

15º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2024

Utilização do Roomba para desenvolvimento de plataforma para ensino de robótica com o ROS - Robot Operating System

Rafael Arueira, Luiz Claudio Marangoni.

Graduando em Engenharia de Controle e Automação, Bolsista PIBIT-CNPq, IFSP, Campus Hortolândia, rafael.arueira@aluno.ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 3.04.05.03-3 – Engenharia de Controle e Automação

RESUMO: O Roomba, da iRobot, é um robô autônomo robusto para ambientes internos, com sensores e uma interface de controle integrável a sistemas microcontrolados. Este tipo de robô impulsiona o desenvolvimento de plataformas programáveis, sendo o ROS (Sistema Operacional para Robótica) um dos *frameworks* mais utilizados por ser aberto e fornecer ferramentas amplas, como pacotes e simuladores. Este trabalho tem como objetivo adaptar o Roomba 650 para ser uma plataforma baseada em ROS, focando no ensino de robótica, especialmente em navegação autônoma e localização (SLAM - *Simultaneous Localization and Mapping*).

PALAVRAS-CHAVE: ROS; Create_robot; Raspberry Pi; Roomba; Ensino; IRobot.

Development of a Roomba based framework for teaching robotics with ROS - Robot Operating System

ABSTRACT: The Roomba, by iRobot, is a robust autonomous robot for indoor use, featuring sensors and a control interface integrable with microcontroller systems. Such robots drive the development of programmable platforms, with ROS (Robot Operating System) being widely used for its open nature and extensive tools, including packages and simulators. This work aims to adapt the Roomba 650 as a ROS-based platform for teaching robotics, focusing on autonomous navigation and localization (SLAM - *Simultaneous Localization and Mapping*).

KEYWORDS: ROS; Create_robot; Raspberry Pi; Roomba; Teaching; IRobot.

1. INTRODUÇÃO: A robótica educacional é fundamental no ensino de STEM (ciência, tecnologia, engenharia e matemática). O *Robot Operating System* (ROS) permite aos alunos explorar a robótica de maneira prática, promovendo colaboração e divisão de tarefas. Estudos indicam que o ROS é eficaz para ensino de robótica, destacando a importância de experimentos simples e trabalho em grupos pequenos.

ROS: Desenvolvido pelo laboratório de Inteligência Artificial de Stanford e pelo Willow Garage, o ROS foi lançado como código aberto em 2009. Ele fornece uma ampla gama de bibliotecas e ferramentas para aplicações robóticas complexas, com uma comunidade ativa que apoia sua evolução (TheConstruct, 2024).

Adoção do ROS: O ROS é adotado por sua flexibilidade e modularidade, com uma arquitetura baseada em nós que facilita a integração de *hardware e software*. Ele é amplamente utilizado para robôs industriais, drones e veículos autônomos. Apesar das melhorias no ROS 2.0, o ROS 1.0 foi escolhido para este projeto devido à sua estabilidade e suporte consolidado (Erös, 2021).

Estrutura: O ROS organiza o sistema em pacotes, a unidade básica de funcionalidade, e nós (*nodes*) que realizam tarefas específicas. A comunicação entre nós é feita por tópicos e serviços, com mensagens padronizadas publicadas por um nó (*publisher*) e lidas por outros (*subscriber*) (Marcato, 2022; Fabro, 2022; ROS, 2024).

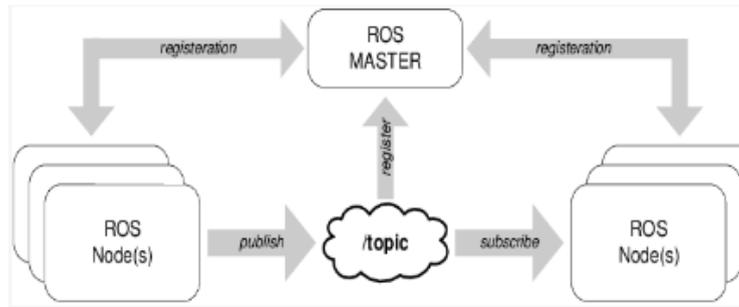


Figura 1 - Estrutura típica do framework.

