



# UFPA

---

ESTEIRA DIDÁTICA DE TRANSPORTE E SELEÇÃO DE  
PEÇAS METÁLICAS E NÃO-METÁLICAS UTILIZANDO A  
PLATAFORMA ARDUINO®

2º Semestre de 2022

---

UNIVERSIDADE FEDERAL DO PARÁ - UFPA  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ - CAMTUC  
FACULDADE DE ENGENHARIA ELÉTRICA - FEE  
TUCURUÍ - PA

---

TUCURUÍ - PA  
2022

Arle Costa Vieira

ESTEIRA DIDÁTICA DE TRANSPORTE E SELEÇÃO DE PEÇAS  
METÁLICAS E NÃO-METÁLICAS UTILIZANDO A  
PLATAFORMA ARDUINO®

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Elétrica do Campus Universitário de Tucuruí da Universidade Federal do Pará como requisito final para a obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Ewerton R. Granhem

Tucuruí - PA  
2022

Arle Costa Vieira

ESTEIRA DIDÁTICA DE TRANSPORTE E SELEÇÃO DE PEÇAS  
METÁLICAS E NÃO-METÁLICAS UTILIZANDO A  
PLATAFORMA ARDUINO®

Trabalho de Conclusão de Curso apresentado  
à Faculdade de Engenharia Elétrica do Cam-  
pus Universitário de Tucuruí da Universidade  
Federal do Pará como requisito final para a  
obtenção do grau de Bacharel em Engenharia  
Elétrica.

**BANCA EXAMINADORA**

---

Prof. Dr. Ewerton R. Granhem  
Universidade Federal do Pará - UFPA  
Orientador

---

Prof. Dr. Andre Felipe Souza da Cruz  
Universidade Federal do Pará - UFPA  
Membro da Banca

---

Eng. Eletricista Anne Keise Monteiro Melo  
Membro da Banca

---

Eng. Eletricista Cleber Pereira Correa  
Membro da Banca

Tucuruí - PA  
2022

*“ Às memórias da minha avó MARIA LUIZA,  
cuja vida e lições quero carregar sempre na  
minha caminhada.*

# Agradecimentos

Em primeiro lugar agradeço a Deus que me deu coragem e força para conduzir este trabalho de conclusão de curso até o final.

Agradeço aos meus pais, JOSE DE ARIMATEIA VIEIRA SOBRINHO e MARIA DE JESUS COSTA VIEIRA por todo o amor e carinho que me ajudaram até aqui. A minha irmã ARIELE COSTA VIEIRA pelas palavras de incentivo e carinho.

Agradeço ao professor EWERTON RAMOS GRANHEM por toda a supervisão e atenção durante o desenvolvimento do presente trabalho.

A todas as pessoas que me ajudaram a concluir esse trabalho diretamente e indiretamente

*“Épigrafe.”*

# Resumo

VIEIRA, Arle Costa. **Esteira Didática de Transporte e Seleção de Peças Metálicas e Não-Metálicas Usando a Plataforma Arduino®**. 2022. Trabalho de Conclusão de Curso. Faculdade de Engenharia Elétrica, Campus Universitário de Tucuruí, Universidade Federal do Pará. Tucuruí, Pará, 2022.

Ao longo dos últimos anos, os sistemas de esteiras automatizadas se tornaram parte fundamental do processo de produção ou seleção em vários setores da atividade industrial, contribuindo para que os processos de fabricação e operacionalização de plantas industriais obtivessem benefícios referentes a velocidade e controle da produção. O objetivo central do trabalho é desenvolver uma esteira didática de seleção de peças metálicas e não-metálicas utilizando-se a plataforma microprocessadora Arduino. Propõe-se desenvolver uma abordagem inicial através do emprego do programa PROTEUS, para simular o funcionamento da plataforma Arduino, conectada a alguns sensores e atuadores, de forma a assegurar a segurança de operação e controle da esteira no estágio de montagem prática. Dessa maneira, foi possível apresentar os resultados obtidos na fase de funcionamento do protótipo - os quais são medições do sinal PWM durante a dinâmica de seleção das peças - para proceder com a comparação e validação dos parâmetros que o programa PROTEUS apresentou durante a simulação virtual.

**Palavras-chaves:** Esteira, Arduino®, Controle Digital, PWM, PROTEUS®, Automação.

# Abstract

Over the last few years, automated conveyor systems have become a fundamental part of the production or selection process in various sectors of industrial activity, contributing to the manufacturing and operationalization processes of industrial plants to obtain benefits in terms of speed and production control. The main objective of the work is to develop a didactic mat for the selection of metallic and non-metallic parts using the Arduino microprocessor platform. It is proposed to develop an initial approach through the use of the PROTEUS program, to simulate the operation of the Arduino platform, connected to some sensors and actuators, in order to ensure the safety of operation and control of the treadmill in the practical assembly stage. In this way, it will be possible to present the results obtained in the prototype working phase - which are measurements of the PWM signal during the dynamics of parts selection - to proceed with the comparison and validation of the parameters that the PROTEUS program presented during the virtual simulation.

**Keywords:** Belt. Arduino®. Digital Control. PWM. PROTEUS®. Automation.

# Lista de Figuras

<b>Figura 1</b>	– Painel de controle implementado com o CLP . . . . .	17
<b>Figura 2</b>	– Transportador de correia do tipo carvoeiro . . . . .	18
<b>Figura 3</b>	– Sistema transportador por correia com blocos inclinados . . . . .	21
<b>Figura 4</b>	– Transporte de caulim no sistema transportador por correia . . . . .	21
<b>Figura 5</b>	– Tela de comando da IDE do Arduino . . . . .	24
<b>Figura 6</b>	– Arduíno MEGA Rev3 . . . . .	26
<b>Figura 7</b>	– (a) Módulo de controle principal do sistema de gerenciamento de lâmpada ; (b) Conjunto de conexão da lâmpada com o relé . . . . .	28
<b>Figura 8</b>	– (a) Sistema de irrigação, mostrando a instalação convencional de tubulações; (b) Módulo de controle do sistema de irrigação implementado pela plataforma Arduino . . . . .	28
<b>Figura 9</b>	– (a) Atuador linear com interruptores de limite residenciais e negativos;(b) Placa Arduino DUE . . . . .	29
<b>Figura 10</b>	– Diagrama de blocos descrevendo a operação e dinâmica de funcionamento da esteira . . . . .	30
<b>Figura 11</b>	– Desenho em 3D da esteira com destaque para alguns componentes principais. . . . .	31
<b>Figura 12</b>	– Chaves lógicas de dois estados para a representação dos sensores . . . . .	32
<b>Figura 13</b>	– Modulação Por Largura de Pulso (PWM). . . . .	33
<b>Figura 14</b>	– Servomotor conectado ao pino 9 (I09) do Arduino para receber o sinal de controle PWM. . . . .	34
<b>Figura 15</b>	– Relação entre a largura de pulso e o ângulo do servomotor. . . . .	34
<b>Figura 16</b>	– (a) Valor de largura de pulso do sinal PWM correspondente a posição inicial do servo na simulação ; (b) Posição inicial do servo na simulação no programa PROTEUS . . . . .	35
<b>Figura 17</b>	– (a) Valor de largura de pulso do sinal PWM correspondente ao ângulo mínimo do servomotor ; (b) Ângulo mínimo do servomotor na simulação no programa PROTEUS . . . . .	35
<b>Figura 18</b>	– (a) Valor de largura de pulso do sinal PWM correspondente ao ângulo máximo do servomotor ; (b) Ângulo máximo do servomotor na simulação no programa PROTEUS . . . . .	36
<b>Figura 19</b>	– Esquema de ligações do circuito de controle . . . . .	37
<b>Figura 20</b>	– Motor Síncrono de acionamento da marca Galanz . . . . .	38
<b>Figura 21</b>	– Módulo relé de 8 canais . . . . .	39
<b>Figura 22</b>	– Sensor Indutivo . . . . .	39
<b>Figura 23</b>	– Sensor de Obstáculos Reflexivo . . . . .	40

<b>Figura 24</b> – (a) Servo Motor ; (b) Conexões de alimentação e controle do servo motor	41
<b>Figura 25</b> – Configurações de variáveis e bibliotecas	41
<b>Figura 26</b> – Inicialização dos pinos de entrada e saída	42
<b>Figura 27</b> – Declaração das estruturas condicionais	43
<b>Figura 28</b> – Construção Inicial da Esteira	44
<b>Figura 29</b> – Instalação da Tomada Para Alimentação da Placa Arduino e do Motor de Passo	45
<b>Figura 30</b> – (a) Conexão interna na fonte de alimentação; (b) Tomada e interruptor instalados.	46
<b>Figura 31</b> – Fixação do Servomotor para seleção do material.	47
<b>Figura 32</b> – (a) Sensor Indutivo; (b) Sensor de Obstáculo	48
<b>Figura 33</b> – Fixação de placas em MDF para direcionamento do material transportado.	49
<b>Figura 34</b> – Objetos usados no processo de teste de funcionamento da esteira de transporte e seleção.	49
<b>Figura 35</b> – Placa Arduino MEGA 2560.	50
<b>Figura 36</b> – Esteira transportadora em funcionamento. As setas em vermelho indicam o sentido de deslocamento do material, da esquerda para a direita.	51
<b>Figura 37</b> – (a) Esteira transportadora em funcionamento no laboratório de circuitos elétricos do NUT/UFPA; (b) Modelo de osciloscópio utilizado para obter o sinal PWM enviado do Arduino para o servomotor	52
<b>Figura 38</b> – (a) Sinal PWM visualizado em laboratório na etapa em que a esteira opera sem detecção de materiais; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS	53
<b>Figura 39</b> – (a) Sinal PWM visualizado em laboratório na etapa em que ocorre a detecção de metal; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS na etapa de detecção de metal	54
<b>Figura 40</b> – (a) Sinal PWM visualizado em laboratório na etapa em que ocorre a detecção de madeira; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS na etapa de detecção de madeira	54
<b>Figura 41</b> – Trem de pulsos PWM na simulação com o Proteus.	55
<b>Figura 42</b> – Trem de pulso PWM representado em laboratório por meio do osciloscópio.	55

# Lista de Tabelas

<b>Tabela 1</b>	– Modelos de plataforma Arduino e suas características . . . . .	23
<b>Tabela 2</b>	– Descrição de alguns menus da IDE do Arduino . . . . .	24
<b>Tabela 3</b>	– Algumas bibliotecas padrão do Arduino . . . . .	25
<b>Tabela 4</b>	– Características Arduino MEGA 2560 . . . . .	26
<b>Tabela 5</b>	– Orçamento Financeiro de Custos do Projeto . . . . .	56

# Lista de abreviaturas e siglas

CLP	Controlador Lógico Programável
PWM	Pulse Width Modulation (Modulação Por Largura de Pulso)
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
IHM	Interface Homem Máquina
DNA	Deoxyribonucleic Acid (Ácido Desoxirribonucleico)

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Proposta de Trabalho, Objetivos e Estrutura Textual	15
1.2	Histórico sobre os Sistemas de Automação com Esteiras	15
<b>2</b>	<b>SISTEMAS DE AUTOMAÇÃO UTILIZANDO A PLATAFORMA ARDUINO</b>	<b>20</b>
2.1	Princípio de Funcionamento dos Sistemas Transportadores	20
2.1.1	Sistemas Transportadores por Correia	20
2.2	Principais Aplicações dos Sistemas Transportadores	21
2.3	Plataforma Arduino	21
2.4	Definição da Plataforma Arduino	22
2.5	Características de Programação e Implementação	23
2.5.1	Ambiente de Desenvolvimento Integrado (IDE)	23
2.5.2	Bibliotecas Arduino e Pacotes de Suporte	25
2.6	A Plataforma Arduino Mega 2560	26
2.7	Histórico de Aplicação de Sistemas de Automação Com o Arduino	27
<b>3</b>	<b>PROJETO DE AUTOMAÇÃO DA ESTEIRA TRANSPORTADORA</b>	<b>30</b>
3.1	Desenho da Esteira Didática de Transporte e Seleção	30
3.2	Simulação do Circuito Eletrônico no Software Isis Proteus	32
3.2.1	Sinal PWM	33
3.2.2	Controle de Posição do Servomotor	33
3.3	Descrição dos Componentes do Sistema de Automação	38
3.3.1	Motor de Acionamento da Esteira	38
3.3.2	Módulo Relé de 8 Canais	38
3.3.3	Sensor Indutivo	39
3.3.4	Sensor de Obstáculos	40
3.3.5	Servomotor	40
3.4	Estrutura de Programação	41
<b>4</b>	<b>MONTAGEM E REALIZAÇÃO DE TESTES PARA AJUSTES NO SISTEMA DE AUTOMAÇÃO</b>	<b>44</b>
4.1	Montagem da Esteira de Transporte	44
4.2	Inserção da Fonte de Alimentação	45
4.3	Fixação do Servomotor	47
4.4	Instalação dos Sensores Indutivo e Reflexivo	47

4.5	Inserção de Placas Para Posicionamento de Material na Correia . .	48
4.6	Fixação da Placa Arduino MEGA 2560 . . . . .	49
4.7	Fase Final de Montagem e Teste de Funcionamento . . . . .	51
5	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>52</b>
5.1	Resultados de Funcionamento do Sistema de Transporte . . . . .	52
5.2	Observações Adicionais . . . . .	56
5.3	Orçamento Financeiro do Protótipo . . . . .	56
5.4	Propostas de Melhoria e Trabalhos Futuros . . . . .	57
5.4.1	Controle de Velocidade do Motor de Acionamento . . . . .	57
5.4.2	Inserção de um Braço Robótico Para Seleção de Itens . . . . .	57
5.4.3	Instalação de Outros Sensores . . . . .	57
	<b>REFERÊNCIAS BIBLIOGRÁFICAS . . . . .</b>	<b>58</b>

# 1 Introdução

## 1.1 Proposta de Trabalho, Objetivos e Estrutura Textual

A proposta do presente trabalho é montar uma esteira que realize a seleção de objetos metálicos e não-metálicos, fazendo o uso da plataforma Arduino. Foi definida uma etapa de planejamento que envolvia o desenho da esteira, escolha dos componentes eletrônicos e simulações do circuito de acionamento e seleção da esteira no programa Proteus. Como resultado tivemos o funcionamento do protótipo em laboratório e a análise da dinâmica de seleção de peças realizada pelo servomotor.

Como objetivos específicos deste trabalho foram definidos o projeto de automação da esteira, a simulação do circuito eletrônico no programa PROTEUS, a montagem da bancada de fixação da esteira, controle da posição do servomotor responsável pela seleção dos materiais e a economicidade e baixo custo do projeto referente a compra dos equipamentos e componentes utilizados.

As principais referências utilizadas para o embasamento do presente trabalho foram (LAMB, 2015), (PURDUM, 2015), (MARGOLIS, 2012), (PETRUZELLA, 2010), (AGRAWAL NIKHIL; SINGHAL, 2015), (EVANS JOSHUA NOBLE, 2013) e (SPINOLA, 2011).

A estrutura textual do trabalho consiste no capítulo 1 como introdução, capítulo 2 com uma revisão bibliográfica dos sistemas de transporte, capítulo 3 com a descrição da fase de planejamento e projeto da esteira, capítulo 4 envolvendo a montagem e realização de testes na esteira e o capítulo 5 descrevendo os resultados, discussões e um orçamento descrevendo quanto em dinheiro foi investido para o desenvolvimento do protótipo.

## 1.2 Histórico sobre os Sistemas de Automação com Esteiras

O estágio primário da automação destacou-se no início na segunda metade do século 18 com a inserção da mecanização no processos produtivos e industriais, ou seja, através da transição do esforço humano ou animal para as atividades executadas por máquinas e equipamentos. Como exemplo, destaca-se a invenção da máquina a vapor por James Watt em 1769, que produzia força rotativa contínua através da atuação de um regulador centrífugo de velocidade (SPINOLA, 2011).

