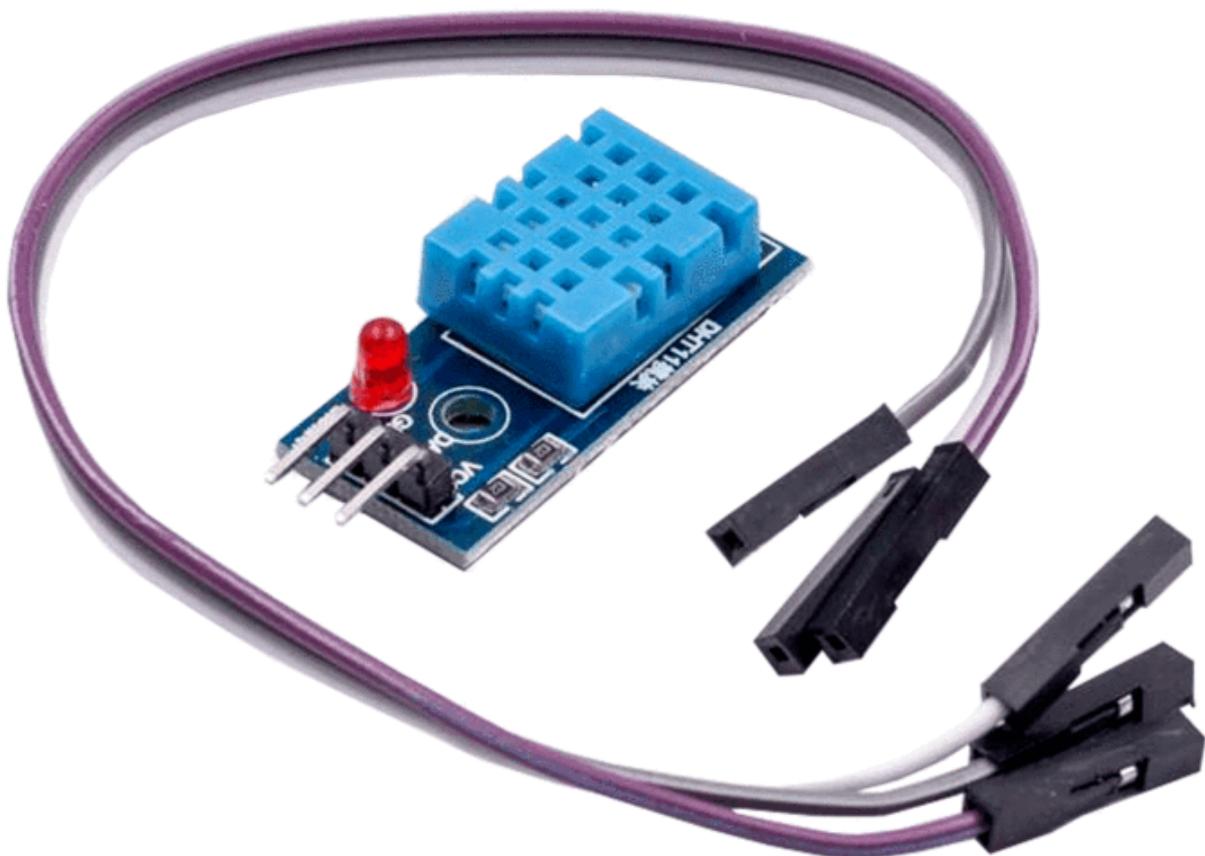


AZ-Delivery

Willkommen!

Vielen Dank, dass sie sich für unser Temperaturmodul vom Typ "DHT11" von AZ-Delivery entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Viel Spaß!



Az-Delivery

Der "DHT-11" ist ein Sensor für die relative Feuchtigkeit/Temperatur, der ein digitales Signal ausgibt. Er verwendet einen kapazitiven Feuchtigkeitssensor und einen Thermistor zur Messung der Umgebungsluft.

