

 [Informatikprojekt 7bi Schuljahr 2017/2018 als PDF exportieren](#)

INFORMATIK PROJEKTTAGE 2017/18

Tagesablauf

- ORT: INFOSAAL 4, BRG Amstetten
- BEGINN: 7:25
- ENDE: 16:00 (MI, DO) bzw. 12:55 (FR)

Inhalte

- Tag 1 - Programmierung einer Android Handy-APP
- Tag 2 - Raspberry PI Einführung
- Tag 3 - Selbstfahrender Roboter mithilfe eines Raspberry PI's

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18

Last update: **2018/04/13 07:56**

PROGRAMMIERUNG einer HANDY-APP (Android)

Übersicht

- Mobile Android-Anwendungsentwicklung

Vorbereiten der Entwicklungsumgebung (Android)

- Für die Anwendungsentwicklung unterstützte Android-Geräte
- Aktivieren des USB-Debugging auf Android-Geräten
- Konfigurieren des Systems für die Erkennung von Android-Geräten
- Hinzufügen von Android-SDKs

Entwickeln von Android-Apps

- Erstellen einer Android-App
- Migrieren von C++-Code zu mobilen Anwendungen vom Desktop aus

Ausführen von Android-Apps

- Ausführen von Android-Anwendungen auf einem Android-Gerät
- Ausführen und Debuggen von Android-Anwendungen auf einem Remote-Android-Gerät

Bereitstellen von Android-Apps

- Vorbereiten von Android-Anwendungen für die Bereitstellung
- Erstellen einer KeyStore-Datei
 - Hinzufügen eines Zertifikats zu einer Keystore-Datei
- Bereitstellen von signierten Android-Anwendungen
- Bereitstellen von unsigneden Android-Anwendungen

Fehlerbehandlung bei Android-Apps

- In der Projektverwaltung werden keine Android-Ziele aufgeführt
- Ausführen auf einem Android-Gerät nicht möglich
- Bereitstellen einer Anwendung für Android nicht möglich

Diverse Links:

- https://de.wikibooks.org/wiki/Programmierkurs:_Delphi
- <https://www.delphi-treff.de/tutorials/>
- <http://www.marcocantu.com/EPascal/German/default.htm>
- <http://www.math.uni-leipzig.de/pool/tuts/Delphi/index.htm>
- <http://www.oszhandel.de/gymnasium/faecher/informatik/delphi/index.htm>
- <http://docwiki.embarcadero.com/RADStudio/Tokyo/de/Hauptseite>
- <http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Tutorials>
- <http://docwiki.embarcadero.com/RADStudio/Tokyo/de/Tutorials>
- http://docwiki.embarcadero.com/RADStudio/Tokyo/de/FireMonkey_Quick_Start_-_Einf%C3%BChrung
- <https://www.youtube.com/watch?v=raLJVnEEjWA>
 - <https://www.youtube.com/watch?v=VpajfLeyiQ>
- https://www.youtube.com/watch?v=ug3-BNNPY_U
 - <https://community.embarcadero.com/blogs/entry/building-mobile-apps-for-ios-and-android-from-one-codebase-webinar-feb-22-11am-amsterdam>

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:01

Last update: **2018/04/14 19:08**

Raspberry PI Einführung

- Was ist ein Raspberry PI?
- GPIO - Schnittstelle
- Betriebssystem
- Was der Raspberry PI alles braucht...
- Raspberry PI Verbindung
- Python als Programmiersprache

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

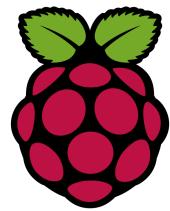
Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02



Last update: **2018/04/12 13:56**

RASPBERRY PI



Der **Raspberry Pi** ist ein Einplatinencomputer, der von der britischen **Raspberry Pi Foundation** entwickelt wurde. Der Rechner enthält ein **Ein-Chip-System** von Broadcom mit einem **ARM-Mikroprozessor**, die Grundfläche der Platine entspricht etwa den **Abmessungen einer Kreditkarte**. Der Raspberry Pi kam Anfang 2012 auf den Markt; sein großer Markterfolg wird teils als Revival des bis dahin weitgehend bedeutungslos gewordenen Heimcomputers zum Programmieren und Experimentieren angesehen. Der im Vergleich zu üblichen Personal Computern sehr einfach aufgebaute Rechner wurde von der Stiftung mit dem Ziel entwickelt, **jungen Menschen den Erwerb von Programmier- und Hardwarekenntnissen** zu erleichtern. Entsprechend niedrig wurde der Verkaufspreis angesetzt, der ca. 35 € beträgt.



What is a Raspberry Pi?

Modelle

Die Tabelle zeigt eine Zusammenfassung der Funktionen der Modelle A und B von Raspberry Pi.

Merkmale	Modell A	Modell A+	Modell B	Modell B+	Pi2 Modell B	Pi3 Modell B
USB-Anschlüsse	1	1	2	4	4	4
HDMI	1	1	1	1	1	1
Composite Video	1	1	1	1	1	1
Kopfhörerausgang	1	1	1	1	1	1
Ethernet	0	0	1	1	1	1
802.11b / g / n	Nein	Nein	Nein	Nein	Nein	Ja
Bluetooth (4.1)	Nein	Nein	Nein	Nein	Nein	Ja
GPIO-Stifte	26	40	26	40	40	40
Kamera (CSI)	1	1	1	1	1	1
Display (DSI)	1	1	1	1	1	1
Prozessorgeschwindigkeit/Kerne	700 MHz Single Core	700 MHz Single Core	700 MHz Single Core	700 MHz Single Core	900 MHz Quad Core	1,2 GHz Quad Core
SDRAM	256 MB	256 MB	512 MB	512 MB	1 GB	1 GB
Betriebssystem	SD	microSD	SD	microSD	microSD	microSD
HAT-kompatibel	Nein	Ja	Nein	Ja	Ja	Ja

Schnittstellen (Raspberry PI 3 B)



From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_01



Last update: **2018/04/12 12:55**

Schnittstelle GPIO

Der Raspberry Pi bietet eine **frei programmierbare Schnittstelle** (auch bekannt als **GPIO**, **General Purpose Input/Output**), worüber **LEDs, Sensoren, Displays und andere Geräte angesteuert** werden können. Es gibt **fünf GPIO-Anschlüsse**, wobei im Allgemeinen nur der Anschluss P1 gebraucht wird. Die GPIO-Schnittstelle P1 besteht je nach Modell aus **26 Pins** bzw. **40 Pins**, jeweils ausgeführt als doppelreihige Stifteleiste, wovon

- **2 Pins eine Spannung von 5 Volt** bereitstellen, aber auch genutzt werden können, um den Raspberry Pi mit Strom zu versorgen,
- **2 Pins eine Spannung von 3,3 Volt** bereitstellen,
- **8 Pins als Masse** dient,
- **26 Pins** sind **frei programmierbar**. Sie sind für eine **Spannung von 3,3 Volt** ausgelegt. Einige von ihnen können Sonderfunktionen übernehmen:
- 2 Pins (27 & 28) als BCM-Schnittstelle



+ 3,3 V	1	2	+ 5 V
(SDA) GPIO 2	3	4	+ 5 V
(SCL) GPIO 3	5	6	GND
(GPCLK0) GPIO 4	7	8	GPIO 14 (TXD)
GND	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
+ 3,3 V	17	18	GPIO 24
(MOSI) GPIO 10	19	20	GND
(MISO) GPIO 9	21	22	GPIO 25
(SCLK) GPIO 11	23	24	GPIO 8 (CE0)
GND	25	26	GPIO 7 (CE1)
ID_SD	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_02



