



**UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
INSTITUTO DE ENGENHARIA E GEOCIÊNCIAS
PROGRAMA EM CIÊNCIA E TECNOLOGIA
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA**

JOSIELE LIMA BARRETO

**DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA VIA *APP DESIGNER*
PARA CALIBRAÇÃO DE UM SENSOR TCS230 MICROCONTROLADO POR
ARDUINO**

SANTARÉM-PA

2023

JOSIELE LIMA BARRETO

**DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA VIA *APP DESIGNER*
PARA CALIBRAÇÃO DE UM SENSOR TCS230 MICROCONTROLADO POR
ARDUINO**

Trabalho de Conclusão de Curso – TCC
apresentado ao Programa em Ciência e
Tecnologia, para obtenção do grau de
Bacharel em Interdisciplinar em Ciência e
Tecnologia; Universidade Federal do
Oeste do Pará, Instituto de Engenharia e
Geociências.

Orientador: Prof. Me. Gilson Fernandes
Braga Jr.

SANTARÉM-PA

2023

JOSIELE LIMA BARRETO

**DESENVOLVIMENTO DE UMA INTERFACE GRÁFICA VIA *APP DESIGNER*
PARA CALIBRAÇÃO DE UM SENSOR TCS230 MICROCONTROLADO POR
ARDUINO**

Trabalho de Conclusão de Curso – TCC
apresentado ao Programa em Ciência e
Tecnologia, para obtenção do grau de
Bacharel em Interdisciplinar em Ciência e
Tecnologia; Universidade Federal do
Oeste do Pará, Instituto de Engenharia e
Geociências.

Data da aprovação: **Santarém-PA, 06 de julho de 2023.**

BANCA EXAMINADORA

Prof. Me. Gilson Fernandes Braga Junior
Orientador – Universidade Federal do Oeste do Pará (Ufopa)

Prof. Dr. Nelson de Souza Amorim
Universidade Federal do Oeste do Pará (Ufopa)

Profa. Dra. Paula Renatha Nunes da Silva
Universidade Federal do Oeste do Pará (Ufopa)

Aos meus pais, Luzinete e Josoel.

Aos meus avós, Eliene e Manoel.

Às minhas irmãs, Janiele e Daiane.

AGRADECIMENTOS

À Deus por ter me dado a oportunidade de fazer este curso e poder concluir o trabalho aqui apresentado.

Ao meu orientador Prof. Me. Gilson Fernandes Braga Junior, pela orientação, paciência e feedback que foram essenciais durante todo o processo de elaboração deste trabalho.

À minha amiga Maria Gorete, por toda ajuda e apoio, durante a jornada acadêmica, principalmente na realização deste trabalho.

Aos meus colegas de classe, agradeço pela colaboração, pelo apoio mútuo e pelas discussões enriquecedoras. Compartilhamos conhecimentos e experiências, e isso certamente influenciou a qualidade deste trabalho.

Aos meus familiares e amigos, meu profundo agradecimento pelo amor, apoio incondicional e encorajamento constante. Vocês estiveram ao meu lado durante toda a jornada acadêmica, me motivando e me lembrando da importância deste projeto. Seus incentivos foram a força que me impulsionou até a conclusão deste trabalho.

Por fim, gostaria de expressar minha gratidão a todos aqueles que, direta ou indiretamente, desejavam o sucesso deste trabalho. Seus conselhos e apoio moral foram inestimáveis e fizeram a diferença.

RESUMO

A calibração é fundamental para o funcionamento de um sensor, tendo em vista que é um componente bastante sensível e pode sofrer influências de um determinado tipo de processo onde está inserido. O presente trabalho tem como objetivo o desenvolvimento de um aplicativo desktop para auxiliar na calibração do sensor de cor TCS230, através do uso do software MATLAB® a partir da ferramenta *App Designer*, aplicado a um protótipo de uma máquina separadora de peças por cor com recebimento de dados através de uma placa Arduino. Foram realizados três testes diferentes, que envolviam o uso e o não uso do aplicativo, assim como a diferença dos aspectos de luz ambiente onde se encontrava o protótipo. Como resultado, foi elaborado um manual do usuário e através de um botão de calibração, foram obtidos intervalos de frequência de valores máximos e mínimos permitindo que o processo de calibração seja de forma rápida e eficaz. Portanto, o trabalho mostrou que a plataforma é eficiente e pode ser utilizado em diversas aplicações que tenham como foco principal realizar a identificação de cores por meio de um sensor do tipo TCS230.

Palavras-chave: Sensor de cor; Arduino; MATLAB; *App Designer*.

ABSTRACT

Calibration is essential for the operation of a sensor, since it is a very sensitive component and can suffer influences from a certain type of process where it is inserted. The present work aims to develop a desktop application to assist in the calibration of the TCS230 color sensor, using MATLAB® software from the App Designer tool, applied to a prototype of a color separator machine with data reception through an Arduino board. Three different tests were carried out, involving the use and non-use of the application, as well as the difference in the aspects of ambient light where the prototype was located. As a result, a user manual was prepared and through a calibration button, frequency ranges of maximum and minimum values were obtained allowing the calibration process to be quick and effective. Therefore, the work showed that the platform is efficient and can be used in several applications that have as main focus to perform color identification through a TCS230 type sensor.

Keywords: Color sensor; Arduino; MATLAB; App Designer.

LISTA DE FIGURAS

Figura 1 – Representação da reflexão em um objeto de cor vermelha.	18
Figura 2 – Matriz fotodiodos e grade de filtro no padrão Bayer.....	19
Figura 3 – Módulo sensor de cor RGB TCS230.....	20
Figura 4 – Sensor TCS3200.....	21
Figura 5 – Módulo TCS34725.	22
Figura 6 – Módulo sensor de gestos e cor RGB APDS-9930.....	23
Figura 7 – Matriz 8x8 de fotodiodos presente no TCS230	24
Figura 8 – Matriz de fotodiodos e fotodiodo verde.	25
Figura 9 – Combinação dos filtros e escala de frequência nos pinos S0 e S1.	25
Figura 10 – Combinação dos filtros e escala de frequência nos pinos S2 e S3.	26
Figura 11 – Vista superior de um sensor de cor TCS230.....	26
Figura 12 – Arduino Uno.	29
Figura 13 – Janela em branco do <i>app designer</i>	31
Figura 14 – Janela do <i>code view - app designer</i>	32
Figura 15 – Protótipo da máquina separadora de peças por cor.....	33
Figura 16 – Ligação do circuito principal.	34
Figura 17 – Diagrama da programação em Arduino.	36
Figura 18 – Janela da GUI desenvolvida para auxiliar na calibração do sensor cor.	37
Figura 19 – Seção 1: Conexão.....	38
Figura 20 – Seção 2: Configuração.....	38
Figura 21 – Painel 2: Dados.....	39
Figura 22 – Painel 3: Ambiente de apresentação dos dados de calibração.....	39
Figura 23 – Corpos de prova organizados em sequência de testes.....	40
Figura 24 – Interface Gráfica calibração da peça vermelha.	43
Figura 25 – Interface Gráfica calibração da peça verde.....	44
Figura 26 – Interface Gráfica calibração da peça azul.	44
Figura 27 – Interface Gráfica calibração da peça laranja.	45
Figura 28 – Interface Gráfica calibração da peça amarela.....	45
Figura 29 – Tela da interface do aplicativo mostra a não detecção de uma peça.	47
Figura 30 – Linha de código com intervalo de calibração da cor vermelha.....	47

LISTA DE TABELAS

Tabela 1 – Comparação de módulos de sensores de cor mais populares.....	23
Tabela 2 – Comparação de servomotores mais populares.	27
Tabela 3 – Intervalos obtidos com primeira análise de testes.	41
Tabela 4 – Resultados obtidos para amostras detectadas e não detectadas.	42
Tabela 5 – Intervalos obtidos no teste II.....	42
Tabela 6 – Resultados para teste II das leituras detectadas e não detectadas.....	43
Tabela 7 – Resultados obtidos em primeira análise do teste III.	46
Tabela 8 – Resultados obtidos para cor vermelha em um ambiente escuro.....	48
Tabela 9 – Resultados cor verde em um ambiente escuro.	49
Tabela 10 – Resultados para cor azul em um ambiente escuro.....	50
Tabela 11 – Resultados para laranja em um ambiente escuro	51
Tabela 12 – Resultados de cor amarela em um ambiente escuro.	52
Tabela 13 – Intervalos obtidos teste III.....	53
Tabela 14 – Média percentual da variação de redução dos valores.	54

LISTA DE ABREVIATURAS E SIGLAS

CA	- Corrente Alternada
CC	- Corrente Contínua
CCD	- <i>Charge-Coupled Device</i>
CMOS	- <i>Complementary Metal-Oxide-Semiconductor</i>
CMYK	- <i>cyan, magenta, yellow and key</i>
EEPROM	- <i>Electrically-Erasable Programmable Read-Only Memory</i>
GND	- <i>Graduated Neutral Density Filter</i>
GUI	- <i>Graphical User Interface</i>
I2C	- <i>Inter-Integrated Circuit</i>
IDE	- <i>Integrated Development Environment</i>
LDR	- <i>Light Dependent Resistor</i>
LED	- <i>Light emitter diode</i>
MATLAB	- <i>Matrix Laboratory</i>
POO	- <i>Object-Oriented Programming</i>
PWM	- <i>Pulse Width Modulation</i>
RGB	- <i>Red, Green and Blue</i>
SPI	- <i>Serial Peripheral Interface</i>
USB	- <i>Universal Serial Bus</i>

LISTA DE SÍMBOLOS

® - marca registrada

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.1.1	Objetivo Geral.....	14
1.1.2	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	AUTOMAÇÃO	15
2.2	SENSORES E ATUADORES	16
2.2.1	Sensores de cor.....	17
2.2.1.1	Tipos de sensores de Cor.....	18
2.2.1.2	Sensor de cor TCS230	20
2.2.1.3	Sensor de cor TCS3200	21
2.2.1.4	Sensor de cor TCS34725	21
2.2.1.5	Sensor de gestos e cor RGB APDS-9930	22
2.3	SENSOR TCS230	24
2.4	SERVOMOTOR.....	27
2.5	ARDUINO	28
2.6	MATLAB	29
2.6.1	App Designer	30
3	METODOLOGIA.....	33
3.1	FUNCIONAMENTO DO PROTÓTIPO.....	33
3.2	ALGORITMO E CALIBRAÇÃO DO SENSOR	35
3.3	INTERFACE GRÁFICA	36
3.4	TESTES E AQUISIÇÃO DE DADOS DO SENSOR DE COR	40
4	RESULTADOS E DISCUSSÕES	41
4.1	TESTE I - MÉTODO EMPÍRICO.....	41
4.2	TESTE II - USO DO APLICATIVO EM UM AMBIENTE ILUMINADO.....	42
4.3	TESTE III - USO DO APLICATIVO EM UM AMBIENTE ESCURO	46
5	CONCLUSÃO.....	55
5.1	TRABALHOS FUTUROS.....	55
	REFERÊNCIAS	56
	APÊNDICE A – MANUAL DE USO DO APLICATIVO	59

APÊNDICE B – EXEMPLO DE TABELA CRIADA PARA ANÁLISE DE DADOS EM TESTE I: PEÇAS VERMELHAS.....	60
---	-----------

1 INTRODUÇÃO

A automação é um elemento indispensável quando o objetivo é fornecer um produto ou serviço o mais confiável e eficiente possível (MEDEIROS, 2017).

Atualmente vê-se uma gama de aplicações de automação nos mais diversos campos, como residências, agricultura, como também em indústrias onde os processos de fabricação são automatizados e controlados. Essa evolução tecnológica permite a substituição de processos manuais por sistemas automatizados, capazes de executar tarefas de forma mais eficiente, precisa e rápida.

Com o progresso da automação o uso de tecnologias de sensores e microcontroladores se tornou amplamente utilizado. O microcontrolador é usado como uma ferramenta que pode ajudar no trabalho humano. E os sensores se tornaram de grande importância devido sua capacidade de sentir o ambiente ao seu redor (SIHOMBING, 2019).

Dentre as aplicações da automação, está a classificação de objetos, principalmente no contexto das cores. Atualmente existem diversas pesquisas sobre a ação exercida de sensores detectores de cor, para diversos setores, como na classificação de grãos, frutas, plantas, em laboratórios químicos, tendo em vista a limitação humana nesses processos, o que não acontece quando há o emprego de máquinas, pois elas têm a capacidade de estabelecer um padrão uniforme visualmente e diminuir o tempo gasto na execução dessas tarefas.

No entanto, a precisão e confiabilidade das medições realizadas por esses dispositivos dependem diretamente de uma calibração adequada. A calibração consiste em comparar as leituras fornecidas pelo sensor com um padrão de referência confiável, a fim de determinar e corrigir eventuais desvios ou erros de medição.

Neste trabalho, será abordada a temática do desenvolvimento de um sistema de calibração para um módulo sensor de cor TCS230, uma vez que este tipo de sensor possui características específicas que podem ser influenciadas e afetar as leituras do sensor, resultando em medições imprecisas, como diferentes condições de iluminação, reflexões, sombras e variações nos objetos.

Dessa forma a utilização do protótipo em outros ambientes pode acarretar variação no envio de dados do sensor. Assim como a troca do componente, pois cada sensor TCS230 pode apresentar variações de sensibilidade em diferentes regiões do

espectro de cores. A calibração permite ajustar essas variações, garantindo uma resposta mais confiável e consistente do sensor em todas as faixas de cores.

Assim, o presente trabalho busca explorar as etapas necessárias para a calibração de sensor, desde a seleção dos padrões de referência até a análise e interpretação dos resultados obtidos.

Por fim, espera-se que este trabalho contribua para o entendimento da importância da calibração de sensores e conhecimentos necessários para o desenvolvimento de um sistema de calibração eficiente e confiável.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver uma interface gráfica no ambiente *app designer* do software MATLAB® para auxiliar na calibração de um módulo sensor de cor TCS230 integrado a um sistema de seleção de peças por cores controlado por Arduino UNO.

