

Embedded, Modular, and Affordable High-Density Loudspeaker Arrays

Maxime Popoff
maxime.popoff@inria.fr
Independent researcher
69004 Lyon, France

Pierre Cochard
pierre.cochard@insa-lyon.fr
INSA Lyon, Inria, CITI, UR3720
69621 Villeurbanne, France

Romain Michon
romain.michon@inria.fr
Inria, INSA Lyon, CITI, UR3720
69621 Villeurbanne, France

Tanguy Risset
tanguy.risset@insa-lyon.fr
INSA Lyon, Inria, CITI, UR3720
69621 Villeurbanne, France

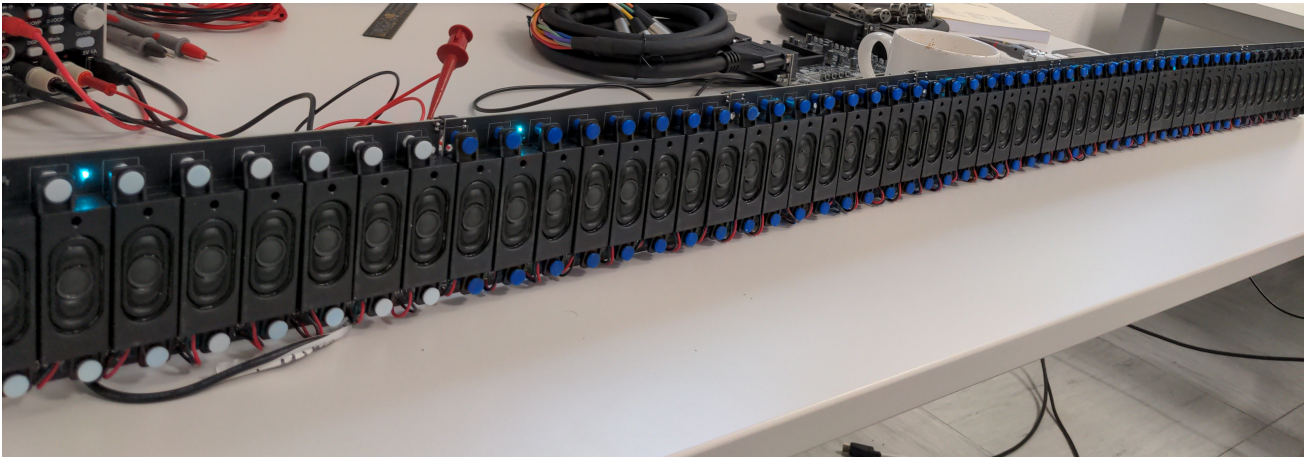


Figure 1: A loudspeaker array in a linear configuration with 64 loudspeakers.

Abstract

Scalable loudspeaker arrays for spatial audio research typically require complex multichannel hardware, limiting the practical deployment of high-density configurations. We introduce a modular loudspeaker array design that supports a wide range of geometries without hardware modification. The system integrates an embedded FPGA system-on-module with compact 8-channel loudspeaker printed circuit boards (PCBs) that can be easily daisy-chained. Each module uses PCB-mounted drivers to maximize spatial density and ensure consistent acoustic and mechanical characteristics across the array. Controlling such a system with hundreds of loudspeakers requires handling timing-critical operations and distributed, synchronized audio streams with minimal I/O overhead. This is achieved using an embedded Xilinx Zynq-7000 FPGA: the current prototype has been tested with 64 speakers, and the architecture supports configurations of up to 768 channels. The modular PCB design enables rapid reconfiguration into linear, circular, or two-dimensional layouts without modifying the underlying signal-distribution or processing architecture. This platform provides an accessible, open-source foundation for evaluating spatial audio applications under high-channel-count conditions.



This work is licensed under a Creative Commons Attribution 4.0 International License.

NIME '26, June 23–26, 2026, London, UK

© 2026 Copyright held by the owner/author(s).

CCS Concepts

• **Hardware** → **Hardware description languages and compilation**; • **Computer systems organization** → **Reconfigurable computing**; • **Applied computing** → **Sound and music computing**.

Keywords

Loudspeaker Arrays, Spatial Audio, FPGA, WFS, Real-Rime Audio DSP

ACM Reference Format:

Maxime Popoff, Romain Michon, Pierre Cochard, and Tanguy Risset. 2026. Embedded, Modular, and Affordable High-Density Loudspeaker Arrays. In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME '26)*. ACM, New York, NY, USA, 7 pages.

