

Robotik und Raspberry Pi



BAG Tagung Informatik
3.10.2017
Klaus Misof

Übersicht

- Fach PC-Labor
- Lehrplan
- Umsetzung
- Chassis
- Motoren
- Sensoren
- Software

Lehrplan

PC-Labor

Zahlensysteme: Kennenlernen unterschiedlicher Zahlensysteme und Verwendung dieser in der Informatik

Digitale Logik und Schaltungen: Kennenlernen unterschiedlicher elektronischer Bauelemente, Schaltungstechnik, Automatentheorie, Löten von einfachen Schaltungen

Hardware in Produktivsystemen: Zusammenstellen, Verwaltung und Problemanalyse von PC-Hardware-Systemen

Betriebssysteme: Installation und Verwaltung verschiedener Betriebssysteme, Virtualisierung

PC-Labor

Partitionierung von Datenträgern: Verwaltung von Datenträgern für unterschiedliche Aufgaben, Kennenlernen unterschiedlicher Dateisysteme

Halbleiterelemente: Aufbau, Verwendung und Aufgabe von Halbleiterelemente (Diode, Transistor, etc.), Eigenschaften und Herstellung von Halbleiterelementen

Grundlagen der Elektronik: Strom, Spannung, elektrischer Widerstand, Darstellung und Berechnung von einfachen Schaltkreisen, Ohm'sches Gesetz, Kirchhoff'sche Regeln

Elektrische Bauteile: elektrischer Widerstand, Kondensator, Leuchtdiode, Schalter, etc.

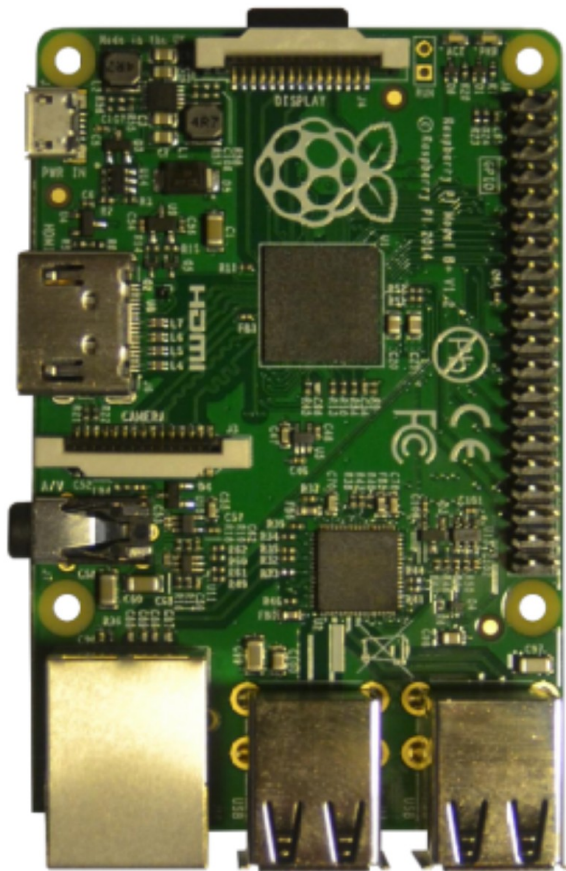
- **Arduino ?**
 - Mikroprozessor, sehr schnell, keine Installation des Betriebssystems notwendig
- **Raspberry Pi ?**
 - Einplatinencomputer, OpenSource, Betriebssystem mit oder ohne Grafik einsetzbar, hohe Flexibilität

- **Python**
 - Interpretersprache, schnelle Erfolge, gute Strukturierung
- **Scratch**
 - Grafische Strukturierung
- **C/C+/C++**
 - Compilersprache, Aufwand für Anfänger relativ hoch

Raspberry Pi

- Linux Betriebssystem
- Passt ins Konzept
- Hohe Flexibilität
- Verbindung der Fächer Informatik, Physik und Mechanik (TEW)
- Erweiterungsmöglichkeiten in der Oberstufe

GPIO Header



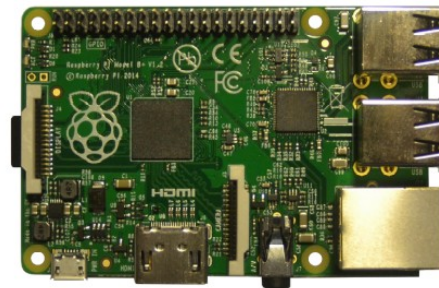
Raspberry Pi2 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	Yellow	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

Rev. 1
29/01/2014

<http://www.element14.com>

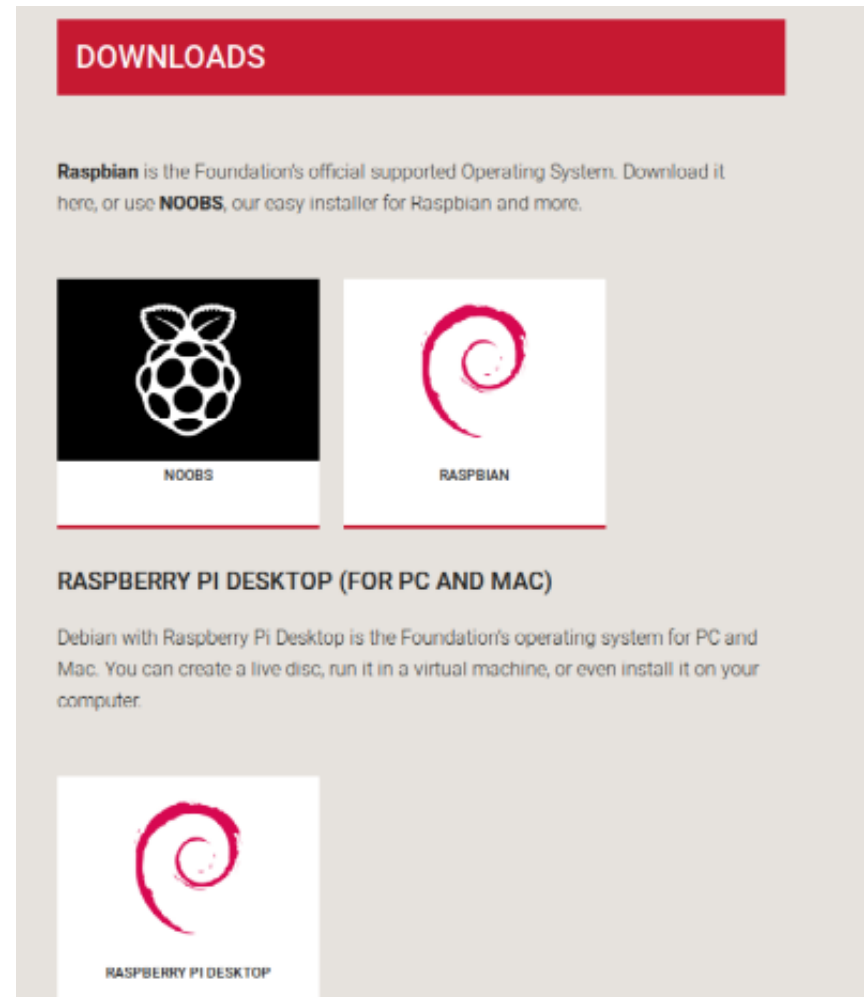
Kommunikation mit Pi



Betriebssystem Raspbian





<https://www.raspberrypi.org/downloads/>




DOWNLOADS

Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

 **NOOBS**  **RASPBIAN**

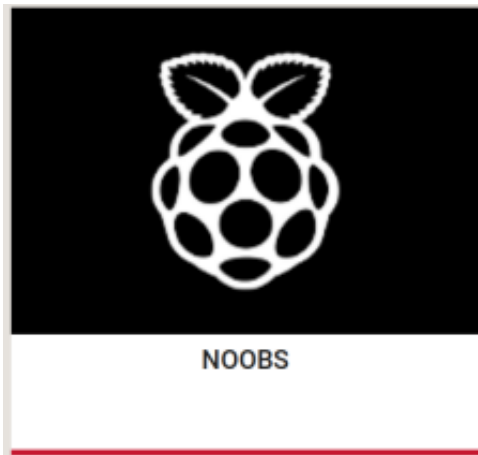
RASPBERRY PI DESKTOP (FOR PC AND MAC)

Debian with Raspberry Pi Desktop is the Foundation's operating system for PC and Mac. You can create a live disc, run it in a virtual machine, or even install it on your computer.

 **RASPBERRY PI DESKTOP**

Betriebssystem Raspian

<https://www.raspberrypi.org/downloads/>



Betriebssystem Raspbian

<https://www.raspberrypi.org/downloads/>



2017-09-07-raspbian-stretch.
zip

1,8 GB



2017-09-07-raspbian-stretch-
lite.zip

362,9 MB



```
dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdX conv=fsync
```

```
unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of=/dev/sdX bs=4M conv=fsync
```

```
dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdX status=progress conv=fsync
```

Problem: /dev/sdX?

