



**Universidade de Brasília  
Faculdade de Tecnologia**

**Desenvolvimento de Tecnologia para  
Monitoramento de Tremores de Parkinson**

Arthur de Araújo Cavalcanti de Lima

TRABALHO DE GRADUAÇÃO  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília  
2022

**Universidade de Brasília  
Faculdade de Tecnologia**

**Desenvolvimento de Tecnologia para  
Monitoramento de Tremores de Parkinson**

Arthur de Araújo Cavalcanti de Lima

Trabalho de Graduação submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Orientador: Prof. Walter de Britto Vidal Filho

Brasília  
2022

A663d Araújo Cavalcanti de Lima, Arthur de.  
Desenvolvimento de Tecnologia para Monitoramento de Tremores de Parkinson / Arthur de Araújo Cavalcanti de Lima; orientador Walter de Britto Vidal Filho. -- Brasília, 2022.  
72 p.

Trabalho de Graduação em Engenharia de Controle e Automação -- Universidade de Brasília, 2022.

1. Doença de Parkinson. 2. Tremor. 3. Frequência de Tremor. 4. Transformada Rápida de Fourier. I. de Britto Vidal Filho, Walter, orient. II. Título

**Universidade de Brasília  
Faculdade de Tecnologia**

**Desenvolvimento de Tecnologia para Monitoramento de  
Tremores de Parkinson**

Arthur de Araújo Cavalcanti de Lima

Trabalho de Graduação submetido como re-  
quisito parcial para obtenção do grau de Enge-  
nheiro de Controle e Automação.

Trabalho aprovado. Brasília, 22 de Setembro de 2022:

---

**Prof. Walter de Britto Vidal Filho,**  
**UnB/FT/ENM**  
Orientador

---

**Prof. Jones Yudi Mori Alves da Silva,**  
**UnB/FT/ENM**  
Examinador interno

---

**Prof. Carlos Humberto Llanos Quintero,**  
**UnB/FT/ENM**  
Examinador interno

---

**Prof. Flávio José Aguiar Soares,**  
**Instituto Federal de Educação, Ciência e**  
**Tecnologia do Amazonas -**  
**IFAM/Engenharia Mecânica**  
Examinador externo

Brasília  
2022

*Este trabalho é dedicado à minha família e em especial a meu falecido avô Mário.*

# Agradecimentos

Agradeço a minha Família por me incentivar a ter o espírito do aprendizado e a buscar conhecer novas experiências sem preconceitos. Agradeço também a todos que me apoiaram e ajudaram nesta longa caminhada.

# Resumo

O mal de Parkinson é uma doença que afeta o sistema nervoso central, atingindo o paciente de forma irreversível, possuindo vários sintomas e efeitos, limitando sua vida e podendo evoluir para estados graves, caso não seja tratado no começo da doença. O presente trabalho visou estudar um dos sintomas iniciais e principais que é o tremor dos membros. Buscou-se obter dados dos tremores através de sensores e adquirir características através de software, obtendo parâmetros para que os médicos possam monitorar a evolução da doença. Foi estudada e desenvolvida uma pulseira para medir as oscilações dos tremores, cujos resultados apresentaram valores condizentes com a realidade.

**Palavras-chave:** Doença de Parkinson. Tremor. Frequência de Tremor. Transformada Rápida de Fourier.

# Abstract

Parkinson's disease is a disorder that affects the central nervous system, reaching the patient irreversibly, having several symptoms and effects, as well as limiting their life and possibly progressing to serious states, if not treated in early stages. The present work aimed to study one of the initial and main symptoms: the limb tremor. The aim was to acquire data on tremors through sensors and characteristics through software, obtaining parameters so that doctors can monitor the evolution of the disease. A bracelet was studied and developed to measure the oscillations of tremors, whose results were consistent.

**Keywords:** Parkinson Disease. Tremor. Tremor Frequency. Fast Fourier Transform.

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Contextualização</b>	<b>14</b>
<b>1.2</b>	<b>A Doença de Parkinson</b>	<b>14</b>
1.2.1	Caracterização dos Tremores	15
<b>1.3</b>	<b>Objetivo Geral</b>	<b>15</b>
<b>1.4</b>	<b>Objetivos Específicos</b>	<b>15</b>
<b>1.5</b>	<b>Metodologia e Organização do Trabalho</b>	<b>16</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>17</b>
<b>2.1</b>	<b>Estado da Arte: Sensores para Caracterização do Tremor da Doença de Parkinson</b>	<b>17</b>
2.1.1	Comparações Entre Trabalhos	19
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
<b>3.1</b>	<b>Movimento Harmônico Simples</b>	<b>20</b>
<b>3.2</b>	<b>Frequência de Tremores da Doença de Parkinson</b>	<b>21</b>
<b>3.3</b>	<b>Discretização e Digitalização de um Sinal Analógico Contínuo</b>	<b>22</b>
3.3.1	Conversor Analógico-Digital	24
<b>3.4</b>	<b>Análise de Fourier</b>	<b>25</b>
3.4.1	Série de Fourier	26
3.4.2	Transformada de Fourier	26
3.4.2.1	Transformada Discreta de Fourier	27
3.4.2.2	Transformada Rápida de Fourier	29
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA</b>	<b>31</b>
<b>4.1</b>	<b>Requisitos</b>	<b>31</b>
<b>4.2</b>	<b>Desenvolvimento do Sistema Embarcado</b>	<b>31</b>
4.2.1	Placa Controladora	31
4.2.1.1	WiFi LoRa32 V2	32
4.2.2	Acelerômetro e Giroscópio MPU6050	33
4.2.3	Alimentação de Energia	36
<b>4.3</b>	<b>Software</b>	<b>37</b>
4.3.1	Arquitetura da Solução	37
<b>4.4</b>	<b>Capa de Proteção</b>	<b>40</b>
<b>4.5</b>	<b>Diagrama de Funcionamento do Sistema</b>	<b>42</b>

<b>5</b>	<b>TESTES E RESULTADOS</b> . . . . .	<b>43</b>
<b>5.1</b>	<b>Testes Realizados</b> . . . . .	<b>43</b>
5.1.1	Bancada de Testes . . . . .	43
5.1.2	Testes Realizados . . . . .	46
<b>5.2</b>	<b>Resultados</b> . . . . .	<b>48</b>
5.2.1	Resultados do Teste com Motor alimentado a 5V . . . . .	48
5.2.2	Resultados do Teste com Motor alimentado a 7V . . . . .	50
5.2.3	Resultados do Teste de Verificação de Funcionamento de Eixos de Orientação	52
5.2.4	Análise de Resultados . . . . .	54
<b>6</b>	<b>CONCLUSÕES</b> . . . . .	<b>55</b>
<b>6.1</b>	<b>Trabalhos Futuros</b> . . . . .	<b>55</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>57</b>
	<b>APÊNDICES</b>	<b>61</b>
	<b>APÊNDICE A – CÓDIGOS DE PROGRAMAÇÃO</b> . . . . .	<b>62</b>
<b>A.1</b>	<b>Códigos de programação</b> . . . . .	<b>62</b>
	<b>APÊNDICE B – VÍDEO DEMONSTRANDO MOVIMENTO DA BAN-</b>	
	<b>CADA DE TESTES</b> . . . . .	<b>72</b>

# Lista de ilustrações

Figura 1 – Oscilador Harmônico Massa-Mola. . . . .	21
Figura 2 – Espectro de um sinal amostrado a partir do teorema de Nyquist. . . . .	23
Figura 3 – Diagrama de funcionamento de um ADC. . . . .	24
Figura 4 – Sinal Amostrado no Domínio do Tempo. (LATHI, 2006) . . . . .	28
Figura 5 – Espectro de Fourier do Sinal Amostrado. (LATHI, 2006) . . . . .	28
Figura 6 – Microcontrolador WiFi LoRa 32.(HELTEC, 2018) . . . . .	32
Figura 7 – Diagrama de Pinos do WiFi LoRa 32.(HELTEC, 2018) . . . . .	33
Figura 8 – MPU6050. . . . .	33
Figura 9 – Acelerômetro.(NEDELKOVSKI, 2018) . . . . .	34
Figura 10 – Giroscópio.(NEDELKOVSKI, 2018) . . . . .	34
Figura 11 – Fluxograma do código implementado. . . . .	39
Figura 12 – Projeto Tridimensional da Pulseira de Medição de Frequência. . . . .	40
Figura 13 – Impressão do corpo da Pulseira de Medição de Frequência. . . . .	40
Figura 14 – Pulseira de Medição de Frequência de Tremores Finalizada. . . . .	41
Figura 15 – Diagrama de Funcionamento do Sistema. . . . .	42
Figura 16 – Testador de Frequência Fixa. . . . .	43
Figura 17 – Ilustração de movimento. . . . .	44
Figura 18 – Sensor de Efeito Hall A3144. . . . .	45
Figura 19 – Bancada de Teste completa. . . . .	45
Figura 20 – Eixos de Orientação do MPU6050 . . . . .	47
Figura 21 – Eixos de Orientação da Pulseira . . . . .	47
Figura 22 – Leitura da Pulseira para Motor em 5V. . . . .	48
Figura 23 – Leitura do Sensor de efeito Hall vista no osciloscópio para entrada de 5V. . . . .	49
Figura 24 – Resultado do cálculo da FFT no MATLAB para entrada em 5V. . . . .	49
Figura 25 – Leitura da Pulseira para Motor em 7V. . . . .	50
Figura 26 – Leitura do Sensor de efeito Hall vista no osciloscópio para entrada de 7V. . . . .	50
Figura 27 – Resultado do cálculo da FFT no MATLAB para entrada em 7V. . . . .	51
Figura 28 – Leitura da Pulseira para o Eixo X. . . . .	52
Figura 29 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo X. . . . .	52
Figura 30 – Leitura da Pulseira para o Eixo Y. . . . .	53
Figura 31 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo Y. . . . .	53
Figura 32 – Leitura da Pulseira para o Eixo Z. . . . .	53
Figura 33 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo Z. . . . .	54
Figura 34 – QR code de Vídeo Demonstrativo. <a href="https://youtu.be/NbcQGNvjPUU">https://youtu.be/NbcQGNvjPUU</a> . . . . .	72

# Lista de tabelas

Tabela 1 – Comparação de Trabalhos. . . . .	19
Tabela 2 – Configuração de Filtro Passa-baixas Digital. (INVENSENSE, 2013) . . .	35
Tabela 3 – Sensitividade do Acelerômetro pelo bit menos significativo(LSB). (INVENSENSE, 2013) . . . . .	35
Tabela 4 – Erros Percentuais dos Testes Realizados. . . . .	54

# Lista de abreviaturas e siglas

$\omega$	Velocidade Angular . . . . .	20
$\Omega_0$	Frequência Fundamental Discreta . . . . .	28
$\omega_0$	Frequência Angular Fundamental . . . . .	28
$\phi_0$	Posição Inicial . . . . .	21
$\pi$	Pi . . . . .	20
$f$	Frequência . . . . .	20
$f_0$	Frequência Fundamental . . . . .	28
$f_s$	Frequência de amostragem . . . . .	23
$f_{max}$	Frequência máxima . . . . .	23
$T$	Período . . . . .	20
$T_0$	Período de Janelamento . . . . .	27
A	Amplitude . . . . .	21
ADC	Conversor Analógico-Digital . . . . .	24
BLE	Bluetooth Low Energy . . . . .	32
DLFP	Digital Low Pass Filter . . . . .	35
DMP	Digital Motion Processor . . . . .	35
DP	Doença de Parkinson . . . . .	14
FFT	Fast Fourier Transform . . . . .	25
GPIO	General Purpose Input/Output . . . . .	32
GRACO	Grupo de Automação e Controle . . . . .	43
Hz	Hertz . . . . .	22
I2C	Inter-Integrated Circuit . . . . .	32
IMU	Inertial Measurement Unit . . . . .	36
LoRa	Long Range . . . . .	32
LSB	Least Significant Bit . . . . .	35
mA	Miliamper . . . . .	36
mAh	Miliamper Hora . . . . .	36

MATLAB	Matrix Labortory .....	46
MHS	Movimento Harmônico Simples.....	20
OLED	Organic Light-Emitting Diode.....	32
s	Segundo .....	20
SAR	Successive-approximation-register.....	32
TDF	Transformada Discreta de Fourier .....	25
V	Volts.....	25

# 1 Introdução

## 1.1 Contextualização

Com a evolução tecnológica e a crescente densificação de circuitos lógicos programáveis, tornaram-se possíveis controladores cada vez menores e mais rápidos. Com essa tecnologia, é possível criar dispositivos para portadores de doenças físicas que não impeçam o indivíduo de realizar tarefas corriqueiras com a dificuldade de carregar um dispositivo que atrapalhe ou deixe desconfortável.

A Doença de Parkinson se encaixa no espectro de doenças que podem ter portadores auxiliados pelo desenvolvimento tecnológico, gerando grande melhora na qualidade de vida. Esta doença afeta a mobilidade do portador por lhe causar, dentre outros sintomas, tremor dos membros. Esta condição impede a realização de tarefas simples e corriqueiras do paciente, como se alimentar ou até mesmo falar, tornando-o dependente de uma ajuda externa e, em muitos casos, abrindo caminho para problemas psicológicos, chegando em alguns casos a uma severa depressão. O estudo físico desses tremores deve ser cada vez mais compreendidos para que soluções possam ser alcançados. (FERNANDES; SILVA, F.; SILVA, T., 2014)

No Brasil estima-se que cerca de duzentas mil pessoas sofram da doença (MAGALHÃES, 2015) e diversos trabalhos foram desenvolvidos com o intuito de compreender a doença e auxiliar seus portadores a reconquistar certo grau de independência. Muitos desse trabalhos buscam desenvolver dispositivos compensadores dos sintomas apresentados, mas para que estes sejam efetivos, é necessária uma compreensão completa da doença. Este trabalho busca, assim como outros que serão tratados com mais detalhe no capítulo 2, busca essa melhor compreensão dos sintomas da Doença de Parkinson, focando com maior especificidade nos tremores envolvidos.

## 1.2 A Doença de Parkinson

A Doença de Parkinson (DP) foi descrita em 1817 por James Parkinson em seu trabalho: "Um Ensaio Sobre a Paralisia Trêmula", apesar de já ter sido notada milênios antes em literaturas indus e chinesas, ou até mesmo por outros autores europeus contemporâneos, como Sylvius la Boë, Sauvages ou Jean-Martin Charcot.(GOETZ, 2011). Este último autor foi o primeiro a nomear o mal como "Doença de Parkinson", rejeitando a designação anterior de paralisia agitante ou paralisia trêmula, pois reconheceu que os pacientes com doença de Parkinson possuem acentuada rigidez muscular e não têm necessariamente tremores.

---

(CHARCOT, 1888)

### 1.2.1 Caracterização dos Tremores

O tremor é um distúrbio muito comum, sendo um movimento involuntário, rítmico e oscilatório de alguma parte do corpo. Ocorre devido a contrações involuntárias de músculos antagonistas. Os tremores são prejudiciais ao paciente, tanto fisicamente, quanto socialmente, sendo que em muitos casos a desordem não é notada até o portador se tornar idoso. É importante diferenciar o tremor de outros movimentos, que podem ser involuntários e rápidos, mas espasmódicos por natureza, em vez de rítmicos. O tremor é normalmente associado à doença de Parkinson, uma vez que esta é uma doença comum e trágica, que acomete milhares de pessoas, podendo levar a morte em muitos casos. (BHAGWATH, 2001)

Um dos principais e mais famosos sintomas da Doença de Parkinson (DP) é o tremor, que apesar de ser uma característica bastante associada à DP, não é a única comorbidade a apresentá-la. Outra doença que apresenta tal distinção, a qual é bastante confundida com a DP, é o Tremor Essencial, a qual, como o próprio nome indica, gera tremores nos membros superiores nos portadores. Dessa forma, faz-se bastante complexa a distinção entre as duas doenças, principalmente ao se estudar a frequência de tremores, como observado em (CICHACZEWSKI et al., 2014).

Os tremores ainda podem ser classificados com base em suas características clínicas. A maioria pode ser classificada como tremor de repouso, postural ou de intenção. O tremor de repouso ocorre em um membro ou parte do corpo que é completamente apoiado contra a gravidade sem absolutamente nenhuma contração, geralmente com os braços apoiados sobre as coxas. Já o tremor postural ocorre quando se mantém os braços esticados à frente e com a postura constante e, por fim, o de intenção ocorre quando o membro se aproxima de algum alvo que será tocado. (FERNANDES; SILVA, F.; SILVA, T., 2014)

## 1.3 Objetivo Geral

O objetivo geral do presente trabalho é desenvolver um dispositivo capaz de medir frequências de tremores de pacientes portadores da Doença de Parkinson, permitindo métricas mais precisas desse sintoma, que ainda é muito medido de forma empírica baseado na experiência e na observação do profissional de saúde.

## 1.4 Objetivos Específicos

O presente trabalho busca:

- O desenvolvimento de uma pulseira medidora de frequência de tremores através de uma placa controladora e um sensor acelerômetro;
- Com os dados colhidos, objetiva-se a análise da frequência dos movimentos através de um software embarcado;
- Disponibilização dos dados tratados e colhidos para o usuário através de leitura direta na própria pulseira, ou em um servidor web.

## 1.5 Metodologia e Organização do Trabalho

O presente documento está estruturado da seguinte forma:

- Introdução: capítulo atual contendo introdução, contextualização, exposição da problemática e definição dos objetivos do projeto;
- Revisão bibliográfica: em acordo com o método científico, este capítulo contempla as pesquisas e revisões bibliográficas acerca de projetos similares que serviram de inspiração, bem como o estado da arte referente a dispositivos desenvolvidos para colheita de dados relacionados à tremores da DP;
- Fundamentação teórica: capítulo que aborda toda a teoria e conhecimentos necessários para a realização deste trabalho;
- Desenvolvimento do Sistema: capítulo que apresenta a implementação do sistema, seus componentes e justificativas para seu uso, além de descrever a lógica implementada em software;
- Testes e Resultados: capítulo que aborda os testes realizados e resultados obtidos;
- Considerações Finais: capítulo contendo a validação dos testes realizados e perspectivas para trabalhos futuros.

