

# Solaranzeige.de

Ab Software Version 4.5.7 Stand Februar 2020

## Steuerbefehle an Wechselrichter / Regler schicken mit Hilfe der Web-Oberfläche.

In dieser Tabelle kann man erkennen, welche Geräte zur Zeit unterstützt werden:

| Regler | Für welche Geräte  | Dateiname              | HTTP-Befehle  |
|--------|--|------------------------|---|
| 3      | Laderegler Tracer Serie  | tracer_regler.php      | LOAD_ON<br>LOAD_OFF   |
| 5      | AEconversion Micro-Wechselrichter<br><br># sind Platzhalter. Siehe unten.                              | aec_wechselrichter.php | L### (0-100 %)<br>B0<br>B2## (Spannung V)<br>S##(Strom A)   |
| 7      | Steca Solarix PLI 5000-48, Effekta AX Serie, Voltronic Axpert Serie, MPPSolar PIP Serie und Baugleiche | ax_wechselrichter.php  | POP00<br>POP01<br>POP02   |
| 9      | MPPSolar MPI Hybrid 3 Phasen   | mpi_3phasen_serie.php  | EDA0<br>EDA1<br>EDB0<br>EDB1<br>EDC0<br>EDC1<br>EDD0<br>EDD1<br>EDE0<br>EDE1<br>EDF0<br>EDF1<br>EDG0<br>EDG1<br>EDH0<br>EDH1<br>SEP00<br>SEP01<br>SEP02<br>DM058<br>DM108 |
