

RaspiLab - Eine Wetterstation in sieben Missionen

Anleitung

3. Auflage

Stand: 11.01.2019

Autoren:

André Delbos

Prof. Dr. Claus Brell

Vorwort zur 3. Auflage

Sie halten die dritte Auflage der Anleitung zu den Raspberry Pi Schülerworkshops in den Händen. Seit 2016 werden die Schülerworkshops am Fachbereich Wirtschaftswissenschaften, Fachgruppe Wirtschaftsinformatik, zusammen mit kooperierenden Gymnasien durchgeführt.

Ziel der Schülerworkshops ist es, jungen Menschen Spaß am Umgang mit Technik und am Programmieren zu vermitteln. Dabei soll auch aufgezeigt werden, was Wirtschaftsinformatik alles sein kann und was es bedeutet, Wirtschaftsinformatik an der Hochschule Niederrhein zu studieren. Schüler als zukünftige Studierende zu gewinnen, ist dabei ein Sekundärziel.

Die Unterlage kann interessierten Lehrern und Lehrerinnen als Handreichung dienen, um selber mit den Schülern weitergehende Experimente mit den Schülern zu machen.

Die Anleitung ist bewusst knapp gehalten. Sie ist weder ein wissenschaftliches Werk, noch ersetzt sie die Lehrkraft oder ein Recherchieren in Literatur oder Internet. Sie soll den Schülerinnen und Schülern als Handlungsanweisung im Workshop dienen.

Die Anleitung ist auf einen dreistündigen Workshop ausgelegt und grundsätzlich für Schüler ab 14 Jahren geeignet. Zielgruppe sind Schüler in der 11. Klasse an Gymnasien.

Die 3. Auflage fokussiert stark auf das Umweltdatenmessnetzwerk, das die Hochschule mit kooperierenden Schulen in Mönchengladbach aufbauen will. Die Schülerinnen und Schüler sollen im Workshop die Grundlagen erlernen, um am Aufbau des Messdatennetzwerkes mitarbeiten zu können.

Mönchengladbach, Januar 2019

André Delbos

Claus Brell

Inhaltsverzeichnis

1	Voraussetzungen und Grundeinstellungen des Raspberry Pi	4
1.1	Benötigte Materialien	4
1.2	Grundeinstellungen des Raspberry Pi	4
1.3	Sprache	4
1.4	Tastatur Layout einstellen	4
1.5	WLAN-Einstellungen	5
2	Raspberry Pi Pin Belegung & Breadboard - Erklärungen	6
2.1	Raspberry Pi Pin Belegung	6
2.2	Breadboard	7
3	Erste Mission: Den Raspberry Pi und Raspbian kennen lernen	8
4	Zweite Mission: LED-Ampel steuern	11
4.1	Beschaltung einer LED	11
4.2	Programmierung der Ampel	14
5	Dritte Mission: Temperaturen messen mit dem DS1820 Sensor	17
5.1	Beschaltung	17
5.2	Verkabelung des DS1820 mit dem Raspberry Pi	18
5.3	Vorbereiten des Raspberry Pi für die Temperaturmessung	19
5.4	Das Programm für die Temperaturmessung erstellen	21
6	Vierte Mission: Luftfeuchte messen mit dem DHT22 Sensor	22
6.1	Beschaltung	22
6.2	Verkabelung des DHT22 mit dem Raspberry Pi	23
6.3	Vorbereiten des Raspberry Pi für die Luftfeuchtemessung	24
6.4	Das Programm für die Feuchtemessung erstellen	25
7	Fünfte Mission: Messdaten mit der LED-Ampel anzeigen	26
7.1	DS1820 Sensordaten mit der LED-Ampel anzeigen (tempampel.py)	26
7.2	DHT22 Sensordaten mit der LED-Ampel anzeigen (feuchtampel.py)	27
8	Sechste Mission: Druck messen mit dem BMP280	28
8.1	Schaltplan	28
8.2	Breadboard Verkabelung	29
8.3	Vorbereiten des Raspberry Pi für die Druckmessung	30
8.4	Das Programm für die Druckmessung erstellen	30
9	Siebte Mission: Messdaten ins Internet übertragen	32
9.1	Das Programm für die Übertragung der Daten an den Webservice erstellen	32

