

Skriptum zum

# Raspberry Pi

Stand 24. März 2016



Raspberry Pi is a trademark of the Raspberry Pi Foundation

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>4</b>
<b>2</b>	<b>Installation des Systems</b>	<b>5</b>
2.1	Installation eines Abbilds . . . . .	5
2.1.1	Erstellen eines eigenen Abbilds . . . . .	6
2.2	Installation mit Noobs . . . . .	7
<b>3</b>	<b>Bedienung, Linux Grundlagen</b>	<b>8</b>
3.1	Starten des Raspberry Pi . . . . .	8
3.1.1	Erste Schritte . . . . .	8
3.1.2	Herunterfahren . . . . .	9
3.2	Bedienung in der Konsole . . . . .	9
3.2.1	Software aktualisieren . . . . .	11
3.2.2	Software installieren . . . . .	11
3.2.3	Zusammenfassen von Befehlen in einem Skript . . . . .	12
3.2.4	Entfernter Zugriff über SSH (Secure Shell) . . . . .	12
3.3	Bedienung in der graphischen Oberfläche . . . . .	13
3.4	Die Verzeichnisstruktur . . . . .	13
<b>4</b>	<b>Python Programmierung</b>	<b>14</b>
4.1	Grundlagen in Python . . . . .	14
<b>5</b>	<b>Elektrische Schaltungen</b>	<b>15</b>
5.1	Elektrische Widerstände . . . . .	15
5.1.1	Farbcodierung von Widerständen . . . . .	15
5.1.2	Widerstände in Schaltzeichnungen . . . . .	16
5.2	Ansteuerung des GPIO mit Python . . . . .	16
5.2.1	Einstellungen . . . . .	16
5.2.2	Ausgänge steuern . . . . .	17
5.2.3	Anschluss einer LED . . . . .	17
5.2.4	Eingänge . . . . .	18
5.2.5	Anschluss eines Schalters oder Tasters . . . . .	18
5.2.6	GPIO Belegung . . . . .	18
5.3	Passiver Infrarotsensor (PIR) als Schalter . . . . .	19
5.4	Lichtschranke . . . . .	20
5.4.1	Abfrage im Programm . . . . .	20
5.5	Zweizeilendisplay . . . . .	21
5.6	Ansteuerung von Motoren . . . . .	22
5.6.1	Geschwindigkeitsregelung über Pulsweitenmodulation . . . . .	23
5.7	Digitales Thermometer . . . . .	23
5.7.1	Treiber laden . . . . .	24
5.7.2	Temperatur auslesen . . . . .	24
5.7.3	Temperatur auslesen mit Python . . . . .	24
5.8	Analog/Digital-Wandler . . . . .	25
5.8.1	SPI einrichten . . . . .	25



5.8.2	Anschluss des A/D-Wandlers . . . . .	25
5.8.3	Einbindung in Python . . . . .	26
5.8.4	Anschluss eines Potentiometers . . . . .	26
5.8.5	Anschluss eines analogen Thermometers . . . . .	27
<b>6</b>	<b>Software</b>	<b>28</b>
6.1	L <sup>A</sup> T <sub>E</sub> X auf dem Raspberry Pi . . . . .	28
6.1.1	Installieren von L <sup>A</sup> T <sub>E</sub> X . . . . .	28
6.1.2	Nutzung von L <sup>A</sup> T <sub>E</sub> X . . . . .	28
6.1.3	Weitere Möglichkeiten . . . . .	29
6.1.4	Installation direkt aus den TeXLive-Quellen . . . . .	29
<b>7</b>	<b>Komponenten</b>	<b>32</b>
7.1	Installation des offiziellen Raspberry Pi Displays . . . . .	32
7.2	Installation des Watterott-Displays . . . . .	32
	<b>Literaturverzeichnis</b>	<b>34</b>



# 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.

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 eingelnagen (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.

