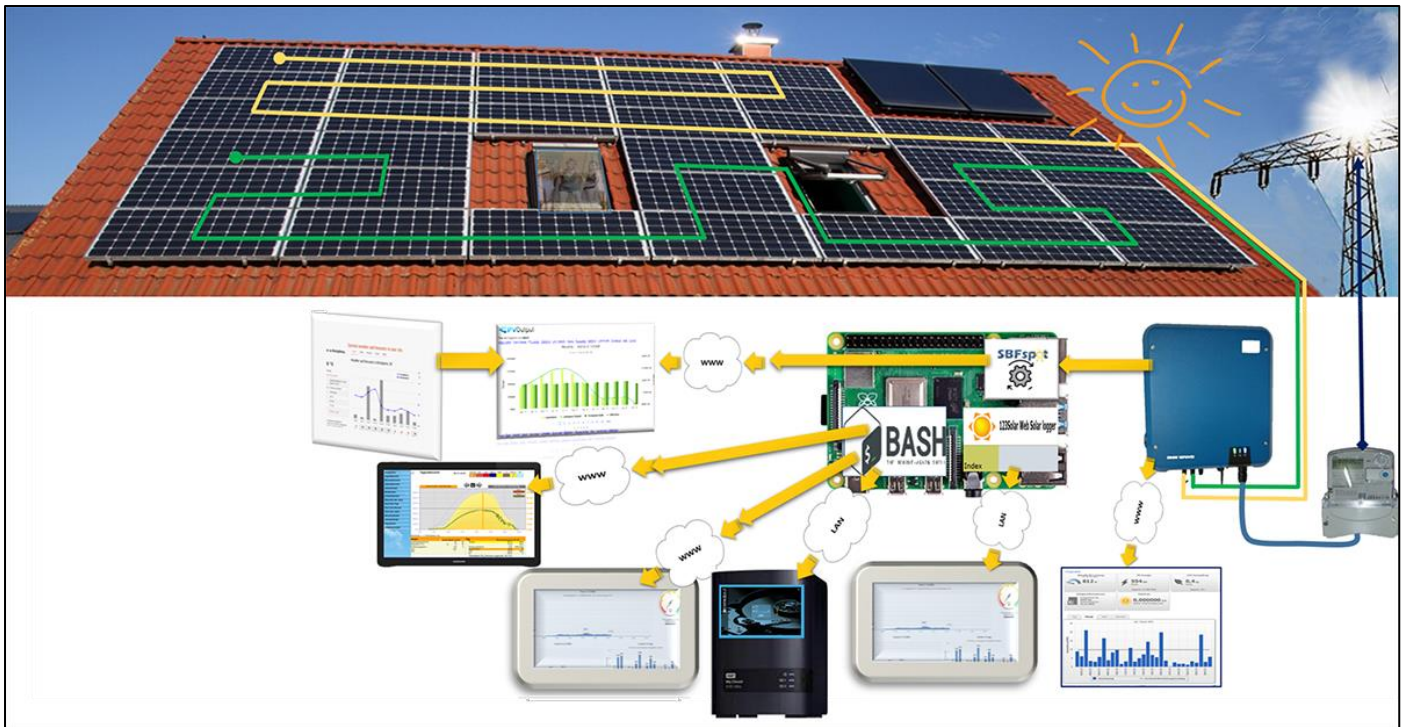




Der [Raspberry Pi](#) ist ein Minicomputer auf einer nur 9x6 cm großen Platine, der sich vielfältig erweitern lässt. Er wurde ursprünglich zu Lern- und Demonstrationszwecken entwickelt, eignet sich aber auch für die Realisierung verschiedener privater oder professioneller Anwendungen, wie z.B. einem PV-Logger mit Visualisierung via Web-Browser.

[SBFspot](#) ist ein Open-Source-Projekt, mit dem aktuelle und archivierte Daten von einem SMA®-Wechselrichter über Bluetooth oder Ethernet (Speedwire®) abgerufen werden können. Das Programm funktioniert unter Linux (Raspberry Pi) und Windows. SBFspot verbindet sich über Bluetooth oder Ethernet mit Ihrem SMA® Solar-/Batteriewechselrichter und liest Archiv-Tag/Monat-Stromerzeugung, Benutzer-/Installateur Ereignisse und die aktuellen (Spot-) Daten aus. Die gesammelten Daten werden in einer SQL-Datenbank oder SMA®-kompatiblen CSV-Dateien gespeichert. Ein separater Service/Daemon lädt die gesammelten Daten auf PVOutput hoch. Eine Liste bekannter Fehler findet man unter den dokumentierten [Problemen](#). Eine ausführliche Dokumentation und häufig gestellte Fragen finden sich in der [SBFspot-Wiki](#). Der aktuelle Quellcode wird auf den [Raspberry Pi](#) heruntergeladen, und anschließend kompiliert und gebunden.

[PVOutput](#) ist ein kostenloser Onlinedienst zum Teilen und Vergleichen der Leistungsdaten von Photovoltaik-Solarmodulen. Es bietet sowohl manuelle als auch automatische Möglichkeiten zum Hochladen von Daten. Die Ausgabedaten können grafisch dargestellt, analysiert und über verschiedene Zeiträume mit anderen Leistungsträgern verglichen werden. Durch die Möglichkeit, mit ähnlichen Systemen in unmittelbarer Nähe zu vergleichen, können kurz- und langfristige Leistungsprobleme leicht identifiziert werden.

[OpenWeatherMap](#) ist ein Online-Dienst, der eine frei nutzbare Programmierschnittstelle (API) für Wetterdaten, Wettervorhersagen sowie historische Wetterdaten für die Entwickler von Webanwendungen und mobilen Geräten bereitstellt. OpenWeatherMap verwendet [OpenStreetMap](#) zur Darstellung von Wetterkarten. Beim Upload der Daten zu Pvoutput werden von OpenWeatherMap gleichzeitig die aktuellen Wetterdaten des Anlagenstandortes bereitgestellt und zusammen mit den der Spot-Daten des Wechselrichters gespeichert.

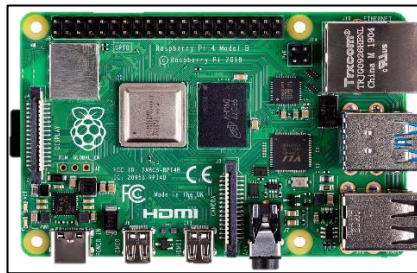
[123Solar](#) ist eine Reihe von PHP/JS-Dateien, mit denen die Daten des Photovoltaik Wechselrichter mittels Browser visualisiert und überwacht werden können. Dazu wird lediglich ein Webserver und PHP benötigt.

[OpenMediaVault](#) (OMV) ist ein freies Betriebssystem zur Realisierung eines Netzwerkspeichers (NAS). Es basiert auf der Linux-Distribution Debian.

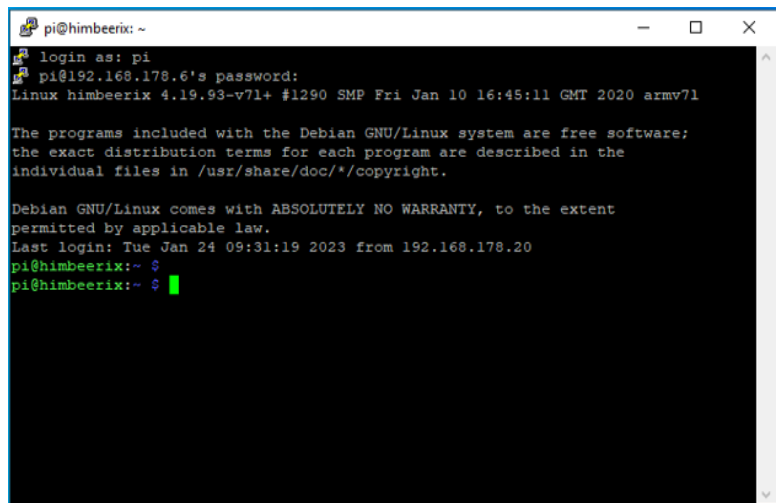
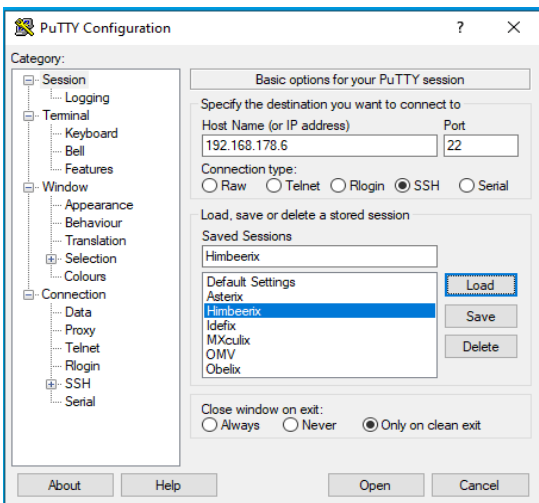
Der Solarlogger kann günstig mit einem Raspberry PI und ein wenig [MUFF](#) selbst gebastelt werden. Einfach ein paar mal mit der Maus fuchteln, einige Dateien auf die SD-Karte des Pi downloaden und die individuelle Einstellungen vornehmen. Hier in der Bastelanleitung des Solarloggers ist das detailliert beschrieben.

Raspberry PI Grundinstallation

Als erstes die neuste (gezippte) Version von **Raspian Buster Lite** auf den PC downloaden und entpacken. Dann mit dem Tool **Win32diskimager** das Image auf SD-Karte schieben. Anschließend Root-Verzeichnis der SD-Karte eine leere Datei SSH (ohne Extension !) erstellen.



Jetzt kann die SD-Karte für den ersten Start in den Raspi eingesteckt und der Raspi gebootet werden. Sobald der Raspi gestartet ist, die vom Router per DHCP vergebene IP-Adresse auslesen und mit putty auf dem Raspi anmelden (*user:pi , pw:raspberry*)



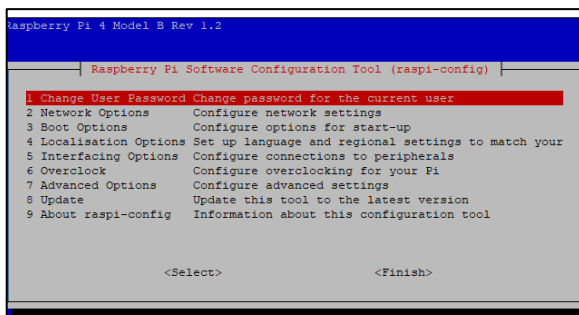
Raspi auf aktuellen Stand bringen und konfigurieren:

```
sudo apt-get update
sudo apt-get upgrade

sudo raspi-config
```

folgende Werte anpassen:

- (1) Passwort ändern: altes pw: raspberrry, neues pw: *****
- (4).(1) Länder Option- Zeichensatzändern: **de_DE.utf.8**
- (4).(2) Länder Option-Zeitzone ändern: **Europa / Berlin**
- (2).(1) Netzwerk / Hostname ändern: **Himbeerix**
- (7).(1) Dateisystem erweitern: **ok**



OpenWeatherMap und PVOutput

Die vom Logger ermittelten PV-Daten sollen zusammen mit den via [OpenWeatherMap](#) ermittelten Wetterdaten auf den Onlinedienst [PVOutput](#) hochgeladen werden. Deshalb müssen zunächst die entsprechenden Accounts erstellt werden.

OpenWeatherMap

Account erstellen und mit get API-Key den Service „Current weather and forecast“ hinzufügen. Anschließend unter „API-Keys“ einen neuen Key generieren. Dieser wird dann an der entsprechenden Stelle zusammen mit der „City ID“ im PVOutput-Profil eingetragen.

Support Center				
Weather in your city Hello mpSol Logout				
OpenWeather				
Weather Maps API Price Partners Stations Widgets News About				
New Products Services API keys Billing plans Payments Block logs Store My profile				
Name	Description	Price plan	Limits	Details
Weather	Current weather and forecast	Free plan	Hourly forecast: unavailable Daily forecast: unavailable Calls per minute: 60 3 hour forecast: 5 days	view

PVOutput

Die Daten des Wechselrichters sollen zusammen mit den aktuellen Wetterdaten des Anlagenstandortes zu [PVOutput](#) hochgeladen werden. Deshalb wird jeweils ein Account bei PVOutput und OpenWeatherMap benötigt. In der Konfigurationsdatei für den Upload SBFspotUpload.cfg müssen die Seriennummer des Wechselrichters zusammen mit der von PVOutput vergebene System Name (71661) sowie der API-Key eingetragen werden. Der API-Key von OpenWeatherMap muss wiederum in der Konfiguration von PVOutput für die Anlage eingetragen werden.

PVOutput

You are logged in as mpSol

[Add Output](#) | [Your Outputs](#) | [PV Ladder](#) | [Statistics](#) | [Live Outputs](#) | [Teams](#) | [Favourites](#) | [Settings](#) | [Community](#) | [D](#)

Account Settings

Login: mpSol

Email: [redacted]

Confidential, password resets, alerts and reports are sent to the above address.

Donation Status: **INACTIVE**

Enable bonus features with a donation. [Donations](#) are active for 1 year

New Password: [redacted]

Leave blank if you wish to keep your existing password

Retype Password: [redacted]

Data Settings

Search Min. Outputs: 0

Systems with less than the number outputs will not be shown on search results

Search and Output: [redacted]

API Settings

API Access: **Enabled** [HELP](#)

The API must be enabled to successfully process requests.

API Key: [redacted] [New Key](#)

Your API key is used to update your data automatically, always keep your API key secret

Read Only Key: [redacted]

Add your own key with read only access to your data, ideal for 3rd party apps

API Referrer: [redacted]

The URL of your webpage. Only applicable if you are embedding [portlets](#).

Registered Systems

System Name	System Id	Status	Add System
mpSol	7 [redacted]	Default	

PVOutput

You are logged in as mpSol

[Add Output](#) | [Your Outputs](#) | [PV Ladder](#) | [Statistics](#) | [Live Outputs](#) | [Teams](#) | [Favourite](#)

Edit System

System Name: mpSol

The new name of your system

Status: **Active**

If you have upgraded this system, decommission it and add a new system.

Number of Panels: 34

Total number of panels installed:

Panel Max Power: 250

Output of your panels in watts:

System Size: 8500

Your calculated system size in watts, including any additional arrays

Panel Brand/Model: asola

Brand and/or model of installed panels, e.g. Sharp NU-A188EY

Orientation: **South West**

Where your panels are facing:

Azimuth: 225.00

Optional: Orientation in degrees, this overrides the orientation setting

Number of Inverters: 1

Total number inverters installed - usually 1

Inverter Brand/Model: SMA TP 8.0

Brand and/or model of installed inverter, e.g. SMA Sunny Boy 3000

Inverter Size: 8000

Total AC power output of all inverters in watts.

Country: **Germany**

Country where your system is installed.

