

UNIVERSIDADE DO ESTADO DE MINAS GERAIS

VISUALIZADOR DE FREQUÊNCIAS DE ÁUDIO EM TEMPO REAL COM ESP32

Divinópolis, MG

2025



Diego Antônio de Melo Morais

Gustavo Laureano de Almeida

João Pedro Ferreira Lemos

Paulo Oliveira Santos

Marcos Alberto Saldanha

Karolliny Danielle Santos

VISUALIZADOR DE FREQUÊNCIAS DE ÁUDIO EM TEMPO REAL COM ESP32

Relatório apresentado à 9ª FEMIC – Feira Mineira de Iniciação Científica.

Orientação do Prof. Karolliny Danielle Santos e
coorientação de Marcos Alberto Saldanha

Divinópolis, MG

2025



RESUMO

Visando demonstrar a aplicação prática de conceitos matemáticos frequentemente vistos como abstratos, este projeto teve como objetivo desenvolver um protótipo funcional para tornar tangível a utilização da Transformada de Fourier. Para isso, foi implementado um sistema embarcado para visualização do espectro de áudio em tempo real, utilizando um microcontrolador ESP32, um microfone e uma matriz de LEDs. O funcionamento do sistema embarcado consiste em captar o som ambiente, processar o sinal através do algoritmo da Transformada Rápida de Fourier (FFT) a partir da biblioteca ArduinoFFT, convertendo-o do domínio do tempo para o da frequência. Como resultado, o sistema exibe de forma gráfica e em tempo real o conteúdo espectral do áudio, com cada coluna da matriz de LEDs representando a intensidade de uma faixa de frequência específica, fornecendo respostas visuais consistentes para estímulos sonoros variados. Conclui-se que os objetivos foram plenamente alcançados, pois o protótipo não apenas validou a capacidade do hardware em realizar o processamento em tempo real, mas também funcionou como uma ferramenta didática eficaz, traduzindo dados matemáticos complexos em uma experiência visual intuitiva e reafirmando o papel fundamental do Cálculo em tecnologias cotidianas.

Palavras-chave: Cálculo diferencial Integral, Transformada de Fourier, Sistemas embarcados, Análise de sinais.



SUMÁRIO

1 INTRODUÇÃO	5
2 JUSTIFICATIVA	6
3 OBJETIVO GERAL	7
4 METODOLOGIA	7
5 RESULTADOS OBTIDOS	9
6 CONCLUSÕES OU CONSIDERAÇÕES FINAIS.....	13
REFERÊNCIAS.....	15
APÊNDICE A	16

1 INTRODUÇÃO

O processamento de sinais é um campo fundamental da engenharia, dedicado à análise, interpretação e manipulação de informações de fontes como áudio, imagens e dados de sensores. Suas aplicações são vastas e presentes no cotidiano, desde a compressão de dados em formatos como MP3 até a filtragem de ruídos em sistemas de comunicação, como Wifi e 5G. Uma das tarefas mais importantes nessa área é a análise espectral, que consiste em entender a composição de um sinal a partir das frequências que o formam.

A ferramenta matemática que fundamenta essa análise é a Transformada de Fourier, que permite a decomposição de um sinal em suas componentes de frequência, revelando sua estrutura harmônica. A ideia de representar uma função como uma série trigonométrica foi inicialmente proposta por Fourier (1822). Para aplicação em sistemas digitais, essa ideia evoluiu para a Transformada Discreta de Fourier (DFT), um conceito que se beneficiou de métodos para intervalos discretos desenvolvidos por matemáticos como Gauss (1866). No entanto, a complexidade computacional da DFT direta era um obstáculo para aplicações em tempo real. O ponto de virada ocorreu com a publicação do algoritmo da Transformada Rápida de Fourier (FFT) por Cooley e Tukey (1965), um algoritmo computacional para calcular a DFT que reduziu drasticamente a complexidade de seu cálculo e viabilizou a análise espectral em sistemas com recursos limitados.

