





**Internet of Everything-Endgeräte in bestehende Plattform integrieren**

**25.09.2023**

**Verfasst von Timon, Diego, Livio**

Inhaltverzeichnis

[**Aufgabenstellung** 4](#_Toc200608531)

[Einleitung 5](#_Toc200608532)

[Informieren 6](#_Toc200608533)

[Usecase 1: Herzfrequenzmessung mit Anzeige auf einem Display 6](#_Toc200608534)

[Usecase 2: Akustische Ausgabe der Messwerte für bessere Zugänglichkeit 6](#_Toc200608535)

[Warum diese Kombination? 6](#_Toc200608536)

[Planen 7](#_Toc200608537)

[Entscheiden 9](#_Toc200608538)

[1. Relevanz & Nutzen 9](#_Toc200608539)

[2. Barrierefreiheit 9](#_Toc200608540)

[3. Technische Umsetzbarkeit 9](#_Toc200608541)

[Zeitplan 10](#_Toc200608542)

[Tag 1 – Projektstart 10](#_Toc200608543)

[Tag 2 – Planung & erste Tests 10](#_Toc200608544)

[Tag 3 – Sensorik & Datenübertragung 10](#_Toc200608545)

[Tag 4 – Anzeige & Sound 10](#_Toc200608546)

[Tag 5 – Integration & Feinschliff 10](#_Toc200608547)

[Realisieren 11](#_Toc200608548)

[Tag 1 – Vorbereitung und Materialprüfung 11](#_Toc200608549)

[Tag 2 – Planung konkretisieren und technische Tests 11](#_Toc200608550)

[Tag 3 – Sensorik & Datenübertragung 11](#_Toc200608551)

[Tag 4 – Anzeige & akustische Ausgabe 11](#_Toc200608552)

[Tag 5 – Integration und Abschluss 11](#_Toc200608553)

[Kontrollieren 12](#_Toc200608554)

[Funktionskontrolle 12](#_Toc200608555)

[Qualitätskontrolle 12](#_Toc200608556)

[Teamkontrolle & Feedback 12](#_Toc200608557)

[Ergebnis 12](#_Toc200608558)

[Auswerten 13](#_Toc200608559)

[Zielerreichung 13](#_Toc200608560)

[Qualität 13](#_Toc200608561)

[Probleme und Lösungen 13](#_Toc200608562)

[Teamarbeit 13](#_Toc200608563)

[Verbesserungspotenzial 13](#_Toc200608564)

[Installationserklärungen 14](#_Toc200608565)

[Aufsetzen des Geräts 17](#_Toc200608566)

# **Aufgabenstellung**

Im Hinblick auf diese Projektarbeit, in der Sie selbständig eine Aufgabe im eigenen betrieblichen Umfeld lösen, sind verschiedene theoretische Grundlagen erforderlich. In den abgeschlossenen Themenbereichen haben Sie Ihre Kenntnisse zu den Themen Aufbau- und Ablauforganisation sowie zu den Elementen einer Organisation vertiefen können.

Weiterhin haben Sie Ihre Kenntnisse zu den Begriffen Projekt und Prozess durch eigene Recherchen und Abklärungen im Betrieb in verschiedenen Arbeitsaufträgen vertiefen können. Darüber hinaus haben Sie die Methode IPERKA kennengelernt, die ein bewährtes Modell zur systematischen Bearbeitung von Projekten und Aufgaben darstellt. Diese Kenntnisse konnten Sie anhand eines Praxisbeispiels vertiefen. Mit der Methode SMART haben Sie eine Methode kennengelernt, um Ziele für ein Projekt oder eine Arbeit genau zu definieren. Diese Methode konnten Sie anhand von Aufgaben vertiefen. Mit diesen Kenntnissen können Sie einen Arbeitsauftrag im eigenen beruflichen Umfeld selbständig lösen.

# Einleitung

In dieser Projektarbeit entwickelten wir ein eigenständiges, mobiles Puls-Oximeter-System („PulsOxy-Meter“), das die Herzfrequenz (BPM) und Puls-Oxymetrie (SpO₂) misst. Ziel war es, ein benutzerfreundliches Gerät zu schaffen, das sowohl visuell über ein OLED-Display als auch akustisch via Buzzer informiert. Damit wird es besonders zugänglich – nicht nur für sportlich aktive Personen, sondern auch für Menschen mit eingeschränktem Sehvermögen.

