Mit dem poweropti Energiezähler auslesen und in das Smart-Home-System Homematic CCU3 integrieren

Wie schon im ELVjournal 2/2021 im Fachbeitrag „Klein, aber oho! – Einfach und komfortabel Energiezähler auslesen“ beschrieben, bietet der poweropti – ein kleines Aufsteckmodul für die optische Schnittstelle moderner Energiezähler – die Möglichkeit, die Zählerwerte unkompliziert auszulesen. Die mit dem poweropti empfangenen Daten lassen sich per Smartphone-App anzeigen und können per API-Schnittstelle über das Internet ausgelesen und in eigene Smart-Home-Systeme integriert werden.

Leserwettbewerb

*siehe Seite 121

Thomas Mühlbayer
hat für seinen Beitrag zum Leserwettbewerb einen Gutscheincode* über 200,– Euro erhalten!

Dieser Beitrag beschreibt die Verknüpfung des poweropti mit der CCU3 von Homematic. Die Momentanleistung (aktueller Verbrauch bzw. aktuelle Einspeisung in Watt) wird abgerufen und in einer Systemvariablen gespeichert. Mit Homematic IP Schaltsteckdosen oder Aktoren können Energieverbraucher dann so gesteuert werden, dass sie nur einschalten, wenn genug „Sonnenstrom" vorhanden ist.

Installation des poweropti und Freischalten des Stromzählers

Nach der Installation der Powerfox App auf dem Handy (Bild 1) wird die Registrierung des Moduls durchgeführt und der poweropti am Zähler befestigt (siehe Fachbeitrag ELVjournal 2/2021). Dazu befindet sich am Zähler eine kleine Stahlplatte und am poweropti ein passender Magnetring mit zusätzlicher ringförmiger Klebefolie.

Es gibt sehr viele verschiedene digitale Stromzähler. In der Whitelist von Powerfox sind die kompatiblen Zähler aufgeführt. Die Zähler zeigen in der Regel nur die Zählerstände an, d. h. die bezogenen und eingespeisten Energiemengen in kWh (Bild 2).

Bild 2: Digitaler Stromzähler mit optischer Schnittstelle. Darüber befindet sich eine graue Taste für die Eingabe der PIN.

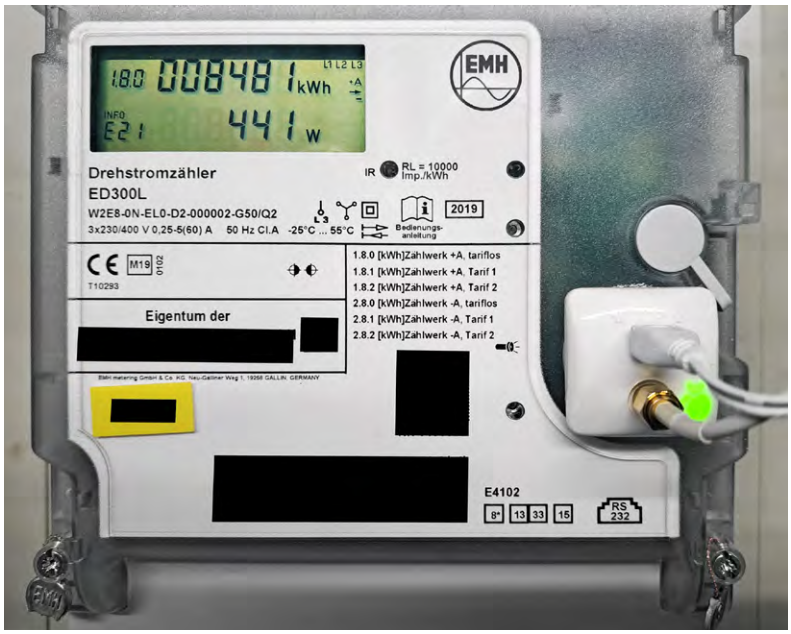

Bild 3: Digitaler Zähler mit auf der optischen Schnittstelle montiertem poweropti. Das Modul hält durch einen Magnetring und zusätzlichen Klebering.

