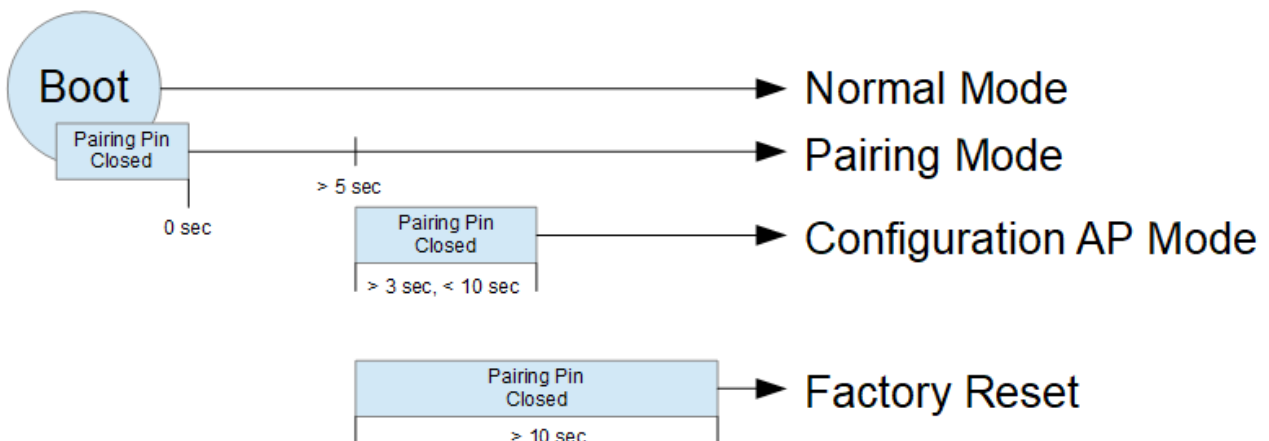


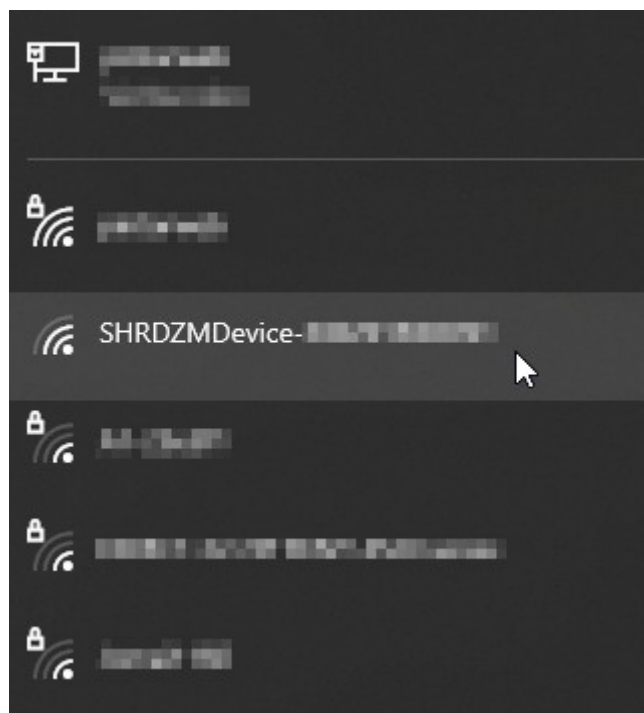
# SHRDZMDevice für SmartMeter

## MBus - Einstellungen

- 1) Pairing-Knopf gedrückt halten und 5V Stromversorgung (mindestens 1A) anstecken.
- 2) Pairing-Knopf loslassen. Die blaue LED blinkt ca. alle halbe Sekunde auf. Das Device befindet sich nun im Pairing-Modus mit einem optionalen Gateway (für Batteriebetrieb, hier nicht der Fall!)
- 3) Mindestens 5 Sekunden warten und Pairing-Knopf erneut für ca. 5 Sekunden gedrückt halten. Loslassen sobald die blaue LED schnell zu blinken beginnt. Nun befindet sich das Device im AccessPoint Modus.



