

RaspiLab -
Lehrerworkshop

**Mit digitalen Werkzeugen den Unterricht bereichern –
einfach messen mit dem Raspberry Pi**

Unterlage

2. Auflage

Stand: 23.09.2019

Autoren:

André Delbos

Prof. Dr. Claus Brell

Vorwort zur 2. Auflage

Sie halten die zweite Auflage der Anleitung zu den Raspberry Pi Lehrerworkshops in den Händen. Seit 2017 werden die Lehrerworkshops am Fachbereich Wirtschaftswissenschaften, Fachgruppe Wirtschaftsinformatik, zusammen mit kooperierenden Gymnasien durchgeführt.

Ziele der Lehrerworkshops sind:

1. Austausch über Möglichkeiten übergreifender didaktische Konzepte
2. Die Hochschule vermittelt die aktuellen Erkenntnisse aus studentischen Projekten.
3. Lehrer der kooperierenden Schulen entwickeln Ideen, wie sie mehr MINT in ihren Unterricht bekommen.
4. Lehrer erfahren die Hochschule Niederrhein als Ort, den sie ihren Schülern und insbesondere Schülerinnen empfehlen können.
5. Die Hochschule erfährt, welche Perspektive Lehrer auf die Hochschule haben.

Damit Lehrerinnen und Lehrer in der Schule gleich weitermachen können, erhalten sie im Workshop Materialien (Raspberry Pi Zero und Elektronik), die sie mit nach Hause nehmen.

Die Unterlage soll interessierten Lehrern und Lehrerinnen als erste Handreichung für weitergehende Experimente mit ihren Schülern dienen.

Die Anleitung ist bewusst knapp gehalten und umfasst im Wesentlichen den praktisch/technischen Teil (eineinhalb Stunden) des insgesamt dreistündigen Workshops:

1. Vorstellung der Möglichkeiten mit Linux-Kleinrechnern und der Demonstratoren der Hochschule (Sensorsysteme, KI, Überwachung Bienen, digitale Nistkästen)(Vortrag)
2. Raspi Zero „from the scratch“ aufsetzen (Vortrag)
3. Den Raspberry Pi kennen lernen (Praxis)
4. Eine Leuchtdiode ein und ausschalten (Praxis)
5. Zwei Temperatursensoren anschließen und auslesen (Praxis)
6. Ein Bild mit der Pi Kamera aufnehmen (Praxis)
7. Bilder ins Internet stellen (Praxis)
8. Anwendungsmöglichkeiten in der Schule (Brainstorming)
9. Austausch weitergehender Ideen (Moderation)

Eine Linksammlung zum Workshop findet man unter

<https://cbrell.de/blog/raspberry-pi-lehrerworkshop/>

Mönchengladbach, September 2019

André Delbos

Claus Brell

Inhaltsverzeichnis

1	Voraussetzungen und Grundeinstellungen des Raspberry Pi	3
2	Raspberry Pi Pinbelegung und arbeiten mit dem Breadboard	4
2.1	Raspberry Pi Pinbelegung.....	4
2.2	Breadboard.....	4
3	Erste Mission: Den Raspberry Pi und Raspbian kennen lernen	5
4	Zweite Mission: Eine LED steuern.....	6
4.1	Ein Python-Programm für die LED-Steuerung erstellen.	7
5	Dritte Mission: Temperaturen messen mit dem DS1820 Sensor	9
5.1	Anschlüsse des DS1820 und Verkabelung mit dem Raspberry Pi	9
5.2	Verkabelung des DS1820 mit dem Raspberry Pi.....	9
5.3	Vorbereitung der Temperaturmessung und erster manueller Test.....	10
5.4	Ein Python- Programm für die Temperaturmessung erstellen	11
6	Vierte Mission: Raspberry Pi Kamera nutzen.....	12
6.1	Raspberry Pi Kamera anschließen und vorbereiten.....	12
6.2	Python Script: Foto schießen und auf den Webserver hochladen.....	12
7	Schlussbemerkung	14

1 Voraussetzungen und Grundeinstellungen des Raspberry Pi

Sie arbeiten im Workshop mit einem Raspberry Zero. Der Zero hat, wenn er als Messrechner eingesetzt wird, viele Vorteile:

- Kleiner Formfaktor
- Geringer Stromverbrauch
- Günstiger Preis

Damit Sie gleich loslegen können, ist im Workshop das Betriebssystem (Raspbian Lite, ein Debian Derivat) schon aufgespielt. Raspbian Lite hat keine grafische Benutzungsoberfläche, wir arbeiten mit Linux-Konsolenbefehlen. Wenn Sie selber einen Raspberry aufsetzen wollen, finden Sie hierzu im Internet viele (gute) Anleitungen, z. B. hier:

<https://cbrell.de/blog/raspilab-wetterstation-grundsystem-aufsetzen/>

Das selber Aufsetzen dauert, wenn Sie etwas Übung haben, 15 Minuten, wenn Sie das das erste Mal machen, veranschlagen Sie eine Stunde.

