

KY-015 DHT11 TEMPERATURSENSOR

Temperatur und Luftfeuchtigkeitssensor

1. ALLGEMEINE INFORMATIONEN

Sehr geehrte*r Kunde*in,
vielen Dank, dass Sie sich für unser Produkt entschieden haben. Im Folgenden zeigen wir Ihnen, was bei der Inbetriebnahme und der Verwendung zu beachten ist.

Sollten Sie während der Verwendung unerwartet auf Probleme stoßen, so können Sie uns selbstverständlich gerne kontaktieren.

Der DHT11 ist ein preiswerter und unkomplizierter digitaler Temperatur- und Feuchtigkeitssensor, der sich besonders gut für den Einsatz mit Raspberry Pi und Arduino eignet.

2. GRUNDLEGENDE INFORMATIONEN

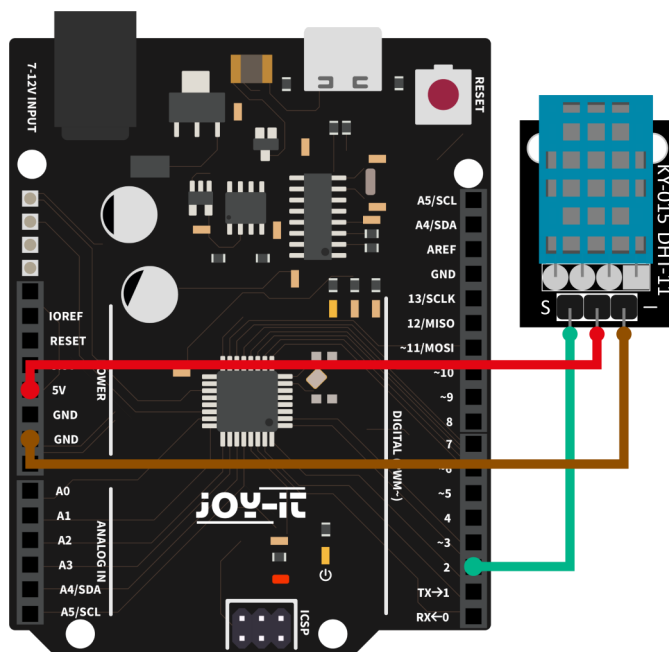
Dieser Sensor ist eine Kombination aus Temperatursensor und Luftfeuchtigkeitssensor, vereint in einer kompakten Bauform. Die Abtastrate der Messungen liegt bei 2 Sekunden. Dieser Sensor ist somit vor allem für Langzeit-Messungen geeignet.

Technische Spezifikationen

| Chipsatz | DHT11 |
|------------------------------------|----------------|
| Kommunikationsprotokoll | 1-Wire |
| Messbereich Temperatur | 0 °C bis 50 °C |
| Messgenauigkeit (Temperatur) | ±2 °C |
| Messbereich Luftfeuchtigkeit | 20-90 %RH |
| Messgenauigkeit (Luftfeuchtigkeit) | ±5 %RH |

3. VERWENDUNG MIT DEM ARDUINO

3.1 Anschluss



| Arduino | DHT11 |
|---------|--------|
| Pin D2 | Signal |
| 5 V | +V |
| GND | GND |

3.2 Codebeispiel

Wir bieten Ihnen für die Verwendung mit dem Arduino ein Codebeispiel an, welches Sie [hier](#) herunterladen können. Dieser Sensor gibt sein Messergebnis nicht als analoges Signal auf einen Ausgangspin aus, sondern kommuniziert diesen digital und kodiert. Zur Ansteuerung dieses Sensormoduls gibt es mehrere Möglichkeiten. Als besonders zugänglich hat sich die [DHT-sensor-library](#) erwiesen, die die Firma [Adafruit](#) unter der [MIT-Lizenz](#) veröffentlicht hat. Sie können sich die Bibliothek in der Arduino IDE direkt hinzufügen.

Dazu klicken Sie auf **Sketch** → **Bibliothek einbinden** → **Bibliotheken verwalten** Dort können Sie in der Suchleiste nach der **DHT sensor library** suchen und diese installieren.

Zum Benutzen des Sensors brauchen Sie nun nur den folgenden modifizierten Code in ihre Arduino IDE kopieren und auf ihren Arduino hochladen

Bevor Sie den Code starten, achten Sie darauf, dass bei **Werkzeuge** → **Board:** das richtige Board und unter **Werkzeuge** → **Port** der richtige COM-Anschluss ausgewählt sind.

