

Tutorial: Temperaturmessung mit I2C-Sensor LM75 via Python

Oft und bei vielen Prozessen ist es nötig oder von Vorteil, wenn man über die Umgebungstemperatur Bescheid weiss. Ein einfacher Weg ist die Temperaturmessung mittels eines vorkalibrierten und deshalb sehr genauen Temperatursensors LM75. Durch die digitale Anbindung Via I2C-Bus sind Messfehler und Leitungsstörungen quasi ausgeschlossen.

In diesem kleinen Tutorial möchte ich Euch mit Hilfe eines kurzen Python-Scriptes zeigen, wie Ihr auf einfachste Weise den Sensor mit dem Raspberry Pi verbinden und dann auch die Temperatur abfragen könnt.

Als zusätzliche Hardware benötigt Ihr nur das Sensormodul, dass Ihr hier im Shop finden könnt: [Sensormodul](#)

Falls Ihr ein Steckbrett und Steckbrücken zur Verfügung habt, benötigt Ihr kein Werkzeug. Ansonsten werden Lötkolben, Lötzinn, Litze und idealerweise ein 40-poliger Pfostenverbinder für den Aufbau benötigt. Der Rest ist Software. Es wäre also auch wichtig, dass Euer Pi grundsätzlich eingerichtet ist, idealerweise über ssh und/oder xrdp fernbedienbar ist und über eine funktionierende Internetverbindung verfügt. Anleitungen für diese Vorbereitungen findet Ihr in den folgenden Tutorials:

- Raspberry Pi - Grundeinrichtung
- Raspberry Pi - Lan und WLAN Konfiguration
- Raspberry Pi - fernsteuern - ssh und Remote-Desktop

Um sicher zu gehen, dass auch Python installiert ist und Ihr wisst, wie die GPIO funktioniert, schaut Euch am besten auch noch dieses Tutorial an:

- Raspberry Pi - GPIO Pinbelegung und WiringPi

Mit diesem Rüstzeug solltet Ihr perfekt für diese Anleitung vorbereitet sein.

Passende Komponenten aus dem Shop zur Realisierung dieses Projektes findet Ihr am Ende dieses Tutorials aufgelistet.