A metodologia utilizada no desenvolvimento deste trabalho para avaliação das propostas levantadas foi através de estudo e pesquisa bibliográfica, além da montagem e experimentação tecnológica. Com o intuito de organizar o trabalho, foram realizadas reuniões semanais com o professor orientador e com colegas de curso, estabelecendo prazos e tarefas nesses encontros.

## 2 Revisão Bibliográfica

Neste capítulo será apresentada a revisão bibliográfica usada para guiar este projeto, a qual auxiliou como inspiração para o presente projeto, as tecnologias utilizadas para a aquisição e análise de dados e, por fim, o estado da arte que outras pesquisas e usam sensores inerciais para caracterização do tremor da DP.

Projetos com cunho similares, realizados na Universidade de Brasília e em outras do Brasil, foram pesquisados para auxiliar no desenvolvimento deste trabalho. Todos estes apresentaram projetos com o intuito da coleta de dados relacionados a tremores da Doença de Parkinson.

Dessa forma, o trabalho aqui proposto tem por intuito o desenvolvimento de um equipamento semelhante que possa adquirir dados relacionados à DP e obter via software parâmetros para que os médicos possam monitorar a evolução da doença.

### 2.1 Estado da Arte: Sensores para Caracterização do Tremor da Doença de Parkinson

A partir do momento em que foi descoberta a Doença de Parkinson, fez-se necessário seu diagnóstico em pacientes. Essa doença é, porém, extremamente gradativa e apresenta um grau de dificuldade alto para sua detecção precoce. Todavia, com o crescente desenvolvimento tecnológico na área de eletrônica e de sensores, torna-se cada vez mais favorável a detecção e o estudo da DP, através da aquisição de dados precisos e o uso de técnicas matemáticas e estatísticas para tratá-los com acurácia.

Na academia há diversos exemplos de dispositivos desenvolvidos para obter e tratar dados relacionados com a doença de Parkinson. A seguir, alguns desses dispositivos serão expostos.

Na Universidade de Brasília, dois trabalhos de Conclusão de Curso foram apresentados que se relacionam ao tópico: (GUIMARÃES, 2017) e (COSTA; PAIXÃO, 2018). O primeiro foi um trabalho visando a construção de aparelho para coleta de sinais eletromiográficos de superfície em pacientes portadores da DP. Já o segundo, foi o desenvolvimento de um detector preliminar da DP com o uso de Aprendizado de Máquina utilizando dados colhidos com o dispositivo do primeiro trabalho e o algoritmo *Random Forest* para a detecção dos padrões da doença.

Outros trabalhos foram desenvolvidos por todo o Brasil, sendo um deles o relatório do Programa de Iniciação Científica que colaborou para o desenvolvimento do trabalho aqui

apresentado, (SOUZA; COLETTA; SOARES, 2018), do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. O projeto propôs o desenvolvimento de um dispositivo capaz medir a frequência de tremores apresentados por portadores de comorbidades associadas, em especial a DP. Este aparelho proposto teria a vantagem de ser compacto e independente de fios para a coleta de dados, armazenando os dados em memória flash e realizando processamento de dados em software de computador, contudo o hardware final desenvolvido não funcionou como esperado, apesar de protótipos com os mesmos componentes sim.

Um dispositivo semelhante foi apresentado em duas teses de Doutorado da Universidade Federal de Uberlândia, (PÁDUA MACHADO, 2017) e (SILVA, A. P. S. P. B. da, 2018). O dispositivo utilizado em ambas as teses, denominado TREMSEM, apresentava componentes semelhantes aos propostos neste trabalho, microcontrolador da família Arduino e IMUs, somando-se ainda sensores eletromiográficos, interligados por cabos. Os dados obtidos foram filtrados e a transformada de Hilbert foi aplicada para a estimação da amplitude e frequência instantâneas.

O artigo previamente citado na introdução (1.2.1) (CICHACZEWSKI et al., 2014), desenvolvido por pós-graduandos da Universidade Tecnológica Federal do Paraná, fez uso de acelerômetros MMA6260Q para a aquisição dos dados apresentados. Há também uma pulseira de medição apresentada no artigo da revista Ensino Superior, (NORTE, 2019), desenvolvida por alunos do Instituto Mauá de Tecnologia, que possui proposições semelhantes a este trabalho, apesar de não entrar em detalhes.

Ademais, trabalhos internacionais como (SPYERS-ASHBY; STOKES, 2000), (PATEL et al., 2009) e (RISSANEN et al., 2011) desenvolveram aparelhos com objetivos semelhantes ao proposto neste trabalho, alguns usando acelerômetros e outros associando ainda o uso de sensores eletromiográficos, ou eletromagnéticos tridimensionais. Em (WILE; RANAWAYA; KISS, 2014), foi proposto o uso de *smart watches* para substituir técnicas tradicionais de medição de tremores através de acelerômetros e similares, uma vez que estes aparelhos já possuem os sensores essenciais para a medição de tremores em seu *hardware* e permitem o uso de dispositivos celulares para realizar o processamento dos dados colhidos.

Em (ZHOU et al., 2018), um dispositivo similar foi desenvolvido, com o uso de acelerômetros e magnetômetros. Porém, este estudo se diferencia de outros por analisar o tremor de pacientes da DP no domínio do tempo e da frequência para propor um estimador de frequência para complementar a performance de dispositivos vestíveis de supressão de tremor.

Na pesquisa para este trabalho não foram encontrados produtos comerciais cujo objetivo específico é a medição de tremores de Parkinson, deixando o desenvolvimento de dispositivos deste cunho para trabalhos acadêmicos. Todavia, o mercado apresenta grande variedade de produtos destinados ao controle dos tremores em si, através de compensadores eletrônicos, amortecedores de líquidos não-newtonianos, e estimulantes elétricos. Além

disso, também existe uma variedade de utensílios que visam a facilitação do uso para usuários com tremores, como colheres com compensador de tremores, roupas adaptativas de fácil vestimenta e utensílios com maior volume e peso para o fácil manuseio.

### 2.1.1 Comparações Entre Trabalhos

A tabela 1 compara trabalhos expostos em 2.1 com a proposta que este trabalho pretende implementar.

<b>Pesquisa</b>	<b>Tipo de Sensor</b>	<b>Observações</b>
(GUIMARÃES, 2017) (COSTA; PAIXÃO, 2018)	Eletromiográfico de superfície	Implementou algoritmo Random Forest para a detecção dos padrões da doença
(SOUZA; COLETTA; SOARES, 2018)	IMU	Compacto e portátil, processamento de dados via software em computador e armazenamento de dados em cartão SD
(PÁDUA MACHADO, 2017) (SILVA, A. P. S. P. B. da, 2018)	IMU + Eletromiográfico de superfície	Sensores conectados via cabo, processamento de dados via software em computador
(CICHACZEWSKI et al., 2014)	Acelerômetros	Estudo voltado para caracterização de valores de frequência de tremores em pacientes da DP
(WILE; RANAWAYA; KISS, 2014)	<i>Smart Watches</i>	Uso de produtos comercializados em larga escala
(ZHOU et al., 2018)	Acelerômetros + Magnetômetros	Proposição de estimador para complementar um compensador pré-desenvolvido
Este Trabalho de Graduação	IMU	Compacto e portátil, processamento de dados embarcado, envio de dados via WiFi

Tabela 1 – Comparação de Trabalhos.

Como pode ser observado na tabela, este projeto busca uma solução portátil e com processamento embarcado diretamente no dispositivo. Essas características diferenciam o projeto aqui proposto de outros previamente realizados.

## 3 Fundamentação Teórica

### 3.1 Movimento Harmônico Simples

O movimento harmônico simples (MHS) é aquele em que um corpo oscila em torno de uma posição de equilíbrio devido à ação de uma força restauradora. No MHS, não há forças dissipativas, como as forças de atrito e arrasto, e, por isso, a energia mecânica total do sistema é conservada. (HELERBROCK, 2020)

O movimento harmônico simples é um modelo físico que pode ser utilizado para investigar diversos sistemas mais complexos onde há forças restauradoras, como nas interações elétricas dos átomos e nas atrações gravitacionais entre planetas. (CARVALHO, 2014)

O MHS é resultado da ação de uma força que tende a manter algum corpo em uma posição de equilíbrio, tal como acontece com uma mola quando esticada ou comprimida, sujeita à ação de uma força elástica. Nesse tipo de movimento, a soma da energia cinética com a energia potencial é sempre constante, por isso pode-se dizer que há conservação da energia mecânica. (HELERBROCK, 2020)

O número de oscilações realizadas por um sistema em MHS que são concluídas a cada segundo é a frequência do sistema. O período é igual ao tempo gasto para que o sistema complete uma oscilação e, portanto, é o inverso da frequência. As unidades de medida da frequência e do período do MHS são, respectivamente, o Hertz (Hz) e o segundo (s). As fórmulas usadas para calcular essas grandezas são as seguintes:

$$f = \frac{n}{\Delta t} \quad T = \frac{1}{f} \quad (3.1)$$

onde  $n$  é o número de oscilações,  $\Delta t$  é o intervalo de tempo,  $f$  a frequência e  $T$  o período. (CARVALHO, 2014)

Além das grandezas frequência e período, o MHS pode ser descrito em função de grandezas angulares. Tais grandezas permitem-nos saber em qual posição um objeto se encontra, bem como indicar quais são suas medidas de energia cinética e potencial instantâneas. A mais importante das grandezas angulares relacionadas ao MHS é a frequência angular, também conhecida como velocidade angular ou pulsação.

$$\omega = 2\pi f, \quad \omega = \frac{2\pi}{T} \quad (3.2)$$

onde  $\omega$  é a velocidade angular. (HELERBROCK, 2020)

As fórmulas mais importantes do movimento harmônico simples são as equações horárias da posição, velocidade e aceleração. Essas equações nos permitem determinar a

posição, a velocidade ou a aceleração de um objeto em MHS em determinado instante de tempo.

$$x(t) = A\cos(\omega t + \phi_0) \quad (3.3)$$

$$v(t) = -\omega A\sin(\omega t + \phi_0) \quad (3.4)$$

$$a(t) = -\omega^2 A\cos(\omega t + \phi_0) \quad (3.5)$$

onde  $A$  é a amplitude (máxima distância que um corpo pode ter do ponto de equilíbrio),  $t$  é o instante de tempo e  $\phi_0$  é a fase inicial (posição inicial do movimento). (HELERBROCK, 2020)

A figura 1 exemplifica um MHS de um sistema massa-mola e ilustra a natureza senoidal da onda obtida.

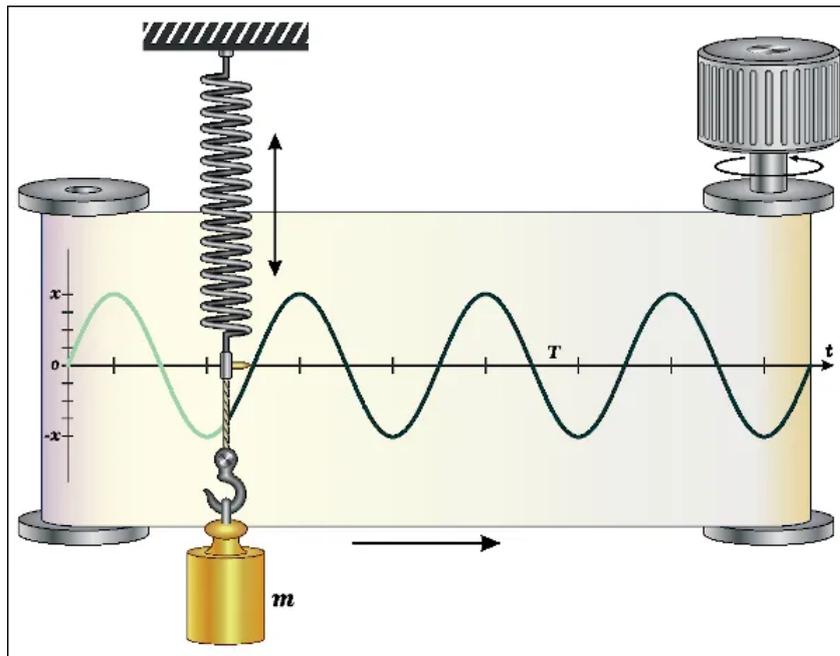


Figura 1 – Oscilador Harmônico Massa-Mola.

No contexto deste trabalho, será buscada uma associação entre determinadas atividades e a frequência de tremor de portadores da doença de Parkinson. Fazendo uso da transformada de Fourier, poderão ser obtidas derivações trigonométricas dos sinais colhidos e, por conseguinte, identificar frequências de MHSs através das equações 3.3, 3.4 e 3.5.

## 3.2 Frequência de Tremores da Doença de Parkinson

Como visto em (BORGES; FERRAZ, 2006) o distúrbio do tremor pode ser classificado de acordo com a etiologia, fenomenologia, frequência e localização. Neste trabalho a

classificação frequencial será a mais estudada, podendo ser baixa ( $<4\text{Hz}$ ), média (4 e  $7\text{Hz}$ ) e alta ( $>7\text{Hz}$ ). De maneira mais específica será estudado o tremor associado à fenomenologia do mal de Parkinson.

O típico tremor Parkinsoniano costuma ser sentido quando o paciente se encontra em repouso, desaparecendo no sono e em completo relaxamento. A presença de tremores postural e de intensão são comuns também na DP, porém somadas ao de repouso. O tremor acomete com mais frequência os membros superiores, mas pode ser presente nos inferiores também. Em momentos iniciais da doença, a frequência típica é média, entre 4 e 8 Hertz (Hz), e posteriormente atinge um grau alto, entre 6 e  $12\text{Hz}$ . (BORGES; FERRAZ, 2006)

Em contraste com a DP, tremores essenciais costumam ser sentidos durante o movimento, além de que seus sintomas não abreviam necessariamente o tempo de vida do paciente, por mais severos que possam se tornar. A Doença de Parkinson, por sua vez, tende a progredir e pode encurtar o tempo de vida de seu acometido.

Como observado em (CICHACZEWSKI et al., 2014), ao se estudar 62 pacientes com DP, foi encontrada uma média de frequência de  $5,29 \pm 1,18$  Hz em repouso,  $5,79 \pm 1,39$  postural e  $6,48 \pm 1,34$  de intensão. Dessa forma, observa-se experimentalmente como os tremores provocados pela DP apresentam uma frequência para os portadores que pode atrapalhar atividades corriqueiras que necessitem de maior delicadeza motora, como comer, escovar os dentes ou mesmo escrever, causando uma dependência do paciente por ajuda externa, ou até mesmo casos de depressão.

Por fim, é importante salientar o motivo de neste trabalho ser dada maior relevância à análise e cálculo da frequência do tremor da doença em detrimento à análise de amplitude. O sensor utilizado neste trabalho mede a aceleração do movimento, o que tornaria necessário a aplicação de uma integral dupla no sinal obtido para obter um valor de amplitude do movimento do tremor. A aplicação de integral dupla via software exige um maior poder de processamento do microcontrolador e aumenta os erros associados à leitura realizada pelo sensor, tornando os valores menos acurados. Para não deixar esse aspecto completamente de fora do escopo deste trabalho, é calculada uma média dos valores de amplitude de aceleração do sinal, a qual pode dar uma ideia do grau de severidade do tremor.

### 3.3 Discretização e Digitalização de um Sinal Analógico Contínuo

Um sinal analógico é um conjunto de dados capaz de assumir quaisquer valores em uma faixa contínua no eixo  $y$ . Em compensação, um sinal que admite uma quantidade finita de valores nesse eixo é denominado digital. Ademais, é importante ressaltar que um sinal ser contínuo ou discreto no tempo (eixo  $x$ ) significa dizer que, respectivamente, ele possuirá

um valor em qualquer ponto desse eixo ou apresentará apenas em determinadas posições (LATHI, 2006).

No estudo de digitalização de modelos, sabe-se que é de extrema importância se ater a algumas características do sinal, visto que estes devem estar em concordância com alguns parâmetros antes do processo de conversão do sinal de analógico para digital de modo a prevenir que ocorram inconvergências que, conseqüentemente, prejudicam a integridade da informação a ser trabalhada.

Um parâmetro de extrema importância nesse processo é a chamada frequência de amostragem. Ela consiste na quantidade de vezes em que informações do sinal, ou amostras, foram coletadas por unidade de tempo (LYONS; FUGAL, 2014).

Como mencionado anteriormente, a frequência de amostragem, ou  $f_s$ , é um fator de imensa importância visto que o seu valor determinará a qualidade e a integridade do sinal. Dependendo do valor que essa frequência assume, informações do sinal podem ser perdidas e não é possível reconstruir um sinal com as mesmas características que o original a partir das amostras coletadas. Por outro lado, há situações em que pode ocorrer um excesso desnecessário de dados coletados.

Com um teorema que leva o seu nome, o engenheiro sueco Harry Nyquist (1889-1976) propôs, para evitar que informações necessárias não sejam perdidas, que  $f_s$  deve ser, no mínimo, igual ou superior do que o dobro da maior frequência presente no sinal (Figura: 2) (LYONS; FUGAL, 2014).

Logo,

$$f_s \geq 2f_{max} \quad (3.6)$$

onde  $f_{max}$  corresponde ao valor da frequência máxima do sinal a ser amostrado.

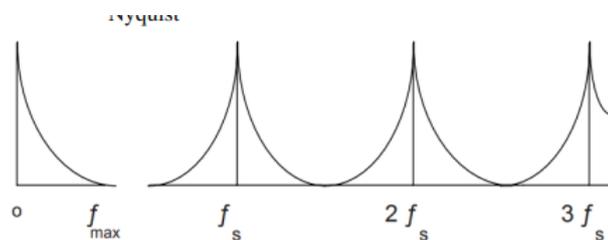


Figura 2 – Espectro de um sinal amostrado a partir do teorema de Nyquist.

Contudo, existe um outro fenômeno responsável por provocar distorções no sinal coletado. O chamado efeito *aliasing*, ocorre quando, ao analisar o espectro, é observado uma sobreposição entre duas réplicas do sinal, o que impossibilita a sua recuperação. Para evitar esse problema, utiliza-se um "filtro *anti-aliasing*".

Um filtro nada mais é do que é um circuito elaborado para permitir a passagem apenas de sinais com o valor de frequência desejado (ALEXANDER, 2013). No caso do filtro

*anti-aliasing*, deve-se atenuar qualquer sinal que apresentar uma  $f_{max}$  igual ou superior a  $(f_s/2)$ .