Hoje, a aplicação da FFT em microcontroladores de baixo custo, como o ESP32, tornou-se uma abordagem comum para o desenvolvimento de ferramentas didáticas e práticas. A capacidade de visualizar o espectro de áudio em tempo real transforma conceitos matemáticos abstratos em uma experiência interativa. Este tipo de projeto é visto em diversos projetos da comunidade *maker*, como o analisador de espectro de 32 bandas desenvolvido por Shajeeb (s.d.), que demonstra uma aplicação prática na plataforma Arduino, similar à proposta neste trabalho. Antes de processar o sinal, no entanto, é crucial garantir que a amostragem seja feita corretamente para evitar o fenômeno de *aliasing*, conforme estabelecido pelo Teorema de Nyquist-Shannon (SHANNON, 1949), que dita que a taxa de amostragem deve ser pelo menos o dobro da frequência máxima de interesse.

Nesse contexto, o presente trabalho tem como objetivo principal o desenvolvimento de um visualizador de espectro de áudio em tempo real. O sistema embarcado proposto utiliza um microcontrolador ESP32, um microfone analógico e uma matriz de LEDs para captar sons do ambiente e exibir graficamente seu conteúdo espectral. Para realizar a análise de frequência, empregou-se a biblioteca ArduinoFFT, que fornece uma implementação otimizada da Transformada Rápida de Fourier para microcontroladores. Dessa forma, o projeto visa demonstrar a aplicação prática da FFT, consolidando os conceitos teóricos em uma plataforma de aprendizado acessível e funcional.

2 JUSTIFICATIVA

O presente trabalho justifica-se pela importância, tanto teórica quanto prática, de demonstrar a aplicação de ferramentas matemáticas como a Transformada de Fourier e seu algoritmo otimizado (FFT) na modelagem e análise de sistemas na Engenharia da Computação e Eletrônica. Embora essenciais para a solução de problemas da área, esses conceitos são frequentemente abordados de maneira abstrata, dificultando a conexão entre a teoria e suas implementações práticas.

A principal contribuição desta pesquisa é o desenvolvimento de uma ferramenta didática e prática que serve como uma plataforma de aprendizado visual e interativa. Ao criar um sistema que traduz um sinal de áudio em uma representação gráfica em tempo real, o projeto facilita a compreensão intuitiva da análise espectral e evidencia a aplicação direta dos fundamentos matemáticos estudados.

Dessa forma, o estudo fortalece a compreensão conceitual e aplicada destes métodos, evidenciando sua utilidade na obtenção de resultados para problemas reais e no aprimoramento de sistemas tecnológicos. O projeto demonstra o papel central que a matemática desempenha no desenvolvimento de sistemas embarcados e reforça a importância de aprofundar o conhecimento sobre essas ferramentas para a inovação em diversas áreas da engenharia.



3 OBJETIVOS

3.1 Objetivo GERAL

Desenvolver um sistema embarcado para a visualização do espectro de áudio em tempo real, a fim de demonstrar a aplicação prática da Transformada Rápida de Fourier (FFT) e servir como uma ferramenta didática para a compreensão de conceitos de processamento de sinais.

3.2 Objetivos específicos

Pesquisar na literatura os fundamentos teóricos e os algoritmos necessários para a decomposição de sinais de áudio, com foco na Transformada Rápida de Fourier (FFT). Montar o circuito eletrônico do protótipo, integrando o microfone para captação de áudio, o microcontrolador ESP32 e a matriz de LEDs para a exibição. Desenvolver o software em linguagem C++ para realizar a amostragem do sinal de áudio e implementar as funções de análise de frequência por meio da biblioteca ArduinoFFT. Visualizar o conteúdo espectral do áudio em tempo real, representando graficamente a intensidade de cada banda de frequência na matriz de LEDs.

4 METODOLOGIA

O processo de construção deste projeto seguiu quatro etapas principais, desde a pesquisa teórica inicial até os testes de validação do protótipo.

O trabalho iniciou-se com a fundamentação teórica. Esta etapa envolveu o estudo dos algoritmos necessários para a decomposição de sinais de áudio, com foco na Transformada Rápida de Fourier (FFT). Ao mesmo tempo, foi feita uma pesquisa sobre a aplicação de análise espectral em sistemas embarcados de baixo custo, o que permitiu analisar projetos similares e estabelecer os requisitos da plataforma de hardware. Também foram definidos os critérios essenciais do processamento digital de sinais, como o Teorema de Nyquist-Shannon, que dita que a frequência de amostragem deve ser, no mínimo, o dobro da frequência máxima de interesse para evitar o fenômeno de *aliasing*.



