

Luftqualitätmessung

Pia Bereuter

15.10.2024

Diese Übung zeigt wie Sensoren mit dem Raspberry Pi genutzt werden können am Beispiel des BME688 Sensors. Dieser Sensor erfasst Temperatur, Luftdruck, Luftfeuchtigkeit und Gas Scanner VOC. Die Übung zeigt wie der Sensor angeschlossen wird, wie die Sensordaten ausgelesen werden und wie die Sensordaten in einer Datei gespeichert werden können.

Inhaltsverzeichnis

1	Einführung	1
2	BME688	2
3	Übungsaufbau	3
4	Aufgabe 1: Sensormessungen ausführen	5
5	Aufgabe 2: Berechnung der Atmosphärenkorrektur für Distanzmessungen (optional)	7
	Referenzen	9

1 Einführung

Ziel dieser ersten Übung ist es den BME688 Sensor kennen zu lernen und die Sensordaten auszulesen. Der BME688 ist ein 4-in-1 Sensor für Temperatur, Luftdruck, Luftfeuchtigkeit und Gas Scanner VOC. Der Sensor verfügt über eine I2C Schnittstelle, die mit der Python Library [bme680-python](#) angesteuert und die Sensordaten ausgelesen werden können.

Unterlagen: [E01_Luftqualitaet.zip](#)

Vorbereitung

- Lest das Kapitel im Anhang zu [Raspberry Pi](#)
 - Konzentriere Dich auf die wichtigsten Details zur Inbetriebnahme des Raspberry Pi
- Lest die Dokumentation zum BME688 Sensor «Gas Sensor BME688» ([2022](#))
 - Konzentriere Dich auf die Beschreibung der Schnittstelle und technische Spezifikation auf dem Datenblatt

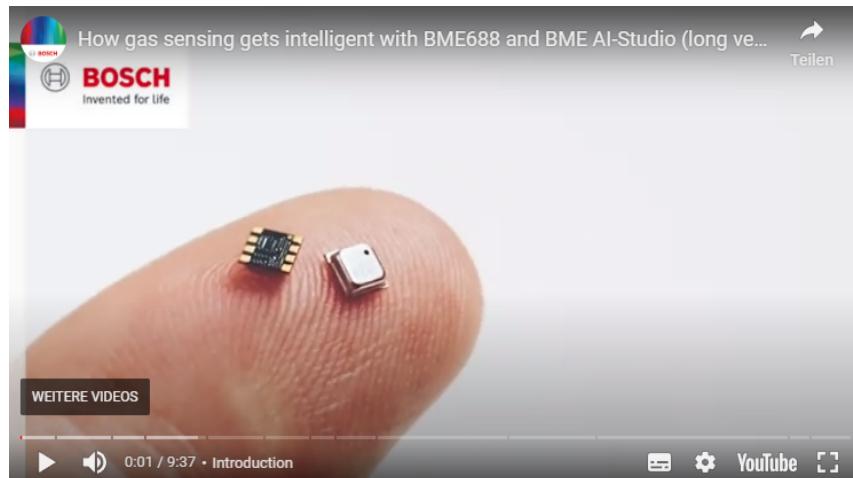


Abb. 1: Youtube Video: How gas sensing gets intelligent with BME688 and BME AI-Studio
[Youtube Video](#)

Unterlagen

Produkt	BME688 4-in-1 Air Quality Breakout
Datenblatt	Bosch Datasheet BME 688
GitHub	bme680-python Library mit Beispielen
Tutorial	Getting started with BME680 Breakout

2 BME688

BME688 - Bosch Sensor für Temperatur, Luftdruck, Luftfeuchtigkeit, Gas Scanner VOC (Abb. 2)

- Temperatur +/- 0.5°C (-40° .. -85°)
- Luftdruck +/- 0.12hPa (300...1100hPa)
- Luftfeuchtigkeit +/- 3% (0 ...100%)
- Gas Scanner VOC, VSCs (AI)
- Python, C Library
- Raspberry Pi Pins 1,3,5,7,9

