

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LEONARDO HISASHI ONAGA**

**PROCESSAMENTO DIGITAL DE SINAIS DE ÁUDIO EM  
MICROCONTROLADOR**

**CAMPO MOURÃO**

**2022**

**LEONARDO HISASHI ONAGA**

**PROCESSAMENTO DIGITAL DE SINAIS DE ÁUDIO EM  
MICROCONTROLADOR**

**Digital processing of audio signals in microcontroller**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica, do Departamento Acadêmico de Engenharia Eletrônica, da Universidade Tecnológica Federal do Paraná (UTFPR).  
Orientador(a): Prof. Dr. Osmar Tormena Júnior.

**CAMPO MOURÃO**

**2022**



Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**LEONARDO HISASHI ONAGA**

**PROCESSAMENTO DIGITAL DE SINAIS DE ÁUDIO EM  
MICROCONTROLADOR**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Eletrônica, do Departamento  
Acadêmico de Engenharia Eletrônica, da Universidade  
Tecnológica Federal do Paraná (UTFPR).  
Orientador(a): Prof. Dr. Osmar Tormena Júnior.

Data de aprovação: 06/junho/2022

---

Eduardo Giometti Bertogna  
Doutorado em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná

---

Márcio Rodrigues da Cunha  
Doutorado em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná

---

Osmar Tormena Júnior  
Doutorado em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO**

**2022**

## AGRADECIMENTOS

Gostaria de agradecer primeiramente à minha família que me proporcionou esta oportunidade, e só por eles que este feito foi possível.

Meu orientador Prof. Dr. Osmar Tormena Júnior que sempre esteve de prontidão, me auxiliou e guiou neste último trajeto da graduação.

A Coordenação do Curso, pela prontidão e disposição para auxiliar com dúvidas e orientações.

A Lia Ohe que também esteve me apoiando e incentivando nos momentos mais difíceis.

Aos meus amigos que fiz nesta fase da minha vida: Maurício Cossich, William Sato, Sergio Schubert, Matheus Protasio, Matheus Zorawski, Eberton Matia, Rebeca Bertolim, Horion Dreher, Leonardo Yukio, Eduardo Garcia, Bruno Maiolli, Lucas Henrique, José Barreto, dentre outros. Levarei em meu coração todos momentos de projetos, grupos de estudos e risadas que aconteceram neste tempo tão curto.

## RESUMO

Neste trabalho foram desenvolvidos filtros digitais IIR (resposta ao impulso infinito) que atuam em sinais de áudio por meio de microcontrolador, visando alterações de magnitude. O trabalho se justifica pelas vantagens de se usar microcontroladores, principalmente sua portabilidade. Os filtros atuam em tempo real e pode ter alterações de amplitudes, ou seja, é possível alterar as atenuações e ganhos conforme seus coeficientes são alterados por um valor proporcional. O controle é feito em 5 divisões da faixa audível (baixas, médias-baixas, médias, médias-altas e altas). Para avaliação do sistema sonoro, foram utilizadas técnicas de medições objetivas e subjetivas. Em vista destes pontos, o trabalho proposto pode contribuir com pesquisas do gênero e trazer melhorias para sistemas digitais.

Palavras-chave: equalizador digital; filtro digital IIR; filtros digitais em microcontroladores.

## **ABSTRACT**

In this work, IIR (infinite impulse response) digital filters were developed that act on audio signals through a microcontroller, aiming at magnitude changes. The work is justified by the advantages of using microcontrollers, mainly its portability. The filters act in real time and can have amplitude changes, that is, it is possible to change the attenuations and gains as their coefficients are changed by a proportional value. Control is done in 5 divisions of the audible range (low, mid-low, mid, mid-high and high). To evaluate the sound system, objective and subjective measurement techniques were used. In view of these points, the proposed work can contribute to similar research and bring improvements to digital systems.

Keywords: digital equalizer; digital IIR filter; digital filters in microcontrollers.

## LISTA DE FIGURAS

Figura 1 – Sinal contínuo e discreto . . . . .	14
Figura 2 – Sinal interpolado . . . . .	15
Figura 3 – Equalização para alteração de resposta em frequência . . . . .	18
Figura 4 – Equalizador do tipo gráfico . . . . .	19
Figura 5 – Equalizador do tipo paramétrico . . . . .	19
Figura 6 – Estrutura resumida do sistema . . . . .	20
Figura 7 – Módulo DAC . . . . .	22
Figura 8 – Fluxograma básico do firmware . . . . .	22
Figura 9 – Fluxograma básico do driver I2S . . . . .	24
Figura 10 – Fluxograma básico do Bluetooth . . . . .	25
Figura 11 – Formado de dado . . . . .	26
Figura 12 – Fluxograma básico da rotina de processamento . . . . .	28
Figura 13 – Resposta magnitude do filtro low-shelving . . . . .	29
Figura 14 – Resposta magnitude dos 5 filtros sobrepostos . . . . .	30
Figura 15 – Fluxograma básico da rotina de interfaceamento . . . . .	31
Figura 16 – Resposta em frequência do filtro low-shelving . . . . .	32
Figura 17 – Comportamento simulado do filtro low-shelving . . . . .	33
Figura 18 – Captura do filtro low-shelving implementado . . . . .	34
Figura 19 – Resposta em frequência do filtro Peaking . . . . .	35
Figura 20 – Comportamento simulado do filtro peaking . . . . .	36
Figura 21 – Captura do filtro peaking implementado . . . . .	36
Figura 22 – Resposta em frequência do filtro high-shelving . . . . .	37
Figura 23 – Comportamento simulado do filtro high-shelving . . . . .	37
Figura 24 – Captura do filtro high-shelving implementado . . . . .	38

## LISTA DE ALGORITMOS

3.1	Configuração I2S . . . . .	23
3.2	Inicialização do <i>driver I2S</i> . . . . .	24
3.3	Código Bluetooth . . . . .	25
3.4	Código Processamento . . . . .	26



## LISTA DE TABELAS

Tabela 1 – Especificações dos filtros . . . . .	30
Tabela 2 – Conversão para decibel do primeiro teste simulado . . . . .	33
Tabela 3 – Conversão para decibel do primeiro teste prático . . . . .	34
Tabela 4 – Conversão para decibéis do segundo teste teórico . . . . .	35
Tabela 5 – Conversão para decibel do segundo teste prático . . . . .	36
Tabela 6 – Conversão para decibel do terceiro teste teórico . . . . .	38
Tabela 7 – Conversão para decibel do terceiro teste prático . . . . .	39
Tabela 8 – Comparativo dos resultados práticos gerais . . . . .	39

## LISTA DE ABREVIATURAS E SIGLAS

IIR	<i>Infinite Impulse Response</i>
IoT	<i>Internet of Things</i>
FIR	<i>Finite Impulse Response</i>
SNR	<i>Signal-to-Noise Ratio</i>
BLE	<i>Bluetooth Low Energy</i>
I2S	<i>Inter-IC Sound</i>
WiFi	<i>Wireless Fidelity</i>
ADC	<i>Analogic-to-Digital Converter</i>
DAC	<i>Digital-to-Analogic Converter</i>
MSB	<i>Most-Significant Bit</i>
IDF	<i>IoT Development Framework</i>
MCLK	<i>Master Clock</i>
BCK	<i>Bit Clock</i>
WS	<i>Word Selector</i>
DMA	<i>Direct Memory Access</i>
A2DP	<i>Advanced Audio Distribution Profile</i>
SBC	<i>Sub-Band Coding</i>
API	<i>Application Programming Interface</i>
hdl	<i>Handler</i>
SOS	<i>Second-Order Sections</i>
dB	Decibel
V	Tensão elétrica
CH1	<i>Channel-1</i>
CH2	<i>Channel-2</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	11
1.1	Objetivos gerais	12
1.2	Objetivos específicos	12
1.3	Justificativa	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	13
2.1	Sinais e sistemas contínuos	13
2.2	Sinais e sistemas discretos	13
2.3	Teorema da amostragem	13
2.4	Filtros digitais	15
2.5	Filtros FIR	15
2.6	Filtros IIR	16
2.7	Ruído	16
2.8	Bluetooth	17
2.9	Matlab	17
2.10	Equalizador	17
2.10.1	Tipos de equalizadores	18
<b>3</b>	<b>METODOLOGIA</b>	20
3.1	Desenvolvimento do Hardware	21
3.2	Desenvolvimento do Firmware	22
3.2.1	Periférico I2S	23
3.2.2	Bluetooth	24
3.2.3	Processamento do sinal	26
3.2.4	Banco de filtros	28
3.2.5	Interfaceamento por <i>encoder</i>	30
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	32
4.1	Análise de desempenho	32
4.1.1	Análise do filtro <i>low-shelving</i>	32
4.1.2	Análise do filtro <i>peaking</i>	34
4.1.3	Análise do filtro <i>high-shelving</i>	37
4.1.4	Análise e discussão dos resultados	39
<b>5</b>	<b>CONCLUSÃO</b>	41
	<b>REFERÊNCIAS</b>	42

## 1 INTRODUÇÃO

O som, em sua essência, é um fenômeno físico que varia a pressão atmosférica em torno da fonte sonora e tem sua propagação na forma longitudinal. No som, pode-se caracterizar duas propriedades indispensáveis para o estudo: a intensidade e a frequência (FERNANDES, 2002).

A intensidade está diretamente relacionada ao nível de energia das vibrações, portanto, quanto mais alto o som, maior será a amplitude da onda sonora. A frequência é unidade de medida para o número de oscilações em um segundo e é pelas diferentes frequências que caracterizam as diversas tonalidades de sons (FERNANDES, 2002).

O sistema denominado equalizador tem como finalidade alterar ganhos de sinais para determinadas frequências afim de corrigir sinais agudos ou graves para que se possa ouvir todas tonalidades em mesma intensidade (VALIANTE FILHO, 2004). Os circuitos analógicos eram predominantes nestes sistemas, isto devido a falta de tecnologia digital para esta área entre as décadas de 1970 a 1990 (BRAGA, 2012). A tecnologia digital já existia no final do século XX, porém seu uso era limitado a grandes processadores de sistemas computacionais (FLOYD, 2007).

Mesmo com a tecnologia que temos atualmente em sistemas de som, considerando o percurso do sinal desde a sua geração à recepção, ainda não é possível reproduzir fielmente o sinal original devido as diversas limitações. Por este motivo, é necessário um esforço contínuo para minimizar os distúrbios gerados nestes percurso. É o papel de pesquisadores capazes, desenvolver métodos e tecnologias que superem os limites atuais levando em conta a viabilidade e a inovação (HERRERA, 2004).

Para avaliar o som com o sentido sonoro, não há parâmetros a se seguir em testes locais para que se obtenha um resultado qualitativo. São parâmetros subjetivos em que a maioria dos profissionais das áreas contam com suas experiências e audição para avaliar a qualidade dos dispositivos. Mesmo que haja um padrão, existe muitas variáveis do ambiente difíceis de se controlar para execução de uma análise subjetiva. Para determinar um padrão de qualidade, os fabricante utilizam parâmetros objetivos que são totalmente computadorizados e em ambientes controlados, como por exemplo: distorção harmônica total, resposta em frequência, relação sinal e ruído. Eles servem para ajustes de som levando em conta a acústica do ambiente, remover ruídos indesejados, regular ganhos em determinadas frequências (BECH, 1992).

O projeto deve compreender o processamento de áudio na faixa audível para os humanos, que vai de 20Hz a 20kHz. A principio, esta equalização será feito em cinco faixas, sendo uma delas para baixas frequências (200Hz), para frequências médias-baixas (4kHz), as médias frequências (8kHz), outra para médias-altas (12kHz) e a última para alta frequência (16kHz). Pode-se alterar o número de bandas dependendo da necessidade e do espaço de armazenamento disponível.

## 1.1 OBJETIVOS GERAIS

Desenvolver um dispositivo de equalização de áudio para dois canais estereofônicos com aquisição de áudio via Bluetooth. Os canais estereofônicos são de sinais separados que distinguem o áudio no espaço, tornando mais realista o ambiente sonoro. A equalização traz a liberdade de escolher o ganho ou atenuação de qualquer faixa de frequência.

## 1.2 OBJETIVOS ESPECÍFICOS

Este trabalho possui os seguintes objetivos específicos:

- Projetar circuitos periféricos do dispositivo.
- Projetar o *hardware* do equalizador.
- Projetar o *firmware* responsável pela aquisição e envio dos sinais.
- Projetar e implementar ao *firmware* os filtros IIR.
- Coletar amostras dos sinais filtrados para as determinadas frequências.

## 1.3 JUSTIFICATIVA

Considerando a notável evolução de circuitos elétricos desde as últimas décadas, principalmente quando se tem em vista a transição de circuitos analógicos para digital, o projeto e desenvolvimento de filtros digitais é vantajoso para sistemas de processamento, visto que pode-se chegar a melhor desempenho se comparado aos analógicos (SMITH, 1999). O mesmo vale para sistemas de sons, que tem como pontos importantes a imersão e a qualidade de reprodução, melhorando assim a qualidade do dispositivo que forneça uma melhor experiência ao usuário.