A seleção de componentes e prototipagem veio a seguir. Como pode ser visto na prototipagem inicial fotografada na Figura 1. Com base nos requisitos definidos, o hardware selecionado incluiu o microcontrolador ESP32, o módulo de microfone MAX9814 e a matriz de LEDs P4 64x32 2. Um desafio central nesta etapa foi construir o circuito de condicionamento, um filtro passa-baixa RC de primeira ordem, necessário para atenuar frequências altas e prevenir o *aliasing*. O cálculo original do filtro (para 20 kHz com capacitor de 220 nF) apontou para um resistor de 36,2 Ω . Visto que este não é um valor comercial padrão, a equipe decidiu substituí-lo pelo resistor de 33 Ω . Esta troca resultou em uma nova frequência de corte real de 21,9 kHz, que foi considerada aceitável para o projeto.

A implementação do software foi feita em C++ na plataforma Arduino IDE. O código foi desenvolvido usando duas bibliotecas principais: a `ArduinoFFT`, para realizar a análise espectral, e a `ESP32-HUB75-MatrixPanel-DMA`, para controlar a matriz de LEDs de forma eficiente.

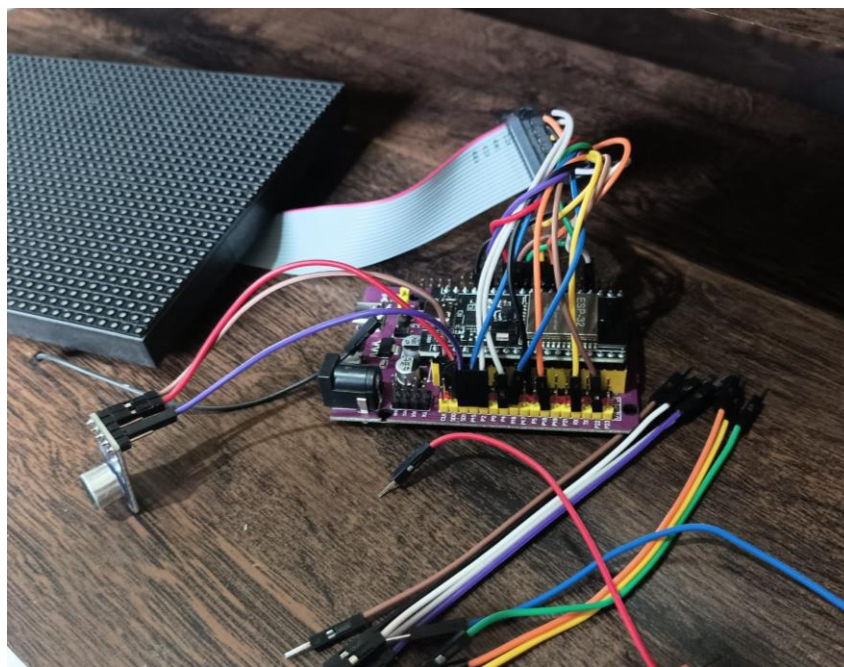


Figura 1 – Prototipagem inicial do sistema. Fonte: Elaborada pelos autores.

A última etapa foi de testes iterativos e resolução de problemas. Logo nos testes iniciais, foi notado um erro: bandas de alta frequência ficavam ativas mesmo em silêncio. A causa foi diagnosticada como interferência elétrica no cabo que levava o sinal analógico ao



microcontrolador. A solução foi simplesmente realocar e isolar melhor este cabo. Um segundo desafio apareceu nas limitações do próprio *driver ADC (Analog-to-Digital Converter)* do ESP32, que na prática reduzia a taxa de amostragem e causava *aliasing* em altas frequências. Este problema foi entendido como uma limitação de hardware do projeto, visto que era uma limitação da plataforma utilizada no projeto.

5 RESULTADOS OBTIDOS

O resultado alcançado neste trabalho foi um sistema embarcado funcional capaz de visualizar o espectro de áudio em tempo real. O dispositivo capta sons do ambiente e os representa graficamente em uma matriz de LEDs.