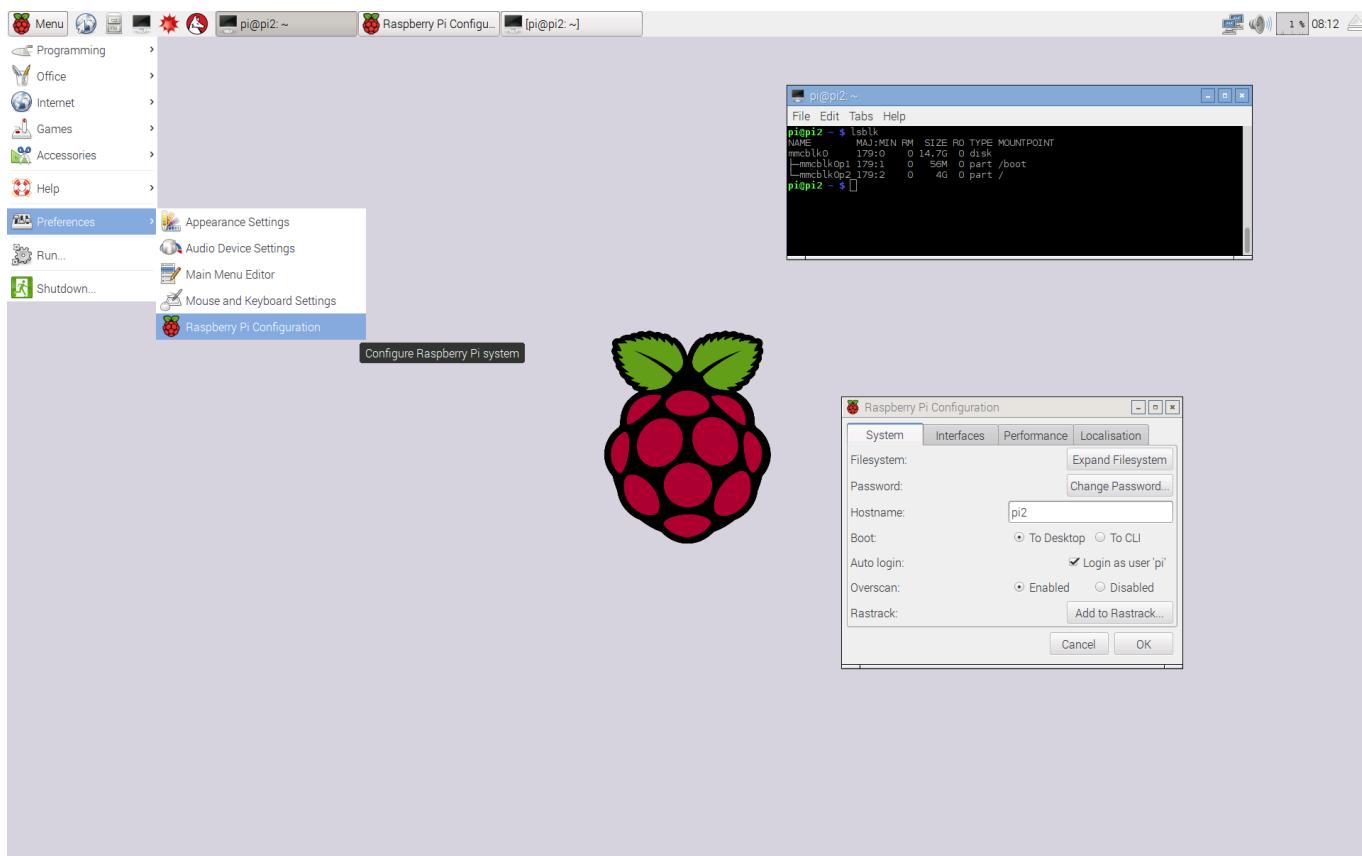
Last update: **2018/04/12 13:11**

Betriebssysteme

Für den Raspberry Pi gibt es gleich mehrere kompatible Betriebssysteme (Operating Systems - OS). Siehe [Raspberry Pi OS](#).

Für dieses Projekt wurde auf dem Raspberry Pi das auf **LINUX basierende Betriebssystem Raspbian** vorinstalliert. Dazu waren folgende Schritte notwendig:

1. Wir laden zuerst das Raspbian Image von der offiziellen Webseite unter <http://www.raspberrypi.org/downloads> herunter. Nach dem Download muss die ZIP-Datei entpackt werden. In dieser befindet sich eine IMG-Datei, welche das Betriebssystem beinhaltet.
2. Windows-Benutzer benötigen außerdem zum Überspielen des Images auf die SD-Karte das [Etcher](#).
3. Nun stecken wir die SD-Karte in Raspberry und stecken die Stromversorgung an.
4. Jetzt bootet der Raspberry Pi von alleine das Betriebssystem Raspbian.



From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_03



Last update: **2018/04/12 13:26**

Bestandteile

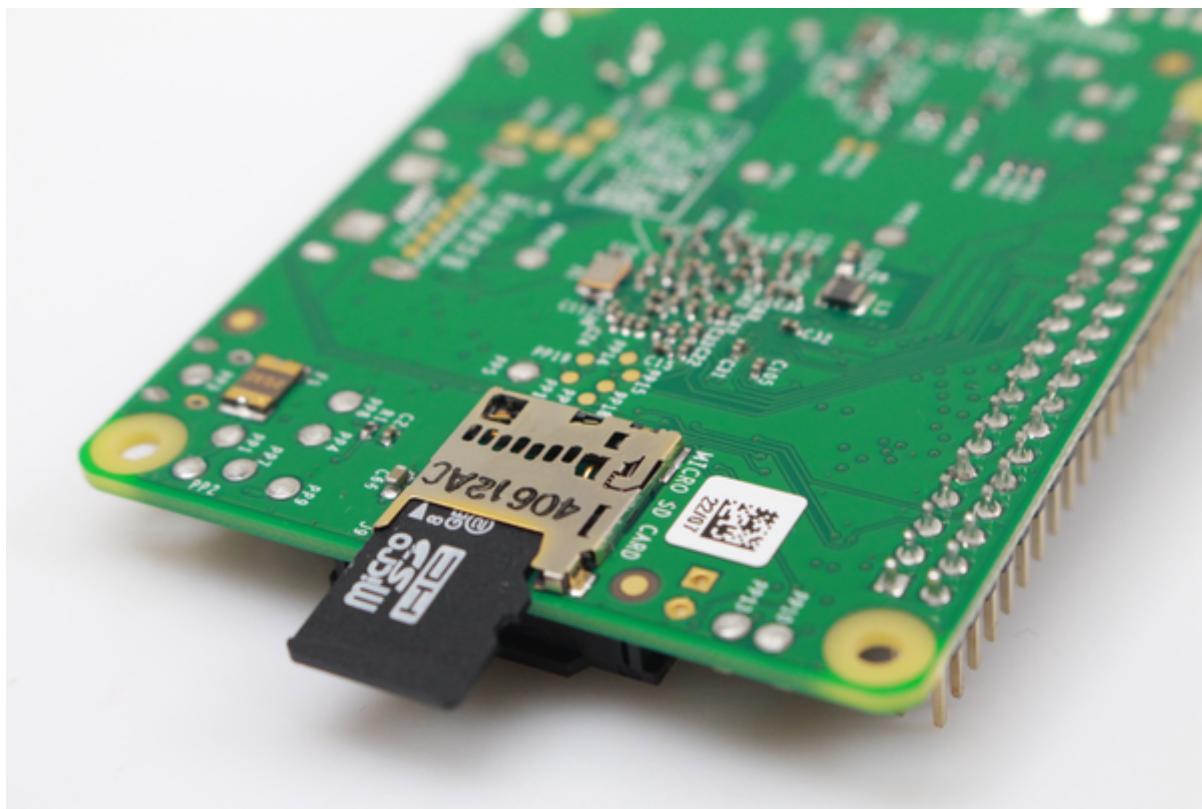
Außer dem Einplatinencomputer selbst braucht man für den Normalbetrieb noch einiges an Zubehör.

Grundbestandteile

1) SD Karte

Die SD Karte ist essenziell, denn das Raspberry Pi kann nicht von einem USB Stick oder Festplatte gestartet werden. Ich empfehle hier der Geschwindigkeit halber mindestens eine Class 10 SD mit **mindestens 8 GB** Kapazität. Prinzipiell sind aber **bis zu 32 GB möglich**. Die minimal benötigte Größe variiert von Betriebssystem zu Betriebssystem zwischen 2 und 6 GB. Falls du eine alte (micro)SD Karte aus der Kamera oder dem Handy hast, kannst du diese benutzen.

Je nachdem welches Modell du hast, brauchst du eine **normale SD Karte** (Model A/B) bzw. eine **microSD Karte** (Model B+, Pi 2, Raspberry Pi 3B).



2) Stromversorgung

Damit der Pi genug Strom bekommt, brauchst du ein **microUSB Kabel mit Netzteil** (gibt es auch in einem). Wenn du bereits ein microUSB Kabel hast, brauchst du nur noch ein USB Netzteil, welches für den Dauerbetrieb geeignet ist und **mindestens 1000mA** liefert (auch diese sind separat erhältlich). Das **Raspberry Pi 2/3** sollte mit einem Netzteil, welches **2000mA** liefert an die Steckdose angeschlossen werden.

