

**Projekt: solaranzeige.de**

## **Die Strompreis geführte Steuerung der Solaranzeige**

**Dokument SG026**

**Stand Februar 2024**



### **Inhaltsverzeichnis**

Übersicht:.....	2
Bist du noch kein Kunde von Tibber?.....	3
Wofür soll so eine Steuerung gut sein?.....	4
Welche Geräte können gesteuert werden?.....	5
Die Installation:.....	6
Die Konfiguration als Demo:.....	8
Die Konfiguration:.....	10
Was wird sonst noch benötigt?.....	10
Die INI Datei:.....	11
Prüfen, ob das Relais auch erreicht wird und schaltet.....	16
Wichtig bei Tibber!.....	16
Die WEB-Konfiguration:.....	17
Menü Einstellungen:.....	18
Menü Relais:.....	19
Menü Schaltpunkte:.....	21
Datenbankfelder der Konfiguration:.....	23
Zusammenfassung der Steueraufgabe:.....	24
Ergebnis an andere Steuerungen weiterreichen:.....	24
Reset der Steuerung .....	25
Update durchführen.....	25
Die LOG Datei.....	25
Grafische Anzeige von Verbrauch und Kosten:.....	26
Aufzeichnung der Schaltvorgänge.....	27
Batterie laden bei niedrigem Strompreis:.....	28
Support:.....	28

## Übersicht:

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Euch von Nutzen sein wird, aber OHNE IRGENDNEINE GARANTIE, sogar ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK. Details findet Ihr in der GNU General Public License.

Die Strompreis geführte Steuerung ist zur Zeit für Tibber und aWATTar Kunden. Nur wenn man dort Kunde ist bekommt man einen API Key, der nötig ist, um die richtigen Preise für seine Region abzurufen. Es gibt in der Regel einen „Demo Key“ mit dem man die Funktion testen kann. Die zugrunde gelegten Börsenpreise passen aber nicht zur eigenen Region. Kann also nur zu Demozwecken genutzt werden.

Die Strompreis-Steuerung ist eine eigenständige Software, die auf jedem Linux läuft, welches einen WEB-Server und PHP ab Version 7.3 installiert hat. Zusätzlich sollte noch Grafana installiert sein, das ist aber keine Voraussetzung. Alle Dateien und die Datenbanken befinden sich nach der Installation im Verzeichnis `/var/www/html/sgs`

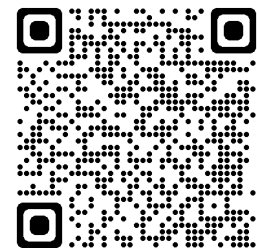
Die Lage der LOG Datei und der Datenbanken kann in der Konfiguration später noch angegeben und dadurch verändert werden. Die Basisinstallation speichert alle Dateien in das Verzeichnis `/var/www/html/sgs`

Die Steuerung arbeitet rein lokal! **Ob die Steuerung in einem Docker Container funktioniert ist ungewiss und nicht getestet. Ohne Anpassungen höchstwahrscheinlich nicht.** Eine Cloud wird nicht benötigt. Es handelt sich hierbei nicht um ein fertiges Produkt, sondern um Open Source Software, die man zum Erstellen einer Steuerung braucht und eine genaue Anleitung.

Konfiguriert kann die Steuersoftware durch eine INI Datei oder per WEB-Interface mit Hilfe einer Datenbank. Beides ist in der Installation vorhanden. Wie man den Konfigurationsweg auswählen kann, wird später noch erklärt.

Sobald die Installation abgeschlossen ist beginnt die Steuerung zu arbeiten. Vorkonfiguriert ist Awattar, sodass eine Minute nach der Installation die Börsendaten von Awattar abgeholt und in eine SQLite3 Datenbank gespeichert werden. Die Datenbank heißt „sgs.sqlite3“ Dort findet man eine Tabelle „awattarPreise“ und „tibberPreise“ Je nach Konfiguration wird eine der Tabellen gefüllt. Bei aWATTar immer 5 Minuten nach jeder vollen Stunde, bei Tibber 5 Minuten nach jeder geraden Stunde sowie nach der Installation sofort. Es werden die Preise vom aktuellen Tag und ab 14:00 Uhr auch die Preise vom Folgetag gespeichert. Für jede Stunde gibt es einen Eintrag mit einem Preis. Der Preis kann sehr stark schwanken. Das ist die Basis zur Berechnung.

Hier geht es zur richtigen Forumseite:



Damit man sich erst einmal mit der Funktionsweise vertraut machen kann, ist es möglich als Relais ein Dummy einzugeben. Wie hier in der INI Datei.

```
[Relais]
Aktiv[1]           = "Ja"                ; "Ja" / "Nein"
Autotimer[1]      = "Ja"                ; "Ja" / "Nein"
Name[1]           = "Heizstab1"         ; Relais Name (frei wählbar)
Typ[1]            = "Dummy"             ; "Shelly", "Tasmota", "Dummy"
Protokoll[1]      = ""                  ; "http", "tasmota", "mpi_pi17"
IP-Adresse[1]     = ""                  ; IP Adresse oder USB Device
Port[1]           = ""                  ; Port
AnzahlKontakte[1]= "1"                  ; Zurzeit nur 1 möglich
Kontaktnummer[1] = "1"                  ; Zurzeit nur 1 möglich
```

Damit wird die gesamte Konfiguration nur simuliert, es wird aber nichts wirklich geschaltet. So kann man die Funktionsweise mehrere Tage prüfen, da alles in der LOG Datei und in der Datenbank festgehalten wird. Sich blind darauf zu verlassen, ist keine gute Idee. Die Dummy Relais Einstellung kann man auch nutzen, wenn die Steuerung nichts direkt steuern soll, sondern man nur die Datenbankeinträge für andere Zwecke weiterverarbeiten möchte. Mit der Steuerung kann man ausschließlich Geräte ein und ausschalten oder Wechselrichter programmieren, in Abhängigkeit der Börsen-Strompreise. In der vollen Ausbaustufe können 30 Relais durch 30 Schaltpunkte (Aufgabenstellungen) gesteuert werden. Andere Steuerungsaufgaben können damit nicht erledigt werden. Dafür ist die Automation oder die „Steuerung der Solaranzeige“ gedacht. Informationen darüber findet ihr im Forum.

## Bist du noch kein Kunde von Tibber?

Dann gebe bitte bei der Anmeldung bei Tibber den Gutschein-Code **„Solaranzeige“** ein. Dann bekommst du 50 € Bonus im Tibber Store und ich bekomme eine kleine Spende von Tibber, wenn dein Vertrag zustande kommt. So haben wir beide etwas. Falls du bei der Anmeldung den Bonuscode „Solaranzeige“ noch nicht eingetragen hast, kannst du das später in der Tibber APP noch nachholen.

So geht es:

- **Öffne die Tibber-APP**
- **Klicke unten auf „Power-ups“**
- **Klicke auf „Bonus“ – „Ich wurde eingeladen“**
- **Gebe den Gutschein-Code „Solaranzeige“ ein**

Du kannst auch noch **bis zu 14 Tage nach Vertragsabschluss** den Code unter "Power-ups" - "Bonus" - "Ich wurde eingeladen" eintragen.

