
10 PACK
Perfekt für Filme, Fotos und Datensicherung
DVD+R
4.7GB
120min
16fache
EDEKA
GUT & GÜNSTIG
SAMSUNG
8 x AA
extrem leistungsstark
POWER CELL

Sensoren & Aktoren am Raspberry Pi

Die Stiftleiste für Steuerungsaufgaben nutzen

Johannes Roith

14.11.2021

Über mich

- Hardwarenaher Softwareentwickler
- GNU/Linux Nutzer seit 9 Jahren
- Youtube Kanal: Johannes 4GNU_Linux
- Webseite & Kontakt: `www.gnu-linux.rocks`

Über den Foliensatz

- Folien lizenziert unter
  - Weitergabe und Remix erlaubt
  - Namensnennung notwendig
  - Remix muss unter selben Lizenz weitergegeben werden
- Folien zum Nachschlagen gedacht
- Programmierbeispiele in Python und manchmal auch Bash

Der Raspberry Pi

Quelle: `https://commons.wikimedia.org/wiki/File:Raspberry-Pi-2-Bare-BR.jpg`

Lizenz: 🄯

Setup Raspberry Pi

Image herunterladen

- Image hier herunterladen
- Zip Datei entpacken

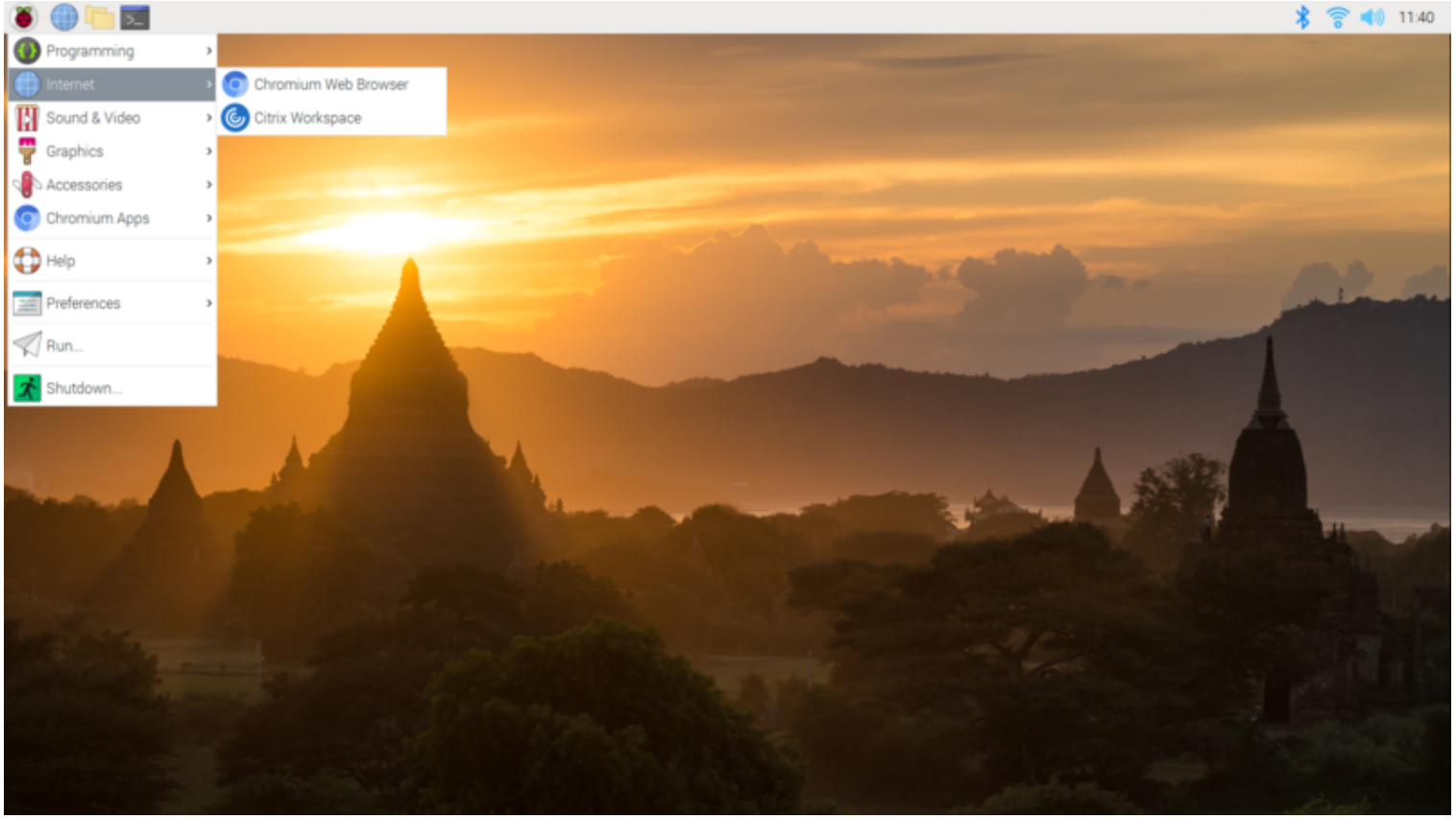
Image auf Karte flashen

- `sudo dd if=<Pfad zu img> of=/dev/mmcblk<x> status=progress`
- Alternativ: Etcher, Fedora Media Writer, rpi-imager, ...

