



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**LUCAS PEDROSA VALENTE**

**SISTEMA DE ANÁLISE DE CARACTERÍSTICAS DE SINAIS DE ÁUDIO PARA  
CONVERSÃO EM PROTOCOLO MIDI**

**SOBRAL**

**2022**

LUCAS PEDROSA VALENTE

SISTEMA DE ANÁLISE DE CARACTERÍSTICAS DE SINAIS DE ÁUDIO PARA  
CONVERSÃO EM PROTOCOLO MIDI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Marcelo Marques Simões de Sousa

SOBRAL

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- V249s Valente, Lucas Pedrosa.  
Sistema de Análise de Características de Sinais de Áudio para Conversão em Protocolo MIDI / Lucas Pedrosa Valente. – 2022.  
64 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia da Computação, Sobral, 2022.  
Orientação: Prof. Dr. Marcelo Marques Simões de Sousa.
1. Processamento de Sinais. 2. Zero Crossing. 3. Microcontroladores. 4. Análise de Sinais. 5. Áudio. I. Título.

CDD 621.39

---

LUCAS PEDROSA VALENTE

SISTEMA DE ANÁLISE DE CARACTERÍSTICAS DE SINAIS DE ÁUDIO PARA  
CONVERSÃO EM PROTOCOLO MIDI

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Engenharia de  
Computação da Universidade Federal do Ceará,  
como requisito parcial à obtenção do grau de  
bacharel em Engenharia de Computação.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Marcelo Marques Simões de  
Sousa (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Ialis Cavalcante de Paula Júnior  
Universidade Federal do Ceará (UFC)

---

Prof. Me. David Nascimento Coelho  
Universidade Federal do Ceará (UFC)

À minha família, pelo apoio e pela confiança depositada em mim. Sem a atenção e o aporte oferecido por vocês não haveria a possibilidade de ter alçado voos tão altos.

## AGRADECIMENTOS

Primeiramente gostaria de agradecer a todas as pessoas que me fizeram chegar aqui. À minha mãe Marcela por todo o suporte e conselhos dados, ao meu pai Fábio pelo companheirismo e amizade, e principalmente a ambos por terem apoiado todas as escolhas que fiz ao longo da minha vida.

Às minhas tias Teresa e Joana pela convivência e cuidado, sempre sendo atenciosas e prestativas.

Aos professores que tive ao longo do curso, em especial ao professor Dr. Iális Cavalcante de Paula Júnior pela tutoria durante meu tempo de bolsa PET, ao professor Me. David Nascimento Coelho pelo apoio e pelas conversas que tivemos durante a graduação, e principalmente ao professor Dr. Marcelo Marques Simões de Sousa pela excelente orientação, não apenas no âmbito acadêmico, mas para a vida.

Gostaria de agradecer aos meus irmãos Mateus e Larissa pelo companheirismo e apoio que tivemos.

Aos meus avós Geraldo, Eladi, Eugênia e Antônio por todo apoio dado à nossa família dando condições para todos chegarmos onde chegamos.

Aos meus amigos de longa data Vitor, Tursi, PR, Talmag, Gabriel Cintra, Rian, Edno, João Pedro, entre tantos outros que estão guardados no meu coração, pela amizade, companheirismo e apoio ao longo da nossa jornada.

À minha namorada Jhulli por estar do meu lado, me apoiar e confiar em mim durante todos os momentos que passamos, por ser meu porto seguro e minha força em diversos momentos.

Aos meus amigos que cultivei ao longo do curso, em especial aos meus amigos Gabriel Sousa, Gabriel Ribeiro, Felipe, Gilson, Lucas França, Wilson, Stefane e Clara por todos os dias de luta e dias de glória que compartilhamos durante todo o percorrer do curso.

E por último gostaria de agradecer à Tia Cecília, por todo o cuidado e atenção até o último instante, pois sem ela eu sei que não estaria aqui vivendo este momento.

“A música é celeste, de natureza divina e de tal beleza que encanta a alma e a eleva acima da sua condição.”

(Aristóteles)

## RESUMO

Desde sua criação, o protocolo MIDI é amplamente utilizado no mundo da música, sendo implementado principalmente em instrumentos controladores e mesas de som digitais. Seu uso abrange o controle de sintetizadores, instrumentos virtuais, e programas de gravação e edição de áudio. Algumas informações importantes aplicadas no protocolo são mensagens de canal, nota e velocidade. Com o intuito de implementar protocolo MIDI em instrumentos acústicos para controlarmos dispositivos de áudio, podemos aplicar técnicas de análise de características dos sinais produzidos por estes instrumentos, no reconhecimento da frequência fundamental e de amplitude destes sinais. Neste contexto, propomos a aplicação de um algoritmo de reconhecimento de frequência, baseado em zero crossing, para detecção da frequência fundamental, e um conversor A/D para detecção da amplitude do sinal gerado por instrumentos musicais. O sistema se mostrou extremamente eficiente para conversão das características de amplitude e de frequência de sinais de áudio para o protocolo MIDI, podendo ser aplicado em diversos instrumentos para controle de outros dispositivos utilizando o protocolo MIDI.

**Palavras-chave:** Processamento de Sinais. Zero Crossing. Microcontroladores. Análise de Sinais. Áudio.



## ABSTRACT

Since its creation, the MIDI protocol has been widely used in the music world, being implemented mainly in controller instruments and digital mixers. Its use embraces the control of synthesizers, virtual instruments, and audio recording and editing programs. Some important information applied in the protocol are channel, note and speed messages. In order to implement MIDI protocol in acoustic instruments to control audio devices, we can apply techniques for characteristics analysis of the signals produced by these instruments, in the fundamental frequency recognition and amplitude of these signals. In this context, we propose the application of a frequency recognition algorithm, based on zero crossing, to detect the fundamental frequency, and an A/D converter to detect the amplitude of the signal generated by musical instruments. The system proved to be extremely efficient for converting the amplitude and frequency characteristics of audio signals to the MIDI protocol, being able to be applied in several instruments for control of other devices using the MIDI protocol.

**Keywords:** Signal Processing. Zero Crossing. Microcontrollers. Signal Analysis. Audio

## LISTA DE FIGURAS

Figura 1 – Telharmonium. Na esquerda o instrumento e a direita o projeto de um <i>tonewheel</i>	16
Figura 2 – Léon Theremin tocando seu instrumento . . . . .	17
Figura 3 – Sintetizador Modular . . . . .	18
Figura 4 – Exemplo de uma DAW (Reaper) . . . . .	19
Figura 5 – Distância entre as notas naturais . . . . .	23
Figura 6 – Interfaceamento MIDI . . . . .	26
Figura 7 – Forma de onda de uma mensagem MIDI . . . . .	27
Figura 8 – Teclado Controlador MIDI . . . . .	29
Figura 9 – Tipos mais comuns de controladores percussivos. Na esquerda uma bateria eletrônica e a direita um controlador por <i>pads</i> , ambos da marca Roland . . .	30
Figura 10 – Guitarras MIDI. Na esquerda uma guitarra MIDI controladora da marca Casio e a direita uma guitarra com dispositivo hexafônico da marca Roland .	30
Figura 11 – Controlador MIDI de Sopros . . . . .	31
Figura 12 – Microfones de diversos tipos . . . . .	32
Figura 13 – Captadores magnéticos mais comuns. Na esquerda um captador <i>single coil</i> (bobina simples) e a direita um captador <i>humbucker</i> (bobina dupla) . . . . .	33
Figura 14 – Captador Piezo . . . . .	33
Figura 15 – Captador Óptico . . . . .	34
Figura 16 – Gráficos de filtros de frequência ideais . . . . .	35
Figura 17 – Filtro RC passa-baixa . . . . .	36
Figura 18 – Diagrama de filtro passa-baixa de ordens distintas . . . . .	37
Figura 19 – Processamento em tempo discreto de sinais de tempo contínuo . . . . .	37
Figura 20 – Diagrama elaborado para modelagem do sistema . . . . .	40
Figura 21 – Filtro <i>Butterworth</i> passa-alta genérico de segunda ordem . . . . .	41
Figura 22 – Relação entre frequência e nota MIDI . . . . .	45
Figura 23 – Relação entre amplitude e <i>velocity</i> MIDI . . . . .	47
Figura 24 – Circuito do filtro <i>Butterworth</i> de segunda ordem para 800 Hz . . . . .	50
Figura 25 – Resposta do filtro <i>Butterworth</i> . . . . .	50
Figura 26 – Retificador de meia onda simulado no <i>software</i> Proteus . . . . .	51
Figura 27 – Retificador de meia onda adicionado ao Filtro . . . . .	52
Figura 28 – Circuito comparador simulado no <i>software</i> Proteus . . . . .	52

Figura 29 – Detecções irregulares do <i>zero cross</i> . . . . .	53
Figura 30 – Formas de onda da entrada do sinal, sinal filtrado e da saída do circuito comparador . . . . .	53
Figura 31 – Circuito completo de pré-processamento . . . . .	54
Figura 32 – Diagrama dos pinos do PIC18F47K40 . . . . .	55
Figura 33 – Respostas dos algoritmos de conversão de frequência e amplitude para protocolo MIDI . . . . .	56
Figura 34 – Esquemático completo do sistema de pré-processamento . . . . .	57
Figura 35 – Circuito completo do sistema desenvolvido . . . . .	57

## LISTA DE TABELAS

Tabela 1 – Amplitude e escalas dB . . . . .	25
Tabela 2 – Relação entre notas musicais, notas midi e frequências . . . . .	28

## LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog to Digital Converter</i>
APF	<i>All Pass Filter</i>
BPF	<i>Band Pass Filter</i>
DAC	<i>Digital to Analog Converter</i>
DAW	<i>Digital Audio Workstation</i>
DSP	<i>Digital Signal Processor</i>
EUSART	<i>Enhanced Universal Synchronous Asynchronous Receiver/Transmitter</i>
HPF	<i>High Pass Filter</i>
LPF	<i>Low Pass Filter</i>
MIDI	<i>Musical Instrument Digital Interface</i>
PCM	<i>Pulse Code Modulation</i>
SNR	<i>Signal-Noise Ratio</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USART	<i>Universal Synchronous Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>
VSTi	<i>Virtual Studio Technology instrument</i>
ZCD	<i>Zero Cross Detection</i>

## LISTA DE SÍMBOLOS

$A$	Unidade de corrente elétrica (Ampere)
$bps$	Bit por segundo
$C$	Capacitância
$dB$	Decibel
$F$	Unidade de capacitância (Faraday)
$Hz$	Unidade de frequência (Hertz)
$I$	Corrente elétrica
$N$	Nota MIDI
$oct$	Oitava
$R$	Resistência
$s$	Segundo
$t$	Tempo
$T$	Período
$V$	Tensão
$\Omega$	Unidade de resistência (Ohms)
$\omega$	Frequência angular

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Objetivos</b>	<b>21</b>
<b>1.1.1</b>	<i>Objetivo Geral</i>	<b>21</b>
<b>1.1.2</b>	<i>Objetivos Específicos</i>	<b>21</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>22</b>
<b>2.1</b>	<b>Teoria Musical</b>	<b>22</b>
<b>2.2</b>	<b>Psicoacústica</b>	<b>24</b>
<b>2.3</b>	<b>Protocolo MIDI</b>	<b>25</b>
<b>2.3.1</b>	<i>Mensagens MIDI</i>	<b>27</b>
<b>2.3.2</b>	<i>Controladores MIDI</i>	<b>29</b>
<b>2.4</b>	<b>Processamento de Sinais</b>	<b>31</b>
<b>2.4.1</b>	<i>Captação de sinais</i>	<b>32</b>
<b>2.4.2</b>	<i>Filtragem de sinais</i>	<b>34</b>
<b>2.4.3</b>	<i>Conversão AD/DA</i>	<b>37</b>
<b>2.4.4</b>	<i>Técnicas de análise de frequências</i>	<b>38</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>39</b>
<b>3.1</b>	<b>Requisitos</b>	<b>39</b>
<b>3.2</b>	<b>Modelagem do sistema</b>	<b>39</b>
<b>3.3</b>	<b>Materiais</b>	<b>40</b>
<b>3.4</b>	<i>Filtro Butterworth</i>	<b>41</b>
<b>3.5</b>	<i>Zero Crossing</i>	<b>42</b>
<b>3.5.1</b>	<i>Implementação via software</i>	<b>42</b>
<b>3.5.2</b>	<i>Implementação via hardware</i>	<b>42</b>
<b>3.6</b>	<b>Conversão frequência para MIDI</b>	<b>43</b>
<b>3.7</b>	<b>Conversão Amplitude para MIDI</b>	<b>45</b>
<b>3.8</b>	<b>Comunicação Serial</b>	<b>48</b>
<b>4</b>	<b>RESULTADOS</b>	<b>49</b>
<b>4.1</b>	<b>Filtragem da entrada do Sinal</b>	<b>49</b>
<b>4.2</b>	<b>Retificador de Meia Onda</b>	<b>51</b>
<b>4.3</b>	<b>Circuito Comparador</b>	<b>52</b>

4.4	<b>Desenvolvimento do <i>Firmware</i></b> . . . . .	54
4.5	<b>Integração do <i>Hardware</i></b> . . . . .	56
5	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	58
	<b>REFERÊNCIAS</b> . . . . .	59
	<b>APÊNDICES</b> . . . . .	61
	<b>APÊNDICE A – Terminologia</b> . . . . .	61
	<b>APÊNDICE B – Código de conversão de período para MIDI</b> . . . . .	62
	<b>APÊNDICE C – Código de conversão de amplitude para MIDI</b> . . . . .	64

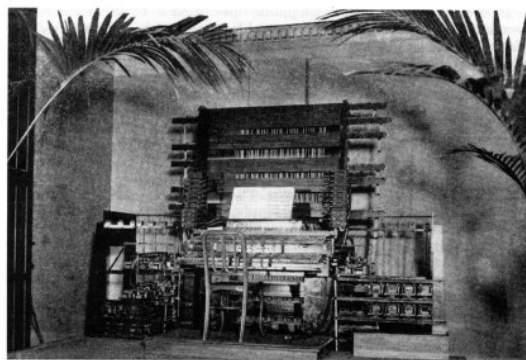


## 1 INTRODUÇÃO

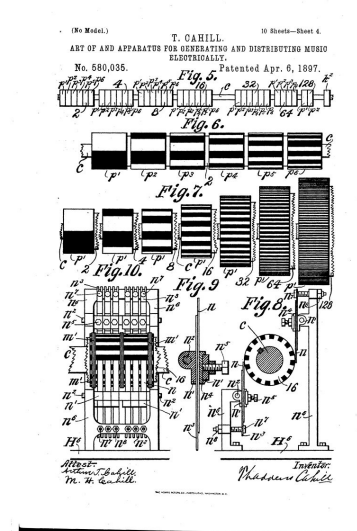
Ao longo da história da música podemos verificar sua relação com o desenvolvimento tecnológico. Zuben (2004) define tecnologia como o estudo das técnicas de um ou mais ofícios ou domínios da atividade humana. Hoje em dia o termo é mais utilizado para significar o conjunto de técnicas que envolvem conhecimentos modernos e complexos. Dentre estes conhecimentos, podemos destacar o desenvolvimento de tecnologias relacionadas à eletrônica. Com este desenvolvimento, surgiram novas formas de se construir instrumentos, como detalhado por Gohn:

“Podemos considerar que a inovação tecnológica de maior destaque para a evolução da eletrônica na música foi o telefone, criado por Alexander Graham Bell, em 1876. Ela comprovou que o som poderia ser transformado em sinais elétricos e vice-versa, estreitando a relação entre tecnologia e arte e abrindo caminhos para a gravação do som e para os meios de comunicação de massa.” (GOHN, 2001)

Figura 1 – Telharmonium. Na esquerda o instrumento e a direita o projeto de um *tonewheel*



(a) Telharmonium



(b) Tonewheel

Fonte – (WIKIPEDIA, 2021a), (WIKIPEDIA, 2021c)

O primeiro instrumento eletrônico a ser idealizado e patenteado foi o *telharmonium*. Seu sistema consistia em utilizar *tonewheels*<sup>1</sup> para gerar os sons, aplicando síntese aditiva<sup>2</sup>, o que permitia explorar uma grande variedade de timbres. Este sistema foi utilizado como base

<sup>1</sup> Tonewheels são sistemas que utilizam um captador magnético (ver 2.4), discos de diversos tamanhos com ondulações e um motor AC para produzir o som.