O momento atual proporciona aos profissionais relacionados a eletrônica, uma alta demanda de conhecimento nas áreas de automação e sistemas digitais, sob o ponto de vista da tendência a automatização de processos industriais nos países com alta tecnologia. Este momento é chamado de Quarta Revolução Industrial e necessita de inovações nas mais diversas áreas tecnológicas, assim como automatização, processamento de dados, internet das coisas (IoT), sistemas embarcados, microeletrônica, entre outras (DRONESHOW, 2019). Dentro desta realidade, há profissionais na área de sistemas digitais, porém, poucos se aprofundam na área de áudio e som, sendo necessário o incentivo à pesquisas e projetos em ambientes acadêmicos e empreendedores.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 SINAIS E SISTEMAS CONTÍNUOS

Um sinal de tempo contínuo é caracterizado pelo seu domínio contínuo e, portanto, é um conjunto contínuo deste eixo. Tem como representação a letra  $t$  como notação de domínio, conhecido como  $x(t)$  (OPPENHEIM; WILLSKY, 2010). O sinal contínuo é aquele obtido na saída de um dispositivo de áudio (LATHI, 2007). O sinal classificado como contínuo que puder assumir qualquer valor de domínio em um intervalo  $(a, b)$  indefinido, pode ser classificado também como analógico (HSU, 2012).

### 2.2 SINAIS E SISTEMAS DISCRETOS

Segundo Oppenheim e Willsky (2010), um sinal discreto pode ser entendido como uma sequência de impulsos igualmente espaçados no domínio do tempo, porém, só é possível esta conversão sob certas condições. O sinal discreto é representado somente em números inteiros no domínio, e em literatura leva  $n$  como notação sendo usual a representação  $x(n)$ .

Para um sinal discreto ser classificado como digital, ele deve assumir uma quantidade finita de valores inteiros e distintos do domínio (HSU, 2012).

A base da área de processamento digital de sinais é o estudo da natureza de sinais de domínio discreto, e também os sistemas que atuam no sinal. Para poder processar um sinal em um hardware digital, é necessário converter o domínio de tempo contínuo para tempo discreto, e também um meio de recuperá-lo para forma contínua. A operação de um sistema em cima de um sinal analógico é muito mais complexa se comparado ao equivalente digital (DINIZ; SILVA; NETTO, 2014).

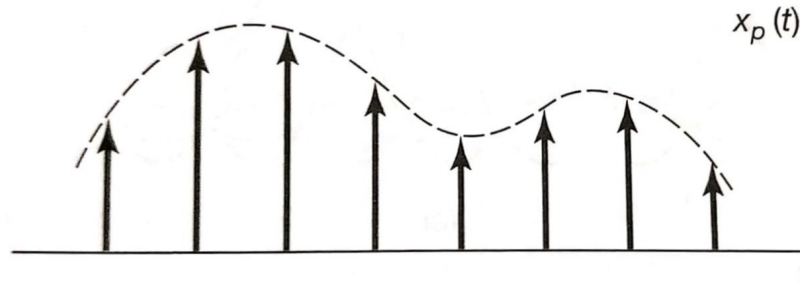
O sistema discreto faz um mapeamento do sinal por meio de cada amostra. Para os sistemas, existem algumas classificações que se deve dar importância, como a linearidade de fase e a invariância no tempo (DINIZ; SILVA; NETTO, 2014). Este modelo é usado por ser conveniente à análise matemática do circuito, devido a aplicação da propriedade de superposição e a característica da invariância ao tempo (OPPENHEIM; WILLSKY, 2010)

### 2.3 TEOREMA DA AMOSTRAGEM

A amostragem é uma ponte entre o sinal contínuo e discreto. Este conceito é de grande importância, pois nas últimas décadas, sistemas digitais tiveram grande avanço tecnológico. Este tipo de sistema possui algumas vantagens se comparado aos analógicos: baixo custo, circuitos mais compactos e reprogramáveis (OPPENHEIM; WILLSKY, 2010). Um sinal contínuo

que possui infinitos pontos na variável independente ao ser amostrado, se transforma em um sinal com informação em determinados pontos do domínio, conforme a Figura 1:

**Figura 1 – Sinal contínuo e discreto**



**Fonte: Adaptado de Oppenheim e Willsky (2010).**

Não só no domínio do tempo deve-se observar o comportamento da amostragem, pois, se tem incontáveis funções de sinais que podem gerar um conjunto de amostras. Uma condição para um sinal ser amostrado é ser limitado em banda, pois, com isto é garantido a unicidade da representação e também a reconstrução (OPPENHEIM; WILLISKY, 2010).

Ainda com base em Oppenheim e Willsky (2010), um trem de impulsos no tempo é conveniente para representar um sinal amostrado, e ao multiplicá-lo com uma função  $x_1(t)$ , se obtém o sinal representado no tempo discreto.

$$x_p(t) = \sum_{n=-\infty}^{+\infty} x_1(nT)\delta(t - nT). \quad (1)$$

Onde, na Equação (1),  $x_p(t)$  é um sinal amostrado com espaçamento de tamanho  $T$  e com amplitudes relativas ao sinal  $x_1(t)$  (OPPENHEIM; WILLISKY, 2010).

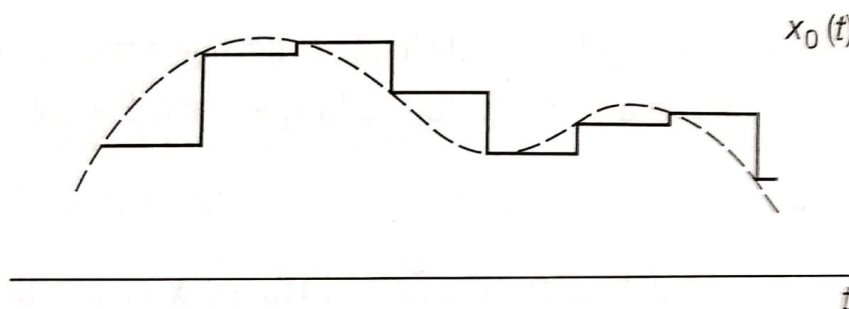
Uma multiplicação de sinais no tempo é uma convolução na frequência, então ao convoluir o sinal com um impulso, o sinal é deslocado. No domínio frequência a resposta se torna periódica com sobreposições do espectro original com centros deslocados igualmente em  $\omega_s$  (frequência de amostragem)

$$X(j\omega) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X(j(\omega - k\omega_s)) \quad (2)$$

Da perspectiva do espectro, a frequência de amostragem é de grande peso neste tópico. Se  $\omega_s$  é menor que a largura de banda do sinal ( $\omega_M$ ), então, neste caso haverá sobreposição de sinal na frequência, causando perda de informação. Caso  $\omega_s$  seja maior ou igual a duas vezes largura de banda, o sinal pode não sofrer sobreposição e podendo ser reconstruído. Mas, é garantido que o sinal seja reconstruído se usar,  $\omega_s > 2 \omega_M$ , sendo conhecido também como taxa de Nyquist (OPPENHEIM; WILLISKY, 2010).

A recuperação do sinal amostrado é feito por interpolação como ilustrado na Figura 2. Este método detecta amostras e retém o valor até um determinado tempo, isto faz com que o sinal fique contínuo novamente (OPPENHEIM; WILLISKY, 2010).

Figura 2 – Sinal interpolado



Fonte: Adaptado de Oppenheim e Willsky (2010).

## 2.4 FILTROS DIGITAIS

Filtros digitais e analógicos têm a mesma finalidade, porém, cada um tem suas especificidades. O filtro em si pode ser usado para separar sinais, ou então a restauração do mesmo. (SMITH, 2003). O hardware digital é dinâmico, pois, pode-se mudar o comportamento do sistema com reprogramação, enquanto analógicos precisam de outro projeto de hardware. Além das variações físicas de temperatura, tempo de uso, entre outros, não interferem no funcionamento do sistema digital (LATHI, 2007).

Os filtros digitais são tratados como sistemas de tempo discreto, que este, por sua vez age sobre os sinais também discretos. O sistema é visto como uma função de transferência  $H(z)$ , que pode ser visto com diferentes formas dependendo do tipo de filtro. Existem duas classificações relevantes para filtros: o normalmente não-recursivo, que possui a vantagem de ser de fase linear, também conhecido por ter resposta finita ao impulso (FIR), e sua  $H(z)$  ter forma polinomial; e o filtro recursivo, conhecido por ter resposta infinita ao impulso com forma racional polinomial em  $H(z)$  (DINIZ; SILVA; NETTO, 2014). O operador mais comum para aplicá-los é a convolução, o qual pode ser tratado no domínio do tempo, ou então a transformada de Fourier que trabalha na frequência (MADISETTI; WILLIAMS, 1998).

## 2.5 FILTROS FIR

O filtro FIR é visto como um sistema não recursivo de ordem N com coeficientes em formato de polinômios a serem determinados. O filtro FIR é estável por natureza e possui fase linear, porém, existe um problema se comparado ao IIR de mesmas especificações: a ordem numérica, isto é, em hardware é necessário um maior valor de armazenamento (DINIZ; SILVA; NETTO, 2014).



Os filtros de ordem maior que a ordem da centena, tendem a serem eficientes na banda de trabalho, ou seja, banda de transição menor. Por outro lado, se tem uma quantidade enorme de operações por segundo, conseqüentemente um maior custo computacional. No filtro existem alguns de seus parâmetros determinados empiricamente por modelos FIR. O filtro pode ser recursivo, porém, o faz insensível a ruído a não-recursividade (MADISETTI; WILLIAMS, 1998).

Para os filtros FIR tem 4 tipos de configuração: tipo I, é simétrico e de ordem par; tipo II, simétrico de ordem ímpar; tipo III, anti-simétrico de ordem par; tipo IV, anti-simétrico de ordem ímpar. Cada modelo tem suas especificidades, como por exemplo, os modelos 3 e 4 que são usados em derivadores e transformadores de Hilbert. Já o tipo I pode-se implementar qualquer tipo de sistema, enquanto o tipo II não se consegue fazer implementação de filtros passa-baixa ou rejeita-faixa (TORMENA JUNIOR, 2018).

## 2.6 FILTROS IIR

O filtro é de resposta impulsiva de suporte infinito. Como já dito, a função de transferência de um filtro IIR é normalmente tida como uma função racional. Na Equação (3),  $N_0$  é constante numérica; a ordem do denominador é a ordem  $N$  do filtro e também a quantidade de polos do filtro; e o numerador quantifica os zeros. A ordem do filtro é baixa, se comparado ao FIR, porém isto causa problemas de quantização, já que a ordem determina a quantidade de amostras anteriores à qual está sendo processada. O filtro tem níveis de ruído consideráveis quando se está trabalhando em um hardware com arquitetura de ponto-fixa, e que podem ser amplificados pela recursividade (MADISETTI; WILLIAMS, 1998).

$$H(\omega) = e^{(j\omega N_0)} \frac{\sum_{k=0}^M b_k e^{-j\omega k}}{\sum_{k=0}^N a_k e^{-j\omega k}} \quad (3)$$

O filtro é projetado no modelo analógico com especificações de módulo, fase com base em modelos clássicos e posteriormente convertido em digital. Apesar do nível de complexidade ser maior, pode-se também projetar e otimizar a função de transferência com especificações de módulo e fase no domínio discreto. Existem dois métodos comumente usados para conversão de sistema: a invariância ao impulso e a transformação bilinear. O primeiro método é a convolução ao impulso no modelo analógico e, onde por meio de manipulação, obtém-se a resposta ao impulso de um filtro digital de mesmo comportamento do filtro analógico já projetado. Já o segundo, transformação bilinear, é um mapeamento da função de transferência para o plano  $z$ . Este método tem vantagens sobre a invariância ao impulso, o *anti-aliasing*, que garante a não-perda dos sinais de borda (DINIZ; SILVA; NETTO, 2014).

## 2.7 RUÍDO

O ruído é um sinal aleatório indesejado que provem de diversas fontes e ocorre sobre um sinal conhecido. Existe várias fontes de ruído, e entre elas, a mais comum é de energia térmica.

Conforme aumenta a intensidade da temperatura no dispositivo eletrônico, mais significativo é o ruído (OTT, 1988).

O nível de ruído em si não se traduz em informações relevantes, porém ao compará-lo com o sinal, pode-se chegar a chamada relação sinal-ruído (SNR), sendo esta, uma medida para comparar a magnitude do sinal desejado com o nível de ruído no mesmo sinal. A relação é obtida através da divisão da potência média do sinal pelo ruído, onde valores maiores que 1 são desejados (OTT, 1988).

Quando há uma amostragem de um sinal contínuo, um ruído é adicionado ao sinal causado por problemas de quantização. Este ruído se tem pelo fato do passo digital não representar exatamente o sinal analógico (OTT, 1988).

