Aktor: FabLock

Der FabLock Actor wird verwendet, um Türschlösser über unseren Client oder die API (z.B. pyfabapi) zu bedienen. Er basiert prinzipiell auf dem Template Aktor: Generisches Python-Template für "Process". Der Aktor unterstützt darüber hinaus jedoch noch die dafür notwendigen Spezialargumente, die über den Spezialzustand `raw` (binäre Daten) übermittelt werden können (siehe hier) und welche in der Dhall-Hauptkonfiguration über das spezielle Trait Prodable definiert werden. Dafür gibt es im Borepin Client die Aktionsknöpfe `UNLOCK` und `IDENTIFY`. Diese senden passende Argumente an den Prozess:

- `"action: unlock"`
- `"action: identify"`

> Die Hard- und Software unterstützt nur nicht-permanente Schlösser.

Zur Kommunikation mit der FabLock-Hardware wird MQTT verwendet.

Jedes physische FabLock-Modul muss dafür eine eindeutige ID haben. Die ID ist 5 Ziffern lang und hat führende Nullen (`--fablock` Parameter), also z.B. `00001`.
Um die einzelnen Schlösser im FabLock-Modul zu unterscheiden, hat jedes eine 5-stellige ID mit führenden Nullen (`--lock` Parameter), also z.B. `00091`.

Da verschiedene Schlösser unterschiedliche Auslösezeiten haben und diese eingehalten werden müssen, um die Funktionalität des Schlosses zu erhalten, können diese nur an der Hardware eingestellt werden. Die LED blinkt alle 500ms Sekunden, wenn sie identifiziert wird. Und leuchtet dauerhaft, wenn das Schloss entriegelt ist. Der Zustand des Riegels wird alle 30 Sekunden gemeldet.

Quellcode für den Aktor: https://gitlab.com/fabinfra/fabaccess/actors/fablock

Installation

```
mkdir -p /opt/fabinfra/adapters/
cd /opt/fabinfra/adapters/
git clone https://gitlab.com/fabinfra/fabaccess/actors/fablock.git
chmod +x fablock/main.py
```

```
chown -R bffh:bffh /opt/fabinfra/adapters/fablock/

cd /opt/fabinfra/adapters/fablock/
python3 -m venv env
. env/bin/activate #activate venv
pip install -r requirements.txt
```

Konfigurationsparameter

- `--host` MQTT Server Addresse
- `--port` MQTT Server Port
- `--user` MQTT User (optional)
- `--password` MQTT Passwort (optional)
- `--fablock` FabLock ID
- `--lock` Lock ID

MQTT Befehle

Das allgemeine Schema lautet: `/fablock/[FabLock-ID]/[Lock-ID]/[event]`. In unserem Python Aktor verwenden wir im Einklang mit der dafür geschriebenen Firmware (FabLock for ESP8266 oder FabLock for Ants Make AM-022) die folgenden Topics:

- `/fablock/00001/00001/trigger` - Schloss betätigen- die passende Aktion für den Borepin-Button `UNLOCK`
- `/fablock/00001/00001/identify` - LED blinken lassen und visuell leicht erkennbar machen - die passende Aktion für den Borepin-Button `IDENTIFY`
- `/fablock/00001/00001/feedback` - Status des Schlossriegels (Deadbolt) erhalten

In FabAccess einbinden

Eine nützliche Beispielkonfiguration für FabLocks findet sich in der tfom2023-Demo.

bffh.dhall Snippets

Ressource

```
        LBoxx_1 =
        {
            name = "FabLock Tools",
```

```
            description = "LBoxx with Tools of the FabLock Project",
            disclose = "tfom23.disclose",
            read = "tfom23.read",
            write = "tfom23.lboxx.write",
            manage = "tfom23.manage",
            category = "LBoxx",
            prodable = True,
        },
```

Aktor

```
        fablock_lboxx_1 = {
        module = "Process",
        params = {
            cmd = "python",
            args = /opt/fabinfra/adapters/fablock/main.py --host 127.0.0.1  --user MQTT_USER
--password MQTT_PASSWOR --fablock 00001 --lock 00001"
            }
       },
```

Zuweisung

```
    actor_connections = [
        { machine = "LBoxx_1", actor = "fablock_lboxx_1" }
        ],
```

Version #13
Erstellt: 2024-12-08 01:41:41 CET von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-04-19 22:00:54 CEST von Mario Voigt (Stadtfabrikanten e.V.)