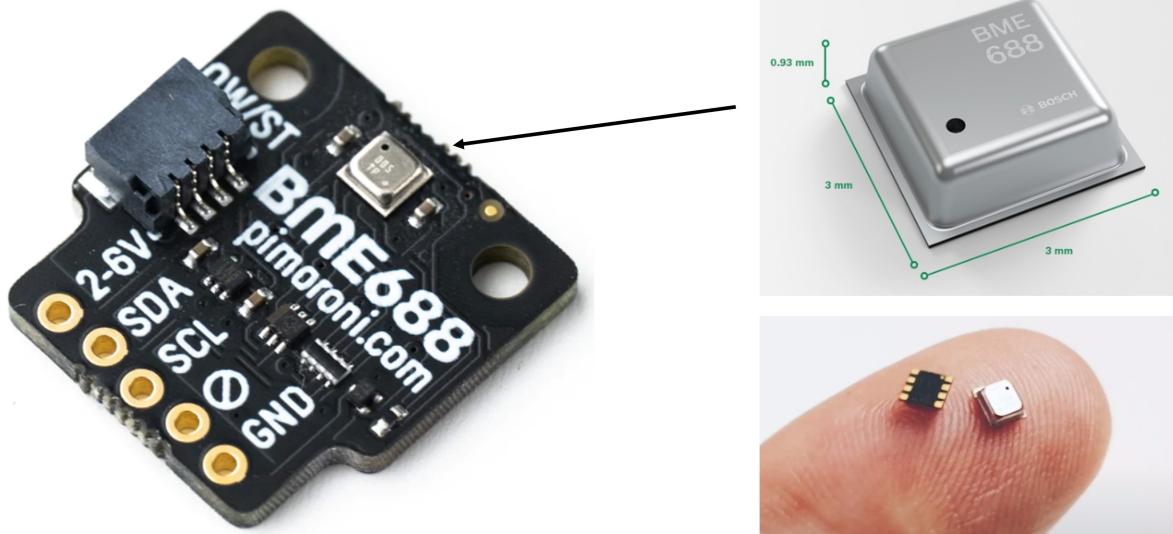


Abb. 2: BME688 Bosch Sensor für Luftqualitätsmessung mit Referenzbild für den Grössenvergleich

3 Übungsaufbau

- Schliesse den Raspberry Pi an Monitor, Keyboard und Maus an oder verbinde Dich mit diesem über SSH (und SFTP).
- Erstelle auf dem Raspberry Pi im `Documents` Ordner einen neuen Ordner `BME688`, in welchem Du Änderungen und neue Dateien für diese Übung speichern kannst.
- Schliesse den Sensor **BME688** an den Raspberry Pi über die Breakout Garden **I2C** Schnittstelle an.

Kontrolle der Hardware

Kontrolliere mit dem Befehl `i2cdetect -y 1` ob der Raspberry Pi mit dem Sensor verbunden ist und der Raspberry Pi Zugriff auf den Sensor hat. Erscheint eine Zahl, dann hat der Raspberry Pi den Sensor auf dem I2C Bus erkannt. Falls Du mehr über das Programm und den Befehl wissen möchtest, kannst Du mit dem Befehl `man i2cdetect` das Manual `man` aufrufen.

```
i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          --- --- --- --- --- --- --- --- -
10:          --- --- --- --- --- --- --- --- -
20:          --- --- --- --- --- --- --- --- -
30:          --- --- --- --- --- --- --- --- -
40:          --- --- --- --- --- --- --- --- -
50:          --- --- --- --- --- --- --- --- -
60:          --- --- --- --- --- --- --- --- -
70:          --- --- --- 76 ---
```

Hinweis

Sensor Ausrichtung beachten

Beim Anschliessen der Sensoren in die Schnittstellen des Breakout Garden **unbedingt** die korrekte Ausrichtung beachten! Die Beschriftung der Anschlüsse auf dem Sensor und dem Breakout Garden müssen übereinstimmen!



Abb. 3: Sensor links korrekt angeschlossen, rechts falsch ausgerichtet angeschlossen.

i2cdetect installieren, falls das Programm nicht existiert:

```
sudo apt-get update
sudo apt install python3-smbus
sudo apt install i2c-tools
```

Hinweis: Damit der Befehl `i2cdetect` funktioniert, muss der I2C Bus aktiviert sein. Dies kann mit dem Befehl `sudo raspi-config` im Menü Interfacing Options und I2C aktiviert werden.