- **Ohne ssd Karte: df -h**

Dateisystem	Größe	Benutzt	Verf.	Verw%	Eingehängt auf
udev	3,9G	0	3,9G	0%	/dev
tmpfs	787M	9,7M	778M	2%	/run
/dev/sda2	57G	11G	44G	20%	/
tmpfs	3,9G	51M	3,8G	2%	/dev/shm
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	3,9G	0	3,9G	0%	/sys/fs/cgroup
/dev/sda3	161G	80G	74G	52%	/home
/dev/sda1	511M	4,6M	507M	1%	/boot/efi
tmpfs	787M	68K	787M	1%	/run/user/1000
/dev/sdc1	15G	13G	2,7G	83%	/media/misof/KLAUS1
/dev/sdb2	691G	406G	251G	62%	/media/misof/casper-rw
/dev/sdb3	686G	240G	446G	35%	/media/misof/KLAUS-DATEN-A

```
dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdX conv=fsync
```

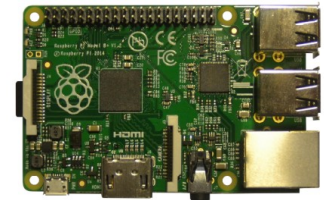
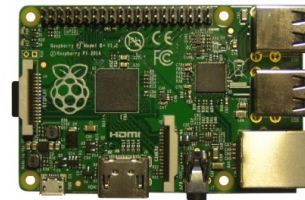
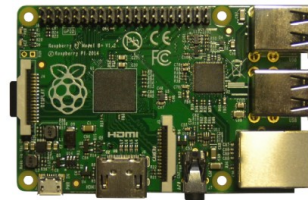
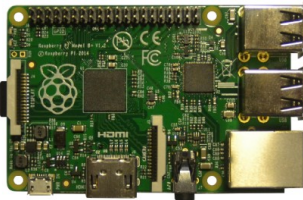
Problem: /dev/sdX?

- Mit **ssd Karte: df-h**

Dateisystem	Größe	Benutzt	Verf.	Verw%	Eingehängt auf
udev	3,9G	0	3,9G	0%	/dev
tmpfs	787M	9,8M	778M	2%	/run
/dev/sda2	57G	11G	44G	20%	/
tmpfs	3,9G	51M	3,8G	2%	/dev/shm
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	3,9G	0	3,9G	0%	/sys/fs/cgroup
/dev/sda3	161G	80G	74G	52%	/home
/dev/sda1	511M	4,6M	507M	1%	/boot/efi
tmpfs	787M	68K	787M	1%	/run/user/1000
/dev/sdc1	15G	13G	2,7G	83%	/media/misof/KLAUS1
/dev/sdb2	691G	406G	251G	62%	/media/misof/casper-rw
/dev/sdb3	686G	240G	446G	35%	/media/misof/KLAUS-DATEN-A
/dev/sdd1	15G	9,8G	5,1G	66%	/media/misof/DR-40

```
dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdd conv=fsync
```

SSH Verbindung -Terminal



Terminal-Befehle

Befehl	Optionen	Übersetzung	Auswirkung
ls		list	Dateien / Ordner anzeigen
mkdir NAME		make directory	Ordner erstellen
mv ALT NEU		move	verschieben ODER umbenennen
cd NAME		change directory	in Ordner rein wechseln
cd ..			aus Ordner raus wechseln
rm -R NAME	-R	remove	Ordner löschen R ... rekursiv ... löscht auch alle darin enthaltenen Dateien
nano datei			öffnet Editor

Datensichern

- **Sichern vom Pi → Laptop:**

```
rsync -avze ssh pi@192.168.1.XYZ:/home/pi/pclabor/servo.py ./.
```

- **Sichern vom Laptop → Pi:**

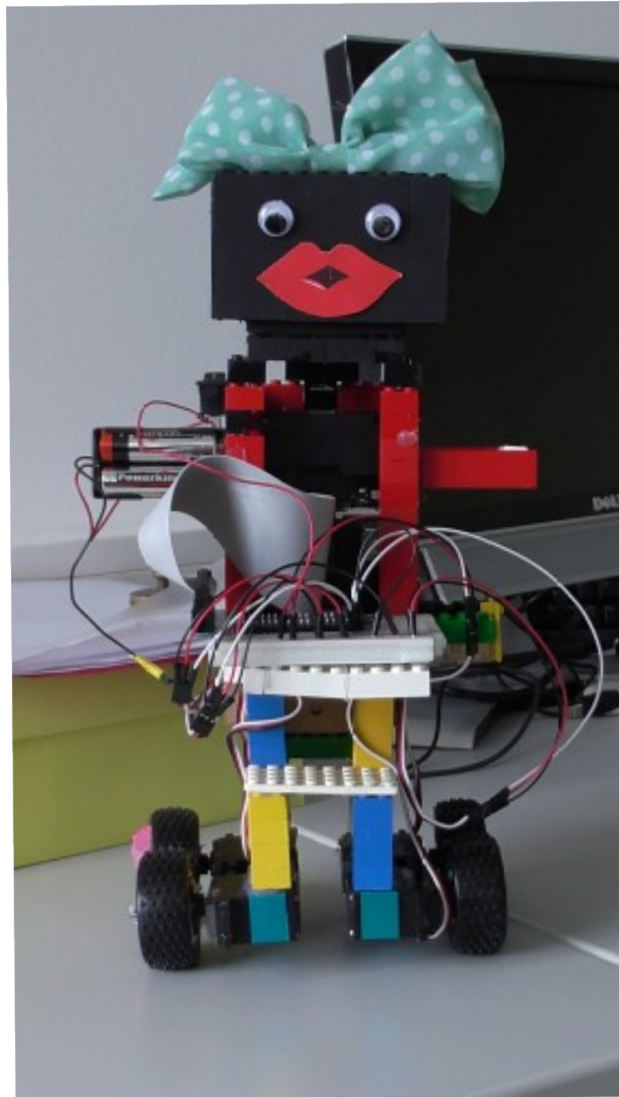
```
rsync -avze ssh ./servo.py pi@192.168.1.XYZ:/home/pi/pclabor/
```

Chassis, Motoren, Energie

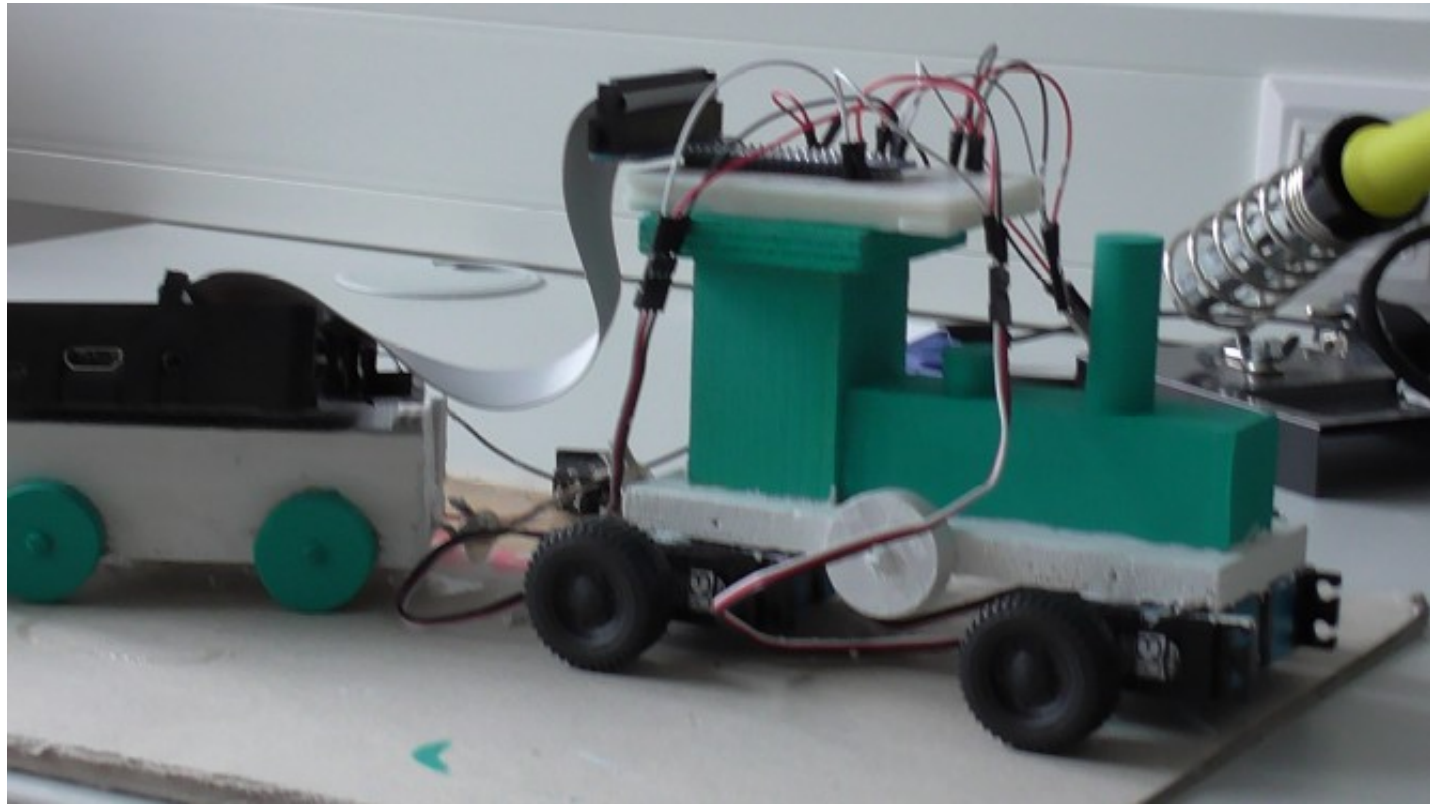


- Kreativität fördern → keine fertigen Chassis
- DC Motoren
- Servos aus dem RC Bereich
- Stromquelle(n)
- Breadboard
- Sensoren

