

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES
CAMPUS PATOS DE MINAS

GUSTAVO TEIXEIRA GOMES

**PROTÓTIPO DE UM SELETOR AUTOMATIZADO
DE BLOCOS POR CORES**

Patos de Minas - MG
2024

GUSTAVO TEIXEIRA GOMES

**PROTÓTIPO DE UM SELETOR AUTOMATIZADO
DE BLOCOS POR CORES**

Projeto Final de Curso apresentado à banca examinadora como requisito parcial de avaliação da disciplina de PFC2 da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas.

Orientador: Prof. Dr. Daniel Costa Ramos

GUSTAVO TEIXEIRA GOMES

**PROTÓTIPO DE UM SELETOR AUTOMATIZADO
DE BLOCOS POR CORES**

Projeto Final de Curso apresentado à banca examinadora como requisito parcial de avaliação da disciplina de PFC2 da graduação em Engenharia Eletrônica e de Telecomunicações, da Faculdade de Engenharia Elétrica, da Universidade Federal de Uberlândia, Campus Patos de Minas.

Orientador: Prof. Dr. Daniel Costa Ramos

Patos de Minas, 22 de abril de 2024.

Prof. Dr. Daniel Costa Ramos (UFU/FEELT)

Prof. Dr. Davi Sabbag Roveri (UFU/FEELT)

Prof.^a Dr.^a Elise Saraiva (UFU/FEELT)

Dedico este trabalho aos meus pais e a toda minha família que sempre me deram forças, me apoiaram e acreditaram em mim durante esta jornada.

AGRADECIMENTOS

Primeiramente a Deus, que me deu força, coragem e resistência para completar esta etapa importante na minha vida.

À Universidade Federal de Uberlândia (UFU), todo o seu corpo docente que me proporcionou um grande aprendizado e me inspirou a sempre buscar mais conhecimento.

Ao Prof. Dr. Daniel Costa Ramos, pela orientação, paciência, apoio e confiança durante a elaboração deste trabalho.

À toda minha família que sempre me incentivou e apoiou durante o curso, em especial a minha mãe que apesar de não ter tido a mesma oportunidade, nunca mediu esforços para me proporcionar o meu crescimento por meio do estudo.

À minha avó e madrinha, que também foram muito importantes para formar a pessoa que sou hoje.

Por fim, também gostaria de agradecer a todos que diretamente ou indiretamente fizeram parte da minha formação.

RESUMO

A identificação de produtos pela sua coloração aumentou devido à grande demanda nos setores farmacêuticos e alimentícios. Além disso, a classificação manual de produtos dentro das indústrias se tornou inviável devido as limitações físicas e fisiológicas do ser humano quando se trata de um grande volume de produção. Desta forma, para evitar o gasto excessivo de tempo e energia, são utilizadas implementações robóticas automatizadas. Para exemplificar de forma didática este procedimento, este trabalho tem como propósito elaborar um protótipo de uma esteira seletora por cor automatizada, que permita separar blocos com colorações diferentes de forma eficiente, mais produtiva e com uma menor taxa de erro. Além disso, o projeto pode ser utilizado para demonstrar conceitos de eletrônica, programação e automação para outros estudantes. Para o protótipo foi utilizado um manipulador robótico para transferir os blocos do dispenser para a esteira, introduzindo uma abordagem diferenciada ao processo automatizado. Além disso, é usada uma esteira para movimentar os blocos, um sensor de cor TCS230, responsável por identificar a cor do bloco, seletores que funcionarão como atuadores para fazerem a separação dos blocos com diferentes colorações e um Arduino Mega, responsável por processar os dados recebidos pelo sensor, acionar os atuadores e controlar automaticamente o movimento do manipulador robótico. Os resultados indicam o correto funcionamento do protótipo, obtendo uma taxa de acerto de 97% na separação por cores.

Palavras-Chave: Separação por cor; TCS230; Manipulador Robótico; Esteira transportadora; Arduino; Seletores.

ABSTRACT

Product identification by color has increased due to high demand in pharmaceutical and food sectors. Additionally, manual sorting of products within industries has become unfeasible due to the physical and physiological limitations of humans when dealing with large production volumes. Thus, to avoid excessive time and energy expenditure, automated robotic implementations are utilized. To illustrate this procedure in a didactic manner, this work aims to develop a prototype of an automated color-selective conveyor belt, allowing for efficient separation of blocks with different colors, with higher productivity and a lower error rate. Furthermore, the project can be used to demonstrate electronics, programming, and automation concepts to other students. For the prototype, a robotic manipulator is used to transfer the blocks from the dispenser to the conveyor belt, introducing a differentiated approach to the automated process. Additionally, a conveyor belt is used to move the blocks, a TCS230 color sensor is employed to identify the block's color, selectors act as actuators to separate blocks with different colors, and an Arduino Mega is responsible for processing data received by the sensor, activating the actuators, and automatically controlling the movement of the robotic manipulator. The results indicate the proper functioning of the prototype, achieving a 97% accuracy rate in color separation.

Keywords: Color Sorting; TCS230; Robotic Manipulator; Conveyor Belt; Arduino; Selectors.

LISTA DE FIGURAS

Figura 1.1 – Design do protótipo de um manipulador robótico.	16
Figura 1.2 – Projeto de um manipulador robótico que separa componentes com colorações diferentes.	16
Figura 1.3 – Projeto utilizando esteira transportadora e seletores.	17
Figura 1.4 – Projeto e design usando esteiras transportadoras para separar objetos com colorações diferentes.	18
Figura 2.1 – Funcionamento da esteira transportadora.	21
Figura 2.2 – Manipulador robótico e suas partes.	22
Figura 2.3 – Garra de um manipulador robótico.	22
Figura 2.4 – Arduino Mega 2560.	24
Figura 2.5 – Ambiente de Desenvolvimento Integrado do Arduino.	24
Figura 2.6 – Sensor de cor TCS230 fabricado pela TAOS.	26
Figura 2.7 – Diagrama em blocos do funcionamento do TCS230.	27
Figura 2.8 – Micro Servo 9g SG90.	28
Figura 2.9 – Funcionamento do Micro Servo.	28
Figura 2.10 – Mini Ponte H L298N.	29
Figura 2.11 – Módulo PCA9685.	30
Figura 3.1 – Componentes principais para desenvolvimento do trabalho.	32
Figura 4.1 – Parte inferior do manipulador robótico (Vista superior e inferior).	36
Figura 4.2 – Base do manipulador encaixada no micro servo para realização de testes.	37
Figura 4.3 – Hastes lateral maior (Vista de ambos os lados).	37
Figura 4.4 – Hastes lateral menor (Vista de ambos os lados).	37
Figura 4.5 - Garra do manipulador robótico (Vistas laterais e inferior).	37
Figura 4.6 – Circuito para teste dos micros servos.	38
Figura 4.7 – Junção da base com a haste maior do manipulador robótico.	40
Figura 4.8 – Junção da haste menor no manipulador robótico.	41
Figura 4.9 – Manipulador robótico montado.	41
Figura 4.10 – Circuito com potenciômetros para controlar e analisar a posição dos servos motores.	42
Figura 4.11 – Leitura dos ângulos dos servos motores no manipulador robótico pelo monitor serial.	44
Figura 4.12 – Esteira Transportadora (Frente e verso).	45

Figura 4.13 – Circuito para funcionamento da Esteira Transportadora.	46
Figura 4.14 – Circuito para testes do sensor de cor TCS230 com Arduino Mega.	47
Figura 4.15 – Suporte e proteção do sensor de cor TCS230 projetados no software Autodesk Fusion.	51
Figura 4.16 – Case e haste para os servos motores dos seletores.	52
Figura 4.17 – Plataforma dos seletores (Vista frontal e traseira).	52
Figura 4.18 – Circuito para testes dos seletores.	53
Figura 4.19 – Dispenser montado e desmontado.	55
Figura 4.20 – Dispenser com blocos no projeto final.	55
Figura 4.21 – Circuito do projeto final.	56
Figura 5.1 – Manipulador robótico antes de pegar um bloco.	58
Figura 5.2 – Manipulador robótico agarrando um bloco.	58
Figura 5.3 – Manipulador robótico tirando um bloco do dispenser.	59
Figura 5.4 – Manipulador robótico levantando um bloco.	59
Figura 5.5 – Manipulador robótico direcionando um bloco para esteira.	59
Figura 5.6 – Manipulador robótico ajustando sua posição para depositar um bloco na esteira.	60
Figura 5.7 – Manipulador robótico aproximando da esteira.	60
Figura 5.8 – Manipulador robótico soltando um bloco na esteira.	60
Figura 5.9 – Bloco movimentando em direção do sensor de cor.	61
Figura 5.10 – Sensor de cor realizando a leitura da cor do bloco.	61
Figura 5.11 – Seletor para cor vermelha acionado e bloco deslocando em direção a primeira haste.	61
Figura 5.12 – Haste acionada com a esteira ligada direcionando bloco vermelho para primeiro recipiente.	62
Figura 5.13 – Bloco vermelho separado no primeiro recipiente.	62

LISTA DE TABELAS

Tabela 2.1 – Sentido de rotação para os pinos IN1 e IN2.....	30
Tabela 2.2 – Sentido de rotação para os pinos IN3 e IN4.....	30
Tabela 3.1 – Principais componentes e seus custos.....	32
Tabela 4.1 – Código usado para teste dos micros servos.....	39
Tabela 4.2 – Algoritmo utilizado para mapear os movimentos repetitivos do manipulador robótico.....	43
Tabela 4.3 – Código para funcionamento da esteira.....	46
Tabela 4.4 – Código para testar a leitura das cores do sensor de cor TCS230.....	48
Tabela 4.5 – Código para testar o funcionamento dos seletores.....	54
Tabela 5.1 – Resultado dos testes no protótipo.....	63

LISTA DE ABREVIATURAS E SIGLAS

DC	<i>Direct Current</i>
DIY	<i>Do It Yourself</i>
GLD	Grau de Liberdade
GND	<i>Ground</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
I2C	<i>Inter-Integrated Circuit</i>
LED	<i>Light Emitting Diode</i>
OE	<i>Output Enable</i>
PWM	<i>Pulse Width Modulation</i>
RGB	<i>Red, Green and Blue</i>
TAOS	<i>Texas Advanced Optoelectronic Solutions</i>
Vcc	Voltage at the Common Collector
3D	Três Dimensões

LISTA DE SÍMBOLOS

V	Volts
kHz	Kilohertz
ppm	Parts per Million
°C	Graus Celsius
A	Ampère
μF	Microfarad
Ω	Ohm
bps	Bits per second (baud rate)

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	TEMA DO PROJETO	13
1.2	PROBLEMATIZAÇÃO	13
1.3	HIPÓTESE.....	14
1.4	OBJETIVOS	14
1.4.1	Objetivo Geral.....	14
1.4.2	Objetivos Específicos	15
1.5	JUSTIFICATIVAS.....	15
1.6	CONSIDERAÇÕES FINAIS.....	19
2	REFERENCIAL TEÓRICO	20
2.1	ESTEIRAS TRANSPORTADORAS.....	20
2.2	MANIPULADOR ROBÓTICO	21
2.3	SELETORES.....	23
2.4	ARDUINO	23
2.5	SENSORES.....	25
2.5.1	Sensor de cor TCS230	25
2.6	MICRO SERVOS.....	27
2.7	MINI PONTE H L298N	29
2.8	MÓDULO PCA9685	30
2.9	CONSIDERAÇÕES FINAIS.....	31
3	MATERIAIS E MÉTODOS.....	32
3.1	MATERIAIS	32
3.2	MÉTODOS	33
4	DESENVOLVIMENTO	36
4.1	MANIPULADOR ROBÓTICO.....	36
4.2	ESTEIRA TRANSPORTADORA	45
4.3	SENSOR DE COR TCS230	47
4.4	SELETORES.....	52
4.5	DISPENSER	54
4.6	CONSIDERAÇÕES FINAIS.....	56
5	RESULTADOS	58
6	CONCLUSÃO.....	65

REFERÊNCIAS.....	66
ANEXO	70

1 INTRODUÇÃO

A tecnologia está cada vez mais integrada com a sociedade e as inovações tecnológicas são responsáveis por diversos impactos na mesma, como por exemplo, a utilização de máquinas na indústria e em trabalhos manuais (1). Nas indústrias, por exemplo, o monitoramento, detecção e identificação de produtos de forma automática se tornou algo indispensável para haver o controle da separação dos produtos inadequados dos adequados (2).

Alguns fatores importantes nas indústrias são a qualidade dos produtos, a eficiência, a produtividade, dentre outros. Quando a separação é feita pelo homem a qualidade dos produtos pode cair devido aos erros e ao cansaço inerente ao ser humano, fazendo com que a produtividade e eficiência também possam diminuir. Por outro lado, as máquinas são muito mais precisas, possuem uma eficiência maior e conseqüentemente são bem mais produtivas do que o trabalho humano nesta situação (1).

A automação é uma área do conhecimento atuante desde a 2ª Revolução Industrial e tem sido uma peça fundamental para o avanço tecnológico neste setor, pois além de conseguir realizar tarefas de forma automática, também fornece proteção para os seres humanos em áreas perigosas, melhora a produtividade e ajuda a ter menos erros e perdas devido à falta de atenção que às vezes os seres humanos cometem (3).

Dentre os equipamentos que compõem a automação em uma indústria, o seletor tem um papel importante de direcionar produtos com certas características para diferentes setores da indústria. Em alguns casos os sistemas rejeitam os materiais inadequados para fora da esteira e aceitam e direcionam os produtos adequados e semelhantes no mesmo recipiente apropriado, já em outros casos, alguns seletores também fazem a separação dos produtos. Este sistema é bom por gastar menos tempo, ter uma menor chance de erros e diminuição da mão de obra necessária (4).

Com o passar do tempo, a necessidade de identificar os produtos pela sua coloração foi aumentando devido a demanda crescente nos setores farmacêuticos e alimentícios. Isso ocorreu pois é importante garantir que no setor farmacêutico seja feita a separação dos encapsulamentos de comprimidos com a cor adequada e que não haja interferência na produção. Já no setor alimentício, a qualidade dos alimentos é, na maioria dos casos, avaliada e classificada, indicando se estão bons para o consumo de acordo com a sua coloração. Um exemplo seria o café, já que seu grau de maturação e seu produto final depende de sua cor (1).

As esteiras são amplamente utilizadas nas indústrias por possibilitar o deslocamento de produtos de forma ininterrupta de um lugar para outro (5). Além das esteiras serem elementos

de suma importância, também são consideradas a espinha dorsal da indústria, pois tornam o processo mais rápido e reduzem a mão de obra (6). Desde a criação até a classificação dos itens para empacotamento, o produto sempre permanece na esteira durante os procedimentos e continua em movimento constante (5).

Um manipulador robótico é um dispositivo programável que pode fazer diversas tarefas específicas em várias aplicações diferentes, sendo adepto às tecnologias digitais para execução das mesmas. As tarefas que demandam um maior esforço também podem ser feitas por um manipulador robótico, sendo assim muito útil nas indústrias. Para fazer aplicações diferentes basta alterar o programa no microcontrolador e substituir as garras para a aplicação que é desejada (7). Os manipuladores robóticos são muito usados em atividades que necessitam de uma maior precisão, execução de movimentos suaves e realização de tarefas muito repetitivas de manuseio de materiais (7, 8).

Um modelo de seleção é a utilização de um manipulador robótico para transferir blocos para a esteira, que por sua vez fará o deslocamento dos produtos até o equipamento de seleção e assim ocorrerá a separação dos devidos blocos, que podem ser identificados por diferentes colorações por meio do sensor de cor. Desta forma é possível separar automaticamente os produtos defeituosos ou inadequados dos demais.

1.1 TEMA DO PROJETO

Ao fazer a junção do manipulador robótico com a esteira e os seletores, surge um seletor automatizado de blocos por cores, que além de ser um sistema que permite separar automaticamente produtos defeituosos ou inadequados, também possui todas as características positivas da esteira e do manipulador robótico integradas, sendo elas uma maior fidelidade e produtividade das mercadorias.

O tema deste projeto é implementar um protótipo de um processo de seleção de blocos, utilizando um manipulador robótico para transferi-los do dispenser até uma esteira que fará o deslocamento destes blocos até o equipamento de seleção, além de um sensor de cor para identificá-los e seletores que realizarão a separação dos blocos com colorações diferentes. Sendo que a separação das cores será somente para as cores vermelho e azul.

1.2 PROBLEMATIZAÇÃO

Os alimentos em uma indústria passam por diversos procedimentos, testes e monitoramentos para que a qualidade seja avaliada e classificada se está bom para o consumo,

em alguns casos esta qualidade pode ser aferida por meio de sua coloração. No caso do café, seu grau de maturação e produto final dependerá de sua cor para que o café seja de boa qualidade. Desta forma, na etapa em que é feita a separação do café, é possível aplicar um seletor de cor para resolver este problema. Além disso, o seletor pode ser aplicado para separar limões e diversos outros alimentos ao fazer a programação do mesmo (1).

Outras aplicações interessantes dos seletores são no setor de recicláveis, que possibilita separar materiais diferentes para fazer a reciclagem, e no setor de farmacêuticos que faz a separação dos encapsulamentos de comprimidos com a cor adequada para que não haja interferência na produção, sendo que ambos possuem uma maior precisão e eficiência do que o trabalho do homem (1).

Como exemplificado acima, o processo de seleção na indústria é muito importante e existe uma grande demanda por profissionais que compreendam o processo e os equipamentos envolvidos. Uma forma de apresentar este conhecimento e atuar na formação de profissionais mais qualificados é por meio da metodologia de “Aprender na Prática”, também denominada *learning by doing*. Esta consiste em uma proposta que tem como ponto central a inserção de situações hipotéticas, mas que estão relacionadas à pontos reais do processo de ensino-aprendizagem do aluno.

A utilização de protótipos configuráveis é uma maneira de forçar os alunos não só a visualizar os conceitos e o problema a ser resolvido, mas também de atuar e aprender resolvendo o problema a ser apresentado. No entanto, o alto custo, e muitas vezes a impossibilidade de alteração de um protótipo comercial, inviabiliza este tipo de metodologia. Dessa forma desenvolver tecnologias e kits que visam esta abordagem irá mitigar este problema e possibilitará aos alunos desenvolver mais as suas capacidades e conhecimentos.

1.3 HIPÓTESE

A hipótese central deste trabalho é testar se é possível desenvolver um protótipo didático de um processo de seleção de blocos, mas de baixo custo e modular, com fins educacionais de demonstração de conhecimentos, conceitos e resolução de problemas.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

O objetivo geral é projetar, implementar e construir um protótipo que utiliza um manipulador robótico para transferir blocos até uma esteira seletora, capaz de separar

automaticamente os blocos com base na sua coloração, sob o ponto de vista educacional, visando demonstração de conceitos de automação, eletrônica e programação.

1.4.2 Objetivos Específicos

Para atingir o objetivo geral é preciso definir os objetivos específicos que servirão como metas a serem atingidas que no final resultará no objetivo geral, desta forma eles são descritos abaixo:

- Implementar a transferência de blocos por meio de um manipulador robótico para esteira.
- Construir uma esteira com deslocamento de pequenos blocos;
- Planejar e implementar o processo de seleção dos blocos, com a utilização de micro servos;
- Avaliar o desempenho do protótipo.

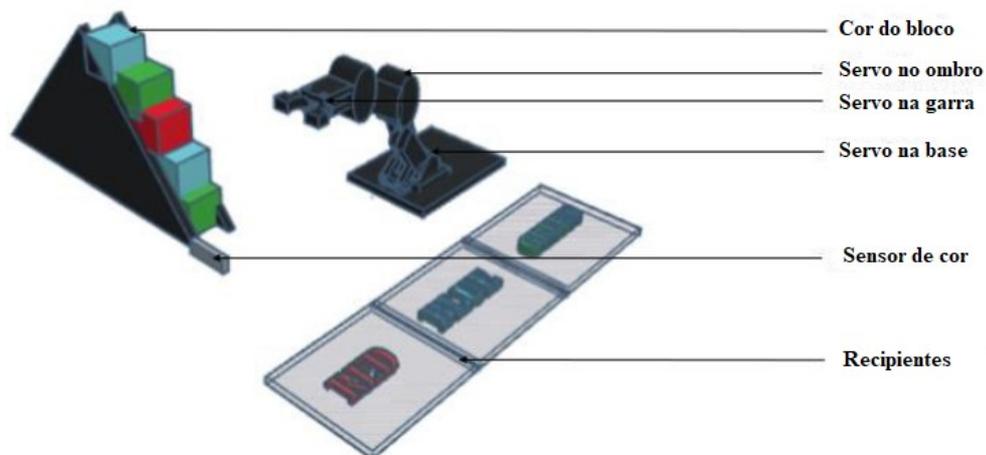
1.5 JUSTIFICATIVAS

Existem diversos trabalhos nos últimos anos que desenvolveram protótipos de equipamentos seletores e que são utilizados como pilares para a construção desta proposta. Desta forma, para justificar o tema deste projeto serão apresentados trabalhos que desenvolveram este tema e, portanto, tem uma relevância no estudo e no aprofundamento do assunto para fazer possíveis melhorias nos protótipos.

O trabalho desenvolvido por KHAIRUDIN et al. (2021), tem como objetivo projetar um manipulador robótico articulado que possa realizar uma tarefa repetitiva simples de separar blocos com colorações diferentes. No artigo foi usado uma pinça com almofadas de pressão para pegar e colocar blocos de 0,1kg em um local designado. No projeto também foi utilizado um sensor de cor TCS3200, quatro motores micro servo MG90S para movimentação do manipulador robótico, um módulo ESP8266 WiFi e o ATMEGA328P-Arduino IDE como microcontrolador principal (7). Para melhor entendimento, a Figura 1.1 ilustra o *design* deste projeto.