## Wofür soll so eine Steuerung gut sein?:

Mit der Steuerung kann ich Relais schalten, immer dann wenn der Börsenpreis besonders niedrig / besonders hoch ist. Gut geeignet ist der Shelly Plus Plug S wie man ihn hier sehen kann. Damit ist die Installation ein Kinderspiel. Alles was einen Schuko Stecker hat, und nicht mehr als 2500 Watt verbraucht, kann so geschaltet werden. Pro Schaltpunkt (Aufgabenstellung) kann immer ein Relais geschaltet werden. Es können viele Schaltpunkte (Aufgaben) definiert werden. (Max 30)



Man kann alle größeren Verbraucher zu der Zeit einschalten, in der der Strom besonders günstig ist. Ein Beispiel: Wenn man das Warmwasser mit einem 160 Liter Boiler erwärmt, so kann der Heizstab immer dann eingeschaltet werden, wenn am Tage oder in der Nacht der Strompreis am günstigsten ist. Die Zeiten mit den hohen Strompreisen können so ausgeschlossen werden. Ein 160 Liter Boiler benötigt ca. 4 Stunden um auf 65°C zu erwärmen und hält die Wärme weit mehr als 24 Stunden. Es werden so genau die 4 Stunden am Tage ausgesucht, die am preiswertesten sind. Dabei muss die Einschaltzeit nicht an einem Stück erfolgen. Die preiswertesten Stunden sucht die Steuerung automatisch heraus.

Sehr gut geeignet sind auch Infrarot-Heizungen, Akkus, eAutos, Poolpumpen usw.

Man kann aber auch klein anfangen. Ein Wasserboiler (Untertischspeicher) mit 10 oder 15 Liter kann im Handumdrehen „smart“ gemacht werden. Mit der Steuerung ist es möglich die 2, 4 oder 6 „teuersten“ Stunden des Strompreises einfach zu „überspringen“. Der Boiler wird nur in der restlichen Zeit geheizt und spart so Geld. Dazu ist nur ein Raspberry oder Ähnliches und ein „Shelly Plus Plug S“ nötig. (Siehe Bild)

Hervorragend geeignet ist eine Warmwasser-Wärmepumpe, die vermehrt installiert werden. Die haben alle in der Regel Steuer Anschlüsse für Tag/Nacht Stromtarife und oder Solaranlagen-Steuerung. Diese Steuereingänge kann man für die Strompreis-Steuerung hervorragend nutzen. Viele Hersteller bieten eine 2 stufige Steuerung mittels potentialfreien Kontakten. Dazu eignen sich entweder 2 x Shelly PLUS 1 Relais oder ein Sonoff 4CH Pro Relais. Es gibt aber noch weitere Relais Typen. Sie müssen nur potentialfreie Kontakte haben.

Es gibt noch viele Beispiele, die man hier aufführen könnte. Die richtigen Ideen werden erst kommen, wenn der Strompreis noch mehr schwankt, was bei mehr Wind und Solarenergie wahrscheinlich ist. Auch kann man davon ausgehen das die Energiepreise eher steigen als fallen werden. Was heute noch als Spielerei gilt, kann Morgen schon wichtig und nötig sein.

**Die Steuerung ist lokal** und kann alles steuern, was man über eine WLAN Steckdose ein- und ausschalten kann. Sie ist erst am Anfang der Entwicklung. Man könnte sich vorstellen, dass auch über MODBUS TCP oder ähnlichem gesteuert werden kann. Noch können nicht alle Möglichkeiten ausgeschöpft werden.

Wichtig ist vor allem, dass das eigene Haus erst einmal „smart“ gemacht wird. Ohne Möglichkeiten, Verbraucher einzeln automatisch ein- und auszuschalten, bringt die beste Steuerung nichts. Man muss also seine eigenen vier Wände erst einmal dafür vorbereiten. Packen wir es an.... Eure Ideen sind im Forum sehr willkommen. Ab 2025 werden alle Stromanbieter einen dynamischen Stromtarif anbieten müssen. Bis dahin sollte man alles vorbereiten.

## Welche Geräte können gesteuert werden?

Die Steuerung wird weiter entwickelt. Folgende Geräte kann man im Moment mit der SGS steuern:

Im Relais Feld „TYP“ kann stehen: "Shelly", "tasmota", "mpi\_pi17", "Dummy"

- **„Dummy“**  
In diesem Fall braucht kein Relais vorhanden sein. Das Relais wird simuliert. Es dient dazu, dass nur die Schaltzeiten in die Tabelle „Relaisschaltungen“ und Feld Kontakt1 geschrieben werden. Das kann zur Simulation benutzt werden oder wenn man die Tabelle „Relaisschaltungen“ anderweitig ausliest und verarbeitet.
- **„Shelly“ WiFi Relais mit originaler Firmware:**  
Shelly Plus 1  
Shelly Plus Plug S  
Shelly Plus 1PM  
Shelly Plus H&T  
Shelly Plus Mini 1
- **„Tasmota“ Fast alle Relais mit TASMOTA Firmware.**
- **„mpi\_pi17“ MPP-Solar Wechselrichter und Baugleiche ( Regler = 9 )**  
Alle Geräte, die mit den Befehlen „EDB0“ und „EDB1“ gesteuert werden können.  
EDB0/1 = Set energy distribution Enable/disable AC charge battery

## Die Installation:

Die Installation sollte auf jedem Linux möglich sein. Bevorzugt bitte Debian benutzen. Auf dem Betriebssystem muss PHP ab Version 7.3 und ein WEB-Server vorhanden sein. Wer eine grafische Anzeige möchte sollte noch Grafana installieren. Die Datenbank SQLite3 ist in PHP 7.3 und höher schon enthalten.

Auf den aktuellen Images der Solaranzeige ab Version 5.1.2 ist die Steuerung schon installiert, aber noch nicht eingeschaltet. Eingeschaltet wird sie mit dem Eintrag in der crontab. Siehe etwas weiter unten. Bitte erst die komplette Konfiguration durchführen, bevor man den Eintrag in der crontab aktiviert.

Bitte mit PUTTY sich auf dem Linux einwählen oder eine Konsole starten. Bei dieser Beschreibung wird davon ausgegangen, dass die Software auf einem Raspberry Pi mit dem User pi durchgeführt wird. (Raspbian Bullseye Debian 10.0) Falls es keinen User pi gibt, sollte einer angelegt werden, da alles darauf ausgelegt ist. Auch wird davon ausgegangen, dass es ein Unterverzeichnis /var/www/html gibt, das von dem installierten WEB-Server benutzt wird.

Auf der Konsole dann bitte folgendes eingeben:

**wget -N [https://solaranzeige.de/sgs\\_install](https://solaranzeige.de/sgs_install)**

**sudo bash sgs\_install** oder wenn die Steuerung schon einmal installiert wurde, dann

**sudo bash sgs\_install -f**

Nach erfolgreicher Installation sollte es das Unterverzeichnis „/var/www/html/sgs“ geben in dem sich mehrere Dateien und 2 Datenbanken befinden. Nach 1 Minute findet man in der Datenbank „sgs.sqlite3“ die ersten Börsendaten.

Ansehen kann man die Börsenpreise mit folgendem Befehl:

```
sqlite3 -column -header sgs.sqlite3 "select * from tibberPreise where Timestamp > STRFTIME('%s') - 3600"
```

Vorkonfiguriert ist Tibber, deshalb werden die Preise nach der Installation auch sofort abgeholt. Aber Achtung, am Anfang ist nur der Demo Token vorhanden und somit handelt es sich um aktuelle Demo Daten aus Schweden in schwedischen Kronen. Erst wenn man seinen eigenen Tibber API Token in die Konfiguration eingegeben hat, bekommt man richtige Daten von seiner Region.