Alternative: Headless Setup

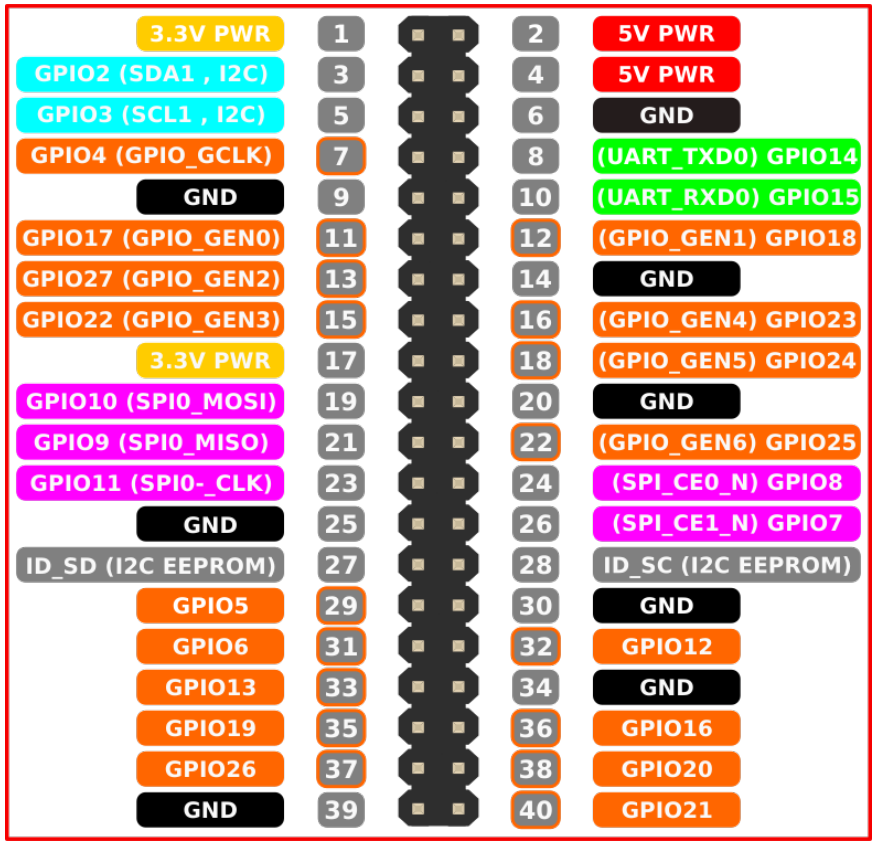
- Setup ohne Maus, Bildschirm und Tastatur
- Link zu den Infos