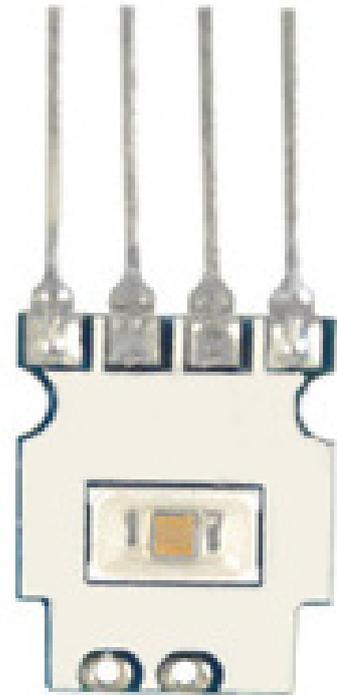
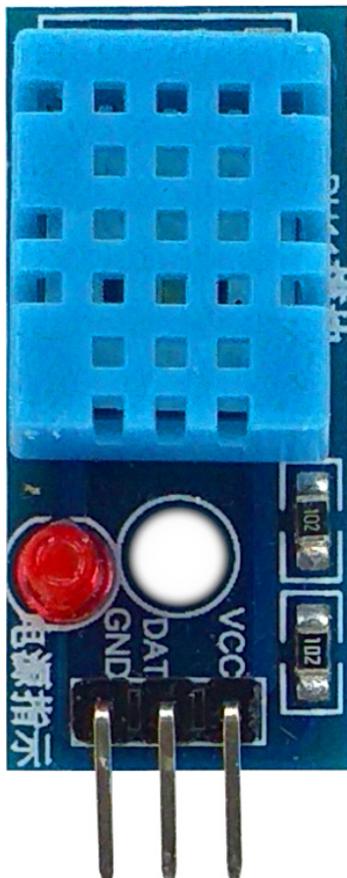
Der Temperaturmessbereich des "DHT11" reicht von 0°C bis +50°C mit einer Genauigkeit von $\pm 2^\circ\text{C}$ und der Feuchtemessbereich von 20% bis 90% mit einer Genauigkeit von $\pm 5\%$.

Technische Daten:

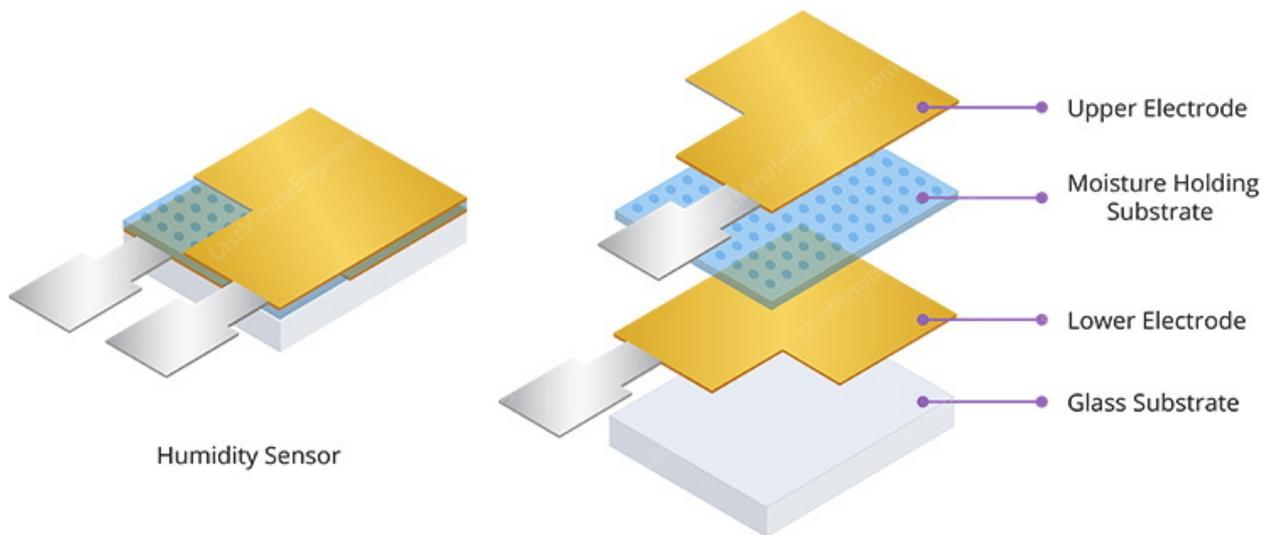
Betriebsspannung:	3 bis 5V
Maximale Betriebsspannung:	2.5mA max.
Feuchtigkeitsbereich:	20% - 90% Abweichung von max. 5%
Temperaturbereich:	0°C - 50°C Abweichung von max. $\pm 2^\circ\text{C}$
Abtastrate:	1Hz (liest jede s)
Dimensionen:	15x32x9mm

Az-Delivery

Im Inneren des Gehäuses, auf der Fühlerseite des DHT11-Sensors, befindet sich ein Feuchtefühler zusammen mit einem NTC-Temperaturfühler (auch Thermistor).



