

Acoustic Source Localization using a Microphone Array

Project Thesis

Florian Baumgartner

florian.baumgartner@ost.ch

Alain Keller

alain.keller@ost.ch

Advisor

Hannes Badertscher

Interdisciplinary Center for Artificial Intelligence
Eastern Switzerland University of Applied Sciences

January 2024

Abstract

In recent years, the rising popularity of drones has led to increased concerns about privacy, espionage, and military misuse. As a result, the development of effective drone localization systems has become crucial. Existing technologies primarily rely on visual, thermal, or radar detection, each with its limitations, especially in low visibility conditions. Humans, however, have a natural ability to locate objects, such as drones, by their audible noise. This observation has inspired the aim of this project, which is to achieve acoustic localization of objects, particularly drones, using sound detection.

The project's approach involves using an array of multiple microphones in conjunction with beamforming algorithms to determine the direction of a sound source. However, identifying the sound's direction alone is insufficient to pinpoint an object's exact location. Therefore, multiple arrays, placed at different locations, are combined to estimate the target's position. The primary focus of this project was the development of a microphone array optimized to analyze the direction of a sound source. Through a custom-built simulation environment and validation measurements, an optimal array geometry was identified. It comprises 32 MEMS microphones in a unique cone-shaped, three-dimensional arrangement based on eight adjustable arms. A specialized hardware setup was developed to process audio in real time and transmit the data over a LAN network to a centralized computer running advanced beamforming algorithms. A peak finder and Kalman tracker was implemented to detect and track sound sources, respectively. The developed applications provides an interactive web interface that visualizes directional sound power heatmaps and a map view displaying each object's location.

The simulation results were compared to real-world measurements and proved to be accurate. The uniquely designed array geometry and beamforming algorithms enabled precise localization of drones at distances exceeding 70 meters. This achievement met the project's requirements, demonstrating the system's potential in various applications beyond drone detection due to its flexible setup. The project lays a strong foundation for future advancements, such as object classification and enhanced robustness in acoustically crowded environments.

Acknowledgement

We are deeply grateful to all who supported our master's project, with special thanks to Hannes Badertscher for his dedicated supervision. Our appreciation extends to Prof. Dr. Guido Schuster and the Interdisciplinary Center for Artificial Intelligence (ICAI) for their generous financial backing.

We also acknowledge Caspar Naef and Marcel Kluser for their promptness in component procurement, and Uwe Heiligensetzer with his team for manufacturing all mechanical parts in time. Special thanks to Joel Meier, whose vital mechanical review and insights notably enhanced our project's quality. Our gratitude goes to Prof. Dr. Markus Kottmann and the Electrical Engineering Department for generously funding this project.

Finally, we extend our sincere thanks to Maya and her team for their exceptional hospitality, providing us with nourishing meals that fueled our work and focus.

Contents

1	Introduction	5
1.1	Background	5
1.2	Scope	6
1.3	Approach	7
1.4	Open Source	7
2	Preliminaries	9
2.1	MEMS Microphones	9
2.1.1	Analog Microphones	10
2.1.2	PDM Microphones	10
2.1.3	PCM Microphones	10
2.1.4	Microphone Port Location	11
2.2	Pulse Density Modulation (PDM)	11
2.3	Time Division Multiplexing (TDM)	12
2.4	GNSS Real-Time Kinematic (RTK)	13
2.4.1	Custom Node as an Anchor	13
2.4.2	Using Data from a Public Provider	13
3	Sound Source Localization	15
3.1	Physical Background	15
3.2	Sound Source Localization Methods	15
3.2.1	Power Based SSL	15
3.2.2	Time Based SSL	16
3.2.3	Direction of Arrival Estimation	17
3.2.4	Beamforming	17
3.3	Tracking	22
3.4	Simulator	24
3.4.1	Simulation Model	24
3.4.2	Output	24
4	Acquisition-System Design	25
4.1	Overview	25
4.2	Key Requirements	26
4.3	Key Decisions	26
4.4	Microphone Evaluation	27
4.4.1	Microphone Types	27
4.4.2	Microphone Breakout Boards	27
4.5	Hardware Design	29

4.5.1	Block Diagram	30
4.5.2	Microcontroller Unit (MCU)	30
4.5.3	Audio Input	31
4.5.4	Headphones Output	31
4.6	Firmware Design	32
4.6.1	Audio Recording	32
4.6.2	Graphical User Interface (GUI)	34
4.6.3	GUI Pages	34
5	Array Evaluation	37
5.1	Overview	37
5.2	Metrics	38
5.2.1	Area Ratio	38
5.2.2	Peak-to-Average Power Ratio	38
5.3	Array Geometry	38
5.3.1	Circular Array	39
5.3.2	Multi Circular Array	40
5.3.3	Archimedean Spiral Array	42
5.4	Mechanical Design	43
5.4.1	Multi Circular Array	43
5.4.2	Archimedean Spiral Array	43
5.4.3	Wooden Prototype Arrays	44
5.5	Measurements & Findings	44
5.6	Final Array Geometry	46
6	Final Design	49
6.1	Overview	49
6.1.1	Key Requirements	50
6.1.2	Key Decisions	50
6.2	Mechanical Design	51
6.2.1	Part List	52
6.2.2	Microphone Arm Design	52
6.2.3	Folding Mechanism	53
6.2.4	Microphone Wind Protection	53
6.3	Hardware Design	54
6.3.1	Block Diagram	55
6.3.2	Power Supply	56
6.3.3	Microcontroller Unit (MCU)	57
6.3.4	Audio Input	57
6.3.5	GNSS Receiver	57
6.3.6	Display Module	58
6.3.7	RGB LEDs	58
6.3.8	Sensors	58
6.3.9	Printed Circuit Boards (PCBs)	60
6.3.10	Manufacturing	60
6.4	Firmware Design	61
6.4.1	Overview	61
6.4.2	Audio Streaming	61
6.4.3	GNSS & RTC Time	63

6.4.4	Remote Configuration	64
6.4.5	Sensor Calibration	65
6.4.6	Human Machine Interface (HMI)	66
6.4.7	Graphical User Interface (GUI)	66
6.4.8	GUI Pages	66
6.5	Software Design	71
6.5.1	Audio Interface	71
6.5.2	Communication	71
6.5.3	Sound Source Tracking	71
6.5.4	GUI	72
7	Measurements	75
7.1	Overview	75
7.2	Stationary Drone	75
7.3	Moving Drone Test	77
7.4	Conclusion	77
8	Conclusion	79
8.1	Continuing Work	80
8.2	Personal Reflections	80
A	Appendix	81
A.1	Declaration of Authorship	82
A.2	Data Archive	83
A.3	Definition of Task	84
A.4	Datasheet of MEMS Microphone (MP34DT05TR-A)	86
A.5	Microphone-Boards Schematics	87
A.6	Microphone-Boards PCB Top-Layer	92
A.7	Microphone-Boards PCB Bottom-Layer	93
A.8	Microphone-Boards PCB Top-Overlay	94
A.9	Microphone-Boards PCB Bottom-Overlay	95
A.10	Microphone-Boards PCB Outline	96
A.11	Microphone-Boards Bill of Materials (BOM)	97
A.12	Acquisition-System Schematics	98
A.13	Acquisition-System PCB Top-Layer	103
A.14	Acquisition-System PCB Mid-Layer 1	104
A.15	Acquisition-System PCB Mid-Layer 2	105
A.16	Acquisition-System PCB Bottom-Layer	106
A.17	Acquisition-System PCB Top-Overlay	107
A.18	Acquisition-System PCB Bottom-Overlay	108
A.19	Acquisition-System PCB Outline	109
A.20	Acquisition-System Bill of Materials (BOM)	110
A.21	Mainboard Schematics	111
A.22	Mainboard PCB Top-Layer	118
A.23	Mainboard PCB Mid-Layer 1	119
A.24	Mainboard PCB Mid-Layer 2	120
A.25	Mainboard PCB Bottom-Layer	121
A.26	Mainboard PCB Top-Overlay	122
A.27	Mainboard PCB Bottom-Overlay	123

A.28 Mainboard PCB Outline	124
A.29 Mainboard Bill of Materials (BOM)	125
A.30 Microphone-Arm Schematics	126
A.31 Microphone-Arm PCB Top-Layer	127
A.32 Microphone-Arm PCB Bottom-Layer	128
A.33 Microphone-Arm PCB Top-Overlay	129
A.34 Microphone-Arm Bill of Materials (BOM)	130
A.35 Angle-Sensor Schematics	131
A.36 Angle-Sensor PCB Top-Layer	132
A.37 Angle-Sensor PCB Bottom-Layer	133
A.38 Angle-Sensor PCB Top-Overlay	134
A.39 Angle-Sensor PCB Bottom-Overlay	135
A.40 Angle-Sensor PCB Outline	136
A.41 Angle-Sensor Bill of Materials (BOM)	137
A.42 Mechanical Drawing of Main Mounting Pole	138
A.43 Mechanical Drawing of Top Mounting Ring	139
A.44 Mechanical Drawing of Bottom Sliding Ring	140
A.45 Mechanical Drawing of Antenna Top Mount	141
A.46 Mechanical Drawing of Antenna Bottom Mount	142
A.47 Mechanical Drawing of Antenna Pole	143
Bibliography	145

Acronyms

- API** Application Programming Interface. 30, 61
- ARM** Advanced RISC Machine. 26
- CAD** Computer Aided Design. 49
- CD** Compact Disc. 22
- CNC** Computerized Numerical Control. 49
- CODEC** Coder-Decoder. 27
- CPU** Central Processing Unit. 28, 65
- DAC** Digital-to-Analog Converter. 27
- DOA** Direction of Arrival. 13, 14, 18, 34, 40
- FPGA** Field Programmable Gate Array. 6
- FPS** Frames per Second. 56
- GLONASS** Global Navigation Satellite System. 55
- GNSS** Global Navigation Satellite System. 9, 48, 54, 55, 58, 61, 65, 66
- GPS** Global Positioning System. 9, 55
- GUI** Graphical User Interface. 30, 48, 64
- HMI** Human Machine Interface. 64
- HTTP** Hypertext Transfer Protocol. 62
- I²C** Inter-Integrated Circuit. 23, 27, 55, 56
- I²S** Inter-IC Sound. 6, 27
- IC** Integrated Circuit. 27, 54, 55, 58
- IDE** Integrated Development Environment. 28
- IEEE** Institute of Electrical and Electronic Engineers. 54
- IMU** Inertial Measurement Unit. 48
- IoT** Internet of Things. 5
- IP** Internet Protocol. 66
- IRQ** Interrupt Request. 28
- JSON** JavaScript Object Notation. 62
- LAN** Local Area Network. 48
- LED** Light-Emitting Diode. 22, 23, 25, 48, 52, 55–57, 64, 68
- LNA** Low-Noise Amplifier. 55
- LSB** Least Significant Bit. 29
- LVGL** Light and Versatile Graphics Library. 30

- MAC** Media Access Control. 65
- MCU** Microcontroller Unit. 6, 22, 26, 54, 56, 59, 61
- MEMS** Micro-Electro-Mechanical Systems. 5–7, 23, 52, 55, 58
- MSB** Most Significant Bit. 29
- OST** Ostschweizer Fachhochschule. 68
- PC** Personal Computer. 63
- PCB** Printed Circuit Board. 7, 23–25, 48, 52, 53, 57, 58
- PCM** Pulse-Code Modulation. 5, 6
- PDM** Pulse Density Modulation. 5–7, 22–24, 27, 54, 55, 58
- PI** Proportional-Integral. 61
- PLL** Phase-Locked Loop. 61
- PoE** Power over Ethernet. 48, 54
- PSRAM** Pseudo Static Random Access Memory. 26, 29, 55
- PWM** Pulse-Width Modulation. 56, 57
- QR-Code** Quick Response Code. 66
- RAM** Random Access Memory. 26
- RF** Radio Frequency. 55
- RGB** Red Green Blue. 22, 25, 48, 52, 56, 64
- RTC** Real-Time Clock. 22, 56, 61
- RTK** Real-Time Kinematic. 9, 48, 55, 62
- SD** Secure Digital Memory Card. 22, 26, 28–30
- SDIO** SD Input/Output. 22
- SMA** SubMiniature Version A. 55
- SNR** Signal-to-Noise Ratio. 23
- SoC** System on a Chip. 26, 55
- SPI** Serial Peripheral Interface. 55, 56
- SSL** Sound Source Localization. 11, 12
- TCP** Transmission Control Protocol. 48, 59, 60
- TDM** Time Division Multiplexing. 8, 27, 54, 55, 58
- TDOA** Time Difference of Arrival. 12
- TFT** Thin-Film Transistor. 22, 25, 56
- UART** Universal Asynchronous Receiver Transmitter. 55
- UDP** User Datagram Protocol. 48
- UI** User Interface. 22, 31
- UNIX** Uniplexed Information and Computing Service. 60
- URL** Uniform Resource Locator. 62, 66
- USB** Universal Serial Bus. 22, 25, 26, 30, 54, 55, 65
- VS Code** Visual Studio Code. 28

Glossary

Adafruit Adafruit Industries is an open-source hardware company based in New York City that designs and manufactures electronic development boards. 23, 24

Arduino Is an open-source company providing software libraries and microcontroller kits. 28

1

Introduction

1.1 Background

The ability to detect the direction of a sound source is a remarkably intuitive skill possessed by humans and animals. This capability is facilitated by the use of just two ears, leading to what is essentially an underdefined system. The uniquely shaped structure of the ears allows the brain to discern subtle differences in sound reception, enabling the identification of whether a sound originates from the front or behind. Replicating this natural phenomenon in a technical setup presents a significant challenge, as the human brain employs complex auditory processing methods honed over a lifetime of learning.

A practical approach to addressing this challenge in sound source localization is the employment of multiple microphones, akin to having more ears. By strategically arranging these microphones in an array configuration, it becomes feasible to utilize beamforming algorithms to ascertain the direction of a sound source.

With the rising popularity of drones, instances of their misuse have also increased. This trend has sparked interest in the development of detection and tracking systems for drones. Utilizing a sound localization system for this purpose offers distinct advantages over other technologies that depend on optical or thermal detection. Humans often rely on their auditory sense as an initial alert to the presence and location of a drone, a method that remains effective even in adverse weather conditions where visual sighting is compromised, or in scenarios where thermal detection is ineffective, such as when a drone is silhouetted against the sun. Compared to radar-based tracking systems, acoustic drone localization technology is considerably simpler and more cost-effective, making it an attractive solution for drone detection and tracking.

1.2 Scope

The primary objective of this project was to conceptualize and develop a system capable of acoustically pinpointing objects, such as drones. The inherent limitation of a single microphone array is its ability to determine only the direction of a sound source, not its distance. Consequently, it became apparent that accurate localization of drones in the three-dimensional space necessitates the integration of multiple microphone arrays. Each array contributes a directional vector towards the sound source. The exact position of the object is then deduced by finding the theoretical point of intersection of these vectors, considering the known locations and orientations of each array. Figure 1.1 illustrates this concept.

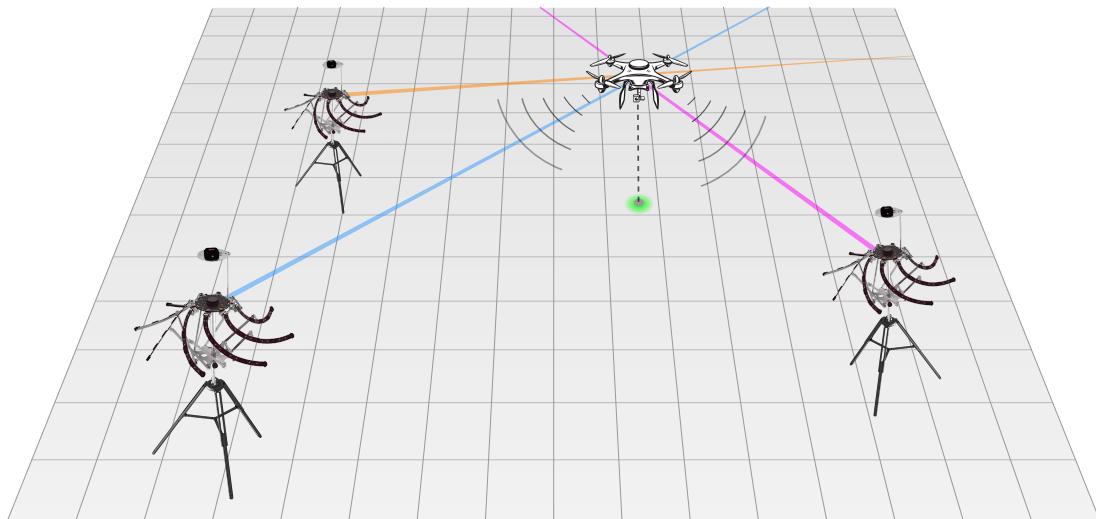


Figure 1.1: Areal Visualizations of Multiple Microphone Arrays

It is important to note that the main focus of this project centers on the development of the microphone array itself, which is responsible for providing a vector to the sound source. The task of combining multiple arrays to enhance the localization accuracy is therefore not directly addressed in this project.

1.3 Approach

In the first part of the project, the emphasis was placed on researching array geometries and beamforming techniques, while simultaneously developing a microphone evaluation and audio acquisition system. A key component was the establishment of a simulation environment to compare various array geometries and understand different influences of parameters.

Based on these simulations, two array geometries were selected and constructed as prototypes. This step allowed the acquisition of real-world measurements that were further compared against the simulated results. Insights from these measurements were key in determining the final array geometry.

Finally, a fully functioning professional microphone array was developed, featuring 32 MEMS microphones arranged on 8 tiltable arms. This three-dimensional cone-shaped structure fulfilled the project's requirements. Specialized hardware and firmware was developed for real-time audio data processing and for transmitting high-quality audio data to a centralized host computer, on which, beamforming algorithms were implemented. Based on a peak detector and Kalman filter, the application is capable of estimating the direction of a sound source and tracking its movement. Additionally, an interactive web interface was designed to provide visual insights, displaying a sound power heatmap and visualizing the located objects on a map.

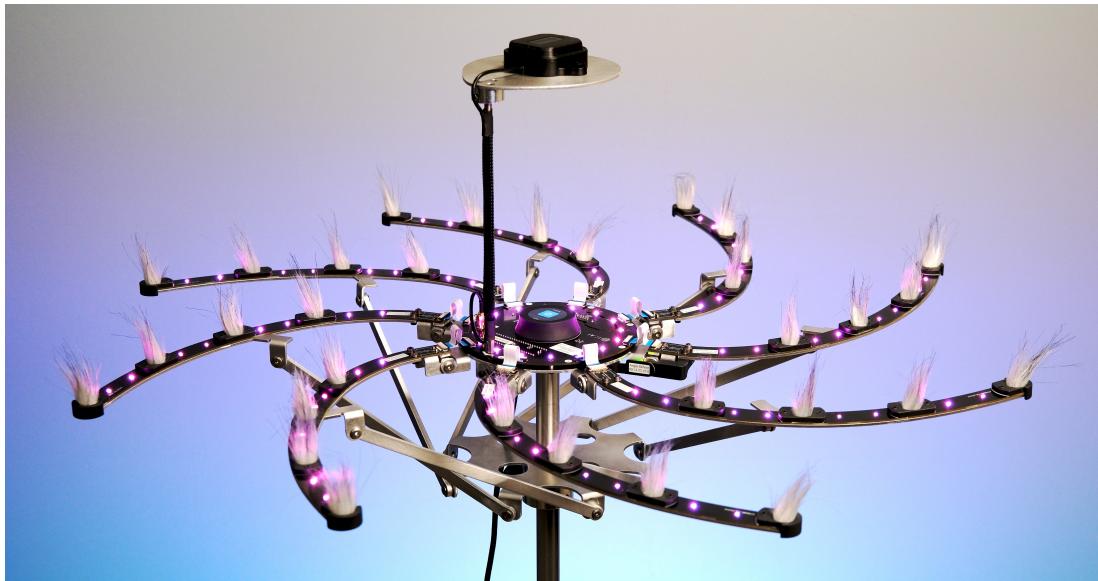


Figure 1.2: Final Product

1.4 Open Source

The project started with a commitment to open source principles, motivated by the belief that open source brings significant advantages to engineering. This approach allowed for the integration of existing open source libraries and code, which significantly accelerated the design process. All documents and files for this project can be found on GitHub. A concise summary of each repository is available in the Appendix, as referenced in A.2.

2

Preliminaries

2.1 MEMS Microphones

Micro-Electro-Mechanical Systems (MEMS) microphones represent a significant evolution in acoustic technology. Unlike traditional microphones that rely on larger, more mechanically complex systems, MEMS microphones integrate acoustic sensing elements with electronic circuits on a tiny silicon chip. These microphones have gained immense popularity due to their compact size, robustness, and cost-effectiveness. MEMS technology allows for the production of microphones with high sound quality and excellent reliability, making them ideal for a wide range of applications including mobile devices, wearable technology, and IoT devices. Their small footprint also enables design flexibility in increasingly miniaturized electronic devices. MEMS microphones differ in their output signal types, leading to three categories: analog microphones, PDM microphones, and PCM microphones.

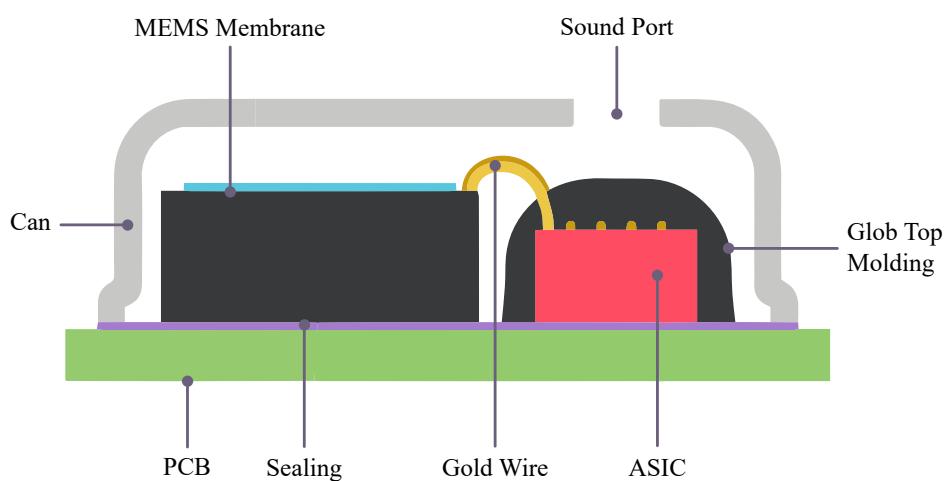


Figure 2.1: Internal Structure of a MEMS Microphone [5]

2.1.1 Analog Microphones

Analog MEMS microphones convert sound into an analog electrical signal. They are simple and easy to integrate in analog circuits but may require additional signal amplification components. A disadvantage of analog microphones is that they are susceptible to noise and interference, making them unsuitable for long-distance transmission.

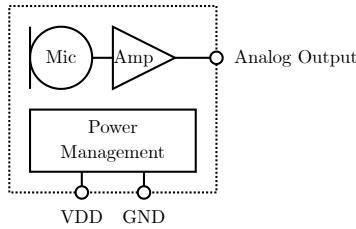


Figure 2.2: Block Diagram of analog MEMS Microphone

2.1.2 PDM Microphones

PDM microphones output a digital signal representing the acoustic waveform. Their digital format is noise-resistant and supports long-distance transmission, making them ideal for multiplexed, multi-microphone setups. These microphone types require a high-frequency clock signal to operate (typically 1.5 MHz to 3.25 MHz), which must be provided by the host (e.g. a MCU or FPGA). A dedicated channel select pin is used to select the microphone's output channel in multiplexed systems.

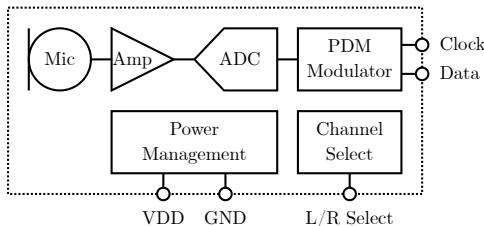


Figure 2.3: Block Diagram of PDM MEMS Microphone

2.1.3 PCM Microphones

PCM microphones provide a digital output using the Pulse-Code Modulation format, most often interfaced via the I²S protocol. Compared to PDM microphones, PCM microphones have a build-in decimation filter, which simplifies the signal processing chain, as the host no longer needs to perform this task. However, PCM microphones are more complex, costly and less common in the industry than PDM microphones.

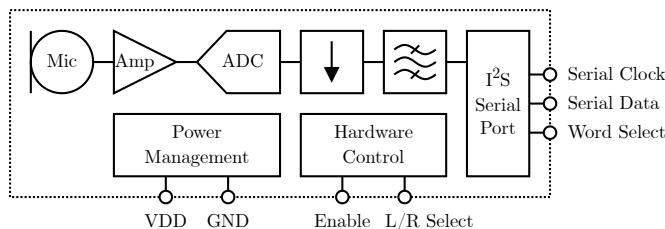


Figure 2.4: Block Diagram of PCM MEMS Microphone

2.1.4 Microphone Port Location

MEMS microphones can be categorized based on their port location: top-port and bottom-port. Top-port microphones have the sound inlet on the top of the package, suitable when the sound source is above the microphone. Conversely, bottom-port microphones have the inlet at the bottom, ideal for mounting on surfaces where sound comes from the side or below. For bottom-port microphones, the sound must travel through a hole in the PCB to reach the microphone, which can affect the sound quality.

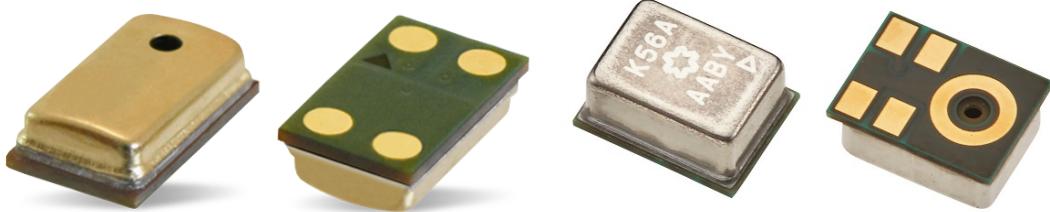


Figure 2.5: Top-Port MEMS Microphone

Figure 2.6: Bottom-Port MEMS Microphone

2.2 Pulse Density Modulation (PDM)

Pulse Density Modulation (PDM) is a modulation technique used to represent an analog signal with a binary sequence. In MEMS microphones, the diaphragm movements modulate a high-frequency carrier signal, resulting in a digital output where the density of pulses corresponds to the amplitude of the input signal. PDM simplifies the microphone design, allowing for smaller and more power-efficient devices. However, it requires a decimation filter in the signal processing chain to convert the high-frequency pulse sequence into a usable digital audio signal. The PDM bitstream is encoded from an analog signal through the process of 1-bit delta-sigma ($\Delta\Sigma$) modulation. This process uses a one-bit quantizer that outputs either a 1 or 0 depending on the amplitude of the analog signal. Due to the nature of real-world analog signals, a quantization error occurs, representing the difference between the 1 or 0 and the actual amplitude it represents. This error is negatively fed back in the $\Delta\Sigma$ process loop, influencing every subsequent quantization measurement and its error. This feedback mechanism averages out the quantization error, enhancing the accuracy of the PDM representation.

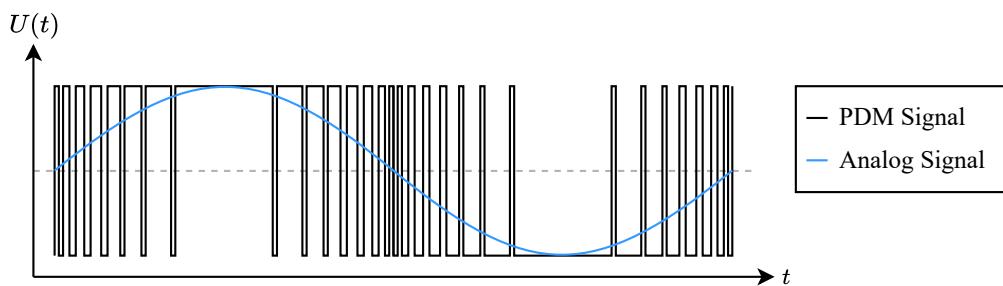


Figure 2.7: PDM Modulation Example

2.3 Time Division Multiplexing (TDM)

Time Division Multiplexing (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line. In the context of digital audio, TDM allows multiple audio streams to share a single communication line, with each stream getting a dedicated time slot. This technique is valuable in systems where multiple audio channels, such as in surround sound systems or multi-microphone arrays, need to be transmitted over a single cable. TDM's main advantage is its ability to efficiently handle multiple audio streams without the need for multiple physical connections. This makes it particularly useful in professional audio applications, broadcast systems, and complex audio setups. However, TDM systems can be more complex to implement and require precise synchronization to ensure that the timing of the different channels is maintained. Figure 2.8 shows an example of a TDM-16 system with 16 bits sample width.

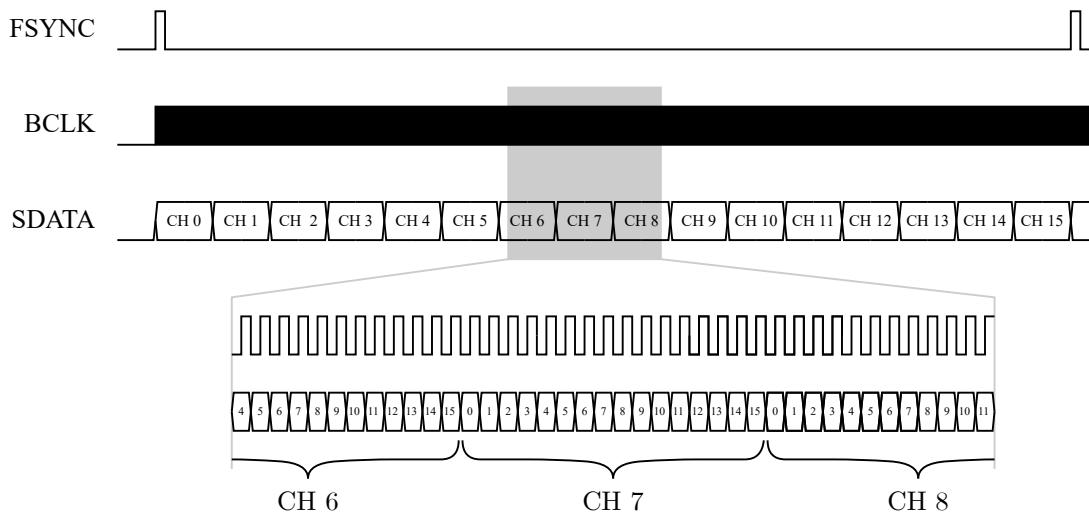


Figure 2.8: Example of TDM-16 with 16 bits sample width

Time Division Multiplexing (TDM) systems introduce tristate outputs, enabling multiple devices to share the same physical bus without interference. The TDM bus consists of three signals: *fsync*, *bclk* and *data*.

- **FSYNC (Frame Sync):** This signal marks the start of a frame in the TDM stream. It acts as a reference point for aligning the data across all channels. In a TDM-16 system, FSYNC indicates the beginning of a sequence of 16 channels (slots).
- **BCLK (Bit Clock):** This clock signal dictates the timing for data transmission. Each edge of the BCLK corresponds to a bit position in the data stream. For 16-bit samples, there are 16 BCLK pulses per channel, hence 256 BCLK pulses per frame.
- **Data:** The actual audio data is transmitted in a series of bits. In a 16-channel, 16-bit TDM system, each channel's data is represented by 16 bits (Left Justified).

2.4 GNSS Real-Time Kinematic (RTK)

The Global Navigation Satellite System (GNSS) Real-Time Kinematic (RTK) is an advanced technique used in geodesy and navigation that enhances the precision of position data derived from satellite-based positioning systems. RTK uses differential techniques to improve the accuracy of position information obtained from satellite systems like GPS, GLONASS, Galileo, or BeiDou. This method is particularly useful in applications requiring high precision, such as surveying, construction, and precision agriculture.

2.4.1 Custom Node as an Anchor

One method to utilize RTK for achieving enhanced position accuracy involves using a custom node as an anchor. This method requires setting up a fixed RTK base station at a known location. The base station calculates error values for satellite signals by comparing the expected signal (based on its known location) with the received signal. These error values, which account for various factors like atmospheric interference and satellite orbit errors, are then transmitted to a mobile RTK receiver. The mobile receiver uses these corrections to adjust its own satellite signal measurements, significantly improving its positional accuracy.

2.4.2 Using Data from a Public Provider

Alternatively, RTK corrections can be obtained from a public provider. In this approach, the user does not set up a custom anchor node but instead subscribes to a service that provides RTK correction data. These services operate a network of RTK base stations and compute correction factors for different regions. The correction data is typically transmitted via the internet directly to the user's RTK receiver. This method is convenient for users who do not have the resources to establish and maintain their own RTK base station. By applying these corrections, which include compensating for atmospheric drift and other errors, the user's receiver can achieve a similar level of enhanced positional accuracy as with a custom anchor node.

3

Sound Source Localization

3.1 Physical Background

This section introduces the sound propagation model from [11] which the following described theory is based on. The sound propagation of an ideal isotropic source follows

$$I = \frac{P}{4\pi r^2}, \quad (3.1)$$

where I is the measurement point, P is the source power and r is the distance between I and the source. In this formula the absorption of the air is neglected. If the sound intensity is measured at two points which are much closer to each other than they are from the source, the intensities are approximately $I_1 \approx I_2$.

The propagation speed of sound waves in air is

$$c = \sqrt{\frac{\aleph RT}{M}} \quad (3.2)$$

with \aleph as the heat capacity ration, R as the gas constant, T as the temperature and M as the molar mass. If c is known for a specific temperature T_0 the equation is equal to

$$c(T) = c_{T_0} \sqrt{1 + \frac{T}{T_0}}. \quad (3.3)$$

3.2 Sound Source Localization Methods

Sound Source Localization (SSL) is a well researched area with many applications. The basic system can be brought into two categories, time based or power based methods.

3.2.1 Power Based SSL

The idea of power based SSL is based of the known propagation properties of sound waves in air. However for this method to work some properties of the sound source have to be known. Additionally the sensors that measure the sound power levels have to be precise enough to be able to capture small differences in Sound power levels. To avoid such demanding hardware requirements this approach was not further studied in this thesis.

3.2.2 Time Based SSL

Another group of SSL is based on the time when the signal is picked up by the microphones. Given a source at a location $\mathbf{S} = (x_S, y_S)^T$ and N microphones at locations $\mathbf{M}_n = (x_n, y_n)^T$ the time it takes for a acoustic signal from the source to reach a microphone is

$$t_n = \frac{\|\mathbf{S} - \mathbf{M}_n\|}{c} = \frac{\sqrt{(x_S - x_n)^2 + (y_S - y_n)^2}}{c}. \quad (3.4)$$

So if t_n is known for a microphone, the location of the source can be limited to points on a circle around M_n with a radius of $t_n c$. Given three or more microphones, the intersection of these circles will show the location of the source. However in many cases this approach is not realistic since t_n is generally not known. It would require some sort of a synchronization between the microphones and the sources.

Near-Field

The next time property that could be used is the Time Differnece of Arrival (TDOA) between two microphones, which is defined as

$$t_{n,m} = t_m - t_n = \frac{\|\mathbf{S} - \mathbf{M}_m\| - \|\mathbf{S} - \mathbf{M}_n\|}{c}. \quad (3.5)$$

With this equation, \mathbf{S} can be interpreted as the set of points that lie on a hyperbola whose fixed points are the Microphones and the vertices are $c t_{n,m}$ m apart. To find the location of \mathbf{S} , a minimum of four microphones is now needed. This approach only works if we can assume that the curvature of the incoming sound waves at the microphones is big enough. Figure 3.1a shows such a arrangement, which is generally known as the near-field scenario. It is generally said, that the near-field assumption holds when the distance from the source to the array is not much greater than the size of the array. In Figure 3.1b, it can be seen how the resulting hyperbolas from the TDOA intersect at the point \mathbf{S} .

Far-Field

When the source is much further away than the size of the Array, as depicted in Figure 3.2, the curvature of the sound wave at the array is negligible. This is generally known as the far-field case where the sound waves are not modeled as spherical waves but rather as planar waves. Given such a planar wave, only the direction in which the source is placed can be determined with the TDOAs.

If the exact localization is required, multiple far field based systems could be used to triangulate the source position given the calculated direction from each system.

In this thesis the focus was set on far-field techniques considering that the goal is to detect drones which are usually several meters above ground. Considering this decision, only the direction in which a sound source is positioned can be determined with a single system.

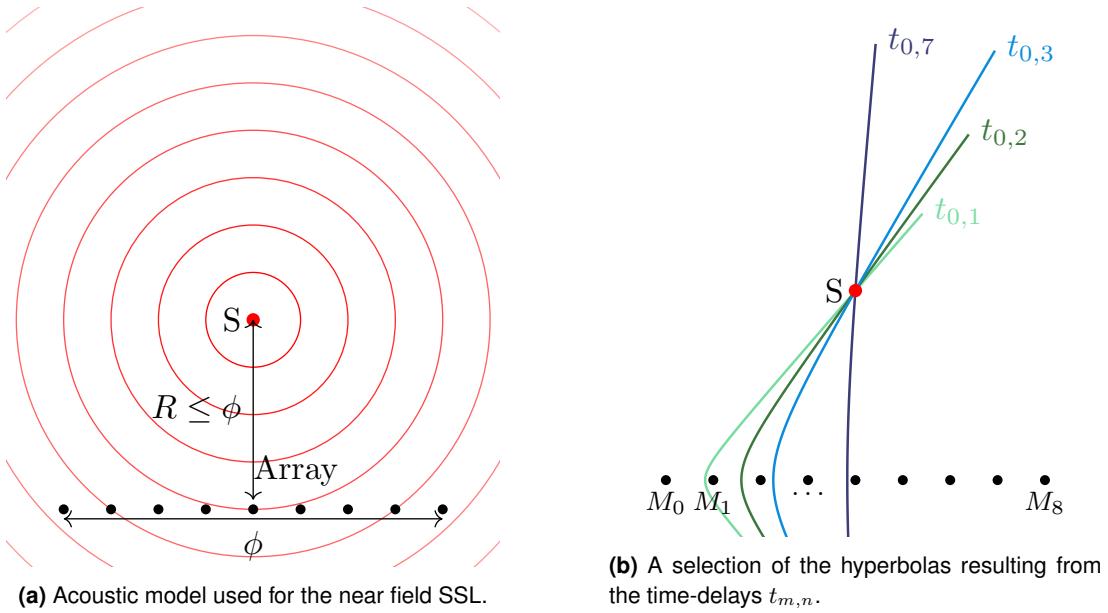


Figure 3.1: Near field SSL with a linear microphone Array. Since the microphones are distributed on a straight line a degree of freedom is lost and S mirrored at this line is also a possible solution.

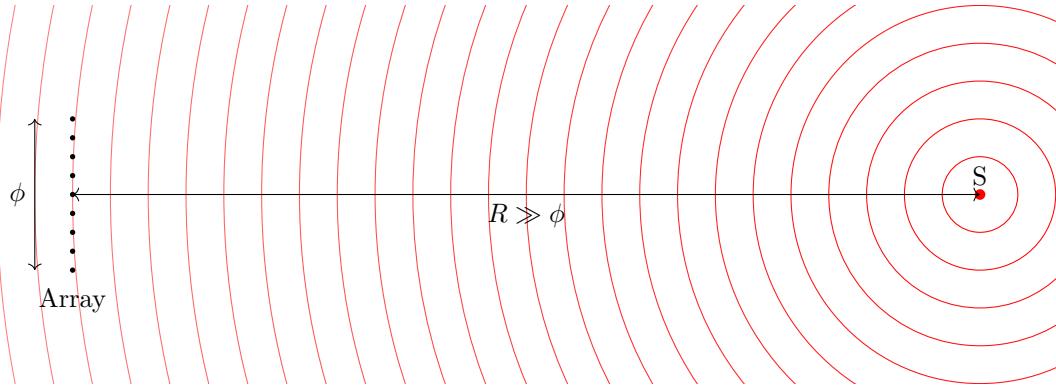


Figure 3.2: Far-Field case. The curvature of the sound waves from S at the array is almost zero.

3.2.3 Direction of Arrival Estimation

Lots of research has been made on how to estimate the Direction of Arrival (DOA) of a source with appropriate sensors. In [12] the authors built a Sound Compass using MEMS to find the directions of multiple sound sources. They also followed the approach to use multiple of their devices to find an exact position of a sound source. To estimate the DOA they used a Delay and Sum beamformer, which is also used in this thesis.

3.2.4 Beamforming

To better show the fundamental principles of beamforming, an example is used where the goal is not to estimate a DOA but to send a sound wave in a desired direction using loudspeakers. Figure 3.3 illustrates such a case in \mathbb{R}^2 , where a linear array of N sound sources beam a sine wave $x(t) = \sin(\omega_0 t)$ in the direction of φ . This is achieved by adding a phase shift to the sine wave at each microphone, resulting in $x_n(t) = \sin(\omega_0 t - \omega_0 t_n)$. If the phase shifts are properly selected the points where the

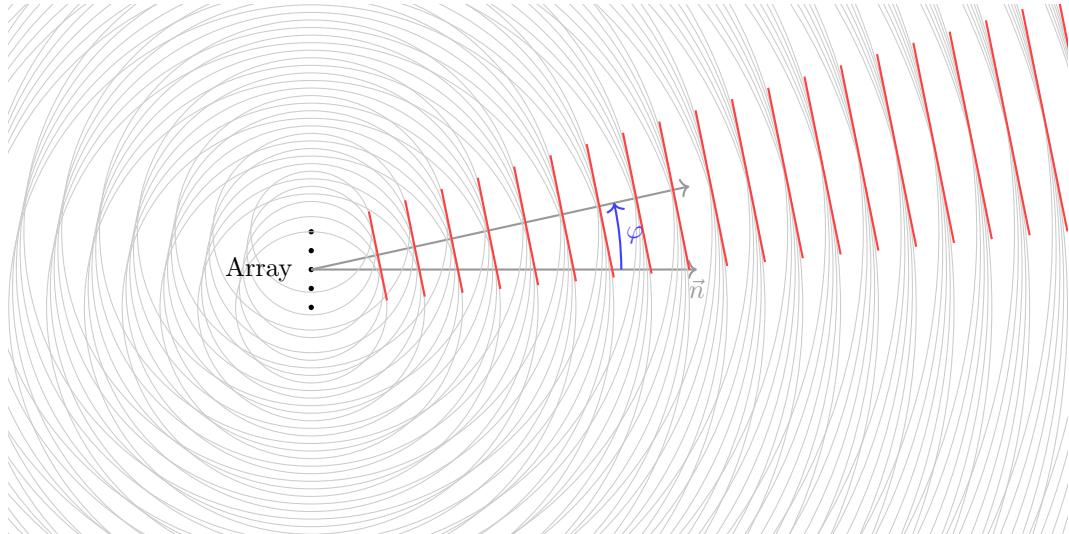


Figure 3.3: Phase differences between the array elements create positive interference at the red lines. The gray circles correspond to the waves' local maxima

waves positively interfere to form a beam. Using this example with a narrow band signal, the output at each source can be written as

$$x_n(t) = \Re(\sin(\omega_0 t) e^{-j\omega t_n}) \quad (3.6)$$

and more generalized as

$$\mathbf{X}(t, \varphi) = \Re \left(\sin(\omega_0 t) \underbrace{\begin{pmatrix} e^{-j\omega_0 t_0(\varphi)} \\ e^{-j\omega_0 t_1(\varphi)} \\ \vdots \\ e^{-j\omega_0 t_{N-1}(\varphi)} \end{pmatrix}}_{\mathbf{W}(\varphi)} \right). \quad (3.7)$$

The vector $\mathbf{W}(\varphi)$ is commonly known as the steering vector. It can be seen that the steering vector is dependent on ω_0 and will therefore only have the desired effect on signals with frequencies close to ω_0 . The idea will be expanded to wide band signals later in this chapter.

If the goal is to estimate the direction of a sound signal with microphones $x_n(t)$ is redefined as the signal measured by the n th microphone. Now (3.7) is rewritten to

$$\mathbf{Y}(t, \varphi) = \underbrace{\begin{pmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{N-1}(t) \end{pmatrix}}_{\mathbf{X}(t)} \odot \underbrace{\begin{pmatrix} e^{-j\omega_0 t_0(\varphi)} \\ e^{-j\omega_0 t_1(\varphi)} \\ \vdots \\ e^{-j\omega_0 t_{N-1}(\varphi)} \end{pmatrix}}_{\mathbf{W}(\varphi)} \quad (3.8)$$

where depending on the beamforming method used either $\mathbf{Y}(t, \varphi)$ is further processed, or the method directly uses $\mathbf{X}(t)$ for the DOA estimation.

Steering Vector

The calculation of the Steering Vector is dependent on different factors. The two main factors can be broken down to the geometry of the microphone array and the direction in which the source lies. Again the formulas are derived for a circular array in \mathbb{R}^2 and are then expanded into \mathbb{R}^3 . A circular array where all the microphones are placed on a circle is used for the derivation, the theory can however be expanded to any array geometry in \mathbb{R}^2 .

Figure 3.4 shows a circular array with five equally spaced microphones M_n . The goal of the steering vector is to delay the measured signal at each microphone, so that they have the same phase and therefore positively interfere. To calculate these delays a reference point must first be defined. Here it is the center of the circle. With the far field model, the sound pressure level is equal along lines perpendicular to its propagation direction. Therefore the magenta line is the reference line and each measured signal from the microphones must be phase shifted with

$$\omega_0 t_n = \omega_0 \frac{\mathbf{d}_n(\varphi_s) \cdot \hat{\mathbf{d}}_s}{c}. \quad (3.9)$$

Simplifying this equation with

$$\mathbf{d}_n(\varphi_s) \cdot \hat{\mathbf{d}}_s = r_{m_n} \cos(\varphi_s - \varphi_{m_n}) \quad (3.10)$$

results in

$$t_n(\varphi_s) = r_{m_n} \cos(\varphi_s - \varphi_{m_n}) \frac{1}{c}. \quad (3.11)$$

This formula can be used for any microphone with polar coordinates (φ_{m_n}, r_{m_n}) and can therefore be used for any array configuration described in polar coordinates.

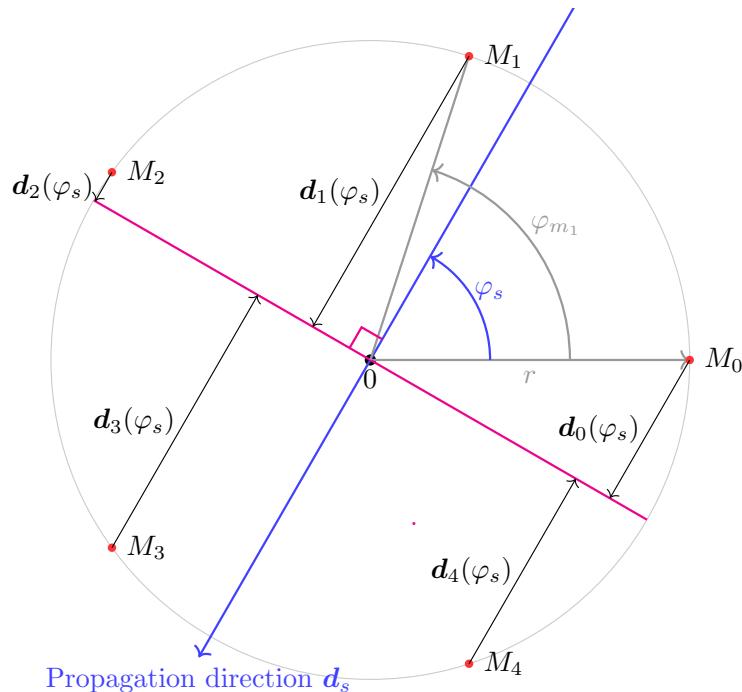


Figure 3.4: Distances between each microphone and the reference line in a circular microphone array.

To expand this idea into \mathbb{R}^3 the spherical coordinate system is used. This adds the angle θ as the inclination. When $\theta = \pi/2$ it is the same situation as with the polar coordinate system. As θ approaches 0 the speed of the wavefront projected onto the surface of the array increases, until the sound wave reaches every microphone simultaneously when $\theta = 0$. This expands (3.11) to

$$t_n(\varphi_s, \theta_s) = r_{m_n} \cos(\varphi_s - \varphi_{m_n}) \frac{\sin(\theta_s)}{c}. \quad (3.12)$$

To simplify the formulas, $t_n(\varphi, \theta)$ is from now on written as t_n .

Delay and Sum Beamformer

The basic beamforming is the delay and sum beamforming

$$y(t, \varphi, \theta) = \underbrace{\begin{pmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_{N-1}(t) \end{pmatrix}}_{\mathbf{X}(t)} \cdot \underbrace{\begin{pmatrix} e^{-j\omega_0 t_0(\varphi, \theta)} \\ e^{-j\omega_0 t_1(\varphi, \theta)} \\ \vdots \\ e^{-j\omega_0 t_{N-1}(\varphi, \theta)} \end{pmatrix}}_{\mathbf{W}(\varphi, \theta)}. \quad (3.13)$$

Notice how instead of the Hadamard product, the scalar product is now used resulting in $y(t, \varphi, \theta)$ as the sum of the delayed signals. If the angles of the steering vector point towards a source its signal will be amplified.

Beamforming can be visualized with the so called beam pattern. It shows how much a sound of a source at a specific location is amplified depending on the steering direction. By using the delays calculated for the steering vector $\mathbf{X}(t)$ can be calculated to simulate a source in a specific direction

$$\mathbf{X}(t, \varphi_s, \theta_s) = e^{j\omega_0 t} \begin{pmatrix} e^{j\omega_0 t_0(\varphi_s, \theta_s)} \\ e^{j\omega_0 t_1(\varphi_s, \theta_s)} \\ \vdots \\ e^{j\omega_0 t_{N-1}(\varphi_s, \theta_s)} \end{pmatrix} \quad (3.14)$$

The response of an array using delay and sum beamforming can now be calculated with

$$G(\varphi, \theta) = \frac{|\mathbf{X}(0, \varphi_s, \theta_s) \cdot \mathbf{W}(\varphi, \theta)|}{N}. \quad (3.15)$$

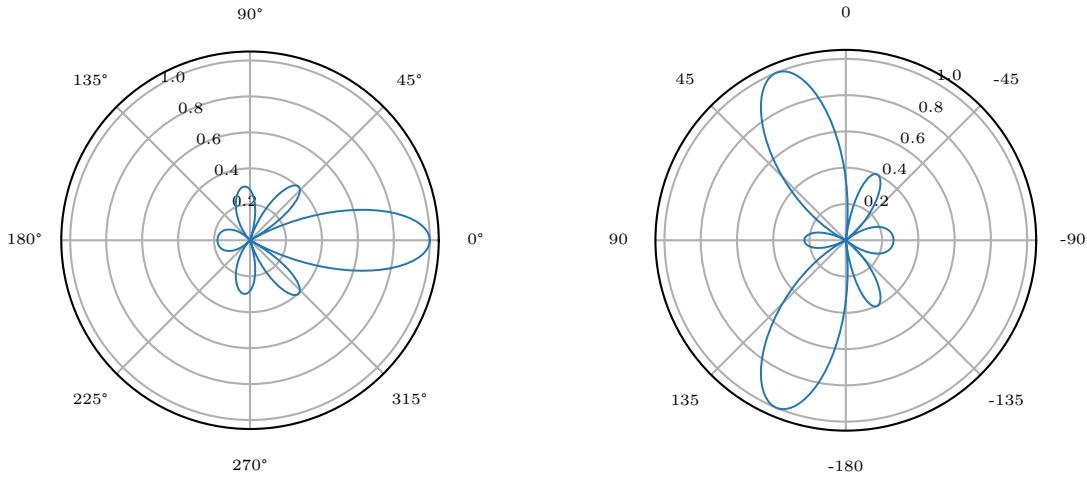
The response is normalized with the amount of microphones to ensure a maximum gain of 1. In Figure 3.5 two such responses with can be seen. The main lobe of $G(\varphi, \theta)$ clearly shows in the direction of the source however there are also side lobes pointing to directions without sources.

