

# Solaranzeige.de

Ab Software Version 4.3.1    Stand Juni 2023

## MQTT Protokoll und Mosquitto-Broker

### Inhaltsverzeichnis

Übersicht:.....	2
MQTT Daten senden:.....	2
MQTT Daten bei einer Multi-Regler-Version senden:.....	3
MQTT Daten empfangen ab Version 4.5.0 [32Bit] bzw. 5.0.0 [64Bit] [aktuelle Version]:.....	5
Daten in die Influx Datenbank schreiben.....	6
Befehle zum angeschlossenen Wechselrichter / Laderegler senden.....	6
Welche Geräte sind für den Empfang von Befehlen schon vorgesehen?.....	7
Wie kann ich ein Befehl am schnellsten testen?.....	9
Besonderheiten:.....	9

## Übersicht:

Das Image beinhaltet zusätzlich den Mosquitto-Broker. Er wird automatisch beim booten gestartet. Weitere Informationen zum Mosquitto-Broker findet Ihr im Internet.

Das MQTT (Message Queue Telemetry Transport) hat sich zu einem der wichtigsten Protokolle im Internet der Dinge entwickelt. Die Firmen IBM und Arcom Control Systems entwickelten 1999 das Protokoll im Zuge eines gemeinsamen Projekts zur Überwachung einer Ölpipeline. Innerhalb der Solaranzeige wird es benutzt, um Daten an einen Broker zu senden und von ihm zu empfangen. Dadurch können die ausgelesenen Daten zur weiteren Verarbeitung an eine Smart Home Zentrale gesendet werden. Diese kann dann Befehle an den Wechselrichter / Laderegler zurück senden und so eine komplette Steuerung der Solaranlage realisiert werden. Ab der Image Version 4.3.1 ist die Kommunikation zu und von einem Broker möglich. Man kann den lokalen Mosquitto-Broker nutzen oder einen Anderen weltweit.

## MQTT Daten senden:

In der Datei `/var/www/html/user.config.php` gibt es den Punkt `$MQTT = false;` Setzt man `$MQTT = true;` so werden alle ausgelesenen Daten des angeschlossenen Gerätes zu dem lokalen Mosquitto-Broker gesendet. Gibt es keinen angeschlossenen Client, so passiert nichts weiter. Der Broker löscht die ankommenden Daten. Alle Daten werden mit den Topic's „solaranzeige/box1/xxxx“ gesendet. Für xxxx steht dann der Name der Variable z.B. Batteriespannung.

Welche Topics gesendet werden, kann man mit folgendem Befehl auf der Raspberry Command Ebene heraus bekommen: **`mosquitto_sub -h localhost -v -t solaranzeige/box1/#`**

Angezeigt wird dann nach ca. einer Minute zum Beispiel:

```
root@solaranzeige:/var/www/html # mosquitto_sub -h localhost -v -t solaranzeige/box1/#
```

```
solaranzeige/box1/firmware 1.19  
solaranzeige/box1/produkt A052  
solaranzeige/box1/optionen 001010111110111101100110  
solaranzeige/box1/ladestatus 5  
solaranzeige/box1/batteriespannung 27.6  
solaranzeige/box1/batterieladestrom 7.3  
solaranzeige/box1/solarleistung 203.11
```

usw.

Die Bezeichnungen sind selbsterklärend. Fragen bitte im Support Forum stellen.

Die Daten können nur an andere Geräte gesendet werden. Z.B. an den ioBroker oder die HomeMatic. Sie können nicht über den Umweg von MQTT in die eigene Datenbank gespeichert werden!

Parallel dazu gibt es noch die Möglichkeit, die Daten per Solaranzeigen-API an andere Geräte zu senden.

## **MQTT Daten bei einer Multi-Regler-Version senden:**

Bei einer Multi-Regler-Version muss die 1.user.config.php den Punkt `$MQTT = true;` enthalten, sowie alle weiteren Angaben für den Broker Zugang.

