

Inbetriebnahme eines Steuerrechners  
für eine Photovoltaik-Insel  
mit einem lokalen Datenspeicher

## Inhalt

Komponenten.....	4
Betriebssystem aufspielen .....	4
Rpiboot.exe .....	4
Raspi vorbereiten: .....	5
Rpiboot.exe ausführen .....	5
Pi Imager.....	6
Raspi booten.....	7
SSH-Verbindung.....	7
Stromversorgung.....	8
Netzteil für Hutschiene.....	8
Stromversorgung des Raspi-IO-Board .....	8
Stromversorgung der SATA-Festplatte.....	9
Einrichten der Festplatte.....	10
Prüfen, ob die Festplatte verfügbar ist mit .....	10
Partitionierung und Formatierung der Festplatte.....	11
Variante 1 .....	11
Variante 2 .....	11
Dateisystem EXT4 erstellen .....	12
Festplatte mounten.....	12
Verzeichnis erstellen und Rechte vergeben .....	12
Variante 1 – manuelles Mounten.....	12
Variante 2 – automatisches Mounten bei jedem Start .....	12
Test-Datei erstellen .....	13
Mount rückgängig machen.....	13
Samba und Freigaben -Zugriff auf die Festplatte von Windows aus .....	14
SMB- Manual anzeigen.....	14
Status der Samba-Dienste abfragen.....	14
Samba-Komponenten installieren.....	14
Freigabe in die Konfig-Datei eintragen.....	14
Samba-Dienste neu starten.....	14
User für Login beim Aufruf der Freigaben unter Windows.....	14
MariaDB.....	15
Installation Version 10.5.....	15
Sicherstellen, dass der Service nach dem Systemstart ausgeführt wird .....	15
Konfigurieren.....	15

Wo liegen die Konfig-Dateien?.....	16
Data-Dir verschieben.....	16
Dienst stoppen .....	16
Verzeichnis auf der Festplatte anlegen und Rechte vergeben.....	16
Konfiguration korrigieren .....	16
Bestehendes Verzeichnis verschieben .....	16
Dienst neu starten .....	16
Remotezugriff ermöglichen.....	16
Mariadb-Service neu starten.....	17
Weiteren User anlegen .....	17
Anmelden .....	17
Datenbank erstellen .....	17
Rechte für Datenbank vergeben .....	17
Test des Remote-Zugriffs von HeidiSql auf MariaDb von Windows aus .....	18
Log .....	18
Abmelden .....	18
Python .....	19
Installation.....	19
Python-Connector für die mariadb .....	19
Script timergesteuert ausführen .....	19
GPIO-Aktoren und -Sensoren .....	20
Aktoren: 3-Kanal-Relais-Board .....	20
Sensoren.....	20
<a href="https://github.com/grasmax/AcOnOff/blob/main/gh_gpointest.py">https://github.com/grasmax/AcOnOff/blob/main/gh_gpointest.py</a> .....	20
Weitere nützliche Befehle .....	21
Ausschalten .....	21
Neustart.....	21

## Komponenten

- Raspberry Pi CM4IO Board
- CM4001032 Raspberry Pi Compute Module 4, 1GB-RAM, 32GB-eMMC, BCM2711, ARM Cortex-A72
- (raspberry pi os lite (32bit) v11, 0,4 GB – getestet aber nicht produktiv)
- raspberry pi os (32bit) v11, 0.9 GB
- IO CREST JMB582 2 Port SATA III PCI-e 3.0 x1 Non-RAID Controller Karte Jmicro Chipsatz SI-PEX40148 (<https://github.com/geerlingguy/raspberry-pi-pcie-devices/issues/64>)
- 3x Relais Board - Raspberry Pi GPIO Erweiterung
- 2TB WD20EFZX
- Meanwell Duo Netzteil 66W 12V/5V

