

# Setup Smart Switches

Tasmota Flashen und Konfiguration

Version 1.4

## Inhaltsverzeichnis

1. Ausgangslage.....	3
2. Typen von Smart Switches.....	3
3. Firmware Smart Switch .....	4
4. Flashen Tasmota .....	5
5. Netzkabel anschliessen.....	8
6. Konfiguration Tasmota Smart Switch.....	10
6.1    Schritt 1 der Konfiguration .....	10
6.2    Schritt 2 der Konfiguration .....	13
6.3    Schritt 3 der Konfiguration .....	14
7. Funktion Smart Switch.....	15
8. Testen des Smart Switches .....	15

## 1. Ausgangslage

Der Switcher wurde ursprünglich (Version 1) für die Steuerung von 433 MHz Funksteckdosen entworfen und entwickelt. Die Version 2, und natürlich auch die Version 3 des Switchers können zudem auch die WiFi-basierten sog. Smart Switches ansteuern. Die Funksteckdosen sind bereits seit einiger Zeit nur noch schwer im Markt zu finden und die Zukunft gehört klar den Smart Switches.



*Old Technology: Funksteckdosen 433 MHz*

## 2. Typen von Smart Switches

Von Sonoff gibt es verschiedene Modelle von Smart Switches, hier Fotos



*Sonoff Modell Basic, verkabelt*



Sonoff Modell TH10 (mit Anschluss für Temp. Sensor)

Daneben gibt es noch den kleinen Shelly 1 Smart Switch



Shelly 1

Switcher 3 kann alle diese WiFi basierten Switches ansteuern. Alle verwenden das MQTT Protokoll.

[Siehe Sonoff Wiki für weiter Details](#)

### 3. Firmware Smart Switch

Diese Devices basieren alle auf ESP8266 oder ESP8265 und sie enthalten eine Firmware, die vorgängig **ersetzt** werden muss (neu flashen).

Die Firmware der Wahl für DIY Anwendung heisst **TASMOTA**, sie hat sich zum Quasi-Standard (Open Source) gemausert und wurde von einem gewissen Theo Arends entwickelt.

Allerdings gibt es Anbieter, die Sonoff Devices anbieten, die bereits Tasmota enthalten, dann muss man das Device nicht selbst flashen.

Tasmota ist sehr **umfangreich**, die Docu ist überwältigend. Dies ist ein typisches Beispiel einer mit Funktionen **überladenen** OpenSource Software. Glücklicherweise braucht man für die Anwendung mit

Switcher nur wenig zu wissen. Das vorliegende Dokument macht dies zudem sehr einfach.

Hier die wichtigsten Links zu Tasmota:

[Tasmota Docu Getting Started](#)

[Tasmota on GitHub](#)

[Ausführliche Doku Tasmota](#)

[Discussion Tasmota](#)

[Tasmota Downloads](#)

Unter Downloads sind viele verschiedene Versionen von Tasmota zu finden - ich verwende meist folgende Version:

- **tasmota-lite.bin**

Dieses .bin muss also auf den ESP Chip geschrieben werden - dies funktioniert genauso wie das flashen sonstiger Software auf die ESP Chips.

Ich verwende dazu meinen Windows 10 Laptop - einziger Grund dafür ist die App **Tasmotizer**, die es leider für Mac OS Big Sur (noch) nicht gibt.

