Projekt: solaranzeige.de

Daten an die openWB Wallbox per MQTT senden.

Stand April 2021

Dokument OW019

Inhaltsverzeichnis

Übersicht:

Möchte man den vorhandenen Wechselrichter / Laderegler / BMS usw. von der Solaranzeige auslesen lassen und die Daten zusätzlich zur Weiterverarbeitung an die openWB weiterleiten, dann funktioniert das nur mit einer „_math“ Erweiterung. Die Erweiterung muss man sich selber schreiben oder schreiben lassen. Es gibt ein Beispiel hier im Dokument, mit dem 2 SolarMax-S Wechselrichter ausgelesen werden und dann die Daten zum openWB Wallbox gesendet werden. Ist diese Funktion eingeschaltet, können keine weiteren Daten an andere Broker gesendet werden! Auch bei einer Multi-Regler-Version nicht. D.h. Ist die openWB Funktion eingeschaltet, werden alle Daten nur an die openWB in einer speziellen Topic Version gesendet.

Warum dieser Aufwand?

Es gibt Wechselrichter, die nur von einer Software ausgelesen werden können. In so einem Fall muss man sich entscheiden. Entweder Solaranzeige oder openWB. Mit dieser Erweiterung ist es möglich, dass beide Geräte die Daten bekommen. Es gibt auch Installationen, die aus mehreren Wechselrichtern unterschiedlicher Hersteller bestehen. In so einem Umfeld können die Leistungen der verschiedenen Wechselrichter addiert und an die openWB zur Steuerung weitergesendet werden.

Wie so eine „_math“ Datei geschrieben wird steht im Dokument „EigeneErweiterungen.pdf“

Was ist zu tun?

In der user.config.php, bzw. bei einer Multi-Regler-Version in der 1.user.config.php müssen folgende Einträge gemacht werden. (In rot)

```
//  Sollen alle ausgelesenen Daten mit dem MQTT Protokoll an einen
//  MQTT-Broker gesendet werden? Bitte das Solaranzeige-MQTT PDF Dokument lesen
$MQTT = true;
//
//  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 = "<IP der openWB>";
//
//  Benutzter Port des Brokers. Normal ist 1883  mit SSL 8883
$MQTTPort = 1883;
//
//  Falls der Broker gesichert ist. Sonst bitte leer lassen.
$MQTTBenutzer = "";
$MQTTKennwort = "";
//
//  Wenn man die Daten mit SSL Verschlüsselung versenden möchte.
//  Wenn hier true steht, muss im Verzeichnis "/var/www/html/" die "cerfile"
//  'ca.crt' vorhanden sein. Nähere Einzelheiten über diese Datei findet
//  man im Internet in der Mosquitto Dokumentation.
$MQTTSSL = false;
//
//  Timeout der Übertragung zum Broker. Normal = 10 bis 60 Sekunden
$MQTTKeepAlive = 60;
//
//  Topic Name oder Nummer des Gerätes solaranzeige/1
//  oder solaranzeige/box1                    (solaranzeige ist fest vorgegeben.)
//  Man kann das Gerät nennen wie man will, nur jedes Gerät, welches Daten
//  senden soll unterschiedlich. Entwerder 1 bis 6 oder Namen Ihrer Wahl vergeben.
$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 und klein geschrieben
//  werden. Zusätzlich müssen sie den Eintrag vom $MQTTGeraet und ein Schrägstrich
//  enthalten. Das ist nötig, da mehrere Geräte an dem Raspberry hängen können.
//  Beispiel mit obigen MQTTGeraet:
//  $MQTTAuswahl = "openWB";
```

Die „_math“ Datei:

