

Module mit Tasmota Firmware

Es gibt verschiedene schaltbare Geräte mit der Open Source Tasmota Firmware. Diese sind alle über einen Tasmota Aktor mit FabAccess kompatibel. Die Geräte sind in der Regel sehr kostengünstig, egal ob Zwischenstecker oder Mehrfachsteckdosenleiste - damit kann die Stromversorgung von Ressourcen sehr leicht eingeschalten werden. Durch die Tasmota Firmware können auch die Aktorenabindungen und das Geräteverhalten sehr flexibel angepasst werden.

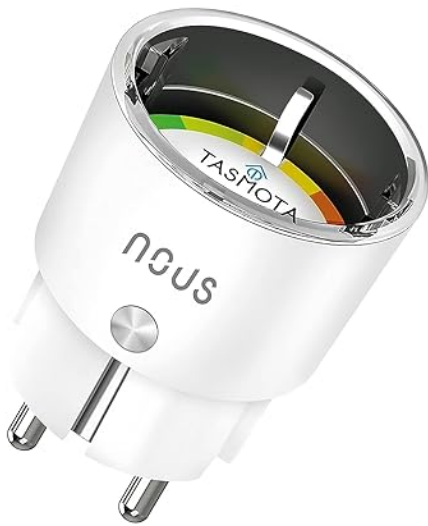
Beliebte Hardware

Besonders beliebte Tasmota-kompatible Hardwarekomponenten für das Schalten von Ressourcen sind z.B.

- Gosund EP2 (Schaltsteckdose Einzelstecker)
- Nous A1T (Schaltsteckdose Einzelstecker)
- Nous A5T (Mehrfachschaltsteckdosenleiste)
- Nous A6T (Schaltsteckdose Einzelstecker **IP44 für Outdoor-Einsatz**)
- Sonoff S26 R2 Plug (Schaltsteckdose Einzelstecker)

Nous A1T

Die im Vergleich zu den Gosund EP2 etwas bessere Steckdose, da sie mehr Leistung verträgt (3680 Watt)



Wichtig ist, dass es sich bei dem Nougat um die Tasmota-Variante handelt, da nur diese ohne externe Cloud funktioniert. Die Nougat A1T lassen sich vorgeflashed erwerben.

Gosund EP2

Der EP2 (2500 Watt) ist der Nachfolger des Gosund SP111 und ist fast baugleich. Durch seine ebenso wie die Nougat A1T kompakte Bauweise passt dieser Stecker in fast jede Ecke und auch lässt er sich recht einfach OTA-Flashen.



Wichtige Links

- <https://tasmota.github.io/docs/Firmware-Builds>
- https://gitlab.com/fabinfra/fabaccess/grafana/-/tree/main?ref_type=heads
- <https://tasmota.github.io/docs/Commands>
- Folgende Links können verwendet werden, um kompatible Hardware aufzuspüren:
 - <https://templates.blakadder.com/search.html>
 - <https://tasmota.github.io/docs/Supported-Modules>

Tasmota Schaltsteckdose ins Netzwerk einbinden - HowTo

Diese Anleitung basiert auf <https://nous.technology/product/a1t/de.html>

Schaltsteckdose einstecken und Web Interface öffnen

1. Die Dose öffnet kurze Zeit nach Einstecken einen eigenen Access Point. Wir verbinden uns mit dieser SSID und rufen das Web Interface <http://192.168.4.1> (HTTPS nicht von Haus aus unterstützt!) vom Nous auf. In der Regel ist die IP-Adresse folglich `192.168.4.1`. Siehe auch <https://nous.technology/product/a1t/de.html>

Mit Wifi SSID verbinden

Wir suchen unsere lokale Werkstatt-SSID aus und verbinden das Modul:

freifunk-erlangen.de#ZAM



PRIMAS-Gast



PRIMAS-WLAN



WestlicheStadtmauer



Scan for all WiFi Networks

Wifi parameters

WiFi Network ()

fabinfra101_2400

WiFi Password ☐

....

WiFi Network 2 ()

Type your Alternative WiFi Network

WiFi Password ☐

....

Hostname (%s-%04d)

%s-%04d

Save

Configuration

Tasmota 12.5.0 by Theo Arends

Tasmota

Successful WiFi Connection

Redirecting to new device's IP address

192.168.188.39

Tasmota 12.5.0 by Theo Arends

Nach der Kopplung findet sich die Schaltsteckdose nun im Netzwerk eingebunden wieder. Wir verwenden die im Beispiel angegebene IP-Adresse `192.168.188.39` und rufen erneut das Web Interface auf: <http://192.168.188.39>. Damit sind wir in der Lage alle weiteren Einstellungen vorzunehmen.

Tipp: Wir empfehlen das Gerät nach dem Einbinden sinnvoll in der Netzwerkübersicht des Routers/Access Points zu benennen.

Modul konfigurieren

Folgende Configs sind - ausgehend von den Werkseinstellungen - je Steckdose zu ändern. Alle hier nicht angegebenen Einstellungen können beim Standard belassen werden.