Folgende Einstellungen sind bereits vorgenommen:

- Einstellung für ein WLAN mit der SSID „raspilab“ und dem Kennwort „raspilab“.
- Deutsche Tastatur und Zeitzone

Sie werden im Workshop mit folgenden Materialien arbeiten:

3 Erste Mission: Den Raspberry Pi und Raspbian kennen lernen

Der Raspberry Pi ist ein Einplatinencomputer, der im Workshop ohne grafische Benutzeroberfläche verwendet wird. Seine Steckkontakte (GPIO) bieten die Möglichkeit, Sensoren anzuschließen und damit etwas zu messen oder zu steuern.

Ein Computer braucht ein Betriebssystem. Wir verwenden Raspbian, das auf dem Linux-Derivat Debian basiert und ganz einfach zu bedienen ist. Mit dem Raspberry Pi und dem Betriebssystem Raspbian wollen wir die Missionen des Workshops bewältigen.

a1) Falls Sie einen Monitor nutzen: Anschließen und starten. Schließen Sie zunächst den Monitor mit dem HDMI-Kabel und die Tastatur mit dem USB-Kabel an den Raspberry Pi an. Schalten Sie den Monitor ein, Schließen Sie das Netzteil an den Raspberry Pi an, stecken Sie das Netzteil in ein Steckdose. Es sollten nun Zahlen und Buchstaben über den Bildschirm huschen. Warten Sie, bis der Startprozess des Raspberry Pi zur Ruhe gekommen ist.

Der Benutzer ist „pi“ und das Passwort „raspberry“.

a2) Falls Sie über Netzwerk und ssh arbeiten: Loggen Sie sich ein mit ssh [pi@<NameDesPi.local](#) oder ssh pi@<ip-adresse>. Auf einem Mac- oder Linux-Rechner verwenden Sie das Terminal, auf einem Windows Rechner benötigen Sie das Programm Putty.

b) Linux Befehle nutzen. Wir werden ausschließlich mit der Konsole und ohne Maus arbeiten. Der Raspberry Pi wird durch Befehle gesteuert.

Geben Sie

```
ls
```

ein und drücken Sie die ENTER-Taste und schauen was passiert.

Erklärung: Der Befehl ls (für list) gibt den Inhalt des Verzeichnisses aus, in dem Sie sich gerade befinden.

Geben Sie

```
cd ..
```

ein und drücken Sie die ENTER-Taste und schauen, was passiert.

Erklärung: Der Befehl cd (für changedirectory) ändert das Verzeichnis, in dem Sie sich gerade befinden. cd .. geht eine Verzeichnisebene höher, cd abc öffnet das Verzeichnis abc, cd /xyz öffnet das Verzeichnis xyz, wenn es sich im Wurzelverzeichnis root befindet.

c) Texteditor nano benutzen. Mit nano können Sie Texte auf dem Raspberry schreiben und speichern. Geben Sie

```
nano beispieltext.txt
```

ein und drücken Sie die ENTER-Taste. Geben Sie einen Text ein, z. B. "Ich arbeite mit dem Raspberry Pi." Speichern Sie dann den Text mit der Tatenkombination *ctrl+o*. Beenden Sie nano mit der Tastenkombination *ctrl+x*.

Erklärung: *nano* ist ein Texteditor unter Linux. Mit *nano* erstellen wir alle Programme.

d) Einen Dateinhalt mit cat anschauen und mit rm löschen. Geben Sie ein

```
cat beispieltext.txt
```

und drücken Sie die ENTER-Taste und schauen, was passiert.

