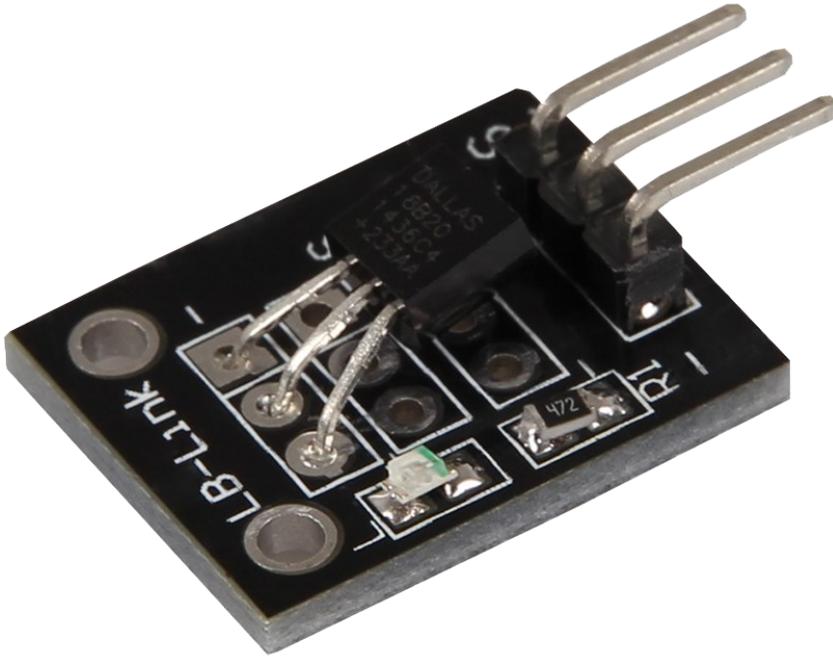


KY-001 TEMPERATURESENSOR

Temperatur Sensor für Arduino und Raspberry PI

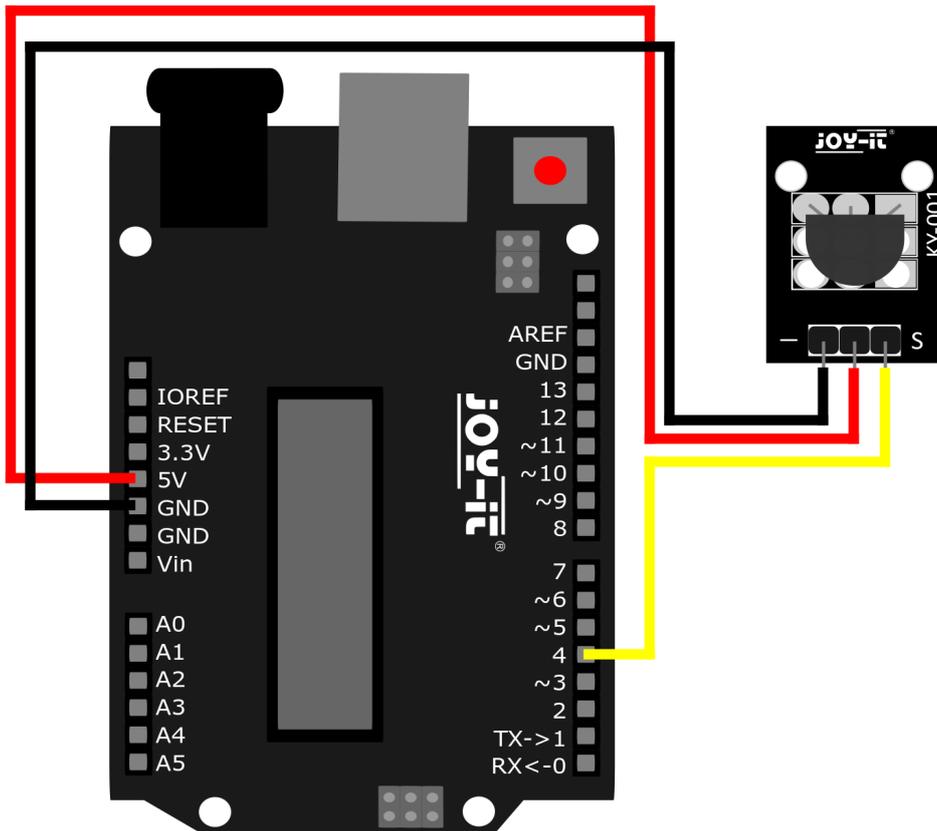


1. ALLGEMEINE INFORMATIONEN

Sehr geehrter Kunde,
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.

