

# Bewegungsmessung

Pia Bereuter

15.10.2024

Bewegungssensoren werden heute fast allgegenwärtig in vielen Bereichen eingesetzt, wie in Smartphones oder auch vielen Messgeräten der Geomatik. Diese Übung führt in die Funktionsweise von MEMS Bewegungssensoren ein und zeigt wie Bewegungsmessungen mit einem ICM20948 9DoF Motion Sensor durchgeführt werden können.

## Inhaltsverzeichnis

1	Einführung	1
2	Beschleunigungssensoren IMU	2
3	ICM20948 9DoF Motion Sensor	3
4	Übungsaufbau	4
5	Aufgabe 1: Bewegungsmessungen durchführen	4
6	Aufgabe 2: Magnetometer	5
	Referenzen	6

## 1 Einführung

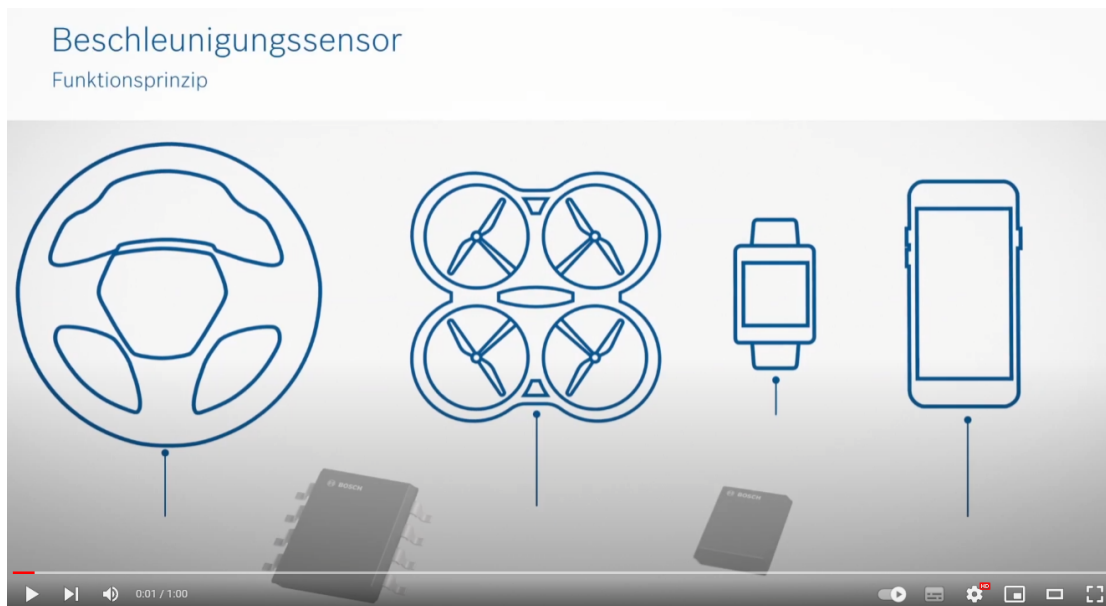
Ziel dieser Übung ist es Bewegungsmessung mit inertialen Messeinheiten (IMU) über den ICM20948 Bewegungssensor kennen zu lernen und die Sensordaten auszulesen und testen. Der *ICM20948* ist ein 9DoF Bewegungssensor, der über eine I2C Schnittstelle mit dem Raspberry Pi verbunden wird und einer Python Library angesteuert werden kann.

**Unterlagen:** *E03\_Bewegungsmessung.zip*

### Vorbereitung

- Schaut das Video zur Funktionsweise von MEMS Bewegungssensoren
  - Video: [How MEMS Accelerometer Gyroscope Magnetometer Work](#) (bis Minute 2:50), sowie folgendes
  - Video: [Bosch Funktionsprinzip eines Beschleunigungssensors](#)
- Studiere das Datenblatt zum ICM-20948 ([InvenSense, 2021](#))
  - In welchen Temperaturbereichen kann der Sensor eingesetzt werden?

- Installiere auf deinem Smartphone die Applikation [phyphox](#) und teste die Beschleunigungssensoren deines Smartphones.



**Abb. 1:** How MEMS Accelerometer Gyroscope Magnetometer Work & Arduino Tutorial  
[Youtube Video](#)

---

## Unterlagen

---

Produkt	<a href="#">ICM20948 Breakout</a>
Datenblatt	<a href="#">ICM 20948</a>
GitHub	<a href="#">icm20948-python</a>

---

## 2 Beschleunigungssensoren IMU

Beschleunigungssensoren, oder inertielle Messeinheit (inertial measurement unit IMU) messen die Beschleunigung von Objekten mit dem Messprinzip der Trägheit und erfassen die Kraft die auf die Masse des Objekt wirkt, wenn dieses beschleunigt wird.