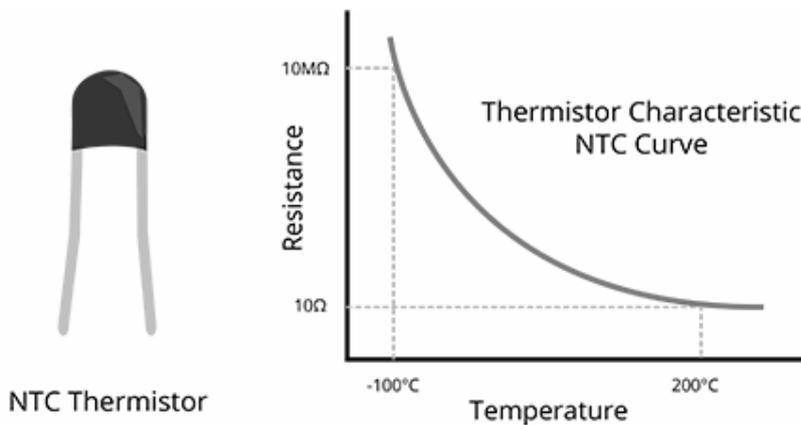
Az-Delivery



Die Feuchtemesskomponente wird zur Messung der Feuchtigkeit verwendet und hat zwei Elektroden mit feuchtigkeitshaltendem Substrat (normalerweise ein Salz oder ein leitfähiges Kunststoffpolymer) zwischen den Elektroden. Die Ionen werden durch das Substrat freigesetzt, wenn Wasserdampf von diesem absorbiert wird, was wiederum die Leitfähigkeit zwischen den Elektroden erhöht. Die Widerstandsänderung zwischen den beiden Elektroden ist proportional zur relativen Luftfeuchtigkeit. Eine höhere relative Luftfeuchtigkeit verringert den Widerstand zwischen den Elektroden, während eine niedrigere relative Luftfeuchtigkeit den Widerstand zwischen den Elektroden erhöht.

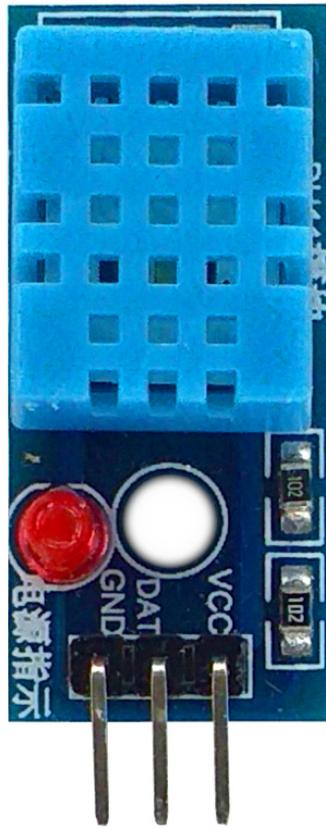
Az-Delivery

Der Temperaturmessteil des Sensors besteht aus einem NTC-Temperaturfühler (Thermistor) zur Temperaturmessung. Ein Thermistor ist ein thermischer Widerstand und dieser Widerstand ändert sich mit der Temperatur. Streng gesehen sind alle Widerstände Thermistoren. Ihr Widerstand ändert sich geringfügig mit der Temperatur, aber die Änderung ist normalerweise klein und schwer zu messen. Thermistoren werden so hergestellt, dass sich der Widerstand mit der Temperatur drastisch ändert, so dass er sich pro Grad Temperatur um 100Ω oder mehr ändern kann. Der Begriff "NTC" bedeutet "Negativer Temperaturkoeffizient", was bedeutet, dass der Widerstand mit der Temperaturerhöhung abnimmt.



Auf der anderen Seite befindet sich eine kleine Leiterplatte mit einem 8-Bit SOIC-14-IC Schaltkreis. Dieser Schaltkreis misst und verarbeitet das Analogsignal mit gespeicherten Kalibrierkoeffizienten, führt eine Analog-Digital-Wandlung durch und gibt ein digitales Signal mit den Daten aus, das Informationen über Temperatur und Feuchte enthält.

