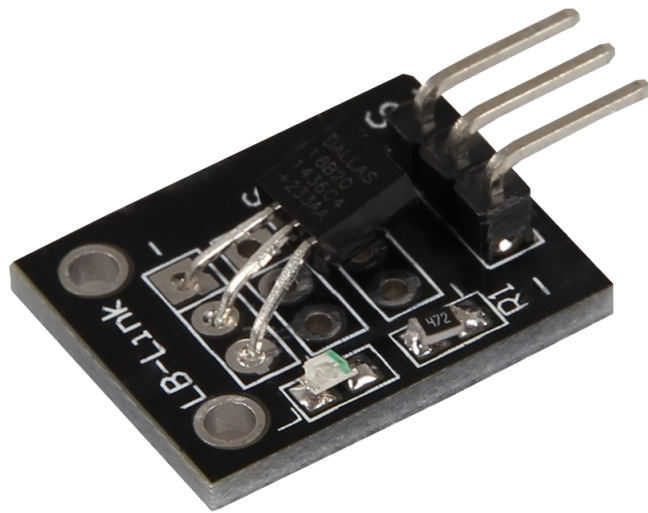


JOY-IT



Temperature Sensor
KY-001

TABLE OF CONTENTS

1. Overview
2. Pin-Assignment
3. Code Example Arduino
4. Code Example Raspberry Pi
5. Information and take-back obligations
6. Support

1. OVERVIEW

Dear customer,

Thank you for choosing our product.

In the following, we will show you what to observe during commissioning and use. Should you encounter any unexpected problems during use, please do not hesitate to contact us.

Chipset	DS18B20
Communication protocol	1-Wire
Accuracy of Measurement	9 - 12 Bit
Range of Measurement	-55°C - 125°C

2. PIN-ASSIGNMENT



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

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

3. CODE EXAMPLE ARDUINO

The following code example requires two additional libraries. 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 are included in the package and must be copied to the "library" folder before starting the Arduino IDE. You can find it by default under the following path of your Windows installation: C:\Users\[Username]\Documents\Arduino\libraries

You can download the code example for Arduinos [here](#) or copy the following one:

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

// The input pin to which the sensor module is connected is declared here.
#define KY001_Signal_PIN 4

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

void setup() {

    // Initialization Serial output
    Serial.begin(9600);
    Serial.println("KY-001 Temperaturmessung");

    // Sensor is initialized
    sensors.begin();
}

// Main program loop
void loop()
{
    // Temperature measurement is started...
    sensors.requestTemperatures();
    // ... and display measured temperature
    Serial.print("Temperatur: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.write(176); // UniCode specification of a char symbol for the "° symbol".
    Serial.println("C");

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

4. CODE EXAMPLE RASPBERRY PI

In order for the Raspberry Pi to be able to communicate with the One-Wire Bus, with which the DS18B20 sensor transmits its measurement data digitally, it must first be activated. The file `"/boot/config.txt"` must be edited. To do this, open a terminal window and enter the following command to open the file:

```
sudo nano /boot/config.txt
```

Now add the following line to the opened file at the end:

```
dtoverlay=w1-gpio,gpiopin=4
```

Save the file with the key combination **CTRL+O** and exit it with the combination **CTRL+X**.

Now restart your Raspberry Pi with the following command so that the changes you have made take effect:

```
sudo reboot
```

You can download the code example for Raspberry Pis [here](#) or copy and paste the following sections.

```
# coding=utf-8
# Required modules are imported and set up
import glob
import time
from time import sleep
import RPi.GPIO as GPIO

# At this point the pause between the individual measurements can be set
sleeptime = 1

# The One-Wire input pin is declared and the integrated pull-up resistor is activated
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# After activation of the pull-up resistor, wait until communication with the DS18B20
# sensor is established.
print 'Warte auf Initialisierung... '

base_dir = '/sys/bus/w1/devices/'
while True:
    try:
        device_folder = glob.glob(base_dir + '28*')[0]
        break
    except IndexError:
        sleep(0.5)
        continue
device_file = device_folder + '/w1_slave'

# Function is defined with which the current measured value can be read out at the
# sensor
def TemperaturMessung():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

# For initialization, the sensor is read "blind" once
TemperaturMessung()

# Temperature evaluation: With Raspberry Pi, detected one-wire slaves are stored in
# the folder /sys/bus/w1/devices/
# This folder contains the file w1-slave
# in which the data sent over the One-Wire bus is stored
# In this function, this data is analyzed and the temperature is read out and shown
```

```
def TemperaturAuswertung():
    lines = TemperaturMessung()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = TemperaturMessung()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c

# main loop
# The measured temperature is sent to the console between the individual measurements
# is a pause whose length can be set with the variable "sleeptime"
try:
    while True:
        print '-----'
        print "Temperatur:", TemperaturAuswertung(), "°C"
        time.sleep(sleeptime)
except KeyboardInterrupt:
    GPIO.cleanup()
```

5. INFORMATION AND TAKE-BACK OBLIGATIONS

Symbol on electrical and electronic equipment



This crossed-out dustbin means that electrical and electronic equipment does not belong in household waste. You must return the old devices to a collection point. Before disposal, you must separate old batteries and accumulators which are not enclosed in the old device from it.

Return options

As an end user, you can dispose of your old device (which essentially fulfils the same function as the new one purchased from us) free of charge when purchasing a new device. Small appliances where no external dimensions are larger than 25 cm can be delivered in normal household quantities, irrespective of the purchase of a new appliance.

Possibility of return at our company location during opening hours

Simac GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn

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 e-mail at Service@joy-it.net or by phone.

Information about 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.

6. SUPPORT

We also support you after your purchase. If there are any questions left or if you encounter any problems, please feel free to contact us by mail, phone or by our ticket-system on our website.

E-Mail: service@joy-it.net
Ticket-System: <http://support.joy-it.net>
Phone: +49 (0)2845 98469 – 66 (11- 18 o'clock)

For more information, please visit our website:

www.joy-it.net