Automatic Uploads

Primary Device: **Weather**

Automatically download data from the above device and save it to your pvout

Poll Interval: **5 minutes**

How often to download data from the API / Device

Shift Time: **None**

Move the time reported by the device to sync with other sources

Secondary Device: **None**

A secondary device can be used to upload in the opposite or same direction

Weather Device: **OpenWeatherMap**

Upload temperature data from the external weather service provider

API Key: [redacted] [TEST](#)

Open weather map api key e.g. 9b4ee0c0fac796a50e54ca34005afd54

City Id: **2949013** [LIST](#)

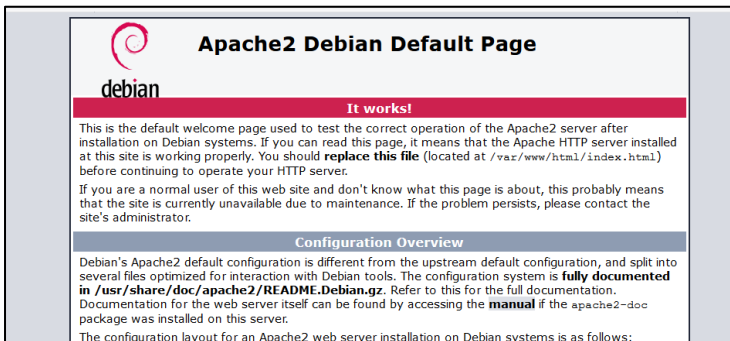
City Id to retrieve weather data from - e.g. 2949013

Zeitdienst installieren/einrichten

```
sudo apt-get install ntpdate
sudo ntpdate -u ptbtime1.ptb.de
```

```
sudo apt-get install apache2
```

Wenn jetzt die IP-Adresse des Pi im Browser eingegeben wird, sollte folgende Seite angezeigt werden:



Htaccess aktivieren

Bei Apache liegen die Webseiten unter `/var/www/`, wo dann unbedingt die HTTP-Authentifikation zugelassen werden sollte. Die Zugriffsrechte werden via `.htaccess`-Dateien geregelt. Um diese verwenden zu können, muss die `AllowOverride` Einstellung in der Apache 2 Konfiguration angepasst werden. In der Konfigurationsdatei (`apache2.conf`) muss im Feld `<Directory /var/www/>` für **AllowOverride All** gesetzt sein !

Zugriffsrechte eintragen

```
sudo nano /etc/apache2/apache2.conf
```

```
pi@himbeerix:~$ sudo nano /etc/apache2/apache2.conf
GNU nano 3.2 /etc/apache2/apache2.conf
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
    #Require all denied
    Require all granted
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Damit der user `www-data` auch den Symlinks folgen kann wurde außerdem unter `<Directory />` die Option `FollowSymLinks` von `Require all denied` auf `Require all granted` geändert.

PHP mit allen zugehörigen Paketen (aktuell Vers. 7.3) installieren

```
sudo apt-get install php7.3 php7.3-curl
```

PHP Konfiguration bearbeiten

```
sudo nano /etc/php/7.3/apache2/php.ini
```

Hier müssen die Zeitzone (date.timezone = "Europe/Berlin") sowie die dynamischen Erweiterungen (extension=calendar.so und extension=shmop.so) eingetragen werden.

```
pi@himbeerix:~$ sudo nano /etc/ php/7.3/apache2/php.ini
GNU nano 3.2 /etc/php/7.3/apache2/php.ini
[PHP]
;;;;;;;;;;;;;;;;
; About php.ini ;
;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.
;
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = "Europe/Berlin"
;
;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;
; If you wish to have an extension loaded automatically, use the following
; syntax:
;
; extension=modulename
extension=calendar.so
extension=shmop.so
```

Webserver neustarten

```
sudo service apache2 restart
```

Danach sollte durch Aufruf der info.php (<http://192.168.178.6/html/info.php>) die folgende Seite angezeigt werden:

System	Linux himbeerix 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l
Build Date	Oct 26 2019 14:14:18
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqld.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysmsg.ini, /etc/php/7.3/apache2/conf.d/20-syssem.ini, /etc/php/7.3/apache2/conf.d/20-sysshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731

SBFspot PV-Daten-Logger installieren

Zuerst werden die notwendige Tools/Bibliotheken installiert.

```
sudo apt-get install make g++ libmariadb-dev-compat libmariadb-dev
sudo apt-get install libboost-date-time-dev libboost-system-dev libboost-filesystem-dev
sudo apt-get install libboost-regex-dev libboost-all-dev
```

Bluetooth kann bei Verwendung von Speedwire weggelassen werden. Die Option `--no-install-recommends` spart Platz, da dadurch keine unnötigen Drucker/Scanner-Treiber installiert werden.

```
sudo apt-get --no-install-recommends install bluetooth libbluetooth-dev libbluetooth3
```

Verzeichnisse anlegen

```
cd /home/pi
mkdir smadata
mkdir sbfspot.3
sudo mkdir /var/log/sbfspot.3
sudo chown -R pi:pi /var/log/sbfspot.3
```

Das Verzeichnis `/home/pi/sbfspot.3` wird nur temporär als Container für den Quellcode benötigt und wird später wieder gelöscht.

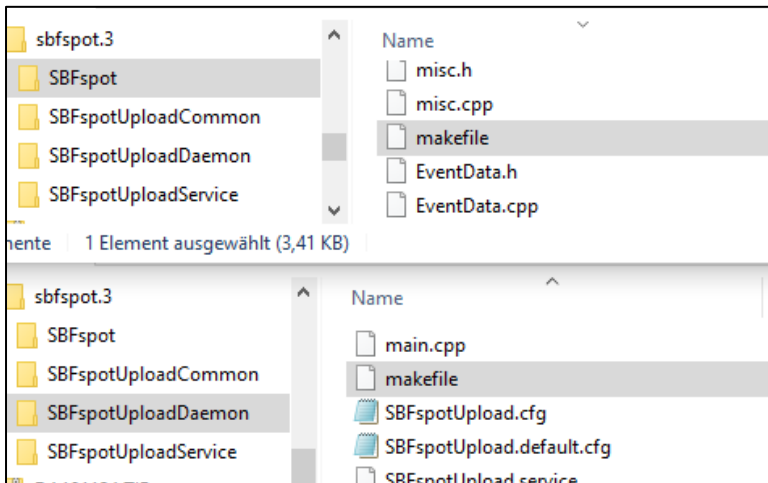
Download und Installation von SBFspot

Mit Putty am PI anmelden und folgende Kommandos eingeben:

```
cd /home/pi
wget -c https://github.com/SBFspot/SBFspot/archive/V3.6.0.tar.gz
tar -xvf V3.6.0.tar.gz -C sbfspot.3 --strip-components 1
```

Im Zielverzeichnis (/home/pi/sbfspot.3) werden durch das Entpacken vier weitere Unterverzeichnisse mit allen benötigten Dateien für das jeweilige Programm (SBFspot, SBFspotUploadDaemon/SBFspotUpload.service) erstellt.

Alternativ kann man **SBFspot** auch separat von <https://github.com/SBFspot/SBFspot/releases> downloaden und in ein leeres Verzeichnis auf dem PC entpacken. Von dort werden die vier Verzeichnisse dann mit Filezilla auf den PI ins Zielverzeichnis /home/pi/sbfspot.3 kopiert.



SBFspot mit SQLite kompilieren

In den Ordner „/usr/local/bin/sbfspot.3/SBFspot“ wechseln, SBFspot (mit SQLite-Support) kompilieren und nach /usr/local/bin/sbfspot.3 binden/installieren.

```
cd /home/pi/sbfspot.3/SBFspot
sudo make sqlite
sudo make install_sqlite
```

Leere SQLite-Datenbank anlegen

```
cd /home/pi/smadata
sqlite3 SBFspot.db < /home/pi/sbfspot.3/SBFspot/CreateSQLiteDB.sql
```

Testen ob die neu angelegte Datenbank in Ordnung ist.

```
sqlite3 SBFspot.db
```

```
pi@himbeerix:~/smadata $ sqlite3 SBFspot.db
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.
sqlite> select * from config;
SchemaVersion|1
sqlite> .quit
pi@himbeerix:~/smadata $
```

Wenn diese Antwort („SchemaVersion|1“) erscheinen, funktioniert die Datenbank.

Konfiguration SPFspot

In die die Konfigurationsdatei SBFspot.cfg werden Standortbezeichnung, Passwort (für den Zugriff auf den Wechselrichter) , IP-Adresse des WR sowie Längen- und Breitengrad des Anlagenstandortes eingetragen. Die Geodaten können mit Google-Maps oder [MAPAPI](#) ermittelt werden.

Sollen mehrere Wechselrichter abgefragt werden, muss für jeden WR eine eigene Konfigurationsdatei (z. B. SMA_TRIP8.cfg u. SMA_TRIP10.cfg) angelegt werden. Der Aufruf von SBFspot muss dann mit der Option -cfg erfolgen "SBFspot -v -cfgSMA_TRIP8.cfg -cfgSMA_TRIP10.cfg".

```
cd /usr/local/bin/sbfspot.3
sudo cp SBFspot.default.cfg SBFspot.cfg
sudo nano SBFspot.cfg
```

```
#####
#
#      / _ | | _ ) | _ | _ _ _ _ | | _
#      \ _ \ | _ \ | _ / _ | ' \ \ _ \ | _
#      _ ) | | | | _ \ \ \ | | | ( ) | | _
#      | _ | / | _ | / | | _ | _ \ \ \ \ | _
#      | _ |
# SBFspot.cfg - Configuration file for SBFspot.exe
# SBFspot - Yet another tool to read power production of SMA solar inverters
# (c)2012-2019, SBF (https://github.com/SBFspot/SBFspot)
#
# DISCLAIMER:
# A user of SBFspot software acknowledges that he or she is receiving this
# software on an as is basis and the user is not relying on the accuracy
# or functionality of the software for any purpose. The user further
# acknowledges that any use of this software will be at his own risk
# and the copyright owner accepts no responsibility whatsoever arising from
# the use or application of the software.
#
# SMA and Speedwire are registered trademarks of SMA Solar Technology AG
#####
# SBFspot.cfg - Created Do 28. Nov 14:29:50 CET 2019 with sbfspot-config V1.0.15
# For detailed configuration info, refer to SBFspot.default.cfg
#
IP_Address=192.168.178.8
# User password (default 0000)
Password=***** *(PW siehe SBFspot.cfg und/oder keePass)
Plantname=mpSol
OutputPath=/home/pi/smadata/%Y
OutputPathEvents=/home/pi/smadata/%Y/Events
CSV_Export=1
CSV_ExtendedHeader=1
CSV_Header=1
CSV_SaveZeroPower=1
CSV_Delimiter=semicolon
CSV_Spot_TimeSource=inverter
CSV_Spot_WebboxHeader=1
Timezone=Europe/Berlin
Latitude=48.912325
Longitude=8.255415
Locale=de-DE
DateTimeFormat=%d/%m/%Y %H:%M:%S
DateFormat=%d/%m/%Y
TimeFormat=%H:%M:%S
DecimalPoint=comma
CalculateMissingSpotValues=0
SQL_Database=/home/pi/smadata/SBFspot.db
# End of Config
```

Um die grundsätzliche Funktion zu prüfen wird SBFspot so gestartet, dass lediglich eine Bildschirmausgabe erfolgt. Dies wird mit folgenden Optionen erreicht:

-v# Set verbose output level: 0-5 (0=none, default=2)
-finq Force Inquiry (Inquire inverter also during the night)
-nocsv Disables CSV export (Overrules CSV_Export in config)
-nosql Disables SQL export

```
/usr/local/bin/sbfspot.3/SBFspot -v -finq -nocsv -nosql
```

```
pi@himbeerix:~$ /usr/local/bin/sbfspot.3/SBFspot -v -finq -nocsv -nosql
SBFspot V3.6.0
Yet another tool to read power production of SMA solar inverters
(c) 2012-2019, SBF (https://github.com/SBFspot/SBFspot)
Compiled for Linux (LE) 32 bit with SQLite support
Commandline Args: -v -finq -nocsv -nosql
Reading config '/usr/local/bin/sbfspot.3/SBFspot.cfg'
Wed Jan 22 09:04:00 2020: INFO: Starting...
sunrise: 08:10
sunset : 17:05
Connecting to Local Network...
Initializing...
SUSyID: 125 - SessionID: 852585912 (0x32D16DB8)
Inverter IP address: 192.168.178.8 from SBFspot.cfg
Logon OK
SUSyID: 408 - SN: 3005977634
Device Name: STP8.0-3AV-40 634
Device Class: Solar-Wechselrichter
Device Type: Sunny Tripower 8.0
Software Version: 03.00.08.R
Serial number: 3005977634
SUSyID: 408 - SN: 3005977634
Device Status: Ok
SUSyID: 408 - SN: 3005977634
Device Temperature: 0.0°C
SUSyID: 408 - SN: 3005977634
GridRelay Status: Geschlossen
SUSyID: 408 - SN: 3005977634
Pac max phase 1: 8000W
Pac max phase 2: 0W
Pac max phase 3: 0W
SUSyID: 408 - SN: 3005977634
Energy Production:
  EToday: 0.000kWh
  ETotal: 548.203kWh
  Operation Time: 564.56h
  Feed-In Time : 524.30h
SUSyID: 408 - SN: 3005977634
DC Spot Data:
  String 1 Pdc: 0.115kW - Udc: 480.95V - Idc: 0.242A
  String 2 Pdc: 0.112kW - Udc: 435.35V - Idc: 0.258A
SUSyID: 408 - SN: 3005977634
AC Spot Data:
  Phase 1 Pac : 0.062kW - Uac: 228.44V - Iac: 0.365A
  Phase 2 Pac : 0.066kW - Uac: 229.46V - Iac: 0.424A
  Phase 3 Pac : 0.073kW - Uac: 231.14V - Iac: 0.420A
  Total Pac : 0.201kW
SUSyID: 408 - SN: 3005977634
Grid Freq. : 49.99Hz
```

```

SUSyID: 408 - SN: 3005977634
Current Inverter Time: 22/01/2020 09:03:59
Inverter Wake-Up Time: 22/01/2020 08:06:07
Inverter Sleep Time : 22/01/2020 09:03:59
*****
* ArchiveDayData() *
*****
startTime = 5E278270 -> 22/01/2020 00:00:00
*****
* ArchiveMonthData() *
*****
startTime = 5E0C7BB0 -> 01/01/2020 12:00:00
Reading events: 2020-Jan-01
Wed Jan 22 09:04:00 2020: INFO: Done.

```

Verfügbare Optionen anzeigen

```
/usr/local/bin/sbfspot.3/SBFspot -?
```

```
pi@himbeerix:~$ /usr/local/bin/sbfspot.3/SBFspot -?
```

SBFspot V3.6.0

Yet another tool to read power production of SMA solar inverters

(c) 2012-2019, SBF (<https://github.com/SBFspot/SBFspot>)

Compiled for Linux (LE) 32 bit with SQLite support

SBFspot [-options]

```

-scan      Scan for bluetooth enabled SMA inverters.
-d#        Set debug level: 0-5 (0=none, default=2)
-v#        Set verbose output level: 0-5 (0=none, default=2)
-ad#       Set #days for archived daydata: 0-300
           0=disabled, 1=today (default), ...
-am#       Set #months for archived monthdata: 0-300
           0=disabled, 1=current month (default), ...
-ae#       Set #months for archived events: 0-300
           0=disabled, 1=current month (default), ...
-cfgX.Y    Set alternative config file to X.Y (multiple inverters)
-finq      Force Inquiry (Inquire inverter also during the night)
-q         Quiet (No output)
-nocsv     Disables CSV export (Overrules CSV_Export in config)
-nosql     Disables SQL export
-sp0       Disables Spot.csv export
-installer Login as installer
-password:xxxx Installer password
-loadlive  Use predefined settings for manual upload to pvoutput.org
-startdate:YYYYMMDD Set start date for historic data retrieval
-settime   Sync inverter time with host time
-mqtt      Publish spot data to MQTT broker

```

Libraries used:

SQLite V3.27.2

BOOST V1.67.0

Prüfen ob in Einträge der DB vorhanden sind

