Universidade Federal de Ouro Preto
Escola de Minas
Engenharia de Controle e Automação

Igor Dos Santos Ferreira Coimbra

Sistema Embarcado para Monitoramento e Segurança de um Robô de Inspeção

Ouro Preto
2026

Igor Dos Santos Ferreira Coimbra

Sistema Embarcado para Monitoramento e Segurança de um Robô de Inspeção

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Orientador: Prof. Alan Kardek Rêgo Segundo, D. Sc.
Coorientador: Mário César Delunardo Torres, M. Sc.

Ouro Preto
2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Igor dos Santos Ferreira Coimbra

Sistema Embarcado para Monitoramento e Segurança de um Robô de Inspeção

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 29 de julho de 2026

Membros da banca

Prof. Dr. Alan Kardek Rêgo Segundo - Orientador (Universidade Federal de Ouro Preto)
Me. Mário César Delunardo Torres - Coorientador (Instituto Tecnológico Vale)
Prof. Dr. Bruno Nazário Coelho - Convidado (Universidade Federal de Ouro Preto)
Prof. Dr. Agnaldo José da Rocha Reis - Convidado (Universidade Federal de Ouro Preto)

Alan Kardek Rêgo Segundo, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 29/07/2026

Documento assinado eletronicamente por **Alan Kardek Rego Segundo**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 29/07/2026, às 14:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1149327** e o código CRC **62722780**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Resumo

A robótica móvel tem ganhado cada vez mais espaço em ambientes de mineração, pois permite a realização de inspeções por meio de sensores embarcados, reduzindo a exposição de operadores a condições perigosas. Nesse contexto, este trabalho apresenta o desenvolvimento de melhorias no sistema embarcado para o EspeleoRobô, um robô móvel destinado à realização de inspeções em ambientes confinados. As melhorias incluem a implementação de um sistema de tempo real para emissão de alertas de risco de tombamento, o controle da intensidade luminosa dos LEDs por meio de PWM e o monitoramento da autonomia da bateria. O desenvolvimento foi realizado no Laboratório de Robótica e Controle do Instituto Tecnológico Vale – Ouro Preto/MG (ITV-RoC), abrangendo o projeto eletrônico, a implementação do sistema de monitoramento do robô e os testes experimentais. Os resultados demonstraram que o sensor de orientação (BNO08X) apresentou boa precisão quando comparado a um sensor de referência de maior desempenho (Xsens MTi-G-710 GNSS/INS), evidenciando seu potencial como uma alternativa de menor custo para substituir a Xsens no monitoramento de inclinação do robô, sem a finalidade de substituir esse sensor em aplicações de navegação autônoma. Além disso, o sistema desenvolvido permitiu o ajuste da intensidade luminosa dos LEDs e o acompanhamento, em tempo real, dos parâmetros relacionados à autonomia das baterias, demonstrando o funcionamento adequado dessas funcionalidades. Conclui-se que a solução proposta contribui para aumentar a segurança das inspeções, possibilitando o processamento dos dados e a emissão de alertas em tempo real.

Palavras-chaves: Robótica Móvel. Sistemas Embarcados. Sistema de Tempo Real. Unidade de Medição Inercial (IMU). Monitoramento de Bateria.

Abstract

Mobile robotics has been increasingly adopted in mining environments, as it enables inspections through onboard sensors, reducing the exposure of operators to hazardous conditions. In this context, this work presents the development of improvements to the embedded system of the EspeleoRobô, a mobile robot designed to perform inspections in confined environments. The improvements include the implementation of a real-time system for issuing tipping risk alerts, controlling LED brightness using PWM, and monitoring battery autonomy. The development was carried out at the Robotics and Control Laboratory of the Vale Technological Institute – Ouro Preto/MG (ITV- RoC), encompassing the electronic design, implementation of the robot monitoring system, and experimental tests. The results demonstrated that the BNO08X orientation sensor presented good accuracy when compared to a higher-performance reference sensor, the Xsens MTi-G-710 GNSS/INS, highlighting its potential as a lower-cost alternative to the Xsens specifically for monitoring the robot's inclination, without the intention of replacing it in autonomous navigation applications. Furthermore, the developed system enabled the adjustment of LED brightness and the real-time monitoring of parameters related to battery autonomy, demonstrating the proper functioning of these functionalities. It is concluded that the proposed solution contributes to increasing inspection safety by enabling data processing and real-time alert generation.

