



TECHNISCHE UNIVERSITÄT
CHEMNITZ

**Fakultät für Naturwissenschaften
Institut für Physik
Fortgeschrittene Themen der Physik und Sensorik**

5. Übung: Photoplethysmographie (PPG)

Lars Hornig

Anna Richter

Wintersemester 19/20

Optische Sensoren für Sauerstoffsättigung und Puls

Photoplethysmographie (PPG)

Optisches Verfahren

Verfahren zur Messung der Volumenschwankungen eines Körperteils oder Organs (griech. plethore = Fülle)

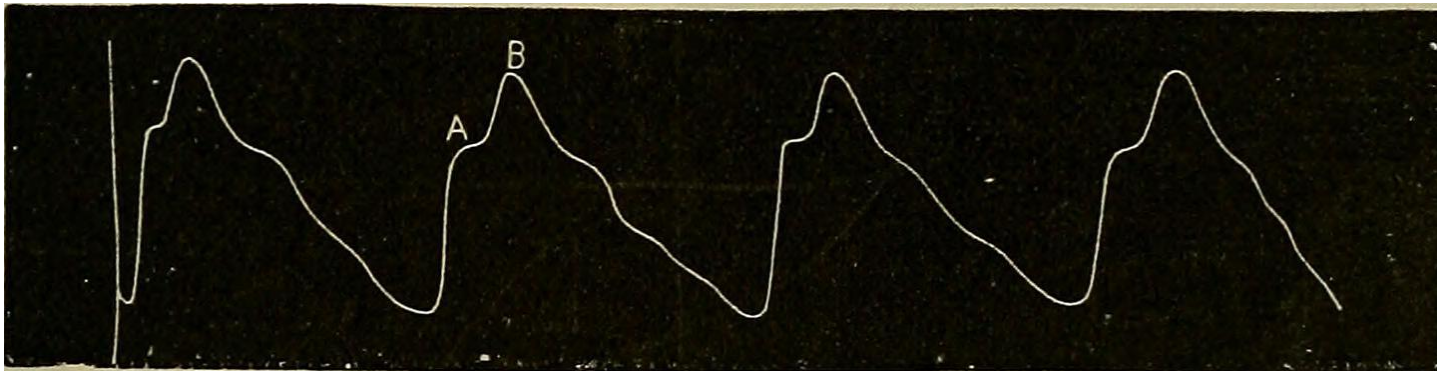
Grundidee:

Blutgefüllte Arterien sind ein optisches Medium, das unterschiedlich viel Lichtintensität absorbiert, transmittiert und reflektiert.

Quelle dieser Folie: Teresa Puder, Vortrag Fallstudien

Was ist der Puls?

- Der **Puls** ist die mechanische Reaktion der Gefäße im Blutkreislauf auf den Herzschlag. Durch das Anspannen der Herzmuskeln (Systole) entsteht eine Druckwelle im Blut, welche die Wände der Blutgefäße ausdehnt. Den Puls beschreibt man anhand seiner Frequenz, seines Verlaufs und Amplitude.
- Welche Informationen kann man aus dem Puls schließen?
Eine Pulsmessung ermöglicht Rückschlüsse auf die Herzaktivität und **Eigenschaften der Arterien (je nach Messort)**.

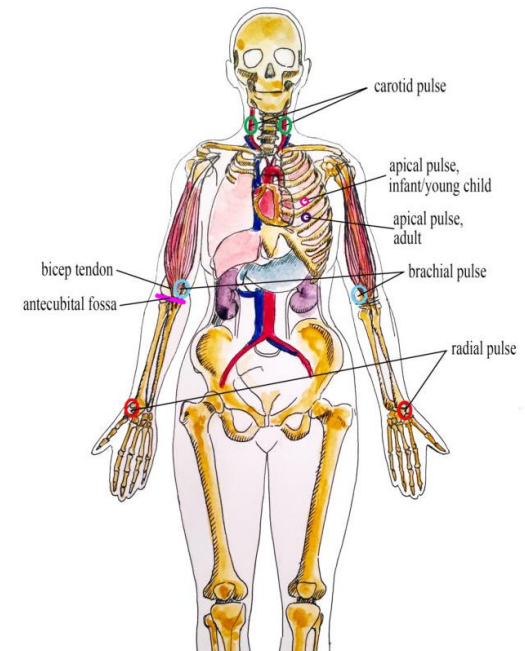


Anstieg und Abfall des Blutdrucks aufgrund des Pulses

Bildquelle: <https://en.wikipedia.org/wiki/Pulse>

Was ist der Puls?

- Was sind zu erwartende Pulsfrequenzen?
Die Pulsfrequenz hängt u.a. vom Alter und der physischen Belastung der Person ab.
Der Ruhepuls eines Erwachsenen liegt zwischen 60-100 Schläge pro Minute.¹
- Wo könnte man den Puls messen?
Es gibt verschiedene Orte für die Pulsmessung, die immer einen Einfluss auf den Messwert haben.
Beispielorte: Handgelenkpuls (Radialispuls)
Halsschlagaderpuls (Carotispuls)
Fingerspitze (Unsere Anwendung)



¹ <https://my.clevelandclinic.org/health/diagnostics/17402-pulse--heart-rate>
Bildquelle: <https://opentextbc.ca/vitalsign/chapter/what-is-pulse/>

Was ist die Sauerstoffsättigung?

- Die **Sauerstoffsättigung** (sO_2) ist die Konzentration des mit Sauerstoff beladenen Hämoglobins (oxygenierten) im gesamten Hämoglobin.¹

$$sO_2 = \frac{c(HbO_2)}{c(Hb_{Gesamt})} \cdot 100\%$$

C ... Konzentration

Das Gesamthämoglobin setzt sich aus verschiedenen Hämoglobinarten.¹

$$c(Hb_{Gesamt}) = c(HbO_2) + c(Hb) + c(MetHb) + c(COHb) + \dots$$

Für die spätere Berechnung der Sauerstoffsättigung im Blut betrachten wir jedoch nicht alle Hämoglobinarten, sondern nur oxygeniertes und desoxygeniertes Hämoglobin.

Dies vereinfacht die Berechnung und ist möglich, da alle anderen Hämoglobinarten im Normalfall nur kaum im Blut vertreten sind.

Wir ermitteln also die prozentuale, partielle Sauerstoffsättigung im Blut, auch funktionelle Sauerstoffsättigung genannt.

$$s_pO_2 = \frac{cHbO_2}{cHbO_2 + cHb} \cdot 100\%$$

¹ Dissertation „MobilePulsoxymetrie“ von N. E. Klug an der TU München
<https://mediatum.ub.tum.de/doc/1224528/1224528.pdf>

Was ist die Sauerstoffsättigung?

- Welche Informationen kann man aus der Sauerstoffsättigung schließen?
Mittels der Sauerstoffsättigung kann auf die Versorgung des Gewebes am Messort und erweitert auf die Fertigkeit der Herzen den Körper zu versorgen, geschlossen werden.

Table 2.1 Organ robustness to anoxia (cardiac arrest), a consequence of metabolic rate and cellular oxygen stores. *Survival* time is the time before cellular damage occurs after total loss of oxygen delivery. *Revival* time is the time before function of the organ can no longer be restored. Revival times are generally four times longer than survival times in most organs except the brain, which has a revival time five times longer than its survival time (adapted from Nunn 1987).

Organ	Survival time after onset of anoxia
Cerebral cortex	less than 1 min
Heart	5 min
Liver and kidney	10 min
Skeletal muscle	2 h

Was ist die Sauerstoffsättigung?

- Was sind zu erwartende Werte für die Sauerstoffsättigung?
In einem gesunden Erwachsenen ist die Sauerstoffsättigung über 90%.
- Wo könnte man die Sauerstoffsättigung messen?
Beispielsorte:
Finger (Vorteil: einfache Fixierung, Nachteil: häufig Bewegungsartefakte)
Ohrläppchen (Vorteil: gute Durchblutung, Nachteil: geringe Gewebsschichtdicke) ¹



¹ Dissertation „MobilePulsoxymetrie“ von N. E. Klug an der TU München
<https://mediatum.ub.tum.de/doc/1224528/1224528.pdf>
Bildquelle: [https://en.wikipedia.org/wiki/Oxygen_saturation_\(medicine\)](https://en.wikipedia.org/wiki/Oxygen_saturation_(medicine))

Optische Sensoren für Sauerstoffsättigung und Puls

Photo**ple**thysmographie (PPG)

Optisches Verfahren

Verfahren zur Messung der Volumenschwankungen eines Körperteils oder Organs (griech. plethore = Fülle)

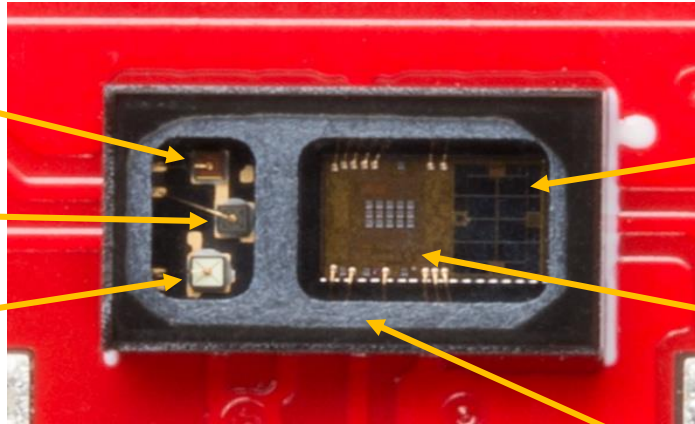
Grundidee:

Blutgefüllte Arterien sind ein optisches Medium, das unterschiedlich viel Lichtintensität absorbiert, transmittiert und reflektiert.

Unser Messgerät: Pulsoximeter (am Finger)

Mittels eines Pulsoximeters wird der Puls gemessen und auf die arterielle Sauerstoffsättigung (= S_aO_2 , wirkliche Sauerstoffsättigung im arteriellen Blut) geschlossen.

Pulsoximeter Max30105



Rote LED
660 nm

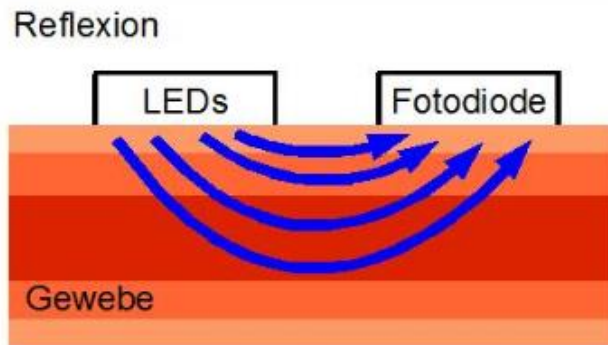
Fotodiode

Grüne LED
537 nm

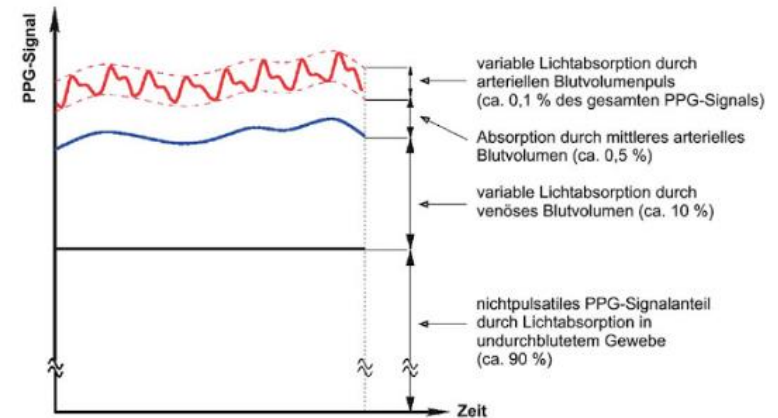
IR LED
880 nm

Application Specific
Integrated Circuit ASIC

Deckglas



Wir nutzen die
Reflexionspulsoxymetrie anstatt
der Transmissionspulsoxymetrie.