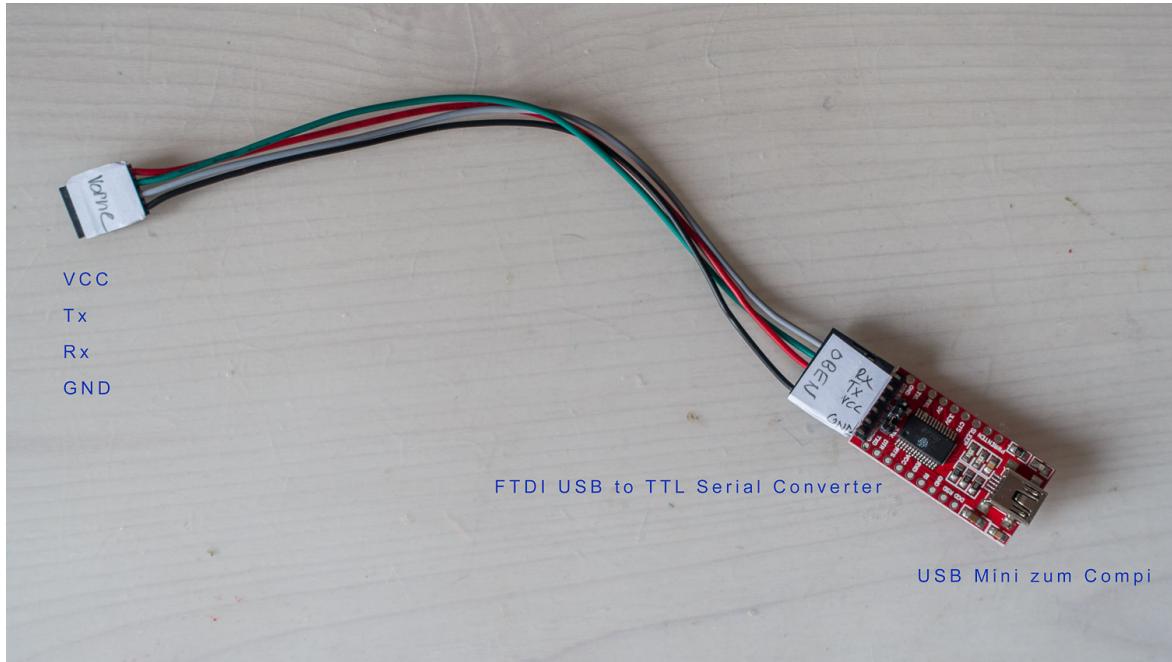
[Download Tasmotizer](#)

## 4. Flashen Tasmota

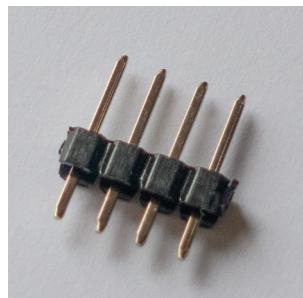
Es braucht also folgendes:

- die App Tasmotizer
- das File tasmota-lite.bin
- ein FTDI UBS to Serial Adapter (unbedingt auf 3.3 Volt einstellen)
- ein 4-pol Kabel, ACHTUNG: TX und RX müssen überkreuzt sein.

FTDI und Kabel sehen so aus:

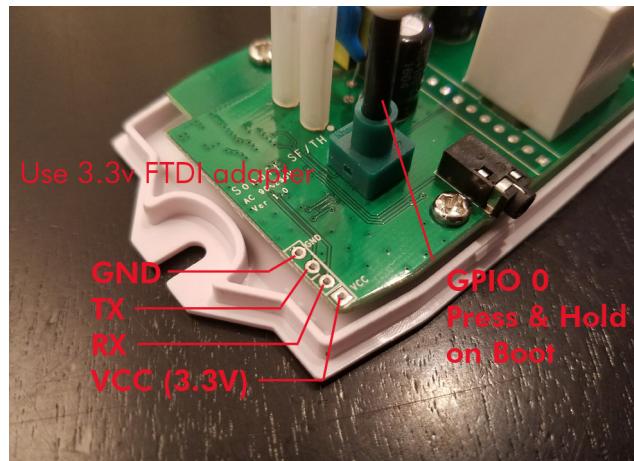


Die grünen PCB der Smart Switches müssen ausgebaut werden und es muss ein 4-poliger Header in die vorbereiteten Löcher gelötet werden. Das ist manchmal etwas schwierig, da der Lötzin oft nicht angenommen wird - zuerst mit feinem Messer etwas kratzen und genügend Flussmittel geben.