Somit sendet das 1. Gerät immer die Daten als MQTT Werte, auch wenn man sie nicht benötigt. Wenn weitere Geräte MQTT Daten senden sollen, dann in deren x.user.config.php auch `$MQTT = true;` eingeben und die weiteren MQTT Daten anpassen. Möchte man nur von einem Gerät MQTT Daten senden, dann ist es am besten, wenn dieses Gerät in der 1.user.config.php beschrieben ist.

In der user.config.php sind noch weitere Variablen, die man ändern kann:

```
/*
// MQTT Protokoll      MQTT Protokoll      MQTT Protokoll      MQTT Protokoll
// Sollen alle ausgelesenen Daten mit dem MQTT Protokoll an einen
// MQTT-Broker gesendet werden? Bitte das Solaranzeige-MQTT PDF Dokument lesen.
MQTT = false;
//
// Wo ist der MQTT-Broker zu finden?
// Entweder "localhost", eine Domain oder IP Adresse "xxx.xxx.xxx.xxx"
// eintragen.
// broker.hivemq.com ist ein Test Broker   Siehe http://www.mqtt-dashboard.com/
$MQTTBroker = "localhost";
//
// Benutzer Port des Brokers. Normal ist 1883
$MQTTPort = 1883;
//
// Falls der Broker gesichert ist. Sonst bitte leer lassen.
$MQTTBenutzer = "";
$MQTTKennwort = "";
//
// Wenn man die Daten mit SSL Verschlüsselung versenden möchte.
// Dazu muss der Broker SSL Verschlüsselung können.
$MQTTSSL = false;
//
// Timeout der Übertragung zum Broker. Normal = 10 bis 60 Sekunden
$MQTTKeepAlive = 60;
//
// Topic Name des Gerätes solaranzeige/box1   (solaranzeige ist fest
// vorgegeben.)
// Man kann das Gerät nennen wie man will, nur jeden Raspberry unterschiedlich
// falls man mehrere im lokalen Netz betreibt.
$MQTTGeraet = "box1";
//
// Welche Daten sollen als MQTT Message übertragen werden? Wenn hier nichts
// aufgeführt ist, werden alle ausgelesenen Daten übertragen.
// Bitte darauf achten, dass keine Leerstellen zwischen den Variablen sind.
// Die einzelnen Variablen müssen mit einem Komma getrennt werden.
// Beispiel:   $MQTTAuswahl = "box1/Ladestatus,box1/Solarspannung"
// Werden hier Variablen eingetragen, dann werden auch nur diese Topics
// übertragen. Alle Einträge müssen klein geschrieben werden! Bei einer
// Multi-Regler-Version gilt das, wenn auch nur bei einer
// x.user.config.php hier etwas angegeben ist.
$MQTTAuswahl = "";
//
*/
```

Wird bei MQTTAuswahl z.B. nur box1/Solarspannung eingetragen, dann wird auch nur dieser Wert übertragen. Auch wenn bei einer Multi-Regler-Version config Dateien ohne Einträge dazwischen sind. Welche Werte es bei Ihrem Gerät gibt können sie mit dem Befehl **mosquitto\_sub -h localhost -v -t solaranzeige/#** sehen.

## MQTT Daten empfangen:

Ab Image Version 4.5.0 kann das Image als Multi-Regler-Version und Single-Geräte-Version installiert werden. Damit ist es auch nötig anzugeben, an welches der angeschlossenen Geräte die Befehle gesendet, bzw. in welcher Datenbank die Daten gespeichert werden sollen. Bei der Single-Geräte Version ist die Gerätenummer immer 1, bei der Multi-Regler-Version kann die Gerätenummer 1 bis 6 sein.

Eingeschaltet wird der MQTT Empfang in der **user.config.php** mit dem Eintrag `$MQTT = true;` Bei der **Multi-Regler-Version** muss die Variable in der **1.user.config.php** stehen! Auch wenn die MQTT daten für ein anderes Gerät bestimmt sind muss jedoch zusätzlich in der 1.user.config.php `$MQTT = true;` stehen.

