PLATAFORMA ROBÓTICA EXPERIMENTAL PARA AUXÍLIO NO MANEJO AGRÍCOLA

Bernardo Andrade[1], Luís Fernando Caparroz Duarte[2], Francisco Sergio Dias Martins Junior[3]

RESUMO: Este trabalho apresenta o desenvolvimento de uma plataforma robótica para navegação autônoma utilizando o microcontrolador ESP32-S3, integrando o *middleware* micro-ROS e o sistema operacional FreeRTOS no *framework* ESP-IDF. Essa configuração permite a comunicação via Wi-Fi com um subsistema de controle ROS 2, responsável pela navegação e pela tomada de decisões da plataforma. A estrutura da plataforma foi montada em uma base robótica comercial com esteiras, equipada com o sensor RPLIDAR A1M8, para mapeamento e detecção de obstáculos, e o sensor inercial BMI-323, para medições de inclinação, aceleração e rotação. Os motores de corrente contínua, que movem as esteiras, contam com *encoders* de efeito Hall, assegurando controle preciso da tração. O gerenciamento de energia é feito pelo circuito integrado MAX17320, que monitora uma bateria de quatro células de Li-Ion 18650, registrando parâmetros críticos como tensão, corrente e temperatura, garantindo segurança e eficiência energética. O *hardware* foi organizado em uma PCB com módulos independentes para o sistema de potência e a ponte H. Testes laboratoriais demonstraram o funcionamento adequado do sistema, com a execução correta de processos de monitoramento e tomada de decisão automáticos.

Palavras-chave: ESP32; micro-ROS; ROS.

1 INTRODUÇÃO

A automação robótica tem se destacado como uma solução para enfrentar os desafios decorrentes do aumento demográfico, aprimorando atividades em diversos setores (HACKENHAAR, HACKENHAAR e ABREU, 2014). Esses sistemas são projetados para reduzir o esforço físico repetitivo e a exposição a condições insalubres, que podem prejudicar a saúde dos trabalhadores. Além disso, eles são úteis em tarefas que exigem alta precisão e baixo risco de erro humano, evitando consequências graves para a segurança e perdas de recursos. Na agricultura, a robótica é particularmente relevante, dado que os trabalhadores enfrentam condições adversas, como exposição ao sol, transporte de cargas pesadas e manuseio de produtos químicos tóxicos.

O ROS 2 (Robot Operating System 2) é uma coleção de *frameworks* de *software* de código aberto que oferece suporte massivo para desenvolvimento de robôs. Embora

[1] Aluno do curso de Engenharia Eletrônica; andrade.bernardo2010@gmail.com.
[2] Professor Doutor da UTFPR campus Cornélio Procópio; lfduarte@utfpr.edu.br.
[3] Aluno do curso de Engenharia Eletrônica; franciscosdm.jr@gmail.com.

seu uso demande um PC com o sistema operacional Linux, ele pode se comunicar com sistemas embarcados através do micro-ROS, um *framework* voltado para dispositivos de recursos limitados que permite a interoperabilidade entre microcontroladores e o ecossistema ROS 2, sendo eficaz em ambientes com restrições de tempo real e recursos computacionais (BELSARE et al., 2023).

O objetivo deste trabalho é desenvolver uma plataforma robótica baseada no microcontrolador ESP32-S3 com micro-ROS, capaz de se comunicar com um subsistema ROS 2 via Wi-Fi, permitindo sua navegação autônoma.

2 METODOLOGIA

Este projeto foi desenvolvido para se comunicar com um servidor Linux com Ubuntu 22.04 LTS e ROS 2 Humble Hawksbill. O subsistema ROS 2, descrito em (MARTINS JUNIOR, DUARTE e ANDRADE, 2024), é central para a coordenação das ações da plataforma, permitindo que os dados dos sensores sejam processados e as decisões sejam tomadas em tempo real.

A plataforma robótica utiliza o microcontrolador ESP32-S3, integrando o *framework* do micro-ROS na versão Humble como um componente do ESP-IDF na versão v5.2.2. Esse microcontrolador, programado com base no FreeRTOS, gerencia as tarefas em tempo real e possibilita a comunicação eficiente via Wi-Fi com o subsistema de controle ROS 2, além de coletar dados do sensores embarcados.

A estrutura física da plataforma é composta por uma base robótica metálica comercial do tipo esteira, equipada com dois motores de corrente contínua com caixa de redução 18.8:1 e *encoders* baseados no sensor de efeito Hall MLX90290, que monitoram a rotação dos motores e garantem o controle preciso da tração das esteiras. Um sensor inercial BMI-323 é responsável pela medição da aceleração e rotação da plataforma. Este sensor se comunica por um barramento I2C e é configurado para operar a uma frequência de 31,25 Hz com uma faixa de aceleração de 4g e rotação angular de 500°/s.