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 2      // Digitaler Pin, der mit dem DHT-Sensor verbunden ist

#define DHTTYPE      DHT11      // DHT 11

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup() {
  Serial.begin(9600);
  // Gerät initialisieren.
  dht.begin();
  // Details zum Temperatursensor ausgeben.
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("-----"));
  Serial.println(F("Temperatursensor"));
  Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print (F("Treiber Ver: ")); Serial.println(sensor.version);
  Serial.print (F("Einzigartige ID: ")); Serial.println
(sensor.sensor_id);
  Serial.print (F("Maximaler Wert: ")); Serial.print
(sensor.max_value); Serial.println(F("°C"));
```

```

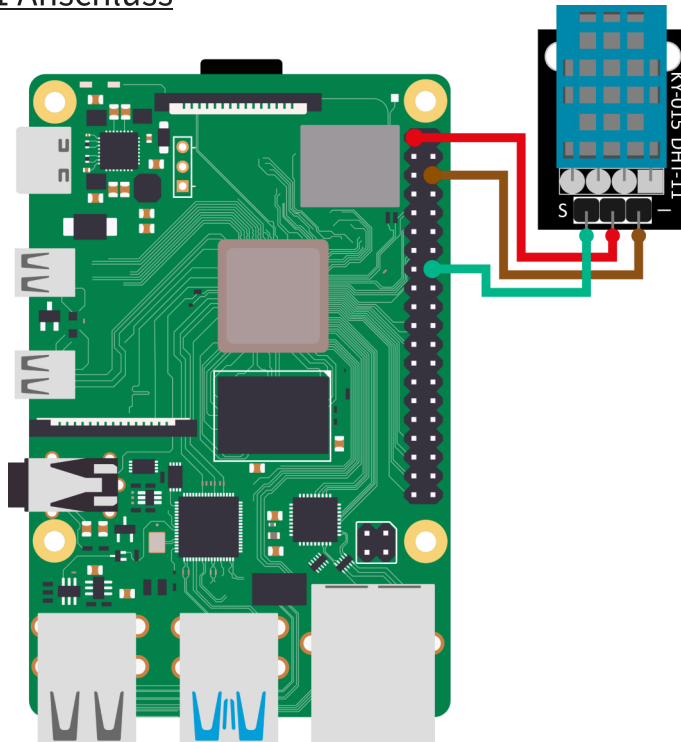
    Serial.print (F("Minimaler Wert:  ")); Serial.print
(sensor.min_value); Serial.println(F("°C"));
    Serial.print (F("Auflösung:  ")); Serial.print(sensor.resolution);
Serial.println(F("°C"));
    Serial.println(F("-----"));
    // Details zum Feuchtigkeitssensor ausgeben.
    dht.humidity().getSensor(&sensor);
    Serial.println(F("Luftfeuchtigkeitssensor"));
    Serial.print (F("Sensor Type:  ")); Serial.println(sensor.name);
    Serial.print (F("Treiber Ver:  ")); Serial.println(sensor.version);
    Serial.print (F("Einzigartige ID:  ")); Serial.println
(sensor.sensor_id);
    Serial.print (F("Maximaler Wert:  ")); Serial.print
(sensor.max_value); Serial.println(F("%"));
    Serial.print (F("Minimaler Wert:  ")); Serial.print
(sensor.min_value); Serial.println(F("%"));
    Serial.print (F("Auflösung:  ")); Serial.print(sensor.resolution);
Serial.println(F("%"));
    Serial.println(F("-----"));
    // Stellen Sie die Verzögerung zwischen den Sensormessungen
    // anhand der Sensordetails ein.
    delayMS = sensor.min_delay / 1000;
}

void loop() {
    // Verzögerung zwischen den Messungen.
    delay(delayMS);
    // Abrufen des Temperaturereignisses und Ausgabe
    // seines Wertes.
    sensors_event_t event;
    dht.temperature().getEvent(&event);
    if (isnan(event.temperature)) {
        Serial.println(F("Fehler beim Ablesen der Temperatur!"));
    }
    else {
        Serial.print(F("Temperatur:  "));
        Serial.print(event.temperature);
        Serial.println(F("°C"));
    }
    // Abrufen des Feuchtigkeitereignisses und Ausgabe
    // seines Wertes.
    dht.humidity().getEvent(&event);
    if (isnan(event.relative_humidity)) {
        Serial.println(F("Fehler beim Ablesen der Luftfeuchtigkeit!"));
    }
    else {
        Serial.print(F("Luftfeuchtigkeit:  "));
        Serial.print(event.relative_humidity);
        Serial.println(F("%"));
    }
}

```

4. VERWENDUNG MIT DEM RASPBERRY PI

4.1 Anschluss



| Raspberry Pi | DHT11 |
|------------------|--------|
| GPIO 23 (Pin 16) | Signal |
| +3,3 V (Pin 1) | +V |
| GND (Pin 6) | GND |

4.2 Codebeispiel

Wir bieten Ihnen für die Verwendung mit dem Raspberry Pi ein Codebeispiel an, welches Sie [hier](#) herunterladen können. Dieses Codebeispiel verwendet die [Adafruit_CircuitPython_DHT](#) Bibliothek von [Adafruit](#) welche unter der [MIT-Lizenz](#) veröffentlicht wurde. Zunächst müssen Sie jedoch die folgenden Befehle ausführen:

```
sudo apt-get update
sudo apt-get install build-essential python3-dev
sudo apt install gpod
sudo apt install python3-pip
```

Erstellen Sie nun die notwendige virtuelle Umgebung für das Projekt mit den folgenden Befehlen:

```
mkdir dein_projekt
```

```
cd dein_projekt
```

```
python -m venv --system-site-packages env
```

```
source env/bin/activate
```

Installieren Sie nun die Bibliothek mit dem folgendem Befehl:

```
pip3 install adafruit-circuitpython-dht
```

Nachdem Sie die Befehle ausgeführt haben, können Sie auch alternativ, wenn Sie sich das Codebeispiel nicht heruntergeladen haben, das folgende identische Codebeispiel in eine neu erstellte Datei kopieren. Dazu müssen Sie auf Ihrem Raspberry Pi folgenden Befehl in die Konsole eingeben:

```
nano DHT11.py
```

Danach brauchen Sie nur noch den folgenden Code in die so eben erstellte Datei einfügen.

```
import time
import board
import adafruit_dht

# Initialisieren Sie den DHT, wobei der Datenpin mit Pin 16
# (GPIO 23) des Raspberry Pi verbunden ist:
dhtDevice = adafruit_dht.DHT11(board.D23)

# Sie können DHT22 use_pulseio=False übergeben, wenn Sie
# pulseio nicht verwenden möchten.
# Dies kann auf einem Linux-Einplatinencomputer wie dem
# Raspberry Pi notwendig sein, aber es wird nicht
# in CircuitPython funktionieren.
# dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)

while True:
    try:
        # Ausgabe der Werte über die serielle Schnittstelle
        temperature_c = dhtDevice.temperature
        temperature_f = temperature_c * (9 / 5) + 32
        humidity = dhtDevice.humidity
        print("Temp: {:.1f} F / {:.1f} C    Luftfeuchtigkeit: {}%
".format(temperature_f, temperature_c, humidity))

    except RuntimeError as error:
        # Fehler passieren ziemlich oft, DHT's sind schwer zu
        # Lesen, einfach weitermachen
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error

    time.sleep(2.0)
```

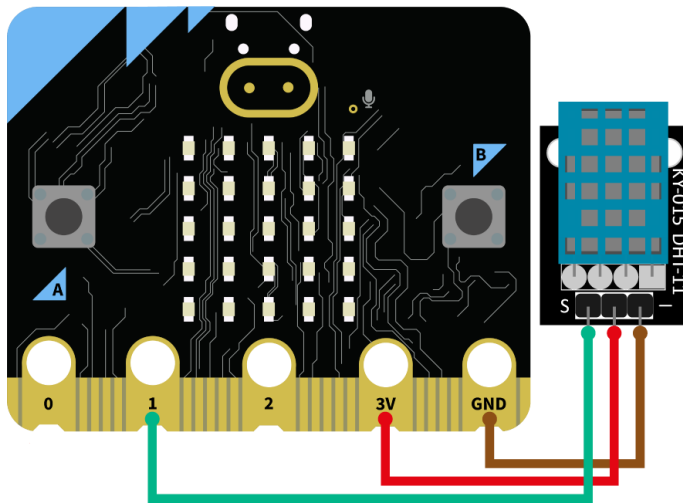
Speichern können Sie die Datei mit **STRG+O** und anschließend mit **STRG+X** schließen. Ausführen können Sie die Datei in der Konsole mit dem Befehl

