

Dokumentation der Integration von gPlugM in Home Assistant

Ausgangslage

Seit dem Jahr 2021 ist in unserem Haus ein Smart Meter vom Typ E450 (Firma Landis & Gyr) installiert und im operationellen Betrieb. Aus diversen Gründen bin ich daran interessiert, wann und wieviel elektrische Energie (Hoch- und Niedertarif) ich beziehe. Der Energiedienstleister ist die [Energie Thun AG](#).

Seit nahezu zwei Jahren betreibe ich zwei Instanzen (eine Test- und Produktions-Plattform) von Home Assistant. Home Assistant ist eine kostenlose, quelloffene Software zum Thema Hausautomation. Mittels dieser Software will ich den elektrischen Energieverbrauch visualisieren/auswerten.

Per Zufall bin ich auf [Produkte der Firma Gantrisch Energie AG](#) gestossen, welche unter anderem auch in Home Assistant integriert werden können. Es besteht bereits eine Beschreibung für das Produkt gPlugD ([Integration gPlug in Home Assistant](#)).

Diese Dokumentation beschreibt die Integration auf der Test Plattform von Home Assistant. Sie ist eine Ergänzung zu den bestehenden Dokumentationen der Firma Gantrisch Energie AG und soll weiteren zukünftigen Benutzern die Integration erleichtern. Die Dokumentation basiert auf dem Wissensstand vom Juli 2025.

Ist Zustand

Als IoT-Adapter für den Landis + Gyr (L+G) Smart Meter E450 eignet sich das [Produkt gPlugM](#). Folgende Komponenten und Software Versionen werden eingesetzt

Software

Smart Meter:

- Landis & Gyr E450

Home Assistant

- Version 2025.7.1 bestehend aus:
 - Core 2025.7.3
 - Supervisor 2025.07.1
 - Operating System 16.0
 - Frontend 20250702.3
- Folgende Add-ons werden benötigt:
 - Mosquito MQTT Broker 6.5.1
 - MQTT Explorer 1.0.3 (Hilfreich für Fehlersuche).

gPlugM IoT-Adapter

- Modell: gPlugM (Tasmota Version: 14.5.0 [tasmota32])

IP/WLAN Netzwerk

- Es wird eine WLAN-Verbindung benötigt. In meinem Fall wird dabei ein dediziertes "Telemetrie" WLAN (mit eigener SSID) genutzt.
- Die WLAN-Verbindung zwischen dem gPlugM IoT Adapter und dem WLAN Access Point ist verschlüsselt.
- Es wird ein IPv4 Subnetz gemäss [RFC 1918](#) verwendet. Der Zugriff auf dieses Subnetz vom Internet aus ist nicht möglich.

Vorgehen

Folgende Arbeitsschritte wurde ausgeführt

1. Kontaktaufnahme mit dem Energiedienstleister mit dem Ziel:
 - a. Eine Ansprechperson zu erhalten
 - b. Um zu erfahren welche Informationen der Stromlieferant bzw. der Smart Meter liefert bzw. zu Verfügung stellt.
 - c. Um Technische Fragen/Probleme bei der Betriebsaufnahme abzuklären
2. Anschluss des gPlugM IoT-Adapters an das WLAN bzw. Die Integration in das IP-Netzwerk.
3. Anschluss des gPlugM IoT-Adapters an den Smart Meter und anschliessend der Verbindungstest zwischen dem Smart Meter und dem gPlugM IoT Adapter.
4. Konfiguration der Verbindung zwischen dem gPlugM IoT Adapter und der Home Assistant Instanz via MQTT.
5. Test der Verbindung zwischen dem gPlugM IoT Adapter und der Home Assistant Instanz auf Stufe Netzwerk als auch Stufe MQTT mittels dem MQTT Explorer.
6. Anpassungen/Konfiguration der Home Assistant Umgebung
7. Konfiguration / Anpassung des Energie Dashboards.

