====
 5c cf 7f bc 4c 81
CUSTOM_ID: 06 40 00 00 00 00 30 00
CUSTOM_ID: 0640000000093000
=====
crc_efuse_4bit: 0
crc_calc_4bit: 5
48bit mac
MAC AP : 5E-CF-7F-BC-4C-81
MAC STA: 5C-CF-7F-BC-4C-81
get mac res: True
('ttest uuuuuuuuuuart : uart reg: ', 462)
(' baudrate: ', 115200)
get crystal: 26611200
get flash id : 0x7714405e
manufacturer_id: 0x5e

device_id: 0x4014

vendor: 94
mode: 64
size: 20
DEBUG!!!
SET FLASH PARAMS
filename: Z:\Mijn ESPEasy\FLASH_DOWNLOAD_TOOLS_V3.4.9.2\bin_tmp\downloadPanel1\
ESPEasy_v2.0.0-dev11_normal_1024.bin_rep
offset : 0
Erasing flash...
=====
pic path: ./RESOURCE/DOWNLOAD_S.bmp
=====
Took 3.19s to erase flash block

Leaving...
com closed
=====
pic path: ./RESOURCE/FINISH_S.bmp
=====
```

Nun sollte man das Sonoff als WiFi-Netz mit SSID: „**ESP\_01**“ finden  
Das WLAN-Passwort ist: **configesp**

Jetzt sollte automatisch ein „Login Screen“ auf einem iPad kommen, und man befindet sich im GUI des Sonoff. Nun eigenes **WLAN** wählen ( in meinem Fall „Piratenbucht“ ) und WLAN-Passwort eingeben.

1. Das Sonoff sagt uns jetzt seine neue IP und sollte von selbst neu starten.
2. Wir verbinden uns mit einem Browser auf die Sonoff-IP  
( in meinem Fall <http://192.168.1.5> )
3. Unter Config (Unit Name) wählt man einen **eindeutigen Namen** ( zb. Sonoff\_1 )
4. Wählt ein neues Admin Passwort
5. Nun zum Reiter „Controllers“

The screenshot shows the configuration page for 'ESP Easy Mega: Sonoff\_1' with the 'Config' tab selected. The page is divided into several sections:

- Main Settings:** Unit Name: Sonoff\_1, Unit Number: 1, Admin Password: [redacted]
- WiFi Settings:** SSID: Piratenbucht, WPA Key: [redacted], Fallback SSID: [empty], Fallback WPA Key: [empty]
- WPA AP Mode Key:** [redacted]
- IP Settings:** ESP IP: [empty], ESP GW: [empty], ESP Subnet: [empty], ESP DNS: [empty]. Note: Leave empty for DHCP
- Sleep Mode:** Sleep enabled:  (with a blue question mark icon), Sleep Delay: 60 [sec], Sleep on connection failure:

A blue 'Submit' button is located at the bottom right of the form.

6. Klicke EDIT

7. **Protocol:** OpenHAB MQTT  