2. MATERIAL E MÉTODOS:

2.1 Roomba 650: Fabricado pela iRobot, será o robô utilizado neste projeto. Equipado com sensores de *cliff*, obstáculos, paredes e navegação, e atuadores como motores de rodas e sistema de escovas e aspiração, o Roomba 650 é conhecido por sua confiabilidade e simplicidade, tornando-o uma escolha prática para experimentos em robótica.

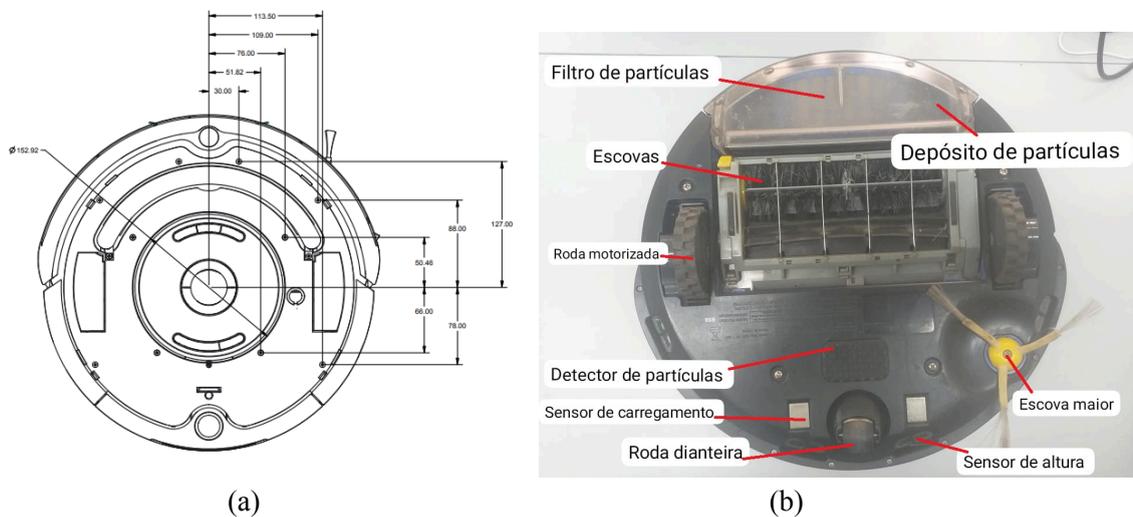


Figura 2 - Visão (a) superior e (b) inferior do roomba i650

2.2 Interface de comunicação: A interface de comunicação serial (SCI) foi disponibilizada pela iRobot em 2006, permitindo que qualquer Roomba seja controlado via dispositivos de programação serial, como Arduino e Raspberry Pi. A Roomba *Open Interface* (OI) possibilita a comunicação com o robô para enviar comandos e ler dados dos sensores através de um conector serial de 7 pinos.

Modos de Operação do Roomba

O Roomba pode operar em diferentes modos, cada um com funcionalidades específicas:

- Off: O robô está desligado.
- Passivo: O robô está ligado, mas não executa nenhuma tarefa de limpeza ou movimentação.
- Safe: O robô opera em um modo seguro, permitindo comandos básicos e evitando operações potencialmente perigosas.
- Full: O robô opera em modo completo, com todas as funcionalidades e sensores ativados. Para mais detalhes sobre cada modo, consulte a página 7 do manual.

- **Adaptador FTDI USB para TTL RS 232 de 4 pinos:**

O adaptador utilizado é do tipo FTDI USB A para TTL RS 232 de 4 pinos, que permite a comunicação serial entre o Roomba e o computador.



Figura 3 - Cabo FTDI USB A para TTL RS232 PL2303HX 4 Pinos.