- 4) Sobald sich das Device im AP Modus befindet, kann mit diesem eine WLAN Verbindung aufgebaut werden:



5) Sobald die Verbindung aufgebaut ist, kann mittels Browser über die IP-Adresse **192.168.4.1** darauf zugegriffen werden:

SHRDZMDevice

General

Settings

Gateway

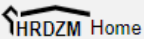
NTP

Cloud

Control


About



Reboot





© Erich O. Pinter

## General



Firmware Version :   
Compile Date : 

Device Type : SMARTMETER  
Chip ID :   
Gateway Mode :   
MQTTTopic Gateway : Disabled  
MQTTTopic Device : Disabled  
MQTTTopic Sensor : Disabled

REST Address : *Currently not reachable*

MQTT Connection State : UNKNOWN  
Date/Time : Unknown

Last Measurement :

IP : (IP unset)  
DNS : (IP unset)  
Gateway : (IP unset)  
Subnet : (IP unset)

Factory Reset!

*Achtung: Ab Firmwareversion 1.2.3 werden nur mehr die für den Devicetyp und Betriebsmodus relevanten Einstellungen angezeigt.*

7) **Save Configuration!** drücken.

8) Auf Gateway Einstellungen wechseln. Hier die WiFi SSID und das Passwort eintragen und Haken bei ‚Device should act as it's own gateway‘ setzen. Damit wählt sich das Modul selbständig ins Wlan ein und braucht kein zusätzliches Gateway.

8a) [optional] Bei Benutzung einer statischen IP, könnte auch diese eingetragen werden. Im Normalfall leer lassen da diese vom Router vergeben wird.

9) MQTT Server, Port und eventuell User und Passwort eintragen. Wenn kein User/Passwort verwendet wird, diesen leer lassen.

9a) [optional] Sollte keine Integration in die eigene Hausautomatisierung gewünscht sein, kann das Häkchen bei ‚MQTT Enable‘ entfernt werden. In solch einem Fall werden dann keine Daten an den MQTT Broker geschickt. (sinnvoll, wenn nur Cloud/Alexa Anbindung gewünscht oder Abfrage mit Webservice)

## ACHTUNG !

**Unbedingt eigenen sicheren und stabilen MQTT Broker verwenden.**

**Auf keinen Fall den ‚test.mosquitto.org‘ als produktiven MQTT Broker verwenden! (ist nur zum Testen geeignet)**

**Abgesehen davon, dass die Daten damit jedem zur Verfügung stehen würden, ist die Stabilität dieses Brokers so schlecht, dass es dadurch auch zu Fehlmessungen kommen kann.**

Anmerkung : Damit Home Assistant das Device Auto-Discovern kann, müssen die Daten am Block im JSon Format geschickt werden

SHRDZMDevice

General

Settings

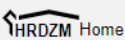
Gateway

NTP

Cloud

About

Reboot



© Erich O. Pintar

## Gateway

### Configuration

WLAN Settings if Device acts as it's own gateway.

☒ Device should act as it's own gateway

---

SSID

Password

☐ Show Password

Static IP

DNS

Gateway

Subnet

---

☒ MQTT Enable (only affected if Device acts as it's on gateway)

MQTT Broker

MQTT Port

MQTT User

MQTT Password

☒ Send values in JSon format

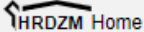
10) **Save Configuration!** drücken.

11) Device booten.

12) Nach dem Rebooten sollte sich das Device im WLAN einwählen und von dort eine eigene IP – Adresse bekommen. Über diese IP-Adresse ist das Device dann wieder mittels Browser erreichbar und Einstellungen können verändert werden.


