

Robô do tipo AMR seguidor de operador

Lukas Araújo da Silva
Engenharia Eletrônica

Faculdade do Gama - UnB

Área Especial, Projeção A, UnB - Setor Leste - Gama

Email: lukas.araujo@aluno.unb.br

Matrícula: 190033771

Victor Orfeu Merlo
Engenharia Eletrônica

Faculdade do Gama - UnB

Área Especial, Projeção A, UnB - Setor Leste - Gama

Email: victor.orfeu@aluno.unb.br

Matrícula: 190020695

Resumo—Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, feli. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

I. INTRODUÇÃO

Com a pandemia de COVID-19, o comércio eletrônico cresceu rapidamente e as companhias envolvidas precisaram acompanhar esse processo. Enquanto não se pode automatizar vans e caminhões de entrega nas ruas, empresas de logística fizeram parcerias com empresas de robótica para desenvolver robôs pequenos a fim de automatizar processos dentro de armazéns [1].

Esses robôs são divididos em diversas categorias de acordo com suas funcionalidades. Robôs do tipo *Automated Storage and Retrieval Systems* (AS/RS) são geralmente braços ou guindastes responsáveis por organizar itens em estantes. *Automated Guided Vehicles* (AGVs) são veículos capazes de se transportar itens sozinhos em caminhos pré-definidos, como empilhadeiras que se dirigem sozinhas seguindo trilhas no chão. *Autonomous Mobile Robots* (AMRs) são robôs que também se movimentam sozinhos, porém através de mapas e sensores que identificam o ambiente ao seu redor e planejam sua rota de acordo [2].

Entretanto, a maioria das empresas que desenvolvem e fabricam essas máquinas são voltadas para automação industrial, geralmente em grande escala, como a *Locus Robotics* [3] e a *SEER* [4]. Um exemplo em menor escala é o da empresa *SMP Robotics*, com um robô pequeno em forma de quadriciclo que pode ter um caminho programado pelo usuário [5].

Este projeto visa desenvolver o protótipo de um robô simples do tipo AMR, capaz de carregar pequenas cargas, identificar e seguir um operador por conta própria. Esse robô poderia ser usado em pequenos armazéns, em mercados ou até em casa. Como o intuito do projeto é testar as funções de eletrônica e software, não haverá foco na estrutura. Não serão definidos requisitos como tamanho, material, peso mínimo

de carga, entre outros, a menos que influenciem nos outros sistemas.

A. Requisitos funcionais

- Movimentar-se no chão.
- Movimentar-se sozinho.
- Movimentar-se sem derrubar a carga.
- Ser capaz de identificar o operador.
- Ser capaz de seguir o operador.

B. Requisitos não-funcionais

- Ter 4 rodas para movimentação.
- Movimentar-se de forma que evite derrubar a carga, com limites de velocidade.
- Usar uma câmera para identificar o operador.
- Usar um Raspberry Pi capaz de processar imagens da câmera e controlar os motores das rodas.
- Mantenha-se a uma distância fixa do operador.
- Na ausência de um operador em seu campo de visão, tenha uma rotina para procurar um. Caso não encontrar um operador próximo, gerar algum sinal (sonoro ou luminoso).

II. DESENVOLVIMENTO

Este projeto está sendo desenvolvido tanto nos computadores pessoais dos integrantes como no Raspberry Pi que será usado no robô. Os conteúdos pertinentes, como códigos desenvolvidos e referências, estão agregados em um repositório hospedado no GitHub, que pode ser acessado através do link: <https://github.com/Awripedes/Projeto-SOE>.

O robô foi inicialmente subdividido em dois sistemas: o sistema de movimentação e o sistema de visão. Cada um desses sistemas possui hardware e software próprios que deverão integrar-se e comunicar-se entre si para compor o robô. Essa divisão foi feita para facilitar definição de requisitos e testes iniciais.

Tabela I. MATERIAIS UTILIZADOS

Componente	Quant.	Especificações
Raspberry Pi 3 Model B	1	Quad Core 1.2GHz, 1GB de RAM, 40 pinos GPIO, 4x portas USB, 1x HDMI, 1x micro SD, conexão bluetooth ou internet por Ethernet ou WiFi.
Pi Camera v1.3	1	5MP, Fotos em resolução 2592x1944, Vídeos 1080p a 30 fps ou 720p a 60 fps.
Módulo L298N	1	Opera de 5V a 12V. Gera 2 sinais de controle separadamente.
Motores DC	4	3V a 6V, com caixa de redução.
Rodas plásticas	4	66mm de diâmetro.
Fonte DC	1	5V, 3A, com interruptor e conector micro-USB.
Baterias 18650	2	5800mAh, 4.2V, Recarregável.