1 Voraussetzungen und Grundeinstellungen des Raspberry Pi

1.1 Benötigte Materialien

Um die Missionen im Workshop durchführen zu können, benötigen Sie folgenden Materialien:

- Raspberry Pi
- Breadboard
- Mehrere Jumperkabel
- 3 verschiedene LEDs (Anzeigen und "Ampel")
- DS1820 Sensor (Temperaturmessung)
- DHT22 Sensor (Luftfeuchtemessung)
- BMP280 Sensor (Luftdruckmessung)
- Widerstände (zur Beschaltung der LED und der Sensoren)
 - o 2 mal 4,7kΩ (Kilo-Ohm, für DS1820 und DHT22)
 - o 3 mal 330 Ω (Ohm, Schutzwiderstand für LED)
- Tastatur und Monitor
- WLAN-Zugang

1.2 Grundeinstellungen des Raspberry Pi

Die Einstellungen sind für Raspberry Pi 3 Model B vorgesehen.

Wenn Sie Ihren eigenen Raspberry Pi mit in den Workshop bringen, sollte der Raspberry schon die folgenden Einstellungen haben.

Wenn das der Fall ist, können Sie die Punkte 1.3 bis 1.5 überspringen.

1.3 Sprache

Schritt 1: Um die Sprache auf Deutsch umzustellen gehen Sie in das Konfigurationsmenü mit dem Befehl

```
sudo raspi-config
```

Schritt 2: Wählen Sie den Unterpunkt *Localisation Option* aus.

Schritt 3: Nun wählen sie den ersten Unterpunkt *Locale* aus.

Schritt 4: Wählen Sie die Einstellung *de_DE ISO-8859-1* aus und bestätigen Sie.

1.4 Tastatur Layout einstellen

Schritt 1: Um die Tastatur von einer Englischen zu einer Deutschen umzustellen gehen Sie in das Konfigurationsmenü mit dem Befehl:

```
sudo raspi-config
```

Schritt 2: Wählen Sie den Unterpunkt *Localisation Option* aus.

Schritt 3: Nun wählen Sie den Unterpunkt *Change Keyboard Layout* aus.

Schritt 4: Wählen Sie dann unter *Other* -> *German* -> *default* aus und bestätigen Sie.

1.5 WLAN-Einstellungen

Schritt 1: Gehen Sie in das Konfigurationsmenü mit dem Befehl

```
sudo raspi-config
```

Schritt 2 : Wählen Sie den zweiten Unterpunkt *Network Options* aus.

Schritt 3: Wählen Sie den zweiten Unterpunkt *Wi-fi* aus.

Schritt 4: Geben sie für die SSID *raspilab* ein,
geben Sie für das Passwort *raspilab* ein und bestätigen Sie.

2 Raspberry Pi Pin Belegung & Breadboard - Erklärungen

2.1 Raspberry Pi Pin Belegung

Der Raspberry Pi hat verfügt über 40 Anschlüsse (Pins), die verschiedene Funktionen haben, z.B. 3,3 und 5 Volt liefern oder schaltbare Ein- und Ausgänge (GPIO) sowie Erdung (GND). Diese Pins haben feste Positionen. Die Lese Richtung geht von der SD-Karte auf dem Raspberry aus (Abb. 1). Der Linke Pin aus Richtung der SD-Karte ist Pin 1 und der rechts daneben ist Pin 2.