O projeto foca na movimentação suave do manipulador e na precisão da detecção de cores. Além disso, os dados do manipulador foram comunicados sem fio e monitorados remotamente usando a plataforma *ThingSpeak* como parte de um aplicativo *Internet of Things* (IoT). Este trabalho obteve sucesso separando os blocos com colorações diferentes mesmo que o ambiente esteja muito escuro ou mesmo que a temperatura da cor esteja mudando, além disso o robô conseguiu detectar, classificar e enviar os dados dos itens com precisão (7).

Figura 1.1 – Design do protótipo de um manipulador robótico.



Fonte: Adaptada de (7).

No artigo de PANIE et al. (2018) foi feito um projeto para reduzir os erros do ser humano no cálculo e posicionamento de mercadorias. Para solucionar este problema foi utilizado o sensor TCS3200 que pode identificar a cor das mercadorias com base na cor que foi ajustada, também foi usado um microcontrolador NodeMCU e quatro micros servos no manipulador robótico (9), conforme indicado na Figura 1.2.

Figura 1.2 – Projeto de um manipulador robótico que separa componentes com colorações diferentes.



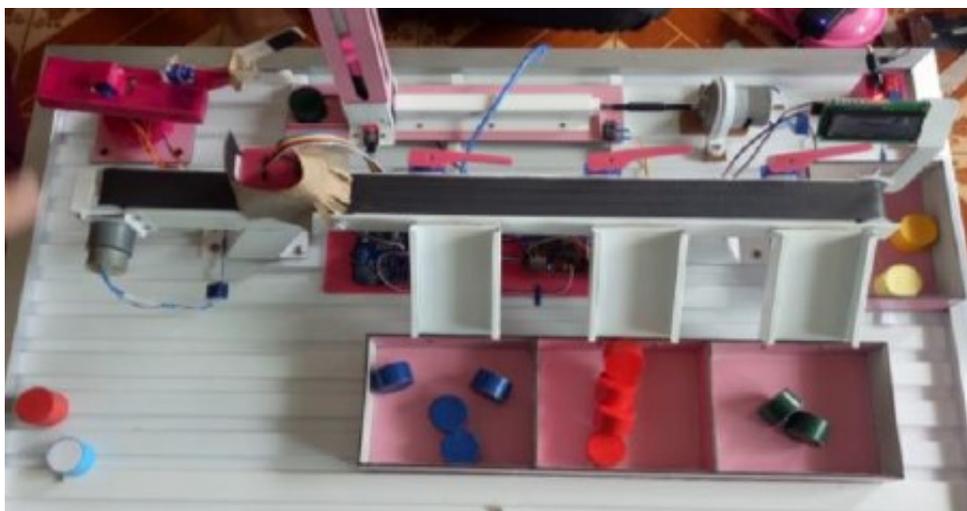
Fonte: (9).

Durante a execução do processo, o microcontrolador processa os dados de entrada do sensor na forma de um valor de cor para identificar o tipo de mercadoria e em seguida o manipulador robótico separa as mercadorias pelas cores e por fim os dados são enviados para um site. O autor obteve sucesso fazendo a separação das mercadorias baseadas nas cores e foi

reconhecido e apoiado pela *Faculty of Computer Science and Information Technology in Gunadarma University, Jakarta, Indonesia* (9).

Por outro lado, o artigo de THIKE et al. (2019) foi desenvolvido devido à produtividade e fidelidade que as máquinas executam tarefas altamente repetitivas melhor que o homem. A finalidade do projeto é separar as diferentes cores dos objetos com eficiência e com erros mínimos. Para fazer isso é preciso uma esteira, um sensor de cor TCS230 que pode medir as três cores primárias *Red, Green and Blue* (RGB) e também tem um detector de luz branca separado, o Arduino UNO e um mini manipulador robótico acionado por três servo motores (4). Para entender o seu funcionamento, a Figura 1.3 ilustra este projeto.

Figura 1.3 – Projeto utilizando esteira transportadora e seletores.



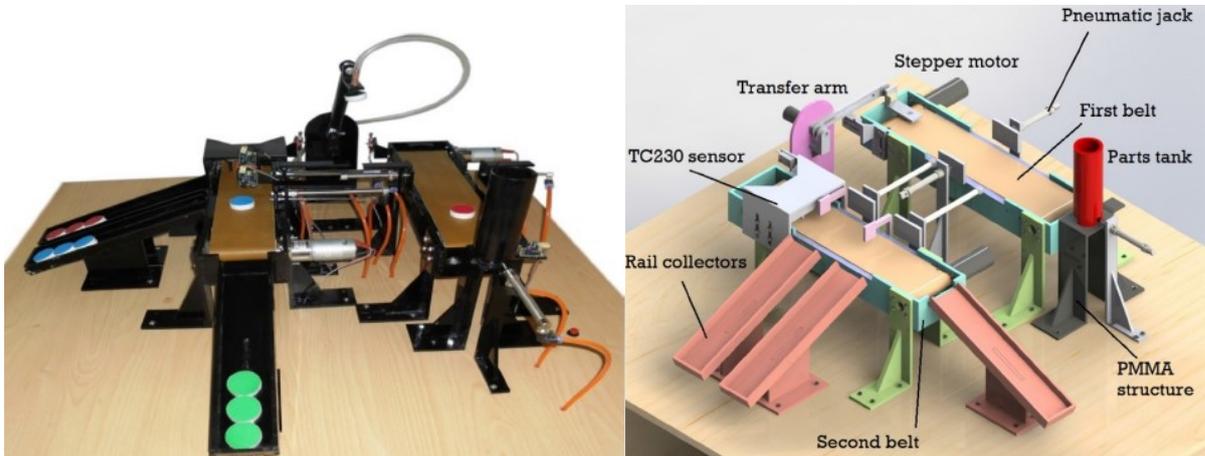
Fonte: (4).

No projeto é feita a separação das cores primárias (vermelho, verde e azul) e também separa as cores secundárias (junção de duas cores primárias) das primárias. Caso for uma das cores primárias a separação ocorre primeiramente quando o sensor identifica a cor do objeto e então o mesmo é empurrado para fora da esteira por um dos três seletores, caso contrário continua na esteira até ser separado das cores primárias. O projeto obteve sucesso e os autores concluíram que o tempo e o esforço humano podem ser reduzidos com a implementação do projeto nas indústrias química, de alimentos, fabricação de chips e dentre outras (4).

Por fim, o artigo de DEZAKI et al. (2022), projeta e desenvolve um sistema de transporte mecatrônico automatizado e inteligente que detecta três cores (vermelho, verde e azul) diferentes de objetos circulares e em seguida é feita a separação das mesmas em recipientes diferentes. O principal objetivo do projeto é desenvolver um novo projeto estrutural de três dimensões (3D) que possua duas esteiras de forma a tornar o procedimento único para alcançar o mais alto nível de precisão e eficiência com um custo mínimo. Este trabalho é

interessante pois ele pode ser usado nas indústrias de fabricação e de embalagem, já que o projeto permite transferir produtos em locais tóxicos e rigorosos que demandam uma baixa taxa de erros (10). Para melhor entendimento, a Figura 1.4 ilustra este projeto e o seu *design*.

Figura 1.4 – Projeto e design usando esteiras transportadoras para separar objetos com colorações diferentes.



Fonte: (10).

O trabalho possui algumas diferenças em relação aos outros por possuir duas esteiras que são separadas uma da outra para fazer a simulação de duas linhas separadas e um seletor pneumático. Foi usado, além das duas esteiras, um sensor de cor TCS230, três coletores, um tanque de peças, um braço de transferência de peças com seu respectivo motor, o movimento das esteiras é fornecido por motores Zheng DC. Este projeto começa distribuindo as peças na primeira esteira e em seguida, ela é posicionada no local correto para ser posteriormente ser coletada pelo braço de transferência e ir para a segunda esteira. Depois disso, o sensor de cor detecta a cor da peça e por fim a peça é despejada no local correspondente a sua cor. O trabalho obteve sucesso, pois depois dos testes foi concluído que o projeto é adequado para detecção e classificação de cores, trazendo assim os melhores resultados por ser eficiente e com uma baixa taxa de erros (10).

Em síntese, os trabalhos citados acima mostram a relevância do assunto e que já está sendo desenvolvido diversas pesquisas de protótipos e projetos a respeito do tema selecionado. Consequentemente, este trabalho será baseado nestes projetos apresentados, mas com algumas adaptações. Será usado um manipulador robótico junto com uma esteira seletora para integrar funcionalidades de dois projetos diferentes. Desta forma, por exemplo, é possível usar um manipulador robótico para mover as peças para uma esteira, com o intuito de aumentar o nível de precisão. Além disso, sob o ponto de vista educacional será mais didático fazer esta combinação, por haver uma maior exploração de recursos e conhecimentos, já que será usado

o manipulador robótico para transferir os blocos do dispenser para a esteira, que por sua vez fará a separação pela cor.

Esta mesclagem de trabalhos também é interessante por poder trazer mais benefícios no futuro ao abordar mais assuntos usando este kit educacional. Desta forma, será usado um manipulador robótico baseado nos projetos de KHAIRUDIN et al. (2021) e de PANIE et al. (2018), que tem como função transferir as peças de um local para outro. A esteira transportadora será baseada no trabalho de DEZAKI et al. (2022) por possuir o mínimo de seletores que separam as cores primárias e também a esteira será baseada no projeto de THIKE et al. (2019), que utiliza três seletores com micro servos que separam os objetos por cores.

1.6 CONSIDERAÇÕES FINAIS

Uma vez que o propósito deste trabalho foi definido, bem como seu contexto e motivações, no capítulo a seguir, serão apresentadas algumas referências para este trabalho. Os próximos segmentos incluirão a discussão das principais referências teóricas que fundamentam esta pesquisa, seguida pela descrição dos materiais e métodos adotados para o desenvolvimento do protótipo. Posteriormente, será abordado o processo de desenvolvimento do trabalho, seguido pela apresentação e discussão dos resultados obtidos. Finalmente, serão apresentadas as conclusões, destacando-se a importância e contribuições do estudo, juntamente com sugestões para pesquisas futuras.

2 REFERENCIAL TEÓRICO

Nesta seção serão abordadas as teorias utilizadas durante o trabalho, consistindo em esteiras transportadoras, manipulador robótico, seletores, sensores de cor e o Arduino.

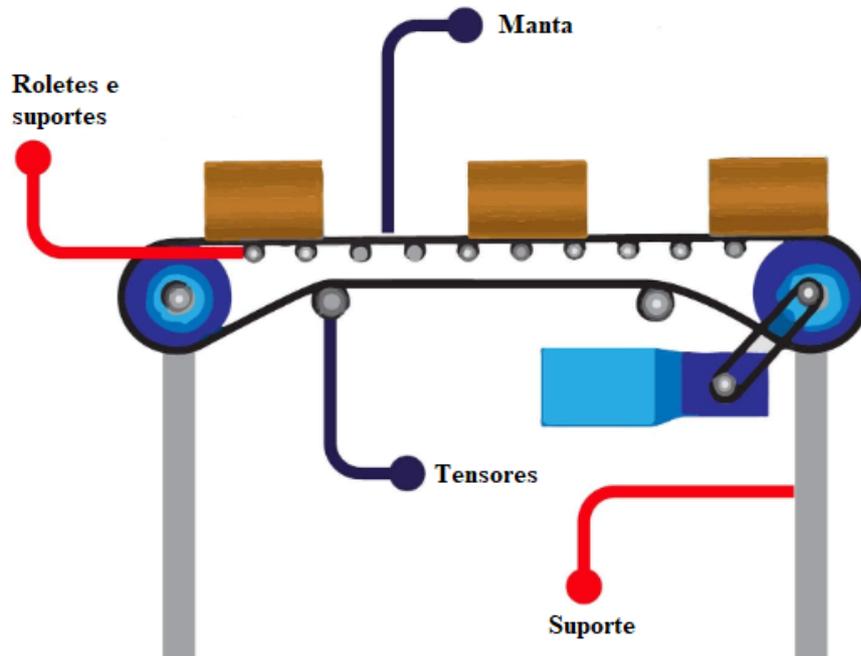
2.1 ESTEIRAS TRANSPORTADORAS

Com seu surgimento em 1919, as esteiras transportadoras se tornaram fundamentais por garantir um transporte ágil e seguro de produtos e insumos em uma indústria, sendo que muitas operações dependem delas para que funcione todo o processo industrial. As esteiras podem ser aplicadas em várias indústrias, sendo elas alimentícias, farmacêuticas, automobilísticas e dentre outras. Sua principal função é reduzir o tempo e o custo nas fábricas, já que haverá uma produção mais eficiente. Uma característica interessante para este projeto é que elas podem ser automatizadas (11, 12, 13).

Existem diferentes tipos de esteiras transportadoras, podendo ser feitas de mantas ou roletes, retas ou curvas e dentre outras classificações. Na composição das esteiras industriais, os materiais utilizados dependem da carga a ser transportada. Para realizar o transporte de materiais menores, como por exemplo grãos de café, farelos e comprimidos, é necessária uma esteira que possua uma manta. Na maioria dos casos é feito em borracha por possuir alta resistência a componentes químicos e por serem mais fáceis de limpar. Elas possuem vantagem por viabilizar o transporte de qualquer tamanho e tipo de produtos, até mesmo materiais úmidos e líquidos (11, 14).

Uma esteira transportadora possui no mínimo duas polias, uma correia, um suporte e a manta. Seu funcionamento começa primeiramente com as duas polias movimentando a esteira transportadora. Porém, para acontecer este movimento, o motor já em funcionamento fará com que a correia gire uma das polias, fazendo com que posteriormente aconteça o movimento da manta (14). Também é importante entender que para sustentar o peso da carga é possível usar roletes que funcionam como apoio para a carga ou é possível usar tensores que apertam a manta para que a esteira fique esticada para evitar que a carga afunde a manta (15). Este funcionamento está ilustrado conforme a Figura 2.1.

Figura 2.1 – Funcionamento da esteira transportadora.



Fonte: Adaptado de (15).

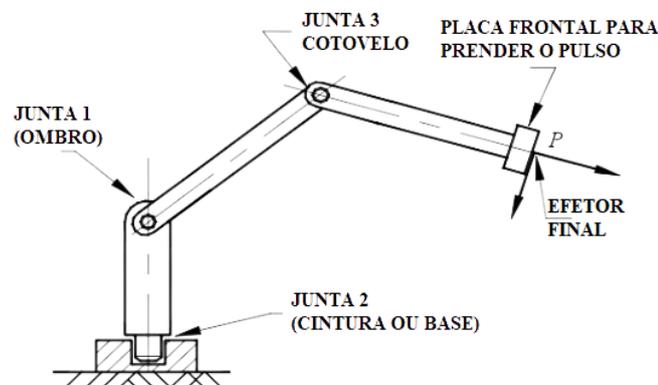
2.2 MANIPULADOR ROBÓTICO

As indústrias estão cada vez mais procurando automatizar os processos de fabricação e dessa forma, os manipuladores robóticos se tornam cada vez mais demandados por serem capazes de realizarem tarefas de forma mais eficiente do que o ser humano. Além disso, eles fornecem uma padronização do produto, uma melhoria na qualidade do produto final, uma maior precisão e conseguem realizar um trabalho contínuo, aumentando a produtividade da empresa (19). Um manipulador robótico é um aparelho programável que busca ter os mesmos comportamentos e funcionalidades de um braço humano, possibilitando ser usado para diversas funções e para vários tipos de processos automatizados (19, 20).

Os manipuladores robóticos possuem diferentes partes que são unidas e conectadas de forma a possibilitar movimentos de translação e rotação (20). Além disso, são formados e classificados por uma quantidade de articulações que determinam o Grau de Liberdade (GLD) do manipulador robótico, que é um fator importante que define seu espaço de trabalho (21). Estas articulações podem ser divididas em juntas rotacionais, que permite grau de liberdade rotacional e juntas prismáticas que permitem grau de liberdade linear (22). A geometria do manipulador e as limitações físicas das juntas são alguns exemplos de limitações no espaço de trabalho devido à quantidade de GLD (21).

Em um manipulador robótico comum, geralmente existem quatro partes principais, sendo elas a base, o ombro, o cotovelo e a garra. Sendo que cada uma dessas partes possui uma função específica. A base ou também conhecida como cintura, possibilita rotacionar o manipulador em 180° ou até mesmo 360° , sendo que este valor é definido pelo tipo de servo motor que é usado. O ombro tem como função levantar ou abaixar o manipulador no eixo vertical, enquanto que o cotovelo move-o para frente ou para trás. Por fim, a garra trabalha abrindo ou fechando para segurar algum bloco (23). Para entender melhor, a Figura 2.2 ilustra um manipulador robótico e suas partes identificadas separadamente e na Figura 2.3 é ilustrada uma garra de um manipulador robótico para melhor compreensão de seu funcionamento.

Figura 2.2 – Manipulador robótico e suas partes.



Fonte: Adaptada de (24).

Figura 2.3 – Garra de um manipulador robótico.



Fonte: (25).

Conforme ilustra a Figura 2.2, na extremidade do manipulador, é possível prender um pulso no efetuador final que pode ser uma garra, pinça, imã ou até mesmo uma ventosa, conforme a necessidade da aplicação (20). Neste projeto, inicialmente será utilizado no pulso uma garra, que como descrito anteriormente, permitindo abrir e fechar para segurar blocos. Para o movimento desta garra, é necessário um servo motor.

2.3 SELETORES

Para que seja feita a separação dos produtos de uma maneira mais eficiente em relação ao manipulador robótico, é recomendável o uso de seletores. Isso ocorre devido ao fato que o manipulador leva um tempo para agarrar e soltar um determinado produto dentro de um processo de seleção, enquanto isso o seletor, consegue separar mais rapidamente aproveitando o movimento natural da esteira. Os tipos de seletores mais comuns são seletores pneumáticos, hidráulicos e com micros servos. Os pneumáticos e hidráulicos geralmente possuem como função empurrar um bloco para fazer a separação. Já o seletor com micro servo, pode ser configurado com uma haste física fixada no micro servo, de forma que quando acionado o mesmo, o servo movimenta a haste para um lado, direcionando o bloco para um recipiente fazendo então a separação dos blocos (4, 10).

Existem diversas outras formas de separar blocos com características diferentes, seja pelo peso, tamanho, coloração e dentre outros, basta utilizar um método de seleção apropriado em um projeto. Por exemplo, o processo de ventilação possibilita separar componentes mais densos de componentes menos densos, enquanto isso usando um eletroímã é possível fazer a separação de materiais metálicos dos demais materiais. Dessa forma, neste projeto será utilizado um seletor com micro servo baseado no artigo de THIKE et al. (2019) por ser mais simples e atender as necessidades do projeto, para que seja feita a separação dos blocos com colorações diferentes (4).

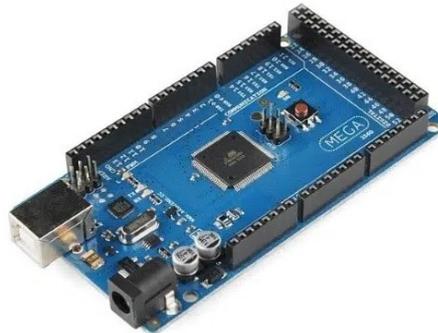
2.4 ARDUINO

O Arduino é uma plataforma de prototipagem eletrônica muito usada por profissionais de várias áreas, estudantes e até mesmo como hobby. Com ele, é possível a criação de diversos projetos de eletrônica, de nível simples até aplicações de níveis intermediárias, como por exemplo para alarmes, IoT, robôs, sistemas de automação residencial ou industrial e dentre outros. Desta forma, o principal objetivo do Arduino é facilitar o contato com esta plataforma por ser multifuncional e acessível. Sendo atrativo para vários trabalhos, inclusive para este, por ser baseado no conceito de *open source*, ou código aberto. Ademais, também é permitido adaptações para expansão de projetos, podendo acrescentar, por exemplo, sensores e atuadores (26, 27).

O Arduino surgiu em 2005 na cidade de Ivrea, na Itália, por meio de um projeto educacional (27, 28). Ele possui um microprocessador e demais componentes que fazem o mesmo funcionar como um microcontrolador. A base do Arduino é justamente a ideia do “faça

você mesmo”, também conhecido pela sigla em inglês *Do It Yourself* (DIY) (27). Neste projeto será usado o Arduino Mega 2560, por possuir o protocolo de comunicação *Inter-Integrated Circuit* (I2C) e mais portas de entrada e saída de dados em comparação com outras placas Arduino, de forma a atender as necessidades de um projeto mais robusto, conforme ilustra a Figura 2.4.

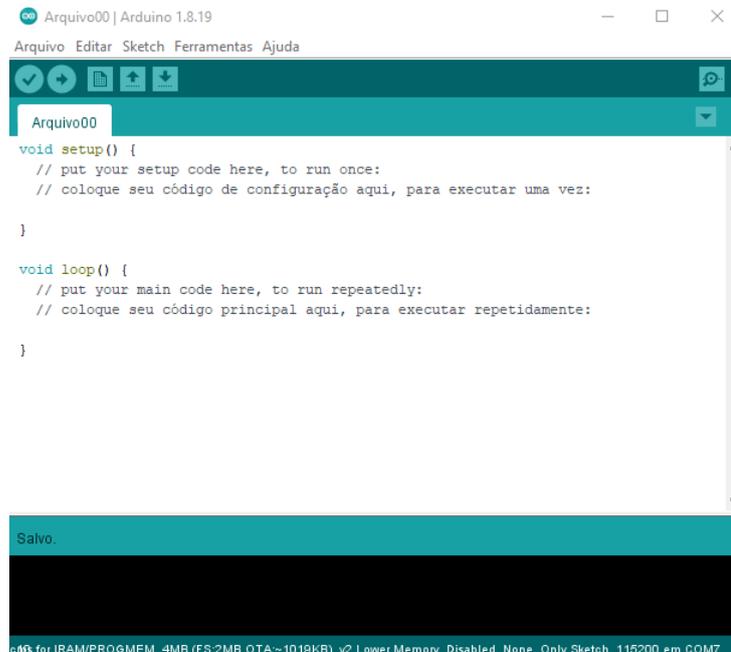
Figura 2.4 – Arduino Mega 2560.