1 Introduction

While immersive audio was once the prerogative of academic institutions, it is progressively becoming “the norm” in concert halls, movie theaters, etc. [2, 6, 14] Some immersive audio techniques such as Ambisonics [5] or Vector-Based Amplitude Panning (VBAP) [12] can be deployed using a limited number of speakers (a few dozen, in general). Other techniques, such as beamforming [8] or Wave Field Synthesis (WFS) [1] provide enhanced results [6], but require a greater number of speakers to work. In this context, some venues are now hosting high-density speaker arrays involving hundreds of speakers (not to say thousands in some cases). The Las Vegas Sphere¹, which hosts a total

¹<https://www.thesphere.com/>

of 167,000 speakers, is a good example of that. Targeting a large number of speakers for immersive audio applications is a complex task which requires the use of expensive hardware. The overall system cost and complexity increase rapidly with the number of channels, making configurations beyond 32 loudspeakers difficult to access for small institutions, independent artists, and hobbyists.

In 2023, we introduced “the Space Bar” [7, 10, 13], a fixed 32 speakers linear WFS array based on an Field-Programmable Gate Array (FPGA) board and a custom Printed Circuit Board (PCB).

This paper presents an evolution of this project where a modular PCB system can be used to design high-density speaker arrays of different shapes and size targeting WFS. For that, it relies on an embedded FPGA system-on-module (ALINX AC7020C) combined with daisy-chained compact custom 8-channels loudspeaker PCBs allowing for up to 3 array of 256 speakers to be used.

The contributions of this paper can be summarized as follows compared to the initial 2023 project:

- Increased loudspeaker density (twice as high).
- A detachable design enabling easy transport and rapid deployment of the loudspeaker array.
- A modular design allowing flexible array geometries (i.e., linear, matrix, convex, etc.)
- Improved FPGA integration through the use of a system-on-module platform.
- Enhanced integration of WFS with the embedded Linux.

This paper is organized into 4 sections. Section 2 presents the hardware architecture of the proposed loudspeaker array, including the FPGA platform and modular PCB design. Section 3 introduces the software framework used for real-time audio processing and control, along with spatial audio case studies. Section 4 evaluates the system in terms of portability, geometric flexibility, and loudspeaker adaptability. Finally, Section 5 concludes the paper and outlines directions for future work. The complete hardware and software platform presented in this paper is released as open-source and available at the project repository.²

2 Hardware Architecture

The array is decomposed into identical modules, each comprising eight PCB-mounted loudspeakers together with their associated DACs and amplifiers. The modules are daisy-chained to adjust the overall array size, and a single FPGA, connected at one end of the array, performs all signal processing.

2.1 FPGA Platform for Embedded Audio

FPGAs are increasingly used for real-time audio DSP due to their low-latency processing, high computational throughput, support for high sampling rates, and extensive I/O capabilities. Unlike CPU-based systems, FPGA designs typically operate without an operating system and can avoid software buffering, enabling sample-accurate timing with predictable latency. This execution model allows a single FPGA to handle high sampling rates and large numbers of audio channels while maintaining strict inter-channel synchronization, which is essential for dense loudspeaker arrays and spatial audio techniques.

Embedding an FPGA directly within the loudspeaker system reduces cabling complexity and enables compact, tightly synchronized signal distribution. However, FPGA integration remains

significantly more complex than microcontroller- or CPU-based designs, as modern devices require external memory, high-speed interfaces, careful power management, and strict signal integrity constraints. Fully custom FPGA carrier boards designed from scratch are therefore costly and difficult to develop. Development boards are often used, but they introduce additional peripherals and higher cost, and allocate a significant portion of the available GPIOs (General Purpose Input Output) to interfaces intended for generic evaluation rather than for the targeted application.

An effective compromise is provided by *FPGA system-on-module platforms*, which integrate the FPGA, external memory, and clocking infrastructure while exposing application-specific I/O through standardized connectors. This approach enables the design of a dedicated carrier (or I/O) board tailored to the needs of a specific application, without requiring the designer to manage the complexity of the FPGA core hardware. Similar in principle to compute-module-based systems such as the Raspberry Pi Compute Module,³ this strategy significantly reduces design complexity while maintaining control over cost, form factor, and peripheral selection.

