

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE ENGENHARIA ELÉTRICA**

JOÃO MARCOS DA SILVA

**PROJETO E DESENVOLVIMENTO DE SISTEMA DE ACIONAMENTO PARA
PRÓTESE DE MÃO OPEN SOURCE**

São Leopoldo

2021

JOÃO MARCOS DA SILVA

**PROJETO E DESENVOLVIMENTO DE SISTEMA DE ACIONAMENTO PARA
PRÓTESE DE MÃO OPEN SOURCE**

Trabalho de Conclusão de Curso
apresentado como requisito parcial para
obtenção do título de Bacharel em
Engenharia Elétrica, pelo Curso de
Engenharia Elétrica da Universidade do
Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. João Olegário de Oliveira de Souza

São Leopoldo

2021

RESUMO

As atuais pesquisas no campo da eletromiografia e acionamento de próteses vêm possibilitando o aprimoramento de sistemas de aquisição de sinais EMG e de controle de motores. Apesar de tais avanços, as próteses de mão ativas ainda não são as mais requisitadas por pessoas amputadas devido à sua complexidade e alto custo. Este trabalho teve como objetivo desenvolver um sistema capaz de acionar os motores de uma prótese utilizando conceitos como: sistema de controle de sinais PWM; programação de microcontroladores; desenvolvimento de aplicativos *open source*. Para isso, o sistema foi implementado tendo como interface um microcontrolador ESP32, responsável por interpretar os gestos e acionar os motores da prótese através de códigos programados via *software*. O envio da informação dos gestos foi dividido em duas etapas: emulador *Bluetooth* do sistema de aquisição de sinais e teste via Raspberry Pi 3 como preparação para futura integração com o sistema de aquisição de sinais. Como modelo de prótese para testes, foi utilizada a prótese *open source* Handi Hand, impressa em 3D. O sistema de acionamento via supervisor integrado à prótese foi capaz de realizar gestos como abrir mão, fechar mão e polegar para cima. Já os movimentos de pegada esférica e cilíndrica, a prótese foi capaz de realizar utilizando apenas 3 dedos (polegar, indicador e médio) devido às limitações de articulação na sua construção. Para os testes de pronação, supinação e mandril, a prótese foi capaz de realizar movimentos simbólicos a fim de validar o sistema de acionamento. Os testes com a Raspberry foram realizados através de comunicação serial entre a Raspberry e a ESP32, utilizando uma rotina cíclica de envio de movimentos da Raspberry Pi para o microcontrolador.

Palavras-chave: Prótese-de-mão. Sinais EMG. Eletrônica de potência. ESP32.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Prótese estética..... | 12 |
| Figura 2 – Prótese de mão mecânica..... | 13 |
| Figura 3 – Prótese Ativa Comercial..... | 14 |
| Figura 4 – Próteses Multiarticulares..... | 15 |
| Figura 5 – Sistema Hannes..... | 16 |
| Figura 6 – Interface entre fonte e carga..... | 17 |
| Figura 7 – Motor DC de dois polos..... | 19 |
| Figura 8– Servomotor DC..... | 20 |
| Figura 9 – Motor DC Sem Escovas..... | 20 |
| Figura 10 – Topologia do Circuito Ponte H..... | 21 |
| Figura 11 – Topologia do Circuito Ponte H – Sentido Horário..... | 22 |
| Figura 12 – Topologia do Circuito Ponte H – Sentido Anti-Horário..... | 22 |
| Figura 13 – Microcontrolador ESP32..... | 23 |
| Figura 14 – Raspberry Pi 3 B+..... | 24 |
| Figura 15 – Exemplo de interface serial..... | 25 |
| Figura 16 – Diagrama de blocos das etapas para implementação de prótese de mão..... | 26 |
| Figura 17 – Arquitetura do Sistema..... | 29 |
| Figura 18 – Diagrama do sistema da prótese Adam’s Hand..... | 30 |
| Figura 19 – Objeto de estudo para trabalho de conclusão de curso..... | 31 |
| Figura 20 – Diagrama de funcionamento do sistema..... | 32 |
| Figura 21 – Instalação dos motores na prótese..... | 33 |
| Figura 22 – Palma da mão da Prótese..... | 35 |
| Figura 23 – Partes componentes - indicador..... | 36 |
| Figura 24 – Partes componentes – polegar..... | 36 |
| Figura 25 – Transmissão de movimento dos dedos..... | 37 |
| Figura 26 – Transmissão de movimento dos dedos..... | 38 |
| Figura 27 – Bloco de funções na plataforma Thinkable..... | 40 |
| Figura 28 – Bloco de <i>scan</i> dos dispositivos..... | 41 |
| Figura 29 – Parametrização do BLE..... | 42 |
| Figura 30 – Bloco de envio de dados via BLE..... | 42 |
| Figura 31 – Layout final do sistema supervisorio..... | 43 |

| | |
|--|----|
| Figura 32 – Fluxograma de funcionamento do sistema de acionamento | 44 |
| Figura 33 – Gesto de repouso da mão (classe 0) | 50 |
| Figura 34 – Gesto de mão fechada (classe 1)..... | 51 |
| Figura 35 – Gesto de polegar (classe 2) | 51 |
| Figura 36 – Gesto de pinça fina e ferramenta de preensão (classes 4 e 9) | 52 |
| Figura 37 – Gestos de pegada esférica e cilíndrica (classes 5 e 8) | 53 |
| Figura 38 – Gestos de pronação, supinação e mandril (classes 6, 7 e 10) | 54 |
| Figura 39 – Driver Motor DC Ponte H | 62 |
| Figura 40 – Shield PWM Servo Motor | 63 |
| Figura 41 – Controlador Thunderborg | 63 |
| Figura 42 – Controlador ZeroBorg Basic..... | 64 |
| Figura 43 – TB6612FNG | 65 |
| Figura 44 – Pi Servo pHAT..... | 65 |
| Figura 45 – Ultraborg | 66 |

LISTA DE QUADROS

| | |
|--|----|
| Quadro 1 – Etapas (rotinas) para criação do banco de dados de sinais EMG e classificação <i>offline</i> | 27 |
| Quadro 2 – Código de comunicação <i>Bluetooth</i> | 39 |
| Quadro 3 – Organização da aplicação do sistema supervisorio..... | 41 |
| Quadro 4 – Parametrização do <i>Bluetooth</i> na ESP32 | 45 |
| Quadro 5 – Código de acionamento dos motores | 46 |
| Quadro 6 – Movimentos do banco de dados KidoPro | 48 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Partes Impressas em 3D | 34 |
| Tabela 2 – Posição de repouso dos motores | 37 |
| Tabela 3 – Posições angulares dos gestos da prótese | 49 |
| Tabela 4 – Tempos de resposta da prótese | 54 |
| Tabela 5 – Modelos comerciais | 66 |
| Tabela 6 – Custos do projeto | 67 |

LISTA DE SIGLAS

| | |
|------------------|--|
| BJT | Bipolar junction transistor (Transistor de junção bipolar) |
| BLE | Bluetooth Low Energy (Bluetooth de baixo consume) |
| EMG | Eletromiografia / Eletromiográfico |
| GPIO | General Purpose Input/output (Porta programável de entrada e saída de dados) |
| I ² C | Inter-integrated circuit (Circuito Inter-integrado) |
| IDE | Integrated Development Enviroment (Ambiente de Desenvolvimento Integrado) |
| IoT | Internet of Things (Internet das Coisas) |
| MOSFET | Metal oxide semiconductor field effect transistor (Transistor de efeito de campo de metal óxido) |
| PWM | Pulse-Widht Modulation (Modulação por Largura de Pulso) |
| UART | Universal Asynchronous Receiver/Transmitter (Transmissor/Receptor Universal Assíncrono) |
| USB | Universal Serial Bus (Barramento Serial Universal) |
| SVM | Support Vector Machine (Máquina de Suporte Vetorial) |

SUMÁRIO

| | | |
|------------|---|-----------|
| 1 | INTRODUÇÃO | 10 |
| 1.1 | Objetivo Geral | 11 |
| 1.2 | Objetivos Específicos..... | 11 |
| 1.3 | Delimitação do Trabalho | 11 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 12 |
| 2.1 | Próteses | 12 |
| 2.1.1 | Próteses Passivas ou Não-Funcionais..... | 12 |
| 2.1.2 | Próteses Funcionais ou Ativas | 13 |
| 2.1.3 | Estado da arte de Próteses Mioelétricas | 14 |
| 2.2 | Eletrônica de Potência | 17 |
| 2.3 | Motores DC | 18 |
| 2.3.1 | Motor DC com escovas..... | 18 |
| 2.3.2 | Servomotor DC | 19 |
| 2.3.3 | Motor DC <i>brushless</i> | 20 |
| 2.4 | Drivers | 21 |
| 2.5 | Microcontrolador ESP32 | 22 |
| 2.6 | Raspberry Pi 3 | 23 |
| 2.7 | Comunicação <i>Bluetooth</i> | 24 |
| 2.8 | Comunicação serial | 24 |
| 3 | TRABALHOS RELACIONADOS | 26 |
| 4 | METODOLOGIA | 32 |
| 4.1 | A prótese de mão <i>Open-Source</i> | 33 |
| 4.2 | Montagem da prótese | 35 |
| 4.3 | Supervisório emulador de movimentos..... | 38 |
| 4.4 | Sistema de acionamento | 43 |
| 4.4.1 | Configuração do <i>Bluetooth Low Energy</i> na ESP32..... | 45 |

| | |
|---|-----------|
| 4.4.2 Classificação dos motores | 45 |
| 4.5 Testes de movimentos | 47 |
| 5 ANÁLISE DOS MOVIMENTOS | 50 |
| 5.1 Testes com a Raspberry..... | 55 |
| 6 CONSIDERAÇÕES FINAIS..... | 56 |
| 6.1 Sugestões para trabalhos futuros | 57 |
| REFERÊNCIAS..... | 58 |
| APÊNDICE A – DRIVERS PARA MOTORES COMERCIAIS | 62 |
| APÊNDICE B – CUSTOS DO PROJETO | 67 |

1 INTRODUÇÃO

Os sinais eletromiográficos de superfície (EMG) são coleções de sinais elétricos obtidos na superfície da pele que representam a atividade neuromuscular originada pelo músculo examinado (KILBY; HOSSEINI, 2004). Ao longo dos últimos 50 anos, a superfície eletromiográfica vem sendo estudada como um potencial controle de aplicações robóticas. Além disso, interfaces mioelétricas utilizam os sinais EMG em tempo real e de forma não invasiva para acessar a atividade muscular, a qual é utilizada em muitas aplicações entre homem e máquina, como próteses e tele operação de robôs (ISON; ARTEMIADIS, 2014).

Uma prótese de mão é capaz de compensar algumas atividades do membro perdido na vida diária. Com o aumento de pesquisas e investimentos nesta área, as próteses de mão aumentaram seu grau de liberdade, proporcionando uma ampla gama de movimentos ao amputado. Entretanto, mesmo com o aperfeiçoamento dos modelos existentes, até recentemente as próteses de membros superiores eram usadas como ferramentas auxiliares ao corpo e não como substitutas da mão ou braço amputado. A principal causa da limitação nas próteses é a própria limitação tecnológica, a qual faz com que a total funcionalidade do membro não seja alcançada na prótese (LAFFRANCHI *et al.*, 2020). Além disso, o aumento do grau de liberdade trouxe algumas desvantagens como baixa intensidade, baixa robustez e alto custo, já que a maior liberdade de movimentos requer uma quantidade maior de acionamento de motores (YONG *et al.*, 2014). Assim, o fato de ser um sistema mais complexo, mais caro e menos robusto pode ser a razão pela qual as pessoas ainda optem por usar próteses acionadas pela energia do próprio corpo (FOUGNER *et al.*, 2012).

Este trabalho teve como objetivo o desenvolvimento de um sistema capaz de acionar uma prótese de mão. Este sistema contemplou a impressão de uma prótese em 3D, a elaboração de um emulador de movimentos e o desenvolvimento de um código de programação responsável pelo controle dos acionamentos e a montagem do *hardware* do sistema. Para isso, foram usadas tecnologias de microcontroladores e técnicas abordadas na eletrônica de potência voltadas para o acionamento de motores.

A seguir, são apresentados o objetivo geral, os objetivos específicos e a delimitação deste trabalho.

1.1 Objetivo Geral

O objetivo deste trabalho foi montar e testar uma prótese de mão, acionada por um sistema de acionamento de motores capaz de realizar o movimento dos cinco dedos e se comunicar via Bluetooth com um sistema supervisor de interface com o usuário da prótese.

1.2 Objetivos Específicos

- a) montar um sistema mecânico de uma prótese de mão *open-source* impressa em impressora 3D, e integrar 5 motores para o acionamento individual de cada dedo;
- b) implementar o acionamento elétrico capaz de comandar 5 motores com sinal PWM via microcontrolador;
- c) desenvolver um sistema emulador de gesto com a função de comunicar via *Bluetooth* a intenção de movimento do amputado, controlado manualmente por via aplicativo de *smartphone*;
- d) implementar um algoritmo que receba do sistema emulador um código de gesto, interprete este gesto e acione a prótese conforme movimento desejado;
- e) implementar um algoritmo na Raspberry Pi que seja capaz de se comunicar com a ESP32 e pré-prepare o sistema para comunicação com sistemas de aquisição já existentes.

1.3 Delimitação do Trabalho

Conforme Fougner (2013), as próteses de mão miolétricas são compostas por três principais componentes: um sistema de aquisição dos sinais; um sistema de pré-processamento e interpretação/classificação; e um sistema de ativação e acionamento dos motores. Este trabalho teve como objetivo desenvolver o terceiro componente da prótese, ou seja, o sistema de acionamento dos motores. Desta forma, não serão avaliados ou discutidos os classificadores e o processamento e aquisição dos sinais EMG.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais tópicos de conhecimento necessários para o desenvolvimento do trabalho. Dentre eles estão as próteses e suas classificações, acionamentos de motores através da eletrônica de potência, as interfaces necessárias para que a prótese seja acionada e a descrição de alguns equipamentos fundamentais para a execução prática do projeto.

2.1 Próteses

Segundo o Instituto Nacional do Seguro Social (INSS) (2017), próteses são equipamentos que substituem a função de um membro e podem ser divididas em cirúrgicas ou implantáveis e em não cirúrgicas ou não implantáveis. Quanto à sua função, as próteses são classificadas em ativas ou passivas. Uma prótese ativa é capaz de proporcionar maior atividade funcional. Já as próteses passivas têm a função de melhorar o aspecto estético ou falta dele.

2.1.1 Próteses Passivas ou Não-Funcionais

Próteses não-funcionais ou passivas são próteses com finalidade estética (Figura 1), têm a função de restabelecer o aspecto externo, sem nenhuma função de movimento. Geralmente este tipo de prótese é semimóvel ou totalmente imóvel e podem ser usadas em todos os níveis de amputação (MOREIRA, 2015).

Figura 1 – Prótese estética



Fonte: Medical Expo (2020).

Segundo Fougner (2013), as próteses estéticas ainda são as mais populares entre os usuários devido a seu baixo custo e longa duração.

2.1.2 Próteses Funcionais ou Ativas

As próteses funcionais ou ativas são próteses acionadas pelo próprio usuário. Este tipo de prótese demanda um intenso treinamento para que se adquira o controle de diferentes movimentos da prótese (MONTEIRO *et al.*, 2017). Conforme Rommers *et al.* (1997), as próteses funcionais são divididas em: próteses mecânicas ou com fonte de energia interna (ativadas pelo movimento do próprio corpo), próteses com fonte de energia externa e próteses híbridas. Na prótese mecânica (Figura 2) a energia aplicada é proveniente do próprio corpo do paciente, transmitida através de cabos para o dispositivo terminal. A ação do membro é feita através de um cinto de segurança aliado a um sistema de cabos.

Figura 2 – Prótese de mão mecânica



Fonte: Bebionic (2020).

Nas próteses com fonte de energia externa, a energia utilizada é externa ao corpo do paciente, normalmente através de baterias. Estas próteses são controladas pela estrutura natural remanescente ou através de atividade eletromiográfica fornecida pelo grupo muscular escolhido (ROMMERS *et al.*, 1997).

Nas próteses híbridas, segundo Monteiro *et al.* (2017), o sistema funciona como uma junção da prótese mecânica com a prótese ativada por fonte de energia externa. Normalmente são utilizadas em amputações de alto nível (igual ou superior ao cotovelo). Este tipo de prótese permite o controle de duas articulações ao mesmo tempo.

2.1.3 Estado da arte de Próteses Mioelétricas

As próteses mioelétricas são um exemplo de prótese ativa, conforme Figura 3. Uma prótese mioelétrica usa os músculos de um membro residual para controlar suas funções. Um ou mais sensores recebem os sinais provenientes do músculo do membro residual quando o usuário executa algum tipo de movimento neste membro. Os sensores enviam o sinal recebido para um controlador, o qual converte os dados em comandos para os motores elétricos e estes movem as juntas da prótese (OTTOBOCK, 2013).

Figura 3 – Prótese Ativa Comercial



Fonte: Ottobock (2013).

Embora existam divergências entre fabricantes, as próteses mioelétricas consistem em três principais componentes: uma interface responsável pela aquisição dos sinais mioelétricos através de sensores ligados em sua superfície, uma fonte de alimentação e um controlador; uma extensão flexível do pulso, responsável por conectar a interface à prótese; e, por último, a prótese propriamente dita - que é composta pelo controlador que interpreta o sinal da interface e o converte em *Pulse-Width Modulation* (PWM ou Modulação por Largura de Pulso) - e os motores dc que recebem o sinal convertido e são responsáveis por movimentar a prótese (LAFFRANCHI *et al.*, 2020).

Segundo Fougner (2013), o acionamento de mãos e cotovelos por fonte de potência externa ao corpo tem sido possível desde a década de 1960 e, posteriormente, se possibilitou o acionamento de pulso e mão em conjunto. Este tipo de acionamento é chamado de prótese multifunção. Ao longo da última década, as

pesquisas em torno de próteses de membros superiores tiveram um crescimento significativo, onde são encontradas em torno de 4000 publicações contendo o termo “prótese de mão”. Este crescimento fez com que muitas próteses de mão antropomórfica multi-articuladas fossem introduzidas ao mercado.