Fonte: (26).

Para a realização deste trabalho, o *software* que será utilizado será o *Arduino Integrated Development Environment* (IDE), que usa como base a linguagem de programação C/C++. (26, 29). A Figura 2.5 ilustra como é a interface do ambiente de desenvolvimento integrado do Arduino.

Figura 2.5 – Ambiente de Desenvolvimento Integrado do Arduino.



Fonte: O autor.

2.5 SENSORES

Os sensores e atuadores em conjunto com um microcontrolador são fundamentais para a automação industrial. De uma maneira geral, os sensores são dispositivos que têm como funções obter as variáveis físicas de um ambiente que está sujeito a um monitoramento e enviar essas informações para um microcontrolador. Estas variáveis físicas podem ser, por exemplo, a temperatura, a umidade, a pressão, a luz e dentre outras. Quando a informação chegar no microcontrolador, um atuador do circuito eletrônico pode ser acionado quando ocorrer algum evento e controlar uma determinada variável (30).

Um sensor possui um sinal de entrada e um sinal de saída, sendo que o sinal de entrada é uma grandeza física ou uma forma de energia do ambiente que será medida, como por exemplo, a energia térmica, a umidade do ar, a luz e dentre outros. Já o sinal de saída é dado no domínio elétrico (uma tensão ou corrente), podendo ser classificado em analógico ou digital. Um sensor com uma saída analógica possui como características a capacidade de assumir qualquer valor dentro de uma faixa de operação, em contrapartida o sensor com uma saída digital só pode assumir valores como zero ou um (30, 31).

Desta forma, um sensor nada mais é do que um dispositivo que obtém uma variável física, converte a mesma em um sinal elétrico e transmite uma informação que por sua vez poderá ser exibida ou controlada por um possível atuador após ter passado por um determinado processamento (31). Os sensores possuem diversas aplicações, podendo ser aplicados junto com um atuador no sistema de estacionamento dos carros mais modernos, em relés de proteção contra o excesso de corrente ou de tensão em um circuito, em sistemas de segurança residencial, no mouse do seu computador, na tela do seu celular e dentre outras aplicações (32). Neste trabalho, para fazer a identificação e separação das cores de forma automática pelos seletores, será necessário o uso de um sensor de cor. Compreendido o funcionamento geral dos sensores, agora será aprofundado mais a respeito dos sensores de cor.

2.5.1 Sensor de cor TCS230

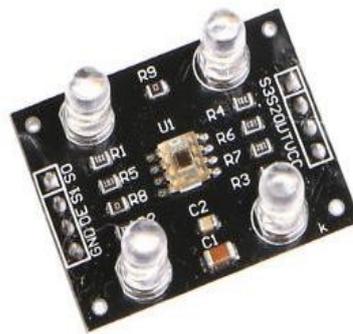
Um sensor de cor é um dispositivo com capacidade de detectar a intensidade da luz, identificar as três cores primárias e enviar esta informação para um microcontrolador. O sensor funciona primeiramente emitindo uma luz de um transmissor e posteriormente o receptor detecta a intensidade da luz refletida do bloco observado nas cores vermelho, verde e azul. Sendo que a distinção das cores no receptor é feita pelos comprimentos de onda ou frequências dessas cores primárias e pela intensidade de cada cor identificada. É importante entender que

este sensor descreve uma cor em termos numéricos para cada uma das cores RGB. Eles são muito aplicados nas indústrias automotivas, alimentícias, farmacêuticas e dentre outras (33, 34).

O sensor utilizado neste projeto foi o TCS230, por conseguir detectar várias cores visíveis. Além disso, ele também possui uma frequência de saída programável, uma alta resolução de conversão da intensidade da luz para frequência, consegue comunicar diretamente com um microcontrolador. Ele opera na tensão de 2,7V a 5,5V, possui uma função de *Power Down*, uma margem de erro típica de 0,2% a 50 kHz e um coeficiente de temperatura estável de 200 ppm / °C.

Esse sensor foi escolhido por ser uma opção mais acessível, com características interessantes e que atendem as demandas do projeto. A Figura 2.6 mostra o sensor de cor TCS230 fabricado pela *Texas Advanced Optoelectronic Solutions* (TAOS) (35).

Figura 2.6 – Sensor de cor TCS230 fabricado pela TAOS.

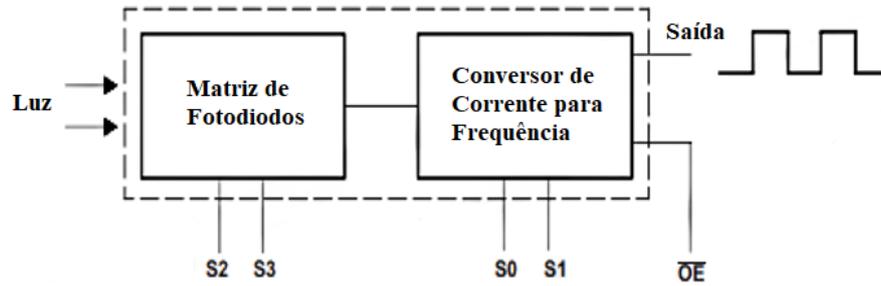


Fonte: (4).

Nota-se que existem quatro *Light Emitting Diode* (LED) e seu funcionamento começa pela emissão de luz destes LEDs, que iluminam um bloco e posteriormente a intensidade de luz refletida nas cores RGB são detectadas pelo receptor, que é uma matriz 8x8 de fotodiodos de silício. Dentre os 64 fotodiodos desta matriz, 16 deles possuem filtros vermelhos, 16 possuem filtros verdes, 16 possuem filtros azuis e 16 não possuem filtros. Estas divisões dos fotodiodos são entrelaçadas para minimizar o efeito da não uniformidade da irradiância incidente. Sendo que cada um dos 16 fotodiodos com o mesmo filtro foi conectado em paralelo (1, 35).

A saída do TCS230 é de uma onda quadrada com um ciclo de trabalho de 50%, sendo que a frequência é diretamente proporcional à intensidade da luz (irradiância). A entrada e saída deste sensor é classificada como sendo digital, por este motivo é possível ter uma interface direta com um microcontrolador. Para entender melhor o funcionamento do TCS230, observe o diagrama em blocos Figura 2.7 (1, 35).

Figura 2.7 – Diagrama em blocos do funcionamento do TCS230.



Fonte: Adaptada de (35).

Conforme mostra a Figura 2.7, a luz refletida do bloco passa primeiramente pelo bloco da matriz de fotodiodos e em seguida passa pelo bloco do conversor de corrente para frequência. Note que há dois pinos no bloco da matriz de fotodiodos (S2 e S3) que possuem a função de selecionar o filtro de cor a ser lido. Observe também que no bloco do conversor de corrente para frequência também há dois pinos (S0 e S1), eles são os pinos de controle, que possibilitam configurar a escala de frequência na saída para 2%, 20% ou 100%. Ademais, na saída deste bloco existe o pino de *Output*, ou de saída, que promove o envio das informações de saída e o pino de *Output Enable* (OE) viabiliza o interrompimento do transdutor do sensor para aumentar a economia e diminuir o gasto de energia. Enquanto este pino estiver desligado, seu estado será de alta impedância de modo a permitir o compartilhamento de várias unidades de uma linha de entrada do microcontrolador (1, 35, 36).

2.6 MICRO SERVOS

Os micros servos são dispositivos eletrônicos compactos e versáteis amplamente empregados em diversas aplicações, destacando-se na robótica e automação. Eles são controlados por um sinal *Pulse Width Modulation* (PWM), o qual determina a posição do eixo do servo dentro de sua faixa de operação. Internamente, consistem em um motor elétrico, uma caixa de redução de engrenagens, um potenciômetro e um circuito de controle eletrônico, que trabalham em conjunto para mover o eixo do servo com precisão para a posição desejada. Essa precisão no controle de movimentos os torna ideais para uma variedade de aplicações (38). A Figura 2.8 ilustra um micro servo com seus componentes principais.

Figura 2.8 – Micro Servo 9g SG90.

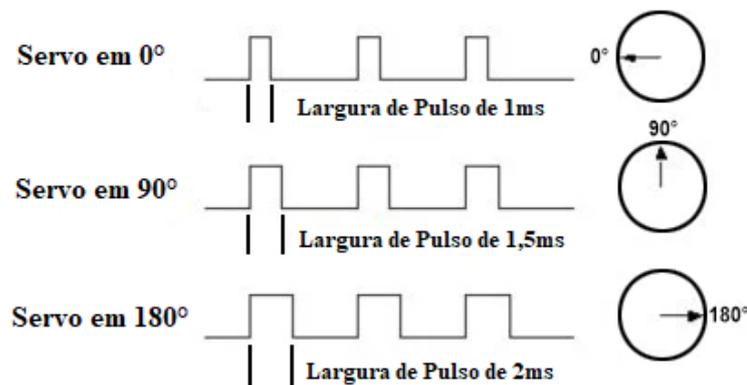


Fonte: (37).

Esses dispositivos são projetados para proporcionar controle preciso de posição em uma saída mecânica, permitindo rotações dentro de um intervalo específico de ângulos, comumente variando de 0 a 180 graus. Além disso, seu tamanho compacto os torna ideais para aplicações em espaços reduzidos, enquanto seu baixo consumo de energia os torna adequados para uso em dispositivos portáteis e alimentados por bateria (38).

O micro servo ajusta sua posição conforme o sinal PWM de controle, geralmente operando a uma frequência de 50Hz, o que equivale a um período de 20ms. Durante esse período, o circuito de controle monitora o sinal e determina a posição do micro servo com base na largura do pulso PWM. Por exemplo, um pulso de 1ms corresponde a um ângulo de 0°, 1,5ms a 90° e 2ms a 180°. Assim, qualquer alteração na largura do pulso resulta em uma alteração proporcional na posição do eixo do servo (38). Esse processo é ilustrado com mais detalhes na Figura 2.9.

Figura 2.9 – Funcionamento do Micro Servo.



Fonte: Adaptada de (38).

Um micro servo é composto por três fios distintos: o vermelho, responsável pela conexão à fonte de alimentação, geralmente operando entre 3,0V e 7,2V; o marrom, que serve como terra, ligado ao terminal negativo da fonte de energia; e o fio amarelo, destinado ao controle do servo por meio de um sinal PWM, determinando sua posição. Embora as cores dos fios possam variar entre fabricantes, esses padrões de fiação são comuns. No projeto, os micros

servos foram empregados tanto para movimentar o manipulador robótico, abrangendo base, ombro, cotovelo e garra, quanto para acionar os seletores.

2.7 MINI PONTE H L298N

Uma mini ponte H L298N, assim como a ponte H L298N, é um dispositivo eletrônico projetado para controlar motores *Direct Current* (DC) bidirecionais ou um motor de passo. Seu nome é baseado no chip controlador de motor H-bridge L298N. A mini ponte H L298N opera de forma semelhante à ponte H padrão. Além disso, com ela é possível controlar dois motores simultaneamente (40). A Figura 2.10 apresenta uma ilustração da mini ponte H.

Figura 2.10 – Mini Ponte H L298N.



Fonte: (39).

A mini ponte H L298N requer uma fonte de alimentação externa para operar, que pode ser fornecida por uma bateria ou outra fonte de energia compatível com os motores que estão sendo controlados. Sua tensão de operação está entre 2V e 10V. No entanto, é fundamental garantir que a tensão de alimentação fornecida à mini ponte H esteja dentro das especificações recomendadas para os motores DC em uso. Isso assegura um funcionamento adequado e evita danos tanto à mini ponte H quanto aos motores (40).

A operação da mini ponte H L298N envolve o controle de sinais de entrada provenientes de um microcontrolador, como um Arduino. Dependendo do sinal de entrada, a mini ponte H L298N ajusta a polaridade aplicada ao motor para determinar sua direção de rotação. Isso permite que o motor gire para frente ou para trás, conforme necessário, facilitando o controle preciso da direção do motor em diversas aplicações (40).

Os pinos IN1, IN2, IN3 e IN4 em uma mini ponte H L298N são usados para controlar a direção de rotação dos motores DC conectados a ela. Cada par de pinos é responsável pelo controle individual de um motor, permitindo determinar a direção em que cada motor girará. Por exemplo, quando o pino IN1 é configurado como estado lógico alto e o pino IN2 como estado lógico baixo, o motor gira em uma direção específica, por outro lado, invertendo esses valores, o motor girará na direção oposta. No entanto, se ambos os pares de pinos estiverem

configurados com o mesmo estado lógico (alto ou baixo), o motor não se moverá (40). A Tabela 2.1 e Tabela 2.2 apresentam o sentido de rotação para cada combinação dos pares de pinos.

Tabela 2.1 – Sentido de rotação para os pinos IN1 e IN2.

IN1	IN2	Motor A
Baixo	Baixo	Parado
Alto	Baixo	Sentido horário
Baixo	Alto	Sentido anti-horário
Alto	Alto	Parado

Fonte: O autor.

Tabela 2.2 – Sentido de rotação para os pinos IN3 e IN4.

IN3	IN4	Motor B
Baixo	Baixo	Parado
Alto	Baixo	Sentido horário
Baixo	Alto	Sentido anti-horário
Alto	Alto	Parado

Fonte: O autor.

Assim, a mini ponte H interpreta esses sinais para ajustar a polaridade aplicada aos terminais dos motores, permitindo um controle preciso sobre suas direções de rotação. Essa capacidade torna a mini ponte H uma ferramenta valiosa para automatizar o funcionamento da esteira, possibilitando desligar e ligar os motores conforme necessário, facilitando o controle do movimento do sistema de forma eficiente e conveniente.

2.8 MÓDULO PCA9685

O módulo PCA9685 é frequentemente utilizado para controlar servo motores devido à sua capacidade de gerar sinais PWM precisos e estáveis. Seu nome é baseado no chip controlador PCA9685. Quando usado para controlar servo motores, ele atua como um gerador de sinal PWM para cada canal independentemente (42). A Figura 2.11 apresenta uma ilustração do módulo PCA9685.

Figura 2.11 – Módulo PCA9685.



Fonte: (41).

Para controlar um servo motor com o PCA9685, cada canal do módulo é conectado ao terminal de controle do respectivo servo motor. O terminal de controle é onde o sinal PWM é enviado para controlar a posição do servo. Em seguida, o microcontrolador envia comandos ao PCA9685 através de uma interface de comunicação, como I2C, para definir o valor de ciclo de trabalho desejado para cada canal. Esse valor de ciclo de trabalho determina a posição do servo motor (42).

Conforme explicado anteriormente, um servo motor tem uma faixa de valores de ciclo de trabalho que correspondem a diferentes posições angulares. Por exemplo, um ciclo de trabalho de 5% pode corresponder a uma posição de 0 grau, enquanto um ciclo de trabalho de 10% pode corresponder a uma posição de 180 graus, e assim por diante. Ao ajustar os valores de ciclo de trabalho enviados para os canais do PCA9685, é possível controlar a posição de múltiplos servo motores simultaneamente (42).

Para utilizar o módulo PCA9685 para controlar servo motores, é essencial fornecer alimentações separadas tanto para o próprio módulo quanto para os servos motores. Geralmente, o módulo PCA9685 é alimentado com uma tensão de 5V, garantindo seu correto funcionamento. Já a alimentação dos servos motores deve ser ajustada de acordo com suas especificações individuais (42).

2.9 CONSIDERAÇÕES FINAIS

Nesta parte foi abordado os principais componentes do protótipo, sendo eles a esteira, o sensor de cor, o Arduino, o seletor e o manipulador robótico, juntamente com outros componentes como o micro servo, módulo PCA9685 e a mini ponte H L298N. Também foi possível distinguir os processos que estes componentes proporcionam, desde a identificação da cor, o processamento de dados e a atuação do manipulador e do seletor. Nos próximos capítulos, será abordada a metodologia e os recursos que serão necessários para o planejamento e desenvolvimento do projeto respectivamente.

3 MATERIAIS E MÉTODOS

Neste capítulo será abordado a respeito dos materiais e métodos que serão utilizados durante o trabalho. Para uma melhor organização, este capítulo é separado em duas partes, onde a primeira aborda os materiais necessários e suas funcionalidades gerais para confecção do protótipo, a segunda parte discute uma metodologia para que seja organizado um passo a passo para realização do mesmo.

3.1 MATERIAIS

Na Tabela 3.1 são apresentados os principais materiais utilizados no projeto e seus custos. A Figura 3.1 ilustra estes componentes para desenvolvimento do trabalho.

Tabela 3.1 – Principais componentes e seus custos.

Quantidade	Componente	Custo
1	Sensor de cor TCS230	R\$ 32,90
1	Arduino Mega 2560	R\$ 151,90
1	Esteiras transportadoras de acrílico	R\$ 374,99
1	Manipulador robótico de acrílico com micro servos	R\$ 127,99
1	Módulo PWM PCA9685 I2C – 16 Canais para Servo Motor	R\$ 59,76
1	Mini Ponte H Dupla L298N	R\$ 7,50
-	Demais componentes (Blocos coloridos, recipientes, jumpers, suportes e dentre outros)	R\$ 100,00
-	Total	R\$ 855,04

Fonte: O autor.

Figura 3.1 – Componentes principais para desenvolvimento do trabalho.



Fonte: Adaptado da internet.

Estes componentes principais integrados possibilitam automatizar a separação dos blocos pela sua coloração. O manipulador robótico é importante para simular a transferência de um bloco do dispenser para a esteira. O sensor de cor TCS230 é essencial, pois é com ele que será feita a identificação da cor dos blocos. Em seguida, será necessário o Arduino Mega 2560, pois ele vai receber, interpretar e processar os dados do sensor de cor, além de acionar os seletores corretos e controlar os movimentos repetitivos do manipulador robótico. Ademais, é

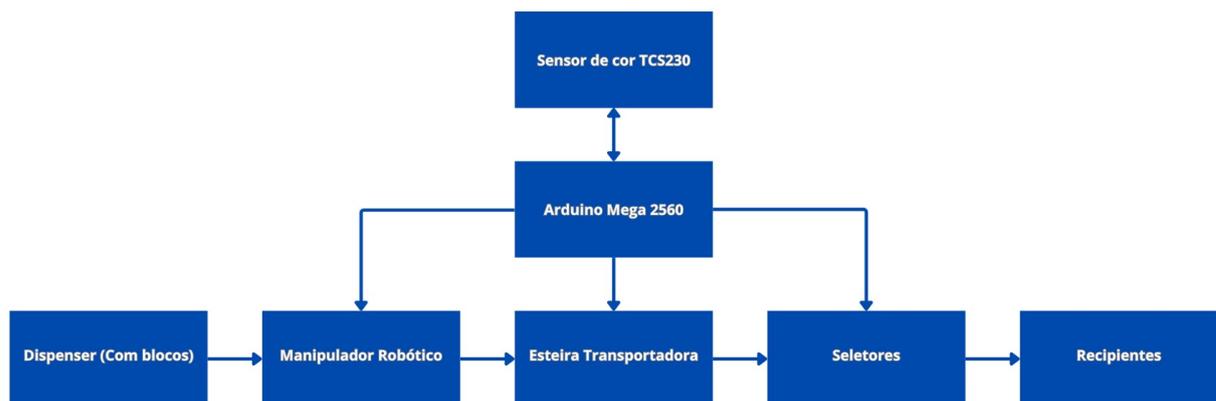
possível regular a velocidade da esteira usando o Arduino ou a Mini Ponte H. Já os atuadores são fundamentais por fazer a separação dos blocos e por fim a esteira transportadora é essencial por locomover os blocos, simulando assim um processo industrial.

Foi escolhido o sensor de cor TCS230 por ser mais econômico e por atender os requisitos necessários do projeto. O Arduino Mega 2560 foi selecionado por ser mais robusto, com mais portas de entrada e saída de dados, em comparação com outras placas Arduino, e também por possuir o protocolo de comunicação I2C que será usado juntamente com o módulo PCA9685 a fim de acionar os seletores.

Para uma compreensão completa do funcionamento do projeto, a Figura 3.2 apresenta um diagrama representativo. Nesta representação, é possível visualizar que o dispenser contém os blocos que serão agarrados pelo manipulador robótico e, posteriormente, depositados sobre a esteira. Uma vez que a peça é colocada na esteira, esta é ativada e prossegue até chegar ao sensor, onde então é interrompida para realizar a leitura da cor do bloco.

Após a leitura, o sensor transmite essa informação para o Arduino Mega, que interpreta os dados recebidos. Em seguida, o seletor correspondente à cor lida é acionado, reiniciando o funcionamento da esteira e direcionando os blocos para os recipientes designados. Observa-se também pelo diagrama que o Arduino Mega desempenha um papel central no controle do manipulador robótico, da esteira transportadora, do sensor de cor e do seletor, interpretando os dados do sensor e acionando os demais componentes.

Figura 3.2 – Diagrama Representativo do Projeto.



Fonte: O autor.

3.2 MÉTODOS

Para a confecção deste trabalho, primeiramente foi feito o levantamento bibliográfico para entender mais sobre o tema e como pode ser feito o projeto, em seguida foi estudado o

referencial teórico e feito a redação do projeto de pesquisa. A metodologia adotada para fazer este projeto possui seis etapas, sendo estas:

1) Estudo e Levantamento Bibliográfico dos Componentes

Na primeira etapa, o objetivo principal consistia em pesquisar as características e o funcionamento dos materiais disponíveis, bem como selecionar quais seriam os mais adequados para o projeto. Essa etapa apresenta grande importância, pois demanda a análise detalhada das características de cada componente, visando à integração deles para o funcionamento do protótipo.

2) Montagem do Manipulador Robótico

Na segunda etapa, o foco principal foi a construção do manipulador robótico, além do desenvolvimento de circuitos e códigos dedicados a mapear as posições dos servos e automatizar tarefas específicas, como a transferência dos blocos do dispenser para a esteira transportadora. Essa etapa exigiu a realização de testes durante a montagem para calibrar a posição das peças conectadas nos micros servos.

