

Zusatzmaterial zu: Der Raspberry Pi im Unterricht

Inhalt

Vorwort	2
Einführende Hinweise zum Umgang mit dem Raspberry Pi	2
Das OS Raspbian installieren	2
VNC Viewer verwenden	3
Scratch Version 2.....	4
Scratch Version 3.....	6
Grundlegende Beispiele	7
Das Breadboard.....	7
Der Cobbler	8
Zwei Leuchtdioden	8
Zwei LED mit Schalter	10
Die Ampelschaltung	11
Motorsteuerung mit Relais	12
Einsatz eines IC - Motorsteuerung mit L293D und L298N	13
Funktionsweise und Belegung des L293D	15
Der Stepper – Ein genau steuerbarer Motor	16
Verwendung eines Sensors	18
Das Projekt Flipper	19
Das Projekt „Turm von Hanoi“	19
Anhang	20
Aufbau eines Relaisbausteins.....	20
Python	21
Umgang mit dem Raspberry Pi	23
Wächterkonfiguration und Schülerordner einrichten	23
Eigene Bauteile mit fritzing erstellen	25

Vorwort

Die Arbeit mit dem Raspberry Pi macht wirklich Spaß. Er bietet mit seinem Mikrocontroller eine Fülle von Möglichkeiten, interessante technische Experimente durchzuführen. Aber auch zum Erlernen von Programmier-Techniken eignet sich das System hervorragend. Schon die grafische Programmiersprache Scratch hat spannende Features und lässt es zu, Programme verschiedener Schwierigkeitsgrade und Qualität zu erstellen.

Mit Python ist der Übergang zu einem Interpreter mit Programmierumgebung möglich. Vorinstalliert ist die Umgebung Thonny, die sogar die Wahl zwischen drei verschiedenen Schwierigkeitsstufen lässt.

Das Gesamtpaket bietet höchste Flexibilität und ist somit an viele unterschiedliche Lerngruppen anpassbar.

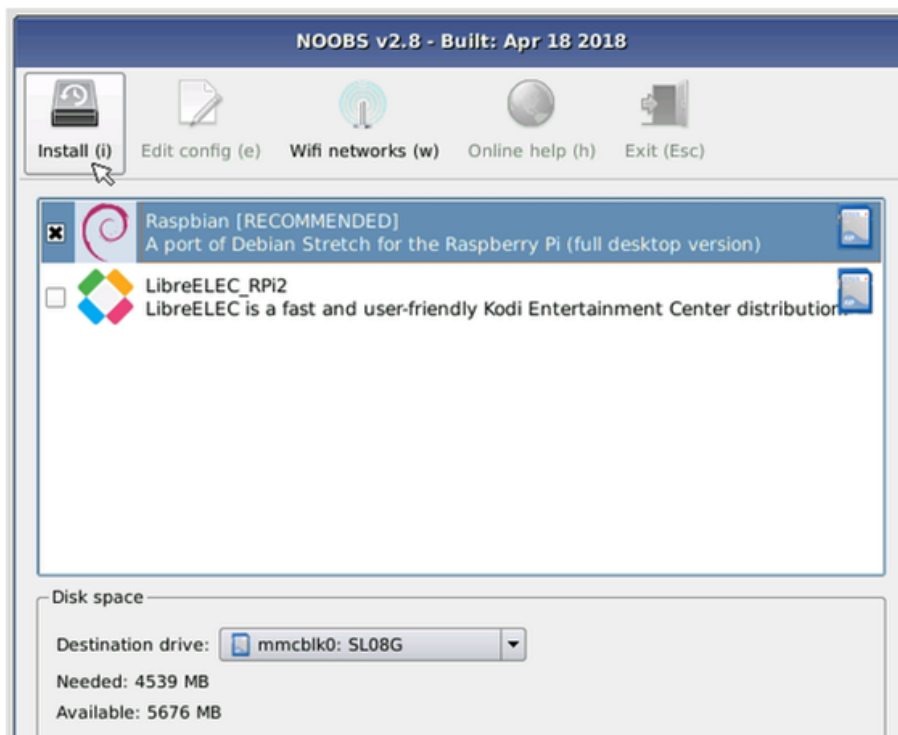
Einführende Hinweise zum Umgang mit dem Raspberry Pi

Das OS Raspbian installieren

Es gibt mehrere Möglichkeiten, das OS für den Raspberry Pi auf die SD-Karte zu bekommen. Eine Möglichkeit ist sicherlich NOOBS. Auf der zum Raspberry Pi gelieferten SD Karte ist oft NOOBS vorinstalliert. Dieses Programm ermöglicht die Auswahl eines Betriebssystems für den Raspi und die anschließende Installation desselben.

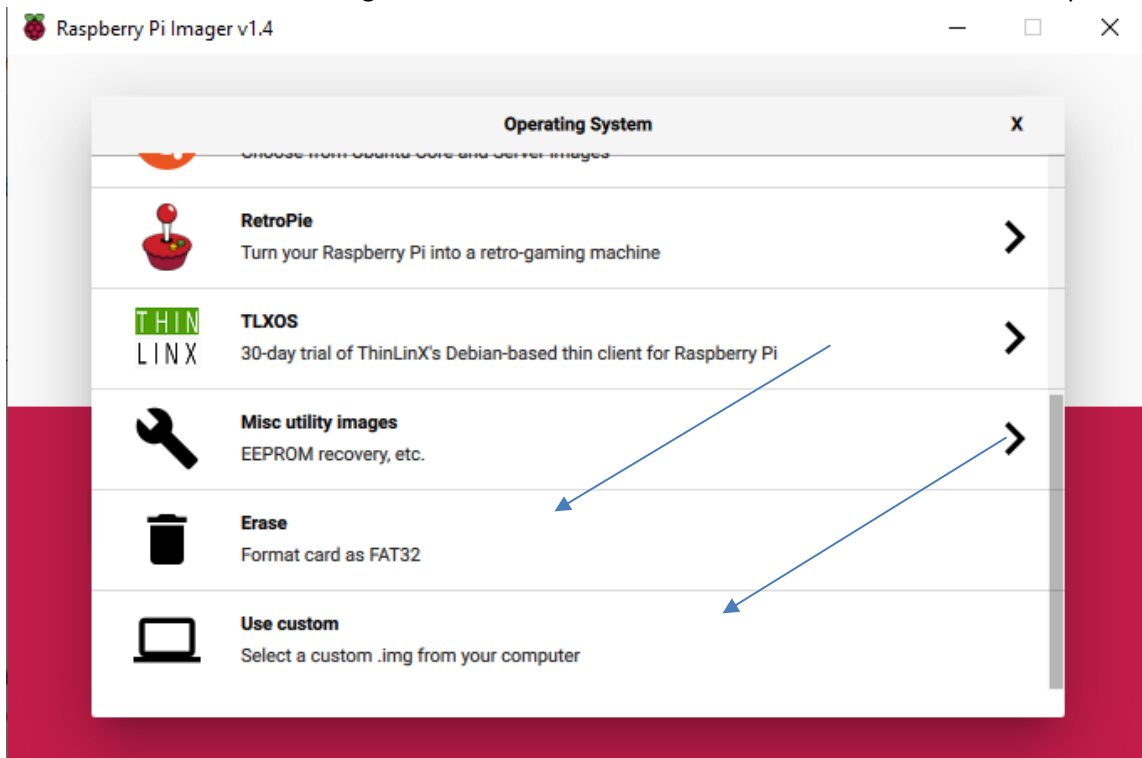
Hierzu wird die SD Karte in den Raspi, wie ich ihn im Folgenden immer kurz nennen werde, eingesteckt. Anschließend wird das Gerät mit allen notwendigen Peripheriegeräten sowie zuletzt mit dem Netzteil verbunden.

Es erscheint ein Bildschirm, der dem unter angezeigten stark ähnelt.



Unterhalb dieses Bildschirms sollte man die Sparschnecke wählen. Falls man andere Betriebssysteme verwenden möchte, muss man den Raspi mit dem Internet verbinden. Dann erscheinen die anderen Auswahlmöglichkeiten. Wir wählen Raspbian und klicken auf INSTALL. Nach erfolgreicher Installation der Software, die tatsächlich mehrere Minuten in Anspruch nimmt, wird man noch einmal nach den Grundeinstellungen gefragt. Sollte man nach dieser Einstellung die Installation von Updates erlauben und der Rechner mit dem Internet verbunden sein, dauert es wieder recht lange, bis der Vorgang abgeschlossen ist.

Eine weitere Möglichkeit Raspbian auf die SD-Karte zu bekommen, ist, sich die Version mit Desktop und Software von der Seite <https://www.raspberrypi.org/downloads/raspbian-pi-os/> herunterzuladen. Bei einem Windowsrechner wählt man die ZIP Datei, entpackt sie und hat das benötigte Image. Nun lädt man sich noch den seit März 2020 verfügbaren Raspberry Pi Imager herunter und installiert ihn. Mit dieser Software kann man sowohl die SD-Karten formatieren als auch ein Image auf eine SD-Karte aufspielen.