Logo,

$$f_{max} \leq \frac{f_s}{2} \quad (3.7)$$

Nota-se, então, que ele corresponde a um do tipo passa-baixas e o termo  $(f_s/2)$  é denominado *frequência de corte*. Há três outros tipos: passa-altas (permite a passagem das frequências maiores que a frequência de corte e atenua as que estiverem abaixo dela), passa-faixa (permite a passagem apenas das frequências que estiverem dentro do intervalo definido) e rejeita-faixa (permite a passagem de frequências que estiverem fora de um intervalo estabelecido) (ALEXANDER, 2013). Ademais, é importante salientar que eles podem ser tanto filtros analógicos quanto digitais.

### 3.3.1 Conversor Analógico-Digital

Um conversor analógico-digital (ADC) nada mais é do que um circuito projetado para realizar a mudança de dados analógicos para digitais (Figura: 3). O processo de conversão é resultante, basicamente, da implementação de duas etapas: Amostragem e Quantização. (LANDAU; GIANLUCA, 2006)

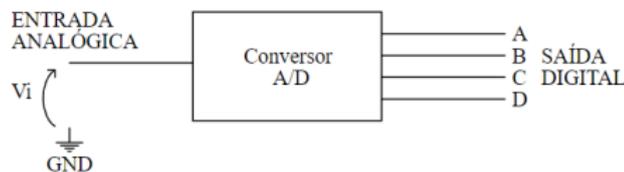


Figura 3 – Diagrama de funcionamento de um ADC.

Na Amostragem, ocorre a coleta de partes do sinal, ou seja, de informações a respeito dele. O parâmetro mais importante nessa fase é a já mencionada  $f_s$ . Isso porque, no projeto de um conversor, é necessário a implementação de um contador (ou *clock*) responsável por gerar um trem de pulsos que, a cada transição, habilita o restante do circuito para realizar as operações necessárias de conversão. A transição de funcionamento escolhida pode ser do tipo positiva (quando o *clock* muda do nível lógico 0 para 1) ou negativa (1 para 0). Entretanto, para que o conversor opere corretamente,  $T = 1/f_s$  não pode ser menor do que o tempo de conversão de uma única amostra (TOCCI; WIDMER; MOSS, 2011).

Por outro lado, na quantização, é feita a codificação desses elementos, ou melhor, os valores coletados são representados por uma sequência de números binários. A precisão da transformação feita no estágio de quantização é determinada tanto pela faixa de amplitude da entrada analógica do conversor em que ele opera de modo linear (por isso, é importante condicionar o sinal de modo que permita seu máximo aproveitamento dentro dessa faixa) quanto pela sua resolução.

A resolução define o menor valor que o módulo é capaz de converter dentro desse intervalo de operação. Ela depende do número de bits da unidade (na Figura 3, essa quantidade é representada pelas letras ABCD). Como o dado é convertido para o sistema binário, o número de possibilidades é da ordem de  $2^n$ , onde  $n$  corresponde ao número de bits (TOCCI; WIDMER; MOSS, 2011). Por exemplo, para um conversor de 10 bits que opera numa faixa de 3 V, tem-se que a resolução é, aproximadamente,

$$\frac{3V}{2^{10} - 1} = 2,93mV \quad (3.8)$$

É importante salientar que, comumente, o termo resolução é utilizado para indicar número de bits do dispositivo. Nota-se que, independente da quantidade de bits, sempre ocorrerá o chamado **erro de quantização**. Esse erro refere-se a diferença entre o valor analógico e o número digital associado a ele. Por outro lado, é possível reduzir esse erro ao aumentar a quantidade de bits do conversor. Entretanto, se há o aumento do número de bits, maior será o tempo necessário para a conversão das informações. A imprecisão do conversor pode ser resultado, também, de irregularidades dos próprios componentes usados para a construção do circuito (TOCCI; WIDMER; MOSS, 2011).

### 3.4 Análise de Fourier

A análise de Fourier é o estudo de como as funções gerais podem ser decompostas em funções trigonométricas ou exponenciais com frequências definidas (MORIN, 2009). Existem dois tipos de expansões de Fourier:

- Série de Fourier: Se uma função é periódica, então ela pode ser escrita como uma soma discreta de funções trigonométricas ou exponenciais com frequências específicas.
- Transformada de Fourier: Uma função geral que não é necessariamente periódica (mas que ainda é razoavelmente bem-comportada) pode ser escrita como uma integral contínua de funções trigonométricas ou exponenciais com um contínuo de frequências possíveis.

Para a aplicação proposta, a Transformada Discreta de Fourier (TDF) é muito útil, uma vez que é possível obter a frequência de sinais colhidos com controladores digitais. Entretanto, o cálculo numérico da transformada exige um grande esforço computacional, o que pode inviabilizar a sua implementação e utilização. Felizmente, foi desenvolvido um algoritmo que reduz consideravelmente a quantidade de operações, o que tornou a TDF acessível. Esse algoritmo é denominado Transformada Rápida de Fourier, ou Fast Fourier Transform (FFT) (LATHI, 2006).

### 3.4.1 Série de Fourier

A razão pela qual a análise de Fourier é tão importante na física se dá pelo fato de muitas equações diferenciais que governam os sistemas físicos serem lineares. Isto implica que a soma de duas soluções é novamente uma solução em si. Portanto, como a análise de Fourier diz que qualquer função pode ser escrita em termos de funções senoidais, pode-se limitar a atenção a essas funções ao resolver as equações diferenciais, podendo-se construir qualquer outra função a partir dessas. (MORIN, 2009)

Considere uma função  $f(x)$  que é periódica no intervalo  $0 \leq x \leq T$ , então o teorema de Fourier afirma que  $f(x)$  pode ser escrita como:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left[ a_n \cos\left(\frac{2\pi nx}{T}\right) + b_n \sin\left(\frac{2\pi nx}{T}\right) \right]$$

onde  $a_n = \frac{2}{T} \int_0^T f(x) \cos\left(\frac{2\pi nx}{T}\right) dx$  e  $b_n = \frac{2}{T} \int_0^T f(x) \sin\left(\frac{2\pi nx}{T}\right) dx$ .

(3.9)

A equação 3.9 é a série trigonométrica de Fourier para a função  $f(x)$ .

Também é possível expressar a equação 3.9 na forma exponencial, da seguinte maneira:

$$f(x) = \sum_{n=-\infty}^{\infty} C_n e^{i2\pi nx/T}, \quad \text{onde} \quad C_n = \frac{1}{T} \int_0^T f(x) e^{-i2\pi nx/T} dx \quad (3.10)$$

### 3.4.2 Transformada de Fourier

Para o caso de  $f(x)$  não ser uma função periódica, ainda é possível escrever a função como uma série envolvendo termos trigonométricos ou exponenciais. Para isso, a série deve ser uma integral contínua em vez de uma soma discreta. (MORIN, 2009)

A vantagem na derivação da integral contínua é que se pode considerar uma função não periódica como uma função periódica no intervalo  $-\frac{T}{2} \leq x \leq \frac{T}{2}$ , com  $T \rightarrow \infty$ . Dessa forma é possível derivar a seguinte equação para a transformada de Fourier:

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \quad (3.11)$$

O termo  $C(k)$  é denominado como a transformada de Fourier da função  $f(x)$  e de maneira geral é utilizada a forma exponencial da série de Fourier para derivar este resultado. (MORIN, 2009)

Para se obter a expressão trigonométrica da transformada de Fourier é necessário separar a função  $f(x)$  em duas funções, uma par e uma ímpar, de forma que  $f(x) = f_p(x) + f_i(x)$  seja respeitado. Estas funções são dadas como:

$$f_p = \frac{f(x) + f(-x)}{2} \text{ e } f_i = \frac{f(x) - f(-x)}{2} \quad (3.12)$$

Assim obtém-se que:

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (f_p(x) + f_i(x))(\cos(-kx) + i\text{sen}(-kx))dx \quad (3.13)$$

A qual também pode ser decomposta em duas equações, uma par e uma ímpar (MORIN, 2009):

$$\begin{aligned} C_p(k) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f_p(x)\cos(kx)dx \\ C_i(k) &= -\frac{i}{2\pi} \int_{-\infty}^{\infty} f_i(x)\text{sen}(kx)dx \end{aligned} \quad (3.14)$$

#### 3.4.2.1 Transformada Discreta de Fourier

Ao se trabalhar com sistemas digitais, muitas vezes, o sinal precisa passar pelo processo de discretização. Felizmente, a série de Fourier pode ser aplicada, também, para o sinal discretizado e digitalizado.

Quando se realiza a operação de conversão do sinal analógico para digital e, em seguida, deseja-se analisar o espectro de frequência dele, nota-se que existem duas etapas de amostragem: uma para a discretização e digitalização do sinal e a outra para o cálculo da transformada de Fourier (LATHI, 2006).

Na primeira,  $T$  corresponde ao tempo em que devem ser coletadas amostras do sinal (Período de Amostragem). Por outro lado, na segunda,  $T_0$  refere-se ao período em que o sinal amostrado deverá ser repetido, ou seja, à janela de tempo total em que informações serão adquiridas do sinal amostrado para o cálculo da transformada (Figura: 4).

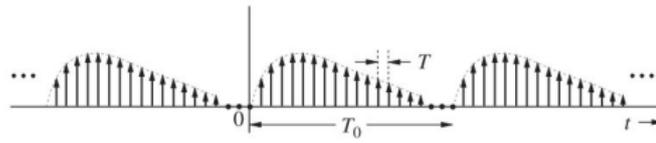


Figura 4 – Sinal Amostrado no Domínio do Tempo. (LATHI, 2006)

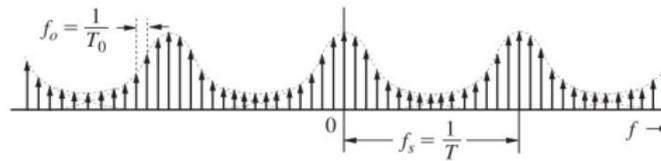


Figura 5 – Espectro de Fourier do Sinal Amostrado. (LATHI, 2006)

Conforme o teorema de amostragem espectral aponta, essa ação resulta na amostragem do espectro a um valor de  $T_0$  amostras/Hz, isso significa que, entre as amostras, há uma separação por  $f_0 = 1/T_0$  (Figura 5).

De acordo com o que é observado nas Figuras 25 e 26, tem-se que

$$N_0 = \frac{T_0}{T} = \frac{f_s}{f_0} \quad (3.15)$$

Assim, as *transformadas Discretas de Fourier* direta ( $F_r$ ) e inversa ( $f_n$ ) são definidas, respectivamente, por

$$F_r = \sum_{n=0}^{N_0-1} x_n e^{-jr\Omega_0 n} \quad (3.16)$$

$$f_n = \frac{1}{N_0} \sum_{r=0}^{N_0-1} X_r e^{jr\Omega_0 n} \quad (3.17)$$

de tal forma que,

$$\Omega_0 = \omega_0 T = \frac{2\pi}{N_0} \quad (3.18)$$

, onde  $\omega_0$  é denominada de frequência angular fundamental. (LATHI, 2006)

Entretanto, é importante salientar que tais definições desconsideram o efeito *aliasing*. Apesar da solução de "filtro *anti-aliasing*" proposta no Item 3.3, o efeito nunca pode ser anulado. Isso se deve ao fato de que se a função  $f(t)$  for limitada no tempo, conseqüentemente, o espectro  $F(\omega)$  não é restrito em faixa. Em contrapartida, se o espectro  $F(\omega)$  de um sinal é delimitado em faixa, ele não seria restrito no tempo, o que provocaria um *aliasing* no domínio do tempo e, assim, a imprecisão das amostras precisaria ser desconsiderada. Deste

modo, independente do método utilizado para a determinação numérica da TDF, direta ou inversa, será possível apenas reduzir o erro. (LATHI, 2006)

Devido aos cálculos computacionais, o sinal  $f(t)$  tem de ser limitado no tempo, porém, ao realizar a truncagem dos dados, há o surgimento de dois fenômenos: espalhamento espectral e vazamento.

No espalhamento, o espectro do sinal se difunde pelo total da largura de faixa do outro sinal com o qual ele realiza a convolução. No caso do vazamento, o espectro  $F(\omega)$  do sinal acaba se espalhando na faixa na qual ele deveria ser zero, conseqüentemente, ele provoca o efeito *aliasing*. Contudo, se um é reduzido, o outro é afetado, visto que, para reduzir o vazamento, uma janela amortecida pode ser usada no processo de truncagem do sinal. Entretanto, isso aumenta o espalhamento, que pode ser reduzido aumentando  $T_0$  (logo, adquirindo mais dados), o que reduz  $f_0$ , provocando o aumento da *resolução em frequência* ou *resolução espectral*. Em face disso, é preciso selecionar uma janela suave de largura suficiente (LATHI, 2006).

Por outro lado, dependendo da distância entre as amostras, pode ocorrer o chamado *efeito de cerca de postes*. Nele, alguns comportamentos do sinal podem passar despercebidos devido a esse intervalo de tempo em que não houve coleta de amostras. Isso provoca um ilusório retrato da realidade. Esses resultados incorretos podem ser prevenidos usando uma quantidade de amostras  $N_0$  suficientemente grande, o que resulta no aumento da resolução (LATHI, 2006).

#### 3.4.2.2 Transformada Rápida de Fourier

O cálculo numérico da TDF exige uma série de operações matemáticas de somas e multiplicações complexas. Esses cálculos, dependendo da quantidade de amostras  $N_0$ , podem exigir muito tempo para serem realizados, limitando até que alguns computadores possam fazer essa operação.

Em face disso, Cooley e Tukey desenvolveram, em 1965, um algoritmo capaz de reduzir radicalmente essa quantidade de cálculos, permitindo que a transformada de Fourier se tornasse realizável para o processamento digital de sinais. A *Fast Fourier Transform*, ou FFT, diminui o número de operações da ordem de  $N_0^2$  para  $N_0 \log N_0$  (LATHI, 2006).

Isso foi possível graças a linearidade apresentada pela TDF, que permitiu que a transformada de Fourier de um sinal pudesse ser calculada como a soma de transformadas de Fourier de partes do sinal com duração menor. Por exemplo, considerando que um sinal apresenta  $N_0 = 4$  amostras, é possível dividir essa sequência em outras menores de tamanho 2. Logo, para realizar o cálculo da TDF de cada um dos segmentos, são necessárias 4 multiplicações e 2 adições, visto que a TDF de um sinal com  $N_0$  amostras exige  $N_0^2$  multiplicações complexas e  $N_0(N_0 - 1)$  somas complexas.

Caso contrário, seria preciso realizar 16 multiplicações e 12 adições. Os valores da FFT são os mesmos aos obtidos pela TDF. Ademais, o algoritmo pode ser simplificado ainda mais se  $N_0$  for uma potência de 2 ([LATHI, 2006](#)).

## 4 Desenvolvimento do Sistema

Neste capítulo será tratada a solução implementada deste projeto, levando como base outras soluções apresentadas, tratadas anteriormente na revisão bibliográfica, além dos conceitos vistos na fundamentação teórica. Com isso será proposto um dispositivo que seja capaz de coletar dados em pacientes da Doença de Parkinson da melhor maneira possível e, ademais, definir seus requisitos.

### 4.1 Requisitos

Considerando os objetivos apresentados para este projeto e o envolvimento de pessoas que já possam sentir desconforto em seu cotidiano em decorrência da Doença de Parkinson, o dispositivo desenvolvido cumpre os seguintes requisitos. Este deve ser capaz de medir os tremores apresentados pelos pacientes portadores da doença, além de apresentar um formato pequeno o suficiente para não impor desconforto durante a colheita de dados.

O aparato deve também realizar as medições de forma rápida e indolor, de maneira que seja respeitada a privacidade do paciente. Por fim, o dispositivo deve ser capaz de obter as frequências fundamentais e amplitude média de aceleração dos movimentos.

### 4.2 Desenvolvimento do Sistema Embarcado

O hardware proposto possui dois elementos principais que possibilitam seu funcionamento: uma placa de controle para comandar as operações realizadas e um sensor para a coleta de dados.

#### 4.2.1 Placa Controladora

A princípio, este projeto foi idealizado com o intuito de se utilizar uma placa Arduino Uno para realizar o controle de sensores e tratamento de dado. Todavia, ao se desenvolver o código do projeto, observou-se que o Arduino não possuía capacidade de processamento e nem memória suficiente para executar as tarefas necessárias.

Dessa forma, buscou-se uma alternativa com um poder de processamento maior, optando-se pela placa WiFi LoRa32 V2 da Heltec Automation, uma vez que a mesma já estava disponível, evitando, assim, novos gastos.

#### 4.2.1.1 WiFi LoRa32 V2

Para realizar a implementação da FFT e, assim, conseguir analisar as informações no domínio da frequência, foi utilizado um microcontrolador da série ESP32, especificamente, o modelo WiFi LoRa32. Dentre as vantagens de utilizar esta série de microcontrolador estão a portabilidade, seu baixo consumo de energia, facilidade de uso, variedade de funcionalidades, entre outras. Ademais, a linguagem utilizada para programação é C/C++, além de possuir integração com o Integrated Development Environment (IDE) da plataforma Arduino.

O Microcontrolador, ou *Microcontroller Unit* MCU, usado WiFi LoRa 32 é uma placa de desenvolvimento IoT clássica projetada e produzida pela Heltec Automation, é um produto altamente integrado baseado no ESP32 + SX127x, possui funções Wi-Fi, também possui um sistema de gerenciamento de bateria de Lítio, além de uma tela OLED(*Organic LED*) de 0,96 polegadas inclusa. Apesar de serem utilizadas neste projeto, a placa ainda possui funções de comunicação BLE(*Bluetooth Low Energy*) e LoRa(*Long Range*).



Figura 6 – Microcontrolador WiFi LoRa 32.(HELTEC, 2018)

Já em relação a interface dos seus periféricos, dentre as principais características, possuem até 18 canais analógicos com ADCs (*Analogic-Digital Converter*) de aproximações sucessiva, ou *Successive-approximation-register* (SAR) ADCs com até 12 bits de resolução e que realizam a leitura de níveis de tensão variados dentro da faixa de 0 V a 3,3 V, além de possuir 2 barramentos do tipo *Inter-Integrated Circuit* (I2C) e 34 *General Purpose Input/Output* (GPIO)s programáveis descritos na Figura: 7.