3) Montagem da Esteira Transportadora

Na terceira etapa, realizou-se a montagem da esteira transportadora, juntamente com o desenvolvimento de um circuito e código para os testes iniciais, marcando o início de sua operação para o deslocamento de blocos. Em seguida, procedeu-se à integração das funcionalidades do manipulador robótico à esteira, estabelecendo um processo em que o manipulador deposita um bloco, acionando automaticamente a esteira após o depósito.

4) Integração do Sensor de Cor

Na quarta etapa, o principal objetivo foi desenvolver um circuito e código para efetuar a leitura das cores dos blocos. Esta etapa foi importante, uma vez que permitiu a detecção das cores dos blocos que estavam na esteira transportadora. Por meio desse processo, o Arduino recebia as informações sobre a cor do bloco, possibilitando a correta separação posteriormente. Além disso, foi nesta fase que foi integrado o suporte e proteção para o sensor de cor.

5) Integração dos Seletores e União das Funcionalidades

Na quinta etapa, o principal objetivo foi desenvolver um circuito e código para garantir o funcionamento adequado dos servos motores dos seletores, além de proceder com a fixação

da plataforma na esteira transportadora. Durante essa fase, realizaram-se testes para calibrar as hastes dos servos motores e assegurar sua correta posição. Esta etapa foi importante para garantir que a separação dos blocos seja eficaz.

Além disso, nesta fase também foram interligados todos os circuitos no Arduino Mega 2560, de forma a obter o circuito final com todos os componentes para que o protótipo final funcionasse em conjunto. Assim como algumas etapas anteriores, esta também consistiu de testes e adaptações até que o protótipo funcionasse de acordo com o esperado. Esse foi o ponto em que o protótipo realmente começou a funcionar, integrando a transferência, deslocamento, detecção, processamento e separação de um bloco com base na sua cor.

6) Ajustes Finais e Validação

Por fim, a sexta etapa teve como objetivo validar o funcionamento e realizar os testes e ajustes finais em cada parte do protótipo para buscar sempre melhorar o mesmo com uma maior eficiência, precisão e menor taxa de erros. Isto foi importante pois, no caso do manipulador robótico foi preciso ter uma boa precisão e sensibilidade para não empurrar ou deixar cair os blocos com movimentos bruscos. Além disso, vale ressaltar que o manipulador robótico precisava de uma precisão ainda maior para depositar o bloco o mais centralizado possível na esteira, garantindo um fluxo de operação suave e eficaz.

4 DESENVOLVIMENTO

O objetivo deste capítulo é apresentar o desenvolvimento rumo à criação do protótipo final, enquanto busca-se também uma abordagem didática para destacar os métodos empregados neste estudo, estabelecendo conexões com os tópicos abordados na grade curricular do curso de Engenharia Eletrônica e de Telecomunicações.

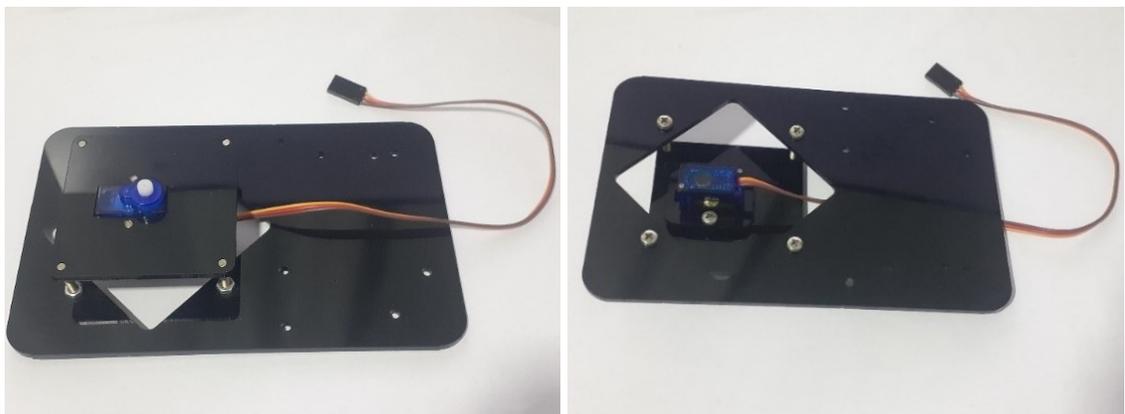
Além disso, serão expostos os resultados e as discussões decorrentes dos estudos teóricos e dos testes práticos realizados, culminando na obtenção de uma versão aprimorada do protótipo de automação da esteira seletora de blocos. Este projeto foi dividido em quatro partes principais: o manipulador robótico, responsável pela transferência dos blocos; a esteira transportadora, encarregada de seu deslocamento; o sensor de cor, incumbido de identificar as cores; e os seletores, encarregados da separação dos blocos. Ainda foi dedicada uma seção para falar dos periféricos, elaborados para acomodar as peças.

4.1 MANIPULADOR ROBÓTICO

No estágio inicial do projeto, deu-se início à montagem do manipulador robótico. Primeiramente, realizou-se a montagem da parte inferior, como mostrado na Figura 4.1. Em seguida, procedeu-se com o encaixe da base do manipulador no micro servo correspondente, conforme ilustrado na Figura 4.2.

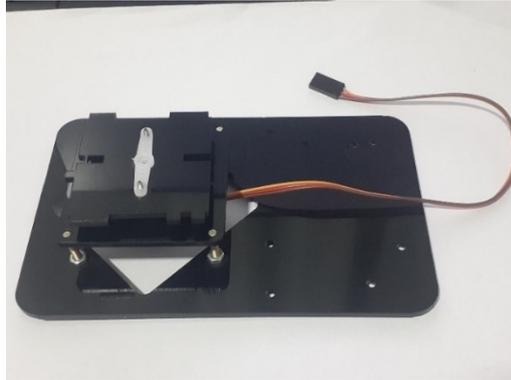
Após a conclusão da etapa anterior, o próximo passo consistiu na montagem das hastes laterais e da garra. Enquanto a maior haste lateral é responsável por movimentar a garra para frente ou para trás, a menor controla o movimento da garra para cima ou para baixo, conforme ilustrado na Figura 4.3 e Figura 4.4, respectivamente. Além disso, a função da garra é abrir e fechar para agarrar os blocos do dispenser, como mostrado na Figura 4.5.

Figura 4.1 – Parte inferior do manipulador robótico (Vista superior e inferior).



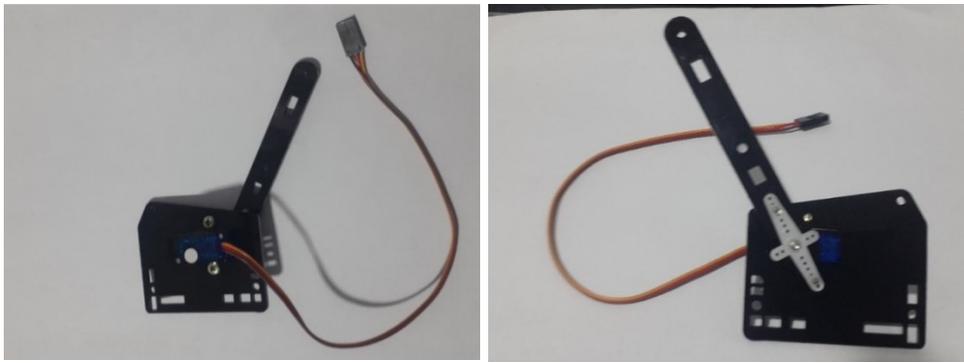
Fonte: O autor.

Figura 4.2 – Base do manipulador encaixada no micro servo para realização de testes.



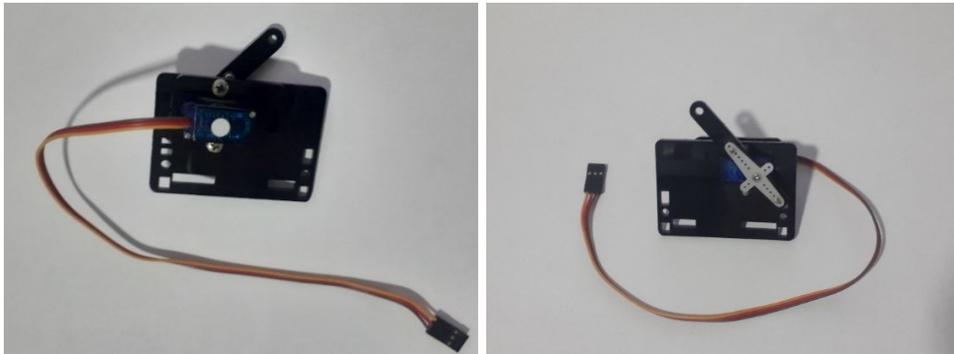
Fonte: O autor.

Figura 4.3 – Hastes lateral maior (Vista de ambos os lados).



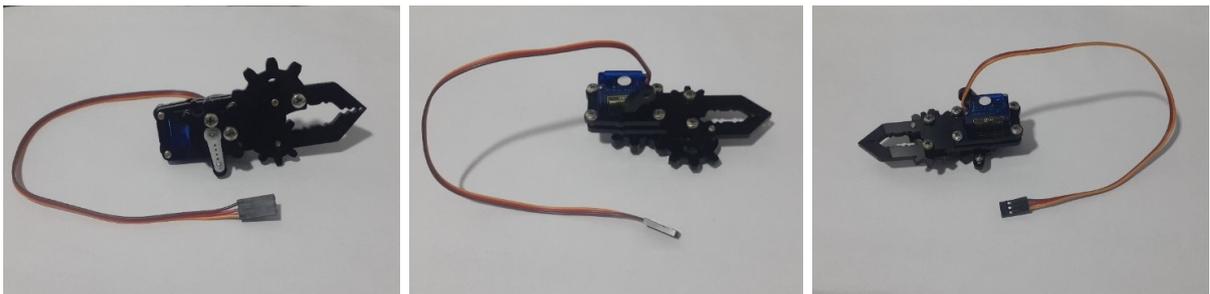
Fonte: O autor.

Figura 4.4 – Hastes lateral menor (Vista de ambos os lados).



Fonte: O autor.

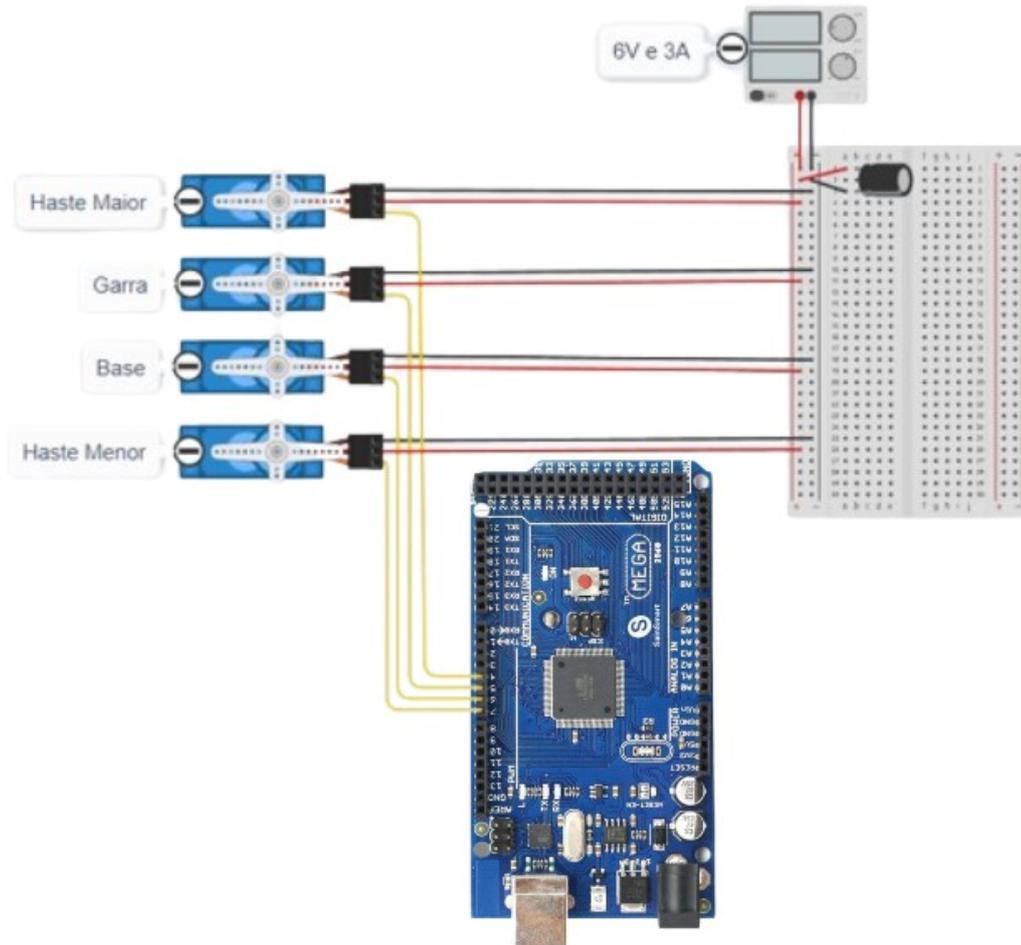
Figura 4.5 - Garra do manipulador robótico (Vistas laterais e inferior).



Fonte: O autor.

Antes de prosseguir com a montagem, é fundamental realizar testes em cada um dos servos para verificar o correto posicionamento dos objetos que serão girados pelo servo motor, para isso é necessário montar um circuito para essas partes já montadas. A Figura 4.6 mostra um circuito feito no *software* Tinkercad para representar o circuito físico de teste nos micros servos do manipulador robótico.

Figura 4.6 – Circuito para teste dos micros servos.



Fonte: O autor.

Ao analisar as especificações do micro servo, foi observado que a sua tensão de operação é de 3,0 a 7,2V. Dessa forma, foi utilizada uma fonte de 6V e 3A, a qual também será empregada para alimentar outras partes do circuito no futuro. Na Figura 4.6 é perceptível a conexão da fonte com os micros servos, destacando-se que a alimentação está em paralelo com cada servo, além do capacitor eletrolítico de 1000µF. Sendo assim, a tensão em cada um desses componentes também será de 6V, que corresponde ao valor da fonte, enquanto os 3A são distribuídos no circuito conforme a necessidade dos componentes. O capacitor eletrolítico mencionado anteriormente terá a função de estabilizar a energia dos micros servos diante de variações externas no circuito.

Além disso, é possível verificar que o Arduino Mega possui quatro ligações nas portas digitais PWM 4, 5, 6 e 7 para enviar o sinal de informação para a haste maior, a garra, a base e a haste menor, respectivamente, conforme ilustrado na Figura 4.6. Entretanto, para efetuar os testes, é necessário programar o Arduino Mega. Nesse sentido, o código apresentado na Tabela 4.1 foi desenvolvido para tais testes. Esse código permite a rotação individual de um micro servo, possibilitando a validação e ajuste da posição mais adequada de uma peça em relação ao eixo do servo.

Tabela 4.1 – Código usado para teste dos micros servos.

```
#include <Servo.h>

Servo servo;

void setup() {
  servo.attach(4); // Pino 4 do Arduino Mega para para controlar a haste maior
}

void loop() {
  // Teste das posições de 90°, 180° e 0° respectivamente
  servo.write(90);      // Mover o servo para a posição de 90°.
  delay(500);          // Aguardar 500ms.
  servo.write(180);    // Mover o servo para a posição de 180°.
  delay(500);          // Aguardar 500ms.
  servo.write(0);      // Mover o servo para a posição de 0°.
  delay(500);          // Aguardar 500ms.

  // Teste para validar a variação das posições com um movimento suave do servo.
  // Faz a variação crescente das posições i com incremento de 1 do ângulo 0° até o ângulo 180°, com delay de
  //15ms.
  for (int i = 0; i <= 180; i += 1) {
    servo.write(i);    // Move o servo para a posição i
    delay(15);         // Aguarda 15ms.
  }
  delay(500);          // Aguarda 500ms.

  // Faz a variação decrescente das posições i com decremento de 1 do ângulo 180° até o ângulo 0°, com delay
  //de 15ms.
  for (int i = 180; i >= 0; i -= 1) {
    servo.write(i);    // Move o servo para a posição i
    delay(15);         // Aguarda 15ms.
  }
  delay(500);          // Aguarda 500ms.
}
```

Fonte: O autor.

O código apresentado tem o propósito de controlar um servo motor, deslocando-o para diferentes posições angulares. Inicialmente, a biblioteca “Servo” é incorporada para facilitar o controle do servo. Um objeto “servo” é então declarado e configurado para usar o pino 4 do Arduino Mega. Na função “loop()”, o servo motor é direcionado para posições específicas de 90°, 180° e 0°, respectivamente, com pausas de 500ms entre cada movimento. Em seguida, o código executa variações suaves na posição do servo de 0° a 180° e de volta a 0°, com

incrementos e decrementos de 1 grau em cada etapa, gerando um movimento contínuo e suave. Cada movimento é acompanhado de uma pausa de 15ms, e após cada ciclo de movimento, há uma pausa adicional de 500ms. Este código ilustra o controle básico de um servo motor com o Arduino, empregando posições específicas e movimentos suaves.

Além disso, o código apresentado na Tabela 4.1, é desenvolvido para testar individualmente um micro servo por vez. No contexto deste código, o teste é conduzido no servo responsável pelo movimento da haste maior do manipulador, o qual está conectado ao pino digital 4 do Arduino Mega. Após a realização do teste e os devidos ajustes da posição da haste em relação ao eixo do servo, é crucial adaptar o código para o próximo servo a ser testado. Por exemplo, ao testar a base, o pino 6 seria utilizado em vez do pino 4, sendo necessário então modificar o código para refletir essa mudança antes de carregá-lo no Arduino. Seguindo essa adaptação, o teste e o ajuste da base podem ser realizados corretamente.

Para realizar o ajuste adequado da base, por exemplo, é essencial verificar se ela está configurada para permitir um movimento de rotação de 90 graus tanto para a esquerda quanto para a direita. Caso seja identificado algum desalinhamento na posição da base, é necessário removê-la e reposicioná-la cuidadosamente até que atenda exatamente à configuração desejada. Após concluir os ajustes, a base pode ser fixada no micro servo para garantir estabilidade e precisão nos movimentos. Essa etapa de ajuste para cada componente principal do manipulador robótico é essencial para o correto funcionamento e desempenho do mesmo.

Após a conclusão dos testes e ajustes individuais para cada servo do manipulador, o próximo passo envolve a montagem física do restante do manipulador robótico. Nesse contexto, as figuras a seguir exemplificam o processo de unir a base com a haste maior (Figura 4.7) e, em seguida, com a haste menor (Figura 4.8). Posteriormente, a garra é incorporada, resultando na montagem completa do manipulador robótico, pronto para uso (Figura 4.9). Vale destacar que, entre as uniões dessas quatro partes que envolvem um servo, são adicionadas outras peças para facilitar o encaixe e garantir a estabilidade do manipulador.

Figura 4.7 – Junção da base com a haste maior do manipulador robótico.



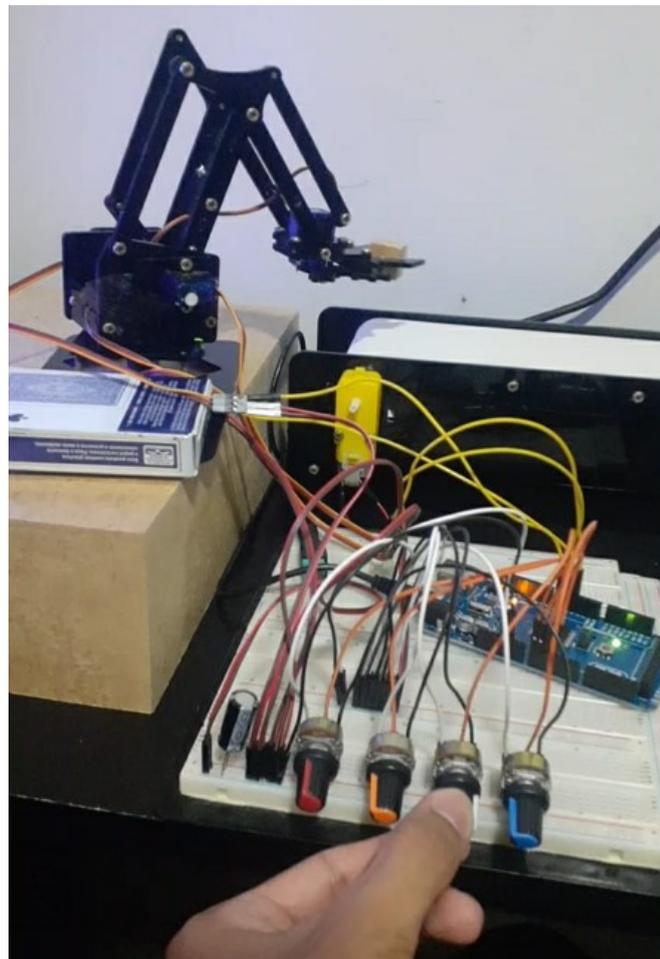
Fonte: O autor.

Figura 4.8 – Junção da haste menor no manipulador robótico.



Fonte: O autor.

Figura 4.9 – Manipulador robótico montado.