### Prüfen ob dieser Eintrag in der crontab vorhanden ist:

```
* * * * * /usr/bin/php /var/www/html/sgs/sgs.php >/dev/null
```

Falls der Eintrag nicht vorhanden ist, dann bitte eintragen, falls er am Anfang ein Doppelkreuz hat, dann dieses bitte entfernen. „#“

**Wichtig!** Haben Sie schon die Testversion der Solaranzeige in Betrieb, dann bitte in der crontab folgende Zeilen löschen, damit die Steuerung nicht doppelt ausgeführt wird!

```
* * * * * /var/www/html/tibber.php >/dev/null
* * * * * /var/www/html/awattar.php >/dev/null
* * * * * /usr/bin/php /var/www/html/strompreisSteuerung.php >/dev/null
```

Ab der Image Version 5.1.2 ist folgendes nicht mehr nötig:

Grafana für die SQLite Datenbank vorbereiten. Auf der Konsole eingeben:

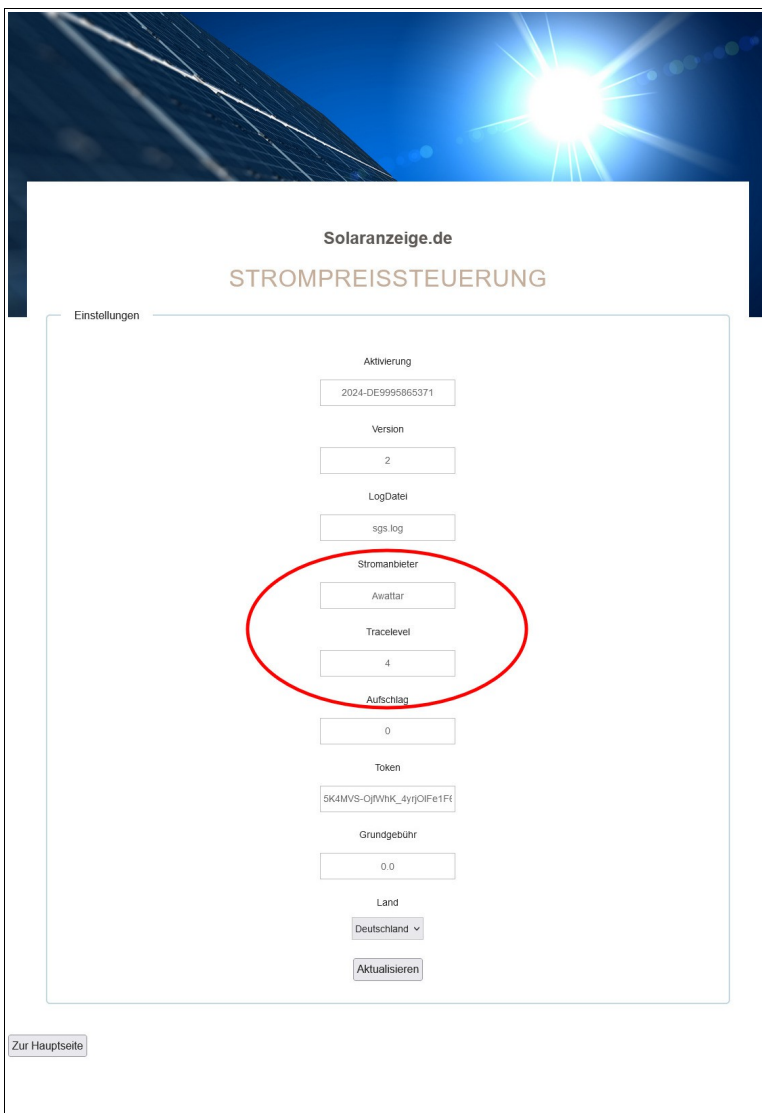
```
grafana-cli plugins install frser-sqlite-datasource
```

und dann `systemctl restart grafana-server` Danach die nötigen Dashboards eventuell importieren.

## Die Konfiguration als Demo:

Die Steuerung kann man als Demo konfigurieren, damit man sich mit der Arbeitsweise schon einmal vertraut machen kann, ohne schon einen dynamischen Vertrag zu haben und ohne dass man schon ein WiFi Relais hat. Die Steuerung funktioniert danach so wie sie soll, jedoch nur „auf dem Papier“. Was ist zu tun?

1. Die Installation durchführen, wie oben beschrieben.
2. Auf der Konsole folgendes eingeben: „rm /var/www/html/sgs/sgs.ini“
3. Auf einem Browser folgendes eingeben: <http://solaranzeige.local/sgs/index.html>



The screenshot shows the 'Einstellungen' (Settings) page for 'Solaranzeige.de STROMPREISSTEUERUNG'. The page contains several input fields for configuration. A red circle highlights the 'Stromanbieter' (Energy Provider) field, which is currently set to 'Awattar'. Other fields include 'Aktivierung' (2024-DE9995865371), 'Version' (2), 'LogDatei' (sgs.log), 'Tracelevel' (4), 'Aufschlag' (0), 'Token' (5K4MVS-OjFWNK\_4yrjOIFe1FE), 'Grundgebühr' (0.0), and 'Land' (Deutschland). A 'Aktualisieren' (Update) button is at the bottom, and a 'Zur Hauptseite' (Back to Home) button is in the bottom left corner.

Bei „Einstellungen“ den Stromanbieter „Awattar“ eintragen und Tracelevel auf „4“ hochsetzen. Der kann auf „2“ herunter gesetzt werden wenn alles zur Zufriedenheit läuft.



Auf der Seite „Relais“ bitte bei „Aktiv“ ==> Ja eintragen und bei „Typ“ ==> Dummy

Auf der Seite „Schaltpunkte“ bitte bei „Aktiv“ ==> Ja, bei „Relaisnummer“ => 1 und bei MaxEinschaltzeit zum Beispiel => 120 was 2 Stunden bedeutet.

Damit wird die Schaltung das Dummy Relais in den 2 preiswertesten Stunden „einschalten“. Da es sich um eine Demo handelt wird man die Einschaltungen nur in der LOG Datei oder in der Datenbank sgs.sqlite3 bzw. auf dem Dashboard verfolgen können.

Jetzt bitte im Browser <http://solaranzeige.local:3000> eingeben und das Dashboard

„AwattarStrompreise.json“ von hier:

<https://solaranzeige.de/phpBB3/download/AwattarStrompreise.json> importieren. Das gleiche Dashboard befindet sich auch im Unterverzeichnis /var/www/html/sgs auf dem Raspberry.

Nach einiger Zeit sollten die Awattarpreise und auch die Schaltstellung des Relais auf dem Dashboard erscheinen. In etwa so:



## Die Konfiguration:

Die Konfiguration ist auf 2 Wegen möglich.

1. Mit Hilfe einer INI Datei
2. Mit Hilfe der WEB basierten Konfigurationsseiten (Ist noch in der Umsetzung)

Beide Varianten werden hier beschrieben.