Kontrolle der Installation

Aktiviere die virtuelle Python Environment (hier unter `~/.env`) und teste ob die für die Übung erforderlichen Python Libraries installiert sind. Aktiviere über folgenden Befehl die virtuelle Umgebung, welche für diese Übungen im Homeordner `~/.env` erstellt wurde¹. Die aktivierte Umgebung wird durch die Anzeige des Umgebungsnamens in der Kommandozeile angezeigt, hier mit `(.env)`.

```
source ~/.env/bin/activate
```

¹In dieser Übung wird die Python Virtual Environment `.env` im Homeordner `~/` erstellt und aktiviert. Diese kann auch an einem Ort erstellt werden, beispielweise projektbasiert. Erstellen einer Virtual Environment: `python -m venv ~/Development/Projekt1/env`

Teste ob die für die Übung erforderlichen Python Libraries installiert sind. Ein einfacher Test zur Kontrolle, ob ein Modul installiert ist, ist über das Import Statement des Moduls über die Python Konsole im Terminal. Falls kein Fehler auftritt, ist das Modul in der virtuellen Environment installiert.

```
python -c "import math"  
0  
python -c "import numpy"  
Traceback (most recent call last):  
  File "<string>", line 1, in <module>  
ImportError: No module named numpy
```

① math Modul existiert

② numpy Modul existiert nicht

Installiere das Modul mit folgendem Befehl, falls es nicht installiert ist.

```
pip install bme680
```

Kopiere (Clone) die Library mit den Beispielen auf den Raspberry Pi

Wechsle in den Ordner *Documents* und kopiere mit folgenden Befehlen die Library auf Deinen Raspberry Pi.

```
cd Documents  
git clone https://github.com/pimoroni/bme680-python  
cd bme680-python/examples
```

4 Aufgabe 1: Sensormessungen ausführen

Teste das Beispiel `read-all.py` im Ordner `examples`. Dieses Beispiel gibt die Temperatur, Luftdruck und Luftfeuchtigkeit des Sensors BME 688 aus.

```
python read-all.py
```

Mit `Ctrl+C` kann das Script wieder gestoppt werden. Die Ausgabe sollte in etwa so aussehen (gekürzt):

```
# Output Beispiel  
read-all.py - Displays temperature, pressure, humidity, and gas.  
Press Ctrl+C to exit!  
  
Calibration data:
```

```
par_gh1: -10
...
Initial reading:
gas_index: 0
gas_resistance: 1338124.79581836
heat_stable: False
humidity: 44.397
meas_index: 0
pressure: 990.82
status: 32
temperature: 28.89

Polling:
28.89 C,990.82 hPa,44.39 %RH
28.91 C,990.82 hPa,44.37 %RH,5684.846331497602 Ohms
28.94 C,990.80 hPa,44.31 %RH,5684.846331497602 Ohms
28.97 C,990.81 hPa,44.24 %RH,5684.846331497602 Ohms
29.00 C,990.81 hPa,44.19 %RH,5684.846331497602 Ohms
29.03 C,990.82 hPa,44.12 %RH,5684.846331497602 Ohms
```

Folgendes Code Snippet zeigt eine gekürzte Version des `read-all.py` Python Beispiels, der die **Temperatur**, **Luftdruck** und **Luftfeuchtigkeit** mit der BME680 Library ausgibt.

```
#!/usr/bin/env python
import bme680

try:
    sensor = bme680.BME680(bme680.I2C_ADDR_PRIMARY)                                     ①
except (RuntimeError, IOError):
    sensor = bme680.BME680(bme680.I2C_ADDR_SECONDARY)

# Oversampling Einstellungen
sensor.set_humidity_oversample(bme680.OS_2X)                                         ②
sensor.set_pressure_oversample(bme680.OS_4X)
sensor.set_temperature_oversample(bme680.OS_8X)
sensor.set_filter(bme680.FILTER_SIZE_3)

print('Sensordaten:')
try:
    while True:
        if sensor.get_sensor_data():                                                 ③
            output = '{0:.2f} C,{1:.2f} hPa,{2:.3f} %RH'.format(
                sensor.data.temperature,
                sensor.data.pressure,
                sensor.data.humidity)
```

```
    print(output)
except KeyboardInterrupt:
    pass
```

- ① Testen der beiden möglichen I2C Adressen
- ② Oversampling Einstellungen können Messungen durch Mitteln verbessern und das Rauschen und Drifts reduzieren
- ③ Sensordaten auslesen

Übung 4.1. BME 688

Studiert die Python Skripte und online Tutorial zum Sensor

- Wie wird auf den Sensor zugegriffen?
- Wie reagiert der Feuchtigkeitssensor auf Änderungen?
- Wie hoch sind die Werte für den Luftdruck, was sind Vergleichswerte?
- Wie könnt Ihr einfach Sensorwerte in eine Datei schreiben?

