

HOCHSCHULE LUZERN
TECHNIK UND ARCHITEKTUR

BACHELOR THESIS

Entwicklung einer PCB zur Analyse von Umgebungslärm

Stefano Nicora

7. Juni 2024

Inhaltsverzeichnis

1	Einleitung	3
1.1	Ausgangslage	3
1.2	Ziele	3
2	Mikrofon	4
2.1	Grundlagen	4
2.1.1	MEMS	4
2.1.2	I2S	4
2.1.3	PDM	5
2.1.4	Schalldruckpegel	5
2.1.5	Frequenzbewertung	5
2.2	Komponentenwahl	5
2.2.1	Kriterien	5
2.2.2	Vergleich	6
2.2.3	Fazit	6
3	Mikrocontroller	7
3.1	Grundlagen	7
3.1.1	DMA	7
3.1.2	Bootloader	7
3.1.3	RTC	7
3.1.4	Low Power	8
3.1.5	Peripherie in Hardware oder Software	8
3.2	Komponentenwahl	9
3.2.1	Kriterien	9
3.2.2	Vergleich	10
3.2.3	Fazit	10
4	LED	11
4.1	Grundlagen	11
4.1.1	Wellenlänge	11
4.1.2	Leistungsaufnahme	11
4.1.3	Lichtleistung	11
4.1.4	Vorwiderstand / Strombegrenzung	11
4.2	Komponentenwahl	11
4.2.1	Kriterien	11

4.2.2	Vergleich	11
5	Entwicklung	12
5.1	Konzept	12
5.2	Hardware	14
5.2.1	PCB	14
5.2.2	Kosten	14
5.3	Software	14
5.3.1	Filterdesign	14
5.3.2	PDM	14
5.3.3	I ² C	14
5.3.4	SPI	14
6	Messungen	15
6.1	Leistungsaufnahme	15
6.2	Mikrofon-Kalibrierung	15
6.3	Vergleich	15
7	Fazit und Ausblick	16
8	Anhang	17
8.1	Fremdwörter	17

1 Einleitung

1.1 Ausgangslage

Die Firma hEar hat es sich zum Ziel gesetzt, gegen **TODO**. Dazu wurde in der Masterarbeit von Sophie Mia Willener eine Marktanalyse durchgeführt, sowie ein erster Prototyp gebaut. Dieser Prototyp ist jedoch noch unhandlich und nicht für den Massenmarkt geeignet.

1.2 Ziele

Das Ziel dieser Arbeit ist es, in einem ersten Schritt, auf Basis des vorhandenen Prototypen, ein funktionales, kompaktes und portables Schalldruckpegel-Messgerät zu entwickeln. Dabei sollen folgende Rahmenbedingungen zwingend eingehalten werden:

- Die Laufzeit des Gerätes soll mindestens 12 Stunden betragen.
- Das Gerät wird mit einem Akku betrieben. Dieser wird via eines USB-C-Anschlusses aufgeladen.
- Der Schalldruckpegel wird mit einem MEMS-Mikrofon aufgezeichnet.
- Die Messdaten werden in regelmässigen Abständen auf dem Gerät gespeichert.
- Das Gerät verfügt über eine BLE-Schnittstelle um die Messdaten drahtlos an ein Zielgerät zu übertragen.
- Der aktuelle Schalldruckpegel wird auf der Vorderseite des Gerätes visuell dargestellt.

In einem zweiten Schritt, wird das Gerät kalibriert und dessen Qualität mit auf dem Markt bereits vorhandenen Geräten verglichen.

2 Mikrofon

2.1 Grundlagen

2.1.1 MEMS

2.1.2 I2S

Inter-Integrated Sound (I²S) bezeichnet eine Bus-Schnittstelle, welche von Philips zur Übertragung von digitalen Audiosignalen entwickelt wurde. Ähnlich wie I²C (Inter-Integrated Circuit) wird die Schnittstelle jedoch nur innerhalb des Gerätes verwendet. Dabei werden drei Pins zwischen Sender (hier das Mikrofon) und dem Empfänger (hier der Mikrocontroller) benötigt:

- **SCK** (Serial Clock)
Generiert die Taktrate, welche gleichzeitig die Datenrate der Übertragung definiert. Die Taktrate wird vom Master (hier der Mikrocontroller) vorgegeben.
- **WS** (Word Select)
Gibt vor, welcher Audiokanal (R, L) übertragen werden soll. Dies ermöglicht es, entweder ein Stereo-Signal oder zwei Mono-Signale wie zum Beispiel zwei Mikrofone zu übertragen.
- **SD** (Serial Data) Beinhaltet den eigentlichen Datenstream mit der Datenrate definiert durch **SCK** und der Länge definiert durch **WS**.

