

Audio-Beamformer

Bachelor's Thesis

Florian Baumgartner

florian.baumgartner@ost.ch

Thierry Schwaller

thierry.schwaller@ost.ch

Advisor

Hannes Badertscher

Examiner

Gabriel Sidler

Interdisciplinary Center for Artificial Intelligence
Eastern Switzerland University of Applied Sciences

June 2022

Abstract

Naturally, sound waves propagate in an omnidirectional pattern. However, in some cases, a directional pattern would be preferred. In comparison to light or electromagnetic waves, it turns out to be a very difficult task to focus audio waves in a specific direction. The underlying reason for this behavior comes from the physical long wavelength of the audible spectrum.

This bachelor's thesis concentrates on how to overcome this effect by using higher frequencies in the ultrasonic spectrum and therefore be able to create a highly directional audio beam. In addition, beam-steering methods are applied to change the direction and focus point by software.

In order to achieve a directional sound beam, a linear phased array has been developed, consisting of 19 rows of 8 ultrasonic transducers each. For this, a single PCB, consisting of over 1300 components was specially designed. Two FPGAs modulate the baseband audio signal onto a 40 kHz ultrasonic carrier. First-order Sigma-Delta-Modulators perform the analog conversation in combination with a Class-D amplifier output stage. In addition, each channel can be delayed and attenuated individually. A Raspberry Pi Compute Module 4 is used to apply real-time digital signal processing techniques to further improve the audio quality. Advanced face-detection algorithms are used to locate a target and therefore be able to direct the sound in its direction. As input sources, Bluetooth® and AirPlay® streaming are supported, as well as other input devices, such as USB-Microphones.

The directivity, beam-steering capability, and overall audio quality have been determined in a comprehensive human expertise test. The Audio-Beamformer performs well in all categories and satisfies the project's goals. Especially the range of up to 50 meters is awe-inspiring.

The Audio-Beamformer could lead to a real alternative to conventional loudspeakers with some further improvements.

Acknowledgement

We take this opportunity to express gratitude to all of the people that supported us in this bachelor's thesis.

Especially to Hannes Badertscher for the supervision, Dr. Guido Schuster for his theoretical explanations in the field of digital signal processing, Dr. Hans-Dieter Lang for his comprehensive support in designing high-speed electronics, Dorian Amiet for answering questions about the FPGA implementation, Luca Jost for developing the face-tracking algorithm, Caspar Naef for his endurance, Anne Marie O'Neill for the proofreading and all participants of the human expertise test.

We also thank Elvis for the unceasing encouragement and support throughout the entire project.

Contents

1	Introduction	3
1.1	Background	3
1.2	Scope	3
1.3	Approach	4
1.4	Open Source	4
2	Preliminaries	5
2.1	Class-D Amplifier	5
2.2	Quadrature Amplitude Modulation	6
2.3	Piezoelectric Ultrasonic Transducer	6
2.4	Diffraction from Slits	7
2.4.1	Single-Slit Diffraction	7
2.4.2	Diffraction on Slits & Fourier Transform	8
2.4.3	Multiple Slits Diffraction	9
2.5	Acoustics	10
2.5.1	Basics	10
2.5.2	Sound Power	10
2.5.3	Directivity	10
2.5.4	Sound Power Level vs. Sound Pressure Level	11
3	Parametric Loudspeaker Array	13
3.1	Introduction	13
3.2	Directivity of Ultrasonic Transducers	13
3.3	Demodulation Process	14
3.4	Modulation Process	15
3.4.1	Amplitude Modulation	15
3.4.2	Modified Amplitude Modulation	15
3.5	Array Signal Processing	17
3.5.1	Phased Array Beam Model	17
3.5.2	Beam Steering	19
3.5.3	Beam Focusing	20
3.5.4	Array Amplitude Weighting	20
4	Design	23
4.1	Overview	23
4.1.1	Key Requirements	23
4.1.2	Key Decisions	23
4.2	Hardware Design	24

4.2.1	Block Diagram	25
4.2.2	Power Management	26
4.2.3	Raspberry Pi Compute Module 4	27
4.2.4	FPGA	27
4.2.5	Sensors and HMI	27
4.2.6	Output Stage	30
4.2.7	Ultrasonic Transducer Array	31
4.3	FPGA Design	34
4.3.1	Integer Arithmetic	34
4.3.2	Signal Flow Diagram	35
4.3.3	Clock & Synchronization	36
4.3.4	I ² S Interface	36
4.3.5	SPI Interface	37
4.3.6	Interpolation	38
4.3.7	Modulation	38
4.3.8	Sigma-Delta-Modulator	38
4.3.9	Channel Delay & Gain	39
4.3.10	Dead Time Generator	40
4.4	Software Design	41
4.4.1	Structure	41
4.4.2	GUI	42
4.4.3	Audio Processing	44
4.4.4	Beam Steering	44
4.4.5	Sensors	45
4.4.6	Face Tracking	45
4.4.7	Operating System	46
4.5	Mechanical Design	47
4.5.1	Concept	47
4.5.2	Enclosure	48
5	Risks & Safety	49
5.1	Risks	49
5.2	Safety	51
6	Measurements	53
6.1	Human Expertise Test	53
6.1.1	Test Setup	53
6.1.2	Audio Quality	54
6.1.3	Audio Volume	56
6.2	Ultrasound Measurements	58
6.2.1	Directivity	58
6.2.2	Beam Steering	58
6.2.3	Beam Focusing	59
6.2.4	Evaluation	59
6.3	Transducer Measurements	60
6.3.1	Frequency Response	60
6.3.2	Power Consumption	60
6.3.3	Impedance Measurements	60

7 Conclusion	61
7.1 Continuing Work	61
7.2 Personal Reflections	62
A Appendix	63
A.1 Declaration of Authorship	64
A.2 Data Archive	65
A.3 Definition of Task	66
A.4 Project Schedule	68
A.5 Datasheet Ultrasonic Transducer	69
A.6 Audio-Beamformer Schematics	70
A.7 PCB Top-Layer	80
A.8 PCB Bottom-Layer	81
A.9 PCB Top-Overlay	82
A.10 PCB Bottom-Overlay	83
A.11 PCB Outline	84
A.12 Bill of Materials (BOM)	85
A.13 Mechanical Drawing Top-Plate	86
A.14 Mechanical Drawing Base-Plate	87
A.15 Mechanical Drawing Connecting-Bar	88
Bibliography	89

Acronyms

- AI** Artificial Intelligence. 46
- AM** Amplitude Modulation. 15, 44, 54, 55
- CAD** Computer Aided Design. 24, 47
- CPU** Central Processing Unit. 43
- DC** Direct Current. 15, 26, 28, 29
- DIN** Deutsches Institut für Normung. 48
- eMMC** embedded Multimedia Card. 27, 30
- FIR** Finite Impulse Response. 20, 44
- FPGA** Field Programmable Gate Array. 23, 27, 34, 36, 37, 40, 44
- FPS** Frames per Second. 27
- GPU** Graphical Processing Unit. 45
- GUI** Graphical User Interface. 29
- HDMI** High-Definition Multimedia Interface. 29, 30
- HMI** Human Machine Interface. 27
- HV** High-Voltage. 26
- I²C** Inter-Integrated Circuit. 28, 29
- I²S** Inter-IC Sound. 27, 36, 44
- IC** Integrated Circuit. 29, 30
- IDFT** Inverse Discrete Fourier Transform. 20
- IIR** Infinite Impulse Response. 38
- IO** Input-Output. 27, 34
- IP** Intellectual Property. 34
- ISO** International Organization for Standardization. 48
- KZK** Khokhlov, Zabalotskaya and Kuznetsov. 14
- LC** Inductive-Capacitive. 5
- LCD** Liquid Crystal Display. 29
- LED** Light-Emitting Diode. 29, 30, 43, 45
- LUT** Lookup-Table. 38
- MAM** Modified Amplitude Modulation. 15, 42, 44, 54, 55
- MNN** Mobile Neural Network. 45, 46
- MOSFET** Metal Oxide Semiconductor Field-Effect Transistor. 5, 30, 31, 40
- MSB** Most Significant Bit. 38

- OS** Operating System. 46
- PCB** Printed Circuit Board. 4, 24, 27, 28, 32, 48
- PLL** Phase Locked Loop. 34, 36
- PUT** Piezoelectric Ultrasonic Transducer. 6, 15
- PWM** Pulse-Width Modulation. 5, 29
- QAM** Quadrature Amplitude Modulation. 6, 15
- RAM** Random Access Memory. 27, 34, 38–40
- RC** Resistive-Capacitive. 29
- RGB** Red Green Blue. 29, 30
- RMS** Root Mean Square. 60
- SD** Secure Digital Memory Card. 27
- SNR** Signal-to-Noise Ratio. 38–40
- SPI** Serial Peripheral Interface. 26, 27, 30, 37, 39, 40, 44
- SPL** Sound Pressure Level. 10
- SUVA** Schweizerische Unfallversicherungsanstalt. 49, 50
- ToF** Time-of-Flight. 28, 43, 45, 51
- UI** User Interface. 29, 42
- USB** Universal Serial Bus. 27, 29, 30

1

Introduction

1.1 Background

The omnidirectionality of loudspeakers is not always something to aspire to but rather something that should be suppressed. This can be done by using nonlinear characteristics of air in a phenomenon known as *Sound from Ultrasound*. With this principle, a highly directional steerable audio beam can be created.

Until now, many studies have been conducted in the field of *Sound from Ultrasound* and the theory of its inner workings has been mostly understood. Nevertheless, no commercially available product has yet been developed. Due to this, we decided to apply this theory gained from years of research and combine it with our ideas to create a fully functional device, the Audio-Beamformer.

There are many potential use cases for such loudspeakers. Be it as a possibility to direct the sound of a phone call to only the person sitting right in front of a computer and mitigating the need of wearing headphones or the possibility to sit with foreign friends on the same couch and watch a movie in different languages.

1.2 Scope

Around the task given, which can be seen in Appendix A.3, the goals of this thesis have been defined by our own. Our first goal is to dive deep into the theory of how to generate a highly directional and steerable audio beam and get a good overview of the different technologies and ideas implemented. The second goal is to design a fully functioning product. For this, we set ourselves several requirements. Developing not only a working *Sound from Ultrasound* loudspeaker, but also making it highly professional and easy to use. A device which is ideal for demonstration purposes.

1.3 Approach

To achieve our goals, we started by researching different theories on creating directional sound. We decided right from the start that the *Sound from Ultrasound* principle is the way to go, mainly because of the higher directivity. To get familiar with this kind of loudspeaker, we've searched for hardware that was ready to use. However, as there exists no commercially available product, we had to develop the hardware ourselves. For this, we made multiple iterations of prototypes, which can be seen in Figure 1.1, starting with a basic buildup on a breadboard, then soldering everything to a perfboard. After that, we designed and ordered a prototype PCB and, in the end, redesigned the prototype once again and ordered the final PCB. To test everything and quantify the obtained results, we measured everything extensively with a dedicated ultrasound microphone and conducted human expertise tests.

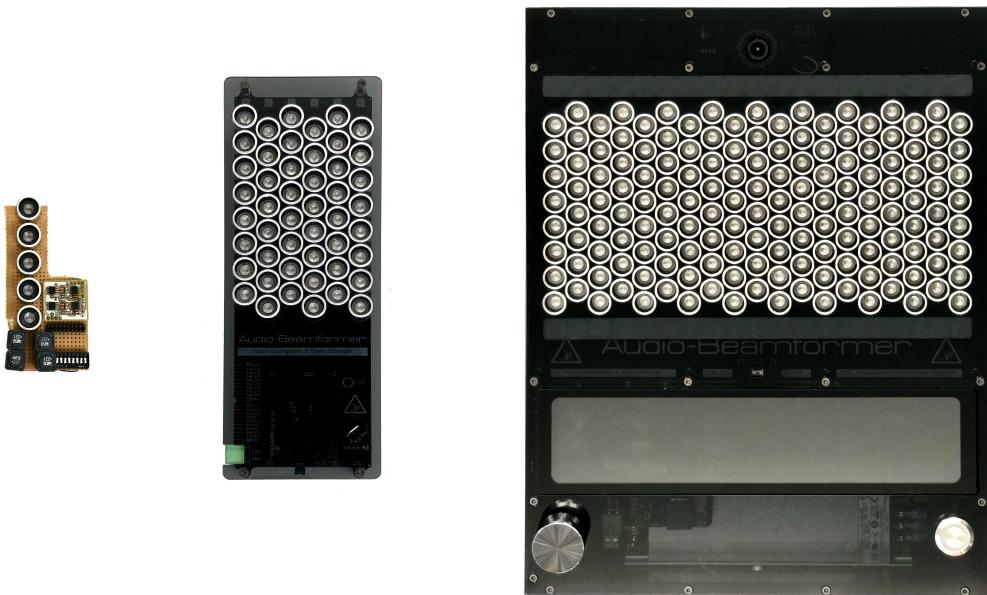


Figure 1.1: Hardware Prototype Revisions

1.4 Open Source

Right from the start it was decided that everything about the project would be released under an open source license. Both of us are huge supporters of open source and believe it will be the future of engineering. Building upon existing libraries and code under open source licenses, allowed us to accelerate the design process. Sometimes open source is considered an act of charity, but in our case, the benefits of using it outweigh any closed source processes. All documents and files for this project can be found on our GitHub page. A short description of all the repositories can be found in the Appendix A.2.

2

Preliminaries

2.1 Class-D Amplifier

A Class-D Amplifier, also known as *digital-amplifier* is a very efficient design of a power amplifier most commonly used in audio applications. The main principle is based on a switching output stage (often constructed with MOSFETs) that is driven by a Pulse-Width Modulation (PWM) signal. The switching frequency is much higher than the bandwidth of the amplified signal (typically one order of magnitude higher). The most common modulation scheme uses a saw-tooth shaped signal and a comparator to generate the PWM signal.

The output signal then has to be low-pass filtered, in order to suppress the high switching noise. This is usually done by using a second order LC-Filter at the output. Class-D amplifiers benefit from a low component count and the very high efficiency of up to 90% and above.

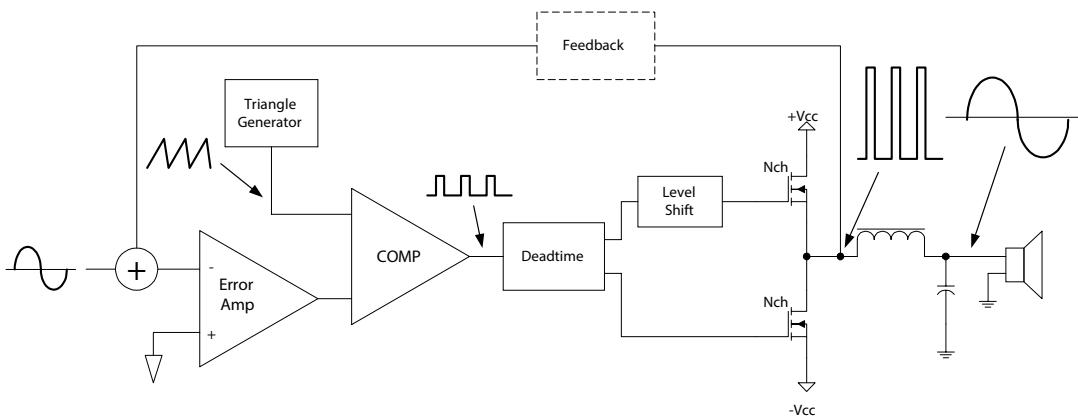


Figure 2.1: Typical Class-D Amplifier [2]

2.2 Quadrature Amplitude Modulation

Quadrature Amplitude Modulation (QAM) [7] is a modulation scheme where an in-phase component $I(t)$ and a quadrature component $Q(t)$ are mixed with orthogonal carriers and then added together, as shown in Figure 2.2.

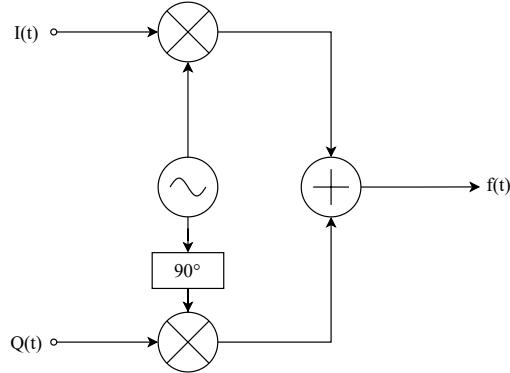


Figure 2.2: Block diagram QAM

The output signal $f(t)$ can be calculated as

$$f(t) = I(t) \sin(\omega_0 t) + Q(t) \underbrace{\sin\left(\omega_0 t + \frac{\pi}{2}\right)}_{\cos(\omega_0 t)}. \quad (2.1)$$

Through the use of some trigonometric identities this can be simplified to

$$f(t) = \sqrt{I^2(t) + Q^2(t)} \sin\left(\omega_0 t + \arctan\left(\frac{Q(t)}{I(t)}\right)\right). \quad (2.2)$$

2.3 Piezoelectric Ultrasonic Transducer

Piezoelectric Ultrasonic Transducer (PUT) emit sound by using the reciprocal piezoelectric effect [15]. By applying an electric voltage to piezoelectric material it is deformed and therefore produces ultrasound. Electrically a PUT is best described by using the Butterworth van Dyke model, which is shown in Figure 2.3. The impedance response of this model is shown in Figure 2.4.

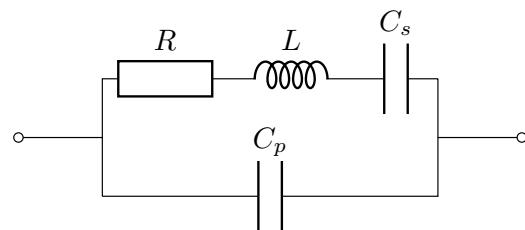


Figure 2.3: Butterworth van Dyke Model

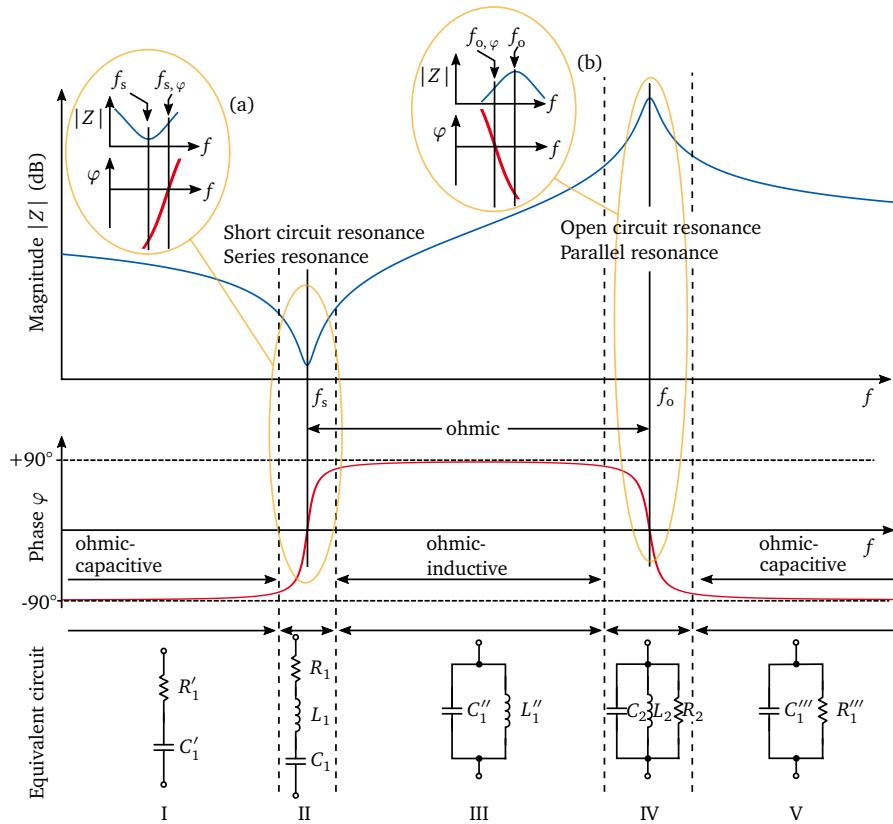


Figure 2.4: Magnitude of the Impedance Response [15]

2.4 Diffraction from Slits

The principle of Huygens-Fresnel [11] states that every element on a wave surface can be viewed as a center of a spherical wave. Even if this is physically viewed not entirely correct, this provides a good model through which the wave changes in time can be calculated. This principle will be used to calculate the diffraction of waves on slits.

The calculations for the diffraction are only valid in the far-field.

2.4.1 Single-Slit Diffraction

All the points inside of the slits are viewed, accordingly to the Huygens-Fresnel principle, as centers of spherical waves. If we now want to know the pressure of a wave of frequency ω at a certain point in space, with distance r and angle φ to the slit, one can superposition all the points inside of the slit [11]

$$p(r, \varphi, \omega) = \frac{A}{rs} \int_0^a \cos(\omega t - kr + kx \sin(\varphi)) dx. \quad (2.3)$$

Where A is the amplitude of the wave at the slit, a is the size of the slit and k is the wave number given as

$$k = \frac{\omega}{c}. \quad (2.4)$$

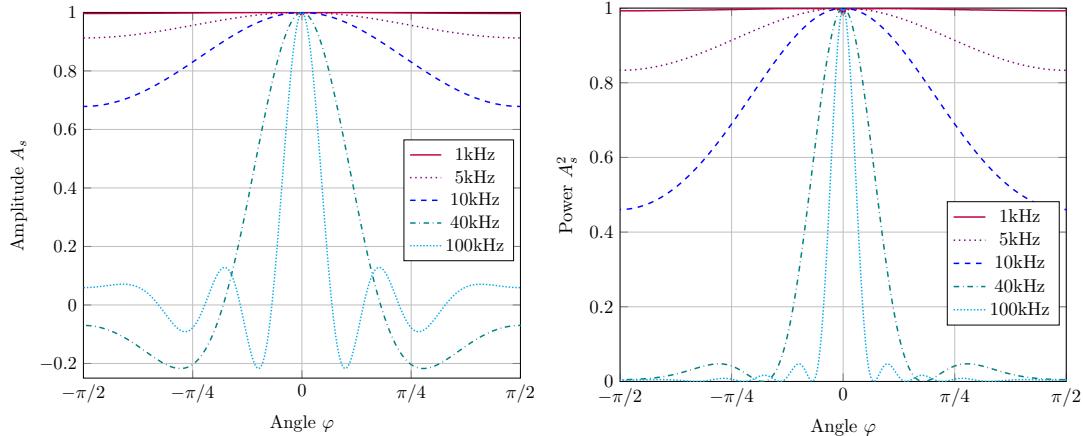


Figure 2.5: Amplitude of Waves with different Frequencies **Figure 2.6:** Power of Waves with different Frequencies

This can be simplified to

$$p(r, \varphi, \omega) = \frac{A}{r} \underbrace{\frac{\sin\left(\frac{ka \sin \varphi}{2}\right)}{\frac{ka \sin \varphi}{2}}}_{A_s(\varphi, k, a)} \cos(\omega t - kr). \quad (2.5)$$

The function $A_s(\varphi, \omega, a)$ shows how the amplitude of the wave varies according to the angle, the frequency and the size of the slit.

In Figure 2.5 it is shown how the amplitude over the angle changes with different frequency while the slit size is hold constant at $a = 0.016$ m for waves with the speed 343 m/s (speed of sound at 22 degrees Celsius). Additionally, Figure 2.6 shows the power.

2.4.2 Diffraction on Slits & Fourier Transform

The diffraction pattern of a sound wave, in the x-y plane, can be calculated using the Fourier transform, because of the similarities in the equations. If the frequency is assumed to be $f = \frac{kx}{2\pi y}$ they are exactly the same. [14]

For example in the case with one slit, which has a size of s and its middle is at the point $(0, 0)$ and edges at $(0, \pm a/2)$ the *Slit function* $s(x)$ would look like

$$s(x) = \begin{cases} 1 & |x| < \frac{a}{2} \\ 0 & \text{else} \end{cases}. \quad (2.6)$$

The Fourier transform of this would be

$$\mathcal{F}\{s(x)\} = A_s \left(2\pi \frac{fx}{cy} \right) = a \frac{\sin\left(\frac{\pi fxa}{cy}\right)}{\frac{\pi fxa}{cy}}. \quad (2.7)$$

Near the y axis this can be simplified to [12]

$$\frac{x}{y} = \sin(\varphi). \quad (2.8)$$

This leads to the same formula for the amplitude as seen before in 2.5

$$A_s(\varphi, k, a) = \frac{\sin\left(\frac{ka \sin \varphi}{2}\right)}{\frac{ka \sin \varphi}{2}}. \quad (2.9)$$

2.4.3 Multiple Slits Diffraction

With the Fourier transform idea, explained in Section 2.4.2, the amplitude and power of any multi-slit layout can now easily be calculated. For M transducers, with the size a and a spacing d the pattern can be easily calculated to be

$$A_s(\varphi, k, a, d, M) = \frac{\sin\left(\frac{ka \sin \varphi}{2}\right)}{\frac{ka \sin \varphi}{2}} \frac{\sin^2\left(\frac{kd \sin \varphi}{2} M\right)}{\sin^2\left(\frac{kd \sin \varphi}{2}\right)}. \quad (2.10)$$

In Figure 2.7 examples for different transducer widths is shown.

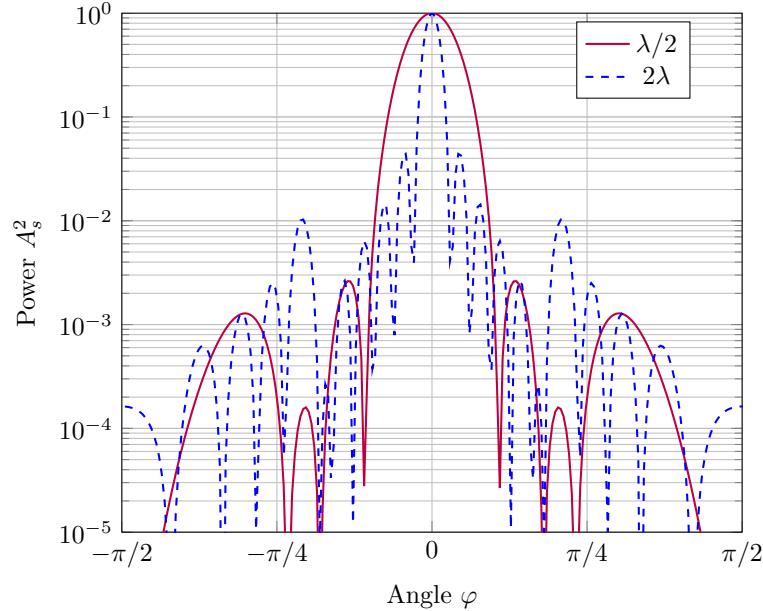


Figure 2.7: Diffraction on Multiple Slits

2.5 Acoustics

2.5.1 Basics

A sound wave is a disturbance propagated through a material (mostly air in acoustics) which causes a variation in pressure. This disturbance in the ambient pressure is the sound pressure and is proportional to the inverse of the squared distance to the source [1]

$$p(r) \propto \frac{1}{r^2}. \quad (2.11)$$

To get a better feeling for sound pressure, in table 2.1 are some examples listed of different sound sources, distances and their approximate sound pressure. [10]

Sound source	Distance	Sound Pressure [Pa]	SPL [dB ref. 20 μPa]
Jet takeoff	60 m	20	120
Loud shouting	1.5 m	2	100
Busy street	-	0.2	80
Normal conversation	1 m	0.02	60
Hearing threshold	-	0.00002	0

Table 2.1: Sound Sources and their respective Sound Pressure (Level)

Additionally, the Sound Pressure Level (SPL) L_p is displayed. It is the logarithmic measure of sound pressure p relative to a reference value p_0 .

$$L_p = 20 \log_{10} \left(\frac{p}{p_0} \right) [\text{dB}]. \quad (2.12)$$

Mostly this reference value is picked to be $20 \mu\text{Pa}$ which is the hearing threshold of humans. [10]

2.5.2 Sound Power

The sound power is the total sound energy emitted by a loudspeaker. Often instead of the sound power the sound power level is used. This sound power level L_W is the logarithmic measure of sound power P relative to a reference power P_0

$$L_W = 10 \log_{10} \left(\frac{P}{P_0} \right) [\text{dB}]. \quad (2.13)$$

The reference power is often chosen to be 1 pW. [10]

2.5.3 Directivity

In this application, especially the directivity of a sound source is of high importance. The directivity of a sound source is quantified by its directivity factor Q_D . This factor is defined as [6]

$$Q_D = \frac{|p_{axis}|^2}{\rho_0 c} \frac{4\pi r^2}{W} = \frac{4\pi p_{rms}^2}{2\pi \int_0^\pi p_{rms}^2(\theta) \sin \theta d\theta}. \quad (2.14)$$

2.5.4 Sound Power Level vs. Sound Pressure Level

The sound pressure level at any point can be calculated through the distance to the source r , the sound power level of the source and the directivity. It is given, according to [4], as

$$L_p = L_W + 10 \log_{10} \left(\frac{Q_D}{4\pi r^2} + \frac{4}{R_c} \right). \quad (2.15)$$

Where R_c is the room constant defined as

$$R_c = \frac{S\alpha_{av}}{1 - \alpha_{av}} [m^3]. \quad (2.16)$$

Here S stands for the volume and α_{av} is the average absorption coefficient of the room. If the room is large enough ($S \gg 1$) then $R_c \gg 1$ follows which means Equation 2.15 can be simplified to

$$L_p = L_W + 10 \log_{10} \left(\frac{Q_D}{4\pi r^2} \right) = L_W + 10 \left(\log_{10}(Q_D) - \log_{10}(r^2) - \underbrace{\log_{10}(4\pi)}_{\approx 1.1 B} \right). \quad (2.17)$$

3

Parametric Loudspeaker Array

3.1 Introduction

Parametric loudspeaker arrays are transducer arrays, which use the demodulation of ultrasound in air to generate a highly directional and steerable sound source. This effect was first discovered by Westerveld [16] and used in a loudspeaker, the *Audio Spotlight*, by Masahide Yoneyama and Jun-ichiroh Fujimoto [17].

