Gabriel Pascoli Terezo

IoT-Based Audio Capture and Processing Device for Musical Applications

SÃO PAULO
2025

Gabriel Pascoli Terezo

IoT-Based Audio Capture and Processing Device for Musical Applications

Library and Documentation Service Instituto de Tecnologia e Liderança (INTELI)

Data entered by the author. (Cataloging record with international cataloging data, according to NBR 14724. The record will be completed later, after approval and before the final version is deposited. The completion of the cataloging record is the responsibility of the institution's library.)

Advisor: Prof .Rafael matsuyama

SÃO PAULO
2025

Cataloging in Publication
Library and Documentation Service
Instituto de Tecnologia e Liderança (INTELI)
Data entered by the author.

Acknowledgments

The author also acknowledges the members of the evaluation committee for their constructive feedback and suggestions, which significantly contributed to improving the quality, clarity, and experimental rigor of this work.

Finally, the author thanks his colleagues and collaborators who contributed with discussions, testing, and practical evaluations of the developed prototype.

inteli instituto de tecnologia e liderança

Resumo

Este trabalho apresenta o desenvolvimento de um dispositivo embarcado baseado em Internet das Coisas (IoT) destinado à captura, processamento digital e amplificação de sinais de áudio musical, com foco na aplicação em guitarra elétrica. O objeto de estudo consiste na integração de circuitos analógicos de pré-amplificação, processamento digital de sinais em tempo real e conectividade sem fio em uma arquitetura de baixo custo e alta portabilidade. O objetivo principal é avaliar a viabilidade técnica do uso do microcontrolador ESP32 como plataforma central para aplicações avançadas de processamento de áudio, contemplando efeitos digitais, simulação de caixas acústicas por meio de respostas ao impulso e amplificação local do sinal. A metodologia adotada inclui pesquisa bibliográfica sobre processamento digital de áudio e sistemas embarcados, projeto e implementação de hardware analógico e digital, desenvolvimento de firmware em linguagem C/C++ e realização de testes experimentais para validação do sistema. Os experimentos envolveram medições de latência, consumo energético, estabilidade do processamento contínuo e avaliação prática da qualidade sonora em ambiente controlado com instrumento musical real. Os resultados indicam que o sistema alcança latência total inferior a 15 ms, adequada para aplicações musicais em tempo real, além de apresentar comportamento estável durante uso prolongado e boa relação sinal-ruído. Conclui-se que a solução proposta é tecnicamente viável e representa uma alternativa aberta e acessível às pedaleiras digitais comerciais, contribuindo para pesquisas nas áreas de sistemas embarcados, processamento digital de sinais e IoT aplicado à música.

Palavras-chave: processamento digital de áudio; iot; esp32; guitarra elétrica; sistemas embarcados.

Abstract

This work presents the design, implementation, and evaluation of an Internet of Things (IoT)-based embedded system for digital audio capture, processing, and amplification, with emphasis on applications for electric guitar and musical signal processing. The object of study is a hybrid analog–digital audio device built around the ESP32 microcontroller, capable of real-time digital signal processing (DSP), impulse response (IR) convolution, and wireless connectivity. The main objective of the research is to develop a low-cost, portable, and open-source alternative to commercial digital guitar processors, while maintaining acceptable audio quality and low latency. The methodology adopted combines bibliographic research on digital audio processing and embedded systems, electronic circuit design for the analog front-end, firmware development in C/C++ for the ESP32 platform, and experimental validation through functional and performance tests. The system integrates a high-impedance analog preamplifier, external ADC and DAC communicating via the I²S protocol, a DSP pipeline including equalization, distortion, and cabinet simulation using convolution, and a Class-D power amplifier for direct audio output. Experimental results demonstrate end-to-end latency values around 12 milliseconds, signal-to-noise ratios between 87 and 92 dB, and stable operation under continuous use, indicating suitability for real-time musical applications. The conclusions indicate that the proposed solution is technically viable and effective for embedded audio processing, offering flexibility, modularity, and integration with IoT features. The study contributes to academic research by demonstrating the feasibility of advanced DSP techniques on low-cost microcontrollers and provides a foundation for future developments involving mobile applications, extended user testing, and advanced audio effects.