## Was wird sonst noch benötigt?

Als Schaltelemente werden Shelly WiFi Relais benötigt. Im Moment benötigt die Steuerung eines der folgenden Relais:

**Shelly Plus 1**  
**Shelly Plus Plug S**  
**Shelly Plus 1PM**  
**Shelly Plus H&T**  
**Shelly Plus Mini 1**

mit der originalen Software, **kein Tasmota!** Weitere Modelle von Shelly werden folgen. Es handelt sich immer um die „Shelly Plus“ Modelle. Das ist wichtig, da sie eine andere API besitzen. (Gen2)

Weiterhin können alle Relais, die die aktuelle TASMOTA Firmware aufgespielt haben, benutzt werden. ( Im Moment nur Kontakt 1, falls das Relais mehrere Kontakte hat. )

Bei der Konfiguration des Relais bitte unbedingt darauf achten, dass es eine feste IP Adresse bekommt. Das ist ganz wichtig. Bluetooth und MQTT, sowie die Cloud können abgeschaltet werden. Diese Zusätze werden nicht benötigt.

Zusätzlich kann jetzt auch ein MPP-Solar Wechselrichter direkt angesteuert werden, wenn man z.B. die angeschlossene Batterie bei niedrigem Strompreis laden möchte. Weitere Wechselrichter werden folgen.

## Die INI Datei:

Die INI Datei ist eine einfache TEXT Datei, die mit jedem Linux-Editor editiert werden kann. Die einstellbaren Werte sind in Gruppen gegliedert.

```
[Allgemein]
Konfiguration      = "Datei"                ; Datei oder Datenbank
Datenbank          = "sgs.sqlite3"          ; Datenbank Dateiname
Aktivierung        = "2024-DE9995865371"    ; Freigeschaltet bis Ende 2024
Version            = "2"                  ; Bitte nicht verändern.
Stromanbieter      = "Tibber"              ; Tibber oder Awattar
Logdatei           = "sgs.log"             ; Logdatei mit Pfad
Tracelevel         = "2"                  ; 1 - 4 normal ist 2 oder 3
Aufschlag          = "0"                  ; Aufschlag von Awattar in
                                        ; Cent => "21"
Land               = "DE"                 ; Land (nur bei Awattar)
                                        ; DE oder AT
Token              = "UU2Xn00ZL_NaI_Cwa7I7Aq9C3EFwVeFAl4HR3P8j4CA" ; (Demo)
AwattarTabelle1    = "awattarPreise"       ; Nicht verändern
TibberTabelle1     = "tibberPreise"        ; Nicht verändern
TibberTabelle2     = "tibberVerbrauch"     ; Nicht verändern
Grundgebuehr       = "0"                  ; Zusätzliche monatliche
                                        ; Grungebühren, Netzentgelte
                                        ; usw. in Euro.

[Relais]
Aktiv[1]           = "Ja"                 ; "Ja" / "Nein"
Autotimer[1]       = "Ja"                 ; "Ja" / "Nein"
Name[1]            = "Heizstab1"          ; Relais Name (frei wählbar)
Typ[1]             = "Shelly"             ; "Shelly" oder "Tasmota"
Protokoll[1]       = "http"               ; "http", "tasmota", "mpi_pi17"
IP-Adresse[1]      = "192.168.2.186"      ; IP Adresse oder USB Device
Port[1]            = "80"                 ; Port
AnzahlKontakte[1] = "1"                   ; Zurzeit nur 1 möglich
Kontaktnummer[1]  = "1"                   ; Zurzeit nur 1 möglich

[Schaltpunkte]
;      0 = Punkt wird nicht beachtet
Aktiv[1]           = "Ja"                 ; "Ja" / "Nein".
RelaisNummer[1]    = "1"                   ; Relaisnummer von oben
ZeitspanneVon[1]   = "00:00"              ; Zeit in der das Relais
                                        ; einschalten kann -
ZeitspanneDauer[1] = "1440"               ; Dauer in Minuten
                                        ; (1440 = 24 Std. maximal)
MaxPreis[1]        = "0"                   ; Maximaler Preis in Cent
MaxEinschaltzeit[1] = "120"               ; maximale Einschaltzeit
                                        ; in Minuten
MinEinschaltzeit[1] = "0"                  ; Wird zur Zeit noch nicht
                                        ; unterstützt.
```

In der INI Datei gibt es die Gruppen:

### [Allgemein]

Hier kann man die LOG Datei anders benennen und die Stromanbieter auswählen. Zur Zeit werden nur aWATTar und Tibber unterstützt. Den Eintrag der Version bitte nicht ändern! Bei Tibber muss hier der eigene Tibber API Token eingetragen werden! Den bekommt man nur als Kunde.

Konfiguration = „Datei“ oder „Datenbank“ Fehlt die INI Datei so wird die Datenbank sgs\_ini.sqlite3 als Speicher der Konfigurationsdaten angenommen. Steht hier „Datenbank“ so kann die INI Datei vorhandensein, es wird aber trotzdem die Datenbank benutzt.

Datenbank = Der Datenbankname der benutzen Datenbank für die aktuellen Werte. Eventuell mit Pfadangabe.

Aktivierung = Den Aktivierungsschlüssel bekommt man auf dem Support Server der Solaranzeige, falls er mal abgelaufen sein sollte. Die Jahreszahl bedeutet, wie lange er gültig ist. 2014 = bis 31.12.2024 23:59:59 Uhr.

Version = Versionsnummer der Steuerung.

Stromanbieter = Zur Zeit gibt es nur Tibber oder Awattar.

LogDatei = Name der LOG Datei. Eventuell mit Pfad, wenn sie nicht im Verzeichnis /var/www/html/sgs liegt.

Land = Ist nur bei Awattar wichtig. Dort kann man Deutschland oder Österreich auswählen. Tibber gibt es zur Zeit nur in Deutschland.

Token = Als Tibber Kunde benötigt man einen Token, um die aktuellen gültigen Preise abrufen zu können.

AwattarTabelle1 = Bitte Eintrag nicht ändern.

TibberTabelle1 = Bitte Eintrag nicht ändern.

TibberTabelle2 = Bitte Eintrag nicht ändern.

Grundgebuehr = Zusätzliche monatliche Grundgebühr, Netzentgelte usw. in Euro.

Hier kann das Relais benannt werden. Ein Schaltpunkt wird mit einem Relais verbunden. Mann kann hier mehrere Relais eintragen, die auf mehrere Schaltpunkte verteilt werden können.

### **[Relais]**

- Aktiv[1] = Soll das Relais aktiv sein? Mit „Ja“ oder „Nein“ beantworten. Mit „Nein“ kann das Relais hier stehenbleiben, wird aber nicht benutzt.
- Autotimer[1] = Zur Sicherheit schaltet das Relais nach 60 Minuten aus. Damit bei einem Ausfall der Steuerung das Relais nicht unendlich an bleibt. Schaltet das Relais automatisch nach 60 Minuten aus, dann wird es eine Minute später wieder angeschaltet, wenn die Mindesteinschaltzeit noch nicht erreicht ist. Falls das kurzzeitige Ausschalten stört, muss Autotimer mit „Nein“ beantwortet werden. Soll das Relais immer wieder einschalten, bei einem Stromausfall oder einer Störung der Steuerung, so muss das Relais über die WEB-Oberfläche konfiguriert werden. In diesem Fall bitte Autotimer = „Nein“ angeben. (Z.B. bei einem Schalten von einem Gefrierschrank)  
Die Shelly Plus Relais können umfangreich über das WEB-Interface Konfiguriert werden.
- Name[1] = hier kann ein frei wählbarer Name stehen. Er findet sich in der LOG Datei wieder. Normalerweise steht da Relais1. Dann funktioniert auch das Standard Dashboard. Wird das Relais anders genannt, dann muss das Dashboard auch geändert werden.
- Typ[1] = Der Relais Typ kann „Shelly“, „Tasmota“ oder „mpi\_pi17“ sein!
- Protokoll[1] = „http“ oder „usb“
- IP-Adresse[1] = Die IP Adresse des Shelly Relais. Das Relais muss eine feste IP Adresse haben. Es kann aber auch eine USB Device sein ähnlich „/dev/ttyUSB0“
- Port[1] = muss „80“ sein.
- AnzahlKontakte[1] = Wieviel Kontakte hat das Relais? Zur Zeit kann davon nur Kontakt 1 benutzt werden.
- Kontaktnummer[1] = muss „1“ im Moment sein.