Configuration → Configure MQTT

- Host (Die IP-Adresse oder Hostname des FabAccess Servers)
- Port (Standard 1883)
- Benutzername
- Passwort

Configuration → Configure Other

- Passwort für den Web Admin setzen (Der Nutzernamen ist `admin` und kann nicht verändert werden)
- Device Name (für die eigene Übersicht)
- Friendly Name 1 (für die eigene Übersicht)
- Topic: entweder statisch vergeben (z.B. `tasmota_1` oder die Standards nutzen. Die ID ist sehr wichtig, da wir genau diese für die FabAccess Actor Konfiguration in `bffh.dhall` benötigen, allerdings dort ohne führendes `tasmota_`). Der Standardwert ist `tasmota_%06X`.
- HTTP API **deaktivieren**

Spezifische Kommandos in der Konsole

- PowerOnState anpassen.
 - Die Steckdose kann so konfiguriert werden, dass sie immer den letzten gespeicherten Zustand einnimmt (d.h. wenn die Dose vorher aus war, wird sie beim nächsten Einstecken auch aus sein oder an sein, wenn sie vorher an war). Das Setzen erfolgt in diesem Fall über die Console durch Ausführen von `PowerOnState 3` (Bestätigen mit Enter).

- falls die Dose einen **Wiederanlaufschutz** simulieren soll, empfiehlt sich `PowerOnState 0` - die Dose ist dann zunächst immer aus, wenn sie eingesteckt wird. So kann keine Maschine aus Versehen los laufen
- Es gibt weitere Modi (0, 1, 2, 3, 4, 5) - siehe Dokumentation.
- Werkseinstellungen durch 7x hintereinander die Steckdose schnell ein- und wieder ausstecken. Diese Option ist praktisch, aber bietet minderen Manipulationsschutz. Das kann mit `SetOption65` per Console unterbunden werden: `SetOption65 1` (Bestätigen mit Enter)
- Den Button an der Steckdose deaktivieren, damit er nicht per Hand gedrückt werden kann. Standardmäßig lässt sich die Nous A1T einfach umschalten per Druck. Die `SetOption73` verhindert das. Das Setzen erfolgt durch Ausführen von `SetOption73 1` (Bestätigen mit Enter)
- Zeitzone setzen. So erhalten wir stichhaltige Informationen über MQTT. Hierzu gibt es ein praktisches Onlinewerkzeug: <https://tasmotatimezone.com>. Für die angegebene Location und Zeitzone erhalten wir den passenden Konfigurationsbefehl für Tasmota. Beispielbefehl: `Backlog Latitude 50.831742; Longitude 12.94054565505144; TimeDST 0,0,3,1,1,120; TimeSTD 0,0,10,1,1,60; TimeZone 99`. Hinweis: Der Zeitserver (NTP) ist automatisch vorkonfiguriert (Befehl `NtpServer`) und muss nicht angefasst werden.

Kalibrierung der Stromwerte (Netzspannung) für korrektes Monitoring/Reporting

- als kurzes Tutorial in Englisch: <https://tasmota.github.io/docs/Power-Monitoring-Calibration/#setup>
- als 2-minütiges Praxisvideo auf Youtube: <https://www.youtube.com/embed/9M2G2EzEXAk>

TLS-Unterstützung für sichere Kommunikation

- Firmware Upgrade / Flash custom TLS enabled Firmware. Diese erlaubt die (fast) automatische Nutzung des Lets Encrypt Root Zertifikats ohne sämtliche weitere Konfiguration auf dem Modul. Lediglich mosquitto und BFFH müssen Lets Encrypt verwenden

MQTT parameters

Host ()
192.168.188.34

Port (1883)
1883

Client (DVES_F09377)
DVES_%06X

User (DVES_USER)
fabinfra101

Password ☐
....

Topic = %topic% (tasmota_F09377)
tasmota_%06X

Full Topic (%prefix%/ %topic%/)
%prefix%/ %topic%/

Save

Configuration

Tasmota 12.5.0 by Theo Arends

```
{"NAME": " NOUS A1T", "GPIO": [32,0,0,0,;
```

☒ Activate

Web Admin Password ☐
....

☐ HTTP API enable
☒ MQTT enable

Device Name (tasmota_1)
tasmota_1

Friendly Name 1 (Tasmota)
tasmota_1

Emulation
☒ None
☐ Belkin WeMo single device
☐ Hue Bridge multi device

Save

Configuration

Tasmota 12.5.0 by Theo Arends

Die Screenshots zeigen ein paar der obigen Einstellungen

Bekannte Probleme

Werkseinstellungen als Manipulationsmethode

Nutzer können die Admins ärgern!

