

SOFTWARE

WECHSELRICHTER UND STROMZÄHLER MIT FRITZ!BOX VERBINDEN

17.01.2022

Es wäre interessant, die Daten der Photovoltaik zur Steuerung von Steckdosen zu nehmen, um Geräte nur bei Überschuss anzuschalten. Und zwar ohne teure extra Hardware oder einen Smartmeter, der auch noch alle Haushaltsdaten nach aussen funkt quasi als Einladung an potentielle Einbrecher ...

Die Kopplung geht einfacher als gedacht. Mein [Fronius Wechselrichter](#) ist im Heimnetz unter http://192.168.187.1/solar_api/v1/GetPowerFlowRealtimeData.fcgi zu erreichen. Die aktuelle Wattzahl kommt im json Format.

```
{
  "Body" : {
    "Data" : {
      "Inverters" : {
        "1" : {
          "DT" : 105,
          "E_Day" : 112,
          "E_Total" : 2623.000244140625,
          "E_Year" : 2623,
          "P" : 399
        }
      },
      "Site" : {
        "E_Day" : 112,
        "E_Total" : 2623.000244140625,
        "E_Year" : 2623,
        "Meter_Location" : "unknown",
        "Mode" : "produce-only",
        "P_Akku" : null,
        "P_Grid" : null,
        "P_Load" : null,
        "P_PV" : 399,
        "rel_Autonomy" : null,
        "rel_SelfConsumption" : null
      },
      "Version" : "12"
    }
  },
  "Head" : {
    "RequestArguments" : {},
    "Status" : {
      "Code" : 0,
      "Reason" : "",
      "UserMessage" : ""
    },
    "Timestamp" : "2022-01-16T08:53:50+01:00"
  }
}
```

Die FRITZ!Box kann die Daten zwar im Augenblick nicht verarbeiten (AVM hat dies aber als Produktvorschlag akzeptiert), [dafür kann die FRITZ!Box eine Steckdose](#) schalten. Also brauchen wir doch einen zweiten Rechner zur Überbrückung. Ich nehme dafür einen [Rasp-berry Pi Zero](#) (der Pi Zero 2 ist vergriffen, aber auch nicht notwendig) am besten in der Ausführung WH mit Pinleiste. Stromversorgung kommt von einem Netzteil in Nähe des Sicherungskasten/Stromzählers, zum Test über 3 Tage reicht aber auch eine übliche Powerbank.

Zunächst wird das Pi [OS](#) auf eine alte micro SD Karte installiert und die Karte in den Zero gesteckt. Sobald der Zero dann im WLAN erscheint, kann man ihn [headless](#) d.h, ohne Peripheriegeräte via SSH weiter konfigurieren. Als Messeinheit brauchen wir noch einen [USB IR Lesekopf](#), der am Stromzähler das Blinken der Diode ausliest.

Mit einem cronjob werden im Minutenabstand die Daten abgeholt. Die Abfrage des [Lo-garex](#) Wechselstromzählers funktioniert dabei mit

```
stty -F /dev/ttyUSB0 300 -parodd cs7 -cstopb parenb -ixoff -crtcts
-hupcl -ixon -opost -onlcr -isig -icanon -iexten -echo -echoe -echoctl
-echok
echo -n -e '\x2F\x3F\x21\x0D\x0A' >> /dev/ttyUSB0
timeout 10s cat /dev/ttyUSB0
```