## [Schaltpunkte]

- Aktiv[1] = Ist der Schaltpunkt (Formel) aktiv oder nicht. So kann man mehrere Schaltpunkte vorgeben, im Moment dürfen bis zu 3 Schaltpunkte aktiv sein.
- RelaisNummer[1] = hier muss eines der Relais angegeben werden, welches weiter oben angegeben wurde. (Zahl) Falls dort mehrere angegeben wurden, muss hier die richtige Nummer des Relais stehen. Ein Schaltpunkt kann immer nur ein Relais ein und aus schalten!
- ZeitspanneVon[1] = ab wann soll der niedrigste Preis errechnet werden? 00:00 heißt ab Mitternacht. 18:00 heißt ab 18 Uhr (default ist 00:00 Uhr)
- ZeitspanneDauer[1] = wie lange ist die Dauer der Zeitspanne in Minuten? 1440 heißt 24 Stunden was auch das Maximum ist. ZeitspanneVon[1] = 00:00 und  
ZeitspanneDauer[1] = 1440 bedeutet die Zeitspanne läuft den ganzen Tag von Mitternacht bis Mitternacht. Möchte man, dass das Relais nur Nachts schaltet, so kann man  
ZeitspanneVon[1] = 19:00 und ZeitspanneDauer[1] = 660 eintragen. Dann würde das Relais zwischen 19:00 Uhr und 6:00 Uhr schalten. Sobald nicht die default Anfangszeit von 00:00 eingegeben ist, kann die Dauer maximal 960 Minuten betragen! Das sind 16 Stunden.  
Damit habe ich aber erst einmal nur die Zeit eingegrenzt, in der das Relais schalten könnte.
- MaxPreis[1] = Dieser Eintrag ist dafür, wenn man das Relais nur bei Preisen unterhalb dieses Maximal Preis einschalten lassen möchte. Steht hier 0 ist der Punkt ausgeschaltet und wird nicht berücksichtigt. Bleibt der Preis die oben angegebene Zeitspanne oberhalb des MaxPreis, so wird das Relais überhaupt nicht, an diesem Tag, eingeschaltet. Ist der Börsenstrompreis den ganzen Tag unterhalb des MaxPreises, so bleibt das Relais den ganzen Tag eingeschaltet. Die Einschaltzeit hängt also rein vom Börsenpreis ab solange MaxEinschaltzeit[1] = 0 ist.
- MaxEinschaltzeit[1] = wenn hier 0 steht, dann muss ein MaxPreis angegeben sein und es bedeutet, dass die Einschaltzeit ausschließlich vom Börsenstrompreis abhängig ist. Steht hier ein Wert, das bedeutet das die maximale Zeit, die das Relais eingeschaltet sein kann innerhalb 24 Stunden. Nach 24 Stunden schaltet das Relais immer bis zur nächsten Einschaltzeit aus. Minimal für 1 Minute.

So würde ein 2. Schaltpunkt aussehen:

Aktiv[2] =  
RelaisNummer[2] =  
ZeitspanneVon[2] =  
ZeitspanneDauer[2] =  
MaxPreis[2] =  
MaxEinschaltzeit[2] =  
MinEinschaltzeit[2] =

aktiv dürfen bis zu 3 Schaltpunkte sein! Bitte aber erst einmal mit einem aktiven Schaltpunkt starten um ein Gefühl zu bekommen, wie die Steuerung arbeitet.

***Welche Punkte müssen unbedingt angepasst werden?***

[Relais]

Aktiv[1] = „Ja“

IP-Adresse[1] = „<IP Adresse>“

[Schaltpunkte]

Aktiv[1] = „Ja“

RelaisNummer[1] = "1"

MaxPreis = „??“                      entweder oder, oder beides...

Maxeinschaltzeit = „???“

Alle anderen Werte können erst einmal so bleiben wie vorgegeben.

## **Prüfen, ob das Relais auch erreicht wird und schaltet.**

Wird die Konfiguration mit einer INI Datei gemacht, dann muss man ja die Funktion der eingetragenen Relais testen. Das wird so durchgeführt:

Auf der Konsole folgendes eingeben:

```
cd /var/www/html/sgs
```

```
php relais.php 1          => 1 ist die Relaisnummer laut INI Datei.
```

Damit wird das Relais 1 aus der INI Datei ein und ausgeschaltet. (toggle)

Möchte man das Relais 2 nur einschalten, dann bitte „php relais.php 2 on“ eingeben. Zum Ausschalten demnach „off“ eingeben.

## **Wichtig bei Tibber!**

**In der Tibber APP dürfen nicht mehrere Standorte „Häuser“ eingetragen sein, bzw. der erste Standort muss der richtige sein.**

**Ist der Tarif bei Tibber noch nicht auf stündlichen Strompreis umgestellt, dann funktioniert auch die Steuerung nicht! In diesem Fall kann man nur die Steuerung als Demo benutzen mit den Preisen von Awatar.**

**Siehe „Konfiguration als Demo“**



## Die WEB-Konfiguration:

Die Konfiguration kann man auf 2 unterschiedlichen Wegen gestalten. Entweder mit einer INI Datei oder mit einer Datenbank. In der Installation befindet sich die Datenbank „sgs\_ini.sqlite3“ in der sind alle Konfigurationsangaben, wie sie auch in der INI Datei sind enthalten. Möchte man die Konfiguration lieber mit der Datenbank machen, so ist das jederzeit möglich. Die Umschaltung erfolgt auf 2 Wegen. Entweder man löscht die INI Datei aus dem Verzeichnis, dann wird automatisch die Datenbank benutzt oder man ändert am Anfang der INI Datei den Eintrag

Konfiguration = "Datei" in

Konfiguration = "Datenbank" um. Beides hat den gleichen Effekt.

Die WEB Oberfläche startet man mit:

<http://solaranzeige.local/sgs>

oder wenn das nicht funktioniert mit:

<http://<IP-des-Raspberry>/sgs/index.html>

Danach sollte dieses Bild erscheinen:



## Menü Einstellungen:

Nach dem Ändern von Werten bitte „Aktualisieren“ drücken, damit werden die Werte in die Datenbank übernommen.

Solaranzeige.de

### STROMPREISSTEUERUNG

Einstellungen

Aktivierung  
2024-DE9995865371

Version  
2

LogDatei  
sgs.log

Stromanbieter  
[ ]

Tracelevel  
3

Aufschlag  
0

Token  
5K4MVS-OjWhK\_4yrjOIFe1Ff

Grundgebühr  
0.0

Land  
Deutschland ▾

Aktualisieren

Zur Hauptseite

[Aktivierung]  
Den Aktivierungsschlüssel bekommt man auf dem Support Server der Solaranzeige, falls er mal abgelaufen sein sollte. Die Jahreszahl bedeutet, wie lange er gültig ist. 2014 = bis 31.12.2024 23:59:59 Uhr.