1 Kontaktaufnahme mit dem Energiedienstleister

Der Energielieferant (Energie Thun AG) stellt 1/4 jährlich Rechnung für den Bezug von Strom, Wasser und Gas. Detaillierte Informationen stehen im Moment (Juli 2025) auf dem Endkundenportal nicht bzw. noch nicht zu Verfügung. Mein Anliegen für die Nutzung der Schnittstelle des Smart Meters formulierte ich mittels einer Supportanfrage via E-Mail an die Energie Thun AG.

Der Smart Meter Landis & Gyr E450 ist in der Lage diverse Informationen zum Thema Energieverbrauch bzw. Energielieferung über die Endkundenschnittstelle zu Verfügung zu stellen. Bei der für den gPlugM IoT Adapter genutzten Schnittstelle handelt es sich um eine M-Bus Schnittstelle (Feldbus). Informationen dazu sind über folgende Links verfügbar:

Informationen zum Thema M-Bus

[https://de.wikipedia.org/wiki/M-Bus_\(Feldbus\)](https://de.wikipedia.org/wiki/M-Bus_(Feldbus))

<https://m-bus.com/>

Informationen der Firma Landis & Gyr

https://www.landisgyr.ch/webfoo/wp-content/uploads/2025/03/LandisGyr_Standard-CIP-Liste.pdf

Messwerte wie Energiemengen (Zählerstände) oder Leistungswerte werden über so genannte OBIS-Kennziffern/Codes gekennzeichnet.

Informationen zum Thema OBIS-Kennziffern/Codes

[OBIS codes - OneMeter Documentation](#)

Erklärvideo auf YouTube: [OBIS-Kennziffer einfach erklärt](#)

2 Anschluss des gPlugM IoT-Adapters an das IP-Netzwerk (WLAN-Verbindung)

Das dedizierte Telemetrie WLAN (SSID: Telemetrie) wurde auf dem gPlugM IoT Adapter konfiguriert.

IoT-Adapter

gPlugM

Telemetrie 

sgz-74297 

SporisHome 

Suche nach WLAN

WLAN

WLAN 1 - SSID ()

Passwort

WLAN 2 - SSID ()