SHRDZMDevice  
v. 1.0.0 - Build 17.07.2022

General  
Settings  
Gateway  
NTP  
Cloud  
Control  
About  
Reboot

SHRDZM Home

© Erich O. Pintar

## General



Firmware Version : 1.0.0 - Build 17.07.2022  
Compile Date : 17.07.2022 10:00:00

Device Type : SMARTMETER  
Chip ID : 1234567890  
Gateway Mode : true  
MQTTTopic Gateway : SHRDZM/1234567890  
MQTTTopic Device : SHRDZM/1234567890  
MQTTTopic Sensor : SHRDZM/1234567890

REST Address : http://192.168.1.1:8080/shrdzm-smartmeter/

MQTT Connection State : Connected  
Date/Time : 2022-04-06T17:15:25

Last Measurement :

```
(0)timestamp=2022-04-06T17:15:25  
(1)1.8.0=349964  
(2)2.8.0=662268  
(3)1.7.0=0  
(4)2.7.0=3409  
(5)32.7.0=234.80  
(6)52.7.0=235.30  
(7)72.7.0=232.40  
(8)31.7.0=4.23  
(9)51.7.0=5.40  
(10)71.7.0=5.04  
(11)13.7.0=99.60  
(12)16.7.0=-3409  
(13)uptime=0000:00:01:28
```

IP : 192.168.1.100  
DNS : 192.168.1.1  
Gateway : 192.168.1.1  
Subnet : 192.168.1.0/24

SSID:pintarweb, Ch:10 (-53dBm), Quality:Excellent

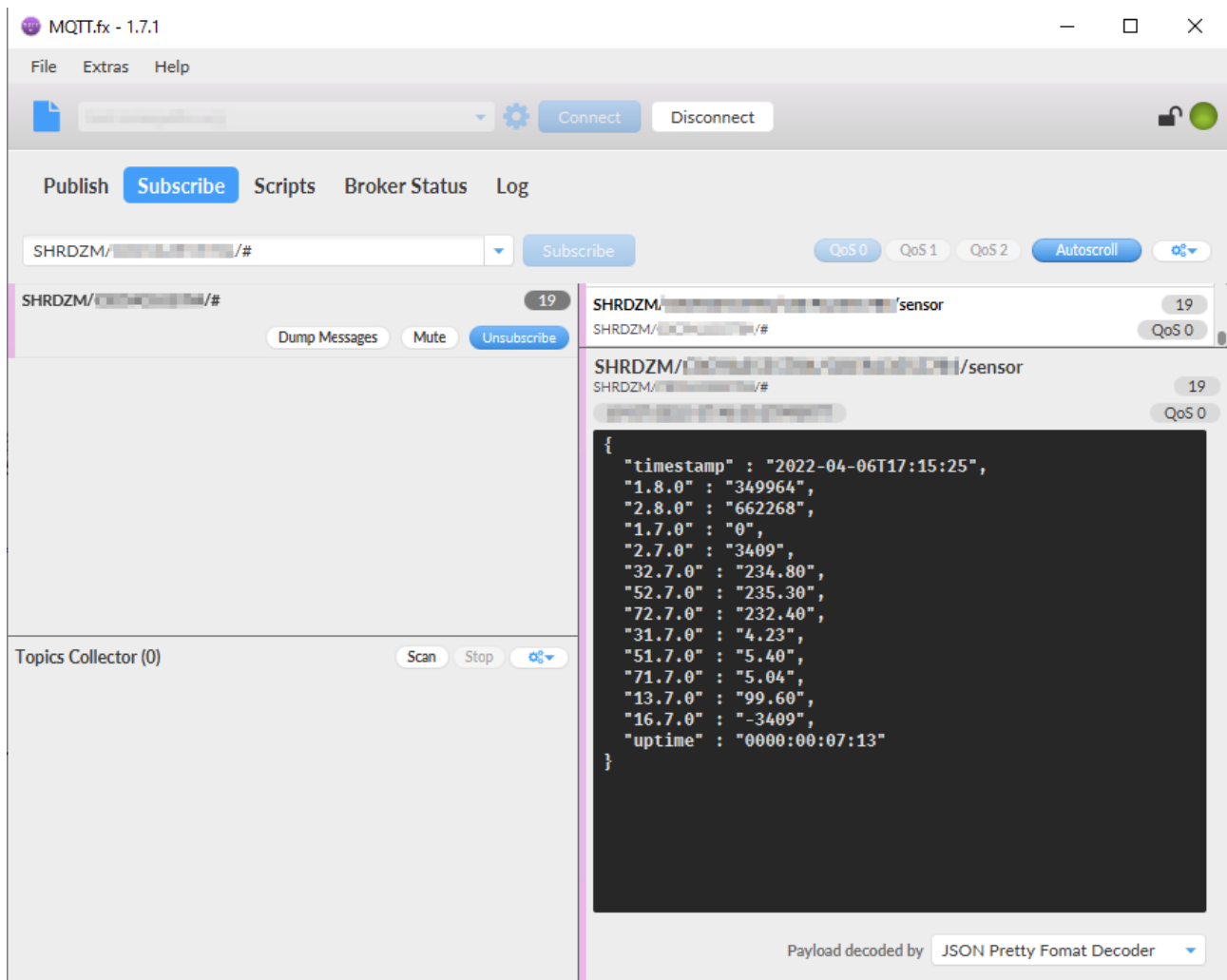
[Factory Reset!](#)