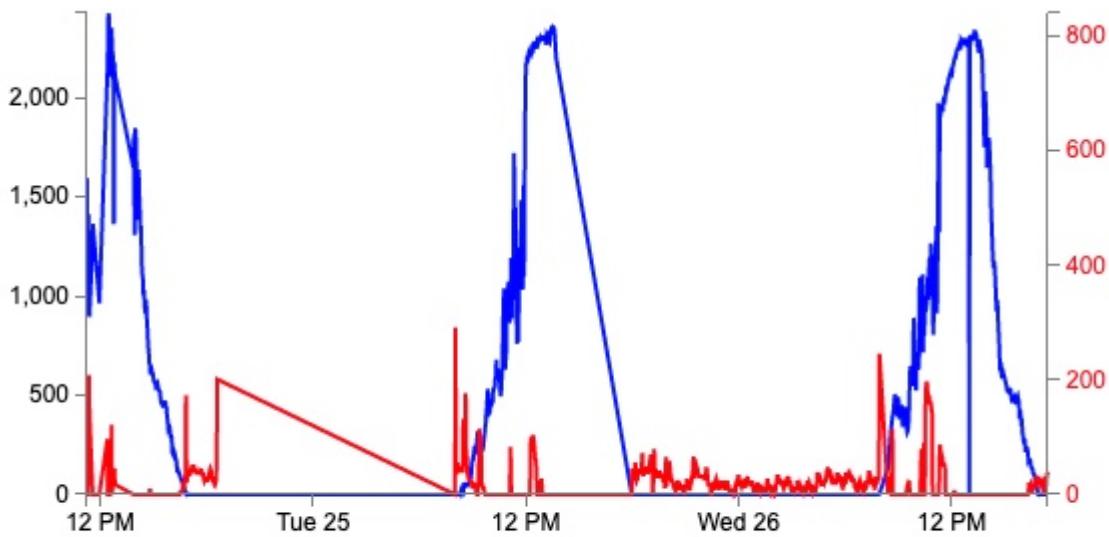
Ab einer bestimmten Leistung der PV kann dann die Steckdose eingeschaltet werden, im Prinzip mit

```
watt=$(curl
http://192.168.187.1/solar_api/v1/GetPowerFlowRealtimeData.fcgi | jq
'.Body.Data.Inverters[1].P')
if [[ $watt -gt 200 ]]
then
http://fritz.box/webservices/homeautoswitch.lua?switchcmd=setsimpleonoff&ain=13077%200012360-1&onoff=1
```

Die Hardware passt in ein kleines Gehäuse, die Kosten liegen unter 50€, dazu kommt noch die schaltbare Steckdose.

Update 26.1.22

Der Abfragecode ist nun in ein PHP Skript verpackt und gibt die Daten in einer hübschen [d3.js Grafik](#) aus.