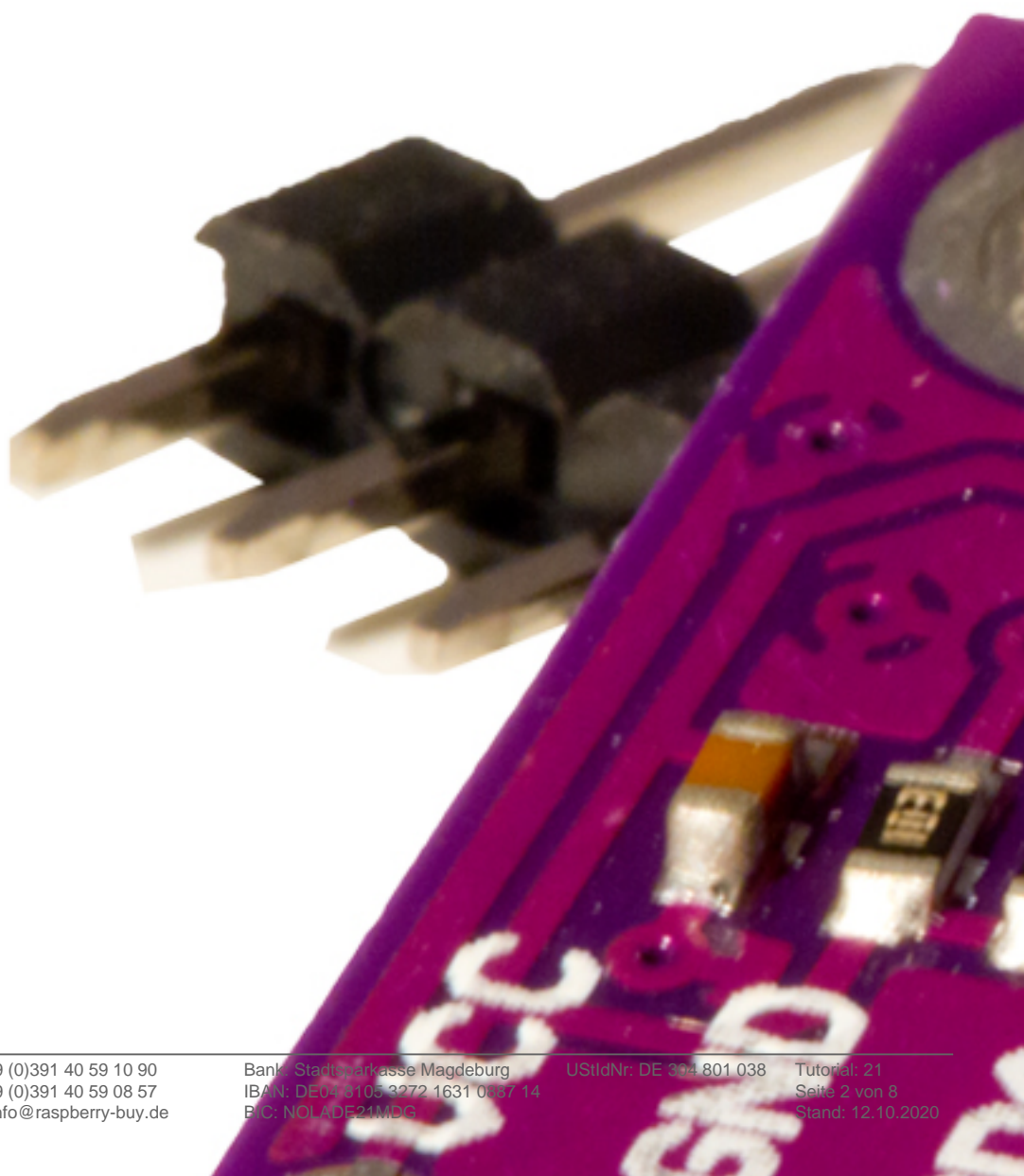
Ich wünsche Euch viel Spass und Erfolg!

Das LM75A I2C Sensormodul

Der LM75A ist ein digitaler Temperatursensor, der mit einer I2C Schnittstelle zum Datenaustausch ausgestattet ist und schon mit einer Betriebsspannung von 3,3V hervorragend funktioniert. Er ist sehr genau und muss nicht kalibriert werden. Hier geht also quasi Plug and Play.

Für den Anschluss des Sensors an den Raspberry Pi genügen, neben der Spannungsversorgung (VCC und GND) noch zwei Leitungen für den I2C-Bus (SDA und SCL), insgesamt kommen wir also mit einem vierpoligen Kabel aus. Alle diese Anschlüsse finden wir auch auf dem GPIO Pfostenverbinder unseres Pi.