Bild 1: poweropti mit Darstellung der Daten in der Smartphone-App

Die Anzeige der aktuellen Leistungswerte in Watt muss erst freigeschaltet werden (Bild 3). Dafür ist eine vierstellige PIN erforderlich, die beim Netzbetreiber angefordert werden kann. Nach Eingabe dieser PIN muss im Zählermenü die erweiterte Anzeige (INF ON) aktiviert werden. Bei dem in Bild 2 und 3 dargestellten Zählern konnte die Anzeige damit aber nur temporär aktiviert werden. Bei diesem Zählertyp musste ein Servicemitarbeiter des Netzbetreibers vor Ort die permanente Freischaltung durchführen.

Abfrage der Daten

Der poweropti liefert die vom Stromzähler abgelesenen Werte nicht direkt, sondern schickt sie in kurzen Abständen an die Server von Powerfox. Die Powerfox App holt die Werte aus dieser Cloud. Sie können aber auch per Browser im Internet über diese Adresse abgerufen werden. Die Webseite verlangt eine Authentifizierung (Bild 4).

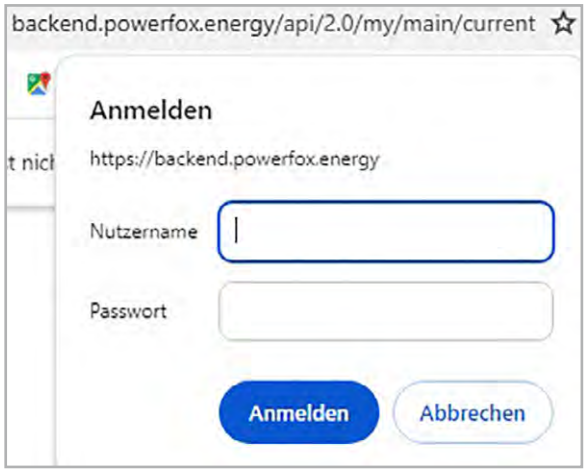

Bild 4: Abfrage des poweropti über die Cloud

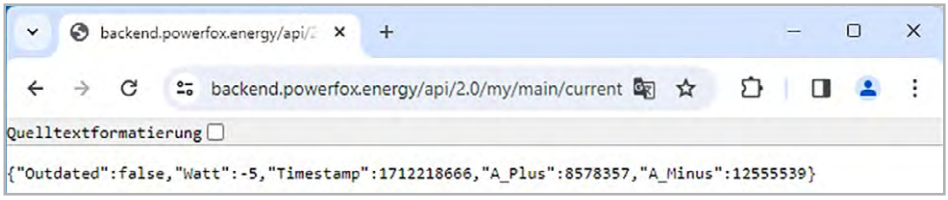

Bild 5: Antwort des Powerfox-Servers mit dem aktuellen Leistungswert in Watt

Gibt man hier den bei der Registrierung des poweropti vereinbarten Nutzernamen und das Passwort ein, erhält man u. a. den aktuellen Verbrauch bzw. die aktuelle Einspeisung (Bild 5). Das Ausgabeformat ist JSON und die Ausgabe kann z. B. so aussehen:

```
{"Outdated":false,"Watt":-811,"Timestamp":1710662976,
"A_Plus":8483780,"A_Minus":12391259}
```

Der Ausgabetext enthält die Leistung in Watt. Bezug wird positiv, Einspeisung negativ dargestellt. Für die Steuerung eines Verbrauchers in Abhängigkeit von der bereitstehenden Photovoltaikenergie braucht es einen Zugriff auf diesen Leistungswert. Ein Script ist notwendig, das Nutzername und Passwort der Webseite übergibt und die Wattzahl anschließend aus dem Ausgabetext isoliert.

Automatisierung der Datenabfrage

Der Aufruf einer Webseite wird in der Regel von einem Browser gemanagt. Er leitet die Anfrage an den entsprechenden Server weiter und verarbeitet die Antworten des Servers. Wenn ein Script eine Webseite aufrufen soll, muss es auch dieses Management übernehmen. Bei der Übergabe von Nutzername und Passwort machen Umlaute und Sonderzeichen Probleme. Deshalb müssen sie vorher in Base64 umkodiert werden. Base64 ist eine Umwandlung von 8-Bit-Code in 7-Bit-Code. Aus jeweils drei Zeichen mit Umlauten und Sonderzeichen werden vier druckbare ASCII-Zeichen. Verschiedene Webseiten wie zum Beispiel base64.guru bieten online eine Umwandlungsmöglichkeit. Der Nutzername bei Powerfox ist eine E-Mail-Adresse. Diese muss – durch einen Doppelpunkt getrennt – zusammen mit dem Passwort umgewandelt werden (Bild 6). So wird z. B. aus *muster@server.de:passwort* dann *bXVzdGVyQHNlcnZlci5kZTpwYXNzd29ydA==* .