Az-Delivery



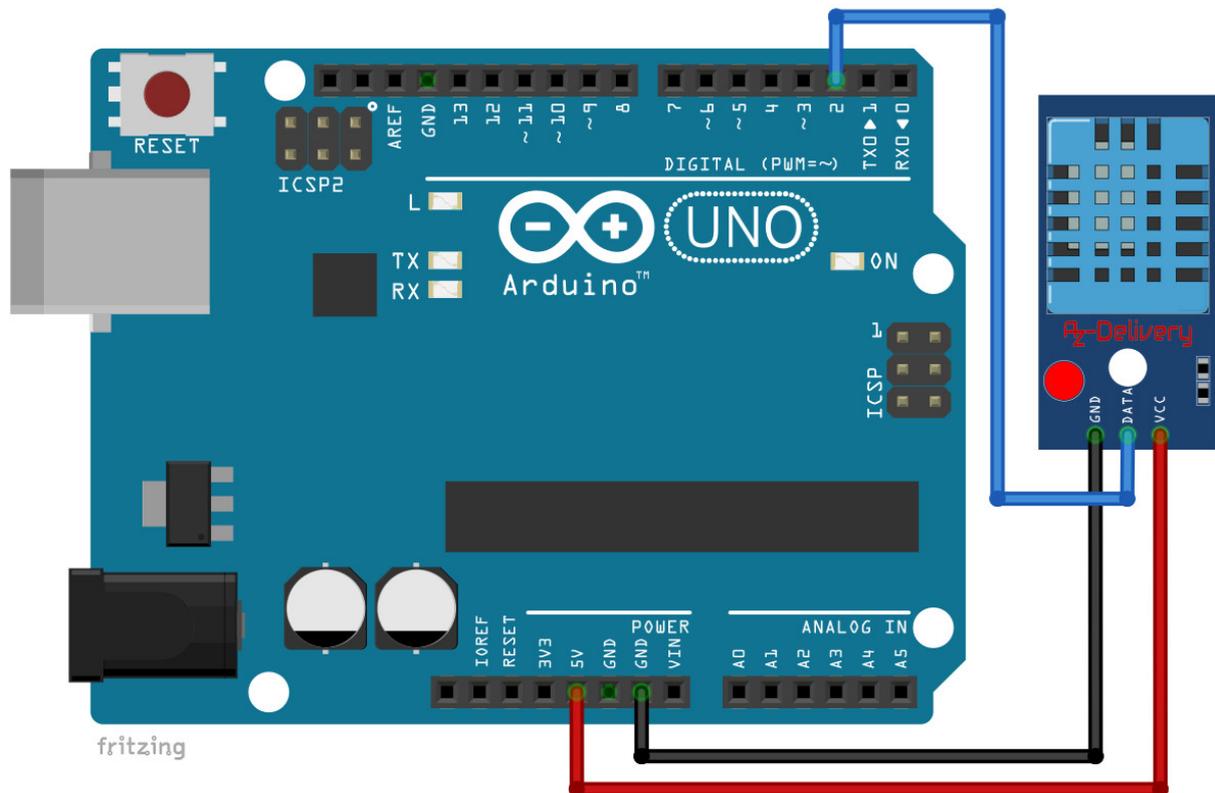
"VCC"-Pin - liefert die Stromversorgung für den Sensor. Obwohl die Versorgungsspannung zwischen 3,3V und 5,5V liegen kann, wird eine 5V-Versorgung empfohlen. Im Falle einer 5V-Versorgung können Sie Kabel mit einer Länge von bis zu 20 Meter verwenden, um den Sensor und den Mikrocontroller zu verbinden. Bei einer Versorgungsspannung von 3,3V darf die Kabellänge jedoch nicht größer als ein Meter sein. Andernfalls führt der Leitungsspannungsabfall zu Messfehlern.

"DATA"-Pin - ist der Datenpin, der die Kommunikation zwischen Sensor und Mikrocontroller ermöglicht

"GND"-Pin - ist der Massepin und sollte mit der gemeinsamen Masse oder 0V (bei Arduino oder Raspberry Pi) verbunden werden.