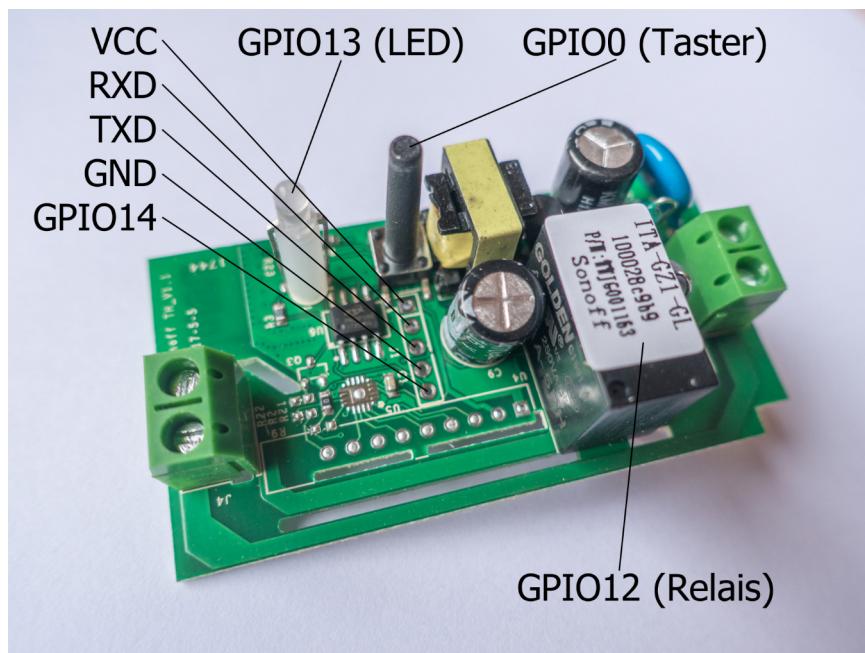


4-Pol Header zum Einlöten auf die Sonoff Platine

Hier Bild des Sonoff TH10 vor dem Löten des Headers



Sonoff TH10 Pin-Out



Sonoff Basic Pin-Out

### Note:

Bei beiden Modellen genügt ein 4-Pin Header, der GPIO14 Pin bei Model Basic braucht keinen Pin.

Achtung auf aufs korrekte Einsticken des Kabels am Sonoff: es sind nicht beide Modelle gleich orientiert bei den Anschlüssen.

Über das Flashen selbst gibt es genügend Tutorials, das sei hier nicht wiederholt. Wiederholt sei hier aber, dass für das Flashen die Stromversorgung (des ESP8266 Chips) nur vom FTDI Adapter kommen darf (3.3 Volt). Auf **keinen Fall** den Switch mit 220 Volt verbinden!

[Tutorial 1](#)

[Tutorial 2](#)

[Tutorial 3](#)

## 5. Netzkabel anschliessen

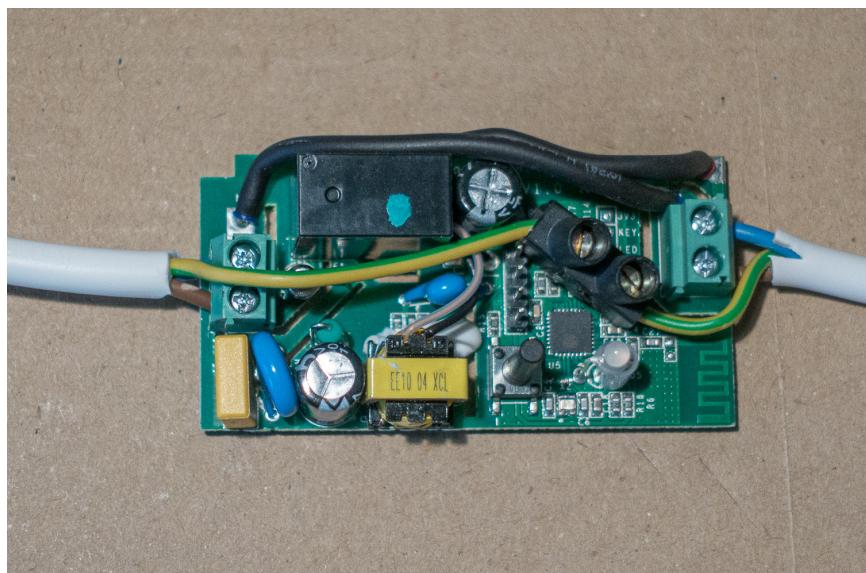
Nach dem Flashen der Tasmota Software können die Netzkabel angeschlossen werden. Danach können die Smart Switches konfiguriert werden.

Bei den Sonoff Basic Smart Switches (siehe folgendes Bild) ist das Anbringen der 220 Volt Zuleitungen nicht ganz trivial, auch muss man die Schutzerde selbst vom Eingang zum Ausgang führen. Und: es kann leicht passieren, dass bei dieser Uebung der Pushbutton mit dem langen Hals **abbricht** - ich musste schon 2 Devices wegwerfen: sind ohne Pushbutton nicht zu verwenden. Also aufpassen.

