

**INSTITUTO
FEDERAL**
Paraíba

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus Campina Grande

Curso de Bacharelado em Engenharia de Computação

Trabalho de Conclusão de Curso

Luiz Henrique da Silva Oliveira

**HARMONY WAVE: SISTEMA EMBARCADO PARA EFEITOS DE GUITARRA EM
TEMPO REAL**

Campina Grande - PB
2025

Luiz Henrique da Silva Oliveira

**HARMONY WAVE: SISTEMA EMBARCADO PARA EFEITOS DE GUITARRA EM
TEMPO REAL**

Trabalho de Conclusão de Curso apresentado ao
Instituto Federal da Paraíba – Campus Campina
Grande, em cumprimento às exigências parciais
para a obtenção do título de bacharel em
Engenharia de Computação.

Orientador: Prof. Dr. Fagner de Araújo Pereira

Campina Grande - PB
2025

O48h Oliveira, Luiz Henrique da Silva

Harmony Wave: sistema embarcado para efeitos de guitarra em tempo real / Luiz Henrique da Silva Oliveira. - Campina Grande, 2025.

66 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Engenharia de Computação) - Instituto Federal da Paraíba, 2025.

Orientador: Prof. Dr. Fagner de Araújo Pereira.

1. Engenharia de computação 2. Processamento digital de sinais 3. Guitarra elétrica - instrumento musical 4. Música - Efeitos sonoros I. Pereira, Fagner de Araújo III. Título.

CDU 621.39

À Deus. À meus pais e amigos, por todo apoio e carinho!

Dedico!

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me deu forças para concluir o curso e perseverar.

Ao professor e orientador Dr. Fagner de Araújo Pereira, pois sem seu apoio esse trabalho teria sido muito mais árduo, seu conhecimento tanto musical quanto de engenharia foi crucial, além de ser um professor muito carismático.

Aos demais professores pelo conhecimento transmitido.

A minha mãe Espedita Lúcia que me apoiou em todos os momentos e nas minhas escolhas, desde o momento que decidi cursar Engenharia de Computação até o dia que vim morar em Campina Grande

Ao meu padrasto José Ivo que me ajudou muito, principalmente nas viagens de São José do Egito para Campina Grande e quando me mudei.

Ao meu pai José Ronaldo, que também sempre me apoiou na minha decisão e sempre esteve ao meu lado.

A minha madrastra Maria de Fátima por todo o carinho.

Aos meus amigos Luiz Eduardo Veronese, João Vinicius e Caio Graco pelo apoio emocional durante a jornada.

Ao meu líder direto no projeto em que atuo na empresa Accenture, Yago Campos, por sempre demonstrar compreensão e apoio sempre que precisei me ausentar do trabalho para me dedicar às aulas da faculdade e ao desenvolvimento deste projeto. Sua disposição em facilitar esse processo fez toda a diferença. Seu incentivo e flexibilidade me permitiram equilibrar as demandas acadêmicas e profissionais, e por isso sou imensamente grato.

Por fim, ao Instituto Federal da Paraíba, pela oportunidade da realização do curso.

“So carry on, there's a meaning to life, which someday we may find...”

Angra, Carry On

RESUMO

A guitarra elétrica é um instrumento musical que, para muitos estilos musicais, necessita de alguns equipamentos extras para atingir o seu objetivo, já que por si só o instrumento não é capaz de reproduzir a variedade de timbres e sons desejados. Este trabalho tem como foco o desenvolvimento de uma estação de efeitos personalizáveis para modificar o som do instrumento, abordando os conceitos de processamento digital de sinais. Foi projetado um sistema microcontrolado utilizando um chip STM32, onde foi desenvolvida uma plataforma de hardware e software. O sistema é um misto de componentes analógicos e digitais: Capacitores, resistores, amplificadores operacionais, CODEC de áudio e microcontrolador. Quanto a parte dos efeitos projetados, foram desenvolvidos: Overdrive, delay e um equalizador de frequências. Após finalizar o desenvolvimento e testes, o trabalho apresenta o esquemático final de hardware, o custo do produto, que conseguiu ficar abaixo dos R\$ 400,00, um custo bem baixo para o que oferece, e algumas amostras de áudio disponíveis na plataforma Soundcloud.

Palavras-chave: Processamento digital de sinais; Guitarra elétrica; STM32; Música.

ABSTRACT

The electric guitar is a musical instrument that, for many musical styles, requires some extra equipment to achieve its purpose, since the instrument alone is not capable of reproducing the variety of timbres and sounds desired. This work focuses on the development of a customizable effects station to modify the sound of the instrument, addressing the concepts of digital signal processing. A microcontrolled system was designed using an STM32 chip, where a hardware and software platform was developed. The system is a mix of analog and digital components: Capacitors, resistors, operational amplifiers, audio CODEC and microcontroller. As for the designed effects, the following were developed: Overdrive, delay and a frequency equalizer. After completing the development and testing, the work presents the final hardware schematic, the cost of the product, which managed to stay below R\$ 400.00, a very low cost for what it offers, and some audio samples available on the Soundcloud platform.

Keywords: Digital signal processing; Electric guitar; STM32; Music.

LISTA DE ILUSTRAÇÕES

Figura 1: Componentes de um captador de guitarra.....	17
Figura 2: Posicionamento comum dos captadores da guitarra elétrica.....	18
Figura 3: representação dos sinais: no tempo contínuo (a), no tempo discreto (b) e trem de impulsos no tempo contínuo (c).....	19
Figura 4: Sinal de 1Hz sendo amostrado a 2 Hz.....	20
Figura 5: Sinal com frequência de amostragem insuficiente $f_s = 1,5$ Hz e $f_c = 2$ Hz.....	20
Figura 6: Possível interpretação errônea do sinal da Figura 5 ao receber as amostras.....	21
Figura 7: Espectro de sinais amostrados.....	21
Figura 8: Exemplo de como um sinal é visto no domínio do tempo e no domínio da frequência.....	23
Figura 9: Plano complexo z com o X simbolizando os polos.....	24
Figura 10: Diagrama de blocos de um filtro FIR.....	25
Figura 11: Resposta ao impulso de um filtro FIR.....	26
Figura 12: Diagrama de blocos de um filtro IIR.....	27
Figura 13: Diagrama de temporização do I2S.....	28
Figura 14: Diagrama de blocos do projeto Harmony Wave.....	30
Figura 15: Esquemático do buffer analógico.....	31
Figura 16: Filtro Anti-Aliasing.....	32
Figura 17: Conversor single-ended para diferencial.....	33
Figura 18: Conversor diferencial para single-ended.....	34
Figura 19: Protótipo do projeto finalizado.....	36
Figura 20: Funcionamento do Double Buffering.....	37
Figura 21: Função para implementação do overdrive em tempo discreto.....	38
Figura 22: Gráfico que representa o efeito do overdrive.....	38
Figura 23: Influência do ganho no formato da onda.....	39
Figura 24: FFT de senóides que passarão pelo overdrive.....	40
Figura 25: Estrutura do filtro IIR que implementa o delay.....	42
Figura 26: Funcionamento dos filtros de pico.....	43
Figura 27: Variáveis que modelam o comportamento do filtro.....	44
Figura 28: Esquemático da simulação com osciloscópio.....	46
Figura 29: Resultado obtido ao injetar uma senoide de 1kHz com 1V de pico..	46
Figura 30: Resultado obtido ao injetar uma senoide de 50kHz com 1V de pico	47
Figura 31: Circuito montado para experimentação dos sinais diferenciais.....	48
Figura 32: Sinal de entrada, senoide de 1kHz com 1V de pico.....	48
Figura 33: Sinal diferencial gerado e validado com sucesso.....	49
Figura 34: Esquemático do projeto Harmony Wave.....	51
Figura 35: Protótipo em funcionamento.....	52
Figura 36: Sinais sem efeito aplicado.....	55

Figura 37: FFT do sinal de entrada.....	55
Figura 38: FFT do sinal de saída.....	56
Figura 39: Atenuação de ruído grave.....	56
Figura 40: Atraso de processamento.....	57
Figura 41: Overdrive com ganho 2 sobre a senoide.....	57
Figura 42: Overdrive com ganho 50 sobre a senoide.....	58
Figura 43: Repetições do delay.....	58
Figura 44: Amplificação do grave.....	59
Figura 45: Atenuação do grave.....	59
Figura 46: Equipamento utilizado para gravar demonstrações.....	60

LISTA DE TABELAS

Tabela 1: Custo dos componentes do projeto.....	53
--	-----------

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog to Digital Converter
AMPOP	Amplificador Operacional
CI	Circuito Integrado
CODEC	Codificador/Decodificador
CPU	Central Processing Unit
DAC	Digital to Analog Converter
DMA	Direct Memory Access
DSP	Digital Signal Processing
FC	Frequência de Corte
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FS	Frequency Sampling
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
IIR	Infinite Impulse Response
LCD	Liquid Crystal Display
PDS	Processamento Digital de Sinais
RAM	Random Access Memory

SUMÁRIO

1.0 Introdução.....	13
1.1 Objetivo Geral.....	13
1.2 Objetivos Específicos.....	13
1.3 Justificativa.....	14
2.0 Fundamentação teórica.....	15
2.1 Evolução dos efeitos de guitarra durante o tempo.....	15
2.2 Processamento Digital de Sinais (PDS).....	16
2.3 Conversores Analógico-Digital.....	16
2.4 Teorema da Amostragem.....	18
2.5 Transformada de Fourier.....	22
2.6 Transformada Z.....	23
2.7 Filtros FIR.....	24
2.8 Filtros IIR.....	26
2.9 Protocolos I2C e I2S.....	27
3.0 Metodologia.....	29
3.1 Levantamento dos requisitos do projeto.....	29
3.2 Arquitetura do Hardware.....	30
3.3 Componentes do circuito.....	34
3.4 Processamento do áudio e estratégia de double buffering.....	36
3.5 Implementação dos efeitos de áudio.....	37
3.5.1 Overdrive.....	37
3.5.2 Delay.....	41
3.5.3 Equalizador de frequências.....	43
3.6 Teste e Validação.....	45
4.0 Resultados.....	51
4.1 Esquemático e montagem do circuito.....	51
4.2 Custo de produção do projeto.....	52
4.3 Sinais e filtros.....	54
4.4 Demonstração sonora.....	59
5.0 Conclusão.....	62
5.1 Trabalhos Futuros.....	62
REFERÊNCIAS.....	64

1.0 Introdução

O aprendizado de instrumentos musicais é um hobby que trás inúmeros benefícios para a vida, habilidades como coordenação motora, concentração, disciplina, raciocínio, memória e etc.

A música é uma arte que está intrinsecamente associada à história humana, onde os primeiros instrumentos surgiram na pré-história, onde eram usados pedras e troncos de árvores para reproduzir sons. Segundo Mulin (2015) a música estabelece relação direta com a experiência humana, tornando-se assim objeto de estudo interdisciplinar.

Um dos instrumentos mais utilizados nos dias de hoje é a guitarra elétrica, instrumento que traz sons marcantes, principalmente na região dos médios e agudos, e que é indispensável para alguns estilos como o rock n roll e o heavy metal.

A guitarra, por ser um instrumento elétrico, necessita de um conjunto de equipamentos para capturar seu som e processá-lo, modelando a gosto do músico, como captadores, pedais de efeitos, pedaleiras e amplificadores.

Visando atender às necessidades de músicos iniciantes que estão explorando a guitarra elétrica, o presente trabalho tem como objetivo o desenvolvimento de uma estação de efeitos baseada em um microcontrolador STM32. O equipamento foi projetado para proporcionar uma experiência sonora versátil e acessível, permitindo a manipulação de efeitos.

1.1 Objetivo Geral

Desenvolver um protótipo de estação de efeitos de guitarra baseado em um sistema microcontrolado onde os efeitos podem ser customizáveis, voltado para iniciantes e curiosos na área, com baixo investimento inicial.

1.2 Objetivos Específicos

- Desenvolver o hardware de processamento digital do sinal da guitarra elétrica;
- Desenvolver os efeitos (overdrive, delay e equalizador) via software para o sistema microcontrolado;
- Desenvolver uma interface de usuário simples;
- Analisar custos do sistema.

1.3 Justificativa

Ao concluir o projeto, aprendizes de guitarra elétrica poderão ter uma pedaleira totalmente customizável, onde ao efetuar a troca do firmware do sistema, novos efeitos podem ser adicionados, os existentes podem ser atualizados e a gama de possibilidades sonoras pode ser expandida.

O sistema possibilitará aos iniciantes conhecer os mais diversos efeitos disponíveis para poder praticar e desenvolver habilidades no instrumento e sem a frustração de ter que ficar limitado aos efeitos pré-carregados na estação de efeitos, sem poder trocá-los. Com uma interface simples, o aprendiz poderá interagir com o sistema sem grandes dificuldades e a curva de aprendizagem será bem menor.

Oferecendo um sistema de PDS customizável de baixo custo, o projeto pode bater de frente com os que estão disponíveis hoje no mercado, que oferecem poucas possibilidades de mudança, e os que oferecem a customização como recursos costumam ter o preço bem mais elevado.

2.0 Fundamentação teórica

2.1 Evolução dos efeitos de guitarra durante o tempo

No passado, eram utilizadas câmaras de eco para simular efeito de reverberação (como os de grandes catedrais, por exemplo), uma caixa de som e um microfone eram posicionadas dentro da câmara e a gravação original sem eco era tocada e regravada. É possível que este seja o efeito sonoro mais antigo (Weir, 2012 apud Freitas, 2019).

Um dos principais efeitos que mudou a história da guitarra elétrica foi o efeito de distorção que antes era alcançado pelo aumento exagerado do volume dos amplificadores valvulados, ou até mesmo nos casos mais extremos, danificando-os. Um dos primeiros exemplos de distorção pode ser ouvido na música de Bob Wills, de 1947 intitulada de “Bob Wills Boogie” (Slavković, s.d.).

Com o surgimento dos transistores, o processamento digital de sinais começou a ser utilizado para produção de novos efeitos, em 1962 surgiu o primeiro pedal de distorção FZ-1 produzido pela Gibson que utilizava alguns transistores de germânio, popularizado somente após o lançamento da música “Satisfaction” da banda Rolling Stones (Wedyke, 2021).

Em 1973 surge como alternativa o primeiro pedal de delay digital, o Lexicon Delta-T, o delay que antes utilizava de fitas magnéticas para gerar o sinal atrasado Wilmering et al. (2020) a partir da distância da cabeça de escrita para cabeça de leitura dos gravadores agora podia ser obtido sem componentes mecânicos.

A partir da década de 80 as primeiras estações digitais multi efeito tiveram seu lançamento, empresas como a Line 6 e a Zoom foram pioneiras nesse ramo, reduzindo a necessidade de estandes de pedais enormes e permitindo a utilização de presets com a mistura de vários efeitos, podendo ser trocados rapidamente (Joseph, 2024).

Com a constante evolução da tecnologia atualmente, tanto os pedais de efeito analógicos/digitais quanto as estações multi efeito continuam aprimorando, há músicos que preferem a sonoridade analógica com um set de pedais muito vasto, também há quem prefira a praticidade das estações multi efeito popularmente conhecida como pedaleira e existe também quem faz a mescla dos dois mundos, possibilitando uma grande variedade de timbres.

2.2 Processamento Digital de Sinais (PDS)

Segundo Freitas (2019), PDS é um ramo da engenharia que aborda muitas áreas. Se trata de uma ciência que busca transformar sinais para extrair informações, melhorá-los ou obter um sinal diferente a partir do sinal recebido. Essas transformações são obtidas através de manipulações matemáticas e as aplicações são diversas.

O PDS teve destaque em duas áreas do ramo tecnológico, sendo elas: processamento digital de imagens e processamento digital de áudio (Freitas, 2019).

Como destacado pelo autor, o PDS hoje é amplamente utilizado, em várias áreas como: biomedicina, processamento de áudio, processamento de imagens, etc.

O fato de conseguirmos representar grandezas contínuas de forma digital para que se faça o processamento em computadores é de extrema importância. Para processar sinais em tempo contínuo, primeiro precisamos efetuar a sua discretização, processar digitalmente a entrada e por fim converter a saída no tempo discreto de volta ao domínio do tempo contínuo (Diniz; Silva; Netto, 2014, p.33).

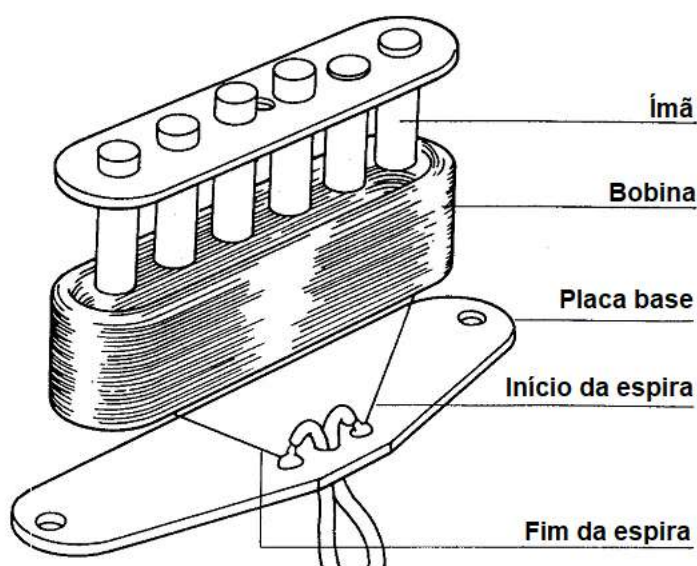
No ramo do Processamento Digital de Sinais, trabalha-se com os sinais discretizados que são representações digitais dos sinais analógicos contínuos do mundo real, essas amostras são obtidas através de um circuito conversor analógico-digital, que será descrito na próxima seção.

2.3 Conversores Analógico-Digital

Como citado por Puhlmann(2015), as grandezas físicas do mundo real não costumam ser grandezas elétricas. Para podermos trabalhar com essas grandezas devemos, de alguma forma, convertê-las em sinais elétricos. Para isso, usamos dispositivos chamados sensores ou transdutores.