1.1.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Descrever os elementos típicos de sistemas de automação;
- Desenvolver um algoritmo capaz de interpretar e processar os dados do sensor de cor, bem como comandar os servomotores;
- Elaborar a interface gráfica que faça a aquisição de dados para calibração utilizando a ferramenta *App Designer* do MATLAB;
- Elaborar a lógica para a comunicação entre o MATLAB e o sistema de aquisição de dados implementado no Arduino;
- Determinar uma escala de cores possíveis de reconhecimento pelo sistema;
- Avaliar os corpos de prova detectados e não detectados pelo intervalo criado a partir dos dados obtidos;
- Aplicar testes para validação do sistema por meio de comparação com e sem o sistema desenvolvido, e de acordo as variações de luz ambiente.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 AUTOMAÇÃO

A automação tem sido uma das principais forças impulsionadoras do progresso tecnológico nas últimas décadas. Através da aplicação de máquinas, inteligência artificial e robótica, a automação tem revolucionado setores industriais, comerciais e até mesmo domésticos.

A automação é um conceito e um conjunto de técnicas por meio das quais se constroem sistemas ativos capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre o qual atuam. Com base nas informações, o sistema calcula a ação corretiva mais apropriada para a execução da ação e esta é uma característica de sistemas em malha fechada, conhecidos como sistemas de realimentação, (...). Para tanto, são utilizados controladores que, por meio da execução algorítmica de um programa ou circuito eletrônico, comparam o valor atual com o valor desejado, efetuando o cálculo para ajuste e correção (SILVEIRA; SANTOS, 1998 p.23).

A automação engloba desde processos simples até processos de sistemas complexos que envolvem máquinas, software e algoritmos avançados. Tem como objetivo aumentar a eficiência, a produtividade e a precisão, além de reduzir o tempo e o esforço humano necessários para execução de determinadas atividades.

A automação pode ser classificada segundo suas diversas áreas de aplicação, por exemplo: automação comercial, bancária, industrial, agrícola, de comunicações e transportes. A automação industrial pode ser dividida em automação de planejamento, de projeto, de produção (FUENTES, 2005).

A automação industrial refere-se à aplicação de tecnologias e sistemas para automatizar e controlar processos industriais. Os processos industriais tratam das atividades envolvidas na produção de bens ou na prestação de serviços em um ambiente industrial. Esses processos podem variar de acordo com o setor ou a indústria específica.

De acordo com Fuentes (2005), a classificação dos processos e tipos de produção e seus respectivos sistemas de produção podem ser representados como:

- **Processo de fluxo contínuo:** Um sistema de produção contínua para grandes quantidades de produtos (geralmente pós ou líquidos). exemplo: refinaria e indústria química.
- **Produção em massa (série):** Um sistema que produz produtos com pouca variação. exemplo: automóveis e eletrodomésticos.
- **Produção em lotes:** Um sistema que produz uma quantidade média de produtos que podem ser repetidos regularmente. exemplo: livros e roupas.
- **Produção de pedidos especiais (fabricação de ferramentas):** Um sistema de produção comum para pequenos lotes de todos os tipos de produtos. exemplo: Protótipos, ferramentas, dispositivos.

A automação industrial promove o desenvolvimento de um sistema otimizado capaz de reduzir custos, aumentar a quantidade e a qualidade dos produtos produzidos, manter padrões consistentes e duradouros na produção, reduzir o tempo gasto para a produção e auxiliar na liberação dos trabalhadores de atividades monótonas, repetitivas e até mesmo perigosas.

Ao estudar automação, como sistemas industriais, comerciais, automotivos, entre outros, é necessário determinar as variáveis do sistema. Nesse contexto é preciso obter o valor de uma variável física no ambiente monitorado e esse é o trabalho do sensor.

2.2 SENSORES E ATUADORES

Parte fundamental de qualquer sistema de automação e responsável pela resposta de um sistema de controle, um sensor pode ser definido como um dispositivo sensível que responde a alguma forma de energia em um sistema (como cinética ou luminosa) e tem a capacidade de mensurar certas quantidades do sistema, como temperatura, posição, velocidade etc. No entanto, os sensores nem sempre possuem as propriedades elétricas necessárias para uso em sistemas de controle, o que significa que a manipulação de sinal deve ser usada para fazer leituras (THOMAZINI; ALBUQUERQUE, 2013).

Geralmente os sensores são divididos em dois tipos: sensores analógicos e sensores digitais. Essa classificação está de acordo com a resposta que o sensor apresenta diante à variação da condição do sistema em análise.

Os sensores analógicos são os dispositivos em que o sinal de saída pode assumir qualquer valor dentro da sua faixa de operação. Enquanto os sensores digitais assumem apenas dois valores ao longo do tempo para seu sinal de saída, que podem ser descritos como valor binário - zero (0) e um (1) (PATSKO, 2006).

Já um atuador, em linhas gerais, se define como um dispositivo que tem a capacidade de modificar o valor de uma variável controlada. Eles recebem sinais do controlador e atuam no sistema controlado. Podem ser encontrados na forma de motores, relés, cilindros, entre outros (THOMAZINI; ALBUQUERQUE, 2013).

Souza (2016) destaca que, assim como os sensores, os atuadores também podem processar sinais digitais e analógicos. No entanto, os atuadores recebem sinais (digitais ou analógicos) do controlador e os sensores enviam sinais (digitais ou analógicos) ao controlador.

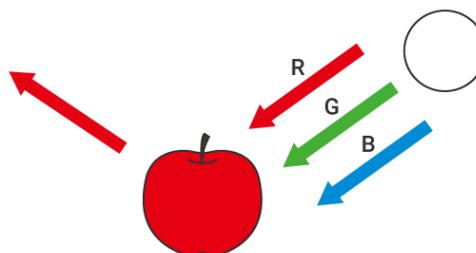
2.2.1 Sensores de cor

Os sensores de cor são sensores baseados no efeito fotoelétrico. Seu funcionamento consiste na emissão de uma luz branca, que ao atingir a superfície em análise retorna refletida. Como resultado dessa reflexão, é produzido um padrão RGB (*Red*, *Blue* e *Green*), que permite o sensor emitir sinais discretos ou de comunicação em termos de especificações de cores que caracterizam o objeto sob inspeção (DA SILVA, 2019).

O sistema RGB, é assim denominado em razão da sua formação das cores vermelho, verde e azul. É um sistema do modelo de cores aditivas, o que significa que a soma das três cores gera o branco e a ausência das três cores gera o preto (ausência de cor) (LOPES, 2013).

Dessa forma, os sensores de cor assumem que todos os materiais têm a capacidade de refletir a intensidade da luz de acordo com a sua cor, como mostra a Figura 1. Ao detectar mudanças na quantidade de luz recebida, é possível detectar a presença ou ausência de um objeto, bem como sua cor. Santos (2011) ressalta que, as principais características que mostra a sofisticação de um sensor de cor, são: sensibilidade, precisão, alcance e velocidade de resposta.

Figura 1 – Representação da reflexão em um objeto de cor vermelha.



Fonte: KEYENCE (2023).

Os sensores de cor podem ser classificados como analógicos ou digitais. São exemplos: Um fotorresistor LDR (*Light Dependent Resistor*) é um tipo analógico cujo valor de resistência muda de acordo com a intensidade da luz. Um fototransistor trata-se de um sensor de cor digital que opera detectando mudanças na quantidade de luz refletida ou bloqueada por um objeto (SANTOS, 2011).

Existe diversos tipos de sensores de cor e inúmeras aplicações em indústrias e mercados, operando na verificação de cores de embalagens e tampas, posições de etiquetas (DA SILVA, 2019).

2.2.1.1 Tipos de sensores de Cor

Um sensor de cor é um dispositivo eletrônico que é projetado para detectar e medir cores em diferentes objetos ou ambientes. Eles são amplamente utilizados em várias aplicações, desde sistemas de visão computacional até controle de iluminação e classificação de produtos. Existem diferentes tipos de sensores de cor, cada um com seus princípios de funcionamento e características específicas. Alguns exemplos comuns incluem:

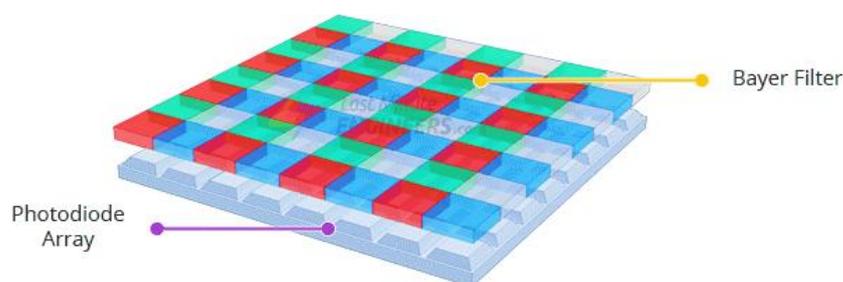
- a) Sensores RGB: Esses sensores possuem fotodetectores sensíveis à luz nas faixas do espectro vermelho, verde e azul. Eles capturam a intensidade de cada cor primária e combinam essas informações para determinar a cor geral. São geralmente utilizados em aplicações de monitoramento de cor, sistemas de visão computacional e iluminação controlada por cor.
- b) Sensores de cor espectrofotométricos: Esses sensores medem a distribuição espectral da luz refletida por um objeto. Eles analisam a

intensidade da luz em várias faixas de comprimento de onda para determinar a composição espectral da cor. São utilizados em aplicações de controle de qualidade, classificação de produtos, análise de cores em alimentos e indústria de tintas;

- c) Sensores de cor de filtro interferencial: Esses sensores utilizam filtros de interferência para medir a intensidade da luz em diferentes faixas de comprimento de onda. Podem ser calibrados para detectar cores específicas ou operar em espaços de cores específicos, como RGB, CMYK, entre outros. São usados em aplicações de classificação de objetos, controle de processos industriais e análise de cores;

Geralmente, a maioria dos sensores de cor utilizam a tecnologia de sensores CCD (*Charge-Coupled Device*) ou CMOS (*Complementary Metal-Oxide-Semiconductor*), que trata de dispositivos que utilizam matrizes formadas por fotodiodos. Fotodiodos são diodos que absorvem fótons de luz, liberando elétrons e gerando corrente elétrica. A Figura 2, apresenta o exemplo de uma matriz de fotodiodos com grade de filtro no padrão Bayer.

Figura 2 – Matriz fotodiodos e grade de filtro no padrão Bayer.



Fonte: LAST MINUTE ENGINEERS (2023).

Os sensores CCD, usam uma matriz de fotodetectores, cada fotodetector captura a luz incidente e converte em carga elétrica proporcional à intensidade da luz, assim captura informações de cor. A matriz de fotodetectores pode ser organizada em diferentes padrões, como RGB ou Bayer. São encontrados em câmeras digitais, scanners de documentos e sistemas de visão computacional.

Enquanto, os sensores CMOS, são dispositivos eletrônicos que convertem a luz em um sinal elétrico. São um tipo de dispositivo de imagem que também usa uma

matriz de fotodetectores, mas possuem uma estrutura de circuito integrado diferente dos sensores CCD, essa tecnologia permite a integração de circuitos analógicos e digitais no mesmo chip, o que resulta em uma construção mais compacta e eficiente. São usados em câmeras de dispositivos móveis, webcams e sistemas de visão em tempo real.

Ambos os sensores dependem da técnica de aproximação de pontos para formação da imagem.

Alguns sensores de cor populares disponíveis no mercado são os módulos: TCS230, TCS3200, TCS34725 e APDS-9930, além desses, os sensores de cor podem ser feitos usando o Arduino. Por apresentar um custo acessível podem ser usados didaticamente, e serão apresentados no presente trabalho.

2.2.1.2 Sensor de cor TCS230

O sensor de cor TCS230, Figura 3, é um sensor de cor RGB (*Red, Green, Blue*) desenvolvido pela AMS (Antiga TAOS Inc.). Ele é um sensor óptico que pode detectar e medir cores com base na intensidade da luz em diferentes faixas do espectro.

Figura 3 – Módulo sensor de cor RGB TCS230.



Fonte: ORANGE MAKER (2023).

Aqui estão algumas características principais do sensor de cor TCS230:

- Matriz de fotodetectores;
- Diferentes frequências de resposta;
- Saída de sinal digital;
- Filtros de cor;
- Controle de ganho ajustável;
- Baixo consumo de energia.

O sensor de cor TCS230 é comumente utilizado em aplicações como detecção de cores, controle de iluminação, classificação de objetos, robótica e automação industrial. Sua simplicidade de uso o tornam uma escolha popular em diversas aplicações.

2.2.1.3 Sensor de cor TCS3200

O sensor de cor TCS3200, Figura 4, é um dispositivo eletrônico que permite a detecção e medição da cor de um objeto. Ele é amplamente utilizado em projetos de eletrônica e robótica que envolvem reconhecimento de cor.

Figura 4 – Sensor TCS3200.



Fonte: Adaptado de ELETROGATE (2023).

Este módulo sensor de cor é baseada no TCS230, e apresenta características semelhantes (ELETROGATE, 2023).

Possui uma matriz de fotodiodos que detectam a luz incidente e a convertem em sinais elétricos proporcionais à intensidade da cor. A matriz de fotodiodos é composta por quatro conjuntos de fotodiodos, cada um correspondendo a uma cor primária (vermelho, verde, azul e branco). Ele utiliza filtros de cor para separar a luz incidente nas diferentes faixas de cores.

O TCS3200 possui um circuito integrado que processa os sinais dos fotodiodos e os converte em valores digitais. Esses valores digitais podem ser lidos por um microcontrolador ou outro dispositivo eletrônico para análise e tomada de decisão.

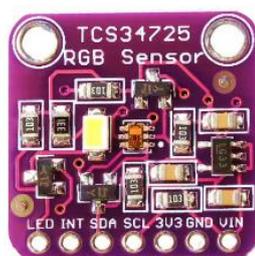
2.2.1.4 Sensor de cor TCS34725

O sensor de cor TCS34725 é um sensor digital de cor da AMS. Ele é uma versão aprimorada do TCS34715 e é projetado para medir e detectar com precisão a cor de objetos ou ambientes. Suas principais características são:

1. Detecção de cores através de fotodetectores sensíveis à luz nas faixas do espectro vermelho, verde, azul e infravermelho próximo;
2. Interface digital I2C (*Inter-Integrated Circuit* - Circuito Inter-integrado), que facilita a integração com microcontroladores e outros dispositivos eletrônicos;
3. Modos de ganho e tempo de integração ajustáveis, para otimizar a captura de cores em diferentes condições de iluminação;
4. Alta sensibilidade e faixa dinâmica, que permite a captura precisa de cores em diferentes condições de iluminação;
5. Filtros de luz ambiente e luz infravermelha, possui filtros incorporados para rejeitar a luz ambiente e filtrar a luz infravermelha;
6. Redução de ruído e compensação de temperatura.