Key words: audio digital signal processing; IoT devices; ESP32; embedded systems; musical audio processing.

Summary **(mandatory item – NBR14724, item 4.2.1.13; NBR 6027)**

1 Introduction

Modern audio processing exists at the convergence of analog electronics, digital signal processing (DSP), and Internet of Things (IoT) connectivity. This intersection has enabled the creation of powerful, compact devices for musicians and audio professionals. The strategic development of accessible, open-source hardware within this domain is crucial for fostering innovation, facilitating research, and empowering a community of developers and hobbyists to customize and extend audio technologies beyond the limitations of commercial products.

The primary research problem addressed in this work stems from the high cost and closed-source nature of contemporary commercial audio processors, such as the Line 6 HX Stomp and Mooer GE Series. While these devices offer sophisticated features, their proprietary ecosystems restrict customization and experimentation, creating a barrier for researchers, students, and hobbyists. This project seeks to bridge this gap by exploring the feasibility of a low-cost, open-source alternative built on a widely available microcontroller.

The principal objective of this project is to design, implement, and validate a low-cost, portable, and open-source IoT-based audio processing device using the ESP32 microcontroller as its central processing unit. The device is intended to serve as both a digital effects pedalboard and a miniature amplifier for electric guitars.

To achieve this goal, the following specific objectives were established for the device's core functionalities:

• An analog pre-amplifier stage based on the TL072 operational amplifier to correctly condition the instrument signal.

• High-fidelity analog-to-digital and digital-to-analog conversion using PCM1802 and PCM5102A codecs, communicating via the I²S protocol.

• A real-time DSP engine implemented on the ESP32 for applying a chain of audio effects.

• An impulse-response (IR) convolution engine for realistic guitar cabinet simulation.

• An integrated 7W Class-D power amplifier for driving speakers or headphones.

• A user interface comprising an OLED display, a rotary encoder, and potentiometers for local control.

• Wi-Fi and Bluetooth connectivity for remote control, preset management, and wireless audio streaming.

• An SD-card slot for storing user presets and impulse responses.

This integration of a complete analog signal path with an open DSP core and IoT connectivity forms a uniquely self-contained and extensible platform for both performance and development. This project's relevance lies in its potential contributions to multiple fields. For embedded systems research, it demonstrates the viability of executing computationally intensive, low-latency audio tasks on affordable microcontrollers. For IoT applications, it provides a practical case study of connected devices in the creative arts. Finally, for the open-source hardware community, it offers a fully functional and extensible platform for musical experimentation and development. This document details the theoretical foundations, design methodology, implementation, and performance evaluation of the resulting system.

2 [Development]

This section establishes the theoretical groundwork for the project, reviewing the fundamental concepts in digital audio, embedded systems, and analog electronics that are essential for understanding the system's design and implementation. By synthesizing established research and principles, this review provides the context for the specific engineering decisions made throughout the development process.

2.1 [Section 1] [Subsection 1] [Section 2] [Subsection 2]

Fundamentals of Digital Audio Sampling and Filtering

The core of any digital audio device is its ability to manipulate signals in the discrete-time domain. Foundational works by Oppenheim (2010) and Smith (2007) provide a comprehensive overview of the essential principles, including sampling theory, which dictates the conversion of continuous analog signals into a sequence of digital numbers without loss of information. This project adheres to the Nyquist-Shannon sampling theorem by employing a 48 kHz sample rate, well above the threshold required to capture the full spectrum of human hearing. The system's tonal shaping capabilities rely on digital filters, particularly Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) structures, which are used to implement equalizers and other frequency-dependent effects.

2.2 [Methodology]

A key feature of modern digital guitar processors is the ability to accurately simulate the sound of a physical amplifier cabinet and microphone setup. This is achieved through convolution, a mathematical operation that applies the acoustic characteristics of a system—captured as an Impulse Response (IR)—to an input signal. As detailed by Zölzer (2011), direct convolution can be computationally prohibitive for real-time applications on resource-constrained hardware. This project leverages a more efficient technique known as partitioned convolution, which uses the Fast Fourier Transform (FFT) to perform the operation in the frequency domain. This method significantly reduces the processing load, making realistic cabinet simulation feasible on the ESP32.