O conceito acima pode ser observado na guitarra elétrica, onde temos os captadores que são ímãs envoltos por uma bobina. Como as cordas da guitarra são constituídas de uma liga metálica com propriedades ferromagnéticas, ao movimentá-las, haverá uma mudança na densidade de campo magnético e esta variação do campo vai induzir uma corrente elétrica na bobina, produzindo assim um sinal elétrico, cuja frequência corresponde à nota que está sendo tocada no instrumento. Podemos observar a estrutura de um captador na Figura 1 e o seu posicionamento no instrumento na Figura 2.

Figura 1: Componentes de um captador de guitarra



Fonte: School of Rock Cerqueira César¹

¹ Disponível em:

<https://www.schoolofrock.com.br/unidades/cerqueiracesar/resources/guitar/captadores-o-que-so-para-que-servem-como-funcionam-pt1>. Acesso em: 05 nov. 2024.

Figura 2: Posicionamento comum dos captadores da guitarra elétrica



Fonte: Beat 360²

Uma vez que o sinal foi capturado, ele ainda é analógico e contínuo no tempo, ou seja, assume infinitos valores dentro de um intervalo, o que impossibilita seu processamento por um sistema computacional, pois nunca haverá memória suficiente para armazenar e processar infinitos valores. Aqui entra um componente essencial: o conversor analógico-digital, que realizará a transformação no sinal, de forma que ele possa ser processado digitalmente.

Uma vez que o sinal analógico foi digitalizado e transformado em um sinal amostrado, ele será representado por valores inteiros finitos, determinados pela resolução (em bits) do conversor (8, 10, 12, 16, 24, etc). (Puhlmann, 2015).

2.4 Teorema da Amostragem

Segundo Diniz et al. (2014, p.34), para obter um sinal amostrado, precisaremos modelar o sinal analógico de forma que possa ser representado por um trem de sinais impulso com tempo $t = nT$.

A equação (1) representa o sinal amostrado por trem de impulsos:

$$x_i(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \delta(t - nT) \quad (1)$$

Onde:

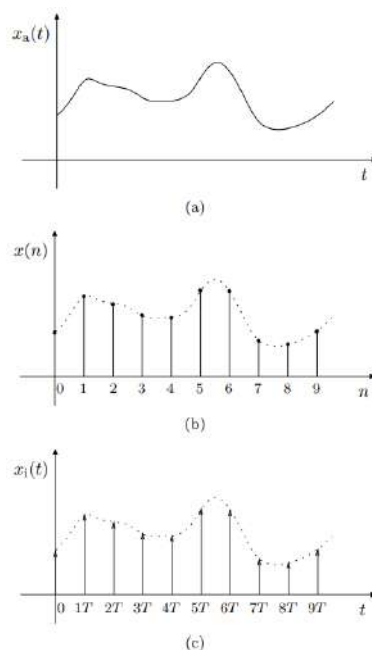
- $x[n]$ é uma amostra do sinal

² Disponível em: <https://beat360.com.br/melhores-captadores-para-guitarra/>. Acesso em 05 nov. 2024.

- $\delta(t - nT)$ é um sinal impulso

Na figura 3 podemos observar como funciona o processo descrito.

Figura 3: representação dos sinais: no tempo contínuo (a), no tempo discreto (b) e trem de impulsos no tempo contínuo (c).

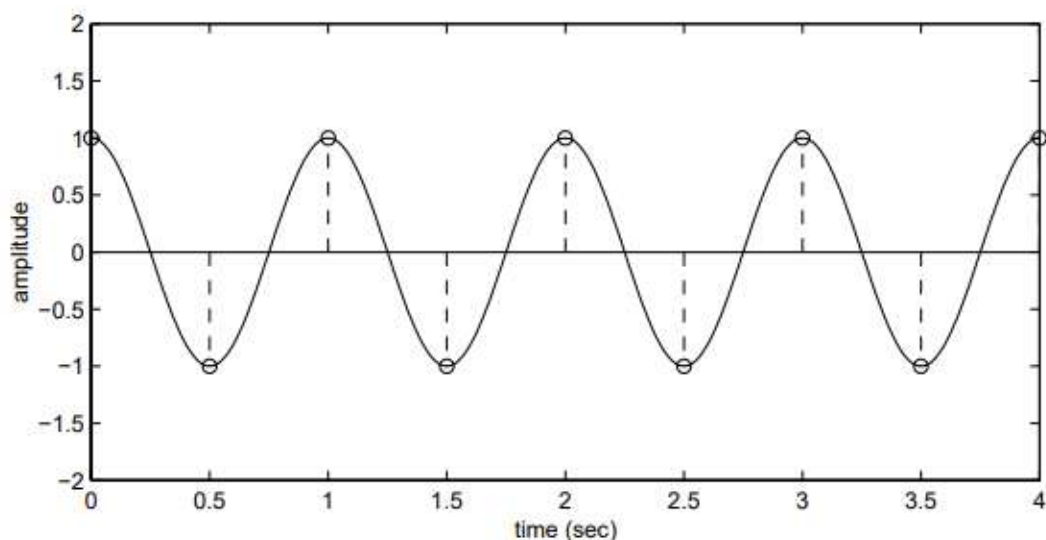


Fonte: Diniz et al. (2014, p.34)

Um conceito bastante importante para amostragem citado por Olshausen (2000) é o teorema de Nyquist que dita o seguinte: “para que um sinal seja amostrado de forma que se possa construir o sinal original de volta, deve-se amostrar o sinal com pelo menos duas vezes a frequência máxima do sinal de entrada”.

Em termos matemáticos: $f_s \geq 2f_c$. Onde f_s é a frequência de amostragem e f_c é a frequência máxima do sinal a ser amostrado.

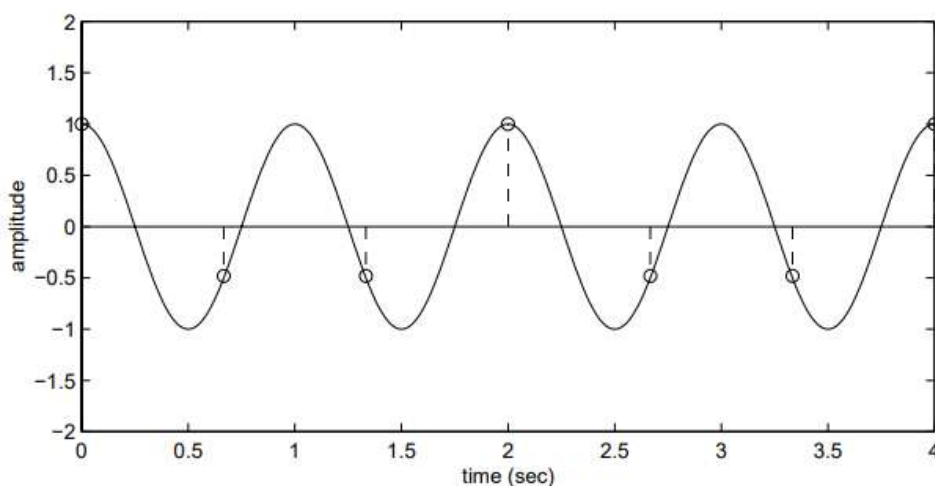
Na figura 4 podemos observar de forma intuitiva o teorema de Nyquist em funcionamento, onde nota-se que na amostragem a 2 Hz é possível capturar todos os picos e vales do sinal de forma precisa, algo que não seria possível com uma frequência de amostragem menor.

Figura 4: Sinal de 1Hz sendo amostrado a 2 Hz

Fonte: Olshausen (2000)

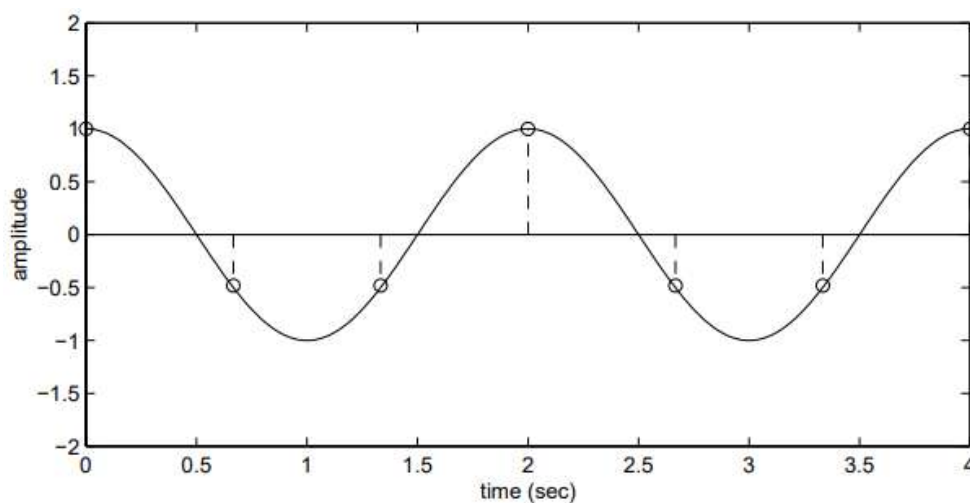
Caso não se respeite o teorema de Nyquist teremos no nosso sinal um fenômeno chamado aliasing, que acarreta não só em perda de informação, mas também a possível obtenção de uma forma de onda errada quando for feita a reconstrução do sinal em um conversor digital-analógico por exemplo.

Na figura 5 podemos ver um sinal que sofre com o problema do aliasing e na figura 6 a possível interpretação que alguém poderia ter do mesmo sinal, dadas as suas amostras.

Figura 5: Sinal com frequência de amostragem insuficiente $f_s = 1,5$ Hz e $f_c = 2$ Hz

Fonte: Olshausen (2000)

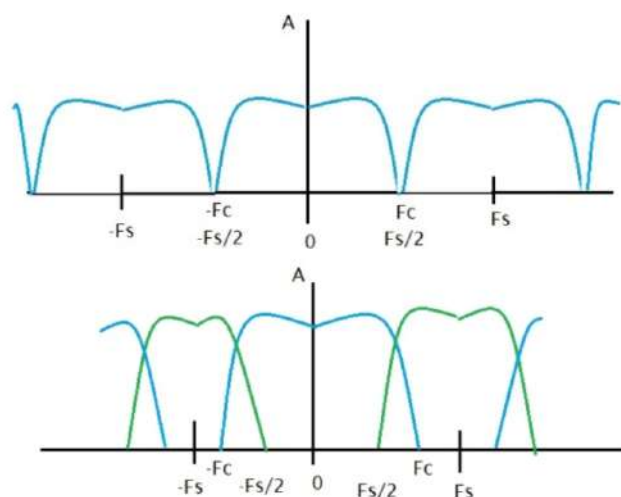
Figura 6: Possível interpretação errônea do sinal da Figura 5 ao receber as amostras.



Fonte: Olshausen (2000)

No domínio da frequência acontece a sobreposição do espectro, ou seja, sinais de frequências mais altas podem ser interpretados como um sinal com componentes de frequência menores, gerando um sinal que não foi o realmente gerado, na parte de cima da figura 7 temos o espectro de um sinal amostrado com exata duas vezes a frequência máxima do sistema e na parte de baixo um sinal que não respeitou o teorema de Nyquist.

Figura 7: Espectro de sinais amostrados



Fonte: Blog do Engenheiro Puhlmann³

³ Disponível em:

<https://consuliteengenharioeletronico.wordpress.com/2014/03/27/processamento-digital-de-sinais-dsp-parte-2/>. Acesso em 14 mar. 2025.

2.5 Transformada de Fourier

Na seção anterior foi apresentado o conceito de amostragem e como a operação funciona no domínio do tempo. Na área de PDS, muitas vezes é necessário analisar os sinais também no domínio da frequência e para isso existe uma ferramenta matemática chamada Transformada de Fourier.

No tempo discreto, a transformada de Fourier representa um sinal como uma soma infinita de senoides complexas, também no tempo discreto Diniz et al. (2014, p.123).

A transformada discreta de Fourier é dada pela seguinte fórmula:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \quad (2)$$

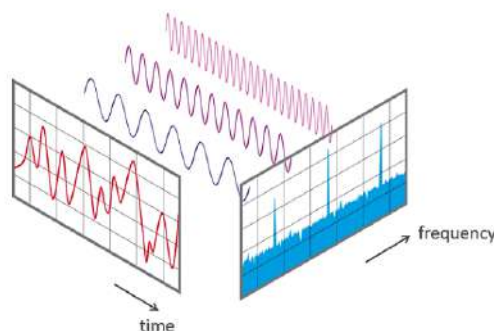
Onde:

- ω é a frequência angular variando de $-\pi$ até π .
- $x[n]$ é a amostra analisada.
- $e^{-j\omega n}$ uma exponencial complexa que vai gerar as oscilações das frequências que o sinal possui.

Para projetar as mais diversas estruturas de filtros, a análise do comportamento em frequência do sinal é extremamente importante. Como o cálculo da equação (2) é bastante custoso computacionalmente, foi descoberto um algoritmo que pode acelerar este cálculo, que recebe o nome de transformada rápida de Fourier ou *Fast Fourier Transform* (FFT), como é conhecido em inglês. Este algoritmo reduz significativamente o tempo de execução da transformada segundo Bergland (1968 apud Spilsbury;Euceda, 2016).

Na figura 8 podemos observar o resultado de uma FFT e as informações que ela nos traz.

Figura 8: Exemplo de como um sinal é visto no domínio do tempo e no domínio da frequência



Fonte: NTI AUDIO⁴

2.6 Transformada Z

Outra transformada bastante importante para análise de sinais em PDS é a transformada Z, que também converte as funções no domínio do tempo para o domínio da frequência, porém na transformada Z também obtemos informações se um determinado sistema é estável ou não.

Como citado por Tonidandel (2010), a transformada Z é equivalente a transformada de Laplace, enquanto a transformada de Laplace é utilizada no tempo contínuo no domínio (s), a transformada Z é utilizada no tempo discreto no domínio (z).

Sua equação é dada por Tonidandel (2010) e é bastante semelhante a transformada de Fourier:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n} \quad (3)$$

⁴ Disponível em: <https://www.nti-audio.com/pt/suporte/saber-como/transformacao-rapida-de-fourier-fft>

Podemos usar a transformada Z para analisar as equações de diferenças de um sistema discreto linear e invariante no tempo, resolvendo sua função de transferência que é dada por Diniz et al. (2014, p.112).:

$$H(z) = \frac{Y(z)}{X(z)} \quad (4)$$

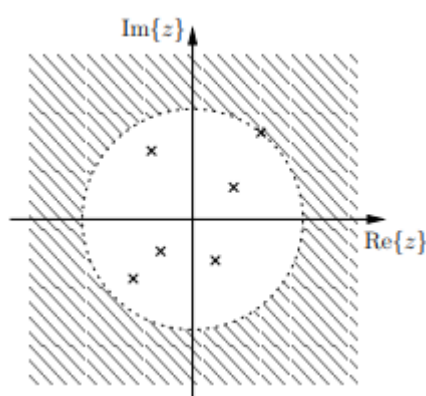
Onde:

- $Y(z)$ é a transformada Z da saída do sistema.
- $X(z)$ é a transformada Z da entrada do sistema.

Outros dois conceitos importantes são os de zeros e pólos da função de transferência, pois eles são úteis no projeto de filtros digitais. Os zeros basicamente são as raízes do numerador $Y(z)$ e os pólos são as raízes do denominador $X(z)$ Phil's Lab (2021).

Para determinar se um sistema é estável, é necessário avaliar seus pólos dentro do plano complexo z que é mostrado na figura 9. Se todos os pólos estiverem dentro do círculo unitário significa que o sistema é estável Phil's Lab (2021).

Figura 9: Plano complexo z com o X simbolizando os polos



Fonte: Diniz et al. (2014, p.87)

2.7 Filtros FIR

O sinal de áudio pode ser representado como a soma de várias senóides com amplitudes e frequências distintas. Os filtros são elementos que alteram a amplitude dessas senóides Zolzer (2011 apud Medeiros, 2019).

Dentro da categoria de filtros digitais, existe a categoria conhecida como filtro FIR (Finite Impulse Response), chamada assim porque sua resposta a um impulso unitário dura apenas um intervalo de tempo finito. Ela permite a construção de filtros sofisticados, permitindo forte atenuação de frequências ou extrair de um sinal suas várias bandas de frequência Zolzer (2011).

Normalmente, no processamento digital de sinais de áudio, são usados filtros de ordem mais alta, que envolvem mais coeficientes. Logo, é necessário mais poder computacional do que os filtros de resposta infinita ao impulso Zolzer (2011) que serão explicados na próxima seção e envolvem muito menos coeficientes.

A estrutura do filtro FIR pode ser descrita pela seguinte equação de diferenças Zolzer (2011):

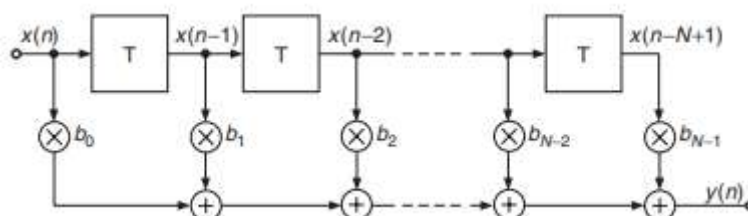
$$y[n] = \sum_{i=0}^{N-1} b_i \cdot x[n - i] \quad (5)$$

Sendo:

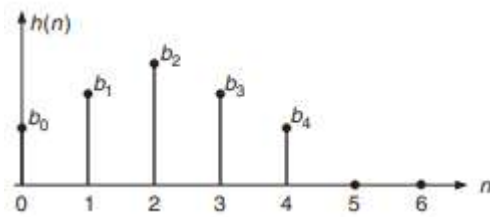
- N a ordem do filtro.
- b os coeficientes.
- $x[n - i]$ a amostra do sinal.

Nas figuras 10 e 11 podemos observar o diagrama de blocos de um filtro FIR e um exemplo de resposta ao impulso.

Figura 10: Diagrama de blocos de um filtro FIR



Fonte: Zolzer (2011, p.40)

Figura 11: Resposta ao impulso de um filtro FIR

Fonte: Zolzer (2011)

2.8 Filtros IIR

Diferente dos filtros FIR, nos quais a saída só depende das entradas do sistema, nos filtros IIR (Infinite Impulse Response) as amostras de saída também são afetadas por uma realimentação das saídas anteriores em combinação com as entradas do sistema Medeiros (2019).