No ano de 1868, James Clark Maxwell se dedica aos estudos e modelos matemáticos para a análise de realimentação de sistemas, direcionando a linha de pesquisa para o controle de rotação de dispositivos mecânicos (SPINOLA, 2011). Como consequência desses

trabalhos, foi verificado um grande avanço na implementação de sistemas de controle automático para a estabilização de navios e aeronaves no início do século 20. Dessa forma, entende-se que os primeiros sistemas de automação eram mecânicos em razão dos avanços tecnológicos alcançados a partir de 1760. Com o advento da eletrônica analógica em 1950, os sistemas de automação adquiriram um aspecto mais flexível e simplificado (SPINOLA, 2011).

Já nas décadas de 1980 e 1990 surgem os primeiros controladores lógicos programáveis com o objetivo de substituir o emprego excessivo de relés nas montadoras da General Motors (LAMB, 2015). Com o avanço tecnológico a partir dos anos 2000, mais empresas optaram pelo setor de automação, criando condições para o desenvolvimento de dispositivos de controle mais avançados. Como exemplo, atualmente existe uma ampla faixa de controladores lógicos desenvolvidos por diversas empresas fabricantes (LAMB, 2015).

De acordo com (LAMB, 2015) o conceito de automação é definido pela utilização de controles e dispositivos que substituem as ações humanas baseadas em escolha de decisões e ordens de resposta. Dessa forma a automação transcende a mecanização simples das atividades, pois diminui a demanda de atuações sensoriais e cerebrais por parte do ser humano. O termo automação foi mencionado pela primeira vez em 1940 por um engenheiro da Ford que fez uma descrição de um conjunto de sistemas de controles que cumpriam atividades antes executadas pela força humana (LAMB, 2015). O processo de modernização e adequação dos sistemas de automação nos dias atuais oferecem suporte para diversos segmentos industriais como por exemplo, empresas alimentícias, metalúrgicas, companhias de extração mineral entre outras.

Neste contexto, são bastante comuns encontrar, na quase totalidade dos processos produtivos, os controladores lógicos programáveis (CLP's), braços mecânicos acionados hidraulicamente e as esteiras transportadoras, cuja ênfase maior é abordada nesse trabalho. A automatização de sistemas transportadores é um processo que tem se tornado mais presente e indispensável em plantas industriais, gerando conseqüentemente, questões relativas ao seu desempenho e possibilidade da ocorrência de falhas ou problemas durante o seu funcionamento.

Como exemplo, podemos citar o defeito frequente decorrente da existência de folgas mecânicas nos módulos de esteiras que efetuam paradas periódicas. Essa falha que ocorre com maior frequência em esteiras de paradas sistemáticas, impõe a necessidade de precisão nesses sistemas transportadores, com o objetivo de evitar movimentações não desejáveis nas posições de parada da correia transportadora (RANIEL, 2011). Outra razão que demanda o largo emprego desse tipo de sistema de transporte são as questões relacionadas a proteção e conservação do meio ambiente (SANTOS, 2014).

Na figura 1 é apresentado um exemplo de aplicação de um sistema de controle

de automação industrial, cuja lógica de acionamento e programação é efetuada pelo Controlador Lógico Programável (CLP). Essa classe de controlador é especificamente usada em aplicações que envolvem a operação de motores de grande porte para os sistemas motrizes de movimentação da correia de carga e o transporte de material pesado, típico em plantas de característica industrial.

**Figura 1** – Painel de controle implementado com o CLP



Fonte: (PETRUZELLA, 2010)

Os benefícios proporcionados aos operadores do CLP se destacam através do fácil diagnóstico de falhas ainda na fase de projeto, redução do espaço físico exigido para instalação dos painéis, capacidade de reprogramação sem a necessidade de interrupção do processo produtivo, ter a capacidade de se comunicar com outros equipamentos, etc. Atualmente com a crescente redução no custo do CLP, observa-se o incremento de sua utilização nas indústrias em geral, desconsiderando-se aspectos relacionados ao seu porte ou ramo de atividades (SILVEIRA, 1968).

A utilização de sistemas transportadores como separadores automáticos ou esteiras tem o objetivo de garantir a efetivação dos processos de reciclagem efetuados nas empresas, como meio de reutilizar os resíduos gerados pela produção. Nas unidades de reciclagem, as esteiras transportadoras cumprem um papel essencial, movimentando os resíduos a serem reciclados para o estágio de separação manual (SANTOS, 2014).

Na análise histórica do desenvolvimento de esteiras transportadoras, as primeiras correias eram fabricadas em camadas de lonas ou tecidos de algodão, com revestimento de borracha natural. As primeiras documentações de utilização de sistemas transportadores por correia, indicam que os celeiros construídos pelas companhias de atividade portuária Mersey Docks e Harbour Board, em Liverpool, Reino Unido, em 1868 utilizavam as esteiras como meio de transporte em larga escala (STACE, 2008). As correias usadas nessas instalações consistiam em duas lonas revestidas de borracha natural, Essas correias tinham

pouca durabilidade, não por causa de seu processo de confecção, mas por causa da sua utilização com rolos que tinham as extremidades queimadas para formar uma calha. Em razão da alta ocorrência de desgaste das esteiras, foram inseridos rolos planos para garantir maior vida útil á correia, gerando conseqüentemente, diminuição da sua capacidade de transporte (STACE, 2008).

Registros do primeiro transportador moderno datam de 1891, na reserva de minério de ferro de Ogden, em Nova Jersey. Os relatos da época mencionam o nome de Thomas Edison como responsável pelo desenvolvimento dos rolos tensores desse sistema transportador (STACE, 2008). Depois da inserção de transportadores nas minas de carvão em 1905 por Richard Sutcliff e da alta disponibilidade de borracha no mercado depois da eclosão da Primeira Guerra Mundial (1914-1918), as esteiras com correias revestidas de borracha foram padronizadas como componente indispensável da maioria dos sistemas transportadores (STACE, 2008). Na figura 2 é mostrado um transportador de correia para escoamento de carvão.

**Figura 2** – Transportador de correia do tipo carvoeiro



Fonte: (STACE, 2008)

O emprego de lonas de algodão causou redução tanto da capacidade de carga como do comprimento dos transportadores que poderiam ser instalados, motivado pela limitação na resistência das correias que poderiam ser fabricadas (STACE, 2008). Em longo prazo, a opção mais viável a se escolher foi de introduzir vários transportadores em cascata, onde cada sistema teria seu acionamento individual, maquinário específico e estruturas de transferência e carregamento. O aperfeiçoamento da resistência e desempenho das lonas de borracha contribuiu para o planejamento de sistemas transportadores com linhas de fornecimento mais longas e mecanismos de acionamento mais robustos, eliminando a necessidade de múltiplos módulos de alimentação, e ocasionando redução de custos relacionados ao capital investido para o projeto das instalações (STACE, 2008).

O ano de 1944 marca a construção e operação do primeiro transportador por correias revestidas de borracha e com linha de escoamento com mais de 1.6 km de comprimento.

---

Em 1960 inicia-se a confecção de correias munidas com fibra sintética. Essa melhoria inserida no processo de fabricação resultou em outro avanço considerável na questão de resistência e comprimento das correias transportadoras (STACE, 2008).

## 2 Sistemas de Automação Utilizando a Plataforma Arduino

Nesse presente capítulo, realiza-se uma revisão bibliográfica sobre os sistemas transportadores e as características da plataforma microprocessada Arduino. Destaca-se a simplicidade de operação da plataforma de programação open source e a diversidade de aplicações a serem desenvolvidas com o uso do microcontrolador ATMEGA 2560, incluindo a definição de bibliotecas e especificações técnicas. Também será explanada uma análise histórica de sistemas de automação e protótipos construídos com base em plataformas microprocessadas com o Arduino.

### 2.1 Princípio de Funcionamento dos Sistemas Transportadores

Um sistema transportador é um equipamento que realiza o transporte de objetos e substâncias. Sua configuração de layout é diversificada e geralmente o meio de acionamento se dá por meio de um motor conectado ao sistema, pelo ar ou pela força gravitacional (LAMB, 2015). Os principais tipos de sistemas transportadores consistem no sistema por correia, por correntes e por paletes(LAMB, 2015). Como o presente trabalho fez o uso do primeiro sistema por correia, é mostrada uma descrição desse sistema transportador na próxima subseção.

#### 2.1.1 Sistemas Transportadores por Correia

Os transportadores de correia plana (com lona de deslocamento) consistem na inserção de duas ou mais polias, através das quais o sistema transportador gira uma correia. Uma ou mais polias podem ser acionadas, transmitindo movimentação para a correia e para o material que está sendo movido pelo sistema transportador. Os rolos alimentados com acionamento motriz são denominados condutores, enquanto que as polias que não recebem acionamento são chamada de rolos intermediários. As polias intermediárias podem ser encontradas na parte inferior do sistema transportador, com o objetivo de garantir sustentação para a fita de retorno da correia. A configuração de posicionamento dos motores para esse tipo transportador é localizada na polia da extremidade do sistema. Nos casos de acionamento de correias em sistemas transportadores reversos, o motor é instalado no meio da esteira (LAMB, 2015). Na figura 3 é mostrado um exemplo de aplicação do sistema transportador com correia montada em blocos salientes.

**Figura 3** – Sistema transportador por correia com blocos inclinados

Fonte: (LAMB, 2015)

## 2.2 Principais Aplicações dos Sistemas Transportadores

Os sistemas transportadores, com destaque para os modelos com correia, encontram uma larga faixa de aplicações como o transporte de materiais a granel, alimentadoras de peças ou processamento térmico. Na figura 4 quatro é mostrado um transportador com correia para a movimentação de caulim.

**Figura 4** – Transporte de caulim no sistema transportador por correia

Fonte (CONFERENCE, 1997)

## 2.3 Plataforma Arduino

Com o objetivo de fomentar o desenvolvimento de projetos de artes associados a tecnologia interativa no Instituto de Design Interativo localizado na cidade de Ivrea, Itália, o professor Massimo Banzi inaugurou a primeira plataforma Arduino de prototipagem eletrônica para seus alunos de design no ano de 2005. Este professor queria ensinar eletrônica e programação de computadores para seus alunos, porém encontrava muita

dificuldade, uma vez que não eram alunos da área de programação ou eletrônica. Outra razão que restringia os planos do professor Massimo Banzi era ocasionado pela inexistência de placas robustas e de baixo custo no mercado (EVANS JOSHUA NOBLE, 2013).

Pensando nestes problemas, Massimo Banzi e o cientista da Universidade de Malmo na Suécia, David Cuartielles, decidiram projetar sua própria placa, juntamente com a ajuda de um aluno de Massimo, David Mellis, que ficou responsável por desenvolver a linguagem de programação da plataforma. Gianluca Martino foi o responsável por criar o protótipo comercial para fabricação em grande escala, traçando metas para a confecção de 200 placas (EVANS JOSHUA NOBLE, 2013).

A plataforma recém-lançada recebeu a denominação Arduino em razão de um bar que era visitado por professores e estudantes do Instituto. As plataformas eram fornecidas com kits com o objetivo de que os alunos tivessem criatividade para desenvolver diferentes aplicações (EVANS JOSHUA NOBLE, 2013). Devido a sua robustez e facilidade de utilização, a publicidade da plataforma Arduino foi crescendo entre alunos de eletrônica e automação. Com o aperfeiçoamento do sistema original, várias versões foram lançadas no mercado, contribuindo para que a quantidade de placas oficiais vendidas chegasse a 300 mil unidades, fornecidas por uma linha diversificada de distribuidores em todo o mundo (EVANS JOSHUA NOBLE, 2013).

## 2.4 Definição da Plataforma Arduino

Arduino é uma plataforma eletrônica de processamento de dados, com código-fonte aberto, constituída por uma única placa que possui suporte para entrada e saída de sinais analógicos ou digitais (I/O). Em razão do código-fonte estar aberto para o desenvolvimento colaborativo, o Arduino pode ser utilizado tanto para aplicações independentes de cunho interativo como para desenvolvimento de projetos que operem em conjunto com outros softwares em seu computador (BANZI MASSIMO; SHILOH, 2015).

A versão inicial da placa Arduino foi concebida com base no microcontrolador ATMEGA8 de 8 *bits* da Atmel AVR. O chip ATMEGA8 possui 28 pinos dos quais 20 pinos podem ser configurados como entradas ou saídas (WARREN JOSH ADAMS, 2011). Dos 20 pinos disponíveis para o controle da placa, 6 pinos podem ser utilizados como entradas analógicas e outros 6 pinos possuem a funcionalidade de serem configurados como saídas PWM (WARREN JOSH ADAMS, 2011).

A placa original Arduino opera com uma frequência de clock de 16MHz e possui três dispositivos de ajuste de temporização para o controle da frequência do sinal PWM (WARREN JOSH ADAMS, 2011). O programa utilizado para o controle da placa Arduino é denominado Ambiente de Desenvolvimento Integrado (IDE), cuja estrutura fundamental foi construída com base na linguagem Processing (BANZI MASSIMO; SHILOH, 2015).

Como a plataforma utiliza microcontroladores da Atmel AVR e eles possuem várias versões, também há várias versões de placas Arduino no mercado. A Tabela 1 mostra algumas versões e suas características.

**Tabela 1** – Modelos de plataforma Arduino e suas características

	<b>Arduino Mega 2560</b>	<b>Arduino Leonardo</b>	<b>Arduino Ethernet</b>
<b>Microcontrolador</b>	ATmega2560	ATmega168	ATmega328
<b>Tensão Operacional</b>	5V	5V	5V
<b>Tensão de entrada</b>	7-12V	7-12V	7-12V
<b>Digital I/O pins</b>	54 (dos quais 15 fornecem saída PWM)	14 (dos quais 7 fornecem saída PWM)	14 (dos quais 4 fornecem saída PWM)
<b>Entradas Analógicas</b>	16	12	6
<b>Memória Flash</b>	256KB (8KB de memória RAM)	16KB (1KB de memória RAM)	32KB (2KB de memória RAM)
<b>Veloc. do Relógio</b>	16 MHz	16 MHz	16 MHz

Fonte: (PURDUM, 2015)

## 2.5 Características de Programação e Implementação

Nesta seção será abordado as características de programação do Arduino, detalhando-se a interface de desenvolvimentos de códigos, disponibilidade de bibliotecas e pacotes de suporte integrado que fornecem funcionalidades adicionais ao dispositivo microprocessador. Essas ferramentas de sub-rotinas predefinidas visam oferecer mais facilidade e otimização ao programa que será desenvolvido.

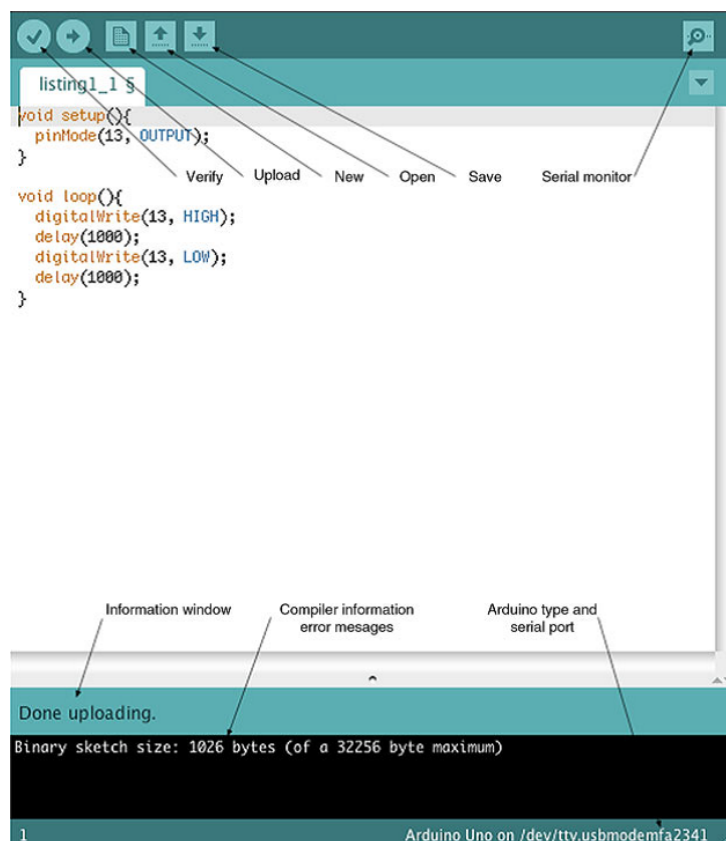
### 2.5.1 Ambiente de Desenvolvimento Integrado (IDE)

O Ambiente de Desenvolvimento Integrado consiste em um software gratuito disponibilizado pela própria desenvolvedora, onde será escrito a sequência de instruções que serão interpretadas pelo Arduino. Ele se conecta ao hardware e realiza a comunicação através de um cabo USB para carregar o código desenvolvido para a placa. O software é especificado por um compilador próprio de programação além do gerenciador de aplicação que já é carregado na placa (MCROBERTS, 2013). Essa característica confere a possibilidade dos seus projetos serem compilados *stand-alone* com o código já carregado no chip.