Nutzer können FabAccess umgehen, wenn sie die Steckdosen auf Werkseinstellung zurücksetzen und die Tasmota Schaltsteckdose umkonfigurieren. Um die Smart-Steckdose auf die Werkseinstellungen zurückzusetzen, benötigen Sie: Stecken Sie das Gerät 6 mal schnell hintereinander ein und aus und lassen Sie es das siebte Mal eingeschaltet. Die LED sollte zu blinken beginnen. Dies bedeutet, dass die Steckdose wieder angeschlossen werden kann. Per `SetOption65` kann dieses Verhalten geändert werden (siehe oben). Allerdings bewirkt diese Option auch, dass im Falle fehlender Verbindung keine Konfiguration der Dosen mehr erfolgen kann. Der einzige Weg ist dann das Zurücksetzen per OTA-Flash.

Schaltsteckdose schaltet per MQTT-Befehl nicht oder blinkt

Die Nous Steckdose blinkt, aber sie ist im Netzwerk erreichbar (nicht im Access Point Modus)!

Wenn die Schaltsteckdose blinkt, sich aber nicht bedienen lässt: Nachprüfen, ob der MQTT Server für Geräte erreichbar ist. Die Nous Steckdosen blinken in der Regel dann, wenn keine Verbindung zum Server besteht oder weil ggf. die IP-Adresse falsch konfiguriert ist. In diesem Fall die Tasmota Settings und die des Mosquitto Servers prüfen. Übrigens blinken die Schaltsteckdosen bei fehlender MQTT-Verbindung nicht, wenn sie eingeschaltet sind - dann leuchtet die LED durchgängig grün.

<https://videos.stadtfabrikanten.org/videos/embed/2023f007-a544-4f05-82c6-d437ffb6c17b>

Tasmota Schaltsteckdose auf MQTT-Funktionalität testen

Nach dem Einbinden und Konfigurieren unserer Schaltsteckdose in unser Netzwerk können wir sie von einem beliebigen Linux Client überprüfen, ohne dabei FabAccess anzufassen. Somit schließen wir von Anfang an Probleme aus. Wir nutzen dazu das Kommando `mosquitto_pub`. Für das Schalten sprechen wir nicht direkt die Nous A1T an (denn wir haben der Nous Dose die Server-Informationen bereits mitgegeben), sondern

den MQTT Server mit seiner IP-Adresse und Begleitinformationen Port, Benutzer und Passwort. Dabei übergeben wir mit `-t` außerdem das Topic und mit `-m` den Wert. In folgenden Beispiel schalten wir die Dose zunächst aus und dann an. Das Topic setzt sich aus dem Präfix `cmnd`, dem Gerätename `tasmota_1` und dem Kommando `POWER` zusammen.

```
#mosquitto_pub verfügbar machen, falls nicht aufrufbar  
sudo apt install mosquitto-clients
```

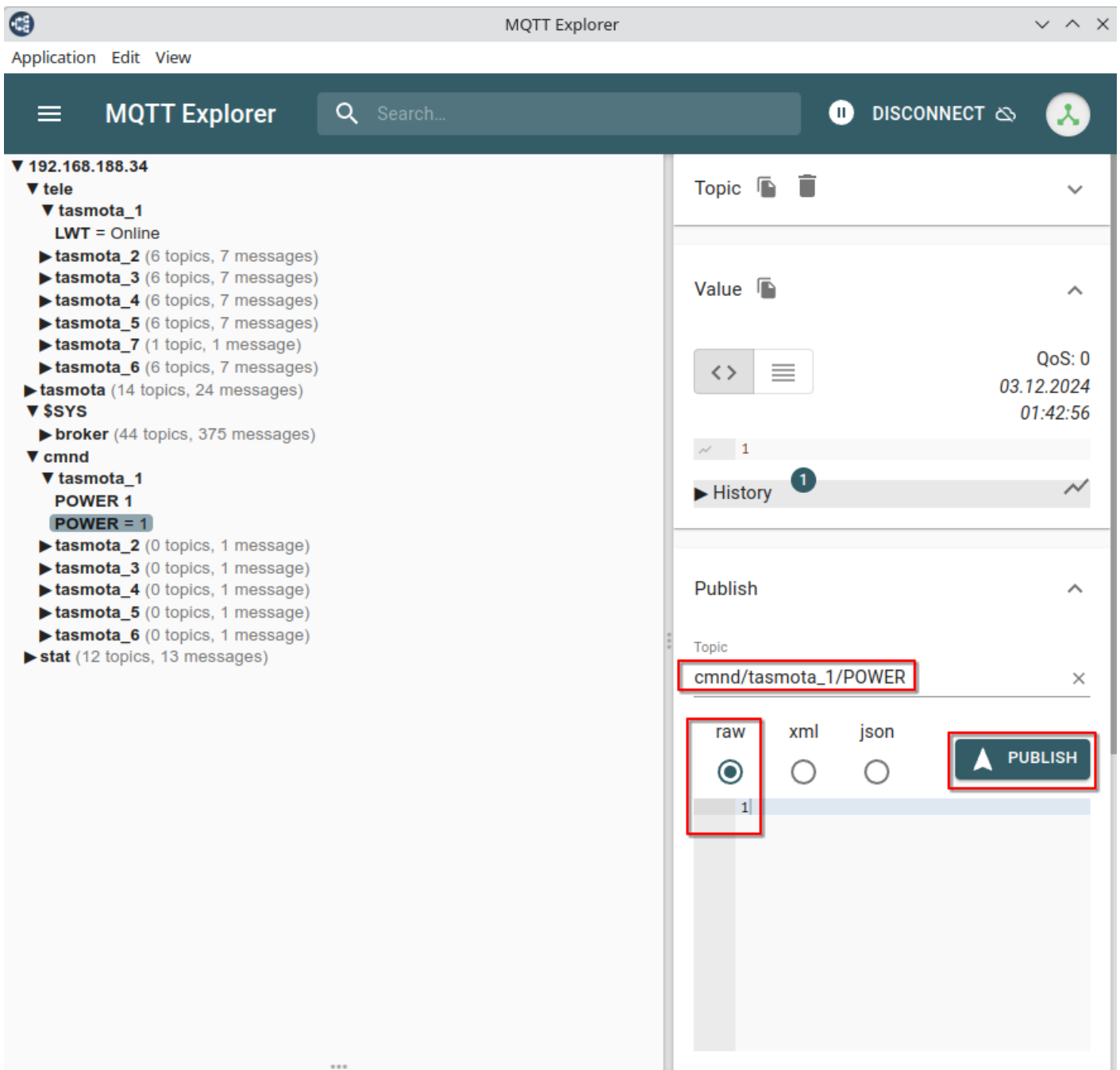
Ausschalten:

```
mosquitto_pub -d -h 192.168.188.34 -u fabinfra101 -P fablocal -p 1883 -t "cmnd/tasmota_1/POWER" -m '0'
```