Why ultrasound generates a higher directivity is shown in Section 3.2, how the demodulation works in Section 3.3 and how signal processing methods can be applied to improve the audio quality and enable beam steering capabilities are shown in Section 3.4 and Section 3.5.

3.2 Directivity of Ultrasonic Transducers

As mentioned, ultrasonic transducer produce a highly directional beam. In this chapter the mathematical foundations for this effect will be laid. The simplest model of a transducer is to think of it as a slit on which an incoming planar sound wave is diffracted on.[12]

The far-field air pressure of a transducer, explained in Section 2.4.1, can be calculated in relation to the angle and distance as

$$p(\varphi, r) = \frac{A}{r} \underbrace{\frac{\sin\left(\frac{ka \sin \varphi}{2}\right)}{\frac{ka \sin \varphi}{2}}}_{D_T(\varphi)}. \quad (3.1)$$

Where the sinc function is better known as the directivity of the transducer.

$$D_T(\varphi) = \frac{\sin\left(\frac{ka \sin \varphi}{2}\right)}{\frac{ka \sin \varphi}{2}}. \quad (3.2)$$

This directivity for an acoustic wave radiated by a loudspeaker size of $a = 16$ mm is shown in Figure 2.5. It is important to keep in mind that this directivity is just a model and the true directivity of a transducer can vary massively depending on its geometry, especially as the angle increases.

3.3 Demodulation Process

The most fundamental equation for modeling the non linear behaviour of air is the Khokhlov, Zabolotskaya and Kuznetsov (KZK) equation [5] given as

$$\frac{\partial^2 p}{\partial z \partial \tau} = \frac{c_0}{2} \nabla_r^2 p + \frac{\delta}{2c_0^3} \frac{\partial^3 p}{\partial \tau^3} + \frac{\beta}{2\rho_0 c_0^3} \frac{\partial^2 p^2}{\partial \tau^2}, \quad (3.3)$$

of which analytical solutions cannot be calculated. However, the solution can be approximated by first solving for the linear ultrasonic field p_1 by setting the nonlinear term to zero $\beta = 0$ and then solve for the nonlinear solution p_2 . The final solution is then the superposition of these two fields $p = p_1 + p_2$. The ultrasonic field p_1 is described by

$$\frac{\partial^2 p_1}{\partial z \partial \tau} = \frac{c_0}{2} \nabla_r^2 p_1 + \frac{\delta}{2c_0^3} \frac{\partial^3 p_1}{\partial \tau^3} \quad (3.4)$$

and this can now analytically be solved. This solution then can be used as an approximation for the non linear part of the equation

$$\frac{\partial^2 p_2}{\partial z \partial \tau} = \frac{c_0}{2} \nabla_r^2 p_2 + \frac{\delta}{2c_0^3} \frac{\partial^3 p_2}{\partial \tau^3} + \frac{\beta}{2\rho_0 c_0^3} \frac{\partial^2 p_1^2}{\partial \tau^2} \quad (3.5)$$

which can now be solved near the axis analytically. The resulting function p_2 turns out to be

$$p_2 = \frac{\beta P_0^2 a^2}{16\rho_0 \alpha c_0^4 a^2} \frac{d^2}{dt^2} E^2(t). \quad (3.6)$$

Where $E(t)$ is the enveloping of the output signal of the transducers. This means that the pressure of the wave in the far-field is proportional to the second derivative of the squared modulated signal.

$$p_2 \propto \frac{d^2}{dt^2} E^2(t). \quad (3.7)$$

If the pressure now should be the desired audio signal $f(t)$, the envelope of the output signal of the transducers has to be

$$E(t) = \sqrt{\left(1 + m \int \int f(t) dt^2\right)} = \sqrt{(1 + x^2)}. \quad (3.8)$$

But it transpires that this is impossible to accomplish because of the limited bandwidth of the ultrasonic transducers. If $E(t)$ is written as a Taylor approximation

$$E(t) = \frac{1}{A} \left(1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \dots\right) = 1 - \sum_{k=0}^{\infty} \frac{2}{k+1} \binom{2k}{k} \left(-\frac{x}{4}\right)^{k+1}. \quad (3.9)$$

The spectrum $E_T(\omega)$ of the optimal signal is

$$E_T(\omega) = \frac{1}{A} \left(1 + \frac{1}{2}X(w) - \frac{1}{8}(X(w) * X(w)) \pm \dots\right). \quad (3.10)$$

This shows that $E_T(\omega)$ has an infinite bandwidth, because each correlation in frequency, or multiplication in time, doubles the bandwidth. So an approximation for the envelope has to be found.

3.4 Modulation Process

Due to the finite bandwidth of transducers and the infinite bandwidth of the ideal envelope, as shown in Section 3.3, an approximation for the envelope has to be made. Figure 3.1 shows the modulation and demodulation process.

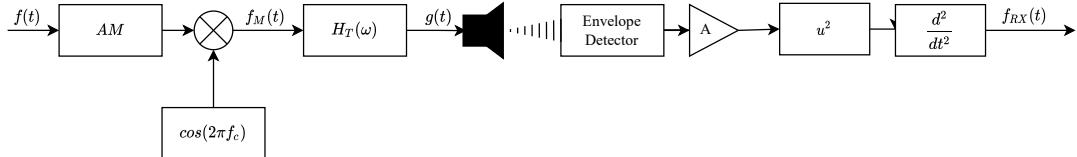


Figure 3.1: Block Diagram of Modulation and Demodulation Process

Following, two possible modulation methods are presented.

3.4.1 Amplitude Modulation

One possible approximation is to use regular amplitude modulation. The emitted signal $g_{AM}(t)$ would be

$$g_{AM}(t) = h_T(t) * (1 + mf(t)). \quad (3.11)$$

Often $H_T(\omega)$ is approximated to be $\frac{1}{s^2}$, due to the narrow bandwidth of the PUT, which would cancel out the second derivative

$$f_{RX}(t) = Ag_{AM}^2(t) = A(2mf + m^2f^2(t)) = \underbrace{2Amf(t)}_{\text{Signal}} + \underbrace{2Am^2f^2(t)}_{\text{Distortion term}}. \quad (3.12)$$

If the simplification of $H_T(\omega)$ is not made the spectrum of the received signal becomes

$$F_{RX}(\omega) = 2A(mF(\omega)H(\omega) + m^2(F(\omega) * F(\omega))H(\omega)). \quad (3.13)$$

As can be seen the modulation index m is squared inside of the distortion term and only linear in the signal. This means that if the modulation index would be chosen small enough the distortion term would vanish, but the power of the signal would also be reduced significantly.

3.4.2 Modified Amplitude Modulation

As seen, if AM is used, there is a problematic distortion term. Modified Amplitude Modulation (MAM) uses a similar idea to Quadrature Amplitude Modulation (QAM) to get rid of this problem [13]. As inphase component the input signal $1 + mf(t)$ with a DC offset is used and as the quadrature component the signal $\sqrt{1 - m^2f^2(t)}$ is used. If now the output signal of the modulation $f_M(t)$ is calculated, as described in 2.2, it turns out to be

$$f_M(t) = \sqrt{2 + 2mf(t)} \sin \left(2\pi f_{ct} t + \arctan \left(\frac{\sqrt{1 - m^2f^2(t)}}{1 + mf(t)} \right) \right). \quad (3.14)$$

If $H_T(\omega)$ is now again assumed to be $H_T(\omega) = \frac{1}{\omega^2}$, the signal turns out to be exactly what is should be.

This modulation method seems to be optimal, but again because of the square root in the quadrature component it cannot be produced due to the limited bandwidth of

ultrasonic transducers. However, the basic idea still can be used. This is done by approximating the distortion terms with a Taylor series

$$Q(t) = \sqrt{1 - m^2 f^2(t)} = \sum_{i=0}^{\infty} \frac{(2i)!}{(1-2i)i!^2 4^i} m^{2i} f^{2i}(t) \approx \sum_{i=0}^k \frac{(2i)!}{(1-2i)i!^2 4^i} m^{2i} f^{2i}(t). \quad (3.15)$$

Depending on the frequency response of the transducers the degree of the approximation k can be chosen. The higher the degree of approximation becomes, the higher the bandwidth of the transducers has to be.

3.5 Array Signal Processing

The theory in this section is mostly taken from the book Fundamentals of Ultrasonic Phased Array [12].

3.5.1 Phased Array Beam Model

To explain phenomena such as beam steering and beam focusing the phased array beam model has to be introduced. Figure 3.2 shows the basic setup of a transducer array

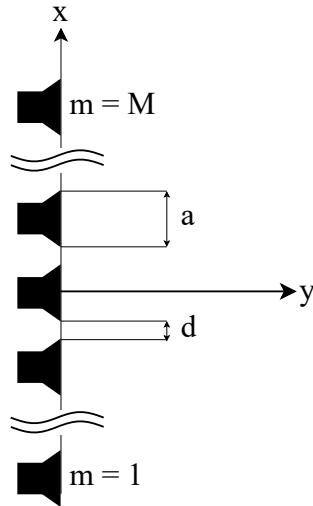


Figure 3.2: Transducer Arrangement

with M elements, where M is odd. The position of the m th element is given as [12]

$$x_m = \left(\frac{2m - 1 - M}{2} \right) \underbrace{(d + a)}_s = \left(\frac{2m - 1 - M}{2} \right) s, \quad (3.16)$$

where d is the distance between the transducers, a is the size of the transducers and s is known as the pitch of the array. The far field pressure of a single element can be calculated as [12]

$$p_m(r_m, \omega) = \underbrace{\rho c V_0 \frac{ka}{2M} \sqrt{\frac{2}{\pi i}}}_{A} D_T(\varphi_m) \frac{e^{jk\frac{a}{2}r_m}}{\sqrt{k\frac{a}{2}r_m}}, \quad (3.17)$$

where

$$r_m = \sqrt{\left(\frac{2x}{a} - \frac{2x_m}{a} \right)^2 + \left(\frac{2y}{a} \right)^2} \quad (3.18)$$

and

$$\varphi_m = \sin^{-1} \left(2 \frac{x - x_m}{ar_m} \right). \quad (3.19)$$

If now each element gets its own weighting factor C_m and a phase delay Δt_m the pressure can be calculated as

$$p(r_m, \omega) = A \sum_{m=1}^M C_m e^{j\omega\Delta t_m} D_T(\varphi_m) \frac{e^{jk\frac{a}{2}r_m}}{\sqrt{k\frac{a}{2}r_m}}. \quad (3.20)$$

Far Field

If only the far field is of interest. Then r_m can be simplified to [12]

$$r_m = R - x_m \sin \varphi, \quad (3.21)$$

and φ_m to

$$\varphi_m = \varphi. \quad (3.22)$$

So the pressure can be approximately written as

$$p(R, \varphi, \omega) = AD_T(\varphi) \underbrace{\frac{e^{jkR}}{\sqrt{kR}} \sum_{i=0}^M C_m e^{j\omega\Delta t_m} e^{-j k x_m \sin(\varphi)}}_{P(R)} \quad (3.23)$$

$$= AD_T(\varphi) P(R) \sum_{i=1}^M C_m e^{j(\omega\Delta t_m - k x_m \sin(\varphi))}. \quad (3.24)$$

Or with 3.16 as

$$p(R, \varphi, \omega) = AD_T(\varphi) P(R) \underbrace{\sum_{i=1}^M C_m e^{j(\omega\Delta t_m - k(\frac{2m-1-M}{2})s \sin(\varphi))}}_{D_s(C, \Delta t, \varphi)}. \quad (3.25)$$

This is the main model used to explain the directivity pattern of parametric transducer arrays. The sum can be understood as an array directivity D_s introduced by the delays and weights.

As an example the weights are set to $C_m = 1$ and the delays to $\Delta t_m = 0$. This leads to

$$D_s(\varphi) = e^{jks(\frac{M+1}{2}) \sin \varphi} \sum_{m=1}^M \left(e^{-jks \sin(\varphi)} \right)^m. \quad (3.26)$$

This can be seen as a geometric series which can be written as

$$\sum_{m=1}^M \left(e^{-jks \sin(\varphi)} \right)^m = e^{-jks \sin(\varphi)} \frac{1 - e^{-jks \sin(\varphi)M}}{1 - e^{-jks \sin(\varphi)}} \quad (3.27)$$

$$= e^{-jks(\frac{M+1}{2}) \sin(\varphi)} \frac{\sin \frac{Mks \sin(\varphi)}{2}}{\sin \frac{ks \sin(\varphi)}{2}}. \quad (3.28)$$

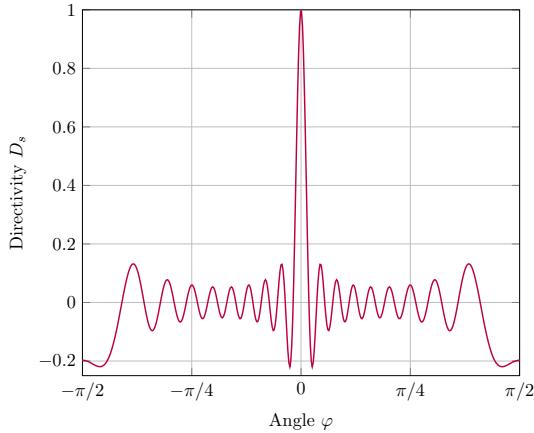
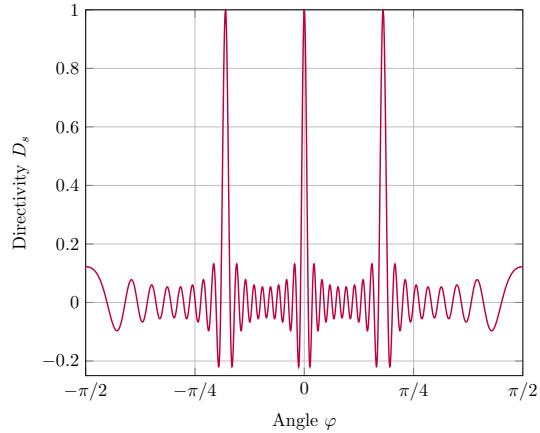
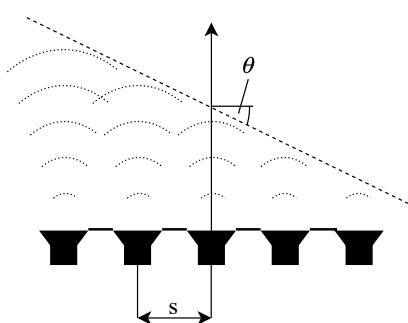
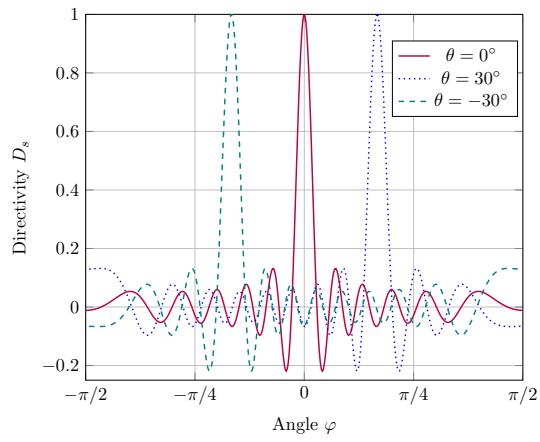
The array directivity can be described as

$$D_s(\varphi) = \frac{\sin \frac{Mks \sin(\varphi)}{2}}{\sin \frac{ks \sin(\varphi)}{2}}. \quad (3.29)$$

Since $k = \frac{2\pi}{\lambda}$, if $s = \lambda$ the directivity turns out to be

$$D_s(\varphi) = \frac{\sin(M\pi \sin(\varphi))}{\sin(\pi \sin(\varphi))}, \quad (3.30)$$

in which the numerator and the denominator are only equal at zero and 2π . So there is only one main lobe in the range between $-\pi/2$ and $\pi/2$, as seen in Figure 3.3. If this is not the case there are multiple lobes with the size of the main lobe, as seen in Figure 3.4.

**Figure 3.3:** Array Directivity $s = \lambda$ **Figure 3.4:** Array Directivity $s = 2\lambda$ **Figure 3.5:** Basic Idea of Beam Forming**Figure 3.6:** Array Directivity with Beam Steering

3.5.2 Beam Steering

The basic idea of beam steering is to delay the different channels in such a way that the wave fronts create a certain angle. This can be seen graphically in Figure 3.5. If the delays are

$$\Delta t_m = \frac{s \sin \theta}{c} \left(\frac{2m - 1 - M}{2} \right), \quad (3.31)$$

where θ is the angle to steer and the weights $C_m = 1$ are inserted into Equation 3.25. The array directivity D_S turns out to be [12]

$$D_S(1, \Delta t, \varphi) = \frac{\sin \frac{Mks(\sin(\varphi) - \sin(\theta))}{2}}{M \sin \frac{ks(\sin(\varphi) - \sin(\theta))}{2}}. \quad (3.32)$$

This shows that the beam steering just moves the the directivity calculated in 3.29 around. This can be seen in Figure 3.6 for different steering angles.

3.5.3 Beam Focusing

To focus the beam at a certain distance R_0 the delays have to be chosen as

$$\Delta t_m = \frac{s^2}{2R_0 c} (m - 1)(M - m). \quad (3.33)$$

This generates delays which are valued in a parabolic shape. This ensures that the waves meet exactly at the focal point, as shown in Figure 3.7.

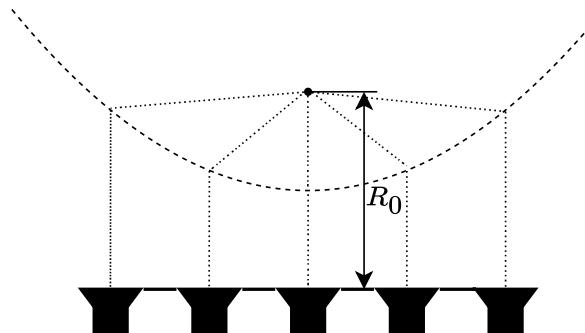


Figure 3.7: Basic Idea of Beam Focusing

3.5.4 Array Amplitude Weighting

As explained in Section 2.4.2 the diffraction process of a grid can be calculated via the Fourier transform. If an array of transducers is used, the function $s(x)$ is a rectangular sequence. If this rectangular sequence is now sampled with a sampling width of d then the signal becomes a rectangular window known from Finite Impulse Response (FIR) filters. The main difference being that the time axis is replaced by a spatial axis.

If now weights are applied to the different channels, the rectangular window can be changed to other known window functions, such as Hamming or Hann. The same theories will hold for the main lobe and sidelobes. However, the most commonly used window in array signal processing is the Dolph-Chebyshev window.

Dolph-Chebyshev Window

The Dolph-Chebyshev window is a special window which minimizes the so called Chebyshev norm of the sidelobes for a given main lobe width. The Chebyshev norm is the maximum absolute value

$$\min_{\omega, \sum \omega=1} \{ \max [| \text{Side lobes}(W(\omega)) |] \}. \quad (3.34)$$

The transform of the window can be written as

$$W(\omega_k) = \frac{\cos [M \cos^{-1} (\beta \cos (\frac{\pi k}{M}))]}{\cosh [M \cosh^{-1} (\beta)]} \quad k = 0, 1, 2, \dots, M - 1 \quad (3.35)$$

Where M is the number of taps of the window and β can be used to control the sidelobe level. If the Inverse Discrete Fourier Transform (IDFT) of $W(\omega_k)$ is taken, the Dolph-Chebyshev window $w(n)$ results. The controlling of the side lobe level is often done by

introducing another variable α which is connected to β in the following way

$$\beta = \cosh \left[\frac{1}{M} \cosh^{-1} (10^\alpha) \right]. \quad (3.36)$$

The maximum side lobe level is now given as

$$\text{Side lobe level} = L_s = -20\alpha[\text{dB}]. \quad (3.37)$$

Whereas the main lobe width is given as

$$\omega_c = 2 \cos^{-1} \left(\frac{1}{x_0} \right) \quad x_0 = \cosh \left[\frac{\cosh^{-1} (10^\alpha)}{M-1} \right]. \quad (3.38)$$

It can be seen that the higher α is chosen, the lower the maximum of the side lobes get. However, the main lobe gets wider.

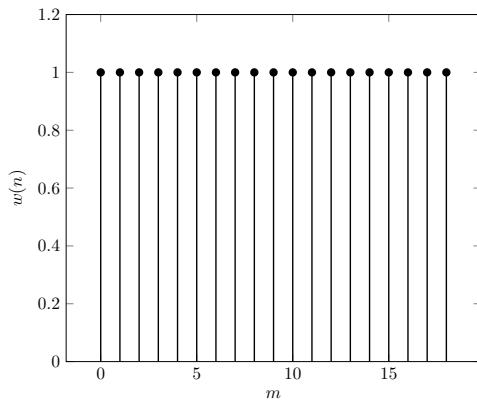


Figure 3.8: Channel Weights: Rectangular

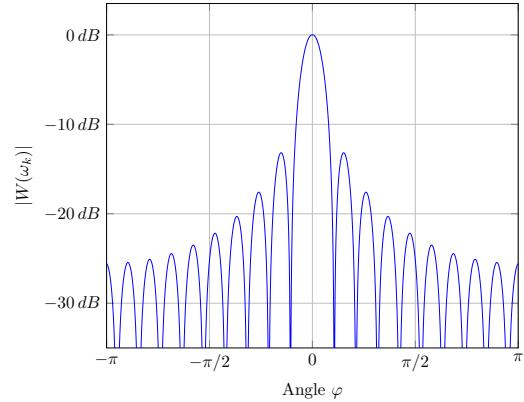


Figure 3.9: Directivity Pattern: Rectangular

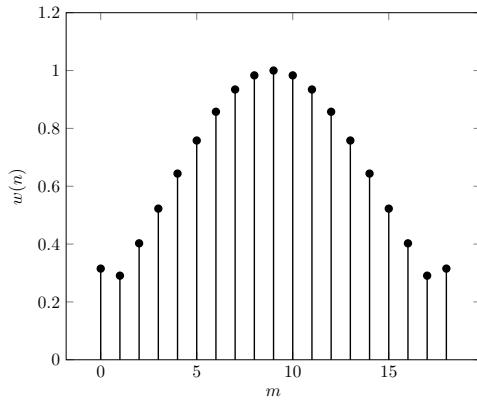


Figure 3.10: Channel Weights: Dolph-Chebyshev

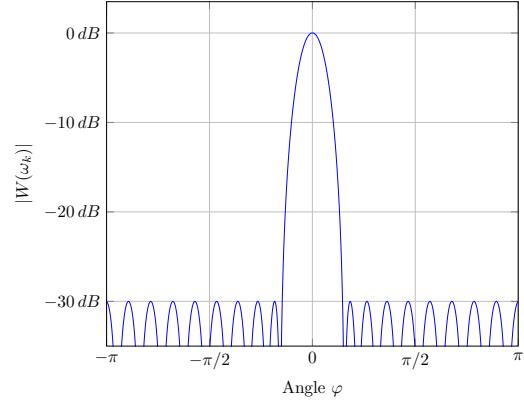


Figure 3.11: Directivity Pattern: Dolph-Chebyshev

4

Design

4.1 Overview

This section covers the complete development process including the hardware, gateware, software and mechanical design of the project. It is important to note that the following documentation concentrates only on the final version of the device. Earlier hardware prototypes are not covered due to the lack of relevance.

4.1.1 Key Requirements

The main focus of the development is to design a professional looking, easy to use and eye-catching device for demonstration purposes. The project name *Audio-Beamformer* has been chosen as it is easy to remember and has potential to be seen as a trademark.

The following key requirements have been set:

- Single power adapter or power cable (e.g. no need of labor power supplies)
- Easy to install (e.g. montage on a camera tripod)
- Intuitive to operate via state-of-the-art graphical user interface
- Multiple audio streaming sources such as Bluetooth and USB input devices
- Great scalability and flexibility of the hardware and software design

4.1.2 Key Decisions

In the conceptional phase of the development, several key decisions had to be made. This contains mainly the signal flow and the division between the processing part on the Raspberry Pi and the FPGA. Further, the question had to be evaluated, if a built-in power supply or an external power adapter is preferred. And most importantly, which type of ultrasonic transducer should be used in the design. In addition, the overall dimension and scale of the final product had to be discussed. In general, most of these decisions were made according to results of simulations, physical measurements and after extensive discussions. In the following sections, each part of the project is explained in detail.

4.2 Hardware Design

The hardware of the Audio-Beamformer was designed using Altium Designer 22. The integrated 3D CAD functionality simplified the overall development and lowered the possibility of errors in the design. The hardware has been improved over several iterations until a final version was built.

The 2-Layer Printed Circuit Board (PCB) with the size of 300.0 mm x 376.0 mm has been manufactured and assembled by JLCPCB, as seen in Figure 4.1.

