



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

RENAN YUI HEBITA

Sistema Embarcado aplicado para Automação de Order Picking

Sorocaba

2022

RENAN YUI HEBITA

Sistema Embarcado Aplicado para Automação de Order Picking

Trabalho de Conclusão de Curso
apresentado ao Instituto de Ciência e
Tecnologia de Sorocaba, Universidade
Estadual Paulista (UNESP), como parte
dos requisitos para obtenção do grau de
Bacharel em Engenharia de Controle e
Automação.

Orientador: Prof. Dr. Everson Martins

Sorocaba

2022

H446s Hebita, Renan Yui
Sistema Embarcado aplicado para Automação de Order Picking /
Renan Yui Hebita. -- Sorocaba, 2022
49 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de
Controle e Automação) - Universidade Estadual Paulista (Unesp),
Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Everson Martins

1. Sistemas Embarcados. 2. Arduino. 3. Filtro Digital. 4. Simulink.
5. Sensor óptico. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de
Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

SISTEMA EMBARCADO APLICADO PARA
AUTOMAÇÃO DE ORDER PICKING

RENAN YUI HEBITA

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE
DO REQUISITO PARA OBTENÇÃO DO GRAU DE
BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof. Dr. Everson Martins
Coordenador

BANCA EXAMINADORA:

Prof. Dr. Everson Martins
Orientador/UNESP - Campus de Sorocaba

Prof. Dr. Ivando Severino Diniz
UNESP - Campus Sorocaba

Prof. Dr. Mauricio Becerra Vargas
UNESP - Campus Sorocaba

Dezembro de 2022

Dedico este trabalho aos professores que auxiliaram em minha formação, a família pelo amor, apoio, incentivo, paciência e colaboração ao longo deste caminho de conhecimento.

AGRADECIMENTOS

Agradeço aos meus pais Celso e Janet pela minha vida como um todo, desde o meu nascimento até a eternidade estarei agradecido a tamanho amor e carinho incondicional e verdadeiro que conheci por eles. Graças a eles incontáveis números de oportunidades surgiram em meu caminho até hoje, junto sempre a aconselhamentos, apoio e paciência.

À minha irmã pelos incentivos e conselhos sempre visando o melhor para seu irmão.

Ao meu orientador, Prof Everson, por todo o tempo e paciência e em todo o trajeto demonstrou disponibilidade em me auxiliar, preocupação nas datas de entrega e calma com a minha pessoa.

Aos meus amigos de infância e da universidade sempre presentes até agora, me trouxeram grandes alegrias, significado à vida e companheirismo. Irmãos aos quais sei que em momentos alegres ou tristes estarei bem acompanhado.

E a todos em minha volta que de alguma maneira auxiliaram no desenvolver deste projeto, tornando-o possível, meus agradecimentos.

HEBITA, Renan Yui **Sistema Embarcado aplicado para Automação de Order Picking** / Renan Yui Hebita. -- Sorocaba, 2022 49 p. : il., tabs. Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba, 2022.

RESUMO

O trabalho apresentado trata-se de um projeto de um sistema auxiliar na contagem, entrega e organização de uma etapa de linha de produção enxuta de forma automática utilizando um sistema embarcado como central de controle. Através de estudos deste dispositivo de controle, tipos de sensores de interesse, análise de dados recebidos dos sensores e softwares de programação auxiliar MatLab e Simulink, foi possível desenvolver um sistema automático contendo um filtro de sinais digitais para a recepção de determinados produtos. A montagem do produto deste projeto foi integrada a partir da placa de controle Arduino Uno, leds, sensores e central de controle via computador. Para a resolução deste sistema foi desenvolvido um algoritmo baseado em médias móveis dentro do próprio ambiente de desenvolvimento integrado (IDE) da placa Arduino junto a possíveis alternativas de algoritmo para o processamento em Simulink. O desempenho do filtro digital do projeto foi registrado em gráficos e satisfaz as condições de simulação inseridas em uma linha de produção. O trabalho logrou êxito em fornecer um meio automático de processamento a receber um número de pedidos e encaminhar o valor exato para a etapa de transporte.

PALAVRAS-CHAVE: Sistemas Embarcados. Arduino. Simulink. Filtro digital. Sensor Óptico.

HEBITA, Renan Yui **Embedded System applied to Order Picking Automation** / Renan Yui Hebita. -- Sorocaba, 2022 49 p. : il., tabs. Completion of course work (Bachelor - Control and Automation Engineering) - São Paulo State University (Unesp), Institute of Science and Technology, Sorocaba, 2022.

ABSTRACT

The work presented is a project of an auxiliary system in the automatic counting, delivery and organization of a stage of a lean production line using an embedded system as a control center. Through studies of this control device, types of sensors of interest, analysis of data received from sensors and auxiliary programming software MatLab and Simulink, it was possible to develop an automatic system containing a filter of digital signals for the reception of certain products. The product assembly of this project was integrated from the Arduino Uno control board, leds, sensors and control center via computer. To resolve this system, an algorithm based on moving averages was developed within the integrated development environment (IDE) of the Arduino board along with possible algorithm alternatives for processing in Simulink. The performance of the digital filter of the project was graphed and satisfied the simulation conditions inserted in a production line. The work was successful in providing an automatic means of processing to receive a number of orders and forward the exact amount to the transport stage.

KEYWORDS: Embedded systems. Arduino. Simulink. Digital Filter. Optic Sensor.

LISTA DE FIGURAS

Figura 1 - Diagrama de blocos do processo em foco do projeto.....	16
Figura 2 - Ilustração de funcionamento de order picking em contexto de uma linha de produção.....	17
Figura 3 - Montagem de sistema de processamento para contagem de objetos.....	18
Figura 4 - Esquema de sensor óptico em barreira (transmissão).....	21
Figura 5 - Esquema de sensor óptico por difusão.....	22
Figura 6 - Esquema de sensor óptico por reflexão.....	22
Figura 7 - Diagrama de blocos de filtro digital para filtrar um sinal de tempo contínuo.....	25
Figura 8 - Sensor TCRT500.....	28
Figura 9 - Destaque das entradas e saídas do Arduino UNO.....	29
Figura 10 - Switch(chave) de 4 conexões comuns.....	30
Figura 11 - Potenciômetro de 100k.....	31
Figura 12 - Add-on de comunicação Matlab-Arduino.....	32
Figura 13 - Add-on de comunicação Simulink-Arduino.....	32
Figura 14 - Inclusão de biblioteca adicional Arduino.....	33
Figura 15 - Esquemático do circuito de testes com potenciômetro. Conexão do Sinal em laranja, Alimentação em vermelho e Terra em preto.....	33
Figura 16 - Blocos de Programação em Simulink.....	35
Figura 17 - Esquemático de circuito de seleção order picking.....	35
Figura 18 - Diagrama de blocos da lógica de programação.....	36
Figura 19 - Sinais gerados em Simulink, com $k=5$	37
Figura 20 - Sinais gerados em Simulink, com $k=10$	38
Figura 21 - Sinais gerados em Simulink, com $k=25$	38
Figura 22 - Sinais Gerados em IDE Arduino, com $k=5$, onde observa-se maior defasagem comparado aos resultados em Simulink.....	39
Figura 23 - Sinais Gerados em IDE Arduino, com $k=10$, onde observa-se melhor redução da componente de ruído de sinal.....	39
Figura 24 - Sinais Gerados em IDE Arduino, com $k=25$, onde observa-se a defasagem de aproximadamente meio período de onda.....	40

LISTA DE TABELAS

Tabela 1 - Comparativa entre sistema biológico e eletrônico.....	19
--	----

LISTA DE SÍMBOLOS

A	Ampère
C++	Linguagem de Programação C++
CA	Corrente Alternada
CC	Corrente Contínua
f	Frequência
GND	Terra
Hz	Hertz
I	Corrente
IDE	Ambiente de Desenvolvimento Integrado
IR	Infra-Red (Infra-vermelho)
MM	Média Móvel
M_T	Média simples
SMM	Sistema Média Móvel
t	Tempo
T	Período
USB	Universal Serial Bus
V	Volts
x(t)	Sinal de entrada contínuo
x[n]	Sinal de entrada discreto
y(t)	Sinal de saída contínuo
y[n]	Sinal de saída discreto
p	Período de amostragem
ω	Frequência angular tempo contínuo
Ω	Frequência angular tempo discreto

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Considerações Iniciais	13
1.2 Objetivos	14
1.2.1 Gerais	14
1.2.2 Específicos	14
1.3 Justificativa	15
2. FUNDAMENTAÇÃO TEÓRICA	16
2.1 Automação	18
2.2 Sensores	20
2.2.1 Sensores ópticos	20
2.2.2 Sensor óptico por barreira	21
2.2.3 Sensor óptico por difusão	21
2.2.4 Sensor óptico por reflexão	22
2.3 Sinais	22
2.3.1 Classificação de sinais	23
2.3.2 Processamento de sinais	23
2.4 Filtros	24
2.5 Microcontroladores	26
3. MATERIAL E MÉTODOS	27
3.1 Componentes	27
3.1.1 Sensor TCRT500	27
3.1.2 Arduino UNO	28
3.1.3 Softwares utilizados	29
3.1.4 Chave	30
3.1.5 Potenciômetro	30
3.1.6 Protoboard	31
3.1.7 LEDs e Cabeamento	31
3.2 Montagem	31
4. RESULTADOS E DISCUSSÃO	37
5. CONCLUSÃO	41

REFERÊNCIAS	42
APÊNDICE A	45
APÊNDICE B	46

1 INTRODUÇÃO

1.1 Considerações Iniciais

Compreende-se por automação atualmente a quaisquer sistema em que se é auxiliado/apoiado em computadores substituindo o trabalho humano em prol de algum aspecto de rendimento da produção, seja segurança, qualidade, quantidade, agilidade ou redução dos custos. Além disso, na automação assume-se a implementação de sistemas interligados e assistidos por meios de comunicação em rede integrando sistemas supervisórios e interface de usuários de modo a auxiliar a interação com o operador em suas análises (MORAES, 2010).

Dentro da automação existe uma auto adaptação a diferentes condições, em grande parte seguindo o seguinte padrão: instrumentos de medidas fornecem sinais ao computador, tais sinais são comparados aos valores ideais e realizar operações matemáticas com a finalidade de se chegar aos resultados desejados e conduzindo a uma ótima produção (ROSÁRIO, 2009).

Em prática todos os aspectos de nossas atividades diárias tem correlação com algum tipo de sistema de controle. Principalmente por estarem presentes em grande escala em todos os setores da indústria (GOLNARAGHI, M.F., 2012).

Este trabalho é focado em uma determinada seção da linha de produção a qual em um armazenamento de peças se é selecionada uma determinada peça, definido a quantidade necessária dela para a produção de um produto ou mesmo a própria ser o produto final e feito seu encaminhamento de transporte para o solicitante de maneira automática.

Tal processo de retirada de produtos em estoque em função de uma solicitação é denominado ‘coleta de pedidos’ ou *'order picking'*. Neste processo há um consumo de tempo e exige mais mão de obra caso feito de maneira manual ou então a demanda de maior capital em um sistema automatizado (MORAES, 2019). O intuito deste projeto é viabilizar o processo de uma maneira automática mais enxuta e baixo custo, sendo possível sua aplicação em pequenas indústrias ou estoques comerciais.