Setup Raspberry Pi



Quelle: `https://commons.wikimedia.org/wiki/File:Rapberry_Pi_OS_Screenshot.png`

Die Stiftleiste



Quelle: `https://freesvg.org/gpiopinsv3`

Lizenz: 🄯

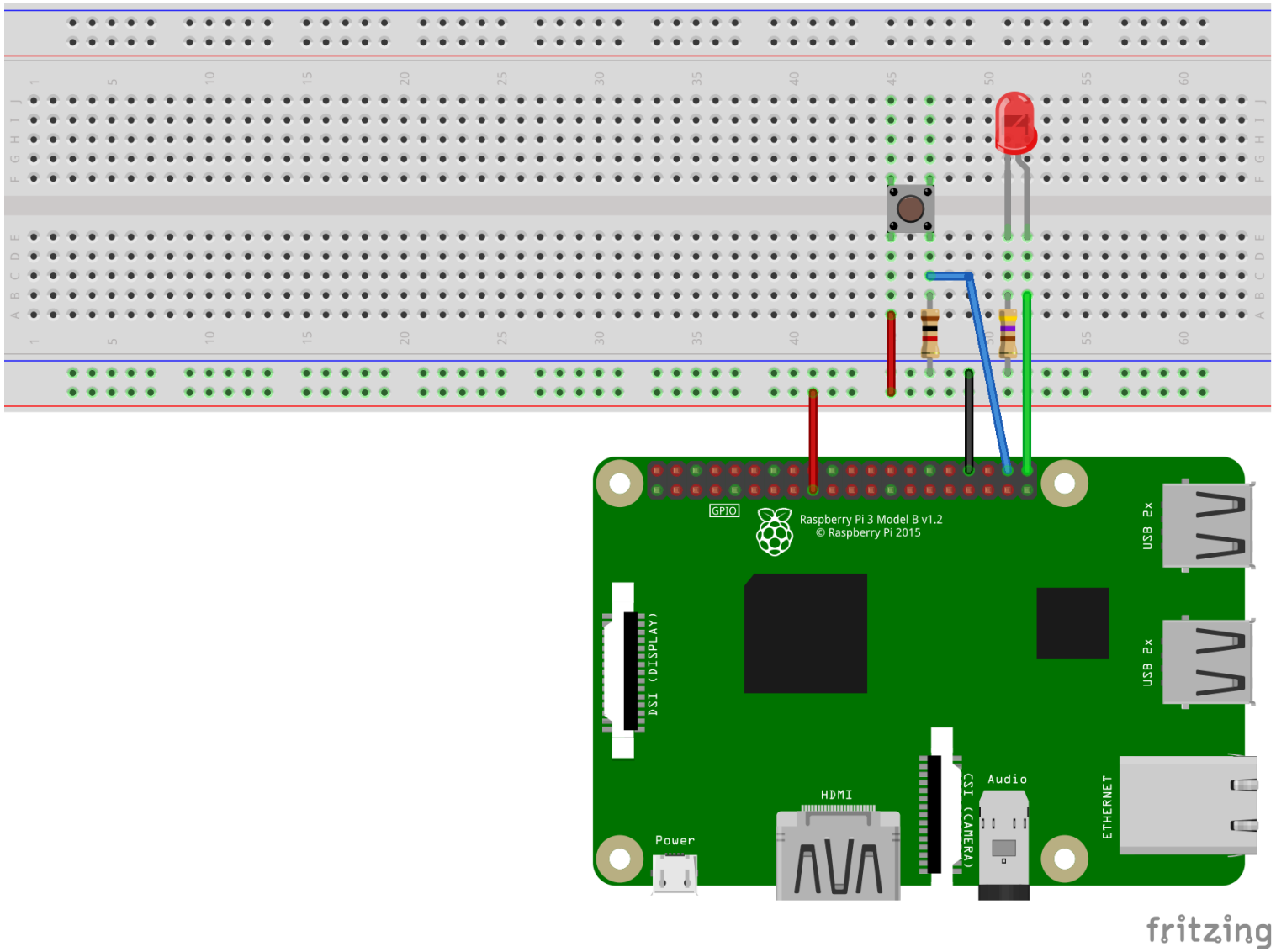
Das schauen wir uns an

1. General Purpose Input/Output Pins
2. I2C Bus
3. SPI Bus
4. Serielle Schnittstelle
5. PWM

General Purpose Input/Output Pins

- Digitale Ein- und Ausgänge
- Spannungspegel beim Raspberry Pi 3,3V
- Achtung: Maximaler Strom begrenzt: Alle GPIOs maximal 50mA

GPIO Schaltung



Ansteuerung mit Bash

Über sysfs

GPIO als Ausgang

```
echo 21 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio21/direction
echo 1 > /sys/class/gpio/gpio21/value
echo 0 > /sys/class/gpio/gpio21/value
```

GPIO als Eingang

```
echo 20 > /sys/class/gpio/export
echo in > /sys/class/gpio/gpio20/direction
cat /sys/class/gpio/gpio20/value
```

Ansteuerung mit Bash

Über gpiochip

GPIO Erkennen

```
gpiodetect gpiochip0
```

GPIO als Ausgang

```
gpioset gpiochip0 21=1
```

GPIO als Eingang

```
gpioget gpiochip0 20
```

Ansteuerung mit Python

Installation

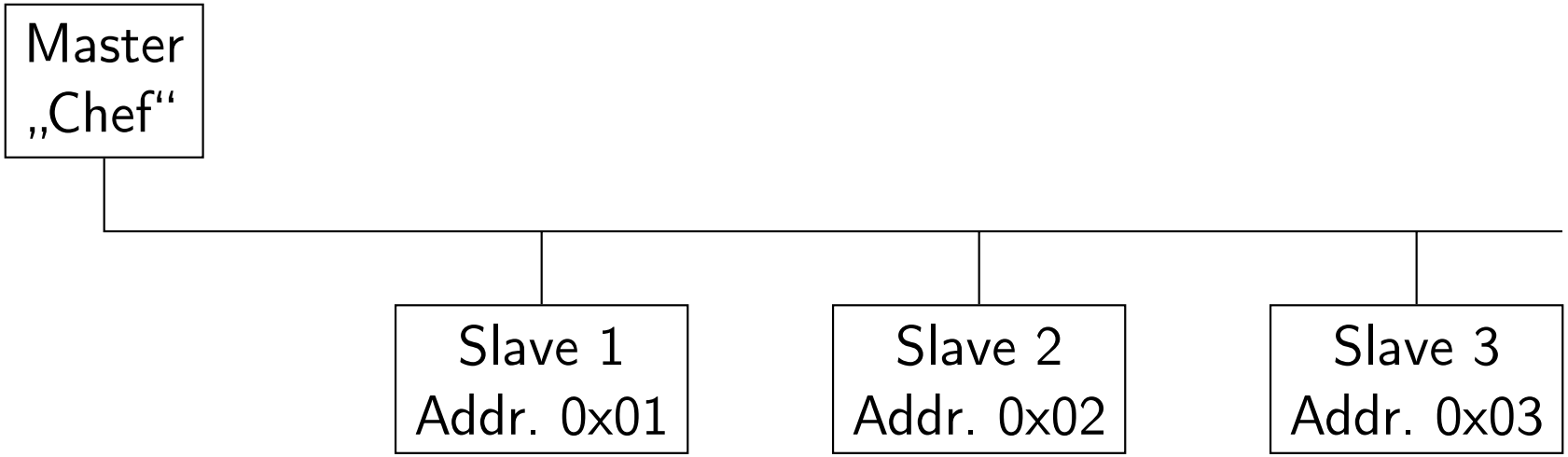
```
sudo pip install gpiodev
```

```python
from gpiodev import GPIOChip
from time import sleep

gpiochip = GPIOChip("/dev/gpiochip0")
# LED und Button Init
led = gpiochip.get_handle((21,))
button = gpiochip.get_handle((20,), mode="in")
# Lese Button
button.get_values()
# Toggle LED
for i in range(10):
        led.set_values((1,))
        sleep(0.5)
        led.set_values((0,))
        sleep(0.5)
```

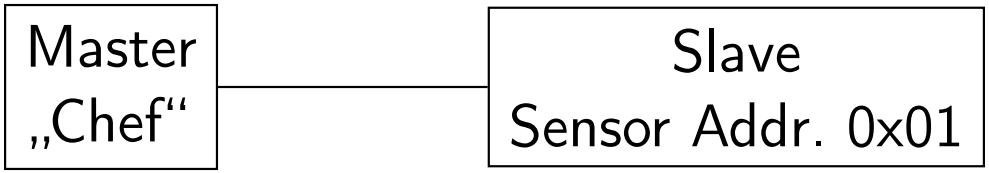
Weitere nützliche Bauteile

- Optokoppler, Relais
- Displays (z.B. HD44780, Segmentanzeigen)
- Ultraschall-Distanzsensor LINKER SEN-US01
- Buzzer
- Taster, DIP-Schalter
- ...