<sup>2</sup> Síntese aditiva se trata da manipulação de timbres aplicando a soma de vários sinais de características semelhantes ou diferentes.

para elaborar o órgão Hammond na primeira metade do século XX. Segundo Dearling (1996), essa invenção não foi valorizada por estar à frente de seu tempo, porém anos mais tarde outros pesquisadores retomaram muitas das suas idéias (apud GOHN, 2001). Nas Figuras 1a e 1b podemos verificar, respectivamente, o instrumento e a folha de patente do sistema de *tonewheel*.

O Theremin, criado e desenvolvido em 1920 pelo físico Léon Theremin, foi o primeiro instrumento eletrônico a ser fabricado em escala comercial. Esse instrumento consiste em duas antenas: uma para controlar a dinâmica e outra para controlar a nota do som. Sua criação nasceu do investimento na Rússia relacionado ao estudo de sensores por aproximação e trouxe como inovação a não necessidade de contato físico entre o instrumento e o instrumentista (THE NY THEREMIN SOCIETY, 2005). Isso é possível pelo efeito capacitivo entre o corpo do instrumentista e as antenas do instrumento. Esse instrumento assemelha-se aos circuitos utilizados em rádios para captação do sinal de radiofrequência como o princípio heteródino<sup>3</sup>, porém aplicado na síntese do seu som. Na Figura 2 podemos ver Léon Theremin tocando seu instrumento homônimo.

Figura 2 – Léon Theremin tocando seu instrumento



Fonte – (THE NY THEREMIN SOCIETY, 2005)

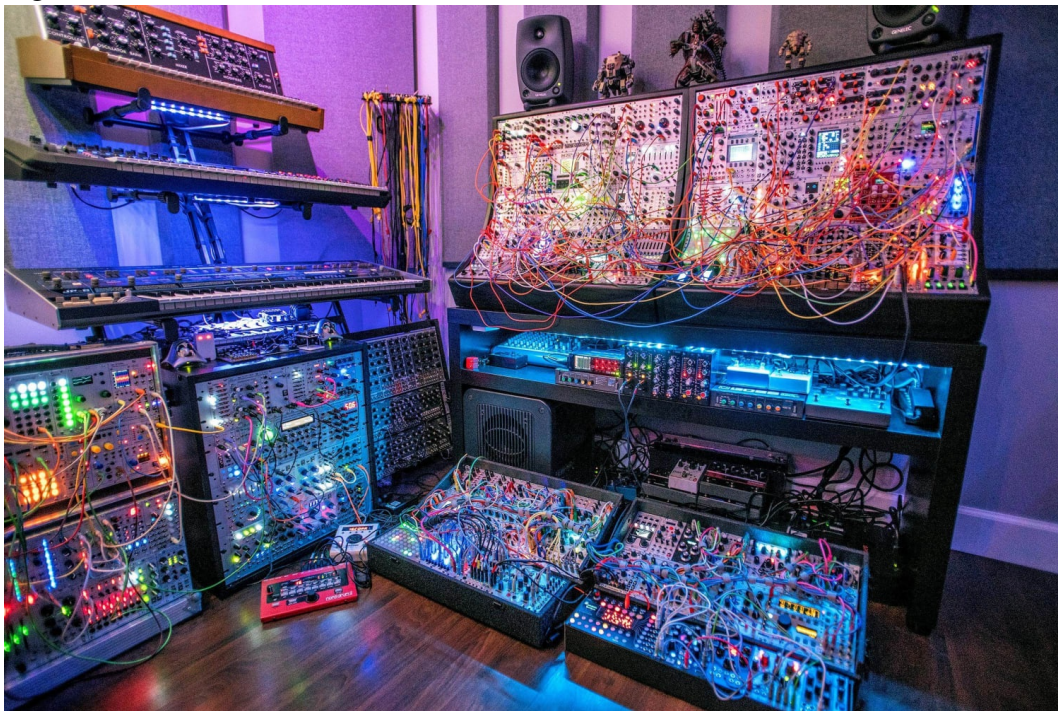
<sup>3</sup> Heteródino consiste em um sistema que possui um circuito somador que recebe o sinal de um oscilador e o sinal externo para produzir um som que é relacionado às duas entradas

Os sintetizadores são instrumentos musicais que utilizam meios eletrônicos para produzir e manipular timbres (RATTON, 2005). Estes instrumentos surgiram na década de 60 e revolucionaram a forma de se fazer música devido as suas vastas possibilidades de criação e manipulação do áudio. Apesar de sua popularização o sistema ainda possuía algumas limitações quanto a polifonia<sup>4</sup>, pois os dispositivos na época eram bastante caros. Como citado por Ratton:

“Atualmente quase todos os sintetizadores utilizam tecnologia “digital”, com chips microprocessadores controlando todo o processo de síntese e demais funções do equipamento, mas quando os primeiros sintetizadores surgiram, os circuitos eletrônicos empregados eram todos de tecnologia “analógica” (transistores, capacitores, resistores, etc) e possuíam limitações e características bastante peculiares” (RATTON, 2005)

Na década de 70, com o desenvolvimento de chips microprocessadores, foi possível desenvolver sintetizadores híbridos. Estes instrumentos apresentam controle digital e síntese analógica, facilitando o controle de diversos osciladores ao mesmo tempo e permitindo o uso de polifonias de forma mais fácil. Cada empresa criou seus próprios padrões de comunicação para controle entre dispositivos, o que causou algumas dificuldades para músicos que possuíam instrumentos de empresas distintas e gostariam de conectá-los.

Figura 3 – Sintetizador Modular



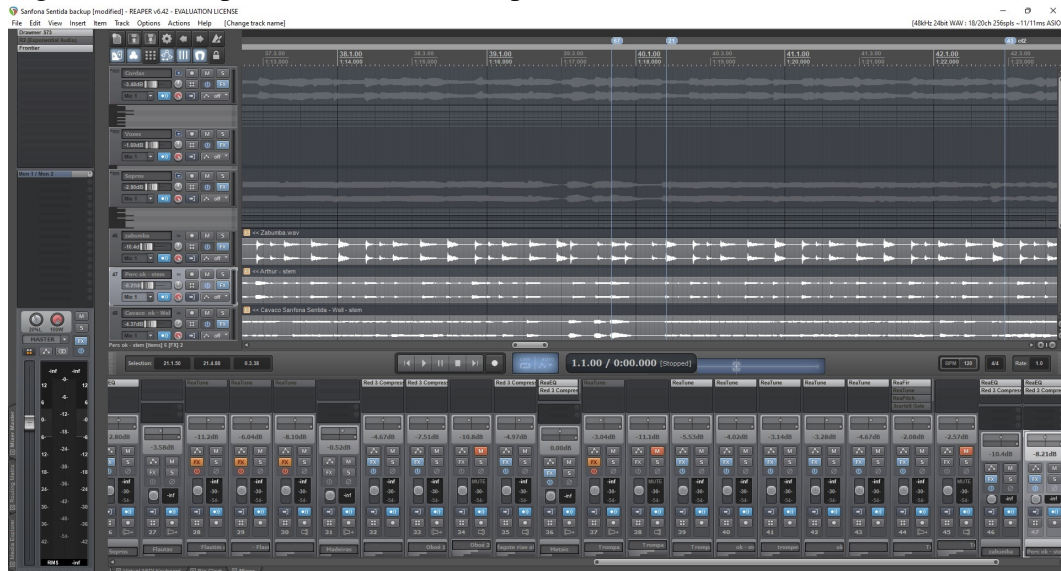
Fonte – (REDBULL, 2019)

<sup>4</sup> Polifonia se trata da possibilidade de se executar várias notas ao mesmo tempo

Para facilitar o controle dos sintetizadores e criar um padrão universal de comunicação, várias empresas do ramo de instrumentos eletrônicos se uniram para desenvolver um sistema de comunicação de áudio universal. Assim, surgiu em 1983 o protocolo MIDI (*Musical Instrument Digital Interface*). Com este protocolo os sintetizadores que antes possuíam sistemas extremamente particulares puderam ter seus controles compartilhados. Assim, com apenas um teclado é possível controlar inúmeros dispositivos simultaneamente, facilitando o uso de sintetizadores modulares, ilustrado na Figura 3.

A partir da década de 90 a distribuição de mídias analógicas começou a entrar em desuso com o surgimento dos CD's. Além disso, a forma de se produzir música também sofreu grandes mudanças com o desenvolvimento dos computadores pessoais e com o surgimentos de DAW's (*Digital Audio Workstation*). DAW's são programas que funcionam como um estúdio de produção musical virtual. Isso tornou a produção musical mais acessível, permitindo que uma pessoa com apenas um computador pudesse produzir e gravar suas músicas sem a necessidade de sair de casa. Na Figura 4 podemos verificar um exemplo da interface de uma DAW, no caso o programa Reaper.

Figura 4 – Exemplo de uma DAW (Reaper)



Fonte – Gerado pelo autor

A tecnologia possibilitou que músicos e produtores musicais pudessem utilizar instrumentos virtuais conhecidos como VSTi's (*Virtual Studio Technology instruments*). Estes instrumentos nasceram com o intuito de simular instrumentos reais, sejam esses eletrônicos ou acústicos. Atualmente a fidelidade sonora dos instrumentos virtuais dificulta o reconhecer se um instrumento é tocado por uma pessoa ou um computador. Além disso, somado com o protocolo

MIDI é possível utilizá-los em tempo real para performances musicais.

Diversos trabalhos envolvendo formas de captação e de conversão de sinais já foram desenvolvidos ao longo dos anos, porém no meio acadêmico, a maioria dos algoritmos são aplicados para outras tarefas, tais como detecção de frequência de ruídos ou em aplicações médicas. No mercado, temos alguns equipamentos que, por exemplo, convertem sinais de guitarras para o protocolo MIDI. Nestes sistemas é possível avaliar algumas limitações quanto à aplicações em variedades distintas de equipamentos. O presente trabalho oferece um método generalista para conversão independente do instrumento, necessitando apenas o conhecimento do sinal de entrada.

A partir do estudo feito por Ratton (2005) relacionado ao controle de dispositivos musicais aplicando o protocolo MIDI diversos obstáculos surgem. Sinais com harmônicos com amplitude maior do que a frequência fundamental podem causar erros no reconhecimento da frequência fundamental. Caso o algoritmo analise o gráfico no domínio da frequência e verifique o sinal de maior amplitude teremos um erro. Um erro também é possível nos casos em que um algoritmo de *zero crossing* é aplicado para o reconhecimento do sinal, pois podem haver vários cruzamentos dentro de um período, como relatado por Ratton (2005):

“(...) existe também a possibilidade de um ou mais harmônicos eventualmente se sobressaírem no som, o que pode “enganar” o conversor, que assumirá uma frequência que é o dobro ou o quádruplo do real, gerando uma nota MIDI completamente errada. Isso pode ocorrer se o guitarrista não tocar a corda de uma forma “limpa” e “regular”.” (RATTON, 2005)

Outro problema comum em sistemas de detecção de frequência para áudio é o tempo de resposta destas frequências. Uma vez que quanto menor a frequência maior será o período, maior será o tempo de resposta para detectar a frequência. Ratton (2005) relata que as notas graves possuem frequências mais baixas, sendo que o conversor MIDI precisa de mais tempo para detectar as notas graves do que as agudas”. Com isso é importante aplicar sistemas de detecção de frequência com rápida resposta.

A partir das informações supracitadas, o projeto de um sistema de conversão de características de áudio para o protocolo MIDI se faz bastante necessário. Sendo assim, alguns questionamentos surgem: como desenvolver um sistema eficiente para conversão de características de áudio para protocolo MIDI? Como tornar o sistema responsivo para diversos instrumentos? Quais as melhores técnicas para se aplicar no projeto?

Este trabalho está estruturado em capítulos e seções. O capítulo 2 está elaborado para facilitar o entendimento teórico englobado ao projeto. Nele são abordados temas como teoria musical, psicoacústica, o protocolo MIDI e por fim processamento de sinais. O capítulo 3 trata das técnicas aplicadas no presente trabalho. O capítulo 4 aborda os resultados obtidos durante a pesquisa. Por fim o capítulo 5 trará as conclusões e apresentará possíveis trabalhos futuros. Todos os algoritmos relacionados ao trabalho foram depositados em <[https://github.com/LPValente/PIC18F47K40\\_Audio\\_Signal\\_to\\_MIDI](https://github.com/LPValente/PIC18F47K40_Audio_Signal_to_MIDI)>.

## **1.1 Objetivos**

### ***1.1.1 Objetivo Geral***

Desenvolver um sistema embarcado de análise de características fundamentais de áudio, como amplitude e frequência fundamental, que converta estas características para o protocolo MIDI, atendendo as especificações do protocolo.

### ***1.1.2 Objetivos Específicos***

1. estudar as técnicas para identificação da frequência fundamental e amplitude de um sinal sonoro;
2. avaliar a necessidade da aplicação de filtros de frequências no sistema;
3. desenvolver algoritmo de conversão da frequência fundamental e amplitude de um sinal para protocolo MIDI;
4. avaliar o desempenho do sistema em diversos sinais;
5. aplicar o sistema para controle de instrumentos virtuais e sintetizadores.

## 2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo aborda os conhecimentos específicos necessários para o desenvolvimento e entendimento do trabalho. Serão abordados assuntos relativos à teoria musical, psicoacústica, ao protocolo MIDI e processamento de sinais seguindo a seguinte organização:

- Seção 2.1: O som, as notas, e tipos de temperamento serão abordados durante esta seção.
- Seção 2.2: Noções de psicoacústica como a percepção de atraso e a percepção de amplitude pelo cérebro serão discutidas durante esta seção.
- Seção 2.3: O protocolo MIDI, seus requisitos básicos, assim como seu funcionamento e exemplos de aplicações serão apresentadas.
- Seção 2.4: Esta seção abordará temas relacionados à processamento analógico e digital de sinais.

### 2.1 Teoria Musical

Abordar conceitos sobre teoria musical permite um melhor entendimento das características dos sinais sonoros a serem consideradas nas tarefas de processamento a serem realizadas.

A música pode ser compreendida como a organização dos sons dentro de um espaço de tempo. O som é caracterizado como um dos elementos mais importantes da música. Uma das características do som é o tipo de vibração. Vibrações regulares são caracterizadas por ter frequência definida, enquanto as vibrações irregulares frequência indefinida. No livro “Teoria da Música” (MED, 2017) o autor diz:

“A matéria prima da música é o som, a sensação produzida no ouvido pelas vibrações de corpos elásticos. Uma vibração põe em movimento o ar na forma de ondas sonoras que se propagam em todas as direções simultaneamente. Estas atingem a membrana do tímpano fazendo-a vibrar. Transformadas em impulsos nervosos, as vibrações são transmitidas ao cérebro que as identifica como tipos diferentes de sons. Consequentemente, o som só é decodificado pelo cérebro” (MED, 2017, p. 11)

Além da característica supracitada, podemos considerar como características principais do som as seguintes:

- Altura: frequência emitida pela fonte sonora em hertz (Hz).
- Duração: tempo em que o som se mantém.
- Intensidade: volume sonoro, podendo ser medido como potência em decibel(dB).

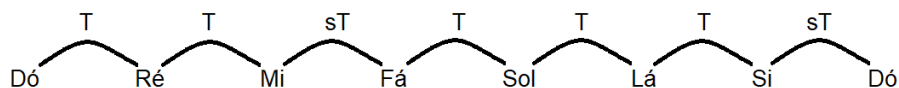
- **Timbre:** característica específica de uma fonte sonora que possibilita reconhecer qual instrumento está executando ou reconhecer quem está a falar.

Os sons são organizados em notas musicais de acordo com a frequência, sendo esses divididos em sete notas. Essas notas possuem duas principais notações, uma utilizada pelo Brasil e outros países e outra utilizada pelos Estados Unidos. A notação abordada no Brasil aplica as sílabas: dó, ré, mi, fá, sol, lá e si. Já a notação americana utiliza as letras: A, B, C, D, E, F, G. No Brasil o uso de letras está mais associado a notação de acordes.

O sistema de notas musicais mais aplicado no ocidente assume 12 sons específicos. Cada som se distancia um do outro por uma distância de um semitom. Para utilizarmos todos estes sons com as sete notas supracitadas utilizamos acidentes musicais. O acidentes utilizados são sustenidos ( $\sharp$ ), bemóis ( $\flat$ ) e bequados ( $\natural$ ). Sustenidos aumentam a altura da nota em um semitom, bemol diminui a altura em um semitom e bequados voltam uma nota com acidente para sua altura natural.