As macro etapas as quais o order picking deverá passar são: a interação do solicitante a máquina, seleção e contagem correta de itens, entrega de material e finalmente a resposta da máquina ao operador. Nas etapas de interação da máquina com o sistema físico deve-se desenvolver métodos aos quais o computador consiga interpretar as informações dos sistemas físicos.

Todos os sistemas físicos estão sujeitos a ruídos em sua operação, sejam eles intrínsecos do mecanismo ou externos. Dessa forma os sistemas de controle devem levar em conta tais distúrbios e serem projetados de modo a serem insensíveis ou atenuar ao máximo as perturbações (GOLNARAGHI, M.F., 2012).

O dispositivo capaz de eliminar componentes indesejáveis de um sinal é chamado de filtro (BORGES, 2018). O desenvolvido neste projeto trata-se de um filtro digital, que se tem o processamento dentro do computador, utilizando o valor de n amostras anteriores na determinação da saída (OPPENHEIM, 1999).

Com o desenvolvimento de circuitos integrados, foi possível o acesso a baixo custo dos microprocessadores dedicados (ROSÁRIO, 2009). E, conseqüentemente, aos sistemas embarcados, pequenos computadores aos quais é possível o processamento de um filtro digital como a placa Arduino utilizada neste projeto.

1.2 Objetivos

1.2.1 Gerais

Construção de um sistema de '*order picking*' o qual o usuário deve enumerar a quantidade de peças solicitadas e automaticamente seja efetuado a contagem e transporte correto dos produtos até a próxima etapa de produção. Tendo enfoque no tratamento de sinais analógicos recebidos a placa. Dessa forma, o trabalho pode contribuir com o desenvolvimento de novos meios de organização de materiais, reduzindo o custo de produção, tempo de deslocamento, qualidade de transporte e possível redução do espaço necessário para armazenamento

1.2.2 Específicos

- Implementar a comunicação analógica digital.
- Desenvolver o filtro digital em diferentes plataformas de software.
- Analisar os resultados obtidos de sinal original com o filtrado.
- Verificar a possibilidade de utilizar sensor IR como medição de distância
- Desenvolver uma aplicação de programação e baixo custo.
- Validar a utilidade prática do algoritmo desenvolvido.

1.3 Justificativa

O presente trabalho justifica-se pela necessidade e relevância no aprimoramento de etapas no meio de produção, ampliando as alternativas de alocação de itens em um estoque tal como de meios de tratamento de sinais, substituindo formas que antes eram compostas de componentes físicos para processamento digital.

O principal fator de estímulo ao estudo de automação nesta específica etapa na linha de produção foi a observação de um procedimento em comum a diversas empresas feito de maneira repetitiva e lenta. Sendo assim, na implementação deste projeto a automatização deste processo traria redução de custos e tempo às empresas que ainda optam pela busca manual de objetos.

2. FUNDAMENTAÇÃO TEÓRICA

Esta revisão bibliográfica foi alicerçada principalmente em livros acadêmicos encontrados na biblioteca da Universidade Estadual Paulista (UNESP) unidade Sorocaba, Julio de Mesquita Filho e Portal de repositórios institucional UNESP utilizando as palavras chave : “order picking” ; “Filtros digitais” ; “Arduino” ; “filtros de sinais” ; “analógico digital” ; “MatLab” ; “automação” sozinhas ou combinadas entre sí. Outros artigos e livros utilizados de embasamento teórico foram referenciados neste trabalho.

O projeto foi teorizado de forma a automatizar a etapa de order picking, como apresentada na seção anterior tem como objetivo a busca de produtos em um estoque a partir de uma solicitação de cliente, portanto este trabalho foi desenvolvido um meio de processamento de contagem a partir de um valor recebido e então enviado um sinal para etapas subsequentes de transporte como apresentado pelo bloco central do diagrama de blocos da Figura (1).

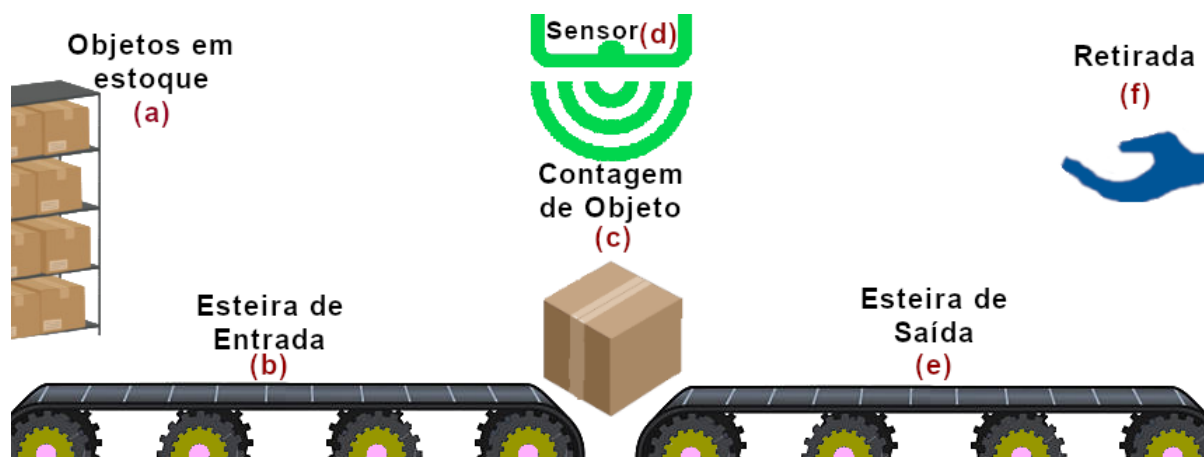
Figura 1 – Diagrama de blocos do processo em foco do projeto.



Fonte: Autoria própria.

Em exemplo deste processo em uma linha de produção, um determinado número de objetos é necessário para a produção de um produto. Tais objetos são mantidos em estoque como ilustrado na Figura 2(a), cabe a um sistema de controle reconhecer a quantia necessária para transporte. Inicialmente é feita a retirada do objeto por meio de uma esteira de entrada exclusiva para o setor Figura 2(b) e encaminhado até o sistema de contagem Figura 2(c) o qual auxiliado por sensores Figura 2(d) efetua o reconhecimento e controle da quantia desejada. Então o objeto deverá ser seguramente transportado por uma esteira geral Figura 2(e), até a área de retirada Figura 2(f).

Figura 2 – Ilustração de funcionamento de order picking em um contexto de uma linha de produção.

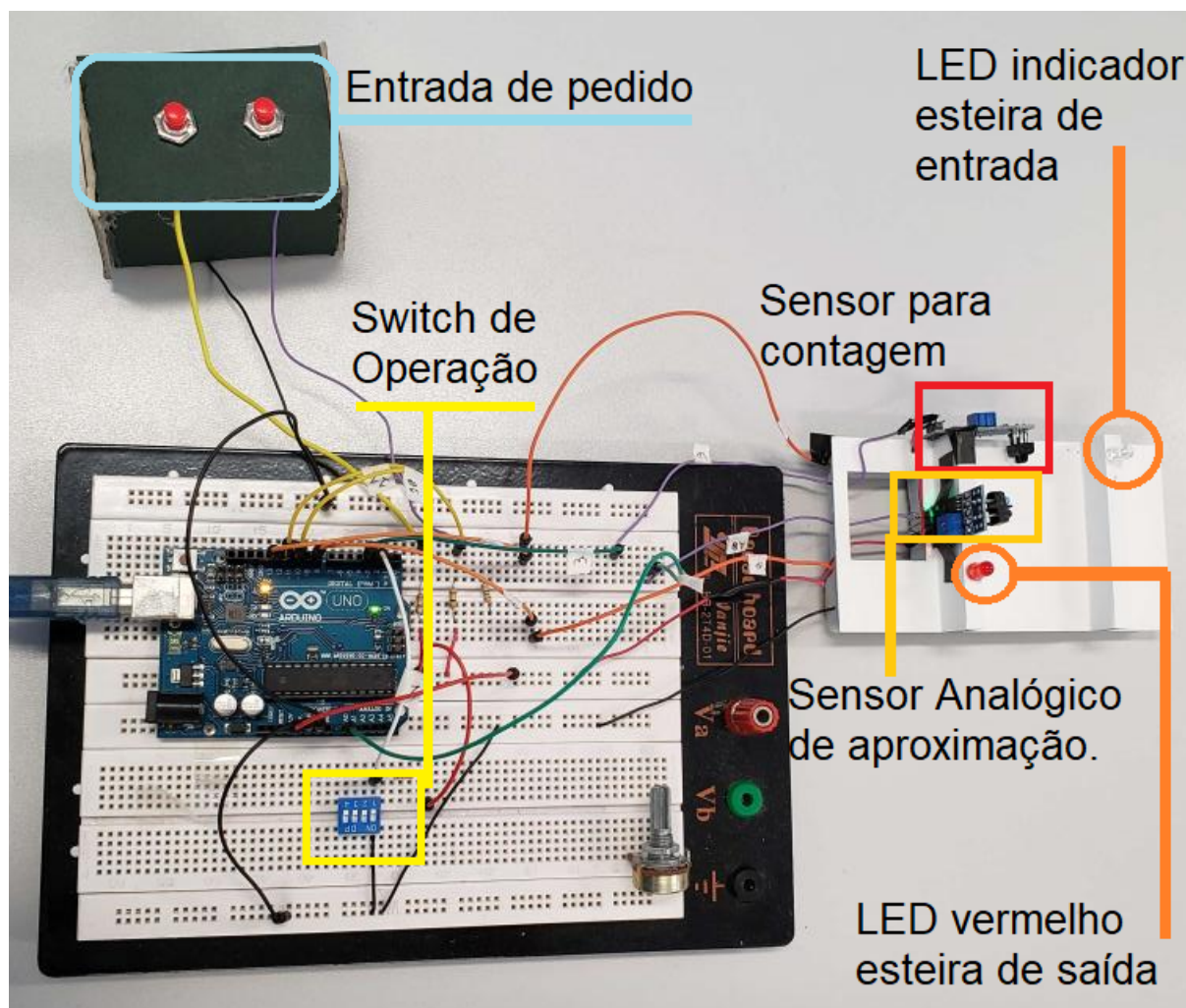


Fonte: Autoria própria.

A partir deste conceito de sistema acima foi planejado a montagem de um sistema no qual não se é voltado no modo o qual receberia os dados de entrega (quantidade de objetos) e nem o meio de transporte de produtos, mas sim o processamento dos dados de entrega em um sistema embarcado e sua comunicação com sensores para realizar a entrega da quantia exata requerida. Então a partir de dois botões, foi feita a entrada para a adição/redução de pedidos.

Enquanto a representação do sinal enviado a uma linha de transporte foi feita pelo acionamento de diferentes LEDs, sendo um verde para a esteira de entrada e um vermelho para a esteira de saída. Este sistema é apresentado na Figura (3) e possui em sua direita o posicionamento de 2 sensores ópticos Infravermelhos, o primeiro fará o papel de enviar sinais analógicos da intensidade de sinais sendo recebidos com a aproximação de objeto, para que o sistema saiba exatamente a localização em que ele se encontra do sensor evitando erros de posicionamento, enquanto o segundo enviará respostas em sinal digital alto ou baixo indicando a presença do objeto para sua saída e efetuar a contagem do mesmo.