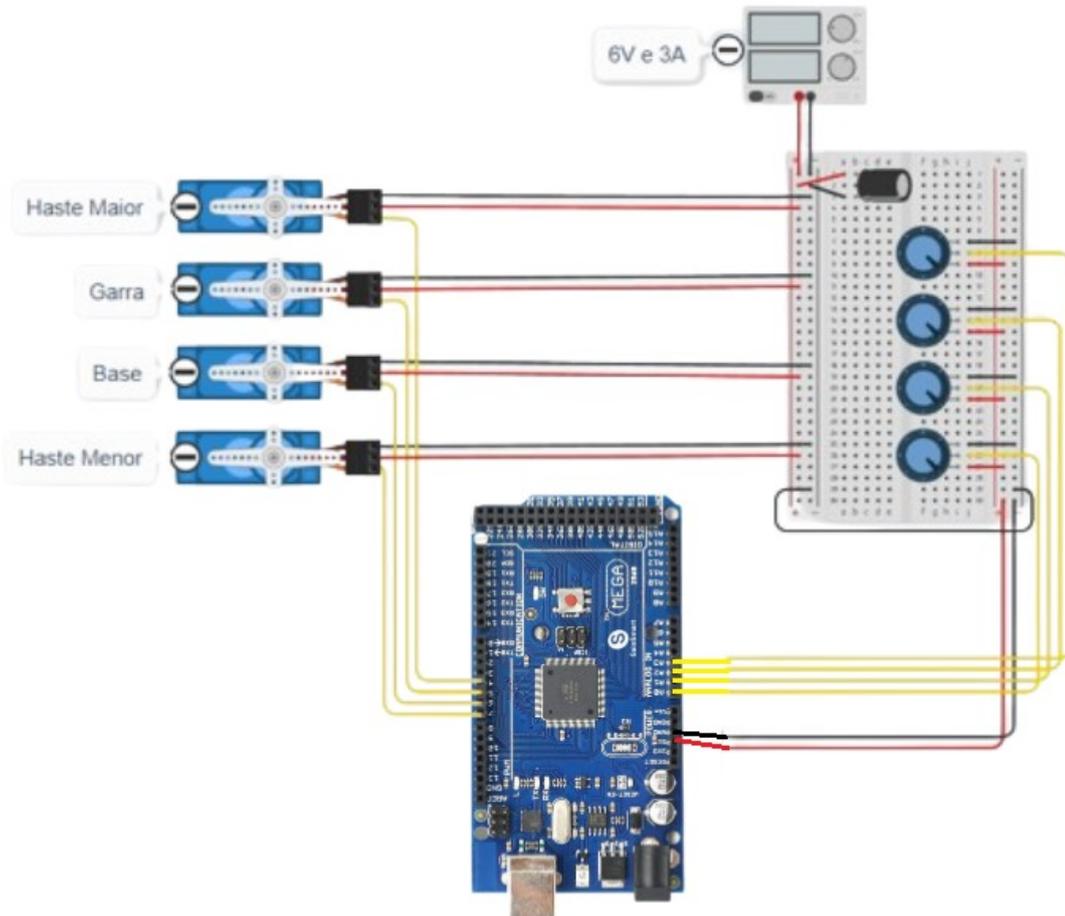


Fonte: O autor.

Para garantir a precisão na execução do movimento repetitivo de pegar o bloco do dispenser e colocá-lo na esteira transportadora, é necessário desenvolver um algoritmo que permita ao manipulador robótico repetir esses movimentos com exatidão. Para isso, primeiramente é importante montar um novo circuito, utilizando potenciômetros, a fim de controlar a posição de cada servo motor e monitorar suas posições por meio do monitor serial. A Figura 4.10 ilustra o circuito utilizado para controlar manualmente a posição dos servos

motores, enquanto a Tabela 4.2 apresenta o código empregado para mapear os movimentos repetitivos do manipulador robótico.

Figura 4.10 – Circuito com potenciômetros para controlar e analisar a posição dos servos motores.



Fonte: O autor.

O circuito apresentado na Figura 4.10 apresenta a inclusão dos potenciômetros em comparação com sua versão anterior, representada na Figura 4.6. Estes dispositivos são alimentados diretamente pela tensão de 5V fornecida pelo Arduino. Cada potenciômetro é conectado individualmente a uma porta analógica do Arduino (A0, A1, A2 e A3), permitindo a medição da variação de tensão resultante da alteração da resistência. Posteriormente, o Arduino converte essa tensão em um valor digital, simplificando a interpretação dos dados obtidos. Além disso, é importante mencionar que as portas digitais PWM 4, 5, 6 e 7 permanecerão conforme abordado no circuito da Figura 4.6. Para garantir uma referência de terra consistente e estável, foi estabelecida a conexão entre o terra da fonte externa e o terra do Arduino.

Tabela 4.2 – Algoritmo utilizado para mapear os movimentos repetitivos do manipulador robótico.

```

#include <Servo.h>

Servo servo1; // Servo da haste maior (Mover a garra para frente ou para trás)
Servo servo2; // Servo da garra
Servo servo3; // Servo da base
Servo servo4; // Servo da haste menor (Mover a garra para cima ou para baixo)

int angulo1;
int angulo2;
int angulo3;
int angulo4;
unsigned long timer = 1;
const unsigned long intervalo = 3000;

void setup() {
  Serial.begin(9600);

  servo1.attach(4);
  servo2.attach(5);
  servo3.attach(6);
  servo4.attach(7);
}

void loop() {
  // Leitura dos potenciômetros
  angulo1 = map(analogRead(A0),0,1023,0,180);
  angulo2 = map(analogRead(A1),0,1023,0,180);
  angulo3 = map(analogRead(A2),0,1023,0,180);
  angulo4 = map(analogRead(A3),0,1023,0,180);

  // Posicionamento dos servos
  servo1.write(angulo1);
  servo2.write(angulo2);
  servo3.write(angulo3);
  servo4.write(angulo4);

  if ((millis() - timer) >= intervalo) {
    Serial.println("*****");

    Serial.print("Pot1:");
    Serial.print(analogRead(A0));
    Serial.print(" Angulo Motor1:");
    Serial.println(angulo1);

    Serial.print("Pot2:");
    Serial.print(analogRead(A1));
    Serial.print(" Angulo Motor2:");
    Serial.println(angulo2);

    Serial.print("Pot3:");
    Serial.print(analogRead(A2));
    Serial.print(" Angulo Motor3:");
    Serial.println(angulo3);

    Serial.print("Pot4:");
    Serial.print(analogRead(A3));
    Serial.print(" Angulo Motor4:");
    Serial.println(angulo4);

    timer = millis();
  }

  delay(100);
}

```

Fonte: O autor.

O código apresentado na Tabela 4.2 implementa um sistema de controle de servo motores utilizando os potenciômetros da Figura 4.10 para definir as posições de cada servo. Inicialmente, são declarados quatro objetos do tipo Servo para controlar quatro servos motores conectados aos pinos 4, 5, 6 e 7 da placa Arduino, que correspondem à haste maior, garra, base e haste menor, respectivamente. No método `setup()`, os objetos Servo são inicializados e associados aos pinos corretos da placa Arduino.

No loop principal, os valores lidos dos potenciômetros conectados às portas analógicas A0, A1, A2 e A3 são mapeadas para o intervalo de 0 a 180 (que corresponde aos ângulos de rotação dos servos) utilizando a função `map()`. Em seguida, esses valores são usados para posicionar os servos, por meio do método `write()`. Além disso, a cada intervalo de 3 segundos, os valores dos potenciômetros e os ângulos correspondentes dos motores são impressos via porta serial para monitoramento. A Figura 4.11 apresenta uma leitura do monitor serial com esses valores exibidos.

Figura 4.11 – Leitura dos ângulos dos servos motores no manipulador robótico pelo monitor serial.

```
*****
Pot1:447 Angulo Motor1:78
Pot2:859 Angulo Motor2:151
Pot3:589 Angulo Motor3:103
Pot4:794 Angulo Motor4:139
```

Fonte: O autor.

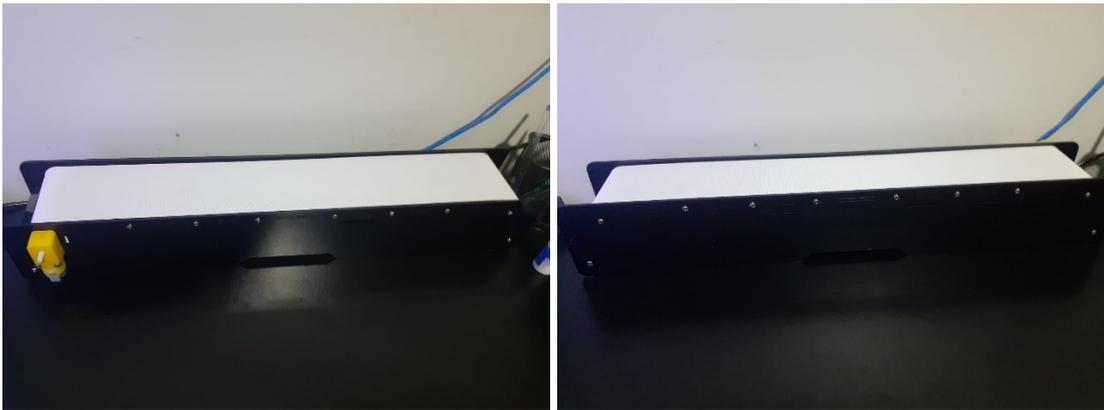
Este código é útil para controlar o manipulador robótico, permitindo ajustar as posições dos servos em tempo real por meio dos potenciômetros e monitorar esses valores usando o monitor serial. Por meio desse código, torna-se possível programar uma sequência ordenada de movimentos para o manipulador, abrangendo desde a definição de sua posição inicial até tarefas específicas, como rotacionar a base, abrir e fechar a garra e até mesmo aproximar dos blocos. A sequência de movimentos que o manipulador precisa realizar depende da posição do dispenser, bem como do bloco no dispenser e também da esteira transportadora. Desta forma, esta sequência foi programada de acordo com estas variáveis.

Após o mapeamento, é importante ressaltar que o circuito retorna à configuração apresentada na Figura 4.6. No entanto, é fundamental manter a conexão entre o *Ground* (GND) do Arduino e o GND da fonte externa. Isso se deve ao fato de que posteriormente será mencionado que outros dispositivos foram conectados em paralelo à alimentação do Arduino, como a Mini Ponte H, o sensor de cor e o módulo PCA9685.

4.2 ESTEIRA TRANSPORTADORA

Concluída a montagem do manipulador robótico, a próxima etapa foi a montagem da esteira transportadora. A Figura 4.12 mostra a esteira montada. Observa-se que a esteira é impulsionada por um único motor DC de 3 a 6V conectado a uma roldana. Quando acionado, o motor faz com que a roldana gire, movimentando assim a correia da esteira e deslocando os blocos sobre ela.

Figura 4.12 – Esteira Transportadora (Frente e verso).

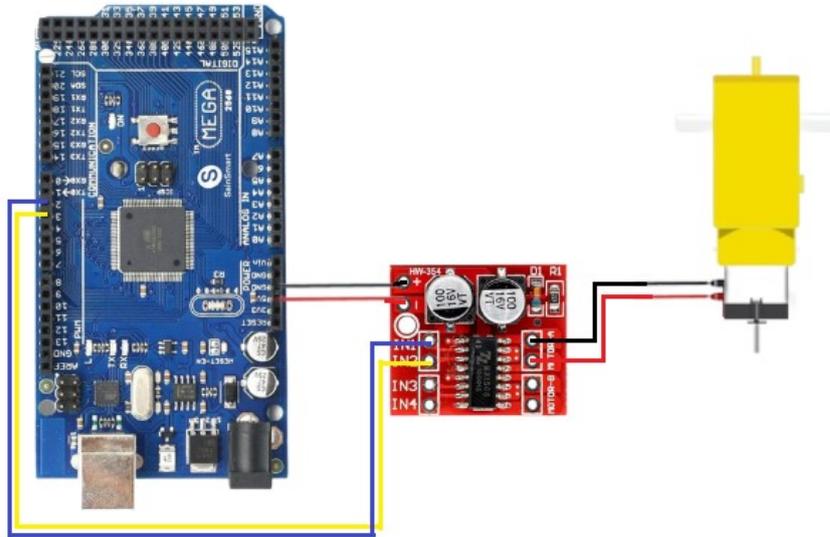


Fonte: O autor.

Pela Figura 4.12 é possível notar que há uma barreira nas laterais da esteira acima da correia, desta forma foi introduzido duas placas de MDF de 3mm cada, unidas uma acima da outra e fixadas acima do acrílico que está logo abaixo da correia da esteira. As extremidades dessas placas foram lixadas para evitar qualquer resistência ao movimento da correia. Desta forma, quando for introduzido os seletores no projeto, os blocos não terão resistência ao cair nos recipientes. Além disso, no projeto final, foram adicionados um suporte para o sensor de cor no início da esteira e uma plataforma para a fixação dos seletores, os quais serão discutidos posteriormente.

Após a montagem e adaptação da esteira, foi elaborado um circuito para controlar seu acionamento por meio do Arduino Mega conforme necessário. A Figura 4.13 ilustra este circuito, onde se destaca o uso de uma Mini Ponte H L298N para controlar o funcionamento do motor DC da esteira. É importante notar que ela é alimentada diretamente pela tensão de 5V fornecida pelo Arduino, dessa forma, os pinos de saída dela que realizam a alimentação do Motor DC. Além disso, o pino IN1 da Mini Ponte H foi conectado à porta digital 2, enquanto o pino IN 2 foi associado à porta digital 3 do Arduino.

Figura 4.13 – Circuito para funcionamento da Esteira Transportadora.



Fonte: O autor.

Entretanto, para funcionar o circuito, é necessário carregar o código para o Arduino Mega para ligar e desligar a esteira, quando necessário. Desta forma, a Tabela 4.3 mostra o código desenvolvido para controlar a esteira.

Tabela 4.3 – Código para funcionamento da esteira.

```
const int IN1 = 2; // Pino 2 do Arduino Mega 2560 conectado no pino IN1 da mini ponte H.
const int IN2 = 3; // Pino 3 do Arduino Mega 2560 conectado no pino IN2 da mini ponte H.

void setup() {

  pinMode(IN1, OUTPUT); // Pino IN1 configurado como saída.
  pinMode(IN2, OUTPUT); // Pino IN2 configurado como saída.

  // Começando com o motor parado
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
}

void loop() {

  // Girar o motor no sentido horário
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);

  // Aguardar 425ms e parar o motor
  delay(425);
  digitalWrite(IN1, LOW);

  // Aguardar 500ms para leitura da cor
  delay(500);

  // Ligar a esteira
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);

  // Aguardar quatro segundos e parar o motor
  delay(4000);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);}
```

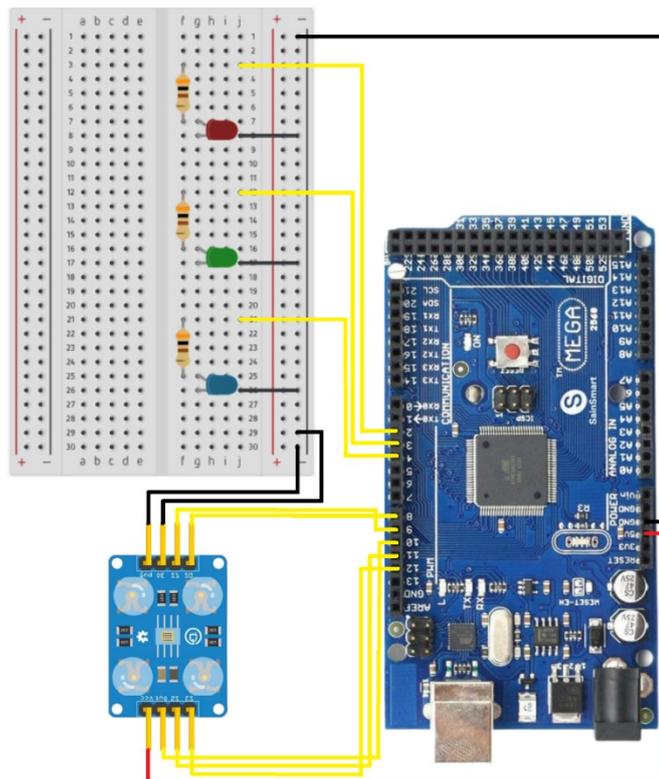
Fonte: O autor.

O código fornecido destina-se ao controle exclusivo da esteira transportadora do projeto final. Quando um bloco é depositado na esteira, o motor DC é ativado por 425ms no sentido horário, movendo a roldana, a correia e, conseqüentemente, o bloco no mesmo sentido. Esse período é suficiente para deslocar o bloco abaixo do sensor de cor TCS230. Após esse intervalo, a esteira é desligada por um momento, esperando 500ms para a leitura do bloco. Em seguida, é religada, desta vez por quatro segundos para separar os blocos, antes de ser desativada novamente.

4.3 SENSOR DE COR TCS230

A análise do sensor de cor é iniciada neste momento, reconhecendo sua importância como componente essencial do sistema, responsável pela detecção e identificação das cores. Antes de integrar diretamente o sensor de cor ao projeto principal, foram realizados testes exclusivamente com o sensor de cor TCS230. Para isso, foi montado um circuito que incluía este sensor, o Arduino Mega e alguns LEDs. Esses LEDs foram programados para acender conforme a detecção de cores específicas, como o LED vermelho ao identificar a cor vermelha, o LED verde ao detectar a cor verde e o LED azul ao reconhecer a cor azul. A Figura 4.14 ilustra este circuito.

Figura 4.14 – Circuito para testes do sensor de cor TCS230 com Arduino Mega.



Fonte: O autor.

No circuito, observa-se que os LEDs, em série com um resistor de 300Ω , estão conectados às portas D2, D3 e D4 do Arduino Mega, representando as cores vermelha, verde e azul, respectivamente. Assim, ao realizar a leitura de uma cor específica, o LED correspondente se acende. Além disso, a alimentação do sensor de cor é feita pelo próprio arduino, desta forma os pinos OE e GND do sensor de cor estão aterrados, o pino *Voltage at the Common Collector* (Vcc) está ligado nos 5V do Arduino e os pinos S0, S1, OUT, S2 e S3 estão conectados aos pinos D8, D9, D10, D11 e D12, respectivamente. Para garantir o funcionamento adequado do sensor, foi desenvolvido um código de teste, conforme detalhado na Tabela 4.4.

Tabela 4.4 – Código para testar a leitura das cores do sensor de cor TCS230.

```

const int S0 = 8;
const int S1 = 9;
const int S2 = 11;
const int S3 = 12;
const int OUT = 10;

const int LED_VERMELHO = 2;
const int LED_VERDE = 3;
const int LED_AZUL = 4;

int VERMELHO = 0;
int VERDE = 0;
int AZUL = 0;

void ler_cor();

void setup() {

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);

  // Configurando os LEDs como pinos de saída.
  pinMode(LED_VERMELHO, OUTPUT);
  pinMode(LED_VERDE, OUTPUT);
  pinMode(LED_AZUL, OUTPUT);

  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

  Serial.begin(9600);
}

void loop() {
  ler_cor();
  Serial.println("Valor do vermelho:");
  Serial.println(VERMELHO);
  Serial.println("Valor do verde:");
  Serial.println(VERDE);
  Serial.println("Valor do azul:");
  Serial.println(AZUL);

  if (VERMELHO < AZUL && VERMELHO < VERDE) {
    digitalWrite(LED_VERMELHO, HIGH);
    digitalWrite(LED_VERDE, LOW);
    digitalWrite(LED_AZUL, LOW);
    Serial.println("***** VERMELHO ACIONADO *****");
  }
}

```

```

}

if (VERDE < VERMELHO && VERDE < AZUL) {
  digitalWrite(LED_VERMELHO, LOW);
  digitalWrite(LED_VERDE, HIGH);
  digitalWrite(LED_AZUL, LOW);
  Serial.println("***** VERDE ACIONADO *****");
}

if (AZUL < VERMELHO && AZUL < VERDE) {
  digitalWrite(LED_VERMELHO, LOW);
  digitalWrite(LED_VERDE, LOW);
  digitalWrite(LED_AZUL, HIGH);
  Serial.println("***** AZUL ACIONADO *****");
}

delay(100);
}

void ler_cor() {

  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);

  if(digitalRead(OUT) == HIGH) {
    VERMELHO = pulseIn(OUT, LOW);
  } else {
    VERMELHO = pulseIn(OUT, HIGH);
  }

  digitalWrite(S3, HIGH);

  if(digitalRead(OUT) == HIGH) {
    AZUL = pulseIn(OUT, LOW);
  } else {
    AZUL = pulseIn(OUT, HIGH);
  }

  digitalWrite(S2, HIGH);

  if(digitalRead(OUT) == HIGH) {
    VERDE = pulseIn(OUT, LOW);
  } else {
    VERDE = pulseIn(OUT, HIGH);
  }
}
}

```

Fonte: O autor.

O código da tabela acima inicia declarando o valor dos pinos digitais que serão usados para controlar o sensor de cor (S0, S1, S2, S3 e OUT) e os LEDs que indicarão as cores vermelha, verde e azul (LED_VERMELHO, LED_VERDE e LED_AZUL). Em seguida, são definidas variáveis para armazenar os valores de vermelho, verde e azul, os quais serão lidos pelo sensor.

Na função setup, os pinos do Arduino são configurados, sendo definidos como saída (OUTPUT) ou entrada (INPUT), conforme necessário. Além disso, é estabelecido o estado inicial dos pinos S0 e S1, como ligado e desligado respectivamente, para configurar o sensor

de cor com uma escala de frequência na saída de 20%, considerada ideal para o Arduino. Por fim, a comunicação serial é iniciada com uma velocidade de 9600 bps.

