Skriptum zum

Raspberry Pi

Stand 24. Juni 2026



Raspberry Pi is a trademark of the Raspberry Pi Foundation

Inhaltsverzeichnis

1 Vorwort **5**

2 Installation des Systems **6**
2.1 Installation eines Abbilds . . . 6
2.1.1 Erstellen eines eigenen Abbilds . . . 7
2.2 Installation mit Noobs . . . 8

3 Bedienung, Linux Grundlagen **9**
3.1 Starten des Raspberry Pi . . . 9
3.1.1 Erste Schritte . . . 9
3.2 Bedienung in der Konsole . . . 10
3.2.1 Herunterfahren und Neustarten . . . 11
3.2.2 Befehle zu Dateien und Verzeichnisse . . . 12
3.2.3 Befehle zur Zugriffssteuerung auf Dateien und Verzeichnisse . . . 14
3.2.4 Befehle zur Systemverwaltung . . . 16
3.2.5 Nützliche Hilfsprogramme . . . 17
3.2.6 Software aktualisieren . . . 19
3.2.7 Software installieren . . . 19
3.2.8 Zusammenfassen von Befehlen in einem Skript . . . 20
3.2.9 Entfernter Zugriff über SSH (Secure Shell) . . . 20
3.2.10 Screenshots in der Konsole . . . 21
3.3 Bedienung in der graphischen Oberfläche . . . 21
3.3.1 WLAN und Bluetooth . . . 22
3.3.2 Browser . . . 23
3.3.3 E-Mail . . . 24
3.3.4 Screenshots . . . 27
3.4 Die Verzeichnisstruktur . . . 27
3.4.1 Dateinamen . . . 28
3.4.2 Strukturen im Benutzerverzeichnis … . . . 28

4 Python Programmierung **32**
4.1 Grundlagen in Python . . . 32
4.2 Programmierumgebung auf der Raspberry Pi . . . 32
4.2.1 Python in der Konsole . . . 34
4.3 Minecraft mit Python beeinflussen . . . 34
4.3.1 Grundlegendes . . . 35

4.3.2 Steine setzen . . . 35
4.3.3 Aktuelle Postion – Wo bin ich? . . . 36
4.3.4 Analyse der Umgebung . . . 36
4.3.5 Mehrspielermodus . . . 36
4.3.6 Weitere Informationen . . . 37

5 Elektrische Schaltungen 38
5.1 Schaltpläne und Schaltzeichen . . . 38
5.1.1 Verbindungen und Anschlüsse . . . 39
5.1.2 Schalter, Taster und Lampen . . . 39
5.1.3 Elektrische Bauteile . . . 39
5.2 Elektrische Widerstände . . . 40
5.2.1 Farbcodierung von Widerständen . . . 40
5.2.2 Widerstände in Schaltzeichnungen . . . 41
5.3 Einfache Elektrische Schaltung am GPIO . . . 42
5.3.1 Anschluss einer LED . . . 42
5.3.2 Erste Versuche auf dem Steckbrett . . . 42
5.3.3 GPIO Belegung . . . 43
5.3.4 Anschluss eines Schalters oder Tasters . . . 45
5.4 Ansteuerung des GPIO mit Python . . . 46
5.4.1 Einstellungen . . . 46
5.4.2 Ausgänge steuern . . . 46
5.4.3 Eingänge . . . 47
5.5 Pulsweitenmodulation . . . 47
5.6 Ansteuerung des GPIO mit Scratch . . . 48
5.6.1 Scratch 1.4 und der GPIO . . . 49
5.6.2 Scratch 2 und der GPIO . . . 51
5.7 Passiver Infrarotsensor (PIR) als Schalter . . . 52
5.8 Lichtschranke . . . 52
5.8.1 Abfrage im Programm . . . 53
5.9 Zweizeilendisplay . . . 53
5.10 Ansteuerung von Motoren . . . 54
5.11 Digitales Thermometer . . . 55
5.11.1 Treiber laden . . . 56
5.11.2 Temperatur auslesen . . . 56
5.11.3 Temperatur auslesen mit Python . . . 57
5.12 Analog/Digital-Wandler . . . 57
5.12.1 SPI einrichten . . . 57
5.12.2 Anschluss des A/D-Wandlers . . . 58
5.12.3 Einbindung in Python . . . 58
5.12.4 Anschluss eines Potentiometers . . . 59
5.12.5 Anschluss eines analogen Thermometers . . . 59

6 Software **61**
6.1 LaTeX auf dem Raspberry Pi 61
6.1.1 Installieren von LaTeX 61
6.1.2 Nutzung von LaTeX 61
6.1.3 Weitere Möglichkeiten 62
6.1.4 Installation direkt aus den TeXLive-Quellen 62
6.2 Verschlüsselung mit GnuPG 64
6.2.1 Installation und Einrichtung von GnuPG 65
6.2.2 Einrichtung von GnuPG in Claws Mail 68

A Alternative Bootmöglichkeiten **69**
A.1 Booten über USB 69

B Displays **71**
B.1 Installation des offiziellen Raspberry Pi Displays 71
B.2 Installation des Watterott-Displays 71

C Erweitertes WLAN **73**
C.1 WLAN mit WPA2-Personal einrichten 74
C.2 WLAN mit WPA2-Enterprise einrichten 74

D Windows **76**
D.1 Lesen und beschreiben der SD-Karte 76

Literatur **77**

Kapitel 1

Vorwort

Der Raspberry Pi wurde als Projekt gestartet, um an der Universität Cambridge der sinkenden Zahl von Informatikstudenten und deren immer geringer werden Programmierkenntnissen entgegen zu wirken. Durch seinen günstigen Preis sollte es geringer ins Gewicht fallen, wenn bei Experimenten mit der Hardware auch mal etwas schief geht. Dieser kleine Computer hat aber weltweit Begeisterte gefunden, so dass er seit dem Verkaufsstart 2012 bis Anfang 2015 bereits über fünf Millionen Exemplare verkauft wurden. Im September 2016 wurde der zehn Millionste Raspberry Pi gefeiert.

Der Name Raspberry Pi stammt zum einen von einer Tradition, Computer nach Früchten zu benennen. So ist hier die Himbeere gewählt worden, die auch stilisiert zum Logo wurde. Das Pi steht eigentlich für Python Interpreter. Python ist eine Programmiersprache, die ursprünglich direkt von diesem Computer ausgeführt werden sollte. Die erste ausgelieferte Version des Raspberry Pi hatte aber ein vollständiges Linux als Betriebssystem, so dass es auch viele andere Programmiersprachen auf dem Raspberry Pi zum Einsatz kommen können.

Da für viele Möglichkeiten des Raspberry Pi entsprechende Hilfsmittel in Python vorhanden sind, setzt dieses Skriptum darauf. Zuerst geht es aber um die Installation des Systems und seine grundlegende Bedienung. Da besonders unter Linux die Shell, auch als Eingabeaufforderung bekannt, viele Möglichkeiten eröffnet, gibt es auch eine kurze Einführung.

Der Hauptteil des Skriptums besteht aber aus Möglichkeiten und Projekten, die mit dem Raspberry Pi umgesetzt werden können. Dabei sind viele Elemente direkt aus Arbeits- bzw. Informationsblättern übernommen, die bereits zum Thema erstellt worden sind. Auch weiterführende Elemente und Grundlagen dafür, wie z. B. die Farbcodierung von elektrischen Widerständen, sind eingefügt. Da hier eine Vielzahl von Möglichkeiten geboten wird, empfiehlt es sich, eine passende Auswahl zu treffen.

Kapitel 2

Installation des Systems

Mittlerweile gibt es verschiedene Systeme, die auf einem Raspberry Pi laufen. Neben dem Standard Raspbian, einem Ableger des Debian Linux-Systems, gibt es unter anderem auch spezielle Versionen von Ubuntu, Windows und einem Mediacenter. Zum Installieren gibt es zwei verschiedene Wege: Entweder man nutzt mit Noobs ein Installationsprogramm, das direkt auf dem Raspberry Pi läuft oder man spielt ein passendes Abbild (Image) auf die SD-Karte.

2.1 Installation eines Abbilds

Um ein Abbild auf eine SD-Karte zu spielen benötigt man einen weiteren Rechner und ggf. zusätzliche Programme. Unter Linux lässt sich dieses am einfachsten mit dem Kommandozeilenprogramm dd machen, dass in der Regel schon installiert ist. Dazu muss man nur wissen, unter welchem Gerätenamen die SD-Karte im Rechner bekannt ist. Sollte die Karte nach dem Einbinden eingehängt (gemountet) worden sein, so hilft z. B. ein Aufruf von mount. Andernfalls kann man auch mit lsblk sich die Geräte des Rechners ausgeben:

```
johpie@joh:~$ lsblk
NAME                    MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda                       8:0    0 465,8G  0 disk
|-sda1                    8:1    0  69,9G  0 part
| `-md0                   9:0    0  69,9G  0 raid1 /
|-sda2                    8:2    0   4,7G  0 part
| `-cryptswap1 (dm-0)   252:0    0   4,7G  0 crypt [SWAP]
`-sda3                    8:3    0 391,2G  0 part
    `-md1                 9:1    0 391,1G  0 raid1 /extra
sdb                       8:16   0 465,8G  0 disk
|-sdb1                    8:17   0  69,9G  0 part
| `-md0                   9:0    0  69,9G  0 raid1 /
|-sdb2                    8:18   0   4,7G  0 part
`-sdb3                    8:19   0 391,2G  0 part
    `-md1                 9:1    0 391,1G  0 raid1 /extra
```

```
sdd          8:48   1    15G  0 disk
|-sdd1       8:49   1    56M  0 part
`-sdd2       8:50   1     3G  0 part
sr0         11:0    1  1024M  0 rom
```

Im diesem Fall ist es sdd mit einer Größe von 16 GB. In jedem Fall sollte dafür gesorgt werden, dass die Karte nicht im System gemountet ist. Der Aufruf erfolgt für die Datei raspberry-pi.img mit dem Aufruf:

```
sudo dd if=raspberry-pi.img of=/dev/sdd bs=1M
```

Anschließend muss man mehrere Minuten warten, bis man wieder die Möglichkeit zur Eingabe bekommt. In dieser Zeit wird das Image auf die SD-Karte geschrieben. In den meisten Fällen ist das Image kleiner als die benutzte SD-Karte. Daher ist es durch den Aufruf von raspi-config möglich, das Dateisystem auf die komplette SD-Karte zu vergrößern. Die Versionen von Raspbian ab Mai 2016 machen dieses automatisch beim ersten Booten.

2.1.1 Erstellen eines eigenen Abbilds

Um vor einigen Experimenten den derzeitigen Stand seinen Systems zu sichern, kann man den kompletten Inhalt der SD-Karte auf z. B. eine Festplatte kopieren. Dieses geht, wie auch die Installation eines Abbilds nicht mit dem Raspberry Pi direkt, sondern benötigt einen weiteren Computer. Auch hier hilft das Kommandozeilenprogramm dd. Analog wie beim Installieren eines Image wird hier davon ausgegangen, dass die Karte unter sdd liegt und eine Größe von 16 GB hat. Will man die komplette SD-Karte als Image haben, kann man sie mit folgendem Aufruf bekommen. Hier ist aber zu beachten, dass das Image die gleiche Größe hat, wie die SD-Karte, also im hier angegeben Fall 16 GB.

```
sudo dd if=/dev/sdd of=name_des_image_jj_mm_tt.img
```

Wenn man Abbilder verteilen will, damit z. B. mehrere Raspberry Pi den gleichen Stand bekommen, will man in der Regel nicht die komplette SD-Karte kopieren, wo der größte Teil des Dateisystems leer ist. Dazu kann man die Angaben von bs und count dem Programm dd mitteilen, entsprechend weniger zu kopieren. Dabei gibt bs die Größe der Blöcke an und count die Anzahl der zu kopierenden Blöcke. Im folgenden Fall werden somit nur 3,5 GB zum Image genommen:

```
sudo dd if=/dev/sdd of=name_des_image_jj_mm_tt.img bs=1M count=3500
```

Macht man dieses mit einer Partition, die bereits auf die komplette SD-Karte ausgeweitet wurde, so kann es zu Problemen führen. Deshalb sollte man die Partition und das entsprechende Dateisystem darauf wieder verkleinern. Einfacher ist es dafür zu sorgen,

dass das Dateisystem nicht auf die komplette SD-Karte ausgeweitet wird. Dazu muss die SD-Karte nach dem Aufspielen des Image und vor dem ersten Booten gemountet werden. In der Bootpartition findet man die Datei »cmdline.txt«, die die ersten Schritte beim Booten regelt. Hier muss `init=/usr/lib/raspi-config/init_resize.sh` am Ende des Aufrufs entfernt werden.

Es empfiehlt sich in jedem Fall das Datum mit in den Namen des Image mit aufzunehmen. So lässt sich direkt erkennen, wann dieses erzeugt wurde.

2.2 Installation mit Noobs

Die Installation mit Hilfe von Noobs ist sehr einfach. Von der Seite https://www.raspberrypi.org/downloads/ lädt man sich die passende Datei herunter. Dabei kann man zwischen zwei Versionen wählen: Es gibt einmal die große Version, die Raspbian direkt beinhaltet und alle anderen möglichen Systeme über das Netzwerk nachlädt. Die Lite-Version installiert alle Systeme über das Netzwerk. In beiden Fällen wird die Datei einfach mit gewohnten Mitteln an einem Computer auf die SD-Karte kopiert. Anschließend wird die SD-Karte in den Raspberry Pi gesteckt. Nach dem Starten des Raspberry Pi kann man eine Auswahl der oder des Systems treffen und die weitere Installation läuft automatisch ab.

Kapitel 3

Bedienung, Linux Grundlagen

In diesem Kapitel geht es um den Anschluss des Raspberry Pi, das erste Starten und Grundlegende Elemente zu seiner Bedienung. Dabei wird sowohl auf die auf die Bedienung in der Konsole, also der reinen Text Ein- und Ausgabe, wie auch die graphische Oberfläche eingegangen. Auch werden Grundlagen und der Aufbau der Verzeichnisstruktur erläutert.

3.1 Starten des Raspberry Pi

Bevor der Raspberry Pi gestartet werden kann, muss er erst mit der nötigen Hardware verbunden werden. Dabei ist zu beachten, dass der Anschluss an das Netzteil erst zum Schluss passiert, da der Raspberry Pi direkt versucht zu starten, sobald er mit Strom versorgt wird. Daher muss unbedingt vorher die SD-Karte in den passenden Anschluss unter dem Raspberry Pi platziert werden. Es empfiehlt sich auch, den Bildschirm über den HDMI-Port zuvor anzuschließen und das entsprechende Gerät so umzustellen, dass das Bildsignal des Raspberry Pi direkt verarbeitet wird. Andernfalls wird der Raspberry Pi die Ausgabe über den Videoausgang starten, wenn nicht andere Einstellungen getroffen worden sind.

3.1.1 Erste Schritte

Je nach Einstellung des Systems wird beim Hochfahren des Raspberry Pi entweder eine graphische Oberfläche gestartet oder man landet einer reinen Textumgebung, der Konsole. Werden Zugangsdaten benötigt, so sind diese im Ursprungszustand »pi« für den Benutzer und »raspberry« als Passwort. Bei der Eingabe muss man ggf. beachten, dass noch eine englische Tastatur eingestellt ist, bei der im Vergleich zur deutschen Tastatur das z und das y vertauscht sind.

Die erste Tätigkeiten bei einem neuen System sollten darin bestehen, das Dateisystem auf die volle Größe der SD-Karte auszuweiten sowie das Passwort zu ändern und die Länder-/Spracheinstellungen anzupassen. In der Konsole geschieht dieses über den Aufruf von

```
sudo raspi-config
```

bei dem man sich anschließend durch ein Textmenü durcharbeiten muss. Unbekannte Optionen sollten in diesem Fall nicht geändert werden. In der graphischen Oberfläche gibt es auch einen passenden Eintrag im Menü unter Einstellungen.

3.2 Bedienung in der Konsole

Die Konsole bzw. das Terminal ermöglicht es, Programme und Befehle direkt über die Tastatur an das System abzugeben. Durch viele kleine Hilfsmittel können so komplexe Aufgaben sehr einfach bewältigt werden. Dafür benötigt man aber die Kenntnisse über die wichtigen Befehle und Möglichkeiten.

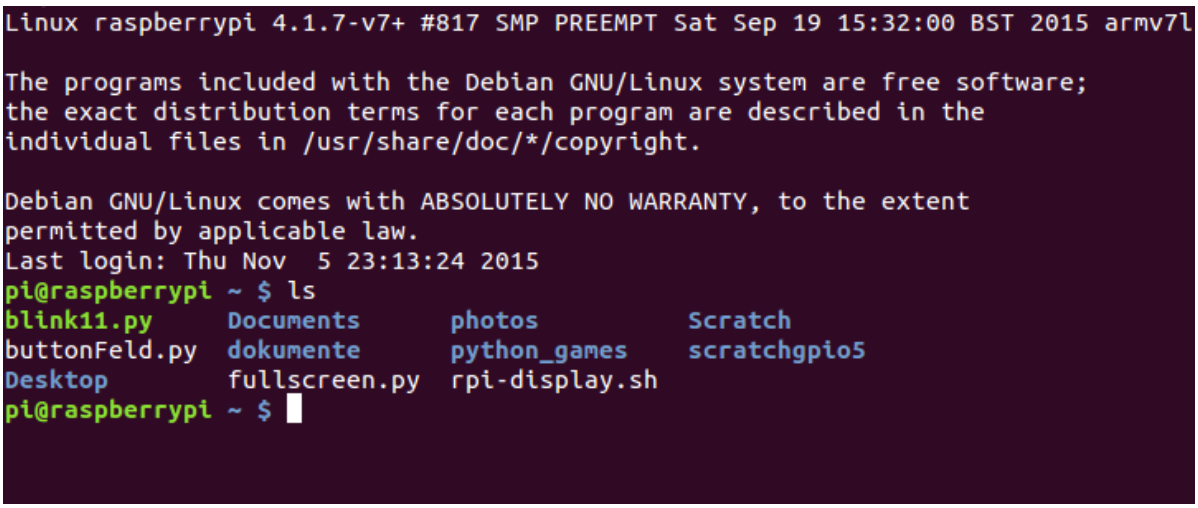


Abbildung 3.1: Konsole auf dem Raspberry Pi

Erreichen kann man eine Konsole auf dem Raspberry Pi entweder dadurch, dass sie direkt gestartet wird[1] oder durch den Aufruf mit Hilfe von Strg+Alt+F1 bis F6[2]. Es lässt sich aber auch in der graphischen Oberfläche mit Hilfe von »lxterminal« eine Konsole starten. Einige der wichtigsten Befehle in der Konsole auf einem Linux-System sind in der Tabelle 3.1 aufgeführt.

Kennt man nur den Anfang eines Befehls, so kann man sich durch Drücken der »Tab-Taste« entsprechende Vorschläge ergänzen lassen. Sie kann aber auch dazu genutzt werden, um bei der Eingabe von z. B. Pfadangaben den Rest des Dateinamens zu ergänzen.

[1] Sie ist die Alternative zum direkten Starten der graphischen Oberfläche.
[2] Sollte die graphische Oberfläche gestartet sein, kann man in sie mit Strg+Alt+F7 zurückkehren

Befehl	Bedeutung
ls	Zeigt den Inhalt eines Verzeichnisses an.
cd	Wechselt das Verzeichnis, wobei der Wechselwunsch als Parameter angegeben werden muss. Ein Verzeichnis hoch gelangt man durch die Angabe von »..« und mit »/« gelangt man ins Hauptverzeichnis bzw. Wurzelebene genannt.
man	Zeigt eine Hilfe zu dem angegebenen Befehl an.
cp	Kopiert Dateien.
mv	Verschiebt Dateien oder Verzeichnisse.
rm	Löscht Dateien.
sudo	Führt die direkt mit angegebenen Befehle als root-User[3] aus.
mkdir	Erstellt ein Verzeichnis.
chmod	Ändert die Rechte an Verzeichnissen oder Dateien.
chown	Ändert den Besitzer von Verzeichnissen oder Dateien.
grep	Sucht nach Vorkommen einer Zeichenkette in einer Datei.
cat	Gibt den Inhalt einer Datei an.
passwd	Ändert Passwörter

Tabelle 3.1: Kleine Übersicht über wichtige Befehle



Aufgabe 3.1

Nutze die oben genannten Befehle und nutze auch die Hilfe »man« um mehr über sie zu erfahren. Fertige in einer Textdatei auf dem Raspberry Pi eine kleine Übersicht zu den oben genannten Befehlen an. Diese sollte neben dem Namen und einer Beschreibung, die teilweise etwas ausführlicher ist als in der oberen Liste, auch Beispiele für die Anwendung mit genauem Aufruf enthalten.

In den folgenden Abschnitten werden einige dieser Befehle ausführlich beschrieben.

3.2.1 Herunterfahren und Neustarten

Möchte man den Raspberry Pi herunterfahren, so geht dieses über die entsprechenden Menüeinträge in der graphischen Oberfläche oder über

```
sudo halt -p
```

in der Konsole. Der entsprechende Aufruf zum Neustarten wäre:

```
sudo reboot
```

Wenn der Raspberry Pi heruntergefahren wurde, so kann man dieses sehen, dass nur noch die Statusleuchte an ist. Danach kann die Stromversorgung vom Raspberry Pi getrennt werden, damit der komplett aus ist.

3.2.2 Befehle zu Dateien und Verzeichnisse

Alle der folgenden Befehle können auf eigentlich alle Dateien bzw. Verzeichnisse angewandt werden, solange der Benutzer nicht durch fehlende Rechte eingeschränkt wird. Sie sind dabei unabhängig vom Typ der Dateien, wie z. B. Text oder Bilddateien.

