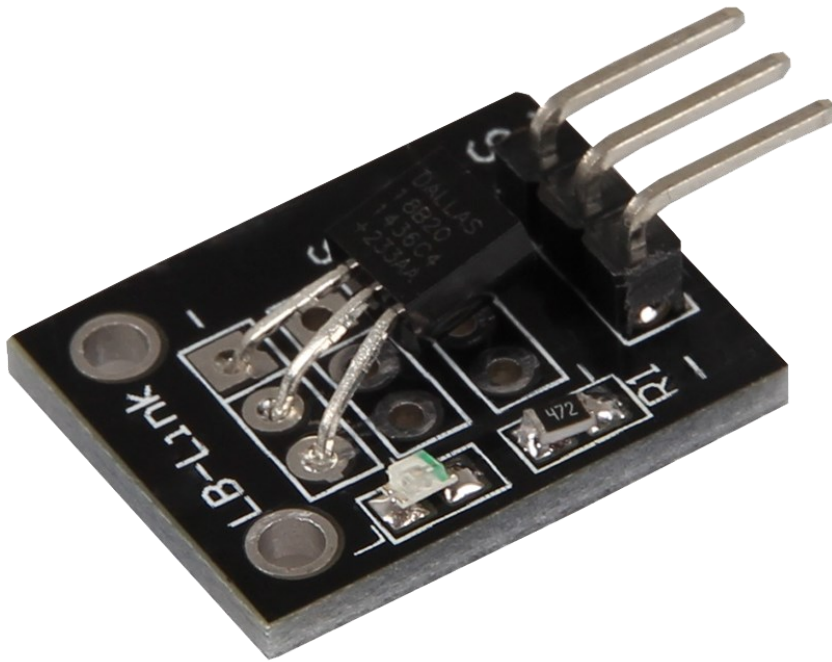


# KY-001 TEMPERATURE SENSOR

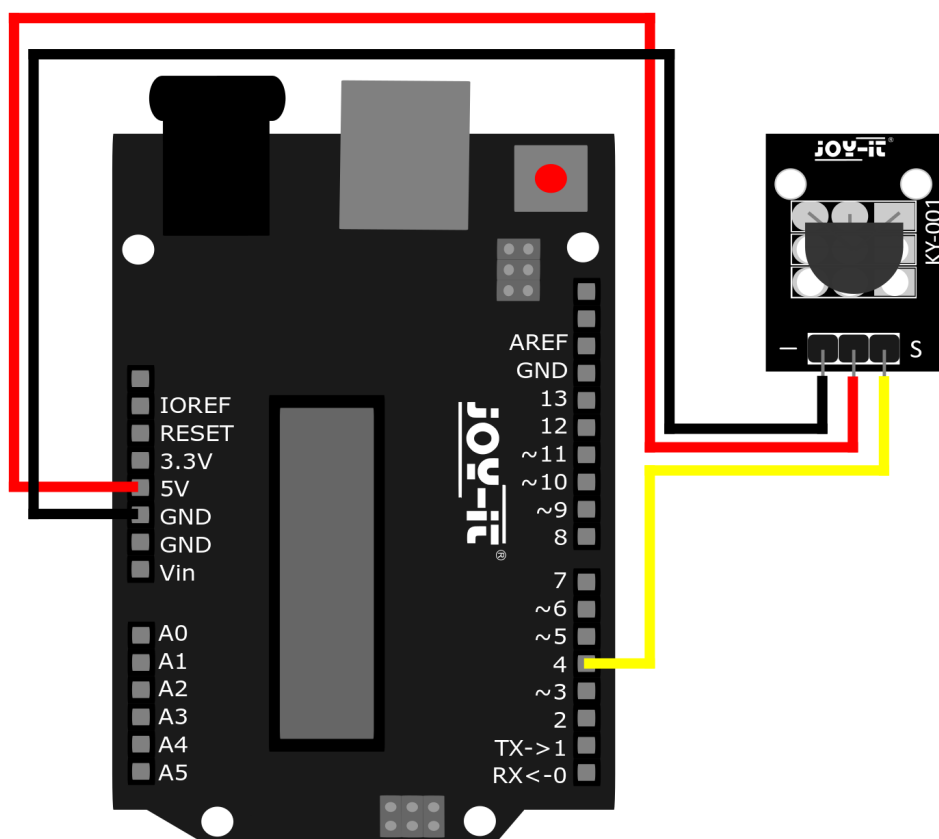
Temperature Sensor for Arduino and Raspberry PI



## 1. GENERAL INFORMATION

Dear customer,  
thank you very much for choosing our product. In following, we will introduce you to what to observe while starting up and using this product. Should you encounter any unexpected problems during use, please do not hesitate to contact us.