Einschalten:

```
mosquitto_pub -d -h 192.168.188.34 -u fabinfra101 -P fablocal -p 1883 -t "cmnd/tasmota_1/POWER" -m '1'
```

Wer nicht auf Kommandozeile operieren will oder etwas Debug-Übersicht benötigt, der kann den MQTT Explorer verwenden. Im Feld `Topic` kann der obige Befehl eingegeben werden. der Wert kann als `raw` Wert eingetippt werden. Dann klicken wir auf `PUBLISH` und führen die Aktion aus.



Tasmota Actor für BFFH installieren

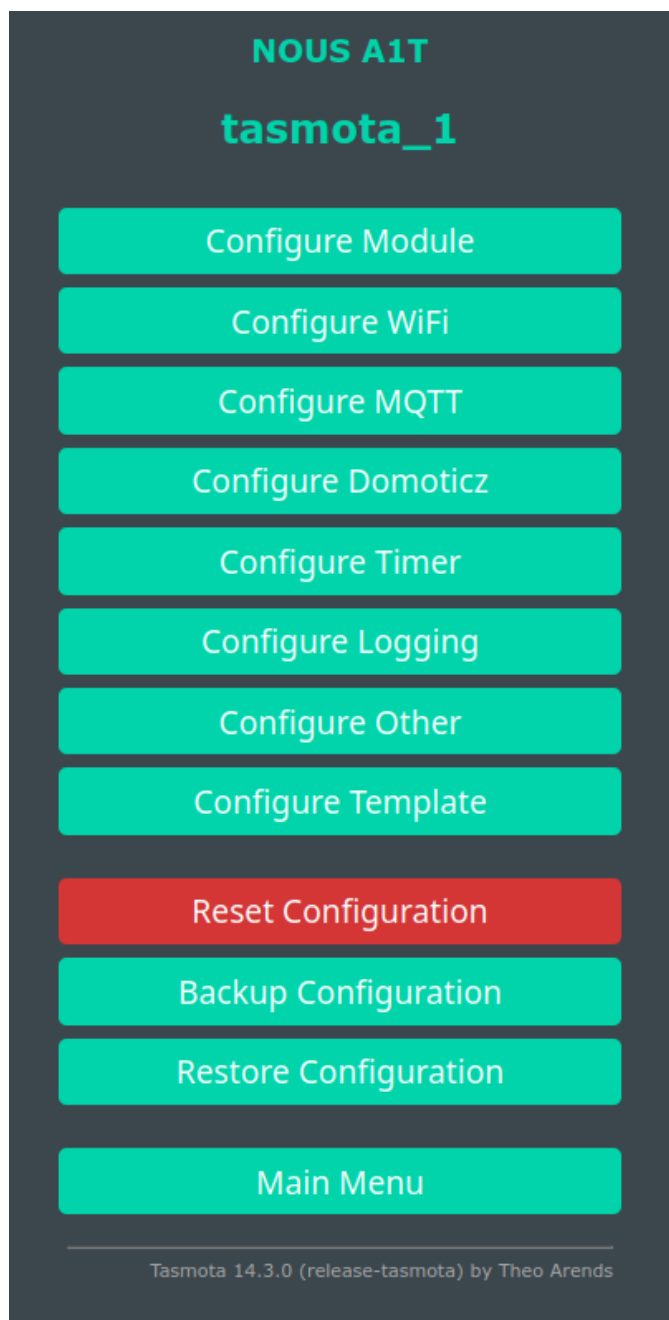
Diese Anleitung findest du unter [Aktor: Tasmota](#).