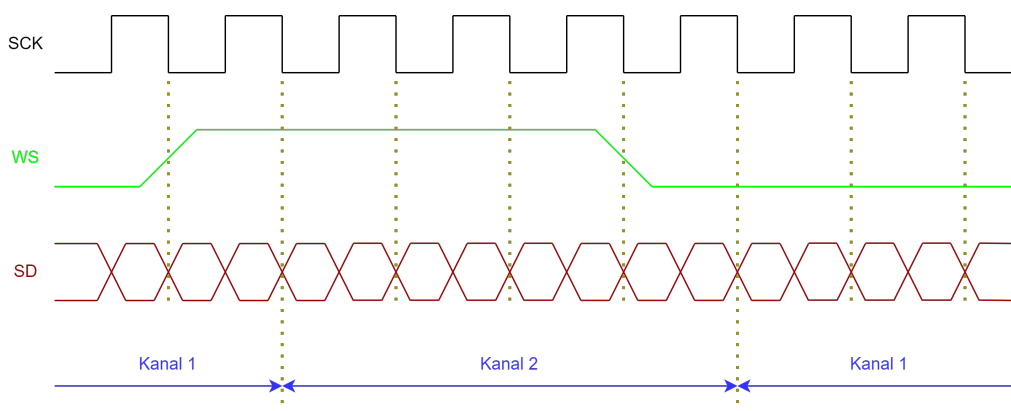


Figure 1: Übersicht I2S-Signale

2.1.3 PDM

Pulse Density Modulation oder **Pulsdichtemodulation** bezeichnet die Darstellung eines analogen Signals in der digitalen Ebene. Dabei entspricht das Verhältnis zwischen der Anzahl der digitalen "1" und digitalen "0" der Amplitude des analogen Signales. Im Gegensatz dazu, wird im weit verbreiteten Pulsweitenmodulationsverfahren, kurz PWM, in jeder Periode das Verhältnis zwischen digitalem "1" und "0" variiert. Dies um zum Beispiel einen FET anzusteuern. **Farben von Bild anpassen**

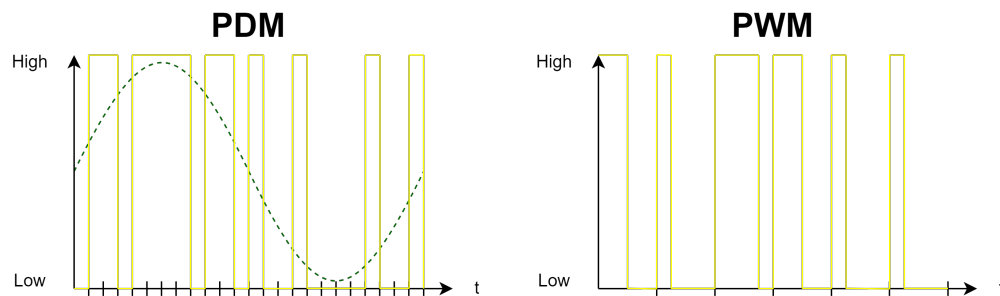


Figure 2: Vergleich PDM und PWM

2.1.4 Schalldruckpegel

2.1.5 Frequenzbewertung

2.2 Komponentenwahl

2.2.1 Kriterien

- Standort der Mikrofonöffnung
- Ausgangssignal
- Maximaler Schalldruckpegel
- MEMS-Technologie
- Frequenzbereich