Geralmente os filtros IIR se aproximam de uma resposta em frequência fazendo menos multiplicações que os filtros FIR. Por isso, para determinadas situações, eles podem ser mais adequados para o processamento de sistemas em tempo real Diniz et al. (2014, p.383), como por exemplo sistemas microcontrolados, que possuem menos recursos computacionais.

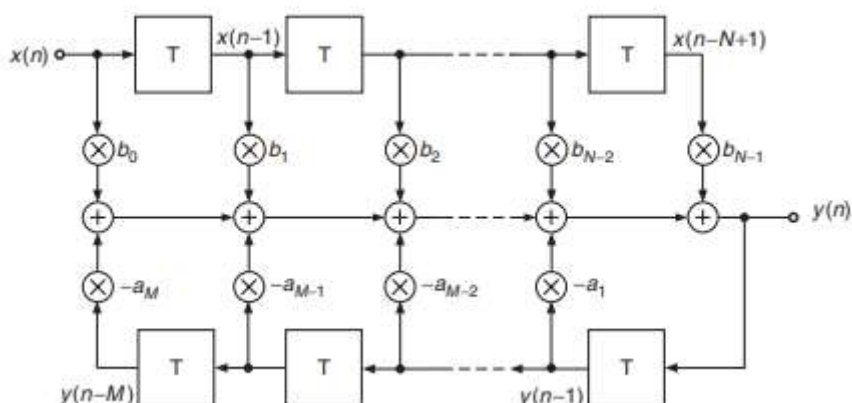
Os filtros IIR podem ser representados pela seguinte equação de diferenças (Oppenheim;Schafer, 1998 apud Medeiros, 2019):

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_Mx(n-M) - a_1y(n-1) - \dots - a_Ny(n-M) \quad (6)$$

Onde $b_0 \dots b_M$ e $a_0 \dots a_N$ são os coeficientes do polinômio que caracteriza o numerador e denominador da transformada Z do filtro.

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_Mz^{-M}}{1 + a_1z^{-1} + \dots + a_Nz^{-N}} \quad (7)$$

Já que na maioria dos casos as funções de transferência citadas na equação (7) resultam em uma resposta ao impulso de comprimento infinito, os filtros IIR recebem este nome justamente por causa deste fator associado.

Figura 12: Diagrama de blocos de um filtro IIR

Fonte: Zolzer (2011, p.38)

Na figura 12 podemos observar como funciona o diagrama de blocos de um filtro IIR, sendo fácil perceber a sua realimentação baseada nas saídas anteriores do sistema.

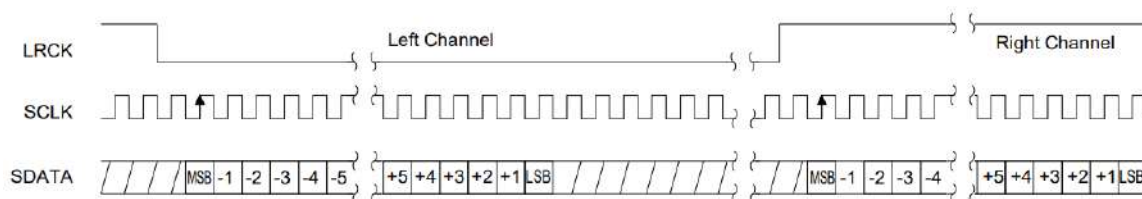
2.9 Protocolos I2C e I2S

Dois protocolos extremamente importantes para o processamento de áudio são os protocolos I2C e I2S. O protocolo I2C é comumente usado para comunicação de circuitos integrados e o protocolo I2S é dedicado para transmissão de amostras de áudio entre circuitos integrados.

O protocolo I2C (Inter-Integrated Circuit) atua com duas linhas bidirecionais: a linha de dados (SDA) e a linha de clock (SCL), nessas duas linhas podem ser conectados vários periféricos, pois no protocolo cada dispositivo terá seu endereço único que será informado durante a construção do pacote de dados enviado.

Já o protocolo I2S (Inter-IC Sound) é um protocolo de comunicação que é utilizado para comunicação entre dispositivos que atuam com áudio digital. Funciona com três linhas: a linha de clock serial (SCK), a linha de dados (SD) e a linha de word select (WS) também conhecida como Left-Right Clock (LRCLK), que é usada para selecionar o canal ao qual o áudio vai ser direcionado em sistemas stereo.

A figura 13 apresenta um diagrama de como funcionam os sinais do protocolo I2S

Figura 13: Diagrama de temporização do I2S

Fonte: CIRRUS LOGIC (2021)

Dados os conceitos apresentados, este trabalho descreve a criação de um protótipo de estação de efeitos digitais, onde os efeitos serão microcontrolados e serão escaláveis, ou seja, para o mesmo hardware, teremos a possibilidade de utilizar quantos efeitos desejarmos somente com uma troca de firmware do sistema.

3.0 Metodologia

A metodologia da construção do protótipo da estação de efeitos de guitarra, foi dividida basicamente em três etapas:

- Levantamento dos requisitos necessários para processamento de sinais de áudio;
- Construção do hardware;
- Programação do DSP.

Essas etapas serão melhor explicadas nas próximas seções, e será possível ver como foi solucionado o problema explanado na introdução do atual trabalho.

3.1 Levantamento dos requisitos do projeto

Para o projeto da estação de efeitos foram considerados os seguintes requisitos funcionais:

1. Efeitos de áudio
 - Overdrive;
 - Delay;
 - Equalizador;
 - Possibilidade de expandir os efeitos com uma simples atualização de firmware.
2. Configuração
 - Deve permitir o ajuste dos parâmetros dos efeitos (tempo do delay, ganho do overdrive, controle de graves, agudos, etc.).
3. Interface do usuário
 - Uso de rotary encoders para selecionar o efeito e o valor dos parâmetros.
 - Uso de display LCD, para acompanhamento do ajuste dos parâmetros.
4. Conectividade
 - Entrada e saída de áudio com conectores no padrão P10.

Requisitos não funcionais:

1. Complexidade de uso.

- Interface intuitiva para guitarristas principiantes.

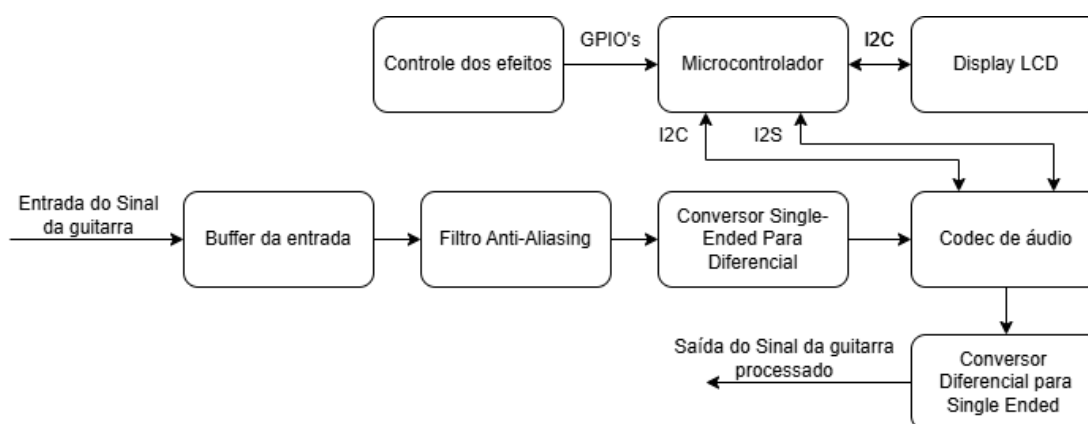
2. Custo

- O preço final não pode ser muito elevado, levando em conta que vai ser um produto de entrada.

3.2 Arquitetura do Hardware

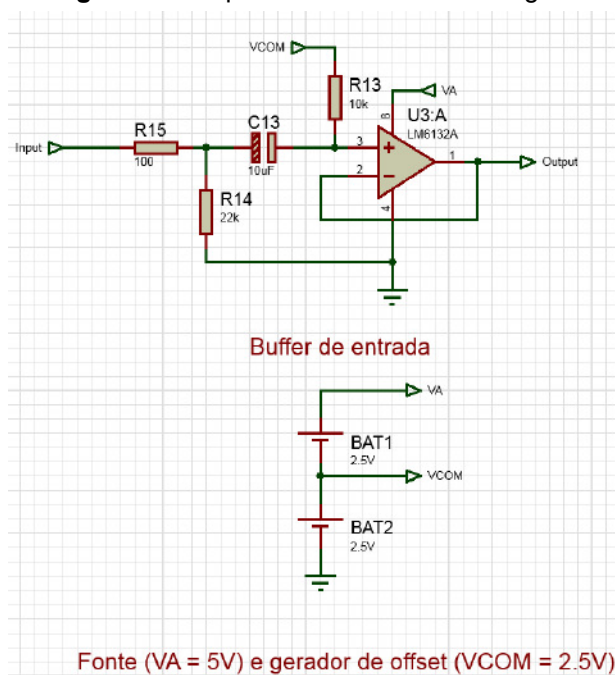
A construção da arquitetura de hardware foi dividida em oito blocos, mostrados na Figura 14.

Figura 14: Diagrama de blocos do projeto Harmony Wave



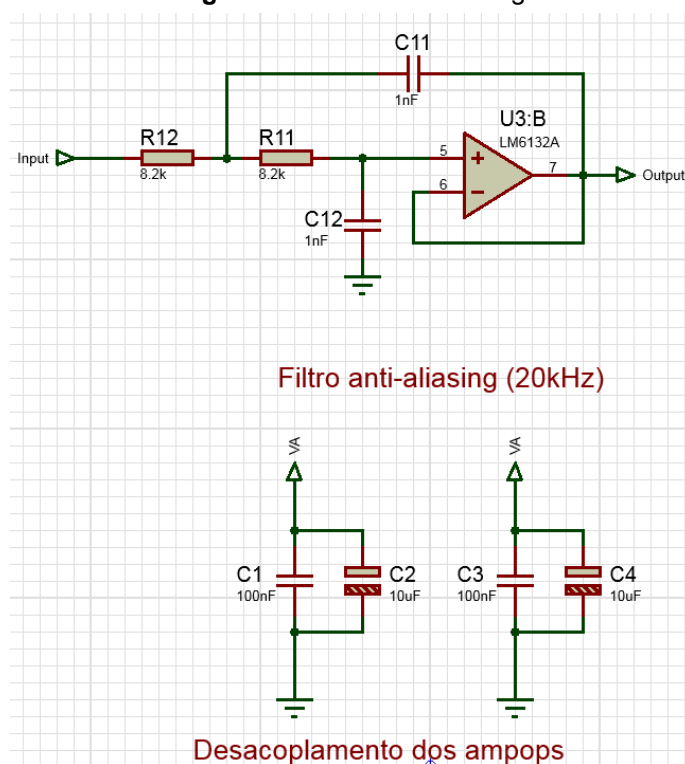
Fonte: Autoria Própria

O primeiro bloco do circuito após a captura do sinal da guitarra via cabo P10 é um buffer analógico de entrada, fornecendo uma alta impedância de entrada e uma baixa impedância de saída, assim isolando a etapa anterior do circuito da próxima etapa. No buffer também é gerado um offset de 2,5 Volts que é metade da tensão de alimentação do subsistema analógico do projeto. Como o sinal da guitarra é um sinal alternado, foi gerado esse offset para retirar as componentes negativas, pois no microcontrolador não podemos ler tensões abaixo de 0 Volts. O sinal de offset, chamado de VCOM, é gerado a partir do bloco CODEC de áudio, que será explicado nas seções adiante.

Figura 15: Esquemático do buffer analógico

Fonte: Autoria própria

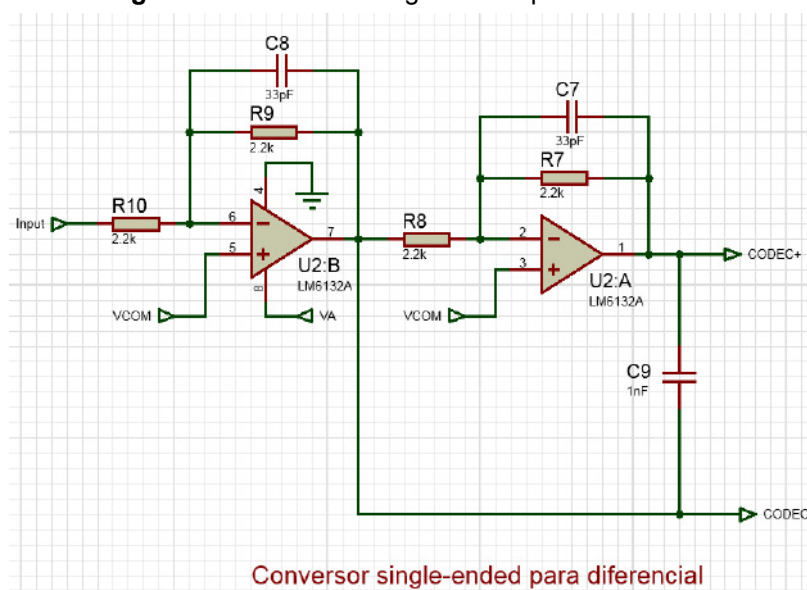
Logo em seguida após a saída do buffer, visto na Figura 15, o circuito possui um filtro anti-aliasing que é um filtro passa baixas que irá atenuar sinais acima de 20 kHz, esse filtro é muito importante pois como esta é a frequência máxima audível pelo ser humano, não é desejado que frequências acima do valor citado sejam recebidas pelo sistema. Também existe o fato de que, mais à frente do projeto, foi feita a seleção da taxa de amostragem. Para áudio, são comuns as taxas de 44100 Hz, 48000 Hz, 96000 Hz e 192000 Hz, no projeto Harmony Wave foi usada a taxa de 48 kHz, respeitando o teorema de Nyquist, para que não haja o fenômeno de aliasing e filtrando sinais acima de 24 kHz no sistema. Na figura 15 temos o esquemático do filtro.

Figura 16: Filtro Anti-Aliasing

Fonte: Autoria própria

Ainda falando de filtragem, na Figura 16 nota-se também alguns capacitores de desacoplamento que serão ligados o mais próximo possível dos CI's dos Amplificadores Operacionais, esses capacitores irão servir como mais uma camada de filtragem. Em caso de interferências eletromagnéticas ou ruídos de alta frequência na alimentação do circuito, será feita a filtragem desses sinais indesejados que podem ser amplificados, garantindo assim estabilidade e evitando oscilações.

Na saída do filtro anti-aliasing será ligado o próximo bloco, o conversor de sinal *single-ended* para diferencial. O intuito desse bloco é preparar o sinal para ser processado no CODEC de áudio. O uso do sinal diferencial é comumente utilizado no ramo sonoro. Ao trabalhar com dois canais, o sinal original e o sinal invertido, caso ocorra a captura de algum ruído nesses canais, no final do circuito costuma-se inverter novamente um dos canais, assim o sinal ruidoso que adentrou nos dois canais vai ser somado e como a fase foi invertida ele vai ser cancelado e o sinal original amplificado. Na figura 17 temos o conversor.

Figura 17: Conversor single-ended para diferencial.

Fonte: Autoria própria.

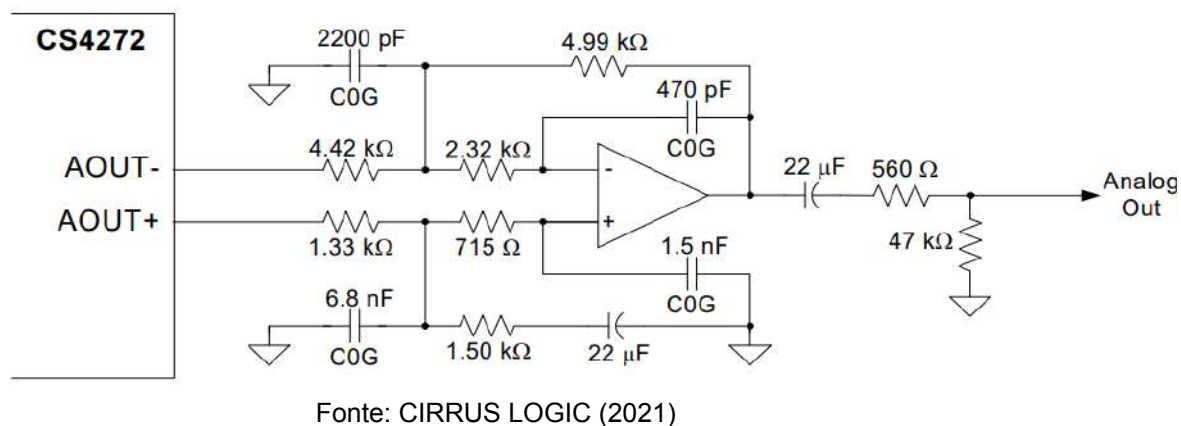
No bloco do CODEC será feita a conversão analógico-digital do sinal recebido dos blocos anteriores. A comunicação do CODEC com o microcontrolador ocorre via protocolo I2S para recebimento e envio do sinal de áudio amostrado e a conversão digital-analógico para podermos reproduzir o áudio processado em algum sistema com alto falantes. Também é utilizado o protocolo I2C para configuração dos parâmetros do CODEC (Taxa de amostragem, Volume de saída, funcionamento mestre/escravo e etc.). No projeto o CODEC foi utilizado no modo escravo, tanto para o protocolo I2S quanto para o protocolo I2C, recebendo os sinais de clock (MCLK, LRCK, SCLK) diretamente do microcontrolador.

No microcontrolador será feito o processamento do áudio e aplicação dos efeitos sonoros, como distorção (overdrive) e delay. Nele, também é feito o processamento das ações do usuário via interface de controle e será mostrado as configurações dos efeitos no display LCD. Após o processamento das amostras recebidas, o microcontrolador devolve as amostras processadas para o CODEC, que executará o processo de conversão digital-analógica e enviará o sinal convertido para o último bloco do fluxo, que converterá os sinais diferenciais gerados de volta para single-ended.