Figura 3 – Montagem de sistema de processamento para contagem de objetos



Fonte: Autoria Própria

2.1 Automação

O ramo de engenharia de controle e automação é uma área multidisciplinar o qual temos um crescimento exponencial de possibilidades visto a grandes avanços tecnológicos em suas diferentes áreas principalmente em computação. É recorrente e essencial sua aplicação nos meios de produção industrial, em conjunto cada vez mais a automação se encontra presente no dia a dia de pessoas fora do chão de fábrica.

É registrado por Rosário (2009), que as primeiras formas de automação deram-se nas indústrias de processos, envolvendo maquinários de controle de medição elétrica e pneumática, mas somente ganhando relevância com o surgimento da máquina de comando

numérico em 1949/1950. Tendo a capacidade de realizar certas operações sem uma intervenção direta de operador.

Já o termo sistema de controle é especificado por Dorf e Bishop (2009) como a interconexão de componentes formando uma configuração de sistema que produzirá uma resposta desejada ao sistema. E descrevem também dois tipos de sistemas de controle:

- Malha aberta no qual utiliza-se um dispositivo de atuação para controlar o processo diretamente, sem o uso de realimentação
- Malha fechada no qual utiliza-se um tipo de medida na saída para realimentação do sinal e comparar com a saída desejada.

Uma comparação interessante entre um sistema biológico e mecatrônico é feito por Rosário (2005) a qual é apresentada a estrutura do corpo humano de aspecto funcional como a de componentes de máquina conforme a tabela 1:

Tabela 1 - Comparativa entre sistema biológico e eletrônico.

Sistema Eletrônico	Sistema biológico
Estrutura mecânica	Esqueleto
Condutores e cabos	Sistema sanguíneo
Alimentação de energia	Sistema digestivo
Transmissores de dado	Sistema nervoso
Atuadores	Músculos
Sensores	Órgãos sensoriais
Software e Base de dados	Conhecimento nas sinapses
Computador e Processador	Cérebro

Fonte: Rosario (2005)

Ou seja, para um dado sistema automatizado são necessárias diversas partes de diferentes funções interagindo entre si para o funcionamento de um todo.

Em exemplo a uma das partes, o atuador é definido por Dorf e Bishop (2009) como o dispositivo que fornece a força motriz para o processo, em exemplo um motor, luz, válvula, pistão...

O ato de se organizar os produtos, assim como diz o artigo de Dias (2014), também vem buscando novas formas de entrega na montagem de maneira correta, quantidade certa e menor tempo possível, esta etapa de manipulação do produto tem o termo em inglês “*order*

picking” o qual em sua maioria conta com funcionários indo até o estoque para separação de peças. Procura-se então neste projeto a automação da contagem de peças necessárias a um produto em prol de garantir a eficiência deste processo.

2.2 Sensores

“Um sensor pode ser definido como um transdutor que altera a sua característica física interna devido a um fenômeno físico externo - presença ou não de luz, som, gás, campo elétrico, campo magnético etc.” (ROSÁRIO, 2005, p. 55).

Da mesma maneira em que percebemos o mundo por meio de nossos sentidos, a máquina ao receber as informações de peças necessitará de meios a detectá-las, contando assim com a presença de sensores e transdutores, dispositivos encarregados de realizar a conversão de magnitudes físicas em dados elétricos. Silva e Costa (2009) definem como um sensor qualquer componente o qual lhe permita a análise de dada circunstância ao meio em que está inserido, seja luminosidade, temperatura, distância, etc. Enquanto Thomazini e Albuquerque (2007) adicionam a classificação dos sensores como transdutores, pela capacidade de transformar um tipo de energia em outra, em exemplo a luz em eletricidade.

Pode-se afirmar do ponto de vista da robótica que um robô o qual não possui um sensor é simplesmente uma máquina. Ou seja, os sensores são necessários ao robô para deixar de ser uma máquina qualquer e passar a ser um dispositivo que percebe o mundo a sua volta e reage a mudanças de situações (MARTIN, 1992).

2.2.1 Sensores ópticos

Conforme descritos tanto por Carneiro e Barbero (2014) como DJP Automação (2020), dentre os diversos tipos de sensores, temos os sensores ópticos, capazes de detectar um objeto sem que haja o contato mecânico. De maneira sucinta são compostos por um receptor e um emissor de feixe de luz, a partir do comportamento de objeto quando exposto ao feixe pode-se determinar sua distância, substância, material, cor entre outras características.

Dependendo de qual característica a se determinar é desejada podem funcionar com diversos tipos de luz: laser, vermelha ou infravermelha. Além dos tipos de luz os sensores podem variar em sua forma de operação, como são arquitetados, estes diferentes formatos

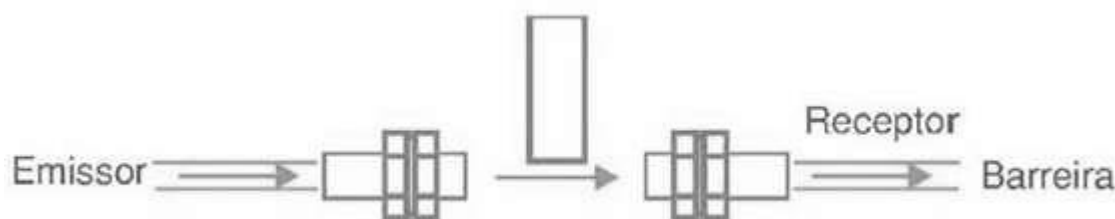
ampliam a gama de funcionalidades para este dispositivo. Três arquiteturas mais recorrentes são citadas por Cazer et al. (2005):

- Sistema por Barreira
- Sistema por Difusão
- Sistema por Reflexão

2.2.2 Sensor óptico por barreira

O sistema por barreira tem como característica a disposição de seu emissor e receptor alinhados frontalmente entre si em uma distância determinada, caso ocorra alguma interrupção da irradiação, irá desencadear um chaveamento do sinal eletrônico. Observa-se na figura 4 um esquema do sensor por barreira.

Figura 4 – Esquema de sensor óptico em barreira (transmissão).

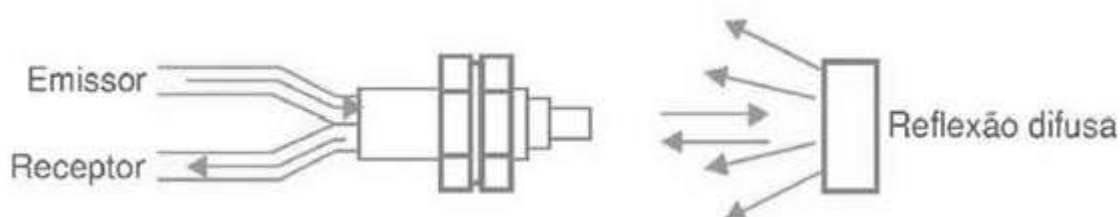


Fonte: SÁ, 2011.

2.2.3 Sensor óptico por difusão

Tanto emissor quanto receptor, no caso do sensor óptico de detecção difusa, ficam presentes do mesmo lado. “A luz emitida pelo emissor cria uma região ativa cuja presença de um objeto faz com que a luz seja refletida de forma difusa, de volta ao receptor, ativando o sensor.” Baptista (2002). Observado na Figura (5).

Figura 5 – Esquema de sensor óptico por difusão.



Fonte: SÁ, 2011.

2.2.4 Sensor óptico por reflexão

Neste caso também temos o emissor e receptor lado a lado, em contrapartida temos do outro lado posicionado frontalmente um refletor, o feixe de luz transmitido pelo emissor retorna ao receptor após refletido pelo refletor, caso tenha algum distúrbio causado por objeto um sinal é enviado, Figura (6).

Uma desvantagem deste tipo de sensor óptico é em casos onde há objetos em que podem refletir o próprio feixe, não ocorrendo a interrupção, ou mesmo quando há objetos translúcidos, claros ou brilhantes. Outro agravante é o caso de ofuscação do refletor devido a sujeiras acumuladas em sua superfície.

Figura 6 – Esquema de sensor óptico por reflexão



Fonte: SÁ, 2011.

2.3 Sinais

Dado a visão geral de um sistema automático, o enfoque do trabalho será direcionado aos tratamentos de sinais obtidos por meio de sensores acoplados. Em maior detalhamento a definição de sinal é formalmente definida como uma função de uma ou mais variáveis, a qual veicula informações sobre a natureza de um fenômeno físico. (HAYKIN ; VEEN, 2001)

Já de acordo com Oppenheim (1999) o termo sinal é aplicado a algo que transmite informação, podendo transmitir informações de estado ou comportamento de um sistema físico. Os sinais são sintetizados em prol de transmitir informações entre humano/máquinas.

2.3.1 Classificação de sinais

Os métodos de classificação de sinais deste trabalho são apresentados como descritos por Haykin e Veen (2001) sendo um deles baseado em como são definidos no tempo, em exemplo diz-se um sinal $x(t)$ é contínuo se ele for definido para todo o tempo. Eles podem ser gerados quando uma forma de onda física (exemplo: acústica ou luminosa) é convertida em sinal elétrico. Entretanto, quando temos um sinal em que se é definido em instantes isolados do tempo, temos um sinal de tempo $x[n]$ discreto, sendo em geral uniformemente espaçados.

Usualmente este tipo de sinal é gerado a partir da derivação de um sinal contínuo ao fazermos uma amostragem dele a uma taxa uniforme. Sendo Δt um período de amostragem, t o tempo e n um número inteiro, a amostragem de um sinal de tempo contínuo $x(t)$ no instante $t=n\Delta t$ gera uma amostra de valor $x(n, \Delta t)$, descrita por:

$$x[n] = x(n, \Delta t), \quad n = 0, \pm 1, \pm 2, \dots \quad (1.1)$$

2.3.2 Processamento de sinais

Haykin e Veen (2001) também nos apresentam as duas principais formas de implementação do processamento de sinais:

- Analógica ou tempo contínuo: como o nome implica o sistema recorre a elementos analógicos como diodos, resistores e capacitores. Tendo como principal atributo a capacidade natural de resolver equações diferenciais que descrevem o sistema físico. E independente do sinal de frequência de entrada nos fornece as soluções em tempo real.
- Digital ou tempo discreto: a qual recorre apenas a três elementos de computador digitais, os somadores, multiplicadores e memória. Tem suas operações dependentes de computação numérica, dependendo do tempo para executá-las, podendo ou não acompanhar as mudanças do sinal de entrada.

Ambos os tipos de processamentos serão abordados no desenvolvimento do projeto para a interação máquina e mundo físico.

2.4 Filtros

A aquisição de sinais de uma variável pelo tempo por meio de dispositivo analógico-virtual inevitavelmente está sujeita a ruídos de sinal, obtendo flutuações aleatórias na variável sendo medida. Assim como define Borges (2018) o dispositivo capaz de eliminar parte das componentes indesejáveis de um sinal é chamado de filtro. E ainda apresenta uma forma de classificar diferentes tipos de filtro de acordo com o comportamento para diferentes frequências:

- Passa altas: elimina componentes de menor frequência.
- Passa baixa: elimina componentes de maior frequência.
- Passa faixa: elimina as componentes com frequência acima ou abaixo de uma frequência de corte superior e inferior
- Rejeita faixa: elimina todas as componentes com frequência entre duas frequências de corte, uma inferior e outra superior
- Notch: elimina a frequência igual a frequência de corte.