2.2.2 Vergleich

2.2.3 Fazit

3 Mikrocontroller

3.1 Grundlagen

3.1.1 DMA

Bei **D**irect **M**emory **A**ccess handelt es sich um eine Art Steuerbaustein, welcher unabhängig und parallel zum eigentlichen Rechenkern arbeitet. Dieser hat, wie der Rechenkern selbst, Zugriff auf den gesamten Datenbus. Dieser ermöglicht es, effizient Daten von Peripherien wie der UART, I²S, SPI, ect. in den RAM und umgekehrt zu laden. Dadurch wird der Rechenkern entlastet und kann für wichtigere Aufgaben eingesetzt oder gar in einen Energiesparmodus versetzt werden. Dabei gibt es zwei Möglichkeiten mit dem DMA-Baustein zu interagieren:

- Fire-and-forget
- **TODO**

3.1.2 Bootloader

Unter einem Bootloader versteht man Code, welcher grundsätzlich nach dem erstmaligen flashen, persistent im System bleibt. Er dient als Ausgangspunkt für die eigentliche Software und ermöglicht es beispielsweise auch, während der Laufzeit des Gerätes, Softwareupdates herunter zu laden und diese anschliessend einzuspielen. Insbesondere wenn Funkprotokolle wie Bluetooth oder WLAN eingesetzt werden, ist ein Bootloader oftmals unerlässlich. Der Nachteil ist jedoch, dass dieser den ohnehin knappen Speicherplatz weiter reduziert.

3.1.3 RTC

Unter einer **R**eal **T**ime **C**lock versteht man ein Stück Hardware (intern oder extern), welche die Zeit ab Inbetriebnahme misst. Übliche RTCs haben, aufgrund ihrer Bauweise eine Genauigkeit von 100ppm. Da dies einer Zeitverschiebung von 4min und 19s @32kHz entspricht, enthalten die meisten RTCs Kompensationsmechanismen, um diese Zeit auf unter 100ms zu bringen. [1]

3.1.4 Low Power

Mit Low Power wird, insbesondere bei Embedded-Systemen, die reduzierte Leistungsaufnahme beschrieben. Oftmals verfügen die Systeme über mehrere Modi mit jeweils unterschiedlichen Eigenschaften. Das Ziel ist es, mit jedem tieferen Modus etwas mehr Energie zu sparen und somit die Akkulaufzeit zu verlängern. Wie in Abbildung 3 ersichtlich, sinkt zwar die Leistungsaufnahme mit jeder Stufe, jedoch werden dabei gewisse Peripherien abgeschaltet und die allgemeine Startzeit des Systems wird erhöht.

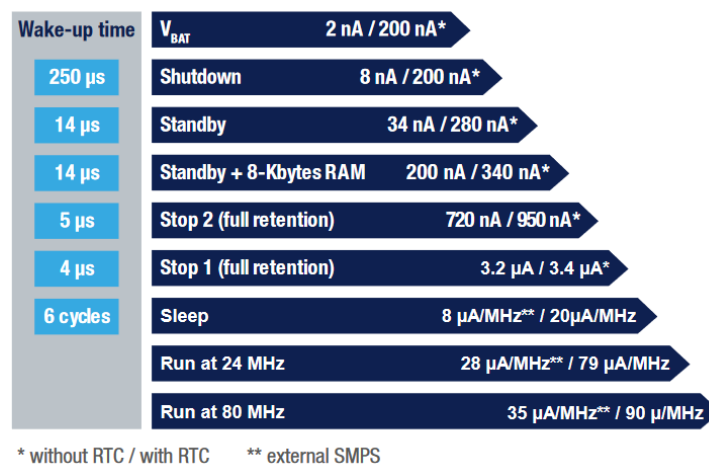


Figure 3: Vergleich Low-Power-Modi STM | Quelle: [2]

3.1.5 Peripherie in Hardware oder Software

Bei allen gängigen Kommunikationsprotokollen ist die Frage, wie diese implementiert werden sollen. Dabei stehen folgende Möglichkeiten zur Auswahl:

- in **Hardware**
 - + Die benötigte Infrastruktur (Register, CRC, etc.) sind bereits physikalisch implementiert. Die Kommunikation kann dementsprechend vom Rechenkern abgekoppelt durchgeführt werden und ermöglicht eine gewisse Parallelität von mehreren Prozessen mit wenig Rechenaufwand. Insbesondere in Verbindung mit DMA, kann so viel Rechenzeit eingespart werden.
 - Bei den meisten Mikrocontrollern ist die spezifische Hardware an

einzelne Pins gebunden. Dies wirkt sich auf die Flexibilität des PCB-Layouts oder gar des Gesamtsystems (zu wenig Anschlüsse) aus.