Für die Erfassung der sechs kinematischen Freiheitsgrade werden drei Achsen der Beschleunigung (Accelerometer) und drei Achsen der Rotation (Gyroskop) gemessen, die die Beschleunigungsmessung und Winkelgeschwindigkeit der Drehraten ausgeben. Für die Erfassung der Orientierung im Raum wird ein Magnetometer eingesetzt, welches die Ausrichtung des Objekts im Magnetfeld der Erde misst, um die Ausrichtung im Raum zu bestimmen. Wenn alle drei Sensoren kombiniert werden, spricht man von 9DoF Motion Sensoren, der neun Freiheitsgrade (9 Degrees of Freedom) misst.

Beschleunigungssensoren werden in vielen Anwendungen eingesetzt, wie z.B. in der Automobilindustrie (Auslösen von Airbags), der Luft- und Raumfahrt, der Medizintechnik (Beschleuni-

gungssensoren in Herzschrittmachern) und der Unterhaltungselektronik (Smartphones für die Ausrichtung des Bildschirms).

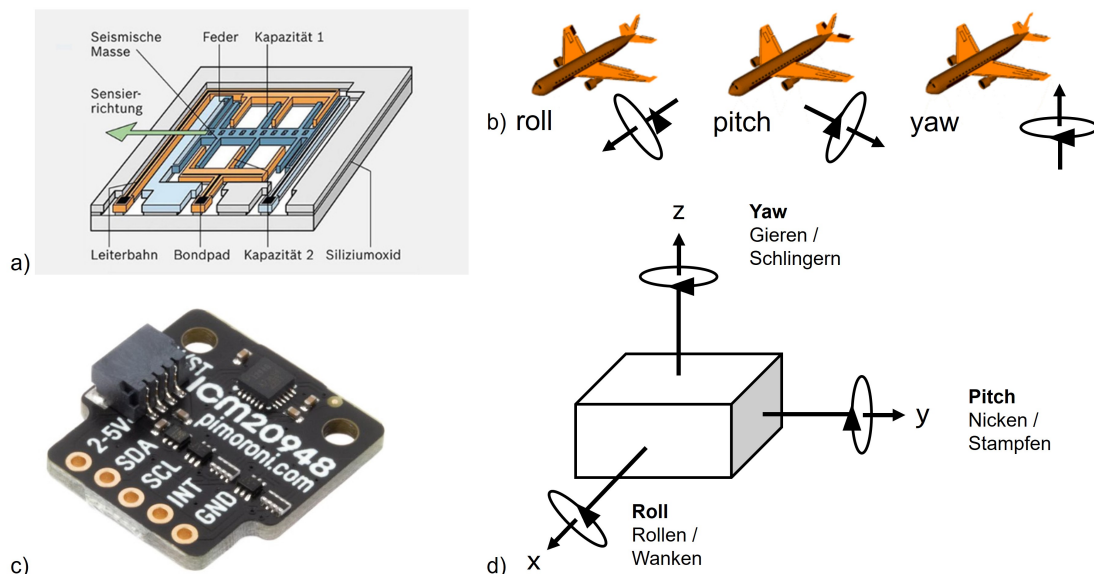
### 3 ICM20948 9DoF Motion Sensor

Der ICM-20948 von **TDK InvenSense** (Abb. 2) ist ein 9-Achsen MEMS Bewegungssensor, mit einem 3-Achsen Gyroskop, einem 3-Achsen Beschleunigungssensor und einem 3-Achsen Magnetometer und sehr geringem Stromverbrauch. Er enthält zwei Chip, einen für die Bewegungsmessung mit Gyroskop und Beschleunigungssensor und einen zweiten für das Magnetometer.

MEMS sind mikroelektromechanische Systeme, die aus mikroskopisch kleinen mechanischen und elektrischen Komponenten bestehen. Diese werden meist aus Silicium hergestellt und sind im Falle von Beschleunigungssensoren sehr kleine Massen, die sich bei Beschleunigung bewegen und die Änderung der elektrischen Kapazität messen.

9DoF Motion Accelero-, Gyro-, Magnetometer

- $\pm 2/\pm 4/\pm 8/\pm 16$  g 3-axis accelerometer
- $\pm 250/\pm 500/\pm 1000/\pm 2000$  DPS (degrees per second) 3-axis gyroscope
- 3-axis compass with wide range up to  $\pm 4900$   $\mu$ T
- Python, C Library
- I2C interface (address: 0x68 or 0x69)
- Qw/ST (Qwiic/STEMMA QT) connector
- I2C interface (address 0x68/0x69 (cut trace))



**Abb. 2:** a) schematische Darstellung eines MEMS Beschleunigungssensors Quelle: Bosch, b) raw, pitch, roll bei Flugzeugen, c) ICM-20948 Breakout von Pimoroni, d) Orientierung von IMU Sensoren.

## 4 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 `ICM20948`, in welchem Du Änderungen und neue Dateien für diese Übung speichern kannst.
- Schliesse den Sensor **ICM20948** an den Raspberry Pi über die Breakout Garden **I2C** Schnittstelle korrekt an (siehe [E01 Luftqualität](#)), so dass die Beschriftung der Anschlüsse am Sensor und bei der Schnittstelle übereinstimmen.
- Kontrolliere mit dem Befehl `i2cdetect -y 1` ob der Raspberry Pi mit dem Sensor verbunden ist. Der Sensor sollte auf der Adresse `0x68` erkannt werden.
- Aktiviere die virtuelle Environment von Python mit `source ~/.env/bin/activate` und kontrolliere, ob die Library `icm20948` installiert ist mit `python -c "import icm20948"`. Bei einer Fehlermeldung muss die Library in der aktivierten virtuellen Environment mit `pip install icm20948` installiert werden.

