

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**EDUARDA LINHARES MELLO**

**MILENA APARECIDA BUBA**

**REDUÇÃO DE CUSTOS PARA AMPLIAR O ACESSO À IMPRESSÃO  
BRAILLE: UMA INICIATIVA DE TECNOLOGIA ASSISTIVA INCLUSIVA**

**CURITIBA**

**2024**

**EDUARDA LINHARES MELLO  
MILENA APARECIDA BUBA**

**REDUÇÃO DE CUSTOS PARA AMPLIAR O ACESSO À IMPRESSÃO  
BRAILLE: UMA INICIATIVA DE TECNOLOGIA ASSISTIVA INCLUSIVA**

**Cost Reduction to expand access to Braille Printing: An Inclusive Assistive  
Technology initiative**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Rubens Alexandre de Faria

**CURITIBA**

**2024**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**EDUARDA LINHARES MELLO  
MILENA APARECIDA BUBA**

**REDUÇÃO DE CUSTOS PARA AMPLIAR O ACESSO À IMPRESSÃO  
BRAILLE: UMA INICIATIVA DE TECNOLOGIA ASSISTIVA INCLUSIVA**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 27/06/2024

---

Rubens Alexandre de Faria  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Clayton Moura Belo  
Mestrado  
Universidade Tecnológica Federal do Paraná

---

Lucas Sangoi Mendonça  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Luciane Agnoletti dos Santos Pedotti  
Doutorado  
Universidade Tecnológica Federal do Paraná

**CURITIBA  
2024**

Dedicamos este trabalho a todos que nos incentivaram a buscar nossos sonhos. Nosso sincero agradecimento à nossa família e amigos pelo apoio constante, e aos professores e orientadores por suas valiosas instruções ao longo desta jornada.

## **AGRADECIMENTOS**

Primeiramente, agradecemos às nossas famílias e amigos, cuja presença constante e apoio foram fundamentais para que chegássemos até aqui. Aos amigos, que com palavras de incentivo, tornaram essa jornada mais leve e prazerosa.

Agradecemos também à Universidade Tecnológica Federal do Paraná e a todos os professores que contribuíram para nossa formação acadêmica. Em especial, ao nosso professor orientador, pela orientação dedicada, paciência e valiosas contribuições ao longo deste trabalho. Aos colegas de curso e a toda a comunidade acadêmica, que proporcionaram um ambiente de aprendizado enriquecedor e desafiador.

Cada palavra de incentivo, cada gesto de apoio e cada momento compartilhado foi essencial para que pudessemos superar os desafios.

*A tecnologia deve existir para todos. O verdadeiro avanço é quando ela se torna acessível e inclusiva, transformando vidas e quebrando barreiras. (Pichai, 2020)*

## RESUMO

A inclusão de pessoas com deficiência visual na sociedade depende, entre outros fatores, do acesso a materiais em braille. No entanto, a produção desses materiais é frequentemente limitada por equipamentos caros e de difícil acesso. Nesse contexto, o presente trabalho propõe o desenvolvimento de uma impressora Braille de baixo custo. O objetivo é fornecer uma solução acessível que possa ser utilizada por instituições e indivíduos para a produção de textos em braille. A impressora utiliza o Arduino para controlar motores de passo e um solenóide, que realizam a impressão dos caracteres braille em papel. Foram implementadas diversas configurações e ajustes para otimizar a qualidade da impressão e a eficiência do dispositivo. A pesquisa foi orientada pela revisão da literatura e análise de projetos existentes, além de testes práticos para garantir a funcionalidade e acessibilidade do protótipo final. Os resultados indicam que é possível desenvolver uma impressora braille funcional e econômica, que pode contribuir significativamente para a disseminação do conhecimento e inclusão social de pessoas com deficiência visual.

**Palavras-chave:** braille; impressora de baixo custo; inclusão social; deficientes visuais.

## ABSTRACT

The inclusion of visually impaired individuals in society depends, among other factors, on access to braille materials. However, the production of these materials is often limited by expensive and hard-to-access equipment. In this context, the present work proposes the development of a low-cost braille printer. The objective is to provide an accessible solution that can be used by institutions and individuals for the production of braille texts. The printer uses Arduino microcontrollers to control stepper motors and a solenoid, which perform the printing of braille characters on paper. Various configurations and adjustments were implemented to optimize the print quality and efficiency of the device. The research was guided by a literature review and analysis of existing projects, as well as practical tests to ensure the functionality and accessibility of the final prototype. The results indicate that it is possible to develop a functional and economical braille printer that can significantly contribute to the dissemination of knowledge and social inclusion of visually impaired individuals.

**Keywords:** braille; low cost printer; social inclusion; visually impaired.

## LISTA DE FIGURAS

Figura 1 – Arduino Uno R3. . . . .	26
Figura 2 – Esquemático da placa de desenvolvimento. . . . .	26
Figura 3 – Funcionamento do relé. . . . .	27
Figura 4 – Módulo relé de 1 canal. . . . .	27
Figura 5 – Esquemático do acionamento. . . . .	28
Figura 6 – Módulo optoacoplador de 2 canais. . . . .	29
Figura 7 – Esquemático do módulo optoacoplador de 2 canais. . . . .	29
Figura 8 – Módulo elevador de tensão XL6009. . . . .	30
Figura 9 – Esquemático do módulo elevador de tensão. . . . .	30
Figura 10 – Módulo <i>driver</i> L298N. . . . .	31
Figura 11 – Esquemático do módulo <i>driver</i> . . . . .	31
Figura 12 – Motor de passo NEMA 17. . . . .	32
Figura 13 – Esquemático dos motores. . . . .	32
Figura 14 – Mini trava solenóide. . . . .	33
Figura 15 – Esquemático da trava solenóide. . . . .	33
Figura 16 – Esquemático da alimentação. . . . .	34
Figura 17 – Células Braille correspondentes a cada letra do alfabeto. . . . .	35
Figura 18 – Tamanhos e espaçamentos da célula Braille. . . . .	36
Figura 19 – Mapeamento feito no código dos caracteres em questão. . . . .	36
Figura 20 – Declaração de constantes e variáveis. . . . .	37
Figura 21 – Uso dos motores de passo. . . . .	37
Figura 22 – Configuração e inicialização. . . . .	38
Figura 23 – Código que monitora a entrada de dados. . . . .	38
Figura 24 – Processamento da mensagem. . . . .	39
Figura 25 – Processamento de caracteres e números. . . . .	40
Figura 26 – Controle de motores e solenóide. . . . .	41
Figura 27 – Posição inicial. . . . .	41
Figura 28 – Partes móveis. . . . .	42
Figura 29 – Corte da estrutura. . . . .	43
Figura 30 – Posicionamento da folha. . . . .	44

<b>Figura 31 – Frase impressa.</b> . . . . .	<b>45</b>
<b>Figura 32 – Diagrama de Gantt.</b> . . . . .	<b>68</b>
<b>Figura 33 – Esquemático Impressora Braille.</b> . . . . .	<b>70</b>

## LISTA DE TABELAS

<b>Tabela 1 – Impressoras braille disponíveis no mercado . . . . .</b>	<b>18</b>
<b>Tabela 2 – Materiais da estrutura . . . . .</b>	<b>20</b>
<b>Tabela 3 – Materiais elétricos e eletrônicos . . . . .</b>	<b>20</b>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Objetivos</b>	<b>14</b>
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
<b>1.2</b>	<b>Estrutura do trabalho</b>	<b>14</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>16</b>
<b>2.1</b>	<b>Tecnologia assistiva inclusiva</b>	<b>16</b>
<b>2.2</b>	<b>O sistema de impressão braille</b>	<b>17</b>
<b>2.3</b>	<b>Mercado de tecnologia assistiva e acessibilidade</b>	<b>17</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>19</b>
<b>3.1</b>	<b>Materiais necessários</b>	<b>19</b>
<b>3.2</b>	<b>Desenvolvimento do algoritmo</b>	<b>21</b>
<b>3.3</b>	<b>Elaboração do <i>hardware</i></b>	<b>21</b>
<b>3.4</b>	<b>Cronograma</b>	<b>22</b>
<b>3.5</b>	<b>Análise de riscos</b>	<b>23</b>
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>25</b>
<b>4.1</b>	<b><i>Hardware</i></b>	<b>25</b>
4.1.1	Componentes do <i>hardware</i>	25
4.1.1.1	Microcontrolador	25
4.1.1.2	Módulos e <i>drivers</i>	26
4.1.1.3	Atuadores	30
4.1.2	Construção do <i>hardware</i>	33
<b>4.2</b>	<b><i>Firmware</i></b>	<b>34</b>
4.2.1	Tradução Braille	35
4.2.2	Construção do <i>firmware</i>	37
<b>4.3</b>	<b>Estrutura mecânica</b>	<b>41</b>
4.3.1	Partes móveis	42
4.3.2	Estrutura externa	42
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>44</b>
<b>6</b>	<b>CONCLUSÕES</b>	<b>46</b>

<b>7</b>	<b>PERSPECTIVAS DE TRABALHOS FUTUROS . . . . .</b>	<b>47</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>48</b>
	<b>APÊNDICE A CÓDIGO COMPLETO DA IMPRESSORA BRAILLE . . . .</b>	<b>51</b>
	<b>APÊNDICE B DIAGRAMA DE GANTT . . . . .</b>	<b>68</b>
	<b>APÊNDICE C ESQUEMÁTICO ELÉTRICO . . . . .</b>	<b>70</b>

## 1 INTRODUÇÃO

A acessibilidade à informação escrita é uma necessidade fundamental para a inclusão social e educacional de pessoas com deficiência visual. A tecnologia de impressão em Braille desempenha um papel crucial nesse contexto, pois permite que textos em formato tátil sejam acessíveis a essas pessoas, promovendo autonomia e igualdade de oportunidades. No entanto, as impressoras Braille tradicionais têm custos elevados, o que limita o acesso de muitas instituições e indivíduos a essa tecnologia (HILL, 2020).

O Braille, sistema de leitura e escrita desenvolvido por Louis Braille no século XIX, continua sendo uma ferramenta indispensável para a alfabetização e comunicação das pessoas cegas. Apesar do avanço das tecnologias assistivas, como os leitores de tela, o Braille mantém sua relevância, especialmente em contextos educacionais e profissionais. Portanto, a democratização do acesso a impressoras Braille pode ampliar significativamente as oportunidades educacionais e de desenvolvimento pessoal para os deficientes visuais (FRANKS, 2017).

A proposta deste projeto é desenvolver uma impressora Braille de baixo custo, utilizando componentes eletrônicos acessíveis e técnicas de fabricação digital. Este dispositivo visa oferecer uma solução mais econômica e acessível, sem comprometer a qualidade e a funcionalidade necessárias para atender às demandas de usuários individuais e instituições (SMITH, 2019).

Atualmente, as impressoras Braille no mercado utilizam tecnologias sofisticadas e componentes caros, resultando em produtos que muitas vezes estão fora do alcance de escolas, bibliotecas e famílias. Esse cenário cria uma barreira significativa para a inclusão educacional e social de pessoas cegas. A redução do custo de produção dessas impressoras é, portanto, um passo crucial para melhorar o acesso ao Braille (GONZALEZ, 2021).

Para o desenvolvimento deste projeto, foi realizada uma pesquisa extensa sobre as tecnologias existentes e suas limitações, bem como uma análise dos componentes eletrônicos e mecânicos que poderiam ser utilizados para reduzir os custos sem sacrificar a eficiência. A pesquisa incluiu uma revisão de artigos publicados entre 2000 e 2022, buscando identificar soluções inovadoras e acessíveis para a fabricação de dispositivos de impressão Braille (JOHNSON, 2018).

A impressora proposta utilizará microcontroladores como o Arduino, juntamente com mecanismos de atuação baseados em motores de passo, que serão controlados para realizar a impressão precisa dos pontos em Braille. Para a estrutura do dispositivo será utilizado um molde de fibra de média densidade, do inglês *medium density fiberboard* (MDF), de um projeto de código aberto. Além disso, o sistema será projetado para ser facilmente montado e mantido, proporcionando uma opção prática e acessível para os usuários (WILLIAMS, 2019).

Este projeto não só visa desenvolver uma solução técnica, mas também contribuir para a inclusão social e educacional de pessoas com deficiência visual, oferecendo-lhes uma ferramenta indispensável para o acesso à informação e à educação (HARRIS, 2020).

## 1.1 Objetivos

Aqui estão expostos o objetivo geral e os objetivos específicos do projeto proposto neste estudo.

### 1.1.1 Objetivo geral

Desenvolver um protótipo de impressora Braille de baixo custo, utilizando componentes eletrônicos acessíveis, para democratizar o acesso à informação escrita em Braille.

### 1.1.2 Objetivos específicos

Os objetivos específicos são:

- Desenvolver um *hardware* capaz de imprimir em relevo o que deve ser escrito em *Braille*;
- Elaborar um *firmware* que permita comunicação da impressora com o computador;
- Desenvolver um protótipo do dispositivo;
- Realizar testes funcionais para validação de *hardware* e *firmware*;

A elaboração deste trabalho propõe uma pesquisa exploratória sobre o tema proposto, para que a execução seja funcional e atenda aos requisitos sobre tecnologia assistiva.

## 1.2 Estrutura do trabalho

O seguinte documento é composto por sete capítulos que exploram o desenvolvimento de uma impressora Braille de baixo custo.

No Capítulo 1, a introdução, o problema da acessibilidade à impressão Braille é contextualizado, destacando a importância da tecnologia assistiva na inclusão social de pessoas com deficiência visual. A definição dos objetivos gerais é apresentada em seguida, os quais, visam desenvolver um protótipo de impressora Braille acessível, e específicos, que incluem o desenvolvimento de *hardware*, *firmware*, construção do protótipo e realização de testes funcionais.

