

CENTRO UNIVERSITÁRIO DE ANÁPOLIS - UNIEVANGÉLICA
BACHARELADO DE ENGENHARIA DE COMPUTAÇÃO

DANIEL VAZ DE OLIVEIRA
JÔNATAS GABRIEL DA SILVA SANTOS

DISPOSITIVO ELETRÔNICO
AUTOMATIZADO PARA AFINAR VIOLÃO

ANÁPOLIS - GO

2019

DANIEL VAZ DE OLIVEIRA
JÔNATAS GABRIEL DA SILVA SANTOS

**DISPOSITIVO ELETRÔNICO
AUTOMATIZADO PARA AFINAR VIOLÃO**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a conclusão da disciplina de Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador: Prof. William Pereira dos Santos Júnior

ANÁPOLIS - GO

2019

**DANIEL VAZ DE OLIVEIRA
JÔNATAS GABRIEL DA SILVA SANTOS**

**DISPOSITIVO ELETRÔNICO AUTOMATIZADO PARA AFINAR
VIOLÃO**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 14 de junho de 2019, composta por:



William Pereira dos Santos Júnior
Presidente da Banca



Alexandre Moraes Tannus
Prof(a). Convidado(a)



Luciana Nishi
Prof(a). Convidado(a)

RESUMO

O propósito deste trabalho acadêmico é a prototipação de um dispositivo eletrônico, que realize a tarefa de afinar um violão automaticamente, e a verificação da eficácia desse dispositivo, fazendo uso dos conceitos da área da tecnologia digital. Esse protótipo será construído utilizando uma placa com microcontrolador (Arduino), e outros componentes eletrônicos, sendo os principais: o sensor piezoelétrico, e o motor de passo. O processo de funcionamento do dispositivo consiste em captar a vibração mecânica produzida pelas cordas do violão, e transformá-las em sinal elétrico, através do sensor piezoelétrico. Este sinal pode ser analisado pelo programa instalado no microcontrolador da placa Arduino. O programa trabalha fazendo a análise da frequência de alternância deste sinal, comparando o seu valor com os valores conhecidos e estabelecidos para as notas das cordas do violão. A partir desta análise, o programa envia sinais de controle a um motor elétrico, ligado diretamente à tarraxa do violão, fazendo este alterar a tensão da corda, aumentando ou diminuindo-a conforme necessário; até esta vibrar na frequência de valor exato, igual ao parametrizado para a nota referente a ela.

Afinar um violão é uma tarefa imprescindível para que o instrumento possa ser usado emitindo corretamente seu som, porém é uma tarefa manual e não simples de fazer. Existem dispositivos eletrônicos que auxiliam nesta tarefa, apontando se as cordas do instrumento estão afinadas, mas ainda assim, a afinação é realizada manualmente. Partindo do ponto que a tecnologia digital tem contribuído com a automação de muitas tarefas, a automação da afinação de um violão se mostra uma tarefa importante, e a prototipação de um dispositivo que realize a afinação de forma automatizada explorando a área da computação é a razão deste trabalho.

Embora já existam alguns dispositivos que realizem a afinação de forma automatizada, estes possuem custo elevado se comparado ao custo do próprio instrumento, e ainda, estes dispositivos não estão presentes no mercado brasileiro.

Os objetivos deste trabalho são: o desenvolvimento de um protótipo de dispositivo e com o menor custo possível, visando ser mais barato em relação aos dispositivos existentes e verificar a acurácia do dispositivo para que seja útil e confiável para o uso.

PALAVRAS CHAVE: Afinador de Violão, Motor de Passo, Arduino

ABSTRACT

The purpose of this academic work is the prototyping of an electronic device, which performs the task of tuning a guitar automatically, and the verifying of the effectiveness of this device, making use of the concepts of the area of digital technology. This prototype will be built using a board with microcontroller (Arduino), and other electronic components, the main ones being: the piezoelectric sensor, and the stepper motor. The process of operating the device consists of capturing the mechanical vibration produced by the guitar strings and transforming them into an electric signal through the piezoelectric sensor. This signal can be analyzed by the program installed on the microcontroller of the Arduino board. The program works by analyzing the frequency of alternation of this signal, comparing its value with the known and established values for the notes of the guitar strings. From this analysis, the program sends control signals to an electric motor, connected directly to the guitar's peg, causing it to change the tension of the string, increasing or decreasing it as needed, until it vibrates at the exact value frequency, equal to the parameterized for the note referring to it.

Tuning a guitar is an essential task to be possible that the instrument can be used emitting its correctly sound, but it is a manual task that is not simple to do. There are electronic devices that assist in this task by pointing if the strings of the instrument are tuned, but still the tuning is performed manually. Starting from the point that digital technology has contributed to the automation of many tasks, the automation of the tuning of a guitar it proves important, and the prototyping of a device that performs the tuning in an automated way exploring the area of the computation is the reason of this work.

Although there are already some devices that perform the tuning in an automated way, these have a high cost if compared even with the cost of the instrument itself, and still, these devices are not present in the Brazilian market.

The objectives of this work are: the development of a prototype of device with the lowest possible cost, aiming to be cheaper compared to existing devices and checking the accuracy of the device - whether it is useful and reliable for use.

KEYWORDS: Guitar Tuner, Stepper Motor, Arduino

SUMÁRIO

1	INTRODUÇÃO	8
1.1	PESQUISAS, PROJETOS E PRODUTOS RELACIONADOS	9
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	MÚSICA	12
2.2	VIOLÃO	13
2.3	TECNOLOGIA	15
2.3.1	ARDUINO	15
2.3.1.1	DADOS TÉCNICOS	16
2.3.2	COMPONENTES PRINCIPAIS	17
2.3.2.1	TRANSDUTOR PIEZOELÉTRICO	17
2.3.2.2	MOTOR DE PASSO	18
2.3.3	PROCESSAMENTO DE ÁUDIO	19
3	DESENVOLVIMENTO	20
3.1	MÓDULOS DE IMPLEMENTAÇÃO	21
3.1.1	SISTEMA SENSOR	22
3.1.2	SISTEMA DE PROCESSAMENTO	29
3.1.2.1	ALGORITMO DE DETECÇÃO DE FREQUÊNCIA	29
3.1.2.2	TESTES	33
3.1.2.3	CIRCUITO	39
3.1.3	SISTEMA MOTOR	41
3.1.3.1	MOTOR DE PASSO: UNIPOLAR X BIPOLAR	41
3.1.3.2	ENCAIXE	44
3.1.3.3	COMPONENTES	46
3.2	CUSTO DO PROTÓTIPO	46
4	CONSIDERAÇÕES FINAIS	48
	REFERÊNCIAS BIBLIOGRÁFICAS	49
	APÊNDICE A – PRIMEIRO ALGORITMO DO PROTÓTIPO COM ADAPTAÇÃO DO ALGORITMO DE KELLY	ERRO! INDICADOR NÃO DEFINIDO.
	APÊNDICE B – CALCULAR TAXA DE AMOSTRAGEM NORMAL	ERRO! INDICADOR NÃO DEFINIDO.
	APÊNDICE C – CALCULAR TAXA DE AMOSTRAGEM COM FUNÇÃO DE INTERRUPÇÃO	ERRO! INDICADOR NÃO DEFINIDO.

**APÊNDICE D – SEGUNDO ALGORITMO DO PROTÓTIPO COM
ADAPTAÇÃO DO ALGORITMO DE KELLY ERRO! INDICADOR NÃO DEFINIDO.**

**ANEXO A – ALGORITMO DE GHASSAEI ERRO! INDICADOR NÃO
DEFINIDO.**

ANEXO B – ALGORITMO DE KELLY ERRO! INDICADOR NÃO DEFINIDO.

1 INTRODUÇÃO

A afinação de um violão é de grande importância para a execução de uma música, pois sem afinação, o instrumento não consegue produzir o som correto de cada nota (GRÜNIG, 2014). A afinação é uma tarefa que exige maior conhecimento e prática por parte do músico, e para auxiliar nessa tarefa, já foram criados diversos dispositivos afinadores que apontam se o instrumento está afinado ou não (GRÜNIG, 2014), mas, mesmo com o uso desses dispositivos, a afinação pode não ter precisão, devido a interferências externas ou falta de prática por parte do usuário. Além disso, ter que afinar um violão pode atrasar uma apresentação de um músico profissional, ou até levar um músico iniciante a desistir do instrumento, por não conseguir afiná-lo.

Devido a esses problemas, surgiu uma nova categoria de afinadores. Estes são encaixados ou presos ao corpo do instrumento, detectam as notas emitidas pelas cordas do violão, e de forma automatizada realizam a afinação. Porém, esses dispositivos afinadores automatizados têm custo relativamente elevado e não estão disponíveis no mercado brasileiro.

Levando em consideração esses motivos, como desenvolver um dispositivo afinador automatizado e preciso, com preço e disponibilidade acessíveis?

O objetivo deste projeto é prototipar um dispositivo para realizar a afinação automática de um violão, com menor custo possível, e verificar a acuracidade do dispositivo – se é útil e confiável para o uso.

Partindo do ponto que a tecnologia digital é amplamente utilizada para simplificar o trabalho do homem em seus diversos aspectos, utilizá-la também para a tarefa de afinar um violão é mais uma contribuição para o avanço tecnológico. Ainda que já existam dispositivos afinadores automatizados, eles fazem parte de uma nova geração do segmento de dispositivos musicais eletrônicos.

O desenvolvimento desse dispositivo visa minimizar dois problemas-chave, encontrados, tanto por músicos iniciantes, quanto pelos que já tocam violão há muito tempo, na hora de afinar um violão. São eles: o tempo empregado na afinação e a acurácia da afinação.

O tempo empregado na afinação pode ser um empecilho para um músico diante de uma apresentação. Não que isto seja um descuido dos músicos, mas um problema muito comum acontece quando um músico sai com seu instrumento de um ambiente para outro – o instrumento pode simplesmente desafinar nessa transição por causa da diferença de temperatura entre os dois ambientes. A dilatação térmica altera o comprimento das cordas do

violão, fazendo-o desafinar (BERNI, 2016). Utilizando o dispositivo afinador automatizado, um músico nesta situação poderia facilmente resolver o problema da afinação e dar início à sua apresentação.

Segundo o Dicio, um dos significados de acurácia é: “Relação de proximidade entre o resultado alcançado, de modo experimental, e o real valor obtido por uma grandeza física.”. Desta forma, a imprecisão da afinação pode causar uma situação, até pior que o problema acima, pois um músico que acredita ter seu instrumento afinado, ao começar a tocar, pode perceber diferenças no som do instrumento, causada pela desafinação, ou num caso possível, má afinação do instrumento. Isso pode acontecer quando se afina “de ouvido”, ou com um afinador de baixa qualidade. A proposta do dispositivo afinador automatizado é garantir sua acurácia na afinação.

De maneira geral, este dispositivo afinador automatizado vem a somar na exploração digital na área musical, utilizando a contribuição do poder computacional.

1.1 PESQUISAS, PROJETOS E PRODUTOS RELACIONADOS

Estudos relacionados a essa área de conhecimento ocorreram principalmente na última década. Foram realizados diversos trabalhos realizados por diferentes autores, que abordaram o uso de Processamento Digital de Sinais (Digital Signal Processing - DSP) a fim de interpretar frequências de um som, ou até mesmo afinar de forma automatizada um instrumento (BECCHI, 2017).

Muitos desses trabalhos foram realizados somente como um estudo sobre detecção de frequências, não chegando a serem implementados em ambiente físico com uma placa microcontroladora como o Arduino. Aqueles que fizeram a implementação física do projeto tiveram diversos problemas com a imprecisão da detecção.

Abaixo serão apresentados alguns destes trabalhos no campo de pesquisa e também alguns produtos do campo comercial. Em diversos trabalhos realizados os autores não obtiveram resultados tidos como satisfatórios.

Em 2009 Lourde R. e Saji (2009 apud BECCHI, 2017, p. 29) propuseram um sistema de afinação motorizada de violão, para o qual desenvolveram um programa utilizando o software MATLAB. Utilizaram a técnica da Transformada Rápida de Fourier (Fast Fourier Transform - FFT) e a análise do espectro das frequências para determinarem a frequência dominante, mesmo quando havia outra frequência diferente com amplitude maior.

Ainda em 2009 Code (2009 apud BECCHI, 2017, p. 29) desenvolveu um algoritmo em C capaz de detectar a frequência fundamental de um fragmento de áudio previamente capturado. Assim como Lourde R. e Saji, utilizou a técnica FFT.

Em 2010 Deambulatory (2010 apud BECCHI, 2017, p. 29) desenvolveu um algoritmo de detecção de frequência baseado em autocorrelação. O grande diferencial para os outros dois anteriores foi a utilização do Arduino. Segundo ele, o programa apresentou bons resultados na estimativa de frequência fundamental de um sinal.

Em 2013 Salvi (2013 apud BECCHI, 2017, p. 30) apresentou alguns fundamentos para criação de um sistema de análise de áudio genérico para Arduino, em que são exibidas diversas técnicas de condicionamento de sinal, envolvendo amplificadores operacionais, transistores e circuitos específicos para microfones de eletreto.

Ainda em 2013, Lima (2013 apud BECCHI, 2017, p. 30) desenvolveu um sistema de afinação para violão utilizando de um componente chamado microfone de eletreto, que possui um circuito similar ao de um transdutor piezoelétrico, mas possui uma saída de 20 milivolts. Baseando-se no algoritmo de Goertzel, o autor desenvolveu um programa no MATLAB. O sistema dele consistia em fazer a leitura das frequências captadas, e compara-las com uma das seis frequências da afinação padrão. O código foi validado dentro do ambiente do MATLAB com auxílio de um software de afinação, mas quando foi colocado dentro do Arduino, o mesmo não apresentou resultados satisfatórios.

Em 2015 Vonk (2015 apud BECCHI, 2017, p. 30) desenvolveu um sistema de detecção de frequência de um som baseando-se em autocorrelação. Foram realizados testes em um piano e clarinete, apresentando boas respostas na faixa média de frequências, entretanto os sons mais agudos foram captados como se fossem oitavas mais graves, e quando foi colocado para oitavas mais graves como a segunda oitava, presente nas cordas quinta e sexta do violão, o mesmo não retornou nenhuma resposta.

Em 2017 Becchi propôs um sistema de afinação automatizada que recebe o sinal diretamente da saída de áudio de um violão. Foi utilizado o cálculo das Transformadas Rápidas de Fourier (FFT) para detecção das frequências, porém esse método necessita de grande processamento por parte do Arduino, o que fez o sistema demonstrar resultados insatisfatórios quando foi colocado em prática em um meio físico com a placa Arduino Mega, porém a metodologia aplicada conseguiu resultados satisfatórios no software MATLAB, e também em um sistema Linux, já que ele não tem as mesmas limitações da placa Arduino utilizada.

Ainda em 2017 Kelly criou um algoritmo de detecção de frequências baseado em técnicas Processamento Digital de Sinais (DSP), mas ao implementar fisicamente o projeto, enfrentou problemas com a capacidade de processamento do Arduino, o que fez com que seu projeto não trouxesse valores exatos, mas sim aproximados. Kelly relatou em seu trabalho que calculou as frequências com um valor de taxa de amostragem parametrizado, que fora medido de um modo não especificado, e com essa taxa de amostragem especificada, o resultado da detecção da frequência do sinal apresentou valores não exatos.

Partindo para a área comercial, os modelos de afinadores automáticos já podem ser encontrados no mercado exterior, como o Roadie Tuner, que possui um dispositivo com um motor, que o músico encaixa na tarraxa de seu violão ou guitarra no momento da afinação (BAND, 2016); e o Min-ETune, composto de seis motores fixos que ficam acoplados junto às tarraxas do violão ou da guitarra (EPIPHONE, 2014).

Apesar de já existirem produtos no segmento, eles não chegaram ao mercado brasileiro até a data de publicação deste trabalho, o que torna difícil o acesso a eles, além do que o preço destes dispositivos fica bastante elevado por conta da importação.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão expostos conceitos teóricos sobre violão, música, tecnologia, e processamento dos dados envolvidos no desenvolvimento deste projeto.