[Version]  
Versionsnummer der Steuerung. (aktuell 2)

[LogDatei]  
Name der LOG Datei. Eventuell mit Pfad, wenn sie nicht im Verzeichnis var/www/html/sgs liegt.

[Stromanbieter]  
Zur Zeit gibt es nur Tibber oder Awattar.

[Tracelevel]  
1 – 4 Normal ist 2. Es gibt an wie viele Meldungen in die LOG Datei geschrieben werden soll.  
1 = Nur fatale Fehler  
2 = Warnungen + 1  
3 = Informationen + 2  
4 = Alle Meldungen

[Aufschlag]  
Hier kann man die zusätzlichen Kosten pro

kWh in Cent eintragen. Nötig bei Awattar.

[Token]  
Tibber Kunden bekommen einen Token, damit die aktuellen Preise abgerufen werden können. Bitte hier eintragen.

[Grundgebühr]  
Grundgebühr und Netzentgelte usw. pro Monat in Euro.

[Land]

Ist nur bei Awattar wichtig. Dort kann man Deutschland oder Österreich auswählen. Tibber gibt es zur Zeit in Deutschland.

## Menü Relais:

Die Steuerung kann mehrere Schaltpunkte haben, mit jeweils einem Relais und einem Kontakt. 5 Relais sind vorab eingefügt und konfiguriert, jedoch alle auf „inaktiv“ gesetzt. Alle die man nicht benötigt kann man löschen, oder wie hier zu sehen, „inaktivieren“

Solaranzeige.de

### STROMPREISSTEUERUNG

Relais

Relaisname

Nummer

Aktiv

Autotimer

Name

Anzahl Kontakte

Typ

Protokoll

IP-Adresse

Port

Kontaktnummer

[Relais1]

Hier sind alle Relais aufgeführt die eingegeben wurden, mit dem jeweiligen Relaisnamen. Ändert man den Relai-Namen im 5. Feld so wird er auch hier geändert gezeigt.

[Nummer]

Die eindeutige Relaisnummer. Sie wird bei den Schaltpunkten später benutzt.

[Aktiv]

Ist das Relais nur konfiguriert oder wird es auch wirklich benutzt. Ja heißt, das Relais wird benutzt und von der Steuerung auch überprüft. Gibt es ein Problem damit, werden Fehlermeldungen in der LOG Datei eingetragen. Bei „Nein“ wird das Relais ignoriert, kann aber hier weiter eingetragen bleiben. Eventuell wenn es später einmal benutzt werden soll oder vorübergehend außer Betrieb ist.

[Autotimer]

Die Shelly Relais kann man so programmieren, dass sie nach einer bestimmten Zeit automatisch wieder ausschalten. Aktiviert man Autotimer mit „Ja“, dann schaltet das Shelly Relais automatisch nach 1 Stunde zur Sicherheit ab. Merkt die Steuerung das, wird es wieder eingeschaltet, hat der Raspberry jedoch z.B. keinen Strom, bleibt das Relais aus Sicherheitsgründen aus. Das soll verhindern, dass ein Relais immer eingeschaltet bleibt, wenn es ein Problem mit der Steuerung gibt. (Funktioniert derzeit nur mit den Shelly Relais.)

[Name]

Hier kann man einen Namen des Relais vergeben.

[Anzahl Kontakte]

Zur Zeit kann bei den Relais nur Kontakt 1 benutzt werden, auch wenn das Relais mehrere Kontakte hat.

[Typ]

Shelly oder Sonoff mit der original Software ( Bitte erst nachsehen, ob das Relais freigegeben ist. )

[Protokoll]

http

[IP-Adresse]

Die feste IP Adresse des Relais. Es muss eine feste IP Adresse sein! Sie darf sich nicht ändern.

[Port]

normalerweise 80

[Kontaktnummer]

Hat das Relais mehrere Kontakte, kann man zur Zeit nur Kontakt 1 benutzen.

## Menü Schaltpunkte:

Die Steuerung kann mehrere Schaltpunkte haben, mit jeweils einem Relais und einem Kontakt. 3 Schaltpunkte sind vorkonfiguriert, jedoch nicht aktiviert.

Solaranzeige.de

### STROMPREISSTEUERUNG

Schaltpunkte

1

Aktiv  
(Ja/Nein)

Nein

Schaltpunktnummer

1

Relaisnummer  
(Die dazugehörige Relaisnummer)

0

Ab Uhrzeit  
(Zeit in der das Relais einschalten kann. Format 00:00)

00:00

Max. Dauer  
(Dauer in Minuten. 1440 = 24 Std. maximal)

1440

Max. Preis  
(Maximaler Preis in Cent)

0,44

MinEinschaltzeit  
(Wird zur Zeit noch nicht unterstützt)

0

MaxEinschaltzeit  
(Maximale Einschaltzeit in Minuten)

0

Hinzufügen Aktualisieren Löschen

Zur Hauptseite

[1]

Hier sind alle Schaltpunkte aufgeführt, die konfiguriert wurden.

[Aktiv]

Ist der Schaltpunkt aktiv? „Ja“ oder „Nein“ Bei „Ja“ wird der Schaltpunkt jede Minute „abgearbeitet“, bei „Nein“ wird er ignoriert. Zur Zeit können bis zu 5 Schaltpunkte aktiviert, jedoch bis zu 30 konfiguriert werden.

[Nummer]

Die eindeutige Schaltpunkt Nummer. Kann nicht geändert werden.

[Relais]

Welches Relais soll mit diesem Schaltpunkt gesteuert werden? Es ist die Nummer aus dem Relais Menü Feld „Nummer“

[Ab Uhrzeit]

Ab wann soll die Steuerung täglich aktiv sein? 00:00 heißt ab Mitternacht. 20:00 heißt nur in der Nacht ab 20:00 Uhr.

[Max. Dauer]

Wie lange soll die Steuerung täglich aktiv sein? 1440 heißt 1440 Minuten, also den ganzen Tag lang bis Mitternacht. Wird hier mehr eingegeben, wird die Dauer auf 1440 Minuten automatisch begrenzt.

[Max. Preis]

Möchte man das Relais rein über den Preis steuern, so bitte hier den maximalen Preis in Euro eintragen. Beispiel 0.32 = 32 Cent. Das Relais wird immer eingeschaltet wenn der Strompreis 32 Cent oder darunter liegt. (Börsenstrompreis ohne Aufschläge oder sonstigem.)

[Min. Einschaltzeit]

Dieser Punkt wird zur Zeit noch nicht benutzt.

[Max. Einschaltzeit]

Wie lange soll maximal das Relais pro Tag, bei dem niedrigsten Preis, eingeschaltet sein? (In Minuten)

Beispiel 1: Steht dort 85, so wird die Steuerung die preiswertesten 85 Minuten innerhalb der eingestellten aktiven Dauer aussuchen. Die Einschaltphase kann, muss aber nicht an einem Stück sein!

Beispiel 2: steht die Dauer aus 1440 und die Maximale Einschaltzeit auf 1260 so wird das Relais sich in den 3 teuersten Stunden des Tages ausschalten. (Die Steuerung such sich die preiswertesten 1260 Min = 21 Stunden aus) Die Einschaltphase kann, muss aber nicht an einem Stück sein!