A lógica de funcionamento do projeto, segue um fluxo contínuo de captação, processamento e visualização, que pode ser resumido na Figura 2.



Figura 2 – Principais componentes e funções no projeto. Fonte: Elaborado pelos Autores.

Lógica de Funcionamento e Partes do Sistema

O sistema opera através de módulos de hardware e software que atuam em conjunto. Abaixo, o funcionamento de cada parte é explicado separadamente.

5.1 Captação e Condicionamento (Hardware)

A primeira etapa do processo é a captação do sinal sonoro, realizada pelo módulo de microfone MAX9814, um componente com pré-amplificador e controle automático de ganho (AGC) integrados. O sinal de saída do microfone é então condicionado por um filtro passa-



baixa RC de primeira ordem. Este filtro, composto por um resistor de $33\ \Omega$ e um capacitor de $220\ \text{nF}$, resulta em uma frequência de corte real de aproximadamente $21,9\ \text{kHz}$. Sua função é minimizar a atenuação de harmônicos relevantes e reduzir o risco de *aliasing* na etapa de digitalização. O diagrama completo do circuito é apresentado na Figura 3.

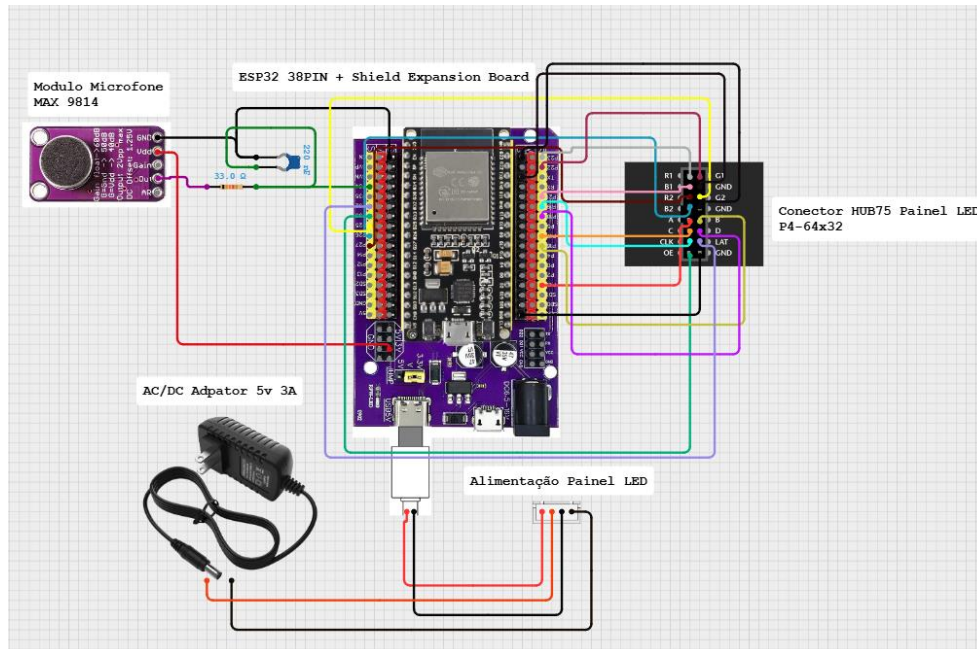


Figura 3 – Diagrama do circuito empregado no projeto. Fonte: Elaborada pelos autores.

5.2 Processamento Digital (Software - ESP32)

Esta é a etapa central do funcionamento. O sinal analógico filtrado é processado pelo ESP32 em uma sequência rigorosa:

Amostragem: O ESP32 realiza a amostragem do sinal através do pino GPIO 34. Com base no Teorema de Nyquist-Shannon, adotou-se uma taxa de amostragem de $40\ \text{kHz}$, o que permite representar componentes de até $20\ \text{kHz}$. Para cada ciclo, são coletadas 1024 amostras, um número que, por ser uma potência de 2, é ideal para otimizar o processamento da FFT. As amostras são armazenadas em um vetor (*Array*) *vReal*.

