



# Home » JOY-It » JOY-it LINKERKIT DS18B20 Water Protected One Wire Temperature Sensor User Manual ₹

#### Contents [hide]

- 1 JOY-it LINKERKIT DS18B20 Water Protected One Wire Temperature Sensor
- 2 Product Usage Instructions
- **3 GENERAL INFORMATION**
- 4 ANSCHLUSSBELEGUNG
- 5 USAGE WITH THE RASPBERRY PI
- 6 USAGE WITH THE ARDUINO
- 7 INFORMATION & TAKE-BACK OBLIGATIONS
- 8 SUPPORT
- 9 FAQ
- 10 Documents / Resources
  - 10.1 References



JOY-it LINKERKIT DS18B20 Water Protected One Wire Temperature Sensor



# **Product Usage Instructions**

- Thank you for choosing our product. Please refer to the following guidelines for commissioning and use.
- For any issues, contact our support team.
- The LK-Temp2 temperature sensor can be connected to digital port no. 4 of the LinkerKit module.
- Alternatively, connect it to your Raspberry Pi or Arduino following the pin assignments provided.
- Enable the One-Wire interface on GPIO pin 4 by editing the config.txt file.
- Create a program file named LK-Temp2.py and insert the provided program code.
- Save the file and execute it using the command `sudo python3 LK-Temp2.py`.
- Install the required libraries OneWire and DallasTemperature before using the sensor with your Arduino.

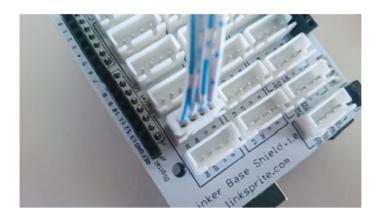
#### **GENERAL INFORMATION**

#### **Dear Customer,**

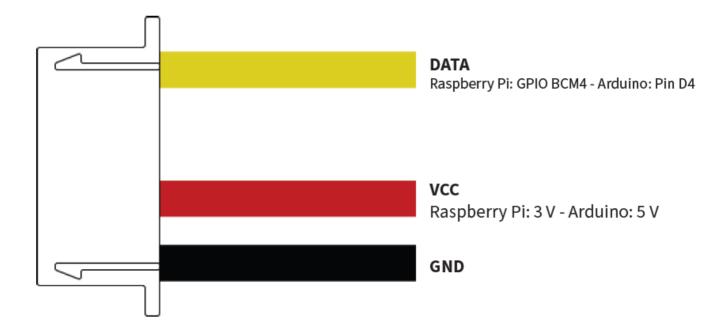
• Thank you for choosing our product. In the following, we will show you what to

- consider during commissioning and use.
- Should you encounter any unexpected problems during use, please feel free to contact us.

#### **ANSCHLUSSBELEGUNG**



- The LK-Temp2 temperature sensor is connected directly to digital port no. 4 of the LinkerKit module, as shown in the illustration.
- Alternatively, the sensor can also be connected directly to your Raspberry Pi or Arduino. The pin assignment can be seen in the following figure:



#### **USAGE WITH THE RASPBERRY PI**

- For use with the Raspberry Pi, it is first necessary to enable the One-Wire interface on the corresponding GPIO pin.
- For this, open a terminal window and edit your config.txt file with the following command:

## sudo nano /boot/firmware/config.txt

• Add the following line to the end of the file:

# dtoverlay=w1-gpio,gpiopin=4

 Now save the file with the key combination CTRL+O, confirm with Enter, and then close the editor with the combination CTRL+X. No,w restart your Raspberry Pi with the following command:

#### sudo reboot

 Now that you have connected the sensor to your Raspberry Pi, you can create the corresponding program file. Start a new terminal window on your Raspberry Pi and enter the following command:

# sudo nano LK-Temp2.py

Now insert the following program code here:

```
# 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
PullUp 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, it waits until
the communication with the DS18B20 sensor is established
print ('Waiting for initialization...')
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 from the sensor
```

```
def TemperatureMeasurement():
   f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines
# For initialization, the sensor is read out "blind" once
TemperatureMeasurement()
# The temperature evaluation: On the Raspberry Pi, detected
one-Wire slaves are stored in the folder
# /sys/bus/w1/devices/ is assigned to its own subfolder.
This folder contains the file w1-slave
# in which the data sent via the One-Wire bus is stored.
# In this function, these data are analyzed and the
temperature is read out and displayed
def TemperatureEvaluation():
    lines = TemperatureMeasurement()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = TemperatureMeasurement()
    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 program loop
# The measured temperature is output to the console -
between the individual measurements
# is a pause, the length of which can be set with the
variable "sleeptime"
try:
    while True:
        print ("Temperature:", TemperatureEvaluation(), "C")
        time.sleep(sleeptime)
except KeyboardInterrupt:
    GPIO.cleanup()
```