Verzeichnis wechseln mit cd

Zum wechseln des Verzeichnisses wird der Befehl cd genutzt, der für »change directory« steht. Das gewünschte Verzeichnis wird dabei relativ, also vom aktuellen Verzeichnis aus, angegeben oder absolut mit dem kompletten Pfad. Eine absolute Pfadangabe ist immer daran zu erkennen, dass sie mit einem / beginnt. In fast allen Kommandozeilen wird nach dem Benutzer und dem Rechnername auch das aktuelle Verzeichnis mit kompletten Pfad vor dem Promt $ angezeigt:

```
pi@raspberrypi:/usr/share $
```

Befindet man sich aber im Homeverzeichnis bzw. dessen Unterverzeichnisse, so wird dieses mit einer Tilde ~ abgekürzt:

```
pi@raspberrypi:~/Documents $
```

cd foo	Wechselt in das Unterverzeichnis foo des aktuellen Verzeichnisses.
cd foo/bar	Wie oben, nur dass in das Unterunterverzeichnis gewechselt wird.
cd ..	Wechselt in das Verzeichnis eine Ebene höher vom aktuellen Verzeichnis aus.
cd ../../ foo	Wechselt zwei Ebenen höher und von dort in das dortige Unterverzeichnis foo.
cd	Es wird in das Homeverzeichnis des aktuellen Benutzers gewechselt.
cd ~	Analog zur Eingabe cd.
cd ~/Bilder	Wechselt direkt in das Unterverzeichnis Bilder des Homeverzeichnis gewechselt.
cd ~foo	Wechselt in das Homeverzeichnis des Benutzers foo.
cd /etc	Wechselt in das Verzeichnis /etc mit dem absoluten Pfad.

Verzeichnisinhalt anzeigen mit ls

Durch den Befehl ls wird der Inhalt des aktuellen oder angegebenen Verzeichnis ausgegeben. Hinter dieser Abkürzung steht der Begriff »list«. Durch entsprechende Parameter lässt sich die Ausgabe den eigenen Wünschen hin anpassen, so dass nur bestimmte Dateien angezeigt werden oder auch weitere Informationen als die Dateinamen.

ls	Die einfache Form liefert die Dateien und Verzeichnisse in mehreren Spalten nebeneinander.
ls -l	Die Dateien werden untereinander mit Informationen zu Besitzer, Gruppe, Rechten, Größe und Änderungsdatum ausgegeben.
ls -a	Gibt auch versteckte Dateien und Verzeichnisse mit aus.
ll	Ist die Kurzform für ls -la
ls -lh	Die Dateigrößen werden statt in Byte in einer passenden Größe wie Kilobyte oder Megabyte angegeben.
ls *.txt	Nur Dateien mit der Endung txt werden ausgegeben.
ls foo/	Auflistung des Inhalts des Unterverzeichnisses foo.

Kopieren mit cp

Der Befehle cp, kurz für »copy«, kopiert eine Datei an einen anderen Ort oder mehrere Dateien in ein Verzeichnis.

cp foo.bar foo2.bar	Kopiert die Datei mit dem Namen foo.bar, zu einer Datei mit dem Namen foo2.bar oder in ein Verzeichnis mit dem Namen foo2.bar, wenn es existiert.
cp foo.bar ../	Kopiert die Datei mit dem Namen foo.bar in das Verzeichnis eine Ebene höher.
cp *.txt neu	Kopiert alle Dateien, die auf .txt enden in das Unterverzeichnis neu.
cp al* neu	Funktioniert analog zum oberen, kopiert aber alle Dateien, die mit al beginnen.
cp * neu	Kopiert alle Dateien im aktuellen Verzeichnis in das Unterverzeichnis neu.
cp bar/* .	Kopiert alles aus dem Unterverzeichnis bar in das aktuelle Verzeichnis, was durch den Punkt angegeben ist.
cp -r foo bar	Kopiert das Verzeichnis foo mit allen Dateien und Unterverzeichnissen in ein Verzeichnis bar, bzw. dort hinein, falls bar bereits existiert.
cp -r foo/* bar	Kopiert den gesamten Inhalt mit ggf. Unterverzeichnissen in das Verzeichnis bar

Verschieben und umbennen mit mv

Sehr ähnlich zum Kopieren lassen sich mit dem Befehl mv Dateien bzw. Verzeichnisse umbenennen oder verschieben. Dabei ist mv die Kurzform von »move«.

mv foo bar	Ändert den Namen der Datei bzw. des Verzeichnisses foo in bar.
mv *.txt bar	Verschiebt alle Dateien, die auf .txt enden, in das Verzeichnis bar.
mv ../bar .	Verschiebt aus dem Überverzeichnis das Verzeichnis bar in das aktuelle Verzeichnis, was durch den Punkt angeben ist.

Verzeichnisse erstellen mit mkdir

Um neue Verzeichnisse erstellen zu können, wird der Befehl mkdir genutzt, mit der Angabe des neuen Verzeichnisnamen. Die Abkürzung »mkdir« steht dabei für den Ausdruck »make directory«.

mkdir foo	Erstellt das Unterverzeichnis foo im aktuellen Verzeichnis.
mkdir -p bar/foo	Im aktuellen Unterverzeichnis wird der Pfad bar/foo erstellt. Durch den Parameter -p werden auch alle Verzeichnisse im Pfad erstellt, die bisher noch nicht existieren.
mkdir ~/bar	Erstellt, unabhängig vom aktuellen Pfad, das Verzeichnis bar im Homeverzeichnis.

Verzeichnisse löschen mit rmdir

Das Gegenteil von mkdir ist rmdir. Dabei muss beachtet werden, dass das Verzeichnis auch leer ist.

rmdir foo	Entfernt das leere Verzeichnis foo im aktuellen Verzeichnis.
rmdir -p bar/foo	Entfernt die Verzeichnisse bar/foo und bar im aktuellen Verzeichnis, wenn beide leer sind.

Dateien löschen mit rm

Zum Löschen von Dateien wird rm genutzt. Dieses kann dabei auch genutzt werden, um Verzeichnisse inklusive ihrem Inhalt zu entfernen.

rm bar	Entfernt die Datei bar im aktuellen Verzeichnis. Handelt es sich bei bar um ein Verzeichnis, so wird eine Fehlermeldung ausgegeben.
rm foo bar	Entfernt die Dateien foo und bar.
rm *.tmp	Entfernt alle Dateien, die auf .tmp enden.
rm -r foo	Entfernt das Verzeichnis foo mit allen Dateien und Unterverzeichnissen.

3.2.3 Befehle zur Zugriffssteuerung auf Dateien und Verzeichnisse

Jeder Datei und jedem Verzeichnis sind auf einem Linuxsystem jeweils eine Benutzer und eine Gruppe als Besitzer zugeordnet. Außerdem gibt es verschiedene Rechte, die ein Benutzer als Besitzer, Mitglied der Besitzergruppe oder als sonstiger Benutzer haben kann. Dazu gehören die Rechte etwas zu verändern (w für write), etwas zu lesen (r für read) oder eine Datei auszuführen bzw. in ein Verzeichnis zu wechseln (x für execute). Diese Zugriffsregeln für eine Datei bzw. Verzeichnis lassen sich durch drei verschiedene Befehle ändern. Dabei sind, je nach Änderung auch administrative Rechte nötig.

Ändern des Besitzers mit chown

Mit chown (change owner) lässt sich der Besitzer von Dateien oder Verzeichnissen ändern. Er kann aber auch dazu genutzt werden Besitzer und Gruppe gleichzeitig zu ändern.

chown pi foo.bar	Ändert den Besitzer der Datei foo.bar auf den Benutzer pi.
chown pi:user foo.bar	Ändert von der Datei foo.bar den Benutzer auf pi und die Gruppe auf user.
chown pi *	Ändert den Besitzer alle Dateien im aktuellen Verzeichnis.
chown -R pi .	Ändert den Besitzer des aktuellen Verzeichnisses.
chown -R pi *	Ändert den Besitzer alle Dateien im aktuellen Verzeichnis und allen Unterverzeichnissen.

Ändern der Gruppe von Dateien mit chgrp

Analog zum Ändern des Besitzers lässt sich mit chgrp (change group) die Besitzergruppe von Dateien oder Verzeichnissen ändern.

chgrp user foo.bar	Ändert die Gruppe der Datei foo.bar auf die Gruppe user.
chgrp user *	Ändert die Gruppe alle Dateien im aktuellen Verzeichnis.
chgrp -R user .	Ändert die Gruppe des aktuellen Verzeichnisses.
chgrp -R user *	Ändert die Gruppe alle Dateien im aktuellen Verzeichnis und allen Unterverzeichnissen.

Rechte ändern durch chmod

Bei den Rechten für Dateien und Verzeichnissen wird in erster Linie zwischen schreiben (w), lesen(r) und ausführen(x) unterschieden. Dabei können diese Rechte jeweils für den Benutzer (u wie user), die Gruppe (g wie group) und alle anderen (o wie other) gesetzt werden. Dieses geschieht mit Hilfe von chmod, dass für »change mode« steht. Dabei wird angegeben, ob ein Recht hinzugefügt wird (+) oder entfernt (-). Neben der Angabe der symbolischen Darstellung kann auch die oktale Darstellung genutzt werden, die hier nicht näher erläutert wird.

chmod g+w *.txt	Setzt die Rechte zum Schreiben aller txt Dateien für die Gruppe.
chmod u+x *.py	Alle py Dateien können vom Besitzer anschließend ausgeführt werden.
chmod o-r *.log	Entfernt die Rechte zum Lesen aller log Dateien für die alle anderen Benutzer.
chmod +w foo.bar	Für alle drei (Benutzer, Gruppe und Andere) wird das Recht zum Schreiben der Datei foo.bar gesetzt.
chmod -R u+w .	Im aktuellen Verzeichnis und allen Unterverzeichnissen bekommt der Besitzer das Recht zum schreiben.

3.2.4 Befehle zur Systemverwaltung

Das System hat auch verschiedene Befehle, die zur Verwaltung benötigt werden. Immer dann, wenn sie Änderungen am System vornehmen, müssen sie dann mit administrativen Rechten, also den sogenannten Root-Rechten ausgeführt werden.

Befehle ausführen mit Root-Rechten durch sudo

Wenn man als Benutzer berechtigt ist, so kann man mit Hilfe von sudo, das für »super user do« steht, Befehle mit root-Rechten ausführen. Dazu muss nur der Befehl, den man ausführen will, direkt hinter sudo angegeben werden. Die Berechtigung erhält man durch einen Eintrag in der Datei /etc/sudoers, die man aber aus Sicherheitsgründen nur mit visudo bearbeiten sollte. Anders kann es sein, dass man alle Benutzer des Systems von den administrativen Rechten aussperrt.

sudo rm foo.bar	Entfernt die Datei foo.bar auch wenn der eigentliche Benutzer dafür keine Rechte hätte.
sudo apt update	Aktualisiert die Paketliste des Systems, für das man administrative Rechte benötigt.
sudo su -	Wechselt in eine Befehlskonsole, in der man sich mit administrativen Rechten bewegt.

Informationen zum System mit uname

Mit Hilfe von uname lassen sich Informationen, wie die Version des aktuell genutzt Kernels ausgeben oder der Name der Maschine ausgeben.

uname	Gibt an, was allgemein für ein Kernel verwendet wird.
uname -n	Gibt den Namen der Maschine aus.
uname -r	Gibt die Release-Nummer des aktuellen Kernels aus.
uname -a	Gibt alle Informationen zum System inklusive der oben genannten aus.

Zeit auslesen und setzen mit date

Durch den Befehle date hat man die Möglichkeit die aktuelle Uhrzeit des Systems sich angeben zu lassen. Es ist damit aber auch möglich, die Uhrzeit von Hand zu setzen. Dieses kann beim Raspberry Pi des öfteren der Fall sein, da dieser über keine Hardwareuhr verfügt und daher ohne Strom die Uhrzeit nicht weiterläuft.

date	Gibt das aktuelle Systemdatum und Uhrzeit aus.
date -s ”2017-12-24 18:30”	Setzt die Uhrzeit des Systems auf das angegebene Datum mit der entsprechenden Uhrzeit. Dieser Vorgang benötigt Root-Rechte voraus.

3.2.5 Nützliche Hilfsprogramme

Neben den Befehlen in der Konsole, die direkte Auswirkungen auf das Dateisystem oder das System allgemein haben, gibt es eine Reihe von weiteren nützlichen Programmen, die in der Konsole genutzt werden können. Dazu gehören neben Texteditoren auch Möglichkeiten zu suchen und den Inhalt von Dateien auszugeben.

Texteditoren

Es gibt viele verschiedene Texteditoren für die Befehlskonsole. Am häufigsten werden dabei nano und vi genutzt. Beide Editoren können entweder direkt aufgerufen werden oder mit Angabe der Datei, die bearbeitet werden soll.

nano	Startet den Texteditor mit einem leeren Text.
nano foo.bar	Die Datei foo.bar wird zum Bearbeiten geöffnet.
vi	Öffnet den Texteditor mit einem leeren Text.
vi foo.bar	Der Texteditor öffnet die Datei foo.bar.

Bei der Bedienung ähnelt nano vielen bekannten graphischen Texteditoren. Die Tastaturcodes für Möglichkeiten wie Speichern, Öffnen und Beenden werden in den unteren Zeilen angegeben. Zu den dort angegebenen Tasten muss jeweils die Strg-Taste gedrückt gehalten werden, wie z. B. Strg+x zum Beenden.

Beim vi wird zwischen einem Befehlsmodus und einem Eingabemodus unterschieden. Der Editor öffnet im Befehlsmodus, indem verschiedene Befehle zum Öffnen, Speichern usw. eingegeben werden können. Möchte man in den Eingabemodus wechseln um am Text etwas zu ändern, so muss man entweder die Einfg.-Taste oder die Taste i drücken. Aus dem Eingabemodus gelangt man wieder in den Befehlsmodus durch das Drücken der ESC-Taste. Nachfolgend sind die wichtigsten Befehle für den vi aufgeführt:

:w	Speichert die Datei.
:q	Beendet den vi.
:q!	Der Editor wird ohne Speichern beendet, auch wenn Änderungen am Text vorhanden sind.
:wq	Speichert die Datei und beendet den Editor.
:12	Springt mit dem Cursor in Zeile 12.
/foo	Sucht nach foo im Text und springt mit dem Cursor zum nächsten Vorkommen.
u	Macht die letzte Änderung rückgängig.

Mit passwd das Passwort ändern

Zum Ändern des Passworts dient den fast gleichlautenden Befehl passwd. Bei diesem ist zu beachten, dass bei der Eingabe des Passworts eine Ausgabe auf der Konsole, also auch keine Sternchen, zu sehen ist. Mit entsprechenden Rechten lassen sich darüber auch

die Passwörter von anderen Benutzern ändern. Die Eingabe des neuen Passworts muss automatisch wiederholt werden, um mögliches Vertippen zu vermeiden.

passwd	Ändert das Passwort für den angemeldeten Benutzer.
passwd alice	Mit administrativen Rechten wird das Passwort des Benutzers alice geändert.

Suchen und Durchsuchen von Dateien mit Hilfe von find und grep

Eine Möglichkeit nach Dateien aufgrund verschiedener Kriterien suchen bietet find. Diese Kriterien können z. B. der Name oder das Datum der letzten Änderung sein. In jedem Fall muss als erstes der Pfad angegeben werden, unterhalb dessen gesucht werden soll. Dabei symbolisiert der Punkt das aktuelle Verzeichnis.

find .	Liefert alle Dateien im aktuellen Verzeichnis und dessen Unterverzeichnissen.
find . -name ”foo*”	Sucht alle Dateien, die mit foo beginnen im aktuellen Verzeichnis.
find bar -name ”foo*”	Im Verzeichnis bar wird nach entsprechenden Dateien gesucht.
find . -name ”foo*” -or -name ”bar*”	Listet alle Dateien die mit foo oder bar beginnen.
find . -name ”foo*” -and -name ”*bar”	Dateien die mit foo beginnen und bar enden werden geuscht.
find . -cmin 30	Alle Dateien, die in den letzten 30 Minuten geändert wurden werden gesucht.

Sucht man stattdessen nach einer Datei aufgrund ihres Inhalts oder nach einer besonderen Stelle innerhalb einer Datei, so sorgt grep für Abhilfe. Damit kann man nach einer Zeichenkette innerhalb einer Datei oder mehreren Dateien suchen.

grep foo bar.txt	Gibt alle Vorkommnisse von foo in der Datei bar.txt aus.
grep foo *	Es wird nach foo in allen Dateien im Verzeichnis gesucht.
grep -n foo *	Bei der Stelle in den Dateien wird auch die Zeilennummer mit angegeben.
grep -r foo *	Es werden auch Dateien in Unterverzeichnissen mit in die Suche einbezogen.

Ausgeben einer Datei mit cat und tail

Möchte man den Inhalt einer Datei in der Befehlskonsole ausgeben, so ist cat das Mittel der Wahl. Besonders nützlich ist es, wenn die Datei auch von anderen Programmen geöffnet ist oder die Ausgabe weiterverwendet werden soll.

`cat foo.bar`	Gibt den Inhalt der Datei foo.bar aus.
`cat foo.bar foo2.bar`	Hängt die beiden Dateien aneinander bei der Augabe.

Wenn man z. B. eine Log-Datei betrachten möchte, in der die Ausgaben zum System oder anderen Abläufen geschrieben werden, dann ist `tail` sehr hilfreich. Es gibt dann die letzten zehn Zeilen der Datei aus. Es lässt sich aber auch dazu nutzen, dass immer die neusten Zeilen ausgegeben werden, die bei der Datei an das Ende angehangen werden.

`tail foo.bar`	Gibt die letzten zehn Zeilen von foo.bar aus.
`tail -n 15 foo.bar`	Es werden die letzten fünfzehn Zeilen ausgegeben.
`tail -f foo.bar`	Nachdem die ersten zehn Zeilen ausgeben werden wird diese Ausgabe immer dann ergänzt, wenn an die Datei weitere Zeilen angehangen werden.

3.2.6 Software aktualisieren

Fast die komplette Software, die auf dem Raspberry Pi installiert ist, wird in Paketen verwaltet. Das Programm apt-get für die Konsole bietet eine Möglichkeit diese Pakete zu verwalten. Da diese Pakete von einer zentralen Stelle angeboten werden, ist es auch damit sehr einfach möglich, das komplette System auf den neusten Stand zu bringen:

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

Da Änderungen am System nur vom Administrator durchgeführt werden dürfen, muss apt-get mit dem Befehl sudo aufgerufen werden. So arbeitet man mit den Rechten eines Administrators. Der erste Befehl aktualisiert die Liste der Pakete. Mit dem zweiten Befehl werden alle Pakete, bei denen eine neuere Version vorhanden ist aktualisiert.



Aufgabe 3.2

Versetze die Software deines Raspberry auf den aktuellsten Stand.

3.2.7 Software installieren

Mit dem Paketmanager lassen sich auch zusätzliche Programme installieren oder deinstallieren. Dazu muss man nur den Namen des Paketes kennen. Mit Hilfe von apt-cache search <Suchbegriff> kann man nach einem entsprechenden Paket suchen. Dieses wird dann mit sudo apt-get install <Paketname> installiert. Auf gleichem Wege kann man auch Pakete mit sudo apt-get remove <Paketname> wieder entfernen.



Aufgabe 3.3

Installiere den Texteditor »joe« und das Paket »subversion«.

3.2.8 Zusammenfassen von Befehlen in einem Skript

Mit Hilfe eines Bash-Skripts mit es möglich mehrere Befehle, die man sonst in der Konsole hintereinander eingegeben hätte, zu einem zusammen zufassen. Das Skript wird in eine einfache Text-Datei geschrieben, deren erste Zeile lautet: #!/bin/bash. Danach folgen die gewünschten Befehle. Außerdem sollte diese Datei das Recht haben, ausgeführt zu werden (+x). Dieses kann durch chmod +x <dateiname> erreicht werden, so dass anschließend das Skript mit Hilfe von .\<dateiname> ausgeführt werden kann.

Die Stärke der Bash-Skripte ergibt sich daraus, dass in ihnen Variablen, Methoden, Wiederholungsstrukturen und bedingte Anweisungen definiert werden können. Bash-Skripte können deshalb auch als Programmiersprache bezeichnet werden.

3.2.9 Entfernter Zugriff über SSH (Secure Shell)

Viele Raspberry Pi werden an Stellen genutzt, an denen sich kein Monitor, Display oder auch Tastatur anschließen lässt. Auch in solchen Fällen gibt es Situationen, in denen man auf den Rechner zugreifen will. Dazu gibt es die Möglichkeit auf dem Raspberry Pi einen SSH-Server laufen zu lassen. Dieses lässt sich über das Konfigurationsprogramm (raspi-config) einstellen.

Um den Raspberry Pi erreichen zu können, muss dieser mit einem Netzwerk, über Kabel oder WLAN, verbunden sein. Sinnvoll ist es auch, wenn man z. B. über ifconfig die IP-Adresse des Raspberry Pi kennt. Dann kann man unter Linux sich mit ssh pi@<ip-adrese> sich mit dem Gerät verbinden und erhält eine Eingabeaufforderung, wie man sie auch direkt auf dem Raspberry Pi hat. Unter Windows muss man andere Programme wie z. B. Putty nutzen, um sich mit dem Raspberry Pi verbinden zu können.