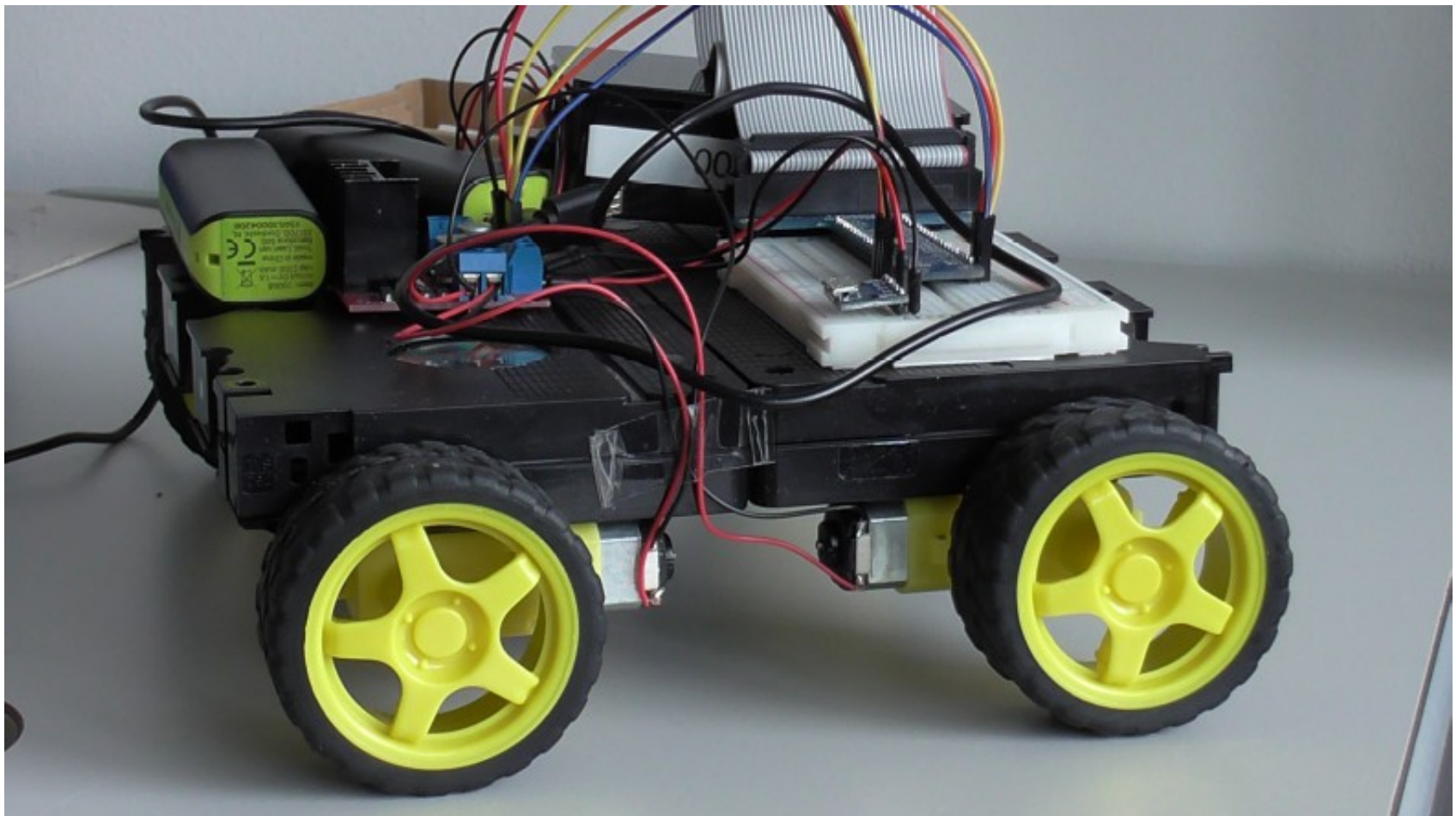
Diverse kreative Chassis



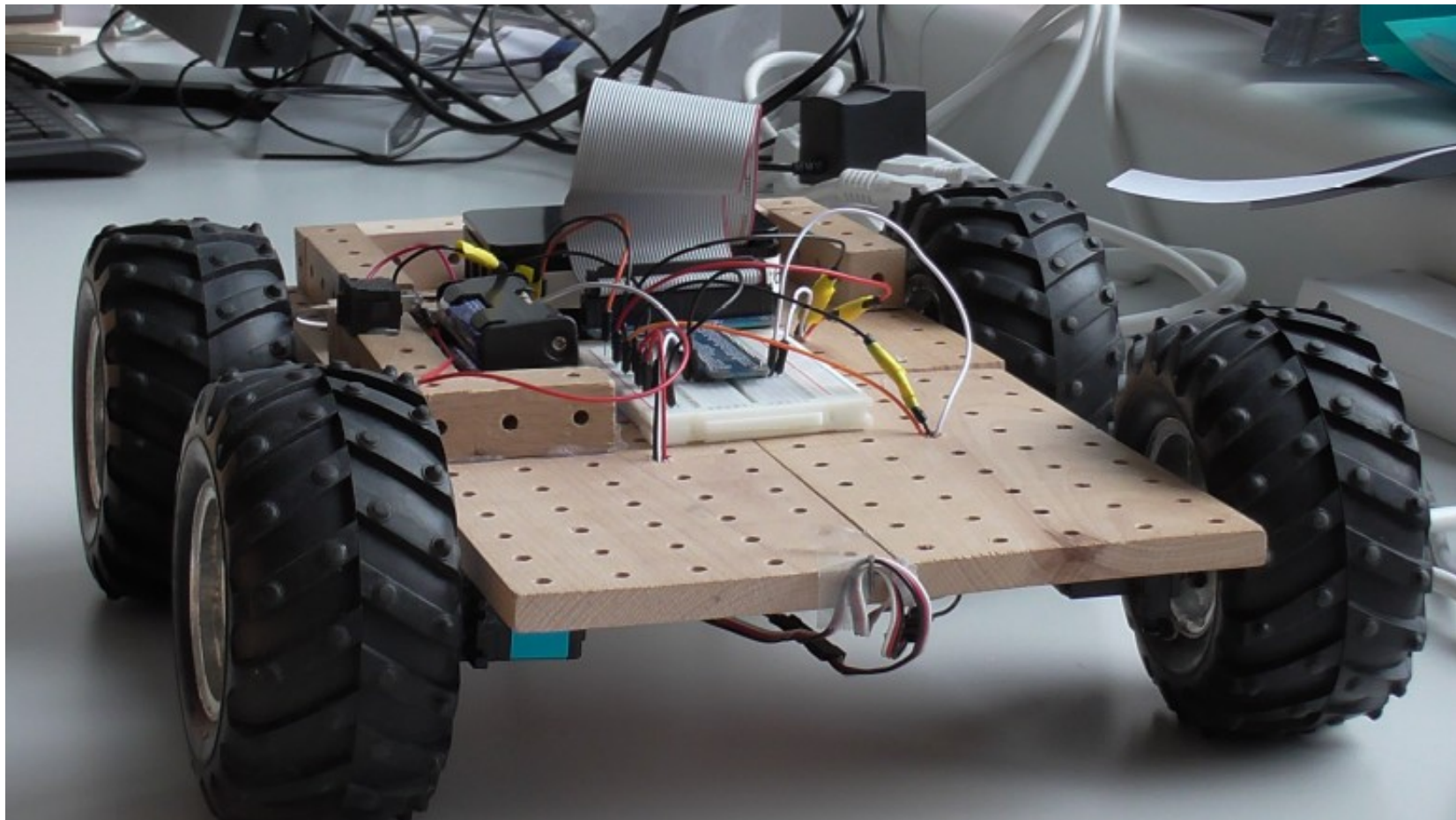
Diverse kreative Chassis



Diverse kreative Chassis



Diverse kreative Chassis



Diverse kreative Chassis



Diverse kreative Chassis

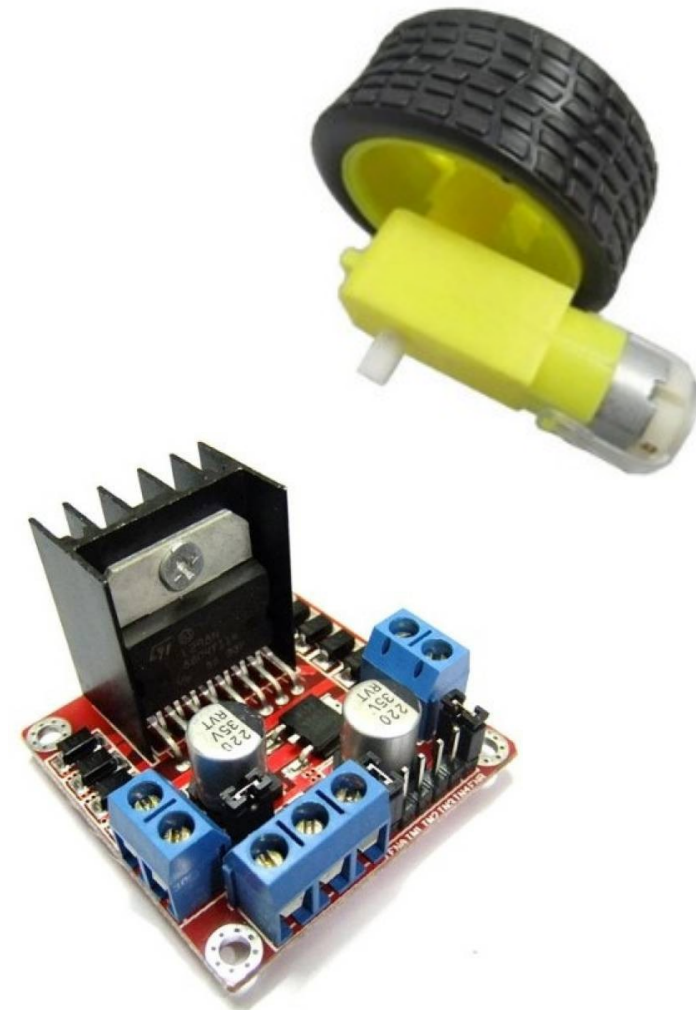


DC-Motoren, H-Brücke

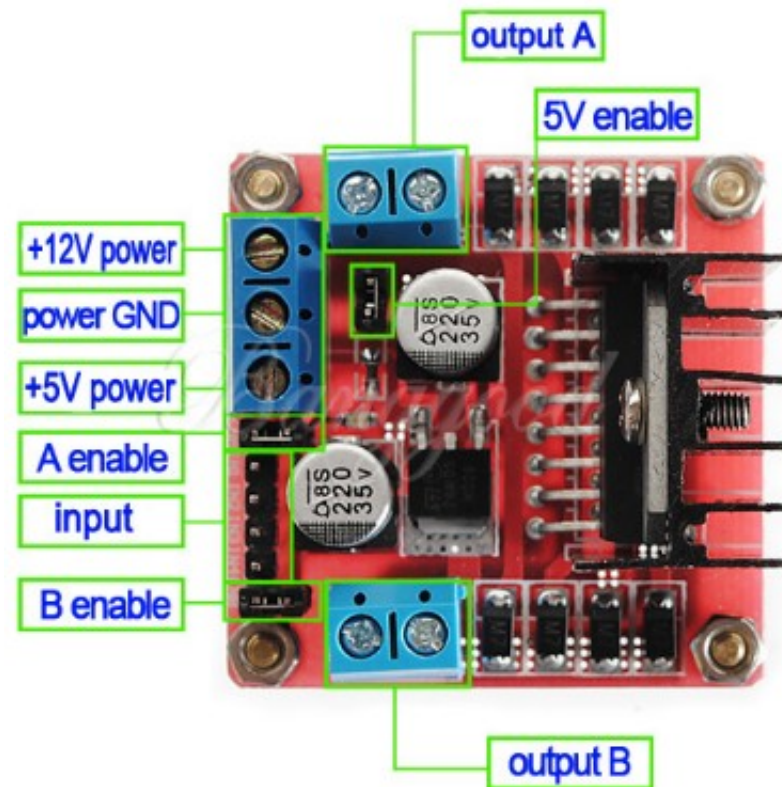
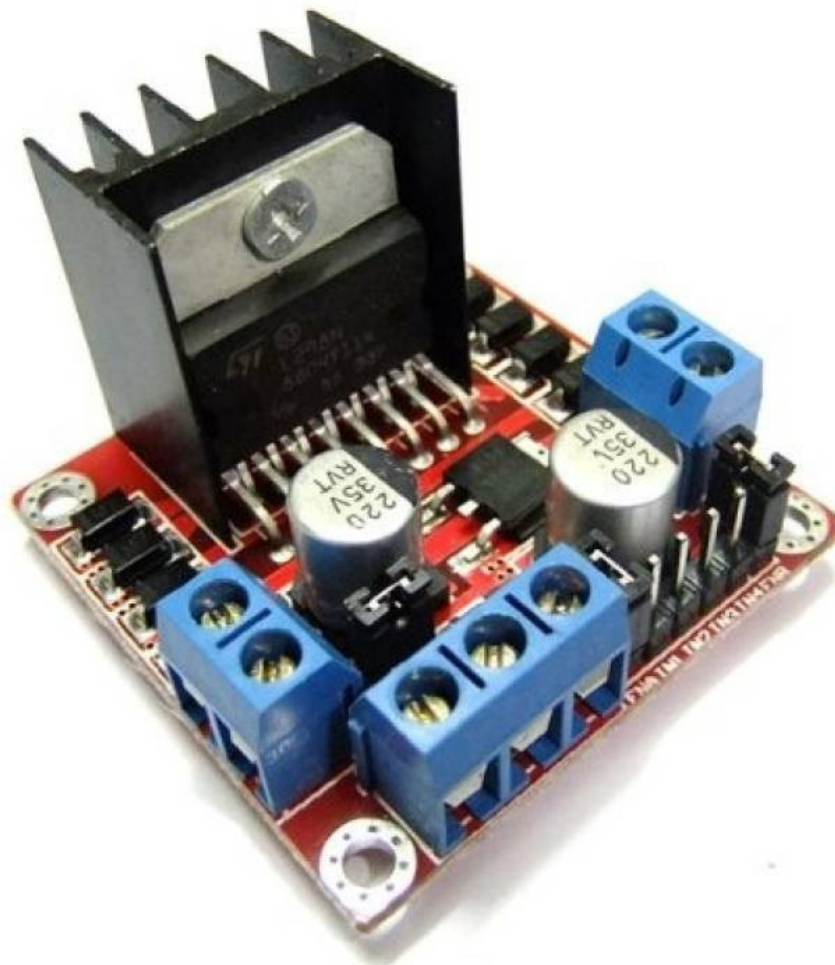
- DC-Motor mit Getriebe



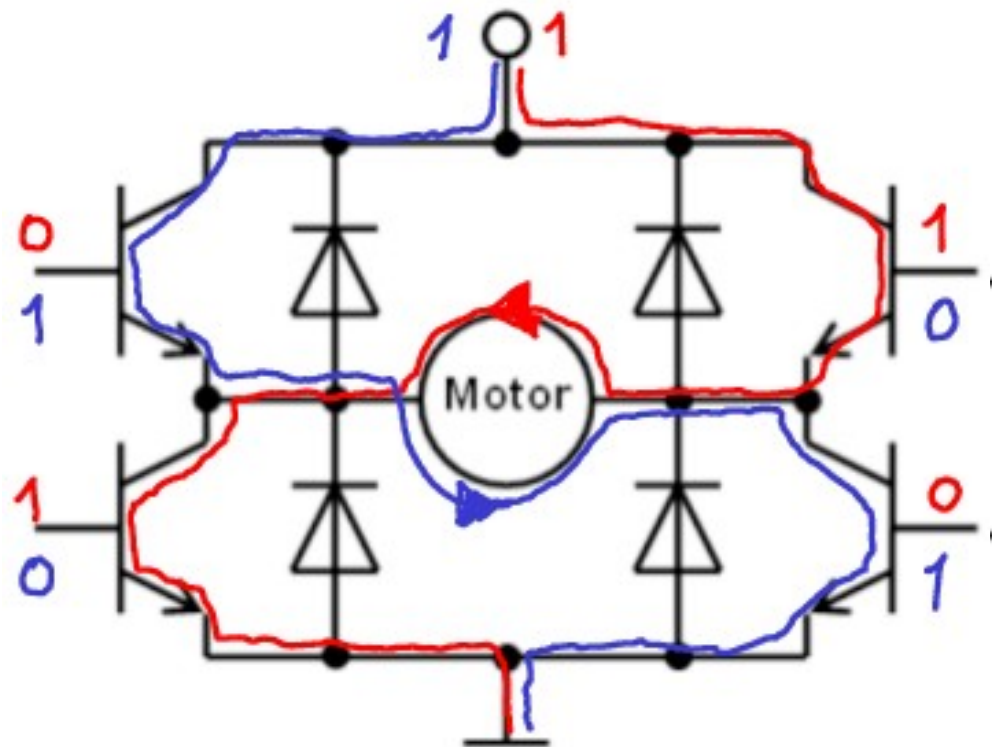
- H-Brücke (L298n)



H-Brücke L298n

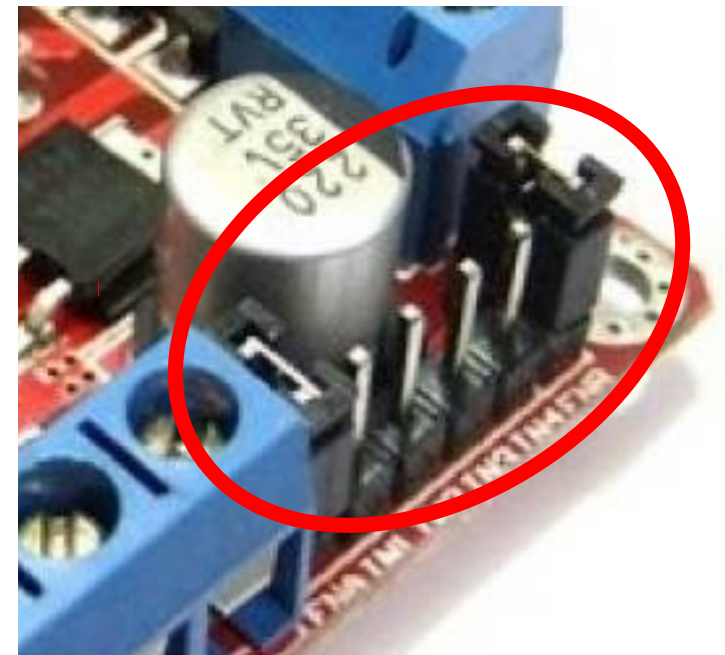


H-Brücke L298n



H-Brücke L298n

EnableA	MotorA1	MotorA2	Wirkung
0	0	0	Motor läuft aus
1	1	1	Motor stoppt
1	1	0	Motor dreht nach linkis
1	0	1	Motor dreht nach rechts
1	0	0	Motor stoppt



H-Brücke L298n

```
#!/usr/bin/python
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

MotorEnableA = 37
Motor1A = 35
Motor2A = 33