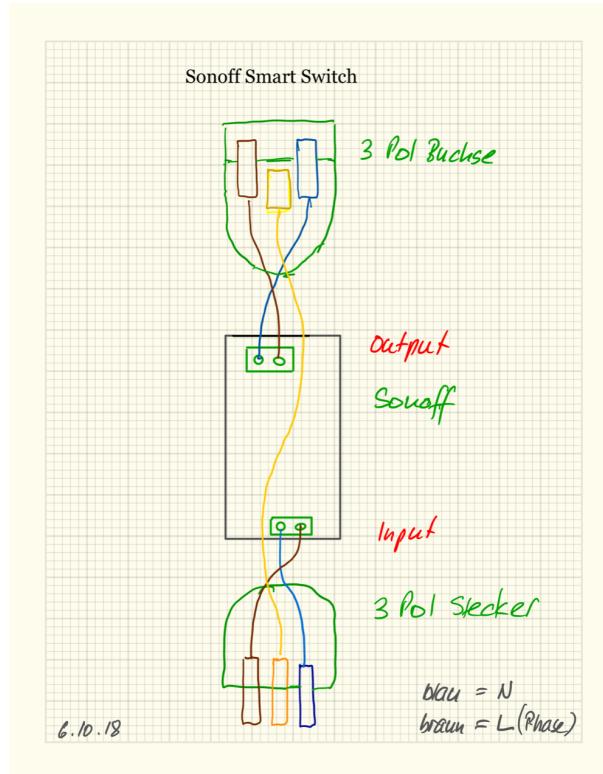


Sonoff Basic Switch

Das folgende Bild zeigt die Anschlüsse bei einem Basic Smart Switch. Das Erdkabel wird mit einer einfachen Lüsterklemme verbunden.



Verkabelung Sonoff Basic



Skizze Verkabelung Sonoff Basic (für CH)

Beim TH10 ist dies wesentlich einfacher, ist auch für Schutzerde designed.



Verkabelung bei Sonoff TH10

Für den Shelly 1 wünscht man sich ein 3-D gedrucktes Gehäuse, das Platz bietet für Lüsterklemmen und Zuführungen

## 6. Konfiguration Tasmota Smart Switch

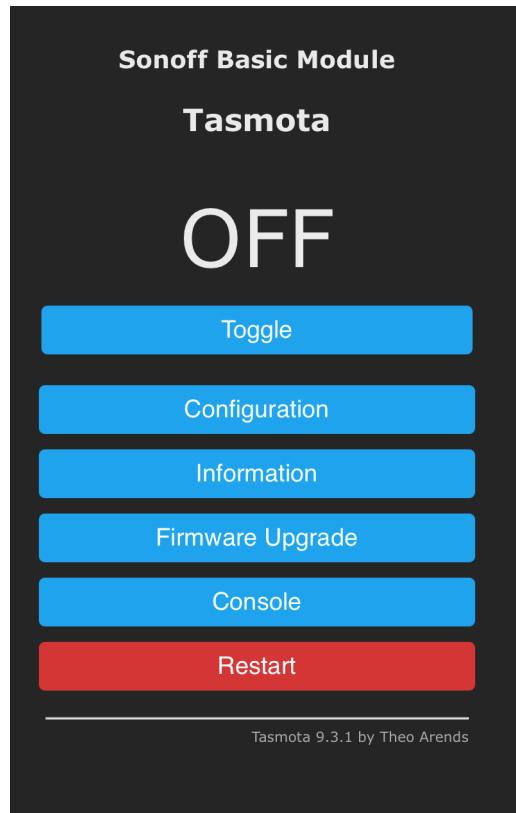
Nach dem Flashen der Tasmota Firmware und dem Anbringen der Netzkabel sind die Schalter bereit für die **Konfiguration**. Es sind umfangreiche Konfigurations-Möglichkeiten vorhanden. Für den Betrieb mit Switcher 3 sind jedoch nur wenige Einstellungen vorzunehmen:

- WLAN, eigene Credentials müssen eingestellt werden, damit sich der Schalter mit dem eigenen WLAN verbinden kann.
- Die MQTT Konfiguration muss vorgenommen werden, dh. die IP-Adresse des MQTT Brokers muss konfiguriert werden (Port 1883 ist schon voreingestellt). Zudem muss MQTT Username/PW eingestellt werden - falls der Broker dies verlangt (siehe Dokument Installation Switcher3).
- Topic muss eingestellt werden: dies muss mit vom Switcher3 publizierten Topics übereinstimmen.