Os programas de instruções escritos neste ambiente de desenvolvimento são chamados de Sketches, onde os arquivos são salvos com a extensão *.ino*. Quando se inicia a IDE,

encontra-se a área para escrever o programa a ser compilado, a barra de ferramentas e o console de textos, que exibe tanto uma lista completa de erros no código como o resultado das instruções enviadas ao Arduino (MCROBERTS, 2013). Alguns botões de operações básicas são indicados na figura 5. A descrição de alguns comandos da barra de ferramentas da IDE é mostrada na tabela 2.

Figura 5 – Tela de comando da IDE do Arduino



Fonte: (EVANS JOSHUA NOBLE, 2013)

Tabela 2 – Descrição de alguns menus da IDE do Arduino

Menu	Descrição
<i>Verify</i>	Verifica todo o código e mostra onde/se há erros
<i>Upload</i>	Compila o código para a placa Arduino®
<i>New</i>	Cria uma nova janela para um novo código
<i>Open</i>	Abre um menu onde se tem a opção de abrir um arquivo, abrir arquivos recentes e carregar exemplos de códigos pré-definidos
<i>Save</i>	Salva o programa atual
<i>Monitor Serial</i>	Exibe os dados seriais enviados do Arduino

Fonte: (MCROBERTS, 2013)

## 2.5.2 Bibliotecas Arduino e Pacotes de Suporte

As bibliotecas são conjuntos de códigos que inserem funcionalidades extras ao ambiente de programação do Arduino. Essas sub-rotinas estruturadas possuem a finalidade de facilitar a comunicação com os dispositivos conectados na placa. Existem diversas bibliotecas disponíveis, algumas são internas e outras são disponibilizadas para *download* e podem ser instaladas muito facilmente. As bibliotecas padrão são mostradas na Tabela 3.

**Tabela 3** – Algumas bibliotecas padrão do Arduino

Biblioteca	Descrição
<b>EEPROM</b>	Leitura e escrita para armazenamento permanente
<b><i>Ethernet</i></b>	Para conexão com a internet usando o Arduino <i>Ethernet Shield</i>
<b>Firmata</b>	Para se comunicar com os aplicativos no computador utilizando um protocolo serial padrão
<b>GSM</b>	Para conectar a uma rede GRPS/GSM com o GSM Shield
<b><i>LiquidCrystal</i></b>	Para controlar telas de cristal líquido (LCDs)
<b>SD</b>	Para ler e escrever cartões SD
<b>Servo</b>	Para controlar servomotores
<b>SPI</b>	Para se comunicar com dispositivos que utilizam a Interface Periférica Serial (SPI)
<b><i>SoftwareSerial</i></b>	Para comunicação serial em qualquer pino digital
<b><i>Stepper</i></b>	Para controlar motores de passo

Fonte: (MARGOLIS, 2012)

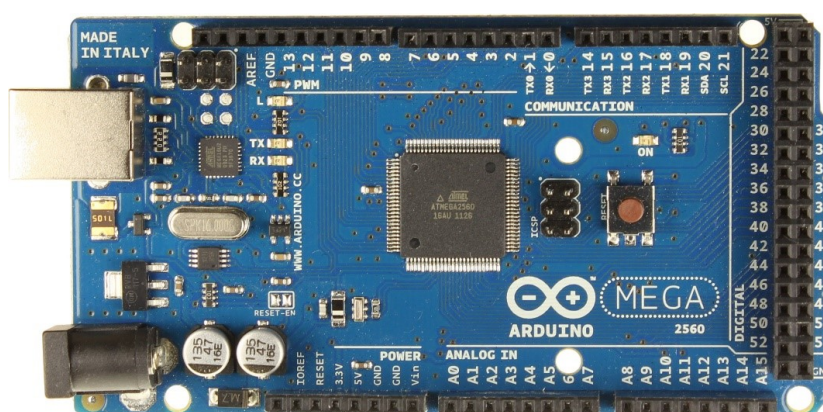
Algumas bibliotecas já vêm pré-instaladas juntamente com a IDE do Arduino, mas como se trata de uma plataforma de código aberto, o usuário tem a liberdade de criar bibliotecas personalizadas (MARGOLIS, 2012). Á título de exemplo, a biblioteca *SoftwareSerial* permite realizar a comunicação serial via protocolo RS232 com outro dispositivo utilizando quaisquer dos pinos digitais da plataforma Mega2560 (MARGOLIS, 2012).

As bibliotecas fornecem versatilidade na implementação de códigos extensos com o objetivo de facilitar sua correta utilização. Como exemplo pode-se destacar a biblioteca *Wire* fornecida em conjunto com o software do Arduino. Sua estrutura omite uma grande parcela da dificuldade de comunicações entre dispositivos (MARGOLIS, 2012). Através de um estudo detalhado da documentação da biblioteca, é possível efetuar sua devida adequação ao programa escrito no editor do IDE por meio da inclusão do código ao script principal. Essa adição de mais linhas de código causa conseqüentemente, um aumento do tamanho do programa no estágio de verificação do sketch, entretanto, o programa de compilação do Arduino utiliza somente o código de biblioteca que realmente está em uso, eliminando preocupações com relação ao excesso de memória carregada durante a compilação (MARGOLIS, 2012).

## 2.6 A Plataforma Arduino Mega 2560

Para este trabalho proposto, foi utilizada a placa Arduino MEGA 2560 (Figura 6). A plataforma microprocessada Arduino MEGA 2560 (Figura 6) é baseada no microcontrolador ATmega2560. Conforme a descrição de especificações de pinagem representada na Tabela 5 a placa contém 54 canais digitais de *input/output* sendo que, 15 pinos podem ser usados como saída PWM, 16 pinos podem ser utilizados como entradas analógicas. Além dessa grande quantidade de pinos para processamento de sinais de controle, a plataforma MEGA 2560 possui um *clock* de 16 MHz (WARREN JOSH ADAMS, 2011).

Figura 6 – Arduíno MEGA Rev3



Fonte: (MCROBERTS, 2013)

Tabela 4 – Características Arduino MEGA 2560

Microcontrolador	ATmega2560
Tensão Operacional	5V
Tensão de Entrada (Limites)	7-12V (6-20V)
Digital I/O <i>pins</i>	54 dos quais 15 possuem saída PWM
Pinos de Entrada Analógica	16
Corrente DC por I/O <i>pin</i>	40 mA
Corrente DC do <i>pin</i> 3,3V	50 mA
Memória Flash	256 KB (ATmega2560) com 8 KB para o <i>bootloader</i>
SRAM	8 KB (ATmega2560)
EEPROM	4 KB (ATmega2560)
<i>Clock</i>	16 MHz
Comprimento	101,6 mm
Largura	53,4 mm
Peso	150 g

Fonte: (EVANS JOSHUA NOBLE, 2013)

A tensão operacional recomendada para o funcionamento da placa Arduino Mega 2560 está indicada na faixa de 7 a 12 volts (WENGER, 2009), entretanto pode-se considerar

uma tensão mínima de serviço em 5 volts para que a plataforma Arduino MEGA 2560 opere normalmente ([BARTMANN DANIELLE LAFARGE, 2015](#)).

Entre as várias vantagens da placa Arduino MEGA 2560, destaca-se as quatro portas UART TTL (5V) de comunicação serial, uma maior quantidade de pinos PWM, bem como o grande espaço de memória que pode ser utilizada no âmbito de aplicações que requeiram muitos sensores e atuadores conectados a plataforma microprocessada Arduino MEGA 2560 ([BARTMANN DANIELLE LAFARGE, 2015](#)).

## 2.7 Histórico de Aplicação de Sistemas de Automação Com o Arduino

Diante do avanço tecnológico dos sistemas de automação e de uma demanda maior por uma alta velocidade nos níveis de processamento e resposta de ação de controle, o uso da placa micro-controladora Arduino de código-fonte aberto para desenvolvimento foi um dos vários meios encontrados para sanar problemas que envolviam o alto custo na fase de construção e uma maior ocupação de espaço físico para interligação de vários dispositivos eletrônicos. Com a simplicidade oferecida pela plataforma Arduino, todos esses dispositivos encontram-se encapsulados em uma mesma placa, garantindo robustez, flexibilidade e economia de espaço em projetos de montagem e implementação robótica. Dentro da linha histórica de utilização de microcontroladores Arduino em sistemas de automação, verificou-se a necessidade crescente de emprego da plataforma Arduino em projetos no setor de automação residencial a partir de 2008.

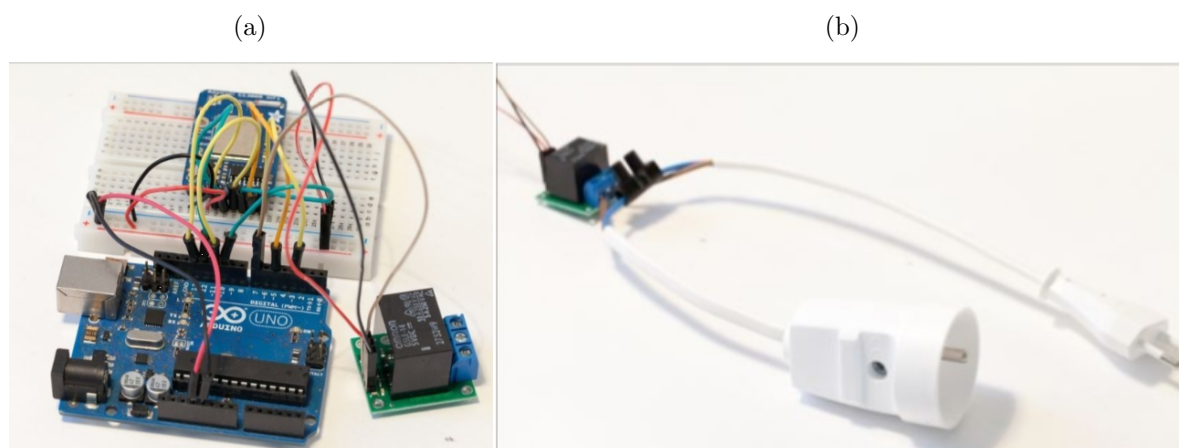
As pesquisas referentes a automação residencial começaram a gerar resultados a partir do início da década de 1970 com a invenção de dispositivos de uso doméstico como a máquina de café Coffee ([RILEY, 2012](#)). Segundo ([SCHWARTZ, 2014](#)), o conceito de automação residencial consiste no fornecimento de um produto ou serviço que executa ações de controle ou libera mensagens de monitoramento para um ambiente residencial, sem a intervenção humana. Sua lógica de atuação se baseia na ação de implementar estratégias de controle e automatização de diversas tarefas executadas em residências como ligar lâmpadas, fechar portas ou acionar dispositivos de alarme ([SCHWARTZ, 2014](#)).

Existe uma diversidade de equipamentos e sistemas de automação residencial fornecidos por lojas e distribuidoras, entretanto, várias dificuldades são geradas em razão do alto custo desses dispositivos e da impossibilidade de personalização desses componentes para situações específicas. Outra questão que inviabiliza o uso de dispositivos para a automação residencial é a ausência de um sistema de interface que permita a comunicação desses equipamentos com computadores ou celulares. Por essas razões restritivas, opta-se pela utilização da plataforma Arduino com o objetivo de implementação de sistemas eletrônicos robustos que atendam as especificidades de um projeto de automação residencial

(SCHWARTZ, 2014).

Na figura 7 é mostrado um exemplo de sistema de automação implementado em 2014. Sua função é direcionada para o controle de lâmpadas residenciais, desenvolvido para ser operado remotamente por celulares ou tablets.

**Figura 7** – (a) Módulo de controle principal do sistema de gerenciamento de lâmpada ; (b) Conjunto de conexão da lâmpada com o relé



Fonte: (SCHWARTZ, 2014)

Outro setor que se destacou nos avanços com relação a automação foram os sistemas de irrigação de plantas, hortas e grandes fazendas. O sistema de irrigação por gotejamento, provido de uma plataforma Arduino conectada a sensores de distância e outros componentes eletrônicos de monitoramento tornam o sistema mais otimizado (AGRAWAL NIKHIL; SINGHAL, 2015). Como exemplo, é apresentada na figura 8 um protótipo de automação de um sistema de irrigação implementado em 2015.

**Figura 8** – (a) Sistema de irrigação, mostrando a instalação convencional de tubulações; (b) Módulo de controle do sistema de irrigação implementado pela plataforma Arduino



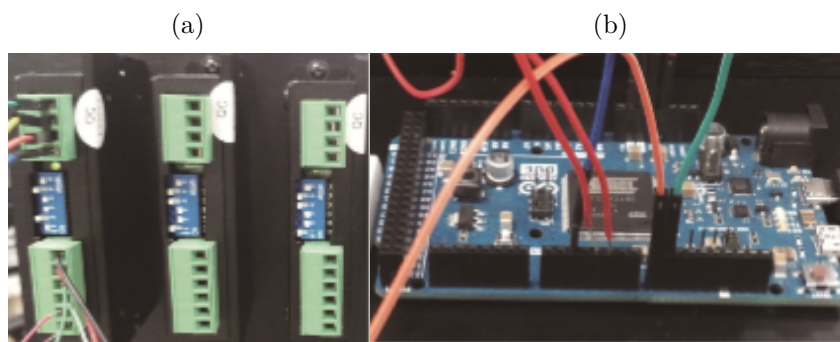
Fonte: (AGRAWAL NIKHIL; SINGHAL, 2015)

Embora o projeto de irrigação automatizada descrito acima tenha atendido os requisitos de regulação da quantidade de água usada no processo, o sistema não foi

preparado para reportar a ocorrência de possíveis falhas nos componentes. Essa complicação da falta de informação sobre o funcionamento dos dispositivos abriu possibilidades para o aperfeiçoamento do sistema (AGRAWAL NIKHIL; SINGHAL, 2015).

A utilização da plataforma Arduino em projetos de automação tem beneficiado a área da saúde com destaque para os estudos e análises genéticas. No ano de 2015, pesquisadores da Universidade de Chongbuk, na Coreia do Sul, desenvolveram um sistema de extração de DNA controlado por um módulo composto por atuadores lineares e de passo e uma placa Arduino DUE. Na figura 9 é mostrada a descrição do sistema de automação para extração de DNA.

**Figura 9** – (a) Atuador linear com interruptores de limite residenciais e negativos;(b) Placa Arduino DUE



Fonte: (HSIANG, 2015)

Os atuadores de passo foram usados para o processo de mistura e movimentação das amostras genéticas analisadas. Através de um programa de computador, o código G das amostras era compilado, sendo que a sequência de operações obtida pela compilação era enviada a placa Arduino, fazendo com que os atuadores lineares conectados ao módulo de controle procedessem com a correta extração de DNA das amostras (HSIANG, 2015).