Darüber hinaus gibt es auch die Möglichkeit **Powerbanks** oder sogar Akkus zu verwenden. Achte darauf, dass genug Stromstärke (Milliampere) vorhanden ist, da sonst die **Kapazität** nicht lange hält. Ich bevorzuge Powerbanks mit **mindestens 20.000 mA**.

3) HDMI Kabel + Monitor (optional)

Je nachdem für welche Anwendung ist ein HDMI Kabel + Monitor optimal. Willst du den Pi als reinen Server per SSH steuern, ist ein HDMI Kabel nicht nötig. Für andere Bereiche, wie z.B. als Multimedia Center sollte auf jeden Fall ein HDMI Kabel vorhanden sein.

4) Netzwerkverbindung (WLAN)

Bei den ersten Modellen gab es noch keine Netzwerkverbindung über Kabel, geschweige denn über WLAN. Somit musste man zuerst einen **WLAN-Usb-Stick** verwenden.



Der **Raspberry Pi 2** hatte zwar schon eine **RJ45-Buchse** für ein Netzwerkkabel aber noch immer keinen eingebauten WLAN-Empfänger.

Erst der **Raspberry Pi 3** hatte sowohl eine **RJ45-Buchse** als auch einen **WLAN-Empfänger**.

Prinzipiell ist eine Drahtlosverbindung in vielen Situationen sehr vorteilhaft.

5) Gehäuse (Optional)

Da der Pi ohne Zubehör, also auch ohne äußere Verkleidung geliefert wird, kann man ein Gehäuse nachkaufen. Es gibt sie in verschiedenen Formen und Farben (ab 3€ aufwärts). Unbedingt notwendig ist es aber nicht, daher sollte jeder für sich selbst entscheiden, ob er eines benötigt oder nicht.

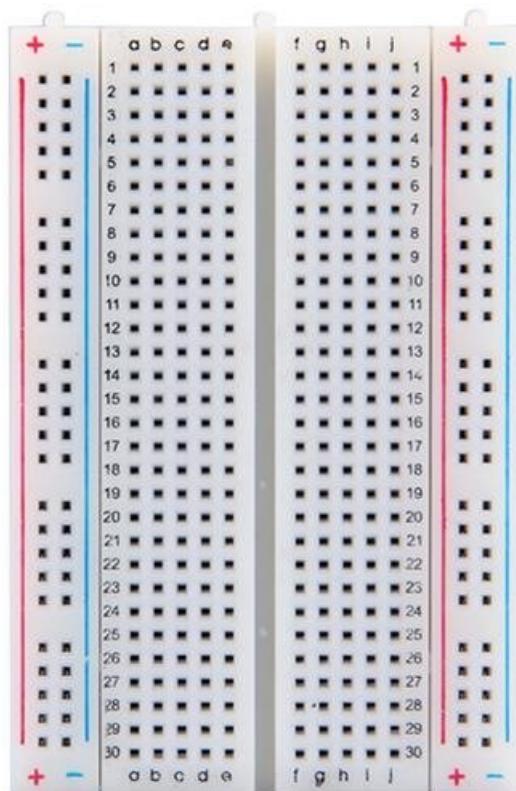


Zusätzliche Hardware zum Experimentieren

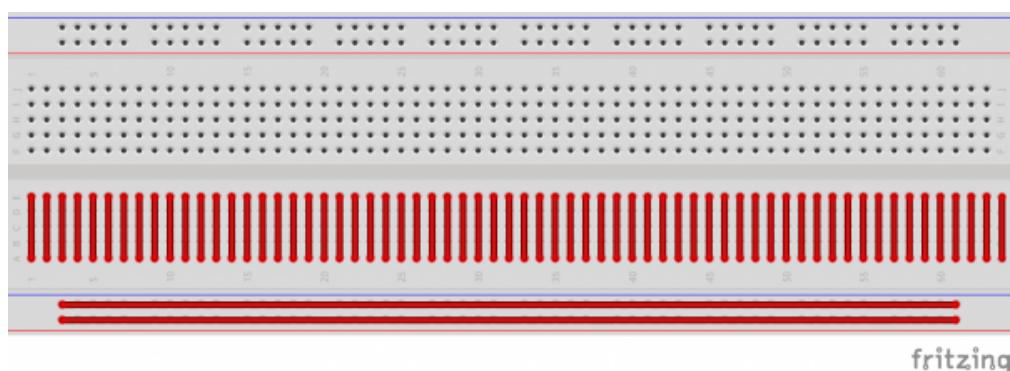
Jeder der sich mit dem Raspberry Pi beschäftigt, kommt früher oder später zum Einsatz der **GPIO Pins**. Diese programmierbaren Pins können Signale einlesen und ausgeben, wodurch sehr einfach programmierbare Schaltungen möglich sind. Dafür sind folgende Bauteile notwendig:

1) Breadboard (Steckbrett)

Um Schaltungen einfach aufzubauen und schnell zu verändern, hast du mit einem Breadboard die perfekte Lösung (2€). Dadurch ersparst du dir ständiges Löten deiner Bauteile, was vor allem nützlich ist, wenn du nur etwas testen willst bzw. die Teile erneut verwenden willst.

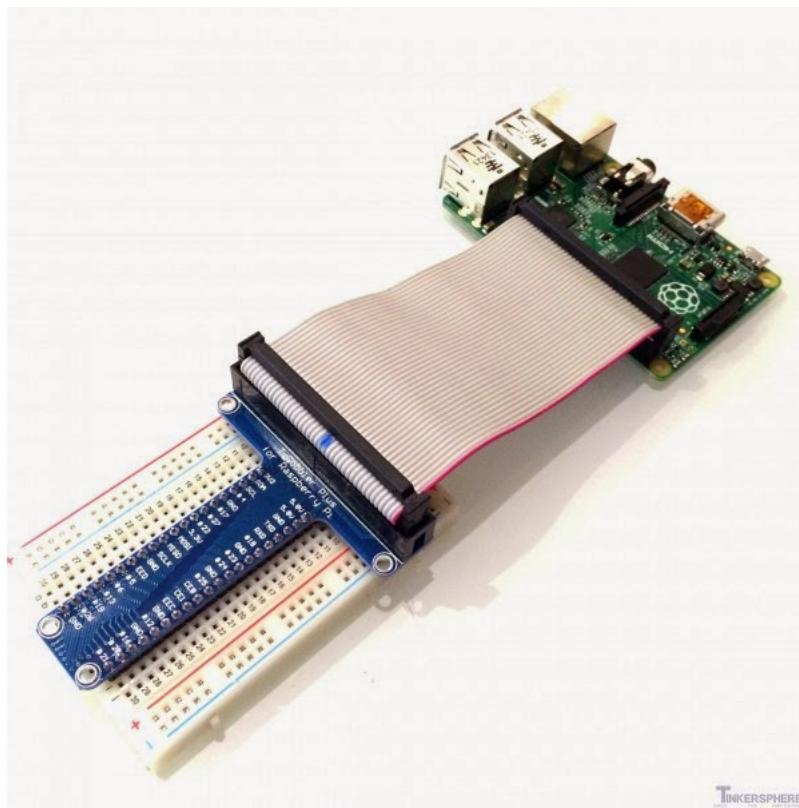
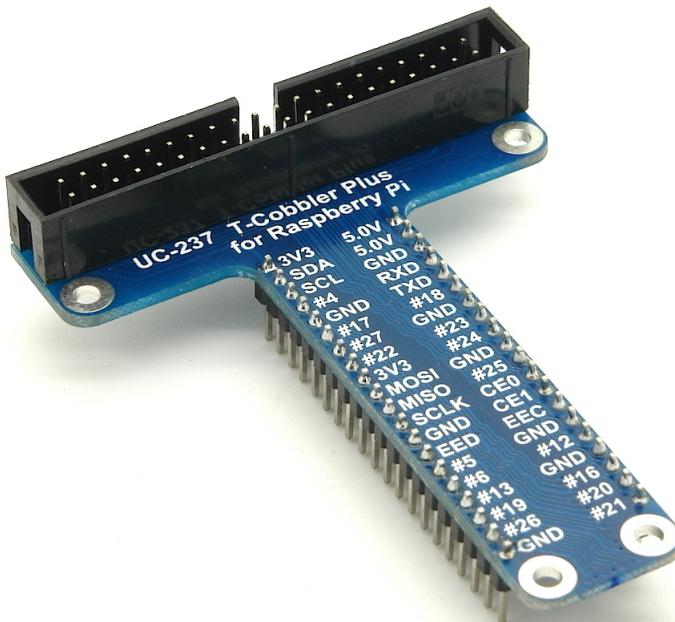


Bei jedem Steckbrett sind die **außenen Löcher (+ und -) vertikal** (1 bis 30) und die **inneren Löcher horizontal** (a bis e und f bis j) verbunden.