I2C Bus - Was ist ein Bus

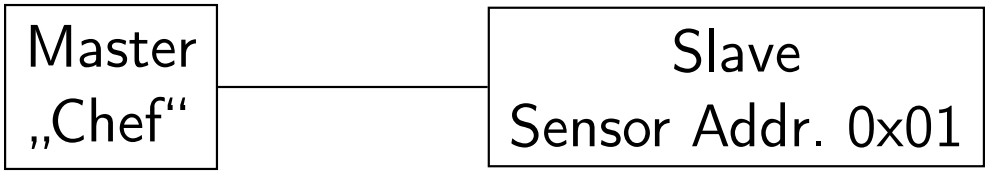


I2C Komplexere Geräte



Interne Adresse:	Bedeutung
0x0	Temperaturwert
0x1	Luftdruckwert

I2C Komplexere Geräte



Interne Adresse:	Bedeutung
0x0	Temperaturwert
0x1	Luftdruckwert

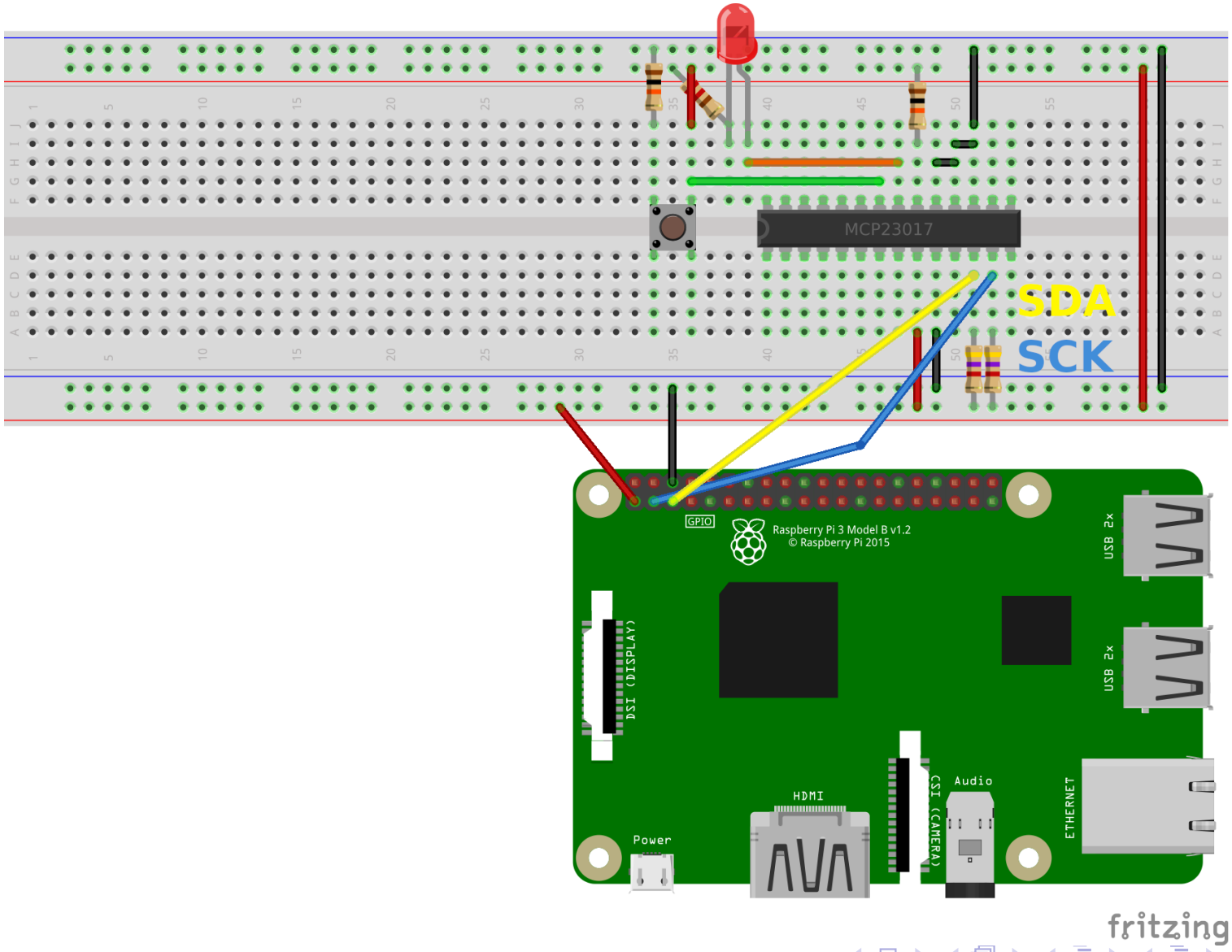
- Erster Zugriff: Master schreibt gewünschte Adresse (0 für Temperatur, 1 für Luftdruck)
- Zweiter Zugriff: Master liest von Slave, Slave liefert Wert entsprechend der gesetzten Adresse

I2C Bus

- Einfacher Zweidrahtbus
- Datenleitung: *SDA*
- Taktleitung: *SCK*
- Frequenzen: 100kbit/s, 400kbit/s, 1Mbit/s
- Pull-Up Widerstand bei Leitungen notwendig

I2C Bus Schaltung

mit MCP23017 GPIO Expander



Ansteuerung mit Bash

I2C Geräte finden

```
i2cdetect -y 0
```

Von I2C Gerät lesen

Lesen von Register 9 des Geräts mit Adresse 0x20

```
i2cget -y 0 0x20 9
```

Register in I2C Gerät schreiben

Den Wert 0x01 in Register 10 des I2C Geräts mit Adresse 0x20 schreiben

```
i2cset -y 0 0x20 0xa 0x1
```

