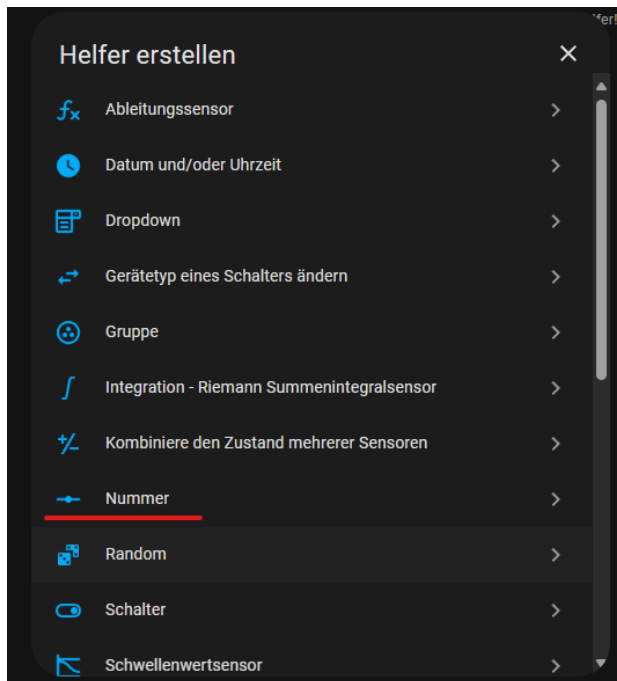


Energy Dashboard - richtig einrichten

Als erstes legen wir uns einen Helfer für den Strompreis an:

Strompreis Helfer anlegen

Als erstes legen wir einen Helfer für unseren Strompreis fest. Hierzu gehen wir auf Einstellungen - Geräte Dienste - Helfer:



Hier müssen wir folgende Config eintragen:

Nummer erstellen

Name*

Strompreis

€

Symbol

mdi:currency-eur

×

▼

Minimaler Wert

0

Maximaler Wert

100

Anzeigemodus

☒ Schieberegler

☐ Eingabefeld

Schrittgröße

0,0001

Maßeinheit

EUR /kWh

ZURÜCK

ERSTELLEN

Schrittgröße:

0,0001

Maßeinheit:

EUR /kWh

optional falls nicht vorhanden - template.yaml anlegen

Solltest du noch keine Template yaml haben legen wir diese jetzt an. Füge folgenden Code in der configuration.yaml ein:

```
template: !include template.yaml
```

das sollte dann in etwa so aussehen:

```

19 http: !include http.yaml
20 tts: !include tts.yaml
21 automation: !include automations.yaml
22 script: !include scripts.yaml
23 scene: !include scenes.yaml
24 camera: !include camera.yaml
25 alexa: !include alexa.yaml
26 light: !include lights.yaml
27 group: !include groups.yaml
28 template: !include template.yaml
29 plant: !include plant.yaml
30 climate: !include climate.yaml
31 sensor: !include sensor.yaml
32 calendar: !include calendar.yaml
33 waste_collection_schedule: !include waste.yaml
34 utility_meter: !include utility_meter.yaml
35 notify: !include notify.yaml
36 powercalc: !include powercalc.yaml
37 ffmpeg:
38 anniversaries: !include geburtstage.yaml
39 ios: !include iosactions.yaml
40

```

jetzt musst du noch die template.yaml als Datei im selben Ordner wie die configuration.yaml anlegen.

Inhalt für die template.yaml

Als erstes saldieren wir die Werte unseres Zählers (in meinem Fall Shelly 3em pro):

```

- sensor:
  - name: "Power Total"
    unique_id: power_total
    unit_of_measurement: W
    device_class: power
    state_class: measurement
    state: >
      {{
        states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_a_active_power')|float(0) +
        states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_b_active_power')|float(0) +
        states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_c_active_power')|float(0) }}
    availability: >
      {{
        [ states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_a_active_power')|float(0) ,
          states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_b_active_power')|float(0) ,
          states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_c_active_power')|float(0) ] |
        map('is_number') | min }}

```

Du kannst hier auch mehrere Werte anfügen oder reduzieren. Hierfür musst du die Zeilen 9-11 und entsprechend 14-16 anpassen - hier ein kleines Beispiel solltest du nur einen Sensor haben:

```
- sensor:
  - name: "Power Total"
    unique_id: power_total
    unit_of_measurement: W
    device_class: power
    state_class: measurement
    state: >
      {{ states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_a_active_power')|float(0) }}
    availability: >
      {{
        [ states('sensor.shellypro3em_XXXXXXXXXXXXXXXXXXXX_phase_a_active_power')|float(0) ] |
        map('is_number') | min }}
  map('is_number') | min }}
```

Ich empfehle dir hier immer diesen Weg zu gehen weil du bei einem Wechsel der Zähleinheit einfach hier nur deinen Sensor anpassen musst und dein Energydashboard etc. einfach weiterläuft.