Passwort

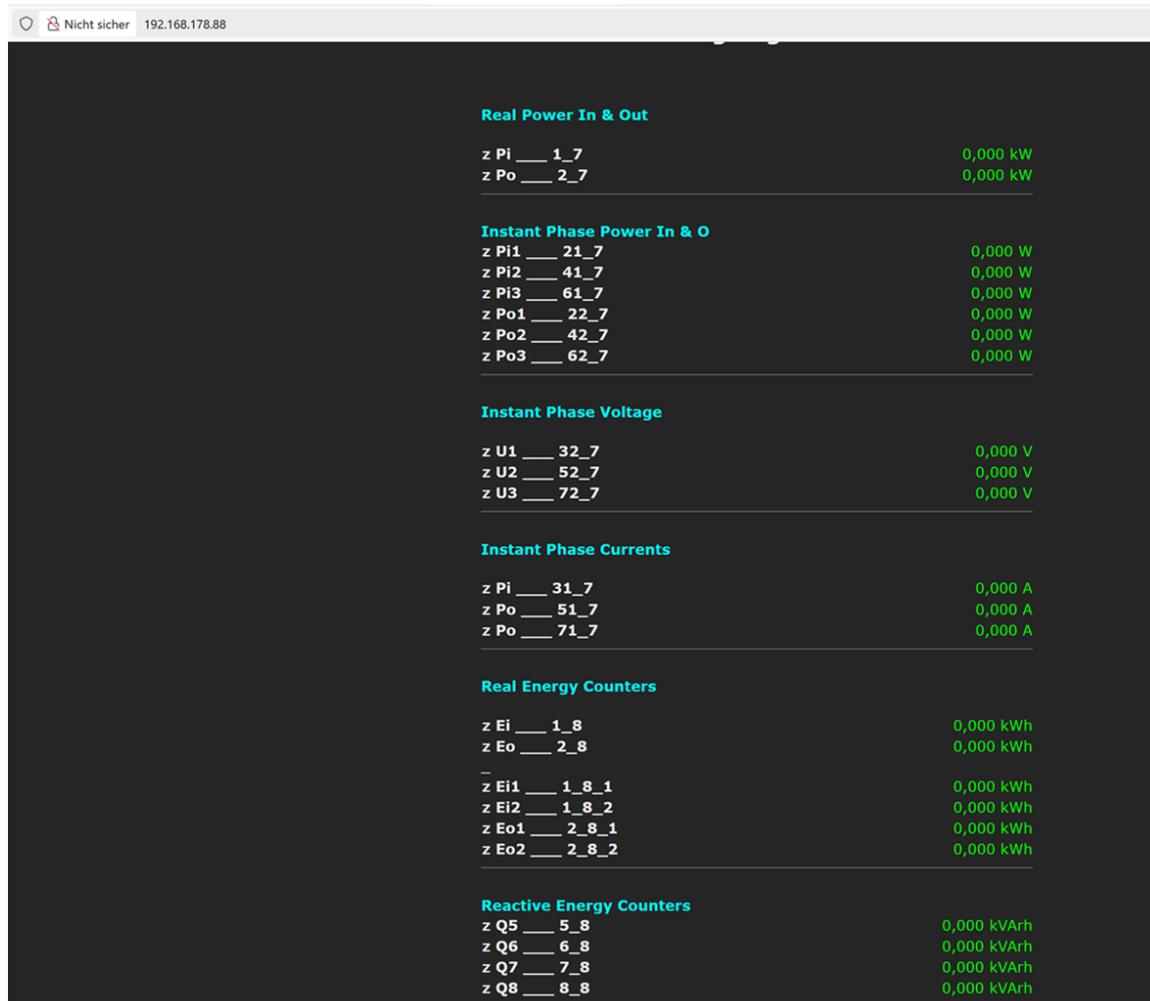
Hostname (gPlugM)

Speichern

Einstellungen

3 Anschluss des gPlugM IoT-Adapters an den Smart Meter

Sobald die Schnittstelle durch den Stromlieferanten freigeschaltet wurde, kann der gPlugM IoT Adapter an das Smart Meter angeschlossen werden. Die ersten Resultate war nicht gerade viel versprechend. Das Web Interface des gPlugM IoT Adapters zeigten keine Informationen an:



The screenshot shows a web browser window with the address bar displaying 'Nicht sicher 192.168.178.88'. The main content area has a dark background and displays several sections of data in a light blue font, with values in green. The sections are: 'Real Power In & Out', 'Instant Phase Power In & O', 'Instant Phase Voltage', 'Instant Phase Currents', 'Real Energy Counters', and 'Reactive Energy Counters'. Each section lists several parameters with their corresponding values.

Real Power In & Out	
z Pi ___ 1_7	0,000 kW
z Po ___ 2_7	0,000 kW
Instant Phase Power In & O	
z Pi1 ___ 21_7	0,000 W
z Pi2 ___ 41_7	0,000 W
z Pi3 ___ 61_7	0,000 W
z Po1 ___ 22_7	0,000 W
z Po2 ___ 42_7	0,000 W
z Po3 ___ 62_7	0,000 W
Instant Phase Voltage	
z U1 ___ 32_7	0,000 V
z U2 ___ 52_7	0,000 V
z U3 ___ 72_7	0,000 V
Instant Phase Currents	
z Pi ___ 31_7	0,000 A
z Po ___ 51_7	0,000 A
z Po ___ 71_7	0,000 A
Real Energy Counters	
z Ei ___ 1_8	0,000 kWh
z Eo ___ 2_8	0,000 kWh
—	
z Ei1 ___ 1_8_1	0,000 kWh
z Ei2 ___ 1_8_2	0,000 kWh
z Eo1 ___ 2_8_1	0,000 kWh
z Eo2 ___ 2_8_2	0,000 kWh
Reactive Energy Counters	
z Q5 ___ 5_8	0,000 kVArh
z Q6 ___ 6_8	0,000 kVArh
z Q7 ___ 7_8	0,000 kVArh
z Q8 ___ 8_8	0,000 kVArh