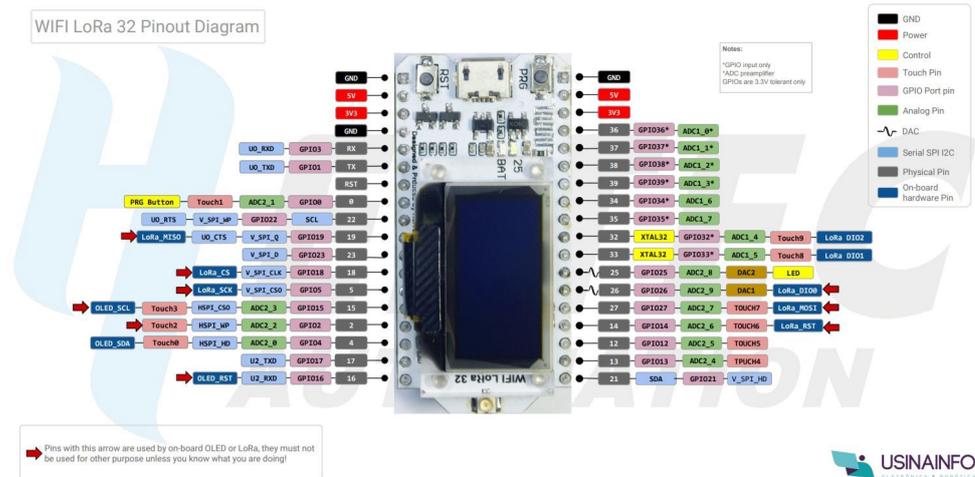


Figura 7 – Diagrama de Pinos do WiFi LoRa 32.(HELTEC, 2018)

#### 4.2.2 Acelerômetro e Giroscópio MPU6050

O sensor que realizará a aquisição dos dados será uma unidade de medição inercial, em inglês "*Inertial Measurement Unit (IMU)*" do modelo MPU6050 da fabricante InvenSense, observado na Figura 8.

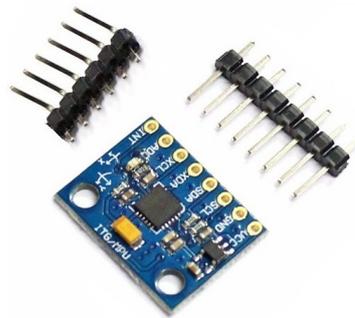


Figura 8 – MPU6050.  
(HU, 2019)

O MPU6050 é um sensor representativo de MEMS (Micro Electro Mechanical Systems) que são sistemas ou dispositivos muito pequenos, compostos por microcomponentes que variam de 0,001 mm a 0,1 mm de tamanho. Esses componentes são feitos de silício, polímeros, metais e/ou cerâmicas, e geralmente são combinados com uma CPU (Microcontrolador) para completar o sistema.

O MPU6050 mede a aceleração medindo a mudança na capacitância. Sua microestrutura se parece com a Figura 9, tendo uma massa presa a uma mola que se limita a se mover em uma direção e placas externas fixas. Portanto, quando uma aceleração na direção específica for aplicada, a massa se moverá e a capacitância entre as placas e a massa mudará. Esta mudança na capacitância será medida, processada e corresponderá a um determinado

valor de aceleração.

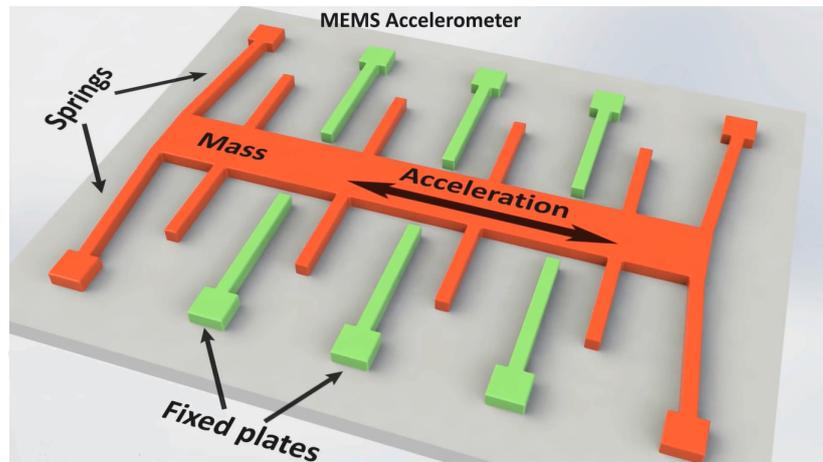


Figura 9 – Acelerômetro.(NEDELKOVSKI, 2018)

O giroscópio mede a taxa angular usando o Efeito Coriolis. Quando uma massa está se movendo em uma determinada direção com uma determinada velocidade e quando uma taxa angular externa for aplicada ocorrerá uma força que causará um deslocamento perpendicular da massa. Assim semelhante ao acelerômetro, esse deslocamento causará mudança na capacitância que será medida, processada e corresponderá a uma determinada taxa angular. A microestrutura do giroscópio está representada na Figura 10. Uma massa que está em constante movimento, ou oscilando, e quando a taxa angular externa for aplicada uma parte flexível da massa se moveria e faria o deslocamento perpendicular. (NEDELKOVSKI, 2018)

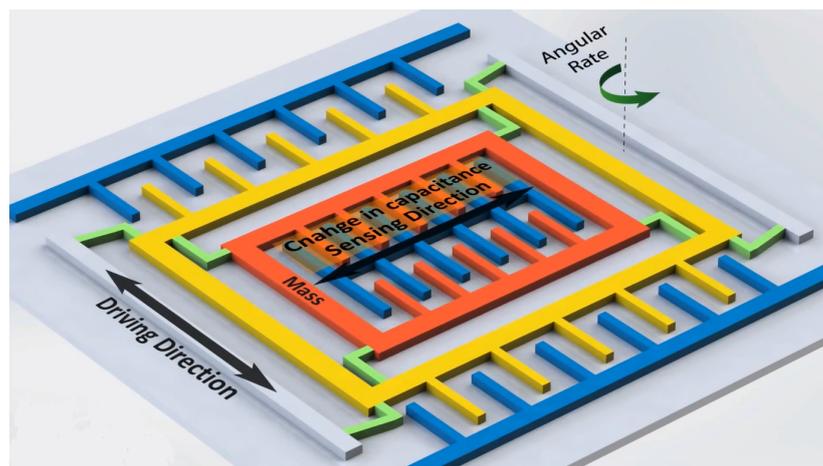


Figura 10 – Giroscópio.(NEDELKOVSKI, 2018)

O sensor MPU6050 também possui uma variedade de funções benéficas à aplicação proposta. Dentre as quais, o sensor utiliza o protocolo de comunicação serial I2C e possui um registrador FIFO de 1024 bytes que é acessível através da Interface Serial, tornando possível o armazenamento de dados do sensor. O FIFO armazena em buffer o conjunto de dados completo, reduzindo os requisitos de tempo no processador do sistema. Dessa forma,

impede-se que a comunicação serial atrapalhe a coleta constante de dados, permitindo que a frequência de amostragem permaneça constante.

O sensor possui ainda um filtro passa-baixas digital programável (*Digital Low Pass Filter* - DLPF) que impede que ocorra o efeito *aliasing* descrito em 3.3. O sensor permite um que o filtro seja ajustado de acordo com a Tabela 2.

DLPF_CFG	Accelerometer (Fs = 1kHz)		Gyroscope		
	Bandwidth (Hz)	Delay (ms)	Bandwidth (Hz)	Delay (ms)	Fs (kHz)
0	260	0	256	0.98	8
1	184	2	188	1.9	1
2	94	3	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19	5	18.6	1
7	RESERVED		RESERVED		8

Tabela 2 – Configuração de Filtro Passa-baixas Digital. (INVENSENSE, 2013)

Outra função muito útil do sensor é a existência de um *Digital Motion Processor* (DMP) que descarrega do MCU os requisitos de computação intensivos de processamento de movimento, minimizando a necessidade de sondagem frequente da saída do sensor de movimento. Este algoritmo unifica os dados do acelerômetro e giroscópio e calcula dados relacionados ao movimento. Este vem embarcado de fábrica no sensor e seu código fonte é inacessível, fazendo-se disponível apenas as funções utilizáveis de sua biblioteca no site do fabricante.

O acelerômetro do sensor possui ainda diferentes níveis de escala programáveis, que podem ser observadas na Tabela 3. Vale ressaltar que cada valor de escala possui uma sensibilidade própria, expressada em LSB/g (*least significant bit per g (gravity)*).

AFS_SEL	Full Scale Range	LSB Sensitivity
0	± 2g	16384 LSB/g
1	±4g	8192 LSB/g
2	±8g	4096 LSB/g
3	±16g	2048 LSB/g

Tabela 3 – Sensitividade do Acelerômetro pelo bit menos significativo(LSB). (INVENSENSE, 2013)

Em um estágio inicial do projeto, foram realizados testes com o ADXL345, que é um sensor acelerômetro similar ao MPU6050, porém com menos funções. Além de não possuir um giroscópio e um sensor de temperatura, esse sensor também não possui um DMP e seu filtro digital interno não é programável. Essas desvantagens favoreceram a escolha do MPU6050 para este projeto.

Além disso, optou-se pelo MPU6050, pois este é um dispositivo amplamente utilizado na comunidade científica e entusiasta de projetos mecatrônicos, possuindo uma larga biblioteca de projetos que podem ser utilizados para auxiliar neste. Por fim, o aparato possui resolução e recursos mais que suficientes para cumprir com o que é proposto neste trabalho.

#### 4.2.3 Alimentação de Energia

O sistema de alimentação de energia é composto por uma bateria de Polímeros de Lítio de 3.7V com capacidade de 950mAh conectada diretamente à placa WiFi LoRa32 através de um conector JST 2P 1.25, uma vez que a placa possui um sistema próprio de gerenciamento e carregamento de baterias para essa especificação, bastando apenas conectar a placa via USB para carregar a bateria.

Uma vez que de a placa WiFi LoRa32 apresenta um consumo de 115mA com o servidor web em funcionamento e o sensor MPU6050 um consumo de 3,9mA, além de que a bateria utilizada possui uma capacidade de 950mAh, estima-se que o tempo de funcionamento autônomo da pulseira seja cerca de 8 horas.

## 4.3 Software

A placa WiFi LoRa32 é compatível com variadas plataformas de desenvolvimento e linguagens de programação, tornando-a ainda mais acessível.

Para a elaboração do trabalho em questão, a plataforma de desenvolvimento escolhida foi a Arduino IDE, desenvolvida pela empresa Arduino. Esse software de código aberto possui uma estrutura bem intuitiva e de fácil compreensão, facilitando a elaboração de códigos e o carregamento deles para a placa utilizada. Ademais, o código principal foi desenvolvido na linguagem de programação C, utilizando bibliotecas, predominantemente, escritas na linguagem C++. Vale ressaltar que, para conseguir carregar o código para a placa ESP32, é preciso instalar o driver da placa na plataforma Arduino IDE.

A priori, para fins de verificação da qualidade da aquisição de dados, criou-se um servidor web com própria placa controladora para coletar os dados numéricos adquiridos no tempo de amostragem definido. O endereço IP a ser acessado é mostrado constantemente durante o uso do dispositivo na última linha do *display* OLED.

Buscando a simplificação do código, foi empregada uma biblioteca para a implementação da FFT denominada *arduinoFFT*. Como ela não faz parte do pacote padrão do software, é preciso instalá-la. A vantagem é que, além de realizar o cálculo numérico da FFT, ela possui outras operações, dentre as quais, estão a remoção do nível contínuo do sinal, variadas opções para janelamento e realiza o cálculo da magnitude através dos coeficientes complexos resultantes da FFT. Optou-se por utilizar a transformada rápida de Fourier neste projeto, uma vez que a mesma é capaz de filtrar sinais ruidosos e obter um valor acurado da frequência fundamental do sistema em questão. (CONDES, 2020)

Como, ao fim do cálculo da transformada, é desejável visualizar o valor em frequência onde ocorreu o maior valor em magnitude, foi utilizado *Display* OLED de 0,96 polegadas incluso na placa. Ele possui uma tensão de operação na faixa de 2,2 V até 5 V, e seus *pixels* são controlados de modo individual via I2C com o chip controlador SSD1306 já embutido no LoRa32.

### 4.3.1 Arquitetura da Solução

O código desenvolvido pode ser dividido em etapas, as quais estão representadas no fluxograma da Figura 11. Inicialmente, há a definição dos parâmetros utilizados (bloco Definição dos parâmetros), dentre os quais estão  $N_0$ ,  $f_s$  e  $T$  ajustados de acordo com o sensor utilizado.

A função de *Setup* começa no bloco *Inicia Servidor Web*, onde é inicializado o servidor web e disponibilizado no *display* OLED o endereço IP. Logo em seguida, ainda nessa função inicial, é feita a calibração do sensor MPU6050, usando-se os dados estabelecidos na definição

de parâmetros previamente: valores de escala de aceleração e banda limite para o filtro passa-baixas digital.

A calibração é enfim realizada ao se calcular o *offset* das leituras, através da realização da média de duas mil medições e em seguida a divisão pelo valor da sensibilidade do acelerômetro. É relevante ressaltar que essa metodologia de calibração foi obtida diretamente da empresa desenvolvedora do sensor MPU6050, através de algoritmos de exemplo disponibilizados na biblioteca do sensor.

A função *Loop principal* é utilizada para que o código possa rodar constantemente, permitindo que as operações de processamento do sinal não sejam realizadas uma única vez. No bloco *Fim do intervalo de leitura*, é feita a verificação se o intervalo de coleta de dados programada foi finalizado.

No caso de o algoritmo se encontrar no intervalo, ele lê os valores de aceleração três eixos do sensor, sem tratamento de movimentação do DMP, e os unifica através de uma soma vetorial, de acordo com a equação 4.1. Nesta etapa, também é realizada uma média dos valores de amplitude medidos no acelerômetro para se disponibilizar no final.

$$\text{Resultante} = \sqrt{x^2 + y^2 + z^2} \quad (4.1)$$

Logo em seguida, armazenam-se os dados colhidos em um *array*, sendo importante salientar que, nesse momento, não apenas 1, mas 2 vetores armazenam informações: um com valores reais e outro com valores imaginários. Contudo, o sinal é armazenado apenas no vetor de valores reais, enquanto o vetor de números imaginários é preenchido com valores nulos. A coleta de sinais é também distribuída de maneira cronologicamente igual durante esta etapa. Vale salientar que as primeiras 100 medições são ignoradas, para evitar perturbações relacionadas ao manuseio inicial da pulseira.

Ao ser finalizado o intervalo de leitura, o algoritmo passa para o bloco *Calcula FFT e exibe Frequência*, onde finalmente é realizado o cálculo numérico da FFT. Ocorre respectivamente o janelamento das amostras, a computação da transformada rápida de Fourier e a transformação dos valores coletados em valores de magnitude. Finalmente, a frequência com maior valor de magnitude é exibida, a qual é a fundamental procurada, juntamente com a média das amplitudes de aceleração encontradas. Por fim, caso seja de interesse, ao se apertar o botão a leitura é realizada novamente.

É importante salientar que, independente da abordagem utilizada para a aquisição de dados, sempre haverá certa imprecisão por parte da conversão AD no sensor, em decorrência do erro de quantização, como explicado em 3.3.1. Porém, ela não interfere de maneira significativa na informação adquirida e, com isso, pode ser relevada, uma vez que são utilizados 2048 bits do conversor do sensor MPU6050 para a sensibilidade selecionada de acordo com a tabela 3 na seção 4.2.2.

O código completo do algoritmo desenvolvido está disposto no Apêndice A.

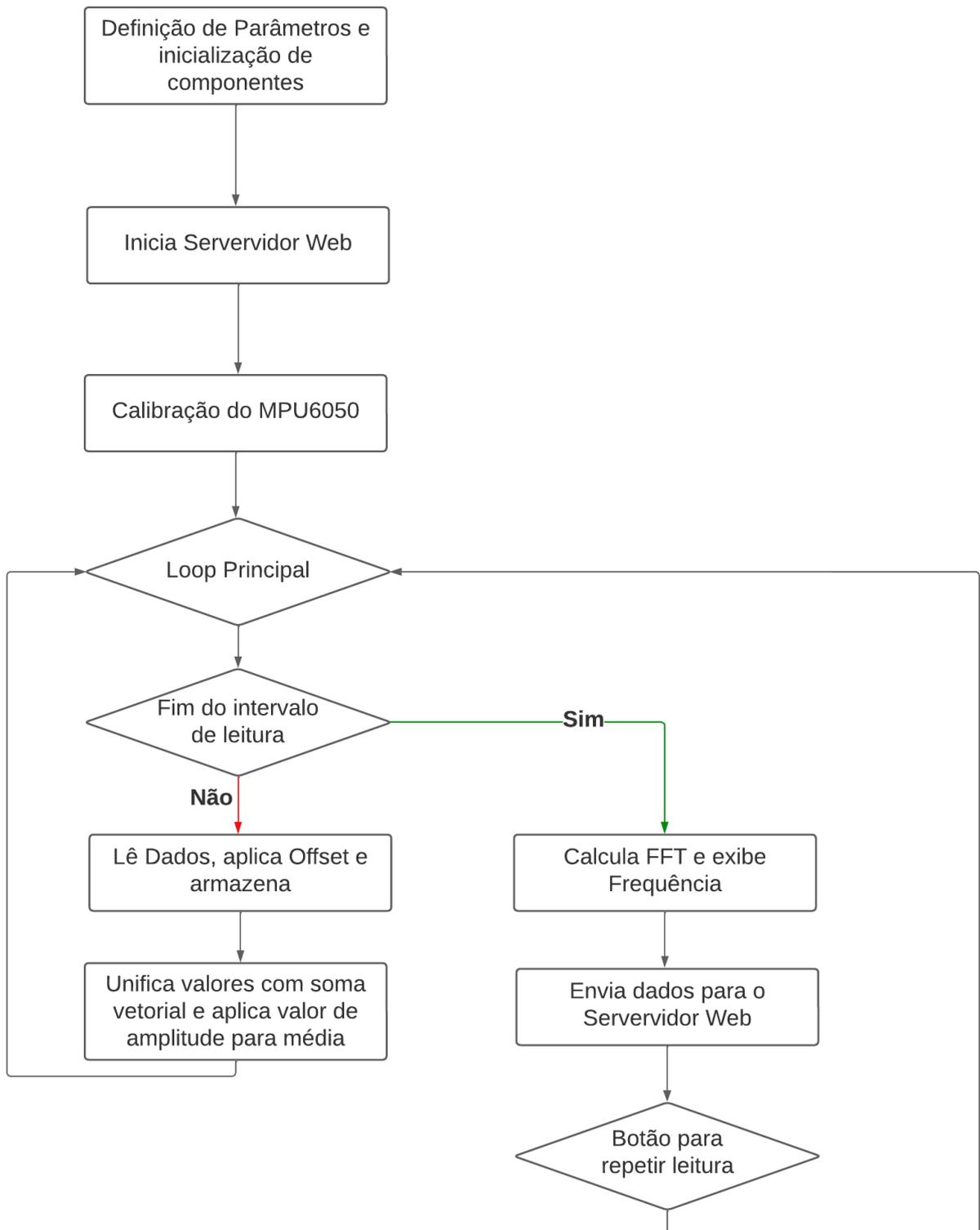


Figura 11 – Fluxograma do código implementado.