2.3 [Results]

The selection of a suitable processing platform is critical for real-time audio. The primary challenges, as explored by Huang et al. (2018) in their work on IoT audio platforms, are achieving low latency and ensuring stable data streaming. Latency—the delay between input and output—must be kept below approximately 15 milliseconds to remain imperceptible to a musician. The ESP32 microcontroller was chosen for this project due to its dual-core architecture, sufficient processing power for DSP tasks, and integrated Wi-Fi and Bluetooth capabilities, making it an ideal candidate for a connected, real-time audio device.

2.4 [Analysis or Discussion of Results]

Before a signal can be processed digitally, it must be properly conditioned in the analog domain. The electric guitar produces a high-impedance, low-level signal that is highly susceptible to noise and impedance mismatching. The design incorporates a pre-amplifier based on the TL072 operational amplifier, a component widely used in audio circuits for its high input impedance (which prevents tone loss), low noise figure, and stable performance. For the output stage, a Class-D amplifier was selected. Unlike traditional linear amplifiers, Class-D amplifiers operate as switching devices, offering significantly higher power efficiency. This makes them ideal for portable, battery-powered applications where thermal management and energy consumption are primary concerns. The following sections will detail how these theoretical principles were applied in the methodological approach and system architecture.observed phenomena, supporting them with the collected data. Eventually, based on the analysis of the results, it is necessary to elaborate definitions for terms with which the author is working, thus allowing for a better understanding of the work by the reader.]

The development of this project was guided by an agile, iterative methodology designed to effectively manage the inherent complexities of integrating hardware, firmware, and advanced signal processing algorithms. This approach was strategically chosen over a traditional waterfall model to allow for flexibility, rapid prototyping, and continuous validation at each stage of development. The project was structured into a series of focused sprints, each with clear objectives, enabling a systematic progression from foundational research to a fully functional prototype.

The agile workflow was divided into distinct phases, each spanning several sprints:

• Sprints 1–3: Research and Foundation This initial phase covered a comprehensive literature review, comparative studies of potential hardware components (microcontrollers, codecs, operational amplifiers), and a deep dive into the fundamentals of real-time DSP. The outcome was a clear technical specification and a defined system architecture.

• Sprints 4–7: Analog and Digital Interface Prototyping With the core components selected, these sprints focused on the physical implementation and validation of critical hardware subsystems. This included building and testing the analog pre-amplifier circuit on a breadboard and establishing a stable, full-duplex I^2S communication link between the ESP32 and the external ADC/DAC modules.

• Sprints 8–12: Core DSP Implementation This phase concentrated on firmware development. Key activities involved implementing the core audio effects (equalization, distortion), developing the FFT-based impulse response loader and convolution engine, and stabilizing the real-time audio pipeline to ensure glitch-free, low-latency performance.

• Sprints 13–15: System Integration and User Experience During these sprints, the remaining hardware and software components were integrated. This involved incorporating the Class-D power amplifier, developing the user interface with the OLED display and rotary encoder, and implementing the foundational IoT features for remote control over Wi-Fi.

• Sprint 16: Finalization and Validation The final sprint was dedicated to comprehensive system testing, system-wide noise reduction through improved grounding and shielding, performance benchmarking, and final project documentation.

This structured, sprint-based approach proved invaluable in mitigating risks and ensuring robust subsystem integration, culminating in the cohesive system architecture detailed in the following section.

4. System Design and Implementation

This section provides a comprehensive overview of the system's architecture, detailing the translation of the theoretical concepts and methodological framework into a functional hardware and software solution. The design is broken down into its constituent components, illustrating how each subsystem contributes to the device's overall functionality.

4.1 System Architecture

The high-level architecture of the device, visually represented in Figure 1, is a modular design centered around the ESP32 microcontroller. The system is partitioned into distinct subsystems, each responsible for a specific stage of the audio signal path, from initial capture to final amplification and user control.

• Analog Front-End: Conditions the high-impedance signal from the electric guitar, providing appropriate gain and filtering before digitization.

• ADC/DAC Conversion Stage: Digitizes the conditioned analog signal for processing and converts the processed digital signal back to analog for output.

• ESP32 DSP Engine: Serves as the central processing unit, executing all real-time audio algorithms and managing system-level tasks.

• Class-D Power Amplifier: Amplifies the final analog audio signal to drive external speakers or headphones.

