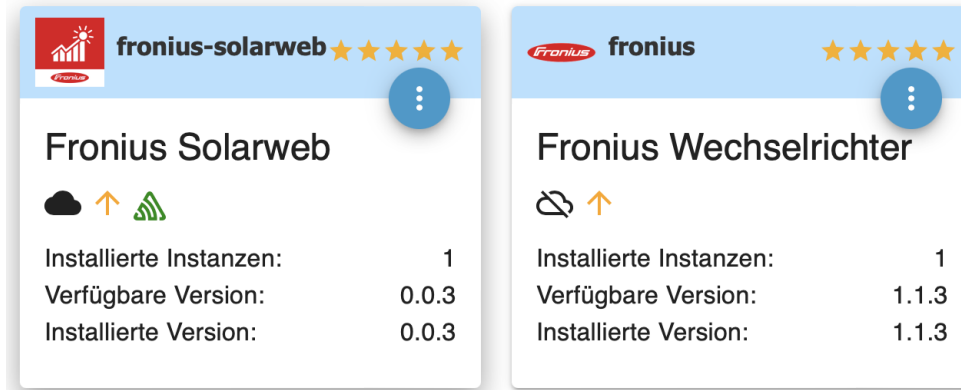


# Visualisierung PV-Anlage auf Basis eines Fronius GEN24 plus

Anfang Februar 2023 wurde unsere PV Anlage in Betrieb genommen. Hier entsteht die Dokumentation zur Visualisierung unserer PV-Anlage, die auf Basis eines Fronius Wechselrichters (GEN24) erfolgt.

Es stehen gleich zwei Adapter für Fronius Geräte zur Verfügung. Basiert der erste auf den Daten auf dem Fronius Server, so greift der zweite ausschließlich auf die lokal verfügbaren Daten zu. Ich habe mich als Anfänger zunächst für den zweiten Adapter entschieden.



## Grundlagen: verwendete Datenpunkte

Hierbei den Überblick zu gewinnen, ist eine echte Herausforderung, zumal Fronius seinerseits kräftig zu Verwirrung beiträgt. Einige scheinen doppelt vorhanden zu sein, während andere gar nicht gepflegt werden. Sind die Leistungsangaben noch leidlich umfassend, so sieht's bei den Energiewerten regelrecht traurig aus.

Die Datenpunktstruktur zeigt die nachfolgenden Abbildungen.

--	--

Aus `fronius.0.inverter.1` werden verwendet ..

- IAC, UAC : Wechselstrom und Spannung vom Inverter
- PAC : vom Inverter gelieferte AC Leistung
- IDC, UDC : Gleichstrom und Spannung aus MPP1
- IDC\_2, UDC\_2 : Strom und Spannung von MPP2
- IDC\_3, UDC\_3 : Gleichstrom aus der Batterie
- TOTAL\_ENERGY : vom Inverter geliefert AC Energie

Leider bleiben `DAY_ENERGY` und `YEAR_ENERGY` leer, werden von Fronius offenbar nicht bereitgestellt.

Auch in dem übergeordneten Verzeichnis bleiben die Datenpunkte E\_Day und E\_Year leer, werden also von Fronius nicht bereitgestellt. Offenbar gilt E\_Total = TOTAL\_ENERGY; diese Datenpunkte liefern die vom Inverter erzeugte Energie auf der Wechselstromseite (AC).

- P\_Akku : Leistung in/aus Batterie
- P\_Grid : Leistungen am Übergabepunkt
- P\_Load : Leistung ins Haus (Verbrauch)
- P\_PV : Leistung aus PV, Gleichstrom DC

Alle weiteren im Folgenden verwendeten Datenpunkte stammen aus dem Smartmeter. Damit die Werte der hier verwendeten Bedeutung entsprechen, muss diese direkt am Übergabepunkt zum Netzanbieter installiert sein.

Für die Visualisierung werden die Ströme (L1, L2, L3 und gesamt) sowie die zugehörigen Spannung verwendet. Darüber hinaus werden noch die zugehörigen Leistungen PowerReal\_P\_Phase\_1, PowerReal\_P\_Phase\_2, PowerReal\_P\_Phase\_3 sowie PowerReal\_P\_Sum verwendet.