## 2. USAGE WITH THE ARDUINO



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

### 2.1 CODE EXAMPLE ARDUINO

For the following code example two additional libraries are needed. These consist of the **OneWire Library** by [Paul Stoffregen](#), published under the MIT License, and the **Dallas Temperature Control Library** by [Miles Burton](#), published under the LGPL License Both libraries can be managed in the Arduino IDE under Tools > Manage Libraries... and installed.

## 2.1 CODE EXAMPLE 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()
{

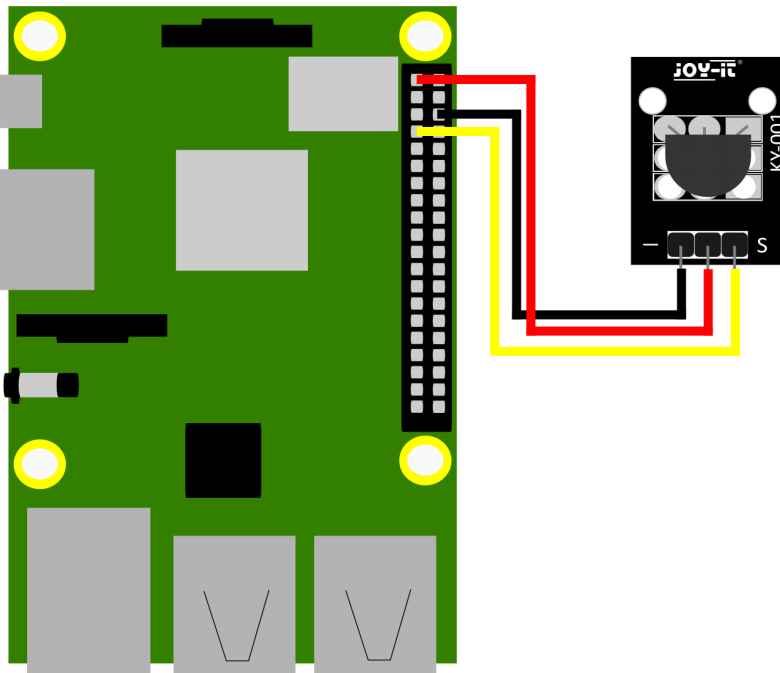
    // Temperature measurement is started...
    sensors.requestTemperatures();
    // ... and the measured temperature is displayed
    Serial.print("Temperature: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.write(176); // char symbol for the "°" symbol
    Serial.println("C");

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

### 3. USE WITH THE RASPBERRY PI



This guide was written under Raspberry Pi OS Bookworm for the Raspberry Pi 4 and 5. It has not been checked with newer operating systems or hardware.



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

#### 3.1 CODE EXAMPLE RASPBERRY PI

In order for the Raspberry Pi to communicate with the One-Wire Bus, with which the DS18B20 sensor sends its measurement data digitally, it must first be activated. To do this, enter the following command:

```
sudo raspi-config
```

Now go to Interface Options and enable the 1-Wire interface. Your Raspberry Pi should restart automatically after that.