Todos os exemplos de aplicação da plataforma Arduino apresentados anteriormente, nos sistemas de automação e controle residenciais, nos sistemas de irrigação ou na pesquisa clínica de extração de genes, provam que a tecnologia empregada tem sido bem difundida no setores onde se necessita de aprimoramentos para garantir segurança de operação, confiabilidade e desempenho considerável. Por essa razão foi escolhida a plataforma Arduino para o desenvolvimento principal da lógica de programação, com o objetivo de controlar e operar a esteira de transporte e seleção desenvolvida no presente trabalho.

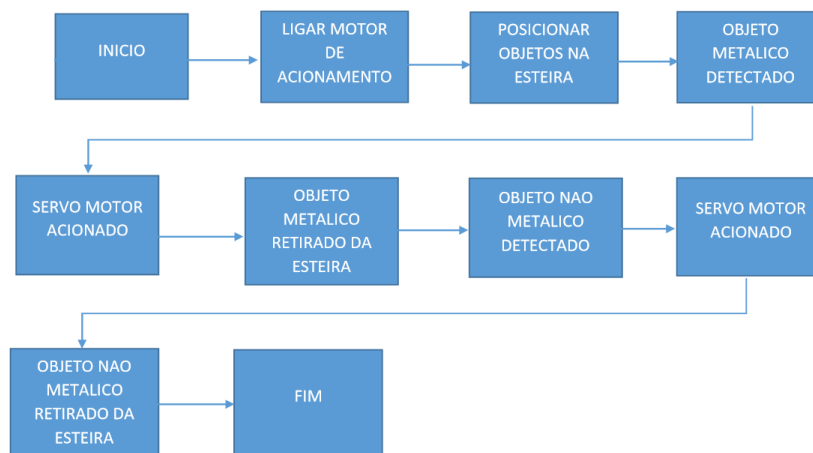
## 3 Projeto de Automação da Esteira Transportadora

Neste capítulo será apresentada uma abordagem detalhada sobre o desenvolvimento do projeto de automação e acionamento da esteira transportadora, levando em consideração os aspectos relacionados ao tipo de acionamento mecânico da correia transportadora, a estrutura de programação utilizada no âmbito da plataforma microprocessada Arduino MEGA 2560 e os conjuntos de restrições de operação.

### 3.1 Desenho da Esteira Didática de Transporte e Seleção

O objetivo principal da esteira de transporte e seleção constitui-se em mover peças retangulares, e automaticamente, selecioná-las de acordo com suas propriedades magnéticas. A representação do projeto em diagrama de blocos é mostrada na figura 10.

**Figura 10** – Diagrama de blocos descrevendo a operação e dinâmica de funcionamento da esteira

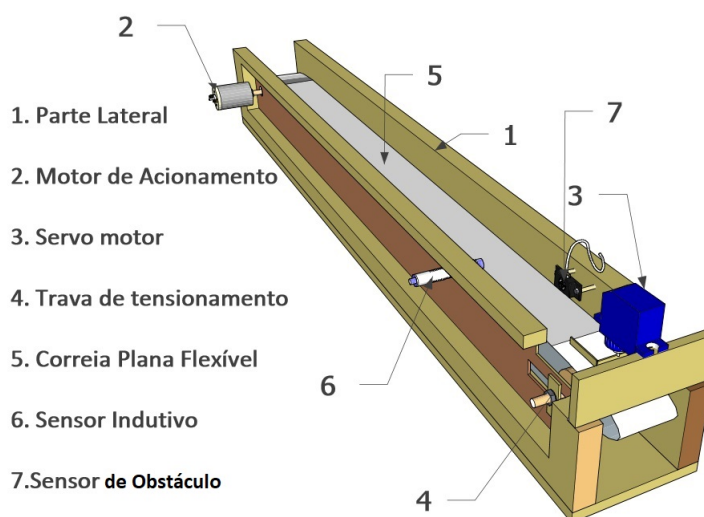


Fonte: Próprio Autor

Na montagem do sistema transportador, foram utilizados materiais simples como compensado para a confecção das bases de fixação laterais da esteira transportadora. Inicialmente nesta fase de projeto, foi pensado em empregar um motor DC (corrente contínua) para movimentar a correia. Entretanto, durante alguns testes em laboratório verificou-se uma alta rotação de partida que poderia danificar a correia ou a polia de acionamento, por essa razão foi utilizado um motor síncrono com uma partida mais leve e que será descrito com mais detalhes na subseção 3.3.1. Uma representação em 3D da esteira antes da montagem é mostrada na figura 11.

O programa usado no desenho tridimensional da esteira foi o Sketchup. Este software é utilizado na criação de ambientes e objetos tridimensionais, contribuindo para a confecção de maquetes virtuais e outras representações esquemáticas em 3D (BLEICHER, 2014).

**Figura 11** – Desenho em 3D da esteira com destaque para alguns componentes principais.



Fonte: Próprio Autor

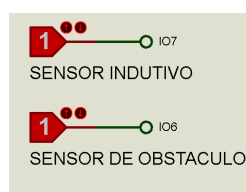
Para a detecção dos materiais foi pensado em um sensor indutivo para detecção de metais e um sensor capacitivo para detectar objetos não-metálicos. Nesta fase de planejamento o sensor capacitivo foi pensado em razão da sua característica de detectar qualquer massa como resinas, madeira ou plástico. Entretanto, em razão do custo de compra do dispositivo e da doação de um sensor de obstáculos para a montagem, o sensor de obstáculos foi o dispositivo utilizado para a detecção do material não metálico.

## 3.2 Simulação do Circuito Eletrônico no Software Isis Proteus

A simulação virtual de funcionamento da esteira foi implementada com o software PROTEUS, cuja versão 8.6 é acompanhada de um compilador próprio para simulações e animações virtuais que empregam a plataforma microprocessadora Arduino.

As ações de entrada obtidas pela leitura dos sensores indutivo e de obstáculos foram representadas por elementos denominados chaves lógicas, cuja saída é digital, se adequando a realidade de saída digital dos dois sensores utilizados no sistema de seleção de peças. A simbologia grafica é mostrada na figura 12.

**Figura 12** – Chaves lógicas de dois estados para a representação dos sensores



Fonte: Próprio Autor

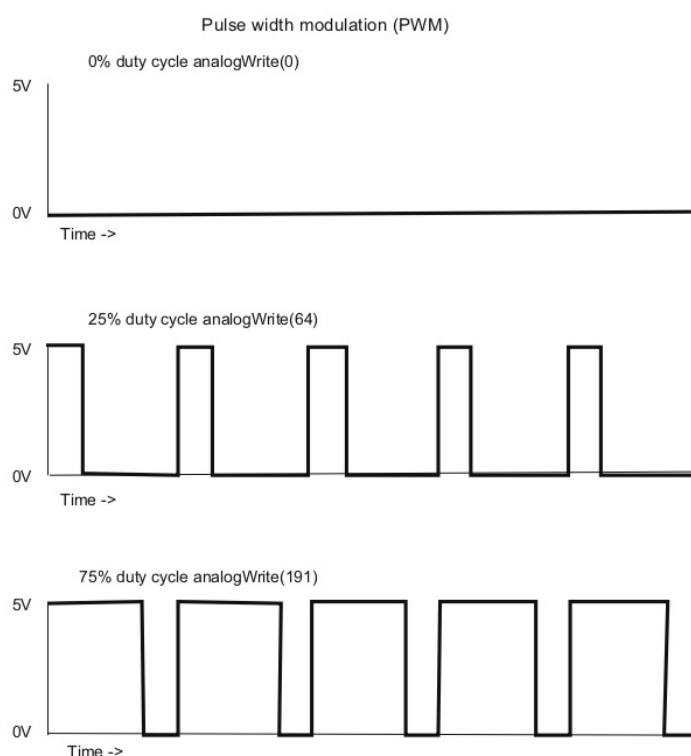
Caso a chave lógica que representa o sensor indutivo seja colocada em nível BAIXO(0), indica que um objeto metálico foi detectado. Se a chave lógica que representa o sensor de obstáculo for colocada em nível BAIXO (0), significa que um objeto não metálico foi detectado.

O motor de acionamento da esteira na simulação no programa PROTEUS foi representado por um motor CC de 12V, alimentado por uma fonte de tensão de 12V. O dispositivo motriz é controlado por um dos quatro canais disponíveis no módulo relé cuja função é proceder com a devida ativação ou parada do motor de acionamento da esteira, ou seja, quando a correia estiver sem material o motor será ligado, caso ocorra detecção de algum objeto no fim de curso da esteira, o motor de acionamento é desligado para que ocorra a identificação do material e a sua respectiva separação.

### 3.2.1 Sinal PWM

A sigla PWM significa Modulação por Largura de Pulso e consiste em pulsar rapidamente um sinal digital, com o objetivo de se reproduzir uma saída analógica. O sinal PWM é fornecido pelo microcontrolador Arduino, através de pinos identificados na placa com o símbolo til. Na figura abaixo vemos exemplos de sinais PWM com diferentes larguras de pulso.

**Figura 13** – Modulação Por Largura de Pulso (PWM).



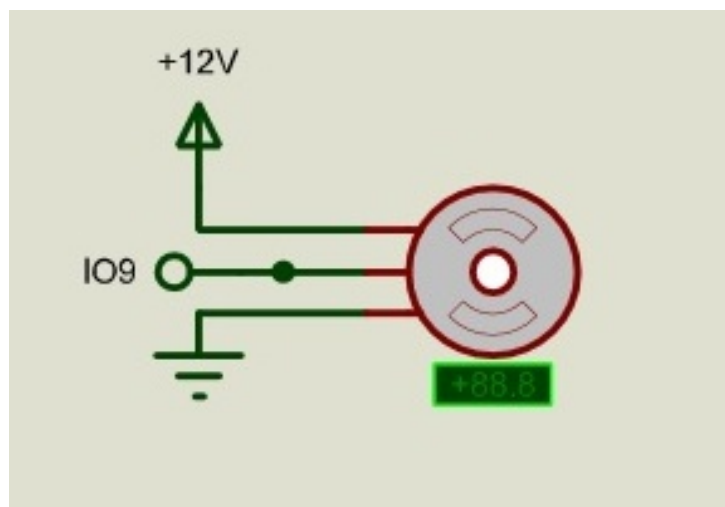
Fonte: (EVANS JOSHUA NOBLE, 2013)

Através da variação da largura de pulso do sinal PWM, pode-se controlar a posição do eixo do servomotor, utilizando alguns comandos da biblioteca servo.h.

### 3.2.2 Controle de Posição do Servomotor

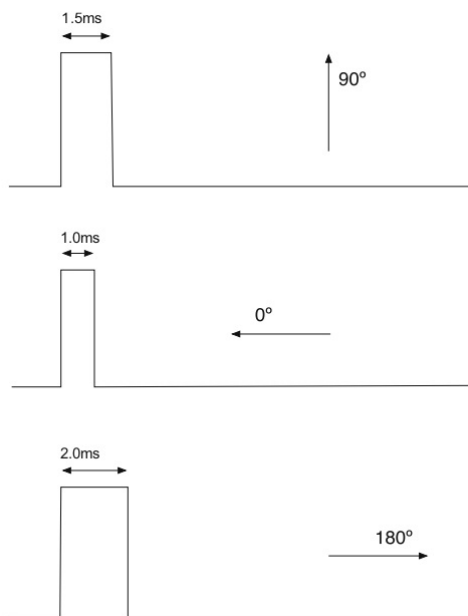
O dispositivo responsável pela ação de seleção na simulação é o servomotor. Através da biblioteca servo.h foi possível programar o movimento do eixo do motor dentro de uma faixa de 180 graus através do sinal PWM, enviado do pino 9 da placa Arduino MEGA. A conexão do referido pino ao servomotor é mostrada na figura 14

Através do comando "write(angulo)" da biblioteca servo.h pode-se definir os ângulos de giro do servo, com base nos sinais digitais de nível BAIXO emitidos pelos sensores indutivo ou de obstáculo. Esse ângulo é correspondente ao tamanho da largura de pulso

**Figura 14** – Servomotor conectado ao pino 9 (I09) do Arduino para receber o sinal de controle PWM.

Fonte: Próprio Autor

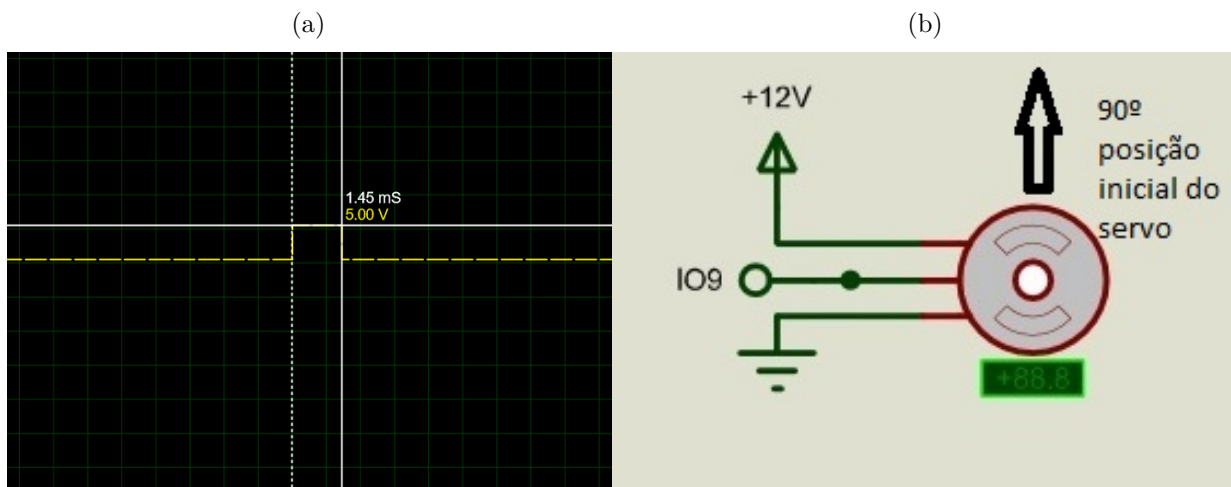
do sinal PWM enviado pelo microcontrolador Arduino. Na figura 15 é mostrado uma representação gráfica da correlação entre a largura de pulso do sinal PWM e a posição do servomotor.

**Figura 15** – Relação entre a largura de pulso e o ângulo do servomotor.

Fonte: (EVANS JOSHUA NOBLE, 2013)

O valor de largura de pulso PWM de 1.45 milissegundos na figura 16a corresponde a posição inicial do servomotor em 90 graus, definida através do comando "servo write(90)", quando a esteira está em movimento e não ocorre detecção de nenhum material. Nesta figura selecionamos apenas um pulso do sinal PWM para melhor visualização da sua largura em milissegundos.

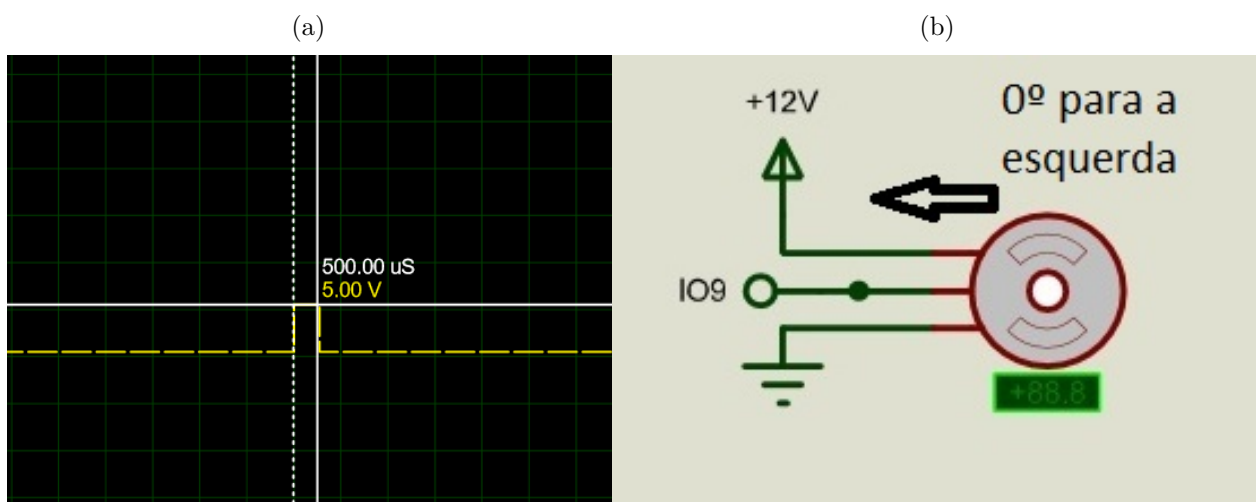
**Figura 16** – (a) Valor de largura de pulso do sinal PWM correspondente a posição inicial do servo na simulação ; (b) Posição inicial do servo na simulação no programa PROTEUS



Fonte: Próprio Autor

O valor de largura de pulso de 500 microssegundos indicado na figura 17a está relacionado a posição mínima do servo definida através do comando "servo write(0)", depois do sensor indutivo retornar o nível lógico BAIXO, indicando que é necessário deslocar um objeto metálico para a esquerda.