2.1 MÚSICA

Segundo Roederer (1998 apud BECCHI, 2017, p. 14), “a percepção da música está essencialmente baseada no processamento de informações acústicas. Constituída pela combinação de vários sons e ritmos, a música pode ser definida como uma sobreposição bem organizada de várias frequências”.

No total existem 12 notas musicais, são sete fundamentais e mais cinco intermediárias. As fundamentais recebem os nomes Dó, Ré, Mi, Fá, Sol, Lá e Si ascendentemente, e podem ser representadas pelas letras C, D, E, F, G, A e B respectivamente.

As notas musicais são contínuas, ou seja, sempre após a última nota (nota Si), tem-se novamente a primeira nota (nota Dó). O agrupamento das 12 notas musicais é chamado de “oitava”. Sendo assim, após o término de uma oitava, tem-se o início de outra.

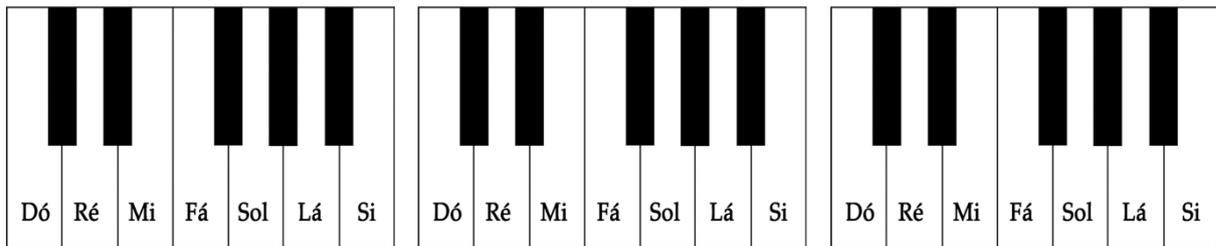
As notas musicais intermediárias localizam-se entre as notas fundamentais (exceto entre as notas Si e Dó, e Mi e Fá). As notas intermediárias possuem dois nomes cada uma. Por exemplo: a nota que fica entre Ré e Mi recebe os nomes Ré Sustenido (D#) ou Mi Bemol (Eb).

O que distingue cada nota é a frequência do som emitido. A nota Lá, por exemplo, possui na segunda oitava a frequência de 110 hertz. Existe uma relação matemática entre as oitavas, de modo que uma nota possui na oitava anterior, metade da frequência, e o dobro da frequência na oitava posterior. Para exemplificar, a nota Lá, que possui frequência de 110 hertz na segunda oitava, tem 55 hertz na oitava anterior, e 220 hertz na oitava posterior (ROEDERER, 1998 apud BECCHI, 2017, p. 15).

Na figura 1, pode-se observar três agrupamentos de notas (oitavas) exemplificados no teclado de um Piano. As teclas pretas representam as notas intermediárias, e as notas brancas, as fundamentais.

Como forma de diferenciar as notas musicais entre oitavas, é acrescentado um sufixo às notas. Por exemplo, C1 para a nota Dó (C), que está localizada na primeira (1) oitava. A3 para a nota Lá, que está localizada na terceira oitava. Por exemplo: Nota A2 = 110 hertz (Lá da segunda oitava); nota A3 = 220 hertz (Lá na terceira oitava).

Figura 1 – Três oitavas em teclas de Piano



Fonte: Os próprios autores (2018)

Como forma de diferenciar as notas musicais entre oitavas, é acrescentado um sufixo às notas. Por exemplo, C1 para a nota Dó (C), que está localizada na primeira (1) oitava. A3 para a nota Lá, que está localizada na terceira oitava. Por exemplo: Nota A2 = 110 hertz (Lá da segunda oitava); nota A3 = 220 hertz (Lá na terceira oitava).

É definido pela ISO 16:1975 que a nota Lá da quarta oitava (A4), possui frequência de 440 hertz, e é a nota a partir da qual se calcula a frequência das demais, seguindo a equação 1 (VONK, 2015):

$$f = 2^{\frac{n}{12}} \times 440$$

Onde:

f é a frequência da nota musical em hertz;

n é o número (positivo ou negativo) da nota a partir de A4.

2.2 VIOLÃO

O violão, assim chamado no Brasil, conhecido como guitarra clássica, ou guitarra acústica em outros países, é um instrumento musical de cordas. Os modelos de violão possuem seis ou sete cordas; cordas que podem ser de nylon ou de aço. O violão possui um corpo oco, e o som é produzido através das vibrações das cordas. As ondas propagadas pelas cordas no ar em seu redor refletem dentro do corpo do violão, amplificando o seu som (PERUCHI, 2014). Cada corda emite uma nota musical específica, que está obrigatoriamente relacionada à casa pressionada pelo músico, no braço do violão. O que define a nota emitida pelo violão, é a frequência da onda propagada pelo movimento de vibração da corda, de modo que, se a corda for afrouxada, a frequência será diminuída; e se for apertada, a frequência será

umentada. As cordas são únicas, e se diferem na espessura, assim, emitindo notas musicais diferentes, mesmo quando são submetidas à mesma tração (BECCHI, 2017, p. 16 apud BRAIN, 2014).

A afinação da corda de um instrumento consiste em girar uma tarraxa que regula a tensão mecânica da corda, assim aumentando ou diminuindo a frequência que a mesma produz ao ser tocada. O processo é repetido para todas as cordas - seis, no caso do violão (BECCHI, 2017, p. 11 apud ROEDERER, 1998; BRAIN, 2014).

As frequências conhecidas das seis cordas de um violão são demonstradas na tabela 1.

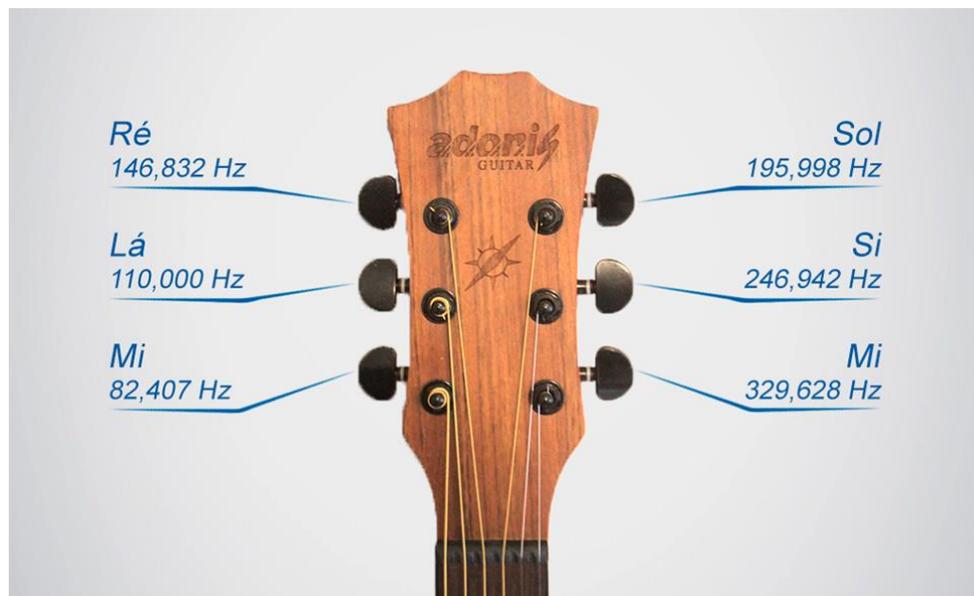
Tabela 1 – Frequência das notas das seis cordas de um Violão

Corda	Nota	Oitava	Nomenclatura	Frequência em hertz
6ª	Mi	2ª	E2	82,407
5ª	Lá	2ª	A2	110,000
4ª	Ré	3ª	D3	146,832
3ª	Sol	3ª	G3	195,998
2ª	Si	3ª	B3	246,942
1ª	Mi	4ª	E4	329,628

Fonte: Os próprios autores (2018)

A localização de cada uma das cordas do violão são demonstradas na Figura 2.

Figura 2 – As Cordas do Violão



Fonte: Os próprios autores (2018)

2.3 TECNOLOGIA

Nos subtópicos seguintes tratam dos conceitos teóricos sobre tecnologia e processamento de dados envolvidos no desenvolvimento deste projeto.

2.3.1 ARDUINO

No contexto atual, existem bastantes opções de placas controladoras e computadores de placa única para prototipação de projetos. Dentre muitas opções, as que mais se destacam são Arduino e Raspberry Pi. O Raspberry Pi é “[...] um sistema de circuito integrado ‘on-in-chip’, ou SOC. Ele executa uma versão completa do Linux e é projetado para ajudar a ensinar você como desenvolver suas ideias.” (SATO, 2018). Já o Arduino é “[...] um microcontrolador [...] e tem uma comunidade de suporte enorme, bem como centenas de opções de expansão e periféricos compatíveis.” (SATO, 2018).

De acordo com a Raspberry Pi Foundation (2011), Raspberry Pi é um computador do tamanho de um cartão de crédito. Com seu alto poder de processamento, é capaz de executar o sistema Linux a partir de um cartão de memória. Ele pode ser usado de diversas maneiras para a realização de tarefas, criação de projetos, construção de robôs e em muitas outras possibilidades. É um computador que num projeto de automação suportaria processar muitas informações paralelamente, mesmo com algoritmos que demandariam grandes recursos de sistema.

Do projeto do dispositivo afinador automatizado, as principais etapas da parte técnica, são: captação, processamento e resposta. Para tal projeto, não é necessário grande poder computacional para processar a entrada e saída de informações, visto que, o dispositivo apenas analisa a frequência de vibração das cordas captada, e controla o motor que aplica o ajuste necessário às cordas. Por este motivo, fica claro que este projeto não necessita de um computador tal como o Raspberry Pi.

Já o Arduino é uma plataforma de prototipação eletrônica livre, e tem base em hardware e software fácil de usar. Esta plataforma é uma placa única com um microcontrolador com suporte de entrada e saída. De acordo com a página principal do projeto, o Arduino serve para a criação de projetos de forma fácil e a baixo custo, tanto para profissionais como para iniciantes.

A placa Arduino é baseada num microcontrolador muito versátil que potencializa suas funções para além de uma simples interface passiva de aquisição de dados, podendo operar sozinha no controle de vários dispositivos e tendo assim aplicações em instrumentação embarcada e robótica. (SOUZA et al., 2011).

O projeto Arduino conta com vários módulos, placas, motores, displays, sensores, LEDs, chaves, transmissores, receptores e outros componentes eletrônicos que podem ser ligados à placa do microcontrolador.

Considerando então, as informações e os motivos, a placa Arduino foi escolhida para a prototipação do dispositivo afinador automatizado; pela sua simplicidade, poder, pelas opções de expansão de periféricos, pela grande comunidade de suporte e principalmente pelo seu custo ser baixo.

2.3.1.1 DADOS TÉCNICOS

O projeto Arduino possui vários modelos de Placas Controladoras. Dentre estes modelos, o mais popular é a Placa Arduino Uno (LIMA-2, 2016). A placa Arduino Uno possui como seu componente principal um microcontrolador Atmega328. Este microcontrolador pode estar presente na placa Arduino como um circuito integrado encaixável ou como um pequeno chip soldado na placa. A vantagem da versão encaixável é que o microcontrolador pode ser facilmente substituído em caso de dano (LIMA-2, 2016).

O Arduino Uno possui uma interface de conexão USB que permite conectá-lo a um computador, o que possibilita fazer comunicação serial e gravar programas na placa (SOUZA, 2013). A placa pode ser alimentada por fontes de energia ou baterias com uma tensão recomendada entre 7 a 20 volts (ARDUINO-3), mas também funciona enquanto está conectada pela interface USB.

A placa Arduino Uno possui 14 pinos digitais e 6 pinos analógicos, que podem ser utilizados como entrada e saída de dados (LIMA-2, 2016). Esses pinos são utilizados para fazer comunicação com os diversos módulos disponíveis. Neste projeto, o a placa Arduino Uno recebe sinais através de um dos pinos analógicos, e envia sinais de controle através dos pinos digitais.

Para a gravação e execução de programas, o Arduino Uno possui em seu microcontrolador uma Memória Flash de 32 KB para gravação de programas, Memória SRAM de 2 KB (memória volátil utilizada para guardar informações em tempo de execução,

como variáveis (ARDUINO-4)), Memória EEPROM de 1 KB (memória não volátil utilizada para guardar informações em longo prazo (ARDUINO-4)), e velocidade de processamento de 16 MHz (LIMA-2, 2016; ARDUINO-3).

2.3.2 COMPONENTES PRINCIPAIS

Além do Sistema de Processamento com a placa Arduino, o dispositivo afinador automatizado é composto por um Sistema Sensor e um Sistema Motor. O Sistema Sensor recebe e trata as informações a serem processadas na placa Arduino, enquanto o Sistema Motor executa as instruções de saída do Sistema de Processamento.

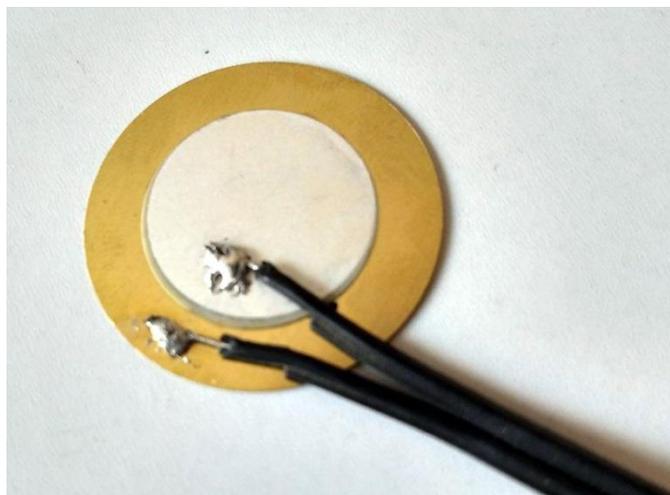
2.3.2.1 TRANSDUTOR PIEZOELÉTRICO

Para a parte de captação o sensor essencial deste projeto é o Sensor de Vibração Piezo. Este sensor tem o nome de Transdutor Piezoelétrico. É um dispositivo que transforma energia de pressão mecânica em energia elétrica (LIMA-1, 2005).

O movimento oscilatório das cordas do violão gera uma energia mecânica no corpo do instrumento, na mesma frequência de oscilação que si próprias (BRAIN, 2014). Essa energia mecânica é captada e convertida em energia elétrica através do transdutor piezoelétrico ligado diretamente ao corpo do violão.

A Figura 3 apresenta o exemplo de um Transdutor Piezoelétrico, sendo a parte interior, de cor branca, o pólo positivo, e a parte exterior, metálica, o pólo negativo.

Figura 3 – O Transdutor Piezoelétrico



Fonte: Os próprios autores (2019)

Após a captação e conversão realizada pelo sensor piezoelétrico, é necessário que o sinal captado seja amplificado e condicionado, de modo a ser realizada a leitura e interpretação correta pela placa Arduino. Mais detalhes sobre este processo na sessão de desenvolvimento.

2.3.2.2 MOTOR DE PASSO

Para o sistema motor do dispositivo afinador automatizado, considerando que é necessário certo nível de força e precisão de posicionamento para ajustar as tarraxas do violão, dois tipos de motor poderiam ter sido escolhidos: O Servo Motor e o Motor de Passo. Ambos os tipos de motor oferecem maior precisão de controle dos movimentos em relação aos motores de correntes contínua e alternada.

O Motor de Passo normalmente tem a mesma força que um Servo Motor de mesmo tamanho, mas proporciona um controle perfeito de posicionamento, enquanto que o Servo Motor pulsa para frente e para trás quando está parado. (TIGERTEK, 2014). Essas pequenas pulsações do Servo Motor poderiam comprometer a acurácia da afinação do violão, por isso, mesmo tendo velocidade menor que o Servo Motor, o Motor de Passo é de maior benefício para este projeto.

Outras vantagens do Motor de Passo são: que ele segue uma lógica digital, que possui alta precisão de posicionamento, e que têm resposta rápida à aceleração e desaceleração, com o perfeito controle de sua velocidade até ao nível máximo. Por esses benefícios, ele é amplamente usado em dispositivos eletrônicos que requerem precisão (BRITES; SANTOS, 2008).