Wide Band Delay and Sum Beamforming

Until now the formulas are only useful for narrow band signals. This theory is now expanded to be applicable for wide band signals.

The narrow band delay and sum beamforming has the property, that it behaves the same in the frequency domain

$$e^{-j\omega_0 t_n} x_n(t) = \mathcal{F}^{-1}(e^{-j\omega_0 t_n} \mathcal{F}(x_n(t))) \quad (3.16)$$



(a) $G(\varphi, \theta, \omega)$ with $\theta = 90^\circ$ for $\omega_0 = 2\pi 1000 \text{ rad/s}$. The source is placed in the direction of $(\varphi = 0^\circ, \theta = 90^\circ)$

(b) $G(\varphi, \theta, \omega)$ with $\varphi = 0^\circ$ for $\omega_0 = 2\pi 1000 \text{ rad/s}$. ($\varphi = 0^\circ, \theta = -45^\circ$) is the same as $(\varphi = 180^\circ, \theta = 45^\circ)$. The mirror along the horizontal axis comes from the flat geometry of the array which makes it impossible to distinguish if a sound comes from θ or $180^\circ + \theta$.

Figure 3.5: Beam patterns of delay and sum beamforming using a circular array with $N = 16$ microphones and a radius $r = 0.25 \text{ m}$. Note that the gains are applied on the amplitudes, so the power levels are multiplied by $G(\varphi, \theta, \omega)^2$

since it is just a multiplication with a scalar. The summation could also be done in the frequency domain due to the linearity of the Fourier transform. The frequency domain has the advantage that an IIR Filter can be constructed, to make a different phase shift for every frequency

$$Y_n(t) = \mathcal{F}^{-1}(e^{-j\omega t_n} X_n(\omega)), \quad (3.17)$$

where $X_n(\omega) = \mathcal{F}(x_n(t))$. By using a discrete fourier transform the beamformer can be written in matrix form

$$\mathbf{Y}(\varphi, \theta) = \begin{pmatrix} X_0(\omega_l) & \dots & X_0(\omega_h) \\ \vdots & \ddots & \vdots \\ X_{N-1}(\omega_l) & \dots & X_{N-1}(\omega_h) \end{pmatrix} \odot \begin{pmatrix} e^{-j\omega_l t_0} & \dots & e^{-j\omega_h t_0} \\ \vdots & \ddots & \vdots \\ e^{-j\omega_l t_{N-1}} & \dots & e^{-j\omega_h t_{N-1}} \end{pmatrix}, \quad (3.18)$$

with a total of M frequency points of interest between ω_l and ω_h . The power of the signal is now calculated with

$$P(\varphi, \theta) = \sum_{m=0}^{M-1} \left| \sum_{n=0}^{N-1} \mathbf{Y}_{n,m} \right|^2 \quad (3.19)$$

using Parsevals' Theorem. This beamformer does not need to be implemented in the frequency domain. Delaying each signal in the time domain is equivalent.

Spatial Sampling

Beamforming can be seen as spatial sampling, where the microphones are the sampling points. So some of the concepts used in digital sampling such as the sampling theorem and the uncertainty principle can also occur in some way in beamforming. Figure 3.6 shows the beam pattern for different frequencies. The array used in Figure 3.6a is the

same array as in Figure 3.5 whereas Figure 3.6a uses a circular array with $r = 0.5$ m and also 16 microphones. It can be seen that at around 1500 Hz more side lobes with a high amplitude start to appear. With this array geometry the smallest distance between two microphones is ≈ 0.098 m, which is the wavelength of a sound wave with a frequency of approximately 3500 Hz. So the effect seen in Figure 3.5 is a kind of aliasing. This results from ambiguity in the phases of the steering vector. Ideally the phase shift between two neighboring microphones is unique for each angle of the steering vector. However when they are more than $\lambda/2$ m apart, phase shifts start to repeat for different steering directions.

Another effect seen in Figure 3.6a is the increase of the main lobe width when the frequency approaches 0 Hz. This is consequence of a too small array. In the used array the maximal distance between two microphones is 0.5 m. For lower frequencies the phase difference between the first microphone and the last microphone in the array is too small to get destructive interferences when the steering vector is not focused on the source. As expected the bigger array from Figure 3.6b has a smaller main lobe in the lower frequencies.

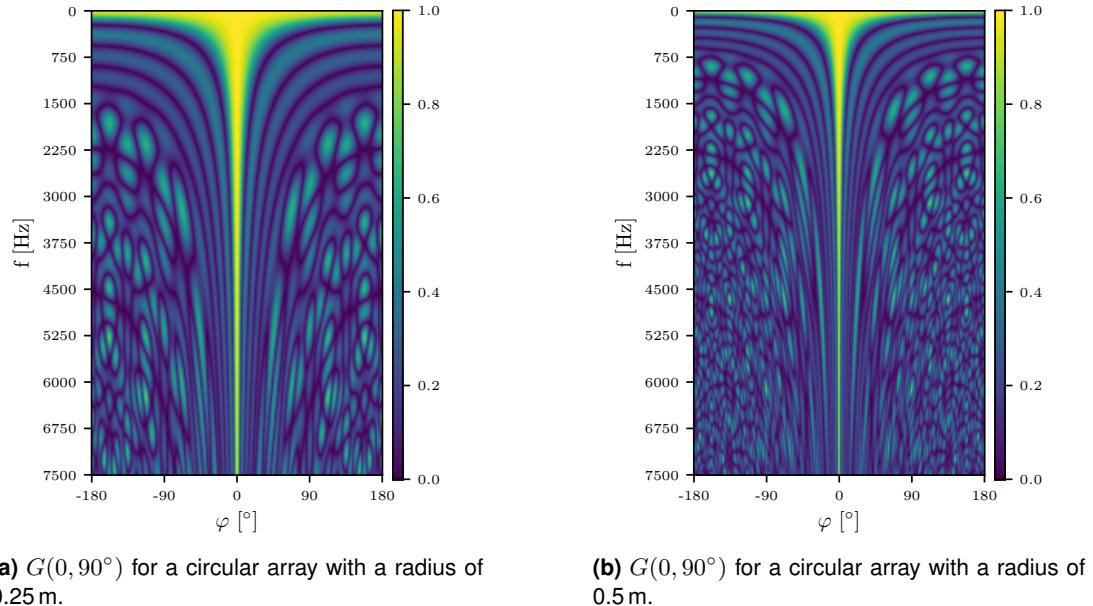


Figure 3.6: Delay and Sum Beamformer response for two circular arrays with $N = 16$ microphones and a radiiuses of 0.25 m and 0.5 m.

3.3 Tracking

By now the DOA of a sound source can be estimated. But since drones expected to be moving, a source once it has been detected, needs to be tracked.

In [7] the tracking is done with by updating the steering vector with an adaptive filter. They however do not go into detail on how the direction is then retrieved from the steering vector. One possibility may be by looking at the $G(\varphi, \theta)$ of the steering vector. But what makes this definitely not used in this thesis, is that its based on a narrow band beamformer. Developing a beamforming based tracker for a wide band beamformer was outside of the scope of this thesis.

Another popular object tracker independent of beamforming is a Kalman filter based tracker. It takes a measured position as an input and calculates the estimated position with a Kalman filter. The basic model for the Kalman filter is, that the source moves with a constant velocity leading to

$$\mathbf{x}(n) = \begin{pmatrix} x_n \\ y_n \\ z_n \\ \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{pmatrix}, \mathbf{A}(n) = \begin{pmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{C}(n) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

with $\mathbf{x}(n)$ as the state vector, $\mathbf{A}(n)$ as the state transition matrix and $\mathbf{C}(n)$ as the observation matrix. The velocity in the state vector also enables blind prediction.

To enable multi object tracking, another processing step is added. This idea resulted from a conversation in December 2023 with Hannes Badertscher. First the new measured objects are assigned to the already existing trackers. To make this assignment somehow optimal, the Hungarian algorithm [1] with the measurement and blind prediction of each Kalman filter as input, is used. If the distance between the prediction and measured object position of a pair is close enough the Kalman filter is run with the measured position as input. However if they are far apart, the measured object is treated as a new object and is assigned to a new Kalman filter instance.

Note that the tracker is in the cartesian coordinate system. The transformation from a DOA to a cartesian point is covered in Section 6.5

3.4 Simulator

In order to experiment with different microphone arrangements and algorithms a simulator in Python was developed. Since the simulator was not the main focus of this Thesis, its functionality was kept simple. The simulator lets you place acoustic sources and microphones in a \mathbb{R}^3 space and calculates the measured signals at each microphone.

3.4.1 Simulation Model

For sake of simplicity the sources were modeled as omnidirectional point-sources and the microphones are omnidirectional as well. The Sound Pressure Level of a source is defined at one meter distance from their position and decreases squarelly with the distance. The perceived sound at any point \mathbf{P} can now be described as

$$y(\mathbf{P}, t) = \sum_I \frac{x_i(t - \|\mathbf{P} - \mathbf{S}_i\|/c)}{\|\mathbf{P} - \mathbf{S}_i\|}. \quad (3.20)$$

Where \mathbf{S}_i is the position of the nth Source and $x_i(t)$ is its output sound. Since the simulation is done numerically and with pre recorded audio files with a given sampling rate f_s (3.20) has to be discretized. This is done by discretize t to

$$t_n = t_0 + n/f_s, \quad n \in \mathbb{N}.$$

The time delay also needs to be discretized to be an integer multiple of $1/f_s$. By rounding it to the nearest number that fulfills this condition results in a new delay d_{P,S_i} . Now the delayed signal is $x_i(t_n - d_{P,S_i})$. To achieve sub sample delays, this signal is then filtered with a Fractional Delay FIR Filter [8] with a delay of $f_s \|\mathbf{P} - \mathbf{S}_i\|/c - d_{P,S_i}$.

Reflections

The simulator also allows the reflective surfaces to be placed into the room. These surfaces are defined with a base vector and two directional vectors and have a infinite size. Additionally a dampening factor can be defined which defines how much a reflecting sound signal is damped when getting reflected. By this stage the simulator can only handle single reflective path. An already reflected sound signal can not be reflected by a second surface.

3.4.2 Output

The simulator saves WAV files for each microphone, with the audio signal from each source as a single chanel. It can also be installed as a python package with its GitHub link. In this case, the simulator can also return the recorded signals as an Numpy array.

4

Acquisition-System Design

4.1 Overview

The first step after understanding the basic theory behind microphone arrays and beamforming is to apply this knowledge in a practical setting. The primary goal is to record and analyze real-world audio data from a variety of microphones. This involves the evaluation of different microphone types and array configurations. Understanding these differences is critical for further development and refinement of beamforming algorithms.

The approach extended to the development of a specialized hardware system, necessary for recording multiple audio channels simultaneously. This capability was not found in existing hardware solutions, leading to the design and development of a new system, supporting up to 32 microphones. The channel limit was chosen due to practical constraints while keeping the system complexity manageable.

Once the audio data is captured, the next phase involves applying algorithms to analyze these recordings. Applying these algorithms to the captured audio data facilitates a detailed comparison between real-world microphone performance and theoretical simulation results. This comparison is substantial for understanding the differences between practical microphone use and simulated scenarios, thus being crucial for further algorithm development and refinement.

In addition to its primary purpose, this system enables possibilities for various other applications where recording a large number of microphones is needed. Its ability to handle multiple channels simultaneously and processing them in real-time makes it a versatile tool for various use cases.

The subsequent sections describe the microphone evaluation and the development process, including hardware and firmware design of the audio acquisition system.

4.2 Key Requirements

The goal of the acquisition system is to provide a flexible microphone recording infrastructure to easily acquire audio signals from multiple microphones.

The following key requirements have been set:

- Simultaneous recording of up to 32 microphone channels
- High-quality audio recording with 16-bit resolution and 44.1 kHz sampling rate (CD-Quality)
- Recording to a removable SD-Card in lossless WAV format
- Real-time monitoring of individual microphone channels
- Easy to use UI for configuration and operation
- Compact and portable design to enable mobile use

4.3 Key Decisions

The following section describes the key decisions made during the development of the acquisition system.

1. **MCU Selection:** As a main Microcontroller Unit (MCU) the **Teensy 4.1** was chosen due to its ability providing two TDM-16 audio interfaces, enabling support for up to 32 audio channels. Its computational performance and extensive software support in audio applications were key factors in this decision. Additionally, the **Teensy 4.1** includes a fast SDIO SD-Card interface with a built-in card holder, ideal for this application.
2. **Microphones:** Preference was given to PDM microphones due to their wide availability and suitability for use with longer cables, in comparison to other microphone types mentioned in section 2.1.
3. **Power Source:** The system is powered via a single USB cable to ensure portability and ease of use in various settings, adhering to the requirement for a compact and mobile design.
4. **Ethernet Port:** An RJ45 ethernet port was added for future development opportunities, such as streaming audio data over ethernet.
5. **Touch Display:** A touch TFT display was integrated to offer an easy-to-use UI, facilitating efficient system configuration, operation, and real-time monitoring.
6. **RGB LEDs:** RGB LEDs were employed for visual feedback on the audio levels of each microphone channel.
7. **Headphone Jack:** The addition of a headphone jack allows for real-time auditory monitoring of individual microphone channels, essential for troubleshooting.
8. **Real-Time Clock (RTC):** An RTC was integrated to tag each recording with the current time and date, simplifying the comparison of measurement results in post.

4.4 Microphone Evaluation

Although a variety of microphone technologies exists, such as condenser, dynamic, and electret, the focus has been set on MEMS microphones due to several compelling reasons. Primarily, their widespread availability and ease of manufacturing, allowing PCB manufacturers to assemble them, make MEMS microphones a practical choice. Their compact size is advantageous in space-constrained applications, while the integrated analog frontend simplifies audio system design. Notably, MEMS microphones deliver excellent audio quality and wide bandwidth, essential for high quality sound reproduction. Moreover, their cost-effectiveness makes them suitable for microphone arrays with a large number of channels.

4.4.1 Microphone Types

In the evaluation process, four different MEMS microphones were selected. Two of them are top-ported, while the other two are bottom-ported. All four microphone types are available in large quantities at *JLCPCB*, a popular PCB manufacturer. Comparing the datasheets of the different microphones, only small differences in there specifications were found. Consequently, it was all the more intriguing to compare these microphones and to determine effective differences in audio quality and Signal-to-Noise Ratio (SNR).

Microphone Type	Port	Manufacturer	Similar Types
MP34DT05TR-A	Top	ST Microelectronics	-
GMA4030H11-F26	Top	INGHAI	<i>Knowles SPK0415HM4H-B-7</i>
GMA3526H10-B26	Bottom	INGHAI	<i>Knowles SPH0641LU4H-1</i>
SD18OB261-060	Bottom	Goertek	-

Table 4.1: Evaluated MEMS Microphone Types

4.4.2 Microphone Breakout Boards

To test the four evaluated microphone types, a two-layer carrier PCB was designed and manufactured. A panelized design was chosen to simplify the manufacturing process and reduce assembly costs. One panel consists of 8 breakout boards for each microphone type, resulting in a total of 32 breakout boards per panel. In total, 5 panels were manufactured, which leads to 40 microphone breakout boards per type. Each breakout board has a size of 14.0 mm x 22.0 mm and is separated by a V-groove, a common technique used in PCB manufacturing. This allows the individual breakout boards to be easily separated from each other.

Every microphone board is equipped with a slide switch for the channel selection, allowing to toggle between the left and right PDM channel. Additionally, a dual-color LED is integrated, serving as a power indicator. It lights up red if the right channel is selected or white if the left channel is selected. To enhance the ease of mounting, each board includes a threaded surface-mounted standoff nut for an M3 screw, enabling straightforward attachment to a frame.

For connectivity, a standard SH 4-Pin JST¹ connector with a 1 mm pin pitch was used. This connector type is commonly found in Adafruit *STEMMA QT / Qwiic JST I²C*

¹JST refers to Japan Solderless Terminal, a leading manufacturer of a diverse range of connectors, including wire-to-board, board-to-board, and wire-to-wire types.



Figure 4.1: Front View of the Microphone Breakout-Boards

accessories boards. By using this specific type, a wide range of pre-assembled cables with different lengths are available, simplifying the connection between the breakout boards and the acquisition system. The microphone carrier PCB follows the same pinout as Adafruit's own PDM-Microphone breakout board (Adafruit part number: 4346), facilitating compatibility. The connector also supplies power (3.3V) to the microphones, allowing multiple units to be connected with a single cable. For the testing setup, cables measuring 40 cm in length were used (Adafruit part number: 5385).

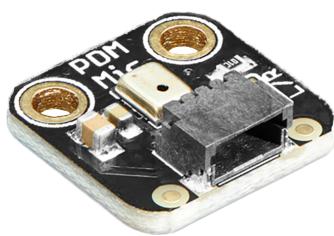


Figure 4.2: Adafruit PDM Microphone (4364)



Figure 4.3: STEMMA QT / Qwiic Cable (5385)

The table below outlines the pinout used for these microphone breakout boards:

Pin Number	Function
1	GND
2	3V3
3	PDM Data (provided by the host)
4	PDM Clock (microphone output signal)

Table 4.2: Microphone Breakout Board Pinout

4.5 Hardware Design

The hardware of the acquisition system is centralized around a 4-Layer PCB with a size of 111.0 mm x 136.0 mm. The connectors for the microphone breakout boards are located on the left and right side of the PCB (16 channels on each side). Each microphone channel is equipped with a pair of RGB LEDs. The LED near the connector visually represents the audio level for its respective channel. Another LED, situated on the white silk screen marking, indicates the active routing of the channel to the monitor headphones output. A *Monitor Selection* push button enables navigation through the microphone channels. Each button press cycles through the channel numbers, directing the selected channel to the headphones output. Additionally, a potentiometer is available for adjusting the volume of the headphones output. On the lower segment of the PCB, a 1.44" TFT touch display is embedded, serving as an interactive interface for controlling the acquisition system. Adjacent to this display is the *Record* button, which is used to start and stop the recording process. The upper part of the device houses a USB Type-C and an RJ45 connector, providing data connectivity to external devices. Moreover, each corner of the PCB is equipped with M3 surface-mounted standoff nuts, allowing for easy mounting of the acquisition system onto a microphone array frame.

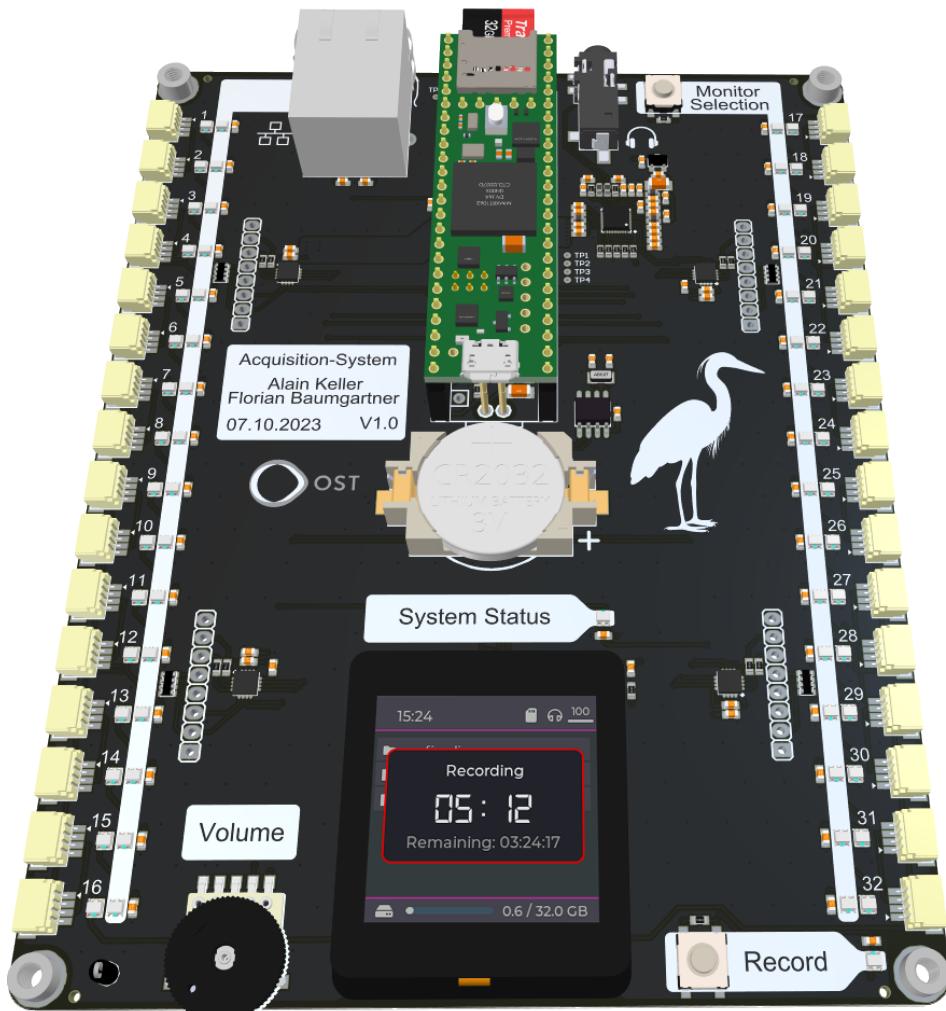


Figure 4.4: Front View of the Acquisition System

4.5.1 Block Diagram

In figure 4.5 the system block diagram is shown.

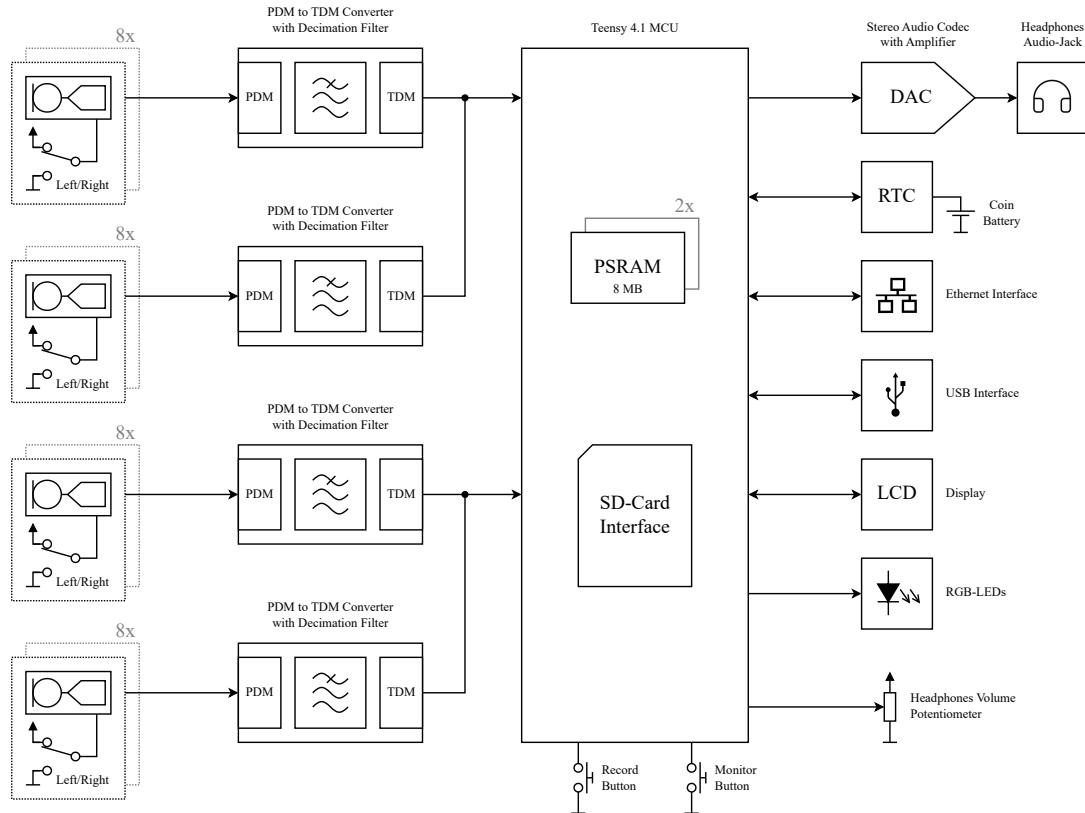


Figure 4.5: System Block Diagram of Acquisition-System

4.5.2 Microcontroller Unit (MCU)

As a main MCU the *Teensy 4.1* was chosen due to its powerful ARM Cortex-M7 processor, running at 600 MHz. The *Teensy 4.1* is a small form-factor development board, manufactured by *PJRC*. It provides a build-in programming interface, which allows to program the MCU directly via USB without the need for an external programmer. Compared to the smaller formfactor *Teensy 4.0*, the *Teensy 4.1* includes a built-in SD-Card holder and additional flash memory of 8 MB. Although the IMXRT1060 SoC provides 1 MB of RAM, for this application, two additional 8 MB PSRAM memory chips are required. They can conveniently be mounted on the backside of the *Teensy 4.1* board as shown in figure 4.7.

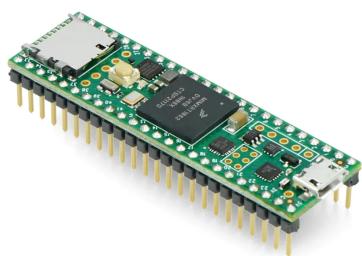


Figure 4.6: Teensy 4.1

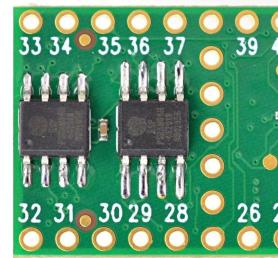


Figure 4.7: Mounting of external PSRAM

4.5.3 Audio Input

As mentioned in section 2.1.2, the use of PDM microphones requires a digital decimation filter to convert their bitstream into a usable audio format. The Teensy 4.1, despite having built-in hardware PDM decimation, is limited to two channels, requiring external decimation for systems with more audio channels.

To overcome this limitation, the *ADAU7118* from Analog Devices is employed. This Integrated Circuit (IC) is specialized in handling PDM signals, offering four PDM inputs and two clock outputs, which collectively can drive up to eight microphones in a multiplexed configuration. It outputs audio data in the TDM protocol. TDM-16, being capable of handling up to 16 channels, allows two *ADAU7118* ICs to share the same physical bus. Hence, individual configuration is necessary to ensure that each converter accesses the correct TDM slots (lower or upper 8 channels). In total, four *ADAU7118* ICs are used, providing 32 microphone channels.

The *ADAU7118* is also equipped with an I²C interface, enabling the configuration of various parameters such as the decimation ratio, TDM bus settings, and signal drive impedances.

4.5.4 Headphones Output

The addition of a headphones output to the acquisition system introduces the capability for live monitoring of microphone inputs. This feature allows a user to connect standard headphones with a 3.5 mm jack directly to the device, enabling real-time audio feedback.

The system employs the *WM8904* audio CODEC as its Digital-to-Analog Converter (DAC). This choice is strategic, as the *WM8904* not only supports digital audio input streams like I²S and TDM, but also includes a built-in headphones amplifier, making it particularly suitable for this application. Additionally, its I²C interface enables configuration adjustments, including device setup and volume control.

However, it is important to note that, as of this writing, this headphones output feature has not been implemented in the firmware. Consequently, the hardware designed to support this functionality has not yet undergone testing.

4.6 Firmware Design

The firmware is written in C++ and is based on the Arduino framework that has been adopted to the Teensy microcontroller environment. As an IDE, the VS Code extension *PlatfromIO* was used, as it provides powerful development tools and a great integration of the Arduino framework.

The firmware is divided into modules running in individual threads, facilitated by *TeensyThreads* on the Teensy 4.1 microcontroller. This lightweight multitasking library allows for concurrent execution of multiple threads, optimizing system performance and resource utilization. *TeensyThreads* has a minimal memory footprint and is optimized for efficient CPU usage, making it ideal for embedded systems. In total, 6 threads are running in parallel, as shown in table 4.3.

Thread	Purpose
Console Interface	Handles USB virtual COM-Port
Console Streaming	Handles queuing of messages
AudioUtils	Audio Processing
HMI	LED Control & RTC
Application	Main Application Logic
Main	Main Thread (Background)

Table 4.3: Overview of all Threads and their Purpose

4.6.1 Audio Recording

The multichannel audio recording is achieved by a dual-context approach, focusing on capturing audio data and storing it on the SD-Card. While the capturing process is very time-critical and therefore runs at high priority, storing data on the SD-Card can be done in the background.

Audio Processing Pipeline

The audio data is processed in blocks, where each block consists of 128 samples from each of the 32 microphone channels. Upon capturing a new block, an Interrupt Request (IRQ) service routine is called by the Teensy Audio Library. A critical step in processing is the interleaving of audio samples, as this is necessary to comply with the WAV file format. This process rearranges the data from individual channels into a sequential stream, as visualized in Figure 4.8.

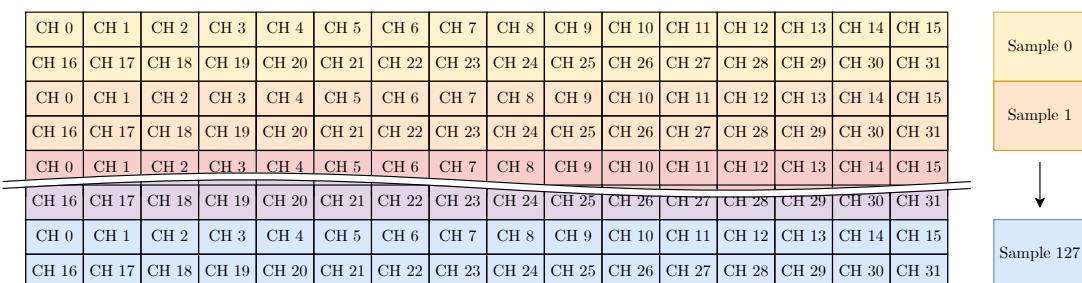


Figure 4.8: Interleaving of Audio Samples

SD-Card File Handling

The interleaved audio data is then stored in a 12 MB circular buffer located in the external PSRAM. A background event handler regularly checks the buffer for new data, which, upon availability, is read and subsequently written to the current audio file on the SD-Card. Each writing operation also involves updating the WAV file header, a necessary step to maintain a valid file structure.

Given the complexities of SD-Card operations, including variable access times and internal cache structures, the system is designed to verify each write operation. In the event of incomplete data transfer, the process is reiterated until all data chunks are successfully written to the file. Figure 4.9 shows a flow diagram of the audio recording process.

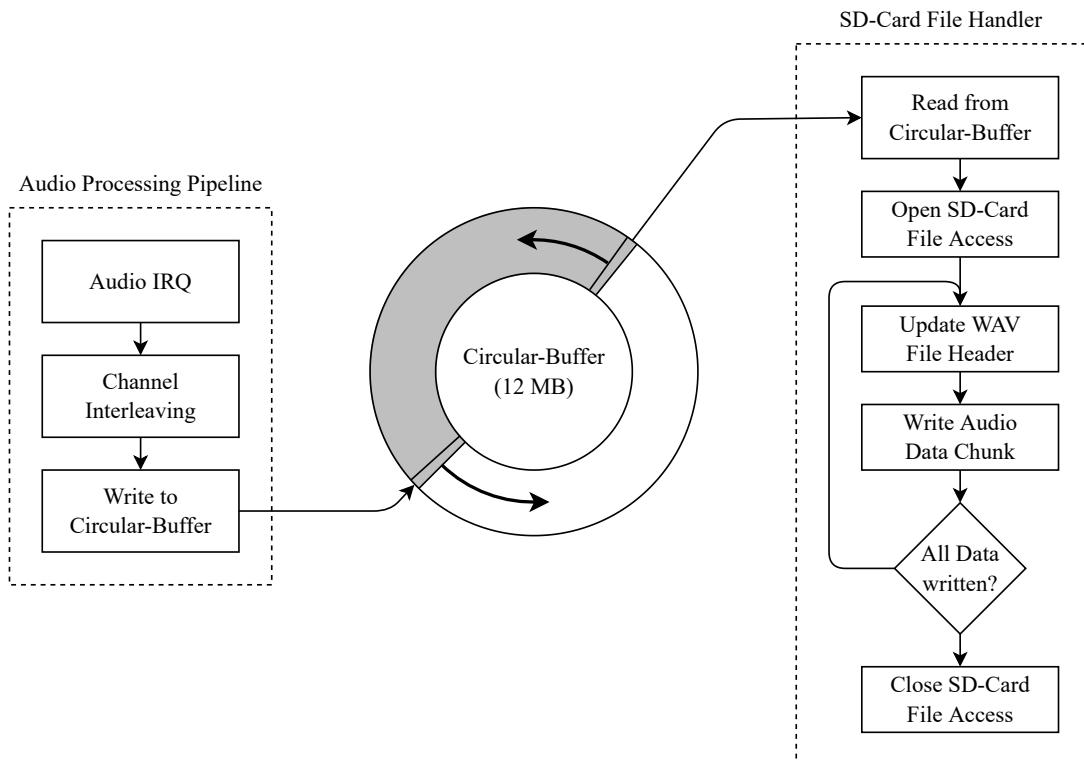


Figure 4.9: Flow Diagram of the Audio Recording Process

File Naming Convention

The convention for naming audio files is as follows:

`YYMMDD_HHMMSS_<CH_COUNT>_<CH_CONF>.wav`

This format incorporates the date and time of the recording, along with the channel count and configuration, thereby a straightforward file identification and organization is possible. The channel count is represented in decimal format, while the channel configuration is a 32-bit hexadecimal number, where each bit corresponds to a channel. The Least Significant Bit (LSB) represents channel 1, while the Most Significant Bit (MSB) represents channel 32. If the bit is set to 1, the channel is active and therefore included in the recording. The following file name example illustrates a 16 channel recording with channels 8 - 24 active: `240118_122417_16_00FFFF00.wav`

4.6.2 Graphical User Interface (GUI)

The GUI provides a user-friendly interface for configuring the device and monitoring its status. It is based on the LVGL framework, which is described in detail in the next section.

Light and Versatile Embedded Graphics Library (LVGL)

Light and Versatile Graphics Library (LVGL) is a free and open-source graphics library, primarily used for creating embedded GUIs. It's designed to be lightweight, consuming minimal memory and processing power, which is essential in embedded systems where resources are limited.

The decision to use LVGL in conjunction with the *NXP GuiGuider*, a graphical design tool, enables a rich set of features and enables rapid development. *GuiGuider* provides a user-friendly interface for designing GUIs, significantly simplifying the process of creating complex, visually appealing interfaces for embedded systems. It provides a code generator, which generates the necessary C-Code to translate the graphic design into LVGL API calls.

4.6.3 GUI Pages

The GUI is minimalist designed and straight forward to use. The navigation between the main pages is done by swiping left or right on the touchscreen. Next, the individual pages are described in detail.



Figure 4.10: Splash Screen

Splash Screen

When the device is powered on, the splash screen is displayed until the boot process is finished. On average this takes about 5 seconds.

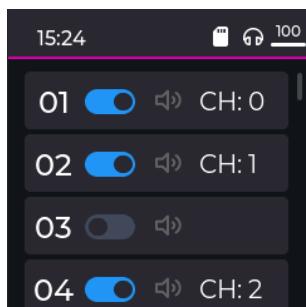


Figure 4.11: Channel Settings

Channel Settings

After the boot process is finished, the channel settings page is displayed. In the header bar located at the top of the page, the current time, USB interface status, SD-Card status and headphones volume are displayed. A list of all 32 microphone inputs is shown in the center of the page. Each input channel can be enabled or disabled by clicking on the corresponding switch. When an input is enabled, its associated channel number on the WAV file is displayed. A speaker symbol shows if the channel is currently routed to the headphones monitor output (green means active, grey means inactive).

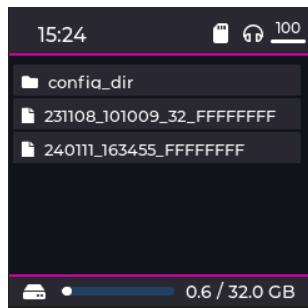


Figure 4.12: File Browser

File Browser

The file browser shows all files and folders located on the SD-Card. On the bottom of the page, a status bar indicates the current free and used space of the SD-Card. Due to the limited amount of memory, only the first 100 files and folders are displayed. This is however sufficient for most use cases.

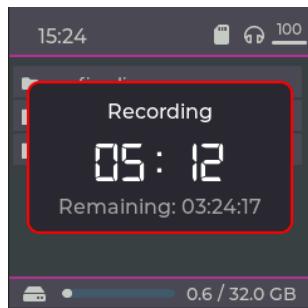


Figure 4.13: Recording

Recording

When the record button is pressed, the recording begins and a panel overlay is displayed. In the centre of the panel, the current recording time is displayed in minutes and seconds. Below, the remaining recording time is shown. When the recording is stopped, the panel overlay disappears. While the device is recording, all UI elements and the navigation are disabled.

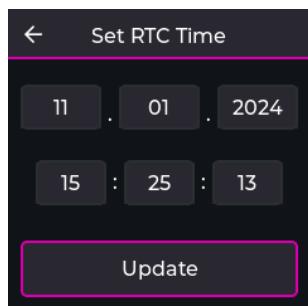


Figure 4.14: Set RTC Time

Set RTC Time

When the user clicks on the time in the header bar, the set time page is displayed. There the user can set the current time and date. After clicking on the *Update* button, the new time is set and the page is closed. To abort the process, the user can click on the arrow in the header bar.

5

Array Evaluation

5.1 Overview

As stated in chapter 3.2.4, the geometry of a microphone array has a impact on the performance of beamforming. The goal is to find a well suited array geometry for the detection and tracking of drones. To achieve this, first some array geometries have been simulated and then a prototype was built to gather information on how they perform in reality. Then the real data were compared to the simulated data to confirm the validity of the simulation. In the next step the findings of the prototypes and simulations were used to create a final array design.

By analyzing the sound of some commercially available drones a desired frequency range of 500 to 2000 Hz was set. This range includes most of the sound's energy. The array geometry has some constraints, such as the number of microphones and the mechanical feasibility. As stated in 4.5.3 the maximum number of microphones in an array is 32.

To compare different array geometries, several simulations were ran for each array. The simulated data was then ran through the beamforming algorithm to create the resulting beam pattern. The steering angles are based on a grid with $-180^\circ \leq \varphi < 180^\circ$ and $0^\circ \leq \theta \leq 90^\circ$ with a spacing of 1° . For each point in this grid the power in the desired frequency band is calculated. Displaying the result leads to an image like Figure 5.1 shows. The visualizations may be misleading, as they show a projection of a function defined on a semisphere onto a flat surface.

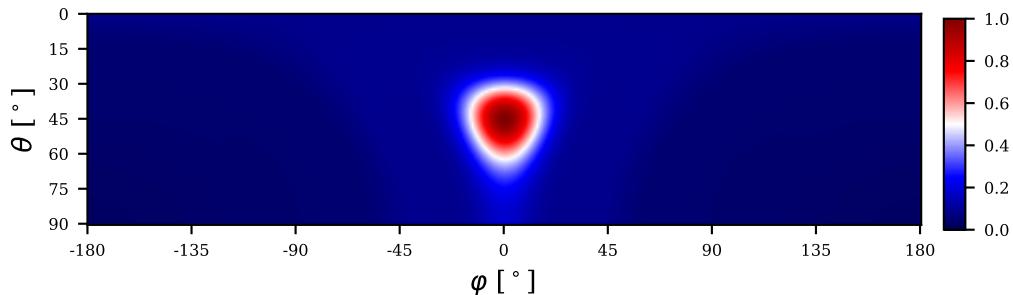


Figure 5.1: $G(\varphi, \theta)$ for $-180^\circ \leq \varphi \leq 180^\circ$ and $0^\circ \leq \theta \leq 90^\circ$. Red blob represents the main lobe where the power is bigger than the half of the maximum power.

5.2 Metrics

To evaluate the performance of an array several metrics are used. In [2] the main lobe width is proposed as an evaluation metric. Instead of the proposed lobe width the main lobe area is used in this thesis. The main lobe is defined as all points around the maximum of the beamforming pattern whose magnitude is larger than half of the maximum. Another metric used is from [7], which is the ratio between the main lobe power and average power over the whole grid.

5.2.1 Area Ratio

Since the grid resulting from the beamforming represents a function on a sphere, simply taking the sum of all the grid points from the main lobe would deliver false results. So each grid point is normalized by the surface element of a sphere

$$dA = r \sin \theta d\varphi \cdot r d\theta. \quad (5.1)$$

Since only the angles and not the radius are of interest, a radius of one is used for all calculations. To make the resulting numbers easier to understand, the calculated area is divided by the area of the semisphere. So the area ratio shows how much of the whole search space is taken by the main lobe.

The area of the main lobe determines how good different sources can be separated. With a big main lobe area two close sources may lead to one bigger main lobe instead of two separable ones. Since the area is calculated for $\theta \leq 90^\circ$, the areas calculated of peaks close to that border are not the actual areas when looking at the full angular range.

5.2.2 Peak-to-Average Power Ratio

The Peak-to-Average Power (PAP) ratio is the ratio between the peak power and the average power over the whole grid. The results from the projection are compensated by weighting the power at each grid cell with the area of that grid cell. With these considerations the PAP ratio is defined as

$$PAP = \frac{\max P(\varphi, \theta)}{\sum_{\varphi, \theta} P(\varphi, \theta) dA(\varphi, \theta)}. \quad (5.2)$$

A high PAP ratio is desired, since it means that little power is leaking into other directions than the real DOA.

5.3 Array Geometry

In [7] the authors use a circular array to detect and track drones. They however did not give any reasoning on why they chose a circular array. The authors of [9] measured and compared different array types. The best scores are reached with the Underbrink style array, a combination of multiple circular arrays. They also included several spiral based arrays. Based on this three main groups of arrays were further analyzed, the circular array, an adaption of the Underbrink array, and a Archimedean spiral array.

5.3.1 Circular Array

The circular array is the most simple shape of these three. It has two degrees of freedom, the circle radius and the number of microphones. Figure 5.2 shows how the PAP ratio changes when more microphones are added to an array with a fixed radius. For smaller arrays, the best possible PAP is reached with less microphones than with bigger arrays. This is expected since a bigger radius means bigger spacings between microphones which introduces aliasing.

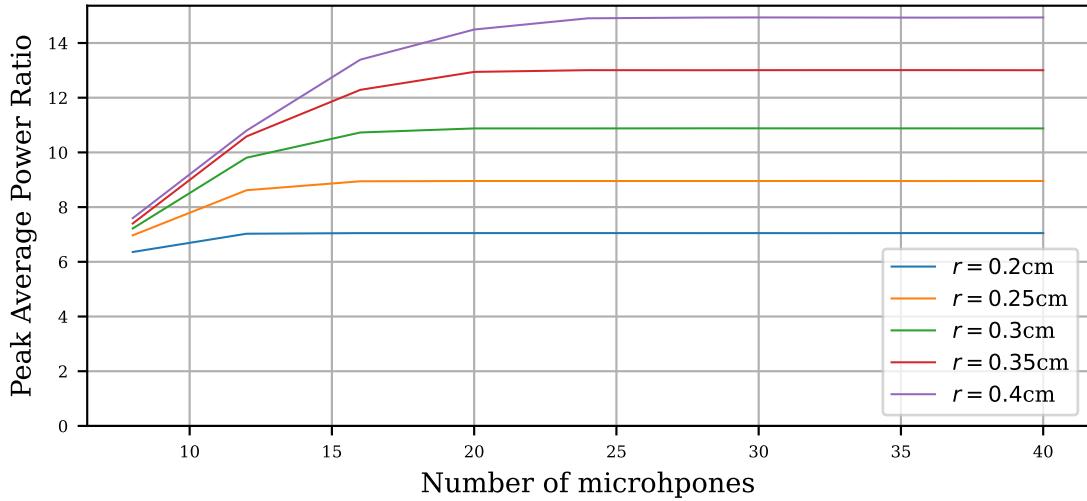


Figure 5.2: PAP ratio for different array radiiuses and total number of microphones.

Looking at Figure 5.3 and Figure 5.4 the change of the main lobe area and PAP ratio can be seen for an array with 32 microphones and varying radius. As expected the area decreases with increasing radius, since the lower frequencies will be sampled better. Also the PAP ratio becomes better with an increase of the radius. At a radius of 0.8 m however, the curve begins to flatten and even starts to decrease. It can as well be observed, that for smaller source angles the main lobe area is smaller. This is mostly an effect of the main lobe smearing in the θ direction. Recalling equation (3.12) with a fixed φ the time delay at a microphone is $t_n = C \sin(\theta)$ with $C = \text{const}$. Looking at the derivative of t_n with respect to θ it becomes clear that with angles close to 0° the change of t_n is high whereas if $\theta = 90^\circ$, t_n changes slowly when θ is changed.

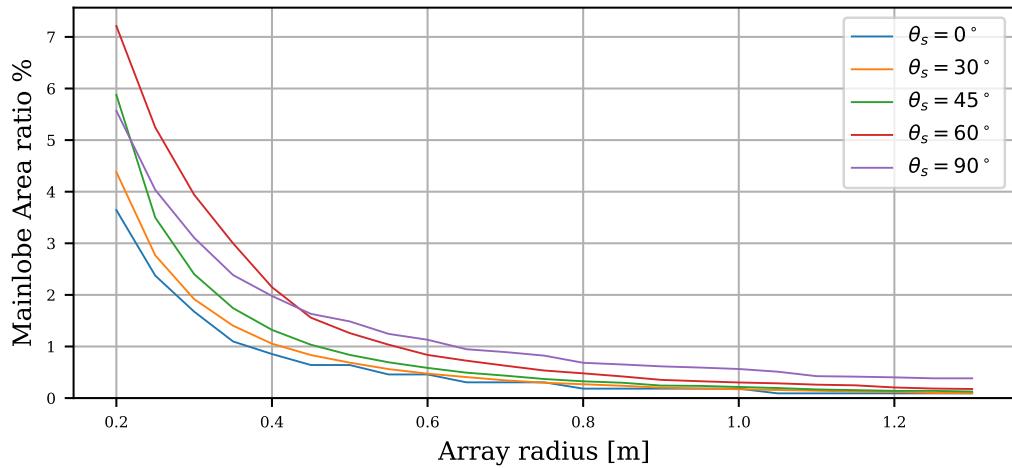


Figure 5.3: Main lobe area ratio of circular arrays with different radius.

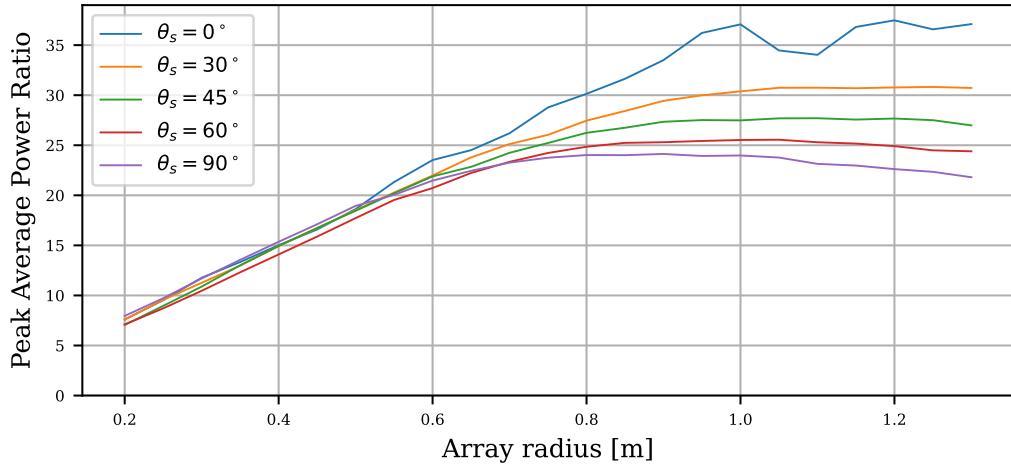


Figure 5.4: Peak Average power ratio of circular arrays with different radius

5.3.2 Multi Circular Array

The multi circular array is an simplified Undebrink array. In the original paper, the Undebrink array is designed, so that each microphone covers the same area of a ring with a fixed inner and outer radius and number of circles. In this thesis the design has been changed so that given the same parameters, the circle were equally spaced and the phase shift between the circles can be defined. An example of such an array is shown in Figure 5.5.

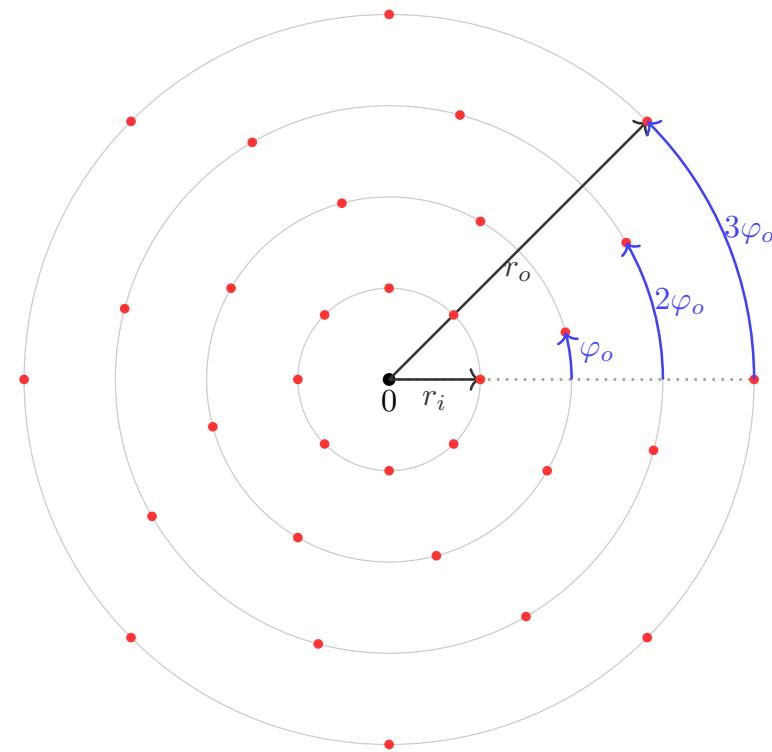


Figure 5.5: A Multi circular array that consists of four circular arrays with an offset angle φ_o between them.

All the parameters together lead to four degrees of freedom when designing such an array. Optimizing these four parameter with some mechanical feasibility constraints is not an simple task. So the effect of single parameters while the others were fixed was studied. First the amount of circles was set. Given the 32 microphones, the number of circles is preferred to be a divisor of 32. Four circles seemed like a good balance between number of circles and number of microphones per circle.

Similar to the circular array, the lower frequencies play a big role in the metrics. Smaller arrays that have trouble localizing these frequencies perform the worst as seen in figures 5.6 and 5.7.

To compare this geometry to the circular a maximum radius is set. In this example 0.4m. For a circular array the area ranges from $\approx 0.9\%$ to $\approx 2.1\%$ an the PAP ratio is between 14 and 16. Comparing this to the multicircular array a PAP ratio of up more than 20 can be reached. The areas however are generally larger and can lie in ranges between one and four percent.

This tradeoff reminds of windowing functions for the discrete Fourier transform.