Tasmota Web UI in FabAccess-Farben

Zum "hübsch" machen können wir alle Farben in Tasmota ändern. Das hat einerseits eine kosmetische Wirkung, andererseits eine warnende bzw. informierende! Sobald wir sehen,

dass unsere Nous Steckdose andere Farben als der Standard hat wissen wir, dass wir sie schonmal konfiguriert haben und dass sie zu unserem FabAccess Setup gehört. In Umgebungen, wo vielleicht noch andere Geräte im Netzwerk ihr (Un)wesen treiben, hilft das:

```
mosquitto_pub -d -h MQTT_SERVER -p MQTT_SERVER_PORT -u MQTT_USER -P MQTT_PASSWORD -t  
"cmnd/tasmota_1/WebColor" -m  
'{"WebColor":["#00d4aa","#3c474d","#3c474d","#000000","#dddddd","#00d4aa","#3c474d","#ff5661","#008  
000","#faffff","#00d4aa","#009275","#d43535","#931f1f","#47c266","#5aaf6f","#faffff","#999999","#00d4aa  
"]}'
```



Screenshot: Das Web Interface umgefärbt

Der Standard kann wie folgt zurückgesetzt werden:

```
mosquitto_pub -d -h MQTT_SERVER -p MQTT_SERVER_PORT -u MQTT_USER -P MQTT_PASSWORD -t
"cmd/tasmota_1/WebColor" -m
'{"WebColor":["#eaeaea","#252525","#4f4f4f","#000000","#dddddd","#65c115","#1f1f1f","#ff5661","#00800
0","#faffff","#1fa3ec","#0e70a4","#d43535","#931f1f","#47c266","#5aaf6f","#faffff","#999999","#eaeaea","#
08405e"]}]'
```

Die exakte Doku der Farben findet sich in

https://github.com/arendst/Tasmota/blob/development/tasmota/my_user_config.h

Erweitertes Setup - Custom Tasmota Firmware mit TLS-Support

Neben der unverschlüsselten Standardvariante mit MQTT lässt sich auch eine sichere MQTTS-Verbindung herstellen, sofern Tasmota entsprechend dafür ausgestattet ist. Hierzu ist das Kompilieren der Firmware notwendig, weil TLS aus Platzgründen standardmäßig nicht eingebaut ist. Wir beziehen uns auf <https://tasmota.github.io/docs/Create-your-own-Firmware-Build-without-IDE/#build-the-firmware>.

Tasmota unterstützt von Haus aus das [Let's Encrypt ISRG Root X1](#) Zertifikat, welches bis zum 04.06.2030 gültig ist. Siehe <https://tasmota.github.io/docs/TLS>.

Umgebung aufsetzen

```
dnf install python python-virtualenv
pip install --upgrade pip

cd /home/tomate/FabInfra

virtualenv platformio-core
cd platformio-core
```

```
. bin/activate
pip install -U platformio
pip install --upgrade pip
```

```
cd /home/tomate/FabInfra
git clone https://github.com/arendst/Tasmota.git
```