**LocateController:** Use IP Adress  
**Controller IP:** 192.168.1.100 ( die IP seines MQTT-Brokers, in diesem Fall der Raspb.)  
**Controller Port:** 1883  
**Username:** vom MQTT Broker ( openhabian )  
**Password:** vom MQTT Broker ( password )

The screenshot shows the configuration page for 'ESP Easy Mega: Sonoff\_1' with the 'Controllers' tab selected. The 'Controller Settings' section is visible:

- Protocol:** OpenHAB MQTT (with a blue question mark icon)
- Locate Controller:** Use IP address
- Controller IP:** 192.168.1.100
- Controller Port:** 1883
- Controller User:** [empty]
- Controller Password:** [redacted]
- Controller Subscribe:** /%sysname%/#
- Controller Publish:** /%sysname%/%tsknam
- Enabled:**

At the bottom right, there are 'Close' and 'Submit' buttons.

8. Nun zum Reiter „Hardware“

9. **Pin LED: GPIO-13 (D7)**  
**Inversed LED: Haken setzen**



10. Im Reiter „**Devices**“ können wir nun noch Parameter für OpenHAB ausgeben lassen. ( sonoff-uptime, WiFi-Stärke,..... )

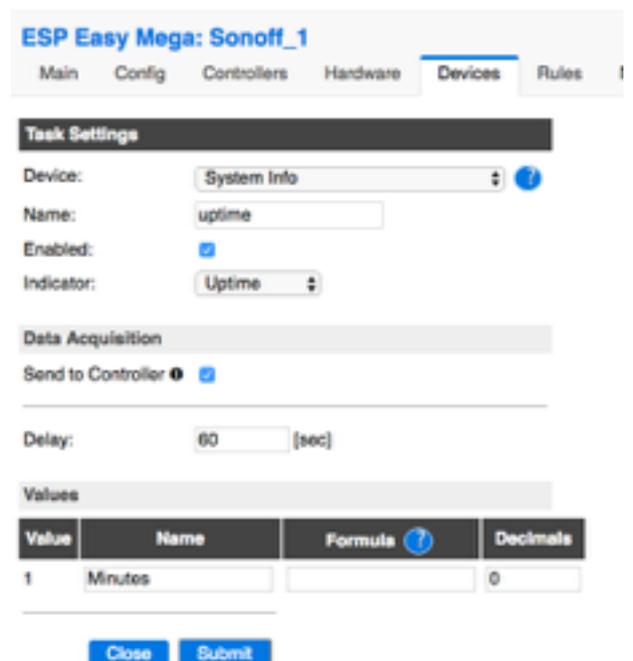
11. Klicke EDIT und lege 4 Devices an:

12. **Sonoff-uptime**

**Device:** System Info  
**Name:** uptime  
**Enabled:** yes  
**Indicator:** uptime  
**Send to Controller:** 60

**Value Name:** Minutes  
**Decimals:** 0

13. Submit & Close



## 14. Wifi-Stärke

**Device:** System Info

**Name:** WI-FI

**Enabled:** yes

**Indicator:** Wifi RSSI

**Send to Controller:** 60

**Value Name:** Minutes

**Decimals:** 0

## 15.lightState

### ESP Easy Mega: Sonoff\_1

Main Config Controllers Hardware **Devices** Rules

#### Task Settings

Device:  ⓘ

Name:

Enabled:

Global Sync:

#### Sensor

Internal PullUp:

Inversed Logic:

*Note: Will go into effect on next input change.*

1st GPIO:

Switch Type:

Switch Button Type:

Send Boot state:

#### Data Acquisition

Send to Controller ⓘ

---

Delay:  [sec] (Optional for this Device)

#### Values

Value	Name
1	<input type="text" value="Switch"/>

## 16.lightSwitch

### ESP Easy Mega: Sonoff\_1

Main Config Controllers Hardware **Devices** Rules

---