O Capítulo 2, a revisão bibliográfica, aborda a tecnologia assistiva inclusiva, explicando conceitos fundamentais e apresentando exemplos relevantes, juntamente com uma análise das tecnologias atuais de impressão Braille. Além disso, uma análise de mercado oferece uma visão abrangente sobre o cenário atual e as inovações existentes na área.

Então o Capítulo 3, trata sobre a metodologia, onde são descritos os estudos das tecnologias existentes, o levantamento dos materiais necessários, o desenvolvimento do algoritmo

para conversão de texto para Braille. Métodos de teste e validação são detalhados, juntamente com o cronograma de atividades e a análise dos riscos envolvidos no projeto.

Já no Capítulo 4, o desenvolvimento, concentra-se na construção do *hardware* e *firmware* da impressora. São detalhados os componentes utilizados, a interface com o usuário, e a estrutura mecânica do dispositivo.

Culminando no Capítulo 5, os resultados e discussões, no qual são discutidos os avanços e resultados proporcionados pelo protótipo.

O Capítulo 6, as conclusões, resume as principais descobertas e realizações do projeto.

Finalmente, no Capítulo 7, as perspectivas de trabalhos futuros, são explorados avanços e integrações futuras, como o uso de inteligência artificial (IA) para aprimorar a funcionalidade da impressora Braille, melhorias em nível de mecânica as quais facilitem ainda mais a utilização e aplicação de material mais leve e sustentável.

As referências e anexos complementam o documento, fornecendo suporte teórico e documentação técnica detalhada. Estes incluem esquemáticos, códigos-fonte e outros materiais relevantes para a compreensão e reprodução do projeto, garantindo que outros pesquisadores e desenvolvedores possam construir sobre o trabalho realizado.

## 2 REVISÃO BIBLIOGRÁFICA

Nesse capítulo, serão abordados conceitos e temas essenciais para uma compreensão aprofundada do contexto em que a impressora Braille está inserida. Inicialmente, será apresentado o conceito de Tecnologia Assistiva Inclusiva, destacando sua importância para a inclusão social e a promoção da igualdade de oportunidades. Em seguida, explorar-se-á o sistema de escrita e impressão Braille, incluindo suas origens e o funcionamento básico do sistema brasileiro de escrita em Braille. Por fim, será oferecida uma visão abrangente sobre o mercado atual da tecnologia assistiva, com ênfase na acessibilidade financeira dessas inovações.

### 2.1 Tecnologia assistiva inclusiva

Segundo a Secretaria Especial dos Direitos Humanos, no livro intitulado Tecnologia Assistiva, a mesma compreende a pesquisa, fabricação, uso de equipamentos, recursos e estratégias utilizadas para potencializar as habilidades funcionais das pessoas com deficiência, indo desde tarefas básicas até o desempenho de atividades profissionais (SEDH, 2009).

Tecnologias essas que dão autonomia às pessoas as quais necessitam deveriam ser acessíveis a todos os cidadãos. O acesso à informação é um direito de todos. Logo, fomentar o desenvolvimento de tecnologia financeiramente acessível é necessário no contexto social. Tarefas básicas às quais são rápidas para muitos, podem requerer mais esforço para outros, e muito mais esforço para os que não conseguem ter acesso às tecnologias para facilitar seu dia-a-dia.

"Para garantir o direito da pessoa com deficiência, cabe à sociedade buscar conhecimento e ofertar as ferramentas de acessibilidade e o apoio necessários e adequados para seu desenvolvimento", diz Viviane Machado, da Federação das Entidades Assistenciais de Campinas, em matéria intitulada "Os desafios do sistema Braille no Brasil" para o Grupo de Instituições, Fundações e Empresas (GIFE, 2023).

Na mesma matéria citada anteriormente, Daniela Reis Frontera exemplifica como o Braille a ajuda; "Aqui em casa eu não fico lendo livros em Braille o tempo todo, mas o Braille facilita a vida da gente. Eu rotulo as coisas e aí passo o dedo e sei o que é. Uma pasta de documentos, uma cor de blusa no cabide, um condimento, você não precisa ficar toda hora chamando, ou apontando a câmera. Te dá autonomia.", diz a para-atleta e palestrante.

Com isso, reflete-se sobre a necessidade de tecnologias mais voltadas ao dia-a-dia das pessoas com deficiência visual, promovendo equidade, independência e bem-estar social, tornando assim os ambientes mais inclusivos e acessíveis a todos.

## 2.2 O sistema de impressão braille

O sistema de escrita Braille representa para as pessoas cegas um dispositivo emancipatório e desafiador, ampliando as formas de se chegar à informação. Com essa reflexão inicial, pode-se facilmente chegar à conclusão do quão importante é a escrita Braille para a comunidade de deficientes visuais, sistema criado por um jovem francês, Louis Braille (Dias de Sá, 2001).

O inventor desse sistema foi um jovem cego, que enfrentou a dificuldade da comunicação. Em meados de 1825, a escola que Louis Braille estudava fazia uso do Código Militar, invenção do oficial do exército francês Charles Barbier para utilização em campo de batalha. Esse sistema utilizava um retângulo com 6 pontos táteis em relevo, sendo 2 pontos de largura e 3 pontos de altura, para facilitar a comunicação entre oficiais quando estavam no escuro. Estas, então, foram as origens do sistema que depois de simplificado por Louis em 1837 e posteriormente padronizado no Congresso Internacional de Paris somente em 1870, José Alvarez de Azevedo trouxe ao Brasil para ser implantado (BOCK; SILVA, 2013).

Segundo a 3ª Edição da Grafia Braille para a Língua Portuguesa publicada pelo Ministério da Educação em 2008, a qual normatiza o sistema Braille para Língua Portuguesa, o sistema de escrita em relevo é constituído por 63 sinais, ou 64 se considerarmos a célula vazia como sinal, formados a partir do sinal fundamental, onde os 6 pontos dispostos em retângulo formam uma célula Braille (MEC, 2018).

Com o formato, é possível obter um alfabeto, letras com diacríticos, pontuações e sinais, números e alguns sinais exclusivos ao sistema.

Diversas são as regras para escrita e grafia correta, porém será feita uma referência apenas aos mais simples no protótipo, priorizando inicialmente a formação das células, tamanho e espaçamento padrão, deixando indicado as demais normas de grafia para trabalhos futuros.

## 2.3 Mercado de tecnologia assistiva e acessibilidade

O Livro Branco da Tecnologia Assistiva no Brasil, iniciativa da Secretaria de Políticas e Programas de Pesquisa e Desenvolvimento, Ministério da Ciência, Tecnologia, Inovações e Comunicações e o Departamento de Políticas e Programas para Inclusão Social, publicado em 2017, reúne propostas e desafios sobre a tecnologia assistiva no Brasil, de modo a orientar ações e investimentos no setor em 10 anos. Neste livro, é apresentada uma pesquisa sobre o uso das tecnologias assistivas onde a convite do ITS BRASIL participaram sete usuários e sete desenvolvedores, os quais indicaram que os altos impostos refletem no maior valor do produto. Considerando que, segundo a pesquisa, grande parte da população que necessita é de baixa renda ou recebe baixos salários, mesmo com nível de formação alta, o valor alto dos produtos impacta diretamente o seu acesso. Além de faltar acesso a orientação para capacitar usuários e

profissionais de diversas áreas a confeccionar soluções de baixo custo e de qualidade (GARCIA; BRASIL, 2017).

Ainda, segundo pesquisas da Organização Mundial da Saúde, o acesso não equitativo representa um grande desafio, pois 90% de quem precisa de tecnologias assistivas não tem acesso a ela, culminando em isolamento social e exclusão financeira, perpetuando essa condição (OMS, 2020).

Para ilustrar essa questão, será apresentado na Tabela 1 a seguir os preços das três impressoras braille mais populares disponíveis no mercado e será usada a mais barata delas como referência comparativa com o valor da impressora desenvolvida neste projeto.

**Tabela 1 – Impressoras braille disponíveis no mercado**

<b>Impressora</b>	<b>Preço</b>
Impressora Braille Columbia	R\$ 27.997,00
Impressora Braille Index Basic	R\$ 40.780,00
Impressora Braille Index Everest	R\$ 45.600,00

**Fonte: Autoria própria (2024).**

Por fim, é possível observar uma oportunidade de aquecimento de mercado do ramo, uma vez que o mesmo tem uma expectativa de crescimento a uma taxa de rendimento anual de 4,5% de 2022 a 2030. Esta condição se torna favorável pois possibilita investimentos em pesquisa e desenvolvimento (RODRIGUES *et al.*, 2024).

### 3 METODOLOGIA

Nesta seção, será discutida a metodologia, detalhando o processo e as ferramentas adotadas para a conclusão bem-sucedida da impressora Braille, culminando no protótipo requerido.

Primeiramente, serão descritos os materiais necessários, acompanhados de uma estimativa de custo. Em seguida, nos dois subcapítulos subsequentes, serão apresentadas as ferramentas utilizadas para o desenvolvimento do algoritmo e do *hardware*, juntamente com as especificações de *hardware* e *firmware* específicas para a impressora.

Posteriormente, será apresentado o cronograma detalhado, incluindo as atividades propostas e suas respectivas datas, bem como um Diagrama de Gantt para organização e orientação do projeto. Por fim, uma análise de riscos será realizada, permitindo a identificação e mitigação de possíveis obstáculos que possam surgir durante o desenvolvimento.

#### 3.1 Materiais necessários

Para que a impressora Braille obtivesse o menor custo possível, foram escolhidos materiais fáceis de encontrar e com baixo custo. No momento será discutido brevemente o funcionamento para que na próxima seção sejam descritos mais profundamente.

Em relação ao *hardware*, o princípio da impressão gira em torno de uma placa de desenvolvimento muito popular e de baixo custo: o Arduino, que controla dois motores de passo e uma trava solenóide. Um dos motores movimenta a folha, enquanto outro está acoplado a um sistema que movimenta o cabeçote de impressão. O cabeçote, por sua vez, é constituído de uma estrutura de MDF que o suporta. O acionamento de uma trava solenóide faz com que os pontos constituintes do Braille sejam estampados por pressão em uma folha A4 adequada.

O papel utilizado é de gramatura 120g, o que é recomendado para impressão Braille. Todos os testes foram realizados com esse tipo de papel, assegurando uma leitura eficiente e de qualidade para pessoas cegas.

Para que tudo isso se conecte, ainda são necessários alguns *drivers* de motor e módulos discutidos posteriormente, como o módulo de elevação de tensão, módulo relé e um módulo optocoplador. A estrutura, feita de MDF molda a impressora. Há eixos de aço para que as partes se movimentem, assim como polias e conectores que garantem funcionamento mecânico.

Na Tabela 2, se encontra a lista de materiais estruturais para o desenvolvimento do protótipo, juntamente com a quantidade necessária e o preço pago, em 2024, por cada item.

**Tabela 2 – Materiais da estrutura**

<b>Materiais da Estrutura</b>				
<b>Material</b>	<b>Especificação</b>	<b>Quantidade</b>	<b>Preço unitário</b>	<b>Valor total</b>
MDF	Corte a laser de uma placa de dimensão 479 x 302 x 6 mm e outra de 224 x 204 x 3 mm	1	R\$150,00	R\$150,00
Haste de metal	Haste com 6 mm e 8 mm	2	R\$15,00	R\$30,00
Acoplamento de eixo	Acoplamento rígido em alumínio, 5 x 8 mm com furo de 14mm de diâmetro externo	2	R\$9,90	R\$19,80
Polia GT2 20	Polia com furo central de 6mm e outra com 8mm	2	R\$33,56	R\$67,12
Correia	Movimentação do eixo X	1	R\$14,00	R\$14,00
Parafusos e porcas	Diversos	1	R\$5,00	R\$5,00
Abraçadeira de Nylon (Pacote)	Diversos, pequeno	1	R\$3,00	R\$3,00

**Fonte: Autoria própria (2024).**

Na Tabela 3, se encontra a lista de materiais elétricos e eletrônicos necessários para o desenvolvimento do protótipo.

**Tabela 3 – Materiais elétricos e eletrônicos**

<b>Materiais Elétricos e Eletrônicos</b>				
<b>Material</b>	<b>Especificação</b>	<b>Quantidade</b>	<b>Preço unitário</b>	<b>Valor total</b>
Jumpers e fios	Diversos	2	R\$9,00	R\$18,00
Motores de Passo	Motores de Passo NEMA 17, 4.8kgf.cm	2	R\$69,95	R\$139,90
Módulo Ponte H	Modulo Driver de Motor com ponte H dupla L298N	2	R\$19,90	R\$39,80
Módulo Relé	Módulo Relé com 1 canal, 5 V	1	R\$7,90	R\$7,90
Módulo Optoacoplador	Módulo de isolamento optoacoplado de 2 canais PC817	1	R\$12,00	R\$12,00
Módulo elevador de tensão	Módulo de elevação de tensão XL6009	1	R\$11,64	R\$11,64
Conector J4	Conector para cabo de alimentação J4	1	R\$2,20	R\$2,20
Placa de desenvolvimento	Arduino Uno	1	R\$55,00	R\$55,00
Solenóide	Mini trava solenóide 12 V, 1 A e 5 N de força	1	R\$33,25	R\$33,25
Placa Perfurada	Soldagem dos pontos GND, 12 V e 5 V. Placa perfurada pequena	1	R\$5,50	R\$5,50
Fonte de alimentação	Fonte de alimentação chaveada de 12 V e 10 A	1	R\$59,90	R\$59,90

**Fonte: Autoria própria (2024).**

Com isso, o custo real do projeto totaliza em R\$674,01, sendo um valor bem abaixo das impressoras Braille no mercado mostradas na Tabela 1, custando aproximadamente 2,5% do preço da mais barata.

### 3.2 Desenvolvimento do algoritmo

Em decorrência da utilização do Arduino Uno no projeto, todo o código e lógica foram desenvolvidos no IDE própria do Arduino, que oferece um ambiente integrado e amigável para programação e testes.