```
sqlite3 /home/pi/smadata/SBFspot.db
select * from vwspotdata
```

```
pi@himbeerix:~$ cd /home/pi/smadata
pi@himbeerix:~/smadata$ sqlite3 SBFspot.db
SQLite version 3.27.2 2019-02-25 16:06:06
Enter ".help" for usage hints.

sqlite> select * from vwspotdata;
2019-10-21 15:11:10|2014-09-21 15:10:00|SN: 2002170358|SB 1600TL-
10|2002170358|208|0|1.213|0.0|172.0|0.0|0|0|0.0|0.0|233.8|0.0|0.0|208|0|0.0|4412|3
580032|50.01|10297.1|8336.77|68.6274|OK|?|0.0

sqlite> select * from inverters;
3005977634|STP8.0-3AV-40 634|Sunny Tripower 8.0|03.00.08.R|1579769402|1482|0|553
125|574.575|533.894|OK|Closed|0.0

sqlite> .quit
```

Automatisierung

Um Daten zu sammeln, sollte SBFspot alle 5 Minuten von 06:00 bis 22:00 Uhr und einmal täglich (05:55) für monatliche Daten und Ereignisse ausgeführt werden. Dazu werden zwei Skripte und Cronjobs erstellt:

Script daydata anlegen

```
sudo nano /usr/local/bin/sbfspot.3/daydata
```

```
#!/bin/bash
#
#####
#--- daydata ---
#--- script zum Abruf der Tagesdaten
#---
#####
log=/var/log/sbfspot.3/mpSol_$(date +%Y%m%d).log
/usr/local/bin/sbfspot.3/SBFspot -v -ad1 -am0 -ae0 -cfg/usr/local/bin/sbfspot.3/SBFspot.cfg &>>$log
```

Script monthdata anlegen

```
sudo nano /usr/local/bin/sbfspot.3/monthdata
```

```
#!/bin/bash
#
#####
#--- monthdata ---
#--- script zum Abruf der Tagesdaten
#---
#####
log=/var/log/sbfspot.3/mpSol_$(date +%Y%m).log
/usr/local/bin/sbfspot.3/SBFspot -v -sp0 -ad0 -am1 -ae1 -finq -cfg/usr/local/bin/sbfspot.3/SBFspot.cfg &>>$log
```

Script ausführbar machen

```
sudo chmod +x /usr/local/bin/sbfspot.3/daydata  
sudo chmod +x /usr/local/bin/sbfspot.3/monthdata
```

Scripts daydata und monthdata in crontab eintragen

```
crontab -e
```

```
sudo chmod +x /usr/local/bin/sbfspot.3/daydata  
sudo chmod +x /usr/local/bin/sbfspot.3/monthdata
```

crontab anzeigen

```
pi@himbeerix:~$ crontab -l  
*/5 6-23 * * * /usr/local/bin/sbfspot.3/daydata  
55 05 * * * /usr/local/bin/sbfspot.3/monthdata
```

Probleme durch Verbindungsabbrüche beseitigen

Das Script **daydata** wird alle 5 Minuten im Zeitraum zwischen 06:00 Uhr und 23:00 Uhr und das Scrip **monthadata** jeden Tag um 05:55 gestartet. Nach einigen Tagen stellte sich allerdings heraus, dass beim einmaligen Aufruf von SBFspot (via daydatay bzw. monthdata) das Scrip sehr oft mit der FM: "CRITICAL: Failed to initialize Speedwire connection" abbricht. Pingt man den WR an kommt immer eine Antwort, es gibt keine Aussetzer! Wenn SBFspot in kurzen Intervallen mehrfach aufgerufen wird, antwortet der WR dann nach spätestens dem vierten/fünften Aufruf.

Laut SBFspot-Forum ([#256](#) und [#354](#)) versendet der Wechselrichter (entgegen der technischen Unterlagen) keine Multicast-Pakete. Da SBFspot sowohl von daydata per cron , als auch von 123solar alle 5 min gestartet wird kommt es hier dann zwangsweise zu Kollisionen bzw. Verbindungsabbrüchen. Um diese zu verhindern mussten die Scripts daydata und monthdata geändert werden. In dem geänderten Script wird SBFspot solange gestartet bis die Verbindung zu Stande kommt.

Script daydata neu erstellen

```
#!/bin/bash
#
#####
#--- daydata ---
#--- script zum Abruf der Tagesdaten
#---
#####
counter=0
grepCount=99
return=""

log=/var/log/sbfspot.3/mpsol_$(date +%Y%m%d).log
logTmp=/var/log/sbfspot.3/daydata_tmp.log

while [ $counter -lt 10 -a $grepCount -ne 0 ];
do
echo -e $return$counter": ***** $(date) ***** from spotdata">$logTmp
/usr/local/bin/sbfspot.3/SBFspot -v -ad1 -am0 -ae0 -finq -cfg/usr/local/bin/sbfspot.3/SBFspot.cfg &>>$logTmp
cat $logTmp >> $log
grepCount=$(grep -c -i "ERROR" $logTmp)
grepCount=$(( $grepCount + $(grep -c -i "CRITICAL" $logTmp) ))
let counter+=1
return="\n"
$(sleep 4)
done
```

Script monthdata neu erstellen

```
#!/bin/bash
#
#####
#--- monthdata ---
#--- script zum abruf der Monatsdaten
#---
#####
counter=0
grepCount=99
return=""
log=/var/log/sbfspot.3/mpSol_$(date +%Y%m').log
logtmp=/var/log/sbfspot.3/monthdata_tmp.log

while [ $counter -lt 10 -a $grepCount -ne 0 ];
do
echo -e $return$counter": ***** $(date) ***** from spotdata">$logTmp
/usr/local/bin/sbfspot.3/SBFspot -v -sp0 -ad0 -am1 -ae1 -finq -cfg/usr/local/bin/sbfspot.3/SBFspot.cfg &>>$logTmp
cat $logTmp >> $log
grepCount=$(grep -c -i "ERROR" $logTmp)
grepCount=$(( $grepCount + $(grep -c -i "CRITICAL" $logTmp) ))
let counter+=1
return="\n"
$(sleep 4)
done
```

Upload zu PVOutput einrichten

Zuerst die benötigte Library installieren. Dann in den Ordner `/usr/local/bin/sbfspot.3/SBFspotUploadDaemon` wechseln und den SBFspotUploadDaemon (mit SQLite-Support) kompilieren, binden und nach `/usr/local/bin/sbfspot.3` installieren.

```
sudo apt-get install libcurl3-dev
cd /home/pi/sbfspot.3/SBFspotUploadDaemon
sudo make sqlite
sudo make install_sqlite
```

Für die Konfiguration werden die Seriennummer des Wechselrichters sowie die System-ID und der API-Key von PVOutput benötigt.

PVoutput_SID <Seriennummer des Wechselrichters> : <Pvoutput System-ID> **(SID u. Key siehe PVOutput und/oder keePass)*

123****890634 : 4***01

PVoutput_Key <API-Key>

5e*****eafre78150d3a4a3d2f10a*****6

```
sudo nano /usr/local/bin/sbfspot.3/SBFspotUpload.cfg
```

SBFspotUpload.cfg

```
#####
# SBFspotUpload.cfg - Configuration file for SBFspotUploadService/Daemon
# (c)2012-2014, SBF (https://sbfspot.codeplex.com)
#
#####
LogDir=/home/pi/smadata/logs

#LogLevel=debug|info|warning|error (default info)
LogLevel=info

#####
### PVoutput Upload Settings ###
#####
#PVoutput_SID
#Map inverters to PVoutput System ID's
#PVoutput_SID=SerialNmbrInverter_1:PVoutput_System_ID_1,SerialNmbrInverter_2:PVoutput_System_ID_2
#e.g. PVoutput_SID=200212345:4321
PVoutput_SID=123****890634:4****01 *(SID siehe PVOutput und/oder keePass)
#PVoutput_Key
#Sets PVoutput API Key
PVoutput_Key=5e*****eafre78150d3a4a3d2f10a***** *(Key siehe PVOutput und/oder keePass)
#####
### SQL DB Settings ###
#####
# SQL_Database (Fullpath to SQLite DB)
# Windows: C:\Users\Public\SMAdat\SBFspot.db
# Linux : /home/pi/smadata/SBFspot.db
#SQL_Database=C:\Users\Public\SMAdat\SBFspot.db
SQL_Database=/home/pi/smadata/SBFspot.db

# Reserved for MySQL
#SQL_Database=SBFspot
#SQL_Hostname=<Network Name> or <IP-address>
#SQL_Username=SBFspotUser
#SQL_Password=SBFspotPassword
```

Autostart des UploadDaemon einrichten

Der UploadDaemon soll nach einem Neustart des Pi automatisch gestartet werden. Dazu wird im Verzeichnis /usr/local/bin/sbfspot.3/ die Service-Datei SBFspotUpload.service erstellt. Ab der SBFspot Version 3.5.x erfolgt der Start via system.d (bei älteren Versionen wurde via local.rc gestartet).

```
sudo nano /usr/local/bin/sbfspot.3/SBFspotUpload.service
```

SBFspotUpload.service

```
[Unit]
Description=SBFspot Upload Daemon
After=mysql.service mariadb.service network.target
[Service]
User=pi
Type=simple
TimeoutStopSec=10
ExecStart=/usr/local/bin/sbfspot.3/SBFspotUploadDaemon
Restart=on-success
RestartSec=10
[Install]
WantedBy=multi-user.target
```

Service aktivieren & Starten

```
sudo systemctl enable /usr/local/bin/sbfspot.3/SBFspotUpload.service
sudo systemctl start SBFspotUpload
```

UploadDaemon prüfen

```
sudo systemctl status SBFspotUpload
```

```
pi@himbeerix:~$ sudo systemctl status SBFspotUpload
● SBFspotUpload.service - SBFspot Upload Daemon
   Loaded: loaded (/usr/local/bin/sbfspot.3/SBFspotUpload.service; enabled; vend
   Active: active (running) since Thu 2020-06-04 17:31:03 CEST; 21min ago
   Main PID: 533 (SBFspotUploadDa)
     Tasks: 1 (limit: 4915)
    Memory: 10.9M
    CGroup: /system.slice/SBFspotUpload.service
            └─533 /usr/local/bin/sbfspot.3/SBFspotUploadDaemon
```

```
Jun 04 17:31:03 himbeerix systemd[1]: Started SBFspot Upload Daemon.
```

Upload prüfen

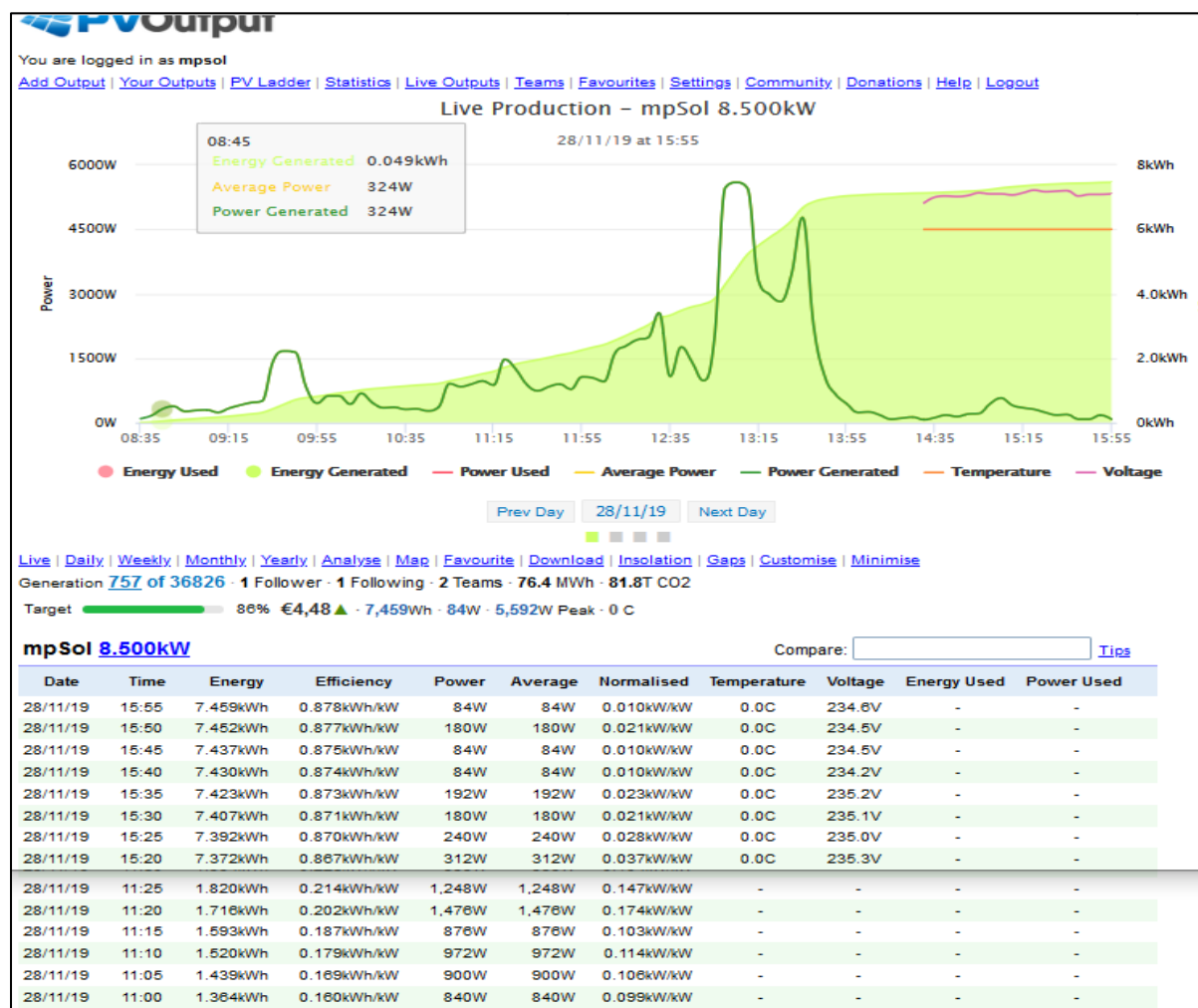
Wenn der Daemon gestartet ist im Logfile unter in /home/pi/smadata/logs nachschauen ob ein Logfile vom aktuellen Tag erstellt wurde und der Upload richtig funktioniert.

```
cd /home/pi/smadata/logs
cat SBFspotUpload20191129.log
```

```
pi@himbeerix:~$ cd /home/pi/smadata/logs
pi@himbeerix:~/smadata/logs$ ls -l
insgesamt 1312
-rw-r--r-- 1 pi pi 100414 Nov 29 23:59 SBFspotUpload20191129.log

pi@himbeerix:~/smadata/logs$ cat SBFspotUpload20191129.log
[13:36:48] INFO: SBFspotUploadDaemon Version 2.0.0
[13:36:48] INFO: Starting Daemon...
[13:36:48] INFO: Uploading 30 datapoints, starting with 20191128,11:00,94403,840 => OK (200)
[13:37:31] INFO: Uploading 30 datapoints, starting with 20191128,13:30,99306,3456 => OK (200)
```

Nach kurze Zeit sollte der Upload auch unter den Live-Daten von PVOutput angezeigt werden.



Raspi neu starten

Zu Schluss sollte der PI noch neugestartet werden. Dieses erfolgt mit dem Shutdown-Befehl als Soft Reboot, mit der Option `-r`

```
sudo shutdown -r 0
```

.... oder als Hard Reboot, mit der Option `-h`, durch Herunterfahren und anschließendem Aus- und Einschalten per Netzschalter.