Kopieren von Dateien über SSH

Neben der entfernten Eingabeaufforderung erlaubt es SSH auch, Dateien auf diesem Wege zu übertragen. Unter Linux steht einem dazu scp zur Verfügung. Bei der Syntax dieses Programms sind aber ein paar Dinge zu beachten: Der Aufbau des Aufrufs ist scp <quelle1> [<quelle2> ...] <ziel> und damit ähnlich dem des normalen Kopieren auf einem Dateisystems. Die lokale Datei kann auch genauso wie beim cp angegeben werden. Die entfernte Datei wird mit Hilfe von <user>@<adresse>:<pfad> angegeben. Mögliche Aufrufe können dann so aussehen:

```
scp muster.txt muster.jpg pi@192.168.4.24:/home/pi/muster/
scp pi@192.168.4.24:foo.bar .
```

Beim ersten Aufruf werden die Dateien muster.txt und muster.jpg auf einen Raspberry Pi kopiert und beim zweiten die Datei foo.bar aus dem Homeverzeichnis in das aktuelle lokale Verzeichnis.

3.2.10 Screenshots in der Konsole

Um Bildschirmphotos in einen der sechs Konsolen anfertigen zu können, kann das Programm fbgrap genutzt werden, dass zuvor durch das Paket fbcat installiert werden muss. An ein einfaches Foto gelangt mit Hilfe des Aufrufs:

```
fbgrap bildname.png
```

Sehr nützlich erweist sich das Programm darin, dass es auch Bilder der anderen Konsolen anfertigen kann. Auch eine Verzögerung lässt sich einstellen. So wird bei folgender Eingabe nach 7 Sekunden auf die Konsole gewechselt, die unter »Strg+Alt+F2« zu erreichen ist. Anschließend wird von dort das Bildschirmphoto angefertigt.

```
fbgrap -C 2 -s 7 bildname.png
```

3.3 Bedienung in der graphischen Oberfläche

Die Bedienung der graphischen Oberfläche verhält sich sehr ähnlich, wie es von anderen Systemen bekannt ist. Auch wenn sich das Aussehen der Oberfläche mit den verschiedenen Versionen leicht verändert hat, sind die Grundstrukturen gleich geblieben.

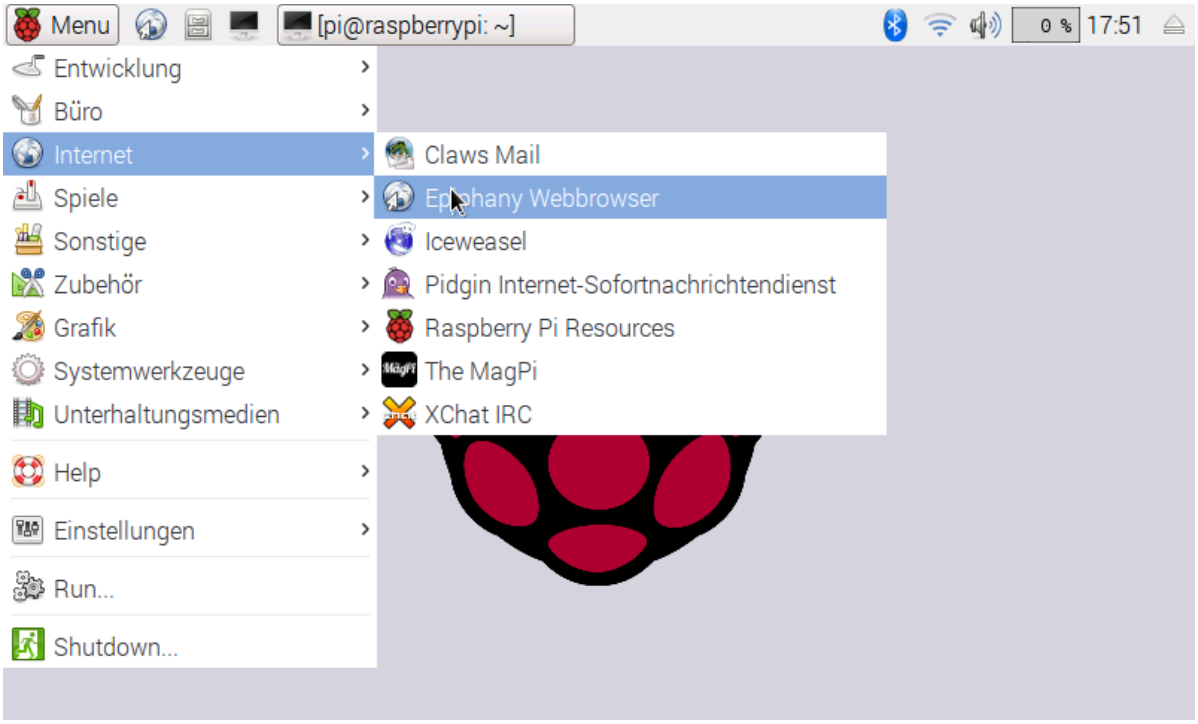


Abbildung 3.2: Desktop

Die meisten installierten Programme lassen sich über das Menü erreichen, dass sich in der linken oberen Ecke befindet. Ist der Name des Programms bekannt, kann es auch direkt aufgerufen werden. Das dafür nötige Dialogfenster lässt sich mit Alt+F2 öffnen. Der Aufruf von lxterminal in diesem Dialog sorgt dafür, dass man eine Konsole geöffnet bekommt.

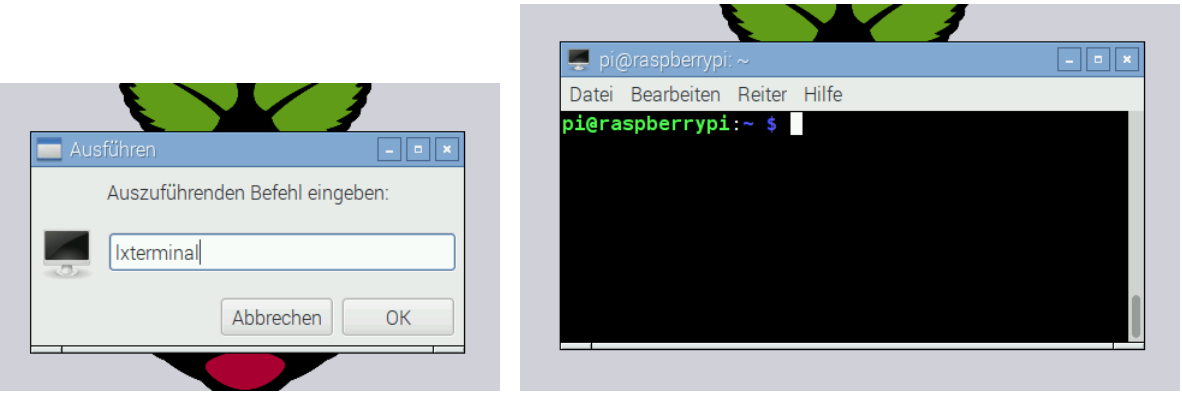


Abbildung 3.3: Konsole auf dem Desktop

3.3.1 WLAN und Bluetooth

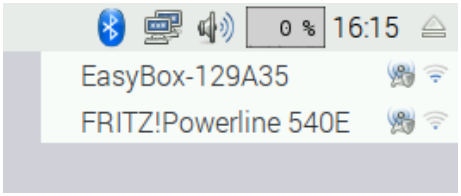


Abbildung 3.4: WLAN-Auswahl

In der graphischen Oberfläche sind auch standardmäßig graphische Steuerungselemente für WLAN und Bluetooth integriert, die nur eine sehr einfache Konfiguration zulassen. Das WLAN findet sich unter dem Netzwerksymbol. Mit einem Klick darauf können, sofern ein WLAN-Adapter erkannt wurde, werden die verfügbaren Netze aufgelistete. Wählt man ein verschlüsseltes davon aus, kann man direkt das Passwort dafür eingeben. Ist die Einwahl erfolgreich so wechselt das Symbol mit Anzeige der Empfangsqualität. Bei WLAN-Netzwerken, die eine besondere Anmeldung benötigen, kommt man an dieser Stelle aber nicht weiter. Im Anhang C ist dafür eine genauere Anleitung zu finden. Diese kann auch nützlich sein, wenn des öfteren das WLAN gewechselt wird und man nicht jedes mal das entsprechende Passwort wieder eingeben will.

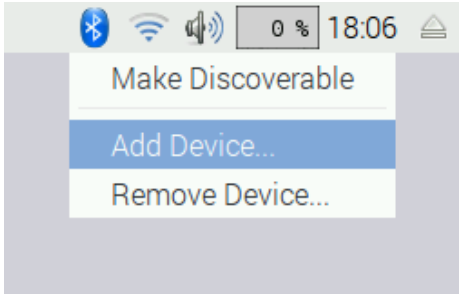


Abbildung 3.5: Bluetooth Menu

Der Raspberry Pi 3B ist das erste Modell mit einen Bluetooth-Adapter und hat damit die Möglichkeit, schnell eine Verbindung mit anderen Bluetooth-Geräten aufbauen zu können, um diese zu nutzen. Bei den vorherigen Modellen muss ein passender Adapter z. B. über USB angeschlossen werden.

Der Aufbau der Verbindung erfolgt wie bei anderen Geräten auch, indem man nach entsprechen-

den Geräten sucht. Dieses geht über das Menü, dass sich beim Klicken auf das entsprechende Icon öffnet. Das gewünschte Gerät muss dann ausgewählt werden und ggf. durch einen Code verifiziert werden.

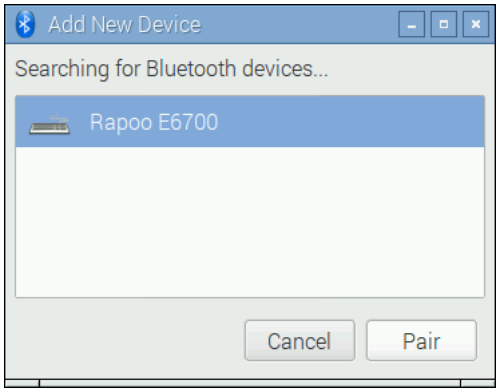


Abbildung 3.6: Hinzufügen eines Bluetooth-Geräts

3.3.2 Browser

Standardmäßig ist auf dem Raspberry der Browser Epiphany installiert. Es handelt sich dabei um einen sehr einfachen und schlanken Browser. In einigen Fällen fehlt dem Browser aber bekannte Eigenschaften, so dass einige Seiten nur beschränkt genutzt werden können.

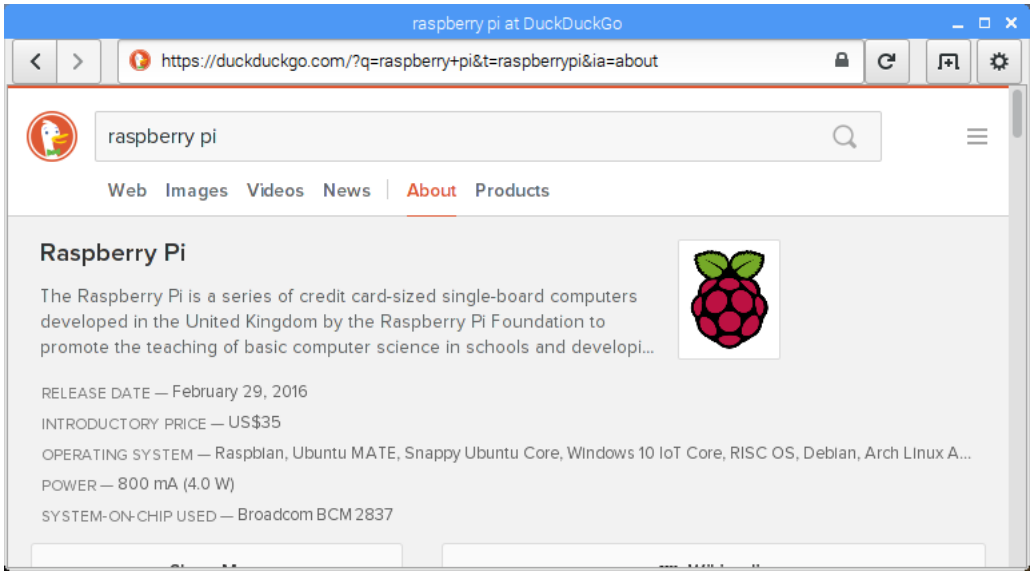


Abbildung 3.7: Standardbrowser auf dem Raspberry Pi

Alternativ kann man sich auch den bekannten Browser Firefox installieren. Dieses liegt im Paket firefox-esr und das Paket firefox-esr-l10n-de beinhaltet die deutschen Sprachelemente. Die Installation von Paketen wird in Abschnitt 3.2.7 erklärt.

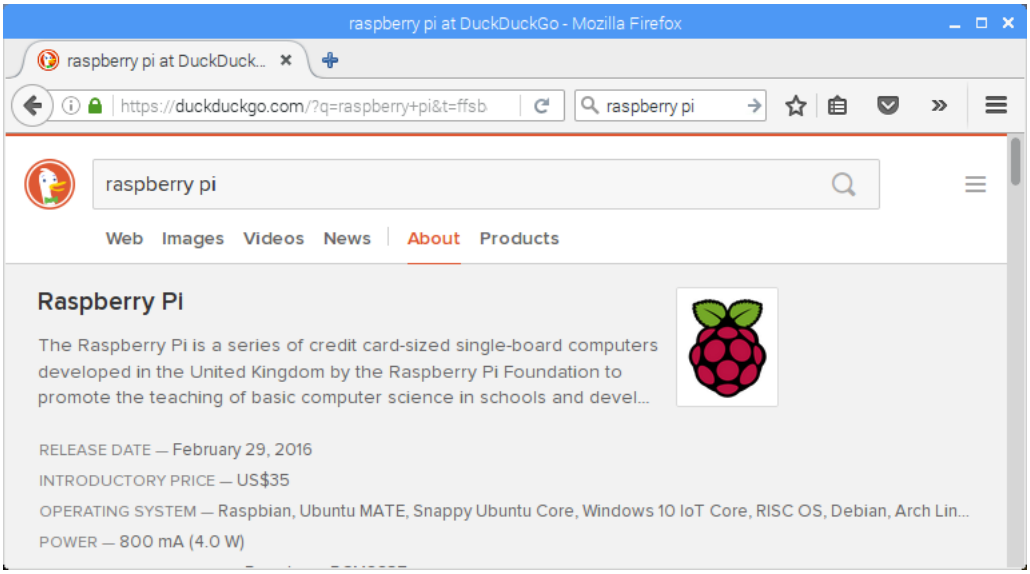


Abbildung 3.8: Firefox auf dem Raspberry Pi

3.3.3 E-Mail

Mit Claws Mail ist auf dem Raspberry Pi direkt ein einfacher E-Mail-Client installiert. Mit ihm ist es möglich, nach entsprechender Einrichtung, E-Mails mit dem Raspberry Pi zu empfangen und senden, ohne dabei einen Webclient im Browser nutzen zu müssen. Claws Mail arbeitet dabei wesentlich schneller und komfortabler, als ein Webclient. Außerdem kann man mithilfe eines einfachen Plugin seine E-Mails verschlüsseln, was in 6.2.2 beschrieben ist.

Einrichtung von Claws Mail

Claws Mail ist im Menü unter Internet zu finden und öffnet beim ersten Mal direkt einen Assistenten zur Einrichtung. Dieser fordert einen auf, in mehreren Schritten allgemeine Angaben zu machen, sowie zum Empfangen und Versenden.

Im ersten Schritt (Abbildung 3.10a) gibt man seinen Namen und seine E-Mail-Adresse an. Dieses muss die Adresse sein, für die das anschließende Konto eingerichtet wird.

Die Angabe des Servers zum Empfangen der E-Mails ist der zweite Schritt (Abbildung 3.10b). Hier ist zuerst der Servertyp bzw. das Protokoll auszuwählen. Dabei kann der Abschnitt 3.3.3 behilflich sein. Die Adresse des Servers bekommt man von seinem Anbieter.

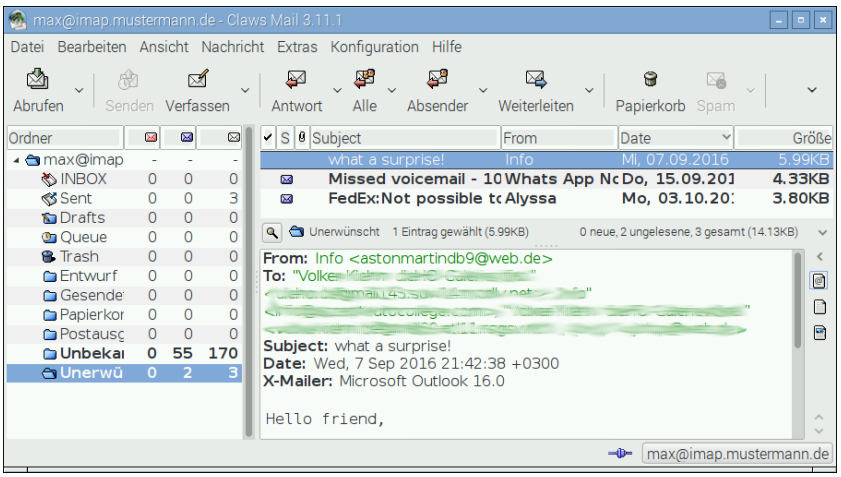


Abbildung 3.9: Oberfläche von Claws Mails mit geöffnetem »Spamordner«

Die Adressen der gängigsten Anbieter sind zusätzlich noch in der Tabelle 3.2 aufgeführt. Der Benutzername ist in der Regel der gleiche, mit dem man sich auch beim Webclient anmeldet.

Gibt man bei diesem Schritt das Kennwort mit an, so wird es gespeichert und man muss es nicht bei jedem Abruf von E-Mails mit angeben. Wenn auch andere Personen Zugriff auf den Raspberry Pi haben, sollte man auf diese Angabe verzichten. In jedem Fall sollte eine SSL-Verbindung zum Server verwendet werden. Damit wird die Verbindung verschlüsselt und es gibt nur noch wenige Anbieter, die überhaupt eine unverschlüsselte Verbindung zulassen.

Der dritte Schritt (Abbildung 3.10c) ist die Angabe des Servers zum Versandt der E-Mails. Hier ist in der Regel die Authentifizierung auszuwählen, da es wie auch bei der Verschlüsselung so gut wie keinen Anbieter gibt, der ohne entsprechende Zugangsdaten E-Mails annimmt. Lässt man die Felder für Benutzername und Kennwort frei, so werden die gleichen Daten wie beim Posteingangsserver genommen.

Beim ersten Empfang wird man noch damit konfrontiert, dass das Zertifikat des Servers unbekannt ist und gespeichert werden muss. Dieses kann auch nach einiger Zeit wieder kommen, wenn das Zertifikat abläuft und der Anbieter ein neues hinterlegt.

Sollte es beim Empfangen zu Problemen kommen, so kann beim Menüpunkt »Konfiguration« der Punkt »Konten bearbeiten« ausgewählt werden (Abbildung 3.11). Dort lassen sich alle getroffenen Einstellungen bearbeiten. Wurde IMAP gewählt, so kann sollte nach der Änderung mit der rechten Maustaste auf dem Ordnerbaum das Kontextmenü geöffnet werden und dort der Punkt »Ordnerbaum erneuern« gewählt werden. An dieser Stelle sei darauf verwiesen, dass einige Anbieter Besonderheiten haben, die zum Teil in 3.3.3 aufgeführt sind.

Wahl zwischen POP3 und IMAP

Bei fast allen E-Mail Anbietern wird einem die Möglichkeit geboten die E-Mails über das POP3 oder das IMAP Protokoll abzurufen. Der Versandt läuft in beiden Fällen über das SMTP-Protokoll. Da es zwischen den beiden Protokollen zum Abrufen einige Unterschiede gibt, sollen diese hier näher beleuchtet werden.

POP3 ist ein Protokoll, dass nur zum Abholen der E-Mails vom Server gedacht ist. Bei den E-Mail-Clients ist dabei die Standardeinstellung, dass anschließend die Mails auf dem Server gelöscht werden. Ruft man mit mehreren Geräten seine E-Mails ab, sollte man die Einstellung wählen, dass die E-Mails nicht gelöscht werden. In diesem Fall ist es auch so, dass jede neu heruntergeladene E-Mail als neu markiert wird, auch wenn sie bereits von einem anderen Gerät heruntergeladen wurden und dort als gelesen markiert wurden.

Bei **IMAP** bleiben die E-Mails eigentlich auf dem Server. Es werden im Client nur lokale Kopien angelegt. Dadurch hat man bei allen verwendeten Clients die gleiche Verzeichnisstruktur, Markierungen sind überall gleich und auch wenn E-Mails gelöscht werden, werden sie bei den anderen Clients entfernt. Über das IMAP Protokoll hat man so die Möglichkeit eine lokale Umgebung zu schaffen, in der man auch offline arbeiten kann, die dem entspricht, was man aus dem Abrufen der E-Mails im Browser kennt. Dieses ist nur ein sogenannter Webclient, der auch auf das IMAP-Protokoll setzt.

Anbieter	**IMAP**	**POP3**	**smtp**
Googlemail	imap.gmail.com	pop.gmail.com	smtp.gmail.com
GMX	imap.gmx.net	pop.gmx.net	mail.gmx.net
T-Online	secureimap.t-online.de	securepop.t-online.de	securesmtp.t-online.de
Web.de	imap.web.de	pop3.web.de	smtp.web.de

Tabelle 3.2: Server gängiger Anbieter

Besonderheiten bei verschiedenen Mail-Anbietern

Bei einigen der größeren E-Mail Anbietern sind noch ein paar Besonderheiten zu beachten, damit der Empfang und Versandt von E-Mails über Claws Mail funktioniert.