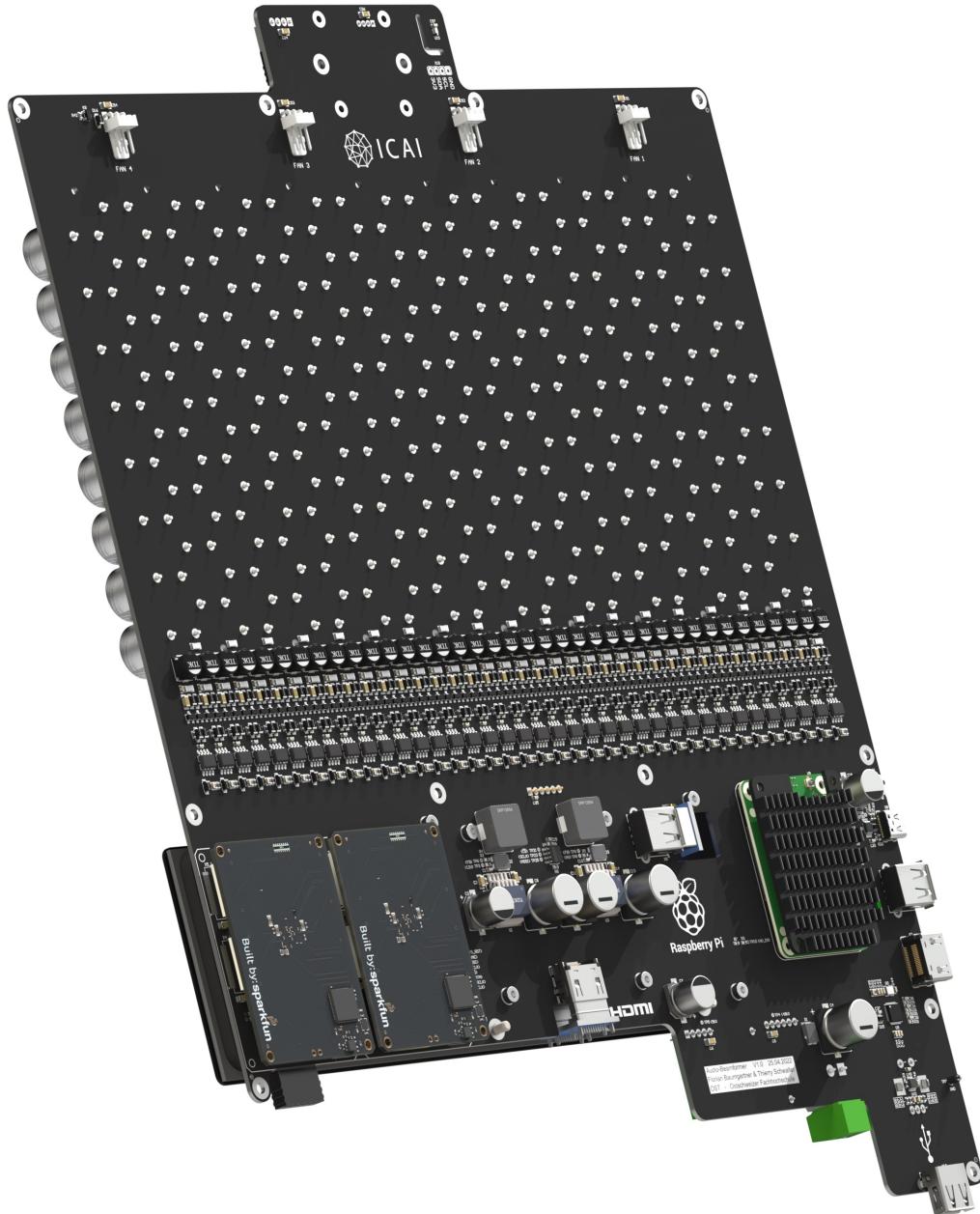


Figure 4.1: 3D-Render of PCB

4.2.1 Block Diagram

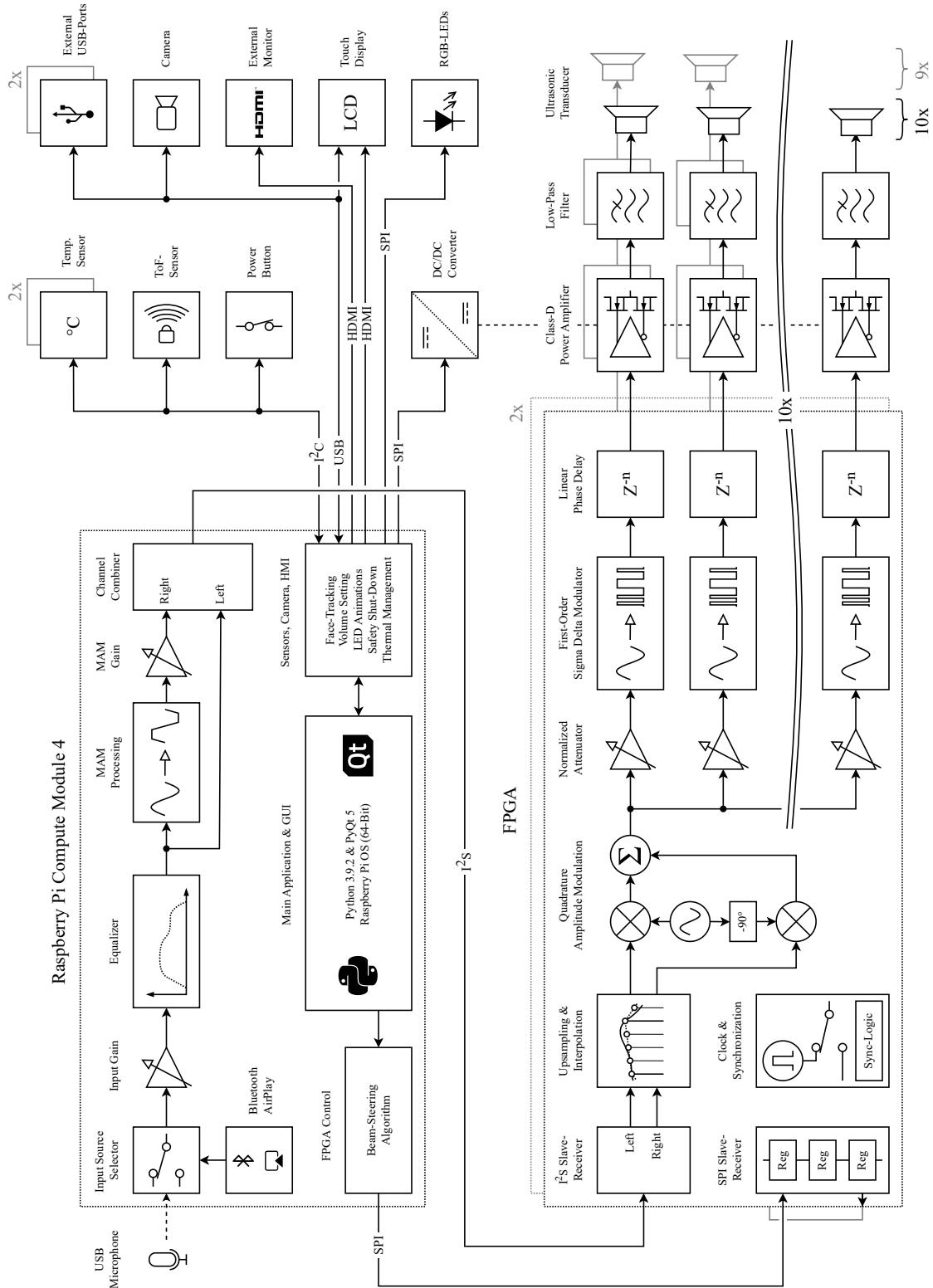


Figure 4.2: System Block Diagram

4.2.2 Power Management

The Audio-Beamformer is powered directly by mains voltage. As a connector, the widely used C14 (IEC 60320) has been chosen. Because of the metallic casing, protective earth is required, which has been connected directly to the back panel. The main switched mode power supply is made by *Mean Well* and delivers 48 V DC and up to 163 W output power. The LSP-Series are specially designed for low-profile applications and therefore ideal for this use case.

The simplified power management diagram in Figure 4.3 provides a better overview of how each voltage rail is created.

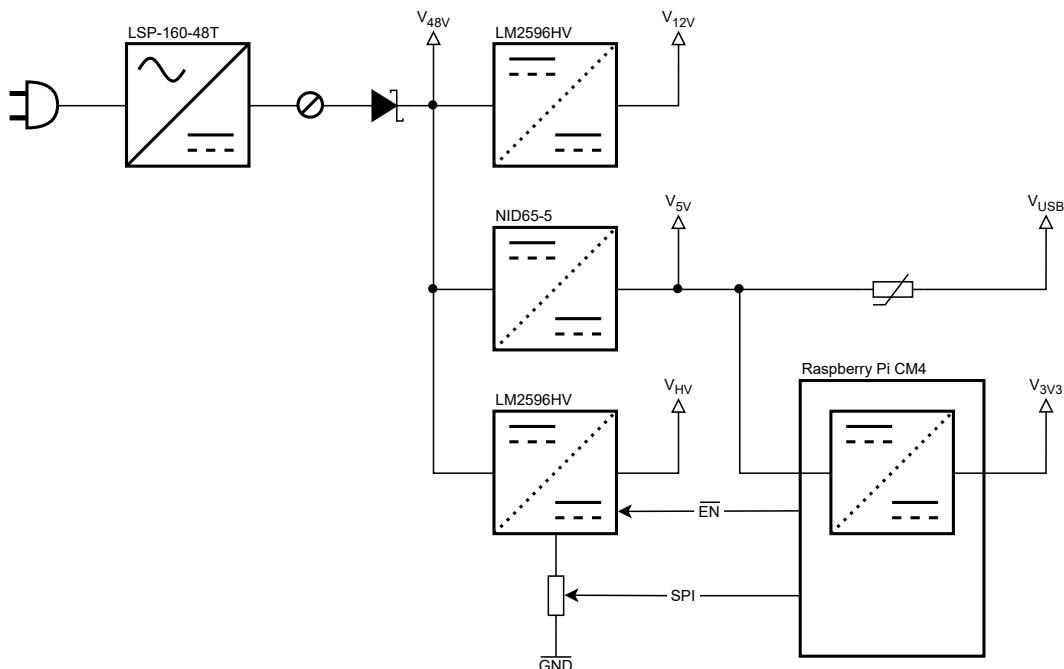


Figure 4.3: Simplified Power Management

As a reversed polarity protection a Schottky diode has been placed directly after the input connector. Further DC-DC buck converters generate fixed 5 V (6.5 A), 12 V (2.0 A) and a variable HV rail.

To adjust the physical output volume of the ultrasonic transducers, the amplifier drive voltage must be changed accordingly. The HV voltage supply can be set between 5.2 V and 23.5 V (2.0 A). This was achieved by using a digital potentiometer (MCP41010T) which is controlled by an SPI-Interface.

For safety reasons, the HV voltage is turned off per default and must be enabled by a physical logic signal provided by the Raspberry Pi.

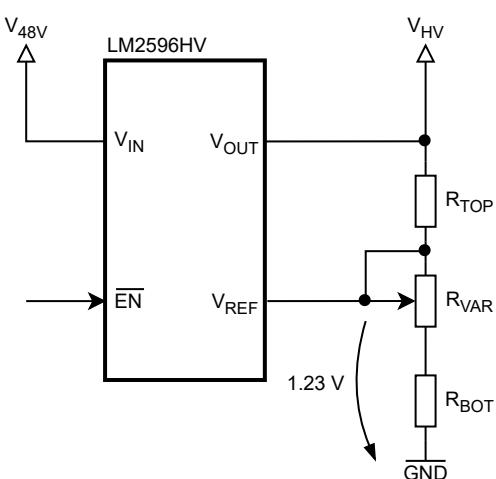


Figure 4.4: Variable Buck-Converter

The output voltage can be calculated as

$$V_{\text{HV}}(R_{\text{VAR}}) = \frac{V_{\text{REF}}}{R_{\text{BOT}} + R_{\text{VAR}}} (R_{\text{BOT}} + R_{\text{TOP}} + R_{\text{VAR}}). \quad (4.1)$$

Where R_{VAR} is proportional to its 8-bit value, d ($0x00 \cong 0\Omega$ and $0xFF \cong 10\text{k}\Omega$). The reference voltage of the DC-DC buck converter (LM2596HV) is 1.23 V. The resistor values used in the design are $R_{\text{TOP}} = 39\text{k}\Omega$, $R_{\text{BOT}} = 2.0\text{k}\Omega$ and $R_{\text{VAR}} = 10\text{k}\Omega$. This leads to the following approximation

$$V_{\text{HV}}(d) \approx 5.2\text{ V} + d \frac{20\text{ V}}{255}. \quad (4.2)$$

4.2.3 Raspberry Pi Compute Module 4

As a main processing platform the Raspberry Pi Compute Module 4 has been chosen due to its powerful quad-core processor and the great software support based on a large community. The exact model used in the design has 4 GB of RAM and fixed 16 GB of embedded eMMC flash storage. This has the advantage of being more reliable than systems that are dependent on SD-Cards. To increase the performance, the Raspberry Pi has been overclocked to 1.0 GHz. Sufficient cooling is provided by a heat sink and four cooling fans. The IO operating voltage has been set to 3.3 V.

4.2.4 FPGA

There are two FPGAs used in the design for generating the drive signals for the ultrasonic transducers. Each FPGA is mounted on a development board called *Alchitry Cu*. They are installed as daughter-boards on the main PCB by high-speed board-to-board connectors. Both FPGAs receive the audio stream via an I²S-Stream and get controlled by a simple SPI-Protocol in an daisy-chain configuration. More details on this in section 4.3.4 and 4.3.5.

The FPGA boards are powered by 5 V.

4.2.5 Sensors and HMI

The Audio-Beamformer contains several sensors that are connected by different interfaces to the Raspberry Pi Compute Module. The Human Machine Interface (HMI) enables easy access to change various settings of the device.

Camera

To be able to direct the sound towards a specific person, a camera is needed. In this case a USB-Camera has been chosen since it is easy to connect and does not need any special device drivers. The type *ELP-USB500W02* provides a resolution of 1280 x 720 pixels and a frame rate of up to 30 FPS. The optics used in this application are designed to match the viewing angle of the camera to the maximal beam-steering angle of the Audio-Beamformer. In this particular case, a camera lens with a focal length of 4.2 mm has been chosen which results in a viewing angle of approx. $\pm 40^\circ$.

However, the camera had to be slightly modified. On startup, it immediately tries to establish an USB connection to the host device. If there is no response during the first 500 ms, the camera goes into a power-down/suspend mode. In this state it cannot

be recognized by the host until it gets power-cycled. This issue has been solved by increasing the RC time constant of the camera internal power-on reset circuitry from 4.7 ms to ca. 4.7 s. In addition a Schottky diode has been added to guarantee a fast discharge time after powering down the device. This ensures a valid reset-pulse even if the device gets power-cycled very rapidly.

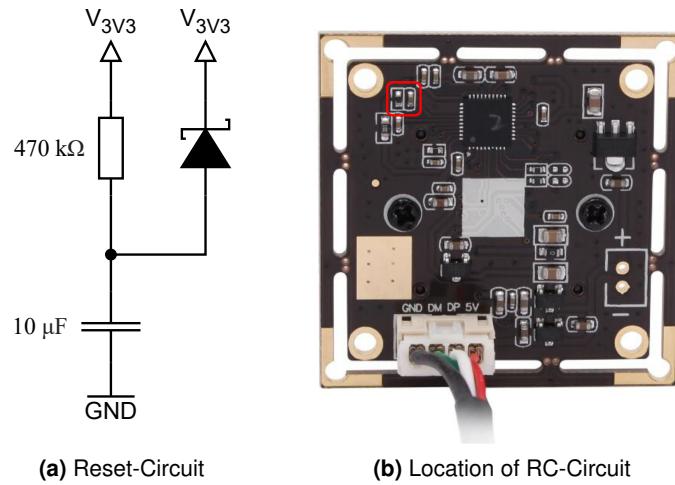


Figure 4.5: Camera Modification

Temperature Sensor

There are two temperature sensors embedded on the PCB. One to measure the ambient temperature (and therefore be able to calculate the speed of sound in air according to the temperature, described in 4.7) and the second-one to measure the system temperature which is used to control the speed of the four DC fans. The temperature sensors used in the design are of the type *TMP112* made by Texas Instruments. They have an accuracy of ± 0.5 °C and are connected to the I²C-Bus of the system with individual IDs (0x48 & 0x49). The temperature is read every 500 ms.

Time-of-Flight Sensor

For various safety reasons, as seen in Chapter 5, it is important to turn off the speaker output when a person enters the near-field of the Audio-Beamformer ($d < 1$ m). This safety mechanism has been realized by using a multi-zone Time-of-Flight (ToF) sensor. The *VL53L5CX*, made by ST Microelectronics is specially designed for a wide field of view (63°). It can measure distances up to 4 m and has a resolution of 8 x 8 zones. It is connected to the I²C-Bus and can be address by the ID of 0x52. The internal update rate has been set to 5 Hz to minimize traffic on the I²C-Bus. To further increase the data throughput, the I²C clock rate has been set to 1 MHz.

The distance-map data then gets further processed as described in Section 4.4.5.

Rotary Encoder

To easily adjust the volume of the Audio-Beamformer, a pushable rotary encoder has been placed below the display. Pressing the knob toggles the mute state of the output stage. This becomes very practical if the volume level should stay the same, but the speaker needs to be turned off temporarily. Both the volume level and the mute state can be overwritten in the GUI. This can only be achieved by the relative position measurement of rotary encoders and would not be possible with ordinary potentiometers.

To suppress contact bouncing and thus prevent false inputs, a simple RC debouncing circuit with a Schmitt-trigger buffer has been implemented.

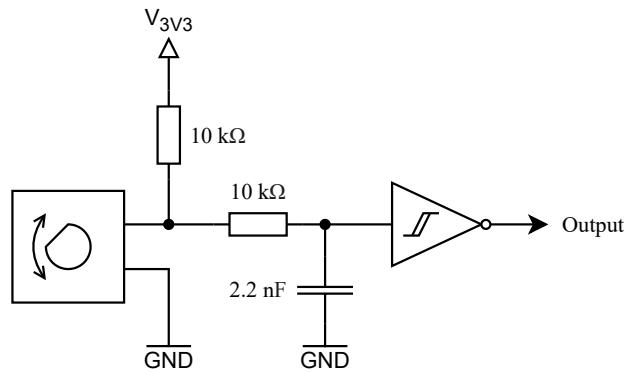


Figure 4.6: Rotary Encoder Debounce-Circuit

Power-Button and Cooling-Fans

On the right side of the Audio-Beamformer, a 22 mm stainless steel push button has been installed to power-on and off the Raspberry Pi Compute Module 4. The integrated RGB-LED ring is driven by a specialized PWM-Driver IC with I²C-Interface. The *PCA9633DP1*, made by Texas Instruments provides four individual addressable 8-Bit PWM channels and is addresses at the I²C-Address of 0x62. PWM-Channel 1, 2 and 3 are used for the red, green and blue LEDs of the bush button, while channel 0 is connected to the four DC-Fans installed in the back of the enclosure. The fans are wired in parallel and operate at a maximal voltage of 12 V.

The system temperature is used to control the speed of the cooling fans. They start running at a temperature of 40 °C and provide proportional air flow as the temperature increases. At 60 °C the operate at full speed.

LCD Touchscreen

Since the beginning of the project, the idea was to create an attractive and modern looking device. One of the key components is the large LCD touchscreen. It has a diagonal size of 11.9 inch and a resolution of 1480 x 320 pixels. HDMI is used for the video stream and USB to transmit the touch-screen data to the Raspberry Pi Compute Module 4. It is important to notice that the corners of the display are rounded (radius of 5 mm), this has been considered when designing the User Interface.

RGB-LEDs

To present a visual feedback of the current beam-steering angle and the active window-function, each row of the array contains one individually addressable RGB-LED on top and on the bottom of the ultrasonic transducers. This type of intelligent LEDs is called *APA102*. They embed an integrated driver IC which can be controlled via a non-standard SPI protocol (special start and end sequence instead of a chip select line) in a daisy-chain configuration. The LEDs have a physical size of 5.0 x 5.0 mm and are directly powered by the 5 V rail.

Next to the 38 LEDs used for the array channel illumination, a further 20 LEDs are placed on a ring around the camera. This gives direct visual insight into the face-tracking algorithm. While a person is tracked, the LEDs are animated in a "breathing" brightness motion. When there is no face detected, a spinning gradient animation is shown.

The maximal power consumption is about 25 W if all LEDs are fully turned on.

External Interfaces

The Audio-Beamformer can easily be connected to an external display by using the HDMI-Port on the left side of the enclosure. In combination with the two spare USB-Ports, e.g. for connecting a keyboard and mouse, the system can easily be debugged. Even developing new software features directly on the target platform is possible. To back-up the data of the eMMC flash storage on the Raspberry Pi Compute Module 4, a USB Type-C Port has been added. To start the back-up procedure, the tiny slide switch next to the USB-C Port must be set to the up-position. This enables the bootloader of the Raspberry Pi Compute Module 4 on power-up of the device. When connected to an external host (such as a computer), the Audio-Beamformer is recognized as a mass-storage-device. A back-up tool like e.g. *Win32DiskImager* can be used to create a binary image of the operating system inclusive all user data.

4.2.6 Output Stage

The output stage represents a Class-D design in a MOSFET full-bridge configuration. This topology has the advantage of high efficiency and low part count. However, the proper design of such output amplifiers has its difficulties. Especially when operated at high switching frequencies (3.125 MHz in this case), transient turn-on and turn-off times must be very short. This can lead to ringing and voltage spikes on the output signal. MOSFETs are known to be very sensitive when it comes to such transient voltage spikes. Exceeding the maximal Drain-Source voltage will likely result in a breakdown and a permanent short circuit. To prevent such behavior, several options were considered:

- Utilize MOSFETs with higher voltage rating
- Reduce switching speed and therefore increase transient time
- Impedance matching of load and driving source (e.g. with series resistor)
- Introducing passive dampening networks, such as RC-Snubber circuits
- Dump excess energy into bypass capacitors using clamping diodes

MOSFETs with higher Drain-Source voltage ratings have typically more gate capacitance and therefore would lead to a drastically higher switching current. A great compromise has been found with the type *BSS123*, it has a Drain-Source voltage rating of 100 V and can withstand a continuous current of 200 mA. The gate capacity is around 32 pF (at a gate voltage of 12 V).

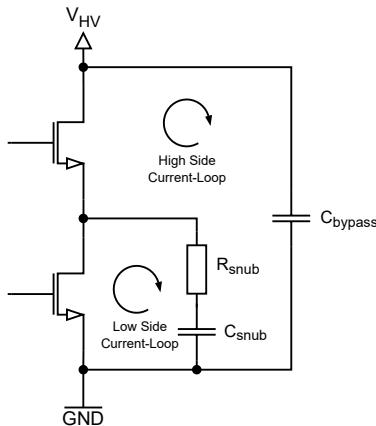


Figure 4.7: Class-D Half-Bridge with RC-Snubber

As mentioned above, RC-Snubber networks can help to suppress voltage spikes by creating a low-impedance path for the current, circling in the low-side of the half-bridge. Figure 4.7 shows how important a sufficient bypass capacitor is to stabilize the high-side. Texas Instruments published a very comprehensive guide on how to design such RC-Snubber networks [3].

The amplified digitally modulated output signal must be low-pass filtered in order to suppress the high frequency switching content. This can easily be achieved by the use of a series inductance. To keep the design symmetrical, on both half-bridge output signals a shielded inductor with an inductance of $220\ \mu\text{H}$ has been installed. The inductance value has experimentally, been evaluated by trying out different combinations of inductors and switch frequencies compared to the output noise level, as shown in Section 6.3.3.

4.2.7 Ultrasonic Transducer Array

Achieving a decent array size is important to get a better directivity, higher output volume and in general better beam steering properties. Thus a large amount of ultrasonic transducers are needed. In order to keep the cost low, one of the key criteria was the unit price. Directly ordering by professional suppliers, such as Digikey, Mouser, Farnell, etc. results in very high prices of more than 7 CHF per piece. This was not acceptable since more than 150 pieces are needed in the final design. Fortunately it was possible to get directly in touch with a manufacturer based in China. The datasheet of the *MA40A16* (attached in the appendix A.5) has been studied firmly, with the conclusion that the type is suitable for this project and satisfies all key parameters. The price per unit is only 0.3 CHF.

They operate at a resonant frequency of 40 kHz and have physical diameter of 16 mm. It is important to note that the metal case makes electrical contact to one of the pins. Thus it is important to leave enough clearance between the transducers to prevent short circuits.

Arrangement & Placement

As prior analysis showed in Section 3.5.1, the denser an array can be built, the better its performance is. Due to the circular shape of the ultrasonic transducers, the optimal arrangement is a hexagonal pattern. This has some further advantage of creating a smaller horizontal spacing between each row of approx. 14.75 mm (when a diagonal distance of 17.0 mm is chosen with a gap of 1 mm between the transducers). In general, a smaller spacing leads to higher efficiency and a better beam pattern (e.g. less dominant sidelobes). The final number of 19 rows has been the result of several considerations. First of all, the row count must be odd, to create a symmetrical design. Furthermore, the total width of the array must comply with the maximal manufacturing and assembly size of PCBs by JLCPCB (max. 480 x 320 mm). And at last, the overall dimension should match the size of the display to create a cleaner visual impression of the final design.

A row height of 8 has been chosen to create stronger beam-characteristics in the vertical direction. In addition, the parallel wiring of multiple ultrasonic transducers lowers the total impedance, which has the advantage of operating at a lower driving voltage to create the same amount of power.

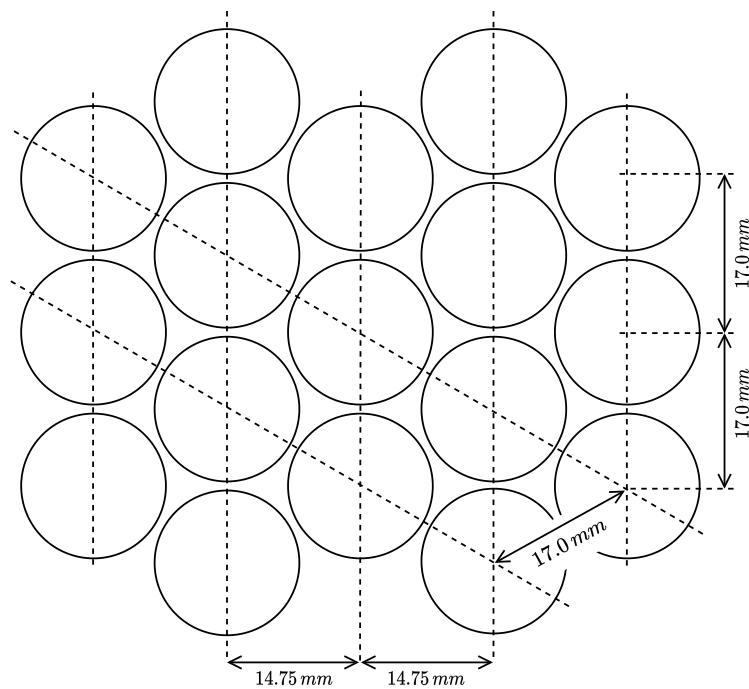


Figure 4.8: Ultrasonic Transducer Placement

Radiation Pattern Simulation

To get an insight into the potential radiation pattern of the Audio-Beamformer the parametric transducer array was simulated using Matlab's Phased Array System Toolbox. The results of this simulation are shown in Figure 4.9 and Figure 4.10.

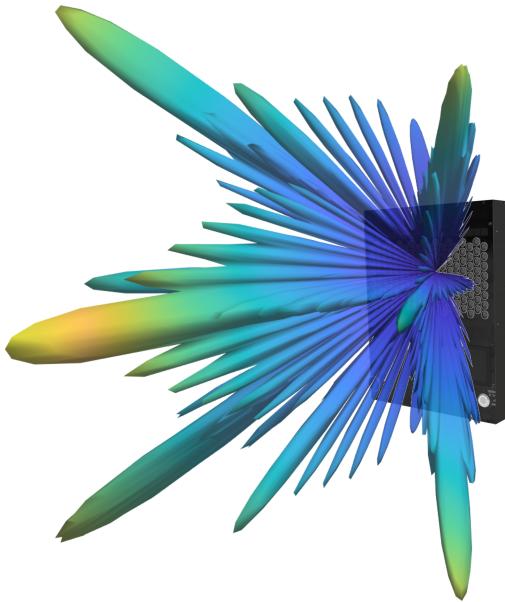


Figure 4.9: Radiation Pattern Simulation (Isometric-Perspective)

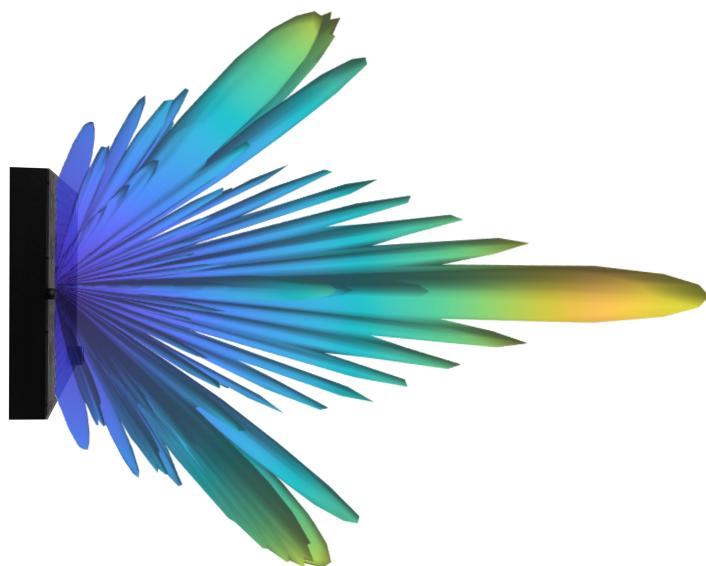


Figure 4.10: Radiation Pattern Simulation (Areal-Perspective)

4.3 FPGA Design

As mentioned in Section 4.2.4, the *Alchitry Cu* FPGA-Board was used. The specific chip is called *iCE40-HX8K*, made by Lattice Semiconductor. It offers a total of 7680 logic cells, 32 blocks of dual-port RAM (4 KBit each), two independent PLLs and 79 IO-Pins. It is operated at 100 MHz which is more than enough for this application.