### 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, achtet man bereits vorher darauf, dass das Dateisystem nur die wirklich nötige Größe bekommt und nicht auf die gesamte SD-Karte expandiert wird. Dann kann man auch nur diesen Teil kopieren. Dieses geschieht durch die Angaben von `bs` und `count`. 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
```

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 »raspberrypi« 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 und auch das Passwort 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.

### 3.1.2 Herunterfahren

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 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.

Erreichen kann man eine Konsole auf dem Raspberry Pi entweder dadurch, dass sie direkt gestartet wird<sup>1</sup> oder durch den Aufruf mit Hilfe von Strg+Alt+F1 bis F6<sup>2</sup>. Es lässt sich aber auch in der graphischen Oberfläche mit Hilfe von »lxterminal« eine Konsole starten.

<sup>1</sup>Sie ist die Alternative zum direkten Starten der graphischen Oberfläche.

<sup>2</sup>Sollte die graphische Oberfläche gestartet sein, kann man in sie mit Strg+Alt+F7 zurückkehren



```
Linux raspberrypi 4.1.7-v7+ #817 SMP PREEMPT Sat Sep 19 15:32:00 BST 2015 armv7L
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov  5 23:13:24 2015
pi@raspberrypi ~ $ ls
blink11.py      Documents      photos         Scratch
buttonFeld.py  dokumente     python_games   scratchgpio5
Desktop        fullscreen.py  rpi-display.sh
pi@raspberrypi ~ $
```

Abbildung 3.1: Konsole auf dem Raspberry Pi

Die wichtigsten Befehle in der Konsole auf einem Linux-System sind folgende:

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 <sup>3</sup> 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

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.

<sup>3</sup>Als root bezeichnet man in Linux-Systemen den Benutzer mit uneingeschränkten Rechten auf dem System.





### 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.

## 3.2.1 Software aktualisieren

Fast die komplette Software die auf dem Raspberry 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 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.

## 3.2.2 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«.



### 3.2.3 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.

### 3.2.4 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 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.



## 3.3 Bedienung in der graphischen Oberfläche

### 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.



# 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.

### 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[BWINF 2015]. Dieses Angebot ist unter der Adresse <http://bit.ly/bwinf-kurs> zu erreichen.



# Kapitel 5

## Elektrische Schaltungen

### 5.1 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.

#### 5.1.1 Farbcodierung von Widerständen

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.

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  $10 \cdot 10^3 \Omega = 10.000 \Omega = 10k\Omega$  und orange, orange und braun  $33 \cdot 10^1 \Omega = 330 \Omega$  hat.

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	$10^9 = 1.000.000.000$



#### Aufgabe 5.1

Bestimme für folgende Widerstände die Farbcodierung bzw. den Wert:



- a) rot, gelb, grün   b) violett, rot, gold   c) braun, schwarz, grün   d) gelb, weiß, blau  
e)  $7,8M\Omega$    f)  $1,6k\Omega$    g)  $420\Omega$    h)  $300m\Omega$

## 5.1.2 Widerstände in Schaltzeichnungen

In Schaltplänen wird oft neben den Schaltzeichen für Widerstände  $\text{—}\square\text{—}$  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  $4,5M\Omega$  an.



### Aufgabe 5.2

Gib für folgende Widerstände die kompakte Schreibweise bzw. ihren Wert an:

- a) 580R   b) 3k4   c) M95   d) 73k  
e)  $1,6k\Omega$    f)  $30\Omega$    g)  $650k\Omega$    h)  $620m\Omega$

## 5.2 Ansteuerung des GPIO mit Python

Mit Hilfe des GPIO lassen sich Elemente durch ein Python-Programm außerhalb des Computers steuern, aber auch von außen kann auf das Programm Einfluss genommen werden. 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. Über sie kann jeweils nur True oder False gesendet bzw. empfangen werden.

### 5.2.1 Einstellungen

Um mit dem GPIO arbeiten zu können, müssen mit `import RPi.GPIO as GPIO` die entsprechenden Bibliotheken 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. 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.



## 5.2.2 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:

```
import RPi.GPIO as GPIO
import time

#Setzen der Nummernzaehlung der Pins. Alternativ GPIO.BCM
GPIO.setmode(GPIO.BOARD)

#Setzen der Pins auf Ausgabe
GPIO.setup(11, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)

#Beide Pins ausschalten
GPIO.output(11, False)
GPIO.output(13, False)

#Und ewiges Wechseln...
while 1:
    GPIO.output(11, True)

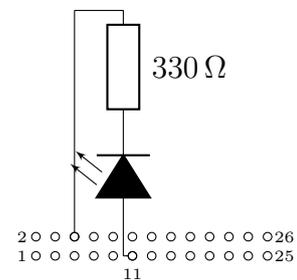
    #Fuer 1/5 Sekunde warten
    time.sleep(0.2)