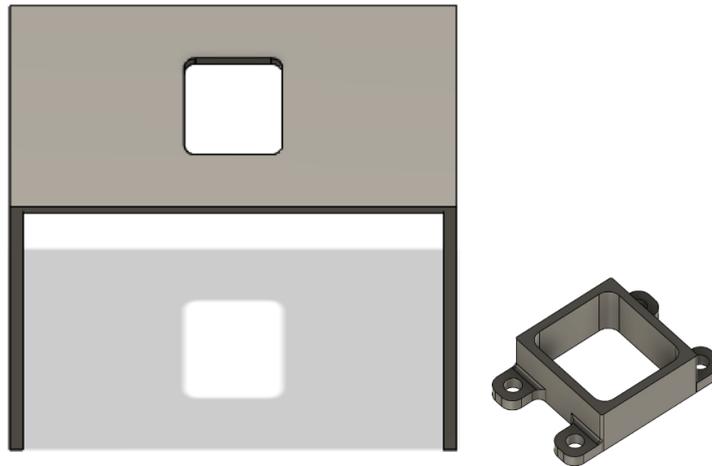
Na função `loop`, há um ciclo para ler as cores continuamente usando a função `ler_cor()`, em seguida mostrar os valores das cores vermelho, verde e azul no monitor serial e posteriormente acender os LEDs correspondentes com base nas cores detectadas. Para determinar qual LED deve ser aceso, é realizado uma verificação. O LED vermelho é aceso quando a variável `VERMELHO` é menor que as variáveis `VERDE` e `AZUL`. Para acender o LED verde, a variável `VERDE` deve ser menor que as variáveis `VERMELHO` e `AZUL`. Já para acender o LED azul, a variável `AZUL` precisa ser menor que as variáveis `VERMELHO` e `VERDE`.

A função `ler_cor()` inicia configurando os pinos S2 e S3 como desligados, para selecionar a matriz de fotodiodos vermelha. Em seguida, verifica-se se o pino OUT de saída do sensor de cor está em nível lógico alto. Se sim, a função `pulseIn()` é utilizada para medir a duração do pulso de saída do sensor quando o pulso está em nível lógico baixo. Caso contrário, a medição é realizada quando o pulso está em nível lógico alto. A duração deste pulso é então armazenada na variável `vermelho`. Posteriormente, o pino S3 é alterado para ligado, selecionando a matriz de fotodiodos azul, e o mesmo procedimento é realizado para ler a duração do pulso, com o resultado atribuído à variável `azul`. Por fim, o pino S2 é alterado para ligado, selecionando a matriz de fotodiodos verde, e o processo é repetido para ler a duração do pulso, com o resultado atribuído à variável `verde`. Assim, os valores das variáveis `vermelho`, `azul` e `verde` representam a duração do pulso quando a matriz de fotodiodos de sua cor correspondente é selecionada.

No projeto final, manteve as mesmas conexões conforme ilustrado na Figura 4.14. No entanto, houve uma pequena alteração: os circuitos dos LEDs foram removidos, assim como as partes do código onde eles eram utilizados.

Ao ligar o projeto em ambientes com diferentes níveis de luminosidade, observou-se uma variação nas leituras do sensor de cor, resultando em uma maior incidência de erros. Além disso, sem uma proteção para direcionar o sensor a ler a cor de uma área específica, ele pode captar as cores de uma região mais ampla. Também foi observado que a proximidade do sensor em relação ao bloco melhora significativamente a precisão da leitura. Para otimizar o desempenho do sensor, foi desenvolvido um suporte personalizado, uma proteção para evitar a interferência de cores externas e uma cortina para reduzir a luminosidade no ambiente de leitura. Os *designs* desses dois primeiros componentes foram criados utilizando o *software* Autodesk Fusion e estão ilustrados na Figura 4.15 a seguir.

Figura 4.15 – Suporte e proteção do sensor de cor TCS230 projetados no software Autodesk Fusion.



Fonte: O autor.

Durante a leitura dos blocos, constatou-se que a cor verde estava somente sendo detectada quando o bloco estava muito próximo do sensor, dificultando então a identificação da cor verde. No início do projeto, também estava planejado a separação das cores secundárias. Contudo, os problemas mencionados anteriormente foram encontrados. Houve tentativas de separar as cores secundárias no final da esteira, porém seria necessário mapear uma faixa específica de cor vermelha, verde e azul para cada bloco com uma das cores primária. Entretanto, levando em consideração as variações na posição em que o manipulador deposita o bloco e as flutuações na velocidade da esteira, o bloco pode estar ligeiramente deslocado para a esquerda, direita, para frente ou para trás do sensor de cor, apresentando uma infinidade de possíveis posições.

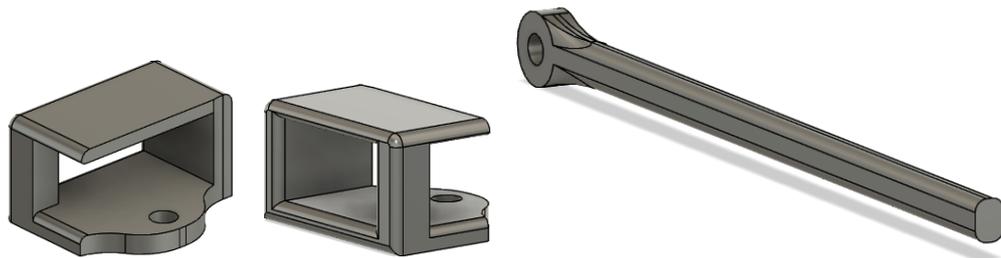
No entanto, para superar esse obstáculo, seria viável empregar um mecanismo para centralizar o bloco no meio da esteira, juntamente com a utilização de um sensor de infravermelho ou ultrassônico. Quando o sensor identificasse que o bloco alcançou a posição ideal, a esteira poderia ser imediatamente parada com o bloco logo abaixo do sensor de cor. Isso resultaria em uma menor variação das cores detectadas pelo sensor e facilitaria a criação de um mapeamento mais preciso das cores.

Além disso, é importante considerar que, para mapear cores secundárias, como o amarelo, é necessário reconhecer que ele consiste em uma combinação de vermelho e verde. Nesse sentido, o sensor de cor poderia identificar uma predominância de vermelho ou verde ao ler um bloco amarelo, exigindo o mapeamento adequado do amarelo para que seja depositado corretamente no final da esteira. Como resultado dessas limitações, o projeto foi ajustado para realizar apenas a separação das cores vermelho e azul, já que as demais apresentaram desafios significativos.

4.4 SELETORES

A seguir, será abordado sobre os seletores, que são peças fundamentais para efetivar a separação dos blocos. Os seletores deste projeto foram construídos utilizando micro servos e hastes fixadas em seus eixos. Inicialmente posicionadas verticalmente, essas hastes são acionadas para se deslocarem à frente da esteira, direcionando assim os blocos para os recipientes adequados. Tanto as hastes quanto as cases para os servos foram projetadas pelo *software* Autodesk Fusion e impressas por uma impressora 3D. A Figura 4.16 apresenta esses componentes desenvolvidos por meio do *software* mencionado.

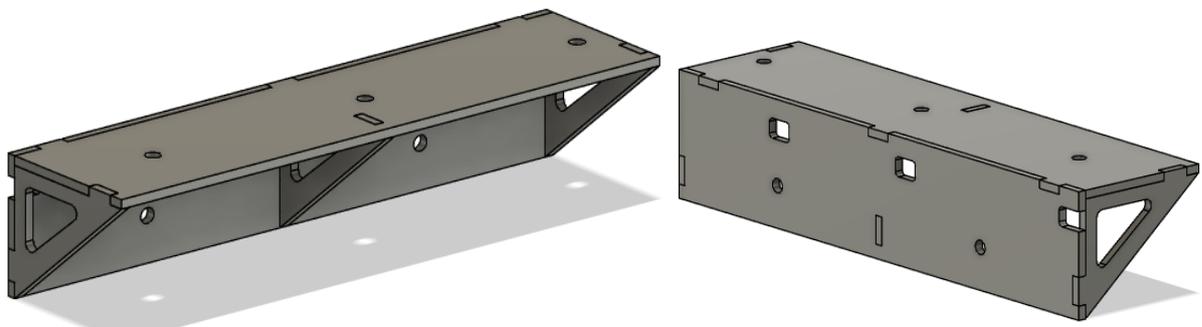
Figura 4.16 – Case e haste para os servos motores dos seletores.



Fonte: O autor.

Na figura acima, destaca-se que a case conta com um orifício específico para fixação do servo na plataforma dos seletores, que está ilustrado na Figura 4.17, assegurando estabilidade mesmo diante de colisões dos blocos com a haste. Da mesma forma que foi necessário calibrar os servos do manipulador robótico para posicionar suas peças adequadamente, também foi fundamental calibrar os seletores. Essa etapa foi realizada de modo que a posição inicial da haste permanecesse na vertical e, quando acionada, se posicionasse horizontalmente à frente da esteira para efetuar a separação dos blocos, conforme explicado anteriormente.

Figura 4.17 – Plataforma dos seletores (Vista frontal e traseira).

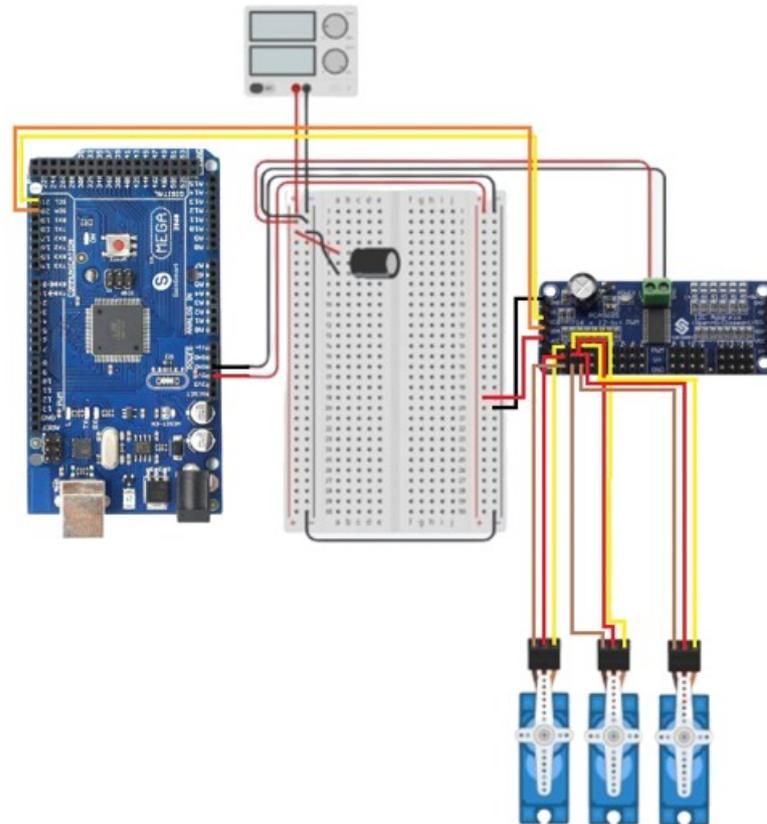


Fonte: O autor.

Desta forma, foi montado um circuito para integrar esses seletores, utilizando o módulo PCA9685 para otimizar o uso das demais portas digitais disponíveis no Arduino Mega. A Figura 4.18 ilustra o circuito para funcionamento dos seletores, enquanto a Tabela 4.5 apresenta

o código utilizado para ativar ou desativar os seletores. É importante ressaltar que tanto o circuito quanto os códigos apresentados foram desenvolvidos exclusivamente para testar os seletores. No final do projeto, todos os circuitos e códigos de cada componente chave serão integrados em um sistema unificado.

Figura 4.18 – Circuito para testes dos seletores.



Fonte: O autor.

No circuito da Figura 4.18, é possível notar que o módulo PCA9685 é alimentado pelo Arduino Mega, utilizando os pinos Vcc e GND conectados aos 5V e GND do Arduino, respectivamente, para fornecer energia ao chip do módulo. Além disso, é importante mencionar que a outra alimentação do PCA9685 é destinada aos servos motores conectados a este módulo, os quais são alimentados por uma fonte de 6V no projeto final. Para estabilizar a alimentação dos servos e garantir sua operação consistente, é necessário adicionar um capacitor eletrolítico de 1000µF em paralelo com sua alimentação, o que ajuda a mitigar possíveis variações no fornecimento de energia do circuito.

Os pinos SDA e SCL do módulo PCA9685 são conectados aos pinos 20 e 21 do Arduino Mega, respectivamente para comunicação I2C. Além disso, é possível visualizar que os servos podem ser conectados diretamente nos pinos da PCA9685, pois esses pinos oferecem as conexões essenciais para alimentação, sinal e aterramento. Por fim, vale destacar que o pino OE não foi utilizado neste circuito.

Tabela 4.5 – Código para testar o funcionamento dos seletores.

```

#include <Servo.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver seletor = Adafruit_PWMServoDriver();

#define servoMIN 150
#define servoMAX 600

Servo seletor1;
Servo seletor2;
Servo seletor3;

void setup() {
  seletor.setPWMFreq(60);
  // Deixar os seletores nas posições iniciais (Não acionados)
  seletor.setPWM(0, 0, servoMIN);
  seletor.setPWM(1, 0, servoMIN);
  seletor.setPWM(2, 0, servoMIN);
}

void loop() {
  // Acionamento do primeiro seletor caso cor vermelha for detectado
  seletor.setPWM(1, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(2, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(0, 0, 350); // Seletor acionado
  delay(1000);
  // Acionamento do segundo seletor caso cor verde for detectado
  seletor.setPWM(0, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(2, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(1, 0, 350); // Seletor acionado
  delay(1000);
  // Acionamento do terceiro seletor caso cor azul for detectado
  seletor.setPWM(0, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(1, 0, servoMIN); // Seletor não acionado
  seletor.setPWM(2, 0, 350); // Seletor acionado
  delay(1000);
}

```

Fonte: O autor.

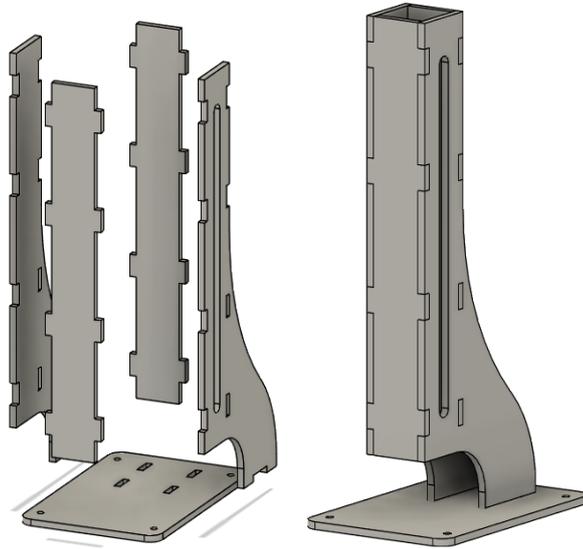
O código apresentado na Tabela 4.5 começa incluindo as bibliotecas “Servo.h”, “Wire.h” e “Adafruit_PWMServoDriver.h” para permitir o controle dos servos motores e a comunicação I2C com o PCA9685. Em seguida, são definidas as constantes “servoMIN” e “servoMAX”, que estabelecem os pulsos mínimos e máximos para os servos motores. No “setup()”, a frequência de PWM do PCA9685 é definida para 60Hz e os seletores são posicionados inicialmente como não acionados. Já no “loop()”, os seletores são acionados em sequência, com um intervalo de 1 segundo entre cada acionamento, ajustando os valores dos pulsos PWM por meio do método “setPWM()”. O valor específico de 350 para acionar o seletor foi determinado empiricamente durante testes práticos.

4.5 DISPENSER

Até o momento, o único componente do projeto que não foi detalhado foi o dispenser, também desenvolvido através do *software* Autodesk Fusion e representado na Figura 4.19. Na

imagem, é possível observar tanto o dispenser montado quanto suas peças individuais para montagem. O dispenser foi elaborado para garantir que o manipulador robótico sempre colete um bloco em uma posição predefinida, simplificando o processo de manipulação. A Figura 4.20 ilustra o dispenser montado, pintado e já com os blocos.

Figura 4.19 – Dispenser montado e desmontado.



Fonte: O autor.

Figura 4.20 – Dispenser com blocos no projeto final.



Fonte: O autor.

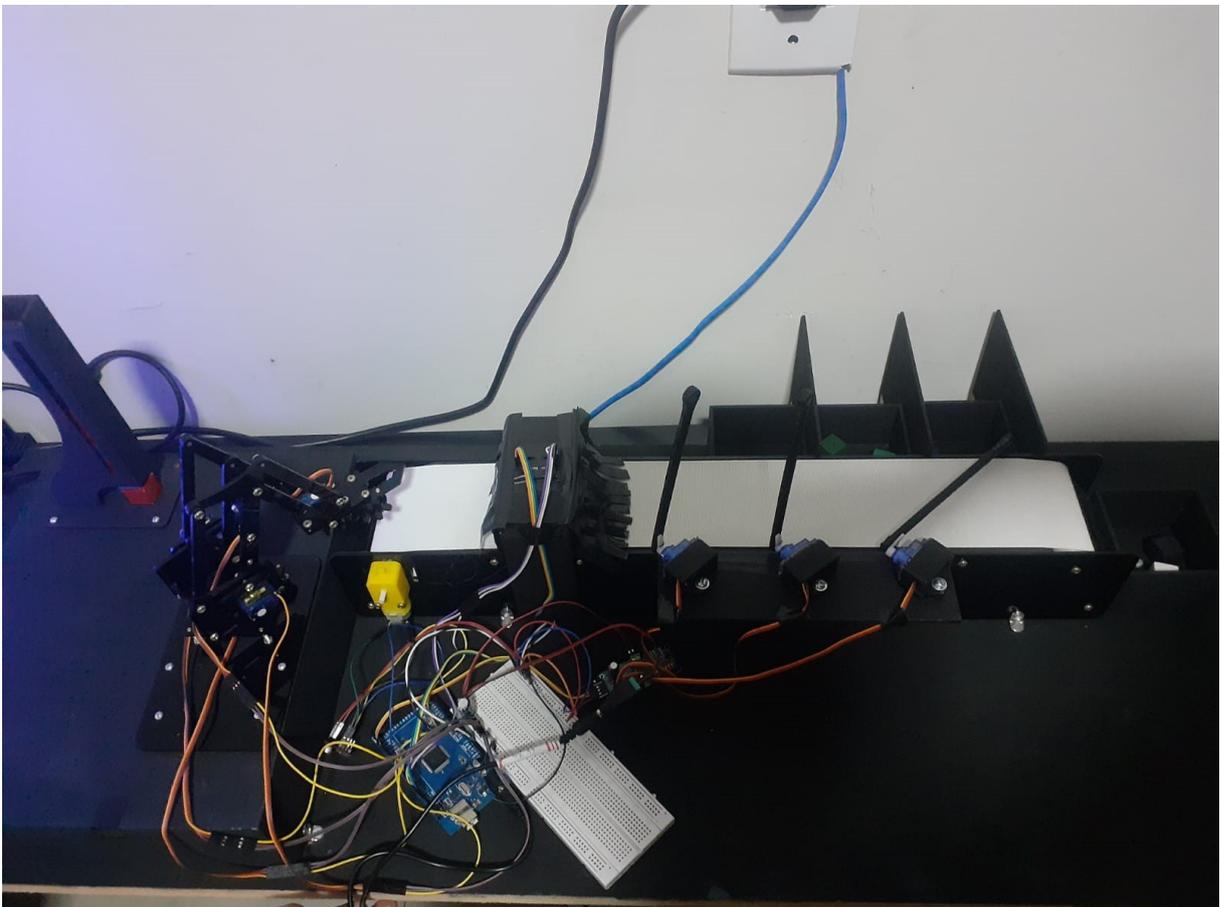
A dimensão interna do dispenser é um pouco maior do que a do bloco, permitindo que os blocos sejam facilmente depositados nele. Dessa forma, quando o manipulador robótico pegar o bloco inferior, os outros deslizam para suas próximas posições. O dispenser tem capacidade máxima para onze blocos, no entanto, é recomendado utilizar apenas seis para

minimizar o atrito entre o bloco que o manipulador manipula com o bloco acima dele, evitando assim resistência excessiva.

4.6 CONSIDERAÇÕES FINAIS

Após o término do desenvolvimento de cada componente principal do projeto, os circuitos de cada parte foram interligados e um único código foi desenvolvido para abranger todo o projeto. O código final está disponível no final do documento em anexo. Além disso, a seguir é mostrado o projeto finalizado, apresentando os demais componentes inclusos, como o dispenser, recipientes, as barreiras dos recipientes, o suporte para o manipulador, entre outros, conforme ilustrado na Figura 4.21.

Figura 4.21 – Circuito do projeto final.



Fonte: O autor.

O circuito final do projeto segue de perto o esquema do manipulador robótico apresentado na Figura 4.6, mantendo a conexão entre o GND do Arduino e o GND da fonte externa. Além disso, foram adicionadas as mesmas conexões para o sensor de cor TCS230, a Mini Ponte H e o módulo PCA9685 no Arduino. É importante notar que o sensor de cor, a Mini Ponte H e a alimentação do chip do módulo PCA9685 estão sendo alimentados em paralelo

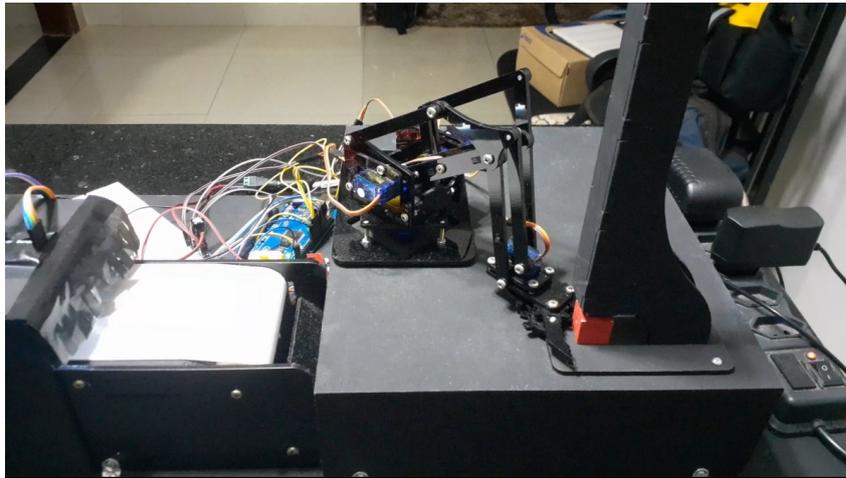
pelo Arduino. Enquanto isso, o manipulador robótico, a alimentação dos servos do módulo PCA9586 e o capacitor estão sendo alimentados pela fonte externa de 6V e 3A, também em paralelo.

