

Einrichten des Raspberry Pi

RPi-PC

Version 1.1

Autor: eQ-3 Entwicklung GmbH
Erstelldatum: 14. Januar 2019
Letzte Aktualisierung: 20.Februar.2020, 13:07
Dateiname: RPi-PC_Einrichtung.doc

Tel: +49 (0)491 6008 700
Fax: +49 (0)491 6008 99 700
Mail: info@entwicklung.eq-3.de

**eQ-3
Entwicklung GmbH**
Maiburger Straße 36
26789 Leer

Geschäftsführer
Prof. H.-G. Redeker

Registergericht:
Amtsgericht Aurich
HRB 110388

Zentrale:
Telefon: +49 (0)491 6008 700
Telefax: +49 (0)491 6008 99 700

Internet:
www.eq-3.de

1 Inhaltsverzeichnis

1	INHALTSVERZEICHNIS.....	2
2	ÄNDERUNGSVERZEICHNIS.....	3
3	ZIEL	4
4	GRUNDLAGEN	4
4.1	Image auf SD-Karte schreiben.....	4
4.1.1	Betriebssystem MS Windows:.....	4
4.1.2	Betriebssystem Linux:.....	5
4.2	Zugriff über Netzwerk.....	5
4.3	Texteditor Nano	7
4.4	Rechte mittels Sudo erweitern	8
5	EINRICHTEN DER VERSCHIEDENEN BETRIEBSSYSTEME.....	8
5.1	Raspbian (2018-06-27) und Retropi (4.4)	8
5.2	Volumio (2.452).....	9
5.3	RecalBox (2018-07-13).....	9
5.4	LibreELEC (8.2.5)	10
6	INHALT DER SKRIPTE:.....	10
6.1	Inhalt autostart.sh.....	10
6.2	Inhalt rpi-pc_shutdown.sh	11
6.3	Inhalt rpi-pc_shutdownService.sh	12

2 Änderungsverzeichnis

Änderung			Geänderte Kapitel	Beschreibung der Änderung	Autor
Nr.	Datum	Version			
1	18.01.19	1.0			
2	20.02.20	1.1	6.2	Fehlendes Zeichen im Skript ergänzt	

Tabelle 1: Änderungsverzeichnis

3 Ziel

Mit diesem Dokument wird die Einrichtung des Raspberry Pi für den Power-Controller RPI-PC erläutert.

4 Grundlagen

4.1 Image auf SD-Karte schreiben

Laden Sie sich ihr gewünschtes Betriebssystem für den Raspberry Pi vorab herunter. Anschließend schreiben Sie das heruntergeladene Image auf eine microSD-Karte oder folgend Sie der Anleitung des Anbieters des Betriebssystems.

4.1.1 Betriebssystem MS Windows:

Unter Windows kann beispielsweise das Tool „Win32DiskImager“ verwendet werden. Win32DiskImager ist nach dem Download zu entpacken und als Administrator auszuführen. Es erscheint das Programm-Fenster, dort wird bei „Image File“ das heruntergeladene Image angegeben. Unter „Device“ ist das Laufwerk der SD-Karte auszuwählen. Danach wird der Schreibvorgang mittels „Write“ gestartet, dieser Vorgang dauert einige Zeit.

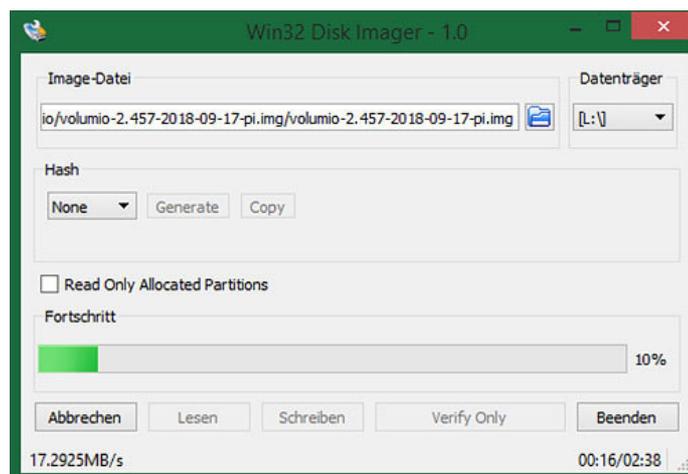


Abbildung 1: Win32 Disk Imager

Nachdem der Vorgang abgeschlossen ist, kann das Programm beendet werden und die SD-Karte ist bereit für den Einsatz im Raspberry Pi.