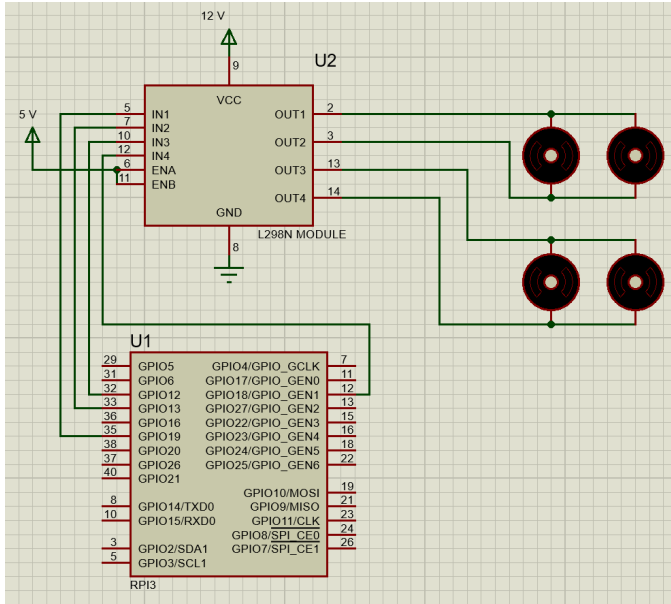


Figura 1. Esquemático do sistema de movimentação, que inclui o Raspberry Pi, o módulo L298N e os motores DC, feito no software Proteus 8 Professional.

A. Hardware

A lista de materiais utilizados e suas especificações pertinentes encontra-se na Tabela I. Os modelos específicos dos componentes foram escolhidos por estarem à disposição dos integrantes ou por serem os mais baratos de se adquirir que atendessem aos requisitos.

1) *Sistema de Movimentação*: Para movimentar o robô, é usado um módulo L298N pois ele consegue gerar 2 sinais de controle de motores DC separadamente e pode ser controlado pelos sinais de 3.3V do Raspberry Pi. Além disso, ele já possui circuitos de proteção e regulação de tensão integrados. No L298N estão conectados 4 pequenos motores DC. Um esquemático das conexões desses componentes é mostrado na Figura 1.

No esquemático, não são mostradas as fontes de alimentação dos componentes. O Raspberry Pi é alimentado por uma fonte DC de 5V, enquanto o L298N é alimentado por duas baterias 18650 de 4.2V em série. O pino de alimentação desse módulo indica que deve ser alimentado por 12V, porém ele consegue operar com tensões menores, como a de 8.4V das baterias. O 5V onde estão conectados os pinos ENA e ENB é fornecido pelo próprio módulo, por um regulador de tensão

7805.

Para controlar os motores, o L298N precisa receber sinais de PWM nas entradas IN1-4 para ativar as saídas OUT1-4. A velocidade dos motores é proporcional ao duty-cycle das entradas. A direção dos motores depende de onde é recebido o PWM. Por exemplo, se o PWM estiver no pino IN1 enquanto IN2 é mantido em nível baixo, os motores conectados em OUT1-2 girarão em uma direção. Se o PWM estiver em IN2 enquanto IN1 é mantido em nível baixo, os motores em OUT1-2 girarão na direção oposta.

2) *Sistema de Visão*: Para providenciar uma "visão" afim de identificar o operador, é utilizada a Pi Camera v1.3, conectada por um cabo flat diretamente ao Raspberry Pi.

B. Software e testes

Com o intuito de testar o funcionamento individual de cada sistema, foram desenvolvidos e utilizados pequenos programas, nas linguagens C, C++ e Python. A estrutura pretendida para o software final consiste em uma parte que identifica o operador e outra que controla os motores para que o robô o siga. A parte que identifica deve produzir dois dados: uma distância com relação ao centro da visão do robô e uma distância do robô ao operador. A parte que controla deve processar esses dados e transformar em movimentos adequados do robô.

A distância com relação ao centro deve indicar o quanto o robô deve girar em torno do próprio eixo para ficar alinhado com o operador. Isso pode ser feito comparando a coordenada do operador no frame de visão do robô com o centro desse mesmo frame. Quanto mais distante do centro, maior deve ser esse valor, indicando que robô deve girar mais para se alinhar.

A distância até o operador deve indicar o quanto o robô deve aproximar-se ou afastar-se dele. Isso pode ser obtido através do tamanho do operador em relação ao frame. Quanto menor for esse tamanho, mais longe o robô está do operador, então ele deve mover-se para frente, e vice-versa.