No conversor diferencial para single-ended foi usado o circuito recomendado pelo próprio datasheet do CODEC utilizado no projeto, como pode ser visto na figura

18. Após passar por este bloco, o sinal está pronto para ser reproduzido no amplificador, bastando plugar um cabo P10 na saída do circuito.

Figura 18: Conversor diferencial para single-ended



3.3 Componentes do circuito

Nesta seção serão descritos os componentes utilizados para desenvolvimento do circuito do projeto, suas características e utilidades dentro do objetivo de processamento de áudio.

O amplificador operacional teve papel crucial para desenvolvimento da parte analógica do projeto pois é um componente muito versátil que tem diversas aplicações, e como já foi visto na seção anterior, ele foi usado em vários blocos, além dos filtros ativos, buffers e conversores, dependendo da configuração dos valores dos resistores utilizados, podemos obter algum ganho no sinal, que pode ser explorado, considerando que o sinal da guitarra é da ordem de miliVolts e costuma ser pré-amplificado antes de ser processado em algum DSP. O modelo utilizado para o projeto foi o LMV358 que é um CI de oito terminais, com dois amplificadores operacionais integrados, com saídas Rail-to-Rail, ou seja, ele consegue chegar o mais próximo possível dos limites de alimentação na saída sem saturar o sinal. No caso do projeto Harmony Wave temos uma alimentação de 5V.

O CODEC de áudio responsável pelas conversões ADC e DAC dos sinais, também foi um componente essencial para a construção do protótipo, ter um

componente otimizado para trabalhar com áudio permite que o projetista possa se preocupar menos com ruídos e distorções indesejadas e garanta um som com alta fidelidade. O CODEC escolhido para o projeto foi o CS4272 da empresa Cirrus Logic, um CI Stereo⁵ de vinte oito pinos. Ele trabalha com taxas de amostragem de 44,1 kHz até 192 kHz e com amostras de até 24 bits, também possui controles de volume e mute.

O microcontrolador escolhido foi o STM32H723VGT6 que opera em até 550 MHz de clock, possui 564 KB de RAM, unidade dedicada para operações com ponto flutuante e também possui instruções de DSP. É um excelente chip para trabalhar com processamento digital de sinais. Também possui recursos de comunicação importantes como 5 interfaces I2C, 4 interfaces I2S e um dos recursos que foi amplamente utilizado que foi o de acesso direto à memória sem que os dados necessitem passar pela CPU, garantindo mais velocidade no recebimento dos dados.

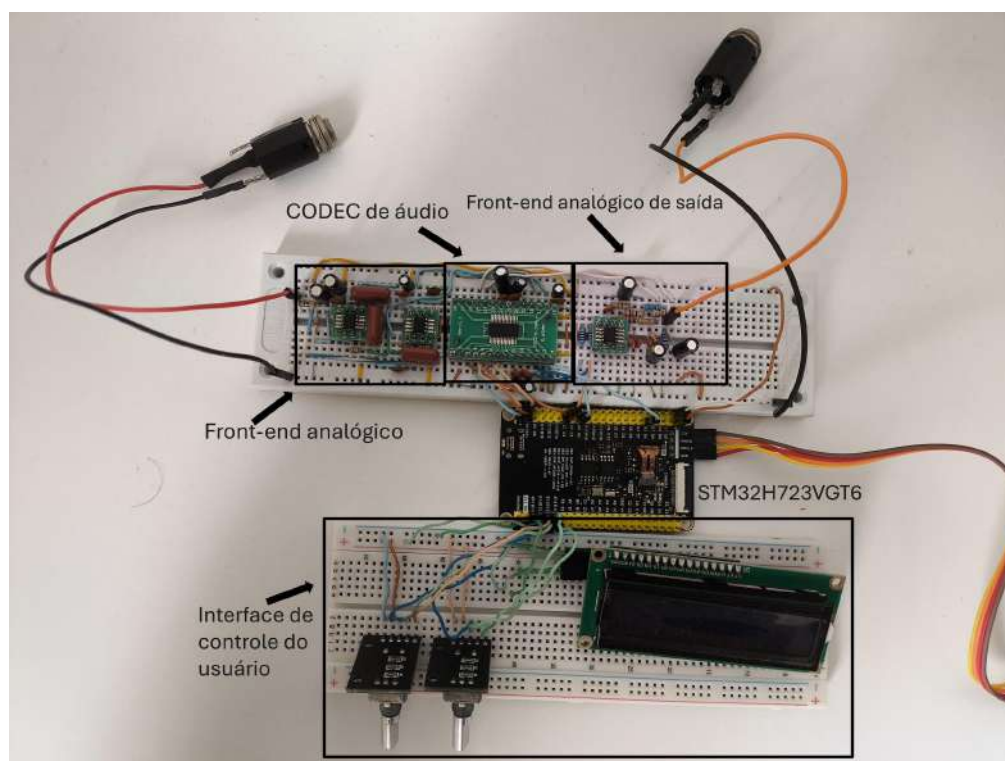
Para interface de usuário foram utilizados dois rotary encoders para seleção dos efeitos e valores dos parâmetros, como os dois possuem switches que funcionam como botões ao serem pressionados, esse click foi utilizado para seleção de efeitos, ao clicar no da direita seleciona o próximo efeito, ao clicar no da esquerda seleciona o efeito anterior, ao girar o encoder da esquerda é selecionado qual parâmetro do efeito o usuário deseja alterar, e ao girar o encoder da direita o valor do parâmetro é ajustado. Além dos encoders o outro componente da interface de usuário é um display LCD HD44780 de 16 colunas por 2 linhas. O display foi utilizado juntamente com um adaptador I2C para que os efeitos selecionados e os valores dos parâmetros sejam mostrados ao usuário em tempo real.

Por fim foram utilizados resistores para controle de corrente no circuito e projeto dos filtros e alguns capacitores para acoplamento e desacoplamento dos sinais. Os capacitores e resistores também foram utilizados como parte integrante dos blocos constituídos pelos AMPOPS.

⁵ Apesar do CI ser Stereo não foi utilizado os dois canais, pois a captura do sinal da guitarra é Mono, no projeto só foi utilizado o canal diferencial A.

Na figura 19 temos o protótipo finalizado e montado na protoboard, onde o plug P10 de fio vermelho é a entrada da guitarra e o plug P10 com o fio laranja é a saída que vai ser ligada no amplificador.

Figura 19: Protótipo do projeto finalizado.



Fonte: Autoria própria

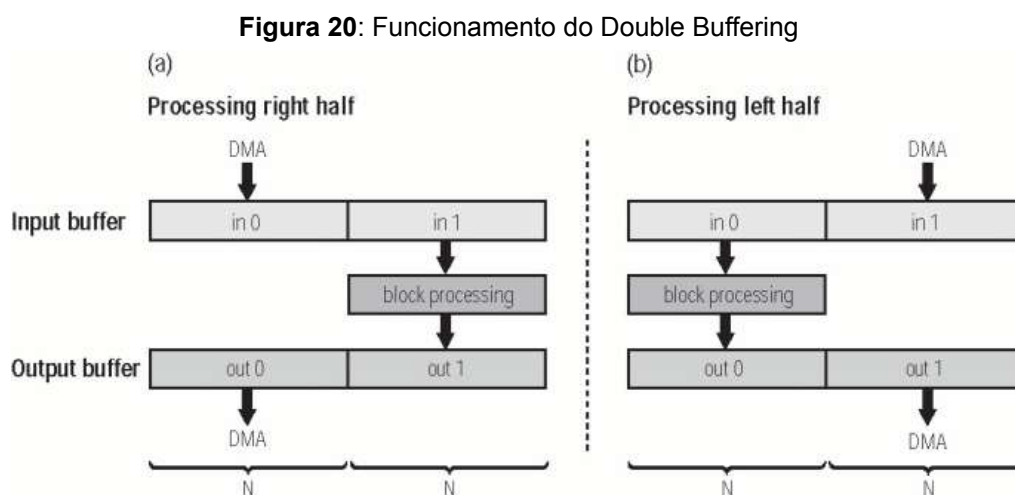
3.4 Processamento do áudio e estratégia de double buffering

Nesta seção será apresentado como foi feito o processamento do áudio recebido no STM32, após a conversão AD do CODEC e envio das amostras pelo protocolo I2S.

Para recebimento e envio das amostras de áudio sem latência perceptível foi utilizado o recurso do DMA no periférico I2S, que permite que os dados sejam repassados da memória direto para o periférico sem passar pela CPU, garantindo velocidade e evitando carga no processador. Também foi configurado no DMA o modo de buffer circular, ou seja, após preencher a última posição do buffer ele volta a preencher os dados no início. As amostras de áudio foram configuradas para

terem 16 bits, que é um tamanho amplamente usado para áudio e oferece um equilíbrio entre qualidade do áudio e complexidade de processamento.

Ao invés de ser adotado uma estratégia de processamento a cada amostra recebida que provocaria sobrecarga sobre o processador, no projeto foi utilizado a estratégia de double buffering que é referenciada por Katz, Lukasiak e Gentile (2007). É uma estratégia de processamento em bloco que consiste em dois buffers circulares, sendo um de entrada e um de saída dos dados, esses buffers serão divididos em duas metades e enquanto o DMA está preenchendo uma das metades dos arrays de entrada e saída de dados, a outra metade está sendo processada, sendo aplicado os filtros e demais operações de DSP. A estratégia também é conhecida como ping-pong buffering. Na figura 20, tem-se a visualização do que foi explicado.



Fonte: Katz et al. (2007)

3.5 Implementação dos efeitos de áudio

Para o protótipo do projeto, foram implementados três efeitos (overdrive, delay e equalizador), e nesta seção será explicado como foram implementados, como cada efeito modifica o sinal e os parâmetros que foram desenvolvidos para cada efeito.

3.5.1 Overdrive

Na implementação do overdrive em tempo discreto é utilizado um processo chamado de não-linearidade estática, que é um tipo de não linearidade que ocorre quando a saída de um sistema depende apenas da entrada em um determinado instante, sem levar em conta o histórico do sinal, ou seja, não é necessário a implementação de algum tipo de memória. O efeito em si funciona achatando os picos do sinal, a função utilizada para fazer esse recorte suave na onda pode ser observada a seguir na figura 21 onde x é uma amostra de áudio de 16 bits que será processada pelo STM32.

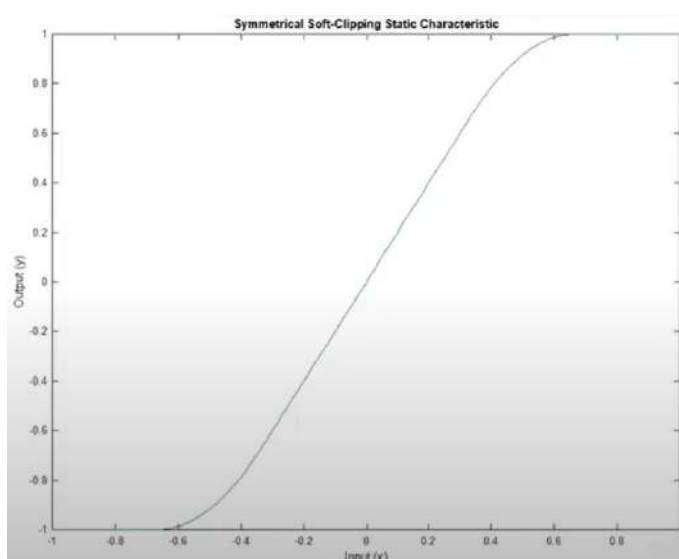
Figura 21: Função para implementação do overdrive em tempo discreto

$$f(x) = \begin{cases} 2x & \text{for } 0 \leq x \leq 1/3 \\ \frac{3-(2-3x)^2}{3} & \text{for } 1/3 \leq x \leq 2/3 \\ 1 & \text{for } 2/3 \leq x \leq 1. \end{cases}$$

Fonte: Zolzer (2011)

Ao aplicar a função acima a um sinal obtém-se o gráfico da figura 22 e nota-se que o sinal começa a ser achatado ao chegar nas regiões do pico superior e inferior. Como o efeito é simétrico, essas entradas negativas serão processadas como valor absoluto e depois convertidas para valor negativo novamente na saída do filtro.

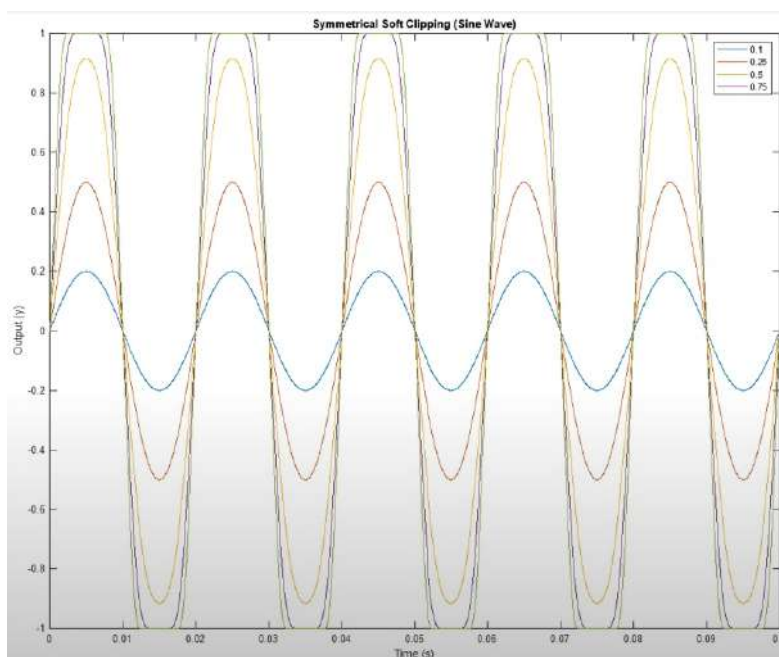
Figura 22: Gráfico que representa o efeito do overdrive



Fonte: Phil's Lab (2023)

A medida que o sinal é distorcido novas harmônicas irão aparecer no sinal e a medida que aumentamos o ganho, que é um dos parâmetros comuns em efeitos de overdrive, a saturação tende a ocorrer mais rápido, o número de harmônicos também tende a ser ainda maior, pois como podemos ver na figura 23 a medida que o ganho do sinal aumenta ele se aproxima de uma onda quadrada que possui infinitos harmônicos.

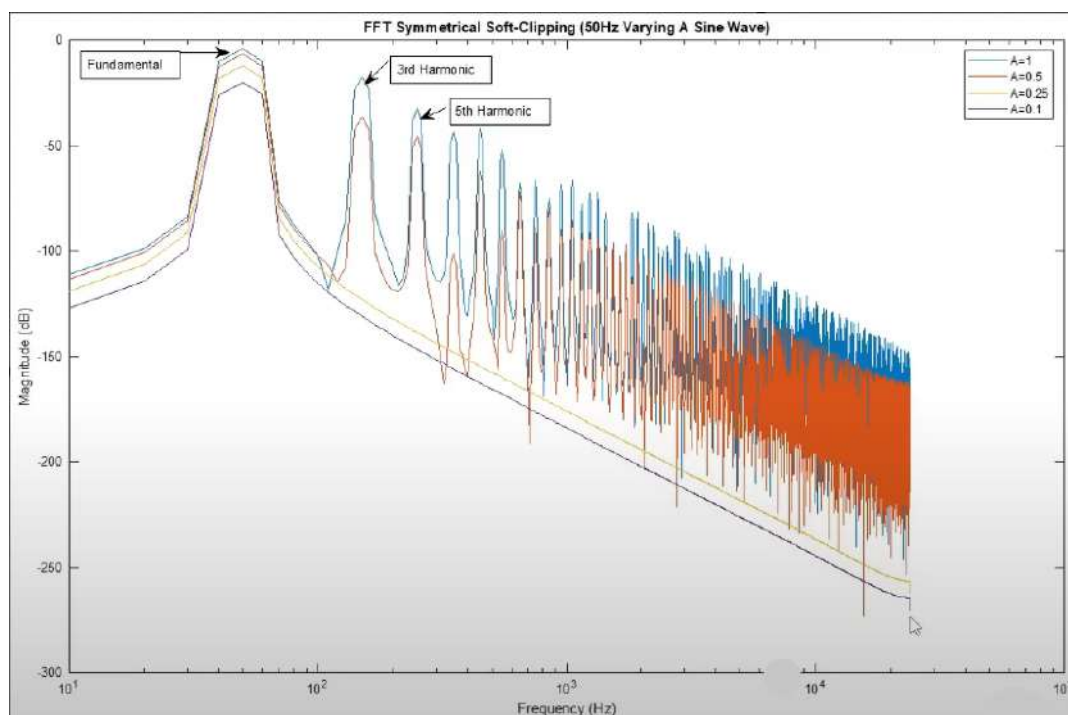
Figura 23: Influência do ganho no formato da onda.



Fonte: Phil's Lab (2023)

Ao fazer a análise do espectro no domínio da frequência é perceptível como a amplitude do sinal que passa pela função do overdrive mostrada na Figura 20 influencia nas suas harmônicas, valores abaixo de $1/3$ simplesmente dobram sua amplitude no domínio do tempo e não tem alteração no domínio da frequência, valores entre $1/3$ e $2/3$ vão gerar harmônicos adicionais e valores acima de $2/3$ vão saturar o sinal de forma rígida, deixando o sinal quadrado gerando infinitos harmônicos que com certeza passarão do limite de Nyquist estabelecidos para processamento de áudio, gerando um efeito conhecido como aliasing distortion, que é um som que não soa muito bem. Na figura 24 é mostrado a FFT de vários sinais de amplitude diferentes que passarão pela função que descreve o overdrive.

Figura 24: FFT de senóides que passarão pelo overdrive.