For this project, we use an ALINX AC7020C FPGA system-on-module (see Fig. 2), which integrates a Xilinx Zynq-7000 XC7Z020 device and exposes its GPIOs through standard 2x20 2.54 mm-pitch connectors. This module provides sufficient computational resources for our application while remaining easy to integrate. Other modules exist that offer denser connectors or more powerful FPGA devices when higher channel counts or increased processing performance are required.

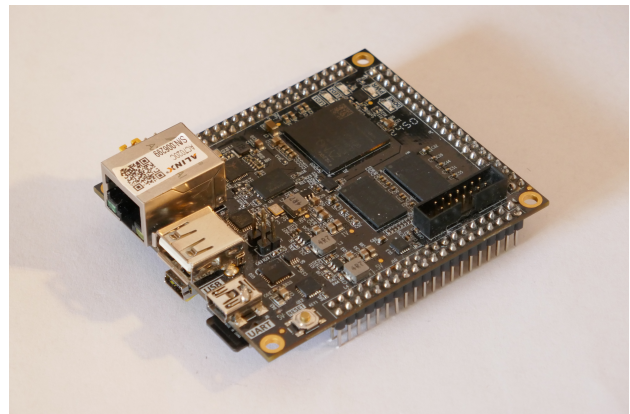


Figure 2: ALINX AC7020C FPGA system-on-module used in this project, integrating the FPGA, DDR memory, and USB/Ethernet peripherals.

2.2 Digital Audio Distribution to the DACs

FPGAs natively process digital signals and therefore require external digital-to-analog conversion and amplification stages to drive loudspeakers. The Inter-IC Sound (I²S) protocol is the most common solution for interfacing digital audio systems. In its standard form, I²S transports two audio channels using three signals: a serial data line, a bit clock (BCLK), and a word-select (WS, or left/right clock).

Time-Division Multiplexing (TDM) extends the I²S protocol by interleaving multiple audio channels on the same data line

²<https://github.com/inria-emmaude/syfala/tree/main/PCB/TheLine>

³<https://www.raspberrypi.com/documentation/computers/compute-module.html>

while preserving the same sampling rate. Depending on the implementation, TDM allows up to 16 (TDM16) audio channels to be transmitted using the same clock signals. This approach makes it possible to share a single I²S bus across multiple digital-to-analog converters or amplifiers, significantly reducing wiring complexity and the number of GPIOs required on the FPGA. When clocks are shared across devices, expanding the system by an additional group of channels only requires adding a new data line, while the existing clock lines remain unchanged.

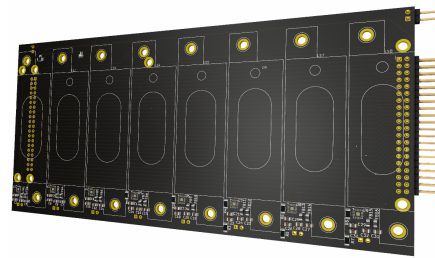
To further reduce system complexity, some audio DAC integrate both a digital-to-analog converter and a power amplifier within a single chip, allowing each loudspeaker to be driven by a single component. In this work, we use the MAX98357A digital audio amplifier (with integrated DAC). These chips were chosen because they are among the few commercially available devices supporting TDM input while integrating a Class-D power stage. This choice simplifies the hardware design but imposes two limitations: the MAX98357A supports TDM8 only (not TDM16), and its compact form factor limits output power to approximately 3 W. While this is sufficient for small- to medium-scale demonstrations, higher sound pressure level applications would require external amplification or alternative devices.

2.3 Modular Daisy-Chain Loudspeaker PCBs

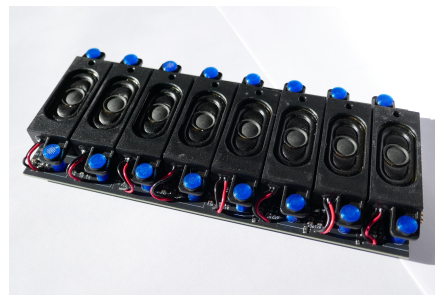
The overall PCB design is based on identical 8-loudspeaker modules (see Fig. 3a) that can be daisy-chained to form arrays of arbitrary size. Each module consists of a single PCB integrating eight MAX98357A digital amplifiers for eight loudspeakers, and forwarding the I²S signals from one side connector to the other. The loudspeakers used in this project (ASB05708CO-LW100-R models) are enclosed, with the PCB serving both as the electronic support and as a mechanical mounting structure through dedicated fixing holes (see Fig. 3b). This configuration enables precise control of the inter-speaker spacing, which is a critical parameter for spatial audio techniques. In addition, the fixed geometry of the side connectors defines a repeatable spacing between adjacent modules, ensuring uniform speaker placement across multiple boards. This approach provides a reliable and fast method for assembling evenly spaced loudspeaker arrays without the need for custom mechanical frames.