Um circuito integrado *fuel gauge* MAX17320 é utilizado para gerenciar a bateria de quatro células de Li-Ion 18650, monitorando parâmetros críticos como tensão, corrente e temperatura. Este componente possui mecanismos de segurança, como proteção contra sobrecarga, curto-circuito e superaquecimento. Ele se comunica com o ESP32-S3 no mesmo barramento I2C do sensor inercial, otimizando o uso dos recursos.

Para a detecção de distâncias, foi utilizado o RPLIDAR A1M8, que opera através de um feixe de laser e realiza medições de 0 a 360° ao redor do dispositivo. Suas leituras variam entre 0,13 m e 12 m, com uma frequência de cerca de 7 Hz. O LIDAR foi conectado ao ESP32-S3 via UART com uma taxa de transmissão de 115200 bps, e o microcontrolador é responsável por interpretar os dados enviados pelo sensor.

Os motores da plataforma robótica são controlados por uma ponte H DBH-12V, que gera a modulação por largura de pulso (PWM) para o controle da velocidade e direção dos motores. Essa ponte H é alimentada por um conversor CC QS-2405CCBD-12A, configurado para uma saída de 10,5 V. Além disso, um conversor *buck* baseado no LM2596 alimenta diretamente o LIDAR, fornecendo os 5 V necessários para seu funcionamento, e outro conversor *buck*, baseado no TPS54360, foi projetado para fornecer 3,3 V ao barramento do microcontrolador. A Figura 1 apresenta o diagrama de blocos da implementação do *hardware* do sistema.

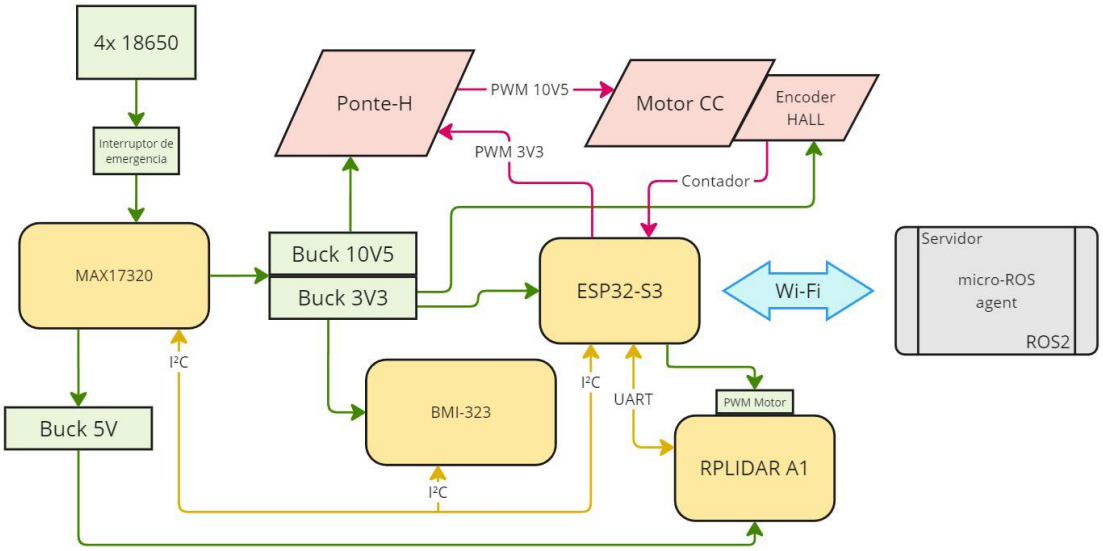
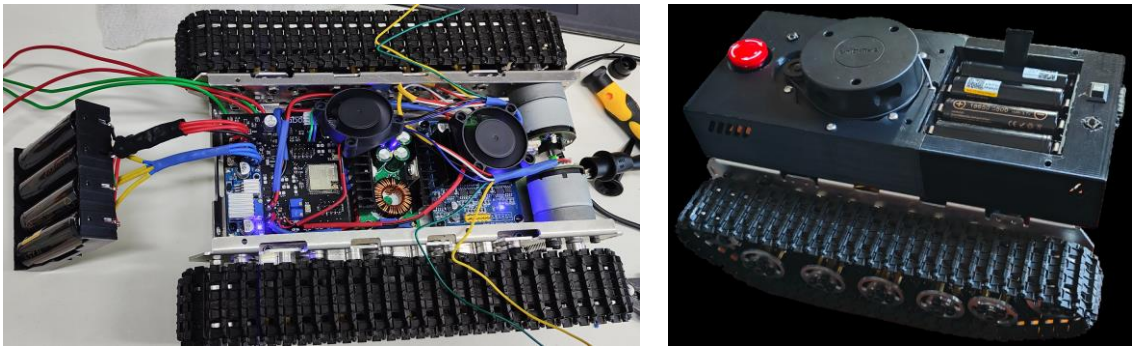