| 13     | Joulie-16 Batterie Management System   | joulie_16_bms.php      | RELAY_ON<br>RELAY_OFF<br>STOP_BALANCING<br>START_BALANCING  |
| 29     | go-e Charger Wallbox   | go-e_wallbox.php       | alw_0<br>stp_#<br>amp_##<br>dwo_###   |

| Regler | Für welche Geräte     | Dateiname          | HTTP-Befehle  |
|--------|-----------------------|--------------------|---|
| 30     | Keba Wallbox          | keba_wallbox.php   | ena_0<br>curtime_#####<br>curr_#####<br>setenergy_##### |
| 33     | Labornetzteil DPM8624 | labornetzteil.php  | 10_#####<br>11_#####<br>12_#                            |
| 35     | Wallbe Wallbox        | wallbe_wallbox.php | start_#<br>amp_#####<br>unlock_#                        |

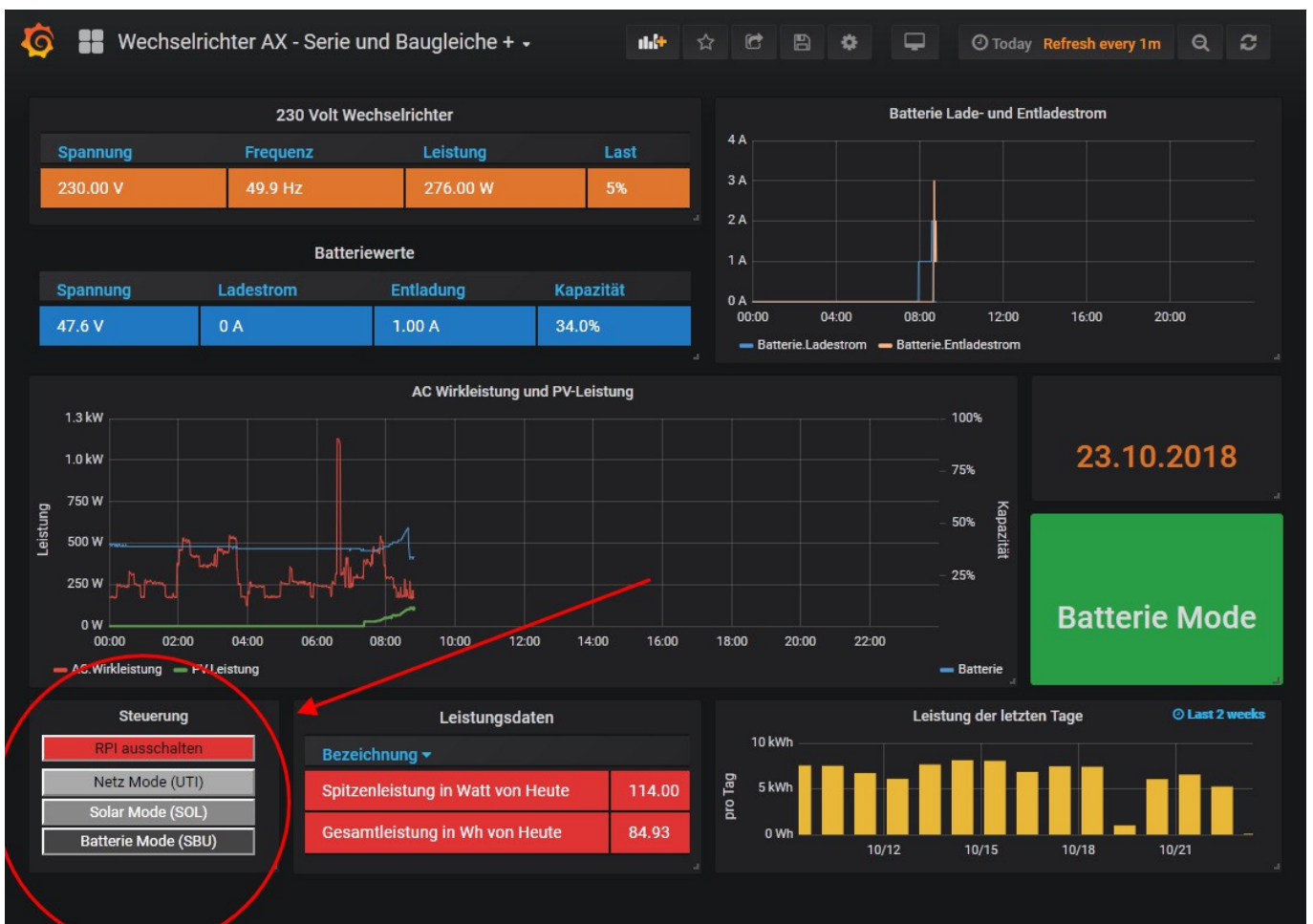
Die Befehle per MQTT werden getrennt gesendet. Befehl + Wert [Befehl = POP Wert = 00, 01 oder 02]