As principais propriedades dos sistemas de controle mioelétrico atuais são: a camada de pré-processamento, a camada de interpretação de intenção e o perfil de ativação. A camada de pré-processamento, resumidamente, é a coleção de informações provenientes do usuário, ou seja, uma aplicação implementada com o uso de sensores e de processamento de sinais. A camada de Interpretação de Intenção é o sistema que interpreta a intenção de movimento do usuário baseada na informação disponibilizada pela camada de pré-processamento. Por fim, a Ativação é uma propriedade da camada de saída do sistema e se divide entre controle proporcional e controle “*on-off*” (FOUGNER, 2013).

De acordo com Fougner *et al.* (2012), o controle sequencial é o mais utilizado atualmente em próteses multifunção. Esta estratégia permite que o usuário intercale o controle em vários estados como força e velocidade. Mesmo sendo um controle sofisticado, os usuários de próteses multifunções a descrevem como pesada e lenta. De um modo geral, o alto custo, a complexidade, a falta de robustez e a necessidade de um controle mais intuitivo faz com o que as pessoas ainda optem por próteses convencionais ou acionadas pelo próprio corpo.

A Figura 4 mostra um exemplo de dois modelos de próteses multiarticulares existentes comercialmente, que são a iLimb Pulse e a BeBionic, respectivamente.

Figura 4 – Próteses Multiarticulares

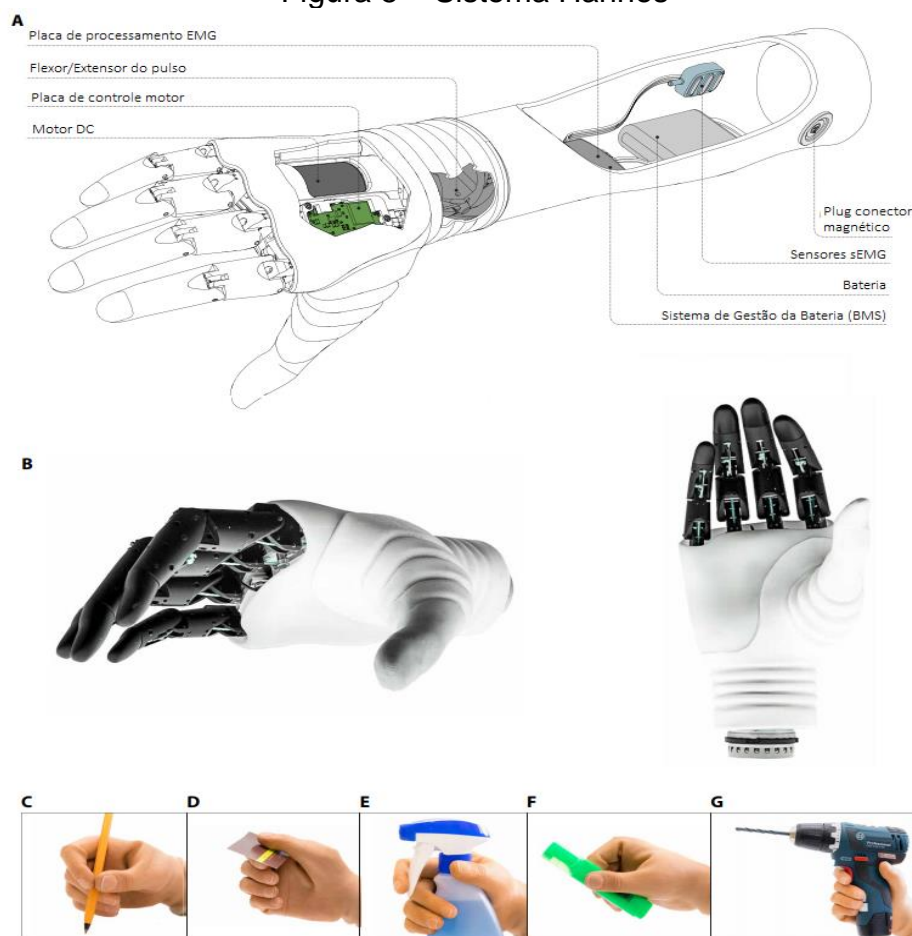


Fonte: Premium Prosthetics (2020).

Um dos estudos mais recentes sobre próteses de mão mioelétricas é a prótese Hannes, desenvolvida por pesquisadores da *IIT's Rehab Technologies Lab*. O objetivo do projeto foi desenvolver uma prótese altamente fiel aos movimentos e ao toque humano, já que a maioria das mãos robóticas hoje no mercado não são efetivas em atender às demandas de massa, tamanho e/ou funcionalidade energética para o uso prático.

Para tanto, a mão artificial Hannes foi desenvolvida com três principais componentes físicos: uma mão prótica mioelétrica poli articulada que explora um mecanismo diferencial sub atuado; um módulo passivo de flexão-extensão do pulso; e uma interface/controlador mioelétrico que inclui dois sensores eletromiográficos (sEMG), pacote de bateria e um controle eletrônico. Estes três componentes, conforme Figura 5, foram desenvolvidos para realizar antropomorfismo, performance biomimética e toque/pegada humana (LAFFRANCHI *et al.*, 2020).

Figura 5 – Sistema Hannes



(A) Arquitetura: o motor DC e sua placa de controle são embutidas dentro da mão mioelétrica, enquanto os sensores sEMG e o controlador são inseridos dentro do soquete. O F/E do pulso é colocado entre estes dois módulos. (B) Vista da mão Hannes sem luva. (C a G) Mão Hannes com luvas realizando movimentos.

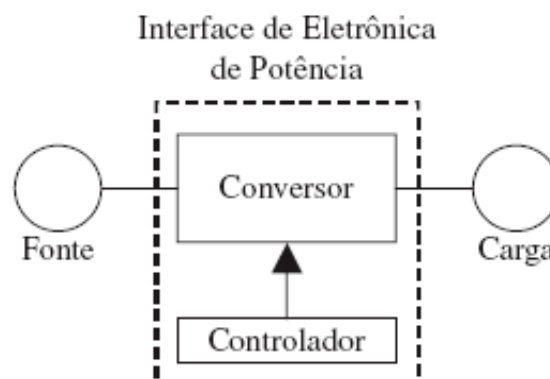
Fonte: Adaptado de Laffranchi *et al.* (2020).

O projeto de mão mioelétrica Hannes, de acordo com Laffranchi *et al.* (2020), possui uma interface intuitiva onde dois sensores sEMG são usados para detectar a atividade muscular do coto do amputado, bem como a contração dos músculos flexores e extensores, permitindo o fechamento e abertura do dispositivo, respectivamente. Os dois sensores são localizados dentro do soquete e realizam uma interface com a mão Hannes através de módulos eletrônicos de controle embutidos na “Placa de Processamento EMG”, a qual processa o sinal e o retorna para a placa de controle dos motores. Como alimentação, o sistema Hannes possui um pacote de baterias de 12 V – 1300 mAh, que também é localizado no interior do soquete e que foi projetada para um dia de duração. O controlador do sistema é proporcional, ou seja, este envia a referência de velocidade para a mão, aumentada proporcionalmente com a ativação muscular.

2.2 Eletrônica de Potência

A tecnologia habilitadora capaz de fornecer uma interface necessária entre uma fonte elétrica e uma carga elétrica é chamada de Eletrônica de Potência, conforme demonstra a Figura 6. Em geral, a fonte elétrica e a carga diferem em amplitude de tensão, frequência ou número de fases. A interface de eletrônica possibilita a transferência de potência da fonte para a carga, convertendo tensões e correntes de uma forma em outra, fazendo com que fonte e carga troquem funções. Um exemplo de aplicação da eletrônica de potência são os acionamentos elétricos de velocidade ajustável (MOHAN, 2014).

Figura 6 – Interface entre fonte e carga.



Fonte: Mohan (2014).

Segundo Hart (2012), a eletrônica de potência está inclusa tanto nos circuitos de aplicação de alta potência - como linhas de transmissão de potência DC – quanto nos circuitos de equipamentos do cotidiano – computadores, carregadores de bateria, ferramentas elétricas portáteis. As aplicações típicas de eletrônica de potência incluem conversão CA em DC, conversão de DC em CA e conversão de uma tensão DC não regulada em uma tensão DC regulada.

De acordo com o fabricante de placas de controle de equipamentos eletrônicos SparkFun (2020), o controlador do servo precisa enviá-lo um sinal preciso, geralmente em uma janela de 20 ms entre cada pulso. A largura do sinal enviado pelo controlador é o que determina a posição a ser atingida pelo servo. Este tipo de controle é chamado de *Pulse Width Modulation* (PWM). A grande maioria dos servomotores de pequeno porte, também chamados de *hobby servos*, possuem uma tensão de acionamento na faixa de 5 V.

2.3 Motores DC

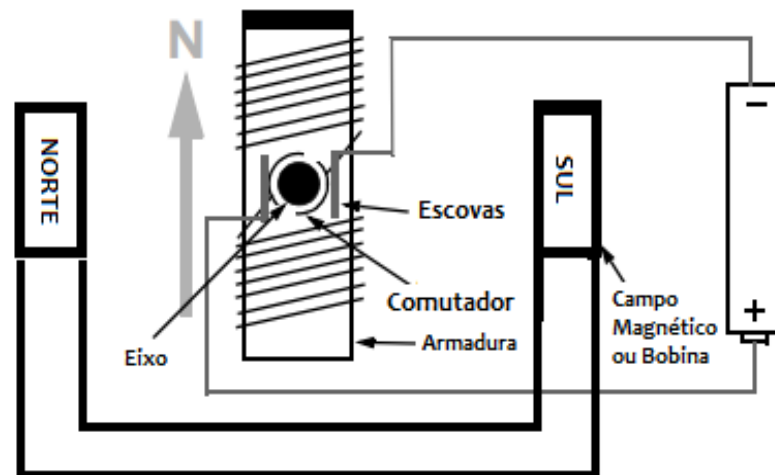
Segundo Monteiro *et al.* (2017), o princípio básico de um motor de corrente contínua é permitir o fluxo de corrente elétrica por uma bobina, criando um campo magnético. As máquinas DC em geral possuem ampla facilidade em serem controladas, fazendo com que seus sistemas sejam usados em aplicações que exigem uma variada faixa de velocidades ou de controle preciso da saída do motor (UMANS, 2014). A troca de energia entre o rotor e o estator pode ser realizada com ou sem escovas. Nos tópicos seguintes foram descritas as características de motores DC com escovas, servomotores DC e motores DC *brushless*.

2.3.1 Motor DC com escovas

O motor DC com escovas trabalha em conexão direta com uma fonte de tensão em corrente contínua. Estes motores podem ser divididos em duas categorias: ímã-permanente e eletromagnético. Os motores DC com excitação por ímã-permanente são os mais usados em aplicações como ferramentas elétricas e indústria automobilística. Dependendo da configuração da armadura, estes motores são divididos entre modelos com ranhuras, sem ranhuras e sem núcleo. Em contrapartida, os motores DC eletromagnéticos obtêm seu fluxo aplicando corrente em uma bobina

eletromagnética. Estes motores são usados em aplicações de média e alta potência (ASPINA, 2020). A Figura 7 mostra a construção de um motor DC com escovas.

Figura 7 – Motor DC de dois polos



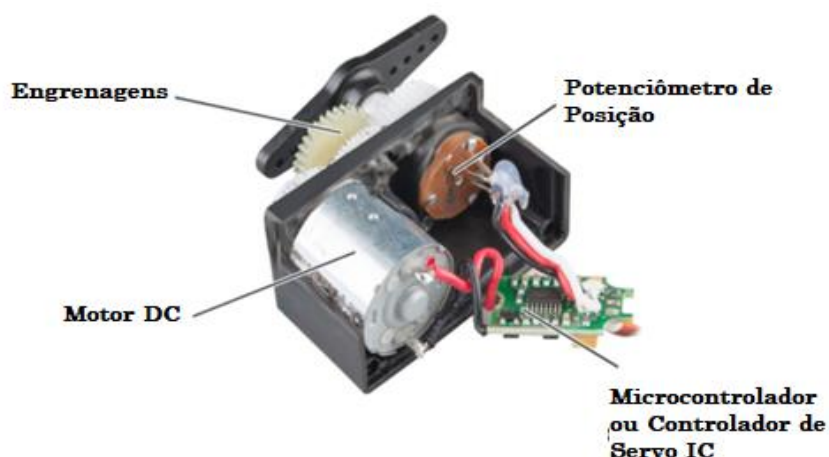
Fonte: Microchip (2004).

2.3.2 Servomotor DC

Um servomotor DC é composto por um motor DC, um potenciômetro de realimentação usado como transdutor de posição, uma caixa de engrenagem e um circuito eletrônico do acionamento e *loop* de controle. Semelhante ao motor DC, o servomotor tem seu estator constituído por uma estrutura cilíndrica e o ímã é acoplado ao interior de sua armação (SILVEIRA, 2012).

O motor DC é acoplado a uma caixa de engrenagens e um eixo de saída de modo a aumentar sua velocidade e torque. O circuito de controle tem a atribuição de interpretar os sinais enviados por um controlador. O potenciômetro tem a função de enviar um *feedback* para o circuito de controle, de modo a monitorar a posição atingida pelo eixo de saída. Normalmente, os pequenos servomotores DC possuem saída a três pinos. As cores dos cabos podem variar entre fabricantes, mas geralmente a ordem de ligação dos pinos permanece a mesma (SPARKFUN, 2020). A Figura 8 mostra uma vista interna dos componentes do servomotor DC.

Figura 8– Servomotor DC

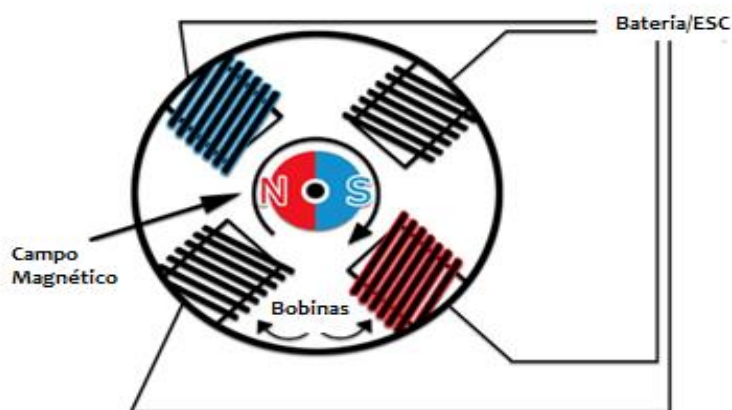


Fonte: SparkFun (2020).

2.3.3 Motor DC *brushless*

Ao contrário dos motores DC com escovas, o motor DC sem escovas (*brushless*) necessita de um circuito *driver* para seu acionamento. Este tipo de motor, por não contar com as escovas, precisa de um circuito externo para controlar sua polaridade. Assim como o motor sem escovas, o motor com escovas também conta um rotor e um estator, conforme Figura 9. Por não haver comutador neste motor, seu controle de velocidade é feito através de modulação por largura de pulso (PWM). (HARRINGTON; KRONINGER, 2013).

Figura 9 – Motor DC Sem Escovas



Fonte: Dronenodes (2020).

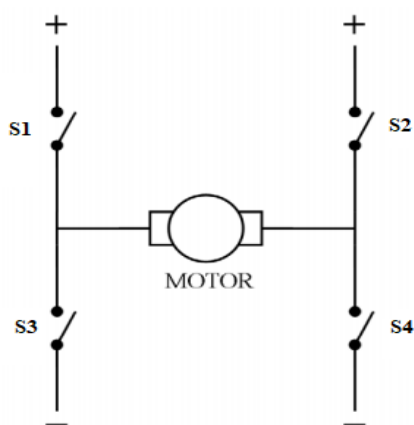
Além do estudo de motores de corrente contínua também foram estudados seus tipos de acionamento, descritos no tópico a seguir.

2.4 Drivers

Drivers e controladores para motores são dispositivos capazes de controlar velocidade, torque e posição. O drive é responsável por modificar o sinal de entrada do circuito de modo a obter-se a saída desejada para o motor. Geralmente, o circuito de controle e o circuito de drive são integrados em um mesmo dispositivo e são divididos em três classes: AC, DC e *Servodrive*. Cada tipo recebe um determinado sinal de entrada e converte esse sinal de acordo com o motor a ser acoplado na saída (THOMAS, 2020). A maioria dos drivers é composto internamente por uma ponte H.

Segundo Sieben (2003) uma ponte H é um circuito de potência que permite controlar a velocidade e a direção de um motor DC. Estes motores são frequentemente controlados por algum tipo de microcontrolador, os quais provém as instruções de posição e velocidade. Entretanto, os microcontroladores não são capazes de prover a potência requerida pelo motor. A função da ponte H é receber as instruções do controlador e amplificá-las para o motor. Conforme apresentado na Figura 10, o diagrama eletrônico deste circuito forma uma configuração semelhante a uma letra “H”, onde as chaves simbolizam chaves eletrônicas como transistores de efeitos de campo (MOSFET) ou transistores de junção bipolar (BJT) (SIEBEN, 2003).

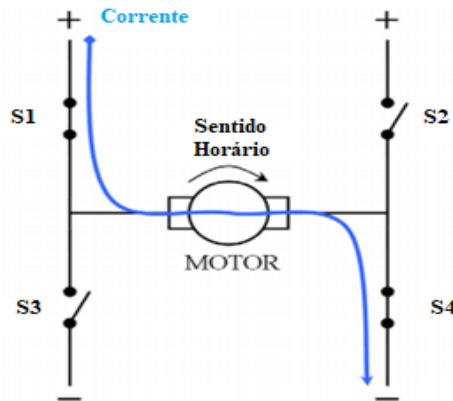
Figura 10 – Topologia do Circuito Ponte H



Fonte: Sieben (2003).

Se for necessário mover o motor no sentido horário, as chaves 1 e 4 devem ser fechadas para acionar o motor, conforme mostrado na Figura 11.

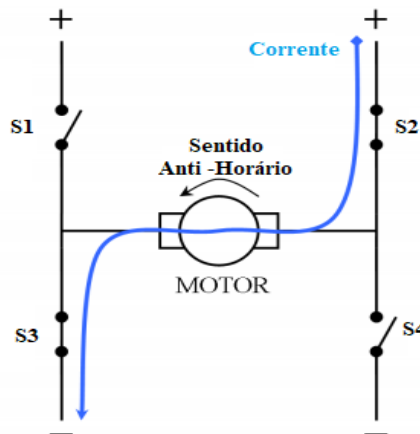
Figura 11 – Topologia do Circuito Ponte H – Sentido Horário



Fonte: Sieben (2003).