O sensor de cor TCS34725, Figura 5, é amplamente utilizado em aplicações que requerem medição e detecção precisa de cores, como controle de iluminação, classificação de produtos, monitoramento de cor em ambientes industriais e sistemas de visão computacional.

Figura 5 – Módulo TCS34725.



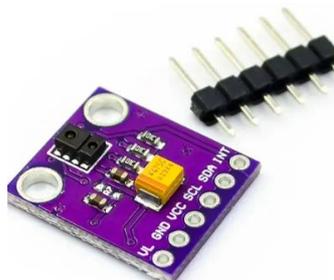
Fonte: Adaptado EASYTRONICS (2023).

2.2.1.5 Sensor de gestos e cor RGB APDS-9930

O módulo sensor de gestos e cor RGB APDS-9930, Figura 6, é um dispositivo eletrônico que combina a detecção de gestos e a medição de cores RGB em um único sensor. Ele pode reconhecer gestos como aproximar, afastar, deslizar e tocar, além de identificar a cor de objetos próximos. É utilizado em diversas aplicações, como dispositivos de interface de usuário baseados em gestos, iluminação inteligente,

sistemas de detecção de cor em robótica, entre outros, onde a detecção de gestos e a medição de cores são importantes.

Figura 6 – Módulo sensor de gestos e cor RGB APDS-9930.



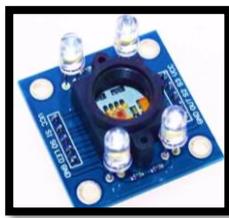
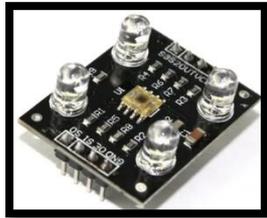
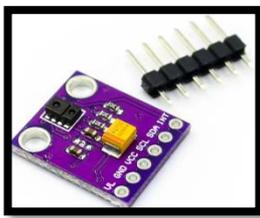
Fonte: NEWPEÇAS (2023).

Além da detecção de gestos, o APDS-9930 também engloba um sensor de cores RGB que pode medir a intensidade das cores vermelha, verde e azul presentes em uma superfície, permitindo a identificação e análise de cores em um ambiente.

É um sensor que se comunica com um microcontrolador ou outro dispositivo eletrônico por meio de interfaces de comunicação como I2C ou SPI (*Serial Peripheral Interface*), e contém configurações personalizáveis como, sensibilidade do sensor de proximidade e de detecção de gestos, bem como configurações de ganho e tempo de integração para a medição de cores RGB.

Na Tabela 1 comparou-se alguns parâmetros essenciais dos quatro módulos de sensores de cor.

Tabela 1 – Comparação de módulos de sensores de cor mais populares.

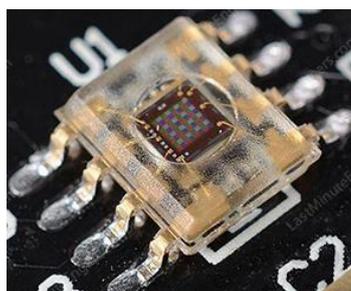
SENSORES DE COR				
	TCS32000	TCS230	TCS34725	APDS 9960
Fator de comparação				
Tensão Operacional	2.7 - 5.5 VCC	2.7 - 5.5 VCC	2.7 - 3.6 VCC	3.8 VCC
Faixa de detecção	10 - 12,5 mm	10 - 12,5 mm	3 - 10 mm	100 - 200 mm
Interface	I2C	I2C	I2C	I2C
Iluminação Externa	SIM	SIM	SIM	NÃO
Entrada de luz	LED RGB	LED RGB	LED RGB	LED RGB
Consumo	25mA < 50mA	25mA < 50mA	235µA	200µA

Fonte: Adaptado de CHÀVES; FIERRO (2021).

2.3 SENSOR TCS230

O TCS230 é um sensor de detecção de cor composto por 4 LEDs (*Light Emitting Diode* – Diodo Emissor de Luz), um componente CMOS com uma matriz 8x8 de fotodiodos, Figura 7, capacitores, resistores e 8 pinos.

Figura 7 – Matriz 8x8 de fotodiodos presente no TCS230



Fonte: LAST MINUTE ENGINEERS (2023).

De acordo com Da Silva (2019), o módulo TCS230, tem seu princípio de funcionamento baseado na conversão da cor da luz em frequência, filtrando os dados RGB da fonte de luz e transformando em uma onda quadrada (50% *duty cycle* – ciclo de trabalho) com frequência diretamente proporcional à intensidade da luz, isto é, a onda apresenta metade do período em nível alto e a outra metade em nível baixo.

Este módulo possui entradas e saídas digitais que podem ser conectadas diretamente a um microcontrolador. Existem 64 fotodiodos, 16 com filtros vermelhos, 16 com filtros azuis, 16 com filtros verdes e 16 sem filtros. Esses quatro fotodiodos são interligados para minimizar os efeitos do brilho desigual da luz incidente. Todos os fotodiodos da mesma cor são conectados em paralelo e o fotodiodo que o dispositivo usa durante a operação pode ser selecionado por pinos (TAOS, 2008).

A Figura 8 exemplifica visualmente como ocorre o funcionamento desses filtros.

Figura 8 – Matriz de fotodiodos e fotodiodo verde.



Fonte: Silva (2010).

Cada quadrado representa um fotodiodo. Do lado esquerdo da imagem há 4 fotodiodos de cores vermelho, verde e azul, sendo a cor verde em maior número e as outras se encontram distribuídas equilibradamente na matriz. Do lado direito há uma representação para o caso de como esta matriz atua refletindo em determinada superfície, o fotodiodo verde com filtro verde permite que apenas fótons de cor verde passem. As outras cores são bloqueadas (ERLO, 2019).

O sensor TCS230 conta com quatro pinos de entradas: S0, S1, S2 e S3. Os pinos S0 e S1 servem para selecionar uma escala de frequência (2%, 20% ou 100%) e os pinos S2 e S3 são utilizados para selecionar o conjunto de fotodiodos desejado. A ativação do sensor atribui-se na programação um estado 'High' ou 'Low' para o pino. As Figura 9 e 10 mostram os resultados das combinações dos pinos S0 e S1 e dos pinos S2 e S3, respectivamente.

Figura 9 – Combinação dos filtros e escala de frequência nos pinos S0 e S1.

S0	S1	Escala da frequência de saída
Low	Low	Desligado
Low	High	2%
High	Low	20%
High	High	100%

Fonte: CORSICO; GLIR (2014).

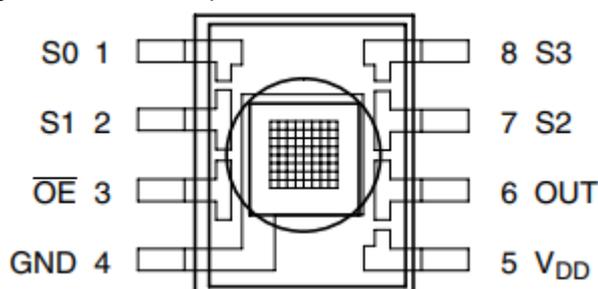
Figura 10 – Combinação dos filtros e escala de frequência nos pinos S2 e S3.

S2	S3	Tipo do fotodiodo
Low	Low	Red (vermelho)
Low	High	Blue (azul)
High	Low	Clear (sem filtro)
High	High	Green (verde)

Fonte: CORSICO; GLIR (2014).

O pino \overline{OE} (*Output Enable*) habilita a saída em estado de alta impedância para o uso de vários sensores em apenas uma linha de entrada do microcontrolador, pode ser ligado ao terminal de referência de acordo o datasheet do sensor. O terminal *OUT* é o pino para o sinal de saída, o pino VDD é para alimentação (tensão) e o GND (terra) é o nó de referência e pode ser conectado ao terminal \overline{OE} . A Figura 11 apresenta uma vista superior do esquema de pinagem do sensor.

Figura 11 – Vista superior de um sensor de cor TCS230.



Fonte: TAOS LUMENOLOGY. Datasheet: TCS230 (2008).

Considerando a teoria de cores RGB, existe três valores distribuídos, o primeiro para R (vermelho), o segundo para G (verde) e o terceiro para B (azul) que são combinados com diferentes intensidades da cor (sem filtro), para obter tons diferentes. As medições são feitas da seguinte forma: Quatro LEDs brancos iluminam o objeto a ser identificado. A partir daí, a luz refletida é enviada para o sensor TCS230. As informações obtidas são enviadas para um microcontrolador, onde um software é utilizado para medir as respectivas frequências dos canais R, G e B. Isso significa que existem três valores diferentes, um para cada variável (QUEIROZ, 2017).

2.4 SERVOMOTOR

Servomotor é um atuador eletromecânico que faz parte do grupo de motores elétricos, podem ser tanto motores CA (corrente alternada) quanto motores CC (corrente contínua), geralmente são do tipo CC.

Para McRoberts (2011), um servo é composto de um motor elétrico CC, um conjunto de engrenagens entre o motor e um eixo de saída, um mecanismo sensor de posição, e um circuito de controle, agregados em uma pequena caixa. E seu funcionamento consiste na transmissão da posição do servo para o circuito de controle através do sensor de posição, que por sua vez faz uso do motor para que o servo seja ajustado na posição que irá ocupar.

Um Servomotor pode ser encontrado em muitos tamanhos, e em velocidades e precisões diferentes. Na maioria dos casos giram 180 graus, mas é possível encontrar servos de rotação contínua.

Para efeito de comparação, a Tabela 2 apresenta as principais características dos servomotores comerciais mais adequados às necessidades do protótipo deste trabalho.

Tabela 2 – Comparação de servomotores mais populares.

SERVOMOTORES				
	MG995	Servomotor SG90	FS5103R	FT5335M
Fator de comparação				
Tensão Nominal	3 - 7,2 VCC	4,8 VCC	4,8 - 6 VCC	6 - 7,4 VCC
Torque	15kg/cm	1,8kg/cm	3kg/cm	35kg/cm
Malha	Metálico	Plástico	Plástico	Metal
Ângulo de Rotação	180°	180°	360°	180°
Consumo	250mA	100mA	500mA	700mA

Fonte: Adaptado de CHÀVES; FIERRO (2021).

O Servomotor escolhido foi o *servomotor de referência SG90*, Chaves; Fierro (2021), trata-se de um motor de baixo custo que atende adequadamente às necessidades do protótipo, pois não realiza funções que exijam força excessiva e é responsável apenas para conduzir as peças.

2.5 ARDUINO

Para obter as informações dos sensores e controlar os atuadores, pode-se utilizar para prototipagem o Arduino, que é uma plataforma de *hardware* e *software* livres, isto é, pode ser utilizada por qualquer pessoa com a liberdade de estudar, adaptar/modificar e distribuir.

A parte de *hardware*, consiste em uma placa com microcontrolador, que pode ser considerado como um pequeno computador, já que é composto por microprocessador, memória RAM, memória flash, temporizadores, contadores, dentre outras.

O software, é a IDE (*Integrated Development Environment*) do Arduino, que permite que o usuário escreva um programa de computador na linguagem compreendida pelo Arduino, baseada na linguagem C. Esse programa executado no computador é denominado de sketch, que fará o envio para a placa Arduino (MCROBERTS, 2011).

A placa Arduino pode ser utilizado juntamente com qualquer dispositivo que emita dados ou possa ser controlado (LEDs, sensores, receptores GPS, módulos Ethernet etc.), o que permite adquirir informações e ser sensível ao ambiente ao seu redor fazendo o controle de luzes, motores e outros atuadores.

A Figura 12 mostra o Arduino Uno, é uma das placas Arduino mais populares e amplamente utilizadas. Ele é baseado no microcontrolador ATmega328P e oferece um bom equilíbrio entre recursos, capacidades e acessibilidade.

Figura 12 – Arduino Uno.



Fonte: Silva (2019).

Esta placa é alimentada pelo microcontrolador ATmega328P, que possui 32 KB de memória flash para armazenamento de código, 2 KB de SRAM para armazenamento de dados e 1 KB de EEPROM para armazenamento não volátil. Além disso, o Uno possui entradas e saídas analógicas e digitais, com 14 pinos, sendo que 6 deles podem ser usados como saídas PWM (*Pulse Width Modulation*). Ele também tem 6 pinos de entrada analógica (DA SILVA, 2019).

O ATmega328P na placa Uno roda a 16 MHz, fornecendo poder de processamento suficiente para a maioria dos projetos Arduino. Através de conexão USB (tipo B padrão) que permite que ele seja conectado a um computador para programação e comunicação. O Arduino Uno pode ser alimentado através da conexão USB ou de uma fonte de alimentação externa, como bateria ou fonte de alimentação CC.

2.6 MATLAB

O MATLAB (*MATrix LABoratory* - Laboratório de Matrizes) é um software de computação bastante flexível, com capacidade para resolver qualquer problema técnico, especializado e otimizado para cálculos científicos e de engenharia.

Esta ferramenta oferece uma ampla gama de recursos, sendo um deles a possibilidade de desenvolver sistemas supervisórios que se comunicam com o Arduino. O MATLAB possui suporte à comunicação serial e programação, recursos gráficos para criar interfaces de usuário personalizadas, permitindo interações com o Arduino para controle, monitoramento e análise de dados em tempo real.

É um software bastante versátil em cálculos matemáticos, modelagens e simulações, análises numéricas e processamentos, visualização e gráficos, desenvolvimento de algoritmos, entre outros (GILAT, 2012).

2.6.1 App Designer

O *App Designer* (Designer de Aplicativos), é uma ferramenta pertencente ao software MATLAB®, voltado para o desenvolvimento de aplicativos com componentes visuais de uma interface gráfica de usuário (GUI) e ao mesmo tempo programar o comportamento do aplicativo.