Beispiel anhand Regler 7 (Axpert Serie von verschiedenen Herstellern)

### Regler 7 (AX-Serie):

Damit kann man den Output Mode des Wechselrichters ändern. Diese 3 Befehle sind als Beispiel zu sehen, wie man die Befehle erweitern kann. In diesem Dokument wird die Erweiterung genau beschrieben. Es funktioniert aber nur mit den Geräten, die in der Tabelle stehen. ( Die Tabelle wird kontinuierlich erweitert )

Das dazugehörige Dashboard mit den Button's:

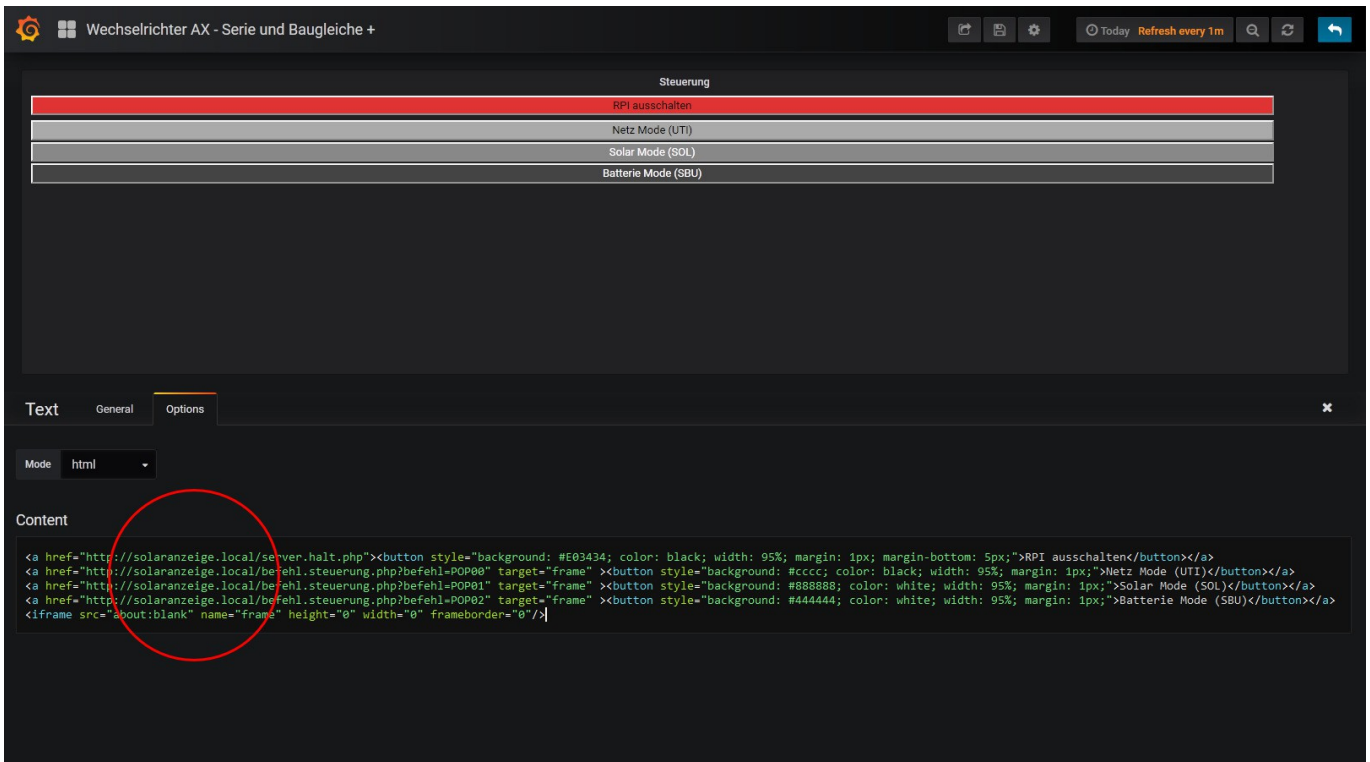


Wie funktioniert es?

Der Button auf dem Desktop ruft die Datei „befehl.steuerung.php“ auf und übergibt einen Parameter an diese Datei. Der Parameter heißt z.B. befehl=POP00. Durch diesen Aufruf wird im Verzeichnis /var/www/pipe eine Datei mit dem Namen „befehl.steuerung“ erzeugt, in dem ein oder mehrere Befehle gespeichert werden. Jeder Befehl in einer Zeile. In diesem Fall „POP00“ Der URL Pfad kann „localhost“ oder „solaranzeige.local“ oder aber auch die IP Adresse des Raspberry sein. „localhost“ funktioniert nur auf dem Raspberry selber. „solaranzeige.local“ funktioniert wenn der Router mDNS unterstützt, im gesamten lokalen Netz. Die IP Adresse funktioniert auch im gesamten lokalen Netz und sollte benutzt werden, wenn „solaranzeige.local“ nicht funktioniert.

Analog dazu funktioniert es auch mit den anderen oben angegebenen Wechselrichtern. Wenn es keine Buttons auf dem Standard Dashboard gibt, muss die Umschaltung mit http URL's oder mit CURL Befehlen erfolgen.

Man kann selber Buttons in sein Dashboard einbauen. Wie das geht, steht hier:



Die IP Adresse kann man am schnellsten mit dem Aufruf auf der Command Ebene „hostname -I“ heraus bekommen.

Möchte man weitere Befehle einbauen, so müssen weitere Button's im Grafana Dashboard angegeben werden.