[OTA Upgrade](#) <http://shrdzm-smartmeter.pintarweb.net/upgrade.php>

*Caution! Do not upgrade a working version if you dont have a valid reason!*

13) Nach dem Verbinden mit dem SmartMeter sollten auch die Messwerte über die auf der Übersichtsseite beschriebenen MQTT Subjects angezeigt werden.  
Bei den Parameternamen handelt es sich um die international standardisierten OBIS Codes welche die Bedeutung der einzelnen Werte beschreibt bzw. stehen im Normalfall auch auf dem Smartmeter.

Das Modul sendet alle Werte die das Smartmeter liefert, Anzahl und Typ hängen vom Stromnetzbetreiber und Smartmeter ab!



Beispielanzeige mit MQTT.fx Testclient

# Optional

## Volkszaehler – Einstellungen

Eigener Volkszähler Server muss bereits aufgesetzt sein, Informationen dazu auf <https://wiki.volkszaehler.org/howto/getstarted>

### 1) Kanal hinzufügen

**Kanal hinzufügen** ✕

Öffentliche Kanäle Private Kanäle **Kanal erstellen**

Eigenschaft	Wert
Middleware:	Local (default) ▾
Typ:	El. Energie (Leistungswerte) ▾
Titel	Aktueller Bezug
Öffentlich	<input type="checkbox"/>
Farbe	<span style="background-color: cyan; border: 1px solid black; display: inline-block; width: 20px; height: 10px;"></span>
Stil	lines ▾
Füllgrad	<input type="text"/>
Linienstil	solid ▾
Linienstärke	<input type="text"/>
Achse	auto ▾
Initialverbrauch	<input type="text"/>
Kosten	<input type="text"/>
Auflösung	<input type="text"/>

**Erstellen** Cookie: ☒

### 2) UUID des Kanals kopieren

**Details für Aktuelle Einspeisung** ✕

Titel	Aktuelle Einspeisung
Typ	powersensor
UUID	894b7650-0525-11ed-957b-dfa7ca63aa4e
Aktiv	✓
Farbe	aqua
Stil	Linien
Füllgrad	0
Linienstil	Solide
Achse	auto

**Daten** **Löschen** **Bearbeiten** **Schließen**

- 3) IP Adresse des volkszaehler servers eingeben und Haken bei Volkszähler setzen.  
Den zu sendenden OBIS Code und die UUID vom Volkszaehler Kanal eingeben (mit Pipe getrennt)  
Jede Zeile ist ein eigener OBIS Code bzw. Volkszähler-Kanal

☒ Volkszaehler

Server

Channel Mapping

1.7.0 | 5757dbe0-0525-11ed-bbcb-0d79d8986835  
2.7.0 | 894b7650-0525-11ed-957b-dfa7ca63aa4e

- 4) In der Volkszaehler Weboberfläche sollten dann die empfangenen Daten den entsprechenden Kanälen zugeordnet erscheinen.