4.1.2 Betriebssystem Linux:

Unter Linux kann das Tool „dd“ [5] zu verwendet werden.

Mittels dem Befehl „df-h“ lassen wir alle Laufwerke auflisten, nun stecken wir unseren Kartenleser an und lassen uns die Laufwerke erneut auflisten. Bei dem neu erscheinenden Laufwerk handelt es sich um die SD-Karte, typisch sind Bezeichnungen wie „/dev/mmcblk0p1“ oder „/dev/sdd1“. Das p1 oder 1 am Ende ist die Kennzeichnung der Partition, da die ganze SD-Karte genutzt werden soll, müssen diese in den folgenden Schritten entfernt werden „/dev/mmcblk0“ oder „/dev/sdd“.

Es können auch mehrere Partitionen auf einer SD-Karte vorhanden sein. Um die Karte neu beschreiben zu können müssen alle Partitionen freigegeben werden:

```
umount /dev/sdd1
```

In dem folgenden Befehl sind die Parameter zu beachten. Bei if= ist der Pfad zu unserem Image anzugeben, bei of= ist die SD-Karte einzutragen:

```
sudo dd bs=4M if=~/.2012-12-16-wheezy-raspbian.img of=/dev/sdd
```

Um sicher zu gehen, dass alles auf die Karte geschrieben wurde und wir diese entfernen können:

```
sudo sync
```

Nun ist die Karte für den Einsatz im Raspberry Pi.

4.2 Zugriff über Netzwerk

Zuerst müssen Sie den Raspberry Pi in das Netzwerk einbinden. Danach muss für den Zugang SSH beim Einrichten des Betriebssystems aktiviert werden oder nachträglich z. B. über das Menü von Kodi oder mittels raspi-config.

Nach einem Neustart sollte der Zugang über SSH nun möglich sein, dafür wird die IP-Adresse des Raspberry Pi benötigt. Diese ist in den Bootinformationen zu

sehen oder im Menü zu finden. Alternativ können Sie auch in Ihrem Router nachsehen, welche IP-Adresse dem Raspberry Pi zugeordnet wurde.

Über das Programm PuTTY oder ein beliebiges anderes Terminal Programm können wir eine Verbindung zu dem Raspberry Pi über das SSH Protokoll herstellen, dafür bei Hostname die IP-Adresse (hier beispielsweise 172.25.50.51), bei Port 22 eintragen und bei Connection type SSH auswählen.