Figura 1: Diagrama de blocos do *hardware*. Fonte: próprio autor.

A montagem do *hardware* foi organizada em uma placa de circuito impresso (PCB), projetada no *software* KiCad e fabricada pela JLCPCB. A PCB foi desenhada para acomodar todos os sensores e módulos, exceto o LIDAR e a bateria. Os componentes foram montados manualmente, e a organização dos módulos foi feita para minimizar a interferência elétrica e a impedância das trilhas. Para fixar os componentes e otimizar o espaço dentro da plataforma robótica, foi utilizado um suporte impresso em 3D, projetado no *software* Fusion 360. As Figuras 2a e 2b apresentam a plataforma robótica montada.



(a) Visão superior interna (b) Visão superior externa

Figura 2: Plataforma robótica. Fonte: próprio autor.

O *software* desenvolvido para a plataforma robótica é dividido entre o sistema em tempo real rodando no ESP32-S3 e o servidor ROS 2. No microcontrolador, o micro-ROS Client gerencia a comunicação entre o ESP32-S3 e o servidor ROS 2, utilizando a API rclc (ROS *Client Library in C language*) do ROS 2, que permite a subscrição e publicação de dados em tópicos específicos. O micro-ROS Agent, rodando no servidor, atua como intermediário entre o ROS 2 e os nós embarcados, garantindo a comunicação eficiente e sincronizada. O nó "/uwaba_node" gerencia três tópicos principais:

- "/micro_encoders": Publica as leituras dos *encoders* acoplados aos motores, com dados estruturados em um *array* de *double* contendo as rotações dos motores, utilizando a interface "uwaba_prototype_interfaces__msg__EncoderMgs".
- "/micro_imu": Contém dados do sensor inercial, como aceleração linear e velocidade angular, utilizando a estrutura de mensagem da interface "sensor_msgs__msg__Imu".
- "/micro_laserscan": Publica as leituras do LIDAR, incluindo distância, ângulo de leitura e qualidade dos dados, utilizando a interface "sensor_msgs__msg__LaserScan".

Para sincronizar as publicações e subscrições no micro-ROS, o sistema utiliza o *executor* fornecido pelo micro-ROS, que verifica periodicamente as atualizações nos tópicos e garante que os dados sejam enviados e recebidos corretamente. O tópico "/cmd_vel", responsável por controlar a velocidade e direção dos motores, recebe comandos do servidor ROS 2, que são aplicados no ESP32-S3 para a movimentação da plataforma.

A implementação do sistema foi testada em laboratório, com foco na sincronização dos dados coletados pelos sensores e a comunicação entre o ESP32-S3 e o servidor ROS 2, garantindo que a plataforma robótica executasse as tarefas de navegação e tomada de decisão de forma correta e autônoma. A Figura 3 apresenta o diagrama de blocos do *software* implementado.