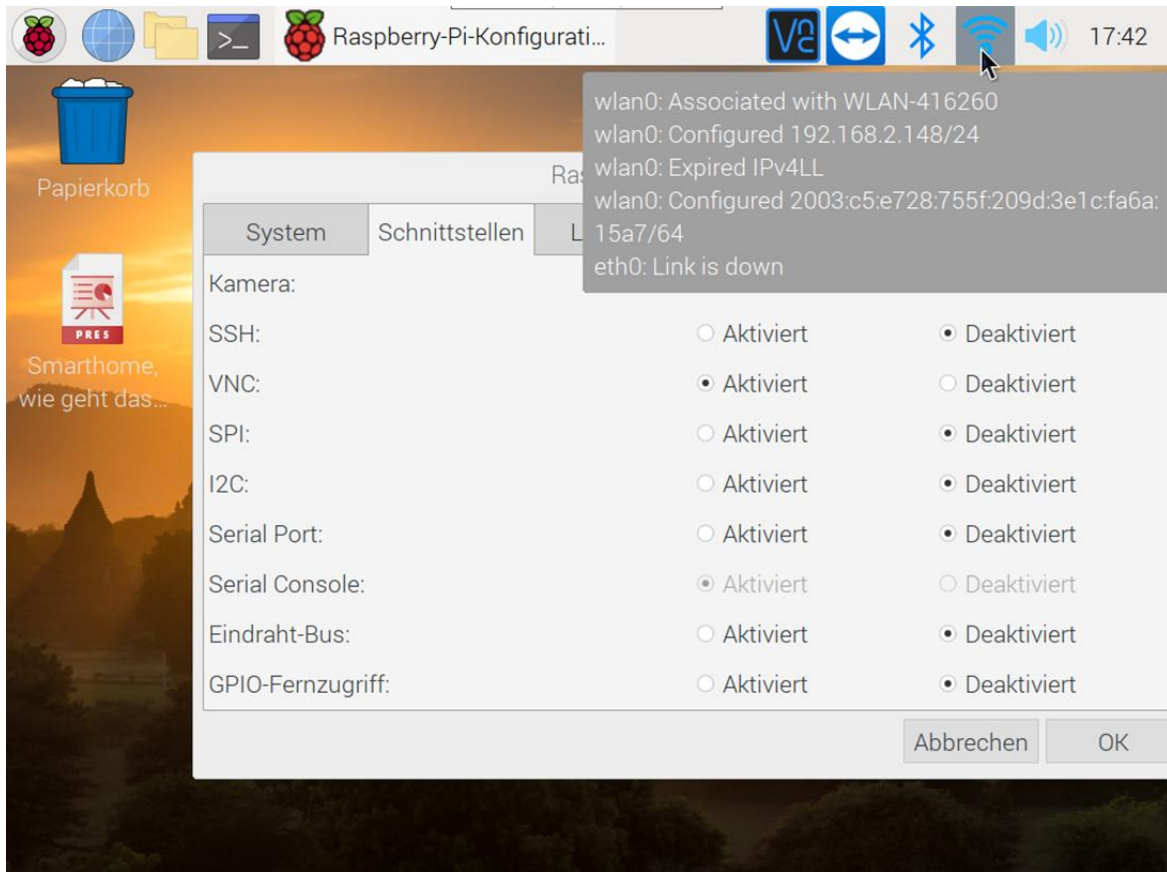
SINUS NRW – August 20

Nun kann man die SD-Karte in den Raspi stecken, ihn mit dem Netzteil verbinden und der Rechner fährt hoch. Man kann die Software anpassen, Dateien auf den Raspberry Pi aufspielen, zusätzliche Software installieren usw. Ist alles erledigt, kopiert man das System inklusive der Änderungen auf eine zweite SD-Karte mit einem USB-SD Card Adapter, indem man auf dem Raspi unter Himbeere-Zubehör-SD Card Copier den Kopierer startet und den Anweisungen folgt.

VNC Viewer verwenden

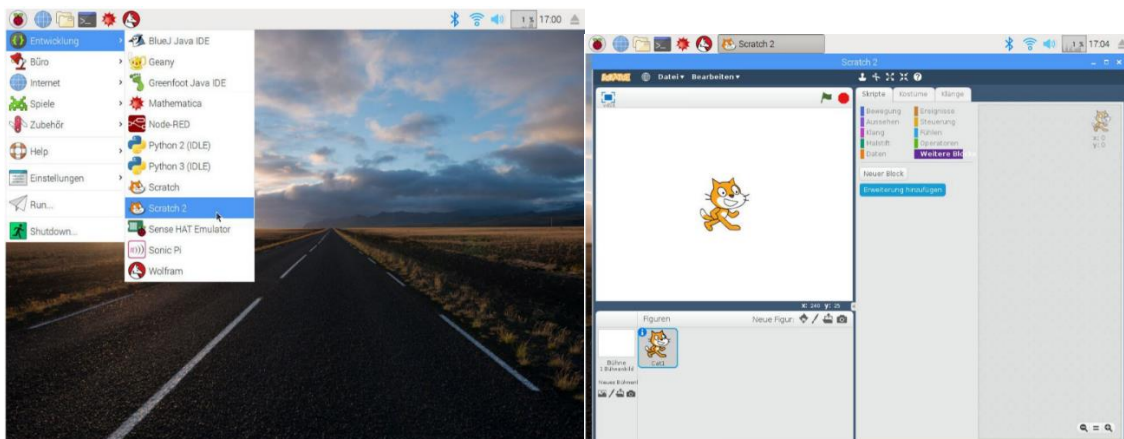
Unter Himbeere-Einstellungen-Raspberry PI Konfiguration-Schnittstellen kann man VNC aktivieren. Ist das geschehen, kann man von einem Windows Rechner über das Programm VNC VIEWER den Raspberry PI fernsteuern. Das ist sehr komfortabel und lässt sich für mehrere Raspis einrichten. So kann man als Lehrer steuernd eingreifen und die Schüler können ihre Ergebnisse präsentieren.

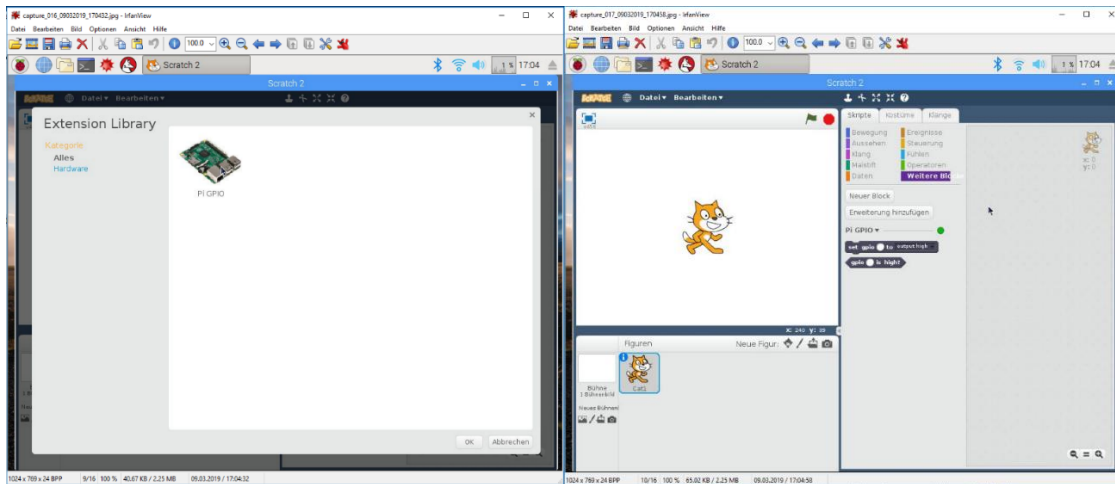
Für den Zugriff benötigt man die IP-Adresse des Raspberry PI. Man kann sie ablesen, indem man den Mauszeiger auf das WLAN Symbol positioniert. (Nicht klicken!!)



Scratch Version 2

In Scratch 2 ist die Handhabung der GPIO-Pins sehr einfach, man muss jedoch im ersten Schritt auf den analogen Teil verzichten, sprich, es funktioniert keine Pulsweitenmodulation. Darauf konnte in diesem Umfeld problemlos verzichtet werden; die notwendige Geschwindigkeitsreduzierung der Motoren wurde einfach durch die Variation der Betriebsspannung vorgenommen. Nun kommt man mit zwei Dingen für die Steuerung der GPIO Pins aus.





Nach dem Start von Scratch 2 geht man zu „Weitere Blöcke“ und wählt die Erweiterung für den Raspberry Pi.

Nun hat man einen neuen Befehlsblock und eine neue Abfrage zur Verfügung:



Das Beispiel zeigt das Prinzip. Wenn die grüne Fahne angeklickt wird, wird GPIO17 auf HIGH gestellt.


Wenn die Leertaste gedrückt wird, wird der GPIO10 laufend abgefragt. Ich kann ihn nun von außen steuern. Verbinde ich ihn mit einem Taster mit GND, dann sagt die Figur Hello! oder eben nicht.

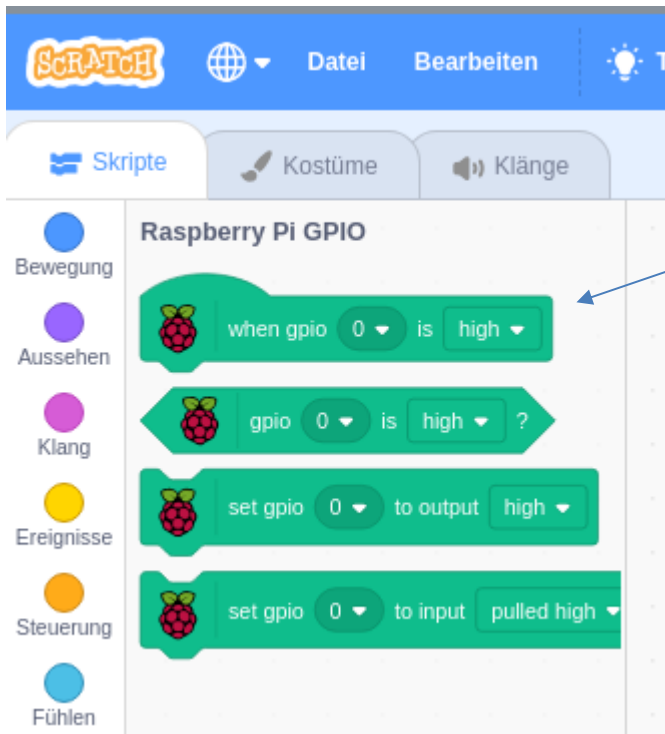
Man kann in SCRATCH2 den Ausgangszustand von einem INPUTPin nicht per Programm steuern, obwohl dies von der Hardware her möglich wäre. Um den Pin in einen definierten Zustand zu versetzen, benötigt man einen Pullup oder Pulldown Widerstand. Dieser Widerstand, ca. 300 Ohm, verbindet

nun den 3,3 V Ausgang mit dem Inputpin. Ein Taster oder Schalte verbindet den Pin mit Ground.

Scratch Version 3

Diese Version kann nur die GPIO-Pins des Raspberry PI 4 ansteuern.

Der Zugriff auf den Raspberry Pi erfolgt durch Anklicken des Feldes  unten links auf dem Startbildschirm. In der Programmierumgebung findet man sich genauso schnell zurecht wie in Scratch 2. Die GPIO Befehle sind erweitert worden.



Auffallend ist, dass es jetzt die Möglichkeit für Pullup und Pulldown Inputpins gibt. Außerdem gibt es nun ein Ereignis, das durch einen GPIO Pin beschrieben wird.

Grundlegende Beispiele

In diesem Kapitel werden wir uns einigen Beispielen widmen, die in die Verwendung einführen werden. Jedes dieser Beispiele ist sehr schnell ausgeführt und bietet eine Fülle von Variationsmöglichkeiten. So können Schüler auf der Basis dieser Schaltungen weitere Dinge planen und zur Ausführung bringen. Die Anwendungsbreite mit dem Relais ist sehr groß, bei fachfremden Kolleginnen und Kollegen muss natürlich eine Einführung in die RISU erfolgen.

