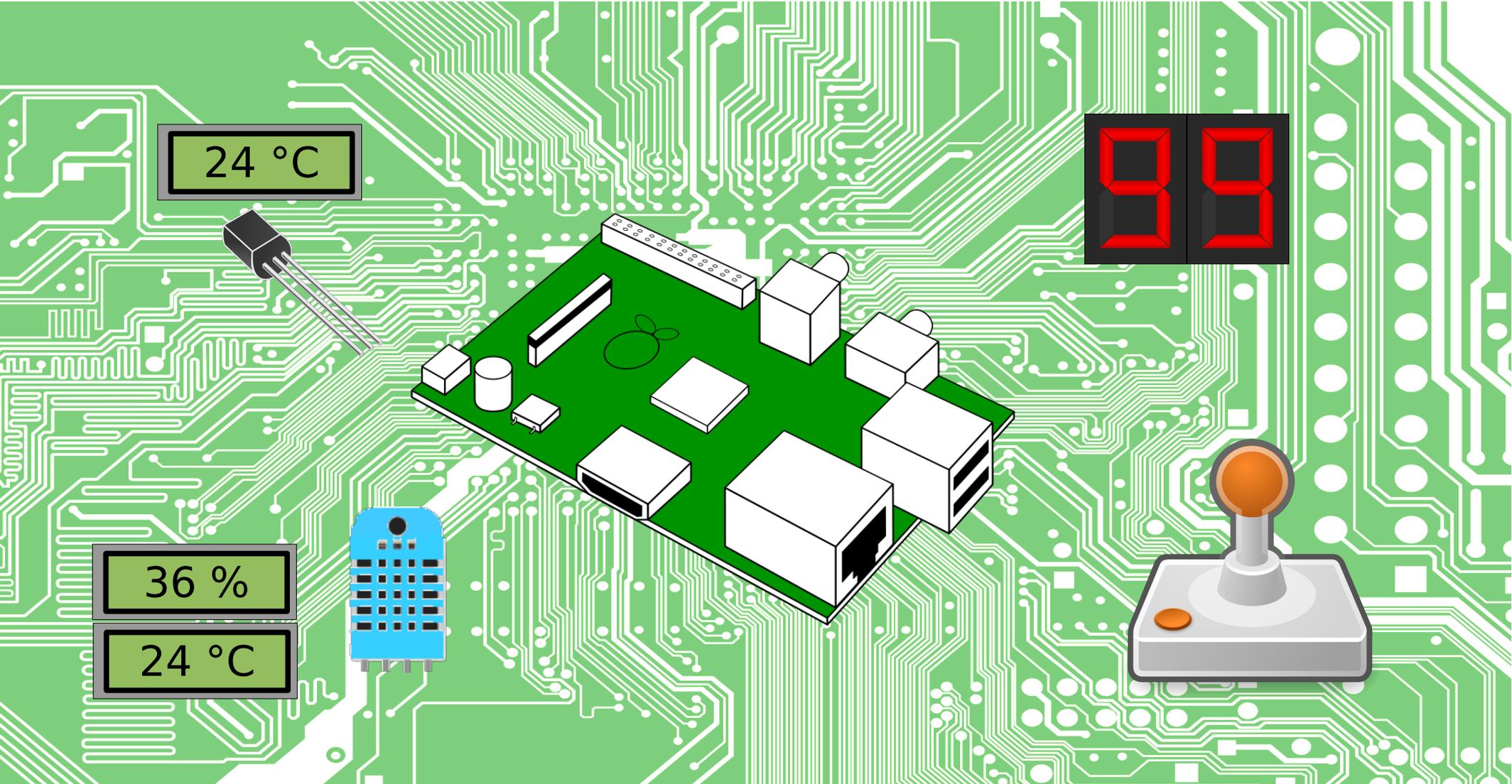


Steuern und Messen mit Raspberry Pi



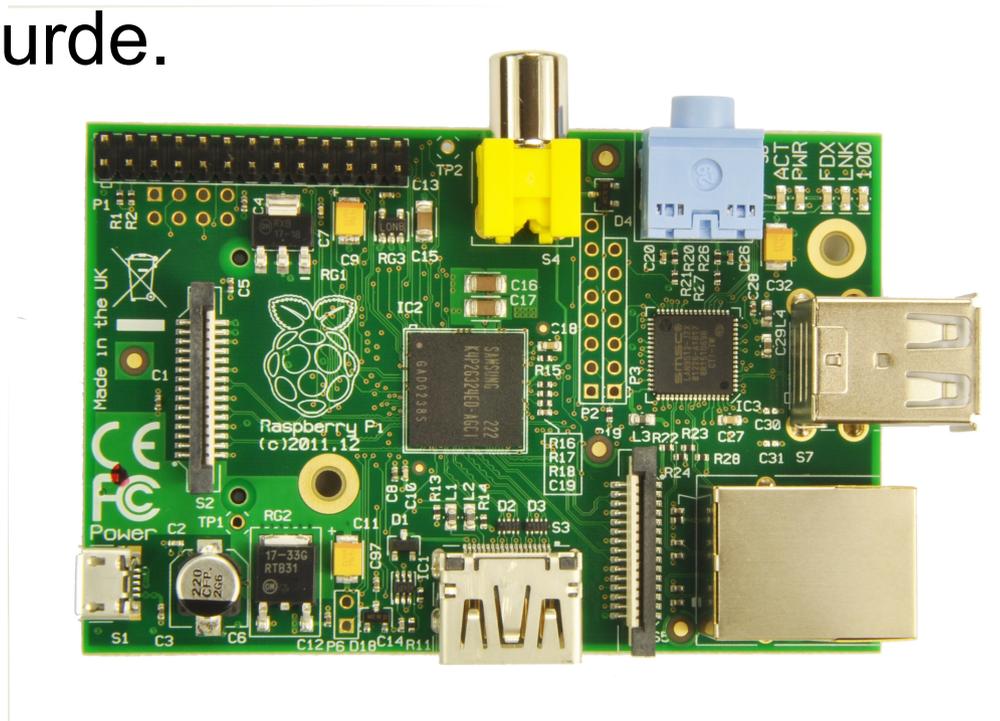
Martin Strohmayer

Thanks to
Openclipart, Fritzing, elinux.org, Liz

Grazer Linuxtage 2014 (GLT14)

Raspberry Pi

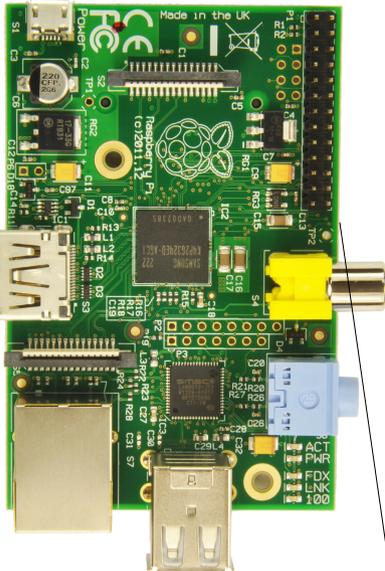
Raspberry Pi ist ein kreditkartengroßer Einplatinen-Computer, der von der Raspberry Pi Foundation entwickelt wurde.



Die Raspberry Pi Foundation ist eine Wohltätigkeitsorganisation in Großbritannien. Sie hat sich zum Ziel gesetzt, das Studium der Informatik und verwandter Themen zu fördern, insbesondere im Schulbereich.

GPIO

General Purpose Input/Output Universelle, digitale Ein-/Ausgabe-Kanäle



5,0 V (max. 300 mA/USB Netzteil)

3,3 V (max. 50 mA)

Ground

UART – 3,3 V (GPIO14, GPIO15)

PWM – 3,3 V (GPIO18)

I2C – 3,3 V (GPIO0/2, GPIO1/3)

SPI (GPIO7- GPIO11)

I/O – 3,3 V

(GPIO4,GPIO17,GPIO21-GPIO25, GPIO27)

Schutzbeschaltung



**GPIO Anschlüsse dürfen nicht in Kontakt mit 5 V Versorgung gelangen
Keine Kurzschlüsse auf den GPIOs (16 mA max.)**

**Vorwiderstand am GPIO bei Schaltern
5 V Pins und Leitungen abdecken, isolieren, trennen
Logik IC oder Spannungsteiler für Pegelanpassung**