```
sudo shutdown -h now
```

Probleme mit fehlerhaften Datensätzen beseitigen

Pvoutput akzeptiert beim Upload Werte von maximal 150% der Peak-Leistung was bei einer Anlage mit 8.5 KWP einem Wert von 12.750KW entspricht. Wird versucht größere Werte wie 12750 W hochzuladen, werden diese nicht akzeptiert.

Da alle Momentanwerte in der SBFspot.db gespeichert bleiben wird der SBFspot-UploadDaemon immer wieder versuchen diesen Datensatz nach pvupload.org hochzuladen. Fehlerhafte Uploads werden im entsprechenden Logfile von SBFspotUpload dokumentiert. Da ein "fehlerhafter" Datensätze den weiteren Upload verhindern, muss der betreffende Datensatz in der DB gelöscht werden. Sobald der fehlerhafte Datensatz gelöscht wurde (oder der zu hohe Wert "Power" <12750) korrigiert wurde, wird der Upload-Daemon seine Arbeit fortsetzen.

Der Fehler tritt laut SBFspot-Forum ab und zu auf und niemand kennt die genaue Ursache und deshalb wurde im Forum ein Workaround veröffentlicht.

In den nachstehenden LOG-Files vom 29.11.2019 bzw. 30.11.2019 ist der Fehler zu erkennen. SBFspot hat erstmal um 02:40 Uhr ein Peak-Wert von <1205988 Watt (also 1.2 MW) gespeichert. Der Upload hat daraufhin gestoppt.

Dieser Fehler tritt laut SBFspot-Forum [#137](#) ab und zu auf und niemand kennt die genaue Ursache. Im Forum wurde deshalb ein Workaround vorgestellt der das manuelle Löschen überflüssig macht.

Aktuelle Log Files prüfen

```
cd /home/pi/smadata/logs
```

SBFspotUpload20191129.log:

```
[13:36:48] INFO: SBFspotUploadDaemon Version 2.0.0
[13:36:48] INFO: Starting Daemon...
[13:36:48] INFO: Uploading 30 datapoints, starting with 20191128,11:00,94403,840 => OK (200)
[13:37:31] INFO: Uploading 30 datapoints, starting with 20191128,13:30,99306,3456 => OK (200)
[14:06:49] INFO: SBFspotUploadDaemon Version 2.0.0
[14:06:49] INFO: Starting Daemon...
[14:06:49] ERROR: Uploading 30 datapoints, starting with 20191129,02:40,100498,0 Bad request 400: Power value [1205988] too high for system size [8500]
```

SBFspotUpload20191130.log:

```
[00:00:31] ERROR: Uploading 30 datapoints, starting with 20191129,08:05,100498,1205988,,,0,234.03 Bad request 400: Power value [1205988] too high for system size [8500]
[00:01:30] ERROR: Uploading 30 datapoints, starting with 20191129,08:05,100498,1205988,,,0,234.03 Bad request 400: Power value [1205988] too high for system size [8500]
```

Manuelles Löschen des fehlerhaften Datensatzes

Datensatz mit zu großen Wert (z.B. >1205987) suchen und Wert auf 0 setzen:

```
cd /home/pi/smadata
sqlite3 SBFspot.db
select * from daydata where power>1205987;
update daydata set power=0 where power>1205987;
```

Workaround bei fehlerhaften Datensätzen

Die Tagesdaten (daydata) werden mit den VIEW „vwPvoData“ für den Upload vorbereitet. Der View wird so geändert, dass nur Datensätze mit korrekten Werten übernommen, fehlerhafte (zu große) Werten auf 0 gesetzt und erst dann an PVOutput übermittelt werden. Dadurch wird erreicht, dass der Upload nicht mit Fehler stehen bleibt. Der fehlerhafte Datensatz wird in der Tabelle daydata nicht gelöscht und es kann dadurch später (select * from daydata where power>8500;) nachvollzogen werden wie oft der Fehler aufgetreten ist. Theoretisch wären Werte bis 12750W möglich, in der Praxis werden jedoch bei einer Anlage von 8,5 KwP maximal 8500W auftreten. Deshalb wird als Grenzwert hier auch die tatsächlich Peak-Leistung der Anlage (8500W) eingetragen

View vwPvoData

```
SELECT dd.Timestamp,
       dd.Name,
       dd.Type,
       dd.Serial,
       dd.TotalYield AS V1,
       dd.Power AS V2,
       cons.EnergyUsed AS V3,
       cons.PowerUsed AS V4,
       spot.Temperature AS V5,
       spot.Uac1 AS V6,
       NULL AS V7,
       NULL AS V8,
       NULL AS V9,
       NULL AS V10,
       NULL AS V11,
       NULL AS V12,
       dd.PVoutput
FROM vwDayData AS dd
     LEFT JOIN vwAvgSpotData AS spot
           ON dd.Serial = spot.Serial AND dd.Timestamp = spot.Nearest5min
     LEFT JOIN vwAvgConsumption AS cons
           ON dd.Timestamp = cons.Nearest5min
ORDER BY dd.Timestamp DESC;
```

Dem VIEW vwPvoData wird die markierte Abfrage hinzugefügt. Da SQLite kein „ALTER VIEW“ kennt, muss der VIEW zuerst gelöscht und anschließend neu erstellt werden.

View vwPvoData neu erstellen

```
cd /home/pi/smadata
sqlite3 SBFspot.db

DROP VIEW IF EXISTS vwPvoData;

CREATE VIEW vwPvoData AS
  SELECT dd.Timestamp,
         dd.Name,
         dd.Type,
         dd.Serial,
         dd.TotalYield AS V1,
         CASE WHEN dd.Power > 8500
              THEN 0
              ELSE dd.Power
         END AS V2,
         cons.EnergyUsed AS V3,
         cons.PowerUsed AS V4,
         spot.Temperature AS V5,
         spot.Uac1 AS V6,
         NULL AS V7,
         NULL AS V8,
         NULL AS V9,
         NULL AS V10,
         NULL AS V11,
         NULL AS V12,
         dd.PVoutput
  FROM vwDayData AS dd
  LEFT JOIN vwAvgSpotData AS spot
    ON dd.Serial = spot.Serial AND dd.Timestamp = spot.Nearest5min
  LEFT JOIN vwAvgConsumption AS cons
    ON dd.Timestamp = cons.Nearest5min
  ORDER BY dd.Timestamp DESC;
```

Aufräumen

Wenn Alles funktioniert kann der Ordner mit dem Quellcode entfernt werden.

```
sudo rm -r /home/pi/sbfspot.3/
```

Daten-Verzeichnis bereinigen

Wenn SBFspot einige Monate lang läuft, werden Verzeichnisse mit Log-Dateien ziemlich voll. Mit den nachstehenden Aufrufen werden entsprechende Einträge in der crontab erstellt um die Log-Dateien die älter als 7 Tage sind zu löschen. Zuvor wurden diese Daten mit dem Backupsript allerdings auf die OMV gesichert.

Einträge zum Löschen der alten Log-Dateien erstellen

```
(crontab -I; echo '0 1 * * * find /var/log/sbfspot.3/ -name "*.log" -mtime +7 -delete')|crontab -
(crontab -I; echo '0 1 * * * find /home/pi/smadata/logs/ -name "*.log" -mtime +7 -delete')|crontab -
(crontab -I; echo '0 1 * * * find /home/pi/smadata/ -name "*.csv" -mtime +7 -delete')|crontab -
(crontab -I; echo '0 1 * * * find /home/pi/solarlog/ -name "*.log" -mtime +7 -delete')|crontab -
(crontab -I; echo '0 1 * * * find /home/pi/logs/ -name "*.log" -mtime +7 -delete')|crontab -
```

```
pi@himbeerix:~$  
pi@himbeerix:~$ crontab -l  
*/5 6-22 * * * /usr/local/bin/sbfspot.3/daydata  
55 05 * * * /usr/local/bin/sbfspot.3/monthdata  
  
0 1 * * * find /var/log/sbfspot.3/ -name "*.log" -mtime +7 -delete  
0 1 * * * find /home/pi/smadata/logs/ -name "*.log" -mtime +7 -delete  
0 1 * * * find /home/pi/smadata/ -name "*.csv" -mtime +7 -delete  
0 1 * * * find /home/pi/solarlog/ -name "*.*" -mtime +7 -delete  
0 1 * * * find /home/pi/logs/ -name "*.log" -mtime +7 -delete  
pi@himbeerix:~$
```

123solar PV-Web-Monitoring einrichten

Benötigte Dateien ([123solar1.8.4.tar.gz](https://github.com/jeanmarc77/123solar/releases/download/1.8.4.4/123solar1.8.4.4.tar.gz) und [123solar.service](https://github.com/jeanmarc77/123solar/blob/main/misc/examples/123solar.service)) von github.com/jeanmarc77/123solar/ (früher 123Solar.org) downloaden und im Verzeichnis `/var/www/` entpacken bzw. ins Verzeichnis `/etc/systemd/system/` kopieren.

```
cd /home/pi/tmp/  
wget -c https://github.com/jeanmarc77/123solar/releases/download/1.8.4.4/123solar1.8.4.4.tar.gz  
wget -c https://github.com/jeanmarc77/123solar/blob/main/misc/examples/123solar.service  
sudo cp /home/pi/tmp/123solar.service /etc/systemd/system/123solar.service  
cd /var/www/  
sudo tar -xzf /home/pi/123solar1.8.4.4.tar.gz
```

Verzeichnis umgruppieren

Den user `www-data` in die Gruppe `uucp` aufnehmen:

```
cd /var/www/  
sudo chown -R www-data:www-data 123solar  
sudo usermod -a -G uucp www-data
```

Konfigurationsdatei für 123solar anlegen

```
cd /etc/apache2/sites-available  
sudo nano 123solar.conf
```

```
pi@himbeerix:~$ sudo nano /etc/apache2/sites-available/123solar.conf  
  
GNU nano 3.2 /etc/apache2/sites-available/123solar.conf  
  
<VirtualHost *:80>  
    DocumentRoot /var/www/123solar  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>  
  
<Directory /var/www/123solar/>  
    AllowOverride All  
</Directory>
```

123solar in der Apache-Konfigurationsdatei eintragen

Der Apache Webserver kann aus einem oder mehreren sog. VirtualHosts bestehen. Diese Hosts können wie weitere kleine, virtuelle „Unterserver“ aufgefasst werden. Wenn der Webserver mehrere Domains bzw. Subdomains bedienen soll für jede Domain ein Virtual Host in der Konfiguration angelegt werden.

```
sudo nano /etc/apache2/sites-available /000-default.conf
```

Hier wird die Document-Root sowie die user/pw Authentifizierung für das Konfigurations-Modul von 123solar eingestellt.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost

    #1. Default: DocumentRoot /var/www/html
    #DocumentRoot /var/www/html
    #2. Funtz aber nur direkt: DocumentRoot /var/www/123solar
    #3. Funzt mit Ordnerauswahl DocumentRoot /var/www
    DocumentRoot /var/www

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

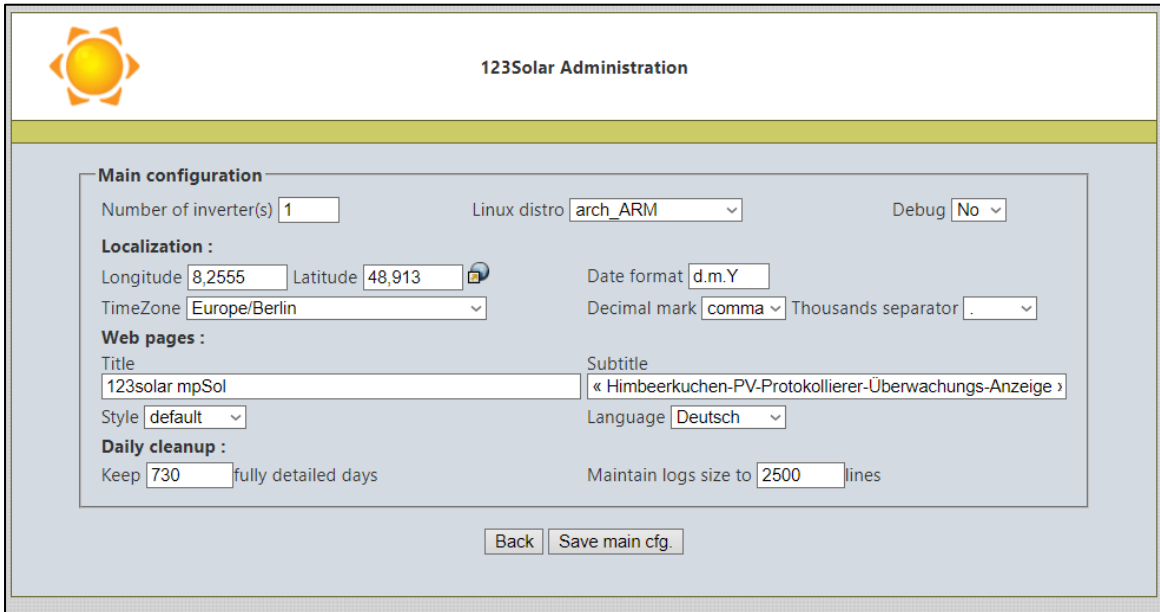
<Directory /var/www/123solar/admin/>
    AuthType Basic
    AuthName 'Restricted'
    AuthUserFile '/var/www/123solar/config/.htpasswd'
    Require valid-user
</Directory>
```