GMX und web.de In den Einstellungen in der Weboberfläche muss der Zugriff über IMAP/POP3 aktiviert werden. Dabei ist zu beachten, dass die Weboberfläche sich nicht mit dem Standardbrowser des Raspberry Pi öffnen lässt, da man dort direkt wieder abgemeldet wird. Eine Installation und Nutzung des Firefox-Browsers (siehe 3.3.2) bringt hier Abhilfe.

GMail Bei GMail müssen zwei Dinge in der Weboberfläche eingestellt werden: Zum einen muss in den normalen Einstellungen IMAP aktiviert sein. Außerdem stuft GMail das Claws Mail-Programm als unsichere App ein. In der Hilfe von Google

unter http://bit.ly/2jwmda7 wird beschrieben, wie man „Weniger sichere Apps“ den Zugriff erlauben kann. Auch hier muss ggf. der Firefox genutzt werden.

3.3.4 Screenshots

Zur Erstellung von Screenshots in der graphischen Oberfläche ist in der Regel das Programm scrot bereits installiert. Dieses kann direkt durch das Betätigen der »Drucken«-Taste gestartete werden. Stattdessen kann kann es auch in der Konsole aufgerufen werden. Ein Aufruf innerhalb der Oberfläche sieht es so aus:

```
scrot -cd 6
```

In beiden Fällen wird ein PNG-Bild im Homeverzeichnis gespeichert. Der Parameter »d 6« sorgt dafür, dass die Bildaufnahme um 6 Sekunden verzögert stattfindet. Durch die Angabe von »c« wird dabei zusätzlich der Countdown herunter gezählt, so dass man sieht wie lange man noch Zeit hat das Fenster wie gewünscht zurecht zu rücken. Das Bild hat automatisch das Aufnahmedatum im Namen. Alternativ kann man auch den Wunschnamen für das Bild angeben.

Auch das Bildbearbeitungsprogramm »gimp« bietet entsprechende Möglichkeiten, um Bildschirmphotos zu erstellen. Dort kann es anschließend direkt bearbeitet werden.

3.4 Die Verzeichnisstruktur

Die Verzeichnisstruktur auf einem Raspberry Pi mit Raspian entspricht der Standardstruktur fast aller Linux Systeme. In ihnen wird alles unterhalb des Hauptzweiges, der mit einem einfachen / angegeben wird, angeordnet. Es gibt also, im Gegensatz zum Windowssystem keine Verzeichnisbuchstaben. Werden weitere Datenträger dem System hinzugefügt, sind sie entweder unter /mnt/ oder /media/ zu finden.

In dieser Verzeichnisstruktur gibt es einige wichtige Verzeichnisse direkt auf Hauptebene. Sie erfüllen dabei bestimmte Aufgaben bzw. haben bestimmte Zuordnungen:

/etc In diesem Verzeichnis sind alle für das System relevanten Einstellungen bzw. Konfigurationsdateien zu finden.

/boot Dieses Verzeichnis enthält alle nötigen Elemente zum Hochfahren des Systems

/home In der Regel erhält jeder Benutzer ein eigenes Verzeichnis, sein »home«. Alle diese Verzeichnisse sind hier zu finden

/root Die obere Regel hat auch eine Ausnahme: Das Homeverzeichnis des Administrativen Benutzers, dem »root« liegt direkt in der untersten Ebene.

/bin Hierin findet sich eine Vielzahl von Programmen, die z. B. in der Shell direkt aufgerufen werden können.

/dev Über dieses Verzeichnis sind alle Geräte und Elemente des Systems zu finden, wie z.B. die SD-Karte oder der Arbeitsspeicher.

3.4.1 Dateinamen

Die Namen der Dateien sind auf einem Linuxsystem ähnlich aufgebaut, wie bei anderen Systemen auch. Es ist aber dabei zu beachten, dass der Groß- und Kleinschreibung eine wichtige Bedeutung zukommt. So sind kann es die Dateien »beispiel.txt« und »Beispiel.txt« in einem Verzeichnis geben. Bei Windowssystemen wäre dieses nicht möglich. Entsprechend sollte man dieses bei der Wahl eines Dateinamens beachten.

Bei der Wahl eines Dateinamens sind aber noch andere Dinge zu beachten. Dieses gilt besonders dann, wenn man die Dateien in Skripten einsetzt oder auf ein anderes System transferieren möchte. Daher sollte auf Umlaute wie ä, ö oder ü verzichtet werden. Stattdessen bietet sich die ae, oe oder ue Schreibweise an. Auch Leerzeichen sollten in den Namen von Dateien nicht auftauchen. Besser ist hier die Verwendung eines Unterstrichs oder der Schreibweise namens »CamelCase« oder im Deutschen »Binnenmajuskel«. Dabei wird der Dateiname, wenn er aus mehreren Wörtern besteht, zusammengeschrieben und jedes neue Wort mit einem großen Buchstaben begonnen. So ergeben sich wie bei einem Kamel mit seinen Höckern entsprechende Erhebungen. Ein Beispiel wäre »AnleitungZumRaspberryPi.txt«.

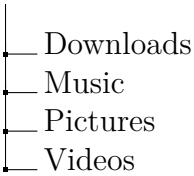
3.4.2 Strukturen im Benutzerverzeichnis ...

...oder Ordnung ist das halbe Leben. Im Benutzerverzeichnis, welches auch oft als Home-Dir bezeichnet wird, sollten die persönlichen Dokumente eines jeden Benutzers auf einem Computer liegen. Jeder Benutzer hat sein eigenes Benutzerverzeichnis, auf das andere Benutzer keinen Zugriff haben sollten. Bei vielen Personen dient aber nur der Desktop direkt als Ablageort für alle Dateien, die irgendwie gespeichert werden müssen. Dadurch wird die Anzeige des Desktops oft schnell voll und vieles lässt sich nur schlecht wiederfinden. Hinzukommt, dass der Desktop eigentlich nur ein Unterverzeichnis des Benutzerverzeichnis ist.

Bei vielen System wird von Beginn an eine Struktur von Verzeichnissen im Benutzerverzeichnis angelegt, die man gut für eine Strukturierung nutzen kann. Ruft man auf einem Raspberry Pi in der Konsole direkt `tree -d` auf[4], so bekommt man eine solche Übersicht:

```
.
├── Desktop
└── Documents
```

[4]Ggf. kann das Programm mit `sudo apt install tree` nachinstalliert werden.



Damit sind Orte für einige Elemente vorgegeben. Diese sollten sinnvoll weiter unterteilt werden, um weiterhin eine leicht eine Übersicht behalten zu können und Dateien einfach wieder finden zu können.

Aufgaben



Aufgabe 3.4

Lasse dir deinen eigenen Verzeichnisbaum deines Benutzerverzeichnisses anzeigen.



Aufgabe 3.5

Überlege dir eine geeignete Struktur für das Verzeichnis Documents. Die verschiedenen Schulfächer und Themen sind dabei mögliche Anhaltspunkte. Zeichne deine Struktur auf und besprich sie anschließend mit einem Partner.

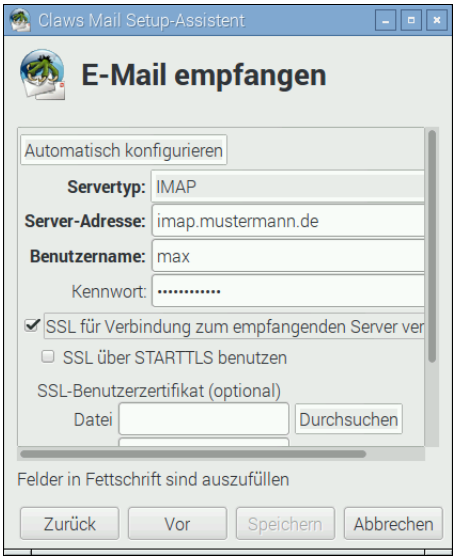


Aufgabe 3.6

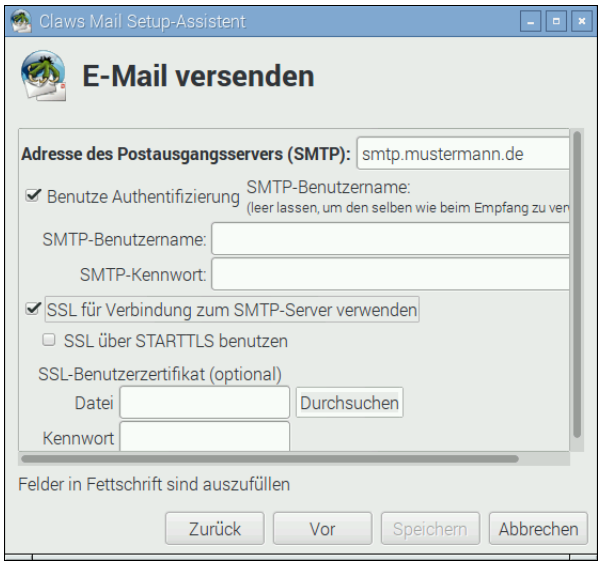
Setze deine Struktur auf dem Raspberry Pi um, indem du mit `mkdir` entsprechende Verzeichnisse erstellst.



(a) Allgemeine Angaben



(b) Empfangen



(c) Senden

Abbildung 3.10: Einrichtung von Claws Mail

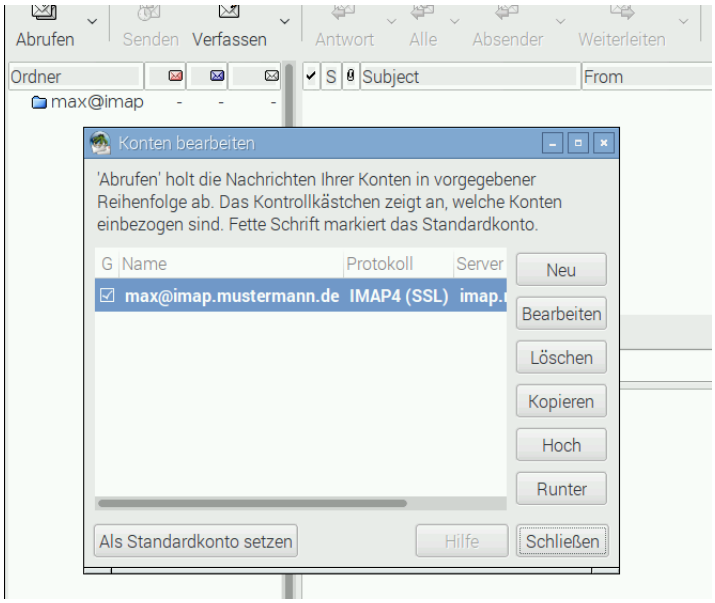


Abbildung 3.11: Änderungen an den E-Mail-Konten

Kapitel 4

Python Programmierung

Auf dem Raspberry Pi können Programme ausgeführt werden, die mit verschiedenen Programmiersprachen erstellt wurden. Es bietet sich aber an Python zu nutzen, da es für diese Sprache alle nötigen Bibliotheken gibt, um die verschiedenen Hardwareelemente des Raspberry Pi anzusteuern. Da beim Wechsel von der Version 2.7 auf 3.0 bei Python einige vordefinierte Methoden umbenannt und Dopplungen entfernt wurden, sind einige Programme, die für 2.x geschrieben wurden, nicht mehr lauffähig. Deshalb wird vorerst die Version 2.7 auch weiter gepflegt. Als Neueinsteiger sollte man deshalb auch mit Python 3 arbeiten.

4.1 Grundlagen in Python

Um die Grundlagen der Programmiersprache Python zu erlernen gibt es verschiedene Bücher, Kurse und Unterrichtsreihen. An dieser Stelle geht die Empfehlung an ein Onlineangebot der Universität Waterloo, um dessen deutsche Fassung sich die Bundesweite Informatikwettbewerbe (BWINF) gekümmert hat (Bundesweite Informatikwettbewerbe 2015). Dieses Angebot ist unter der Adresse http://bit.ly/bwinf-kurs zu erreichen.

Dieses Angebot des BWINF setzt komplett auf Python 3 und bietet auch die Möglichkeit, nach einer Registrierung, Fragen zu den Aufgaben zu stellen. Bei den Aufgaben gibt es aber auch verschiedene Hilfen und eine automatische Überprüfung, ob das Programm die geforderte Aufgabe erfüllt.

4.2 Programmierumgebung auf der Raspberry Pi

Auf dem Raspberry Pi ist Python 2 und Python 3 in der Regel schon vorinstalliert, so dass man sofort damit arbeiten kann. Auch eine integrierte Entwicklungsumgebung (engl. Integrated Development EnvironmentIDE kurz IDE) die man über das Menü aufrufen kann, ist für beide Versionen vorhanden.

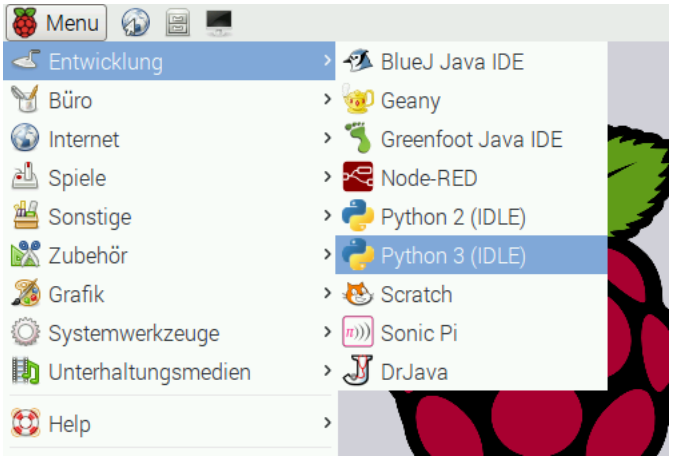


Abbildung 4.1: Starten der Python IDE

Die Entwicklungsumgebung startet mit der sogenannten Python-Shell (siehe Abbildung 4.2), in der man Kommandos eingeben kann, so dass sie direkt ausgeführt werden. So lassen sich schnell einzelne Elemente ausführen.

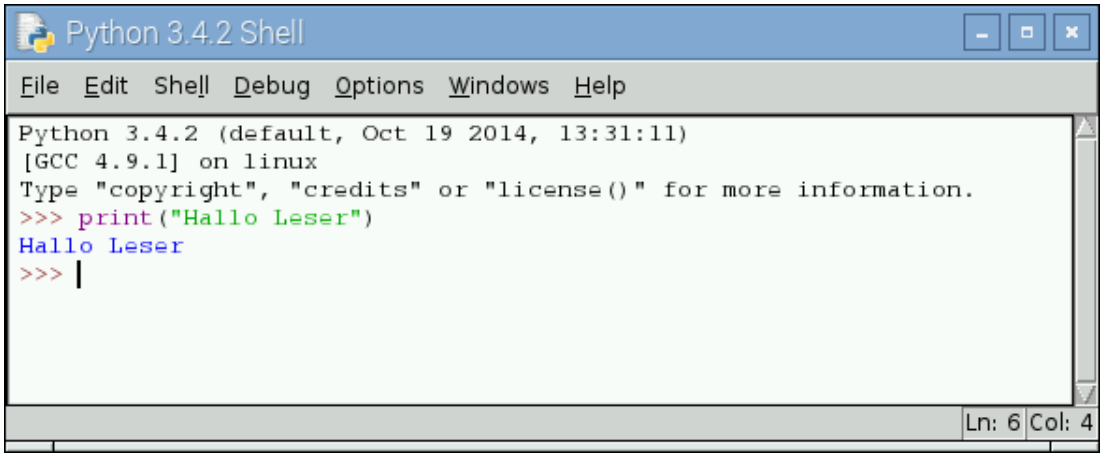


Abbildung 4.2: Ausführen in der Python-Shell

Um kleine und größere Programme zu entwickeln reicht die Shell alleine aber nicht aus. Hier sollte der Quelltext in eine Datei geschrieben werden. Einen passenden Editor bekommt man über das Menü der Shell unter dem Punkt »File«. Dort muss man entweder eine neue Datei erstellen oder eine bestehende öffnen. Dabei sollte man die Regeln für gute Dateinamen beachten (siehe 3.4.1).

Das dort abgespeicherte Programm lässt sich dann über den Menüpunkt »Run« und »Run Module« (siehe Abbildung 4.3) starten. Die Ausführung findet in der geöffneten Python-Shell statt.

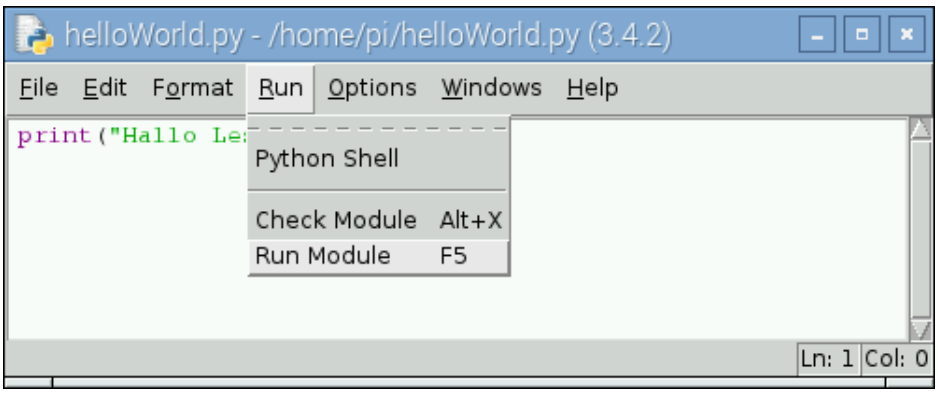


Abbildung 4.3: Starten eines Python Programms in der IDE

4.2.1 Python in der Konsole

Python lässt sich aber auch ohne IDE direkt in der Konsole ausführen. Ein einfacher Aufruf von python3 öffnet einem die Python-Shell. Dabei ist zu beachten, dass der kürzere Aufruf python die Shell der Version 2.7 öffnet.

```
pi@raspberrypi:~ $ python3
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hallo Leser")
Hallo Leser
>>>
```

Abbildung 4.4: Python-Shell in der Konsole

Hängt man dem Aufruf von Python den Namen einer Pythondatei als Parameter an, so wird diese direkt ausgeführt. Für die Bearbeitung der Datei ist hier einer der gängigen Editoren, wie z. B. vi oder nano zu nutzen.

```
pi@raspberrypi:~ $ python3 halloLeser.py
Hallo Leser
pi@raspberrypi:~ $
```

Abbildung 4.5: Starten eines Python Programms in der Konsole

4.3 Minecraft mit Python beeinflussen

Das Spiel Minecraft gibt es in verschiedenen Versionen. Auf dem Raspberry Pi läuft eine ältere Version die für den Raspberry Pi angepasst wurde. Es gibt in ihr nur den

Modus, in dem man die Welt auch mit mehreren Spielern verändern kann, aber z. B. nicht gegeneinander kämpfen kann. Dafür gibt es die Möglichkeit mit Hilfe von Python direkt Einfluss auf ein laufendes Spiel zu nehmen.

4.3.1 Grundlegendes

Um von Python in ein Minecraft-Spiel eingreifen zu können, muss zuerst in Python die passenden Pakete importiert werden. Mit dessen Hilfe lässt sich eine Verbindung zum aktuellen Minecraft-Spiel erstellen, das auf dem Raspberry Pi läuft, auf dem auch Python ausgeführt wird:

```
1 from mcpi.minecraft import Minecraft
2 import mcpi.block as block
3 mc = Minecraft.create()
```

Mit dem dann erzeugten Minecraft-Objekt, das über mc zu erreichen ist, lassen sich weitere Dinge ansteuern, die dann auch über die Python-Konsole direkt nutzbar sind.

4.3.2 Steine setzen

Am einfachsten lässt sich die Minecraft-Welt dadurch beeinflussen, dass man an bestimmte Positionen Steine setzt. Es können sowohl einzelne Steine, als auch mehrere Steine als Block gesetzt werden. Einen einzelnen Stein setzt man mit Hilfe von setBlock(args) der mehrere Argumente benötigt. Die ersten drei Argumente sind für die x-, y- und z-Koordinaten. Das darauf folgende Argument ist der Blocktyp, mit dem man angibt, welche Steinsorte gesetzt werden soll. Bei einigen Steinen lässt sich noch optional ein weiteres Argument übergeben, was z. B. beim Treppenstein die Ausrichtung ist.

Um einen ganzen Block zu setzten wird die Methode setBlock(args) benutzt. Sie erwartet bei den ersten sechs Argumenten die Koordinaten von den zwei Ecken, die den Block begrenzen. Das folgende Argument gibt wieder den Typ an und als letztes Argument kann wieder optional die weitere Einstellungen, die vom gewählten Steintyp abhängig sind, gesetzt werden.

Im folgenden Beispiel wird dadurch ein Dreckstein an der Position (0,20,0) gesetzt und ein Block aus 9 Holzsteinen im Bereich von (-1,-1,-1) bis (1,1,1) gesetzt.

```
1 mc.setBlock(0, 20, 0, block.DIRT.id)
2 mc.setBlocks(-1, -1, -1, 1, 1, 1, block.WOOD.id)
```

Eine Auswahl an möglichen Steintypen sind: AIR, STONE, DIRT, WATER, ICE und CACTUS.

4.3.3 Aktuelle Postion – Wo bin ich?

Um im seinen eigenen Sichtbereich die Blöcke zu setzten, sollte man aber auch die Postion des Spielers wissen. Diese wird im Spiel immer in der oberen linken Ecke angezeigt. Sie lässt sich aber auch mit Hilfe von x,y,z=mc.player.getPos() auslesen. Dabei werden in x, y, und z die jeweilige Koordinate gespeichert, die dann weiter genutzt werden kann.

Aufgabe 4.1

Platziere in der Nähe der Position der Spielfigur verschiedenen einzelne Steine und Blöcke. Setze damit ein Muster zusammen, wie z. B. eine Pyramide.