O primeiro passo do desenvolvimento foi compreender e controlar os motores de passo NEMA 17, responsáveis pelo movimento dos eixos X e Y da impressora. Para isso, utilizou-se a biblioteca 'Stepper.h', que facilita a implementação de controle sobre esses motores. Realizaram-se testes individuais para cada motor, ajustando parâmetros como velocidade, aceleração e número de passos, garantindo um movimento suave e preciso.

Com os motores de passo funcionando corretamente, o próximo componente testado foi a solenóide, responsável por criar os relevos Braille no papel. A implementação do seu acionamento e os testes subsequentes garantiram que a solenóide poderia ser acionada de maneira confiável.

Após os testes individuais bem sucedidos, iniciou-se a estruturação do algoritmo que combinasse as movimentações dos motores com a ativação da solenóide para criar as células Braille correspondentes ao texto inserido pelo usuário. As letras do alfabeto, números e alguns caracteres especiais foram mapeados para uma sequência específica de movimentos que formassem o correspondente em Braille.

Esta metodologia permitiu desenvolver um sistema eficiente e funcional, garantindo que a impressora possa converter texto digital em uma representação tátil para pessoas com deficiência visual.

### 3.3 Elaboração do *hardware*

Estudos e testes foram conduzidos para que os componentes escolhidos fossem compatíveis com os objetivos, ou seja, uma impressora Braille de baixo custo, a qual performasse uma impressão em alto relevo. Para isso, foram escolhidos motores de passo de forma que fosse possível controlar o espaçamento com exatidão e confiabilidade em conjunto com seus *drivers*. A trava solenóide de acionamento via relé, para que os pontos fossem feitos por pressão na folha A4 com força necessária para marcação das células Braille de forma simples e confiável, além de uma única fonte de alimentação que suportasse as necessidades do protótipo.

Após os testes foi identificada a necessidade da utilização de dois módulos adicionais, sendo eles um módulo de elevação de tensão para que com 24 V e um pulso mais curto, o

solenóide performasse um ponto mais marcante e tátil, e o outro um módulo optoacoplador para que o ruído indutivo proveniente de alguns dos componentes, como o solenóide e o relé não interferissem no restante do sistema.

Para que houvesse uma visão abrangente do *hardware*, após serem estabelecidos os módulos e componentes os quais compõe o protótipo, por meio de testes descritos na seção 5, foi desenvolvido o diagrama da impressora. Seu modelo em totalidade se encontra no Apêndice C deste documento.

O projeto citado anteriormente foi desenvolvido com auxílio do conjunto de ferramentas disponibilizadas pela plataforma gratuita de simulação e design de PCB EasyEDA e permitiu que alterações fossem documentadas de forma a ter um guia de montagem confiável, além da documentação necessária para a execução do projeto (EasyEDA, 2024).

Os testes relacionados à corrente e tensão máximas de alguns componentes para definição da fonte e o ajuste da força do solenóide foram executados com o auxílio de equipamentos dos laboratórios nas dependência da UTFPR, sendo utilizada, uma fonte regulável de tensão e um osciloscópio.

### 3.4 Cronograma

Para fins de organização e planejamento da execução do protótipo é imprescindível o acompanhamento de um cronograma bem estruturado. Então, uma descrição das atividades juntamente com um Diagrama de Gantt foi elaborado para dar apoio às realizações de cada etapa.

O cronograma foi dividido em 4 grupos, sendo eles o projeto mecânico e de *hardware*, *firmware*, integração e testes finais. O Diagrama de Gantt pode ser encontrado em sua totalidade no Apêndice B do documento.

Primeiramente, o projeto mecânico e de *hardware* compreende as tarefas referentes à pesquisa, compra, testes de componentes e também a compra e montagem da estrutura. O tempo estimado de conclusão dessa etapa foi de dois meses e meio, tendo início em janeiro de 2024 e sendo concluída na segunda semana de março do mesmo ano.

Então a elaboração do *firmware* teve início, sendo as tarefas elaboração, teste, validação e implementação do código. Por essa etapa ser concorrente com a de testes, foi a mais longa do planejamento, tendo início na primeira semana de março, em conjunto com os testes de *hardware* e conclusão somente na segunda semana de junho, devido à vários ajustes necessários durante o percurso do desenvolvimento. Foi seguro estender a duração da etapa em questão.

A integração teve objetivo de validar e ajustar *hardware*, *firmware* e estrutura mecânica para que funcionassem em consonância, obtendo um funcionamento conjunto. Essa etapa teve início na terceira semana de abril e concluída no final de maio.

Então, como etapa final, os testes finais foram elaborados. As tarefas constituintes incluem demais correções, funcionando como margem para integração, validação do protótipo como todo e elaboração deste documento. As possíveis correções começaram concomitantemente com os testes de integração, com previsão de término no início de junho, com uma semana de margem. Já a elaboração da estrutura do documento, pesquisas adicionais e a escrita tiveram início em março, com o *firmware*, e com previsão de conclusão na segunda semana de junho, fechando o processo de desenvolvimento da impressora Braille.

### 3.5 Análise de riscos

A análise de riscos relacionados ao projeto busca identificar e compreender os potenciais obstáculos e desafios para que seja possível mitigar problemas ao longo do desenvolvimento e operação do dispositivo. Este processo foi realizado principalmente para fornecer diretrizes sobre o que deve ser considerado caso terceiros queiram reproduzir o projeto.

Um dos principais riscos é a exatidão no encaixe das peças na estrutura mecânica, que pode comprometer a eficiência operacional da impressora. Para mitigar esse risco, optou-se por utilizar a estrutura mecânica de MDF do projeto de código aberto Cheap Braille Embosser (La Picoreuse) da Autodesk Instructables (IAPAFOTO, 2018), sabendo que a parte mecânica desse projeto funciona adequadamente. Se a equipe optasse por desenvolver a estrutura por conta própria, qualquer material comprado ou projetado no tamanho errado comprometeria a engrenagem. Além disso, no momento do uso, como em qualquer impressora, há o risco do papel embolar ou ser puxado de maneira errada, se não for inserido corretamente. Este ponto requer a atenção do usuário, que deve garantir o correto manuseio do papel durante a operação da impressora.

Outro ponto de atenção é a indisponibilidade de componentes, o que pode resultar em atrasos significativos no projeto, afetando diretamente o cronograma de desenvolvimento. Portanto, faz-se necessário encontrar fornecedores locais e ter um plano de contingência para a aquisição de componentes alternativos.

O elevado custo do projeto é um risco que deve ser considerado, tendo em vista que um dos principais objetivos do projeto é ser financeiramente viável ao público. Os custos podem exceder o orçamento planejado, especialmente se houver variações nos preços dos componentes ou dificuldades na aquisição de materiais específicos. Este risco pode ser gerenciado com um controle financeiro rigoroso e a busca por alternativas mais econômicas sem comprometer a qualidade.

A possibilidade de falha do *hardware* ou do projeto mecânico é outro desafio que deve ser enfrentado. Defeitos de fabricação, uso inadequado ou desgaste natural ao longo do tempo podem comprometer o desempenho da impressora e exigir intervenções de manutenção.

A acessibilidade do material produzido para deficientes visuais é um risco significativo. Testes com usuários não estão no escopo do presente trabalho, portanto existe a possibilidade

do material final não ser adequado, mesmo seguindo diretrizes impostas. Para abordar essa questão, recomenda-se para trabalhos futuros a inclusão de testes com usuários deficientes visuais e a consulta a especialistas em acessibilidade para garantir que o produto final atenda a todas as necessidades específicas deste grupo.

Mudanças no mercado de tecnologia assistiva representam outra ameaça importante. A evolução rápida e constante da tecnologia pode introduzir novos concorrentes, mudanças nas preferências dos consumidores ou avanços que tornem obsoletos os recursos planejados neste projeto. É essencial monitorar o mercado para antecipar essas mudanças e adaptar o projeto conforme necessário. Alterações na legislação relacionada à acessibilidade também podem exigir ajustes para garantir conformidade legal. Isso inclui padrões de acessibilidade em dispositivos tecnológicos e requisitos de *design* para produtos destinados a pessoas com deficiência.

Apesar de vislumbrados todos esses riscos, a atenção estará focada principalmente nos aspectos que possam impedir que a impressora seja produzida ou funcione tecnicamente, que é o escopo principal do projeto. Outros riscos são importantes e serão considerados para um contínuo aprimoramento futuro.

## 4 DESENVOLVIMENTO

Nesta etapa, será detalhado o processo de desenvolvimento do protótipo da impressora Braille de baixo custo. Serão incluídos diagramas e esquemáticos, além da descrição minuciosa das etapas de construção do *hardware*, desenvolvimento do *firmware* e da parte estrutural e mecânica, integrando todos esses elementos de forma coesa.

Serão explicados os critérios de seleção para os componentes de *hardware*, assim como a montagem das partes, abordando cada etapa com clareza. No desenvolvimento do *firmware*, serão descritos o processo de codificação, os testes e os ajustes necessários para garantir o bom funcionamento do sistema. Quanto à parte estrutural e mecânica, serão descritas a estrutura e as adaptações e modificações feitas no projeto original para adequar a funcionalidade e eficiência do dispositivo.

### 4.1 *Hardware*

O *hardware* da Impressora Braille é composto por um microcontrolador para ser o centro de comando do protótipo; módulo relé em conjunto com um optoacoplador e um elevador de tensão para acionamento de uma solenóide, *drivers* para os motores de passo e uma fonte de alimentação.

#### 4.1.1 Componentes do *hardware*

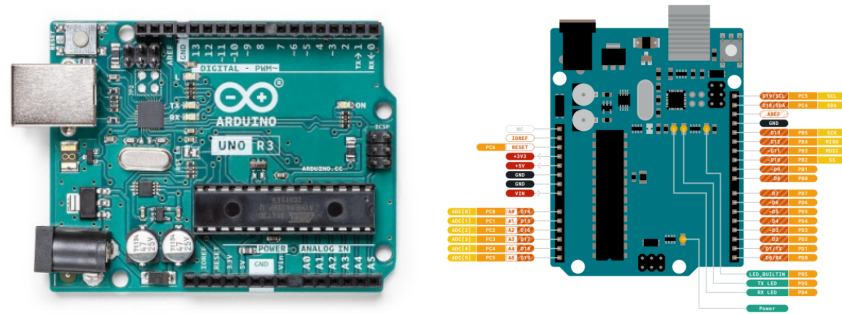
Serão melhor descritos a seguir os componentes utilizados.

##### 4.1.1.1 Microcontrolador

Um microcontrolador pode ser definido como um sistema onde estão incluídos CPU, memória de dados e programa, *clock*, portas de I/O e possíveis periféricos (DENARDIN, 2014). Estes, integrados numa variedade de equipamentos, estão presentes constantemente em diversos dispositivos digitais no nosso dia-a-dia. Devido à popularidade, estão se tornando mais acessíveis e ainda mais difundidos. Versões com mais facilidades em relação à programação aparecem no mercado e inovações do ramo representam um mercado aquecido.

A placa de desenvolvimento escolhida para o protótipo foi o Arduino Uno R3, apresentada na Figura 1 e na Figura 2. Isto é devido à facilidade de programação e utilização, custo acessível, versatilidade, compatibilidade com diversos periféricos e portas de comunicação. Ela inclui 14 pinos de I/O e o microprocessador ATmega328P da Atmel, com arquitetura RISC AVR de 8 bits (Arduino, 2024). Sendo assim suficiente para o desenvolvimento.

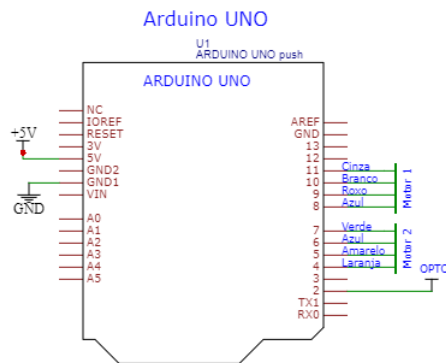
Figura 1 – Arduino Uno R3.



Fonte: (Arduino, 2024).

Na impressora Braille desenvolvida, ela atua acionando os *drivers* dos motores indicados no esquemático como motor 1 e motor 2, coordenando o sinal de acionamento indicado por OPTO. Além disso, os pinos de 5 V e GND são utilizados no restante do circuito.

Figura 2 – Esquemático da placa de desenvolvimento.



Fonte: Autoria própria (2024).

#### 4.1.1.2 Módulos e *drivers*

Módulos e *drivers* foram utilizados para permitir que equipamentos periféricos de atuação mais robusta fossem incluídos. São eles o módulo relé, optoacoplador, elevador de tensão e os *drivers* dos motores. Discutir-se-ão os mesmos, divididos em 2 funcionalidades, acionamento da solenóide e acionamento dos motores.

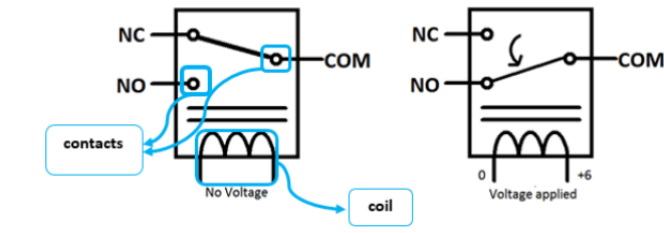
Para que o acionamento correto e eficiente da solenóide constitua a célula Braille foi necessário um conjunto de módulos.

- Módulo relé

Este equipamento tem o objetivo de acionar a trava solenóide por meio de um pino de sinal intitulado IN.

O funcionamento de um módulo relé é simples e pode ser visto na Figura 3. Um sinal enviado por um microcontrolador é recebido pelo pino IN do módulo, o qual comuta os contatos do relé. Este, tem três terminais: um comum, um normalmente fechado e um normalmente aberto. O sinal comuta o comum de normalmente fechado (NF) para normalmente aberto (NO) (Future Eletronic Corporation, 2022).