Nach Änderung den Webserver neustarten

```
sudo service apache2 restart
```

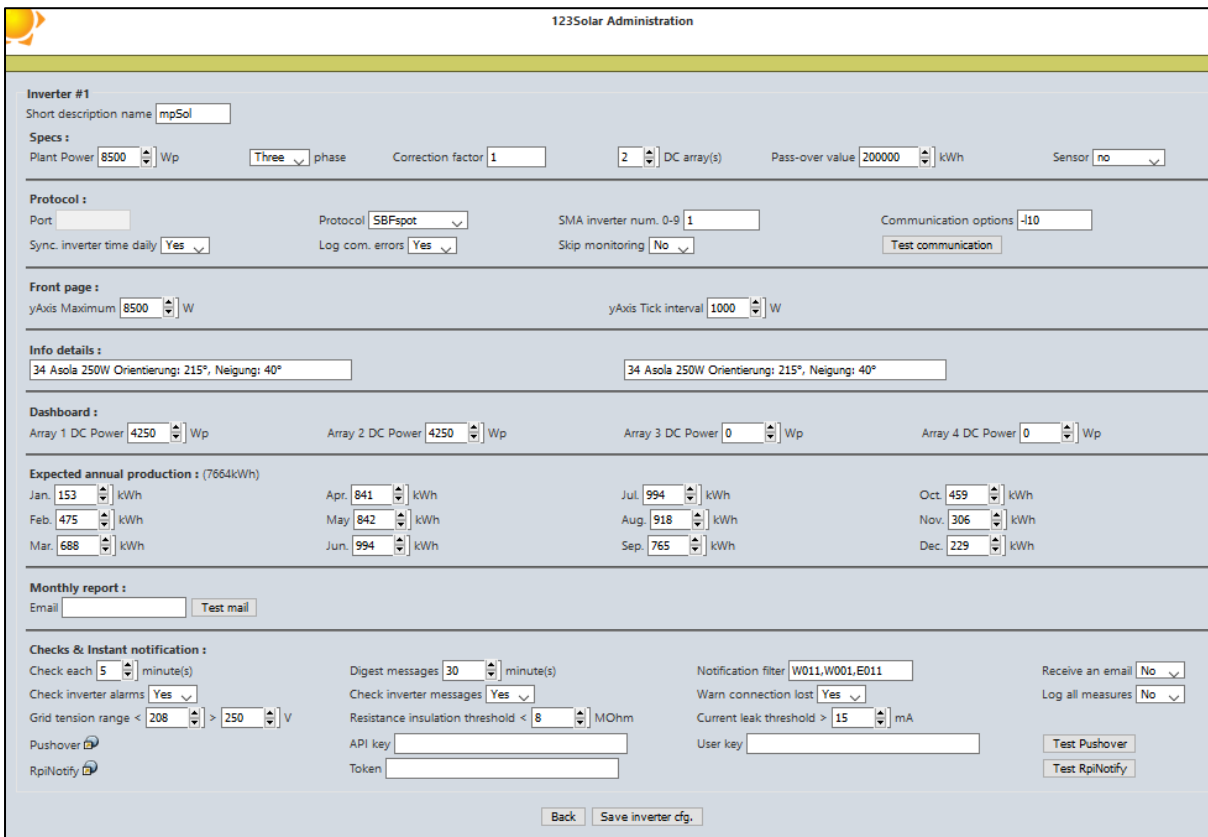
Jetzt die Seite <http://192.168.178.6/123solar/admin/admin.php> aufrufen und die Konfiguration von 123solar fertigstellen:

Grundkonfiguration (config_main.php)



The screenshot shows the '123Solar Administration' interface. At the top left is a logo of a sun with rays. The title '123Solar Administration' is centered. Below the title is a 'Main configuration' section. It contains several input fields and dropdown menus: 'Number of inverter(s)' is set to 1; 'Linux distro' is set to 'arch_ARM'; 'Debug' is set to 'No'. Under 'Localization', 'Longitude' is 8,2555, 'Latitude' is 48,913, 'Date format' is 'd.m.Y', 'Timezone' is 'Europe/Berlin', 'Decimal mark' is 'comma', and 'Thousands separator' is '.'. Under 'Web pages', 'Title' is '123solar mpSol', 'Subtitle' is '« Himbeerkuchen-PV-Protokollierer-Überwachungs-Anzeige »', 'Style' is 'default', and 'Language' is 'Deutsch'. Under 'Daily cleanup', 'Keep' is set to 730 'fully detailed days' and 'Maintain logs size to' is set to 2500 'lines'. At the bottom of the form are two buttons: 'Back' and 'Save main cfg.'.

Wechselrichter (config_inv1.php)



The screenshot shows the '123Solar Administration' interface for configuring 'Inverter #1'. The title '123Solar Administration' is centered. Below the title is a section for 'Inverter #1'. It contains several input fields and dropdown menus: 'Short description name' is 'mpSol'. Under 'Specs', 'Plant Power' is 8500 Wp, 'Three' phase, 'Correction factor' is 1, '2' DC array(s), 'Pass-over value' is 200000 kWh, and 'Sensor' is 'no'. Under 'Protocol', 'Port' is empty, 'Protocol' is 'SBFspot', 'SMA inverter num. 0-9' is 1, 'Communication options' is '-10', 'Sync. inverter time daily' is 'Yes', 'Log com. errors' is 'Yes', and 'Skip monitoring' is 'No'. There is a 'Test communication' button. Under 'Front page', 'yAxis Maximum' is 8500 W and 'yAxis Tick interval' is 1000 W. Under 'Info details', there are two identical text boxes: '34 Asola 250W Orientierung: 215°, Neigung: 40°'. Under 'Dashboard', there are four text boxes: 'Array 1 DC Power' 4250 Wp, 'Array 2 DC Power' 4250 Wp, 'Array 3 DC Power' 0 Wp, and 'Array 4 DC Power' 0 Wp. Under 'Expected annual production : (7664kWh)', there are 12 text boxes for each month: Jan. 153 kWh, Feb. 475 kWh, Mar. 688 kWh, Apr. 841 kWh, May 842 kWh, Jun. 994 kWh, Jul. 994 kWh, Aug. 918 kWh, Sep. 765 kWh, Oct. 459 kWh, Nov. 306 kWh, Dec. 229 kWh. Under 'Monthly report', there is an 'Email' input field and a 'Test mail' button. Under 'Checks & Instant notification', there are several settings: 'Check each' 5 minute(s), 'Check inverter alarms' 'Yes', 'Grid tension range' 208 V to 250 V, 'Digest messages' 30 minute(s), 'Check inverter messages' 'Yes', 'Resistance insulation threshold' 8 MOhm, 'Notification filter' 'W011,W001,E011', 'Warn connection lost' 'Yes', 'Current leak threshold' 15 mA, 'Receive an email' 'No', and 'Log all measures' 'No'. There are also fields for 'API key', 'Token', and 'User key', and buttons for 'Test Pushover' and 'Test RpiNotify'. At the bottom are two buttons: 'Back' and 'Save inverter cfg.'.

Nach dem Speichern der Konfigurations-Datei (**UND VOR DEM TEST DER COMMUNICATION!**) die SBFspot-Konfiguration des Inverter-Config anpassen (hier: SBFspot_1.cfg)

Um mehrere Wechselrichter via 123solar zu visualisieren, wird für jeden Wechselrichter ein separate Konfigurationsdatei (mit der jeweilige IP-Adresse und zugehörigem Passwort) benötigt. Die Namen der Konfigurationsdateien besteht dann aus Name des Protokolls und laufende Nummer des Wechselrichters (<PROTOKOLL>_<WR_NUMMER>.cfg). Bei 3 Wechselrichtern (Protokoll SBFspot) müssen dann die Konfigurationsdateien SBFspot_1.cfg, SBFspot_2.cfg und SBFspot_3.cfg vorhanden sein.

Kopie der Konfigurationsdatei erstellen

```
sudo cp /usr/local/bin/sbfspot.3/SBFspot.cfg /usr/local/bin/sbfspot.3/SBFspot_1.cf
```

Kommunikation testen

Durch den Kommunikationstest wird 123solar gestoppt und muss danach wieder gestartet werden!

123Solar Administration

inverter #1

description name: mpSol

Power: 8500 Wp, Three phase, Correction factor: 1, DC array(s): 2, Pass-over value: 200000 kWh, Sensor: 1

Protocol: SBFspot, SMA inverter num: 0-9: 1, Communication options: -10

inverter time daily: Yes, Log com. errors: Yes, Test communication

page: Maximum: 8500 W

details: sola 250W Orientierung: 215°, Neigung: 40°

board: 1 DC Power: 4250 Wp, Array 2 DC Power: 4250 Wp, Array 3 DC Power: 0 Wp, Array 4 DC Power: 0 Wp

predicted annual production: (7664kWh)

Monthly report: Test mail

Notifications & Instant notification: each 5 minute(s), Digest messages 30 minute(s), Notification filter: W011,W001,E011, Receive an, Check inverter messages: Yes, Warn connection lost: Yes, Log all mes, version range < 208 > 250 V, Resistance insulation threshold < 8 MOhm, Current leak threshold > 15 mA, Test Run

Communication is ok !

Back

Command

```
timeout --kill-after=15s 10s /usr/local/bin/sbfspot.3/SBFspot -finq -q -123s=DATA -cfgSBFspot_1.cfg
```

Arrays values

I1V : 507.44 V	I1A : 2.528 A	I1P : 1282 W
I2V : 539.77 V	I2A : 2.321 A	I2P : 1253 W
I3V : V	I3A : A	I3P : W
I4V : V	I4A : A	I4P : W

Grid values

G1V : 235.85 V	G1A : 3.523 A	G1P : 826 W
G2V : 234.71 V	G2A : 3.56 A	G2P : 829 W
G3V : 235.42 V	G3A : 3.595 A	G3P : 840 W

Inverter values

FRQ : 49.99 Hz	EFF : 98.422 %	INVT : 0 °
BOOT : 0 °	KWHT : 813.046 kWh	Sensor : W/m²

SBFspot Aufruf von 123solar verzögert starten

Da 123solar in der aktuellen Einstellung alle 5 Minuten, also zeitgleich mit dem cron-Aufruf von daydata ebenfalls SBFspot startet, kann es zu Netzwerk Kollisionen kommen (siehe SBFspot Probleme durch Verbindungsabbrüche). Deshalb wird im Ordner /123solar/scripts/protocols/ eine Kopie der dortigen Protokolldatei SBFspot.php als SBFspot_mp.php erstellt und so geändert, dass der der Aufruf von SBFspot um einige Sekunden verzögert wird. In der WR-Konfiguration (config_inv1.php) wird diese anschließend selektiert.

```

<?php
/**
 * /srv/http/123solar/scripts/protocols/SBFspot.php
 *
 * @package default
 */

if (!defined('checkaccess')) {
    die('Direct access not permitted');
}
// For SBFspot https://sbfspot.codeplex.com/
// Use $COMOPTION for SBFspot CSV command switches (-ad# -am# -nocsv etc)

// Timeout setup : for SBFspot timeout management by 123Solar (relies on timeout command)
$CMD_RETURN = "";

$cfgdir = realpath(' ../../config/'); // /srv/http/123solar/config/SBFspot_X.cfg

sleep(3); // mp20201025 verzögerter aufruf weil wr kein multicast protokol benutzt

if (!$DEBUG) {
    $CMD_POOLING = 'timeout --kill-after=15s 10s /usr/local/bin/sbfspot.3/SBFspot -finq -q -123s=DATA -cfg' .
    $cfgdir . "SBFspot_{$ADR'.'$invt_num}.cfg {$COMOPTION'.'$invt_num}";
} else {
    $CMD_POOLING = 'timeout --kill-after=15s 10s /usr/local/bin/sbfspot.3/SBFspot -finq -q -123s=DATA -cfg' .
    $cfgdir . "SBFspot_{$ADR'.'$invt_num}.cfg {$COMOPTION'.'$invt_num}";
    // This output is really verbose and does not respect the 123s data frame hence why you will had no data !
    // $CMD_POOLING = $timeout_setup . " SBFspot -finq -d5 -v5 -123s=DATA -cfg" . $cfgdir .
    "SBFspot_{$ADR'.'$invt_num}.cfg {$COMOPTION'.'$invt_num}";
}

$CMD_RETURN = exec($CMD_POOLING);
$dataarray = preg_split('/[[:space:]]+/', $CMD_RETURN);
if (isset($dataarray[24])) { // SBFspot might send trames shorter than 24
    if ($dataarray[24] == '>>>S123:OK') {
        $SDTE = $dataarray[0];
        $G1V = $dataarray[1]; // GridMs.PhV.phsA
        settype($G1V, 'float');
        $G1A = $dataarray[2]; // GridMs.A.phsA
        settype($G1A, 'float');
        $G1P = $dataarray[3]; // GridMs.W.phsA
        settype($G1P, 'float');
        $G2V = $dataarray[4]; // GridMs.PhV.phsB
        settype($G2V, 'float');
        $G2A = $dataarray[5]; // GridMs.A.phsB
        settype($G2A, 'float');
        $G2P = $dataarray[6]; // GridMs.W.phsB
        settype($G2P, 'float');
        $G3V = $dataarray[7]; // GridMs.PhV.phsC
        settype($G3V, 'float');
        $G3A = $dataarray[8]; // GridMs.A.phsC
        settype($G3A, 'float');
        $G3P = $dataarray[9]; // GridMs.W.phsC
        settype($G3P, 'float');
        $FRQ = $dataarray[10]; // GridMs.Hz
        settype($FRQ, 'float');
        $EFF = $dataarray[11]; // Value computed by SBFspot
        settype($EFF, 'float');
        $INVT = $dataarray[12]; // Inverter temperature - n/a for SMA inverters
        settype($INVT, 'float');
    }
}

```

```

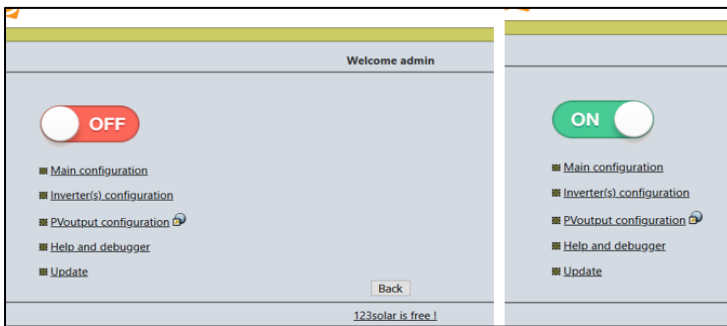
$BOOT = $dataArray[13]; // Booster temperature - n/a for SMA inverters
settype($BOOT, 'float');
$KWHT = $dataArray[14]; // Metering.TotWhOut (kWh)
settype($KWHT, 'float');
$I1V = $dataArray[15]; // DcMs.Vol[A]
settype($I1V, 'float');
$I1A = $dataArray[16]; // DcMs.Amp[A]
settype($I1A, 'float');
$I1P = $dataArray[17]; // DcMs.Watt[A]
settype($I1P, 'float');
$I2V = $dataArray[18]; // DcMs.Vol[B]
settype($I2V, 'float');
$I2A = $dataArray[19]; // DcMs.Amp[B]
settype($I2A, 'float');
$I2P = $dataArray[20]; // DcMs.Watt[B]
settype($I2P, 'float');
if ($KWHT > 0) { // Avoid null values at early startup
    $RET = 'OK';
} else {
    $RET = 'NOK';
}
} else {
    $RET = 'NOK';
}
} else {
    $RET = 'NOK';
}
}
?>

```

Wechselrichterkonfiguration (inverter.cfg)

Inverter #1			
Short description name mpSol			
Specs :	Plant Power 8500 Wp	Three phase	Correction factor 1
		2 DC array(s)	Pass-over value 20000 kWh
Protocol :			
Port	Protocol SBFspot_mp	RS485 IP adress 0	Communication options
Sync. inverter time daily Yes	Log com. errors Yes	Skip monitoring No	Test communication
Front page :			
yAxis Maximum 8500 W		yAxis Tick interval 1000 W	
Info details :			
34 Asola 250W Orientierung: 225°, Neigung: 40°		34 Asola 250W Orientierung: 225°, Neigung: 40°	
Dashboard :			
Array 1 DC Power 4250 Wp	Array 2 DC Power 4250 Wp	Array 3 DC Power 0 Wp	Array 4 DC Power 0 Wp
Expected annual production : (7652kWh)			
Jan. 153 kWh	Apr. 842 kWh	Jul. 995 kWh	Oct. 459 kWh
Feb. 459 kWh	May. 842 kWh	Aug. 918 kWh	Nov. 306 kWh
Mar. 688 kWh	Jun. 995 kWh	Sep. 765 kWh	Dec. 230 kWh
Monthly report :			
Email Test mail			
Checks & Instant notification :			
Check each 5 minute(s)	Digest messages 30 minute(s)	Notification filter W011.W001.E011	
Check inverter alarms Yes	Check inverter messages Yes	Warn connection lost Yes	
Grid tension range < 208 > 250 V	Resistance insulation threshold < 8 MOhm	Current leak threshold > 15 mA	
Pushover	API key	User key	
Rpilotify	Token		
Back Save inverter cfg.			