A functional block-level schematic of the PCB is shown in Fig. 4. To enable daisy-chaining, all PCB modules share an identical design. Each module is equipped with two 2×20, 2.54 mm pitch connectors placed on opposite sides. Two pins are reserved for distributing the shared clock signals, 6 for power supply, and 32 pins are available for data lines driven by the FPGA. Each data line feeds a single module of 8 loudspeakers and is connected internally to the eight amplifiers on that board. Module addressing and differentiation are handled entirely at the hardware level. Each PCB includes a set of 32 solder jumpers on its rear side that select which FPGA data line is connected to the module. Each module is therefore assigned a unique data line, while the two clock signals remain shared. Within a module, the eight amplifiers are configured to extract different time slots from the incoming TDM stream, allowing the FPGA to independently address all loudspeakers across the array. These 32 data lines can therefore address at most 32 modules, corresponding to 256 loudspeakers.

The 2×20 connector used on each loudspeaker module was chosen to match the connectors exposed by the FPGA system-on-module. The ALINX AC7020C provides three identical 2×20



(a) 3D rendering of the PCB.



(b) Module with PCB-mounted speakers installed.

Figure 3: Single custom-made PCB module embedding eight MAX98357A amplifiers and footprints for ASB05708CO-LW100-R loudspeakers.

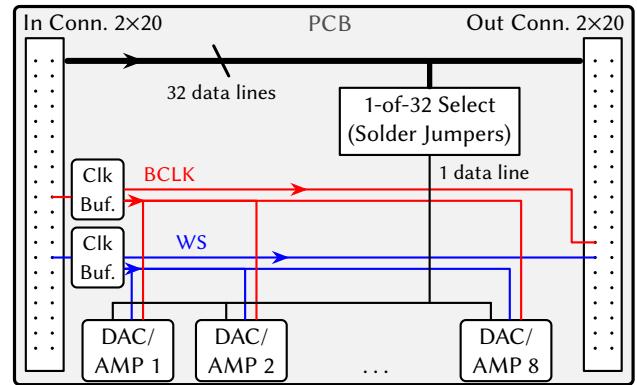


Figure 4: Block-level schematic of a single 8-loudspeakers PCB module. One data line is selected from a forwarded 32-line bus and distributed to eight DAC/amplifiers using TDM, while clock signals are buffered locally and forwarded to the next module.

connectors with consistent pin assignments, allowing any connector to be used interchangeably for this system. This enables the FPGA module to be mounted directly at the end of a loudspeaker bar using a single mechanical and electrical interface. In addition, the three connectors allow up to three independent loudspeaker arrays to be driven in parallel, each with its own clock and data signals. This configuration increases the theoretical number of loudspeakers controllable by a single FPGA to up to 3 × 256 channels.

2.4 Scalability Limits

To operate reliably over long chains of modules, signal integrity of the clock lines must be carefully managed. The clocks are shared by many DACs in parallel and the array can span several meters, so directly driving all devices from a single FPGA pin would exceed its drive capability due to the accumulated capacitive load. A clock buffering strategy is therefore required.

Rather than distributing separate clock lines to each module, which would increase wiring and GPIO usage, we adopt a cascaded topology where each module contains a local 1-to-2 clock buffer. One output of the clock buffer drives the eight local DACs while the other forwards a regenerated clock to the next module. This approach ensures that the FPGA only drives the first buffer and that the clock signal is restored at every stage without increasing the number of inter-module connections. A clock buffer is also used for the WS clock.