Auf Anraten des Supports von Gantrisch Energie AG wurde überprüft, ob überhaupt ein Datentransfer zwischen dem Smart Meter und dem gPlugM IoT Adapter stattfindet. Die entsprechenden Informationen zum Vorgehen sind hier zu finden: [Smart Meter Daten prüfen](#).

Dank dem Support (Merci Herr Hüni!) konnte festgestellt werden, dass die Daten verschlüsselt übertragen werden. In einem ersten Schritt konnte das Problem dadurch behoben werden, dass die Verschlüsselung durch den Energielieferanten ausgeschaltet wurde (Smart Meter und gPlugM IoT Adapter sind nur wenige cm voneinander entfernt installiert. Der physische Zugriff auf beide Geräte ist abgesichert). Nachdem die Verschlüsselung ausgeschaltet wurde, sind nun auf der Konsole des gPlugM IoT Adapters die durch den Smart Meter gelieferten Daten ersichtlich.

IoT-Adapter gPlugM

Real Power In & Out

z Pi ___ 1_7	0,348 kW
z Po ___ 2_7	0,000 kW

Instant Phase Power In & O

z Pi1 ___ 21_7	0,000 W
z Pi2 ___ 41_7	0,000 W
z Pi3 ___ 61_7	0,000 W
z Po1 ___ 22_7	0,000 W
z Po2 ___ 42_7	0,000 W
z Po3 ___ 62_7	0,000 W

Instant Phase Voltage

z U1 ___ 32_7	0,000 V
z U2 ___ 52_7	0,000 V
z U3 ___ 72_7	0,000 V

Instant Phase Currents

z Pi ___ 31_7	0,000 A
z Po ___ 51_7	0,000 A
z Po ___ 71_7	0,000 A

Real Energy Counters

z Ei ___ 1_8	16268,899 kWh
z Eo ___ 2_8	5,420 kWh
—	
z Ei1 ___ 1_8_1	12096,166 kWh
z Ei2 ___ 1_8_2	4172,732 kWh
z Eo1 ___ 2_8_1	5,420 kWh
z Eo2 ___ 2_8_2	0,000 kWh

Reactive Energy Counters

z Q5 ___ 5_8	103,749 kVAh
z Q6 ___ 6_8	2,080 kVAh
z Q7 ___ 7_8	1,091 kVAh
z Q8 ___ 8_8	5196,824 kVAh
—	
z Q51 ___ 5_8_1	75,330 kVAh
z Q52 ___ 5_8_2	28,419 kVAh
z Q61 ___ 6_8_1	2,080 kVAh
z Q62 ___ 6_8_2	0,000 kVAh
z Q71 ___ 7_8_1	1,091 kVAh
z Q72 ___ 7_8_2	0,000 kVAh
z Q81 ___ 8_8_1	3542,171 kVAh
z Q82 ___ 8_8_2	1654,653 kVAh

L+G E450, data freq. 1/5 Hz, counters 1/60 Hz

SM-ID	55521260
Date & Time	2025-07-14T11:38:28
Uptime	333.1 h

4 Konfiguration der Verbindung zwischen dem gPlugM IoT Adapter und der Home Assistant Instanz via MQTT

Für die Integration des gPlugM IoT Adapter wird ein MQTT-Broker benötigt. Der Mosquito MQTT-Broker ist bereits seit einiger Zeit auf beiden Home Assistant Instanzen (Test und Produktion) installiert und in Betrieb. Für die Integration des gPlugM IoT Adapters wird ein entsprechender User mit Passwort benötigt. Die benötigten Parameter können nun auf der Konsole des gPlugM IoT Adapters konfiguriert werden.