## 4.4 Capa de Proteção

Para a proteção do dispositivo foi projetada um encapsulamento que seja acoplável ao pulso do usuário, facilitando também a aquisição da frequência dos tremores. O corpo do revestimento foi projetado na plataforma *Fusion 360* da empresa *Autodesk* e a Figura 12 demonstra o objeto virtual tridimensional desenhado.

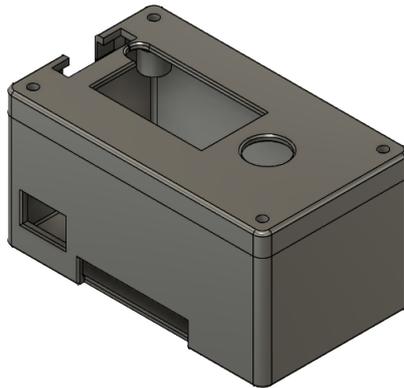


Figura 12 – Projeto Tridimensional da Pulseira de Medição de Frequência.

Em seguida foi realizada a impressão 3D do objeto na Mecajun(Empresa Júnior de Mecatrônica da UnB), que pode ser observado na Figura13.

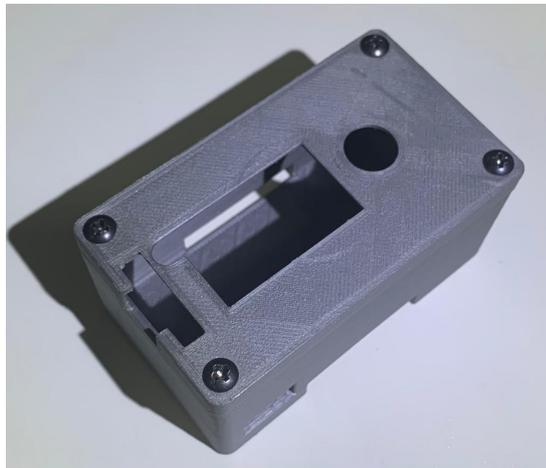


Figura 13 – Impressão do corpo da Pulseira de Medição de Frequência.

Por fim, foi realizada a pintura e adicionada uma pulseira de nylon costurada a uma fita de velcro para prender a capa ao pulso do usuário. O resultado pode ser observado na Figura 14.



Figura 14 – Pulseira de Medição de Frequência de Tremores Finalizada.

## 4.5 Diagrama de Funcionamento do Sistema

A Figura 15 ilustra o diagrama de funcionamento do dispositivo proposto nesse trabalho. Além disso, inclui o funcionamento da bancada de testes desenvolvida para testar o aparato.

Sistema de medição de Frequência de Tremores e Sistema de Teste de Frequência

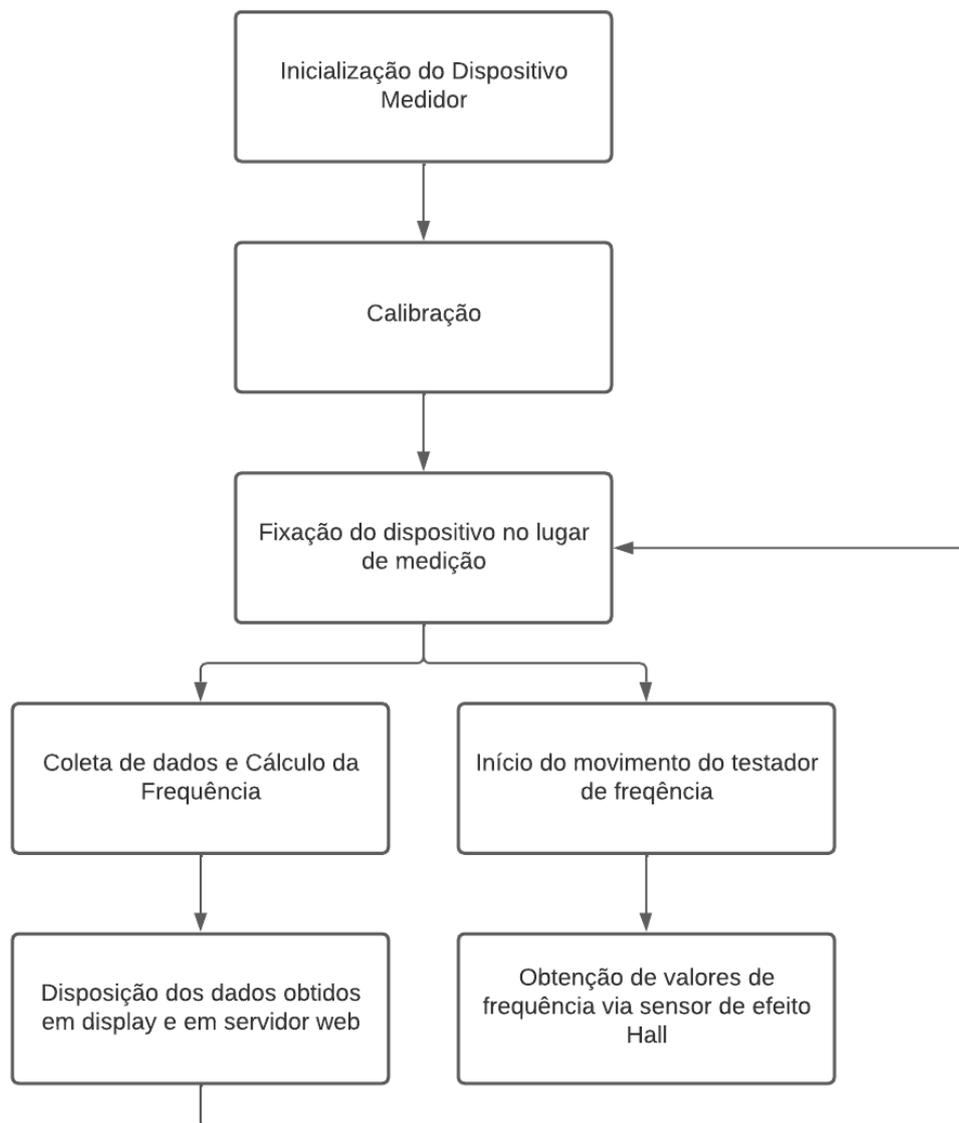


Figura 15 – Diagrama de Funcionamento do Sistema.

## 5 Testes e Resultados

Neste capítulo será exposta a implementação da sistema proposto no capítulo 4, objetivando a análise de seu funcionamento e qualidade de dados.

### 5.1 Testes Realizados

#### 5.1.1 Bancada de Testes

Com o intuito de testar o correto funcionamento do dispositivo medidor de frequências e os códigos desenvolvidos, foi projetado e construído pelo aluno um aparato que realiza movimentos repetitivos, com uma frequência conhecida. O dispositivo de teste é composto por um motor DC com escovas, um regulador de tensão, uma fonte 9V, uma polia, um braço pivotante e uma estrutura de madeira, podendo ser observado na Figura 16 acoplado ao dispositivo de medição sem capa protetora.



Figura 16 – Testador de Frequência Fixa.

A peça foi completamente construída no GRACO (Grupo de Automação e Controle) da Faculdade de Tecnologia da Universidade de Brasília. A carcaça de suporte foi construída

com madeira aproveitada da prefeitura de UnB. O motor DC é aparafusado diretamente à carcaça e acoplado, por uma peça de plástico, ao disco, a qual também é acoplada ao braço pendulante em um pivô composto por um parafuso, porca e espaçador de plástico. Por fim, o braço foi aparafusado à carcaça com um espaçador de plástico no ponto pivotante. Todas as peças de plástico foram produzidas no torno da oficina. Com essa montagem, quando o motor e o disco giram, o pivô corre no rasgo do braço pivotante causando um movimento pendular em suas extremidades.

Com o intuito de se verificar a precisão da medição de frequência da pulseira medidora, fixou-se no dispositivo testador um sensor de efeito Hall A3144 (Figura 18) e sob a superfície do braço pivotante, foi fixado um ímã de Neodímico para atuar no sensor de efeito Hall na extremidade do movimento pendular. Dessa forma, quando ambos se aproximam fisicamente o sensor é excitado e pode-se colher o tempo que o durou o movimento completo do braço pivotante. Tomando a Figura 17 como referência, pode-se perceber que o sensor de efeito Hall somente é excitado quando o braço está na posição do lado esquerdo.

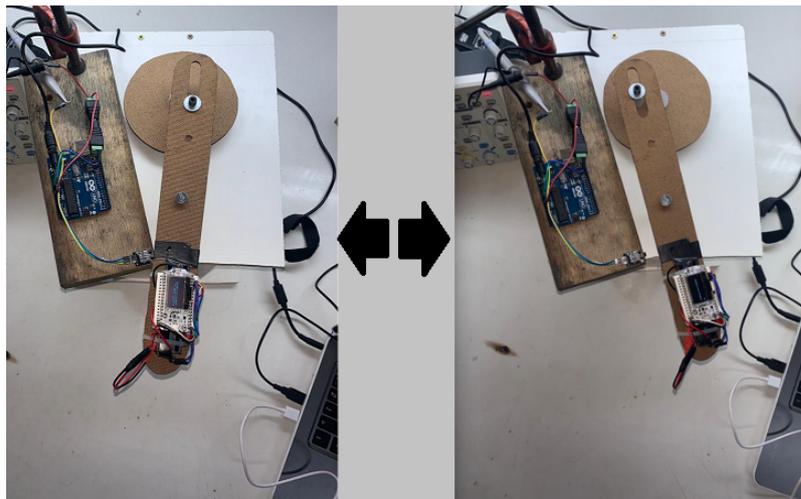


Figura 17 – Ilustração de movimento.

O pino de sinal digital do sensor (mais próximo à letra "s" impressa, Figura 18) foi, em seguida, conectado a um osciloscópio da marca *OWON* modelo SDS1102 e comparado ao pino terra (central) do sensor. O pino da outra extremidade é a alimentação do sensor, onde se conecta 5V. Quando o sensor é excitado pelo ímã, sua saída digital muda de 0V para 5V. Assim, foi possível medir o período ( $T$ ) e, conseqüentemente, a frequência do movimento do braço pivotante ( $f = 1/T$ ), cujo cálculo foi feito diretamente no osciloscópio.

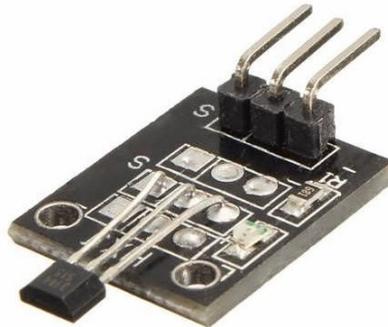


Figura 18 – Sensor de Efeito Hall A3144.

Dessa forma, a bancada estava pronta para iniciar os testes, bastando apenas ligar a pulseira e aguardar sua calibração para iniciar o movimento. O dispositivo de teste é capaz de gerar movimentos com frequências em torno de 2 a 4 Hz e apresenta uma pequena amplitude, o que o torna compatível com as frequências apresentadas por portadores da Doença de Parkinson.

O movimento do braço pivotante pode ser observado no [vídeo](#), cujo [link](#) pode ser acessado pelo código ilustrado na Figura 34 presente no Apêndice B.

Por fim, é possível observar na Figura 19 como ficou a bancada de teste em seu estágio final, cujos resultados serão expostos mais adiante neste documento.

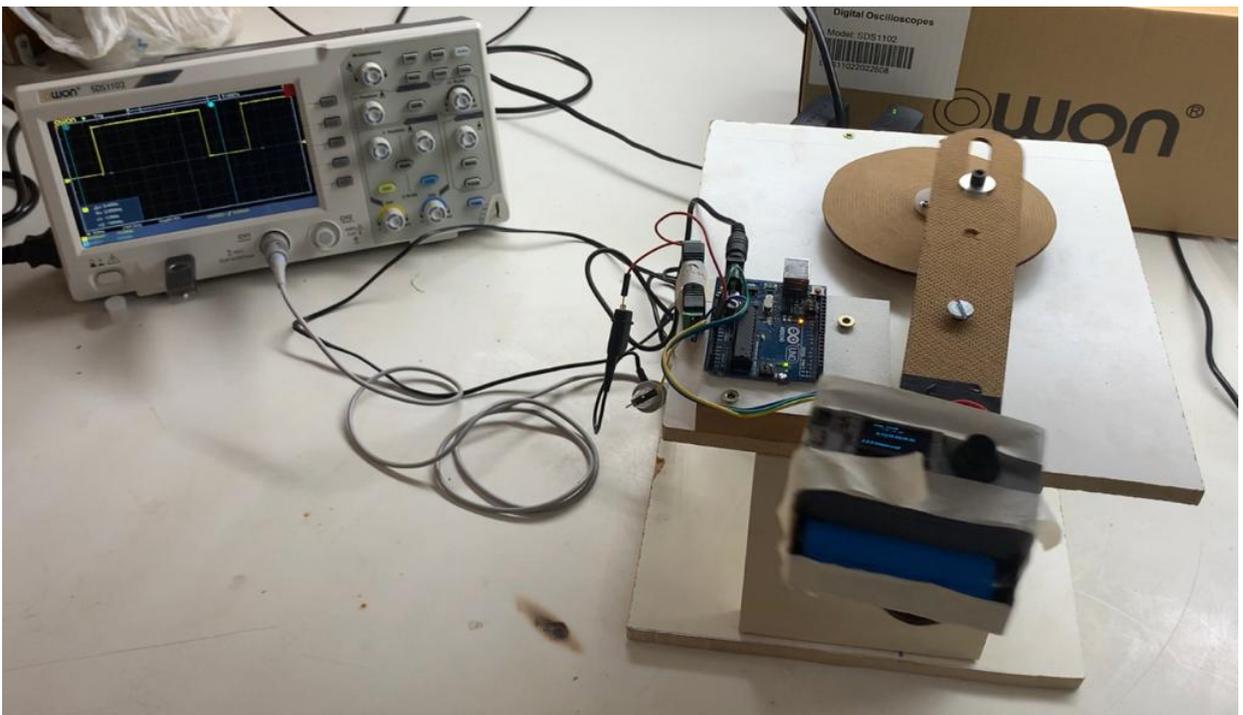


Figura 19 – Bancada de Teste completa.

### 5.1.2 Testes Realizados

Após a montagem da bancada de testes, foram realizados dois testes, um com o regulador de tensão de entrada do motor ajustado para 5V e o outro para 7V, aumentando, assim, sua velocidade. Foram medidas as frequências de três formas: a primeira pelo aparato desenvolvido neste trabalho, após sua calibração, e a segunda através do sinal colhido pelo sensor de efeito Hall A3144 e capturado no osciloscópio, como previamente explicado em 5.1.1. Para a terceira forma de medição de frequência, os dados do acelerômetro foram colhidos através da porta serial do LoRa32, com a ferramenta monitor serial da IDE do Arduino, e, com esses dados, foi feita a FFT com o Software MATLAB (Matrix Laboratory)((MATLAB..., s.d.)) da companhia *MathWorks Inc.* Isso foi feito com o intuito de se verificar o funcionamento da transformada implementada na própria pulseira, através da biblioteca *arduinoFFT*. O script utilizado para realizar os cálculos no MATLAB está descrita a seguir.

Código 5.1 – Código de Matlab

```

1 Fs = size(data,1)/10      % Sampling frequency
2 T = 1/Fs;                % Sampling period
3 L = 10000;               % Length of signal
4 t = (0:L-1)*T;          % Time vector
5
6 f = Fs*(0:(L/2))/L;
7 Y = fft(data);
8 P2 = abs(Y/L);
9 P1 = P2(1:L/2+1);
10 P1(2:end-1) = 2*P1(2:end-1);
11
12 plot(f,P1)
13 title('FFT')
14 xlabel('f (Hz)')
15 ylabel('|P1(f)|')
```

Em seguida, foram realizados testes para a verificação do funcionamento individual de cada eixo de medição do osciloscópio. Para isso, a pulseira foi acoplada ao braço pivotante, de modo que o eixo a ser testado estivesse perpendicular ao movimento, utilizando a Figura 20, obtida no *datasheet* do MPU6050, como referência para o correto posicionamento.

Para que a verificação de cada eixo fosse realizada, alterou-se o algoritmo para que fossem aplicados à FFT apenas os valores lidos do eixo desejado. Dessa forma, o resultado da FFT foi focado em cada eixo por vez, fazendo-se, também, o ajuste na orientação da pulseira para que o movimento verificado fosse o predominante no braço pivotante, seguindo a orientação demarcada na Figura 21.