Alle Linux-Befehle schließen Sie mit der ENTER-Taste ab. Linux-Befehle lassen sich gut googlen. Weitere Linux-Befehle sind:

mv bewegt eine Datei von einem Verzeichnis in ein anderes.

cp kopiert eine Datei.

rm löscht eine Datei

Lassen Sie sich Ihre Datei beispieltext.txt anzeigen mit dem Befehl ls, löschen Sie sie mit der Befehlsfolge

```
rm beispieltext.txt
```

und überzeugen Sie sich mit ls, dass die Datei nicht mehr da ist.

Erklärung: Dateien sind wichtig in Linux und auf dem Raspberry Pi, alles wird über Dateien gesteuert. Daher ist es notwendig, dass Sie Dateien erzeugen, löschen, verschieben und verändern können.

In den weiteren Missionen werden Sie Schritt für Schritt eine Leuchtdiode, Temperatursensoren und dann eine Kamera mit dem Raspberry Pi in Betrieb nehmen. Alle Bauteile gelangen nach und nach auf das Breadboard und werden mit den bunten Kabeln (Jumperkabel) an den Raspberry Pi angeschlossen. Nach der letzten Mission wird Ihr Breadboard ähnlich wie in Abb. 2 aussehen.

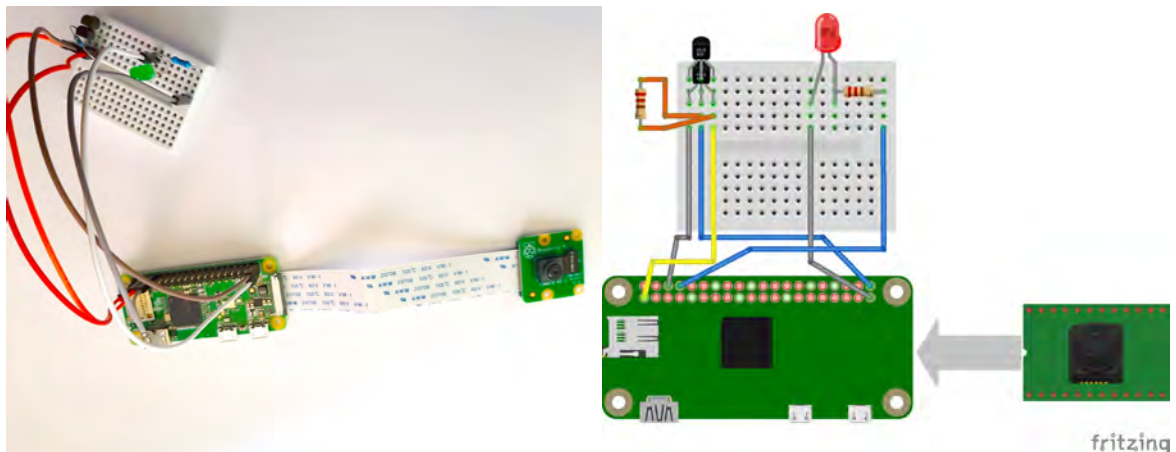


Abb. 2: Verkabelung nach Abschluss aller Missionen

Erste Mission geschafft!

4 Zweite Mission: Eine LED steuern.

Die Mission hat das Ziel, mit dem Raspberry Pi eine LED (Leuchtdiode) zu schalten. Eine Leuchtdiode sendet Licht aus, wenn Strom in der richtigen Richtung durch sie fließt. Dadurch kann sie als Signallampe verwendet werden.

Abb. 3 zeigt den Schaltplan für eine LED am Beispiel der LED:

Pin 8 (GPIO14)

Ground (Pin 6)