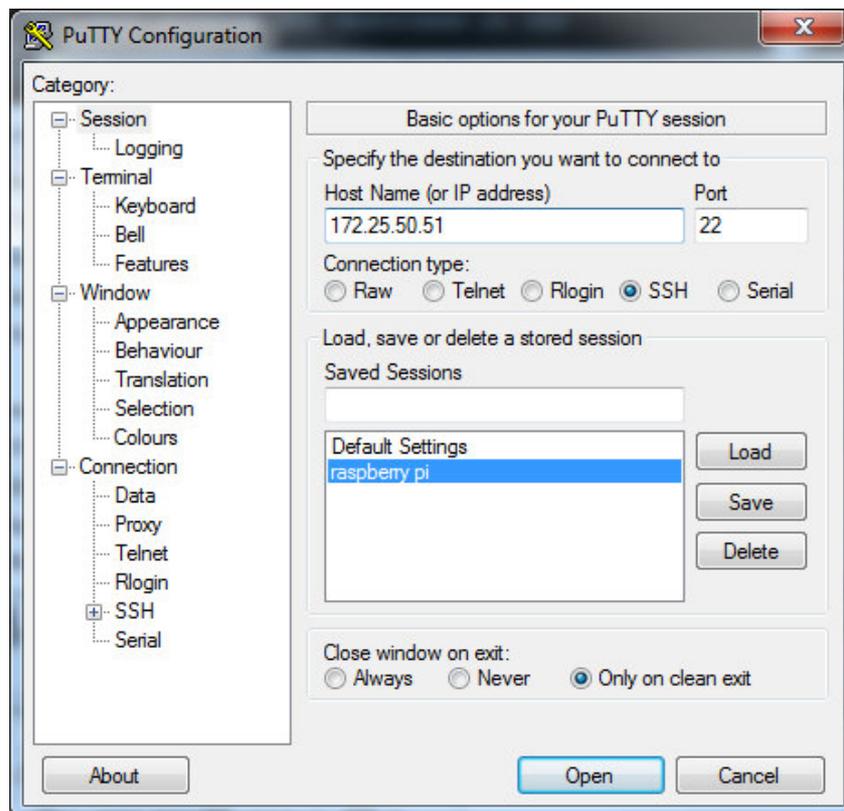
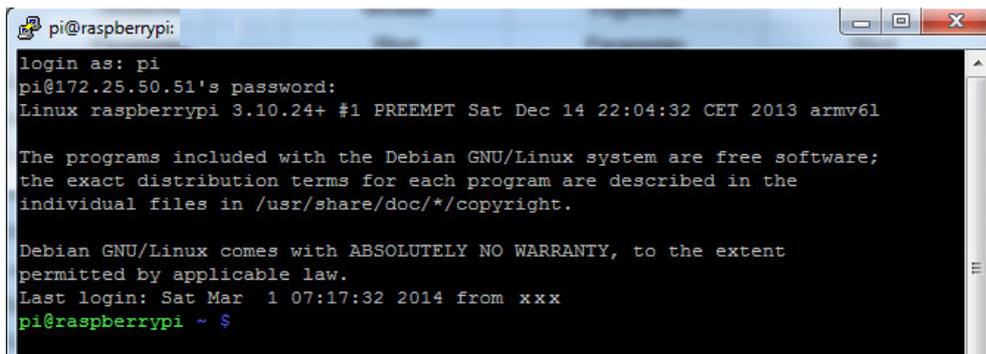


Abbildung 2: PuTTY Configuration

Nach Klick auf Open öffnet sich ein Fenster, wo wir nach Benutzernamen und Passwort vom Raspberry Pi gefragt werden.



```
pi@raspberrypi:
login as: pi
pi@172.25.50.51's password:
Linux raspberrypi 3.10.24+ #1 PREEMPT Sat Dec 14 22:04:32 CET 2013 armv6l

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.

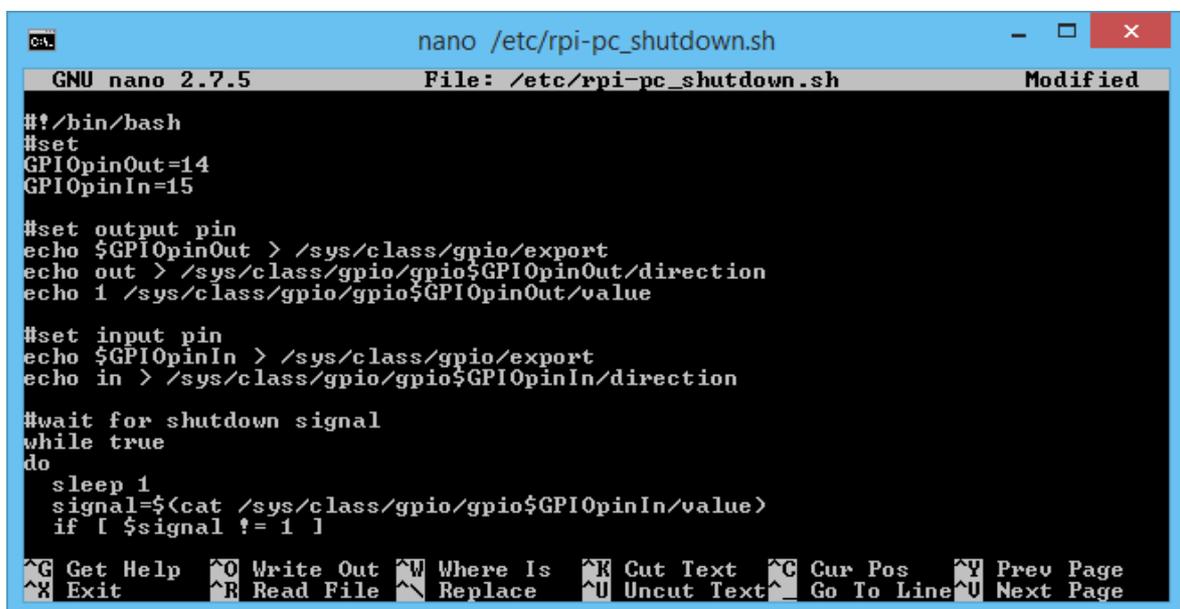
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar  1 07:17:32 2014 from xxx
pi@raspberrypi ~ $
```