## 2.8 BLUETOOTH

A transferência de dados sem fio sem a perda de dados junto ao conjunto versátil de soluções para sua implementação é o que faz desta tecnologia muito difundida. Atualmente, existe duas versões de Bluetooth: o *classic* e o *low energy* (LE) (BLUETOOTH, 2022).

O Bluetooth *classic* é um sinal de baixa potência que cobre canais de frequência de 2.4 *Ghz* do tipo industrial, científica e médica. Esta versão é voltada para *streaming* de áudio e transferência de dados. Possui um tipo de comunicação conhecida como ponto-a-ponto, onde o *host* pode enviar dados para um ou mais dispositivos conectados a ele. O Bluetooth LE cobre menos canais de 2.4 *Ghz*, já que é um sinal de baixa energia. Esta versão suporta várias topologias de comunicações, assim como a ponto-a-ponto, também suporta o tipo *broadcast* e *mesh* que possibilita obtenção de dados entre os dispositivos como distância, direção, presença e proximidade (BLUETOOTH, 2022).

## 2.9 MATLAB

O Matlab é um *software* de programação baseado em matrizes, de linguagem própria e desenvolvida pela MathWorks. O *software* permite implementação de algoritmo, manipulação de sistemas, plotagem de funções, além de possuir interfaceamento com outras linguagens (MATLAB, 2022).

## 2.10 EQUALIZADOR

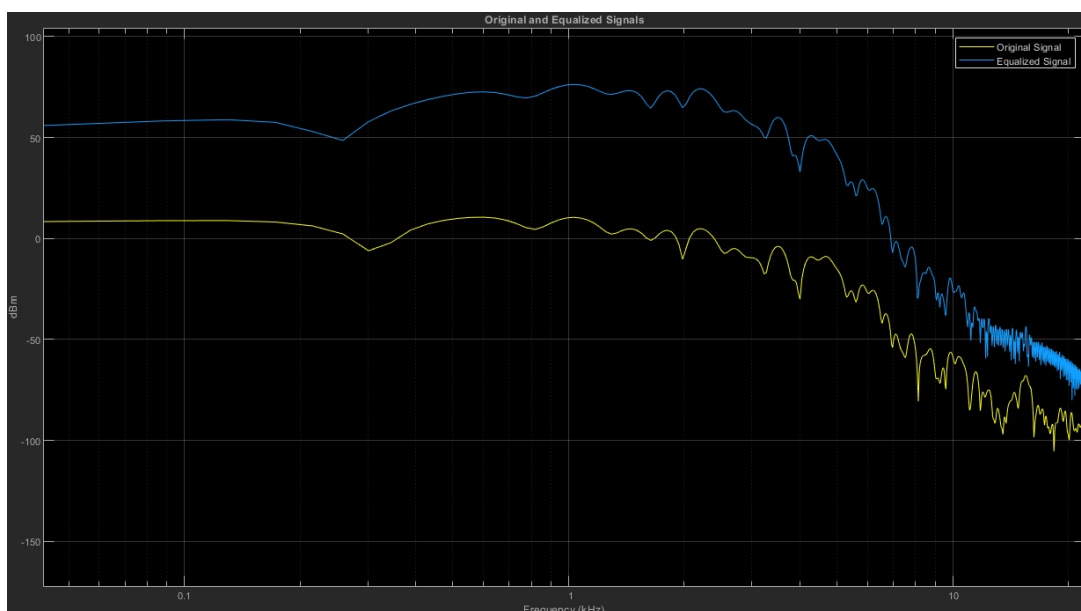
Um sistema de equalização em sua essência, tem como finalidade manter a linearidade da resposta em frequência de um sistema de áudio, em busca do mais cristalino e próximo ao real (ORFANIDIS, 2010). Nos sistemas de reprodução de áudio, que vem desde a captura, passando pelo sistema amplificador, até os transdutores, tem como objetivo trazer o som mais fiel à realidade. Porém há limitadores, seja pela tecnologia, dimensionamento de projeto ou

materiais, esta limitação introduz distorções no sinal de áudio. Deste modo, pode-se definir que a equalização é a inserção de distorções controladas a fim de compensar a não-linearidade da resposta em frequência do conjunto (HERRERA, 2004).

Existe também uma outra visão sobre a equalização, denominada de "equalização artística" que visa alterar a resposta em frequência para fins cosméticos como mostra a Figura 3. Sua maior utilização fica por parte de usuário que não busca compromisso com a fidelidade à gravação (HERRERA, 2004).

### 2.10.1 Tipos de equalizadores

**Figura 3 – Equalização para alteração de resposta em frequência**

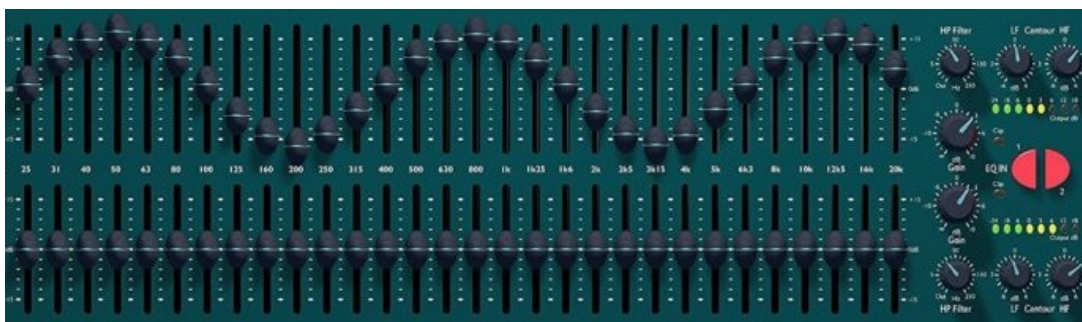


**Fonte: Adaptado de MATLAB (2022).**

Há quatro modelos amplamente conhecidos de equalizadores: o gráfico, semi-paramétrico, paramétrico e o totalmente paramétrico. O modelo gráfico é o mais intuitivo e de fácil aprendizado para se controlar e chegar na sonoridade desejada (ROSA, 2012). Já o paramétrico envolve outros controles além da magnitude, como a frequência central e largura de banda (ORFANIDIS, 2010).

O equalizador gráfico é o mais comum e intuitivo, isto se deve ao motivo deste modelo apresentar o interfaceamento em *sliders* de forma paralela, onde cada um representa uma frequência, como mostra a Figura 4. Ele é composto, a maioria das vezes, por filtros do tipo *peaking*, com frequência e fator Q fixos (parâmetro de amortecimento), tendo somente a magnitude manipulável (WARPSOUND, 2017). Por este motivo, ele também é conhecido como não-paramétrico, onde a única variável manipulável é o ganho (OSSIA, 2016).

Figura 4 – Equalizador do tipo gráfico



Fonte: Adaptado de WARPSOUND (2017).

Existe também o equalizado semi-paramétrico, onde além de poder alterara a magnitude, a frequência central também pode ser alterada (OSSIA, 2016).

Já o equalizador paramétrico, ilustrado na Figura 5, exige um entendimento dos parâmetros que estão a disposição para serem manipulados. Neste, não só a magnitude, mas a largura de banda, o fator Q e até mesmo a frequência central podem ser alterados, oferecendo um controle maior sobre a sonoridade (WARPSOUND, 2017).

Figura 5 – Equalizador do tipo paramétrico



Fonte: Adaptado de WARPSOUND (2017).

O equalizador totalmente paramétrico, oferece a liberdade para escolha do tipo de filtro que será utilizado, além de todos os outros parâmetros citado no equalizador paramétrico (OSSIA, 2016).

### 3 METODOLOGIA

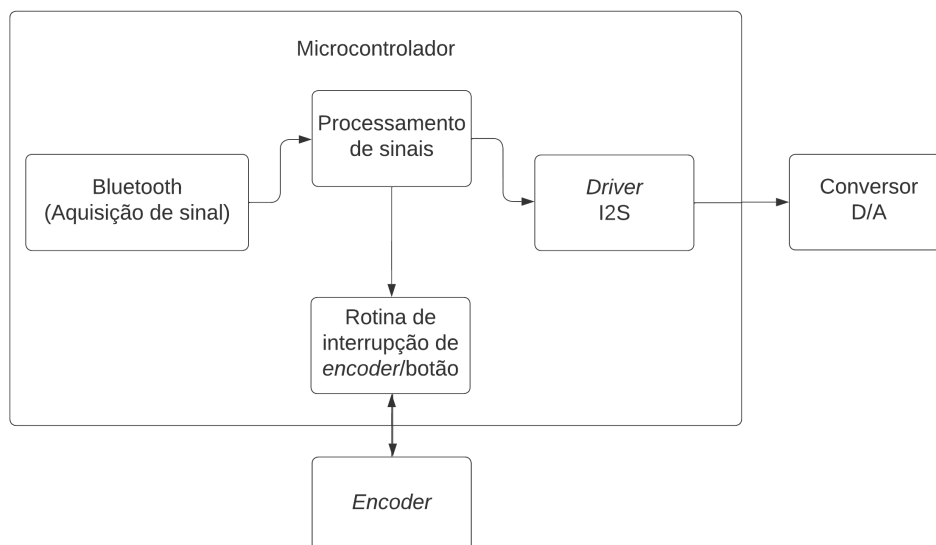
Foi necessário para o desenvolvimento deste trabalho as seguintes etapas: estudo e pesquisa, caracterização do projeto, desenvolvimento e simulações dos filtros, projeto e implementação de *software* e *hardware*, e por último, testes para ajustes.

Na etapa de estudo e pesquisa foi realizado um levantamento de materiais e trabalhos já realizados nesta área, a fim de aprofundar conhecimento para um melhor embasamento teórico. A segunda etapa, definiu-se as características intrínsecas do projeto, como a quantidade de bandas controladas, interfaces de controle e a configuração de outras funcionalidades que o equalizador deve ter como, por exemplo, conversores digitais-analógicos.

Posteriormente, teve-se a implementação dos filtros digitais no *software* Matlab da MathWorks, para o banco de filtros onde, basicamente, são dos tipos *shelving* e *peaking* paramétrico. O projeto será limitado pelo poder de processamento do *hardware*, uma vez já definido. Ainda nesta etapa, houve uma avaliação subjetiva de desempenho do processamento, visando a amplitude de ruído causado pelos recursos finitos das operações matemáticas, ou seja, o ruído provém de erros de quantização.

Na quarta etapa, se deu ao projeto e desenvolvimento do *firmware*, o qual engloba toda parte de códigos, assim como a implementação da biblioteca Bluetooth responsável pela entrada do sinal, implementação dos *drivers* do periférico *inter-IC sound* (I2S) e a programação dos níveis do *encoder* para os filtros. Também houve o projeto e desenvolvimento do *hardware*, que consiste em montar o protótipo do circuito equalizador atendendo todas especificações e a interligação entre os componentes do circuito.

**Figura 6 – Estrutura resumida do sistema**



**Fonte: Autoria própria (2022).**

Ao final da junção do algoritmo com circuito do projeto, realizou-se testes para verificar a funcionalidade do sistema e avaliar os resultados obtidos por meio de medições objetivas.

### 3.1 DESENVOLVIMENTO DO HARDWARE

A escolha do *hardware* foi baseada em pesquisa de tecnologias disponíveis no microcontrolador.

O microcontrolador deste projeto é o ESP32, desenvolvido pela Espressif System, e é o componente responsável pelo armazenamento do *firmware* e execução do mesmo. Este modelo possui tecnologias importantes para IoT como Bluetooth e WiFi (Wireless Fidelity), além de possuir dois núcleos de processamento em um único *chip* semiconductor. Para facilitar o desenvolvimento, foi escolhido uma placa de desenvolvimento ESP-WROOM-32, que possui regulador de tensão para alimentação de 5 V para 3.3 V, um conversor USB-UART para gravar o *firmware*, e outras características do próprio *chip* como:

- 2 Núcleos Xtensa 32-bit LX6 com frequência entre 80 – 240 MHz.
- 4 MB de memória *flash* externa.
- 520 KB de SRAM para instruções.
- Suporte para comunicação I2S.

O Bluetooth, responsável pela aquisição do sinal de áudio, não se faz necessário um módulo em vista que o *chip* ESP32 oferece em seu *hardware* e sua implementação é totalmente via *firmware*. Caso contrário, além de necessitar de um módulo ADC (*Analogic-to-Digital Converter*) externo, o dispositivo *host* de áudio ficaria limitado à distância, já que a entrada seria via cabo.