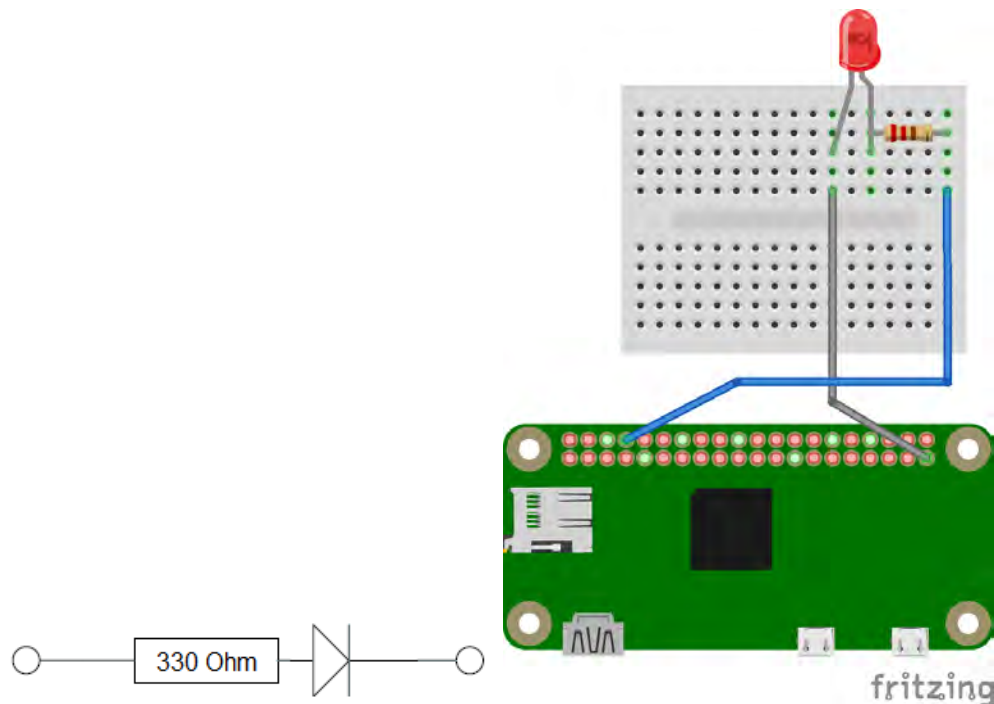


Abb. 3: Schaltung für den Anschluss einer LED

Schließen Sie nun die LED an: Die LED mit der langen Seite mit dem Widerstand (330 Ohm) verbinden. Pin 8 über Jumperkabel mit den Widerstand verbinden, die kurze Seite der LED über Jumperkabel mit Pin 39 (GND) verbinden wie in Abb. 3.

4.1 Ein Python-Programm für die LED-Steuerung erstellen.

Die Programmierung auf dem Raspberry erfolgt im Workshop durchgängig mit der Programmiersprache *Python*.

Sie werden zwei Programmiermethoden kennen lernen (ohne Funktionen und – optional - mit Funktionen), um die Ampel zu steuern.

Schritt 1: Erstellen Sie mittels des folgenden Befehls eine Datei mit dem Namen *ledampel.py*:

```
nano LED.py
```

Schritt 2: Fügen Sie nun den folgenden Quelltext ein (Groß / Kleinschreibung ist wichtig):

```
#!/usr/bin/python
#coding utf-8
import RPi.GPIO as GPIO
import time

#GPIOs vorbereiten
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.OUT)
#LED anschalten
GPIO.output(8, GPIO.HIGH)
time.sleep(2)

#LED Ausschalten
GPIO.output(8, GPIO.LOW)
```

```
time.sleep(2)
GPIO.cleanup()
```

Schritt 3: Speichern Sie den Quelltext mit der Tastenkombination *Strg + o*, verlassen Sie dann *nano* mit der Tastenkombination *Strg + x*

Schritt 4: Geben Sie Ihrem Programm mit dem Befehl

```
sudo chmod 777 led.py
```

alle Rechte, so dass Sie es ausführen können.

Erklärung: *sudo* heißt, einen Befehl mit den Rechten eines Administrators auszuführen. *chmod* ändert die Rechte einer Datei, *777* heißt, Sie dürfen alles mit der Datei tun.

Schritt 5: Führen Sie Ihr Programm aus mit dem Befehl

```
./led.py
```

Erklärung: „./“ bedeutet, dass Sie das Programm in dem Verzeichnis ausführen, in dem Sie sich gerade befinden.

Zweite Mission geschafft.

Nur optional: Mission 2a:

Falls Sie sehr schnell fertig sind und sobald Ihr Programm funktioniert oder Sie zuhause weiter experimentieren wollen, können Sie die Variante mit Python-Funktionen ausprobieren:

Schritt 1: Rufen Sie Ihr Programm wieder auf mit

```
nano led.py
```

Schritt 2: Ändern Sie Ihr Programm, dass es wie folgt aussieht (Einrückung ist wichtig und muss immer gleich sein):

```
#!/usr/bin/python
#coding utf-8
import RPi.GPIO as GPIO
import time
#GPIOs vorbereiten
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8, GPIO.OUT)
#LED anschalten
def AN():
    GPIO.output(8, GPIO.HIGH)
#LED Ausschalten
def AUS():
    GPIO.output(8, GPIO.LOW)

AN()
time.sleep(1)
AUS()
time.sleep(1)
AN()
time.sleep(1)
AUS()
time.sleep(1)
```



```
GPIO.cleanup()
```