Key-words: Mobile Robotics. Embedded Systems. Real-Time System. Inertial Measurement Unit (IMU). Battery Monitoring.

Lista de figuras

Lista de abreviaturas e siglas

ITV-RoC	Laboratório de Robótica e Controle do Instituo Tecnológico Vale
ITV	Instituo Tecnológico Vale
PWM	Pulse Width Modulation
MG	Minas Gerais
IMU	Inertial Measurement Unit
UFRJ	Universidade Federal do Rio de Janeiro
FreeRTOS	Free Real-Time Operating System
SMBus	System Management Bus
STR	Sistemas de Tempo Real
SLAM	Simultaneous Localization and Mapping
I2C	Inter-Integrated Circuit
AGABP	Adaptive Genetic Algorithm Optimized Back Propagation
VMMS	Vehicle Monitoring and Maintenance System
OBD-II	On-Board Diagnostics II
FTC	Fault-tolerant control
IP	Índice de proteção
SDA	Serial Data
SCL	Serial Clock Line
MPU	Motion Processing Unit
UART	Universal Asynchronous Receiver/Transmitter
LGA	Land Grid Array
ROS	Robot Operating System
RVIZ	Robot Operating System Visualization
NUC	Next Unit of Computing

PTU	Pan-Tilt Unit
MUX	Multiplexador
CRC-16	Cyclic Redundancy Check de 16 bits
GNSS	Global Navigation Satellite System
CSV	Comma-Separated Values
PCI	Placa de Circuito Impresso

Lista de símbolos

MHz	megahertz
KB	kilobyte
ms	milissegundos
V	volt
Wh	watt-hora
°C	grau Celsius
Kg	quilograma
mm	milímetros
°	grau
MB	megabyte
kΩ	quilo-ohm

Sumário

1 Introdução

A robótica tem demonstrado um crescimento contínuo, expandindo sua aplicação em diversas áreas, incluindo inspeções em ambientes de mineração. Os robôs de serviço têm se mostrado indispensáveis ao auxiliar operadores na identificação de falhas atuais e potenciais, evitando danos significativos aos equipamentos e garantindo a segurança dos profissionais (RESENDE *et al.*, 2023, p.1). Em ambientes hostis, a robótica tem se destacado na realização de inspeções em diferentes tipos de cenários e aplicações. Uma das primeiras aplicações dos robôs na indústria, por exemplo, foi manipular metais a alta temperatura, uma tarefa que anteriormente era realizada por operadores com instrumentos pesados de difícil manuseio. Já um robô adequado realiza essa tarefa sem maiores inconvenientes (PAZOS, 2002, p.10).

Em ambientes de mineração, a utilização de robôs para inspeção tem se tornado cada vez mais comum. Diversos tipos de robôs, com características distintas, são empregados para realizar inspeções específicas. Um exemplo é o projeto Robominers (Figura 1), financiado pelo programa Horizonte 2020 da União Europeia, que desenvolveu um robô capaz de operar em locais de difícil acesso, como minas abandonadas, pequenas jazidas e minas ultraprofundas, alcançando profundidades de até 3000 metros abaixo da superfície (BERNER; SIFFERLINGER, 2021, p.2).

Figura 1 – Robominers.

Fonte: Robominers (2023).

Um outro exemplo de robôs para inspeção é a plataforma robótica ROSI (Figura 2), resultado de uma parceria entre o Instituo Tecnológico Vale (ITV) e a Universidade Federal do Rio de Janeiro (UFRJ). Essa plataforma robótica é equipada com quatro *flippers*, um

mecanismo usado para locomoção, permitindo que o robô supere terrenos difíceis, como lama ou obstáculos e até mesmo escadas, um braço manipulador, quatro rodas e diversos sensores, sendo amplamente utilizada em inspeções de estruturas de transportadores de correias (CUNHA, 2022, p.2).

Figura 2 – ROSI realizando uma atividade de inspeção termográfica em uma correia transportadora.

Fonte: Acervo ITV.