In den Konfigurationsdateien `/var/www/html/x.user.config.php` gibt es eine weitere Variable die sich auf den Empfang bezieht:

```
//*****  
// MQTT Empfang      MQTT Empfang      MQTT Empfang      MQTT Empfang  
// Subscribing      Subscribing      Subscribing      Subscribing      Subscribing  
//*****/  
  
// Welche Daten sollen empfangen werden. Hier können die Topics, die  
// empfangen werden sollen aufgeführt werden. Dabei gibt es 2 Möglichkeiten  
// Entweder ein einzelner Wert oder eine Reihe von Werten.  
// Wichtig! Das Basis-Topic heißt immer 'solaranzeige'. Dann muss entweder befehl  
// oder anzeige kommen, dann die Gerätenummer und dann die Bezeichnung des  
// Wertes. Die Gerätenummer ist immer 1, außer bei Multi-Regler-Versionen.  
// Beispiel: solaranzeige/anzeige/1/PV-Spannung  
// In diesem Beispiel wird der Wert der PV-Spannung in die Influx Datenbank  
// geschrieben unter dem Measurement MQTT  
// oder  
// Beispiel: solaranzeige/befehl/1/POP mit Wert 00  
// Der Befehl POP00 wird zum Wechselrichter geschickt. Er wird jedoch nur  
// ausgeführt wenn es sich um einen erlaubten Befehl handelt, der in der  
// Datei "befehle.ini.php" enthalten ist.  
//  
// Beispiele:  
// $MQTTTopic[1] = "solaranzeige/befehl/1/POP";  
// $MQTTTopic[2] = "solaranzeige/befehl/1/PCP";  
// $MQTTTopic[3] = "solaranzeige/anzeige/1/Wasserboiler";  
//  
// Oder auch  
// $MQTTTopic[1] = "solaranzeige/befehl/1/#";  
// Es können so viele Topics wie benötigt aufgeführt werden. Sie müssen nur  
// durch nummeriert werden [1] bis [n]  
// Bei Multi-Regler-Versionen muss zusätzlich noch die Gerätenummer angegeben  
// werden. Weitere Informationen finden Sie auf dem Support Forum.  
  
$MQTTTopic[1] = "solaranzeige/befehl/1/#";
```

Es werden nur Daten / Befehle empfangen, deren Basis-Topic „solaranzeige“ ist und die von anderen Geräten, also nicht vom eigenen Raspberry aus, gesendet werden. Alle empfangenen Daten Topic's müssen also mit „solaranzeige/anzeige/“ und alle empfangenen Befehle mit „solaranzeige/befehl/“ beginnen. Das ist zwingend vorgeschrieben. Von welchem Broker die Daten empfangen werden sollen steht in der `./var/www/html/x.user.config.php` (x ist die Gerätenummer) Bzw. in der `user.config.php` wenn es sich um eine single Geräte Version handelt. Es kann nur der selbe Broker zum Senden und Empfangen benutzt werden! Kann das sendende Gerät die Topic's nicht nach dem vorgegebenen Muster ausgeben, ist auch der Empfang nicht möglich.

Was ist die Gerätenummer?

Bei einer Multi-Regler-Version können mehrere Geräte mit einem Raspberry Pi ausgelesen werden. Für jedes Gerät muss eine 'user.config.php' vorhanden sein, die mit einer Nummer beginnt. Die Dateien müssen nummeriert zwischen 1 und 6 sein (Z.b. 1.user.config.php und 4.user.config.php) Die Geräte werden sortiert nach dieser Nummer ausgelesen. Zahlen über 6 werden ignoriert. Diese Geräte werden nie ausgelesen. Mit dieser Nummer werden das 1. bis 6. Gerät nummeriert. (Gerätenummer)

**Bei der single Geräte Version ist die Gerätenummer immer 1.**

## *Daten in die Influx Datenbank schreiben*

Es ist möglich, Schaltzustände oder Daten fremder Geräte mit in das Dashboard aufzunehmen. Die Daten müssen als MQTT Topic zum angeschlossenen Broker gesendet werden.

Ein Beispiel: Eine Smart Home Zentrale schaltet einen Wasserboiler ein und dieses soll im Dashboard visualisiert werden. Die Smart Home zentrale sendet also den Topic „solaranzeige/anzeige/1/Wasserboiler“ mit dem Wert „ein“ im MQTT Protokoll an den angeschlossenen Broker.

