

**Projekt: solaranzeige.de**

## **Allgemeine Steuerung**

**Stand Februar 2021**

### **Inhaltsverzeichnis**

Übersicht:.....	2
Verwendete Geräte:.....	3
Verwendete Relais:.....	3
Steuerung Steuerkreis x Kontakt x.....	4
Relais Abschaltung nach Zeit.....	5
Die Datei auto-math.php.....	6
Installation.....	7
BETA Version testen:.....	8

## Übersicht:

Mit der allgemeinen Steuerung, können Wärmepumpen, Heizungen, Poolpumpen usw. in Abhängigkeit der Solarerzeugung gesteuert werden. Die Steuerung erfolgt mit Relais, die mit der TASMOTA Firmware laufen, wie z.B. Sonoff Basic, Sonoff TH16, Sonoff 4CH oder auch Shelly 2.5 und viele andere Relais. Es können 2 unabhängige Relais mit jeweils bis zu 4 Kontakten benutzt werden. Die Relais gibt es in Schwachstrom und in 230Volt Ausführungen, sowie mit 230 Volt und potentialfreien Kontakten. Welches Relais das richtige ist, hängt von dem zu steuernden Gerät ab.

Je nachdem welche PV Anlage man in Betrieb hat, können folgende Werte zur Steuerung herangezogen werden. Die PV Leistung eines Ladereglers, die erzeugte AC Leistung eines Wechselrichters, die Einspeisung oder den Bezug in Watt eines SmartMeters und die aktuelle Kapazität (SOC) einer Batterie. Die Steuerung kann anhand aller Werte ermittelt werden oder aber auch nur Teile davon. Die nötigen Werte können von mehreren Geräten (Laderegler, Wechselrichter SmartMeter und BMS) kommen oder aber nur von einem, wenn das Gerät alle Daten liefern kann.

Die einfachste Konfiguration wäre ein Laderegler, der die PV Leistung liefert. Damit könnte schon die Steuerung funktionieren. Alle Geräte, die benutzt werden sollen, müssen an der Solaranzeige angeschlossen sein.

Die Konfiguration der Steuerung erfolgt über eine WEB-Seite die wie folgt aufgerufen wird:

<http://solaranzeige.local/automation.web.php>

Funktioniert in ihrem Netzwerk die URL „solaranzeige.local“ nicht, dann muss sie mit der IP Adresse des Raspberry ersetzt werden. Je nachdem wie viele Relais und wie viele Kontakte sie in der Konfiguration angeben, desto mehr Eingabe Möglichkeiten haben sie. Die Seite verlängert bzw. verkürzt sich entsprechend der Konfiguration.



Solaranzeige.de  
ALLGEMEINE STEUERUNG

**Verwendete Geräte**

Laderegler (PV Leistung)	<input type="text" value="4"/>	<input type="text" value="1"/>	<input type="text" value="solaranzeige"/>
Wechselrichter (AC Leistung)	<input type="text" value="Regler Nummer"/>	<input type="text" value="Geräte ID"/>	<input type="text" value="Datenbankname"/>
SmartMeter (Bezug und Einspeisung)	<input type="text" value="Regler Nummer"/>	<input type="text" value="Geräte ID"/>	<input type="text" value="Datenbankname"/>
Batterie-Management-S (SOC)	<input type="text" value="Regler Nummer"/>	<input type="text" value="Geräte ID"/>	<input type="text" value="Datenbankname"/>

**Verwendete Relais**

Relais 1 [ TH16-Garage ]  Anzahl der Kontakte:

Relais 2 [ 4CHPro ]  Anzahl der Kontakte:

**Steuerung Steuerkreis 1 Kontakt 1 [ EIN ]**

Relais 1 Kontakt 1 einschalten bei: (leere Felder werden nicht berücksichtigt.)

>>  W PV Leistung  und  oder

>>  W AC Leistung  und  oder

>>  W Einspeisung  und  oder

>>  % Batterie SOC

**Steuerung Steuerkreis 1 Kontakt 1 [ AUS ]**

Relais 1 Kontakt 1 ausschalten bei: (leere Felder werden nicht berücksichtigt.)

<<  W PV Leistung  und  oder

<<  W AC Leistung  und  oder

<<  W Bezug  und  oder

<<  % Batterie SOC

**Steuerung Steuerkreis 1 Kontakt 2 [ EIN ]**

Relais 1 Kontakt 2 einschalten bei: (leere Felder werden nicht berücksichtigt.)

>>  W PV Leistung  und  oder

>>  W AC Leistung  und  oder

>>  W Einspeisung  und  oder

>>  % Batterie SOC

**Steuerung Steuerkreis 1 Kontakt 2 [ AUS ]**

Relais 1 Kontakt 2 ausschalten bei: (leere Felder werden nicht berücksichtigt.)