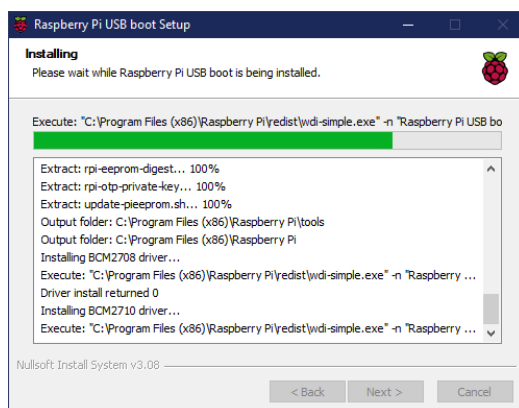
## Betriebssystem aufspielen

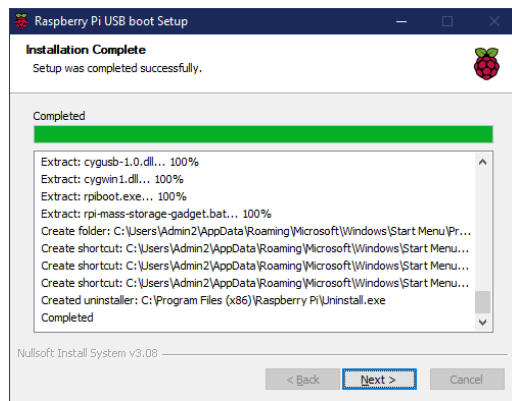
Rpiboot.exe

<https://core-electronics.com.au/guides/how-to-flash-write-raspbian-os-onto-raspberry-pi-compute-module-4-cm4/>

**usbboot/win32/rpiboot\_setup.exe** runtergeladen und installiert nach

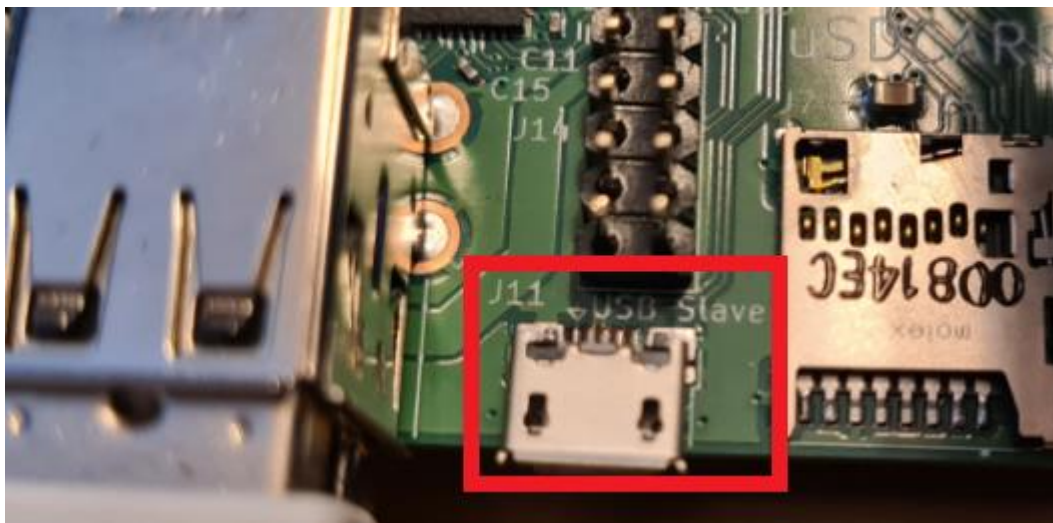
C:\Program Files (x86)\Raspberry Pi\rpiboot.exe





### Raspi vorbereiten:

USB-Slave-Port des IO-Boards mit einem Micro-Usb/USB-Kabel mit dem Windows-PC verbinden. Achtung: nicht jeder USB-Port am Windows-PC ist geeignet!



Brücke/Jumper1 setzen:



Strom einschalten

### Rpiboot.exe ausführen

Wenn Endlosschleife „Loading embedded: bootcode4.bin“, dann anderen USB-Port benutzen.