2) GPIO-Cobbler + Ribbon Kabel

Um die Signale vom den GPIO-Pins am Raspberry Pi zum Breadboard zu übertragen ist ein sogenannter **GPIO-Cobbler + Ribbon Kabel** notwendig.



3) Steckkabel (Jumper Kabel)

Kabel gibt es in 3 verschiedenen Ausführungen

- Male-Male
- Male-Female
- Female-Female

Du solltest von allen einige haben, jedoch sind die Male – Female am wichtigsten, da du damit die Verbindung der GPIOs mit dem Breadboard herstellen kannst.



4) LEDs, Sensoren, Widerstände, Taster

Je nach Projekt sind LEDs, verschiedene Sensoren (Ultraschallsensor, LineTrackerSensor,..), Widerstände (10k, 3.3k, ..) oder Taster notwendig.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_04



Last update: **2018/04/12 14:25**

Raspberry PI Verbindung

Um am Raspberry PI nicht jedes Mal einen Monitor, Maus und Tastatur anstecken zu müssen, sollte eine Verbindung über das Netzwerk ermöglicht werden.

Dazu gibt es wiederum mehrere Möglichkeiten. Zwei Möglichkeiten sind

- Terminalverbindung über SSH (ohne grafische Oberfläche)
- Remote-Zugriff über VNC oder RDP (mit grafischer Oberfläche)

Da beide Dienste eine bereits aufgebaute Netzwerkverbindung brauchen, muss man die Netzwerkverbindung zuerst über WLAN oder LAN herstellen.

SSH - Secure Shell

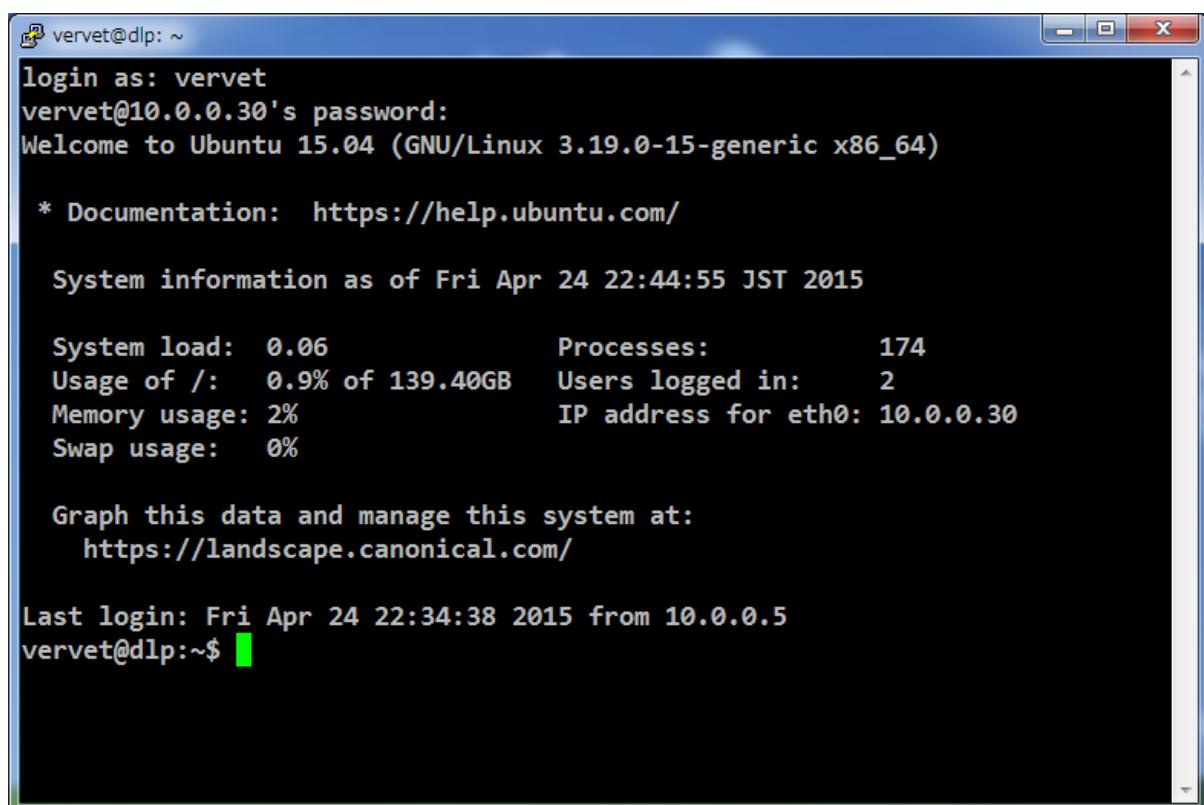
SSH bezeichnet ein Netzwerkprotokoll über das man eine gesicherte und authentifizierte Verbindung zu einem entfernten Gerät herstellen kann. Diese Methode wird häufig verwendet, um lokal eine entfernte Kommandozeile sicher verfügbar zu machen.

Das heißt die Tastatureingaben werden am lokalen PC werden an die entfernte Konsole gesendet und ausgeführt.

Die **IANA** hat dem Protokoll den TCP-Port 22 zugeordnet.

In Windows wird dafür zumeist das Programm **PuTTY** verwendet.

In Linux ist dafür kein zusätzliches Programm notwendig.



```
vervet@dlp: ~
login as: vervet
vervet@10.0.0.30's password:
Welcome to Ubuntu 15.04 (GNU/Linux 3.19.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Apr 24 22:44:55 JST 2015

System load:  0.06           Processes:          174
Usage of /:   0.9% of 139.40GB  Users logged in:   2
Memory usage: 2%              IP address for eth0: 10.0.0.30
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

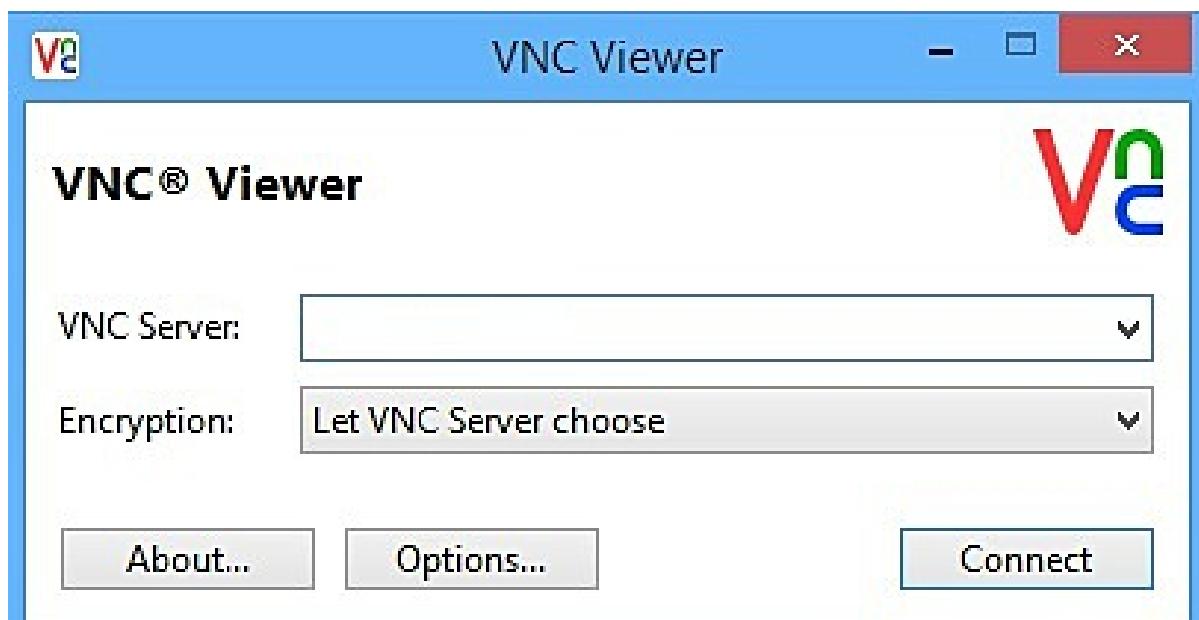
Last login: Fri Apr 24 22:34:38 2015 from 10.0.0.5
vervet@dlp:~$
```

VNC - Virtual Network Computing

VNC ist eine Software, die den **Bildschirminhalt eines entfernten Rechners** (Server) auf einem **lokalen Rechner (Client)** **anzeigt** und im Gegenzug Tastatur- und Mausbewegungen des lokalen Rechners an den entfernten Rechner sendet.