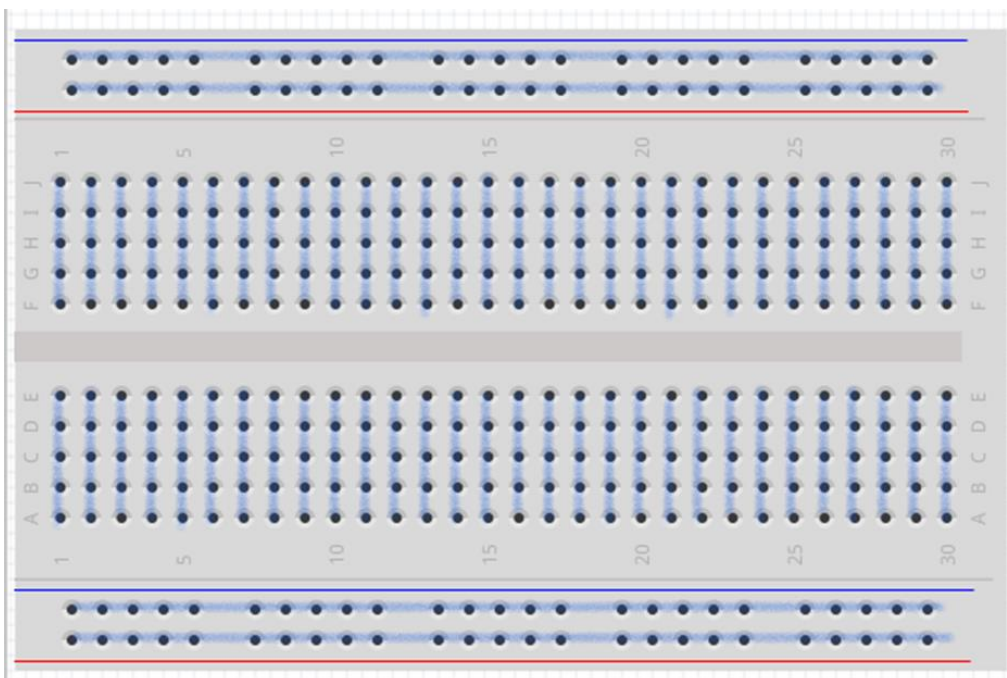
Intendiert ist hier nur, dass man schnell einige grundlegende Beispiele kennenlernt und zur Ausführung bringen kann. In den Beispielen werden teilweise Fischertechnikteile eingesetzt. Diese sind jedoch durch entsprechende Produkte im Handel ersetzbar.

Die realistischen Schaltbilder wurden mit Fritzing (einfaches googeln des Wortes liefert sofort einen Link) erstellt, was in späteren Kapiteln noch vorgestellt wird. Vorteile von Fritzing sind die realistische Darstellung und die Möglichkeit der Erstellung eigener Bausteile. Nachteile liegen in der Komplexität des Handlings. Außerdem wurde scheinbar die Entwicklung eingestellt.

In der vorliegenden Version von Fritzing ist der Raspi bereits enthalten, es empfiehlt sich aber, nach weiteren Bibliotheken Ausschau zu halten wie zum Beispiel der von Adafruit. Der folgende Link ist vom 23.10.2018. Die Einbindung in Fritzing erschließt sich dem Interessierten schnell, weshalb ich auf eine detaillierte Beschreibung des Vorgangs verzichte. (<https://learn.adafruit.com/using-the-adafruit-library-with-fritzing/download-the-fritzing-library-from-github>)

Das Breadboard

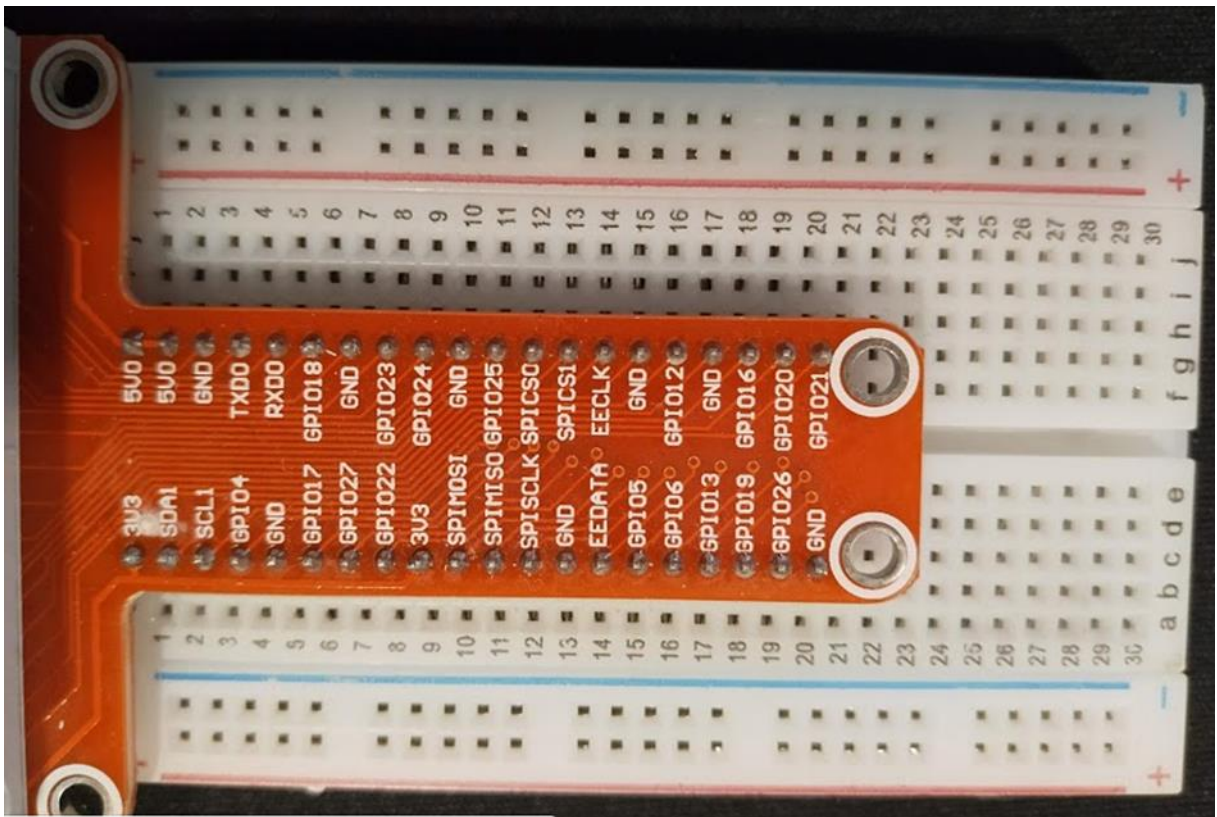
Alle Schaltungen werden auf einem Breadboard, einer Steckplatine, gesteckt. Hier muss man die Schülerinnen und Schüler sorgfältig anleiten, sodass sie sicher wissen, welche Leiterbahnen leitend verbunden sind.



Das Breadboard dient zur lotfreien Verbindung der elektronischen Bausteile. Die blauen Linien zeigen, welche Löcher elektrisch leitend miteinander verbunden sind.

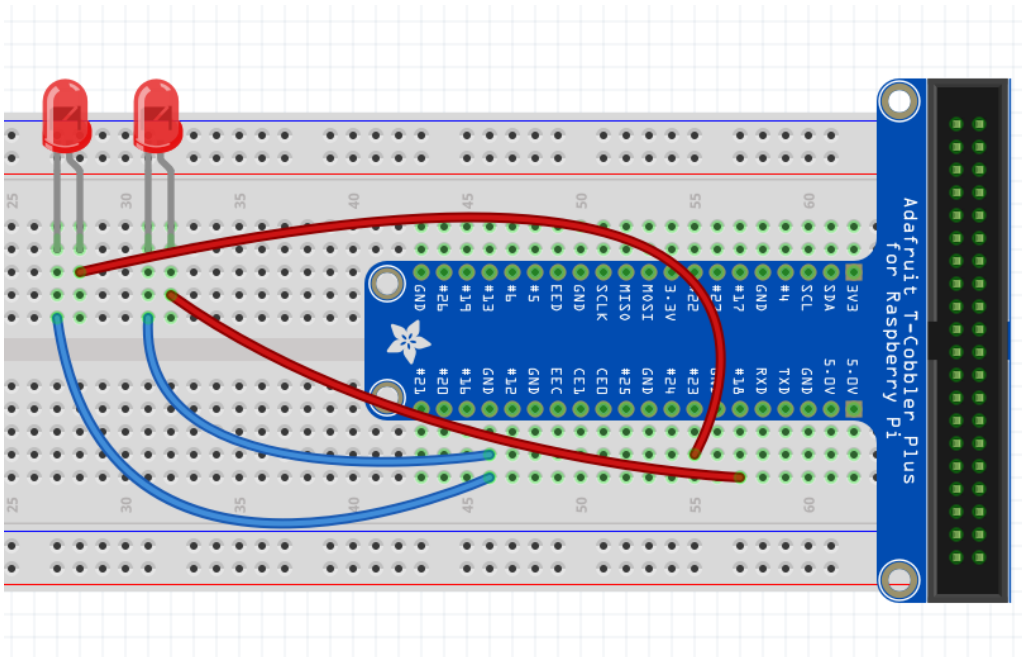
Der Cobbler

Im Zusammenhang mit dem Breadboard wird grundsätzlich ein Cobbler zum Raspberry Pi verwendet. Er ist annähernd vollständig beschriftet und erleichtert die Arbeit sehr.



SINUS NRW – August 20

Zwei Leuchtdioden



Die längeren Beinchen der LED müssen an +, also am Gpio-Out anliegen.

Programmierung in Scratch 2

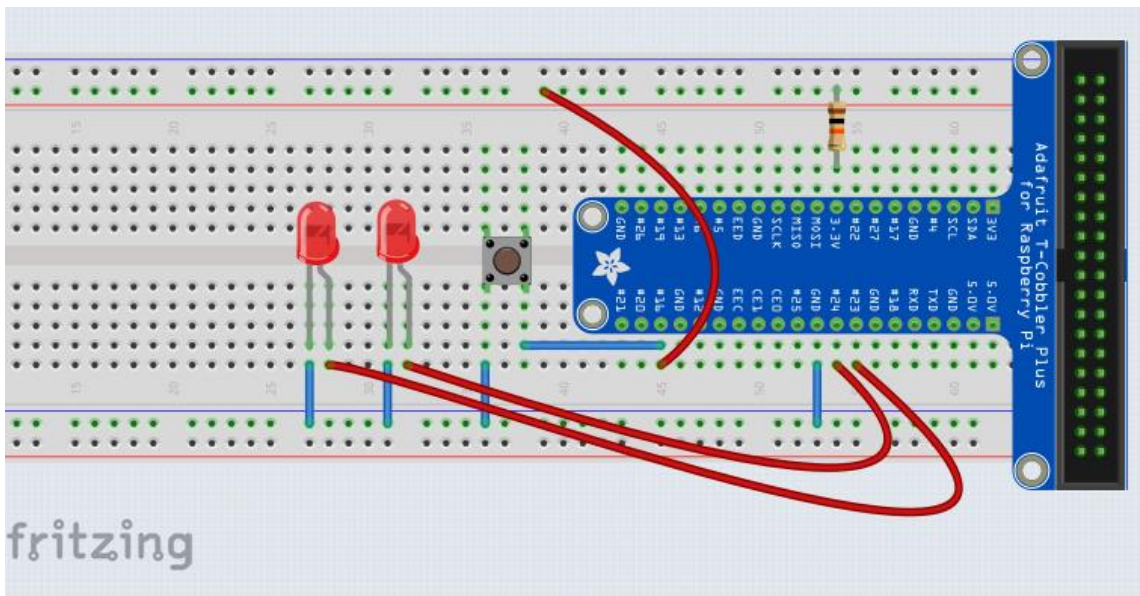


Zwei LED mit Schalter

Der 10 kΩ Widerstand schützt den Raspberry Pi. Der Inputpin ist sehr hochohmig, er würde den Strom nicht durchlassen. Aber wenn der Schalter gedrückt wird, liegen die 3,3 V direkt an Ground. Das kann Schaden anrichten.