Damit die Daten in der besonderen Form zum openWB Broker gesendet werden können muss man selber eine Erweiterung dafür schreiben. Wie so eine Erweiterung geschrieben wird, steht in dem Dokument „EigeneErweiterungen.pdf“ Hier ist ein Beispiel so einer Datei abgedruckt. Es ist für eine Multi-Regler-Version mit 2 Wechselrichtern geschrieben. Die openWB nimmt Daten von Solarreglern, Batterie-Management-Systemen oder Zählern entgegen.

```
<?php

/*****************************************************************************
//  Daten an die openWB Wallbox per MQTT senden.
//  Bei einer Multi-Regler Version erst die Gesamtsumme ermitteln und
//  dann per MQTT senden
//
*****************************************************************************/

$openWB_DB[1] = "solaranzeige";    // Datenbankname des 1. Wechselrichters
$openWB_DB[2] = "solaranzeige";    // Datenbankname des 2. Wechselrichters

$openWB_MQTT = array();

for ($b = 1; $b <= count($openWB_DB); $b++) {
  $timeFilter = " time >= ".strtotime(date("Y-m-d"))."000ms and time <= ".strtotime(date("Y-m-d")." + 1day")."000ms ";
  $Measurement = "PV";
  $sql = urlencode('select * from '.$Measurement.' order by time desc limit 1');
  $ch = curl_init('http://localhost/query?db='.$openWB_DB[$b].'&precision=s&q='.$sql);
  curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
  curl_setopt($ch, CURLOPT_TIMEOUT, 15);           //timeout in second s
  curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 12);
  curl_setopt($ch, CURLOPT_PORT, 8086);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

  $Ergebnis["result"] = curl_exec($ch);
  $Ergebnis["rc_info"] = curl_getinfo ($ch);
  $Ergebnis["JSON_Ausgabe"] = json_decode($Ergebnis["result"],true,10);
  $Ergebnis["errorno"] = curl_errno($ch);
  if ($Ergebnis["rc_info"]["http_code"] == 200 or $Ergebnis["rc_info"]["http_code"] == 204) {
    $Ergebnis["Ausgabe"] = true;
  }

  curl_close($ch);
  unset($ch);

  if (!isset($Ergebnis["JSON_Ausgabe"]["results"][0]["series"])) {
    log_schreiben("Es fehlt die Datenbank solaranzeige mit dem Measurement PV oder sie ist leer.","|- ",3);
    log_schreiben("Fehler: ".$Ergebnis["JSON_Ausgabe"]["results"][0]["error"],"|- ",4);
  }
  else {
    for ($h = 1; $h < count($Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["columns"]); $h++) {
      $DB[$b][$Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["columns"][$h]] =
      $Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["values"][0][$h];
    }
    log_schreiben("Datenbank: solaranzeige DB ".print_r($DB,1),"  ",10);
  }
  $Ergebnis = array();
}


for ($b = 1; $b <= count($openWB_DB); $b++) {
  $timeFilter = " time >= ".strtotime(date("Y-m-d"))."000ms and time <= ".strtotime(date("Y-m-d")." + 1day")."000ms ";
  $Measurement = "Summen";
  $sql = urlencode('select * from '.$Measurement.' order by time desc limit 1');
  $ch = curl_init('http://localhost/query?db='.$openWB_DB[$b].'&precision=s&q='.$sql);

  curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
  curl_setopt($ch, CURLOPT_TIMEOUT, 15);           //timeout in second s
  curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 12);
```

```
  curl_setopt($ch, CURLOPT_PORT, 8086);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

  $Ergebnis["result"] = curl_exec($ch);
  $Ergebnis["rc_info"] = curl_getinfo ($ch);
  $Ergebnis["JSON_Ausgabe"] = json_decode($Ergebnis["result"],true,10);
  $Ergebnis["errorno"] = curl_errno($ch);

  if ($Ergebnis["rc_info"]["http_code"] == 200 or $Ergebnis["rc_info"]["http_code"] == 204) {
    $Ergebnis["Ausgabe"] = true;
  }

  curl_close($ch);
  unset($ch);


  if (!isset($Ergebnis["JSON_Ausgabe"]["results"][0]["series"])) {
    log_schreiben("Es fehlt die Datenbank solaranzeige mit dem Measurement PV oder sie ist leer.","|- ",3);
    log_schreiben("Fehler: ".$Ergebnis["JSON_Ausgabe"]["results"][0]["error"],"|- ",4);
  }
  else {
    for ($h = 1; $h < count($Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["columns"]); $h++) {
      $DB[$b+2][$Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["columns"][$h]] =
      $Ergebnis["JSON_Ausgabe"]["results"][0]["series"][0]["values"][0][$h];
    }
    log_schreiben("Datenbank: solaranzeige DB ".print_r($DB,1),"  ",10);
  }
  $Ergebnis = array();
}
log_schreiben("Datenbank: solaranzeige DB ".print_r($DB,1),"  ",10);


$openWB_MQTT["openWB/set/pv/1/W"] = "-".round($DB[1]["Leistung"] + $DB[2]["Leistung"]);
$openWB_MQTT["openWB/set/pv/1/WhCounter"] = $DB[3]["Wh_Heute"] + $DB[4]["Wh_Heute"];



//  Die Pipe wird geöffnet
//  Ab hier werden die MQTT Topics in die Pipe geschrieben

$fifoPath = "/var/www/pipe/pipe";

if (! file_exists($fifoPath)) {
  $funktionen->log_schreiben("Pipe wird neu erstellt.","   ",5);
  posix_mkfifo($fifoPath, 0644);
}
$fifo = fopen($fifoPath, "w+");
if (is_resource($fifo)) {
  foreach($openWB_MQTT as $key=>$wert) {
    $rc = fwrite($fifo, $key." ".$wert."\r\n");
    $funktionen->log_schreiben($key." ".$wert." rc: ".$rc,"   ",10);
  }
  $rc = fwrite($fifo,"|");
  fclose($fifo);
}

?>
```