Damit kann man auf einem entfernten Rechner arbeiten, als säße man direkt davor.

VNC implementiert das Remote Framebuffer Protocol und ist damit, im Gegensatz zu anderer Fernwartungssoftware, plattformunabhängig benutzbar.



Aktivierung + Autostart

Normalerweise sind beide Dienste (SSH und VNC) am Raspberry PI vorinstalliert aber nicht gestartet. Damit beide Verbindungen möglich sind, muss man diese starten.

Will man diese nicht immer manuell aktivieren, so kann man diese auch per Autostart starten lassen.

Entweder man startet die Dienste über

- **grafische Oberfläche** (Menu → Preferences → Raspberry PI Configuration) oder
- **über die Konsole** (sudo raspi-config)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_05



Last update: **2018/04/12 14:13**

PYTHON

Python auf Deutsch ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Sie hat den Anspruch, einen gut lesbaren, knappen Programmierstil zu fördern.

So werden beispielsweise **Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert**. Wegen ihrer **klaren und übersichtlichen Syntax** gilt Python als einfach zu erlernen.

Python unterstützt mehrere Programmierparadigmen, z.B. die **objektorientierte, die aspektorientierte und die funktionale Programmierung**.

1) Einführung in PYTHON

2) GPIOs ansteuern

3) LED blinken lassen

4) LED per Taster schalten

Python Beispiel

```
geb_tag = 30
geb_monat = 7
geb_jahr = 1980
ausgabe = "" # leerer String

if geb_jahr < (2017 - 18):
    # Nutzer ist alt genug
    ausgabe = "Alter > 18"
else:
    if (geb_monat >= 6) and (geb_monat <= 9):
        ausgabe = "Kind hat im Sommer Geburtstag"
    else:
        # hier kann z.B. noch ein weiterer Befehl stehen
        ausgabe = "kein Sommerkind"
print(ausgabe)
```

Damit du ein Python-Programm ausführen kannst, musst du es mit der Endung „.py“ speichern.

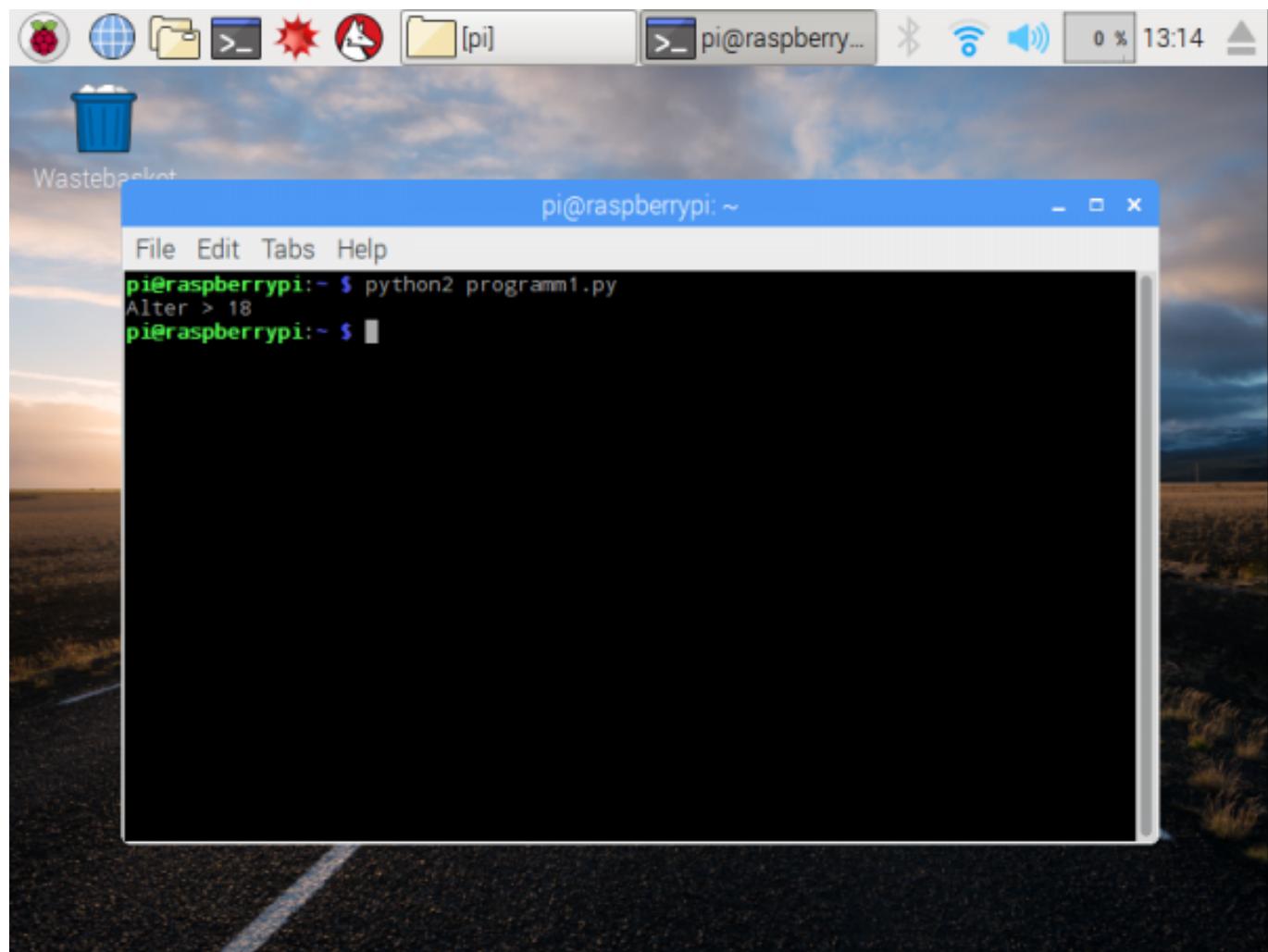
Nachdem du das Programm gespeichert hast, kannst du es mit

```
python programmname.py
```

ausführen.

Das Programm wird nun von Anfang bis Ende durchlaufen. Dabei werden eben alle Abfragen ausgeführt, bis das Programm am Ende angelangt ist, wo es keine weiteren Befehle gibt. Danach

terminiert es.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_06



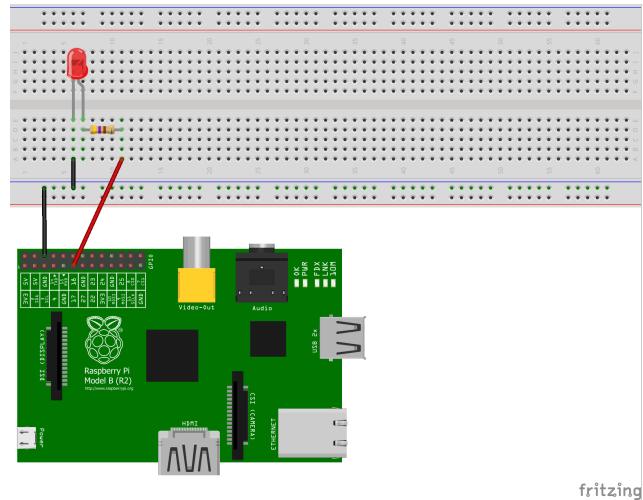
Last update: **2018/04/12 14:40**

ÜBUNG: Blinkendes LED

Für diese Übung braucht ihr folgende Bestandteile:

- LED
- Widerstände (330 Ohm oder 470 Ohm)

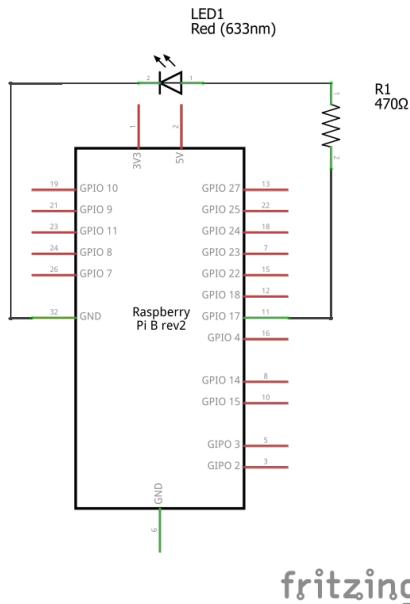
Bauplan



ACHTUNG:

- Schließe niemals mehr als 3.3V an die GPIOs an, da sie sonst kaputt gehen können.
- Die LED hat zwei unterschiedlich lange Enden. Das längere Ende kommt an die positive 3.3 Volt Spannung des GPIO 17 (Pin 11). Der gewählte Widerstand ist 330Ω (Ohm).