Verbindung des Moduls mit dem Arduino Uno

Verbinden Sie den Arduino Uno mit dem DHT11, wie unten abgebildet:



DHT11-Pin > Arduino Uno Pin

VCC > 5V

DATA > D2

GND > GND

Roter Draht

Blauer Draht

Schwarzer Draht

Az-Delivery

Arduino Skizzenbeispiel

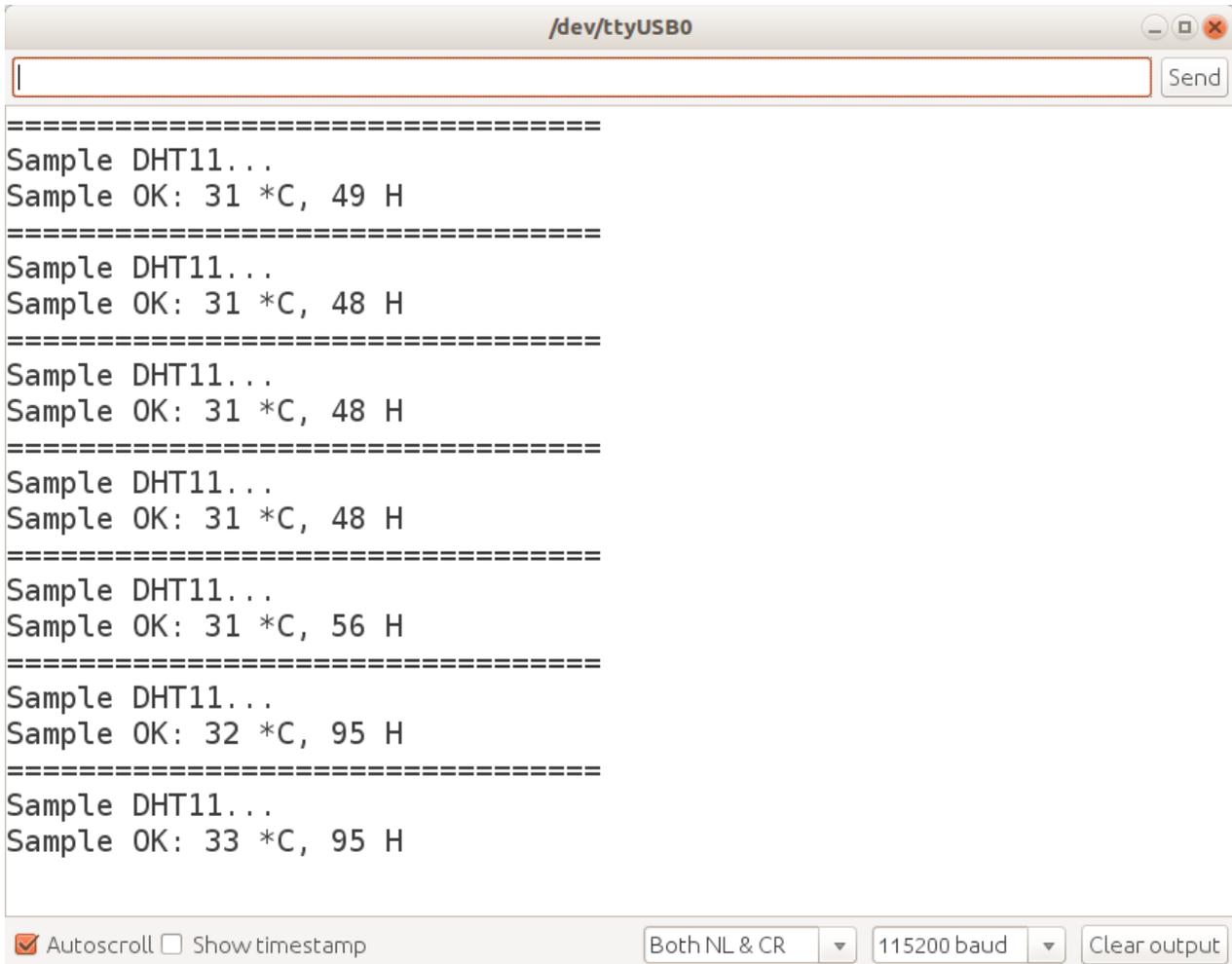
Um eine Skizze für das DHT11-Modul zu schreiben, benötigen wir zunächst die Library. Die einfachste Library, die wir empfehlen können, ist die SimpleDHT-Bibliothek, die unter <https://github.com/winlinvip/SimpleDHT> heruntergeladen werden kann.