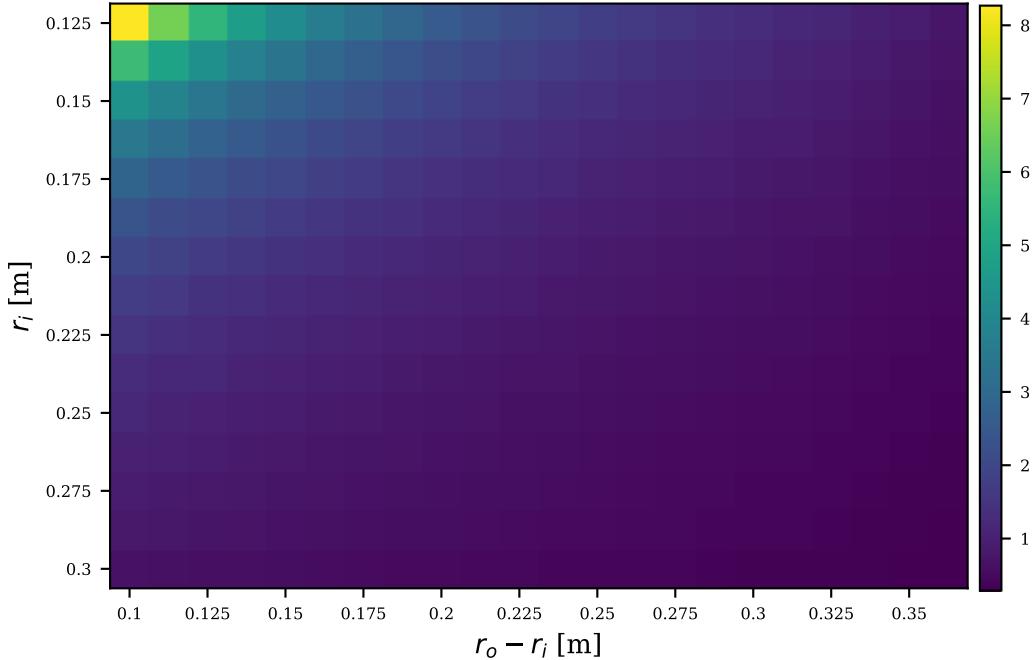


Figure 5.6: Area of the main lobe for multi circular arrays with different sizes.

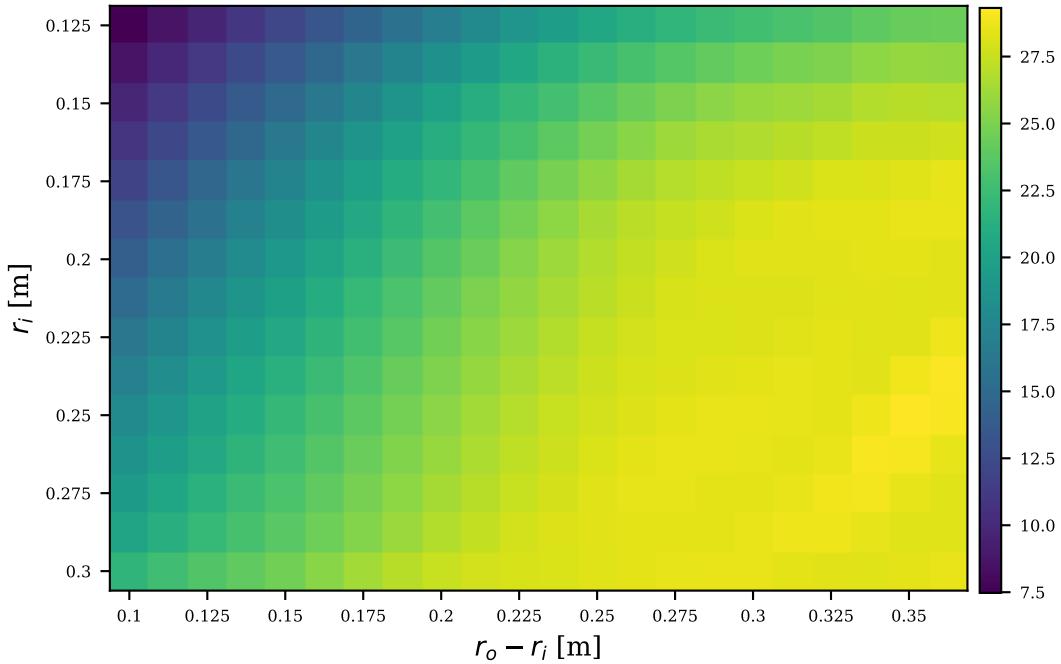


Figure 5.7: PAP ratio for multi circular arrays with different sizes.

5.3.3 Archimedean Spiral Array

An archimedean spiral in polar coordinates is defined as

$$r(\varphi) = a\varphi. \quad (5.3)$$

Making an array with this shape has three degrees of freedom, r_i of the first microphone, r_o of the last microphone and the spiral arm spacing d . Using these parameters, (5.3) is rewritten to

$$r(\varphi) = \frac{d}{2\pi}\varphi + r_i \quad (5.4)$$

for $0 \leq \varphi \leq (r_o - r_i)2\pi/d$ rad. Similar to the circular and the multi circular array, the archimedean array performs better when the max radius is increased. However care has to be taken when specifying the array as with certain configurations the microphones are not spread out similarly in each direction. As a consequence the performance can vary depending on φ_s .

5.4 Mechanical Design

The mechanical design of the array prototypes was centered around the objective of testing a wide range of array configurations. To achieve this, two flexible microphone array frames were developed. A significant aspect of the design process involved determining the overall size of the arrays. While larger arrays typically offer better performance, practicality and manufacturability had to be considered as well. In practice, a maximal outer diameter of 60 cm was chosen for the array prototypes. Based on simulation results, two specific array types were explored: The multi circular array and the Archimedean Spiral Array, as described in the next sections.

5.4.1 Multi Circular Array

This array setup allows to build two different multi-circular arrays with four rings. One with an $\varphi_o = 11.25^\circ$ and the second one with $\varphi_o = 22.5^\circ$. On the three outer circles a total of 32 microphones can be installed, to build circular arrays with radii of 20 cm, 25 cm and 30 cm. The inner circle can mount up to 16 microphones. It has a radius of 15 cm. The orange marked dots in Figure 5.8 show a possible microphone arrangement of a multi circular array, where each ring has an increased angle offset. The resulting pattern clearly shows an arm shaped structure.

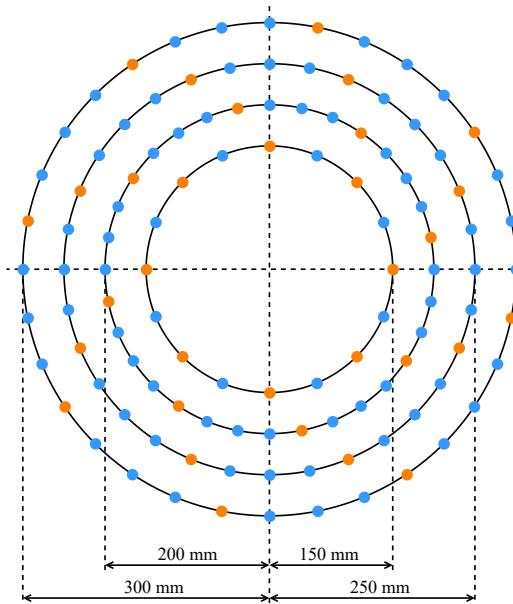


Figure 5.8: Multi-Circular Array

5.4.2 Archimedean Spiral Array

The archimedean Spiral Array was made with an inner radius r_i of 5 cm and an outer radius r_o of 30 cm. The spiral makes a total of four turns which leads to a separation distance d of 6.25 cm. The 32 microphones are placed in equal angle offsets on the spiral. In this particular setup the angle between microphones is $(4 \cdot 360^\circ)/(32 - 1) = 46^\circ$. Since the spiral has a distinct start and end point, the number of sections is one less than the number of microphones.

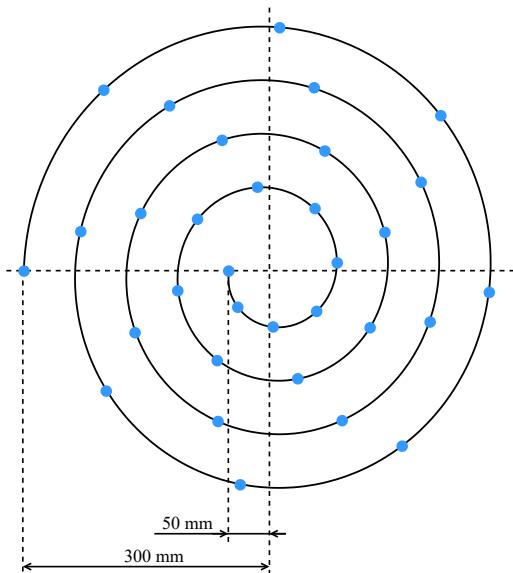


Figure 5.9: Archimedean Spiral Array

5.4.3 Wooden Prototype Arrays

Two wooden prototypes, a multi-circular array and an Archimedean spiral array, were manufactured by laser-cutting 5 mm plywood. Both arrays had to be split into several pieces due to the limited size of the laser-cutter and were later glued together. In the centre of each array, a mechanical mount for the Acquisition-System hardware was integrated.

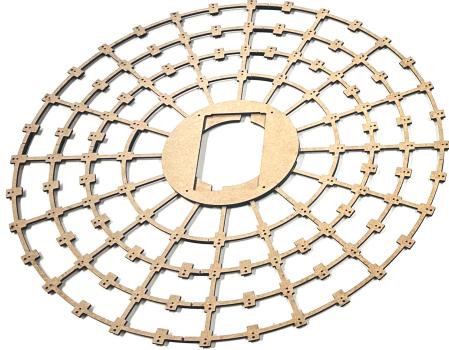


Figure 5.10: Wooden Circular Array

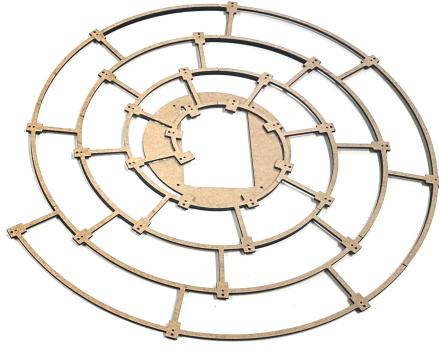


Figure 5.11: Wooden Archimedean Spiral Array

5.5 Measurements & Findings

The performance of the physical arrays were evaluated with outdoor measurement. To simulate a drone the sound of a *DJI Mavic Pro* drone was played back on *UE BOOM* loudspeakers. Various angles were tested by turning and tilting the array such that the loudspeakers are in the direction to be tested. The distance of the loudspeaker to the array was set to 15 m. All of the tests except the archimedean were made in an open field with little reverberation. The archimedean array tests were cut short and had to be carried out between buildings due to rain. In Table 5.1 the tested arrays and their parameter are listed.

	Type	Max radius	Other parameters
Circ 20	Circular	0.2 m	-
Circ 25	Circular	0.25 m	-
Circ 30	Circular	0.3 m	-
Multi circ 1	Multi-circular	0.3 m	$\varphi_o = 11.25^\circ$
Multi circ 2	Multi-circular	0.3 m	$\varphi_o = 22.5^\circ$
Arch	Archimedean	0.3 m	$r_i = 0.05 \text{ m}$, $d = 0.0625 \text{ m}$

Table 5.1: Test arrays.

During the first tests, the wind caused the microphone to overdrive, thus making the signal useless. To reduce this a *RØDE DeadWombat* microphone cover was cut into small pieces and glued onto the microphones. This lead to significantly less wind noises. Two examples of test configurations with the wind covers are depicted in Figure 5.12.

The measured data was evaluated using the main lobe area and PAP ratio. Additionally, to compare the real arrays to the simulated one, each measurement was also simulated. The result for a $\theta = 90^\circ$ and $\theta = 30^\circ$ are shown in Figure 5.13. In most

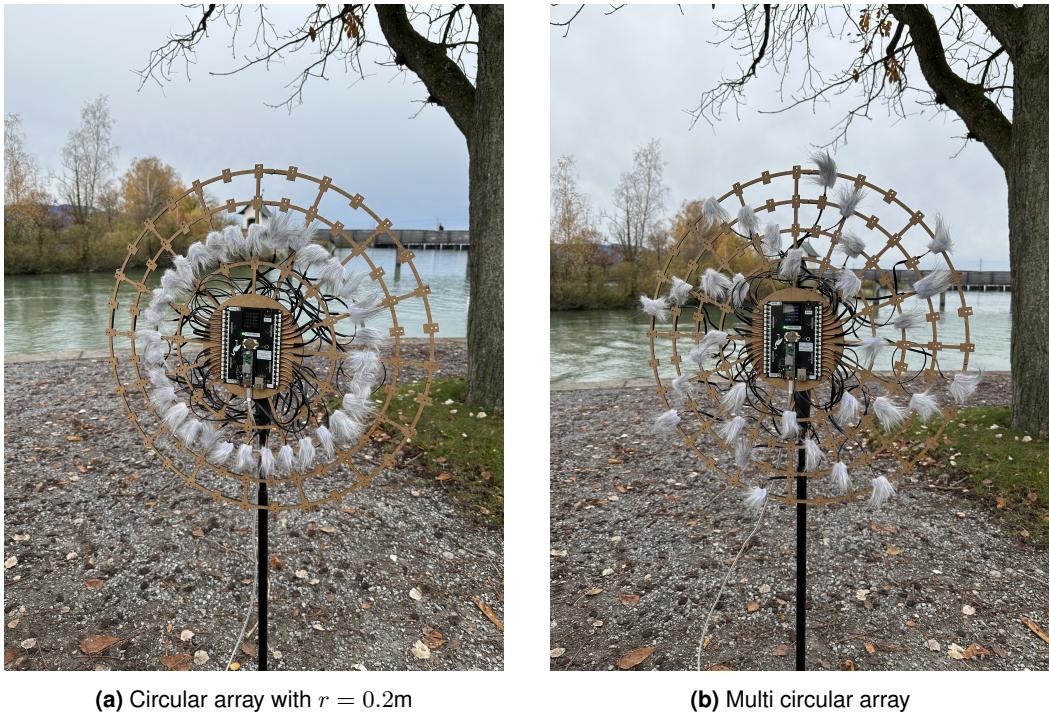


Figure 5.12: Two examples of the test arrays used for the measurement.

cases the simulated arrays performed better. This is probably due to lower background noise in the environment and other quite extreme differences can be seen with the Archimedean array which is suspected to be caused by the more reverberant measurement environment.

All the two dimensional array geometries tested have the drawback that their performance is dependent on θ_s . A directional independent array design is the spherical array [10]. It uses the theory of spherical harmonics to sample the sound at specific points on a sphere. The number of microphones used is depending on the order of the harmonics and so is the beam pattern. It was decided to not build such a spherical array due to its mechanical challenging geometry.

Expanding the idea of a three dimensional geometry several other possible advantages come to light. Until now the measurement space is a semisphere above the array with the array as the cutting surface. This can lead to ambiguity, when a sound source is under the array. The beamforming delays are the same for a source mirrored at the cutting surface. A three dimensional array does not have this problem.

Another positive effect of a three dimensional array is that the microphones are more spread out for sounds coming from low thetas. Ideally all the microphones are sampled at the exact same time, but in reality this may not be the case. From the microphone to the *Teensy 4.1* the signals are dependent on different clocks and filters, which are not guaranteed to be synchronous. With a low spread of the microphones, a small error in the sampling times could induce a larger error in the measured DOA than if the microphones were more spread out.

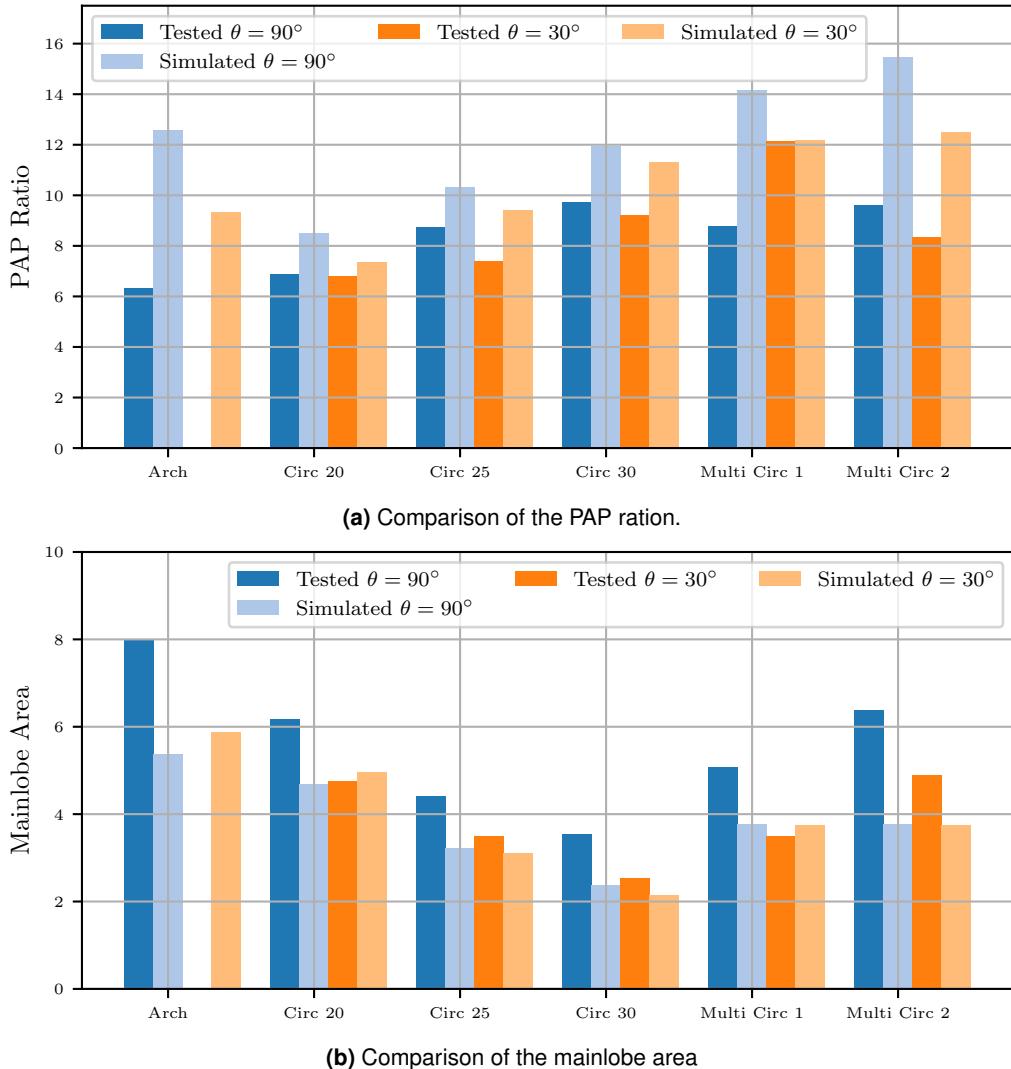


Figure 5.13: Measurements and simulation comparisons with a source located at $(r, \varphi, \theta) = (15 \text{ m}, 0, \theta)$. A test measurement for $\theta = 30^\circ$ was not performed with the Archimedean array.

5.6 Final Array Geometry

Finally, the idea of a multicircular array where the circles can be set to a different height came up. In the measurements the multicircular array performed well. To get an optimal array geometry, the maximal radius of 0.38 m was first set. Even though a larger radius would yield better results, it also comes with more mechanical challenges. The angle φ_o chosen to be the same as in the test array *Multi circ 1* as it performed the best of the two multicircular configurations.

To find a suited r_i , the main lobe area and PAP ratio was calculated for various r_i . The results are depicted in Figure 5.14. Figure 5.14b shows that the PAP does not steadily increase but rather peak at a specific radius and then begins to decrease again. A r_i of 0.2 m was chosen since it is a good tradeoff between main lobe area and PAP ratio. Figure 5.15 shows the final properties of the array.

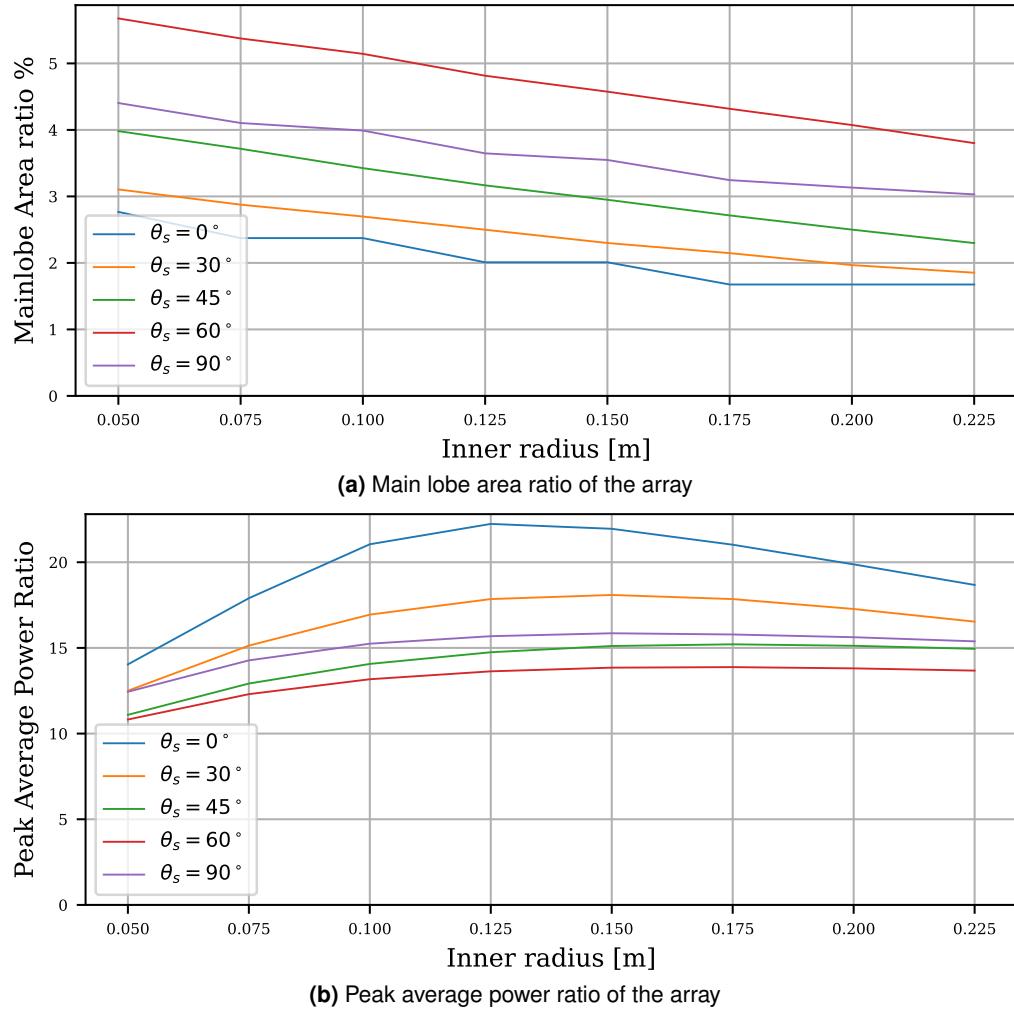


Figure 5.14: Metrics of a four circles multi circular array with $r_o = 0.38\text{ m}$, $\varphi_o = 11.25^\circ$ and 32 Microphones, depending on r_i

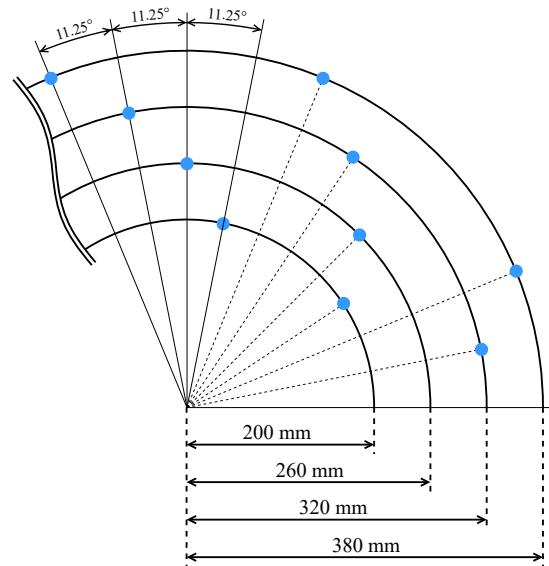


Figure 5.15: Final Array Concept Design

To make the circular sub array on different height levels a mechanical system is used as described in Section 6.2. The performance of the array changes with the angle of the microphone arms. Since a bigger angle lowers the distances between the microphones it is to be expected, that the performance decreases due to the influence of the lower frequencies. Figure 5.16 shows how the metrics change with the arm angle. Up to an angle of 30° performance decreases slowly. Therefore the predicted optimal angle range is between 0° and 30° .

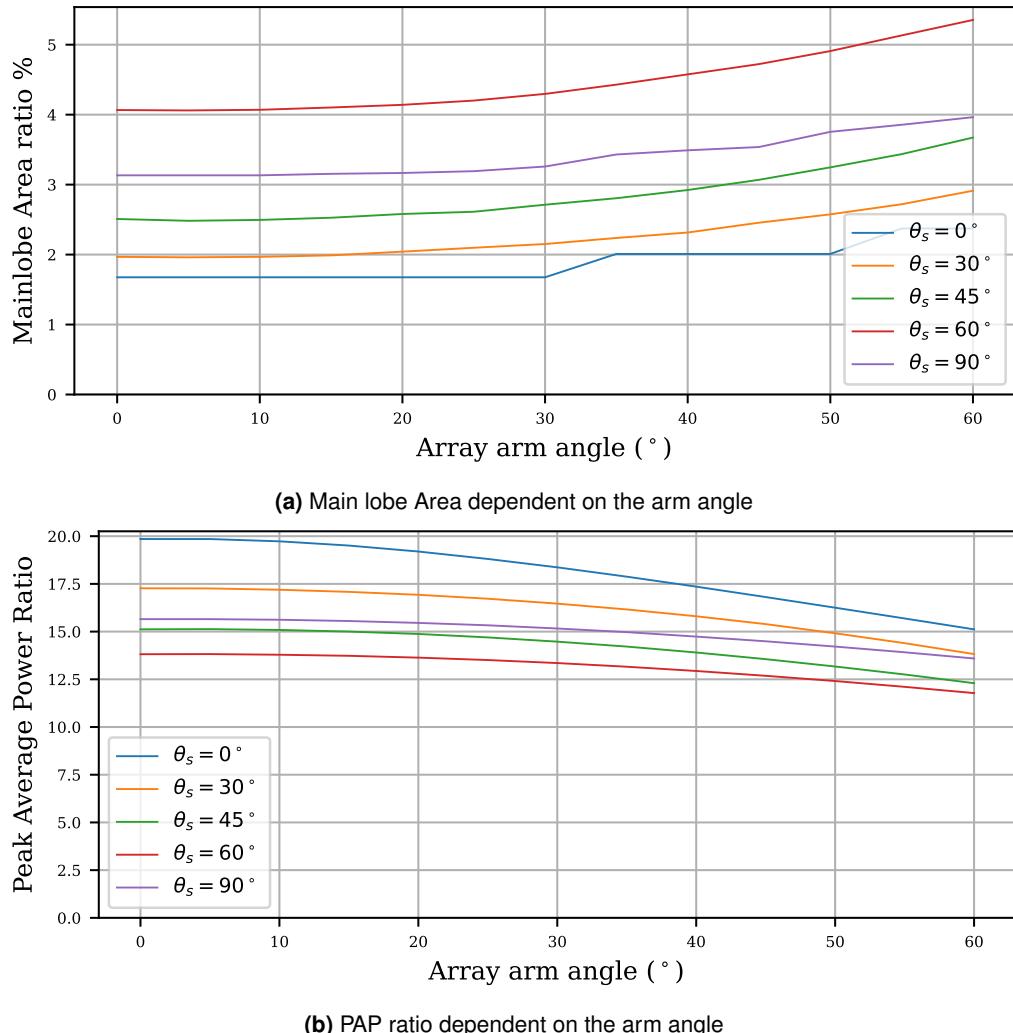


Figure 5.16: The metrics dependent on the Array arm angle and θ_s based on simulated data.

6

Final Design

6.1 Overview

Now that a wealth of valuable experience has been gathered through simulations and extensive testing, a specific microphone type has been selected and an optimal array geometry for this project's use case has been identified. This selection process was critical for achieving the project's goal of designing a fully functional, professional end product. This chapter outlines the final design of the sound source localization system.

In the research detailed in section 5.5, a key hypothesis emerged regarding the potential for improved sound source localization with a three-dimensional array structure as compared to a planar geometry. This led to the development of a new mechanical concept, which is detailed later in this chapter.

For the system to function independently, it was essential to incorporate all necessary components within the design. However, due to the computational intensity of beam-forming algorithms, they cannot be processed on a microcontroller like the Teensy 4.1. Consequently, the design had to accommodate external processing capabilities.

Another significant consideration was that a single microphone array could only provide a directional vector of located sound sources without information about the distance to these sources. To pinpoint an object's location accurately, multiple microphone arrays are needed. This requirement necessitated a design that allows for easy integration with a centralized computer or server, capable of connecting to multiple microphone arrays.

The optimal solution was to design a system that could stream all lossless audio data from each microphone array to the server in real-time. Such a design ensures maximal flexibility, as it allows to apply basically any sort of localization algorithm on the server side. In addition, it enables the usage of multiple microphone arrays, which can be placed at different locations, thus increasing the overall accuracy of the system.

6.1.1 Key Requirements

The following key requirements have been set:

- Simultaneous streaming of 32 microphone channels in real-time
- Lossless audio transmission with a sampling rate of 44.1 kHz and 16 bit resolution
- GNSS RTK receiver for accurate positioning and timing information
- IMU for geographic heading and leveling information
- Ambient pressure and temperature sensor for calculating the exact speed of sound
- Ethernet interface for audio streaming and device configuration
- PoE power supply for single cable operation over long distances (up to 100 m)
- Intuitive Graphical User Interface (GUI) for device configuration and monitoring
- Flexible mounting option for installation on a standard camera tripod
- Microphone integration that is resistant to the influence of wind
- Adjustable array arm angle for optimal beamforming performance
- RGB LEDs for visual feedback and status indication

6.1.2 Key Decisions

The following section describes the key decisions made during the development of the final design.

1. **Microphone Type:** The *MP34DT05TR-A* model by ST Microelectronics was selected for its top-ported design, offering superior audio quality and sensitivity during tests.
2. **Microphone Integration:** To streamline the design and assembly process, dedicated PCBs were designed to accommodate four microphones each. This approach minimizes the need for multiple connections, as only one flex-cable is required to connect each arm to the mainboard.
3. **Ethernet Protocol:** Given the necessity for reliable audio transmission, a TCP connection was required, since a regular UDP connection does not guarantee packet delivery.
4. **MCU Selection:** The *Teensy 4.1* was chosen again for its proven capability in handling audio processing. Its built-in 100Base-T Ethernet interface is particularly beneficial for enabling audio streaming over a LAN network.
5. **GNSS Receiver:** A GNSS RTK module from *uBlox* was selected due to its high accuracy and extensive software support.
6. **Angle Sensor:** To accurately determine the positioning of the microphone array arms, a magnetic angle sensor was used. This sensor is essential for adjusting the beamforming algorithm to the current array geometry.
7. **Mechanical Construction:** An aluminum sheet metal construction was chosen for its cost-effective manufacturing and robustness.
8. **Touch Display:** A touch display was integrated to facilitate easy monitoring and configuration of the system.

6.2 Mechanical Design

The mechanical design of the sound source localization system is characterized by its unique foldable microphone arms, conceptually inspired by the functionality of an umbrella. *SolidWorks 2023* was utilized for the design process, facilitating efficient development.

Aluminum, chosen for its lightweight and durable properties, was the primary material used. It also ensures resistance to corrosion, an important consideration for long-term use. The CAD software's advanced sheet metal design capabilities strategically accelerated the design phase.

All sheet metal parts were laser-cut, then underwent post-processing including threading and countersinking. These components were fabricated by *Blexon*, a Swiss metal manufacturing service, at a cost of approximately 350 CHF per unit. Other aluminum parts essential to the design were produced through CNC machining at the university's workshop.

Figure 6.1 shows a 3D rendering of the final design.



Figure 6.1: 3D Rendering of the Final Design

6.2.1 Part List

The following list provides an overview of all custom manufactured metal parts:

- 1x Top Mounting Platform: 2.5 mm aluminum sheet metal
- 1x Bottom Sliding Platform: 2.5 mm aluminum sheet metal
- 8x Microphone Arm Support Frame: 2.5 mm aluminum sheet metal
- 8x Connection Bars: 5 mm aluminum sheet metal
- 1x Antenna Platform: 2.5 mm aluminum sheet metal
- 1x Main Mounting Pole: $\varnothing 20 \times 400$ mm aluminum rod (Drawing: A.42)
- 1x Top Mounting Ring: $\varnothing 44 \times 45$ mm aluminum part (Drawing: A.43)
- 1x Bottom Sliding Ring: $\varnothing 44 \times 15$ mm aluminum part (Drawing: A.44)
- 1x Antenna Top Mount: $\varnothing 26 \text{ mm} \times 12 \text{ mm}$ aluminum part (Drawing: A.45)
- 1x Antenna Bottom Mount: $29.5 \times 14.5 \times 15.0$ mm aluminum part (Drawing: A.46)
- 1x Antenna Pole: $\varnothing 6 \times 240$ mm stainless steel rod (Drawing: A.47)

6.2.2 Microphone Arm Design

Figure 6.2 shows the mechanical dimensions of the microphone arm. The pivot point is located at 95 mm from the axial center of the microphone array. Each microphone arm is oriented at an angle of 45° to the center pole.

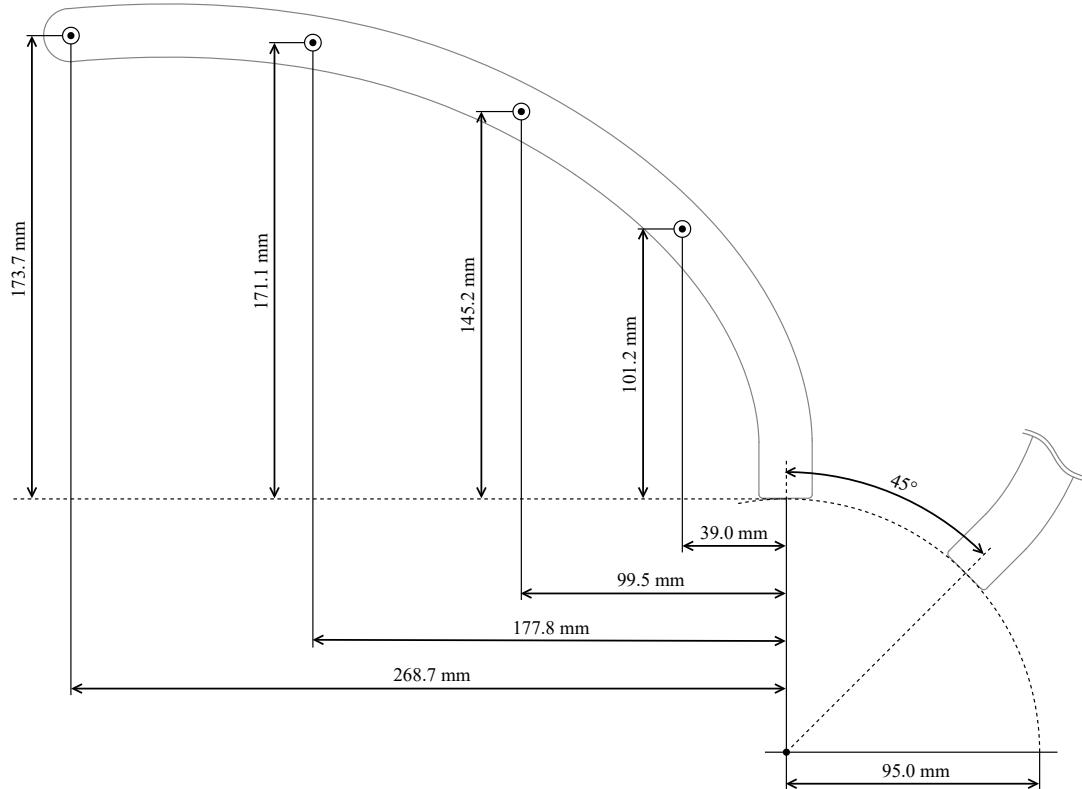


Figure 6.2: Mechanical Dimensions of the Microphone Arm

6.2.3 Folding Mechanism

The microphone arms can be fully extended to achieve a planar array geometry or folded towards the center pole to a minimum angle of 90° . As figure 6.3 shows, a cone shaped array structure results from this design. The angle can be freely adjusted or locked in place using a M6 knurled screw located at the lower mounting platform.

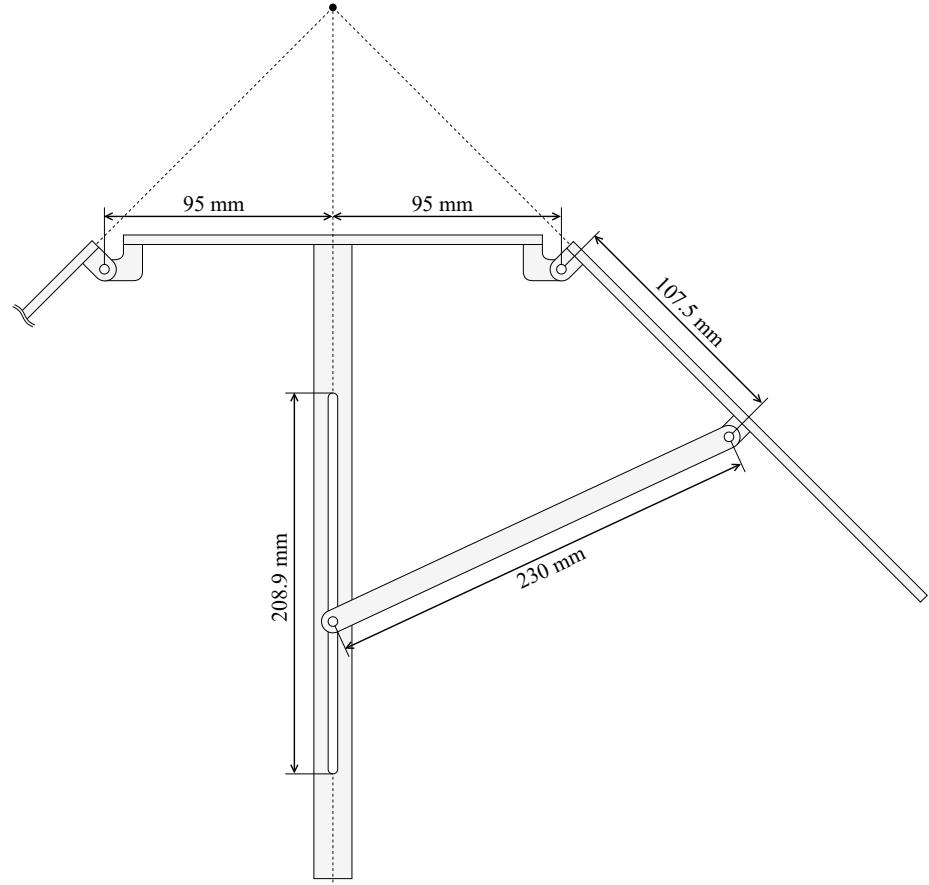


Figure 6.3: Mechanical Design of Folding Mechanism

6.2.4 Microphone Wind Protection

As mentioned in section 5.5, the microphone wind protection is a critical component of the design. Therefore, each microphone is equipped with specialized microphone wind protection fur, which is attached to the microphone arm using a 3D-printed clip, as shown in figure 6.4.



Figure 6.4: 3D-Printed Fur Clip



Figure 6.5: Microphone Wind Protection Fur

6.3 Hardware Design

The hardware design of the final sound source localization system is characterized by its Multi-PCB construction. The system's central component is the mainboard, which houses the processing unit, various sensors, and the power supply. In the center of the mainboard, a round touch display is mounted, which serves as the primary interface for user interaction.

Surrounding the mainboard are eight microphone arm PCBs, each connected to the mainboard by 50 mm long, 10-Pin flex-cables with a 1 mm pin pitch. These flex-cables provide the mechanical flexibility required adjusting the angle of the microphone arms. Mounted on each microphone arm PCB are four MEMS microphones. These microphones are strategically positioned as specified in section 5.6. Additionally, each arm is equipped with eight RGB LEDs, which serve both functional and aesthetic purposes, offering visual feedback and enhancing the system's overall design. To accurately measure the angle of the microphone arms, a dedicated angle sensor PCB was developed.

The entire hardware was designed using *Altium Designer 23*, a leading software in PCB design.



Figure 6.6: Front View of the Mainboard

6.3.1 Block Diagram

The system block diagram in figure 6.7 provides an overview of the hardware design. Note that components marked inside dashed boxes are located on separate PCBs.

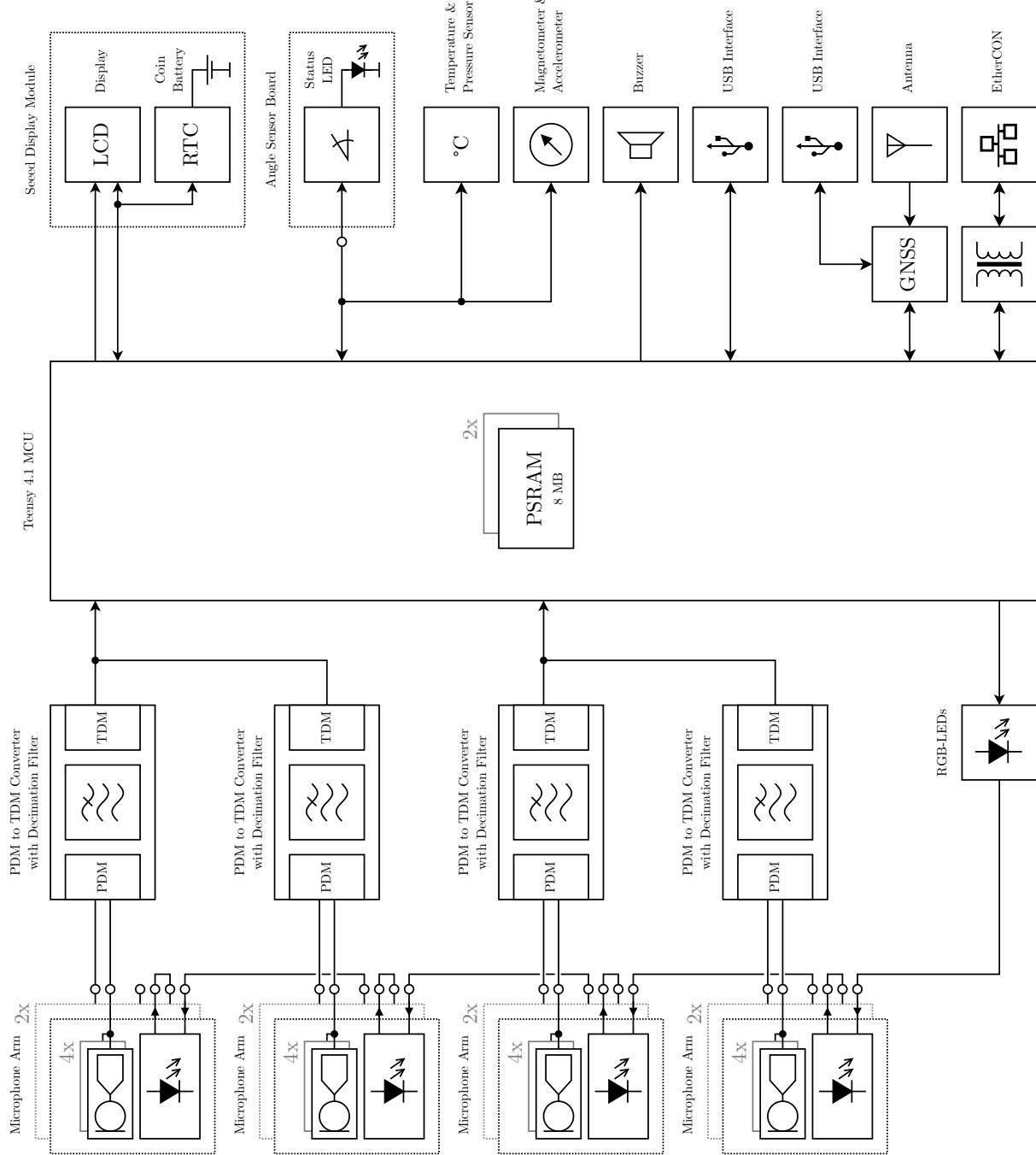


Figure 6.7: System Block Diagram

6.3.2 Power Supply

The device can be powered by multiple sources. Primary the device is powered by the ethernet interface, which provides a Power over Ethernet (PoE) connection. This is the preferred method, as it is the most convenient way to use the microphone array in the field. In addition, there are two USB-C ports (MCU and GNSS) that can be used to supply power when no PoE connection is available (e.g. in a laboratory environment for programming and debugging). Each supply method can be used in conjunction with each other. However, the source must be able to provide at least 12.5 W of power (5 V, 2.5 A). In figure 6.8 an overview of the power supply is shown. Note that the Teensy 4.1 has an internal 3.3 V regulator, which is powered by the systems internal 5 V supply. The 1.8 V rail is generated by a dedicated linear voltage regulator and is mainly used for the PDM to TDM converters (ADAU7118).

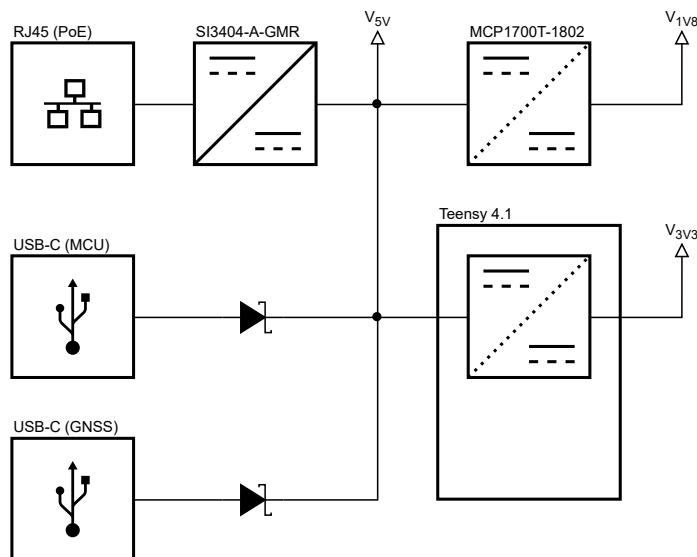


Figure 6.8: Power Supply Overview

Power over Ethernet (PoE)

For the power supply of the device, the Power over Ethernet (PoE) standard IEEE 802.3at was implemented. This standard allows for up to 12.95 W of power to be delivered to the consumer side. The system employs active PoE, necessitating the use of a negotiation Integrated Circuit (IC). The *SI3404* by *Skyworks* was selected for this purpose, which, in addition to handling the communication protocol, incorporates a flyback step-down converter controller.

To adhere to PoE standards, it is mandatory for every power sink device to be galvanically isolated from the power sourcing equipment. In this design, a transformer is used to step down the PoE voltage, typically around 56 V, to the system voltage of 5 V. Furthermore, a second Ethernet signal transformer is employed to isolate the two data transmission pairs. The total output power of the system is designed to be 2.5 A at 5 V.

For the Ethernet connection, the *EtherCON* connector by *Neutrik* was chosen. This connector offers enhanced mechanical robustness compared to standard RJ45 connectors. However, it is also compatible with standard RJ45 connectors.

6.3.3 Microcontroller Unit (MCU)

As the main processing unit, the *Teensy 4.1* was selected again, since it proved to fulfill all requirements. However, during the firmware development, it became apparent that to meet the necessary ethernet throughput and internal processing power for the project, the *Teensy 4.1* had to be overclocked to 912 MHz. This increase in processing speed necessitated additional measures for thermal dissipation. Consequently, a passive heatsink was mounted onto the SoC to effectively manage the increased heat generation.

Furthermore, the use of two external 8 MB PSRAM chips was again deemed necessary. These additional memory chips provide sufficient audio buffering capabilities, essential for the system's performance in handling and processing the audio stream.

6.3.4 Audio Input

The audio input hardware builds upon the previous design detailed in section 4.5.3, featuring four *ADAU7118* PDM to TDM converters. Two of these converters are connected to each TDM-16 interface on the *Teensy 4.1*. Each converter IC manages eight MEMS microphones, effectively handling two microphone arms.

6.3.5 GNSS Receiver

The mainboard of the sound source localization system employs a *NEO-M8P-2* module by *uBlox* as its GNSS receiver, chosen for its ease of integration into embedded systems through its versatile serial interfaces, including SPI, I²C, UART, and USB. For this specific application, the I²C interface was utilized to connect the module to the *Teensy 4.1*. Additionally, the module's USB interface is extended to a dedicated USB-C port, offering enhanced debugging capabilities with tools provided by *uBlox*.

The design of the antenna frontend is tailored to drive active GNSS antennas, incorporating an inductor to couple the 3.3 V supply voltage into the RF line. This setup powers the built-in LNA of the external antenna, ensuring optimal signal reception. A standard SMA connector is used for connecting the antenna.

The module is also equipped with direct digital outputs for conveying status information. A yellow LED, linked to the RTK-Status output, illuminates to indicate when the module is in RTK mode and has precisely determined its position. Additionally, a red LED connected to the 1 Hz time pulse output serves as an indicator, flashing briefly every second to signify a fixed GNSS status.

External Antenna

The external GNSS antenna selected for this project is the *ANN-MB1-00* from *uBlox*. It supports both the L1 and L5 bands, making it versatile for various satellite systems including GPS, GLONASS, Galileo, and BeiDou. The original 5 m cable was shortened to approximately 30 cm, and a new SMA connector was attached to the end. A crucial aspect of these multi-band antennas is their need for a metal base with a diameter of at least 10 cm to enhance signal reception. To meet this requirement, the antenna is mounted on a dedicated aluminum platform, as shown in figure 6.1.



Figure 6.9: GNSS Antenna (ANN-MB1-00)

6.3.6 Display Module

The display module, a round TFT unit from *Seeed Electronics*, features a 240 x 240 pixel resolution and a 1.28 inch (3.3 cm) diameter. It operates via a high-speed SPI interface at 80 MHz, allowing a smooth 30 FPS refresh rate. Notably, the module includes an integrated Real-Time Clock (RTC) of the type *PCF8563*, powered by a *CR 927* (3 V) coin cell battery, ensuring continuous timekeeping even when the main power is off. Additionally, the backlight brightness is adjustable via a Pulse-Width Modulation (PWM) output pin on the *Teensy 4.1*.

Both the touchscreen and the RTC are connected to the *Teensy 4.1* through the same I²C. The module is mounted to the mainbaord with three M2 screws and a custom 3D-printed mounting ring.

6.3.7 RGB LEDs

In the sound source localization system, a total of 81 individually addressable RGB LEDs are utilized. These LEDs are of the type *WS2812B-0807* and have a size of 2.0 x 1.8 mm. They feature a single wire data interface, which operates at 800 kHz and are powered by the 5 V supply. However, due to the high potential power consumption when all LEDs are set to full brightness (white), it is crucial to manage the power draw carefully. The maximal power consumption of all LEDs is approximately

$$P_{\text{tot}} = V_{\text{sup}} \cdot N_{\text{LED}} \cdot I_{\text{max}} = 5 \text{ V} \cdot 81 \cdot 19 \text{ mA} \approx 7.7 \text{ W}. \quad (6.1)$$

Consequently, the global brightness of the LEDs must be limited through software.

6.3.8 Sensors

The mainboard of the sound source localization system is equipped with a variety of sensors. Firstly, a combined magnetometer and accelerometer is implemented. It provides crucial heading and leveling information for the device, which is essential for providing accurate positioning information, specifically when utilizing multiple microphone arrays.

Additionally, the system incorporates a combined ambient pressure and temperature sensor. This sensor plays a vital role in monitoring the ambient air conditions, a factor that significantly influences the accuracy of the beamforming algorithms used in sound localization.

Furthermore, a magnetic angle sensor is integrated into the design. This sensor determines the current positions of the array arms. Since all the arms are mechanically coupled and maintain the same angle relative to each other, only one of these angle sensors is required.