Montagem do Cabo Serial: O cabo serial teve sua confecção de acordo com o esquema de pinagem detalhado no documento "Create cable pinout" disponível na documentação oficial do *Create 2*. O esquema detalha como conectar os pinos do adaptador FTDI aos pinos correspondentes no Roomba.

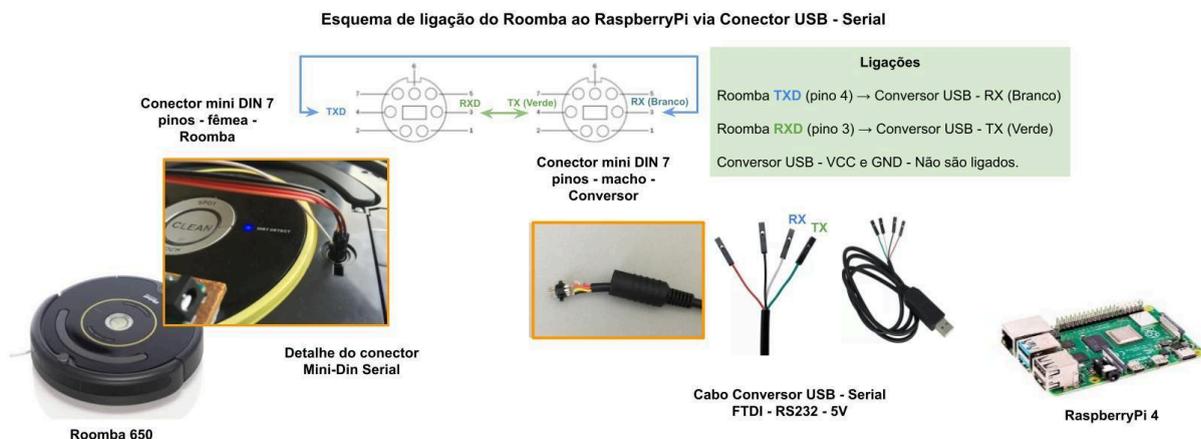


Figura 4 - Esquema de ligação que utiliza o cabo confeccionado.

2.3 Pacote: O pacote `create_robot` no ROS fornece um driver para os robôs iRobot Create 1 e Create 2. Este pacote encapsula os comandos da biblioteca C++ `libcreate` e os disponibiliza no ROS na forma de pacotes, tópicos, nós e mensagens.

Subpacotes:

- **create_bringup:** Arquivos de configuração para plataformas Create/Roomba.
- **create_description:** Descrições URDF e malhas 3D.
- **create_driver:** Driver principal para comunicação serial.
- **create_msgs:** Mensagens para a interface com o driver.

Tópicos Publicados:

- **Battery/capacity:** Informações sobre a bateria.
- **Joint_states:** Estado das juntas e escovas.
- **Odom:** Informações de odometria.

Tópicos Assinados:

- **check_led:** Comando para LEDs.
- **cmd_vel:** Controle de velocidade e movimentação.

Este conjunto de pacotes e tópicos oferece uma interface robusta e flexível para a implementação de aplicações robóticas com o Roomba em um ambiente ROS.

RESULTADOS E DISCUSSÃO: Proposta Inicial de *Hardware*: A proposta inicial para a adaptação do Roomba envolvia a integração do Roomba com um Raspberry Pi e um Arduino, conforme mostrado no diagrama abaixo. O Arduino seria utilizado para comunicação com o Roomba via o pacote rosserial para ROS.



Figura 5 - Diagrama da ideia inicial de comunicação.

Mudança na Proposta de *Hardware*: A integração do pacote create_robot eliminou a necessidade do Arduino. Este pacote fornece uma interface direta para o Roomba, facilitando a comunicação e controle via ROS sem intermediários.