**3,3 V → 5 V (Ausgang)
HCT Logikbausteine
(74HCT244)
OC – Schaltung/IC**

**5 V → 3,3 V (Eingang)
74HC4050
Spannungsteiler**

**I2C 3,3 V ↔ 5 V
PCA9515**

WiringPi

C Library für GPIOs

Install:

```
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
```

Commandline Tool:

```
gpio
```

Verwendung mit C-Source:

```
gcc -o example example.c -lwiringPi
```

Aufrufe C-Programm:

```
sudo ./example
```

Example.c

```
#include <wiringPi.h>

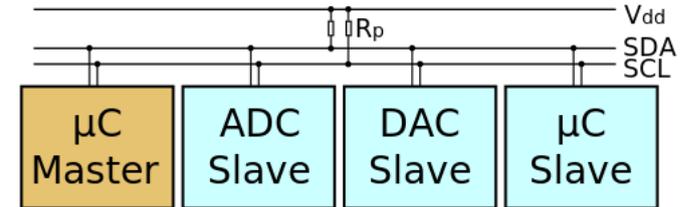
const int GPIO4 = 4;
const int GPIO17 = 17;

int main (void) {
    int Button=LOW;
    WiringPiSetupGpio();
    pinMode (GPIO4, OUTPUT) ;
    pinMode (GPIO17, INPUT) ;
    Do {
        digitalWrite (GPIO4, HIGH) ;
        delay (500) ;
        digitalWrite (GPIO4, LOW) ;
        Delay (500) ;
        Button=digitalRead(GPIO17);
    } while (Button==LOW);
    return 0 ;
}
```

I2C-Bus

Inter-Integrated Circuit (I-Quadrat-C)

- Serieller Bus (synchron)
- Master-Slave-Bus
- Standard Mode (Sm) 100 kbit/s bidirektional



Console

```
i2cdetect -y 1
```

	0	1	2	3	4	5	6	7	8	9	a	...
00:	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--
40:	--	--	--	--	--	--	--	--	--	--	--	--
50:	--	51	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--
70:	--	--	--	--	--	--	--	--	--	--	--	--

Slave Adresse

```
i2cset -y 1 0x51 0x03 0xff
```

```
i2cget -y 1 0x51 0x03  
0xff
```

Daten

Slave Adresse

Speicher/Daten Adresse

C Programm

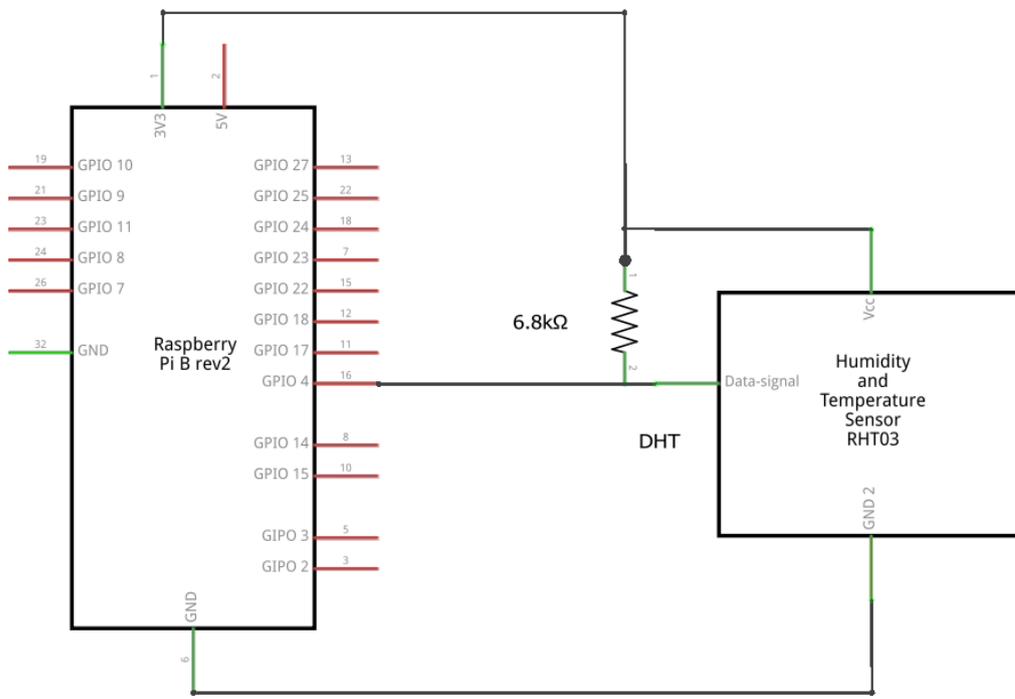
```
if ((fd = open("/dev/i2c-1", O_RDWR)) < 0)  
    exit(1);  
if (ioctl(fd, I2C_SLAVE, SlaveAddr) < 0) {  
    close(fd);  
    exit(2);  
}  
  
int I2CWrite(int fd, unsigned char Reg, unsigned char Value){  
    unsigned char buf[2];  
    buf[0] = Reg;  
    buf[1] = Value ;  
    printf("write \n");  
    if (write(fd, buf, 2) != 2)  
        return false;  
    return true;  
}  
  
int I2CRead(int fd, unsigned char Reg, unsigned char* Value)  
{  
    unsigned char ValueL = 0;  
    if (write(fd, &Reg, 1) != 1)  
        return false;  
    if (read(fd, &ValueL, 1) != 1)  
        return false;  
    *Value = ValueL;  
    return true;  
}
```