All sensors are connected to the *Teensy 4.1* via a shared I²C bus running at only 100 kHz due to the relativly long physical distance between the sensors and the MCU.



Figure 6.10: Seeed Round Display Module

Magnetometer & Accelerometer Sensor

The *LSM303AGRTR* by *ST Microelectronics* was selected as the magnetometer and accelerometer, largely due to its wide availability, extensive software support, and high resolution. This sensor uniquely combines a sensitive 3-axis magnetic flux measurement unit with a 3-axis *LIS2MDL* accelerometer, providing comprehensive motion and orientation sensing capabilities.

A critical aspect of using any kind of magnetometer is the compensation of interference from ferromagnetic and paramagnetic materials. This necessitates regular calibration of the sensor, especially when there are changes in the surrounding environment. This calibration is essential to maintain the accuracy and reliability of the sensor's readings. Detailed information and guidelines for the calibration process can be found in section 6.4.5

Ambient Pressure & Temperature Sensor

The *BMP388* by *Bosch Sensortec* was selected as the ambient pressure and temperature sensor. This sensor has a pressure measurement range of 300 - 1250 hPa, with an accuracy of ± 8 Pa (≈ 0.66 m) within the typical operating range of 900 - 1100 hPa. The integrated temperature sensor features a measurement accuracy of ± 0.3 °C in the range of 0 – 65 °C.

During the PCB design for the mainboard, however, the thermal placement of this sensor was not adequately considered. Consequently, heat-emitting components on the board affect the *BMP388*'s temperature readings, leading to a temperature offset of approximately 2.5 °C. A better approach would have been to isolate the sensor thermally, e.g. by creating a slot around the sensor on the PCB.

Angle Sensor

The design utilizes the *AS5600* magnetic angle sensor from *AMS-Osram Semiconductors*. This sensor offers 12-bit resolution and can accurately detect the position of a circular magnet, mounted at a distance of 0.5 mm to 3 mm. Additionally, this sensor features a PWM digital output, that has been connected to a dedicated LED on the angle sensor PCB. This LED provides a visual indication of the current angle being measured and confirms magnet detection.

Figure 6.11 shows the principle of operation of the *AS5600* angle sensor.

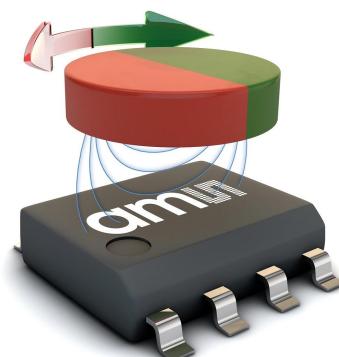


Figure 6.11: Operation Principle of the AS5600 Angle Sensor [3]

6.3.9 Printed Circuit Boards (PCBs)

The sound source localization system is based on three custom designed PCBs.

Mainboard

The Mainbaord is based on a 4-layer round PCB with a diameter of 150 mm. It utilizes a double-sided component assembly, primarily positioning elements on the PCB's underside, as detailed in the 3D visualization 6.12. A key design challenge involved accommodating components taller than 4 mm. These had to be placed within the white-marked areas, corresponding to cutouts in the aluminum top mounting plate. The PCB is securely mounted using five surface-mounted standoff M3 nuts.



Figure 6.12: Mainbaord PCB

Microphone Arm

The Microphone Arm PCB is a 2-layer design, measuring 288.7 x 190.4 mm, with all components mounted on the top side. It features two surface-mounted M3 nuts near the connector side, which provide mechanical rigidity when mounted on the support frame. Additionally, the PCB features seven 3.2 mm mounting holes placed for attaching the microphone fur clips.



Figure 6.13: Microphone Arm PCB

Angle Sensor

The Angle Sensor PCB is a 2-layer design, measuring 61.25 x 16.5 mm, with all components mounted on the bottom side. A standard 2.54 mm pin header is used for connecting it to the mainboard. The magnetic angle sensor IC is placed exactly at the pivot point of the microphone arm. The PCB is mounted to the main frame using a 3D-printed fixture.

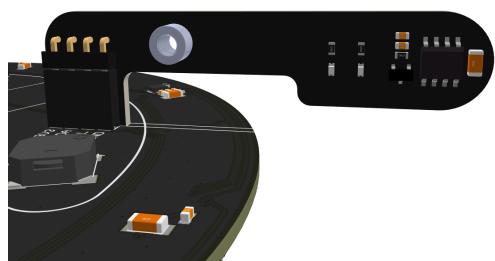


Figure 6.14: Angle Sensor PCB

6.3.10 Manufacturing

The PCBs were manufactured and assembled by *JLCPCB*. Only a few additional components were soldered by hand, such as the PDM to TDM converters (ADAU7118), the EtherCON connector and the GNSS module (NEO-M8P-2). It turned out that multiple microphone arm PCBs were faulty (damaged microphones). This was most likely caused by the soldering process, as the MEMS microphones are very sensitive to heat.

6.4 Firmware Design

The firmware of the sound source localization system is build upon the audio acquisition system firmware. It runs again on the *Teensy 4.1* MCU and the same toolchain has been used as described in section 4.6. One of the main components of the firmware is the integration of the *QNEthernet* stack. However, integrating it proved to be quite challenging. The *QNEthernet* stack is not designed for a multithreaded environment, necessitating the main ethernet handler to operate in the main thread, i.e., the background thread.

6.4.1 Overview

The firmware is structured into multiple software modules, most of them running in their own thread. Careful balancing of the thread priorities was essential to ensure the system's stability and performance. Especially, enough processing headroom for the background thread had to be reserved, for high ethernet throughput and a fluent display refresh rate. Table 6.1 provides an overview of all threads and their purpose.

Thread	Purpose
Console Interface	Handles USB virtual COM-Port
Console Streaming	Handles queuing of messages
Utils	Updates operation time
AudioUtils	Audio Processing
GNSS	GNSS Data Handling
HMI	LED Control & RTC
Buzzer	Buzzer Control
Application	Main Application Logic
Main	Main Thread (Background)

Table 6.1: Overview of all Threads and their Purpose

6.4.2 Audio Streaming

The audio streaming module handles the buffering and transmission of the 32 audio channels. It is based on a TCP server that provides a socket connection on port 6666. The audio data is transmitted in a lossless 16-bit signed integer format, with a sample rate of 44.1 kHz.

The TCP connection assures a reliable data transmission, which is essential for a continuous audio stream. To ensure low latency, the audio data is buffered in a circular buffer of 12 MB size. This allows a maximal buffering time of approximately 4.4 s. A minimal transmission rate of

$$\frac{32 \text{ channels} \cdot 16 \text{ bit} \cdot 44100 \text{ Hz}}{10^6 \text{ bit/s}} = 22.1184 \text{ Mbit/s} \quad (6.2)$$

is required. The theoretical maximum transmission rate of 100Base-T Ethernet is 100 Mbit/s. However, the real-world tests with the *Teensy 4.1* showed a maximum transmission rate of around 60 Mbit/s, which is still sufficient for the audio stream.

Due to the maximal TCP packet size of 1460 bytes, the audio data is split into concatenated packets. A frame consists of 128 interleaved samples (32 channels) and is transmitted in 8 packets. Each frame starts with a 20-byte header, beginning with a magic sequence that allows the receiver to synchronize the incoming data stream. Next is the packet index, which is incremented for each frame and enables the receiver to detect missing packets. The last field is the timestamp in the UNIX epoch format in nanoseconds. Table 6.2 shows the structure of the packet header.

Byte Offset	Data Format	Description	Example Value
0-7	String	Magic Sequence	HERON666
8-11	Integer (Little Endian)	Packet Index	12345
12-19	Integer (Little Endian)	Timestamp (ns)	1713586842696942069

Table 6.2: Description of the 20-byte Packet Header

Figure 6.15 shows an example of a transmission sequence. Note the interleaved sample order, which is based on the WAV file format.

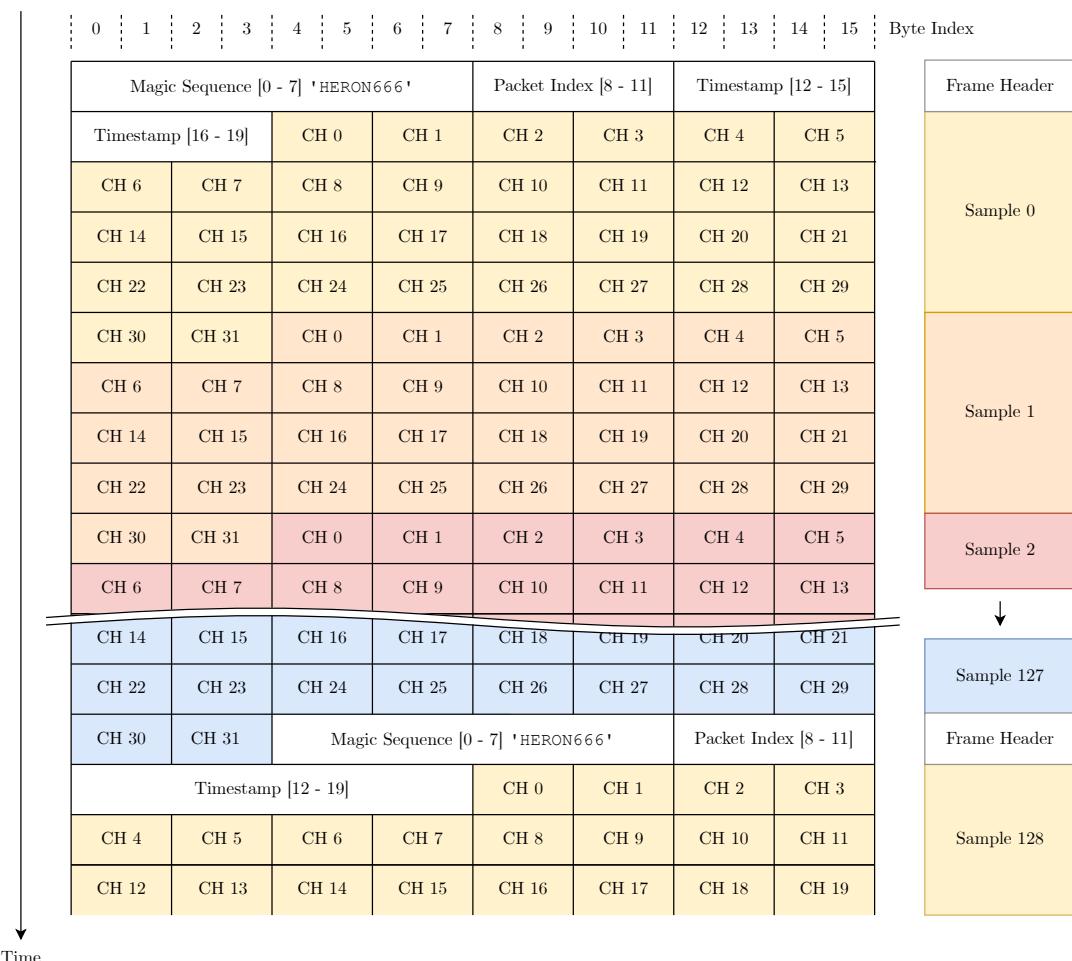


Figure 6.15: Example of a Frame with 128 Samples (32 Channels)

6.4.3 GNSS & RTC Time

The system incorporates two methods for timekeeping: A GNSS module and a Real-Time Clock (RTC). The GNSS module is highly accurate, providing time information down to the nanosecond, which is essential for providing accurate timestamps for the audio stream. In contrast, the RTC is designed to maintain time even if the device loses power. Once the GNSS time is established and stable, it automatically synchronizes the RTC. The GNSS module sends periodically its timing information to the MCU at a rate of 5 Hz. Whenever a new time packet is received, the system logs the current internal system time in microseconds. This allows calculating the time offset to any subsequent API call which asks for the current time.

In situations where GNSS data is not available, the system falls back to the RTC time. While this ensures continuous timekeeping, the RTC's limitation to second-level resolution presents a challenge. It does not provide detailed information about the exact moment whenever a second has passed. To address this and achieve more precise timing information, a specific algorithm is required as described in the following section.

Phased Locked Loop (PLL)

To estimate the subseconds of the current RTC time, a Phase-Locked Loop (PLL) has been used in combination with a PI-controller. This approach effectively smooths out the RTC's stair-step signal, enhancing time resolution. The Proportional (P) component of the PI-controller is set to zero to avoid amplifying the RTC's abrupt second changes. Instead, the focus is set on the Integral (I) component, which has been tuned by simulating the RTC's behavior in *Python*. In practice, an integral gain of 0.01 was found to provide best results. The PLL algorithm is executed 5 times per second and can be described as

$$\begin{aligned} t_{\text{PLL}} &= t_{\text{SYS}} - t_{\Delta} + 0.5 \text{s} \\ t_{\Delta} &= K_p \cdot e + K_i \cdot \sum e \\ e &= t_{\text{SYS}} - (t_{\text{RTC}} + t_{\Delta}) . \end{aligned} \quad (6.3)$$

Figure 6.16 shows the measured timing data of the RTC and the PLL output. It can clearly be seen that the PLL output is approaching the interpolated RTC time.

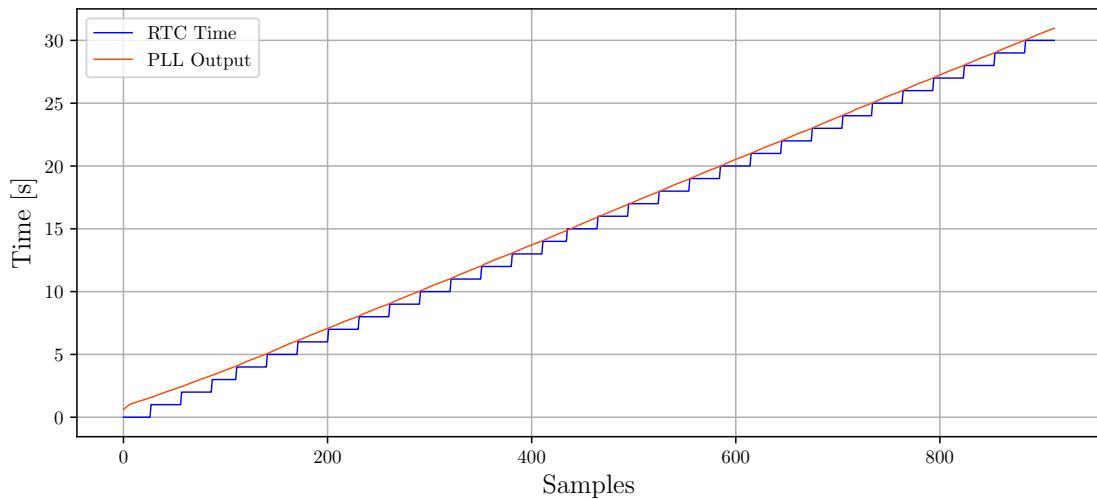


Figure 6.16: Measured Timing Data of the RTC and PLL Output

6.4.4 Remote Configuration

The firmware includes an HTTP server on port 6667 that offers access to a JSON file containing key system information. This file, essential for remote management, details parameters such as the device's position, orientation, and status. To retrieve this JSON file, a simple HTTP GET request is made to the server's root URL. This should be done periodically to ensure the most up-to-date information. Table 6.3 provides an overview of all data fields and their purpose.

Data Field	Description	Data Type	Unit	Example Value
device_firmware_version	Firmware Version	String	-	"V0.1"
device_firmware_build	Firmware Build Date	String	-	"240110"
device_cpu_frequency	CPU Frequency	Integer	Hz	912000000
device_cpu_temperature	CPU Temperature	Float	°C	36.5
device_operating_time	Device Operating Time	Integer	s	15615
device_system_warning	System Warning Status	Boolean	-	false
ethernet_mac	MAC Address	String	-	"48:53:52:20:3C:33"
ethernet_ip	IP Address	String	-	"192.168.1.10"
streaming_state	Streaming State	Boolean	-	true
streaming_speed	Streaming Speed	Float	Mbit/s	22.21
streaming_buffer	Streaming Buffer Fill Level	Float	%	15.1
sensor_heading	Device Heading	Float	°	270.0
sensor_pitch	Device Pitch	Float	°	5.2
sensor_roll	Device Roll	Float	°	2.4
sensor_temperature	Ambient Temperature	Float	°C	22.3
sensor_pressure	Ambient Pressure	Float	hPa	1013.2
sensor_altitude	Device Altitude	Float	m (MSL)	434.5
sensor_angle	Arm Angle	Float	°	45.7
sensor_magnet_detected	Magnet Detection Status	Boolean	-	true
sensor_magnet_too_weak	Magnet Too Weak Status	Boolean	-	false
sensor_magnet_too_strong	Magnet Too Strong Status	Boolean	-	false
gnss_latitude	GNSS Latitude	Float	°	40.7128
gnss_longitude	GNSS Longitude	Float	°	-74.0060
gnss_altitude	GNSS Altitude	Float	m (MSL)	344.8
gnss_magnetic_declination	GNSS Magnetic Declination	Float	°	-5.0
gnss_satellite_count	GNSS Satellite Count	Integer	-	8
gnss_fix	GNSS Fix Status	Boolean	-	true
gnss_fix_type	GNSS Fix Type	Integer	-	3
gnss_time_valid	GNSS Time Validity	Boolean	-	true

Table 6.3: Device Data JSON File

In addition to providing system information, the device is designed to receive commands from the host. This is achieved by sending a JSON file to the web server via a POST request. As of now, the system demonstrates this functionality with a single implemented command ("clear_warning"). While currently limited to this one command, plans include introducing further data fields, such as RTK coefficients. Table 6.4 provides an overview of all control commands and their purpose.

Data Field	Description	Data Type	Unit	Example Value
clear_warning	Clears the warning status	Boolean	-	true
gnss_coefficient_x	GNSS RTK Coefficients (not implemented)	Float	-	-

Table 6.4: Command JSON File

6.4.5 Sensor Calibration

The calibration of the magnetometer involves a customized approach, building upon the *Motion Sensor Calibration Tool* originally developed by *PJRC*. This tool, initially designed for external use on a PC, has been adapted to run directly on the *Teensy 4.1*. The fundamental principle of the algorithm is to collect data points while the sensor undergoes rotation in various directions. Ideally, these points, when plotted in 3D space, should form a perfect sphere centered at the origin.

In practice, due to hard and soft magnetic offsets, the resulting shape deviates from a perfect sphere, forming an ellipsoid that is both deformed and offset from the origin. The transformation caused by these offsets can be mathematically expressed as

$$\mathbf{m}_{\text{cal}} = \mathbf{A} \cdot \left(\begin{bmatrix} \tilde{m}_x \\ \tilde{m}_y \\ \tilde{m}_z \end{bmatrix} - \mathbf{b} \right) \quad (6.4)$$

where \mathbf{b} represents the hard-iron offsets and \mathbf{A} expresses the soft-iron coefficients as a 3x3 matrix.

To address this, *NXP* has developed an advanced algorithm that accurately fits this ellipsoid using a least squares approach, complemented by outlier detection. The intricacies of this algorithm are thoroughly explained in a specific application note provided by *NXP* [4].

Once the calibration process is successfully completed, the resulting coefficients are stored in the *Teensy 4.1*'s flash memory. These saved coefficients are then automatically recalled and utilized each time the system is rebooted.

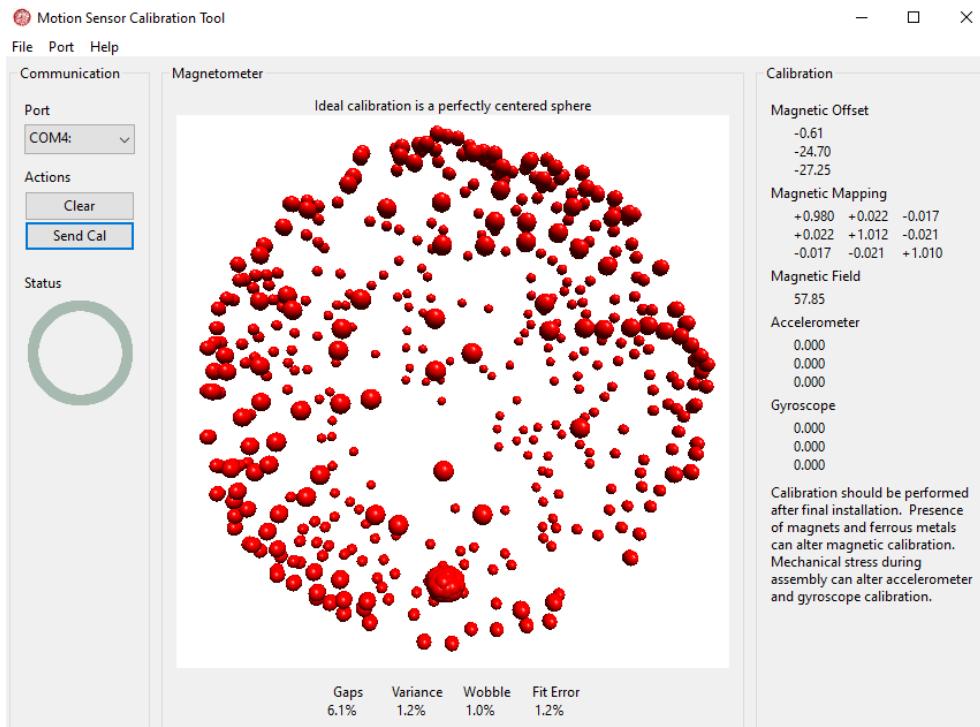


Figure 6.17: Motion Sensor Calibration Tool by PJRC

6.4.6 Human Machine Interface (HMI)

The Human Machine Interface (HMI) of the mainboard consists of 17 RGB LEDs, with one specifically dedicated as a Status LED. To provide acoustic feedback, a buzzer is integrated into the design. This buzzer is particularly useful for alerting users to specific events, such as an audio buffer overflow, by playing a distinctive beep sequence. Table 6.5 provides an overview of the LED indication patterns for the various system states.

System Status	Blink Pattern	LED Color	Streaming State	GNSS Fix
Running	1 Pulse/Second	White	No	No
Running	2 Pulse/Second	White	Yes	No
Running	1 Pulse/Second	Green	No	Yes
Running	2 Pulse/Second	Green	Yes	Yes
Warning	2 Hz Blinking	Yellow	-	-
Error	2 Hz Blinking	Red	-	-

Table 6.5: System Status LED Indication

6.4.7 Graphical User Interface (GUI)

The GUI provides a user-friendly interface for configuring the device and monitoring its status. It is based on the acquisition system's GUI and makes again use of the LVGL framework.

6.4.8 GUI Pages

In the following sections, the different GUI pages are explained in detail. Navigating between the pages is done by clicking on the corresponding icon in the home menu. To return to the home menu, the sub-page header bar with the arrow symbol can be pressed. Figure 6.18 shows an overview of all GUI pages and how navigation between them is done.

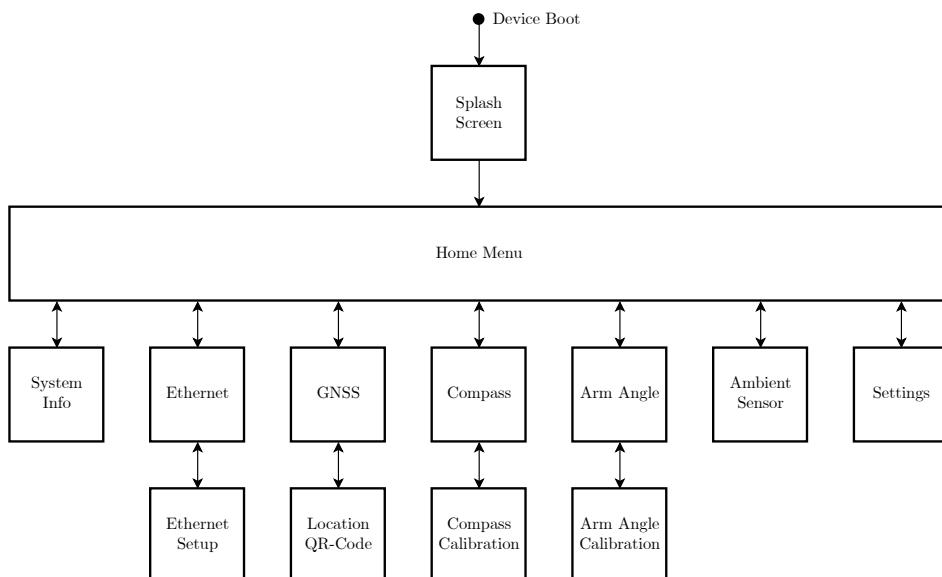


Figure 6.18: GUI Pages Overview



Figure 6.19: Splash Screen

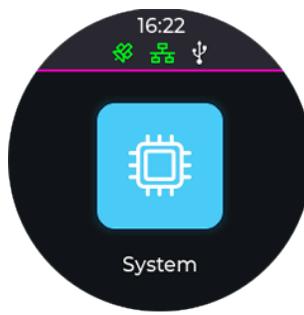


Figure 6.20: Home Menu



Figure 6.21: System Information

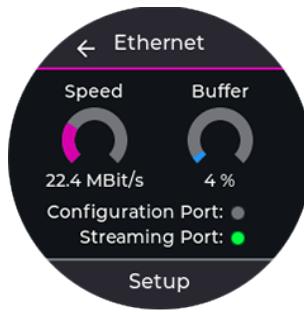


Figure 6.22: Ethernet

Splash Screen

When the system is powered on, the splash screen is displayed for 5 seconds. On the bottom of the screen, the current firmware version and build date is displayed. After the system has booted, the home menu is displayed.

Home Menu

The home menu presents an intuitive way of navigating through the different submenus. On the top of the screen, the current time is displayed. A dedicated icon for the GNSS, ethernet and USB interface is shown. Per default, the icons are grayed out. When the GNSS location is valid (fix), the icon turns green. When the device is connected to the network but is not yet streaming, the icon is filled white. As soon as the audio stream is active, the icon turns green. The USB icon is filled white when a device is connected to the USB port. As soon as the virtual COM port is opened, the icon turns green.

System Information

The system information page displays important device information such as the firmware version, firmware build date, etherent MAC address, CPU frequency and temperature, as well as the operating time.

Ethernet

The ethernet page displays the current connection state of the streaming and configuration port. While a connection is established to either of the ports, the corresponding gray circle is filled green. Two gauge bars display the current streaming speed and the fill level of the streaming buffer. In normal operation, the indicated streaming speed should be around 22 Mbit/s and the buffer fill level near 0 %.



Figure 6.23: Ethernet Setup



Figure 6.24: GNSS



Figure 6.25: Location QR-Code



Figure 6.26: Compass

Ethernet Setup

The ethernet setup page allows for configuring the IP address of the device. Below the IP address, the hard-coded streaming port (6666) and configuration port (6667) is displayed. When the IP address is changed, the user can either confirm the change or discard it by returning to the previous page. When a new IP address is confirmed, the device will instantly change its IP address and restart the servers.

GNSS

The GNSS page displays the current status of the GNSS module. When the Fix Status is valid (2D-Fix or 3D-Fix), the latitude, longitude and altitude is displayed. In addition, the QR-Code button gets enabled, which allows for displaying the current location in a QR-Code. The coordinates are displayed in degrees, minutes and seconds. The altitude is displayed in meters above sea level (MSL). When the time is fully resolved, it is displayed in the UTC+0 format. While the Fix Status is invalid, all fields are grayed out.

GNSS Location QR-Code

The GNSS location QR-Code page displays a QR-Code containing a google maps link to the current location. When the QR-Code is scanned with a smartphone, the web browser will immediately redirect the user to the google maps app. The URL is formed as follows:

`google.com/maps/place/<latitude>,<longitude>`

For example, the URL directs to:

`google.com/maps/place/47.222400,8.816460`

Compass

The compass page displays the current heading and leveling of the device. The heading is indicated by a purple arrow, pointing towards geographic north. When the needle points exactly towards the top of the screen it turns green, meaning the device is facing north. The circular graphic in the centre of the screen shows the current leveling of the device. When the device is perfectly leveled, the circle turns green. When the device is tilted, the circle turns red and the tilt angle is displayed in degrees (pitch and roll). To calibrate the built-in magnetometer, the user can press the calibrate button.

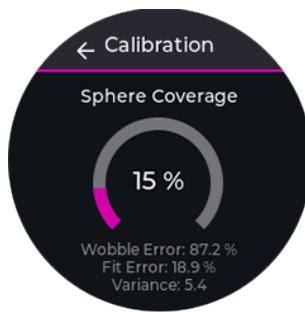


Figure 6.27: Compass Calib.



Figure 6.28: Arm Angle

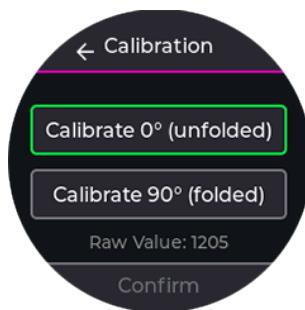


Figure 6.29: Arm Angle Calib.

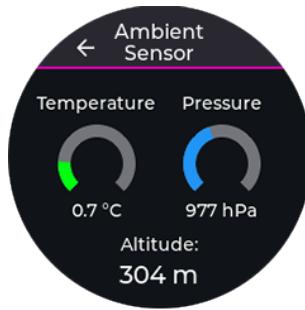


Figure 6.30: Ambient Sensor

Compass Calibration

The compass calibration page displays the current status of the magnetometer calibration. By entering this page, the calibration process starts immediately. To calibrate the magnetometer, the device has to be rotated around all three axes. This process takes around 30 seconds. As soon as the sphere coverage indicator reaches 100 %, the calibration is finished, a success message is displayed and the buzzer plays a short melody. The calibration process can be aborted at any time by returning to the previous page (the calibration progress is not saved).

Arm Angle

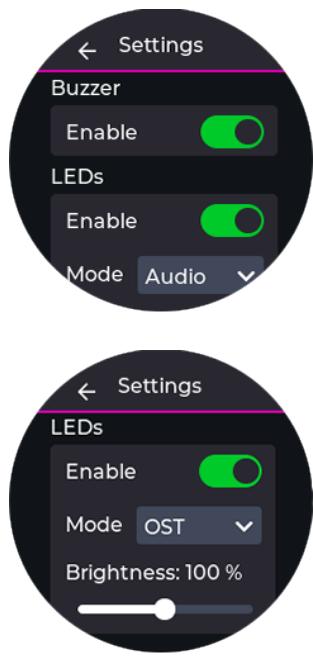
The arm angle page displays the current angle of the microphone array arms in degree. When the arms are fully unfolded (horizontal) the angle is 0.0°. When the arms are fully folded towards the centre pole the angle is 90.0°. Below the angle indicator, the current status of the magnet detection is displayed. When the magnet is detected, the status indicator turns green. If the magnet is too weak or too strong, the corresponding status indicator turns yellow. Otherwise, the status indicators are grayed out.

Arm Angle

The arm angle calibration page lets the user calibrate the angle sensor. To calibrate the angle sensor, the arms have to be fully unfolded (horizontal). Then the upper calibrate button *Calibrate 0° (unfolded)* has to be pressed. Next, the arms have to be fully folded towards the centre pole. Then the lower calibrate button *Calibrate 90° (folded)* has to be pressed. To confirm the calibration, the user has to press the *Confirm* button.

Ambient Sensor

The ambient sensor page displays the current ambient temperature in degree Celsius and the ambient pressure in hectopascal. A gauge bar visualizes both values. Based on the current ambient pressure, the altitude above sea level is calculated and displayed in meters. Note that the altitude is only an approximation and can be inaccurate. It is strongly influenced by the current weather conditions.



Settings

The settings page allows the user to configure the device. First, the buzzer can be enabled or disabled. Second, the LEDs can be permanently enabled or disabled. The mode selection provides a selection of different animations for the LEDs. Currently, there are two animation types implemented:

- **Audio:** The LEDs are controlled by the audio level of the microphone array.
- **OST:** The LEDs show a fluent animation in the OST color scheme.

Below the mode settings, the LED brightness can be adjusted. All settings are saved in the internal flash memory of the device and are restored after a reboot.

Figure 6.31: Settings

6.5 Software Design

The software is made to perform the tracking of the sound sources. It runs on a external device, in this case a *Dell Latitude 5310* Laptop with *Ubuntu* installed.

6.5.1 Audio Interface

The Audio Interface provides the application with the audio data from either the TCP stream or a pre recorded WAV File. In both methods, the data is stored into an adapted ring buffer until it is used by the main application.

Ring Buffer

A slightly modified ring buffer has been implemented to buffer the audio data. It is thread save since the storing and reading functions are called from different threads. The main difference to a conventional ring buffer is, that the data is not read from the tail, but rather from the head. This allows the application to always read the last N samples while the data in the buffer does not get split up.

Additionally the ring buffer is able to write newly input data into a WAV file.

TCP Stream

The TCP streamer allows to stream the audio directly via TCP form the microphone array. The streaming runs via a socket to which the application connects to. Using the control sequence the received data is preprocessed and then stored into the buffer.

WAV Stream

The WAV stream is made with the *PyAudio* and the *WAVE* library from python. With the *WAVE* library a .wav file is opened and then played back with *PyAudio*. Via a callback in the played audio is also stored into the ring buffer. On *Windows* and *macOS* *PyAudio* could not stream a 32 channel WAV file. Hence the application needs to be run on *Ubuntu*. Other Linux distributions may also work, but were not tested during this thesis.

6.5.2 Communication

To retrieve the sensor data from the microphone array a communicator class was created. It continuously reads the JSON containing various data via a POST request from the microphone array.

6.5.3 Sound Source Tracking

The sound source tracking consists of three parts, the beamforming, peak detector and Kalman Tracker.

Beamforming

First a data block is ran through the beamforming filter bank. Contrary to prior beamforming grid, a pseudo uniformly sampled semisphere is used to define the steering angles. During testing, a sample size of 1700 points, which gives an angle resolution of $\approx 2^\circ$, was used. The number can be adjusted if higher resolution is necessary, however

the computation time increases with a finer resolution. With these samples and the current array arm angle, the filter bank can be calculated. The beamforming used is the Delay and Sum beamforming, which gives a power level for each steering direction.

Peak Detection

To detect where possible sound source are, a peak detector was implemented. If only one source needs to be tracked, a simple argmax on the beamforming response is sufficient.

For the case of multiple sources a more elaborate peak detector is used. To simplify the peak detector, the beamformer response, which is defined on a sphere is mapped onto a flat surface with

$$x = \theta \cos \varphi \quad (6.5)$$

$$y = \theta \sin \varphi. \quad (6.6)$$

This mapping is later also used for the Kalman tracker. The mapped response is then converted to an image with linear interpolation. This image allows the use of a 2D peak detector like the one described in [6]. The basic function is to use grayscale dilation to find local maxima and then filter out larger plateaus with a median blur filter. The resulting peaks can be further filtered by minimal height and minimal ratio to the maximum value. Other filters may be useful in the future, but were not implemented in this version.

Kalman Tracker

The found peaks are given to the Kalman Tracker. With the Hungarian method the peaks are assigned to their corresponding trackers. If a existing tracker does not correspond to a peak, the Kalman Filter can predict it's new position. By default it makes ten such blind predictions until the track is discarded.

6.5.4 GUI

To visualize the beamforming responses and control the tracker a web based GUI with Plotly and Dash was created. Figure 6.32 shows a screenshot of the GUI while tracking is performed on a pre recorder WAV file. With the GUI and the current tracker settings the software runs on 5 FPS.

Connections

On the left side the user can manage the connections to the audio streamer. New connections to a Wav streamer can be established by providing a audio file and an array configuration. To open a TCP connection, the IP address and port of the array is needed. Connections can also be closed with the *Disconnect* button.

Open connections are listed below and via a dropdown list, the displayed connection can be selected.

The streamed audio data can be recorded to a WAV file with the *Start Recording* button.

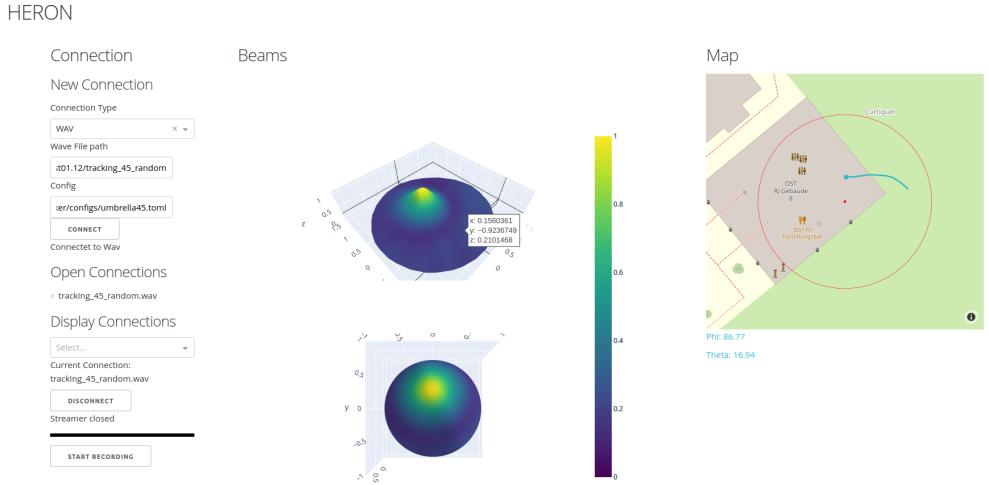


Figure 6.32: Screenshot of the GUI.

Beamforming

The beamforming response is shown in two different ways. The top plot shows the mapped beamforming response in a surface plot, where the z-axis represents the beam-formed power. To increase readability the power is normalized between zero and one. The bottom plot displays the sampled semisphere where the color represents the beam-formed power.

Map

The map shows the current position of the array. If no GNSS data is available the default position is by the building 8 at OST in Rapperswil. The tracks are also displayed on the map. Since with one array the exact position of the sound source can not be determined, the directions are mapped onto the map with 6.6. To make the circle bigger, θ is multiplied by 30 before mapping it. The red circle indicates a $\theta = 90^\circ$. By using the compass data, the alignment of the array does not matter for the tracks to be displayed in the right compass direction.

If the external system does not have an internet connection to display the map, the application can be run in an offline mode.

7

Measurements

7.1 Overview

The measurements were performed in an open field with a *DJI Mavic Pro* drone. A measurement location with low ambient noise was chosen. However, a nearby farm was sometimes audible in the recordings.

7.2 Stationary Drone

In the stationary drone test, the drone flew to different points, where it then stayed still. For the test five points in different directions were chosen. Table 7.1 shows these five points. These points were measured using the GPS of the drone so their position are not known with great precision. At each of the five points the array was tested with five different arm angles.

	φ_s	θ_s	r_s
P_1	-120°	70°	31 m
P_2	-158°	50°	17 m
P_3	-158°	25°	32 m
P_4	-158°	15°	52 m
P_5	0°	0°	50 m

Table 7.1: Positions of the test points.

The measurements include the calculated DOA and the two metrics introduced in section 5.2. Tables 7.2 and 7.3 show the signed error for the two angles φ and θ . The measurement of φ is relatively independent of the array arm angle. However for positions with a high θ_s the error in θ increases when the arm angle increases. Since this was not the case in the simulation, it is suspected that these errors come from physical effects of the sound waves in the array and the directionality of the used microphones.

The mainlobe areas from the measurements are as expected from the simulations. It is however observed that the mainlobe area at P_1 , P_3 and P_4 is better when the arm is

Arm angle	P_1	P_2	P_3	P_4	P_5
0°	-2°	+5°	+2°	+3°	0°
15°	-2°	-1°	+0°	+2°	+0°
30°	-2°	+6°	+1°	+4°	+0°
45°	-1°	+6°	+2°	+5°	+0°
60°	-2°	+6°	+4°	N/A	N/A

Table 7.2: Signed error in φ . The errors for P_5 are 0 due to its indefiniteness when $\theta = 0^\circ$.

Arm angle	P_1	P_2	P_3	P_4	P_5
0°	+2°	+5°	+0°	+0°	+1°
15°	+5°	+5°	+0°	+0°	+1°
30°	+10°	+9°	+1°	+0°	+1°
45°	+20°	+10°	+1°	+0°	+1°
60°	+20°	+11°	+2°	N/A	N/A

Table 7.3: Signed error in θ .

angled. With the current measurements it can not be stated if this results from noise or from other effects.

The PAP ratio is significantly lower than in the simulated cases. Since the measurement was taken outdoors with background noise, this is no surprise. Nonetheless the PAP ratio follows similar trends when θ_s or the arm angle is changed.

Arm angle	P_1	P_2	P_3	P_4	P_5
0°	4.85%	4.37%	2.42%	2.90%	1.45%
15°	4.24%	4.52%	2.01%	1.82%	2.00%
30°	4.15%	6.71%	2.92%	2.30%	1.82%
45°	4.10%	6.62%	4.62%	2.95%	2.31%
60°	4.26%	6.76%	5.16%	N/A	N/A

Table 7.4: Mainlobe area ratio at the testpoints.

Arm angle	P_1	P_2	P_3	P_4	P_5
0°	7.17	10.20	12.23	7.65	16.82
15°	9.30	9.95	11.81	13.94	14.51
30°	9.45	8.35	10.61	11.30	13.45
45°	9.14	7.98	7.74	8.26	11.56
60°	9.68	8.34	5.16	N/A	N/A

Table 7.5: PAP ratio at the testpoints.

7.3 Moving Drone Test

Making a verifiable test for a flying drone was not possible due to a lack of time. To test the localization and tracking of a flying drone qualitative tests were performed. For the test the drone flew along paths in the test environment. Ideally the track created by the drone is parallel to these paths. To check the θ angle, the same track was flew in different height. A comparison of two such flights are shown in Figure 7.1. In Figure 7.1b, it is apparent that when the drone ascended by 10 meters, the measured θ was lower than in Figure 7.1a, as anticipated.



Figure 7.1: Drone flying along the road from west to east in two different elevations. The distance between the red center point and the track is linearly dependent on θ whereas the angle between east and the track represents φ . The thicker point right of the tracks shows the current tracker positions.

7.4 Conclusion

With these measurements it has been shown that the proposed microphone array and sound source tracker are able to localize and track flying drones. To make better quantitative statements, more elaborate tests would be needed.

8

Conclusion

This Master's project delved into the complex task of localizing and tracking drones using a microphone array, exploring both hardware and software aspects to gain new insights and develop a custom-built system. The choice of MEMS microphones proved to be advantageous, as they offer high audio quality and are easy to integrate.

A key strategy was the development of a simulation environment for testing various array geometries. This approach led to the accumulation of in-depth knowledge and the identification of an optimized microphone array. This array is distinguished by its unique cone-shaped, three-dimensional structure with adjustable arms.

Through qualitative and quantitative measurements, it has been demonstrated that the proposed system is capable of effectively localizing and tracking drones in open fields. In conclusion, the structured methodology, evolving from research and prototyping to the final product development, proved to be highly successful.

8.1 Continuing Work

Although the final product is equipped with an extensive range of features, there remains room for improvement and the addition of new capabilities. One significant aspect of this system is its design, structured to accommodate future enhancements. A primary opportunity for advancement lies in incorporating more microphone arrays into the system, which could transform it into a more comprehensive drone detection and tracking solution.

Possible future developments include:

- Adding more microphone arrays to the system
- Integrating a camera to visually show the drone's position
- Improve computation time of the beamforming algorithm
- More elaborate testing
- Classifying sound sources to distinguish between drones and other sources
- Testing new algorithms based on TDOA between multiple arrays
- Development and evaluation of a beamforming based tracker
- Utilizing GNSS RTK compensation for more precise positioning
- Building a stand-alone, battery-powered base station for greater mobility

8.2 Personal Reflections

Florian Baumgartner

Reflecting on this Master's project, I gained immense knowledge in audio processing, MEMS microphones, low-level networking, and GNSS RTK positioning. Developing a fully functional microphone array in just four and a half months was a challenging yet rewarding experience that I'm very proud of. Working with Alain Keller was a great experience, as we each brought different skills that complemented each other, making for a productive collaboration. Although we faced some time management challenges, we were able to overcome them and deliver a working system. I hope that this thesis will serve as a foundation for future projects that can build upon the work we have done.

Alain Keller

This thesis was enriching for me because it combined the development and integration of algorithms, software and hardware. I learned a lot of new concepts in the realms of signal processing, and software development with Python. Working with Florian Baumgartner gave me many insights on how to tackle a hard and firmware project. Along that I gained new experience in working on a project and also got my learnings on what I could do better in the future. Ultimately I am proud that we were able to fulfill our task and create a working system to track drones.

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. Furthermore, we would like to disclose that we have utilized the assistance of ChatGPT, an artificial intelligence language model developed by OpenAI, to improve the clarity and coherence of the text. While ChatGPT has provided suggestions and refinements, the overall content, ideas and analysis presented in this thesis remain our own.

Location, Date

Rapperswil, 24. January 2024



Florian Baumgartner



Alain Keller

A.2 Data Archive

All created files and documents of this project are publicly available on GitHub. An institution called **PA-OST-2023** (<https://github.com/PA-OST-2023>) 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:

heron-administration

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

URL: <https://github.com/PA-OST-2023/heron-administration>

Type: Private

heron-literature

Description: This repository contains all literature used in this project.

URL: <https://github.com/PA-OST-2023/heron-literature>

Type: Private

heron-documentation

Description: This repository contains this document.

URL: <https://github.com/PA-OST-2023/heron-documentation>

Type: Public

heron-hardware

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

URL: <https://github.com/PA-OST-2023/heron-hardware>

Type: Public

heron-firmware

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

URL: <https://github.com/PA-OST-2023/heron-firmware>

Type: Public

heron-simulator

Description: This repository contains the simulator source code written in Python.

URL: <https://github.com/PA-OST-2023/heron-simulator>

Type: Public

heron-application

Description: This repository contains the application source code written in Python.

URL: <https://github.com/PA-OST-2023/heron-application>

Type: Public

heron-mechanical

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

URL: <https://github.com/PA-OST-2023/heron-mechanical>

Type: Public

heron-bastelstube

Description: This repository contains temporary and experimental files.

URL: <https://github.com/PA-OST-2023/heron-bastelstube>

Type: Private

A.3 Definition of Task



AUFGABENSTELLUNG

Thema: Lokalisierung einer Audioquelle

Studierende: Florian Baumgartner, Alain Keller

Betreuer: Hannes Badertscher

Partner: ICAI

Fachgebiet: Digital Signal Processing

Kurzbeschreibung

Das Ziel dieser Arbeit ist es, ein Gerät zur Lokalisierung von Audioquellen zu entwickeln. Damit soll die Basis für weiterführende Arbeiten zur Lokalisierung und Klassifizierung von Flugobjekten wie Drohnen mittels eines Mikrofon-Arrays gelegt werden. In dieser Arbeit soll eine leistungsfähige Hardware zur Aufnahme von Audiosignalen, sowie eine Echtzeitfähige Software zur Lokalisierung von Quellen entwickelt werden.

Aufgabenstellung

- Einarbeitung in die theoretischen Grundlagen zur Lokalisierung von Audioquellen mittels Beamforming. Recherche zu bestehenden Systemen und deren Aufbau.
- Entwicklung einer einfachen Simulationsumgebung einer Soundquelle und eines Mikrofon-Arrays zur Unterstützung des Hardware-Designs, sowie zum Entwickeln erster Lokalisierungs-Algorithmen.
- Design und Berechnung eines geeigneten Mikrofon-Arrays zur Lokalisierung einer einzelnen Audioquelle mit gegebener Frequenz (5 kHz).
- Entwicklung und Aufbau des designten Mikrofon-Arrays, sowie der zugehörigen Auswertelektronik, des Aufnahmesystems, sowie einer geeigneten Hardware zur genauen Positionsbestimmung als Referenz.
- Entwicklung und Aufbau einer Referenz-Audioquelle mit Hardware zur genauen Positionsbestimmung zur Validierung des Mikrofon-Arrays. Messung und Charakterisierung der Referenz-Audioquelle.
- Aufnahme von mehreren Datensätzen mit dem Mikrofon-Array und der Referenz-Audioquelle zur Implementation und zur Auswertung von Algorithmen.



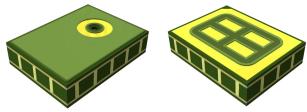
- Entwicklung eines Algorithmus in Python (o. ä.) zur Lokalisierung der Referenz-Audioquelle in der Simulationsumgebung. Erweiterung und Verbesserung des Algorithmus zur Lokalisierung der Referenz-Audioquelle in den gemessenen Daten.
- Implementation einer Software zum Auslesen des Mikrophon-Arrays und zum Ausführen des Lokalisierungs-Algorithmus in Echtzeit auf einem PC oder einem Embedded System.
- Durchführung von Messungen zur Auswertung des Gesamtsystems mit der Referenz-Audioquelle, sowie mit verschiedenen natürlichen Audioquellen wie z. B. einer Drohne.
- Auswertung der Performance und der technischen Möglichkeiten der entwickelten Hardware und Software, Beurteilung der Eignung des Aufbaus zur Detektion von Drohnen.

A.4 Datasheet of MEMS Microphone (MP34DT05TR-A)



MP34DT05-A
 Datasheet

MEMS audio sensor omnidirectional digital microphone



HCLGA - 4LD (3 x 4 x 1 mm)



Features

- Single supply voltage
- Low power consumption
- AOP = 122.5 dB SPL
- 64 dB signal-to-noise ratio
- Omnidirectional sensitivity
- $-26 \text{ dBFS} \pm 3 \text{ dB}$ sensitivity
- PDM output
- HCLGA package
 - Top-port design
 - SMD-compliant
 - EMI-shielded
 - ECOPACK, RoHS, and "Green" compliant

Applications

- Mobile terminals
- Laptop and notebook computers
- Portable media players
- VoIP
- Speech recognition
- A/V eLearning devices
- Gaming and virtual reality input devices
- Digital still and video cameras
- Antitheft systems

Product status link	
MP34DT05-A	
Product summary	
Order code	MP34DT05TR-A
Temperature range [°C]	-40 to +85
Package	HCLGA (3 x 4 x 1 mm) 4LD
Packing	Tape and reel

Description

The MP34DT05-A is an ultra-compact, low-power, omnidirectional, digital MEMS microphone built with a capacitive sensing element and an IC interface.

The sensing element, capable of detecting acoustic waves, is manufactured using a specialized silicon micromachining process dedicated to producing audio sensors.

The IC interface is manufactured using a CMOS process that allows designing a dedicated circuit able to provide a digital signal externally in PDM format.