Schritt 3: Speichern Sie den Quelltext mit *Strg + o*, verlassen Sie dann *nano* mit *Strg + x*

Schritt 4: Führen Sie Ihr Programm aus mit den Befehl

```
./led.py
```

5 Dritte Mission: Temperaturen messen mit dem DS1820 Sensor

Ziel ist, mit Hilfe von zwei Temperatursensoren Temperaturen zu messen. Der DS18B20 Sensor misst analoge Temperaturen gibt die Messwerte digital über ein sogenanntes 1-wire-Interface (seriell) aus. Der DS1820 hat den Vorteil, dass man sehr einfach bis zu zehn dieser Sensoren an einen Raspberry Pi anschließen kann. Dazu werden die Sensoren einfach parallel geschaltet.

5.1 Anschlüsse des DS1820 und Verkabelung mit dem Raspberry Pi

Abb. 4 zeigt die Beschaltung des DS1820. Der Widerstand mit 4,7 Kilo-Ohm ist ein sogenannter Pull-Up-Widerstand und sorgt dafür, dass an Anschluss DQ eine definierte Spannung anliegt.

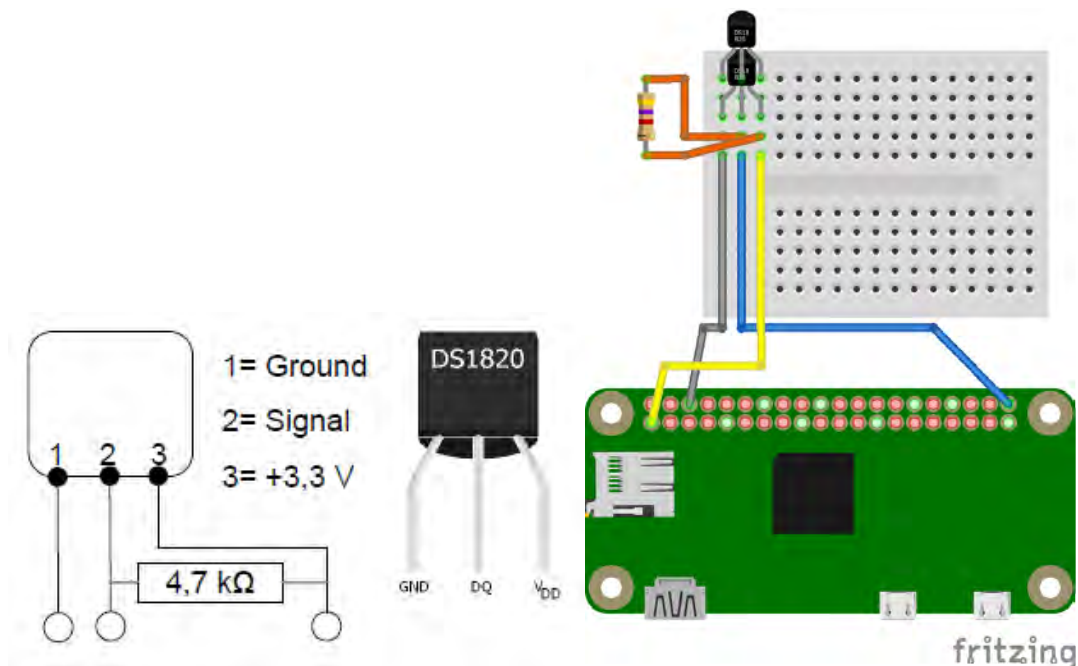


Abb. 4: Beschaltung des Temperatursensors DS1820, zwei parallel auf einem kleinen Breadboard

5.2 Verkabelung des DS1820 mit dem Raspberry Pi

Den Sensor DS1820 stecken Sie nun auf das Breadboard, möglichst genau so wie in Abb 8. Achtung: Fläche Seite mit der Beschriftung zeigt zu den Kabeln, falsch herum zerstört den Sensor)

- VDD Anschluss des Sensors zu Pin 1 (+3,3 V)
- DQ-Anschluss des Sensors zu Pin 40
- GND-Anschluss des Sensors zu Pin 6 (GND)

Erklärung: Die Abrundung des Sensors muss von Ihnen weg schauen und die Schrift zu Ihnen hin, siehe Abb. 8

5.3 Vorbereitung der Temperaturmessung und erster manueller Test

Schritt 1: Konfigurationsdatei config.txt im Verzeichnis /boot mit nano öffnen:

```
sudo nano/boot/config.txt
```

Schritt 2: Folgenden Inhalt unten in config.txt hinzufügen:

```
dtoverlay=w1-gpio,gpiopin=21
```

Erklärung:

Für den verbunden Pin muss die GPIO-Nummer angegeben werden.