Wie können nun diese Autorisierungsangaben zusammen mit der Webadresse übermittelt werden? Dazu dient das Kommandozeilenprogramm „curl“. Es ist in den gängigen Betriebssystemen (Windows, MacOS, Linux) implementiert und somit auch in CCU3-Scripten verfügbar. Bild 7 zeigt beispielhaft den Aufruf im cmd-Fenster von Windows. Damit kann der curl-Aufruf vorab getestet werden. Der Base64-Code aus dem Beispiel muss natürlich durch den Code mit den eigenen Autorisierungsdaten ersetzt werden. Im CCU3-Script wird „system.Exec()“ verwendet, um Betriebssystembefehle auszuführen.

```
string stdout;
string stderr;
string s_cmd = "curl -H \"Authorization: Basic  bXVzdGVyQHNlcnZlci5kZTpwYXNzd29ydA== \"
https://backend.powerfox.energy/api/2.0/my/main/current";
system.Exec(s_cmd, &stdout, &stderr);
```

Die Variablen „stdout“ und „sterr“ braucht „system.Exec“ für die Speicherung der Rückmeldungen bzw. Ausgaben des curl-Befehls. Die Variable „s_cmd“ enthält den Befehl. Da doppelte Anführungszeichen den String einschließen, müssen die doppelten Anführungszeichen im curl-Befehl hier mit einem Backslash (\) gekennzeichnet werden.

Nach der Ausführung dieser Scriptzeilen steht in der Variablen „stdout“ die Ausgabe der Webseite, also z. B.

```
{"Outdated":false,"Watt":-811,"Timestamp":1710662976,
"A_Plus":8483780,"A_Minus":12391259} .
```

Aus diesem String wird der Wattwert (hier -811) extrahiert und in einer Systemvariablen gespeichert.

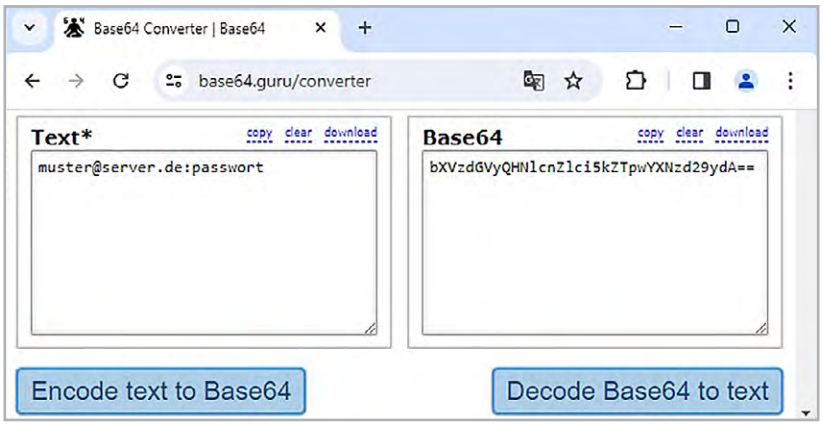

Bild 6: Kodierung von Nutzername:Passwort in Base64

Bild 7: Abfrage mit curl im Windows-cmd-Fenster. Der Parameter -H weist curl an, mit der URL einen „Header" zu übermitteln – hier den Authorization-Code.

Speichern des Wattwerts in der Systemvariablen „aktuelle_Leistung"

Zunächst werden die beiden Systemvariablen „aktuelle_Leistung_0" und „aktuelle_Leistung" entsprechend Bild 8 in der CCU3 angelegt (Einstellungen → Systemvariable → Neu). Dann wird ein Programm erstellt, das den aktuellen Wattwert bestimmt, indem es diesen aus dem Ausgabetext extrahiert und in der Systemvariablen „aktuelle_Leistung_0" speichert (Programme und Verknüpfungen → Programme & Zentralenverknüpfungen → Neu). Das Programm wird zwischen 7 und 22 Uhr alle 5 Minuten ausgeführt. Die Variable „aktuelle_Leistung_0" wird verwendet, weil die Abfrage der Leistung manchmal fälschlicherweise den Wert 0 liefert. In einem zweiten Programm wird dies geprüft und der Wert nur dann an die Variable „aktuelle_Leistung" weitergegeben, wenn dieser ungleich 0 ist (Bild 9 bis Bild 11).