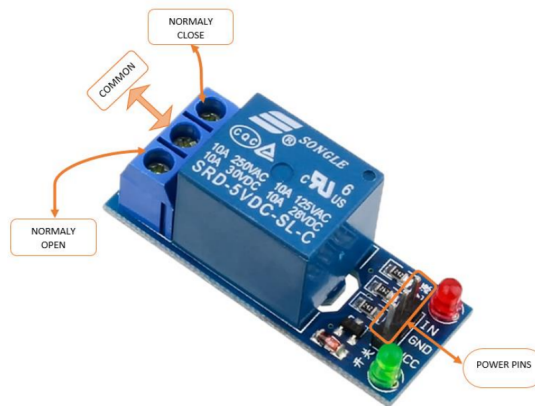
**Figura 3 – Funcionamento do relé.**



**Fonte: (Future Eletronic Corporation, 2022).**

O módulo escolhido para o protótipo, ilustrado na Figura 4, apresenta 1 canal com tensão de sinal de 3 a 5 V, tensão de operação de 5 a 12 V e limite de corrente nos contatos de 10 A. Por essas especificações, foi implementado (Future Eletronic Corporation, 2022).

**Figura 4 – Módulo relé de 1 canal.**

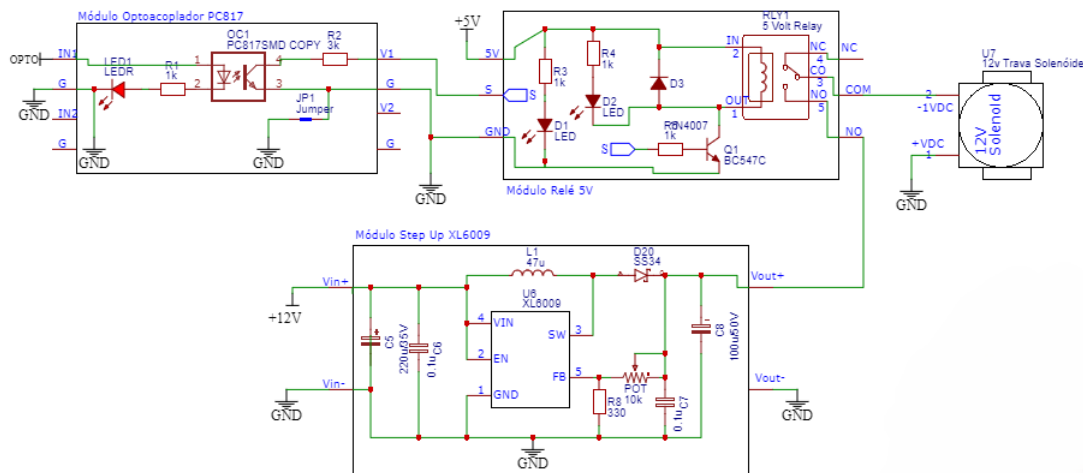


**Fonte: (Future Eletronic Corporation, 2022).**

Na Figura 5 é possível observar sua utilização. Este é conectado à saída de um optoacoplador, alimentado em 5 V do Arduino e faz comutação da tensão advinda de um elevador de tensão do pino NO com a trava solenóide no pino COM.

**Figura 5 – Esquemático do acionamento.**

Acionamento da Solenóide



Fonte: Autoria Própria (2024).

- Módulo optoacoplador

Um optoacoplador é altamente indicado em sistemas nos quais há possibilidade de ruído eletromagnético advindo dos indutivos de certos componentes, então para isolamento elétrico e proteção do circuito, um módulo contendo o componente PC817 foi incluído no esquemático. O princípio de funcionamento de um optoacoplador gira em torno de dois componentes internos do CI PC817, onde um led emite luz e um foto-transistor recebe. O led interno acende, fazendo o transistor polarizar e fechar contato na junção coletor-emissor, conduzindo. (Usinainfo, 2023).

O módulo escolhido é o da Figura 6 e apresenta 2 canais, pois foi encontrado dificuldades em obter um módulo de apenas um canal por menor preço em comparação ao de dois.

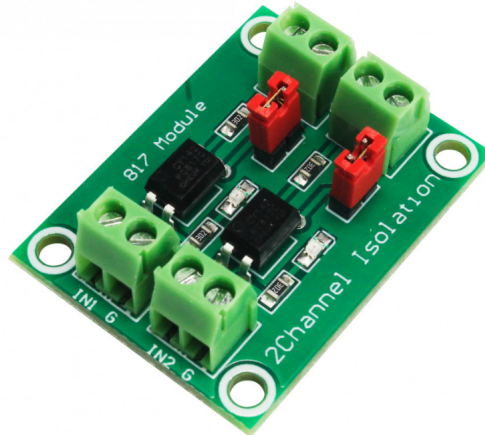
Na Figura 7 pode ser observado o sinal enviado do arduino intitulado OPTO, o qual em nível alto acende o led interno do CI PC817, polarizando o transistor interno do mesmo. Com isso, o sinal agora isolado eletricamente sai do módulo após passar por um resistor de saída, entrando no módulo relé.

- Módulo elevador de tensão

Nos testes iniciais, a solenóide era acionada em 12 V e resultava em um relevo suave, inadequado para a leitura tátil. Para melhorar a eficácia do acionamento e garantir que o ponto fosse marcado de forma condizente com as normas do Braille, testes foram feitos para descobrir qual seria a corrente ideal para um relevo satisfatório.

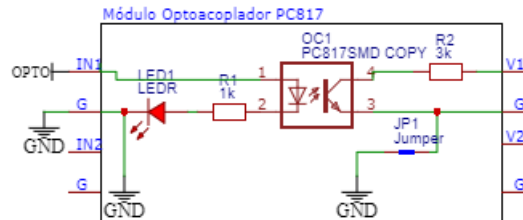
Isolando o sistema de acionamento do restante do circuito, foi aplicada uma tensão crescente no pino NO do módulo relé, e reduzindo o tempo do pulso no código, en-

**Figura 6 – Módulo optoacoplador de 2 canais.**



**Fonte: Adaptado de (Usinainfo, 2023).**

**Figura 7 – Esquemático do módulo optoacoplador de 2 canais.**



**Fonte: Autoria Própria (2024).**

quanto era monitorada a corrente consumida no solenóide e os pontos impressos eram analisados. Ao atingir 24 V, a solenóide apresentou uma impressão satisfatória consumindo 800 mA de corrente e sendo acionada por um intervalo de 25 milissegundos.

Para simplificar a experiência do usuário e evitar a necessidade de conectar duas fontes, foi incorporado um módulo elevador de tensão, também conhecido como *step-up*, ao protótipo.

Um *step-up*, contemplado na Figura 8, funciona aumentando a tensão de entrada para uma tensão de saída mais alta, utilizando um controlador de comutação de alta frequência que armazena e libera energia através de um indutor. O módulo XL6009E1 é chaveado e apresenta tensão de entrada na faixa de 3 a 32 V e de saída 5 a 35 V, com corrente máxima de 4 A. Uma vez que os módulos chaveados com LM2577 estão caindo em desuso e pelo fato de o XL6009 ter 1 A a mais de corrente máxima e eficiência de cerca de 94%, foi escolhido para integrar o sistema. (Absolute Native Electronics, 2014)

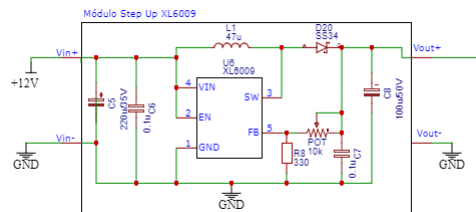
**Figura 8 – Módulo elevador de tensão XL6009.**



**Fonte: (Absolute Native Eletronics, 2014).**

A Figura 9 ilustra o módulo elevador de 12 para 24 V.

**Figura 9 – Esquemático do módulo elevador de tensão.**



**Fonte: Autoria Própria (2024).**

Para que os motores se movimentassem sem problemas, também foi necessário um módulo *driver* com ponte H dupla, que controla bidirecionalmente a direção e o fluxo de corrente, pois os motores de passo escolhidos exigiram cerca de 3 A, considerando os picos de corrente de partida.

Por esse motivo, um *driver* de motor mais robusto deve ser incorporado. Para tal, o módulo *driver* L298N, que pode ser visto na Figura 10, foi escolhido. Contando com diodos de proteção, conexão facilitada para microcontroladores e baseada no CI L298, ele permite o acionamento de 2 motores com corrente máxima de 2 A cada, sendo suficiente para integrar e acionar adequadamente os motores de passo sem aquecimento. (Handson Technology, 2018)

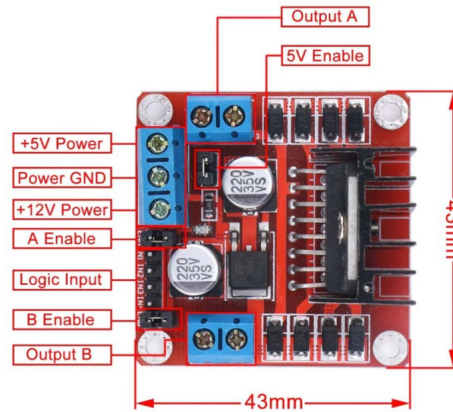
A utilização desses módulos apresentam-se no esquemático representados na Figura 11, os quais fazem a ponte entre o Arduino Uno e os motores NEMA 17.

#### 4.1.1.3 Atuadores

Atuadores podem ser definidos como elementos que reagem com o objetivo de atender comandos, sejam eles manuais ou automáticos. Existem vários tipos de atuadores, mas este texto se limitará aos de acionamento elétrico (UNICAMP, 2007).

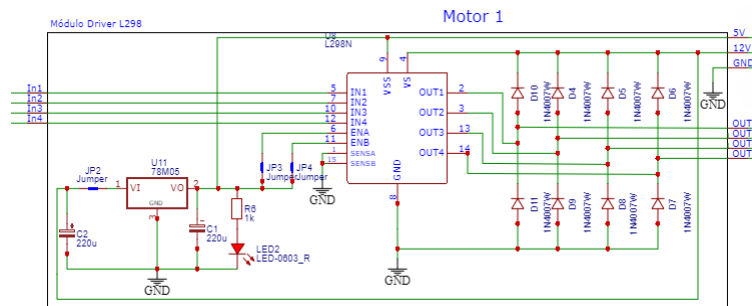
Tendo definido o termo, agora é possível abordar o motor de passo utilizado, assim como a trava solenóide.

Figura 10 – Módulo *driver* L298N.



Fonte: (Handson Technology, 2018).

Figura 11 – Esquemático do módulo *driver*.



Fonte: Autoria Própria (2024).

- Motor de passo NEMA 17

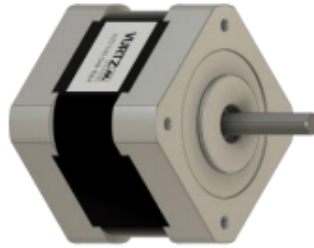
O motor de passo é capaz de controlar seus movimentos de forma precisa, onde pulsos elétricos são transformados em variações angulares (UFF, 2008). Por esse motivo, um motor de passo é ideal para aplicações onde é necessário controle de posição, como no eixo X e Y da impressora Braille proposta.

Com força radial máxima de 28 N e axial de 10 N, o motor de passo escolhido NEMA 17, ilustrado na Figura 12, apresenta conexão bipolar e ângulo de passo de 1.8°, com precisão de  $\pm 5\%$  (VURTZ Motors, 2022).

Para que o Arduino possa se comunicar com os motores, os mesmos necessitam de um *driver*, já especificado anteriormente. Os quatro fios: preto, vermelho, azul e verde representam os enrolamentos internos do motor e devem ser conectados aos respectivos pinos de saída dos *drivers* L298N, como especificado na Figura 13.

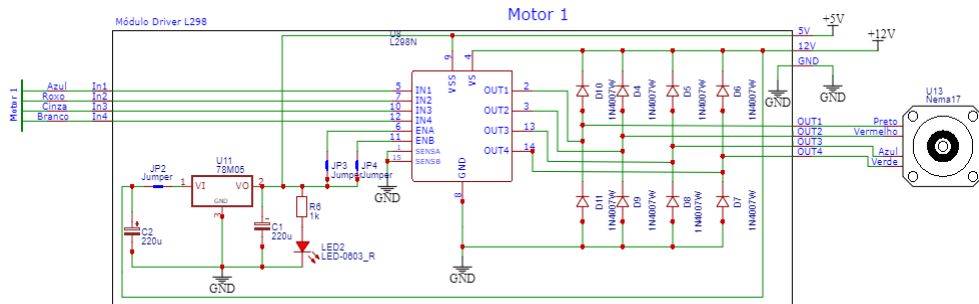
- Mini trava solenóide

**Figura 12 – Motor de passo NEMA 17.**



Fonte: (VURTZ Motors, 2022).

**Figura 13 – Esquemático dos motores.**



Fonte: Autoria Própria (2024).

De funcionamento bem simples, uma trava solenóide é constituída de um enrolamento e um pino em material ferromagnético. Energizada, a solenóide retrai o pino e assim que a corrente for cortada, o mesmo retorna à posição inicial por meio mecânico, através de uma mola. Desta maneira, seu estado original é normalmente fechado (Usinainfo, 2019).

A trava solenóide é ideal para o cabeçote de impressão, pois proporciona o pressionamento da ponteira no papel de forma rápida, simples e efetiva. Para tal, a mini trava solenóide JF-0530B, ilustrada na Figura 14, foi escolhida. Com força de 5 N, corrente nominal máxima de 1 A se torna suficiente para o objetivo. Seu tamanho reduzido também contribuiu com a escolha. Com dimensão de apenas 13 x 15 x 30 mm, é adequada para a parte móvel da estrutura, compondo a impressão no eixo X (Bit Maker, 2024).