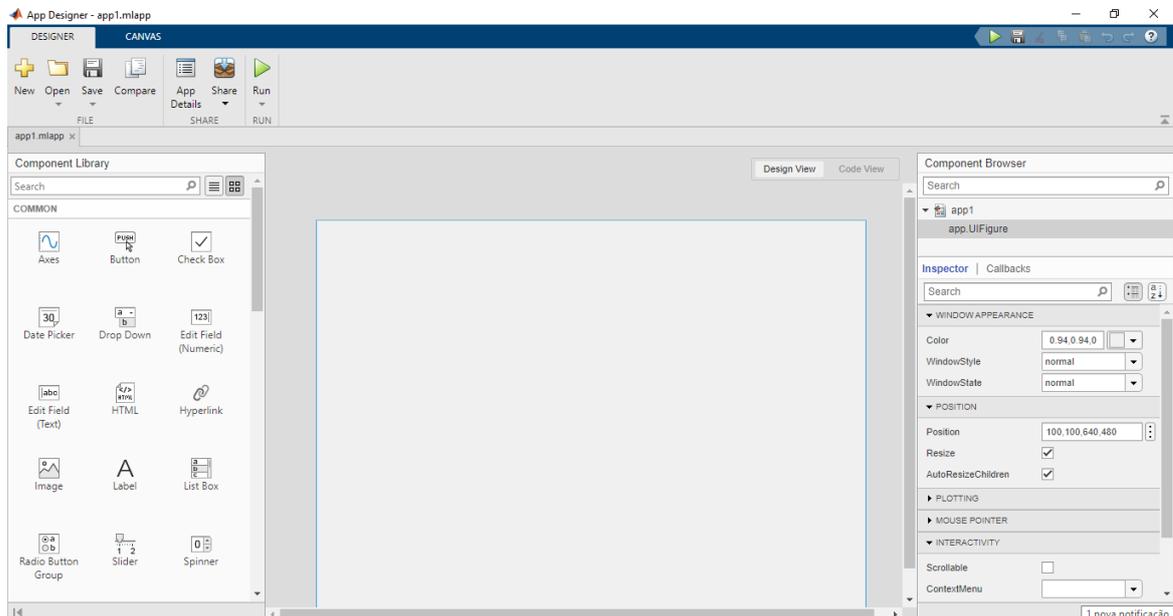
Apresenta uma interface “amigável” ao usuário, permitindo que qualquer pessoa crie aplicativos profissionais sem precisar ser um desenvolvedor de software profissional, pois apresenta recursos como:

- **Projeto de interface do usuário:** O usuário pode arrastar e soltar os componentes visuais na tela de design e usar dicas de alinhamento para um layout preciso e gera automaticamente código orientado a objetos que define o layout e o design do aplicativo.
- **Comportamento do aplicativo:** fornece uma versão totalmente integrada do MATLAB *Editor* que auxilia no comportamento do aplicativo criado, uma vez que, o *App Designer* pode usar analisadores de código (*Code Analyzer*) para verificar automaticamente problemas de codificação. À medida que escreve o código, é possível visualizar mensagens de aviso e erro sobre o código em desenvolvimento, garantindo que o código seja modificado com base nas mensagens.
- **Biblioteca de componentes:** a ferramenta possui componentes padrão de aplicativos, como botões, caixas de seleção, árvores e listas suspensas, e fornece controles como medidores, lâmpadas, botões e interruptores que permitem replicar a aparência e as ações dos painéis de instrumentação. E possibilita a organização da interface através de componentes de contêiner, como guias, painéis e layouts de grade. Além disso, apresenta opções de refluxo automático para fazer o aplicativo criado detectar e responder a mudanças no tamanho da tela

- **Interação do componente:** é possível adicionar retornos de chamada de componentes e ações personalizadas de mouse e teclado que são executadas quando o usuário interage com o aplicativo. Gráficos e tabelas 2D e 3D permitem que os usuários explorem os dados de forma interativa.
- **Compartilhar aplicativo:** o projeto criado pode ser compartilhado com outros usuários do MATLAB via MATLAB Online e MATLAB Drive para que eles possam executar o aplicativo, colaborar em *designs* e ter permissão para editar os arquivos. Além disso, é permitido guardar o aplicativo em um único arquivo, criando um programa executável da interface com extensão (. *mlappinstall*). Os arquivos do instalador permitem que o usuário e outras pessoas instalem o aplicativo e acessem-no na galeria de aplicativos, sem necessidade de comprar a licença do software MATLAB® (MATHWORKS, 2023).

A janela de início do *App Designer* pode ser vista na Figura 13.

Figura 13 – Janela em branco do *app designer*.



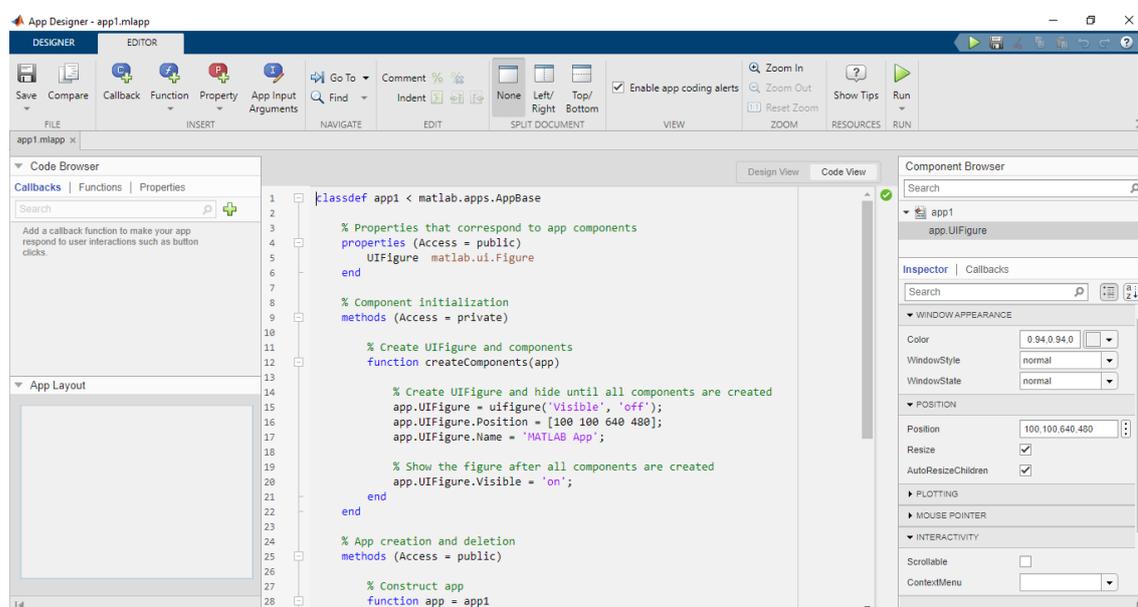
Fonte: Autoria própria (2023).

No lado esquerdo da imagem está a biblioteca de componentes visuais fornecida pela caixa de ferramentas (*Toolbox*), e o meio mostra uma prévia da interface onde serão criadas as telas do aplicativo. O lado direito mostra o componente do menu propriedades dos componentes.

O *Axes* permite o *plot* (plotar) de gráficos no aplicativo, o *Button* (botão) é programado para agir com o click do usuário, e o *EditField numeric* (caixa de edição numérica) permite a inserção de valores numéricos pelo usuário. Esses são alguns elementos presentes no *App Designer*.

A programação do aplicativo no *App Designer* é feita por meio da guia *Code View* (exibição de código), conforme Figura 14, que usa *callbacks* (função de retorno de chamada) para executar o código quando ocorre um evento, ou seja, é um comando executado em resposta a alguma ação predefinida do usuário, como clicar em um objeto gráfico ou fechar uma janela de imagem (MATHWORKS, 2023).

Figura 14 – Janela do *code view* - *app designer*.



Fonte: Autoria própria (2023).

Na visualização do código, na aba esquerda, é possível adicionar *call-backs* do tipo públicos e/ou privados, funções (métodos) e propriedades ao seu contexto de POO (*Object-Oriented Programming* – Programação Orientada a Objetos). Estas modificações são automaticamente implementadas no código. Ao salvar o projeto no *App Designer* é gerado um arquivo (.mlapp). Esse arquivo pode ser executado diretamente do *App Designer* ou chamado do código para inicializar seu aplicativo gráfico (LOPES; LOPEZ; MARTHA, 2022).

3 METODOLOGIA

O desenvolvimento do sistema de calibração para o sensor de cor foi realizado em duas etapas principais. Na primeira definiu-se a estrutura física, o código do Arduino e os primeiros testes de leitura do sensor. E a segunda etapa consistiu no desenvolvimento da interface gráfica de usuário (GUI), em código MATLAB com a opção *App Designer* e testes para validação do sistema.

3.1 FUNCIONAMENTO DO PROTÓTIPO

A estrutura física onde o sensor de cor está inserido, é um projeto elaborado por Da Silva (2019), cujo sistema trata do protótipo de uma máquina separadora de peças por cor. O protótipo, mostrado na Figura 15, consiste em uma estrutura de madeira contendo um sensor de cor do tipo TCS230 e dois servos motores que estão ligados ao microcontrolador Arduino, de acordo a Figura 16, que é o responsável pelo funcionamento do sistema, o que o torna viável para dimensão do projeto por tratar-se de um sistema em pequena escala (DA SILVA, 2019).

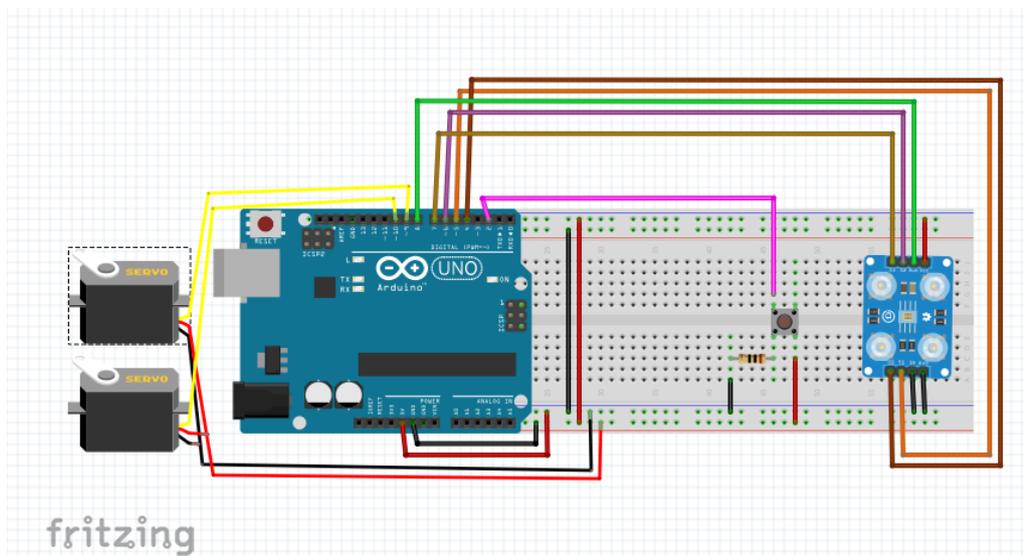
Figura 15 – Protótipo da máquina separadora de peças por cor.



Fonte: Autoria própria (2023).

A Figura 16 mostra o esquema de ligação de circuito entre os principais componentes do protótipo.

Figura 16 – Ligação do circuito principal.



Fonte: Autoria própria (2023).

Para este trabalho a operação do sistema será de forma manual através de um botão *push button*, para facilitar o processo de calibração. A atuação do sistema consiste em quatro etapas principais.

Etapa 1: Este é o início do processo, em que se encontra o coletor de peças, um tubo que recebe as peças coloridas, com um clique no botão o primeiro servomotor 1 recebe a peça;

Etapa 2: por meio de um segundo clique ao *push button* (botão de pressão) a peça é levada até o sensor de detecção de cores, TCS320;

Etapa 3: nesta etapa o sensor de identificação de cores faz a leitura da escala de frequência que é enviado pelo objeto em análise. O sistema permanece nesse estado até o botão ser pressionado novamente;

Etapa 4: o botão é pressionado uma terceira vez, para dispensar a peça na rampa que leva ao descarte no recipiente, nesse estágio está o servomotor 2, que nesta aplicação não recebe comandos de movimentação, seu uso é apenas para suporte da rampa, uma vez que o objetivo neste projeto não é a separação das peças em si.

3.2 ALGORITMO E CALIBRAÇÃO DO SENSOR

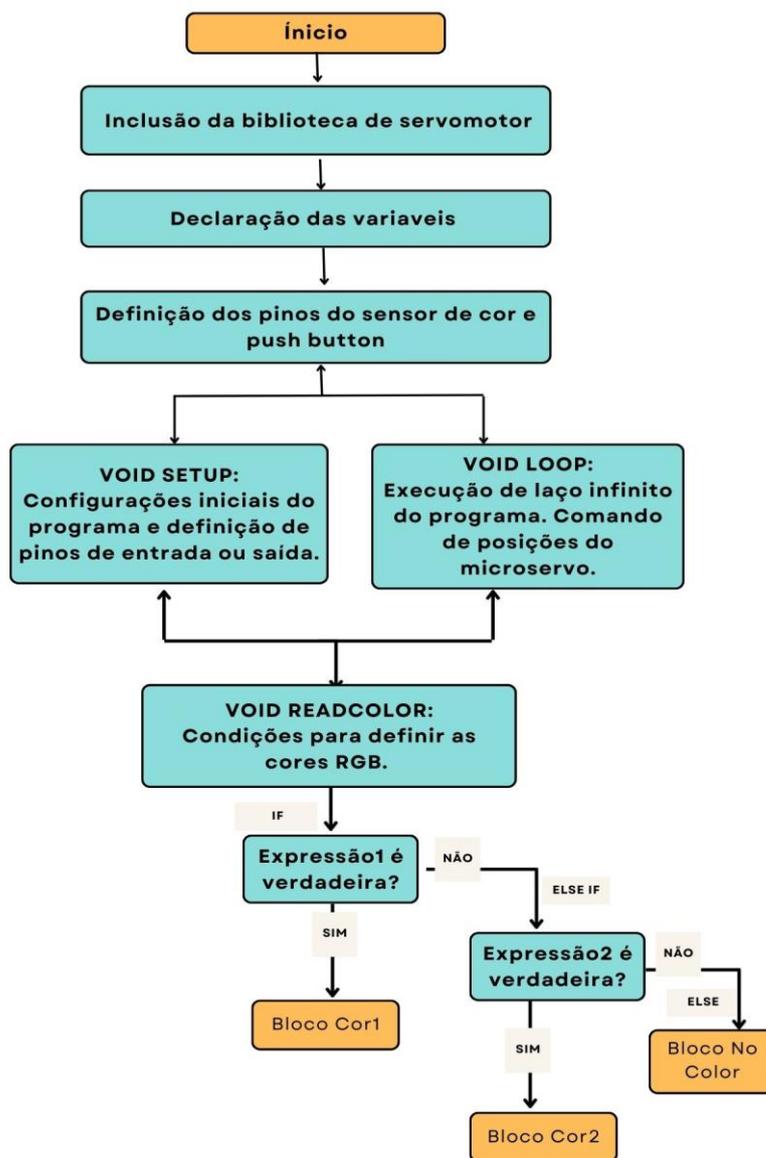
O código para o projeto foi desenvolvido no Ambiente de Desenvolvimento Integrado (IDE). A programação é a chave principal para a comunicação entre o Arduino e o sensor, que permite a obtenção dos dados emitidos pelo sensor, com intuito de fazer uma leitura desses valores no sistema de cores RGB. Sendo possível observar os valores RGB medidos pelo sensor através do monitor serial e assim determinar as faixas correspondente a cada cor. Ademais, o software apresenta a facilidade de modificar o código e testá-lo, e possibilita a comunicação com outros softwares.