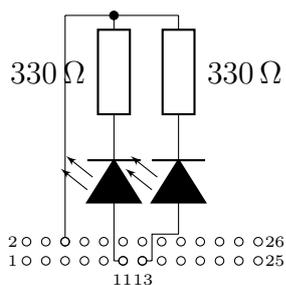
    GPIO.output(11, False)
    GPIO.output(13, True)

    time.sleep(0.2)
    GPIO.output(13, False)
```

## 5.2.3 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. Das längere der beiden Beine muss auf der Seite des geschalteten Ports liegen, während das kürzere mit der Masse (GND) verbunden wird. In der rechten Schaltskizze ist die LED so verbunden, dass sie über den Port 11 geschaltet wird.





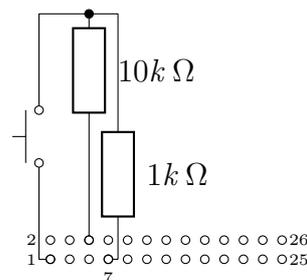
Sollen mehrere LEDs beschaltet werden, bekommt jede LED einen eigenen Vorwiderstand. Es gibt aber eine gemeinsame Verbindung zur Masse. Eine solche Schaltung ist in der linken Schaltskizze aufgezeichnet worden. Bei dieser lassen sich die zwei LEDs über die Ports 11 und 13 schalten.

## 5.2.4 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 Booleschen Wert zurück.

## 5.2.5 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 nebenstehenden Schaltskizze 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.



## 5.2.6 GPIO Belegung

Um eine einfache Übersicht über die Ports des GPIO und ihre Belegung zu erhalten, kann man die untere linke Übersicht 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+ und 2B stimmen die ersten 26 Pins in ihrer Belegung mit dem Modell A und B überein. **Wichtig:** Bei diesem Vorgang sollte der Raspberry Pi nicht mit der Stromversorgung verbunden sein, da es sonst zu Beschädigungen kommen kann.

Bei der rechten Übersicht sind die Seiten vertauscht. Sie wird dann benötigt, wenn man z. B. mit einem alten Festplattenkabel die Anschlüsse des GPIO direkt an ein Steckbrett legt. Durch die Kabelführung kommt es zu dieser Vertauschung.



Möchte man die Pins in Python mit den Nummern ansprechen, die zusätzlich angegeben sind, so ist `GPIO.BCM` bei `GPIO.setmode` anzugeben.

### Raspberry Pi

```

3.3V 1 0 0 2 5V
SDA 2 3 0 0 4 5V
SCL 3 5 0 0 6 GND
4 7 0 0 8 14 TXD
GND 9 0 0 10 15 RXD
17 11 0 0 12 18
27 13 0 0 14 GND
22 15 0 0 16 23
3.3V 17 0 0 18 24
MOSI 10 19 0 0 20 GND
MISO 9 21 0 0 22 25
SCKL 11 23 0 0 24 8
GND 25 0 0 26 7
    
```

### Board

```

5V 2 0 0 1 3.3V
5V 4 0 0 3 2 SDA
GND 6 0 0 5 3 SCL
TXD 14 8 0 0 7 4
RXD 15 10 0 0 9 GND
18 12 0 0 11 17
GND 14 0 0 13 27
23 16 0 0 15 22
24 18 0 0 17 3.3V
GND 20 0 0 19 10 MOSI
25 22 0 0 21 9 MISO
8 24 0 0 23 11 SCKL
7 26 0 0 25 GND
    
```

### Raspberry Pi

```

3.3V 1 0 0 2 5V
SDA 2 3 0 0 4 5V
SCL 3 5 0 0 6 GND
4 7 0 0 8 14 TXD
GND 9 0 0 10 15 RXD
17 11 0 0 12 18
27 13 0 0 14 GND
22 15 0 0 16 23
3.3V 17 0 0 18 24
MOSI 10 19 0 0 20 GND
MISO 9 21 0 0 22 25
SCKL 11 23 0 0 24 8
GND 25 0 0 26 7
ID_SD 27 0 0 28 ID_SC
5 29 0 0 30 GND
6 31 0 0 32 12
13 33 0 0 34 GND
19 35 0 0 36 16
26 37 0 0 38 20
GND 39 0 0 40 21
    