Fonte: Phil's Lab (2023)

Para resolver o problema do aliasing distortion foi implementado um filtro anti-aliasing FIR utilizando a ferramenta TFilter⁶, que é uma ferramenta web acessível pelo navegador. Para o filtro passa-baixas, foi necessário decidir qual seria a frequência de corte e a escolha foi baseada na função quadrática do overdrive, o filtro não vai eliminar todo o efeito do aliasing mas vai suprimir consideravelmente de forma que ele não incomode o ouvido. Analisando a identidade trigonométrica da equação (8) nota-se que ao elevar uma senóide ao quadrado, a frequência do sinal vai dobrar na função cosseno, a partir disto foi decidido que a frequência de corte para o filtro deve ser 1/4 da frequência de amostragem, como foi utilizado 48 kHz de amostragem a frequência de corte ficou em 12 kHz, pois ao dobrar a frequência, após passar pela seção do overdrive o sinal ainda ficará dentro do limite da amostragem respeitando o teorema de nyquist. A escolha do filtro FIR foi baseada na sua estabilidade e performance que é melhor no domínio da frequência, os filtros FIR são computacionalmente mais caros pois envolvem muitos coeficientes, porém é algo que o microcontrolador STM32H723VGT6 consegue processar sem problemas.

$$\sin^2(\theta) = \frac{1 - \cos^2(\theta)}{2} \quad (8)$$

⁶ Disponível em: <http://t-filter.engineerjs.com/>

Por fim antes da seção de saturação foi implementado um filtro IIR passa altas de primeira ordem, para tirar algumas componentes graves do sinal que tendem a soar turvas, a escolha do filtro IIR foi baseado na sua flexibilidade para poder trocar os coeficientes em tempo real, algo que não é possível nos filtros FIR que para mudar os coeficientes é necessário refazer todo o projeto do filtro, o filtro foi projetado de forma que a frequência de corte possa ser alterada pelos rotary encoders da interface de usuário a qualquer momento.

3.5.2 Delay

Para implementação do segundo efeito, um delay, não foi necessário modificar o sinal recebido na entrada, mas é necessário guardar algumas amostras em um buffer para repeti-las após um tempo que é especificado em um dos parâmetros configuráveis pela interface de usuário. Além da configuração do tempo também são configuráveis os parâmetros de mix, que basicamente ajusta quanto do sinal com delay vai se misturar ao sinal que entra no bloco de processamento, onde os extremos são somente o som de entrada e somente o som com atraso; O outro parâmetro é o feedback que controla quantas repetições o delay vai fazer.

O delay é obtido a partir de uma mudança de fase do sinal de entrada, após receber uma quantidade de amostras, o buffer circular vai encher e o sinal do delay vai ser jogado na saída do sistema, a fórmula para converter o tempo do delay em segundos para número de amostras é mostrada na equação 9:

$$k = T_d * F_s \quad (9)$$

- k é o número de amostras
- T_d é o tempo do delay em segundos
- F_s é a frequência de amostragem

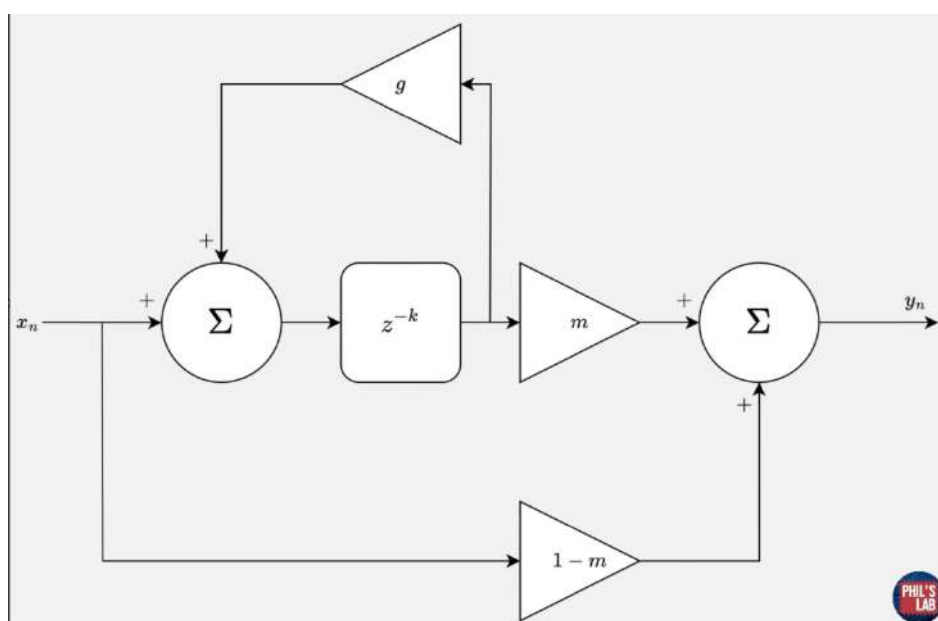
Um cuidado necessário ao implementar um efeito de delay em um microcontrolador é o consumo de memória RAM, em pedais dedicados para o efeito é comum ter algum tipo de memória externa, mas no projeto tratado nesta monografia, foi utilizado somente a memória interna do STM32, como o modelo

usado tem 564 kB de memória, para o buffer circular do efeito foi utilizado um tamanho de 32500 amostras, ao jogar na equação 9 é obtido um tempo máximo de aproximadamente 677 milissegundos que já dá para utilizar de forma satisfatória em muitos estilos musicais.

$$T_d = \frac{32500}{48000} \approx 0,677$$

Durante o processamento do áudio em que a amostra é recebida em um formato inteiro de 16 bits, há uma conversão para float e uma normalização para garantir que o dado esteja entre os valores de -1 e 1, como uma variável do tipo float consome 32 bits que equivale a 4 bytes, o consumo de memória RAM do buffer foi de 130 kB cerca de 23% da memória RAM disponível no microcontrolador o que é um valor considerável.

Figura 25: Estrutura do filtro IIR que implementa o delay



Fonte: Phil's Lab (2024)

Na figura 25 é mostrada a estrutura do delay implementado e pode-se ver como os parâmetros interagem com o fluxo do sinal de entrada.

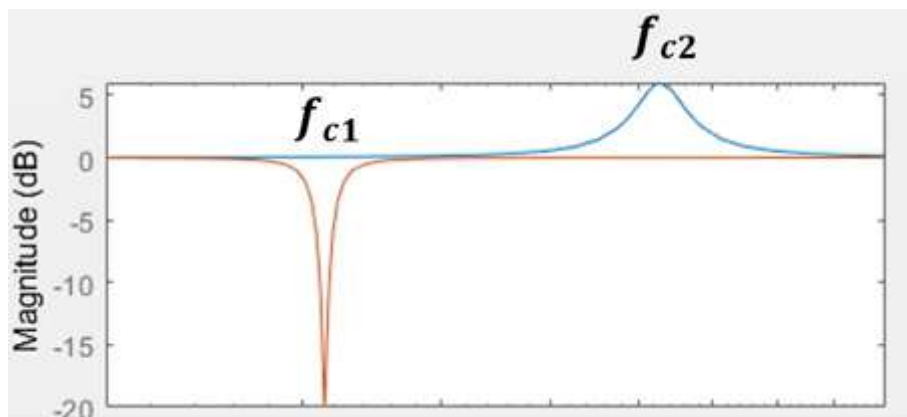
- k no bloco do delay é o que controla o tempo entre cada repetição;

- g é o ganho do feedback que controla quantas repetições serão geradas e a cada repetição o volume vai decaindo;
- m é a variável de mix que controla a mistura do sinal puro e o sinal com delay.

3.5.3 Equalizador de frequências

O último efeito implementado foi um equalizador de três bandas para controle dos graves, médios e agudos que utiliza de uma estrutura chamada filtros de picos que provê um ganho ou uma atenuação do sinal em uma frequência central específica e com uma largura de banda definida, como longe da frequência central o ganho do sinal é unitário isso significa que mais de um filtro pode ser cascadeados para formar um equalizador de várias frequências conforme visto na Figura 26.

Figura 26: Funcionamento dos filtros de picos.



Fonte: Phil's Lab (2022)

Para projeto do filtro digital foi utilizado o método de tustin que consiste em utilizar uma função de transferência de um sinal analógico de tempo contínuo para um sinal digital de tempo discreto. A equação 10 modela o funcionamento do filtro de picos da seguinte forma:

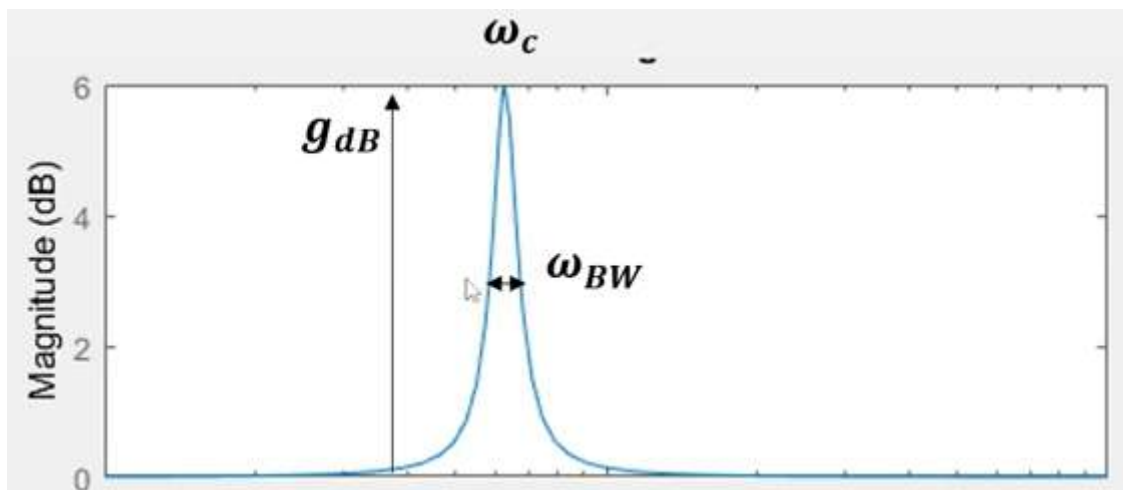
$$H(s) = \frac{\left(\frac{s}{\omega_c}\right)^2 + \left(\frac{g}{Q}\right) \cdot \left(\frac{s}{\omega_c}\right) + 1}{\left(\frac{s}{\omega_c}\right)^2 + \left(\frac{1}{Q}\right) \cdot \left(\frac{s}{\omega_c}\right) + 1} \quad (10)$$

- **g** é o ganho
 - $g = 1$, passa tudo;

- $0 < g < 1$, atenuação na frequência central;
- $g > 1$, ganho na frequência central.
- **wc** é a frequência central do filtro em rad/s;
 - $wc = 2\pi * fc$ (fc é a frequência em Hz)
- **Q** é o fator de qualidade que controla a largura de banda onde o filtro vai atuar.
 - $Q = wc/wBW = fc/fBW$

Na figura 27 podemos ver como as variáveis interagem com o sinal.

Figura 27: Variáveis que modelam o comportamento do filtro.



Fonte: Phil's Lab (2022)

Após efetuar o método do Tustin da transformada bilinear, deve-se converter a transformada Z para uma equação de diferenças, que resultou na equação 11, um filtro IIR de 2º ordem:

$$y_n = \frac{1}{b_0} \left((a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2}) + (b_1 y_{n-1} + b_2 y_{n-2}) \right) \quad (11)$$

Os coeficientes a e b são obtidos da seguinte forma:

$$a_0 = 4 + 2 \cdot \frac{g}{Q} \cdot \omega_c \cdot T + (\omega_c \cdot T)^2$$

$$a_1 = 2 \cdot (\omega_c \cdot T)^2 - 8$$

$$a_2 = 4 - 2 \cdot \frac{g}{Q} \cdot \omega_c \cdot T + (\omega_c \cdot T)^2$$

$$b_0 = 4 + 2 \cdot \frac{1}{Q} \cdot \omega_c \cdot T + (\omega_c \cdot T)^2$$

$$b_1 = -(2 \cdot (\omega_c \cdot T)^2 - 8)$$

$$b_2 = -(4 - 2 \cdot \frac{1}{Q} \cdot \omega_c \cdot T + (\omega_c \cdot T)^2)$$

Por fim, como o método de Tustin ocasiona uma certa distorção no sinal ao discretizar, é necessário fazer uma correção na frequência central antes de calcular o fator de qualidade **Q**. A equação (12) mostra a correção necessária.

$$\omega_{cd} = \frac{2}{T} \cdot \tan(\omega_c \cdot \frac{T}{2}) \quad (12)$$

Na implementação deste projeto, foram cascadeados três filtros de pico com frequências variadas para controle dos graves, médios e agudos:

- Filtro dos graves:
 - Frequência central = 200Hz;
 - Largura de banda = 100 Hz.
- Filtros dos médios:
 - Frequência central = 1000Hz;
 - Largura de banda = 1000 Hz.
- Filtro dos agudos:
 - Frequência central = 4000 Hz;
 - Largura de banda = 2000 Hz.

Para controle dos parâmetros dos efeitos foram respeitados os seguintes limites:

- Ganho da distorção - De 0 até 50;
- Frequência de corte dos graves - De 20 Hz até 250 Hz;
- Mix do delay - De 0(somente entrada) até 1(somente som com delay);
- Tempo do delay - De 0 até 677 milissegundos;
- Ganho do feedback - De -1 até 1;
- Ganho do equalizador(graves, médios e agudos) - De 0,1 até 10.

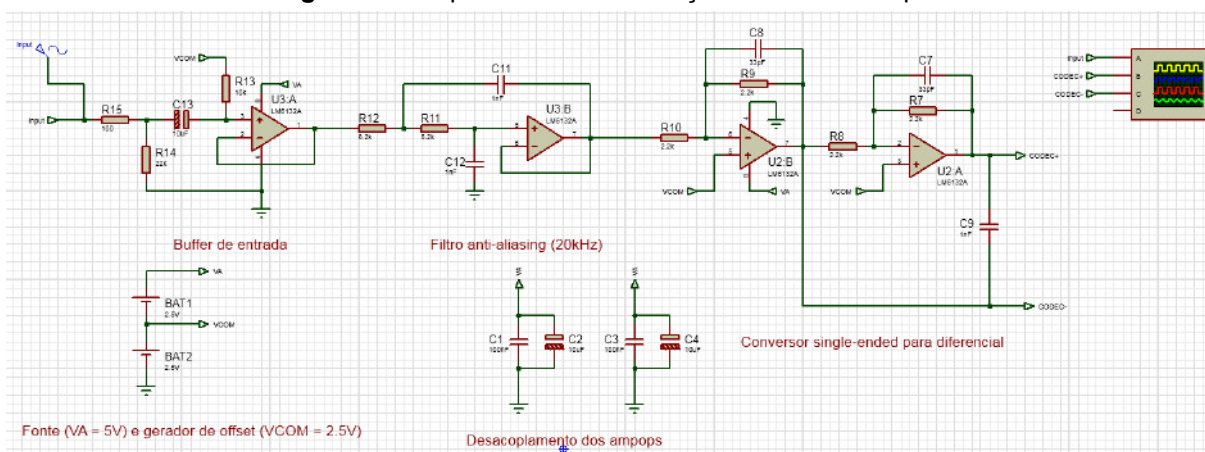
3.6 Teste e Validação

Os testes do protótipo foram realizados desde a simulação da parte analógica em software, utilizando o Proteus até a etapa de conectar a guitarra no protótipo e analisar se os efeitos estão interagindo corretamente com o áudio.

Todos os blocos analógicos foram validados por meio da simulação de injeção de alguns sinais senoidais na entrada do sistema e verificando pelo osciloscópio que está nas ferramentas de Debug se estava ocorrendo o offset DC, a atenuação das senóides acima de 20 kHz e a geração dos sinais diferenciais.

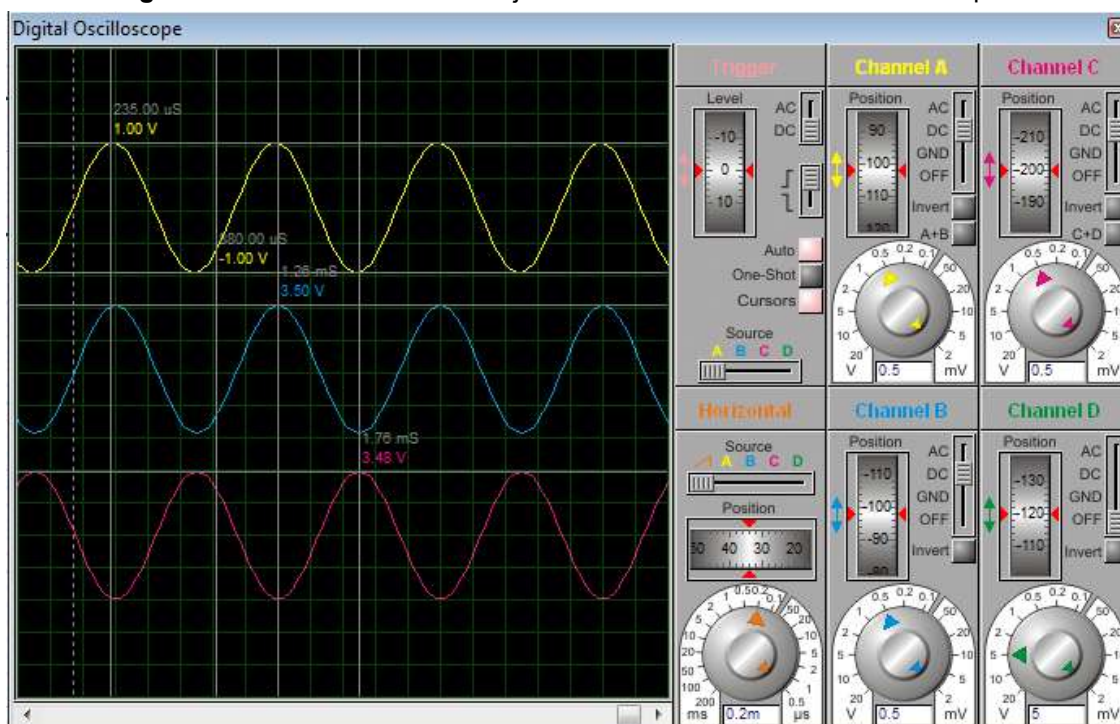
Nas figuras 28, 29 e 30 é possível observar o esquemático do circuito e alguns dos testes efetuados no Proteus, o sinal de cor amarela refere-se ao sinal de entrada, o sinal de cor azul a entrada positiva do sinal diferencial e o sinal de cor vermelha a entrada negativa do sinal diferencial.