Entretanto, se o objetivo for acionar o motor no sentido anti-horário, as chaves S2 e S3 devem ser fechadas a fim de inverter o sentido da corrente, conforme Figura 12.

Figura 12 – Topologia do Circuito Ponte H – Sentido Anti-Horário



Fonte: Sieben (2003).

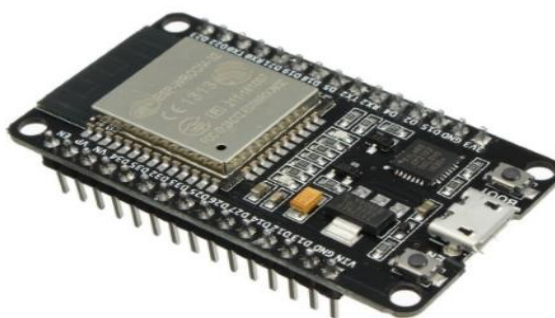
Os modelos de drivers pesquisados para o desenvolvimento deste trabalho estão descritos no Apêndice A.

2.5 Microcontrolador ESP32

O microcontrolador ESP32 (Figura 13) é um chip de controle com módulos *Bluetooth* e *Wi-Fi* de 2.4 GHz, com baixo consumo de energia. A placa conta com antena embutida, interface *USB-serial* e regulador de tensão 3,3 V. Sua programação

pode ser feita em LUA ou usando a IDE do Arduino. Este controlador também é usado para aplicações IoT, acesso remoto, *webservers* e *dataloggers* (ESPRESSIF SYSTEMS, 2020).

Figura 13 – Microcontrolador ESP32



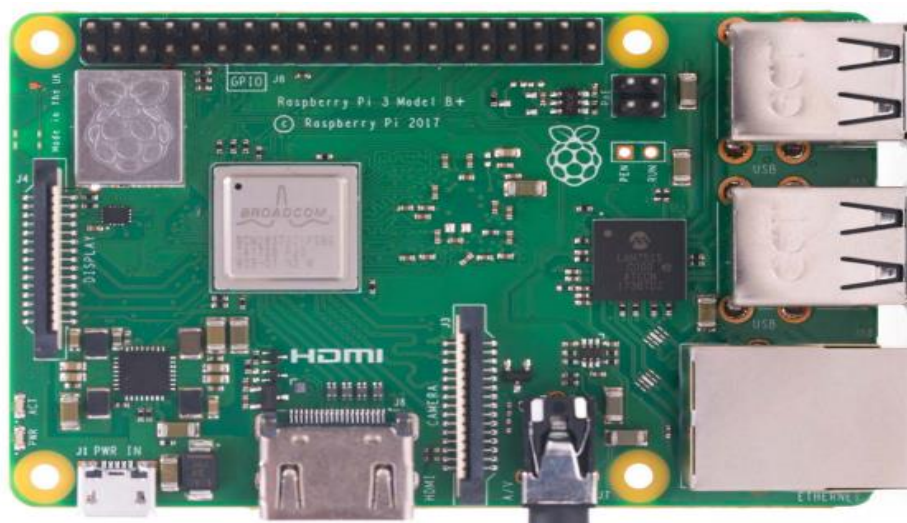
Fonte: Filipeflop (2020).

Além do microcontrolador ESP32, utilizado neste trabalho, a Raspberry Pi 3 também foi estudada a fim de tornar a prótese integrável com outros dispositivos de aquisição de sinais.

2.6 Raspberry Pi 3

A Raspberry Pi 3 modelo B+ Anatel (Figura 14) é um minicomputador que roda sistemas operacionais com distribuição Linux, como Raspbian e Ubuntu. Nesta versão, a Raspberry possui processador de 1,4 GHz, 1 GB de memória e suporta redes wireless. Para gravar o sistema operacional e servir como memória do sistema, a Raspberry necessita de um cartão micro SD. A alimentação da placa é 5 V e pode ser conectada via micro USB ou GPIO (RASPBERRY, 2021).

Figura 14 – Raspberry Pi 3 B+



Fonte: Raspberry (2021).

Após apresentar os tipos de controle utilizados, os próximos tópicos abordam os protocolos de comunicação aplicados: comunicação *Bluetooth* e serial.

2.7 Comunicação *Bluetooth*

Bluetooth é uma comunicação *wireless* de baixo alcance, a qual permite que dispositivos como celulares, computadores e outros periféricos transmitam dados ou voz sem o uso de cabos a uma curta distância. A comunicação *Bluetooth* tem o propósito de substituir os cabos que normalmente conectam os dispositivos, mantendo uma comunicação segura e confiável (CÂMARA, 2012).

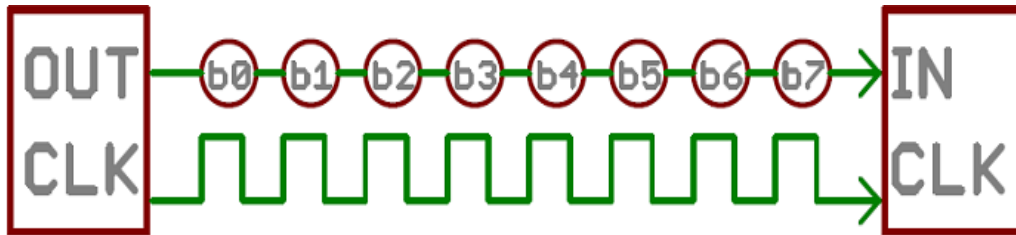
O sistema *Bluetooth* utiliza frequência de rádio de onda curta (2.4 GHz), permitindo a comunicação entre dispositivos próximos com baixo consumo de energia. A comunicação acontece através de uma rede chamada piconet, a qual podem ser conectados, no máximo, oito dispositivos. Em geral, este tipo de comunicação tem o alcance de 1 a 10 metros, dependendo do dispositivo (CÂMARA, 2012).

2.8 Comunicação serial

Interfaces paralelas são capazes de transferir múltiplos *bits* ao mesmo tempo. Geralmente, este tipo de comunicação requer barramentos de dados, transmitindo através de oito, dezesseis ou mais fios. Os dados são transferidos em grandes

pacotes de “uns” e “zeros”. Já as interfaces seriais transmitem seus dados à taxa de um *bit* por segundo. A comunicação serial é capaz de ser feita através de um único fio, nunca ultrapassando quatro fios. A Figura 15 mostra um exemplo de interface serial a dois fios, transmitindo um *bit* a cada pulso de *clock* (SPARKFUN, 2021).

Figura 15 – Exemplo de interface serial



Fonte: Sparkfun (2021).

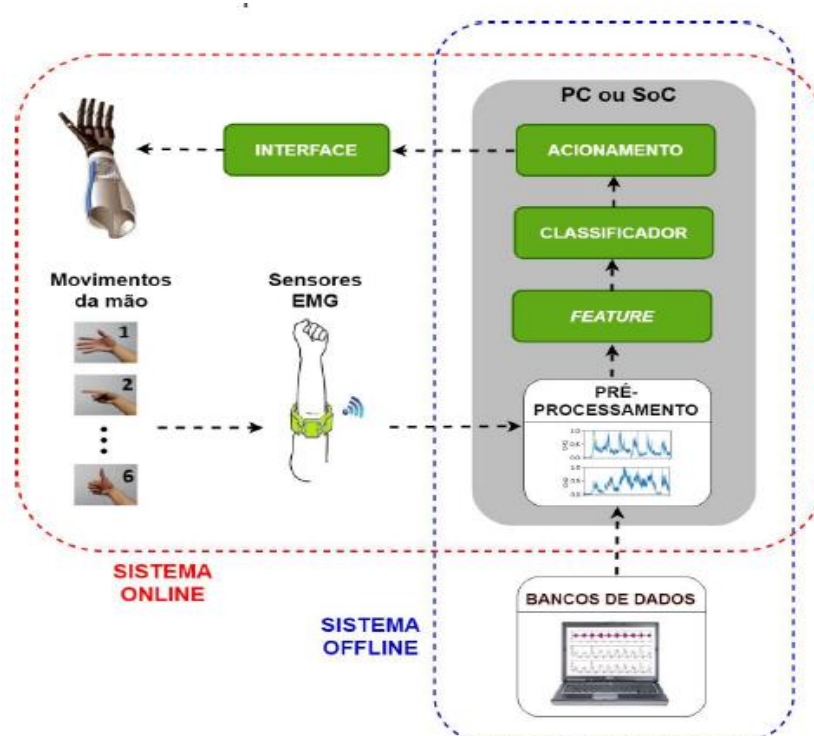
Após esta fundamentação teórica abordando próteses, acionamento de motores e microcontroladores, o próximo capítulo apresenta os trabalhos relacionados ao acionamento de próteses e temas adjacentes.

3 TRABALHOS RELACIONADOS

De modo a desenvolver um sistema capaz de interpretar os movimentos recebidos do sistema de aquisição de sinais e acionar a prótese com os movimentos pré-programados, tomou-se como base de estudo a tese de doutorado *Investigação de Diferentes Métodos e Recursos para Controle de Prótese de Mão Através da Classificação de Sinais EMG Via Aprendizado de Máquina*, de autoria Souza (2021). A arquitetura do trabalho desenvolvido por Souza (2021) contemplou todas as etapas do acionamento da prótese de mão: integração de dispositivos de aquisição de sinais eletromiográficos (EMG), plataforma para implementação do classificador em tempo real e interface para acionamento elétrico de servomotores para uma prótese de mão *open-source*.

A Figura 16 demonstra os componentes implementados na tese, como componentes mecânicos (prótese), o *hardware* utilizado (sistema de aquisição de sinais, sensores EMG de superfície, *driver/interface* para acionamento elétrico da prótese), os elementos/componentes de *software* (pré-processamento, extração de *features*, classificadores), os sinais EMG a partir de movimentos/gestos de mão realizados e bases de dados de sinais EMG (SOUZA, 2021).

Figura 16 – Diagrama de blocos das etapas para implementação de prótese de mão



Fonte: Souza (2021).

O objetivo do sistema *offline* é o reconhecimento de padrões de movimento do segmento mão-braço através de sinais EMG. Segundo Souza (2021), o sistema é composto pelos seguintes elementos: bancos de dados de sinais eletromiográficos, componentes de pré-processamento e segmentação de sinais, extração de características (*features*), redução de dimensionalidade e classificação.

De acordo com Souza (2021), o sistema *offline* foi dividido em duas etapas, sendo a primeira etapa uma série de cinco ensaios para avaliação do desempenho das arquiteturas implementadas a partir da alteração e/ou incorporação de diferentes recursos ou classificadores, utilizando dois bancos de dados NinaPro. A segunda etapa foi realizar o processo para aquisição e geração de um novo banco de dados de sinais EMG a partir de movimentos de mão de voluntários locais e posteriormente sua classificação baseada nos gestos realizados.

O sistema *online* teve como objetivo realizar a classificação dos gestos/movimentos de mão e gestos de preensão através de ensaios para aquisição de sinais EMG em voluntários. O Quadro 1 apresenta as etapas implementadas.

Quadro 1 – Etapas (rotinas) para criação do banco de dados de sinais EMG e classificação *offline*

- Inicializa o(s) dispositivo(s) de aquisição de sinais EMG
- Realiza a aquisição dos sinais EMG (dados brutos)
- Rotula os dados
 - Método TKEO
 - Formatação do arquivo
- Separa os dados (treinamento, validação e teste)
- Pré-processamento
 - Ajuste da escala dos sinais EMG
 - Filtragem
 - Retificação
 - Normalização
 - Segmentação
- Calcula as *features* no domínio do tempo
- Redução da dimensionalidade (LDA ou PCA)
- Treina o(s) classificador(es)
- Testa (*offline*) o(s) classificador(es)
- Salva as configurações e parâmetros do(s) classificador(es)

Fonte: Souza (2021).

De acordo com os testes realizados por Souza (2021), a arquitetura mais bem avaliada (em termos de classificação e portabilidade) foi implantada em uma plataforma *System on Chip* (SoC) de baixo custo. Os testes do sistema online foram realizados utilizando uma Raspberry Pi 3 B+ como classificador e um Arduino Mega como interface. Os sinais EMG de leitura dos braceletes foram enviados via comunicação Bluetooth para a Raspberry, sendo pré processados e posteriormente enviados para o Arduino via comunicação Serial.

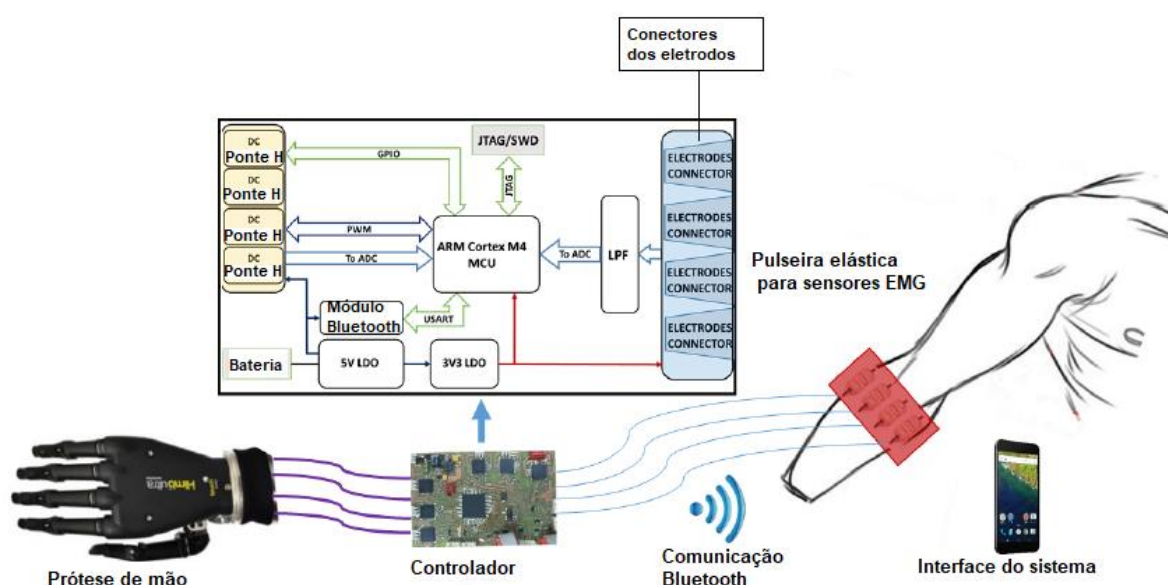
Ainda no campo da eletromiografia, Atzori e Müller (2015), descrevem o banco de dados NinaPro como o repositório atual com maior número de testes em voluntários do mundo. Este banco de dados possui como *setup* de aquisição uma série de sensores com o objetivo de mapear correspondentes musculares e gravar a cinemática e a dinâmica dos movimentos. Segundo Atzori e Müller (2015), a cinemática da mão é medida usando uma luva *CyberGlove* de 22 sensores, capazes de detectar movimentos das mãos através de medições de ângulo muito precisas. Um inclinômetro de 2 eixos modelo Kübler IS40 foi fixado no pulso do sujeito para medir sua orientação com um range de 120° e resolução menor que 0,15°. A dinâmica da mão é medida com um sensor de força linear, baseado na tecnologia de *strain gages*. A atividade muscular é medida através de uma combinação de eletrodos EMG dos modelos OttoBock MyoBock 13E200-50 e dos modelos *wireless* da Delsys Trigno Wireless System.

No segmento de próteses *open-source*, Cognolato *et al.* (2018) realizaram testes de controle de próteses de mão 3D em sujeitos com amputação transradial, usando um sistema de controle baseado em reconhecimento de gestos. Cognolato *et al.* (2018) recrutaram dois sujeitos do sexo masculino como voluntários para a pesquisa. O nível de amputação dos sujeitos era muito diferente um do outro: o sujeito 1 tinha uma amputação no nível do pulso, enquanto o sujeito 2 tinha uma amputação transradial em um nível mais alto do braço. Para poderem comparar os resultados dos testes entre os dois sujeitos, o comprimento do membro residual foi normalizado de acordo com o comprimento do membro intacto. Como *setup*, Cognolato *et al.* (2018) usaram a prótese 3D HANDi Hand, eletrodos sEMG Myo, da Thalmic Labs, microcontrolador Arduino Leonardo, *software* MyoDuino, soquete fabricado pelos autores, e um laptop para aquisição dos sinais. O protocolo de aquisição consistiu em pegar objetos repetidas vezes. O controle em tempo real da prótese foi baseado na classificação de gestos de mão performado pelo classificador do Myo. Assim, cada

gesto classificado pelo Myo disparava um movimento específico para a HANDi Hand em tempo real. Segundo Cognolato *et al.* (2018), o resultado de sua pesquisa mostra a efetividade de próteses e *setups* de baixo custo compostos por próteses 3D e classificados de gestos EMG (aproximadamente 100 vezes menos onerosos que as próteses comerciais).

Benatti *et al.* (2017) analisaram os sinais de aquisição EMG e integraram ao sistema de reconhecimento de gestos SVM (*Support Vector Machine* – Máquina de Suporte Vetorial) com controle em tempo real. Para implementar o sistema à arquitetura demonstrada na Figura 17, Benatti *et al.* (2017) utilizaram o seguinte *setup*: (1) pulseira elástica com os sensores EMG; (2) um controlador responsável pela aquisição dos dados, classificação, atuação da prótese e comunicação Bluetooth; (3) uma prótese de mão poli articulada e (4) um gateway pessoal para aquisição de dados, algoritmo para treinar o reconhecimento de gestos e customização dos parâmetros do sistema.