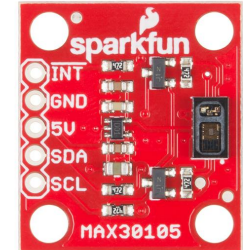
Bildquelle Reflexion: Dissertation „MobilePulsoxymetrie“ von N. E. Klug an der TU München
<https://mediatum.ub.tum.de/doc/1224528/1224528.pdf>

Pulsoximeter Max30105



Bildquelle Sensor:
https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library

Pulsoximeter Max30105



1. Wie kommunizieren wir mit dem Max30105?
2. Was müssen wir **auf** dem Max30105 einstellen?
3. Wie erhalten wir Daten **vom** Max30105?

1. Wie kommunizieren wir mit dem Max30105?

Für die Kommunikation nutzen wir den I²C-Datenbus.

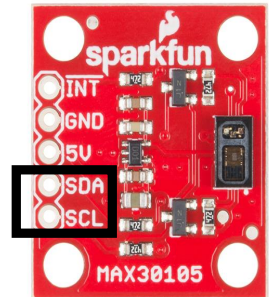
Der I²C-Datenbus benötigt die zwei Leitungen „Serial Data“ (**SDA**) und „Serial Clock“ (**SCL**). Über **SDA** werden die Daten übertragen während **SCL** den Busstakt (Zeitpunkte zu denen Daten gesendet werden) regelt.

Am Max30105 sind bereits Pins für **SDA** und **SCL** vorgegeben.
Am ESP32 müssen wir die Pins definieren.

```
#include<Wire.h>          /* I2C Bibliothek*/

void setup() {
  Serial.begin(115200);

  // Wire.begin (SDA, SCL);
  Wire.begin (26, 25);
  Wire.setClock(400000);   // Gibt an in welcher Frequenz die Verbindung sich aktualisiert. Hier sind es 400KHz
}
```



2. Was müssen wir auf dem Max30105 einstellen?

Register Maps and Descriptions

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
STATUS											
Interrupt Status 1	A_FULL	PPG_RDY	ALC_OVF	PROX_INT				PWR_RDY	0x00	0x00	R
Interrupt Status 2							DIE_TEMP_RDY		0x01	0x00	R
Interrupt Enable 1	A_FULL_EN	PPG_RDY_EN	ALC_OVF_EN	PROX_INT_EN					0x02	0x00	R/W
Interrupt Enable 2							DIE_TEMP_RDY_EN		0x03	0x00	R/W
FIFO											
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W
Overflow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W
CONFIGURATION											
FIFO Configuration	SMP_AVE[2:0]			FIFO_ROLL_OVER_EN	FIFO_A_FULL[3:0]				0x08	0x00	R/W
Mode Configuration	SHDN	RESET				MODE[2:0]			0x09	0x00	R/W
SpO ₂ Configuration	0 (Reserved)	SPO2_ADC_RGE [1:0]		SPO2_SR[2:0]			LED_PW[1:0]		0x0A	0x00	R/W
RESERVED									0x0B	0x00	R/W
LED Pulse Amplitude	LED1_PA[7:0]								0x0C	0x00	R/W
	LED2_PA[7:0]								0x0D	0x00	R/W
	LED3_PA[7:0]								0x0E	0x00	R/W
RESERVED									0x0F	0x00	R/W
Proximity Mode LED Pulse Amplitude	PILOT_PA[7:0]								0x10	0x00	R/W
Multi-LED Mode Control Registers		SLOT2[2:0]				SLOT1[2:0]			0x11	0x00	R/W
		SLOT4[2:0]				SLOT3[2:0]			0x12	0x00	R/W

Register Maps and Descriptions (continued)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
RESERVED									0x13-0x17	0xFF	R/W
RESERVED									0x18-0x1E	0x00	R
DIE TEMPERATURE											
Die Temp Integer	TINT[7:0]								0x1F	0x00	R
Die Temp Fraction					TFRAC[3:0]				0x20	0x00	R
Die Temperature Config								TEMP_EN	0x21	0x00	R
RESERVED									0x22-0x2F	0x00	R/W
PROXIMITY FUNCTION											
Proximity Interrupt Threshold	PROX_INT_THRESH[7:0]								0x30	0x00	R/W
PART ID											
Revision ID	REV_ID[7:0]								0xFE	0xFF*	R
Part ID	PART_ID[7]								0xFF	0x15	R

*XX denotes a 2-digit hexadecimal number (00 to FF) for part revision identification. Contact Maxim Integrated for the revision ID number assigned for your product.

Wir greifen über den I²C-Datenbus auf einen Speicher im Max30105 zu, in dem alle Einstellungen und Messwerte des Sensor gespeichert sind.

Solch ein Speicher heißt **Register**. Diese Art der Gerätekonfiguration und Datenauslesung wird häufig und bei verschiedensten Geräten verwendet.

2. Was müssen wir auf dem Max30105 einstellen?

Register Maps and Descriptions

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
STATUS											
Interrupt Status 1	A_FULL	PPG_RDY	ALC_OVF	PROX_INT				PWR_RDY	0x00	0x00	R
Interrupt Status 2							DIE_TEMP_RDY		0x01	0x00	R
Interrupt Enable 1	A_FULL_EN	PPG_RDY_EN	ALC_OVF_EN	PROX_INT_EN					0x02	0x00	R/W
Interrupt Enable 2							DIE_TEMP_RDY_EN		0x03	0x00	R/W
FIFO											
FIFO Write Pointer				FIFO_WR_PTR[4:0]					0x04	0x00	R/W
Overflow Counter				OVF_COUNTER[4:0]					0x05	0x00	R/W
FIFO Read Pointer				FIFO_RD_PTR[4:0]					0x06	0x00	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W
CONFIGURATION											
FIFO Configuration	SMP_AVE[2:0]			FIFO_ROLL_OVER_EN	FIFO_A_FULL[3:0]				0x08	0x00	R/W
Mode Configuration	SHDN	RESET				MODE[2:0]			0x09	0x00	R/W
SpO ₂ Configuration	0 (Reserved)	SPO2_ADC_RGE [1:0]		SPO2_SR[2:0]			LED_PW[1:0]		0x0A	0x00	R/W
RESERVED									0x0B	0x00	R/W
LED Pulse Amplitude	LED1_PA[7:0]								0x0C	0x00	R/W
	LED2_PA[7:0]								0x0D	0x00	R/W
	LED3_PA[7:0]								0x0E	0x00	R/W
RESERVED									0x0F	0x00	R/W
Proximity Mode LED Pulse Amplitude	PILOT_PA[7:0]								0x10	0x00	R/W
Multi-LED Mode Control Registers		SLOT2[2:0]				SLOT1[2:0]			0x11	0x00	R/W
		SLOT4[2:0]				SLOT3[2:0]			0x12	0x00	R/W

Register Maps and Descriptions (continued)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
RESERVED									0x13-0x17	0xFF	R/W
RESERVED									0x18-0x1E	0x00	R
DIE TEMPERATURE											
Die Temp Integer	TINT[7:0]								0x1F	0x00	R
Die Temp Fraction					TFRAC[3:0]				0x20	0x00	R
Die Temperature Config								TEMP_EN	0x21	0x00	R
RESERVED									0x22-0x2F	0x00	R/W
PROXIMITY FUNCTION											
Proximity Interrupt Threshold	PROX_INT_THRESH[7:0]								0x30	0x00	R/W
PART ID											
Revision ID	REV_ID[7:0]								0xFE	0xFF*	R
Part ID	PART_ID[7]								0xFF	0x15	R

*XX denotes a 2-digit hexadecimal number (00 to FF) for part revision identification. Contact Maxim Integrated for the revision ID number assigned for your product.

Wir greifen über den I²C-Datenbus auf einen Speicher im Max30105 zu, in dem alle Einstellungen und Messwerte des Sensor gespeichert sind.

Solch ein Speicher heißt **Register**. Diese Art der Geräte-konfiguration und Datenauslesung wird häufig und bei verschiedensten Geräten verwendet.

2. Was müssen wir auf dem Max30105 einstellen?

- Wie funktioniert ein Register? (Aus dem Datenblatt herauskopiert)

SpO₂ Configuration (0x0A)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
SpO ₂ Configuration		SPO2_ADC_RGE<1:0>		SPO2_SR[2:0]			LED_PW[2:0]		<u>0x0A</u>	0x00	R/W

Was befindet sich in diesem Registerplatz?

Der Inhalt des Registers in einzelne Bit unterteilt. 8 Bit sind maximal in einem Registerplatz gespeichert.

Wo befindet sich der Registerplatz?

Darf ich in dem Registerplatz nur lesen (R) oder auch schreiben (W)? (R/W = beides)

2. Was müssen wir auf dem Max30105 einstellen?

- Wie funktioniert ein Register? (Aus dem Datenblatt herauskopiert)

SpO₂ Configuration (0x0A)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
SpO ₂ Configuration		SPO2_ADC_RGE<1:0>		SPO2_SR[2:0]			LED_PW[2:0]		0x0A	0x00	R/W

Bits 4:2: SpO₂ Sample Rate Control

These bits define the effective sampling rate with one sample consisting of one IR pulse/conversion and one RED pulse/conversion.

The sample rate and pulse width are related in that the sample rate sets an upper bound on the pulse width time. If the user selects a sample rate that is too high for the selected LED_PW setting, the highest possible sample rate is programmed instead into the register.

Table 6. SpO₂ Sample Rate Control

SPO2_SR[2:0]	SAMPLES PER SECOND
000	50
001	100
010	200
011	400
100	800
101	1000
110	1600
111	3200



2. Was müssen wir auf dem Max30105 einstellen?

- Wie schreiben wir in das Register?