Processamento FFT: O código, desenvolvido em C++ com a biblioteca *ArduinoFFT*, executa quatro funções principais (Figura 4):



- **FFT.dcRemoval():** Remove o offset DC do sinal, comum em sensores como o MAX9814, evitando a concentração de energia na frequência zero.
- **FFT.windowing():** Aplica uma janela de *Hamming* para suavizar as bordas do sinal amostrado e prevenir o vazamento espectral (*spectral leakage*).
- **FFT.compute():** É o núcleo do processo. Esta função executa o algoritmo da FFT e converte os dados do domínio do tempo para o domínio da frequência, gerando coeficientes complexos.
- **FFT.ComplexToMagnitude():** Calcula a magnitude de cada coeficiente e armazena o resultado (a intensidade de cada frequência) de volta no vetor vReal[].

```
FFT.dcRemoval();  
FFT.windowing(FFT_WIN_TYP_HAMMING, FFT_FORWARD);  
FFT.compute(FFT_FORWARD);  
FFT.complexToMagnitude();
```

Figura 4 – Chamada de funções da biblioteca ArduinoFFT no código. Fonte: Elaborada pelos autores.

5.3 Interpretação e Visualização (Software e Hardware)

O resultado do processamento da FFT são 1024 "*bins*" (faixas de frequência), cada uma cobrindo aproximadamente 39,06 Hz. Devido à simetria espectral de sinais reais, apenas os primeiros 512 *bins* (correspondentes a 0-20 kHz) contêm informações únicas e são utilizados.

Para exibir esses dados na matriz de 32 colunas, os 512 *bins* são agrupados em 32 bandas. O resultado alcançado é que este agrupamento é logarítmico, uma abordagem que reflete de forma mais fiel a percepção auditiva humana. Esse procedimento resulta em faixas mais estreitas nas baixas frequências (graves) e progressivamente mais largas nas altas (agudos). A Figura 5 ilustra essa distribuição e o esquema de cores utilizado.

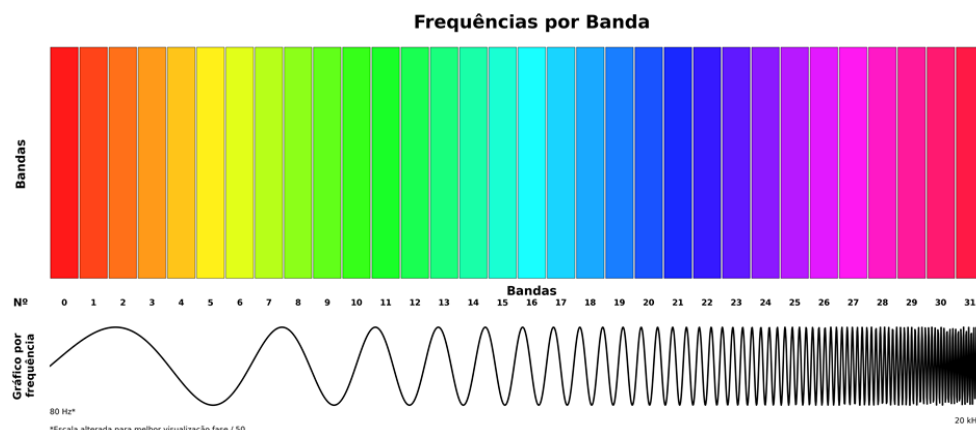


Figura 5 – Diagrama ilustrativo do agrupamento logarítmico e esquema de cores das 32 bandas. Fonte: Elaborada pelos autores.

5.4 Resultados Alcançados (Validação Visual)

O resultado é a exibição gráfica na matriz de LEDs RGB-P4 64x32. A lógica de funcionamento conjunta faz com que a altura das barras exibidas seja proporcional à magnitude (intensidade) de cada banda, e a cor varia conforme a faixa de frequência. A Figura 6 demonstra o protótipo em funcionamento, validando visualmente o resultado: o dispositivo reage em tempo real, fornecendo uma representação intuitiva do conteúdo espectral do áudio.



Figura 6 – Resultado do protótipo em funcionamento, reagindo a um estímulo musical. Fonte: Elaborada pelos autores.