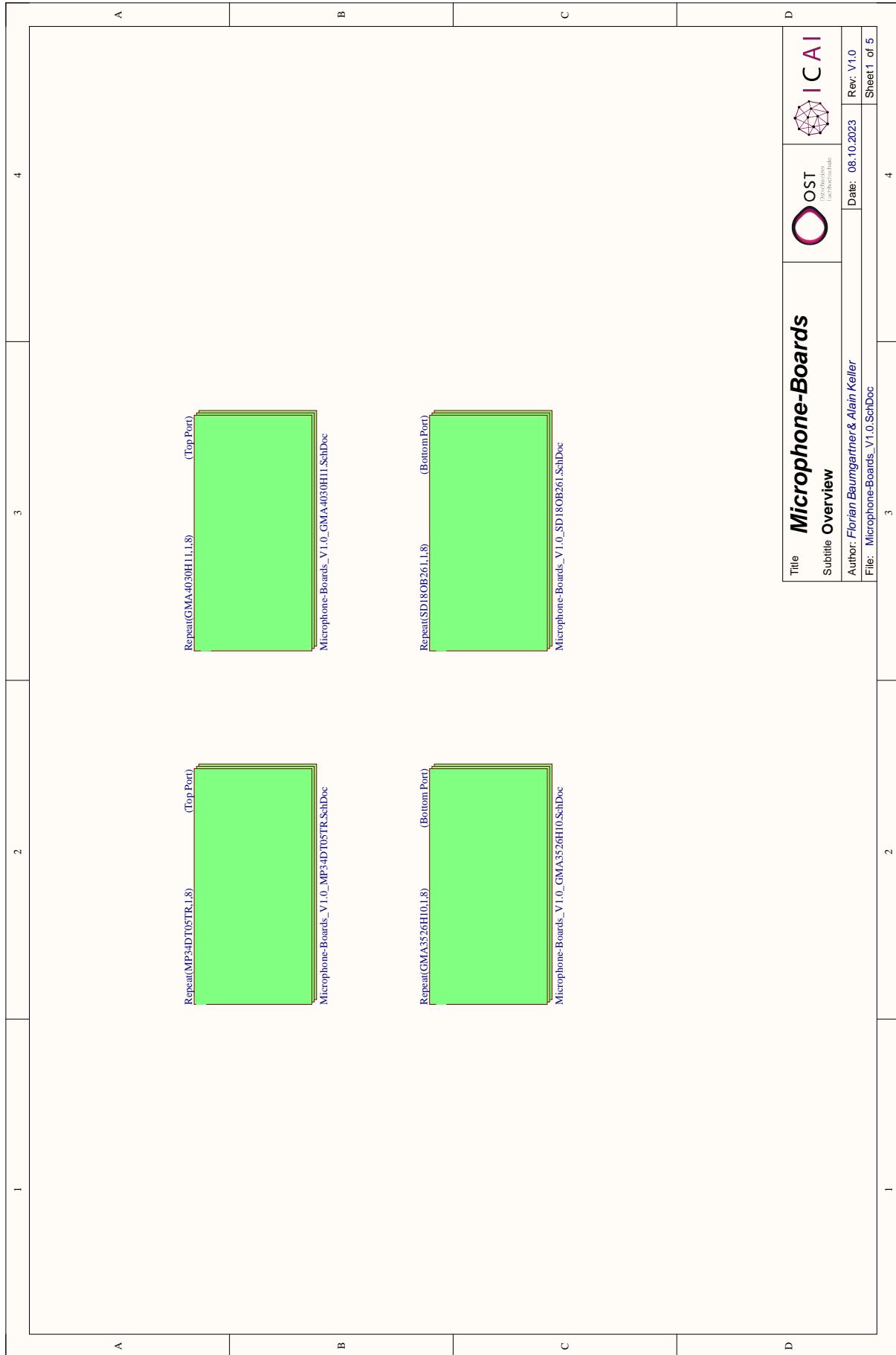
The MP34DT05-A is a low-distortion digital microphone with a 64 dB signal-to-noise ratio and $-26 \text{ dBFS} \pm 3 \text{ dB}$ sensitivity.

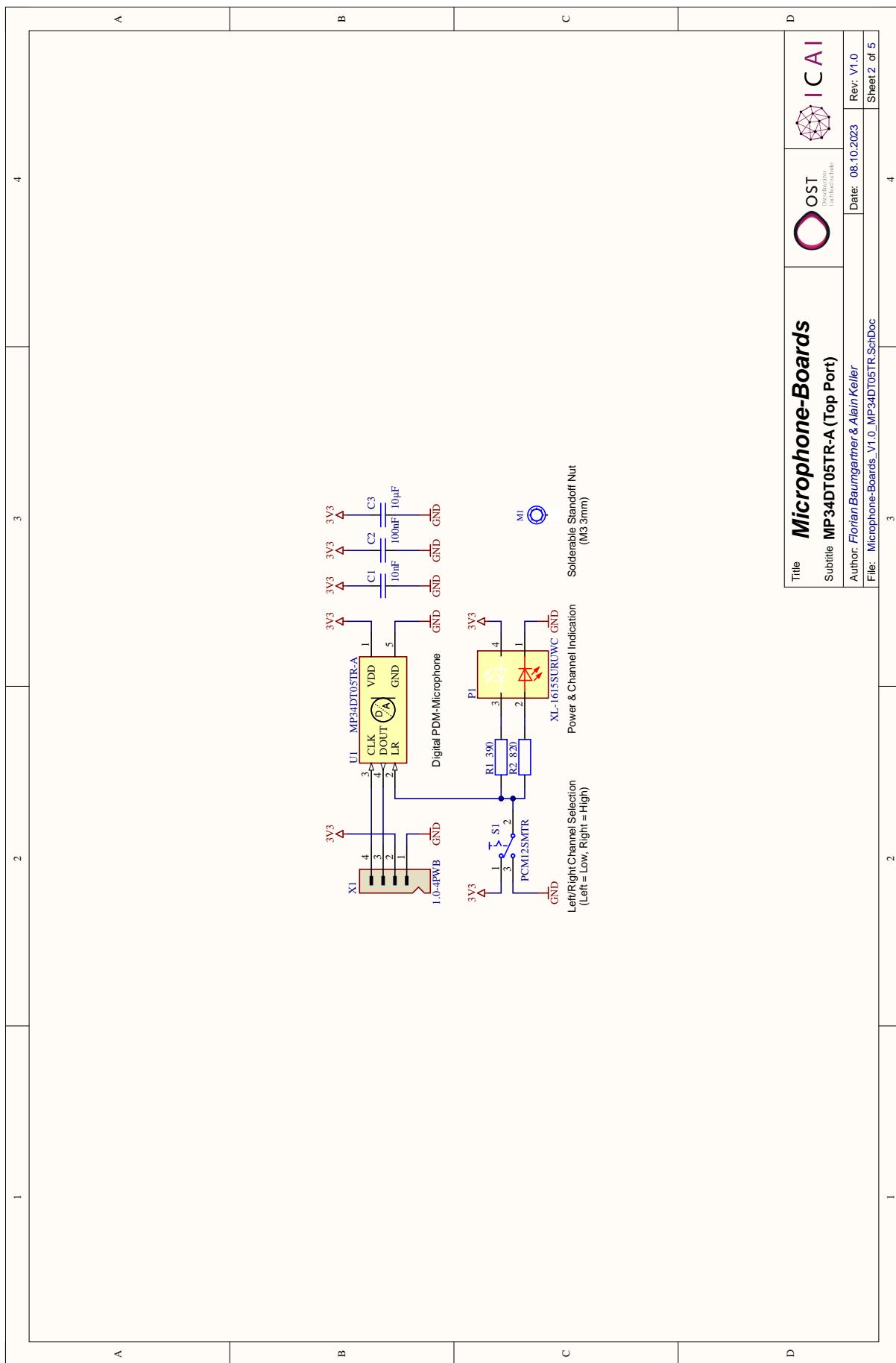
The MP34DT05-A is available in a top-port, SMD-compliant, EMI-shielded package and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

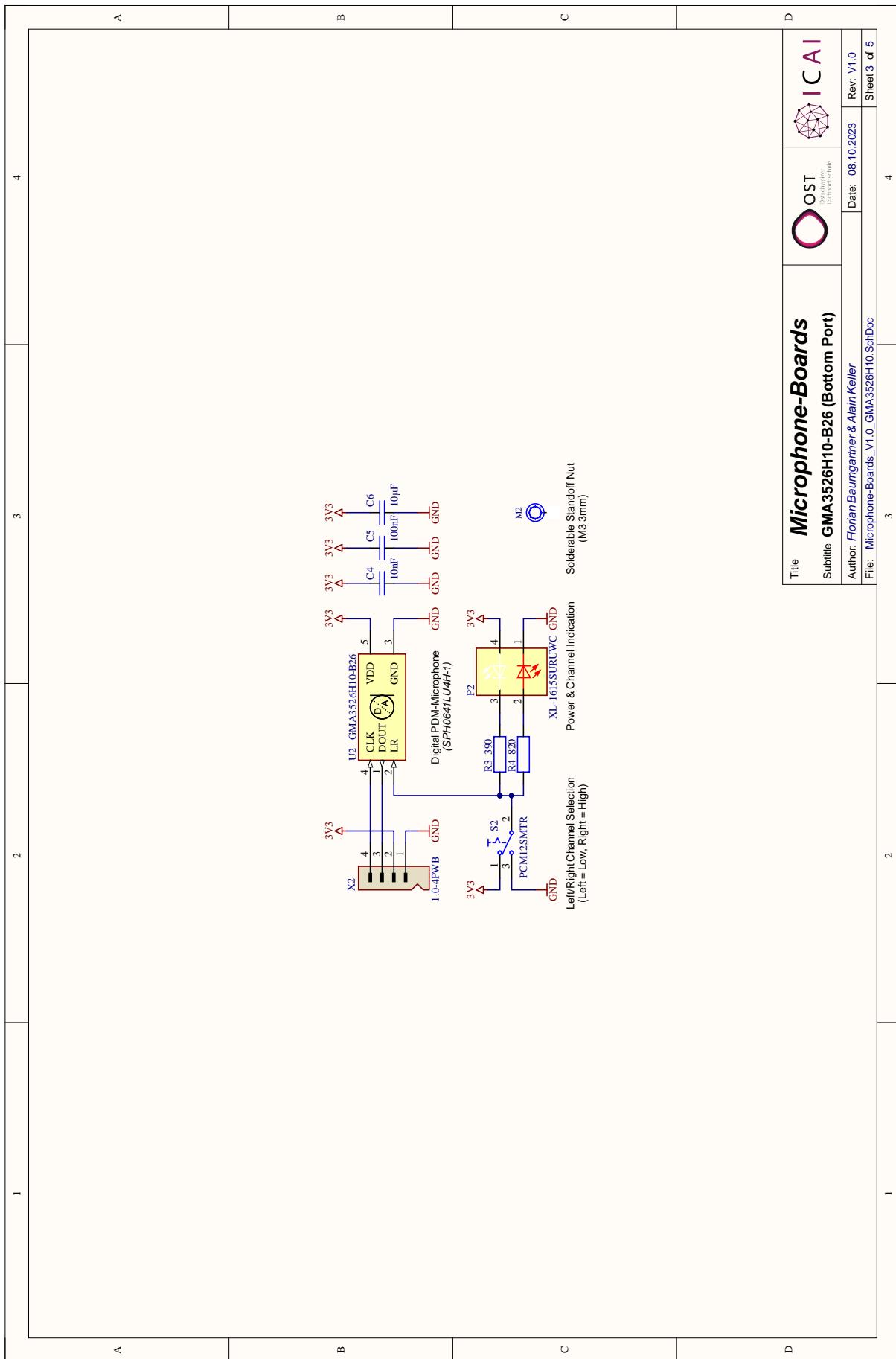
DS12239 - Rev 4 - April 2019
 For further information contact your local STMicroelectronics sales office.

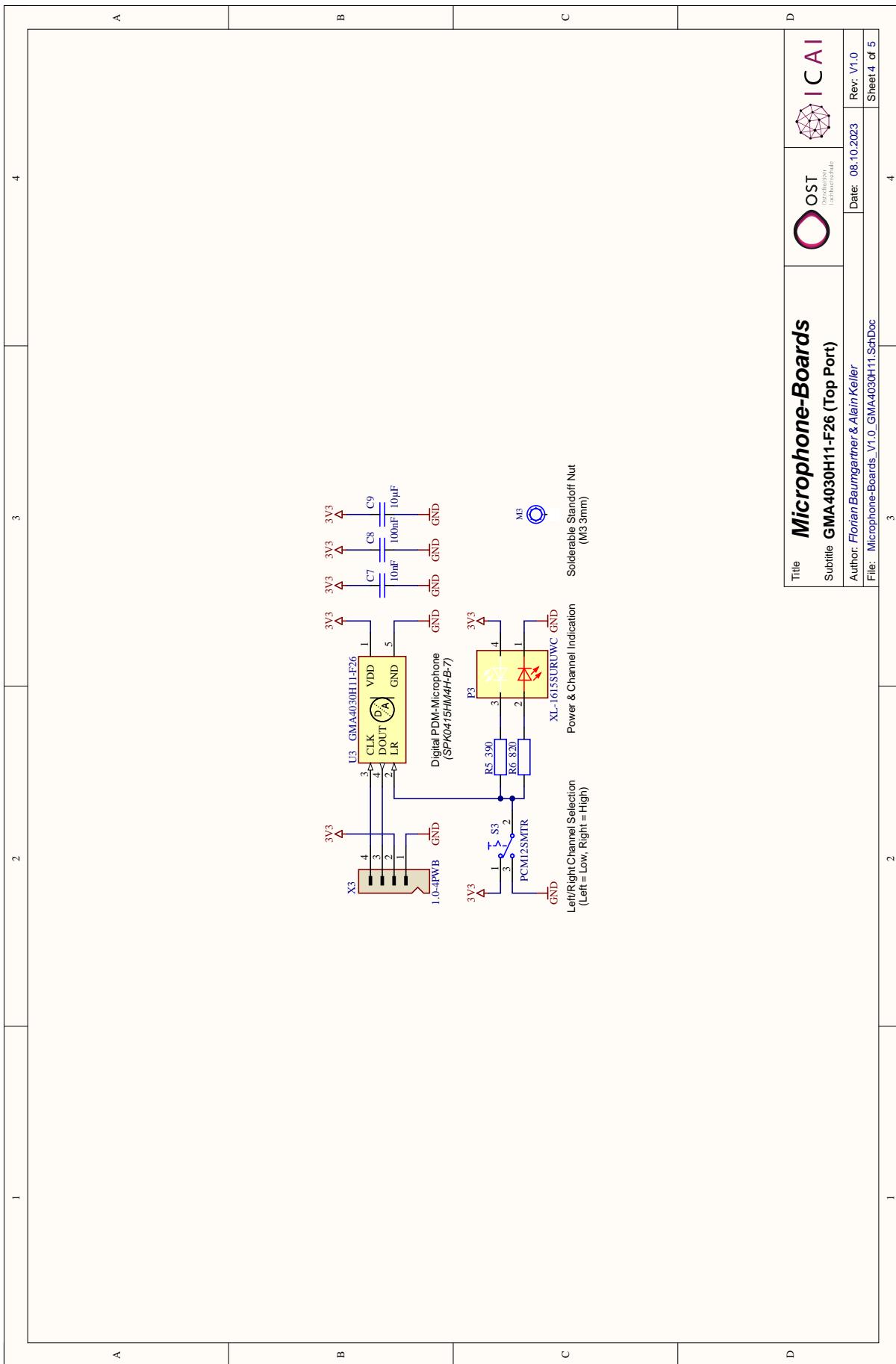
www.st.com

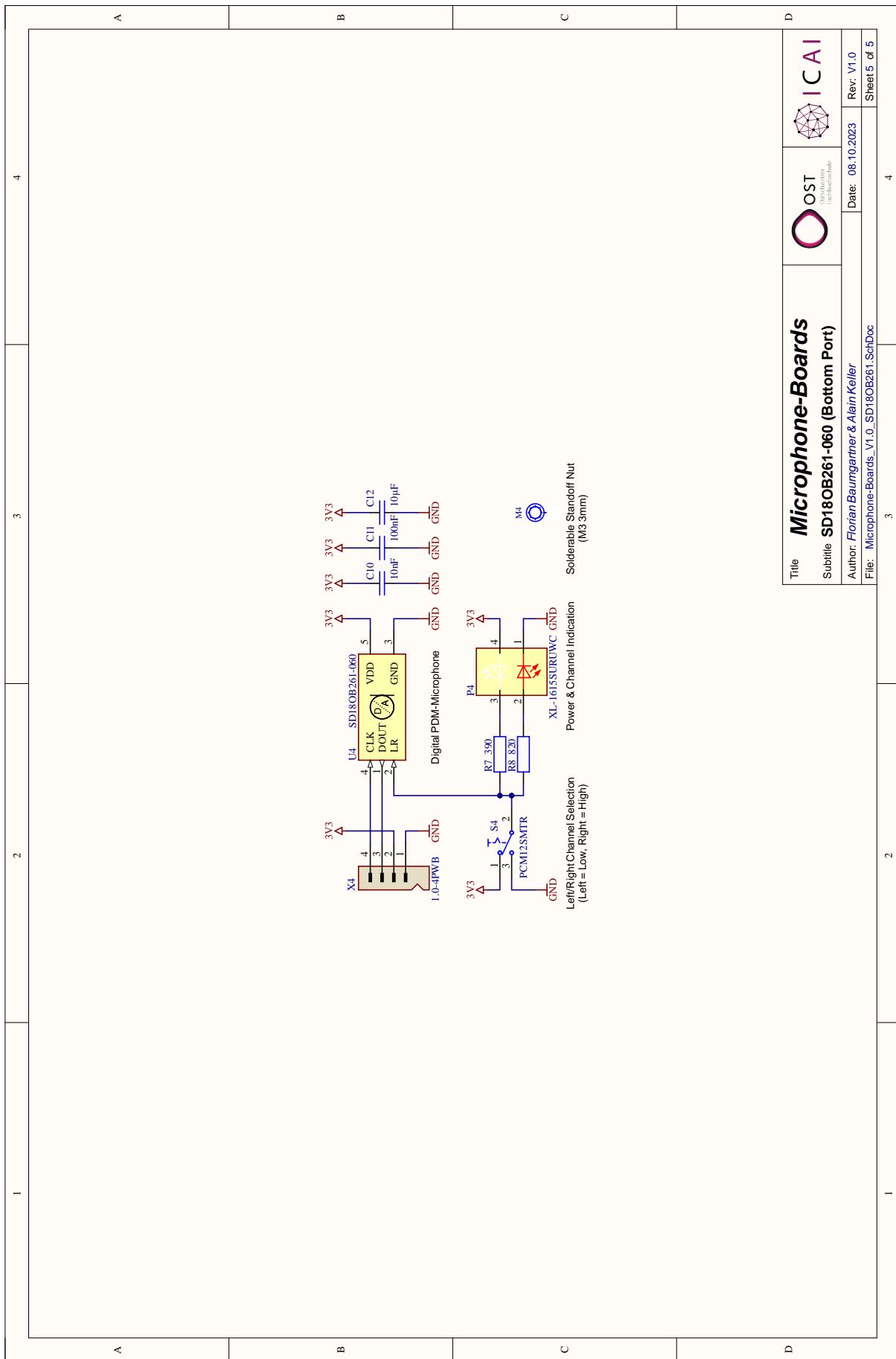
A.5 Microphone-Boards Schematics



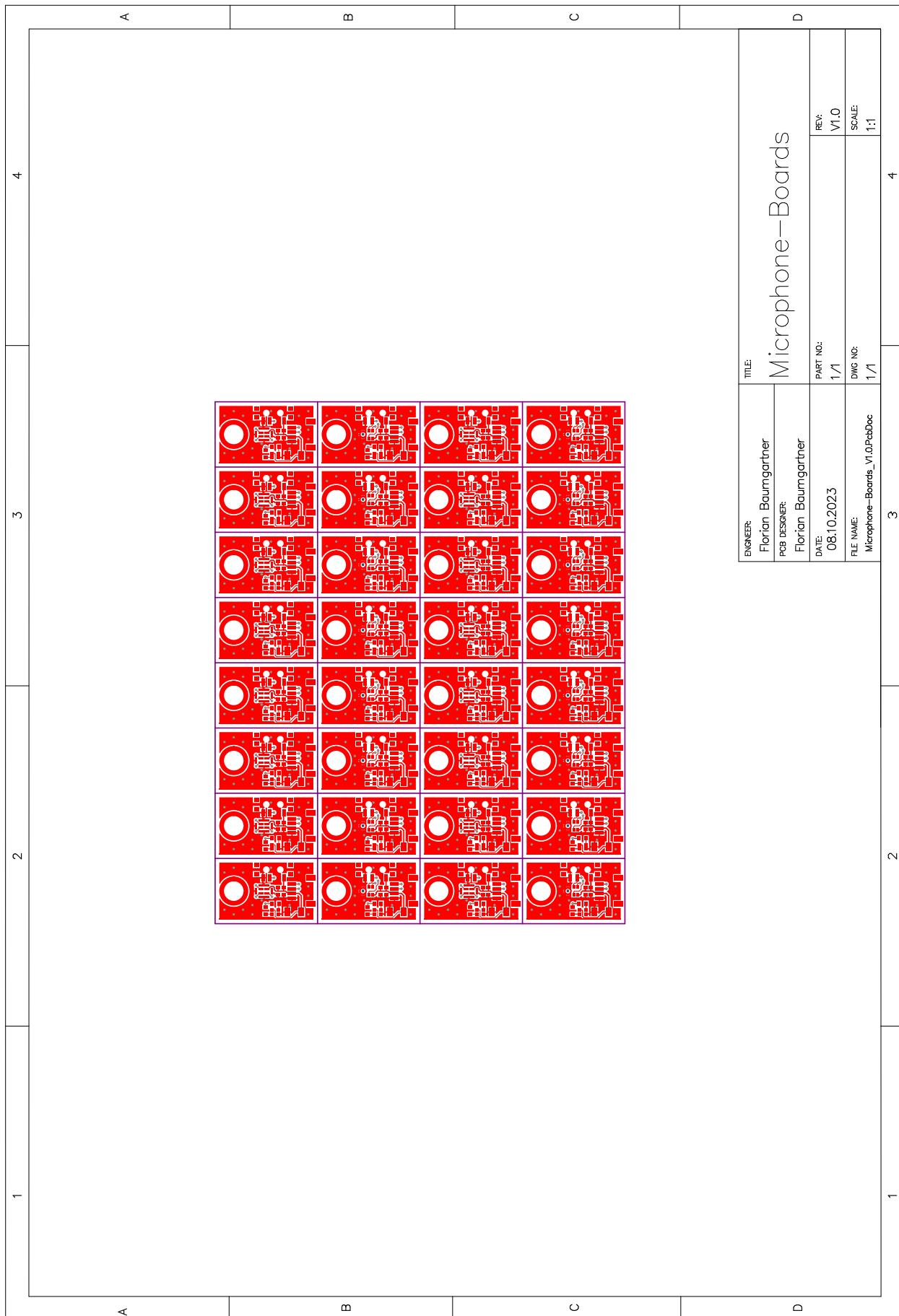




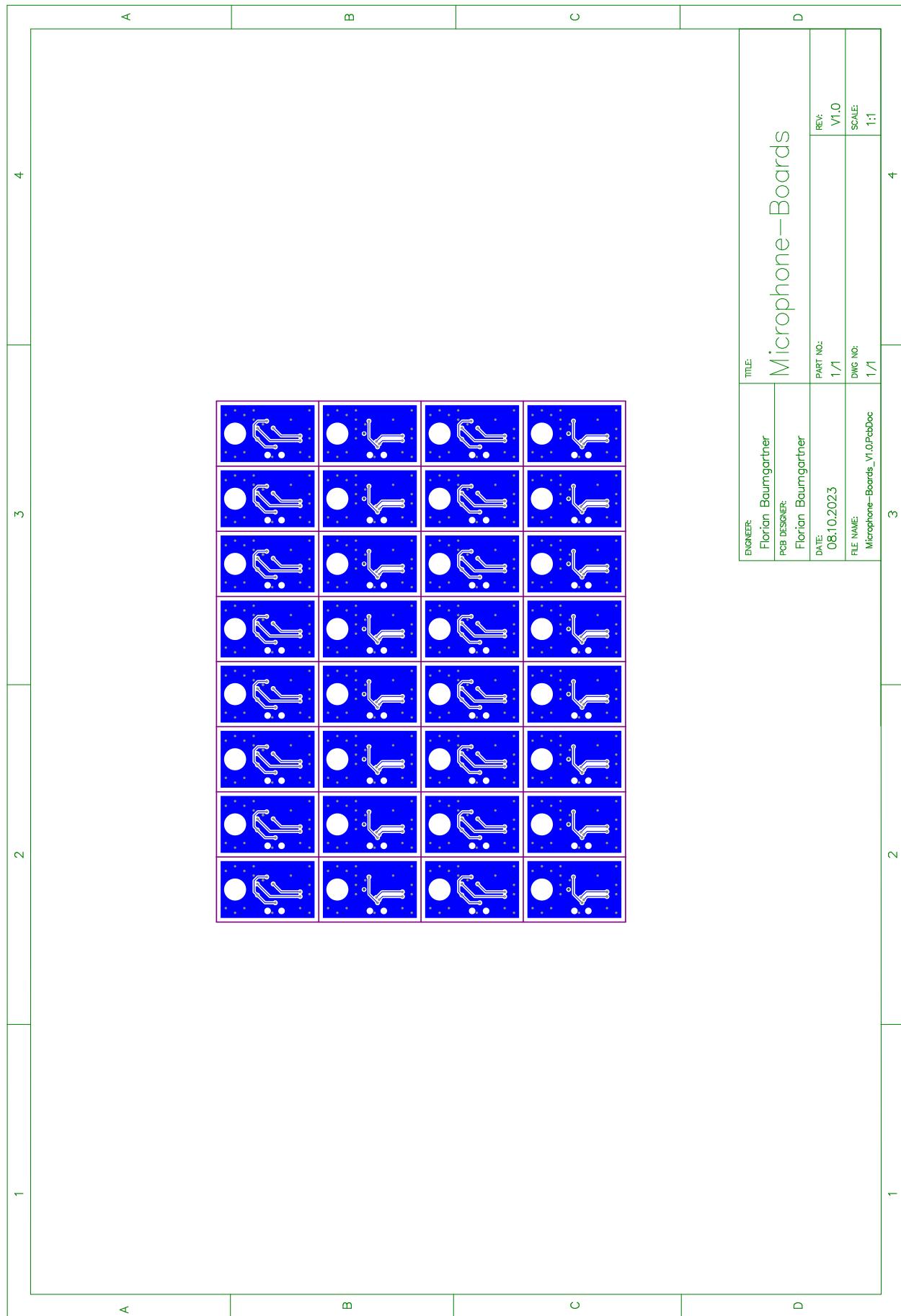




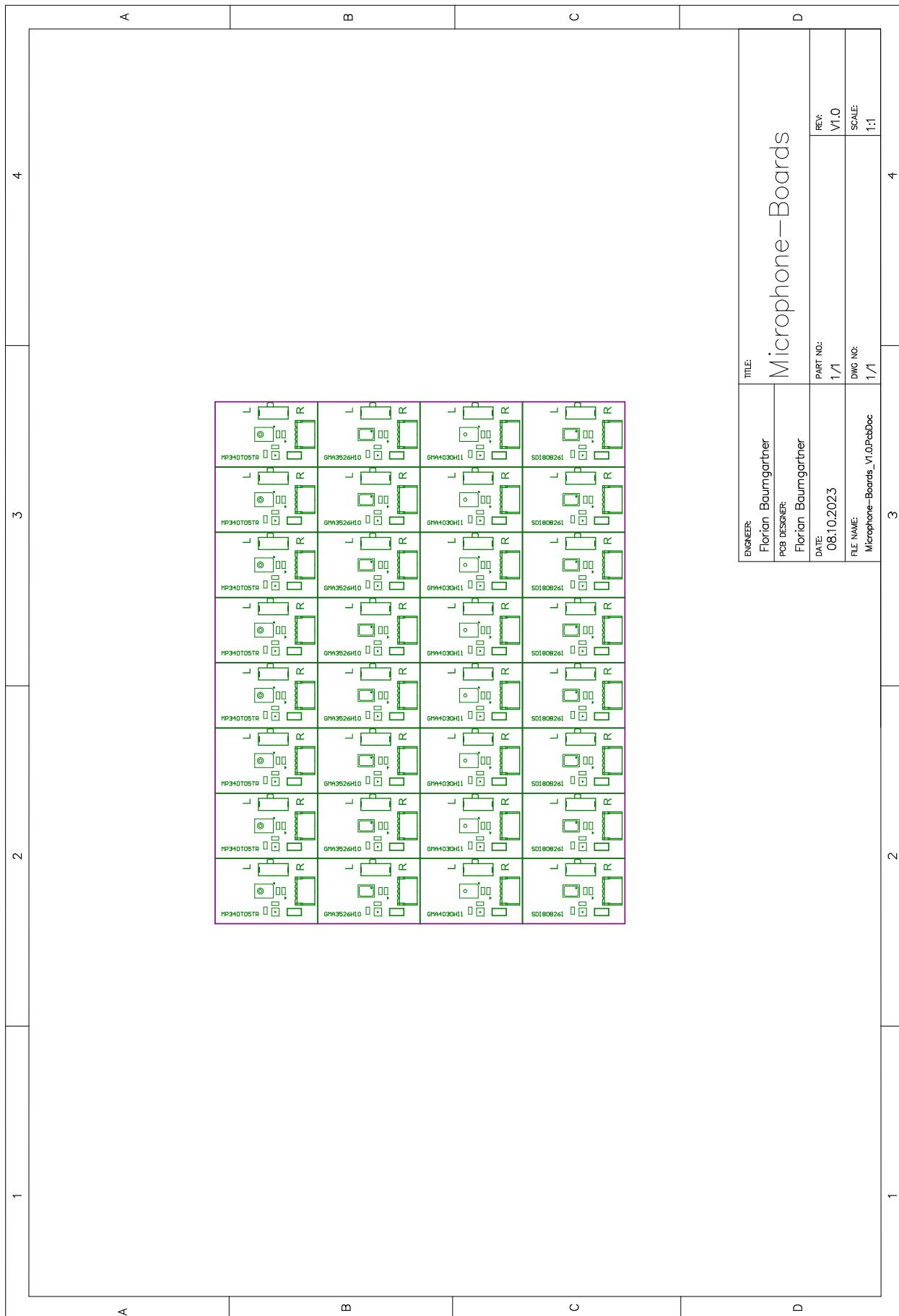
A.6 Microphone-Boards PCB Top-Layer



A.7 Microphone-Boards PCB Bottom-Layer

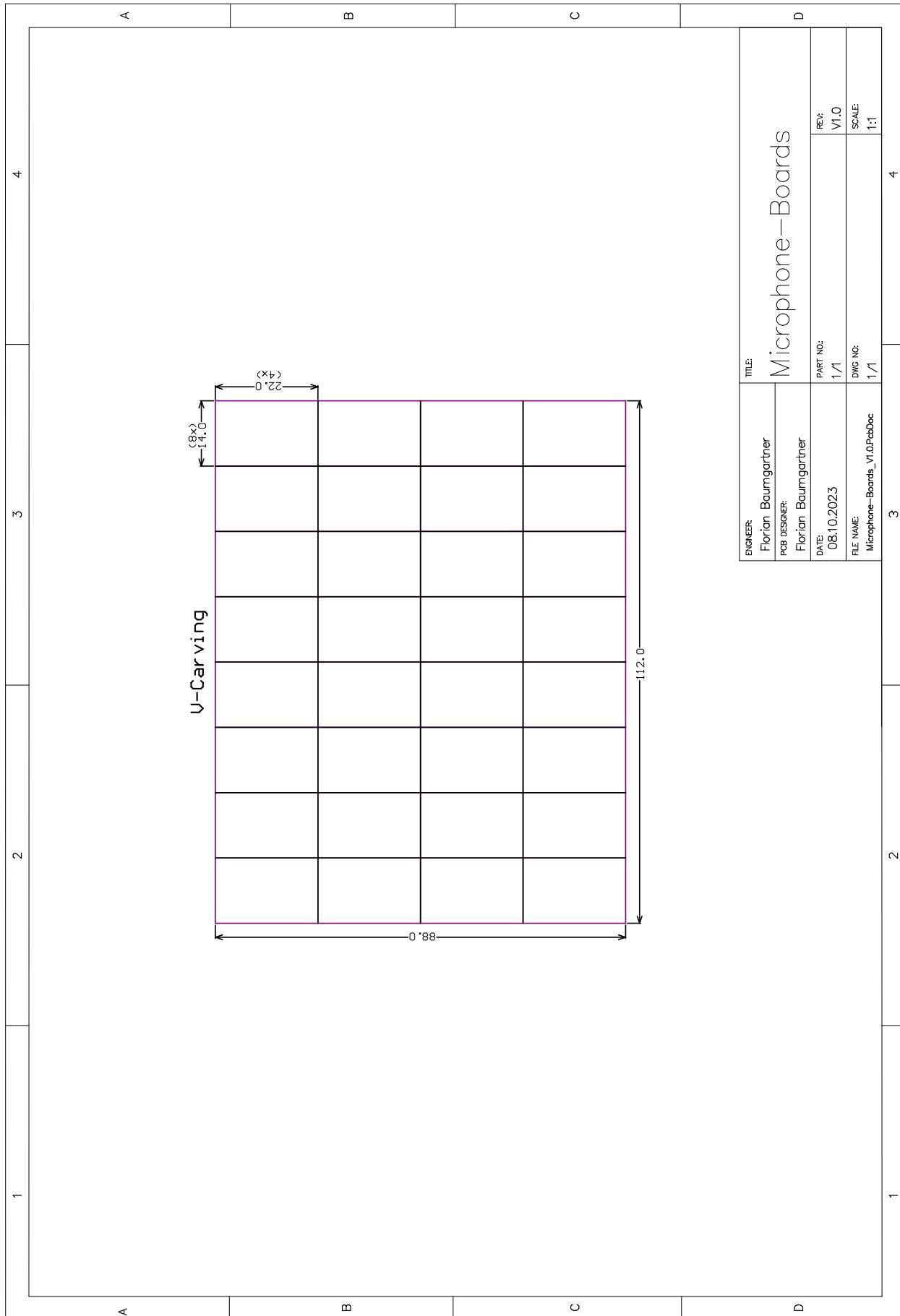


A.8 Microphone-Boards PCB Top-Overlay



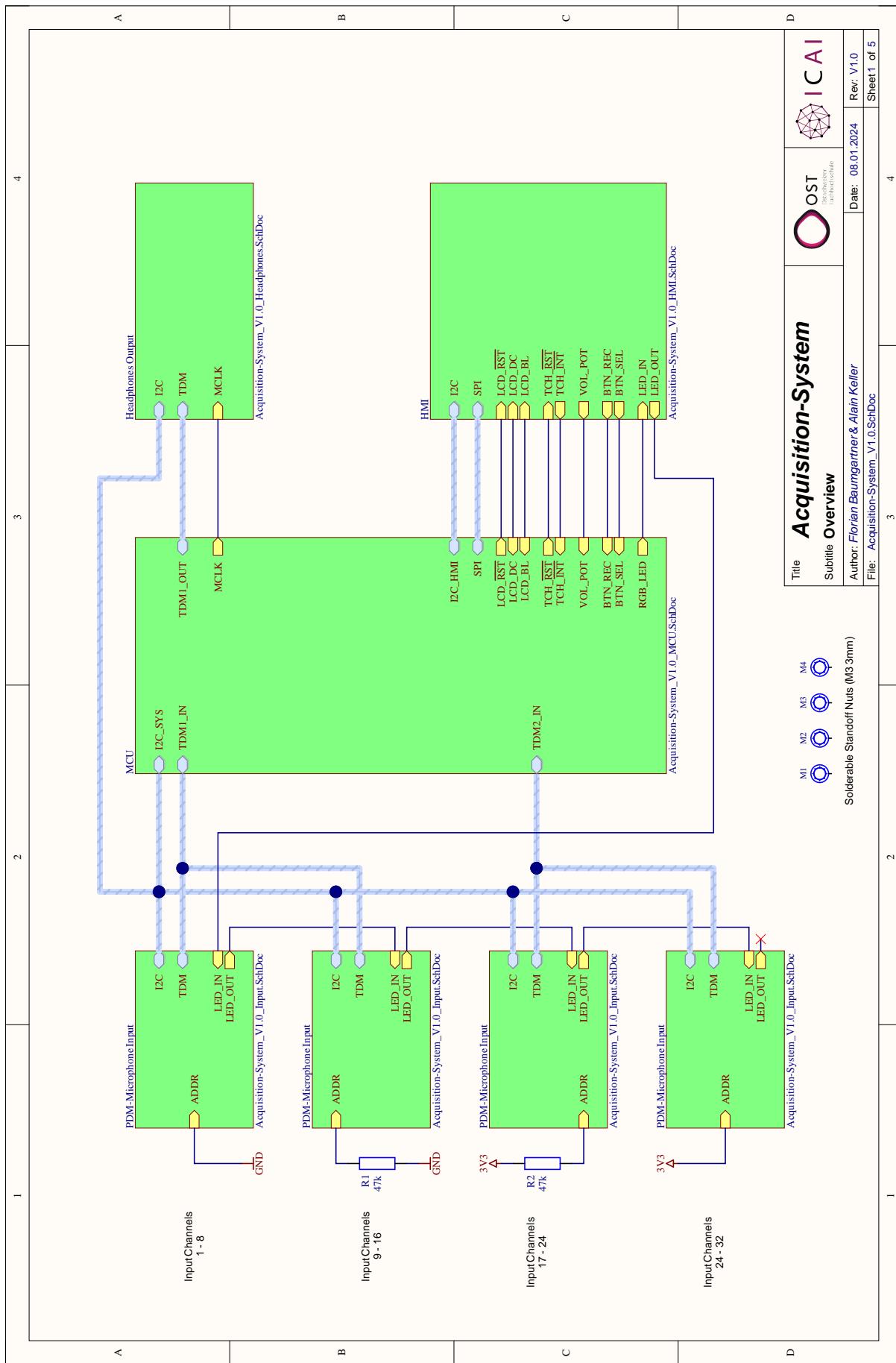
A.9 Microphone-Boards PCB Bottom-Overlay

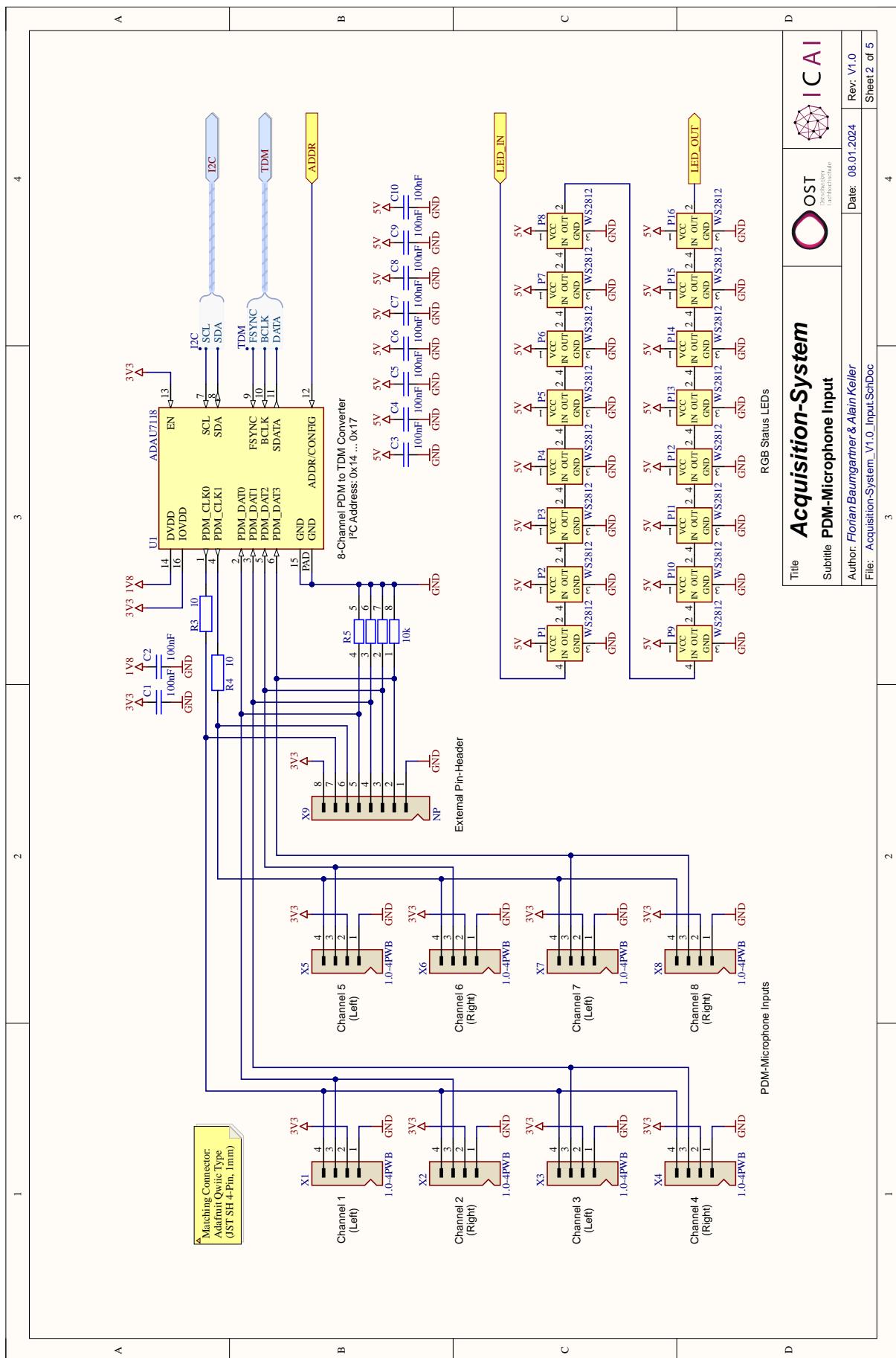
A.10 Microphone-Boards PCB Outline

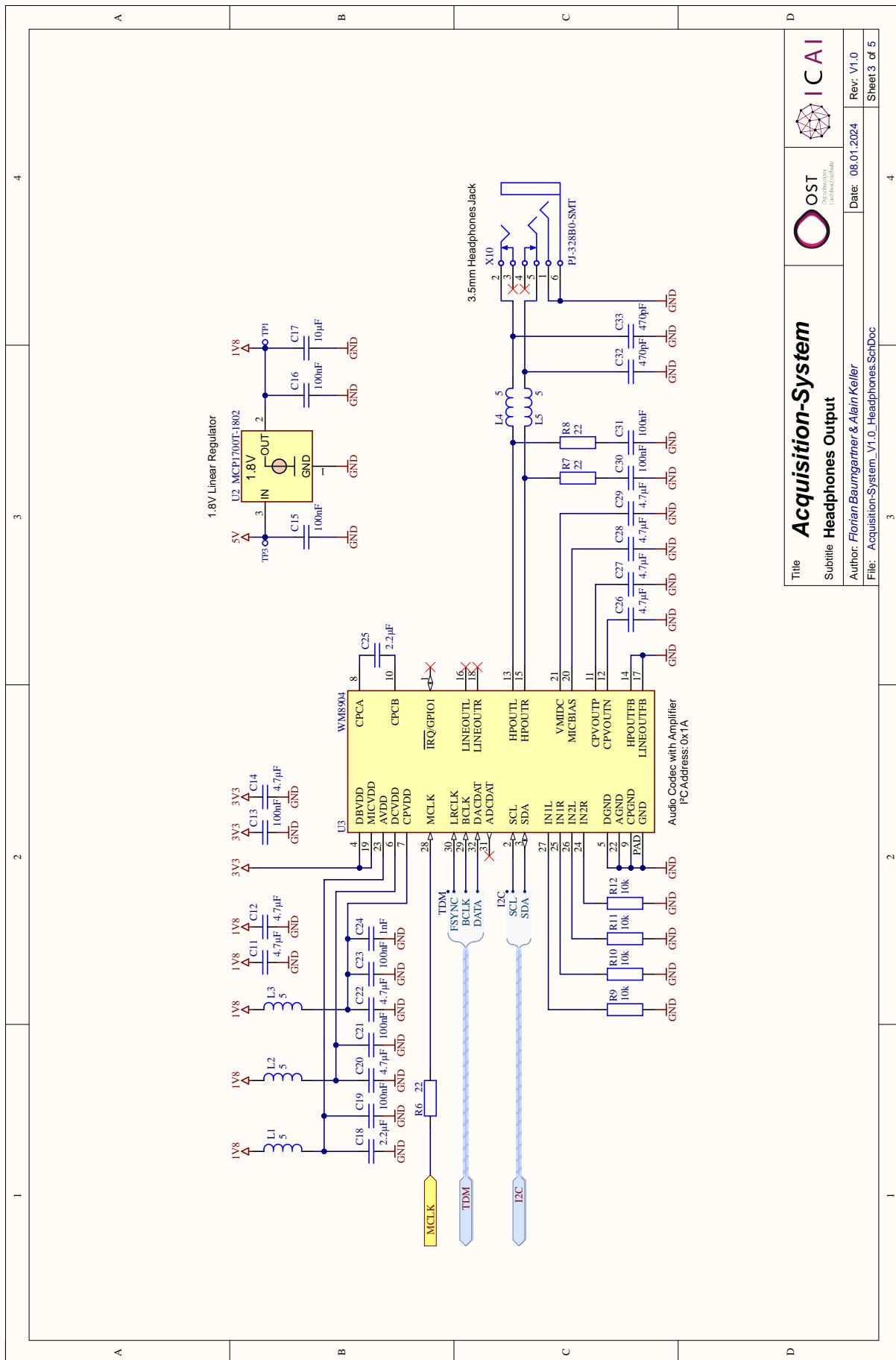


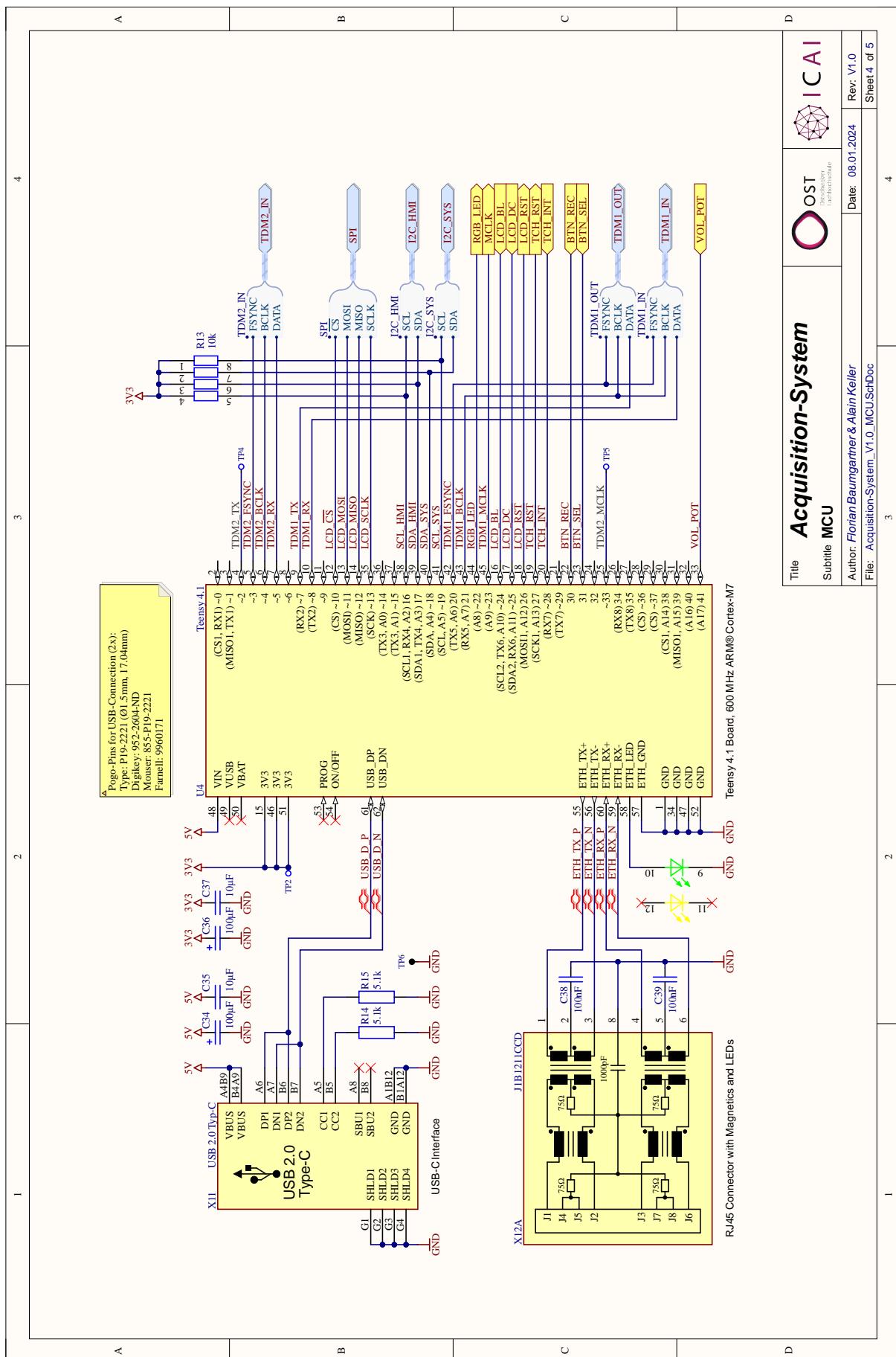
A.11 Microphone-Boards Bill of Materials (BOM)