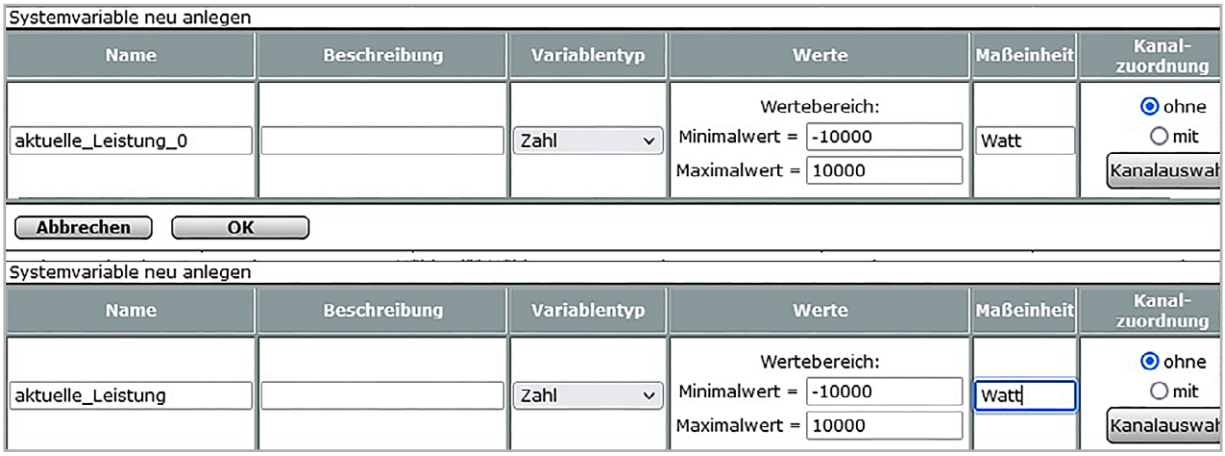

Bild 8: Anlegen der Systemvariablen "aktuelle_Leistung_0" und "aktuelle_Leistung"