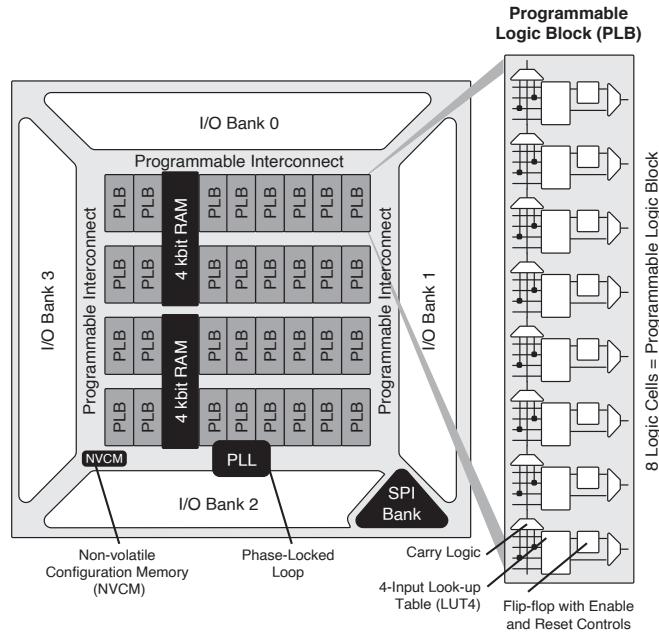


Figure 4.11: Lattice iCE40 FPGA Family Architecture [8]

To keep the design process as efficient as possible, a development environment has been chosen which is easy to use and offers lots of predesigned IP-Blocks. Namely the *IceStudio* has been used in combination with the open source tool-chain called *IceStorm* which currently supports all FPGAs of the iCE40-Family.

The tool offers a combination of a graphical design work-flow with the option of embedding custom blocks of Verilog code. Due to the open source nature, lots of individuals published extension libraries containing hundreds of IP-Blocks. This can be very practical, specially for integer arithmetic operations, such as addition and multiplication.

4.3.1 Integer Arithmetic

Signed integers with a fixed width of 16 bits have been chosen as the datatype used in this design. This has the advantage of making use of dedicated hardware accelerators built into the FPGA. The maximum range of a 16 bit integer reaches from -32,768 to +32,767. In order to make calculations easier, this range can be interpreted as normalized values between -1.0 (representing -32,767) and +1.0 (representing +32,767). Note that the most negative value of -32,768 should never be used, as the same positive value cannot be reached (preventing asymmetric behaviour).

Using normalized values simplifies particularly the multiplication operation, since signals can be attenuated or mixed without the risk of a sign flip or the values being clipped.

4.3.2 Signal Flow Diagram

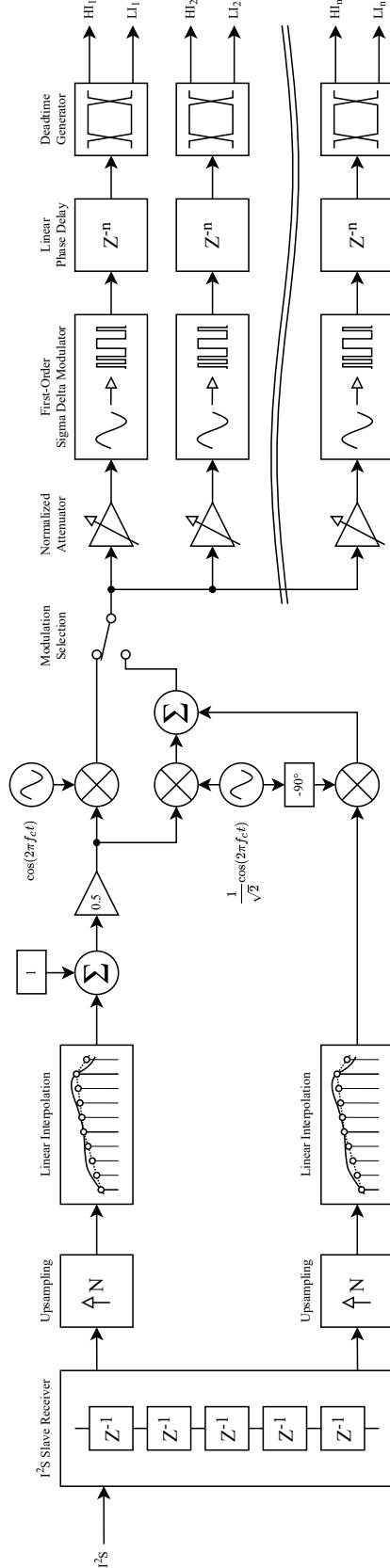


Figure 4.12: FPGA Signal Flow Diagram

4.3.3 Clock & Synchronization

Making sure that both FPGAs run simultaneously and do not drift apart, the system clocks must be shared. This means that one of the FPGAs acts as a master, which provides a clock signal to all slaves. In order to use the same logic configuration, a dedicated ID-Pin has been added to differentiate the master (logic level 0) and the slave (logic level 1).

The clock is initially created by an 100 MHz oscillator built onto the development board. With the help of a PLL it can be adjusted to the needs of the application. However, in this case it turned out to be a well functioning core speed and no further adjustments have been done. If a reduction of power consumption is needed, the clock speed could be decreased.

In addition, a synchronization pulse gets created by the Raspberry Pi Compute Module 4 at the startup of the device. This signal is used to reset the local oscillators on all FPGAs, to make sure that the phase of the carrier signal is in sync.

4.3.4 I²S Interface

The I²S-Interface is a well known and widely used protocol for real time digital audio streaming. It is mainly used for inter-chip communication (therefore its name of Inter-IC Sound). The interface consists of 3 signals: BCLK or just BCK (Bit Clock), LRCLK (Left-Right Clock) and the unidirectional data signal. The LRCLK is used for frame synchronization and indicates the current channel that is being transmitted (logic level 0 means left channel). The start of a frame is defined by the falling edge of the LRCLK signal. On the rising edge of the BCK signal, the current data bit is transmitted and fetched.

There exist various frame lengths, but the most common is 24 bit per channel, in a "left-aligned" manner. The default I²S configuration of the Raspberry Pi Compute Module 4 has been implemented on the FPGA. This means a sample rate of 44'100 kHz has been used. However, only the upper 16 bits of the 24 bit audio-stream get utilized for further processing, since a higher dynamic range is not needed. This has the advantage of simplifying the processing due to the use of fixed 16 bit integer arithmetic operations.

The implementation in the FPGA is straight forward. Basically a chain of D-Flip-Flops is created where the data is fed through. On the falling edge of the LRCLK signal, the data is then fetched and ready to be processed.

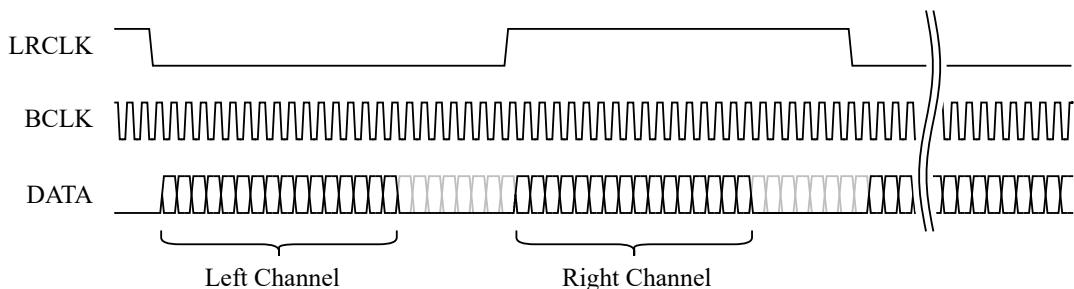


Figure 4.13: I²S-Interface Signals

4.3.5 SPI Interface

The Serial Peripheral Interface (SPI) is used for controlling the configuration of the FPGAs. The protocol is very simple and can easily be implemented as a chain of shift registers (consisting of cascaded D-Flip-Flops). This structure allows daisy-chain operation, where the output of the first FPGA is fed into the input of the second one. The interface consists of 3 signals: CS (Chip select), SCLK or just SCK (Serial Clock) and the data signal, often called MOSI (Master-Out → Slave-In). The data is shifted through the shift registers at each rising edge of the clock signal by one position. At the rising edge of the chip select signal, the current state then is fetched from the shift-register chain to the output.

A clock frequency of 8 MHz has been chosen to ensure fast update rates. In Table 4.1 the protocol is further described.

Byte Number	Data Type	Description
[0]	binary	[2:0] Interpolation: 1 ... 64 [3] Modulation Type: AM / MAM [7:4] Reserved
[2:1]	boolean	[9:0] Channel Enable (0 ... 9)
[4:3]	unsigned integer	Sigma-Delta Coefficient
[6:5]	unsigned integer	Channel 0 Delay
[8:7]	signed integer	Channel 0 Gain
[10:9]	unsigned integer	Channel 1 Delay
[12:11]	signed integer	Channel 1 Gain
...
[42:41]	unsigned integer	Channel 9 Delay
[44:43]	signed integer	Channel 9 Gain
[45]	binary	<i>Same as above, Byte 0</i>
[47:46]	boolean	[9:0] Channel Enable (19 ... 10)
[49:48]	unsigned integer	Sigma-Delta Coefficient
[51:50]	unsigned integer	Channel 10 Delay
[53:52]	signed integer	Channel 10 Gain
[55:54]	unsigned integer	Channel 11 Delay
[57:56]	signed integer	Channel 11 Gain
...
[87:86]	unsigned integer	Channel 19 Delay
[89:88]	signed integer	Channel 19 Gain

Table 4.1: SPI Protocol Description

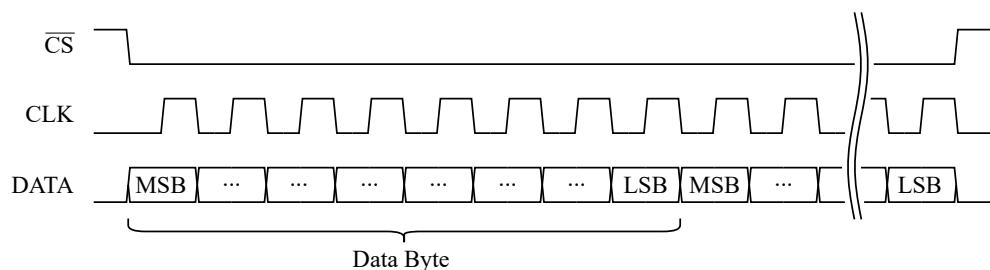


Figure 4.14: SPI-Interface Signals

4.3.6 Interpolation

To enhance the Signal-to-Noise Ratio (SNR), the incoming base band signal gets up-sampled to a sampling rate of 6.25 MHz. Oversampling improves the SNR by the following factor:

$$\Delta SNR \approx 0.5 \log_2 \left(\frac{6.25 \text{ MHz}}{44.1 \text{ kHz}} \right) 6.02 \text{ dB} = 21.5 \text{ dB} \quad (4.3)$$

The formula however implies that the signal is perfectly interpolated and reconstructed. This is definitely not the case if simple sample-and-hold methods are applied. To improve the signal quality, linear interpolation is used. The implementation allows to change the interpolation depth between 1 (no interpolation) and a maximum of 64. However in practice, there is not really a noise reduction noticeable when the linear-interpolation gets enabled or the interpolation depth gets increased. The audible noise is mostly caused by other factors, such as the sigma-delta-modulator, clock jitter, phase noise, etc.

4.3.7 Modulation

There are two modulation types implemented as discussed in Section 3.4: Regular amplitude modulation and modified amplitude modulation. Both modulation types make use of a mixer (multiplier), which multiplies the base band signal with the carrier oscillator, in this case a 40 kHz sine wave. This sinusoidal signal gets created by a Lookup-Table (LUT) which is stored in Block-RAM. The table is automatically generated by a Python script.

It is important to note that there are two different LUTs for each modulation type. The values for the regular amplitude modulation are fully scaled and calculated like this: $f_{AM}(t) = \cos(2\pi f_c t)$, where for the modified amplitude modulation a reduced amplitude is used of the factor: $1/\sqrt{2}$.

This prevents the case whereby a value of more than ± 1 occurs and could cause an arithmetic overflow

$$f_{MAM}(t) = Q(t) \frac{1}{\sqrt{2}} \cos(2\pi f_c t) + I(t) \frac{1}{\sqrt{2}} \cos(2\pi f_c t - \frac{\pi}{2}) \stackrel{!}{\leq} 1. \quad (4.4)$$

4.3.8 Sigma-Delta-Modulator

In order to keep the overall complexity as low as possible, the focus has been set onto the implementation of a first-order sigma-delta modulator. Higher order sigma-delta modulators tend to have better noise-shaping capabilities and thus will result in a lower SNR. The implementation complexity, however, increases drastically with higher orders, as e.g. specially tuned IIR-Filters are needed.

The first-order implementation is very straightforward. The core consists of an integrator (binary adder) and a feedback-loop constructed with a single D-Flip-Flop stage. Basically the output of the integrator tries to "follow" the input signal, similar to any control loop. The error (difference between the target and actual value) is the source of the modulating output signal. The MSB (sign bit) is used to determine if the error is positive or negative. This binary output single tends to toggle at a very high frequency (up to half the sigma-delta clock rate). The spectral view shows a clearly shifted noise level to the higher frequency range. When the output signal is low-pass filtered (by a

physical analog filter), the switching noise gets suppressed and the main low-frequency target signal shows up with a much higher SNR.

The steepness of the integrator can be adjusted by the "addition-factor". This coefficient has major impact on the spectral noise figure. It turned out to be rather complex to find an optimal value, the implementation allows to adjust the coefficient by software over the SPI-Interface. Extensive tests showed that in combination with the chosen analog filter (inductance value), a sigma-delta-modulator coefficient of 8,192 performs best.

As described in Section 4.3.6, the higher the over-sampling ratio, the greater the SNR. Increasing the clock rate has however the major disadvantage of creating much more switching losses. A clock frequency of 6.25 MHz has proved to be sufficient for both, low audible noise and reasonable switching losses. This results in an output bandwidth of $B = 3.125$ MHz.

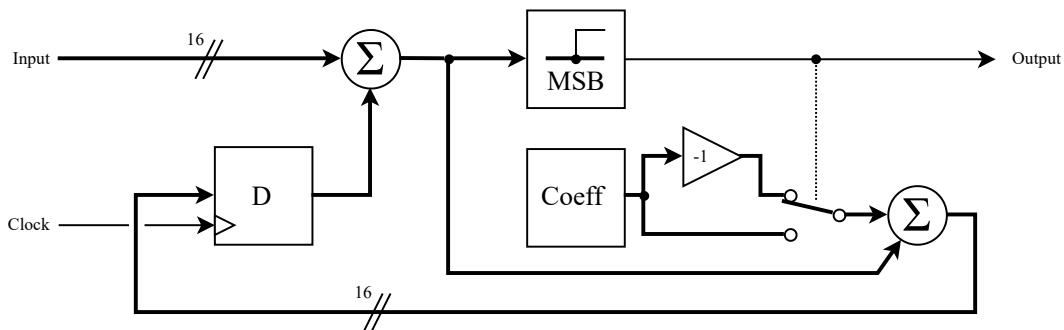


Figure 4.15: First-Order Sigma-Delta-Modulator

4.3.9 Channel Delay & Gain

Each channel can be attenuated and delayed individually. The gain-factor can either be positive or negative, which can be used to flip the phase polarity to 180°. This is needed for some window-functions. Note that for normal operation (no attenuation) a gain-factor of 1.0 (integer value of +32,767) must be set.

The delay line has been implemented as a ring buffer structure in Block-RAM. The dual port configuration is needed, since data is being written and read at the same time at different memory addresses. The signal flow diagram on Figure 4.12 shows that the delay-line block has been placed at the far end of the processing pipe-line. This means that only a single bit (namely the binary output signal of the sigma-delta-modulator) must be stored instead of a 16 bit integer vector.

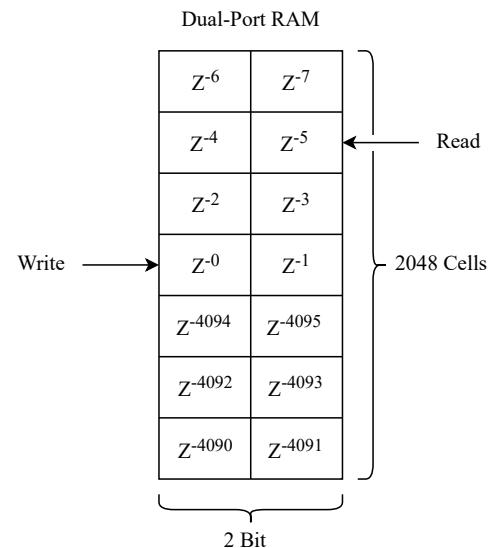


Figure 4.16: Delay Line in Block-RAM

Due to the maximal capacity of 4 kBit per RAM-Block, the signal can be delayed by a maximum of 4,092 ticks. However the configuration of those RAM-Blocks allows only a minimum bus-width of 2 bits, however this means that one memory cell must be filled with two single-bit samples. As a result, the signal can be delayed in 2,047 adjustable steps (0 ... 2,046). The tick time is derived by the Sigma-Delta-Modulator frequency, in this case: $\Delta t = 1/6.25 \text{ MHz} = 160 \text{ ns}$. This results in a maximal delay time of $t_{max} = 4092 \Delta t \approx 654.7 \text{ ns}$. In Section 4.4.4 the calculation of the delay time, based on the beam-steering angle, is further described.

4.3.10 Dead Time Generator

The FPGA directly drives the high and low side of the Class-D output stage. In any half-bridge design it is key to make sure that at no time, both the high-side and low-side MOSFETs are turned on at the same time. Violating this restriction, causes a temporary short circuit when switching. This can lead to inefficiencies and must be prevented in any case. The solution is very simple. Adding a fixed delay before turning on any MOSFET makes sure, that the complementary side is for sure not conductive anymore. This time is called "dead-time".

In the FPGA a dedicated dead-time-block has been implemented to create the exact timings for this particular hardware configuration. A dead time of 50 ns has been chosen as a good trade-off since it leaves enough timing margins, but also ensures high efficiency. If the dead time is set to high, the output power will decrease significantly. In addition the SNR will become smaller and harmonic distortion will become noticeable.

The dead time generator block also provides a possibility to shut down the output completely. This can be convenient for debugging purposes or to show the effect of a smaller array. The channel enable function is configurable per SPI-Interface.

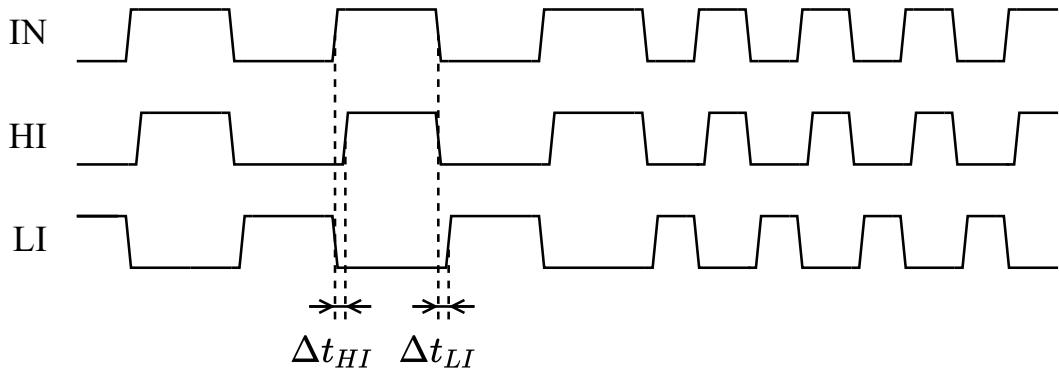


Figure 4.17: Dead Time Generator Signals

4.4 Software Design

The software on the Raspberry Pi was written in Python 3.9.2 due to its simplicity. However, most Python libraries make use of precompiled C-Code in the background. This enables fast real-time audio processing. The program was also built to run on Windows and Linux without the need of the hardware being available.

4.4.1 Structure

The software structure of the Audio-Beamformer program is shown in Figure 4.18. The *AudioBeamformer* module creates all instances of other modules and distributes references to submodules. Each module has a `begin` function for its initialisation and an `end` function for proper termination.

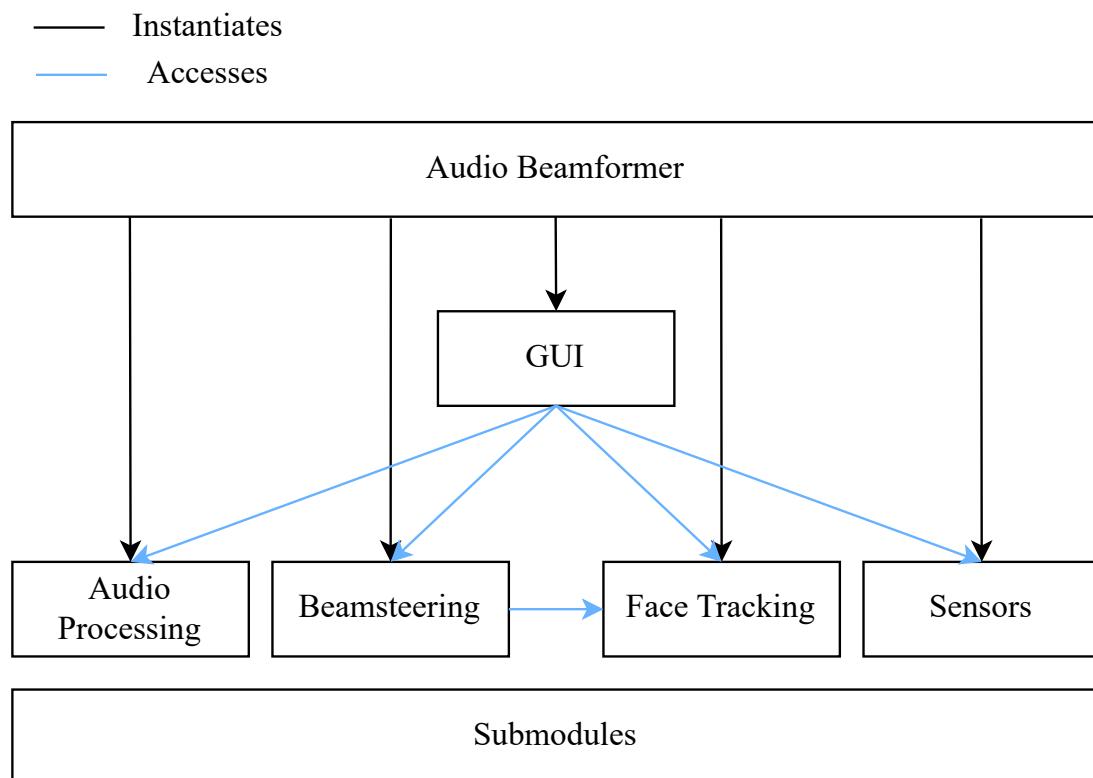


Figure 4.18: Software Structure

4.4.2 GUI

The GUI was made with PyQt, a wrapper for Qt in Python. The main goal was to create an intuitive, easy-to-use and informative graphical user interface. The three main sections of the UI can be reached through the buttons on the left. The configurable parameters were separated into three sections: Processing, Channels, and Settings. General settings such as mute, volume, and output level are always present on the right side of the screen.

Processing

The processing window is divided into five different sections. All of these sections contain settings for the audio processing.

1. Source

The audio input source and gain can be adjusted in this section. A gauge was added to get direct visual feedback of the input amplitude level.

2. Equalizer

In the second section, the equalizer can be enabled and chosen from the preset list. A bode plot can be seen at the bottom to give more information about the current equalizer used.

3. Interpolation

In this section, the interpolation can be enabled and the interpolation depth can be set to one of these values: 2, 4, 8, 16, 32 and 64.

4. Modulation Type

In the last section the modulation type can be chosen. Additionally, if the modulation type is set to MAM, the gain of the distortion channel can be adjusted.

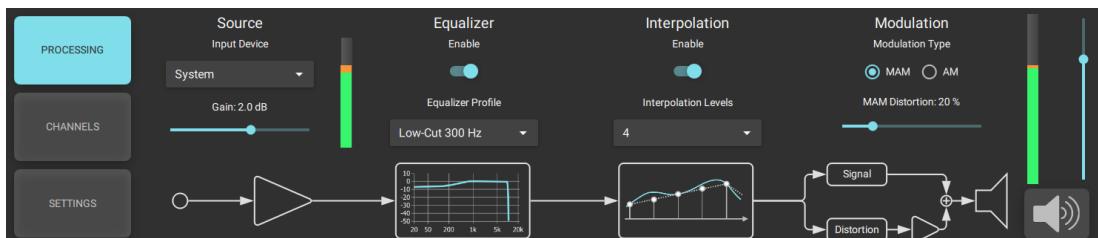


Figure 4.19: GUI Processing View

Channels

The *Channels* window is split up into 3 different areas.

1. Beam Steering

In this section, beam steering can be enabled, and the angle source can be set. If the angle source is set to *Camera* the face tracking is activated. If the source is set to *Manual* a slider appears with which the angle can be set manually. The last angle source is *Pattern* for which a predefined pattern of angles is played in a loop.

2. Window

In this section the window function can be enabled. If it is disabled the *Rectangle*

window is used. To give more information about the current window function, a plot of the different channel gains is shown at the bottom.

3. Video Stream

The video stream visualizes all detected faces and highlights the currently tracked person with a light blue rectangle. All other faces are surrounded by a grey rectangle.



Figure 4.20: GUI Channels View

Settings

The settings page is split up into 6 different areas.

1. LED

In this section the LEDs can be enabled and the global brightness can be set.

2. ToF Sensor

In this section the ToF sensor can be enabled and its sensitivity can be changed. To get a visual feedback of the sensitivity a gauge is present.

3. Max Volume

With this slider the maximum volume which can be reached by the loudspeaker can be adjusted. This can be done if the device is used indoors to guarantee safety.

4. Beam Focusing

In this section the beam focusing can be enabled and the focus distance can be adjusted.

5. Stats

This section shows temperature and performance information about the system and the CPU.