## Datenbankfelder der Konfiguration:

Wer mit der SQLite Datenbank umgehen kann, kann auch sofort mit der Datenbank die Konfiguration durchführen. Es gibt folgende Tabellen mit folgenden Datenfeldern in der Datenbank **sgs\_ini.sqlite3**

### **Tabelle: Konfiguration**

- Datenbank
- Aktivierung
- Version
- Stromanbieter
- Logdatei
- Tracelevel
- Aufschlag
- Land
- Token
- AwattarTabelle1
- TibberTabelle1
- TibberTabelle2
- Installation\_neu
- ~~AnzRelais~~
- ~~AnzSchaltpunkte~~

### **Tabelle: Relais**

- Nummer
- Aktive
- Autotimer
- Name
- AnzahlKontakte
- Typ
- Protokoll
- IP-Adresse
- Port
- Kontaktnummer

### **Tabelle: Schaltpunkte**

- Nummer
- Aktiv
- Relaisnummer
- ZeitspanneVon
- ZeitspanneDauer
- MaxPreis
- MaxEinschaltzeit
- MinEinschaltzeit

## Zusammenfassung der Steueraufgabe:

Erst einmal kann man eine Zeitspanne wählen, in dem das Relais eingeschaltet werden könnte. Im einfachsten Fall ist das ab 00:00 Uhr, 1440 Minuten = 24 Stunden lang, also den ganzen Tag. Es ist aber auch möglich, z.B. nur die Nacht von 20:00 Uhr bis 6:00 Uhr auszuwählen.

Nur diese Zeitspanne wird für die Berechnung des preiswertesten Börsenstrompreises herangezogen. In dieser Zeitspanne kann dann mit einem Maximalpreis die Einschaltdauer ausgewählt werden oder mit einer maximalen Einschaltdauer oder mit beiden Angaben. Das Relais kann in dieser Zeit einmal oder mehrmals einschalten. Je nach Preisgestaltung.

### **MaxPreis ist größer 0 eingetragen heißt:**

Wird nur der Maximalpreis angegeben, so zieht das Relais an, immer wenn der Preis unter den Maximalpreis fällt. Das kann zeitweise, immer oder nie sein! Ist die MaxEinschaltzeit und der MaxPreis angegeben, dann heißt diese Kombination, dass bei einem niedrigen Preis das Relais nur bis zur maximalen Einschaltdauer eingeschaltet ist. Danach schaltet es aus, auch wenn der Strompreis noch weiterhin preiswert bleibt. Der Preis muss in Euro mit einem Punkt dazwischen angegeben werden. Beispiel „0.17“ = 0,17 € = 17 Cent

### **MaxEinschaltzeit ist größer 0 eingetragen heißt:**

Gibt man keinen Maximalpreis an (MaxPreis = 0), sondern eine maximale Einschaltdauer, dann sucht die Steuerung die preiswertesten Zeiten aus, sodass die maximale Einschaltzeit mit einem optimalen Preis innerhalb der angegebenen Zeitspanne erreicht wird. Die Eingabe ist in Minuten. Das Relais kann dabei mehrfach ein und ausschalten. Die Zeitspanne kann der ganze Tag sein, aber auch eine Spanne von z.B. 18:00 Uhr 10 Stunden lang, also bis 3:00 Uhr Morgens.

300 Minuten = 5 Stunden. Maximal können 1440 Minuten = 24 Stunden eingetragen werden.

Später wird es noch weitere Auswahlkriterien geben. Jetzt, in der Testphase, ist die Steuerung begrenzt.

## Ergebnis an andere Steuerungen weiterreichen:

Wann die Steuerung das Relais einschaltet, also die Bedingungen erfüllt sind, wird in der Datenbank festgehalten. Die SQLite Datenbank kann man mit vielen Programmiersprachen auslesen. In der Datenbank „sgs.sqlite3“ in der Tabelle Relaisschaltungen findet man die Information, welches Relais mit welchem Kontakt angezogen bzw. abgefallen ist. Diese Information kann man so für weitere Verknüpfungen benutzen.



## Reset der Steuerung

Ist ein Schaltpunkt aktiv und man möchte diesen ändern, so muss er erst für 1 Minute ausgeschaltet werden. Also bei dem Schaltpunkt „Aktiv“ = „Nein“ eintragen, abspeichern und dann etwas mehr als eine Minute abwarten. Danach kann der Schaltpunkt für neue Aufgaben konfiguriert werden. Wird der Schaltpunkt einfach nur geändert, ohne ihn zu resetten, ist das Ergebnis für diesen Tag nicht vorhersehbar.

## Update durchführen

Im Moment wird die Steuerung noch täglich erweitert. Deshalb ist es angebracht, wenn man auf Probleme stößt, zuerst ein Update zu machen, so wie hier beschrieben. Dafür bitte folgendes eingeben:

```
sudo sgs_update
```

Konfigurationsdaten werden dadurch nicht überschrieben. Ein Update kann man jederzeit machen, ist aber normalerweise nicht nötig.

## Die LOG Datei

Die LOG Datei ist eine einfache Text Datei, in der man die auftretenden Ereignisse nachverfolgen kann. Die Menge der Einträge kann in der Konfiguration unter dem Eintrag „Tracelevel“ eingestellt werden. Es gibt folgende Level:

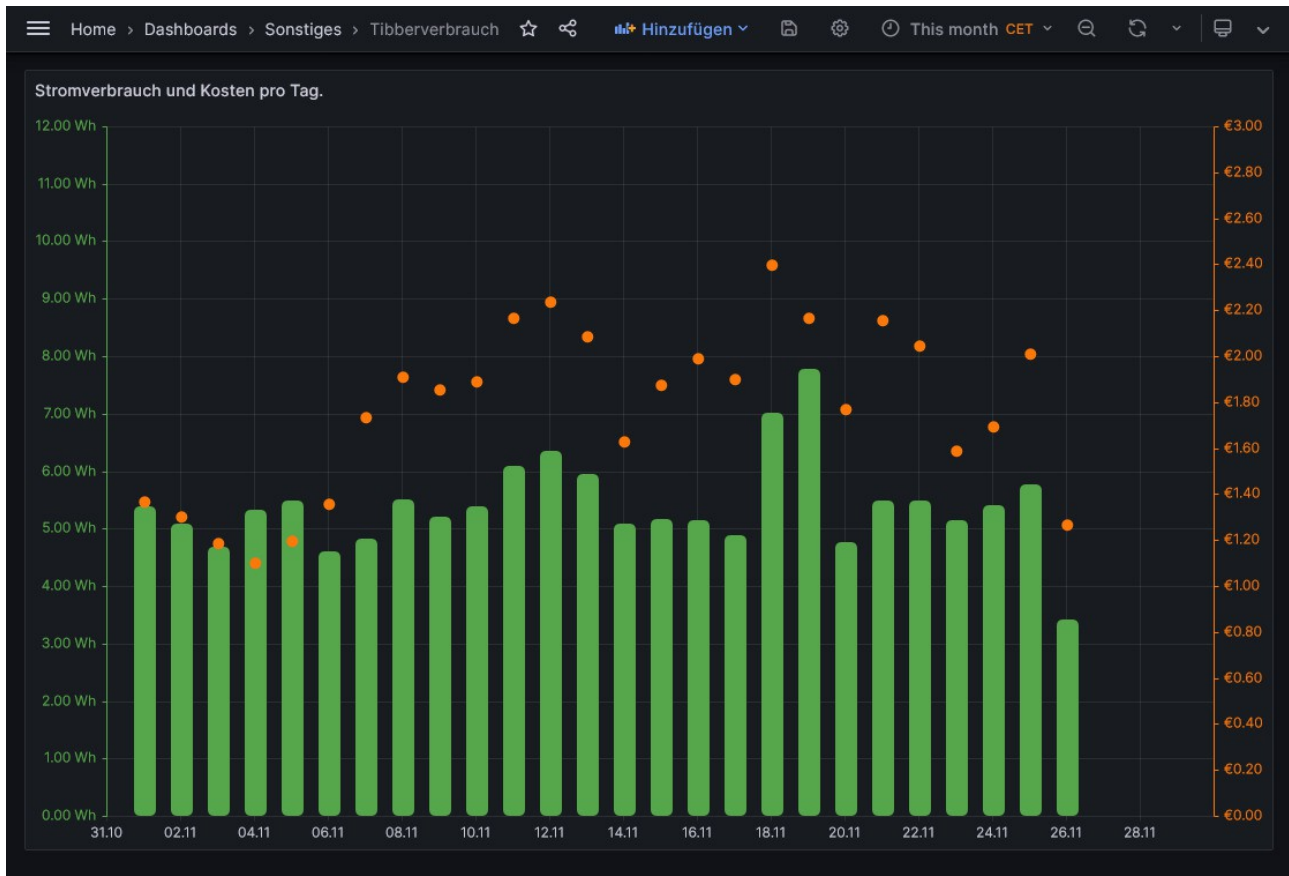
- 1 = Nur Fehler werden festgehalten
- 2 = Fehler und Warnungen
- 3 = Fehler, Warnungen und Informationen (normale Einstellung)
- 4 = Trace Informationen. Alles was Relevant sein könnte.