SpO₂ Configuration (0x0A)

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
SpO ₂ Configuration		SPO2_ADC_RGE<1:0>	SPO2_SR[2:0]			LED_PW[2:0]			0x0A	0x00	R/W

Im `void setup()`

```
// Samplerate festlegen
```

```
Wire.beginTransmission(0x57);
```

```
Wire.write(0x0A);
```

```
Wire.write(0b00010100);
```

```
Wire.endTransmission();
```

- Übertragung an den Max30105 vorbereiten
- Adresse des Registerplatzes auf dem Max30105 wählen
- Inhalt des Registerplatzes wählen
- Alle gewählten Einstellungen abschicken an den Max30105

0 0 0 1 0 1 0 0

2. Was müssen wir auf dem Max30105 einstellen?

- Wie lesen wir aus einem Register?

```
Wire.beginTransaction(max30101);           // Übertragung an Sensor vorbereiten
Wire.write(0x07);                          // Adresse des 1. zu lesenden Registers
Wire.endTransmission();
Wire.requestFrom(max30101, 6, true);       // 6 Bytes anfordern

if (Wire.available())                     // Wenn die Daten angekommen sind:
{
    Bytel = Wire.read()
}
```

2. Was müssen wir auf dem Max30105 einstellen?

- Welche Einstellungen sollten wir vor einer Messung mindestens festgelegt haben?
 - Das Register zurücksetzen um sicher zu sein, dass in allen Registerplätzen die Anfangswerte stehen (*Registerplatz 0x09*)
 - Den SpO2 Modus aktivieren (*Registerplatz 0x09*)
 - Die rote und infrarote LED mit Strom versorgen (6.4 mA)
(*Registerplatz 0x0C und 0x0D*)
 - Samplerate (*Registerplatz 0x0A*) (400 Sample pro Sekunde)
 - Pulsweite (*Registerplatz 0x0A*) (411µs)

3. Wie erhalten wir Daten vom Max30105?

Welche Daten nehmen wir auf?

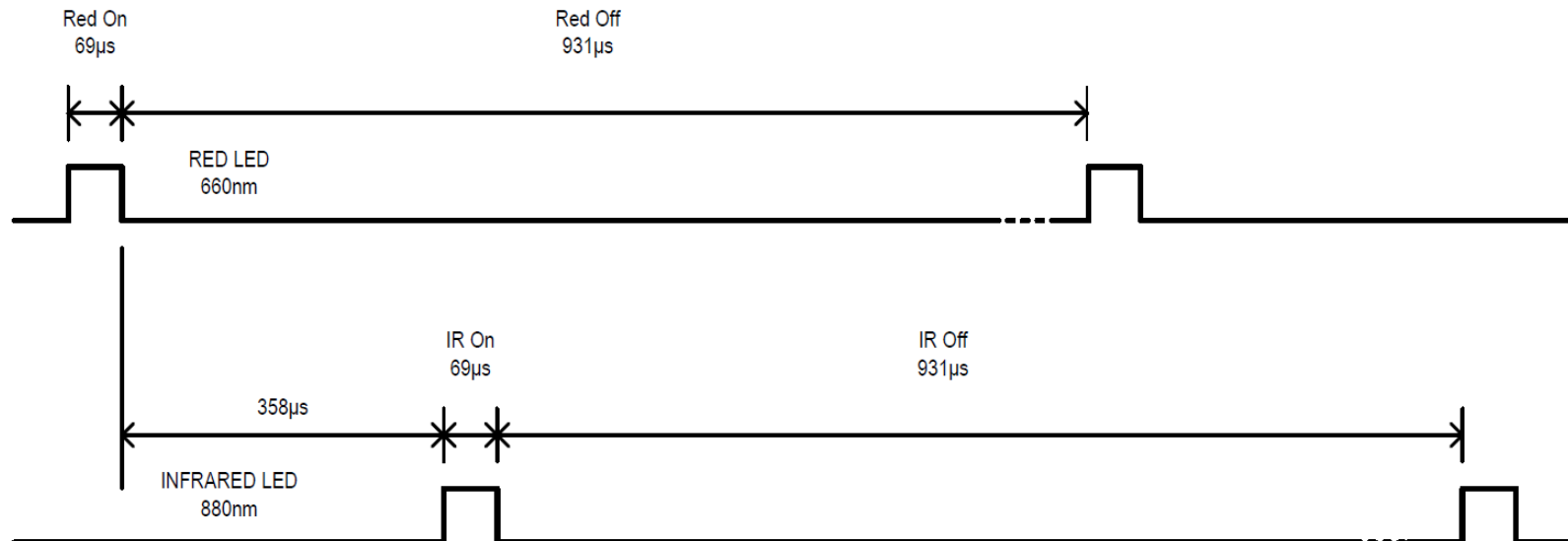
- Licht mittels der Photodiode
- Das Licht wird von den LEDs ausgestrahlt.

Die LEDs werden sequentielle ein-geschaltet.

ADC Integration während der Einschalt-Zeit der jeweiligen LED.

Als Referenz zur Raumlicht-Unterdrückung wird das Signal zwischen den Pulsen verwendet.

Beispiel: SpO₂ Bestimmung mit roter und IR LED und 1 kHz Messrate



3. Wie erhalten wir Daten vom Max30105?

ADC Integrationszeiten: 69 μ s, 118 μ s, 215 μ s oder 411 μ s

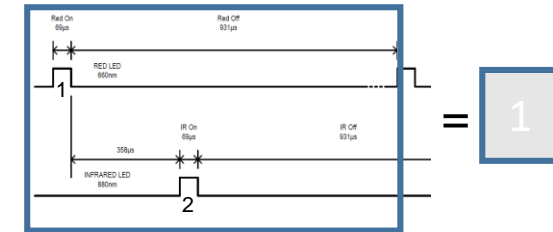
Zeiten zwischen den Einschaltflanken der drei LEDs: 427 μ s, 525 μ s, 720 μ s oder 1107 μ s

Taktraten: 50 Hz, 100 Hz, ... 1000 Hz, 3200 Hz

SpO₂ Modus

SAMPLES PER SECOND	PULSE WIDTH (μ s)			
	69	118	215	411
50	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
100	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
200	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
400	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
800	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
1000	<input type="radio"/>	<input type="radio"/>		
1600	<input type="radio"/>			
3200				

3. Wie erhalten wir Daten vom Max30105?



Bei jeder Aktivierung einer LED misst die Photodiode jeweils einen Wert. Unser Messzyklus besteht aus der Aktivierung der roten und dann der infraroten LED (sowie einer Totzeit zwischen den Aktivierungen). Die Werte der roten und infraroten LED werden zusammen betrachtet als ein Messwert (Sample).

Wo werden die Messwerte gespeichert?

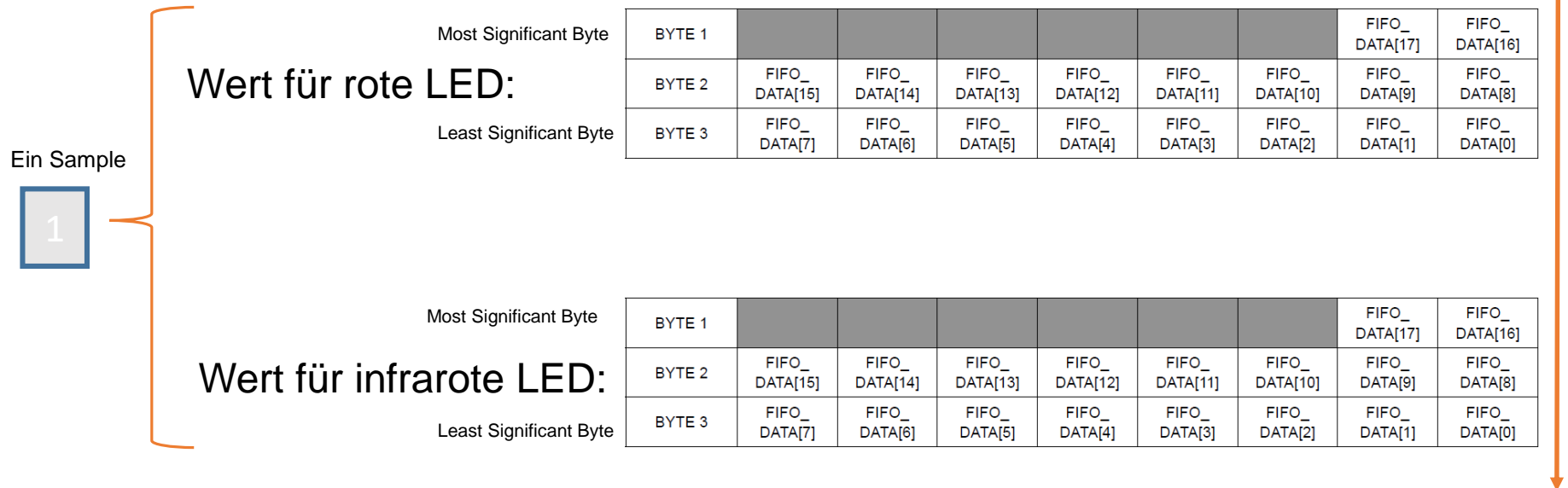
- Im FIFO (First In – First Out)
- In den FIFO passen 32 Messwerte (Samples).

Wenn alle 32 Plätze mit ungelesenen Messwerten gefüllt sind, können keine neuen Werte in den FIFO gespeichert werden.



3. Wie erhalten wir Daten vom Max30105?

Aufbau eines Samples im FIFO:



- In den FIFO passen 32 Samples (ein Samples = 2 Lichtwerte).
Jeder Wert für eine LED besteht aus 3 Bytes.
Die Information für einen Messzyklus setzt aus damit aus 6 Bytes zusammen.

3. Wie erhalten wir Daten vom Max30105?

Sobald ein neuer Messwert bereit ist, schreibt der Max30105 diesen Wert in einen FIFO-Platz.

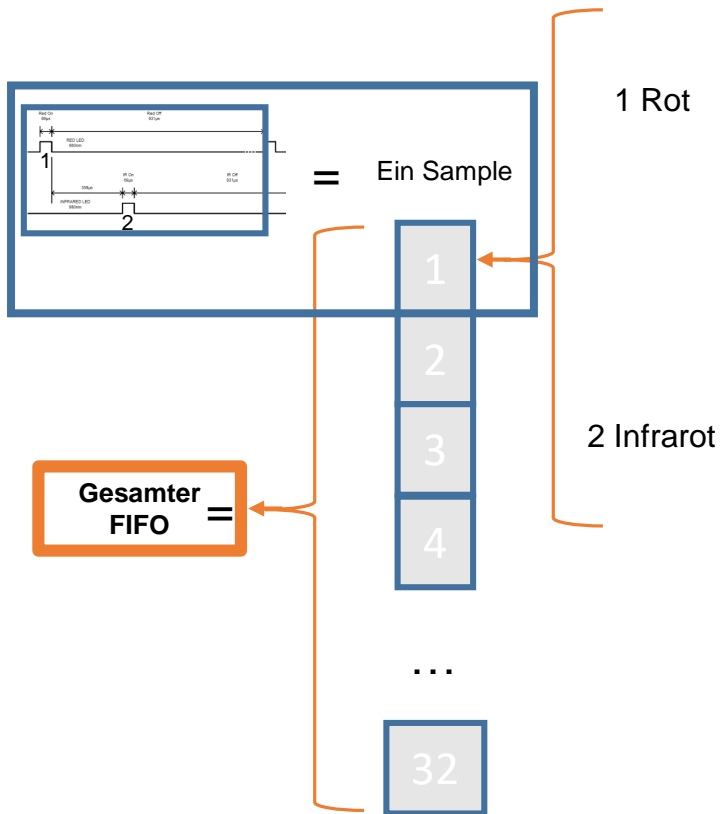
Um Daten aus dem FIFO zu lesen nutzen wir nur eine Registeradresse:

REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W

In diesem Registerplatz steht das nächste ungelesene Byte eines Lichtwertes einer LED.

