

# Biometrie

Pia Bereuter

16.10.2024

Biometrische Messungen ermöglichen die Erfassung von physiologischen Daten wie Herzfrequenz oder Sauerstoffsättigung. Während diese Sensoren nicht typische Sensoren für IoT Projekte sind, können sie für spezielle Anwendungen wie Gesundheitsüberwachung oder Fitness eingesetzt werden. Diese Übung führt in die biometrischen Messungen mit dem MAX30101 Sensor ein und zeigt wie die Herzfrequenz gemessen werden kann.

## Inhaltsverzeichnis

1	Einführung	1
2	MAX30101 Breakout Heart Rate, Oximeter, Smoke Sensor	2
3	Übungsaufbau	3
4	Aufgabe 1: Biometriemessung Konsole	3
	Referenzen	5

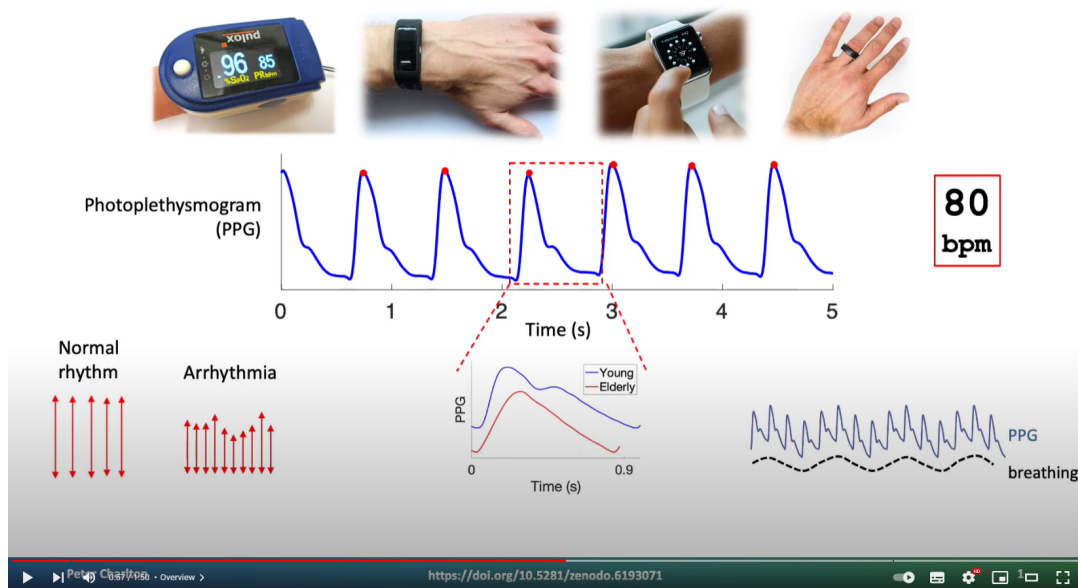
## 1 Einführung

Ziel dieser Übung ist es biometrische Messungen mit dem *MAX30101* Sensor durchzuführen und den Sensor in seiner Funktionsweise zu untersuchen. Der *MAX30101* ist ein Sensor zur Messung der Herzfrequenz und der Sauerstoffsättigung im Blut und ein Rauch-/Partikelsensor. Er verfügt über eine I2C Schnittstelle und kann über eine Python Library angesteuert werden.

**Unterlagen:** *E06\_Biometrie.zip*

### Vorbereitung

- Schaut Euch folgendes Video von Peter Charlton zur Funktionsweise der Herzfrequenzmessung an: [Photoplethysmography in a minute \(and a bit\) - Peter Charlton](#)
- Studiere das Datenblatt zum MAX30101 ([maxim integrated, 2020](#))
  - In welchen Temperaturbereichen kann der Sensor eingesetzt werden?
  - Welches ist die höchste Abtastrate für die Sauerstoffmessungen?
  - Was sind Anwendungsgebiete für diesen Sensor?



**Abb. 1:** Photoplethysmography in a minute (and a bit) - Peter Charlton [Youtube Video](#)

## Unterlagen

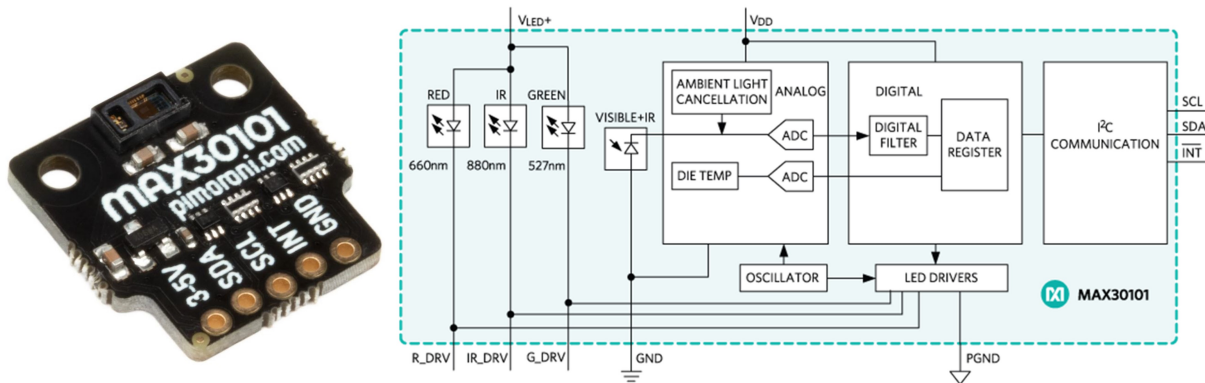
Produkt	<a href="#">MAX30101 Breakout</a>
Datenblatt	<a href="#">MAX30101</a>
GitHub	<a href="#">max30105-python</a>