<https://github.com/raspberrypi/usbboot/issues/72>

```

Eingabeaufforderung
Loading embedded: bootcode4.bin
Loading embedded: bootcode4.bin
Loading embedded: bootcode4.bin
Loading embedded: bootcode4.bin

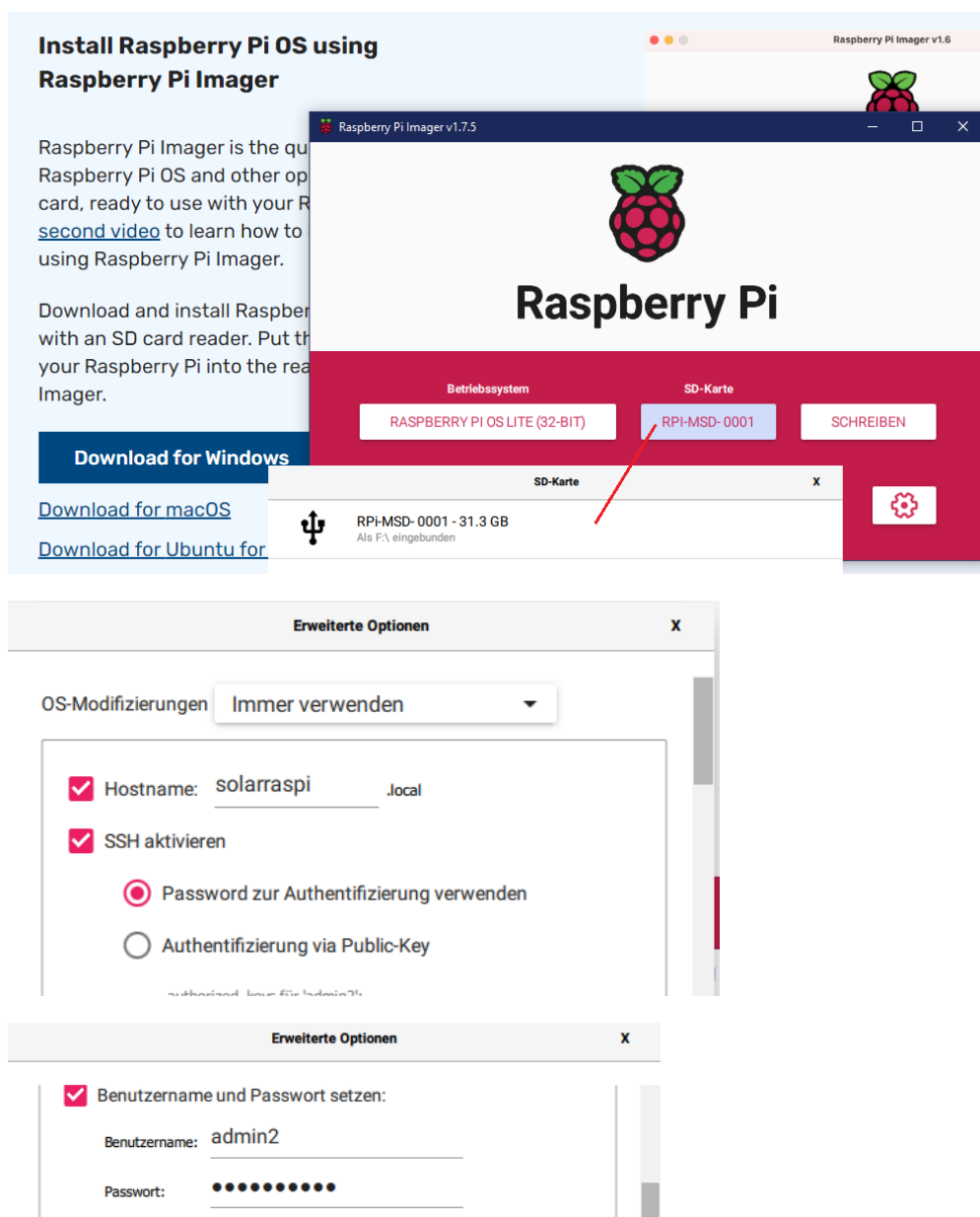
C:\Program Files (x86)\Raspberry Pi>rpiboot.exe
RPiBOOT: build-date Dec 16 2022 version 20221215~105525 1afa26c5
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Sending bootcode.bin
Successful read 4 bytes
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Failed to claim interface
Loading embedded: bootcode4.bin
Second stage boot server
Cannot open file config.txt
Cannot open file pieeprom.sig
Loading embedded: start4.elf
File read: start4.elf
Cannot open file fixup4.dat
Second stage boot server done

C:\Program Files (x86)\Raspberry Pi>

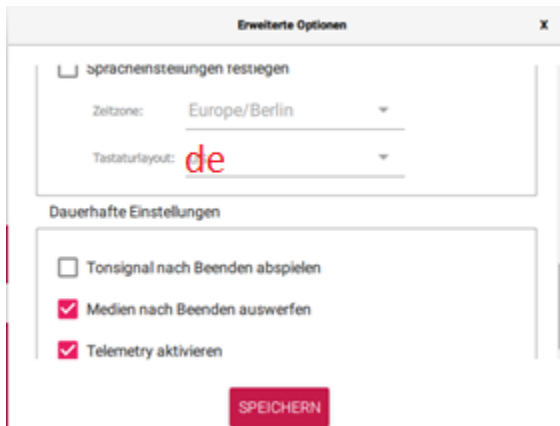
```

## Pi Imager

Pi Imager installieren und ausführen: eMMC des CM\$ erscheint unter „SD-Karte“ zur Auswahl:



Tastatur auf „de“ einstellen! Das erspart y/z-Ärger bei z.B. Passwörtern!