Warum überhaupt der Aufwand. In Scratch 2 haben wir, im Gegensatz zu Scartch 3, noch nicht die Möglichkeit, den GPIO-Input Pin als High oder Low einzustellen. Er ist also in einem undefinierten Zustand. Wenn ich ihn aber über einen 10 kΩ Widerstand mit 3,3 V verbinde und der Input Pin einen annähernd unendlich hohen Eingangswiderstand hat, dann liegen an PIN 16 fast 3,3 V an, er ist high. Man kann sich das über einen Spannungsteiler klarmachen. Faustregel: Die hohe Spannung liegt immer am hohen Widerstand.

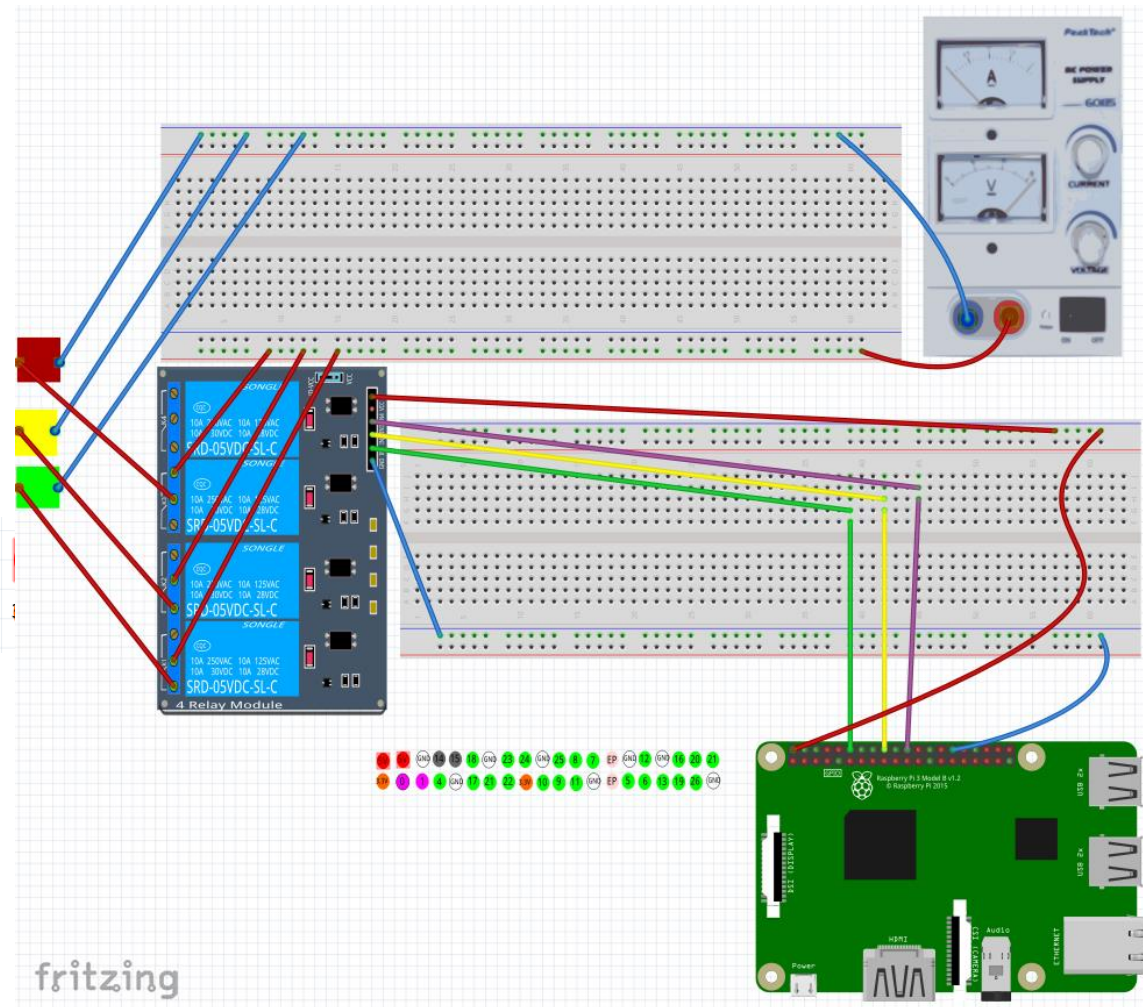
Drückt man nun den Taster, dann bricht die Spannung an PIN 16 zusammen und er ist low. Man nennt den Widerstand auch Pullup-Widerstand.



```
Wenn [Geheiß] angeklickt
  set gpio 16 to input
  wiederhole fortlaufend
    falls nicht gpio 16 is high? dann
      set gpio 23 to output high
      warte 1 Sek.
      set gpio 24 to output low
      warte 1 Sek.
    falls gpio 16 is high? dann
      set gpio 23 to output low
      warte 1 Sek.
      set gpio 24 to output high
      warte 1 Sek.
```

Die Ampelschaltung

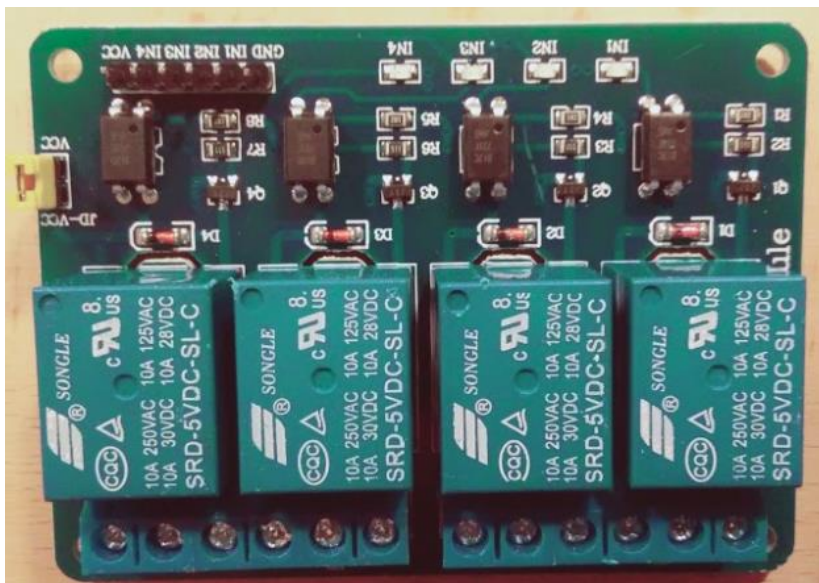
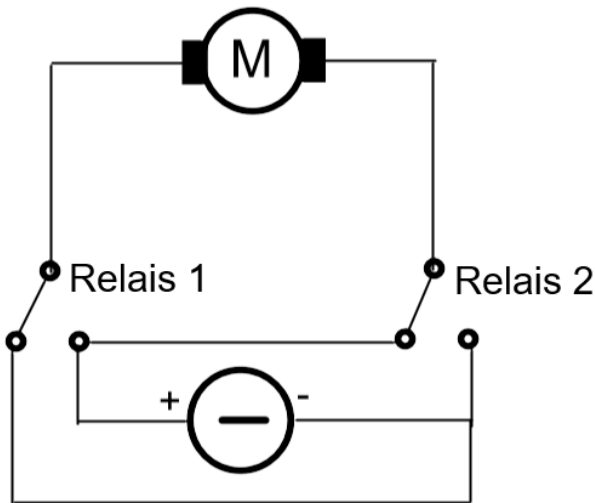
Will man Lampen steuern, deren Spannungsversorgung über den Möglichkeiten des Raspis liegt, kann man einen Relaisbaustein einsetzen. Die Schaltung ist unten ohne Cobbler dargestellt (sie stammt aus einer Zeit als wir noch ohne Cobbler gearbeitet haben), funktioniert aber natürlich auch mit Cobbler. Es ist sehr empfehlenswert mit zwei Breadboards zu arbeiten, damit die Schüler*innen die beiden Stromkreise deutlich unterscheiden können. Die Lampen sind Fischertechnik-Baukästen entnommen. Der Vierer-Relaisbaustein hat 6 Anschlüsse. In1 bis In4 werden mit vier GPIO-Outputpins verbunden. Für VCC kann man die 5V des Raspis verwenden, Ground wird mit GND verbunden. Durch Wechsel der GPIO's von High auf Low kann nun jedes Relais einzeln geschaltet werden. Im Schaltkreis besitzt der Relaisbaustein Wechselschalter.



Motorsteuerung mit Relais

Die Motorsteuerung mit dem Relais ist schon eine Aufgabe, die den Schülern viel abverlangt. Hier wird mit einem Gleichstrommotor von Fischertechnik gearbeitet. Das Problem ist, den Motor vorwärts und rückwärts laufen zu lassen. Hierzu werden zwei Relais des Bausteins benötigt, die folgendermaßen geschaltet werden müssen. Die Relais werden mit den GPIO-Pins gesteuert, was nun erlaubt, den Motor vorwärts und rückwärts laufen zu lassen.