GPIO.setup(MotorEnableA,GPIO.OUT)
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor2A,GPIO.OUT)
```

Raspberry Pi2 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 1
26/01/2014 <http://www.element14.com>

H-Brücke L298n

```
#!/usr/bin/python
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

MotorEnableA    = 37
Motor1A         = 35
Motor2A         = 33

GPIO.setup(MotorEnableA,GPIO.OUT)
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor2A,GPIO.OUT)
```

```
def all_off() :
    GPIO.output(Motor1A,GPIO.LOW)
    GPIO.output(Motor2A,GPIO.LOW)
    GPIO.output(MotorEnableA,GPIO.LOW)

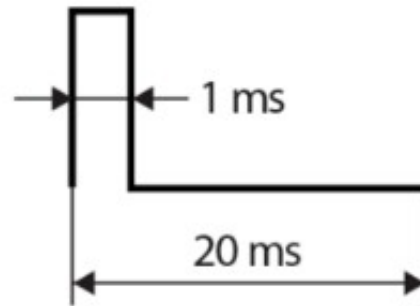
def forward():
    GPIO.output(MotorEnableA,GPIO.HIGH)
    GPIO.output(Motor1A,GPIO.LOW)
    GPIO.output(Motor2A,GPIO.HIGH)

def back():
    GPIO.output(MotorEnableA,GPIO.HIGH)
    GPIO.output(Motor1A,GPIO.HIGH)
    GPIO.output(Motor2A,GPIO.LOW)
```