The clock buffer introduces a cumulative propagation delay along the chain, which may desynchronize the data line relative to the clock signal. The DAC requires a minimum setup time and hold time of 10 ns for the data value around the clock sampling edge (BCLK). Since all data lines share the same BCLK, it must be ensured that both the worst-case delayed signal and the zero-delay case satisfy the DAC timing requirements. The tolerable skew between clock and data signals is therefore:

$$t_{\text{margin}} \approx T_{\text{BCLK}} - 2 \times 10 \text{ ns.}$$

For a TDM8 configuration at 48 kHz with 16-bit samples, the bit clock is $\text{BCLK} = 6.144 \text{ MHz}$, corresponding to a period of $T_{\text{BCLK}} \approx 162.8 \text{ ns}$ and a timing margin of approximately 140 ns. Using LMK1C1102A buffers with a propagation delay of up to 2 ns per stage, up to $140/2 = 70$ cascaded stages are theoretically achievable before timing constraints are violated, corresponding to 560 loudspeakers. Since the chain is symmetric and clock and data lines travel the same distance on each board, PCB trace propagation delay cancels out and does not contribute to inter-module skew. This assumption was experimentally validated: a skew of approximately 13 ns was measured between the first and eighth modules, consistent with a per-stage buffer delay of $13/7 \approx 1.9 \text{ ns}$, in close agreement with the datasheet value.

Beyond this limit, the data line would require re-timing either by inserting hardware buffering on the data line to re-synchronize it with the clock, or by applying a controlled phase shift within the FPGA logic, avoiding any modification of the existing hardware. In practice, the limit of 32 modules imposed by GPIO availability, as discussed in Section 2.3, is expected to be reached before clock propagation or signal-integrity constraints become a limiting factor.

Even if the signals are properly time-aligned, long physical interconnects can degrade signal quality through attenuation, reflections, and crosstalk. Excessive chain length may therefore still require re-driving the data line to maintain valid logic levels. In practice, stable operation was verified with eight modules, corresponding to a total array length of 152 cm.

3 Software

A common limitation of FPGA-based systems is that they are notoriously difficult to program, due to their inherently low-level and hardware-centric architecture. However, recent work has significantly lowered this barrier by providing higher-level abstractions tailored to real-time audio processing [9].

3.1 Hardware Abstraction and Control with Syfala

In this project, we rely on the Syfala toolchain⁴ [11], which enables FPGA programming using high-level audio programming languages such as Faust⁵ as well as C/C++. Syfala generates the FPGA logic, handles data transfers, and deploys a custom embedded Linux system running on the FPGA's integrated CPU [3].

This environment provides standard control and communication interfaces, including OSC, MIDI over USB, HTTP, and network-based audio/control mechanisms, allowing the loudspeaker array to be controlled using familiar software tools.

At the hardware interface level, Syfala fully abstracts communication with the loudspeaker array. The I²S/TDM interface is provided by the platform and automatically scaled according to the number of output channels. As a result, the user only needs to focus on implementing the audio DSP algorithm itself. When using Faust, the complete signal-processing chain (including audio routing, parameter control, and integration with the embedded Linux environment) is generated automatically, enabling rapid prototyping and deployment. When using C++, which allows the user to better optimize its FPGA design, the design is split between two components. The sample-level DSP computation is implemented in a dedicated function synthesized onto the FPGA fabric, while higher-level control logic runs on the embedded CPU under Linux. Communication between these two domains is handled transparently through an optimized shared-memory interface. A dedicated API is provided to expose DSP parameters to the control layer, which can then be connected to external interfaces such as OSC or MIDI without additional low-level development. Code examples are omitted here for clarity, but detailed documentation and examples are available in the Syfala project resources.⁴

The toolchain also facilitates rapid integration and development of graphical user interfaces for controlling the implemented algorithms. For example, Fig. 5 shows a simple interface developed in Processing to control a Wave Field Synthesis algorithm running on the FPGA. The interface allows users to intuitively position sound sources within a virtual room using drag-and-drop interaction. Source coordinates are transmitted as (x,y) values via OSC over Ethernet to the embedded CPU, which then updates the corresponding delay and gain parameters for each loudspeaker in real time. This interface was later extended to enable direct control from a digital audio workstation (DAW) using OSC, allowing seamless integration into existing audio production workflows.

Syfala proposes an ad-hoc TCP/UDP real-time streaming protocol for sending multichannel audio to the FPGA (see "Ethernet Link" on Fig. 5). It relies on the JACK Audio Connection Kit, as well as a dedicated FPGA IP and application running on the embedded Linux [4].

Finally, while the platform presented here leverages an FPGA and the Syfala framework to achieve deterministic timing and high channel counts, the loudspeaker array hardware itself is not FPGA-specific. The use of a standard I²S/TDM interface means that the array could, in principle, be driven by a microcontroller or CPU-based system, provided that sufficient throughput and timing accuracy are available to implement the required protocol.