```

### Board

```

5V 2 0 0 1 3.3V
5V 4 0 0 3 2 SDA
GND 6 0 0 5 3 SCL
TXD 14 8 0 0 7 4
RXD 15 10 0 0 9 GND
18 12 0 0 11 17
GND 14 0 0 13 27
23 16 0 0 15 22
24 18 0 0 17 3.3V
GND 20 0 0 19 10 MOSI
25 22 0 0 21 9 MISO
8 24 0 0 23 11 SCKL
7 26 0 0 25 GND
ID_SC 28 0 0 27 ID_SD
GND 30 0 0 29 5
12 32 0 0 31 6
GND 34 0 0 33 13
16 36 0 0 35 19
20 38 0 0 37 26
21 40 0 0 39 GND
    
```

## 5.3 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.

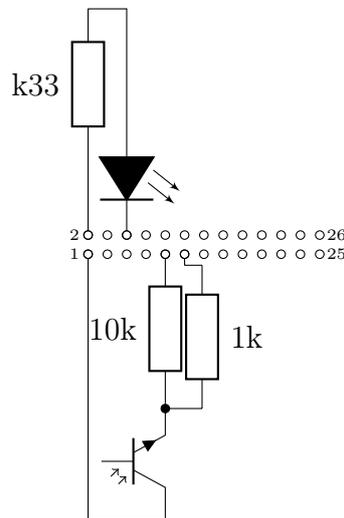
Die für Raspberry Pi nutzbare Version des PIR hat drei Anschlüsse: Die Spannungsversorgung mit 5V und die Masse können direkt mit den passenden Pins des GPIO verbunden werden. Über den dritten Anschluss kann der PIR abgefragt werden. Da bei Vorliegen eines Signals nur eine Spannung von 3,3V gegenüber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



## 5.4 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 bei maximal etwa einem cm liegen darf. Mit einer sehr hellen weißen LED bei etwa 10cm. 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.

### 5.4.1 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.

```
#!/usr/bin/env python3

import time, RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN)
```



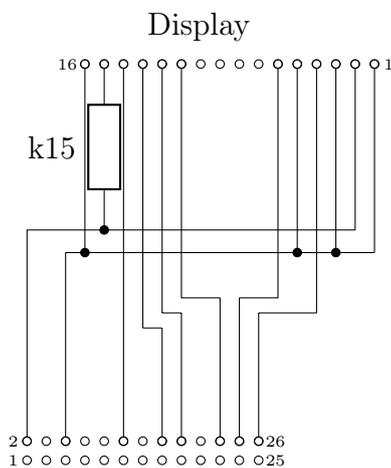
```