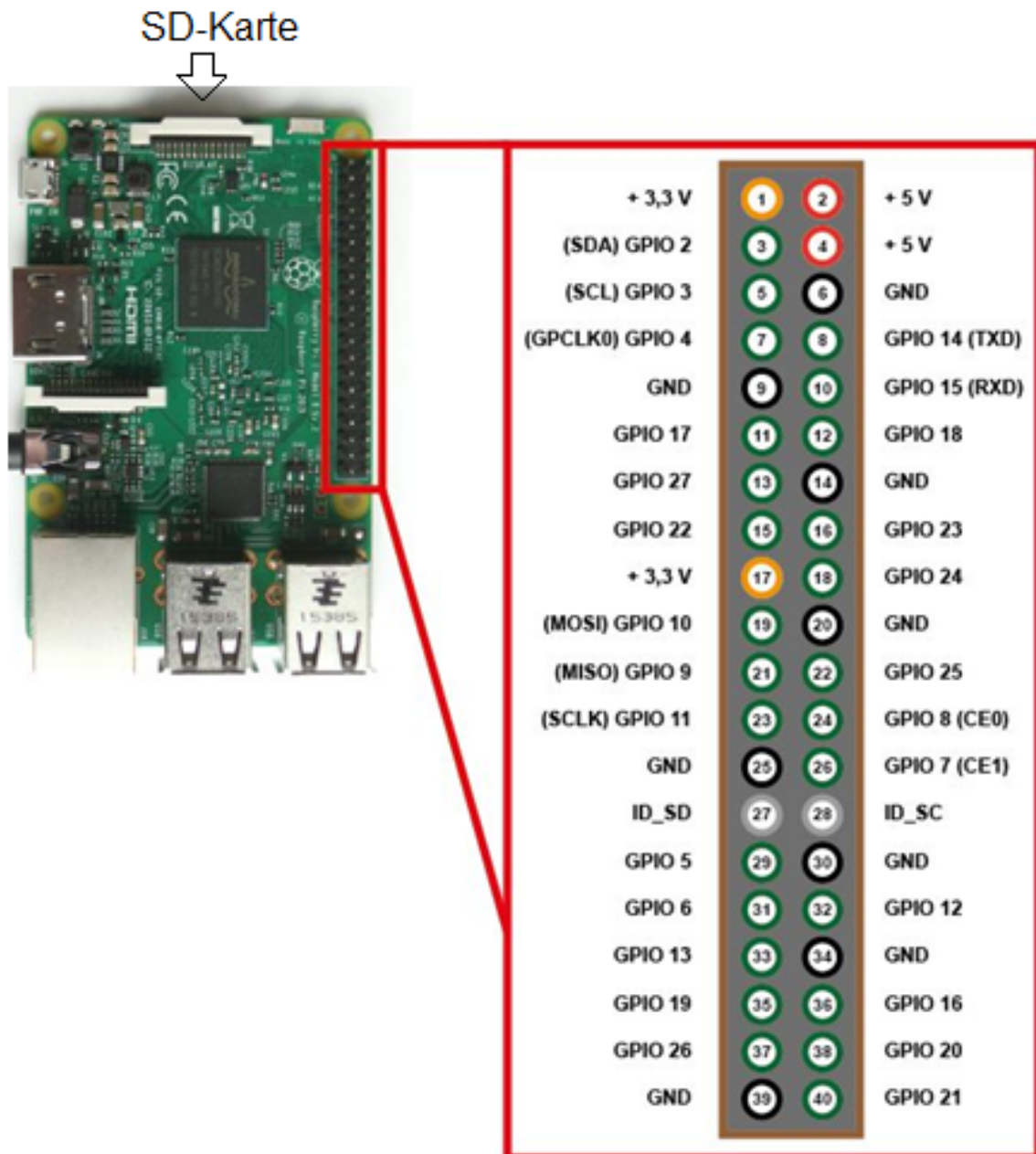


Abb. 1: Raspberry Pi Pin Belegung

2.2 Breadboard

Ein Breadboard ist eine Steckplatine, mit der schnelle und einfache Verkabelungen ohne Löten möglich sind. Die Steckplätze des Breadboard werden mit Zahlen und Buchstaben gekennzeichnet (Abb. 2). Steckplätze auf der gleichen Nummer sind miteinander verbunden. Es gibt auf jeder Seite zwei Außenleisten, die mit plus (+) und minus (-) markiert sind. Diese sind mit einer roten (+) und blauen (-) Linie gekennzeichnet.:

Beispiel: Eine Verbindung besteht, wenn zwischen A1 – E1 oder F1 – J1 Steckplätze belegt sind. Mittels Jumperkabel oder Widerständen können Brücken erstellt werden.

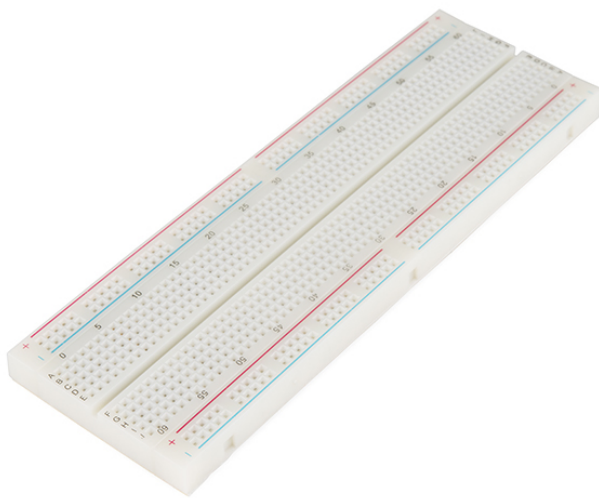


Abb. 2: Breadboard

Erklärung: Diese Außenleisten werden dafür verwendet um Stromversorgung oder Erdung zu sammeln.

3 Erste Mission: Den Raspberry Pi und Raspbian kennen lernen

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

Ein Computer braucht ein Betriebssystem. Raspbian basiert auf dem Linux-Derivat Debian und ist ein kostenloses Betriebssystem für den Raspberry Pi (vergleichbar zu Windows auf einem PC).

Mit dem Raspberry Pi und dem Betriebssystem Raspbian wollen wir die sieben Missionen des Workshops bewältigen.

a) 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.

b) Linux Befehle nutzen - ls und cd. Wir werden ausschließlich mit der Konsole arbeiten, es gibt keine Maus. 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 change directory) 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*.