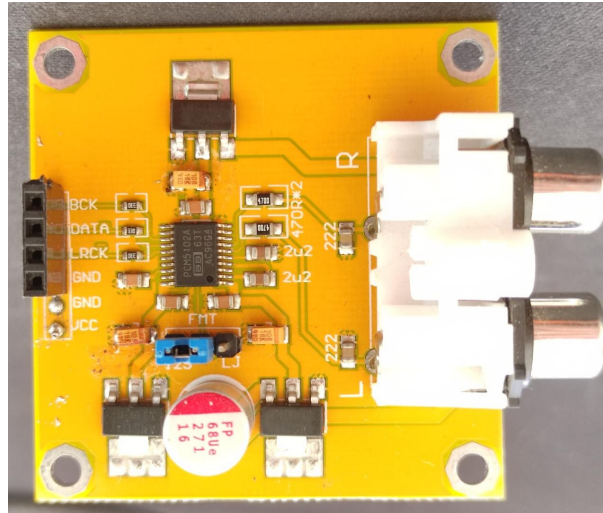
A saída do sinal de áudio é feita por um DAC (*Digital-to-Analogic Converter*) externo que pode ser conectado tanto a um transdutor de baixa potência (fones de ouvido), quanto a etapa de potência (amplificador) por meio de dois canais, proporcionando um sinal estéreo. O *chip* DAC escolhido para o trabalho é o PCM5102, o qual traz uma resolução em 16, 24 e 32 *bits* por amostra, sendo superior ao DAC interno do microcontrolador de apenas 8 *bits*. Aceita comunicação no formato I2S e *Left-justified*, o formato de dado é aceito em MSB (*most significant bit*) ou complemento de dois e tem escala de saída com valor de tensão de 2.1  $V_{RMS}$ . Sua construção gera pouco ruído trazendo um SNR de 112 dB.

Foi escolhido um módulo PCM5102 já montado, possuindo três reguladores de tensão (LM1117 - 3.3 V) individuais para as três entradas de fonte. Todos os conectores do *chip* são em nível digital, porém a saída do áudio é analógica e possui filtros passa baixa em ambos canais de áudio para atenuar ruídos de alta frequência, como mostra a Figura 7.

A interface entre o microcontrolador e o usuário é feita através de um *encoder* modelo EC11, o qual além de possuir o eixo de rotação, havendo também um botão em sua base que

é acionado quando pressionado. Este componente possui cinco conectores, onde três são para leitura da rotação e dois são do botão.

**Figura 7 – Módulo DAC**



**Fonte: Autoria própria (2022).**

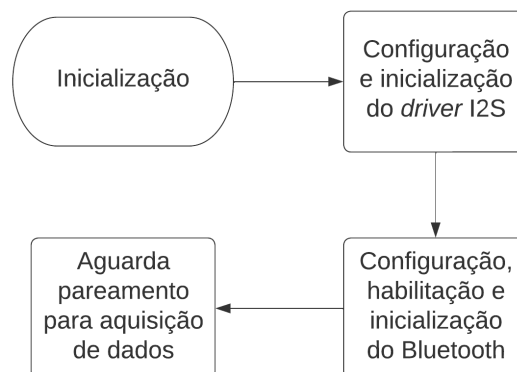
A rotação do *encoder* é atribuída à rotina de variação dos coeficientes dos filtros e manipula um filtro por vez. O sentido horário incrementa o valor de ganho e o anti-horário decrementa até atingir os valores máximos e mínimos, respectivamente estipulados. Dos três conectores que possui, dois são para sinais e o terceiro para referência.

Já o botão é implementado na mesma rotina que o *encoder* e é responsável pela seleção do filtro o qual será alterado.

### 3.2 DESENVOLVIMENTO DO FIRMWARE

A rotina base de inicialização do microcontrolador é ilustrado na Figura 8.

**Figura 8 – Fluxograma básico do firmware**



**Fonte: Autoria própria (2022).**

O desenvolvimento do *firmware* foi feito na linguagem C e compilado no próprio *framework* da Espressif System, conhecido como ESP-IDF (*IoT development framework*).

### 3.2.1 Periférico I2S

A rotina denominada *app\_main* inicializa primeiramente o *driver* do I2S, com fluxo ilustrado na Figura 9, e que faz configuração de parâmetros de comunicação entre microcontrolador e o DAC, como mostra a implementação na Listagem 3.1.

#### Listagem 3.1 – Configuração I2S

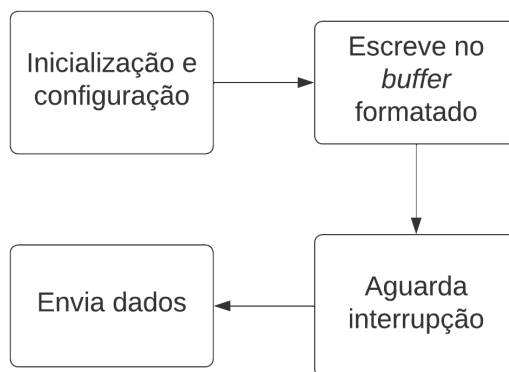
```
1 app_main(){
2   i2s_config_t i2s_config = {
3       .mode = I2S_MODE_MASTER | I2S_MODE_TX,
4       .sample_rate = 44100,
5       .bits_per_sample = 16,
6       .channel_format = I2S_CHANNEL_FMT_RIGHT_LEFT,
7       .communication_format = I2S_COMM_FORMAT_I2S_MSB,
8       .dma_buf_count = 6,
9       .dma_buf_len = 60,
10      .intr_alloc_flags = 0,
11      .tx_desc_auto_clear = true
12  };
13
14  i2s_driver_install(0, &i2s_config, 0, NULL);
15
16  i2s_pin_config_t pin_config = {
17      .bck_io_num = CONFIG_EXAMPLE_I2S_BCK_PIN,
18      .ws_io_num = CONFIG_EXAMPLE_I2S_LRCK_PIN,
19      .data_out_num = CONFIG_EXAMPLE_I2S_DATA_PIN,
20      .data_in_num = -1
21  };
22
23  i2s_set_pin(0, &pin_config);
24
25  [...]
26 }
```

A função *i2s\_driver\_install* inicia o periférico I2S, onde o primeiro parâmetro indica qual *driver* I2S é instanciado (*chip* oferece dois periférico I2S), carrega as configurações contidas na *struct i2s\_config* como segundo parâmetro, e os dois últimos parâmetros indicam tamanho e endereço da fila, respectivamente, e que não são utilizados. Já a função *i2s\_set\_pin* configura



os pinos do microcontrolador para a comunicação que é passado a *struct pin\_config* no segundo parâmetro, e o primeiro indica qual dos dois *drivers*.

**Figura 9 – Fluxograma básico do driver I2S**



**Fonte: Autoria própria (2022).**

O microcontrolador é configurado como mestre e neste modo o *chip* DAC aciona o PLL (*phase-locked loop*) interno, o qual é usado como referência para gerar o *clock master* (MCLK). Com esta configuração a entrada *BCK* (Bit Clock) tem uma frequência de  $1.411\text{ MHz}$ , fazendo referência para o *clock master* de  $11.289\text{ MHz}$  no DAC, e a frequência do seletor de canal (WS) em  $44.1\text{ kHz}$ , que também é provida do microcontrolador, como mostra a Listagem 3.2 do monitor de inicialização do mesmo.

**Listagem 3.2 – Inicialização do driver I2S**

```

1 | (597) I2S: APLL: Req RATE: 44100, real rate: 44099.988, BITS:
2 | 16, CLKM: 1, BCK_M: 8, MCLK: 11289597.000, SCLK: 1411199.625000,
3 | diva: 1, divb: 0
  
```

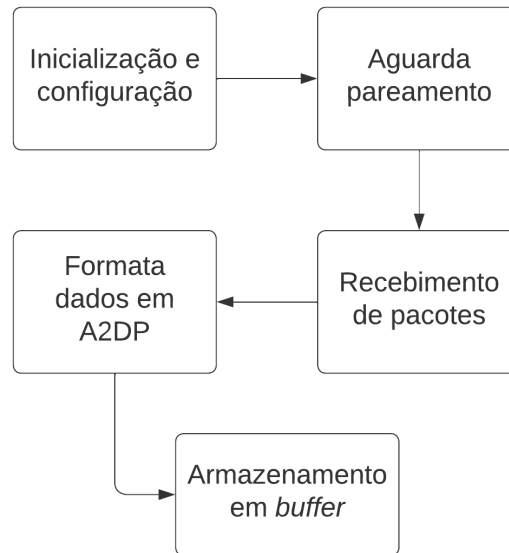
As configurações na *struct i2s\_config* do I2S no microcontrolador são: modo mestre e transmissor,  $44.1\text{ kHz}$  de taxa de amostragem, 16 *bits* por amostra, canal estéreo, formato I2S MSB, 6 *buffers* DMA (Direct Memory Access), 60 amostras por *buffer*, prioridade da interrupção, e por último, a limpeza automática do *buffer* em caso de *underflow*. A quantidade de *bits* por amostra fica limitada a 16 pelo fato do Bluetooth não suportar 24 e 32 *bits*.

### 3.2.2 Bluetooth

Ainda na rotina principal, também é inicializado o *driver* Bluetooth e suas configurações com fluxo na Figura 10. Dentro da tecnologia Bluetooth, existem alguns formatos para transferência de dados de áudio, e o escolhido para este trabalho é conhecido como *advanced audio distribution profile* (A2DP). Este foi o primeiro formato a ser utilizado em dispositivos de som portáteis e fones de ouvido sem fio, e por este motivo é muito difundido. E o *codec* utilizado é o SBC (codificação em sub-banda) que tem suporte até  $44.1\text{ kHz}$  de frequência de

amostragem e formata dados em *unsigned int* MSB de 16 *bits*, como citado anteriormente, onde este é compatível com protocolo I2S.

**Figura 10 – Fluxograma básico do Bluetooth**



**Fonte: Autoria própria (2022).**

O *framework* da Espressif System disponibiliza uma API (*application programming interface*) do Bluetooth e exemplifica a inicialização com um código nomeado *A2DP\_Sink*, o qual foi usado como base para o trabalho, mostrado na Listagem 3.3.

**Listagem 3.3 – Código Bluetooth**

```
1 app_main() {
2   [...]
3
4   esp_bt_controller_config_t bt_cfg = BT_CONTROLLER_INIT_CONFIG
5   _DEFAULT();
6
7   esp_bt_controller_init(&bt_cfg);
8
9   esp_bt_controller_enable(ESP_BT_MODE_CLASSIC_BT);
10
11  esp_bluedroid_init();
12
13  esp_bluedroid_enable();
14
15  /* create application task */
16  bt_app_task_start_up();
17
```

```

18 /* Bluetooth device name, connection mode and profile set up */
19 bt_app_work_dispatch(bt_av_hdl_stack_evt, BT_APP_EVT_STACK_UP,
20 NULL, 0, NULL);
21
22 }

```

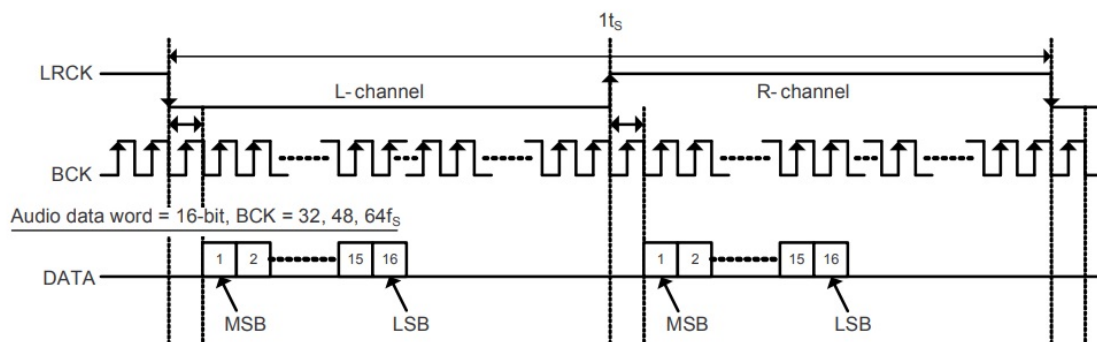
Nesta parte da inicialização é definida primeiramente a *struct bt\_cfg*, que contém parâmetros padrão para a rotina do Bluetooth, e é uma variável que não necessita alterações. As funções *esp\_bt\_controller\_init* e *esp\_bt\_controller\_enable* carrega e habilita as configurações padrão e no modo *Classic*. Já as funções *esp\_bluedroid\_init* e *esp\_bluedroid\_enable* são responsáveis pelo *driver* Bluetooth e todas inicializações pertinente.

A função *bt\_app\_task\_start\_up* cria a rotina de aplicativo Bluetooth de verificação do *status* do mesmo, e a ultima função *bt\_app\_work\_dispatch* configura a aplicação com as configurações carregadas anteriormente. O primeiro parâmetro é uma função *handler* (hdl) o qual é responsável por chamadas de rotinas que atualizam eventos como por exemplo: avanço ou retrocesso de faixa, *pause*, *play*, entre outros. E em meio às rotinas, se encontra *bt\_i2s\_task\_handler* que recebe os dados de áudio e envia ao periférico I2S, que será detalhado adiante. O segundo parâmetro é uma variável de valor zero, que indica as configurações do aplicativo, e os três últimos parâmetros são dispensáveis.

### 3.2.3 Processamento do sinal

A rotina de processamento, que tem o fluxo ilustrado na Figura 12 e denominada *Process\_data*, é chamada dentro da função de aquisição dos sinais por Bluetooth. O dado vem no formato MBS, assim como foi configurado e ilustrado na Figura 11.