Figura 17 – Arquitetura do Sistema

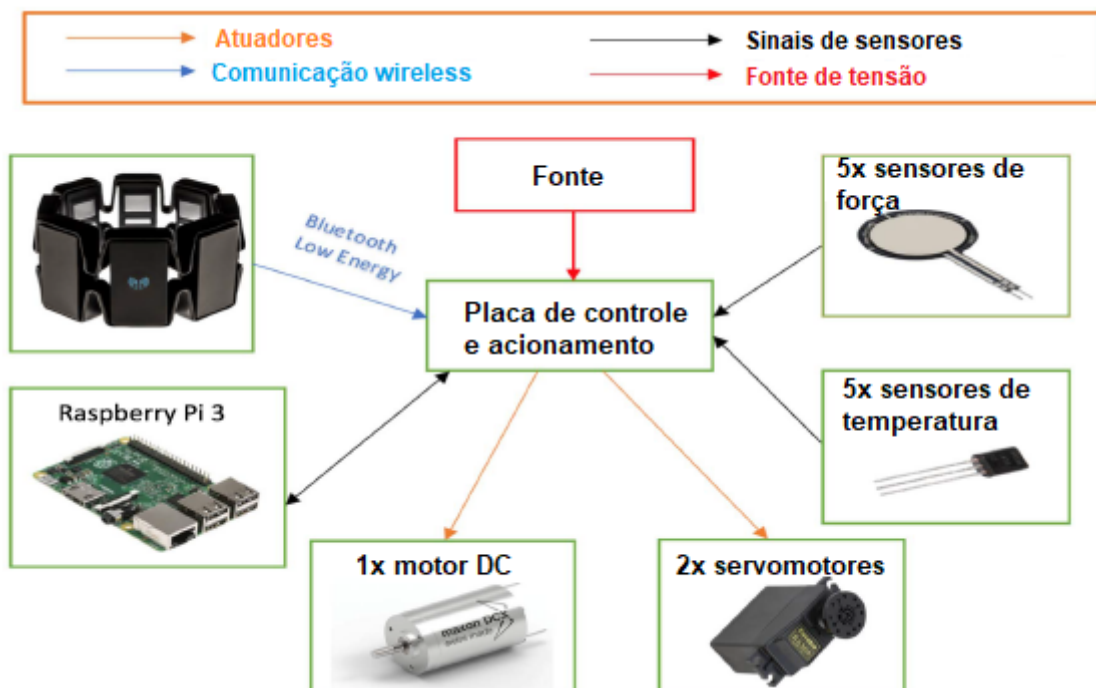


Fonte: Adaptado de Benatti *et al.* (2017).

Benatti *et al.* (2017) testaram o sistema em 5 pessoas saudáveis e 4 pessoas amputadas com 1/3 de amputação distal. O sistema proposto pelos autores apresentou uma taxa de erro médio de classificação de gestos de 1 para 60, para todas as pessoas testadas.

Gaetani *et al.* (2018) apresentaram uma prótese implementada - chamada de Adam's Hand - por um mecanismo sub atuado, formado por engrenagens rígidas e compactas e atuado por um único motor DC. O *setup* utilizado para o acionamento da prótese era composto de (1) 5 sensores de temperatura LM35 e 5 sensores de pressão FSR400, um para cada dedo; (2) Sistema de aquisição de sinais Myo, da Thalmic Labs; (3) Microcontrolador Arduino para receber e processar os sinais de instrumentação do sistema e atuar os motores; (3) 2 servomovores Futaba S3305, usados para articular o punho da prótese e (4) o motor DC Maxon DCX 19 S DC, usado para abrir/fechar a prótese. O Arduino recebia os dados via *Bluetooth Low Energy* do sistema Myo, usando um módulo BLE HM11 e trocava dados com a Raspberry 3 usando comunicação USB. A Raspeberry Pi 3 era usada para gerenciar um display LCD *touch-screen*, o qual servia como interface para gerenciar uma série de aspectos do sistema, como autonomia, dados dos sensores e conexão wireless. Ainda, a Raspberry enviava os dados via nuvem para um servidor web dedicado, onde poderiam ser monitorados em tempo real pela equipe ortopédica responsável pelo usuário da prótese. A prótese implementada por Gaetani *et al.* (2018) foi capaz de realizar 15 gestos de mão e dois movimentos de pulso. A Figura 18 demonstra o sistema implementado por Gaetani *et al.* (2018).

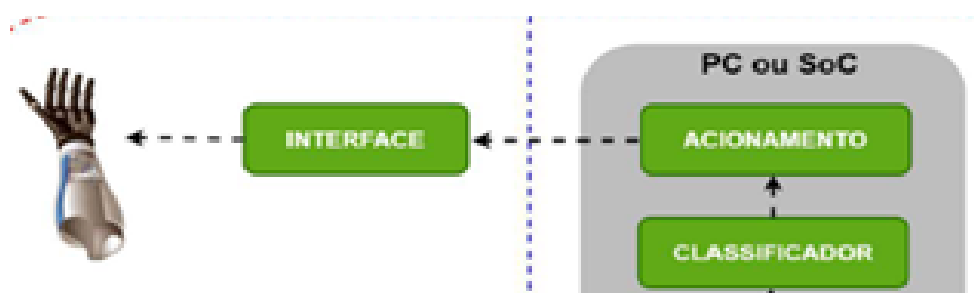
Figura 18 – Diagrama do sistema da prótese Adam's Hand



Fonte: Adaptado de Gaetani *et al.* (2018).

Conforme descrito anteriormente, este trabalho contempla a etapa de acionamento da prótese. Os estudos de bancos de dados de sinais EMG não foram abordados nesta pesquisa, sendo o trabalho de Souza (2021) utilizado como orientação. Todos os testes, códigos e padrões de movimento foram desenvolvidos de modo que a prótese pudesse ser testada na implementação de Souza (2021). A Figura 19 mostra a etapa de estudo realizada neste trabalho.

Figura 19 – Objeto de estudo para trabalho de conclusão de curso



Fonte: Adaptado de Souza (2021).

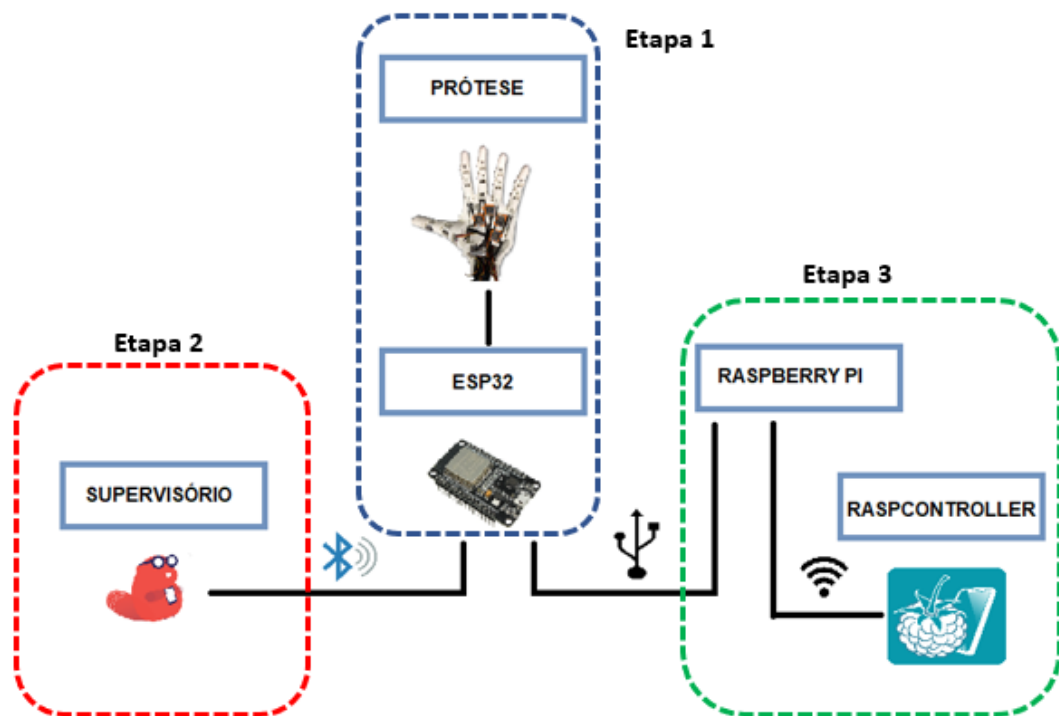
Apesar deste trabalho não se enquadrar no campo de pesquisas de sinais EMG diretamente, os artigos, teses e trabalhos publicados nesta área foram usados como base para o desenvolvimento e o ensaio dos gestos da prótese. Desta forma, foi possível seguir uma linha de trabalho coerente com as atuais pesquisas que estão sendo desenvolvidas neste ambiente.

4 METODOLOGIA

Neste capítulo são apresentadas as etapas para o projeto e desenvolvimento de acionamento elétrico para prótese de mão *open-source*.

A Figura 20 demonstra um diagrama de funcionamento do sistema de acionamento da prótese. Conforme citado anteriormente, o sistema projetado não é capaz de efetuar a aquisição e a classificação dos sinais EMG. Portanto, para que o sistema pudesse ser testado com fidelidade aos trabalhos relacionados com a pesquisa - citados anteriormente no capítulo 3 deste trabalho - um sistema supervisor via aplicativo Android foi desenvolvido na plataforma Thunkable (THUNKABLE, 2021). Posteriormente, testou-se a integração deste sistema com o aplicativo RaspController (EGAL NET, 2021), para testes com a Raspberry Pi 3 B.

Figura 20 – Diagrama de funcionamento do sistema



Fonte: Elaborado pelo autor (2021).

Conforme Figura 20, o sistema de acionamento da prótese foi feito em três principais etapas, sendo elas:

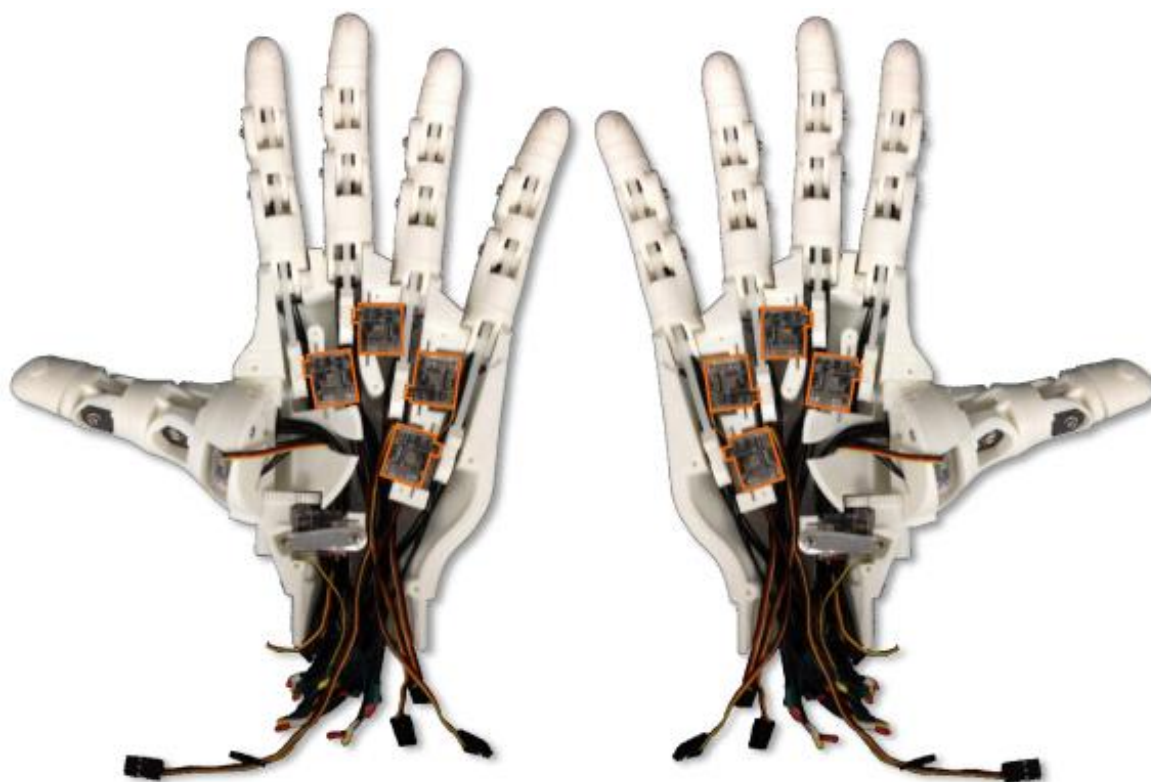
- Etapa 1: montagem da prótese e teste dos movimentos com a ESP32;
- Etapa 2: desenvolvimento do sistema supervisor e comunicação com a ESP32 via protocolo BLE;
- Etapa 3: integração do sistema com a Raspberry Pi 3 B.

As etapas descritas acima serão detalhadas nos próximos tópicos deste capítulo.

4.1 A prótese de mão *Open-Source*

Para a realização da padronização e dos testes de movimento, foi realizada a impressão de uma prótese em impressora 3D. Esta prótese é chamada de HANDi Hand e foi projetada na Universidade de Aberta no Canadá como uma plataforma de robótica de código aberto para pesquisas de aprendizado de máquina (*machine learning*). Através do site blincdev.ca, são disponibilizados todos os arquivos .stl para impressão 3D e também um manual com todos os componentes e instruções de montagem da prótese. Segundo Brenneis e Austin (2017), o modelo de prótese HANDi Hand - conforme Figura 21 - demanda um total de 30 horas para ser construído.

Figura 21 – Instalação dos motores na prótese



Fonte: Brenneis e Austin (2017, p. 17).

A impressão da prótese é dividida em 15 partes e tem suas especificações de impressão e tempo, conforme Tabela 1.

Tabela 1 – Partes Impressas em 3D

| Nome | Especificações de Impressão | Tempo de Imp. Estimado | Peso estimado |
|-------------------------------|--|------------------------|---------------|
| Palma Dorsal | Camada de 0,2mm, 10% de preenchimento; imprimir com raft | 6h0min | 64 |
| Palma Ventral | Camada de 0,2mm, 10% de preenchimento | 1h55min | 22 |
| Tampa de Rosca do Polegar | Camada de 0,2mm, 10% de preenchimento | 0h 15min | 2 |
| Indicador Completo | Camada de 0,2mm, 10% de preenchimento | 1h10min | 9 |
| Médio Completo | Camada de 0,2mm, 10% de preenchimento | 1h10min | 9 |
| Anelar Completo | Camada de 0,2mm, 10% de preenchimento | 1h10min | 9 |
| Mínimo Completo | Camada de 0,2mm, 10% de preenchimento | 1h10min | 9 |
| Polegar Completo | Camada de 0,2mm, 10% de preenchimento | 2h30min | 25 |
| Polegar Aduável | Camada de 0,2mm, 10% de preenchimento | 3h0min | 33 |
| Breadboard | Camada de 0,1mm, 10% de preenchimento | 0h25min | 3 |
| Hub Conector | Camada de 0,2mm, 10% de preenchimento | 0h15min | 3 |
| Ativador Potenciômetro (15un) | Camada de 0,1mm, 10% de preenchimento | 0h10min | 1 |
| Marcador de Posição (6un) | Camada de 0,2mm, 10% de preenchimento | 0h10min | 2 |
| Carretel do Servo | Camada de 0,1mm, 10% de preenchimento | 0h25min | 4 |
| Engrenagem Dentada do Servo | Camada de 0,1mm, 10% de preenchimento | 0h5min | 1 |
| TOTAL | | 16h50min | 163 |

Fonte: Adaptado de Brenneis e Austin (2017, p. 4).

Impressa em material PLA (polímero termoplástico), foram respeitadas as medidas padrão de projeto da prótese num primeiro momento, para que pudesse ser feita a escolha dos motores a serem utilizados, dentro dos modelos comercialmente disponíveis. O modelo que mais se adequou à prótese em medidas de comprimento foi o *Feetech Fs0307*. Este motor é um servomotor que possui tensão nominal de 6 V, corrente de pico de 600 mA, torque de 7 kg.cm, 20 mm de comprimento, 8,3 mm de largura e 19,3 mm de altura.

Apesar de apresentar um comprimento aceitável ao projeto da prótese, apenas 3 motores puderam ser fixados à palma de mão da prótese - polegar, indicador e anelar -, devido à sua altura. Isso fazia com que, ao fixar os motores dos dedos médio

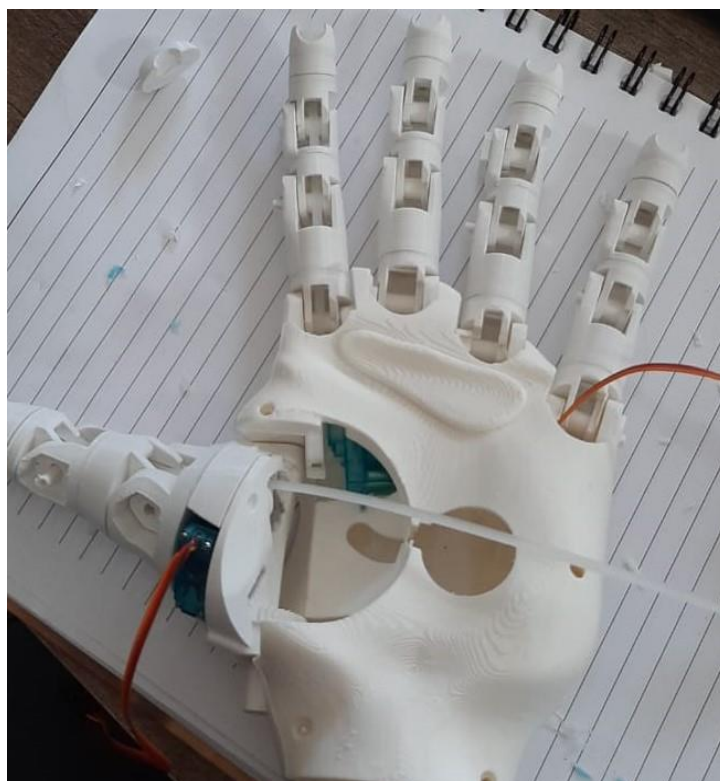
e mínimo, não sobrasse espaço para acoplar as polias aos eixos dos motores. A partir deste ponto, uma nova estratégia de trabalho foi adotada: alterar a escala de impressão da prótese de modo a comportar os motores adquiridos.

Para se adequar às medidas dos motores, a prótese foi novamente impressa, porém aumentada em 1,3 em sua escala. Com as dimensões validadas, os testes mecânicos foram realizados nas seguintes etapas: montagem da prótese, desenvolvimento de sistema emulador de movimentos e desenvolvimento de código de acionamento da prótese.

4.2 Montagem da prótese

A montagem da prótese foi realizada em algumas etapas: montagem dos dedos indicador, médio, anelar e mínimo; montagem do dedo polegar; montagem dos dedos na prótese; acoplamento dos motores. Os dedos indicador, médio, anelar e mínimo foram montados juntos pois estes eram muito semelhantes, mudando apenas a peça da base de cada dedo. Esta peça se adequava à parte da palma da mão a qual seria encaixada, conforme demonstrado na Figura 22.