O funcionamento básico do motor de passo é dado pelo uso de solenoides alinhados dois a dois que quando energizados atraem o rotor fazendo-o se alinhar com o eixo determinado pelos solenoides, causando assim uma pequena variação de ângulo que é chamada de passo. A velocidade e o sentido de movimento são determinados pela forma como cada solenoide é ativado (sua ordem e a velocidade entre cada ativação) (BRITES; SANTOS, 2008, p. 5).

O processamento dos dados recebidos e as escolhas feitas pelo programa na placa Arduino, serão totalmente programados neste projeto, assim como toda a estrutura de hardware. Para escrever programas para rodar na placa Arduino, utiliza-se a linguagem de alto nível C++ com algumas pequenas modificações (CHAVIER, 2013). O Arduino possui um

Ambiente de Desenvolvimento Integrado (IDE) de código aberto. Nele é possível escrever e enviar o código para a placa.

2.3.3 PROCESSAMENTO DE ÁUDIO

Após as vibrações serem captadas e transformadas pelo Transdutor Piezoelétrico, é necessário fazer a amplificação e o condicionamento desse sinal de saída.

Para que a leitura pela placa e interpretação dos sinais pelo software implementado seja possível, o sinal elétrico passa por uma parte do circuito que realiza a conversão dos sinais elétricos em números que são conhecidos como bits, esse circuito é chamado “Conversor Analógico/Digital” (A/D) (ZUBEN, p.41 2004).

O conversor A/D faz a análise do sinal de entrada, e escolhe pequenas amostras do sinal captado em pequenos espaços de tempo que são chamados samples (ZUBEN, p.41 2004). A quantidade de amostras obtidas do sinal de entrada é denominada “taxa de amostragem” ou “sample rate”.

Um sinal contínuo no tempo pode ser processado através do processamento de suas amostras em um sistema discreto no tempo. Para isso, a taxa de amostragem deve ser alta o suficiente para permitir a reconstrução desse sinal a partir de suas amostras, sem erro ou com um erro tolerável. Para isso, é necessário ter em mente o Teorema da Amostragem (BARBOSA; BURIAN JR, 2013, p. 4).

De acordo com o Teorema da Amostragem, para que a representação do sinal de entrada seja precisa e esse sinal possa ser reconstruído sem erro a partir de suas amostras, é necessário que a taxa de amostragem seja pelo menos duas vezes maior que o valor de frequência máximo do sinal de entrada (LATHI, 2005 apud BARBOSA; BURIAN JR, 2013, p. 4).

Neste projeto, a conversão de sinal analógico para digital ocorre na própria placa Arduino. Especificações de amostragem estão detalhadas na sessão de desenvolvimento. Os samples amostrados serão interpretados pelo programa no Arduino, que será responsável por enviar comandos de controle para o Sistema Motor.

3 DESENVOLVIMENTO

O acúmulo de referências e informações se estendeu durante toda a pesquisa e desenvolvimento deste trabalho. As áreas de conhecimento incluem eletrônica, circuitos digitais e analógicos, música, propriedades do som e de materiais, programação digital, componentes eletrônicos, física do som e outras áreas afins, que podem ser descobertas durante o processo de pesquisa, por compreenderem a temática do projeto.

A metodologia de pesquisa que foi utilizada no decorrer deste trabalho é a Aplicada: o conhecimento obtido na pesquisa será aplicado de forma prática na realização do projeto, dirigido à solução de problemas específicos (PRODANOV; FREITAS, 2013, p. 51), e que é uma construção baseada em conhecimento e tecnologias já existentes.

Durante o desenvolvimento foi utilizado o método experimental, no qual o conhecimento adquirido e consolidado no decorrer do trabalho de pesquisa e experimentação, foi empregado à produção do resultado deste trabalho (PRODANOV; FREITAS, 2013, p. 37 apud GIL, 2008), contribuindo com a evolução e aperfeiçoamento do mesmo.

Na primeira etapa foram realizadas pesquisas bibliográficas, provendo uma base para dar início ao projeto, e viabilidade para executá-lo de forma prática. Tendo em conta que o produto deste trabalho é uma implementação de hardware/software para a solução de um problema específico.

Na segunda etapa, utilizando-se da base adquirida no decorrer da primeira etapa, foi desenvolvido o protótipo do dispositivo afinador automatizado, e durante todo o desenvolvimento deste protótipo foram relatados os procedimentos e testes realizados.

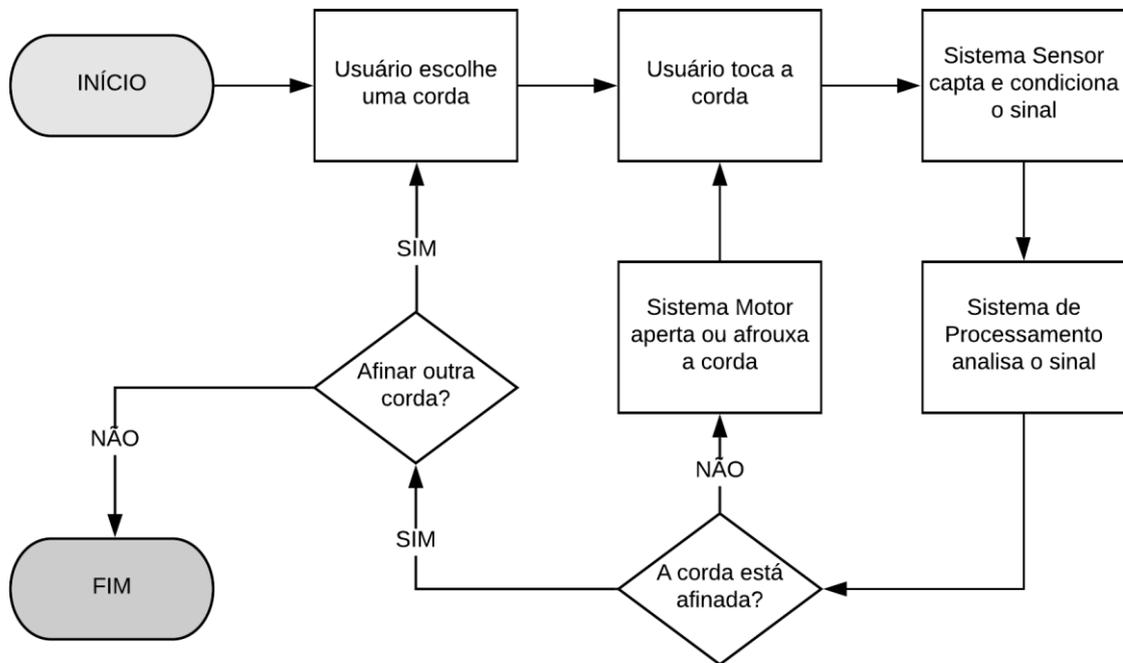
Foram realizados testes para verificar a acurácia e a fidelidade do dispositivo. Esses testes auxiliaram na implementação e na correção de erros do protótipo do dispositivo.

Para medir a acurácia e a fidelidade do dispositivo, os testes se basearam em: realizar a experiência de afinação com o dispositivo desenvolvido; mensurar os resultados utilizando ferramentas já conhecidas como dispositivos afinadores e aplicativos já existentes no mercado, e comparar estes resultados obtidos com os resultados esperados.

A Figura 4 apresenta o processo de afinação proposto, que começa quando o usuário escolhe uma corda para afinar e toca ela no violão. As vibrações da corda são captadas pelo Transdutor Piezoelétrico, que envia sinais elétricos para o circuito de amplificação e condicionamento até chegarem ao Arduino para serem processados. No Arduino, a frequência das vibrações é identificada e seu valor é comparado com o valor ideal, já conhecido para a determinada corda. Conforme necessário, o Arduino envia sinais de controle ao sistema motor

para ajustar a tensão da corda no violão até que esta vibre na frequência ideal. O processo se repete para cada corda do violão.

Figura 4 – Diagrama de Blocos do Dispositivo Eletrônico Automatizado para Afinar Violão



Fonte: Os próprios autores (2019)

Nas seções subsequentes serão expostos os procedimentos que foram realizados durante a prototipação do dispositivo afinador, e os resultados obtidos com as experimentações.

3.1 MÓDULOS DE IMPLEMENTAÇÃO

O projeto do dispositivo afinador foi dividido em três módulos principais, aqui definidos como sistemas: Sistema Sensor, Sistema de Processamento e Sistema Motor. Cada um destes módulos possui um dos três componentes principais do projeto. O Sistema sensor possui o Transdutor Piezoelétrico, o Sistema de Processamento possui o Arduino, e o Sistema Motor possui o Motor de Passo. Os subtópicos seguintes apresentam os detalhes da implementação de cada módulo bem como todos os seus componentes.

3.1.1 SISTEMA SENSOR

O Sistema Sensor além de captar as vibrações das cordas do violão, tem a função de tratar esse sinal captado. Este sistema é composto por um circuito que realiza a amplificação do sinal recebido e o condiciona para poder ser processado.

Figura 5 – Transdutor Piezoelétrico no Corpo do Violão



Fonte: Os próprios autores (2019)

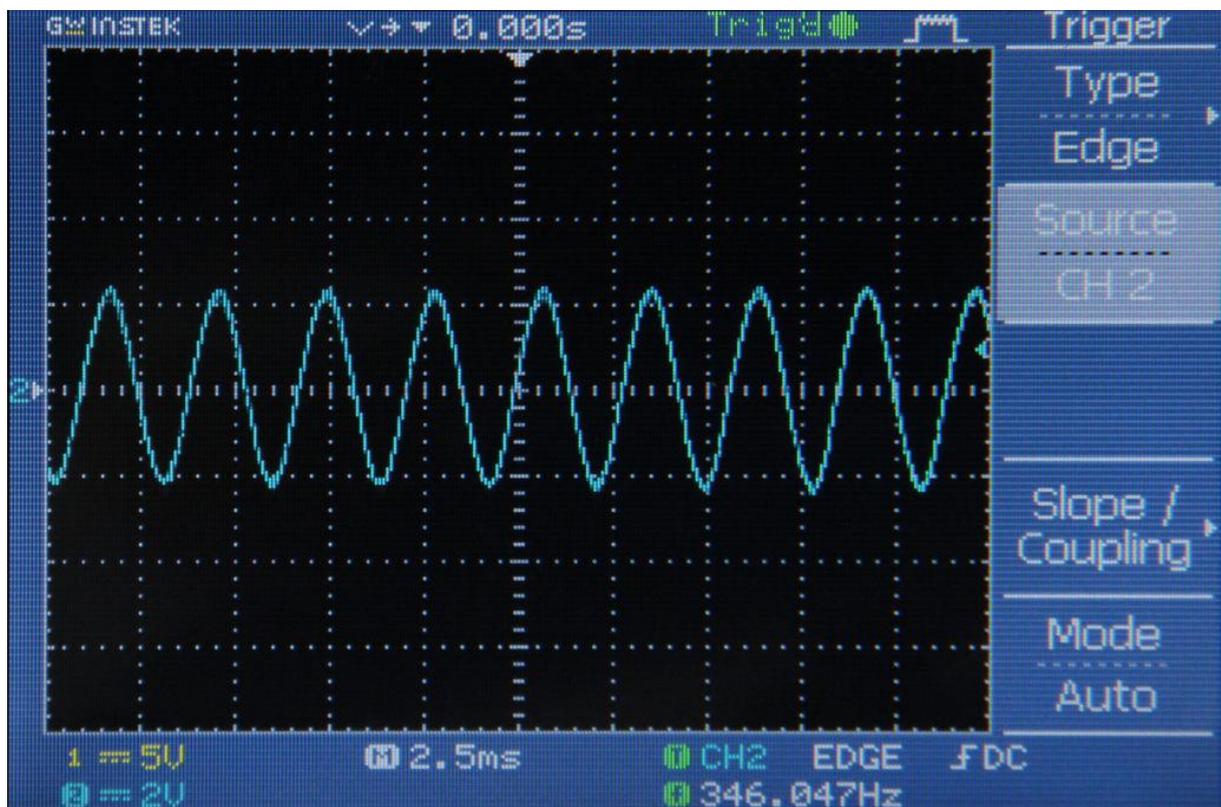
O Transdutor Piezoelétrico em contato com o corpo do violão faz a captação da frequência da corda tocada (Figura 5). O sinal que sai desse transdutor é igual à onda sonora da corda, com um valor de frequência alternada abaixo de um volt. O Arduino Uno trabalha com tensão que varia de zero a cinco volts em seus pinos (ARDUINO-5), fazendo a amostragem do sinal nos pinos analógicos e medindo a variação da tensão, fazendo a atribuição de valores numéricos relativos, correspondes ao valor da tensão. Por esse motivo a amostragem de valores de tensão tão baixos geraria valores muito próximos entre si e imprecisos, além do que o Arduino não pode amostrar corrente alternada, pois esta oscila abaixo de zero, de modo que somente seriam obtidos valores quando a corrente fosse positiva. Isto faz com que seja necessário um circuito de amplificação e polarização do sinal, para que

se tenha maior amplitude e oscilção em tensão maior que zero volts. Só assim o Arduino poderia processar o sinal captado, e com essas informações, foi possível chegar à definição do circuito.

O circuito foi construído baseado no circuito de Entrada de Áudio para Arduino de Ghassaei (GHASSAEI-1, 2012). Este circuito utiliza um amplificador operacional para elevar a tensão do sinal captado, e faz a polarização deste sinal de modo que o mesmo oscile com a mesma forma de onda não em torno de zero volts, mas em torno de dois volts e meio, que pode ser considerado o ponto central na amostragem no Arduino.

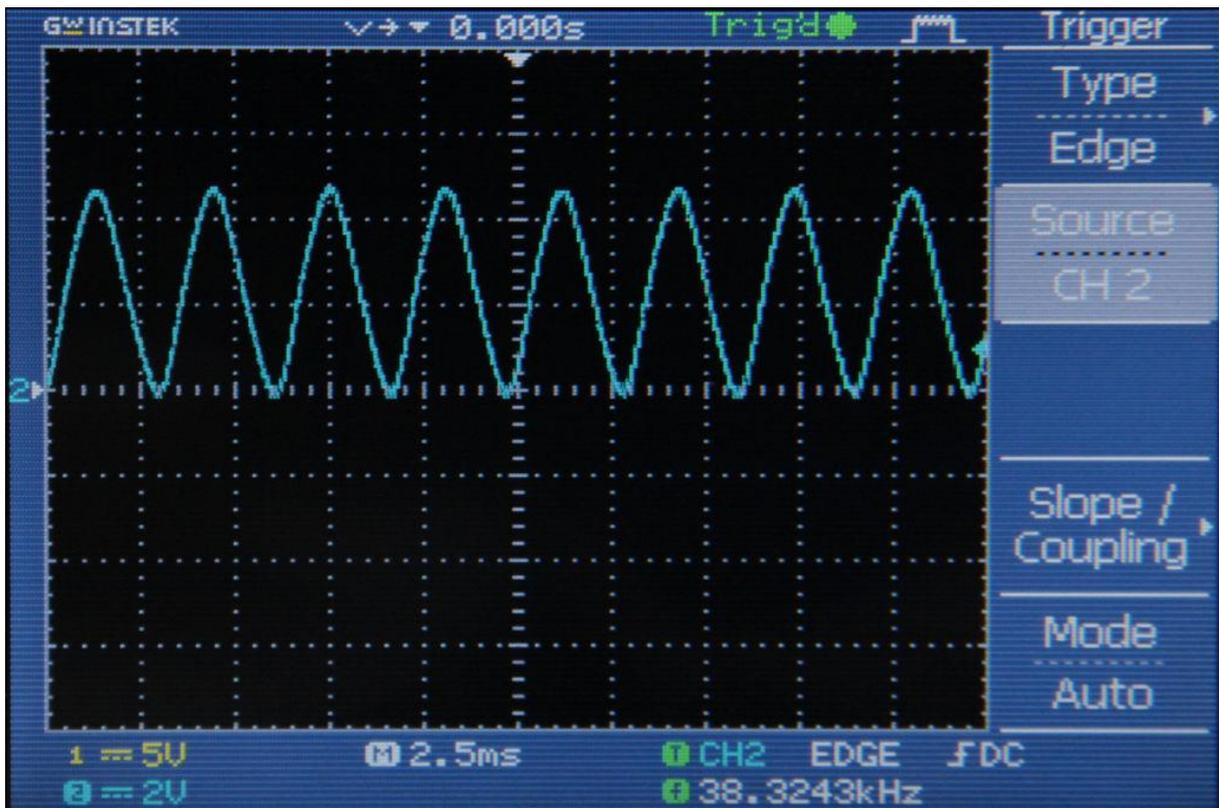
A Figura 6 apresenta um sinal de corrente alternada antes de passar pelo circuito, e a Figura 7, um sinal polarizado após passar pelo circuito. Nessas duas figuras, o limiar de 0 volts é representado pelo pontilhado mais grosso, localizado horizontalmente ao centro do gráfico. É possível perceber que, na Figura 6 a onda oscila para cima e para baixo do limiar de 0 volts, enquanto na figura 7 ela oscila sempre acima de zero volts, ou seja, a corrente sempre é positiva.

