

# Raspberry Pi für die Programmierarbeit einrichten + Nutzungstipps

Mattscheibe, Januar 2021

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

1. Zuerst RaspberrypiOS von [raspberrypi.org](http://raspberrypi.org) herunterladen und auf eine (ansonsten leere) (Micro-)SD-Karte installieren (Anleitung auf der Seite!)
2. Beim ersten Start das **Password** des Users `pi` ändern! Danach eine Netzwerkverbindung herstellen.
3. Nötige Einstellungen (Locales, Tastaturbelegung, Zeitzone) vornehmen<sup>1</sup> (später erneut via “`sudo raspi-config`” oder im Einstellungsdialog für die Maus).
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 screen gedit git
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.<sup>2</sup>

6. Noch zur Information: Falls **weitere Software** benötigt wird, lässt sie sich auch leicht über das Programm `synaptic` anwählen (ca. 60000 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` . (Vorher einmal `sudo apt-file update` laufenlassen.)

Weitere *extern* angebotene Software kann bei Bedarf ebenfalls vom RPi aus geholt werden (z.B. aus Github-Repositories), kompiliert und installiert werden.<sup>3</sup> 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 und seiner Unterverzeichnisse 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 z.B. `chromium-browser`.

(Netzwerk- und andere) **Drucker** richtet man ein, indem man die Seite <http://localhost:631> im Browser öffnet, nachdem man zuvor das Paket `cups` installiert hat.

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. Man führt sie mit `./name_des_Skripts` von ihrem Verzeichnis aus aus.

---

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

<sup>2</sup>Die Arduino-IDE sollte aber besser mal von der Hersteller-Seite in neuerer Version heruntergeladen und nach Anleitung installiert werden. Die Version in Raspberrypi OS ist ziemlich uralte, kommt nicht mit Minus-Zeichen in Verzeichnisnamen zurecht und stellt sich sperrig an beim Hinzufügen externer Bibliotheken.

<sup>3</sup>Hierzu 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.