Wenn das Byte ausgelesen wurde, steht in FIFO_DATA ein anderes/das nächste Byte.

3. Wie erhalten wir Daten vom Max30105? - FIFO Schema



REGISTER	B7	B6	B5	B4	B3	B2	B1	B0	REG ADDR	POR STATE	R/W
FIFO Data Register	FIFO_DATA[7:0]								0x07	0x00	R/W
BYTE 1									FIFO_DATA[17]	FIFO_DATA[16]	
BYTE 2	FIFO_DATA[15]	FIFO_DATA[14]	FIFO_DATA[13]	FIFO_DATA[12]	FIFO_DATA[11]	FIFO_DATA[10]	FIFO_DATA[9]	FIFO_DATA[8]			
BYTE 3	FIFO_DATA[7]	FIFO_DATA[6]	FIFO_DATA[5]	FIFO_DATA[4]	FIFO_DATA[3]	FIFO_DATA[2]	FIFO_DATA[1]	FIFO_DATA[0]			

BYTE 1									FIFO_DATA[17]	FIFO_DATA[16]	
BYTE 2	FIFO_DATA[15]	FIFO_DATA[14]	FIFO_DATA[13]	FIFO_DATA[12]	FIFO_DATA[11]	FIFO_DATA[10]	FIFO_DATA[9]	FIFO_DATA[8]			
BYTE 3	FIFO_DATA[7]	FIFO_DATA[6]	FIFO_DATA[5]	FIFO_DATA[4]	FIFO_DATA[3]	FIFO_DATA[2]	FIFO_DATA[1]	FIFO_DATA[0]			

Wenn das Byte ausgelesen wurde, steht in FIFO_DATA ein anderes/das nächste Byte.



3. Wie erhalten wir Daten vom Max30105?

Es gibt für uns zwei Möglichkeiten den FIFO auszulesen.

- **A.)** Wir lesen immer aus, wenn ein neues Sample da ist.
Wir messen dauerhaft (kontinuierlicher Verlauf) und es bleibt keine Zeit für Berechnungen.

- **B.)** Wir warten bis bspw. 25 Messwerte gemessen wurden, bevor wir auslesen.
Wir können Berechnungen durchführen, erhalten die Messwerte jedoch erst nach einem bestimmten Zeitintervall.

3. Wie erhalten wir Daten vom Max30105?

A.) Wir lesen immer aus, wenn ein neues Sample da ist.

! Den Code hierfür kann man auf dem grundlegenden Interrupt-Code (Übung 2) aufbauen.

+ das Datenblatt runterladen: https://cdn.sparkfun.com/assets/learn_tutorials/5/7/7/MAX30105_3.pdf

Vor dem `setup()`

- Einbinden der Wire Library
`#include <Wire.h>`
- Adresse des Max30105 angeben
`#define max30101 0x57`
- Drei verschiedene Bytes für das Auslesen eines LED-Wertes deklarieren
bspw. `volatile byte` Byte1;
- Variablen für den Lichtwert deklarieren
bspw. `volatile unsigned long int` LED1_Lichtwert;
- Alle anderen Teile für die Interrupt Routine definieren

3. Wie erhalten wir Daten vom Max30105?

A.) Wir lesen immer aus, wenn ein neues Sample da ist.

! Den Code hierfür kann man auf dem grundlegenden Interrupt-Code (Übung 2) aufbauen.

Vor dem `setup()` - Wichtige Begriffe für den Interrupt-Funktion

`void IRAM_ATTR handleInterrupt() {`

- ➔ Wenn die 3 Bytes nacheinander aus dem Register gelesen wurden:
 - sicherstellen, dass das erste Byte (Most Significant Byte) in den ersten Bit-Werten leer ist

BYTE 1							FIFO_DATA[17]	FIFO_DATA[16]
BYTE 2	FIFO_DATA[15]	FIFO_DATA[14]	FIFO_DATA[13]	FIFO_DATA[12]	FIFO_DATA[11]	FIFO_DATA[10]	FIFO_DATA[9]	FIFO_DATA[8]
BYTE 3	FIFO_DATA[7]	FIFO_DATA[6]	FIFO_DATA[5]	FIFO_DATA[4]	FIFO_DATA[3]	FIFO_DATA[2]	FIFO_DATA[1]	FIFO_DATA[0]

Byte1 = Byte1 & 0b00000011;

- Es muss immer ein Sample (6 Bytes) in einem Interrupt ausgelesen werden!
- Bytes mittels Shifting zusammenfügen

LED1_Lichtwert = (((long)Byte3) | (((long)Byte2) << 8) | (((long)Byte1) << 16);

}

3. Wie erhalten wir Daten vom Max30105?

A.) Wir lesen immer aus, wenn ein neues Sample da ist.

! Den Code hierfür kann man auf dem grundlegenden Interrupt-Code (Übung 2) aufbauen.

Im `setup()` {

- **Wire** konfigurieren (siehe Folie 11)
- Max30105 konfigurieren (siehe Folie 18)
 - + Den Interrupt „PPG_RDY_EN im Register 0x02 aktivieren, indem man eine 1 nur dort reinschreibt
 - + Danach im Setup einmal den Interrupt lesen

```
Wire.beginTransaction(max30101);  
Wire.write(0x00);  
Wire.endTransmission();  
Wire.requestFrom(max30101, 2, true);
```

- Wir nutzen hier einen externen Interrupt
(INT-Pin vom Max30105 zum Pin 14 am ESP32)

Code um den Pin und Interrupt zu konfigurieren:

```
pinMode(interruptPin, INPUT_PULLUP);  
attachInterrupt(digitalPinToInterrupt(interruptPin), handleInterrupt, FALLING);  
Wenn das Logik-Level am interruptPin (Pin 14) fällt (FALLING), dann rufe  
die Funktion „handleInterrupt“ auf.
```

}

3. Wie erhalten wir Daten vom Max30105?

A.) Wir lesen immer aus, wenn ein neues Sample da ist.

! Den Code hierfür kann man auf dem grundlegenden Interrupt-Code (Übung 2) aufbauen.

Im loop()

{

Auslesen vom Lichtwert der LEDs wie im Interrupt-Code (Übung 2)

}

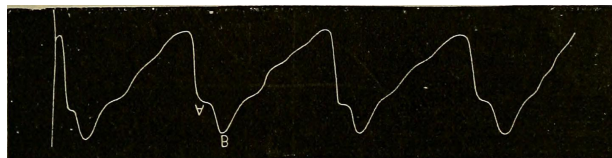
3. Wie erhalten wir Daten vom Max30105?

Aufgabe:

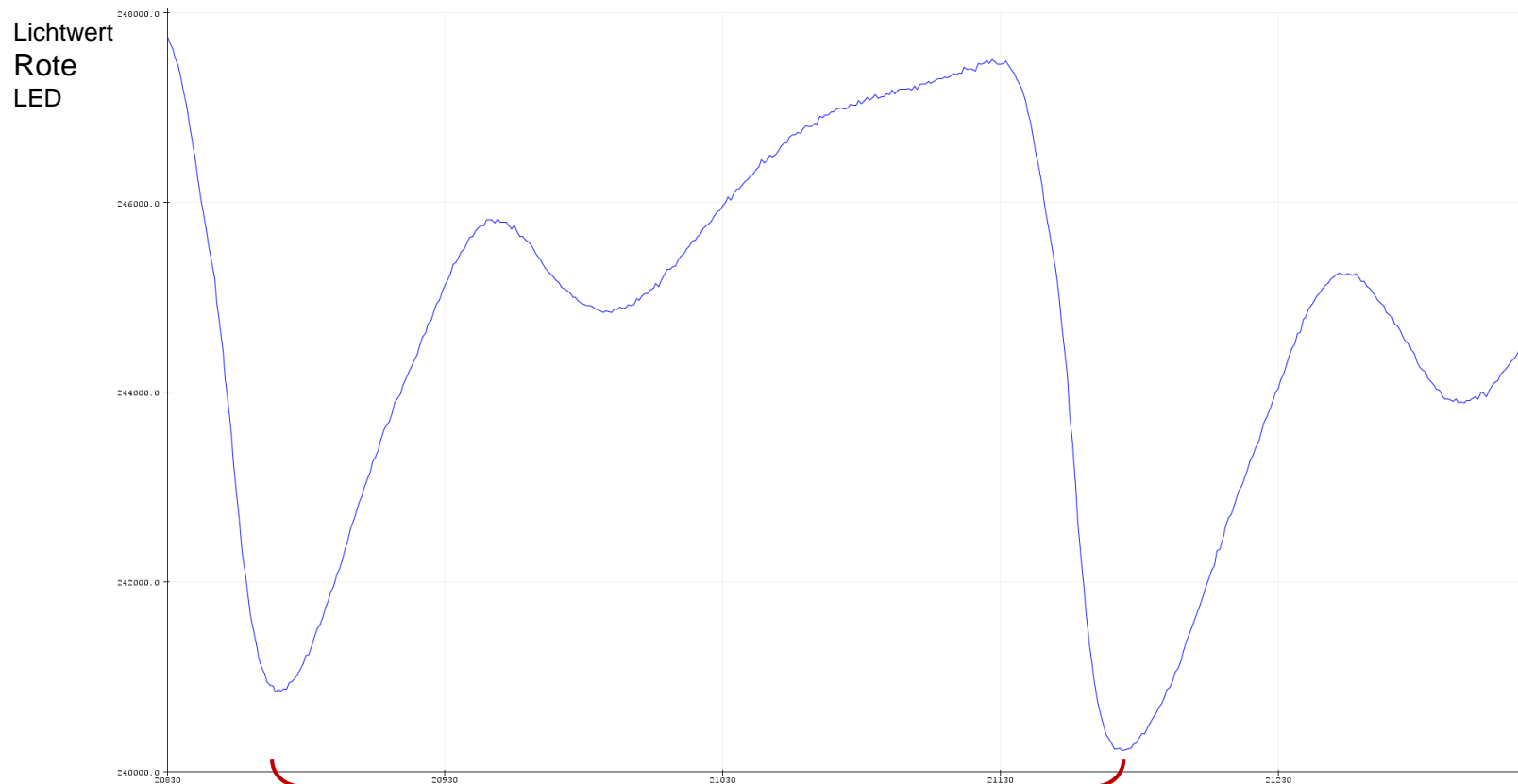
Schreibt aus dem ursprünglichen Interrupt-Code (Übung 2) einen Code, der die Max30105 Messwerte mittels I2C kontinuierlich ausliest!