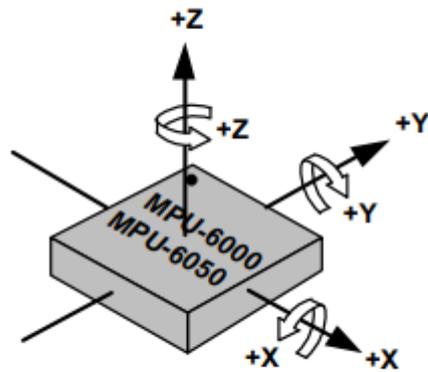


Figura 20 – Eixos de Orientação do MPU6050

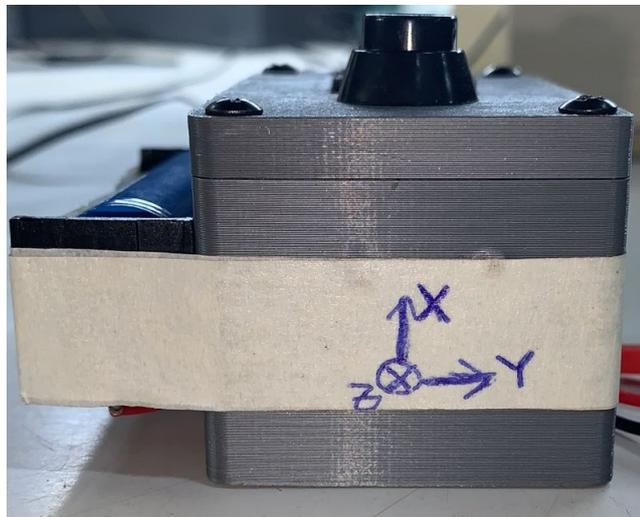


Figura 21 – Eixos de Orientação da Pulseira

Por fim, este último teste teve o intuito de validar o correto funcionamento do dispositivo, para qualquer movimento tridimensional que o mesmo possa ser submetido.

## 5.2 Resultados

Os testes a seguir seguiram a explicação dada em 5.1.2.

### 5.2.1 Resultados do Teste com Motor alimentado a 5V

O primeiro teste foi realizado aplicando-se 5V nos terminais do motor de corrente direta e gerando um movimento com frequência fixa. A Figura 22 demonstra o valor encontrado pela pulseira desenvolvida, que foi de  $11,33 \text{ m/s}^2$  para a amplitude em aceleração e  $2,39\text{Hz}$  para a frequência fundamental do movimento.

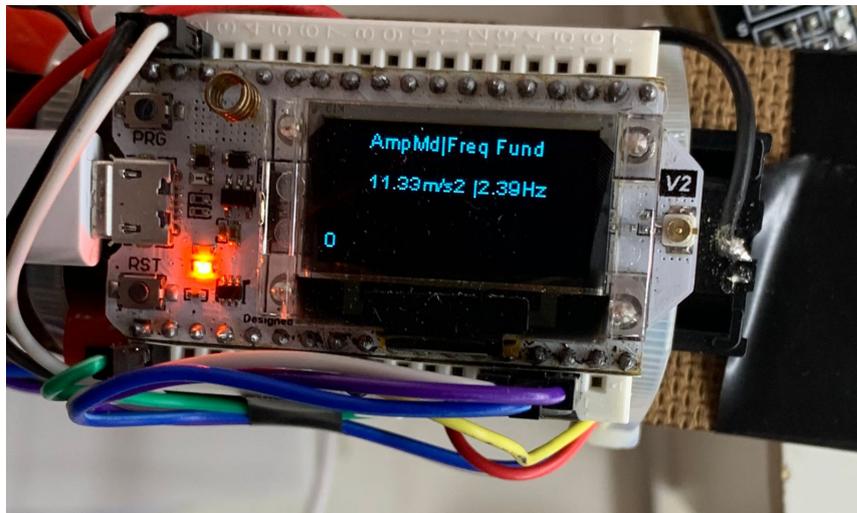


Figura 22 – Leitura da Pulseira para Motor em 5V.

A Figura 23 demonstra a tela do osciloscópio capturada com o sinal colhido pelo sensor de efeito Hall A3144, que resultou em uma frequência observada de  $2,381 \text{ Hz}$ .

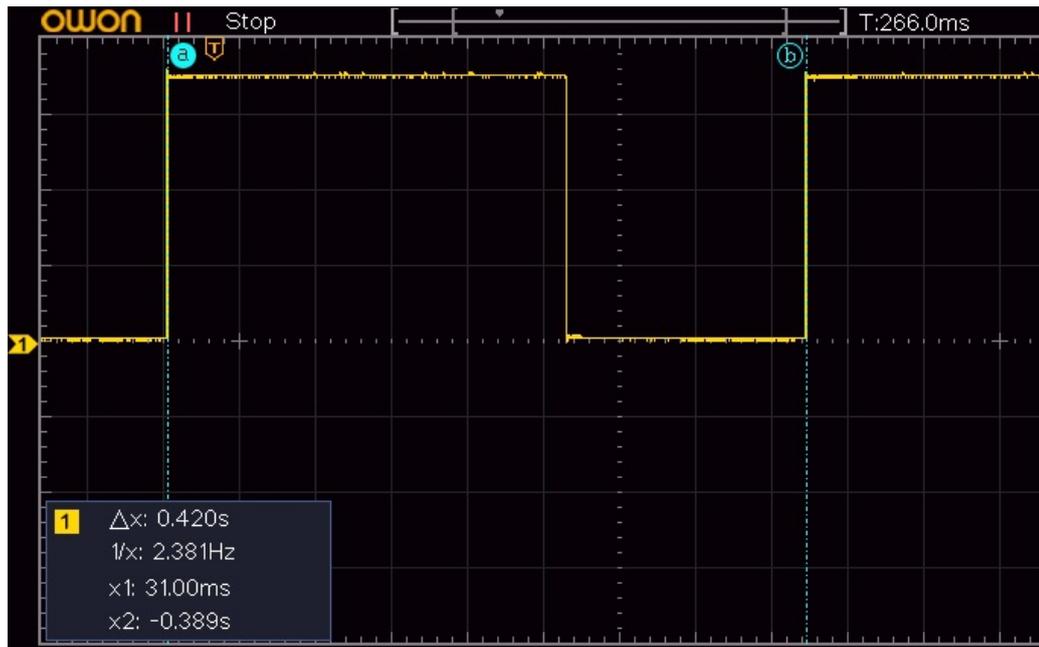


Figura 23 – Leitura do Sensor de efeito Hall vista no osciloscópio para entrada de 5V.

A Figura 24 demonstra o resultado do cálculo da FFT no MATLAB para entrada em 5V, que apresentou uma frequência fundamental de 2,45Hz.

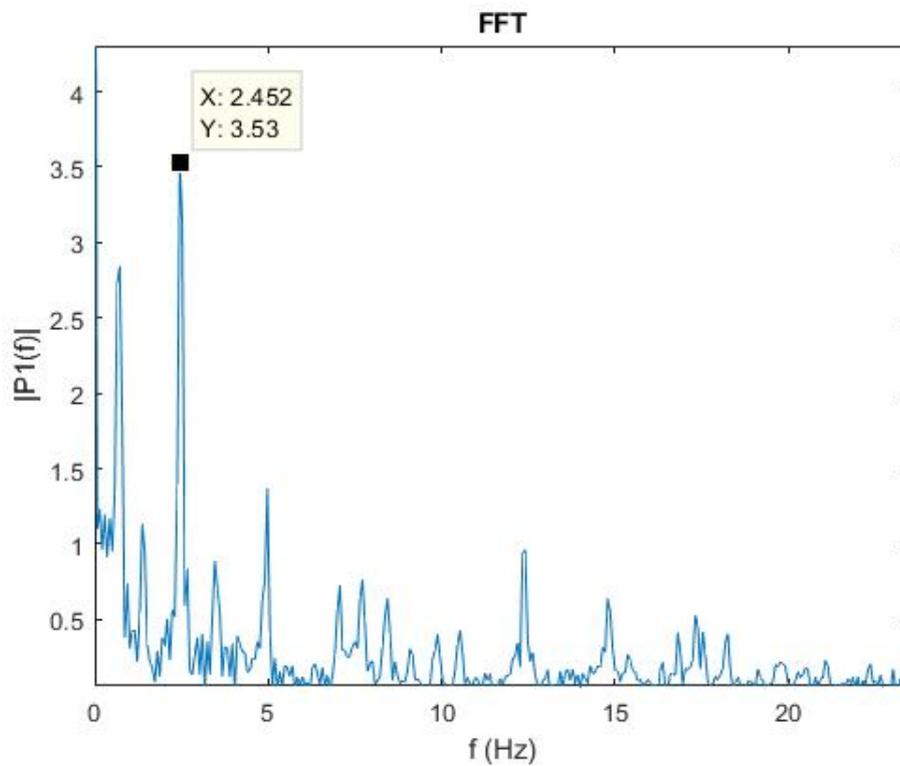


Figura 24 – Resultado do cálculo da FFT no MATLAB para entrada em 5V.

### 5.2.2 Resultados do Teste com Motor alimentado a 7V

O segundo teste foi realizado aplicando-se 7V nos terminais do motor de corrente direta e gerando um movimento com frequência fixa maior que o teste anterior. A Figura 25 demonstra o valor encontrado pela pulseira desenvolvida, que foi de  $14,55m/s^2$  para a amplitude média de aceleração e 3,67Hz para a frequência fundamental do movimento.

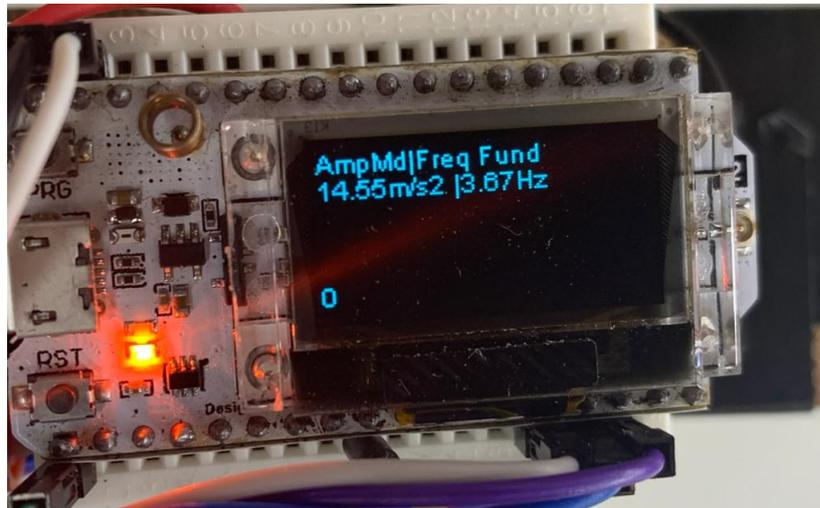


Figura 25 – Leitura da Pulseira para Motor em 7V.

A Figura 26 demonstra a tela do osciloscópio capturada com o sinal colhido pelo sensor de efeito Hall A3144, que resultou em uma frequência observada de 3,666 Hz.

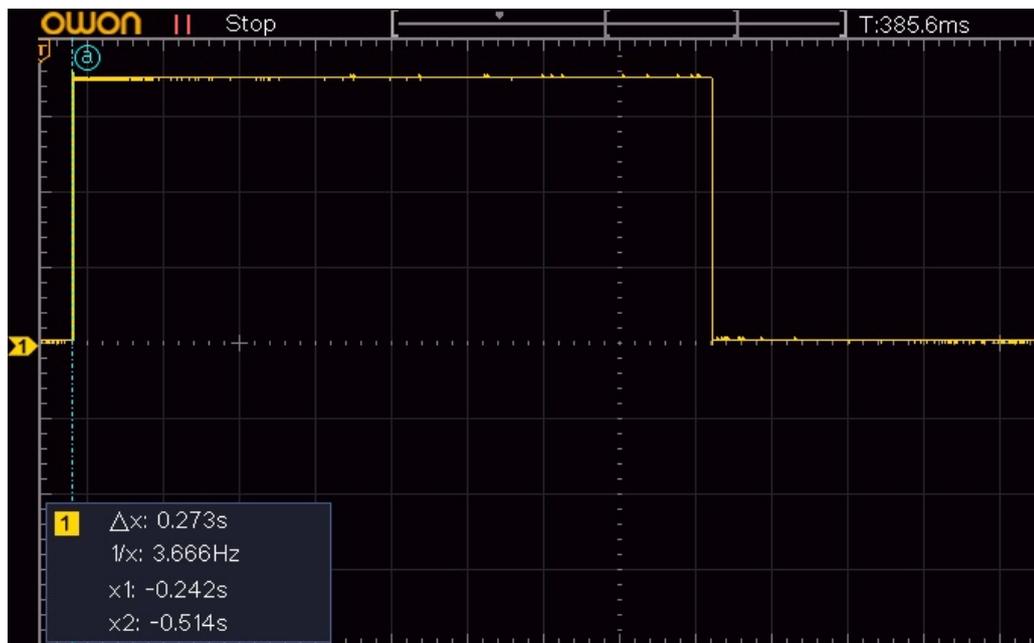


Figura 26 – Leitura do Sensor de efeito Hall vista no osciloscópio para entrada de 7V.

A Figura 27 demonstra o resultado do cálculo da FFT no MATLAB para entrada em 7V, que apresentou uma frequência fundamental de 3,695Hz.

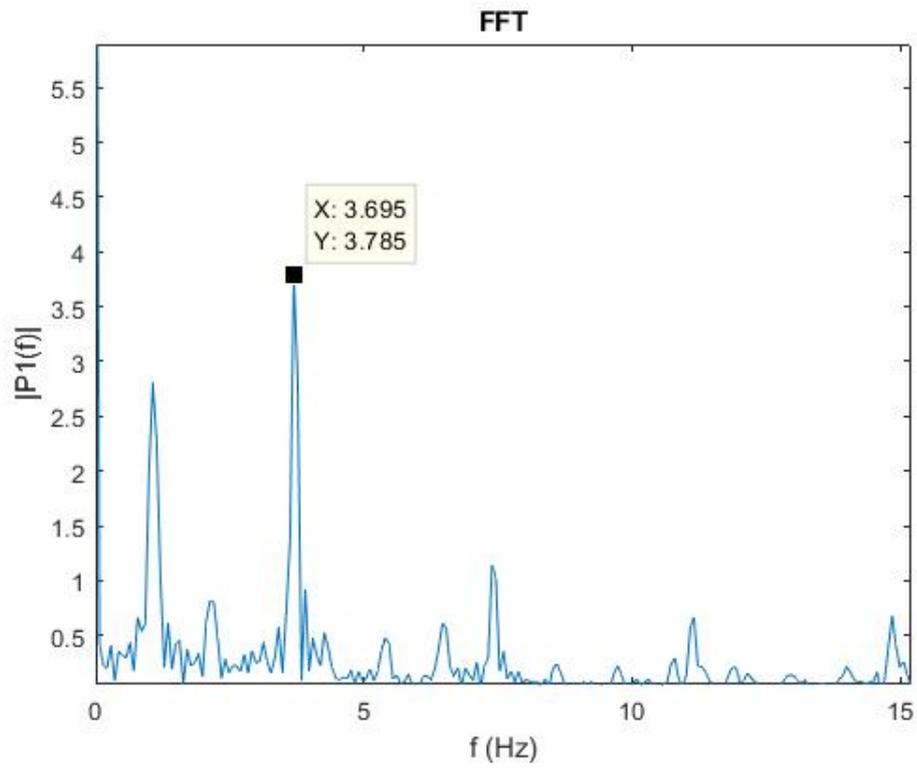


Figura 27 – Resultado do cálculo da FFT no MATLAB para entrada em 7V.

### 5.2.3 Resultados do Teste de Verificação de Funcionamento de Eixos de Orientação

Por fim, foram realizados testes com o intuito de verificar o correto funcionamento de cada eixo do sensor MPU6050. Foram realizados 3 ensaios na bancada de testes com a tensão de entrada do motor ajustada para 5V. A explicação dos testes em cada eixo está exposta em 5.1.2. Os resultados dos testes estão dispostos a seguir.

As Figuras 28 e 29 são referentes aos resultados dos testes com o dispositivo orientado com o eixo X no movimento do braço pendular.



Figura 28 – Leitura da Pulseira para o Eixo X.

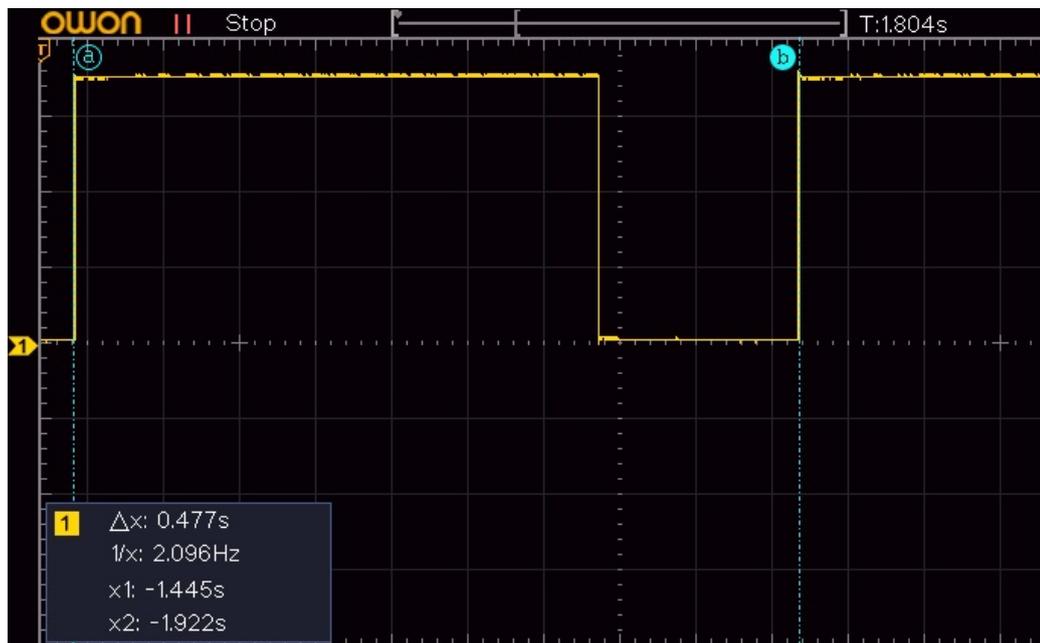


Figura 29 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo X.

As Figuras 30 e 31 são referentes aos resultados dos testes com o dispositivo orientado com o eixo Y no movimento do braço pendular.



Figura 30 – Leitura da Pulseira para o Eixo Y.