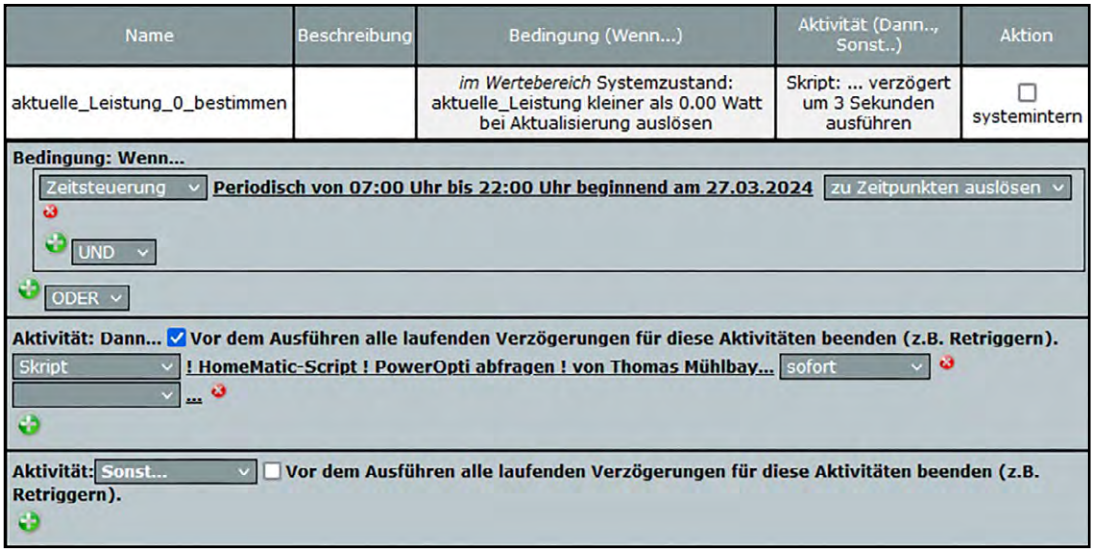

Bild 9: Programm zur poweropti-Abfrage

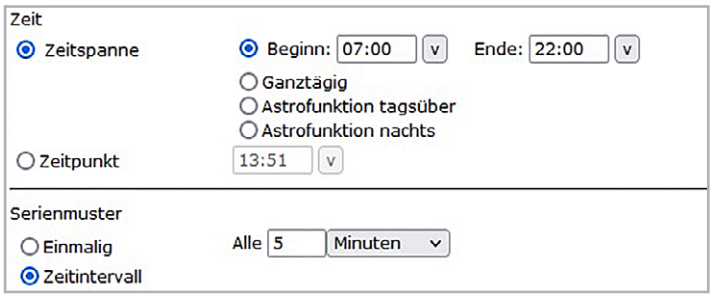

Bild 10: Einstellungen der Zeitsteuerung

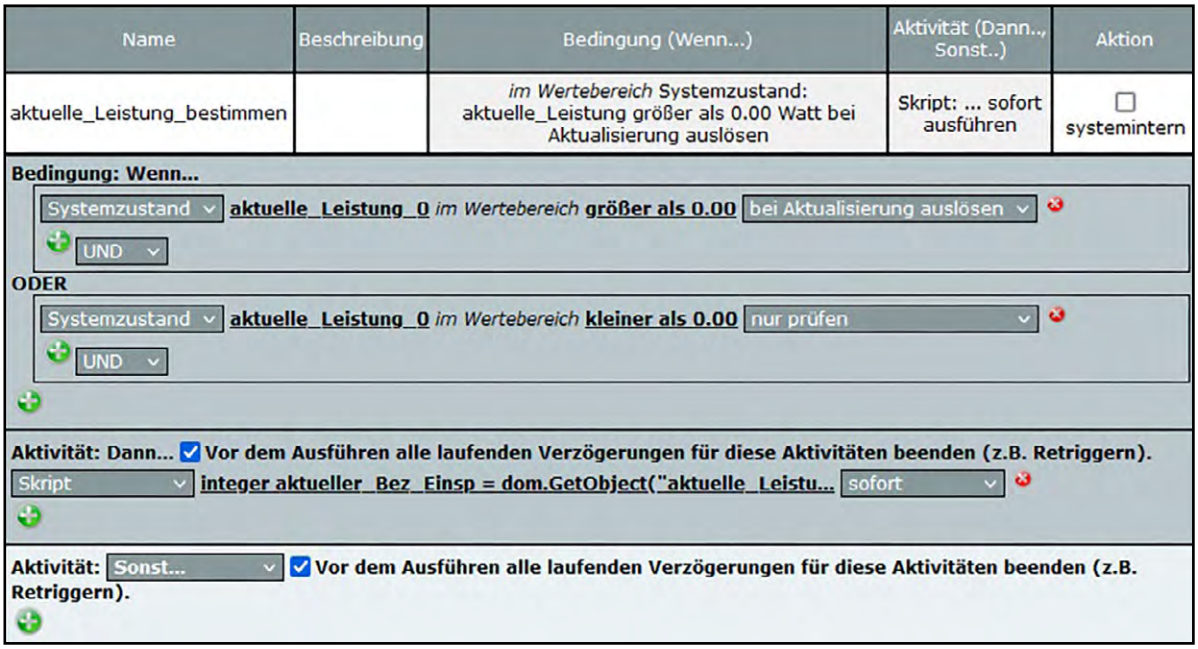

Bild 11: Programm speichert nur den Wert ungleich 0