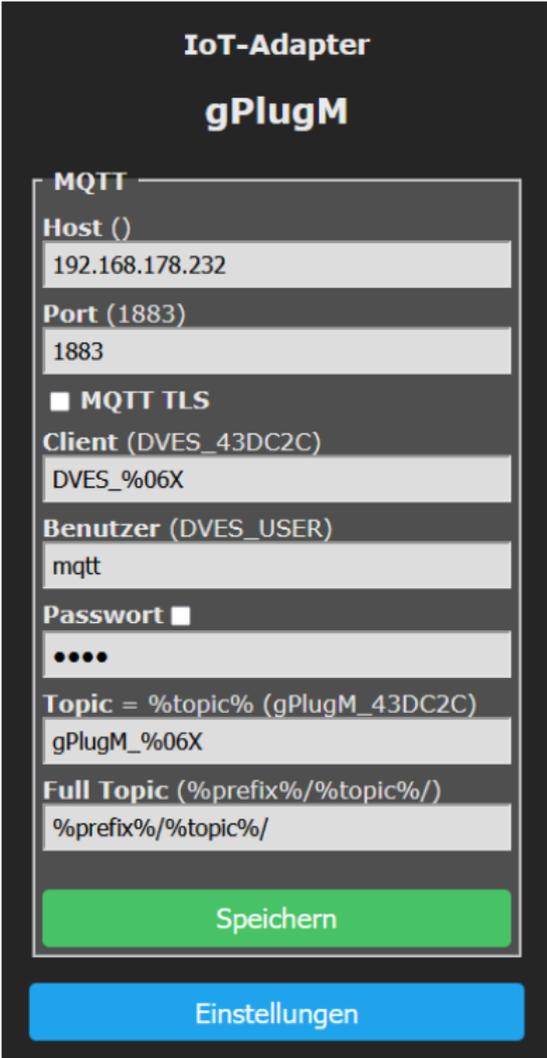
IPv4 Adresse: 192.168.178.232

Port Nr.: 1883

Benutzer: xxxx

Passwort: xxxx

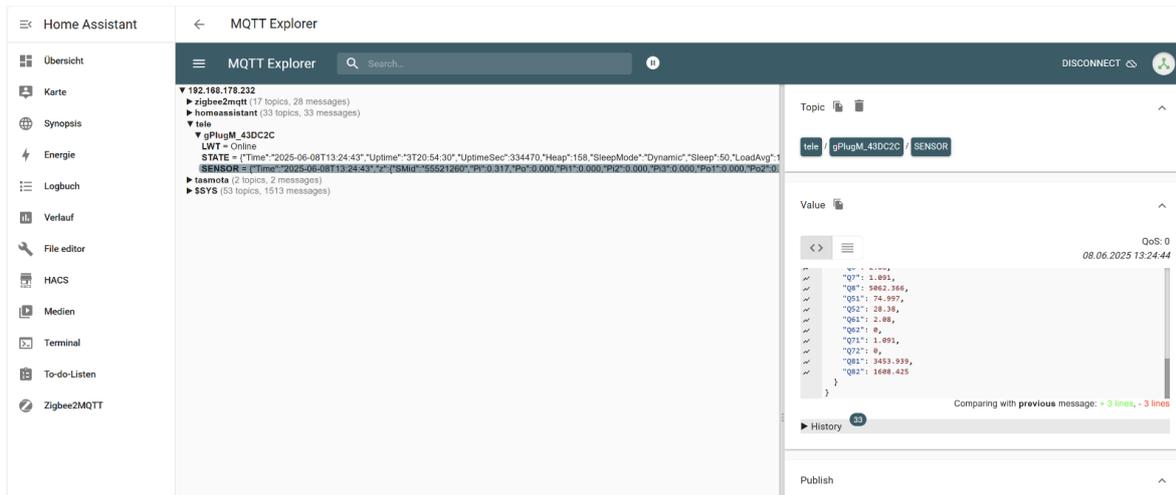
(User und Passwort sind in der MQTT zugehörigen Dokumentation aufgeführt)