Der Schlüssel dieser Lösung ist die Kombination aus praktischer Gesundheitsüberwachung und Barrierefreiheit. Während viele kommerzielle Wearables auf Smartphone-Apps setzen, haben wir uns bewusst für ein autarkes System entschieden – ohne App-Abhängigkeit, direkt und mobil einsetzbar. Damit adressieren wir Nutzer, die eine einfache, unmittelbare Kontrolle ihrer Vitalwerte wünschen.

Technisch basiert unser Projekt auf bewährten Komponenten: dem MAX30100-Sensor für Puls und SpO₂, einem OLED-Display zur Anzeige, einem Buzzer zur akustischen Ausgabe sowie MQTT und Node‑RED für die Datenübertragung und Visualisierung. Die Umsetzung erfolgt in einem kompakten Zeitrahmen von fünf Tagen gemäss der IPERKA-Methode (Informieren, Planen, Entscheiden, Realisieren, Kontrollieren, Auswerten).

Diese Arbeit dokumentiert:

1. Grundlagen und Anwendungsfälle (Use Cases),
2. detaillierte Planung einschliesslich technischer Struktur und Zeitplan,
3. praktische Umsetzung,
4. abschliessende Kontrolle,
5. Auswertung und Reflexion hinsichtlich Zielerreichung, Qualität, Nutzerfreundlichkeit und Teamarbeit.

Damit demonstrieren wir, dass wir in der Lage sind, selbstständig eine praxisnahe Aufgabenstellung im betrieblichen Umfeld zu lösen – mit methodischer Struktur, technischer Kompetenz und klarer Orientierung an Benutzerbedürfnissen.

# Informieren

Für unser IoT-Projekt haben wir uns entschieden, die Herzfrequenz eines Nutzers zu messen und diese Informationen auf einfache und zugängliche Weise darzustellen. Unsere Motivation basiert auf dem Wunsch, die Gesundheit besser überwachen zu können – speziell die Herzaktivität, die ein wichtiger Indikator für das Wohlbefinden ist. Herzfrequenzmessungen sind heutzutage vor allem durch Fitness-Tracker und Smartwatches verbreitet, jedoch wollten wir ein eigenständiges, leicht verständliches Gerät entwickeln, das diese Messwerte nicht nur anzeigt, sondern auch akustisch ausgibt.

## Usecase 1: Herzfrequenzmessung mit Anzeige auf einem Display

Im ersten Usecase liegt der Fokus darauf, die Herzfrequenz in Schlägen pro Minute (BPM) zuverlässig zu messen und auf einem kleinen Bildschirm anzuzeigen. Die Messung erfolgt mittels eines optischen Sensors (MAX30100), der Puls- und Sauerstoffwerte misst. Das Display gibt die aktuelle Herzfrequenz in Echtzeit aus, sodass Nutzer jederzeit ihre Herzaktivität einfach beobachten können. Diese visuelle Darstellung ist für Personen geeignet, die ihre Herzwerte regelmässig kontrollieren möchten, zum Beispiel beim Sport, zur Gesundheitsvorsorge oder einfach zur Selbstüberwachung.

## Usecase 2: Akustische Ausgabe der Messwerte für bessere Zugänglichkeit

Beim zweiten Usecase haben wir überlegt, wie wir die Messwerte auch für Menschen zugänglich machen können, die visuelle Informationen schlecht wahrnehmen oder Schwierigkeiten beim Ablesen eines kleinen Displays haben. Hierfür haben wir einen Mini-Lautsprecher integriert, der die Herzfrequenz und andere wichtige Gesundheitsinformationen akustisch ausgibt. Diese Funktion ermöglicht es auch sehbehinderten oder älteren Menschen, ohne Hindernisse über ihren Gesundheitszustand informiert zu sein. Die akustische Ausgabe sorgt für Barrierefreiheit und erweitert den Nutzen unseres Geräts deutlich.

## Warum diese Kombination?

Wir wollten ein Gerät schaffen, das sowohl visuell als auch auditiv funktioniert, um möglichst vielen Menschen eine einfache und verständliche Gesundheitsüberwachung zu ermöglichen. Dabei soll das System leicht zu bedienen und mobil sein, ohne aufwendige Installation oder komplizierte Einstellungen. Die Verbindung aus Messung, Anzeige und akustischer Rückmeldung macht das Gerät zu einem flexiblen und nützlichen Werkzeug im Alltag.

# Planen

**Ziel:**

Einen funktionierenden PulsOxy-Meter in 5 Tagen umzusetzen, mit klarer Aufgabenverteilung, technischer Struktur und funktionsfähiger Endpräsentation. Die Zeit ist knapp, daher müssen Planung, Umsetzung und Tests effizient koordiniert werden. Die Dokumentation wird parallel zur technischen Umsetzung geführt, um Zeit zu sparen.