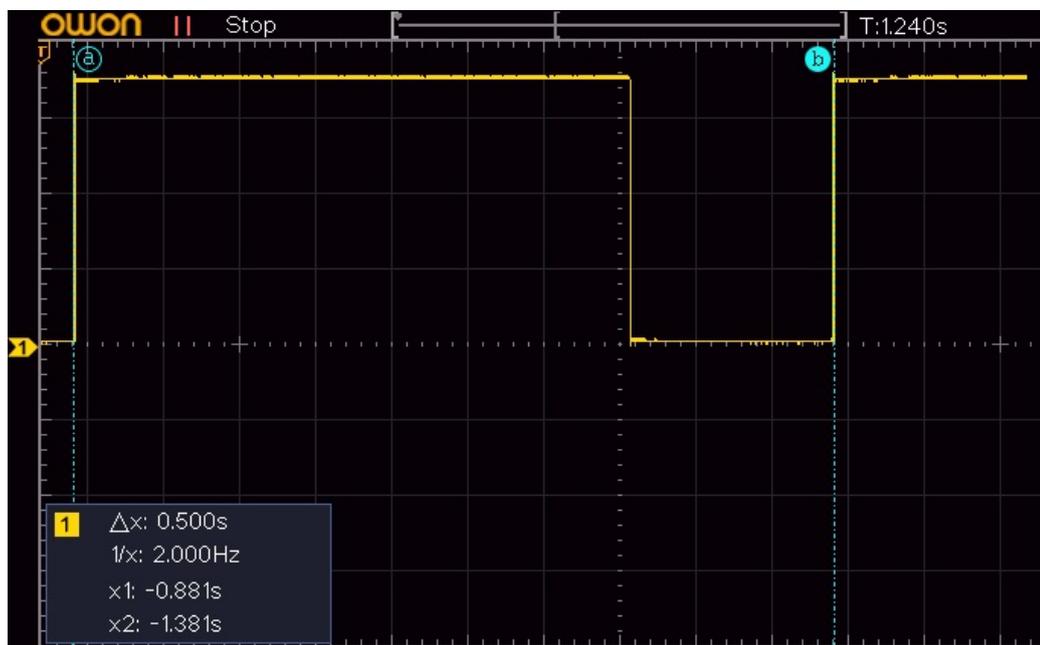


Figura 31 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo Y.

As Figuras 32 e 33 são referentes aos resultados dos testes com o dispositivo orientado com o eixo Z no movimento do braço pendular.



Figura 32 – Leitura da Pulseira para o Eixo Z.

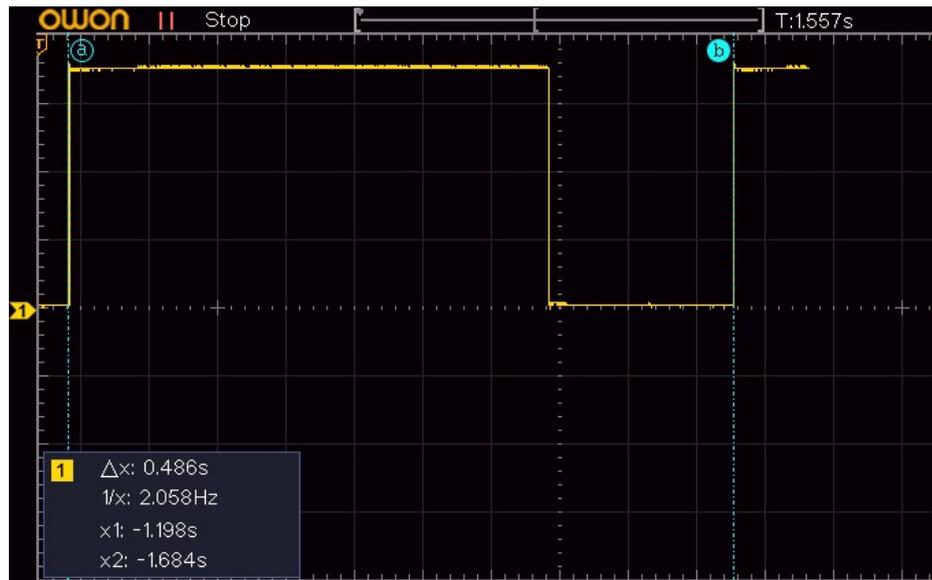


Figura 33 – Leitura do Sensor de efeito Hall vista no osciloscópio para Eixo Z.

Como pode ser observado nas Figuras 28 à 33, todos os eixos apresentaram uma resposta congruente com a leitura de frequência da bancada de testes.

#### 5.2.4 Análise de Resultados

Como observado em 5.2.1, 5.2.2 e 5.2.3 os resultados obtidos em todos os testes apresentaram uma discrepância pequena (Tabela 4), o que indica um excelente funcionamento do sistema desenvolvido e uma boa precisão dos dados. Todos os resultados de controle apresentaram valores muito parecidos com o obtido, apresentando diferenças centesimais.

Testes	Sensor de Efeito Hall e Pulseira	MATLAB e Pulseira
5V	0,42%	2,59%
7V	0,11%	0,68%
Eixo X	1,13%	-
Eixo Y	1,48%	-
Eixo Z	0,39%	-

Tabela 4 – Erros Percentuais dos Testes Realizados.

É importante notar que para a aplicação proposta a precisão da leitura não é de extrema relevância, principalmente pelo fato de que as análises de tremores de pacientes da doença de Parkinson são realizadas atualmente de maneira pouco precisa. Dessa forma, o dispositivo desenvolvido poderá melhorar os dados colhidos em pacientes com a DP e auxiliar em análises estatísticas mais precisas.

Os bons resultados obtidos nos testes podem ser atribuídos à aplicação da transformada rápida de Fourier ao código do projeto, uma vez que esta filtra movimentos ruidosos que poderiam interferir na leitura da frequência. Por conseguinte, apenas a frequência fundamental do movimento é obtida e o resultado se torna mais preciso.

## 6 Conclusões

O presente trabalho foi desenvolvido com o intuito de gerar os conhecimentos necessários para o projeto e desenvolvimento de um dispositivo capaz de medir tremores de pacientes portadores da Doença de Parkinson e indicar quais são suas frequências harmônicas fundamentais. Os resultados obtidos, por meio de testes, são promissores, visto que foi possível realizar a medição de frequências conhecidas, com os resultados obtidos muito próximos dos valores esperados, apesar de existirem perturbações e ruídos no sinal obtido.

O projeto cumpre com os requisitos estabelecidos de tamanho, portabilidade e conforto, estabelecidos na idealização do trabalho. Este possui grande potencial para estudos futuros a respeito dos movimentos apresentados por portadores da Doença de Parkinson e outras doenças que apresentem tremores de membros como sintoma, como o Tremor Essencial.

Também é possível a implementação de *threads* para a implementação de processamento paralelo, uma vez que o microcontrolador possui dois núcleos disponíveis, tornando possível o cálculo da FFT em tempo real. Outra opção seria a implementação de uma análise da amplitude de movimento do tremor através de uma integral dupla.

Com a conclusão desse trabalho, espera-se que o dispositivo desenvolvido possa somar à evolução dos estudos para o controle da Doença de Parkinson. Porém, vale ressaltar que o dispositivo desenvolvido é versátil o suficiente para ser adaptado para quaisquer outras aplicações que envolvam o estudo da frequência fundamental de movimentos, bastando apenas acoplar o dispositivo ao corpo que realize o deslocamento.

### 6.1 Trabalhos Futuros

Com base nos resultados obtidos neste trabalho é possível projetar uma perspectiva de futuros trabalhos que esse dispositivo pode auxiliar:

- Estudo de caracterização de frequências de tremores de portadores da Doença de Parkinson;
- Estudo de caracterização de frequências de portadores de outras doenças relacionadas ao tremor de membros, como a doença de Tremor Essencial;
- Análise de vibrações em frequência abaixo de 21Hz (Valor ajustado para o Filtro Passa-Baixas Digital do MPU6050);
- Desenvolvimento de interface gráfica para usuário final;

- Desenvolvimento de comunicação sem fio mais robusta, com menor probabilidade de perda de pacotes, e de menor gasto energético;
- Implementação e processamento paralelo com cálculo da FFT em tempo real;
- Implementação de análise de amplitude de movimento.

# Referências

- ALEXANDER, C. K. **Fundamentos de Circuitos Elétricos**. 5 ed.: McGraw-Hill, 2013. ISBN 978-85-8055-173-0. Citado nas pp. 23, 24.
- BHAGWATH, G. Tremors in Elderly Persons: Clinical Features and Management, 2001. Citado na p. 15.
- BORGES, V.; FERRAZ, H. B. Tremores. **Revista Neurociências**, v. 14, p. 43–47, 1 mar. 2006. ISSN 1984-4905. DOI: 10.34024/RNC.2006.V14.8786. Disponível em: <<https://periodicos.unifesp.br/index.php/neurociencias/article/view/8786>>. Citado nas pp. 21, 22.
- CARVALHO, T. **Movimento Harmônico Simples (MHS) - InfoEscola**. Mar. 2014. Disponível em: <<https://www.infoescola.com/fisica/movimento-harmomico-simples-mhs/>>. Citado na p. 20.
- CHARCOT, J. Leçons du Mardi: Policlinique de la Salpêtrière, 1887–1888. **Bureaux du Progrès Médical, Paris: Lesson of June**, v. 12, p. 1888, 1888. Citado na p. 15.
- CICHACZEWSKI, E.; MUNHOZ, R. P.; MAIA, J. M.; NOHAMA, P.; NÓVAK, E. M.; TEIVE, H. A. Electrophysiologic characteristics of tremor in Parkinson's disease and essential tremor. **Arquivos de Neuro-Psiquiatria**, Academia Brasileira de Neurologia - ABNEURO, v. 72, p. 301–306, 4 2014. ISSN 0004-282X. DOI: 10.1590/0004-282X20140006. Disponível em: <<http://www.scielo.br/j/anp/a/FNMcgRcwCDZgzgfc3qPLwn/?lang=en>>. Citado nas pp. 15, 18, 22.
- CONDES, E. **arduinoFFT - Arduino Reference**. 2020. Disponível em: <<https://www.arduino.cc/reference/en/libraries/arduinofft/>>. Citado na p. 37.
- COSTA, J. L. L.; PAIXÃO, F. C. **Sistema para Auxiliar a Detecção da Doença de Parkinson Interpretando Sinais de Tremor de Repouso Parkinsoniano Utilizando Aprendizado de Máquinas**. 2018. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Faculdade UnB Gama – FGA. Citado na p. 17.
- FERNANDES, S.; SILVA, F.; SILVA, T. PROPOSTA DE DESENVOLVIMENTO DE EMPUNHADURA PERSONALIZADA PARA USUÁRIOS COM MAL DE PARKINSON. **Anais do 11º Congresso Brasileiro de Pesquisa e Desenvolvimento em Desing**, Editora Edgard Blucher, Ltda., v. 1, p. 2387–2399, 4 dez. 2014. ISSN 2318-6968. DOI: 10.5151/DESIGNPRO-PED-00782. Citado nas pp. 14, 15.

- GOETZ, C. G. The History of Parkinson's Disease: Early Clinical Descriptions and Neurological Therapies. **Cold Spring Harbor Perspectives in Medicine**, Cold Spring Harbor Laboratory Press, v. 1, a008862, 1 set. 2011. ISSN 21571422. DOI: 10.1101/CSHPERSPECT.A008862. Disponível em: <<http://perspectivesinmedicine.cshlp.org/content/1/1/a008862.full>>. Citado na p. 14.
- GUIMARÃES, I. J. A. **Desenvolvimento tecnológico de um dispositivo de coleta de sinais sEMG aplicado à avaliação da doença de Parkinson**. Dez. 2017. F. 1–70. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Eletrônica) – Faculdade UnB Gama – FGA. Citado na p. 17.
- HELERBROCK, R. **Movimento harmônico simples: o que é, cálculo - Mundo Educação**. 2020. Disponível em: <<https://mundoeducacao.uol.com.br/fisica/movimento-harmonico-simples.htm>>. Citado nas pp. 20, 21.
- HELTEC, A. **WIFI LoRa 32(V2) Pinout Diagram**. 2018. Disponível em: <[https://resource.heltec.cn/download/WiFi\\_LoRa\\_32/WIFI\\_LoRa\\_32\\_V2.pdf](https://resource.heltec.cn/download/WiFi_LoRa_32/WIFI_LoRa_32_V2.pdf)>. Citado nas pp. 32, 33.
- HU, I. **Módulo Acelerômetro/Giroscópio MPU-6050**. 2019. Disponível em: <<https://www.huinfinito.com.br/modulos/923-modulo-acelerometrogiroscopio-mpu-6050.html>>. Citado na p. 33.
- INVENSENSE, I. **MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2 MPU-6000/MPU-6050 Register Map and Descriptions**. 2013. Disponível em: <<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf>>. Citado na p. 35.
- LANDAU, I. D.; GIANLUCA, Z. **Digital Control Systems**. 1. ed.: Springer London, 2006. ISBN 978-1-84628-056-6. DOI: 10.1007/978-1-84628-056-6. Citado na p. 24.
- LATHI, B. P. **Sinais e Sistemas Lineares**. 2.ed: Bookman, 2006. ISBN 85-60031-13-8. Citado nas pp. 23, 25, 27–30.
- LYONS, R.; FUGAL, D. L. **The Essential Guide to Digital Signal Processing**. 1.ed: Prentice Hall, 2014. ISBN 9780133804423. Citado na p. 23.
- MAGALHÃES, A. B. **Dia Mundial de Conscientização da Doença de Parkinson | Biblioteca Virtual em Saúde MS**. 2015. Disponível em: <<https://bvsmms.saude.gov.br/dia-mundial-de-conscientizacao-da-doenca-de-parkinson/>>. Citado na p. 14.
- MATLAB – Wikipédia, a enciclopédia livre. Disponível em: <<https://pt.wikipedia.org/wiki/MATLAB>>. Citado na p. 46.
- MORIN, D. **Fourier analysis**. 2009. Disponível em: <[https://scholar.harvard.edu/files/david-morin/files/waves\\_fourier.pdf](https://scholar.harvard.edu/files/david-morin/files/waves_fourier.pdf)>. Citado nas pp. 25–27.

- NEDELKOVSKI, D. **MEMS Accelerometer Gyroscope Magnetometer Arduino**. 2018. Disponível em: <<https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>>. Citado na p. 34.
- NORTE, D. B. Alunos criam pulseira de monitoramento para portadores de Parkinson. **234**, abr. 2019. Disponível em: <<https://revistaensinosuperior.com.br/alunos-parkinson/>>. Citado na p. 18.
- PÁDUA MACHADO, A. R. de. **Visualização e Classificação de Características para a Discriminação Entre Indivíduos com a Doença de Parkinson Submetidos a Tratamento com Levodopa e Estimulação Profunda do Cérebro**. Fev. 2017. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Uberlândia. DOI: <http://dx.doi.org/10.14393/ufu.te.2017.49>. Disponível em: <<http://clyde.dr.ufu.br/bitstream/123456789/18291/1/VisualizacaoClassificacaoCaracteristicas.pdf>>. Citado na p. 18.
- PATEL, S.; LORINCZ, K.; HUGHES, R.; HUGGINS, N.; GROWDON, J.; STANDAERT, D.; AKAY, M.; DY, J.; WELSH, M.; BONATO, P. Monitoring motor fluctuations in patients with parkinsons disease using wearable sensors. **IEEE Transactions on Information Technology in Biomedicine**, v. 13, p. 864–873, 6 nov. 2009. ISSN 10897771. DOI: [10.1109/TITB.2009.2033471](https://doi.org/10.1109/TITB.2009.2033471). Citado na p. 18.
- RISSANEN, S. M.; KANKAANPÄÄ, M.; TARVAINEN, M. P.; NOVAK, V.; NOVAK, P.; HU, K.; MANOR, B.; AIRAKSINEN, O.; KARJALAINEN, P. A. Analysis of EMG and acceleration signals for quantifying the effects of deep brain stimulation in parkinsons disease. **IEEE Transactions on Biomedical Engineering**, v. 58, p. 2545–2553, 9 set. 2011. ISSN 00189294. DOI: [10.1109/TBME.2011.2159380](https://doi.org/10.1109/TBME.2011.2159380). Citado na p. 18.
- SILVA, A. P. S. P. B. da. **O Uso De Sensores Inerciais Para Caracterização E Classificação Do Tremor De Punho Em Indivíduos Com A Doença De Parkinson E Correlação Com A Escala De Avaliação Subjetiva: Updrs**. Dez. 2018. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Uberlândia. Disponível em: <<http://repositorio.ufu.br/bitstream/123456789/24649/1/UsoSensoresInerciais.pdf>>. Citado na p. 18.
- SOUZA, L. F. M. de; COLETTA, M. V. D.; SOARES, F. J. A. **Identificação do Tremor de Parkinson**. 2018. Citado na p. 18.
- SPYERS-ASHBY, J. M.; STOKES, M. J. Reliability of tremor measurements using a multidimensional electromagnetic sensor system. **Clinical Rehabilitation**, Arnold, v. 14, p. 425–432, 4 jul. 2000. ISSN 02692155. DOI: [10.1191/0269215500cr328oa](https://doi.org/10.1191/0269215500cr328oa). Disponível em: <<https://journals.sagepub.com/doi/abs/10.1191/0269215500cr328oa>>. Citado na p. 18.

- TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. **Sistemas Digitais: Princípios e Aplicações**. 11. ed.: Pearson, 2011. ISBN 978-85-4300-694-9. Citado nas pp. [24](#), [25](#).
- WILE, D. J.; RANAWAYA, R.; KISS, Z. H. Smart watch accelerometry for analysis and diagnosis of tremor. **Journal of Neuroscience Methods**, Elsevier, v. 230, p. 1–4, jun. 2014. ISSN 0165-0270. DOI: [10.1016/J.JNEUMETH.2014.04.021](#). Citado na p. [18](#).
- ZHOU, Y.; JENKINS, M. E.; NAISH, M. D.; TREJOS, A. L. Characterization of parkinsonian hand tremor and validation of a high-order tremor estimator. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, Institute of Electrical e Electronics Engineers Inc., v. 26, p. 1823–1834, 9 set. 2018. ISSN 15344320. DOI: [10.1109/TNSRE.2018.2859793](#). Citado na p. [18](#).