**Figura 17** – (a) Valor de largura de pulso do sinal PWM correspondente ao ângulo mínimo do servomotor ; (b) Ângulo mínimo do servomotor na simulação no programa PROTEUS

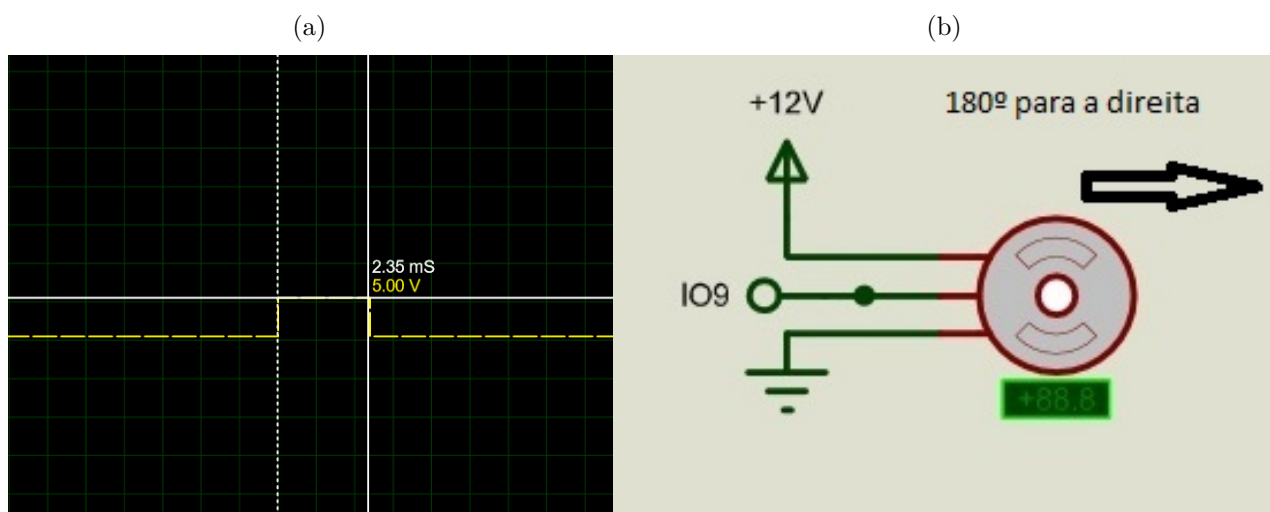


Fonte: Próprio Autor

O valor de largura de pulso de 2.35 milissegundos indicado na figura 18a está relacionado a posição máxima do servo definida através do comando "servo write(180)", depois do sensor de obstáculo ter apresentado nível lógico BAIXO, indicando que é necessário deslocar um objeto não metálico para a direita.

Dessa forma, foi possível realizar o controle de posição do servomotor na simulação realizada no programa PROTEUS, para posteriormente, ser aplicada no controle de posição

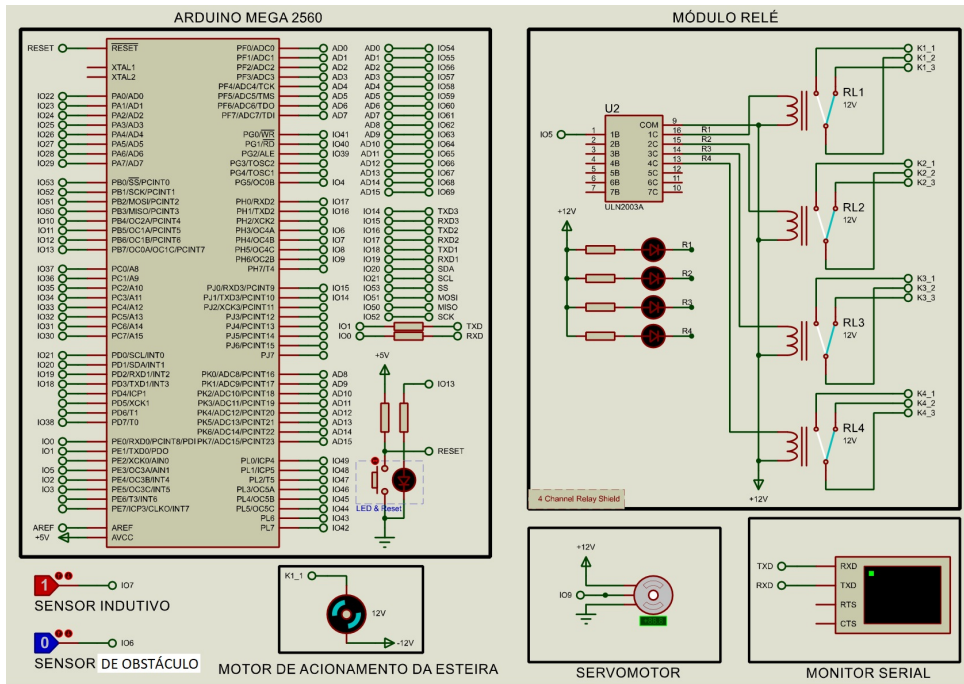
**Figura 18** – (a) Valor de largura de pulso do sinal PWM correspondente ao ângulo máximo do servomotor ; (b) Ângulo máximo do servomotor na simulação no programa PROTEUS



Fonte: Próprio Autor

do servomotor utilizado na etapa de montagem do protótipo

Figura 19 – Esquema de ligações do circuito de controle



Fonte: Próprio Autor

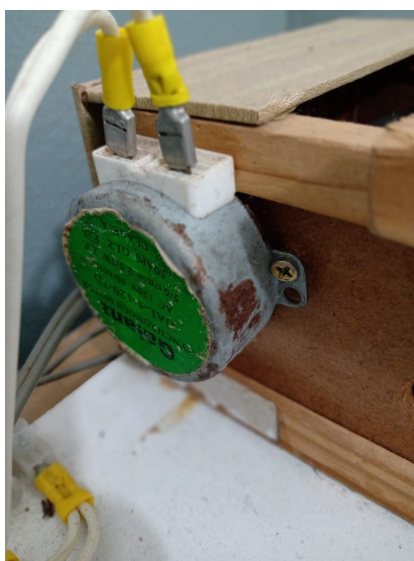
Foi inserido na simulação um monitor serial que transmite informações da plataforma microprocessada Arduino sobre o estado de operação do motor de acionamento e sobre o tipo de material detectado na extremidade do fim de curso da esteira, informando se o material é metálico ou não-metálico. Na figura 19 é mostrada a configuração final do circuito eletrônico implementado no programa PROTEUS.

## 3.3 Descrição dos Componentes do Sistema de Automação

### 3.3.1 Motor de Acionamento da Esteira

Foi utilizado o motor síncrono da marca GALANS para proporcionar movimento a correia. Esse modelo opera com tensão de 120V AC, o que permite que seja alimentado com uma tensão de 127V de uma tomada simples. Sua estrutura é constituída da carcaça de proteção e de dois pinos de conexão para fornecer a alimentação necessária para seu funcionamento. Na figura 20 é mostrado uma imagem do motor síncrono usado para desempenhar a função de acionamento da correia.

**Figura 20** – Motor Síncrono de acionamento da marca Galanz



Fonte: Próprio Autor

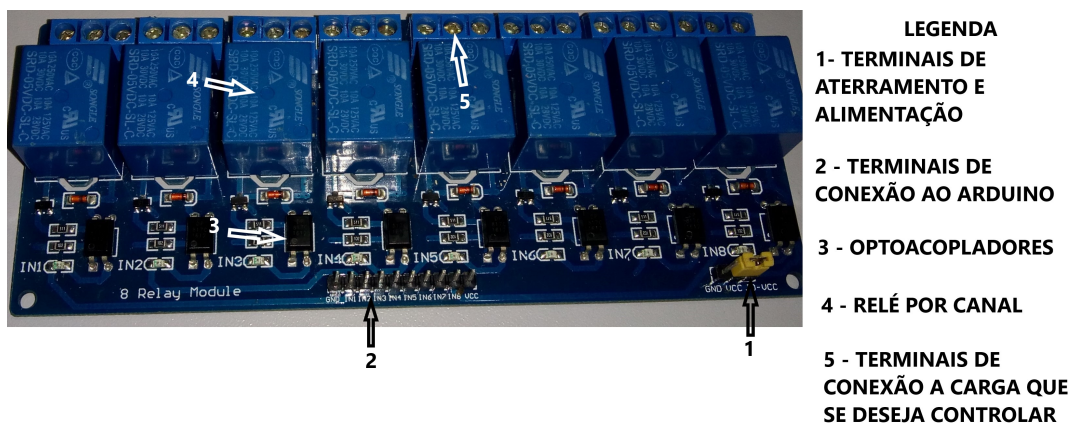
O modelo apresentado na figura acima apresenta uma velocidade de 5 RPM (rotações por minuto). Esta velocidade se mantém constante pois em um motor síncrono a velocidade de rotação é proporcional á frequência de alimentação do dispositivo, sendo que para o referido modelo, a frequência de operação é de 60 Hz.

### 3.3.2 Módulo Relé de 8 Canais

O módulo relé de 8 canais mostrado na figura 21 foi o dispositivo escolhido na etapa de planejamento para o controle do motor de acionamento da correia.

Em razão da ausência de recursos financeiros para o projeto, optou-se por um módulo relé com 8 canais baseado no modelo SRD-05VDC-SL-C, o qual foi doado, para a finalidade de auxiliar as ações de ativação e desativação do motor principal. Cada canal possui um relé equipado com um optoacoplador que realiza o isolamento do circuito eletrônico de baixa tensão do Arduino em relação ao circuito de alimentação do motor de

Figura 21 – Módulo relé de 8 canais



Fonte: Próprio Autor

acionamento. Seus pinos digitais de entrada recebem sinal de nível lógico alto ou baixo para acionar ou desligar as cargas conectadas aos terminais NA (normalmente aberto) ou NF (normalmente fechado).

### 3.3.3 Sensor Indutivo

O sensor indutivo com modelo de numeração LJ12A3-4-Z/BX foi o componente empregado no processo de detecção de objetos metálicos transportados pela esteira. Na figura 22 é mostrado uma foto do referido sensor utilizado.

Figura 22 – Sensor Indutivo



Fonte: Próprio Autor

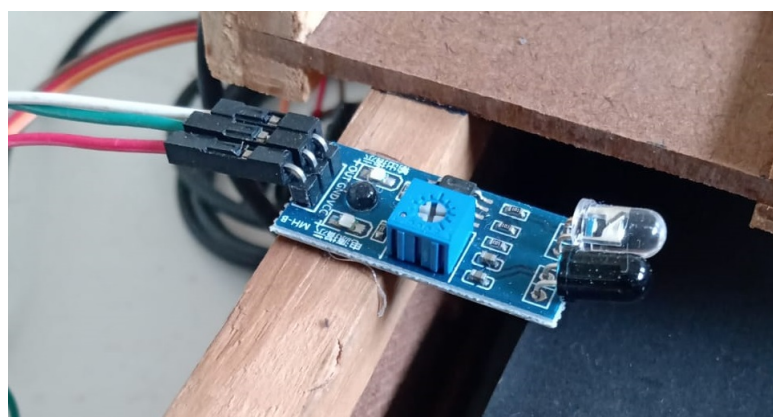
A tensão de operação do sensor indutivo é definida entre 6V a 36V de tensão contínua, com um alcance de detecção de 4mm. Seu funcionamento se dá através da variação do campo eletromagnético gerado pelo oscilador em conjunto com a bobina na extremidade faceada de cor azul do dispositivo, no momento em que um objeto se aproxima

da distancia de detecção. Isso gera uma resposta de nível lógico BAIXO na saída do sensor indutivo.

### 3.3.4 Sensor de Obstáculos

O sensor de obstáculos infravermelho apresentado na figura 23 foi o componente utilizado para a detecção do objeto não-metálico. Sua tensão de operação é de 3,5V a 5V. Possui um potenciômetro ajustável que permite regular a distância de detecção na faixa de 2cm a 30cm.

**Figura 23** – Sensor de Obstáculos Reflexivo



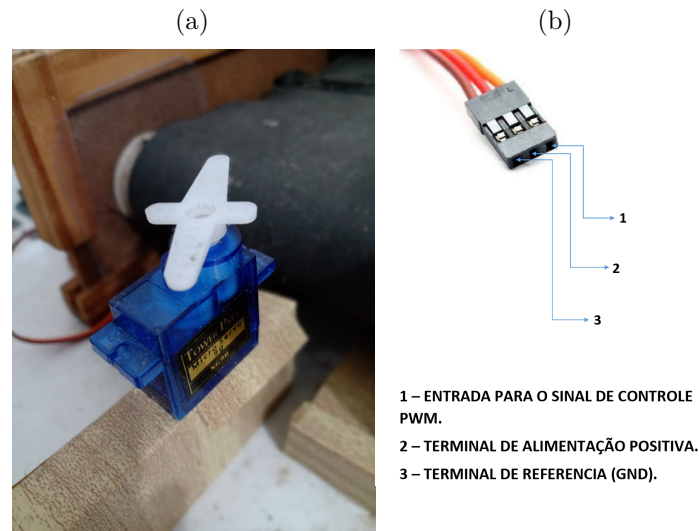
Fonte: Próprio Autor

Seu funcionamento consiste na reflexão de luz infravermelha. Quando um objeto passa dentro do ângulo de reflexão, o sinal infravermelho é enviado ao receptor do sensor, gerando uma saída de nível lógico BAIXO.

### 3.3.5 Servomotor

O servomotor SG90 foi o componente usado para selecionar o material metálico e não metálico, removendo o referido objeto de cima da correia. Sua tensão de operação é de 5V DC, sendo alimentado pelo pino VCC da placa Arduino que fornece essa tensão. Na figura 24 é mostrado o componente.

A figura 24a detalha a estrutura física do componente, a qual é constituída pela carcaça que protege as engrenagens e por três cabos de conexão que desempenham a função de energizar o servomotor e de atuar como entrada para o sinal de controle PWM enviado pela plataforma Arduino.

**Figura 24** – (a) Servo Motor ; (b) Conexões de alimentação e controle do servo motor

Fonte: Próprio Autor

### 3.4 Estrutura de Programação

A parte inicial de programação na IDE do programa Arduino é definida pelas configurações de inicialização e definição das bibliotecas de suporte como por exemplo a biblioteca servo.h configurada na linha 5 do programa. Temos também a definição das variáveis responsáveis pela alocação da informação proveniente dos sensores indutivo e reflexivo. Na figura 25 é mostrada essa parte do programa .