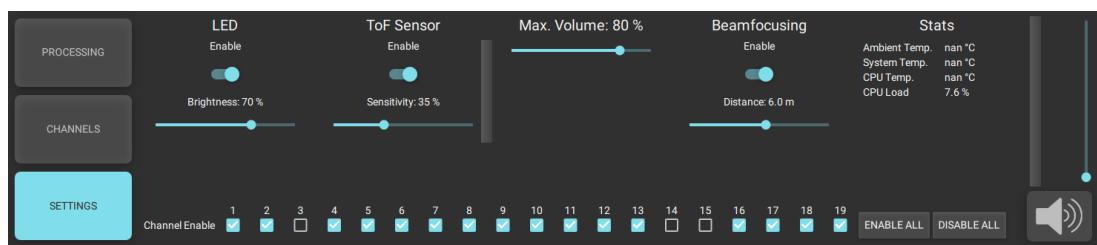


Figure 4.21: GUI Settings View

4.4.3 Audio Processing

In this module the audio stream gets processed. First the audio input is read block based into the program. Then the audio is filtered through an equalizer and changed for the modulation. In the end it is prepared for the transmission over the I²S interface to the FPGA.

Audio Stream

The audio stream is implemented in python using a library called *SoundDevice*, which wires the input to the output in a non blocking way. The audio is read block based with a block size of 8192 samples. The block size has chosen to be as small as possible to guarantee low latency if a microphone is used.

Equalizer

To compensate the distortion generated by the frequency response of the transducer, which is shown in Section 6.3.1, a FIR equalizer can be enabled between the input and the output of the audio stream. New equalizers can be easily be added in a text file.

Modulation Type

Currently the two modulation types implemented are AM and MAM. The left channel carries always the processed audio information. In the case of AM the right channel is just a copy of the left channel, but in case of MAM it is a second order approximation of the distortion term shown in Equation 3.15

$$\text{Left Channel} = 1 - \frac{1}{2}i(t)^2 - \frac{1}{8}i(t)^4. \quad (4.5)$$

Where $i(t)$ is the incoming audio signal.

4.4.4 Beam Steering

In the *BeamSteering* module, the delay and gain of each channels is calculated and transmitted via the SPI interface to the FPGA.

Channel Delays

The formula used for calculating the individual delays is

$$\tau_m = m \frac{d}{c_0} \sin \varphi, \quad (4.6)$$

where d is the distance between the transducers, φ the angle to steer to and c_0 the speed of sound. The speed of sound is also directly calculated in this module by using the ambient temperature, measured by the temperature sensor using the formula [9]

$$c_o = 331.5 + 0.607 \cdot T_{\text{Ambient}}. \quad (4.7)$$

The minimum angle which a phased array can reach is determined by the physical properties of the construction and by the smallest delay that can be applied to a signal. The physical properties of the transducer array are a spacing of $d = 14.75$ mm and the

number of channels $M = 19$. The smallest delay in this case is $\tau_{min} = \frac{2}{6.25\text{MHz}} = 320\text{ ns}$ and is determined by the output sampling rate

$$\varphi_{min} = \arcsin\left(\frac{\tau_{min}c_0}{Md}\right) \approx 0.21^\circ. \quad (4.8)$$

The maximal angle that can be reached is determined by the largest delay that can be applied to a signal. This is determined by the maximum number of memory cells, which in this case is $N_{MC} = 4092$. The largest delay that can be applied is $\tau_{max} = \tau_{min} \cdot N_{MC} \approx 654\mu\text{s}$. This leads to a maximal angle of

$$\varphi_{max} = \arcsin\left(\frac{\tau_{max}c_0}{Md}\right) \approx 53.4^\circ. \quad (4.9)$$

Channel Gains

For controlling the main lobe width and side lobe level, different windows can be applied to the loudspeaker. In addition to the gain of the channel, this also effects the brightness of the LEDs to give the user more information about the window.

4.4.5 Sensors

In this module, all sensors of the Audio-Beamformer are handled in a corresponding submodule. The data is processed and prepared for distribution. This is done in a way that other modules can easily access the information asynchronously.

Near-Field Avoidance System

The ToF sensor returns a 8×8 matrix of distance values. In order to detect a person a simple algorithm was implemented. At first, the distance map is convolved with a 2×3 mask. Further, two distance thresholds are used to identify an object. The hysteresis ensures a stable output even if noise is present. The threshold level can be adjusted by the sensitivity setting.

4.4.6 Face Tracking

To direct the audio beam towards a specific person, a face-detection algorithm is needed. It is key to differentiate between multiple recognized faces. An additional restriction is the limited processing power of the Raspberry Pi Compute Module 4, since it does not have a dedicated Graphical Processing Unit (GPU) or other hardware accelerators.

This part of the project has mostly been outsourced, due to overall lower priority compared to other parts of this thesis. In particular, Luca Jost has developed the face-detection and tracking algorithms. The detection is based on a neural network called MNN, more on that later in Section 4.4.6.

In order to keep track of each individual person as people are moving around, a tracking algorithm was implemented. To achieve this task, a Kalman filter was used. After a certain amount of time, a newly detected face is being tracked. This prevents rapid flickering of unwanted falsely detected faces (popping up on single frames). The algorithm keeps track of each face, even if the view gets interrupted for a short amount of time, e.g. the person tilts its head or moves behind an obstacle.

The *FaceTracking* module provides a chronologically sorted list containing the coordinates of each currently tracked face. The beam-steering module picks always the lowest

index of this list, meaning the person who is tracked the longest amount of time. As soon as this person disappears on the frame, the list is shifted to the left and the next person (now again oldest in list) is picked as a new target.

Mobile Neural Network

To detect multiple faces on an image, deep learning learning techniques are applied. The Mobile Neural Network (MNN), developed and trained by the Alibaba Group, offers advanced trained models and a highly efficient processing core, which is specially optimized for AI applications on mobile devices with limited processing power. With the help of the python library called *PyTorch*, machine learning algorithms can easily be applied on a high level abstraction layer and used with scripting languages like Python.

4.4.7 Operating System

The Operating System (OS) on the Raspberry Pi Compute Module 4 is called *Raspberry Pi OS* and is based on Linux. It is key to use the 64 Bit version, since frameworks such as MNN and PyTorch cannot be run on a 32 Bit OS. In this project, the graphical OS version has been used, as it is more convenient to work with. The disadvantage is however, that the boot time is significantly longer (approx. 20 s).

Wireless Audio Streaming

As input sources, Bluetooth® audio streaming and AirPlay® (product of Apple Inc.) are supported. This offers the very convenient possibility to stream music directly from any mobile device (such as smartphones or laptops) to the Audio-Beamformer. It is important to note that for connecting via AirPlay®, both devices need to be in the same wireless network.

The wireless signal strength is attenuated by the metal enclosure. This leads to a smaller range of a reliable audio streaming transmission. Tests have shown that a distance of more than 3 m can become problematic. This issue could be overcome by connecting an external antenna to the Raspberry Pi Compute Module 4.

4.5 Mechanical Design

The mechanical design has proven to be a substantial part of the overall development process. With the help of modern CAD tools like *SolidWorks 2022*, the design of the mechanical parts could vastly be accelerated.

A further advantage of creating an exact 1:1 model of the complete product is the possibility to render photo-realistic images or videos. They can be used as attractive illustrations or for advertising purposes. As a 3D rendering tool, *SolidWorks Visulize 2022* has been used.

4.5.1 Concept

The mechanical concept of the Audio-Beamformer connects several considerations. First of all, acoustic constraints had to be discussed (e.g. avoiding vibrations of loosely connected parts). Next, the electrical part must comply with regularisation and common standards (e.g. safe to operate at mains voltage). Furthermore, all mechanical parts should be easy to manufacture and assemble. This helps reducing the cost, makes the design more accessible and improves the overall service capability. Finally, the complete assembly should have a professional and modern appearance. Combining all those factors resulted in a fully customized and sturdy construction.



Figure 4.22: 3D-Render of Final Product

4.5.2 Enclosure

The case of the Audio-Beamformer is a combination of an aluminium sheet metal construction with a transparent acrylic front panel. The PCB is held in place by M3 spacers from both the front and rear side ("sandwich" construction). The top and bottom plate is made out of 8 mm thick aluminium bars. This is necessary, since they create a solid connection point to all other mechanical parts. The back and front panel have a thickness of 3 mm and the side panels (left and right) of 1.5 mm.

In the centre of the base plate, a 3/8" thread (DIN 4503-1 / ISO 1222) was cut. This provides the possibility of attaching the Audio-Beamformer onto any standard camera tripod. Care was taken that the fixture is capable of carrying the total weight of the Audio-Beamformer.

All aluminium sheet metal parts, as well as the acrylic front panel, were laser-cut. All other components were manufactured by conventional methods in the workshop at the eastern university of applied science.

The final dimensions of the Audio-Beamformer are: 304 x 393 x 46 mm. The total weight is: 3.88 kg.

5

Risks & Safety

5.1 Risks

The maximum sound pressure level that is allowed by SUVA is 140 dB and the averaging level 8h/day has to be below 110 dB. The averaging level L_m is given as

$$L_m = 10 \log_{10} \left(\frac{1}{8} \int_0^T 10^{0.1 L_p(t)} dt \right). \quad (5.1)$$

This can be solved for the sound pressure level $L_p(t) = L_p$ if it is assumed to be constant over a certain time τ

$$L_p = L_m - 10 \log_{10} \left(\frac{\tau}{8} \right). \quad (5.2)$$

To calculate the sound pressure level at any given point, the directivity of the used transducers has to be calculated. As explained in Section 3.2, the sound directivity can be calculated as

$$Q_D = \frac{2p(0)^2}{\int_0^\pi p^2(\theta) \sin \theta d\theta}. \quad (5.3)$$

The sound pressure ratio emitted by the transducers according to its datasheet is displayed in Figure 5.1.

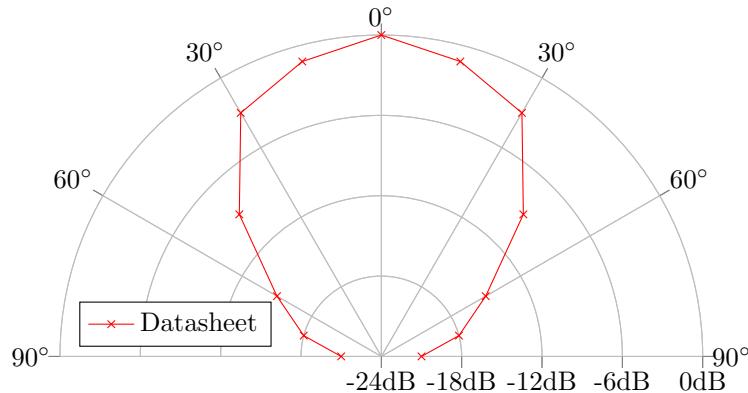


Figure 5.1: Directivity of a Transducer

With this information, the directivity index results to be $Q_D = 22$. This then can be used to calculate the maximum allowed sound power level in relation to the distance of the listener

$$L_{wmax} = \underbrace{L_{pmax}}_{140dB} - 10 \left(\underbrace{\log_{10}(Q_D)}_{1.35B} - \log_{10}(r^2) - \underbrace{\log_{10}(4\pi)}_{\approx 1.1B} \right) = 137.5 + 20 \log_{10}(r). \quad (5.4)$$

The maximum sound power allowed in relation to the distance is plotted in Figure 5.2. In Figure 5.3 the maximum sound power allowed over different periods of time is shown.

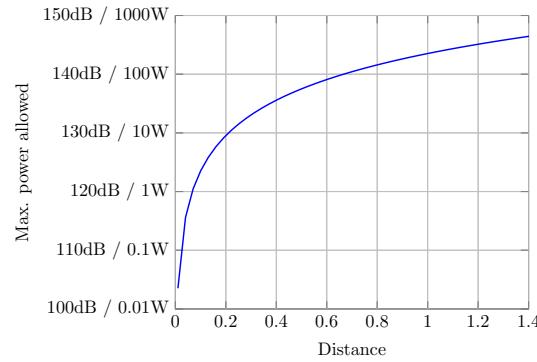


Figure 5.2: Maximum allowed Sound Power

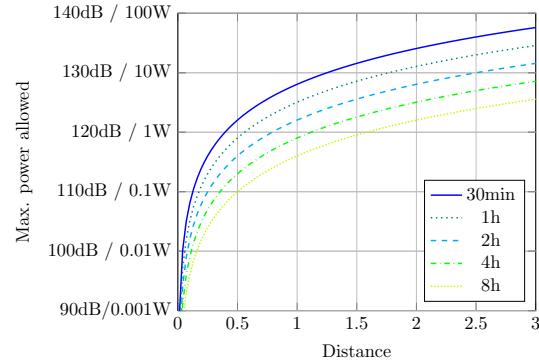


Figure 5.3: Maximum Power allowed (daily)

This was calculated by using Equation 5.2 and Equation 5.4 where $L_m = 110 \text{ dB}$ as stipulated by SUVA.

5.2 Safety

Through measuring the voltage over a resistor R right in front of each line of transducer arrays, the total current going into the transducers could be measured. Additionally the voltage over the transducers was measured. From these measurements the total possible sound power which the whole transducer array could produce can be calculated

$$L_{P,max} = M \cdot \frac{U_R}{R} U_T \eta_T. \quad (5.5)$$

Where M is the number of channels, U_R is the voltage over the resistor, U_T is the voltage over the transducer and η_T is the efficiency of the transducer. To guarantee maximal safety, the efficiency η_T is assumed to be one, which is highly overestimated. In this particular case the number of channels is $M = 19$. For the transducers the voltages measured were $U_R = 1.78V$ and $U_T = 9.5V$ and the resistor is $R = 22\Omega$. This can be used to calculate the maximum sound power

$$L_{P,max} = 19 \cdot \frac{1.78V}{22\Omega} 9.5V = 14.36 \text{ W} \quad (5.6)$$

So, to guarantee that a person could listen to the Audio-Beamformer on full volume for half an hour daily without any harm, the minimum distance was calculated using Equation 5.2 and Equation 5.4 to be 2.5 meters. The minimum distance was set to 2.5 meters. This distance is measured by a Time-of-Flight (ToF) sensor, the implementation of this is explained in Section 4.4.5.

6

Measurements

6.1 Human Expertise Test

In order to better grasp the audio quality, directivity and beam steering of the Audio-Beamformer, a human expertise test was conducted. The device was shown to 17 people in different test settings.

6.1.1 Test Setup

To fully test the capabilities of the device a free standing location was chosen so that the reverberation could be neglected. In Figure 6.1 the five different points (A, B, C, D, E) where the measurements took place are shown. These points are all in a distance of ten meters to the loudspeaker. The points A and E are at an angle of 30 degrees and B and D are at 15 degrees. The point C is directly in front of the device.

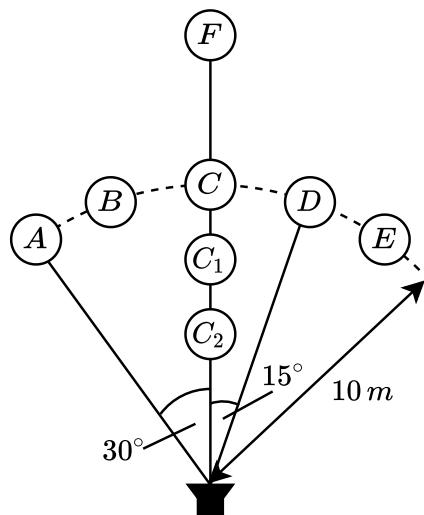


Figure 6.1: Measurement Setup Final Tests

6.1.2 Audio Quality

To evaluate the audio quality, a rating scale has been developed, based on the Swiss grading system. This is shown in Table 6.1.

1	2	3	4	5	6
Completely unrecognizable audio, very distorted and noisy	Hardly anything can be recognized, noise and distortion are dominant	Mostly recognizable audio, distortion and noise clearly audible	Acceptable hearing experience, speech recognizable without effort	Enjoyable hearing experience, appropriate for daily use	Outstanding Hi-Fi audio quality, no noise audible

Table 6.1: Audio Quality Grading Scale

Measurements

1. General Audio Quality

To test the general audio quality, music and speech was played. All settings were set to default.

Test	Average	Variance
Music quality	4.2	0.46
Speech quality	4.5	0.38

Table 6.2: Audio Quality Score

2. Equalizer

For the second test, the equalizer was changed and the music quality of each preset was tested. The equalizer *Main* was not specifically tested but it is the default equalizer and is mentioned in Table 6.3 as a comparison.

Test	Average	Variance
No equalizer	4.1	0.67
Equalizer: <i>Lowcut 300Hz</i>	4.25	0.60
Equalizer: <i>Sharp</i>	4	0.66
Equalizer: <i>Main</i>	4.2	0.46

Table 6.3: Audio Quality of different Equalizers

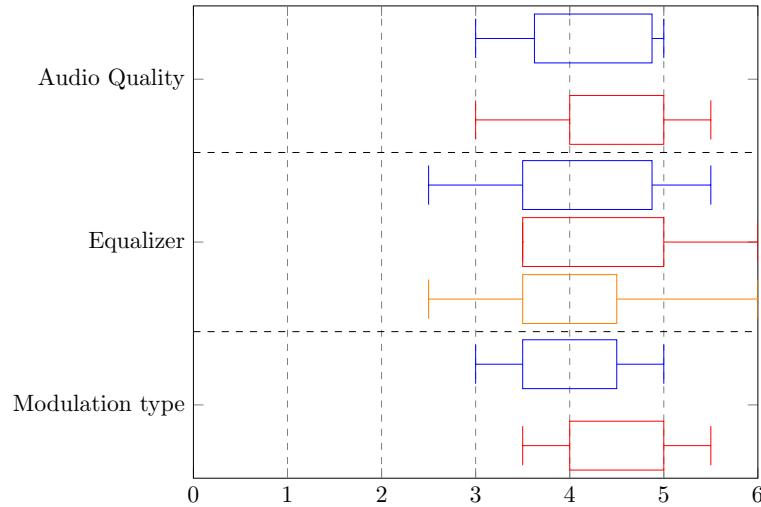
3. Modulation Type

At last, the audio quality of the two modulations types AM and MAM were compared. The other settings were again set to default.

Test	Average	Variance
AM	3.9	0.40
MAM	4.4	0.34

Table 6.4: Audio Quality of different Modulation Types

A more detailed overview of the quality measurements can be seen in Figure 6.2.

**Figure 6.2:** Audio Quality Box-Plot

Evaluation

The general audio quality test turned out well, as both music and speech were rated on average better than an acceptable hearing experience. Especially the audio quality of speech surprised us and showed that this could be an important use case. In the equalizer test all results turned out to be about the same and statistically show no significant difference between them. We think with more time and more testing better and more impressive equalizer settings can be found and these results can be improved on. The modulation type test painted a strong picture that MAM modulation is the way to go. Overall, we are very pleased with these results and think a real alternative to conventional loudspeakers can be created with further improvements.

6.1.3 Audio Volume

To evaluate the beam steering capability and overall directivity, a grading system was developed. This is shown in Table 6.5.

-4	-3	-2	-1	0
Nearly nothing audible, not noticeable volume level	Noticeable if background is quiet (no one is talking)	Strongly noticeable volume level, even with background noise (speech)	Clearly audible volume level	Very present volume level, strongly dominates background noise

Table 6.5: Audio Volume Grading Scale

Measurements

Due to the absolute value of the volume being very objective, the results of each individual test was normalized to be between 1, very loud, and zero, nothing audible. This means that the following list of test results says nothing about the absolute value of the Audio-Beamformer.

1. Directivity

To test the directivity, the test person had to rate the volume at every point (A, B, C, D and E). Once with no window applied and once with the Dolph-Chebyshev window, which should, in theory, generate a more directive beam.

Test	A	B	C	D	E
No window	0.34	0.57	1	0.66	0.43
Dolph-Chebyshev	0.32	0.5	1	0.54	0.34

Table 6.6: Audio Directivity

2. Beam Steering

To test the beam steering, two different tests were carried out.

Point C

In the first test, the test person stood on Point C and the beam was steered to an angle of 0, 15 or 30 degrees. This was tested with no window and the Dolph-Chebyshev window.

Test	0°	15°	30°
No window	0.98	0.8	0.83
Dolph-Chebyshev	1	0.72	0.70

Table 6.7: Beam Steering Point C

Point A

For the second test, the test person stood on Point A and the beam was steered to an angle of 0, 15 or 30 degrees. This was tested with no window or the Dolph-Chebyshev window.

Test	0°	15°	30°
No window	0.55	0.71	0.96
Dolph-Chebyshev	0.38	0.51	1

Table 6.8: Beam Steering Point A

Evaluation

The directivity test showed that already at around $\pm 30^\circ$ the sound is almost only noticeable if the background is quiet. One can also see that the Dolph-Chebyshev window is, as expected, more directive. From the beam steering tests one saw that, especially with no window it is very difficult to keep the point directly in front of the speaker quiet. This is in our opinion mainly due to the inherent directivity of the loudspeaker. The second test, at point A, shows that the beam steering works as expected.

6.2 Ultrasound Measurements

As an addition to the human expertise test, a measurement series was carried out. Because we had no sound chamber at our disposal we had to carry out these measurements outside. But due to the surrounding noise, a measurement in the audible spectrum was impossible so all the measurements were carried out only with the carrier. As a result of this, and the difficulties of sound measurement without an ideal setup, measurement device and location these measurements have to be viewed with a pinch of salt.

6.2.1 Directivity

The directivity of the ultrasound was measured with and without a window at 9 distinct points between -30° and 30° . The results of these measurements are shown in Figure 6.3.

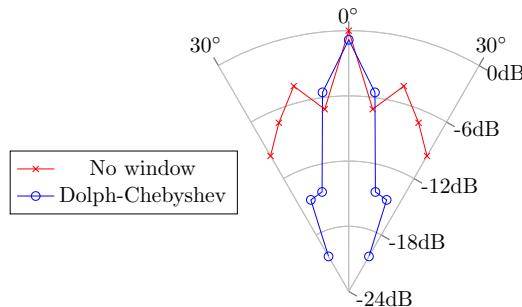


Figure 6.3: Directivity Measurements

6.2.2 Beam Steering

To measure the effects of the beam steering, the beam was once directed to 15° , as seen in Figure 6.4, and once directed to 30° , as seen in Figure 6.5. The dashed lines in both figures show the mirrored points to give a better feeling of how the beam looks. Due to the inherent directional nature of the loudspeaker, these dashed lines would in practice fall off a lot quicker than shown here.

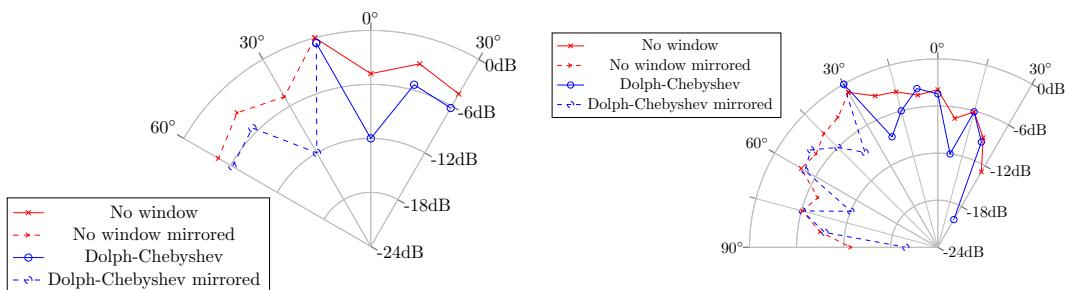


Figure 6.4: Beam steered to 15°

Figure 6.5: Beam steered to 30°

6.2.3 Beam Focusing

The beam focusing was tested on the Points C1 (7.5m), C2 (5m) and F (15m). The results of this are shown in Figure 6.6.

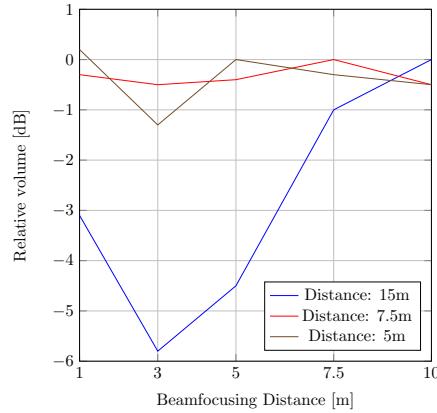


Figure 6.6: Beam Focusing Measurements

6.2.4 Evaluation

As already said, due to the imperfect measurement setup these measurements have to be looked at critically. That said, the directivity and beam steering reflect the human expertise tests really well. In comparison to the calculations made in 3.5 the measurements look similar, the main difference is the size of the side lobes, which in our measurements are a lot smaller than calculated. This is most likely due to the directivity of the transducers not being the their assumed sinc function, as shown in Figure 5.1. The beam focusing, on the other hand seems to only work if the distance to the loudspeaker is big enough. To verify further tests have to be made under more ideal conditions.

6.3 Transducer Measurements

To get a better feeling for the transducers and to see how they behave under different circumstances, several measurements where taken.

6.3.1 Frequency Response

To be able to design more accurate equalizers, the sound pressure level output at each frequency between 22.5 kHz and 55 kHz was measured and is shown in Figure 6.7. But again due to no access to a sound chamber these measurements have to be looked at critically.

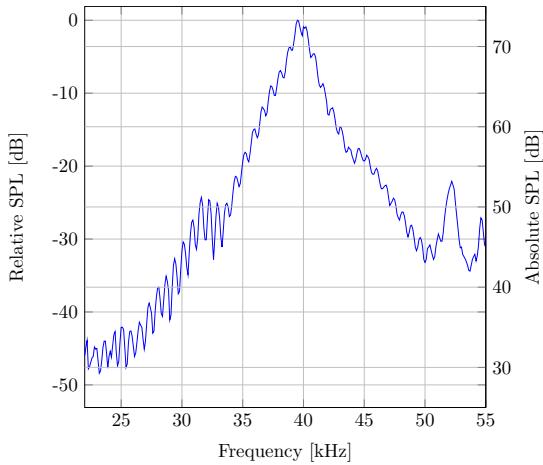


Figure 6.7: Frequency Response Transducer

6.3.2 Power Consumption

The power consumption of each of the 19 transducer lines is very important to access the output power of the loudspeaker. Over a 22Ω resistor in front of the transducer the current could be calculated. The voltage over the transducer and the voltage over the resistor was measured for 4 hours and it turned out that both these values were very constant. The voltage measured over the transducer was $U_T = 9.5\text{ V RMS}$ and the voltage over the resistor $U_R = 1.78\text{ V RMS}$.

6.3.3 Impedance Measurements

To get information about the electrical side of the transducers, an impedance measurement was carried out with a series of 20 transducers. It was measured using a vector network analyser. These measurements can be found in the Audio-Beamformer-software repository (Appendix A.2).

7

Conclusion

Even though not all the tasks of the assignment were possible to accomplish with the method chosen, we think that the project was a huge success. The main goals of a professionally built, easy to use, directive and steerable loudspeaker were reached. The ultrasound measurements, in Section 6.2, and the human expertise tests, in Section 6.1, objectively showed this. Additionally, we think we brought new ideas into how parametric loudspeaker arrays are built by utilising signal processing methods such as the sigma delta modulation.

7.1 Continuing Work

Although the final product provide a rich set of features, there are some aspects to improve or features to add. Of particular note is the fact that the system is designed for future enhancements. As an example, the flash storage of the Raspberry Pi Compute Module 4 provides lots of additional storage for larger software editions. The following continuations are possible:

- Beam-Focusing improvement
- Higher order sigma-delta-modulator
- Improve wireless performance with external antenna
- Develop Remote or app to control the Audio-Beamformer
- Frequency Response Measurement and compensation

7.2 Personal Reflections

Florian Baumgartner

This bachelor's thesis proved to be very challenging, since it covered pretty much every field of electrical engineering. This, however, made it very attractive to work on the project, due to the enormous amount of variety in different topics. I personally could make use of my previously gained knowledge to accelerate the development process. It was a fantastic experience to design a fully working and professional looking product in such a small time frame. I'm very happy with the end result and hope that it will satisfy its purpose of convincing potential new students to start studying electrical engineering. It was a pleasure to work with Thierry Schwaller and we had overall a great time working on this project.

Thierry Schwaller

I am personally really proud of the things we've reached throughout this bachelor's thesis. I am convinced that the Audio-Beamformer could become a commercial product if more research is done. Additionally, I learned a lot during this assignment about the theory behind the inner workings of phased arrays and the difficulties in connecting theory to the real world. The collaboration with Florian Baumgartner helped me a lot to see and try new angles to solve a problem.