The image shows a configuration screen for the gPlugM IoT Adapter. The title is "IoT-Adapter gPlugM". Under the "MQTT" section, there are several input fields: "Host ()" with the value "192.168.178.232", "Port (1883)" with the value "1883", "Client (DVES_43DC2C)" with the value "DVES_%06X", "Benutzer (DVES_USER)" with the value "mqtt", and "Passwort" with four dots. Below these are "Topic = %topic% (gPlugM_43DC2C)" with the value "gPlugM_%06X" and "Full Topic (%prefix%/ %topic%/)" with the value "%prefix%/ %topic%/". At the bottom of the MQTT section is a green "Speichern" button. Below the entire configuration area is a blue "Einstellungen" button.

5 Verbindungstest zwischen dem gPlugM IoT Adapter und Home Assistant

Auf der Home Assistant Instanz waren zuerst keine Entitäten ersichtlich. Das Problem bzw. Die Ursache konnte mit dem Add-on *MQTT Explorer* lokalisiert werden. Auf dem MQTT Explorer sind die durch den gPlugM IoT Adapter gelieferten Daten ersichtlich. Das zugehörige MQTT-Topic lautet **"tele/gPlugM_43DC2C/SENSOR"**.



Interessant!

MQTT-Topics die ein Präfix "homeassistant" aufweisen, werden automatisch durch Home Assistant erkannt! Alle anderen Topics müssen manuell konfiguriert werden bzw. werden durch das zugehörige Add-on (z.B.: beim Add-on zigbee2mqtt) konfiguriert.

Die manuelle Konfiguration erfolgt über die Benutzeroberfläche des Mosquitto MQTT Brokers: "Einstellungen / Geräte & Dienste" danach wird die Integration MQTT gewählt. In den MQTT-Einstellungen kann dann das MQTT-Topic (in unserem Falle: **"tele/gPlugM_43DC2C/SENSOR"**) abonniert werden.

MQTT-Einstellungen

MQTT-OPTIONEN KONFIGURIEREN

Ein Paket veröffentlichen

Topic

QoS
0



Retain

Template zulassen

Payload

1



VERÖFFENTLICHEN

Ein Topic abonnieren

JSON-Inhalt formatieren

Topic, das abonniert werden soll

QoS
0

ABO BEGINNEN

6 Anpassungen/Konfiguration der Home Assistant Umgebung

Nachdem das Topic erfolgreich abonniert wurde, sind entsprechenden Entitäten in Home Assistant sichtbar. Diesen Entitäten kann ein passender Name zugewiesen werden. In unserem Fall wird der Entität `sensor.gplugm_z_ei` der Name "Bezug Gesamtenergie (HT und NT)" zugewiesen.