#### Task Settings

Device:  ⓘ

Name:

Enabled:

Global Sync:

---

#### Sensor

Internal PullUp:

Inversed Logic:

*Note: Will go into effect on next input change.*

1st GPIO:

Switch Type:

Switch Button Type:

Send Boot state:

---

#### Data Acquisition

Send to Controller ⓘ

---

Delay:  [sec] (Optional for this Device)

---

#### Values

Value	Name
1	<input type="text" value="Switch"/>

---

Powered by [www.letscontrolit.com](http://www.letscontrolit.com)

## 17. Nun zum Reiter „Tools“

18. Im Reiter „Tools“ auf **advanced“**

**Rules:** JA

**MQTT Retain MSG:** JA

**Enable Serial Port:** 115200

**Global Sync:** JA

**UDP port:** 65500

19. Klicke auf **Tools**

20. Klicke auf **Reboot**

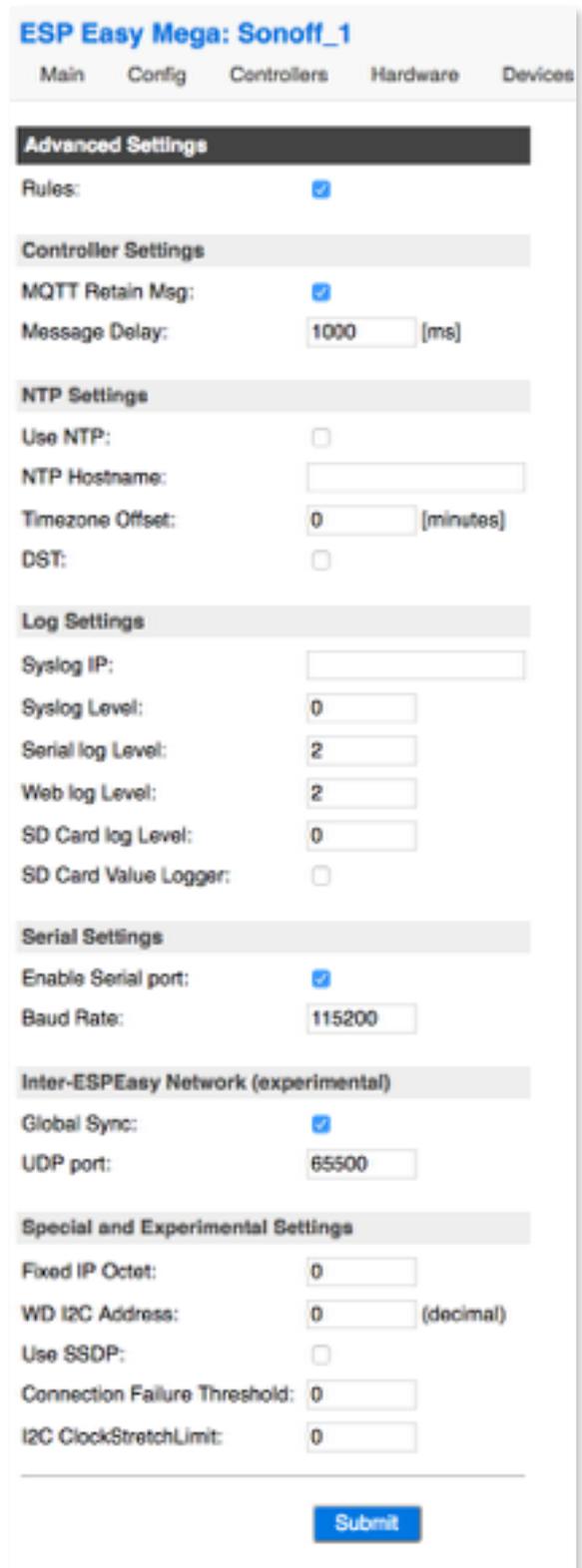
21. **Verbinde dich erneut mit dem Sonoff** ( <http://192.168.1.5> )

22. Klicke auf **Tools**

23. Klicke auf **LOG**

24. Überprüfe:

**MQTT: Connected to Broker**



25. **geschafft !** - Nun testen wir mittels **MQTT.fx** die Devices

# MQTT - Getting Started

## Raspberry PI mit openhabian

1. Install and set up a broker. (**EDIT: „services/mqtt.cfg“**)  
You've used the one provided through openHABian, perfect
2. Now get the openHAB binding connected (then tested)
  - Install the binding through Paper UI (**mqtt-binding**)
  - Configure it via /services/mqtt.cfg
    - **broker.url=192.168.1.100:1883**
    - **broker.user=openhabian**
    - **broker.pwd=password**
3. Ein **reboot** könnte hier nicht schaden
4. **Check the log** to see if the broker connects successfully:  
<http://openhabdevice-ip:9001> in meinem Fall <http://192.168.1.100:9001>
5. **Check the log** - You should be seeing something like  
[INFO ] [penhab.io.transport.mqtt.MqttService] - MQTT Service initialization completed.  
[INFO ] [t.mqtt.internal.MqttBrokerConnection] - Starting MQTT broker connection ‚broker‘
6. Mit **MQTT.fx** auf den **Broker verbinden** und testen.  
**Siehe dazu: MQTT Subscribe Test**  
(mqtt-spy is a better choice but mqtt.fx is okay)

## Ein einfacher MQTT - Test

1. In der **home.items** folgendes Eintragen:

```
Switch MQTT_Test "Testing..." { mqtt="<[broker:testing/mqtt/topic:state:default]" }
```

2. In der **home.sitemap** folgendes Eintragen:

```
Switch item=MQTT_Test
```

3. Erster Test mit **MQTT.fx**

4. **Publish** the Message **“ON”** to Topic **“testing/mqtt/topic”**

5. Check the **log** once again!

<http://192.168.1.100:9001>

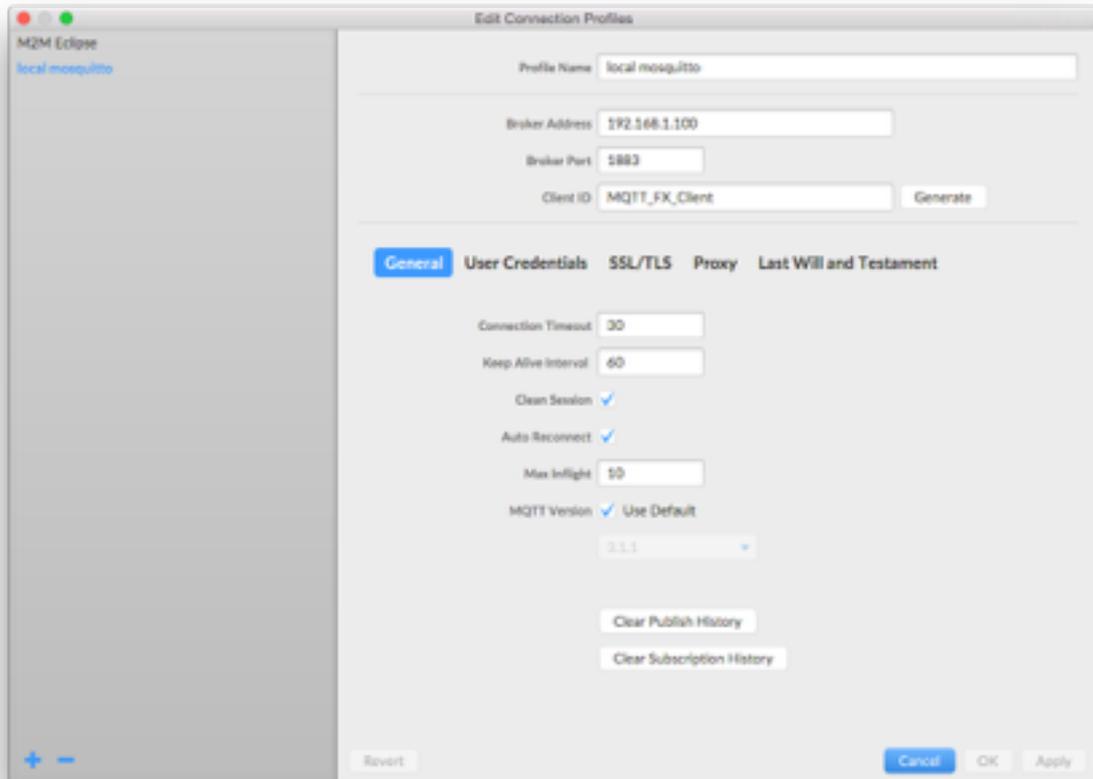
6. MQTT-Test - Du solltest das im LOG sehen:

```
[ItemStateChangedEvent ] - MQTT_Test changed from OFF to ON
```

7. **Super MQTT läuft!**

# MQTT.fx - Publish Test

1. MQTT.fx öffnen und auf das Zahnrad klicken
2. Unter **Broker Adresse** die Raspberry-IP ( 192.168.1.100 ) und Port 1883 wählen



3. **Apply** und **Cancel**

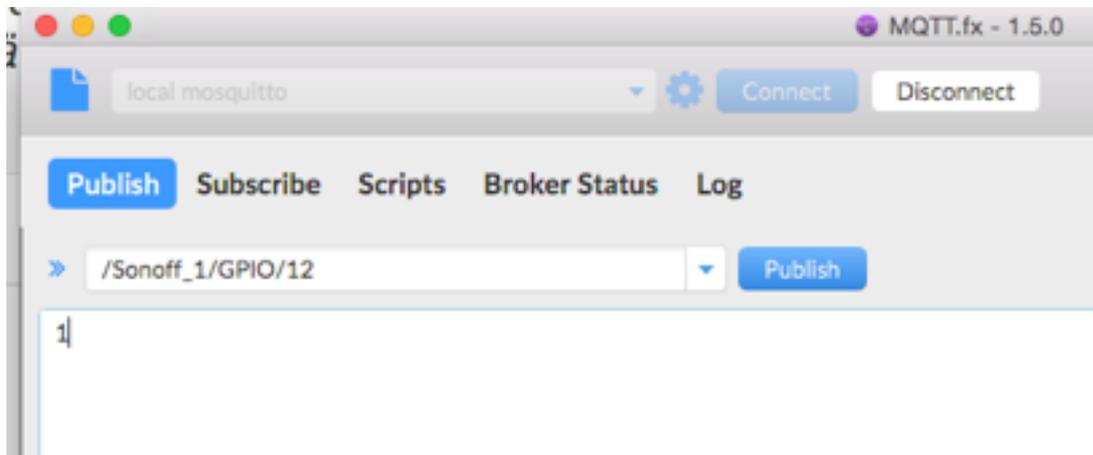
4. Klicke **CONNECT**

5. Jetzt testen wir ob der Schalter auch funktioniert

6. Wir öffnen auf dem Sonoff das **LOG** und klicken in MQTT.fx auf den Reiter **Publish**

7. Dort in das DropDown: **/Sonoff\_1/GPIO/12**

8. Als Text darunter **1** oder **0**



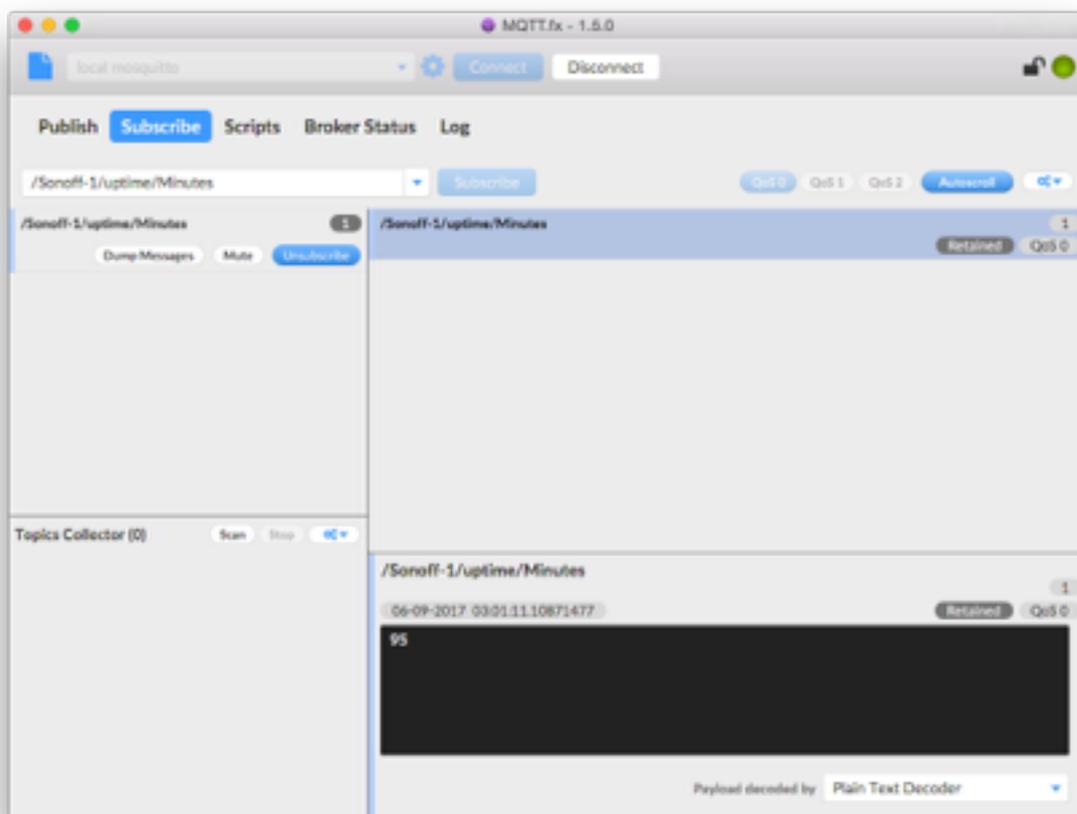
9. Jetzt klicken wir auf **Publish**

10. **Das Log sollte die Schaltung nun anzeigen**