Anpassungen vornehmen

```
vim platform.ini
```

```
; uncomment the following to enable TLS with 4096 RSA certificates
-DUSE_4K_RSA
```

```
#ganz oben
lib_extra_dirs      =
                    ${common.lib_extra_dirs}
                    lib/lib_ssl
```

```
vim tasmota/user_config_override.h
```

```
#ifndef _USER_CONFIG_OVERRIDE_H_
#define _USER_CONFIG_OVERRIDE_H_

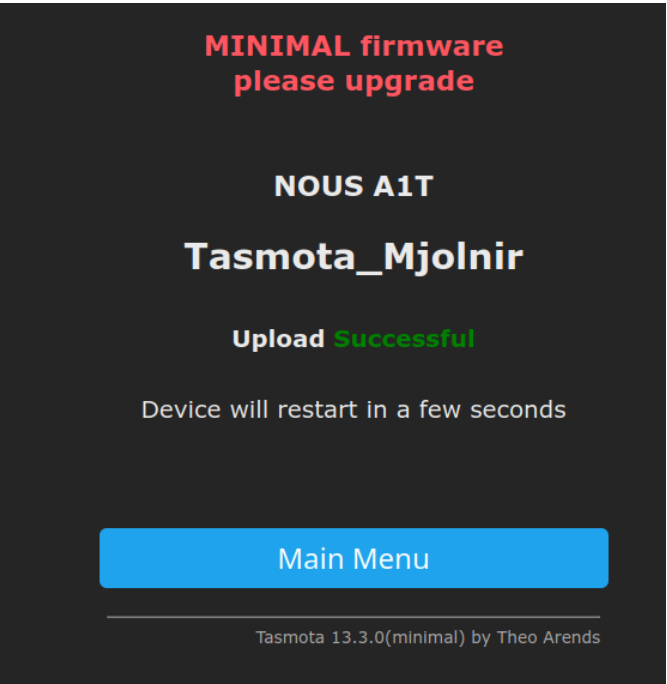
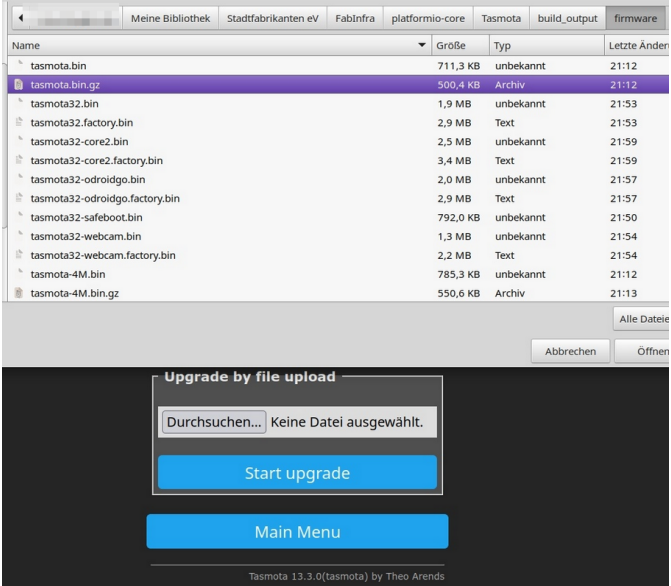
#ifndef USE_MQTT_TLS
#define USE_MQTT_TLS           // Use TLS for MQTT connection (+34.5k code, +7.0k mem and +4.8k
additional during connection handshake)
#define MQTT_TLS_ENABLED      true           // [SetOption103] Enable TLS mode (requires TLS version)
#define USE_MQTT_TLS_CA_CERT   // Force full CA validation instead of fingerprints, slower, but
simpler to use. (+2.2k code, +1.9k mem during connection handshake)
                                // This includes the LetsEncrypt CA in tasmota_ca.ino for verifying server
certificates
// #define USE_MQTT_TLS_FORCE_EC_CIPHER      // Force Elliptic Curve cipher (higher security) required by
some servers (automatically enabled with USE_MQTT_AWS_IOT) (+11.4k code, +0.4k mem)
#endif

#endif // _USER_CONFIG_OVERRIDE_H_
```

Flashen



om/tasmota/tasmota-minimal.bin.gz
tasmota.bin.gz



TLS-Verschlüsselung verwenden

Standard ist:

- Host: 192.168.1.192
- Port: 1883

Angepasst ist:

- Host: fabaccess.fablabchemnitz.de
- Port: 8883

The screenshot shows the 'MQTT parameters' configuration screen for 'Tasmota_Mjolnir' on a 'NOUS A1T' device. The interface is dark-themed with white and green text. The 'MQTT parameters' section is highlighted with a green border. It contains several input fields and a checkbox. The 'Host' field is set to 'fabaccess.fablabchemnitz.de'. The 'Port' field is set to '8883'. The 'MQTT TLS' checkbox is checked. The 'Client' field is set to 'DVES_F0AC9D'. The 'User' field is set to 'fablab'. The 'Password' field is masked with dots. The 'Topic' field is set to 'tasmota_%06X'. The 'Full Topic' field is set to '%prefix%/%topic%/'. A green 'Save' button is at the bottom of the configuration section. Below the configuration section is a blue 'Configuration' button. At the very bottom, the text 'Tasmota 13.3.0.2(tasmota) by Theo Arends' is displayed.

NOUS A1T

Tasmota_Mjolnir

MQTT parameters

Host ()
fabaccess.fablabchemnitz.de

Port (1883)
8883

☒ **MQTT TLS**

Client (DVES_F0AC9D)
DVES_%06X

User (DVES_USER)
fablab

Password ☐
....

Topic = %topic% (tasmota_F0AC9D)
tasmota_%06X

Full Topic (%prefix%/%topic%/)
%prefix%/%topic%/

Save

Configuration

Tasmota 13.3.0.2(tasmota) by Theo Arends

Prüfen, ob der DNS-Eintrag klappt und der Port offen ist. Sonst kann keine MQTTS Verbindung aufgebaut werden:

Der DNS-Eintrag ist aktuell im public DNS eingetragen (neycerha) und zeigt auf 192.168.1.192 → das funktioniert nicht, weil wir das DNS der Fritzbox nicht überschreiben können