2. VERWENDUNG MIT DEM ARDUINO



Sensor	Arduino
Signal	Pin 4
+V	Pin 5V
GND	Pin GND

2.1 CODEBEISPIEL ARDUINO

Für das folgende Codebeispiel werden zwei zusätzliche Bibliotheken benötigt. Diese bestehen aus der **OneWire Library** von [Paul Stoffregen](#), veröffentlicht unter der MIT License, und der **Dallas Temperature Control Library** von [Miles Burton](#), veröffentlicht unter der LGPL License. Beide Bibliotheken können in der Arduino IDE unter **Werkzeuge > Bibliotheken verwalten...** heruntergeladen und installiert werden.

2.1 CODEBEISPIEL ARDUINO

```
// Required libraries are imported
#include <DallasTemperature.h>
#include <OneWire.h>

// Here the input pin is declared,
// where the sensor module is connected
#define KY001_Signal_PIN 4

// Libraries are configured
OneWire oneWire(KY001_Signal_PIN);
DallasTemperature sensors(&oneWire);

void setup() {

    // Initialization of Serial Output
    Serial.begin(9600);
    Serial.println("KY-001 Temperature Measurement");

    // Sensor is initialized
    sensors.begin();
}
// Main program loop
void loop()
{

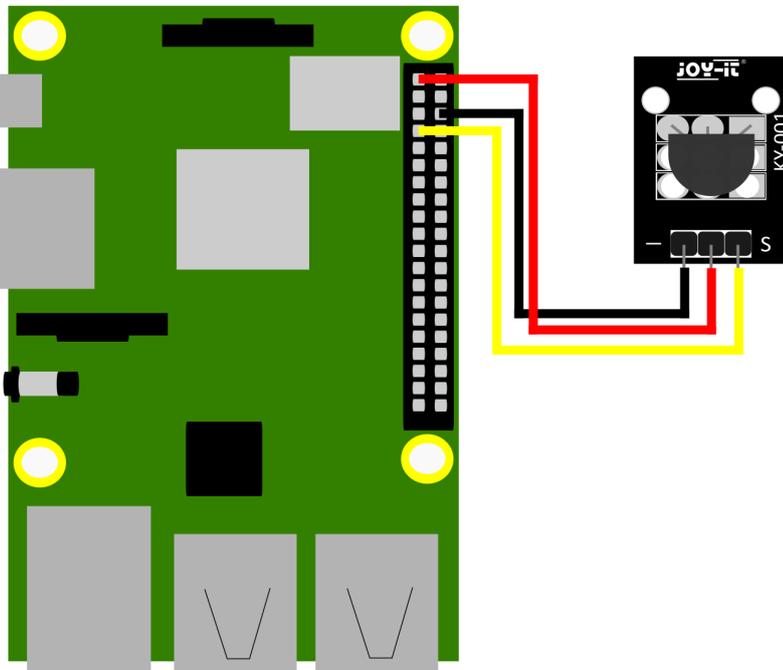
    // Temperature measurement is started...
    sensors.requestTemperatures();
    // ... and the measured temperature is displayed
    Serial.print("Temperature: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.write(176); // char symbol for the "°" symbol
    Serial.println("C");

    delay(1000); // 5s pause until the next measurement
}
```

3. VERWENDUNG MIT DEM RASPBERRY PI



Diese Anleitung wurde unter Raspberry Pi OS Bookworm für den Raspberry Pi 4 und 5 geschrieben. Sie wurde nicht mit neueren Betriebssystemen oder Hardware überprüft.



Sensor	Raspberry Pi
Signal	GPIO4 [Pin 7]
+V	3,3V [Pin 1]
GND	Masse [Pin 6]

3.1 CODEBEISPIEL RASPBERRY PI

Damit der Raspberry Pi über den One-Wire-Bus mit dem der DS18B20-Sensor kommunizieren kann, muss er zunächst aktiviert werden. Geben Sie dazu den folgenden Befehl ein:

```
sudo raspi-config
```

Gehen Sie nun auf Interface Options und aktivieren Sie die 1-Wire Schnittstelle. Ihr Raspberry Pi sollte danach automatisch neu starten.