No esquemático mostrado na Figura 15, a trava solenóide é conectada ao módulo relé, o qual juntamente com o módulo de elevação de tensão tem a função de acionar a trava, fazendo com que o pino pressione o papel, fazendo o ponto constituinte da célula Braille.

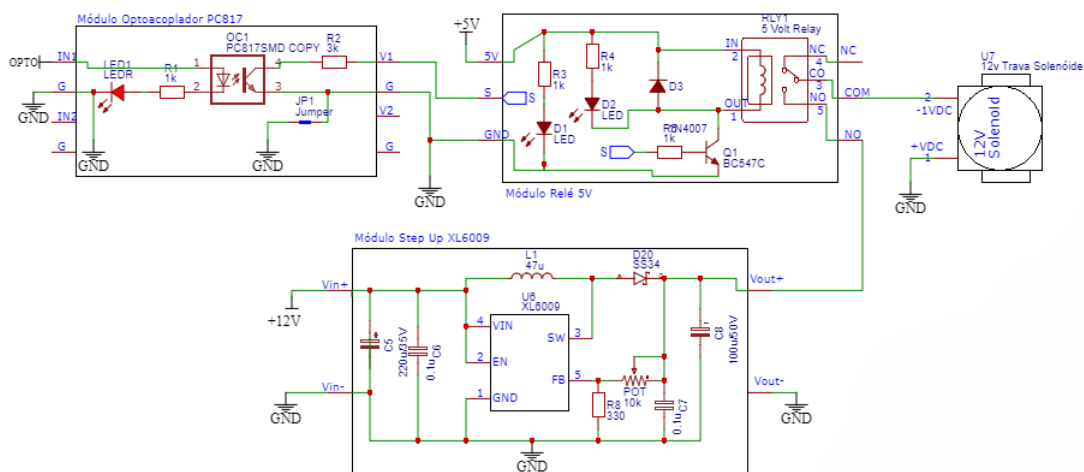
Figura 14 – Mini trava solenóide.



Fonte: Adaptado de (Bit Maker, 2024).

Figura 15 – Esquemático da trava solenóide.

Acionamento da Solenóide



Fonte: Autoria Própria (2024).

#### 4.1.2 Construção do *hardware*

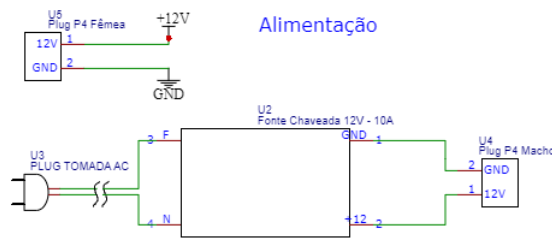
A seguir, é descrita a integração do *hardware*, explicando o funcionamento do conjunto como um todo. Lembrando que o esquemático completo se encontra nos anexos do documento.

A Figura 16 mostrana alimentação do projeto. Uma fonte chaveada de 12 V e 10 A distribui a tensão para os *drivers* L298N e para o módulo elevador de tensão, o qual eleva de 12 para 24 V, para que a solenóide desempenhe plena função.

É importante mencionar que o GND proveniente da fonte de 12 V é conectado ao GND proveniente do Arduino, formando uma só referência.

O controle do sistema fica a cargo do Arduino Uno, onde os pinos de I/O digitais de 4 a 11 coordenam os motores. O pino digital 2 envia o sinal para o circuito de acionamento da solenóide. Além disso, o pino que fornece 5 V é utilizado também nos módulos dos motores

**Figura 16 – Esquemático da alimentação.**



**Fonte: Autoria Própria (2024).**

e no módulo relé. A comunicação com o computador que envia os dados a serem impressos acontece por meio de uma interface.

Para que os motores sejam acionados devidamente, os *drivers* L298N são conectados ao Arduino. Dois motores de passo NEMA 17 são utilizados, sendo um deles para movimento de entrada e saída do papel da estrutura e o outro responsável pelo movimento do sistema que leva verticalmente o cabeçote até o local onde o ponto será marcado.

Por fim, tem-se o conjunto o qual aciona a solenóide para que o ponto Braille seja efetuado. Para isolar eletricamente o sistema e evitar problemas com ruídos, um optoacoplador faz a ponte entre o Arduino que proporciona o comando e o relé que efetivamente faz o acionamento, comutando a chave interna do comum ao normalmente aberto, fechando o circuito e acionando a trava solenóide.

O contato normalmente aberto do relé é alimentado pelos 24 V da saída do módulo elevador de tensão, fazendo com que a solenóide acione com a tensão maior que a da fonte.

## 4.2 Firmware

Neste projeto, foi utilizado o Arduino Uno como base para desenvolver um sistema de impressão Braille, no qual todo código e lógica foram elaborados na IDE própria do Arduino.

O primeiro desafio enfrentado foi a compreensão e controle dos motores de passo NEMA 17, responsáveis pelo movimento dos eixos X e Y da impressora. Para isso, foi utilizada uma biblioteca específica chamada 'Stepper.h', que auxiliou na implementação do controle desses motores, ajustando parâmetros como velocidade, aceleração e número de passos, garantindo um movimento suave e preciso.

Com os motores de passo funcionando corretamente, a atenção se voltou ao acionamento da solenóide, o componente responsável por criar os relevos Braille no papel. A implementação do acionamento e testes subsequentes asseguraram que a solenóide funcionasse.

Após a validação individual dos componentes, estruturou-se um algoritmo que combina os movimentos dos motores com a ativação da solenóide para formar as células Braille correspondentes ao texto inserido pelo usuário. As letras do alfabeto, números e alguns caracteres

especiais foram mapeados para uma sequência específica de movimentos, garantindo a correta representação do Braille.

Será detalhado nos tópicos posteriores a tradução para Braille, na qual será explicado como cada caractere é transformado em uma sequência de pontos em relevo e a construção do *firmware*, destacando as principais funções e como elas colaboram para o funcionamento do sistema de impressão Braille.

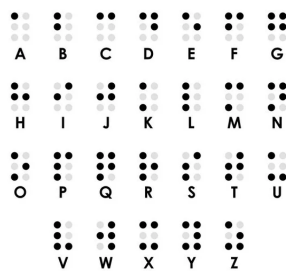
#### 4.2.1 Tradução Braille

No sistema de escrita tátil Braille, cada caractere é representado por uma célula composta por até seis pontos em relevo, dispostos em duas colunas de três linhas cada. As combinações desses pontos formam diferentes letras, números e símbolos. No presente trabalho, foi decidido contemplar as 26 letras do alfabeto, números e alguns caracteres especiais. As acentuações foram excluídas devido a limitações do monitor serial utilizado.

Cada célula Braille possui uma altura e largura padrão e o espaçamento entre as células é crucial para garantir a legibilidade, fazendo com que a resolução que movimenta os eixos X e Y e o acionamento da solenóide sejam de extrema importância.

De maneira a trazer visualização para as explicações acima, é possível observar na Figura 17, como são em Braille cada letra do alfabeto e na Figura 18 os espaçamentos e tamanhos padronizados.

**Figura 17 – Células Braille correspondentes a cada letra do alfabeto.**



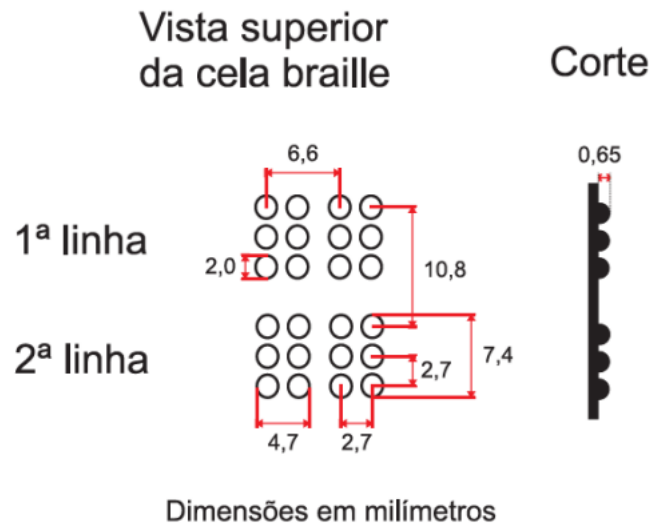
**Fonte: (Toda Matéria, 2024).**

O primeiro passo para a tradução do texto em Braille foi mapear cada caractere do alfabeto, cada número e os caracteres especiais selecionados para suas correspondentes células Braille. O código desenvolvido contém as movimentações dos motores do eixo X e Y e acionamento de solenóide específicas para cada caractere, conforme a combinação de pontos que formam a célula Braille.

É importante lembrar que pela direção que a solenóide faz pressão no papel, o relevo é criado na parte posterior do papel, sendo necessário fazer a impressão espelhada da célula Braille no código para que no resultado final esteja correto.

A Figura 19 ilustra como o mapeamento foi feito e como ele reflete no código.

Figura 18 – Tamanhos e espaçamentos da célula Braille.



Fonte: (Ministério da Educação, 2018).

Figura 19 – Mapeamento feito no código dos caracteres em questão.



Fonte: Autoria Própria (2024).

O processo de mapeamento foi feito para as 26 letras do alfabeto, números de 0 a 9 (incluindo o símbolo que antecede uma sequência de números) e os caracteres especiais vírgula, ponto e vírgula, dois-pontos, ponto, interrogação, exclamação e hífen. Através da função 'processChar(char c)' a frase inserida pelo usuário se transforma em acionamentos de motores e movimentações de solenóide para formar letra por letra.

#### 4.2.2 Construção do *firmware*

A seguir, serão detalhadas as principais partes do código, destacando as funções e a lógica por trás de cada módulo.

A Figura 20 mostra o início do código, onde são declaradas as constantes e variáveis utilizadas para configurar os motores de passo e a solenóide. A constante 'stepsPerRevolution' define o número de passos necessários para uma revolução completa do motor de passo NEMA17. Os pinos de controle dos motores e da solenóide são definidos como constantes e é incluída a biblioteca 'Stepper.h', que fornece funções essenciais para o controle do motor de passos.

**Figura 20 – Declaração de constantes e variáveis.**

```
#include <Stepper.h>

// Número de passos por revolução para nema17
const int stepsPerRevolution = 200;

// Pinos de controle dos motores
const int motorYPin1 = 8;
const int motorYPin2 = 9;
const int motorYPin3 = 10;
const int motorYPin4 = 11;

const int motorXPin1 = 4;
const int motorXPin2 = 5;
const int motorXPin3 = 6;
const int motorXPin4 = 7;
```

**Fonte: Autoria Própria (2024).**

Os motores de passo para os eixos X e Y são configurados utilizando a biblioteca 'Stepper.h'. Dois objetos da classe 'Stepper' são criados para representar os motores dos eixos X e Y, que pode ser observado na Figura 21.

**Figura 21 – Uso dos motores de passo.**

```
// Objetos Stepper para os dois motores
Stepper motorY(stepsPerRevolution, motorYPin1, motorYPin3, motorYPin2, motorYPin4);
Stepper motorX(stepsPerRevolution, motorXPin1, motorXPin3, motorXPin2, motorXPin4);
```

**Fonte: Autoria Própria (2024).**

A configuração e inicialização são feitas na função 'setup()', mostrado na Figura 22, na qual a comunicação serial é iniciada para interação com o usuário e a velocidade dos motores de passo é definida. Além disso, a solenóide é configurada como saída digital.

O 'loop()' principal do código monitora a entrada de dados do usuário via monitor serial. Cada caractere inserido é lido e armazenado em um *buffer* até que o caractere de nova linha seja detectado, indicando o final da mensagem. Pode ser visualizado na Figura 23.

Na Figura 24, a função 'processMessage()' é encarregada de processar a mensagem completa. Ela percorre o *buffer* de caracteres, identificando cada caractere e chamando funções

**Figura 22 – Configuração e inicialização.**

```
void setup() {
  delay(1000);
  Serial.begin(9600);
  Serial.println("Digite uma frase: ");
  pinMode(solenoid, OUTPUT);

  // Definição da velocidade dos motores
  motorY.setSpeed(60);
  motorX.setSpeed(60);
}
```

**Fonte: Autoria Própria (2024).**

**Figura 23 – Código que monitora a entrada de dados.**

```
void loop() {
  if (Serial.available() > 0) {
    // Lê o caractere do monitor serial
    char receivedChar = Serial.read();

    // Verifica se é o final da mensagem (Enter)
    if (receivedChar == '\n') {
      if (messageIndex > 211) {
        Serial.println("Número de caracteres acima do possível para impressão.");
      } else {
        messageBuffer[messageIndex] = '\0'; // Termina a string
        Serial.print("Número total de caracteres inputados: ");
        Serial.println(messageIndex);
        processMessage();
      }
      messageIndex = 0; // Reinicia o índice para a próxima mensagem
    } else {
      // Armazena o caractere no buffer se não for o final
      if (messageIndex < bufferSize - 1) {
        messageBuffer[messageIndex++] = receivedChar;
      }
    }
  }
}
```

**Fonte: Autoria Própria (2024).**

específicas para tratar letras, números e caracteres especiais. O controle de linhas e colunas garante que o texto seja impresso corretamente no papel A4.

**Figura 24 – Processamento da mensagem.**

```
// Função para processar toda a mensagem
void processMessage() {
  int printedChars = 0;
  int lineCount = 0;
  bool inNumberSequence = false;
  bool paperPulled = false;

  for (int i = 0; messageBuffer[i] != '\0'; i++) {
    char currentChar = messageBuffer[i];

    if (isdigit(currentChar)) {
      if (!inNumberSequence) {
        processChar('@');
        inNumberSequence = true;
        printedChars++;
      }
      processNumber(currentChar);
      Serial.println(printedChars);
    } else if (currentChar == '[' || currentChar == ']') {
      processChar(currentChar);
      paperPulled = true;
      Serial.println(printedChars);
    } else {
      inNumberSequence = false;
      processChar(currentChar);
      printedChars++;
      Serial.println(printedChars);
    }
    if (printedChars == 15) {
      moveMotorY(5); // Muda para a próxima linha
      delay(1000);
      moveMotorX(40); // volta para o início da linha
      printedChars = 0;
      lineCount++;
      Serial.println(lineCount);
    }
  }
  if (!paperPulled) {
    initialPosition(printedChars, lineCount);
  } else {
    paperPulled = false;
  }
}
}
```

**Fonte: Autoria Própria (2024).**

A Figura 25 mostra as funções 'processChar()' e 'processNumber()', que são responsáveis por determinar a ação a ser tomada para cada caractere. A função 'processChar()' identifica letras e caracteres especiais, acionando a solenóide e movimento os motores de acordo com a célula Braille correspondente.