Beispiel: Verbunden mit Pin 40 (=GPIO21)

```
dtoverlay=w1-gpio,gpiopin=21
```

Eintrag muss auskommentiert werden, wenn GPIO anderweitig (z. B. für LED Steuerung) verwendet werden soll.

Auskommentieren erfolgt dann mit „#“ also

```
#dtoverlay=w1-gpio,gpiopin=21
```

Schritt 3: System rebooten (damit die Änderungen gültig werden) mit:

```
sudo reboot
```

(Der folgende Schritt kann weggelassen werden und war für ältere Versionen notwendig)

Schritt 3a: Folgende Befehle in die Konsole eingeben und bestätigen:

```
sudo modprobe w1-gpio pullup=1
```

```
sudo modprobe w1-therm
```

(Achtung 1 in w1 ist eine „eins“ und kein „L“)

Schritt 4: Gehen Sie mit cd ins Verzeichnis

```
cd /sys/bus/w1/devices/
```

und geben sie den Befehl

```
dir
```

ein, um eine Übersicht die Temperatursensoren zu erhalten.

Für jeden Sensor wird automatisch ein Verzeichnis angelegt. Es sollten bei zwei Sensoren drei Verzeichnisse vorhanden sein, die so ähnlich aussehen wie

```
28-00000YYYYYYY
```

```
28-00000XXXXXXX
```

```
w1_bus_master1
```

Schritt 5: Gehen Sie in das Verzeichnis, das so heißt wie die Zahlenkolonne, z.B.

```
cd 28-00000XXXXXXX
```

Schritt 6: Lesen Sie testweise die Temperatur eines Sensors aus mit

```
cat w1_slave
```

Erklärung: Der Befehl cat zeigt den Inhalt einer Datei an.

```
2e 01 4b 46 7f ff 02 10 38 : crc=38 YES
2e 01 4b 46 7f ff 02 10 38 t=18875
```

Abb. 5: Ausgabe eines DS1820

Erklärung:

t=18875 in Abb. 5 muss durch 1000 geteilt werden, um die Temperatur 18,875 °C zu erhalten

„Von Hand“ können Sie nun bereits Temperaturen messen.

5.4 Ein Python- Programm für die Temperaturmessung erstellen

Schritt 1: Datei für das Python-Programm mit *nano* anlegen

```
nano temp.py
```

Schritt 2: Geben Sie den Quellcode ein:

```
#!/usr/bin/python
# coding=utf-8
import os, sys, time
# Funktionen
#Liste anlegen
tempSensorBezeichnung = []
tempSensorAnzahl = 0
tempSensorWert = []

#Sucht alle angeschlossenden DS1820 Sensoren und Speichert die Namen
in einer Liste
def ds1820einlesen():
    global tempSensorBezeichnung, tempSensorAnzahl, programmStatus
    for x in os.listdir("/sys/bus/w1/devices"):
        if (x.split("-")[0] == "28") or (x.split("-")[0] == "10"):
            tempSensorBezeichnung.append(x)
            tempSensorAnzahl = tempSensorAnzahl + 1

#Liest alle Sensoren aus der Liste aus
def ds1820auslesen():
    global tempSensorBezeichnung, tempSensorAnzahl, tempSensorWert,
programmStatus
    x = 0
    while x < tempSensorAnzahl:
        dateiName = "/sys/bus/w1/devices/" + tempSensorBezeichnung[x] +
"/w1_slave"
        file = open(dateiName)
        filecontent = file.read()
        file.close()
        stringvalue = filecontent.split("\n")[1].split(" ")[9]
        sensorwert = float(stringvalue[2:]) / 1000
        temperatur = '%6.2f' % sensorwert
        tempSensorWert.insert(x,temperatur)
        x = x + 1

#Programmablauf
ds1820einlesen()
ds1820auslesen()
x = 0
print "Sensorbezeichnung und Temperaturwert:"
```

```
while x < tempSensorAnzahl:  
    print tempSensorBezeichnung[x] , " " , tempSensorWert[x] , " °C"  
    x = x + 1
```

Speichern Sie die Dateien mit *Strg+o*, schließen Sie *nano* mit *Strg+x*.