Tais filtros usualmente são utilizados com filtros analógicos, e suas respostas estão no domínio do tempo contínuo. No entanto, ao tratarmos de sistemas em tempo discreto implementados com diferentes equações, temos outro tipo de categorização de filtros por (OPPENHEIM,1999) que são os filtros digitais, os quais executam uma equação utilizando o valor de n amostras anteriores na determinação da saída:

- FIR (finite pulse response): utiliza amostras somente de entrada.
- IIR (infinite impulse response): utiliza n amostras tanto de entrada como saída.

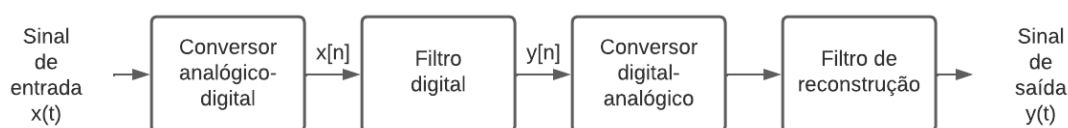
Os filtros digitais IIR (filtros digitais de resposta ao impulso de duração infinita) tem sua entrada e saída regidas por equações lineares de diferenças com coeficientes constantes de natureza recursiva. O uso de um filtro IIR em geral, dado para uma resposta em frequência prescrita, tem o comprimento menor ao de um FIR correspondente. Tal vantagem é somente adquirida às custas de distorção de fase e um transitório que não se limita em intervalo de tempo finito.

Os filtros digitais FIR (filtros digitais de resposta ao impulso de duração finita) tem sua operação regida por equações lineares de diferenças com coeficientes constantes de natureza não-recursiva. Uma propriedade inerente dos mesmos é de que eles podem realizar uma resposta em frequência com fase linear. Isso resulta em um retardo constante, dessa forma o problema da aproximação no projeto de filtros digitais FIR se torna simplificado.

Outra propriedade é o uso de memória finita e, portanto, qualquer transitório terá duração limitada. Tal tipo de filtro também pode implementar uma resposta em módulo desejada com resposta em fase exatamente linear. (HAYKIN; VEEN, 2001)

Um filtro digital opera por meio da *computação* e deve ser executado em um sinal contínuo. (HAYKIN; VEEN, 2001) O projetar de um filtro seletivo pode ser representado por um diagrama de blocos como mostrado na Figura (7):

Figura 7 – Diagrama de blocos de filtro digital para filtrar um sinal de tempo contínuo.



Fonte: Haykin; Veen, 2001.

O sinal de entrada é um sinal de tempo contínuo $x(t)$ e entra no bloco de conversor analógico-digital no qual é convertido em uma sequência $x[n]$ de números. O bloco de filtro digital por sua vez os processa em uma base de amostragem individual e produzindo uma nova sequência de números $y[n]$, sendo novamente convertido no bloco conversor digital-analógico para um sinal no tempo contínuo. E finalmente chegando a um filtro de reconstrução (passa-baixas) na saída do sistema resultando no sinal $y(t)$ de tempo contínuo.

Um método de filtro de sinais amplamente utilizado é o método de média móvel. O sistema de média móvel (SMM) possui implementação direta e fácil formulação, tanto em uso geral matemático, como em análises técnicas, ou em linguagem de programação. (de la Vega, 2018). Graças a estas características foi escolhido este sistema para a programação de nosso projeto.

A forma em que se baseia o sistema de média móvel é descrita por de la Vega (2018) como um sistema que tem a função de calcular o n -ésimo valor da sequência de saída $y[n]$ por meio da média simples de M_T valores da sequência (vetor na programação) de entrada $x[n]$. Definido matematicamente como:

$$y[n] = (x[n - M_2] + x[n - (M_2 - 1)] + \dots + x[n - M_1 + 1] + x[n - M_1]) / M_T \quad (1.2)$$

ou

$$y[n] = \frac{1}{M_T} \sum_{k=M_T}^{M_2} x[n - k], \quad (1.3)$$

onde $M_T = (M_2 - M_1 + 1)$, $M_2 > M_1$ e $\{M_2, M_1\} \in \mathbb{Z}$

Ademais, de la Vega (2018) nos introduz o conceito de sistemas não recursivos junto ao SSM, definindo-os como sistemas os quais o cálculo da sua variável de saída $y[n]$ não emprega a própria variável de saída. Sendo assim, o SMM é interpretado como um caso particular de sistema não recursivo.

2.5 Microcontroladores

Um meio de realizar a comunicação entre os sensores e o computador/sistema de controle é pelo hardware de aquisição de sinais, um destes amplamente aplicados devido ao menor custo são os microcontroladores, sendo um circuito integrado, podendo ser programado em diferentes lógicas para controle de sistemas (MURILO, 2021). Contém em si memória, processador e interface de entrada e saída no mesmo chip.

Os microcontroladores podem ser aplicados e adaptados nas mais diversas aplicações, no entanto assim como cita Bica (2019) eles requerem maior conhecimento de eletrônica e condicionamento de sinais por seus desenvolvedores, ainda assim, uma alternativa é a sua aplicação como componente em plataformas de prototipagem, uma destas mais utilizadas atualmente é o Arduino. Este aparelho possui uma IDE(Integrated Development Environment) a qual permite ao usuário escrever em linguagem de programação para definir as decisões lógicas de processamento.

3 MATERIAL E MÉTODOS

Neste tópico serão apresentados e especificados os materiais utilizados na montagem do sistema desejado Figura(3), e como foram integrados. A listagem destes componentes é a seguinte:

- Uma placa Arduino UNO R3
- Dois LEDs de cores distintas
- Dois resistores de 2k Ω
- Dois resistores de 100 Ω
- Dois sensores ópticos
- Uma chave de circuito
- Um potenciômetro
- Um cabo de conexão Arduino-USB
- Uma protoboard
- Fios e cabos de conexão

3.1 Componentes

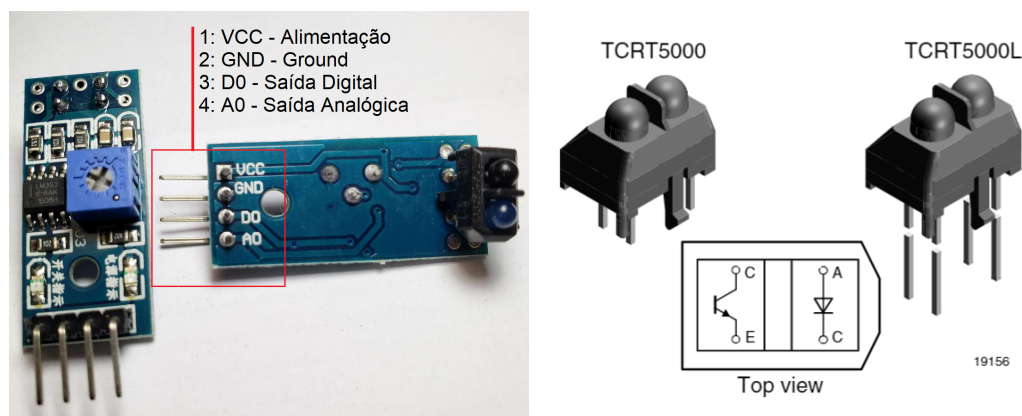
3.1.1 Sensor TCRT500

O sensor óptico utilizado para detectar a passagem de peças utilizará a luz infravermelha como detecção, sendo do modelo TCRT500, um modelo o qual é de conhecimento boa interação com o hardware de aquisição escolhido e devido ao seu baixo custo comparado a demais sensores ópticos. Sua lógica de funcionamento segue como visto anteriormente, um emissor enviará luz infravermelha e um receptor irá receber o sinal, dependendo da proximidade.

Este dispositivo é composto por quatro pinos como observado na Figura(8) Um para a alimentação indicado por VCC; outro para conexão com o terra GND (ground); e por fim os últimos são pinos de sinal, sendo o D0 para saída digital booleana significando que contém dois estados somente ligado e desligado (ON OFF) enquanto a saída A0 é a saída analógica enviando as nuances de sinal de aproximação do objeto. É também observado neste módulo um potenciômetro acoplado, ele auxilia na construção no protótipo pois regula a sensibilidade do sensor.

Sendo este dispositivo comparado a um sensor ultrasônico, menor, mais barato e podendo cumprir a medição de distância de objetos, optou-se pelo uso de 2 sensores IR de mesmo tipo neste projeto.

Figura 8 - Módulo de Arduino com Sensor TCRT500 e seu esquemático.

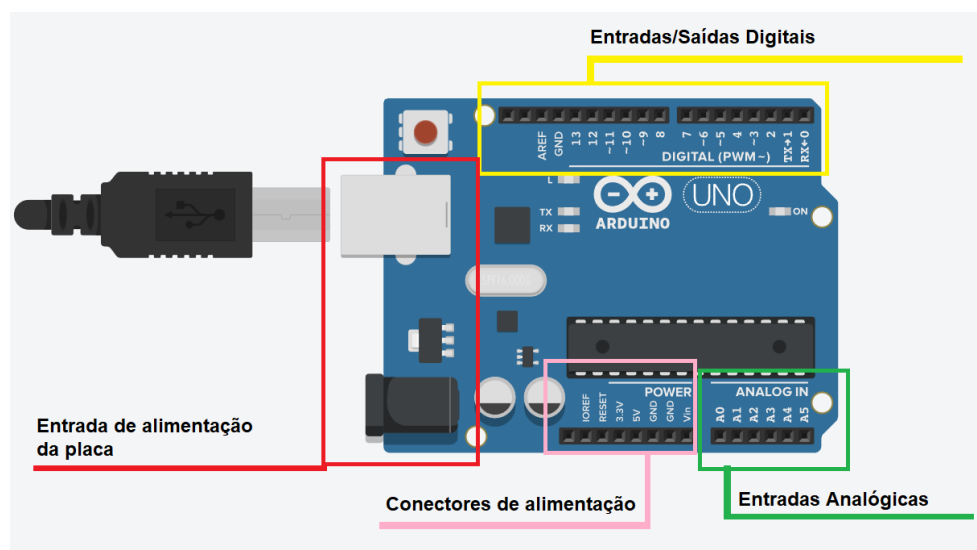


Fonte: Autoria própria e Datasheet TCRT500 - Reflective Optical Sensor with Transistor Output

3.1.2 Arduino UNO

Para o hardware de aquisição o qual fará a comunicação com o computador foi escolhida a placa Arduino UNO, devido ao fácil acesso, menor custo e entradas e saídas de pinos o suficiente para a aplicação projetada. Tem como especificação ser baseado no microcontrolador ATmega328P, um dispositivo de 8 bits, 2KB de RAM e 1KB de EEPROM. Possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um ressonador cerâmico de 16 MHz, um conector ICSP e um botão de reset. Figura(9)

Figura 9 - Destaque das entradas e saídas do Arduino UNO



Fonte: Autoria Própria.

A placa Arduino conta também com uma entrada USB para comunicação e alimentação, porém a alimentação também pode ser feita por outro conector de fonte externa de 2,1 mm. Deve-se ater às especificações de alimentação sendo recomendado o uso entre 7 a 12 volts, pois abaixo de 7 V o pino de alimentação de módulos 5V pode não ter energia o suficiente para suprir possíveis componentes, e caso acima de 12V há um risco de super aquecimento da placa.