Silva (2021) ressalta que, como o sensor lê valores analógicos que representam o comprimento de onda emitido pela cor, é preciso calibrar na programação para que o processo de captura de dados em comprimentos de onda e conversão para códigos de cores RGB seja automático.

Então, a calibração do sensor pode ser realizada através da programação, inserindo nas linhas de código os intervalos de frequência que corresponde a cor a ser lida.

A Figura 17, mostra um diagrama de blocos da programação desenvolvida na IDE do Arduino para o circuito do sistema do protótipo.

Figura 17 – Diagrama da programação em Arduino.



Fonte: Autoria própria (2023).

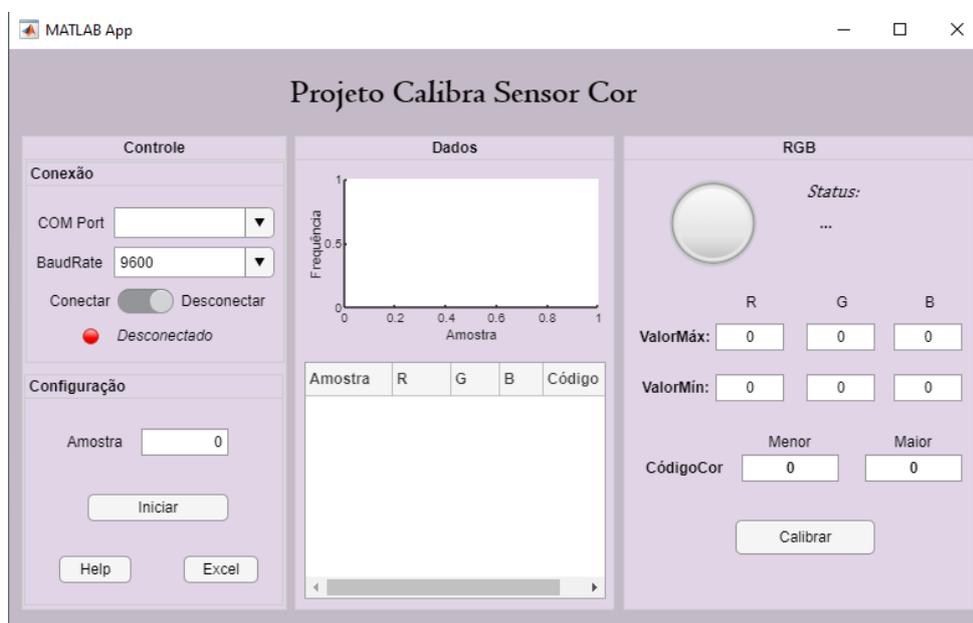
O programa desenvolvido tem capacidade de fazer a leitura de qualquer cor, desde que sejam adicionadas essas linhas de código e o usuário encontre o intervalo da escala da cor que deseja. Para este projeto, foram escolhidas cinco cores (vermelho, verde, azul, laranja e amarelo).

3.3 INTERFACE GRÁFICA

O desenvolvimento da plataforma do sistema de calibração do sensor consiste na elaboração da interface gráfica a ser apresentada para o usuário. Esta foi desenvolvida com o uso da ferramenta *App Designer* do software MATLAB, que auxilia na aquisição de dados, plotagem de gráficos e criação de tabelas.

A Figura 18 a seguir ilustra a interface gráfica desenvolvida para este projeto, para o seu desenvolvimento foi utilizado ferramentas como *Lamps*, *Edit text*, *Edit numeric*, botão *switch*, gráfico, tabela, botões e *Label* (texto) presentes no ambiente *App Designer*.

Figura 18 – Janela da GUI desenvolvida para auxiliar na calibração do sensor cor.



Fonte: Autoria Própria (2023).

A interface conta com três painéis, o primeiro de controle, o segundo de recebimento dos dados e o terceiro de calibragem do sensor. O painel de controle se divide em duas seções principais, na seção 1, como mostra a Figura 19, pode-se realizar a comunicação/conexão com o Arduino através de um botão *switch* (liga/desliga). A seção é acompanhada por um botão lâmpada (*Lamp*) que se mostrará verde para uma conexão bem-sucedida ou vermelho caso não haja conexão, como também a presença de um texto que indica “conectado” e “desconectado”.

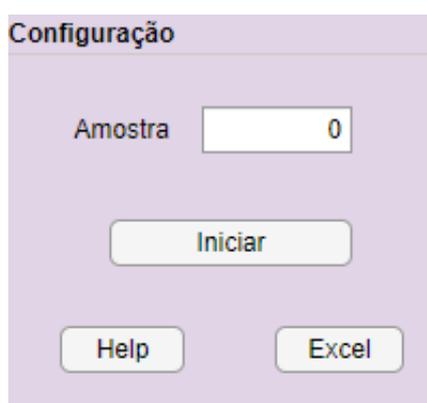
Figura 19 – Seção 1: Conexão.



Fonte: Autoria Própria (2023).

E na seção 2, de acordo a Figura 20, denominada configuração, há um campo de texto editável onde o usuário deve inserir a quantidade de amostras (leituras de frequência RGB) que deseja coletar e o botão de 'Iniciar', para que a aquisição de dados seja realizada. Além disso, a interface conta com o botão 'Help' que direciona a um arquivo, Apêndice A, com instruções para auxiliar os usuários no funcionamento do aplicativo e o botão 'Excel' para enviar os dados recebidos para uma planilha do Microsoft Excel para possíveis análises.

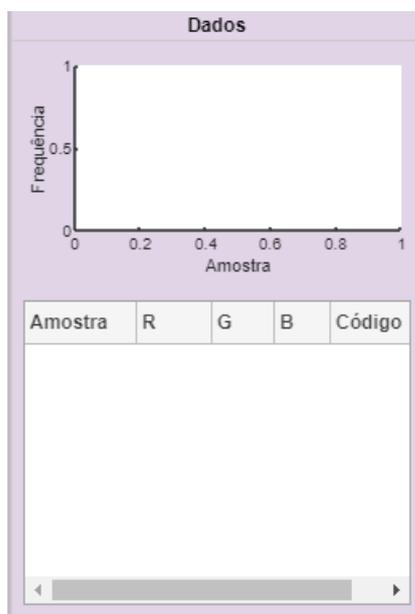
Figura 20 – Seção 2: Configuração.



Fonte: Autoria Própria (2023).

No segundo painel o usuário visualizará os dados recebidos, através de um gráfico e de uma tabela que reproduz a leitura dos dados enviados pelo Arduino e a quantidade de amostra solicitada pelo usuário, demonstrado pela Figura 21.

Figura 21 – Painel 2: Dados.



Fonte: Autoria Própria (2023).

Finalmente, a interface mostra um terceiro painel. Este painel tem a capacidade de exibir as frequências R, G e B máximas e mínimas alcançadas ao final do monitoramento. Possui um espaço para o código da cor lida que possibilita o usuário verificar se a aquisição dos dados fora bem-sucedidas. Além disso, contém o botão 'Calibrar', uma *Lamp* que indica visualmente a cor detectada e um texto contendo o nome de cada cor correspondente, conforme pode ser observado na Figura 22.

Figura 22 – Painel 3: Ambiente de apresentação dos dados de calibração.

The 'RGB' panel includes a circular indicator labeled 'Status: ...'. Below it are input fields for 'ValorMáx' and 'ValorMin' for R, G, and B, all set to 0. There are also 'Menor' and 'Maior' fields for 'CódigoCor', both set to 0. A 'Calibrar' button is at the bottom.

Fonte: Autoria Própria (2023).

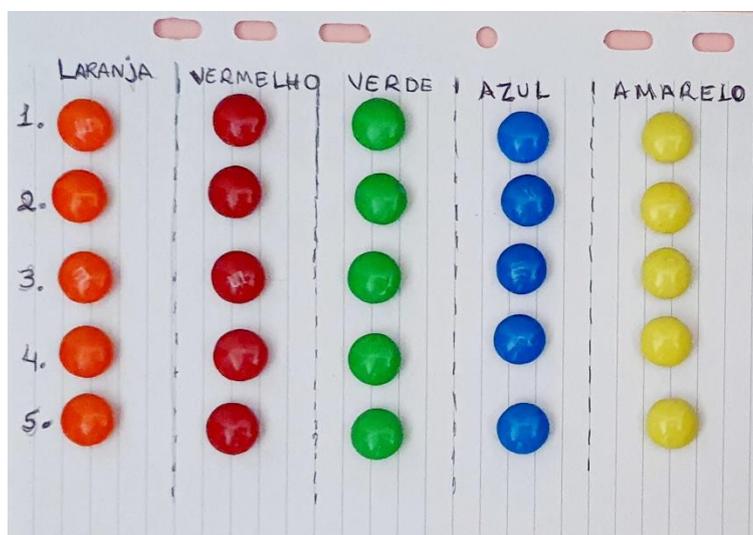
3.4 TESTES E AQUISIÇÃO DE DADOS DO SENSOR DE COR

O Corpo de prova utilizado para os testes foram doces de chocolate coloridos, que exibe cores uniformes e atende os requisitos para o protótipo como em tamanho e forma. Para que os confeitos pudessem ter um tempo de durabilidade maior e não tivesse suas características alteradas durante o uso foi aplicado esmalte incolor para revestir cada peça.

Para a verificação da confiabilidade do sistema foram realizados três tipos de testes, para calibrar o sensor. No primeiro a calibração feita pelo método empírico, em que os dados foram coletados manualmente por meio do monitor serial. O segundo e o terceiro realizaram-se utilizando o Sistema 'Projeto Calibra Sensor Cor', um com ambiente iluminado e outro com ambiente sem iluminação, respectivamente. Todos os testes foram realizados com o mesmo sensor e tiveram como objetivo observar o índice de confeitos detectados ou não detectados.

Para isso, foram separados cinco confeitos, conforme Figura 23, de cada uma das cores. Os corpos de prova foram inseridos, individualmente, sobre o sensor. Os resultados obtidos foram implementados no código principal. Para cada caso, as frequências detectadas e não detectadas foram registradas e organizadas em tabelas. Depois de recolher todas as informações recebidas, os resultados foram comparados e analisados para tirar conclusões sobre a eficiência do aplicativo.

Figura 23 – Corpos de prova organizados em sequência de testes.



Fonte: Autoria Própria (2023).

4 RESULTADOS E DISCUSSÕES

Nesta seção serão apresentados e discutidos os resultados obtidos. O foco do trabalho foi dado para a implementação de um sistema que tornasse a calibração do sensor de cor mais ágil, por meio do desenvolvimento de um aplicativo com o *App Designer*.

4.1 TESTE I - MÉTODO EMPÍRICO

Inicialmente com auxílio do monitor serial do Arduino e um tempo de amostragem de 10s, foram retiradas 10 (dez) leituras das frequências RGB emitidas pelo sensor para cada uma das peças, que foram enviadas para o Microsoft Excel onde usando as funções "Média", "Mínimo" e "Máximo", foi calculado a média dos valores obtidos em R, G e B de cada um dos corpos de prova, da mesma forma que foram obtidos os valores mínimos e máximos. Os valores foram dispostos em tabelas, exemplo no Apêndice B para cor vermelha.

A partir desta análise foi criado o intervalo da frequência do sinal recebido pelo sensor para implementar nas linhas de código que calibra o sensor. Os resultados podem ser visualizados na Tabela 3.

Tabela 3 – Intervalos obtidos com primeira análise de testes.

Cores	Filtro		
	R	G	B
Vermelho	170-135	285-240	215-170
Verde	200-150	200-150	200-150
Azul	235-189	215-160	137-110
Laranja	125-90	240-210	205-180
Amarelo	115-90	140-110	155-135

Fonte: Autoria Própria (2023).

A Tabela 4, mostra uma análise dos resultados obtidos através da leitura realizada para todas os corpos de prova a partir da programação do código a partir dos intervalos da Tabela 3. A nomenclatura D para Detectados e ND para não detectados, e amostra quer dizer as leituras de frequência RGB coletadas do sensor.

Tabela 4 – Resultados obtidos para amostras detectadas e não detectadas.

Cor		Vermelho		Verde		Azul		Laranja		Amarelo	
Quantidade de Peças	Total de Amostra	D	ND	D	ND	D	ND	D	ND	D	ND
1	10	7	3	10	0	9	1	9	1	7	3
2	10	8	2	8	2	7	3	4	6	10	0
3	10	8	2	10	0	1	9	10	0	9	1
4	10	10	0	7	3	9	1	1	9	9	1
5	10	8	2	8	2	10	0	8	2	9	1

Fonte: Autoria Própria (2023).

4.2 TESTE II - USO DO APLICATIVO EM UM AMBIENTE ILUMINADO

Os experimentos foram repetidos com o sistema ‘Projeto Calibra Sensor Cor’, utilizando os mesmos critérios citados no Teste I. Algumas peças foram detectadas com facilidade, enquanto outras não. Dessa forma, o código foi atualizado através da criação de novos intervalos, conforme Tabela 5, que não foi considerado na análise do teste I.

Tabela 5 – Intervalos obtidos no teste II.

Cores	Filtro		
	R	G	B
Vermelho	180-130	300-240	280-170
Verde	280-150	280-150	280-150
Azul	255-189	220-190	150-120
Laranja	128-90	245-210	215-165
Amarelo	125-90	150-102	160-130

Fonte: Autoria Própria (2023).

Os intervalos foram criados com uma escala maior, com objetivo de identificar a leitura de mais de uma peça. As leituras obtidas foram satisfatórias, com a atualização, todas as peças de cor vermelha foram detectadas, para as cores verde e azul foram detectadas nove peças, e os “não detectados” foram diminuídos para as cores amarela e laranja, como mostra a Tabela 6, resultado da análise feita para todas os corpos de prova.