- a.) Deklariert erst alle notwendigen Variablen für dem **setup**!
- b.) Konfiguriert dann die **Wire**-Bibliothek und den Max30105!
Wenn alles richtig ist, sollte die rote LED sichtbar leuchten.
- c.) Schreibt dann die Interrupt Funktion, in der immer ein Messwert (**Roter und infraroter Lichtwert**) in Byte-Form ausgelesen und in jeweils ein „**long int**“-Wert umgewandelt werden.
Gebt diese beiden Lichtwerte, wie aus dem ursprüngliches Interrupt-Code gewohnt, in der loop()-Funktion aus!

Ziel: In den Rohdaten (Serial.Plotter) soll ein Puls erkennbar sein!



3. Signal vom Max30105



Reaktion auf einen Herzschlag
anhand des wieder
aufgenommenen Lichts



Aus mehreren dieser Reaktionen sollen

- der Puls und
 - die Sauerstoffsättigung
- bestimmt werden.



Methoden:

FFT (Fast Fourier Transformation)



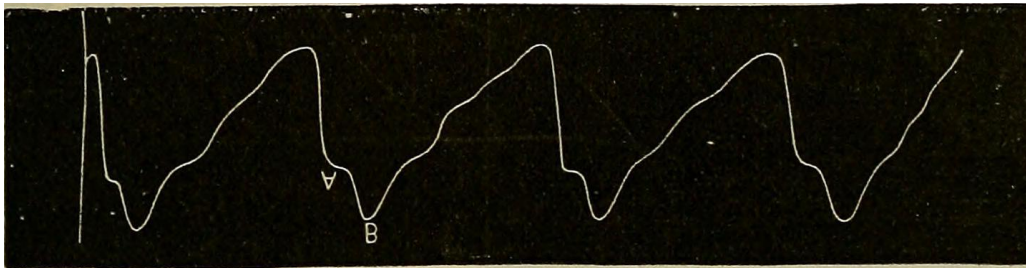
Lambert Beer Ansatz

Bestimmung des Pulses - Was wissen wir über unser Signal?

Der **Puls** ist die mechanische Reaktion der Gefäße im Blutkreislauf auf den Herzschlag.

→ Ein periodischer Verlauf über die Zeit.

Uns interessiert die Wiederholungsrate des Pulses (Frequenz).



„Der Ruhepuls eines Erwachsenen liegt zwischen 60-100 Schläge pro Minute.“ (Folie 4)
= 1 - 1.66 Hz

Wir wollen die Frequenzen in unserem Messsignal mit Hilfe der Fourier-Transformation berechnen.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Anforderungen an unseren Code:

- A. Um die Fourier-Transformationen anzuwenden, brauchen wir zunächst eine Anzahl von Messpunkten (512 Werte), aus denen wir Informationen ziehen können.
- B. Zusätzlich brauchen wir einen ausreichend großen Zeitraum, in dem keine Messwerte an den ESP32 gesendet werden (kein Interrupt), um die Fourier-Transformation zu berechnen.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Umsetzung des Codes:

B. Zusätzlich brauchen wir einen ausreichend großen Zeitraum, in dem keine Messwerte an den ESP32 gesendet werden (kein Interrupt), um die Fourier-Transformation zu berechnen.

- Wir müssen den Interrupt einstellen (`setup()`).
- Die „`handleInterrupt`“-Funktion muss auf darauf angepasst werden,
 - ➔ 32 Samples auf einmal zu lesen und
 - ➔ nur in unser Messwerte-Feld zu schreiben wenn es noch nicht voll mit neuen Messwerten ist. (Counter < 512)

Ergebnis:

Sobald der Max30105 32 Samples gemessen hat, wird ein Interrupt ausgelöst. Diese 32 Werte werden in der „`handleInterrupt`“-Funktion ausgelesen.

Solange noch keine 512 neuen Messwerte im Feld sind, wird das Feld in 32er Schritten aufgefüllt.

Wenn das Feld voll ist, wird die Berechnung durchgeführt.

Anschließend wird das Feld wieder neu voll aufgefüllt.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Umsetzung des Codes:

- A. Zwei „double“-Felder mit der Größe 512 erstellen für Messwerte der roten und infraroten LED.
- B. Interrupt einstellen, indem in der Adresse 0x02 nur das erste Bit auf 1 geschrieben wird (im `setup()`).

In der „`handleInterrupt`“-Funktion die Registeradressen 0x04 (**Write-Zeiger**) und 0x06 (**Read-Zeiger**) auslesen in jeweils eine Integer-Variabel.

Verfügbare Samples = (**Write-Wert** – **Read-Wert**) & `0b00000111`
(Maskierung um unbeschriebene Bit-Werte abzuschneiden)

Lese für jedes **verfügbare Sample** jeweils 6 Bytes aus und füge sie zu zwei Lichtwerten (Rot und Infrarot) zusammen.

Wenn noch nicht 512 neue Messwerte im Feld sind, speichere die beiden neuen Lichtwerte jeweils in ihre Felder. Nutze dafür einen Counter der bis 512 hochzählt.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Umsetzung des Codes:

B. Beende das Auslesen, indem du ein Byte aus der Adresse 0x00 ausliest und dadurch das Interrupt zurücksetzt.

In der `loop()`:

Wenn der Counter gleich 512 ist, lese alle 512 neuen Messwerte aus und setze den Counter zurück.

(Außerhalb von der Interrupt-Counter Abfrage)

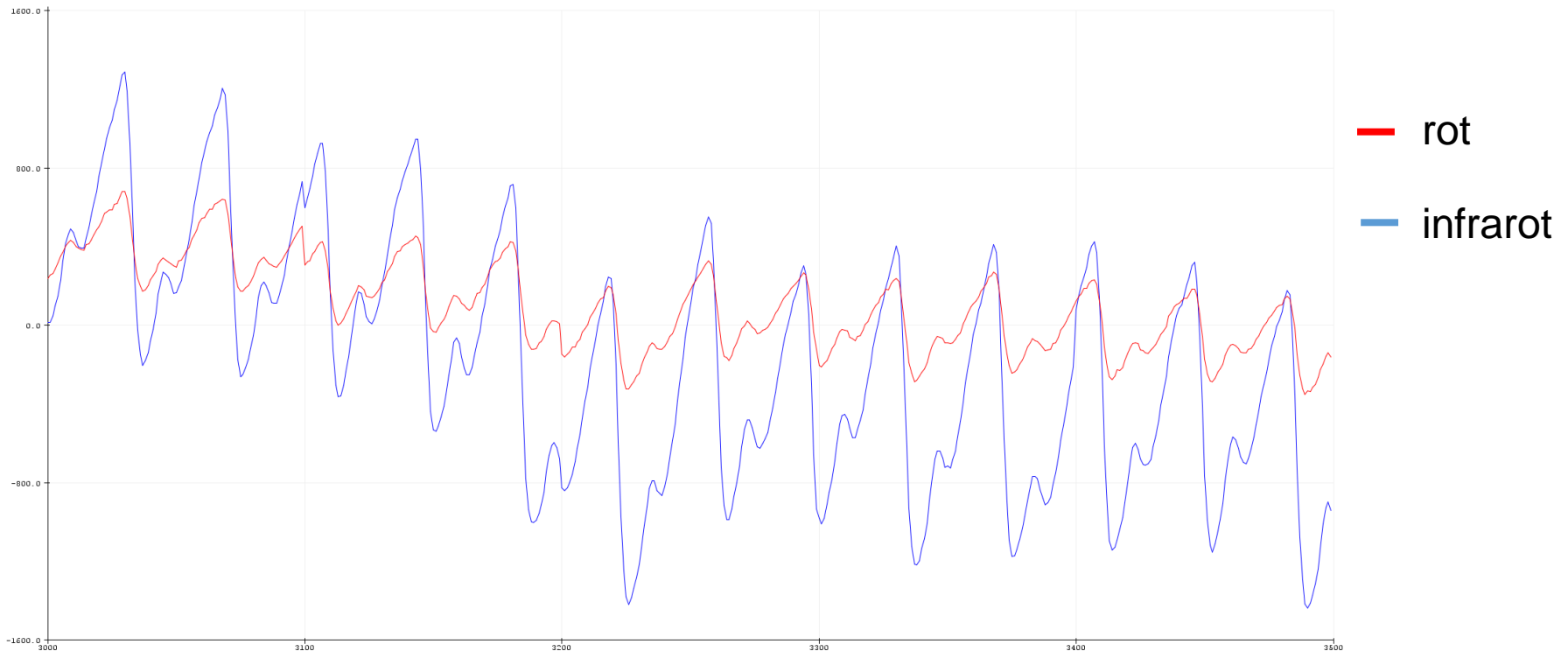
Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Aufgabe:

1. Verändere deinen Code, sodass
der Sensor nur alle 32 Samples einen Interrupt schickt und
diese Samples in ein 512 Sample großes Feld gespeichert werden!
Wenn das Feld voll mit neuen Messdaten ist, gebe alle 512 Daten aus!

Rote oder Infrarote LED?

Die Werte der roten und infraroten LED sind für die gleichen "Sample" dargestellt. Ihr Mittelwert wurde abgezogen.



Die Amplitude der roten LED ist geringer und der Störeinfluss auf die Werte der roten LED ist höher relativ zur infraroten LED.

Deswegen nutzen wir zur Pulsbestimmung die Werte der infraroten LED.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Die Berechnung der Fourier-Transformation übernimmt die Bibliothek **arduinoFFT**.

Um sie zu nutzen binden wir die Bibliothek ein und erstellen ein Objekt.

```
#include "arduinoFFT.h",  
arduinoFFT FFT = arduinoFFT();
```

Anschließend wenden die beiden Befehle auf das Feld der „Infrarot LED“-Daten (irFFTReal) an (samples = Anzahl der Messwerte im Feld).

```
FFT.Compute(irFFTReal, irFFTImag, samples, FFT_FORWARD);  
FFT.ComplexToMagnitude(irFFTReal, irFFTImag, samples);
```

Das Feld der „Infrarot LED“-Daten wird von Befehlen neu befüllt mit den Magnitude-Werten der Fourier-Transformation.

irFFTImag ist hier ein 512 großes Feld voller Nullen.

Vom kontinuierlichen Signal zum Puls / Sauerstoffsättigung

Aufgabe 2.) Umwandlung von Rohdaten in Frequenzspektrum

- a) Gebe die Rohdaten des Max30105 aus, sodass du deinen Puls sehen kannst!
- b) Gebe anschließend das Frequenzspektrum aus!
Probiere einmal eine Ausgabe mit dem nullten Bin und ohne den nullten Bin. (`int i = 0;` vs. `int i = 1;`)
Welche Veränderungen siehst du?

Zusatzinformation:

Wir nutzen die Fast Fourier Transformation (FFT). Die Anzahl der Messwerte muss immer einer Zweierpotenz entsprechen.

Bestimmung des Pulses - FFT-Spektrum



Wodurch wird die Fourier-Transformation beeinflusst?