# Apêndices

# APÊNDICE A – Códigos de programação

## A.1 Códigos de programação

Código A.1 – Código de Arduino

```

1  #include "heltec.h"
2
3  #define BAND 433E6 //Frequencia do radio - exemplo : 433E6, 868E6,
   915E6
4  /* Offset de linhas no display OLED */
5  #define LINE1      0
6  #define LINE2      20
7  #define LINE3      40
8  #define LINE4      50
9
10
11 //----- FFT -----
12 #include "arduinoFFT.h"
13
14
15 arduinoFFT FFT = arduinoFFT(); // Create FFT object
16
17 #define SCL_INDEX 0x00
18 #define SCL_TIME 0x01
19 #define SCL_FREQUENCY 0x02
20 #define SCL_PLOT 0x03
21
22 //----- Variables for FFT -----
23     const uint16_t samples = 4096; //Deve ser potencia de 2
24     const double samplingFrequency = 1000;
25     double vReal[samples];
26     double vImag[samples];
27
28
29
30 //----- I2C -----
31 #include <Wire.h>
32 //SCL-15  SDA-4
33
34 //----- WebSerial -----
35
36 #include <Arduino.h>
37 #include <WiFi.h>
38 #include <AsyncTCP.h>

```

```

39 #include <ESPAsyncWebServer.h>
40 #include <WebSerial.h>
41
42 AsyncWebServer server(80);
43
44 const char* ssid = "";           // Your WiFi SSID
45 const char* password = "";      // Your WiFi Password
46
47 String LocalIP;
48
49 //----- MPU6050 Measurement & Filtering Range -----
50
51 #define AFS_SEL 2 // Accelerometer Configuration Settings
52   AFS_SEL=2, Full Scale Range = +/- 8 [g]
53 #define DLPF_SEL 4 // DLPF Configuration Settings Accel BW
54   21Hz, Delay 8.5ms / Gyro BW 20Hz, Delay 8.3ms, Fs 1KHz
55
56 //----- Variables for gravity -----
57
58 const int MPU_ADDR = 0x68; // I2C address of the MPU-6050
59 int AcX, AcY, AcZ; // Accelerometer values
60 int Acavg; // Accelerometer average
61
62 long Cal_AcX, Cal_AcY, Cal_AcZ, Cal_AcRes; //Calibration values
63
64 float GAcX, GAcY, GAcZ, GAcRes; // Convert accelerometer to
65   gravity value
66
67 float Min_GAcX = 0, Max_GAcX = 0, PtoP_GAcX, Min_GAcY = 0,
68   Max_GAcY = 0, PtoP_GAcY, Min_GAcZ = 0, Max_GAcZ = 0,
69   PtoP_GAcZ, Min_GAcRes = 0, Max_GAcRes = 0, PtoP_GAcRes=0; //
70   Finding Min, Max & Peak to Peak of gravity value
71
72 float Min = 0, Max = 0; // Initial value of Min, Max
73
74 int cnt,PtoP_GAcRes_count=0; // Count of calibration process
75
76 float Grvt_unit; // Gravity value unit
77
78 long period, prev_time; // Period of calculation
79
80 //----- General Variables -----
81
82 unsigned long previousMillis = 0;
83 const long interval = 10000;
84 int i = 0,j = 0, rst = 22, smplcnt=0;
85
86 #include "MPU6050_6Axis_MotionApps20.h"
87 MPU6050 mpu;
88 uint8_t mpuIntStatus; // holds actual interrupt status byte from
89   MPU
90

```

```
84 volatile bool mpuInterrupt = false; // indicates whether MPU
    interrupt pin has gone high
85
86 void dmpDataReady() {
87     mpuInterrupt = true;
88 }
89
90 void setup() {
91
92     Heltec.begin(true /*DisplayEnable Enable*/, true /*Heltec.LoRa
        Enable*/, true /*Serial Enable*/, true /*PABOOST Enable*/,
        BAND /*long BAND*/);
93
94     attachInterrupt(digitalPinToInterrupt(2), dmpDataReady, RISING);
95     mpuIntStatus = mpu.getIntStatus();
96
97     Wire.begin(4,15);
98
99     pinMode(rst, INPUT_PULLUP);
100
101     init_MPU6050();
102
103     Serial.begin(115200);
104
105     while (!Serial) {
106         ; // wait for serial port to connect. Needed for native USB
            port only
107     }
108
109     Heltec.display->init();
110
111     Heltec.display->setFont(ArialMT_Plain_10);
112
113     WiFi.mode(WIFI_STA);
114     WiFi.begin(ssid, password);
115     if (WiFi.waitForConnectResult() != WL_CONNECTED) {
116         Heltec.display->drawString(0, LINE4, "Sem conexão WiFi!");
117         Heltec.display->display();
118     }
119     Serial.print("IP Address: ");
120     Serial.println(WiFi.localIP());
121     LocalIP = String() + WiFi.localIP()[0] + "." + WiFi.localIP()[1]
        + "." + WiFi.localIP()[2] + "." + WiFi.localIP()[3];
122     Heltec.display->drawString(0, LINE4, LocalIP + "/webserial");
123     Heltec.display->display();
124     // WebSerial is accessible at "<IP Address>/webserial" in browser
125     WebSerial.begin(&server);
126     WebSerial.msgCallback(recvMsg);
127     server.begin();
128
129
130     Gravity_Range_Option();
```

```
131
132 Calib_MPU6050(); // Calculating calibration value
133
134 Heltec.display->drawString(0, LINE2, "Pressione o botão para");
135 Heltec.display->drawString(0, 30, "continuar com a medição");
136 Heltec.display->display();
137
138 while(digitalRead(rst)!=LOW);
139 delay(2000);
140 previousMillis = millis();
141 }
142
143 void loop() {
144
145     unsigned long currentMillis = millis();
146
147
148     if ( currentMillis - previousMillis <= interval) {
149
150         ReadData_MPU6050();
151
152         Calc_Grvt();
153
154         Display_Grvt();
155
156
157         if(smplcnt<(samples+100) ){
158             if(smplcnt>=100) {
159                 vReal[smplcnt-100] = GAcRes;
160                 WebSerial.println(String(vReal[smplcnt-100]));
161                 //vRealx[smplcnt-100] = GAcX;
162                 //vRealy[smplcnt-100] = GAcY;
163                 //vRealz[smplcnt-100] = GAcZ;
164                 vImag[smplcnt-100] = 0;
165                 delay(((int)(round(int(interval-1000)/int(samples) ) ) ) ) );
166             }
167             smplcnt++;
168         }
169
170     }
171     else if(j==0) { //Fim Medicao
172
173         CalculaFFT();
174         //CalculaFFTx();
175
176         j++;
177
178     }
179
180     if ( digitalRead(rst) == LOW && currentMillis - previousMillis >
181         interval){//Reset
```

```

182     Heltec.display->clear();
183     //     cnt=0;           //Recalibrar
184     //     Calib_MPU6050(); //Recalibrar
185     PtoP_GAcRes_count=0;
186     PtoP_GAcRes=0;
187     previousMillis = millis();
188     currentMillis = millis();
189     j=0;
190     smplcnt=0;
191     i=0;
192 }
193 }
194
195 void init_MPU6050() {
196     //MPU6050 Initializing & Reset
197     Wire.beginTransmission(MPU_ADDR);
198     Wire.write(0x6B); // PWR_MGMT_1 register
199     Wire.write(0); // set to zero (wakes up the MPU-6050)
200     Wire.endTransmission(true);
201
202     //MPU6050 Clock Type
203     Wire.beginTransmission(MPU_ADDR);
204     Wire.write(0x6B); // PWR_MGMT_1 register
205     Wire.write(0x03); // Selection Clock 'PLL with Z axis
206     // gyroscope reference'
207     Wire.endTransmission(true);
208
209     //MPU6050 Accelerometer Configuration Setting
210     Wire.beginTransmission(MPU_ADDR);
211     Wire.write(0x1C); // Accelerometer Configuration register
212     if (AFS_SEL == 0) Wire.write(0x00); // AFS_SEL=0, Full Scale
213     // Range = +/- 2 [g]
214     else if (AFS_SEL == 1) Wire.write(0x08); // AFS_SEL=1, Full
215     // Scale Range = +/- 4 [g]
216     else if (AFS_SEL == 2) Wire.write(0x10); // AFS_SEL=2, Full
217     // Scale Range = +/- 8 [g]
218     else Wire.write(0x18); // AFS_SEL=3, Full Scale Range = +/-
219     // 16 [g]
220     Wire.endTransmission(true);
221
222     //MPU6050 DLPF(Digital Low Pass Filter)
223     Wire.beginTransmission(MPU_ADDR);
224     Wire.write(0x1A); // DLPF_CFG register
225     if (DLPF_SEL == 0) Wire.write(0x00); // Accel BW 260Hz, Delay
226     // 0ms / Gyro BW 256Hz, Delay 0.98ms, Fs 8KHz
227     else if (DLPF_SEL == 1) Wire.write(0x01); // Accel BW 184Hz,
228     // Delay 2ms / Gyro BW 188Hz, Delay 1.9ms, Fs 1KHz
229     else if (DLPF_SEL == 2) Wire.write(0x02); // Accel BW 94Hz,
230     // Delay 3ms / Gyro BW 98Hz, Delay 2.8ms, Fs 1KHz
231     else if (DLPF_SEL == 3) Wire.write(0x03); // Accel BW 44Hz,
232     // Delay 4.9ms / Gyro BW 42Hz, Delay 4.8ms, Fs 1KHz

```

```
224     else if (DLPF_SEL == 4) Wire.write(0x04);    // Accel BW 21Hz,
        Delay 8.5ms / Gyro BW 20Hz, Delay 8.3ms, Fs 1KHz
225     else if (DLPF_SEL == 5) Wire.write(0x05);    // Accel BW 10Hz,
        Delay 13.8ms / Gyro BW 10Hz, Delay 13.4ms, Fs 1KHz
226     else Wire.write(0x06);    // Accel BW 5Hz, Delay 19ms / Gyro
        BW 5Hz, Delay 18.6ms, Fs 1KHz
227     Wire.endTransmission(true);
228 }
229
230 void Gravity_Range_Option() {
231     switch (AFS_SEL) { // Selecting Gravity unit value, LSB/g
232         case 0:
233             Grvt_unit = 16384;
234             break;
235         case 1:
236             Grvt_unit = 8192;
237             break;
238         case 2:
239             Grvt_unit = 4096;
240             break;
241         case 3:
242             Grvt_unit = 2048;
243             break;
244     }
245 }
246
247 void Calib_MPU6050() {
248     int x=10;
249     for (int i = 0 ; i < 2000 ; i++) { // Summing Iteration for
        finding calibration value
250         if (i % 200 == 0) { // Display progress every 200 cycles
251             cnt++;
252             if (cnt == 1) { // Characters to display first
253                 Heltec.display->clear();
254                 Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
255                 Heltec.display->drawString(20, LINE1, "Calibrando");
256                 Heltec.display->drawString(0, LINE2, ".");
257                 Heltec.display->drawString(0, LINE4, LocalIP +
                    "/webserial");
258                 Heltec.display->display();
259             }
260             else { //Display progress by point
261                 Heltec.display->drawString(x,LINE2, +".");
262                 Heltec.display->display();
263                 x+=10;
264             }
265         }
266
267         ReadData_MPU6050(); // Read Accelerometer data
268
269         delay(10);
270
```

```

271     // Soma dos dados
272     Cal_AcX += AcX;
273     Cal_AcY += AcY;
274     Cal_AcZ += AcZ;
275 }
276
277 // Media dos dados
278 Cal_AcX /= 2000;
279 Cal_AcY /= 2000;
280 Cal_AcZ /= 2000;
281
282 Heltec.display->clear();
283 Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
284 Heltec.display->drawString(20, LINE1, "ValorCalib X|Y|Z");
285 Heltec.display->drawString(20, 10, String(Cal_AcX) + "|" +
    String(Cal_AcY) + "|" + String(Cal_AcZ));
286 Heltec.display->drawString(0, LINE4, LocalIP + "/webserial");
287 Heltec.display->display();
288 delay(1000);
289 }
290
291 void ReadData_MPU6050() {
292     Wire.beginTransmission(MPU_ADDR);
293     Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
294     Wire.endTransmission(false);
295     //Wire.requestFrom(MPU_ADDR,14,true); // request a total of 14
    registers
296     Wire.requestFrom(MPU_ADDR, 6, true); // request a total of 14
    registers
297     AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) &
    0x3C (ACCEL_XOUT_L)
298     AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) &
    0x3E (ACCEL_YOUT_L)
299     AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) &
    0x40 (ACCEL_ZOUT_L)
300 }
301
302 void Calc_Grvt() {
303     AcX = (AcX - Cal_AcX); // Calibrated Accelerometer value
304     AcY = (AcY - Cal_AcY); // Calibrated Accelerometer value
305     AcZ = (AcZ - Cal_AcZ); // Calibrated Accelerometer value
306
307     Acavg = sqrt(pow(AcX,2) + pow(AcY,2) + pow(AcZ,2));
308
309     GAcX = AcX / Grvt_unit; // Converting the Calibrated value to
    Gravity value
310     GAcY = AcY / Grvt_unit; // Converting the Calibrated value to
    Gravity value
311     GAcZ = AcZ / Grvt_unit; // Converting the Calibrated value to
    Gravity value
312
313     GAcRes = Acavg / Grvt_unit;

```

```

314
315 PtoP_GAcRes = PtoP_GAcRes + GAcRes;
316 PtoP_GAcRes_count ++;
317
318 //----- Calculating Min, Max & Peak to Peak of Gravity
319 -----
320 Min_GAcX = min(Min_GAcX, GAcX);
321 Max_GAcX = max(Max_GAcX, GAcX);
322 PtoP_GAcX = Max_GAcX - Min_GAcX;
323
324 Min_GAcY = min(Min_GAcY, GAcY);
325 Max_GAcY = max(Max_GAcY, GAcY);
326 PtoP_GAcY = Max_GAcY - Min_GAcY;
327
328 Min_GAcZ = min(Min_GAcZ, GAcZ);
329 Max_GAcZ = max(Max_GAcZ, GAcZ);
330 PtoP_GAcZ = Max_GAcZ - Min_GAcZ;
331
332 }
333
334 void Display_Grvt() {
335 //----- Serial print -----
336 //WebSerial.print("AcX= " + String(AcX));
337 //WebSerial.print(" |AcY= " + String(AcY));
338 // WebSerial.println(" |AcZ= " + String(AcZ));
339 Serial.println(String(GAcRes));
340
341 //----- OLED Display -----
342 period = millis() - prev_time;
343 //
344 if (period > 1000) {
345
346     prev_time = millis();
347     Heltec.display->clear();
348     Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
349     Heltec.display->drawString(20, LINE1, "Amp X|Y|Z");
350     Heltec.display->drawString(20, LINE2, String(PtoP_GAcX) + "|"
351         + String(PtoP_GAcY) + "|" + String(PtoP_GAcZ));
351     Heltec.display->drawString(0, LINE4, LocalIP + "/webserial");
352     Heltec.display->display();
353
354
355     Min_GAcX = 0;
356     Max_GAcX = 0;
357     Min_GAcY = 0;
358     Max_GAcY = 0;
359     Min_GAcZ = 0;
360     Max_GAcZ = 0;
361 }
362 }

```

```
363 void PrintVector(double *vData, uint16_t bufferSize, uint8_t
    scaleType, const double samplingFrequency)
364 {
365     for (uint16_t k = 0; k < bufferSize; k++)
366     {
367         double abscissa; // Print abscissa value
368         switch (scaleType)
369         {
370             case SCL_INDEX:
371                 abscissa = (k * 1.0);
372             break;
373             case SCL_TIME:
374                 abscissa = ((k * 1.0) / samplingFrequency);
375             break;
376             case SCL_FREQUENCY:
377                 abscissa = ((k * 1.0 * samplingFrequency) / bufferSize);
378             break;
379         }
380         WebSerial.print(String(abscissa, 6));
381         if(scaleType==SCL_FREQUENCY)
382             WebSerial.print("Hz");
383             WebSerial.print(" ");
384             WebSerial.println(String(vData[k], 4));
385     }
386     WebSerial.println();
387 }
388
389 void CalculaFFT()
390 {
391
392     //WebSerial.println("Data:");
393     //PrintVector(vReal, samples, SCL_TIME, samplingFrequency);
394
395     FFT.Windowing(vReal, samples, FFT_WIN_TYP_HAMMING,
        FFT_FORWARD); //Weigh data
396
397     //WebSerial.println("Weighed data:");
398     //PrintVector(vReal, samples, SCL_TIME, samplingFrequency);
399
400
401     FFT.Compute(vReal, vImag, samples, FFT_FORWARD); // Compute FFT
402     //WebSerial.println("Computed Real values:");
403     //PrintVector(vReal, samples, SCL_INDEX, samplingFrequency);
404     //WebSerial.println("Computed Imaginary values:");
405     //PrintVector(vImag, samples, SCL_INDEX, samplingFrequency);
406
407     FFT.ComplexToMagnitude(vReal, vImag, samples); // Compute
        magnitudes
408
409     double res = (FFT.MajorPeak(vReal, samples,
        samplingFrequency))/2;
410     WebSerial.print("Major Peak: ");
```

```
411     WebSerial.println(String(res, 6)+"Hz");
412
413     Heltec.display->clear();
414     Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
415     Heltec.display->drawString(0, LINE1, "AmpMd|Freq Fund");
416     Heltec.display->drawString(0, 10,
        String(PtoP_GAcRes/PtoP_GAcRes_count) + "m/s2 |" +
        String(res,2) + "Hz");
417     Heltec.display->drawString(0, LINE4, LocalIP + "/webserial");
418     Heltec.display->display();
419     // PrintVector(vReal, (samples >> 1), SCL_FREQUENCY,
        samplingFrequency);
420
421
422         WebSerial.printf("Fundamental Freq : %f Hz\t Mag: %f
            g\n", FFT.majorPeakFreq(),
            (FFT.majorPeak()/10000)*2/FFT_N);
423     //     for (int k=0; k< 10; k++) {
424     //         WebSerial.printf("%f:%f\n",
        FFT.frequency(k),fft_output[k]);
425     //     }
426
427 }
428
429 void recvMsg(uint8_t *data, size_t len)
430 {
431     WebSerial.println("Received Data...");
432     String d = "";
433     for(int i=0; i < len; i++){
434         d += char(data[i]);
435     }
436     WebSerial.println(d);
437     if (d == "ON"){
438         //digitalWrite(LED, HIGH);
439     }
440     if (d=="OFF"){
441         //digitalWrite(LED, LOW);
442     }
443 }
```

## APÊNDICE B – Vídeo Demonstrando Movimento da Bancada de Testes

Use um leitor de QRcode para acessar o vídeo na Figura 34.



Figura 34 – QR code de Vídeo Demonstrativo. <https://youtu.be/NbcQGNvjPUU>