Figura 11 – Formado de dado



Fonte: Adaptado de (TEXAS INSTRUMENTS, 2022).

A rotina processa o áudio contido no *buffer* trazida pela função *handler* Bluetooth, contendo no total 1024 amostras. A Listagem 3.4 mostra a rotina de processamento.

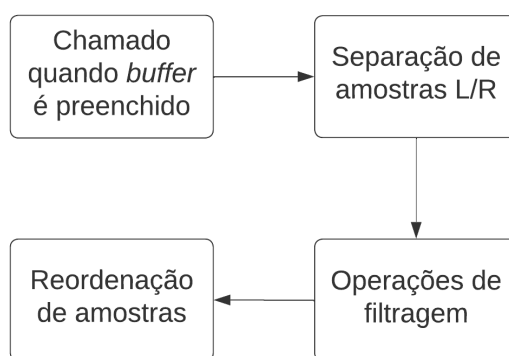
## Listagem 3.4 – Código Processamento

```
1 static float Process_iir (float inSampleF, struct iir_filt
2 *config, uint8_t fs, float *w) {
3
4 float output = (* config).b0[fs] * inSampleF + w[0];
5   w[0] = (* config).b1[fs] * inSampleF - (* config).a1[fs]
6   * output + w[1];
7   w[1] = (* config).b2[fs] * inSampleF - (* config).a2[fs]
8   * output;
9
10  return output;
11 }
12
13 static void process_data(uint8_t *data, size_t item_size){
14
15 int16_t * samples = (int16_t *) data;
16 for (int i=0; i < item_size; i++) {
17     float lsample = (float)*samples;
18     samples++;
19     float rsample = (float)*samples;
20     samples++;
21     float left;
22     left = process_iir(lsample, &shelvLO200, magn[0], wl1);
23     left = process_iir(left, &peaklo4k, magn[0], wl2);
24     left = process_iir(left, &peakmid10k, magn[0], wl3);
25     left = process_iir(left, &peakhi16k, magn[0], wl4);
26     left = process_iir(left, &shelvHI19k, magn[0], wl5);
27
28     float right;
29     right = process_iir(rsample, &shelvLO200, magn[0], wr1);
30     right = process_iir(right, &peaklo4k, magn[1], wr2);
31     right = process_iir(right, &peakmid10k, magn[2], wr3);
32     right = process_iir(right, &peakhi16k, magn[3], wr4);
33     right = process_iir(right, &shelvHI19k, magn[4], wr5);
34
35     *outsample = (int16_t) lwsample;
36     outsample++;
37     *outsample = (int16_t) highsample;
38     outsample++;
```

39 }

O *buffer* contém as amostras na sequência de sinal esquerdo e direito, respectivamente, como mostra a Figura 11, e são apontadas nas linhas 6 e 8 da Listagem 3.4 pelos ponteiros e processadas em sequência, passando pela chamada da rotina *Process\_iir* e sendo atribuídos às variáveis locais *left* e *right*. Por fim, elas são realocadas ao *buffer* em suas devidas posições através do ponteiro *outsample*, como pode ser compreendido na linha 35 e 37. Cada canal passa pelos cinco filtros, pois são sinais estéreos, e o controle de equalização são cinco bandas de frequência.

Figura 12 – Fluxograma básico da rotina de processamento



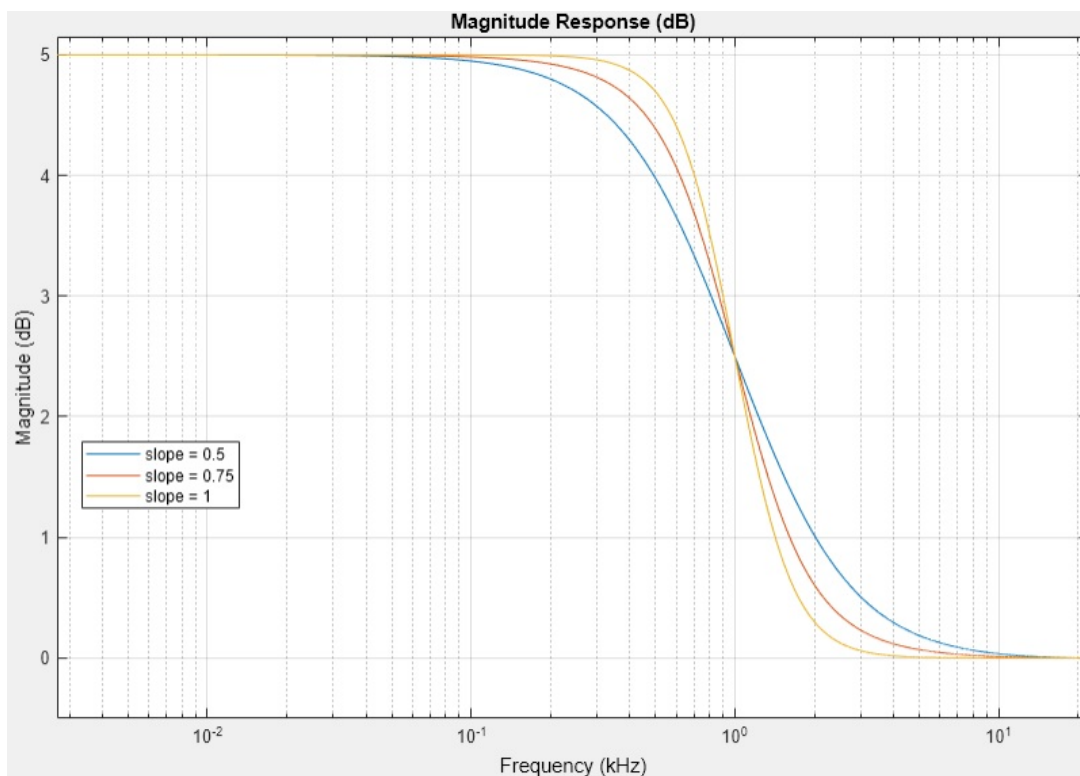
Fonte: Autoria própria (2022).

A rotina *Process\_iir* é onde acontecem as operações no sinal, a qual foi adaptada a partir de uma função disponibilizada pela Espressif System: *dsps\_biquad\_f32\_ansi*. Esta função traz a estrutura na forma direta I implementada, e foi adaptada para formato direta II transposta. A forma direta I possui baixa complexidade e um número significativo de operações, o que gera baixo custo computacional, porém, está sujeito à perdas de significância gerando assim uma instabilidade numérica, além de exigir  $2N$  elementos de memória. A direta forma II transposta exige  $2N$  em número de multiplicações,  $2N - 2$  adições, e  $N$  variáveis para armazenamento.

#### 3.2.4 Banco de filtros

Foram estudados modelos de filtros para escolher a melhor implementação e funcionamento no espectro de frequência. Nas bordas do espectro foram escolhidos filtros do tipo *Shelving*, um filtro ajustável através de parâmetros e que altera o ganho para além da frequência selecionada de forma igual, como ilustrado na Figura 13. Pode ser considerado filtro de borda, pois, *low-shelving* altera o ganho abaixo da frequência de interesse, e o *high-shelving* acima da frequência.

Figura 13 – Resposta magnitude do filtro low-shelving



Fonte: Adaptado de MATLAB (2022).

Com modelo de filtro de borda definido, foram coletados os coeficientes utilizando o *software* Matlab, onde a frequência selecionada *low* foi em  $200\text{ Hz}$  e *high*  $16\text{ kHz}$ . A alteração de ganho em passos de  $1\text{ dB}$  e um intervalo de  $-12\text{ dB}$  a  $+12\text{ dB}$ , resultando em um total de 25 coeficientes para cada filtro. Portanto, o banco de filtros consiste em *structs* contendo todos os coeficientes nos intervalos de ajuste da equalização (ganho).

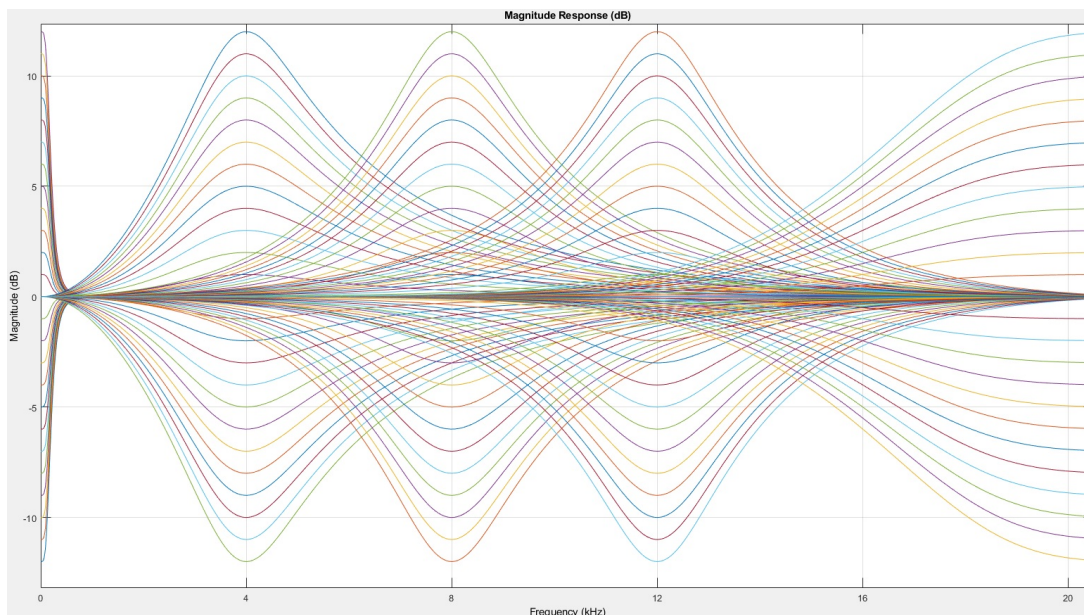
Ao utilizar da função *designShelvingEQ(gain, slope, Fc, type)* do Matlab para o projeto do filtro, é retornado dois vetores de coeficientes no formato SOS (Second-Order Sections) a partir dos parâmetros de entrada, sendo eles: o *gain* aumenta ou atenua a magnitude do sinal; *slope* que altera o tamanho do transiente no domínio da frequência; *Fc* é a frequência de interesse em *Hz*; *type* define se é *low* ou *high shelving*.

Para o meio do espectro, o filtro do tipo *peak* é o que mais se adéqua ao modelo, pois, é possível o controle do ganho, controle sobre a largura de banda do filtro e ordem. Este filtro abre a possibilidade de controlar várias faixas de frequência com precisão, onde cada um é responsável por uma banda. Deste modo, foram modelados três filtros *peak*, que atuam em diferentes faixas a fim de controlar a magnitude de todo espectro audível.

O filtro *peak* foi projetado pela função *designParamEQ(N, gain, Fc, Bw)* no Matlab e retorna duas matrizes totalizando  $2N + 1$  quantidade de coeficientes. Como a função de processamento é de segunda ordem, este filtro também possui ordem 2. Esta função traz liberdade de escolher um valor de ganho em *dB* positivo, além da escolha da largura do filtro,

trazendo maior abrangência no espectro. Foram projetados três destes filtros para se ter esta abrangência, ilustrado na Figura 14, nas frequências centrais  $4kHz$ ,  $8kHz$  e  $12kHz$ , com uma largura de banda de  $4kHz$ .

**Figura 14 – Resposta magnitude dos 5 filtros sobrepostos**



Fonte: Autoria própria (2022).

Ao final, o banco de filtros são separados em cinco *structs*, sendo duas *shelving* e três *peaking*, onde esta armazena cinco variáveis de coeficientes ( $a_1$ ,  $a_2$ ,  $b_0$ ,  $b_1$  e  $b_2$ ) com detalhes na Tabela 1. Cada filtro tem um intervalo de ganho de  $25\text{ dB}$ , conseqüentemente, é gerado um total de 125 coeficientes.

**Tabela 1 – Especificações dos filtros**

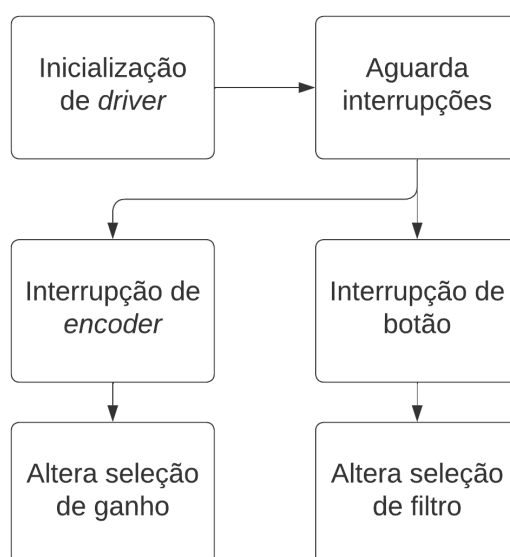
nº	Filtro	Frequência [Hz]	Faixa	Intervalo [dB]
1	<i>shelving</i>	200	baixas	-12 a 12
2	<i>peaking</i>	4000	médias baixas	-12 a 12
3	<i>peaking</i>	8000	médias	-12 a 12
4	<i>peaking</i>	12000	médias altas	-12 a 12
5	<i>shelving</i>	16000	altas	-12 a 12

Fonte: Autoria própria (2022).