## Raspi booten

Strom ausschalten

USB-Kabel abziehen

Jumper-Stecker auf dem IO-Board entfernen

LAN-Kabel anstecken

Feste IP-Adresse im Router konfigurieren

Maus, Tastatur und Bildschirm anstecken

PCIe-\*2Sata-Board einstecken und mit Festplatte verbinden

Strom einschalten

Wenn Tastatur und Maus nicht funktionieren:

Usb-Kabel steckt noch am Slave-Port des IO-Boards.

Texas\_Dave – vor 2 Jahren

Nur ein Kommentar, um zu verhindern, dass jemand anderes in die gleiche Grube fällt wie ich. Ich habe diese Zeile zu config.txt hinzugefügt, indem ich sie über J11 verbunden und als USB-Speichergerät booten ließ. Sobald ich es in Windows sehen konnte, habe ich config.txt bearbeitet. Aber egal was, USB-Maus und -Tastatur würden nie funktionieren. Schließlich wurde mir klar, dass J11 immer noch über das Micro-USB-Kabel mit Strom versorgt wurde. Ich erinnere mich an den Schaltplan, der dadurch einen Schalter umlegt, der die USB-Signale vom HUB zu J11 umleitet. Von J11 getrennt und alles in Ordnung! Dabei gingen nur 20 Minuten verloren. :- ) Nicht ungewöhnlich für unbekannte Hardware. Lektion gelernt. Ich hoffe, das hilft jemand anderem, nicht 20 Minuten oder mehr zu verschwenden.

## SSH-Verbindung

mit z.B. putty testen

SSH-Einstellungen geändert (in der Hoffnung, dass dann der Login auch mit root funktioniert)

```
sudo nano /etc/ssh/sshd_config
```

u.a. PAMAuth aus

login mit root war trotzdem nicht möglich, zum Glück funktionierte der andere Login noch...

## Stromversorgung

### Netzteil für Hutschiene

Die Stromversorgung für das Raspi-IO-Board und die Festplatte kommt von einem

Meanwell Duo Netzteil 66W 12V/5V

Com/V1: 5VDC, Com/V2: 12 VDC



### Stromversorgung des Raspi-IO-Board

Die Stromversorgung erfolgt über einen 5,5x2,1mm Hohlstecker gewinkelt



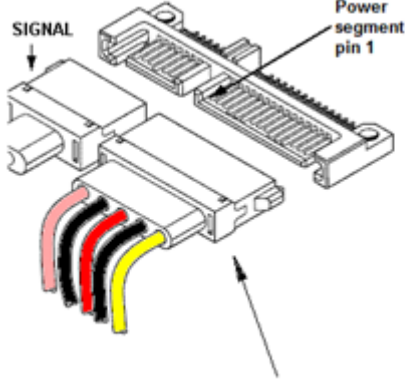


## Stromversorgung der SATA-Festplatte

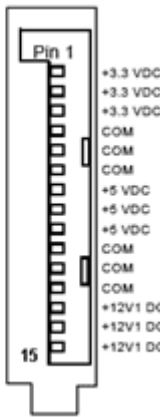
Wie kommen die 5V/12V an die Festplatte?

12V/0,45A über schwarz und gelb

5V/0,6A über schwarz und rot




**SERIAL ATA POWER CONNECTOR**




**Support 5/2A and 12V/2A**  
**Two specifications for power supply**

Red core wire is 5V power supply, yellow core wire is 12V power supply



Pin	Signal	PSU Color Wire
1	+3.3VDC	orange
2	+3.3VDC	orange
3	+3.3VDC	orange
4	GND	black
5	GND	black
6	GND	black
7	+5VDC	red
8	+5VDC	red
9	+5VDC	red
10	GND	black
11	Optional	black
12	GND	black
13	+12VDC	yellow
14	+12VDC	yellow
15	+12VDC	yellow