Servo-Motoren, PWM

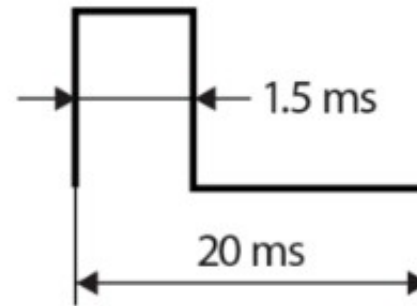


5%



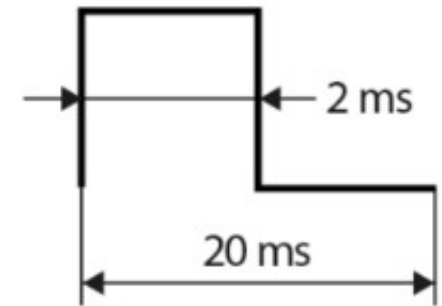
-90°

7,5%



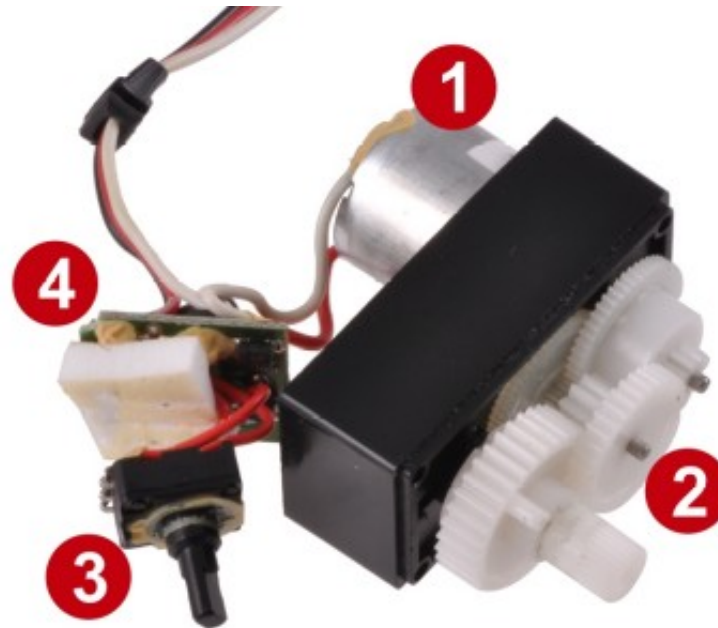
neutral

10%



90°

360° Servo Umbau



Python Code

```
#!/usr/bin/python
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

ServoA = 37
FrequA = 50
GPIO.setup(servoA,GPIO.OUT)
pwmA = GPIO.PWM(servoA,frequA)
Neutral = 7.5

def servoA(velocity):
    wert = neutral + velocity
    pwmA.start(wert)

#GPIO.cleanup
```



Energiequelle

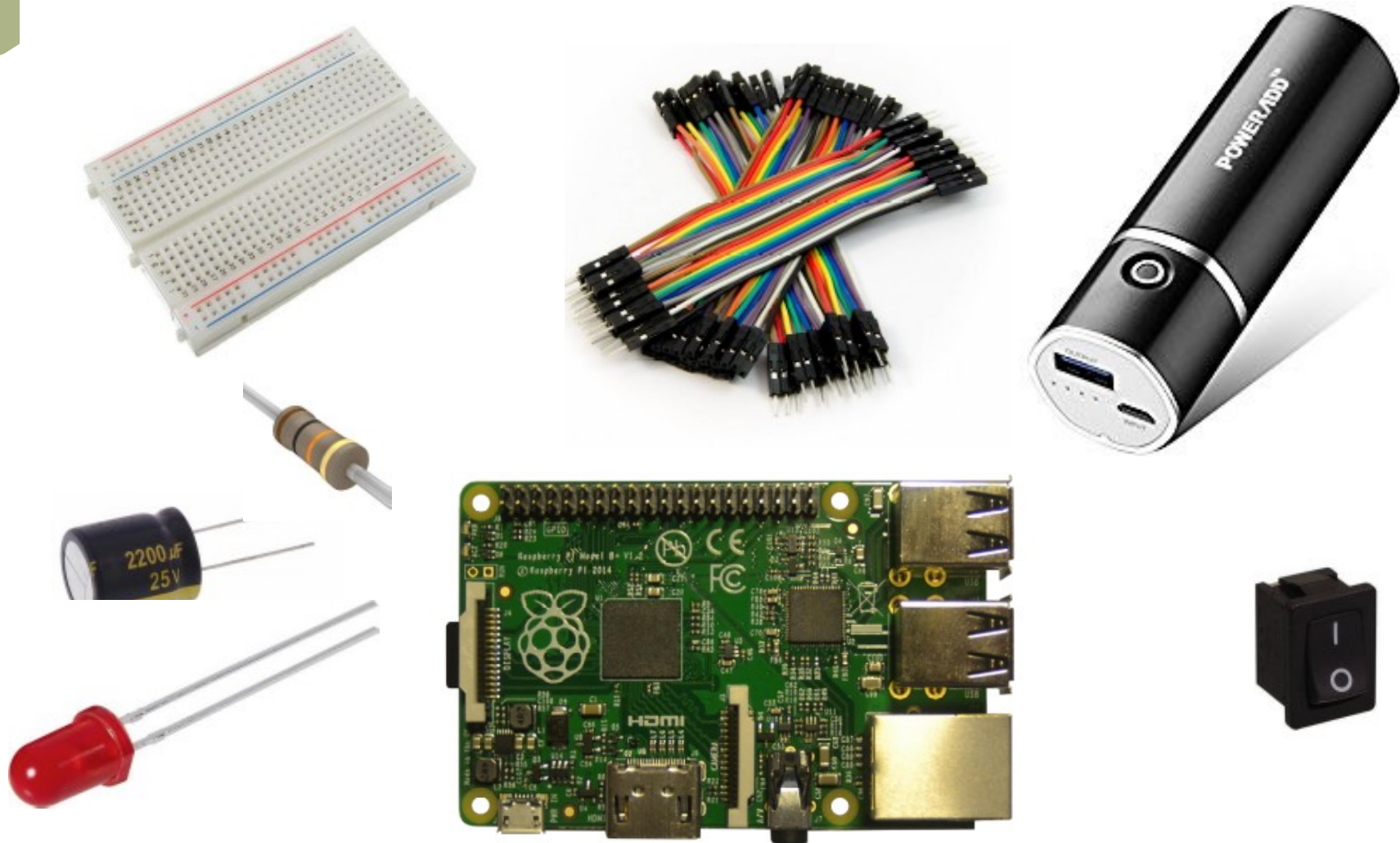
- Raspberry Pi
- DC-Motoren bzw. Servos
- Sensoren