A

Appendix

A.1 Declaration of Authorship

We hereby certify that the thesis we are submitting is entirely our own original work except where otherwise indicated. We are aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Location, Date

Rapperswil, 03. June 2022



Florian Baumgartner



Thierry Schwaller

A.2 Data Archive

All created files and documents of this project are publicly available on GitHub. An institution called **BA-OST-2022** (<https://github.com/BA-OST-2022>) has been founded which contains repositories for each individual part of the project. A quick description of the repositories including the associated web link is listed below:

audio-beamformer-admin

Description: This repository contains all confidential information of the project.

URL: <https://github.com/BA-OST-2022/audio-beamformer-admin>

Type: Private

audio-beamformer-docs

Description: This repository contains all additional documentation of the project.

URL: <https://github.com/BA-OST-2022/audio-beamformer-docs>

Type: Public

audio-beamformer-thesis

Description: This repository contains this document.

URL: <https://github.com/BA-OST-2022/audio-beamformer-thesis>

Type: Public

audio-beamformer-hardware

Description: This repository contains hardware related documents (Schematics, PCB).

URL: <https://github.com/BA-OST-2022/audio-beamformer-hardware>

Type: Public

audio-beamformer-firmware

Description: This repository contains firmware source code written in C++.

URL: <https://github.com/BA-OST-2022/audio-beamformer-firmware>

Type: Public

audio-beamformer-software

Description: This repository contains the device software written in Python.

URL: <https://github.com/BA-OST-2022/audio-beamformer-software>

Type: Public

audio-beamformer-gateware

Description: This repository contains the FPGA description code.

URL: <https://github.com/BA-OST-2022/audio-beamformer-gateware>

Type: Public

audio-beamformer-mechanical

Description: This repository contains mechanical related documents (CAD-Files).

URL: <https://github.com/BA-OST-2022/audio-beamformer-mechanical>

Type: Public

A.3 Definition of Task



AUFGABENSTELLUNG

Thema: Entwicklung eines steuerbaren Lautsprecher-Arrays

Studierende: Florian Baumgartner, Thierry Schwaller

Betreuer: Hannes Badertscher

Partner: ICAI

Fachgebiet: Digital Signal Processing

Kurzbeschreibung

In vielen Audio Anwendungen ist es gewünscht, das Audio-Signal der Umgebung anzupassen. So messen z. B. Sound Bars den Raum akustisch aus und nutzen Wandreflexionen um einen virtuellen Surround Sound zu erzeugen. In vielen Anwendungen ist es gewünscht, die Person für welche der Sound erzeugt wird lokalisieren zu können, und den Sound so auszugeben, dass er für diese Person optimal hörbar ist, ohne andere Personen zu stören.

Das Ziel dieser Arbeit ist es, einen Demonstratoren zu entwickeln und zu bauen, welcher aus einem 1D Lautsprecher-Array und einer Stereokamera besteht. Mit der Stereokamera (z. B. Intel RealSense) soll eine Person (oder zu Test-Zwecken ein verschiebbbares Objekt, welches eine Person symbolisiert) im 3D-Raum lokalisiert und deren Bewegungen getracked werden. Das Lautsprecher-Array soll so angesteuert werden, dass der Sound für die lokalisierte Person möglichst gut hörbar, und an allen anderen Orten im Raum möglichst leise ist. Mittels des Mikrofon-Arrays soll die Performance gemessen und visuell dargestellt werden.

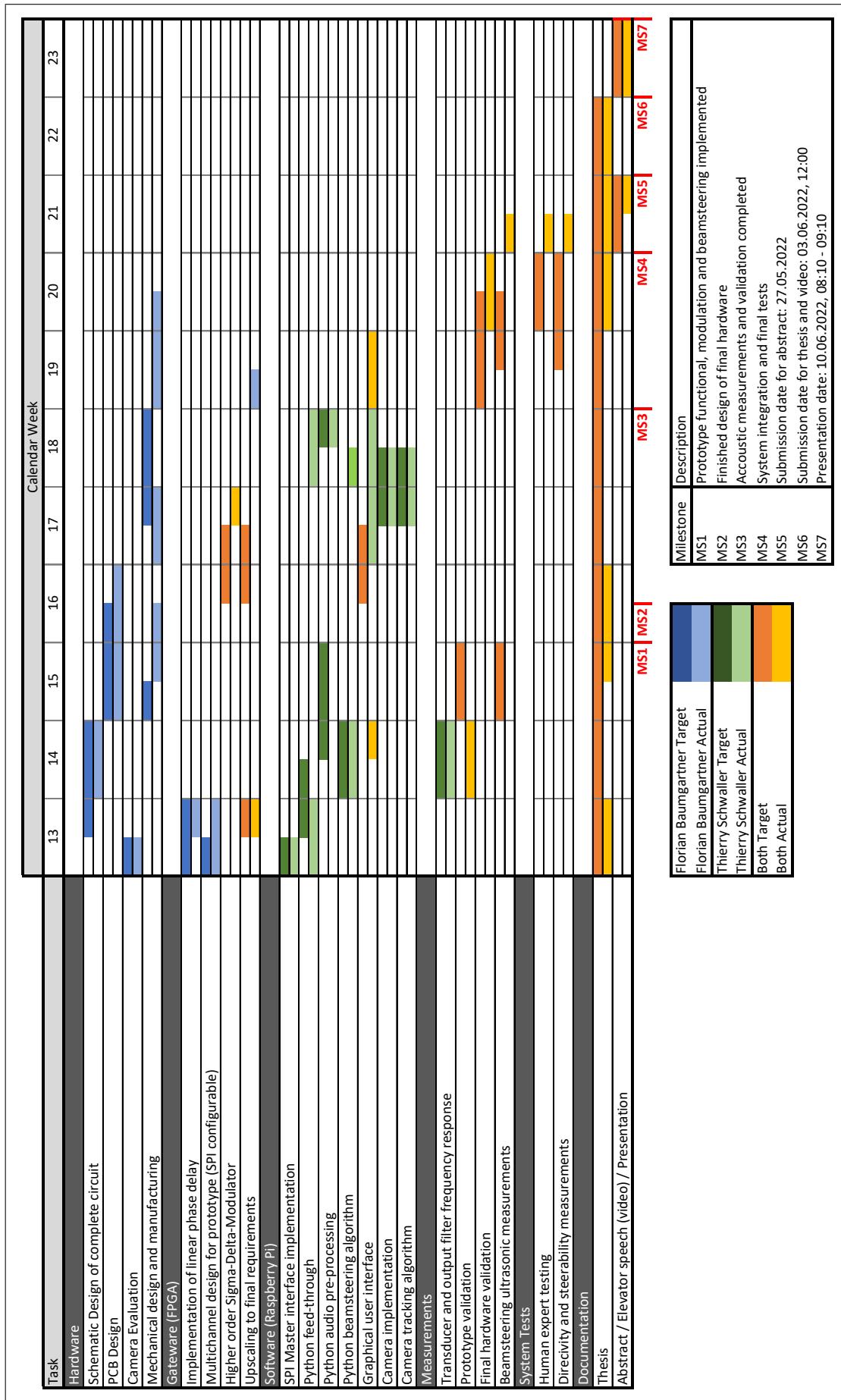
Aufgabenstellung

- Literaturrecherche zu Phased Arrays und Beam Steering insbesondere im Audiobereich.
- Entwicklung der Elektronik und Mechanik eines Lautsprecher-Arrays, sowie einer fix montierten Stereokamera zur Objekterkennung.
- Kalibration und Charakterisierung der Lautstärke und Richt-Charakteristik des Lautsprecher-Arrays
- Entwicklung der Elektronik und Mechanik eines Mikrofon-Arrays.
- Kalibration und Charakterisierung der Empfindlichkeit und Richt-Charakteristik des Mikrofon-Arrays.



- Entwicklung eines Algorithmus zur Lokalisierung einer Person bzw. einem Objekt das eine Person symbolisiert mithilfe des Stereokamera-Bilds, sowie Tracking der Bewegung der Person und graphische Darstellung des Trackings.
- Entwicklung eines Algorithmus zum Beam Steering, wobei der Sound automatisch auf die erkannte Person bzw. das erkannte Objekt gelenkt wird.
- Messung der Performance des Lautsprecher-Arrays mit dem entwickelten Algorithmus, insbesondere der Sidelobes und der Performance über verschiedene Frequenzen.
- Test des entwickelten Systems in verschiedenen Umgebungen, insbesondere in Räumen mit starken Echos und/oder Hall.
- Optimierung des entwickelten Systems und der Algorithmen zur Verbesserung der Performance in schwierigeren Umgebungen.
- Entwicklung eines einfach zu installierenden und zu bedienenden Demonstrators, welcher Sound automatisch auf eine Person oder ein verschiebbbares Objekt lenkt.

A.4 Project Schedule



A.5 Datasheet Ultrasonic Transducer

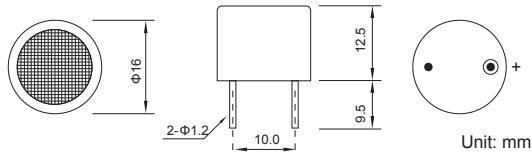
MA40A16

40kHz ultrasonic transducer

Appearance



Dimensions



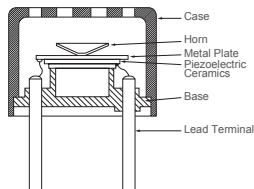
Specification

● MA40A16T:	Transmitter
● MA40A16R:	Receiver
● MA40A16B:	Dual Type(Transmitter+Receiver)
● Operating Frequency:	40kHz ± 2%
● Transmitting Sensitivity:	117dB Min.
● Echo Sensitivity:	-66dB
(0dB = 20uBar·10Vpp, 10Bursts,30cm)	
● Capacitance at 1 kHz:	2400 pF ± 20%
● Typical Sensing Range:	0.3 ~ 10m
● Directivity:	60° ± 10°
● Max. Driving Voltage:	80 Vpp
(2% Duty Cycle Tone Burst)	
● Operating Temperature:	-30°C to 80°C
● Storage Temperature:	-40°C to 90°C
● Housing Material:	Plastic
● Color:	Black

*All above parameters were measured at 25°C .

*Actual properties are depending on drive circuit.

Construction Structure



Applications

- Specially use in transmitting function.
- Use for generate high ultrasonic sound pressure.
- Use for dog, mouse or insect repellent device.
- Use for long distance detection.

Applications

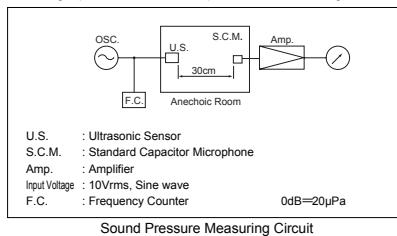
- 336 pcs/inner box, 12 boxes/carton box.

Sound Pressure Characteristics

Sound pressure level (S.P.L.) is unit indicating the volume of sound and is expressed by the following formula.

$$S.P.L. = 20 \log \frac{P}{P_0} (\text{dB})$$

where "P" is Sensor sound pressure (Pa) and "P₀" is reference sound pressure (20μPa). Follow graph shows a sound pressure measuring circuit.



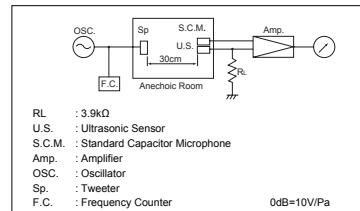
Sound Pressure Measuring Circuit

Sensitivity Characteristics

Sensitivity is the unit indicating the sound receiving level and is expressed by the following formula.

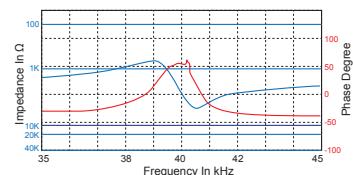
$$\text{Sensitivity} = 20 \log \frac{S}{S_0} (\text{dB})$$

where "S" is Sensor voltage (V) and "S₀" is reference sound pressure (V/Pa). The follow graph shows a sensitivity measuring circuit. The 3.9kΩ resistor connected with the electrode terminal of the sensor is used to avoid the influence of outside noise.