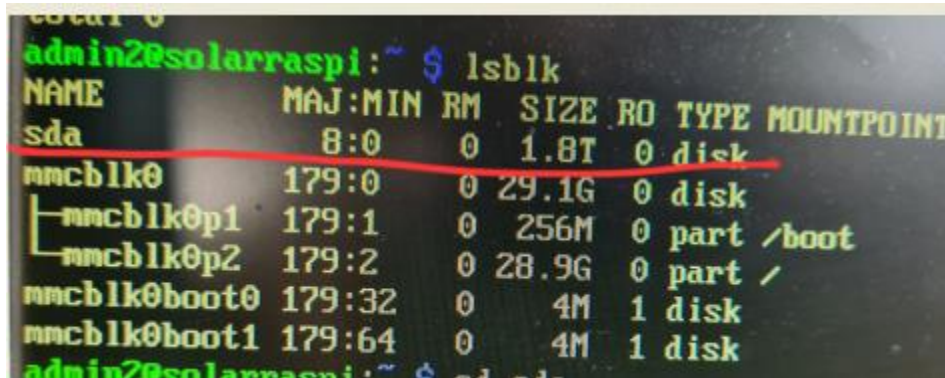


## Einrichten der Festplatte

Prüfen, ob die Festplatte verfügbar ist mit

<https://www.elektronik-kompodium.de/sites/raspberry-pi/2102191.htm>

lsblk



```
admin2@solarrraspi:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                8:0    0  1.8T  0 disk
mmcblk0                            179:0    0  29.1G  0 disk
├─mmcblk0p1                        179:1    0   256M  0 part /boot
└─mmcblk0p2                        179:2    0  28.9G  0 part /
mmcblk0boot0                      179:32    0    4M   1 disk
mmcblk0boot1                      179:64    0    4M   1 disk
```

## Partitionierung und Formatierung der Festplatte

Sudo apt-get update

Variante 1

Sudo apt install gparted - I läuft nur in OS-Version mit Desktop!

Variante 2

GPT- Partition erstellen mit fdisk

sudo fdisk /dev/sda

m (help)

**g (create GPT partition table)...created**

w

```
admin2@solarrraspi: ~ $ sudo fdisk /dev/sda
Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write the
changes to disk. Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xad8b89fa.

Command (m for help): m

Help:
DOS (MBR)
a toggle a bootable flag
b edit nested BSD disklabel
c toggle the dos compatibility flag

Generic
d delete a partition
F list free unpartitioned space
l list known partition types
n add a new partition
p print the partition table
t change a partition type
u verify the partition table
i print information about a partition

Misc
m print this menu
u change display/entry units
x extra functionality (experts only)

Script
I load disk layout from sfdisk script file
O dump disk layout to sfdisk script file

Save & Exit
w write table to disk and exit
q quit without saving changes

Create a new label
g create a new empty GPT partition table
G create a new empty SGI (IRIX) partition table
o create a new empty DOS partition table
s create a new empty Sun partition table

Command (m for help): g
Created a new GPT disklabel (GUID: B780C602-69EB-B14F-92DF-3483685CEB7B)

Command (m for help): u
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
admin2@solarrraspi: ~ $ sudo fdisk -l /dev/sda
Disk /dev/sda: 1.82 TiB, 2000398934016 bytes, 3907029168 sectors
Disk model: WDC WD20EFZX-68A
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: B780C602-69EB-B14F-92DF-3483685CEB7B
```

## Dateisystem EXT4 erstellen

`sudo mkfs.ext4 /dev/sda`

```
mke2fs 1.46.2 (28-Feb-2021)
Found a gpt partition table in /dev/sda
Proceed anyway? (y,N) y
Creating filesystem with 488378646 4k blocks and 122101760 inodes
Filesystem UUID: 11fb3855-d4b8-4f02-a0a9-49af140350dc
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

## Festplatte mounten

Verzeichnis erstellen und Rechte vergeben

`sudo mkdir /mnt/wd2tb`

`sudo chmod 777 -R /mnt/wd2tb`

Variante 1 – manuelles Mounten

Achtung nicht das media- sondern das mnt-Verzeichnis benutzen:

`sudo mount /dev/sda /mnt/wd2tb`

Variante 2 – automatisches Mounten bei jedem Start

ID der Festplatte ermitteln:

```
sudo blkid
```

```
/dev/sda: UUID="11fb3855-d4b8-4f02-a0a9-49af140350dc" BLOCK_SIZE="4096" TYPE="ext4"
```

ID in die fstab-Datei eintragen:

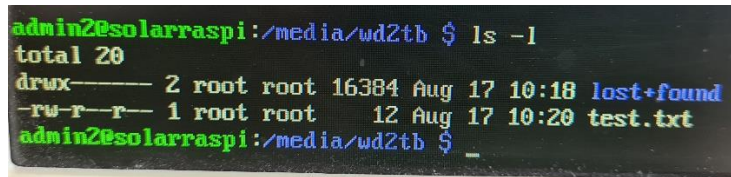
```
sudo nano /etc/fstab
```

diese Zeile hinzufügen

```
UUID=11fb3855-d4b8-4f02-a0a9-49af140350dc /mnt/wd2tb ext4 defaults,auto,users,rw,nofail,x-
gvfs-name=wd2tb 0 0
```

Test-Datei erstellen

test.txt erstellt mit sudo nano