Após a conclusão da montagem integral do projeto, foi fixado o suporte do manipulador e do dispenser, assim como a esteira transportadora e os recipientes, em uma placa de MDF de 1,8cm de espessura. Essa decisão foi tomada visando proporcionar uma base sólida e estável para o sistema como um todo, garantindo um desempenho consistente e confiável ao longo de sua operação.

5 RESULTADOS

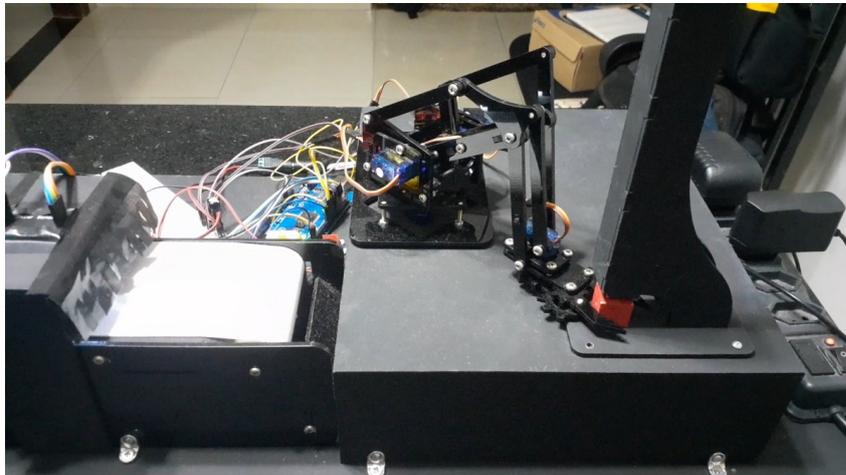
O protótipo finalizado executa uma ordem de tarefas repetitivas. Primeiramente, o manipulador robótico é posicionado para agarrar o bloco mais abaixo do dispenser (Figura 5.1). Em seguida, a garra é acionada para segurar firmemente o bloco selecionado (Figura 5.2).

Figura 5.1 – Manipulador robótico antes de pegar um bloco.



Fonte: O autor.

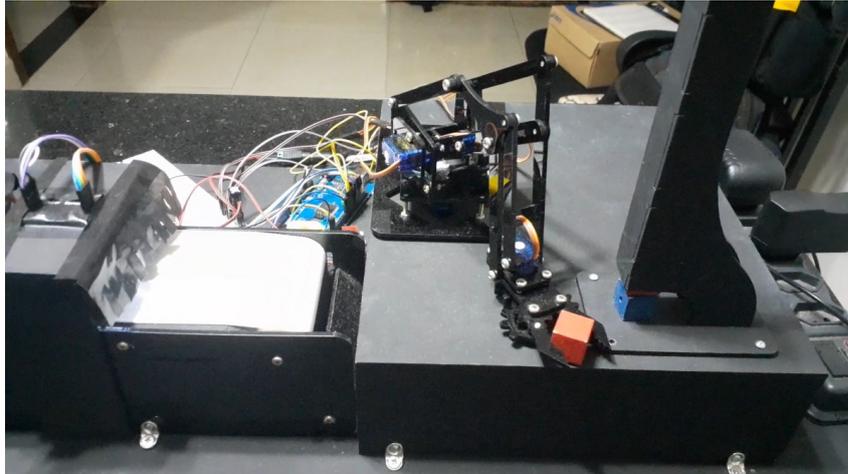
Figura 5.2 – Manipulador robótico agarrando um bloco.



Fonte: O autor.

Assim que o bloco é retirado do dispenser, o bloco de cima desliza para baixo, acompanhado pelos demais blocos, mantendo a continuidade do processo (Figura 5.3).

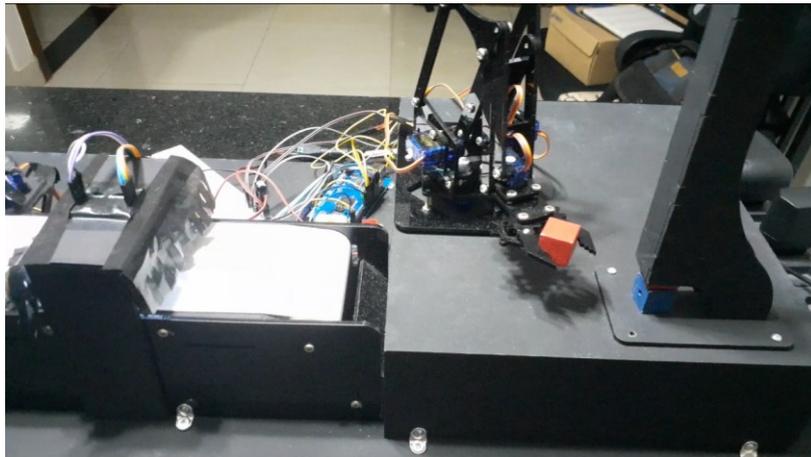
Figura 5.3 – Manipulador robótico tirando um bloco do dispenser.



Fonte: O autor.

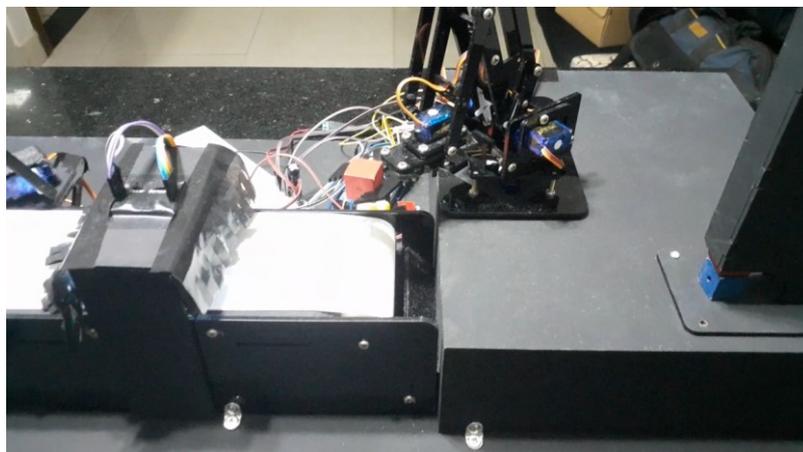
O manipulador, então, eleva o bloco capturado (Figura 5.4) e o direciona para a esteira (Figura 5.5), ajustando sua posição para a transferência e aproximando da esteira (Figura 5.6 e Figura 5.7).

Figura 5.4 – Manipulador robótico levantando um bloco.



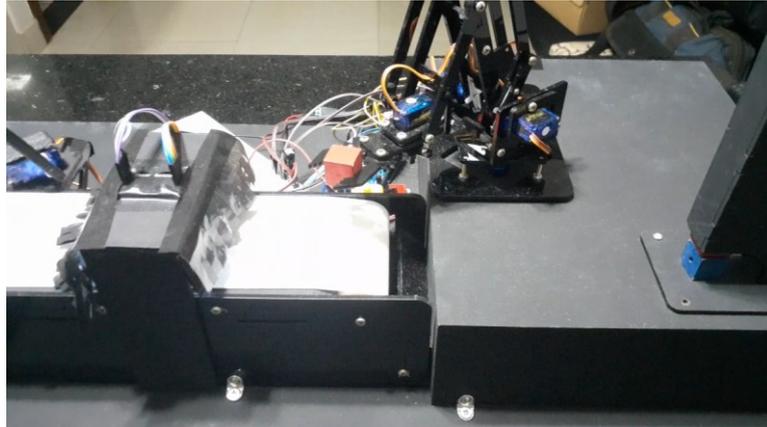
Fonte: O autor.

Figura 5.5 – Manipulador robótico direcionando um bloco para esteira.



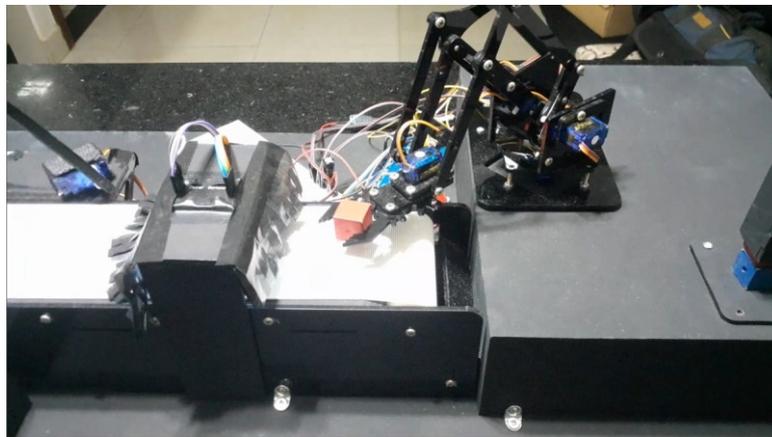
Fonte: O autor.

Figura 5.6 – Manipulador robótico ajustando sua posição para depositar um bloco na esteira.



Fonte: O autor.

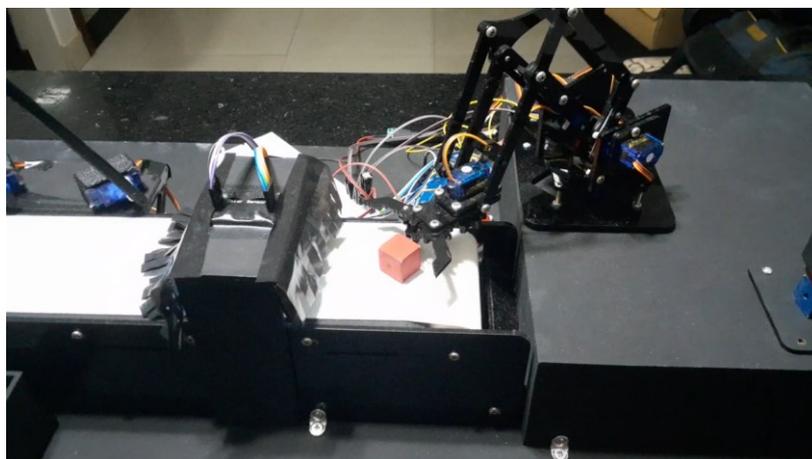
Figura 5.7 – Manipulador robótico aproximando da esteira.



Fonte: O autor.

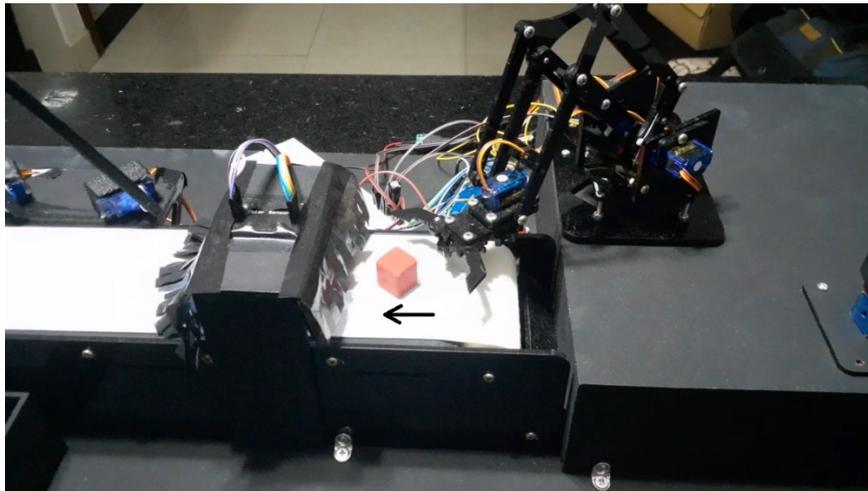
Com precisão, o bloco é depositado na esteira (Figura 5.8), que é ativada para deslocá-lo abaixo do sensor de cor (Figura 5.9). O sensor realiza a leitura da cor do bloco (Figura 5.10), e com base nessa informação, o seletor correspondente é acionado, religando a esteira (Figura 5.11).

Figura 5.8 – Manipulador robótico soltando um bloco na esteira.



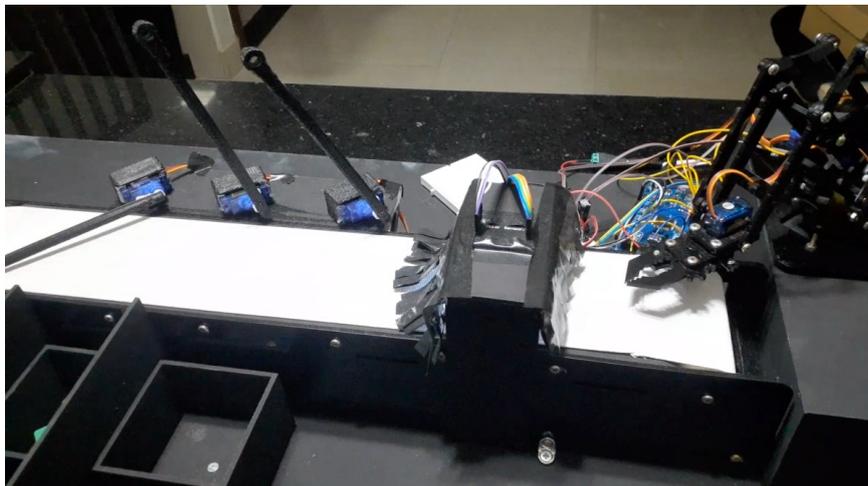
Fonte: O autor.

Figura 5.9 – Bloco movimentando em direção do sensor de cor.



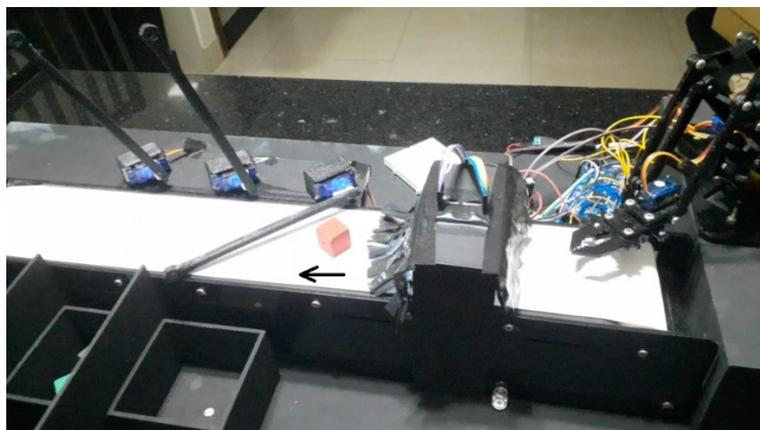
Fonte: O autor.

Figura 5.10 – Sensor de cor realizando a leitura da cor do bloco.



Fonte: O autor.

Figura 5.11 – Seletor para cor vermelha acionado e bloco deslocando em direção a primeira haste.

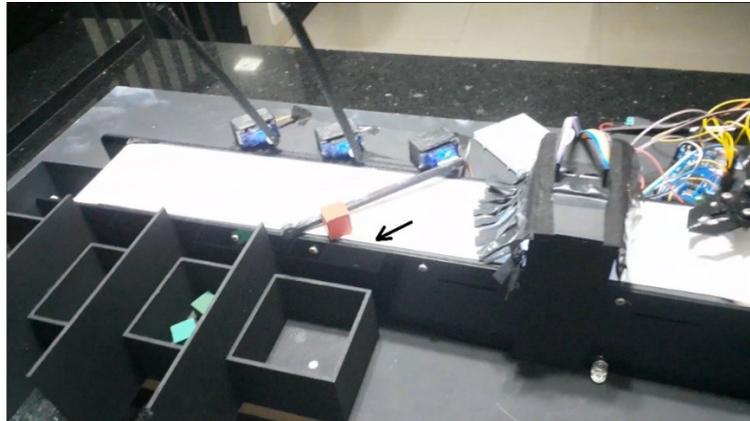


Fonte: O autor.

O bloco é então direcionado pela haste para o recipiente adequado (Figura 5.12), onde a separação é realizada de acordo com a cor detectada (Figura 5.13). Este fluxo contínuo de

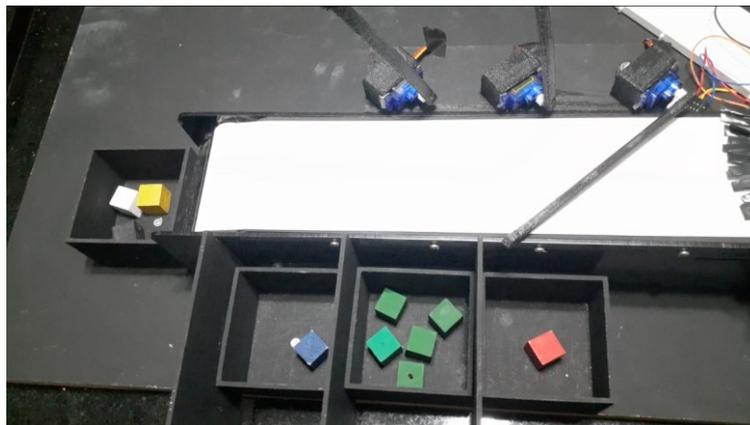
operações garante uma seleção e separação eficaz dos blocos, demonstrando a funcionalidade completa e automatizada do protótipo.

Figura 5.12 – Haste acionada com a esteira ligada direcionando bloco vermelho para primeiro recipiente.



Fonte: O autor.

Figura 5.13 – Bloco vermelho separado no primeiro recipiente.



Fonte: O autor.

O protótipo concluído demonstrou eficácia ao manipular os blocos do dispenser e encaminhá-los para a esteira transportadora. No entanto, há uma inconsistência do dispenser, ocasionando quedas dos blocos em posições ligeiramente desalinhadas. Além disso, a presença de outros blocos acima daqueles que o manipulador tenta retirar gera resistência, interferindo um pouco no processo.

Para avaliar a precisão do projeto, foram conduzidos cinco séries de testes, cada uma composta por 20 blocos, conforme detalhado na Tabela 5.1. Os resultados revelaram que o primeiro e o quarto teste atingiram uma taxa de sucesso de 100%, enquanto o segundo, terceiro e quinto testes apresentaram uma taxa de acerto de 95%, representando um único erro em cada série. Assim, calculou-se a média desses resultados, alcançando uma precisão geral de 97% para o protótipo.

Tabela 5.1 – Resultado dos testes no protótipo.

Série	Acertos	Percentual de Acerto
1ª Série	20 acertos em 20 testes	100%
2ª Série	19 acertos em 20 testes	95%
3ª Série	19 acertos em 20 testes	95%
4ª Série	20 acertos em 20 testes	100%
5ª Série	19 acertos em 20 testes	95%
Média	97 acertos em 100 testes	97%

Fonte: O autor.

Durante os testes, observou-se que o dispenser apresenta limitações quanto à quantidade de blocos posicionados acima daquele que será manipulado pelo manipulador robótico. Desta forma, por meio de testes, concluiu-se que o número ideal de blocos no dispenser é de aproximadamente seis. Portanto, para manter um funcionamento eficiente do sistema, minimizando a resistência mecânica, bem como prevenindo desalinhamentos após a remoção dos blocos, é aconselhável reabastecer o dispenser à medida que o manipulador os remove, mantendo sempre seis unidades disponíveis. A avaliação do projeto foi conduzida usando um dispenser com seis blocos, garantindo que, à medida que o manipulador retirava um bloco do dispenser, outro era imediatamente reposicionado.

Além disso, parte da margem de erro na transferência dos blocos para a esteira transportadora está relacionada à limitação do manipulador robótico em aplicar uma pressão adequada sobre o servo da garra. Isso resulta em uma aderência insuficiente entre o bloco e a garra, aumentando a possibilidade de queda do bloco. Uma sugestão para melhorar esse aspecto no futuro seria incorporar um mecanismo adicional à garra, como aplicação de elásticos ou a fixação de pequenos pedaços de EVA, proporcionando uma aderência mais eficaz.

Ao realizar os testes com o manipulador robótico no começo do projeto, percebeu-se que os movimentos para pegar os blocos eram muito bruscos. Portanto, uma nova função foi desenvolvida e implementada no código para permitir uma manipulação mais suave e controlada dos blocos. Além disso, durante o processo de deposição, surgiram dificuldades com mal contato tanto no manipulador robótico quanto na esteira. Um mal contato no manipulador resultava em um posicionamento descentralizado do bloco na esteira, comprometendo sua correta leitura pelo sensor de cor.

Da mesma forma, um mal contato na esteira afetava o posicionamento do bloco em relação ao sensor de cor, prejudicando a leitura exata da sua cor e aumentando o risco de erros. A maioria dos erros na leitura da cor dos blocos vermelhos e azuis são devidos a mal posicionamento abaixo do sensor de cor.

Por outro lado, durante o processo de deposição, as vezes ocorrem variações na velocidade da esteira devido ao atrito entre as correias e as placas de MDF. Estas placas aumentaram significativamente o atrito, especialmente na junção da correia. Para enfrentar este desafio, foi necessário polir as extremidades do MDF, assegurando um funcionamento mais eficiente da esteira e reduzindo o desgaste excessivo.

Após o início do projeto, logo após a montagem completa do manipulador robótico, também foi enfrentado dificuldades com seu movimento. Foi notado que havia uma resistência significativa nas hastes, ocasionada pelo aperto excessivo de um parafuso. Após afrouxá-lo para o ajuste adequado, o manipulador passou a funcionar melhor. Essa experiência destacou a importância de garantir o aperto adequado de seus parafusos, evitando apertar muito ou deixar frouxo.

Além disso, houve um desafio extra com o manipulador robótico relacionado à necessidade de estabilizar seus movimentos. Para resolver este problema, uma ponta de um jumper foi fixada na base do manipulador robótico, diminuindo seus movimentos irregulares. Também, para simplificar o processo de deposição dos blocos na esteira, foi utilizado um suporte para elevar o manipulador, facilitando assim a sua operação e depositando os blocos na esteira com maior facilidade.