<<  W PV Leistung  und  oder

<<  W AC Leistung  und  oder

<<  W Bezug  und  oder

<<  % Batterie SOC

---

**Steuerung Steuerkreis 2 Kontakt 1 [ EIN ]**

Relais 2 Kontakt 1 einschalten bei: (leere Felder werden nicht berücksichtigt.)

>>  W PV Leistung  und  oder

>>  W AC Leistung  und  oder

>>  W Einspeisung  und  oder

>>  % Batterie SOC

**Steuerung Steuerkreis 2 Kontakt 1 [ AUS ]**

Relais 2 Kontakt 1 ausschalten bei: (leere Felder werden nicht berücksichtigt.)

<<  W PV Leistung  und  oder

<<  W AC Leistung  und  oder

<<  W Bezug  und  oder

<<  % Batterie SOC

**Steuerung Steuerkreis 2 Kontakt 2 [ EIN ]**

Relais 2 Kontakt 2 einschalten bei: (leere Felder werden nicht berücksichtigt.)

>>  W PV Leistung  und  oder

>>  W AC Leistung  und  oder

>>  W Einspeisung  und  oder

>>  % Batterie SOC

**Steuerung Steuerkreis 2 Kontakt 2 [ AUS ]**

Relais 2 Kontakt 2 ausschalten bei: (leere Felder werden nicht berücksichtigt.)

<<  W PV Leistung  und  oder

<<  W AC Leistung  und  oder

<<  W Bezug  und  oder

<<  % Batterie SOC

## Verwendete Geräte:

Hier werden die Geräte eingetragen, die an der Multi-Regler Solaranzeige hängen und zur Auswertung benutzt werden sollen. Es können 4 einzelne Geräte hier eingetragen werden. Eingetragen werden muss hier die Regler-Nummer aus der user.config.php! Hat ein Wechselrichter den Laderegler und SmartMeter mit eingebaut, wie z.B. der Fronius Symo, so muss dieses Gerät bei Laderegler, Wechselrichter und SmartMeter eingetragen werden. Auch der Datenbankname muss in diesem Fall 3 mal gleich lauten.

## Verwendete Relais:

Hier bitte die bis zu 2 Relais eintragen, die benutzt werden sollen. Zuerst den Relaisstyp auswählen, dann die Anzahl der benutzten Kontakte von diesem Relais und dann den Topic Namen, so wie er im Relais angegeben wurde. Auf Groß- und Kleinschreibung achten! (Siehe Bild)

Sonoff TH Modul  
TH16-Garage

**MQTT-Einstellungen**

Host ()

Port (1883)

client (DVES\_315251)

Benutzer (DVES\_USER)

Passwort

topic = %topic% (tasmota\_315251)

full topic (%prefix%/ %topic%/)

Tasmota 9.2.0 von Theo Arends

Als Host() muss die IP Adresse des Raspberry eingetragen werden.

„garage“ ist in diesem Fall der Topic Name. Es ist wichtig, dass der eingetragene Topic Name im Relais und hier in der Konfigurationsdatei gleich lautet. Auf Groß- und Kleinschreibung ist zu achten! Es ist am Besten, die MQTT Konfiguration im Relais vorher zu konfigurieren. Bitte den eingetragenen Topic Namen notieren. Im Topic Namen bitte keine Sonderzeichen oder Umlaute verwenden. Jedes Relais muss einen anderen Topic Namen haben, da dadurch die einzelnen Relais erkannt werden. Alle anderen Einträge auf dieser Seite so lassen wie sie sind.

The screenshot shows the configuration page for a Sonoff TH16-Garage module. The title is 'Sonoff TH Modul TH16-Garage'. Under 'Sonstige Einstellungen', there is a 'Vorlage' section with a text input field containing the JSON configuration: `{"NAME":"Generic","GPIO":[1,1,1,1,1,1]}` and an 'Aktivieren' checkbox. Below that is a 'Passwort für Web Oberfläche' field with a masked password. The 'MQTT aktivieren' checkbox is checked. The 'Device Name (TH16-Garage)' field contains 'TH16-Garage' and is circled in red. Below it is the 'Name [friendly name] 1 (Tasmota)' field with 'TH16-Garage'. The 'Emulation' section has three radio button options: 'keine', 'Belkin WeMo Einzelnes Gerät', and 'Hue Bridge Mehrfachgerät'. At the bottom, there are 'Speichern' and 'Einstellungen' buttons, and a footer 'Tasmota 9.2.0 von Theo Arends'.