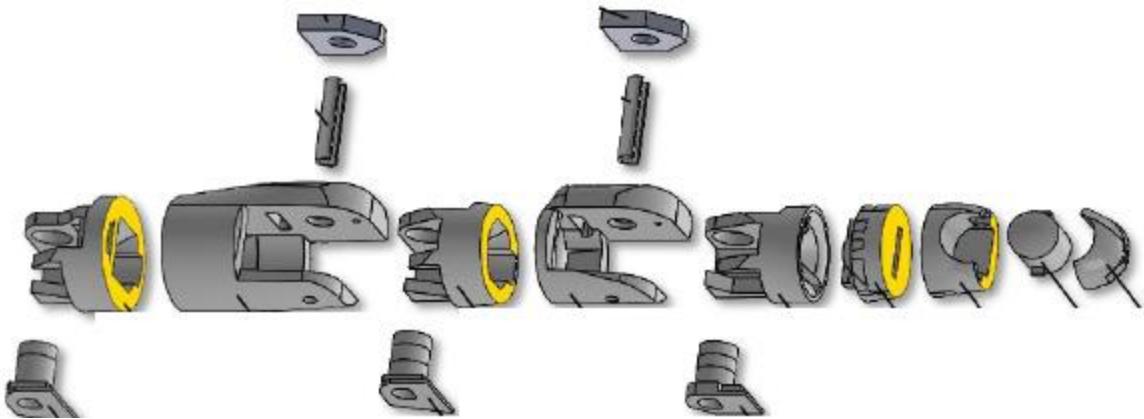
Figura 22 – Palma da mão da Prótese



Fonte: Elaborado pelo autor (2021).

A Figura 23 demonstra as partes componentes do dedo indicador, nas quais as faces pintadas de amarelo são pontos colados com supercola. Já nas partes das articulações, foi acoplado um pino específico para cada parte do dedo, responsável por articulá-lo durante a movimentação.

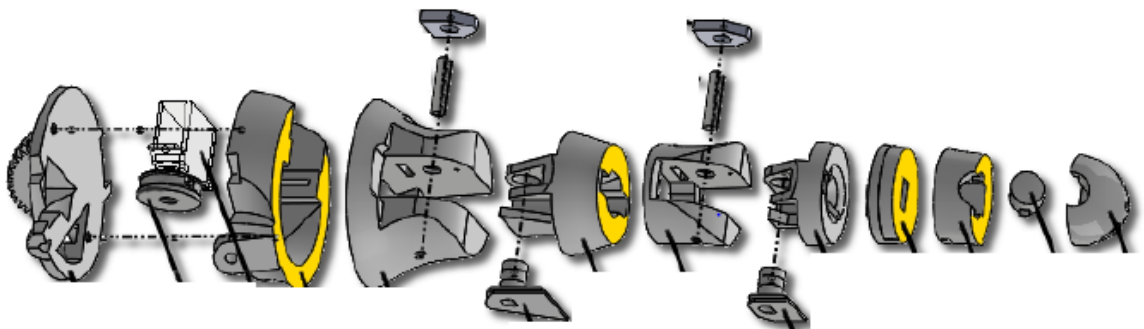
Figura 23 – Partes componentes - indicador



Fonte: Adaptado de Brenneis e Austin (2017, p. 16).

O dedo polegar foi montado com seu motor acoplado, pois este ficava localizado na parte interna do dedo. Assim como para os demais dedos, o polegar foi montado seguindo o padrão de colagem das faces e pinagem das juntas, demonstradas na vista explodida das partes componentes do polegar, conforme Figura 24 – Partes componentes – polegar.

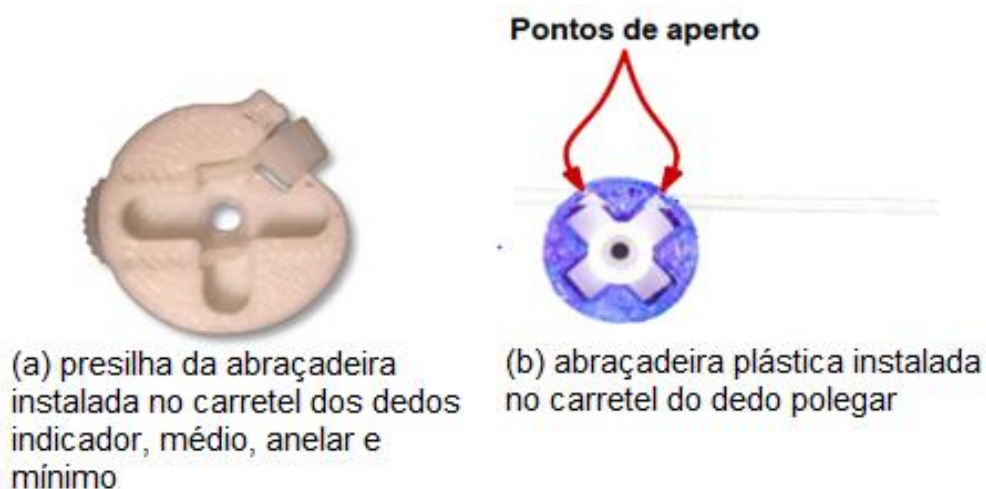
Figura 24 – Partes componentes – polegar



Fonte: Adaptado de Brenneis e Austin (2017, p. 13).

A transmissão de movimento dos motores para os dedos foi feita através de um carretel acoplado diretamente ao eixo do motor e uma cinta plástica presa na ponta do dedo e no carretel, conforme Figura 25 – Transmissão de movimento dos dedos

Figura 25 – Transmissão de movimento dos dedos



Fonte: Adaptado de Brenneis e Austin (2017, p. 17).

Após a montagem individual dos dedos, realizou-se a montagem dos dedos e dos motores na prótese. Os dedos e motores foram montados e testados um a um, de modo a determinar sua posição de fixação e seu ângulo de repouso (mão aberta). Teoricamente, pressupôs-se que o ângulo de repouso de todos os motores seria 0° . Entretanto, como a posição de fixação do motor muda de dedo para dedo, o teste foi realizado individualmente. Para a realização dos testes, girou-se o eixo do motor no sentido anti-horário até o final do curso. Os resultados das posições de *software* foram apresentados na Tabela 2.

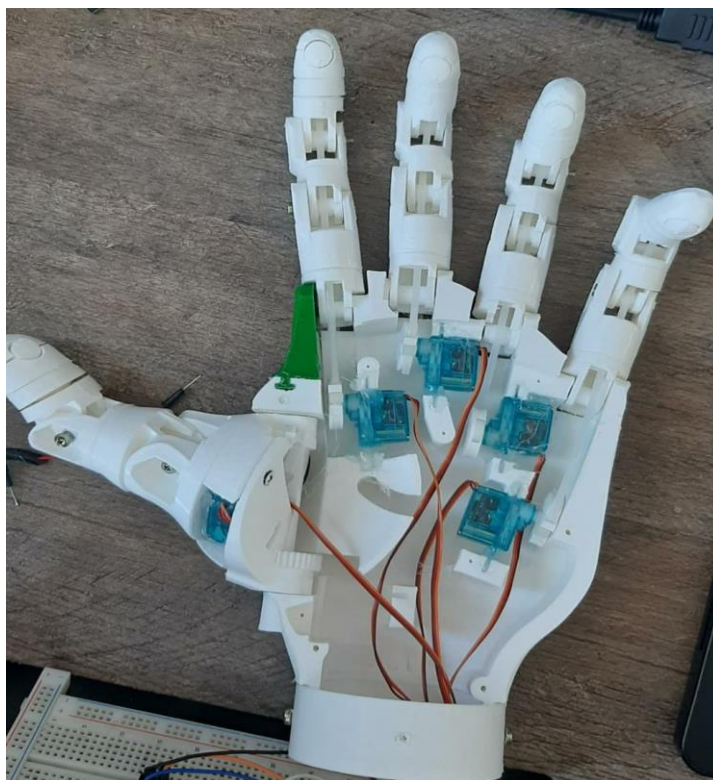
Tabela 2 – Posição de repouso dos motores

| Motor | Dedo | Posição Angular ($^{\circ}$) |
|-------|-----------|--------------------------------|
| 1 | Polegar | 0 |
| 2 | Indicador | 180 |
| 3 | Médio | 180 |
| 4 | Anelar | 0 |
| 5 | Mínimo | 0 |

Fonte: Elaborado pelo autor, 2021.

A Figura 26 demonstra a disposição dos motores dentro na mão, justificando a variação de posição de repouso entre os dedos.

Figura 26 – Transmissão de movimento dos dedos



Fonte: Elaborado pelo autor (2021).

Para testar o movimento dos dedos foi desenvolvido um código genérico no microcontrolador ESP32. O conceito desse código foi utilizar a entrada analógica da placa ligada a um potenciômetro de 10 k Ω , onde a posição transmitida para o motor seria proporcional à tensão elétrica da entrada analógica (0 a 5 V). Para acionar os motores foi utilizada a biblioteca ESP32-Arduino-Servo-Library (ROBOJAX, 2019).

Os custos dos componentes utilizados na prótese foram descritos na Tabela 6 do Apêndice B.

4.3 Supervisório emulador de movimentos

O supervisório emulador de movimentos foi criado com o objetivo de emular um sistema de aquisição de sinais comunicando-se com o sistema de acionamento da prótese. Como o sistema elaborado por Souza (2021) - descrito no capítulo 3 - enviava os dados EMG do braço via Bluetooth para o classificador de movimentos, determinou-se que a comunicação entre o supervisório e a ESP32 também seria feita via comunicação Bluetooth.

O desenvolvimento da aplicação poderia ser feito através de um computador ou de um *smartphone*. As aplicações da ESP32 já estavam sendo feitas e testadas via computador, e como necessitava-se constantemente do uso da serial para *download* dos códigos implementados, optou-se por utilizar um *smartphone* como emulador de movimentos. O *smartphone* utilizado foi um Samsung Galaxy A30s, modelo SM-A307GT, com sistema Android versão 10.

Os primeiros testes realizados entre o *smartphone* e a ESP32 foram através do aplicativo *Serial Bluetooth Terminal*, onde informações poderiam ser escritas no terminal e enviadas via *Bluetooth* para o microcontrolador. O segundo passo dos testes foi configurar a comunicação *Bluetooth* na ESP32 e desenvolver um código para interpretar as informações recebidas e então acionar os motores. Para realizar esta comunicação foi utilizada a biblioteca *BluetoothSerial*. O Quadro 2 demonstra o código implementado para testar a comunicação entre o *smartphone* e a ESP32.

Quadro 2 – Código de comunicação *Bluetooth*

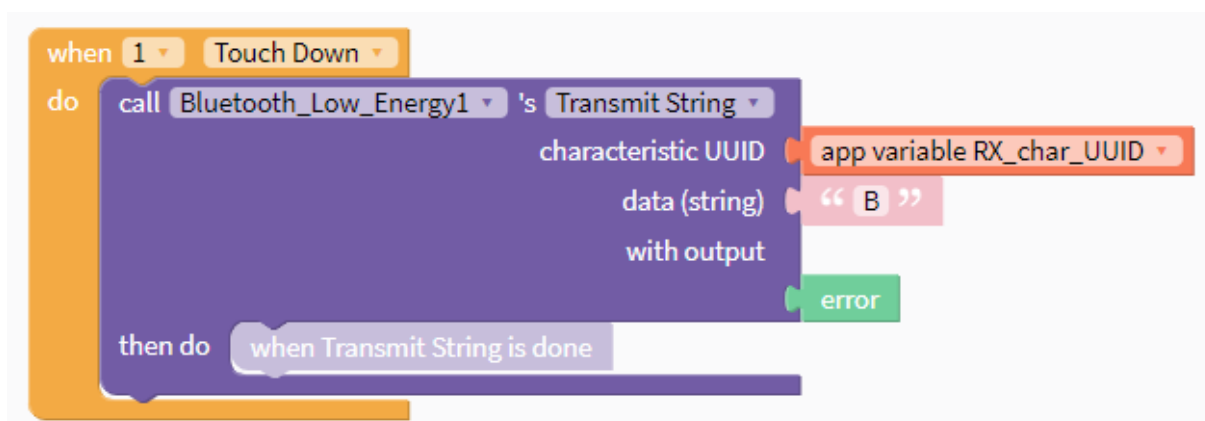
```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
void setup() {
  Serial.begin(115200);
  SerialBT.begin("ESP32");
}
void loop() {
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

Fonte: Elaborado pelo autor (2021).

Conforme o Quadro 2, a comunicação *Bluetooth* da ESP32 era ativada e, em seguida, testava a conexão com algum dispositivo dentro da função *void loop*. Este sistema funcionou a nível de comunicação, porém o aplicativo mantinha um envio cíclico de informações pelo terminal, gerando ruídos na comunicação, fazendo com que o *Bluetooth serial* do microcontrolador recebesse informações não coerentes com a aplicação.

Para que fosse resolvido o problema dos ruídos de comunicação, optou-se por desenvolver uma aplicação na plataforma de desenvolvimento de aplicativos Thunkable. Thunkable é uma plataforma *drag-and-drop* de desenvolvimento de aplicativos *online* que possui um editor com dois principais componentes: *design* e blocos. Na aba *design* é possível implementar os aspectos visuais do aplicativo, como imagens, botões, indicadores. A plataforma também conta com alguns *features* como gravador de voz, tradutor, *barcode scanner*, câmera, *BLE*, entre outros. Na aba *blocks* existe um menu de opções de blocos de controle, usados para programar as funções implementadas na aba *design*. A Figura 27 demonstra um exemplo de bloco de função possível de ser implementado na plataforma.

Figura 27 – Bloco de funções na plataforma Thunkable



Fonte: Thunkable (2021).

Apesar da comunicação via *Bluetooth* entre o aplicativo *Serial Bluetooth Terminal* e a ESP32 ter funcionado, o mesmo código implementado anteriormente (Quadro 2) não poderia ser utilizado para comunicar com a plataforma Thunkable, pois esta possui apenas comunicação via BLE, a qual é configurada de outra maneira.

A aplicação do supervisor foi organizada conforme Quadro 3.

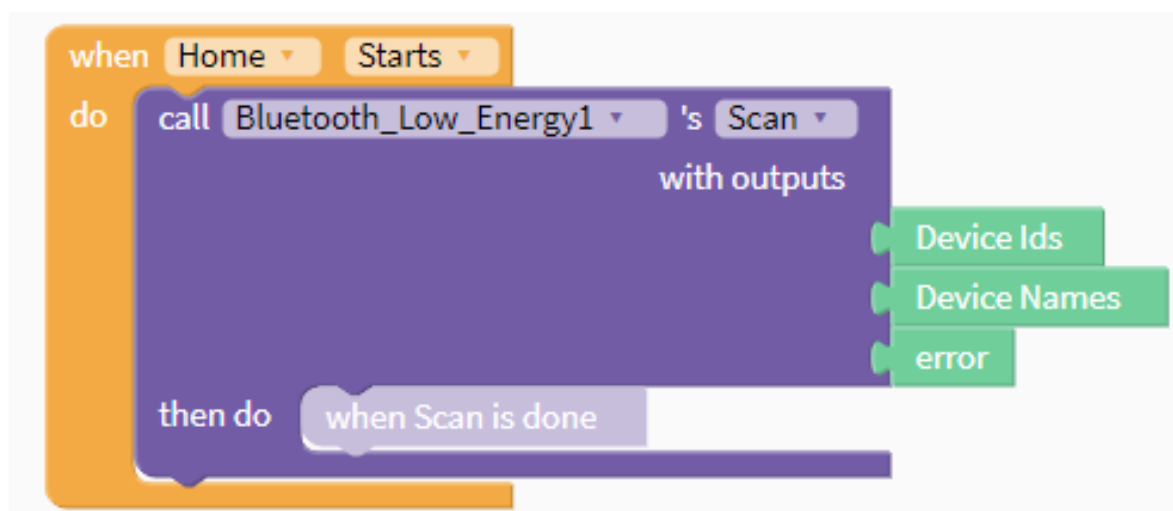
Quadro 3 – Organização da aplicação do sistema supervisorio

- 1) Tela principal com números de 0 a 10, representando os movimentos da prótese;
- 2) Bloco de *scan* dos dispositivos *Bluetooth*;
- 3) Diagrama de blocos com etapa de comunicação com a ESP32 (*service_UUID*, *RX_char_UUID* e *TX_char_UUID*);
- 4) Blocos de envio de dados via BLE.

Fonte: Elaborado pelo autor (2021).

A representação dos movimentos citado no Quadro 3 foi projetada de 0 a 10 de modo a ser compatível com o trabalho realizado por Souza (2021), caso os dois sistemas venham a ser testados juntos futuramente. A relação de movimentos será descrita no item 4.5 deste trabalho.

O bloco de *scan*, representado na Figura 28 tem a função de fazer uma varredura nos dispositivos disponíveis, coletando seus dados de configuração para que o bloco de conexão possa encontrar o dispositivo a ser conectado.

Figura 28 – Bloco de *scan* dos dispositivos

Fonte: Thinkable (2021).

Este bloco indica que, quando a tela principal inicializar, o componente BLE da aplicação vai iniciar sua busca pelos “Ids” e pelos nomes dos dispositivos disponíveis, sem estabelecer nenhuma conexão com eles. Os blocos de parametrização da comunicação BLE definem a identificação *Universal Asynchronous Receiver/Transmitter* (UART) do aplicativo. Por definição, os parâmetros atribuídos ao

BLE do aplicativo devem ser os mesmos que forem atribuídos a ESP32. A Figura 29 demonstra o bloco de parametrização da aplicação.

Figura 29 – Parametrização do BLE

```

initialize app variable service_UUID to " 6E400001-B5A3-F393-E0A9-E50E24DCCA9E "
initialize app variable RX_char_UUID to " 6E400002-B5A3-F393-E0A9-E50E24DCCA9E "
initialize app variable TX_char_UUID to " 6E400003-B5A3-F393-E0A9-E50E24DCCA9E "
  
```

Fonte: Thunkable (2021).

Os blocos de envio de dados via BLE foram programados para enviarem o valor do movimento desejado enquanto fossem pressionados, e para enviarem o valor referente à posição de repouso da prótese quando fossem soltos. A Figura 30 demonstra um exemplo desta configuração.

Figura 30 – Bloco de envio de dados via BLE

```

when 1 Touch Down
do
  call Bluetooth_Low_Energy1's Transmit String
  characteristic UUID: app variable RX_char_UUID
  data (string): " B "
  with output
  error
  then do
    when Transmit String is done
  
```

a) envia a letra "B" quando pressionado para baixo