Figura 28: Esquemático da simulação com osciloscópio



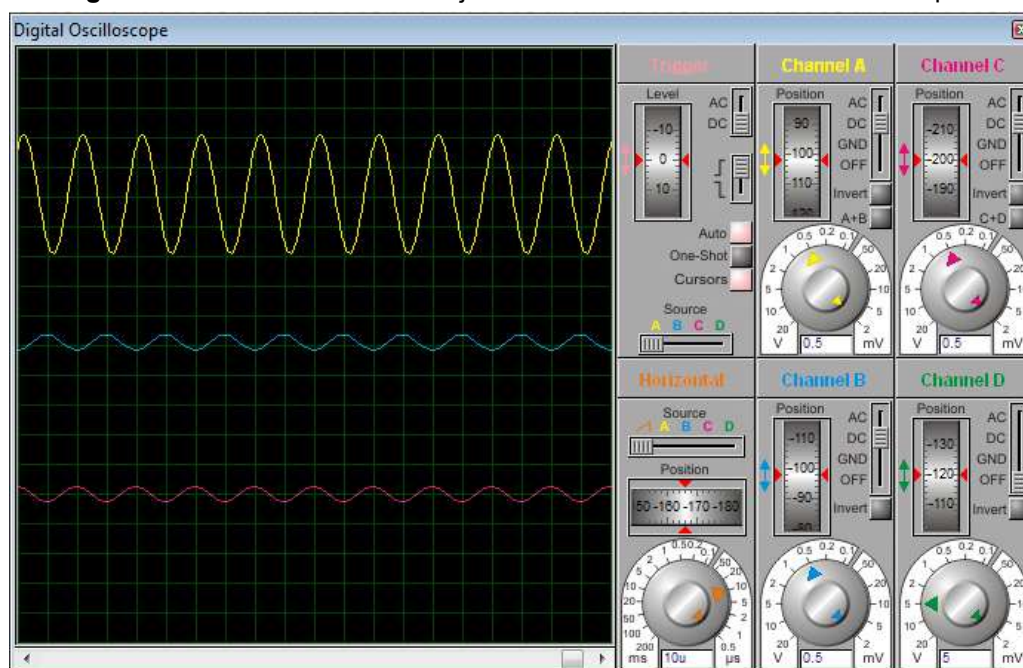
Fonte: Autoria própria

Figura 29: Resultado obtido ao injetar uma senoide de 1kHz com 1V de pico



Fonte: Autoria própria

Figura 30: Resultado obtido ao injetar uma senoide de 50kHz com 1V de pico

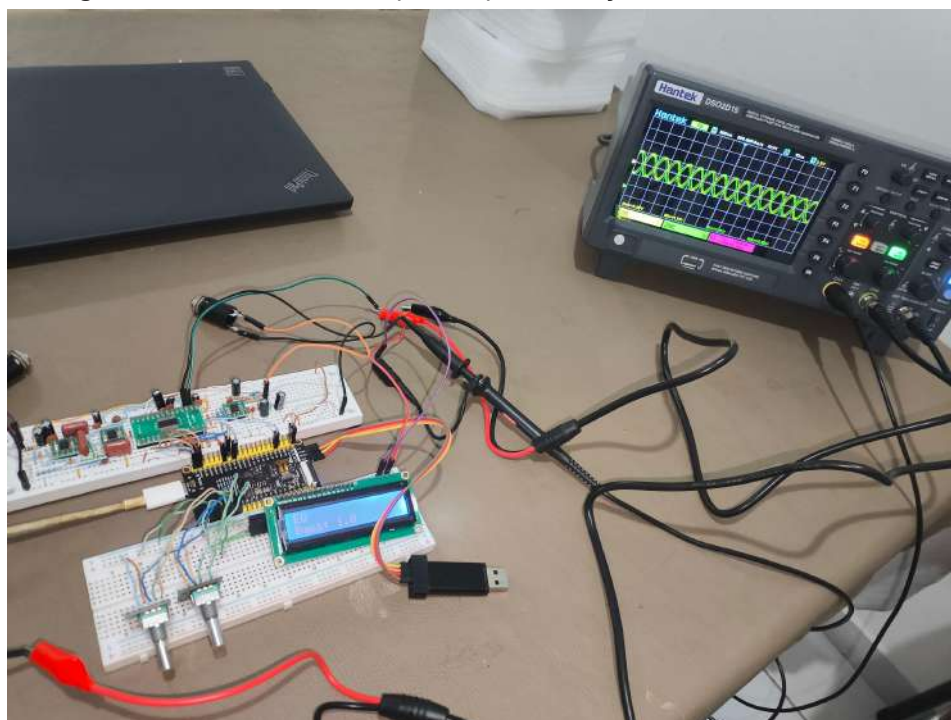


Fonte: Autoria própria

Após obter sucesso na simulação da parte analógica no Proteus, a montagem foi feita no ambiente real em protoboard e testado utilizando os recursos disponibilizados pelos laboratórios do IFPB como osciloscópio e gerador de funções para repetir os mesmos testes efetuados no Proteus. Como o CODEC ainda não estava integrado no sistema, foi utilizado uma pilha AA de 1,5V somente para testar a geração do offset do sinal (VCOM).

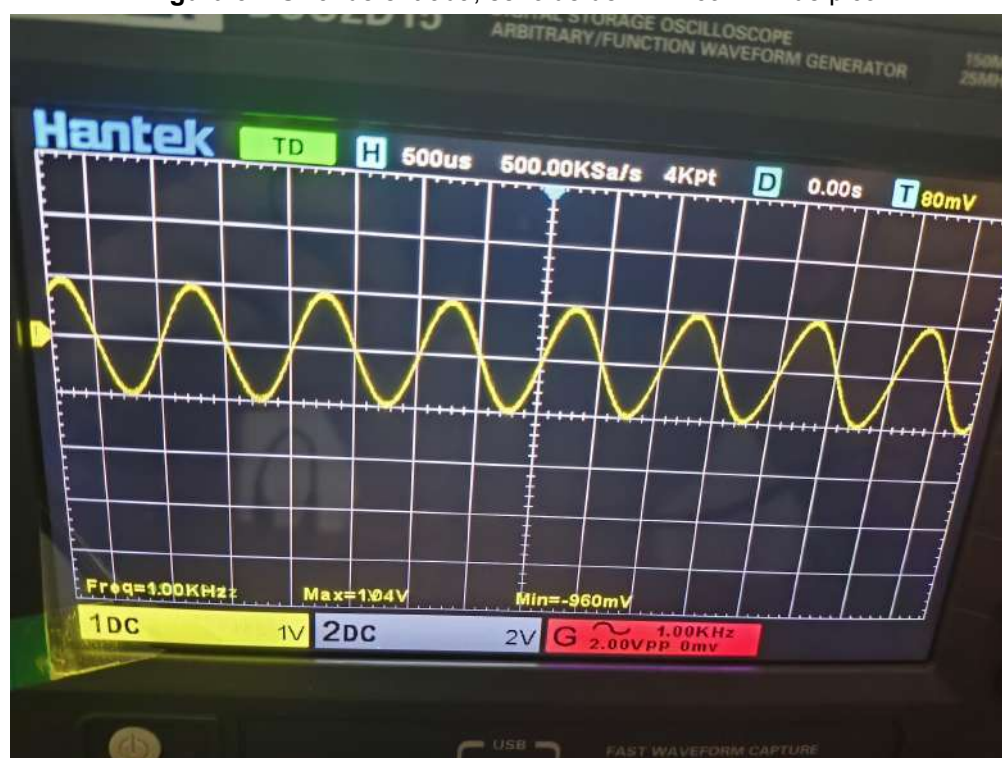
Após toda validação da parte analógica com a pilha AA como recurso de teste temporário, o próximo passo foi ligar o CODEC de áudio no sistema e configurar via I2C utilizando o STM32 assim podendo gerar o sinal de VCOM definitivo, nas figuras 31, 32 e 33 é possível observar o teste da geração do sinal diferencial feitos no circuito real.

Figura 31: Circuito montado para experimentação dos sinais diferenciais



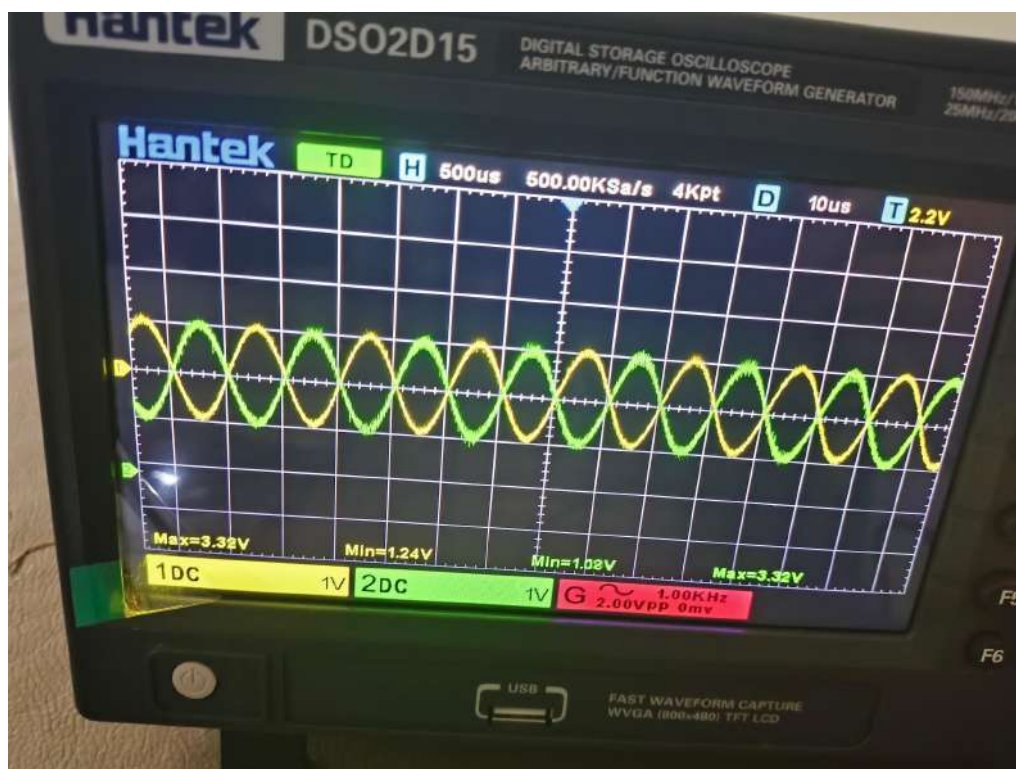
Fonte: Autoria própria

Figura 32: Sinal de entrada, senoide de 1kHz com 1V de pico



Fonte: Autoria própria

Figura 33: Sinal diferencial gerado e validado com sucesso



Fonte: Autoria própria

Na primeira montagem do codec ainda não foi utilizada a interface do I2S. Existe um recurso no CODEC utilizado no projeto chamado de loopback, onde o sinal que é recebido no ADC é jogado direto na saída do DAC, assim já seria possível fazer os testes com a guitarra plugada no sistema e ouvir o áudio. Como ainda não tinha sido implementado o circuito conversor de sinal diferencial para sinal single-ended, o conector P10 de saída foi ligado em uma das saídas diferenciais do CODEC e assim foi possível ouvir o som limpo da guitarra sem nenhum efeito aplicado.

Validado o loopback com sucesso, chegou a hora de fazer o áudio passar pelo protocolo I2S e de fato ser processado pelo STM32, o resto das conexões faltantes foram feitas e o teste na guitarra ainda foi feito utilizando somente um dos canais diferenciais, como o sinal é o mesmo somente com a fase invertida, tanto faz usar um canal como o outro.

Com o I2S funcionando como esperado, o conversor diferencial para single-ended foi implementado e o último teste com o som limpo sem nenhum efeito foi realizado com a guitarra plugada no circuito.

Após a validação do som limpo do instrumento com sucesso, foi implementada a parte dos efeitos. Com os efeitos implementados, os primeiros testes foram feitos com valores hard coded, pois o circuito de interface de usuário ainda não tinha sido implementado, todos os testes dos efeitos também foram realizados com a guitarra plugada no circuito e ouvindo se o som estava satisfatório.

Uma vez que todos os efeitos estavam funcionando corretamente, foi implementada a interface de usuário com os encoders e o display LCD e novamente com a guitarra conectada ao circuito, a medida que os parâmetros iam sendo alterados pela interface de usuário sempre verificando se o som ia respondendo conforme a alteração.

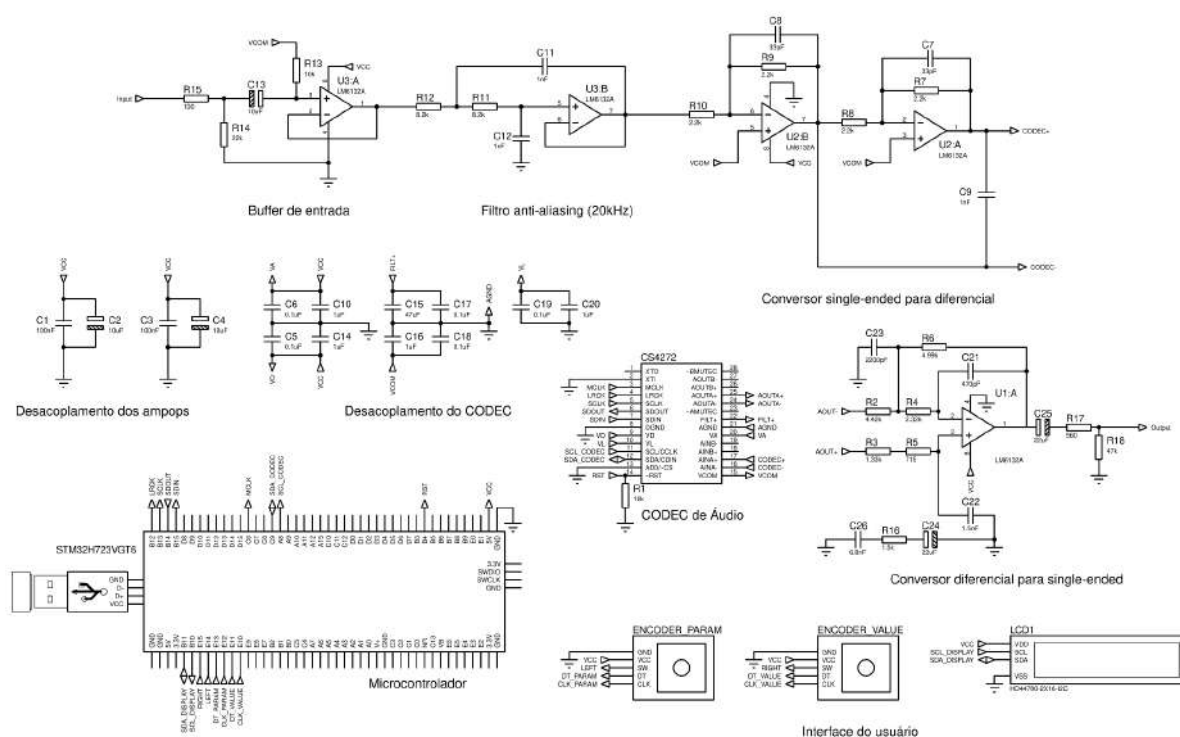
4.0 Resultados

Na seção atual, serão discutidos os resultados obtidos pelo protótipo e se todos os objetivos propostos foram alcançados. Serão abordados três tópicos: O desenho final do circuito, o custo dos componentes do protótipo e por último serão anexados links com as amostras de áudio dos testes feitos com o circuito em funcionamento.

4.1 Esquemático e montagem do circuito.

Para fins de reprodutibilidade da montagem do circuito, o esquemático foi gerado no software Proteus que também foi utilizado para as simulações do front-end analógico, o desenho pode ser observado na figura 34.

Figura 34: Esquemático do projeto Harmony Wave

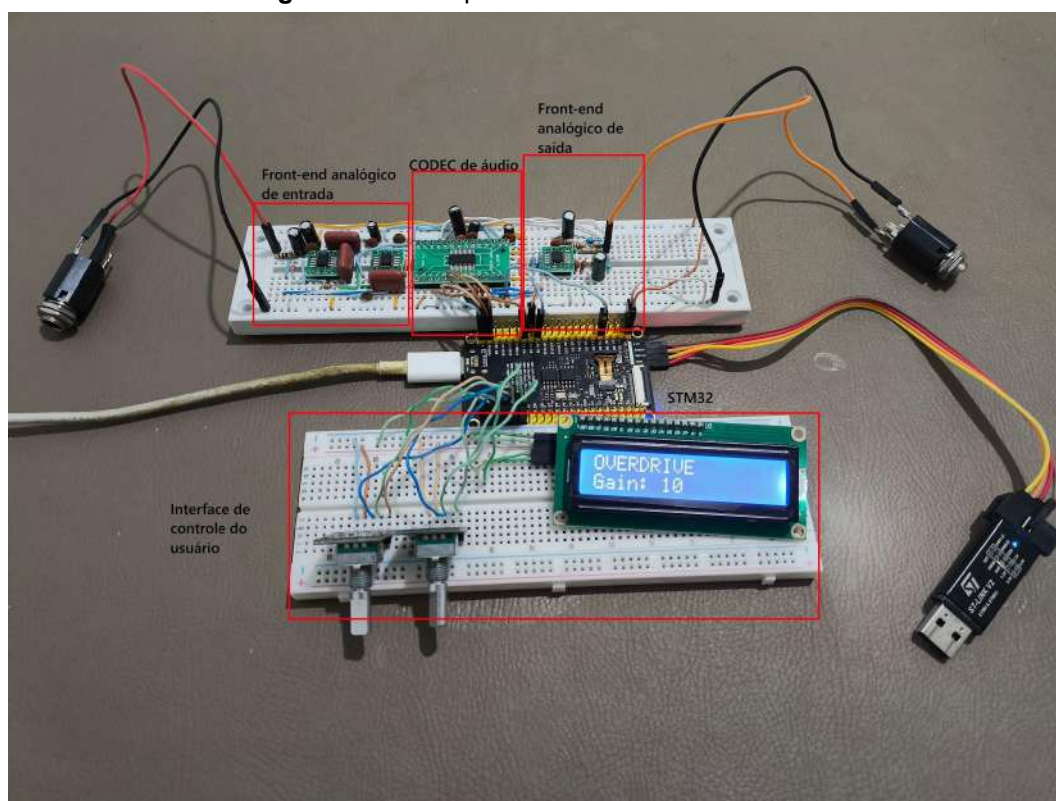


Fonte: Autoria própria

A elaboração do circuito esquemático facilita a identificação de erros na montagem e possibilita a rápida geração de novos protótipos. Além disso, evita a necessidade de consultas constantes aos datasheets dos componentes para a realização das conexões, justificando, assim, a decisão do autor em desenvolver este artefato.