Uma escala que utiliza apenas notas naturais (sem acidentes) começando da nota dó possui uma estrutura conhecida como escala maior de dó. A distância considerada entre as notas naturais está representada na Figura 5. Como podemos verificar a distância entre a nota mi e a fá já possui um semitom, assim se utilizarmos o acidente sustenido na nota mi, tornando-a mi sustenido, temos uma nota com mesmo som de fá. O mesmo acontece entre a nota si e a dó. Estas notas são chamadas de notas enarmônicas<sup>1</sup>.

Figura 5 – Distância entre as notas naturais



Fonte – Gerado pelo autor

Além das notas, o conceito de oitava se faz importante, pois as notas são consideradas de modo cíclico. Sempre que atingimos notas mais agudas podemos reiniciar a sequência de notas. Um exemplo prático: se estivermos executando a nota si e quisermos tocar uma nota meio tom acima iremos voltar para a nota dó, reiniciando o ciclo.

<sup>1</sup> Notas enarmônicas se tratam de notas com mesmo som ou mesma frequência



## 2.2 Psicoacústica

Psicoacústica é o estudo da relação entre as ondas sonoras e sua percepção humana. Entre esses conceitos temos o atraso entre execução e escuta, e a percepção de amplitude de um sinal de áudio pelo ouvido humano.

Quando produzimos algum som, o tempo entre a produção e a escuta possui um atraso independente de haver algum processamento do sinal, seja digital ou analógico. Caso esse atraso seja perceptível na execução de uma peça musical ocorre distorção do sincronismo entre os instrumentos, comprometendo a percepção adequada dos mesmos.

O tempo no qual o atraso se torna perceptível pode variar de pessoa pra pessoa, porém por segurança podemos considerar um valor próximo a 10 ms. Não há como remover totalmente a latência em dispositivos de processamento digital, mas é possível reduzir ao máximo este atraso de maneira que ele se torne imperceptível (PRACTICAL MUSICAL PRODUCTION, 2021).

Em relação à amplitude, o ouvido humano interpreta os sinais de áudio de forma logarítmica, sendo necessária a aplicação de uma escala logarítmica, como a escala decibel, para linearizar a amplitude de um sinal. O autor Boulanger, no livro “*The Audio Programming Book*”, aborda a questão da aplicação da escala decibel:

“Há uma razão simples do porquê o decibel ser tão importante para engenheiros de áudio (e para todos que trabalham com áudio): a escala decibel [...] reflete a maneira como o ouvido humano percebe a mudança no volume sonoro[...].” (BOULANGER RICHARD; LAZZARINI, 2011, p. 210-211, traduzido pelo autor)

Na Tabela 1 temos alguns valores PCM (*Pulse Code Modulation*<sup>2</sup>). Sinais digitais possuem uma relação sinal/ruído, conhecida como SNR (*Signal-To-Noise Ratio*) relacionada à quantidade de bits do PCM. Quanto maior a quantidade de bits maior a precisão do áudio e, por consequência, maior será o SNR.

O erro de quantização é utilizado como base de cálculo para o SNR. Para o cálculo consideramos o valor máximo absoluto do sinal,  $2^{n-1}$ , por conta do sinal ser bipolar. Como a amplitude do sinal é definida primeiramente em valores inteiros, temos que a distância mínima do sinal é o valor 1 em amplitude PCM de valores inteiros, assim, definimos que o erro máximo de quantização é de 0,5. Como visto no livro “*The Audio Programming Book*” (BOULANGER

<sup>2</sup> *Pulse Code Modulation*, ou modulação por código de pulsos é um código para conversão linear de amplitude de um sinal para binário

Tabela 1 – Amplitude e escalas dB

Floating-point PCM amplitude	16-bit PCM amplitude	10-bit PCM amplitude (aprox)	db
1.0	32768	512	0
0.75	24576	384	-2.4987
0.5	16384	256	-6.0206
0.25	8192	128	-12.0412
0.1	3277	51	-20.0
0.05	1638	26	-26.0206
0.01	328	5	-40.0
0.001	33	NA	-60.0

Fonte – produzido pelo autor, baseado em (BOULANGER RICHARD; LAZZARINI, 2011, p. 211)

RICHARD; LAZZARINI, 2011, p. 436, traduzido pelo autor): “A informação perdida conhecida como erro de quantização é a diferença entre o valor de saída após a quantização e a amplitude contínua de entrada”. Define-se a SNR como:

$$SNR = \frac{2^{n-1}}{0,5} = 2^n. \quad (2.1)$$

A Eq. 2.1 costuma não ser muito utilizada, sendo mais aceita a sua notação em decibéis. Para aplicarmos o valor de SNR em decibéis aplicamos a equação que se segue:

$$SNR_{dB} = 20 \times \log_{10} 2^n = 6,02n. \quad (2.2)$$

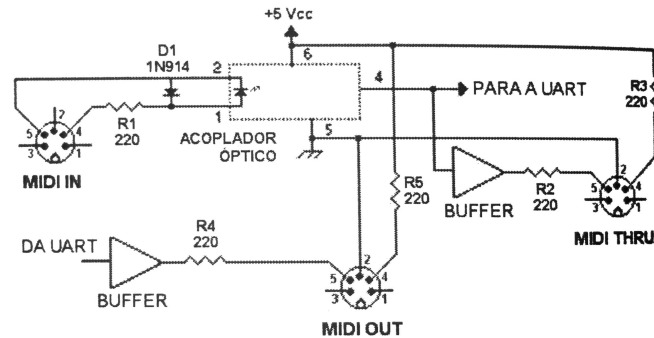
### 2.3 Protocolo MIDI

O protocolo MIDI surgiu na primeira metade dos anos 80 para padronização da comunicação entre equipamentos musicais eletrônicos. Sua fácil implementação e eficiência possibilitou sua consolidação até os dias atuais e com poucas mudanças (RATTON, 2005). No ano de 2020, a versão 2.0 do protocolo MIDI foi criada. No entanto, tal versão não é considerada nesse trabalho por ser muito incipiente.

Os requisitos de hardware necessários da interface do protocolo MIDI, tanto para recepção quanto para a emissão dos dados, são bem simples, sendo necessários poucos compo-

mentos para seu pleno funcionamento. É possível observar o circuito básico de interfaceamento MIDI na Figura 6. O acoplador óptico provê isolamento elétrico entre os dispositivos para redução dos ruídos. As entradas e saídas MIDI IN, MIDI OUT e MIDI THRU utilizam conectores do tipo DIN5<sup>3</sup> como padrão, porém existem dispositivos com outras entradas, como conectores do tipo TRS<sup>4</sup>.

Figura 6 – Interfaceamento MIDI



Fonte – (RATTON, 2005)

O sistema possui comunicação de dados *full-duplex*<sup>5</sup>, sendo sua transmissão e recepção de dados aplicadas em pinos distintos de um microcontrolador. Há também duas portas conectadas a UART (*Universal Asynchronous Receiver/Transmitter*): um vindo da porta MIDI IN indo para a UART, e outro saindo da UART e indo para a porta MIDI OUT. A porta MIDI THRU é utilizada nos casos em que há a necessidade de se enviar o mesmo sinal da entrada MIDI IN diretamente para outro dispositivo.

O sistema de transmissão e recepção do MIDI é assíncrono. Assim, não existe sinal de *clock* compartilhado entre os dispositivos, sendo necessária a geração de um *clock* definido pelo protocolo dentro de cada dispositivo. No caso, o protocolo MIDI determina um *clock* com uma taxa de transmissão de 31250 bps (*bit per second*).

Para que o início e o fim de uma transmissão assíncrona seja reconhecido é necessário enviar bits que indiquem o início e término de uma comunicação. No estado em que o sistema aguarda a recepção de novas informações, a entrada é mantida em nível lógico alto. Para que a transmissão ocorra, é enviado um bit de início, deixando o nível lógico do sinal em baixo, um byte de informação é enviado, e um bit de término de nível lógico alto é enviado. Todas as mensagens aplicadas no protocolo possuem tamanho de um byte. É possível observar o formato

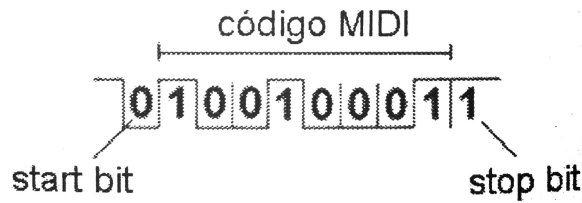
<sup>3</sup> Seu nome vem da organização alemã de padronização *Deutsches Institut für Normung*

<sup>4</sup> Seu nome se deve a sua estrutura possuindo as partes ponta (*Tip*), anel (*Ring*) e manga (*Sleeve*)

<sup>5</sup> *full-duplex* consiste em sistemas que são capazes de transmitir e receber sinais simultaneamente

de onda de uma mensagem MIDI na Figura 7.

Figura 7 – Forma de onda de uma mensagem MIDI



Fonte – (RATTON, 2005)

Para cada mensagem, considerando os bits de início e fim, são transmitidos dez bits. Sendo assim, o tempo para transmissão de cada mensagem MIDI é de 300 microssegundos. Doravante veremos a estrutura das mensagens e como são interpretados os dados de mensagens de canal e mensagens de sistema.

### 2.3.1 Mensagens MIDI

As mensagens MIDI correspondem as informações enviadas entre dispositivos pelo protocolo MIDI. Elas podem ser categorizadas em dois tipos, sendo estes: mensagens de canal e mensagens de sistema.

As mensagens de canal são as principais mensagens enviadas por um dispositivo MIDI. através mensagens são escolhidos os canais e enviadas as notas, dinâmica e controles. Todas as mensagens de canal possuem o primeiro bit em estado lógico baixo enquanto mensagens de sistema possuem o primeiro bit em estado lógico alto. Assim, as informações contidas nas mensagens são localizadas nos sete bits menos significativos.

Para o envio de notas são necessárias três mensagens. A primeira mensagem é chamada de *note on*, que informa a ativação de uma nota e o canal. A primeira metade do byte de mensagem informa que uma nota será ativada e a outra metade define em qual canal será aplicado este comando. Como temos 16 canais precisamos reservar quatro bits para estes canais. A segunda mensagem é chamada de *note*. Esta informa qual nota será executada. Como temos sete bits para informar a nota, temos 128 possibilidades. Na Tabela 2 podemos ver a relação entre frequências, notas e sua correspondente no protocolo. A terceira mensagem é chamada de *velocity*. Esta envia a dinâmica no qual o som será reproduzido. O nome *velocity* está relacionado à velocidade no qual uma tecla é pressionada. Assim como as notas, temos 128 valores possíveis

de dinâmica.

Para que uma nota deixe de ser executada podemos efetuar duas operações: envio de mensagem para desligar a nota (também chamado de *note off*), ou então mandamos a mesma mensagem de ativação, porém com o valor zero de *velocity*. A mensagem de *note off* funciona similar a mensagem *note on*, apenas a primeira metade da primeira mensagem é diferente.

Uma técnica aceita pelo protocolo para economia de mensagens, e consequentemente de tempo, é que no caso de várias mensagens de ativação precisem ser enviadas há a possibilidade de mandar a primeira mensagem apenas uma vez. Com uma economia de 0,3 ms para cada mensagem de ativação consecutiva. Esta técnica é aplicada na reprodução de notas simultâneas, e funciona tanto para mensagens *note on* quanto para mensagens *note off*.

Tabela 2 – Relação entre notas musicais, notas midi e frequências

Nota	Oitavas										
	-1	0	1	2	3	4	5	6	7	8	9
C	8.18 (0)	16.35 (12)	32.70 (24)	65.41 (36)	130.81 (48)	261.62 (60)	523.25 (72)	1046.50 (84)	2093.00 (96)	4186.00 (108)	8372.02 (120)
C#/Db	8.66 (1)	17.32 (13)	34.65 (25)	69.30 (37)	138.59 (49)	277.18 (61)	554.36 (73)	1108.73 (85)	2217.46 (97)	4434.92 (109)	8868.84 (121)
D	9.18 (2)	18.35 (14)	36.71 (26)	73.42 (38)	146.83 (50)	293.66 (62)	587.33 (74)	1174.66 (86)	2349.32 (98)	4698.64 (110)	9397.27 (122)
D#/Eb	9.72 (3)	19.44 (15)	38.89 (27)	77.78 (39)	155.56 (51)	311.12 (63)	622.25 (75)	1244.51 (87)	2489.02 (99)	4978.03 (111)	9956.06 (123)
E	10.30 (4)	20.60 (16)	41.20 (28)	82.41 (40)	164.81 (52)	329.63 (64)	659.25 (76)	1318.51 (88)	2637.02 (100)	5274.04 (112)	10548.08 (124)
F	10.91 (5)	21.83 (17)	43.65 (29)	87.30 (41)	174.61 (53)	349.23 (65)	698.46 (77)	1396.91 (89)	2793.83 (101)	5587.65 (113)	11175.30 (125)
F#/Gb	11.56 (6)	23.12 (18)	46.25 (30)	92.50 (42)	185.00 (54)	369.99 (66)	739.99 (78)	1479.98 (90)	2959.95 (102)	5919.91 (114)	11839.82 (126)
G	12.25 (7)	24.50 (19)	49.00 (31)	98.00 (43)	196.00 (55)	391.99 (67)	783.99 (79)	1567.98 (91)	3135.96 (103)	6271.93 (115)	12543.85 (127)
G#/Ab	12.98 (8)	25.96 (20)	51.91 (32)	103.83 (44)	207.65 (56)	415.30 (68)	830.61 (80)	1661.22 (92)	3322.44 (104)	6644.87 (116)	13289.75 (128)
A	13.75 (9)	27.50 (21)	55.00 (33)	110.00 (45)	220.00 (57)	440.00 (69)	880.00 (81)	1760.00 (93)	3520.00 (105)	7040.00 (117)	14080.00 (129)
A#/Bb	14.57 (10)	29.13 (22)	58.27 (34)	116.54 (46)	233.08 (58)	466.16 (70)	932.33 (82)	1864.65 (94)	3729.31 (106)	7458.62 (118)	14917.24 (130)
B	15.43 (11)	30.87 (23)	61.73 (35)	123.47 (47)	246.94 (59)	493.88 (71)	987.77 (83)	1975.53 (95)	3951.07 (107)	7902.13 (119)	15804.27 (131)

Fonte – elaborado pelo autor, baseado em tabela encontrada na Wikipedia2021b

As mensagens de sistema estão associadas à controles gerais do funcionamento do dispositivo. Suas mensagens começam com o primeiro bit em estado lógico alto. Alguns dos controles possíveis são de controles gerais de dispositivos de reprodução de música como iniciar, pausar ou pular uma música. É possível utilizar as mensagens de sistema para sincronizar dispositivos distintos. Um exemplo é se dois dispositivos estiverem reproduzindo instrumentos

distintos em uma mesma música.

Dentro das mensagens de sistema também são abordadas as mensagens exclusivas de sistema (SysEx). Estas mensagens foram amplamente utilizadas para expansões no protocolo MIDI como podemos verificar no livro “MIDI Total”(RATTON, 2005): “Por serem mensagens com grande capacidade de dados, é nesta categoria têm sido incluídas as novas aplicações”.

### 2.3.2 Controladores MIDI

Existem diversos modelos de controladores MIDI para variados usos, dentre os quais podemos citar: teclados controladores, baterias eletrônicas, guitarras controladoras, mesas digitais, entre outros dispositivos. Estes controladores são amplamente utilizados tanto para controle de sons em dispositivos de síntese sonora, como em dispositivos de estúdio, como mesas de som e *racks* de efeito sonoro.

Os teclados controladores foram os primeiros dispositivos MIDI desenvolvidos devido à sua relação mais direta com os sintetizadores (RATTON, 2005). Sua forma de execução é similar a de um piano, possuindo além das teclas uma seção de controle para envio de mensagens para outros dispositivos. Usualmente, cada tecla possui dois botões, sendo a diferença de tempo em que eles são pressionados utilizada para identificar a velocidade que a tecla foi pressionada e ser aplicada na dinâmica da execução. Na Figura 8 podemos verificar um teclado controlador da marca Arturia.