d) Einen Dateiinhalt 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 beispieldtext.txt anzeigen mit dem Befehl ls, löschen Sie sie mit der Befehlsfolge

```
rm beispieldtext.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 sechs Missionen werden Sie Schritt für Schritt eine Wetterstation aufbauen. Alle Bauteile gelangen nach und nach auf das Breadboard und werden mit den bunten Kabeln (Jumperkabel) an den Raspberry Pi angeschlossen. Am Ende nach der siebten Mission wird Ihr Breadboard ähnlich wie in Abb. 3 und Abb. 4 aussehen.

6 Vierte Mission: Luftfeuchte messen mit dem DHT22 Sensor

Der DHT22 Sensor misst die Luftfeuchtigkeit (und auch die Temperatur) in der Umgebung.

6.1 Beschaltung

Abb. 9 zeigt die Beschaltung des DHT22. Der Widerstand mit 4,7 Kilo-Ohm ist ein sogenannter Pull-Up-Widerstand und sorgt dafür, dass Anschluss Signal eine definierte Spannung hat.

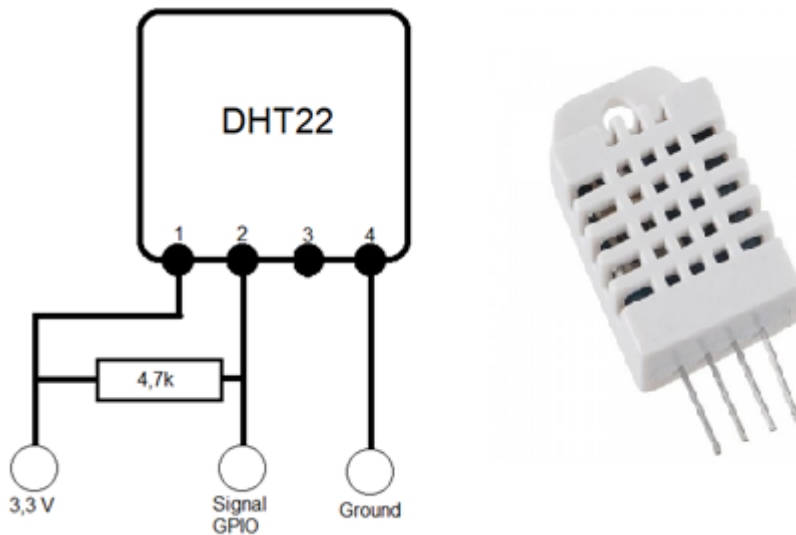


Abb. 9: Beschaltung des Feuchtesensors DHT22

6.2 Verkabelung des DHT22 mit dem Raspberry Pi

Sensor in J21, J22, J23 und J24

Widerstand G22 und G17

Jumperkabel F21 in rote (+) Außenleiste

Jumperkabel F17 in rote (+) Außenleiste

Jumperkabel rote (+) Außenleiste in Pin 1 (3,3V)

Jumperkabel F22 in Pin 38 (GPIO20)

Jumperkabel F24 in Pin 34 (Ground)

Erklärung: Der Kopf des Sensors schaut in Richtung F

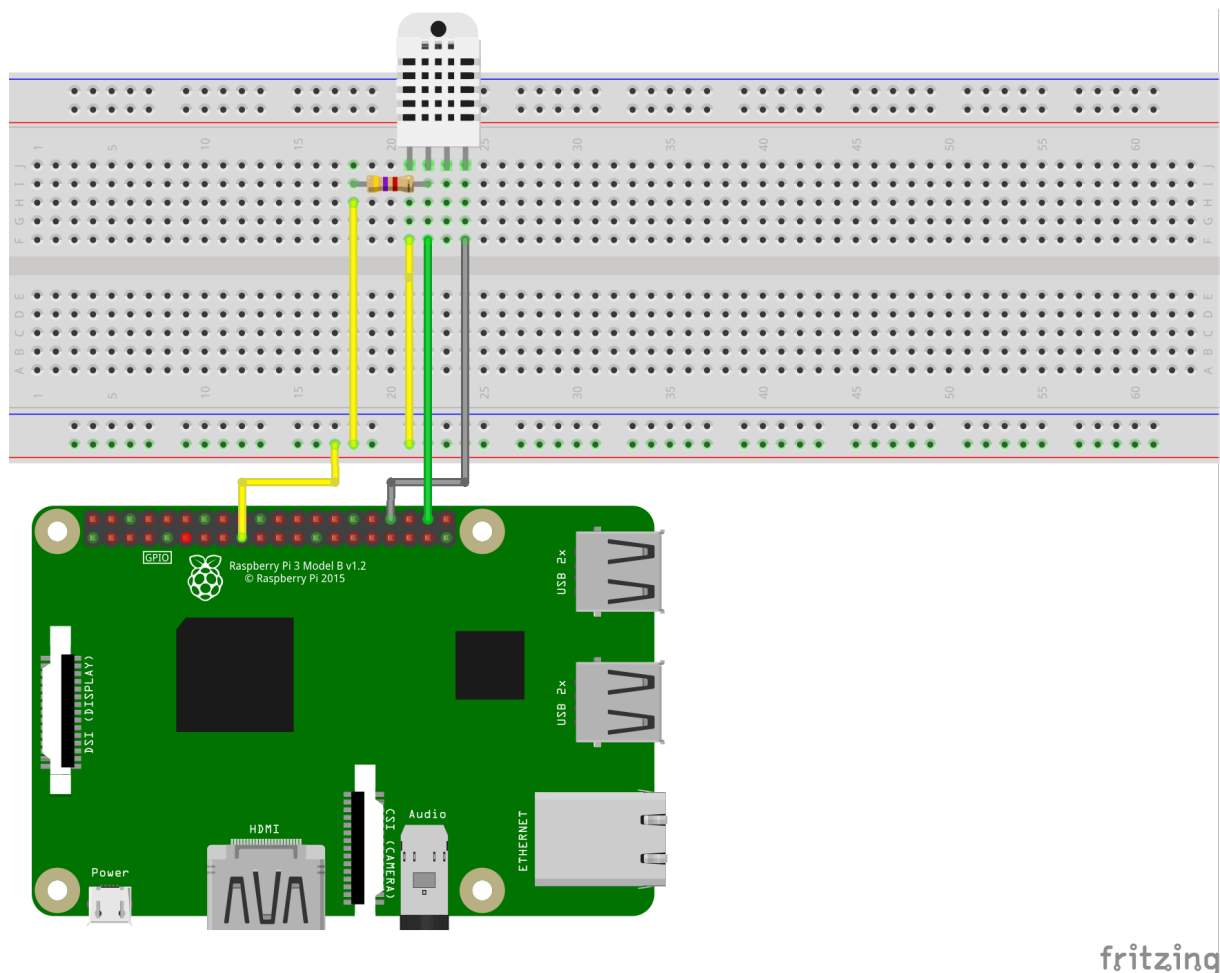


Abb. 10: Verkabelung des Raspberry Pi mit dem Sensor DHT22

6.3 Vorbereiten des Raspberry Pi für die Luftfeuchtemessung

Schritt 1: Der Sensor DHT22 benötigt seine eigenen Python-Pakete. Um die Pakete einzubinden zu können, geben sie folgenden Befehl ein:

```
sudo apt-get install build-essential python-dev python-openssl git
```

Schritt 2: Laden Sie die Adafruit Bibliothek, die den Sensor DHT22 unterstützt:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
```

und wechseln Sie danach in das Verzeichnis mit

```
cd Adafruit_Python_DHT
```

Schritt 3: Installieren Sie die heruntergeladene Bibliothek

```
sudo python setup.py install
```

Schritt 3: Starten Sie den Raspberry Pi neu mittels des Befehls

```
sudo reboot
```

Schritt 4: Nun können Sie testen ob Sensor funktioniert. Gehen Sie in den Ordner Adafruit_Python_DHT mit dem Befehl:

```
cd Adafruit_Python_DHT
```

Nun in das Verzeichnis examples

```
cd examples
```

Testen Sie mit

```
sudo python AdafruitDHT.py 22 20
```

Die Ausgabe müsste so ähnlich wie folgt aussehen:

```
Temp=22.4* Humidity=34.8%
```

Erklärung: Die 22 bei der Eingabe steht für die Sensornummer. Die 20 steht für die GPIO Nummer (nicht für den belegten Pin auf Raspberry Pi).

6.4 Das Programm für die Feuchtemessung erstellen

Schritt 1: Erstellen Sie mittels den folgenden Befehl die Datei für den Quelltext:

```
nano feucht.py
```

Schritt 2: Fügen Sie den Inhalt des Quellcodes in den Quelltext ein.

```
#!/usr/bin/python
import Adafruit_DHT
# Messdaten holen
# (22 steht fuer die Sensornummer
# und die 20 fuer die GPIO Nummer)
luftfeuchtigkeit, temperatur = Adafruit_DHT.read_retry(22, 20)
#Messdaten ausgeben und auf eine Stellenrunden
print "Luftfeuchtigkeit ",round(luftfeuchtigkeit) , "%"
print " Temperatur ", round(temperatur) , " Grad"
```

Schritt 3: Geben Sie dem Programm Ausführungsrechte mit

```
sudo chmod 777 feucht.py
```

Schritt 4: Testen Sie Ihr Script mit

```
./feucht.py
```

Vierte Mission geschafft!