Figura 6 – Sinal de Corrente Alternada cruzando a linha de 0 volts



Fonte: GHASSAEI-1 (2012).

Figura 7 – Sinal polarizado após passar pelo circuito

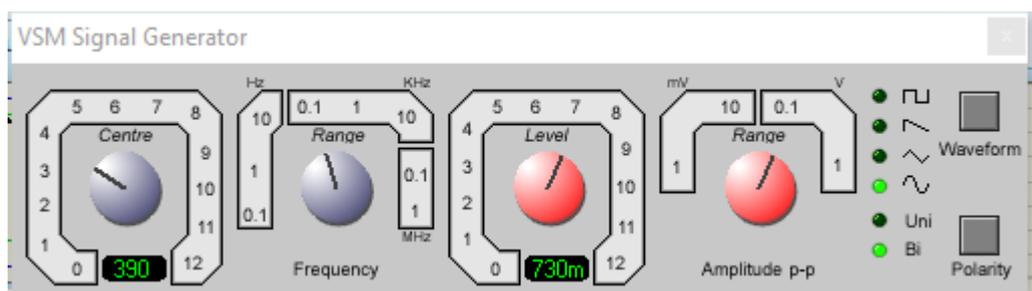


Fonte: GHASSAEI-1 (2012).

Antes de ser construído com componentes reais, o circuito do Sistema Sensor foi construído e simulado virtualmente utilizando a versão de demonstração do software Proteus Design Suite 8.9 da Labcenter Electronics.

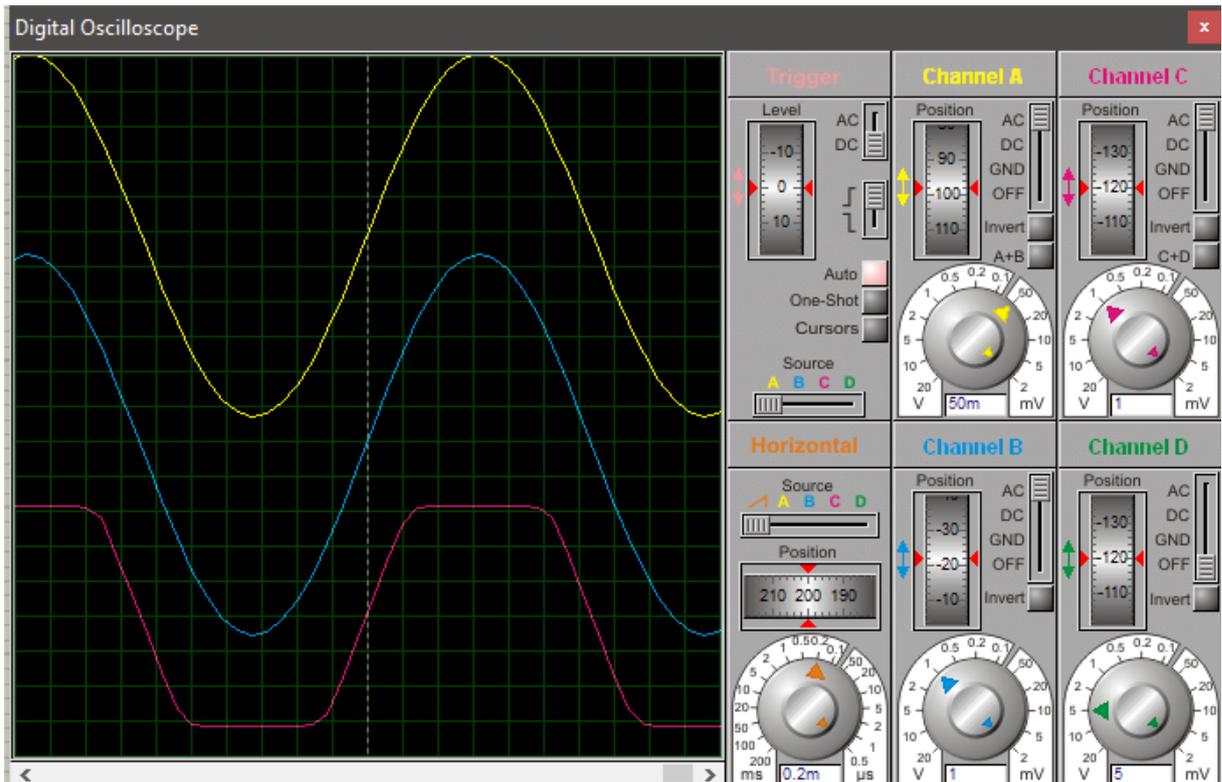
Na entrada do circuito, para simular o som captado do violão, foi utilizado um Gerador de Sinal (elemento 1 na Figura 10), gerando um sinal de corrente alternada com valores de tensão de abaixo de 1 volt. O sinal foi submetido ao circuito, e tanto a saída como a entrada foram analisadas com o uso de um Osciloscópio (elemento 2 na Figura 10). A simulação é apresentada nas Figuras 8, 9 e 10.

Figura 8 – Gerador de Sinal virtual



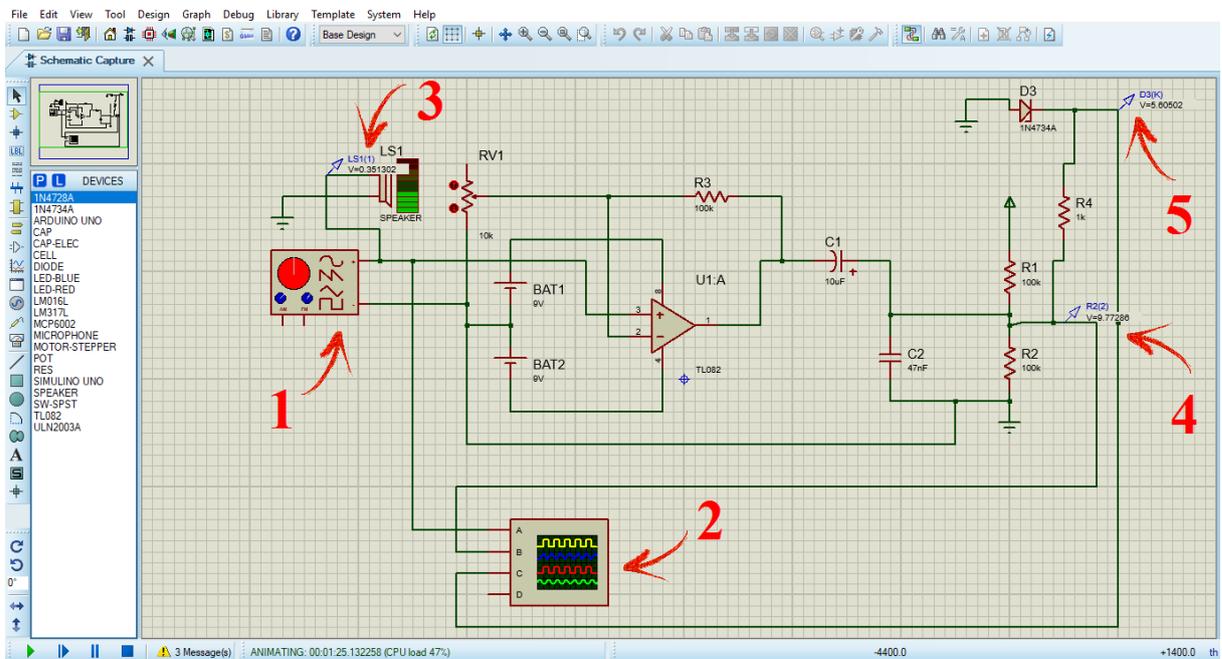
Fonte: Os próprios autores (2019)

Figura 9 – Osciloscópio



Fonte: Os próprios autores (2019)

Figura 10 – Circuito Virtual do Sistema Sensor



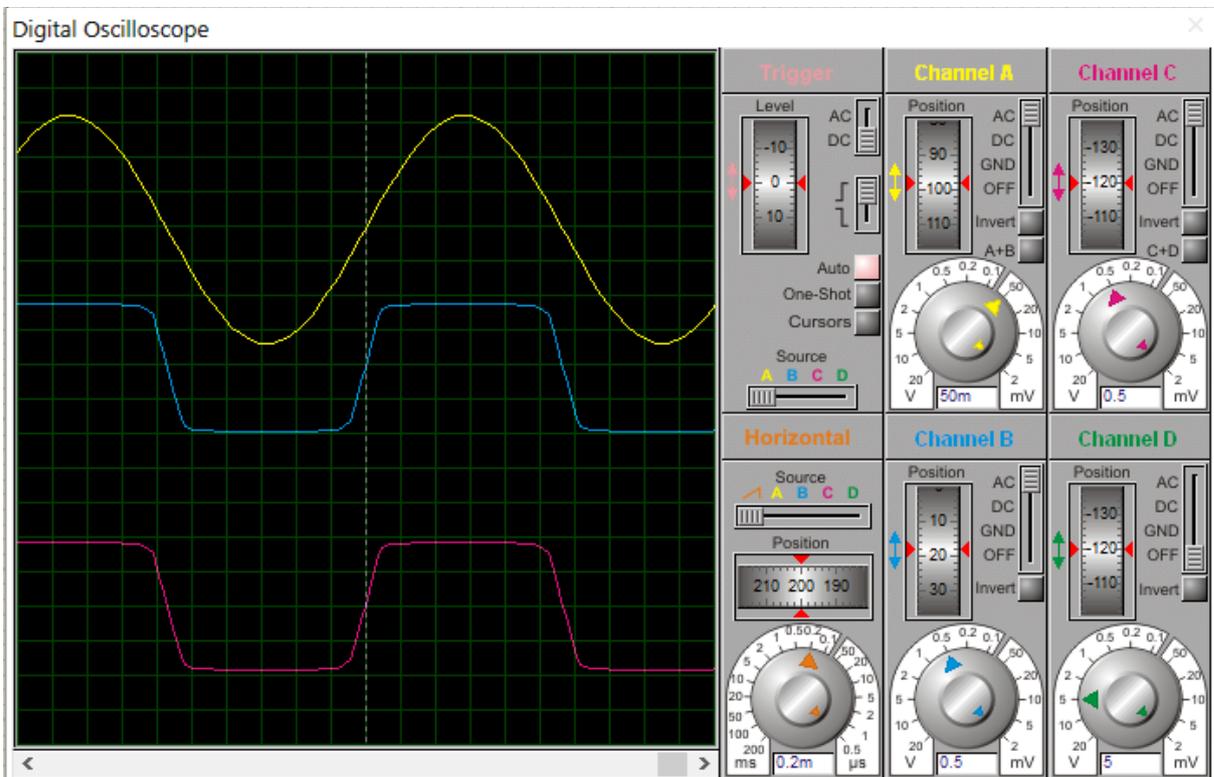
Fonte: Os próprios autores (2019)

O amplificador operacional TL072, utilizado no circuito do sistema Sensor recebe a alimentação recomendada de dezoito volts através de duas baterias (BAT1 e BAT2 na Figura

10), e o sinal de saída pode alcançar tensão de até treze volts e meio (TEXAS INSTRUMENTS, 2017). Como o Arduino Uno suporta tensão de até cinco volts (ARDUINO-3), um pico de tensão acima desse valor poderia danificar a placa.

Para contornar este problema, foi experimentado alimentar o amplificador com apenas cinco volts, mas ao utilizar essa tensão, a saída do amplificador apresentou uma onda distorcida em relação à onda original, conforme mostra a Figura 11, na qual a linha amarela representa a onda do sinal original e a linha azul representa a onda do sinal de saída do amplificador.

Figura 11 – Osciloscópio no Circuito com Amplificador alimentado com 5 volts de tensão



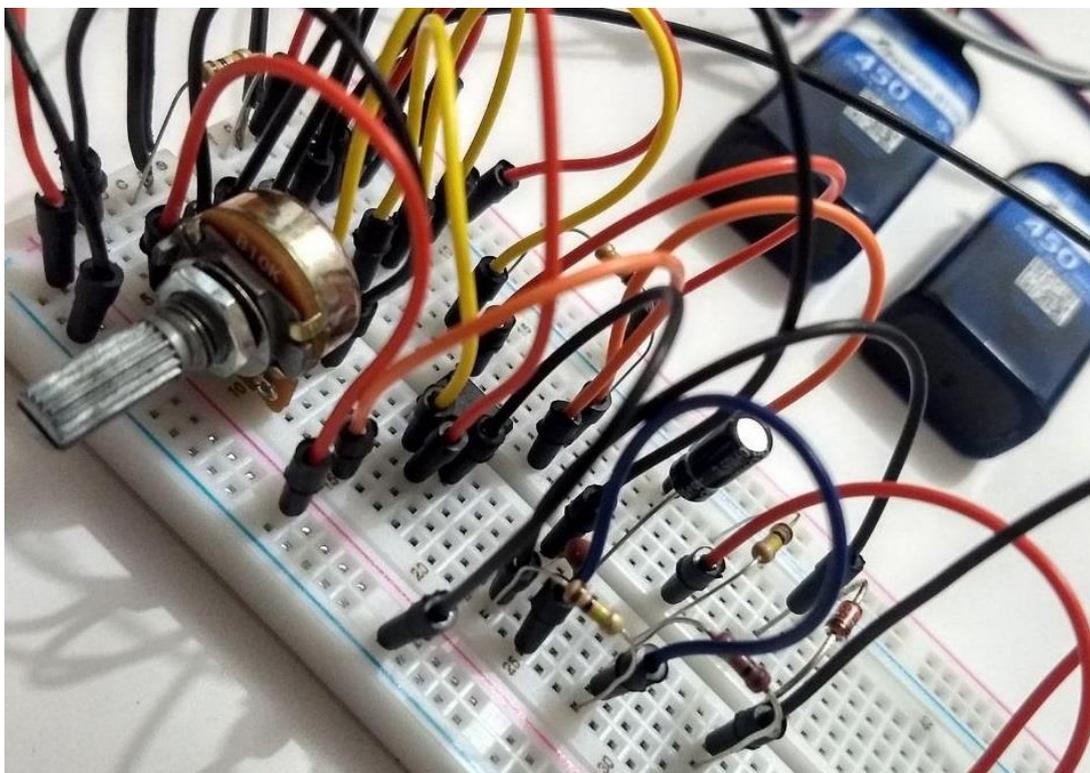
Fonte: Os próprios autores (2019)

A distorção causada pela alimentação insuficiente poderia resultar em um processamento com valores incorretos, por não ser a representação fiel do sinal original. Dessa maneira, foi necessário construir um pequeno circuito de proteção, para que protegesse o Arduino de sobretensão. Este circuito foi construído com base no circuito Regulador de Tensão com Diodo Zener de CLOTHIER (2013). Este circuito no Sistema Sensor possui apenas um Diodo Zener 5,6 volts e um Resistor de 100 KOhms.

Na Figura 10 o elemento 4 apresenta a tensão de saída do circuito, ainda sem proteção, marcando 9,77286 volts em determinado momento. O elemento 5 apresenta a tensão de saída após passar pelo circuito de proteção, marcando 5,60502 volts no mesmo momento. Na Figura 9, pode-se observar a onda de saída do circuito sem proteção representada pela linha azul, enquanto a linha rosa representa a onda de saída do circuito com proteção. É notável que a onda é cortada no limiar de 5,6 volts, tensão máxima permitida pelo Diodo Zener. Nesse exato momento da simulação em que a imagem foi capturada, o pequeno circuito de proteção protegeria a placa Arduino Uno contra sobretensão.

