

Ebene 0 - StatusQuo

In dieser Ebene werden die meisten Sensoren und Stromzähler eingebunden. Sie enthält alle Komponenten, welche sich nicht direkt beeinflussbar sind.

- Erzeugung und Verbrauch
- Bilanzierung / Saldo
- Vorhersage / Prognose
- GrünstromIndex

Erzeugung und Verbrauch

ebene0_erzeugung_verbrauch.png

Erzeugung

Bei der Referenzimplementierung existieren zwei Quellen zur Stromerzeugung.

Eine **Aufdach-PV-Anlage**, welche bereits initial zur Eigenstromnutzung installiert wurde. Diese Anlage besitzt folgende Elemente:

- SMA Wechselrichter (Tripower)
- SMA Energy Meter für Wirkleistungsbegrenzung
- SMA HomeManager
- Discovery Produktionszähler (Easymeter via Discovery API)

Zur Messung der erzeugten Strommengen wird lediglich der Discovery Zähler genutzt in der Referenzimplementierung genutzt, da die SMA Komponenten insgesamt zu ungenaue Werte lieferten. Die Werte des SMA Energy Meter lassen sich jedoch recht einfach in Node-Red über einen `UDP In` Node integrieren.

Funktion Produktionszähler

```
msg.payload = msg.payload["303fbb8ca6404ebba48c196b4dbbc176"];
return msg;
```

Funktion Wirkleistung

```
msg.payload = Math.round((msg.payload.power1 + msg.payload.power2 + msg.payload.power3) / 1000)
const subSUM=' Generation';

let sum = flow.get(subSUM) * 1;
if(isNaN(sum)) sum = 0;

let previous = context.get("previous") * 1;
if(isNaN(previous)) previous = 0;

if(flow.get("SaldoID") != context.get("SaldoID")) {
    context.set("SaldoID", flow.get("SaldoID"));
} else {
```

```

    sum -= previous;
}
sum += msg.payload;

context.set("previous", msg.payload);
node.status({text: msg.payload + " W"});
flow.set(subSUM, sum);
return msg;

```

Die Discovery API liefert die Leistungswerte je Phase. Da es sich um einen 3-Phasen Wechselrichter handelt, muss aus den Einzelwerten die Summe gebildet werden (Zeile 1). Zum Schluss wird der rohe Wert der Discovery API noch auf die innerhalb von Casa Corrently verarbeitete Dimension (Watt) umgerechnet.

Als Ergänzung existiert eine kleine **Balkon-PV Anlage**. Angeschlossen ist diese Anlage mittels eines auf ZigBee auslesbaren Wechselrichters, da jedoch der Hersteller insolvent ist, können die Daten nicht direkt vom Wechselrichter ausgelesen werden. Stattdessen wird eine für diesen Betrieb modifizierte ein AVM FRITZ!DECT 200 Smart Home Stecker verwendet, welcher über die Fritz-Box API ausgelesen werden kann.

Funktion BalkonPV

```

msg.payload.ain = "087610221618"; // Hier die AIN aus der Fritz Box eintragen
return msg;

```

Funktion Wirkleistung

```

msg.payload = Math.round(msg.payload);
const subSUM='Generation';

let sum = flow.get(subSUM) * 1;
if(isNaN(sum)) sum = 0;

let previous = context.get("previous") * 1;
if(isNaN(previous)) previous = 0;

if(flow.get("SaldoID") != context.get("SaldoID")) {
    context.set("SaldoID", flow.get("SaldoID"));
} else {
    sum -= previous;
}
sum += msg.payload;

```

```
context.set("previous", msg.payload);

flow.set(subSUM, sum);
node.status({text: msg.payload + " W"});
return msg;
```

Verbrauch

Die Verbrauchsmessung für den gesamten Haushalt muss den in **Ebene 1** definierten Speicher herausrechnen. Dies geschieht bei der Referenzimplementierung durch die Betrachtung Ermittlung des Strombezugs am Netzanschluss (**Ebene 2**) sowie die via MQTT verfügbare Entladeleistung des Speichers.

Funktion Wirkleistung

```
msg.payload = msg.payload["781ffa307e434529be9f747eece1b8dc"];

let power = Math.round((msg.payload.power1 + msg.payload.power2 + msg.payload.power3)/1000)

msg.payload = power;

return msg;
```

Trigger

Lediglich die Abfrage der Discovery-API benötigt einen Trigger (Auslöser), damit regelmäßig neue Leistungsdaten vorhanden sind. Der Speicher sowie die Balkon-PV Installation benötigen keinen Trigger, da hier von den Sensoren fortlaufend Messwerte gesendet werden.

Ausgang zum Saldo / Bilanzierung

[linkout_ebene0.png](#) msg.type: unknown

Damit bei jeglicher Aktualisierung der Saldo ebenfalls aktualisiert wird, kommt ein "Link Out" Node zum Einsatz

Bilanzierung / Saldo

[ebene0_saldo.png](#) unknown

In der Ebene 0 wird der Saldo über die **im Flow gespeicherten** Summen der Erzeugung (Generation) und des Verbrauchs (Consumption) gebildet.

Funktion Saldo bilden

```
msg.payload = flow.get(' Consumption' ) - flow.get(' Generation' );  
flow.set(' Saldo' ,msg.payload);  
node.status({text: "P: "+msg.payload+" W"});  
return msg;
```

Die erneute Aufteilung der Einzelwerte dient lediglich der getrennten Speicherung in der InfluxDB.

Vorhersage / Prognose

[ebene0_vorhersage.png](#)

Wie in der [Konzeptbeschreibung zur Prognose](#) aufgeführt, erfolgt die Erstellung der Prognose in zwei Schritten. Zur Vorhersage der Erzeugung wird in der Referenzimplementierung der `esolar` Wert des GrünstromIndex genutzt und mit den tatsächlichen Erträgen der Anlage präzisiert.

[ebene0_vorhersage_grafana.png](#)

Für die Aktualisierung der Prognose wurde ein Intervall von 20 Minuten gewählt, da gerade für die Erzeugung lediglich alle 60 Minuten neue Werte aus dem GrünstromIndex vorliegen und somit eine häufigere Aktualisierung keine besseren Ergebnisse liefert.

GrünstromIndex

[ebene0_gsi.png](#)
Image not found. Type unknown

Basierend auf dem prognostizierten Saldo der Ebene wird ein GrünstromIndex gebildet und zur Visualisierung mittels Grafana in die InfluxDB gespeichert.

Vergleiche: [GrünstromIndex je Ebene](#)

Funktion Ebenen GSI

```
let saldo = flow.get("forecastSaldo");
let min = 9999999999;
let max = -999999999999;

for(let i=0; i<saldo.length;i++) {
    saldo[i].measurement = "e0gsi";
    if(saldo[i].fields.w > max) max = saldo[i].fields.w ;
    if(saldo[i].fields.w < min) min = saldo[i].fields.w ;
}

let delta = max - min;

for(let i=0; i<saldo.length;i++) {
    saldo[i].fields.gsi = 100-Math.round(((saldo[i].fields.w - min) / delta)*100);
}

flow.set("gsi", saldo);
global.set("e0gsi", saldo);
msg.payload = saldo;

return msg;
```

[ebene0_gsi_prognose.png](#)
Image not found. Type unknown