### 6.1 Schritt 1 der Konfiguration

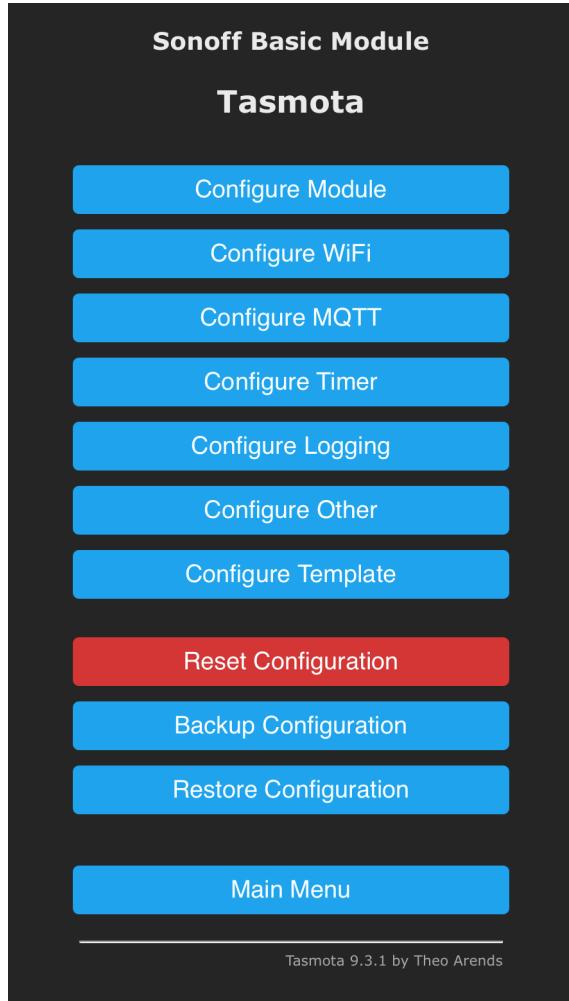
Zuerst sind die WLAN-Einstellungen anzupassen. Zu diesem Zweck hat der Smart Switch (Tasmota Software) die Möglichkeit, als **WiFi Hotspot** zu arbeiten. Ich verwende für diese Konfiguration mein iPad Tablett oder mein iPhone. Es ist so vorzugehen:

- Smart Switch korrekt mit Netzanschlüssen verkabeln und Netzstecker anbringen.
- Netzanschlüsse doppelt kontrollieren.
- Smart Switch am Netz (220 Volt) einstecken
- Pushbutton 6 mal kurz hintereinander drücken
- iPad oder iPhone mit dem entstehenden WLAN verbinden
- Wenn ok verbunden, im Browser **192.168.4.1** eingeben. Dann wird das Main-Konfigmenü ausgegeben. Das sieht so aus:



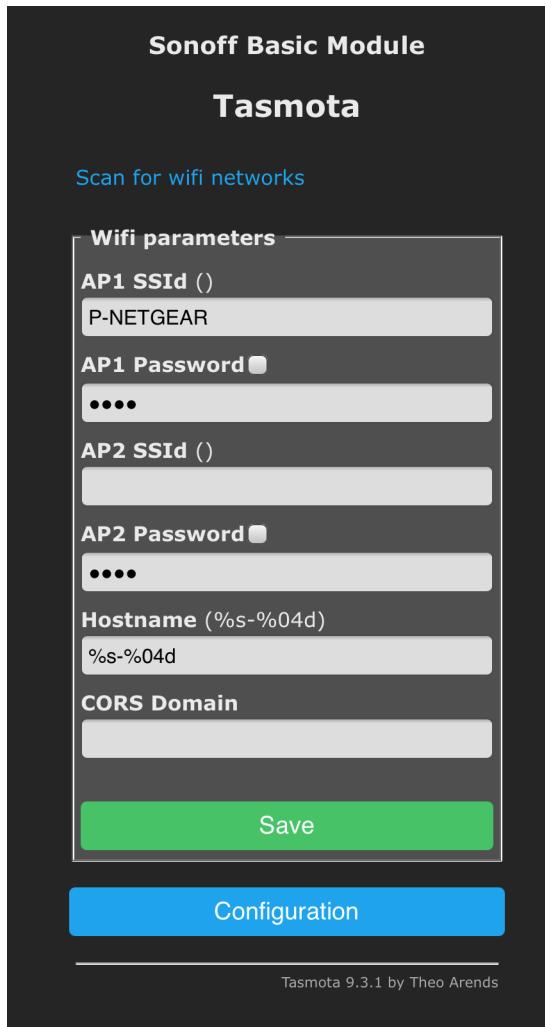
*Main Menu*

Wir brauchen jetzt den Menupunkt **Configuration**. Auch interessant sind **Information** und **Console**.



*Configurations Ansicht*

Hier ist es wichtig, zuerst die WiFi Konfig vorzunehmen, dies also anklicken und dann hier die WiFi Credentials eingeben:



*WiFi Konfig*

Hier mindestens die erste SSID und AP1 Password eingeben, es können auch Credentials eines zweiten WLAN's eingegeben werden, falls gewünscht/nötig. **Achtung:** auch wenn das eingegebene PW lang ist, werden nur 4 Punkte angezeigt - sehr verwirrend und man denkt, PW sei falsch...

Sorgfältig Eingabe prüfen und dann Save klicken - danach wird sich der Smart Switch als Hotspot verabschieden und macht einen Restart.

Aus Erfahrung ist es angezeigt, jetzt das Device kurz von 220 V zu trennen (für Restart - ist nicht immer nötig, aber kann helfen).

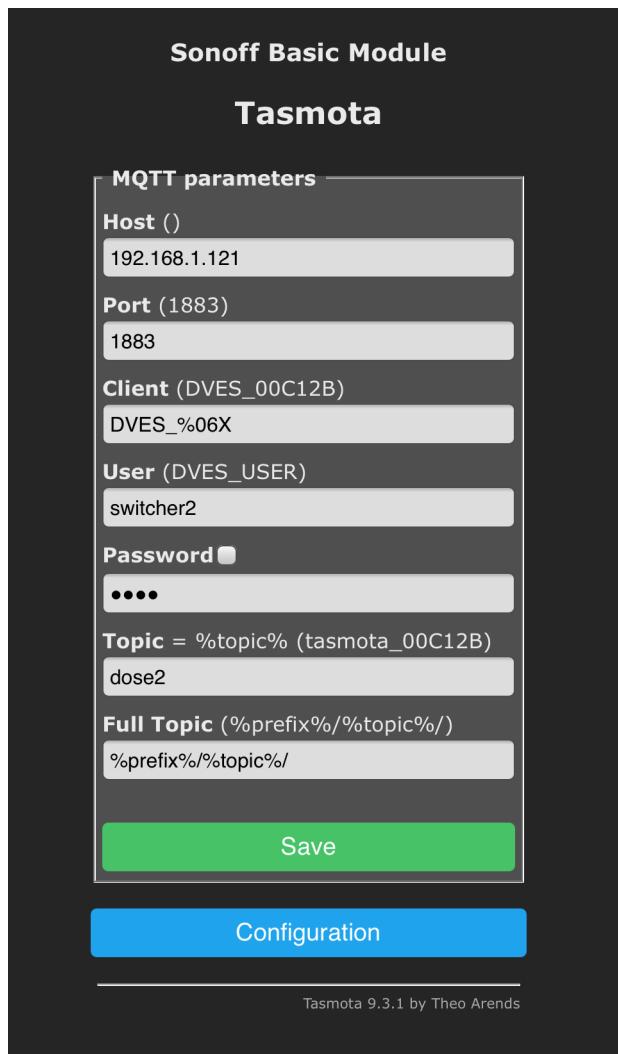
Wenn Credentials ok sind, wird er sich mit dem WLAN verbinden und er sollte bei einem Netzwerkscanner sichtbar sein. Dort kann man die zugeteilte IP-Adresse rauslesen. Wenn dies ok ist, wird im zweiten Schritt MQTT Konfig vorgenommen. Der Smart Switch kann in einem Netzwerkscanner durch die Hersteller-Info **Espressif** erkannt werden.