- in **Software**
 - + Die meisten Pins können flexibel für jedes Protokoll verwendet werden. Dadurch vereinfacht sich auch das PCB-Layouten.
 - Bei Verwendung von Ein-Kern-Systemen, welche im Embedded-Bereich mehrheitlich anzutreffen sind, kann keine echte Parallelität von Prozessen erfolgen. Dies verringert die Ausführungsgeschwindigkeit.

Grundsätzlich ist, nach Möglichkeit und Verfügbarkeit, eine Hardware-Peripherie zu bevorzugen. Insbesondere bei batteriebetriebenen Systemen mit langer Laufzeit, ist diese ausschlaggebend.

3.2 Komponentenwahl

Es existieren eine Vielzahl von Mikrocontroller-Herstellern. Viele dieser verfügen über eine breite Palette an BLE-tauglichen Chips. Um den geeignetsten darunter zu finden, werden nachfolgend die benötigten Schnittstellen definiert und mehrere, vorselektionierte Mikrocontroller, miteinander verglichen.

3.2.1 Kriterien

- **I²S + PDM** zur Ansteuerung des Mikrofons.
- **SPI** zur Ansteuerung von möglichen Peripherien wie externe Speicherbausteine oder Displays.
- **I²C** zur Ansteuerung von Laderegler und LED-Treiber-IC.
- **DMA** zur energieeffizienten Verarbeitung der Daten.
- **RTC** um die gemessenen Werten mit einem Zeitstempel verknüpfen zu können.
- **BLE-Fähiger** Mikrocontroller mit vorgefertigter Antenne um die Integration zu vereinfachen.

3.2.2 Vergleich

Hersteller	Typ	I2S	I2C	SPI	DMA	BLE	PDM	RTC	ANT	Flash (kB)	uA/MHz	Zusatz
Infineon	CY8C63x6	x	x	x	x	x	x	x	-	<=512	40	Nicht via normale Distributoren
Infineon	CY8C63x7	x	x	x	x	x	x	x	-	<=512	40	Nicht via normale Distributoren
Microchip	BM83	x	x	-	-	x	x	-	-	-	-	Charge-controller für Akku Not recommended for new Designs
Nordic Semiconductor	nRF52832	x	x	x	x	x	x	x	x	<=512	58	ANT: BMD-350-A-R (u-blox)
Nordic Semiconductor	nRF52833	x	x	x	x	x	x	x	x	512	52	ANT: NINA-B406-00B (u-blox) BL653 (Lairdconnect)
Nordic Semiconductor	nRF52840	x	x	x	x	x	x	x	x	1024	52	ANT: NINA-B302-00B (u-blox) BMD-340-A-R (u-blox) BL654PA (Lairdconnect)
Nordic Semiconductor	nRF5340	x	x	x	x	x	x	x	x	1024	48	ANT: NORA-B106-00B (u-blox) BL5340 (Lairdconnect)
Silicon Labs	EFR32BG22	x	x	x	x	x	x	x	x	<=512	27	ANT: BGM220P Lyra P (Lairdconnect)
Silicon Labs	EFR32BG27	x	x	x	x	x	x	x	-	768	29	-
STmicro	STM32WB	x	x	x	x	x	x	x	x	1024	53	ANT: STM32WB5MMG
STmicro	BlueNRG-LP	x	x	x	x	x	x	x	-	256	18	-