Outra observação relevante foi a divergência nos resultados da leitura das cores em ambientes distintos. Essa diferença pode ser atribuída ao sensor de cor captar informações que não correspondem ao bloco, ou pela interferência da iluminação ambiente sobre a leitura. Para mitigar este problema, foi colocado uma pequena cortina para escurecer o ambiente de leitura, além do uso da case para proteger o sensor de interferências externas, conforme explicado no capítulo anterior.

O projeto oferece uma perspectiva abrangente sobre o processo de automação industrial, proporcionando uma plataforma para demonstrações práticas, permitindo uma compreensão mais tangível de conceitos abstratos de robótica. Desta forma, o projeto facilita a transição dos aspectos teóricos para a prática, simplificando e facilitando a assimilação dos conceitos para outros estudantes. Além disso, ao evidenciar a importância da automação nos processos industriais e destacar os princípios fundamentais de eletrônica e programação, os resultados do projeto têm o potencial de promover uma compreensão mais ampla e profunda desses temas essenciais.

6 CONCLUSÃO

Em conclusão, os objetivos propostos foram alcançados de forma satisfatória durante o desenvolvimento do projeto do manipulador robótico automatizado com uma esteira seletora de blocos por cor. A implementação da transferência dos blocos por meio do manipulador robótico, juntamente com a construção da esteira e a integração do processo de seleção baseado na coloração dos blocos, evidencia o cumprimento dos objetivos específicos estabelecidos.

A realização desses objetivos resultou na demonstração eficaz de conceitos fundamentais de automação, eletrônica e programação, conforme planejado inicialmente. Além disso, a avaliação do desempenho do protótipo confirmou sua funcionalidade na separação automática de blocos com base em suas cores, bem como uma ótima taxa de acerto de 97% na separação das cores vermelho e azul.

A principal limitação identificada durante a implementação do projeto foi a dificuldade em separar a cor verde e as cores secundárias. Desta forma, como sugestão para pesquisas futuras, é proposto um estudo mais aprofundado para desenvolver um método mais preciso de separação dessas cores no protótipo. Isso poderia envolver a exploração de algoritmos de processamento de imagens mais avançados ou a incorporação de sensores adicionais, como infravermelho ou ultrassom. Essas linhas de investigação têm o potencial de melhorar significativamente a funcionalidade e o desempenho do protótipo, expandindo suas capacidades e aplicabilidades.

Desta forma, o projeto trouxe contribuições significativas para o curso de engenharia eletrônica e de telecomunicações, oferecendo aos alunos a chance de aplicar teorias aprendidas em aula de forma prática e realista. Além disso, atua como uma ferramenta educacional valiosa, demonstrando conceitos essenciais em automação, robótica, eletrônica e programação.

REFERÊNCIAS

- 1 QUEIROZ, Mariana Emer de. **ELABORAÇÃO E IMPLEMENTAÇÃO DE UM PROTÓTIPO DE ESTEIRA SELETORA POR COR**. 2017. 74 f. TCC (Graduação) - Curso de Engenharia Eletrônica, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná, Campo Mourão, 2017. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/6082/2/prototipoesteiraseletoracor.pdf>. Acesso em: 1 de jun. 2022.
- 2 SHRESTHA, Abhishek; KARKI, Nishan; YONJAN, Rupesh; SUBEDI, Monica; PHUYAL, Sudip. Automatic Object Detection and Separation for Industrial Process Automation. **2020 Ieee International Students' Conference On Electrical, Electronics And Computer Science**. Bhopal, Índia, p. 1-1. 07 de maio de 2020. Disponível em: <https://ieeexplore-ieee.org.ez34.periodicos.capes.gov.br/document/9087035>. Acesso em: 1 de jun. 2022.
- 3 S.KRISHNAKUMAR; SNEHA, K.; A.REETHIKA. Review on Sensor based Colour Sorting Robot for Candy Manufacturing. **Iop Conference Series: Materials Science And Engineering**. Sathy, Tamil Nadu, Índia, p. 1-1. jan. 2021. Disponível em: <https://iopscience.iop.org/article/10.1088/1757-899X/1084/1/012094/pdf>. Acesso em: 1 jun. 2022.
- 4 THIKE, Aung; SAN, Zin Zin Moe; OO, Dr. Zaw Min. Design and Development of an Automatic Color Sorting Machine on Belt Conveyor. **International Journal Of Science And Engineering Applications**. Thanlyin, Myanmar, p. 176-179. 2019. Disponível em: <https://ijsea.com/archive/volume8/issue7/IJSEA08071002.pdf>. Acesso em: 1 jun. 2022.
- 5 KOMOL, Md. Mostafizur Rahman; PODDER, Amit Kumer. Design and Construction of Product Separating Conveyor based on Color. **2017 3rd International Conference On Electrical Information And Communication Technology (Eict)**. Khulna, Bangladesh, p. 1-1. dez. 2017. Disponível em: <https://ieeexplore-ieee.org.ez34.periodicos.capes.gov.br/document/8275163>. Acesso em: 1 jun. 2022.
- 6 YAMAMOTO, Allan Cesar Seiji; SILVA, Caio Felipe Drago da; RIBEIRO, João Marcio de Oliveira; NASCIMENTO, Marcelo Rezende do; CESARE, Rômulo Augusto. **Esteira Seletora e Contadora de Objetos Metálicos**. 2014. 44 f. TCC (Graduação) - Curso de Técnico em Automação Industrial, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Campus São José dos Campos, 2014. Disponível em: https://ifsp-sjc3.webnode.com/_files/200000315-993d69b31b/Esteira%20Seletora%20e%20Contadora%20de%20Objetos%20Met%C3%A1licos.pdf. Acesso em: 1 jun. 2022.
- 7 KHAIRUDIN, Ahmad Rafiq Mohd; KARIM, Muhammad Hanif Abdul; SAMAH, Asma Abu; IRWANSYAH, Dyanatasya; YAKOB, Muhammad Yunus; ZIAN, Norhaslina Mat. Development of Colour Sorting Robotic Arm Using TCS3200 Sensor. **2021 Ieee 9Th Conference On Systems, Process And Control (ICSPc 2021)**. Malacca, Malaysia, p. 108-112. 4 fev. 2022. Disponível em: <https://ieeexplore-ieee.org.ez34.periodicos.capes.gov.br/stamp/stamp.jsp?tp=&arnumber=9689114>. Acesso em: 1 jun. 2022.

8 GÜNTÜRKÜN, Rüştü; HIZ, Osman; ŞAHİN, Hasan. Design and Application of Plc Controlled Robotic Arm Choosing Objects According to Their Color. **Electronic Letters On Science & Engineering**. Konya / Turkey, p. 52-52. 30 dez. 2020. Disponível em: <https://dergipark.org.tr/en/download/article-file/1270917>. Acesso em: 1 jun. 2022.

9 PANIE, Gregorio Imanuel Efraim; MUTIARA, Achmad Benny. Development of Robotic Arm for Color Based Goods Sorter in Factory Using TCS3200 Sensor with a Web-Based Monitoring System. **2018 Third International Conference On Informatics And Computing (Icic)**. Palembang, Indonesia, p. 1-6. 1 ago. 2019. Disponível em: <https://ieeexplore-ieee.org.ez34.periodicos.capes.gov.br/document/8780461/authors>. Acesso em: 1 jun. 2022.

10 DEZAKI, Mohammadreza Lalegani; HATAMI, Saghi; ZOLFAGHARIAN, Ali; BODAGHI, Mahdi. A pneumatic conveyor robot for color detection and sorting. **Cognitive Robotics**. Nottingham, Uk, p. 60-71. 12 mar. 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667241322000040>. Acesso em: 1 jun. 2022.

11 ESTEIRAS transportadoras: quais são os tipos e para que servem? Siembra Automação. Disponível em: <https://www.siembra.com.br/noticias/esteiras-transportadoras-quais-sao-os-tipos-e-para-que-servem/>. Acesso em: 1 nov. 2022.

12 TUDO o que você precisa saber sobre esteiras transportadoras. 2020. COBRA Esteiras e correntes transportadoras. Disponível em: <https://www.cobra.ind.br/blog/tecnologia/tudo-o-que-voce-precisa-saber-sobre-esteiras-transportadoras>. Acesso em: 1 nov. 2022.

13 CONHEÇA os tipos de esteiras transportadoras para indústrias. Daido Industrial e Comercial Ltda. Disponível em: <https://daido.com.br/industrial/conheca-os-tipos-de-esteiras-transportadoras-para-industrias/>. Acesso em: 1 nov. 2022.

14 COMO funciona as esteiras transportadoras industriais. Daido Industrial e Comercial Ltda. Disponível em: <https://daido.com.br/industrial/esteiras-transportadoras/>. Acesso em: 1 nov. 2022.

15 LOS componentes de: Las bandas transportadoras. 2018. Vcrobotica. Disponível em: <https://www.vcrobotica.com/home/los-componentes-de-las-bandas-transportadoras/>. Acesso em: 1 nov. 2022.

16 ATUADORES – O que são e para o que servem. 2022. Nepin Acessórios Industriais. Disponível em: <https://www.nepin.com.br/blog/solucoes-industriais/atuadores-o-que-sao-e-para-o-que-servem/>. Acesso em: 1 nov. 2022.

17 O QUE é um atuador? 2020. Unival Válvulas e conexões. Disponível em: <https://www.unival.com.br/2020/11/25/o-que-e-um-atuador/>. Acesso em: 1 nov. 2022.

18 ATUADOR elétrico: o que é, vantagens, tipos e funções. Kalatec Automação. Disponível em: <https://blog.kalatec.com.br/atuador-eletrico/>. Acesso em: 1 nov. 2022.

19 SIMPLÍCIO, Paulo Victor Galvão; LIMA, Beatriz Rêgo; JUNKES, Janaína Accordi. Manipuladores Robóticos Industriais. **Caderno de Graduação - Ciências Exatas e**

Tecnológicas - **Unit**. Aracaju, p. 85-94. out. 2016. Disponível em: <https://periodicos.set.edu.br/cadernoexatas/article/view/3572/1950>. Acesso em: 10 nov. 2022.

20 O BRAÇO robótico industrial ganha espaço no armazém. 2021. Mecalux. Disponível em: <https://www.mecalux.com.br/blog/braco-robotico-industrial>. Acesso em: 10 nov. 2022.

21 COCOTA JÚNIOR, José Alberto Naves. **Elementos de Robótica**. Disponível em: <http://professor.ufop.br/sites/default/files/cocota/files/elemroboticaintro.pdf>. Acesso em: 10 nov. 2022.

22 'DEGREES of Freedom' VS 'Functions' of a Robotic Arm. 2021. Reach Robotics. Disponível em: [https://reachrobotics.com/blog/degrees-of-freedom-vs-functions-of-a-robotic-arm/#:~:text=A%20Degree%20of%20Freedom%20\(DoF\)%20as%20it%20relates,a%20Single%20Degree%20of%20Freedom](https://reachrobotics.com/blog/degrees-of-freedom-vs-functions-of-a-robotic-arm/#:~:text=A%20Degree%20of%20Freedom%20(DoF)%20as%20it%20relates,a%20Single%20Degree%20of%20Freedom). Acesso em: 17 nov. 2022.

23 PROGRAMMED Robot Arm. Instructables. Disponível em: <https://www.instructables.com/Programmed-Robot-Arm/>. Acesso em: 17 nov. 2022.

24 FORWARD and Inverse Kinematic model for the Manipulator. Coursehero. Disponível em: <https://www.coursehero.com/tutors-problems/Electronics/29816802-Please-refer-to-the-attachment-to-answer-this-question-This-question/>. Acesso em: 17 nov. 2022.

25 HIWONDER Metal Picking Gripper for Robotic Arm and Robot DIY. Disponível em: <https://hiwonder.hk/products/hiwonder-metal-picking-gripper-for-robotic-arm-and-robot-diy>. Acesso em: 17 nov. 2022.

26 THOMSEN, Adilson. **O que é Arduino, para que serve e primeiros passos**. Maker Hero. Disponível em: <https://www.makerhero.com/blog/o-que-e-Arduino/>. Acesso em: 17 mar. 2024.

27 CIRIACO, Douglas. **O que é Arduino?** 2015. Canaltech. Disponível em: <https://canaltech.com.br/hardware/o-que-e-Arduino/>. Acesso em: 27 out. 2022.

28 ARDUINO ou Raspberry Pi? Saiba qual micro PC é melhor para seu projeto. 2015. Techtudo. Disponível em: <https://www.techtudo.com.br/noticias/2015/04/Arduino-ou-raspberry-pi-saiba-qual-micro-pc-e-melhor-para-seu-projeto.ghtml>. Acesso em: 31 out. 2022.

29 LOUSADA, Ricardo. **O que é Arduino: Para que Serve, Vantagens e como Utilizar**. 2020. Blog Eletrogate. Disponível em: <https://blog.eletrogate.com/o-que-e-Arduino-para-que-serve-vantagens-e-como-utilizar/>. Acesso em: 31 out. 2022.

30 WENDLING, Marcelo. **Sensores**. 2010. Universidade Estadual Paulista UNESP. Disponível em: <https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>. Acesso em: 20 out. 2022.

31 PRABHU, Vidya. **WHAT IS A SENSOR?** 2021. YoungWonks. Disponível em: <https://www.youngwonks.com/blog/What-is-a-sensor>. Acesso em: 20 out. 2022.

32 MATTEDE, Henrique. **O que são sensores e quais as suas aplicações?** Disponível em: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>. Acesso em: 20 out. 2022.

- 33 COLOR Sensor. ROHM Semiconductor. Disponível em: <https://www.rohm.com/electronics-basics/sensor/color-sensor>. Acesso em: 20 out. 2022.
- 34 WHAT is a Color sensor? Sensor Basics: Introductory Guide to Sensors. Sensor Basics: Introductory Guide to Sensors. KEYENCE. Disponível em: <https://www.keyence.com/ss/products/sensor/sensorbasics/color/info/#:~:text=A%20color%20sensor%20is%20a,detection%20object%20with%20a%20receiver>. Acesso em: 20 out. 2022.
- 35 TEXAS Advanced Optoelectronic Solutions, **TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER**, 2004. Disponível em: <http://bonavolta.ch/hobby/files/TCS230.pdf>. Acesso em: 25 de out. 2022.
- 36 CASTRO, Giovanni de; CASSIOLI, Matheus. **Usando o Sensor de Cor TCS230**. Robo Core. Disponível em: https://www.robocore.net/tutoriais/usando-sensor-de-cortcs230?gclid=Cj0KCQjwkt6aBhDKARIsAAyeLJ2Jux4DjIW9LLxQ0pdozcXzuYVm6dlqCNhzxyXKIIvxlr2ekQW1CD0aAvtCEALw_wcB. Acesso em: 20 out. 2022.
- 37 ROBOCORE. Micro Servo 9g SG90. Disponível em: <https://www.robocore.net/servo-motor/micro-servo-9g-sg90-towerpro>. Acesso em: 04 abr. 2024.
- 38 JAMECO. How Servo Motors Work | Servo Motor Controllers. Disponível em: <https://www.jameco.com/jameco/workshop/howitworks/how-servo-motors-work.html>. Acesso em: 04 abr. 2024.
- 39 Mini Ponte H Dupla L298N. Disponível em: <https://www.eletragate.com/mini-ponte-h-dupla-l298n>. Acesso em: 04 abr. 2024.
- 40 Mini Ponte H L298N: como usar com Arduino e sem Arduino. Disponível em: https://www.youtube.com/watch?v=asyDvC_Sylc. Acesso em 04 abr. 2024.
- 41 SMARTKITS. Módulo PCA9685 – Controlador PWM 16 Canais 12 bits para Servo Motor. Disponível em: <https://www.smartkits.com.br/modulo-pca9685-controlador-pwm-16-canais>. Acesso em: 5 abr. 2024.
- 42 Super módulo para Arduino! Controla 16 servos com dois fios no Arduino! Disponível em: <https://www.youtube.com/watch?v=xmM1sl7oRcl&t=248s>. Acesso em 5 abr. 2024.
- 43 STROSKI, P. N. PCA9685: Como usar no Arduino? Disponível em: <https://www.electricalibrary.com/2020/07/25/pca9685-como-usar-no-arduino/>. Acesso em 5 abr. 2024.

ANEXO

```

#include <Servo.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver seletor = Adafruit_PWMServoDriver();

#define servoMIN 150
#define servoMAX 600

Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

Servo seletor1;
Servo seletor2;
Servo seletor3;

const int S0 = 8;
const int S1 = 9;
const int S2 = 11;
const int S3 = 12;
const int OUT = 10;

const int IN1 = 2;
const int IN2 = 3;

int VERMELHO = 0;
int VERDE = 0;
int AZUL = 0;

void moverSuavemente();
void ler_cor();

void setup() {
  servo1.attach(4); // Servo da haste maior (Movimenta a garra para frente ou para trás)
  servo2.attach(5); // Servo da garra
  servo3.attach(6); // Servo da base
  servo4.attach(7); // Servo da haste menor (Movimenta a garra para cima ou para baixo)

  // Posição inicial do manipulador robótico
  moverSuavemente(servo1, 78);
  moverSuavemente(servo2, 150);
  moverSuavemente(servo3, 103);
  moverSuavemente(servo4, 137);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);

  // Começando com o motor parado
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);

  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);

  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

```

```

Serial.begin(9600);
seletor.begin();
seletor.setPWMFreq(60);

// Deixar os seletores nas posicoes iniciais (Nao acionados)
seletor.setPWM(0, 0, servoMIN);
seletor.setPWM(1, 0, servoMIN);
seletor.setPWM(2, 0, servoMIN);
}

void loop() {

  moverSuavemente(servo3, 154); // Virar o manipulador
  delay(50);
  moverSuavemente(servo2, 57); // Abrir a garra
  delay(50);
  moverSuavemente(servo1, 103); // Aproximar o bloco
  delay(50);
  moverSuavemente(servo4, 110); // Ajustar posicao
  delay(50);
  moverSuavemente(servo1, 147); // Aproximar mais
  delay(50);
  moverSuavemente(servo2, 130); // Fechar a garra
  delay(50);
  moverSuavemente(servo3, 135); // Tirar bloco do dispenser
  moverSuavemente(servo1, 110); // Tirar bloco do dispenser
  delay(50);
  moverSuavemente(servo3, 44); // Termina de virar
  delay(50);
  moverSuavemente(servo4, 120); // Aproximar da esteira
  delay(50);
  moverSuavemente(servo1, 135); // Aproximar da esteira
  delay(50);
  moverSuavemente(servo2, 33); // Abrir garra
  delay(50);
  // Girar o motor no sentido horario
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  // Aguardar 425ms e parar o motor
  delay(425);
  digitalWrite(IN1, LOW);
  moverSuavemente(servo2, 150); // Fechar a garra
  delay(50);
  // Fazer a leitura da cor
  ler_cor();
  delay(500); // Delay para garantir que foi feita a leitura do bloco
  Serial.println("Valor do vermelho:");
  Serial.println(VERMELHO);
  Serial.println("Valor do verde:");
  Serial.println(VERDE);
  Serial.println("Valor do azul:");
  Serial.println(AZUL);

  if (VERMELHO < AZUL && VERMELHO < VERDE) {
    Serial.println("***** VERMELHO ACIONADO *****");
    // Acionar o seletor vermelho
    seletor.setPWM(1, 0, servoMIN); // Seletor nao acionado
    seletor.setPWM(2, 0, servoMIN); // Seletor nao acionado
    seletor.setPWM(0, 0, 350); // Seletor acionado
    // Ligar a esteira
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    // Aguardar quatro segundos e parar o motor
    delay(4000);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
  }
}

```

```

if (VERDE < VERMELHO && VERDE < AZUL) {
  Serial.println("***** VERDE ACIONADO *****");
  // Acionar o seletor verde
  seletor.setPWM(0, 0, servoMIN); // Seletor nao acionado
  seletor.setPWM(2, 0, servoMIN); // Seletor nao acionado
  seletor.setPWM(1, 0, 350); // Seletor acionado
  // Ligar a esteira
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  // Aguardar quatro segundos e parar o motor
  delay(4000);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
}

if (AZUL < VERMELHO && AZUL < VERDE) {
  Serial.println("***** AZUL ACIONADO *****");
  // Acionar o seletor azul
  seletor.setPWM(0, 0, servoMIN); // Seletor nao acionado
  seletor.setPWM(1, 0, servoMIN); // Seletor nao acionado
  seletor.setPWM(2, 0, 350); // Seletor acionado
  // Ligar a esteira
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  // Aguardar quatro segundos e parar o motor
  delay(4000);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
}

moverSuavemente(servo1, 97);
moverSuavemente(servo4, 124);
delay(50);
}

// Função para movimentar o servo suavemente
void moverSuavemente(Servo servo, int posicao) {
  int posicaoAtual = servo.read();
  int passo = (posicaoAtual < posicao) ? 1 : -1;

  for (int i = posicaoAtual; i != posicao; i += passo) {
    servo.write(i);
    delay(10);
  }
  servo.write(posicao);
  delay(500);
}

// Função para fazer a leitura das cores
void ler_cor() {

  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);

  if(digitalRead(OUT) == HIGH) {
    VERMELHO = pulseIn(OUT, LOW);
  } else {
    VERMELHO = pulseIn(OUT, HIGH);
  }

  digitalWrite(S3, HIGH);

  if(digitalRead(OUT) == HIGH) {
    AZUL = pulseIn(OUT, LOW);
  } else {
    AZUL = pulseIn(OUT, HIGH);
  }
}

```

```
}  
  
digitalWrite(S2, HIGH);  
  
if(digitalRead(OUT) == HIGH) {  
  VERDE = pulseIn(OUT, LOW);  
} else {  
  VERDE = pulseIn(OUT, HIGH);  
}  
  
}
```