```
admin2@solarraspi:/media/wd2tb $ ls -l
total 20
drwx----- 2 root root 16384 Aug 17 10:18 lost+found
-rw-r--r-- 1 root root    12 Aug 17 10:20 test.txt
admin2@solarraspi:/media/wd2tb $
```

Mount rückgängig machen

Umount /mnt/wd2tb

Wenn umount fehlschlägt, dann mit

sudo lsof /mnt/wd2tb

ermitteln, welche Prozesse auf das Verzeichnis zugreifen, z.B. die Samba-Freigaben. Diese Prozesse vor umount beenden.

## Samba und Freigaben -Zugriff auf die Festplatte von Windows aus

### SMB- Manual anzeigen

`man smb.conf`

### Status der Samba-Dienste abfragen

`sudo service smbd status`

`sudo service nmbd status`

Dienste laufen, wenn ‚active‘ angezeigt wird.

### Samba-Komponenten installieren

`sudo apt-get install samba samba-common`

ggf auch smbclient installieren

### Freigabe in die Konfig-Datei eintragen

`sudo nano /etc/samba/smb.conf`

[SambaWd2Tb]

comment = Samba-Freigabe fuer 2 TB WD Red NAS am Raspi

path = /mnt/wd2tb

### Samba-Dienste neu starten

`sudo service smbd restart`

`sudo service nmbd restart`

### User für Login beim Aufruf der Freigaben unter Windows

<https://www.elektronik-kompendium.de/sites/raspberry-pi/2007071.htm>

Wenn man sich unter Windows mit einer Samba-Freigabe eines Raspi verbinden möchte, wird man aufgefordert, Nutzer und Passwort einzugeben.

Dieser Nutzer muss im Raspi bereits existieren und hier noch einmal mit einem „SMB-Passwort“ registriert werden:

`sudo smbpasswd -a <user>`

zweimal das Passwort eingeben

Ggf noch mal die Dienste neu starten (s.o.)

Und schon funktionieren die Freigaben 😊



## MariaDB

Installation Version 10.5

<https://raspberrypi.com/install-mariadb-raspberry-pi/>

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install mariadb-server
```

Prüfen, wohin installiert wurde:

```
sudo find / -name "*mariadb*"
```

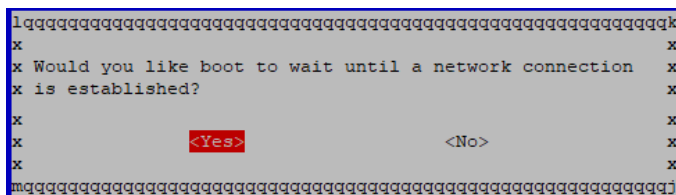
Sicherstellen, dass der Service nach dem Systemstart ausgeführt wird

Dazu muss der Raspi so konfiguriert werden, dass er beim Booten wartet, bis die Netzwerkverbindung steht. Das geht mit

Sudo raspi-config

1 .. system options

S6 .. network at boot



## Konfigurieren

```
sudo mysql_secure_installation
```

Passwort für root vergeben

### Weitere Einstellungen:

Switch to unix\_socket authentication [Y/n]: n

Remove anonymous users? [Y/n] y

Disallow root login remotely? [Y/n] n

Remove test database and access to it? [Y/n] y

Reload privilege tables now? [Y/n] y

## Wo liegen die Konfig-Dateien?

```
admin2@solarrraspi:/ $ cd /etc/mysql
admin2@solarrraspi:/etc/mysql $ ls- l
-bash: ls-: command not found
admin2@solarrraspi:/etc/mysql $ ls -l
total 24
drwxr-xr-x 2 root root 4096 Aug 24 08:41 conf.d
-rw----- 1 root root 544 Aug 24 08:42 debian.cnf
-rwxr-xr-x 1 root root 1731 Feb 10 2023 debian-start
-rw-r--r-- 1 root root 1126 Feb 10 2023 mariadb.cnf
drwxr-xr-x 2 root root 4096 Aug 24 08:42 mariadb.conf.d
lrwxrwxrwx 1 root root 24 Aug 24 08:41 my.cnf -> /etc/alternatives/my.cnf
-rw-r--r-- 1 root root 839 Feb 8 2021 my.cnf.fallback
admin2@solarrraspi:/etc/mysql $
```

## Data-Dir verschieben

Leider kann das Datenverzeichnis bei der Installation nicht angegeben werden und liegt nun im eMMC-Speicher des CM4-Moduls und muss noch auf die Sata-Platte verschoben werden.