1-Bus Sensor

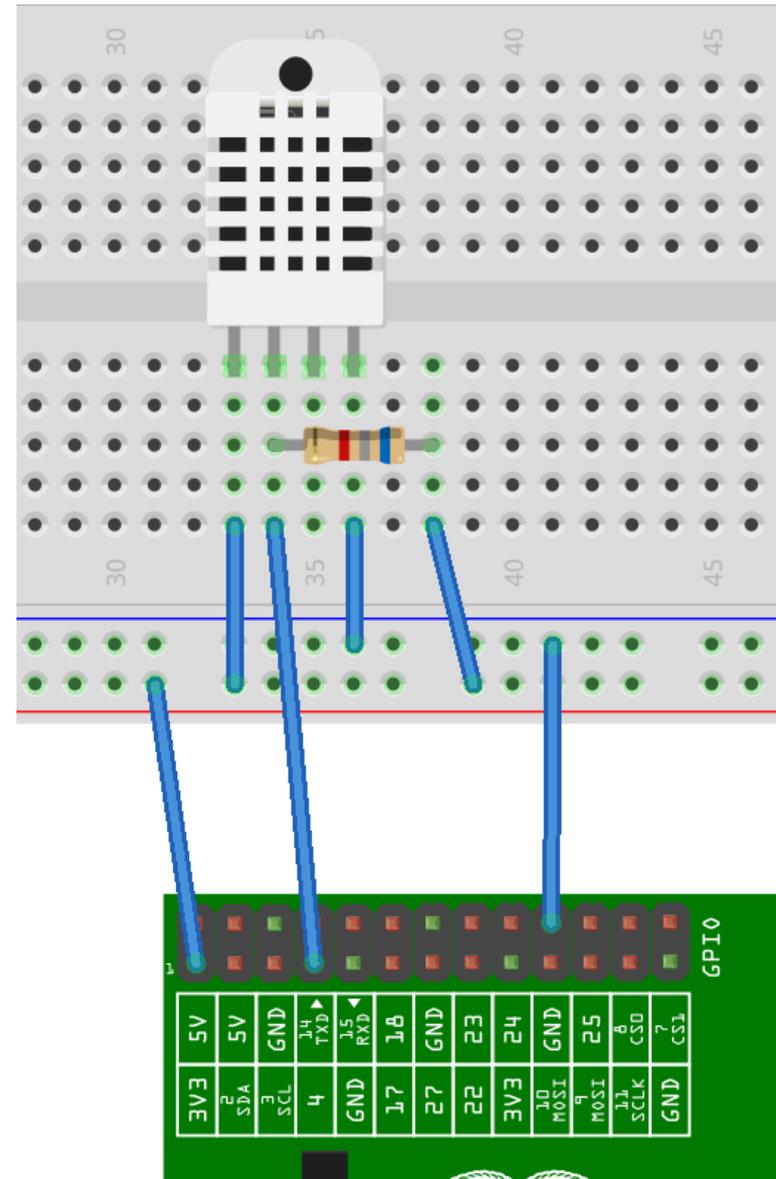
DHT11/DHT22

Lufttemperatur und Luftfeuchte

- Ultra Low Cost / Low Cost
- Wenig Bauteile
- Einfaches Auslesen mit C und Python
- Programme/Source schon vorhanden