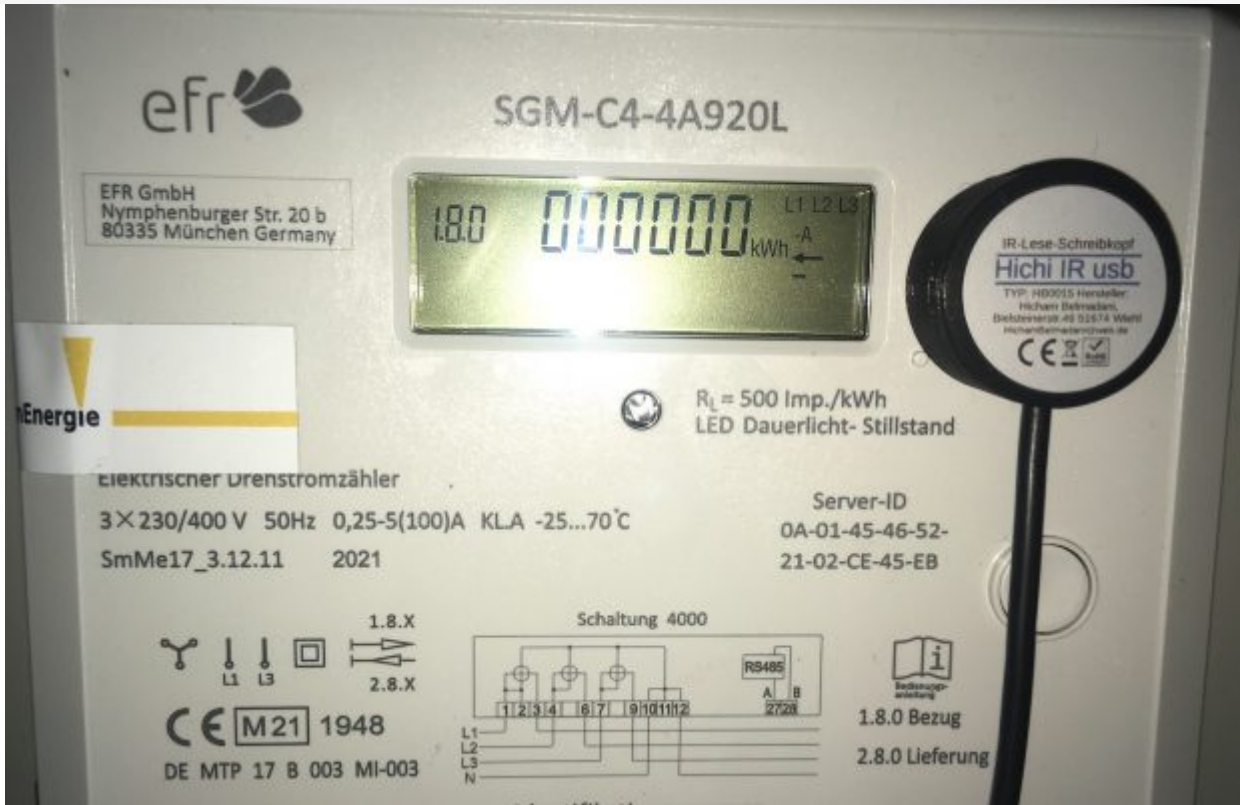


Vorhersage für 18:00
deg=0 uvi=0 vis=10000 few clouds

Im Augenblick läuft die Datensammlung inklusive regelmässiger [openweathermap](#) Abfrage um später ein [R-CNN](#) zur Vorhersage zu programmieren.

Update 23.2.22

Der Lesekopf funktionierte einwandfrei mit dem Drehstromzähler Logarex LK13BDxxxx obwohl die Shell Kommandos nur sehr primitiv in ein PHP Skript eingebaut waren.



Mit dem Einbau eines neuen bidirektionalen Smartmeters [SGM-C4-xxxxxx](#) durch unseren Stromversorger funktioniert das leider nicht mehr. Problem ist nicht etwa, daß die Kommunikation nun mit 9600/8/N/1 läuft, das Problem ist daß die binäre SML Zeichen, die nun jede Sekunde kommen, nicht richtig interpretiert werden.