```
python3 DHT11.py
```

und stoppen können Sie die Datei mit **STRG+C**.

5. VERWENDUNG MIT DEM MICRO:BIT

5.1 Anschluss



| Micro:Bit | DHT11 |
|-----------|--------|
| Pin 1 | Signal |
| +3,3 V | +V |
| GND | GND |

5.2 Codebeispiel

Für das folgende Codebeispiel wird eine zusätzliche Library benötigt. Um diese zu installieren, klicken Sie in Ihrem Make-Code-Sketch auf „Erweiterung hinzufügen“ und suchen Sie hier nach „DHT11“. Installieren Sie hier die [„DHT11_DHT22“ Bibliothek von alankrantas](#).

Übernehmen Sie den folgenden Beispielcode in Ihren Sketch, oder importieren Sie die [.hex-Datei](#).

dauerhaft

Query DHT11 ▾

Data pin P1 ▾

Pin pull up wahr ▾

Serial output falsch ▾

Wait 2 sec after query wahr ▾

beim Start

pausiere (ms) 1000 ▾

zeige Text verbinde "T:" Read temperature ▾ "C" − +

pausiere (ms) 2000 ▾

Query DHT11 ▾

Data pin P1 ▾

Pin pull up wahr ▾

Serial output falsch ▾

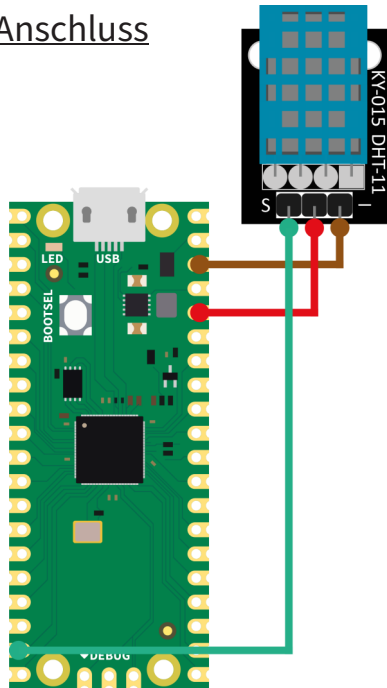
Wait 2 sec after query wahr ▾

zeige Text verbinde "H:" Read humidity ▾ "%" − +

pausiere (ms) 2000 ▾

6. VERWENDUNG MIT DEM RASPBERRY PI PICO

6.1 Anschluss



| Raspberry Pi Pico | DHT11 |
|-------------------|--------|
| GPIO14 | Signal |
| +3,3 V | +V |
| GND | GND |

6.2 Codebeispiel

Für das folgende Codebeispiel wird eine zusätzliche Bibliothek benötigt: [micropython-stubs](#) von Jos Verlinde | veröffentlicht unter der [MIT Lizenz](#). Laden Sie sich das Codebeispiel [hier](#) herunter oder übertragen Sie den folgenden Code vollständig auf Ihren Raspberry Pi Pico.

```
# Bibliotheken laden
from machine import Pin
from utime import sleep
from dht import DHT11

# Initialisierung GPIO14 und DHT11
sleep(1)
dht11_sensor = DHT11(Pin(14, Pin.IN, Pin.PULL_UP))

# Wiederholung (Endlos-Schleife)
while True:
    # Messung durchführen
    dht11_sensor.measure()
    # Werte lesen
    temp = dht11_sensor.temperature()
    humi = dht11_sensor.humidity()
    # Werte ausgeben
    print("Temperature: {}°C   Humidity: {:.0f}% ".format(temp, humi))
    print()
    sleep(3)
```

7. SONSTIGE INFORMATIONEN

Unsere Informations- und Rücknahmepflichten nach dem Elektroggesetz (ElektroG)



Symbol auf Elektro- und Elektronikgeräten:

Diese durchgestrichene Mülltonne bedeutet, dass Elektro- und Elektronikgeräte **nicht** in den Hausmüll gehören. Sie müssen die Altgeräte an einer Erfassungsstelle abgeben. Vor der Abgabe haben Sie Altbatterien und Alttakkumulatoren, die nicht vom Altgerät umschlossen sind, von diesem zu trennen.

Rückgabemöglichkeiten:

Als Endnutzer können Sie beim Kauf eines neuen Gerätes, Ihr Altgerät (das im Wesentlichen die gleiche Funktion wie das bei uns erworbene neue erfüllt) kostenlos zur Entsorgung abgeben. Kleingeräte bei denen keine äußeren Abmessungen größer als 25 cm sind können unabhängig vom Kauf eines Neugerätes in haushaltsüblichen Mengen abgeben werden.

Möglichkeit Rückgabe an unserem Firmenstandort während der Öffnungszeiten:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

Möglichkeit Rückgabe in Ihrer Nähe:

Wir senden Ihnen eine Paketmarke zu mit der Sie das Gerät kostenlos an uns zurücksenden können. Hierzu wenden Sie sich bitte per E-Mail an Service@joy-it.net oder per Telefon an uns.

Informationen zur Verpackung:

Verpacken Sie Ihr Altgerät bitte transportsicher, sollten Sie kein geeignetes Verpackungsmaterial haben oder kein eigenes nutzen möchten kontaktieren Sie uns, wir lassen Ihnen dann eine geeignete Verpackung zukommen.

8. SUPPORT

Wir sind auch nach dem Kauf für Sie da. Sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: service@joy-it.net

Ticket-System: <https://support.joy-it.net>

Telefon: +49 (0)2845 9360-50 (Mo - Do: 09:00 - 17:00 Uhr,
Fr: 09:00 - 14:30 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net