Em sua lateral logo abaixo da alimentação de fonte externa como visto na Figura (9) localizam-se os pinos de saída para conexão de shields, alimentação de módulos acessórios ou mesmo um reset externo à placa.

3.1.3 Softwares utilizados

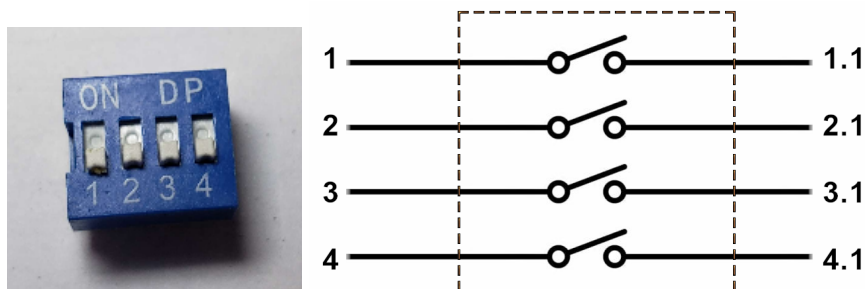
O software de programação da placa utilizado foi da própria empresa Arduino (IDE 2.0), sendo este o software de leitura, análise e apresentação de dados por meio de ferramentas Serial Plotter e Serial Monitor. Outro software de interesse é o Matlab que nos permite utilizar diferentes linguagens de programação como Python, C/C++, Fortran, Java, entre outras. Além disso, abrange a possibilidade de implementação do sistema com uma rede em nuvem ou IoT (Internet of Things) onde poderia se checar ou controlar os dados dos sistemas por meio remoto conectado à rede. Neste projeto foi utilizada uma de suas

ferramentas acopladas chamada Simulink, um outro ambiente de programação baseado em blocos, porém mantendo a comunicação com o próprio MatLab.

3.1.4 Chave

Um switch(chave) Figura (10) mini foi utilizado, tal dispositivo possui 4 contatos em cada lado, os quais ele irá dar contato ou interromper a ligação dependendo da posição da chave. Neste projeto somente um caminho foi necessário em sua aplicação.

Figura 10 - Switch (chave) de 4 conexões comuns.

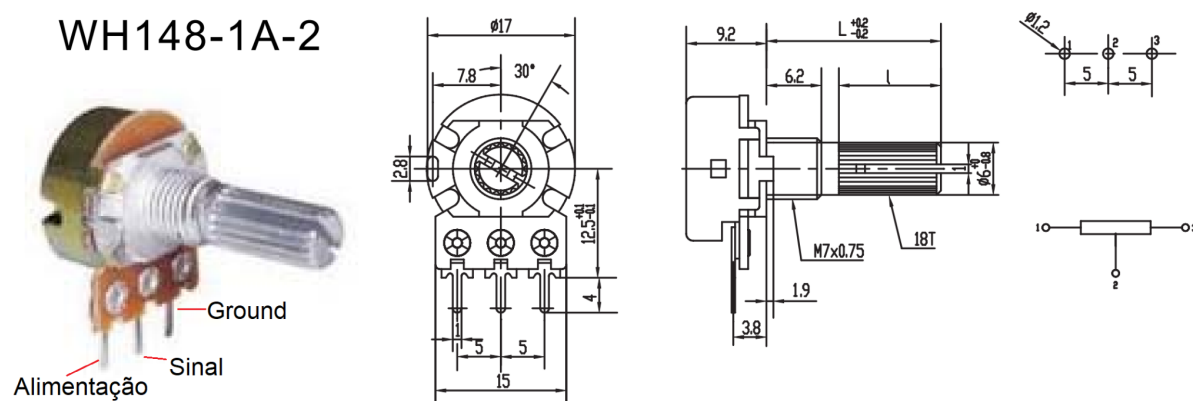


Fonte: Autoria própria

3.1.5 Potenciômetro

Este dispositivo conta com 3 pinos onde um será a alimentação, outro será o terra e por fim o pino central o sinal a ser mensurado Figura (11). Apesar de não ser utilizado na aplicação final este componente teve como finalidade a facilitar a visualização de sinais analógicos recebidos a placa Arduino por meio de diferentes programações desenvolvidas para filtros digitais. Sua operação se deve a um resistor circular interno o qual pode-se variar o valor pelo girar do pino metálico acoplado a ele, dando contato em diferentes pontos do resistor, sendo o valor mínimo de 0V e máximo de 100K para este modelo.

Figura 11 - Potenciômetro de 100k modelo WH148 e seu esquemático.



Fonte: Datasheet - 16mm Rotary Potentiometers

3.1.6 Protoboard

É um componente amplamente utilizado na área acadêmica, nela pode-se projetar diversos tipos de circuitos. Ela é uma placa contendo diversos orifícios de entrada a qual abaixo estão interconectados por linhas (5 entradas), enquanto as colunas centrais e laterais são interconectados por colunas (2 entradas) usualmente utilizadas no GND e VCC. Podendo variar como são suas posições dependendo do modelo de protoboard.

3.1.7 LEDs e cabeamento

As conexões dos componentes será feita com cabos finos adequados para uso em protoboard, enquanto os LEDs serão apenas indicativos de uma ativação de contato em relés para o acionamento de esteiras de locomoção.

3.2 Montagem

Inicialmente para introdução da comunicação das entradas arduino com MatLab e Simulink foi necessário a instalação de dois add-on de terceiros. O caminho para se realizar a instalação de add-ons é encontrar o ícone de Add-ons no menu de interface do MatLab, em seguida ‘*Get Hardware Support Package*’ O primeiro chamado **MATLAB Support Package for Arduino Hardware** (versão 22.2.0) Figura(12) para que a placa seja reconhecida e tenha comunicação com o software MatLab e o segundo chamado **Simulink Support Package for Arduino Hardware** (versão 22.2.0) Figura(13) o qual fará comunicação da placa com

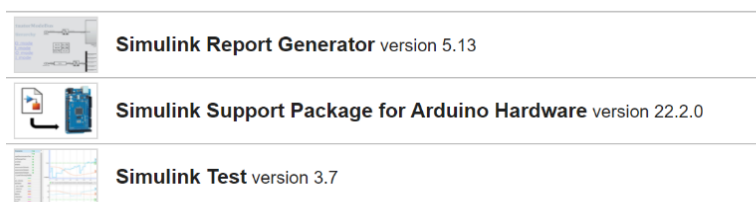
Simulink e adicionar novos blocos de função de interesse para nossos testes de implementação.

Figura 12 - Add-on de comunicação Matlab-Arduino



Fonte: Autoria própria

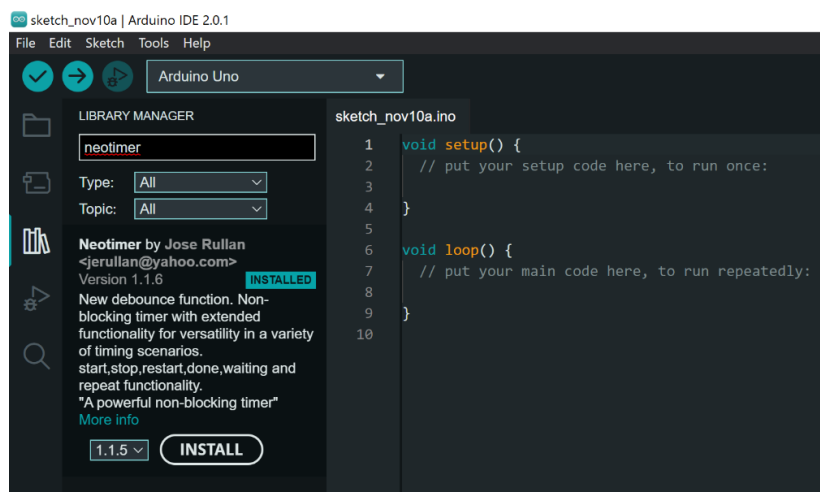
Figura 13 - Add-on de comunicação Simulink-Arduino



Fonte: Autoria própria

Outro passo importante para o projeto é a instalação de uma nova biblioteca ao IDE do Arduino chamada **Neotimer** (versão 1.1.6). Para sua instalação, deve-se localizar a aba ‘Tools’ no menu do deste software e então clicar em ‘*Manage Libraries...*’ ou então utilizar do atalho Ctrl+Shift+I do teclado. Na área de pesquisa ao digitar Neotimer será a única opção encontrada, Figura (14) biblioteca de comandos feita por Jose Rullan. Esta Biblioteca tem uma importante função de auxiliar nos delays de componentes sem a necessidade de todo o comando ter de aguardar toda execução de timer.

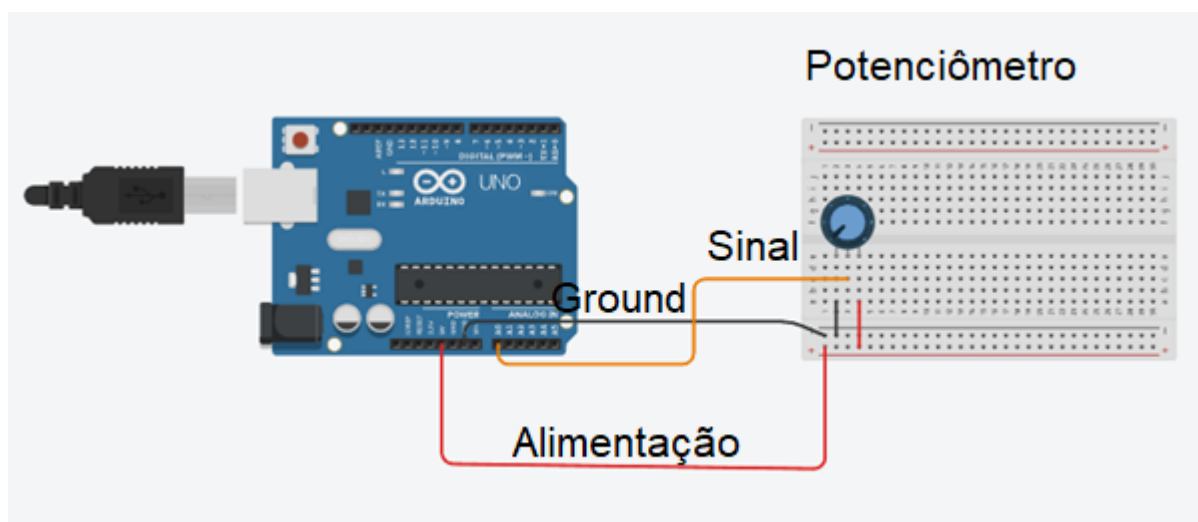
Figura 14 - Inclusão de biblioteca adicional Arduino.



Fonte: Autoria própria

Para os testes iniciais de leitura de entradas analógicas apenas um potenciômetro fora acoplado ao Arduino UNO conforme apresentado na Figura (15). Então seguindo a lógica de programação e cálculos de média móvel foi escrita a programação na própria IDE do Arduino conforme anexo(C) onde é criado um vetor de tamanho determinado para memorizar os valores de uma faixa do sinal e então retirar-se o valor médio deste período, esperando-se atenuar os ruídos de sinal, como um filtro digital deve operar.

Figura 15 - Esquemático do circuito de testes com potenciômetro. Conexão do Sinal em laranja, Alimentação em vermelho e Terra em preto.



Fonte: Autoria Própria.