```

when 1 Touch Up
do
  call Bluetooth_Low_Energy1's Transmit String
  characteristic UUID: app variable RX_char_UUID
  data (string): " A "
  with output
  error
  then do
    when Transmit String is done
  
```

b) envia a letra "A" quando o botão é solto

Fonte: Adaptado de Thunkable (2021).

A Figura 31 apresenta um *layout* da versão final da tela principal do aplicativo, a qual servirá como interface entre a prótese e a pessoa que estiver operando o sistema.

Figura 31 – Layout final do sistema supervisorio



Fonte: Elaborado pelo autor (2021).

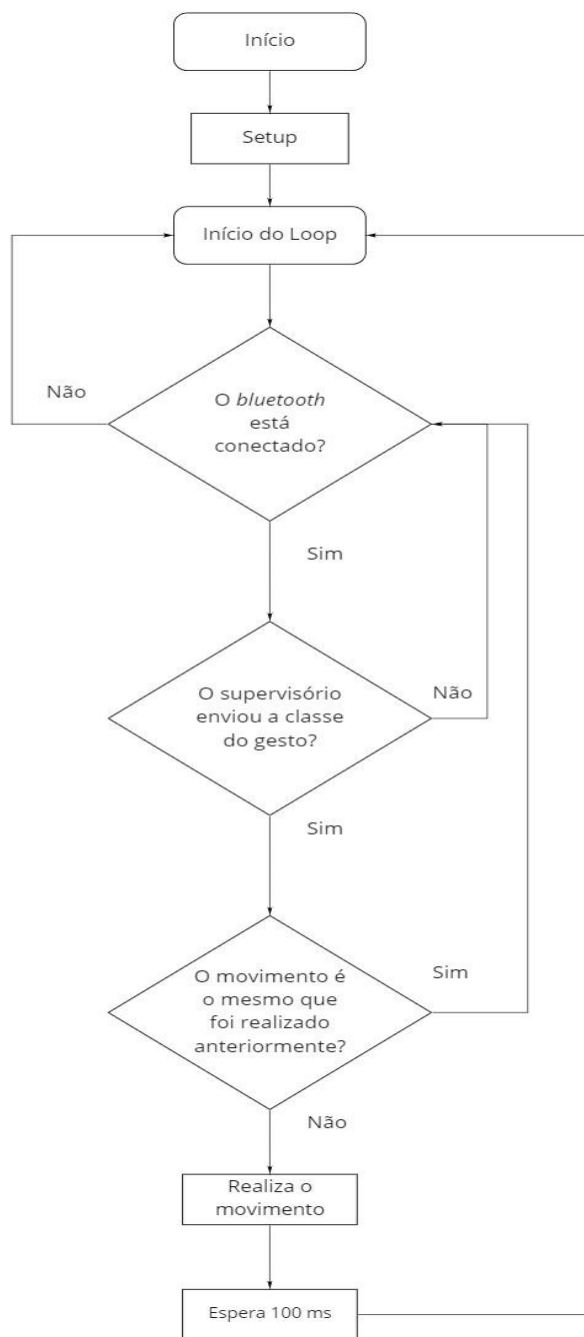
Os botões “*Connect*” e “*Disconnect*” têm a função de realizar a conexão Bluetooth entre o aplicativo e a ESP32. Conforme citado anteriormente, os números da tela têm como função enviar os movimentos para o sistema de acionamento da prótese.

4.4 Sistema de acionamento

Estando a prótese montada e o sistema supervisorio desenvolvido, iniciou-se a etapa de codificação do sistema de acionamento da prótese. O objetivo desta codificação era interpretar a classe de movimento recebida do sistema supervisorio e acionar a prótese de acordo com a posição angular e o incremento de posição correspondente a cada movimento. É válido salientar que nenhuma classe de movimento teve sua posição previamente programada através de bancos de dados. Todas as posições foram programadas mediante testes e repetições de variados

valores, até que se chegasse aos resultados minimamente satisfatórios. A Figura 32 apresenta um fluxograma inicial simplificado de como deve funcionar o sistema de acionamento gerenciado pela ESP32.

Figura 32 – Fluxograma de funcionamento do sistema de acionamento



Fonte: Elaborado pelo autor (2021).

A partir do fluxograma apresentado na Figura 32, os próximos tópicos deste trabalho irão abordar brevemente cada etapa desenvolvimento do sistema.

Configuração do *BLE* na ESP32

Conforme citado anteriormente no item 4.3 deste trabalho, a configuração do BLE na ESP32 deve seguir o mesmo padrão adotado no sistema supervisorio. Toda a programação do microcontrolador foi feita através do *software* Arduino IDE.

Para a configuração foi usada as bibliotecas BLEDevice, BLEServer, BLEUtils e BLE2902 (Arduino.cc, 2021). Juntamente com as bibliotecas, o próprio *software* disponibiliza um *template* de comunicação BLE, o qual foi utilizado como base para realizar a configuração/comunicação *Bluetooth* entre a ESP32 e o sistema supervisorio. O Quadro 4 demonstra a configuração utilizada para parametrizar o microcontrolador.

Quadro 4 – Parametrização do *Bluetooth* na ESP32

```
#define SERVICE_UUID      "6E400001-B5A3-F393-E0A9-E50E24DCCA9E"  
#define CHARACTERISTIC_UUID_RX "6E400002-B5A3-F393-E0A9-E50E24DCCA9E"  
#define CHARACTERISTIC_UUID_TX "6E400003-B5A3-F393-E0A9-E50E24DCCA9E"
```

Fonte: Elaborado pelo autor (2021).

Além disso, foi criada uma variável nomeada de “rxValue” responsável por receber os dados do sistema supervisorio. Não foram criadas variáveis de envio de dados, pois, à *priori*, nenhum dado de comando ou *feedback* seria enviado da ESP32 para o aplicativo. Apesar de não enviar dados ao sistema supervisorio, o BLE da ESP32 é configurado como *server* da aplicação, ou seja, este recebe uma requisição do seu *client* (aplicativo) e executa a tarefa solicitada, não enviando nenhuma resposta como retorno para o supervisorio (DEV MEDIA, 2021).

4.4.1 Classificação dos motores

Para realizar o acionamento elétrico dos cinco motores utilizados na prótese, foi adicionada ao projeto a biblioteca ESP32-Arduino-Servo-Library (ROBOJAX, 2019). De modo a realizar o movimento, usou-se do recurso suportado pela linguagem de programação do Arduino, a programação orientada a objeto. Por conta de possuir cinco motores, com vários destes acionando ao mesmo tempo, optou-se por utilizar este recurso para ganhar dinâmica e não tornar o programa repetitivo com diversos

Loop For. Assim, criou-se uma classe de motores em que, através da função membro chamada *Update()*, era capaz de acionar diversos motores com posições e incrementos de posição diferentes.

Para tornar possível a chamada de cinco classes de motores para execução do movimento síncrono, ou seja, para que a prótese executasse o movimento dos cinco dedos em conjunto, utilizou-se a função *millis()* do Arduino para memorizar a última posição do motor, como uma máquina de estados (FILIFELOP, 2021). O Quadro 5 mostra uma parte do código implementado para acionamento dos motores.

Quadro 5 – Código de acionamento dos motores

```
int Update(int interval, int finalPos)
{
  updInterval = interval;
  endPos = finalPos;

  if(endPos < 60)
  {
    //Serial.println(endPos);
    if ((millis() - lastUpdate) > updInterval)
    {
      lastUpdate = millis();

      if(0 != pos > endPos)
      {
        pos -= increment;
        servo.write(pos);
      }
    }