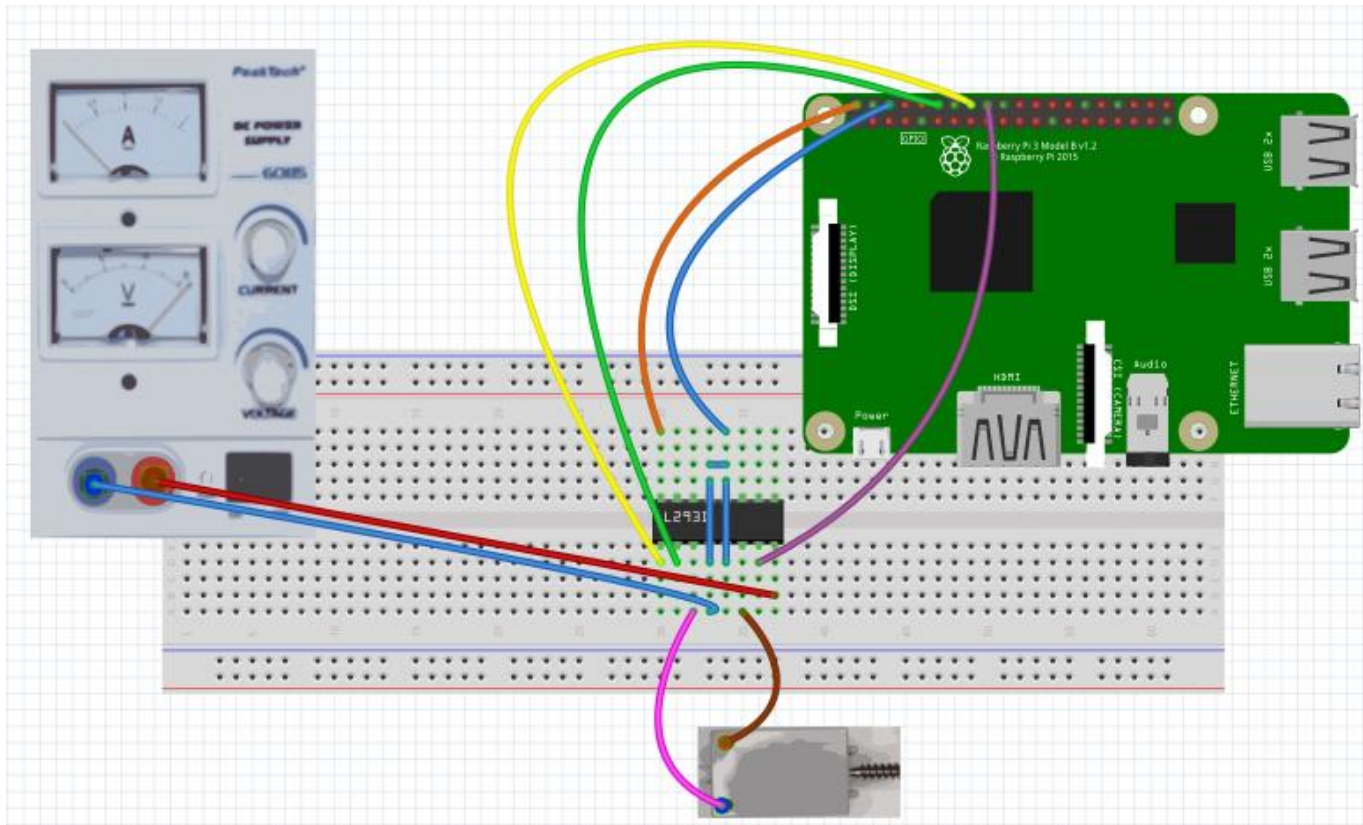
Es bietet sich an dieser Stelle an, mit dem Physikunterricht zu kooperieren.



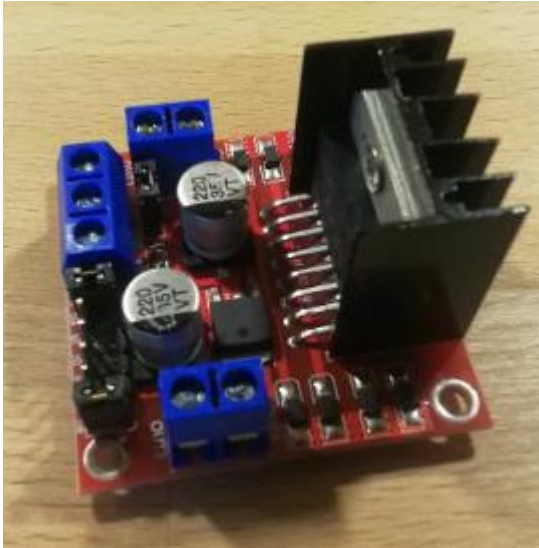
Einsatz eines IC - Motorsteuerung mit L293D und L298N

Jeder hat schon einmal einen IC gesehen und ihn unter Technikwunder abgetan. Doch wie werden diese IC's verwendet. Zur Herstellung von IC's kann man eine Fülle von Videos und Erklärungen im Internet abrufen. Der Einsatz eines Spezialisten, des L293D, löst unser Problem der Motorsteuerung auf sehr platzsparende Weise.

Leider wird der IC dabei auf die Dauer sehr heiß. Uns ist nach längerem Betrieb ein Breadboard angeschmolzen. Der IC ist also keine Dauerlösung.



Zur Lösung des Hitzeproblems bietet sich der Einsatz eines vorgefertigten Bausteins mit IC und Kühlung an. Wir haben uns für den L298N entschieden. Mit diesem Baustein gehören die Wärmeprobleme der Vergangenheit an.



An die blauen Outputs können zwei Motoren angeschlossen werden. Der blaue Dreierblock trägt von oben nach unten eine Spannungsversorgung, Ground und einen 5V Output, um einen Microcontroller mit Spannung versorgen zu können. Das ist beim Raspberry Pi natürlich nicht nötig.

Die vier freien Pins dienen zur Steuerung von Motor 1 und 2, analog zum L293D. Eine schöne Schüleraufgabe ist, herauszubekommen, welchen Zustand die Pins haben müssen damit sich die Motoren rechts- oder linksherum drehen.

Die Jumper sorgen dafür, dass die äußeren Pins mit 5V verbunden sind. Entfernt man die Jumper und legt eine Pulsweiten modulierte (kurz PWM) Spannung an, so kann man die Drehzahl steuern.

Funktionsweise und Belegung des L293D

Damit man die Anschlüsse identifizieren kann, hat jeder IC eine halbkreisförmige Einbuchtung. Sie liegt in unserem Bild links.

Die acht Anschlüsse des L293D reichen für zwei Motoren. Die einzelnen Pins haben folgende Funktion:

Pin 1 und 9 E_1 und E_2 sind die Ein und Ausschalter für die Motoren. Werden sie auf 1 gesetzt, läuft der entsprechende Motor.

Pin 4,5,12 und 13 werden an Ground angeschlossen. Hier liegt die gemeinsame Masse.

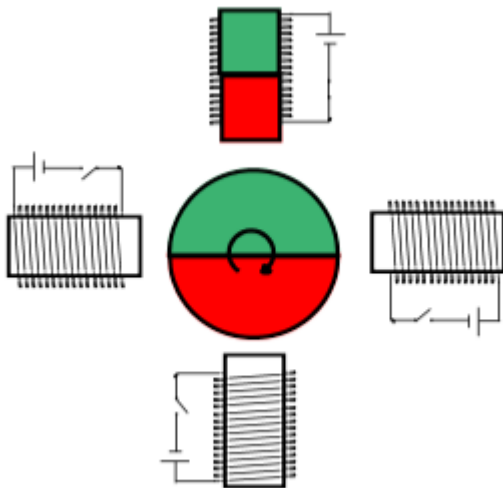
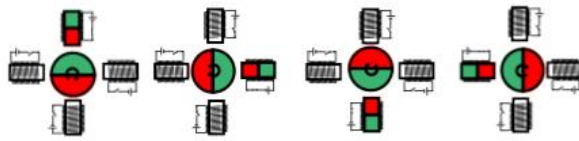
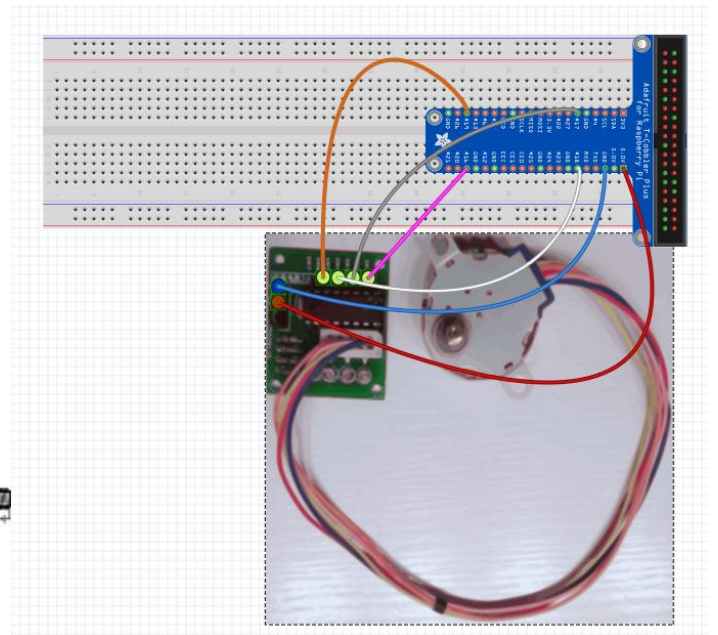
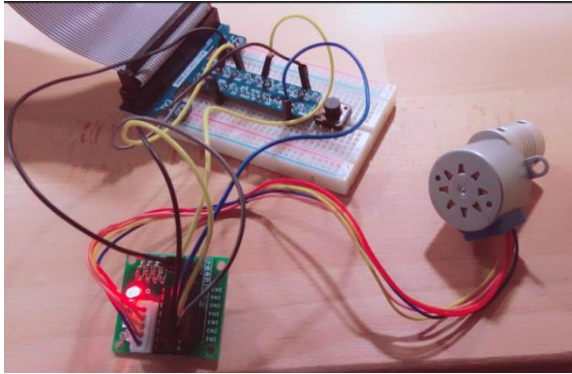
Pin 3 und 6 sind die Anschlüsse für Motor 1, Pin 11 und 14 die für Motor 2.

Pin 2 und 7 steuern Motor 1, Pin 10 und 15 steuern Motor 2. Liegt an Pin 2 High und Pin 7 Low, läuft der Motor in die eine Richtung, liegt an Pin 2 Low und Pin 7 High läuft er in die andere. Entsprechendes gilt bei Pin 10 und 15 für Motor 2. Über Pulsweitenmodulation kann die Drehzahl gesteuert werden. Hier muss man mit unterschiedlichen Werten für PIN 2 und PIN 7 bzw. Pin 10 und Pin 15 experimentieren.

Achtung, das Folgende darf nicht vertauscht werden: PIN 8 erhält die Spannungsversorgung der Motoren. Hier sind Werte bis ca. 30 V möglich. An PIN16 wird die Spannungsversorgung des L293D angeschlossen. Hier reichen die 5V des Raspberry Pi.



Der Stepper – Ein genau steuerbarer Motor



Der Stepper funktioniert so, wie schematisch dargestellt. Zunächst wird der erste Elektromagnet angesteuert, dann der zweite usw. Die längs magnetisierte Achse dreht sich. Wie schon beim Elektromotor haben wir einen IC, der das digitale Plus in den Einschaltvorgang des entsprechenden Elektromagneten übersetzt. Die Folge 1000 0100 0010 0001 ergibt den Zyklus einer Umdrehung. Wenn das Drehmoment so nicht ausreicht, der kann folgenden achtschrittigen Zyklus verwenden: 1000 1100 0100 0110 0010 0011 0001 1001. Die Umdrehung des Steppermagneten wird durch ein Getriebe übersetzt.