Analisando as Figuras 9 e 11, percebe-se que a onda do sinal de saída do amplificador alimentado com 5 volts é semelhante à onda do sinal de saída do circuito de proteção, concluindo assim que, nos dois casos, a saída seria a mesma. Porém, em sinais com ondas harmônicas (que possuem várias oscilações de frequências diferentes juntas), como é o caso do sinal captado no violão, o resultado mostra-se diferente, pois os picos das ondas harmônicas mais altas que a onda de frequência fundamental iriam distorcer todo o sinal em um circuito, no qual o amplificador é alimentado por 5 volts, ao contrário do que, com o circuito de proteção, esses picos seriam somente evitados.

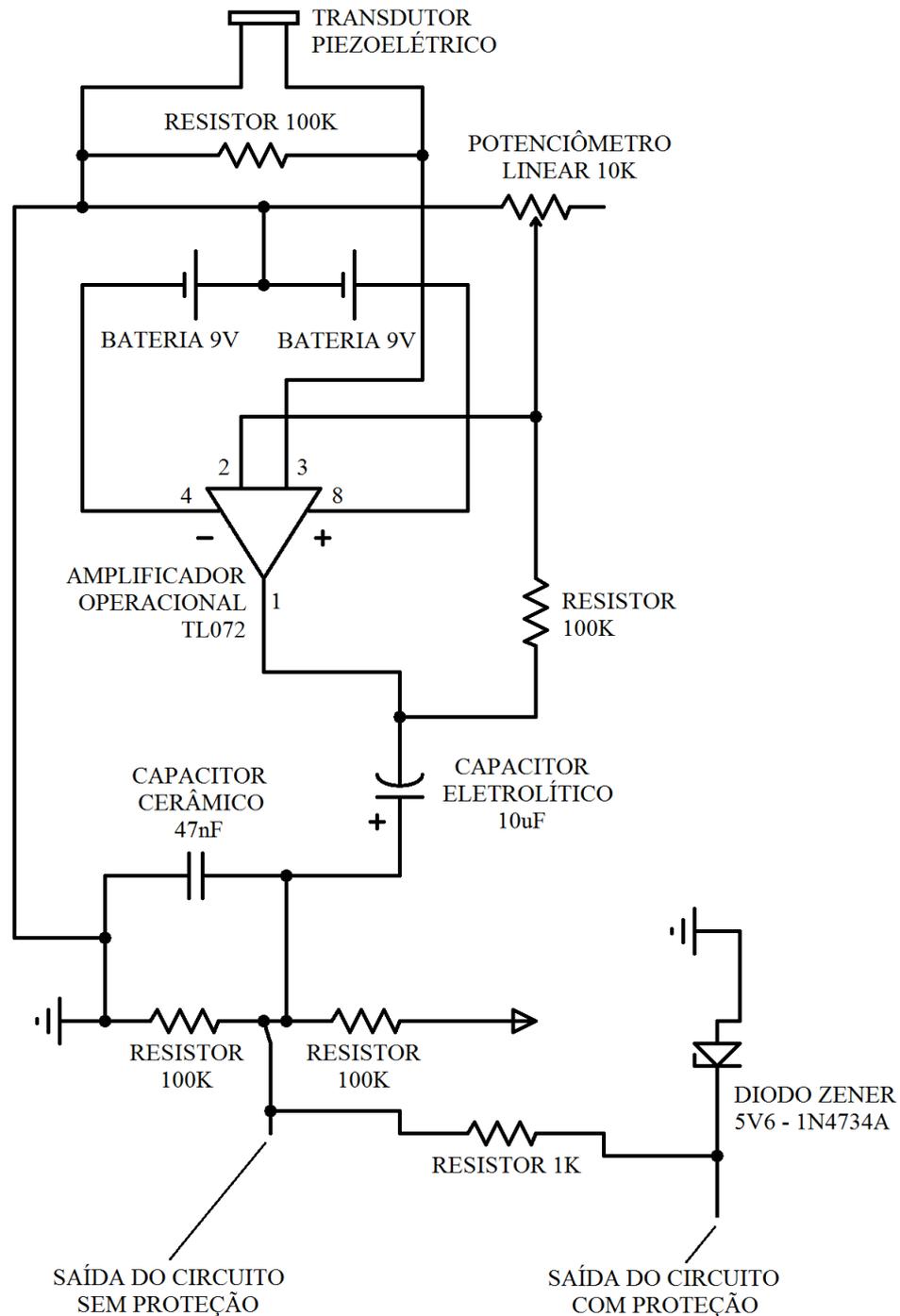
Figura 11 – Circuito montado do Sistema Sensor do Dispositivo Eletrônico Automatizado para Afinar Violão



Fonte: Os próprios autores (2019)

Após ter sido simulado virtualmente, o circuito do Sistema Sensor foi construído com componentes reais (Figura 11), seguindo o esquema elétrico apresentado na Figura 12.

Figura 12 – Esquema Elétrico do Sistema Sensor do Dispositivo Eletrônico Automatizado para Afinar Violão



Fonte: Os próprios autores (2019)

Os componentes ilustrados na Figura 12 e utilizados pra construir o circuito do Sistema Sensor estão listados na Tabela 2. A tabela também apresenta o preço pago por cada

componente para a realização deste projeto. Os componentes foram adquiridos entre Abril e Maio de 2019.

Tabela 2 – Lista de Componentes do Circuito do Sistema Sensor

Quantidade	Descrição do Componente	Preço Unitário	Preço Total
1	65 Jumpers para Protoboard – Macho / Macho	R\$ 10,90	R\$ 10,90
1	Amplificador Operacional TL072	R\$ 1,77	R\$ 1,77
1	Capacitor Cerâmico 47nF / 50V	R\$ 0,07	R\$ 0,07
1	Capacitor Eletrolítico 10uF / 25V	R\$ 0,06	R\$ 0,06
1	Cápsula Piezoelétrica 35MM	R\$ 1,05	R\$ 1,05
2	Clip de Bateria 9V Tipo 1 [Vertical]	R\$ 0,65	R\$ 1,30
1	Diodo Zener 1N4734A [5V6 / 1W]	R\$ 0,14	R\$ 0,14
1	Potenciômetro Linear de 10K	R\$ 1,53	R\$ 1,53
1	Protoboard 830 Pontos MB-102	R\$ 15,90	R\$ 15,90
4	Resistor 100K 5% (1/4W)	R\$ 0,08	R\$ 0,32
1	Resistor 1K 5% (1/4W)	R\$ 0,07	R\$ 0,07
		TOTAL	R\$ 33,11

Fonte: Os próprios autores (2019)

3.1.2 SISTEMA DE PROCESSAMENTO

O Sistema de Processamento tem a função de amostrar e analisar o sinal recebido do Sistema Sensor a fim de identificar qual é a frequência de oscilações presente no sinal. Quando a frequência é identificada, o programa compara o valor dessa frequência com um valor conhecido e esperado. Se houver diferença de valores, fora de uma margem aceitável, o programa então executa as instruções de controle do Sistema Motor, a fim de ajustar a tensão da corda que está sendo afinada. Essas instruções de controle recebem parâmetros que definem se a corda será apertada ou afrouxada pelo Motor de Passo.

O componente principal do Sistema de Processamento é a placa Arduino Uno. Nesse módulo, a implementação do código do programa que é executado no Arduino é a parte mais importante.

O código do programa pode ser dividido em duas partes: O algoritmo de detecção da frequência do sinal analógico, e a lógica de controle que interpreta a saída desse algoritmo e dá as saídas necessárias.

3.1.2.1 ALGORITMO DE DETECÇÃO DE FREQUÊNCIA

Na primeira tentativa de executar um algoritmo de detecção de frequência, tendo uma base de referenciais, o algoritmo que se mostrou mais atraente foi o desenvolvido por Ghassaei (GHASSAEI-2, 2012). Este algoritmo foi embasado no projeto de Entrada de Áudio para Arduino, também desenvolvido por Ghassaei (GHASSAEI-1, 2012), e que serviu de base para a elaboração do Sistema Sensor deste projeto. Esse algoritmo de detecção trabalha com uma taxa de amostragem de 38,5 KHz, o que significa que o Arduino experimenta o valor de entrada analógica 38.500 vezes por segundo. Para que fosse possível amostrar com esta alta taxa, o Arduino foi configurado para que analisasse somente um dos seis pinos analógicos disponíveis. Para que essa taxa fosse garantida, o algoritmo utiliza uma função de interrupção (GHASSAEI-1, 2012). Na plataforma Arduino, quando uma função de interrupção é chamada, as instruções em execução são pausadas e somente as instruções dentro da função de interrupção são executadas, até que termine a função de interrupção (ARDUINO-6). No caso desse algoritmo de detecção, a função de interrupção é chamada toda vez que há uma leitura do pino analógico.

Este algoritmo de detecção de frequência (Anexo A) foi adaptado e executado neste projeto, mas os testes e experimentos não deram um resultado satisfatório. O algoritmo mostrou constância nos resultados com ondas senoidais únicas, mas em um sinal com ondas harmônicas, como o do violão, houve muita divergência. Os resultados da análise de frequência foram monitorados através da comunicação serial com o Arduino. A Figura 13 apresenta uma amostra dos resultados obtidos durante este experimento: ao tentar obter a frequência da corda E4 do violão (valor ideal: 329,628 Hz), obtinha-se valores não constantes, o que não permitia, por exemplo, identificar uma tendência de erro.

Após algumas tentativas de obter melhores resultados, com a configuração de parâmetros e o tratamento da saída do algoritmo para identificar algum padrão, o mesmo foi deixado de lado por um momento. Na segunda etapa na busca de sucesso, foi utilizado o algoritmo de detecção desenvolvido por Kelly (2017). Este algoritmo (Anexo B), com poucas linhas de código apresentou melhores resultados.

O algoritmo proposto por Kelly trabalha com a taxa de amostragem padrão que pode chegar até 9615 Hz, mas durante os testes essa taxa não foi alcançada. Conforme aumenta o número de instruções e conversões que o Arduino tem que processar, essa taxa de amostragem diminui. Ao executar o algoritmo de Kelly sem fazer modificações, os resultados das frequências identificadas mantiveram uma constância considerável para cada corda, mas em todas as cordas mostrou um valor diferente do real.

Figura 13 – Experimento: identificar frequência da corda E4 do violão

```
00:12:05.725 -> 69.43 hz
00:12:05.759 -> 69.43 hz
00:12:05.759 -> 346.50 hz
00:12:05.793 -> 346.50 hz
00:12:05.793 -> 346.50 hz
00:12:05.827 -> 346.50 hz
00:12:05.827 -> 346.50 hz
00:12:05.860 -> 346.50 hz
00:12:05.860 -> 343.41 hz
00:12:05.860 -> 349.65 hz
00:12:05.894 -> 346.50 hz
00:12:05.894 -> 349.65 hz
00:12:05.928 -> 346.50 hz
00:12:05.928 -> 337.39 hz
00:12:05.962 -> 352.86 hz
00:12:05.962 -> 340.37 hz
00:12:05.996 -> 349.65 hz
00:12:05.996 -> 343.41 hz
```

Auto-rolagem Show timestamp Nova-linha ▾

Fonte: Os próprios autores (2019)

O próprio Kelly (2017) identificou e relatou essa deficiência no processamento do algoritmo. O valor da frequência de amostragem é utilizado para fazer o cálculo da frequência do sinal recebido. O algoritmo de Kelly guarda esse valor como um parâmetro constante, e em seu código original, esse valor vem configurado como 8919, que é o valor real da taxa de amostragem medido por Kelly.

Por mostrar uma constância aceitável na identificação dos valores de frequência do sinal recebido, este algoritmo foi adaptado para este projeto, porém ao serem feitas experimentações, notou-se que o valor da taxa de amostragem mudou, resultando em detecção de valores de frequências ainda mais distantes dos valores reais. Então ao invés de utilizar métodos de depuração para identificar o valor da nova taxa de amostragem, foi utilizada a regra de três simples. Conhecendo o valor parametrizado da taxa de amostragem e o valor errado que o algoritmo retorna para uma frequência conhecida, poderia ser calculado o valor da taxa de amostragem para que o retorno do algoritmo fosse correto. A Tabela 3 apresenta a lógica matemática que foi utilizada para calcular o valor da taxa de amostragem que faz com que seja mostrado a frequência correta no resultado do algoritmo:

Tabela 3 – Lógica da Regra de Três Simples para Cálculo da Frequência de Amostragem Correta

TAXA DE AMOSTRAGEM	FREQUÊNCIA
ATUAL	VALOR DA MEDIÇÃO ERRADA
X	VALOR CORRETO CONHECIDO

Fonte: Os próprios autores (2019)

Deste modo, foi realizado o cálculo para obter o valor acertado da taxa de amostragem para a corda E4 (329,628 Hz), exemplificado na tabela 4.

Tabela 4 – Cálculo da Frequência de Amostragem para a nota E4

TAXA DE AMOSTRAGEM	FREQUÊNCIA
8919	343,04 Hz
X = 8570,34156	329,63 Hz

Fonte: Os próprios autores (2019)

Utilizando o valor calculado da taxa de amostragem no algoritmo adaptado, o resultado para a corda E4 foi ideal: O algoritmo mediu a frequência da corda corretamente. Porém, utilizando esse mesmo valor de amostragem para identificar a frequência das outras cordas, o algoritmo retornou respostas não satisfatórias. Havendo sucesso para a identificação da frequência de uma corda com um parâmetro específico, foi feito então um ajuste no algoritmo: o valor constante da taxa de amostragem passou a ser dinâmico, de maneira que para cada corda o algoritmo recebesse um valor específico como parâmetro. Seguindo a mesma regra utilizada para calcular o parâmetro para a corda E4, foram calculados os parâmetros específicos para cada corda do violão. A Tabela 5 apresenta os valores obtidos.

Tabela 5 – Valores de Taxa de Amostragem calculados para cada corda do Violão

CORDA	TAXA DE AMOSTRAGEM	FREQUÊNCIA
E2	8406	82,41 Hz
A2	8509	110,00 Hz
D3	8516	146,83 Hz
G3	8624	196,00 Hz
B3	8643	246,94 Hz
E4	8570	329,63 Hz

Fonte: Os próprios autores (2019)

Utilizando esses valores calculados como parâmetros, foram realizados experimentos para tentar afinar cada corda do violão, e nesses experimentos foi obtido resultado satisfatório. Esse algoritmo adaptado encontra-se no Apêndice A.

O modelo de utilização proposto por este projeto consiste em escolher uma corda através de um botão e tocá-la para que o dispositivo realize a afinação automaticamente. Cada vez que uma corda é escolhida, os parâmetros são reconfigurados de acordo com a corda.

Outros parâmetros importantes são os valores de corte de frequência estabelecidos para cada corda, esses valores foram definidos para funcionarem como uma forma de filtro de modo que se algum valor de frequência fora do intervalo definido for detectado, o valor será ignorado pelo programa, para a corda A2 temos como frequência de corte mínima 95,15 e como máxima 127,14 (Apêndice A). A Figura 13 apresenta a comunicação serial com Arduino rodando essa implementação do programa, enquanto se afina a corda E4.