- Frequenzauflösung $\Delta f_{\text{Fourier}}$:

$$\Delta f_{\text{Fourier}} = \frac{\text{SampleRate}}{\text{Anzahl der Samples im Fenster}} = \frac{50 \frac{\text{Samples}}{\text{Sekunde}}}{512 \text{ Samples}} = 0.0976 \text{ Hz}$$

→ Ein Bin ist 0.0976 Hz breit. Das heißt, wir können nur in 5.856 „Schläge pro Minute“-Schritten den Puls unterscheiden.

- Maximal bestimmbare Frequenz f_{max} :

$$f_{\text{max}} = \frac{1}{2} \cdot \text{SampleRate} = 25 \text{ Hz} = 1500 \text{ Schläge pro Minute}$$

Für uns ist jedoch nicht der gesamte Bereich von 0 bis 1500 Schläge pro Minute wichtig. Schon sowohl 200 Schläge pro Minute als auch und 6 Schläge pro Minute sind schlecht für unser Herz.

In welchem Frequenzbereich sollte also unser Signal maximal/minimal liegen?

Wodurch wird die Fourier-Transformation beeinflusst?

- Annahmen:

→ Minimaler Herzschlag:

$$30 \text{ Schläge pro Minute} = 0.5 \text{ Hz} \approx \text{Bin } 5 \quad (0.5 \text{ Hz / Frequenzauflösung})$$

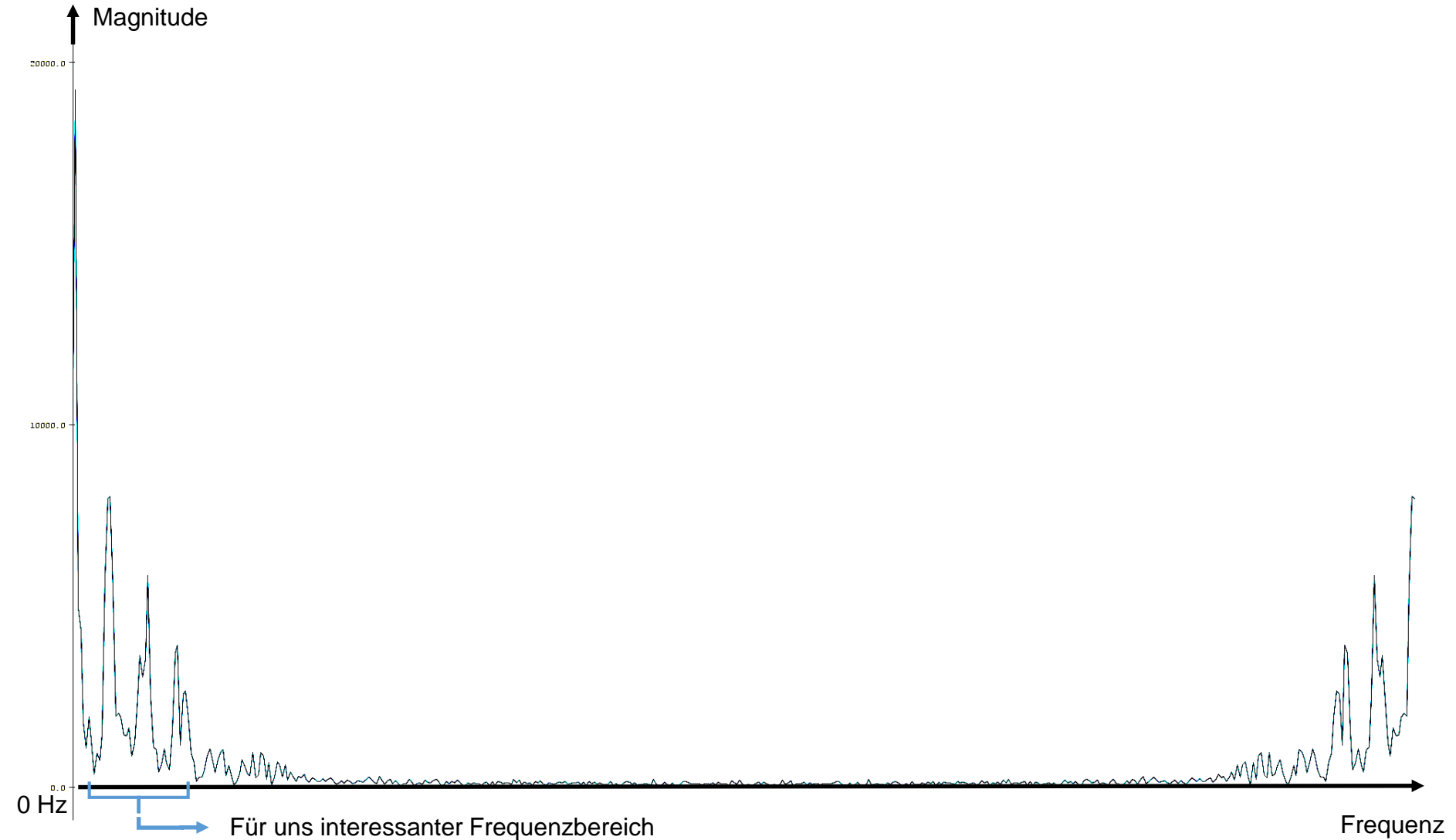
→ Maximaler Herzschlag:

$$180 \text{ Schläge pro Minute} = 3 \text{ Hz} \approx \text{Bin } 31 \quad (3 \text{ Hz / Frequenzauflösung})$$

Unser Pulssignal sollte sich also bezogen auf die Möglichkeiten unseres Körpers zwischen den Frequenzen von Bin 5 bis Bin 31 bewegen.

Im Mustercode wird von Bin 5 bis 31 die Magnitude der FFT verglichen. Es wird der Bin mit der höchsten Magnitude gewählt. Die Frequenz, für die der Bin steht, ist unsere Pulsfrequenz.

Bestimmung des Pulses - FFT-Spektrum



Wodurch wird die Fourier-Transformation beeinflusst?

Aufgabe 3.):

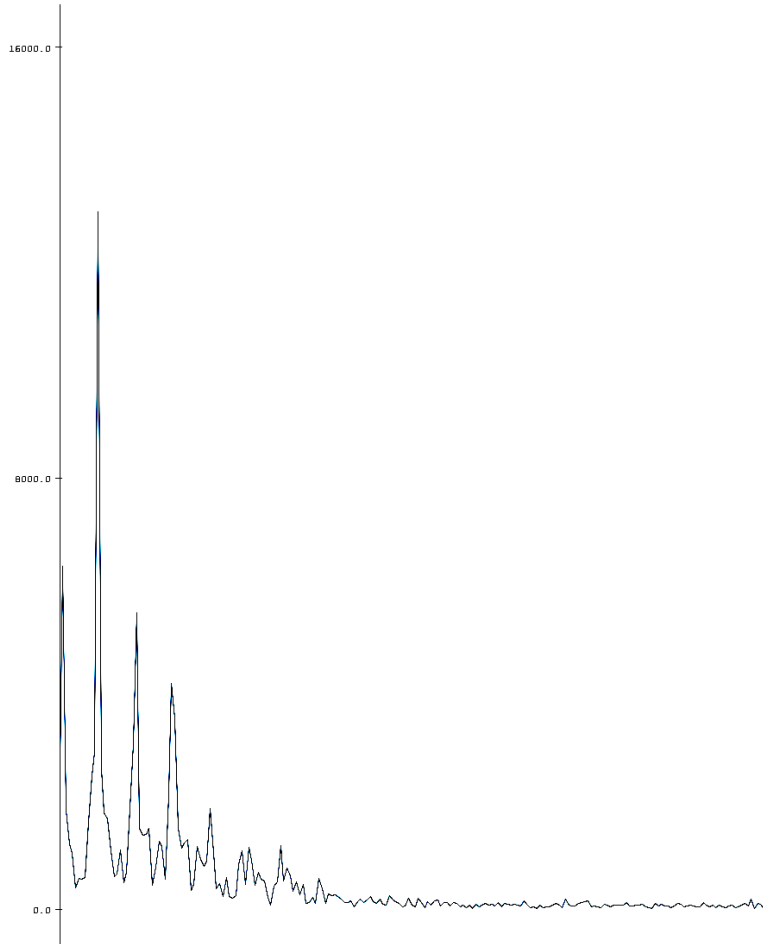
Beispiel von unterschiedlichen Konfigurationen anhand der Samplerate

- a) Erhöhe die Samplerate von 50Hz auf 100Hz und anschließend 200Hz!
Gebe das Frequenzspektrum aus und fertige einen Screenshot davon an.
Wie unterscheiden sich die Frequenzspektren?

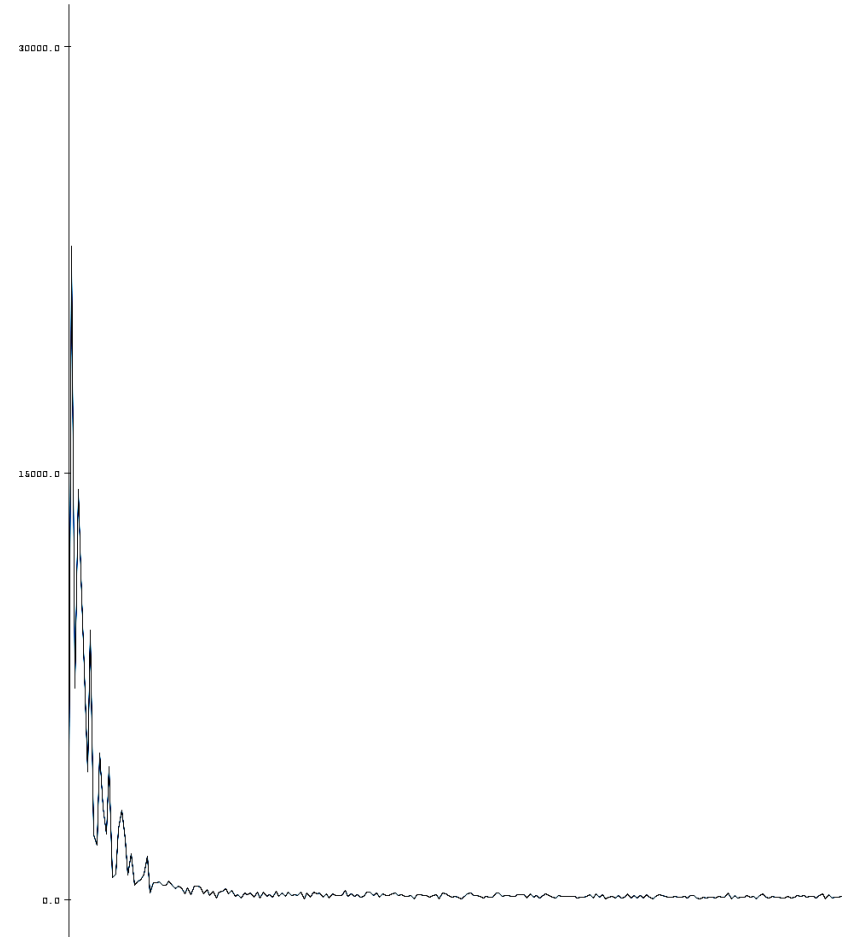
Wodurch wird die Fourier-Transformation beeinflusst?