- 5V DC
- Je nach Bestückung 2-4A

Energiequelle



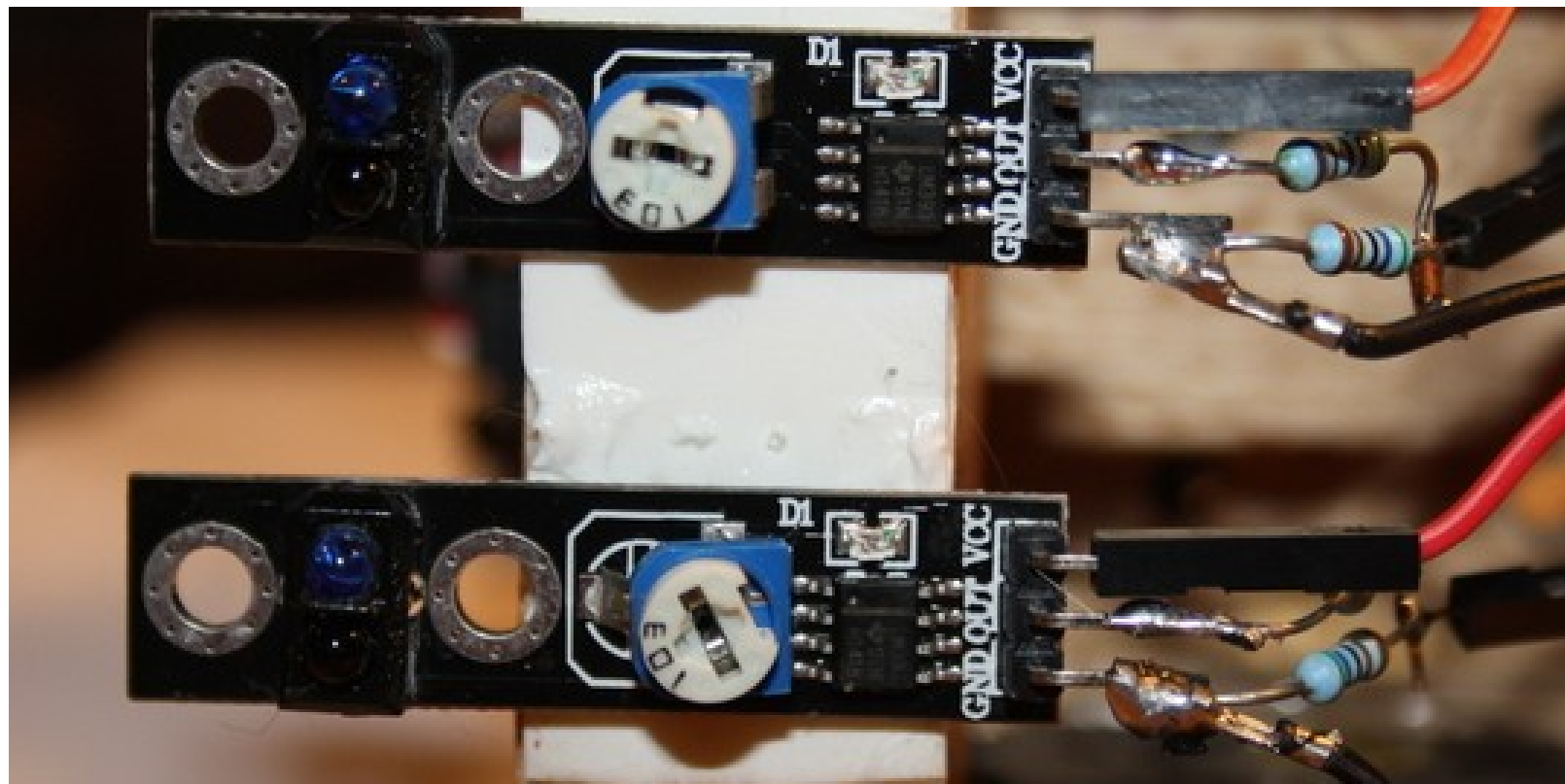
Energieverteilung



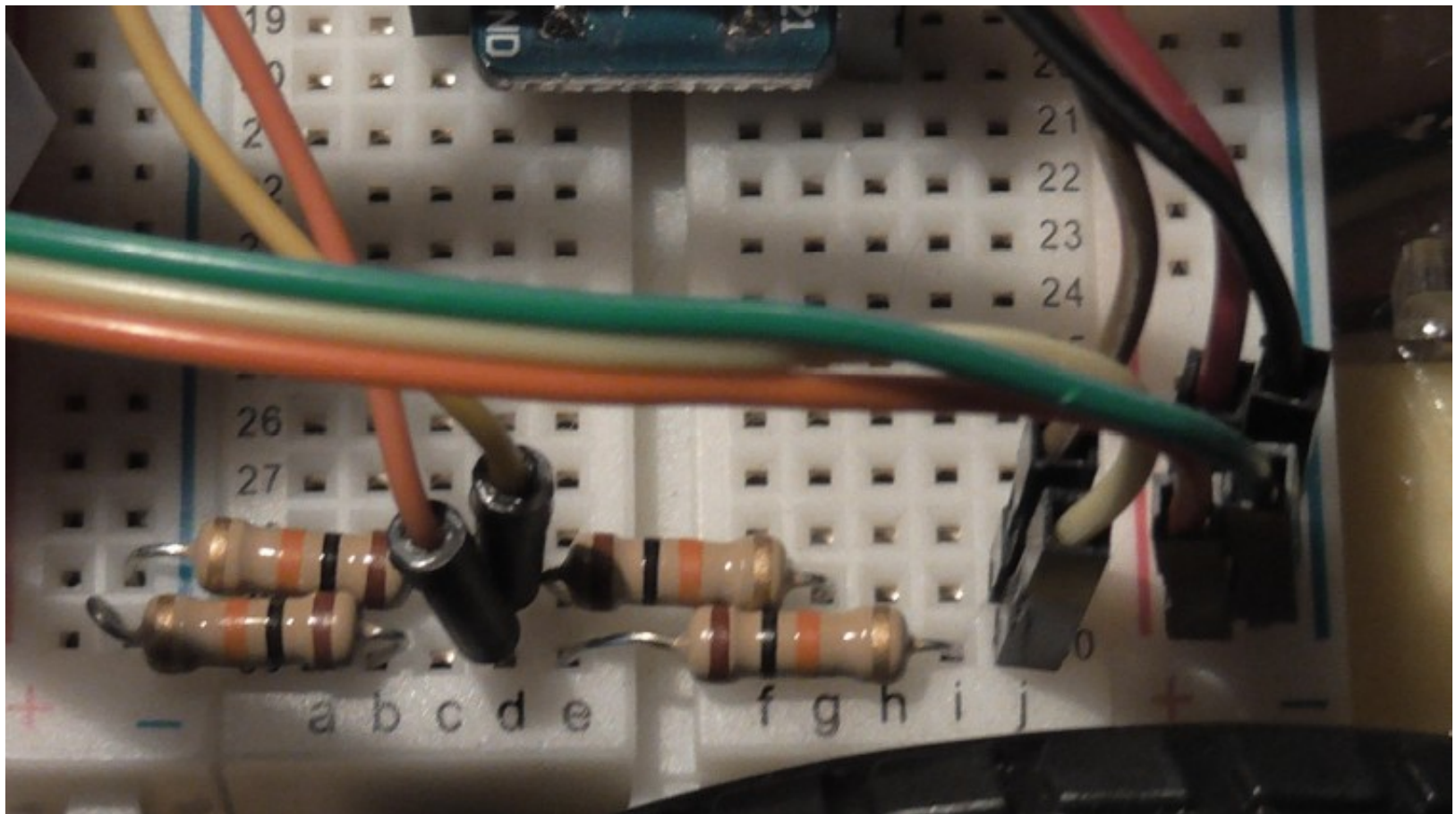
Sensoren

- IR Sensor
- Ultrasonic Sensor
- Temperatur Sensor
- Luftfeuchtigkeits Sensor
- Relais
- Funkmodul
- Webcam
-

IR Sensoren



IR Sensor, Spannungsteiler



IR Sensor

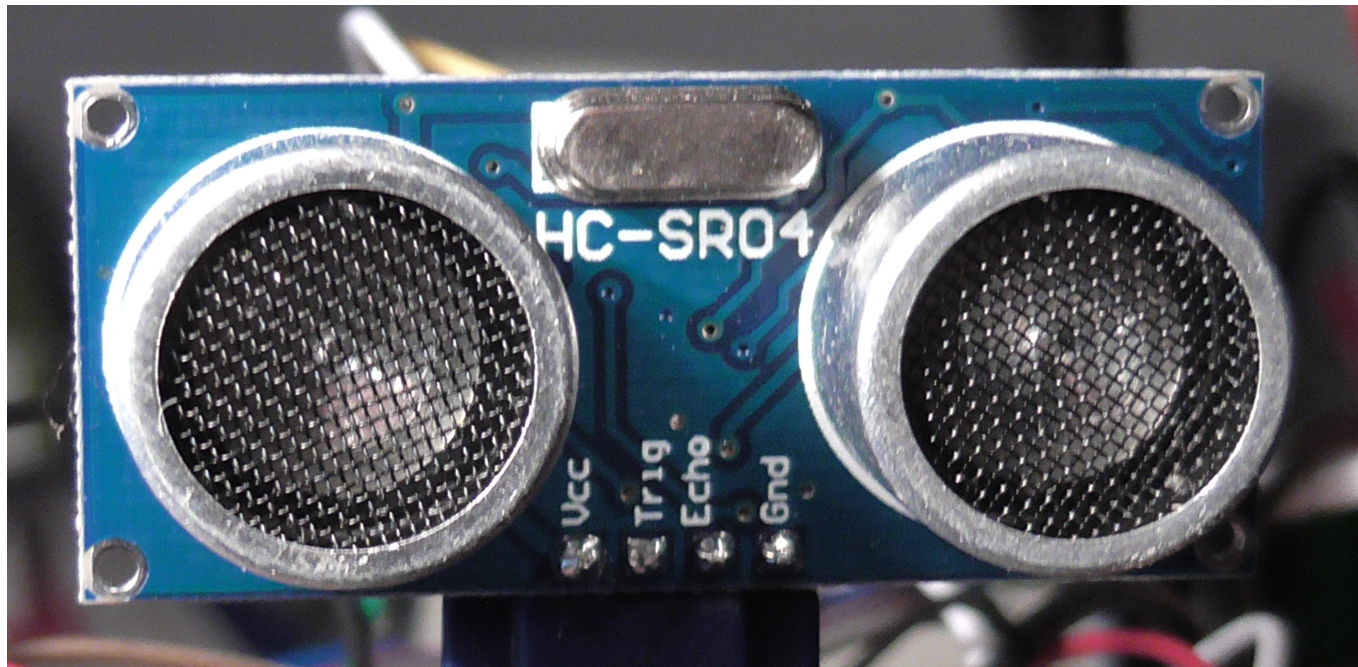
```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time
import sys

GPIO.setmode(GPIO.BOARD)
lineright = 29
lineleft = 31

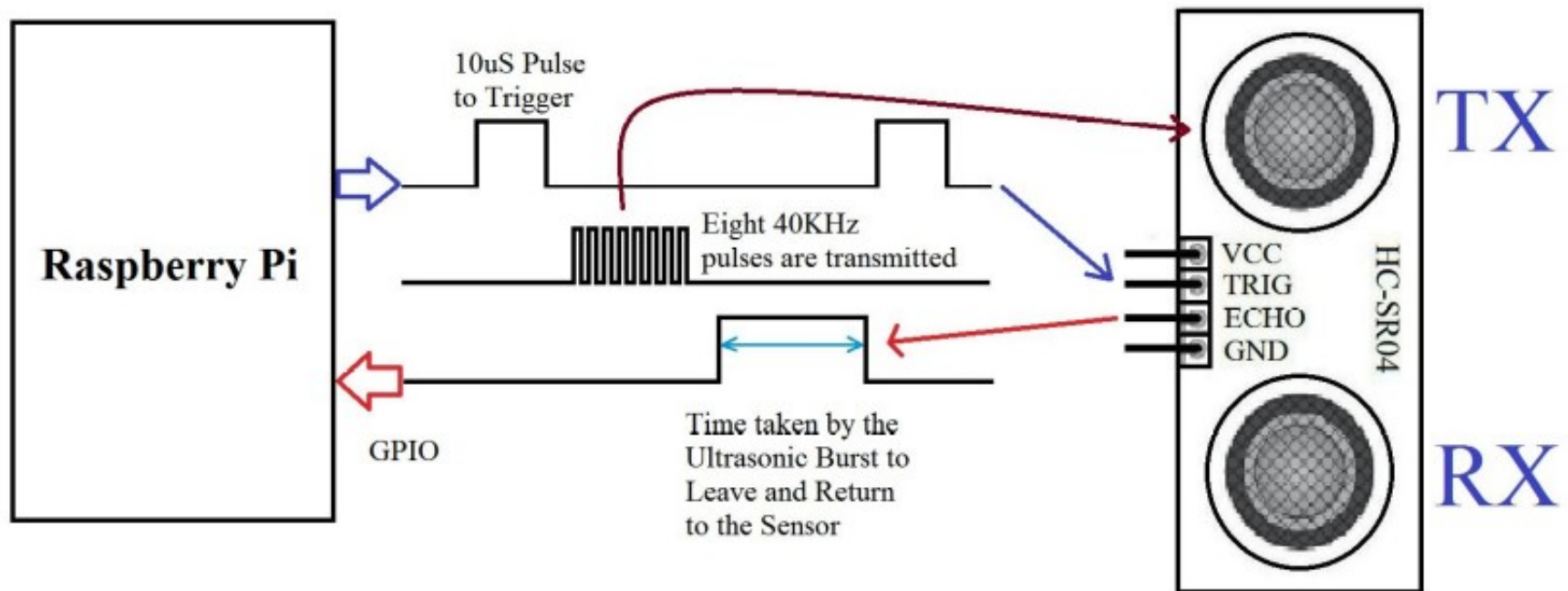
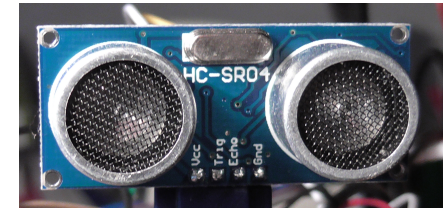
GPIO.setup(lineright,GPIO.IN)
GPIO.setup(lineleft,GPIO.IN)
```

```
def erkennung():
    if GPIO.input(lineright) == True and ↓
        GPIO.input(lineleft) == False:
        ausgabe = "TRUE_RIGHT"
    elif GPIO.input(lineright) == False and ↓
        GPIO.input(lineleft) == True:
        ausgabe = "TRUE_LEFT"
    elif GPIO.input(lineright) == True and ↓
        GPIO.input(lineleft) == True:
        ausgabe = "BOTH_TRUE"
    elif GPIO.input(lineright) == False and ↓
        GPIO.input(lineleft) == False:
        ausgabe = "BOTH_FALSE"
    else:
        ausgabe = "NOINFO"
    return ausgabe
```