Figura 8 – Teclado Controlador MIDI

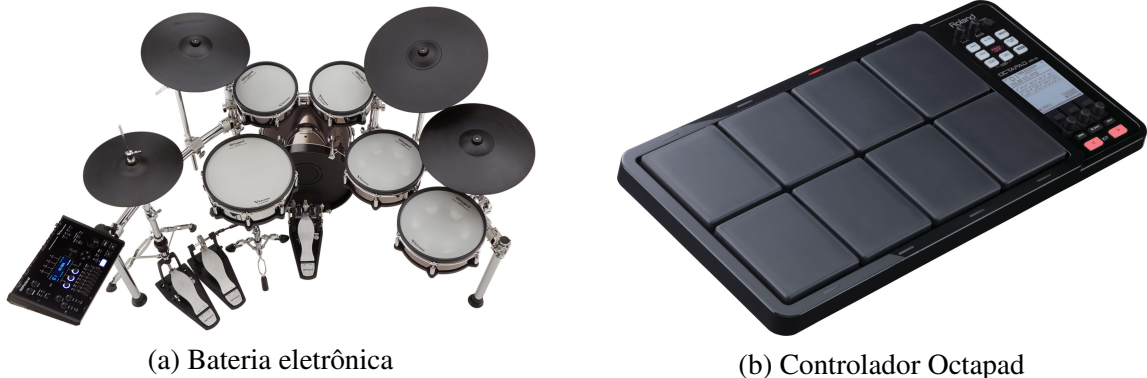


Fonte – (ARTURIA, 2021)

Outros controladores bastante difundidos são os percussivos. Alguns destes assemelham-se com instrumentos acústicos, enquanto outros possuem uma aparência mais eletrônica. Apesar das aparências diversas, estes controladores possuem o mesmo princípio de funcionamento. A

detecção de impacto é realizada através de sensores de vibração (como cristais piezoelétricos) ou de pressão (baseado em arranjos de resistores). Na Figura 9 observam-se dois tipos diferentes de controladores percussivos, sendo na esquerda uma bateria eletrônica, com aparência bastante similar com uma acústica, e, na direita um controlador baseado em *pads*.

Figura 9 – Tipos mais comuns de controladores percussivos. Na esquerda uma bateria eletrônica e a direita um controlador por *pads*, ambos da marca Roland



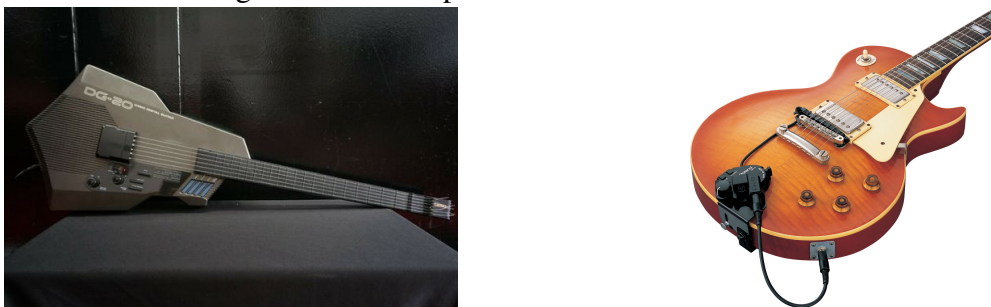
(a) Bateria eletrônica

(b) Controlador Octapad

Fonte – (ROLAND, 2021)

Um dos controladores mais populares atualmente são as guitarras MIDI<sup>6</sup>. Existem dois tipos de guitarra MIDI. Uma delas é uma controladora que usa cordas metálicas, porém sem afinação. O sistema detecta a posição do dedo no braço do instrumento e se a corda está vibrando. O outro tipo de guitarra MIDI possui um captador hexafônico (capta a vibração individual de cada corda de uma guitarra de seis cordas) com um sistema de detecção de frequência individual por corda, que converte cada frequência para midi. Na Figura 10 podemos verificar os dois modelos de guitarra MIDI citados.

Figura 10 – Guitarras MIDI. Na esquerda uma guitarra MIDI controladora da marca Casio e a direita uma guitarra com dispositivo hexafônico da marca Roland



(a) Guitarra MIDI Casio

(b) Captador hexafônico Roland

Fonte – (a) (PINTEREST, 2019) (b) (ROLAND, 2021)

<sup>6</sup> O presente trabalho possui similaridades com este projeto, porém não se limita a instrumentos de corda

Dentre os dispositivos controladores MIDI temos ainda os controladores de sopro. Estes controladores utilizam o sopro do usuário e botões para controle de notas. Estes instrumentos estão mais voltados para um público que toca instrumentos como saxofone ou similares. Na Figura 11 podemos verificar um controlador MIDI de sopro da marca Yamaha.

Figura 11 – Controlador MIDI de Sopro



Fonte – (YAMAHA, 2019)

## 2.4 Processamento de Sinais

Processamento de sinais constitui um conjunto de técnicas de análise e manipulação de sinais, seja no domínio analógico ou digital. Para o processamento analógico são aplicados componentes elétricos (diodos, capacitores, indutores, transistores, etc). Já para o processamento digital é possível efetuá-lo com dispositivos microprocessadores, sendo mais usualmente aplicados os DSP's (*Digital Signal Processor*).

Ambas formas de processamento possuem suas vantagens e desvantagens. O processamento digital costuma ter sistemas que ocupam um menor espaço físico, principalmente no caso de aplicações em filtros de ordens muito altas. Estes sistemas também são menos suscetíveis a ruídos. Além disso a atualização dos seus filtros é facilitadas, muitas vezes necessitando trocar um chip ou até mesmo apenas atualizá-lo. Sistemas digitais necessitam de passos de conversão do sinal que produzem latência entre a entrada e a saída.

Para sistemas que utilizam saturação do sinal é comum a preferência do uso de processamento analógico. Sistemas de processamento analógico possuem uma facilidade maior de implementação quanto a necessidade de equipamentos, pois não há a necessidade de um computador ou programas para que sua implementação seja feita.



### 2.4.1 Captação de sinais

A primeira etapa de qualquer sistema de processamento é a captação de sinais. As técnicas empregadas nesse processo envolve o uso de elementos transdutores, que transformam vibrações mecânicas em sinais elétricos. Entre os transdutores destacam-se os microfones; captadores magnéticos; captadores piezoelétricos; e captadores ópticos.

Os microfones são os transdutores mais comuns, sendo aplicados em diversos instrumentos acústicos e em amplificadores de instrumentos elétricos. Existem diversos tipos de microfones, sendo os mais comuns os microfones dinâmicos e condensadores. Cada microfone possui características próprias de resposta em frequência e de zona de captação, sendo mais comum os cardióides<sup>7</sup>. Na Figura 12 podemos verificar diversos tipos de microfones produzidos pela marca de equipamentos musicas Shure.

Figura 12 – Microfones de diversos tipos

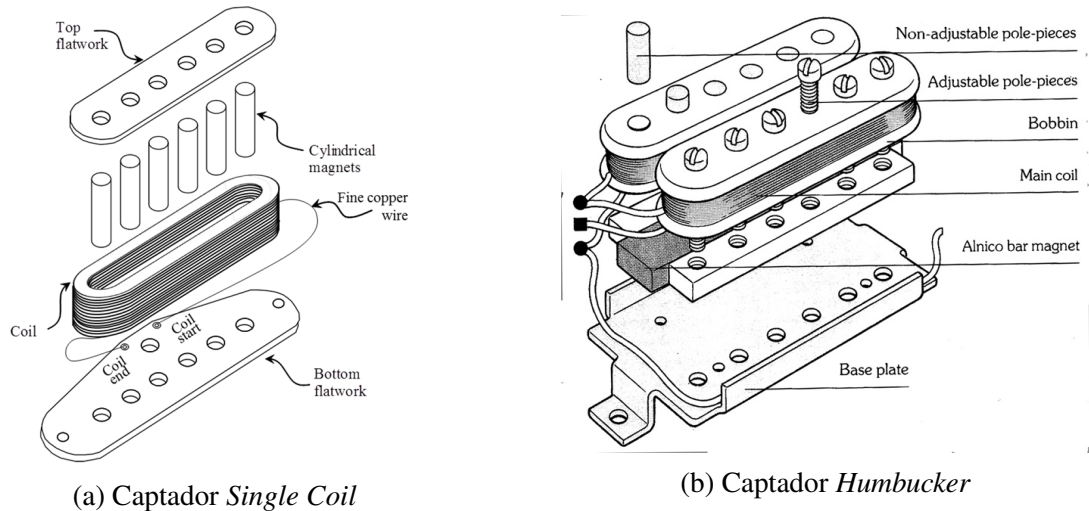


Fonte – (SHURE, 2012)

Os captadores magnéticos são os tipos de captadores mais comuns para instrumentos eletro-acústicos sendo aplicados amplamente em guitarras e baixos elétricos. Seu sistema consiste na aplicação do efeito indutivo de um fio de cobre enrolado em um material magnético. Geralmente as cordas de instrumentos de corda são construídas de elementos com características magnéticas. Quando as cordas vibram elas causam um distúrbio no campo magnético, variando proporcionalmente a corrente no fio que está enrolado. Na Figura 13 podemos verificar dois tipos comuns de captadores magnéticos.

<sup>7</sup> Microfones cardióides são chamados assim pois sua zona de captação possui um gráfico com um formato que se assemelha ao de um coração

Figura 13 – Captadores magnéticos mais comuns. Na esquerda um captador *single coil* (bobina simples) e a direita um captador *humbucker* (bobina dupla)



Fonte – (SAM ASH, 2019)

Os captadores piezoelétricos são utilizados em instrumentos acústicos como violões e violinos. Estes captadores geralmente são constituídos com cristais que apresentam a propriedade piezoelétrica, ou seja, podem converter energia mecânica de vibrações em energia elétrica. Possuem um custo reduzido e uma facilidade maior de serem implementados. O sal de La Rochelle é um material piezoelétrico que pode facilmente ser produzido em casa. Apesar destas vantagens podemos destacar algumas desvantagens, como maior suscetibilidade à ruídos de impacto entre outros. Na Figura 14 podemos verificar um tipo comum de captador piezo.

Figura 14 – Captador Piezo



Fonte – (AMERICANAS, 2021)

Os captadores ópticos consistem em um emissor de luz infravermelho e um fototransistor. Para seu funcionamento temos o sinal infravermelho de alta frequência sendo emitido pelo emissor incidindo na corda e sendo refletido para o fototransistor. Quando a corda vibra temos uma variação no sinal recebido pelo fototransistor. Esse sinal, que corresponde ao sinal emitido, é filtrado para remoção de alta frequência, restando apenas o sinal gerado pelo instrumento. Embora nova, essa tecnologia vem sendo testada em instrumentos acústicos como alternativa aos captadores mais comuns por ter uma tolerância maior a ruídos e aceitar cordas de qualquer tipo de material. Na Figura 15 podemos verificar o exemplo de um captador óptico.

Figura 15 – Captador Óptico



Fonte – (Premier Guitar, 2021)

#### 2.4.2 Filtragem de sinais

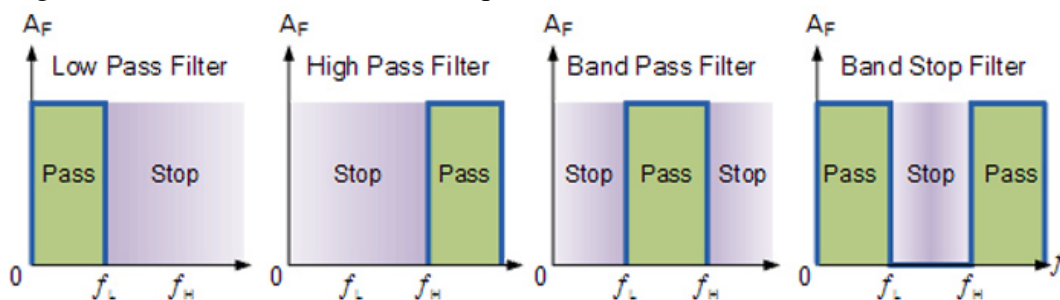
A filtragem é o processo de se remover elementos que não são desejados de um sinal, como ruídos, e/ou amplificar os elementos que se deseja destacar. Apesar de existirem filtros de amplitude e de frequências, esta seção irá detalhar especificamente dos filtros de frequência. Entre eles temos:

- HPF: O filtro HPF (*High Pass Filter*), ou filtro passa-alta, é um filtro que atenua sinais de frequência abaixo da frequência de corte e permite a passagem de sinais de frequência acima da frequência de corte.

- LPF: O filtro LPF (*Low Pass Filter*), ou filtro passa-baixa, é um filtro complementar ao HPF, atenuando sinais de frequência acima da frequência de corte e permitindo a passagem de sinais de frequência abaixo da frequência de corte.
- BPF: O filtro BPF (*Band Pass Filter*), ou filtro passa-faixa, é um filtro construído a partir da junção de um filtro HPF e um filtro LPF, sendo o filtro HPF operando com uma frequência de corte menor que a frequência de corte do filtro LPF.
- Notch: O filtro *notch* é um filtro complementar ao filtro BPF, sendo assim ele também é constituído pela junção de um filtro HPF e um filtro LPF, porém na sua construção o filtro HPF opera em uma frequência de corte maior que a frequência de corte do filtro LPF.
- APF: O filtro APF (*All Pass Filter*), ou filtro passa-tudo, é um filtro que simplesmente permite a passagem de todas as frequências de áudio. É comumente aplicado para correção de fase de sinais.

Os filtros ideais relativos aos listados acima estão representados graficamente na Figura 15. É possível verificar que o corte existe apenas a partir da frequência determinada pelo filtro. Nas regiões fora da zona de corte temos as frequências sem nenhuma modificação no seu ganho.

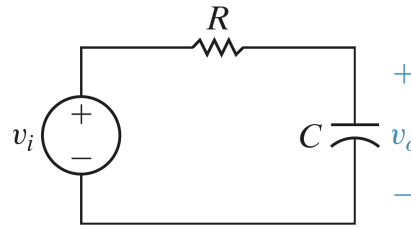
Figura 16 – Gráficos de filtros de frequência ideais



Fonte – (Sunpower UK, 2019)

Filtros de frequência ideais não existem no mundo real, assim necessitamos elaborar filtros reais. Os filtros reais podem ser construídos por meio de dispositivos analógicos ou digitais. Para a implementação de filtros analógicos são utilizados componentes eletrônicos. Estes filtros podem ser passivos ou ativos. Filtros passivos não necessitam de alimentação elétrica, enquanto filtros ativos não funcionam sem esta alimentação. O tipo mais simples de filtro é o filtro RC. Como o nome indica, este filtro é constituído apenas por elementos passivos, sendo eles um resistor e um capacitor. Este filtro pode ser analisado na Figura 17.

Figura 17 – Filtro RC passa-baixa



Fonte – (NILSSON JAMES W.; RIEDEL, 2015)

Considerando as relações de saída e entrada de um sistema é possível conceituar função de transferência. A função de transferência trata da razão, no domínio da frequência, entre a transformada de Laplace do sinal de saída e a transformada de Laplace do sinal de entrada de um sistema ((NILSSON JAMES W.; RIEDEL, 2015)). A função de transferência do circuito da Figura 17 é dada por:

$$H(s) = \frac{\frac{1}{RC}}{s + \frac{1}{RC}} \quad (2.3)$$

Fazendo  $s = j\omega$  podemos definir o módulo da expressão resultante como:

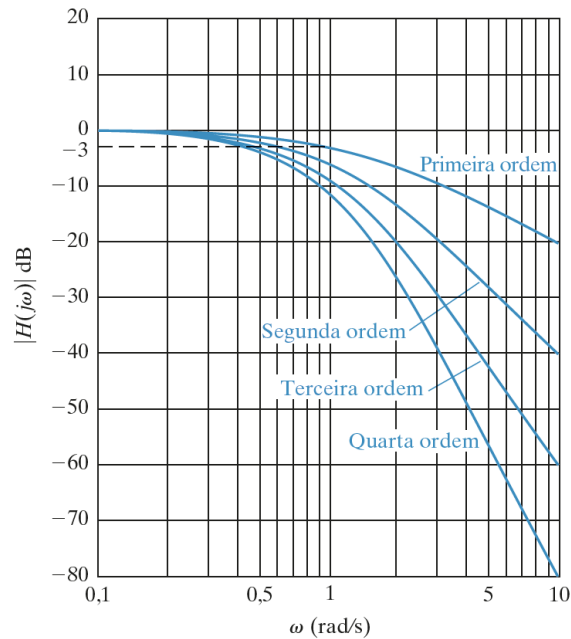
$$|H(j\omega)| = \frac{\frac{1}{RC}}{\sqrt{\omega^2 + \frac{1}{RC}}} \quad (2.4)$$

Para um filtro passa-baixa temos que, na frequência de corte  $\omega_c$ ,  $|H(j\omega)|$  é igual a  $(1/\sqrt{2})H_{max}$ , sendo  $H_{max} = 1$ . Então, podemos definir a frequência de corte  $\omega_c$  como:

$$\omega_c = \frac{1}{RC} \quad (2.5)$$

Uma técnica para melhor aproximar os filtros reais aos ideais é usar filtros de ordens superiores. Uma das formas de conseguir filtros de ordens mais altas é cascadeando filtros. A Figura 18 apresenta um diagrama com a curva de corte de um filtro passa-baixa real de várias ordens. O resultado de se usar filtros de ordens mais altas é uma aproximação do efeito de um filtro ideal, porém isto somente seria possível com um filtro de ordem infinita.