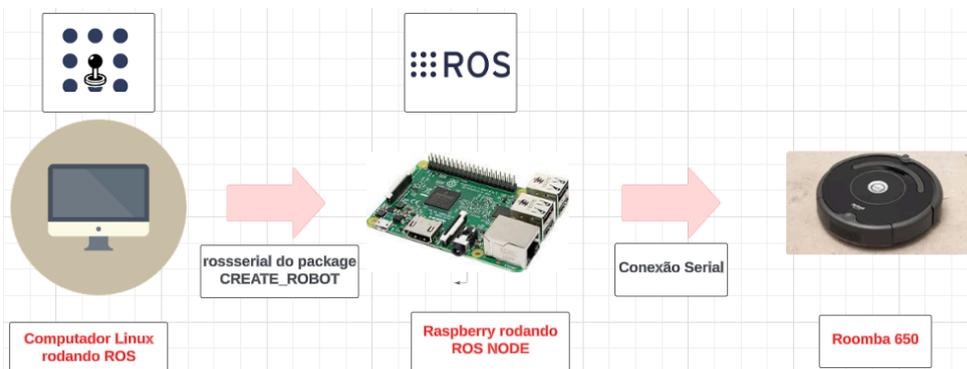


Figura 6 - Diagrama representando a atual forma de comunicação.

Adaptações Físicas: Modificações foram feitas no Roomba para acomodar o Raspberry Pi e um power bank. Placas de MDF foram cortadas e montadas para suportar os novos equipamentos.

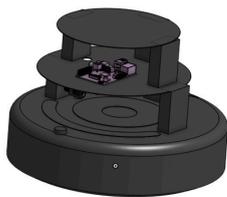


Figura 7 - Modelagem do Roomba com adaptações, feita no software OnShape.

A base projetada em MDF foi fixada no lugar da tampa original, conforme o projeto no software Inkscape, garantindo a estrutura necessária e acesso à conexão serial.

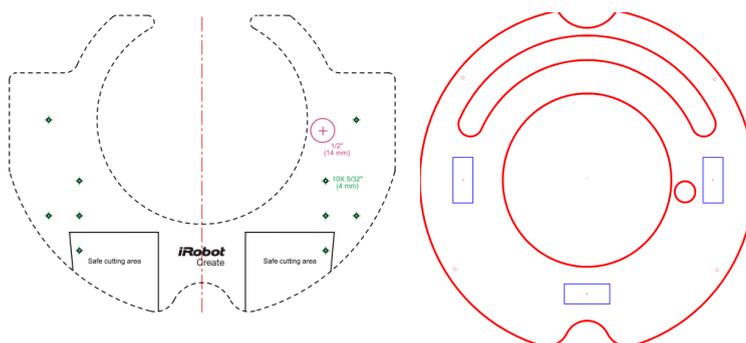


Figura 8 - Tampa do manual do fabricante Roomba ao lado da projetada no Inkscape

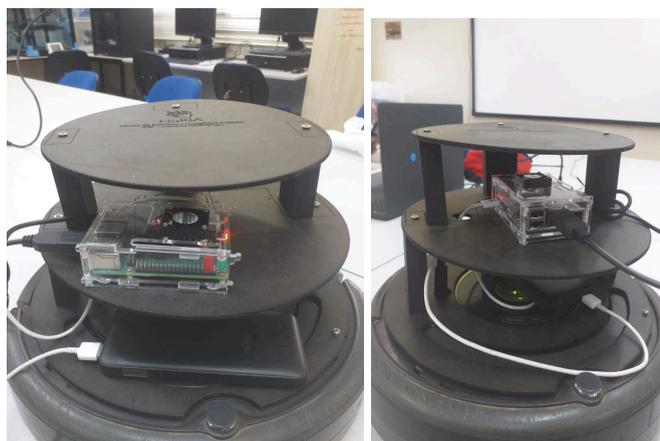


Figura 9 - Montagem do Roomba adaptado e conectado ao Raspberry Pi 4.