Die Datei findet man im Verzeichnis `„/var/www/html/sgs/sgs.log“`

## Grafische Anzeige von Verbrauch und Kosten:

Verbrauch und Kosten werden von der Tibber API ausgelesen und können so angezeigt werden. Die Dashboards sind auf dem Support Server der Solaranzeige zum Download vorhanden.

<https://solaranzeige.de/phpBB3/viewtopic.php?t=188>

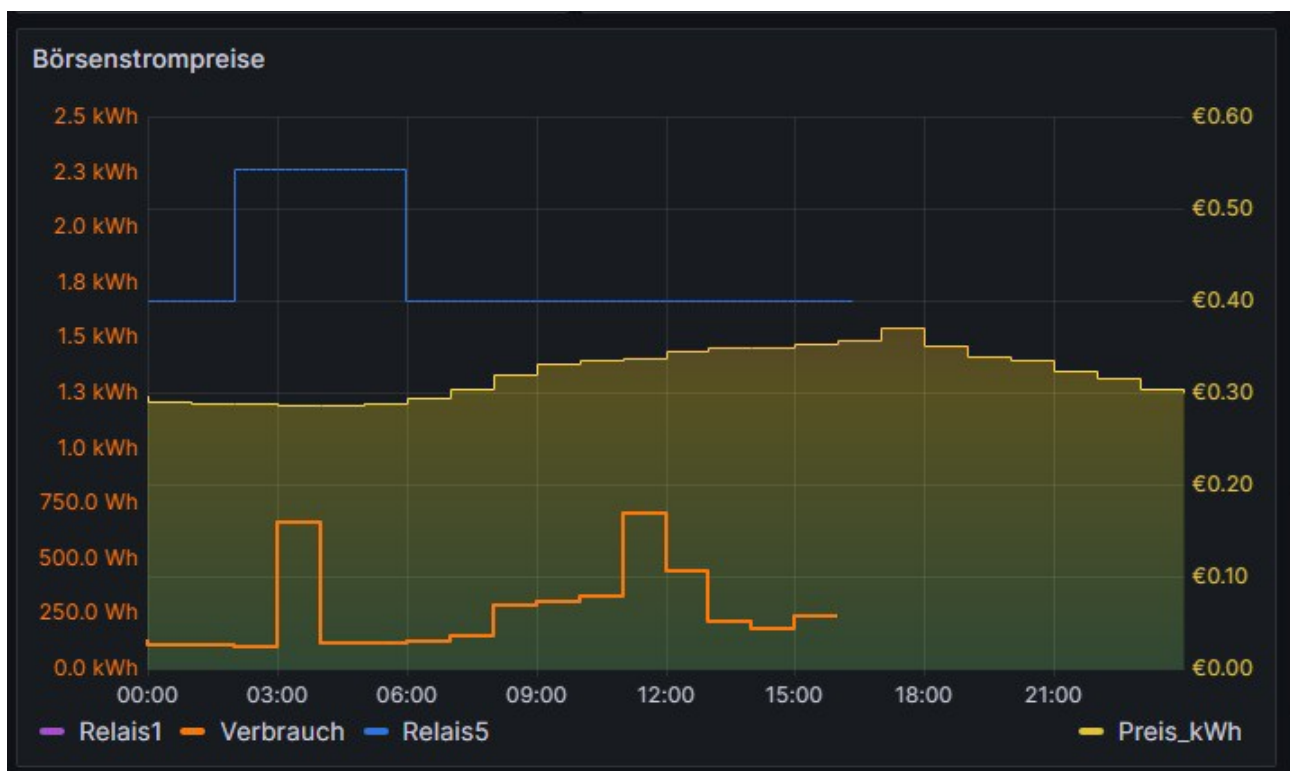


## Aufzeichnung der Schaltvorgänge

Alle Schaltvorgänge werden sofort in die Datenbank Tabelle „Relaisschaltungen“ in der Datenbank „sgs.sqlite3“ geschrieben. Zusätzlich noch alle 10 Minuten, auch wenn kein Schaltvorgang passiert ist. Diese Schaltvorgänge können dann in Grafana grafisch dargestellt werden. Dafür gibt es ein Standard Dashboard was sich auch in dem Verzeichnis /var/www/html/sgs befindet. Es muss in Grafana noch importiert werden. Die aktuelle Version kann man auch hier herunterladen:

<https://solaranzeige.de/phpBB3/download/Strompreissteuerung.json>

Außerdem muss in Grafana die Datenbank sgs.sqlite3 als neue Data Source mit dem Modul „SQLite“ angelegt werden



Möchte man die Schaltvorgänge in der Datenbank prüfen, dann folgendes eingeben:

```
sqlite3 -column -header sgs.sqlite3 "select * from Relaisschaltungen where Timestamp > STRFTIME('%s') - 3600"
```

## Batterie laden bei niedrigem Strompreis:

Gerade im Winter kann es von Vorteil sein, wenn man seine vorhandene Solarbatterie bei niedrigem Strompreis auflädt. So etwas erledigt die SGS (Strompreis geführte Steuerung) Als erster Wechselrichter kann der MPP-Solar 5048MK und viele andere baugleiche Geräte mit dem Befehl EDB0 / EDB1 (Netzladen an / aus) gesteuert werden. So können täglich die preiswertesten Stunden zum Laden benutzt werden. Angedacht sind noch Parameter zu berücksichtigen, die erfüllt sein müssen um die Ladung zu starten.

## Support:

Support Daten kann man auf folgende Weise senden, wenn man eine Support Nummer bekommen hat. Auf der Konsole bitte folgendes eingeben:

```
sudo ./var/www/html/sgs/sgs_support <Support Nummer>
```

### *Beispiel:*

```
sudo ./var/www/html/sgs/sgs_support 1234567890
```

Danach wird die LOG Datei, die INI Datei falls vorhanden und die INI Datenbank 'sgs\_ini.sqlite3' zum Support Server übertragen. Ohne die richtige Support Nummer ist die Übertragung nicht erfolgreich, sondern es werden Fehlermeldungen ausgegeben.