1) *Teste de Movimentação*: Para verificar as funcionalidades das portas GPIO do Raspberry Pi, foi desenvolvido um programa em C que, com auxílio da biblioteca *pigpio*, envia sinais de PWM para as entradas IN1-4 do L298N. Os sinais no programa são valores inteiros na faixa 0-255. O programa fica em loop numa rotina que varia a velocidade dos motores, do mínimo ao máximo e de volta ao mínimo.

Com esse programa, foi possível identificar as velocidades mínima e máxima dos motores, e os valores de duty-cycle em que eles conseguem se movimentar. A partir do repouso,

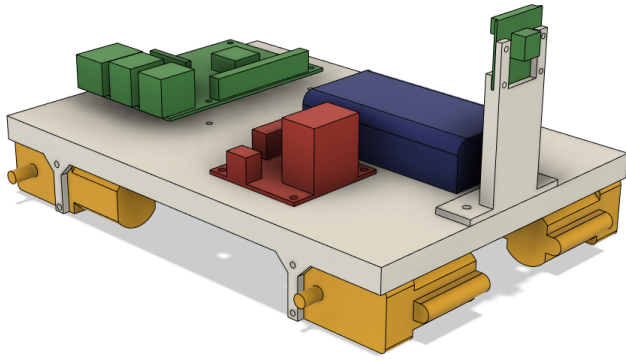


Figura 2. Estrutura desenhada no software Autodesk Fusion 360. Em verde, o Raspberry Pi e a Pi Camera. Em vermelho o L298N. Em azul as baterias 18650. Em amarelo os motores DC.

é necessário enviar o valor 100 para que eles comecem a girar. Estando em movimento, pode-se enviar até 35 que eles se mantêm girando, valores mais baixos fazem com que os motores parem.

2) *Identificando Círculo:* Para testar a Pi Camera v1.3 e habituar o uso da biblioteca OpenCV para visão computacional, foi utilizado um programa em Python que identifica um círculo em tempo real [6]. O programa utiliza de 4 filtros diferentes para aumentar a precisão. Uma imagem é gerada para cada filtro em que são sobrepostas para uma única imagem. Após este processo, uma função é passada no resultado para identificar o círculo.

Após a realização de testes, apresentando um círculo luminoso branco, o resultado foi considerado insatisfatório. A variação da iluminação no ambiente causou ruído suficiente para tornar o programa muito impreciso, identificando círculos onde não havia.

3) *Identificando QRCode:* Outra solução encontrada para identificar o operador foi utilizar um QRCode, pois é uma forma simples de ser identificada e distinta de elementos comuns ao seu redor, como círculos e retângulos. Considera-se que o operador estaria utilizando uma peça de roupa com um QRCode estampado de alguma forma.

A biblioteca ZBar, disponível tanto em Python quanto em C++, possui funções de decodificação de inúmeros tipos de códigos de barras, além de suas estruturas interfacearem bem com as da OpenCV. O programa desenvolvido analisa cada frame e, ao encontrar um QRCode, exibe no terminal seu tipo, seu dado (nesse caso, um link), sua posição no frame e suas dimensões altura e largura.

C. Estrutura

Por não ser o foco do projeto, foi pensada uma estrutura simples que mantivesse os componentes fixos e permitisse fácil troca dos mesmos. Foi desenhada a estrutura, mostrada na Figura 2, para ser produzida na impressora 3D do integrante Lukas. Os componentes foram modelados considerando apenas detalhes pertinentes às dimensões da estrutura.

III. RESULTADOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

IV. CONCLUSÃO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

REFERÊNCIAS

- [1] J. Hirsch, "The rise of the warehouse robot," 2020, disponível em: <https://www.ttnews.com/articles/rise-warehouse-robot>.
- [2] W. Allen, "Guide to warehouse robots: types of warehouse robots, uses, navigation & more," 2021, disponível em: <https://6river.com/guide-to-warehouse-robots/>.
- [3] L. Robotics, "Automated warehouse robots," disponível em: <https://locusrobotics.com/>.
- [4] S. S. I. T. Corporation, "Laser slam small stacker smart forklift: Sfl-cdd14," 2022, disponível em: <https://www.seer-group.com/agv-forklifts/SFL-CDD14>.
- [5] S. Robotics, "Transport robots," disponível em: https://smprobotics.com/application_autonomus_mobile_robots/transport-robots/.
- [6] amey_s, "Raspberry Pi Ball Tracking," 2015, disponível em: <https://www.instructables.com/Raspberry-Pi-Ball-tracking/>.