Schaltplan



Code

```

import time
import RPi.GPIO as GPIO

# Pin-Nummern wie auf dem Raspberry Board verwenden
GPIO.setmode(GPIO.BCM)

#LED ist an GPIO Pin 17 angeschlossen
LED=17

#GPIO 17 (=Pin 17) als Output definieren (Signal vom Raspberry an
#Steckbrett)
GPIO.setup(LED, GPIO.OUT)

# Dauerschleife für das Blinken
while 1:
    # LED aus -> niedriges Signal senden
    GPIO.output(LED, GPIO.LOW)
    # eine Sekunde warten
    time.sleep(1)
    # LED an -> hohes Signal senden
    GPIO.output(LED, GPIO.HIGH)
    # eine Sekunde warten
    time.sleep(1)

```

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_06:02_06_01



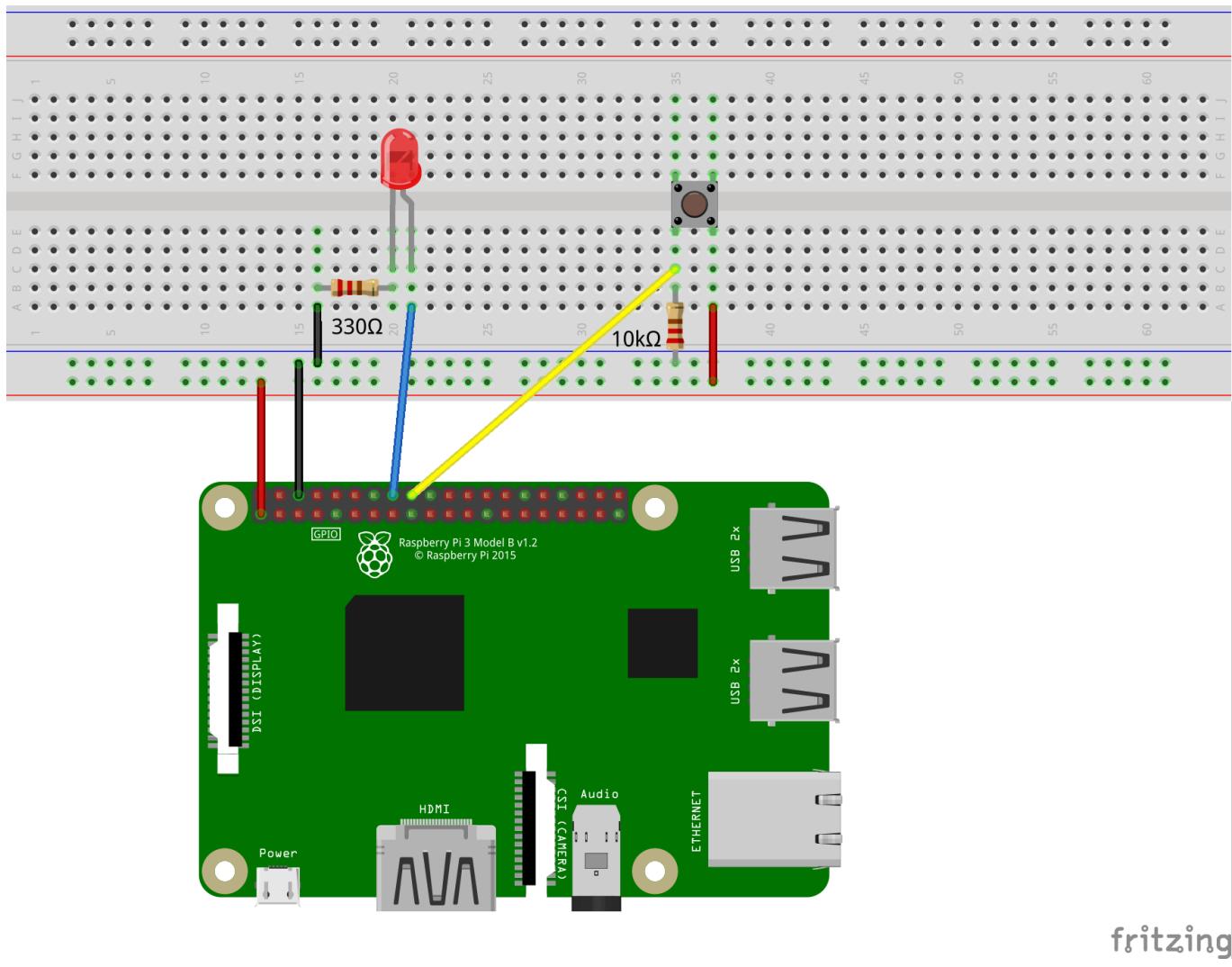
Last update: **2018/04/12 15:00**

ÜBUNG: LED per TASTER schalten

Für diese Übung braucht ihr folgende Bestandteile:

- LED
- Taster
- 330 Ohm Widerstand (als Vorwiderstand für die LED)
- 10 kOhm Widerstand