Figura 25 – Processamento de caracteres e números.

```

void processNumber(char c) {
  switch (c) {
    case '0':
      processChar('j');
      break;
    case '1':
      processChar('a');
      break;
    case '2':
      processChar('b');
      break;
    // Casos para cada letra do alfabeto
  }
}

void processChar(char c) {
  c = toLowerCase(c);
  switch (c) {
    case '[':
      Serial.println("Pull paper:");
      pullPaper();
      break;
    case ']':
      Serial.println("Pull paper:");
      releasePaper();
      break;
    // Casos para cada letra do alfabeto
    // Outros casos para caracteres especiais
    case 'a':
      Serial.println("Letra A: ");
      embosse();
      moveMotorX(-3);
      break;
    // ... e assim por diante para as outras letras e caracteres
  }
}

```

Fonte: Autoria Própria (2024).

Funções específicas são responsáveis por controlar os motores e a solenóide, garantindo que cada célula Braille seja corretamente impressa. Na Figura 26, as funções 'moveMotorX()' e 'moveMotorY()' movimentam os eixos X e Y, respectivamente, enquanto 'embosse()' aciona a solenóide para criar os pontos em relevo. O parâmetro 'amount' na função 'moveMotorX()' corresponde a um número inteiro que irá multiplicar a movimentação de 2,7 mm, um valor negativo indica movimento para a esquerda, enquanto um valor positivo indica movimento para a direita. Da mesma forma, na função 'moveMotorY()', a variável 'amount' também corresponde ao fator de multiplicação do movimento de 2,7 mm, com valores negativos fazendo o motor girar no sentido horário e valores positivos no sentido anti-horário.

**Figura 26 – Controle de motores e solenóide.**

```

void embosse() {
  for (int i = 0; i < 10; i++) {
    digitalWrite(solenoid, HIGH);
    delay(25);
    digitalWrite(solenoid, LOW);
    delay(25);
  }
  digitalWrite(solenoid, LOW);
  delay(1000);
}

void moveMotorX(int amount) {
  motorX.step(20 * amount);
  releaseMotorX();
  delay(1000);
}

void moveMotorY(int amount) {
  motorY.step(18 * amount);
  releaseMotorY();
  delay(1000);
}

```

**Fonte: Autoria Própria (2024).**

Após a impressão da mensagem, a função 'initialPosition()' retorna o cabeçote de impressão e o papel à posição inicial, preparando o sistema para uma nova mensagem. Tal função pode ser vista na Figura 27.

**Figura 27 – Posição inicial.**

```

void initialPosition(int charactersInLine, int lineCount) {
  int stepsToMoveY = 8;
  int stepsToMoveX = 20;

  lineCount = lineCount+1;

  int moveForwardY = (14 - lineCount)*stepsToMoveY;
  int moveBackX = charactersInLine*stepsToMoveX;

  moveMotorY(moveForwardY);
  moveMotorX(moveBackX);
}

```

**Fonte: Autoria Própria (2024).**

### 4.3 Estrutura mecânica

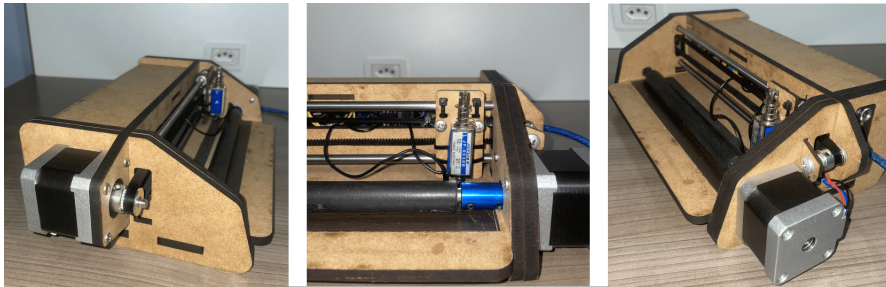
A estrutura mecânica deste projeto, inspirada no projeto Cheap Braille Embosser (La Picoreuse) da Autodesk Instructables (IAPAFOTO, 2018), desempenha um papel crucial na operação precisa e eficiente da impressora braille. A decisão de adotar o projeto de código aberto

se baseou na certeza de que suas dimensões e especificações se integrariam perfeitamente ao nosso contexto. Isso permitiu concentrar os esforços na parte do circuito eletrônico, na seleção dos componentes e no desenvolvimento de um novo código para atender às necessidades específicas da aplicação.

#### 4.3.1 Partes móveis

Um dos aspectos centrais da estrutura mecânica são as partes móveis, responsáveis por conduzir o movimento preciso necessário para a impressão braille. Este sistema inclui componentes como correias, polias e solenóides, cuidadosamente posicionados e coordenados para garantir a precisão e consistência do processo de impressão, que podem ser vistos na Figura 28. A adaptação do projeto original permitiu ajustes no *design* e na disposição dessas partes móveis para otimizar o desempenho e a confiabilidade da impressora.

**Figura 28 – Partes móveis.**

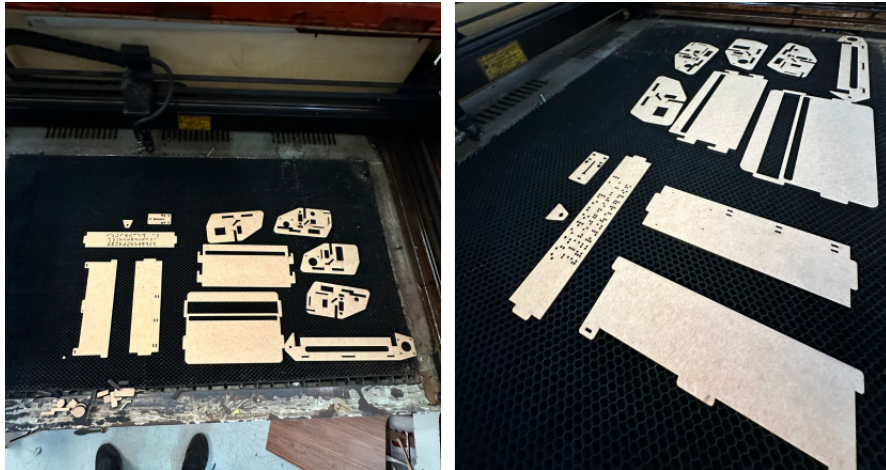


**Fonte: Autoria Própria (2024).**

#### 4.3.2 Estrutura externa

A estrutura externa desempenha um papel essencial ao fornecer suporte e proteção aos componentes internos, ao mesmo tempo em que facilita as operações de movimentação e impressão. Para isso, foram utilizadas as partes em MDF cortadas a laser, mostradas na Figura 29, conforme especificado pelo projeto de código aberto. A escolha desse material se deve à sua ampla disponibilidade, resistência e facilidade de personalização. O corte a laser garantiu não apenas o encaixe preciso das peças, mas também uma estrutura robusta e durável, fundamental para o bom funcionamento da impressora braille.

**Figura 29 – Corte da estrutura.**



**Fonte: Autoria Própria (2024).**

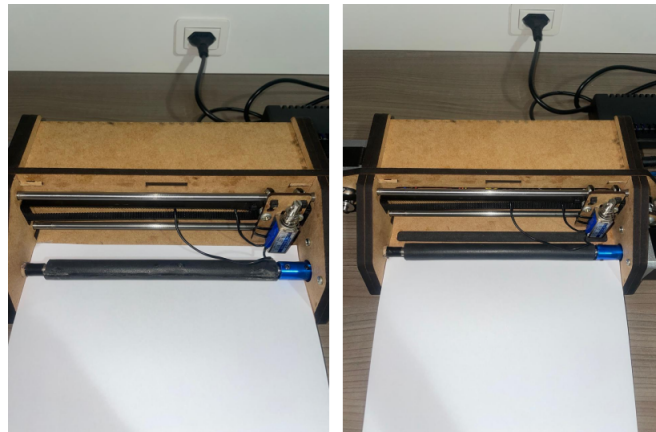
## 5 RESULTADOS E DISCUSSÕES

Neste capítulo serão abordados os resultados obtidos. As discussões têm o objetivo de certificar a funcionalidade, eficiência e desempenho do dispositivo, garantindo que ele atenda aos requisitos estabelecidos.

O funcionamento da impressora Braille de baixo custo foi cuidadosamente projetado para garantir simplicidade de uso e eficiência na produção de materiais em Braille. O usuário deve iniciar o processo ligando a impressora, conectando-a à tomada e ao computador via cabo USB, e certificando-se de que o cabeçote está na posição inicial e o código do Arduino está carregado na placa. Em seguida, o usuário insere uma folha A4 de 120g na beirada da impressora, pois esse tipo de papel é recomendado para impressão Braille devido à sua adequação para formar o relevo necessário.

Ao ver a mensagem "Digite seu texto" no monitor serial, o usuário deve digitar o caractere '[' para que a impressora puxe o papel, posicionando-o corretamente para a impressão. Esta etapa pode ser vista na Figura 30. Depois disso, o usuário pode digitar seu texto no monitor serial, respeitando o limite de 210 caracteres, que corresponde ao espaço disponível em uma folha A4, considerando quantos caracteres cabem por linha e quantas linhas preenchem o papel. Ao digitar uma frase, a impressora começará a processar o texto e iniciará a impressão.

**Figura 30 – Posicionamento da folha.**

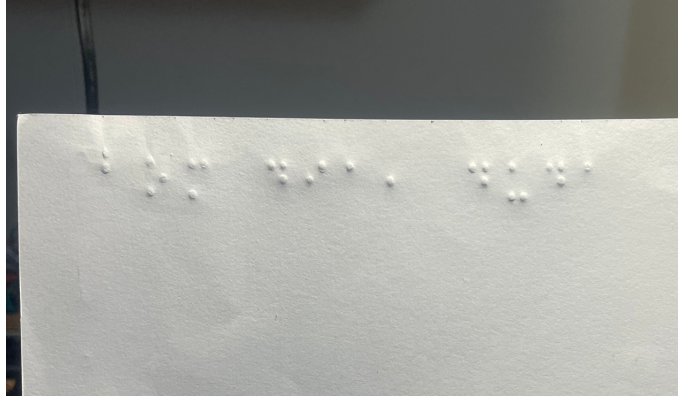


**Fonte: Autoria Própria (2024).**

Caso o texto exceda o limite de 210 caracteres, uma mensagem será exibida no monitor serial, informando o usuário sobre a necessidade de reduzir o texto. Se o texto estiver dentro do limite, a impressora iniciará automaticamente o processo de impressão. Durante a impressão, a impressora verifica se há caracteres não contemplados e os ignora, passando para o próximo caractere na frase. Este processo garante que apenas os caracteres válidos sejam impressos em Braille. Os testes realizados com a impressora demonstraram que o processo de impressão ocorreu conforme o esperado, com o relevo formado nas folhas de papel dentro dos padrões de

qualidade e legibilidade do Braille. A Figura 31 mostra um exemplo da impressão da frase "bom dia duda".

**Figura 31 – Frase impressa.**



**Fonte: Autoria Própria (2024).**

A utilização de papel A4 de 120g mostrou-se adequada para garantir a qualidade do relevo e minimizar possíveis problemas durante o processo de impressão. Os testes também revelaram que a impressora foi capaz de lidar eficientemente com textos de até 210 caracteres, proporcionando uma solução prática e eficaz para a produção de materiais em Braille. Assim que a impressora termina de imprimir, ela libera a folha e o cabeçote retorna à posição inicial, pronto para uma nova operação.

O funcionamento da impressora demonstra a simplicidade e eficácia do dispositivo, facilitando a produção de materiais em Braille de forma rápida e precisa, o que tem o potencial de promover a inclusão e a igualdade de oportunidades para pessoas com deficiência visual. Além disso, os resultados obtidos durante os testes confirmam a viabilidade e eficiência do projeto da impressora Braille de baixo custo. O uso de materiais acessíveis e a implementação de um processo de impressão simplificado contribuem para tornar a tecnologia Braille mais acessível e amplamente disponível.

Em alguns testes, foi observado que o papel acabou enroscando no próprio mecanismo de puxar a página, possivelmente devido à inserção falha. Este é um problema que pode ocorrer até mesmo em impressoras comuns, mas que merece atenção para evitar interrupções no processo de impressão.

Em resumo, o projeto da impressora Braille de baixo custo apresenta um potencial significativo para melhorar o acesso à informação escrita em Braille, tanto para indivíduos quanto para instituições. A continuação do desenvolvimento e aprimoramento deste projeto podem contribuir significativamente para promover a inclusão e a igualdade de oportunidades para pessoas com deficiência visual.

## 6 CONCLUSÕES

Com base na análise das informações apresentadas, é evidente que o projeto da impressora Braille de baixo custo desempenha um papel fundamental na democratização do acesso à informação escrita em Braille. A implementação de um dispositivo acessível para a produção de materiais em Braille não só atende às necessidades de indivíduos com deficiência visual, mas também beneficia instituições educacionais e sociais, promovendo a inclusão e a igualdade de oportunidades.

Os materiais utilizados no projeto, como MDF, haste de metal, polia, correia e uma variedade de componentes eletrônicos, foram selecionados cuidadosamente para garantir a funcionalidade e a durabilidade da impressora, ao mesmo tempo em que mantêm os custos baixos. A escolha desses materiais reflete o compromisso do projeto em fornecer uma solução econômica e eficiente para a produção de materiais em Braille.