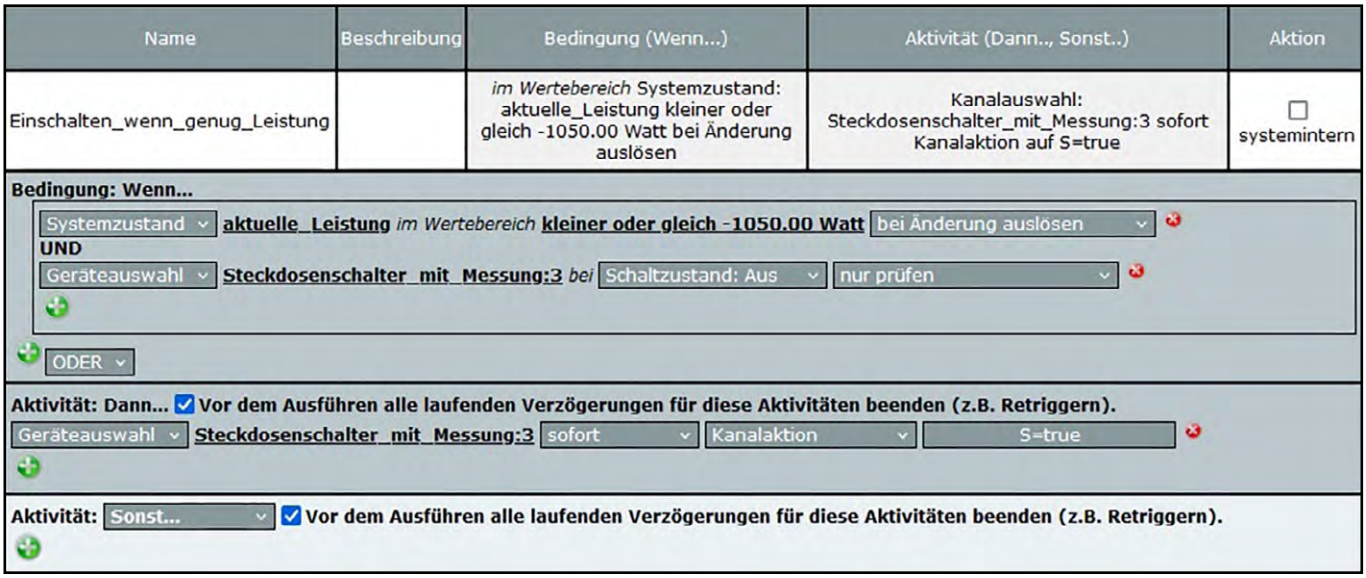

Bild 12: Schaltsteckdose einschalten, wenn sie aus ist und genug Leistung vorhanden ist

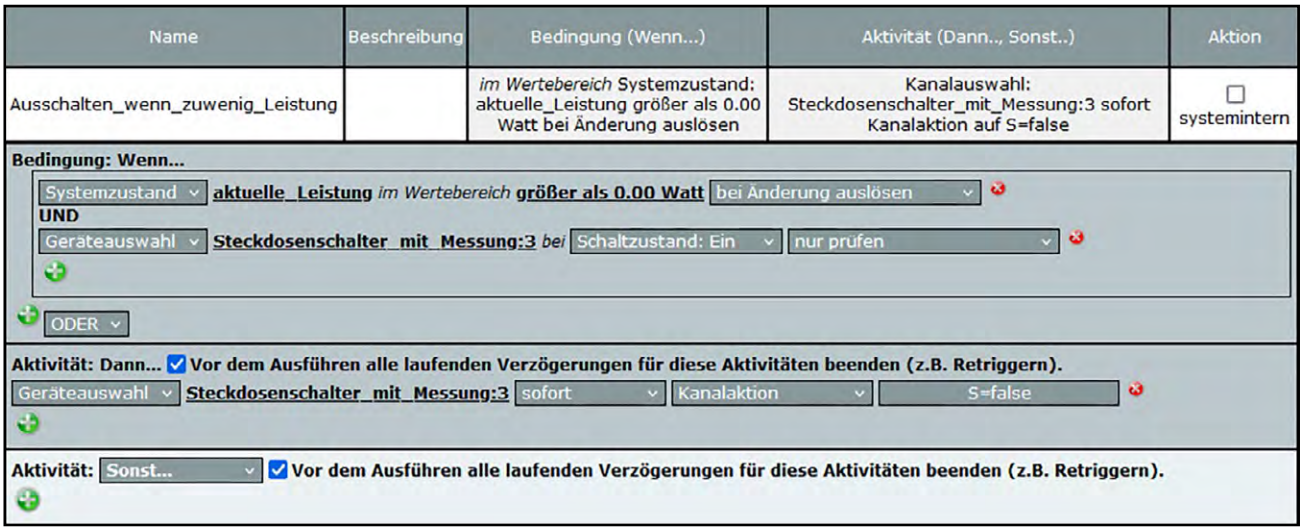

Bild 13: Schaltsteckdose ausschalten, wenn sie an ist und zu wenig Leistung vorhanden ist.