Figura 18 – Diagrama de filtro passa-baixa de ordens distintos



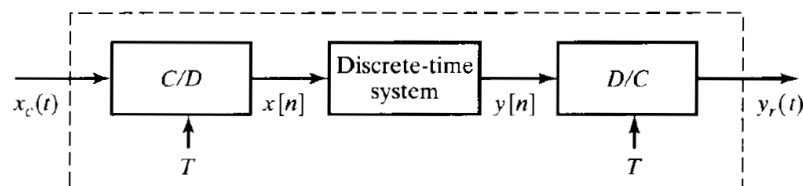
Fonte – adaptado de (NILSSON JAMES W.; RIEDEL, 2015)

### 2.4.3 Conversão AD/DA

Uma das estruturas principais para o processamento digital de sinais são conversores ADC e DAC. Estes convertem sinais analógicos para digitais e digitais para analógicos (OPPENHEIM ALAN V.; SCHAFER, 1998). Como circuitos digitais apenas processam informações binárias é necessária a conversão dos sinais analógicos para digitais. A resolução do conversor possui grande importância, sendo essencial para uma boa precisão do sinal processado.

Atualmente existem conversores especializados para diversas operações. Para áudio os conversores costumam ter 16 bits de resolução ou 65536 valores possíveis de amplitude. Além disso a taxa de amostragem costuma ser da ordem de 44 kHz. Apesar destas especificações os conversores atuais possuem resolução de 32 bits e taxa de amostragem de 192 kHz. Na Figura 19 podemos analisar o diagrama de blocos de um sistema que aplica conversores AD/DA para processamento em tempo discreto de sinais contínuos.

Figura 19 – Processamento em tempo discreto de sinais de tempo contínuo



Fonte – (OPPENHEIM ALAN V.; SCHAFER, 1998)

#### 2.4.4 Técnicas de análise de frequências

A análise de frequência possibilita a aquisição de informações em sinais. Diversas técnicas foram desenvolvidas para este fim no domínio do tempo e domínio frequência, dentre as quais:

- Transformada de Fourier
- Transformada de Laplace
- Transformada Z
- Zero Crossing

Dentre as técnicas supracitadas temos as três primeiras de análise no domínio da frequência e a última no domínio do tempo. As transformadas levam problemas que estão no domínio do tempo para o domínio da frequência para simplificar a análise dos sinais. Podemos verificar a definição de algumas transformadas nas equações Eq. 2.6, Eq. 2.7 e Eq. 2.8 para sistemas contínuos e discretos.

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (2.6)$$

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st} dt \quad (2.7)$$

$$X[z] = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \quad (2.8)$$

A detecção *Zero Crossing* é uma técnica simples de análise do sinal no domínio do tempo, o que reduz o tempo de processamento computacional. Esta técnica baseia-se na análise básica de amplitude do sinal. Caso o sinal inverta sua polaridade temos um cruzamento de zero. Considerando um sinal fundamental (ondas senoidais, triangulares, dentes-de-serra ou quadradas) temos dois cruzamentos de zero para cada período. Sabendo disso é possível, a partir do período do sinal, calcular sua frequência fundamental. Em casos de sinais mais complexos podem existir erros na detecção nos casos em que ocorram mais de dois cruzamentos por período.

### 3 METODOLOGIA

O presente capítulo descreve a metodologia de desenvolvimento do sistema proposto para análise, processamento e conversão de sinais de instrumentos musicais para o protocolo MIDI. A organização do capítulo segue a seguinte sequência:

- Seção 3.1: os requisitos do sistema serão apresentados nesta seção.
- Seção 3.2: esta seção apresentará um modelo em blocos do sistema.
- Seção 3.3: os materiais necessários para o desenvolvimento do projeto serão definidos durante esta seção.
- Seção 3.4: a estrutura do filtro *Butterworth* será apresentada ao longo desta seção.
- Seção 3.5: a definição de um sistema de detecção *zero cross* será abordada no decorrer desta seção.
- Seção 3.6: o desenvolvimento de um sistema de conversão de frequência para o protocolo MIDI será apresentado nesta seção.
- Seção 3.7: a conversão de amplitude para protocolo MIDI será desenvolvida ao longo desta seção.
- Seção 3.8: os conceitos e requisitos de comunicação serial serão abordados no decorrer desta seção.

#### 3.1 Requisitos

Os requisitos levantados para o sistema do presente trabalho foram definidos analisando os principais desafios que foram surgindo. Para uma maior acurácia do sistema de *zero cross* um sistema de filtragem de frequência é necessário. Além disso, para a análise de amplitude, é necessária a aplicação de conversor analógico para digital. O desenvolvimento do firmware necessita de um algoritmo de conversão de frequência e de amplitude para o protocolo MIDI. Para a saída do sistema há necessidade de implementação de um sistema de comunicação serial.

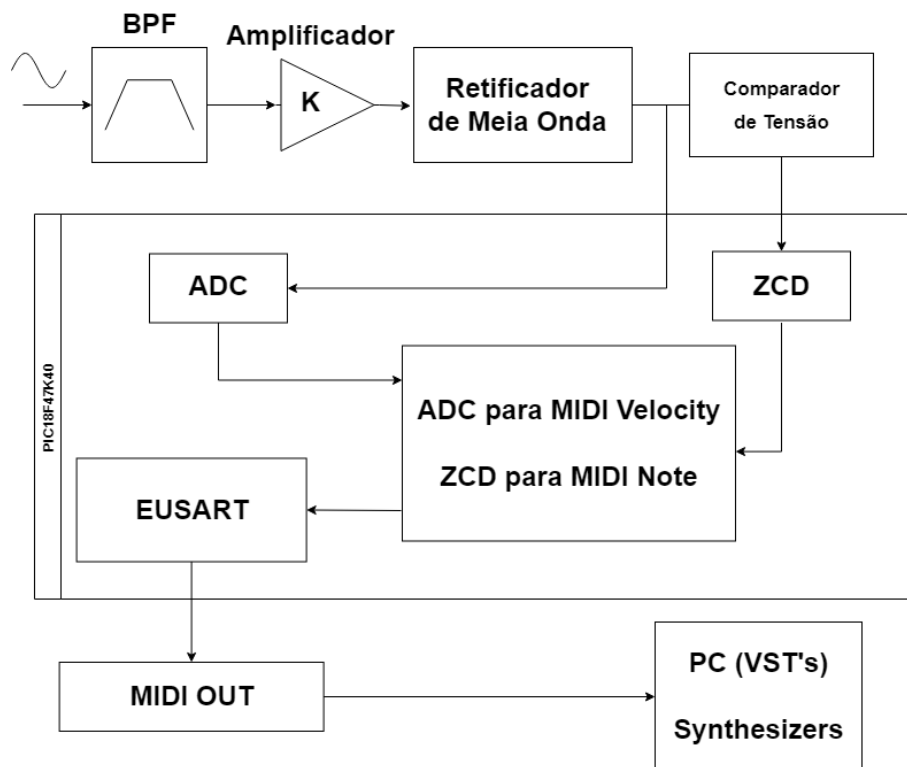
#### 3.2 Modelagem do sistema

A partir dos requisitos analisados o modelo do projeto fora elaborado. Na Figura 20 é possível verificar as partes elaboradas em blocos para o projeto. A necessidade de um comparador de tensão para o projeto se deve a adquirir uma precisão maior durante o cruzamento



do sinal a ser processado. Dentro dos requisitos do modelo de microprocessador trabalhado é possível verificar que o sistema necessita que o sinal fique acima de 0,75 V para que o cruzamento seja identificado pelo módulo ZCD. A amplificação do sinal antes do módulo ADC é aplicada para uma normalização do sinal de entrada, equiparando o valor de pico máximo com o valor máximo do módulo ADC. O filtro na entrada foi projetado para trabalhar na faixa de frequências fundamentais do sinal de entrada.

Figura 20 – Diagrama elaborado para modelagem do sistema



Fonte – Elaborado pelo autor

### 3.3 Materiais

Em se tratando de processamento digitais de sinais, há necessidade de se escolher um dispositivo processador que atenda aos requisitos funcionais da aplicação. Como utilizaremos o algoritmo de detecção *zero crossing* e uma transmissão serial para o protocolo MIDI optamos por utilizar o microcontrolador PIC18F47K40 (Microchip Technology Inc., 2018), pois este possui integrado entre os periféricos um módulo ZCD e dois módulos EUSART. Além disso o microcontrolador escolhido também possui módulos ADC que são necessários para detecção da amplitude do sinal.

Como filtro de sinal escolhemos o filtro *Butterworth* devido a sua facilidade de

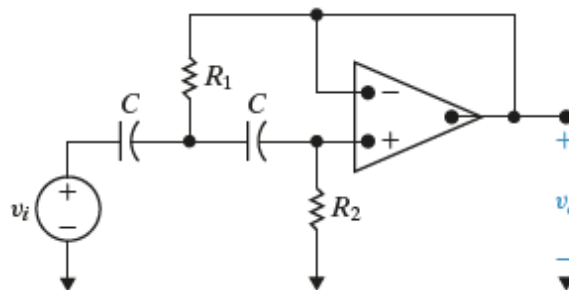
ser implementado e sua facilidade de atingir ordens maiores. O AMP-OP escolhido para a construção do filtro foi o LM324, que é um dispositivo de uso geral cujas especificações atendem as demandas do projeto. O ajuste dos valores dos resistores e capacitores do filtro foi realizado por cálculos e confirmados através de um simulador de circuitos.

Para transmissão da mensagem MIDI temos duas opções de conexões: o chip conversor de MIDI para USB CH345, e a saída padrão MIDI com o conector DIN5.

### 3.4 Filtro *Butterworth*

A escolha do filtro *Butterworth* se deve ao fato deste ser um dos filtros ativos mais aplicados em sistemas de equalização em áudio, além de ser um filtro de fácil implementação. O uso deste filtro facilita o projeto de filtros analógicos de ordens mais altas. Na Figura 21 temos o esquemático do circuito de um filtro *Butterworth* passa-alta genérico de segunda ordem. Trocando a posição dos resistores com os capacitores temos um filtro *Butterworth* passa-baixa de segunda ordem.

Figura 21 – Filtro *Butterworth* passa-alta genérico de segunda ordem



Fonte – (NILSSON JAMES W.; RIEDEL, 2015)

Sua construção é feita a partir de três componentes básicos: amplificador operacional, resistores e capacitores. Os resistores são elementos que não modificam sua impedância perante uma variação de frequência, porém os capacitores são elementos que possuem reatância capacitiva, variando o valor de sua impedância perante uma variação de frequência. O cálculo da frequência de corte do circuito se dá por:

$$f_{OH} = \frac{1}{2\pi R_1 C_1} \quad (3.1)$$

Para cada ordem do filtro o valor de atenuação da zona de corte é definida como -20 dB por década (BOYLESTAD ROBERT L.; NASHELSKY, 2013). No caso do filtro *Butterworth*

de segunda ordem temos uma atenuação de -40 dB por década. Estes valores também podem ser considerados em -6 dB por oitava para cada ordem, assim temos -12 dB por oitava na estrutura do filtro *Butterworth*.

### 3.5 *Zero Crossing*

A análise de frequência por *Zero Crossing* é uma técnica de implementação simples e com rápida resposta, porém com algumas limitações. O *Zero Crossing* permite uma resposta quanto a frequência fundamental rápida, logo após o primeiro período de um sinal. Desta forma, o tempo de atraso entre a execução e o reconhecimento da frequência fundamental é próximo ao período fundamental do sinal. Entretanto, a resposta exata depende dos harmônicos do sinal. Caso um harmônico do sinal possua uma amplitude muito significativa, este pode ocasionar vários cruzamentos de zero em um período fundamental do sinal, dificultando a resposta exata. Uma técnica que pode ser aplicada para reduzir este problema é a aplicação de filtros de sinal para as frequências principais que serão analisadas. A implementação do sistema de detecção *zero crossing* pode ser implementada via software ou via hardware.

#### 3.5.1 *Implementação via software*

A implementação via *software* necessita de um conversor analógico-digital para aquisição do formato de onda. O conversor deve realizar a aquisição do sinal tanto da parte negativa quanto positiva, sendo o número de bits dividido para a parte positiva e para a parte negativa, com valores normalizados entre -1 e 1. Desta forma um cruzamento de zero é detectado quando uma amostra anterior for negativa em relação à amostra atual e vice-versa. Durante a análise da aquisição é necessário um *clock* para a contagem do tempo entre os cruzamentos para que a frequência seja determinada. Uma vantagem do sistema implementado via *software* é a possibilidade de criar um sistema mais compacto, com menos componentes.

#### 3.5.2 *Implementação via hardware*

A implementação via *hardware* emprega um amplificador operacional no modo comparador. Neste modo temos o sinal de entrada do sistema na entrada não-inversora e o valor de tensão de referência na entrada inversora. No caso, para o cruzamento de zero a entrada inversora ficará conectado ao terra. Quando o sinal de entrada do sistema assumir um valor de

tensão superior ao valor de tensão da referência teremos na saída o valor de alimentação positiva do amp-op, já no caso do valor ser inferior teremos o valor da alimentação negativa. Assim como sua implementação via *software*, a utilização de um *clock* para contagem do tempo se faz necessária para que seja possível adquirir o valor em frequência. Uma vantagem da aplicação via hardware é a possibilidade de aplicação do sistema com interrupções de software.

### 3.6 Conversão frequência para MIDI

Para convertermos o sinal recebido para MIDI é necessário transformar a frequência fundamental do sinal em nota musical MIDI.

Tal transformação baseia-se na definição das notas musicais, aonde, a cada oitava acima dobramos a frequência e a cada oitava abaixo dividimos a frequência pela metade. Podemos tomar como exemplo a nota A4(lá na quarta oitava), que equivale a 440 Hz por padrão. Esta nota é usualmente aplicada para afinação de instrumentos. Caso queiramos verificar a frequência quando se varia a oitava podemos aplicar a Eq. 3.2:

$$f = 440 \times 2^{oct}. \quad (3.2)$$

sendo oct um número real associado à quantas oitavas desejamos subir ou descer.

Já no protocolo MIDI esse valor se relaciona linearmente com as notas musicais, aonde as notas são codificadas como inteiros entre 0-127. Para estabelecermos a relação entre esses últimos valores e as notas musicais consideramos o valor 0 em MIDI relacionado à nota C-1, seguindo a padrão de oitavas aplicado nos Estados Unidos. Para cada valor acrescentado em MIDI subimos um semitom na nota musical relativa. Logo, possuímos 12 notas possíveis dentro de uma oitava, incluindo os acidentes musicais (sustenidos e bemóis) de forma enarmônica. Se partirmos da nota A4 que possui o valor MIDI 69 podemos efetuar a seguinte equação para variar sua oitava:

$$N = 69 + 12oct. \quad (3.3)$$

sendo N um número real associado à nota MIDI resultante. Para relacionarmos a nota e a oitava podemos desenvolver a Eq. 3.3:

$$N - 69 = 12oct$$

$$oct = \frac{N - 69}{12}. \quad (3.4)$$

Geralmente se aplica como frequência de afinação dos instrumentos a nota A4 assumindo o valor de frequência de 440 Hz, porém este valor costuma variar de 432 Hz a 448 Hz. A partir da escolha do valor de frequência padrão podemos desenvolver uma equação para efetuar a conversão entre a frequência e o valor equivalente no protocolo MIDI. Partindo da Eq. 1 e substituindo o valor de n aplicando a Eq. 3.4 podemos assumir que:

$$f = 440 \times 2^{\frac{N-69}{12}}$$

$$\frac{f}{440} = 2^{\frac{N-69}{12}}. \quad (3.5)$$

A fim de retirarmos o expoente do lado direito da nossa igualdade podemos aplicar logaritmo na base 2 de ambos os lados, tendo assim:

$$\log_2\left(\frac{f}{440}\right) = \frac{N - 69}{12}$$

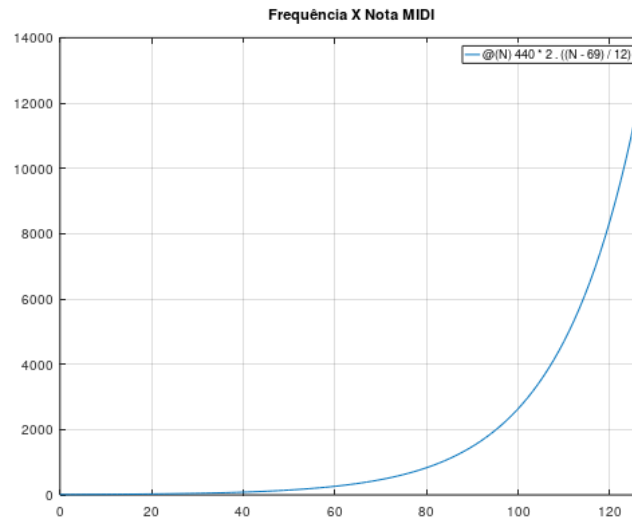
$$12 \times \log_2\left(\frac{f}{440}\right) = N - 69$$

$$N = 12 \times \log_2\left(\frac{f}{440}\right) + 69. \quad (3.6)$$

Aplicando a Eq. 3.6 temos que N se torna uma variável em função da frequência. Caso nossa frequência de entrada possua o valor 440 Hz teremos como argumento do logaritmo o valor, que resulta em zero, caso coloquemos um valor de o dobro poderemos perceber que o valor 12 será somado à 69, efetuando assim a subida no valor da oitava.