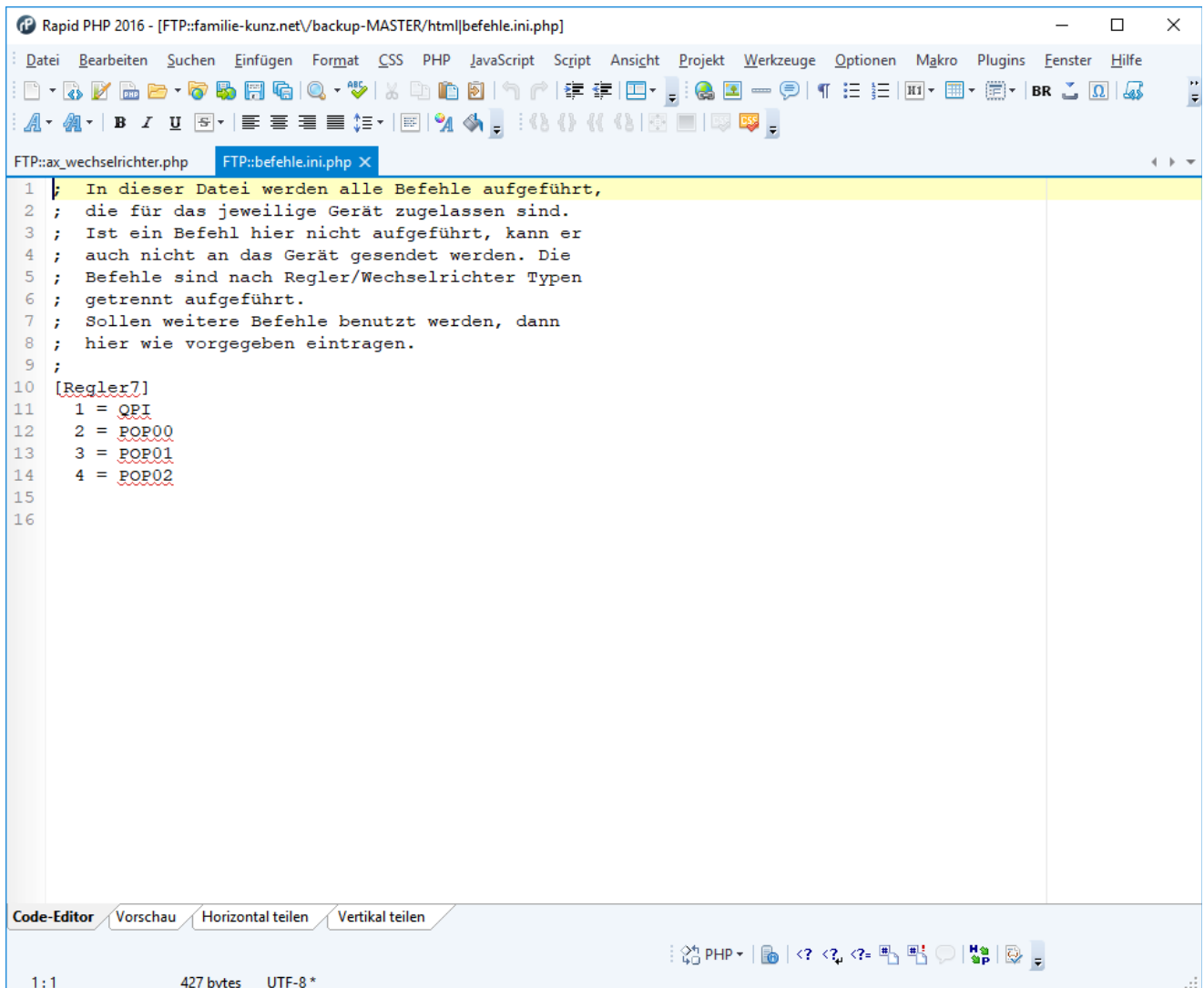
Die Datei /var/www/pipe/<Gerätenummer>.befehl.steuerung wird zur Synchronisation mit dem Auslesen der Geräte benutzt. Jede Minute werden die Geräte ein oder mehrmals ausgelesen. Zuerst wird nachgesehen, ob es die Datei „<Gerätenummer>.befehl.steuerung“ gibt. Wenn ja, wird in der Datei „befehle.ini.php“ nachgesehen ob der Befehl für das jeweilige Gerät freigegeben ist. Das ist noch einmal eine Sicherheit, dass nicht willkürliche Befehle zum Gerät gesendet werden können.

Sind die Befehle freigegeben, werden die enthaltenen Befehle der Datei „<Gerätenummer>.befehl.steuerung“ an das jeweilige Gerät gesendet und die Datei „<Gerätenummer>.befehl.steuerung“ wird gelöscht. Ist ein Befehl nicht freigegeben oder unbekannt, wird das Senden von Befehlen abgebrochen. Erst danach wird das Gerät ausgelesen.

Befehle können auch über den Broker im MQTT Format gesendet werden. Alle Befehle werden gesammelt und einmal pro Minute an das Gerät gesendet. Maximal 3 Befehle auf einmal!