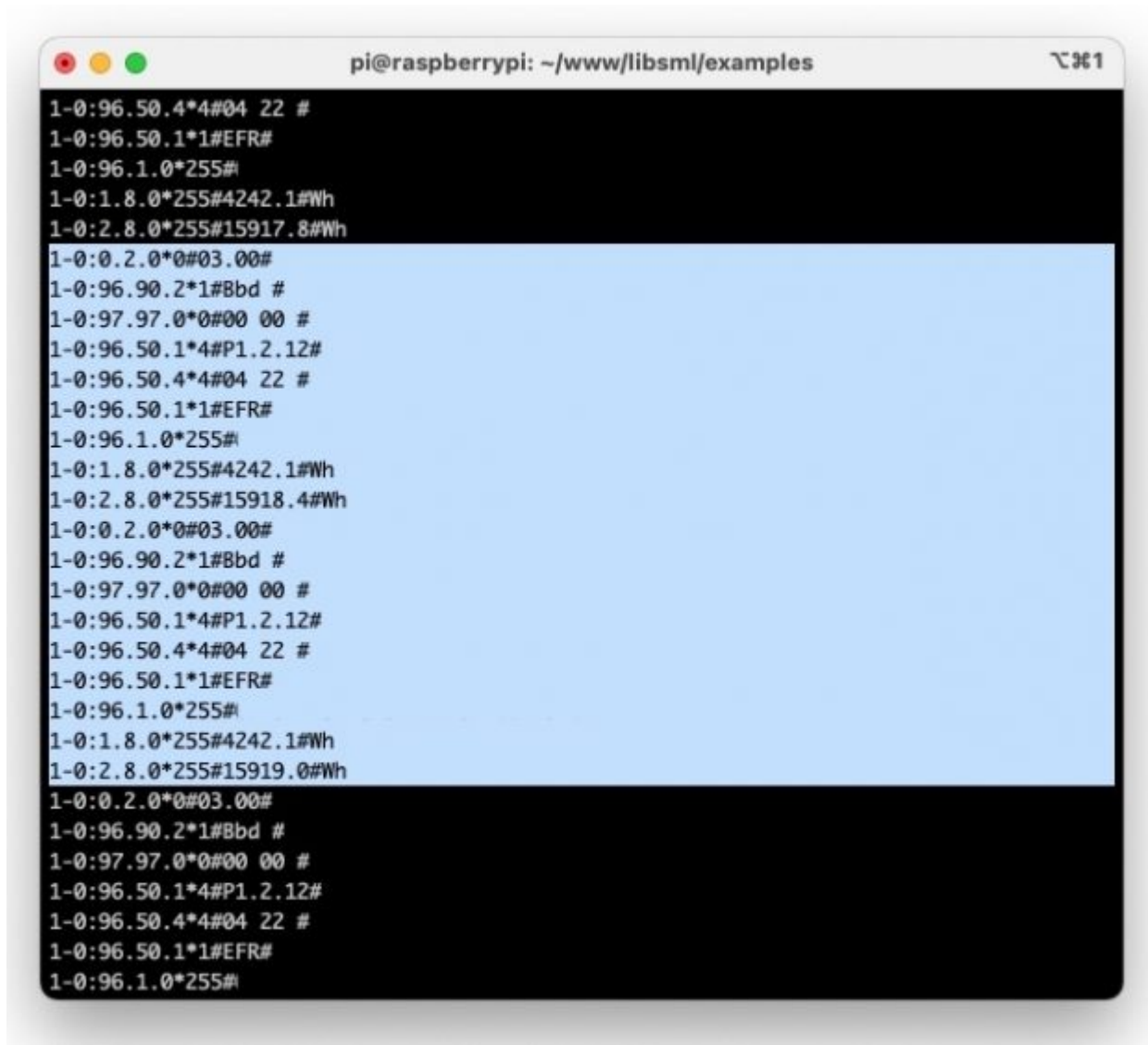
```
stty -F /dev/ttyUSB0 9600 cs8 -cstopb -parenb
timeout 2s cat /dev/ttyUSB0 | od -tx1
```

Python downgrade für den [php_loader](#) machte wenig Sinn, node liess sich auch nicht auf dem Zero installieren und [smeterd](#) brach leider mit einer Fehlermeldung ab. Einzig mit [libsm1](#) hatte ich aber Erfolg

```
apt-get install uuid-dev uuid-runtime
git clone https://github.com/volkszaehler/libsm1
cd libsm1
make
examples/sml_server -s /dev/ttyUSB0
```

Die -s Option ist dabei undokumentiert, aber wichtig denn nur sie findet genau einen Datensatz. Aus [Zeilen 1.8.0 und 2.8.0](#) lässt sich über einen regex Ausdruck der Zählers-

tand auslesen.



Update 2.3.22

Auf Anfrage kann ich auch das komplette Messgerät verleihen, falls jemand denselben Zähler testweise auslesen oder die SD Karte klonen möchte. Wenn sich der Zero nicht das eigenes WLAN findet, [startet er einen eigenen Hotspot](#), ist damit also autonom überall verwendbar.

Update 6.9.22

Läuft und läuft ... Allerdings boote ich nun einmal in 24 Stunden per crontab , da sich manchmal nach 10 Tagen der Zero aufhängt.

```
0 1 * * * shutdown -r now
```

Update 8.11.22

Läuft nun seit 10 Monaten ohne einen Ausfall, genauso wie der [Gaszähler](#).

CC-BY-NC Science Surf , accessed 03.05.2026, [click to save as PDF](#)