Considerando a parte decimal é possível fazer um ajuste fino na afinação, permitindo algumas técnicas ou até mesmo abrindo a possibilidade de tocar simultaneamente um instrumento virtual com o som original do instrumento, evitando batimento entre frequências próximas. Na Figura 22 podemos analisar o gráfico gerado da relação entre frequência e nota MIDI.

Figura 22 – Relação entre frequência e nota MIDI



Fonte – Gerado pelo autor

### 3.7 Conversão Amplitude para MIDI

Para aquisição do valor de amplitude do sinal o microcontrolador utilizado possui um módulo ADC de resolução de 10 bits. Assim temos valores possíveis entre 0 a 1023, que normalizados ( $V_{norm} \in [0, 1]$ ) é dado por:

$$V_{norm} = \frac{V_{in} + 1}{1024} \quad (3.7)$$

Dentro do argumento de um logaritmo não se pode usar o valor 0. Para aplicação no valor de  $V_{norm}$  no argumento do logaritmo a eq. 3.5 é possível aplicar um valor pequeno. No caso podemos assumir este valor como sendo a unidade e assumindo o valor do divisor como 1024.

O *velocity* do protocolo MIDI opera de forma linear perante a nossa audição. Nossos ouvidos interpretam a amplitude de um sinal de forma logarítmica. Para que esta conversão seja aplicada para o protocolo MIDI é necessário aplicar uma conversão para decibéis. Assim,

$$V_{dB} = 20 \times \log_{10}\left(\frac{V_{norm}}{V_0}\right). \quad (3.8)$$

Considerando um valor máximo de 12 dB e o valor de tensão máximo normalizado em 1 podemos desenvolver a Eq. 3.8, obtendo:

$$\begin{aligned}
 12 &= 20 \times \log_{10}\left(\frac{1}{V_0}\right) \\
 \frac{12}{20} &= \log_{10}\left(\frac{1}{V_0}\right)
 \end{aligned}
 \tag{3.9}$$

Extrai-se o logarítmo pela seguinte operação:

$$\begin{aligned}
 10^{\frac{12}{20}} &= \frac{1}{V_0} \\
 V_0 &= 10^{-\frac{12}{20}} \\
 V_0 &\approx 0.2512.
 \end{aligned}
 \tag{3.10}$$

Assim temos:

$$V_{dB} = 20 \times \log_{10}\left(\frac{V_{norm}}{0.2512}\right) \tag{3.11}$$

Para assumir o valor mínimo podemos considerar a Eq. 3.7 com  $V_{in}$  em zero, assim temos:

$$\begin{aligned}
 V_{norm} &= \frac{0 + 1}{1024} \\
 V_{norm} &\approx 0.001
 \end{aligned}
 \tag{3.12}$$

Considerando o valor encontrado na Eq. 3.10 e colocando na eq. 3.8 podemos encontrar o valor mínimo em dB:

$$\begin{aligned}
 V_{dB} &= 20 \times \log_{10}\left(\frac{0.001}{0.2512}\right) \\
 V_{dB} &\approx -48.
 \end{aligned}
 \tag{3.13}$$

A partir do valor encontrado na Eq. 3.13 e do valor máximo determinado em 12 dB podemos verificar uma faixa de valores de 60 dB.

O *velocity* do MIDI possui uma resolução de 7 bits, com valores de 0 a 127. Para converter a faixa de valores de dB encontrado para MIDI podemos considerar uma conversão linear aplicando:

$$Vel = V_{dB}A + B \quad (3.14)$$

$$0 = -48A + B$$

$$B = 48A$$

A partir do resultado encontrado acima e dos valores máximos encontrados para  $V_{dB}$  e para *velocity* obtemos:

$$127 = 12A + 48A$$

$$A = \frac{127}{60}$$

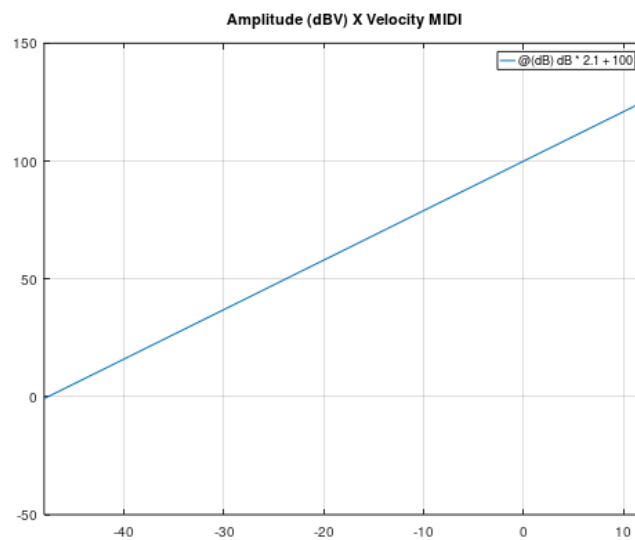
Assim podemos determinar como o valor de A como:

$$A \approx 2.1 \quad (3.15)$$

Substituindo o valor de A encontrado na Eq. 3.15 temos a equação de conversão de tensão em dB para *velocity* MIDI:

$$Vel = 2.1(V_{dB} + 48) \quad (3.16)$$

Figura 23 – Relação entre amplitude e *velocity* MIDI



Fonte – Gerado pelo autor



Como podemos verificar na Eq. 3.16 e na Figura 23 temos uma relação linear entre a amplitude e *velocity* MIDI.

### 3.8 Comunicação Serial

A necessidade da comunicação serial se faz por conta de o protocolo MIDI ser implementado utilizando módulo de comunicação serial assíncrona (ver 2.3) para comunicação entre os dispositivos.

A maioria dos microcontroladores possui um ou mais módulos USART (*Universal Synchronous Asynchronous Receiver/Transmitter*). O microcontrolador escolhido possui dois módulos EUSART (*Enhanced USART*) (Microchip Technology Inc., 2018). A preparação do dispositivo para o protocolo MIDI inicia na configuração do *baud rate* para o valor definido pelo protocolo.

Para o funcionamento básico da transmissão e recepção das mensagens de forma assíncrona é necessária a inserção de bits de inicialização e de término<sup>1</sup>. Sua inserção é feita pelo próprio *hardware* do módulo de comunicação serial. Como não temos uma sincronização entre os dispositivos, cada um utiliza de um *clock* interno de mesma frequência.

Os módulos operam com registradores de um byte para o envio e a recepção das mensagens. Para o envio de uma mensagem sua sequência de operação é bem simples. Quando uma mensagem é armazenada no registrador de envio a transmissão se inicia. Quando uma transmissão é finalizada uma *flag* é ativada para informar para o sistema que a transmissão fora concluída. Este passo é importante para que o tempo de envio seja respeitado e o sistema mande as informações corretamente.

Para a recepção de mensagens o procedimento é similar. Quando uma mensagem é recebida ela é armazenada no registrador de recepção e uma *flag* é ativada para informar que uma informação foi recebida. Nos casos de recepção o uso de interrupções se faz extremamente útil. Ao final do procedimento de recepção a mensagem pode ser salva em alguma variável.

---

<sup>1</sup> Algumas especificações de alguns protocolos não necessitam de um bit de término para a transmissão

## 4 RESULTADOS

Nesta seção os resultados obtidos durante a pesquisa serão apresentados seguindo a ordem relativa ao caminho que o sinal percorre.

- Seção 4.1: nesta seção os resultados relacionados à filtragem da entrada do sinal serão apresentados.
- Seção 4.2: os resultados da implementação de um retificador de meia onda estão definidos nesta seção.
- Seção 4.3: os resultados obtidos com a implementação de um circuito comparado serão apresentados durante esta seção.
- Seção 4.4: os resultados do desenvolvimento do *firmware* serão apresentados nesta seção.
- Seção 4.5: durante esta seção os resultados da integração do *hardware* serão apresentados.

### 4.1 Filtragem da entrada do Sinal

Os algoritmos de *zero crossing* apresentam limitações decorrentes da influência de harmônicos na identificação da frequência fundamental real.

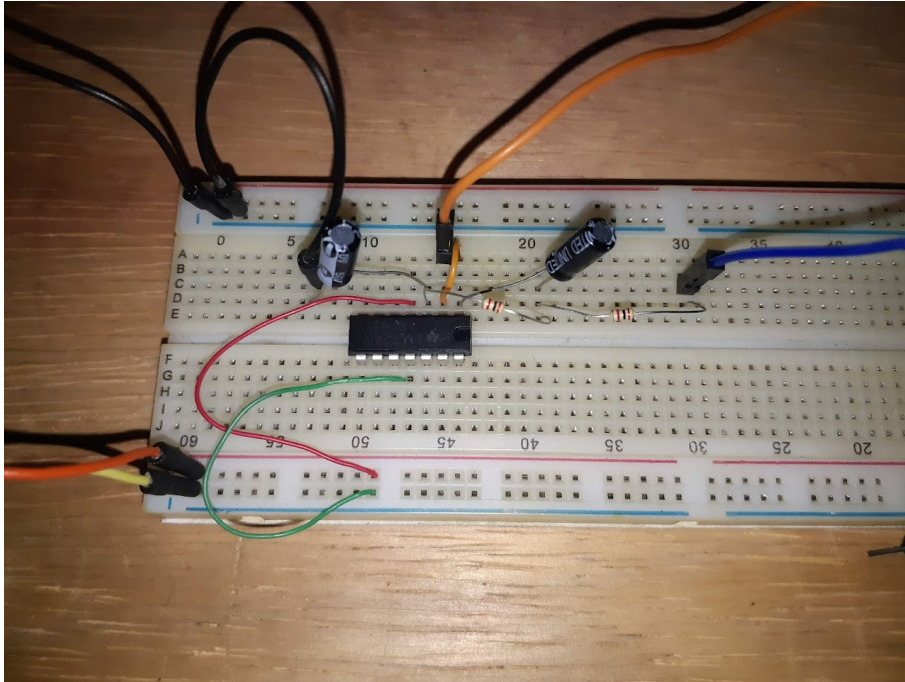
A introdução de um filtro passa baixa de entrada, sintonizado na frequência máxima de sinal desejada, ajudou a contornar tais limitações. Assim os componentes do filtro foram determinados para efetuar a filtragem de sinais acima de 800 Hz. Para facilitar os procedimentos de escolha dos componentes primeiro escolhemos o valor de capacitor para depois ser possível escolher dentre os valores de resistores. O capacitor escolhido fora o com capacitância de 0,1  $\mu\text{F}$ . Substituindo na Eq. 3.1 o valor de C temos:

$$800 = \frac{1}{2\pi R \times 0.1 \times 10^{-6}}$$

$$R \approx 1989.44\Omega \quad (4.1)$$

Um problema que é necessário considerar perante a resposta da Eq. 4.1 é que o valor encontrado não é um valor de resistência comercial, sendo assim o valor mais próximo escolhido fora o valor de 2 k $\Omega$ . Como o filtro escolhido fora o *Butterworth* necessitamos de dois capacitores e dois resistores de mesmo valor. Na Figura 24 podemos ver o circuito do filtro passa-baixa *Butterworth* de segunda ordem construído para o presente trabalho. O amplificador operacional LM324 com alimentação simétrica fora utilizado para elaboração do filtro.

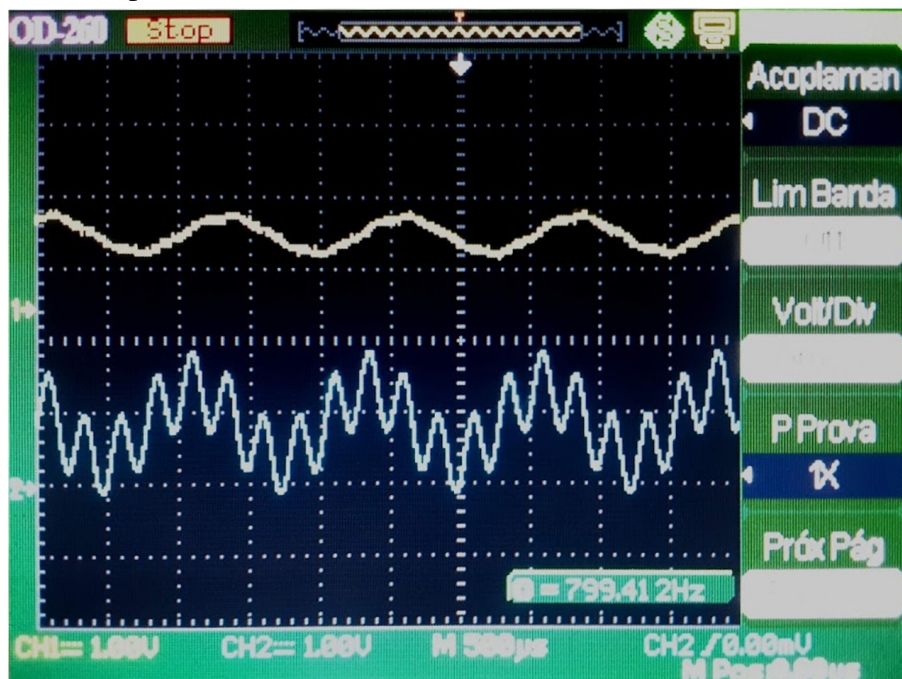
Figura 24 – Circuito do filtro *Butterworth* de segunda ordem para 800 Hz



Fonte – Gerado pelo Autor

O desempenho do filtro foi avaliado em bancada injetando um sinal de teste na entrada e monitorando o sinal de saída com um osciloscópio. Na Figura 25 é possível verificar em azul o sinal de entrada com seus harmônicos, causando diversos cruzamentos de zero e em amarelo o sinal de saída filtrado possuindo apenas a frequência fundamental.

Figura 25 – Resposta do filtro *Butterworth*

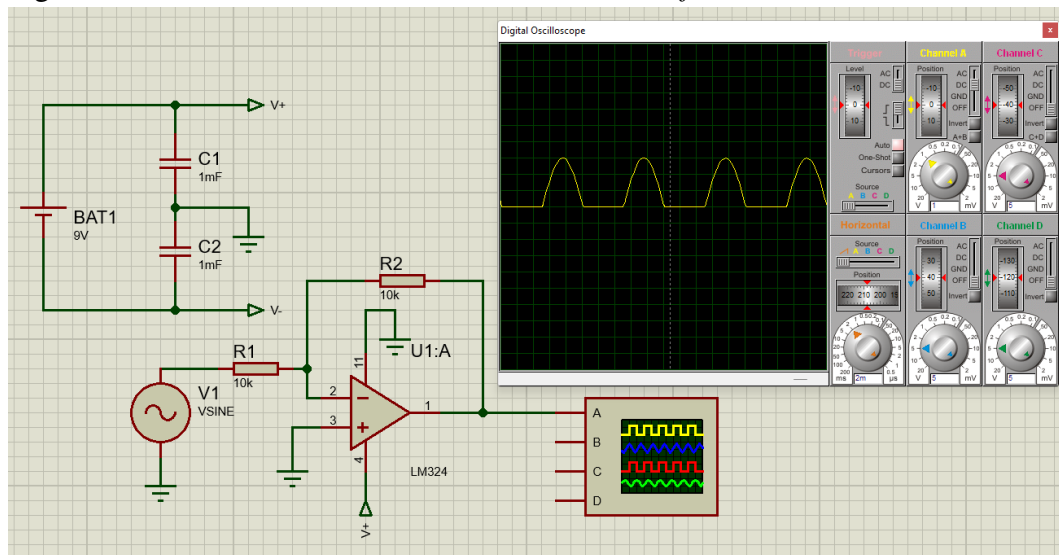


Fonte – Gerado pelo Autor

## 4.2 Retificador de Meia Onda

Como ferramenta para melhorar a aplicação do conversor AD foi necessária a implementação de um retificador de meia onda, permitindo que os 10 bits de resolução pudessem ser utilizados para captura de valores entre zero e o valor de pico do sinal. Este circuito é um dos componentes do pré-processamento do sinal. Para a elaboração do retificador de meia onda o *software* Proteus fora utilizado para realizar a simulação do circuito a ser projetado. Na Figura 26 podemos verificar o resultado dessa simulação. O sinal testado para efetuar a retificação fora uma onda senoidal. Como é possível verificar, a onda foi corretamente retificada, mantendo apenas sua parte positiva na saída.

