

---

# Stephans Homepage

Modellbahn, Motorrad und dann mal sehen...

---

[Home](#)

[Modellbahn](#)

[Motorrad fahren](#)

[Stephans Blog](#)

[Podcasts und Links](#)

[Über mich](#)

[Kontakt](#)

[Impressum, Haftungsausschluss und Datenschutzerklärung](#)

---

## MoBa-Pi Software Quick Guide

Liste der benötigten SW:

1. [Raspbian Jessie](#)
2. [srcpd](#)
3. [s88udp](#)
4. [bcm2835 Bibliothek](#) (für s88udp)
5. [Rocrail](#)

Tools (die werdet Ihr aber sowieso haben):

1. [Putty](#)
2. [FileZilla](#)

Auf dem Raspberry Pi wird die letzte [Raspbian Jessie](#) Version installiert.

Wie das geht ist [hier](#) beschrieben.

Dann müssen ein paar Settings mit raspi-config gemacht werden:

1. Serielle Schnittstelle zur Ausgabe von Logs abschalten.
2. ssh einschalten.

### Seitennavigation

- [Home](#)
- [Modellbahn](#)
  - [Warum Spur N?](#)
  - [Modellbahn Digitalsteuerung](#)
    - [Welche Digitalzentrale ist die beste?](#)
    - [Rückmeldung](#)
  - [Stromfühler](#)
  - [MoBa-Pi](#)
    - [MoBa-Pi Hardware](#)
    - [MoBa-Pi3 Hardware](#)
    - [MoBa-Pi Software](#)
      - [MoBa-Pi Software Quick Guide](#)
  - [Modellbahn Versuchsanlage](#)

3. Wer mag ändere das Passwort (ich nehme immer ein kürzeres, schließlich geht es um Modellbahn und nicht um Banking).
4. Boot to console (x brauchen wir nicht und kostet nur unnötig Rechenzeit).

Wer mag kann in .bashrc seine Alias Settings bearbeiten (ich schalte mir immer „ll“ frei):

```
cd /home/pi  
sudo nano .bashrc
```

Dann erst mal alles auf den neuesten Stand bringen:

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

### Das folgende gilt nur für den Raspberry Pi3!

Beim Pi3 hat die Raspberry Foundation Bluetooth eingebaut. Dummerweise haben sie den Bluetooth Chip an den von uns für den srcpd benötigten UART angeschlossen. Aber das kann man umbiegen. Bluetooth geht dann natürlich nicht mehr, aber das brauchen wir ja auch nicht.

Pi3 Bluetooth ausschalten und UART0/ttyAMA0 über die GPIOs 14 & 15 wieder herstellen.

config.txt öffnen um den UART wieder mit dem GPIO 14 & 15 zu verbinden:

```
sudo nano /boot/config.txt
```

und am Ende der Datei folgende Zeile einfügen:

```
dtoverlay=pi3-disable-bt
```

Jetzt muss noch das Bluetooth ausgeschaltet werden, dazu folgenden Befehl einmalig eingeben:

```
sudo systemctl disable hciuart
```

Jetzt erst mal neu starten.

**Ab hier geht es wieder für alle weiter.**

srcpd:

- [Aufbau der Versuchsanlage](#)
- [neue Modellbahn Versuchsanlage](#)
- [Motorrad fahren](#)
  - [Wie ich zu meiner XJ 600 N gekommen bin](#)
  - [Wie ich das Motorrad auf den ersten Winter vorbereitet habe](#)
  - [Was mir gut geholfen hat](#)
  - [Ein Motorrad, das mir wie angegossen passt!](#)
- [Stephans Blog](#)
- [Podcasts und Links](#)
  - [Mein eigener Podcast](#)
  - [Über mich](#)
  - [Kontakt](#)
  - [Impressum, Haftungsausschluss und Datenschutzerklärung](#)

Mit FileZilla den srcpd kopieren und anschließend installieren:

```
tar -xzf srcpd-2.1.3.tar.gz  
cd srcpd-2.1.3/  
sudo apt-get install libxml2 libxml2-dev telnet
```

(„J“ antworten)

```
./configure  
make
```

Wenn als letzte Zeile „make[1]: Leaving directory  
./home/pi/srcpd-2.1.3“ stehenbleibt, hat das Erstellen  
geklappt, und das Programm kann installiert werden.

```
sudo make install
```

Dann muss die [srcpd.conf](#) bearbeitet werden. Die von mir  
verwendete Version könnt Ihr Euch [hier](#) runter laden und  
entweder selbst benutzen oder als Vorlage für Eure  
eigene verwenden. (Nach dem kopieren dos2unix  
verwenden.)

```
cd /usr/local/etc/  
sudo nano srcpd.conf
```

Jetzt kann der srcpd getestet werden:

```
/usr/local/sbin/srcpd
```

Wer einen srcpd Client hat (z.B. [j-man](#)) kann jetzt die  
Funktion testen.