6 CONCLUSÕES OU CONSIDERAÇÕES FINAIS

Este trabalho sistematizou o processo de desenvolvimento de um visualizador de espectro de áudio em tempo real, desde a concepção teórica até a validação do protótipo funcional. O projeto alcançou seu objetivo geral, que era demonstrar a aplicação prática da Transformada Rápida de Fourier (FFT) e servir como uma ferramenta didática para a compreensão do processamento de sinais.

Ao cruzar os objetivos específicos com os resultados obtidos, constata-se que todos foram atingidos:

- A pesquisa teórica foi realizada e fundamentou a escolha da FFT e a definição dos critérios de amostragem.
- O circuito eletrônico foi montado, integrando com sucesso o microfone MAX9814, o filtro RC e a matriz de LEDs ao ESP32, conforme demonstrado no diagrama do sistema.
- O software foi desenvolvido utilizando a biblioteca ArduinoFFT, implementando todas as etapas de processamento necessárias, desde a remoção do offset DC até o cálculo da magnitude.
- O objetivo final de visualizar o conteúdo espectral foi plenamente alcançado, com o sistema exibindo em tempo real o espectro de forma logarítmica, facilitando a interpretação visual.

Durante o desenvolvimento, a equipe encontrou desafios que exigiram soluções. O principal problema identificado foi a presença de ruído em bandas de alta frequência, mesmo com o ambiente em silêncio. A causa foi diagnosticada como interferência elétrica no cabo analógico. A solução encontrada foi a realocação e o melhor isolamento deste cabo, o que resolveu a questão. Também foi identificada uma limitação no *driver* ADC do ESP32, que provocava *aliasing* em frequências muito altas. Esta questão foi registrada como uma limitação da plataforma de hardware adotada, não comprometendo o objetivo didático do protótipo.

Conclui-se que a proposta funcionou como esperado. O balanço geral é positivo, pois o projeto não apenas validou a capacidade do ESP32 em realizar a análise espectral em



tempo real, mas também resultou em uma ferramenta eficaz para traduzir conceitos matemáticos abstratos em uma experiência visual intuitiva, reforçando a contribuição da engenharia para a educação.

REFERÊNCIAS

BAKER, Bonnie C. Anti-Aliasing, Analog Filters for Data Acquisition Systems. Application Note AN699. Chandler, AZ: Microchip Technology Inc., 1999. Disponível em: <https://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en011061>. Acesso em: 28 maio 2025.

BRIGHAM, E. Oran. The Fast Fourier Transform and Its Applications. Englewood Cliffs, NJ: Prentice Hall, 1988.

COOLEY, James W.; TUKEY, John W. An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation, v. 19, n. 90, p. 297-301, 1965.

FOURIER, Jean Baptiste Joseph. Théorie analytique de la chaleur. Paris: Gauthier-Villars, 1888.

GAUSS, C. F. Theoria interpolationis methodo nova tractata. In: Werke band 3. Göttingen: Königliche Gesellschaft der Wissenschaften, 1886. p. 265–327.

KAPER, Hans G.; TIPEI, Sever; WIEBEL, Elizabeth. Data sonification and sound visualization. arXiv:cs/0007007v1 [cs.SD], 5 jul. 2000. Disponível em: <https://arxiv.org/abs/cs/0007007v1>. Acesso em: 26 maio 2025.

OPPENHEIM, A. V.; WILLSKY, A. S.; NAWAB, S. H. Signals and Systems. 2. ed. Upper Saddle River, NJ: Prentice Hall, 1997.

SHANNON, Claude E. Communication in the Presence of Noise. Proceedings of the IRE, v. 37, n. 1, p. 10–21, jan. 1949.

SHAJEEB. 32-Band Audio Spectrum Visualizer Analyzer. Arduino Project Hub, s.d. Disponível em: <https://projecthub.arduino.cc/shajeeb/32-band-audio-spectrum-visualizer-analyzer-924af5>. Acesso: 18 abril 2025.



APÊNDICE A – Código-fonte do projeto

O código-fonte completo do trabalho, desenvolvido pelos autores, está disponível no seguinte repositório do GitHub:

https://github.com/jotapedro7/ESP32-FFT_AudioVisualizer.