Kanäle								
<input type="checkbox"/>	Titel	Typ	Min	Max	Ø	Aktuell	Verbrauch	Aktion
<input checked="" type="checkbox"/>	 Aktuelle Einspeisung	El. Energie (Leistungswerte)	3.51 kW	3.51 kW	3.51 kW	3.51 kW	21.4 Wh	 
<input checked="" type="checkbox"/>	 Aktueller Bezug	El. Energie (Leistungswerte)	0 W	0 W	0 W	0 W	0 Wh	 



## Thingspeak – Einstellungen

Auf die Bedienung von Thingspeak wird hier nicht eingegangen.  
Informationen dazu auf <https://thingspeak.com/>

- 1) Channel für Smartmeterdaten und Write API Key bei thingspeak erstellen

### Smartmeter

Channel ID: [REDACTED]

Author: [REDACTED]

Access: Private

Private View

Public View

Channel Settings

Sharing

API Keys

### Write API Key

Key

[REDACTED]

Generate New Write API Key

Read API Keys

- 2) Erzeugten Write API Key eintragen und den Haken bei Thingspeak setzen.  
(Thingspeak Server Adresse braucht im Normalfall nicht verändert werden)

☒ Thingspeak

<http://api.thingspeak.com/update>

Thingspeak Server

[REDACTED]

Write API Key

- 3) Channels setzen an denen man interessiert ist  
Hierbei ist es notwendig, die Felder in der Reihenfolge zu definieren wie diese geliefert werden.

[Private View](#) [Public View](#) [Channel Settings](#) [Sharing](#) [API Keys](#) [I](#)

## Channel Settings

Percentage complete 30%

Channel ID XXXXXXXXXX

Name

Description

Field 1	<input type="text" value="Gesamtverbrauch"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Gesamteinspeisung"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text"/>	<input type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>
Field 5	<input type="text" value="Verbrauch"/>	<input checked="" type="checkbox"/>
Field 6	<input type="text" value="Einspeisung"/>	<input checked="" type="checkbox"/>
Field 7	<input type="text"/>	<input type="checkbox"/>
Field 8	<input type="text"/>	<input type="checkbox"/>