Durante o desenvolvimento do *hardware* e do *firmware* da impressora, foram enfrentados desafios técnicos e de engenharia que exigiram soluções criativas e iterativas. A utilização de um Arduino Uno como plataforma de controle, juntamente com motores de passo, solenóide, as pontes H e outros componentes eletrônicos, permitiu a criação de um sistema funcional e acessível para a impressão em relevo.

A análise dos testes realizados com o protótipo da impressora demonstrou resultados promissores em relação aos objetivos estabelecidos. Apesar de pequenas discrepâncias e ajustes necessários durante o processo de desenvolvimento, o protótipo mostrou-se capaz de converter texto digital em uma representação tátil em Braille de forma eficiente e precisa.

Em síntese, o projeto da impressora Braille demonstrou ser uma solução promissora para atender às necessidades de pessoas com deficiência visual. Os objetivos propostos, que incluíam desenvolver um *hardware* capaz de imprimir em relevo, elaborar um *firmware* para comunicação com o computador, desenvolver um protótipo funcional e realizar testes para validação de *hardware* e *firmware*, foram cumpridos com sucesso. Os próximos passos incluem a otimização contínua do dispositivo e a sua disponibilização para um público mais amplo, com o objetivo de promover ainda mais a inclusão e a acessibilidade.

## 7 PERSPECTIVAS DE TRABALHOS FUTUROS

Olhando para o futuro do projeto da impressora Braille de baixo custo, várias perspectivas promissoras merecem consideração. Primeiramente, a integração de inteligência artificial poderia proporcionar benefícios significativos. A aplicação da IA poderia ser incorporada para aprimorar a precisão e eficiência do processo de impressão, além de desenvolver recursos avançados de automação. Exemplos incluem correção de erros de impressão, otimização de *layout*, aprendizado adaptativo, detecção de problemas de *hardware* e tradução automática de texto.

Além disso, a otimização na estrutura externa e *design* é essencial para tornar o dispositivo mais ergonômico, fácil de usar e esteticamente agradável. Isso poderia envolver a simplificação dos componentes, a redução do tamanho e peso, e a incorporação de materiais mais duráveis e modernos.

Garantir uma maior conformidade com as diretrizes já impostas, como padrões de acessibilidade e segurança, é fundamental para a aceitação e adoção do produto. Isso incluiria a realização de testes de usabilidade e acessibilidade, bem como a certificação de conformidade com regulamentações pertinentes. A importância de testes com usuários não pode ser subestimada, pois são essenciais para garantir que o dispositivo esteja correto e acessível a todos os seus potenciais utilizadores.

Adicionar caracteres que ainda não são contemplados, como símbolos especiais ou letras adicionais, ampliaria a utilidade e acessibilidade do dispositivo para usuários de diferentes idiomas e contextos culturais.

Desenvolver um aplicativo separado para controlar o dispositivo, ao invés de depender exclusivamente do comando direto do monitor serial, ofereceria uma experiência de usuário mais flexível e intuitiva. Isso poderia incluir recursos adicionais, como a capacidade de imprimir arquivos *.txt* e *Word* diretamente do aplicativo.

Desenvolver um aplicativo separado para controlar o dispositivo, ao invés de depender exclusivamente do comando direto do monitor serial, ofereceria uma experiência de usuário mais flexível e intuitiva. Nesse sentido, seria essencial ter integração *bluetooth* e permitir que o usuário possa digitar por comando de voz, além da possibilidade de imprimir arquivos *.txt* e *Word*.

Por fim, a inclusão de materiais sustentáveis na construção do dispositivo é uma consideração importante. Isso poderia envolver o uso de plásticos reciclados, materiais biodegradáveis ou outras opções ecologicamente conscientes para reduzir o impacto ambiental do produto.

Em resumo, o futuro do projeto da impressora Braille de baixo custo está repleto de oportunidades emocionantes para melhorias tecnológicas, de design e sustentabilidade, todas com o objetivo de tornar o dispositivo mais acessível, eficiente e ecologicamente consciente.

## REFERÊNCIAS

- Absolute Native Eletronics. **XL6009 Step-up Power Module - Manual**. [S./], 2014. Acesso em 12 de maio de 2024. Disponível em: <https://www.aneindia.com/wp-content/uploads/2014/10/XL6009-Manual.pdf>.
- Arduino. **Arduino Uno Rev3**. [S./], 2024. Acesso em 5 de abril de 2024. Disponível em: <https://docs.arduino.cc/hardware/uno-rev3/>.
- Bit Maker. **Mini Solenoide 12V Trava Elétrica JF-0530B**. [S./], 2024. Acesso em 21 de maio de 2024. Disponível em: <https://www.bitmaker.com.br/mini-solenoide-12v-trava-eletrica-jf-0530b>.
- BOCK, G. L. K.; SILVA, S. C. d. **Simbologia Braille: Caderno Pedagógico**. 1. ed. Florianópolis: DIOESC, Diretoria da Imprensa Oficial e Editora de Santa Catarina, 2013.
- DENARDIN, G. W. **Microcontroladores**. [S./], 2014. Acesso em 25 de março de 2024. Disponível em: <https://wiki.ifsc.edu.br/mediawiki/images/d/dd/Microcapostila.pdf>.
- Dias de Sá, E. **O sistema de escrita Braille**. [S./]: Editora Exemplo, 2001.
- EasyEDA. **EasyEDA - Plataforma de Desenvolvimento de Circuitos Online**. [S./], 2024. Acesso em 20 de março de 2024. Disponível em: <https://easyeda.com/pt>.
- FRANKS, R. Braille literacy and its importance in the digital age. **Disability Studies Quarterly**, v. 37, p. 98–109, 2017.
- Future Eletronic Corporation. **Datasheet do Módulo Relé de 1 Canal**. [S./], 2022. Acesso em 12 de abril de 2024. Disponível em: <https://curtocircuito.com.br/datasheet/modulo/rele-1canal.pdf>.
- GARCIA, J. C. D.; BRASIL, I. d. T. S. I. **Livro Branco da Tecnologia Assistiva no Brasil**. São Paulo: ITS BRASIL, 2017. 16, 54–55, 61–62 p.
- GIFE. **Os desafios do sistema Braille no Brasil**. 2023. Acesso em 20 de janeiro de 2024. Disponível em: <https://gife.org.br/as-novas-tecnologias-nao-substituem-o-braille-diz-regina-oliveira-da-fundacao-dorina-nowill/#:~:text=O%20sistema%20braille%20foi%20criado,uma%20educaç~ao%20inclusiva%20no%20país>.
- GONZALEZ, M. Challenges in the production of low-cost braille printers. **International Journal of Educational Technology**, v. 8, p. 77–88, 2021.
- Handson Technology. **L298N Dual H-Bridge Motor Driver Datasheet**. [S./], 2018. Acesso em 12 de maio de 2024. Disponível em: <https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf>.
- HARRIS, T. The role of technology in inclusive education for the visually impaired. **International Journal of Inclusive Education**, v. 24, p. 755–767, 2020.
- HILL, A. The cost and accessibility of braille technology. **Journal of Assistive Technologies**, v. 14, p. 115–125, 2020.
- IAPAFOTO. **Cheap Braille Embosser (La Picoreuse)**. [S./], 2018. Acesso em 5 de janeiro de 2024. Disponível em: <https://www.instructables.com/Cheap-Braille-Embosser-La-Picoreuse/>.

JOHNSON, L. Innovations in braille printing technology. **Journal of Visual Impairment Blindness**, v. 112, p. 573–584, 2018.

MEC. **Grafia Braille para a Língua Portuguesa**. [S.l.], 2018. Acesso em 2 de fevereiro de 2024. Disponível em: <http://portal.mec.gov.br/docman/dezembro-2018-pdf/104041-anexo-grafia-braille-para-lingua-portuguesa/file>.

Ministério da Educação. **Normas Técnicas para a Produção de Textos em Braille**. 3. ed. Brasília, DF, 2018. Elaboração: Dos Santos, Fernanda Christina; Oliveira, Regina Fátima Caldeira de. Disponível em: <http://portal.mec.gov.br/docman/dezembro-2018-pdf/105451-normas-tecnicas-para-a-producao-de-textos-em-braille-2018/file>.

OMS. **Síntese de evidências para políticas: Acesso à tecnologia assistiva**. [S.l.], 2020. Acesso em 12 de fevereiro de 2024. Disponível em: <https://iris.who.int/bitstream/handle/10665/332222/9789240011045-por.pdf>.

Pichai, S. **A tecnologia deve existir para todos. O verdadeiro avanço é quando ela se torna acessível e inclusiva, transformando vidas e quebrando barreiras**. 2020. Citação.

RODRIGUES, A. A. *et al.* Perspectiva econômica do mercado de inovação em tecnologia assistiva: cenário nacional e projeções mundiais. **Jornal Brasileiro de Economia da Saúde**, Abril 2024.

SEDH . **Tecnologia Assistiva**. Brasília: [s.n.], 2009. 11 p.

SMITH, J. Affordable braille printing solutions: A review. **Advances in Assistive Technology**, v. 11, p. 34–47, 2019.

Toda Matéria. **Células Braille correspondentes a cada letra do alfabeto**. [S.l.], 2024. Acesso em: 2 de fevereiro de 2024. Disponível em: <https://www.todamateria.com.br/braille/>.

UFF. **Motor de Passo**. [S.l.], 2008. Acesso em 16 de maio de 2024. Disponível em: <https://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>.

UNICAMP. **Engenharia Elétrica e Instrumentação de Atuadores**. [S.l.], 2007. Acesso em 16 de maio de 2024. Disponível em: [http://ftp.demec.ufpr.br/disciplinas/EngMec\\_NOTURNO/TM372/Conte%fadofados/3%20Atuadores/Eng%20Eletrico%20instrumenta%20o\\_Atadores\\_%20UNICAMP.pdf](http://ftp.demec.ufpr.br/disciplinas/EngMec_NOTURNO/TM372/Conte%fadofados/3%20Atuadores/Eng%20Eletrico%20instrumenta%20o_Atadores_%20UNICAMP.pdf).

Usinainfo. **Mini Solenóide 12V NF Tipo Tranca JF-0520B**. [S.l.], 2019. Acesso em 20 de maio de 2024. Disponível em: [https://www.usinainfo.com.br/mini-fechadura-eletrica-solenoid/mini-solenoid-12v-nf-tipo-tranca-jf-0520b-5246.html?gad\\_source=1&gclid=Cj0KCQjwvb-zBhCmARIsAAfUI2s9xtiwZrQdJ58W8aH3LmNdxMmu5X5nVqM65LTVuea4I6WkFut4N7YaAo](https://www.usinainfo.com.br/mini-fechadura-eletrica-solenoid/mini-solenoid-12v-nf-tipo-tranca-jf-0520b-5246.html?gad_source=1&gclid=Cj0KCQjwvb-zBhCmARIsAAfUI2s9xtiwZrQdJ58W8aH3LmNdxMmu5X5nVqM65LTVuea4I6WkFut4N7YaAo).

Usinainfo. **Isolador Óptico PC817 - 2 Canais 3.6-24V**. [S.l.], 2023. Acesso em 12 de maio de 2024. Disponível em: <https://www.usinainfo.com.br/outros-modulos-arduino/isolador-optico-pc817-2-canais-36-24v-8305.html>.

VURTZ Motors. **STEPPER MOTOR NEMA 17**. [S.l.], 2022. Acesso em 20 de maio de 2024. Disponível em: [https://curtocircuito.com.br/datasheet/VZS1740-048-1004\\_EN\\_V1.pdf](https://curtocircuito.com.br/datasheet/VZS1740-048-1004_EN_V1.pdf).

WILLIAMS, D. 3d printing for accessible education: The case of braille printers. **Journal of Educational Research and Technology**, v. 6, p. 290–302, 2019.