Figure 4: Vergleich von möglichen Mikrocontrollern

3.2.3 Fazit

Wie dem Vergleich zu entnehmen ist, unterscheiden sich die Mikrocontroller meist nur in der integrierten Flash-Grösse sowie der Leistungsaufnahme bei einer gewissen Taktrate. Dadurch wird die Wahl nicht eingeschränkt, was eine geeignete Evaluation nicht vereinfacht. Ausschlaggebend ist jedoch der Formfaktor. Bis auf Silicon Labs und STmicro muss für einen Mikrocontroller mit vorgefertigter Antenne auf Drittanbieter ausgewichen werden. In Zeiten von Katastrophen wie COVID oder Kriegen inmitten von Handelsrouten ist die Produktverfügbarkeit einer der zentraler Aspekte. Aus diesem Grund wird nachfolgend der EFR32BG22 von Silicon Labs bzw. dessen Modul-Variante, der **BGM220P**, eingesetzt.

4 LED

4.1 Grundlagen

4.1.1 Wellenlänge

4.1.2 Leistungsaufnahme

4.1.3 Lichtleistung

4.1.4 Vorwiderstand / Strombegrenzung

4.2 Komponentenwahl

4.2.1 Kriterien

- Lichtleistung
- Wellenlänge

4.2.2 Vergleich

5 Entwicklung

5.1 Konzept

Die im Unterkapitel 1.2 definierten Ziele sollen nun umgesetzt werden. Dazu soll die von aussen sichtbare Vorderseite möglichst wenig Bauteile aufweisen, um das visuelle Design nicht zu beeinträchtigen. Mit einer angestrebten Gesamtgrösse von 60mm Durchmesser, können die gewählten Bauteile geradenoch auf dem PCB in einer symmetrischen Art platziert werden.

Front Die Vorderseite weist folgende Bauteile auf:

- 8 LEDs
- Mikrofon
- LED-Treiber-IC

Zudem sind die LEDs sind ringförmig angeordnet, um der Anforderung eines visuellen Ladebalkens entsprechen zu können.

Back Die Rückseite hingegen, beinhaltet folgende Bauteile:

- USB-C-Ladeanschluss
- Polyfuse (Automatisch rückstellbarer Überstromschutz)
- Switch um generell das Gerät ein- und auszuschalten
- BMS, um den Akku sicher auf- und entladen zu können
- Mikrocontroller, welcher sich aufgrund seiner Antenne am Rand des PCBs befinden muss
- Akkuhalter
- Optionaler Speicherbaustein