EnergyReal\_WAC\_Sum\_Consumed, EnergyReal\_WAC\_Sum\_Produced erfassen die bezogen und ins Netz eingespeiste Energie.

powerflow	detailed information about the power flow
inverter1	
BackupMode	BackupMode <i>BackupMode</i>
BatteryStandby	Battery standby <i>Battery standby</i>
E_Day	pv power day <i>pv power day</i>
E_Total	pv power total <i>pv power total</i>
E_Year	pv power year <i>pv power year</i>
Meter_Location	SmartMeter location <i>Smartmeter location</i>
Mode	Mode <i>Power Flow Mode</i>
P_Akku	akku power <i>akku power</i>
P_Grid	grid power <i>grid power</i>
P_Load	load power <i>load power</i>
P_PV	pv power <i>pv power</i>
rel_Autonomy	pv Autonomy <i>pv autonomy</i>
rel_SelfConsumption	pv SelfConsumption <i>pv self consumption</i>

meter	detailed information about Met
1	meter with device ID 1
Current_AC_Phase_1	AC current L1
Current_AC_Phase_2	AC current L2
Current_AC_Phase_3	AC current L3
Current_AC_Sum	AC current Sum <i>Sum of all currents</i>
Enable	Enable <i>Enable</i>
EnergyReactive_VArAC_Sum_Consumed	SUM REACTIVE ENERGY CONS
EnergyReactive_VArAC_Sum_Produced	SUM REACTIVE ENERGY PROD
EnergyReal_WAC_Minus_Absolute	ACTIVE Energy feed in
EnergyReal_WAC_Plus_Absolute	ACTIVE Energy consumed from
EnergyReal_WAC_Sum_Consumed	SUM ACTIVE ENERGY CONSUM
EnergyReal_WAC_Sum_Produced	SUM ACTIVE ENERGY PRODUC
Frequency_Phase_Average	AVG GRID FREQUENCY
Meter_Location_Current	Meter_Location_Current <i>Meter_Location_Current</i>
Model	ManufacturerModel <i>Manufacturer &amp; Model</i>

## Visualisierung der Leistungsflüsse

Es werden die Leistungsflüsse zwischen Netz/Versorger, PV-Anlage, Batterie und Hausverbrauch erfasst. Hierzu werden aus dem Adapter *Fronius Wechselrichter* die folgende Datenpunkte entnommen ..

- $P_{AC} = \text{fronius.0.inverter.1.PAC} = \text{fronius.0.powerflow.inverter1.P}$
- $P_{PV} = \text{fronius.0.powerflow.P}_{PV}$
- $P_{GRID} = \text{fronius.0.powerflow.P}_{Grid}$
- $P_{LOAD} = \text{fronius.0.powerflow.P}_{Load}$
- $P_{AKKU} = \text{fronius.0.powerflow.P}_{Akku}$

Die Vorzeichen der Leistungen spiegeln die Flussrichtungen wider. Wird Leistung aus dem Netz bezogen, so wird das Icon in rot ansonsten grün dargestellt. Ebenso trifft dies für die Batterie zu. Die Pfeile werden ebenfalls entsprechend der Richtungen der Leistungsflüsse dargestellt.

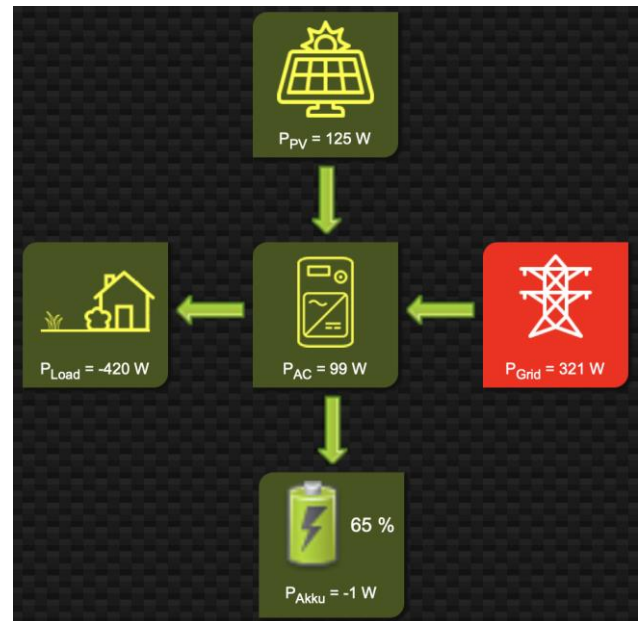
## Konfiguration der Widgets Netz und Batterie

Exemplarisch wird hier das Icon für den Netzzugang dokumentiert. Hierzu wird ein Standardwidget aus *Visicontwo* vom Typ Background verwendet.