fritzing

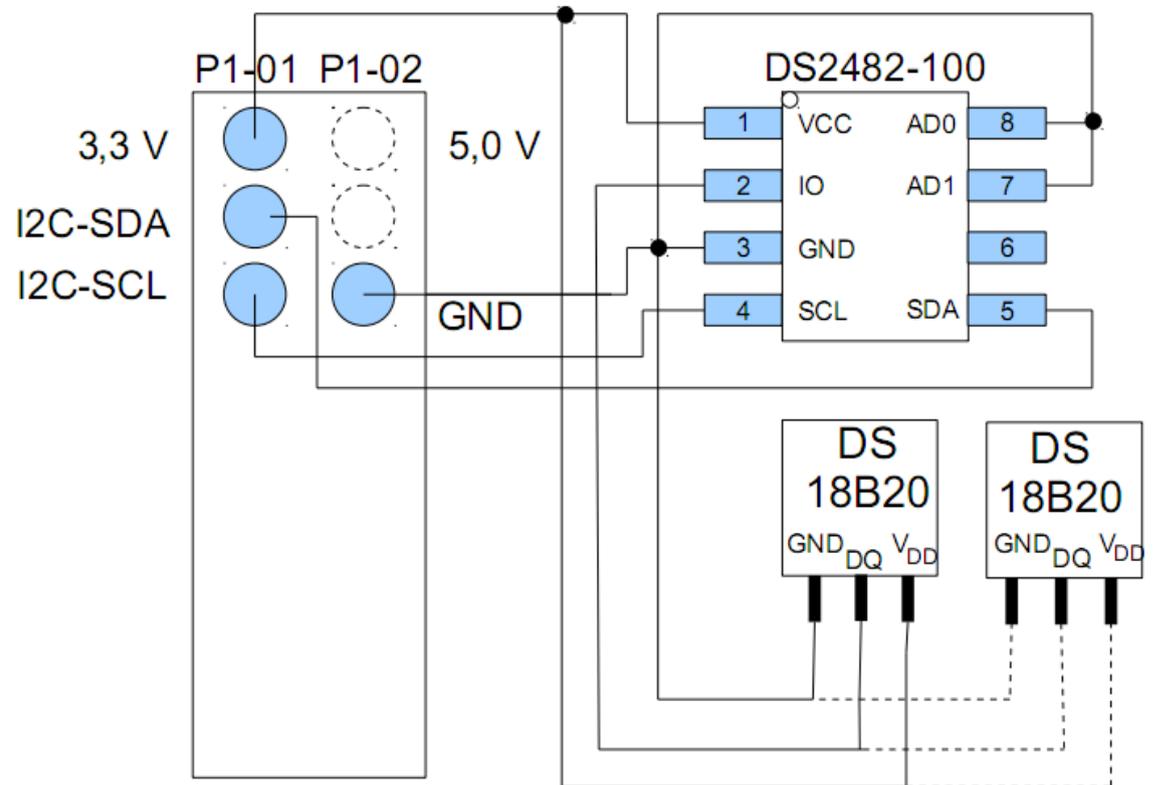
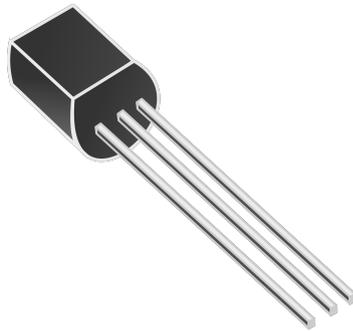


1-Wire Sensor

DS18B20

Temperatur

- Low Cost
- Unterstützung über OWFS 1-Wire File System
- Sensoren mit Adresse
- Busmaster für mehrfach Sensoren am Bus



```
owfs -m /owfs /dev/i2c-1  
cat /owfs/28.A5F672020000/temperature && echo " C"  
24.69 C
```