Bauplan



Code

```
import RPi.GPIO as GPIO
import time

#Bestimmte GPIO Mode
GPIO.setmode(GPIO.BCM)
```

```
#Definiere GPIO Pins als Output oder Input
GPIO.setup(23, GPIO.OUT)  #LED
GPIO.setup(24, GPIO.IN)   #Taster

# Endlosschleife
while True:
    #Taster ist nicht gedrückt
    if GPIO.input(24) == 0:
        # LED Ausschalten
        GPIO.output(23, GPIO.LOW)
    #Taster ist gedrückt
    else:
        # Einschalten
        GPIO.output(23, GPIO.HIGH)
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

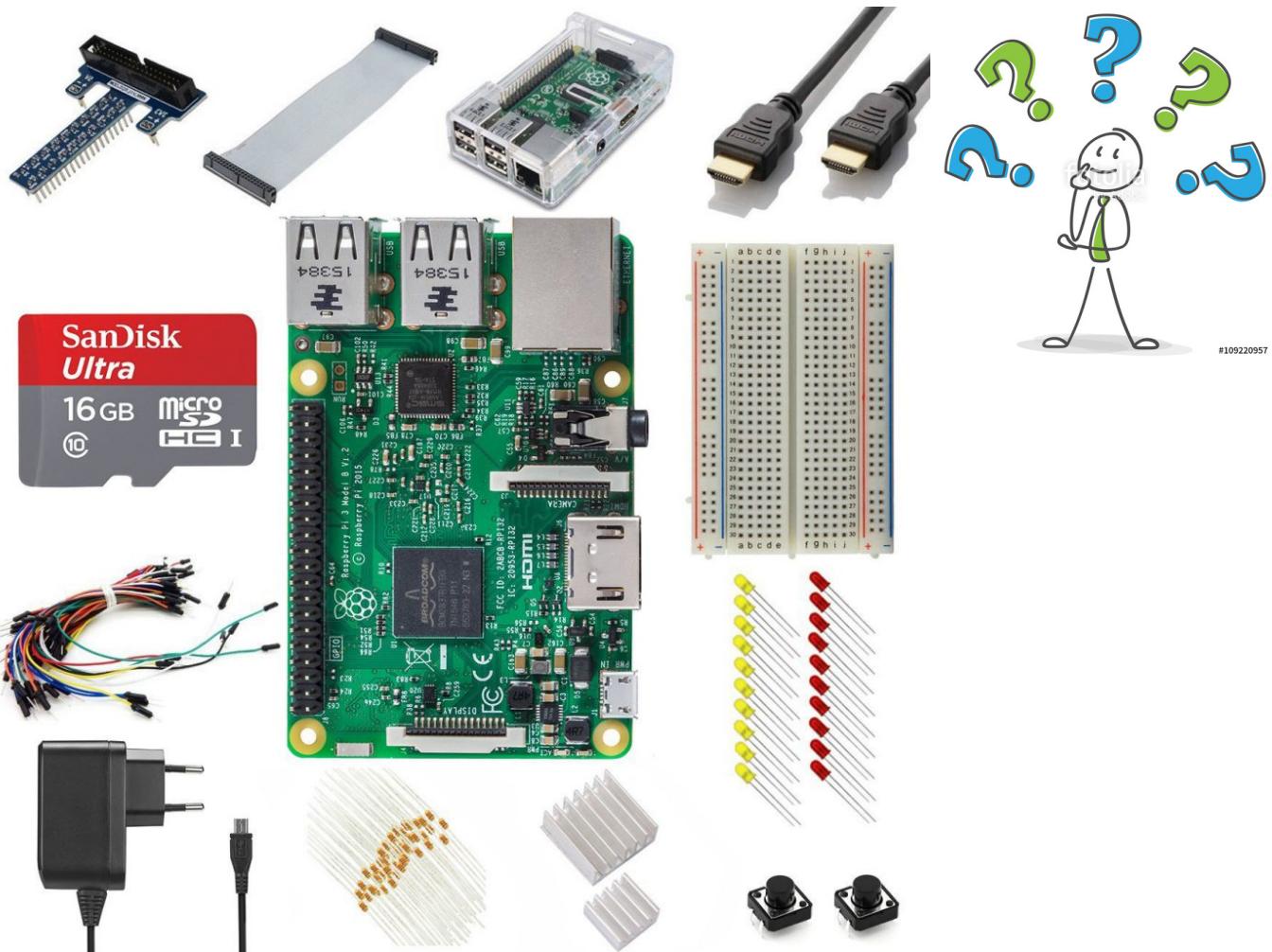
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:02:02_06:02_06_02



Last update: **2018/04/12 14:56**

Ferngesteuerter Roboter

START



ZIEL



Ferngesteuerter Roboter

Der WEG vom START zum ZIEL

- Zusammenbau
- Verkabelung
- H-Brücke Funktionsweise
- Programmierung

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:03

Last update: **2018/04/13 08:05**

Roboterbau

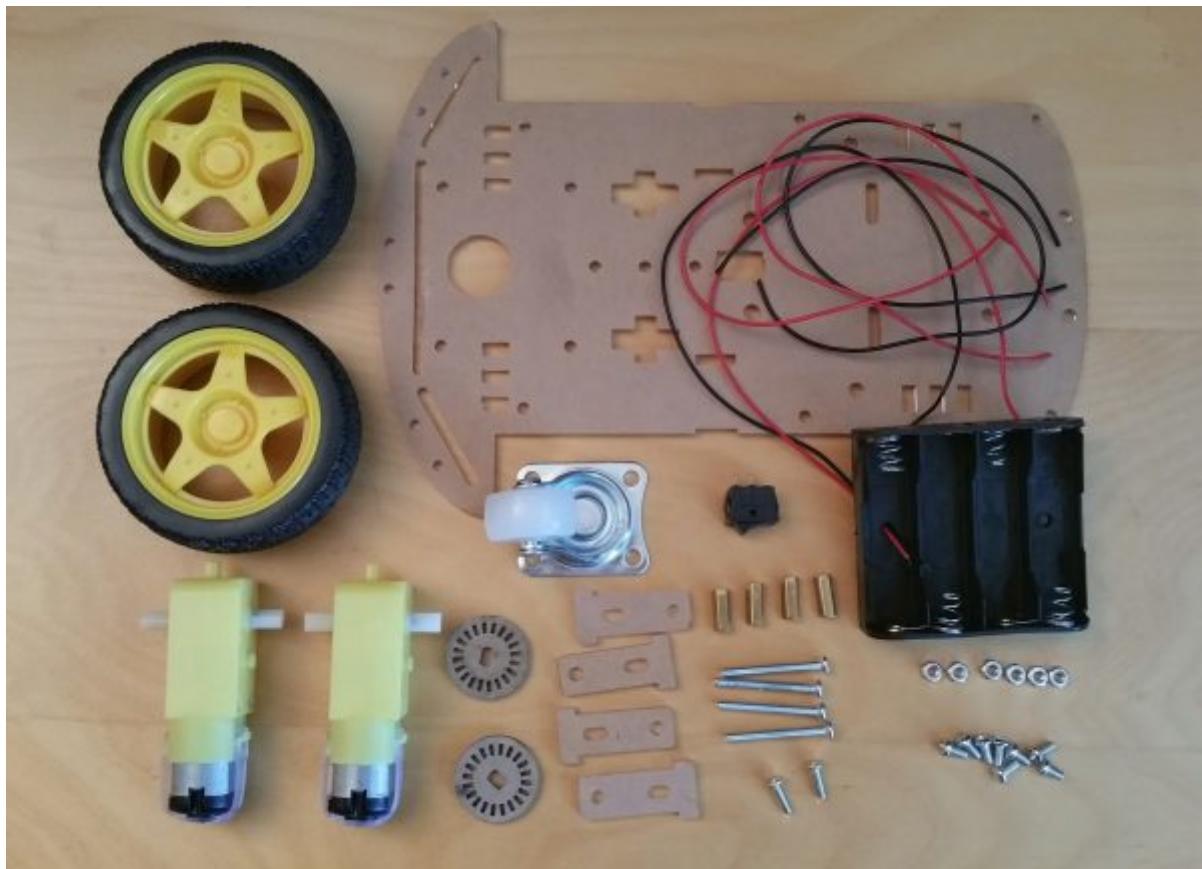
Bevor wir mit dem Zusammenbau beginnen, prüfe, ob du alle Komponenten hast!

Checkliste

- 1) Roboter Bausatz mit 2 Rädern + Motor
- 2) 4xAA Batterien (1.5V)
- 3) Ein/Aus-Schalter
- 4) Powerbank
- 5) Mini Breadboard
- 6) H-Brücke
- 7) Jumper Kabel + Widerstände
- 8) Heißklebepistole
- 9) Lötkolben
- 10) Schmiermittel
- 11) Doppelseitiges Klebeband
- 12) Marmeladenglas (=Halterung Powerbank)

Zusammenbau des Chassis

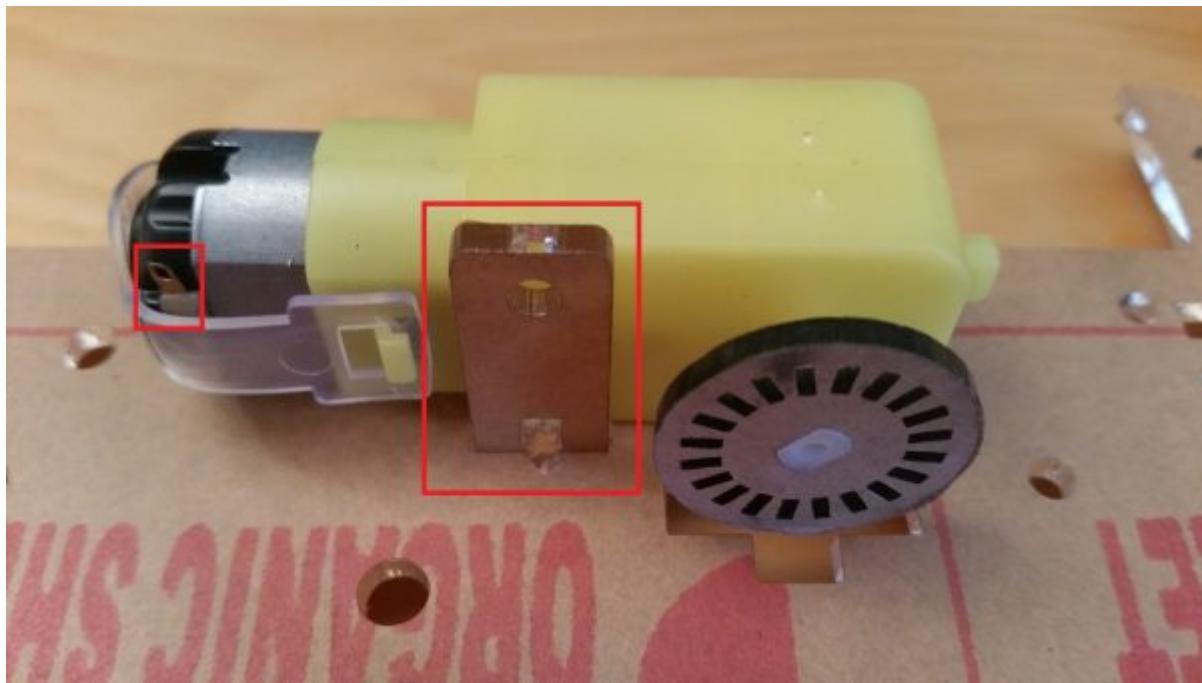
Neben den Rädern und Motoren kommt der Bausatz mit einigen Schrauben und Platten, welche wir zusammenbauen werden. Die enthaltenen Teile des zweirädrigen Bausatzes sind folgende:



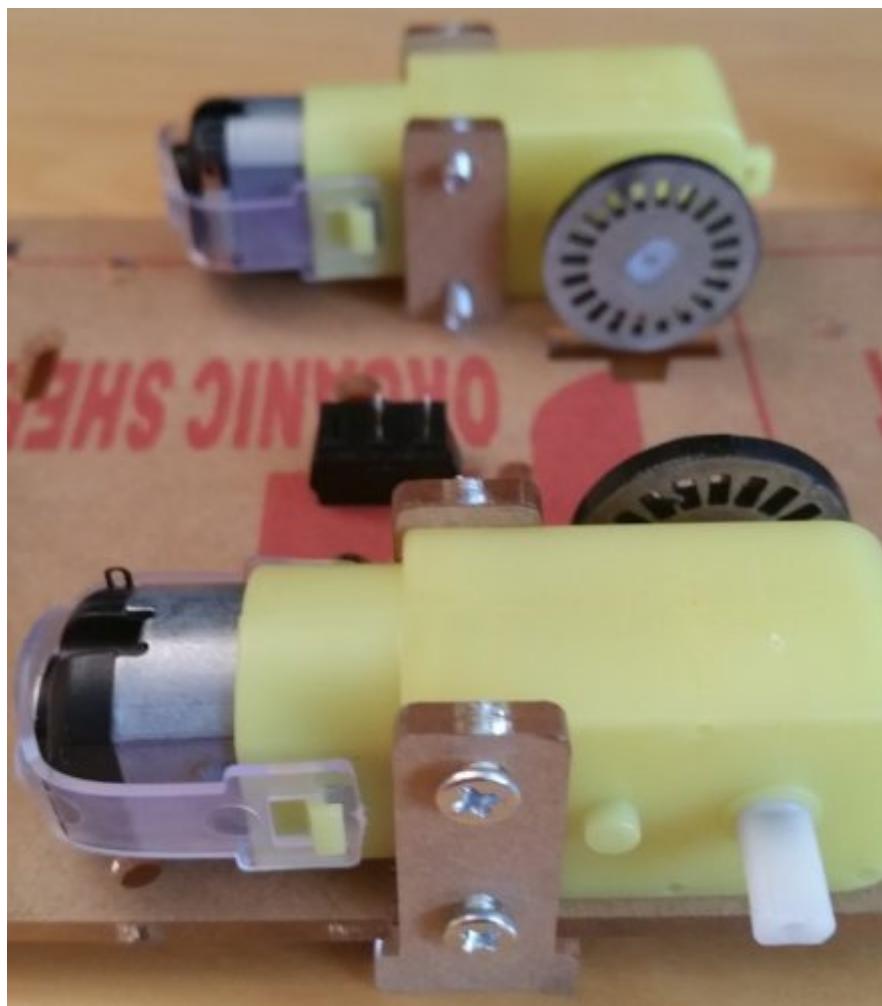
Als erstes werden wir die Motoren fest schrauben. Dazu nehmen wir erst einmal die beiden Motoren und befestigen die Rädchen daran:



Mit Hilfe der **4 langen Schrauben sowie wie den 4 kleineren Platten** werden die Motoren an der großen Platte befestigt. Dazu wird auf mittig beiden Seiten eine Platte durch das passende Loch geschoben. Der Motor wird daneben platziert. Es muss darauf geachtet werden, dass die Verbindungen beider Motoren jeweils nach innen zeigen.