Ao se guiar pelo esquemático, o resultado obtido na montagem em protoboard pode ser observado na figura 35 destacando os blocos citados no esquemático da Figura 34.

Figura 35: Protótipo em funcionamento.



Fonte: Autoria própria

Todos os componentes e fios foram montados em perfil baixo rentes a protoboard, isso facilita a visualização, troca de componentes e organização do circuito, além disso os fios foram cortados bem curtos para evitar interferência e ruído indesejado principalmente nos sinais de alta frequência como os clocks do I2S, que foi algo que ocorreu algumas vezes durante a montagem, e simplesmente ao diminuir o tamanho do fios boa parte dos ruídos sumiram.

4.2 Custo de produção do projeto.

Nesta seção será apresentado o custo da aquisição dos componentes necessários para a construção do projeto Harmony Wave. Grande parte dos componentes foram importados da china com objetivo de deixar a montagem mais barata. Alguns componentes foram adquiridos no comércio local de Campina Grande e também na loja BitMaker, que é um comércio online, mas que a sede fica

na cidade de Campina Grande. Na tabela 1 pode-se observar o custo de cada componente e onde foi adquirido.

Tabela 1: Custo dos componentes do projeto

Componente	Local de aquisição	Custo em R\$
STM32H723VGT6	AliExpress	88,19
CS4272	AliExpress	46,88
LMV358	AliExpress	31,15
ST-LINK V2	AliExpress	12,68
Placas adaptadoras de SMD para PTH	AliExpress	68,98
Resistores e capacitores	Comércio Local	30,00
Jack's P10	Comércio Local	30,00
Rotary Encoders	BitMaker	17,80
Display LCD com adaptador I2C	BitMaker	29,90
Total ⁷		355,58

Fonte: Autoria própria

O preço final se manteve bastante competitivo dados os recursos que o projeto oferece de alta customização, que não é algo comum nas pedaleiras comerciais. Nos dias de hoje, se customiza bastante os impulse responses que são simulações de caixas de amplificadores gravadas com um microfone, mas os efeitos costumam ser pré-configurados não permitindo novas cargas de distorções, delays, compressores, etc.

⁷ Os jumpers não foram considerados, pois foram retirados de um cabo ethernet reciclado, os jumpers dos jacks P10 são de outro tipo mas podem ser trocados pelos de cabo ethernet sem problema algum.

Alguns modelos de pedaleiras comerciais que possuem capacidade de importar os impulse response:

- Zoom G6 e G11;
- Mooer GE150 e GE200, GE300;
- Nux MG-30, MG-300 e MG-400;
- Cuvave Cube Baby;
- Ampero One e II Stomp;
- Line 6 POD Go e Helix;
- Boss GT-1000 e GX-100;
- Headrush MX5, Pedalboard e Prime;
- Fractal FM3 e FM9;
- Kemper Stage;
- Neural DSP Quad Cortex.

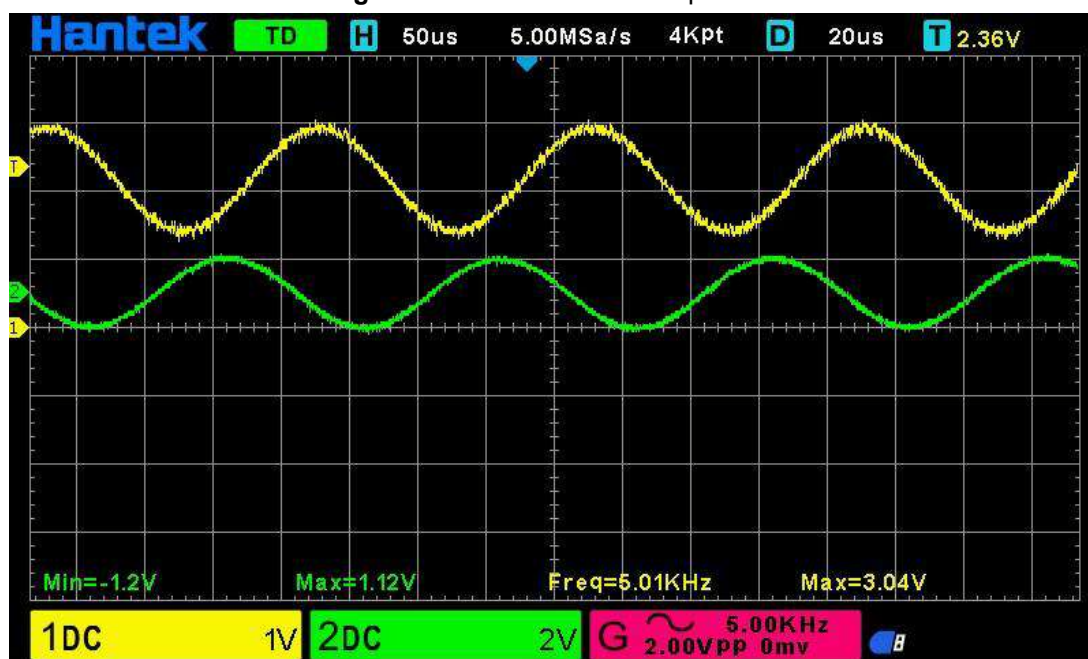
Desses modelos, o único que atualmente é mais barato que o projeto Harmony Wave é o Cuvave Cube Baby que se encontra em uma faixa de R\$ 200,00 a R\$ 300,00.

4.3 Sinais e filtros.

Visando demonstrar o resultado dos filtros, nesta seção serão apresentados alguns sinais capturados em osciloscópio e seu comportamento após passar pelo sistema.

A figura 36 demonstra em verde o sinal de entrada do sistema, uma onda senoidal de 5 kHz com amplitude de 2 V pico a pico e em amarelo o sinal de saída, nenhum efeito está sendo aplicado, é possível notar que o sinal de saída possui a captura de alguns ruídos, possivelmente pela montagem em protoboard sem um plano terra adequado, algo que seria implementado em uma placa de circuito impresso.

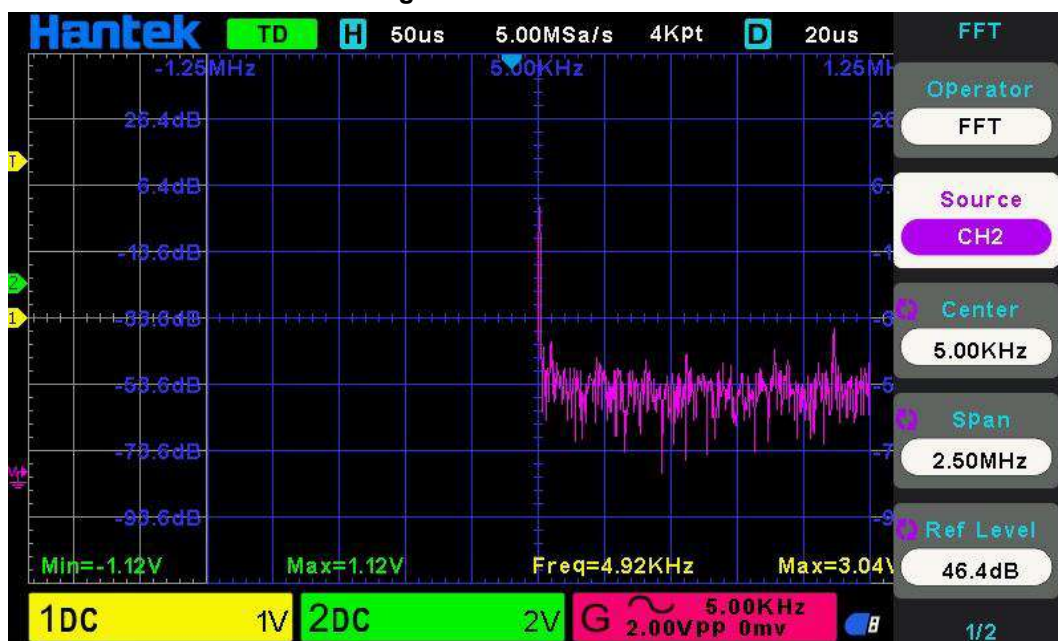
Figura 36: Sinais sem efeito aplicado.



Fonte: Autoria própria.

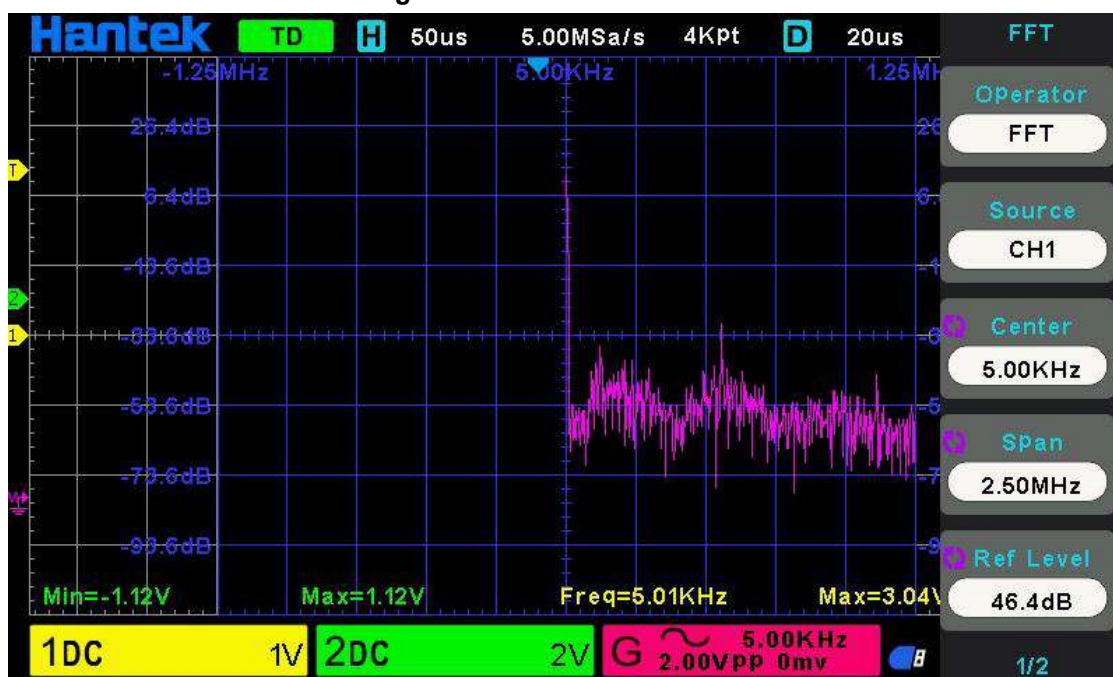
Nas figuras 37 e 38 temos a FFT do sinal de entrada e do sinal de saída respectivamente, onde é possível observar que na saída realmente possui mais componentes de alta frequência geradas pelo ruído.

Figura 37: FFT do sinal de entrada.



Fonte: Autoria própria.

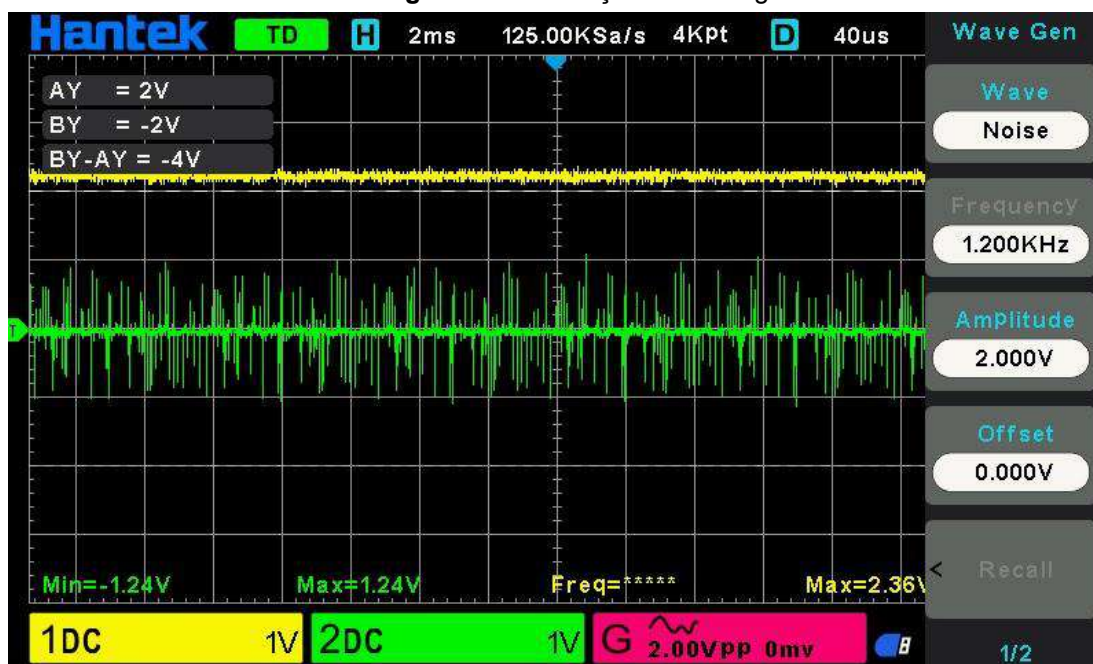
Figura 38: FFT do sinal de saída



Fonte: Autoria própria.

Para demonstrar a atenuação de ruídos mais problemáticos na entrada do sistema foi acoplado um sinal extremamente ruidoso e na saída o sinal foi bastante atenuado como é possível observar na figura 39.

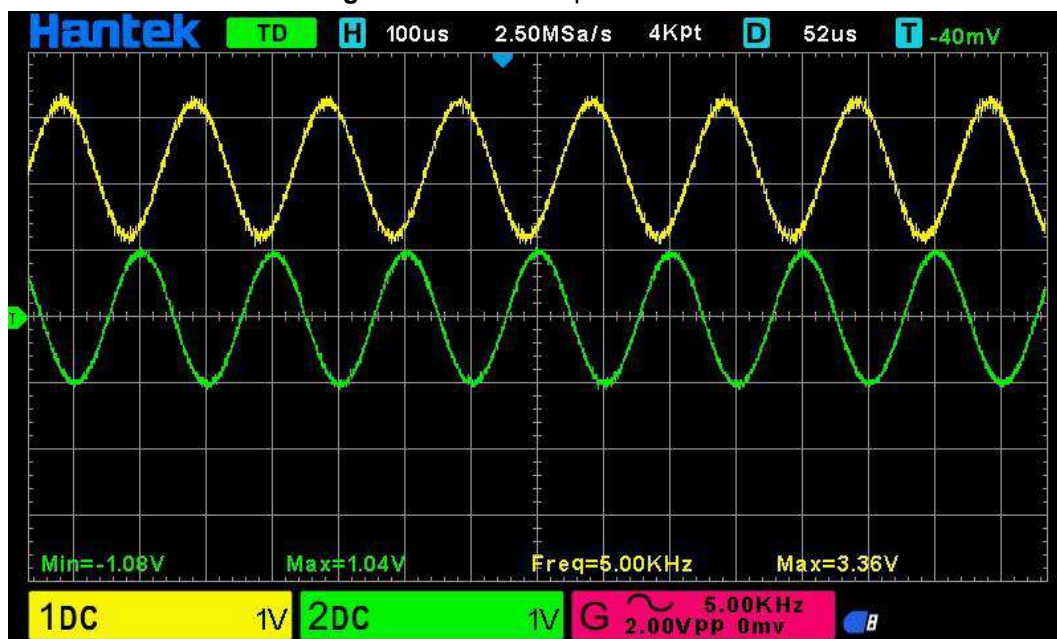
Figura 39: Atenuação de ruído grave.



Fonte: Autoria própria.

Quanto ao atraso decorrente do tempo de processamento no microcontrolador, podemos observar na figura 40 focando no eixo central, que o tempo para que o sinal seja reproduzido na saída é de menos de 100 microssegundos, um tempo muito bom, considerando aplicações de áudio.