Figura 13 – Afinação da corda E4

```

COM63
18:09:02.309 -> 252.06 - FORA
18:09:02.751 -> 1714.00 - FORA
18:09:03.193 -> 65.42 - FORA
18:09:03.635 -> inf - FORA
18:09:04.077 -> 56.38 - FORA
18:09:06.701 -> 317.41 - 🎵 | 329.63 (329.13 ~ 330.13) Apertando +400
18:09:13.243 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:13.685 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:18.515 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:18.960 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:21.611 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:22.053 -> inf - FORA
18:09:24.265 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:24.708 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:25.150 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:27.363 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:27.804 -> 329.62 - 🎵 | 329.63 (329.13 ~ 330.13) AFINADO!!!
18:09:28.246 -> 67.48 - FORA

```

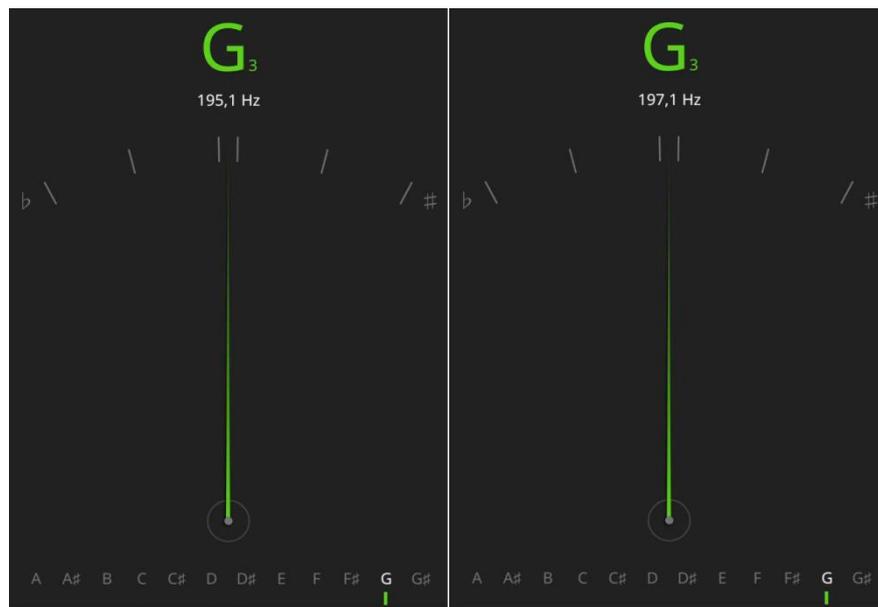
Fonte: Os próprios autores (2019)

3.1.2.2 TESTES

Ao analisar outros dispositivos afinadores, foi notado que alguns possuem uma margem de erro de até um hertz, classificando cordas com frequências dentro deste intervalo de erro como afinadas. A Figura 14 apresenta duas telas de um afinador ao afinar a corda G3. Na tela à esquerda o afinador mostra o valor de frequência 195,1 hertz, que está 0,898 hertz abaixo do valor ideal para a corda (195,998 Hertz), e na tela à direita o afinador mostra o

valor 197,1 hertz, que está 1,102 hertz acima do valor ideal. Mesmo com esse desvio, nos dois casos o afinador aponta que a corda está afinada mostrando o ponteiro e o nome da nota na cor verde. Para referência, um exemplo de quando a corda está desafinada é apresentado na Figura 15, numa tela em que o afinador mostra o ponteiro e o nome da nota na cor laranja. Visando um resultado de afinação mais próximo do ideal, a margem de erro adotada para este projeto foi de 0,5 Hz para todas as cordas.

Figura 14 – Telas do Afinador para a corda G3



Fonte: Os próprios autores (2019)

Figura 15 – Tela do Afinador para a corda G3 desafinada

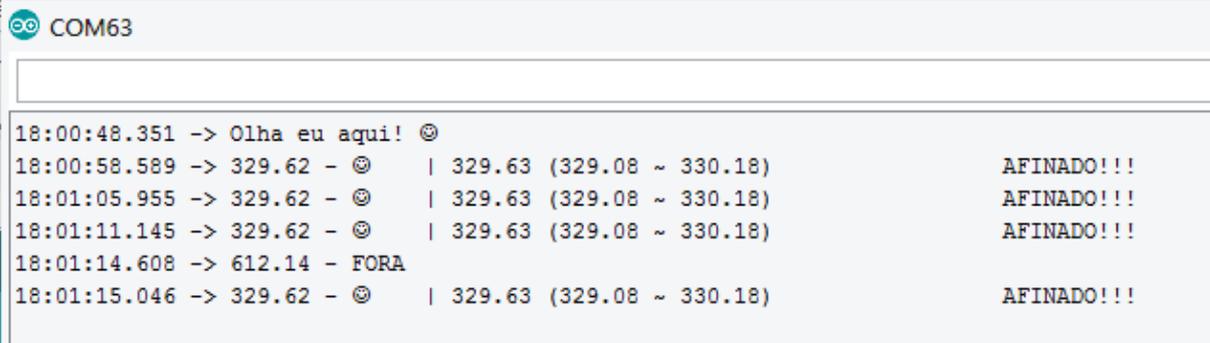


Fonte: Os próprios autores (2019)

Como critério para avaliar se havia sucesso nos experimentos do tópico anterior, foram utilizados dois afinadores já conhecidos no mercado, do tipo que apontam se a corda está afinada. O afinador mais simples, utilizado no teste do parágrafo anterior, é a aplicação para dispositivos Android “Afinador”, na versão 3.4.16, fabricado pela Studio Sol do Brasil. O outro afinador é parte de uma pedaleira profissional para guitarra, modelo “G2Nu” fabricado pela ZOOM Corporation do Japão.

A Figura 16 apresenta a comunicação serial com o Arduino ao tocar a corda E4, quando esta se encontra afinada. Na imagem é identificado o valor de frequência de 329,62 hertz e a informação de que a corda está afinada. A Figura 17 apresenta a aplicação “Afinador” apontando que corda está afinada e mostrando o valor de frequência de 329,6 hertz, próximo do valor mostrado pelo protótipo. Como validação final, a Figura 18 apresenta o afinador profissional apontando a corda como afinada, dado indicado pela luz verde e pelo nome da nota na tela, porém este último afinador não mostra o valor da frequência. Este teste foi realizado para comprovar que os valores calculados no tópico anterior para a taxa de amostragem levaram a um resultado satisfatório.

Figura 16 – Comunicação serial do algoritmo do projeto para a corda E4 afinada



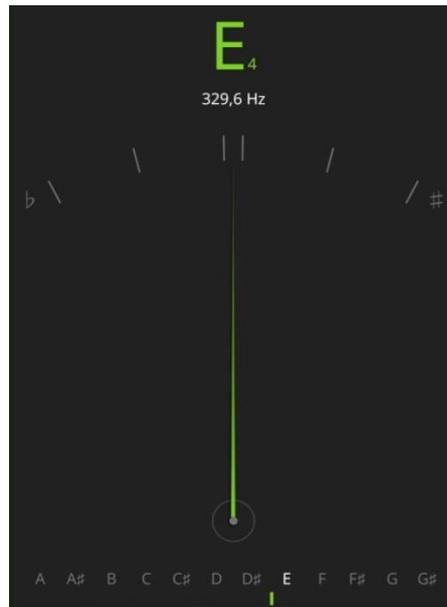
```

COM63
18:00:48.351 -> Olha eu aqui! ☺
18:00:58.589 -> 329.62 - ☺ | 329.63 (329.08 ~ 330.18) AFINADO!!!
18:01:05.955 -> 329.62 - ☺ | 329.63 (329.08 ~ 330.18) AFINADO!!!
18:01:11.145 -> 329.62 - ☺ | 329.63 (329.08 ~ 330.18) AFINADO!!!
18:01:14.608 -> 612.14 - FORA
18:01:15.046 -> 329.62 - ☺ | 329.63 (329.08 ~ 330.18) AFINADO!!!

```

Fonte: Os próprios autores (2019)

Figura 17 – Tela do Afinador para a corda E4 afinada



Fonte: Os próprios autores (2019)

Figura 18 – Foto do afinador profissional para a corda E4 afinada



Fonte: Os próprios autores (2019)

Mesmo com um bom resultado, foram detectadas variações na detecção da frequência do sinal, conforme destacado na Figura 19. Essas variações faziam o afinador detectar um valor de frequência diferente do real em alguns momentos, o que compromete o resultado da afinação. Foi realizado então um teste para verificar a taxa de amostragem real do Arduino enquanto executando o algoritmo do projeto. Este teste foi realizado com um algoritmo que foi desenvolvido com base em alguns trechos de código idealizados por Maes (2018). Com esse algoritmo (Apêndice B) foi possível verificar que o Arduino não mantinha uma taxa de amostragem constante como se esperava (Figura 20), o que resultava nas tais variações na detecção da frequência do sinal, com isso, concluiu-se que os valores de taxa de amostragem calculados anteriormente eram apenas aproximados dos valores reais.

Dado este resultado, foi feita uma tentativa de fazer com que a taxa de amostragem do Arduino fosse constante, utilizando funções de interrupção para a leitura do sinal, ou seja, o Arduino pararia de executar qualquer instrução para fazer a leitura do sinal no pino analógico. Para isso, outro algoritmo foi desenvolvido, também com base nos códigos de Maes (2018), mas ao realizar o teste com este algoritmo (Apêndice C), verificou-se que a taxa de amostragem continuava a variar, conforme apresenta a Figura 21.

Outra tentativa de resolver este problema foi realizada. Desta vez ao invés de tentar fixar a taxa de amostragem, foi desenvolvido um algoritmo que calcula a taxa de amostragem antes de fazer a detecção da frequência. Ao testar esse algoritmo (Apêndice D), percebeu-se que o valor calculado nunca era correto e a frequência detectada pelo protótipo apresentava maior variação que nos testes anteriores, conforme apresenta a Figura 22.

Figura 19 – Variações na detecção pelo protótipo da corda E4

```

COM63
|
22:27:09.635 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:13.088 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:18.275 -> 113.45 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:21.701 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:26.915 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:31.654 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:35.967 -> inf - FORA
22:27:37.735 -> 116.56 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:41.155 -> 114.99 - © | 110.00 (109.45 ~ 110.55) Afrouxando +400
22:27:44.599 -> 531.81 - FORA
22:27:45.039 -> inf - FORA

```

Fonte: Os próprios autores (2019)

Figura 20 – Cálculo da Taxa de Amostragem em tempo real

```

12:26:33.722 -> Taxa de amostragem: 8925.07 Hz
12:26:33.722 ->
12:26:33.848 -> Tempo médio por amostra: 112.05 µs
12:26:33.848 -> Taxa de amostragem: 8924.75 Hz
12:26:33.848 ->
12:26:33.945 -> Tempo médio por amostra: 112.04 µs
12:26:33.945 -> Taxa de amostragem: 8925.38 Hz
12:26:33.945 ->
12:26:34.074 -> Tempo médio por amostra: 112.04 µs
12:26:34.074 -> Taxa de amostragem: 8925.38 Hz
12:26:34.074 ->
12:26:34.175 -> Tempo médio por amostra: 112.04 µs
12:26:34.175 -> Taxa de amostragem: 8925.38 Hz
12:26:34.175 ->
12:26:34.311 -> Tempo médio por amostra: 112.04 µs
12:26:34.311 -> Taxa de amostragem: 8925.38 Hz
12:26:34.311 ->
12:26:34.421 -> Tempo médio por amostra: 112.05 µs
12:26:34.421 -> Taxa de amostragem: 8924.75 Hz
12:26:34.421 ->
12:26:34.522 -> Tempo médio por amostra: 112.04 µs
12:26:34.522 -> Taxa de amostragem: 8925.07 Hz
12:26:34.522 ->
12:26:34.623 -> Tempo médio por amostra: 112.05 µs
12:26:34.623 -> Taxa de amostragem: 8924.43 Hz
12:26:34.657 ->
12:26:34.747 -> Tempo médio por amostra: 112.04 µs
12:26:34.747 -> Taxa de amostragem: 8925.07 Hz

```

Fonte: Os próprios autores (2019)

Figura 21 – Cálculo da Taxa de Amostragem em tempo real com funções de interrupção

```

12:23:58.978 -> Taxa de amostragem: 38467.46 Hz
12:23:58.978 ->
12:23:59.012 -> Tempo médio por amostra: 26.00 µs
12:23:59.012 -> Taxa de amostragem: 38461.54 Hz
12:23:59.012 ->
12:23:59.046 -> Tempo médio por amostra: 26.01 µs
12:23:59.046 -> Taxa de amostragem: 38449.71 Hz
12:23:59.046 ->
12:23:59.046 -> Tempo médio por amostra: 26.00 µs
12:23:59.046 -> Taxa de amostragem: 38455.62 Hz
12:23:59.046 ->
12:23:59.081 -> Tempo médio por amostra: 25.98 µs
12:23:59.081 -> Taxa de amostragem: 38485.22 Hz
12:23:59.081 ->
12:23:59.122 -> Tempo médio por amostra: 26.00 µs
12:23:59.122 -> Taxa de amostragem: 38461.54 Hz
12:23:59.122 ->
12:23:59.153 -> Tempo médio por amostra: 25.98 µs
12:23:59.153 -> Taxa de amostragem: 38485.22 Hz
12:23:59.153 ->
12:23:59.153 -> Tempo médio por amostra: 26.00 µs
12:23:59.153 -> Taxa de amostragem: 38467.46 Hz
12:23:59.196 ->
12:23:59.196 -> Tempo médio por amostra: 26.00 µs
12:23:59.196 -> Taxa de amostragem: 38461.54 Hz
12:23:59.196 ->
12:23:59.237 -> Tempo médio por amostra: 26.01 µs
12:23:59.237 -> Taxa de amostragem: 38449.71 Hz

```

Fonte: Os próprios autores (2019)

Figura 22 – Algoritmo do projeto com Taxa de Amostragem calculada antes de cada detecção de frequência

```

-> Tempo médio por amostra: 120.28 µs
-> Amostras por segundo: 8313.71
->
-> Tempo médio por amostra: 120.63 µs
-> Amostras por segundo: 8289.48
->
->
->
-> 243.81 Hz - FORA
->
-> Tempo médio por amostra: 120.51 µs
-> Amostras por segundo: 8297.81
->
->
->
-> 244.05 Hz - FORA
->
-> Tempo médio por amostra: 120.34 µs
-> Amostras por segundo: 8309.66
->
->
->
-> 251.81 Hz - FORA
->
-> Tempo médio por amostra: 120.58 µs
-> Amostras por segundo: 8293.38
->
->
->
-> 243.92 Hz - FORA
->
-> Tempo médio por amostra: 120.31 µs
-> Amostras por segundo: 8312.09

```

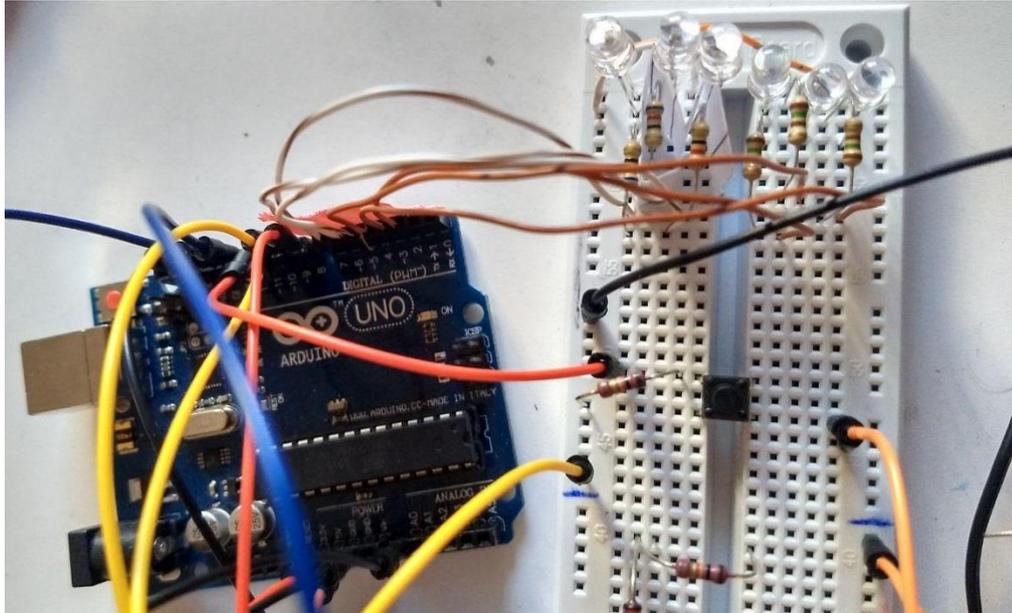
Fonte: Os próprios autores (2019)

Com esses resultados, ficou comprovado que o Arduino não é um microcontrolador ideal para esse tipo de projeto, por causa da variação de sua taxa de amostragem dos pinos analógicos. Com isso, conclui-se que utilizando Arduino, não é possível garantir a acuracidade do dispositivo.