Ansteuerung mit Python

Installation

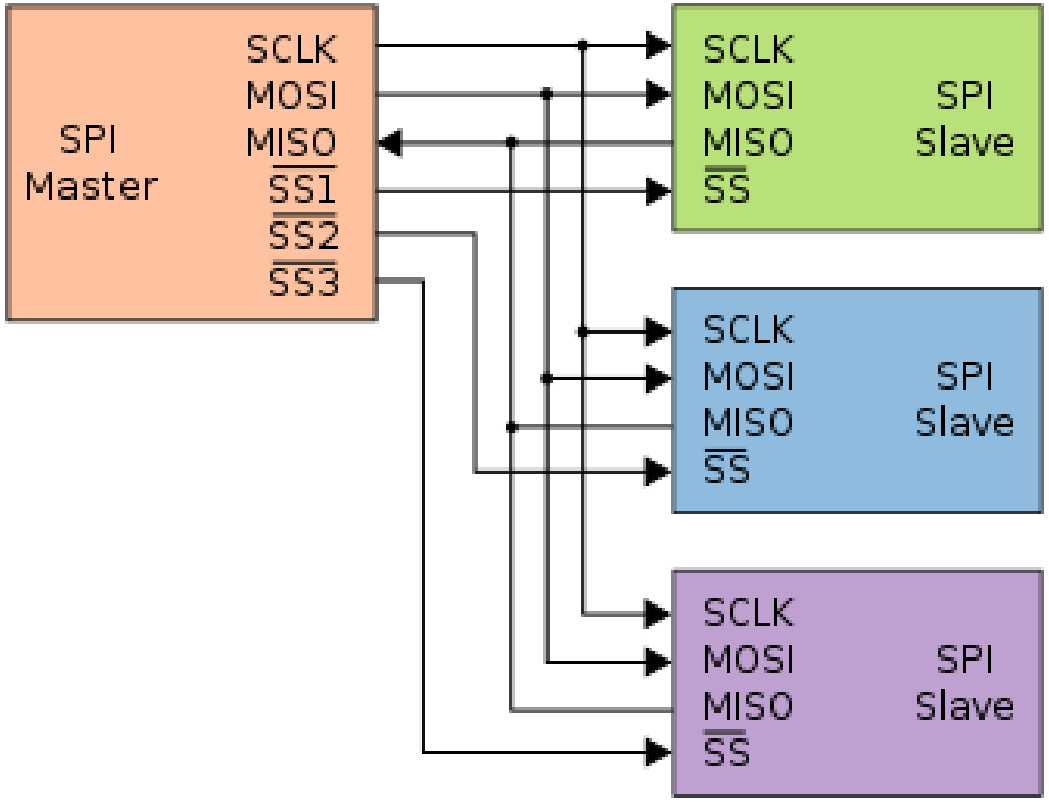
```
sudo pip install smbus
```

```
import smbus
mcp23017 = smbus.SMBus(0)

# GPIOA konfigurieren
mcp23017.write_byte_data(0x20, 0x0, 0xfe)

# Button einlesen
mcp23017.read_byte_data(0x20, 0x9);

# GPIO setzen
mcp23017.write_byte_data(0x20, 0xa, 0x1)
```

Weitere nützliche Bauteile

- ADCs / DACs
- BMP280 Temperatur Sensor
- LCD Controller
- GPIO Expander
- Sensoren (Beschleunigung, Luftqualität, ...)
- ...

SPI Bus



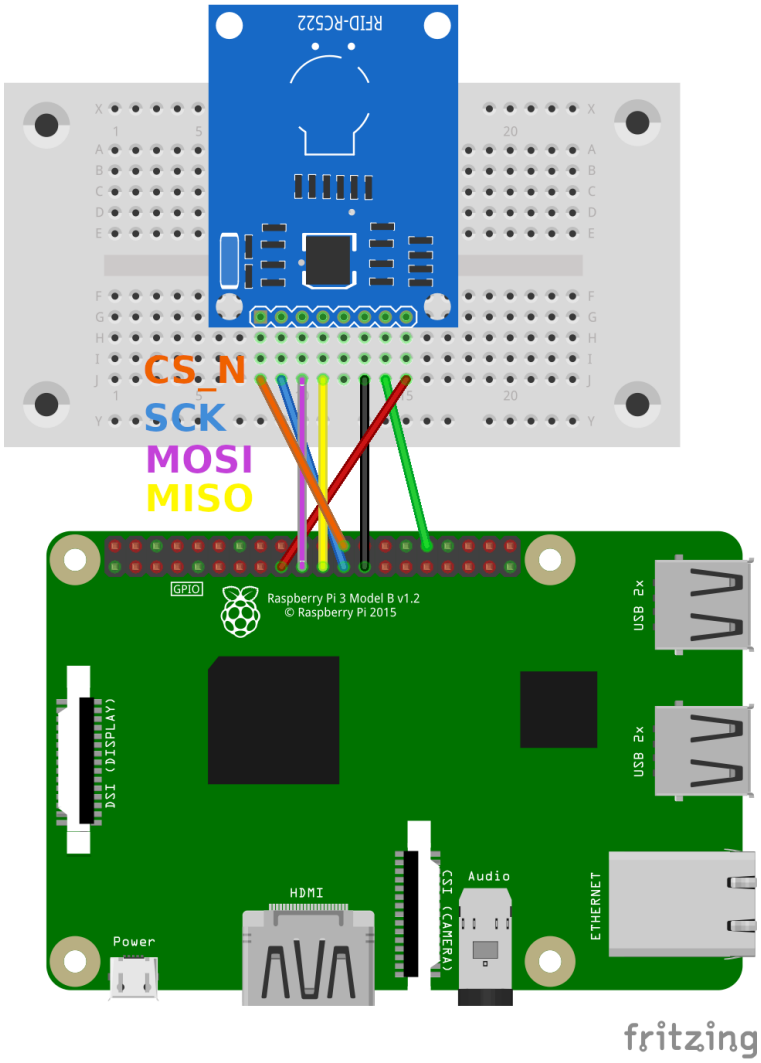
Quelle: `https://commons.wikimedia.org/wiki/File:SPI_three_slaves.svg`