Ultrasonic Sensor



Ultrasonic Sensor



Ultrasonic Sensor

```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time
import os

GPIO.setmode(GPIO.BOARD)
TRIG = 29
ECHO = 31
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
GPIO.output(TRIG,0)

# Konstante
ruhezeit = 0.1
siglength = 0.000001
```

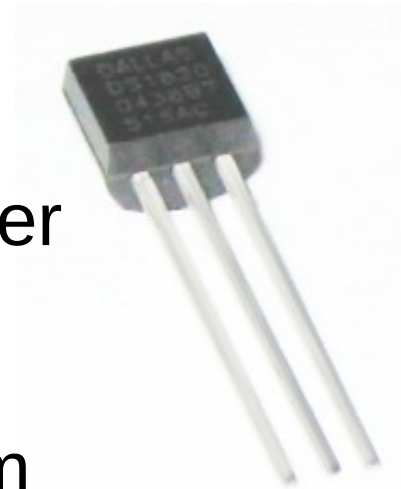
```
def distance():
    abstand = 0
    zeit = time.time()
    start = time.time()

    GPIO.output(TRIG,1)
    time.sleep(siglength)
    GPIO.output(TRIG,0)

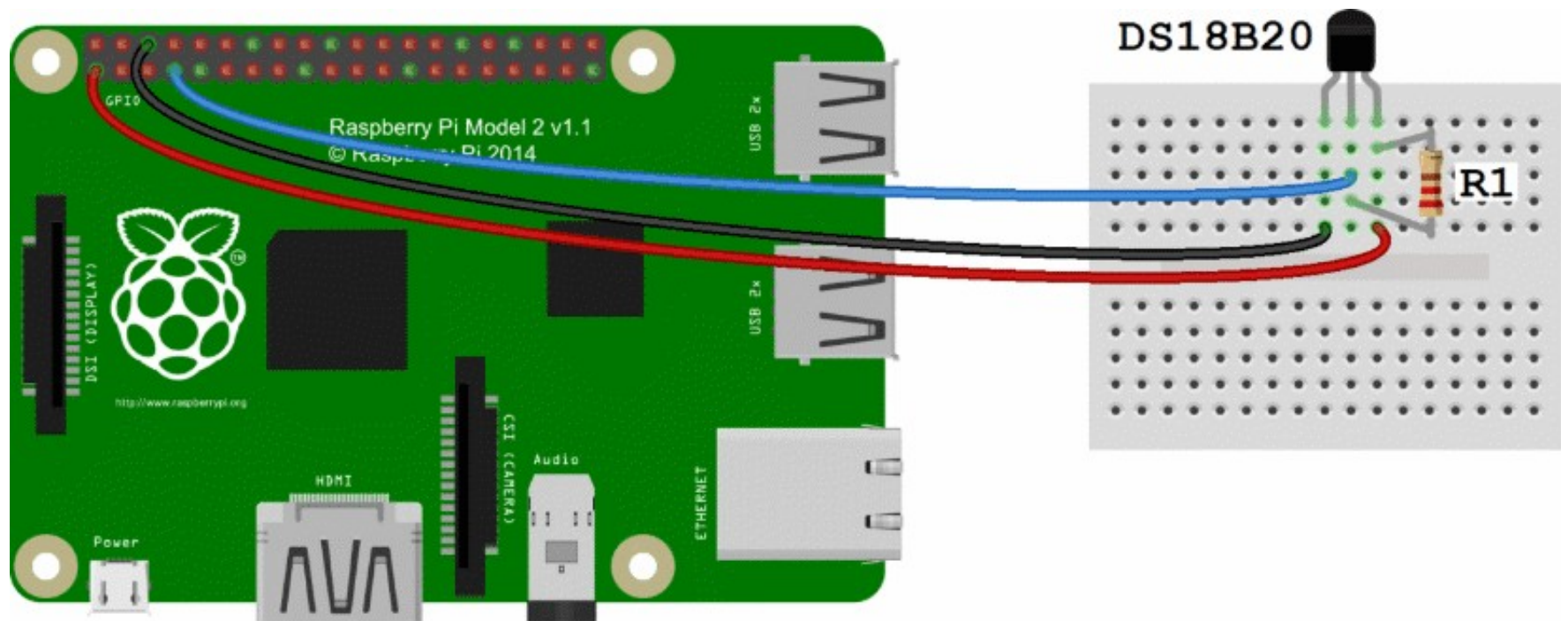
    while (GPIO.input(ECHO) == 0):
        start = time.time()
    while (GPIO.input(ECHO) == 1):
        stop = time.time()
    # Abstandsberechnung
    abstand = (stop – start) * 17000
    time.sleep(ruhezeit)
    # Abstand in cm zurueckliefern
    return int(abstand)
```

Temperatur Sensor

- Digitales Thermometer
- 9-bit Auflösung
- -55°C to $+125^{\circ}\text{C}$
- $\pm 0.5^{\circ}\text{C}$ Accuracy from -10°C to $+85^{\circ}\text{C}$



Temperatur Sensor



fritzing

DS18S20 Kernelmodul

- **1-Wire Bus Aktivierung (alt):**

```
sudo modprobe wire  
sudo modprobe w1-gpio  
sudo modprobe w1-therm
```

- **Ab Kernel 3 (neu):**

- sudo nano /boot/config.txt:

```
dtoverlay=w1-gpio  
gpiopin=4
```


DS18S20 Kernelmodul

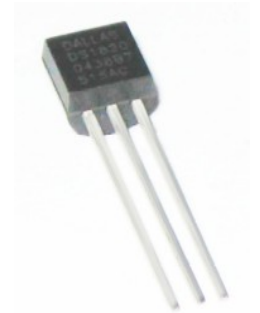
1
cat /sys/bus/w1/devices/28-000005d2e508/w1_slave:

```
33 00 4b 46 ff ff 02 10 f4 : crc=f4 YES
```

```
33 00 4b 46 ff ff 02 10 f4 t=25625
```

DS18S20 Kernelmodul

1



```
#!/usr/bin/python
import os, sys, time

def aktuelleTemperatur():
    file = open('/sys/bus/w1/devices/28-000005d2e508/w1_slave')
    fileinhalt = file.read()
    file.close()

    wert = fileinhalt.split("\n")[1].split(" ")[9]
    temperatur = float(wert[2:]) / 1000

    rueckgabewert = '%6.2f' % temperatur
    return(rueckgabewert)
```