```
67310 : SYS : 1.00
67310 : EVENT: uptime#Minutes=1.00
67376 : SYS : -68.00
67377 : EVENT: WI-FI#=-68.00
87945 : SW : GPIO 12 Set to 1 ←
87947 : SW : State 1
87968 : EVENT: lightState#Switch=1.00
90230 : SW : GPIO 12 Set to 0 ←
90324 : SW : State 0
90324 : EVENT: lightState#Switch=0.00
```

# MQTT.fx - Subscribe Test

1. Klicke in MQTT.fx auf den Reiter **Subscribe**
2. Im DropDown eintippen: **/Sonoff\_1/uptime/Minutes**
3. Klicke auf **Subscribe** (Blauer Button)  
Links sollten wir den Subscribe sehen und rechts unten den Wert in (95) Minuten uptime



4. **Super - es funktioniert!**

# SONOFF in openHAB einbinden

1. In der **home.items** folgendes Eintragen: ( hier für „Sonoff\_1“ )

Switch Sonoff\_1 "Stehlampe Büro" <light> (LR,gLight)

```
{ mqtt=">[broker:/Sonoff_1/GPIO/12:command:ON:1],  
  >[broker:/Sonoff_1/GPIO/12:command:OFF:0],  
  <[broker:/Sonoff_1/GPIO/12:state:ON:1],  
  <[broker:/Sonoff_1/GPIO/12:state:OFF:0]" }
```

Number Sonoff\_1\_uptime "Sonoff-Uptime [%].0f Minuten]"

```
{mqtt="<[broker:/Sonoff_1/uptime/Minutes:state:default]"}
```

Number Sonoff\_1\_wifi "WiFi-Signalstärke [%].2f dB]"

```
{mqtt="<[broker:/Sonoff_1/WI-FI/:state:default]"}
```

Number Sonoff\_1\_state "Sonoff-state [%].0f 1=Ein/0=Aus]"

```
{mqtt="<[broker:/Sonoff_1/lightState/Switch:state:default]"}
```

```
// SONOFF - FIRST ITEM SUCCESS  
  
Switch Sonoff_1 "Stehlampe Büro" <light> (LR,gLight)  
  { mqtt=">[broker:/Sonoff_1/GPIO/12:command:ON:1],  
    >[broker:/Sonoff_1/GPIO/12:command:OFF:0],  
    <[broker:/Sonoff_1/GPIO/12:state:ON:1],  
    <[broker:/Sonoff_1/GPIO/12:state:OFF:0]" }  
  
Number Sonoff_1_uptime "Sonoff-Uptime [%].0f Minuten]"  
  {mqtt="<[broker:/Sonoff_1/uptime/Minutes:state:default]"}  
  
Number Sonoff_1_wifi "WiFi-Signalstärke [%].2f dB]"  
  {mqtt="<[broker:/Sonoff_1/WI-FI/:state:default]"}  
  
Number Sonoff_1_state "Sonoff-state [%].0f 1=Ein/0=Aus]"  
  {mqtt="<[broker:/Sonoff_1/lightState/Switch:state:default]"}
```

2. In der **home.sitemap** folgendes Eintragen:

```
Frame label="Büro"  
{  
  Switch item=Sonoff_1  
  Text item=Sonoff_1_state  
  Text item=Sonoff_1_uptime  
  Text item=Sonoff_1_wifi  
}
```

```
Frame label="Büro"  
{  
  Switch item=Sonoff_1  
  Text item=Sonoff_1_state  
  Text item=Sonoff_1_uptime  
  Text item=Sonoff_1_wifi  
}
```

3. **Fertig** - Nun solltest du das Sonoff mit OpenHab steuern können

**Hinweis:** Das Relais hört man erst, wenn 230V angeschlossen sind.

**Büro**

Stehlampe Büro  Sonoff-state 1 1=Ein/0=Aus

Sonoff-Uptime 0 Minuten WIFI-Signalstärke -62.00 dB