SPI Bus

- Bidirectionaler Bus
- Master/Slave Prinzip
- *Chip Select (CS):* Auswahl des Slaves
- *Master Output Slave Input (MOSI):* Datenleitung vom Master zum Slave
- *Master Input Slave Output (MISO):* Datenleitung vom Slave zum Master
- *Serial Clock (SCK):* Taktleitung

SPI Bus Schaltung

mit RFID Chip



Nützliche Bash Werkzeuge

`flashrom` Zum Auslesen von SPI Flashes (BIOS Chips)

```
sudo flashrom -p linux_spi:dev=/dev/spidev0.0 -r read01.bin
```

SPI Bus

Installation

```
sudo pip install spidev
```

Ansteuerung über Python

eines MCP23S08 GPIO Expanders

```
import spidev

# SPI Bus oeffnen
spibus =  spidev.SpiDev()
# spi-bus 0, chip select 0
spibus.open(0, 0)
# Setze Parameter
spi.max_speed_hz = 5000
spi.mode = 0b01
# Schreibe Daten raus
spibus.xfer([0x40, 0x00, 0xfe])
# Setze LED
spibus.xfer([0x40, 0x0a, 0x01])
# Lese Taster
tmp = spibus.xfer([0x41, 0x09, 0x00])
print("Taster: " + str((tmp[2] & (1<<1)) > 0))
```

Weitere nützliche Bauteile

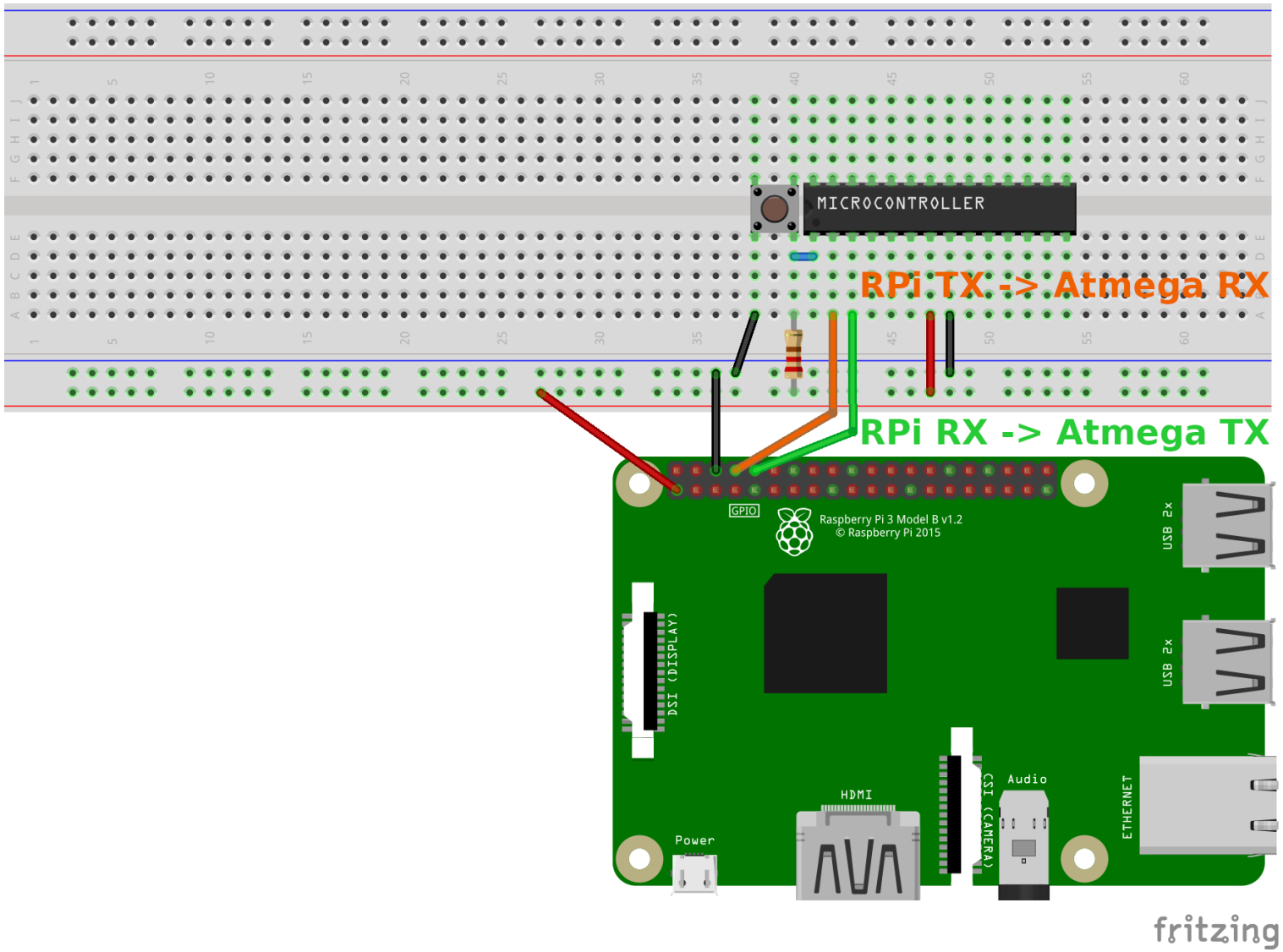
- SPI Flashes
- RFID Chip Reader
- Displays (Nokia 5510, . . .)
- ADCs und DACs
- Ethernet Controller (ENC28J60, Wiznet W5500)
- GPIO Expander
- . . .