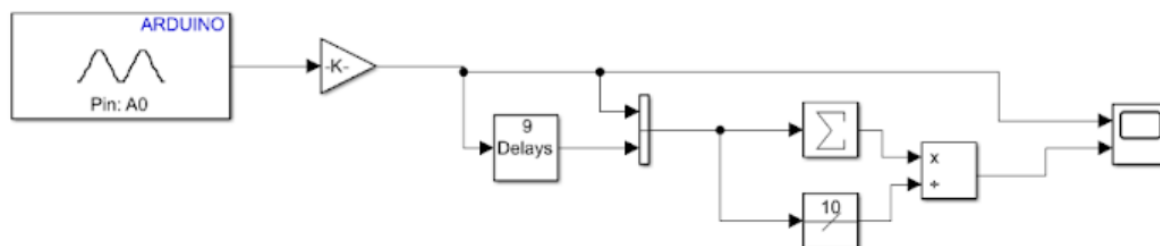
Tanto o sinal original de entrada como o de saída foram impressos no mesmo gráfico para melhor visualização do efeito de filtro digital de média móvel.

O desenvolver de um possível filtro em Simulink do MatLab é demonstrado pela Figura (16) a qual não foram necessárias escritas de código, uma vez que o próprio software possui blocos de função tanto para entradas de Arduino, como de saídas, leitor de sinais, operações lógicas e matemáticas. No entanto, cabe ao programador alocá-los de forma a executar as ordens de operações corretas e definir dentro dos blocos as suas configurações corretas.

Neste trabalho os blocos utilizados foram:

- Analog Input: Entrada de sinal analógica da placa arduino. Deve-se definir qual pino está a utilizar e o tempo de amostragem.
- Gain: Ganho do sinal. Deve-se definir o ganho de sinal para o gráfico de voltagem, uma vez que o pino de Arduino tem um sinal variante de 1-1024 a configuração utilizada foi de 5/1023 para se obter um gráfico de até 5 volts de sinal.
- Tapped Delay: Delay onde define-se o número de amostras em certo intervalo de sinal.
- Vector Concatenate: Cria o vetor de interesse para a média móvel.
- Sums of elements: Soma o valor dos elementos contidos no vetor.
- Width: Busca o tamanho máximo do vetor.
- Product: Realiza a operação matemática com a soma e números de elementos para média móvel.
- Scope: Gera a interface gráfica do Simulink para análise dos sinais. Deve ser configurado para mostrar as legendas, label, cor, e estar de acordo com o tempo de amostragem.

Figura 16 - Blocos de programação em Simulink

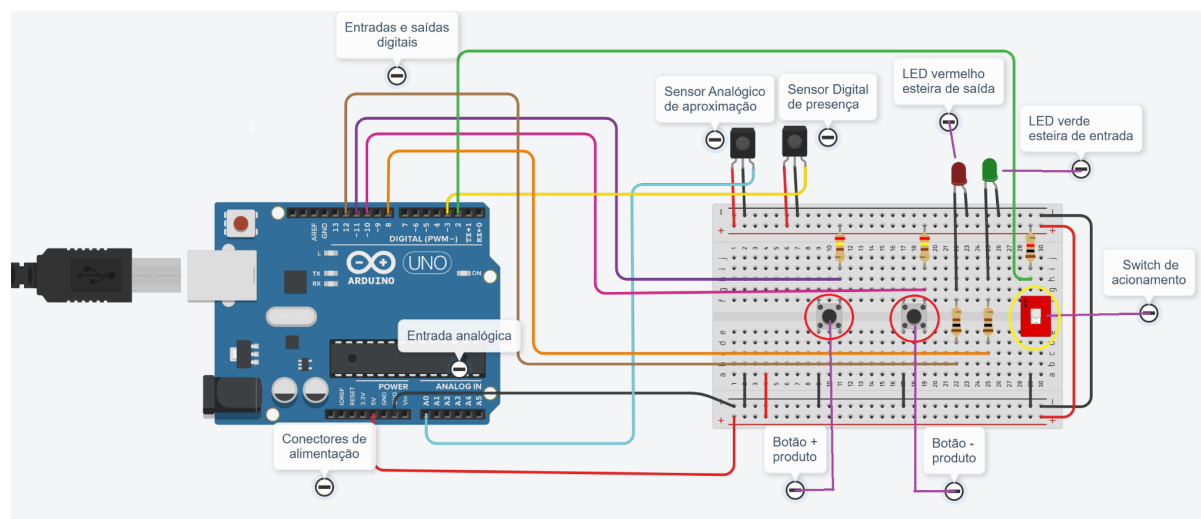


Fonte: Autoria própria

Uma vez definidos os valores de amostragem para média móvel, o circuito para a implementação em um sistema de contagem e busca de materiais foi montado conforme a Figura (17). Sua sequência de operação é apresentada no diagrama de blocos da Figura (18). O operador deve primeiro utilizar os dois botões para ajustar o número de componentes desejados para a busca, o primeiro adiciona uma unidade ao contador e o segundo reduz.

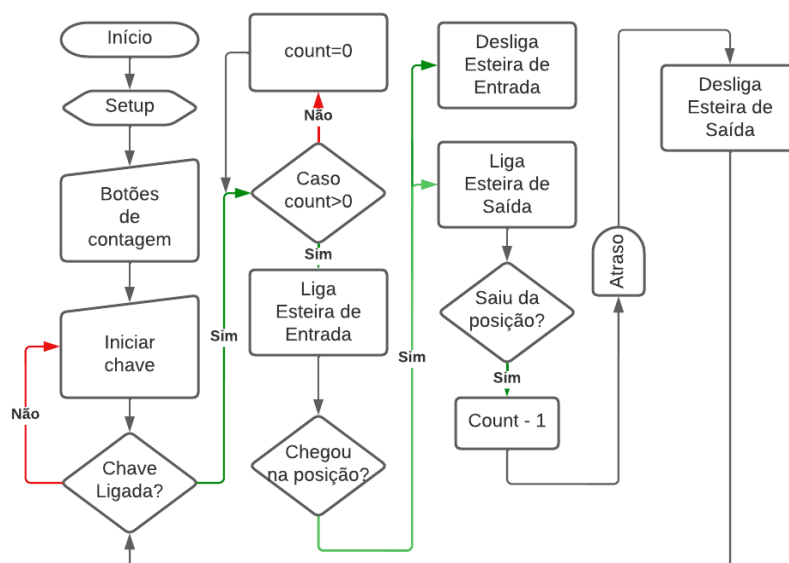
Após definido o número desejado no contador é ligada a chave de contato onde a primeira esteira (representada por um led) trará um componente.

Figura 17 - Esquemático de circuito de seleção order picking.



Fonte: Autoria própria

Figura 18 - Diagrama de blocos da lógica de programação.



Fonte: Autoria própria

Dois sensores ópticos estão posicionados para a leitura de presença, sendo um para medidas em sua saída analógica com o filtro digital, este terá a função de um sensor de aproximação e terá maior sensibilidade ajustada em seu potenciômetro embutido, e o segundo com menor sensibilidade e leitura de saída digital para realizar comportamento em valores booleanos, fornecendo o disparo para a contagem de itens já encontrados.

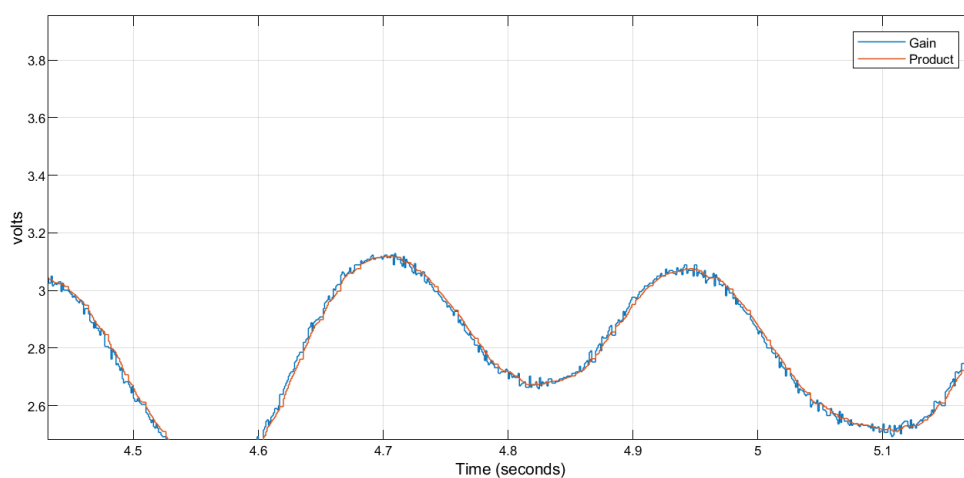
Quando o objeto chega à posição dos sensores infravermelho a primeira esteira deve parar e então a segunda esteira (representada pelo segundo led) ativará e subtrai-se uma unidade da contagem.

A programação deste sistema é detalhada pelo anexo (B). Foi realizada toda em linguagem de programação para a IDE de Arduino. É incluída a biblioteca Neotimer para o uso apropriado de botões de contagem sem a utilização de delays padrões. Todos os pinos são endereçados em variáveis para caso necessite a alteração de pinagem possa ser feita a alteração em apenas uma linha de código. Uma única variável de controle para a contagem é definida, sendo as demais variáveis inteiras auxiliares para a construção de troca de estados de componentes. Para a construção de filtro a mesma função de média móvel, anteriormente utilizada com o potenciômetro, é aplicada neste algoritmo. Uma cadeia de 'se' ('if') é utilizada para determinar as etapas condicionais de programa, a qual é aplicada em laço contínuo, seguido das definições da apresentação de dados e gráficos.

4 RESULTADOS E DISCUSSÃO

Dentro dos testes de programação de um filtro de média móvel para um potenciômetro foram colhidos diferentes gráficos, variando-se o tamanho do vetor gerado, o qual retém valores de sinal analógico para a obtenção da média. Como dentro do abordado em estudo, sua variação afeta no desempenho de filtragem e em sua defasagem. Inicialmente são apresentados os gráficos obtidos do filtro feito em Simulink nas Figuras seguintes.

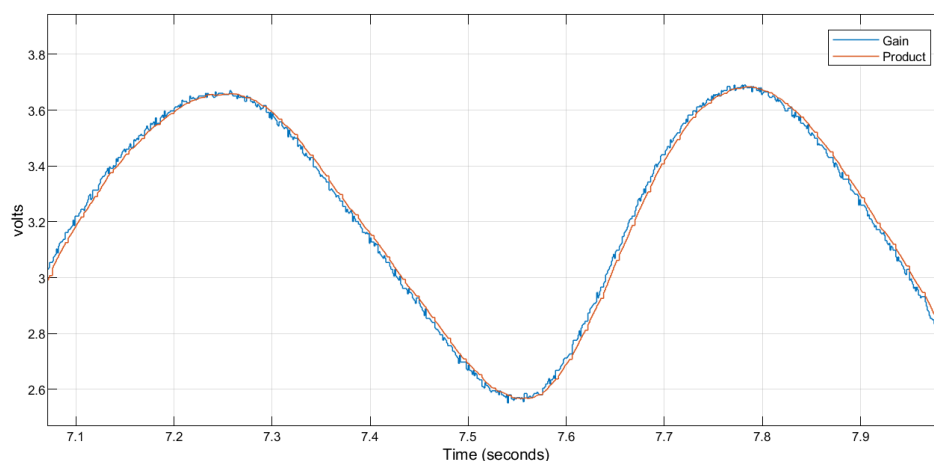
Figura 19 - Sinais Gerados em Simulink, com $k=5$.