## Schritt für Schritt Anleitung.

1. In der obigen Tabelle nachsehen, ob für den eingesetzten Regler/Wechselrichter das Senden von Befehlen freigegeben ist. Für Regler 7 und 9 ist es zum Beispiel der Fall. Mit einem Editor, z.B. mcedit die Datei /var/www/html/befehle.ini.php öffnen. Sie sieht wie im Bild zu sehen aus.



```
1 ; In dieser Datei werden alle Befehle aufgeführt,  
2 ; die für das jeweilige Gerät zugelassen sind.  
3 ; Ist ein Befehl hier nicht aufgeführt, kann er  
4 ; auch nicht an das Gerät gesendet werden. Die  
5 ; Befehle sind nach Regler/Wechselrichter Typen  
6 ; getrennt aufgeführt.  
7 ; Sollen weitere Befehle benutzt werden, dann  
8 ; hier wie vorgegeben eintragen.  
9 ;  
10 [Regler7]  
11 1 = QPI  
12 2 = POP00  
13 3 = POP01  
14 4 = POP02  
15  
16
```

Die aktuell freigegebenen Befehle finden Sie in der Datei /var/www/html/befehle.ini.php

Möchte man weitere Befehle einbauen, dann wie vorgegeben diese in die Datei eintragen. Jeder Befehl muss so, wie er in den Hersteller Unterlagen steht, hier eingetragen werden. Er muss sich unter der jeweiligen Reglernummer befinden.

Das dient dazu, dass kein ungewollter Befehl gesendet werden kann und eventuell das Gerät beschädigt. Hat man die Datei geändert, dann die Änderungen abspeichern.

Ist die Datei noch nicht vorhanden, dann bitte diese wie hier zu sehen erstellen. Der Name der Datei muss „befehle.ini.php“ lauten. Man kann die aktuelle Version dieser Datei auch von unserem Support Server herunter laden. Sie wird bei einem Software Update nicht erneuert, damit Änderungen von Ihnen nicht überschrieben werden!

Das Zeichen „#“ dient als Platzhalter. Wenn man zum Beispiel POP## als Befehl einträgt, dann würde jeder Befehl der 5 Zeichen lang ist und mit POP beginnt als gültig erkannt werden. Wenn unterschiedliche Zahlenwerte in Befehl vorkommen ist das ganz nützlich. Die Länge des Befehls mit Platzhalter muss mit der Länge des eigentlichen Befehls übereinstimmen.

Beispiel:

Der Befehl lautet POP00 dann muss der Platzhalter POP## lauten und nicht POP#

Der Befehl PCVV60.0 muss dann dementsprechend PCVV##.# lauten und nicht PCVV###

- Die Befehle können zusätzlich als Button's in das Dashboard eingebaut werden. Wie das geht ist auf dem nächsten Bild zu sehen.

Am besten die 2. oder 3. Zeile kopieren und dann so abändern, dass der neue Befehl benutzt wird. Die letzte Zeile „<iframe src= .....“ dient dazu, das Ergebnis des Aufrufes zu „verstecken“ Bitte die Zeile so lassen wie sie ist.

Steuerung

|                     |
|---------------------|
| RPI ausschalten     |
| Netz Mode (UTI)     |
| Solar Mode (SDL)    |
| Batterie Mode (SBU) |

Text

General Options

Mode html

Content

```
<a href="http://solaranzeige.local/sever.halt.php"><button style="background: #E03434; color: black; width: 95%; margin: 1px; margin-bottom: 5px;">RPI ausschalten</button></a>
<a href="http://solaranzeige.local/beh1.steuerung.php?befehl=POP00" target="frame" ><button style="background: #cccc; color: black; width: 95%; margin: 1px;">Netz Mode (UTI)</button></a>
<a href="http://solaranzeige.local/beh1.steuerung.php?befehl=POP01" target="frame" ><button style="background: #888888; color: white; width: 95%; margin: 1px;">Solar Mode (SDL)</button></a>
<a href="http://solaranzeige.local/beh1.steuerung.php?befehl=POP02" target="frame" ><button style="background: #444444; color: white; width: 95%; margin: 1px;">Batterie Mode (SBU)</button></a>
<iframe src="about:blank" name="frame" height="0" width="0" frameborder="0" />
```

## **Steuerbefehle an Wechselrichter / Regler schicken mit Hilfe von MQTT Befehlen.**

Mit Hilfe von dem MQTT Protokoll können Befehle zu den Wechselrichtern / Reglern gesendet, sowie Daten in die lokale Influx Datenbank geschrieben werden. Die Steuerung kann so z.B. von einer Smart Home Zentrale vorgenommen werden, die dann den Wechselrichter steuert und Befehle in die InfluxDB schreibt, sodass sie graphisch dargestellt werden können.

Haben Sie Fragen zu dieser Beschreibung, dann bitte auf unserem Support Server <https://solaranzeige.de> stellen oder eine email an [support@solaranzeige.de](mailto:support@solaranzeige.de)