

TP-Link HS110 Integration mit MQTT auf Docker Basis

Mit dieser Anleitung richtest du ein Docker-basiertes Setup ein, um Echtzeitdaten von deinem TP-Link HS110 Smart Plug abzurufen und über MQTT zu veröffentlichen.

Voraussetzungen

- **Docker:** Ein Containerdienst, der das Skript in einer isolierten Umgebung ausführt.
- **Docker Compose:** Ein Tool, das die Konfiguration und Verwaltung des Containers vereinfacht.

Schritt 1: Docker installieren

Installation unter Linux

Führe folgende Befehle aus, um Docker zu installieren:

```
sudo apt update
sudo apt install -y docker.io
```

Prüfe anschließend, ob Docker korrekt installiert wurde:

```
docker --version
```

Installation von Docker Compose

Installiere Docker Compose:

```
sudo apt install -y docker-compose
```

Prüfe die Installation mit:

```
docker-compose --version
```

Hinweis: Auf anderen Systemen wie Windows oder macOS kannst du Docker Desktop installieren, das Docker und Docker Compose kombiniert.

Schritt 2: Projekt bereitstellen

Ordner erstellen

Erstelle einen Ordner für das Projekt:

```
mkdir tplink-mqtt
cd tplink-mqtt
```

`docker-compose.yml` erstellen

Erstelle eine `docker-compose.yml`-Datei mit folgendem Inhalt:

```
version: '3.9'

services:
  tplink-mqtt:
    image: nano94/tplink-hs110-mqtt:latest
    container_name: tplink-mqtt
    environment:
      MQTT_SERVER: "xxx.xxx.xxx.xxx" # MQTT-Broker-Adresse
      MQTT_PORT: 1883 # MQTT-Broker-Port
      MQTT_USER: "loxberry" # Benutzername für den MQTT-Broker
      MQTT_PASSWORD: "XXXXXXX" # Passwort für den MQTT-Broker
      MQTT_TOPIC: "tplink/kasa/<Name des Gerätes vergeben>" # MQTT-Topic für
Sensordaten
      MQTT_CONTROL_TOPIC: "tplink/kasa/<Name des Gerätes vergeben>/control"
# MQTT-Topic für Steuerung
      TARGET_IP: "xxx.xxx.xxx.xxx" # Ziel-IP des TP-Link-Geräts
      INTERVAL: 5 # Intervall für Datenabfragen (Sekunden)
    restart: always
```

Projekt starten

Führe den folgenden Befehl aus, um das Projekt zu starten:

```
docker-compose up -d
```

Schritt 3: MQTT-Daten überprüfen

Wenn das Setup erfolgreich ist, werden die Daten deines TP-Link HS110 über das angegebene MQTT-Topic veröffentlicht. Du kannst diese Daten mit einem MQTT-Client wie **MQTT Explorer** oder **mosquitto_sub** überprüfen.

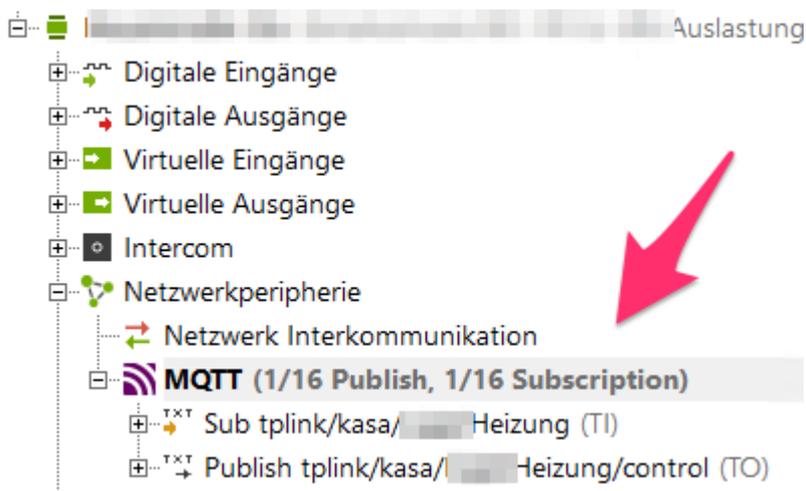
Beispiel für die ausgegebenen Daten:

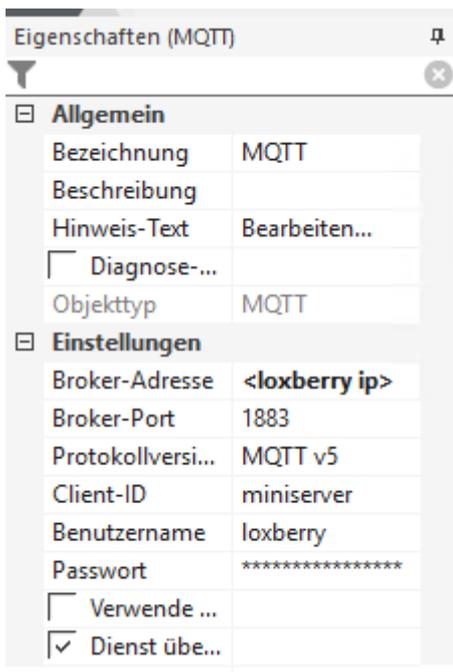
```
{  
  "voltage_mv": 231500,  
  "current_ma": 500,  
  "power_mw": 115000,  
  "total_wh": 150,  
  "err_code": 0,  
  "voltage_v": 231.5,  
  "current_a": 0.5,  
  "power_w": 115.0,  
  "relay_state": 1,  
  "timestamp": "2024-12-03 12:34:56+0100"  
}
```

Schritt 4: Einbindung in Loxone Config

Wenn die vorherigen Schritte erfolgreich sind kann nun MQTT im Miniserver konfiguriert werden. Die Anleitung basiert auf einem Release Stand > 15.3.0.

Hierfür muss MQTT unter "Miniserver"→"Netzwerkperipherie" eingerichtet werden.





Anschließend kann die Subscription und Publish angelegt werden.

Subscription Topic: tplink/kasa/<Vergebener Name des Gerätes>

Publish Topic: tplink/kasa/<Vergebener Name des Gerätes>/control

Schritt 5: Befehl per Loxone senden

Im finalen Schritt kann man nun über einen EIB-Taster den HS-110 ansteuern, sowie den aktuellen Status abgreifen.



Hierfür bindet man die MQTT Subscription an einen "Befehlserkennung"-Baustein an.

Befehlserkennung

```
\i"relay_state": \i\v
```

Am Ausgang des EIB-Tasters fügt man einen "Status"-Baustein am Input "I1" an, dessen "Txt" Output an den MQTT Publish geht.

Status-Baustein Konfiguration:

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Symbolfarbe	Statustext	Statuswert
I1	==	0	-	==	0	-	==	0	-	==	0			{"command": "off"}	0
I1	==	1	-	==	0	-	==	0	-	==	0			{"command": "on"}	0
-	==	0	-	==	0	-	==	0	-	==	0				0

Sobald eine Bedingung zutrifft, wird sie verwendet (Reihenfolge der Texte ist wichtig). Wenn keine Bedingung angegeben ist, dann trifft sie immer zu, und sollte folglich ganz unten sein. Alle Bedingungen sind UND-verknüpft. Einzelne Zeilen können per Drag and Drop verschoben werden.
 Zahl-Vergleiche: == (gleich), > (größer), >= (größer gleich), < (kleiner), <= (kleiner gleich), != (ungleich)
 Text-Vergleiche: == (gleich), != (ungleich), * (enthält), !* (enthält nicht), := (beginnt mit), != (beginnt nicht mit)

Textfelder: <v1> bis <v4> entsprechen den Werten von I1 bis I4
 <v1> = Wert I1 ohne Nachkommastellen oder Text
 <v1.2> = Wert I1 mit 2 Nachkommastellen
 <v2.2*100> = Wert I2 mit 2 Nachkommastellen mal 100
 <v2.2/100> = Wert I2 mit 2 Nachkommastellen durch 100
 <vn> = eigener Objektname
 <v1.t> = Zeitpunkt der Änderung von I1, <v1.d> = EIB Datum I1, <v1.m> = EIB Zeit I1
 Fehlerhafte Texte werden rot angezeigt!

Jalousie-Beispiel laden Werte-Beispiel laden Text-Beispiel laden

Status-Baustein

```

{"command": "off"}
{"command": "on"}

```

Viel Erfolg beim Einrichten! Falls du Fragen hast, melde dich gern!

From:
<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:
https://wiki.loxberry.de/modifikationen_hacks/tp-link_hs110_mqtt_mit_docker?rev=1733386200

Last update: **2024/12/05 09:10**