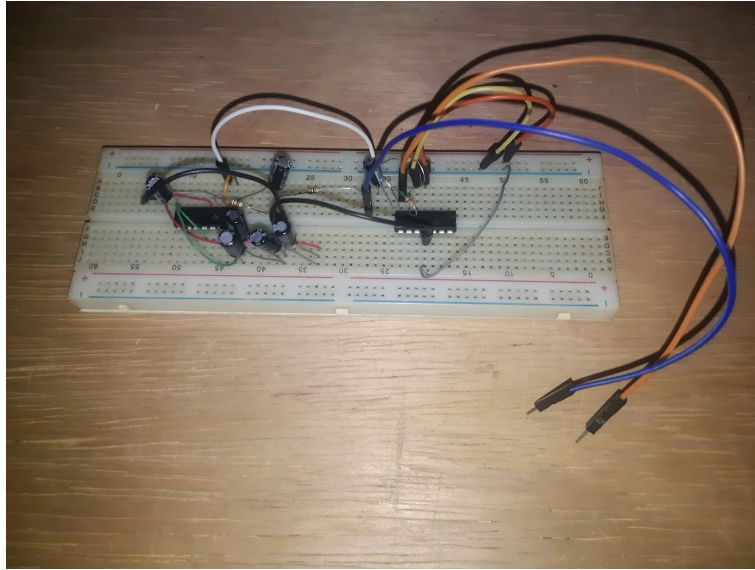
Figura 26 – Retificador de meia onda simulado no *software* Proteus



Fonte – Gerado pelo Autor

O amplificador operacional como retificador se mostrou eficaz até mesmo em sinais de baixa amplitude, quando comparado com técnicas mais comuns como a retificação com diodos, permitindo inclusive aplicar amplificação no sinal. A amplificação do sinal de entrada permitiu o ajuste de sua amplitude para que o valor máximo correspondesse ao valor máximo (5V) e mínimo (0V) do conversor AD. Para criá-lo fora utilizado o amplificador operacional LM324 com alimentação simples, ou seja, sua porta de referência negativa fora aterrada. Podemos verificar o seu circuito montado na Figura 27, este fora adicionado ao circuito de filtro.

Figura 27 – Retificador de meia onda adicionado ao Filtro

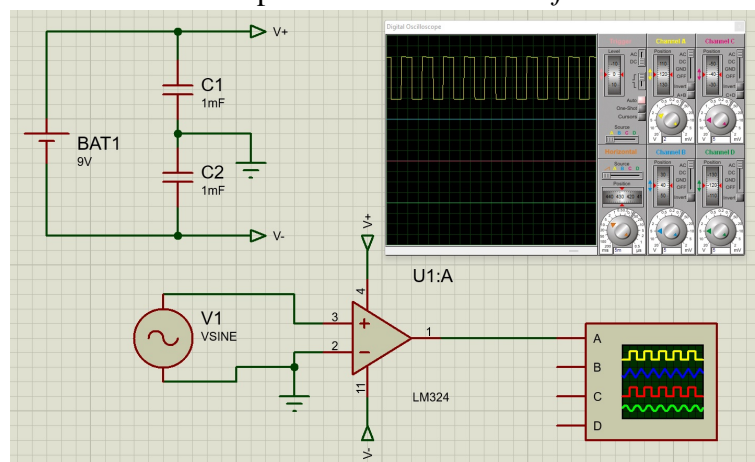


Fonte – Gerado pelo Autor

### 4.3 Circuito Comparador

Para aumentar a precisão da detecção *zero cross* foi empregado um circuito comparador com amplificador operacional. Como este circuito gera na sua saída formas de onda quadradas relativas ao sinal de entrada, torna este sinal mais fácil de ser detectado durante a aplicação do *zero cross*. Como é possível verificar na Figura 28 sua estrutura é bem simples, necessitando de poucas conexões e apenas um amplificador operacional alimentado.

Figura 28 – Circuito comparador simulado no *software* Proteus

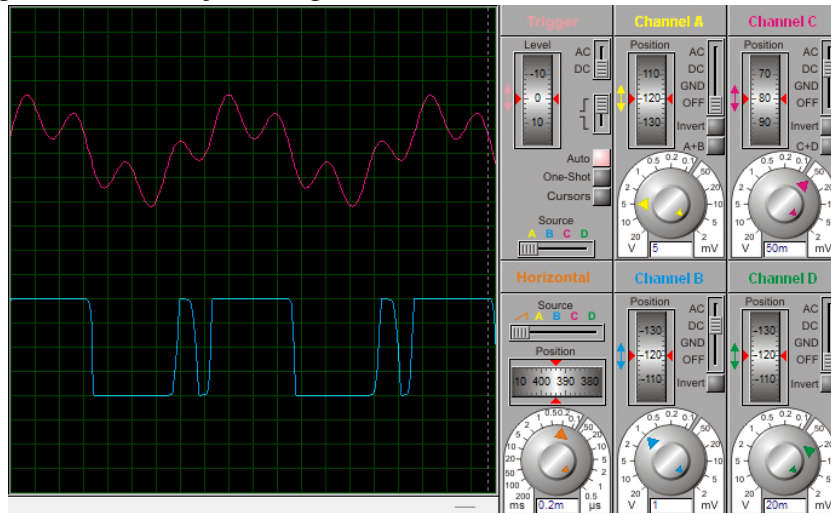


Fonte – Gerado pelo Autor

O circuito comparador quando aplicado para detecção *zero cross* necessita que na sua entrada esteja um sinal com ondas de formato próximo às ondas fundamentais, como: ondas senoidais, quadradas, triangulares, e dente de serra. Harmônicos presentes no sinal, aplicação

de acordes, ou alterações causadas por ruídos podem causar cruzamentos no zero irregulares. Nesses casos há a necessidade da aplicação de filtro de frequência. Quando aplicamos o circuito comparador em ondas que possuem harmônicos ou ruídos temos detecções irregulares no cruzamento. Esse caso pode ser verificado na Figura 29.

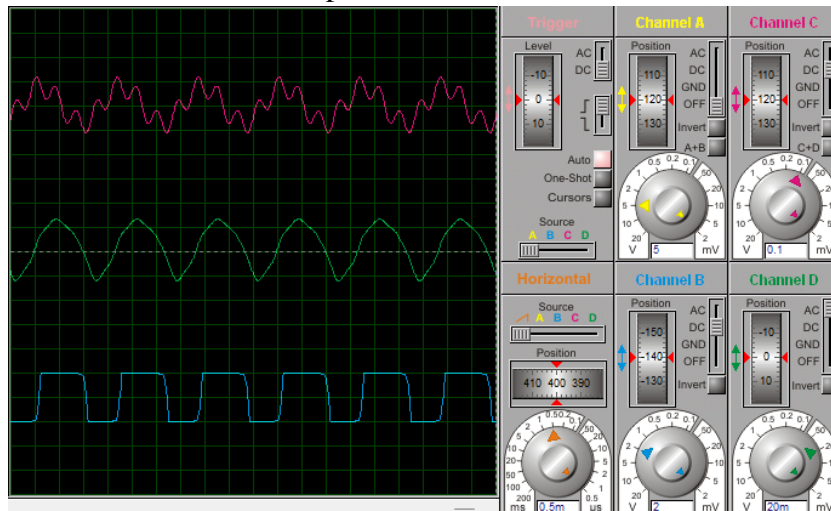
Figura 29 – Detecções irregulares do zero cross



Fonte – Gerado pelo Autor

Aplicando filtragem ao sinal de entrada no circuito comparador é possível corrigir a regularidade da detecção do cruzamento. Na Figura 30 podemos verificar que quando aplicado o filtro antes do circuito comparador conseguimos aproximar a onda de entrada de uma onda senoidal e com isso tornar a detecção de cruzamento regular, se assemelhando a uma onda quadrada.

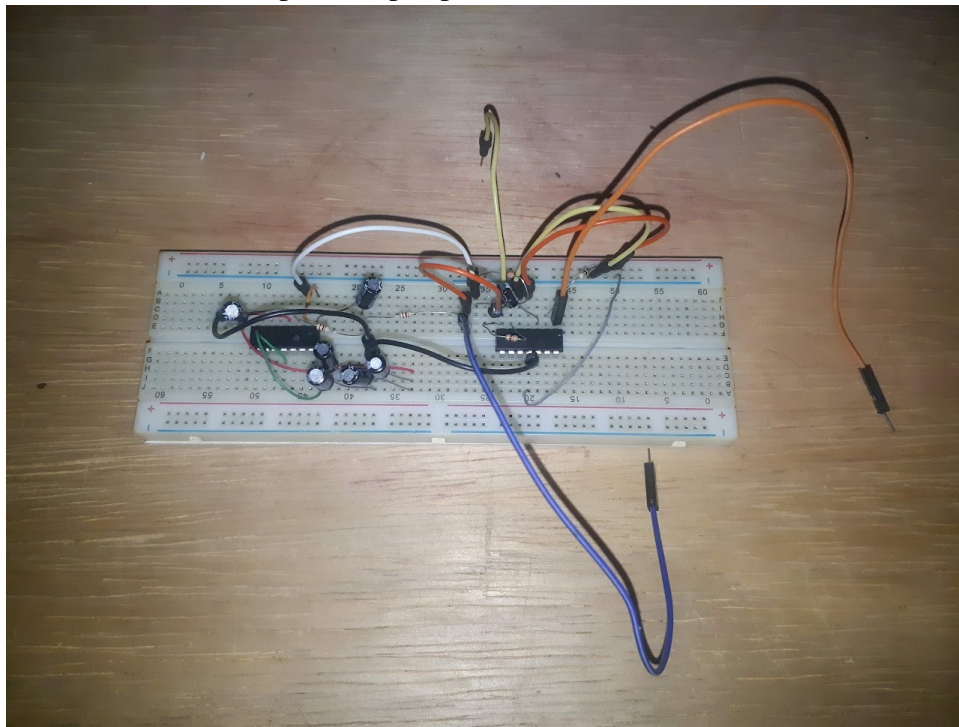
Figura 30 – Formas de onda da entrada do sinal, sinal filtrado e da saída do circuito comparador



Fonte – Gerado pelo Autor

O circuito completo de pré-processamento pode ser analisado na Figura 31. As saídas deste circuito vão para as entradas de detecção *zero cross* e para o conversor AD. O sistema proposto atingiu seu principal objetivo reduzindo os possíveis erros de detecção de frequência e permitindo uma detecção de amplitude correspondente à entrada do microcontrolador.

Figura 31 – Circuito completo de pré-processamento



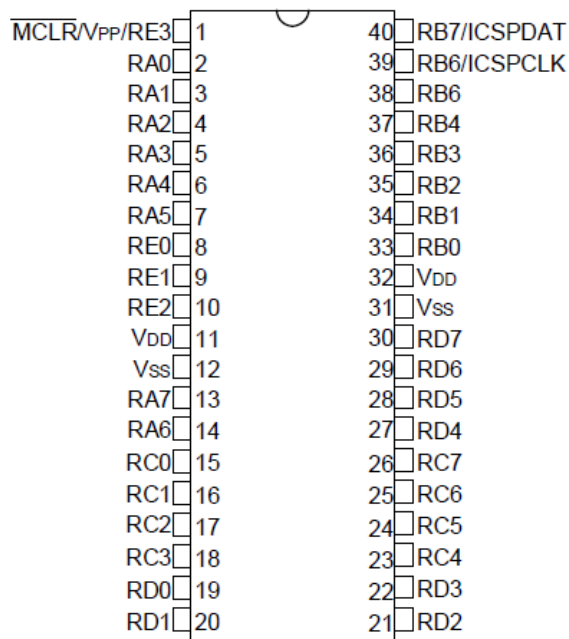
Fonte – Gerado pelo Autor

#### 4.4 Desenvolvimento do *Firmware*

O *firmware* desenvolvido trabalha com diversos recursos do microcontrolador abordado neste projeto. Este possui entre suas possibilidades de *clock* um interno de 64 MHz, eliminando a necessidade de circuito oscilador externo. Para o funcionamento básico dos principais módulos o uso de contadores de tempo se mostraram de extrema importância, pois muitos recursos do sistema proposto trabalham com contagem de tempo. Com isso foi verificado que o PIC18F47K40 possui sete módulos temporizadores, porém apenas dois foram utilizados no projeto, o *timer 1* e o *timer 3*, sendo ambos contadores de 16 bits. O *timer 1* foi projetado para servir como contador para o módulo ZCD e o *timer 3* foi projetado para ser utilizado em conjunto com um dos módulos contadores do microcontrolador para mandar as mensagens MIDI periodicamente.

Os três principais módulos utilizados para receber informações de entrada e enviar mensagens MIDI são os módulos ADC, ZCD e EUSART. Para os módulos ADC e ZCD uma função foi elaborada para cada um utilizando as equações 3.6 e 3.16. As duas funções elaboradas podem ser verificadas nos apêndices B e C deste trabalho. Estas funções efetuam a conversão dos valores das entradas para o protocolo MIDI. Para envio das mensagens MIDI para outros dispositivos o módulo EUSART foi configurado no modo assíncrono. Como o *baud rate* aplicado no protocolo MIDI pode ser dividido exatamente pelo *clock* empregado no microcontrolador a configuração do módulo EUSART se tornou bem mais simples.

Figura 32 – Diagrama dos pinos do PIC18F47K40



Fonte – (Microchip Technology Inc., 2018)

Os pinos aplicados para cada módulo foram determinados pelo *datasheet* do dispositivo (Microchip Technology Inc., 2018). Como apenas uma porta fora utilizada para o módulo ADC o pino 2 (RA0) fora utilizado para sua implementação. A entrada do módulo ZCD do dispositivo é aplicada no pino 33 (RB0). Como apenas a transmissão de dados pelo o módulo EUSART fora utilizado, temos como saída o pino 25 (RC6).

Os algoritmos produzidos foram testados em programas na linguagem de programação C. Alguns testes foram feitos aplicando frequências no algoritmo de conversão de frequência para protocolo MIDI e comparando com os valores da Tabela 2. Para os testes de amplitude outro programa foi feito e testado, para analisar se os valores eram correspondentes aos teóricos. Na Figura 33 podemos verificar alguns testes feitos utilizando os algoritmos propostos.