### 3.2.5 Interfaceamento por *encoder*

Para alteração de ganho em cada filtro se faz necessário um meio de interação entre o usuário e o dispositivo, e o método clássico para este interfaceamento são os potenciômetros, utilizado em equalizadores analógicos. Sendo um sistema digital, será utilizado um *encoder*, devido ao seu valor de indicação não ser fixo, pois, é possível manipular seu registrador.

Figura 15 – Fluxograma básico da rotina de interfaceamento



Fonte: Autoria própria (2022).

Para este periférico, a ESP-IDF também oferece uma biblioteca, denominada *rotary\_encoder\_percent\_ec11* com o seguinte fluxo na Figura 15, podendo instanciar rotinas de leitura para a obtenção dos valores trazidas pelo *encoder*, e esta função é instanciada junto da função de processamento. A rotação do eixo altera a magnitude de um único filtro para amplificação ou atenuação, dependendo do sentido. O botão integrado tem como função alterar o filtro que está sendo manipulado pelo *encoder*.



## 4 RESULTADOS E DISCUSSÃO

Sendo um dispositivo de equalização de áudio, as análises para avaliação de qualidade foram feitas com sinais de áudio. Uma avaliação objetiva pode ser feita através de medições com entradas definidas para cada frequência utilizada nos filtros. Foram usados como referência, os sinais com mesma frequência no *software* Matlab e para, posteriormente, comparar os resultados.

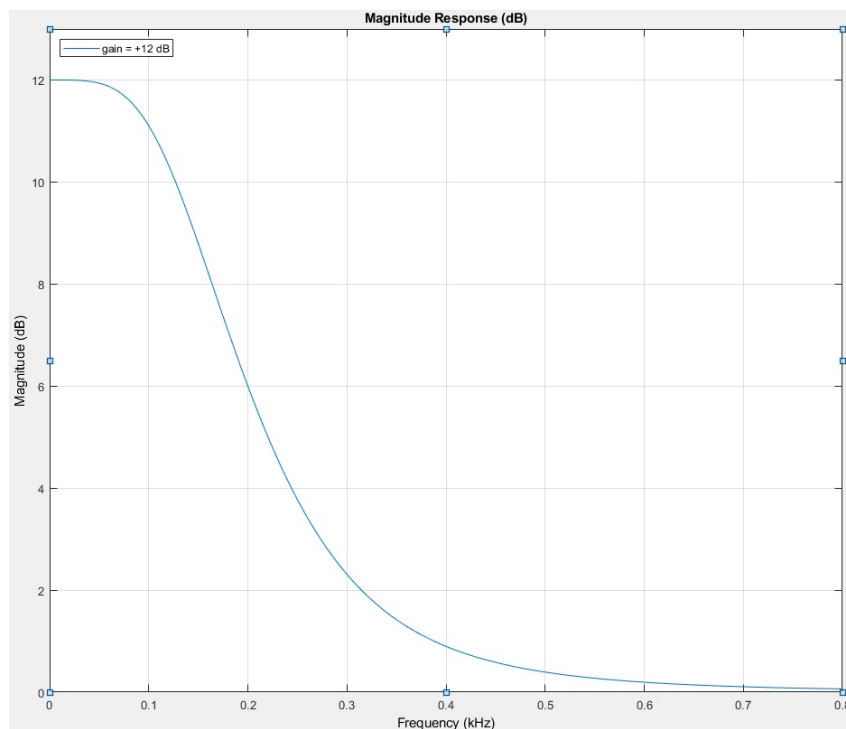
### 4.1 ANÁLISE DE DESEMPENHO

Para a avaliação objetiva foi coletado a resposta magnitude dos filtros projetados no Matlab e submetido um sinal em sua respectiva frequência para obtenção do gráfico. Fazer uma avaliação para cada filtro em cada valor de ganho é inviável, devido à sua grande quantidade, por este motivo foram avaliados três filtros e para cada um deles feito mais dois testes em frequências diferentes.

Para análise e a coleta do sinal do dispositivo, a conexão Bluetooth ocorreu com um celular, e o sinal com as mesmas características foram usadas. A captura dos sinais foram feitas por osciloscópio, onde um canal passa pelo processo de equalização e o outro não.

#### 4.1.1 Análise do filtro *low-shelving*

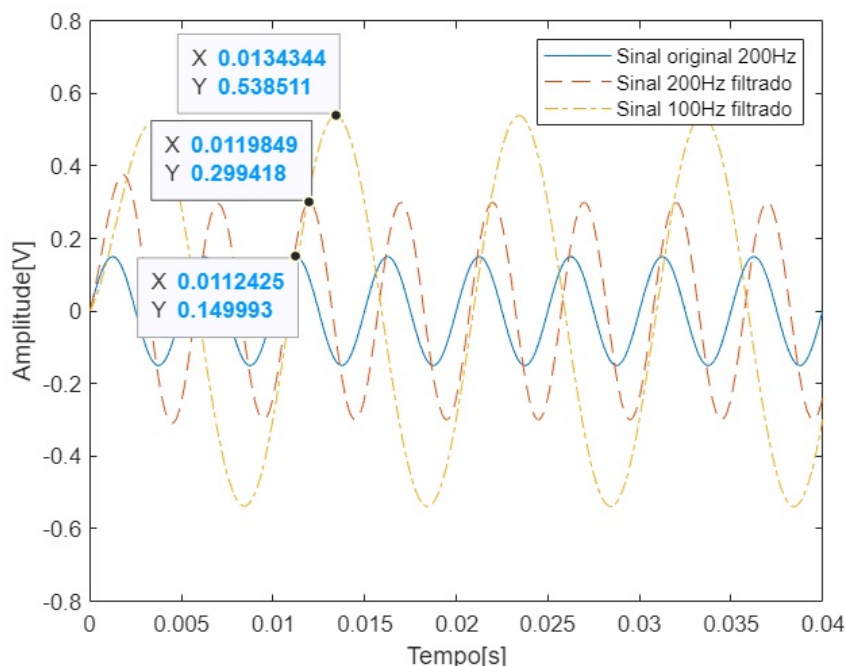
**Figura 16 – Resposta em frequência do filtro *low-shelving***



Fonte: Autoria própria (2022).

O primeiro filtro testado é o *low-shelving* com ganho em +12 dB. Neste, foram realizados testes em 200 Hz e 100 Hz, onde o primeiro se encontra em um ganho de, aproximadamente, +6 dB e o segundo com +11 dB de ganho, segundo a Figura 16. Nesta simulação foram gerados dois sinais com pico de 150 mV utilizando no formato sinusoidal e com as frequências citadas anteriormente. A Figura 17 contém dois sinais filtrados e o terceiro sinal original para referência de amplitude.

**Figura 17 – Comportamento simulado do filtro low-shelving**



Fonte: Autoria própria (2022).

Os sinais apresentados na resposta de simulação acima trazem seus respectivos valores de amplitude máxima indicado pelo caractere *Y*, e se convertidos a unidade *decibel* (dB) apresentam as seguintes medidas:

**Tabela 2 – Conversão para decibel do primeiro teste simulado**

u.m.	Sinal original 200 Hz	200 Hz filtrado	100 Hz filtrado
Tensão elétrica [V]	0.149993	0.299418	0.538511
Decibel [dB]	-14.26	-8.26	-3.12

Fonte: Autoria própria (2022).

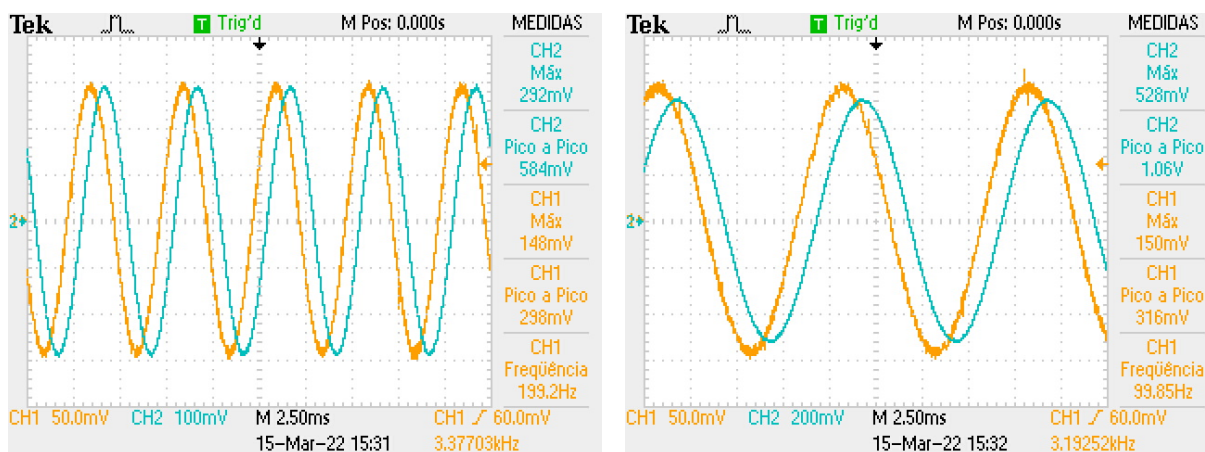
As diferenças em decibéis entre os sinais correspondem a resposta esperada trazida pela Figura 16.

Através do osciloscópio foi capturado o sinal equalizado pela ESP32, e pela Figura 18, apresenta os mesmos três sinais testados no Matlab. O canal 1 (CH1) do equipamento representa o sinal original, já o canal 2 (CH2) representa o sinal filtrado pelo *low-shelving* e

ambos com frequência em  $200\text{ Hz}$ , na Figura 18a. Já a Figura 18b apresenta o sinal para referência de amplitude no CH1, e o sinal filtrado capturado pelo CH2d ambos em  $100\text{ Hz}$ .

De acordo com a interface de unidade de medida, onde o indicador 'Máx' é o valor de pico de seu respectivo canal (CH1 ou CH2), os valores em amplitude máxima do teste prático e a simulação são próximos, indicando a funcionalidade do sistema equalizador prático.

**Figura 18 – Captura do filtro low-shelving implementado**



(a) Sinal de  $200\text{ Hz}$

(b) Sinal de  $100\text{ Hz}$

Fonte: Autoria própria (2022).

Com os valores da Tabela 3, é possível notar a diferença mínima entre os valores ao compará-los. Para ambos os casos, ao comparar o sinal de  $200\text{ Hz}$  original com o de mesma frequência processado, a diferença em decibel é aproximadamente  $+6\text{ dB}$  de amplificação. O sinal em  $100\text{ Hz}$  de mesma amplitude em comparação com original, o ganho é de  $+11\text{ dB}$  de amplificação. A mínima diferença entre os valores teóricos e práticos podem se dar ao erro de aferição do equipamento devido ao ruído.

**Tabela 3 – Conversão para decibel do primeiro teste prático**

u.m.	Original 200 Hz	200 Hz filtrado	Original 100 Hz	100 Hz filtrado
Tensão [V]	0.148	0.292	0.15	0.528
Decibel [dB]	-14.38	-8.48	-14.25	-3.33

Fonte: Autoria própria (2022).

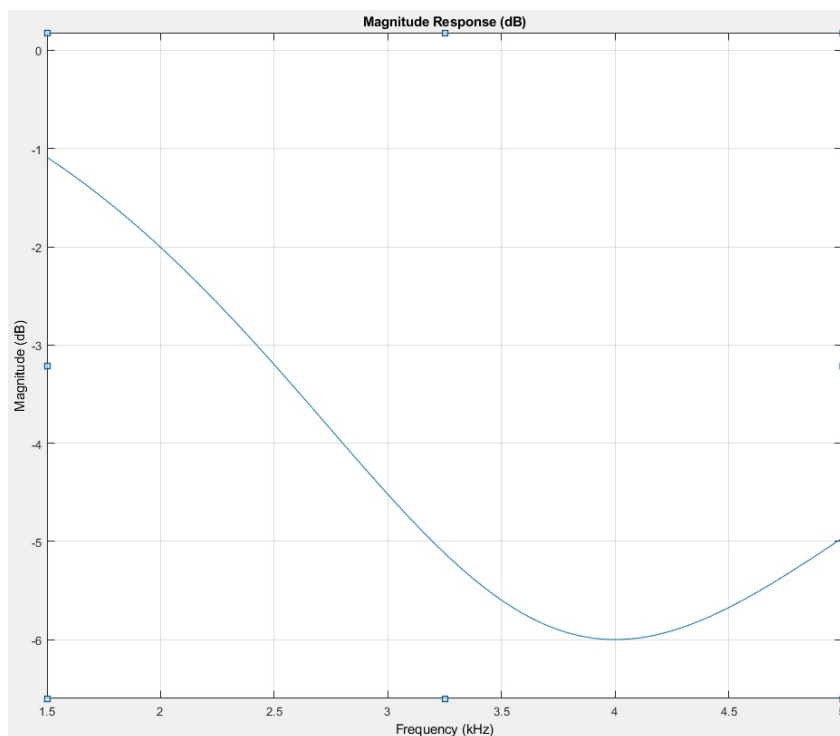
#### 4.1.2 Análise do filtro *peaking*

O segundo teste foi feito para o filtro *peaking* na frequência central de  $4\text{ kHz}$  com atenuação de  $-6\text{ dB}$ . A curva da resposta em magnitude da Figura 19 nos dá uma ideia de quais valores de atenuação é esperada em cada frequência.