Wenn Sie die .ZIP-Datei herunterladen, öffnen Sie den Arduino Uno und gehen auf: *Sketch > Include Library > Fügen Sie die .ZIP Library* und die heruntergeladene .ZIP-Datei hinzu. Danach gehen Sie zu: *File > Examples > SimpleDHT > DHT11Default* und öffnen die Skizze. Das sollte so aussehen:

```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);
void setup() { Serial.begin(115200); }
void loop() {
  Serial.println("=====");
  Serial.println("Sample DHT11...");
  float temperature = 0;
  float humidity = 0;
  int err = SimpleDHTErrSuccess;
  if((err=dht11.read2(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess){
    Serial.print("Read DHT11 failed, err=");
    Serial.println(err);
    delay(2000);
    return;
  }
  Serial.print("Sample OK: ");
  Serial.print((float)temperature);
  Serial.print(" *C, ");
  Serial.print((float)humidity);
  Serial.println(" RH%");
  delay(1500); // DHT11 sampling rate is 1HZ.
}
```

Az-Delivery

Wenn Sie den Serial Monitor (*Tools > Serial Monitor*) öffnen, sollte die Anzeige so aussehen:



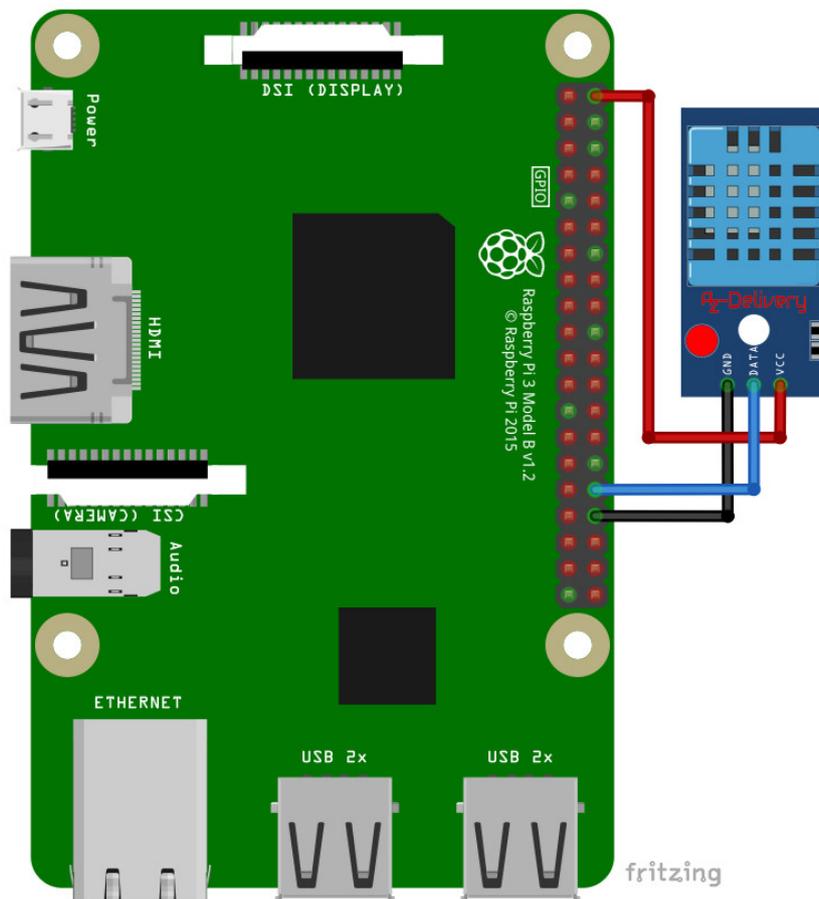
The screenshot shows a Serial Monitor window titled "/dev/ttyUSB0". The window contains a text area with the following output:

```
=====  
Sample DHT11...  
Sample OK: 31 *C, 49 H  
=====  
Sample DHT11...  
Sample OK: 31 *C, 48 H  
=====  
Sample DHT11...  
Sample OK: 31 *C, 48 H  
=====  
Sample DHT11...  
Sample OK: 31 *C, 48 H  
=====  
Sample DHT11...  
Sample OK: 31 *C, 56 H  
=====  
Sample DHT11...  
Sample OK: 32 *C, 95 H  
=====  
Sample DHT11...  
Sample OK: 33 *C, 95 H
```

At the bottom of the window, there are several controls: a checked checkbox for "Autoscroll", an unchecked checkbox for "Show timestamp", a dropdown menu set to "Both NL & CR", a dropdown menu set to "115200 baud", and a "Clear output" button.