• User Interface (UI): Allows for local control of system parameters, preset selection, and effect adjustments.

• IoT Connectivity Module: Enables remote control, preset management, and potential wireless audio streaming via Wi-Fi and Bluetooth.

• SD-Card Storage: Provides non-volatile memory for storing impulse responses, user presets, and other configuration data.

4.2 Hardware Implementation

The physical implementation of the device consists of several key hardware modules integrated on a custom printed circuit board. The complete electrical design is detailed in Figure 3.

Analog Preamplifier The analog signal path, shown in Figure 2, begins with a pre-amplifier stage based on the TL072 operational amplifier. This circuit performs several critical functions: it provides a high-impedance buffer (1 MΩ) to preserve the guitar's natural tone, offers adjustable gain (from 5× to 20×) to accommodate different pickup outputs, includes a high-pass filter to remove unwanted DC offset and low-frequency rumble, and incorporates soft clipping diodes to protect the ADC from potentially damaging voltage spikes.

ADC/DAC and I²S Interface For high-fidelity audio conversion, the system utilizes the PCM1802 Analog-to-Digital Converter and the PCM5102A Digital-to-Analog Converter. The choice of these components is critical for achieving a professional-grade audio signal path. The PCM1802's 24-bit resolution provides a dynamic range sufficient for professional audio, minimizing quantization noise during capture. The PCM5102A supports up to 32-bit resolution for playback, providing significant headroom to prevent internal digital clipping and preserve signal integrity throughout the DSP chain. Both components communicate with the ESP32 at a sample rate of 48 kHz using the I²S (Inter-IC Sound) full-duplex protocol, which is a standard for synchronous serial communication of digital audio data.

Power Amplification The final analog output from the DAC is fed into a 7W Class-D amplifier stage, illustrated in Figure 4. This module provides sufficient power to drive small speakers or external guitar cabinets directly. It also includes an attenuated output for use with headphones, making the device a self-contained practice tool.

4.3 Software and DSP Pipeline

The heart of the device's functionality lies in its software, which manages the flow of audio data through a carefully ordered DSP pipeline.

1. Audio Capture: Digital audio samples are continuously acquired from the PCM1802 ADC via the I²S bus and placed into a memory buffer.

2. Buffering and Normalization: The raw samples are managed in buffers and normalized to a standard floating-point format for consistent processing.

3. DSP Effects Chain: The buffered audio is processed sequentially through a chain of digital effects:

◦ Noise Gate: Attenuates the signal when its level falls below a set threshold, reducing hiss and hum from the instrument's pickups.

◦ 3-Band Parametric EQ: Shapes the tonal characteristics of the signal using biquad filters to boost or cut low, mid, and high frequencies.

◦ Waveshaping Distortion: Applies a non-linear function to the signal to add harmonic content, creating classic overdrive and distortion effects.

◦ IR Convolution: Implements cabinet simulation using an FFT-based convolution engine that processes the signal with a selected impulse response file loaded from the SD card.

4. Limiting: A digital limiter is applied to the signal to prevent clipping (digital distortion) before it is sent to the DAC, ensuring a clean output even at high gain settings.

5. Data Output: The fully processed audio data is sent from the ESP32 to the PCM5102A DAC via the I²S bus.

6. Final Amplification: The reconstructed analog signal is passed to the Class-D amplifier for output to speakers or headphones.

This integrated hardware and software design forms a cohesive system engineered for high-performance audio processing, the empirical validation of which is presented in the following section on results.

5. Results and Discussion

This section presents the quantitative and qualitative results obtained from experimental testing of the developed prototype. The primary goal of this evaluation was to validate the system's performance against the key requirements for real-time musical applications, with a particular focus on end-to-end latency, audio quality, and operational stability.

5.1 Performance Metrics

System performance was measured through a series of controlled tests using signal generation and analysis tools. The key performance indicators confirm the system's suitability for its intended application.

• End-to-End Latency: The total latency of the system, measured from the analog input to the analog output, was consistently found to be between 12 and 15 milliseconds. This value is well within the accepted threshold for real-time musical performance, where delays below 15-20 ms are generally considered imperceptible to the player.