A simulação foi realizada com sinal de  $150\text{ mV}$  em valor de pico nas frequências de  $4\text{ kHz}$  e  $2.5\text{ kHz}$ , onde se seguiu o padrão de um sinal estar na frequência de interesse e o

outro na região da curva de atuação do filtro. Nestas frequências, os valores esperados de atenuação são de  $-6\text{ dB}$  para  $4\text{ kHz}$  e aproximadamente  $-3.2\text{ dB}$  em  $2.5\text{ kHz}$ .

**Figura 19 – Resposta em frequência do filtro Peaking**



**Fonte: Autoria própria (2022).**

A Figura 20 mostra o sinal original para referência de amplitude, junto aos outros dois sinais filtrados conforme a legenda e seus valores de pico indicado pela variável  $Y$ . Os valores de pico de cada sinal convertidos em decibéis para melhor comparação são:

**Tabela 4 – Conversão para decibéis do segundo teste teórico**

u.m.	Original 4 kHz	4 kHz filtrado	2.5 kHz filtrado
Tensão [V]	0.149981	0.0749031	0.105258
Decibel [dB]	-14.26	-20.30	-17.34

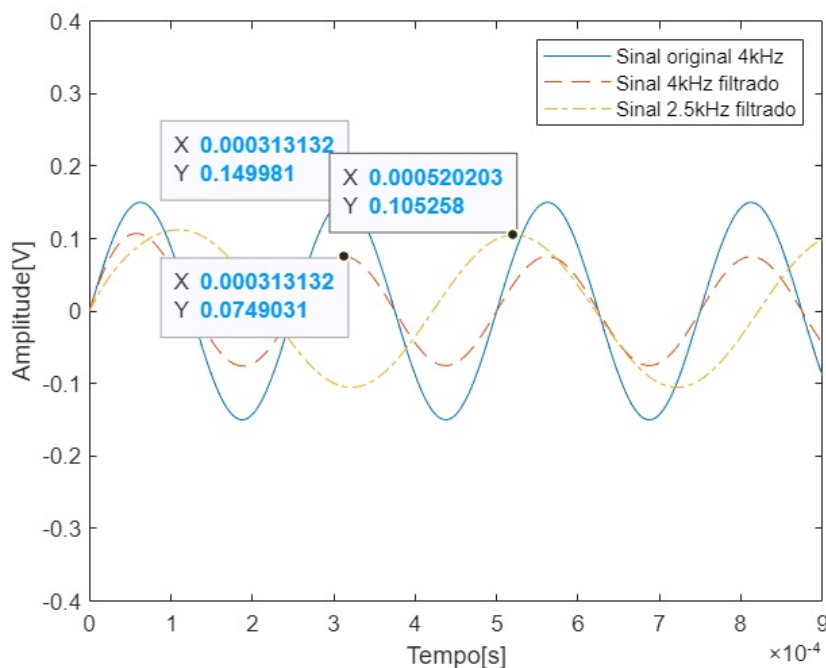
**Fonte: Autoria própria (2022).**

É possível verificar então que o comportamento teórico e a resposta esperada da simulação estão de acordo.

Como padronizado na captura dos sinais por osciloscópio na Figura 21a, o CH1 é responsável pelo sinal original em  $150\text{ mV}$  e o CH2 o sinal filtrado com diferente amplitude, onde neste caso é atenuado. E na Figura 21b representado pelo CH1 um sinal para referência, e no CH2 o sinal de  $2.5\text{ kHz}$  filtrado.

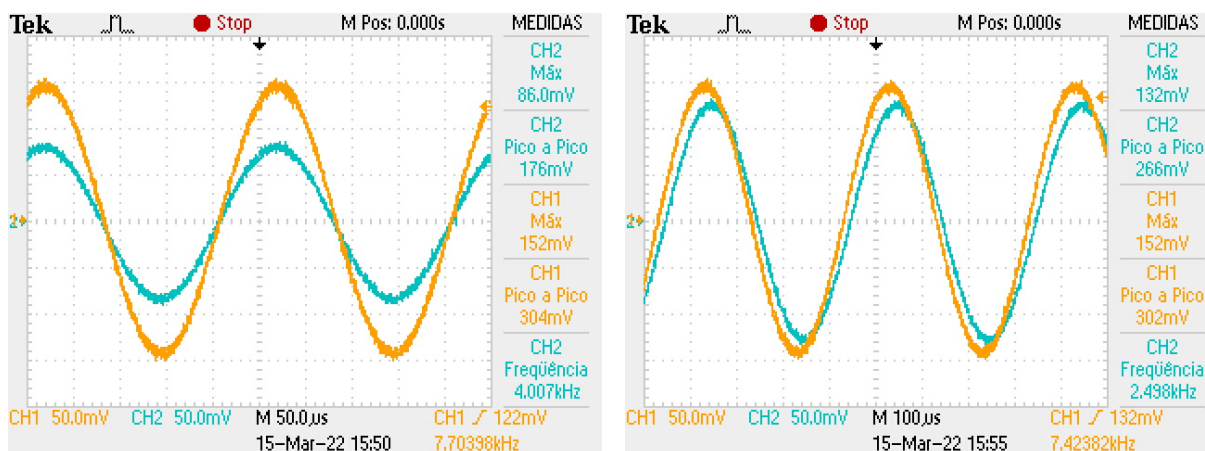
Os valores de pico dos sinais estão de acordo com o esperado pelo teste teórico, havendo uma diferença mínima. A representação do sinal com a linha mais acentuada nas Figuras 21 representa um ruído de *ripple*, o que trás uma medição com erro.

Figura 20 – Comportamento simulado do filtro peaking



Fonte: Autoria própria (2022).

Figura 21 – Captura do filtro peaking implementado



(a) Sinal de 4000 Hz

(b) Sinal de 2500 Hz

Fonte: Autoria própria (2022).

Os valores de magnitude de cada sinal mostram erros que podem ser consideráveis, conforme a Tabela 5. O sinal de 4 kHz teve uma diferença de aproximadamente  $-1$  dB do esperado. Já o segundo sinal filtrado apresenta um erro de  $-2$  dB.

Tabela 5 – Conversão para decibel do segundo teste prático

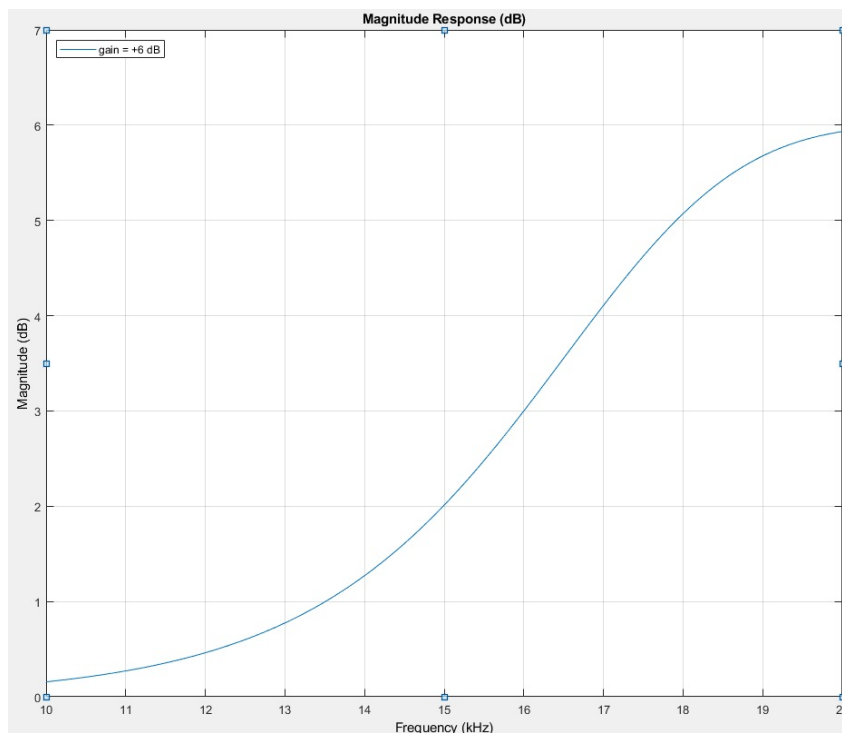
u.m.	Original 4 kHz	4 kHz filtrado	Original 2.5 kHz	2.5 kHz filtrado
Tensão [V]	0.152	0.86	0.152	0.132
Decibel [dB]	-14.15	-19.10	-14.15	-15.37

Fonte: Autoria própria (2022).

### 4.1.3 Análise do filtro *high-shelving*

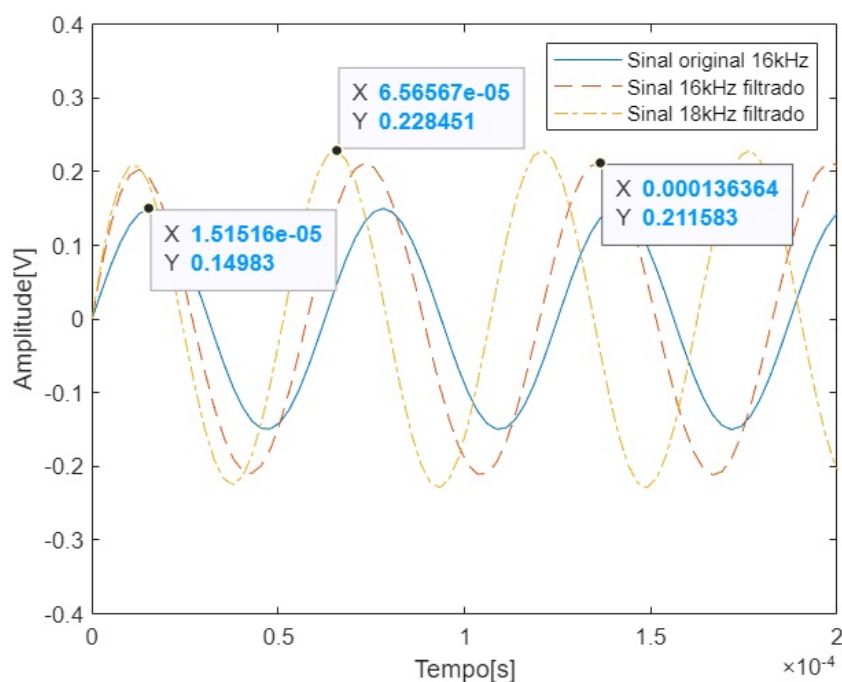
O terceiro e último teste foi feito para o filtro *high-shelving* que possui a curva de resposta na frequência representada na Figura 22, o qual possui o valor de ganho em +6 dB.

**Figura 22 – Resposta em frequência do filtro *high-shelving***



Fonte: Autoria própria (2022).

**Figura 23 – Comportamento simulado do filtro *high-shelving***



Fonte: Autoria própria (2022).

Para este filtro, foram realizados testes, primeiro em  $16\text{ kHz}$  o qual é a frequência de projeto do filtro, e o segundo em  $18\text{ kHz}$  de frequência que também está na região do filtro. Os valores de ganhos esperados para cada uma são:  $+3\text{ dB}$  e  $+5\text{ dB}$ , respectivamente. Junto ao teste do *low-shelving*, nota-se que na frequência de interesse, a magnitude está sempre a metade do valor projetado.

Nesta última simulação, segue o mesmo padrão com três sinais identificados, sendo o original para referência de amplitude, e outros dois com em resposta a um único filtro.