Verbindung des Moduls mit dem Raspberry Pi

Verbinden Sie den Raspberry Pi mit dem DHT11, wie unten abgebildet:



DHT11 Pin > Raspberry Pi Pin

VCC > 5V [PIN 2]

DATA > GPIO12 [PIN 32]

GND > GND [PIN 30]

Roter Draht

Blauer Draht

Schwarzer Draht

Az-Delivery

Skriptbeispiel

Bevor wir anfangen, Skripte für den DHT11 zu erstellen, müssen wir zuerst eine Library installieren. Die Library, die wir verwenden werden, heißt "Adafruit_DHT". Um sie zu installieren, müssen wir zuerst sicherstellen, dass Rasbian auf dem neuesten Stand ist. Starten Sie Ihren Raspberry Pi, öffnen Sie das Terminal und führen Sie folgende Befehle aus:

Der erste Befehl updatet das System:

```
sudo apt-get update && sudo apt-get upgrade -y
```

Der zweite Befehl installiert die python2-pip-Anwendung. Wir verwenden diese um die Library zu installieren:

```
sudo apt-get install python3-pip
```

Der dritte Befehl installiert andere Anwendungen, die die pip-App für die Installation der Library benötigt:

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

Danach können wir die notwendige Library installieren. Führen Sie folgenden Befehl aus:

```
sudo pip3 install Adafruit_DHT
```

```
pi@raspberrypi:~ $ sudo pip3 install Adafruit_DHT
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting Adafruit_DHT
  Downloading https://www.piwheels.org/simple/adafruit-dht/Adafruit_DHT-1.4.0-cp35-cp35m-linux_armv7l.whl
Installing collected packages: Adafruit-DHT
Successfully installed Adafruit-DHT-1.4.0
```

Az-Delivery

Nachdem alles installiert wurde, können wir mit dem Schreiben des Skripts anfangen. Öffnen Sie den Texteditor in Rasbian und fügen Sie folgendes Skript ein:

```
import Adafruit_DHT
from time import sleep
sensor = Adafruit_DHT.DHT11
# DHT11 sensor connected to GPIO12.
pin = 12
print("[press ctrl+c to end the script]")
try: # Main program loop
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(sensor,
                                                         pin)

        sleep(1.5)
        if humidity is not None and temperature is not None:
            print("Temp={0:0.1f}*C Humidity={1:0.1f}%"
                  .format(temperature, humidity))
        else:
            print("Failed to get reading. Try again!")

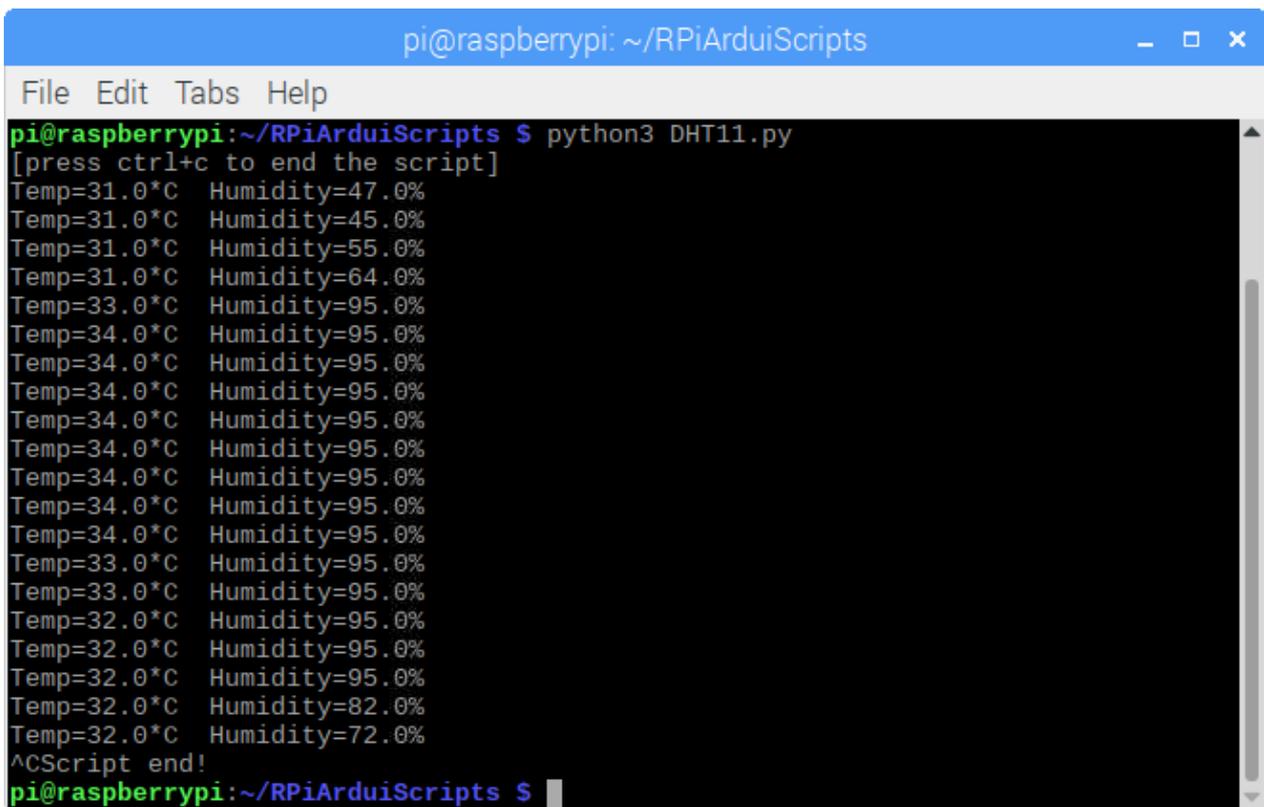
# Scavenging work after the end of the program
except KeyboardInterrupt:
    print("Script end!")
```