- Now save the file with the key combination CTRL+O, confirm with Enter and then exit the editor with the combination CTRL+X.
- Alternatively, you can download the program file <u>here</u> and copy it manually to your Raspberry Pi.
- Now you can execute the file with the following command:

# sudo python3 LK-Temp2.py

#### **USAGE WITH THE ARDUINO**

- To use the sensor with your Arduino, the library installation is required first. For this, we recommend the OneWire and DallasTemperature libraries that we have customized.
- You can download the libraries here.
- First unzip the downloaded ZIP archive. Copy both folders into the
- Arduino library directory. directory. This is located in the following location:
- C:\Users\[YOUR USERNAME]\Dokumente\Arduino\libraries
- After you have connected the sensor to your Arduino, you can transfer the corresponding program file. Copy the following program example into your Arduino IDE and transfer it to your Arduino:

```
#include <DallasTemperature.h>
#include <OneWire.h>
#define LKTemp2 4
OneWire oneWire(LKTemp2);
DallasTemperature sensors(&oneWire);
void setup() {
    Serial.begin(9600);
    Serial.println("LK-Temp2 temperature measurement");
    sensors.begin();
}
void loop()
    sensors.requestTemperatures();
    Serial.print("Temperature: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.println(" C");
    delay(1000); // 1s pause until next measurement
}
```

- Alternatively, you can download the program file <u>here</u>.
- After you have transferred the file to your Arduino, the temperature measurement is

started automatically and the results are output via the serial monitor of your Arduino IDE.

#### **INFORMATION & TAKE-BACK OBLIGATIONS**

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



#### Symbol on electrical and electronic equipment

- This crossed-out trash can means that electrical and electronic equipment does not belong in the household trash.
- You must hand in the old equipment at a collection point.
- Before handing in, you must separate used batteries and accumulators that are not enclosed in the old device from the old device.

#### **Return options:**

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

# Possibility of returning to our company location during opening hours:

• SIMAC Electronics 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. To do this, please contact us by e-mail at <a href="mailto:Service@joy-it.net">Service@joy-it.net</a> or by phone.

# Packaging information:

Please pack your old device securely for transport. 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.

#### **SUPPORT**

- We are also there for you after the purchase. If any questions remain or problems arise, we are also available to assist you via e-mail, telephone and ticket support system.
- E-Mail: <u>service@joy-it.net</u>
- Ticket-System: <a href="http://support.joy-it.net">http://support.joy-it.net</a>
- Phone: +49 (0)2845 98469 66 (10 17 Uhr)
- For more information, visit our website: www.joy-it.net
- www.joy-it.net
- SIMAC Electronics GmbH
- Pascalstr. 8 47506 Neukirchen-Vluyn

#### **FAQ**

- Q: How do I connect the sensor to my Raspberry Pi?
  - A: Connect the sensor to digital port no. 4 or follow the pin assignment for GPIO BCM4.
- Q: What libraries do I need to install to use the sensor with Arduino?
  - A: Install the OneWire and DallasTemperature libraries as recommended for Arduino usage.

# **Documents / Resources**



JOY-it LINKERKIT DS18B20 Water Protected One Wire Temperature Sen sor [pdf] User Manual

LK-Temp2, LINKERKIT DS18B20 Water Protected One Wire Temperature Sensor, LINKERKIT DS18B20, Water Protected One Wire Temperature Sensor, Protected One Wire Temperature Sensor, One Wire Temperature Sensor, Wire Temperature Sensor, Temperature Sensor, Sensor

#### References

- User Manual
- JOY-It
- ▶ JOY-It, LINKERKIT DS18B20, LINKERKIT DS18B20 Water Protected One Wire Temperature Sensor, LK-Temp2, One Wire Temperature Sensor, Protected One Wire Temperature Sensor, Sensor, Temperature Sensor, Water Protected One Wire Temperature Sensor, Wire Temperature Sensor

# Leave a comment

Your email address will not be published. Required fields are marked \*

Name

Email

☐ Save my name, email, and website in this browser for the next time I comment.

**Post Comment** 

#### Search:

Website

e.g. whirlpool wrf535swhz

Search

This website is an independent publication and is neither affiliated with nor endorsed by any of the trademark owners. The "Bluetooth®" word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. The "Wi-Fi®" word mark and logos are registered trademarks owned by the Wi-Fi Alliance. Any use of these marks on this website does not imply any affiliation with or endorsement.