Serielle Schnittstelle

- auch *Universal Asynchronous Receiver Transmitter (UART)* genannt
- Bidirektionale Byteweise Übertragung von Daten (meistens ASCII Characters (Buchstaben))
- Datenleitung zum Senden: *Transmit (TX)*
- Datenleitung zum Empfangen: *Receive (RX)*
- Leitungen müssen kreuzweise mit Gegenstelle verbunden werden
- Datenübertragung bis 3Mbit/s

Schaltung mit serieller Schnittstelle

Kommunikation zwischen Atmega uC & Raspberry Pi



Nützliche Bash Tools

Serielle Terminals, wie `screen` oder `minicom`

Beispiel: Aufbau einer seriellen Verbindung mit `screen`

```
sudo screen /dev/ttyAMA0 115200
```

115200: Verwendete *Baudrate*

Ansteuerung über Python

Installation

```
sudo pip install pyserial
```

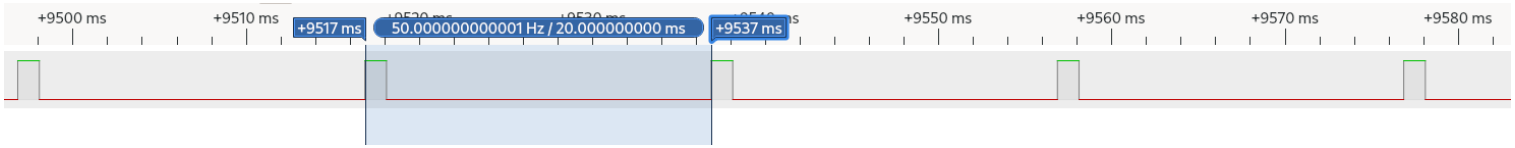
```python
import serial

# Schnittstelle oeffnen
ser = serial.Serial("/dev/AMA0", 9600, timeout=1)

# Schreibe Daten raus
ser.write("Hello World".encode("utf8"))

# Empfange 5 Byte
data = ser.read(5).decode()

print(data)
```

Weitere nützliche Bauteile

- Pegelwandler MAX232N für RS232
- USB-UART Converter: MCP2200, CH340G, FT232RQ
- ...

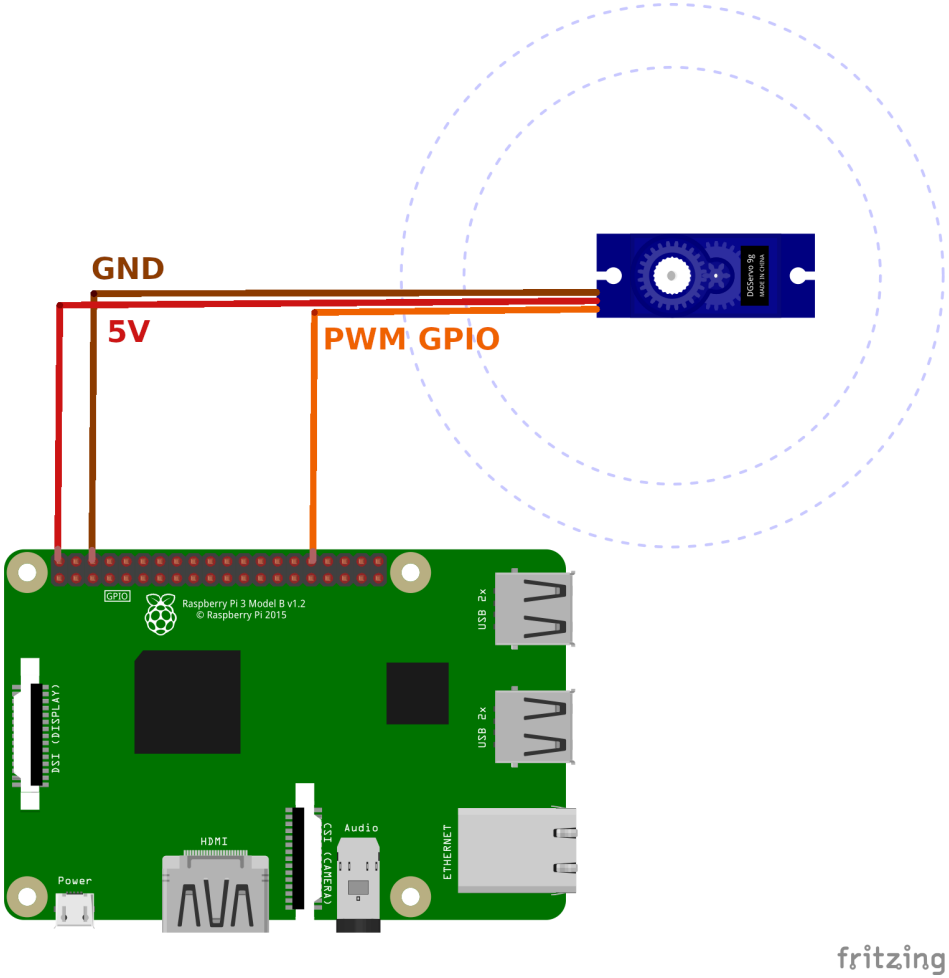
PWM

- Ausgabe einer analogen Spannung über Pulseweitenmodulation
- Z.B. Zum Ansteuern von kleinen Servo-Motoren
- Unterscheidung von Hardware und Software PWM