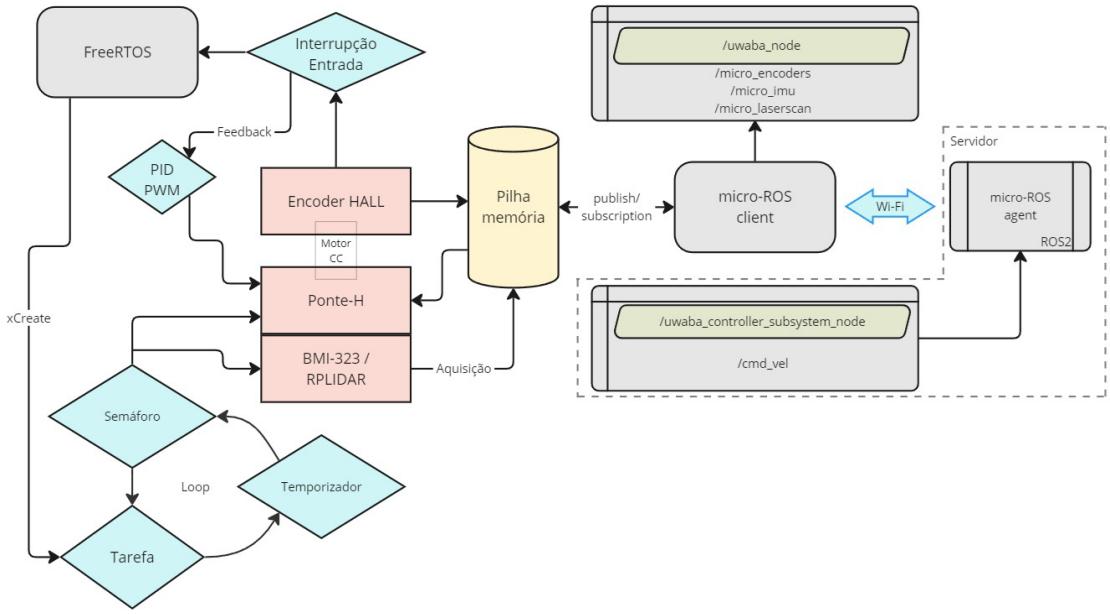


Figura 3: Diagrama de blocos do *software*. Fonte: autoria própria.

3 RESULTADOS E DISCUSSÃO

A partir da utilização do FreeRTOS em conjunto com o ESP-IDF foi possível realizar a aquisição dos dados individuais de cada sensor. Os dados foram corretamente adquiridos e armazenados em uma pilha onde constantemente uma tarefa com a sincronização do executor do micro-ROS foi executada. A utilização de semáforos no código permitiu que todas as tarefas fossem executadas de maneira a manter a frequência das mesmas e possibilitando que as aquisições dos sensores fossem sempre executadas.

As contagens das interrupções dos *encoders* provindas da rotação dos motores tiveram uma frequência no recebimento do servidor de aproximadamente 31 Hz. As aquisições obtidas pela comunicação do ESP32-S3 com os sensores conectados no barramento $I2C$ também tiveram uma frequência no recebimento de aproximadamente 31 Hz. O LIDAR, por se tratar de um dispositivo com comunicação assíncrona e limitado a frequência de envio do próprio modelo, apresentou a frequência de aquisição de aproximadamente 7 Hz.

O código desenvolvido neste trabalho é aberto e está disponível no GitHub: [https://github.com/uSANMA/microros-uwaba-prototype].

4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento uma plataforma robótica experimental utilizando o microcontrolador ESP32-S3 com programação baseada em micro-ROS capaz de se comunicar com um subsistema ROS 2 via Wi-Fi. A plataforma demonstrou-se funcional em testes laboratoriais, conseguindo gerenciar a navegação autônoma com a integração dos sensores LIDAR, BMI-323 e dos *encoders*. O projeto utilizou uma estrutura modular, empregando FreeRTOS para gerenciar as tarefas em tempo real, com sucesso na comunicação e na tomada de decisões automatizada pelo subsistema de controle ROS 2. O sistema demonstrou potencial para futuras aplicações em cenários reais, evidenciando a viabilidade da integração de dispositivos embarcados com ROS 2. Futuros aprimoramentos podem incluir o aumento da robustez da plataforma, visando sua aplicação em ambientes externos com condições desafiadoras de revelo em tarefas voltadas para o meio agrícola.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da UTFPR, Universidade Tecnológica Federal do Paraná — Brasil (Edital UTFPR/PROPPG 02/2023 — PIVIC/PIVIT).

REFERÊNCIAS

BELSARE, K. et al. Micro-ROS. In: Koubaa, A. (eds) Robot Operating System (ROS). Studies in Computational Intelligence, vol 1051. Springer, Cham. 2023. Disponível em: https://doi.org/10.1007/978-3-031-09062-2_2. Acesso em: 10 set. 2024.

HACKENHAAR, N. M.; HACKENHAAR, C.; ABREU, Y. V. Robótica na agricultura. Interações (Campo Grande). 16 (1). 2014. Disponível em: https://doi.org/10.1590/1518-70122015110. Acesso em: 10 set. 2024.

MARTINS JUNIOR, F. S. D.; DUARTE, L. F. C.; ANDRADE, B. Sistema autônomo para ações de monitoramento e manejo agropecuário – Subsistema fase 1. In: XIV Seminário de Extensão e Inovação & XXIX Seminário de Iniciação Científica e Tecnológica da UTFPR, 2024, Francisco Beltrão. Anais do XIV Seminário de Extensão e Inovação & XXIX Seminário de Iniciação Científica e Tecnológica da UTFPR. 2024.