Wenn alles eingetragen ist, kann das Relais mit dem „Test“ Button geprüft werden. Es muss bei jedem Klick ein und ausschalten. Damit kann man die Einträge auf Richtigkeit und die Verbindung zum Relais überprüfen. Zwischen den eckigen Klammern muss der Device Name, wie links zu sehen, erscheinen. Damit steht fest, dass das Relais auch ausgelesen werden kann. Außerdem kann man die Relais so mit sprechenden Namen versehen, damit man sie einfacher unterscheiden kann.

Gibt man mehr Kontakte an, als das Relais hat, so erscheinen Fehlermeldungen in der LOG Datei. (Wie in diesem Beispiel. Ein Sonoff TH16 hat nur einen Kontakt.) In der BETA Version können nur maximal 2 Kontakte benutzt werden, später bis zu 4

Die LOG Datei findet Ihr in dem Unterverzeichnis `/var/www/log/` und sie kann so aufgerufen werden `„mcedit /var/www/html/automation.log“`

### **Steuerung Steuerkreis x Kontakt x**

Hier wird die Logik der einzelnen Relais Kontakte ausgewählt. In die leeren Felder bitte den jeweiligen Wert eintragen. Hinter dem Eingabefeld kann man die Verknüpfung auswählen. „und“ / „oder“. Lässt man ein Feld leer, dann wird es bei der Berechnung übersprungen und hat keine Wirkung. Mindestens ein Feld muss jedoch ausgefüllt sein. Pro Kontakt gibt es eine Checkbox, die markiert sein muss, damit der Kontakt auch aktiv ist. Mit

dieser Checkbox können Kontakte aktiviert / deaktiviert werden, ohne dass die Einstellung verändert werden muss.

Liefert ein Gerät negative Werte, z.B. bei der Leistung (negativ = Einspeisung, positiv = Bezug) dann können diese Werte nicht für die Steuerung benutzt werden! Nur wenn ein SmartMeter für die Einspeisung und auch für den Bezug positive Werte liefert, kann das Gerät benutzt werden.

Keine der benutzten Werte darf negativ sein!

Deaktiviert man ein Relais, dann bleibt es in der momentan aktiven Stellung stehen. Auch wenn es zeitgesteuert abfallen soll, wird das nicht mehr ausgeführt. Deaktiviert man ein Relais, so muss man es manuell in die Stellung bringen, die es dauerhaft annehmen soll. [ ein / aus ]

Der Schriftzug [ EIN ] und [ AUS ] wird rot, wenn dieser Zustand aktiv ist und man „Daten absenden“ oder „Test“ gedrückt hat. Die Auswertung läuft immer zur vollen Minute! Die Relais werden also immer nur zur vollen Minute ein / aus geschaltet.

## **Relais Abschaltung nach Zeit**

Soll das Relais aufgrund der PV Werte eingeschaltet, aber dann nach einer bestimmten Zeitspanne wieder ausgeschaltet werden, dann bitte wie folgt verfahren.

Auf der Konsole bitte folgendes eingeben:

```
„sudo sqlite3 /var/www/html/database/automation.sqlite3 'update waermepumpen set Relais1K1ausMinuten = xxx where Id = 1' “
```

Wobei XXX für ein Wert in Minuten steht.

0 = Funktion ausschalten.

Beispiel:

```
sudo sqlite3 /var/www/html/database/automation.sqlite3 „update waermepumpen set Relais1K1ausMinuten = 30 where Id = 1“
```

In diesem Beispiel bleibt der Kontakt 1 von Relais 1 ca. 30 Minuten (+/- 1 Minute) eingeschaltet.

Durch die minütliche Ausführung kann die genaue Einschaltzeit sich um 1 Minute verschieben. Bitte ausprobieren. Zeitlängen unter 2 Minuten sind zu ungenau.

Möchte man den Kontakt Nr. 2 so schalten, dann bitte

**Relais1K1ausMinuten** in **Relais1K2ausMinuten** ändern. Genauso verfahren mit Relais 2

Deaktiviert man über die Konfigurationsseite ein Relais, so wird die Zeitsteuerung abrupt beendet und das Relais schaltet nicht mehr aus. Dieses muss dann manuell geschehen. Z.B. mit dem „Test“ Button.

## **Die Datei auto-math.php**

In der Datei kann man eigene Berechnungen, zum Schalten der Relais, durchführen. Die Relais Kontakte werden mit einer Variablen ein oder ausgeschaltet. Sind die Variablen vorhanden, so wird die Berechnung der Steuerung damit übersprungen.

Die Variablen heißen

```
$UserKontaktAuswertung[„Relais1“][„Kontakt1“] = 1;           (0 = aus, 1 = ein)
$UserKontaktAuswertung[„Relais2“][„Kontakt2“] = 0;
```

Ist eine der Variablen vorhanden, wird der Kontakt so geschaltet, egal was die Steuerung berechnet.