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

```
cd Documents
git clone https://github.com/pimoroni/icm20948-python
cd icm20948-python/examples
```

## 5 Aufgabe 1: Bewegungsmessungen durchführen

Teste das Beispiel `read-all.py` im Ordner *examples*. Dieses Beispiel gibt die Messungen der einzelnen Bewegungsmessungen aus, der Beschleunigung, Winkelgeschwindigkeit und Orientierung mit dem Accelerometer, Gyrometer und Magnetometer.

Startet das Script mit `python read-all.py`<sup>1</sup>. Mit `Ctrl+c` kann das Script wieder gestoppt werden. Die Ausgabe sollte in etwa so aussehen (gekürzt):

```
python read-all.py
read-all.py
Reads all ranges of movement: accelerometer, gyroscope and compass heading.
Press Ctrl+C to exit!

Accel: 01.01 -0.02 00.01
Gyro:  -0.42 01.73 00.01
Mag:   -86.85 57.45 34.05

Accel: 01.01 -0.02 00.02
```

<sup>1</sup>Nicht vergessen zuerst die korrekte virtuelle Environment mit den installierten Libraries über `source ~/.env/bin/activate` zu starten

```
Gyro:  -0.40 01.50 -0.16
Mag:   -85.35 55.50 34.80
```

Folgendes Code Snippet zeigt eine gekürzte Version des `read-all.py` Python Beispiels für die Ausgabe der Beschleunigungsmessung.

```
#!/usr/bin/env python
import time
from icm20948 import ICM20948

imu = ICM20948()

while True:
    x, y, z = imu.read_magnetometer_data() ①
    ax, ay, az, gx, gy, gz = imu.read_accelerometer_gyro_data() ②

    print(""" # <3>
Accel: {:05.2f} {:05.2f} {:05.2f}
Gyro:  {:05.2f} {:05.2f} {:05.2f}
Mag:   {:05.2f} {:05.2f} {:05.2f}""".format(
        ax, ay, az, gx, gy, gz, x, y, z
    ))

    time.sleep(0.25) ④
```

- ① Auslesen des Magnetometers (x,y,z)
- ② Auslesen des Accelerometers (ax,ay,az) und Gyrometers (gx,gy,gz)
- ③ Messwerte auf der Konsole ausgeben
- ④ Warten 0.25 Sekunden (damit die Ausgabe nicht zu schnell ist)

## 6 Aufgabe 2: Magnetometer

Das Beispiel `bargraph.py` zeigt die Messwerte des Magnetometers in einem Balkendiagramm an und zeigt die Orientierung des Sensors an je nach dem über welche Achse gemessen wird. Der Befehl `python bargraph.py --help` zeigt die Optionen des Skripts an.

Die Ausgabe sollte für die Option `--graph` in etwa so aussehen:

```
python bargraph.py --graph
bargraph.py - Convert raw values to heading

Rotate the sensor through 360 degrees to calibrate.

Press Ctrl+C to exit!
```

### Übung 5.1. Bewegungsmessung

- Führe das Beispiel `read-all.py` aus und beobachte die Messwerte.
- Versuche den Sensor jeweils leicht in eine Richtung zu bewegen, zu drehen, zu kippen und beobachte die Messwerte.
- Versuche den Sensor zu leicht zu schütteln und beobachte die Messwerte.
- Vergleiche die Messwerte mit den Messwerten deines Smartphones mit der App *phyphox*.
- Versucht zu eruieren wie die Achsen orientiert sind und vergleicht mit anderen Gruppen.
- Schreibe die Messwerte in eine Datei und visualisiere diese mit einem Plot, modifiziere dazu das Beispiel `read-all.py` und speichere die Datei als `read-csv.py`, so dass die Messungen zeilenweise mit einem Separator gespeichert werden. Ausgaben aus einem Plot können mit dem Befehl `python read-csv.py > imu_horizontal.csv` in eine Datei geschrieben werden. Nun könnt ihr verschiedene Versuche mit der IMU durchführen und in einer Datei speichern. Die Datei könnt ihr beispielsweise mit *LibreOffice Calc* oder *Excel* öffnen und die Daten visualisieren.

043.5 

↪

### Übung 6.1. Bar graph

- Kalibriere den Sensor und vergleiche die Orientierung des Sensors mit den Himmelsrichtungen
- Vergleiche die Ausgaben auch mit der Orientierung des Smartphones und den Messwerten der App *phyphox*.
- Studiert den Code und versucht die Funktionsweise zu verstehen.

## Referenzen

InvenSense, T. (2021). *ICM-20948 World's Lowest Power 9-Axis MEMS MotionTracking™ Device*.