Nach dem Speichern der Wechselrichterkonfiguration (**UND VOR DEM TEST DER COMMUNICATION!**) die SBFspot-Konfiguration des Inverter (hier: SBFspot_1.cfg) ggf. anpassen und im Verzeichnis /usr/local/bin/sbfspot.3 bereitstellen.



PVOutput-Konfiguration (config_pvoutput.php)

Autostart von 123Solar einrichten

Damit nach einem Neustart des PI 123Solar ebenfalls automatisch gestartet wird muss der entsprechende Service erstellt werden. Einfacher gehts wenn der fertige Service von [github.com/jeanmarc77/123solar/](https://github.com/jeanmarc77/123solar/blob/main/misc/examples/123solar.service) heruntergeladen und entsprechend angepasst wird. Um einen sicheren Start zu gewährleisten sollte der Start um 10 Sekunden verzögert werden. Falls noch nicht geschehen, jetzt den [123solar.service](https://github.com/jeanmarc77/123solar/blob/main/misc/examples/123solar.service) von github.com downloaden und ins Verzeichnis /etc/systemd/system/ kopieren.

```
cd /home/pi/tmp/
wget -c https://github.com/jeanmarc77/123solar/blob/main/misc/examples/123solar.service
sudo cp /home/pi/tmp/123solar.service /etc/systemd/system/123solar.service
```

Start-Verzögerung in der Service-Datei 123solar.service eintragen

```
sudo nano /etc/systemd/system/123solar.service
```

123solar.service

```
[Unit]
Description=123Solar
Requires=network.target
After=network.target nginx.service php-fpm.service

[Service]
Type=oneshot
ExecStart=sleep 10
ExecStart=/usr/bin/curl http://localhost/123solar/scripts/boot123s.php
#ExecStart=/usr/bin/curl --insecure https://localhost/123solar/scripts/boot123s.php

[Install]
WantedBy=default.target
```

123solar Start Service aktivieren und überprüfen

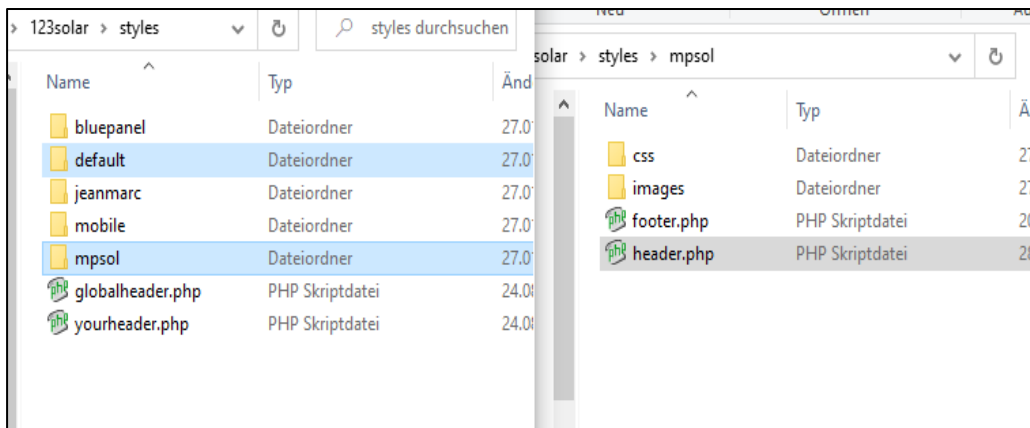
```
sudo systemctl enable 123solar
systemctl status 123solar.service
```

```
pi@himbeerix:~$ systemctl status 123solar.service
pi@himbeerix:~$ sudo systemctl enable 123solar
pi@himbeerix:~$ systemctl status 123solar
• 123solar.service - 123Solar
   Loaded: loaded (/etc/systemd/system/123solar.service; enabled; vendor preset: enabled)
   Active: inactive (dead)

Jun 04 17:31:03 himbeerix systemd[1]: Starting 123Solar...
Jun 04 17:31:13 himbeerix curl[643]: % Total % Received % Xferd Average Speed Time Time
Jun 04 17:31:13 himbeerix curl[643]: Dload Upload Total Spent
Jun 04 17:31:14 himbeerix curl[643]: [237B blob data]
Jun 04 17:31:14 himbeerix systemd[1]: 123solar.service: Succeeded.
Jun 04 17:31:14 himbeerix systemd[1]: Started 123Solar.
lines 1-10/10 (END)
pi@himbeerix:~$
```

Webseiten Header anpassen

Die Webseite von 123solar (/var/www/123solar/index.php) wird via vorgeschalteter Webseite (/var/www/index.html) aufgerufen. Um von dort zurückzukommen wird im Header von 123Solar ein zusätzliche Button „Home“ benötigt. Deshalb wird unter Styles eine Kopie des vorhandene Ordner „default“ als „mpSol“ abgelegt und dem Kopfbereich (header.php) der „Home“-Button („Home | |“) hinzugefügt.



Style kopieren und Header (mpSol) anpassen

```
sudo cp -R /var/www/123solar/styles/default /var/www/123solar/styles/mpsol
sudo nano /var/www/123solar/styles/mpsol/header.php
```

```

<head>
</head>

<body>
<table width="95%" border=0 cellpadding=0 align="center">
  <tr bgcolor="#FFFFFF" height=64>
    <td class="cadretopleft" width=128></td>
    <td class="cadretop" align="center"><b><?php echo "$TITLE";?></b><br><font size="-1"><?php echo
"$SUBTITLE";?></font></td>
    <td class="cadretoprigh" width=128 align="right"></td>
  </tr>
  <tr bgcolor="#CCCC66" valign="top">
    <td COLSPAN=3 class="cadre">
      <div class="menu">
        <font class="menu">
          <table width="100%" border=0 cellpadding=0>
            <tr><td align="left">&nbsp;<b><a href="..">Home</a></b> || <a href="index.php"><?php echo
"$lgMINDEX";?></a>
            <?php
            if ($NUMINV>1) {
              for ($i=1;$i<=$NUMINV;$i++) {
                echo " | <a href='index.php?selectinv=$i'>$lgINVT$i</a>";
              }
            }
            ?>
            | <a href="detailed.php"><?php echo "$lgMDETAILED";?></a> | <a href="production.php"><?php echo
"$lgMPRODUCTION";?></a> | <a href="comparison.php"><?php echo "$lgMCOMPARISON";?></a> | <a
href="info.php"><?php echo "$lgMINFO";?></a> || <a href='admin/'>admin</a>&nbsp;<td
align="right">&nbsp;</td>
          </tr>
        </table>
        </font>
      </div>
    </td></tr>
    <tr valign="top">
      <td COLSPAN=3 class="cadrebot" bgcolor="#d3dae2">
        <!-- #BeginEditable "mainbox" -->

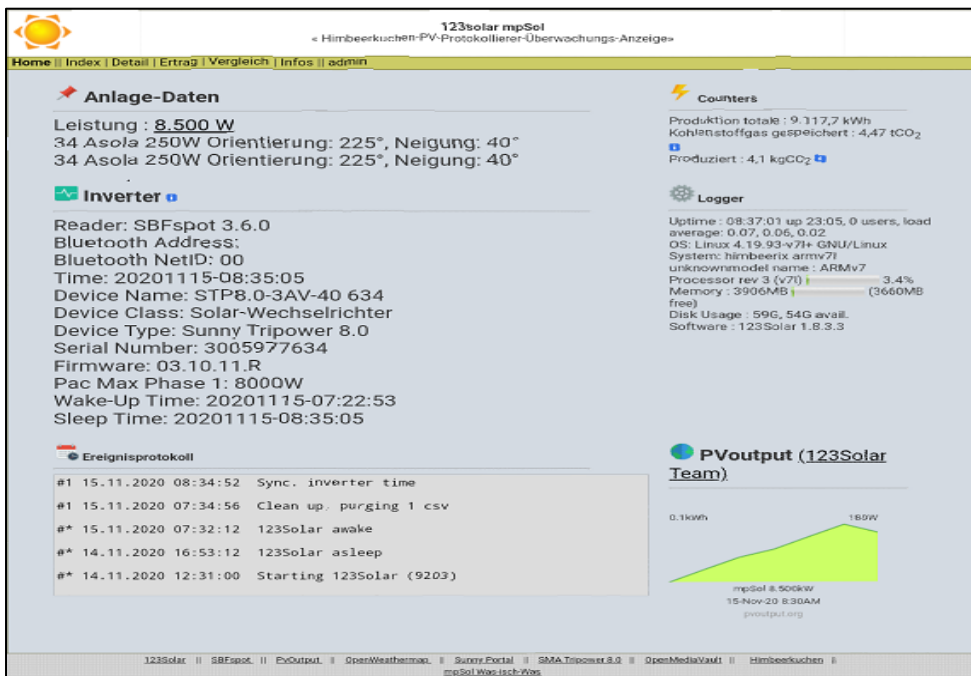
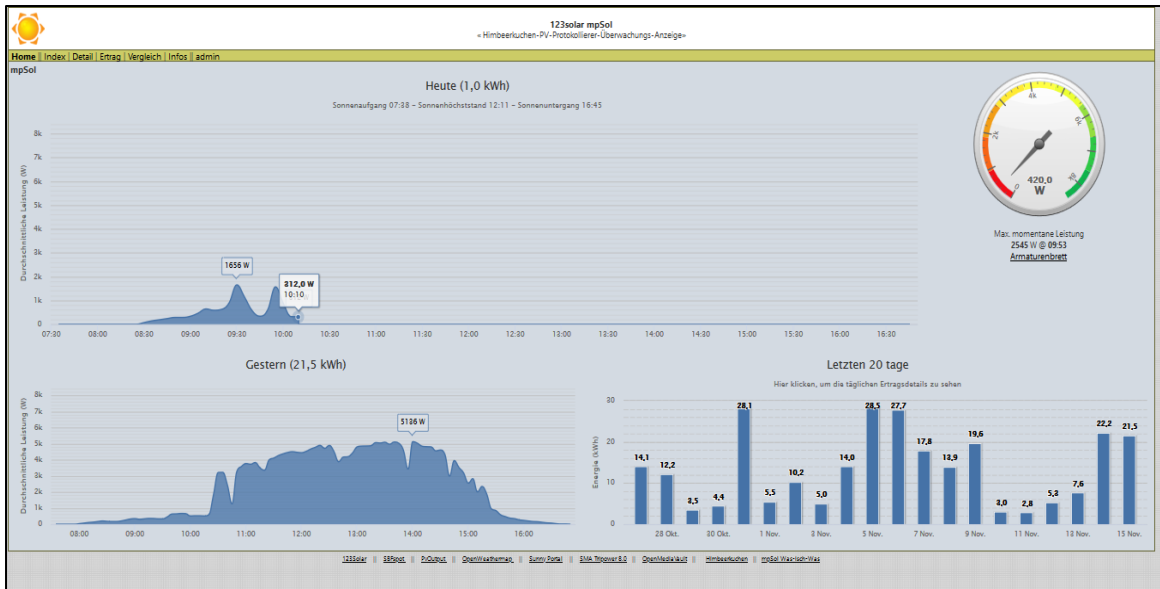
```

Anschließend sicherstellen, dass der Ordner „mpSol“ dem user www-data zugeordnet ist.

```
sudo chown -R www-data:www-data /var/www/123solar
```

Auf der Konfigurationsseite von 123Solar kann dann der Style entsprechend ausgewählt werden.

Der eingefügten Home-Button steht dann im Header zur Verfügung.



Rekonstruieren fehlender 123-CSV-Dateien

Sobald 123Solar aktiviert ist werden die aktuellen Daten (seit Aktivierung 123solar) aus der SBFspot-DB gelesen und im Ordner /etc/www/123solar\data\inv1\ als csv-Datei abgelegt und entsprechend visualisiert. Werte die vor dem Zeitpunkt der Aktivierung liegen werden nicht berücksichtigt.

Um diese „historischen“ Daten ebenfalls in 123solar zu sehen müssen die entsprechenden CSV-Dateien manuell bzw. mit Script erzeugt werden. Mit dem Script 123repair.sh das sich im Ordner /home/pi/scripts/old befindet, können die fehlenden CSV-Dateien aus SBFspot.db gelesen und erstellt werden. Diese sollten geprüft und anschließend in den Ordner /etc/www/123solar\data\inv1\ kopiert werden. Ggf. muss noch der Owner der Dateien angepasst werden. (sudo -R chown www-data:www-data 123solar).

Script 123repair.sh

```
#!/bin/bash
#
#####
# 123repair.sh
# script um fehlende 123-csv-dateien zu generieren
#
#
#####

JAHR=2021
MONAT=11
TAG=15

read -p "Jahr:" JAHR
read -p "Monat:" MONAT
read -p "Tag:" TAG

TODAYSQL=`date +%JAHR-%MONAT-%TAG`
CSVFILE=`date +%JAHR$MONAT$TAG`.csv
CSVPATH=/home/pi/tmp/
SMADB=/home/pi/smadata/SBFspot.db
CSVFILE=/home/pi/tmp/`date +%JAHR$MONAT$TAG`.csv
CMDFILE=/home/pi/tmp/mp_123repair
TMPFILE=/home/pi/tmp/tmp.csv

echo
"Time,I1V,I1A,I1P,I2V,I2A,I2P,I3V,I3A,I3P,I4V,I4A,I4P,G1V,G1A,G1P,G2V,G2A,G2P,G3V,G3A,G3P,FRQ,EFF,INVT,BOO
T,SR,KWHT">>$CSVFILE

echo .header off >$CMDFILE
echo .separator , >>$CMDFILE
echo .mode csv >>$CMDFILE

echo .output $TMPFILE >>$CMDFILE
echo "select time(TimeStamp ,
'unixepoch','localtime'),Udc1,Idc1,Pdc1,Udc2,Idc2,Pdc2,NULL,NULL,NULL,NULL,NULL,Uac1,
Iac1,Pac1,Uac2,Iac2,Pac2,Uac3,Iac3,Pac3,Frequency,NULL,NULL,NULL,Serial, ((ETotal*1.0)/1000) from SpotData
where date(TimeStamp, 'unixepoch')=\""$TODAYSQL"\";" >>$CMDFILE

echo .quit >>$CMDFILE

sudo sqlite3 $SMADB <$CMDFILE
sudo cat $TMPFILE >> $CSVFILE

ls -l $CSVPATH
```

Datensicherung und weitere Optimierungen

Jeden Abend soll eine Sicherung der SBFspot- und der 123solar-Daten per rsync und Public-Key-Authentifizierung auf die OMV-NAS erfolgen. Hierzu wird mit ssh-keygen die entsprechenden Key-Files erstellt, die anschließend im Verzeichnis /home/pi/.ssh zur Verfügung stehen.