<https://www.digitalocean.com/community/tutorials/how-to-change-a-mariadb-data-directory-to-a-new-location-on-centos-7>

<https://www.tecmint.com/change-default-mysql-mariadb-data-directory-in-linux/>

## Dienst stoppen

```
sudo service mysql stop
```

## Verzeichnis auf der Festplatte anlegen und Rechte vergeben

```
mkdir /mnt/wd2tb/mariadb/datadir
```

```
sudo chmod 777 -R /mnt/wd2tb
```

## Konfiguration korrigieren

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Eintrag ändern: datadir = /mnt/wd2tb/mariadb/datadir

## Bestehendes Verzeichnis verschieben

Mit Raspi-Dateiexplorer gescheitert.

```
sudo mv /var/lib/mysql/* /media/wd2tb/mariadb/datadir/ ok
```

## Dienst neu starten

```
sudo service mysql start
```

## Remotezugriff ermöglichen

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

bind-address auf 0.0.0.0 ändern

Dienst neu starten (s.u.)



### Mariadb-Service neu starten

```
sudo service mariadb restart
```

### Weiteren User anlegen

Nur möglich, wenn man als root angemeldet ist:

```
sudo mysql -uroot -p
```

pwd: hier eingeben

```
CREATE USER 'master'@'%' IDENTIFIED BY 'raspi';
```

### Anmelden

```
sudo mysql -uroot -p
```

oder

```
sudo mysql -umaster -p
```

pwd: hier eingeben

### Datenbank erstellen

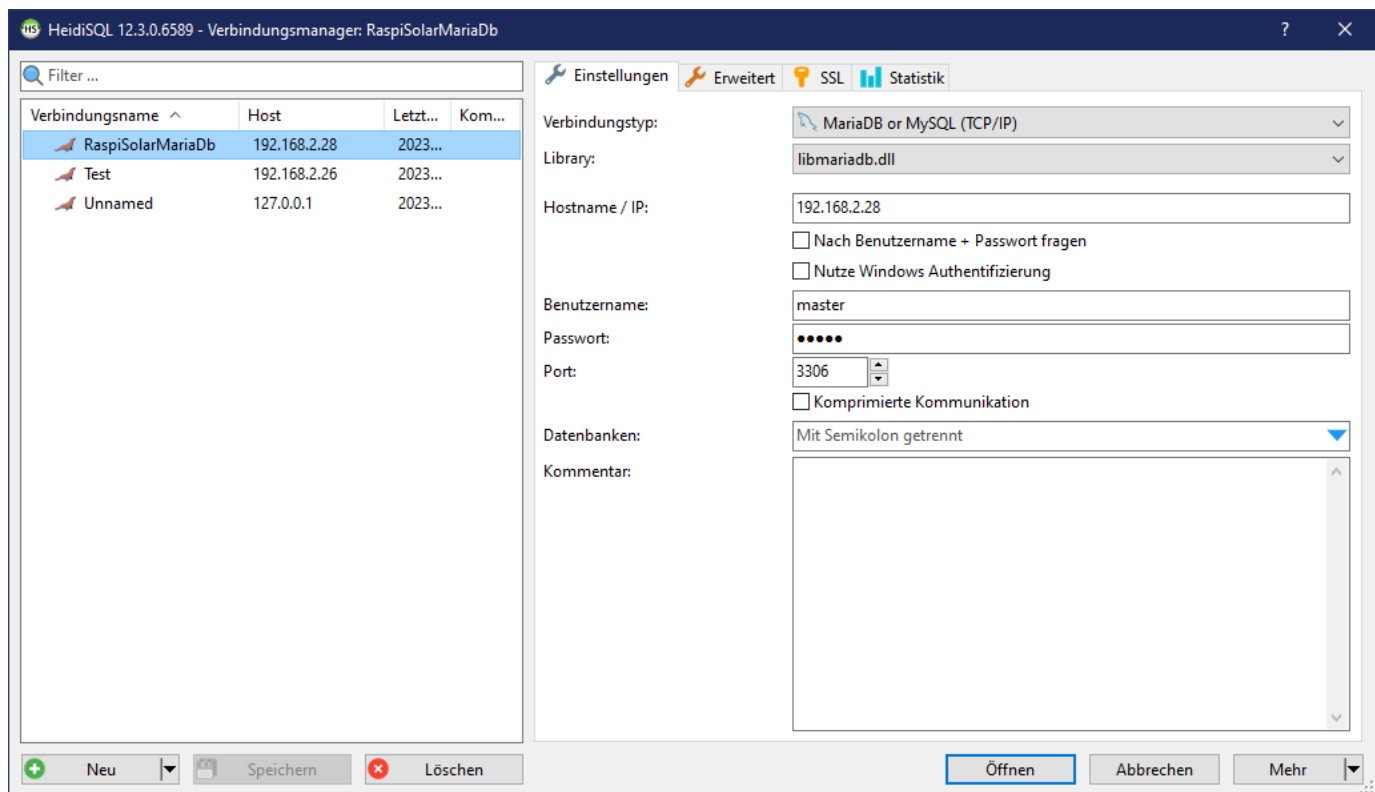
```
create database solar2023;
```

### Rechte für Datenbank vergeben

```
GRANT ALL PRIVILEGES ON solar2023.* TO 'master'@'%';
```

```
FLUSH PRIVILEGES;
```