Jetzt kommen wir zum Netzbezug und der Netzeinspeisung:

Netzbezug

```
- sensor:
  - name: "Power vom Netz (Watt)"
    unique_id: power_vom_netz
    unit_of_measurement: W
    device_class: power
    state_class: measurement
    state: >
      {% if states('sensor.power_total') | float(0) >= 0 %}
        {{ states('sensor.power_total') | float(0) | round (3) }}
      {% else %}
        {{ (states ('0.0') | float(0)) | round(3) }}
      {% endif %}
    availability: >
```

```
{{
  [ states('sensor.power_total') ] | map('is_number') | min }}
```

Solltest du eine PV Anlage verwenden müssen wir auch noch die Netzeinspeisung konfigurieren
- falls du keine besitzt kannst du einfach zum Punkt Helfer weiterspringen.

Netzeinspeisung

```
- sensor:
  - name: "Power ins Netz (Watt)"
    unique_id: power_ins_netz
    unit_of_measurement: W
    device_class: power
    state_class: measurement
    state: >
      {% if states('sensor.power_total') | float(0) < 0 %}
        {{(states('sensor.power_total') | float(0))*-1 | round (3) }}
      {% else %}
        {{(states ('0.0') | float(0)) |round(3) }}
      {% endif %}
    availability: >
      {{
        [ states('sensor.power_total') ] | map('is_number') | min }}
```

Als nächstes kommt noch die Eigenverbrauchs Berechnung hier musst du :

Eigenverbrauch

```
- sensor:
  - name: "PV Eigenverbrauch (Watt)"
    unique_id: pv_eigenverbrauch_watt
    state_class: measurement
    icon: mdi:transmission-tower
    unit_of_measurement: W
    device_class: power
    state: >
```

```

{% if states('sensor.power_total') | float(0) < 0 %}
    {{ ( (states('sensor.power_total') | float(0) ) + (states('sensor.SENSOR VON DEINER PV
MESSUNG') | float(0) ) ) | round (3) }}
{% else %}
    {{ states('sensor.SENSOR VON DEINER PV MESSUNG') | round (3) }}
{% endif %}
availability: >
{{
    [ states('sensor.power_total'),
      states('sensor.SENSOR VON DEINER PV MESSUNG') ] | map('is_number') | min }}

```

Hier musst du nun in Zeile 12 und 17 deinen Watt-Sensor (Power) von der PV Anlage eintragen

Ersparnis PV Anlage berechnen

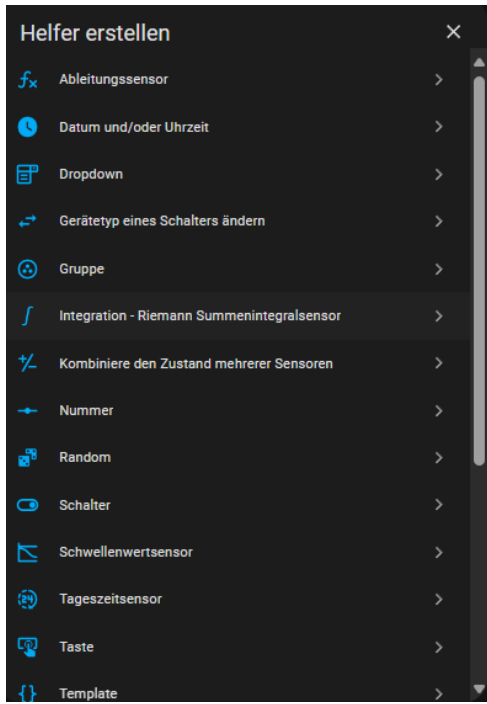
```

- sensor:
  - name: "PV Vorteil"
    unique_id: pv_vorteil
    unit_of_measurement: EUR/h
    state: >
      {{ states('sensor.pv_eigenverbrauch_watt')|float(0) *
states('input_number.strompreis')|float(0)/1000 }}

```

Jetzt legen wir noch ein paar Helfer an:

Helfer



Hier legen wir jeweils einen Integration - Riemann Summenintegralsensor an:

Netzverbrauch =

Riemann Summenintegralsensor hinzufügen



Erstelle einen Sensor, der eine Riemann-Summe berechnet, um das Integral eines Sensors zu schätzen.