Die vollständigen Scripte zu den beiden Programmen sind:

```
! HomeMatic-Script
! PowerOpti abfragen von Thomas Mühlbayer
! Den Authorization-Code durch eigenen ersetzen!

string stdout;
string stderr;
string s_cmd = "curl -H \"Authorization: Basic bXVzdGVyQHNlcnZlci5kZTpwYXNzd29ydA==\"
https://backend.powerfox.energy/api/2.0/my/main/current";
system.Exec(s_cmd, &stdout, &stderr);

! In der String-Variablen stdout wird die Position des Wortes „Watt" bestimmt.
! Ab dieser Position wird ein Teilstring der Länge 15 rauskopiert
! und in der Variablen teilstring gespeichert.

integer pos_Watt = stdout.Find("Watt");
string teilstring = stdout.Substr(pos_Watt,15);

! In diesem teilstring werden die Positionen der Zeichen ":" und "," bestimmt.
! Dazwischen steht der aktuelle Wattwert des Zählers.

integer pos_anf = teilstring.Find(":") + 1;
integer pos_end = teilstring.Find(",");
integer laenge = pos_end - pos_anf;

! Der aktuelle Wattwert wird aus teilstring rauskopiert und in eine Zahl umgewandelt.

string Wattstring = teilstring.Substr(pos_anf,laenge);
integer aktuelle_Lstg = Wattstring.ToInteger();

! Der Wattwert (die aktuelle Leistung des Stromzaehlers)
! wird in der Systemvariablen aktuelle_Leistung_0 abgelegt.

dom.GetObject ("aktuelle_Leistung_0").State (aktuelle_Lstg);
```

```
! Systemvariable aktuelle_Leistung_0 auslesen
integer aktueller_Bez_Einsp = dom.GetObject("aktuelle_Leistung_0").State()

! und in Systemvariable aktuelle_Leistung kopieren
dom.GetObject("aktuelle_Leistung").State(aktueller_Bez_Einsp)
```

Schalten eines Verbrauchers in Abhängigkeit von der Photovoltaikleistung

Beispielhaft wird hier eine Homematic IP Schaltsteckdose verwendet, um eine 1000-W-Poolheizung zu schalten. Im Programm laut Bild 12 wird die Schaltsteckdose bei Überschussleistung ab 1050 W eingeschaltet. Da der Wert bei Einspeisung negativ ist, muss die Abfrage „kleiner als -1050" lauten. Um unnötigen Funkverkehr zu vermeiden, wird aber nur dann ein Schaltbefehl gesendet, wenn die Steckdose nicht schon an ist. Das Programm laut Bild 13 schaltet entsprechend wieder aus, wenn nichts mehr eingespeist wird (Leistung 0 W). Warum wird die Einschaltschwelle um 50 W höher definiert als die Heizleistung? Dies soll unnötiges Hin- und Herschalten verhindern. Erreicht die Einspeiseleistung 1050 W, schaltet die Heizung ein. Die resultierende Einspeiseleistung sinkt auf 50 W. Die Heizung bleibt an. Erst wenn die Einspeiseleistung um weitere 50 W auf 0 W absinkt, wird sie wieder ausgeschaltet. Man kann den Einschaltschwellenwert auch höher setzen (z. B. auf 1100 W oder 1200 W). Die Heizung schaltet dann zwar erst bei der entsprechenden höheren Solarleistung ein, die Anzahl der Schaltvorgänge wird aber weiter reduziert.

Fazit

Die Implementierung der Leistungsabfrage des poweropti braucht einen gewissen Aufwand. Dann liefert das System aber zuverlässig den Wert des Überschusses der Photovoltaik. Damit sind der Nutzung der Überschussleistung durch Verbraucher wie Elektroboiler oder Ladeeinrichtungen keine Grenzen gesetzt. ELV

* Tagesaktueller Preis unter der Artikel-Nr. im ELVshop