Auf den äußeren Seiten werden die verbliebenen zwei Platten platziert und die langen Schrauben (von außen) durchgeführt. Innen werden die kleinen Muttern angebracht und damit die Motoren befestigt.

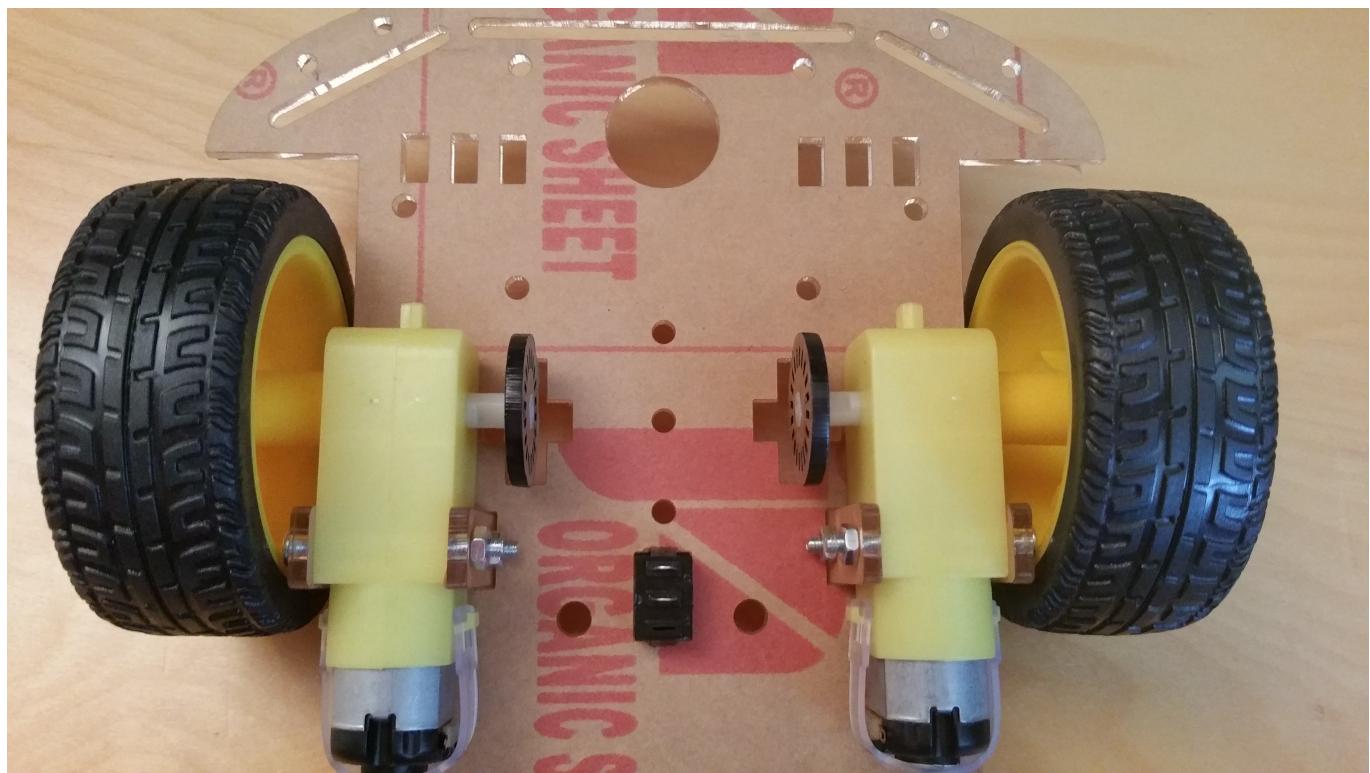


Falls du die Batterien anstelle einer Powerbank verwenden willst, kannst du den Schalter noch in das mittige Loch stecken.

Als nächstes wird das Hinterrad befestigt. Wir nehmen uns dazu das **drehende Rad**, die 4 Abstandshalter sowie **8 Schrauben**. Ich empfehle das Rad etwas zu **ölen**, bevor es befestigt wird, da es sich manchmal nur schwer bewegen lässt.



Die vier Abstandshalter werden zuerst an das Rad geschraubt. Anschließend werden sie mit den restlichen Schrauben vorne an der Platte befestigt. Außerdem kannst du die Hinterräder an den Motoren befestigen.



Nun müssen nur noch jeweils zwei Kabel an die „Laschen“ (siehe 2. Bild sowie 3. Bild, linke Seite) des Motors gelötet werden, was ein wenig Feingefühl benötigt. Wenn du die Motoren testen möchtest kannst du anschließend je ein Kabel an 3.3V (Pin 1) vom Raspberry Pi heften und eines an GND (Pin 6). Der Motor sollte sich nun bewegen (ob vorwärts oder rückwärts spielt erst einmal keine Rolle).

Nachdem die Motoren einen höheren Strom brauchen, als der Raspberry Pi zur Verfügung stellen kann werden Batterien benötigt. Dazu muss der Batteriehalter mit den verbleibenden zwei Schrauben und Muttern auf der vorderen Unterseite der Platte befestigt werden. Die Kabel können durch das kleine runde Loch in der Mitte nach unten verlegt werden:

BATTERIENBILD

Am Ende werden noch die H-Brücke und das Marmeladenglas für die Powerbank am Raspberry PI fixiert.

BILD RASPBERRY PI

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:infoprojekt_2017_18:03:03_01

Last update: **2018/04/12 15:45**