    if(pos == endPos)
    {
      return 1;
    }
  }
}
```

Fonte: Elaborado pelo autor (2021).

A variável “interval” define o tempo entre cada incremento de posição do motor. Isto faz com que o movimento executado pela prótese seja mais rápido ou mais lento.










A variável “finalPos” recebida pela função *Update()* define a posição final de acionamento dos motores. Para isto, duas condições de acionamento foram criadas: movimentos incrementais e movimentos decrementais. O motivo desta definição não é apenas o movimento de abrir e fechar mão, mas também pois a posição inicial de alguns motores é 180° e não 0°. Desta forma, enquanto alguns motores fecham o dedo incrementando posições, outros fazem a mesma função através do decremento de posições.

4.5 Testes de movimentos

Tendo as etapas 1 e 2 da Figura 20 sido concluídas, o projeto entrou na fase de testes de validação dos dois sistemas, onde ambos trabalhariam em conjunto. Para a realização destes testes foi utilizada a base de dados KidoPro (SOUZA, 2021), inspirada no banco de dados NinaPro DB2 (ATZORI *et al.*, 2015).

Por possuir apenas 5 motores disponíveis para movimento, a prótese limitou-se a realizar movimentos que não necessitassem de movimentos de pulso ou alguma articulação específica. Ainda assim, alguns movimentos de pulso como supinação e pronação foram considerados e executados simbolicamente através dos dedos, para que o sistema estivesse de acordo com o sistema projetado por Souza (2021). O Quadro 6 apresenta o banco de dados KidoPro, juntamente com a caractere enviado pelo sistema supervisorio (2021).

Quadro 6 – Movimentos do banco de dados KidoPro

| Classe | Descrição | Imagem | Caractere |
|---------------|------------------------|--|------------------|
| 0 | Repouso |  | A |
| 1 | Mão fechada |  | B |
| 2 | Polegar |  | C |
| 3 | Mão aberta |  | D |
| 4 | Pinça fina |  | E |
| 5 | Pegada cilíndrica |  | F |
| 6 | Pronação de punho |  | G |
| 7 | Supinação de punho |  | H |
| 8 | Pegada esférica |  | I |
| 9 | Ferramenta da preensão |  | J |
| 10 | Agarrar tampa |  | K |

Fonte: Adaptado de Souza (2021).

Inicialmente, foram necessários alguns testes iniciais para que as posições angulares de cada movimento fossem testadas e validadas no código. Para isto, cada posição foi repetida por 10 vezes na posição estipulada para que os valores fossem coerentes com a posição desejada e as capacidades mecânicas da prótese. A Tabela 3 apresenta as posições angulares dos motores para cada movimento realizado.

Tabela 3 – Posições angulares dos gestos da prótese

| Classe | Descrição | Posição angular (°) | | | | |
|--------|------------------------|---------------------|-------------|-------------|--------|--------|
| | | Polegar | Indicador | Anelar | Médio | Mínimo |
| 0 | Repouso | 0 | 180 | 180 | 0 | 0 |
| 1 | Mão fechada | 180 | 0 | 0 | 180 | 200 |
| 2 | Polegar | 0 | -10 | -10 | 200 | 200 |
| 3 | Mão aberta | 0/80/0 | 180/100/180 | 180/100/180 | 0/80/0 | 0/80/0 |
| 4 | Pinça fina | 180 | 0 | 180 | 0 | 0 |
| 5 | Pegada cilíndrica | 110 | -10 | 0 | 100 | 140 |
| 6 | Pronação de punho | 10 | 180 | 180 | 120 | 120 |
| 7 | Supinação de punho | 180 | 180 | 180 | 0 | 120 |
| 8 | Pegada esférica | 150 | 0 | 0 | 150 | 150 |
| 9 | Ferramenta da preensão | 180 | 0 | -10 | 200 | 200 |
| 10 | Agarrar tampa | 180 | 180 | 180 | 180 | 180 |

Fonte: Elaborado pelo autor (2021).

Apesar de serem utilizados servomotores com *range* de 0 a 180°, algumas posições extrapolaram estes batentes virtuais. Por mais que isto denote uma falha de configuração ou posicionamento dos motores, todos os motores foram zerados e instalados em suas posições 0 ou 180°, dependendo do motor. O que aconteceu, neste caso, foi que as posições reais e virtuais eram divergentes. Por vezes, era enviado o comando de 180° para o motor e este parava antes do seu fim-de-curso de hardware. Portanto, as posições de *software* apresentadas na Tabela 4 não correspondem às posições angulares físicas dos motores, apenas valores ensaiados para que a prótese pudesse atingir a posição real desejada.

Além das rotinas e testes de movimentos, também foi desenvolvida uma rotina de memorização do último movimento da prótese. Essa rotina era responsável por armazenar o último gesto em uma variável e monitorá-la a cada 100 ms. Se no ciclo de 100 ms seguinte ao gesto realizado fosse solicitado o mesmo gesto pelo sistema supervisor, a rotina mantinha o gesto sem voltar a prótese para o repouso.

5 ANÁLISE DOS MOVIMENTOS

Este capítulo destina-se à análise física dos movimentos realizados pela prótese, como tempo de execução e qualidade de execução. Para isso, cada movimento foi cronometrado e repetido por 10 vezes, visualmente analisado e comparado com movimentos humanos reais. Os movimentos humanos apresentados em comparação com os movimentos da prótese na Tabela 5, foram ensaiados e repetidos por 10 vezes pelo autor deste trabalho, simulando uma ação real do cotiado de um possível usuário da prótese. Estes movimentos representam também uma comparação entre os tempos de resposta da mão real e da prótese de mão, para o caso de a prótese ser futuramente usada em testes com sinais EMG.

O movimento de repouso da prótese (classe 0) apresentou uma leve curvatura dos dedos, a qual buscou-se corrigir através de elásticos colados atrás dos dedos. Em um primeiro momento, os elásticos mantiveram os dedos em uma posição esticada. Entretanto, no momento dos testes, os servomotores não foram capazes de executar o movimento completo por falta de torque. A Figura 33 apresenta o movimento de repouso (classe 0).

Figura 33 – Gesto de repouso da mão (classe 0)

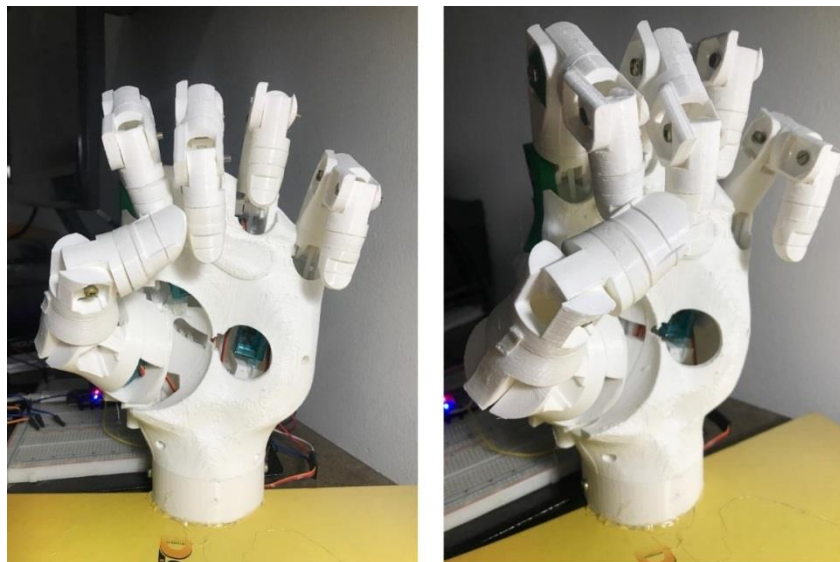


Fonte: Elaborado pelo autor (2021).

Para o gesto de mão fechada (classe 1), apresentado na Figura 34, a prótese demonstrou certa resistência em realizar os últimos movimentos dos dedos, os quais exigiam mais força dos motores e não contavam com o auxílio da aceleração da gravidade. Outro ponto observado foi a falta de algumas articulações do polegar, pois

este executou seu movimento de fechamento, porém sua base se manteve estática, já que não possui motores para este movimento.

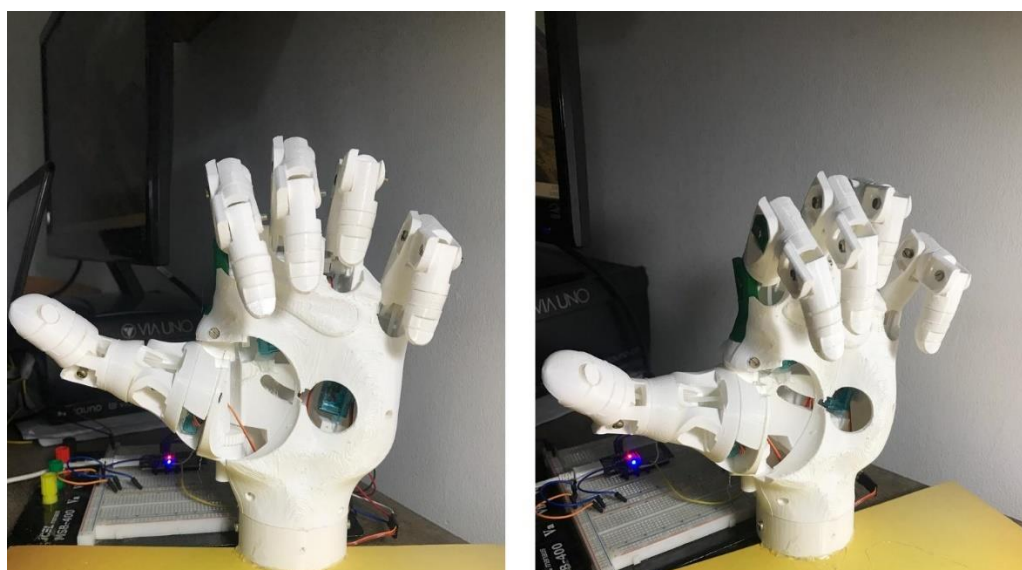
Figura 34 – Gesto de mão fechada (classe 1)



Fonte: Elaborado pelo autor (2021).

O gesto de polegar apresentou características de movimento similares aos de mão fechada, com a exceção do polegar que se manteve em sua posição de repouso. A posição de repouso do polegar manteve-se levemente curvada, não havendo o esticamento total do dedo. A Figura 35 apresenta o movimento do polegar.

Figura 35 – Gesto de polegar (classe 2)

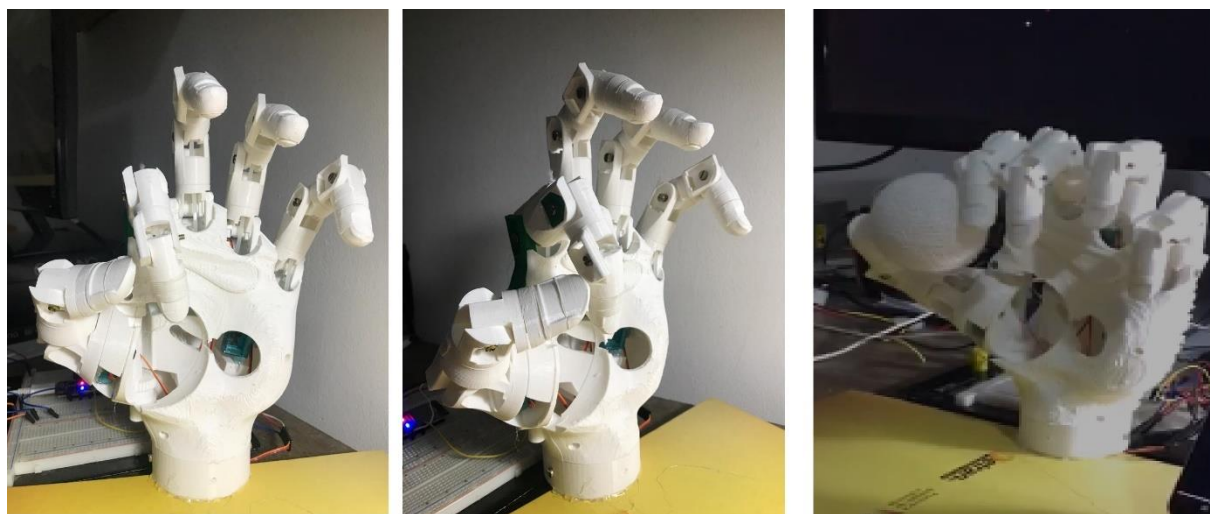


Fonte: Elaborado pelo autor (2021).

O gesto de abertura de mão (classe 3) foi simbolizado com um leve fechamento da mão e em seguida sua abertura. As conclusões sobre este movimento mantiveram-se as mesmas do movimento de repouso, já que a prótese não é capaz de esticar os dedos além do repouso. Portanto, a figura de referência deste gesto é a mesma do gesto de mão em repouso, ou seja, a Figura 33.

Os gestos de pinça fina e ferramenta de preensão (classe 4 e 9) apresentaram boa execução enquanto movimento, mas pouca efetividade prática. Em outras palavras, os movimentos foram pouco capazes de realizar a preensão de objetos através dos dedos polegar e indicador. Ao longo dos 10 testes, alguns testes obtiveram êxito e em outros o objeto caiu da prótese. Foi observado que a prótese executa a preensão do objeto com a lateral dos dedos em vez de usar sua parte interna, a qual possui maior área de contato. Assim como no gesto de mão fechada, os gestos de pinça fina e de ferramenta de preensão demonstraram ser afetados pela falta de articulações auxiliares no dedo polegar. A Figura 36 demonstra os gestos.

Figura 36 – Gesto de pinça fina e ferramenta de preensão (classes 4 e 9)



(a) Pinça fina

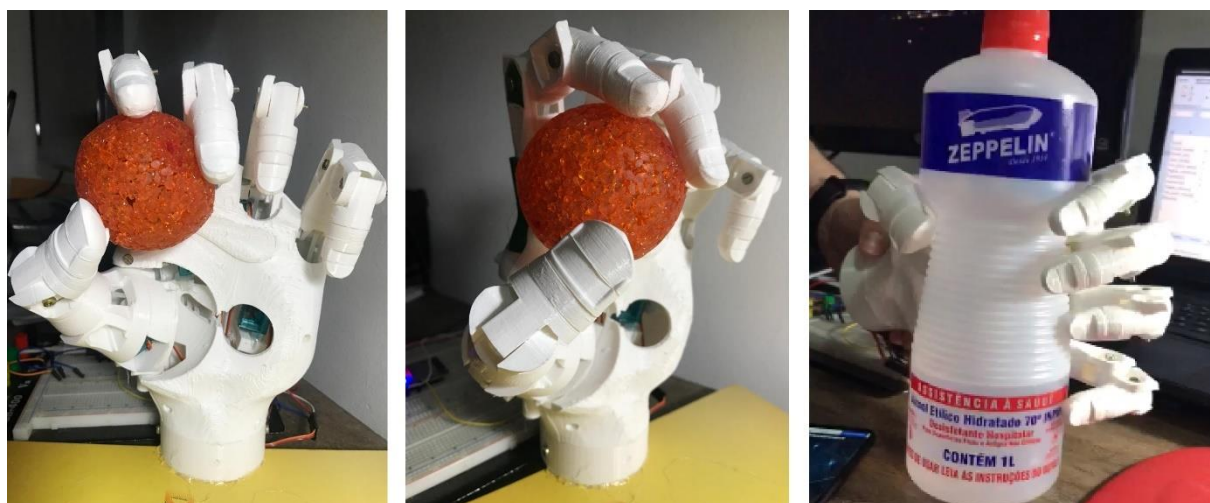
(b) Ferramenta da preensão

Fonte: Elaborado pelo autor (2021).

O gesto de pegada esférica e de pegada cilíndrica, apresentado na Figura 37, foi executado utilizando uma bola de entretenimento para gatos e uma garrafa de álcool de uso doméstico, respectivamente. Estes gestos foram analisados juntos, pois apresentaram as mesmas características quanto à mecânica de movimento. Inicialmente, os testes de pegada cilíndrica foram realizados com garrafas PET de 500 ml, porém o formato do corpo da garrafa fazia com que esta escorregasse pelos dedos da prótese, sem que houvesse a preensão. Com a garrafa de álcool doméstico e a

bola, contudo, ambas pegadas somente foram possíveis através dos dedos polegar, indicador e médio. Por possuírem uma configuração diferente de montagem dos motores, os dedos anelar e mínimo iniciavam seus movimentos através das falanges distais (cabeças) (CAIUSCA, 2019), fato que fazia com que as pontas dos dedos escorregassem pelo objeto sem que o dedo pudesse efetivamente “abraçá-lo”.

Figura 37 – Gestos de pegada esférica e cilíndrica (classes 5 e 8)



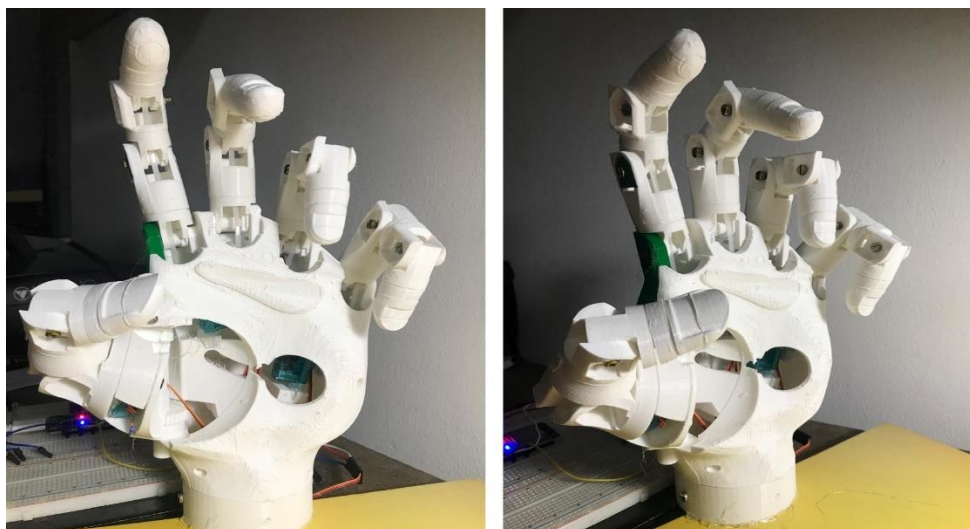
(a) Pegada esférica

(b) Pegada cilíndrica

Fonte: Elaborado pelo autor (2021).

Assim como no caso das pegadas esférica e cilíndrica, os movimentos de supinação, pronação e mandril (agarrar tampa) foram analisados em conjunto por possuírem a mesma característica limitante: a prótese não é capaz de executá-los. Os movimentos de pronação e supinação não foram possíveis de executar, pois a prótese não possui articulação de pulso. Já no caso do movimento de mandril, a limitação da prótese foi a falta de articulações auxiliares na palma da mão, já que o movimento exigia uma aproximação dos dedos anelar e mínimo não executável pela prótese. Para representar o movimento, os três gestos foram representados com movimentos dos dedos, apresentados na Figura 38.

Figura 38 – Gestos de pronação, supinação e mandril (classes 6, 7 e 10)



Fonte: Elaborado pelo autor (2021).

Após os movimentos serem individualmente testados, fotografados e filmados, estes foram submetidos a testes de tempo de execução. Conforme citado anteriormente, a Tabela 4 apresenta a média e o desvio padrão dos tempos de execução de movimento da prótese.

Tabela 4 – Tempos de resposta da prótese

| Classe | Descrição | Tempo de execução (s) +/- Desvio padrão (s) | |
|--------|------------------------|---|----------------|
| | | Prótese | Mão humana |
| 0 | Repouso | - | - |
| 1 | Mão fechada | 1,21 +/- 0,02 | 0,85 +/- 0,017 |
| 2 | Polegar | 0,52 +/- 0,014 | 0,46 +/- 0,015 |
| 3 | Mão aberta | - | - |
| 4 | Pinça fina | 1,13 +/- 0,02 | 0,52 +/- 0,016 |
| 5 | Pegada cilíndrica | 0,68 +/- 0,026 | 0,29 +/- 0,014 |
| 6 | Pronação de punho | 0,61 +/- 0,016 | 0,35 +/- 0,012 |
| 7 | Supinação de punho | 0,94 +/- 0,013 | 0,38 +/- 0,017 |
| 8 | Pegada esférica | 1,27 +/- 0,017 | 0,29 +/- 0,015 |
| 9 | Ferramenta da preensão | 1,31 +/- 0,021 | 0,68 +/- 0,021 |
| 10 | Agarrar tampa | 0,68 +/- 0,015 | 0,31 +/- 0,011 |

Fonte: Elaborado pelo autor (2021).

Os gestos de repouso e mão aberta não foram ensaiados pois o tempo de movimento dependeria do gesto anterior ao seu movimento.

Através da Tabela 4, pode-se observar que os movimentos de prótese que possuíram maiores divergências em relação aos movimentos de mão reais foram os que exigiram ação maior do polegar. Este dedo, em especial, na posição de mão em que os testes foram realizados, foi o único que sofreu ação contrária à aceleração da gravidade, enquanto todos os outros tinham a gravidade a seu favor (desconsiderando os movimentos de “abrir mão”). Outro fator importante a ser considerado é que cada movimento foi programado com tempos de acionamento diferentes, a fim de respeitar as limitações mecânicas da prótese e dos motores. É válido ressaltar que os tempos dos gestos de mão reais são meramente representativos e valem somente a nível de comparação com a prótese, podendo variar de pessoa para pessoa, já que nenhuma média mundial ou banco de dados foi consultado para este fim.

5.1 Testes com a Raspberry

O objetivo dos testes com a Raspberry Pi eram tornar a prótese de mão deste trabalho possível de ser integrada com o sistema desenvolvido por Souza (2021). Os primeiros testes realizados entre a Raspberry e o sistema de acionamento da prótese foram desenvolvidos através de um código cíclico em Python 3.7, o qual enviava gestos de abrir e fechar mão durante 100 ms cada, a fim de testar e validar a comunicação serial entre os dispositivos.

Após os testes de comunicação entre a Raspberry e a ESP32, iniciou-se uma tentativa de configuração do BLE na Raspberry, com o objetivo de integrá-la com o sistema supervisor. Entretanto, após várias tentativas sem sucesso, mudou-se a estratégia para o aplicativo RaspController. Este aplicativo é capaz de enviar alguns comandos para a Raspberry, entre eles o controle da GPIO da placa. Contudo, por mais que o aplicativo estabelecesse comunicação via internet *wireless* com a placa, este provou-se instável e, por vezes, perdia a comunicação com a GPIO. Portanto, decidiu-se manter apenas os testes entre a ESP32 e a Raspberry via USB (serial), de modo que, caso o sistema venha a ser integrado com o trabalho desenvolvido por Souza (2021) via Raspberry Pi, bastará configurar o *Bluetooth* para receber os dados do sistema de aquisição de sinais. Assim, os testes de movimentos foram realizados somente entre o sistema supervisor, a ESP32 e a prótese.

6 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho de conclusão de curso foi desenvolver um sistema eletrônico capaz de acionar uma prótese de mão que, neste caso, era a prótese de mão *open source* Handi Hand, desenvolvida por Brenneis e Austin (2017) na Universidade de Alberta, no Canadá. A partir dos resultados observados e descritos neste trabalho, avalia-se que os objetivos foram parcialmente alcançados, devido a alguns empecilhos encontrados ao longo do seu desenvolvimento.

Integrar sistemas eletro/eletrônicos à sistemas mecânicos demonstra certos desafios do ponto de vista de que o ambiente eletrônico ou de *software* responde a comandos em uma velocidade impossível de se comparar com a velocidade de resposta de um sistema mecânico. Seja pelas influências externas como gravidade e temperatura, ou pelas limitações dos próprios materiais, sistemas mecânicos nem sempre serão capazes de acompanhar a complexidade dos sistemas eletrônicos. Avaliando os resultados obtidos, conclui-se que nem sempre a realidade pode ser reproduzida de forma trivial, de forma que algumas mudanças de escopo se fizeram necessárias ao longo do desenvolvimento do trabalho.

O primeiro desvio de escopo encontrado foi a alteração da escala da prótese no início do projeto. Este aumento de tamanho fez com que a prótese se tornasse mais pesada e mais lenta, exigindo mais torque dos motores, os quais eram de pequeno porte. Talvez com a prótese em tamanho original houvesse maior capacidade de movimentos. Já o desenvolvimento da prótese enquanto sua capacidade de realizar os movimentos propostos foi razoável, já que executou ou simbolizou os gestos propostos.

O segundo desvio de escopo foi a não capacidade de comunicação em BLE da Raspberry com o sistema supervisor. Este ponto não prejudicou completamente o objetivo principal de desenvolver um acionamento capaz de movimentar a prótese, mas infelizmente não foi capaz de se comunicar via *Bluetooth* com o sistema supervisor.

Embora seja necessário apontar as limitações da prótese, é preciso endossar a importância e a capacidade desta de ser um ótimo instrumento de estudo e testes para futuras pesquisas no ramo da eletromiografia ou desenvolvimento de sistemas de acionamento. É prudente ressaltar que pesquisas nessa área são de extrema

relevância, e mesmo pequenas contribuições auxiliam no desenvolvimento de inovações e aproximam os estudos cada vez mais da realidade.

Por fim, conclui-se que o sistema de acionamento foi capaz de acionar uma prótese de mão *open-source*, com potencial para ser explorado em pesquisas e trabalhos futuros.

6.1 Sugestões para trabalhos futuros

Com a finalização deste trabalho, algumas oportunidades de continuidade e melhorias foram descritas a seguir.