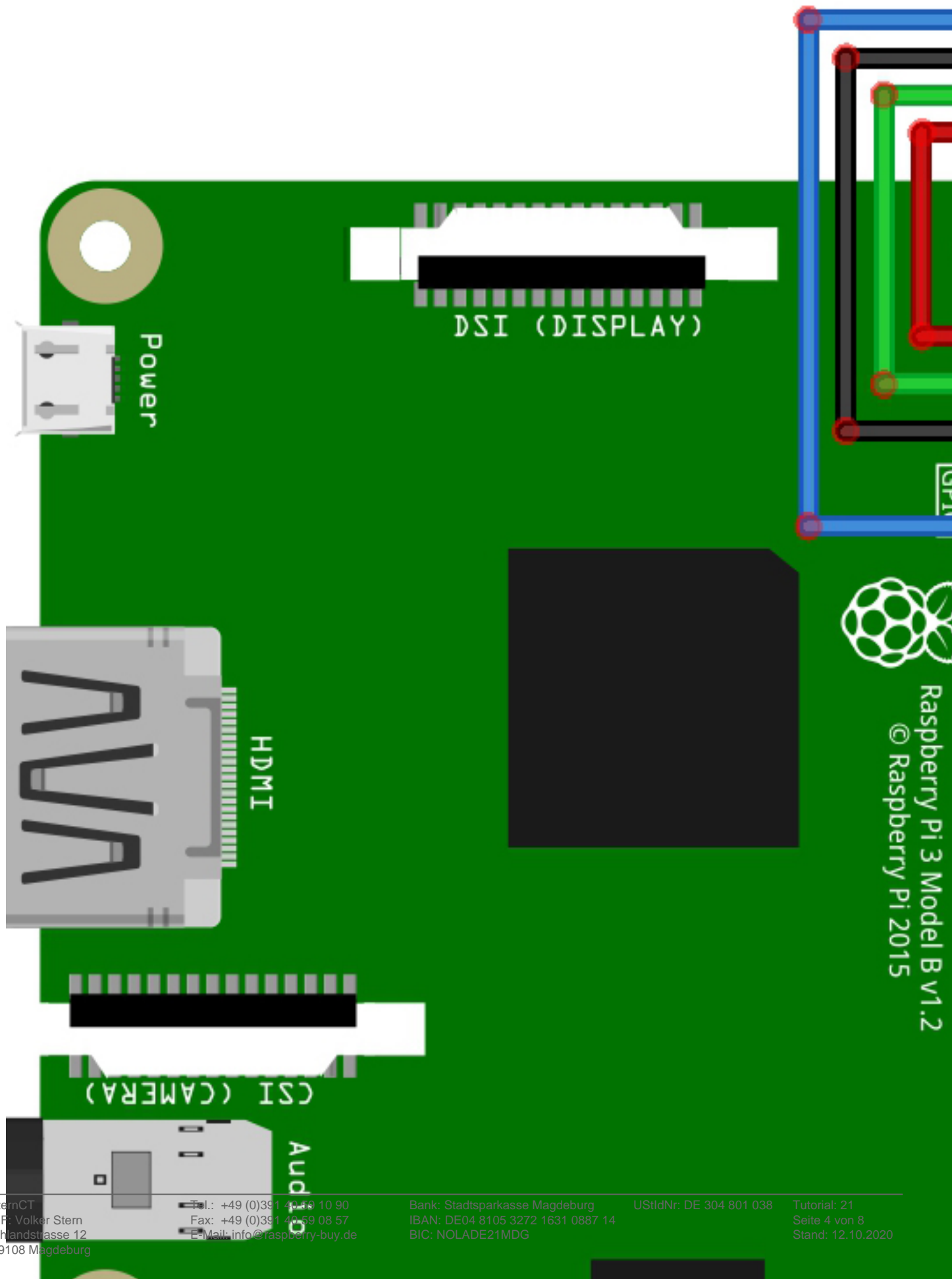
Die folgende Abbildung zeigt Euch das Sensormodul mit seinen Anschlüssen:



Im nächsten Schritt verbinden wir den Sensor mit dem Raspberry Pi.

Anschluss des LM75A Temperatursensors an den Raspberry Pi

Die folgende Abbildung zeigt Euch, wie Ihr den LM75A-Temperatursensor korrekt mit dem GPIO-Anschluss des Raspberry Pi verbinden könnt:



Nachdem nun unser Temperatursensor korrekt mit dem Raspberry Pi verbunden ist, können wir unseren Raspberry Pi starten und im nächsten Schritt erste Versuche mit dem I2C-Bus durchführen.

Überprüfung des korrekten Anschlusses des LM75 am I2C des Raspberry Pi

Nachdem wir nun unseren Temperatursensor erfolgreich mit dem Raspberry Pi verbunden haben, ist es an der Zeit zu überprüfen, ob der Sensor auch am I2C-Bus des Pi erkannt wird. Dafür gibt es eine ganz einfache Methode. Wir öffnen ein Terminal (direkt am Desktop, per SSH oder per XRDP) und tippen den folgenden Befehl ein:

```
pi@rechnername ~ $ i2cdetect -y 1 (enter)
```

Das Ergebnis dieser Abfrage sollte wie folgt aussehen:

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ i2cdetect -y 1  
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
40:  --  --  --  --  --  --  --  --  48  --  --  --  --  --  --  
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  
pi@raspberrypi:~ $
```

Die am I2C-Bus angeschlossenen Geräte identifizieren sich durch individuelle Geräte-Adressen. Beim LM75 Temperatursensor ist das die Hex-Adresse 0x48. Die Adressen sind in der Hardware des Sensors kodiert. Der LM75 hat dabei die Besonderheit, dass (um ggfs. mehrere Sensoren an einem Bus zu betreiben) hier über Lötbrücken die Adresse geändert werden kann.

Die Anzeige der Adresse 48 in der Abbildung sagt uns also, dass ein Gerät mit dieser Adresse (folglich ein LM75) mit dem I2C-Bus des Raspberry Pi verbunden ist.

So weit so gut.

Im nächsten Schritt zeige ich Euch, wie Ihr per Programmiersprache Python den I2C-Bus des Raspberry Pi benutzen und dann auch gleich unseren Temperatursensor abfragen könnt.

Abfragen der Temperatur über I2C mit Programmiersprache Python

Nachdem wir nun sicher wissen, dass unser Temperatursensor erfolgreich mit dem Raspberry Pi verbunden und auch korrekt am Bus erkannt worden ist, gehen wir dazu über, den Sensor mit Hilfe der Programmiersprache Python abzufragen.

Natürlich steht es den Puristen unter Euch frei, alles mit einem Editor zu bearbeiten, ich persönlich bevorzuge für die Bearbeitung der Scriptdatei die Nutzung einer Desktopumgebung. Wir benutzen also entweder direkt den Desktop des Raspberry Pi oder verbinden uns via XRDP mit dem Raspi.

Dort erstellen wir uns mit Hilfe des Dateimanagers zunächst einmal ein neues Verzeichnis für unsere Scriptdateien. In meinem Fall ist das im Ordner /home/pi das Unterverzeichnis Python.

Dort erstelle ich eine neue Datei und nenne Sie lm75a.py

Diese Datei öffnet Ihr am besten mit dem Programm Thonny (Himbeerchen > Entwicklung > Thonny Python IDE)

Jetzt können wir mit der Programmierung beginnen. Der Quelltext für unsere Temperaturabfrage lautet wie folgt:

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import smbus
import time

# Konstanten definieren
I2C_TMP_LM75_ADRESS = 0x48          # I2C-Adresse des LM75 Temperatursensors (0x48)
LM75_TMP_BYTE = 0x00               # Adresse im LM75, ab der Daten ausgelesen werden

# I2C-Bus initialisieren
i2c = smbus.SMBus(1)

# Funktionen definieren
def getTemp():
    temp = ""
    raw = i2c.read_i2c_block_data(I2C_TMP_LM75_ADRESS, LM75_TMP_BYTE)
    tempMsb = raw[0]
    tempLsb = raw[1]
    temp = (((tempMsb << 8) | tempLsb) >> 7) * 0.5
    if temp > 125:
        temp = temp - 256
    return(temp)

# Hauptprogramm starten
while True:
    temp = getTemp()
    print("LM75A-Temperatur: "+ str(getTemp()) + " &deg;C")
    time.sleep(5)
```

Die erste Zeile sagt Linux beim Doppelklicken der Datei, dass diese Datei mit Python ausgeführt werden möchte und wo Python zu finden ist.

Die zweite Zeile sorgt dafür, dass wir im folgenden Text keine Probleme mit irgendwelchen Umlauten bekommen.

Dann werden die Pakete smbus (i2C) und time (für die Pause) importiert.

Nachdem die notwendigen Pakete vorhanden sind, werden die Konstanten definiert.

Anschliessend initialisieren wir den I2C-Bus

Die Funktion getTemp übernimmt dann die eigentliche Arbeit, die Temperatur auszulesen und in einen Dezimalwert zu konvertieren.

Das Hauptprogramm ist dann eine Schleife, die mit jeweils 5 Sekunden Abstand die Temperatur abfragt und anzeigt.

Fertig :-)

Kombiniert mit einem I2C-OLED-Display habt Ihr dann schon ein komplettes Thermometer mit Anzeige gebaut. Ich hoffe mal wieder, dass ich Euch ein bisschen inspirieren konnte.

Bei Fragen zum Tutorial, Hinweisen oder Verbesserungsvorschlägen schickt mir einfach eine Mail.