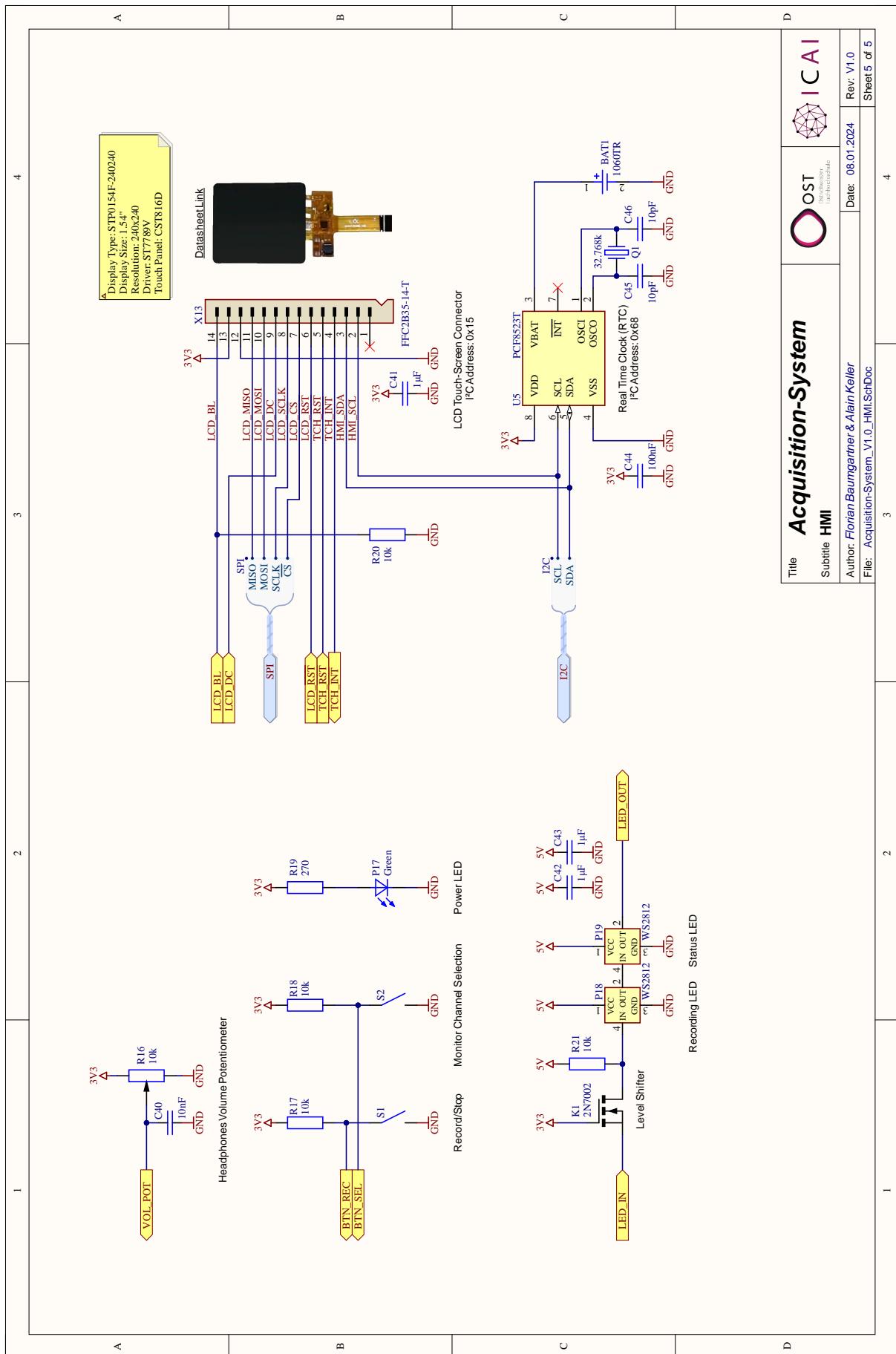
A.12 Acquisition-System Schematics



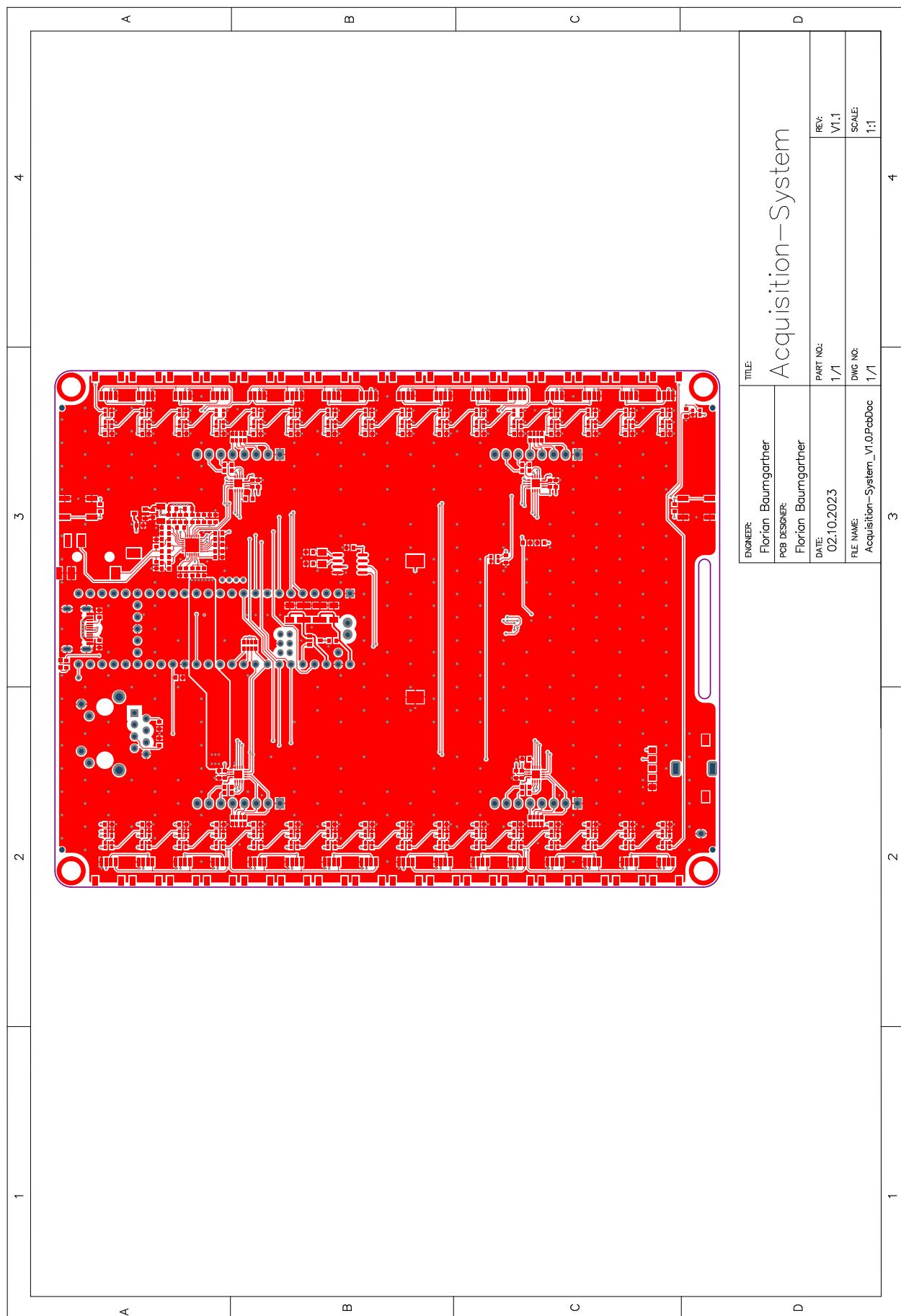




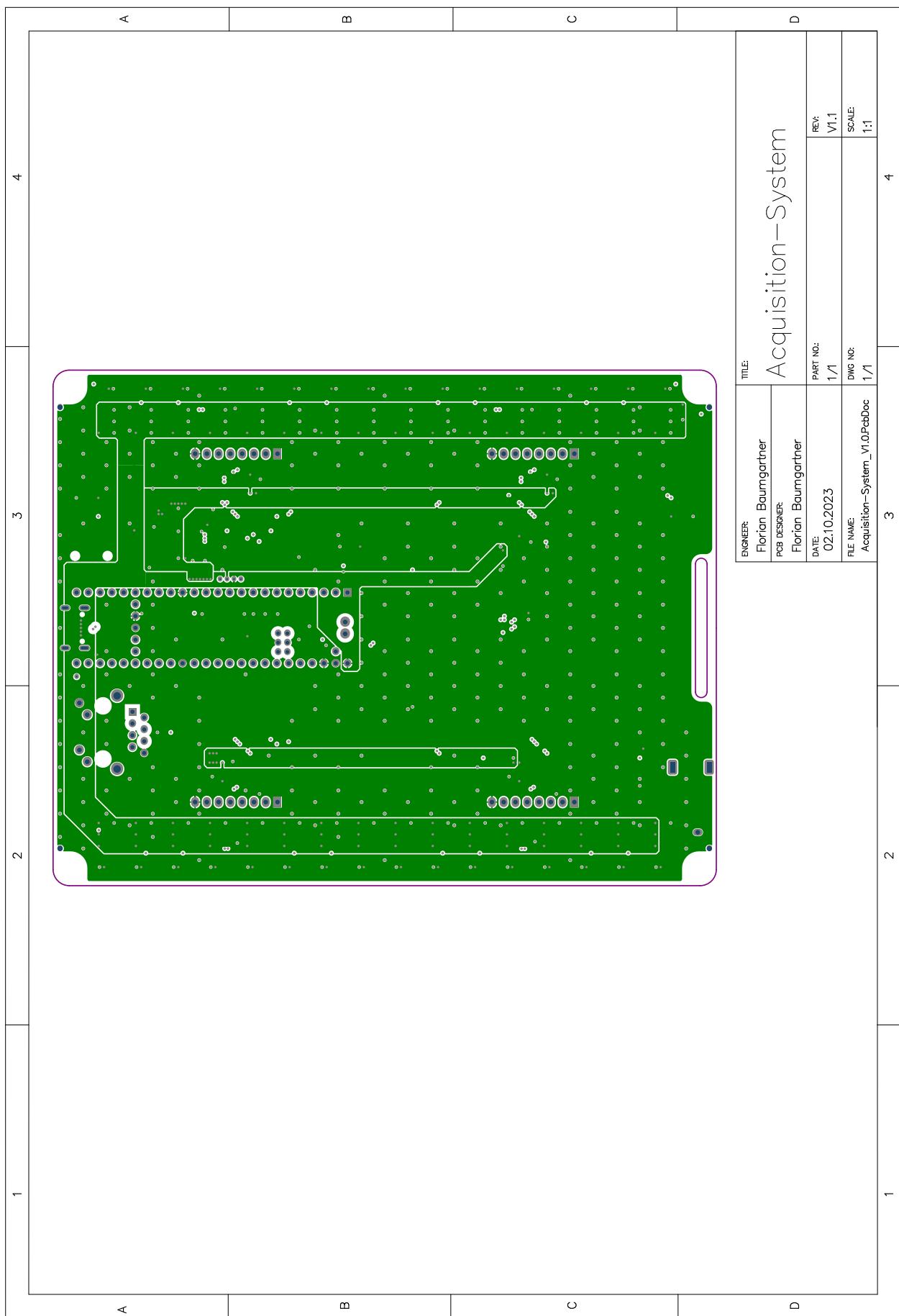




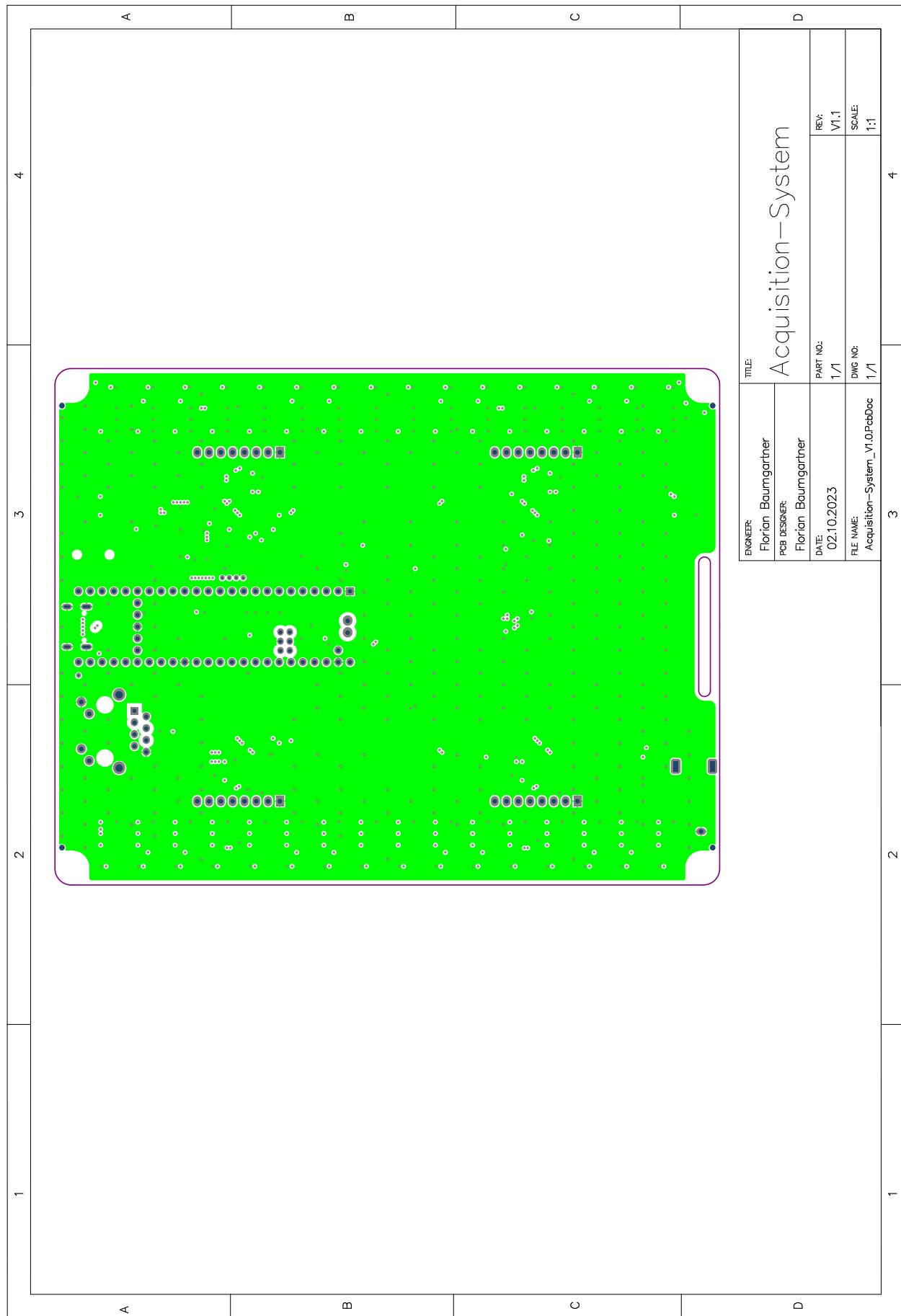
A.13 Acquisition-System PCB Top-Layer



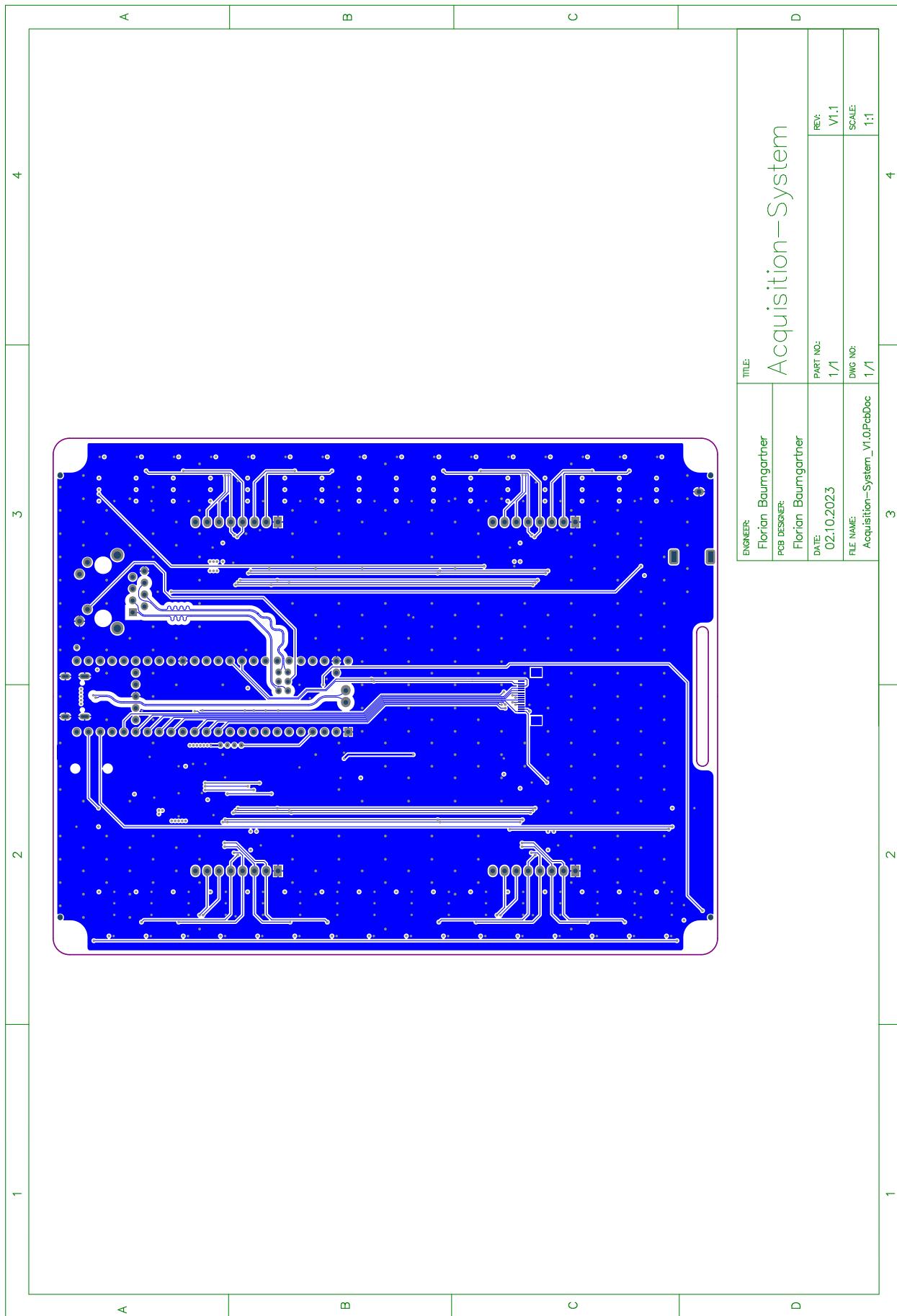
A.14 Acquisition-System PCB Mid-Layer 1



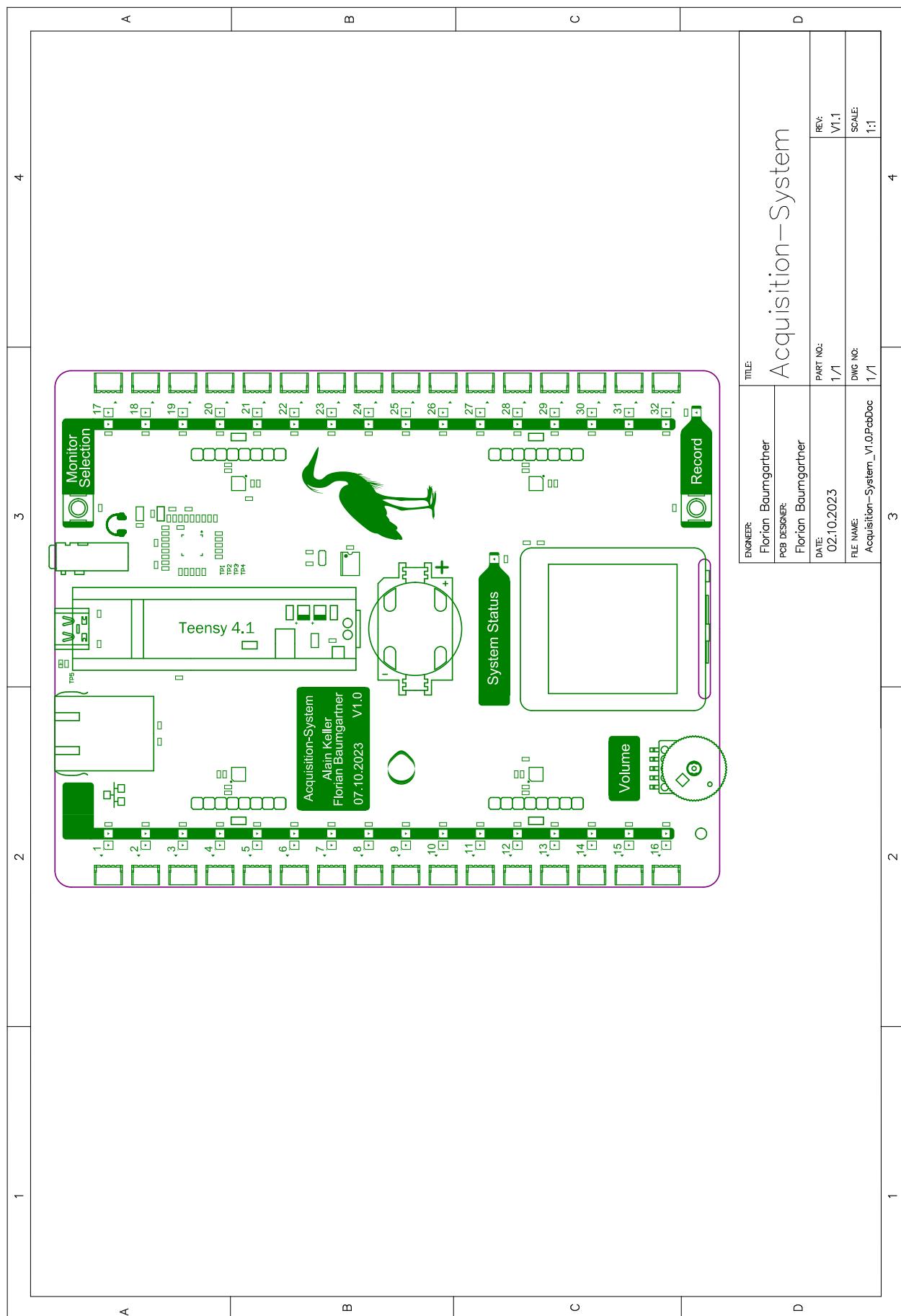
A.15 Acquisition-System PCB Mid-Layer 2



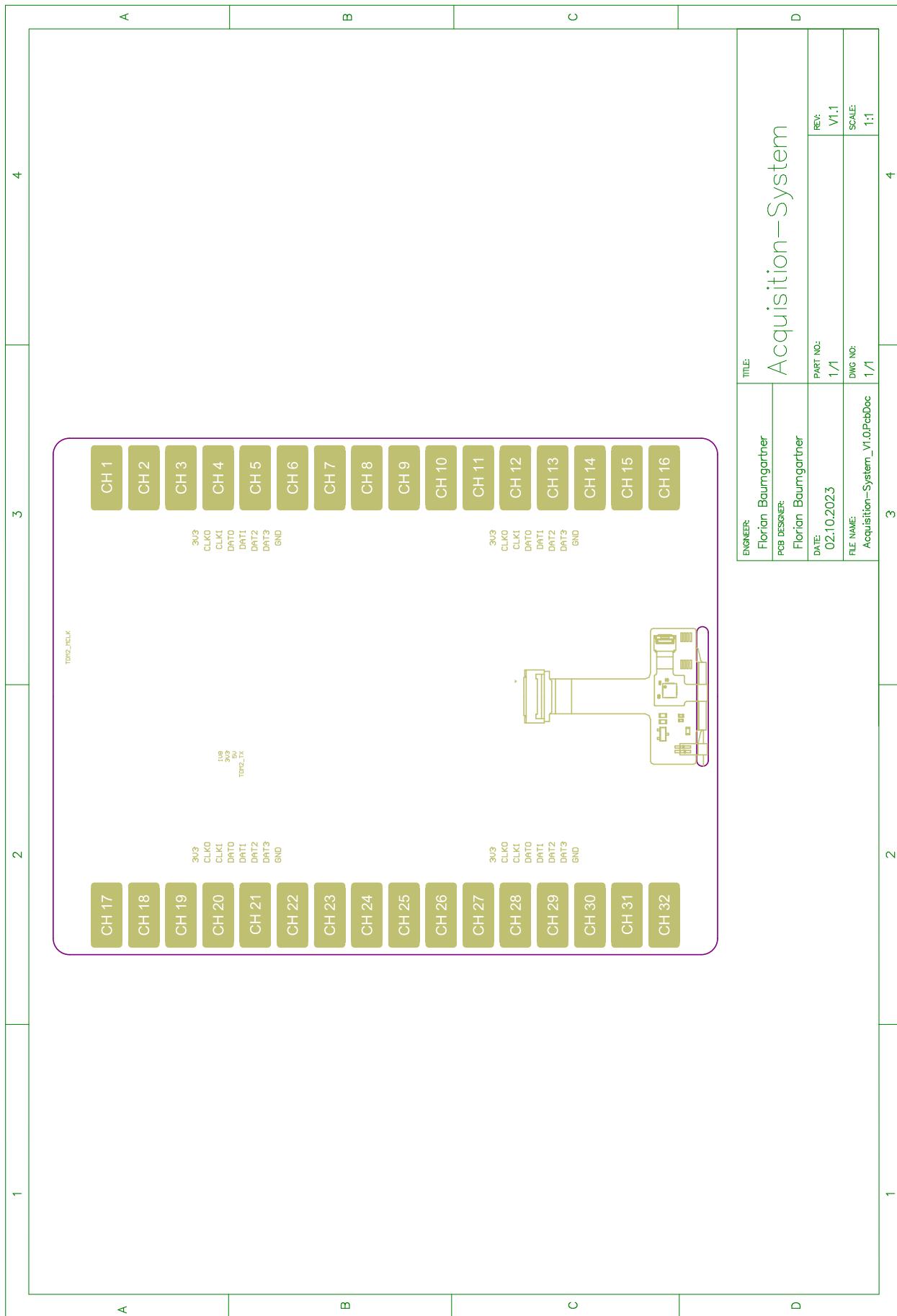
A.16 Acquisition-System PCB Bottom-Layer



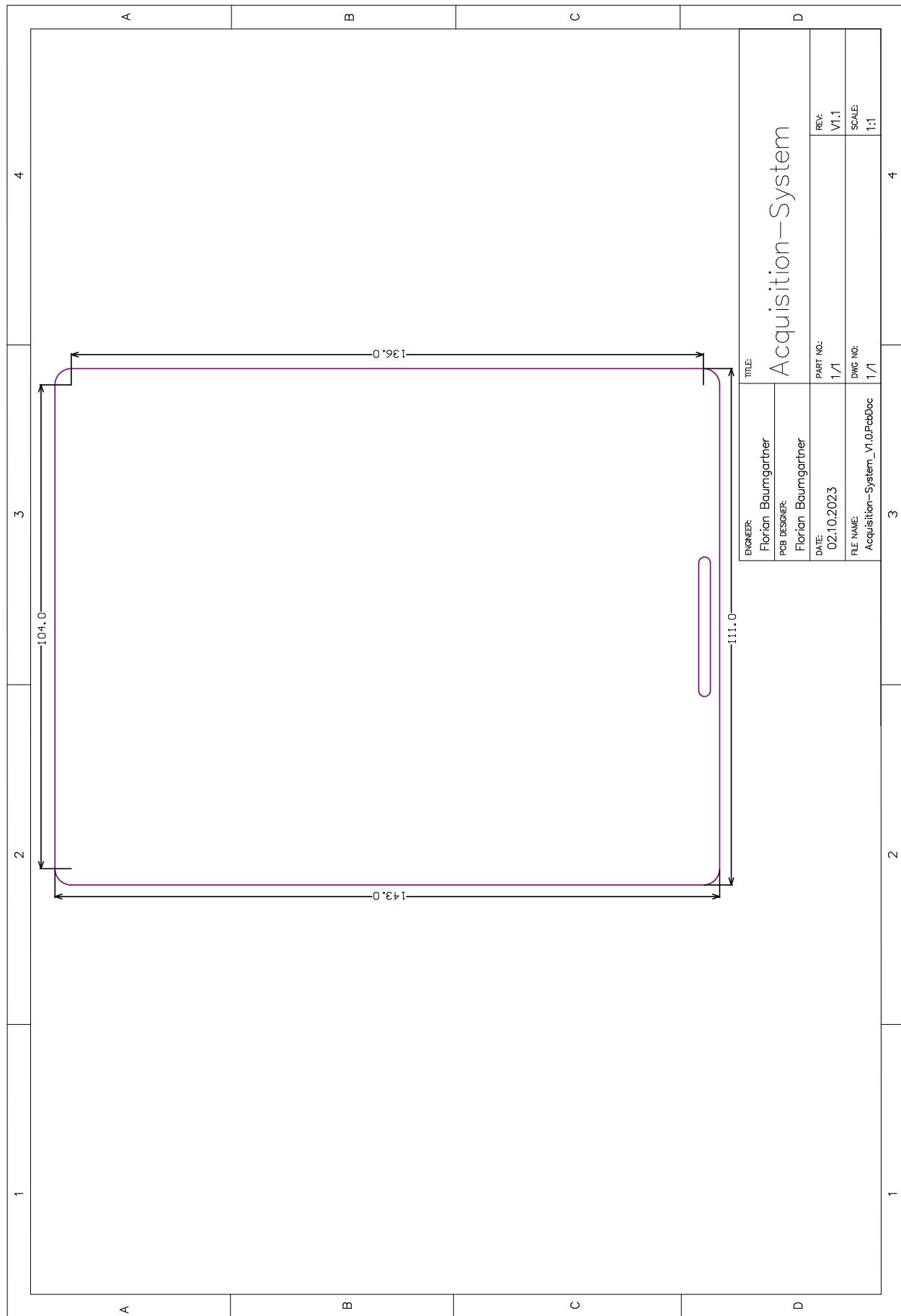
A.17 Acquisition-System PCB Top-Overlay



A.18 Acquisition-System PCB Bottom-Overlay



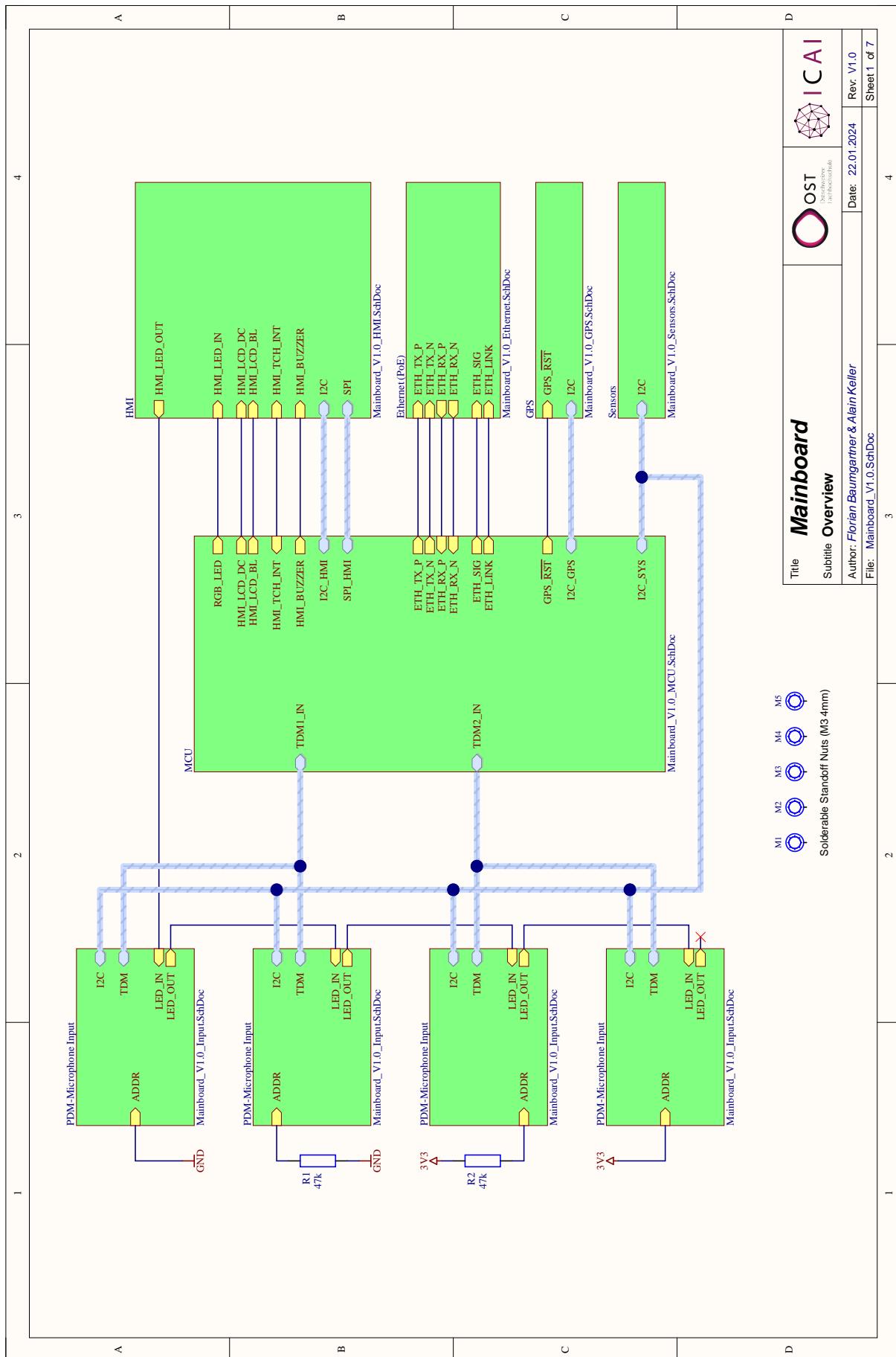
A.19 Acquisition-System PCB Outline

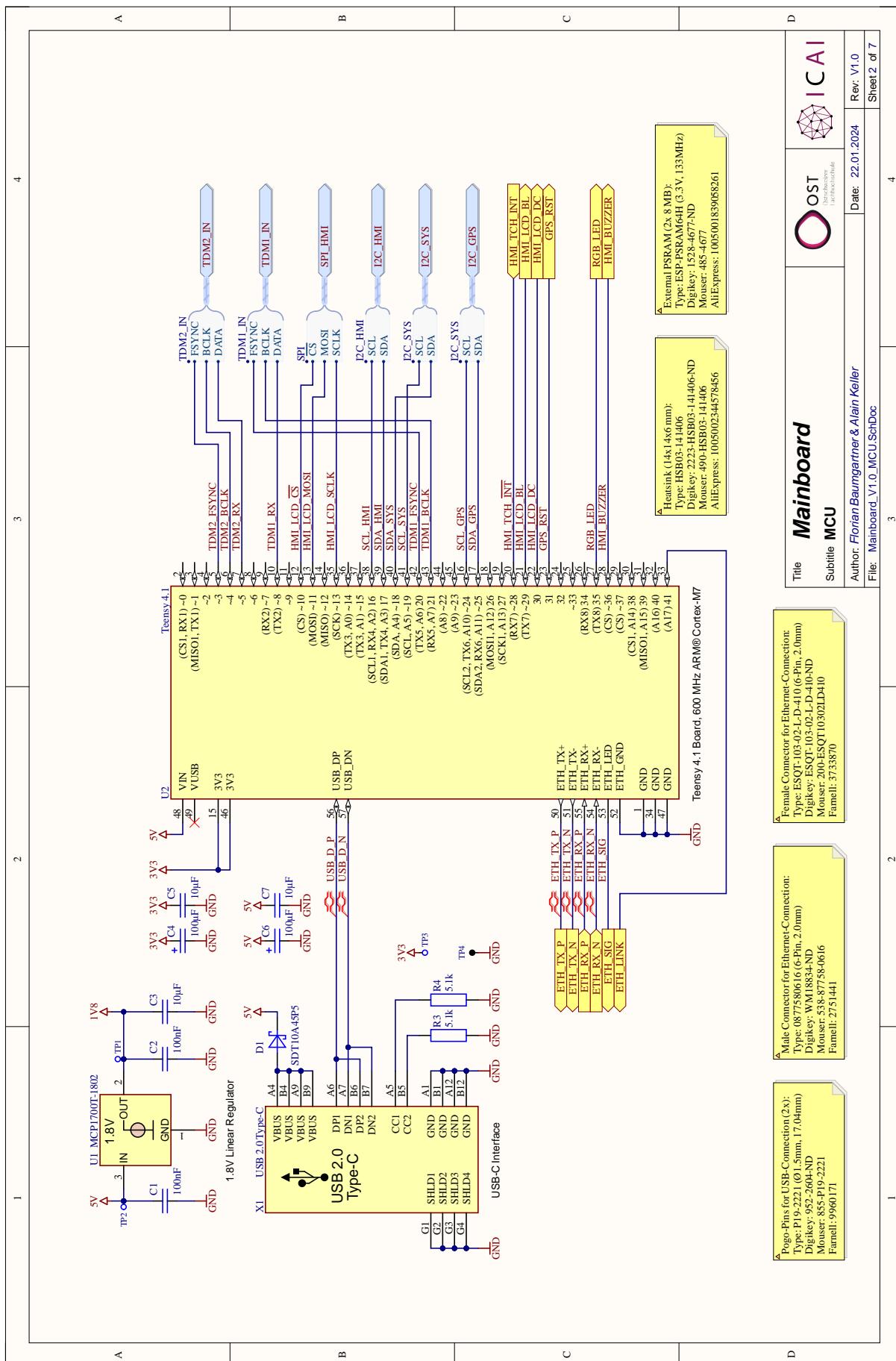


A.20 Acquisition-System Bill of Materials (BOM)

Bill of Materials for Project [Acquisition-System.PrjPcb]							
File name: Project: Variant: Creation Date: Print Date:		Component Description				Quantity	
Designator	Component Value	Footprint	JLC Part#	Family	Manufacturer	Quantity	Total Price
SAT1	100nF 50V	SMD KeraMetac Kondensator 0603	C14663	253470	50V 100nF 50V	1	SR,-2%
C5D, C6A, C8B, C9C, C10D, C22A, C28, C30, C32, C34, C40, C42, C44, C46, C50, C52, C54, C56, C58, C60, C62, C64, C66, C68, C70, C72, C74, C76, C78, C80, C82, C84, C86, C88, C90, C92, C94	Multilayer Ceramic Capacitors	SMD KeraMetac Kondensator 0603	C19566	144-56393-1-ND	51	SR, 0.4%	
C11, C12, C14, C20, C26, C27, C28, C29	Multilayer Ceramic Capacitors M_LCC	SMD KeraMetac Kondensator 0603	C19567	144-56394-5-CTND	9	SR, 0.0%	
C24, C25	Multilayer Ceramic Capacitors	SMD KeraMetac Kondensator 0603	C19568	144-56395-1-ND	3	SR, 0.0%	
C32, C33	Multilayer Ceramic Capacitors	SMD KeraMetac Kondensator 0603	C19569	144-56396-1-ND	1	SR, 0.0%	
C34, C36	Tantalum Ceramic Capacitors	SMD KeraMetac Kondensator 0603	C19570	144-56397-1-ND	2	SR, 0.0%	
C40	10nF 50V	SMD KeraMetac Kondensator 0603	C16133	1130408	50V 10nF 50V	2	SR, 0.0%
C41, C42, C43	Multilayer Ceramic Capacitors	SMD KeraMetac Kondensator 0603	C27112	2340962	77-VA0603X103Z2XAPBC	1	SR, 0.0%
C44, C46	1nF 50V	SMD KeraMetac Kondensator 0603	C15649	3013863	187-Q-110A105BBNINN	3	SR, 0.0%
K1	2nH/50V	SMD KeraMetac Kondensator 0603	C1634	141601	80-C0603C1005U	1	SR, 0.0%
K1, K2, K3, K4, K5, K6, K7, K8, K9, K10, K11, K12, K13, K14, K15	MOSFET N-Chanel	SOT-23	C2545	51-2027402	2070293CT-ND	1	SR, 0.0%
P10, P10C, P10D, P10E, P10F, P10G, P10H, P10I, P10J, P10K, P10L, P10M, P10N, P10O, P10P, P10Q, P10R, P10S, P10T, P10U, P10V, P10W, P10X, P10Y, P10Z	Surface Mount Standard Nut	3.2mm	C291750	3.650622	151-SMT-SG-465SET	4	SR, 0.0%
P10, P10B, P10C, P10D, P10E, P10F, P10G, P10H, P10I, P10J, P10K, P10L, P10M, P10N, P10O, P10P, P10Q, P10R, P10S, P10T, P10U, P10V, P10W, P10X, P10Y, P10Z	Surface Mount Standoff Nut	3.2mm	C29186	3.650629	151-SMT-SG-465SET	66	SR, 0.0%
P17	General Purpose LED	SMD LED 0603	C272043	1465991	648-588-801-0107F	1	SR, 0.34%
S1, S2	Crystal	SMD Crystal 22x 5x6.9	C22346	732-532-7901K-15	1	SR, 0.18%	
R1, R2	Thick Film Chip Resistor	SMD Thick Film Chip Resistor 0603	C23119	2301213	172-501-051-01	2	SR, 0.02%
S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13	Resistors	SMD Thick Film Chip Resistor 0603	C23120	2300955	172-501-051-01	8	SR, 0.0%
R6, R7, R8	Isolated Resistor Active	SMD Widetrend Active 4-Fach 0603	C23118	2306167	YAO1052-10V	5	SR, 0.0%
R9, R10, R11, R12, R17, R18, R20, R21	Thick Film Chip Resistor	SMD Widetrend 0603	C23345	2320096	697-ERJ-36KF1002V	3	SR, 0.01%
R14, R15	Thick Film Chip Resistor	SMD Widetrend 0603	C23196	2305192	P5-10KH67-ND	8	SR, 0.01%
R16	Powermeter	RK1011 TROPM (Darl 10A)	C270766	688-RK1011TROPM	2	SR, 0.02%	
S1, S2	Thick Film Chip Resistor	E1020 100mW 1%	C235227	2303096	697-ERJ-36KF120V	1	SR, 0.06%
S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13	Tactile Switch	E1020 100mW 1%	C235227	2303092	697-EVJ-02020V	2	SR, 0.19%
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	6V225-002-5401mW-Schwarz	C24212	2301766	36-501-010-01	8	SR, 0.0%
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24213	59-430-0171-19-C079BL	4	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24214	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24215	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24216	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24217	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24218	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24219	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24220	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24221	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24222	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24223	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24224	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24225	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24226	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24227	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24228	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24229	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24230	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24231	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24232	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24233	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24234	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24235	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24236	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24237	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24238	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24239	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24240	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24241	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24242	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24243	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24244	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24245	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24246	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24247	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24248	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24249	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24250	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24251	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24252	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24253	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24254	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24255	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24256	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24257	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24258	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24259	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24260	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24261	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24262	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24263	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24264	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24265	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24266	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24267	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24268	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24269	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24270	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24271	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	Optocoupler	SM15-010-01	C24272	59-430-0171-19-C079BL	1	SR, 0.0%	
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U							