5 Aufgabe 2: Berechnung der Atmosphärenkorrektur für Distanzmessungen (optional)

Geodätische Distanzmessverfahren wie bei der Tachymetrie (Totalstationen) und Laserscanning benötigen eine Korrektur der Messwerte, um die Distanz zwischen zwei Punkten auf der Erde zu berechnen. Über eine *Massstabskorrektur* (ppm [mm/km]) können distanzproportionale Reduzierungen berücksichtigt werden, die *Projektionsverzerrung*, *Reduktion auf Meereshöhe* und die *Atmosphärenkorrektur* berücksichtigen. Die Luftfeuchtigkeit beeinflusst Distanzmessungen in feuchtem und heissen Klima. Diese muss den zur Messzeit geltenden atmosphärischen Bedingungen angepasst werden.

Die Korrektur wird als *Atmosphärenkorrektur* bezeichnet und ist abhängig von der *Temperatur*, dem *Luftdruck* und der *Luftfeuchtigkeit*. Die Korrektur wird in ppm (parts per million [mm/km]) angegeben und kann mit folgender Gleichung 1 berechnet werden ([Grimm, 2023; Leica Geosystems, 2022, S. 87](#)) und gilt für die Distanzmessung mit sichtbaren roten Laser

$$\Delta D_1 = 286.338 - \left[\frac{0.29535 \cdot p}{(1+\alpha \cdot t)} - \frac{4.126 \cdot 10^{-4} \cdot h}{(1+\alpha \cdot t)} \cdot 10^x \right] \quad (1)$$

ΔD_1	Atmosphärische Korrektur	[ppm]
p	Luftdruck	[mbar]
t	Lufttemperatur	[°C]
h	relative Luftfeuchte	[%]
α	= $\frac{1}{273.15}$	
x	= $(7.5 \cdot \frac{t}{237.3 + t}) + 0.7857$	

Folgende Funktion in Python berechnet die Atmosphärenkorrektur für Geodätische Distanz-messverfahren basierend auf der Gleichung 1

```
import math
def calculate_atmospheric_correction(temperature, air_pressure, humidity):
    # Constants
    ALPHA = 1 / 273.15
    X = (7.5 * temperature / (237.3 + temperature)) + 0.7857

    # Interim results
    denominator = 1 + ALPHA * temperature

    formula0 = 286.338
    formula1 = 0.29535 * air_pressure / denominator
    formula2 = 4.126 * 10 ** (-4) * humidity / denominator
    formula3 = 10 ** X

    # ppm-calculation
    correction_ppm = round(formula0 - (formula1 - formula2 * formula3), 2)

    # Return important values as a dictionary
    result = {
        "temperature": temperature,
        "air_pressure": air_pressure,
        "humidity": humidity,
        "correction_ppm": correction_ppm
    }
    return result
```

Für Distanzmessungen höchster Genauigkeit sollte die atmosphärische Korrektur auf 1ppm genau bestimmt werden, folglich sollten die Messwerte mindestens folgenden Genauigkeit aufweisen: die Temperatur auf 1°C, der Luftdruck auf 3mbar (1Millibar = 1hPa) und die Luftfeuchtigkeit auf 20 genau bestimmt werden (Leica Geosystems, 2022, S. 87).

Übung 5.1. Atmosphärenkorrektur

Nutzt die Funktion `calculate_atmospheric_correction` um eine atmosphärische Korrektur mit den momentan gemessenen amtmosphärischen Bedingungen zu berechnen.

- Wie hoch ist die Korrektur mit den Werten des BME688 Sensors?
- Wie hoch ist dieselbe Korrektur bei doppelter Luftfeuchtigkeit?
- Wie hoch ist die Korrektur bei 20°C, 1000hPa und 50% sowie 100% Luftfeuchtigkeit?
- Erfüllt der BME688 Sensor die Anforderungen für Distanzmessungen höchster Genauigkeit?

Referenzen

Gas Sensor BME688. (2022). In *Bosch Sensortec*. <https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme688/>.

Grimm, D. (2023). *2030 Geodätische Messtechnik GMT I*.

Leica Geosystems. (2022). *Leica TS60/MS60/TM60 Gebrauchsanweisung*.