**Aufgabenverteilung:**

* **Diego:** Sensorik mit MAX30100, MQTT-Senden. Er übernimmt die Programmierung des Sensors und sorgt dafür, dass valide Werte gesendet werden.
* **Livio:** Anzeige mit Display, Buzzer, MQTT-Empfang. Er kümmert sich um die grafische Darstellung und die akustische Rückmeldung.
* **Timon:** Node-RED Flow, Visualisierung, Steuerlogik. Er entwickelt die Datenverarbeitung und -steuerung sowie das Dashboard zur Visualisierung.

**Grobzeitplan (5 Tage):**

* **Tag 1:** Material organisiert, Auftrag gelesen, Komponenten überprüft
* **Tag 2:** Projekt planen, Verantwortlichkeiten klären, technische Recherchen, Sensor ansteuern
* **Tag 3:** Sensorik-Programmierung abschliessen, MQTT-Datenübertragung testen, erste Integration in Node-RED
* **Tag 4:** Anzeigelogik und Displayausgabe umsetzen, MQTT-Empfang und Buzzer testen, Verfeinerung des Node-RED Flows
* **Tag 5:** Gesamtsystem zusammenführen und testen, Troubleshooting, Projektdokumentation und Präsentation vorbereiten

**Technische Planung:**

* **Ansteuerung Display:** Grafiklinie für Pulsverlauf, numerische Anzeige für SpO2
* **Soundlogik:** Node-RED analysiert die Frequenz und sendet entsprechende Signale zum Buzzer

**Ressourcen:**

* Entwicklungsumgebung: Arduino IDE (mit passenden Bibliotheken)
* Kommunikation: MQTT über lokalen Broker (Mosquitto)
* Visualisierung: Node-RED Dashboard mit Live-Grafik und Steuerung
* Dokumentation: Word und Screenshot-Sammlungen aus Arduino/Node-RED

**Risikoanalyse und Vorsorge:**

* **Risiko:** MAX30100 liefert keine stabilen Werte → Alternative Messdauer oder Mittelwertbildung
* **Risiko:** MQTT-Verbindung instabil → LAN-Testumgebung, KeepAlive und Error Handling
* **Risiko:** Kompatibilität Displaybibliothek → Test mit einfacher Textanzeige vor Grafik

# Entscheiden

Bei der Planung unseres IoT-Projekts haben wir mehrere Ideen diskutiert, uns aber gezielt für die Kombination aus Herzfrequenzmessung, Display-Anzeige und akustischer Ausgabe entschieden. Diese Entscheidung basiert auf drei zentralen Überlegungen:

## 1. Relevanz & Nutzen

**Herzfrequenz** und **Sauerstoffsättigung** sind wichtige Gesundheitsparameter. Viele Menschen – ob beim Sport, im Alltag oder im Alter – profitieren davon, ihre Vitalwerte einfach überwachen zu können. Ein kompaktes, eigenständiges Gerät ohne App-Zwang ist dabei besonders attraktiv.

## 2. Barrierefreiheit

Die Kombination aus **visueller** und **akustischer** Rückmeldung war uns wichtig. Während das Display eine direkte Anzeige für Sehende bietet, stellt der Lautsprecher sicher, dass auch Menschen mit eingeschränktem Sehvermögen ihre Werte wahrnehmen können.

## 3. Technische Umsetzbarkeit

Die eingesetzten Komponenten – MAX30100, OLED-Display, Buzzer, MQTT, Node-RED – sind gut dokumentiert und miteinander kompatibel. Dadurch war eine **schnelle Entwicklung in nur fünf Tagen** realistisch und sinnvoll.

# Zeitplan

## Tag 1 – Projektstart

Material organisieren

Projektauftrag verstehen und offene Fragen klären

Komponenten überprüfen und testen, ob alles funktionsfähig ist

## Tag 2 – Planung & erste Tests

Aufgabenverteilung final abstimmen

Technische Recherche (Sensor, Display, MQTT, Node-RED)