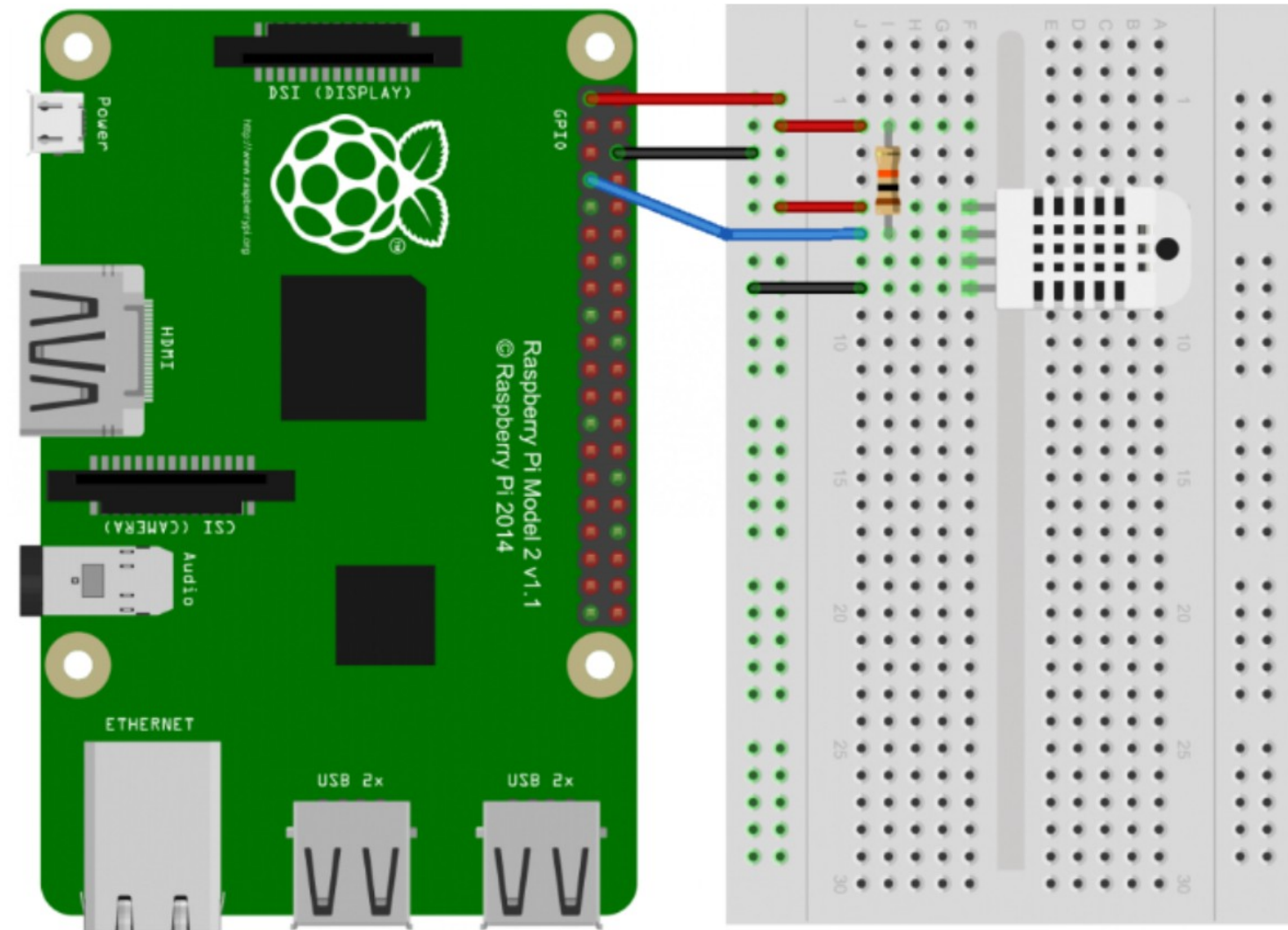
Luftfeuchtigkeit/Temperatur Sensor

DHT22 oder DHT11



- **DHT11:**
20-80% (5% Genauigkeit) und
im Temperaturbereich von ca.
0-50°C (2°C Genauigkeit)
- **DHT22:**
0-100% (2-5% Genauigkeit)
und bei der Temperatur im
Bereich von -40 bis 125°C
(0,5°C Genauigkeit)

Luftfeuchtigkeit/Temperatur Sensor



Luftfeuchtigkeit/Temperatur Sensor

- Installation der Software:

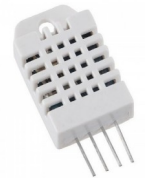


```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

- Werte Auslesen:

```
sudo ./AdafruitDHT.py 22 4
```

Luftfeuchtigkeit/Temperatur Sensor



```
#!/usr/bin/python
import Adafruit_DHT

sensor = Adafruit_DHT.DHT22
gpio = 4

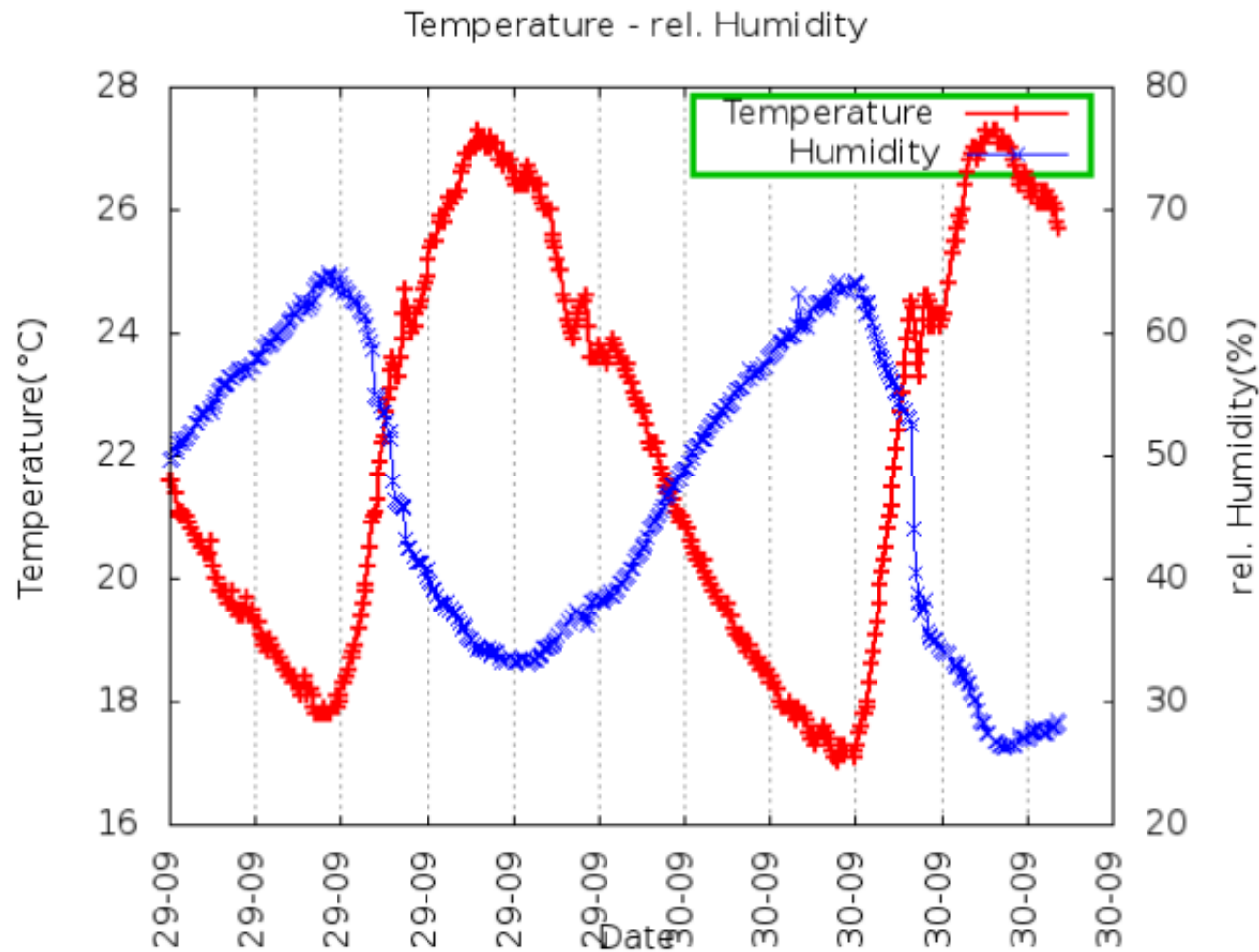
humidity, temperature = Adafruit_DHT.read_retry(sensor, gpio)

print 'Temperatur: {0:0.1f}°C Luftfeuchtigkeit: {1:0.1f}
%'.format(temperature,humidity)
```


Wiring PIN Belegung

BCM	WiringPi	Name	Physical	Name	WiringPi	BCM
		3.3v	1	2	5v	
2	8	SDA.1	3	4	5V	
3	9	SCL.1	5	6	0v	
4	7	1-Wire	7	8	TxD	15 14
		0v	9	10	RxD	16 15
17	0	GPIO.0	11	12	GPIO.1	1 18
27	2	GPIO.2	13	14	0v	
22	3	GPIO.3	15	16	GPIO.4	4 23
		3.3v	17	18	GPIO.5	5 24
10	12	MOSI	19	20	0v	
9	13	MISO	21	22	GPIO.6	6 25
11	14	SCLK	23	24	CE0	10 8
		0v	25	26	CE1	11 7
0	30	SDA.0	27	28	SCL.0	31 1
5	21	GPIO.21	29	30	0v	
6	22	GPIO.22	31	32	GPIO.26	26 12
13	23	GPIO.23	33	34	0v	
19	24	GPIO.24	35	36	GPIO.27	27 16
26	25	GPIO.25	37	38	GPIO.28	28 20
		0v	39	40	GPIO.29	29 21
BCM	WiringPi	Name	Physical	Name	WiringPi	BCM

Messungen im Klimagarten



Workshops Raspberry Pi

- Ab der 4. Klasse
- 3-4 stündige Workshops möglich
- Vorbereitete Chassis
- Einen Wlan-Router
- Gruppen zu je zwei bis drei Schüler
- Ein Laptop pro Gruppe

Das Team



Martin Gruber

Rene Schwarzinger

Benedikt Gottsbachner

Klaus Misof

email: klausmisof@gmx.at