Ausgabe Bin 1 bis Bin 250

Samplerate 50 Hz



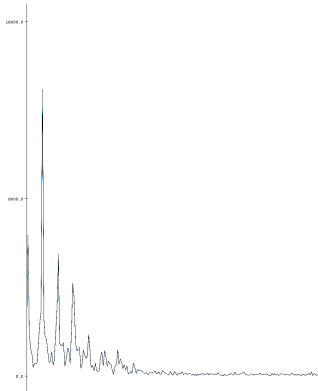
Samplerate 200 Hz



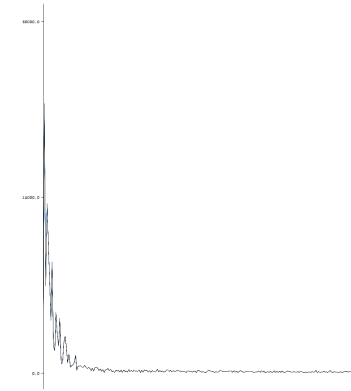
Wodurch wird die Fourier-Transformation beeinflusst?

Ausgabe Bin 1 bis Bin 500

SampleRate 50 Hz



SampleRate 200 Hz



Die Magnitude-Zahl ist höher und die Peaks sind näher beieinander, da die Bin-Breite größer geworden ist. Ein großer Anteil des Signal geht in wenige Bins.

Die Frequenzauflösung verschlechtert sich
wegen der SampleRate

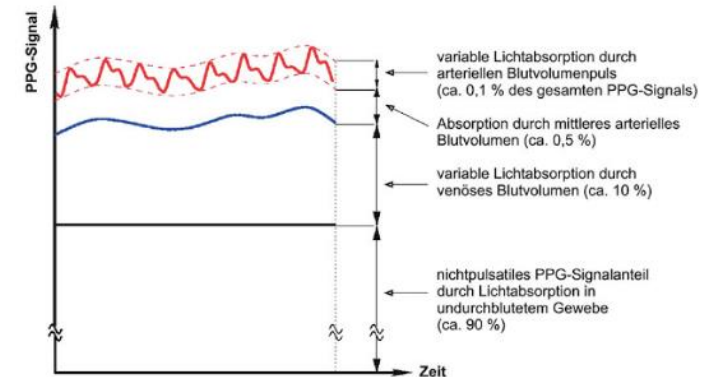
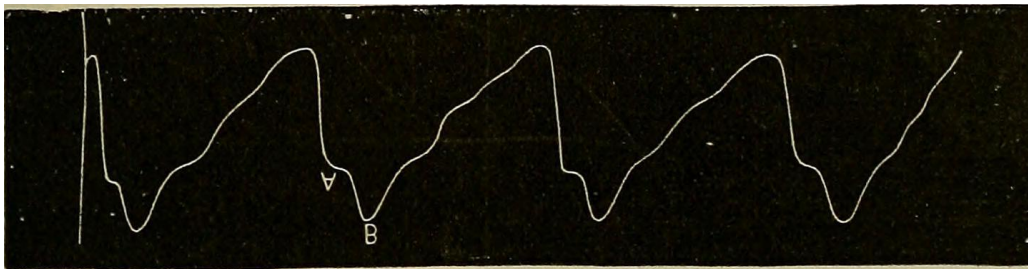
$$\Delta f_{\text{Fourier}} = \frac{\text{SampleRate}}{\text{Anzahl der Samples im Fenster}}$$

Bestimmung der Sauerstoffsättigung - Was wissen wir über unser Signal?

Blut wird aufgrund des Herzschlages in unseren Finger gepumpt. Arterielle Adern dehnen sich aus (mehr Blutvolumen im Finger), während das Gewebe und venöse Adern das vernachlässigbar wenig tun. Mehr Flüssigkeit bedeutet höhere Extinktion des Lichts, wodurch der Photodiodenwert abfällt. Dieses zusätzliche Blut **solte** eine hohe S_aO_2 aufweisen.

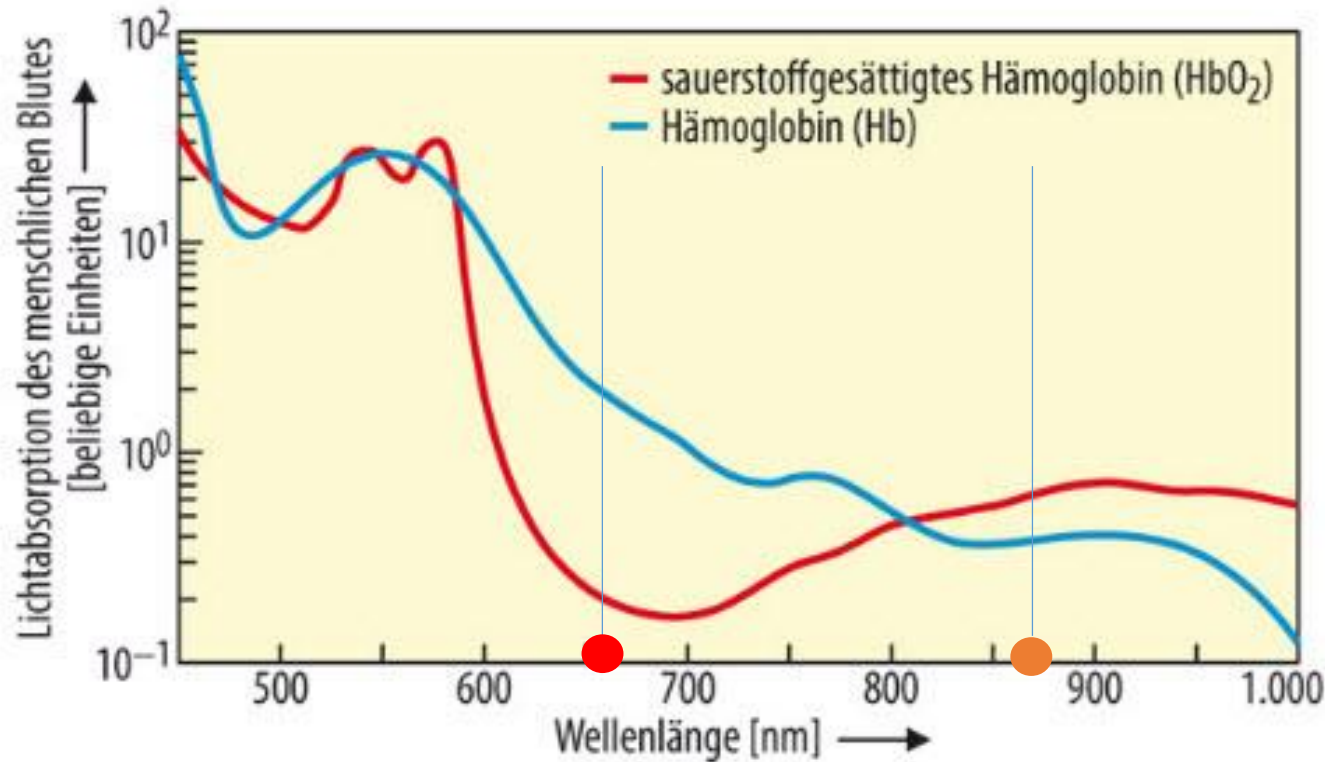
Unsere Frage:

Wie hoch ist die Konzentration des Sauerstoffs in dem Blut, dass in die Arterie gepumpt wird? (S_aO_2)



Bestimmung der Sauerstoffsättigung - Messprinzip

Bildquelle:
https://www.unifr.ch/biochem/assets/files/albrecht/cours/9_1_FunktionHb.pdf



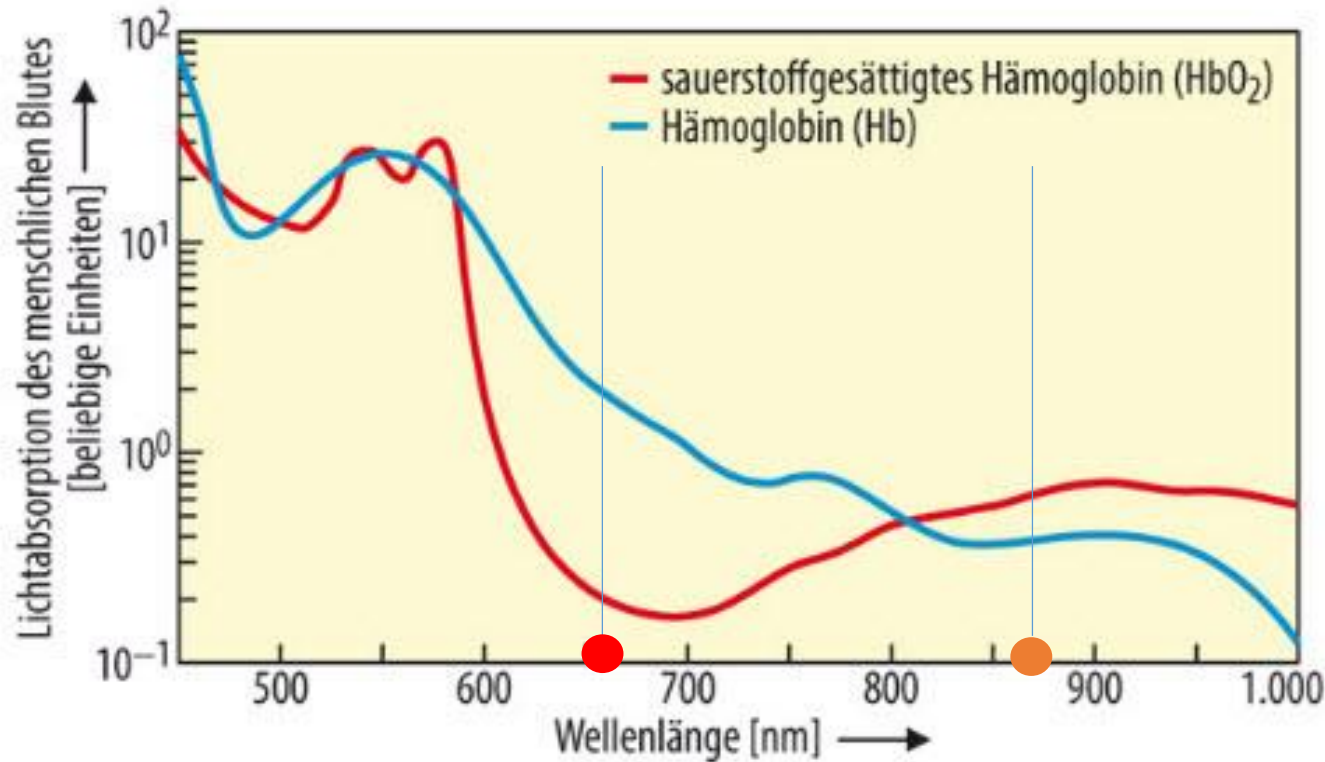
Wellenlänge unserer LEDs:

● Rot = 660nm

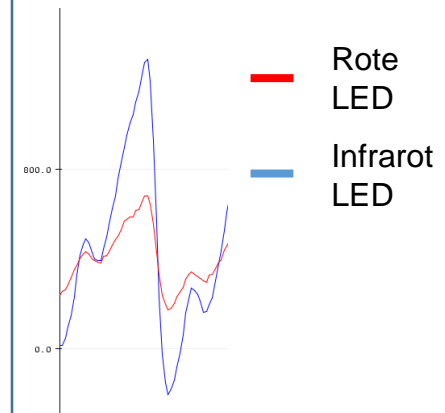
● Infrarot = 880nm

Bestimmung der Sauerstoffsättigung - Messprinzip

Bildquelle:
https://www.unifr.ch/biochem/assets/files/albrecht/cours/9_1_FunktionHb.pdf



Andere Kurven! Wirkliche Lichtwerte:



Wellenlänge unserer LEDs:

- Rot = 660nm
- Infrarot = 880nm

Die Form des Pulses ändert sich nicht zwischen den LEDs. Nur die Amplitude des Pulses ändert sich.