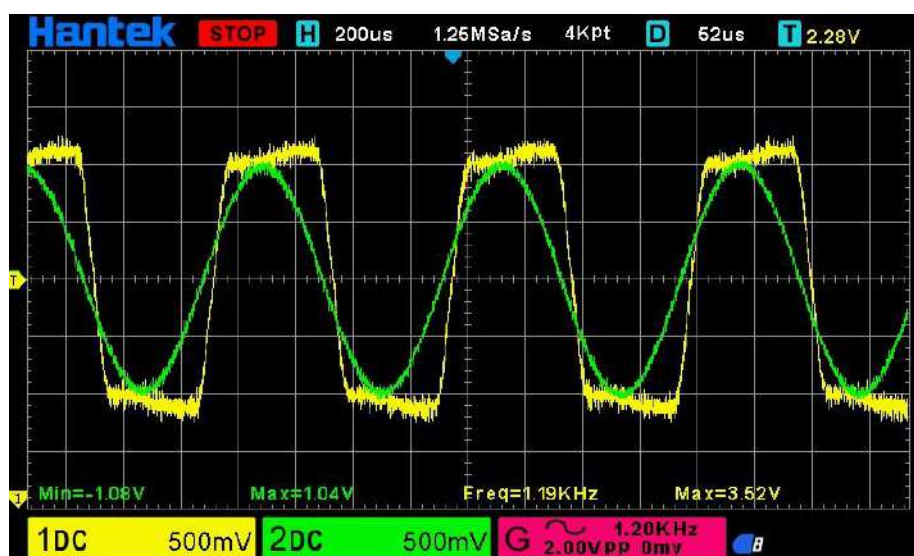
Figura 40: Atraso de processamento



Fonte: Autoria própria

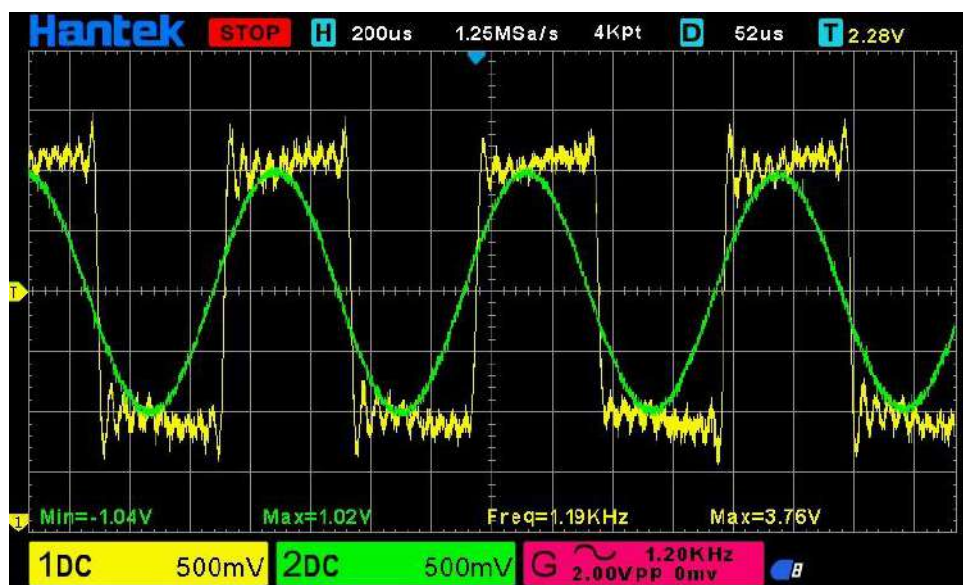
Partindo para os filtros dos efeitos, nas figuras 41 e 42 é demonstrado como o ganho do overdrive influencia no formato da onda, fazendo com que ela vá cada vez mais se aproximando de uma onda quadrada.

Figura 41: Overdrive com ganho 2 sobre a senoide.



Fonte: Autoria própria.

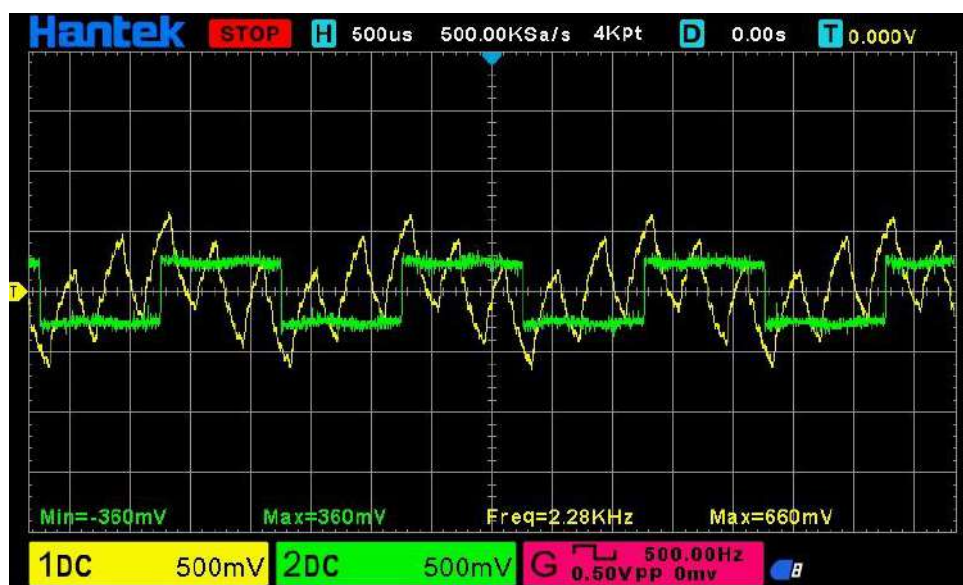
Figura 42: Overdrive com ganho 50 sobre a senoide.



Fonte: Autoria própria.

No delay ao utilizar uma onda quadrada é possível enxergar as repetições do sinal, feedback está no máximo na figura 43.

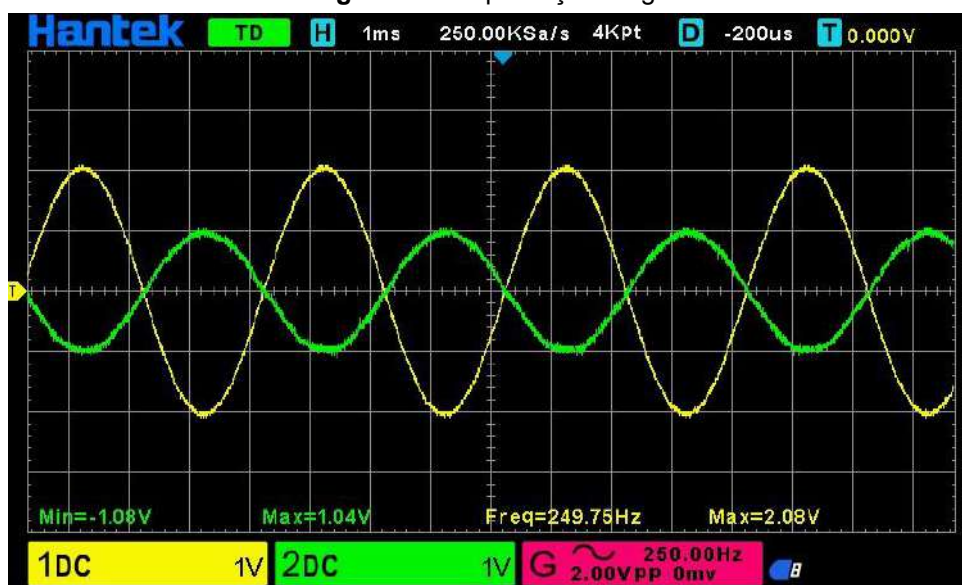
Figura 43: Repetições do delay.



Fonte: Autoria própria.

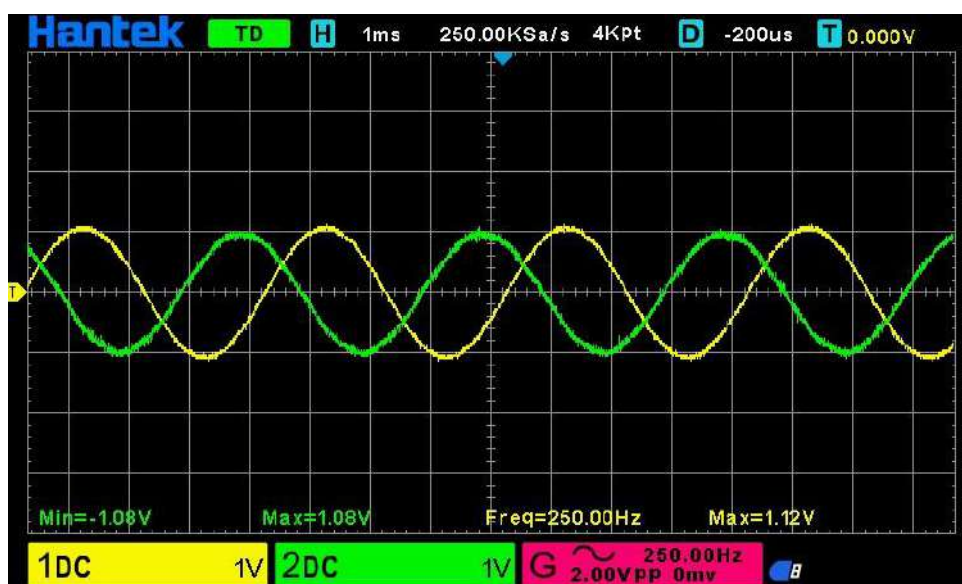
Por fim, nas figuras 44 e 45 é possível observar a amplificação e atenuação do filtro de pico, para demonstrar foi utilizado na frequência de 250 Hz o grave do equalizador projetado.

Figura 44: Amplificação do grave.



Fonte: Autoria própria.

Figura 45: Atenuação do grave.



Fonte: Autoria própria.

4.4 Demonstração sonora.

Nesta seção será demonstrado amostras de áudio que foram gravadas utilizando do Harmony Wave para geração dos efeitos sonoros, além do protótipo foram utilizados os equipamento a seguir: Uma guitarra Cort modelo KX300 etched, amplificador SG-15 da Strinberg e cabos Santo Angelo, que podem ser vistos na figura 46, a gravação do áudio foi feita com o microfone do smartphone na frente do

amplificador. Nenhuma mixagem foi feita no áudio, somente alguns cortes no começo, meio e final dos áudios para remover os intervalos de silêncio.

Para hospedagem do áudio, foi utilizado a plataforma SoundCloud que oferece até 180 minutos de armazenamento de áudio de forma grátis, e facilita o compartilhamento por meio de links personalizados.

Figura 46: Equipamento utilizado para gravar demonstrações.



Fonte: Autoria própria

Abaixo estão disponíveis os links para as amostras de áudio e seus respectivos parâmetros utilizados:

- [Som limpo](#) : Som puro da guitarra sem nenhum efeito ligado.
- [Overdrive com pouco ganho](#):
 - Ganho = 4.
- [Overdrive com muito ganho](#):
 - Ganho = 40.
- [Delay com feedback médio](#):
 - Feedback = 0;
 - Mix = 0,5;

- TimeInMs = 250.
- Delay com feedback alto:
 - Feedback = 0,9;
 - Mix = 0,2;
 - TimeInMs = 250.
- Delay com feedback baixo:
 - Feedback = -0,9;
 - Mix = 0,2;
 - TimeInMs = 250.
- Solo combinando overdrive + delay:
 - Ganho = 20;
 - Mix = 0,2;
 - TimeInMs = 160;
 - Feedback = 0,1.
- Equalização com foco nos graves (som limpo + overdrive):
 - Grave = 10;
 - Médio = 1,3;
 - Agudo = 0,1;
 - Ganho (trecho com overdrive) = 6.
- Equalização com foco nos agudos (som limpo + overdrive):
 - Grave = 0,1;
 - Médio = 0,7;
 - Agudo = 10;
 - Ganho (trecho com overdrive) = 6.
- Equalização customizada (som limpo + overdrive):
 - Grave = 1;
 - Médio = 0,9;
 - Agudo = 5;
 - Ganho (trecho com overdrive) = 6.

5.0 Conclusão

O objetivo deste trabalho foi desenvolver uma estação de efeitos customizável, com baixo custo comparado às opções mais famosas do mercado que oferecem algum tipo de customização. Sendo assim, essa é uma opção viável para iniciantes na guitarra que não querem gastar muito, mas que desejam ter uma gama de efeitos à sua disposição para praticar o instrumento.

Os objetivos foram atingidos, pois foram implementados alguns efeitos que funcionaram conforme o esperado, modificando o som do instrumento. O sistema também é expansível, não sendo necessário modificar o hardware a menos que se queira fazer algum tipo de aprimoramento, mas para implementar novos tipos de efeito somente uma atualização de firmware pode inserir os mais diversos efeitos, o limite sendo somente a memória flash do microcontrolador que armazena o programa e as amostras de áudio. A interface também é bastante simples, tendo somente três tipos de interações: Avançar efeito, Selecionar parâmetro e Selecionar valor do parâmetro, facilitando a vida do iniciante.

Apesar dos objetivos terem sido alcançados, ainda é possível ouvir um pequeno ruído agudo de fundo, muito provavelmente por ser uma montagem em protoboard com vários sinais de alta frequência e não ter um plano de terra adequado que seria algo implementado em uma placa de circuito impresso.

5.1 Trabalhos Futuros

Para implementações futuras, o protótipo está bastante aberto a atualizações. Na parte de hardware, pode-se incluir uma memória RAM externa para possibilitar efeitos de eco com tempos maiores, e a possibilidade de implementar o reverb que consome bastante memória também. Para interface com o usuário, pode-se acrescentar um foot switch para permitir que o músico consiga interagir com o sistema com os pés enquanto toca o instrumento com as mãos, garantindo agilidade na troca das configurações de efeitos. Também seria interessante projetar uma placa de circuito impresso para o circuito para poder fazer um aterramento mais eficiente e conseguir o som mais limpo possível.

Para software, as possibilidades são muitas, desde atualizar e melhorar os efeitos existentes com novos parâmetros até implementar novos efeitos. Uma sugestão interessante seria criar um sistema de patches onde o usuário possa salvar várias configurações de efeitos diferentes e associar a um banco onde ele possa ir trocando pelos foot switches os conjunto de efeitos salvos.

É notório que o projeto tem um ótimo potencial open source e que ao montar uma comunidade de desenvolvedores na área de DSP o projeto pode crescer muito mais, sendo esse um fator que pode ser explorado também.

REFERÊNCIAS

CIRRUS LOGIC, Inc. CS4272: 24-Bit, 192 kHz Stereo Audio CODEC. Datasheet. Austin: Cirrus Logic, Inc., outubro 2021. Disponível em: https://statics.cirrus.com/pubs/proDatasheet/CS4272_F2.pdf. Acesso em: 4 nov. 2024.

DINIZ, Paulo; SILVA, Eduardo; NETTO, Sergio. Processamento Digital de Sinais Projeto e Análise de Sistemas. 2. ed. Porto Alegre: Editora Bookman, 2014.

FREITAS, Lucas Bastos de. Efeitos sonoros parametrizáveis para guitarra através de processamento de sinais digitais. 2019. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Bauru, 2019. Disponível em: <https://repositorio.unesp.br/bitstreams/04acbc75-e5b8-487b-9cc8-0752f5a48063/download>. Acesso em: 2 nov. 2024.

JOSEPH, Gareth. History of Guitar Effects Pedals: From Fuzz to Modern Tech. Pedal Players, 2024. Disponível em: <https://pedalplayers.com/history-of-guitar-effects-pedals/#96d8cf6b-d9b2-48d9-9a1e-b9747fc2dfb5>. Acesso em: 14 mar. 2025.

KATZ, David; LUKASIAK, Tom; GENTILE, Rick. Fundamentals of embedded audio, part 3. EETimes, 17 set. 2007. Disponível em: <https://www.eetimes.com/fundamentals-of-embedded-audio-part-3/>. Acesso em: 7 dez. 2024.

MEDEIROS, Hugo Gayoso Meira Suassuna de. Projeto e execução de um pedal de guitarra baseado em FPGA. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal de Campina Grande, Campina Grande, 2019. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/19032>. Acesso em: 10 nov. 2024.

MULIN, Priscila. INVESTIGANDO A EXPERIÊNCIA MUSICAL. 2015. Dissertação de Mestrado (Educação, Artes e História da Cultura) - Universidade Presbiteriana Mackenzie, 2015. Disponível em: <https://dspace.mackenzie.br/bitstreams/140f45eb-224c-41a8-b108-42063a1c7f39/download#page=25>. Acesso em: 13 mar. 2025.

OLSHAUSEN, B. A. Aliasing. 2000. Disponível em: <https://www.rctn.org/bruno/npb261/aliasing.pdf>. Acesso em: 06 nov 2024.

PHIL'S LAB. Audio EQ Software Implementation (STM32) - Phil's Lab #89. YouTube, 29 dez. 2022. Disponível em: https://www.youtube.com/watch?v=4o-gUht_Xc. Acesso em: 25 jan. 2025.

PHIL'S LAB. Delay-Based Audio FX Software Implementation (DSP with STM32) - Phil's Lab #140. YouTube, 7 jul. 2024. Disponível em: https://www.youtube.com/watch?v=wd6l_wrGK_k. Acesso em: 11 dez. 2024.

PHIL'S LAB. DSP Overdrive Algorithm in Software (STM32) - Phil's Lab #117. YouTube, 8 ago. 2023. Disponível em: https://www.youtube.com/watch?v=_0ys155xv1Q. Acesso em: 9 dez. 2024.

PHIL'S LAB. Z-Transform - Practical Applications - Phil's Lab #27. YouTube, 21 jul. 2021. Disponível em: <https://www.youtube.com/watch?v=5Z3KAKs-EZs>. Acesso em: 9 nov. 2024.

PUHLMANN, Henrique. Trazendo o mundo real para dentro do processador – Conversor A/D. 2015. Disponível em: <https://embarcados.com.br/conversor-a-d/>. Acesso em: 5 nov. 2024.

SLAVKOVIC, David. A história da distorção: como o mais importante efeito de guitarra veio para ficar. Maquiavelli, [s.l.], [s.d.]. Disponível em: <https://www.maquiavelli.com/colunas/david-slavkovic-a-historia-da-distorcao-como-o-mais-importante-efeito-de-guitarra-veio-para-ficar>. Acesso em: 14 mar. 2025.

SPIILSBURY, M. J.; EUCEDA, A. Transformada Rápida de Fourier. Revista de la Escuela de Física, [S. l.], v. 4, n. 2, p. 45–52, 2016. DOI: 10.5377/ref.v4i2.8276. Disponível em: <https://camjol.info/index.php/fisica/article/view/8276>. Acesso em: 08 nov. 2024.

TONIDANDEL, Danny Augusto Vieira. Decifrando a transformada Z. In: XVIII Congresso Brasileiro de Automática, Bonito. 2010. p. 79.

WEDYKE, Nicholas. The history of the Maestro FZ-1: how the first-ever fuzz pedal changed music forever. Guitar World, [s.l.], 20 out. 2021. Disponível em: <https://www.guitarworld.com/features/history-of-the-maestro-fz1>. Acesso em: 14 mar. 2025.

WILMERING, Thomas; MOFFAT, David; MILO, Alessia; SANDLER, Mark B. A history of audio effects. Applied Sciences, Basel, v. 10, n. 3, p. 791, 2020. Disponível em: <https://doi.org/10.3390/app10030791>. Acesso em: 14 mar. 2025.

ZOLZER, U. DAFX: Digital Audio Effects. [S.l.]: John Willey & Sons, Ltd, 2011. ISBN 978-0-470-66599-2.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Luiz Henrique
Tipo do Documento:	Livro
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Luiz Henrique da Silva Oliveira, ALUNO (201811250020) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 19/03/2025 21:21:57.

Este documento foi armazenado no SUAP em 19/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1426791
Código de Autenticação: 0aa8185e6f