Schaltung mit PWM

Ansteuerung eines kleinen Servo-Motors



Ansteuerung über Python

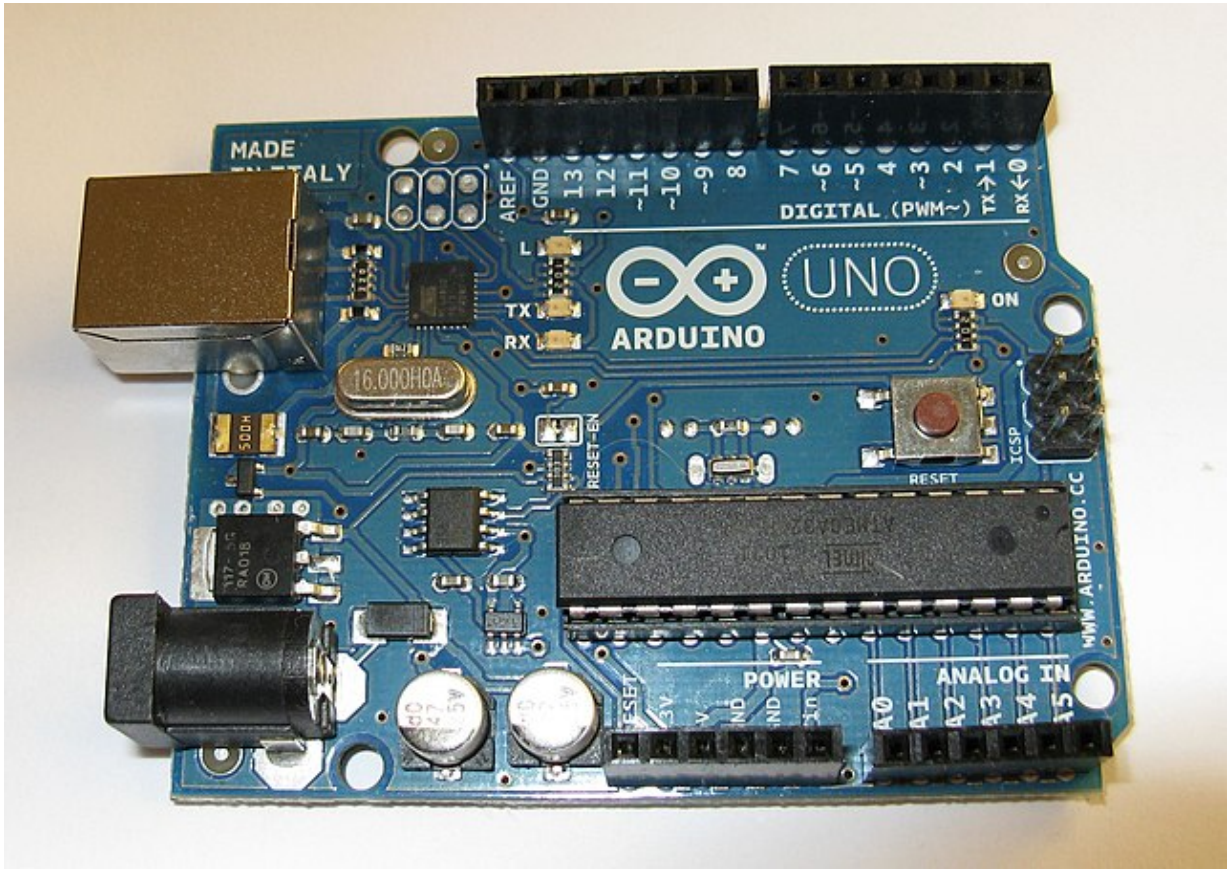
am Beispiel eines kleinen Servo-Motors

```
import RPi.GPIO as GPIO

servoPIN = 12
GPIO.setmode(GPIO.BCM)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 als PWM mit 50Hz
p.start(2.5) # Initialisierung

sleep(1)

# Aendere Duty Cycle
p.ChangeDutyCycle(5)

sleep(1)
p.stop()
GPIO.cleanup()
```

Weitere nützliche Bauteile

- Micro RC Mini Servo SG90 9G
- Transistoren
- RC-Glieder (Spule und Widerstand), um analogen Sinus zu modellieren

Warum genau ein Raspberry Pi?



Quelle: `https://commons.wikimedia.org/wiki/File:Arduino_uno_1.jpg`

Lizenz: 🄯

Ein Arduino kann das alles doch auch!

Vorteile Raspberry Pi & Linux

- Einfacheres Debugging
- Interpreterfähige Sprachen (Python)
- Multitaskingfähig
- Einfache Einbindung von Netzwerkfunktionen (Webserver, Mail, File Sharing, ...)

Nützliche Programme

Webserver

- Apache
- Nginx
- lighttpd

Kommunikation

- sendmail
- smstool

Grafische Nutzeroberflächen

- GTK
- Tkinter
- qt

Nützliche Programme

Weiter nützliche Tools

- Musikserver: mpd
- Industrielle Kommunikation/Home-Automation: OPC UA, MQTT, Modbus, CAN, ...
- Anbindung an Datenbanken (sqlite, PostgreSQL, MariaDB, ...)
- ...

Beispiel

- BMP280 wird von Python Programm zyklisch ausgelesen
- Lighttpd Webserver läuft
- Python Programm aktualisiert Temperaturwert in Webserver


10 PACK
Perfekt für Filme, Fotos und Datensicherung
DVD+R
4.7GB
120min
16fache
EDEKA
GUT & GÜNSTIG
SAMSUNG
extrem leistungsstark
POWER CELL
8 x AA

Quellen

- Richardson & Wallace, Raspberry Pi für Einsteiger, O'Reilly 2013
- Prof. Platte, Jürgen: Raspberry Pi und Linux, `http://www.netzmafia.de/skripten/hardware/RasPi/`, letzter Aufruf 03.11.2021