Fonte: Autoria própria

Em um primeiro momento na Figura(19) foi efetuado a filtragem em simulink utilizado um vetor de tamanho 5 o qual demonstrou um bom contorno (laranja) do sinal original (azul), no entanto nota-se também o acompanhamento em maiores picos de ruídos do sinal requisitando assim um melhor refinamento de filtragem.

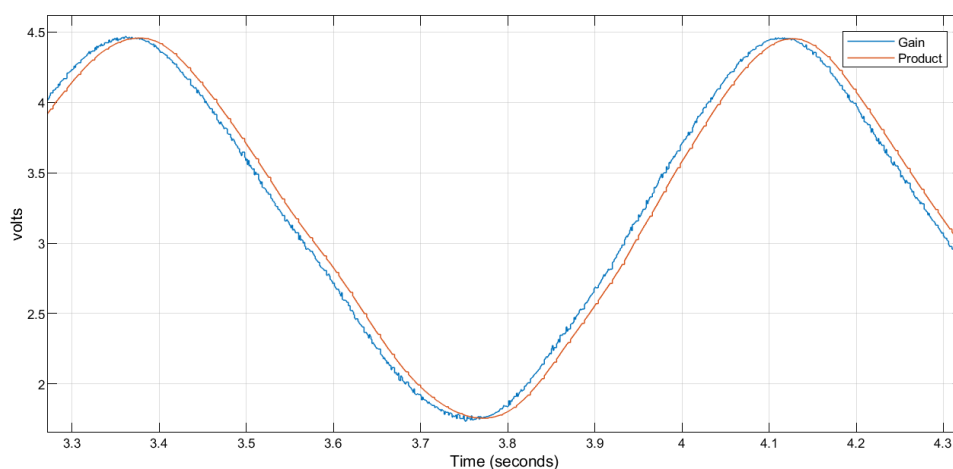
Figura 20 - Sinais Gerados em Simulink, com $k=10$.



Fonte: Autoria própria

Ao aumentarmos o tamanho de vetor k para 10 Figura (20) a filtragem de sinal recebe uma maior atenuação de ruídos, sem grandes variações de valores como o original, começamos a notar leve defasagem de sinal, porém não tão significativa.

Figura 21 - Sinais Gerados em Simulink para leitura de potenciômetro. com $k=25$



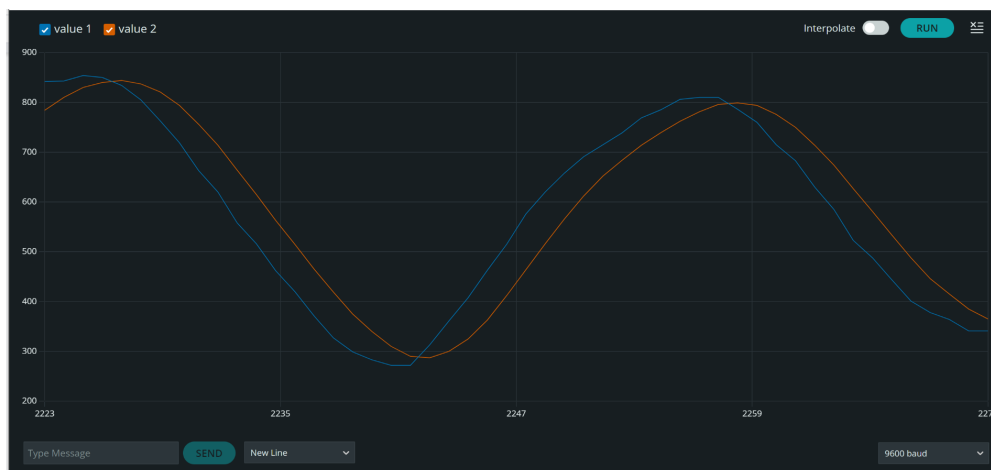
Fonte: Autoria própria

Por fim aumentando-se ainda mais o valor de k para 25 Figura (21) a defasagem de sinal entre o original (azul) e filtrado(laranja) torna-se evidente, esta tendência se mantém conforme se aumenta o número de tamanho de vetor k e dependendo de uma aplicação mais rápida isto poderia acarretar no atraso de resposta do sistema na detecção de aproximação do objeto, resultando em falhas de segurança da encomenda.

Desenvolvido a codificação de filtro média móvel inicialmente para o teste somente com potenciômetro (apêndice A) dentro da IDE do Arduino, quando o valor do vetor é

colocado em 5, o sinal resultante de filtro possui diversas ranhuras acompanhando o ruído de sinal Figura (22).

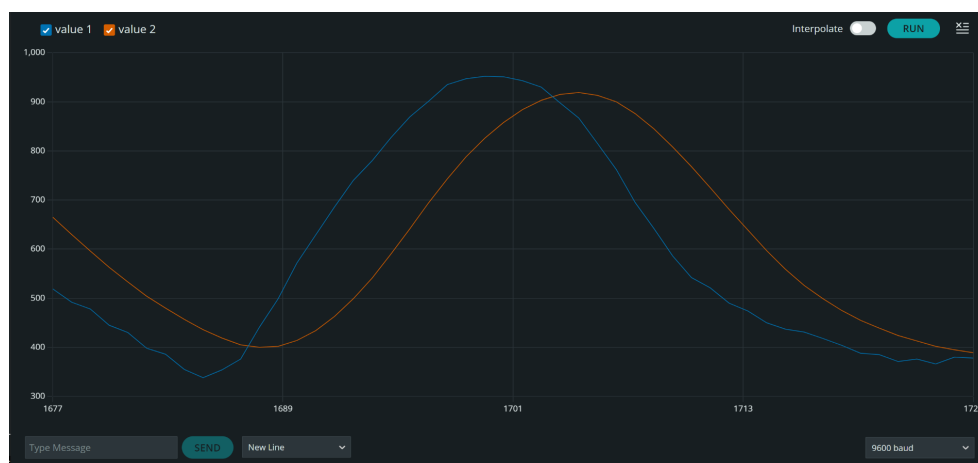
Figura 22 - Sinais Gerados em IDE Arduino, com $k=5$, onde observa-se maior defasagem comparado aos resultados em Simulink.



Fonte: Autoria própria

Nota-se também que desta vez maior defasagem comparado ao Simulink, podendo ser resultante do processamento e sequenciamento de código maiores para sua execução de ciclos.

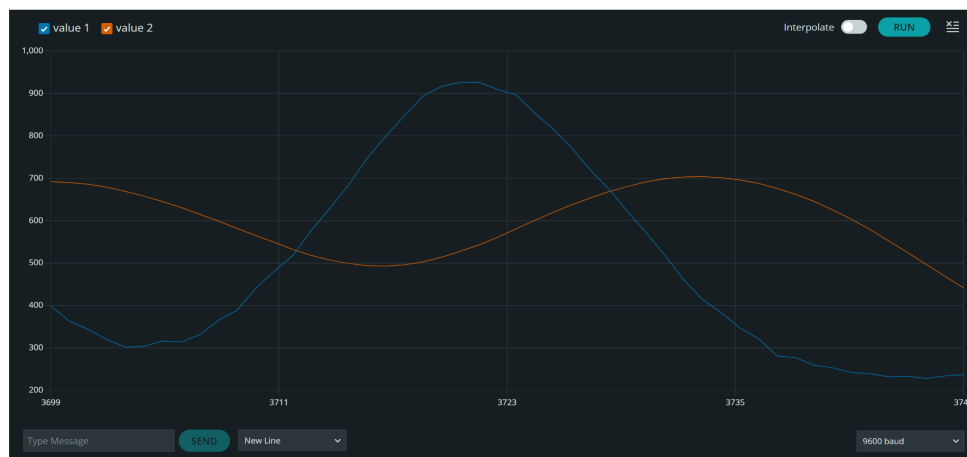
Figura 23 - Sinais Gerados em IDE Arduino, com $k=10$, onde observa-se melhor redução da componente de ruído de sinal.



Fonte: Autoria própria

Dado o valor de $k=10$ Figura(23) percebe-se pelo começo e fim de gráfico em baixo sinal uma boa redução da componente de ruído de sinal junto a uma defasagem perceptível porém dentro de uma faixa de tempo aceitável.

Figura 24 - Sinais Gerados em IDE Arduino, com $k=25$, onde observa-se a defasagem de aproximadamente meio período de onda.



Fonte: Autoria própria

Dentro da situação de $k=25$ Figura (24), os sinais para um valor mesmo valor de vetor em arduino já não corresponde um bom acompanhamento do sinal original, ocorrendo uma grande defasagem no processamento de filtragem neste software (quase meio período).

Apesar de utilizar uma taxa de amostragem diferente na construção de gráficos, o comportamento de sinais perante a alteração em k se manteve, desta vez com maior taxa de deslocamento perante o aumento de k . Optou-se para a utilização de valor $k=10$ na aplicação de circuito protótipo para realizar o '*order picking*' de uma indústria. É natural e já esperado tal comportamento de sistema embasado nos estudos de média móvel.

Na aplicação o tempo de reação do sistema foi satisfatório para a leitura de presença de objeto, sendo possível também alterar a sensibilidade de sensor pelo seu potenciômetro ou mesmo na distância limite definida na codificação verificada no anexo (B).

5 CONCLUSÃO

O sistema desenvolvido se provou funcional e seguiu as diretrizes gerais desejadas. Em uma análise parcial, o sensor óptico com saída analógica detectou a aproximação de objetos reduzindo o valor de sinal conforme a aproximação de acordo com o modo em que se foi projetado. Já o segundo sensor, utilizando a saída digital, foi codificado para reduzir a contagem somente com o pulso de sinal, quando o objeto sai da área de contagem e efetivamente parando todo o sistema após zerar o contador.

O filtro digital de média móvel teve um desempenho melhor em simulink do que o da IDE Arduino em questão a sua defasagem com o aumento de k . Porém, seguiu cumprindo a função de atenuação de ruídos, demonstrando um sinal mais nítido e contínuo ao original.

Espera-se então a possibilidade de uma aplicação deste sistema ou semelhante em armazenamento de materiais, seja uma indústria, estoques de comércio ou correio. Onde ainda há o trabalho manual de busca de mercadorias. Gerando uma economia de tempo, mão de obra e consequentemente capital, sendo os componentes utilizados relativamente baratos.

Sugestões de trabalho futuro incluem o tratamento de sinal dentro do próprio MatLab, desenvolver de uma interface digital, app para solicitar a adição e subtração de componentes, tal interface possivelmente utilizada ao celular e conectada em nuvem para acesso remoto de controle.

Assim, o presente trabalho pode ser aplicado na automação de seleção e contagem de objetos dentro de um armazenamento, seja em uma pequena loja, em um estoque de distribuição de entregas ou mesmo em uma linha de produção de montagem. O sistema irá efetuar o processamento de quantidade e designar os produtos selecionados corretamente.

REFERÊNCIAS

ALMEIDA, Marcos Proença de. **“Eliminação de ruído impulsivo em imagens coloridas usando um filtro mediano seletivo e retoque digital”**/ Marcos Proença de Almeida - São José do Rio Preto: [s.n.], 2010.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 10520: **Informação e documentação - apresentação de citações em documentos**. Rio de Janeiro, 2002. 7 p.