Aufgabe 4.2

Grabe im Spiel einen Tunnel in den Boden oder in einen Berg. Lasse dann durch Python eine große Halle um deine Figur herum entstehen.

4.3.4 Analyse der Umgebung

Der ideale Baumeister geht auf seine Umgebung ein. So sollte z. B. ein Haus nicht in der Luft schweben, sondern auf dem Boden aufsetzten, auch wenn dieses in Minecraft möglich ist. Die Untersuchung der Umgebung ist mit blockType = mc.getBlock(x,y,z) möglich. Diese Methode liefert den Typ des Blocks zurück. So wird im folgenden Beispiel mögliches Wasser an der übergebenen Position durch Eis ersetzt.

```
1 if mc.getBlock(3, 5, 9) == block.WATER.id :
2     mc.setBlock(3, 5, 9, block.ICE.id)
```

Aufgabe 4.3

Begebe dich in die Nähe eines Sandstrands und ersetze in einem kleinen Bereich (10 x 10 x 10 Steine) alle Sandsteine durch Granit.

Aufgabe 4.4

Erstelle an einem Berg oder Hügel eine 10 x 10 Platzform. Sorge dabei dafür, dass sie an allen Punkten auf oder über dem aktuellen Bodenlevel liegt. Da sie nicht schweben soll, muss ggf. automatisch bis zum Boden aufgefüllt werden.

4.3.5 Mehrspielermodus

Wenn man mit mehreren Spielern an einem Spiel teilnimmt ist es wichtig, dass man sich auch mit dem Spiel der Person direkt verbindet, auf dessen Raspberry das Spiel gestartet wurde. Ansonsten bekommen alle anderen von den eigenen Aktionen nichts mit.

Dazu muss man beim Erzeugen des Minecraft-Objekts die entsprechende IP-Adresse des Servers angeben. Dazu wird im Kopf des Programms die entsprechende Zeile durch mc = Minecraft.create(<Adresse des Servers>) ersetzt. Hier lässt sich einfach die IP-Adresse eingeben, die man in Minecraft im Mehrspielermodus bei der Auswahl des Spieles angezeigt bekommt.

Jeder Spieler wird im Spiel mit einer eigenen ID gelistet. Eine Liste aller Spieler-IDs kann man mit entityIds = mc.getPlayerEntityIds() erhalten. Diese kann man auch nutzen, um dann die Position eines Spieler unter Angabe seiner ID zu erhalten. Dieses geht mit x,y, z = mc.entity.getPos(entityId). Das folgende Beispiel gibt die IDs aller Spieler und deren Position aus:

```
1 playerIds = mc.getPlayerEntityIds()
2 print("Spieler-IDs:", playerIds)
3 for id in playerIds
4     x, y, z = mc.entity.getPos(id)
5     print("Spieler", id, "bei␣x:", x, "y:", y, "z:", z)
```

4.3.6 Weitere Informationen

Eine Referenz zu vielen weiteren möglichen Befehlen ist auf der Seite http://www.stuffaboutcode.com/p/minecraft-api-reference.html zu finden.

Kapitel 5

Elektrische Schaltungen

Eine Vielzahl der Einsatzmöglichkeiten des Raspberry Pi basieren auf der Möglichkeit, weitere Hardware anzuschließen. Besonders die selbstgebaute Hardware setzt sich aus elektrischen Schaltungen zusammen, die in der Regel erst provisorisch aufgebaut werden und mit dem GPIO des Raspberry Pi verbunden werden. Dabei sind diese Schaltungen in vielen Fällen aus Leuchtdioden, Schaltern und Chips mit besonderen Aufgaben aufgebaut. Fast alle dieser Schaltungen benötigen auch elektrische Widerstände zur Absicherung oder um den Stromfluss zu regeln. Werden diese Schaltungen länger gebraucht bietet es sich an, diese auf eine Platine zu löten. Wie diese Schaltungen genau aufgebaut sind kann man in der Regel Schaltplänen entnehmen.

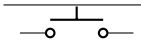
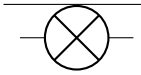
5.1 Schaltpläne und Schaltzeichen

In Schaltplänen werden verschiedene Schaltzeichen genutzt um die Verbindungen zwischen den Bauteilen zu verdeutlichen. Generell ist bei Schaltplänen zu beachten, dass alle Verbindungen nur waagerecht und senkrecht gezeichnet werden. Bei handschriftlichen Zeichnungen ist dabei die Benutzung eines Lineal verpflichtend. Im folgenden sind die einzelnen Elemente mit kurzer Erläuterung angegeben.

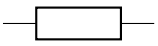
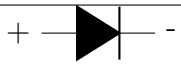
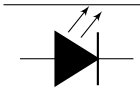
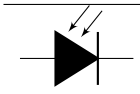
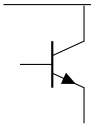
5.1.1 Verbindungen und Anschlüsse

	Verbindungen um die Ecke.
	Kreuzende Verbindungen ohne elektrischen Kontakt.
	Kreuzende Verbindungen mit elektrischen Kontakt zwischen den beiden Verbindungen.
	Abzweigende Verbindung.
	Batterie, wobei der längere Strich den Pluspol angibt.
	Spannungs- oder Stromquelle.
	Pin mit abgehender Verbindung.
	Nicht angeschlossener Pin.
	Massenanschluss, der auch mehrmals innerhalb eines Schaltplans genutzt werden kann. Alle Massenanschlüsse sind mit einander verbunden, so dass man darüber Verbindungen einsparen kann.

5.1.2 Schalter, Taster und Lampen

	Ein normaler Schalter, der seinen Zustand nach der Betätigung behält.
	Ein Taster, der nur während der Betätigung geschlossen ist.
	Eine einfache Lampe.

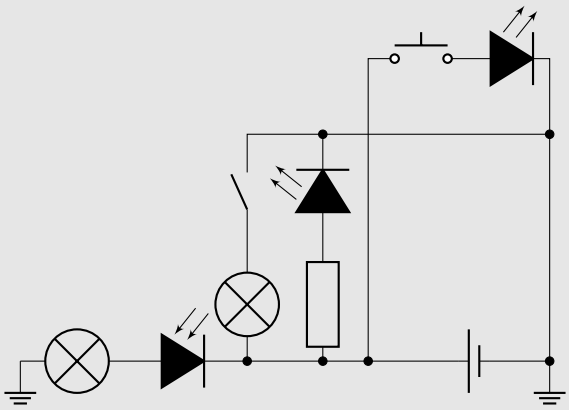
5.1.3 Elektrische Bauteile

	Ein Widerstand, der den elektrischen Strom hemmt. In der Regel wird auch die Stärke mit angegeben.
	Eine Diode, die den Strom nur in eine Richtung durchlässt.
	Eine Leuchtdiode (LED), die leuchtet, wenn sie den Strom in der vorgesehenen Richtung durchlässt. In der anderen Richtung sperrt sie.
	Eine Fotodiode, die den Strom in Abhängigkeit von der darauf fallenden Lichtstärke, auch in Sperrrichtung leitet.
	Ein Transistor, mit dem entweder Strom verstärkt werden kann oder der als elektrischer Schalter dient.



Aufgabe 5.1

Zähle auf, aus welchen Bauteilen die folgende Schaltskizze enthält und mit welchen Bauteilen diese verbunden sind.





Aufgabe 5.2

Zeichne einen Schaltplan für vier LEDs, die jeweils an einem einzelnen Pin angeschlossen sind. Beachte dabei, dass jede LED ihren eigenen Vorwiderstand benötigt. Zwei der LEDs sollen sich auch alternativ durch einen Taster bzw. einen Schalter anschalten lassen.

5.2 Elektrische Widerstände

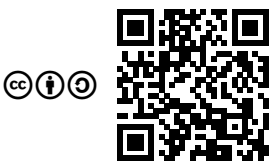
Mit Hilfe von elektrischen Widerständen kann man Ströme und Spannungen regulieren, um damit andere elektrische Bauteile, wie z. B. Leuchtdioden zu schützen. Je nach Anwendungsbereich gibt es verschiedene Bauformen von Widerständen und auch unterschiedliche Stärken.

5.2.1 Farbcodierung von Widerständen



Abbildung 5.1: Ein Widerstand mit 1 kΩ

Bei Schaltungen mit dem Raspberry Pi wird in der Regel eine zylindrische Bauform von Widerständen benutzt, deren Werte man durch eine entsprechende Farbcodierung ablesen kann. Diese besteht aus vier Ringen, bei denen die ersten drei Ringe die Stärke



angeben. Der letzte Ring, in normalen Gebrauch ein silberner oder goldener, gibt an, wie genau der Widerstand gefertigt ist.

Farbe	1. Ring	2. Ring	3. Ring
silber	–	–	$10^{-2} = 0{,}01$
gold	–	–	$10^{-1} = 0{,}1$
schwarz	–	0	$10^{0} = 1$
braun	1	1	$10^{1} = 10$
rot	2	2	$10^{2} = 100$
orange	3	3	$10^{3} = 1.000$
gelb	4	4	$10^{4} = 10.000$
grün	5	5	$10^{5} = 100.000$
blau	6	6	$10^{6} = 1.000.000$
violett	7	7	$10^{7} = 10.000.000$
grau	8	8	$10^{8} = 100.000.000$
weiß	9	9	$10^{9} = 1.000.000.000$

Tabelle 5.1: Farbcodierung für die Ringe an Widerständen

Zur Berechnung des Wertes werden die Zahlen der ersten beiden Ringe hintereinander gestellt und dann mit dem Faktor des dritten Rings multipliziert. So ergibt sich, dass ein Widerstand mit braun, schwarz und orange einen Wert von $10 \cdot 10^3\,\Omega = 10.000\,\Omega = 10\,k\Omega$ und orange, orange und braun $33 \cdot 10^1\,\Omega = 330\,\Omega$ hat.



Aufgabe 5.3

Bestimme für folgende Widerstände den Wert bwz. die Farbcodierung:
a) rot, gelb, grün b) violett, rot, gold c) braun, schwarz, grün d) gelb, weiß, blau
e) $7,8\,M\Omega$ f) $1,6\,k\Omega$ g) $420\,\Omega$ h) $300\,m\Omega$

5.2.2 Widerstände in Schaltzeichnungen

In Schaltplänen wird oft neben den Schaltzeichen für Widerstände deren Wert in einer kompakten Schreibweise angegeben. Dabei wird der Buchstabe R als Dezimaltrennzeichen genutzt. So bezeichnet 10R einen Widerstand mit $10\,\Omega$ und 0R5 einen mit $0,5\,\Omega$. In gleicher Weise können auch die Präfixe bei Einheiten wie z. B. k für kilo (1000) genutzt werden. So gibt 4M5 einen Widerstand mit $4,5\,M\Omega$ an.



Aufgabe 5.4

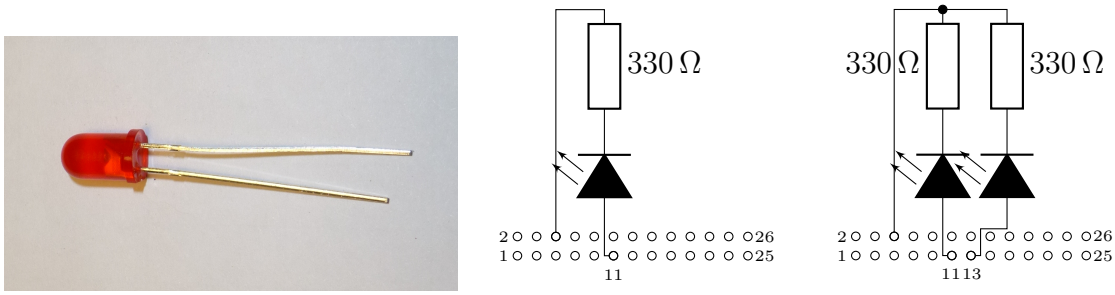
Gib für folgende Widerstände die kompakte Schreibweise bzw. ihren Wert an:
a) 58R b) 3k4 c) M95 d) 73k
e) $1,6\,k\Omega$ f) $30\,\Omega$ g) $650\,k\Omega$ h) $620\,m\Omega$

5.3 Einfache Elektrische Schaltung am GPIO

Mit Hilfe des GPIO lassen sich Elemente wie Leuchtdioden durch Programme steuern und die Eingabe von Schaltern abfragen. Dazu stehen alle programmierbaren Pins als Eingang und als Ausgang zur Verfügung. Man muss nur im Programm festlegen, welche dieser beiden Eigenschaften sie jeweils einnehmen sollen. Da sie Digital sind kann über sie jeweils nur True oder False gesendet bzw. empfangen werden. Alternativ gehen auch Angaben wie 1 oder 0.

5.3.1 Anschluss einer LED

Beim Anschluss einer LED an den GPIO muss beachtet werden, dass diese einen Vorwiderstand benötigen, damit sie nicht durchbrennen. Außerdem ist die Anschlussrichtung bei der LED zu beachten. Wie auf dem Foto (5.2a) zu erkennen hat die LED zwei verschieden lange Beine. Das längere ist der Pluspol und muss auf der Seite des geschalteten Ports liegen, während das kürzere mit der Masse (GND) verbunden wird. In der mittleren Schaltskizze (5.2b) ist die LED so verbunden, dass sie über den Port 11 geschaltet wird.



(a) Eine rote LED **(b)** Schaltung LED **(c)** Schaltung von zwei LEDs

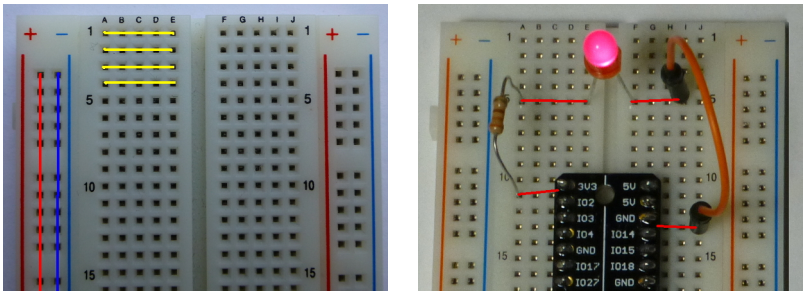
Abbildung 5.2: Leuchtdioden mit ihrer Beschaltung

Sollen mehrere LEDs beschaltet werden, bekommt jede LED einen eigenen Vorwiderstand. Es gibt aber eine gemeinsame Verbindung zur Masse. Eine solche Schaltung für zwei LEDs ist in der rechten oberen Schaltskizze (5.2c) aufgezeichnet worden. Bei dieser lassen sich die beiden über die Ports 11 und 13 schalten.

5.3.2 Erste Versuche auf dem Steckbrett

In vielen Fällen will man Schaltungen ausprobieren, bevor man sie z. B. auf eine Platine lötet. Dafür eignet sich am besten ein Steckbrett, auf dem die verschiedenen Bauteile in die Löcher eingesteckt werden können und mit einfachen Steckkabeln entsprechend

verbunden werden. Die Löcher eines Steckbretts sind dabei teilweise miteinander verbunden. Deshalb muss man nicht mehrere Kabel in ein Loch quetschen, sondern kann sie nebeneinander setzen. In der Mitte sind immer fünf nebeneinander liegende Löcher elektrisch leitend miteinander verbunden. Dieses sind auf dem unteren linken Foto (5.3a) gelb markiert. Die Reihen untereinander sind voneinander isoliert.



(a) Interner Belegung **(b)** Angeschlossene LED

Abbildung 5.3: Steckbrett

Zusätzlich gibt es außen noch Lochbahnen, auf dem Foto rot und blau markiert, die das gesamte Steckbrett entlang laufen. In der Regel werden diese für die allgemeine Spannungsversorgung genutzt. Dazu müssen sie aber erst mit den entsprechenden Pins des GPIO oder einer externen Spannungsversorgung verbunden werden. Genauso sind auch die Bahnen auf der linken Seite nicht mit der rechten Seite verbunden, solange nicht ein entsprechendes Kabel gesteckt wurde.

Im rechten oberen Bild (5.3b) kann man sehen, wie auf dem Steckbrett mit einem entsprechenden Adapter eine LED angeschlossen wurde. Der Widerstand ist dabei, mit den roten Linien eingezeichnet über das Steckbrett sowohl mit der LED, als auch mit dem 3,3 V des Raspberry Pi verbunden, der an Pin 1 liegt. Das sogenannte Jumperkabel schließt dann die Verbindung zwischen der LED und dem Ground des Raspberry Pi an Pin 6. In diesem Fall leuchtet die LED dauerhaft, so dass man auch kontrollieren kann, ob die LED richtig herum eingesteckt ist.

5.3.3 GPIO Belegung

Um einen einfache Übersicht über die Ports des GPIO und ihre Belegung zu erhalten, kann in der Übersicht (5.4) die linke obere Zeichnung für den Raspberry Pi Modell B oder Modell A genutzt werden. Schneidet man sie aus und versieht sie an den passenden Stellen mit Löchern, so kann sie auf den GPIO gesteckt werden. Beim Modell B+, A+, 2B, Zero und 3B stimmen die ersten 26 Pins in ihrer Belegung mit dem Modell A und B überein. Die komplette Belegung kann der linken unteren Zeichnung entnommen werden.

Hinweis: Bei allen Steckvorgängen sollte der Raspberry Pi nicht mit der Stromversorgung verbunden sein, da es sonst zu Beschädigungen kommen kann.

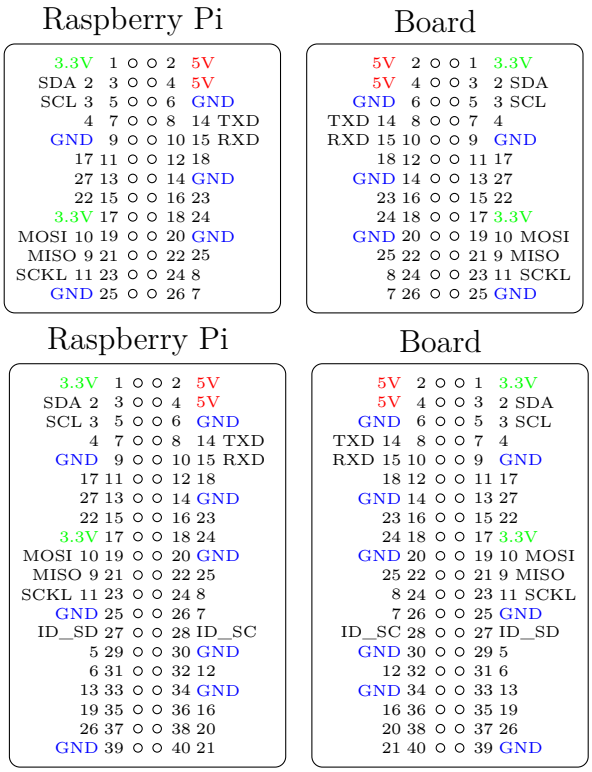


Abbildung 5.4: GPIO Belegung

Um die Pins des Raspbery Pi auf ein Steckbrett zu bekommen kann man einen entsprechenden Adapter nutzen, wie er z. B. im Foto (5.3b) zu sehen ist. Eine Alternative dazu ist z. B. ein altes Festplattenkabel, wie es in den Abbildungen (5.5) zu sehen ist. Für diesen Fall ist die rechte Seite der Übersicht (5.4), die die Überschrift Board trägt. Hier sind die Seiten vertauscht gegenüber der Belegung am Raspberry Pi vertauscht. Diese Vertauschung ist durch die Kabelführung bedingt. Steckt man durch einige Kabel die Beschriftung auf dem Ende des Festplattenkabels fest, so kann man direkt die anderen Pins abgreifen. Beim Auflegen und Anschließen sollte dabei beachtet werden, dass die Markierung am Kabel sich auf der Seite mit dem Pin 1 befindet. Auf dem Raspberry Pi liegt dieser an der Seite, an der auch die SD-Karte eingesteckt wird.

Je nach Programmiersprache werden die verschiedenen Nummerierung des GPIO genutzt. Scratch nutzt z. B. die zusätzlich angegebenen Nummern. Bei Python lässt es sich einstellen, ob die Durchnummerierung genutzt werden soll oder die zusätzlichen.

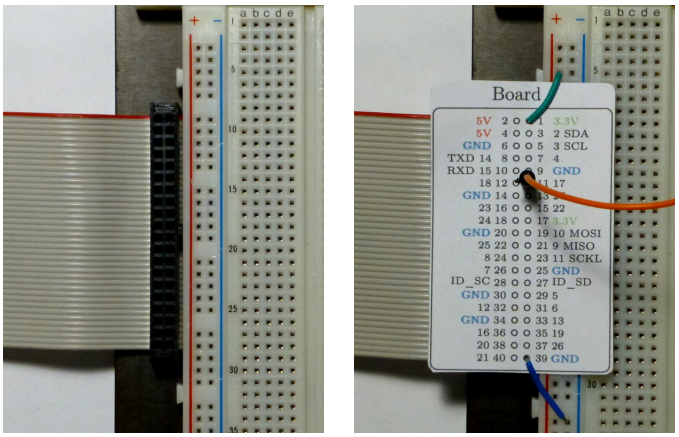
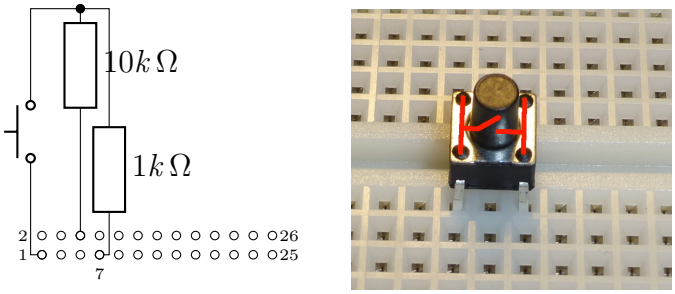


Abbildung 5.5: Festplattenkabel am Steckbrett

5.3.4 Anschluss eines Schalters oder Tasters

Für den Anschluss eines Schalters oder Tasters an den GPIO werden immer drei Anschlüsse benötigt, damit sichergestellt ist, dass der Eingang entweder mit der Masse (GND) oder einer Spannung verbunden ist. Dabei ist zu beachten, dass hier maximal der 3,3 V Anschluss genutzt wird. Bei der unten stehenden Schaltskizze (5.6a) ist bei geöffnetem Schalter eine direkte Verbindung zwischen dem Pin 7 und der Masse vorhanden. Wird der Schalter gedrückt, so liegt durch den geringeren Widerstand am Schalter eine Spannung von 3,3 V an.