**Figura 25** – Configurações de variáveis e bibliotecas

```

1 // programa de Controle Principal da Esteira - Sub-rotina de controle
2
3
4
5 #include <Servo.h> // chamada da biblioteca servo
6 #define Relay1 5 // Definição do pino de controle do Rele 1
7 #define Transistor 10 // Definição do pino de controle do Rele 2
8 int sensorIndutivo = 7; // pino de controle do sensor indutivo
9 const int sensorReflexivo = 6; // pino de controle do sensor de obstáculos infravermelho
10 int valor; // variável de leitura do sensor reflexivo
11 Servo myservo; // cria um objeto para o servo
12

```

Fonte: Próprio Autor

Em seguida, na função void setup, mostrada na figura 26, são declarados os pinos da placa microprocessadora Arduino que serão utilizados como entradas ou saídas como os pinos dos sensores para recepção do sinal enviado pelos mesmos e os pinos de saída para controle do módulo relé e do transistor. Dentro da função de inicialização void setup foram definidos o pino de controle do servomotor e a posição do eixo do servomotor, setado para o ângulo inicial de 90°, desse modo garantimos que o eixo do servomotor estará na posição desejada para o recebimento do material que deverá ser detectado e selecionado.

**Figura 26** – Inicialização dos pinos de entrada e saída

```
14 void setup() {
15   Serial.begin(9600);
16   pinMode(Relay1, OUTPUT);           //Pino declarado como saída
17   pinMode(Transistor, OUTPUT);      // Pino declarado como saída
18   digitalWrite(Transistor, LOW);    // PINO DIGITAL 10 DO TRANSISTOR, SETADO PARA NIVEL ALTO
19   pinMode(sensorReflexivo, INPUT);   //pino declarado como entrada
20   pinMode(sensorIndutivo, INPUT);   // pino declarado como entrada
21   myservo.attach(9);                // atribui o objeto servo ao pino digital 9
22   myservo.write(90);
23 }
```

Fonte: Próprio Autor

Na função void loop descrita na linha 27 do programa, são definidos os comandos de automação para a seleção dos materiais na esteira com base na informação obtida pela leitura dos sensores. Com a utilização da estrutura condicional if, configuramos a dinâmica de funcionamento do motor de acionamento da esteira e do servomotor responsável pela seleção do material. Na figura 27 é apresentada quatro estruturas condicionantes definidas para as possíveis combinações de respostas digitais fornecidas pelos sensores do sistema de automação da esteira. Com a disposição aleatória de materiais metálicos e não-metálicos na parte física da correia, espera-se que o sistema de automação identifique e faça a correta seleção do material baseando-se na indicação dos sensores que ora podem apresentar nível lógico ALTO indicando ausência de objetos ou nível lógico BAIXO indicando a detecção de material na esteira.

Figura 27 – Declaração das estruturas condicionais

```

--
27 void loop() {
28   int val = digitalRead(sensorIndutivo);
29   valor = digitalRead(sensorReflexivo);
30   if (valor == 1 & val == 1) { // quando os sensores indutivo e de obstáculo enviam sinal alto 1, significa que não
// há detecção de objetos na extremidade final da esteira
31     Serial.println("sem detecção"); // escreve a informação no monitor serial
32     digitalWrite(Relay1,HIGH); // liga o motor
33     myservo.write(90); // seta o eixo do servomotor para o ponto inicial
34     delay(10); // aguarda 10 milissegundos
35   }
36
37   /* controle de separação de materiais, usando os comandos enviados ao servo */
38
39   if (valor == 0 & val == 1){
40     Serial.println("objeto nao-metalico detectado"); // imprime na tela a mensagem de objeto detectado pelo sensor de obstáculos
41     digitalWrite(Relay1,LOW); // desliga o motor de acionamento da esteira
42     delay(400); // aguarda 400 milissegundos
43     myservo.write(180); // ativa o servo para a retirada do material, enviando um tamanho de pulso de 2ms
// correspondente a 180°
44     delay(1000); //aguarda 1 segundos
45   }
46
47   if ( val == 0){ // se o sensor indutivo enviar sinal BAIXO,
48     Serial.println("objeto metalico detectado"); // imprime na tela que um objeto metálico foi detectado
49     digitalWrite(Relay1,LOW); // desliga o motor de acionamento da esteira
50     delay(400); // aguarda 400 milissegundos
51     myservo.write(0); //ativa o servo para a retirada de material enviando um tamanho de pulso de 1ms
// correspondente a 0°
52     delay(1000);
53   }
54
55   if ( val == 0 & valor == 0){ // se o sensor indutivo enviar sinal BAIXO,
56     Serial.println("objeto metalico e reflexivo"); // imprime na tela que um objeto metálico e reflexivo foi detectado
57     digitalWrite(Relay1,LOW); // desliga o motor de acionamento da esteira
58     delay(400); // aguarda 400 milissegundos
59     myservo.write(0); //ativa o servo para a retirada de material enviando um tamanho de pulso de 1m
// correspondente a 0°
60     delay(1000);
61   }
62 }

```

Fonte: Próprio Autor

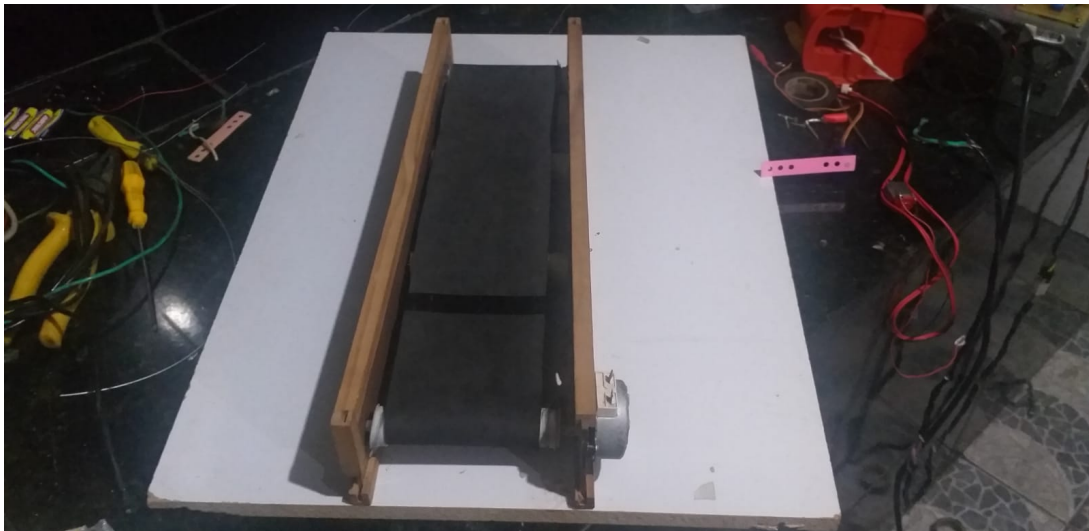
## 4 Montagem e Realização de Testes para Ajustes no Sistema de Automação

No capítulo que é apresentado agora, serão descritas as etapas de montagem do sistema de automação responsável pelo funcionamento da esteira de seleção. É importante salientar que durante essa fase, alguns componentes como motores ou sensores tiveram que ser substituídos, conforme as necessidades identificadas durante a realização dos testes e ensaios de operação. Essas alterações ocorreram até a última fase onde o protótipo atingiu as metas e configurações de projeto previamente determinadas.

### 4.1 Montagem da Esteira de Transporte

A implementação da esteira de transporte foi o primeiro passo tomado, já que seria necessário verificar as condições de torque e força necessárias antes da conexão de outros componentes eletrônicos indispensáveis no processo de automatização do sistema transportador de correia. Na figura 28 abaixo apresentamos a versão inicial concebida para a esteira, com as bases laterais de sustentação, correia e rolos condutores e intermediários.

**Figura 28** – Construção Inicial da Esteira



Fonte: Próprio Autor

Foi usada uma base retangular para a fixação das peças laterais de sustentação da correia, bem como o uso do motor síncrono de velocidade constante para acionar o rolo intermediário e proporcionar movimento a correia.

## 4.2 Inserção da Fonte de Alimentação

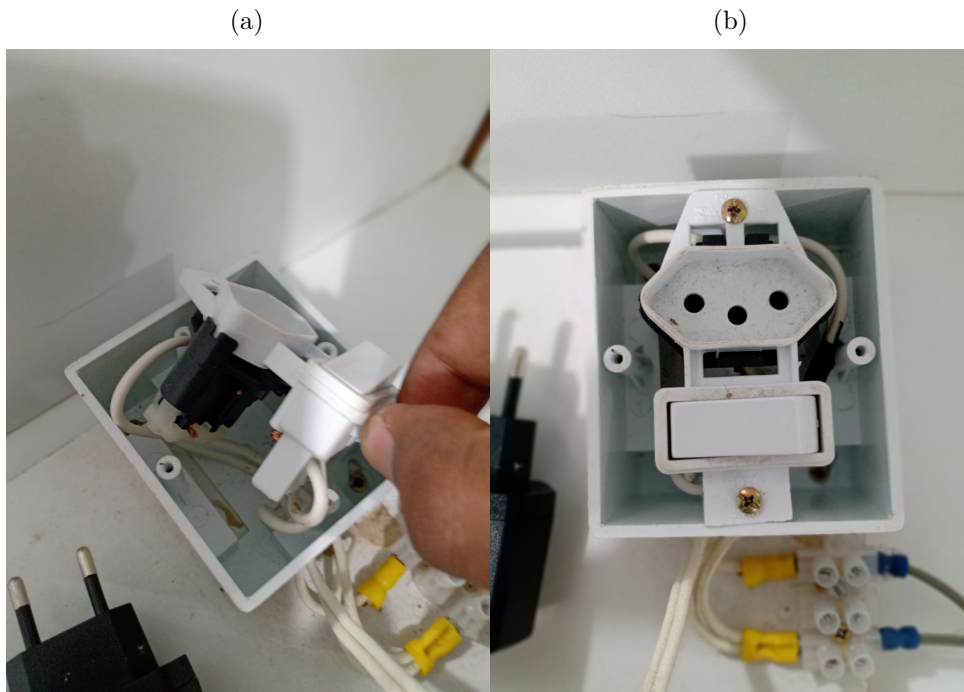
No decorrer da montagem, fez-se necessário fixar uma tomada na base retangular do projeto para suprir a demanda de energia requerida pela placa de controle Arduino e pelo motor de passo responsável pelo acionamento da correia. Para o devido funcionamento da placa Arduino MEGA 2560, foi empregado um adaptador para fornecer a alimentação de 5V necessária. A figura 29 mostra a bancada com o ponto fixo de alimentação instalado.

**Figura 29** – Instalação da Tomada Para Alimentação da Placa Arduino e do Motor de Passo



Fonte: Próprio Autor

Nesta etapa de instalação da fonte de alimentação, a escolha do ponto de fixação foi pensada de forma a facilitar as conexões de alimentação do motor de acionamento. Em um cenário de necessidade de manutenção ou troca do motor, as dificuldades seriam minimizadas sem recorrer a uma excessiva troca de cabos e condutores.

**Figura 30** – (a) Conexão interna na fonte de alimentação; (b) Tomada e interruptor instalados.

Fonte: Proprio Autor

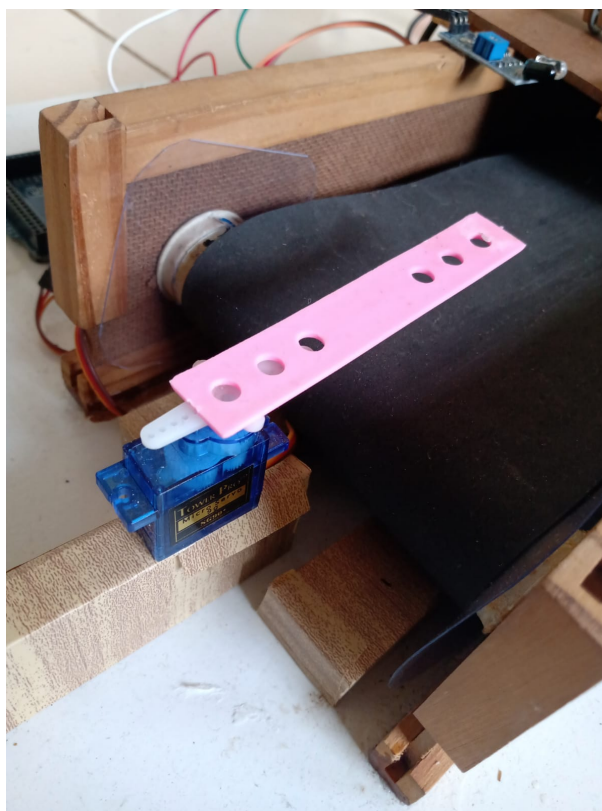
Na figura 30, é representado o esquema do circuito implementado na fonte de alimentação. Foi inserido o interruptor para ativar e desativar o motor síncrono, bem como uma ligação em derivação para a tomada que fornecerá a tensão para o adaptador da placa Arduino MEGA 2560.

Após a inserção da tomada, foi realizado o teste de desempenho do motor síncrono no processo de acionamento da correia transportadora a fim de verificar se ocorreriam problemas como torque insuficiente nos rolos condutores e intermediários ou deslizamento da correia sobre os rolos. Depois da fase de ensaios e testes de desempenho, foi verificado que o motor síncrono cumpria os requisitos de projeto necessários para que a correia fosse acionada sem possíveis falhas mecânicas durante sua movimentação.

### 4.3 Fixação do Servomotor

A inserção do servomotor na esteira transportadora foi uma das etapas mais importantes na confecção do projeto já que o dispositivo atuaria na seleção do material posicionado ao longo da correia. Foi determinado que o servomotor fosse instalado no fim de curso da esteira seguindo o esquema definido na figura 11. Conforme mostrado na figura 31 foi instalada uma base fixa para o servo bem como um separador composto de EVA para proceder com a separação do material no fim de curso da esteira. O eixo foi setado de forma que os sinais PWM do Arduino fornecessem o controle de ângulo correto para o processo de seleção.

Figura 31 – Fixação do Servomotor para seleção do material.

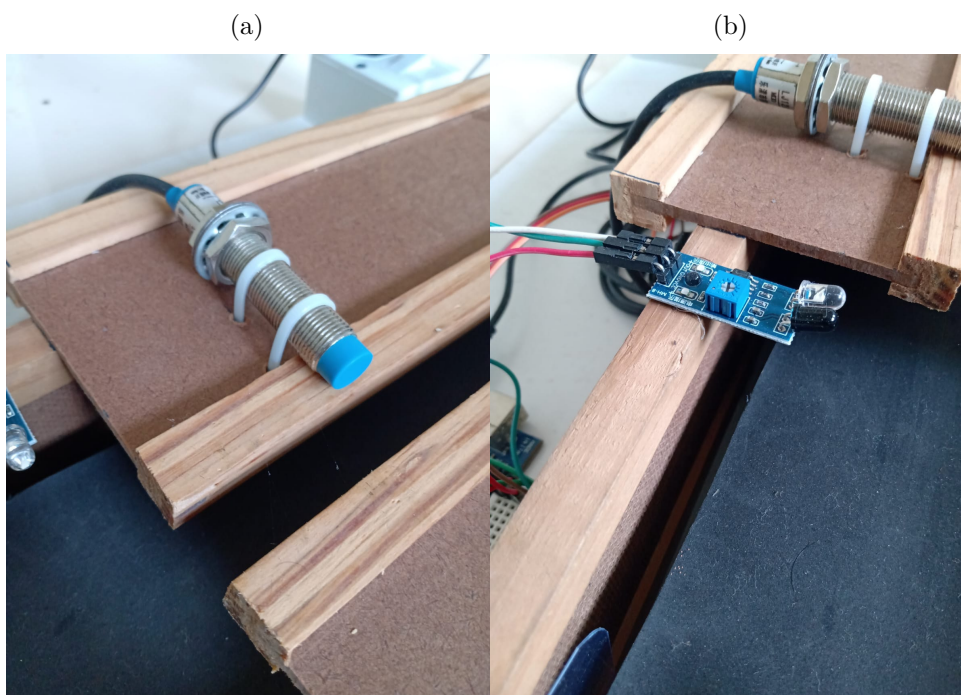


Fonte: Próprio Autor

### 4.4 Instalação dos Sensores Indutivo e Reflexivo

Nesta etapa de montagem, o trabalho inicial consistiu em determinar os melhores pontos para fixação dos sensores que atuariam na identificação do material disposto na esteira transportadora.