Bei unserem Motor müssen für eine Umdrehung 512 Zyklen durchlaufen werden. Um einen Winkel von ca 20° zu überstreichen, werden also $(\text{Anzahl Durchläufe } (20^\circ) = \frac{512}{360^\circ} \cdot 20^\circ \cong 28)$ 28 Durchläufe benötigt.

```

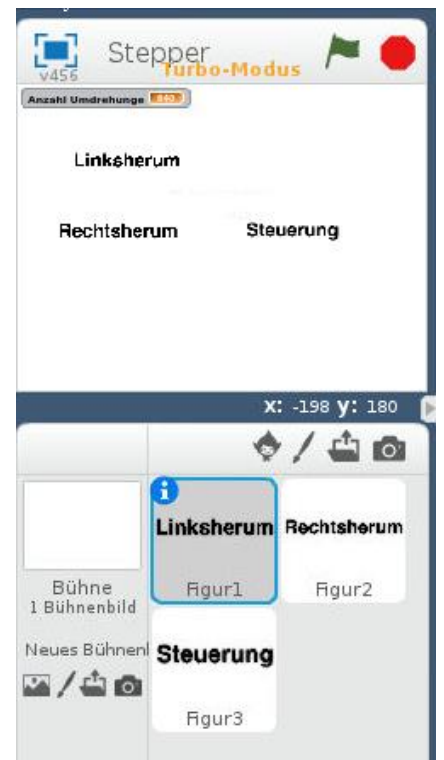
Wenn angeklickt
  frage Gib den orientierten Drehwinkel an und warte
  falls Antwort > 0 dann
    setze Anzahl Umdrehungen auf  $512 / 360 * \text{Antwort}$  gerundet
    sende rechtsherum an alle
  sonst
    setze Anzahl Umdrehungen auf  $-512 / 360 * \text{Antwort}$  gerundet
    sende linksherum an alle
  
```

```

Wenn ich linksherum empfange
  wiederhole Anzahl Umdrehungen mal
    set gpio 16 to output high
    set gpio 17 to output low
    set gpio 18 to output low
    set gpio 19 to output low
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output low
    set gpio 18 to output low
    set gpio 19 to output high
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output low
    set gpio 18 to output high
    set gpio 19 to output low
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output high
    set gpio 18 to output low
    set gpio 19 to output low
  warte 0 Sek.
  
```

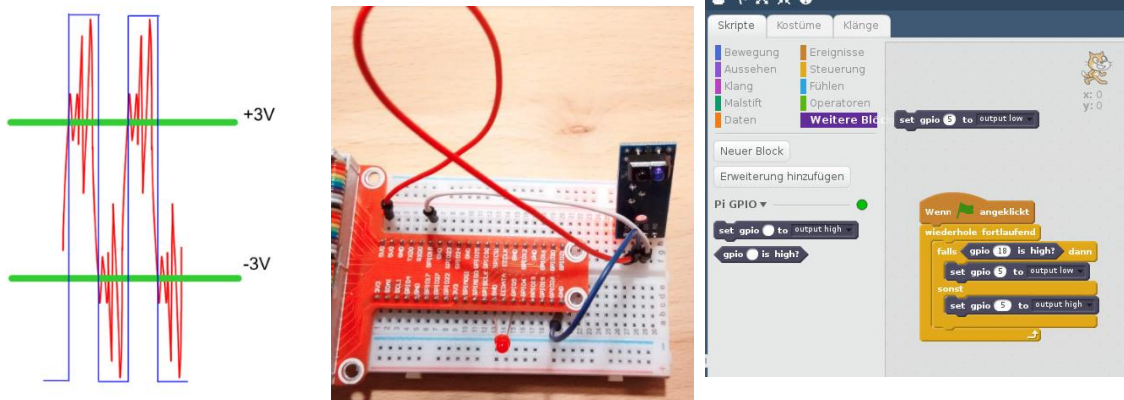
```

Wenn ich rechtsherum empfange
  wiederhole Anzahl Umdrehungen mal
    set gpio 16 to output high
    set gpio 17 to output low
    set gpio 18 to output low
    set gpio 19 to output low
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output high
    set gpio 18 to output low
    set gpio 19 to output low
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output low
    set gpio 18 to output high
    set gpio 19 to output low
    warte 0 Sek.
    set gpio 16 to output low
    set gpio 17 to output low
    set gpio 18 to output low
    set gpio 19 to output high
  warte 0.00001 Sek.
  
```



Verwendung eines Sensors

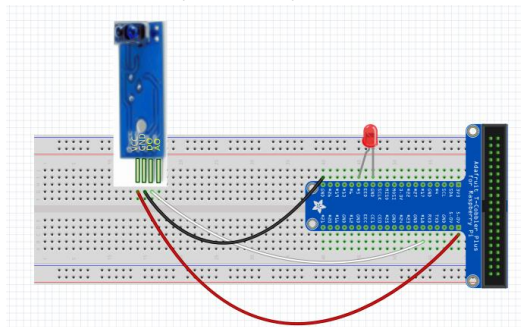
Wir haben uns einen Infrarotsensor angesehen. Man fragt sich ja, wieso sich die Bausteine, die als Sensor verwendet werden, alle so ähneln. Es liegt daran, dass die Signale, die von Infrarot-, Hall- und vielen anderen Empfängern kommen, nicht sauber sind, sie flackern sozusagen analog. Wir benötigen sie aber digital. Schön wäre es, wenn wir in einem Intervall schalten könnten. Beispiel: Ich lasse Licht auf einen lichtempfindlichen Widerstand fallen, die Stromstärke erhöht sich. Sie schwankt jedoch, da der Lichteinfall nicht stabil ist und der Widerstand auch nicht präzise arbeitet. Dann wäre es gut, wenn eine Schaltung z. Bsp. ab 3 Volt einschalten würde. Fällt die Spannung aber ein klein wenig unter 3 Volt, soll nicht sofort wieder ausgeschaltet werden. Schön wäre es, wenn erst ab z. Bsp. -3V wieder ausgeschaltet würde. Dann wären die Schwankungen um den Einschalt- und Ausschaltzeitpunkt verziehen (sh. Abbildung). Wir wollen also aus dem roten Signal das blaue Signal machen. Dieses Problem wurde durch den Studenten Otto



Schmitt gelöst, der eine Schaltung mit einem Operationsverstärker entwarf, die man Schmitt-Trigger nennt.

Diese Schaltung ist auf sehr vielen Sensorbauteilen mit einem OP- und einem regelbaren Widerstand enthalten. Unter https://www.flowcad.de/FlowCAD_Uni_Schmitt-Trigger.htm ist eine sehr gute Erklärung der Triggerschaltung abrufbar.

Die Verwendung des IR-Sensors und vieler anderer mit Schmitt-Trigger ist immer ähnlich. Es gibt meist vier Pins, ein VCC, der mit dem 5V-Ausgang des Raspi verbunden wird, ein GND, der mit



Ground verbunden wird, AO was für Analog OUT steht und hier nicht verwendet wird und DO, Digital OUT, was mit einem GPIO-Inputpin verbunden wird. Nähert man dem Sensor ein Hindernis auf ca. 1-2 cm, dann springt DO auf LOW. Die Entfernung kann über die kleine Schraube am Poti des Triggers eingestellt werden.

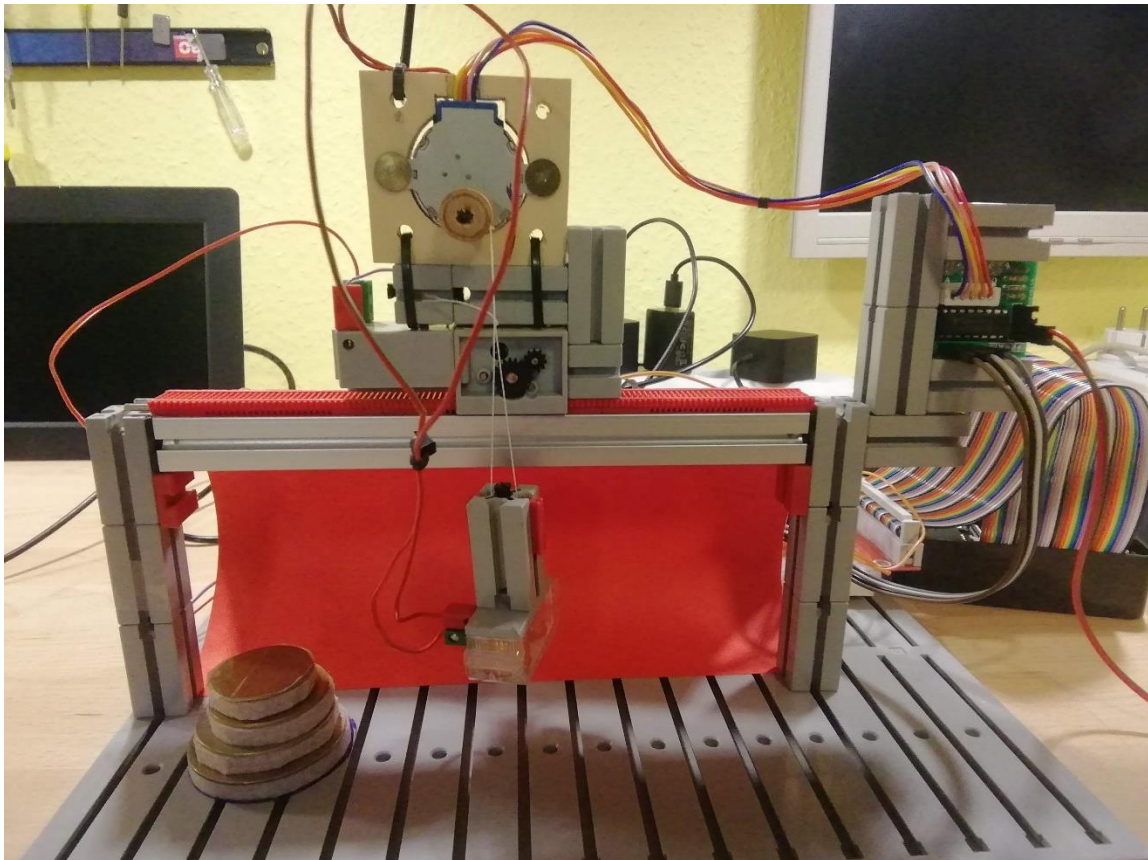
Das Projekt Flipper

Eine ausführliche Bauanleitung für den Flipper ist unter Zusatzmaterialien hinterlegt.



Das Projekt „Turm von Hanoi“

Das Projekt kann hier nicht umfassend beschrieben werden. Es soll aber ein Eindruck vermittelt werden, was das System leisten kann. Die einzelnen Komponenten des Roboters wurden teils selbst gefertigt, teils aus alten Fischertechnikbeständen entnommen.



Man sieht, dass hier ein Schrittmotor die Auf- und Abwärtsbewegung über eine Rolle und ein Band ermöglicht. Die horizontale Bewegung übernimmt ein Fischertechnik Motor mit Getriebe.

Die Positionen der Horizontalbewegung werden durch drei Schalter bestimmt, die von dem Schlitten ausgelöst werden. Als Greifer dient ein Elektromagnet.