Dieser Wert „ein“ wird von der Solaranzeige in die lokale Influx Datenbank des 1. Gerätes geschrieben unter dem Measurement „mqtt“ und der Spalte „Wasserboiler“. Alle Daten werden immer als „String“ in die Datenbank geschrieben. Eine Zahl 9 also als „9“

Um das zu sehen muss man auf der Raspberry Command Ebene folgende Befehle eingeben:

```
influx + ENTER
```

```
>use 'solaranzeige' + ENTER („anstatt des Datenbanknamens 'solaranzeige' bitte ihr gewählter Datenbankname eintragen. In der Multi-Regler-Version muss jedes Gerät einen anderen Datenbanknamen besitzen. Diesen Namen gibt man in der x.user.config.php an.)
```

```
>show measurements + ENTER
```

```
>select * from MQTT limit 10 + ENTER
```

Jetzt sollte man die Spalte „Wasserboiler“ mit dem Wert „ein“ sehen. Eventuell ist es aber besser anstatt die Werte „ein“ und „aus“ „1“ und „0“ zu senden, da man jeder Zahl in Grafana umwandeln und einem Textstring zuordnen kann. So könnte man die „1“ in den Textstring „Wasserboiler ein“ umsetzen.

Aber Achtung! Wenn erst einmal „ein“ gesendet wurde, kann man nicht mehr „1“ senden. Das würde erst gehen, wenn man das Measurement „MQTT“ wieder löscht. Der Influx Befehl heißt dazu „drop measurement MQTT“

## *Befehle zum angeschlossenen Wechselrichter / Laderegler senden*

Möchte man seinen Wechselrichter steuern, so können Befehle an das angeschlossene Gerät gesendet werden. Nicht jeder Wechselrichter / Laderegler ist in der Lage Befehle zu empfangen, bzw. man hat die Rechte dazu eventuell nicht oder es gibt keine Unterlagen der Befehle. Ob Ihr Gerät dazu in der Lage ist, steht im Support Forum oder kann dort erfragt werden.

Beispiel: Ich möchte den Befehl POP02 an einen Wechselrichter der Effekta AX Serie senden. Das Topic muss „solaranzeige/befehl/1/POP“ mit dem Wert 02 heißen. D.h. Bei einer single Geräte Version ist das immer eine 1 und bei der Multi-Regler-Version ist das die Nummer der x.user.config.php. So können alle möglichen Befehle an den Wechselrichter gesendet werden. Welche Befehle es gibt, müssen Sie vom Hersteller oder im Support Forum erfragen. Eventuell finden sie auch Unterlagen im Internet oder auf anderen Foren. In der Solaranzeige ist noch ein Schutzmechanismus aktiv. Man kann nicht einfach alle möglichen Befehle ausprobieren, da so ein Wechselrichter / Laderegler eventuell Schaden nehmen kann. Deshalb muss jeder Befehl, der zum Gerät durchgelassen werden soll in der „/var/www/html/befehle.ini.php“ eingetragen sein. Ist der Befehl dort nicht vorhanden, dann wird er auch

nicht zum Gerät geschickt.

Sind Stufenlose Werte erlaubt, z.B. 1 bis 100% so muss man nicht 100 Einträge in die befehle.ini.php machen, sonder kann das auch mit einem Eintrag Abdecken. Bei dem Aeconversion gibt es die Befehle „L1“ für 1% Leistung bis „L100“ für 100% Leistung. Es reicht wenn man in die befehle.ini.php den Befehl „L###“ einträgt. Dann sind alle Befehle von L1 bis L100 erlaubt.

# = Wildcard

## Welche Geräte sind für den Empfang von Befehlen schon vorgesehen?

In dieser Tabelle kann man erkennen, welche Geräte zur Zeit unterstützt werden und welche Befehle freigegeben sind:

Regler0 ist ein Dummy, um den Raspberry selber zu steuern. So kann man ihn mit MQTT Befehlen stoppen oder neu starten.

