



Temperatur Sensor
KY-001

INHALTSVERZEICHNIS

1. Übersicht
2. Pin-Belegung
3. Codebeispiel Arduino
4. Codebeispiel Raspberry Pi
5. Informations- und Rücknahmepflichten
6. Support

1. ÜBERSICHT

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.

| | |
|-------------------------|---------------|
| Chipsatz | DS18B20 |
| Kommunikationsprotokoll | 1-Wire |
| Messgenauigkeit | 9 - 12 Bit |
| Messbereich | -55°C - 125°C |

2. PIN-BELEGUNG



| Anschlussbelegung Arduino | |
|---------------------------|---------|
| Signal | Pin 4 |
| +V | Pin 5V |
| GND | Pin GND |

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

3. 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 sind im Paket enthalten und müssen vor dem Start der Arduino IDE in den "library"-Ordner kopiert werden. Diesen finden Sie standardmäßig unter dem folgenden Pfad Ihrer Windows-Installation:

C:\Benutzer\[Benutzername]\Dokumente\Arduino\libraries

Das Codebeispiel für Arduinos können Sie [hier](#) herunterladen oder den folgenden kopieren:

```
// Benötigte Libraries werden importiert
#include <DallasTemperature.h>
#include <OneWire.h>

// Hier wird der Eingangs-Pin deklariert, an dem das Sensor-Modul angeschlossen ist
#define KY001_Signal_PIN 4

// Libraries werden konfiguriert
OneWire oneWire(KY001_Signal_PIN);
DallasTemperature sensors(&oneWire);

void setup() {

    // Initialisierung Serielle Ausgabe
    Serial.begin(9600);
    Serial.println("KY-001 Temperaturmessung");

    // Sensor wird initialisiert
    sensors.begin();
}

//Hauptprogrammschleife
void loop()
{
    // Temperaturmessung wird gestartet...
    sensors.requestTemperatures();
    // ... und gemessene Temperatur ausgeben
    Serial.print("Temperatur: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.write(176); // UniCode-Angabe eines char-Symbols für das "°-Symbol"
    Serial.println("C");

    delay(1000); // 5s Pause bis zur nächsten Messung
}
```

4. CODEBEISPIEL RASPBERRY PI

Damit der Raspberry Pi mit dem One-Wire Bus, mit dem der Sensor DS18B20 seine Messdaten digital sendet, kommunizieren kann, muss dieser vorerst aktiviert werden. Hierbei muss die Datei **"/boot/config.txt"** editiert werden. Öffnen Sie dazu ein Terminalfenster und geben Sie den folgenden Befehl ein, um die Datei zu öffnen:

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

Ergänzen Sie nun die geöffnete Datei am Ende um die folgende Zeile:

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

Speichern Sie die Datei mit der Tastenkombination **STRG+O** und verlassen Sie diese mit der Kombination **STRG+X**.

Starten Sie Ihren Raspberry Pi nun mit dem folgenden Befehl neu, damit die vorgenommenen Änderungen wirksam werden:

```
sudo reboot
```

Das Codebeispiel für Raspberry Pis können Sie [hier](#) herunterladen oder die folgenden Ausschnitte kopieren und zusammensetzen.

```
# coding=utf-8
# Benötigte Module werden importiert und eingerichtet
import glob
import time
from time import sleep
import RPi.GPIO as GPIO

# An dieser Stelle kann die Pause zwischen den einzelnen Messungen eingestellt werden
sleeptime = 1

# Der One-Wire EingangsPin wird deklariert und der integrierte PullUp-Widerstand aktiviert
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Nach Aktivierung des Pull-UP Widerstandes wird gewartet,
# bis die Kommunikation mit dem DS18B20 Sensor aufgebaut ist
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'

# Funktion wird definiert, mit dem der aktuelle Messwert am Sensor ausgelesen werden kann
def TemperaturMessung():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

# Zur Initialisierung, wird der Sensor einmal "blind" ausgelesen
TemperaturMessung()

# Die Temperatúrauswertung: Beim Raspberry Pi werden erkannte one-Wire Slaves im Ordner
# /sys/bus/w1/devices/ einem eigenen Unterordner zugeordnet. In diesem Ordner befindet sich die Datei w1-slave
# in dem Die Daten, die über dem One-Wire Bus gesendet wurden gespeichert.
# In dieser Funktion werden diese Daten analysiert und die Temperatur herausgelesen und ausgegeben
```

```
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

# Hauptprogrammschleife
# Die gemessene Temperatur wird in die Konsole ausgegeben - zwischen den einzelnen
# Messungen
# ist eine Pause, deren Länge mit der Variable "sleeptime" eingestellt werden kann
try:
    while True:
        print '-----'
        print "Temperatur:", TemperaturAuswertung(), "°C"
        time.sleep(sleeptime)
except KeyboardInterrupt:
    GPIO.cleanup()
```

5. INFORMATIONS- UND RÜCKNAHMEPFLICHTEN

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 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.

6. 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: <http://support.joy-it.net>
Telefon: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

www.joy-it.net