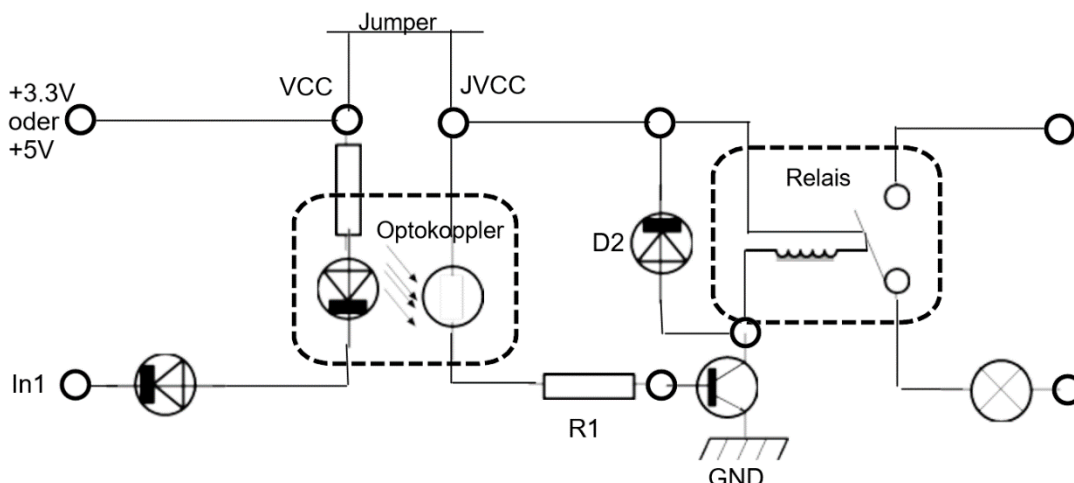
Damit man einen Eindruck davon bekommt, was in Python möglich ist, ist das Programm, das sicher noch viele Optimierungsmöglichkeiten bietet, ebenfalls in den Zusatzmaterialien enthalten. Es beinhaltet die wesentlichen Steuerungsmöglichkeiten für In- und Outputpins.

Anhang

Aufbau eines Relaisbausteins

Sollte man z. Bsp. in einem Elektronikkurs Bauteile wie den Transistor, das Relais und die Diode besprochen haben, so bietet der Relaisbaustein eine schöne Anwendung für dieses Wissen. Die belastete Spule als Elektromagnet und die Freilaufdiode, die den Induktionsstrom beim Ausschalten zur Spule zurückführt, sind in Anknüpfung an den Physikunterricht problematisierbar.

Das Schaltbild eines Relaisbausteins ist gar nicht einfach zu bekommen. Ich habe versucht, es mit den Mitteln einer einfachen Software nachzuzeichnen. In der Regel muss man die 5V Versorgungsspannung wählen, um das Relais durchzuschalten. Falls man viele Relais schalten will, sollte man beobachten, ob die Stromstärke, die beim Raspberry PI nicht überschritten werden sollte/kann, erreicht wird. In diesem Fall sollte man den Jumper ziehen und JVCC über eine externe Stromquelle versorgen. Ich kann keine seriöse Abhandlung über die verschiedenen Stromstärken des PI machen, aber man kann grob sagen, dass die 5V Anschlüsse vom Netzteil aus durchgeschleift werden, also die höchste Stromstärke bieten. Danach kommen die 3,3V Anschlüsse und danach die GPIO-PINS. Durch das Schaltbild sieht man sehr schnell die Funktionsweise. An In1 ist ein GPIO-Pin, z. Bsp. 18, angeschlossen. Wird dieser auf high gestellt, ist die Potentialdifferenz zwischen dem 5 bzw. 3,3V Eingang und In1 nicht hoch genug, um die LED des Optokopplers zum Leuchten zu bringen. Der Widerstandswert des lichtempfindlichen Widerstands gegenüber der Leuchtdiode (beides ist in einem Bauteil gekapselt) ist so hoch, dass kein Strom fließt und der Transistor nicht durchschaltet. Wird der GPIO-Pin auf Low gesetzt, reicht der Potenzialunterschied aus, die LED leuchtet, der Widerstandswert des lichtempfindlichen Widerstands ist sehr gering, es fließt ein Basisstrom für den Transistor und dieser schaltet durch. Das Relais zieht an, da nun ein Strom von JVCC über den Transistor nach GND fließen kann. D2 ist eine Freilaufdiode die den Induktionspeak beim Ausschalten aufnimmt. R1 begrenzt den Transistorstrom.



Python

Die Programmiersprache Python besitzt eine Fülle von weiteren Möglichkeiten zur Programmierung des Raspberry Pi. Hierzu können verschiedene Bibliotheken, die bereits fertige Module für die Programmierung der GPIO-Pins beinhalten, verwendet werden. Wir beschränken uns hier auf die Verwendung der Bibliothek RPi.GPIO. Diese ist standardmäßig auf dem Raspberry Pi installiert und muss nur noch zu Beginn jedes Python-Programms importiert werden. Ebenfalls bereits installiert ist die Programmierumgebung Thonny Python IDE, die wir benutzen werden.

Die Programmierumgebung Thonny

Bemerkenswert ist, dass Thonny drei verschiedene Modi besitzt, simple, regular und expert, die über Tools – Options – General anwählbar sind. Von simple nach regular geht es mit einem Schalter oben rechts, der Switch ro regular mode heißt. Im simple Mode hat man alles, was ein Programmieranfänger benötigt. Hat man einmal verstanden, was ein Debugger tut, ist die Befehlszeile selbsterklärend.



Kommunikation mit den GPIO-Pins

Voraussetzung ist, dass man das RPi Modul importiert. Um später Text zu sparen, bekommt es den kurzen Aliasnamen gpio.


```
import RPi.GPIO as gpio
```

gpio.setmode (gpio.BCM) dient dazu, für die Pins die gleichen Namen wie in Scratch2 verwenden zu können.

Die Inputpins können pullup oder pulldown voreingestellt werden. Der Name kommt aus der Elektronik in der Pullup und Pulldownwiderstände verwendet werden. Der Status des Inputpins, HIGH oder LOW, wäre ohne äußere Beschaltung unbestimmt. In Scratch2 hat man sich entschieden, als Standardstatus des Inputpins HIGH zu wählen. In Python ist durch den Zusatz pull_up_down=gpio.PUD_UP bzw. pull_up_down=gpio.PUD_DOWN im setup Kommando des Inputpins Beides möglich,

Nun kann es losgehen:

Python	Scratch2
<pre>gpio.setup(16,gpio.OUT) gpio.output (16,gpio.HIGH) gpio.output (16,gpio.LOW)</pre>	
<pre>gpio.setup(16,gpio.IN,pull_up_down=gpio.PUD_UP) gpio.setup(16,gpio.IN,pull_up_down=gpio.PUD_DOWN)</pre>	

<pre> if gpio.input(16)==1: print ("HIGH") time.sleep(2) else: print ("LOW") time.sleep(2) </pre>	
---	--

Beispiel1: Blinkende LED an Pin 23

```

import RPi.GPIO as gpio
import time
gpio.setmode(gpio.BCM)
gpio.setup(23,gpio.OUT)
for i in range(10):
    gpio.output(23,gpio.HIGH)
    time.sleep(1)
    gpio.output(23,gpio.LOW)
    time.sleep(1)
gpio.cleanup(23)
        
```

Ein Programmanfang, den man immer nutzen kann

Wenn wir ein Programm immer so beginnen lassen, kann man es durch ctrl-c beenden. Mit gpio.cleanup werden alle Pins wieder freigegeben.

```

# Wir importieren die Module, die wir benötigen.
import RPi.GPIO as gpio
import time
import signal
import sys
# Dieser Befehl ist sehr wichtig, da wir sonst mit falschen GPIO Nummern arbeiten!
gpio.setmode(gpio.BCM)

# Diese Funktion ist nur dazu da, damit nach ctrl-c alle Pins wieder freigegeben werden.
def keyboardInterruptHandler(signal, frame):
    print("ctrl-c wurde gedrückt. Das Programm wird beendet.")
    gpio.cleanup()
    sys.exit(0)

# Das ist ein Befehl, der bewirkt, dass das Programm fortwährend aufpasst ob ctrl-c gedrückt wird
signal.signal(signal.SIGINT, keyboardInterruptHandler)
        
```

Beispiel 2 Der Zustand von INPIN 16 wird abgefragt und ausgegeben


```

# Wir importieren die Module, die wir benötigen.
import RPi.GPIO as gpio
import time
import signal
import sys
# Dieser Befehl ist sehr wichtig, da wir sonst mit falschen GPIO Nummern arbeiten!
gpio.setmode(gpio.BCM)

# Diese Funktion ist nur dazu da, damit nach ctrl-c alle Pins wieder freigegeben werden.
def keyboardInterruptHandler(signal, frame):
    print("ctrl-c wurde gedrückt. Das Programm wird beendet.")
    gpio.cleanup()
    sys.exit(0)

# Das ist ein Befehl, der bewirkt, dass das Programm fortwährend aufpasst ob ctrl-c gedrückt wird
signal.signal(signal.SIGINT, keyboardInterruptHandler)

# 16 wird als Inputpin definiert.
inpin=16

#In Python kann man bestimmen, ob der PIN anschlussfrei auf HIGH oder LOW liegt.
gpio.setup(inpin,gpio.IN,pull_up_down=gpio.PUD_UP)

# Das ist das eigentliche Programm. Es ist leicht lesbar und gibt alle 2 Sekunden aus,
# ob der inpin HIGH oder LOW ist.
while True:
    if gpio.input(inpin)==1:
        print ("HIGH")
        time.sleep(2)
    else:
        print ("LOW")
        time.sleep(2)

```

Umgang mit dem Raspberry Pi

Man kann den Raspberry Pi direkt als Bundle bestellen. Bei der Wahl des Bundels muss man allerdings aufpassen, dass das Gehäuse den Zugriff auf die GPIO Pins auch mit einem Cobbler erlaubt. Deshalb sollte man ein Bundle wählen, das bereits einen Cobbler enthält. Um starten zu können, benötigt man noch einen Bildschirm, eine USB-Tastatur und eine Maus. Sollten die Bildschirme keinen HDMI Eingang besitzen, muss man noch ca. 10 € für einen HDMI-VGA Adapter einplanen.

Auf der SD-Karte des Raspberry Pi ist eine LINUX-Distribution aufgespielt. Es macht wenig Sinn, diese zu beschreiben, da man im Internet schon nach kurzer Suche Anweisungen findet, wie man ein neues System auf eine SD-Karte bekommt und was zu tun ist, wenn man eine Karte klonen möchte.

Im praktischen Umgang erweist es sich als sehr nützlich, wenn man einen USB-Stick verwenden kann und wenn man den Raspi soweit schützen kann, dass Änderungen am Erscheinungsbild und System zumindest erschwert werden.

Wächterkonfiguration und Schülerordner einrichten

Ziele

- Das Erscheinungsbild des Raspberry Pi soll nach jedem Neustart wieder gleich aussehen.
- Die Benutzer können nur in vier Ordnern speichern, die auf dem Desktop angezeigt werden.
- Die Benutzer können auf einem USB Stick speichern.