Test des Remote-Zugriffs von HeidiSql auf MariaDb von Windows aus  
Ohne Probleme:



Log

**Nicht aktiviert**, weil es alles sehr langsam machen soll.

Abmelden

quit

# Python

## Installation

Das RaspberryPI OS (32bit), v11, bringt 3.9. bereits mit:

```
admin2@solarrraspi: ~  
admin2@solarrraspi:~ $ python  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux
```

## Python-Connector für die mariadb

Für

```
import mariadb
```

im python-Script wird der Connector gebraucht. Installation mit

```
pip install mariadb
```

Dabei wird automatisch versucht, bei 1.1.7 beginnend, eine funktionierende Version zu finden:

Rückwärts bis 1.1.2 brach die Installation mit Fehlern ab.

Aber 1.0.11 wurde dann installiert:

```
Downloading https://www.piwheels.org/simple/mariadb/mariadb-1.0.11-cp39-cp39-linux_armv7l.whl (198 kB 1.3 MB/s)  
Installing collected packages: mariadb  
Successfully installed mariadb-1.0.11
```

## Script timergesteuert ausführen

Für Ausführung als root eintragen mit sudo crontab -e.

Für Ausführung als angemeldeter Nutzer: crontab -e

Beispiel: zwei Zeilen eingetragen, um ein shellscript zu starten:

```
# von 8-20 Uhr in Minute 50 das shell script ausfuehren
```

```
50 8-20 * * * sh /mnt/wd2tb/releases/meteoblue_forecast/mb_pvpro.sh
```

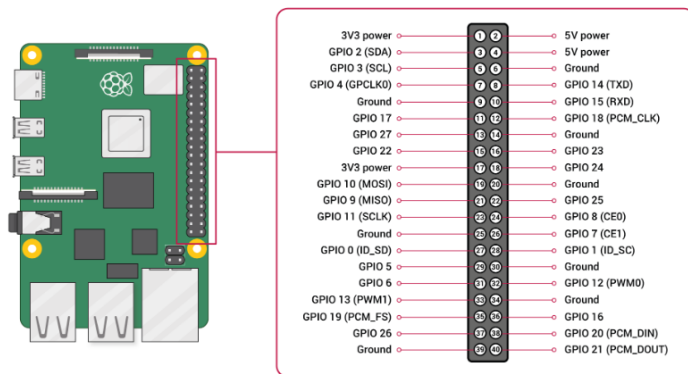
Beispiel für ein shell-Script:

```
cd /mnt/wd2tb/script/meteoblue_forecast
```

```
# Solarprognose von MeteoBlue holen und speichern
```

```
python mb_pvpro.py
```

## GPIO-Aktoren und -Sensoren



Aktoren: 3-Kanal-Relais-Board

[https://www.waveshare.com/wiki/RPi\\_Relay\\_Board](https://www.waveshare.com/wiki/RPi_Relay_Board)



Channel No.	RPi Pin No.	wiringPi	BCM	Descriptions
CH1	37	P25	26	Channel 1
CH2	38	P28	20	Channel 2
CH3	40	P29	21	Channel 3

Testscript zum Ansteuern der Relais siehe

[https://github.com/grasmax/AcOnOff/blob/main/gh\\_gpiorelaytest.py](https://github.com/grasmax/AcOnOff/blob/main/gh_gpiorelaytest.py)

## Sensoren

Wichtig: Die als Sensor geschalteten Pins müssen mit einem Pulldown-Widerstand versehen werden.

Dieser sollte 10-100 kΩ haben.

Alternativ kann der interne Pulldown aktiviert werden:

`GPIO.setup(p, GPIO.IN, GPIO.PUD_DOWN)` # den internen pulldown-Widerstand aktivieren

Testscript

[https://github.com/grasmax/AcOnOff/blob/main/gh\\_gpiointest.py](https://github.com/grasmax/AcOnOff/blob/main/gh_gpiointest.py)

## Weitere nützliche Befehle

### Ausschalten

`systemctl poweroff -i`

`sudo shutdown -h now`

### Neustart

`sudo shutdown -r now`