Ist alles erfolgreich verlaufen kann man den srcpd beim  
Start des Raspberry Pi direkt mit starten. Achtung, beim  
Raspberry Pi3 haben wir oben die UART – GPIO  
Verbindung umgebogen. Meine Erfahrung hat mir  
gezeigt, dass es Probleme gibt, wenn srcpd zu früh  
gestartet wird, da dann die Verbindung von UART zu  
GPIO noch nicht gemacht ist.

Ein einfaches „sleep“ in der rc.local reicht aber auch  
nicht, da meiner Erfahrung nach auch der Prozess  
pausiert, der sich um die UART – GPIO Verbindung  
kümmert.

Die Lösung besteht darin, die rc.local abarbeiten zu  
lassen und trotzdem den srcpd Client später zu starten,

das machen wir wie folgt.

```
sudo nano /etc/rc.local <- öffnen der rc.local
```

Diese Zeile vor exito einfügen:

```
(/bin/sleep 30 && /usr/local/sbin/srcpd) &
```

sleep 30 verzögert den Start um 30 Sekunden und das „&“ hinter der Klammer sorgt dafür, dass der Befehl in einer eigenen bash ausgeführt wird. Dadurch wird dann rc.local direkt weiter abgearbeitet.

#### Shutdown Taster:

Als nächstes richten wir das Ausschalten des MoBa-Pi per Taster ein, alle details dazu findet Ihr in diesem Forumsbeitrag:

<http://www.forum-raspberrypi.de/Thread-tutorial-hoch-und-runterfahren-mittels-taster-incl-status-led>

Folgende Pakete installieren (kann sein, dass es die schon gibt, macht aber nix):

```
sudo apt-get install python-dev
```

```
sudo apt-get install python-rpi.gpio
```

Shutdown.py [hier](#) laden und dann mit dem FileZilla [shutdown.py](#) ins Verzeichnis /home/pi/ kopieren.

Rechte anpassen und in rc.local einfügen.

```
sudo chmod +x /home/pi/shutdown.py
```

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

Diese Zeile vor exito einfügen:

```
(/bin/sleep 90 && python  
/home/pi/shutdown.py) &
```

#### s88udp:

Wie auf der Seite von [Siggi](#) beschrieben wird [s88udp](#) auf dem MoBa-Pi installiert.

Erst muss die [Bibliothek bcm2835](#) installiert werden von der ich die Version 1.50 der Bibliothek verwende. Laden und mit FileZilla kopieren, dann installieren.

```
tar zxvf bcm2835-1.50.tar.gz
```

```
cd bcm2835-1.50
```

```
./configure  
make  
sudo make check  
sudo make install
```

Jetzt könne wir s88udp auf den MoBa-Pi kopieren und installieren.

```
tar -xzf s88udp.tar.gz  
cd s88udp  
make  
sudo make install
```

Auf [s88udp.de](http://s88udp.de) ist erklärt, wie man das testen kann.

Dann noch folgende Zeile in rc.local einfügen.

```
(/bin/sleep 40 && /usr/local/bin/s88udp -m x)  
& <- x ist die Anzahl der angeschlossenen Module.
```

Rocrail:

Rocrail kann als fertiges .deb Paket [hier](#) von der [Rocrail Seite](#) geladen werden.

Das Paket dann per FileZilla nach /home/pi/ auf den Raspberry Pi kopieren.

Die Installation des Paketes erfolgt mit

```
sudo dpkg -i rocrail-11465-linuxarmhf-wx3.0-  
armhf.deb
```

Leider ergeben sich Abhängigkeiten.

```
dpkg: Abhängigkeitsprobleme verhindern  
Konfiguration von rocrail:  
rocrail hängt ab von libwxgtk3.0-dev |  
libwxgtk2.8-dev; aber:  
Paket libwxgtk3.0-dev ist nicht installiert.  
Paket libwxgtk2.8-dev ist nicht installiert.
```

Diese lassen sich aber automatisch mit

```
sudo apt-get -f install
```

auflösen.

Danach wird die Installation einfach noch einmal versucht:

```
sudo dpkg -i rocrail-11465-linuxarmhf-wx3.0-  
armhf.deb
```

War die Installation erfolgreich, finden wir Rocrail nun unter

```
cd /opt/rocrail/
```

und wir können Rocrail mit

```
./rocrail.sh
```

testweise starten.

Vom PC aus sollten wir Rocview nun mit Rocrail auf dem MoBa-Pi verbinden können.

Jetzt muss unter Rocrail noch der srcpd als Zentrale eingerichtet werden.

Schnittstellenkennung: srcp

Hostname: 127.0.0.1

Port: 4303

Gerät: /dev/ttyAMA0

Sub-Bibliothek: TCPIP

Auch Rocrail lässt sich natürlich beim Booten des Raspberry Pi automatisch starten, dazu noch folgende Zeile in rc.local einfügen:

```
(/bin/sleep 60 && su pi -c '/opt/rocrail/r
```

Die von mir verwendete [rc.local](#) könnt Ihr Euch [hier](#) anschauen.

Und nun viel Spaß mit dem MoBa-Pi!

-Stephan