Feuchte Sensor

HH10D

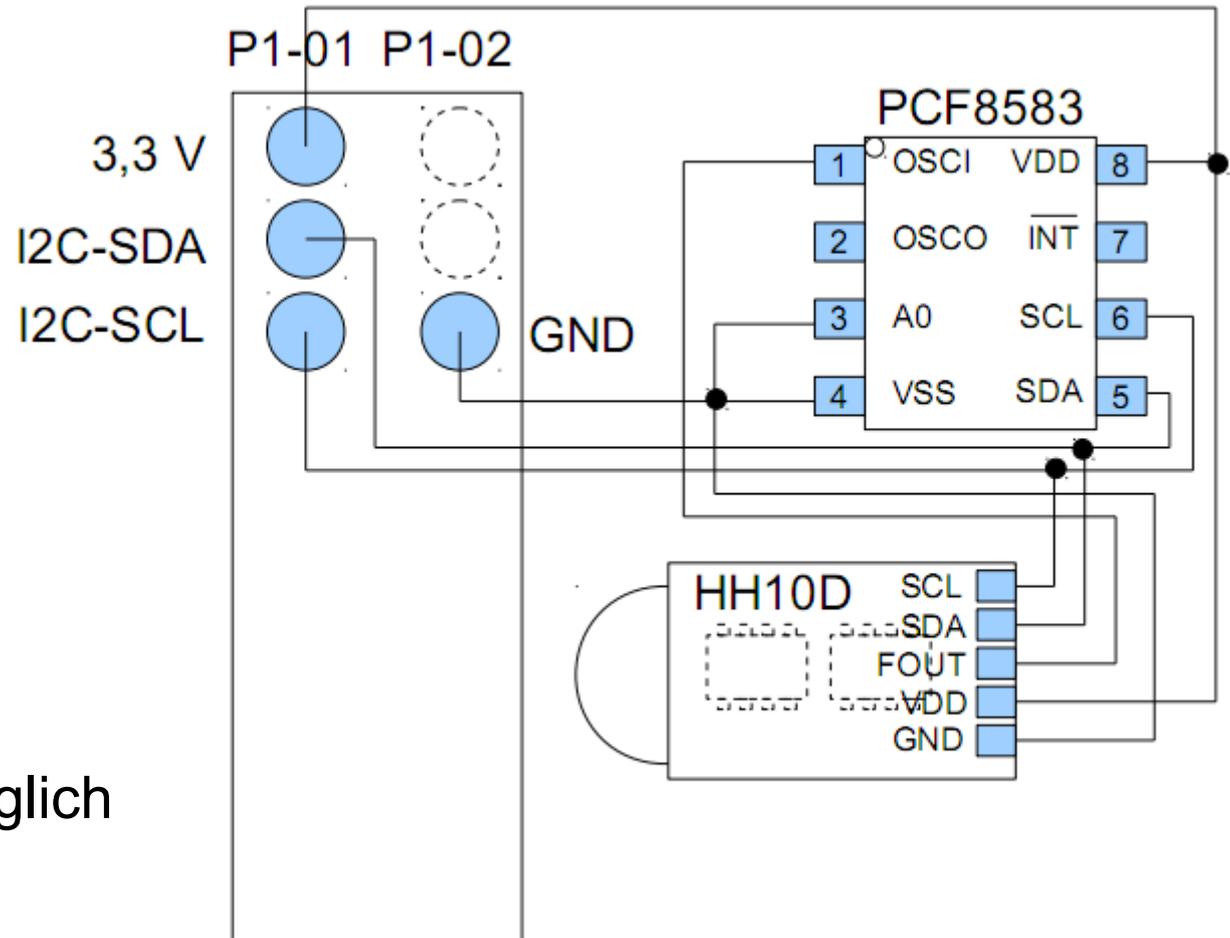
Luftfeuchte

- Kein Protokoll
- Feuchte → Frequenz
- Kalibrierter Sensor
- Kalibrierdaten per I2C-Bus auslesbar