Na figura 32 são mostrados os sensores indutivo e de obstáculo instalados em pontos estratégicos para a correta identificação dos materiais transportados

**Figura 32** – (a) Sensor Indutivo; (b) Sensor de Obstáculo

Fonte: Proprio Autor

Podemos destacar diversas vantagens observadas no uso do sensor de obstáculo como a possibilidade de variação da distância de detecção através de um regulador presente na placa do dispositivo e a excelente resposta de detecção obtida na etapa de operação da esteira.

Com relação ao sensor indutivo, foi registrada uma limitação operacional evidenciada pelos dados técnicos de fabricação referentes a distância de detecção estipulada em 4mm. Por essa razão, a distância entre o material metálico e a face detectora do sensor teria que estar nessa faixa para que ocorresse a correta identificação do material.

## 4.5 Inserção de Placas Para Posicionamento de Material na Correia

Nesta etapa de montagem foi necessário instalar placas paralelas a correia para o direcionamento do material em razão de algumas vibrações na correia que poderiam interferir na detecção dos objetos. Para reduzir o efeito das vibrações na correia, foi efetuada a regulagem da folga entre a correia e os rolos condutores evitando que o tensionamento elevado da correia ocasionasse o seu rompimento e provocasse outras falhas no decorrer do acionamento da esteira transportadora.

Na figura 33 abaixo é mostrada a configuração de fixação escolhida para o equilíbrio do material transportado ao longo da esteira.

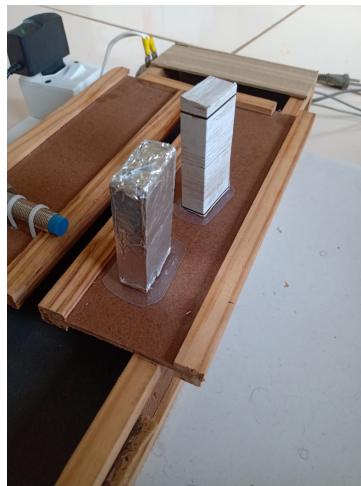
**Figura 33** – Fixação de placas em MDF para direcionamento do material transportado.



Fonte: Próprio Autor

Dois objetos sendo um confeccionado em madeira e outro recoberto com papel-alumínio foram definidos como os materiais que representariam respectivamente o objeto não-metálico e metálico que seriam usados nos testes e ensaios para verificar a capacidade de resposta dos sensores e a velocidade de atuação do servomotor no processo de separação do material transportado. Na figura 34 abaixo os objetos para teste de funcionamento da esteira são apresentados.

**Figura 34** – Objetos usados no processo de teste de funcionamento da esteira de transporte e seleção.



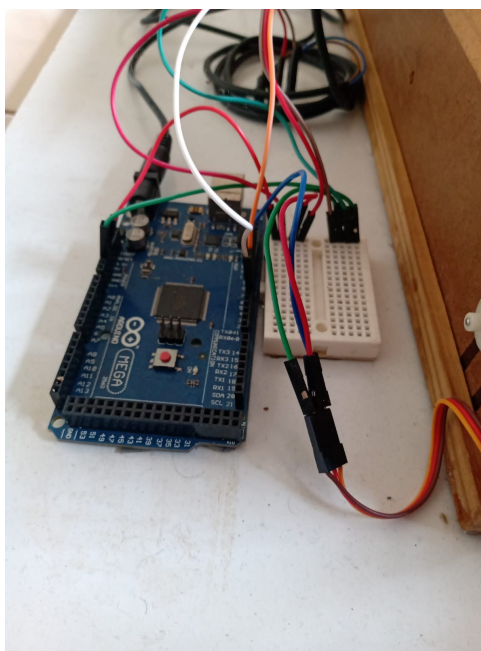
Fonte: Próprio Autor

## 4.6 Fixação da Placa Arduino MEGA 2560

Foi realizada a fixação da placa Arduino MEGA 2560, cuja alimentação esteve assegurada com um adaptador apropriado conectado ao terminal de alimentação da

placa Arduino. A localização da placa Arduino MEGA 2560 na bancada foi determinada analisando-se a disposição dos sensores e do servomotor que necessitavam estar próximos para evitar o uso excessivo de fios condutores no sistema transportador. Na figura 35 abaixo é mostrado a placa Arduino conectada a um pequeno protoboard para auxílio das conexões.

**Figura 35** – Placa Arduino MEGA 2560.

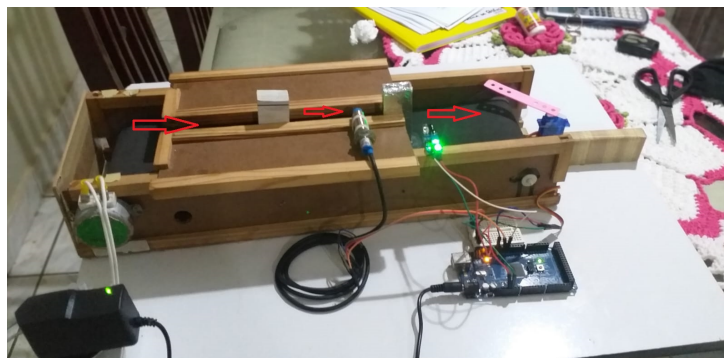


Fonte: Próprio Autor

## 4.7 Fase Final de Montagem e Teste de Funcionamento

A fase final de montagem e testes de funcionamento da esteira transportadora consistiu na verificação dos componentes e acionamento do motor de passo para avaliação dos objetivos de desempenho determinados na fase de planejamento de projeto, definidos como a estabilidade da correia, a seleção correta e precisa dos objetos transportados, revisão do programa principal de controle da placa Arduino bem como a promoção de melhorias e outras alterações que acrescentassem mais funcionalidades no sistema transportador que foi implementado. Na figura 36 abaixo é apresentada a versão final do projeto montado para a última etapa de testes e verificação de desempenho operacional.

**Figura 36** – Esteira transportadora em funcionamento. As setas em vermelho indicam o sentido de deslocamento do material, da esquerda para a direita.



Fonte: Próprio Autor

## 5 Resultados e Discussões

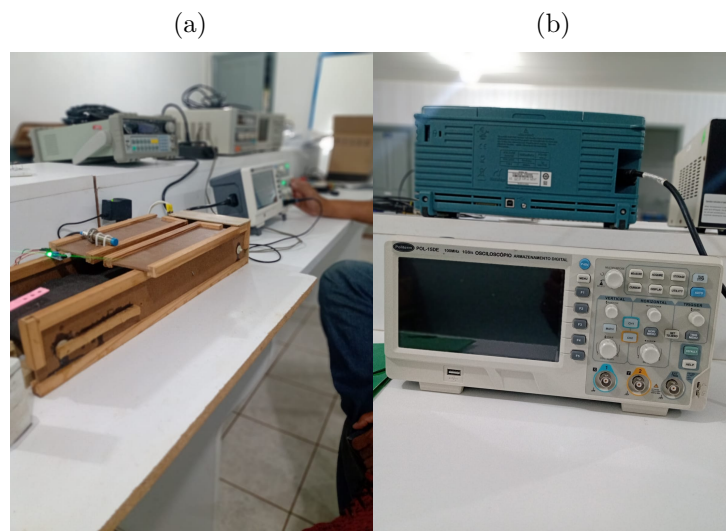
No presente capítulo serão apresentados os resultados do projeto desenvolvido com base nas especificações de desempenho definidas nas etapas de planejamento. Serão abordadas propostas de melhoria para o sistema de transporte desenvolvido no referido trabalho, dando ênfase em ideias como o controle de velocidade do motor de acionamento da correia ou a instalação de um braço robótico articulado para selecionar objetos com base em outras características físicas detectadas por diferentes tipos de sensores.

### 5.1 Resultados de Funcionamento do Sistema de Transporte

O pequeno sistema de transporte implementado e controlado pela plataforma Arduino obteve êxito em seu funcionamento, apesar de todas as dificuldades causadas pelas surpresas do trabalho manual e prático referente a montagem possibilitando que novas perspectivas de operação e funcionalidades pudessem ser exploradas em um futuro trabalho de aprimoramento do sistema.

Na figura 37a, é apresentada uma foto do sistema transportador em funcionamento no laboratório de circuitos elétricos da UFPA.

**Figura 37** – (a) Esteira transportadora em funcionamento no laboratório de circuitos elétricos do NUT/UFPA; (b) Modelo de osciloscópio utilizado para obter o sinal PWM enviado do Arduino para o servomotor



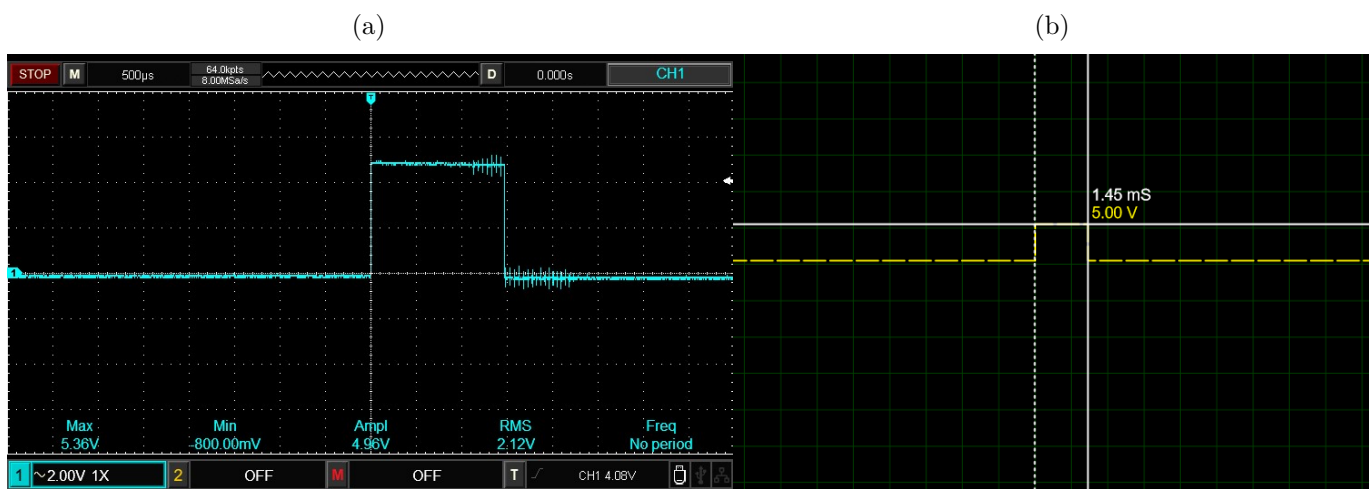
Fonte: Proprio Autor

Foi utilizado no laboratório de circuitos elétricos da UFPA o osciloscópio digital modelo POL-15DE da marca Politerm, o qual é mostrado na figura 37b. O equipamento

possui várias especificações como largura de banda de 100 MHz, medições automáticas de 28 formas de ondas e disponibilidade de dois canais para coleta de sinais digitais.

Foram coletados os sinais PWM nos instantes de detecção dos materiais metálicos e não-metálicos. O objetivo seria comparar os valores de tensão e largura do sinal com os parâmetros apresentados nas figuras 15 e 16. Na figura 38 é mostrado o sinal PWM na etapa de funcionamento da esteira sem detecção de materiais. A amplitude do sinal é de 4.96V, sendo que na simulação no programa PROTEUS tivemos um valor de amplitude de 5V do sinal PWM.

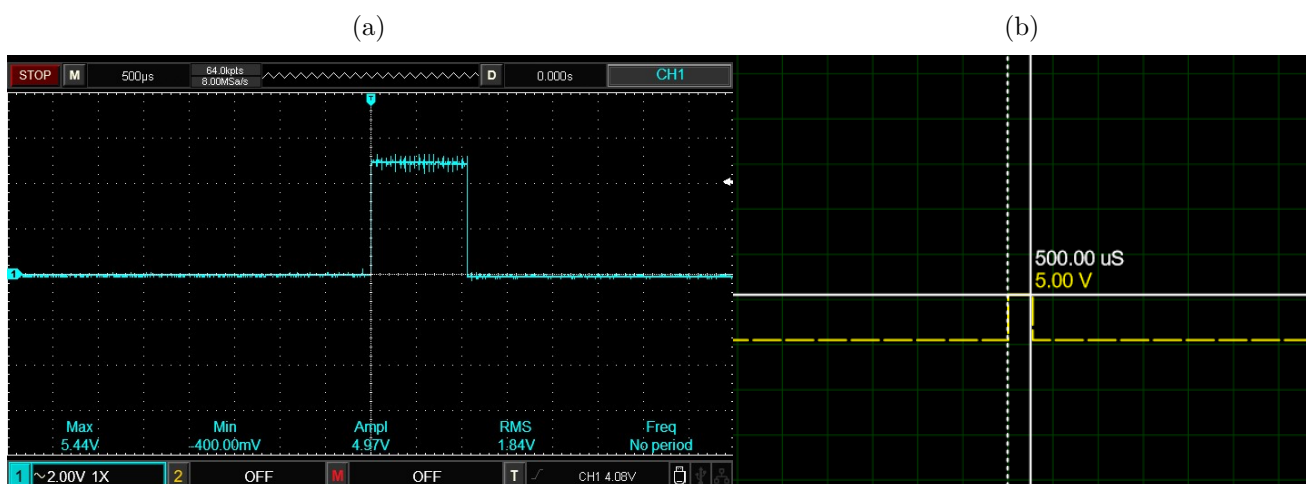
**Figura 38** – (a) Sinal PWM visualizado em laboratório na etapa em que a esteira opera sem detecção de materiais; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS



Fonte: Proprio Autor

Na etapa de detecção do material metálico, o sinal PWM manteve sua amplitude em 4.97V, apresentando uma diminuição do duty cycle positivo, que é a largura do sinal durante o período em que a amplitude está no seu valor máximo. Na figura 39 é representado a comparação entre os sinais PWM visualizados em laboratório e na simulação no programa PROTEUS, durante a detecção de metal.