A.21 Mainboard Schematics





A

B

C

D

4

3

2

1

3

1

1

1

2

2

2

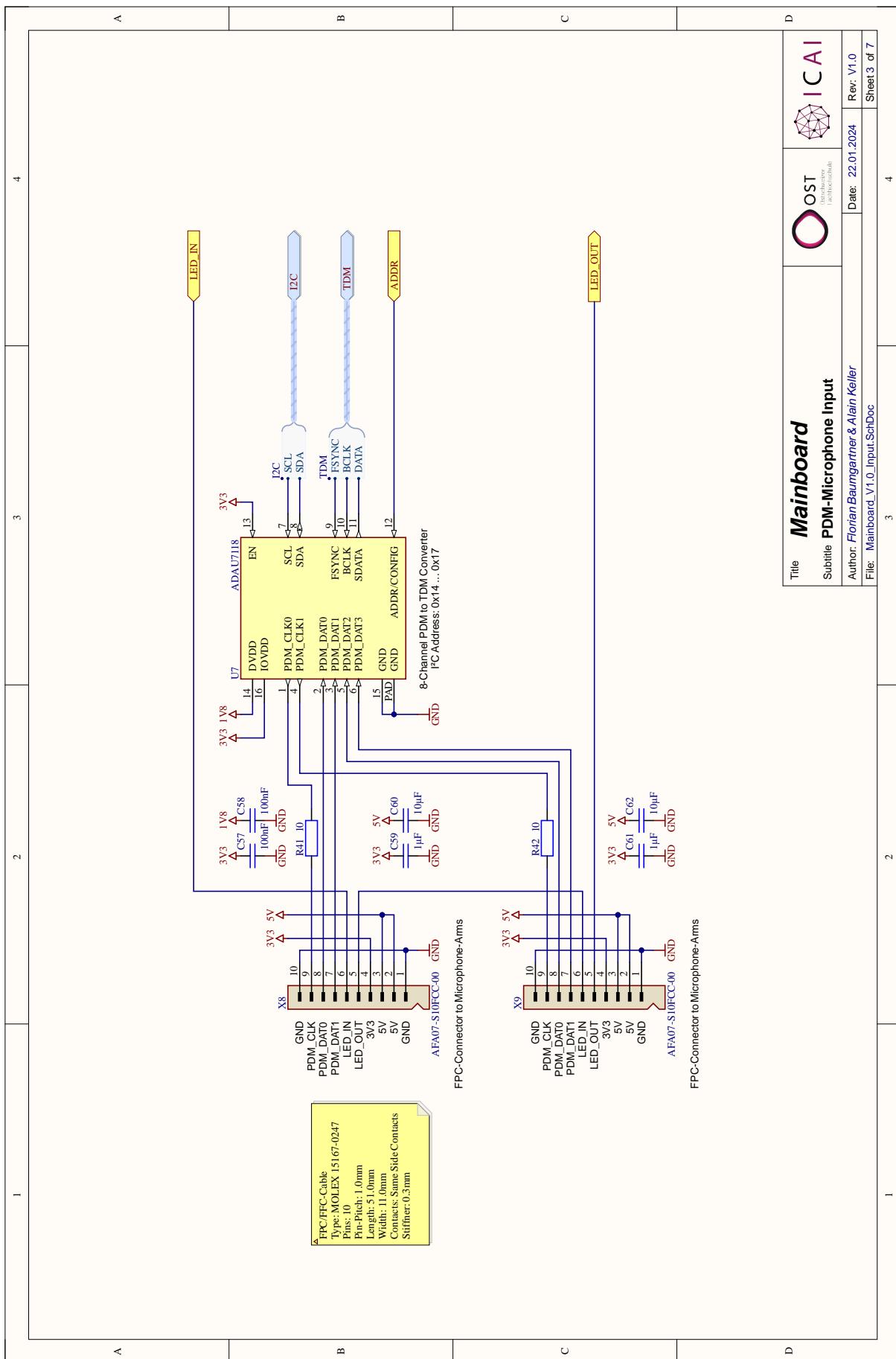
2

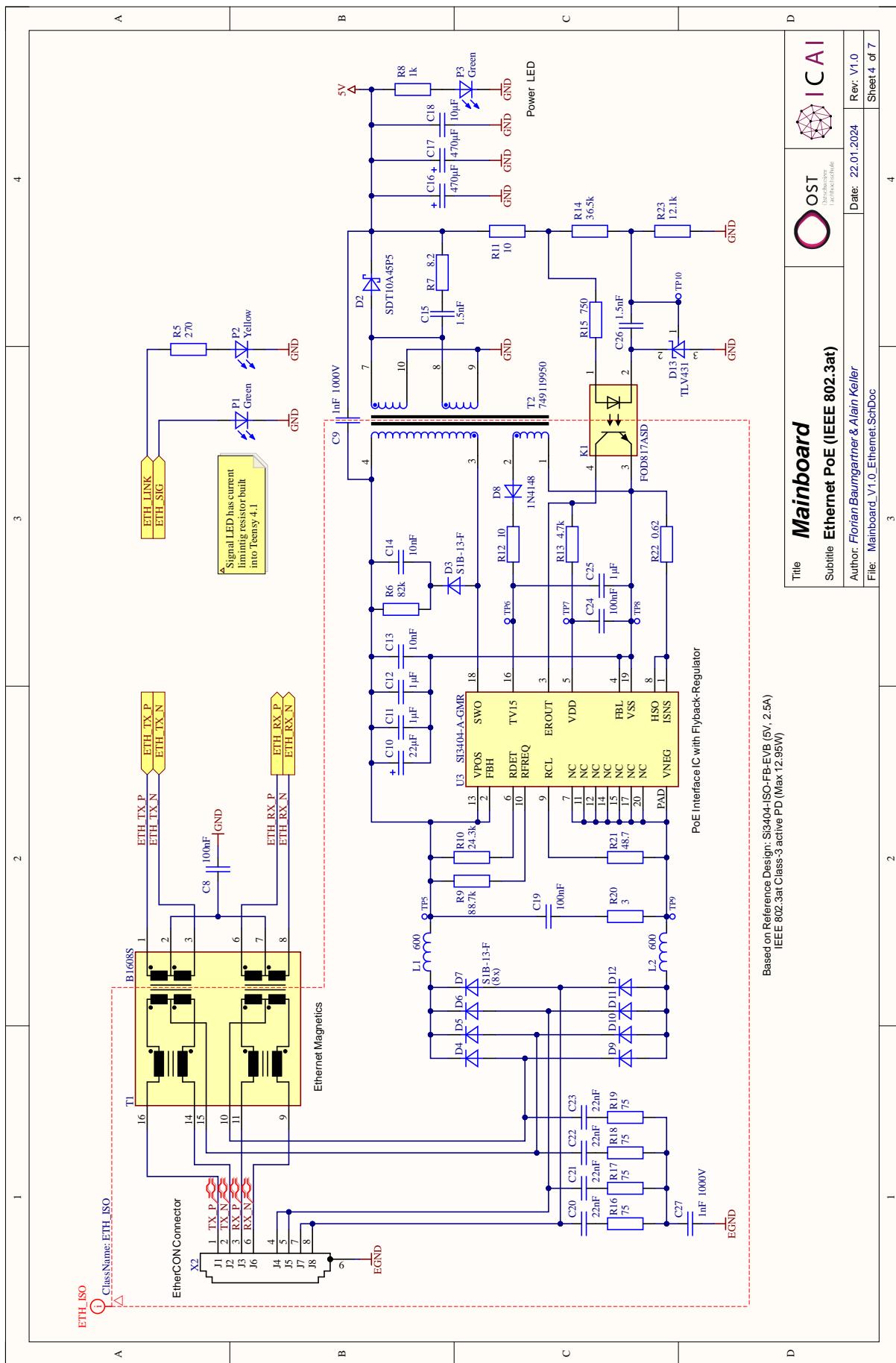
1

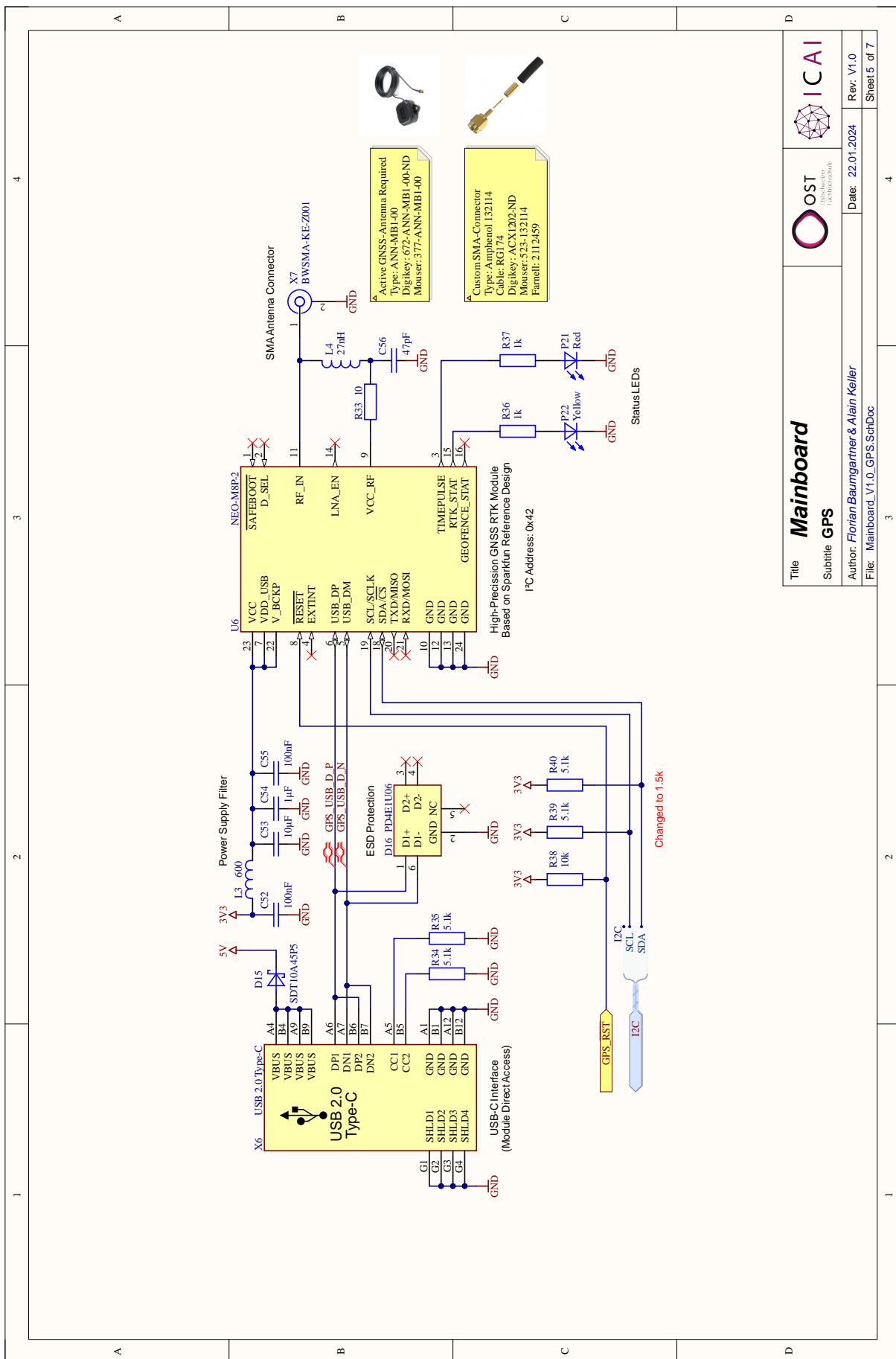
1

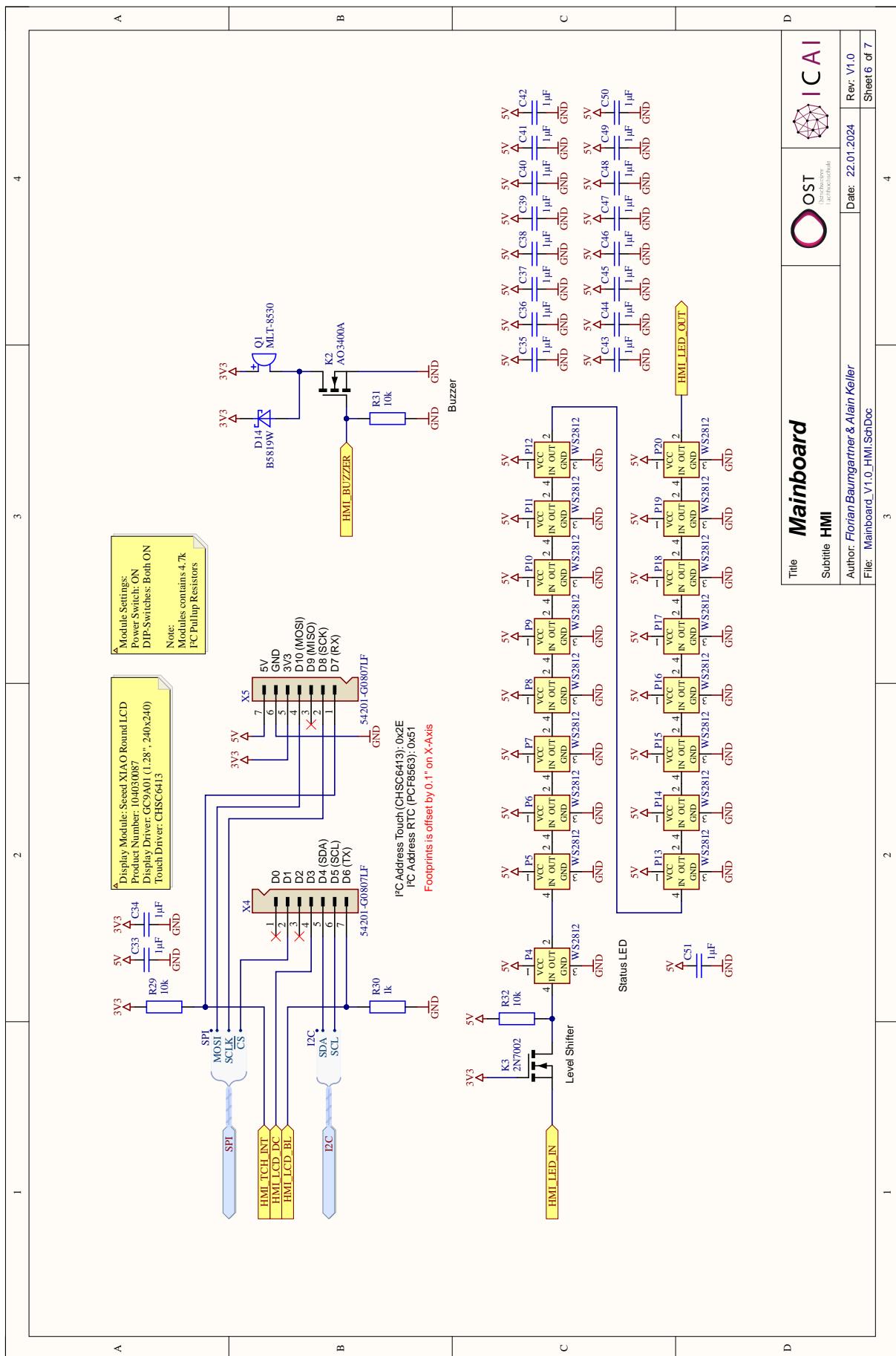
1

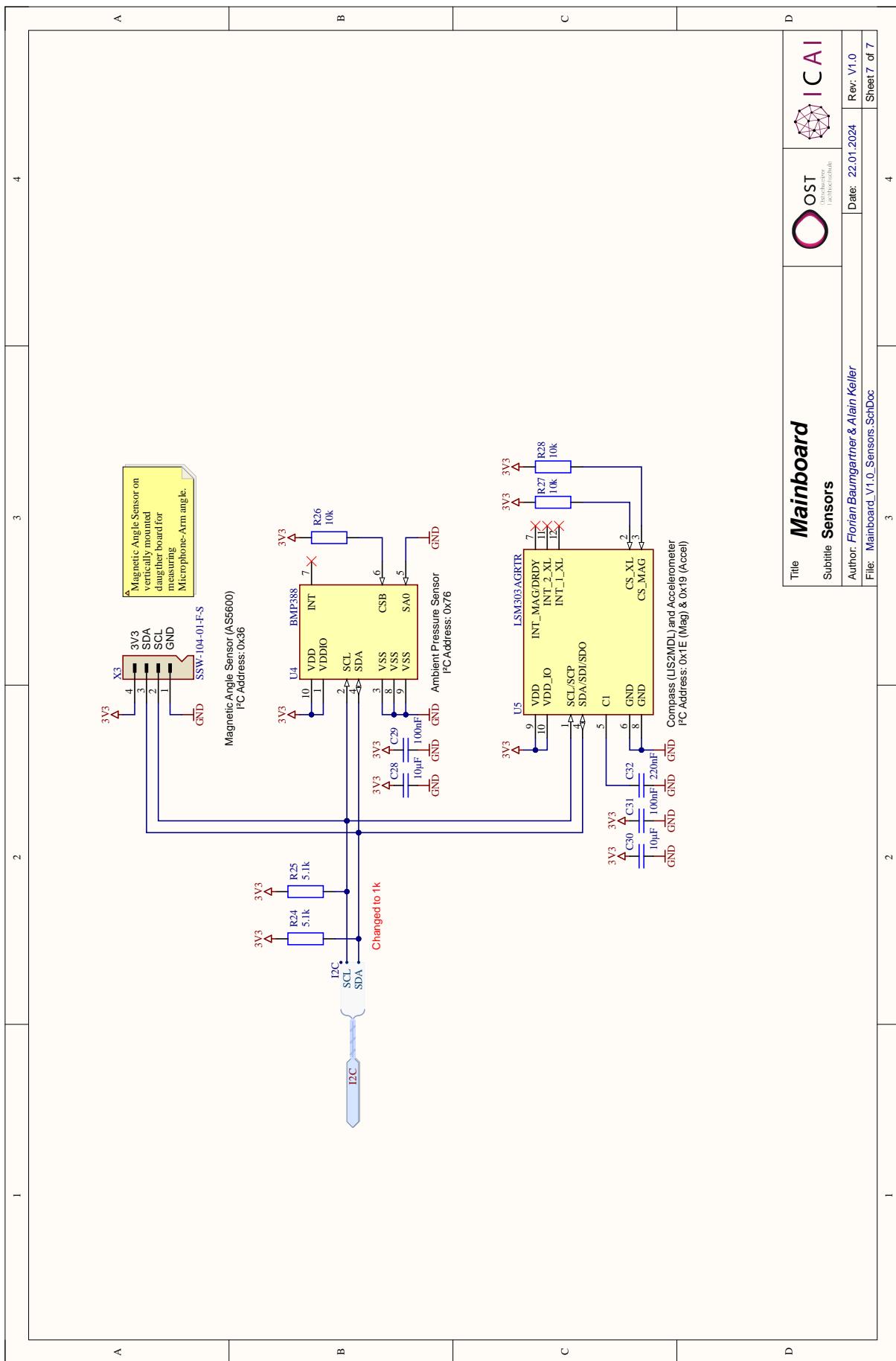
1



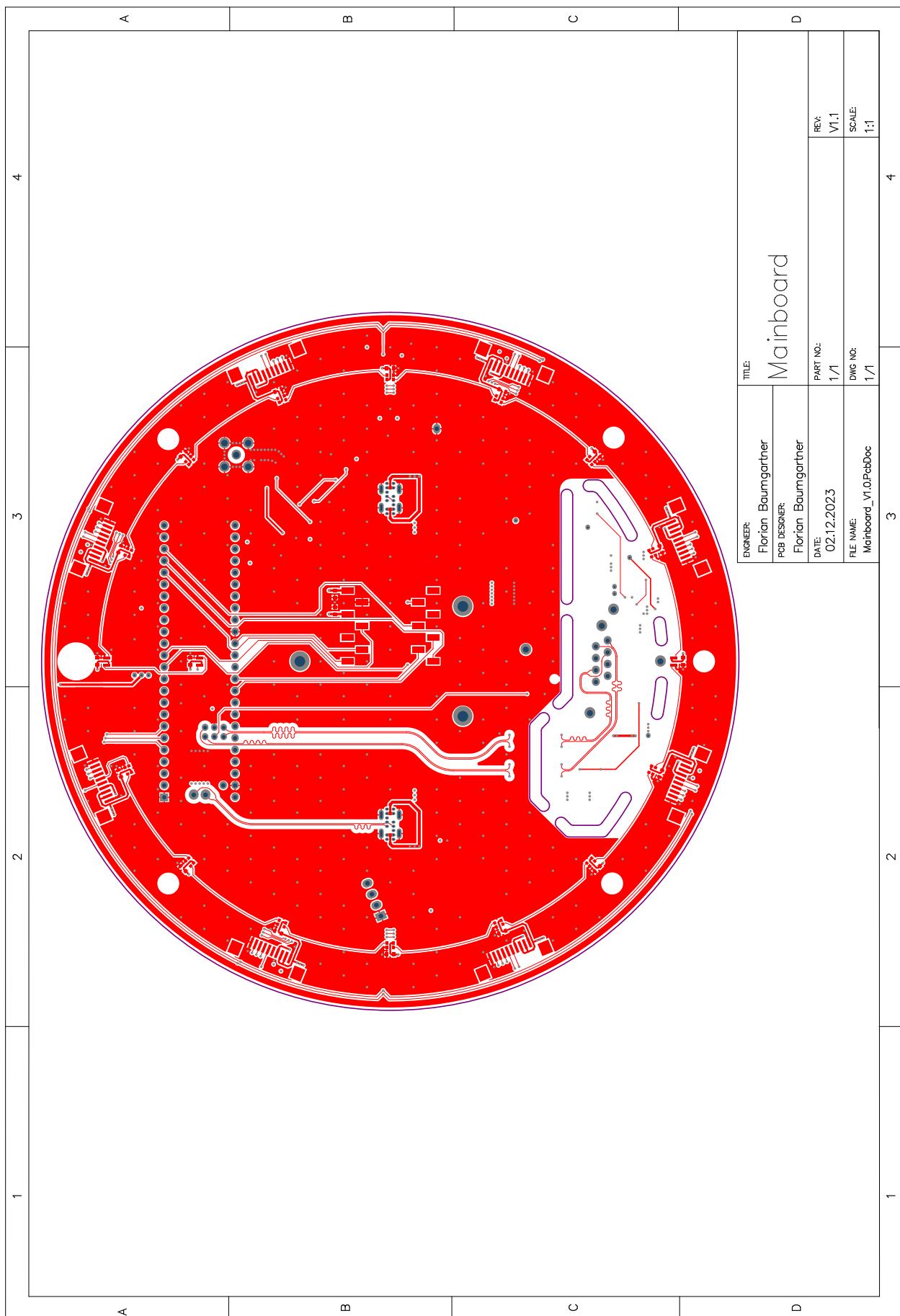




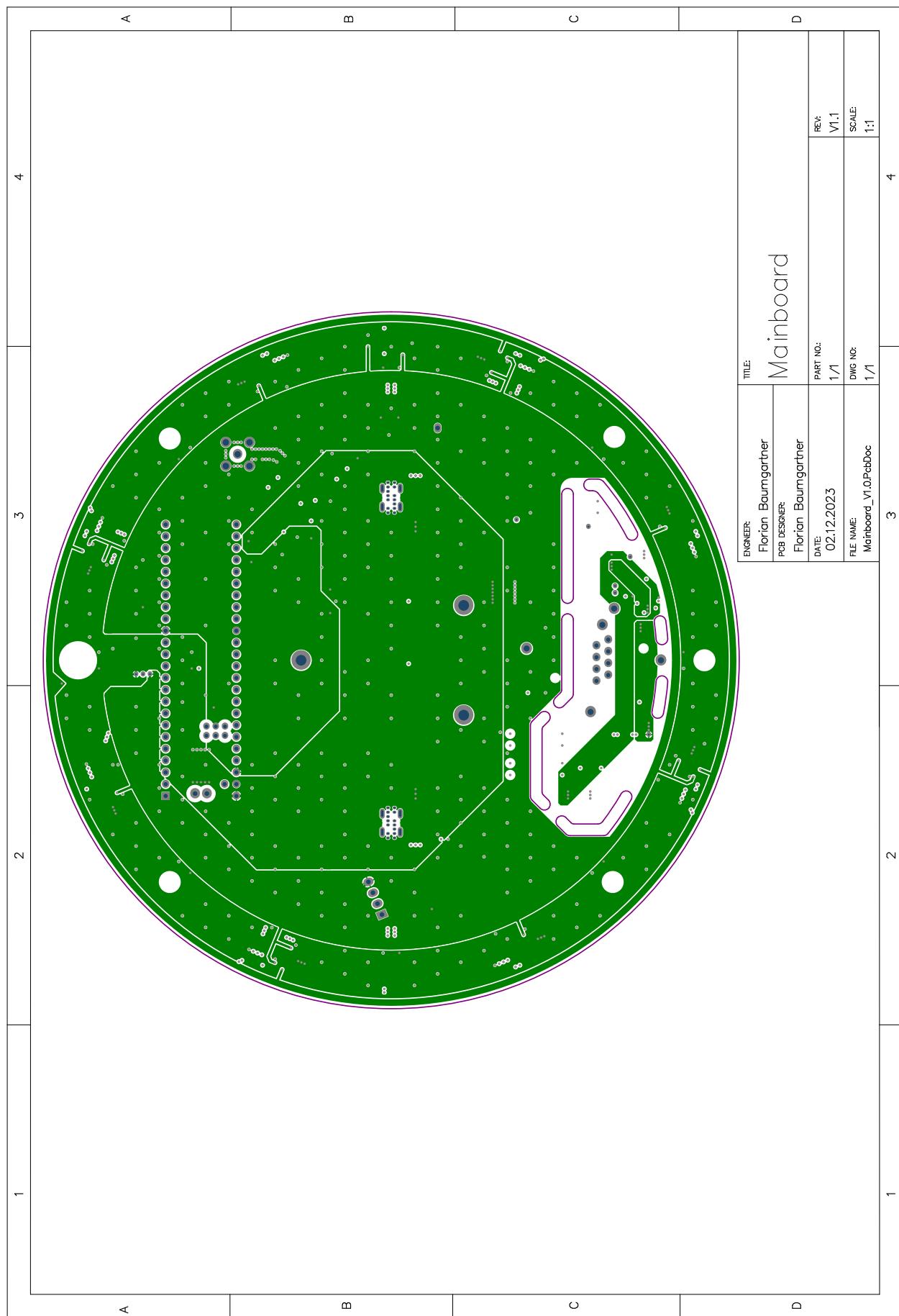




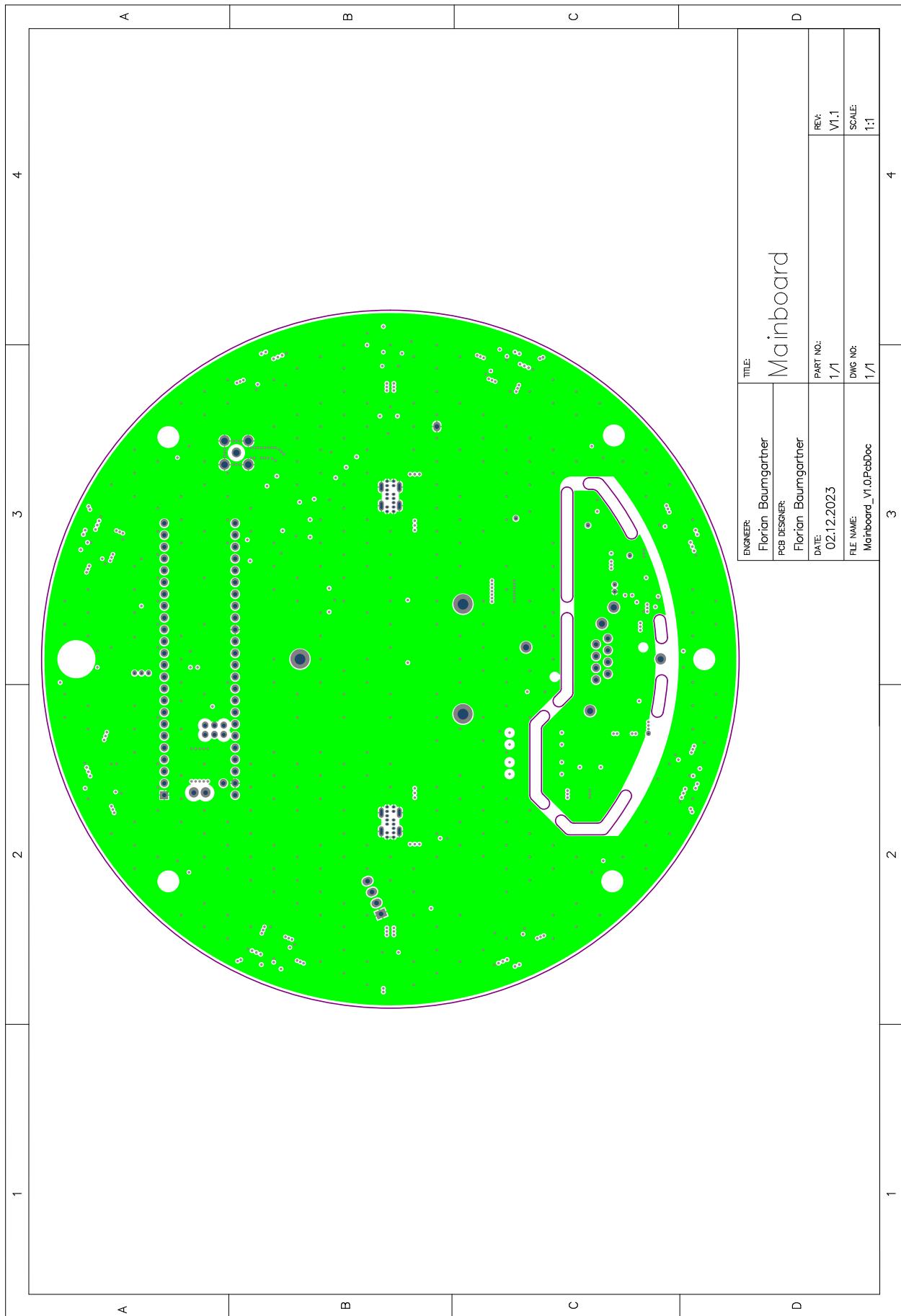
A.22 Mainboard PCB Top-Layer



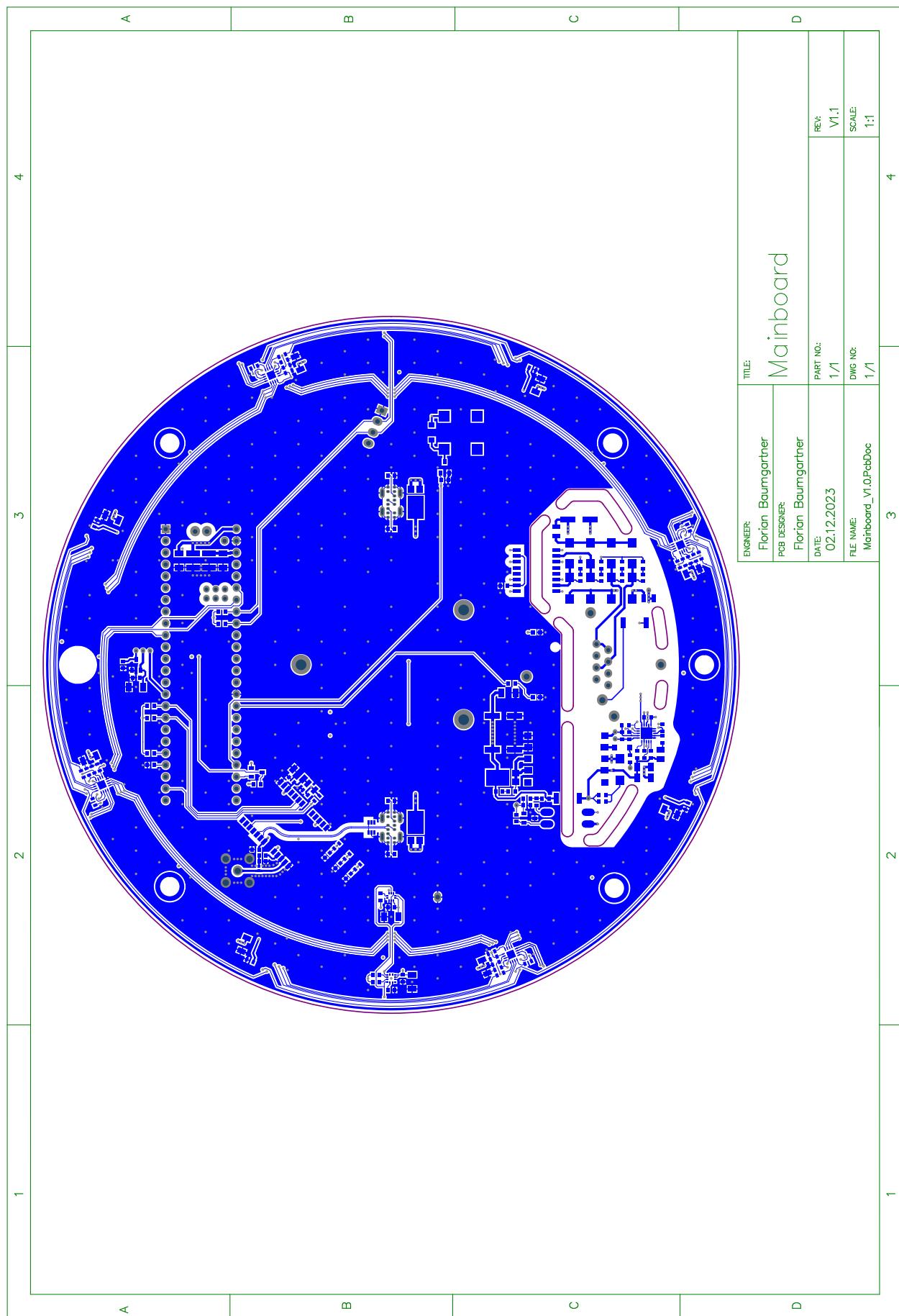
A.23 Mainboard PCB Mid-Layer 1



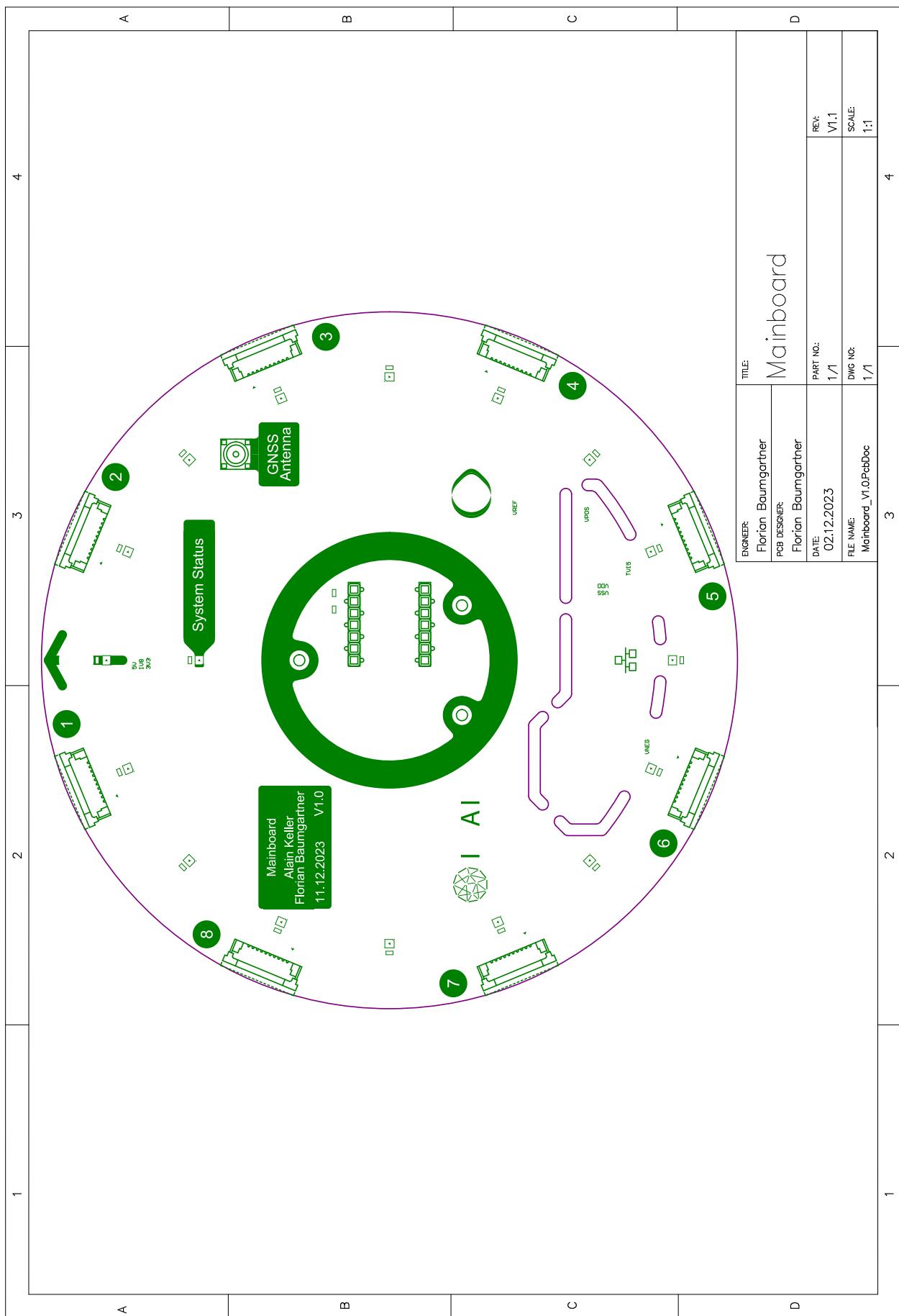
A.24 Mainboard PCB Mid-Layer 2



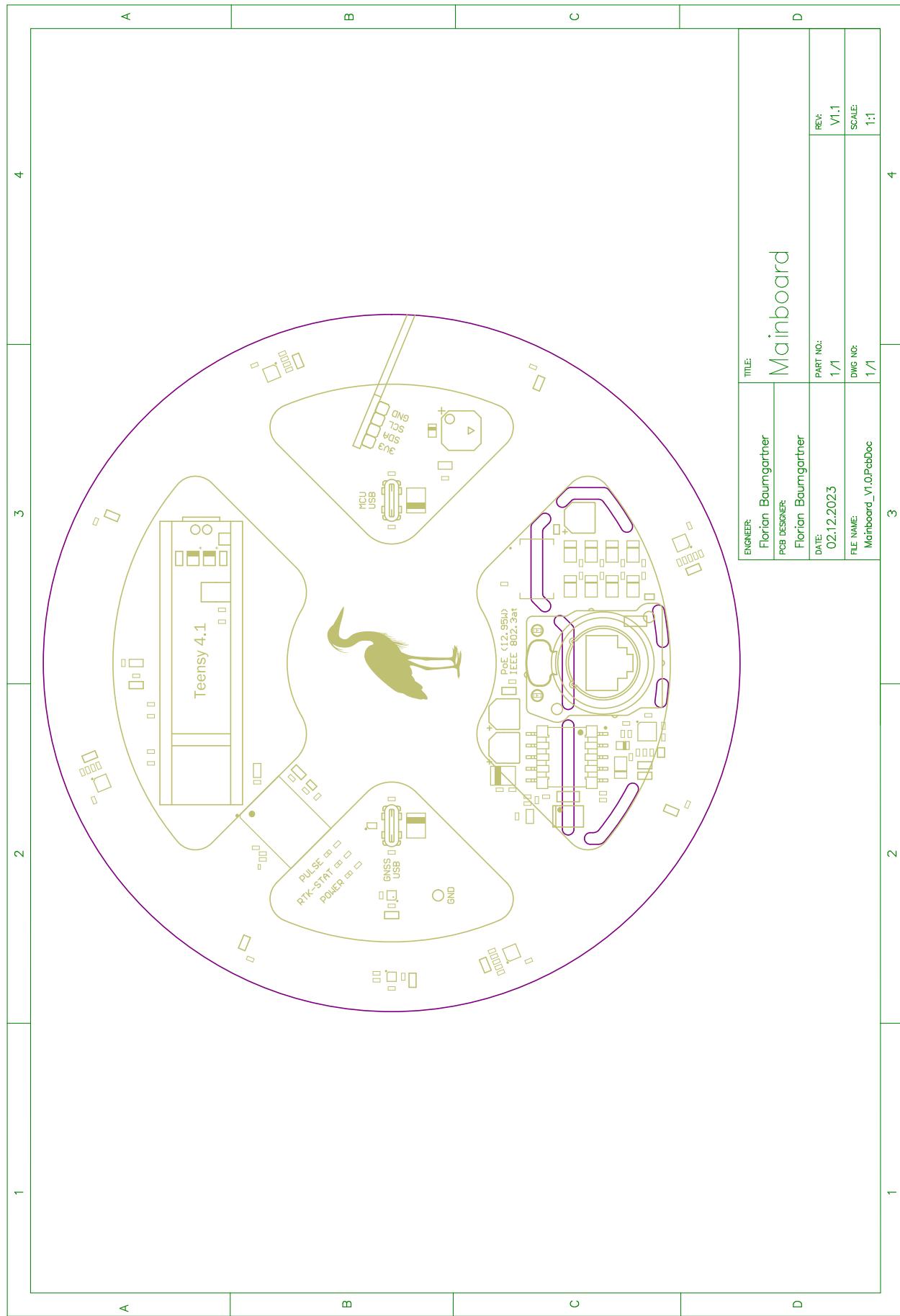
A.25 Mainboard PCB Bottom-Layer



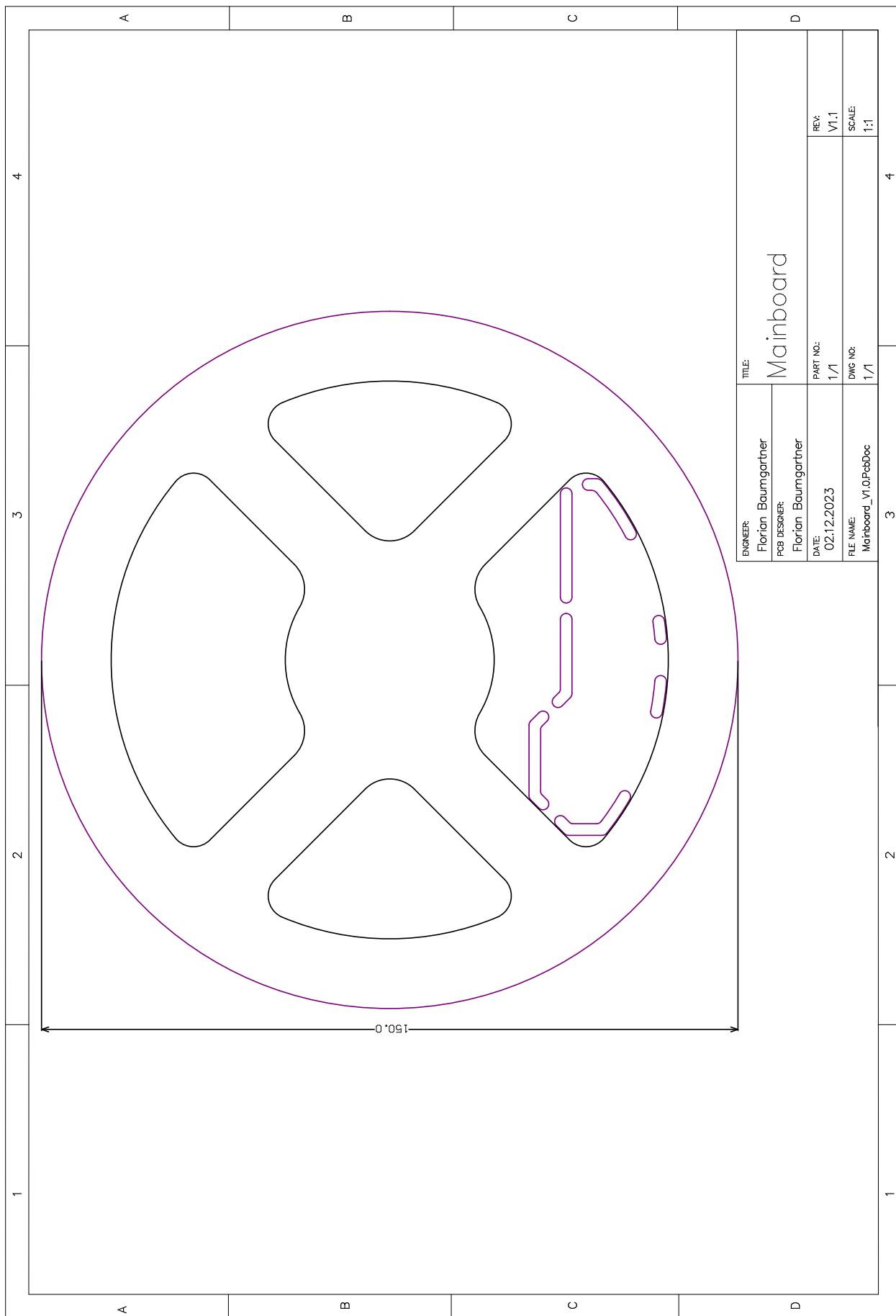
A.26 Mainboard PCB Top-Overlay



A.27 Mainboard PCB Bottom-Overlay



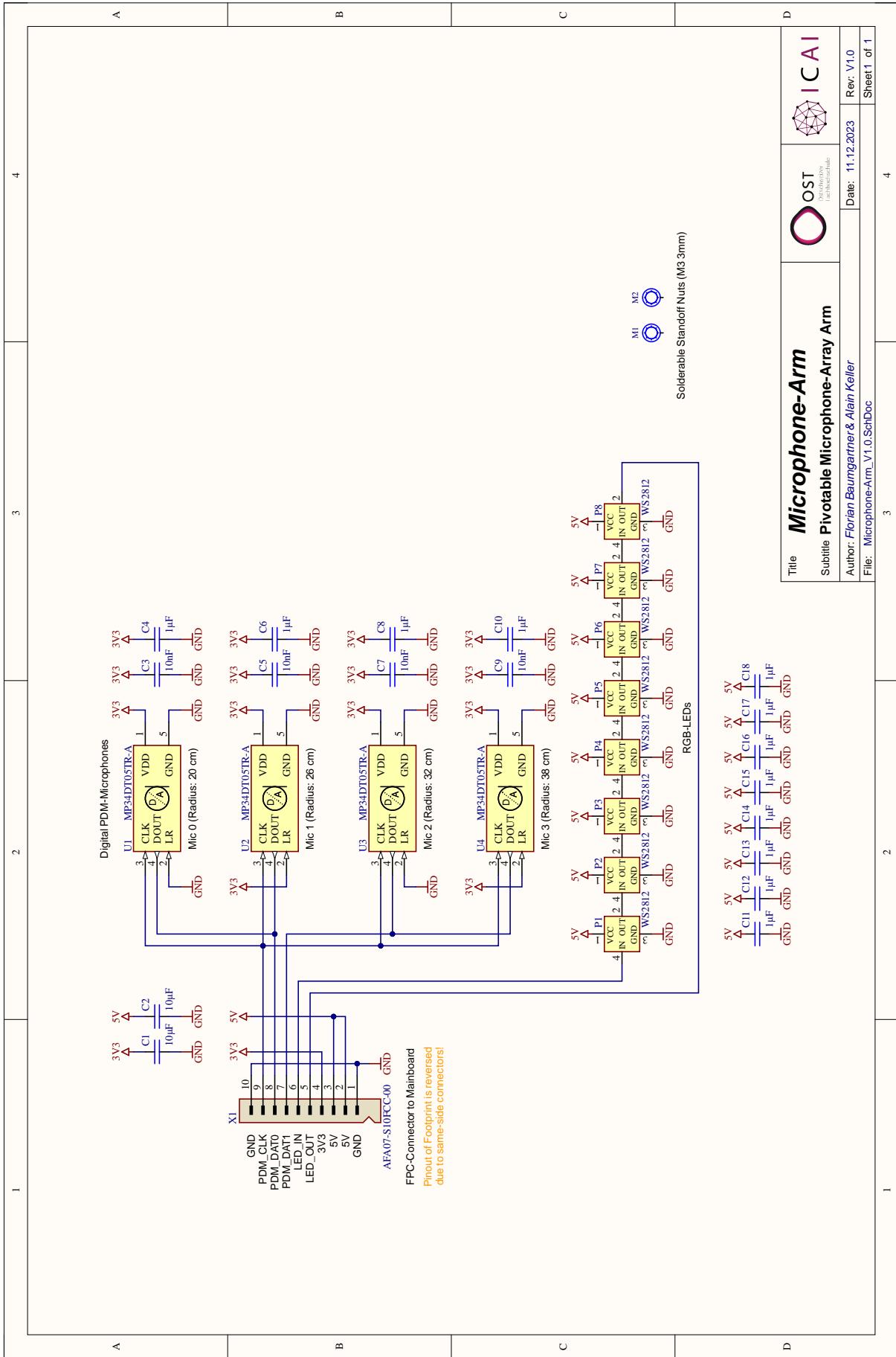
A.28 Mainboard PCB Outline

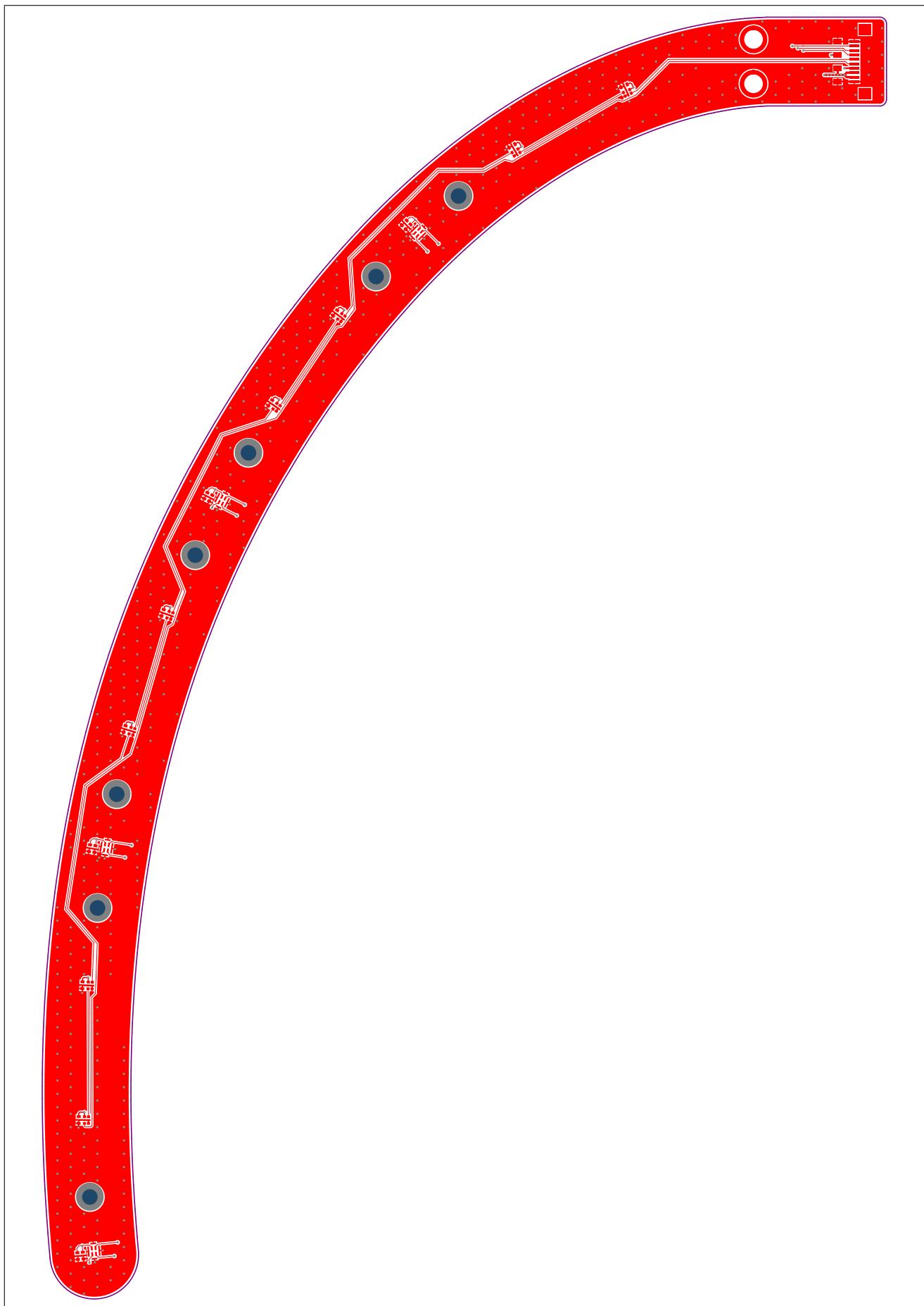


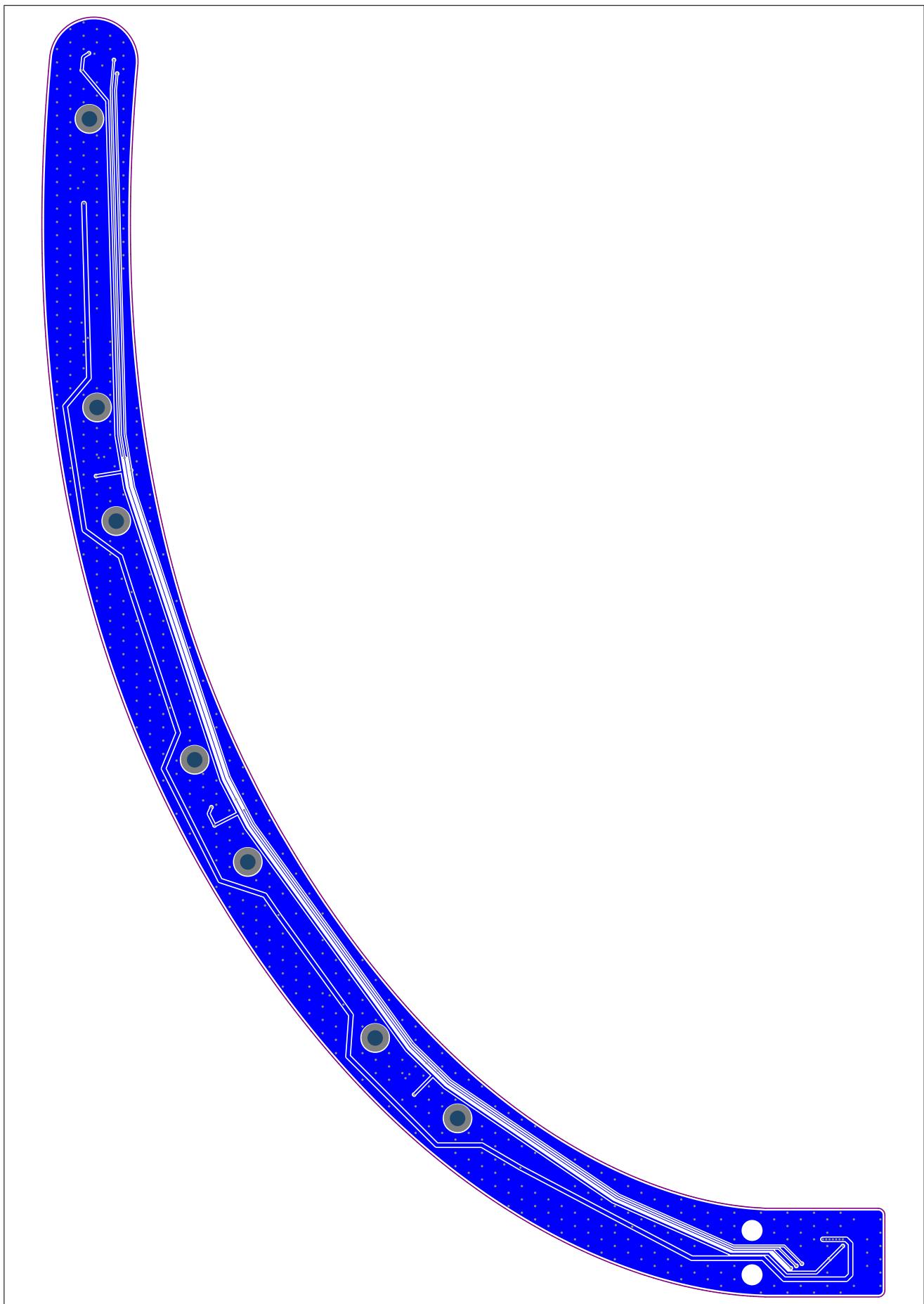
A.29 Mainboard Bill of Materials (BOM)

Bill of Materials for Project [Mainboard.PjrPcb]										
Mainboard		Mainboard Bill of Materials (BOM)								
Designator:	Date:	Mainboard Bill of Materials (BOM)								
Designator:	Date:	Component Description	Value	Footprint	J-L-C Parts	Part Ref.	Quantity	Unit Price	Total Price	
C1..C8, C24, C25, C31..C52, C55..C7A, C7B/C7C, C7D..C7A, C8B/C8C, C8E/C8D	22.01.2024	Multilayer Ceramic Capacitors M/C/C, Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C1463	2110822, 2522432	810	0.01	81.00	
C3, C5, C7, C18, C29, C50..C53, C56..C58, C60B..C60C, C60D..C60E, C60F..C60G	22.01.2024	Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C1506	2110822, 2522432	15	0.01	15.00	
C3, C5, C7, C18, C29, C50..C53, C56..C58, C60B..C60C, C60D..C60E, C60F..C60G	22.01.2024	Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C17224	2110822, 2522432	2	0.01	2.00	
C10..C12	22.01.2024	Electronic Capacitor	100nF 100V	SMD Kermikondensator 0603	C4567161	1657939	77V/100V/104MKBAT	1	0.01	0.06
C11..C14	22.01.2024	Electronic Capacitor	100nF 100V	SMD Kermikondensator 0603	C3832	2470432	81V/100V/104MKBAT	2	0.01	0.06
C15..C18	22.01.2024	Electronic Capacitor	15pF 50V	SMD Kermikondensator 0603	C1585	2455474	81V/100V/104MKBAT	2	0.01	0.06
C19..C21	22.01.2024	Electronic Capacitor	100nF 100V	SMD Kermikondensator 0603	C16250	2470440	81V/100V/104MKBAT	2	0.01	0.06
C22..C24	22.01.2024	Electronic Capacitor	100nF 100V	SMD Kermikondensator 0603	C16234	2484195	81V/100V/104MKBAT	4	0.01	0.06
C25..C28	22.01.2024	Electronic Capacitor	100nF 100V	SMD Kermikondensator 0603	C15449	2484195	81V/100V/104MKBAT	20	0.01	20.00
C29..C31	22.01.2024	Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C16121	2104587, 2124646	445-1782-1ND, 445-1782-1ND	1	0.01	0.01
C32..C34	22.01.2024	Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C17045	2104587, 2124646	445-1782-1ND	1	0.01	0.01
C35..C37	22.01.2024	Multilayer Ceramic Capacitors M/C/C	100nF 50V	SMD Kermikondensator 0603	C15045	2104587, 2124646	445-1782-1ND	3	0.01	0.03
D1..D4, D5..D8, D10..D12	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C15625	1655793	1655-1360-1ND	9	0.01	0.27
D8	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C21598	2390563	760-1448W-HF	1	0.01	0.01
D13	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C216583	2390563	595-1759-1ND	1	0.01	0.01
D14	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C2858	2390563	815-05819W-TP	1	0.01	0.01
D15	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C28628	2390563	295-5515-1ND	1	0.01	0.01
K1..K3	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C232315	512-00017ASD	785-1000-1ND	1	0.01	0.01
K4..K6	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C2645	2173822	512-00017ASD	1	0.01	0.01
K7..K9	22.01.2024	Surface Mount Diode	SOD-123	SMD Diode SOD-123	C2645	1838838	1934-1487-1ND	3	0.01	0.03
L1..L2, L3	22.01.2024	Fast Switching Diode	SOD-123	SMD Diode SOD-123	C21669	21717784	495-2028-1ND	5	0.01	0.05
M1..M2, M4..M5	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	5	0.01	0.05
P1..P2	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P3..P4	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P5..P6	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P7..P8	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P9..P10	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P11..P12	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P13..P14	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P15..P16	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P17..P18	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
P19..P20	22.01.2024	Low Voltage Adjustable Precision Shunt Regulator	T784150CBZR	T784150CBZR	C216583	153-AMT-SCM-4ET	50-2025-1ND	2	0.01	0.02
T21	22.01.2024	Red LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	17	0.01	0.01
C1	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
E1..E2	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
F1..F4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
G1..G4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
H1..H4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
I1..I4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
J1..J4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
K1..K4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
L1..L4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
M1..M4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
N1..N4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
O1..O4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
P1..P4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
Q1..Q4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
R1..R4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
S1..S4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
T1..T4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
U1..U4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
V1..V4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
W1..W4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
X1..X4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
Y1..Y4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
Z1..Z4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
A1..A4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
B1..B4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
C1..C4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
D1..D4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
E1..E4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
F1..F4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
G1..G4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
H1..H4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
I1..I4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
J1..J4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
K1..K4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
L1..L4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
M1..M4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
N1..N4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
O1..O4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
P1..P4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
Q1..Q4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
R1..R4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
S1..S4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
T1..T4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
U1..U4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.01	0.01
V1..V4	22.01.2024	General Purpose LED	PLR-100-1000mW	SMD LED 0603	C2886	1465988	359-2028-107F	1	0.0	