Abbildung 3: PuTTY Terminal Fenster

4.3 Texteditor Nano

Nano ist ein einfacher freier Texteditor für Linux/Unix, welcher zur Bearbeitung von Dateien in der Konsole oder über ein Terminal-Zugang eingesetzt werden kann. Nano bietet im Vergleich zu anderen Editoren einen geringeren Funktionsumfang, ist dadurch einfacher zu bedienen und für einfache Anpassungen wie in unserem Fall vollkommen ausreichend.

Wichtig beim Umgang mit Nano sind die Tastenkombinationen STRG+O zum Speichern und STRG+X zum Schließen der Datei.



```
nano /etc/rpi-pc_shutdown.sh
GNU nano 2.7.5 File: /etc/rpi-pc_shutdown.sh Modified
#!/bin/bash
#set
GPIOpinOut=14
GPIOpinIn=15

#set output pin
echo $GPIOpinOut > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio$GPIOpinOut/direction
echo 1 /sys/class/gpio/gpio$GPIOpinOut/value

#set input pin
echo $GPIOpinIn > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio$GPIOpinIn/direction

#wait for shutdown signal
while true
do
sleep 1
signal=$(cat /sys/class/gpio/gpio$GPIOpinIn/value)
if [ $signal != 1 ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^M Replace ^U Uncut Text ^G Go To Line ^N Next Page
```

Abbildung 4: Editor Nano

4.4 Rechte mittels Sudo erweitern

Einige Dateien und Befehle lassen sich in Linux nur mit Root-Rechten ausführen, da z. B. der normale Benutzer Pi diese Rechte nicht direkt besitzt, können wir über den Befehl `sudo` einzelne Befehle/Programme mit den Rechten eines anderen Benutzers (Root) ausführen.

5 Einrichten der verschiedenen Betriebssysteme

Je nach verwendetem Betriebssystem unterscheidet sich die Installation des RPi-PC-Shutdown-Skriptes, wobei das eigentliche Skript immer dasselbe bleibt. Lediglich das Einfügen in die jeweilige Autostart Sequenz ist unterschiedlich. Dazu müssen mit einem Editor, z. B. nano, Dateien neu erstellt oder bearbeitet und anschließend gespeichert werden. Der Inhalt der Skripte folgt im nächsten Abschnitt.

5.1 Raspbian (2018-06-27) und Retropi (4.4)

Benutzername: `pi`

Passwort: `raspberrry`

Über die Konsole bzw. über Terminal einloggen und folgenden Befehle ausführen:

```
sudo nano /etc/rpi-pc_shutdown.sh
```

Inhalt von `rpi-pc_shutdown.sh` einfügen und speichern, danach die Datei ausführbar machen. Dafür werden mittels dem Programm `chmod` die Dateirechte um das Ausführ-Recht (x) erweitert:

```
sudo chmod +x /etc/rpi-pc_shutdown.sh
```

Das gerade erstellt Skript muss in den Autostart einfügen, dafür wird die `rc.local` editiert:

```
sudo nano /etc/rc.local
```

Über `rc.local` können Programme und Skripte sehr einfach bei einem Systemstart automatisch geladen werden, indem wir unser Skript einfach in die Datei eintragen. Die Datei `rc.local` wird bei Systemstart automatisch mit Root-Rechten ausgeführt, sodass hier das `sudo` entfällt. Wichtig ist, dass „`exit 0`“ muss am Ende

der Datei stehen bleiben. Das & am Ende unseres Eintrags bewirkt, dass unser Skript in einer neuen Shell im Hintergrund ausgeführt wird.

Fügen Sie vor „exit 0“ folgende Zeile ein:

```
(/etc/rpi-pc_shutdown.sh)&
```

Anschließend wird auch diese Datei gespeichert und ebenfalls ausführbar gemacht:

```
sudo chmod +x /etc/rc.local
```

5.2 Volumio (2.452)

Benutzername: *volumio*

Passwort: *volumio*

Über SSH einloggen und die Schritte wie bei Raspbian durchführen.

