

Aktor: Tasmota

FabAccess bietet über Aktoren-Schnittstellen die passenden Möglichkeiten, um per Python Script eine entsprechende Verbindung zu Tasmota-basierten Geräten aufzubauen.:

<https://gitlab.com/fabinfra/fabaccess/actors/tasmota>

Zum Koppeln der Schaltsteckdose mit FabAccess wird einerseits die Wifi-Verbindung zwischen Steckdose und Netzwerk benötigt, andererseits auch eine Datenverbindung per MQTT-Protokoll.

Installation

```
cd /opt/fabinfra/  
git clone https://gitlab.com/fabinfra/fabaccess/actors/tasmota.git adapters/tasmota  
chmod +x adapters/tasmota/main.py  
chown -R bffh:bffh /opt/fabinfra/adapters/tasmota  
  
python3 -m venv env  
. env/bin/activate #activate venv  
pip install -r requirements.txt
```

Achtung: in main.py sind einige Angabe statisch. Das Topic "tasmota_" wird vorrangestellt, sodass in die bffh Konfiguration nur noch die ID eingetragen werden muss. Hier im Beispiel "F0AC9D"

MQTT Host	192.168.1.192
MQTT Port	1883
MQTT User	fablab
MQTT Client	DVES_F0AC9D
MQTT Topic	tasmota_%06X
MQTT Group Topic 1	cmnd/tasmtas/
MQTT Full Topic	cmnd/ <u>tasmota</u> .F0AC9D/
MQTT Fallback Topic	cmnd/DVES_F0AC9D_fb/
MQTT No Retain	Disabled

Testen

Das Script kann manuell (unabhängig von bffh) getestet werden, um auszuschließen, dass es Probleme mit dem Server an sich gibt:

Grundsätzlicher Syntax:

```
/opt/fabinfra/adapters/tasmota/env/bin/python3 /opt/fabinfra/adapters/tasmota/main.py
```

usage: main.py [-h] --host HOST [--port [PORT]] [--user USER] [--password PASSWORD] --tasmota TASMOTA
name {free,inuse,tocheck,blocked,disabled,reserved,raw} ...

main.py: error: the following arguments are required: --host, --tasmota, name, state

#--user USER = Nutzer des MQTT Servers

#--password PASSWORD = Passwort des MQTT Servers

#TASMOTA name = Device Name, aber ohne führendes "tasmota_"

#userid = FabAccess Nutzer (users.toml)

#Gerät "tasmota_1" als "admin" user nutzen (aktivieren)

```
/opt/fabinfra/adapters/tasmota/env/bin/python3 /opt/fabinfra/adapters/tasmota/main.py --host localhost --user  
fabinfra101 --password fablocal --tasmota 1 state inuse Admin
```

#Gerät "tasmota_1" wieder freigeben (ausschalten)

```
/opt/fabinfra/adapters/tasmota/env/bin/python3 /opt/fabinfra/adapters/tasmota/main.py --host localhost --user  
fabinfra101 --password fablocal --tasmota 1 state free
```

In FabAccess einbinden

bffh.dhall Snippet

```
YOUR_ACTOR_ID =  
{  
  module = "Process",  
  params =  
  {  
    cmd = "/usr/bin/python3",  
    args = "/opt/fabinfra/adapters/tasmota/main.py --host 127.0.0.1 --user MQTT_USER --password  
MQTT_PASSWORD --tasmota YOUR_ACTOR_ID",  
  }  
},
```

FabAccess Config Generator Snippet

```
vim /opt/fabinfra/fabaccess-config-generator/actors.ini
```

```
[tasmota]
```

```
module = Process
```

```
param_cmd = "/opt/fabinfra/adapters/tasmota/env/bin/python3"
```

```
param_args = "/opt/fabinfra/adapters/tasmota/main.py --host 127.0.0.1 --user MQTT_USER --password  
MQTT_PASSWORD --tasmota $actor_id"
```

Version #3

Erstellt: 8 Dezember 2024 01:28:42 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 8 Dezember 2024 01:37:15 von Mario Voigt (Stadtfabrikanten e.V.)