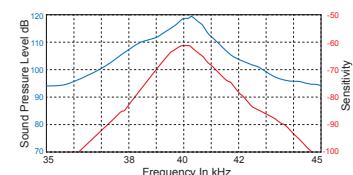


Sensitivity Measuring Circuit

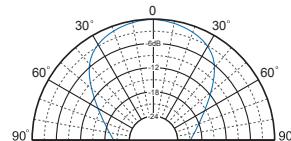
Impedance & Phase



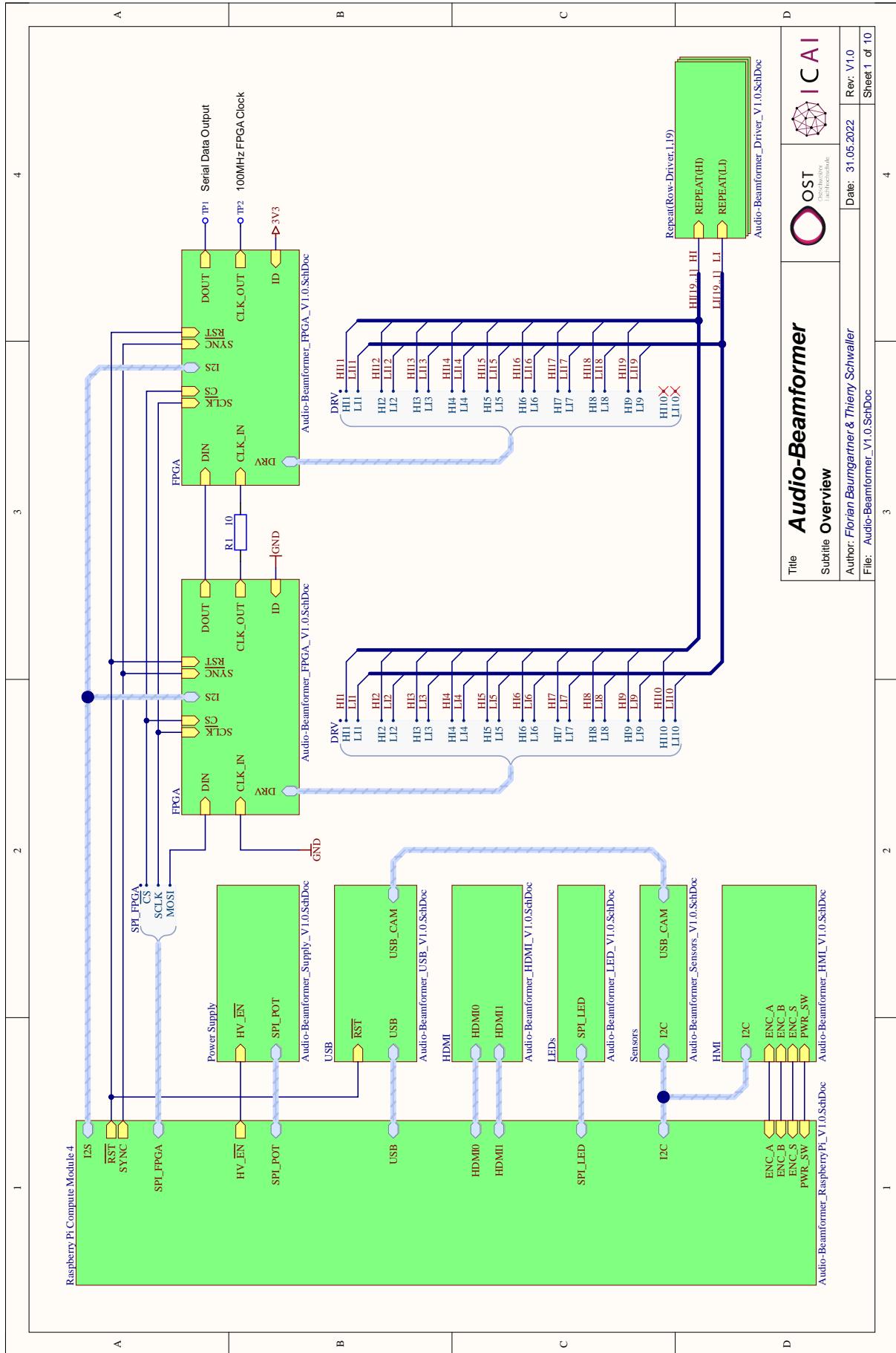
Sound Press & Sensitivity

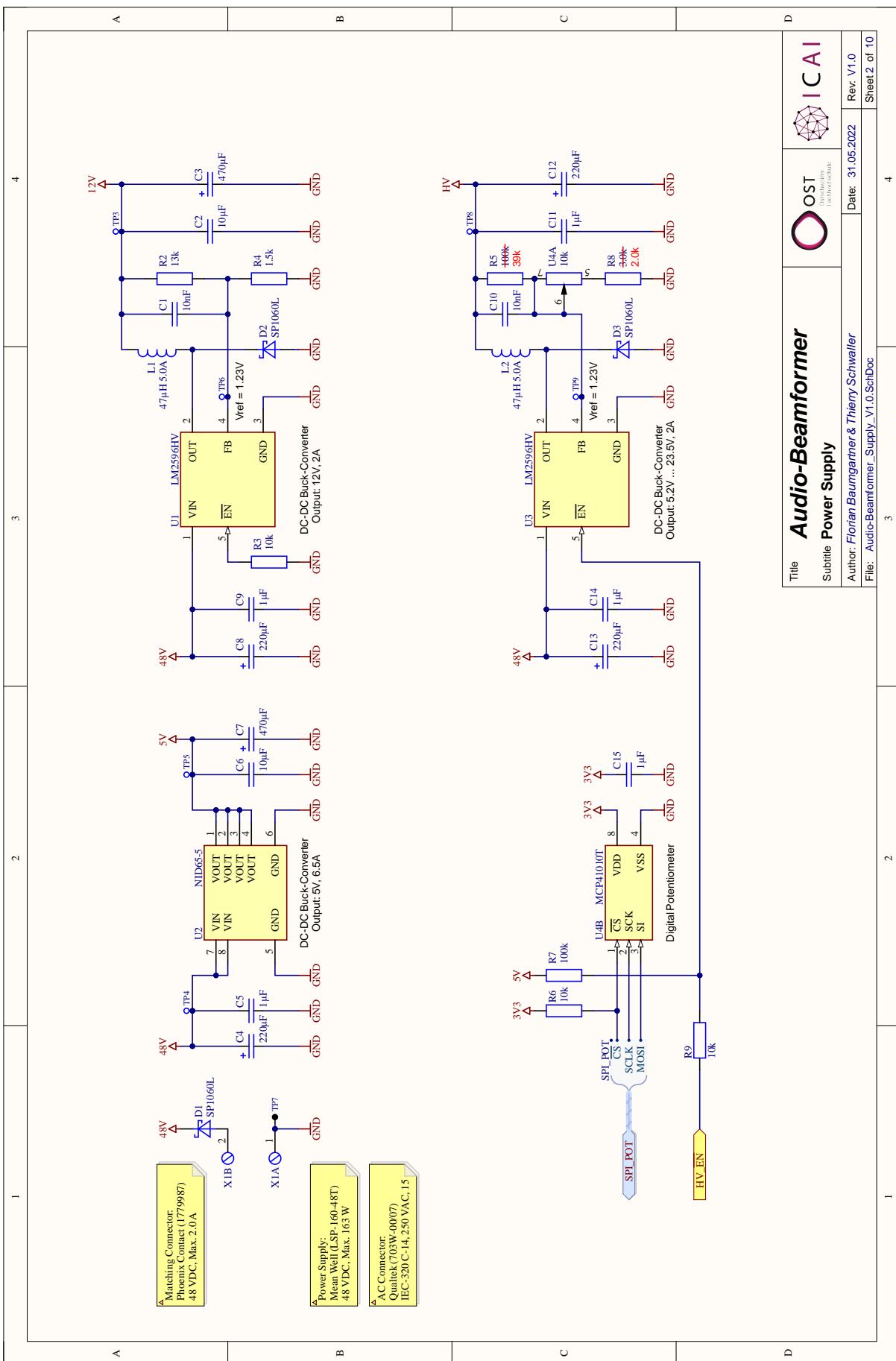


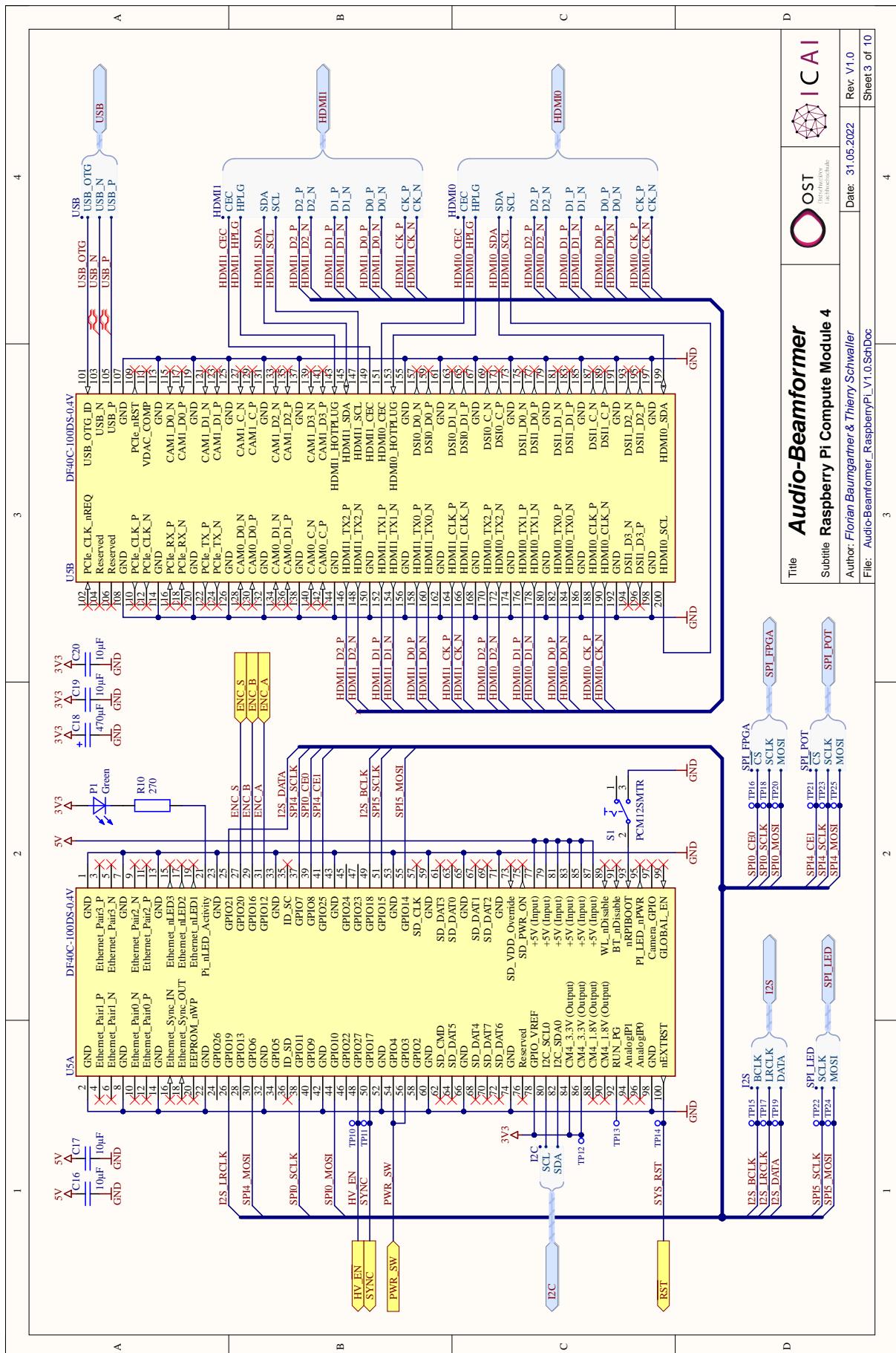
Directivity

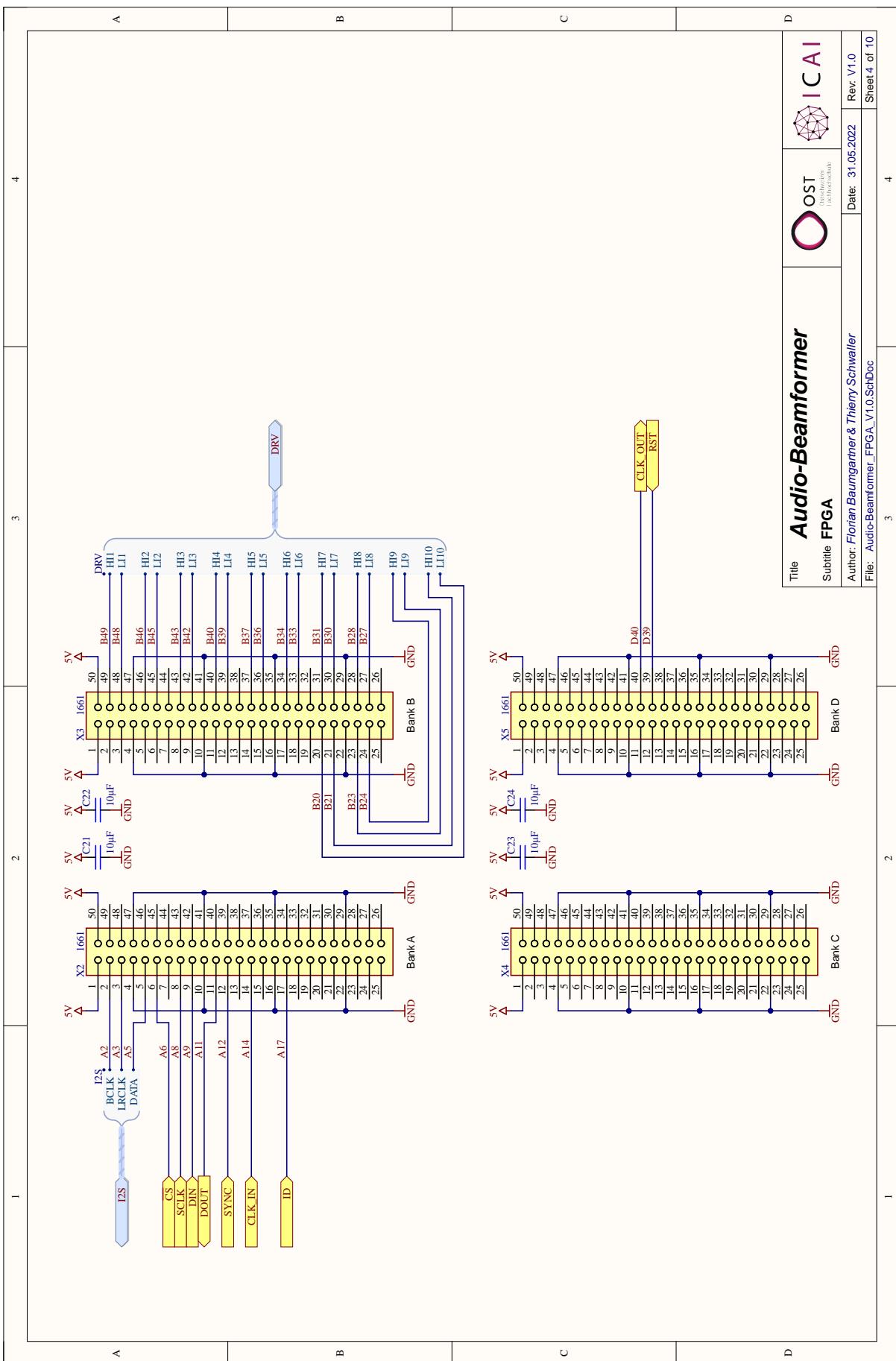


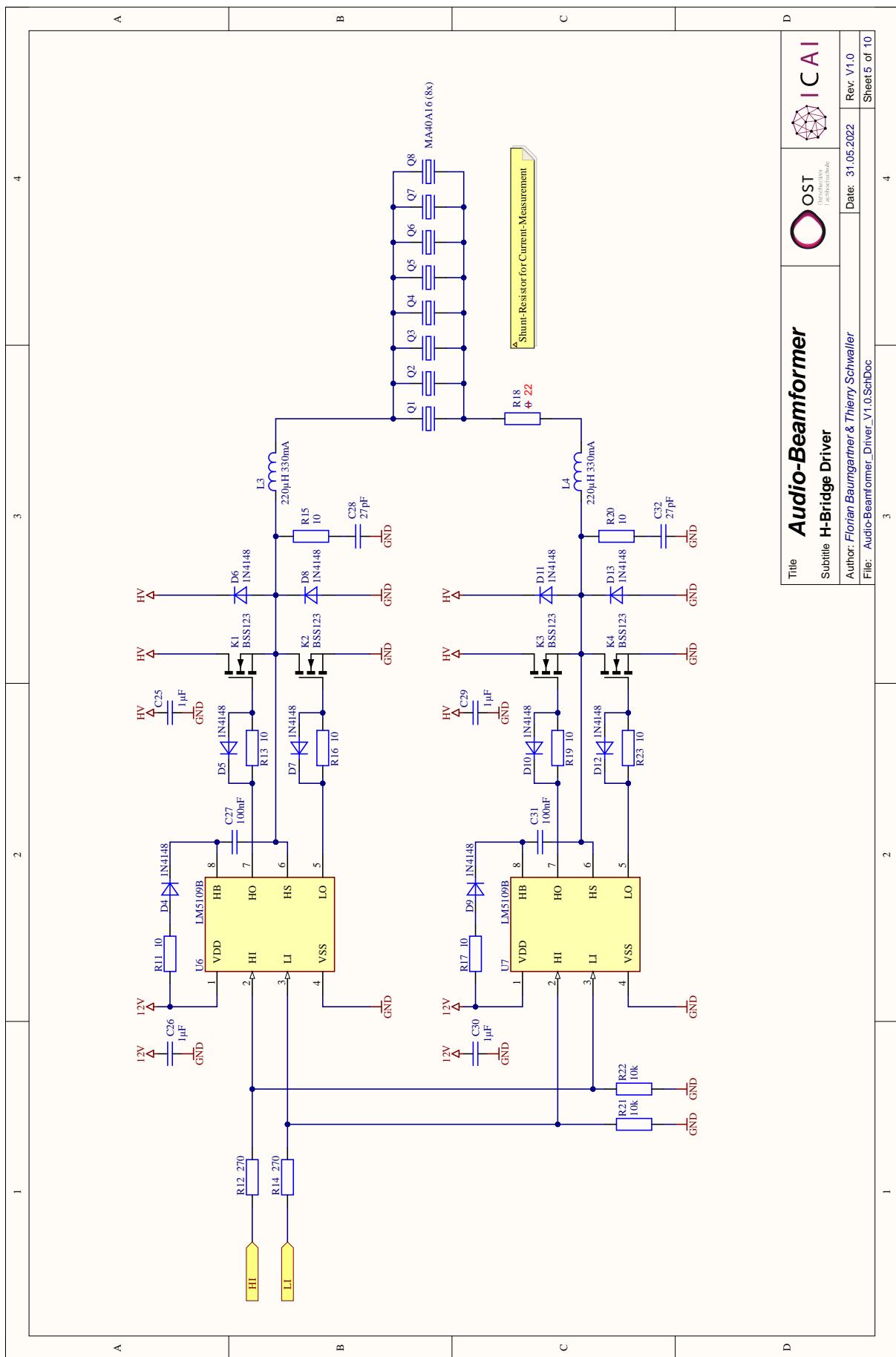
A.6 Audio-Beamformer Schematics

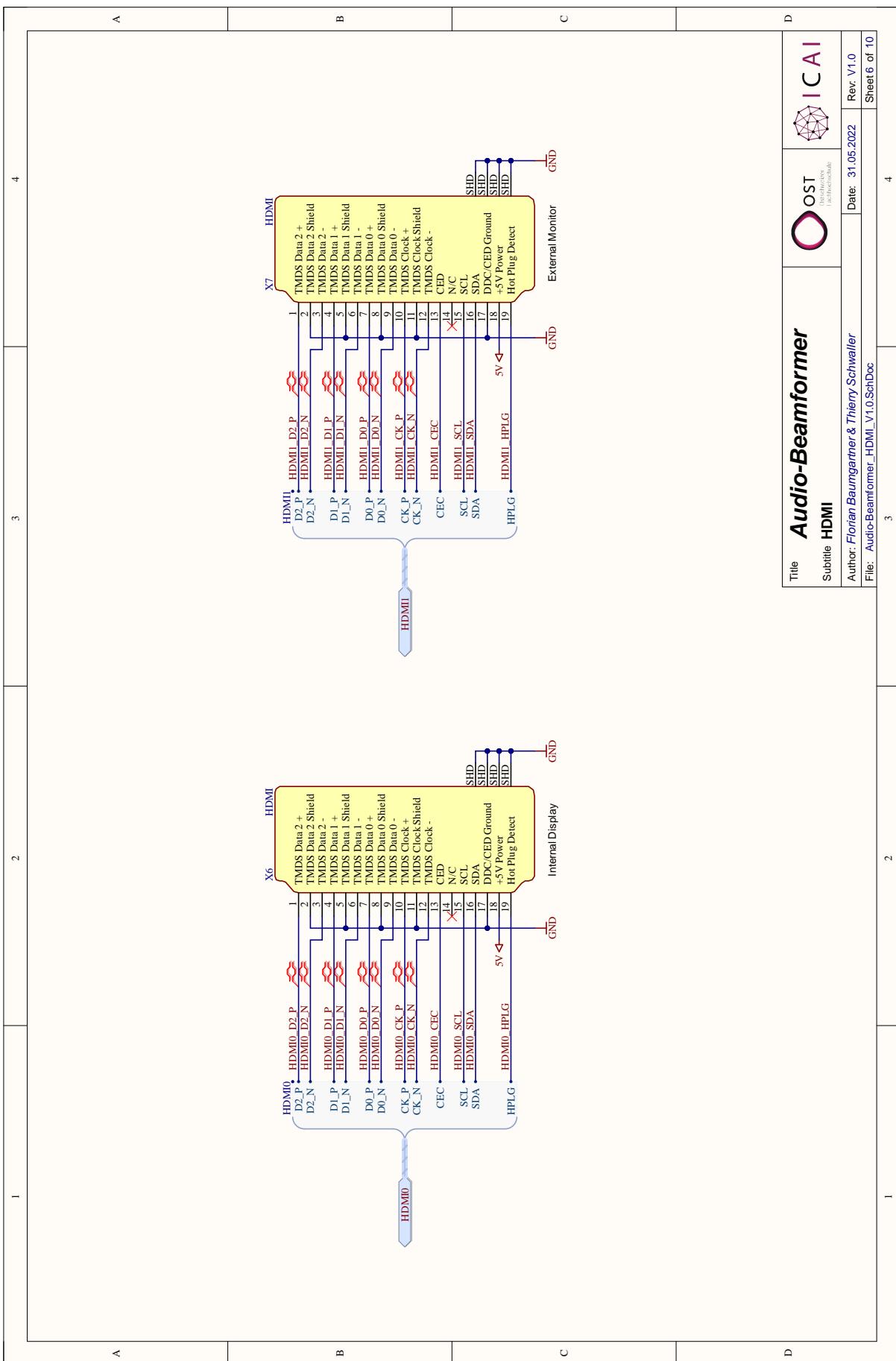


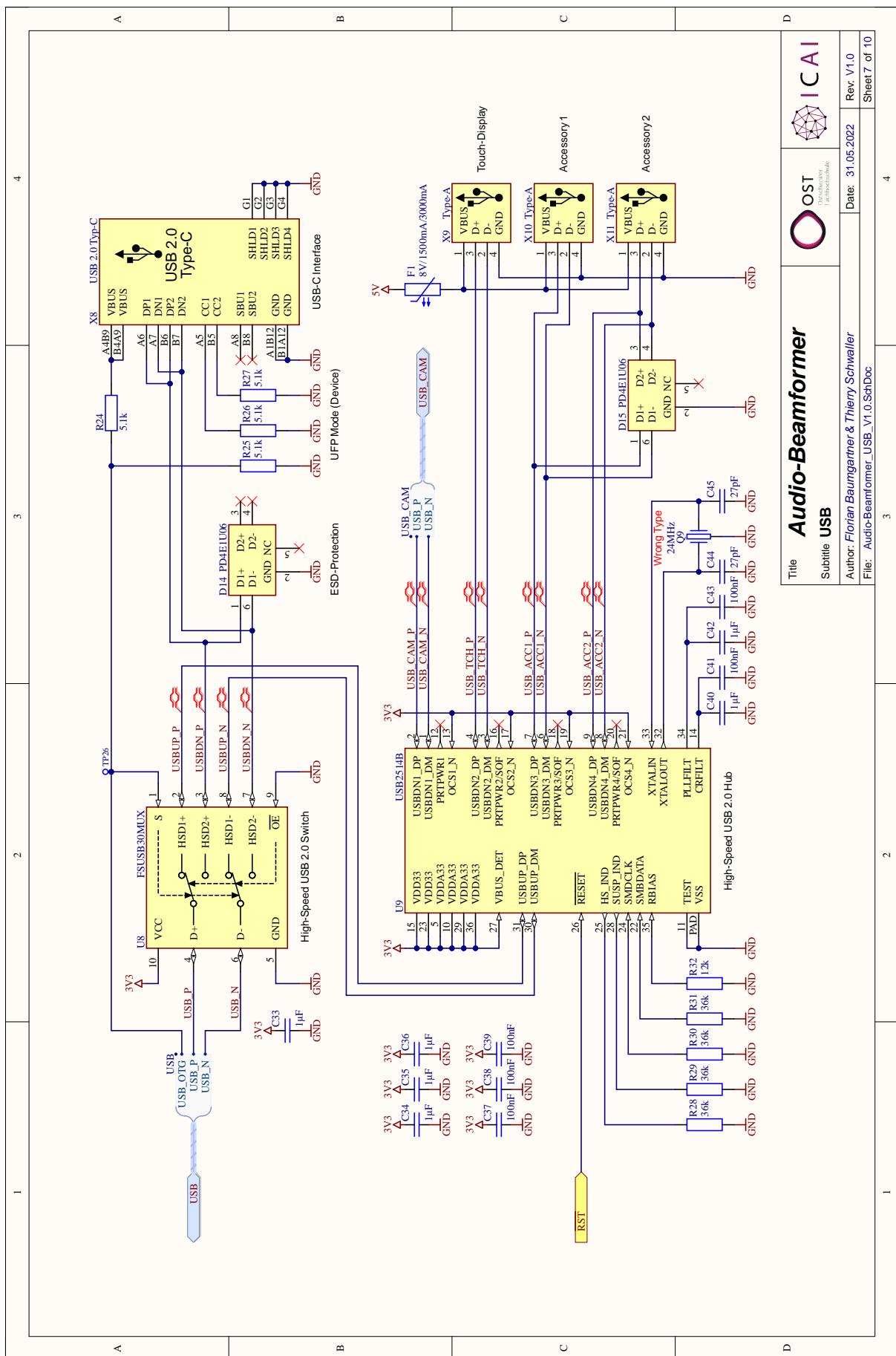




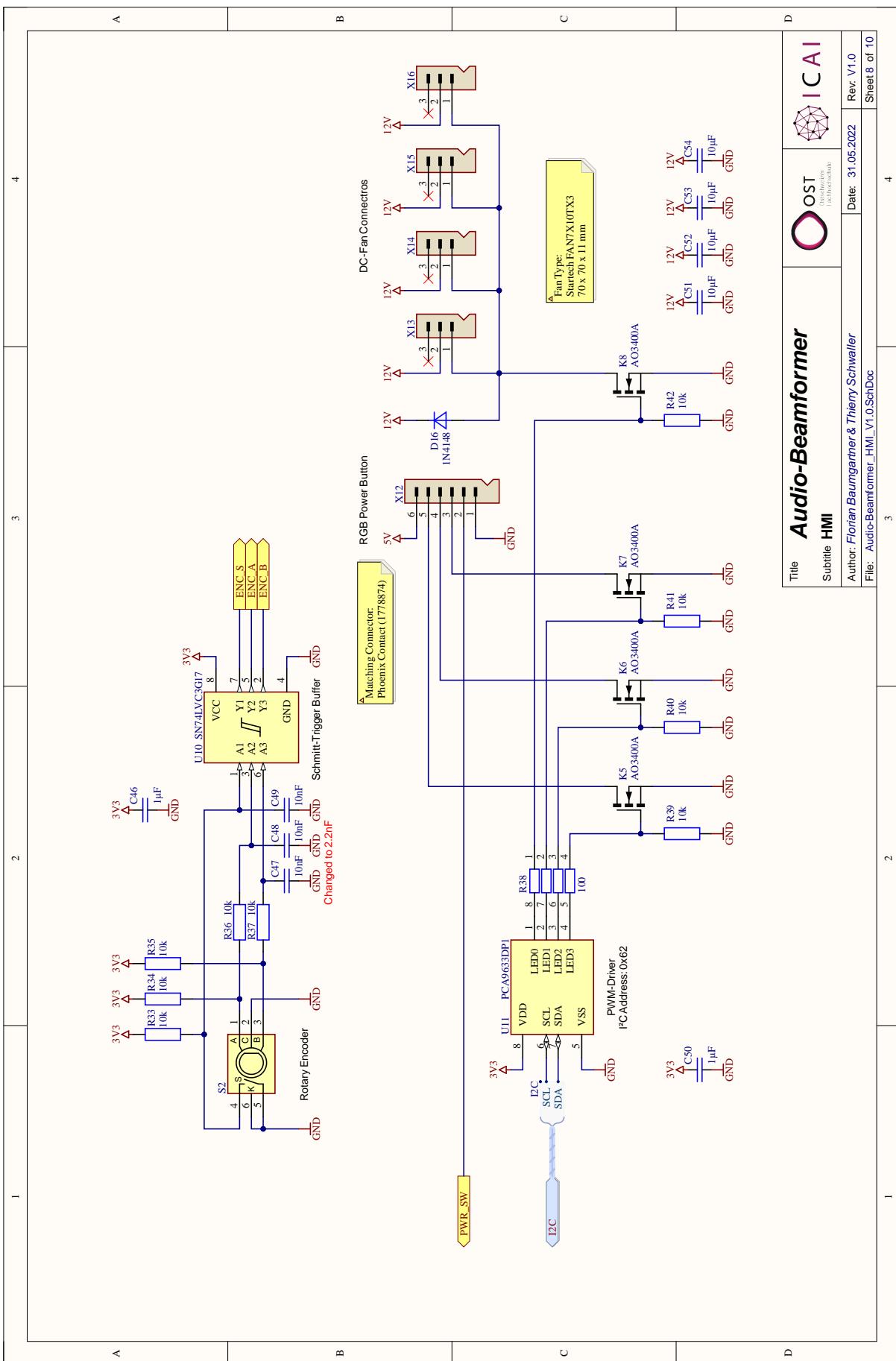


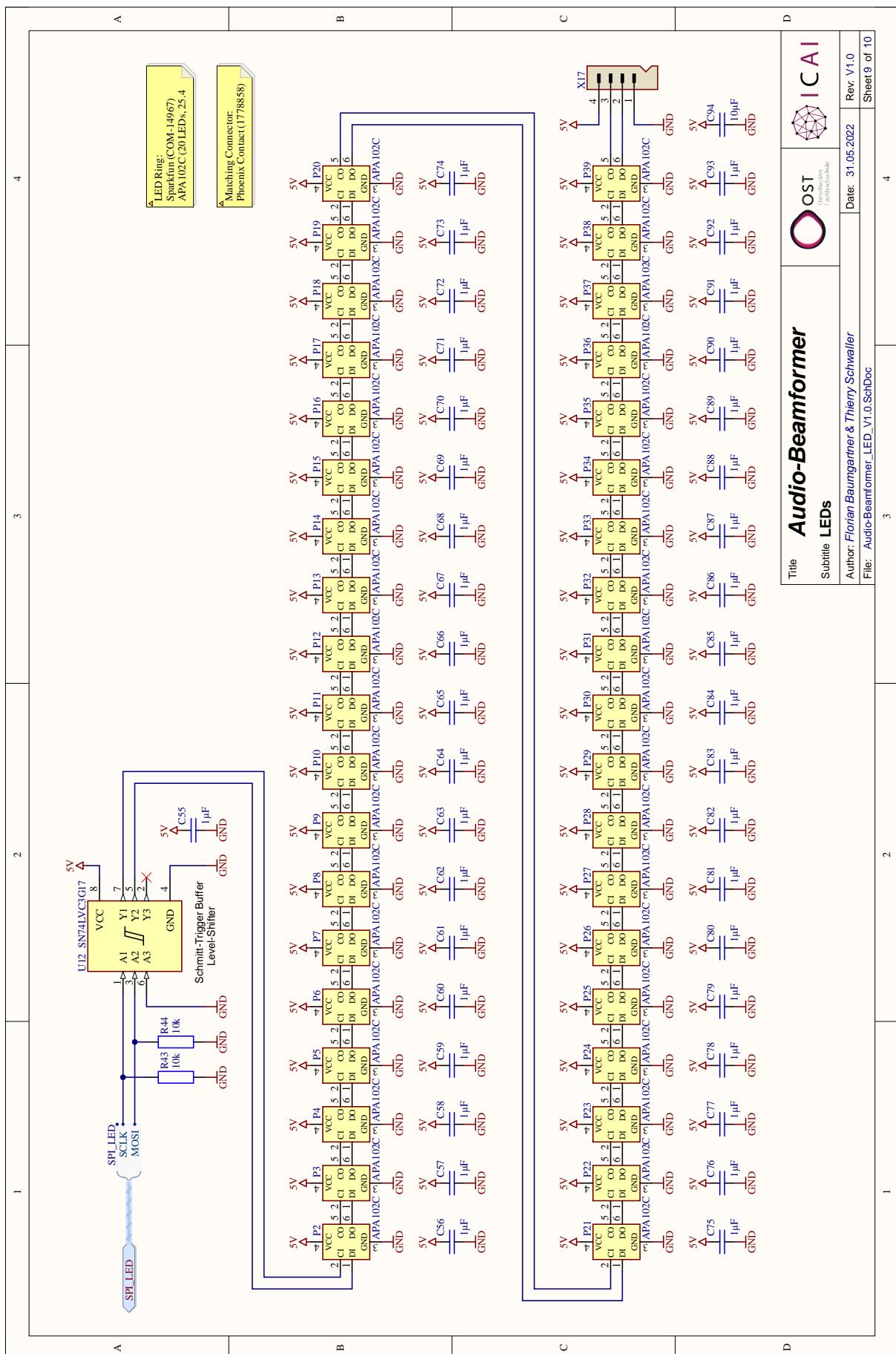


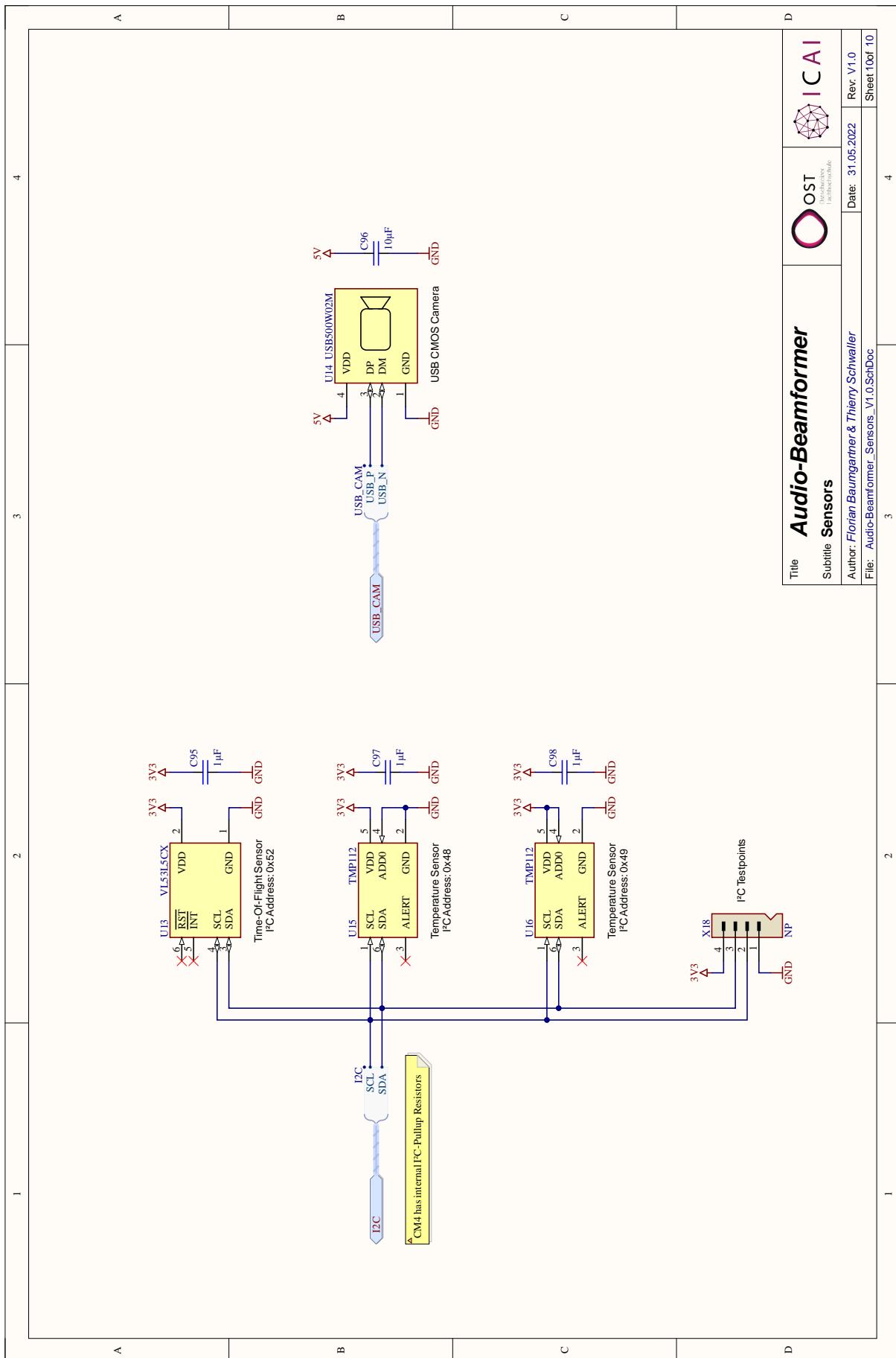




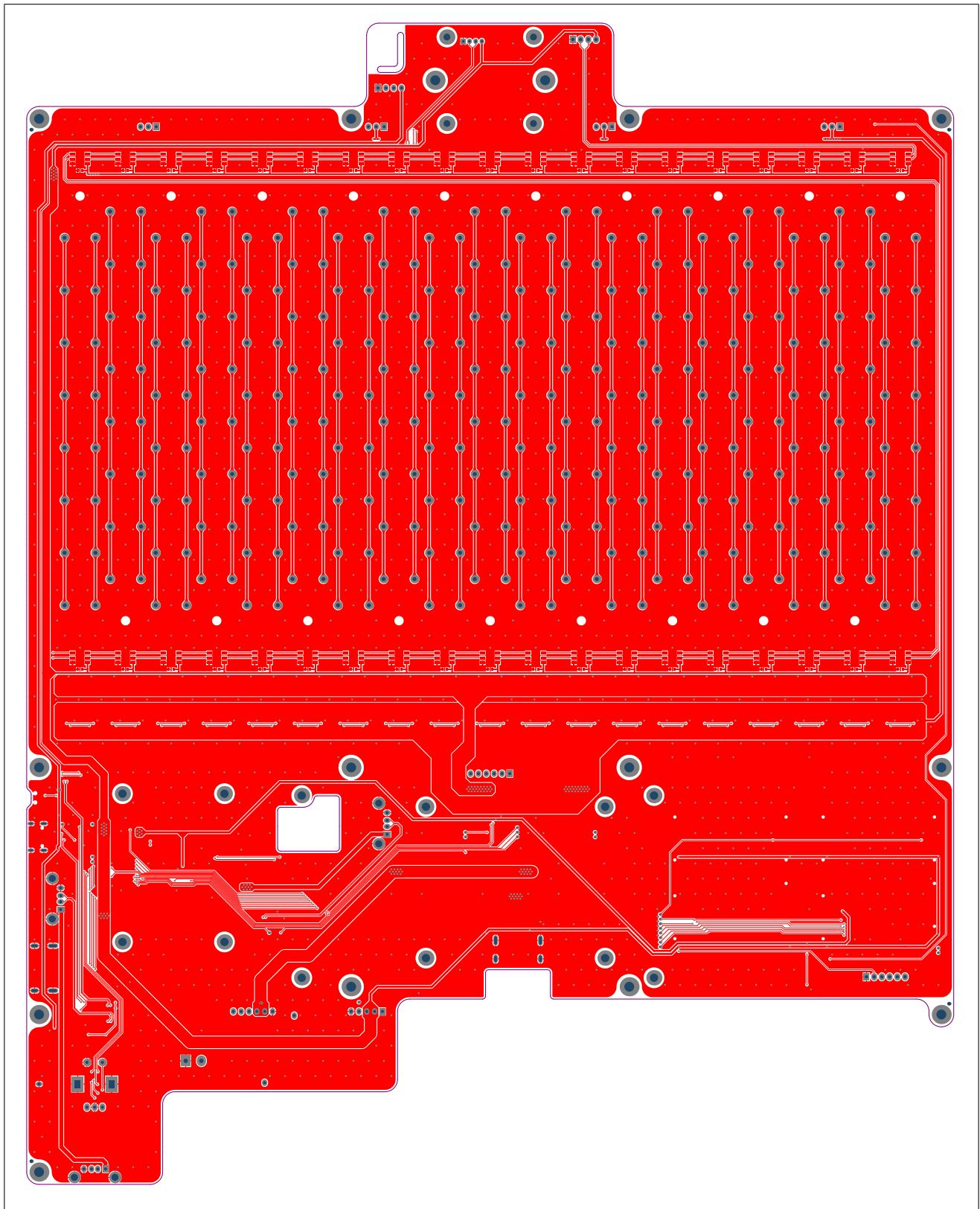
Title: **Audio-Beamformer** | **USB** | **OST** | **CA**
 Subtitle: *Florian Baumgartner & Thierry Schwallier*
 Author: *Florian Baumgartner & Thierry Schwallier*
 File: *Audio-Beamformer_USB_V1.0.SchDoc*
 Date: *31.05.2022* Rev. *V1.0*
 Sheet 7 of 10