Für ein LAN wird ein eigener DNS-Server gebraucht, z.B. Unifi DreamMachine oder ein pi-hole.

```
dig fabaccess.fablabchemnitz.de +short
dig @ns2.fablabchemnitz.de fabaccess.fablabchemnitz.de +short
dig @8.8.8.8 fabaccess.fablabchemnitz.de +short

#private DNS Server zuhause
dig @76.76.2.2 fabaccess.fablabchemnitz.de +short
dig @76.76.10.2 fabaccess.fablabchemnitz.de +short
dig @2606:1a40::2 fabaccess.fablabchemnitz.de +short
dig @2606:1a40:1::2 fabaccess.fablabchemnitz.de +short
dig @192.168.1.22 -> empty. Warum? Weil die FritzBox mit einem eigenen Eintrag bereits 192.168.1.192 inne hält und sich nicht überschreiben lässt #fritzbox

telnet fabaccess.fablabchemnitz.de 8883
```

In Tasmota Console:

<https://tasmota.github.io/docs/Commands/#setoptions>

```
#setzen
SetOption103 1
SetOption132 0

#output prüfen:
SetOption103
SetOption132

#DNS Server 1 "IPAddress4" verändern (testweise)
IPAddress4 192.168.1.22 #default
IPAddress4 8.8.8.8 restart 1

#DNS Server 2 "IPAddress5" verändern (testweise)
IPAddress5 0.0.0.0 #default
IPAddress5 8.8.4.4
```

```
IPAddress5 45.136.31.74 restart 1
```

```
#mDNS enablen - nur testweise
```

```
SetOption55 1
```

Deprecated Fingerprint Methode

Fingerprint erzeugen: <https://github.com/issacg/tasmota-fingerprint/releases>

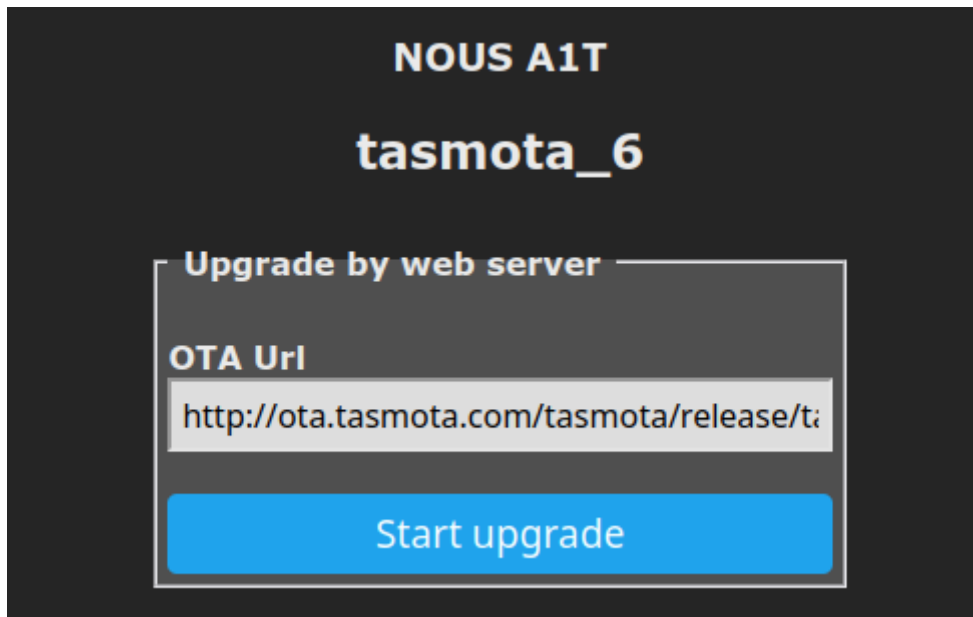
```
cd /opt/  
wget https://github.com/issacg/tasmota-fingerprint/releases/download/v1.0.0/tasmota-  
fingerprint_1.0.0_Linux_armv7.gz  
gunzip tasmota-fingerprint_1.0.0_Linux_armv7.gz  
chmod +x tasmota-fingerprint_1.0.0_Linux_armv7  
./tasmota-fingerprint_1.0.0_Linux_armv7 /etc/ssl/certs/ISRG_Root_X1.pem
```