Regler	Für welche Geräte	Dateiname	Befehle	Wert
0	Das ist der Raspberry selber		RPI_STOP RPI_REBOOT	Ab Version 4.5.0
5	AEconversion Micro-Wechselrichter	aec_wechselrichter.php	L B S	1 bis 100 0 2## (Volt) ## (Ampere)
7	Steca Solarix PLI 5000-48, Effekta AX Serie, Voltronic Axpert Serie, MPPSolar PIP Serie und Baugleiche	ax_wechselrichter.php	POP	00 01 02
9	MPPSolar MPI Hybrid 3 Phasen	mpi_3phasen_serie.php	EDA EDB EDC EDD EDE EDF EDG EDH	0 1
			SEP	00 01 02
			DM	058 108
13	Joulie-16 Batterie Management System	joulie_16_bms.php	RELAY	ON OFF
			STOP START	BALANCING
29	go-e Charger Wallbox	go-e_wallbox.php	alw	0 1

Diese Tabelle ist nur eine Momentaufnahme. Die aktuelle Tabelle finden sie im Dokument „Befehle senden“ im Support Forum.

In der Standard Software werden 2 Befehle für den Raspberry selber und Regler 29, 3 Befehle für Regler Nr. 5, 3 Befehle für Regler Nr. 7, 21 Befehle für Regler 9 und 4 Befehle für Regler 13 unterstützt. Benötigen Sie Befehlsfolgen auch mit anderen Geräten, melden sie sich bitte.

Möchte man ein Steuern des Gerätes verhindern, obwohl man das MQTT Protokoll aktiviert hat, dann alle Befehle des zutreffenden Gerätes aus dieser Datei entfernen. Damit werden alle empfangenen Befehle nicht beachtet.

Eine andere Lösung, Daten in die Influx Datenbank zu schreiben bietet die Solaranzeige API. Darüber gibt es ein anderes Dokument.

## Wie kann ich ein Befehl am schnellsten testen?

Wenn alles so wie hier in der Beschreibung eingerichtet ist, gibt es 2 Möglichkeiten so ein Befehl zu testen.

1. In einem Browser folgende URL eingeben:

<http://solaranzeige.local/befehl.steuerung.php?befehl=<hier den Befehl>&id=<hier die GeräteID>>

Den Befehl „alw“ mit Wert 0 zu einem go-e Charger senden.

[http://solaranzeige.local/befehl.steuerung.php?befehl=alw\\_0&id=1](http://solaranzeige.local/befehl.steuerung.php?befehl=alw_0&id=1)

Falls der Domain Name solaranzeige.local nicht funktioniert, dann die IP Adresse dort eintragen.

Bei manchen Geräten muss zwischen Befehl und Wert ein Unterstrich stehen. Z.B.

Befehl = alw und der Wert der gesetzt werden soll = 1 Dann muss der Befehl lauten „alw\_1“

Andere Geräte haben Befehle, die z.B. POP00 heißen. Näheres im Support Forum.

Das gilt nur bei Befehlen die per http Aufruf absetzt werden.

2. Mit PUTTY sich einwählen und folgendes eingeben:

```
mosquitto_pub -t solaranzeige/befehl/<GeräteID>/alw -m 1 -q 1
```

Den Befehl „alw“ mit Wert 1 zu einem go-e Charger senden, der die GeräteID 4 hat.

```
mosquitto_pub -t solaranzeige/befehl/4/alw -m 1 -q 1
```

## Besonderheiten:

Es werden in der Regel viel mehr Daten als MQTT Protokoll versendet, als in die Influx Datenbank gespeichert. Zum Broker werden alle Daten, die ausgelesen werden gesendet, auch wenn sie zur Anzeige gar nicht oder selten benutzt werden. So werden z.B. alle Daten des Befehls QPIRI des Reglers 7 ausgelesen und versendet. Wundern sie sich nicht, wenn eine ganze Menge mehr Daten im Broker ankommen. Möchten Sie nur einige, wenige Daten zum Broker übermitteln, dann bitte diese in der x.user.config.php auflisten wie auf Seite 3 beschrieben.

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)

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