Der Datenpunkt  $P_{Grid}$  wird negativ, wenn Leistung ins Netz eingespeist wird. Bei Netzbezug ist der Wert positiv. Als Standardfarbe für den Hintergrund wird rot verwendet. Ist der Wert des Datenpunktes kleiner 0, so wird das Icon grün – also als aktiv dargestellt.

Für die Darstellung der Pfeile und des Batterie-symbols wird auf die Icon-Sammlung *icons-fatcow-hosting* zurückgegriffen.

Aufwendiger wird es beim Icon für den Wechselrichter, da hier neben den beiden Zustände Bereitschaft (standby/sleeping) und aktiv (running) auch berücksichtigt werden muss, dass der Inverter keine Energie erzeugt, sondern im Gegenteil welche verbraucht. Dies tritt immer dann ein, wenn auf der Gleichstromseite (PV und Batterie) zu wenig Energie bereitsteht. Dies sollte zwar nie passieren, da der Inverter in diesem Fall



**Eigenschaften**

View w01028 CSS Skripte

**Generell**

**Sichtbarkeit**

**Allgemein**

**Anleitung:** [Zum Wiki](#)

Typ Info: Soll es ein Switch, State, Navigation oder Hintergrund sein?

Widget Typ: Background

Objekt ID: fronius.0.powerflow.P\_Grid

Typ: value

Typ Info: Wenn Typ ist 'boolean', können Wert Felder leer bleiben

Vergleichsoperator: Kleiner

Wert wahr: 0

Beschriftung falsch:

Beschriftung wahr:

Bild falsch: /vis-icontwo/Electricity/transmission\_tower.png

Bild wahr: /vis-icontwo/Electricity/transmission\_tower.png

**CSS inventwo Widget**

**Hintergrund:**

Farbe: #ff0000

Farbe Aktiv: #455618

Farbe bei Hover:

automatisch in Bereitschaftsstellung (standby) wechsel. Aber die Erfahrung zeigt, dass es eine Weile dauern kann, bis diese Umschaltung erfolgt.

Daher wird hierzu ein sog. Multi-Widget mit drei Zuständen verwendet. Zur Auswahl der aktuellen Darstellung wird ein zusätzlicher Datenpunkt benötigt, der durch seine drei Werte 1, 2 und 3 die gültige Anzeige gewährleistet. Es wird ein Skript verwendet, diesen Datenpunkt mit den jeweils gültigen Werten zu füllen.

Die Konfiguration der Pfeile bedarf hier einer Erweiterung im CSS. Es steht nur ein nach rechts gerichteter Pfeil zu

**Eigenschaften**

View w01023 CSS Skripte

**Generell**

Name: SolarUnten

Kommentar:

CSS Klasse: rotate90

Filterwort:

Zeige in Views: Nur in aktueller View

Inaktiv(locked):

**Sichtbarkeit**

Object ID: fronius.0.powerflow.P\_PV

Bedingung: >

Wert für die Bedingung: 0

Nur für Gruppen: Alle Gruppen

Falls Anwender nicht in der Gruppe: Hirse

**Allgemein**

Quelle: /icons-fatcow-hosting/arrow\_right.png

Strecken:

Updatezeit(ms): 0

Update bei Aufwachen:

Update bei Viewwechsel:

Addiere nicht zu URL:

allowUserInteractions:

**Eigenschaften**

View w01023 CSS Skripte

Global

Global

Projekt

```
1 .rotate90{  
2   transform: rotate(90deg);  
3 }  
4  
5 .rotate180{  
6   transform: rotate(180deg);  
7 }  
8  
9 .rotate270{  
10  transform: rotate(270deg);  
11 }  
12  
13 .greater{  
14  transform: scale(2,1);  
15 }
```

Verfügung. Daher wurde im Visualisierungsprojekt Ergänzungen der CSS vorgenommen, wie sie aus der Abbildung ersichtlich ist.

Für die Darstellung des Pfeiles wird ein Widget aus der Sammlung *basic* verwendet und entsprechend den Erfordernissen konfiguriert. Um die unterschiedlichen Pfeilorientierungen darzustellen, werden zwei entsprechend orientierte Pfeile übereinander dargestellt. Auf den Vorzeichen der Datenpunkte basierende Sichtbarkeitsregeln sorgen für eine korrekte Darstellung der jeweils gewünschten Pfeilrichtung.



Die nötigen Ergänzungen in CSS werden global für das gesamte Projekt bereitgestellt. Neben den Anweisungen der Drehungen (rotateXY) ist hier auch eine zur Skalierung (Streckung) in x- und y-Richtung dargestellt.