Figura 33 – Respostas dos algoritmos de conversão de frequência e amplitude para protocolo MIDI

```

digite a frequencia:      Digite o valor da entrada:
440                      1023
Nota MIDI:               Valor em dB e MIDI:
69                       11.999607, 125

digite a frequencia:      Digite o valor da entrada:
660                      512
Nota MIDI:               Valor em dB e MIDI:
76                       5.995955, 113

digite a frequencia:      Digite o valor da entrada:
220                      0
Nota MIDI:               Valor em dB e MIDI:
57                       -48.206390, 0

digite a frequencia:      Digite o valor da entrada:
330                      128
Nota MIDI:               Valor em dB e MIDI:
64                       -5.994597, 88

digite a frequencia:      Digite o valor da entrada:
115                      32
Nota MIDI:               Valor em dB e MIDI:
46                       -17.836113, 63

```

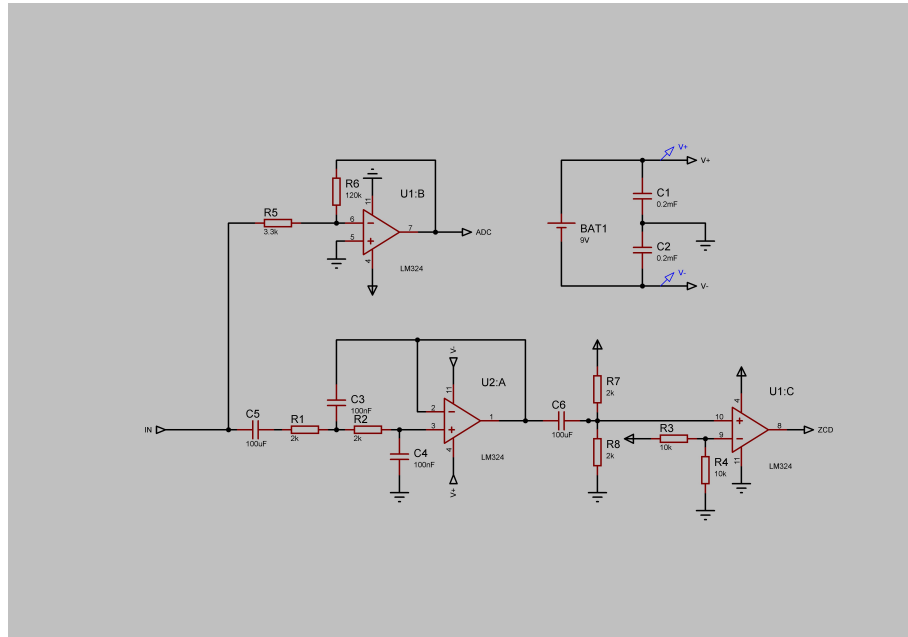
Fonte – Gerado pelo autor

O código se mostrou bastante robusto apresentando o resultado esperado dos algoritmos desenvolvidos. Como o sistema trabalha para frequências na faixa audível de 20 Hz a 20 kHz, os valores encontrados na conversão permitem uma precisão maior para baixas frequências e, considerando um temporizador de 16 bits, com precisão de 250 ns. A frequência mais baixa que o sistema é capaz de reconhecer é de 61 Hz, ou período de 16,384 ms. Todos os testes comprovaram a eficácia do algoritmo na conversão de frequência para MIDI e da amplitude do sinal para MIDI.

#### 4.5 Integração do *Hardware*

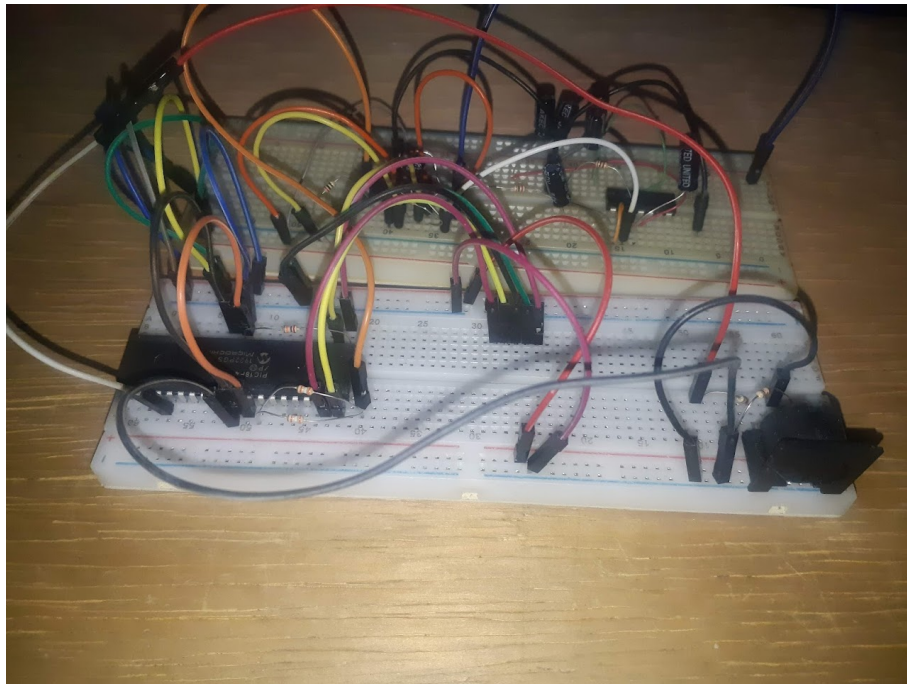
Por fim todas as partes do *hardware* foram integradas com o microcontrolador e o *firmware* desenvolvido, com as devidas conexões e a saída MIDI com o conector DIN5. Algumas considerações válidas quanto ao *hardware* desenvolvido e sobre as características do pino de entrada do módulo ZCD, que não permite a aplicação de altas tensões na entrada e de correntes acima de 300  $\mu$ A. Além disso o uso do comparador em pré-processamento se fez útil para corrigir problemas de detecção da frequência por conta de a detecção ser feita em 0,75 V. No caso de ondas senoidais o erro de detecção se tornou perceptível quando se utilizou a detecção na subida e na descida do sinal. Na Figura 34 podemos analisar o esquemático do circuito completo de pré-processamento enquanto na Figura 35 podemos verificar o projeto do *hardware* completo montado em uma *proto-board*.

Figura 34 – Esquemático completo do sistema de pré-processamento



Fonte – Gerado pelo Autor

Figura 35 – Circuito completo do sistema desenvolvido



Fonte – Gerado pelo Autor

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Os resultados apresentados por este projeto corresponderam às expectativas cumprindo os objetivos propostos. O algoritmo de detecção *zero cross* se mostrou bastante eficaz para algumas formas de onda, como formas de ondas similares às ondas fundamentais, como as senoides e ondas quadradas. Para as formas de onda que poderiam causar erros na detecção a aplicação de filtro se mostrou eficiente para aproximar o formato destas ondas ao das ondas fundamentais. As funções elaboradas durante a pesquisa se mostraram de grande valia para este projeto e para projetos envolvendo conversão de características de áudio para o protocolo MIDI e para outros tipos de comunicação entre instrumentos. Este projeto abriu portas para uso de instrumentos eletroacústicos para controle de instrumentos virtuais e de instrumentos eletrônicos. Apesar dos resultados positivos, uma limitação que ainda se mostrou presente foi na detecção de frequência fundamental para instrumentos que podem executar mais de uma nota simultaneamente, sendo necessária a captação individual para cada corda, no caso de instrumentos de corda.

Com os conhecimentos adquiridos a cerca do projeto, novos horizontes foram apresentados. Como possibilidade para trabalhos futuros temos a possibilidade de implementar algoritmos de inteligência artificial para predição de frequência fundamental. Estes algoritmos surgem como soluções para detecção de frequência fundamental em sinais mais complexos, além de permitir que o próprio sistema reconheça o instrumento que está sendo executado e permita selecionar algoritmos específicos para o instrumento identificado.

Além disso o presente projeto pode ser embarcado em diversos instrumentos, permitindo que estes passem a ter a capacidade de conversão de seus sinais em protocolo MIDI. A criação de novos instrumentos e de mais possibilidades se faz de grande importância para o desenvolvimento criativo de artistas.

O presente trabalho também pode ser aplicado em sistemas de inclusão social utilizando o protocolo MIDI para controle de luzes, permitindo que as frequências detectadas sejam relacionadas a projetos visuais utilizando cores específicas para relacionar às frequências, permitindo que pessoas com deficiência auditiva tenham uma nova forma de aproveitar uma performance musical.

## REFERÊNCIAS

- AMERICANAS. **Captador Piezo Pré-Instalado**. 2021. Disponível em: <[https://images\discretionary{-}{-}{-}americanas.b2w.io/produtos/589261860/imagens/captador-piezo-pre-instalado-com-botao-de-controle-de-volume-para-pecas-de-guitarra-de-caixa-de-charuto-589261860\\\_1\\\_xlarge.jpg](https://images.discretionary{-}{-}{-}americanas.b2w.io/produtos/589261860/imagens/captador-piezo-pre-instalado-com-botao-de-controle-de-volume-para-pecas-de-guitarra-de-caixa-de-charuto-589261860\_1\_xlarge.jpg)>. Acesso em: 05 jan. 2022.
- ARTURIA. **Keylab 88 MKII**. 2021. Disponível em: <<https://www.arturia.com/products/hybrid-synths/keylab-88-mkii-black/overview>>. Acesso em: 21 dez. 2021.
- BOULANGER RICHARD; LAZZARINI, V. M. M. **The Audio Programming Book**. Massachusetts: The MIT Press, 2011. ISBN 978-0-262-01446-5.
- BOYLESTAD ROBERT L.; NASHELSKY, L. **Dispositivos Eletrônicos: e teoria de circuitos**. 11. ed. São Paulo: Pearson Education do Brasil, 2013. ISBN 978-85-64574-21-2.
- DEARLING, R. **The Encyclopedia of Musical Instruments**. Nova York: Smithmark Editors, 1996.
- GOHN, D. M. A tecnologia na música. In: **XXIV Congresso Brasileiro da Comunicação**. Campo Grande: [s.n.], 2001.
- MED, B. **Teoria da Música**. 5. ed. Brasília, DF: Musimed, 2017. ISBN 978-85-7092-058-4.
- Microchip Technology Inc. **Datasheet: PIC18(L)F47K40**. 2018.
- NILSSON JAMES W.; RIEDEL, S. A. **Circuitos Elétricos**. 10. ed. São Paulo: Pearson Education do Brasil, 2015. ISBN 978-85-430-0478-5.
- OPPENHEIM ALAN V.; SCHAFER, R. W. B. J. R. **Discrete-Time Signal Processing**. 2. ed. New Jersey: Prentice-Hall, 1998. ISBN 0-13-754920-2.
- PINTEREST. **Casio DG-20: Digital guitar synthesizer MIDI controler**. 2019. Disponível em: <<https://br.pinterest.com/pin/12666442690486740/>>. Acesso em: 22 dez. 2021.
- PRACTICAL MUSICAL PRODUCTION. **How Audio Latency Can Affect Your Recording and Mixing**. 2021. Disponível em: <<https://www.practical-music-production.com/audio-latency/>>. Acesso em: 14 nov. 2021.
- Premier Guitar. **Optical Bass Pickup**. 2021. Disponível em: <<https://www.premierguitar.com/pro-advice/bass-bench/optical-bass-pickup>>. Acesso em: 10 jan. 2022.
- RATTON, M. B. **MIDI Total: fundamentos e aplicações**. 1. ed. Rio de Janeiro: Musica & Tecnologia, 2005. ISBN 85-89402-05-3.
- REDBULL. **How to Buy Modular Synth Guide**. 2019. Disponível em: <<https://www.redbull.com/int-en/how-to-buy-modular-synth-guide>>. Acesso em: 24 jan. 2022.
- ROLAND. **Produtos**. 2021. Disponível em: <<https://www.roland.com/global/products>>. Acesso em: 22 dez. 2021.
- SAM ASH. **Single Coil vs Humbucker: What's the difference**. 2019. Disponível em: <<https://www.samash.com/spotlight/single-coil-vs-humbucker-whats-the-difference/>>. Acesso em: 05 jan. 2022.

SHURE. **How to Choose the Right Mic**. 2012. Disponível em: <<https://www.shure.com/es-MX/desempeno-y-produccion/louder/shure-sound-tech-how-to-choose-the-right-mic>>. Acesso em: 23 dez. 2021.

Sunpower UK. **What is a Frequency Filter**. 2019. Disponível em: <<https://www.sunpower-uk.com/glossary/what-is-a-frequency-filter/>>. Acesso em: 10 jan. 2022.

THE NY THEREMIN SOCIETY. **History of the Theremin**. 2005. Disponível em: <<https://www.nythereminsociety.org/history-2>>.

WIKIPEDIA. **Electronic Musical Instrument**. 2021. Disponível em: <[https://en.wikipedia.org/wiki/Electronic\\_musical\\_instrument](https://en.wikipedia.org/wiki/Electronic_musical_instrument)>. Acesso em: 24 jan. 2022.

WIKIPEDIA. **Notação Científica de Altura**. 2021. Disponível em: <[https://pt.wikipedia.org/wiki/Notaç~ao\\_científica\\_de\\_altura](https://pt.wikipedia.org/wiki/Notaç~ao_científica_de_altura)>. Acesso em: 21 dez. 2021.

WIKIPEDIA. **Tonewheel**. 2021. Disponível em: <<https://en.wikipedia.org/wiki/Tonewheel>>. Acesso em: 24 jan. 2022.

YAMAHA. **Digital Wind Instruments**. 2019. Disponível em: <[https://usa.yamaha.com/products/musical\\_instruments/winds/digital\\_wind\\_instruments](https://usa.yamaha.com/products/musical_instruments/winds/digital_wind_instruments)>. Acesso em: 22 dez. 2021.

ZUBEN PAULO; CAZNOK, Y. **Música e tecnologia: o som e seus novos instrumentos**. São Paulo: Irmãos Vitale, 2004. ISBN 978-85-740-7178-7.

## APÊNDICE A – TERMINOLOGIA

Este apêndice do trabalho visa abordar alguns termos em música que serão relevantes para um melhor entendimento dos capítulos seguintes. Alguns termos abordados como: oitavas, nota, acordes, temperamento, entre outros serão apresentados.

1. Nota: som com uma frequência específica. Aparece uma vez por oitava.
2. Oitava: distância de um ciclo de notas antes de sua repetição. Usualmente é também definida como a extensão entre uma frequência específica e seu valor dobrado.
3. Escala Musical: sequência definida de notas dentro de uma oitava.
4. Acorde: conjunto de notas tocadas simultaneamente.
5. Dinâmica: em música se trata da variação de volume sonoro
6. Temperamento: em música trata de sistemas que padronizam como as oitavas são divididas. O temperamento mais utilizado é o igual, no qual as notas possuem a mesma distância entre elas

## APÊNDICE B – CÓDIGO DE CONVERSÃO DE PERÍODO PARA MIDI

A função abaixo é utilizada para converter o período do sinal analisado para nota MIDI. Importante citar que na linha 8 a função de pegar o período e transformar em frequência está multiplicado por 1000 por conta da precisão do tipo *double* em C.

Código-fonte 1 – Função de conversão de período para MIDI

```

1 void tempToMIDI(int* note, int* pitchM, int* pitchL, double
    temp){
2     double tone = 0; // Frequencia
3     float pError = 0; // Desvio da nota em decimais
4     int mPitchLSB = 0; // Byte menos significante do "Pitch
        wheel"
5     int mPitchMSB = 0; // Byte mais significante do "Pitch
        wheel"
6     int mNote = 0; // Nota MIDI
7
8     tone = 1000.0/temp; // adquire a frequencia a partir do
        periodo
9
10    // Calcula o valor convertido da frequencia para midi com
        parte decimal
11    pError = 12*(log2((tone/440.0))) + 69.0;
12
13    // Separa a parte inteira da decimal
14    mNote = (int) pError;
15    pError -= (float) mNote;
16
17    // Normaliza a nota e o erro perante o desvio em decimais
18    if(pError>0.5){
19        pError -= 0.5;
20        mNote++;
21    }else pError += 0.5;

```

```
22
23 // Separa a informacão em 7bits mais e 7bits menos
    significativos
24 mPitchMSB = (int) 16384*pError;
25 mPitchLSB = mPitchMSB%128;
26 mPitchMSB = mPitchMSB/128;
27
28
29 *note = mNote; // Salva a informacão da nota MIDI
    no endereço de memoria note
30 *pitchM = mPitchMSB; // Salva a informacão da nota
    MIDI no endereço de memoria pitchM
31 *pitchL = mPitchLSB; // Salva a informacão da nota
    MIDI no endereço de memoria pitchL
32 }
```



## APÊNDICE C – CÓDIGO DE CONVERSÃO DE AMPLITUDE PARA MIDI

A função abaixo é utilizada para converter a amplitude do sinal analisado para *velocity* MIDI. O sinal fora normalizado entre 0 e 1 pois a análise é feita em meia onda.

Código-fonte 2 – Função de conversão de amplitude para MIDI

```
1 int ADToMIDI_DC(int in){
2     // valor máximo ADC 1111111111b ou 0x3FF ou 1023
3     // range velocity MIDI 0 - 127
4
5     float v = (in+1) / 1024; // normaliza o sinal entre 0 e
6     1
7     v = 20*log10((v/0.2512)); // converte a amplitude para
8     decibel
9
10    // range de v -48.156312848 dB - 12.049686285 dB
11    int out = (int) ((v+48)*2.1);
12    return out;
13 }
```