⁴<https://github.com/inria-emeraude/syfala>

⁵<https://faust.grame.fr/>

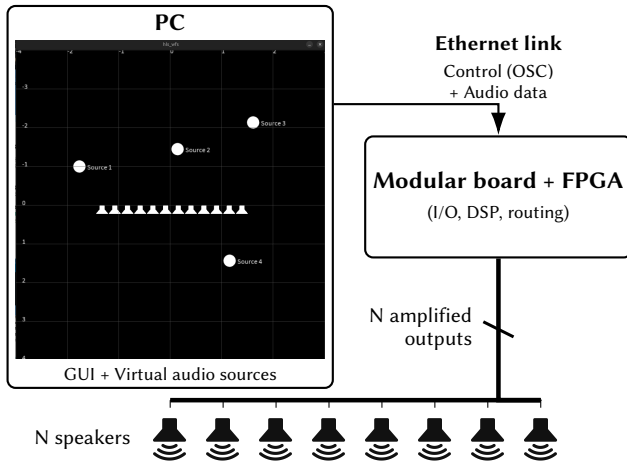


Figure 5: Control and signal flow for the spatial audio system. Virtual audio sources are positioned in a GUI (generated with Processing) running on a PC and transmitted via Ethernet to the platform, which performs DSP, routing, and amplification before driving the loudspeaker array.

3.2 Case Study: WFS and Beam Steering

To evaluate the proposed loudspeaker bar with spatial audio algorithms, a Wave Field Synthesis implementation was developed. The algorithm follows the approach described in our previous work [7], where WFS is implemented by simulating the relative delays of arrival between a virtual source and each loudspeaker in a linear array.

Compared to our earlier implementation, the algorithm has been adapted to exploit the increased loudspeaker density afforded by the present hardware design. Increasing the density of secondary sources in a WFS array improves the spatial sampling of the reproduced sound field and directly extends the alias-free frequency range. In the proposed system, reducing the loudspeaker spacing from 78.3 mm to 24 mm increases the spatial Nyquist frequency from approximately 2.2 kHz to about 7.1 kHz. As a result, high-frequency wavefront curvature and source localization cues are more accurately reproduced, and spatial stability is improved.

We also implemented a beamsteering algorithm to further evaluate the capabilities of the proposed array. The approach relies on applying progressive phase shifts across the loudspeaker elements to steer the main radiation lobe toward a desired direction. This technique provides a simple yet effective means of controlling sound directivity and serves as a complementary test case for dense array operation. Despite the limited frequency response of the compact loudspeakers used in this prototype, the beamsteering experiments produced convincing directional effects within the usable bandwidth. The high spatial density of the array contributed to smooth beam formation and reduced sidelobe artifacts, demonstrating that the proposed hardware platform is well suited for directional sound control and array-based rendering techniques beyond WFS.

4 Practical Considerations and Deployment

This section discusses practical aspects of the proposed system, focusing on portability, geometric flexibility, and hardware adaptability. These considerations highlight the suitability of the platform for real-world spatial audio experiments, installations, and artistic deployments.

4.1 Portability and Geometry

The modular construction of the loudspeaker bar provides two key mechanical advantages. First, it enables straightforward transport by allowing the array to be disassembled into small, manageable modules. Second, it facilitates geometric flexibility, as each module of eight loudspeakers acts as an elementary building block that can be rearranged to form different array configurations.

Each module measures 190×90 mm with a height of 18 mm (excluding connectors), resulting in a compact form factor. This allows the complete system (including the loudspeaker modules, the FPGA unit, and the power supply) to be transported in a relatively small volume. As illustrated in Fig. 6, the entire 8 modules setup fits within a 27×37 cm carrying case. This compactness makes the system suitable for mobile use, including artistic performances and on-site installations.

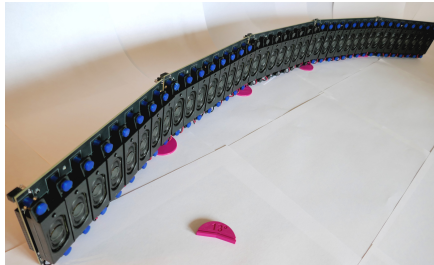


Figure 6: Carrying case (27×37 cm) capable of accommodating up to eight modules (64 loudspeaker), together with the FPGA unit and the power supply.

