

Solaranzeige.de

Ab Software Version 4.3.1 Stand Mai 2019

MQTT Protokoll und Mosquitto-Broker

Inhaltsverzeichnis

Übersicht:.....	2
MQTT Daten senden:.....	2
MQTT Daten empfangen:.....	3
Daten in die Influx Datenbank schreiben.....	4
Befehle zum angeschlossenen Wechselrichter / Laderegler senden.....	5

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.

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.
// Achtung! SSL Verschlüsselung ist zur Zeit noch nicht implementiert!
$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 = "Ladestatus,Solarspannung,Solarstrom"
// Werden hier Variablen eingetragen, dann werden auch nur diese Topics
// übertragen.
$MQTTAuswahl = "";
//

```

Wird bei MQTTAuswahl z.B. nur Solarspannung eingetragen, dann wird auch nur dieser Wert übertragen. Welche Werte es bei Ihrem Gerät gibt können sie mit dem Befehl **mosquitto_sub -h localhost -v -t solaranzeige/box1/#** sehen [\$MQTTAuswahl = „Solarspannung“;] Siehe weiter oben.

MQTT Daten empfangen:

Ab Image Version 4.3.1 ist es auch möglich MQTT Befehle und Daten zu empfangen. Dabei unterscheidet man zwischen Befehlen, die an das angeschlossene Gerät weiter geleitet werden und Daten, die in die lokale Influx Datenbank geschrieben werden.

Um Daten empfangen zu können muss zuerst einmal die Datei „/var/www/html/mqtt.config.neu.php“ in die Datei /var/www/html/mqtt.config.php kopiert oder umbenannt werden. Sobald die Datei in dem Unterverzeichnis vorhanden ist, hat sie höhere Priorität als die user.config.php. D.h. Alle Variablen in der mqtt.config.php überschreiben die gleichlautenden Variablen in der user.config.php. Also nach dem Erstellen dieser Datei müssen alle Variablen noch einmal mit den aktuellen Einstellungen bearbeitet werden.

Außerdem gibt es weitere Variablen die sich auf den Empfang beziehen:

```

/*****
// 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 Topics ist immer solaranzeige. Dann muss entweder befehl
// oder anzeige kommen und dann die Bezeichnung des Wertes.
// Beispiel: solaranzeige/anzeige/PV-Spannung
// In diesem Beispiel wird der Wert der PV-Spannung in die Influx Datenbank
// geschrieben unter dem Measurement MQTT
// oder
// Beispiel: solaranzeige/befehl/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/POP";
// $MQTTTopic[2] = "solaranzeige/befehl/PCP";
// $MQTTTopic[3] = "solaranzeige/anzeige/Wasserboiler";
//
// Oder auch
// $MQTTTopic[1] = "solaranzeige/befehl/#";
// Es können so viele Topics wie benötigt aufgeführt werden. Sie müssen nur
// durchnummeriert werden [1] bis [n]
$MQTTTopic[1] = "solaranzeige/befehl/#";

```

Es werden nur Daten / Befehle empfangen, deren basis Topic „solaranzeige“ ist. 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/mqtt.config.php](#)“ Es kann nur der selbe Broker zum Senden und Empfangen benutzt werden!

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/Wasserboiler“ mit dem Wert „ein“ im MQTT Protokoll an den angeschlossenen Broker.

Dieser Wert „ein“ wird von der Solaranzeige in die lokale Influx Datenbank geschrieben unter dem Measurement „mqtt“ und der Spalte „Wasserboiler“.

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

```
influx + ENTER
```

```
>use solaranzeige + ENTER
```

```
>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 einen 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/POP“ mit dem Wert 02 heißen. 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 „LXXX“ einträgt. Dann sind alle Befehle von L1 bis L100 erlaubt.

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

Regler	Für welche Geräte	Dateiname	Befehle	Wert
5	AEconversion Micro-Wechselrichter	aec_wechselrichter.php	L B S	1 bis 100 0 2xx (Volt) xx (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
13	Joulie-16 Batterie Management System	joulie_16_bms.php	RELAY	ON OFF
			STOP START	BALANCING

In der Standard Software werden 3 Befehle für Regler Nr. 5, 3 Befehle für Regler Nr. 7, 19 Befehle für Regler 9 und 4 Befehle für Regler 13 unterstützt. Benötigen Sie Befehlsfolgen auch mit anderen Geräten, melden sie sich bitte.

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