De acordo com os dados do gráfico na Figura 23, os valores em magnitude são:

**Tabela 6 – Conversão para decibel do terceiro teste teórico**

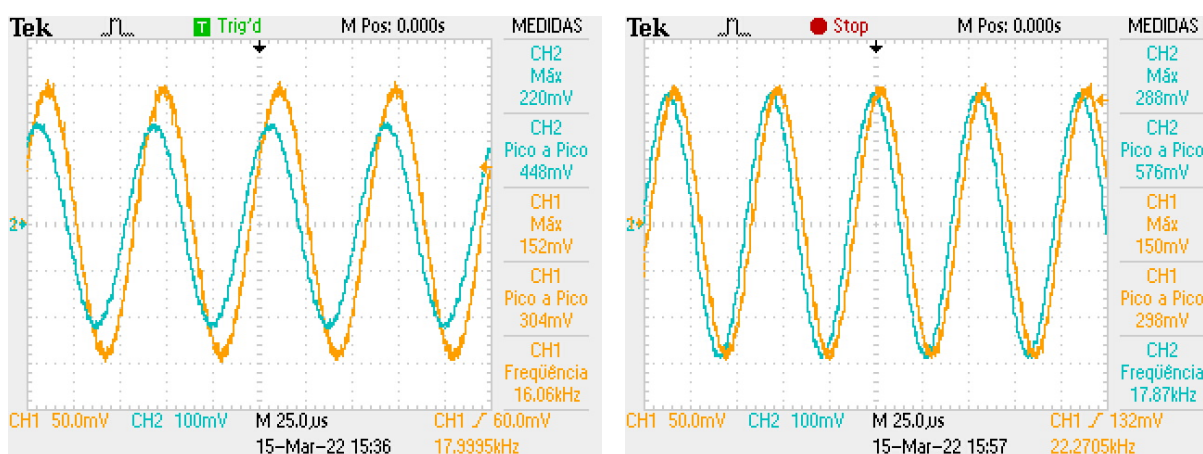
u.m.	Sinal original 16kHz	16kHz filtrado	18kHz filtrado
Tensão [V]	0.14983	0.211583	0.228451
Decibel [dB]	-14.27	-11.27	-10.61

Fonte: Autoria própria (2022).

De acordo com a Tabela 6, os valores de diferenças em magnitude entre os sinais estão perto do esperado. O sinal de  $16\text{ kHz}$  está em seu resultado esperado,  $+3\text{ dB}$  de amplificação, porém, o sinal de  $18\text{ kHz}$  está com  $+4\text{ dB}$  de amplificação ( $1\text{ dB}$  abaixo do esperado).

Nesta Figura 24a, também foi seguido o padrão dos anteriores, com CH1 trazendo sinal sem filtragem, CH2 com sinal de mesma frequência que o primeiro, porém, de maior amplitude, e na Figura 24b, CH1 sendo sinal de referência e o CH2 traz o sinal com diferença na amplitude. Também é possível observar linha mais larga dos sinais, indicando presença de ruído no sinal.

**Figura 24 – Captura do filtro high-shelving implementado**



(a) Sinal de  $16\text{ kHz}$

(b) Sinal de  $18\text{ kHz}$

Fonte: Autoria própria (2022).

**Tabela 7 – Conversão para decibel do terceiro teste prático**

u.m.	Original 16 kHz	16 kHz filtrado	Original 18 kHz	18 kHz filtrado
Tensão [V]	0.152	0.220	0.150	0.288
Decibel [dB]	-14.15	-10.94	-14.25	-8.71

Fonte: Autoria própria (2022).

Com os valores de magnitude em decibéis, na Tabela 7 é possível perceber o erro dos valores que deveriam atender em relação à magnitude do original. O sinal de 16 *kHz* apresenta um ganho aproximado de +3 *dB* de amplificação e que está dentro do esperado, já o sinal de 18 *kHz* apresenta um ganho de quase +7 *dB* de amplificação, uma diferença de +2 *dB* em relação ao esperado.

#### 4.1.4 Análise e discussão dos resultados

Baseado nos resultados obtidos em testes práticos, os valores de ganho apresentados indicam a magnitude amplificada ou atenuada em relação à referência.

De modo geral, ao observar os resultados, o sistema de equalização está em conformidade com a proposta, comparado com as simulações do *software* Matlab, os resultados foram similares, o que torna válido esta implementação.

Durante o teste do dispositivo, foi verificado também se a equalização está performando da maneira esperada, o qual não apresentou nenhuma falha. Como dito anteriormente, este é o teste subjetivo, que se fez necessário pois existem parâmetros de qualidade que não podem ser medidos de forma objetiva. Em contrapartida, o sistema auditivo humano tem baixa sensibilidade em pequenas diferenças de magnitude pela frequência. Deste modo, o teste subjetivo foi feito variando magnitudes entre as bandas de frequências enquanto é reproduzido um áudio verificando a ocorrência de algum tipo de ruído ou artefatos no sinal de áudio.

**Tabela 8 – Comparativo dos resultados práticos gerais**

Low-shelving			
Teste	Frequência [Hz]	Ganho [dB]	Erro %
1	199.2	+5.9	1.666
2	100	+11.92	8.364
Peaking			
Teste	Frequência [kHz]	Ganho [dB]	Erro %
1	4.007	-4.95	17.5
2	2.510	-1.22	60.645
High-shelving			
Teste	Frequência [kHz]	Ganho [dB]	Erro %
1	16.06	+3.21	7.0
2	18.09	+5.54	10.8

Fonte: Autoria própria (2022).



Já para o cálculo dos valores de erro, foi usado como referência o valor de ganho da curva plotada de cada um dos filtros na figuras de resposta em frequência. Nota-se na Tabela 8, que em todos os testes, os que não eram em frequências de projeto dos respectivos filtros, apresentam um maior erro. Uma das possíveis fontes de incertezas que causam o erro nestes teste pode ser vinculado ao regime transitório em que se encontra as frequências neles teste. Porém este erro não tem significância de modo que insira ruído ou cause alguma distorção a ponto de ser percebida.

## 5 CONCLUSÃO

O interesse inicial em dispositivos de áudio foi o que motivou a pesquisa em trazer à topologia digital um equipamento tradicionalmente analógico. Na busca por trazer um dispositivo compacto e igualmente funcional, optou-se pela implementação em um microcontrolador da Espressif System, o ESP32, que em suas especificações de *hardware* é o modelo mais completo e satisfatório, tanto em questão de periféricos, quanto em velocidade de núcleo e memória. Os outros componentes e tecnologias escolhidas foram baseadas em pesquisas posteriores.

Com estas pesquisas foram compreendidas as necessidades que se fazem para implementação de um equalizador digital, o qual envolvem o estudo de amostragem, design de filtros IIR, implementações em *firmware* as estruturas IIR para operações de filtragem, implementação da API do Bluetooth, conversores DAC, protocolo de comunicação I2S. Portanto, este trabalho apresentou uma metodologia para implementação deste equalizador de baixas dimensões e atende às funcionalidades de um equalizador.

Ao final da montagem de todo *hardware* e ajustes de *firmware*, foram feitos testes objetivos para avaliar o desempenho do sistema. Estes testes consistiram em uma comparação de sinais equalizados e capturados por osciloscópio com uma simulação pelo Matlab, com os mesmos parâmetros de filtragem e sinais. Ao avaliar estes resultados, chegou-se a conclusão que o sistema de equalização tem uma performance satisfatória.

Os erros apresentados não representam uma fonte de ruído, apenas mostra o quão próximo do esperado a curva de ganho se comporta. Estes erros podem ser justificados pela quantização numérica ser insuficiente, e também pela estrutura de filtro implementada que não traz uma robustez numérica e que altera a curva de resposta inicialmente projetada.

A fim de validar o projeto, um teste subjetivo ao final do objetivo foi feito para verificar inconsistências no funcionamento geral do equalizador. Enquanto era reproduzido um sinal de forma ininterrupta e alterado o ganho, não foi detectado nenhum tipo de ruído ou artefatos mesmo em volume alto. O mesmo teste por um período mais longo foi feito a fim de detectar falhas na reprodução em decorrência de erros por *underflow* do Bluetooth, o qual também não teve ocorrência.

Uma possível abordagem para a continuidade do trabalho seria a implementação de cálculo dos coeficientes no chamado *on-the-fly*, onde são calculados em tempo real não sendo necessário armazenamento de coeficientes pré-calculados, e uma rotina é chamada toda vez que é alterado o parâmetro de ganho. Vale também a implementação de outras estruturas de filtro IIR, como por exemplo de espaço de estados, oferecendo uma maior precisão numérica.

## REFERÊNCIAS

BECH, S. Selection and training of subjects for listening tests of soundreproducing equipment. **Audio Eng**, v. 40, p. 590–610, jul, 1992.

BLUETOOTH. **Learn About Bluetooth**. 2022. Disponível em: <https://www.bluetooth.com/learn-about-bluetooth/>. Acesso em: 14 jan. 2022.

BRAGA, N. C. **Eletrônica analógica**. São Paulo: NCB, 2012. E-book Disponível em: [https://docs.google.com/gview?url=http://www.newtoncbraga.com.br/arquivos/ce\\_ea.pdf/](https://docs.google.com/gview?url=http://www.newtoncbraga.com.br/arquivos/ce_ea.pdf/). Acesso em: 07 jul. 2019.

DINIZ, P. S. R.; SILVA, E. A. B. da; NETTO, S. L. **Processamento digital de sinais**. 2. ed. Porto Alegre: Bookman, 2014. 976 p.

DRONESHOW. **Indústria 4.0 impulsiona mercado de trabalho para Engenheiros**. 2019. Disponível em: <https://droneshowla.com/industria-4-0-impulsiona-mercado-de-trabalho-para-engenheiros/>. Acesso em: 07 jul. 2019.

FERNANDES, J. C. **Acústica e Ruído**. 2002. Apostila. Universidade Estadual de São Paulo - UNESP. Disponível em: <http://wwwp.feb.unesp.br/jcandido/acustica/Apostila/Capitulo%2002.pdf/>, 2002. Acesso em: 10 maio 2019.

FLOYD, T. L. **Sistemas digitais**. Tradução: José Lucimar do Nascimento. 9. ed. Porto Alegre: Bookman, 2007. 888 p.

HERRERA, C. G. **Projeto de Sistemas de Processamento Digital de Sinais de Áudio Utilizando Metodologia Científica**. 2004. Dissertação (Mestre em Engenharia Elétrica) - Universidade Federal de Minas Gerais: Belo Horizonte, 2004. Disponível em: <https://www.ppgee.ufmg.br/defesas/486M.PDF/>. Acesso em: 15 abr. 2019.

HSU, H. P. **Sinais e sistemas: 570 problemas totalmente resolvidos**. Tradução: Anatólio Laschuk. 2. ed. Porto Alegre: Bookman, 2012. 495 p.

LATHI, B. P. **Sinais e sistemas lineares**. Tradução: Gustavo Guimarães Parma. 2. ed. Porto Alegre: Bookman, 2007. 856 p.

MADISSETTI, V. K.; WILLIAMS, D. B. **The Digital Signal Processing Handbook**. 1. ed. Bosa Roca: CRC Press LLC, 1998. 1776 p.

MATLAB. **What is MATLAB?** 2022. Disponível em: [https://www.mathworks.com/products/matlab.html?s\\_tid=hp\\_products\\_matlab/](https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab/). Acesso em: 10 jan. 2022.

OPPENHEIM, A. V.; WILLSKY, A. S. **Sinais e sistemas**. Tradução: Daniel Vieira, Rogério Bettoni. 2. ed. São Paulo: Pearson, 2010. 568 p.

ORFANIDIS, S. J. **Signal processing**. Rutgers University: Prentice-Hall, 2010. E-book Disponível em: <https://www.ece.rutgers.edu/~orfanidi/intro2sp/orfanidis-i2sp.pdf/>. Acesso em: 01 mar. 2022.

OSSIA. **Como Funciona o Equalizador**. 2016. Disponível em: <https://ossia.com.br/como-funciona-o-equalizador/>. Acesso em: 11 mar. 2022.

OTT, H. W. **Noise Reduction Techniques in Electronic Systems**. 2. ed. New York: Wiley-Interscience, 1988. 448 p.

ROSA, I. E. E. **Projeto de Equalizador Paramétrico Avançado para Aplicações em MP3**. 2012. Monografia (Tecnólogo em Sistemas de Telecomunicações) - Instituto Federal de Santa Catarina: São José, 2012. Disponível em: [https://wiki.sj.ifsc.edu.br/images/9/97/TCCCSTST007\\_2011.pdf/](https://wiki.sj.ifsc.edu.br/images/9/97/TCCCSTST007_2011.pdf/). Acesso em: 10 mar. 2022.

SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. San Diego: California Technical Publishing, 1999. E-book. Disponível em: [https://www.analog.com/en/education/education-library/scientist\\_engineers\\_guide.html/](https://www.analog.com/en/education/education-library/scientist_engineers_guide.html/). Acesso em: 11 abr. 2019.

SMITH, S. W. **Digital signal processing**. 3. ed. Amsterdam: Newnes, 2003. 650 p.

TEXAS INSTRUMENTS. **PCM5102 Datasheet**. 2022. Disponível em: <https://www.ti.com/lit/gpn/pcm5102/>. Acesso em: 24 fev. 2022.

TORMENA JUNIOR, O. **Processamento digital de sinais**. Notas de aula. Universidade Tecnológica Federal do Paraná - UTFPR: Campo Mourão, 2018.

VALIANTE FILHO, F. **Apostila básica de Audio**. Taboão da Serra: [s.n.], 2004. E-book Disponível em: <http://www.ibam-concursos.org.br/documento/Audio.pdf/>. Acesso em: 10 maio. 2019.

WARPSOUND. **O que é equalizador e quais são os seus tipos: Saiba tudo sobre equalizadores**. 2017. Disponível em: <https://warpsound.com.br/o-que-e-equalizador-e-quais-sao-os-seus-tipos-saiba-tudo-sobre-equalizadores/>. Acesso em: 10 mar. 2022.