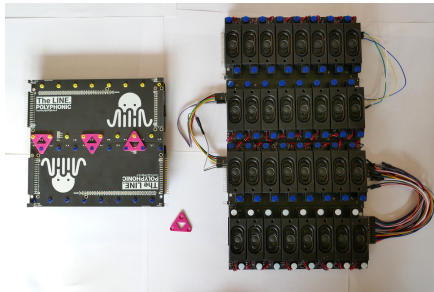
The modular design enables rapid reconfiguration of the loudspeaker array geometry, which is a key parameter in spatial audio rendering. The geometry of a loudspeaker array directly influences the spatial sampling of the sound field, the directivity of reproduced wavefronts, and the effective listening area. While linear arrays are commonly used for WFS due to their simplicity and well-established theoretical framework, alternative geometries can better address specific perceptual, spatial, or practical requirements depending on the intended application and listening context. Figure 7 illustrates two example configurations. In the first, a concave array is formed using small 3D-printed spacers (pink in the figure). In the second, the modules are arranged into a two-dimensional matrix using 3D-printed supports. Depending on the chosen geometry, extended inter-module cables or connector adapters may be required, but the electrical and signal architecture remains unchanged.

Concave configurations can be advantageous in small or medium-sized spaces by focusing wavefronts toward a defined listening

region, improving spatial stability and localization accuracy near the center of curvature. Two-dimensional matrix layouts, on the other hand, enable spatial control in both horizontal and vertical dimensions, making them suitable for immersive rendering, near-field experiments, and beamsteering applications. These examples illustrate how geometric flexibility extends the practical scope of the system beyond a single rendering paradigm.



(a) Concave configuration



(b) Matrix configuration

Figure 7: Examples of array geometries enabled by the modular design: concave and two-dimensional matrix configurations.

4.2 Loudspeaker Adaptability

The MAX98357A used in this project is a compact 16-bit, 3 W class-D audio amplifier, chosen for its suitability for dense integration and low power operation, at the expense of ultimate audio fidelity and dynamic range. Similarly, the loudspeakers were selected primarily for their small form factor, enabling a minimal enclosure and an inter-speaker spacing of only 24 mm, which is a central design objective of the system. However, this choice results in limited acoustic performance. Furthermore, the loudspeaker manufacturer does not provide detailed acoustic specifications, preventing a rigorous performance evaluation. Consequently, no acoustic measurements are presented in this paper.

It should be noted that substantially improved sound quality could be achieved by replacing the loudspeakers while preserving the same PCB design. As an illustration, the first prototype of this board (presented in [7]), which featured different mechanical properties but identical electronics, was used at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University. In this context, the board served as a custom interface for an existing loudspeaker array: the EMPAC high-resolution modular loudspeaker array for wave field synthesis, based on Tang Band W2-2136S loudspeakers. Normally driven by Dante-controlled amplifiers, this array was instead directly connected to the proposed board, providing a compact solution that enabled the deployment of custom signal-processing algorithms,

controlled via OSC and supplied with data over Ethernet with Syfala.

To improve reusability and adaptability, a script has been developed to generate parametric PCB layouts and is available in the project repository. By specifying the loudspeaker dimensions (width and height) and the desired spacing, the script automatically generates the corresponding PCB design, including the placement of custom loudspeaker footprints. This is made possible by the circuit architecture, which is based on repeated, identical functional blocks (one per loudspeaker) connected through parallel, extendable signal and power tracks. This approach remains constrained by PCB manufacturing limits: for the manufacturer used in this project, the maximum multi-layer board size of 600×700 mm would allow a minimum loudspeaker spacing of approximately 75 mm in the most restrictive configuration.

5 Conclusion and Future Work

This paper presented a modular, cost-effective, and high-density loudspeaker array platform for embedded spatial audio applications. By combining compact, daisy-chainable loudspeaker PCBs with an FPGA-based processing system, the proposed platform enables large channel counts and flexible array geometries within a portable and reproducible form factor. The modular design supports rapid reconfiguration and adaptation to different use cases, making the platform well suited for spatial audio research, experimentation, and artistic deployment.

The current implementation is intentionally optimized for high spatial density and ease of integration, which imposes limitations on loudspeaker size and amplifier power. However, the use of a parametric PCB generation script makes it possible to accommodate larger loudspeakers.