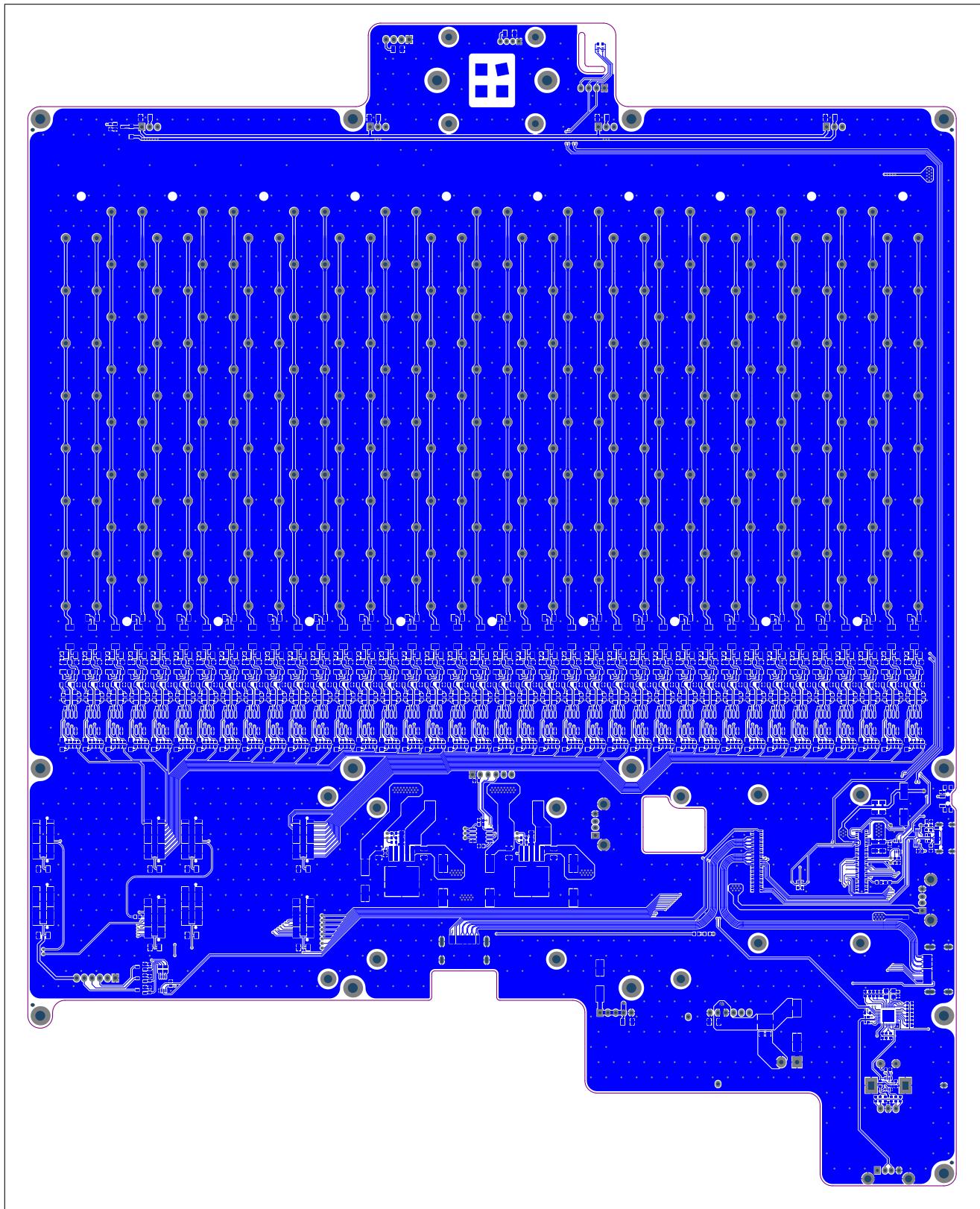




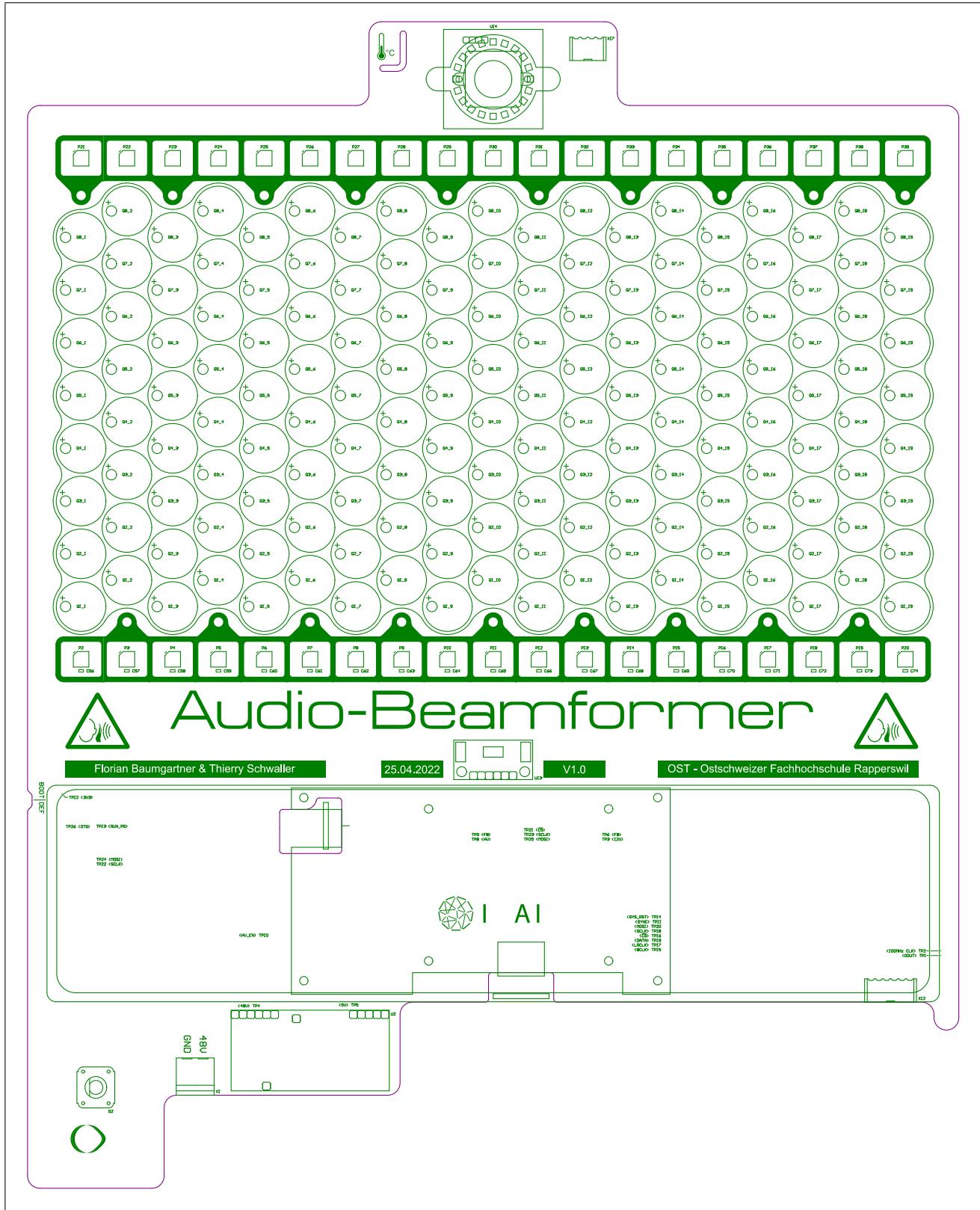
A.7 PCB Top-Layer



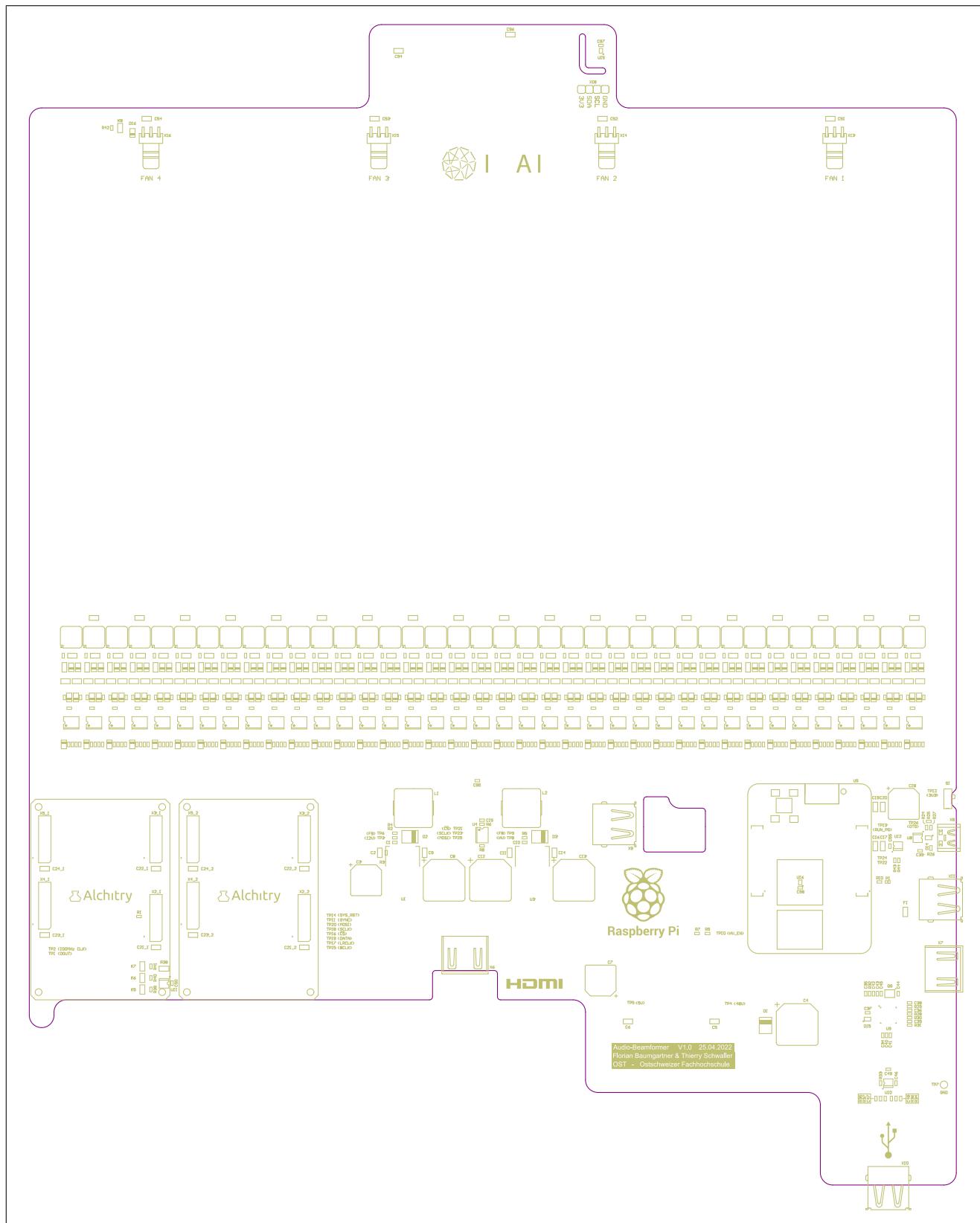
A.8 PCB Bottom-Layer



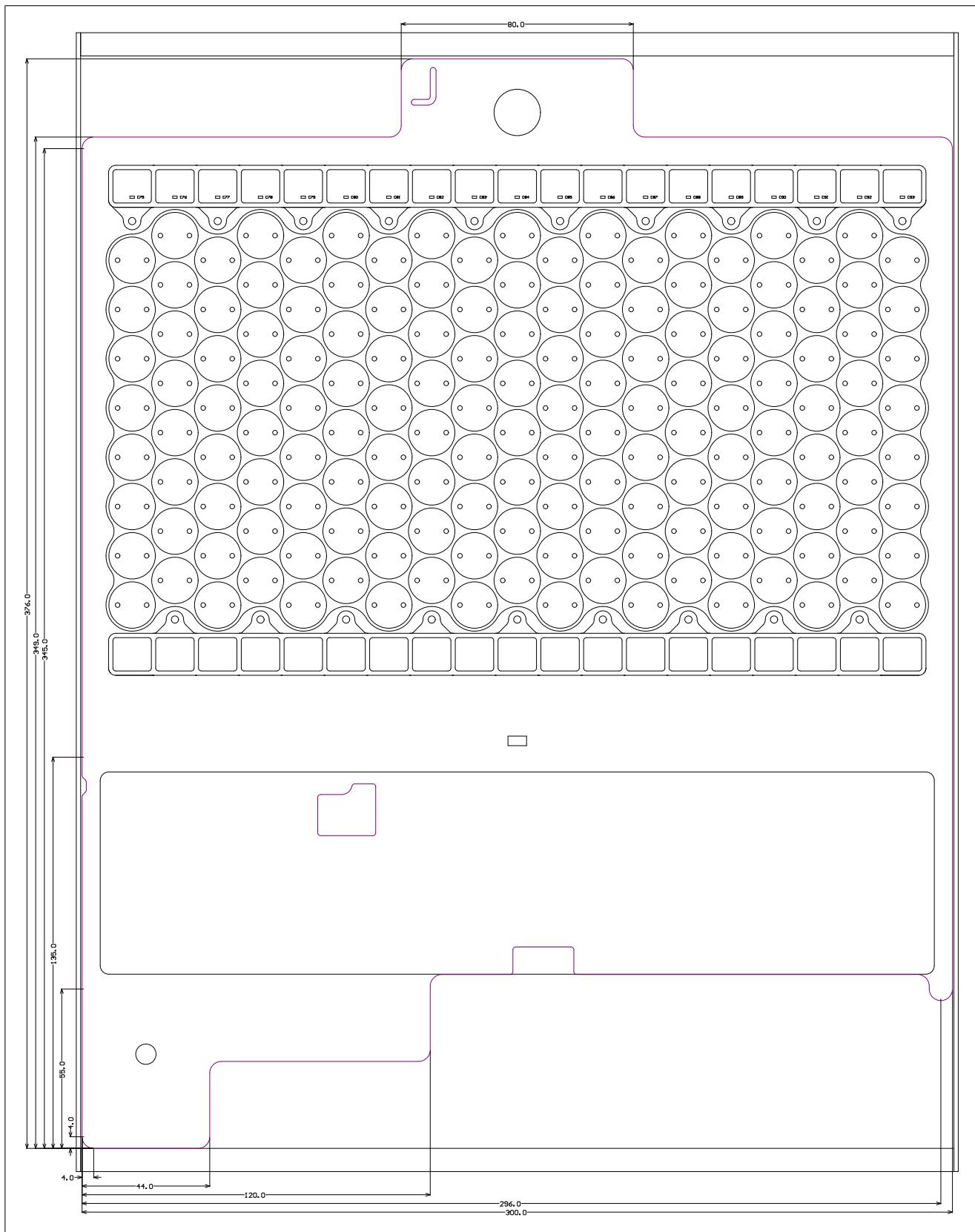
A.9 PCB Top-Overlay



A.10 PCB Bottom-Overlay

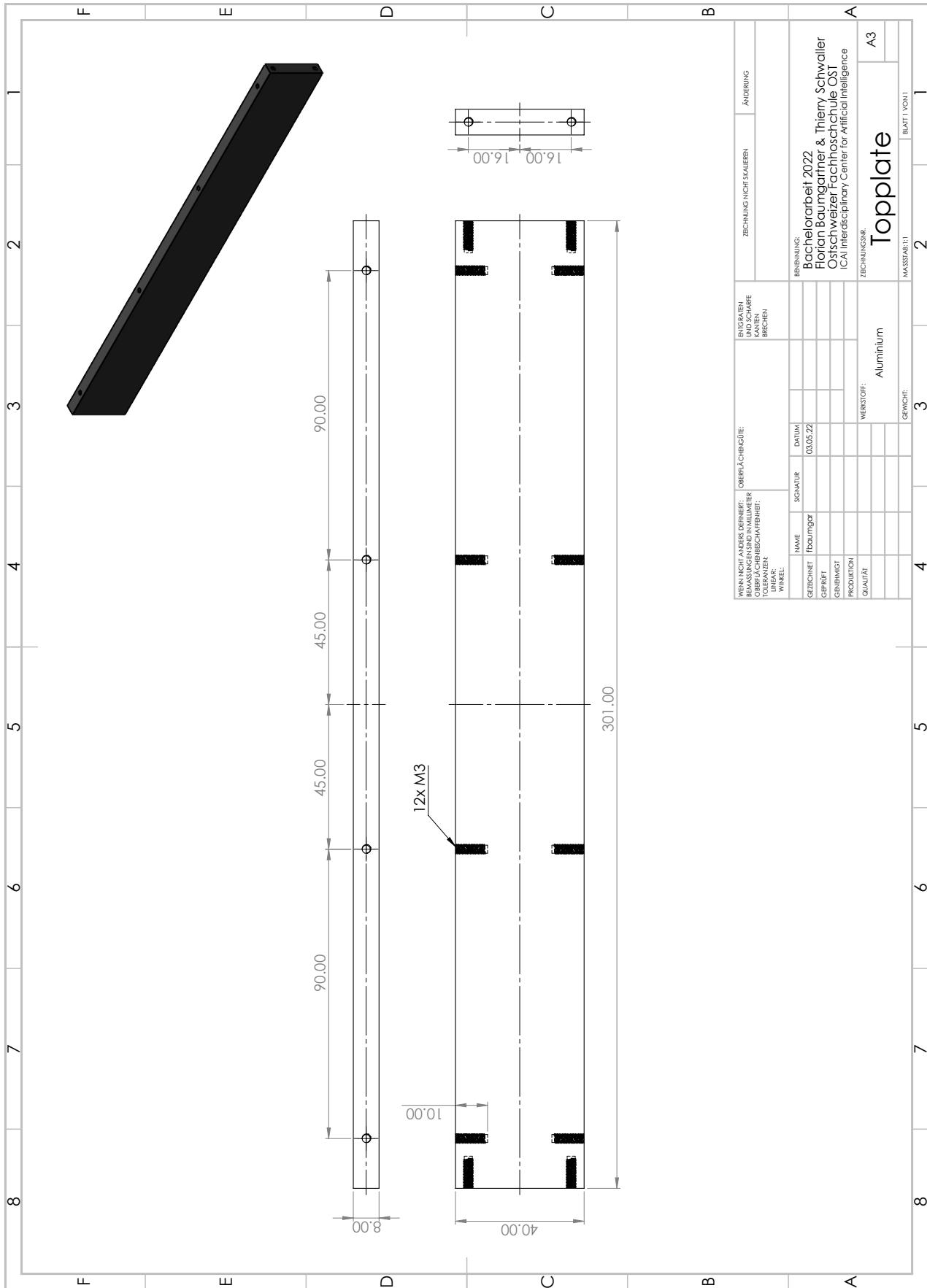


A.11 PCB Outline

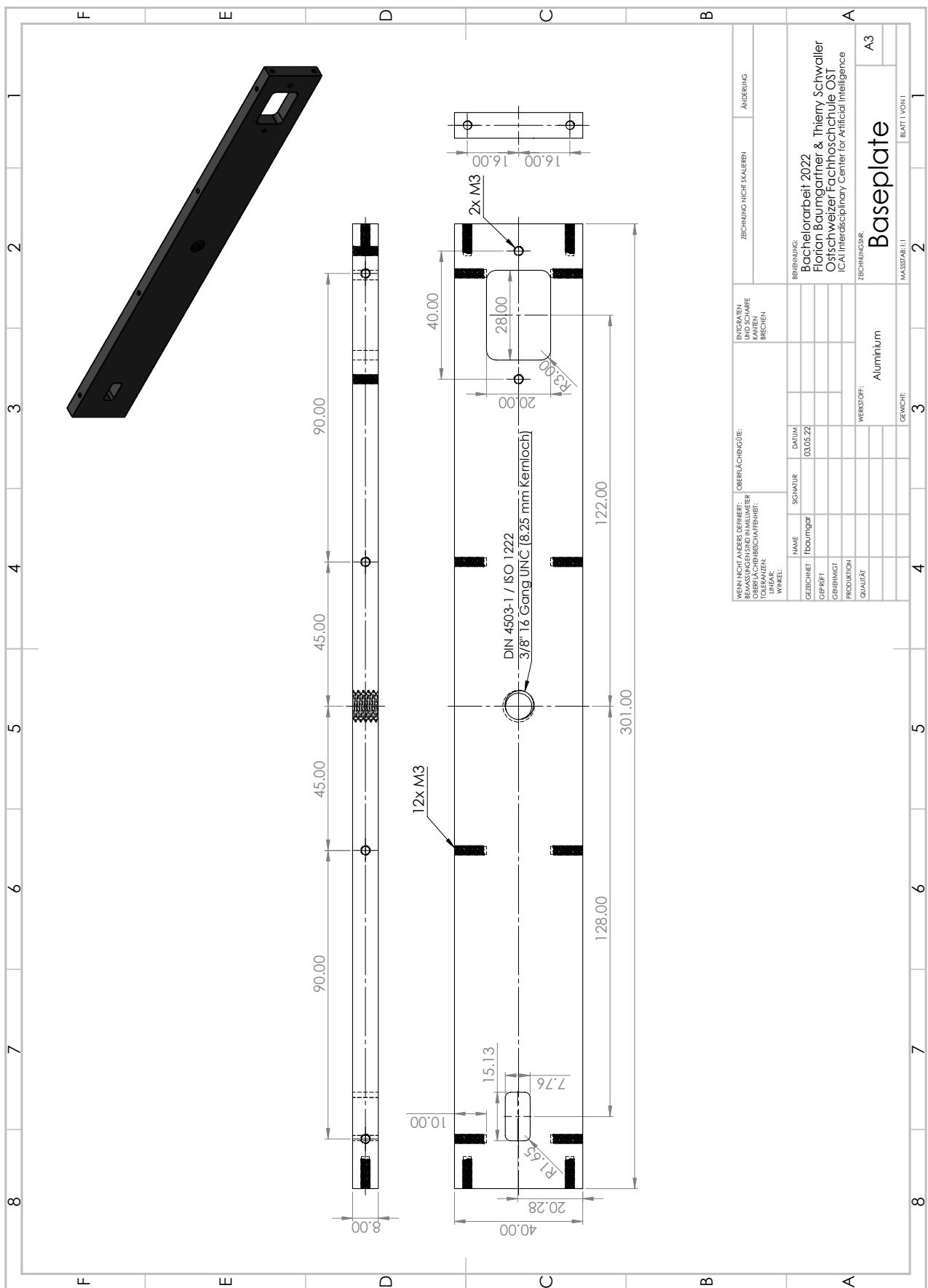


A.12 Bill of Materials (BOM)

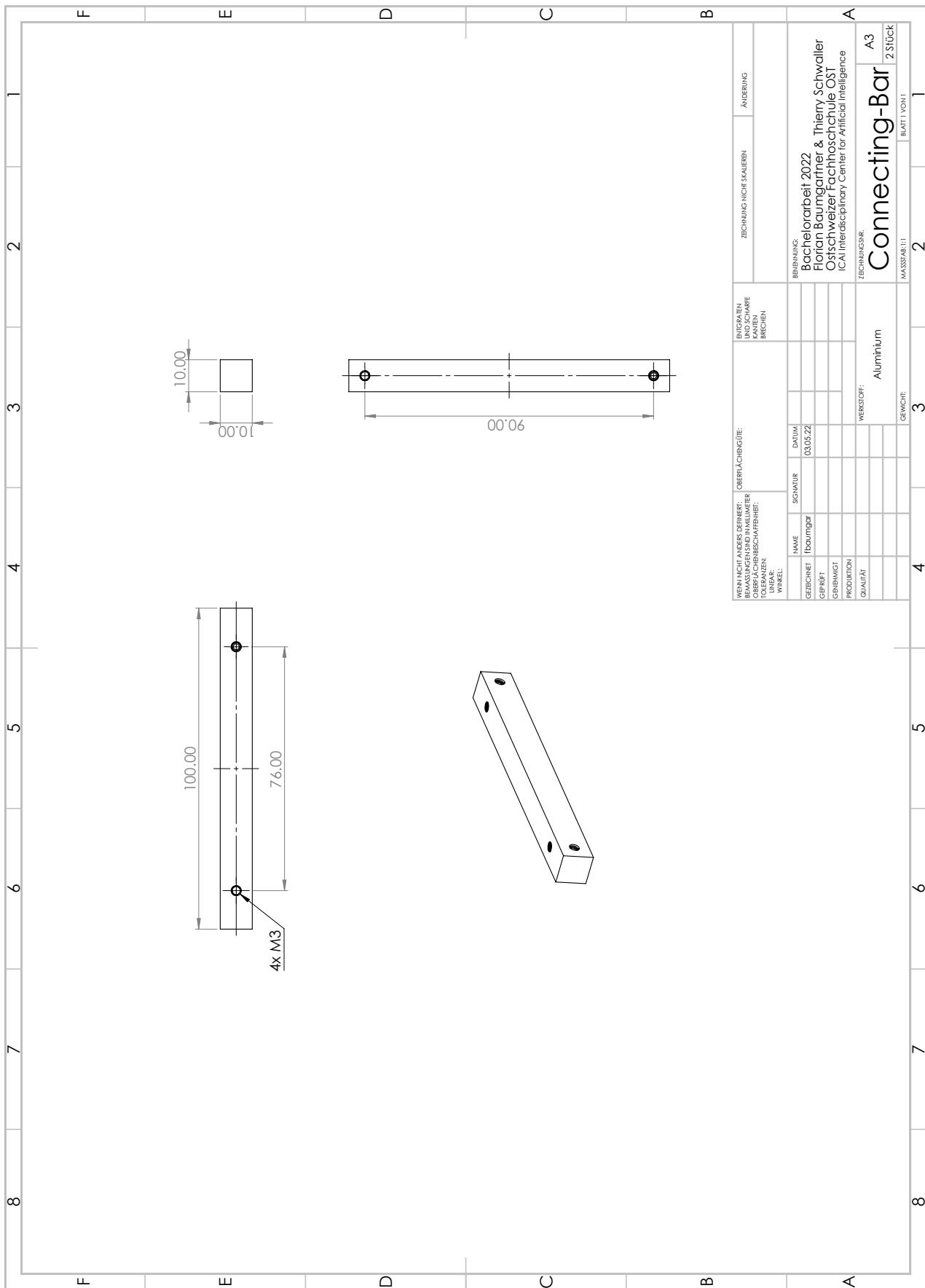
A.13 Mechanical Drawing Top-Plate



A.14 Mechanical Drawing Base-Plate



A.15 Mechanical Drawing Connecting-Bar



Bibliography

- [1] Leo L. Beranek and Tim J. Mellow. Chapter 1 - introduction and terminology. In Leo L. Beranek and Tim J. Mellow, editors, *Acoustics: Sound Fields and Transducers*, pages 1–19. Academic Press, 2012.
- [2] Class D Audio Amplifier Basics. <https://www.infineon.com/dgdl/an-1071.pdf?fileId=5546d462533600a40153559538eb0ff1>. (accessed: 29.05.2022).
- [3] Class-D Output Snubber Design Guide. <https://www.ti.com/lit/an/sloa201/sloa201.pdf?ts=1653184608718>. (accessed: 22.05.2022).
- [4] W. T. W. CORY. Relationship between sound pressure and sound power levels, 2018.
- [5] Pompei Joseph. Sound from ultrasound : the parametric array as an audible sound source, 08 2005.
- [6] Edgar Y. Choueiri Joseph G. Tylka. On the calculation of full and partial directivity indices. Technical report, Princeton University, 2016.
- [7] Thomas Kneubühler. Nachrichtentechnik 1 + 2, 2020.
- [8] Lattice iCE40 FPGA Architecture. <https://www.latticesemi.com/~/media/LatticeSemi/Documents/DataSheets/iCE/iCE40LPHXFamilyDataSheet.pdf>. (accessed: 31.05.2022).
- [9] R. Nave. Sound speed in gases. <http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/souspe3.html>, 2016.
- [10] T.D. Rossing. *The Science of Sound*. Addison-Wesley Publishing Company, 1990.
- [11] B. Ruh, A. Bucher. Physik 2, 2014.
- [12] Lester W. Schmerr Jr. *Fundamentals of Ultrasonic Phased Arrays*. Solid Mechanics and Its Applications, 215. Springer International Publishing, Cham, 1st ed. 2015. edition, 2015.
- [13] Ee-Leng Tan, Peifeng Ji, and Woon-Seng Gan. On preprocessing techniques for bandlimited parametric loudspeakers. *Applied Acoustics*, 71:486–492, 05 2010.
- [14] UCA Calgary. Fraunhofer diffraction. <http://people.ucalgary.ca/~lvov/471/labs/fraunhofer.pdf>.

- [15] Alexander Unger. *Air Coupled Ultrasonic Transducers for Industrial Application.* PhD thesis, Technische Universität Darmstadt, July 2018.
- [16] Peter J. Westervelt. Parametric acoustic array. *The Journal of the Acoustical Society of America*, 35(4):535–537, 1963.
- [17] Masahide Yoneyama, Jun-ichiroh Fujimoto, Yu Kawamo, and Shoichi Sasabe. The audio spotlight: An application of nonlinear interaction of sound waves to a new type of loudspeaker design. *The Journal of the Acoustical Society of America*, 73(5):1532–1536, 1983.