## **APÊNDICE A – Código Completo da Impressora Braille**

```
#include <Stepper.h>

// Número de passos por revolução para nema17
const int stepsPerRevolution = 200;

// Pinos de controle dos motores
const int motorYPin1 = 8;
const int motorYPin2 = 9;
const int motorYPin3 = 10;
const int motorYPin4 = 11;

const int motorXPin1 = 4;
const int motorXPin2 = 5;
const int motorXPin3 = 6;
const int motorXPin4 = 7;

// Pino da solenoide
const int solenoid = 2;

// Objetos Stepper para os dois motores
Stepper motorY(stepsPerRevolution, motorYPin1, motorYPin3, motorYPin2, motorYPin4);
Stepper motorX(stepsPerRevolution, motorXPin1, motorXPin3, motorXPin2, motorXPin4);

// Buffer para armazenar a mensagem recebida
const int bufferSize = 300;
char messageBuffer[bufferSize];
int messageIndex = 0;

void setup() {
    delay(1000);
    Serial.begin(9600);
    Serial.println("Digite uma frase: ");
    pinMode(solenoid, OUTPUT);

    // Definição da velocidade dos motores
    motorY.setSpeed(60);
    motorX.setSpeed(60);
}
```

```

}

void loop() {
  if (Serial.available() > 0) {
    // Lê o caractere do monitor serial
    char receivedChar = Serial.read();

    // Verifica se é o final da mensagem (Enter)
    if (receivedChar == '\n') {
      if (messageIndex > 211) {
        Serial.println("Número de caracteres acima do
        possível para impressão.");
      } else {
        messageBuffer[messageIndex] = '\0'; // Termina a string
        processMessage();
      }
      messageIndex = 0; // Reinicia o índice para a próxima mensagem
    } else {
      // Armazena o caractere no buffer se não for o final
      if (messageIndex < bufferSize - 1) {
        messageBuffer[messageIndex++] = receivedChar;
      }
    }
  }
}

```

```

// Função para processar toda a mensagem
void processMessage() {
  int printedChars = 0;
  int lineCount = 0;
  bool inNumberSequence = false;
  bool paperPulled = false;

  for (int i = 0; messageBuffer[i] != '\0'; i++) {
    char currentChar = messageBuffer[i];

    if (isdigit(currentChar)) {
      if (!inNumberSequence) {

```

```

        processChar('@');
        inNumberSequence = true;
        printedChars++;
    }
    processNumber(currentChar);
} else if (currentChar == '[' || currentChar == ']') {
    processChar(currentChar);
    paperPulled = true;
    // Não incrementa printedChars para '[' e ']'
} else {
    inNumberSequence = false;
    processChar(currentChar);
    printedChars++;
}
if (printedChars == 15) {
    moveMotorY(5); // Muda para a próxima linha
    delay(1000);
    moveMotorX(40); // volta para o início da linha
    printedChars = 0;
    lineCount++;
}
}
if (!paperPulled) {
    initialPosition(printedChars, lineCount);
} else {
    paperPulled = false;
}
}

// Função para processar o número
void processNumber(char c) {
    switch (c) {
        case '0':
            processChar('j');
            break;
        case '1':
            processChar('a');
            break;
    }
}

```

```
    case '2':
        processChar('b');
        break;
    case '3':
        processChar('c');
        break;
    case '4':
        processChar('d');
        break;
    case '5':
        processChar('e');
        break;
    case '6':
        processChar('f');
        break;
    case '7':
        processChar('g');
        break;
    case '8':
        processChar('h');
        break;
    case '9':
        processChar('i');
        break;
}
}

// Desligar motor X
void releaseMotorX() {
    digitalWrite(motorXPin1, LOW);
    digitalWrite(motorXPin2, LOW);
    digitalWrite(motorXPin3, LOW);
    digitalWrite(motorXPin4, LOW);
}

// Desligar motor Y
void releaseMotorY() {
    digitalWrite(motorYPin1, LOW);
```

```
digitalWrite(motorYPin2, LOW);
digitalWrite(motorYPin3, LOW);
digitalWrite(motorYPin4, LOW);
}

// Função para converter um caractere para minúscula
char toLowerCase(char c) {
    if (c >= 'A' && c <= 'Z') {
        return c + ('a' - 'A');
    }
    return c;
}

// Função que aciona as batidas da solenoide
void embosse() {
    for (int i = 0; i < 10; i++) {
        digitalWrite(solenoid, HIGH);
        delay(25);
        digitalWrite(solenoid, LOW);
        delay(25);
    }
    digitalWrite(solenoid, LOW);
    delay(1000);
}

// Função que move o motor X de forma padronizada
void moveMotorX(int amount) {
    motorX.step(20 * amount);
    releaseMotorX();
    delay(1000);
}

// Função que move o motor Y de forma padronizada
void moveMotorY(int amount) {
    motorY.step(18 * amount);
    releaseMotorY();
    delay(1000);
}
```

```
// Função que puxa o papel
void pullPaper() {
    moveMotorY(8);
    delay(1000);
}

// Função que libera o papel
void releasePaper() {
    moveMotorY(-8);
    delay(1000);
}

// Função para voltar na posição inicial
void initialPosition(int charactersInLine, int lineCount) {
    int stepsToMoveY = 8;
    int stepsToMoveX = 20;

    lineCount = lineCount+1;

    int moveForwardY = (14 - lineCount)*stepsToMoveY;
    int moveBackX = charactersInLine*stepsToMoveX;

    moveMotorY(moveForwardY);
    moveMotorX(moveBackX);
}

// Função que aciona comportamentos específicos com base no caractere
void processChar(char c) {
    c = toLowerCase(c);
    switch (c) {
        case '[':
            Serial.println("Pull paper:");
            pullPaper();
            break;
        case ']':
            Serial.println("Pull paper:");
            releasePaper();
            break;
        case 'a':
```

```
    Serial.println("Letra A: ");
    embosse();
    moveMotorX(-3);
    break;
case 'b':
    Serial.println("Letra B: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-3);
    break;
case 'c':
    Serial.println("Letra C: ");
    embosse();
    moveMotorX(-1);
    embosse();
    moveMotorX(-2);
    break;
case 'd':
    Serial.println("Letra D: ");
    embosse();
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'e':
    Serial.println("Letra E: ");
    embosse();
    moveMotorX(-1);
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'f':
```

```
Serial.println("Letra F: ");
embosse();
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-1);
embosse();
moveMotorX(-2);
break;
case 'g':
Serial.println("Letra G: ");
embosse();
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-1);
embosse();
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-2);
break;
case 'h':
Serial.println("Letra H: ");
embosse();
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-1);
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-2);
break;
case 'i':
Serial.println("Letra I: ");
moveMotorY(1);
embosse();
moveMotorY(-1);
```

```
        moveMotorX(-1);
        embosse();
        moveMotorX(-2);
        break;
case 'j':
    Serial.println("Letra J: ");
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'k':
    Serial.println("Letra K: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-3);
    break;
case 'l':
    Serial.println("Letra L: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-3);
    break;
case 'm':
    Serial.println("Letra M: ");
    embosse();
    moveMotorY(2);
    embosse();
```

```
    moveMotorY(-2);
    moveMotorX(-1);
    embosse();
    moveMotorX(-2);
    break;
case 'n':
    Serial.println("Letra N: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'o':
    Serial.println("Letra O: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'p':
    Serial.println("Letra P: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
```

```
    embosse();
    moveMotorX(-2);
    break;
case 'q':
    Serial.println("Letra Q: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'r':
    Serial.println("Letra R: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 's':
    Serial.println("Letra S: ");
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
```

```
    moveMotorX(-1);
    embosse();
    moveMotorX(-2);
    break;
case 't':
    Serial.println("Letra T: ");
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
    break;
case 'u':
    Serial.println("Letra U: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case 'v':
    Serial.println("Letra V: ");
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
```

```
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case 'w':
    Serial.println("Letra W: ");
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case 'x':
    Serial.println("Letra X: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case 'y':
    Serial.println("Letra Y: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
```

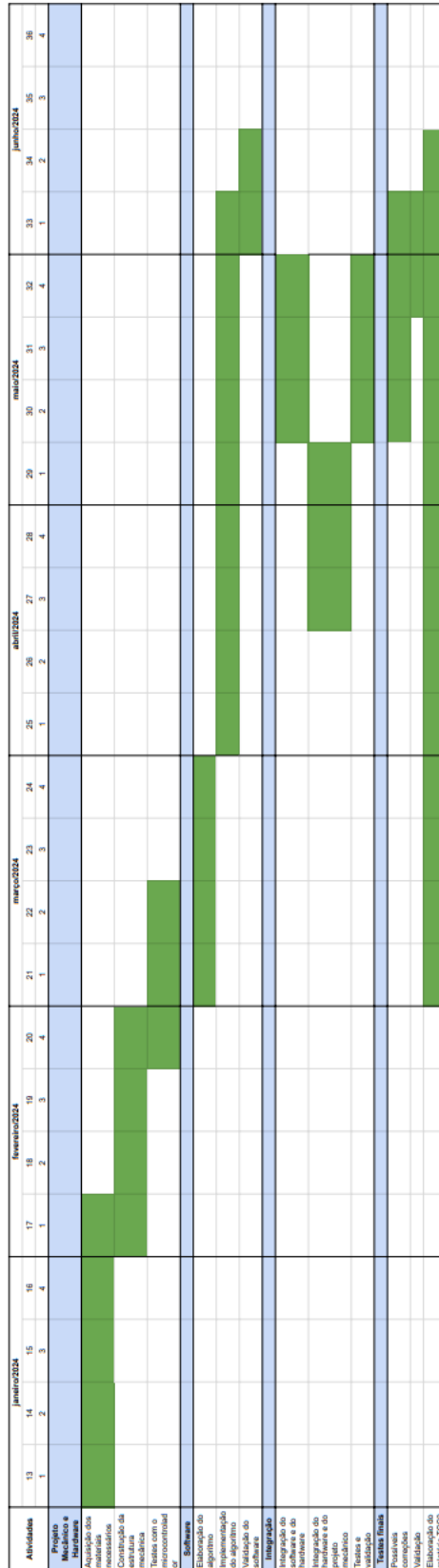
```
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case 'z':
    Serial.println("Letra Z: ");
    embosse();
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-2);
    break;
case ',':
    Serial.println("Caractere , ");
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-3);
    break;
case ';':
    Serial.println("Caractere ; ");
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-3);
    break;
case ':':
```

```
Serial.println("Caractere : ");
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-1);
moveMotorY(1);
embosse();
moveMotorY(-1);
moveMotorX(-2);
break;
case '.':
    Serial.println("Caractere . ");
    moveMotorY(2);
    embosse();
    moveMotorY(-2);
    moveMotorX(-3);
    break;
case '?':
    Serial.println("Caractere ? ");
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-1);
    embosse();
    moveMotorY(2);
    moveMotorX(-2);
    break;
case '!':
    Serial.println("Caractere ! ");
    moveMotorY(1);
    embosse();
    moveMotorY(1);
    embosse();
    moveMotorY(-2);
    moveMotorX(-1);
    moveMotorY(1);
    embosse();
    moveMotorY(-1);
    moveMotorX(-2);
```

```
        break;
    case '-':
        Serial.println("Caractere - ");
        moveMotorY(2);
        embosse();
        moveMotorY(-2);
        moveMotorX(-1);
        moveMotorY(2);
        embosse();
        moveMotorY(-2);
        moveMotorX(-2);
        break;
    case '@':
        Serial.println("Caractere que simboliza numero: ");
        moveMotorY(2);
        embosse();
        moveMotorY(-2);
        moveMotorX(-1);
        embosse();
        moveMotorY(1);
        embosse();
        moveMotorY(1);
        embosse();
        moveMotorY(-2);
        moveMotorX(-2);
        break;
    case ' ':
        Serial.println("Espaço: ");
        moveMotorX(-3);
        break;
    default:
        Serial.print("Caractere não tratado: ");
        Serial.println(c);
        break;
}
}
```

## **APÊNDICE B – Diagrama de Gantt**

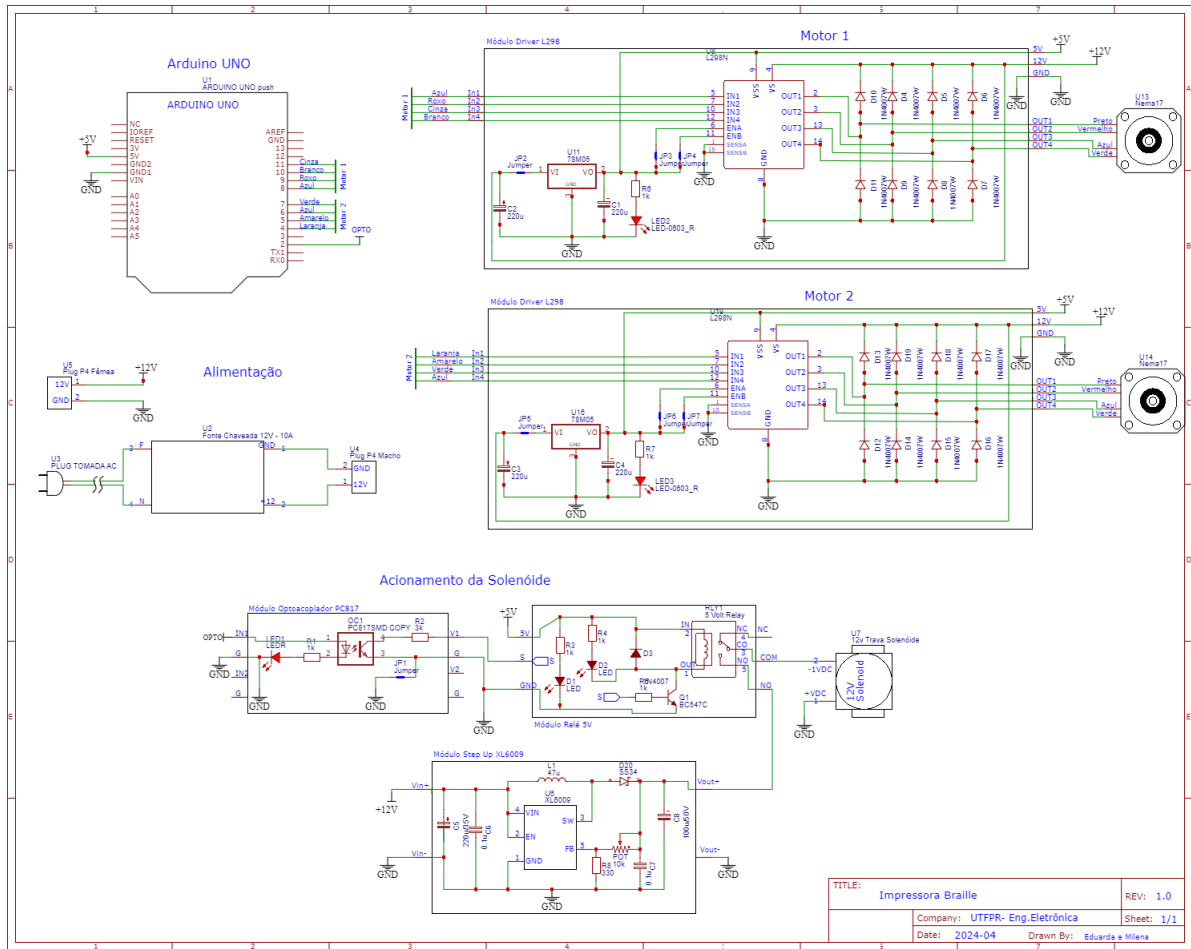
Figura 32 – Diagrama de Gantt.



Fonte: Autoria Própria (2024).

## **APÊNDICE C – Esquemático Elétrico**

Figura 33 – Esquemático Impressora Braille.



Fonte: Autoria Própria (2024).