Außerdem kann man die Datenbank hier zusätzlich noch einmal auslesen und alle Werte benutzen, die in dem Script automation.php ausgelesen wurden. Durch das include der Datei auto-math.php stehen auch in dieser Datei alle Variablennamen zur Verfügung. So können die wichtigen Werte der Variablen:

```
$PVLeistung
$ACLeistung
$Bezug
$Einspeisung
$SOC
```

anders berechnet werden. Diese 5 Werte benutzt die Steuerung zur Berechnung. In der Datei auto-math.php können die Werte vorher manipuliert oder erst einmal berechnet werden. Dadurch wird die Steuerung noch flexibler. Man muss jedoch PHP Kenntnisse haben.

Der Vorteil von „include“ ist, dass die Datei auto-math.php vom Benutzer angelegt und geändert werden kann, ohne dass sie bei einem Update überschrieben wird. Falls es nur die Datei „auto-math.template“ gibt, dann bitte die Datei in auto-math.php umbenennen. Die Datei auto-math.template kann bei einem Update überschrieben werden und dient als Vorlage. Sie enthält einige wichtige Informationen.

## Installation

Damit die Steuerung funktioniert, muss ein Eintrag in der Cronstab geändert / hinzugefügt werden. Dazu bitte auf der Konsole „sudo crontab -e“ eingeben. Es wird nach einem Editor gefragt. Bitte 2 = mcedit auswählen. Dann erscheint nachfolgendes Fenster:

```
root@solaranzeige: /var/www/html
/tmp/crontab.7HidOX/crontab  [-M--] 49 L:[ 2+21 23/ 45] *(937 /1702b) 0010 0x00A [*][X]
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin
MAILTO=""

#@reboot sleep 40 && /solaranzeige/support >/dev/null
@reboot /var/www/html/usb_init.php >/dev/null

* * * * * /var/www/html/regler_auslesen.php >/dev/null
* * * * * /var/www/html/checkreboot.sh >/dev/null
20,50 * * * * /var/www/html/wetterdaten.php >/dev/null
* * * * * /var/www/html/mqgtt_prozess.php >/dev/null
55 23 * * * /var/www/html/wartung.php >/dev/null
## * * * * * /var/www/html/wallbox_steuerung.php >/dev/null
#01 */4 * * * /var/www/html/awattar.php >/dev/null
## * * * * * /var/www/html/automation.php >/dev/null

1Hilfe 2Spe~ern 3Mar~ren 4Ers~zen 5Kop~ren 6Ver~ben 7Suchen 8Löschen 9Menüs 10Beenden
```

Ist die letzte Zeile vorhanden, dann bitte das Doppelkreuz „##“ entfernen, damit die Zeile aktiv wird. Ist die Zeile nicht vorhanden, dann bitte genau so eintragen. ( ohne # )

Danach F2 , speichern und F10 drücken.



## **BETA Version testen:**

Möchten Sie die BETA Version schon einmal testen, dann bitte die Datei „update-steuerung“ vom Support Server herunterladen und in das HOME Verzeichnis /home/pi/ speichern.

Das geht so:

Auf der Konsole in das Home Verzeichnis von pi springen und folgendes eingeben:

```
„cd /home/pi“
```

```
„wget http://solaranzeige.de/update-steuerung“
```

```
„sudo php update-steuerung“
```

```
„sudo crontab -e“
```

Dann 2 x „w“ drücken, dann sollte die Installation fertig sein. Den Eintrag in der crontab nicht vergessen und dann <http://solaranzeige.local/automation.web.php> im Browser aufrufen.

Die BETA Version hat zur Zeit folgende Einschränkungen.

Es funktionieren beide Relais jedoch mit maximal 2 Kontakten.

Die volle Version wird später mit 2 Relais und jeweils bis zu 4 Kontakten funktionieren.

### **TIP:**

Sie können den kompletten Update mehrmals durchführen, falls es neue Versionen der Dateien gibt. Die Datenbank „automation.sqlite3“ im Verzeichnis /var/www/html/database/ wird, wenn sie einmal vorhanden ist, nicht mehr überschrieben, sodass Ihre Konfiguration bestehen bleibt. Es könnte aber einmal nötig sein, auch die Datenbank zu überschreiben, z.B. wenn sie weitere Felder enthält. In diesem Fall muss der Update „sudo php update-steuerung force“ heißen. Damit wird dann die neue Datenbank auch übertragen. Ihre Konfiguration auf der WEB Seite automation.web.php sind dann aber wieder auf 0 gesetzt. Wenn das nötig ist, wird es bekannt gegeben.

© Solaranzeige.de Nachdruck und Verbreitung nur mit unserer schriftlichen Genehmigung.