Erste Tests mit dem MAX30100-Sensor zur Pulserkennung

## Tag 3 – Sensorik & Datenübertragung

Sensor vollständig programmieren und Kalibrierung durchführen

MQTT-Datenübertragung aufsetzen und testen

Node-RED vorbereiten und erste Daten empfangen

## Tag 4 – Anzeige & Sound

Displayansteuerung mit BPM-Ausgabe umsetzen

Akustische Ausgabe über Buzzer programmieren

Node-RED Dashboard mit ersten Visualisierungen verbinden

## Tag 5 – Integration & Feinschliff

Gesamtsystem zusammenführen und ausführlich testen

Fehlerbehebung, Optimierungen

Projektdokumentation abschliessen und Präsentation vorbereiten

# Realisieren

In der Umsetzungsphase wurde das Projekt Schritt für Schritt entsprechend dem zuvor erarbeiteten Zeitplan realisiert. Die Zusammenarbeit im Team erfolgte effizient und zielorientiert. Jeder Teilnehmende übernahm seine definierte Rolle und trug massgeblich zum Fortschritt bei. Die Kommunikation im Team lief reibungslos, sodass Herausforderungen schnell erkannt und gemeinsam gelöst werden konnten.

## Tag 1 – Vorbereitung und Materialprüfung

Alle benötigten Komponenten wurden beschafft und auf ihre Funktionalität getestet. Dabei zeigte sich, dass alle Teile einsatzbereit waren. Erste Ideen zur konkreten Umsetzung wurden besprochen, um ein gemeinsames Verständnis für das Projekt zu schaffen.

## Tag 2 – Planung konkretisieren und technische Tests

Die Aufgabenverteilung wurde bestätigt und technische Informationen zu den verwendeten Bauteilen eingeholt. Erste Testläufe mit dem MAX30100-Sensor zeigten Schwankungen, die durch Kalibrierung und angepasste Abtastraten später ausgeglichen werden konnten.

## Tag 3 – Sensorik & Datenübertragung

Die Ansteuerung des Sensors und das Auslesen der Messwerte wurden erfolgreich implementiert. Über MQTT konnten erste Werte an den Broker gesendet werden. Parallel dazu wurde in Node-RED ein einfacher Flow erstellt, der die eingehenden Daten sichtbar machte.

## Tag 4 – Anzeige & akustische Ausgabe

Das OLED-Display wurde in das Projekt integriert. Die aktuelle Herzfrequenz wurde numerisch sowie grafisch dargestellt. Zudem wurde ein Buzzer so programmiert, dass dieser in bestimmten Abständen einen Ton ausgab – abhängig von der aktuellen Herzfrequenz. Damit war auch die Barrierefreiheit sichergestellt.

## Tag 5 – Integration und Abschluss

Am letzten Tag wurden alle Einzelsysteme zu einem funktionierenden Gesamtsystem verbunden. Es folgten zahlreiche Tests zur Stabilität und Zuverlässigkeit. Die Darstellung im Node-RED Dashboard wurde finalisiert, ebenso wie die Dokumentation des Projekts mit Screenshots und Codeauszügen. Abschliessend wurde eine kurze Präsentation zur Vorstellung des Projekts vorbereitet.

# Kontrollieren

Die Kontrollphase diente dazu, die Funktionalität und Qualität unseres Projekts systematisch zu überprüfen. Ziel war es, sicherzustellen, dass alle Komponenten wie geplant zusammenarbeiten und das Gerät die Anforderungen erfüllt, die wir in der Planungs- und Entscheidungsphase definiert hatten.

## Funktionskontrolle

Nach der technischen Integration aller Module (Sensor, Display, Lautsprecher, MQTT-Anbindung, Node-RED) führten wir gezielte Funktionstests durch. Dabei wurde überprüft:

ob der MAX30100 verlässlich Puls- und SpO₂-Werte misst,

ob die Daten korrekt über MQTT gesendet und empfangen werden,

ob die Anzeige auf dem OLED-Display aktuell und klar lesbar ist,

ob der Buzzer bei definierter Frequenz korrekt auslöst,

ob die Daten im Node-RED-Dashboard in Echtzeit visualisiert werden.

## Qualitätskontrolle

Neben der reinen Funktionsprüfung legten wir Wert auf eine stabile Datenübertragung, eine intuitive Bedienbarkeit und eine barrierefreie Ausgabe der Messwerte. Hierzu simulierten wir unterschiedliche Anwendungssituationen (z. B. schnelle Herzfrequenzwechsel, Displayausfall, MQTT-Verbindungsunterbrechung) und beobachteten die Reaktion des Systems.

## Teamkontrolle & Feedback

Die Überprüfung erfolgte im Team. Jedes Mitglied testete auch die Arbeit der anderen, um Fehlerquellen objektiv zu erkennen. Dabei nutzten wir eine einfache Checkliste mit den wichtigsten Prüfpunkten. Auffälligkeiten wurden gemeinsam diskutiert und korrigiert. Durch das Vier-Augen-Prinzip konnten kleine Schwächen rechtzeitig erkannt und behoben werden.