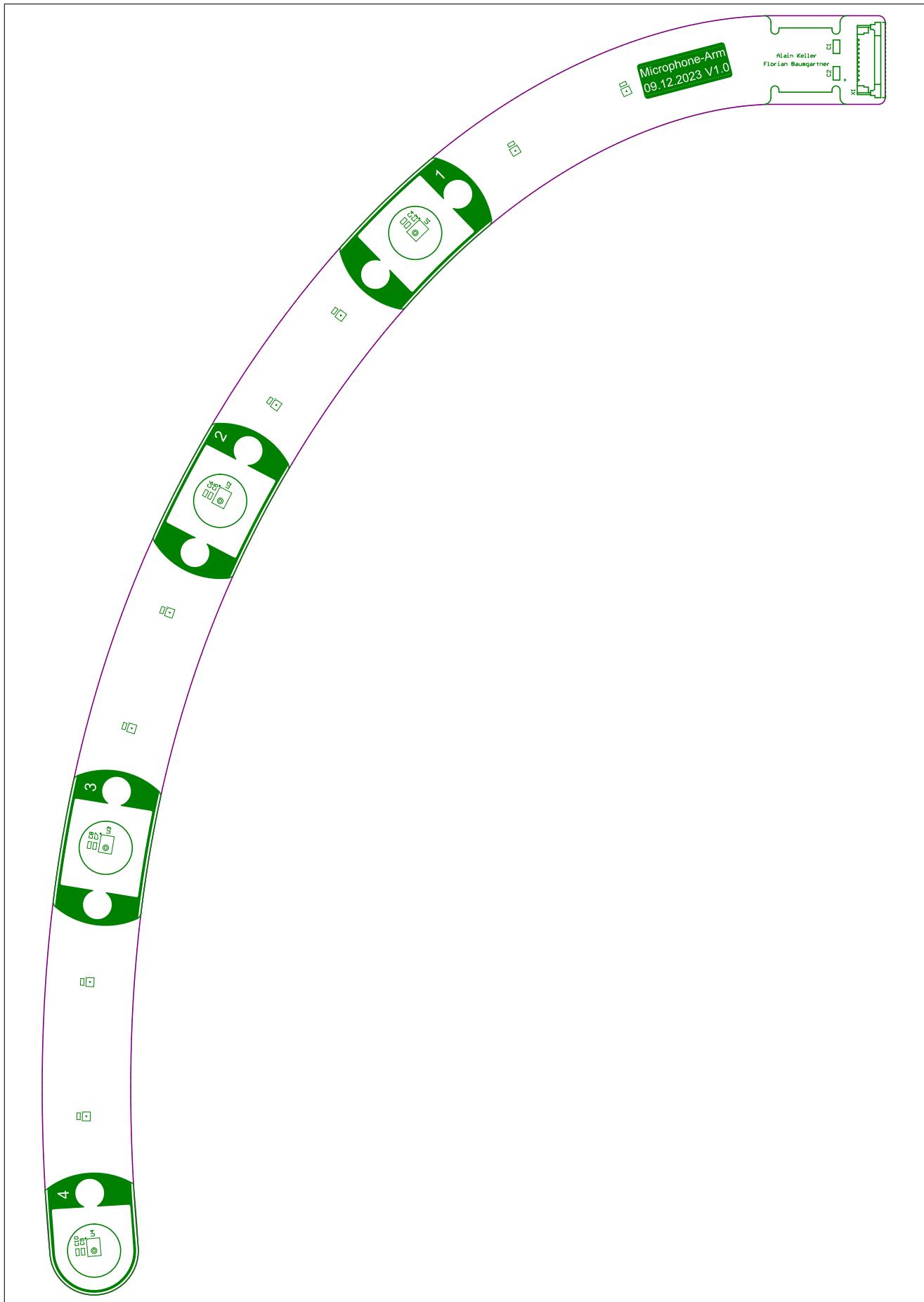
A.30 Microphone-Arm Schematics



A.31 Microphone-Arm PCB Top-Layer

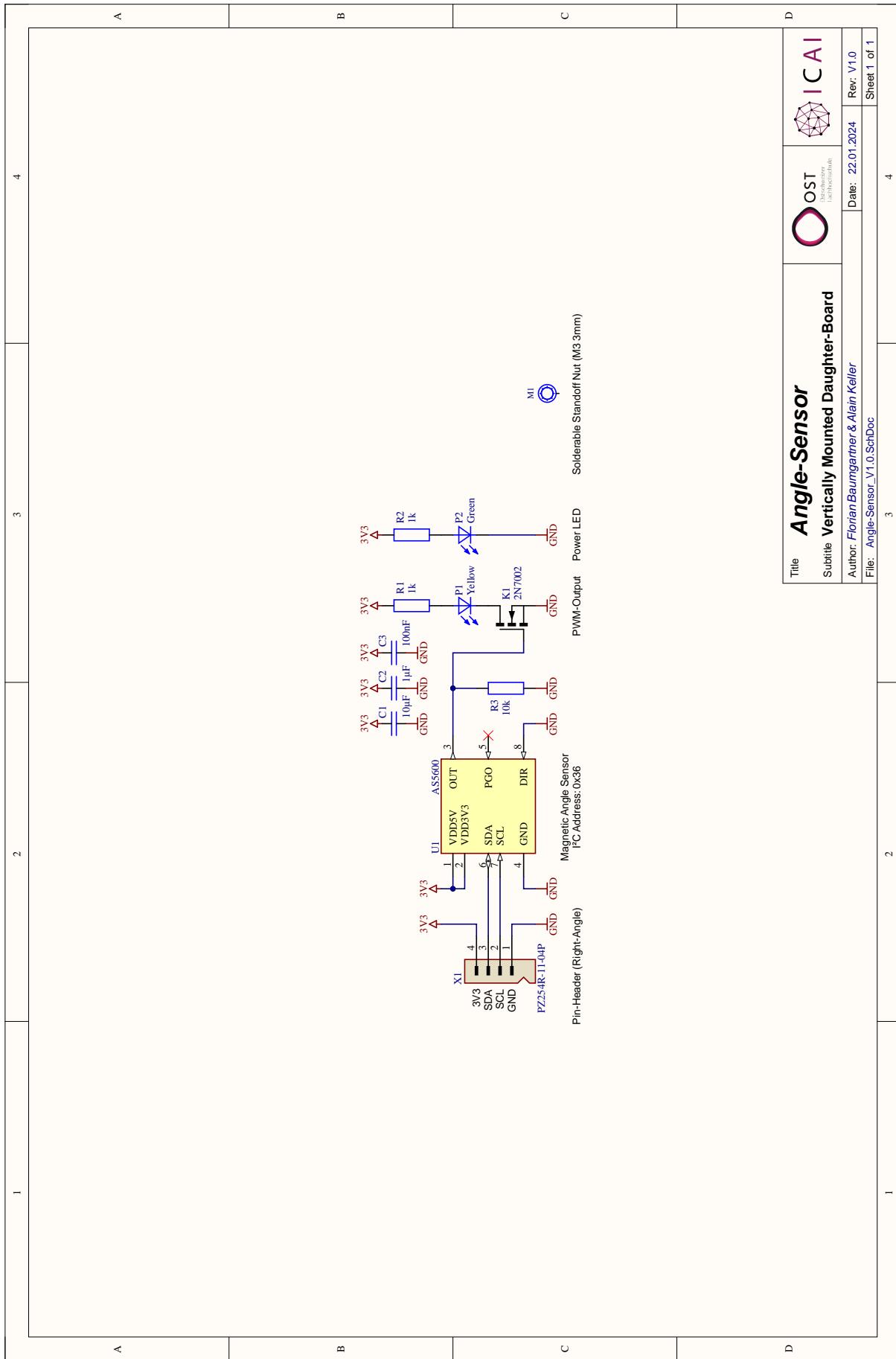
A.32 Microphone-Arm PCB Bottom-Layer

A.33 Microphone-Arm PCB Top-Overlay

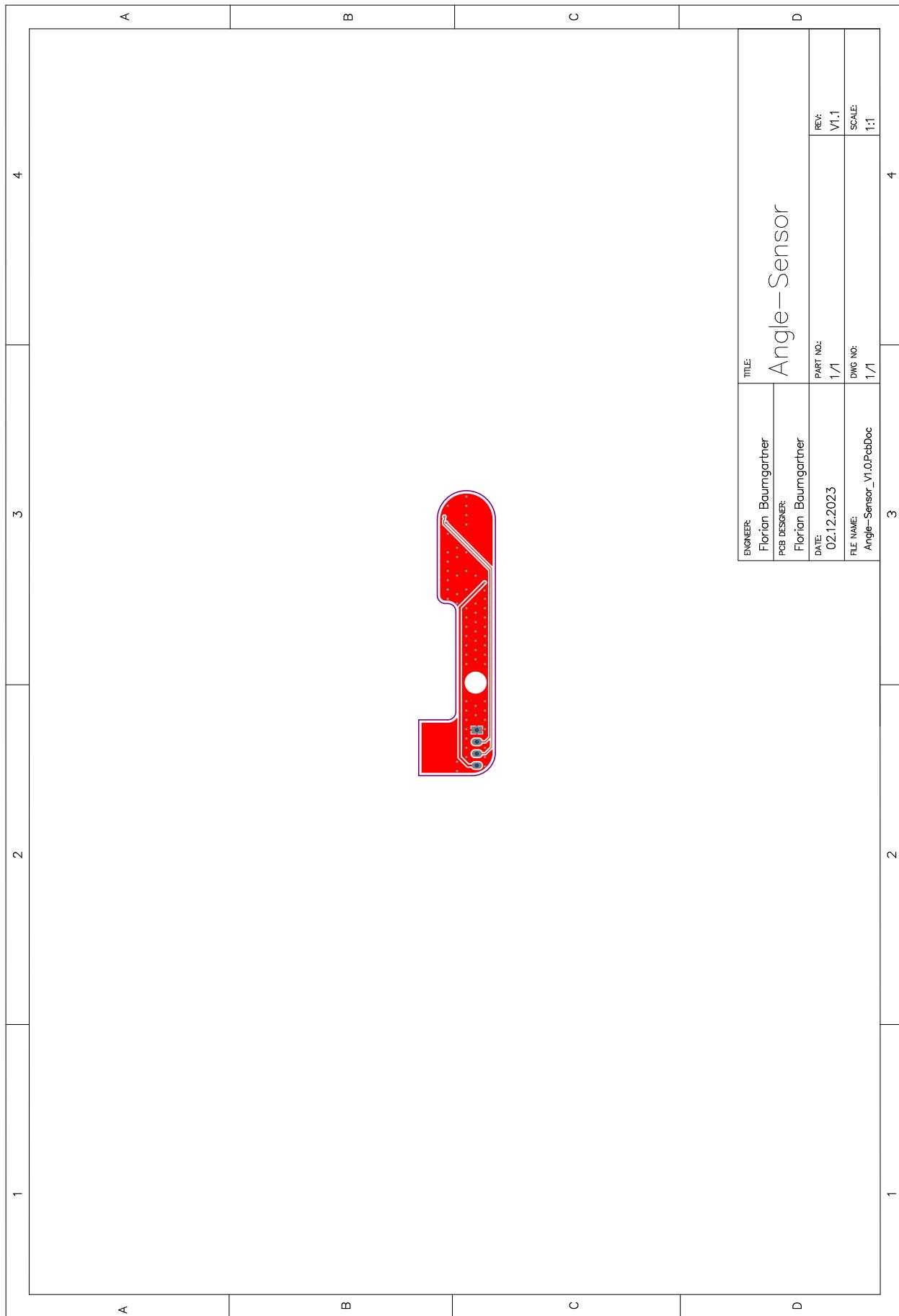


A.34 Microphone-Arm Bill of Materials (BOM)

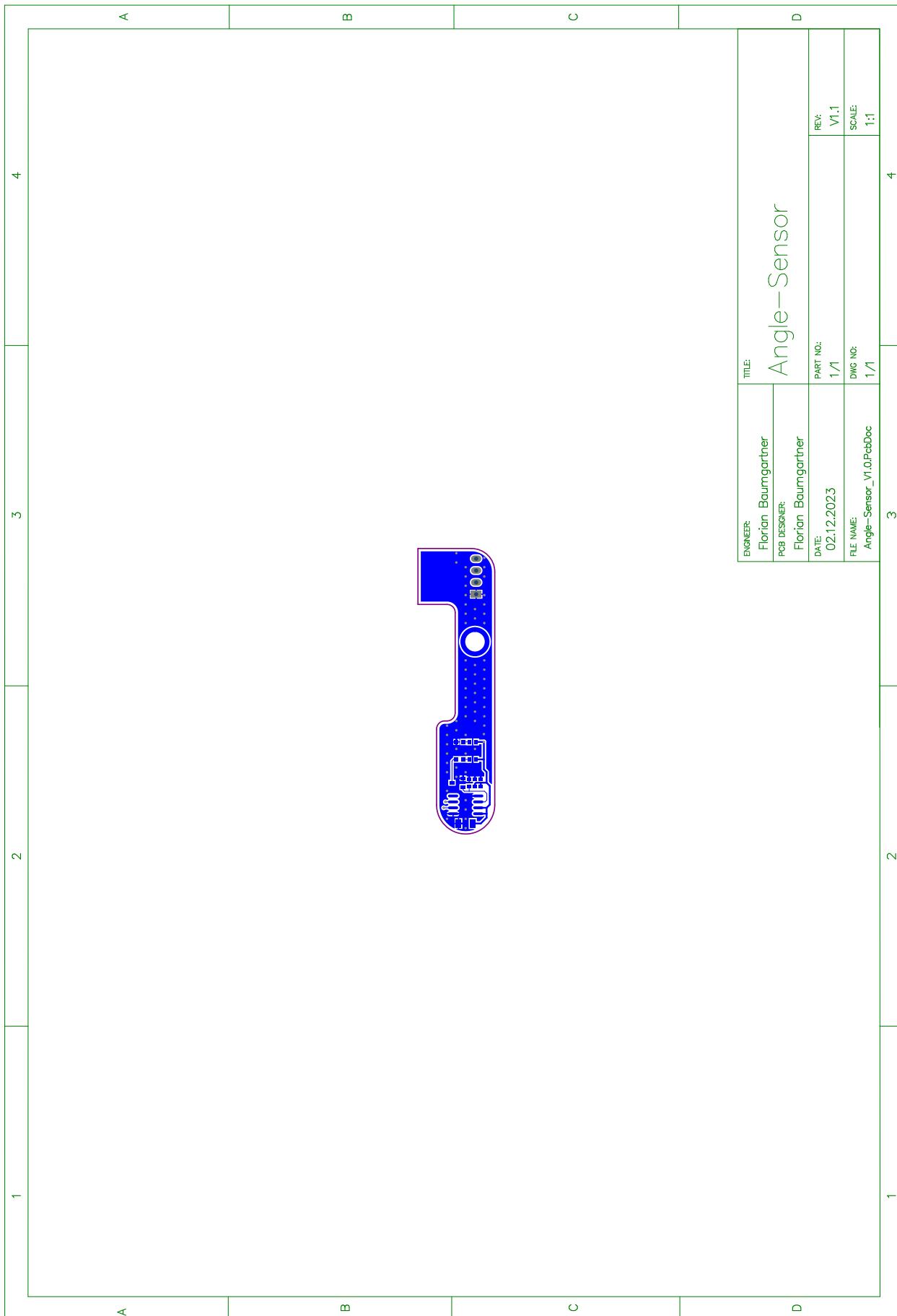
A.35 Angle-Sensor Schematics



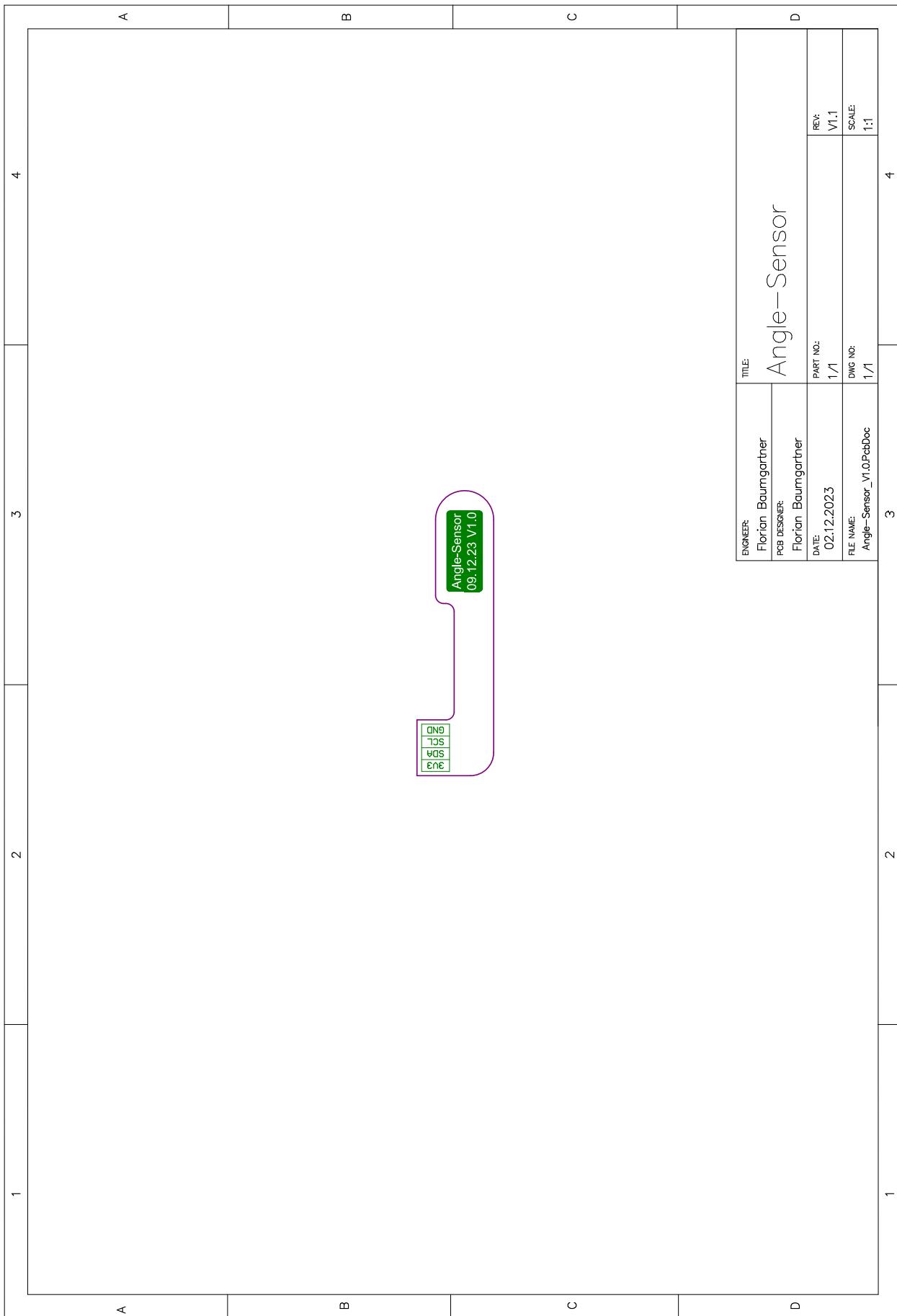
A.36 Angle-Sensor PCB Top-Layer



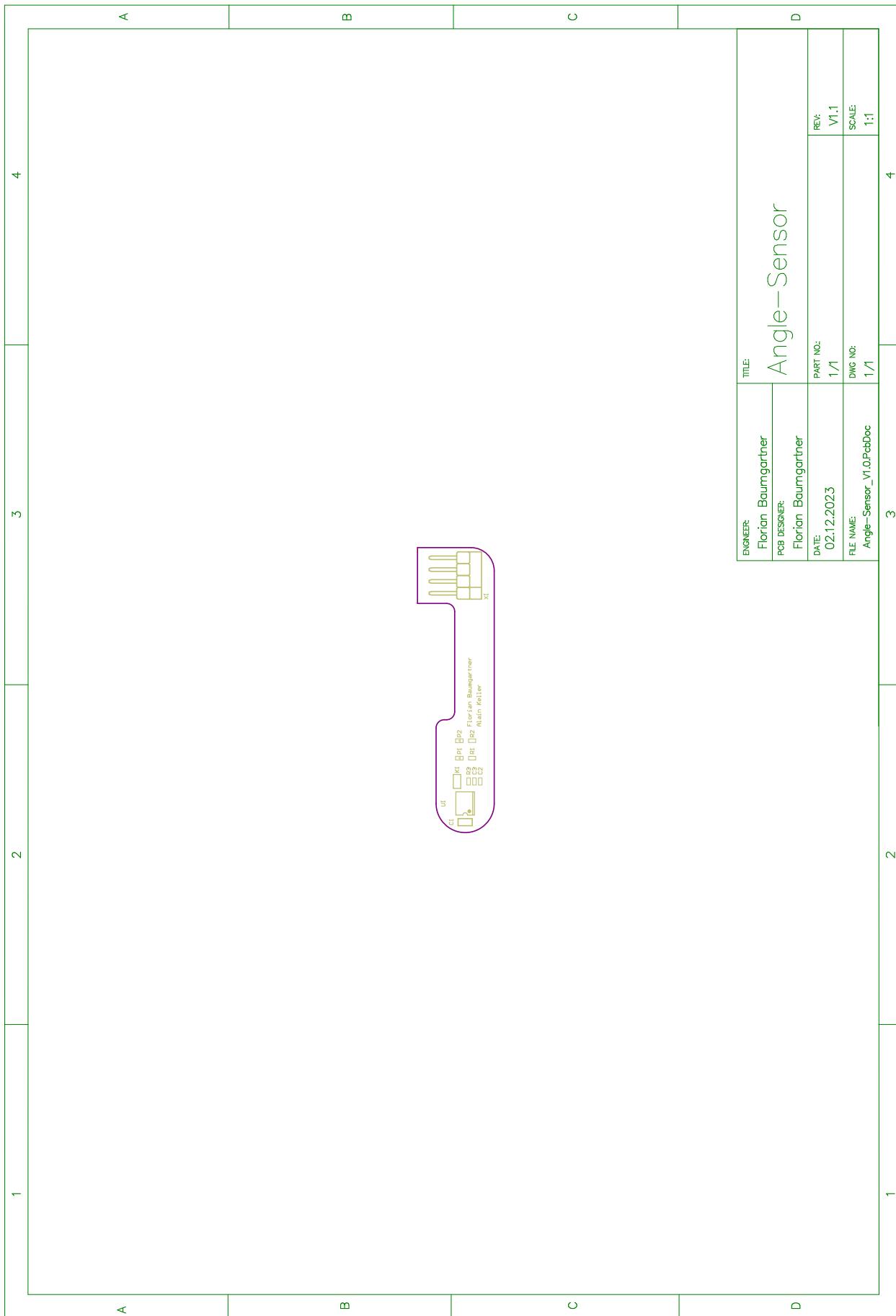
A.37 Angle-Sensor PCB Bottom-Layer



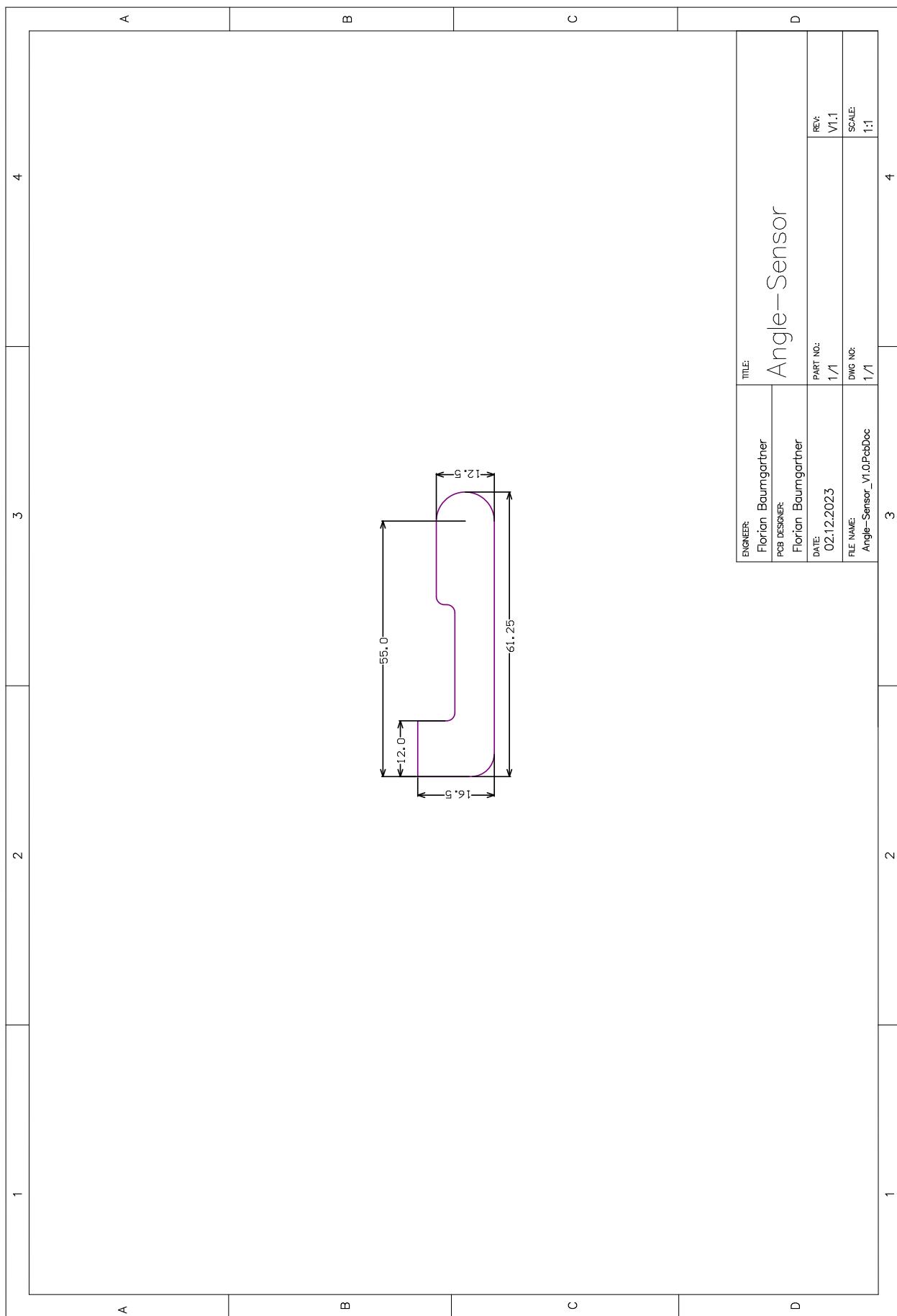
A.38 Angle-Sensor PCB Top-Overlay



A.39 Angle-Sensor PCB Bottom-Overlay



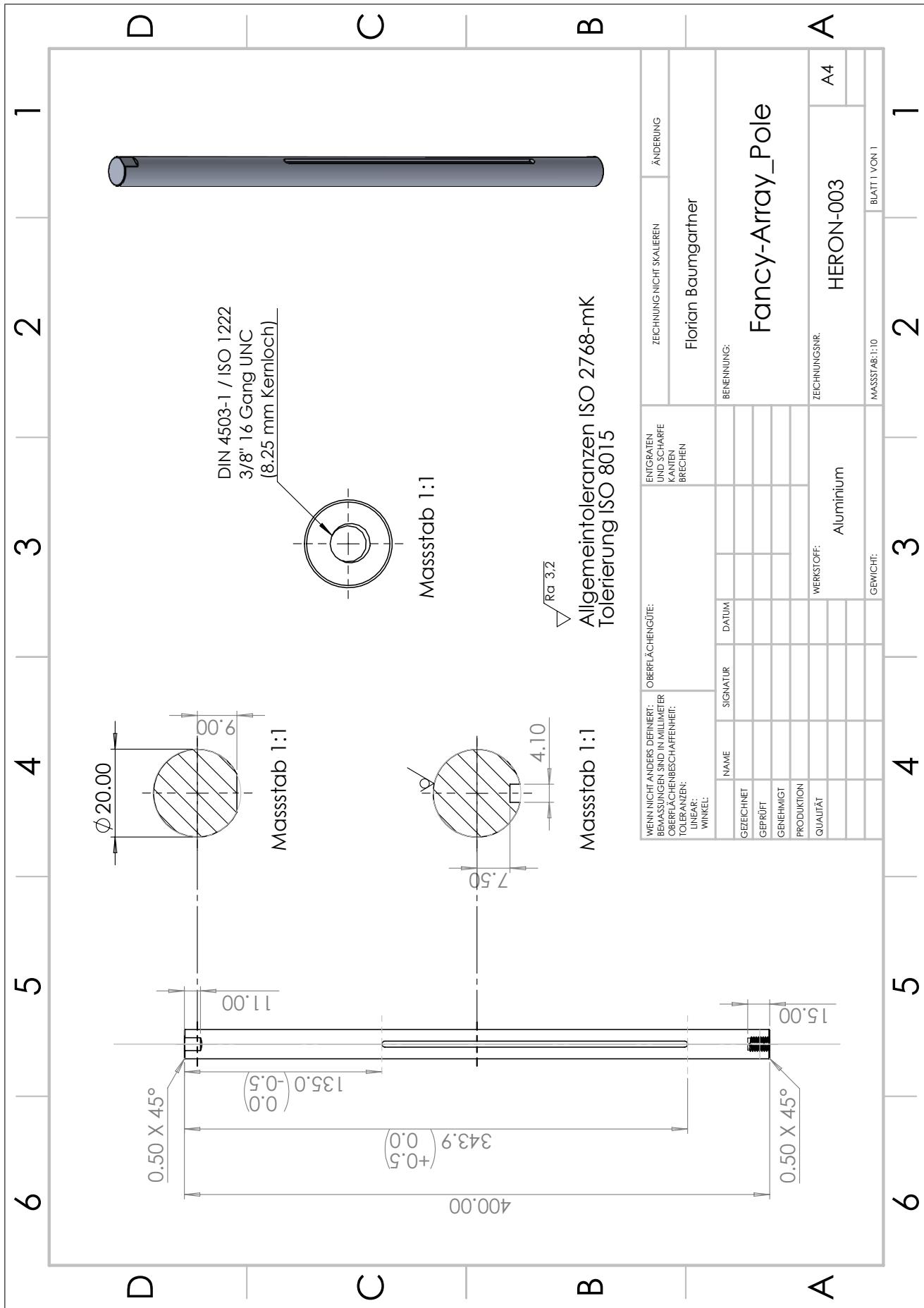
A.40 Angle-Sensor PCB Outline



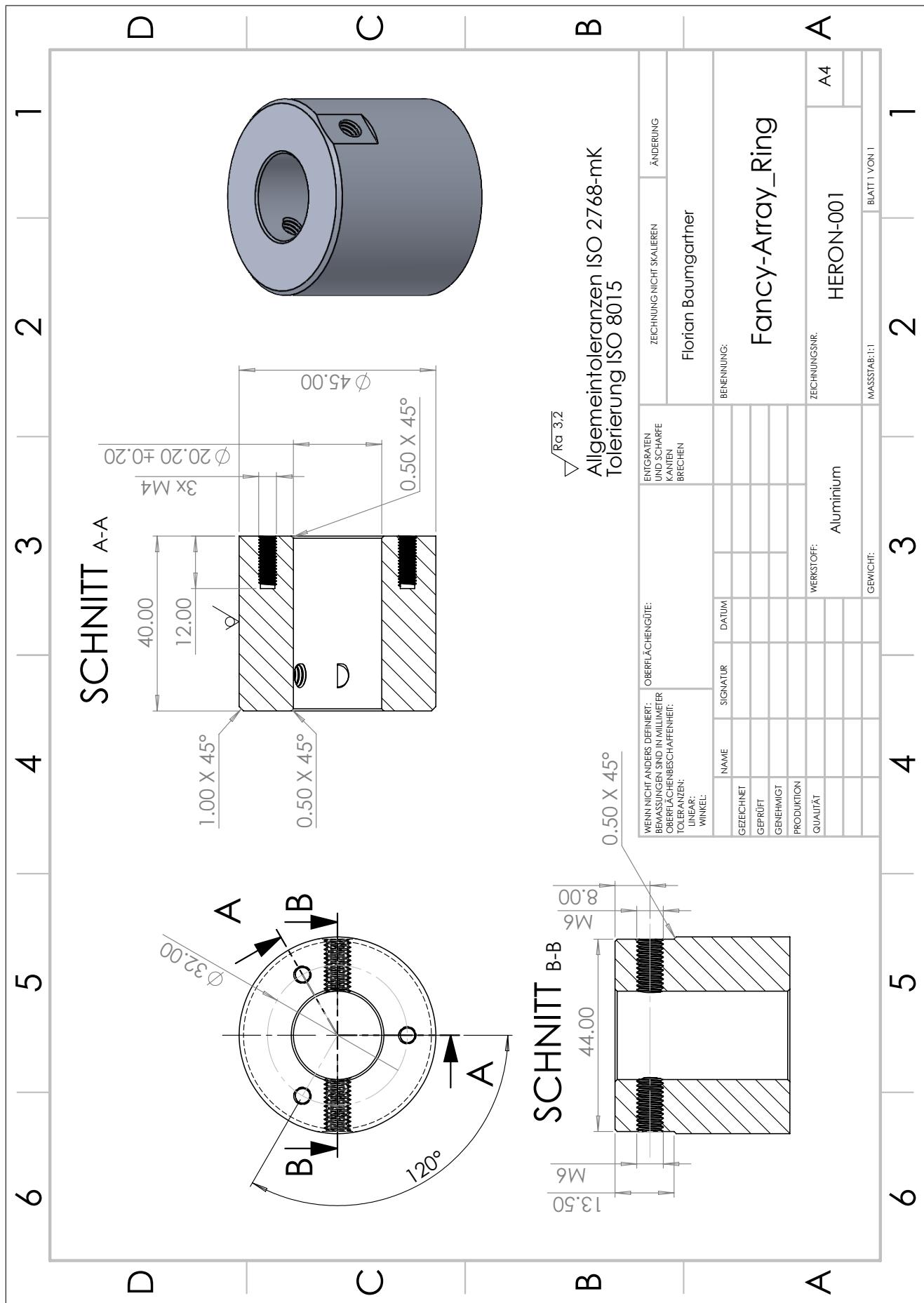
A.41 Angle-Sensor Bill of Materials (BOM)

Bill of Materials								
Bill of Materials for Project [Angle-Sensor.PjPcb]								
File name: Angle-Sensor.PjPcb Version: V1.0		Print Date: 22.01.2024 Creation Date: 22.01.2024						
Component Description							Total Price	
Designator	Component Value	Footprint	J.C.-Parts	Farnell	Mouse	Digikey	Quantity	Unit Price
C1	MLCC 100μF 50V	SMD Keramikkondensator 1206	C1-3685	21113907061000ALT	963-LJMK107EB1061A1ND	587-325B-1-ND	1	CHF 0.01
C2	Multi-layer Ceramic Capacitors	Taif 50V	C1-3649	3013383	187-CQ110A105KBNNINC	1276-1890-1-ND	1	CHF 0.01
C3	Multi-layer Ceramic Capacitors	100nF 50V	C1-4663	2524232	772A063011042XAPBC	446-568B-2-ND	1	CHF 0.01
C4	MOSFET N-Channel	2N7002	C1-3645	17138623	512-2970002	2N7002HC1G7002	1	CHF 0.02
C5	Surface Mount Standoff Nut	M3 3mm	C1-3608	155-SAMSG-A143-3-ET	155-SAMSG-A143-1-ET	1	CHF 0.05	
C6	General Purpose LED	Yellow 2mA, 2.1V	C1-2349	1967103	645-598-80369-107-E	350-2933-2-ND	1	CHF 0.34
C7	General Purpose LED	Green 2mA, 2.1V	C1-2443	1465901	645-598-8070-107-E	350-2935-2-ND	1	CHF 0.34
C8	Surface Mount Standoff	1K (80% Thread, 1%)	C1-159	230345	667-ER-31K-F00U	667-ER-31K-F00U	2	CHF 0.02
C9	Thick Film Chip Resistor	10k 5% 0.05W, 1%	C1-3604	2303192	667-ER-31K-F00V	667-ER-31K-F00V	1	CHF 0.01
C10	Digital Magnetic Angle Sensor	AT56100	C1-3646	4156825	4156825	AT56100-ASDMT-TND	1	CHF 1.32
C11	Pin Header (Right-Angle)	P225-R-11-QFP	C1-22412	3226010	855-M20-919445	952-3285-ND	1	CHF 0.02
Notes:							CHF 0.15	
Supplier:							Ostschweizer Fachhochschule	
Author:							Franan Baumgartner & Alain Keller franam.baumgartner@ost.ch	
Logos:							 	

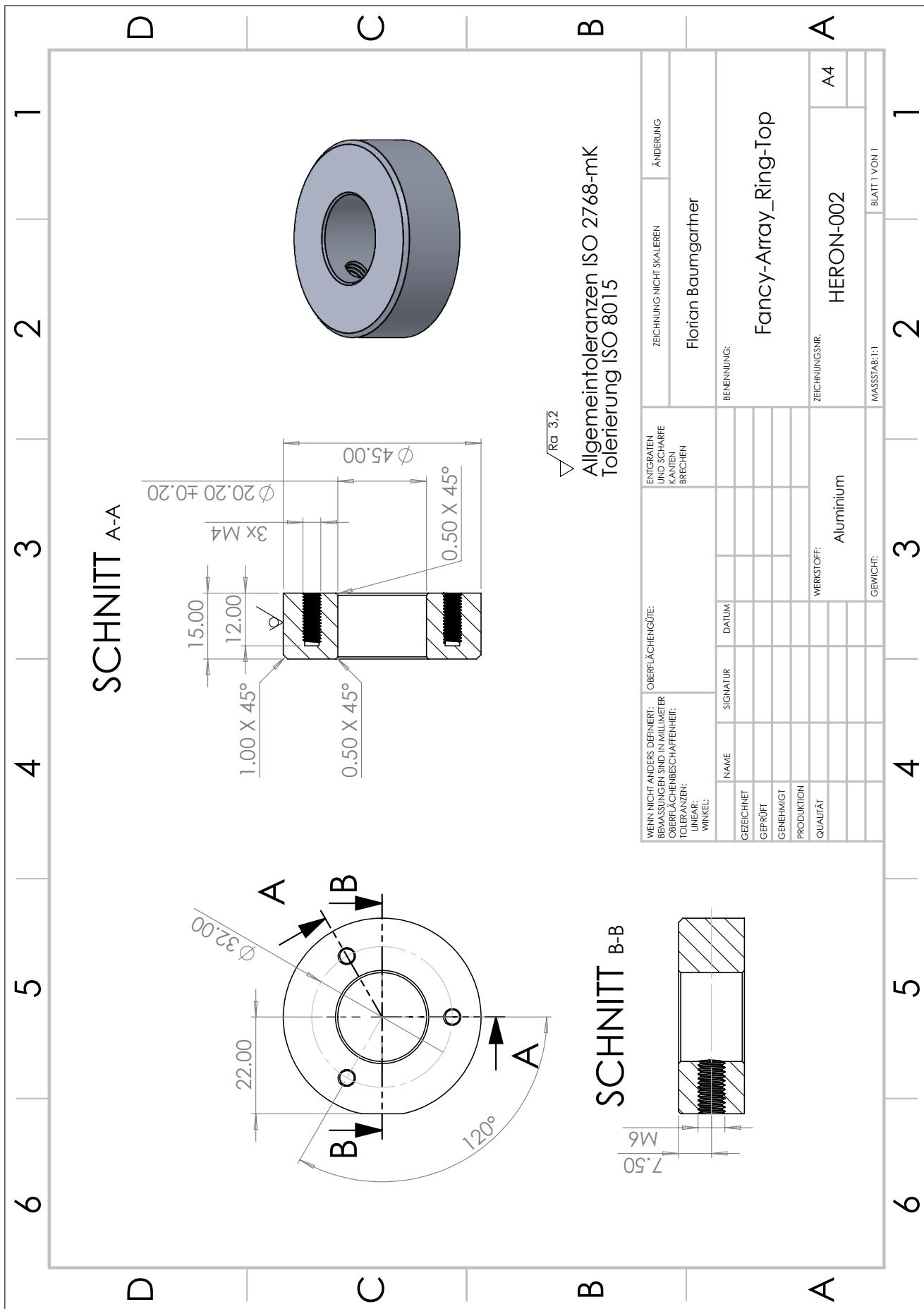
A.42 Mechanical Drawing of Main Mounting Pole



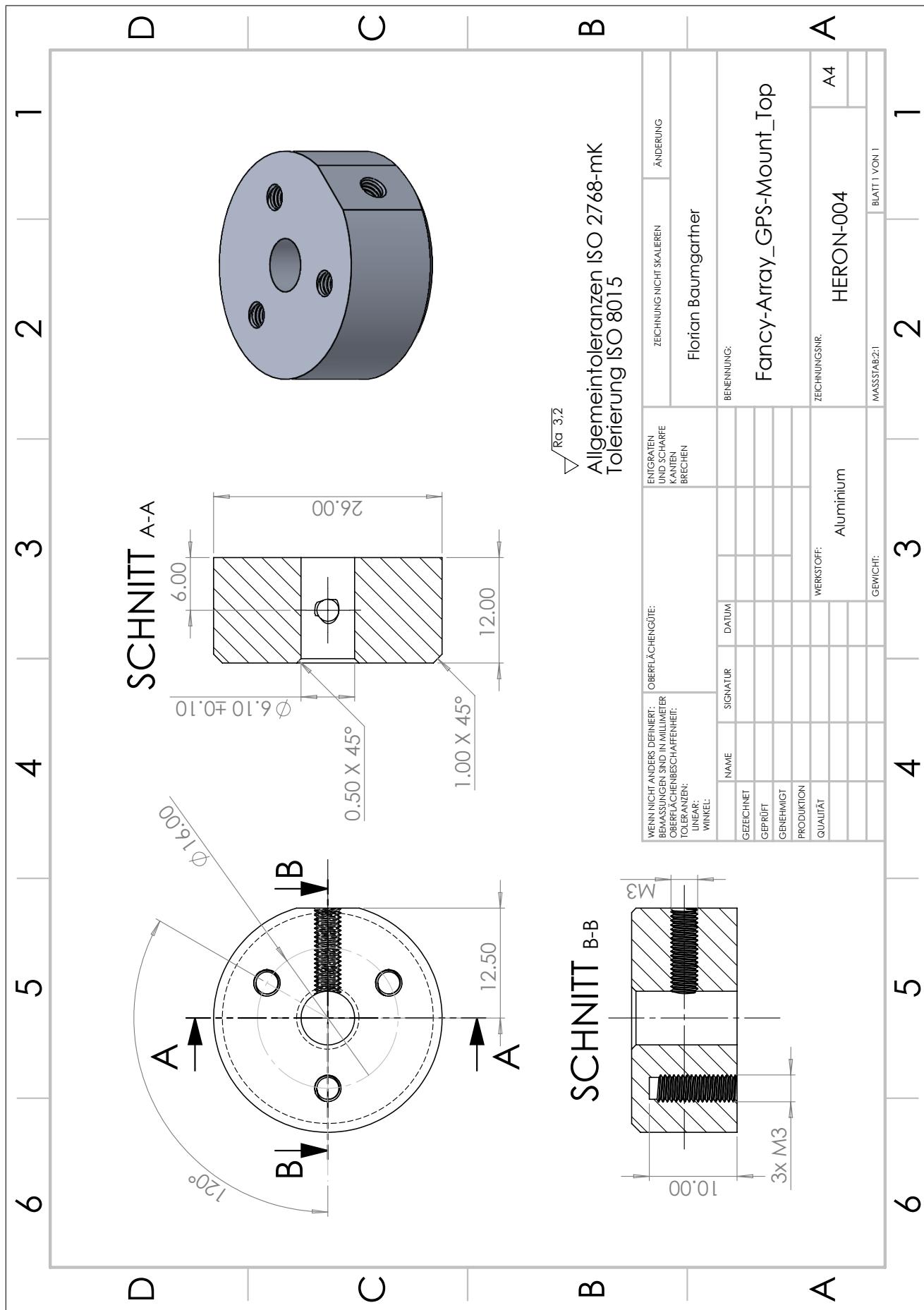
A.43 Mechanical Drawing of Top Mounting Ring



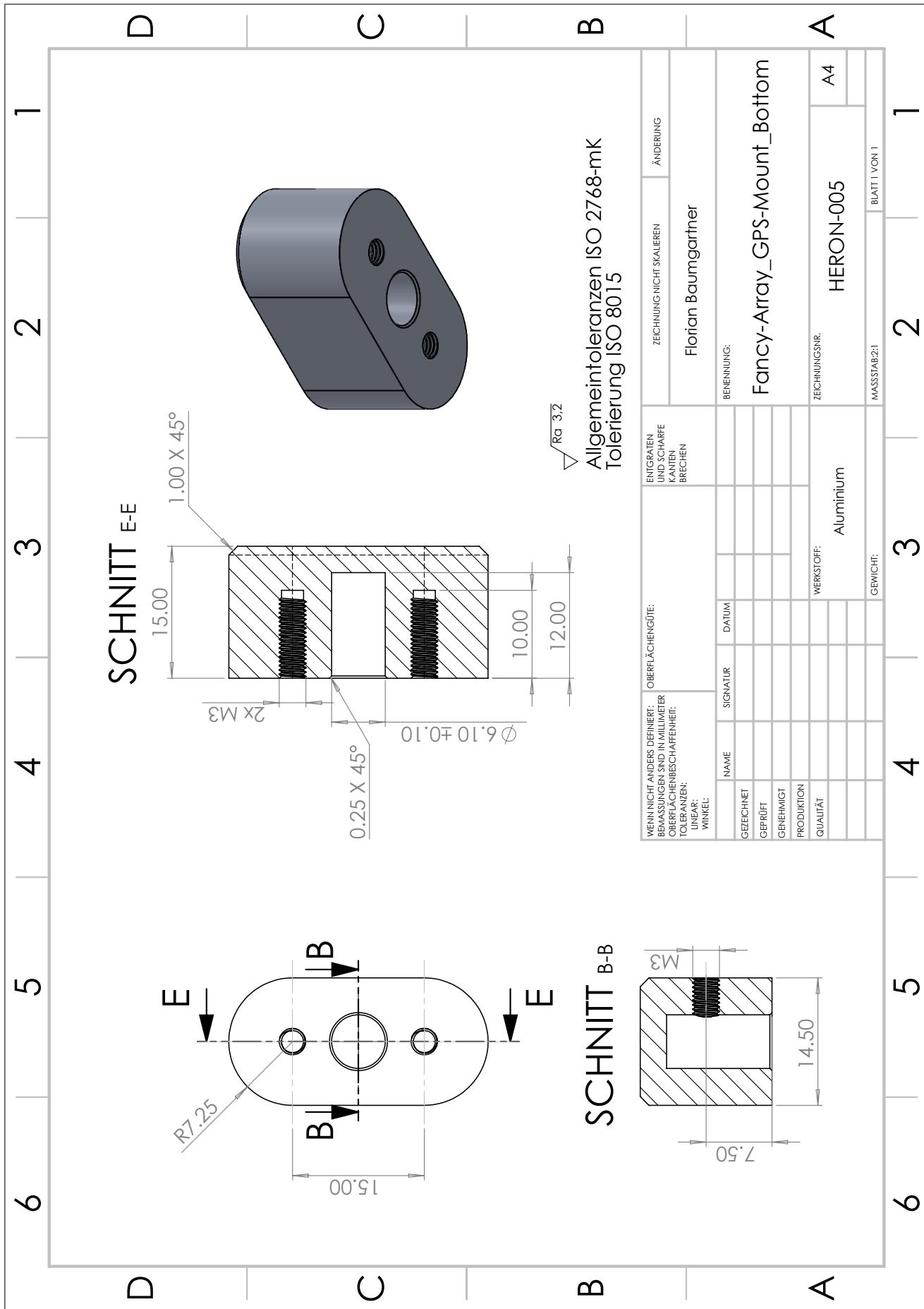
A.44 Mechanical Drawing of Bottom Sliding Ring



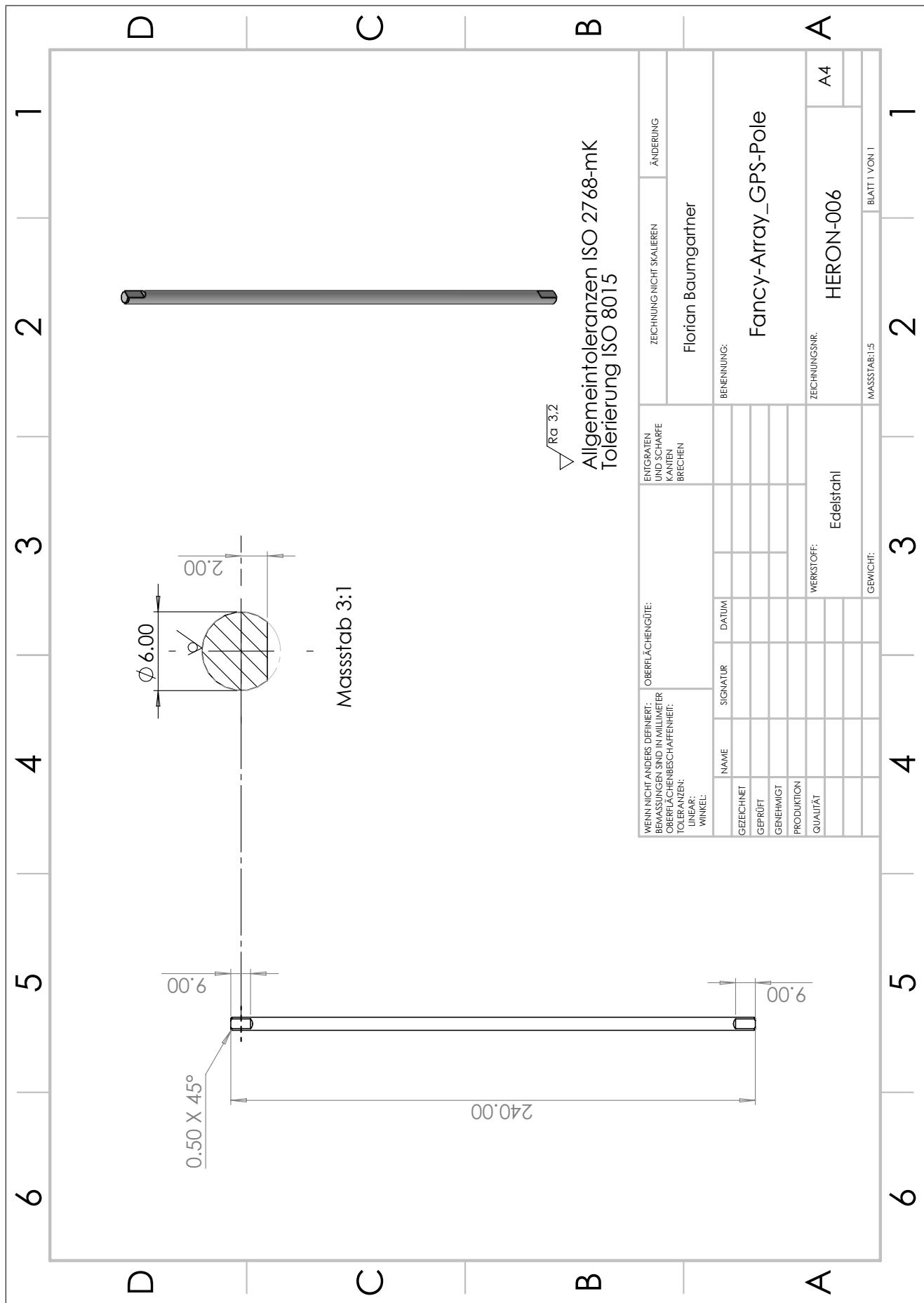
A.45 Mechanical Drawing of Antenna Top Mount



A.46 Mechanical Drawing of Antenna Bottom Mount



A.47 Mechanical Drawing of Antenna Pole



Bibliography

- [1] M. Kühne A. Müller. Mathematisches Seminar Matrizen, 2021.
- [2] Filipe Ramos do Amaral, Juan Carlos Serrano Rico, and Marcello Augusto Faraco de Medeiros. Design of microphone phased arrays for acoustic beamforming. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 40(7):354, Jun 2018.
- [3] AS5600 Position Sensor. <https://ams.com/as5600>. (accessed: 20.01.2024).
- [4] Calibration of an Embedded Compass Magnetometer. <https://www.nxp.com/docs/en/application-note/AN4246.pdf>. (accessed: 22.01.2024).
- [5] Comparing MEMS and Electret Microphones. <https://www.cuidevices.com/blog/comparing-mems-and-electret-condenser-microphones>. (accessed: 22.01.2024).
- [6] crackwitz. Locating local maximums - python - opencv. <https://forum.opencv.org/t/locating-local-maximums/1534>. (accessed: 22.01.2024).
- [7] Junfenga Guo, Ahmad, Ishtiaq, Chang, and KyungHi. Classification, positioning, and tracking of drones by hmm using acoustic circular microphone array beamforming. *EURASIP Journal on Wireless Communications and Networking*, 2020(9), 2020.
- [8] Mathworks. PDesign Fractional Delay FIR Filters. <https://ch.mathworks.com/help/dsp/ug/design-of-fractional-delay-fir-filters.html>. (accessed: 23.01.2024).
- [9] Zebb Prime and Con Doolan. A comparison of popular beamforming arrays. In Terrance McMinn, editor, *Science Technology and Amenity, Annual Conference of the Australian Acoustical Society*. Australian Acoustical Society, November 2013.
- [10] Boaz Rafaely, Yotam Peled, Morag Agmon, Dima Khaykin, and Etan Fisher. *Spherical Microphone Array Beamforming*, pages 281–305. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [11] B. Ruh, A. Bucher. Physik 2, 2014.
- [12] Jelmer Tiete, Federico Domínguez, Bruno Da Silva, Laurent Segers, Kris Steenhaut, and Abdellah Touhafi. Soundcompass: A distributed mems microphone array-based sensor for sound source localization. *Sensors*, 14(2):1918–1949, 2014.