A.V. Oppenheim and R.W. Schafer **“Discrete-Time Signal Processing”** Prentice Hall, 1989/1999

BAPTISTA, J. M.. **Análise e Desenvolvimento de Sensores de Fibra Óptica de Intensidade auto-Referenciados**. Tese de Doutorado. Faculdade de Engenharia da Universidade do Porto, Portugal, 2002.

BORGES, Fernando T. **“Filtro Notch digital para 60 Hz: projeto, simulação e montagem”**. 2018. 35 f. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) - Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2018.

BICA, Marcos R. R., **“Sistema de aquisição de dados e comunicação sem fio para monitoramento de temperatura em silo de armazenamento de grãos”**. 2019. Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências Agrônômicas, Botucatu, 2019.

CARNEIRO, A. C.; BARBERO, A. P. L. **“Principio de funcionamento dos sensores ópticos baseados em grades em fibras ópticas e sua importância nas diferentes áreas tecnológicas”**. Universidade Federal Fluminense (UFF), Departamento de engenharia de telecomunicações, ENGEVISTA, V. 16, n. 4, p. 389-403, dezembro 2014.

CAZER, A. de O.; ALTOÉ, A. V.; JÚNIOR, A. P. **“ Sensores Ópticos de Passagem e medição de distância”**. Universidade de Brasília (UnB), Faculdade de Tecnologia, Departamento de Energia Elétrica, Brasília, junho 2005.

DE LA VEGA, Alexandre Santos “**Tutorial sobre Sistemas de Média Móvel para Fundamentos de Processamento Digital de Sinais**” / Alexandre Santos de la Vega. - Niterói: UFF/TCE/TET, 2018.]

DIAS Jéssica, “**Tecnologias de separação e preparação de pedidos**”, engenharia360, junho 2014. Disponível em
<<https://engenharia360.com/tecnologias-de-separacao-e-preparacao-de-pedidos/>> acesso em 26 de agosto de 2021.

DJP Automação: “**Sensores ópticos**”. 7 de julho de 2020. Disponível em
<<https://djpaotomacao.com/sensores-opticos/>> acesso em 13 de setembro de 2021.

DORF , Richard C. “**Sistemas de controle modernos**” / Richard C. Dorf, Robert H. Bishop ; tradução e revisão técnica Jackson Paul Matsuura. - Rio de Janeiro : LTC, 2009

GOLNARAGHI, M. F. “**Sistemas de controle automático**” / Farid Golnaraghi, Benjamin C. Kuo ; tradução e revisão técnica Fernando Ribeiro da Silva. - 9.ed. Rio de Janeiro: LTC, 2012.

HAYKIN, Simon “**Sinais e sistemas**” / Simon Haykin, Barry Van Veen; tradução José Carlos Barbosa dos Santos. Porto Alegre: Bookman, 2001. 668p. ; 28 cm.

MARTIN, F. G. “**The 6270 robot builder 's guide, MIT, LEGO robot design competition.**” Cambridge, MA: The Media Laboratory, Massachusetts Institute of Technology, 1992.

MORAES, Cícero Couto de “**Engenharia de automação industrial**” / Cícero Couto de Moraes, Plínio de Lauro Castrucci. - 2.ed. - [Reimpr.]. - Rio de Janeiro :LTC, 2010

MORAES, Larissa Lemos de. “**Alocação ótima de itens baseada no número de clientes e demanda dos produtos utilizados order picking**” [manuscrito] / Larissa Lemos de Moraes. - 2019

MURILO, F. “**Introdução aos microcontroladores. Microcontroladores e microprocessadores. Tecnologia em manutenção industrial**”. FATEC Cariri, Centec – Instituto Centro de Ensino Tecnológico, janeiro 2021. Disponível em <<https://www.muriloleal.com.br/visao/repositorio/centec/tmimicro//AULA%2002%20-%20INTRODU%C3%87%C3%83O%20AOS%20MICROCONTROLADORES.pdf>> acesso 30 de agosto de 2021

ROSÁRIO, João Maurício “**Automação industrial**” / João Mauricio Rosário. - São Paulo: Baraúna, 2009

ROSÁRIO, João Maurício “**Princípios de mecatrônica**” / João Maurício Rosário. – São Paulo : Prentice Hall, 2005

SÁ, T. E. “**Automatização das descargas dos mictórios**”. Monografia, (Tecnólogo em Manutenção Industrial), Universidade Tecnológica Federal do Paraná, Pato Branco, 2011, 44p.

SANTOS, Bruno Feu de Brito “**Alimentador automático para animais utilizando Arduino**” / Bruno Feu de Brito Santos – Guaratinguetá : [s.n], 2015.

SILVA, R. F. A.; COSTA, J. A. D. “**Sensor por reflexão difusiva com múltiplas faixas de detecção e com saída binária sequencial**”. VIII Semetro. João Pessoa, PB, Brasil, junho, 2009.

SOUZA, F. “**Arduino uno**”. Disponível em: <<https://embarcados.com.br/arduino-uno/>> . Acesso em: 30 abr. 2022.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. “**Sensores Industriais Fundamentos e Aplicações**”. Editora Érica, 4 ed. São Paulo, Brasil, 2007.

USHIZIMA, Mariana Monteiro “**Resolução do problema de sequenciamento de múltiplas linhas de montagem de modelos mistos em ambiente de produção enxuta**” / Mariana Monteiro Ushizima – Guaratinguetá, 2014.

APÊNDICE A - Código Arduino para teste em potenciômetro

```

//===== potenciometro
#define pot A1                //mapeamento potenciometro
//nome funcao
long mov_avg();
//=====global
int sinal;                    //entrada analogica
int filtro;                   //saída filtro
//=====AUX
#define k 25                   //número de pontos de MM
int num[k];                   //criacao de vetor num
void setup() {
    Serial.begin(9600);        //selecao de Baudrate
    pinMode(pot, INPUT);
} //end setup
void loop() {
    sinal = analogRead(pot);    //leitura de sinal
    filtro = mov_avg();         //chamada de funcao
    Serial.print(sinal);        //imprime sinal original
    Serial.print(" ");
    Serial.println(filtro);     //imprime sinal filtrado
    delay(20);
} //loop
long mov_avg()                 //inicio de funcao
{
    for (int i=k-1; i>0; i--)
        num[i] = num[i-1];     //deslocamento de vetor de MM
    num[0] = sinal;             //posicao inicial do vetor
    long acc = 0;               //acumulador
    for (int i=0; i<k; i++)
        acc += num[i];         //somatoria de numeros dos pontos
    return acc/k;               //retorna MM
}

```

APÊNDICE B - Código Arduino para sistema de ‘Order Picking’

```
//=====Filtro media movel
//=====

#include <neotimer.h>
Neotimer t1; //cria um timer especifico
Neotimer t2; //segundo timer opcional
//=====Mapeamento=====

#define pin0 A0 //define pino analogico de sensor IR
//=====Auxiliar=====

#define x 3 //define o tamanho do vetor de funcao media movel variavel k
//=====funcao med=====

long mov_avg1(); //define a funcao
//=====variaveis=====

int orig1; int filt1; // variaveis de sinal de saida original e filtrada
int sensor1 =3;

bool s1 = 0; //variavel booleana
bool bot = 0; //variavel booleana

int botON=2; int bo; //botao acionamento
int add1 = 11; int m1; //botao soma 1
int sub1 = 10; int n1; //botaosubtrai 1

static bool estado = LOW;
int on = 0;

int ledin1 = 12;
int ledout1 = 8;

int count1 =0; //define variavel e inicio do contador

int numero1[x]; //vetor para media movel 1
```

```
//=====SETUP=====

void setup() {
  Serial.begin(115200);
  pinMode(2,INPUT);    //configura liga desliga
  t1.set(250);         //configura valor de timer de 250 ms
  t2.set(500);         //configura valor de timer de 500 ms
  //LINHA 1
  pinMode(pin0, INPUT); ///configura entrada sensor 1
  pinMode(3, INPUT);   //configura entrada sensor de contador 1

  pinMode(11, INPUT);  //add linha 1
  pinMode(10, INPUT);  //sub linha 1
  pinMode(12, OUTPUT); //LEDin1
  pinMode(8, OUTPUT);  //LEDout1
  //LINHA 2

} //end setup

void loop() {
  //inicio
  orig1 = analogRead(pin0); //definicao da leitura de sinal analogico
  filt1 = mov_avg1();       //definicao para a funcao

  //=====BOTAO ADD SUB=====
  //linha 1
  if (t1.repeat()){ //utiliza neotime para reconhecimento de sinal de botao
    m1=digitalRead(add1); // define leitura do botao +
    if (m1==HIGH){        //verifica estado do botao
      count1+=1;          //adiciona 1 unidade ao contador
    }
    n1=digitalRead(sub1); // define leitura do botao -
    if (n1 == HIGH){      //verifica estado do botao
      count1-=1;          //subtrai 1 unidade ao contador
    }
  }
}
```

```

}
//FIM Botao ADD
//=====LIGA DESLIGA=====
estado = digitalRead(botON); //define leitura da chave
if (estado==LOW){           //verifica estado da chave
  if (count1>0){             //verifica estado do contador se maior de 0 executa
    digitalWrite(ledin1, HIGH); //Liga a esteira de entrada
    if (filt1<400 && !digitalRead(3)){
//verifica estado do sensor IR analogico
      digitalWrite(ledin1, LOW); //Desliga a esteira de entrada
    }
    if (digitalRead(3)) s1=1; //verifica estado do sensor IR em saida digital
    if (!digitalRead(3) && s1){ //verifica estado de saida do material
      s1=0; //muda o estado s1
      count1-=1; //subtrai o contador apos passagem de material
      digitalWrite(ledout1, HIGH); //Liga a esteira de saida
      delay(2500); //Tempo para manter esteira ligada
      digitalWrite(ledout1, LOW); //Desliga esteira de saida
    }

  } //end contador
else{count1=0; //caso contador sej um numero negativo, sera zerado
  digitalWrite(ledout1, LOW); //Certifica desligamento de esteiras
  digitalWrite(ledin1, LOW);
} //end else
//=====IMPRIMIR SINAL=====

Serial.print(orig1); //imprime sinal original
Serial.print(" "); //imprime espaco
Serial.print(filt1); //imprime sinal filtrado
Serial.println();
delay(20);
} //end ligado

```

```

else{
    digitalWrite(ledout1, LOW);
    digitalWrite(ledin1, LOW);
    if (t2.repeat()){          //utiliza neotimer para delay de impressao
        Serial.print("desligado "); //imprime o texto
        //=====Imprimir dados=====
        Serial.print("Conta: ");
        Serial.println(count1);    //imprime valor do contador
    }
} //end else deligado

} //=====end LOOP

//=====FUNCAO=====
//linha producao 1
long mov_avg1(){
    for (int i= x-1; i>0; i--) numero1[i] = numero1[i-1];
    //deslocamento do vetor pelo numero de pontos
    numero1[0] = orig1; //posicao 0
    long acc1 = 0; //acumulador
    for (int i=0; i<x; i++) acc1+=numero1[i]; //somatoria do n de pontos
    return acc1/x; //retorna a media movel
} //end funcao linha 1

```