- Manter a prótese em seu tamanho original e desenvolver um sistema de transmissão de movimentos com motores DC, capazes de acionar a prótese com maior torque;
- Integrar o sistema de acionamento com o sistema de aquisição de sinais a fim de testar os tempos de resposta da prótese em uma situação real;
- Acrescentar elementos de instrumentação na prótese, como sensores resistivos de força e potenciômetros nas articulações de modo que haja uma resposta sensorial dos movimentos realizados;
- Integrar à prótese um sistema de controle de torque por corrente, fazendo com que se possa medir, ensaiar e gravar os esforços necessários para gestos de pegada de diferentes objetos.

REFERÊNCIAS

- ALLEGRO Micro Systems, LLC. **Fully integrated, hall effect-based linear current sensor IC with 2.1 kVRMS isolation and a low-resistance current conductor.** Massachusetts, 2013. Disponível em: <https://www.allegromicro.com/en/products/sense/current-sensor-ics/zero-to-fifty-amp-integrated-conductor-sensor-ics/acs712> Acesso em: 12 out. 2021.
- ASPINA Group. **Differences between brushed and brushless DC motors.** Japão, 30 mar. 2020. Disponível em: <https://www.aspina-group.com/en/learning-zone/columns/what-is/002/>. Acesso em: 4 out. 2020.
- ATZORI, M.; GIJSBERTS, A.; KUSBORSKIJ, I.; ELSIG, S.; HAGER, A.G.M.; DERIAZ, O.; CASTELLINI, C.; MÜLLER, H.; CAPUTO, B. Characterization of a benchmark database for myoelectric movement classification. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 23, n. 1, p. 73-83, 2015.
- ATZORI, M.; MÜLLER, H. The NinaPro database: a resource for semg naturally controlled robot hand prosthetics. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY (EMBC), 37. **Proceedings...** Switzerland: Swiss National Science Foundation, 2015.
- BENATTI, S.; MILOSEVIC, B.; FARELLA, E.; GRUPPIONI, E.; BENINI, L. A Prosthetic Hand Body Area Controller Based on Efficient Pattern Recognition Control Strategies. **Sensors**, v. 17, n. 4, p. 869, 2017.
- BRENNEIS, Dylan; AUSTIN, James. **Handi Hand**: Assembly manual. Canadá: University of Alberta, 2017. 31 p.
- CAIUSCA, Alana. **Guia de estudos**: ossos da mão. Disponível em: <https://www.guiaestudo.com.br/ossos-da-mao/amp> Acesso em: 12 out. 2021.
- CÂMARA, Marlon. **Bluetooth**: o que é e como funciona. TechTudo Informática, 2012. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/01/Bluetooth-o-que-e-e-como-funciona.html>. Acesso em: 4 out. 2020.
- COGNOLATO, M.; ATZORI, M.; MARCHESINI, C.; MARANGON, S.; FACCIO, D; TIENGO, C.; BASSETTO, F.; GASSERT, R.; PETRONE, N.; MÜLLER, H. Multifunction control and evaluation for a 3D printed hand prosthesis with the Myo armband by hand amputees. **BioRxiv**, out. 2018. Disponível em: <https://www.biorxiv.org/content/10.1101/445460v1> Acesso em: 4 out. 2020.
- DEV MEDIA. **Arquitetura Client/Server e Three-tier.** Rio de Janeiro, 2021. Disponível em: <https://www.devmedia.com.br/arquitetura-client-server-e-three-tier/5865>. Acesso em: 12 out. 2021.
- EGAL NET. **Rasp controller.** Disponível em: https://www.gallinaettore.com/android_apps/raspcontroller/ Acesso em: 4 out. 2020.

ESPRESSIF Systems. **ESP32 Series**: Datasheet. Shanghai, 2020.

FILIFELOP. **Driver Motor Ponte H L298n**. Disponível em: <https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/> Acesso em: 4 out. 2021.

FILIFELOP. **Módulo WiFi ESP32 Bluetooth**. Disponível em: https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/?utm_source=google&utm_medium=organic&utm_campaign=shopping&utm_content=surfaces_across_google&gclid=CjwKCAiA1aiMBhAUEiwACw25MScwox5q2_NoyGYRT-HasUCHuQiBhxYRv4cS6K3o2TNZ1twe2t8tFRoC30AQAvD_BwE Acesso em: 4 out. 2020.

FOUGNER, A. **Robust, Coordinated and Proportional Myoelectric Control of UpperLimb Prostheses**. 2013. 108p. (Doctoral theses) Faculty of Information Technology, Mathematics and Electrical Engineering Department of Engineering Cybernetics. Norwegian University of Science and Technology, Norway, 2013.

FOUGNER, A.; STAVDAHL, O.; KYBERD, P. J.; LOSIER, Y. G.; PARKER, P. A. Control of Upper Limb Prostheses: Terminology and Proportional Myoelectric Control – A Review. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, v. 20, n. 5, p. 663-677, 2012.

GAETANI, F.; ZAPPATORE, G. A.; PRIMICERI, P.; VISCONTI, P. Design of an Arduino-based platform interfaced by Bluetooth Low Energy with MYO armband for controlling an under-actuated transradial prosthesis. In: INTERNATIONAL CONFERENCE ON IC DESIGN & TECHNOLOGY (ICICDT). **Proceedings...** Otranto, Italy, 2018. p. 185-188.

HARRINGTON, A. M.; KRONINGER, C. **Characterization of small DC brushed and brushless motors**. Aberdeen Proving Ground, MD: Army Research Laboratory, 2013. Disponível em: <https://people.engr.ncsu.edu/mlsichit/Teaching/492/References/smallDCMotorEfficiency.pdf> Acesso em: 4 out. 2020.

HART, D. W. **Eletrônica de potência**: análise e projetos de circuitos. Porto Alegre: AMGH, 2012. *E-book*.

INSTITUTO NACIONAL DO SEGURO SOCIAL (INSS). **Manual técnico de prescrição de órteses, próteses ortopédicas não implantáveis e meios auxiliares de locomoção**: diretrizes para a perícia médica. 2017. Disponível em: <https://extra268.files.wordpress.com/2017/10/rs611presinssmanual1.pdf>. Acesso em: 4 out. 2020.

ISON, M.; ARTEMIADIS, P. **Enhancing Practical Multifunctional Myoelectric Applications through Implicit Motor Control Training Systems**. Arizona: Arizona State University, 2014.

KILBY, J.; HOSSEINI, G. **Wavelet Analysis of Surface Electromyography Signals**. Electrotechnology Department, Auckland University of Technology. New Zealand, 2004.

LAFFRANCHI, M.; BOCCARDO, N.; TRAVERSO, S.; LOMBARDI, L.; CANEPA, M.; LINCE, A.; SEMPRINI, M.; SAGLIA, J.A.; NACERI, A.; SACCHETTI, R.; GRUPPIONI, E.; MICHIELI L. de. **The Hannes hand prosthesis replicates the key biological properties of the human hand**. Washington, p. 1-16. September, 2020.

MOHAN, N. **Eletrônica de Potência**: curso introdutório. Rio de Janeiro: LTC, 2014.

MONTEIRO, A.; ZIMPEL, A.; OLIVEIRA, C.; MONTEIRO, L.; CARNEIRO, T. Prevalência de amputações de membros superiores e inferiores no estado de Alagoas atendidos pelo SUS entre 2008 e 2015. **Fisioter Pesqui.**, Alagoas, v. 24, n.4, p.378-384, 2017. Disponível em: <http://www.scielo.br/pdf/fp/v24n4/2316-9117-fp-24-04-378.pdf> Acesso em: 09 nov. 2021.

MOREIRA, F. B. **Controlo de próteses multifuncionais através da simulação de sinais mioelétricos**. 2015. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Universidade da Beira Interior, Covilhã, 2015.

MULTILOGICA-SHOP. **Módulo controlador duplo de motores TB6612FNG**. Disponível em: <https://multilogica-shop.com/modulo-controlador-duplo-de-motores-tb6612fng> Acesso em: 09 nov. 2021.

OTTOBOCK. **Myoelectric Prosthetics Hands**. Alemanha, 2013. Disponível em: https://www.Ottobock.co.uk/prosthetics/info_for_new_amputees/prosthetic-technology-explained/function___action/. Acesso em: 6 out. 2020.

PIBORG. **Thunderborg**. Disponível em: <https://www.piborg.org/motor-control-1135/thunderborg>. Acesso em: 24 set. 2020a.

PIBORG. **Ultraborg**. Disponível em: <https://www.piborg.org/sensors-1136/ultraborg>. Acesso em: 24 set. 2020b.

POLOLU Robotics & Eletronics. **TB6612FNG Dual Motor Driver Carrier**. Disponível em: <https://www.pololu.com/product/713/specs>. Acesso em: 24 set. 2020.

RASPBERRY. **Raspberry Pi 3 Model B +**. 2021. Disponível em: <https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>. Acesso em: 4 ago. 2021.

ROBOJAX. **Control Servo motor using ESP32**. Disponível em: ESP32-Arduino-Servo-Library Acesso em: 4 ago. 2019.

ROMMERS, G. M.; VOS, L. D.; GROOTHOFF, J. W.; SCHUILING, C. H.; EISMA, W. H. Epidemiology of lower limb amputees in the north of The Netherlands: aetiology, discharge destination and prosthetic use. **Prosthet Orthot Int.**, Netherlands, v. 21, n. 2, p.92-9, Aug 1997.

SIEBEN, V. **A High Power H-Bridge**. ARVP - Autonomous Robotic Vehicle Project, 2003. 22 p. Disponível em: <https://aws1.discourse-cdn.com/arduino/original/3X/a/5/a5fd757a24790660a9b4ad6aae68235f565ffb18.pdf>. Acesso em: 09 nov. 2021.

SILVEIRA, C. B. **Servo Motor**: veja como funciona e quais os tipos. Citisystems. São Paulo, 2012. Disponível em: <https://www.citisystems.com.br/servo-motor/>. Acesso em: 1 out. 2020.

SOUZA, J. O. de O. de. **Investigação de diferentes métodos**. 2021. Tese (Doutorado em Computação Aplicada) – Programa de Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2021.

SPARKFUN Start Something. **Serial Communication**. Colorado, 2021. Disponível em: <https://learn.sparkfun.com/tutorials/serial-communication/all>. Acesso em: 03 nov. 2021.

SPARKFUN Start Something. **Sparkfun Servo pHAT for Raspberry Pi**. Colorado, 2020. Disponível em: <https://www.sparkfun.com/products/15316>. Acesso em: 23 set. 2020.

THOMAS Publishing Company. **Types of Motor Controllers and Drives**. Disponível em: <https://www.thomasnet.com/articles/instruments-controls/types-of-motor-controllers-and-drives/>. Acesso em: 5 set. 2020.

THUNKABLE. **Thunkable Platform**. Disponível em: <https://thinkable.com/#/>. Acesso em: 5 set. 2021.

UMANS, S. D. **Máquinas elétricas**. 7. ed. Porto Alegre: AMGH, 2014.

YONG, X.; JING, X.; JIANG, Y.; YOKOI, H.; KATO, R. Tendon drive finger mechanisms for an EMG prosthetic hand with two motors. In: INTERNATIONAL CONFERENCE ON BIOMEDICAL ENGINEERING AND INFORMATICS (BMEI), 7th. **Proceedings...** Dalian, China, 2014.

APÊNDICE A – DRIVERS PARA MOTORES COMERCIAIS

Este apêndice descreve os drivers comerciais pesquisados para o desenvolvimento deste trabalho:

a) Driver Motor DC

Os controladores e drivers para motor DC têm a finalidade de ajustar a entrada de tensão de um sinal alternado ou contínuo para uma saída pulsada, saída em corrente contínua com duração de pulso variável ou frequência de saída. Comercialmente, as especificações do driver contém sua aplicação, modo de operação, tipo de motor, classe de tensão, interface de comunicação, bem como especificações de entrada e saída de tensão (THOMAS, 2020). Na Figura 39 é apresentado um dos modelos comerciais de driver do tipo Ponte H;

Figura 39 – Driver Motor DC Ponte H



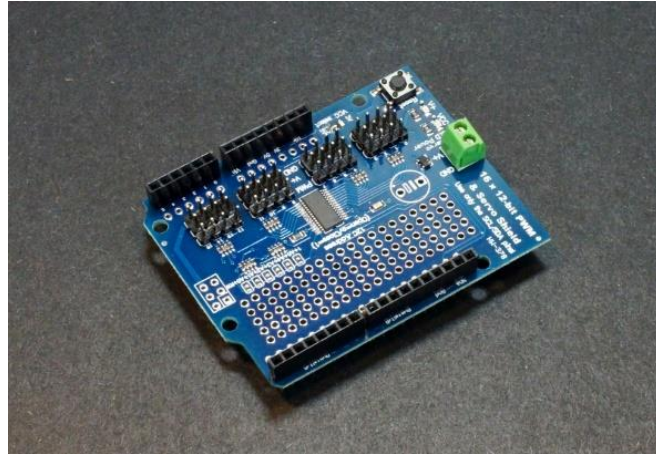
Fonte: Filipeflop (2021).

b) Driver Servomotor

Assim como os drivers para motor DC, os *servodrives* também modulam seu sinal de saída em pulso, variação de pulso ou variação de frequência. Estes são especificados comercialmente através da aplicação desejada, tipo de motor, modo de operação do drive, taxa de potência, tipo de sinal de saída, interface de comunicação e outras especificações elétricas. Os controladores e drivers para servomotor são usados em aplicações de controle de movimento, controle de posição, velocidade e torque e podem ser tanto AC como DC. Servomotores são usados em aplicações como máquinas ferramentas, micro posicionamento, robótica ou sistemas de *drive*

spindle (THOMAS, 2020). A Figura 40 apresenta um modelo de *Shield PWM* que serve para ser usada em combinação com um microcontrolador ou circuito de controle externo;

Figura 40 – Shield PWM Servo Motor



Fonte: Proto Supplies (2020).

c) Controlador Thunderborg

O controlador DC/DC Thunderborg (Figura 41), fabricado pela PiBorg, é um driver para motor DC usado em aplicações robóticas em conjunto com a Raspberry Pi. Compatível com a maioria dos modelos de Raspberry, o Thunderborg é capaz de controlar dois motores de até 5 Ampères, com variação de velocidade e sentido de rotação através de sinal PWM. Este modelo de driver usa os pinos I2C SCK/DAS e 3.3 V/GND para comunicar com a Raspberry (PIBORG, 2020a).

Figura 41 – Controlador Thunderborg



Fonte: PiBorg (2020a).

d) Controlador ZeroBorg Basic

Usada em combinação com a RaspBerry Pi Zero, a ZeroBorg é uma placa de controle de motores DC de pequeno tamanho utilizada em projetos de pequenos robôs. A ZeroBorg, exibida na Figura 42, é capaz de controlar até 4 motores independentes de até 2 A de pico de corrente, sendo possível acoplar várias placas para controlar quantos motores forem necessários. A placa possui adicionais como sensor infravermelho para controlar os motores via controle remoto. A troca de informações com a RaspBerry Pi Zero é feita através de comunicação I²C;

Figura 42 – Controlador ZeroBorg Basic



Fonte: PiBorg (2020).

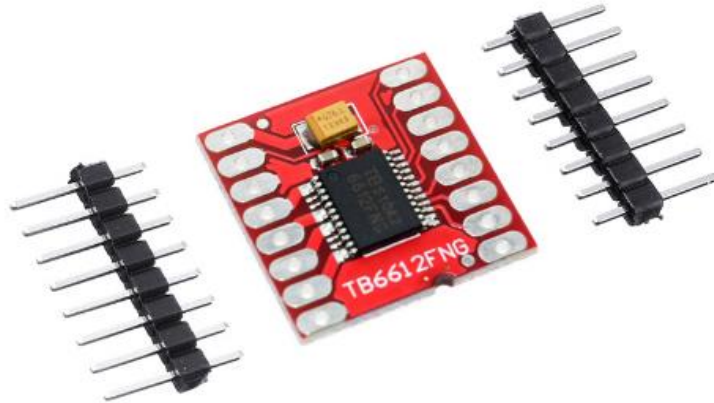
e) Módulo L298n

O chip L298n, apresentado na Figura 14 do item 2.7.1, é um driver para motor DC do tipo ponte h, geralmente utilizado com uma placa Arduino. O L298n é capaz de controlar até dois motores individualmente como uma corrente de pico máxima de até 3 A. O modelo Ponte H L298n Arduino permite controlar o sentido de rotação e velocidade dos motores através de um sinal PWM enviado pelo microcontrolador.

f) Controlador TB6612FNG

O módulo controlador TB6612FNG, conforme Figura 43, é um driver para motor DC com escovas que utiliza um circuito base de Mosfets do tipo ponte h. A placa possui pinos de inversão de sentido de giro e também de pulsos PWM com frequência de até 100 kHz e é capaz de acionar motores de até 3 A de pico. O TB6612FNG é indicado para aplicações de menor escala e tensão que a ponte h L298n. (POLOLU, 2020).

Figura 43 – TB6612FNG

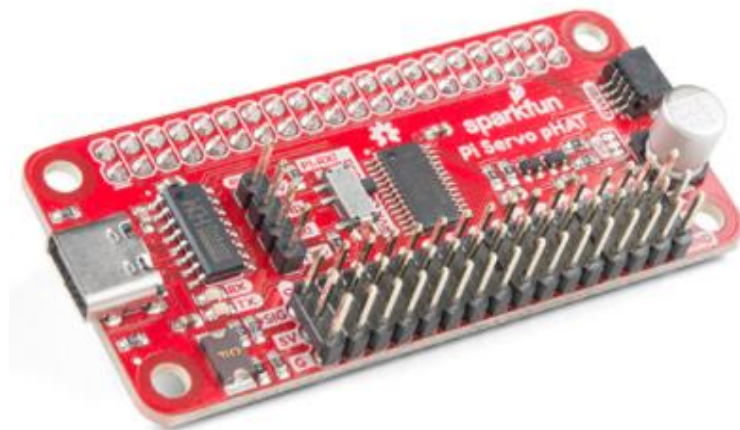


Fonte: Multilógica (2020).

g) Controlador SparkFun Servo pHAT

A SparkFun Servo pHAT, apresentada na Figura 44, é uma placa utilizada para controle de servomotores combinada com uma Raspberry Pi, via comunicação I²C. Este driver possui 16 canais PWM para controle de servomotores (SPARKFUN, 2020).

Figura 44 – Pi Servo pHAT



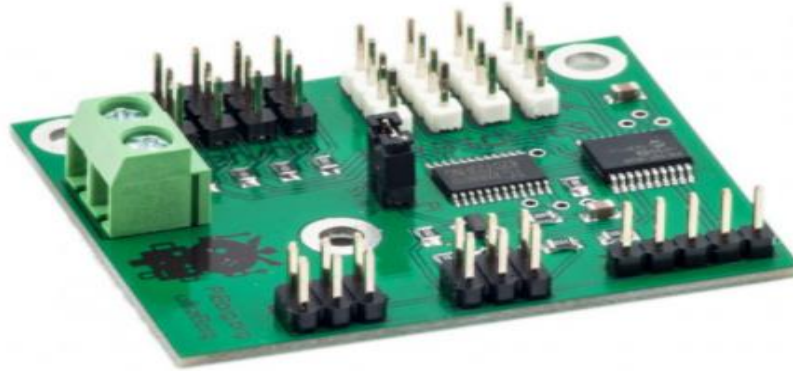
Fonte: SparkFun (2020).

h) Controlador Ultraborg

A Ultraborg (Figura 45) é uma placa de controle para servomotores fabricada pela PiBorg. Ela pode operar em conjunto com uma Raspberry Pi ou Arduino por comunicação I²C. Este driver é capaz de controlar até 4 servomotores simultâneos através de saídas PWM de 16 bits. A placa Ultraborg também possui a funcionalidade sensores ultrassônicos de posicionamento. Além do controle dos motores, a placa permite salvar

dados de limites máximos e mínimos de posição em uma memória EEPROM (PIBORG, 2020b).

Figura 45 – Ultraborg



Fonte: PiBorg (2020b).

i) Tabela Comparativa

A Tabela 5 apresenta uma comparação entre as placas de drives descritas nos itens anteriores. A tabela não trata de uma comparação entre qual modelo possui mais recursos que o outro, mas sim de um resumo de suas funcionalidades e modos de aplicação.

Tabela 5 – Modelos comerciais

| Modelo | Tipo de Controle | Número de Motores Acionados | Corrente Máxima de Saída (A) | Comunicação | Alimentação (Vdc) |
|----------------|------------------|-----------------------------|------------------------------|------------------|-------------------|
| Thunderborg | Motor DC | 2 | 5 | I ² C | 7 a 30 |
| ZeroBorg Basic | Motor DC | 4 | 2 | I ² C | 9 |
| L298n | Motor DC | 2 | 2 | I ² C | 5 |
| TB6612FNG | Motor DC | 2 | 3 | I ² C | 2,7 a 5,5 |
| pHAT | Servo | 1 | * | I ² C | * |
| Ultraborg | Servo | 4 | * | I ² C | 5 |

*Não informado pelo fabricante.

Fonte: Elaborado pelo autor (2021).

APÊNDICE B – CUSTOS DO PROJETO

A Tabela 6 descreve os custos dos materiais utilizados na montagem da prótese.

Tabela 6 – Custos do projeto

| Item | Modelo - Fabricante | Quantidade (un) | Custo unitário | Total |
|--------------------------------------|------------------------|-----------------|----------------|---------------------|
| Prótese | - | 1 | R\$ 395,00 | R\$ 395,00 |
| Servomotor | FS 0307 - Feetech | 5 | R\$ 15,40 | R\$ 77,00 |
| Microcontrolador | ESP32 - Espressif | 1 | R\$ 73,90 | R\$ 73,90 |
| Kit protoboard + regulador de tensão | - | 1 | R\$ 45,00 | R\$ 45,00 |
| Parafuso M3 | - | 12 | R\$ 0,15 | R\$ 1,80 |
| Custos com frete | - | - | - | R\$ 308,85 |
| Imposto sobre operações financeiras | - | 1 | R\$ 24,33 | R\$ 24,33 |
| Imposto de importação | - | 1 | R\$ 243,50 | R\$ 243,50 |
| Fonte de tensão | Hipro | 1 | * | R\$ 0,00 |
| Raspberry Pi | Raspberry Pi 3 B | 1 | * | R\$ 0,00 |
| | | | Total | R\$ 1.169,38 |

* Emprestado pela Unisinós

Fonte: Elaborado pelo autor (2021).