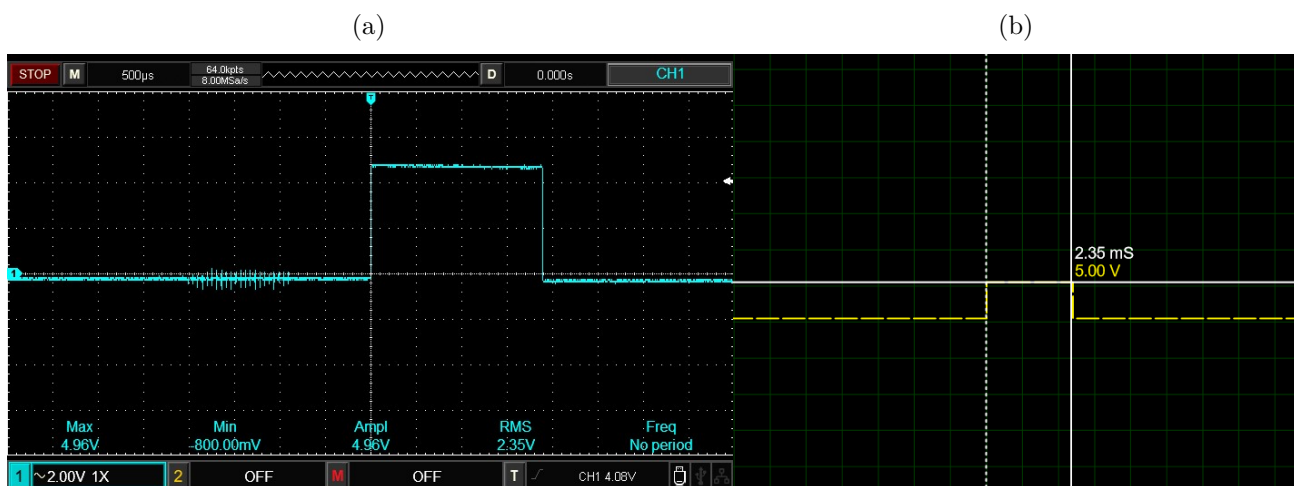
**Figura 39** – (a) Sinal PWM visualizado em laboratório na etapa em que ocorre a detecção de metal; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS na etapa de detecção de metal



Fonte: Proprio Autor

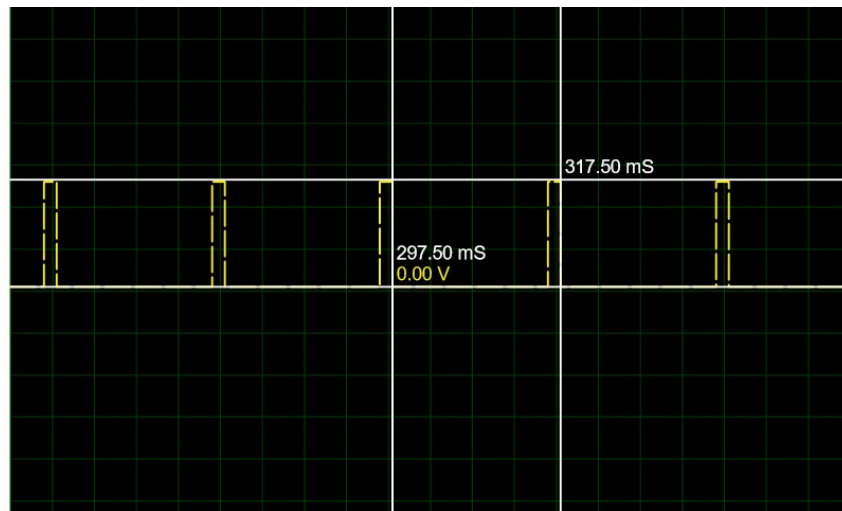
Na etapa de detecção de material não-metálico, o sinal PWM visualizado em laboratório manteve a mesmo valor de amplitude de 4.96V, mostrando um aumento de largura do pulso, na qual o sinal apresenta a amplitude máxima. Na figura 40 é mostrado a comparação entre os dois respectivos sinais PWM visualizados em laboratório e durante a simulação.

**Figura 40** – (a) Sinal PWM visualizado em laboratório na etapa em que ocorre a detecção de madeira; (b) Sinal PWM visualizado na simulação por meio do programa PROTEUS na etapa de detecção de madeira



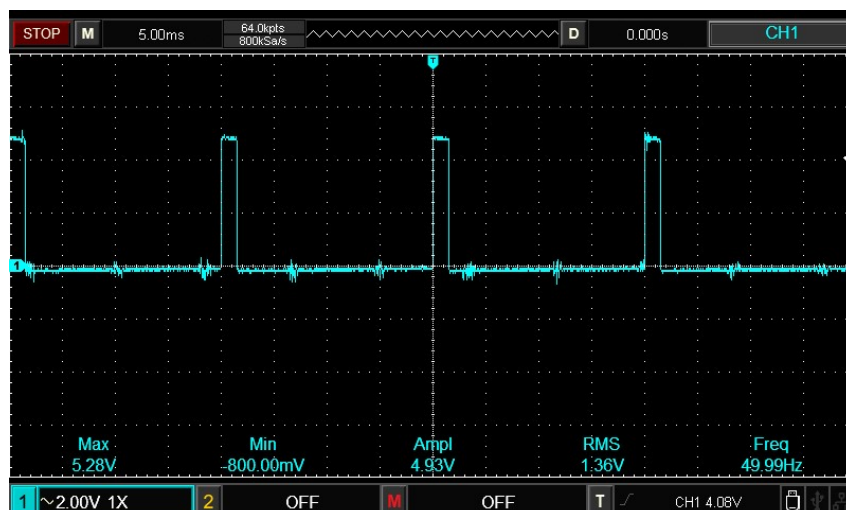
Fonte: Proprio Autor

Se inserirmos na figura 38b uma divisão horizontal de 5 milissegundos, obtemos uma forma de onda em que podemos coletar o período do sinal PWM enviado pelo Arduino ao servomotor. Aplicando-se a configuração a simulação obtemos a figura 41.

**Figura 41** – Trem de pulsos PWM na simulação com o Proteus.

Fonte: Próprio Autor

O espaço entre os valores de milissegundos destacados na figura 41 correspondem ao período do sinal PWM. Logo, se subtrairmos os valores, obtemos um resultado de 20 milissegundos. Como a frequência é inversamente proporcional ao período de um sinal periódico, logo determinamos que a frequência do sinal PWM na simulação é de 50 Hz. Aplicando a subdivisão horizontal de 5 milissegundos no osciloscópio em laboratório, obtemos a forma de onda representada na figura 42 indicando uma frequência de 50 Hz, mostrada no canto inferior direito da tela.

**Figura 42** – Trem de pulso PWM representado em laboratório por meio do osciloscópio.

Fonte: Próprio Autor

Com base nos resultados gráficos obtidos com o osciloscópio, pode-se dizer que o sistema de transporte e seleção de peças apresentou um funcionamento eficiente e adequado com as simulações feitas no software PROTEUS.

## 5.2 Observações Adicionais

Um aspecto importante que merece ser destacado foi a decisão de não ser utilizado o módulo relé de 8 canais para a ação de acionamento do motor responsável pela movimentação da correia. Durante o processo de implementação, com a mudança da opção de um motor DC para um motor síncrono de velocidade constante, verificou-se que o módulo relé seria empregado sem relevância. Por essa razão, foi efetuada a instalação de um simples interruptor integrado a um adaptador, com o objetivo de fornecer a tensão de 120V AC necessárias para o acionamento do motor síncrono, economizando espaço físico e conferindo ao projeto uma configuração mais compacta e integrada.

Os sensores de detecção de objetos foram fundamentais na dinâmica de operação do sistema transportador conferindo praticidade e precisão ao trabalho. O sensor de obstáculos reflexivo, apresentou uma excelente capacidade de detecção, potencializada pelo mecanismo de variação de alcance do sensor, conferindo ao sistema de transporte propostas de melhorias baseadas em outros pontos de fixação do dispositivo para diversas funcionalidades como detecção de objetos estranhos na correia ou acionamento de mensagem de alerta caso fosse detectado a aproximação de materiais que apresentassem perigo ao funcionamento da esteira.

## 5.3 Orçamento Financeiro do Protótipo

Conforme a tabela abaixo, foi investido um valor total de 60 reais referente a compra da placa Arduino MEGA e do sensor indutivo. Todos os demais componentes e materiais foram obtidos por meio de doações privadas de pessoas que se solidarizaram com a iniciativa. Dessa forma foi atingida a meta de economicidade e baixo custo financeiro no desenvolvimento da esteira.

**Tabela 5** – Orçamento Financeiro de Custos do Projeto

<b>Produto</b>	<b>preço unitário</b>
<b>Arduino MEGA 2560</b>	50 reais
<b>Sensor Indutivo</b>	10 reais

Fonte: Próprio Autor

## 5.4 Propostas de Melhoria e Trabalhos Futuros

Durante a etapa de planejamento do presente trabalho, foram concebidas algumas ideias de aprimoramento do sistema de transporte implementado. No decorrer da montagem foram obtidas boas propostas de melhoria para incluir mais funcionalidades no sistema transportador.

### 5.4.1 Controle de Velocidade do Motor de Acionamento

A proposta de controle da velocidade do motor de acionamento da esteira confere ao projeto uma ampla gama de funcionalidades que poderiam ser implementadas no projeto. Com base nas características físicas dos objetos ou na quantidade de material transportado, seria necessário uma velocidade maior no processamento e possível seleção desses itens, dependendo da aplicação e dos requisitos de projeto definidos na fase planejamento de uma versão aprimorada do sistema de transporte desenvolvido no presente trabalho.

### 5.4.2 Inserção de um Braço Robótico Para Seleção de Itens

Nesta proposta, a ideia é que um braço robótico poderia ser instalado numa configuração modular, possibilitando que os itens transportados pudessem ser colocados sobre a correia sem intervenção humana. A integração entre dois braços robóticos confeccionados com o mesmo modelo de servomotor utilizado no presente trabalho poderia agregar mais precisão na disposição dos itens na esteira e uma melhor alocação do material transportado quando fosse identificado.

### 5.4.3 Instalação de Outros Sensores

A instalação de diversos tipos de sensores em conjunto com uma nova definição de parâmetros de controle poderia conferir mais praticidade e versatilidade no sistema de transporte e seleção. Uma esteira que selecionasse itens com base na temperatura, na cor ou no formato do objeto teria uma aplicabilidade diversa tanto no meio acadêmico como em uma micro empresa que necessite de um sistema de seleção e transporte com esses requisitos de funcionamento.

As possibilidades de melhoria do presente trabalho desenvolvido são diversas e abrem caminho para que mais projetos possam ser desenvolvidos utilizando-se a plataforma Arduino. A capacidade de aplicação prática desses sistemas transportadores no dia a dia mostra que as propostas de melhoria e novas ideias podem contribuir para que o conhecimento gerado seja difundido, beneficiando a sociedade de um modo geral, onde o conhecimento acadêmico em conjunto com a integração social pode resultar em parcerias e meios eficazes de apresentar os projetos e encontrar formas de beneficiar ainda mais a comunidade.

## Referências Bibliográficas

AGRAWAL NIKHIL; SINGHAL, S. [ieee 2015 international conference on computing, communication s automation (iccca) - greater noida, india (2015.5.15-2015.5.16)] international conference on computing, communication s automation - smart drip irrigation system using raspberry pi and arduino. 2015. Disponível em: <<http://gen.lib.rus.ec/scimag/index.php?s=10.1109/CCAA.2015.7148526>>. Citado 3 vezes nas páginas 15, 28 e 29.

BANZI MASSIMO; SHILOH, M. *Getting started with Arduino*. Third edition. Maker Media, Inc, 2015. (Make). ISBN 1449363334,978-1-4493-6333-8. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=2488be1179c5c4eab8315df96a11f467>>. Citado na página 22.

BARTMANN DANIELLE LAFARGE, P. C. E. *Le grand livre d'Arduino*. 2e édition. ed. Eyrolles, 2015. ISBN 2212141173,9782212141177. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=edcd95c5e6b4e594f6e4e35f5538e993>>. Citado na página 27.

BLEICHER, R. d. J. T. *SketchUp 2014 for Architectural Visualization*. Second edition. Packt Publishing, 2014. ISBN 9781783558414; 1783558415. Disponível em: <[libgen.li/file.php?md5=84ac922d5766eb6c4bb2adf26fcf1d53](http://libgen.li/file.php?md5=84ac922d5766eb6c4bb2adf26fcf1d53)>. Citado na página 31.

CONFERENCE, C. E. M. A. E. *Belt conveyors for bulk materials*. 5. ed. CEMA, 1997. ISBN 1891171186,9781891171185. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=F4266DA3C91AA20E7A68DBF8B8FE47E0>>. Citado na página 21.

EVANS JOSHUA NOBLE, J. H. M. *Arduino in Action*. Manning, 2013. ISBN 978-1-61729-024-4. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=75b52ff30c7d2d531e71034657a02881>>. Citado 6 vezes nas páginas 15, 22, 24, 26, 33 e 34.

HSIANG, K. K. W. L. M. S. R. M. H. K. J. W. H. W. Arduino-based automation of a dna extraction system. *Technology and Health Care*, v. 24, 12 2015. Disponível em: <<http://gen.lib.rus.ec/scimag/index.php?s=10.3233/THC-151048>>. Citado na página 29.

LAMB, F. *Automação Industrial na Prática*. McGraw Hill, 2015. (Serie Tekne). Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=57994654d39c2b0fc4706efe32819830>>. Citado 4 vezes nas páginas 15, 16, 20 e 21.

MARGOLIS, M. *Arduino Cookbook, 2nd Edition*. 2. ed. O'Reilly Media, 2012. ISBN 1449313876,9781449313876. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=4999C90AB9CA07843A1C5999252041AB>>. Citado 2 vezes nas páginas 15 e 25.

MCROBERTS, M. *Beginning Arduino, 2nd Edition*. Apress, 2013. ISBN 978-1-4302-5016-6. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=7d5f5d2b4b5b9e5cf75961498972431a>>. Citado 3 vezes nas páginas 23, 24 e 26.

PETRUZELLA, F. D. *Programmable Logic Controllers, 4th Edition*. 4th. ed. Career Education, 2010. ISBN 0073510882,9780073510880. Disponível em:

<<http://gen.lib.rus.ec/book/index.php?md5=2179ba845d4a87e8fa82cb07280006cc>>. Citado 2 vezes nas páginas 15 e 17.

PURDUM, J. *Beginning C for Arduino, Second Edition: Learn C Programming for the Arduino*. 2. ed. Apress, 2015. ISBN 1484209419,9781484209417. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=78e91d68931bcdfd8cc89424c43b2795>>. Citado 2 vezes nas páginas 15 e 23.

RANIEL, T. *Desenvolvimento e Implementação de um Sistema de Controle de Posição e Velocidade de uma Esteira Transportadora usando Inversor de Frequência e Microcontrolador*. Dissertação (Mestrado) — Universidade Estadual Paulista, 2011. Citado na página 16.

RILEY, M. *Programming Your Home: Automate with Arduino, Android, and Your Computer*. Pragmatic Bookshelf, 2012. (Pragmatic Programmers). ISBN 1934356905,9781934356906. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=3cf55ed902752b18dcf7fcaafb343ad1>>. Citado na página 27.

SANTOS, B. E. e Douglas Lopes Silva e E. Esteira para separação automática de material reciclado. *Revista de Gestão & Tecnologia*, v. 2, n. 1, 2014. ISSN 2358-3126. Disponível em: <<http://www.revista.unisal.br/lo/index.php/reget/article/view/139>>. Citado 2 vezes nas páginas 16 e 17.

SCHWARTZ, M. *Arduino Home Automation Projects : Automate your Home using the powerful Arduino Platform*. Packt Publishing, 2014. (Community Experience Distilled). ISBN 1783986069,9781783986064. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=f59b71e369ea0085a0d6c13875f14a96>>. Citado 2 vezes nas páginas 27 e 28.

SILVEIRA, P. R. da. *Automação e Controle Discreto*. [S.l.]: Érica Ltda, 1968. ISBN 8571945918. Citado na página 17.

SPINOLA, M. *Introdução à Automação*. Elsevier Acadêmico, 2011. (Em Portuguese do Brasil). ISBN 8535248897,9788535248890. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=7f80c78df0f1635cc7b11c5a20f1fc17>>. Citado 2 vezes nas páginas 15 e 16.

STACE, E. Y. L. *Belt Conveying of Minerals*. 1. ed. [s.n.], 2008. ISBN 9781420076066,142007606X,1845692306,9781845692308,9781845694302. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=26A24D3323CC310CA14B3458BBE3E136>>. Citado 3 vezes nas páginas 17, 18 e 19.

WARREN JOSH ADAMS, H. M. J.-D. *Arduino Robotics*. 1. ed. Apress, 2011. (Technology in Action). ISBN 1430231831,9781430231837. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=BD49797B7A36A513A26408FFBD312CC9>>. Citado 2 vezes nas páginas 22 e 26.

WENGER, M. O. J. F. A. *Arduino - physical computing fur Bastler, Designer und Geeks : [Microcontroller-Programmierung fur alle, Prototype your life, mit kompletter Programmiersprachenreferenz]*. 1. aufl. ed. O'Reilly, 2009. (O'Reilly basics). ISBN 9783897218932,3897218933. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=e8e213fca2fe80c84f04542abe1500f1>>. Citado na página 26.

---

XU JOYCE ZHANG, K. K. J. *Conveyor Belt Furnace Thermal Processing*. 1. ed. Springer International Publishing, 2018. ISBN 978-3-319-69729-1, 978-3-319-69730-7. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=85d7cee6f4b42907f56b686e4b973e2b>>. Nenhuma citação no texto.