PCF8583

Zähler I2C-Bus

- Frequenz Messung möglich

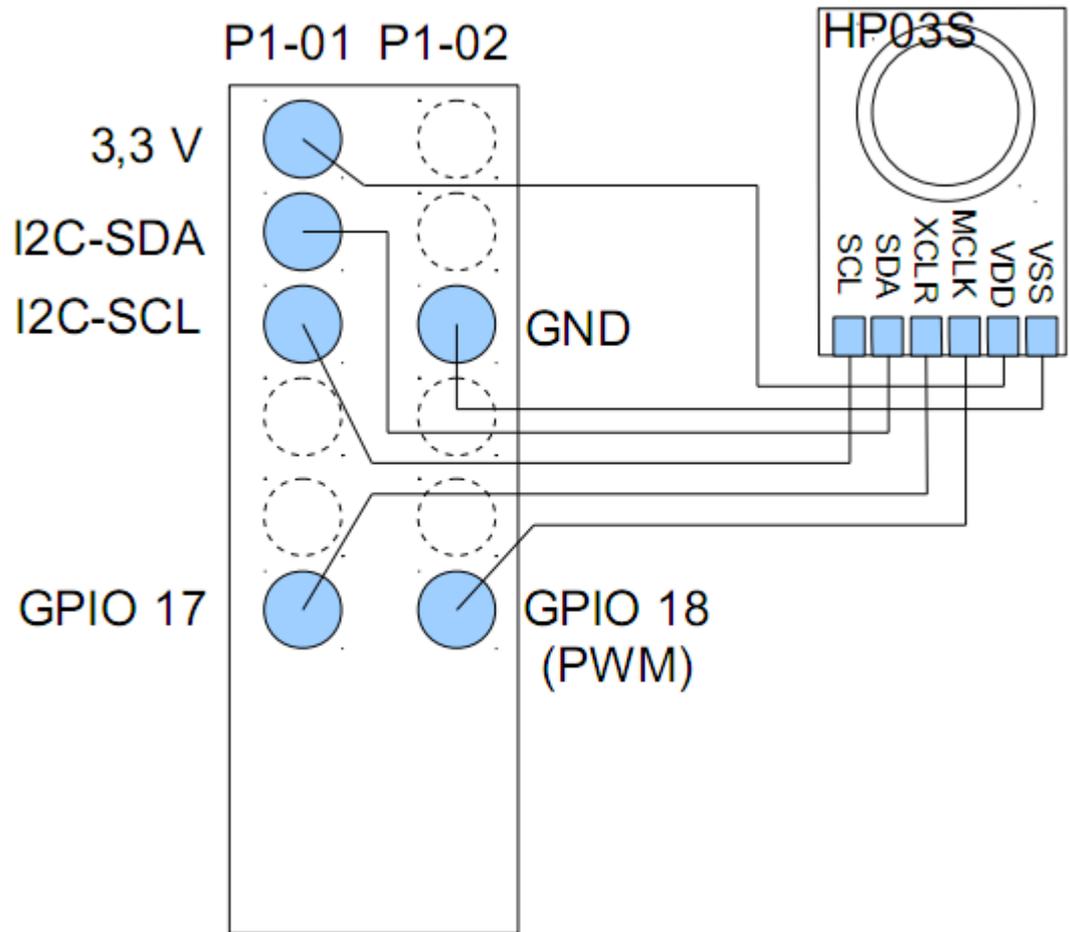
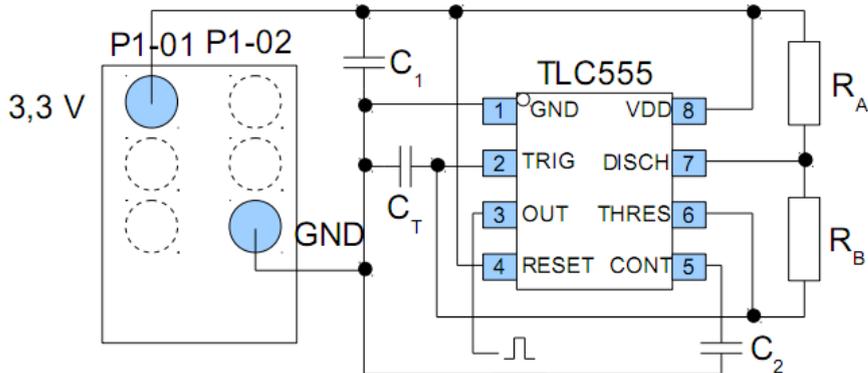