## 6.2 Schritt 2 der Konfiguration

Im **Browser** dann diese IP-Adresse eingeben und es wird dasselbe Main Menu erscheinen. Dann kann die MQTT Konfiguration vorgenommen werden Ein Blick in **Information** oder **Console** ist ebenfalls

aufschlussreich.

Dies ist die MQTT Konfiguration:

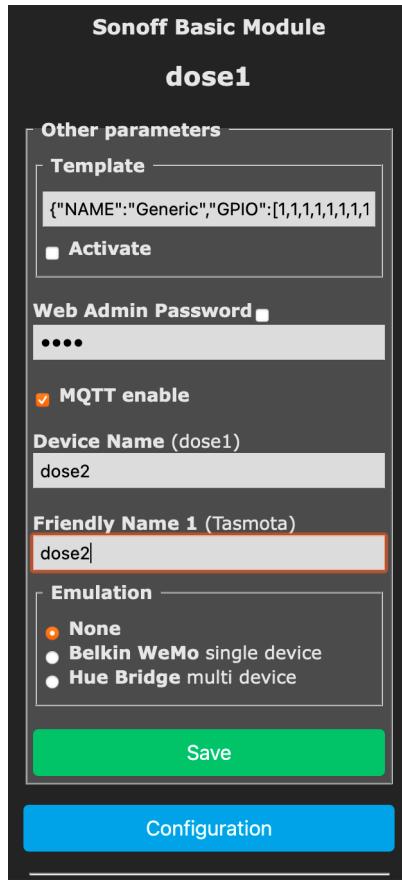


*MQTT Konfiguration*

- Zuerst die IP-Adresse des Computers eingeben, auf dem der MQTT Broker läuft (der Raspi). Port 1883 nicht ändern, das ist richtig.
- MQTT Username im Feld User auf **dagobert** setzen und Password auf **itscool** (**Achtung im Bild ist Username switcher2 - das ist alt**)
- Ganz wichtig ist das Feld Topic: hier die Dose eingeben, die dieser Smart Switch darstellen soll; eingeben: **dose1, dose2, dose3, dose4 oder dose5**
- Der Switcher wird MQTT Messages mit diesem Topic senden (Beispiel Dose 2): „**cmand/dose2/POWER**“, Payload ist dann **ON** oder **OFF**

## 6.3 Schritt 3 der Konfiguration

Zum Schluss ist der Name des Device noch anzugeben im Menu **Configure Other**. Damit hat das Device im Browser Fenster den richtigen Namen. Nach Eingabe Save klicken.



Menu Configure Other

Damit ist die Konfiguration eines Smart Switches abgeschlossen.

Die ganze Sache ist nicht immer einfach, mal funktioniert der Hotspot nicht korrekt, mal kommt keine Verbindung mit dem WLAN zustande. Oft musste ich mehrmals den Switch wieder aus- und einstecken (220 V).

## 7. Funktion Smart Switch

Der Tasmota Smart Switch kann auf zwei Arten geschaltet werden:

- Durch Senden einer MQTT Message, wie oben erklärt
- Durch Druck auf die Taste am Smart Switch selbst.

### Note:

Nach **jedem** Schalten publiziert der Smart Switch eine MQTT Message, mit welcher er den neuen Schaltzustand meldet. Topic ist dabei: „stat/dose2/POWER“, die Payload ist ON oder OFF.

Diese Meldung wird vom Switcher **empfangen** und der Schaltzustand geändert, auch auf dem WebGui.

## 8. Testen des Smart Switches

Es gibt verschiedene Möglichkeiten, das Funktionieren zu überprüfen (unabhängig und vor dem Einsatz mit Switcher3):

- Via http Aufruf, siehe Doku Tasmota
- Ich verwende auf dem iPad eine **iOS App** genannt **MQTT Inspector**. Damit können Publish und Subscribe gemacht werden und es können beliebige Topics/Payloads gesetzt werden. Damit ist es einfach, den Smart Switch zu testen.
- Für **Android** (Google PlayStore) ist es die App **MyMQTT**, die dafür gut geeignet ist.

Juni 2021, Peter K. Boxler