(a) Anschluss **(b)** Interne Beschaltung

Abbildung 5.6: Schalter oder Taster

Einfachen Taster die nutzen kann haben insgesamt vier Beine, wie er in Abbildung 5.6b zu sehen ist. Bei ihnen sind, wie mit der aufgemalten Schaltskizze verdeutlicht, immer die zwei gegenüberliegende Beine miteinander verbunden. Der Taster schließt also nur die Verbindung zwischen den beiden Beinpaaren. Dieses ist bei der Beschaltung eines solchen Tasters zu beachten.

5.4 Ansteuerung des GPIO mit Python

An den GPIO angeschlossene Elemente können mit Hilfe der RPi.GPIO Bibliothek mit einem Python-Programm gesteuert werden. Mit ihr stehen alle programmierbaren Pins als Eingang und als Ausgang zur Verfügung. Dieses muss nur innerhalb des Programms festlegen bzw. die Festlegung übernehmen. Wurde für einen Pin bereits eine Festlegung getroffen und nicht aufgehoben, bekommt man sonst eine entsprechende Warnung.

5.4.1 Einstellungen

Die Bibliothek wird in Python mit import RPi.GPIO as GPIO eingebunden werden. Dann kann man mit GPIO.setmode(GPIO.BOARD) die Zählweise der Pins setzen, so dass man sie von 1 bis 26 herunter zählen kann. Möchte man die Pins in Python mit den Nummern ansprechen, die zusätzlich angegeben sind, so ist GPIO.BCM stattdessen als Mode anzugeben. Durch GPIO.setup(11, GPIO.OUT) oder GPIO.setup(12, GPIO.IN) sollte anschließend festgelegt werden, ob ein Pin als Ausgang oder Eingang genutzt wird.

5.4.2 Ausgänge steuern

Wurden die entsprechenden Einstellungen getroffen, so kann man an entsprechenden Stellen im Programm mit GPIO.output(11, False) den entsprechenden Pin ausschalten. In diesem Fall ist es Pin 11. Ein komplettes Programm, was zwischen zwei LEDs hin und her schaltet sähe dann so aus:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 #Setzen der Nummernzaehlung der Pins. Alternativ GPIO.BCM
5 GPIO.setmode(GPIO.BOARD)
6
7 #Setzen der Pins auf Ausgabe
8 GPIO.setup(11, GPIO.OUT)
9 GPIO.setup(13, GPIO.OUT)
10
11 #Beide Pins ausschalten
12 GPIO.output(11, False)
13 GPIO.output(13, False)
14
15 #Und ewiges Wechseln...
16 while 1:
17   GPIO.output(11, True)
18
```

```
19   #Fuer 1/5 Sekunde warten
20   time.sleep(0.2)
21
22   GPIO.output(11, False)
23   GPIO.output(13, True)
24
25   time.sleep(0.2)
26   GPIO.output(13, False)
```

5.4.3 Eingänge

Analog zum Belegen der Ausgänge lässt sich auch in Python mit GPIO.input(11) abfragen, ob ein Eingang belegt ist oder nicht. Diese Methode liefert einen entsprechenden Boolschen Wert zurück. Eine Python-Programm könnte dann so aussehen:

```
1 import RPi.GPIO as GPIO
2 import time
3
4 #Einstellungen
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(7, GPIO.IN)
7
8 while 1:
9   if GPIO.input(7):
10    print("Taster unten")
11  else :
12    print("Taster oben")
13  #Fuer 1/10 Sekunde warten
14  time.sleep(0.1)
```

5.5 Pulsweitenmodulation

Alle Pins am GPIO des Raspberry Pi sind digital. Das bedeutet, dass sie bei ihnen nur zwei verschiedene Zustände möglich sind: An/Aus bzw. 1/0 oder Wahr/Falsch. Oft genug gibt es aber Fälle in denen Zwischenwerte gewünscht werden. So soll eine LED nicht mit voller Helligkeit leuchten oder ein Motor nur mit halber Geschwindigkeit drehen. Besonders bei einer RGB-Diode kann man erst dann die verschiedensten Farben aus Rot, Grün und Blau mischen, wenn die Farbanteile unterschiedlich sind. So wird die Farbe Orange aus 100% Rot, 50% Grün und 0% Blau gemischt.

Diese Zwischenwerte am GPIO können nicht direkt realisiert werden, sondern nur mit einem Trick. Wenn bei LEDs immer nur für kurze Zeit der Strom fließt und dieses sehr oft hintereinander wiederholt, so nimmt unser Auge es so wahr, als würde die LED

etwas schwächer leuchten. Auch Motoren drehen sich dann durch ihre Trägheit nur halb so schnell, wenn nur bei der Hälfte der Zeit eine Spannung anliegen hat.

Für dieses schnelle Umschalten kann man die Pulsweitenmodulation nutzen. Dabei wird eine Folge Strom an und aus, mit einer wählbaren Frequenz wiederholt. Außerdem kann man dabei angeben, bei welchem Anteil dieser Folge der Strom angeschaltet ist.

```
1 # Pin 18 mit Frequenz von 50 Hz
2 p = GPIO.PWM(18, 50)
3 # Starten mit 30% Strom an
4 p.start(30)
5 # Ändern auf 50% Strom an
6 p.ChangeDutyCycle(50)
7 # Stoppen
8 p.stop()
```

Der oben dargestellte Quellcode zeigt, wie man die Pulsweitenmodulation in Python einbindet. In der unteren Zeichnung ist der zeitliche Verlauf der Spannung dargestellt: Bei 50 Hz wird die Zeitspanne, in der der Strom an und abgeschaltet wird 50 mal in der Sekunde wiederholt.

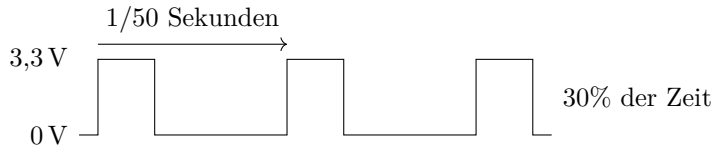


Abbildung 5.7: Zeitlicher Verlauf der Spannung bei der Pulsweitenmodulation



Aufgabe 5.5

a) Schließe eine einzelne LED an den Raspberry Pi an und lasse sie in verschiedenen Helligkeiten aufleuchten.

b) Schließe eine zweite LED an den Raspberry Pi. Beide sollen sich mit der vollen Helligkeit abwechseln. Dabei soll die Helligkeit in 5% Schritten steigen bzw. fallen.

5.6 Ansteuerung des GPIO mit Scratch

Von Scratch aus lassen sich Elemente wie Leuchtdioden, Summer und Schalter am GPIO steuern. Je nach Version von Scratch muss aber unterschiedlich vorgegangen werden. Auch der mögliche Funktionsumfang unterscheidet sich etwas. Daher wird hier zuerst das Vorgehen in Scratch 1.4 erklärt und anschließend in Scratch 2.

5.6.1 Scratch 1.4 und der GPIO

Für die Scratch-Version 1.4 gibt es eine spezielle Anpassung für den Raspberry Pi. Bei dieser gibt es über den Menüpunkt »Bearbeiten« die Möglichkeit den GPIO-Server zu starten. Erst dann zeigen alle nachfolgenden Anweisungen erst ihre Wirkung.

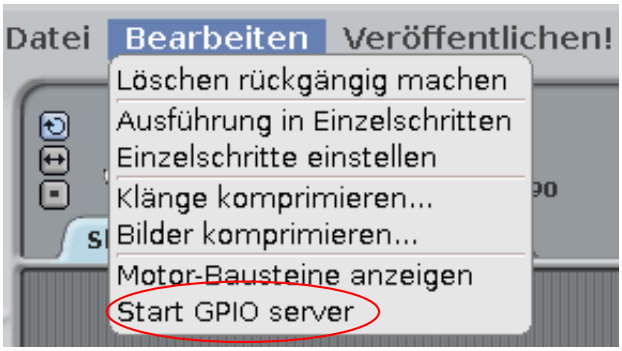


Abbildung 5.8: Starten des GPIO-Servers in Scratch

Ausgänge

Wenn man an den GPIO ein Element angeschlossen hat, dass man von Scratch aus schalten will, so muss zuerst angegeben werden, an welchem Pin etwas angeschlossen ist. Dazu nutzt man die »Senden«-Funktion von Scratch und sendet config<PIN>out an alle, wie in Abbildung 5.9a zu sehen. Hierbei wird PIN durch die Nummer des Pins ersetzt. In Scratch werden dabei die Pins mit ihrer GPIO-Nummer angesprochen.



(a) Als Ausgang setzen



(b) Für 3 Sekunden einschalten

Abbildung 5.9: Pin 22 als Ausgang

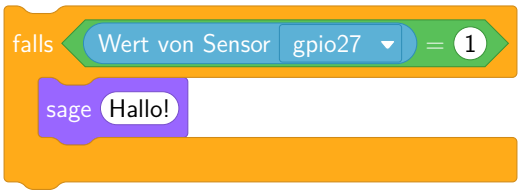
Das Setzen der Pins muss nur ein Mal geschehen. Anschließend kann man über die »Senden«-Funktion mit gpio<PIN>on den Pin anschalten und mit gpio<PIN>off ausschalten. Ein kleines Beispiel dafür ist in Abbildung 5.9b dargestellt.

Eingänge

Mit dem Senden Genauso von config<PIN>in wird, wie bei den Ausgängen, Scratch die Eingänge mitgeteilt. Daraufhin wird der Pin als Sensor aufgeführt. Dieser Sensor liefert dann die Werte 0 oder 1, je nachdem ob der an den Pin angeschlossene Schalter offen oder geschlossen ist.



(a) Als Eingang setzen



(b) Auf Sensor reagieren

Abbildung 5.10: Nutzen eines GPIO-Eingang an Pin 27

Pulsweitenmodulation

Die Ausgänge am GPIO lassen sich nur ein- und ausschalten, was zur Folge hat, dass keine Zwischenwerte möglich sind. Dieses kann man etwas dadurch umgehen, dass die Ausgänge immer nur kurz einschaltet. So etwas wird durch die sogenannte Pulsweitenmodulation erreicht, die man bei einem Eingang durch das Senden von config<PIN>outputpwm einstellen kann.

Dann lässt sich dem Pin ein Wert zwischen 0 und 1024 zuordnen, der angibt, wie häufig der Pin eingeschaltet sein soll. So sendet man gpio<PIN>pwm<WERT> um einen entsprechenden Wer zu setzen. Das Ausschalten funktioniert analog zu den Ausgabenpins mit gpio<PIN>off.



(a) Konfigurieren



(b) Für 3 Sekunden 50% und anschließend 70%

Abbildung 5.11: Pin 24 mit Pulsweitenmodulation

5.6.2 Scratch 2 und der GPIO

Die Version 2 von Scratch ist in vielen Elementen überarbeitet worden und bietet nun bei den Kategorien der Blöcke »Weitere Blöcke« an. Damit gibt es die Möglichkeit eigene Blöcke zu erstellen oder Erweiterungen hinzuzufügen, die neue Blöcke mitbringen. Beim Raspberry Pi lässt sich dort die Erweiterung »GPIO« laden, die zur weiteren Arbeit nötig ist.

Ausgänge

Gegenüber der der Version 1.4 muss man hier nicht vorher angeben, dass ein Pin als Ausgang genutzt werden soll. Es genügt ihn direkt auf output high für ein und output low für aus zu stellen.

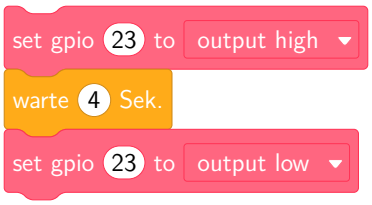
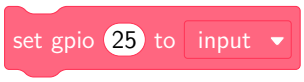


Abbildung 5.12: Pin 23 für 4 Sekunden einschalten

Eingänge

Um einen Pin als Eingang zu nutzten, muss dieses auch in Version 2 zuvor mitgeteilt werden, indem man den Pin auf input setzt. Anschließend kann man über einen Abfrageblock überprüfen, ob am Eingang ein Signal anliegt.



(a) Als Eingang setzen



(b) Auf Sensor reagieren

Abbildung 5.13: Nutzen eines GPIO-Eingang an Pin 25

Weitere Möglichkeiten mit Scratch 2

Die Vorinstallation von Scratch 2 hat auf dem Raspberry Pi auch die Steuerung für den Sense-Hat Aufsatz mit dabei. Die Pulsweitenmodulation wie bei Scratch 1.4 ist aber nicht ohne weiteres möglich. Dafür gibt es aber passende Erweiterung im Internet zu finden, die man erst auf den Raspberry Pi installieren muss. Darüber können auch weitere Möglichkeiten in Scratch eingebunden werden.

5.7 Passiver Infrarotsensor (PIR) als Schalter

Ein Bewegungsmelder, oder kurz PIR von Passive Infrared, ist ein elektrischer Sensor, der auf Temperaturveränderungen in seiner Umgebung reagiert und dieses in der Funktion als Schalter weiter gibt, der für eine Zeitspanne eingeschaltete bleibt.

Die für Raspberry Pi nutzbare Version des PIR hat drei Anschlüsse: Die Spannungsversorgung mit 5 V und die Masse können direkt mit den passenden Pins des GPIO verbunden werden. Über den dritten Anschluss kann der PIR abgefragt werden. Da bei Vorliegen eines Signals nur eine Spannung von 3,3 V gegenüber der Masse herrscht, kann dieser Anschluss direkt mit einem Pin des GPIO verbunden werden. Dieser Pin kann so auf gleiche Weise wie bei einem Taster abgefragt werden.

Die Reihenfolge und Platzierung der Anschlüsse bei dem PIR ist je nach Bauart verschieden. Daher muss man sich an die aufgedruckten Bezeichnungen auf der Platine des PIR halten. Ähnlich sieht es mit den beiden Reglern aus, mit denen sich einstellen lässt, mit welcher Sensitivität auf Veränderungen reagiert wird und wie lange anschließend das Signal am Ausgang anliegt.

5.8 Lichtschranke

Es gibt verschiedene Anwendungen, bei denen durch das Auslösen einer Lichtschranke Aktionen gesteuert werden sollen. Dabei muss der Lichtsensor als Schalter dienen. Eine Art dieses zu realisieren ist die Verwendung eines Phototransistors. Die Beschaltung am GPIO ist der folgenden Schaltskizze zu entnehmen. Dabei liegt das Signal in diesem Fall an PIN 11 an.

Als Gegenstück wird eine Lichtquelle benötigt. Dieses kann auch eine einfache LED sein. In einem solchen Fall ist zu beachten, dass der Abstand zwischen LED und Phototransistor

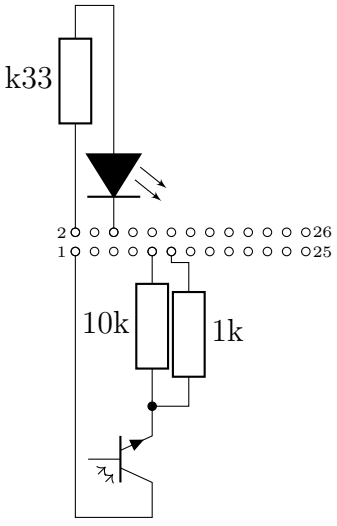


Abbildung 5.14: Lichtschranke aus einer LED und einem Phototransistor

bei maximal etwa einem cm liegen darf. Mit einer sehr hellen weißen LED bei etwa 10 cm. Dafür können diese LEDs aber direkt durch den Raspberry Pi betrieben werden, wie in der oberen Schaltskizze mit angegeben. Beim Testen des maximalen Abstands ist zu beachten, dass der Abstand immer nur verringert werden sollte, bis es ein Signal durch den Schalter gibt.

5.8.1 Abfrage im Programm

Die Lichtschranke kann genau wie andere Schalter angesprochen und verarbeitet werden. Dieses lässt sich auch an folgendem Python-Programm erkennen, mit dem z. B. der maximale Abstand zwischen LED und Phototransistor getestet werden kann.

```
1 import time, RPi.GPIO as GPIO
2
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(11, GPIO.IN)
5
6 while True:
7   input_value = GPIO.input(11)
8   print(input_value)
9   time.sleep(0.1)
```

5.9 Zweizeilendisplay

Die meisten verkauften Zweizeilen Displays besitzen den Controllerchip HD44780. Dadurch verstehen diese einheitliche Steuerbefehle und auch ihr Anschluss durch 16 Pins ist gleich. Neben der Spannungsversorgung und Leitungen für Steuersignale kann man das Display über vier oder acht Datenleitungen ansprechen. In der unten aufgeführten Schaltskizze ist die Variante für vier Bit, also vier Datenleitungen aufgezeichnet.

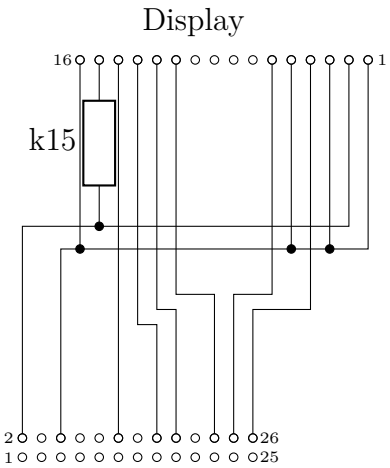


Abbildung 5.15: Displayschaltung

Display	GPIO
1 (GND)	6 (GND)
2 (+5V)	2 (+5V)
3 (Kontrast)	6 (GND)
4 (RS)	26 (GPIO7)
5 (R/W)	6 (GND)
6 (E)	24 (GPIO8)
11 (Daten 4)	22 (GPIO25)
12 (Daten 5)	18 (GPIO24)
13 (Daten 6)	16 (GPIO23)
14 (Daten 7)	12 (GPIO18)
15 (LED +5V)	2 (+5V)
16 (LED GND)	6 (GND)

Tabelle 5.2: Pinbelegung Display

Die Pins 15 und 16 sind für die Hintergrundbeleuchtung wichtig und können bei Bedarf auch nicht belegt werden. Beim Anschluss ist zu beachten, dass auf jeden Fall ein passender Widerstand dazwischen geschaltet wird, da sonst die LED-Beleuchtung zerstört werden kann. Bei der obigen Beschaltung ist der Pin für den Kontrast mit der Masse verbunden, so dass der Kontrast maximal ist. Es ist aber auch möglich über einen Potentiometer den Kontrast regelbar zu machen.

Angesteuert wird das Display über die vier Datenleitungen, bei denen jeder 8-Bit Befehl immer in zwei mal 4-Bit zerlegt werden muss. Mit der Wahl des Registers (RS) wird dabei angegeben, ob es sich um einen Befehl oder Text für das Display handelt. Über die Taktleitung (E) wird dem Display mitgeteilt, dass etwas an den Datenleitungen anliegt. Theoretisch ist es auch möglich das Display auszulesen (R/W). Durch die dann anliegende Spannung von 5 V kann aber der Raspberry Pi zerstört werden.

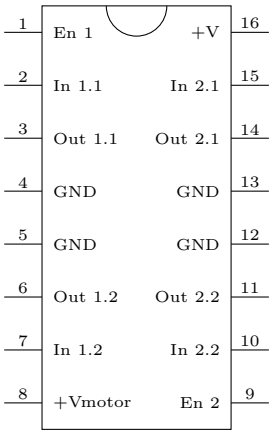
Ist das Display richtig angeschlossen, so wird beim Starten des Raspberrys eine Zeile komplett mit schwarzen Feldern gefüllt. Zum Testen gib es ein Python-Programm unter http://tinyurl.com/ente-display-code. Dieses kann als Grundlage für eigene Programme genutzt werden, die mit dem Display arbeiten sollen.

Weitere Informationen zu den Displays und wie sie angesteuert bzw. angeschlossen werden, ist auf den Seiten http://www.sprut.de/electronic/lcd/ und http://tinyurl.com/ente-display erhältlich.

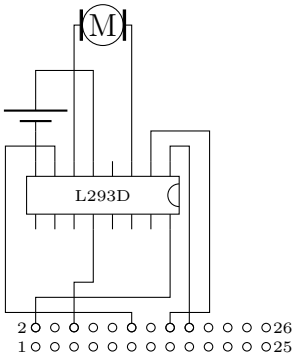
5.10 Ansteuerung von Motoren

Die meisten Motoren benötigen eine höhere Spannung als 3,3 V und damit verbunden einen höheren Strom, als ihn der Raspberry Pi liefern kann. Um trotzdem über den GPIO Motoren ansteuern zu können, benötigt man passende Schaltungen. Der Chip L293D ist

dafür ideal, da er durch die eingebaute Brückenschaltung auch die Möglichkeit bietet, die Laufrichtung des Motors direkt zu beeinflussen.



(a) Beschaltung des Chips



(b) Komplette Schaltung mit einem Motor

Abbildung 5.16: Ansteuerung eines Motors mit dem L293D

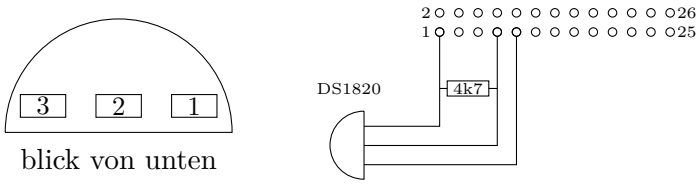
An den Chip können zwei Motoren angeschlossen werden, die unabhängig voneinander gesteuert werden können, aber die gleiche Spannungsversorgung nutzen. Diese Spannungsversorgung für den Motor wird mit dem Pluspol an den Pin 8 angeschlossen. Die Masse wird, genauso wie die Masse vom Raspberry, an einen der Pins 4,5,12 oder 13 angeschlossen. Außerdem wird zur Spannungsversorgung Pin 16 mit dem 5 V-Pol des Raspberry verbunden.

Der erste Motor wird durch die Ausgänge gesteuert, die sich Out 1.1 und Out 1.2 nennen. Dann kann durch Anlegen einer Spannung an In 1.1 bzw. In 1.2 die Laufrichtung gesteuert werden. Dabei ist zu beachten, dass nur jeweils einer der beiden Eingänge geschaltet werden darf. Unabhängig von der Laufrichtung muss auch der Enable-Eingang En 1 geschaltet sein, damit der Motor sich bewegt. Der Anschluss des zweiten Motors erfolgt nach dem gleichen Prinzip an die anderen Anschlüsse.

Wird der Chip und der Motor wie in der nebenstehenden Darstellung beschaltet, so kann mit den Pins 12 und 16 die Laufrichtung des Motors gesteuert werden. Erst wenn zu einem der beiden Pins auch der Pin 18 belegt ist, wird sich dann der Motor in Bewegung setzen.

5.11 Digitales Thermometer

Möchte man mit dem Raspberry Pi die Temperatur messen, so eignet sich dafür der Thermometerchip DS1820. Dieser besitzt drei Anschlüsse: Eins ist die Masse, zwei ist die Datenleitung und über drei erfolgt die Versorgung des Chips mit 3, 3 V.



(a) Pins des DS1820

(b) Anschluss am Raspberry Pi

Abbildung 5.17: Thermometerchip DS1820 am Raspberry Pi

Für die Nutzung mit dem Raspberry Pi gibt es schon passende Treiber. Dazu müssen aber die Versorgung und die Datenleitung mit einem $4,7\,\mathrm{k\Omega}$ Widerstand verbunden sein und die Datenleitung mit dem Pin mit der Bezeichnung 4, was der Pin 7 auf dem Board ist. Will man mehrere dieser Thermometerchips anschließen, so müssen diese nur parallel zu den anderen Chips angeschlossen werden.

5.11.1 Treiber laden

Bevor die Treiber geladen werden können, muss der »Device Tree Overlay« aktiviert werden. Dazu muss, mit Root-Rechten in die Datei /boot/config.txt folgende Zeile ergänzt werden:

```
dtoverlay=w1-gpio,gpiopin=4,pullup=on
```

Unter Linux werden Treiber als Kernelmodule geladen. Für den DS1820 sind zwei solcher Module notwendig: w1-gpio und w1-therm. Diese können als Superuser über den Befehl modprobe geladen werden. Beim ersten muss aber durch die besondere Nutzung mit dem Widerstand noch der Zusatz pullup=1mit angegeben werden. Daraus ergibt sich also gesamt:

```
sudo modprobe w1-gpio pullup=1
sudo modprobe w1-therm
```

Will man die Treiber bei jedem Systemstart automatisch geladen haben, muss man den Inhalt der Datei /etc/modules um folgende Zeilen ergänzen:

```
w1-gpio pullup=1
w1-therm
```

5.11.2 Temperatur auslesen

Sind Thermometer angeschlossen und die Treiber geladen, so findet man im Verzeichnis /sys/bus/w1/devices/ Unterverzeichnisse die jeweils ein Thermometer repräsentieren und

deren Namen sich aus der jeweiligen Seriennummer zusammensetzt. Innerhalb solcher Verzeichnisse kann man die Datei w1_slave finden, die man mit cat w1_slave ausgeben kann. Die letzte Zahl der zweiten Zeile ist die Temperatur multipliziert mit dem Faktor 1000.

5.11.3 Temperatur auslesen mit Python

Um die gemessene Temperatur auch in einem Programm nutzen zu können, muss die Datei ausgelesen und die passende Stelle extrahiert werden. Dieses kann z. B. mit dem folgendem Python-Code geschehen. Dazu muss der Dateiname noch dem eigenen Thermometer angepasst werden. Nähere Informationen dazu findet man unter anderem auf der Seite http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/temperature/.

```
1 tfile = open("/sys/bus/w1/devices/10-000802824e58/w1_slave")
2 text = tfile.read()
3 tfile.close()
4 secondline = text.split("\n")[1]
5 temperaturedata = secondline.split(" ")[9]
6 temperature = float(temperaturedata[2:])
7 temperature = temperature / 1000
8 print(temperature)
```

5.12 Analog/Digital-Wandler (AD)

Von Hause aus kann der Raspberry Pi nur digitale Signale verarbeiten. Er kann also nur überprüfen, ob an einem Pin eine Spannung anliegt oder nicht. Dieses wird bei dem Gebrauch von Schaltern eingesetzt. Um analoge Messungen auswerten zu können, bei denen es auch auch Zwischenwerte geht, wird ein Analog/Digital-Wandler, oder auch kurz AD, benötigt. Dieser misst eine Spannung und wandelt sie in eine Zahl in einem bestimmten Zahlenbereich um. Der MCP3008 ist ein solcher 10-Bit A/D-Wandler. Er kann Zahlenwerte zwischen 0 und $2^{10} - 1 = 1023$ erfassen. Der MCP3208 hat sogar eine Auflösung von 12 Bit und liefert somit Werte von 0 bis $2^{12} - 1 = 4095$.

5.12.1 SPI einrichten

Damit der A/D-Wandler nicht mit 10 Leitungen angeschlossen werden muss, kommuniziert er mit dem Raspberry Pi über das SPI-Protokoll. Dazu hat der Pi vorbelegte Pins die im Device-Tree aktiviert werden müssen. Dieses kann in raspi-config unter »Advanced Options« eingestellt werden oder durch hinzufügen folgender Zeile in die /boot/config.txt:

```
dtparam=spi=on
```

Nach einem Neustart des Raspberry Pi steht dann das SPI zur Verfügung.

5.12.2 Anschluss des A/D-Wandlers

Im ersten Schritt wird nur der A/D-Wandler an den Raspberry Pi angeschlossen. Die Anschlüsse des MCP3008 sind identisch zum MCP3208 und in der linken unteren Zeichnung abgebildet. Wie dieser A/D-Wandler mit dem Raspberry Pi verbunden werden muss, ist der der Schaltzeichnung und der rechten Tabelle zu entnehmen. In dieser Aufstellung ist bereits die Referenzspannung mit angeschlossen, weshalb in dieser Anordnung die zu messende Spannung niemals 3,3 V überschreiten darf.

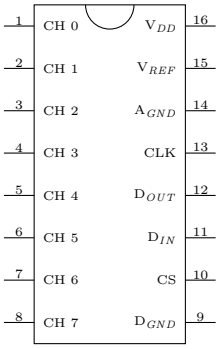


Abbildung 5.18
MCP3208

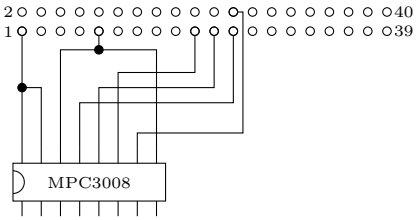


Abbildung 5.19
MCP3208 am Raspberry Pi

A/D	GPIO
9 (D_{GND})	9 (GND)
10 (CS)	24 (CE0)
11 (D_{IN})	19 (MOSI)
12 (D_{OUT})	21 (MISO)
13 (CLK)	23 (SCLK)
14 (A_{GND})	9 (GND)
15 (V_{REF})	1 (3,3 V)
16 (V_{DD})	1 (3,3 V)

Tabelle 5.3
Pinbelegung des ADs am Raspberry Pi

5.12.3 Einbindung in Python

Die Einbindung des MCP3008 in Python ist durch die Python Bibliothek »gpiozero«[1] sehr einfach. Es muss nur bei der Erzeugung des A/D-Wandler Objekts der Kanal angegeben werden, von dem der Wert ausgelesen werden soll.

```
1 import gpiozero
2 adw = gpiozero.MCP3008(channel=3)
3 value = adw.value # liefert Wert von 0 bis 1
4 value2 = adw.raw_value # liefert Wert von 0 bis 1023
```

Bei dem oberen Beispiel ist der Unterschied zwischen value und raw_value zu beachten. Beim ersten wird der des A/D-Wandlers so umgerechnet, dass er immer zwischen 0 und 1 liegt. Beim zweiten liegt der Wert direkte Wert vor, der beim MCP3008 bis 1023 geht, beim MCP3208 aber bis 4095.

[1] Diese Bibliothek kann ggf. mit sudo apt-get install python3-gpiozero nachinstalliert werden

5.12.4 Anschluss eines Potentiometers

Das Potentiometer ist ein regelbarer Widerstand. Schließt man ihn zwischen der Masse und 3,3 V an, so erhält man zwischen Masse und dem Anschluss für den regelbaren Widerstand eine Spannung zwischen 0 V und 3,3 V, je nachdem wie weit das Potentiometer eingestellt ist. Mit Hilfe des A/D-Wandlers kann man so aus einem Potentiometer einen Regler für Programme machen.

Der Anschluss eines Potentiometers ist der Schaltzeichnung zu entnehmen. In diesem Fall ist er an Kanal 0 angeschlossen, es wäre aber auch jeder andere der 8 Kanäle möglich. Entsprechend sieht auch das dazugehörige Python-Programm aus.

```
1 import time, gpiozero
2
3 adc = gpiozero.MCP3008(channel=0)
4
5 while True:
6   wert = int(adc.value * 100)
7   print('Der Poti steht auf', wert, '%')
8   time.sleep(1)
```

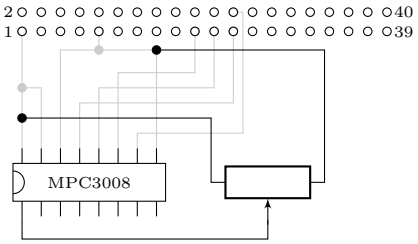
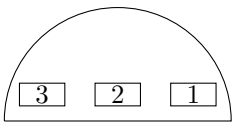


Abb. 5.20: Regelbarer Widerstand am AD-Wandler

5.12.5 Anschluss eines analogen Thermometers

Ein analoger Thermometerchip hat am Datenpin je nach gemessener Temperatur eine unterschiedliche Spannung. Für den Einsatz am Raspberry Pi eigenen sich dabei Thermometer aus der TMP36-Reihe. Diese haben folgende Belegung: Pin 1 ist für die Spannung, Pin 2 liefert die Daten und an Pin 3 muss die Masse angeschlossen werden.



blick von unten

Abb. 5.21: Analoges Thermometer TMP36

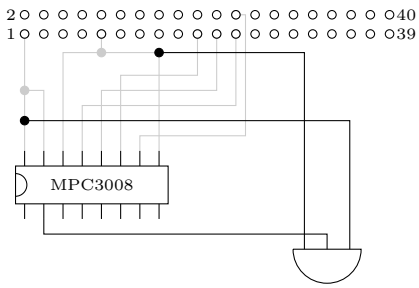


Abb. 5.22: Analoges Thermometer am AD-Wandler

Dem Schaltplan ist zu entnehmen, wie das Thermometer an den Kanal 1 angeschlossen wurde. Um aus dem gemessenen Wert die Temperatur bestimmen zu können, muss dieser passend umgerechnet werden. Beim TMP36 kann man folgende Formel dafür nutzen:

$$t = (v \cdot 3,3 - 0.5) \cdot 100$$

Dabei ist v der gemessene Wert zwischen 0 und 1 und die $3,3$ kommt durch die Referenzspannung am A/D-Wandler.

Beim Auslesen ist zu beachten, dass die gemessenen Werte des TMP36 bereits bei der Messung um 0.5° vom echten Wert abweichen können. Daher sollte man die errechneten Werte im Programm auf maximal eine Nachkommastelle runden. Das komplette Python-Programm zum auslesen des Thermometers sieht dann so aus:

```
1 import time, gpiozero
2
3 adc = gpiozero.MCP3008(channel=1)
4
5 while True:
6   temp = (adc.value * 3.3 -0.5) * 100
7   temp = round(temp, 1)
8   print('Die␣Temperatur␣ist', temp, 'C')
9   time.sleep(1)
```

Kapitel 6

Software

6.1 LaTeX auf dem Raspberry Pi

Die Textsatzbeschreibungssprache LaTeX ist eine verbreitete Möglichkeit Texte zu schreiben und gleichzeitig Anweisungen zu geben, wie dieser Text gesetzt werden soll. Auch dieses Skript ist mit Hilfe von LaTeX geschrieben worden. LaTeX wird durch verschiedene Programme auf fast allen Betriebssystemen unterstützt und benötigt neben dem Übersetzungsprogramm nur einen einfachen Texteditor und einen Betrachter für PDF-Dokumente.

6.1.1 Installieren von LaTeX

Auf dem Raspberry Pi kann das System TeXLive installiert werden, dass alle nötigen Tools zur Verfügung stellt. Installiert wird es mit folgendem Befehl:[1]

```
sudo apt-get install texlive texlive-lang-german
```

Neben vielen verschiedenen Paketen, mit denen unterschiedliche Funktionen unterstützt werden, ist es in der Regel hilfreich noch zwei weitere Pakete zu installieren:

```
sudo apt-get install texlive-generic-extra texlive-latex-extra
```

6.1.2 Nutzung von LaTeX

Um Texte für LaTeX zu schreiben reicht der einfache Texteditor Leafpad aus, der in der Regel schon installiert ist. Andere Editoren, wie z. B. Gummi können auch eingesetzt werden und bieten zusätzliche Extras.

Ein Einfacher Text wird damit wie folgt geschrieben, der z. B. in der Datei meintext.tex abgespeichert werden kann.

[1] Die Raspbian-Quellen sind stabil aber nicht immer die neusten. Eine Installation mit aktuellen Quellen ist unter 6.1.4 zu finden.

```
1 \documentclass[parskip=half]{scrartcl}
2 \usepackage{ngerman}
3 \usepackage[utf8]{inputenc}
4
5 \begin{document}
6
7 \section{Überschrift erster Ebene}
8
9 Ganz einfacher Text. Ganz normal herunter geschrieben.
10
11 \end{document}
```

Die Übersetzung des Textes in ein PDF-Dokument erfolgt dann mit dem Befehl:

```
pdflatex meintext.tex
```

Installiert ist in der Regel der PDF-Betrachter xpdf, mit dem man sich anschließend das Werk anschauen kann.

6.1.3 Weitere Möglichkeiten

Mit dem oben aufgezeigten Text sind die Möglichkeiten von LaTeX bei weitem nicht deutlich. Weitere kleine Beispiele gibt es auf verschiedenen Seiten im Internet. Zu empfehlen ist dabei unter anderem das LaTeX-Kochbuch unter http://archiv.dante.de/TeX-Service/TSP/tex/cookbook/cookbook.html oder der entsprechende Wikipedia-Artikel mit entsprechenden Verweisen zu anderen Seiten unter http://de.wikipedia.org/wiki/LaTeX.



Aufgabe 6.1

a) Installiere das Grundsystem von TeXlLive

b) Schreibe deinen eigenen kleinen Text und lasse ihn in ein PDF-Dokument übersetzen.

c) Verfasse einen Text darüber, was du mit dem Raspberry bisher alles gemacht hast und was du noch vorhast mit ihm durchzuführen. Setze dabei auch weitere Möglichkeiten von LaTeX ein.

6.1.4 Installation direkt aus den TeXLive-Quellen

Fast täglich gibt es Ergänzungen und Verbesserungen bei den LaTeX-Paketen. Die Quellen von Debian, auf dem das Raspbian aufsetzt, haben aber einen eingefrorenen Zustand, bis eine neue Version von Debian veröffentlicht wird. Zwar werden Sicherheitsupdates eingepflegt, aber diese betreffen eher die Programmteile von TeXlLive und nicht die

Pakete zum setzen. Aus diesen Gründen kann es sinnvoll sein, direkt auf die TeXlLive-Quellen zu setzen.

Ist bereits eine Installation aus den Raspbian-Quellen erfolgt, so sollte diese entfernt werden. Dazu dienen folgende Aufrufe, die zuerst die wichtigen Pakete entfernen und im zweiten Schritt die verbliebenen Pakete denen dann die Grundlage fehlt.

```
sudo apt-get remove texlive
sudo apt-get autoremove
```

Für die Installation von TeXlLive für den Raspberry Pi wir der Netzwerkinstaller benötigt, der folgendermaßen heruntergeladen und anschließen entpackt wird:

```
wget mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
tar -xf install-tl-unx.tar.gz
```

Entpackt wird eine Verzeichnis install-tl-XXXX, wobei der letzte Teil einem Datum entspricht. In dieses Verzeichnis muss man wechseln und das eigentliche Installationsprogramm starten:

```
sudo ./install-tl
```

Bevorzugt man eine graphische Variante und hat das Paket perl-tk auf dem Raspberry Pi installiert, so kann man auch folgendes aufrufen:

```
sudo ./install-tl gui
```

In beiden Fällen empfiehlt es sich für den Anfang das Installationsschema bzw. der Installationsumfang »scheme-small« zu wählen. Fehlende benötigte Teile können anschließend immer noch der Startinstallation hinzugefügt werden. Die anschließende Installationsroutine wird einige Zeit benötigen, da die weiteren Daten aus dem Internet erst heruntergeladen werden.

Nach der Installation ist noch die Datei /etc/profile.d/texlive.sh anzulegen mit dem folgenden Inhalt. Dieses ist nötig, damit man die Programme wie pdflatex direkt aufrufen kann. Das 20xx muss dabei entsprechend der lokalen Installation angepasst werden.

```
1 export PATH=/usr/local/texlive/20xx/bin/armhf-linux:$PATH
2 export MANPATH=/usr/local/texlive/20xx/texmf-dist/doc/man:$MANPATH
3 export INFOPATH=/usr/local/texlive/20xx/texmf-dist/doc/info:
    $INFOPATH
```

Bis zum nächsten Neustart des Raspberry Pi muss man in jeder einzelnen Shell die obigen Einträge per Hand nachladen:

```
source /etc/profile
```

Danach lässt sich TeXLive über den eigenen Manager tlmgr aktualisieren und erweitern. Mit einer graphischen Oberfläche geschieht dieses über:

```
sudo su -
tlmgr gui
```

Die reine Aktualisierung kann erfolgen durch:

```
sudo su -
tlmgr update --self
tlmgr update --list
tlmgr update --all
```

Will man nun andere Programme über den Raspbian Paketmanager installieren, die ein LaTeX-System voraussetzen, so fehlen bei dieser Variante die entsprechenden Einträge. Möchte man diese nachziehen sollte man der Anleitung unter http://tinyurl.com/texliveinstallation im Ubuntu-Wiki folgen.

6.2 Verschlüsselung mit GnuPG

Die Standardverfahren zur Verschlüsselung von Dateien und E-Mails basieren auf einem Schlüsselpaar aus öffentlichem und privaten Schlüssel. Die gängigste Variante, besonders im E-Mail-Bereich, ist das freie Kryptografiesystems GNU Privacy Guard, kurz GnuPG. Das dabei verwendet Verfahren basiert auf »Pretty Good Privacy« von Phil Zimmermann, was auf deutsch »ziemlich gute Privatsphäre« bedeutet.

Bei diesem Verfahren können mit dem öffentlichen Schlüssel verschlüsselte Daten nur mit dem privaten Schlüssel wieder entschlüsselt werden. Um dieses System zur sicheren Kommunikation einsetzen zu können verschlüsselt man die Nachricht mit dem öffentlichen Schlüssel des Empfängers. Damit möglichst viele an diesen Teil des Schlüsselpaars kommen, kann man diesen auf einem Schlüsselserver deponieren. Insgesamt ist das System mit einem Vorhängeschloss zu vergleichen, das sich ohne Schlüssel zudrücken lässt aber nur mit dem passenden Schlüssel öffnen. Hierbei ist das Vorhängeschloss der öffentliche Schlüssel.

Bei der PGP-Technik lässt sich der Weg aber auch umdrehen. So lassen sich Elemente, die mit dem privaten Schlüssel verschlüsselt wurden mit dem öffentlichen wieder entschlüsseln. Auf diese Weise ist es möglich, Dinge zu unterschreiben, wenn man davon ausgeht, dass der private Schlüssel nur der richtigen Person bekannt ist.

6.2.1 Installation und Einrichtung von GnuPG

Der Paketname für GnuPG lautet auf dem Raspberry »gnupg2« da man bei der Kryptographie oft auch Zufall benötigt, sollte man direkt das Paket »haveged« mit installieren, so dass der Aufruf zur Installation dann wie folgt aussieht:

```
sudo apt install gnupg2 haveged
```

Dieses beinhaltet nur die reine Konsolenanwendung, die in den nächsten Schritten behandelt wird. Mit Programmen wie GPA (Paketname: gpa) oder Seahorse (Paketname: seahorse) gibt es Möglichkeiten einer graphischen Oberfläche.

Erzeugung eines Schlüssels

Um mit GnuPG überhaupt arbeiten zu können, benötigt man ein passendes Schlüsselpaar. Die Erzeugung funktioniert sehr einfach über die Eingabe von:

```
gpg --gen-key
```

Anschließend wird man nach dem Namen und seiner E-Mail-Adresse gefragt. Über den Namen und die E-Mail-Adresse wird es anderen Nutzern ermöglicht, den öffentlichen Schlüssel einfach zu finden. Dieses ist wichtig, damit diese einem auch verschlüsselte Nachrichten zuschicken können.

Hat man die Daten korrekt eingegeben, so kann man zur eigentlichen Schlüsselerzeugung kommen, nachdem man eine sogenannte Passphrase angegeben hat. Dabei handelt es sich um ein Passwort, dass später für alle Aktionen mit dem privaten Schlüssel benötigt wird. Es sollte nicht leicht zu erraten sein und bereits vom Namen her etwas länger sein, als die normal üblichen Passwörter.

Der erzeugte Schlüssel entspricht dann den Standardwerten und hat eine Gültigkeit von zwei Jahren. Will man detailiertere Einstellungen treffen, so kann man den Schlüssel auch mit folgendem Aufruf erzeugen:

```
gpg --full-generate-key
```

Bei der ersten Auswahl sollte man die Voreinstellung »RSA und RSA« wählen und als Länge wird heutzutage 3072 Bit oder mehr vorgeschlagen.