Applications requiring higher acoustic output would necessitate replacing the integrated amplifiers with external amplification stages. Beyond PCB re-routing, this would introduce additional design constraints, including thermal management, power delivery, and mechanical integration, which constitute natural directions for future system extensions.

6 Acknowledgments

This work was partially supported by the Inria Start-up Studio program and partially funded by the French ANR through the FAST project (ANR-20-CE38-0001). The authors would like to thank the reviewers for their valuable comments.

7 Ethical Standards

This project has been entirely funded by French public money. It doesn't involve human participants and it has been entirely open-sourced.

References

- [1] A. J. Berkhout. 1988. A Holographic Approach to Acoustic Control. *Journal of the Audio Engineering Society* 36, 12 (Dec. 1988), 977–995. AES e-library article #5117.
- [2] Thibaut Carpentier. 2022. Spatialisation sonore - Perception, captation et diffusion de scènes sonores. *Techniques de l'Ingénieur Le traitement du signal et ses applications*, BR1150 (Nov. 2022).
- [3] Pierre Cochard, Maxime Popoff, Antoine Fraboulet, Tanguy Risset, Stéphane Letz, and Romain Michon. 2023. A Programmable Linux-Based FPGA Platform for Audio DSP. In *Proceedings of the 20th Sound and Music Computing*. Royal College of Music and KTH Royal Institute of Technology, Bresin, R., & Falkenberg, K., Stockholm, Sweden, 110–116.
- [4] Pierre Cochard, Jurek Weber, Romain Michon, Tanguy Risset, and Stéphane Letz. 2024. Ethernet Real-Time Audio Transmission to FPGA. In *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)*. <https://doi.org/10.1109/IS262782.2024.10704104>

- [5] Michael A Gerzon. 1985. Ambisonics in Multichannel Broadcasting and Video. *Journal of the Audio Engineering Society* 33, 11 (1985), 859–871.
- [6] Shoichi Koyama, Enzo De Sena, Prasanga Samarasinghe, Mark R. P. Thomas, and Fabio Antonacci. 2025. Past, Present, and Future of Spatial Audio and Room Acoustics. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Hyderabad, India, 1–5.
- [7] Romain Michon, Joseph Bizien, Maxime Popoff, and Tanguy Risset. 2023. Making Frugal Spatial Audio Systems Using Field-Programmable Gate Arrays. In *Proceedings of the 2023 New Interfaces for Musical Expression Conference (NIME-23)*. nime.org, Mexico City, Mexico.
- [8] Ferdinando Olivieri, Filippo Maria Fazi, Philip A. Nelson, Mincheol Shin, Simone Fontana, and Lang Yue. 2016. Theoretical and experimental comparative analysis of beamforming methods for loudspeaker arrays under given performance constraints. *Journal of Sound and Vibration* 373 (2016), 302–324.
- [9] Maxime Popoff. 2024. *Compilation of real-time audio DSP on FPGA*. These, no 2024ISAL0128. INSA de Lyon.
- [10] Maxime Popoff, Romain Michon, and Tanguy Risset. 2024. Enabling Affordable and Scalable Audio Spatialization With Multichannel Audio Expansion Boards for FPGA. In *Proceedings of the 2024 Sound and Music Computing Conference (SMC-24)*. IEEE, Porto, Portugal.
- [11] Maxime Popoff, Romain Michon, Tanguy Risset, Pierre Cochard, Stéphane Letz, Yann Orlarey, and Florent de Dinechin. 2023. *Audio DSP to FPGA Compilation: The Syfala Toolchain Approach*. Technical Report RR-9507. Univ Lyon, INSA Lyon, Inria, CITI, Grame, Emeraude.
- [12] Ville Pulkki. 1997. Virtual Sound Source Positioning Using Vector Base Amplitude Panning. *Journal of Audio Engineering Society* 45, 6 (June 1997), 456–466.
- [13] Benjamin Quiédeville, Romain Michon, Tanguy Risset, and Stéphane Letz. 2025. The Space Bar: An Embedded WFS Sound System. In *Proceedings of the 32th Journées d'Informatique Musicale*. Lyon, France.
- [14] Zhiyuan Zhu, Yu Zhang, Wenxiang Guo, Changhao Pan, and Zhou Zhao. 2025. ASAudio: A Survey of Advanced Spatial Audio Research. arXiv:2508.10924 [eess.AS]