7 Fünfte Mission: Messdaten mit der LED-Ampel anzeigen

Ziel dieser Mission ist es, die Ampel als Anzeige für die Sensoren zu verwenden. Dazu verwenden Sie Ihre bis jetzt erreichten Ergebnisse und kombinieren sie. Der Weg ist der gleiche wie in den vorigen Missionen:

1. Erzeugen Sie ein Programm mit nano (für Temperaturen *nano tempampel.py* und für Feuchtwerte *nano feuchtampel.py*)
2. Geben Sie dem Programm Ausführungsrechte mit `chmod 777`
3. Führen Sie das Programm aus

Sie finden im Folgenden die Quellcodes für die Ansteuerung der Ampel mit dem Temperatursensor DS1820 und dem Feuchtesensor DHT22.

7.1 DS1820 Sensordaten mit der LED-Ampel anzeigen (tempampel.py)

```
#!/usr/bin/python
import time
import RPi.GPIO as GPIO
i=0
#Die Devicesnummer muss angepasst werden
while i<=10:
    i=i+1
    #GPIO Pins fuer LEDs festlegen
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(11, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)
    GPIO.setup(15, GPIO.OUT)
    #Messdatenholen
    time.sleep(1)
    tempfile = open("/sys/bus/w1/devices/28-0000074b57ed/w1_slave")
    inhalt = tempfile.read()
    tempfile.close()
    tempdata = inhalt.split("\n")[1].split(" ")[9]
    temperatur = float(tempdata[2:])
    temperatur = temperatur/1000
    #Temperatur unter 20 Grad ROTE LED leuchtet
    iftemperatur< 20:
        GPIO.output(11, GPIO.LOW)
        GPIO.output(13, GPIO.LOW)
        GPIO.output(15, GPIO.HIGH)
    #Temperatur ueber 20 Grad und unter 25 GELBE LED leuchtet
    eliftemperatur> 20 and temperatur< 25:
        GPIO.output(11, GPIO.LOW)
        GPIO.output(13, GPIO.HIGH)
        GPIO.output(15, GPIO.LOW)
    #Temperatur ueber 25 Grad ROTE LED leuchtet
    else:
        GPIO.output(11, GPIO.HIGH)
        GPIO.output(13, GPIO.LOW)
        GPIO.output(15, GPIO.LOW)
    time.sleep(3)
    GPIO.cleanup()
```

7.2 DHT22 Sensordaten mit der LED-Ampel anzeigen (feuchtampel.py)

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

i=0

while i<5:
    i=i+1
    #GPIO Pins fuer LEDs festlegen
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(11, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)
    GPIO.setup(15, GPIO.OUT)

    #Messdaten holen (22 steht fuer den Sensornummer und die 20 fuer
    die GPIO-PIN)
    luftfeuchtigkeit, temperatur = Adafruit_DHT.read_retry(22, 20)

    if temperatur < 20:
        GPIO.output(11, GPIO.HIGH)
        GPIO.output(13, GPIO.LOW)
        GPIO.output(15, GPIO.LOW)
    elif temperatur > 20 and temperatur < 25:
        GPIO.output(11, GPIO.LOW)
        GPIO.output(13, GPIO.HIGH)
        GPIO.output(15, GPIO.LOW)
    elif temperatur > 25:
        GPIO.output(11, GPIO.HIGH)
        GPIO.output(13, GPIO.LOW)
        GPIO.output(15, GPIO.LOW)

    time.sleep(3)
    GPIO.cleanup()
```

Erstellen Sie eines der beiden Python-Programme und probieren Sie aus, ob Sie durch anfassen oder durch Anhauchen des Sensors die Ampel zum Umschlagen bringen können.

Fünfte Mission geschafft!