## Ergebnis

Am Ende der Kontrollphase funktionierte das Gesamtsystem wie gewünscht: Die Messdaten wurden korrekt verarbeitet, angezeigt und akustisch wiedergegeben. Das System reagierte zuverlässig auf Veränderungen und war benutzerfreundlich. Damit konnten wir die Funktionstüchtigkeit unseres Projekts bestätigen.

# Auswerten

Nach Abschluss der Umsetzung und Inbetriebnahme des PulsOxy-Meters haben wir unser Projekt systematisch ausgewertet. Dabei wurden die Zielerreichung, die Funktionalität, die Benutzerfreundlichkeit sowie die Teamarbeit beurteilt.

## Zielerreichung

Unser Hauptziel war es, ein funktionierendes Puls- und Oximeter-System zu entwickeln, das Herzfrequenz und Sauerstoffsättigung misst, diese Daten an einen MQTT-Broker sendet und sie grafisch sowie akustisch visualisiert. Dieses Ziel konnten wir vollständig erreichen:

* Der MAX30100-Sensor misst zuverlässig Puls und SpO₂.
* Die Werte werden nach Tastendruck korrekt über MQTT gesendet.
* Die Anzeige visualisiert die Daten wie geplant: Herzfrequenz als Kurvengrafik, Sauerstoffsättigung als Zahlenwert.
* Der Buzzer reagiert korrekt auf den Herzschlag oder sendet bei zu niedriger Frequenz einen Dauerton.

## Qualität

Die Hardware war stabil aufgebaut, die Steckverbindungen waren ordentlich verlötet oder gesteckt. Die Software funktionierte zuverlässig, der Node-Red Flow konnte problemlos mit dem MQTT-Broker und den ESP8266-Modulen kommunizieren. Auch die Einstellung der Messdauer im Flow war nützlich und funktionierte.

## Probleme und Lösungen

Einige Herausforderungen traten während der Projektarbeit auf:

* **Sensorinitialisierung**: Der MAX30100 wollte anfangs nicht korrekt starten – wir lösten das durch eine korrekte Spannungsversorgung und Anpassungen in der Bibliothek.
* **Verbindungsaussetzer** beim MQTT-Broker konnten durch stabilere WLAN-Einstellungen im ESP-Code minimiert werden.
* **Anzeigefehler** auf dem Display traten durch eine falsche Initialisierung der Displaybibliothek auf – das wurde durch korrektes Timing beim Setup gelöst.

## Teamarbeit

Die Zusammenarbeit in der Gruppe war insgesamt sehr gut. Die Aufgaben wurden sinnvoll aufgeteilt: Eine Person übernahm den Node-Red Flow, eine andere die Anzeige- und Sensoreinheit, die dritte kümmerte sich um Tests und Integration. Wir hielten regelmässig Rücksprache, um sicherzustellen, dass die Module kompatibel bleiben.

## Verbesserungspotenzial

Das Gehäuse könnte verbessert werden, um das System robuster und transportabler zu machen. Da die Hardware ziemlich schwach ist, können wir nicht genauere Werte auslesen.

# Installationserklärungen

(Das Projekt lauft nur wenn eine VM mit Red Node verbunden ist.)

Node Red

1. VM mit Ubuntu aufsetzen (Standardinstall mit GUI)

2. Node-Red installieren: <https://nodered.org/docs/getting-started/raspberrypi>

bash <(curl -sL <https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered>)

3. Node-Red Dashboard installieren: (wenn gewünscht)

cd ~/.node-red

npm install @flowfuse/node-red-dashboard

Node-Red Neustart:

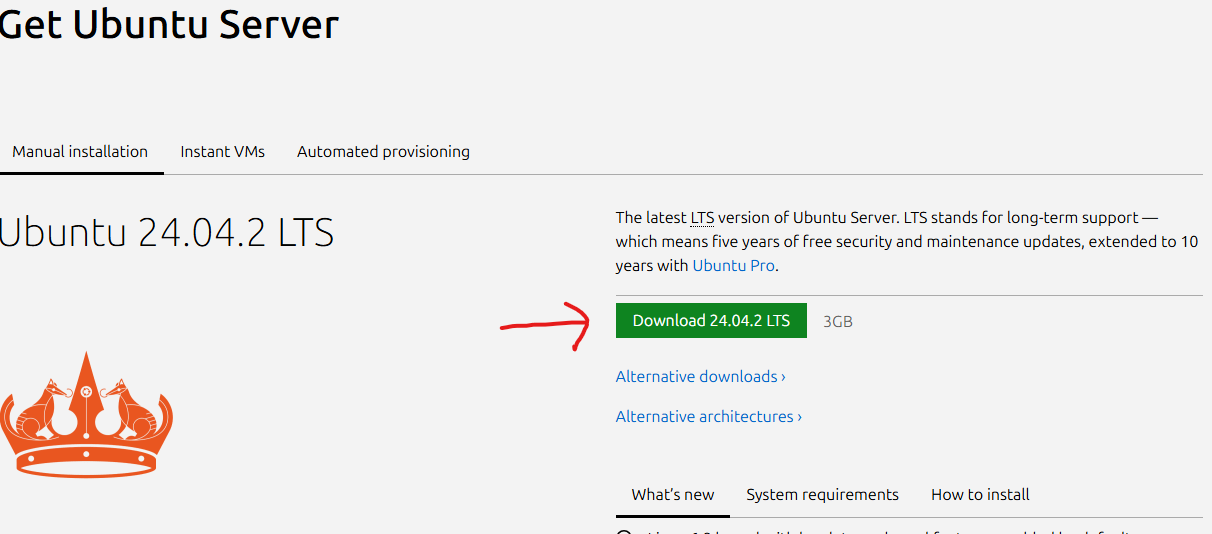
node-red-restart

Node-Red im Browser öffnen: Editor: http localhost:1880 (oder allenfalls die IP statt localhost) Dashboard2: <http://localhost:1880/dashboard/page1> (page1 ist default)

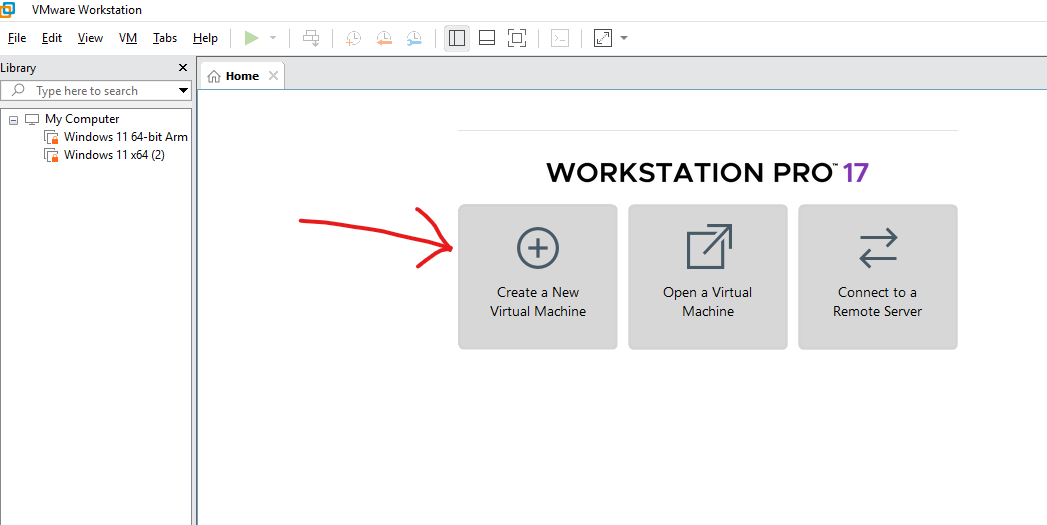
Mehr Infos online:

<https://nodered.org/docs/getting-started/>

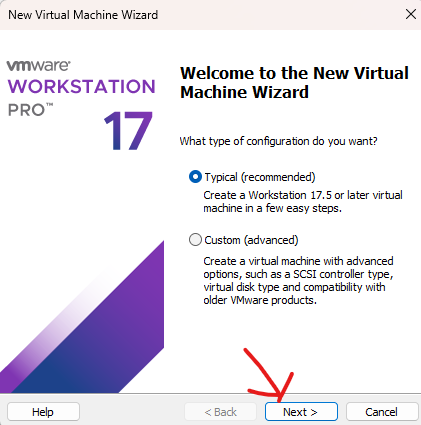
VM aufsetzen mit Ubuntu



Dowload die neuste Ubuntu server versionäpö



Mit den Standardeinstellungen die Daten einfügen



Dem Installationstutorial folgen

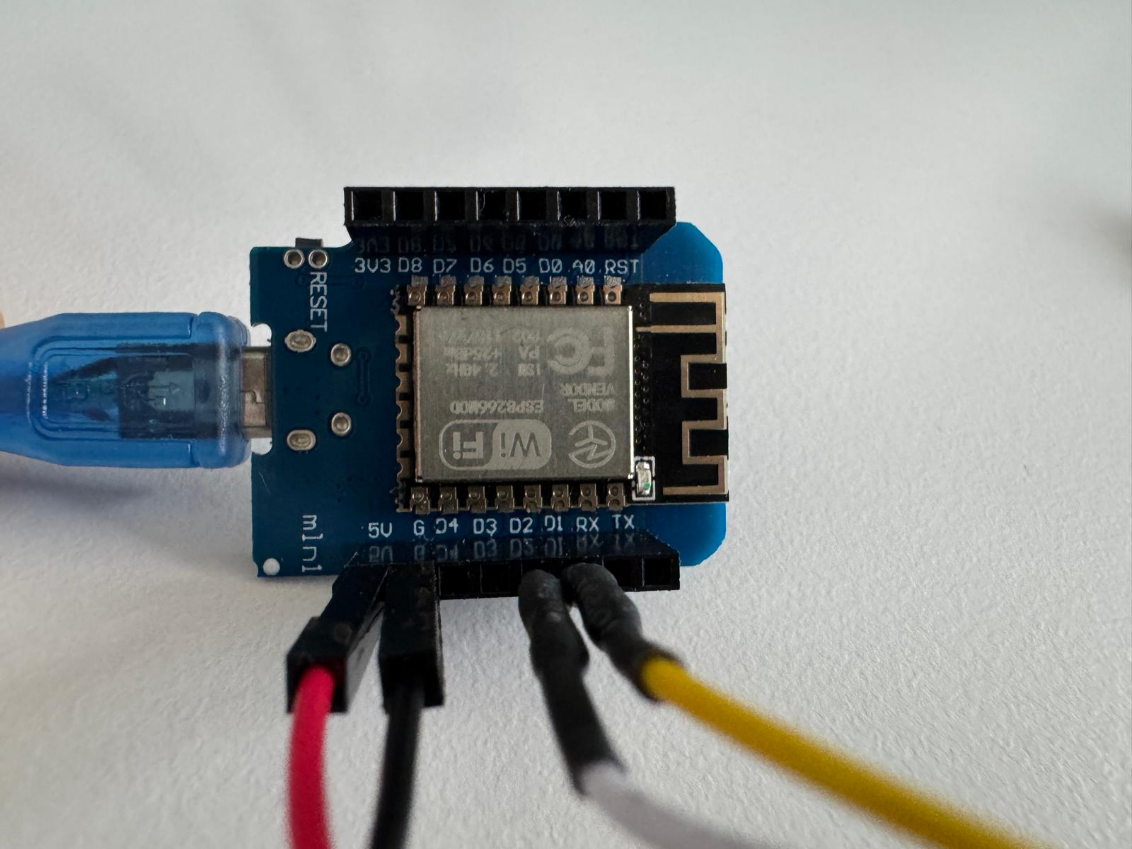
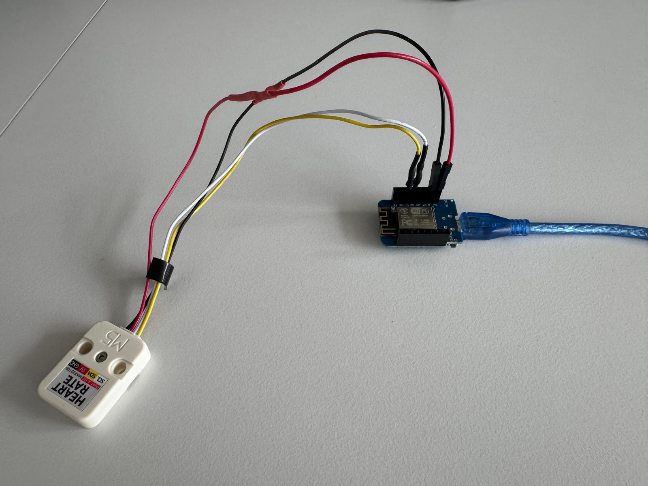
Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

So soll es am Ende aussehen

# Aufsetzen des Geräts

Der erste Schritt des Zusammenbaus bestand darin, den Herzfrequenzsensor mit dem ESP zu verbinden.Der Sensor besitzt vier Kabel: Rot für 5V Stromversorgung, Schwarz für Ground sowie Weiss und Gelb für die Datenübertragung. Die Datenkabel wurden an die Pins D2 und D1 des Boards angeschlossen.Wichtig: Im Code muss man diese Pins entsprechend anpassen – falls man z. B. stattdessen D6 oder D7 verwendet, müssen auch dort die Pin-Nummern geändert werden.