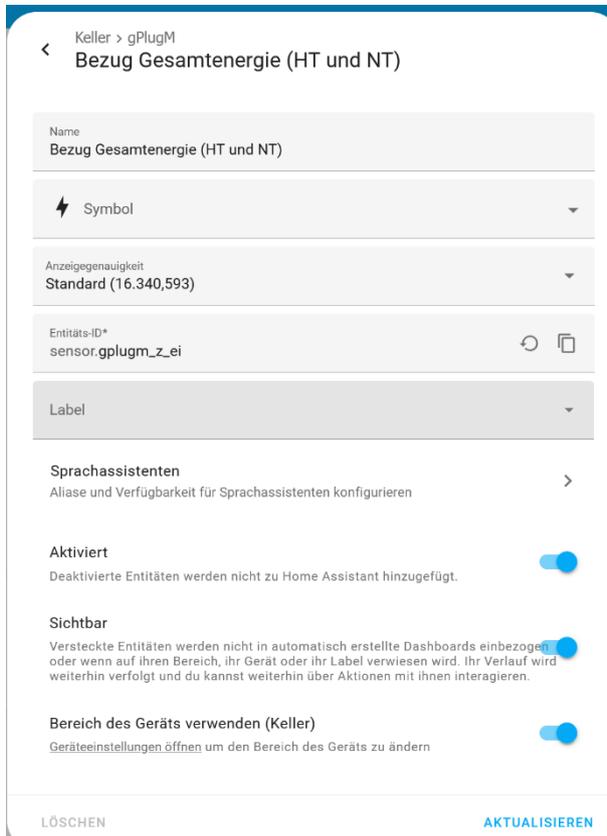


Abbildung: Bezug Gesamtenergie (HT und NT)

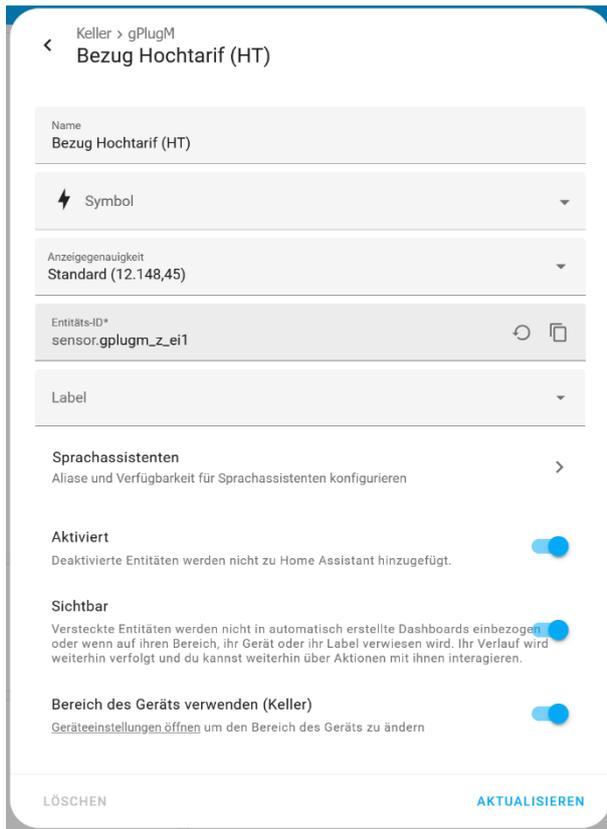


Abbildung: Bezug Hochtarif (HT)

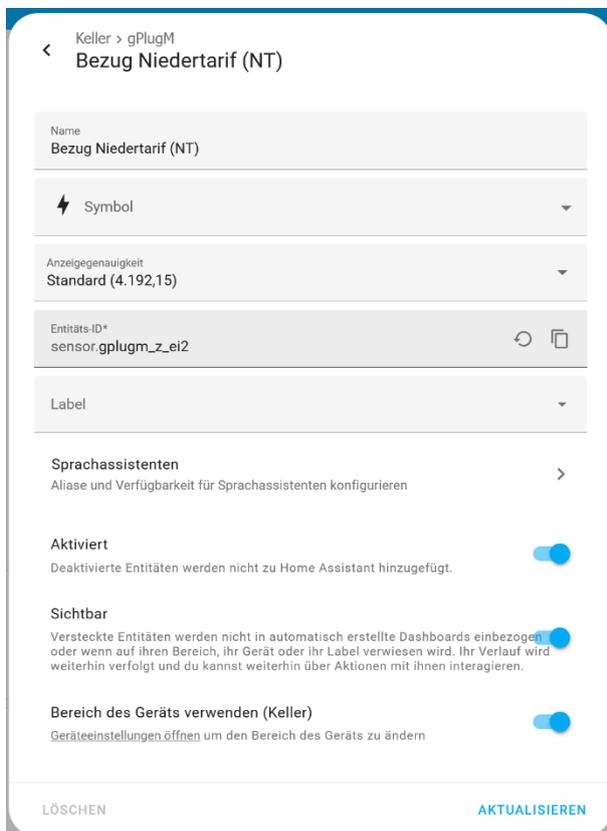


Abbildung: Bezug Niedertarif (NT)