SSH-Key erstellen

```
ssh-keygen
```

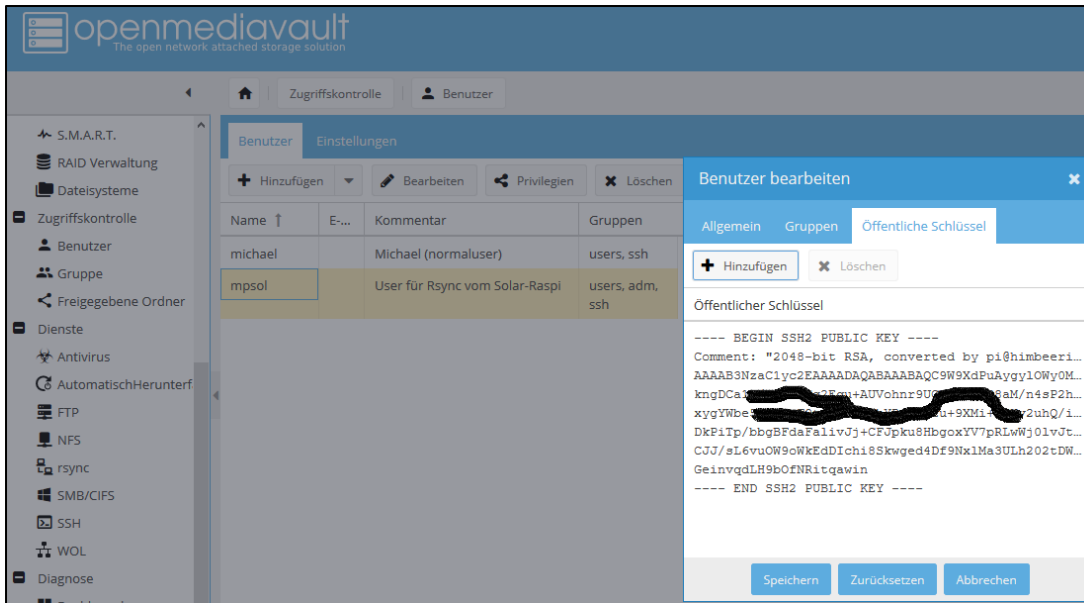
```
pi@himbeerix:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pi/.ssh/id_rsa): id_mpsol
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_mpsol.
Your public key has been saved in id_mpsol.pub.
The key fingerprint is:
SHA256:8hnTm9iPyqZF54R2Ter1RJfKltMnVJm0fSUY2DpPSIA pi@himbeerix
The key's randomart image is:
+---[RSA 2048]-----+
|  ....=OB+o.|
|  E .oB+ o+.|
|  ..+o= o o|
|  . *+o.o .|
|  . S *+   |
|  = @ o.   |
|  = =     |
|  o. o    |
|  .oo..   |
+----[SHA256]-----+
pi@himbeerix:~$
pi@himbeerix:~$ ls -l .ssh
insgesamt 12
-rw----- 1 pi pi 1823 Jan 17 10:44 id_mpsol
-rw-r--r-- 1 pi pi 394 Jan 17 10:44 id_mpsol.pub
-rw-r--r-- 1 pi pi 222 Jan 17 10:45 known_hosts
pi@himbeerix:~$
```

SSH-Key auf der Konsole ausgeben und markieren (in die Zwischenablage kopieren)

```
ssh-keygen -e -f ~/.ssh/id_mpsol.pub
```

```
pi@himbeerix:~$ ssh-keygen -e -f ~/.ssh/id_mpsol.pub
----- BEGIN SSH2 PUBLIC KEY -----
Comment: "2048-bit RSA, converted by pi@himbeerix from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQAC9W9XdPuAyglOWy0MCbkPTm/w5RtHusGFveYT3u
kngDCa1X/SUXVwGg2Equ+AUVohnr9UCbRCnkiB8aM/n4sP2hiACf/Bwvk0Ujam07/GPkSE
xygYWbe5rZrZy.....Vzzu+9XMi+CSHv2u.....7yOcPYP+wlq
DkPiTp/bbgBFdaFa.....pRLwWj0lvJtJFAQtopDAxOBY1ZqKAezCt
CJJ/sL6vuOW9oWkEdDIchi8Skwge.....DW+ZcyVOeSXzt55/XZgUyThr
GeinvqdLH9bOfNRitqawin
----- END SSH2 PUBLIC KEY -----
pi@himbeerix:~$
```

.... jetzt den SSH-Key markieren und per cut&paste beim user „mpsol“ auf der OMV-NAS eintragen und speichern.



Zugriff auf OMV via SSH-Key testen

```
ssh -i /home/pi/.ssh/id_mpsol 'mpsol@192.168.178.12'
```

```
pi@himbeerix:~$ ssh 'MPSOL@192.168.178.12'
```

```
Linux OMV 4.14.0-0.bpo.3-amd64 #1 SMP Debian 4.14.13-1~bpo9+1 (2018-01-14) x86_64
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

```
Last login: Sun Feb 9 10:20:14 2020 from 192.168.178.6
```

```
MPSOL@OMV:~$ exit
```

```
Abgemeldet
```

```
Connection to 192.168.178.12 closed.
```

```
pi@himbeerix:~$
```

```
pi@himbeerix:~$ ssh -i /home/pi/.ssh/id_mpsol 'mpsol@192.168.178.12'
```

```
Linux OMV 4.19.0-0.bpo.9-amd64 #1 SMP Debian 4.19.118-2+deb10u1~bpo9+1 (2020-06-09) x86_64
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

```
Last login: Mon Jan 18 11:52:19 2021 from 192.168.178.6
```

```
pi@himbeerix:~$
```

```
pi@himbeerix:~$
```

WakeOnLan einrichten

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install etherwake
```

WOL testen (OMV_MAC: 90:2B:34:8B:B1:E0, OMV_IP:192.168.178.12)

```
ping 192.168.178.12
sudo etherwake 90:2B:34:8B:B1:E0
...
... ca.1 Minute warten ...
...
ping 192.168.178.12
```

```
pi@himbeerix:~$ ping 192.168.178.12
PING 192.168.178.12 (192.168.178.12) 56(84) bytes of data.
From 192.168.178.6 icmp_seq=9 Destination Host Unreachable
From 192.168.178.6 icmp_seq=10 Destination Host Unreachable
From 192.168.178.6 icmp_seq=11 Destination Host Unreachable
^C
--- 192.168.178.12 ping statistics ---
14 packets transmitted, 0 received, +3 errors, 100% packet loss, time 550ms
pipe 4
pi@himbeerix:~$ sudo etherwake 90:2B:34:8B:B1:E0
pi@himbeerix:~$ ping 192.168.178.12
PING 192.168.178.12 (192.168.178.12) 56(84) bytes of data.
64 bytes from 192.168.178.12: icmp_seq=1 ttl=64 time=0.592 ms
64 bytes from 192.168.178.12: icmp_seq=2 ttl=64 time=0.136 ms
64 bytes from 192.168.178.12: icmp_seq=3 ttl=64 time=0.215 ms
64 bytes from 192.168.178.12: icmp_seq=4 ttl=64 time=0.232 ms
64 bytes from 192.168.178.12: icmp_seq=5 ttl=64 time=0.153 ms
64 bytes from 192.168.178.12: icmp_seq=6 ttl=64 time=0.142 ms
^C
--- 192.168.178.12 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 231ms
rtt min/avg/max/mdev = 0.136/0.245/0.592/0.159 ms
pi@himbeerix:~$ ^C
pi@himbeerix:~$
```

Sicherungsscript erstellen

```
cd /home/pi/scripts
nano bkup_mpsol.sh
```

rsync Optionen

-a, --archive	archive modus; gleich wie -rlptgoD (no -H,-A,-X)
-v, --verbose	zeigt während des Synchronisierens alle ausgeführten Schritte an
-z, --compress	aktiviert die Komprimierung für die Datenübertragung
-e, --rsh=COMMAND	definiert die zu benutzende remote shell
--numeric-ids	damit die UID des Besitzers nicht an den entfernten Rechner angepasst wird
-n, --dry-run	simuliert nur was passieren würde ("dry run")

Script bkup_mpsol.sh

```
#!/bin/bash
#
#####
# -- bkup_mpsol.sh ---
# script um die mpsol-Daten auf OMV-NAS zu sichern
#
#####
#OMV_LOG=/home/pi/logs/bkup_`date +%Y-%m-%d`.log
LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
RSCFILE=/home/pi/logs/mSolrsc`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte Script bkup_mpsol.sh" >> $LOGFILE

OMV_MAC=90:2B:34:8B:B1:E0
OMV_IP=192.168.178.12
OMV_ZIEL=mpsol@192.168.178.12

start=$(date +%s)
diff=0
max=300

echo "OMV aufwecken ...." >> $LOGFILE

sudo etherwake $OMV_MAC
sudo etherwake -b $OMV_MAC

while ! ping -c1 $OMV_IP &>/dev/null ;
do
if [ $diff -eq 0 ];
then
echo "OMV ist offline. Warte max. 5 Minuten..." >> $LOGFILE
fi
echo -ne " " .$diff >> $LOGFILE

diff=$((($date +%s) - start))
if [ $diff -ge $max ];
then
break ;
fi
done

ping $OMV_IP -c2
if [ $? -eq 0 ]
then
echo -e "\nOMV ist jetzt online: " >> $LOGFILE
date >> $LOGFILE
fi

### warte 10 Sec. bis OMV-Dienste sicher gestartet sind

echo " warte 10 Sec. bis OMV-Dienste sicher gestartet sind" >> $LOGFILE
sleep 10

if [ $diff -lt $max ];
then
#date >> $LOGFILE

## raspi /home sichern
echo "Starte Sicherung home:(/home)" >> $LOGFILE

rsync --log-file=$RSCFILE --numeric-ids -Oavze "ssh -i /home/pi/.ssh/id_mpsol" /home/ $OMV_ZIEL:/srv/dev-
disk-by-label-RAID1/MPSOL/_raspi/home/ >> $LOGFILE
```

```

## raspi var/www sichern
echo "Starte Sicherung www:(/var/www)" >> $LOGFILE

rsync --log-file=$RSCFILE --numeric-ids -Oavze "ssh -i /home/pi/.ssh/id_mpsol" /var/www/ $OMV_ZIEL:/srv/dev-
disk-by-label-RAID1/MP SOL/_raspi/var/www/ >> $LOGFILE

## /var/log/sbfsot.3 sichern
echo "Starte Sicherung sbfspot-logs:(/var/log/sbfspot.3)" >> $LOGFILE

rsync --log-file=$RSCFILE --numeric-ids -Oavze "ssh -i /home/pi/.ssh/id_mpsol" /var/log/sbfspot.3/
$OMV_ZIEL:/srv/dev-disk-by-label-RAID1/MP SOL/_raspi/var/log/sbfspot.3/ >> $LOGFILE

else
echo "Zeitüberschreitung, OMV konnte nicht hochgefahren werden. Fehler." >> $LOGFILE
exit 1
fi

## rsync-logfile ans Log-file anhaengen
cat $RSCFILE >> $LOGFILE

echo "Backup beendet" >> $LOGFILE

date >> $LOGFILE
echo "Ende Script bkup_mpsol.sh " >> $LOGFILE
echo -e "\n-----\n" >> $LOGFILE

```

Script ausführbar Machen und testen:

```
sudo chmod 755 bkup_mpsol.sh
```

Scriptaufruf in crontab eintragen

```
crontab -e
```

```

*/5 6-22 * * * /usr/local/bin/sbfspot.3/daydata
55 05 * * * /usr/local/bin/sbfspot.3/monthdata

30 22 * * * /home/pi/scripts/bkup_mpsol.sh

```

Crontab anzeigen

```

pi@himbeerix:~$
pi@himbeerix:~$ crontab -l
*/5 6-22 * * * /usr/local/bin/sbfspot.3/daydata
55 05 * * * /usr/local/bin/sbfspot.3/monthdata

30 22 * * * /home/pi/scripts/bkup_mpsol.sh

0 1 * * * find /var/log/sbfspot.3/ -name "*.log" -mtime +7 -delete
0 1 * * * find /home/pi/smadata/logs/ -name "*.log" -mtime +7 -delete
0 1 * * * find /home/pi/smadata/ -name "*.csv" -mtime +7 -delete
0 1 * * * find /home/pi/solarlog/ -name "*.*)" -mtime +7 -delete
0 1 * * * find /home/pi/logs/ -name "*.log" -mtime +7 -delete
pi@himbeerix:~$

```

Ertragsdaten zusätzlich im Solarlog-Format bereitstellen

Beim Solarlog-Format werden die Ertrags-Daten im Wesentlichen in vier verschiedenen Dateien für Tages- Monats- Jahres-Ertrag und bereitgestellt. Es müssen folgende Dateien erstellt werden:

minYYMMDD.js - Tagesertrag im 5-Minuten Intervall
days_hist.js - Tages-Historie
months.js - Monatsertrag
years.js - Jahresertrag

Tagesertrag im 5-Minuten Intervall (min120508.js)

Es wird pro Tage eine Datei benötigt. Diese Dateien enthalten die fortlaufend Messwerte des jeweiligen Tages im 5-Minuten-Intervall (in absteigender Reihenfolge), d.h. alle 5 Minuten wird eine neue Zeile mit folgendem Zeilenaufbau hinzugefügt:

Kennung		Datum und Uhrzeit		PAC [W]		PDC [W]		PDC2 [W]		Tages- Summe [Wh]		UDC1 [V]		UDC2 [V]	
m[mi++]=	"	08.05.2012 15:35		873	;	488	;	488	;	17746	;	350	;	350	"
m[mi++]=	"	08.05.2012 15:30		912	;	498	;	492	;	17744	;	352	;	351	"

Gesamt-Tagesertrag je Tag (days_hist.js)

Die Datei days_hist.js enthält die Tageswerte aller vergangenen Tage (in absteigender Reihenfolge), d.h. am Tagesende bzw. nach Sonnenuntergang wird jeweils eine neue Zeile mit folgendem Zeilenaufbau hinzugefügt:

Kennung		Datum		Tagesertrag [Wh]		Max PAC des Tages [W]	
da[da++]=	"	19.06.2020		1689	;	1236	"
da[da++]=	"	18.06.2020		46279	;	7428	"
da[da++]=	"	17.06.2020		21077	;	6912	"

Monatsertrag (months.js)

Die Datei enthält den jeweiligen Monatsertrag bis zum aktuellen Tag (in absteigender Reihenfolge). Die Datei wird täglich mit folgendem Zeilenaufbau aktualisiert:

Kennung		Datum		Monatsertrag [Wh]	
mo[mx++]=	"	26.09.20		758239	"
mo[mx++]=	"	31.08.20		1007000	"
mo[mx++]=	"	31.07.20		1274000	"

Jahresertrag (years.js)

Die Datei enthält den Jahresertrag (in absteigender Reihenfolge). Die Datei wird täglich mit folgendem Zeilenaufbau aktualisiert:

Kennung		Datum		Jahresertrag [Wh]	
ye[yx++]=	"	31.12.2019		8557620	"
ye[yx++]=	"	31.12.2018		8672925	"
ye[yx++]=	"	31.12.2017		8399968	"

DB View für Solarlog-Daten erstellen

Um die aktuellen Daten in das Solarlog-Format zu konvertieren wird ein zusätzlicher View (**vwSolarlogData**) benötigt mit dem die Werte entsprechend konvertiert werden können:

```
cd /home/pi/smadata
sqlite3 SBFspot.db
```

```
DROP VIEW IF EXISTS vwSolarlogData;

CREATE VIEW vwSolarlogData AS
SELECT dd.Timestamp,
CASE WHEN dd.Power > 8500
  THEN 0
ELSE
  dd.Power
END AS PAC,
CASE WHEN spot.Pdc1 is Null
  THEN 0
ELSE
  spot.pdc1
END AS PDC1,
CASE WHEN spot.Pdc2 is Null
  THEN 0
ELSE
  spot.pdc2
END AS PDC2,
dd.TotalYield AS DaySum,
CASE WHEN spot.udc1 is Null
  THEN 0
ELSE
  spot.udc1
END AS UDC1,
CASE WHEN spot.udc2 is Null
  THEN "0"
ELSE
  spot.udc2
END AS UDC2
FROM vwDayData AS dd
LEFT JOIN vwAvgSpotData AS spot
ON dd.Serial = spot.Serial AND dd.Timestamp = spot.Nearest5min
LEFT JOIN vwAvgConsumption AS cons
ON dd.Timestamp = cons.Nearest5min
ORDER BY dd.Timestamp DESC;
```

Scripts zur Erstellung der Solarlog-Daten

Die Erstellung der Solarlog-Dateien erfolgt durch die scripts mSolToday.sh, mSolDayhist.sh, mSolMonths.sh und mSolYears.sh die per cron gestartet werden. Der Upload der Daten aktuellen Daten auf die Webserver erfolgt dann anschließend durch das Script mSolUpload.sh.

Die Solarlogdaten werden im Verzeichnis =/home/pi/solarlog/mpsol abgelegt, per ftp auf den/die Webserver kopiert und zusätzlich per Backup-Skript zusammen auf die OMV gesichert.