while True:
    input_value = GPIO.input(11)
    print(input_value)
    time.sleep(0.1)

```

## 5.5 Zweizeilendisplay

Die meisten verkauften LCD-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.



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)

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). Dieses sollte aber nicht gemacht werden, da der Raspberry durch die dann anliegende Spannung von 5 V zerstört werden kann.



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 muss mit sudo-Rechten ausgeführt werden und 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.

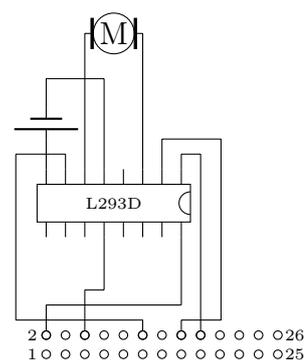
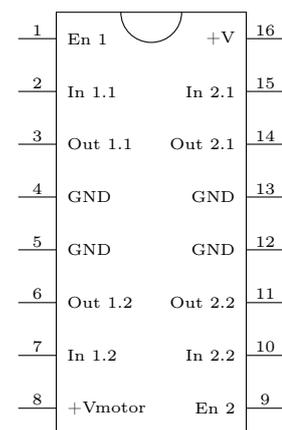
## 5.6 Ansteuerung von Motoren

Die meisten Motoren benötigen eine höhere Spannung als 3,3 V und damit verbunden einen höheren Strom, als ihn der Raspberry 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.

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.

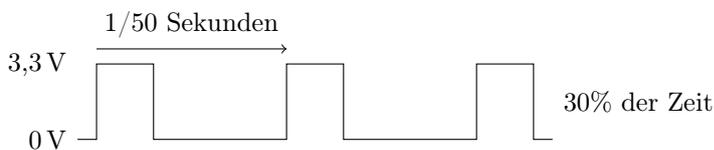
Wird der Chip und der Motor wie in der nebenstehenden Darstellung beschaltet, so kann mit den Pins 12 und 16 die Laufrichtung des Motors gesteuert werden. Erst wenn zu einem der beiden Pins auch der Pin 18 belegt ist, wird sich dann der Motor in Bewegung setzen.



## 5.6.1 Geschwindigkeitsregelung über Pulsweitenmodulation

Bei der Motorsteuerung ergibt sich beim Raspberry das Problem, dass er ein Pin nur an oder ausgeschaltet werden kann. Um damit die die Geschwindigkeit regulieren zu können, muss man sehr schnell an- und ausschalten, so dass der Motor z. B. nur bei der Hälfte der Zeit eine Spannung anliegen hat. Geschieht dieses Schalten schnell genug, dann dreht sich der Motor auch nur noch halb so schnell.

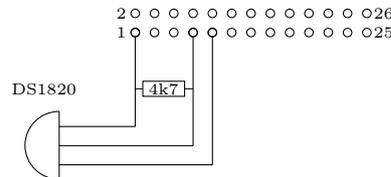
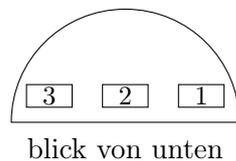
Für dieses schnelle Umschalten kann man die Pulsweitenmodulation nutzen. Dabei wird eine Folge Strom an und aus, mit einer wählbaren Frequenz wiederholt. Außerdem kann man dabei angeben, bei welchem Anteil dieser Folge der Strom angeschaltet ist.



```
# Pin 18 mit Frequenz von
50 Hz
p = GPIO.PWM(18, 50)
# Starten mit 30% Strom an
p.start(30)
# Ändern auf 50% Strom an
p.ChangeDutyCycle(50)
# Stoppen
p.stop()
```

## 5.7 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.



Für die Nutzung mit dem Raspberry Pi gibt es schon passende Treiber. Dazu müssen aber die Versorgung und die Datenleitung mit einem 4,7 kΩ 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.



### 5.7.1 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=1` mit 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
```

### 5.7.2 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.

### 5.7.3 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 folgenden 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)
```

## 5.8 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  $2^{12} - 1 = 4095$ .

### 5.8.1 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`:

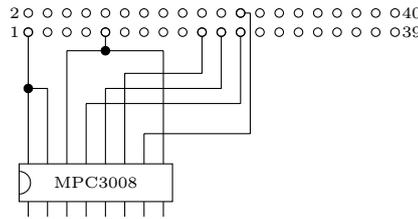
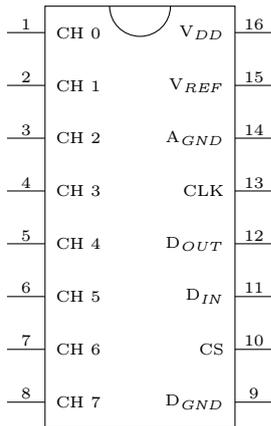
```
dtoverlay=spi
```

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

### 5.8.2 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.





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)

### 5.8.3 Einbindung in Python

Die Einbindung des MCP3008 in Python ist durch die Python Bibliothek »gpiozero«<sup>1</sup> 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.

### 5.8.4 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.

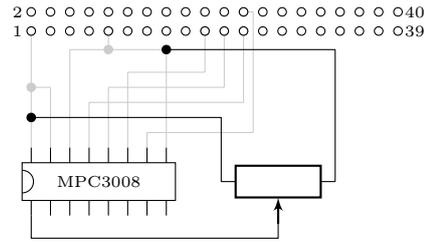
<sup>1</sup>Wenn es noch nicht vorhanden ist, kann es mit `sudo apt-get install python3-gpiozero` nachinstalliert werden