Home Assistant benötigt für die Visualisierung/Auswertung der durch den gPlugM IoT Adapter gelieferten Messwerte zugehörige Einheiten. Die dazu benötigte Konfiguration muss in der Datei *configuration.yaml* vorgenommen werden:

Hier ein Ausschnitt (aus der Datei *configuration.yaml*) für die Messwerte bzw. Entitäten:

- `sensor.gplugm_z_ei`
- `sensor.gplugm_z_ei1`
- `sensor.gplugm_z_ei2`

homeassistant:

customize:

```
sensor.gplugm_z_ei:  
  unit_of_measurement: "kWh"  
  device_class: energy  
  state_class: total_increasing  
  last_reset: 1970-01-01T00:00:00+00:00  
sensor.gplugm_z_ei1:  
  unit_of_measurement: "kWh"  
  device_class: energy  
  state_class: total_increasing  
  last_reset: 1970-01-01T00:00:00+00:00  
sensor.gplugm_z_ei2:  
  unit_of_measurement: "kWh"  
  device_class: energy  
  state_class: total_increasing  
  last_reset: 1970-01-01T00:00:00+00:00
```

Tipp: Die Konfiguration bzw. Die YAML-Syntax unbedingt prüfen (*Entwicklerwerkzeuge* von Home Assistant)

Interessant sind die Helpers in Home Assistant. Sie sind ein sehr nützliches Tool, um den Verbrauch von Zählern (z.B. für Strom, Wasser, Gas, Heizung) über bestimmte Zeiträume zu verfolgen und automatisch zurückzusetzen. Sie sind ideal, um aggregierte Verbrauchsdaten für das Home Assistant Energie-Dashboard oder für eigene Auswertungen zu generieren. Weitere Infos zum Thema Utility Meter sind hier zu finden:

https://www.home-assistant.io/integrations/utility_meter/

Die nachfolgend aufgeführten Helpers nutzen folgende Entitäten:

- *sensor.gplugm_z_ei*
- *sensor.gplugm_z_ei1*
- *sensor.gplugm_z_ei2*

utility_meter:

energy_in_daily_ei:

source: sensor.gplugm_z_ei

cycle: daily

energy_in_weekly_ei:

source: sensor.gplugm_z_ei

cycle: weekly

energy_in_monthly_ei:

source: sensor.gplugm_z_ei

cycle: monthly

energy_in_yearly_ei:

source: sensor.gplugm_z_ei

cycle: yearly

energy_in_daily_ei1:

source: sensor.gplugm_z_ei1

cycle: daily

energy_in_weekly_ei1:

source: sensor.gplugm_z_ei1

cycle: weekly

energy_in_monthly_ei1:

source: sensor.gplugm_z_ei1

cycle: monthly

energy_in_yearly_ei1:

source: sensor.gplugm_z_ei1

cycle: yearly

energy_in_daily_ei2:

source: sensor.gplugm_z_ei2

cycle: daily

energy_in_weekly_ei2:

source: sensor.gplugm_z_ei2

cycle: weekly

energy_in_monthly_ei2:

source: sensor.gplugm_z_ei2

cycle: monthly

energy_in_yearly_ei2:

source: sensor.gplugm_z_ei2

cycle: yearly

7 Konfiguration / Anpassung des Energie Dashboards.

Home Assistant beinhaltet bereits bei der Installation ein Dashboard zum Thema Energie. Damit kann der der Bezug aber auch die Lieferung von Energie erfasst werden. In diesem Fall ist nur die Lieferung von elektrischer Energie erfasst.