ESP

8266



Sensor

Im nächsten Schritt wurde ein Button-Modul auf einen separaten ESP gesteckt, welcher als mobiler Controller fungiert.Das Modul wird direkt auf die GPIO-Pins des ESPs gesteckt. Dabei ist darauf zu achten, dass es korrekt ausgerichtet ist – die Pin-Beschriftungen wie GND, VCC etc. müssen mit den entsprechenden Pins des ESP übereinstimmen (z. B. GND auf GND).Da das Modul bereits auf ein Shield gelötet ist, ist keine weitere Verkabelung notwendig.



ESP

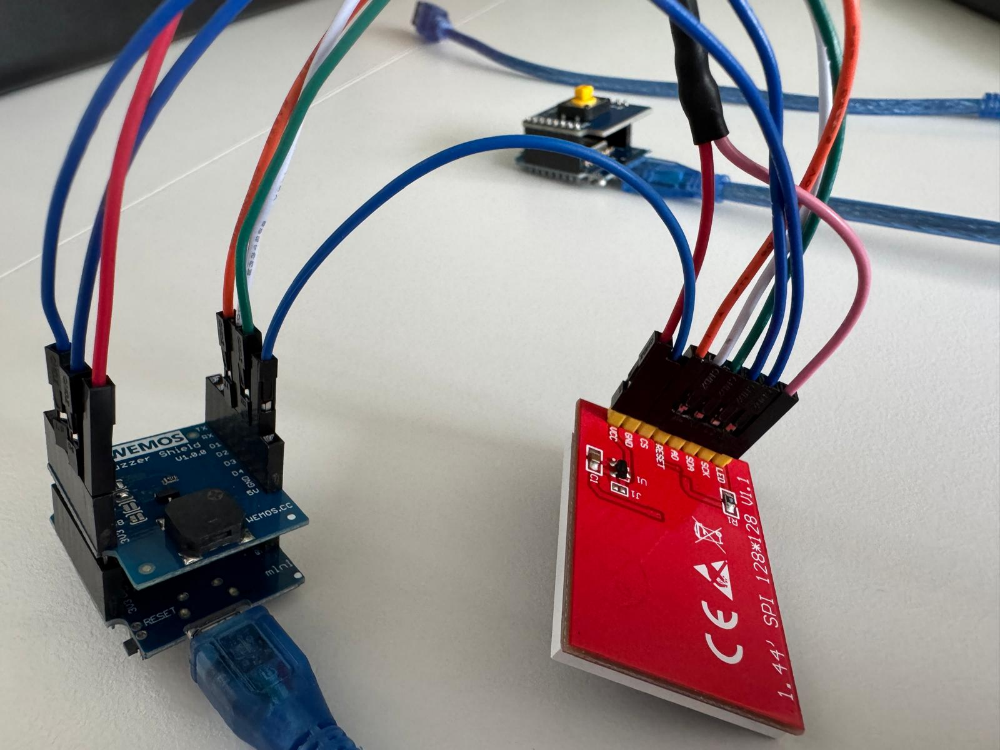
8266

Button-Shield

Als Nächstes stand der anspruchsvollste Schritt an: die Kombination von zwei Modulen – dem Display und dem Buzzer.Damit sich die Pins nicht gegenseitig blockieren, musste Peter das Buzzer-Shield zunächst umlöten. So konnten wir die Pinbelegung so anpassen, dass das Display unabhängig vom Buzzer funktioniert.  
Das Buzzer-Shield selbst liess sich – wie schon beim Button-Shield – einfach auf den ESP stecken.

Beim Display sah das Ganze etwas komplexer aus:  
Hier mussten mehrere Kabel manuell verbunden werden. Zum Glück hatten wir durch die Teams-Unterlagen genügend Material, das die genaue Belegung zeigte.  
Neben **GND** (Ground) und **VCC** (Strom) mussten **6 weitere Pins** korrekt angeschlossen werden:

* **CS**
* **RESET**
* **AD (auch: DC)**
* **SDA (auch: MOSI oder DIN)**
* **LED** (für Hintergrundbeleuchtung)
* **SCK** (Clock)



Buzzer-Shield

Display

ESP

8266

Das ist das Word-Dokument, das uns zur Verfügung stand und genau zeigt, wie welche Kabel eingesteckt werden müssen.

Ein Bild, das Screenshot, Kabel, Text, Verbindungsstück enthält.

KI-generierte Inhalte können fehlerhaft sein.