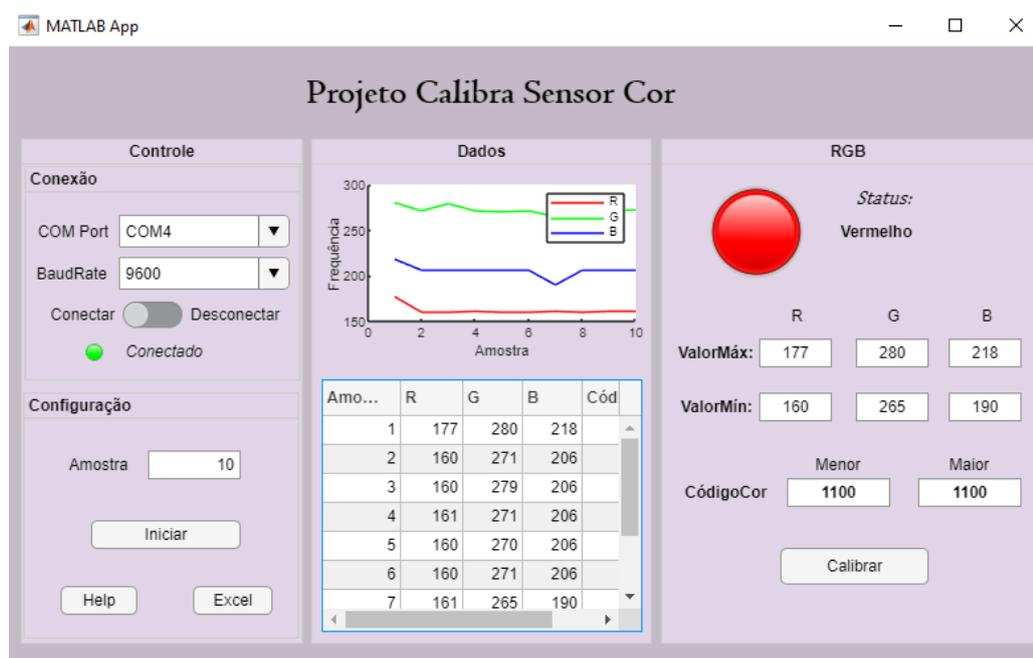
Tabela 6 – Resultados para teste II das leituras detectadas e não detectadas.

Cor		Vermelho		Verde		Azul		Laranja		Amarelo	
Quantidade de Peças	Total de Amostra	D	ND	D	ND	D	ND	D	ND	D	ND
1	10	10	0	10	0	10	0	10	0	10	0
2	10	10	0	10	0	9	1	10	0	9	1
3	10	10	0	9	1	10	0	9	1	10	0
4	10	10	0	10	0	10	0	9	1	9	1
5	10	10	0	10	0	10	0	10	0	10	0

Fonte: Autoria Própria (2023).

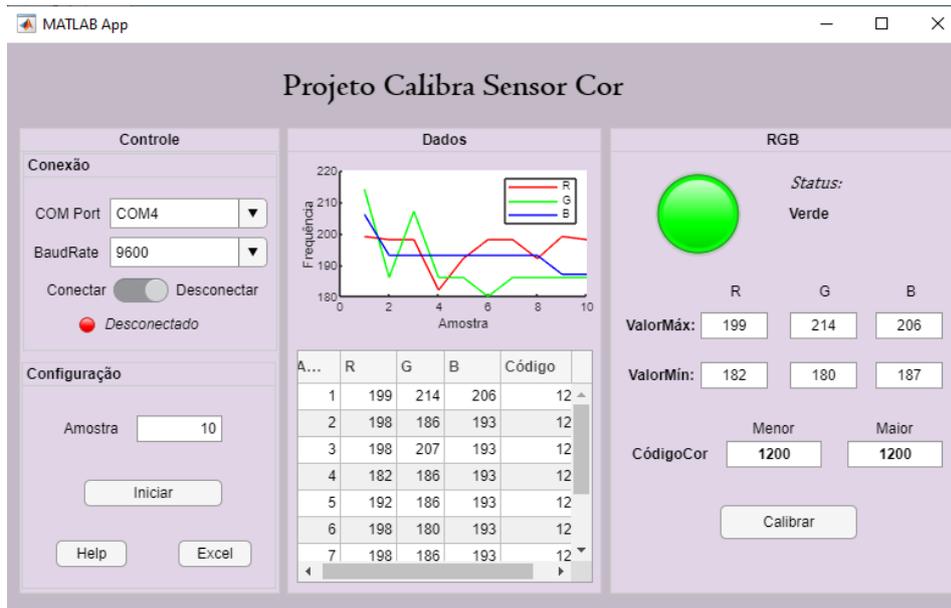
As Figuras 24, 25, 26, 27, 28 apresentam, respectivamente, o comportamento da interface para cada cor analisada neste trabalho.

Figura 24 – Interface Gráfica calibração da peça vermelha.



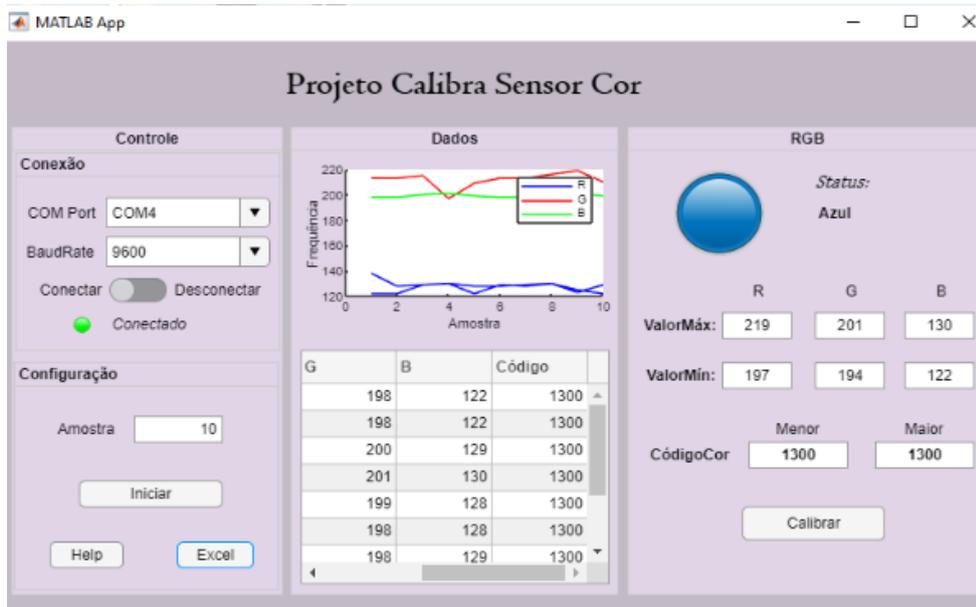
Fonte: Autoria própria (2023).

Figura 25 – Interface Gráfica calibração da peça verde.



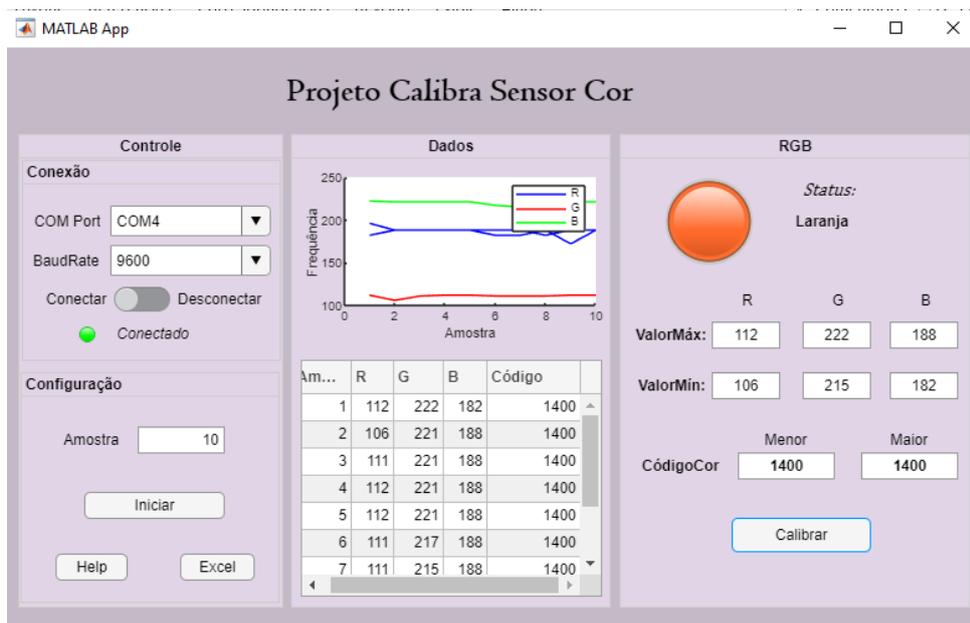
Fonte: Autoria própria (2023).

Figura 26 – Interface Gráfica calibração da peça azul.



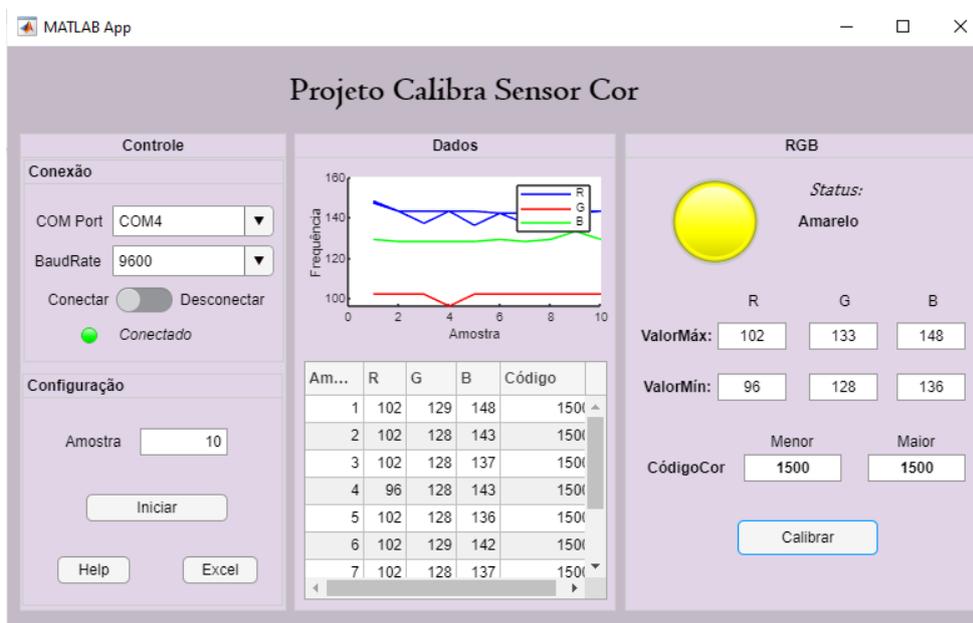
Fonte: Autoria própria (2023).

Figura 27 – Interface Gráfica calibração da peça laranja.



Fonte: Autoria própria (2023).

Figura 28 – Interface Gráfica calibração da peça amarela.



Fonte: Autoria própria (2023).

4.3 TESTE III - USO DO APLICATIVO EM UM AMBIENTE ESCURO

Neste caso, usando as informações do Teste II, porém considerando um ambiente sem luz, para as peças de número 1 de cada cor foi utilizado as mesmas linhas de código do teste II, com o objetivo de observar o comportamento dos resultados.

Cor Vermelha

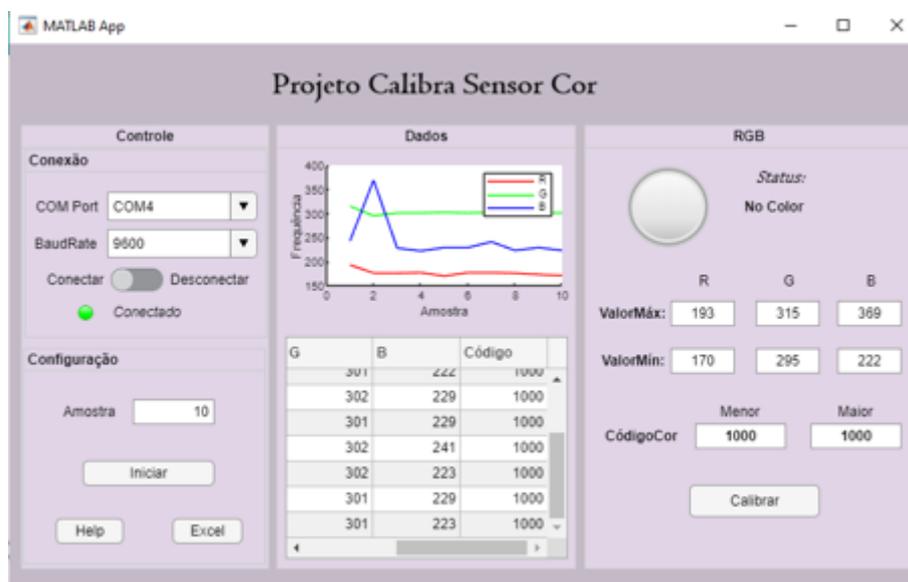
A Tabela 7 apresenta os valores obtidos, para peça vermelha em um ambiente escuro e como o aplicativo desenvolvido interpretou os dados, Figura 29.

Tabela 7 – Resultados obtidos em primeira análise do teste III.

Amostra	R	G	B	CódigoCor
1	193	315	243	1000
2	176	295	369	1000
3	176	301	228	1000
4	177	301	222	1000
5	170	302	229	1000
6	177	301	229	1000
7	177	302	241	1000
8	176	302	223	1000
9	173	301	229	1000
10	171	301	223	1000

Fonte: Autoria própria (2023).

Figura 29 – Tela da interface do aplicativo mostra a não detecção de uma peça.



Fonte: Autoria própria (2023).

Percebe-se que o sensor passou a refletir uma frequência maior em G, não permitindo que a peça fosse identificada com o intervalo criado no Teste II, logo é necessário analisar as novas leituras e alterar a linha de código para os intervalos da cor vermelha, e assim calibrar o sensor novamente para detectar a peça no ambiente escuro.

Pela Tabela 7 e a Figura 29 é possível analisar como o intervalo de frequência na programação deve ser alterado. Com o objetivo de calibrar mais de uma peça, criou-se a linha de código a seguir, Figura 30.