Testes e Resultados: Com o Raspberry Pi 4 conectado ao Roomba via cabo serial, e o ROS instalado, foram realizados testes utilizando o pacote `create_robot`. A comunicação via Wi-Fi entre o Roomba e o computador rodando Linux foi bem-sucedida, e os principais resultados foram:

Comunicação Estabelecida: A comunicação entre o Roomba e o Raspberry Pi funcionou corretamente, permitindo controle e leitura de dados.

Consulta a Tópicos: Os tópicos disponíveis no pacote `create_robot`, como o estado da bateria e informações dos sensores, foram consultados com sucesso, confirmando o funcionamento adequado do pacote.

Esses resultados indicam que a integração do Roomba com o ROS usando o pacote `create_robot` está funcionando conforme o esperado, com uma comunicação eficiente e correta.

CONCLUSÕES: O trabalho comprovou a viabilidade do uso do ROS para o controle do Roomba 650, superando desafios de integração de *hardware* e comunicação. A adaptação do Roomba para o ROS, usando o pacote `create_robot`, foi bem-sucedida, e um modelo 3D desenvolvido no Onshape permite simulação virtual no Gazebo e RViz, facilitando testes de controle e navegação.

Como próximos passos, planejamos implementar controle remoto via terminal e *joystick* e realizar estudos para otimizar o comando dos tópicos, buscando um controle mais eficiente e intuitivo.

CONTRIBUIÇÕES DOS AUTORES

- Rafael Arueira de Souza: Responsável pela concepção da ideia, desenvolvimento do experimento com o Roomba, implementação do sistema ROS e redação inicial do artigo.
- Luiz Claudio Marangoni: Orientou o trabalho em todas as fases, incluindo a estruturação da pesquisa, revisão crítica do conteúdo e supervisão técnica dos experimentos.

AGRADECIMENTOS: Agradeço ao CNPq pelo financiamento através da bolsa PIBIT, à Direção do Campus Hortolândia e à Coordenação do Ubuntu Maker pelo suporte no uso dos recursos do laboratório, e à equipe do lab Maker e CTI pelo apoio técnico ao projeto.

REFERÊNCIAS

ARAÚJO, A. et al. Integrating Arduino-Based Educational Mobile Robots in ROS. *Journal of Intelligent and Robotic Systems: Theory and Applications*, v. 77, n. 2, p. 281–298, fev. 2015.

CRAIG, John J. *Robótica*. 3. ed. São Paulo: Pearson, 2013.

ERÖS, Ákos. *ROS for Beginners - A Hands-On Introduction to the Robot Operating System*. 2021. Disponível em: <https://www.ros.org/>. Acesso em: 5 set. 2024.

FABRO, André. *Fundamentos do Robot Operating System (ROS)*. São Paulo: Érica, 2022.

IROBOT. Create® 2 Serial to USB Cable Creation. Disponível em: <https://edu.irobot.com/learning-library/getting-started-with-create-2>. Acesso em: 5 set. 2024.

INSTRUCTABLES. Roomblock: a Platform for Learning ROS Navigation with a Modified iRobot Create 2. Disponível em: <https://www.instructables.com/Roomblock-a-Platform-for-Learning-ROS-Navigation-W/>. Acesso em: 5 set. 2024.

MJROBOT. Controlando o robô aspirador Roomba com Arduino e Android. 2016. Disponível em: <https://mjrobot.org/2016/07/04/controlando-o-robô-aspirador-roomba-com-arduino-e-android/>. Acesso em: 5 set. 2024.

RASPBERRY PI. Raspberry Pi 4 Model B. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. Acesso em: 5 set. 2024.

REIS, Fábio. *Revolução 4.0. A Educação na era dos Robôs*. São Paulo: Cultura, 2019.

ROLDÁN-ÁLVAREZ, Juan; MAHNA, Rafael; CAÑAS, Francisco. *Manual de Utilización del Roomba con ROS*. 2022.