```

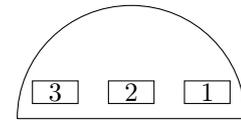
1 import time, gpiozero
2
3 adc = gpiozero.MCP3008(channel=0)
4
5 while True:
6     wert = int(adc.value * 100)
7     print('Der_Poti_steht_auf', wert, '%')
8     time.sleep(1)

```

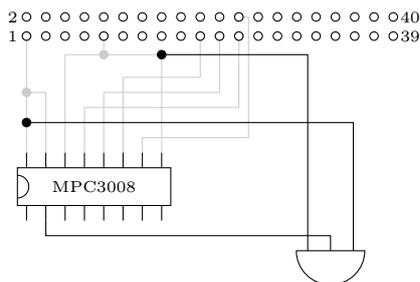


### 5.8.5 Anschluss eines analogen Thermometers

Ein analoger Thermometerchip hat am Datenpin je nach gemessener Temperatur eine unterschiedliche Spannung. Für den Einsatz am Raspberry Pi eignen 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



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  $0,5^\circ$  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:<sup>1</sup>

```
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.

---

<sup>1</sup>Die Raspbian-Quellen sind stabil aber nicht immer die neusten. Eine Installation mit aktuellen Quellen ist unter 6.1.4 zu finden.



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

```
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 L<sup>A</sup>T<sub>E</sub>X bei weitem nicht deutlich. Weitere kleine Beispiele gibt es auf verschiedenen Seiten im Internet. Zu empfehlen ist dabei unter anderem das L<sup>A</sup>T<sub>E</sub>X-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>.

#### Aufgaben 6.1

- Installiere das Grundsystem von TeXLive
- Schreibe deinen eigenen kleinen Text und lasse ihn in ein PDF-Dokument übersetzen.
- 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 L<sup>A</sup>T<sub>E</sub>X ein.

### 6.1.4 Installation direkt aus den TeXLive-Quellen

Fast täglich gibt es Ergänzungen und Verbesserungen bei den L<sup>A</sup>T<sub>E</sub>X-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 TeXLive und nicht die Pakete zum setzen. Aus diesen Gründen kann es sinnvoll sein, direkt auf die TeXLive-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 TeXLive für den Raspberry Pi wird der Netzwerkinstaller benötigt, der folgendermaßen heruntergeladen und anschließend 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.

```
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 *L<sup>A</sup>T<sub>E</sub>X*-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.



# Kapitel 7

## Komponenten

### 7.1 Installation des offiziellen Raspberry Pi Displays

Seit Mitte 2015 gibt es von der Raspberry Pi Foundation ein Display, das sich direkt an den Displayport des Raspberry Pi anschließen lässt. Hat man eine aktuelle Version des Systems (siehe 3.2.1), 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]).

### 7.2 Installation des Watterott-Displays

Um das Display von Watterott <http://www.watterott.com/de/RPi-Display-B-Plus> in Betrieb nehmen zu können, muss es entweder über die Adapter oder direkt auf den GPIO aufgesetzt werden. Nach dem Starten des Systems muss 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 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 fbcf remove
```

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



# Literaturverzeichnis

- [BWINF 2015] BUNDESWEITEN INFORMATIKWETTBEWERBEN (Hrsg.): *Computer Science Circles*. August 2015. – URL <http://cscircles.cemc.uwaterloo.ca/0-de/>. – Zugriffsdatum: 2015-08-20
- [PieperMueller 2014] PIEPER, Johannes (Hrsg.) ; MÜLLER, Dorothee (Hrsg.): *Material für den Informatikunterricht*. Juli 2014. – URL <http://ddi.uni-wuppertal.de/material/materialsammlung/>. – Zugriffsdatum: 2015-01-15
- [Plate 2016] PLATE, Jürgen: *Raspberry Pi und Linux*. 2016. – URL <http://www.netzmafia.de/skripten/hardware/RasPi/index.html>. – Zugriffsdatum: 2016-03-19
- [Raspberry Pi Forum 2015] RASPBERRY PI FORUM: *Deutsches Raspberry Pi Forum | Offizielles Raspberry Pi 7“ Display*. September 2015. – URL <http://www.forum-raspberrypi.de/Thread-offizielles-raspberry-pi-7-display?page=7>. – Zugriffsdatum: 2015-11-30
- [Wikipedia 2015] WIKIPEDIA: *Raspberry Pi*. Wikipedia-Artikel. August 2015. – URL [https://de.wikipedia.org/wiki/Raspberry\\_Pi](https://de.wikipedia.org/wiki/Raspberry_Pi). – Zugriffsdatum: 2015-08-18