Dieses ist nur ein Beispiel, wie man so eine „_math“ Datei erstellt. Dieses Beispiel funktioniert mit 2 SolarMax-S Wechselrichtern. Ganz am Anfang der Datei müssen noch die aktuellen Namen der beiden Datenbanken eingetragen werden.

Die Solarleistungsdaten werden aus den beiden Datenbanken der Wechselrichter ausgelesen, addiert und dann in die nötige MQTT Topic Form für die openWB gebracht. Zum Schluss werden die Daten in die Pipe geschrieben, die die Solaranzeige intern benutzt. Eine Pipe ist eine Spezialdatei, die immer „First in First out“ (FiFo) gelesen wird. Über die Pipe werden die Daten an den mqtt_prozess.php übergeben, der dann nach und nach die Daten an den Broker übergibt. In diesem Fall, der Broker der openWB Wallbox

Die openWB Wallbox kann folgende Topic's verarbeiten:

```
Für das EVU Modul:

Per MQTT zu schreiben:
openWB/set/evu/W Bezugsleistung in Watt, int, positiv Bezug, negativ Einspeisung
openWB/set/evu/APhase1 Strom in Ampere für Phase 1, float, Punkt als Trenner, positiv Bezug, negativ Einspeisung
openWB/set/evu/APhase2 Strom in Ampere für Phase 2, float, Punkt als Trenner, positiv Bezug, negativ Einspeisung
openWB/set/evu/APhase3 Strom in Ampere für Phase 3, float, Punkt als Trenner, positiv Bezug, negativ Einspeisung
openWB/set/evu/WhImported Bezogene Energie in Wh, float, Punkt als Trenner, nur positiv
openWB/set/evu/WhExported Eingespeiste Energie in Wh, float, Punkt als Trenner, nur positiv
openWB/set/evu/VPhase1 Spannung in Volt für Phase 1, float, Punkt als Trenner
openWB/set/evu/VPhase2 Spannung in Volt für Phase 2, float, Punkt als Trenner
openWB/set/evu/VPhase3 Spannung in Volt für Phase 3, float, Punkt als Trenner
openWB/set/evu/HzFrequenz Netzfrequenz in Hz, float, Punkt als Trenner


Für das PV Modul:

Per MQTT zu schreiben:
openWB/set/pv/1/W PVleistung in Watt, int, negativ
openWB/set/pv/1/WhCounter Erzeugte Energie in Wh, float, nur positiv


Für das Speicher Modul:

Per MQTT zu schreiben:
openWB/set/houseBattery/W Speicherleistung in Watt, int, positiv Ladung, negativ Entladung
openWB/set/houseBattery/WhImported Geladene Energie in Wh, float, nur positiv
openWB/set/houseBattery/WhExported Entladene Energie in Wh, float, nur positiv
openWB/set/houseBattery/%Soc Ladestand des Speichers, int, 0-100
```

Hilfe bei der Erstellung der „_math“ Datei bekommt ihr, wie immer, im Support Forum der Solaranzeige.