Figura 30 – Linha de código com intervalo de calibração da cor vermelha.

```
if(R<200 || R>120 || G<400 || G>240 || B<380 || B>170){
    Serial.println("1100");
    //Serial.println("vermelho");
}
```

Fonte: Autoria própria (2023).

Com o intervalo criado a partir da peça 1 foi possível identificar as outras quatro peças. A Tabela 8 montada através do botão “Excel” do sistema, mostra o

comportamento dos dados emitidos pelo sistema para cada peça de cor vermelha com os novos intervalos criados.

Tabela 8 – Resultados obtidos para cor vermelha em um ambiente escuro.

Nº de amostra	Peça 1				Peça 2			
	R	G	B	Código	R	G	B	Código
1	176	303	225	1100	178	299	221	1100
2	170	301	222	1100	174	299	226	1100
3	176	296	229	1100	174	298	221	1100
4	177	296	229	1100	175	292	226	1100
5	177	304	230	1100	175	299	228	1100
6	170	279	223	1100	177	294	227	1100
7	176	302	230	1100	176	299	226	1100
8	176	297	229	1100	170	283	220	1100
9	175	296	229	1100	171	300	222	1100
10	176	295	229	1100	170	296	218	1100
Nº de amostra	Peça 3				Peça 4			
	R	G	B	Código	R	G	B	Código
1	161	294	223	1100	163	285	210	1100
2	162	294	223	1100	157	284	217	1100
3	162	382	222	1100	163	340	216	1100
4	167	288	223	1100	162	285	216	1100
5	168	288	223	1100	163	285	216	1100
6	168	293	222	1100	163	284	216	1100
7	168	293	198	1100	163	285	211	1100
8	167	272	222	1100	157	366	217	1100
9	168	294	223	1100	163	285	216	1100
10	167	294	223	1100	163	285	216	1100
Nº de amostra	Peça 5							
	R	G	B	CódigoCor				
1	195	301	236	1100				
2	168	288	217	1100				
3	168	347	216	1100				
4	169	279	214	1100				
5	168	279	215	1100				
6	168	279	215	1100				
7	168	279	216	1100				
8	168	295	209	1100				
9	162	286	209	1100				
10	168	295	215	1100				

Fonte: Autoria própria (2023).

Cor verde

A mesma análise foi feita para a peça verde, a Tabela 9 mostra o comportamento do sistema para cada peça de cor verde. Todas as peças foram detectadas com a programação criada no Teste II e não foi necessário alteração no código.

Tabela 9 – Resultados cor verde em um ambiente escuro.

Nº de amostra	Peça 1				Peça 2			
	R	G	B	Código	R	G	B	Código
1	193	188	191	1200	203	228	210	1200
2	193	188	191	1200	201	200	201	1200
3	193	182	191	1200	200	272	200	1200
4	193	182	192	1200	206	201	194	1200
5	193	182	191	1200	200	200	194	1200
6	193	182	191	1200	206	200	200	1200
7	193	188	191	1200	205	200	200	1200
8	193	188	192	1200	206	202	200	1200
9	193	181	176	1200	206	199	194	1200
10	193	158	192	1200	199	200	194	1200
Nº de amostra	Peça 3				Peça 4			
	R	G	B	Código	R	G	B	Código
1	197	186	187	1200	195	185	195	1200
2	197	186	193	1200	196	191	196	1200
3	197	170	194	1200	195	192	189	1200
4	196	186	188	1200	191	191	166	1200
5	197	186	193	1200	195	191	195	1200
6	197	180	193	1200	195	191	189	1200
7	197	186	187	1200	189	191	195	1200
8	197	186	194	1200	189	192	195	1200
9	181	186	170	1200	196	191	180	1200
10	197	186	194	1200	195	192	195	1200
Nº de amostra	Peça 5							
	R	G	B	CódigoCor				
1	182	183	187	1200				
2	188	177	187	1200				
3	188	183	187	1200				
4	188	183	187	1200				
5	188	236	187	1200				
6	188	182	181	1200				
7	188	183	187	1200				
8	188	183	187	1200				
9	188	177	187	1200				
10	188	167	187	1200				

Fonte: Autoria própria (2023).

Cor azul

A Tabela 10 apresenta os resultados para as leituras das peças de cor azul. Visto que as medidas obtidas entre os filtros são de maior magnitude, para obter uma leitura completa de todos os confeitos é necessário alterar a linha de código para a cor azul, no filtro G, como mostra as peças 2 e 3.

Tabela 10 – Resultados para cor azul em um ambiente escuro.

Nº de amostra	Peça 1				Peça 2			
	R	G	B	Código	R	G	B	Código
1	232	200	130	1300	231	210	134	1300
2	231	199	130	1300	230	210	134	1300
3	231	206	123	1300	231	186	134	1000
4	231	206	124	1300	225	210	134	1300
5	231	206	129	1300	231	210	134	1300
6	232	200	129	1300	231	210	134	1300
7	231	184	130	1000	225	210	134	1300
8	231	200	130	1300	225	211	128	1300
9	231	206	130	1300	225	210	134	1300
10	231	200	130	1300	224	211	128	1300
Nº de amostra	Peça 3				Peça 4			
	R	G	B	Código	R	G	B	Código
1	237	217	143	1300	233	208	137	1300
2	237	223	144	1000	234	208	137	1300
3	237	222	137	1000	234	209	137	1300
4	231	223	143	1000	210	215	137	1300
5	237	223	143	1000	234	215	137	1300
6	237	222	144	1000	233	214	137	1300
7	237	216	144	1300	234	215	137	1300
8	237	217	143	1300	234	214	137	1300
9	237	223	143	1000	234	214	131	1300
10	236	217	144	1300	228	215	131	1300
Nº de amostra	Peça 5							
	R	G	B	CódigoCor				
1	221	201	127	1300				
2	221	201	127	1300				
3	221	201	127	1300				
4	227	201	127	1300				
5	227	201	127	1300				
6	227	201	127	1300				
7	227	201	127	1300				
8	227	201	126	1300				
9	226	201	127	1300				
10	227	201	127	1300				

Fonte: Autoria própria (2023).

Cor laranja

A análise executada para as amostras laranjas foi gerado a Tabela 11.

Tabela 11 – Resultados para laranja em um ambiente escuro

Nº de amostra	Peça 1				Peça 2			
	R	G	B	Código	R	G	B	Código
1	166	260	212	1100	120	247	199	1000
2	120	244	203	1400	120	247	200	1000
3	120	311	202	1000	120	247	205	1000
4	120	244	203	1400	120	240	205	1400
5	120	244	202	1400	120	247	205	1000
6	120	238	203	1400	120	241	205	1400
7	120	238	202	1400	120	246	205	1000
8	120	244	203	1400	120	247	200	1000
9	157	244	186	1100	120	247	206	1000
10	120	244	202	1400	120	241	205	1400
Nº de amostra	Peça 3				Peça 4			
	R	G	B	Código	R	G	B	Código
1	120	244	202	1400	144	267	215	1100
2	120	244	202	1400	121	241	205	1400
3	120	238	202	1400	121	318	205	1000
4	120	238	204	1400	105	247	198	1000
5	120	244	202	1400	115	247	205	1000
6	120	244	202	1400	121	247	205	1000
7	120	244	196	1400	121	241	205	1400
8	120	243	202	1400	121	247	199	1000
9	120	244	202	1400	121	247	202	1000
10	120	244	201	1400	121	247	205	1000
Nº de amostra	Peça 5							
	R	G	B	CódigoCor				
1	121	234	202	1400				
2	120	238	202	1400				
3	120	238	204	1400				
4	121	244	298	1400				
5	122	244	196	1400				
6	121	241	202	1400				
7	121	239	201	1400				
8	115	244	204	1400				
9	120	241	205	1400				
10	120	240	205	1400				

Fonte: Autoria própria (2023).

Em um ambiente sem luminosidade externa, algumas peças apresentaram características de peças de cor vermelha. A calibração dos corpos de prova se mostra bastante dificultosa, pois para não detectar peças vermelhas, em ambientes como o analisado, a programação deve ser bem específica, o que pode acarretar a não detecção de muitas peças.

Maia (2019), em seu trabalho sobre erros com esta cor relata que, em todos os testes realizados, o percentual de defeitos foi associado a partículas de cor laranja. Fazendo, o leitor de cores RGB entender que o objetivo final dessa cor era passar pelo canal de seletividade com uma tonalidade avermelhada.

Cor amarela

O estudo para peças amarelas, também apresentou necessidade de alteração no programa, como pode ser visto na Tabela 12, no filtro R (peça 1 e 5) e em B (peças 4 e 5). Observa-se que algumas peças não foram detectadas e outras emitem frequências que são detectadas como uma peça verde.

Tabela 12 – Resultados de cor amarela em um ambiente escuro.

Nº de amostra	Peça 1				Peça 2			
	R	G	B	Código	R	G	B	Código
1	113	135	155	1500	108	133	151	1500
2	114	141	149	1500	107	134	151	1500
3	113	135	155	1500	107	134	144	1500
4	113	125	148	1500	107	134	150	1500
5	89	141	155	1000	107	134	144	1500
6	113	141	155	1500	108	134	150	1500
7	107	141	155	1500	108	127	150	1500
8	113	141	155	1500	107	134	150	1500
9	114	141	155	1500	107	132	150	1500
10	113	141	155	1500	107	120	144	1500
Nº de amostra	Peça 3				Peça 4			
	R	G	B	Código	R	G	B	Código
1	109	134	148	1500	152	177	153	1200
2	109	134	142	1500	107	133	127	1000
3	109	134	148	1500	101	133	149	1500
4	109	134	147	1500	107	133	149	1500
5	109	134	148	1500	107	134	149	1500
6	109	134	147	1500	107	127	149	1500
7	108	134	147	1500	107	134	149	1500

8	109	134	147	1500	107	133	149	1500
9	109	134	148	1500	107	133	149	1500
10	108	134	148	1500	107	117	149	1500
Nº de amostra	Peça 5							
	R	G	B	CódigoCor				
1	134	159	154	1000				
2	106	132	148	1500				
3	106	132	148	1500				
4	106	132	125	1000				
5	106	131	148	1500				
6	106	131	148	1500				
7	106	131	148	1500				
8	106	132	148	1500				
9	106	132	148	1500				
10	106	132	148	1500				

Fonte: Autoria própria (2023).

Através da análise de cada tabela os intervalos foram alterados, conforme a Tabela 13, para cada cor, exceto a cor verde que teve as peças detectadas com as informações do Teste II.

Tabela 13 – Intervalos obtidos teste III.

Cores	Filtro		
	R	G	B
Vermelho	200-130	400-240	380-170
Verde	280-150	280-150	280-150
Azul	355-189	220-190	150-120
Laranja	128-90	245-210	215-165
Amarelo	129-90	245-200	215-165

Fonte: Autoria Própria (2023).

Diante das análises realizadas, percebe-se que em ambiente com menos luminosidade os valores se apresentam mais altos, enquanto em ambientes com mais luminosidade os valores se apresentam mais baixos, a Tabela 14 apresenta uma média percentual da variação de redução destes valores.

Tabela 14 – Média percentual da variação de redução dos valores.

	Média Percentual				
	Vermelho	Verde	Azul	Laranja	Amarelo
R	10%	6%	7%	4%	3%
G	9%	5%	3%	7%	3%
B	7%	4%	1%	4%	2%

Fonte: Autoria Própria (2023).

A calibração de forma empírica se mostra satisfatória, uma vez que o sistema não falha em todas as cores, porém, em uma análise de muitas amostras feita humanamente pode acarretar uma taxa mais alta de erros na medição. Ademais, exige maior tempo e dedicação na análise quando comparada a inspeção feita por meio do sistema desenvolvido (Projeto Calibra Sensor Cor), pois este apresenta a capacidade de trabalhar com um número exato de amostras.

A aplicação desenvolvida se mostrou bastante eficiente para o objetivo buscado, uma vez que proporciona a aquisição de dados, evitando que as amostras desejadas sejam selecionadas de forma manual e permite a obtenção dos valores máximos e mínimos necessários para as escalas de frequência. Além disso, os testes realizados através do aplicativo, mostra que dentre as cinco peças examinadas foi possível detectar uma faixa/ intervalo que realizasse a leitura para até 3 (três) peças de uma mesma cor, sendo possível definir facilmente o intervalo para as cores em análise.

Todavia, não se trata de uma ferramenta com conceitos de aprendizagem de máquina (utiliza valores de treino), o que pode acarretar erros no reconhecimento das cores e a nível de comparação não é possível fazer uma análise eficaz. Além disso, é possível perceber alguns pontos negativos do projeto desenvolvido como, a opção *códigoCor* no aplicativo ainda é limitado, em questão da codificação das cores, isso porque o programa tanto sendo programado para o código máximo ou o mínimo está sujeito a erros.

Destaca-se que o software foi desenvolvido com intenção de alcançar o usuário final, e um dos resultados obtidos foi o desenvolvimento do manual com instruções de uso (APÊNCIDE A) do aplicativo que permite ao usuário entendimento para manusear o software.

5 CONCLUSÃO

Os resultados obtidos mostram que o objetivo proposto foi alcançado com êxito. A interface consegue mostrar as frequências RGB dos sinais enviados pelo Arduino. O sistema “Projeto Calibra sensor Cor” mostrou-se de fácil uso e estável em relação à calibração do componente, sendo necessária apenas uma calibração para obter uma detecção de cor satisfatória para mais de uma peça em qualquer momento ou condição ambiental de acordo com a variação da luminosidade.

Convém ressaltar que o aplicativo desenvolvido fornece comodidade, pois a pessoa não precisa copiar dados do monitor serial, pois é possível fazer a aquisição dos dados e visualizar direto na tela do aplicativo de forma estática por meio de tabela e gráfico, sendo um programa que não visa apenas verificar valores obtidos com a variação da luminosidade, mas uma metodologia que auxilia na definição dos intervalos, em análise de dados e permite que o tempo gasto na calibração seja menor.

Por fim, para outras aplicações que optar pelo uso de um sensor de cor que seja do tipo TCS230 ou apresente funcionamento semelhante, o uso da plataforma se mostra útil para calibração dos mesmos. Como o aplicativo desenvolvido com o pensamento no usuário final, este apresenta recursos de orientação, como um manual de instruções de uso e a possibilidade de visualizar os dados bem como exportar ao Microsoft Excel para análises que vai além das funcionalidades apresentadas pela plataforma.