Druck Sensor

HP03S

Luftdruck und Lufttemperatur

- I2C
- 32 kHz Takt per PWM oder Taktgenerator

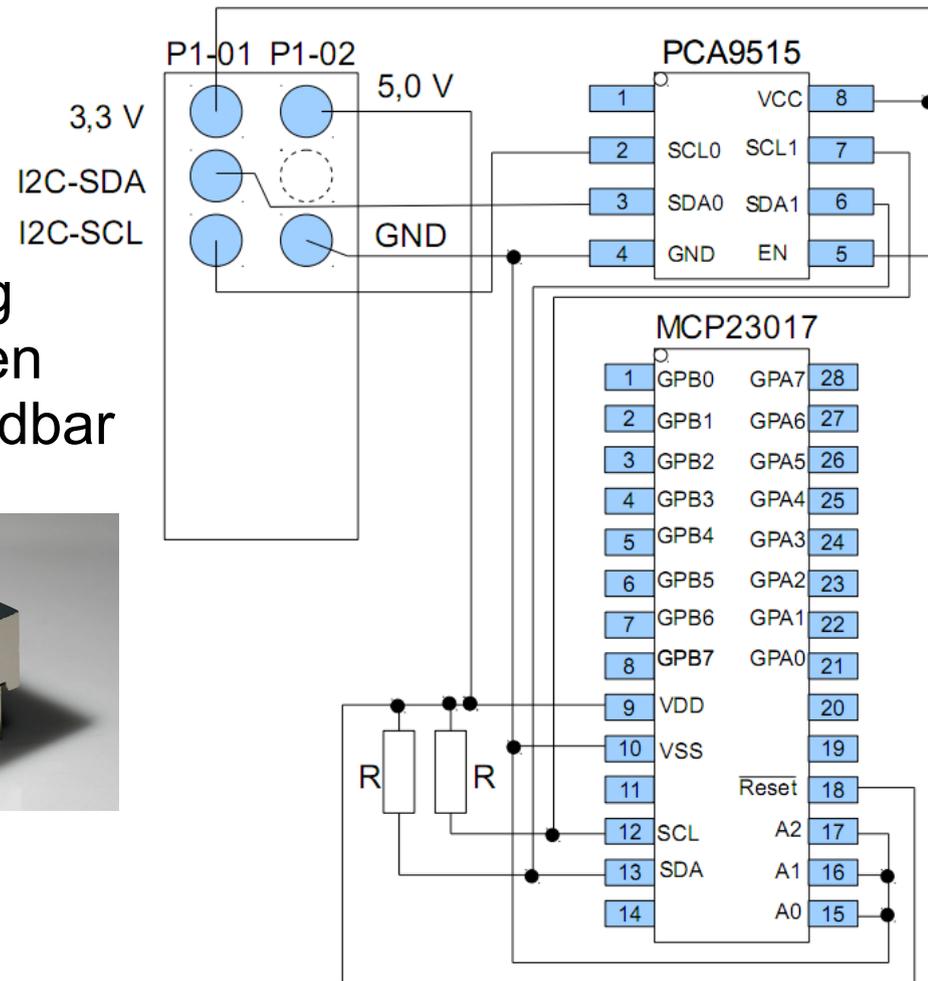
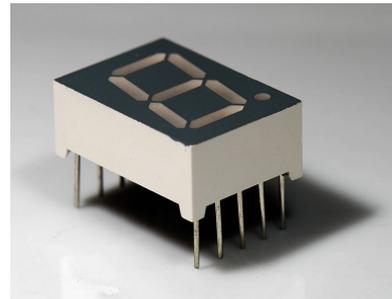


Port-Erweiterung 3,3 V/5 V

MCP23017

16 Bit Port-Erweiterung

- I2C Bus
- 3,3 V oder 5 V Betriebsspannung
- 8 Bit gemeinsam lesen/schreiben
- Bis zu 8 IC pro I2C-Bus verwendbar



Links

Raspberry Pi Hardware

http://elinux.org/RPi_Low-level_peripherals

Wiringpi

<https://projects.drogon.net/raspberry-pi/wiringpi/>

Vortrag kann auf

<http://glt14-programm.linuxtage.at>

bewertet werden

Buch zum Vortrag und Projektstand:

Raspberry Pi - Projekte: Raspberry Pi als HTPC, Retro-Spielkonsole und für Elektronikprojekte nutzen

Meine Page: <http://evil.hn.vc/>

Twitter: @m_stroh

Meine e-mail: m.stroh@ymail.com