Schritt 3: Geben Sie Ihrem Programm Ausführungsrechte:

```
sudo chmod 777 temp.py
```

Schritt 4: Führen Sie das Temperaturmessprogramm aus mit dem Befehl

```
./temp.py
```

Dritte Mission geschafft!

6 Vierte Mission: Raspberry Pi Kamera nutzen

Das Ziel ist es, mit der Raspberry Pi Kamera und einem Python Programm Fotos zu schießen und diese ins Internet hochzuladen.

6.1 Raspberry Pi Kamera anschließen und vorbereiten



Abb. 6: Die Raspberry Pi Kamera am Raspberry Pi Zero

Die Kamera wird mit Flachbandkabel an die CSI-Schnittstelle angeschlossen. Diese befindet sich beim Raspberry Zero an einem Ende der Platine (Abb. 6). Zuerst muss die CSI-Schnittstelle geöffnet werden. Das Flachbandkabel gehört dann mit den Kontakten Richtung Platine (Raspberry Zero) in die Schnittstelle.

Dem Raspberry Pi muss mitgeteilt werden, dass eine Kamera verwendet werden soll:

Schritt 1: Rufen Sie das Raspberry Pi Einstellungsprogramm auf mit

```
sudo raspi-config
```

Schritt 2: Gehen Sie zu den Punkt 5 Interfacing Options und bestätigen Sie.

Schritt 3: Wählen Sie P1 Camera aus und Schalten Sie auf „enable“.

6.2 Python Script: Foto schießen und auf den Webserver hochladen

Mit der Paketverwaltung *apt* installieren Sie ein zusätzliche Programme, um die Kamera mittels Python nutzen zu können (*picamera...*) bzw. Anfragen an das Internet stellen zu können (*requests...*).

Schritt 1: Geben Sie folgende Befehle ein

```
sudo apt-get install python-picamera python3-picamera
```

```
sudo apt-get install python-requests
```

Schritt 2: Erstellen Sie mit nano ein Programm mit dem folgenden Befehl

```
nano picam.py
```

Schritt 3: Tragen Sie den folgenden Inhalt ein und Speichern Sie mit strg + o

```
#!/usr/bin/python
#coding utf-8
import picamera
import time
import requests
# Voreinstellungen Bildgroesse:
WIDTH=640
HEIGHT=480
# Kamera initialisieren:
camera = picamera.PiCamera()
camera.resolution = (WIDTH,HEIGHT)
# Dateiname fuer Bild:
localpicname = 'picam.jpg'
# 2 Sek. Pause, damit Kamera alles einstellen kann:
time.sleep(2)
# Ein Bild aufnehmen:
camera.capture(localpicname)
camera.close()
#Bild hochladen:

url = 'https://cbrell.de/raspilab/lehrer/picamWS.php'
files = {'file': open(localpicname, 'rb')}
r = requests.post(url, files=files)
# Zur Kontrolle Rueckgabe ausgeben
print(r)
```

Schritt 4: Geben Sie dem Programm Ausführungsrechte.

```
chmod 777 picam.py
```

Probieren Sie Ihr Skript aus mit

```
./picam.py
```

Die hochgeladenen Bilder können Angesehen und heruntergeladen werden unterfolgender Webadresse:

<http://cbrell.de/raspilab/lehrer/ADM>

Vierte Mission geschafft!

7 Schlussbemerkung

Wenn Sie bis hierhin durchgehalten haben: Prima!

Sie können nun Lichtsignale geben, Temperaturen messen und eine Internet-Überwachungskamera bauen.

Damit können Sie vielleicht zusätzliche Impulse in Ihren Unterricht bringen:

- Schüler bauen eine Wetterstation
- Schüler machen Zeitrafferaufnahmen von wachsenden Pflanzen
- Schüler bauen einen „Lichtwecker“
- ...

Alle weiteren Ideen (Feuchtigkeitssmesser, kleine Roboter, Soundaufnahmen ...) lassen sich genauso leicht verwirklichen – sie werden zu fast jeder Fragestellung geeignete Tutorials im Internet finden.

Unser Tipp: Probieren Sie es mit Ihren Schülern aus.