## 2 MAX30101 Breakout Heart Rate, Oximeter, Smoke Sensor

Der MAX30101 ist hochentwickelter Herzfrequenz-, Oximeter- und Rauch-/Partikelsensor. Der Sensor verfügt über drei LEDs (rot, grün, IR) und Photodetektoren. Mit der Photoplethysmographie (photoplethysmography PPG) kann über die Farbveränderung der Haut bei jedem Herzschlag dieser detektiert werden, wenn der Sensor leicht auf den Finger gedrückt wird. Der Sensor kann auch dazu benutzt werden um Partikel in der Luft wie Rauch zu erkennen, indem er die Lichtmenge, die von Partikeln in der Luft zurückgeworfen wird misst.

MAX30101 Breakout - Heart Rate, Oximeter, Smoke Sensor Breakout

- MAX30101 - heart rate, oximeter, smoke sensor
- Green, red, and infra-red LEDs
- Photodetectors
- Ambient light rejection
- Temperature sensor



**Abb. 2:** links: MAX30101 Breakout von Pimoroni, rechts: funktionale Diagramm des MAX30101 Moduls Quelle: maxim integrated (2020)

### 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 `MAX30101`, in welchem Du Änderungen und neue Dateien für diese Übung speichern kannst.
- Schliesse den Sensor **MAX30101** 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 `0x57` erkannt werden.
- Aktiviere die virtuelle Environment von Python mit `source ~/.env/bin/activate` und kontrolliere, ob die Library `v15315cx_ctypes` installiert ist mit `python -c "import v15315cx_ctypes"`. Bei einer Fehlermeldung muss die Library in der aktivierten virtuellen Environment mit `pip install v15315cx_ctypes` 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/max30105-python
cd max30105-python/examples
```

### 4 Aufgabe 1: Biometriemessung Konsole

Teste das Beispiel `read-heartbeat.py` im Ordner `examples`. Dieses Beispiel liest die Pulsschläge pro Minute in PPM (beats per minute). Ein erkannter Pulsschlag wird mit einem `<3` angezeigt.

Startet das Script mit `python read-heartbeat.py`. Mit `Ctrl+c` kann das Script wieder gestoppt wrden. Die Ausgabe sollte in etwa so aussehen (gekürzt):

```
python read-heartbeat.py
NOTE! This code should not be used for medical diagnosis.
...
Starting readings in 10 seconds...

    BPM: 0.00  AVG: 0.00
    BPM: 0.00  AVG: 0.00
    BPM: 0.00  AVG: 0.00
    BPM: 0.00  AVG: 0.00
    BPM: 45.55  AVG: 45.96
<3 BPM: 55.75  AVG: 59.90
    BPM: 55.75  AVG: 59.90
<3 BPM: 59.64  AVG: 70.65
    BPM: 59.64  AVG: 70.65
```

Folgendes Code Snippet zeigt eine gekürzte Version des `read-heartbeat.py` Python Beispiels für die Ausgabe des Herzschlags.

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

# NOTE! This code should not be used for medical diagnosis. It's
# for fun/novelty use only, so bear that in mind while using it.

import time
from max30105 import MAX30105, HeartRate

max30105 = MAX30105()
max30105.setup(leds_enable=2)
max30105.set_led_pulse_amplitude(1, 0.2)
max30105.set_led_pulse_amplitude(2, 12.5)
max30105.set_led_pulse_amplitude(3, 0)
max30105.set_slot_mode(1, 'red')
max30105.set_slot_mode(2, 'ir')
max30105.set_slot_mode(3, 'off')
max30105.set_slot_mode(4, 'off')

def display_hearttrate(beat, bpm, avg_bpm):
    print("{} BPM: {:.2f}  AVG: {:.2f}".format("<3" if beat else " ",
        bpm, avg_bpm))
hr = HeartRate(max30105)

delay = 10
print("Starting readings in {} seconds...\n".format(delay))
time.sleep(delay)
```

```
try:  
    hr.on_beat(display_hearttrate, average_over=4)  
except KeyboardInterrupt:  
    pass
```

⑥

- ① Sensor initialisieren
- ② Sensor konfigurieren (LED Pulse Amplitude, Slot Mode der LED)
- ③ Funktion zur Darstellung des Herzschlags in der Konsole
- ④ Initialisierung des Herzschlagdetektors
- ⑤ 10 Sekunden warten und dann die Herzschläge ausgeben
- ⑥ Bei einem erkannten Herzschlag wird die Funktion `display_hearttrate` aufgerufen und der Herzschlag wird über 4 Sekunden gemittelt in der Konsole ausgegeben.

### Übung 4.1. Pulsmessung

- Führe das Beispiel `read-heartbeat.py` aus und beobachte die Messwerte.
- Führe unterschiedliche Tests durch und behandle den Sensor freundlich.
- Vergleiche die Messwerte und kontrolliere die Werte in dem Du den eigenen Puls misst oder mit einer Sportuhr vergleichst.
- Studiere den Code der Beispiele und versuche die Funktionsweise zu verstehen.

## Referenzen

maxim integrated. (2020). *MAX30101 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*.