TLS Fingerprint eintragen und TLS aktivieren (In Tasmota Console):

```
#setzen  
SetOption103 1  
SetOption132 1  
MqttFingerprint F4 EA FC 42 1A 8B 2D 2D 2E 1F 65 21 58 BF D7 3B 35 3F 90 4E  
  
#output prüfen:  
SetOption103  
SetOption132  
MqttFingerprint  
  
#reset Fingerprint:  
MqttFingerprint 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
  
#allow all (Warnung: nicht empfohlen):  
MqttFingerprint FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

Firmware Upgrade

Das Upgrade kann direkt vom Web Interface aus erledigt werden und funktioniert grundlegend einfach. Das Update dauert ca. 3-5 Minuten und wird bei zwischenzeitlicher Seitenaktualisierung u.U. verschiedene Fehler anzeigen. Diese sind jedoch normal. Zuletzt getestet am 21.11.2024 mit Nous A1T mit Tasmota 12.5.0 (minimal) auf 14.3.0 (minimal).



Weitere Infos finden sich unter <https://tasmota.github.io/docs/Upgrading/#decode-config-tool>

3D-Druck Einhausung vom MakerSpace Gütersloh

gitlab.com/igami/nous-a1-safebox

Ein kurzes Nous A1T Erklärvideo

Falko Richter von 35 Services e.V. Berlin erklärt grundlegende Schritte und demonstriert, wie man Nous A1T Schaltsteckdosen mit Tasmota Firmware verwendet.

<https://www.youtube.com/embed/S6n5mJKozBU>

Rules (Regeln)

Mit Regeln können wir angepasstes Verhalten ermöglichen und zusätzlichen Komfort- bzw. Logikgewinn erzielen. Hier finden sich ein paar praktische Regeln für Copy-Paste. Im Anschluss jeder Konfiguration sollte das Script nochmal ausgiebig getestet werden!

Angelegte Regeln können auch wieder gelöscht werden:

```
rule1 "
```

Die Anzeige der aktuell vergebenen Regel ist möglich per:

```
rule1 ?
```

Nachlaufsteuerung für Geräte (60 Sekunden)

Diese Regel wurde von Joris Bijkerk beigesteuert und erlaubt, dass eine Gerätschaft nach dem Ausschalten noch 60 Sekunden nachläuft. Die Zeit `RuleTimer1` kann beliebig angepasst werden. Die folgenden Kommandos fügen wir in der Console ein und führen sie aus.

Diese Regel funktioniert nicht für Schaltsteckdosen, weil beim Verändern des Power-Status kein Event#Switch1 getriggert wird.

```
Backlog SwitchMode1 1
```

```
Rule1
ON Event#Switch1#State=1 DO
    Backlog Power1 ON; RuleTimer1 0
ENDON
ON Event#Switch1#State=0 DO
    RuleTimer1 60
ENDON
ON Rules#Timer=1 DO
    Power1 0
ENDON
```

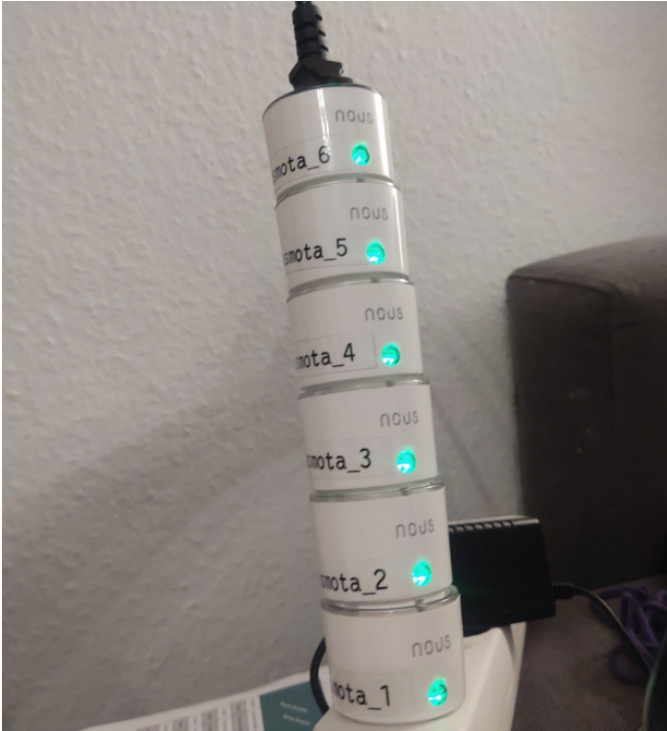
```
Rule1 1
```

Do's und Dont's

Ein paar Hinweise zur Elektrik.

Nicht Stapeln

Das Stapeln mehrerer Nous Schaltsteckdosen funktioniert in der Praxis, aber ergibt technisch wenig Sinn und ist elektrisch bedenklich, zumal alle Dosen aus gehen, die hinter der jeweils vorigen Schaltsteckdose klemmen.



Keine Verteilersteckdosenleisten anstecken

An die Schaltsteckdosen sollten im Idealfall keine Verteilerleisten angeschlossen werden. Eine Nous A1T kann bis zu 3680 Watt abgeben. Die sind u.U. schnell erreicht. Konzeptionell gesehen ist eine Schaltsteckdose diesen Typs für genau einen Verbraucher ausgelegt. Wenn Du mit einer solchen Schaltsteckdose mehrere Geräte zusammen schalten möchtest, dann empfehlen wir dir einen anderen Typ von Schaltsteckdose - zum Beispiel die Nous A5T. Diese kann bis zu drei Schuko-Geräte aufnehmen und hat zusätzlich USB-Ports.

Version #65

Erstellt: 21 Oktober 2024 14:32:28 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 11 März 2025 20:43:45 von Mario Voigt (Stadtfabrikanten e.V.)