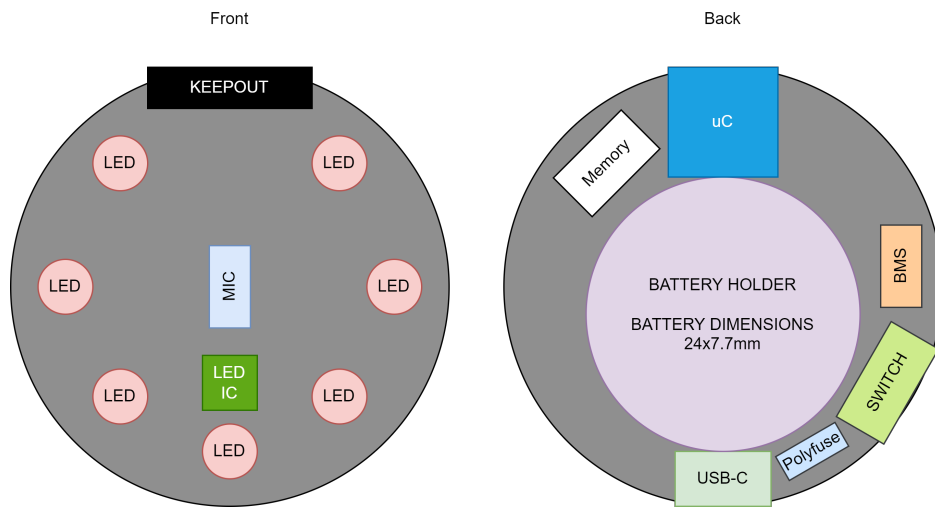


Figure 5: Konzept PCB-Layout

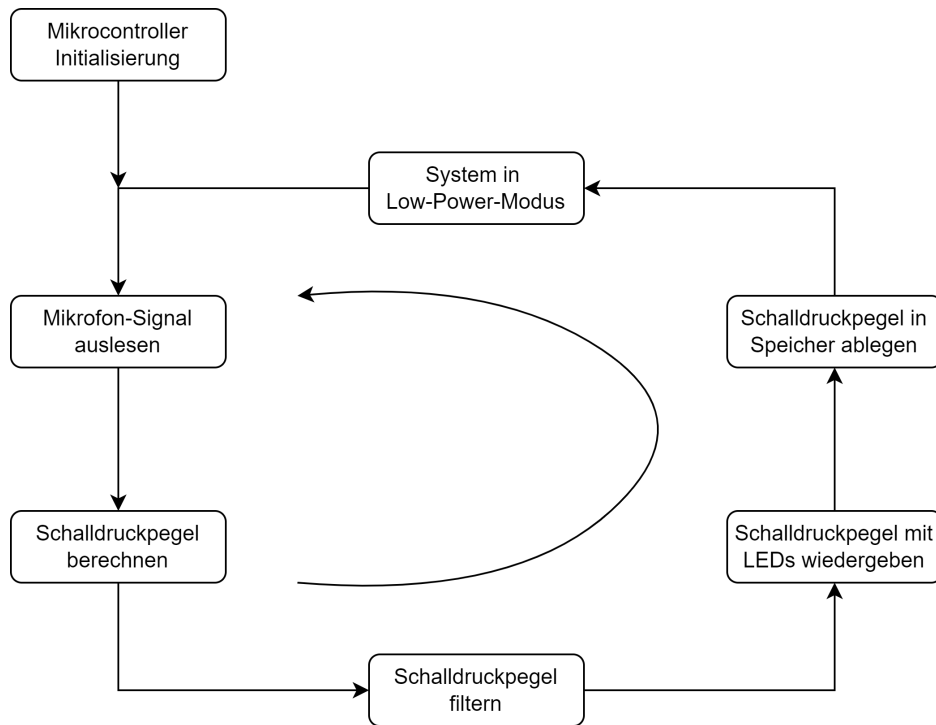


Figure 6: Konzept Softwareablauf

5.2 Hardware

5.2.1 PCB

Laderegler

5.2.2 Kosten

5.3 Software

5.3.1 Filterdesign

Wie im Unterkapitel 2.1.5 erläutert, reagiert unser Gehör nicht auf alle Frequenzen gleich. Um diese Frequenzbewertung in die Schalldruckpegel-Berechnung einfließen zu lassen, wird ein digitales, diskretes Filter benötigt. Die benötigten Frequenzbewertungsfilter werden jedoch fast ausschliesslich in der S-Ebene (Laplace-Bereich, zeitkontinuierlich) veröffentlicht. Dadurch müssen diese mit der bilinen Transformation [3] vom Frequenz- in den Zeitbereich transformiert werden.

$$G_A(s) = \frac{7.39705 \cdot 10^9 \cdot s^4}{(s + 129.4)^2 \cdot (s + 676.7) \cdot (s + 4636) \cdot (s + 76617)^2} \quad (1)$$

$$G_C(s) = \frac{5,91797 \cdot 10^9 \cdot s^2}{(s + 129,4)^2 \cdot (s + 76617)^2} \quad (2)$$

5.3.2 PDM

5.3.3 I²C

5.3.4 SPI

6 Messungen

6.1 Leistungsaufnahme

6.2 Mikrofon-Kalibrierung

6.3 Vergleich

7 Fazit und Ausblick

8 Anhang

References

- [1] G. Dighe, “TPS65950 Real-Time Clock Timing Compensation,” Oct. 2008.
- [2] “STM32L4 Series - PDF Documentation,” Mar. 2024.
- [3] A. V. Oppenheimer, *Zeitdiskrete Signalverarbeitung*. R. Oldenbourg.

Abbildungsverzeichnis

2	Vergleich PDM und PWM	5
3	Vergleich Low-Power-Modi STM Quelle: [2]	8
4	Vergleich von möglichen Mikrocontrollern	10
5	Konzept PCB-Layout	13
6	Konzept Softwareablauf	13

8.1 Fremdwörter

CRC
Register
Pins
Embedded
ppm
flashen