Oder Sie können den folgenden Befehl verwenden, um manuell neu zu starten:

```
sudo reboot
```

Sie können nun das Codebeispiel für Raspberry hier kopieren:

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import glob
import time

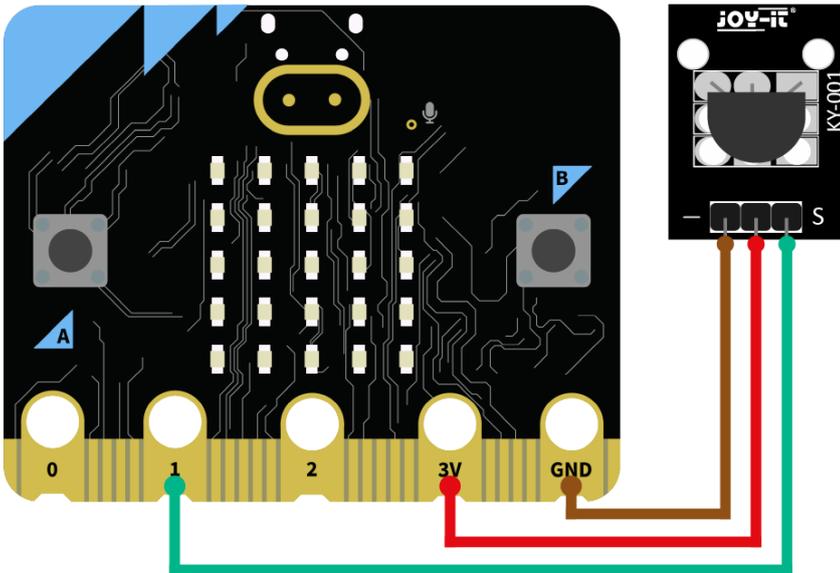
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c, temp_f

while True:
    print(read_temp())
    time.sleep(1)
```

4. VERWENDUNG MIT MICRO:BIT



Sensor	Micro:Bit
Signal	Pin 1
+V	3 V
GND	GND

4.1 CODEBEISPIEL MICRO:BIT

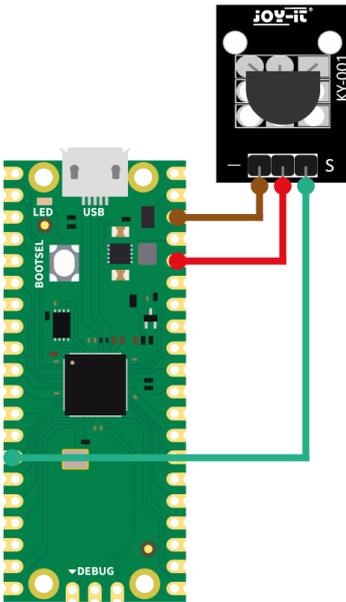
Für das folgende Codebeispiel wird eine zusätzliche Library benötigt: [pxt-ds18b20](https://github.com/DFRobot/pxt-ds18b20) von [DFRobot](https://github.com/DFRobot) | veröffentlicht unter der GUI License.

Fügen Sie die Library zu Ihrem Projekt hinzu, indem Sie auf "Erweiterungen" klicken und die folgende URL in das Suchfeld eingeben: <https://github.com/DFRobot/pxt-ds18b20.git>.

Bestätigen Sie die Suche mit [Enter]. Hierbei handelt es sich um ein Beispielprogramm, welches nach der Initialisierung des Sensors die gemessene Temperatur seriell ausgibt:

```
forever
  serial write value "temp " = pin1 Temperature_number
  pause (ms) 1000
  serial write line join "temp : " pin1 Temperature_string
  pause (ms) 1000
  show number pin1 Temperature_number
  pause (ms) 100
```

5. VERWENDUNG MIT DEM RASPBERRY PI PICO



Sensor	Micro:Bit
Signal	GP2
+V	3,3 V
GND	GND

5.1 CODEBEISPIEL RASPBERRY PI PICO

Für das folgende Codebeispiel werden zwei zusätzliche Bibliotheken benötigt:

[OneWire Library](#) von Damien P. George | veröffentlicht unter der [MIT-Lizenz](#).

[DS18x20 Library](#) von Damien P. George | veröffentlicht unter der [MIT-Lizenz](#).

Hierbei handelt es sich um ein Beispielprogramm, welches nach der Initialisierung des Sensors die gemessene Temperatur seriell ausgibt.

```
# Load libraries
import machine, onewire, ds18x20
from time import sleep

# Initialization of GPIO2
ds_pin = machine.Pin(10)

# Initialization of the sensor object
ds_sensor = ds18x20.DS18X20(onewire.OneWire(ds_pin))

# Search for all matching sensors
roms = ds_sensor.scan()

# Serial output
print("Found DS devices")
print("Temperature (°C)")

# Endless loop for continuous reading of the temperature
while True:
    ds_sensor.convert_temp()
    sleep(1)
    # Based on the number of compatible sensors found it will count up
    for rom in roms:
        # Serial output of the measured temperature
        print(ds_sensor.read_temp(rom))
        sleep(3)
```

6. ADDITIONAL INFORMATION

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 Altakkumulatoren, 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 äußere 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.

7. 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