```
#!/bin/bash
#
#####
# mSolToday.sh
# script generiert aus SBFspot.db die solarlog-dateien vom aktuellen Tag
#####
LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte Script mSolToday.sh " >> $LOGFILE

#####w#####
#--- als erstes die ggf. zu grossen werte in der DB-tabelle daydata loeschen
#####w#####
SMADB=/home/pi/smadata/SBFspot.db
CLRFILE=/home/pi/sol_tmp/cleanup.cmd
CLRLOG=/home/pi/logs/cleanup.log

echo "In der DB $SMADB wird die Tabelle Daydata geprüft und ggf. bereinigt: " >> $LOGFILE

echo .header on >$CLRFILE
echo .mode list >>$CLRFILE
echo .separator ";" >>$CLRFILE
echo .output $CLRLOG >>$CLRFILE

echo "select date(TimeStamp, 'unixepoch'),* from daydata where power>1205987;" >> $CLRFILE
echo "update daydata set power=0 where power>1205987;" >> $CLRFILE
echo .quit >> $CLRFILE
#sudo sqlite3 $SMADB <$CLRFILE
sqlite3 $SMADB <$CLRFILE

cat $CLRLOG >> $LOGFILE

date >> $LOGFILE
echo "Ende Prüfung/Bereinigung Tabelle Daydata " >> $LOGFILE
#####
#--- jetzt die solarlogdaten erstellen
#####
SMADB=/home/pi/smadata/SBFspot.db
CMDFILE=/home/pi/sol_tmp/mSolToday.cmd
TODAYSQL=`date +%Y-%m-%d`
MINFILE="/home/pi/solarlog/min`date +%y%m%d`.js"

date >> $LOGFILE
echo "Solarlog 5-Minuten-Daten werden erstellt als: $MINFILE" >> $LOGFILE

DAYS_MIN=$(sqlite3 $SMADB "select min(daysum) from vwSolarlogdata where
substr(TimeStamp,1,10)='$TODAYSQL'")

echo .header off > $CMDFILE
echo .mode list >> $CMDFILE
echo .separator ";" >> $CMDFILE
echo .output $MINFILE >> $CMDFILE

echo " select 'm[mi++]='\'' || strftime('%d.%m.',timestamp) || substr(timestamp,3,2) || strftime('
%H:%M:%S',timestamp) || '|' || PAC || ';' || cast(pdc1 as integer) || '|' || cast(pdc2 as integer) || ';' || (daysum-
($DAYS_MIN)) || ';' || cast(udc1 as integer) || ';' || cast(udc2 as integer) || '\'' from vwSolarlogdata where
substr(TimeStamp,1,10)='\''$TODAYSQL'\'';" >>$CMDFILE

echo .quit >> $CMDFILE
#sudo sqlite3 $SMADB < $CMDFILE
sqlite3 $SMADB < $CMDFILE
```

```
echo "Solarlog 5-Minuten-Daten wurden erstellt" >> $LOGFILE  
date >> $LOGFILE  
echo "Ende Script mSolToday.sh " >> $LOGFILE  
echo -e "\n-----\n" >> $LOGFILE
```

```
#!/bin/bash
#
#####
# mSolDayhist.sh
# Solarlog Tagesdatei days.js und Tageshistorie day_hist.js generieren
#
# zielformat: da[dx++]="25.03.20|5980;1204"
#####
LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte script mSolDayhist.sh " >> $LOGFILE

SMADB=/home/pi/smadata/SBFspot.db
CMDFILE=/home/pi/sol_tmp/dayshist.cmd

DAY_FILE="/home/pi/solarlog/days.js"
HST_FILE="/home/pi/solarlog/days_hist.js"
TMP_FILE="/home/pi/sol_tmp/days_hist_tmp.txt"

echo "Solarlog Tagesdatei: $DAY_FILE und Tageshistorie: $HST_FILE wird erstellt" >> $LOGFILE

TODAYSQL=`date +%Y-%m-%d`

echo .header off > $CMDFILE
echo .mode list >> $CMDFILE
echo .separator ";" >> $CMDFILE
echo .output $DAY_FILE >> $CMDFILE

echo "select
'da[dx++]=\"\"| | strftime('%d.%m.',date(timestamp,'unixepoch'))| | substr(strftime('%Y',datetime(TimeStamp,'unixepoch')),3,2) | |' | | cast(max(Totalyield) - min(Totalyield) as text), max(power) | |\"' from daydata where
date(TimeStamp,'unixepoch')=\\\"$TODAYSQL\\\"";" >> $CMDFILE

echo .quit >> $CMDFILE
sqlite3 $SMADB < $CMDFILE

FIRST_LINE=$(head -n1 $HST_FILE)
FILE_DATUM=${FIRST_LINE:10:8}

DATUM=`date +%d.%m.%y`

#echo "erste zeile: $FIRST_LINE " >> $LOGFILE
#echo "datum der ersten zeile: $FILE_DATUM " >> $LOGFILE
#echo "datum von heute : $DATUM " >> $LOGFILE

if [ "$DATUM" = "$FILE_DATUM" ];
then
#echo "eintrag von Heute bereits vorhanden, erst ab zeile zwei uebernehmen" >> $LOGFILE
#echo " $HST_FILE als $TMP_FILE ab zeile zwei uebernehmen" >> $LOGFILE
tail +2 $HST_FILE > $TMP_FILE
else
#echo "$HST_FILE kopieren nach $TMP_FILE " >> $LOGFILE
cp $HST_FILE $TMP_FILE
fi

cat $DAY_FILE $TMP_FILE > $HST_FILE

echo "Solarlog Tages-Datei und Tageshistorie wurde erstellt" >> $LOGFILE

date >> $LOGFILE
echo "Ende script mSolDayhist.sh " >> $LOGFILE
echo -e "\n-----\n" >> $LOGFILE
```

```
#!/bin/bash
#
#####
# mSolMonths.sh
# solarlog-monatsertrag-datei months.js aktualisieren
#
# zielformat: mo[mx++]="24.10.20|313053" monatsertrag bisd zum aktuellen tag
#             mo[mx++]="30.09.20|839526" monatsertrag bis zum monatsende
#
# startdatum 01.aktueller monat.jahr
# min_daysum ermitteln vom startdatum
# max_daysum ermitteln alles > startdatum
# monatertrag = max -min
# zeile schreiben
# wenn aktueller tag == 01 dann zeile als erste zeile zu month.js zufügen
#             sonst zeile als austauschen .. cat +2 eile + tail +2 month.js
#
#####
LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte script mSolMonths.sh " >> $LOGFILE

TAG=$(date +%d)
MONAT=$(date +%m)
JAHR=$(date +%y)

SMADB=/home/pi/smadata/SBFspot.db
CMDFILE=/home/pi/sol_tmp/mSolMonths.cmd

MON_FILE="/home/pi/solarlog/months.js"
TMP_FILE="/home/pi/sol_tmp/months_tmp.txt"
NEW_FILE="/home/pi/sol_tmp/months_new.txt"
STARTSQL=`date +%Y-%m-01`

echo "Solarlog Monatserträge werden aktualisiert als $MON_FILE " >> $LOGFILE

min_daysum=$(sqlite3 $SMADB "select min(totalyield) from daydata where
date(TimeStamp,'unixepoch')>=$STARTSQL")
max_daysum=$(sqlite3 $SMADB "select max(totalyield) from daydata where
date(TimeStamp,'unixepoch')>=$STARTSQL")

echo 'mo[mx++]="'$TAG.$MONAT.$JAHR'|'${(($max_daysum-$min_daysum))}'" > $NEW_FILE

if [ $TAG -eq "01" ]
then
#echo "es ist der erster tag im monat, alle zeilen uebernehmen"
cat $MON_FILE > $TMP_FILE
else
#echo " es ist nicht der erste tag im monat, ab zeile zwei uebernehmen"
tail +2 $MON_FILE > $TMP_FILE
fi

cat $NEW_FILE $TMP_FILE > $MON_FILE

date >> $LOGFILE
echo "Ende script mSolMonths.sh " >> $LOGFILE
echo -e "\n-----\n" >> $LOGFILE
```

```
#!/bin/bash
#
#####
# mSolYears.sh
# solarlog-jahresertrag-datei years.js aktualisieren
#
# zielformat: ye[yx++]="24.10.20|8450000" jahresertrag bis zum aktuellen tag
#           ye[yx++]="31.12.19|8557620" jahresertrag bis zum jahresende
#
#####
LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte script mSolYears.sh " >> $LOGFILE

TAG=$(date +%d)
MONAT=$(date +%m)
JAHR=$(date +%y)

SMADB=/home/pi/smadata/SBFspot.db
CMDFILE=/home/pi/sol_tmp/mSolYears.cmd

YRS_FILE="/home/pi/solarlog/years.js"
TMP_FILE="/home/pi/sol_tmp/years_tmp.txt"
NEW_FILE="/home/pi/sol_tmp/years_new.txt"
STARTSQL=`date +%Y-01-01`

echo "Solarlog Jahreserträge werden aktualisiert als $YRS_FILE " >> $LOGFILE

min_sum=$(sqlite3 $SMADB "select min(totalyield) from monthdata where
date(TimeStamp,'unixepoch')>='`$STARTSQL`'")
max_sum=$(sqlite3 $SMADB "select max(totalyield) from monthdata where
date(TimeStamp,'unixepoch')>='`$STARTSQL`'")

echo 'ye[yx++]="'$TAG.$MONAT.$JAHR'|'`${($max_sum-$min_sum)}`'" > $NEW_FILE

if [ $TAG -eq "01" ]
then
if [ $MONAT -eq "01" ]
then
#echo "es ist der erster tag im ersten monat, alle zeilen uebernehmen" >> $LOGFILE
cat $YRS_FILE > $TMP_FILE
fi
else
#echo " es ist nicht der erste tag im monat, ab zeile zwei uebernehmen" >> $LOGFILE
tail +2 $YRS_FILE > $TMP_FILE
fi

cat $NEW_FILE $TMP_FILE > $YRS_FILE

date >> $LOGFILE
echo "Ende script mSolYears.sh " >> $LOGFILE
echo -e "\n-----\n" >> $LOGFILE
```

```
#!/bin/bash
#
#####
# mSolUpload.sh
# script kopiert die aktuellen Daten im Solarlog-Format
# auf die entsprechenden Webserver/webspace
#
#####
#
# upload der solarlog-Daten zur webseite home.solarlog-web.de
#
# ftp-server=ftp://home.solarlog-web.de
#
# zielverzeichnis=default-website/date
# user=f***** *(user und PW siehe SBFspot.cfg und/oder keePass)
# pw=r*****
#####

LOGFILE=/home/pi/logs/mSol`date +%Y-%m-%d`.log
date >> $LOGFILE
echo "Starte script mSolUpload.sh " >> $LOGFILE

HOST='ftp://home.solarlog-web.de'
USER='f*****' *(user und PW siehe SBFspot.cfg und/oder keePass)
PASS='r*****'

TARGETFOLDER='/'
SOURCEFOLDER='/home/pi/solarlog'
echo "Starte Upload der Solarlog-Dateien zu $HOST " >> $LOGFILE

lftp -f "
open $HOST
user $USER $PASS
lcd $SOURCEFOLDER
mirror --reverse --verbose $SOURCEFOLDER $TARGETFOLDER >> $LOGFILE
bye
" >> $LOGFILE
echo "Upload der Solarlog-Dateien beendet" >> $LOGFILE

#####
# upload der solarlog-Daten zur webseite https://himbeerix.12hp.de/
#
# ftp-server=ftp://himbeerix.lima-ftp.de
#
# zielverzeichnis=default-website/daten
# user=h***** *(user und PW siehe SBFspot.cfg und/oder keePass)
# pw=e*****
#####

HOST='ftp://himbeerix.lima-ftp.de'
USER='h*****' *(user und PW siehe SBFspot.cfg und/oder keePass)
PASS='e*****'
TARGETFOLDER='/default-website/daten'
SOURCEFOLDER='/home/pi/solarlog'
echo "Starte der Solarlog-Dateien Upload zu $HOST " >> $LOGFILE

lftp -f "
open $HOST
user $USER $PASS
lcd $SOURCEFOLDER
mirror --reverse --verbose $SOURCEFOLDER $TARGETFOLDER >> $LOGFILE
bye
```

```
" >> $LOGFILE
```

```
echo "Upload der Solarlog-Dateien beendet" >> $LOGFILE
```

```
date >> $LOGFILE
```

```
echo "Ende script mSolUpload.sh " >> $LOGFILE
```

```
echo -e "\n-----\n" >> $LOGFILE
```

Skripte in crontab eintragen

```
crontab -e
```

```
05 22 * * * /home/pi/scripts/mSolToday.sh
10 22 * * * /home/pi/scripts/mSolDayhist.sh
12 22 * * * /home/pi/scripts/mSolMonths.sh
14 22 * * * /home/pi/scripts/mSolYears.sh
20 22 * * * /home/pi/scripts/mSolUpload.sh
```

crontab anzeigen

```
pi@himbeerix:~$ crontab -l
```

```
*/5 6-23 * * * /usr/local/bin/sbfspot.3/daydata
```

```
55 05 * * * /usr/local/bin/sbfspot.3/monthdata
```

```
05 22 * * * /home/pi/scripts/mSolToday.sh
```

```
10 22 * * * /home/pi/scripts/mSolDayhist.sh
```

```
12 22 * * * /home/pi/scripts/mSolMonths.sh
```

```
14 22 * * * /home/pi/scripts/mSolYears.sh
```

```
20 22 * * * /home/pi/scripts/mSolUpload.sh
```

```
25 22 * * * /home/pi/scripts/m123Upload.sh
```

```
30 22 * * * /home/pi/scripts/bkup_mpsol.sh
```

```
0 1 * * * find /var/log/sbfspot.3/ -name "*.log" -mtime +7 -delete
```

```
0 1 * * * find /home/pi/smadata/logs/ -name "*.log" -mtime +7 -delete
```

```
0 1 * * * find /home/pi/smadata/ -name "*.csv" -mtime +7 -delete
```

```
0 1 * * * find /home/pi/solarlog/ -name ".*" -mtime +7 -delete
```

```
0 1 * * * find /home/pi/logs/ -name "*.log" -mtime +7 -delete
```

```
pi@himbeerix:~$
```