3.1.2.3 CIRCUITO

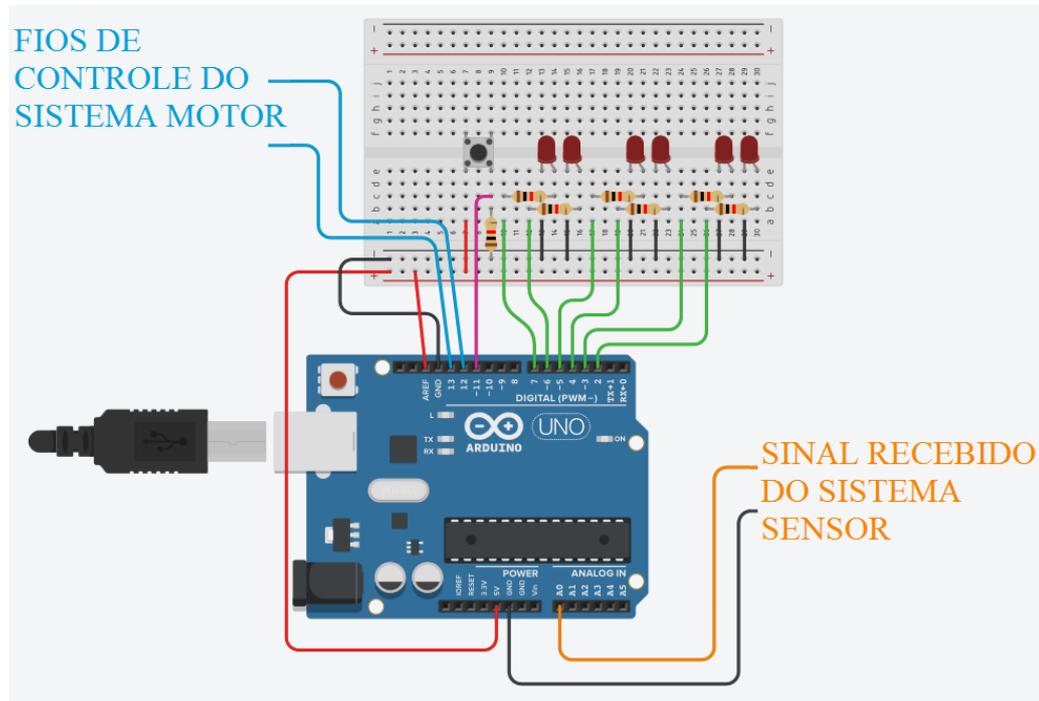
A Figura 23 apresenta o circuito montado do Sistema de Processamento. A Figura 24 apresenta o Circuito Virtual do Sistema de Processamento.

Figura 23 – Circuito do Sistema de Processamento do Dispositivo Eletrônico Automatizado para Afinar Violão



Fonte: Os próprios autores (2019)

Figura 24 – Circuito Virtual de Processamento do Dispositivo Eletrônico Automatizado para Afinar Violão



Fonte: Os próprios autores (2019)

Os componentes ilustrados na Figura 24 e utilizados pra construir o circuito do Sistema de Processamento estão listados na Tabela 6. A tabela também apresenta o preço pago por cada componente para a realização deste projeto. Os componentes foram adquiridos entre Abril e Maio de 2019.

Tabela 6 – Lista de Componentes do Circuito do Sistema de Processamento

Quantidade	Descrição do Componente	Preço Unitário	Preço Total
1	Arduino Uno R3 - Compatível + Cabo USB 2.0	R\$ 54,90	R\$ 54,90
1	Chave Táctil 6x6x7mm 4 Terminais	R\$ 0,12	R\$ 0,12
6	LED de Alto Brilho 5mm Verde	R\$ 0,25	R\$ 1,50
1	Resistor 100K 5% (1/4W)	R\$ 0,08	R\$ 0,08
6	Resistor 62R 5% (1/4W)	R\$ 0,08	R\$ 0,48
		TOTAL	R\$ 57,08

Fonte: Os próprios autores (2019)

3.1.3 SISTEMA MOTOR

O Sistema Motor tem a simples função de obedecer às instruções recebidas pelo Sistema de Processamento. O componente principal deste módulo é o Motor de Passo. Outro componente, necessário para que o Motor de Passo funcione é o Circuito Controlador do Motor de Passo.

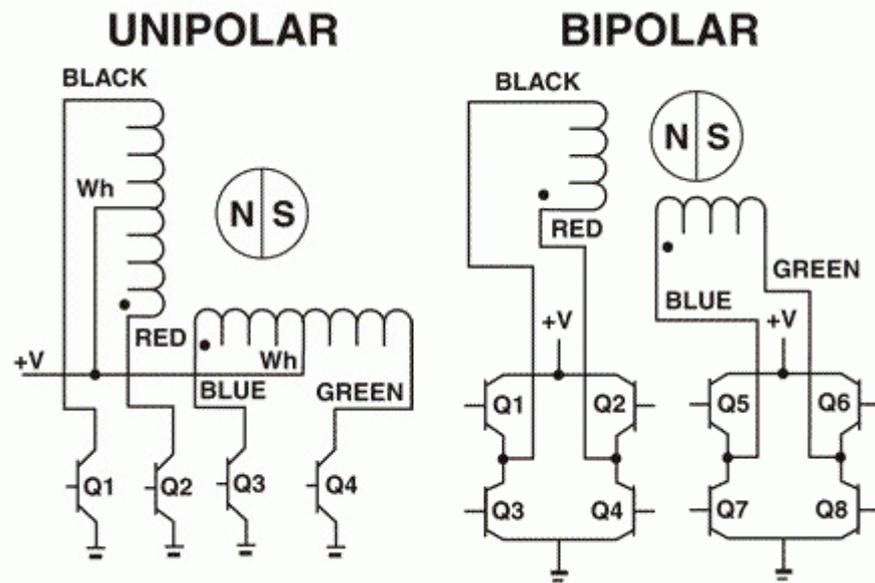
3.1.3.1 MOTOR DE PASSO: UNIPOLAR X BIPOLAR

Existem dois tipos de Motor de Passo: Motor Unipolar e Motor Bipolar. Para cada tipo de motor há um circuito controlador específico.

Os motores de passo do tipo unipolar possuem apenas uma fase. Cada bobina possui uma derivação no meio, o que permite acionar apenas uma metade da bobina para direcionar o motor, sem ser necessário inverter o sentido da corrente. Esse tipo de motor é fácil de ser controlado, mas possui menos torque que o motor bipolar (GEORGE, 2012).

Os motores do tipo bipolar possuem duas fases e requerem um circuito mais complexo. Cada bobina é ativada por inteiro, e a direção do motor depende do sentido da corrente na bobina. Esse tipo de motor possui mais torque que o motor unipolar (GEORGE, 2012). A Figura 25 exemplifica o formato de cada tipo de motor de passo e a abstração de seus circuitos de controle.

Figura 25 – Tipos de Motor de Passo

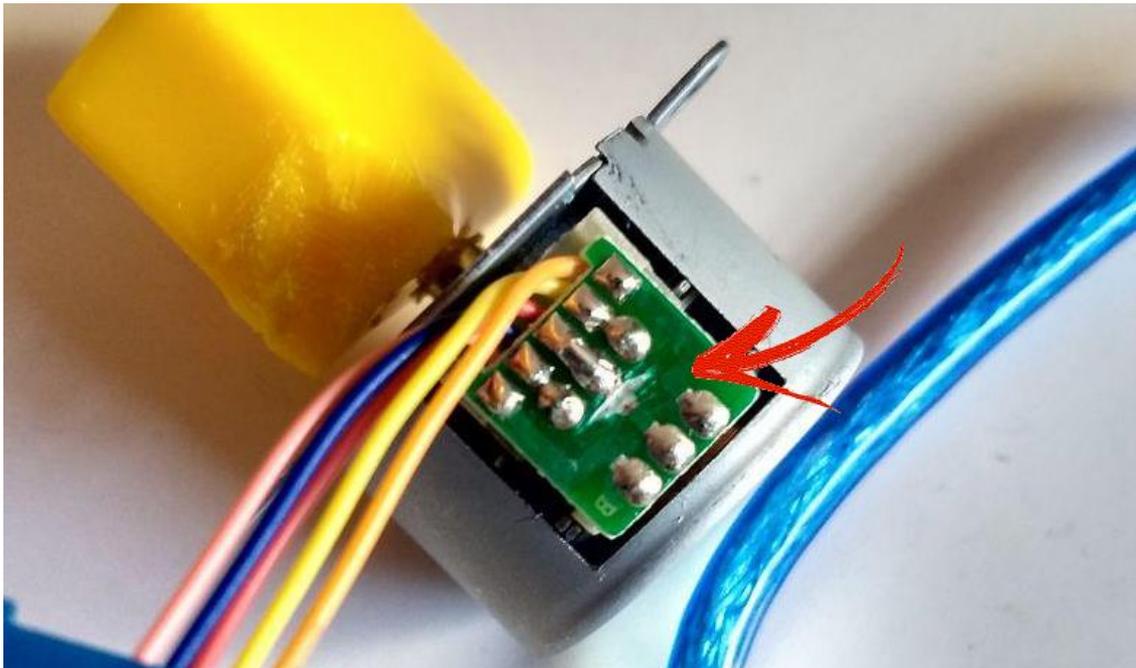


Fonte: George (2012)

Na primeira etapa de testes com o Sistema Motor, foi utilizado o Motor de Passo Unipolar 28BYJ-48 na versão de 5 volts com o circuito controlador ULN2003. Segundo as especificações (KIATRONICS), este motor tem torque de tração maior que 34,3 mN.m, porém, durante os testes de afinação, o Motor não teve força suficiente para apertar as tarraxas de um violão. Para contornar este problema, ao invés de partir diretamente para um novo motor, foi feita uma alteração no Motor Unipolar para que este operasse como Bipolar. Essa alteração foi proposta Jangeox (2013), que relatou ter obtido em testes com o motor bipolar, três vezes mais força do que quando unipolar. É uma alteração simples. Consiste cortar a ligação entre os fios do meio das bobinas para que essas possam ser controladas por inteiro separadamente, operando como um verdadeiro motor bipolar (JANGEOX, 2013). A Figura 26 apresenta o corte que é necessário ser feito na placa do motor para desligar os centros das bobinas.

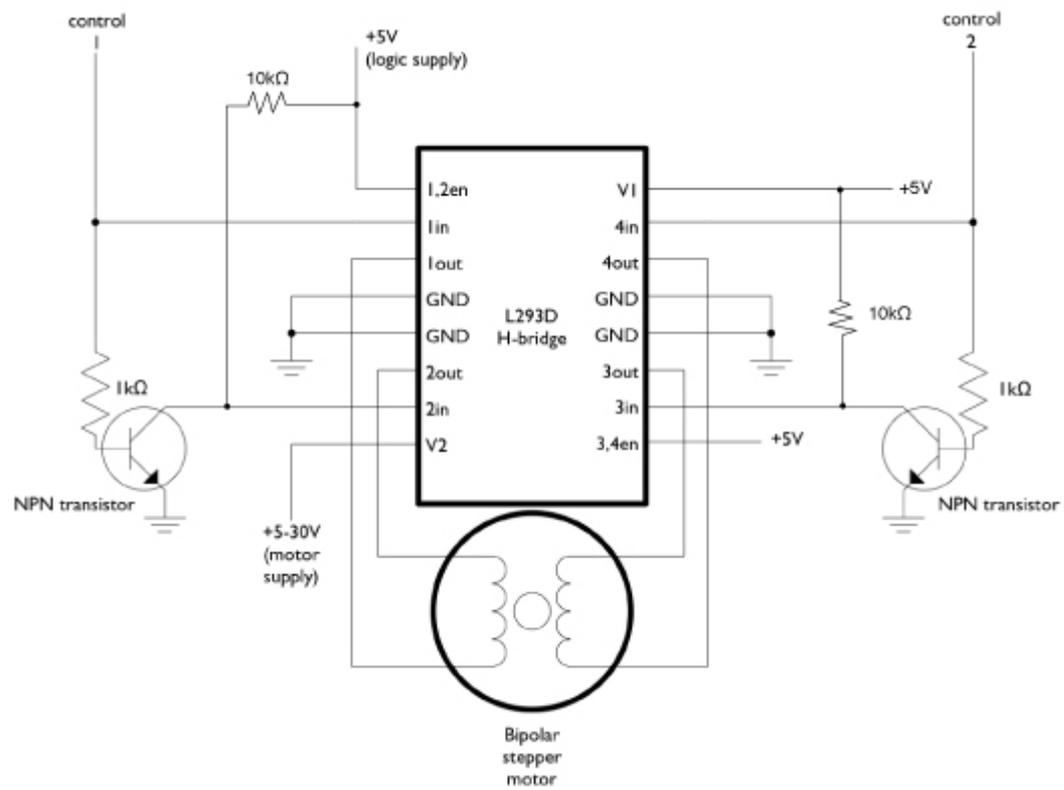
Feita esta alteração, é necessário um novo controlador para operar o Motor Bipolar, pois o Circuito ULN2003 não pode inverter o sentido da corrente. Para substituir este circuito, foi montado um novo circuito com o CI L293D. Este circuito utilizado juntamente com dois Transistores BC549-NPN necessita apenas de dois fios para ser controlado pelo Arduino Uno (ARDUINO-7). Com essa adaptação, o motor precisa ser alimentado com o dobro da tensão (10 volts). A Figura 27 apresenta o Circuito para Motor Bipolar com dois pinos de controle.

Figura 26 – Motor de Passo 28BYJ-48 Bipolar



Fonte: Os próprios autores (2019)

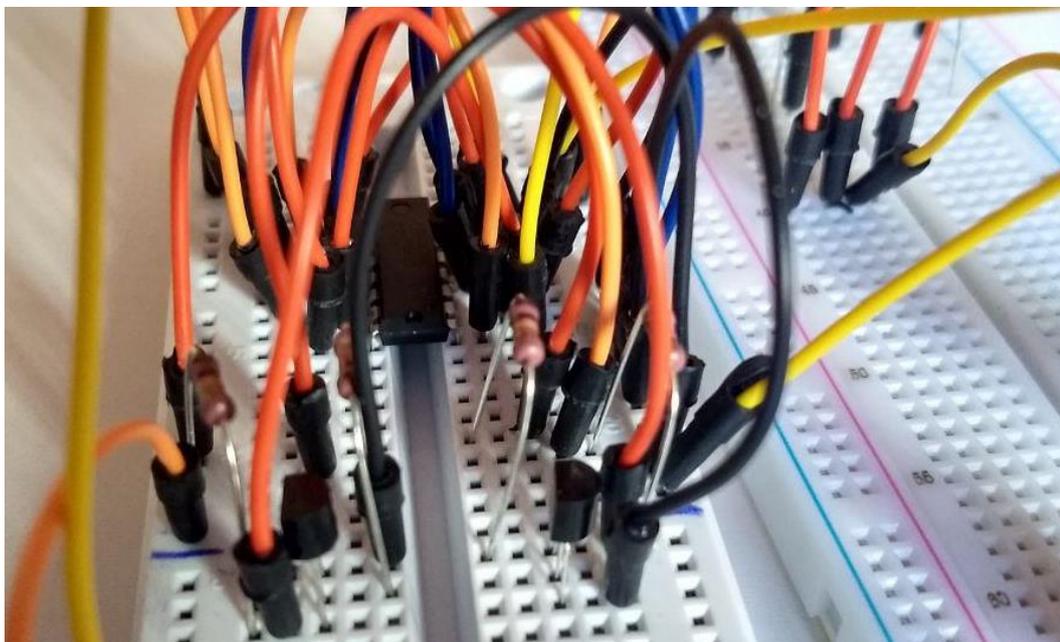
Figura 27 – Circuito para Motor de Passo Bipolar com 2 pinos



Fonte: Arduino-7

A Figura 28 apresenta o circuito montado para Motor Bipolar com dois pinos de controle.