```
Bitte wählen Sie, welche Art von Schlüssel Sie möchten:
    (1) RSA und RSA (voreingestellt)
    (2) DSA und Elgamal
    (3) DSA (nur unterschreiben/beglaubigen)
    (4) RSA (nur signieren/beglaubigen)
Ihre Auswahl? 1
RSA-Schlüssel können zwischen 1024 und 4096 Bit lang sein.
Welche Schlüssellänge wünschen Sie? (3072) 3072
```

Eine weitere wichtige Einstellung ist die Gültigkeitsdauer des Schlüssels. Diese sollte nicht zu lange sein, für den Fall dass der Schlüssel einmal verloren geht. Da sich ein Schlüssel ohne Probleme verlängern lässt, ist ein guter Vorschlag der Zeitraum von einem Jahr.

```
Bitte wählen Sie, wie lange der Schlüssel gültig bleiben soll.
                0 = Schlüssel verfällt nie
              <n>  = Schlüssel verfällt nach n Tagen
              <n>w = Schlüssel verfällt nach n Wochen
              <n>m = Schlüssel verfällt nach n Monaten
              <n>y = Schlüssel verfällt nach n Jahren
Wie lange bleibt der Schlüssel gültig? (0) 1y
```

Bei der nächsten Abfrage muss man zum Namen und seiner E-Mail-Adresse auch einem möglichen Kommentar angeben. Der Kommentar ist optional und dient dazu, seine Adresse genauer zu spezifizieren, dass sie z. B. für die Arbeit genutzt wird.

Verbreiten eines Schlüssels und importieren anderer öffentlicher Schlüssel

Die einfachste Möglichkeit, seinen eigenen öffentlichen Schlüssel zu verbreiten ist die, ihn auf einen der Key-Server hochladen. Dort kann ihn jeder suchen und herunterladen, genauso wie man auch selber dort nach anderen öffentlichen Schlüssel suchen kann. Das Hochladen funktioniert mit:

```
gpg --send-keys <ID des Schlüssels>
```

Hierbei wird der auf dem Pi eingestellt Server (keys.gnupg.net) verwendet. Möchte man einen anderen Key-Server nutzen, so kann man dieses mit dem Parameter --keyserver einstellen.

Um an die ID des eigenen Schlüssels zu kommen, muss man sich alle bekannten Schlüssel anzeigen lassen. Dabei lässt sich die Anzeige aber über ein optionales Suchwort einschränken:

```
gpg -k [<Suchwort>]
```

Wenn man nach einem öffentlichen Schlüssel einer anderen Person sucht, muss man ggf. zwischen mehreren auswählen und kann ihn dann direkt in den eigenen Schlüsselbund mit aufnehmen. Der Aufruf ist wie folgt:

```
gpg --search-keys <Bezeichnung>
```

Da sich Schlüssel auch verändern können, weil sie z. B. verlängert wurden, sollte man von Zeit zu Zeit seinen Schlüsselbund mit dem Daten vom Key-Server aktualisieren. Dieses geht am einfachsten über den folgenden Aufruf:

```
gpg --refresh-keys
```

Verschlüsseln und Entschlüsseln von Dateien

Den erzeugten Schlüssel und die öffentlichen Schlüssel anderer Personen können genutzt werden, um Dateien vor der Übermittlung passend zu verschlüsseln. Natürlich ist es aber auch möglich, Dateien zur Sicherheit für sich selber zu verschlüsseln, so dass andere darauf keinen Zugriff haben. Die Verschlüsselung einer Datei funktioniert mit folgendem Aufruf:

```
gpg --encrypt --recipient <Empfänger ID oder E-Mail-Adresse> <
    Dateiname>
```

In diesem Fall wird eine verschlüsselte Kopie der Datei angelegt, die die Endung .gpg hat. Verschlüsselt man so die Datei für eine andere Person, so kann man sie selber nicht wieder entschlüsseln. Daher sollte man ggf. auch seine eigene ID mit angeben. Dieses ist durch die wiederholte Angabe von -- recipient möglich. Als Beispiel dient folgender Aufruf:

```
gpg --encrypt --recipient ich@ich.de --recipient du@du.de
    botschaft.txt
```

Die dadurch erzeugte Datei botschaft.txt.gpg kann nur noch von ich@ich.de und du@du.de entschlüsselt werden.

Die Entschlüsselung verläuft auf sehr ähnlichem Wege. Dabei wird man nach seiner Passphrase gefragt und standardmäßig das Ergebnis der Entschlüsselung in der Konsole angegeben. Beim folgenden Aufruf die Datei botschaft.txt.gpg entschlüsselt und in die Datei botschaft.txt geschrieben:

```
gpg --decrypt --output botschaft.txt botschaft.txt.gpg
```

Weitere E-Mail-Adressen

Sehr oft kommt es vor, dass eine Person auch mehrere E-Mail-Adressen hat. Um nicht für jede Adresse eigenes Schlüsselpaar zu erzeugen und die entsprechende Passphrase merken zu müssen, kann man auch einem Schlüssel weitere Identitäten Hinzufügen. Dazu muss man den Schlüssel editieren:

```
gpg --edit-key <ID oder E-Mail-Adresse>
```

In dem dann erscheinen Promt lässt sich mit Hilfe von adduid der bekannte Dialog für die Identität aufrufen. Dieses kann auch mehrfach durchgeführt werden. Um anschließend die Änderungen am Schlüsselpaar zu speichern und das Programm zu verlassen genügt der save Aufruf.

6.2.2 Einrichtung von GnuPG in Claws Mail

Standardmäßig ist das E-Mail-Programm Claws Mail auf dem Raspberry Pi installiert, dessen Einrichtung in 3.3.3 beschrieben ist. Um mit Claws Mail E-Mails verschlüsseln zu können, werden zwei Plugins benötigt, die zuerst mit folgendem Befehl installiert werden müssen:

```
do apt install claws-mail-pgpinline claws-mail-pgpmime
```

Nach der Installation auf dem System müssen die Plugins in Claws Mail noch aktiviert werden, was über Konfiguration → Erweiterungen/Plugins geschieht.

Nach dieser Aktivierung lassen sich bei jedem eingerichtet E-Mail-Konto in den Einstellungen die Punkte Datenschutz und GPG finden. Dort lässt sich auswählen, ob E-Mails automatisch verschlüsselt werden sollen. Genauso lässt sich auch das Signieren und das verschlüsselte Speichern der E-Mails dort standardmäßig aktivieren. Diese Voreinstellungen lasen sich aber für jede E-Mail individuell ändern.

Verschlüsselter Versand einer E-Mails

Wurde GnuPG für Claws Mail eingerichtet und der verschlüsselte Versand aktiviert, so wird man beim Absenden der E-Mail immer nach dem Schlüssel des Empfängers gefragt, wenn dieser nicht bekannt ist oder nicht richtig zugeordnet werden kann. Dadurch bekommt man aber auch die Möglichkeit, z. B. beim Fehlen eines Schlüssels für den Empfänger, die Nachricht unverschlüsselt zu versenden. Anschließend an diese mögliche Aktion wird man aufgefordert die Passphrase anzugeben, wenn man die E-Mail verschlüsselt oder nur signiert versenden will.

Anhang A

Alternative Bootmöglichkeiten

Mit dem Raspberry Pi 3 wurde die Möglichkeit eingefügt, das System nicht nur von der SD-Karte zu booten. Bei dieser Version wurde ein neuer Chip verwendet, der nicht nur schneller ist als die Vorgänger, sondern auch einen kleinen Speicher hat, indem die Booteinstellungen abgelegt werden können. So lässt sich der Raspberry Pi nun auch über USB oder Netzwerk booten.

A.1 Booten über USB

Um das Booten über USB zu aktivieren muss die Datei /boot/config.txt um folgende Zeile ergänzt werden:

```
program_usb_boot_mode=1
```

Beim nächsten Neustart des Raspberry Pi wird dann der entsprechende Eintrag in den Speicher des Chips übertragen. Das Ergebnis lässt sich einfach mit Hilfe von

```
vcgencmd otp_dump | grep 17:
```

überprüfen. Liefert dieses die Ausgabe 17:3020000a, so ist das Booten über USB aktiviert. Eine Ausgabe von 17:1020000a könnte darauf hindeuten, dass die Firmware nicht aktuell genug ist. Dieses lässt sich durch den Aufruf von rpi-update beheben.

Als USB-Medium kann theoretisch sowohl ein USB-Stick als auch eine USB-Festplatte herangezogen werden. In der Praxis zeigt sich aber, dass nicht alle Modelle funktionieren und entsprechend ausprobiert werden muss. Das USB-Medium muss dazu aber auch mit den entsprechenden Daten bespielt werden.

Das Bespielen kann analog zur SD-Karte mit einem Image geschehen, wie es in 2.1 beschrieben ist. Anschließend müssen aber Dateien auf dem Medium angepasst werden: Auf der Bootpartition muss in der Datei cmdline.txt der Eintrag root=/dev/mmcblk0p2 in root=/dev/sda2 geändert werden. Zusätzlich muss in der Hauptpartition in der Datei etc/fstab an zwei Stellen aus /dev/mmcblk0p ein /dev/sda werden. Die Nummer dahinter

muss aber weiterhin die 1 für /boot und die 2 für / sein. Bei der Verwendung des Image ist zu beachten, dass das Expandieren des Dateisystems auf die volle Größe des Mediums nicht mit den Funktionen analog zur SD-Karte gemacht werden kann. Hier kann z. B. das Programm gparted genutzt werden. Dabei darf dass USB-Medium aber nicht gemountet sein.

Alternativ kann man auch auf das das USB-Medium eine komplette Kopie des bestehenden Systems aufspielen. Dazu ist es an den von der SD-Karte gestarteten Raspberry Pi angeschlossen werden. In der Regel ist es dann als /dev/sda eingebunden. Zuerst muss dazu das Medium passend partitioniert werden. Dieses geschieht mit:

```
sudo parted /dev/sda
```

Innerhalb von parted geht man wie folgt vor:

```
mktable msdos
mkpart primary fat32 0% 100M
mkpart primary ext4 100M 100%
quit
```

Nach einem Reboot müssen die passenden Dateisystem formatiert werden. Anschließend im passenden Verzeichnis gemountet und die Daten auf das Medium überspielt werden. Dieses geht mit folgenden Befehlen:

```
sudo mkfs.vfat -n BOOT -F 32 /dev/sda1
sudo mkfs.ext4 /dev/sda2

sudo mkdir /mnt/usbstick
sudo mount /dev/sda2 /mnt/usbstick/

sudo mkdir /mnt/usbstick/boot
sudo mount /dev/sda1 /mnt/usbstick/boot/

sudo rsync -ax --progress / /boot /mnt/usbstick
```

Auch in diesem Fall muss, wie bei der Nutzung des Images, die config.txt und die fstab angepasst werden.

Um den Start vom USB-Medium dann durchzuführen muss die SD-Karte aus dem Raspberry Pi entfernt werden. Da auch weiterhin zuerst nach einer bootfähigen SD-Karte gesucht wird, ist der Start des Systems über USB entsprechend verzögert. Etwas lässt sich dieser Vorgang dadurch beschleunigen, indem man eine leere SD-Karte in den passenden Slot packt.

Anhang B

Displays

B.1 Installation des offiziellen Raspberry Pi Displays

Seit Mitte 2015 gibt es von der Raspberry Pi Foundation ein Display, dass sich direkt an den Displayport des Raspberry Pi anschließen lässt. Hat man eine aktuelle Version des Systems (siehe 3.2.6), wird das Display auch direkt angesteuert, so dass die Ausgabe darüber erfolgt.

Startet man das System mit Noobs (siehe 2.2), so wird das Display nicht angesteuert. Hier muss die weitere Installation des Systems zuerst über einen normalen Bildschirm erfolgen.

Bei der Benutzung einiger Halterungen für das Display kommt es dazu, dass das Bild auf dem Kopf steht. Dieses ist dadurch bedingt, dass in den ersten Versionen der Software das Display entsprechend gedreht war und sich die Hersteller daran angepasst haben. Es wurde aber festgestellt, dass man so aber den falschen Blickwinkel auf das Display hatte und daher wurde die Einstellung geändert. Möchte man diese Verbesserung wieder rückgängig machen um das Gehäuse entsprechend nutzen zu können muss man lcd_rotate =2 in die Datei /boot/config.txt einfügen (vgl. (Raspberry Pi Forum 2015)).

B.2 Installation des Watterott-Displays

Wesentlich länger als das offizielle Display, gibt es von Watterott ein kleineres 2,8-Zoll Display (http://bit.ly/2jpvcX4), das entweder mit einem Adapter oder direkt auf den GPIO gesteckt wird. Um es in Betrieb zu nehmen, muss nach dem Starten des Systems ein Skript heruntergeladen werden mit:

```
wget -N https://github.com/watterott/RPi-Display/raw/master/rpi-
    display.sh
```

Die weitere Installation erfolgt dann geführt mit dem Aufruf:

```
sudo /bin/bash rpi-display.sh 270
```

Dabei werden zuerst nur die wichtigsten Komponenten für den Kernel installiert, so dass am Ende neu gestartet werden muss, damit die weiteren Funktionen überhaupt durchgeführt werden können. Nach dem Neustart muss das Skript erneut aufgerufen werden. Dabei ist zu beachten, dass man den „Framebuffer Copy“ nicht installieren sollte, wenn man die Anzeige ausschließlich über das kleine Display haben will. Andernfalls wird dort das Bild des HDMI-Ausgangs in klein dargestellt. Sollte man dennoch in installiert und aktiviert haben, kann man dieses rückgängig machen mit dem folgenden Aufruf. Ggf. ist es dazu nötig erst mit Alt+Strg+F2 in ein anderes Fenster zu wechseln, damit man sich anmelden kann.

```
sudo update-rc.d fbcp remove
```

Durch den Aufruf des Skripts kann man auch die Einstellungen ändern, so dass man die Konsole wieder über den HDMI-Ausgang angezeigt bekommt.

Anhang C

Erweitertes WLAN

In der Regel sind fast alle Netzwerke über die WPA2-Personal abgesichert. In diesem Fall reicht es in der Regel aus, in der graphischen Oberfläche in der Statusleiste das Netzwerk auszuwählen und das Passwort anzugeben, wie im Abschnitt 3.3.1 angegeben ist. Will man aber mehrere dieser Netzwerke regelmäßig nutzen oder auf graphische Oberfläche verzichten, so lassen sich die Einstellungen als WLAN-Client auch dauerhaft in einer Konfiguration ablegen.

Diese Form der Konfiguration ist auch für Netzwerke nötig, bei denen jeder Benutzer eigenen Zugangsdaten hat. Dieses ist z. B. in den Universitäten der Fall oder Schulen, die das WLAN für ihrer Schüler bereitstellen. Dabei handelt es sich in den meisten Fällen um eine WPA2-Enterprise Lösung.

In beiden Fällen müssen zwei Dateien ergänzt werden. Die /etc/network/interfaces sollte dann folgende drei Zeilen enthalten, wobei der Name der WLAN-Karte gegebenenfalls angepasst werden muss:

```
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Die Ergänzungen in /etc/wpa_supplicant/wpa_supplicant.conf sind davon abhängig, welcher Typ von WLAN genutzt werden soll. Die entsprechenden Einträge werden in den beiden folgenden Abschnitten jeweils erläutert. Um die Konfiguration zu überprüfen, müssen zuerst die Netzwerkdienste neu gestartet werden und anschließend der WPA-Supplicanten von Hand gestartet werden. Wenn die WLAN-Karte als wlan0 geführt wird lautet der Aufruf wie folgt:

```
sudo service networking restart
sudo wpa_supplicant -iwlan0 -c/etc/wpa_supplicant/wpa_supplicant.conf
```

Ein -d als Parameter liefert eine Debug-Ausgabe die ggf. bei der Fehlersuche helfen kann. Welche Netzwerkkarten aktuell mit welchem Namen und welcher IP-Adresse im System vorhanden sind, lässt sich mithilfe von ifconfig ausgeben.

C.1 WLAN mit WPA2-Personal einrichten

Um den Zugang zu einem mit WPA2-Personal gesichertes Netzwerk zu konfigurieren ist folgender Eintrag in der wpa_supplicant.conf zu ergänzen:

```
network={
    ssid="WLAN-Name"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    group=CCMP
    psk=Verschlüsseltes Passwort
}
```

Dabei muss den Eintrag für ssid mit dem Namen des Netzwerkes (SSID) ersetzt werden. Dass verschlüsselte Passwort erhält man dadurch, dass man in der Eingabeaufforderung wpa_passphrase mit der Angabe der SSID aufruft. Ein solcher Aufruf könnte dann wie folgt aussehen:

```
$ wpa_passphrase HeimWLAN
# reading passphrase from stdin
Klartextpasswort
network={
    ssid="HeimWLAN"
    #psk="Klartextpasswort"
    psk=8b4683f8c83d4484e5288f34b9a829e9222ead0a282aa49472653164dd0a13b9
}
```

Die auskommentierte Zeile mit dem Passwort im Klartext sollte man nicht in die Konfiguration mit aufnehmen.

C.2 WLAN mit WPA2-Enterprise einrichten

Die Konfiguration für ein WPA2-Enterprise gesichertes Netzwerk sieht sehr ähnlich aus. Hier sind im folgenden Konfigurationsblock neben dem WLAN-Namen und dem Passwort auch die Identität zu ersetzen:

```
network={
    ssid="WLAN-Name"
    priority=1
    proto=RSN
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="Identität"
    password=hash:Verschlüsseltes Passwort
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
}
```

Hier wird das Passwort durch einen Hash gesichert. Dazu muss in eine Textdatei das Passwort im Klartext in der einzigen Zeile stehen. Hier im Beispiel hat die Datei den Namen pass.txt. Wie im folgenden Beispiel erhält man dann den Passworthash:

```
$ cat pass.txt | xargs echo -n | iconv -t utf16le | openssl md4
(stdin)= e14140a4e0e02b64f054481320d8a845
```

Damit wird in der Konfigurationsdatei die passende Zeile geändert zu:

```
password=hash:e14140a4e0e02b64f054481320d8a845
```

Bei einigen Netzwerken, wie z. B. das universitätsübergreifende »Eduroam« muss man mit ca_cert="/etc/cert/zertifikatsname.cer" noch das nötige Zertifikat mit angeben.

Anhang D

Windows

D.1 Lesen und beschreiben der SD-Karte

Die SD-Karte mit Raspbian besitzt zwei verschiedene Partitionen, die Boot- und die Systempartition. Während die Bootpartition unter Windows direkt lesbar ist, kommt man an die Daten auf der SD-Karte nicht so einfach dran. Die Systempartition ist im ext4-Format angelegt, dass Windows nicht von Hause aus lesen kann. Es wird dazu ein passender Treiber benötigt. Einen solchen stellt z. B. die Firma Paragon bereit, der über ihrer Homepage unter https://www.paragon-software.com/de/home/extfs-windows-pro/index.html verfügbar ist.

Literatur

Bundesweite Informatikwettbewerbe, Hrsg. (Aug. 2015). *Computer Science Circles*. Bonn, Waterloo. URL: http://cscircles.cemc.uwaterloo.ca/0-de/ (besucht am 20. 08. 2015).

Foundation, Raspberry Pi (März 2017). *Scratch GPIO*. URL: https://is.gd/uELYrc (besucht am 10. 08. 2017).

Pieper, Johannes und Dorothee Müller, Hrsg. (Juni 2013–Juli 2014). *Material für den Informatikunterricht*. Arnsberg, Dortmund, Hamm, Wuppertal, Solingen. URL: http://web.archive.org/web/20221006170419/https://ddi.uni-wuppertal.de/www-madin//material/materialsammlung/ (besucht am 05. 07. 2023).

Plate, Jürgen (2016). *Raspberry Pi und Linux*. München. URL: http://www.netzmafia.de/skripten/hardware/RasPi/index.html (besucht am 19. 03. 2016).

Raspberry Pi Forum (Sep. 2015). *Deutsches Raspberry Pi Forum / Offizielles Raspberry Pi 7“ Display*. Fredenbeck. URL: http://www.forum-raspberrypi.de/Thread-offizielles-raspberry-pi-7-display?page=7 (besucht am 30. 11. 2015).

Wikipedia (Aug. 2015). *Raspberry Pi*. Wikipedia-Artikel. URL: https://de.wikipedia.org/wiki/Raspberry_Pi (besucht am 18. 08. 2015).