5.1 TRABALHOS FUTUROS

Para futuros trabalhos, tem-se a ampliação da pesquisa para o aprimoramento do protótipo na busca de estimular discentes no trabalho com a área de automação e controle, como a elaboração de roteiros para experimentos de automação e microcontroladores, verificação de funcionamento de servomotores avaliando a capacidade e rapidez em mover as peças. Outra sugestão é criar soluções de programação tanto para o Arduino quanto no MATLAB para tornar a calibração mais automática.

REFERÊNCIAS

ALLDATASHEET. **Sensor RGB TCS34725**. Disponível em: TCS34725 Folha de dados (PDF) - ams AG (alldatasheet.com) .Acesso em: 06 jun. 2023.

AVAGO TECHNOLOGIES. Datasheet: APDS-9960 **DIGITAL PROXIMITY, AMBIENT LIGHT, RGB AND GESTURE SENSOR**. 2007. Disponível em: https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf. Acesso em: 26 mai. 2023.

CHÁVEZ, Johann Andrés Cruz; FIERRO, Juan David Castro. **PROTOTIPO FUNCIONAL DE UN SISTEMA DE CLASIFICACIÓN PARA LAS CEREZAS DE CAFÉ CASTILLO, EN FUNCIÓN DE SU ETAPA DE MADURACIÓN**. Disponível em: < <https://repository.ucatolica.edu.co/server/api/core/bitstreams/d6769816-d922-4833-8f76-4858a7965836/content>>. Acesso em: 26 mai. 2023.

CORSICO, Cesar Augusto; GLIR, Guilherme Hatschbach. **SISTEMA AUTOMATIZADO PARA SEPARAÇÃO DE OBJETOS BASEADO EM CORES**. 67f.Curitiba,2014. Disponível em: <http://www.eletrica.ufpr.br/p/arquivostccs/312.pdf>. Acesso em: 08 maio 2023.

DA SILVA, ADONIAS WANDERSON DUARTE. **AUTOMAÇÃO DE UM SISTEMA DE SELEÇÃO DE PEÇAS POR CORES COM ARDUINO E SUPERVISÓRIO SCADA**. 2022. 59 p. Trabalho de conclusão de curso (Bacharel em Engenharia de Controle e Automação) - Universidade Federal do Oeste do Pará – UFOPA, SANTARÉM, PA, 2019.

Datasheet TCS3200, TCS3210 – **Programmable Color Light-to-Frequency Converter** – TAOS, 2011. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/560507/AMSCO/TCS3200.html>. Acesso em: 18 jun. 2023.

EASYTRONICS. **Sensor RGB TCS34725 | Reconhecimento de Cores**. Disponível em: <https://www.easytronics.com.br/sensor-rgb-tcs34725>. Acesso em: 05 jun. 2023.

ELETROGATE. **Módulo Sensor De Cor TCS230/TCS3200 - GY-31**. Disponível em: <https://www.eletrogate.com/modulo-sensor-de-cor-gy-31-tcs230-tcs3200>. Acesso em: 06 jun. 2023.

ERLO, Gibran. **ROBÓTICA EDUCACIONAL: RECICLAGEM DE RESÍDUOS**. 2019. 64f.Caxias do Sul, 2019. Disponível em: <https://repositorio.ucs.br/xmlui/handle/11338/6300?locale-attribute=en>. Acesso em: 18 jun. 2023.

FUENTES, Rodrigo Cardozo. Apostila de Automação Industrial: **Sensores**. 2005.Disponível em: http://w3.ufsm.br/fuentes/index_arquivos/CA03.pdf. Acesso em: 26 mai. 2023.

GILAT, Amos. **MATLAB com aplicações em engenharia**. Bookman Editora, 2012.

Keyence. **Guía de sensores para fábricas clasificados por principios, Fundamentos del sensor**, ¿Qué es un sensor de color? Disponível em: <<https://www.keyence.com.mx/ss/products/sensor/sensorbasics/color/info>>. Acesso em: 16 jun. 2023.

LAST MINUTE ENGINEERS. **INTERFACE DO SENSOR DE CORES TCS230/TCS3200 COM ARDUINO**. Disponível em: <https://lastminuteengineers.com/tcs230-tcs3200-color-sensor-arduino-tutorial/>. Acesso em: 18 jun. 2023

LOPES, João Manuel Brisson. **Cor e Luz**. 2013. 47 páginas. **Cor e Luz**: Texto elaborado para a disciplina de Computação Gráfica no curso de Licenciatura em Engenharia Informática e de Computadores. Instituto Superior Técnico. Universidade de Lisboa. Disponível em: <http://disciplinas.ist.utl.pt/leic-cg/textos/livro/Cor.pdf>. Acesso em: 03 jun. 2023.

LOPES, Pedro Cortez; LOPEZ, Rafael Rangel; MARTHA, Luiz Fernando. **O Ambiente App Designer do MATLAB**. CIV2801 – Fundamentos da Computação Gráfica Aplicada, 2022. Disponível em: https://www.tecgraf.puc-rio.br/ftp_pub/lfm/CIV2801-AmbienteAppDesigner.pdf. Acesso em: 22 mai. 2023.

MAIA, Andrada Cassio. **ELABORAÇÃO DE PROTÓTIPO SELETOR DE GRÃOS DE CAFÉ, BASEADO EM SENSOR DE CORES RGB PARA SE DETERMINAR O INÍCIO DA COLHEITA**. 40f. Caratinga, 2019. Disponível em: <https://dspace.doctum.edu.br/handle/123456789/3394>. Acesso em: 18 maio 2023.

MATHWORKS. **MATLAB App Designer**. 2022. Disponível em: <https://www.mathworks.com/products/matlab/app-designer.html>. Acesso em: 21 mai. 2023.

McROBERTS, Michael. **Arduino Básico**. São Paulo: Editora Novatec LTDA. 2011.

MEDEIROS, P. H. S. de. **Controle de uma maquete de ponte rolante com interface gráfica no MATLAB utilizando o microcontrolador ATMEGA 2560 embarcado em uma plataforma Arduino**. Vitória-Es, 2017. Disponível em: https://ele.ufes.br/sites/engenhariaeletrica.ufes.br/files/field/anexo/pedro_h_s_medeiros.pdf. Acesso em: 1 jun. 2023.

NEWPEÇAS. **Sensor De Gestos E Cor Rgb Apds-9930**. Disponível em: <https://www.newpecas.com.br/produto/sensor-de-gestos-e-cor-rgb-apds-9930>. Acesso em: 05 jun. 2023.

ORANGEMAKER. **Sensor de cores RGB TCS3200**. Disponível em: <https://orangemaker.com.br/produto/sensor-de-cores-rgb-tcs3200/>. Acesso em: 05 jun. 2023.

PATSKO, Luis Fernando. **Tutorial, Aplicações, Funcionamento e Utilização de Sensores**. 2006. Disponível em: https://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf. Acesso em: 15 mai. 2023.

QUEIROZ, Mariana Emer de. **ELABORAÇÃO E IMPLEMENTAÇÃO DE UM PROTÓTIPO DE ESTEIRA SELETORA POR COR.** 73 f. Campo Mourão, 2017. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/6082>. Acesso em: 15 maio 2023.

SANTOS, Jackson Henrique Aranha dos. **SENSORES DE COR: UMA ABORDAGEM TEÓRICA.** 2011.f30. Paraíba, 2011. Disponível em: <<http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/17567>>. Acesso em: 15 mai. 2023.

SIHOMBING, Poltak et al. **The citrus fruit sorting device automatically based on color method by using tcs320 color sensor and arduino uno microcontroller.** In: **Journal of Physics: Conference Series.** IOP Publishing, 2019. p. 012064.

SILVA, Fabrício Dias da. **DESENVOLVIMENTO DE UMA PLATAFORMA PARA CARACTERIZAÇÃO DE CORES E ÍNDICE DE MATURAÇÃO EM GOIABA (Psidium guajava L.) UTILIZANDO UM MICROCONTROLADOR PROGRAMÁVEL ARDUÍNO.** 45f. Pombal/Pb, 2021. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/xmlui/handle/riufcg/21673>. Acesso em: 08 maio 2023.

SILVA, Wendley. **Como funcionam os sensores das câmeras fotográficas?** 20 set. 2010. 1 desenho. Disponível em: <https://fotobellarte.wordpress.com/2010/09/20/comofuncionam-os-sensores-das-cameras-fotograficas/>. Acesso em: 18 jun. 2023.

SILVEIRA, P. R. da.; SANTOS, W. E. **Automação e Controle Discreto.** Editora Erica, São Paulo, 1998.

SOUZA, Allan Clemente de. **DESENVOLVIMENTO DE SISTEMA DE AUTOMAÇÃO DE LAVAGEM DE FILTROS PARA ETA POR DUPLA FILTRAÇÃO MODIFICADA.** 99 f. Fortaleza, 2016. Disponível em: <https://repositorio.ufc.br/handle/riufc/36335>. Acesso em: 10 maio 2023.

TAOS LUMENOLOGY. Datasheet: TCS230 **PROGRAMMABLE COLOR LIGHT TO FREQUENCY CONVERTER.** 2008. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/202765/TAOS/TCS230.html>. Acesso em: 06 abr. 2023.

THOMAZINI, Daniel; DE ALBUQUERQUE, Pedro Urbano Braga. **Sensores industriais: fundamentos e aplicações.** Saraiva Educação SA, 2013.

APÊNDICE A – MANUAL DE USO DO APLICATIVO

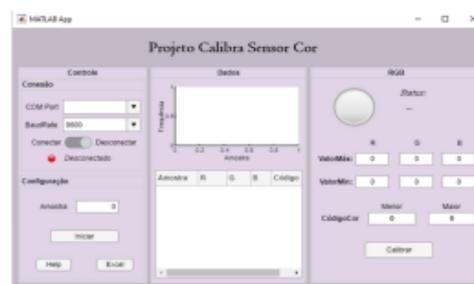
Manual: Como Utilizar o Projeto Calibra Sensor

Bem-vindo ao aplicativo Projeto Calibra Sensor! Este manual de instruções fornecerá informações detalhadas sobre como usar o aplicativo. Nós recomendamos que você leia atentamente todas as instruções antes de começar a usar o aplicativo. O Projeto Calibra Sensor é uma plataforma que auxilia os usuários a calibrar um módulo sensor de cor, visualizando os dados recebidos e com a opção de enviar os dados para o Excel.

Interface do Usuário e Como utilizar

O aplicativo é fácil de usar e possui uma interface intuitiva. Ao abrir o aplicativo, você será direcionado para a tela inicial, onde poderá encontrar as seguintes funcionalidades:

- "Painel de Controle": Conecta ao Arduino e cria a quantidade de amostra que deseja analisar.
- "Painel de Dados": Visualize os dados recebidos por meio do gráfico e da tabela.
- "Painel RGB": Te auxilia a captar os máximos e mínimos da frequência RGB.



1. Conecte-se ao Arduino

Antes de iniciar a coleta de dados, você precisa estabelecer uma comunicação entre o Aplicativo e o Arduino usando o botão "Conectar".

Certifique-se de que o LED vermelho está verde, e se o texto desconectado passou a ser conectado.



Da mesma forma, quando terminar de usar o programa, você deve usar o botão "Desconectar" para encerrar a comunicação.

2. Criando uma Base de Dados

Em seguida, na seção "configuração", crie a quantidade de amostras (leitura de frequência RGB emitidas pelo sensor) que deseja coletar para análise. Clique no botão "Iniciar" e o processo é iniciado automaticamente.

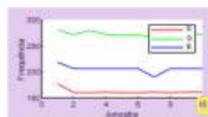


- O botão "Help" no canto inferior esquerdo da tela, ajuda o entendimento do app.
- Selecione "Excel" e crie um banco de dados em uma Planilha da Microsoft Excel para análises que vai além, das funções deste aplicativo.

3. Visualização dos Dados

- O gráfico plota os dados de Frequência X Amostra.

Os dados estarão completamente coletados quando o número de amostra solicitado está presente no eixo X do gráfico.



No eixo Y é possível visualizar os intervalos que R, G e B se encontram.

- Através da tabela também é possível visualizar e analisar os dados.

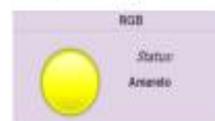
Amo...	R	G	B	Cód
1	177	280	218	
2	180	271	208	
3	180	279	208	
4	181	271	208	
5	180	270	208	
6	180	271	208	
7	181	265	190	

4. Painel para auxiliar na calibração

- Ao pressionar o botão calibrar os valores máximos e mínimos são mostrados para R, G e B.

	R	G	B
ValorMár:	177	280	218
ValorMín:	180	265	190

- A lâmpada apresenta a cor do objeto detectado, caso esteja de acordo com a programação do Arduino. É no campo 'Status' o nome da cor. Caso contrário a lâmpada permanece na cor padrão do aplicativo e o 'Status' será No Color.



- O processo de calibração será bem sucedido quando os campos de "CódigoCor" apresentar valores idênticos.



Essas são apenas algumas das maneiras pelas quais você pode usar para aumentar sua produtividade.

**APÊNDICE B – EXEMPLO DE TABELA CRIADA PARA ANÁLISE DE DADOS EM
TESTE I: PEÇAS VERMELHAS**

Amostra	Peça Vermelha 1			Peça Vermelha 2			Peça Vermelha 3			Peça Vermelha 4			Peça Vermelha 5		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	149	272	206	151	271	199	147	271	199	148	261	197	157	272	207
2	154	266	205	146	271	205	153	271	205	150	262	196	151	273	202
3	154	273	207	152	271	206	153	267	207	148	259	195	159	269	208
4	156	275	199	153	273	207	155	269	207	147	258	195	157	272	206
5	155	272	206	153	268	206	154	272	206	147	259	195	151	272	207
6	139	266	206	152	271	199	153	271	205	148	255	197	158	269	209
7	154	267	207	145	271	205	147	271	200	149	261	196	159	273	207
8	156	275	207	152	265	205	155	266	206	148	259	195	157	272	207
9	155	272	200	151	272	206	153	265	206	148	259	195	157	273	207
10	148	271	205	154	274	201	154	274	207	141	259	193	153	276	208
Média	152	271	205	151	271	204	152	270	205	147	259	195	156	272	207
Máximo	156	275	207	154	274	207	155	274	207	150	262	197	159	276	209
Mínimo	139	266	199	145	265	199	147	265	199	141	255	193	151	269	202