Figura 28 – Circuito Montado para Motor de Passo Bipolar com 2 pinos



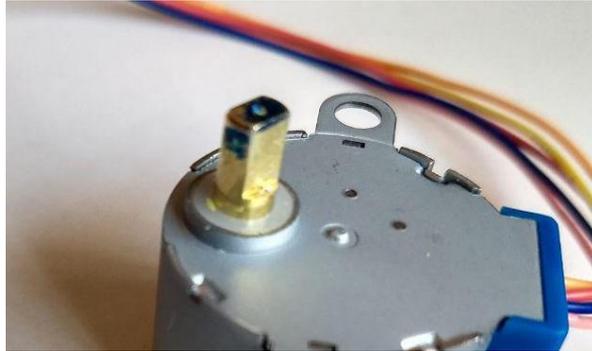
Fonte: Os próprios autores (2019)

Após ser feita a alteração e montado o circuito de controle, foram realizados alguns testes de afinação com o novo Motor Bipolar. Os resultados foram satisfatórios: o Motor de Passo conseguiu apertar a maioria das tarraxas dos violões utilizados nos testes. As tarraxas que se encontravam muito duras não foram apertadas pelo Motor de Passo, porém, deve-se considerar que essas tarraxas estavam em mal estado.

3.1.3.2 ENCAIXE

O Motor de Passo possui um encaixe em seu eixo conforme mostra a Figura 29. Para encaixar o Motor de Passo na tarraxa de um violão foi feita uma adaptação utilizando um encordoador. A parte do encordoador que é encaixada na tarraxa foi cortada, e foi feito um furo no centro desta peça para encaixá-la ao eixo do motor. A Figura 30 exemplifica um encordoador. A Figura 31 apresenta a adaptação realizada.

Figura 29 – Encaixe do Eixo do Motor de Passo



Fonte: Os próprios autores (2019)

Figura 30 – Encordoador Pegwinder



Fonte: STUDIO SOM JOÃO

Figura 31 – Adaptação com Encordoador



Fonte: Os próprios autores (2019)

3.1.3.3 COMPONENTES

Os componentes ilustrados nas Figuras 26 e utilizados pra construir o circuito do Sistema Motor estão listados na Tabela 7. A tabela também apresenta o preço pago por cada componente para a realização deste projeto. Os componentes foram adquiridos entre Abril e Maio de 2019.

Tabela 7 – Lista de Componentes do Circuito do Sistema Motor

Quantidade	Descrição do Componente	Preço Unitário	Preço Total
1	Circuito Ponte Dupla H L293D	R\$ 14,30	R\$ 14,30
1	Encordoador Pegwinder Parts Land Pw10 Amarelo	R\$ 15,00	R\$ 15,00
1	Motor de Passo + Drive ULN2003	R\$ 16,90	R\$ 16,90
1	Protoboard 830 Pontos MB-102	R\$ 15,90	R\$ 15,90
2	Resistor 10K 5% (1/4W)	R\$ 0,08	R\$ 0,16
2	Resistor 1K 5% (1/4W)	R\$ 0,08	R\$ 0,16
2	Transistor NPN - BC547	R\$ 0,16	R\$ 0,32
		TOTAL	R\$ 62,74

Fonte: Os próprios autores (2019)

3.2 CUSTO DO PROTÓTIPO

Considerando os valores apresentados como gastos em cada módulo do projeto, o custo total da Prototipação do Dispositivo Afinador Automatizado pode ser calculado. A Tabela 8 apresenta estes cálculos e o custo total da prototipação.

Tabela 8 – Cálculo dos Valores Totais da Prototipação do Dispositivo

Módulo	Preço
Sistema Sensor	R\$ 33,11
Sistema de Processamento	R\$ 57,08
Sistema Motor	R\$ 62,74
TOTAL	R\$ 152,93

Fonte: Os próprios autores (2019)

Para constatar se a prototipação do dispositivo afinador pode ser considerada de baixo custo, é necessário comparar o valor do custo total do protótipo com os valores de custo dos

dispositivos existentes apresentados anteriormente. A Tabela 9 apresenta uma comparação entre o custo do protótipo e o custo desses dispositivos.

Tabela 9 – Comparação de custos do Protótipo

Dispositivo	Preço
Protótipo de Afinador deste Trabalho	R\$ 152,93
Gibson Min ETune (usado)	R\$ 1.000,00
TronicalTune PLUS Set *	R\$ 898,90
Roadie Tuner Roadie 2 *	R\$ 543,49

Fonte: Os próprios autores (2019)

Os itens marcados com um * (asterisco) são itens que só podem ser importados, e os preços apresentados na tabela para esses itens já incluem o valor do frete. O valor dos itens e a cotação do Dólar foram consultados no dia 14 de junho de 2019 (Cotação do Dólar: R\$ 3,91).

Os valores na tabela dos dispositivos que só podem ser importados não incluem o valor do imposto de importação que, de acordo com a Portaria MF156/99 (BRASIL, 1999) pode chegar até a 60% do valor do produto, o que configura valor total de até R\$ 1.438,24 para o “TronicalTune”, e de até R\$ 869,58 para o “Roadie Tuner”.

O dispositivo “Gibson Min ETune” não está disponível no mercado nem para importação, mas uma unidade usada foi encontrada no site de vendas “Mercado Livre”, e mesmo que usado, o dispositivo tem alto valor comercial.

4 CONSIDERAÇÕES FINAIS

Um dos objetivos deste projeto era prototipar um dispositivo com o menor custo possível, e comparando o custo total deste projeto com o de outros dispositivos semelhantes, pode-se afirmar que o custo é relativamente baixo.

Durante o desenvolvimento do Sistema de Processamento, trabalhando com o algoritmo de detecção, percebeu-se que a taxa de amostragem do Arduino varia conforme as instruções são executadas. Como consequência dessa variação, a frequência do sinal amostrado é calculada com erro, pois está em função da taxa de amostragem, que é parâmetro para o cálculo. Em teste realizado, ficou comprovado que mesmo utilizando funções de interrupção para amostrar o sinal analógico, a taxa de amostragem não é constante no Arduino. Ainda em tentativa de calcular a taxa de amostragem anteriormente à realização dos cálculos, os resultados não foram satisfatórios.

O melhor resultado obtido em todos os testes, foi a utilização dos valores relativos de Taxa de Amostragem, calculados e fixados como parâmetro para cada corda do violão (algoritmo apresentado no Apêndice A). Mas por causa das variações nos valores detectados, não foi possível garantir a acuracidade do dispositivo para realizar a afinação.

Dentre os trabalhos pesquisados, tanto nos que serviram de base teórica a este projeto quanto nos encontrados durante a busca por uma solução para o problema de amostragem do Arduino, não foi encontrado um estudo sobre o porquê do Arduino não trabalhar bem no processamento de sinais. A conclusão deste trabalho de pesquisa e desenvolvimento deixa uma resposta mais aprofundada sobre a ineficiência do Arduino ao trabalhar com processamento de sinais, uma vez que a principal causa de erros na detecção de padrões analógicos está na incapacidade do Arduino de manter uma taxa de amostragem constante. Essas implicações evidenciam que o Arduino não é uma plataforma apropriada para o desenvolvimento de um dispositivo afinador automatizado.

Os resultados produzidos com pesquisas e experiências realizadas estão relatados neste Trabalho de Conclusão de Curso. O conteúdo técnico e as conclusões obtidas contribuem para pesquisas e estudos afins da comunidade acadêmica, quer para a verificação de informações quanto para a implementação de um dispositivo semelhante em uma plataforma de prototipação diferente. Sugere-se a trabalhos futuros que seja utilizada uma plataforma com melhor tecnologia de processamento, como a Raspberry Pi.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARDUINO-1. **What is Arduino?** Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 22 mai. 2018.
- ARDUINO-2. **Why Arduino?** Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 22 mai. 2018.
- ARDUINO-3. **Arduino Uno Rev3**. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>>. Acesso em: 27 abr. 2019.
- ARDUINO-4. **Arduino - Memory**. Disponível em: <<https://www.arduino.cc/en/Tutorial/Memory>>. Acesso em: 27 abr. 2019.
- ARDUINO-5. **Analog Read**. Documentação de Referência do Arduino. Disponível em: <<https://www.arduino.cc/reference/pt/language/functions/analog-io/analogread/>>. Acesso em: 27 abr. 2019.
- ARDUINO-6. **Attach Interrupt**. Documentação de Referência do Arduino. Disponível em: <<https://www.arduino.cc/reference/pt/language/functions/external-interrupts/attachinterrupt/>>. Acesso em: 18 mai. 2019.
- ARDUINO-7. **Circuit for Bipolar Stepper Motor**. Arduino Reference. Disponível em: <<https://www.arduino.cc/en/Reference/StepperBipolarCircuit>>. Acesso em 18 mai. 2019.
- BAND, ROADIE TUNER, AUTOMATIC GUITAR TUNER. **Roadie Tuner**. 2016. Disponível em: <<https://www.roadietuner.com/roadie1>>. Acesso em: 10 fev. 2019.
- BARBOSA, Daniele de Moraes A.; BURIAN JR, Yaro. **Iniciação Científica: Instrumentos Digitais e Teorema da Amostragem**. FADEP – Faculdade de Pato Branco, Pato Branco, jul. 2013. Disponível em: <https://www.fadep.br/engenharia-eletrica/congresso/pdf/116582_1.pdf>. Acesso em: 17 jun. 2019.
- BECCHI, Natan Gabriel. **Sistema automatizado para afinação de violão elétrico**. 2017. Monografia (Graduação em Engenharia de Controle e Automação) – Centro Universitário UNIVATES, Lajeado, 13 jul. 2017. Disponível em: <<http://hdl.handle.net/10737/1664>>. Acesso em: 20 mar. 2018.
- BERNI, Alexandre A. **Temperatura X Afinação de Instrumentos Musicais**. 2016. Santo angelo – Música na veia, 18 mai. 2016. Disponível em: <<http://blog.santoangelo.com.br/temperatura-x-afinacao-de-instrumentos-musicais/>>. Acesso em: 12 nov. 2018.
- BRASIL. Ministério da Fazenda. Secretaria da Receita Federal. Portaria MF nº 156, 24 jun. 1999. Disponível em: <<http://normas.receita.fazenda.gov.br/sijut2consulta/link.action?idAto=23977&visao=anotado>>. Acesso em: 19 jun. 2019.
- BRITES, Felipe Gonçalves; SANTOS, Vinicius Puga de Almeida. **Motor de Passo**. Pet Tele Universidade Federal Fluminense. Niterói, jul. 2008. Disponível em:

<<http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>>. Acesso em: 22 mai. 2018.

CERVI, Gabriel. **Tudo o que você sempre quis saber sobre: Aulas de Violão**. 2013. <<http://escolabackstage.blogspot.com/2013/11/tudo-o-que-voce-sempr-quis-saber-sobre.html>>. Acesso em: 27 abr. 2019.

CHAVIER, Luís Fernando. **Programação para Arduino - Primeiros Passos: Conceitos iniciais de programação para Arduino**. 2013. Circuitar, 16 out. 2013. Disponível em: <<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>>. Acesso em: 13 nov. 2018.

CLOTHIER, Kurt E. **Zener Diode Shunt Regulator**. Instructables, 2013. Disponível em: <<https://www.instructables.com/id/Arduino-Audio-Input/>>. Acesso em: 1 mai. 2019.

DICIO, DICIONÁRIO ONLINE DE PORTUGUÊS. **acurácia**. 7Graus. Disponível em: <<https://www.dicio.com.br/acuracia/>>. Acesso em: 13 nov. 2018.

EPIPHONE, MIN-ETUNE THE WORLD FIRST AFFORDABLE SEFL TUNNING GUITARS. **Min-ETune**. 2014. Disponível em: <<http://www.epiphone.com/Min-ETune.aspx>>. Acesso em: 13 nov. 2018.

GEORGE. **Unipolar Stepper Motor vs Bipolar Stepper Motors**. Simply Smarter Circuitry Blog, 2012. Disponível em: <<https://www.circuitspecialists.com/blog/unipolar-stepper-motor-vs-bipolar-stepper-motors/>>. Acesso em: 18 mai. 2019.

GHASSAEI-1, Amanda. **Arduino Audio Input**. Instructables, 2012. Disponível em: <<https://www.instructables.com/id/Arduino-Audio-Input/>>. Acesso em: 1 mai. 2019.

GHASSAEI-2, Amanda. **Arduino Frequency Detection**. Instructables, 2012. Disponível em: <<https://www.instructables.com/id/Arduino-Frequency-Detection/>>. Acesso em: 1 mai. 2019.

GRÜNIG, Bruno. **Teoria musical aplicada ao violão**. 1. ed., 2014.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 16:1975 – **Acoustics – Standard tuning frequency (Standard musical pitch)**. Genebra, 1975. 1p.

JANGEOX. **Change unipolar 28BYJ-48 to bipolar stepper motor**. Jangeox' blog, 2013. Disponível em: <<http://www.jangeox.be/2013/10/change-unipolar-28byj-48-to-bipolar.html>>. Acesso em: 18 mai. 2019.

KELLY, Anthony. **Arduino Guitar Tuner**. 2017. Disponível em: <<http://www.akellyirl.com/arduino-guitar-tuner/>>. Acesso em: 17 mai. 2019.

KIATRONICS. **28BYJ-48 – 5V Stepper Motor**. Disponível em: <<http://blog.eletragate.com/wp-content/uploads/2018/07/Motor28BYJ48Kiatronics.pdf>>. Acesso em: 17 mai. 2019.

LIMA-1, Edilson Gomes de. **Dicionário de termos e nomenclaturas em nanotecnologia, biotecnologia e novas ciências**. São Paulo: Estudos acadêmicos, 2005.

LIMA-2, Isabelle. **Arduino**: especificações por modelo. 2016. Blog AutoCore Robótica, 28 ago. 2016. Disponível em: <<http://autocorerobotica.blog.br/arduino-especificacoes-por-modelo/>>. Acesso em: 20 abr. 2019.

MAES, Willem. **How to make Arduino fast enough to...** 2018. Disponível em: <<http://www.optiloading.be/willem/Arduino/speeding.pdf>>. Acesso em: 12 jun. 2019.

PERUCHI, Charlene. **Física dos Instrumentos Musicais: Violão**. 2014. Disponível em: <<http://charlene-peruchi.wixsite.com/fisica-e-musica/violo>>. Acesso em: 13 nov. 2018.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. 2. ed. – Novo Hamburgo: Feevale, 2013.

RASPBERRY PI FOUNDATION. **What is a Raspberry Pi?** 2011. Disponível em: <<https://www.raspberrypi.org/about/>>. Acesso em: 19 dez. 2018.

SATO, Alberto M. **Raspberry Pi e Arduino: Qual é a diferença e qual é o melhor para o seu projeto?** 2018. Portal do Saber, 2 jun. 2018. Disponível em: <<http://portaldosaber.net/2018/02/raspberry-pi-e-arduino-qual-e-diferenca-e-qual-e-o-melhor-para-o-seu-projeto/>>. Acesso em: 13 nov. 2018.

SOUZA, Anderson R. de et al. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**, v. 33, n. 1, 1702, 2011. Disponível em: <<http://sbfisica.org.br/rbef/pdf/331702.pdf>> Acesso em: 18 dez. 2018.

STUDIO SOM JOÃO. **Encordoador Pegwinder Parts Land Pw10 Amarelo**. Disponível em: <<http://www.studiosomjoao.com.br/produto/encordoador-pegwinder-parts-land-pw10-amarelo/136891>> Acesso em 20 mai. 2019.

TEXAS INSTRUMENTS. **TL07xx Low-Noise JFET-Input Operational Amplifiers**. 2017. Disponível em: <<http://www.ti.com/lit/ds/symlink/tl074b.pdf>>. Acesso em: 17 mai. 2019.

TIGERTEK. **Differences Between Servo Motors and Stepper Motors**. TigerTek Industrial Services, 2014. Disponível em: <<http://www.tigertek.com/servo-motor-resources/differences-between-servo-stepper-motors.html>>. Acesso em: 20 dez. 2018.

VONK, Johan. **Pitch Detection on Arduino using Autocorrelation**. Los altos, 2015. Disponível em: <<https://coertvonk.com/sw/embedded/arduino-pitch-detector-13252/6/>>. Acesso em :14 mai. 2018.

ZUBEN, Paulo. **Música e tecnologia: o som e seus novos instrumentos**. 68 p., Irmãos Vitale, 2004.