Name*

Netzverbrauch

Eingangssensor*

Power vom Netz (Watt)



Integrationsmethode



Trapezregel



Linke Riemannsche Summe



Rechte Riemannsche Summe

Genauigkeit*

3



decimals

Steuert die Anzahl der Dezimalstellen in der Ausgabe.

Metrisches Präfix



k (kilo)



M (mega)



G (giga)



T (tera)

Die Ausgabe wird entsprechend dem gewählten metrischen Präfix skaliert.

Zeiteinheit*

Stunden



Die Ausgabe wird entsprechend der gewählten Zeiteinheit skaliert.

ABSENDEN

Netzeinspeisung =

Riemann Summenintegralsensor hinzufügen



Erstelle einen Sensor, der eine Riemann-Summe berechnet, um das Integral eines Sensors zu schätzen.

Name*

Netzeinspeisung

Eingangssensor*

Power ins Netz (Watt)



Integrationsmethode



Trapezregel



Linke Riemannsche Summe



Rechte Riemannsche Summe

Genauigkeit*

3

decimals

Steuert die Anzahl der Dezimalstellen in der Ausgabe.

Metrisches Präfix



k (kilo)



M (mega)



G (giga)



T (tera)

Die Ausgabe wird entsprechend dem gewählten metrischen Präfix skaliert.

Zeiteinheit*

Stunden



Die Ausgabe wird entsprechend der gewählten Zeiteinheit skaliert.

ABSENDEN

Ersparnis =

Riemann Summenintegralsensor hinzufügen



Erstelle einen Sensor, der eine Riemann-Summe berechnet, um das Integral eines Sensors zu schätzen.

Name*

PV Ersparnis

Eingangssensor*

PV Vorteil



Integrationsmethode



Trapezregel



Linke Riemannsche Summe



Rechte Riemannsche Summe

Genauigkeit*

2

decimals

Steuert die Anzahl der Dezimalstellen in der Ausgabe.

Metrisches Präfix



k (kilo)



M (mega)



G (giga)



T (tera)

Die Ausgabe wird entsprechend dem gewählten metrischen Präfix skaliert.

Zeiteinheit*

Stunden



Die Ausgabe wird entsprechend der gewählten Zeiteinheit skaliert.

ABSENDEN

Jetzt musst du diese Helfer nur noch richtig in deinem Energy Dashboard einfügen und du bist fertig ;)

Energy Dashboard

Hierzu gehen wir auf Einstellungen - Dashboards - Energie



Netzverbrauch konfigurieren

Der Netzverbrauch ist die Energie, die aus dem Energienetz zu deinem Haus fließt.

Wähle einen Sensor, der den Netzverbrauch in GJ, kWh, MJ, MWh, Wh misst.

Verbrauchte Energie
Netzbezug



Wähle aus, wie Home Assistant die Kosten der verbrauchten Energie verfolgen soll.



Kosten nicht verfolgen



Verwende eine Entität, die die Gesamtkosten verfolgt



Verwende eine Entität mit aktuellem Preis

Entität mit dem aktuellen Preis (EUR/kWh)

Strompreis



Verwende einen statischen Preis

ABBRECHEN

SPEICHERN

Netzeinspeisung konfigurieren

Die Netzeinspeisung ist die Energie, die von deinen Solarmodulen ins Netz eingespeist wird.

Wähle einen Sensor, der die Netzproduktion in GJ, kWh, MJ, MWh, Wh misst.

ins Netz eingespeiste Energie
Netzeinspeisung




Bekommst du Geld zurück, wenn du deine Energie ins Netz zurückspeist?

- ☒ Ich bekomme kein Geld zurück
- ☐ Verwende eine Entität, die das gesamte erhaltene Geld verfolgt
- ☐ Verwende eine Entität mit aktuellem Preis
- ☐ Verwende einen statischen Preis


ABBRECHEN

SPEICHERN

jetzt werden bei dir folgende Fehler aufschlagen:

 **statistics_not_defined**

- sensor.netzeinspeisung
- sensor.netzbezug

 **Entität nicht verfügbar**

Der Status dieser konfigurierten Entitäten ist derzeit nicht verfügbar:

- sensor.netzeinspeisung (unknown)
- sensor.netzbezug (unknown)

das ist soweit auch korrekt. Warte einfach bis die Sensoren einiges an Daten gesammelt haben (spätestens nach einer Stunde sollten die Fehler verschwunden sein)

Für ein eigenes Dashboard hab ich euch hier noch ein paar Infos hinterlegt:

Eigene Übersichtsseite

Die Normalen Cards von Homeassistant findet ihr [hier](#)

Ich verwende noch zusätzlich die :

Power Flow Card Plus

Revision #11

Created 1 May 2024 18:26:52 by Cryd

Updated 2 June 2024 12:15:44 by Cryd