5.3 RecalBox (2018-07-13)

username: *root*

password: *recalboxroot*

Über SSH einloggen und das Dateisystem für den Schreibzugriff vorbereiten.

Bei RecalBox ist das Dateisystem normalerweise schreibgeschützt, deswegen müssen wir das Verzeichnis mit schreibrechten öffnen, dies geschieht über den Befehl mount mit der Option -o remount gefolgt von den Zugriffsrechten (rw für Lese- und Schreibzugriff) und dem Root-Verzeichnis:

```
mount -o remount,rw /
```

Erst danach kann das Skript erstellt und gespeichert werden.

```
nano /etc/rpi-pc_shutdown.sh
```

Inhalt von rpi-pc_shutdown.sh einfügen und speichern, danach die Datei ausführbar machen:

```
chmod +x /etc/rpi-pc_shutdown.sh
```

Das gerade erstellte Skript als Service in den Autostart einfügen, dafür wird folgendes Skript neu erstellt. (Die Zahl bitte durch eine noch nicht vorhandene ersetzen).

```
nano /etc/init.d/S80_rpi-pc_shutdownService.sh
```

Inhalt von S80_rpi-pc_shutdownService.sh einfügen, speichern und die Datei ausführbar machen:

```
chmod +x /etc/init.d/S80_rpi-pc_shutdownService.sh
```

Zum Abschluss versehen wir das Verzeichnis wieder mit Schreibschutz (nur Leserechte):

```
mount -o remount,r /
```

5.4 LibreELEC (8.2.5)

username: *root*

password: *Libreelec*

Im Menü SSH aktivieren und über Terminal einloggen:

```
nano /storage/.config/rpi-pc_shutdown.sh
```

Inhalt von rpi-pc_shutdown.sh einfügen und speichern

```
chmod +x /storage/.config/rpi-pc_shutdown.sh
```

Das gerade erstellte Skript in den Autostart einfügen, dafür wird folgendes Skript neu erstellt bzw. nur erweitert.

```
nano /storage/.config/autostart.sh
```

Inhalt von autostart.sh einfügen, speichern und die Datei ausführbar machen:

```
chmod +x /storage/.config/autostart.sh
```

6 Inhalt der Skripte:

Die nun folgenden Blöcke müssen jeweils in die mittels des Editors Nano erstellten Dateien eingefügt werden:

6.1 Inhalt autostart.sh

```
#!/bin/bash
(/storage/.config/rpi-pc_shutdown.sh)&
```

6.2 Inhalt rpi-pc_shutdown.sh

```
#!/bin/bash
#set
GPIOpinOut=14
GPIOpinIn=15

#set output pin
echo "$GPIOpinOut" > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio$GPIOpinOut/direction
echo "1" > /sys/class/gpio/gpio$GPIOpinOut/value
#set input pin
echo "$GPIOpinIn" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio$GPIOpinIn/direction

#wait for shutdown signal
while true
do
    sleep 1
    signal=$(cat /sys/class/gpio/gpio$GPIOpinIn/value)
    if [ $signal != 1 ]
    then
        echo "Shutdown Raspberry"
        shutdown -h now
        #echo "0" > /sys/class/gpio/gpio$GPIOpinOut/value
        exit 0
    fi
done
```

Inhalt rpi-pc_shutdownService.sh

```
#!/bin/bash
### BEGIN INIT INFO
# Provides: rpi-pc_shutdown.sh
# Required-Start: $local_fs $remote_fs
# Required-Stop: $local_fs $remote_fs
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start power controller daemon at boot time
# Description: Enable power controller service provided by daemon.
### END INIT INFO
case "$1" in
    start)
        echo "Starting script rpi-pc_shutdown.sh"
        /etc/rpi-pc_shutdown.sh&
        ;;
    stop)
        echo "Stopping script /rpi-pc_shutdown.sh"
        killall rpi-pc_shutdown.sh
        ;;
    *)
        echo "Usage: /etc/init.d/rpi-pc_shutdownService {start/stop}"
        exit 1
        ;;
esac
exit 0
```