Metadata

- 4) Visualisierungen bei Thingspeak für die anzuzeigenden Datenhinzufügen

Click on a visualization to add it to the Channel

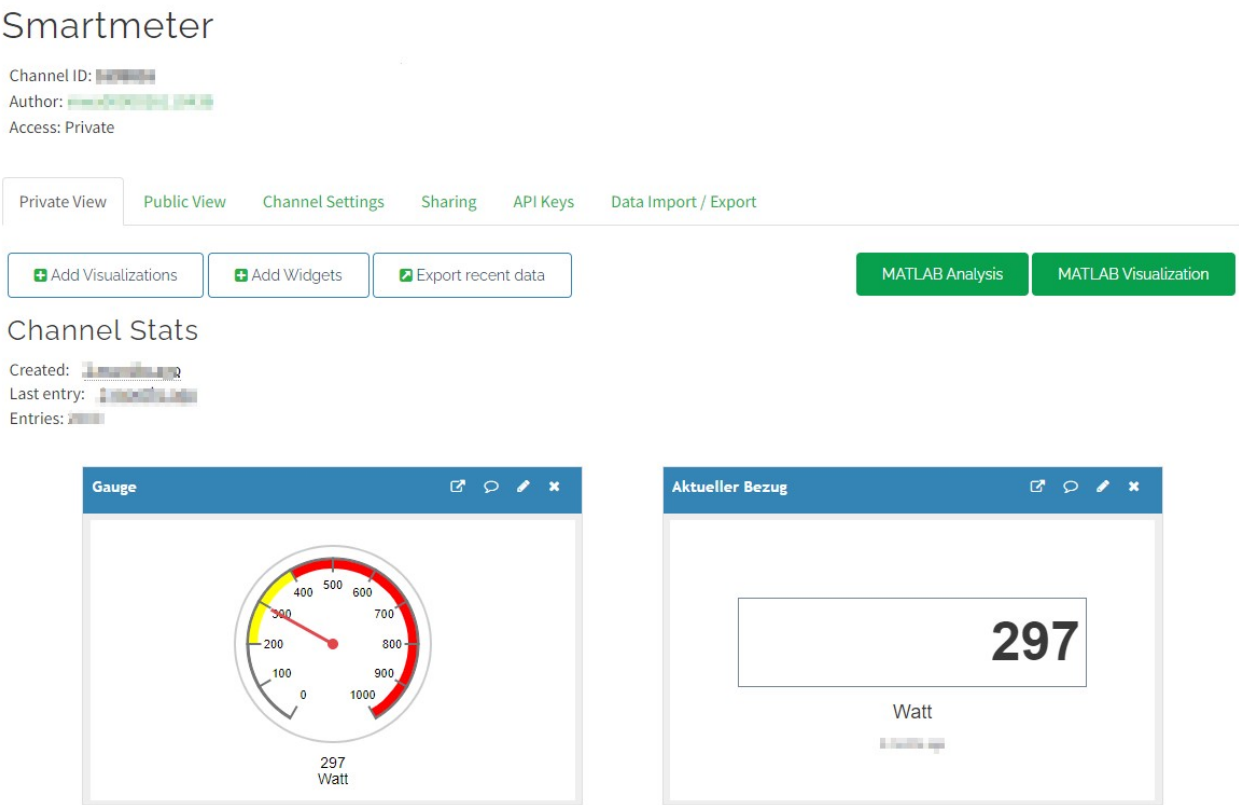
×



Save

Cancel

5) Beispielanzeige der Daten ...



## Schalten von Geräten bei Schwellwertüberschreitungen

Um zB.: bei Balkonkraftwerken so gut als möglich den selbst erzeugten Strom zu verwenden anstatt diesen einzuspeisen (Nulleinspeisung) kann das Modul Geräte welche über eine Webservice Schnittstelle verfügen, schalten. (zB.: Tasmota/ESPURNA geflashte Zwischensteckdosen)

Dazu im Reiter ‚Control‘ den Haken bei ‚HTTP Request‘ setzen und die Einträge bei ‚Upper Limit Reaction‘ setzen bei wessen Überschreitung ein Webrequest abgesetzt werden soll.

Dasselbe gilt für die ‚Lower Limit Reaction‘ bei dessen Unterschreitung ein Webrequest gesendet werden soll.

Dazu ist folgendes Format einzuhalten:

<Name des Parameters>|<Limitwert>|<Webrequest>

The screenshot shows the 'SHRDZMDevice' web interface. On the left is a sidebar with menu items: General, Settings, Gateway, NTP, Cloud, Control (selected), About, and Reboot. Below the menu is a 'Home' button with a house icon and a copyright notice. The main area is titled 'Control' and contains the text 'Control other devices.' Below this is a section for 'HTTP Request' which is checked. Under 'Upper Limit Reaction', there are two entries in a table-like format: '2.7.0|800|http://192.168.1.165/api/relay/0?apikey=E1B1BF8DCA2834A1&value=1' and '2.7.0|1600|http://192.168.1.164/cm?cmd=Power%20On'. Callout boxes explain these: the first is 'zB.: ESPURNA Schalter Einschalten bei Einspeisung über 800 Watt' and the second is 'zB.: Tasmota Zwischensteckdose Einschalten bei Einspeisung über 1600 Watt'. Under 'Lower Limit Reaction', there are two entries: '2.7.0|200|http://192.168.1.165/api/relay/0?apikey=E1B1BF8DCA2834A1&value=0' and '2.7.0|500|http://192.168.1.164/cm?cmd=Power%20Off'. Callout boxes explain these: the first is 'zB.: ESPURNA Schalter Ausschalten bei Einspeisung unter 200 Watt' and the second is 'zB.: Tasmota Zwischensteckdose Ausschalten bei Einspeisung unter 500 Watt'. At the bottom is a 'Save Control Settings!' button.

Das Format des jeweiligen Webrequests für Tasmota bzw. ESPURNA bitte auf deren Wiki nachschlagen.

Es können jedoch auch beliebige andere Webrequests abgesendet werden.