Bestimmung der Sauerstoffsättigung - Formelansatz

Prozentuale, partielle Sauerstoffsättigung im Blut:

$$S_pO_2 = \frac{cHbO_2}{cHbO_2 + cHb} \cdot 100\%$$

Ansatz: Lambert-Beer Gesetz

Die Intensität monochromatischen Lichts nach Durchgang durch ein Medium entspricht

$$I_1 = I_0 \cdot e^{-\varepsilon(\lambda) \cdot c \cdot d}$$

Unsere Annahme:

Die Veränderung des Lichtwertes
ist nur von sauerstoffgesättigtem
und –ungesättigtem Blut abhängig.
(c_{HbO_2} und c_{Hb})

I_0 ... Anfängliche Lichtintensität
 $\varepsilon(\lambda)$... Extinktionskoeffizient abhängig von der Wellenlänge
 c ... Konzentration des Stoffs
 d ... Lichtweg durch das Medium
 I_1 ... Übrig gebliebene Lichtintensität

Bestimmung der Sauerstoffsättigung - Formelansatz

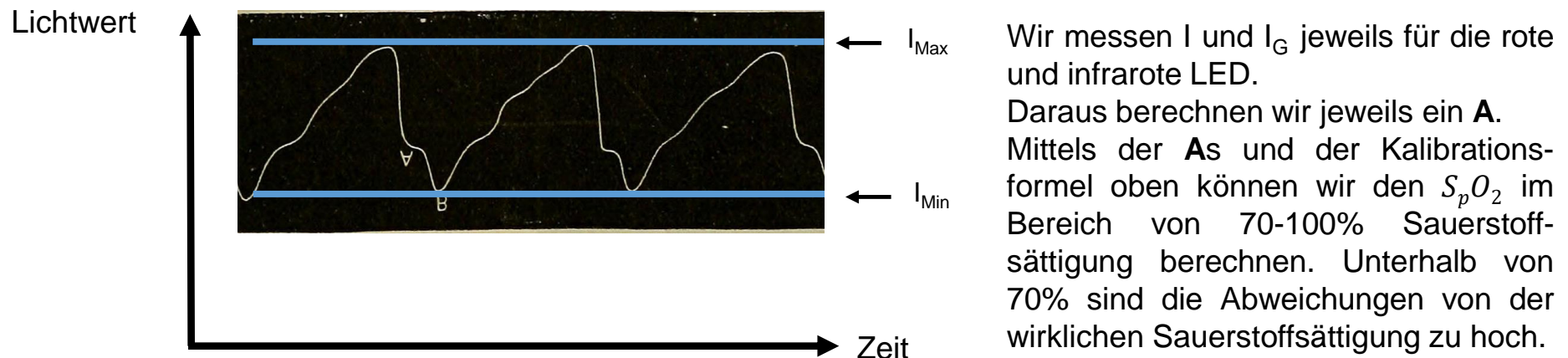
Lösung: Lambert-Beer Gesetz

$$S_p O_2 = 110 - \frac{A(\lambda_{rot})}{A(\lambda_{infrarot})} \cdot 37.5$$

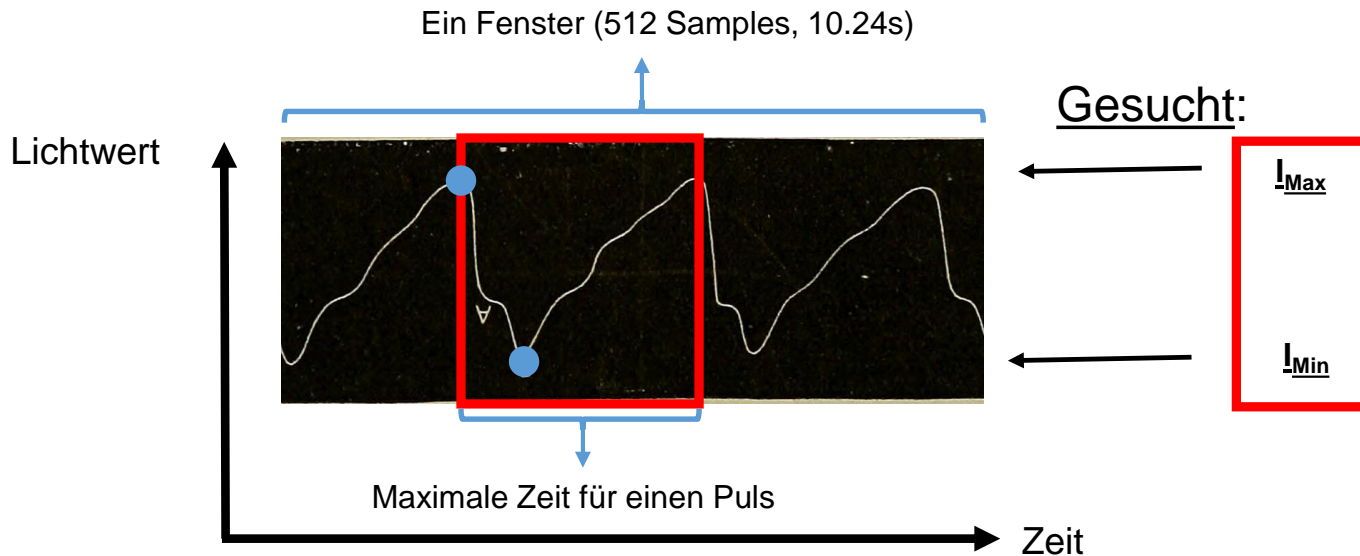
$$A = -\ln\left(\frac{I}{I_G}\right)$$

← I_{Min} ... Minimaler Wert an Intensität
kommt zurück zur Photodiode (Mehr Volumen in Finger)

← I_{Max} ... Maximaler Wert an Intensität
kommt zurück zur Photodiode (Weniger Volumen in Finger)



Bestimmung der Sauerstoffsättigung - Peak to Peak Algorithmus



Um die Sauerstoffsättigung zu berechnen brauchen wir den maximalen und minimalen Intensitätswert für **eine** Pulsperiode.

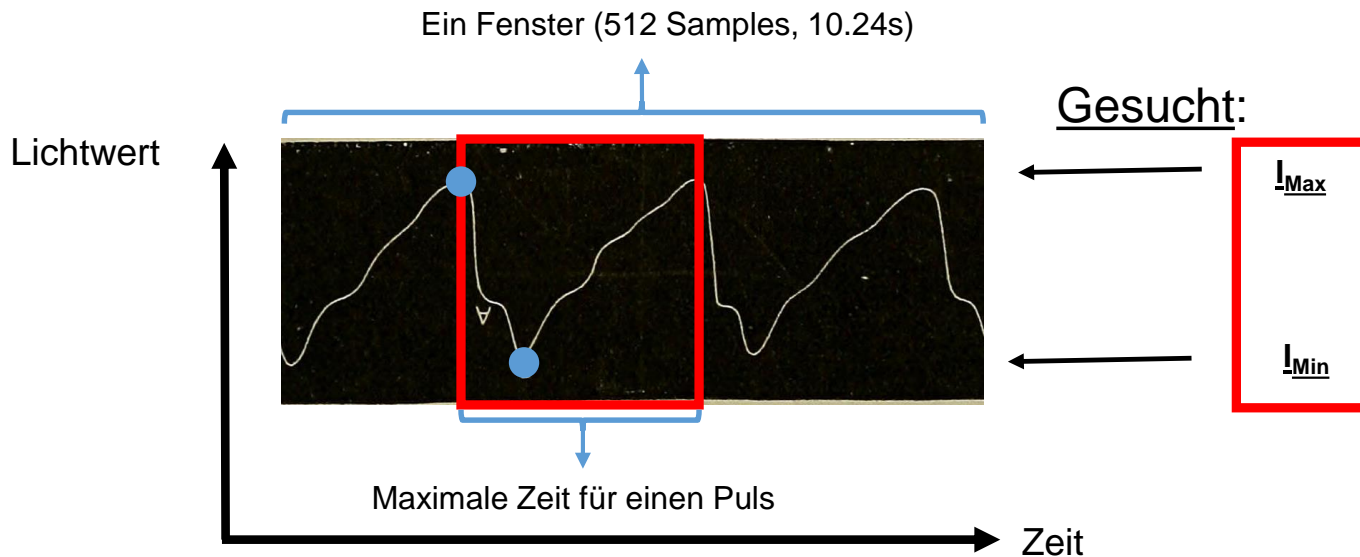
Das Grundniveau des Signals wird über die 10.24s durch Bewegungseinflüsse des Fingers schwanken. Würden wir das absolute Maximum und Minimum des gesamten Fensters nehmen, würden wir keine sinnvollen Werte für S_pO_2 erhalten. Deswegen konzentrieren wir uns auf **einen** Puls.

Wir nehmen an, dass ein Puls maximal 1s dauert (50 Punkte bei 50Hz Samplerate).

Wir suchen das absolute Maximum im gesamten Fenster (= I_{max})

und suchen in den 50 Samples nach I_{max} nach dem minimalen Wert für diesen Puls (= I_{min}).

Bestimmung der Sauerstoffsättigung - Peak to Peak Algorithmus



Vorteil: Geringerer Einfluss von Bewegungsartefakten
im Vergleich zum absoluten Maximum und Minimum über das gesamte Fenster,

Nachteil: Anfällig gegenüber Rauscheinflüssen bei Maxima- und Minimabestimmung,
Die hochfrequenten Peaks durch Umgebungsrauschen werden als Teil der Extrema interpretiert.

Wodurch wird die Fourier-Transformation beeinflusst?

Aufgabe 4.):

Sauerstoffsättigungsberechnung

- a) Nutzt den vorgegebenen Algorithmus und gebt euren Wert der Sauerstoffsättigung aus!

Pfad für die Arduino ESP32 Library

1. Einen Ordner öffnen
2. Strg + L drücken
3. „ /usr/lib64/arduino-esp32-1.0.2/espressif/esp32/tools “ einfügen + Enter
4. „ xtensa-esp32-elf “ – Ordner kopieren
5. Im ESP32 Verzeichnung an gleichem Ort einfügen
(../espressif/esp32/tools)

! Warten bis die Dateien zu Ende kopiert sind (circa 20s) !