Or you can use the following command to restart manually:

```
sudo reboot
```

You can now copy the code example for Raspberry here :

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

import glob
import time

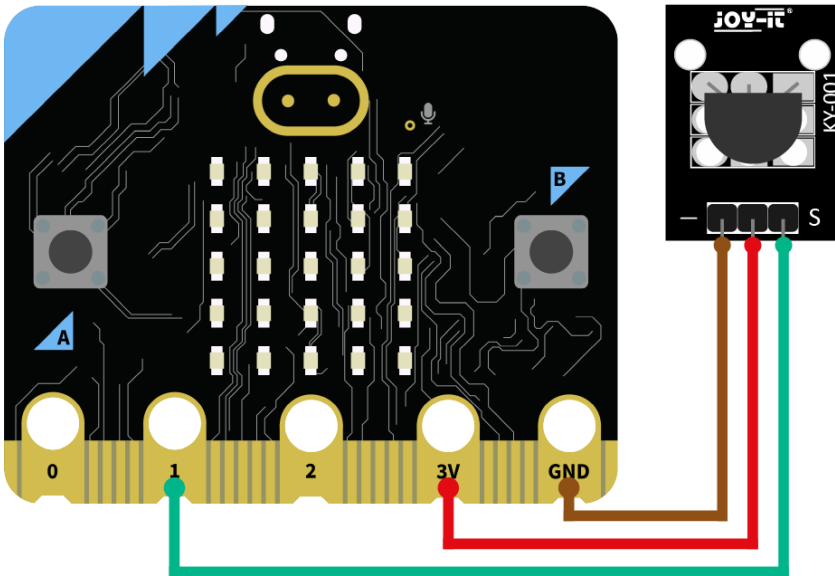
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

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. USAGE WITH MICRO:BIT



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

### 4.1 CODE EXAMPLE MICRO:BIT

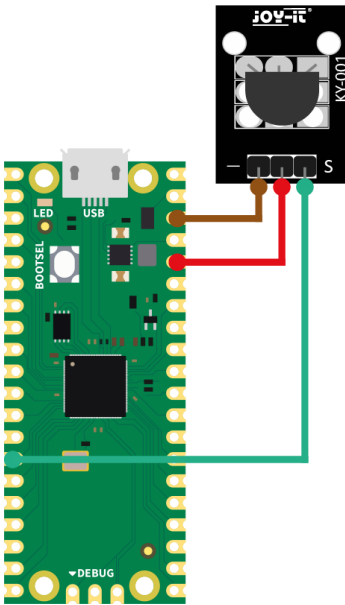
The following code example requires an additional library: [pxt-ds18b20](https://github.com/DFRobot/pxt-ds18b20) by [DFRobot](https://www.dfrobot.com/) | released under the GUI License.

Add the library to your project by clicking on "Extensions" and entering the following URL in the search field: <https://github.com/DFRobot/pxt-ds18b20.git>

Confirm the search with [Enter]. This is an example program which outputs the measured temperature serially after initializing the sensor:

```
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. USAGE WITH RASPBERRY PI PICO



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

### 5.1 USAGE WITH RASPBERRY PI PICO

For the following code example two additional libraries are needed:

[OneWire Library](#) by Damien P. George | published under the [MIT-Lizenz](#).

[DS18x20 Library](#) by Damien P. George | published under the [MIT-Lizenz](#).

This is an example program which outputs the measured temperature serially after initialization of the sensor.

```
# 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

Our information and take-back obligations according to the Electrical and Electronic Equipment Act (ElektroG)



### Symbol on electrical and electronic equipment:

This crossed-out dustbin means that electrical and electronic appliances do not belong in the household waste. You must return the old appliances to a collection point. Before handing over waste batteries and accumulators that are not enclosed by waste equipment must be separated from it.

### Return options:

As an end user, you can return your old device (which essentially fulfils the same function as the new device purchased from us) free of charge for disposal when you purchase a new device. Small appliances with no external dimensions greater than 25 cm can be disposed of in normal household quantities independently of the purchase of a new appliance.

### Possibility of return at our company location during opening hours:

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

### Possibility of return in your area:

We will send you a parcel stamp with which you can return the device to us free of charge. Please contact us by email at [Service@joy-it.net](mailto:Service@joy-it.net) or by telephone.

### Information on packaging:

If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging.

## 7. SUPPORT

If there are still any issues pending or problems arising after your purchase, we will support you by e-mail, telephone and with our ticket support system.

Email: [service@joy-it.net](mailto:service@joy-it.net)

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

Telephone: +49 (0)2845 9360—50 (Mon - Thur: 09:00 - 17:00 o'clock CET,  
Fri: 09:00 - 14:30 o'clock CET)

For further information please visit our website:

[www.joy-it.net](http://www.joy-it.net)