Az-Delivery

Speichern Sie das Skript als *"DHT11.py"* und führen Sie es aus. Führen Sie danach folgenden Befehl im Terminal aus:

```
python3 DHT11.py
```

Die Anzeige sollte wie folgt aussehen:



```
pi@raspberrypi: ~/RPiArduiScripts
File Edit Tabs Help
pi@raspberrypi:~/RPiArduiScripts $ python3 DHT11.py
[press ctrl+c to end the script]
Temp=31.0*C Humidity=47.0%
Temp=31.0*C Humidity=45.0%
Temp=31.0*C Humidity=55.0%
Temp=31.0*C Humidity=64.0%
Temp=33.0*C Humidity=95.0%
Temp=34.0*C Humidity=95.0%
Temp=33.0*C Humidity=95.0%
Temp=33.0*C Humidity=95.0%
Temp=32.0*C Humidity=95.0%
Temp=32.0*C Humidity=95.0%
Temp=32.0*C Humidity=95.0%
Temp=32.0*C Humidity=82.0%
Temp=32.0*C Humidity=72.0%
^CScript end!
pi@raspberrypi:~/RPiArduiScripts $
```

Sie haben es geschafft. Sie können jetzt unser Modul nun für Ihre Projekte nutzen.

AZ-Delivery

Jetzt sind Sie dran! Entwickeln Sie Ihre eigenen Projekte und Smart-Home Installationen. Wie Sie das bewerkstelligen können, zeigen wir Ihnen unkompliziert und verständlich auf unserem Blog. Dort bieten wir Ihnen Beispielskripte und Tutorials mit interessanten kleinen Projekten an, um schnell in die Welt der Mikroelektronik einzusteigen. Zusätzlich bietet Ihnen auch das Internet unzählige Möglichkeiten, um sich in Sachen Mikroelektronik weiterzubilden.

Falls Sie nach noch weiteren hochwertigen Produkten für Arduino und Raspberry Pi suchen, sind Sie bei AZ-Delivery Vertriebs GmbH goldrichtig. Wir bieten Ihnen zahlreiche Anwendungsbeispiele, ausführliche Installationsanleitungen, E-Books, Bibliotheken und natürlich die Unterstützung unserer technischen Experten.

<https://az-delivery.de>

Have Fun!

Impressum

<https://az-delivery.de/pages/about-us>