

Raspberry Pi für die Bastel-/Programmierarbeit einrichten + Nutzungstipps

Für größere Bastelvorhaben: Der erste RPi wird “normal” eingerichtet, ab danach wird seine SD-Karte für weitere Rechner gecloned.

1. Zuerst das NOOBS-Image von raspberrypi.org herunterladen, auf eine (ansonsten leere) (Micro-)SD-Karte kopieren, diese in den RPi stecken (noch keine Netzwerkverbindung! Sicherheit!), diesen einschalten, dann Raspbian als Betriebssystem wählen. Alternative: Direkt das Raspbian-Image auf die Karte spielen. Läuft schon irgendwo ein Linux (z.B. ein anderer RPi), geht das mit dem Kommando `dd`. Nach dem ersten Start die Expansion der Root-Partition anstoßen.
2. Beim ersten Start das **Password** des Users `pi` ändern! Danach eine Netzwerkverbindung herstellen.
3. Nötige Einstellungen (Locales, Tastaturbelegung, Zeitzone) vornehmen¹ (später erneut via “`sudo raspi-config`”).
4. Software updaten: (*hierbei muss bei Nachfrage das in Schritt 2 vergebene Passwort eingegeben werden*)

```
sudo apt-get update ; sudo apt-get upgrade
sudo apt-get install synaptic joe avrdude gcc-avr arduino arduino-mk xombrero screen gedit
sudo apt-get install gedit-plugins atool tcl-tclreadline xfig xfig-doc mercurial apt-file
sudo apt-get install locate evince git checkinstall gdebi raspi-gpio libx11-dev imagemagick
```

Spätere (Sicherheits-)Updates stets mit “`sudo apt-get update;sudo apt-get upgrade`” oder via `synaptic` anstoßen.

5. Der RPi kann nun für das Programmieren von **Arduino-Boards** verwendet werden, welche die WS2812B-LED-Ketten z.B. autonom steuern können.
Außerdem enthält er nun Grundlagen, um selbst WS2812B-LED-Ketten anzusteuern. Programme können entweder in der `arduino-IDE` oder in `gedit` entwickelt werden.
6. Noch zur Information: Falls **weitere Software** benötigt wird, lässt sie sich leicht über das Programm `synaptic` anwählen (ca. 40000 Programme stehen zur Auswahl). Fehlt ein bestimmter Befehl und ist nicht klar, in welchem Paket er steckt, kann in der Konsole ein

```
apt-file find Name_des_gesuchten_Befehls
```

helfen, also z.B. `apt-file find arduino`.

Weitere *extern* angebotene Software kann bei Bedarf ebenfalls vom RPi aus geholt werden (z.B. aus Github-Repositories), kompiliert und installiert werden.² Statt “`sudo make install`” sollte man dabei besser “`sudo checkinstall`” nehmen. Dann kann man a) alles gut im Paketmanager wieder entfernen und man hat b) eine `*.deb`-Installationsdatei, um sie z.B. auf weitere Rechner zu verteilen.

Sucht man im laufenden Betrieb eine bestimmte **Datei** oder ein bestimmtes **Verzeichnis**, so kann ein

```
locate Name_des_Gesuchten
```

helfen.

Die Datenbank von `locate` wird mit `updatedb` aktualisiert, was aufzurufen vor der ersten Nutzung von `locate` und nach Änderung von vielen Dateien sinnvoll sein kann.

Innerhalb eines Verzeichnisses findet

```
find -name Name_des_Gesuchten
```

mehr (**Suche nach Dateinamen**, die “`Name_des_Gesuchten`” enthalten).

Um nach **Zeichenketten in (allen) Dateien zu suchen**, bemüht man:

```
fgrep Gesuchte_Zeichenkette *
```

Gibt man einen Dateinamen statt “`*`” an, wird nur in dieser Datei gesucht. Hängt man dem Befehl noch ein “`-R`” an, sucht er auch in Unterverzeichnissen.

Ein guter **Dokumentenbetrachter**, z.B. für PDFs, ist `evince`.

Ein schneller **Browser** ist `xombrero`. Damit er JavaScript versteht (das muss man bei ihm seitenweise *aktivieren* — ist halt ein Hacker-Browser...), muss man STRG-J drücken, während die Seite geöffnet ist.

(Netzwerk- und andere) **Drucker** richtet man ein, indem man die Seite `http://localhost:631` im Browser öffnet.

Innerhalb einer Konsole wechselt man mit `cd Name` das Verzeichnis, mit `mkdir Name` legt man ein neues Verzeichnis an, mit `ls` listet man die vorhandenen Dateien (ausführlicher mit `ls -l` oder mit `ls -al` auch für versteckte Dateien).

Programmskripte macht man mit `chmod +x Name_des_Skripts` ausführbar.

¹Locales: mit Leertaste (nur) `de_DE.UTF-8` wählen / Zeitzone: `Europe` → `Berlin` / Tastatur: German mit “dead grave acute”.

²Hierzu noch ein Tipp, den Nicht-Linuxer zu Beginn mitunter benötigen: Steht in Anleitungen ein “`#`” als Prompt (statt “`$`”) vor einem Befehl, muss man Root-Rechte zu seiner Ausführung haben. Dazu setzt man ein “`sudo`” vor den Befehl. Den Prompt gibt man jeweils *nicht* mit ein. Fertige `*.deb`-Dateien kann man mit “`sudo gdebi datei.deb`” installieren, das holt dann evtl. Benötigtes selbständig mit.