Vorgehen

Anlegen der Benutzer schueler und standardschueler


```
sudo useradd -m schueler
sudo passwd schueler
fsg
```

```
sudo useradd -m standardschueler
sudo passwd standardschueler
sesam
```

Der Benutzer schueler wird so eingerichtet, wie man es haben will. Zum Beispiel wird der Desktop eingerichtet und das Hintergrundbild gewählt.

Anlegen der Ordner

Unter dem Verzeichnis /dateien legen wir für jede Schülergruppe, die den Rechner nutzt, ein Verzeichnis an.

```
sudo mkdir -p /dateien/klasse1 /dateien/klasse2 /dateien/klasse3 /dateien/klasse4
```

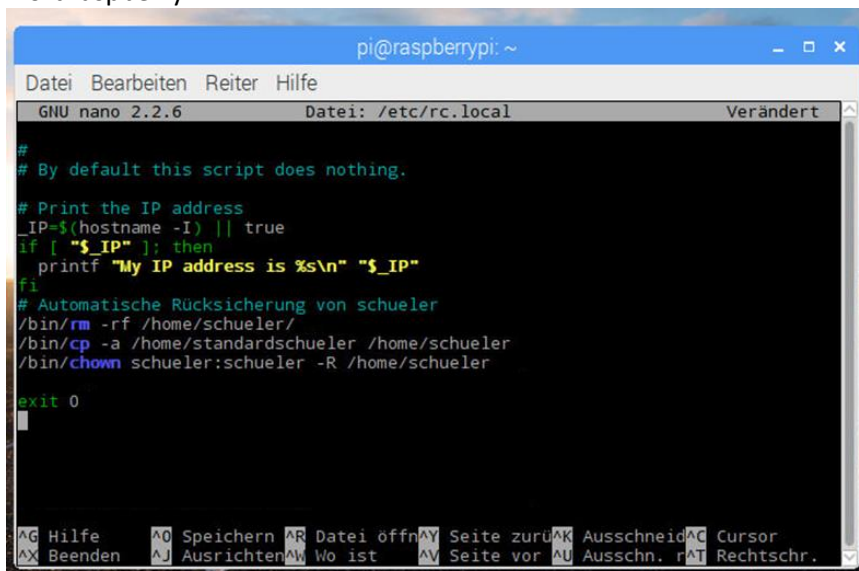
Der Benutzer schueler bekommt Schreibrechte für diese Dateien.

```
sudo chown -R schueler:schueler /dateien/klasse1
sudo chown -R schueler:schueler /dateien/klasse2
sudo chown -R schueler:schueler /dateien/klasse3
sudo chown -R schueler:schueler /dateien/klasse4
```

Anmeldung bei schueler und standardschueler

Um das Homeverzeichnis von schueler und standardschueler aufzubauen und zum Beispiel den Ordner „Schreibtisch“ anzulegen, meldet man sich einmal unter den jeweiligen Kennungen an. Hierzu muss man zunächst im Hauptmenü unter *Raspberry-Pi-Konfiguration System* das Häkchen bei *Automatische Anmeldung Als Benutzer ‚pi‘ anmelden* entfernen. Danach meldet man sich bei beiden Benutzern an und legt beim Benutzer schueler die gewünschten Einstellungen für das Hintergrundbild u. a. fest.

Abschließend meldet man sich wieder beim Benutzer pi an. Dieser hat standardmäßig das Kennwort raspberry.



```
pi@raspberrypi: ~
Datei Bearbeiten Reiter Hilfe
GNU nano 2.2.6 Datei: /etc/rc.local Verändert
#
# By default this script does nothing.
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
# Automatische Rücksicherung von schueler
/bin/rm -rf /home/schueler/
/bin/cp -a /home/standardschueler /home/schueler
/bin/chown schueler:schueler -R /home/schueler
exit 0
AG Hilfe AO Speichern AR Datei öffn AY Seite zurück AK Ausschneid AC Cursor
AX Beenden AJ Ausrichten AW Wo ist AV Seite vor AU Ausschn. r AT Rechtschr.
```

Anlegen der Verweise auf die Ordner

Auf dem Schreibtisch von schueler legen wir Verknüpfungen zu diesen Dateien an.

```
sudo mkdir -p home/schueler/schreibtisch  
sudo chown -R schueler:schueler /home/schueler/schreibtisch
```

```
cd /home/schueler/Schreibtisch/  
sudo ln -s /dateien/klasse1 Klasse1sudo  
sudo ln -s /dateien/klasse2 Klasse2  
sudo ln -s /dateien/klasse3 Klasse3  
sudo ln -s /dateien/klasse4 Klasse4  
cd
```

Einmalige Sicherung der Einstellungen von schueler

Wir kopieren den Inhalt von home/schueler, er enthält die Einstellungen, in den Ordner home/standardschueler.

```
sudo cp -a /home/schueler/* /home/standardschueler
```

Implementierung der Rücksicherung beim Hochfahren

Wir passen die /etc/rc.local an, indem wir das Rückspielen der Sicherung implementieren.

```
sudo nano /etc/rc.local
```

Die folgenden drei Zeilen werden **vor** dem **exit 0** eingefügt

```
/bin/rm -rf /home/schueler/  
/bin/cp -a /home/standardschueler /home/schueler  
/bin/chown schueler:schueler -R /home/schueler  
<strg> o und anschließendes RETURN speichert  
<strg> x beendet den Editor
```

Ändern des Kennwortes und des Bootverhaltens

Jetzt muss noch das Kennwort für den Benutzer PI geändert werden und in den Raspberry Pi Einstellungen der automatische Start des Benutzers PI verhindert werden.

Alle künftigen Änderungen können vorgenommen werden, indem man sich beim Benutzer standardschueler anmeldet und die Änderungen vornimmt.

Eigene Bauteile mit fritzing erstellen

Bei den Bildern oben ist ihnen sicher aufgefallen, dass das Netzteil und der Fischertechnikmotor sehr speziell sind. Die Fritzing-Bauteile kann man sich selbst erstellen. So können Schüler*innen leichter mit den Zeichnungen umgehen und erkennen ihre Geräte und Bauteile in den Grafiken leichter wieder.

Vektorgrafiken mit INKSCAPE

Zur Erstellung von Fritzing-Bausteinen werden Vektorgrafiken benötigt. Diese können aus z. Bsp. png Bildern (selbst fotografiert) mit INKSCAPE erstellt werden. Hierbei ist folgendermaßen vorzugehen:

1. Es ist zu überprüfen, wie groß das png-Bild ist. Größen über 10 cm machen hier wenig Sinn und verlangsamen die Bearbeitung mit INKSCAPE spürbar. Um ein Gefühl für die Größe zu bekommen und um den Lochabstand des Breadboards als Vorlage zu haben, kann man sich in der Standardinstallation unter C:\Program Files (x86)\Fritzing\fritzing.0.9.3b.64.pc\fritzing-parts\svg\obsolete\breadboard\LED-5mm-red.svg die LED in Inkscape laden.
2. Das eigene Bild wird in INKSCAPE über <Datei> <öffnen> geladen.
3. Anschließend <Bearbeiten> <Seitengröße auf Auswahlgröße> wählen.
4. Das Bild durch Linksklick auswählen.
5. <Pfad> <Bitmap nachzeichnen> mit ausgewähltem Button o-Farbe, liefert die gewünschte Vektorgrafik. Sie liegt auf dem ursprünglichen Bild und kann bei gedrückter Maustaste bewegt werden. Das Ursprungsbild kann nun gelöscht werden.
6. Nun zeichnet man mit INKSCAPE Figuren (Kreise, Rechtecke, Quadrate, ... auf die Vektorgrafik. Sie bilden später in Fritzing die Anschlüsse, die zum Beispiel mit Kabeln verbunden werden können.
7. Die einzelnen Objekte werden verbunden, indem man oben links den Auswahlpfeil aktiviert, die Objekte mit gedrückter linker Maustaste umfährt und loslässt.
8. Ein Rechtsklick auf eine der Komponenten der Grafik liefert ein Auswahlmeneue, in dem Gruppieren gewählt wird. Damit ist die Grafik fertig und kann als svg-Datei gespeichert werden.

Für Fritzing werden zwei Grafiken benötigt, eine als Bauteilbild und eine für das Ikon. Als Ikonbild kann auch das Bauteilbild verwendet werden, es ist aber auch möglich ein kleineres anderes Bild zu verwenden.

Bauteile in fritzing aus der Vektorgrafik erzeugen

Das Verfahren ist sehr ungewöhnlich. Um den Editor für die Bauteile aufzurufen, sucht man sich ein schon vorhandenes Bauteil aus, das genauso viele Anschlüsse wie das neu geplante Bauteil besitzt. Dieses wird rechts angeklickt und im Menü wird „Bauteil bearbeiten“ gewählt. Nun wird im Hauptmenü unter Datei-Bild zur Ansicht laden... gewählt und die oben erstellte Vektorgrafik ausgewählt. Unter Verbinder können nun die Stellen der Grafik ausgewählt werden, die die neuen Anschlüsse bezeichnen. Die Namen und die Beschreibungen der Anschlüsse werden angepasst. Mit Datei – Als neues Bauteil speichern ... wird das Bauteil abgespeichert. Der Vorgang wird für Symbol wiederholt.

Da wir die Schaltung nicht simulieren oder den Schaltplan verwenden wollen, reicht dieses Vorgehen aus. Wir haben nun ein Bauteil, das wir in unseren Bildern verwenden können. Der Weg ist unten noch einmal Schritt für Schritt dargestellt.

1. Fritzing starten und Steckplatine wählen.
2. Ein Bauteil wählen, das dem zu erstellenden möglichst ähnlich ist. Mindestens muss es die gleiche Anzahl von Anschlüssen haben.

3. RK auf des Bauteil und –Bearbeiten Neuer Bauteileeditor- wählen.
4. Den Reiter Steckplatine wählen und dann Datei – Bild zur Ansicht laden.
5. Nun wählt man die Vektorgrafik, die vorher in Inkscape erstellt wurde. Dabei wird die alte Grafik überschrieben.
6. Rechts sind die Verbinder eingeblendet. Das sind die Punkte, an die in der Grafik virtuelle Anschlüsse angelegt werden können. Diese kann man in der Vektorgrafik an allen Grafikelementen positionieren, die zusätzlich in die Zeichnung eingebracht wurden (sh. Erstellung der Vektorgrafik). Dazu den Namen des Anschlusses anklicken, Grafik auswählen anklicken und dann den Punkt in der Grafik anklicken, an dem der Anschluss liegen soll.
7. Nun wird der Reiter Symbol aufgerufen und mit Datei – Bild zur Ansicht laden wird ein neues Bild als Symbol gewählt. Man kann auch des Bild für die Steckplatine noch einmal verwenden.
8. Nun wird mit Datei Als neues Bauteil speichern en Name vergeben und das Bauteil gespeichert. Das Symbol erscheint nun unter MINE, also unter Meine Bauteile.