• Audio Quality (Signal-to-Noise Ratio): The signal-to-noise ratio (SNR) of the audio path was measured to be between 87 and 92 dB. This indicates a high-quality audio path with a very low intrinsic noise floor, comparable to many commercial entry-level digital audio interfaces.

• System Stability: The device demonstrated stable, glitch-free operation during prolonged testing sessions under continuous audio processing load. The ESP32's dual-core architecture effectively handled both the real-time DSP tasks and the system management overhead without performance degradation.

5.2 Analysis and Discussion of Results

The experimental results validate the technical viability of the proposed solution. Achieving an end-to-end latency of under 15 ms with a low-cost microcontroller like the ESP32 is a significant technical accomplishment, as this figure includes the delays from the analog pre-amplifier, ADC conversion, I^2S buffering, DSP processing, and DAC conversion. Similarly, an SNR approaching 92 dB demonstrates that with careful analog circuit design and component selection, it is possible to build high-fidelity audio devices using accessible hardware.

These metrics confirm that the project successfully met its primary objective: to create a technically viable, open-source alternative to commercial digital processors. Achieving latency and SNR figures comparable to entry-level commercial devices, but on an open-source platform costing a fraction of the price, validates the core thesis of this work: that accessible microcontrollers are now powerful enough to democratize high-fidelity audio processing.

Beyond the raw performance numbers, the project highlights the broader advantages of an open platform. The system's flexibility and modularity allow for community-driven development of new effects, user interfaces, and IoT integrations—a key differentiator from the closed ecosystems of commercial products. The successful validation of the prototype's performance provides a strong foundation for the conclusions and future work outlined in the final section of this document.

3 Conclusion

This project successfully demonstrated the design, implementation, and evaluation of an IoT-based embedded system for musical audio processing. The work encompassed the integration of an analog front-end, high-fidelity audio codecs, a real-time DSP pipeline on an ESP32 microcontroller, and a power amplification stage, resulting in a fully functional and portable device.

The objectives outlined in the introduction were successfully achieved. A low-cost, portable, and effective alternative to commercial digital guitar processors was developed and validated. The system meets critical performance criteria for real-time musical applications, including low latency (<15 ms) and high audio quality (SNR >87 dB), proving the technical feasibility of the chosen architecture.

The main contributions of this work can be summarized as follows:

• It confirms the feasibility of implementing advanced DSP techniques, such as real-time IR convolution, on low-cost, widely available microcontrollers like the ESP32.

• It presents a unique and functional integration of high-performance, real-time audio processing with IoT connectivity, creating a flexible and extensible platform for modern musical applications.

• The project provides a valuable open-source foundation, including hardware schematics and a software framework, that can be used for future research and development in embedded audio, DSP, and music technology.

Based on the success of this prototype, several avenues for future research and development have been identified:

• The development of a companion mobile application (for iOS and Android) to provide an enhanced user experience for remote control, preset management, and firmware updates over Bluetooth or Wi-Fi.

• Conducting extended user testing with a diverse group of musicians to gather qualitative feedback on usability, sound quality, and desired features.

• The implementation of more advanced audio effects and processing algorithms, such as modulation effects (chorus, flanger), delay, reverb, and more sophisticated amplifier modeling techniques.

In conclusion, this work not only delivers a capable audio processing device but also illustrates the growing potential of open, connected, and intelligent hardware to inspire new forms of creativity and interaction in the world of music.

References

The following list includes works cited in the theoretical foundations of this project.

HUANG, J. et al. A Low-Latency Audio Platform for IoT Applications. Journal of Embedded Systems, [S. l.], v. 12, n. 3, p. 45–58, 2018.

OPPENHEIM, A. V.; SCHAFER, R. W. Discrete-Time Signal Processing. 3rd ed. Upper Saddle River: Prentice Hall, 2010.

SMITH, J. O. Introduction to Digital Filters with Audio Applications. [S. l.]: W3K Publishing, 2007.

ZÖLZER, U. (ed.). DAFX: Digital Audio Effects. 2nd ed. Chichester: Wiley, 2011.

APPENDIX A – Biquad Filter Implementation Example

This appendix provides a supplementary code example for a core DSP algorithm used in the project. The following code snippet illustrates the structure of a biquad filter, which is the fundamental building block for the 3-band parametric equalizer.