Além disso, o EspeleoRobô (Figura 3), inicialmente desenvolvido pela Equipe de Espeleologia da Vale, foi posteriormente aprimorado pela equipe do (ITV-RoC) em Ouro Preto/MG (TORRES, 2022, p.15). Esse robô é muito aplicado em atividades de inspeção em locais de difícil acesso, ambientes confinados e áreas de acesso restrito, como dutos, bueiros de ferrovia, galerias de barragem, onde é difícil determinar sua posição em relação ao solo durante a atividade de inspeção.

Figura 3 – EspeleoRobô fazendo inspeção em uma tubulação.

Fonte: Acervo ITV.

Um dos desafios enfrentados por esses tipos de robôs está relacionado à mobilidade, especialmente no que diz respeito à estabilidade, que pode ser definida como a capacidade do corpo de se manter em equilíbrio sem tombar (ROCHA *et al.*, 2019, p.5). Na maioria dos casos, o EspeleoRobô é utilizado principalmente em ambientes confinados, onde não há a visão direta entre o operador e o robô. Nesses cenários, o uso de um módulo com giroscópio e acelerômetro torna-se essencial, pois, após a determinação de um ângulo crítico, é possível determinar se o ambiente em que o robô está se locomovendo é seguro para evitar possíveis tombamentos.

Além disso, o alto consumo de energia e as variações incomuns no padrão energético do robô podem afetar diretamente o desempenho dos sensores, tornando crucial o monitoramento ativo de sua alimentação. Embora existam técnicas de baixo consumo para sistemas embarcados, como modos de suspensão e operação em frequência reduzida, Simonović e Saranovac (2013, p.1) destacam que a maioria dos sistemas operacionais em tempo real carece de um gerenciamento nativo eficiente para esses modos de economia. Para contornar essa limitação no EspeleoRobô, desenvolveu-se um monitoramento contínuo de sua autonomia energética integrado ao FreeRTOS, assegurando o funcionamento da IMU. Adicionalmente, o sistema passou a monitorar o ângulo crítico do protótipo para evitar possíveis tombamentos. A Figura 4 apresenta um exemplo dessa vulnerabilidade operacional: durante uma atividade de inspeção experimental em um bueiro ferroviário, o robô sofreu um tombamento devido à ausência de feedback em tempo real sobre a sua inclinação.

Figura 4 – Imagem obtida pela câmera interna do EspeleoRobô no momento do tombamento.

Fonte: Acervo ITV.

1.1 Justificativas e Relevância

O controle de sistemas, como o de movimento, posição e velocidade, é amplamente aplicado na robótica para garantir precisão e eficiência. Esse recurso é especialmente

importante em tarefas onde o tempo e a exatidão são cruciais. Para obter um controle eficiente nos processos robóticos, é indispensável um sistema robusto que assegure o bom funcionamento do robô. Os sistemas de tempo real representam uma das formas de atender às restrições temporais impostas por esses controladores (MEDEIROS, 2017, p.15).

Um exemplo significativo é o modelo A*daptive Genetic Algorithm Optimized Back Propagation* (AGABP), que integra algoritmos genéticos adaptativos com redes neurais artificiais para detecção preditiva de falhas. Este sistema realiza coleta contínua de dados sensoriais (como temperatura e pressão), processamento em tempo real e análise comparativa, proporcionando diagnósticos precisos em motores-foguete líquidos, uma aplicação onde a margem para erro é mínima (YU; WANG, 2021, p.7). Essa capacidade de prever falhas com alta acurácia é crucial em sistemas aeroespaciais, onde falhas podem resultar em consequências catastróficas e perdas irreparáveis.

Na indústria automotiva, sistemas como o *Vehicle Monitoring and Maintenance System* (VMMS) representam outra aplicação importante dessas tecnologias. Utilizando a interface padrão *On-Board Diagnostics II* (OBD-II) combinada com algoritmos de *machine learning*, o VMMS possibilita o monitoramento constante dos parâmetros do veículo, a antecipação de possíveis falhas mecânicas e o envio de alertas preventivos. Essa solução traz benefícios tanto operacionais quanto econômicos, como maior segurança (principalmente para veículos autônomos), redução dos gastos com reparos emergenciais e prolongamento da vida útil dos componentes (SHAFI *et al.*, 2018, p.4).

A implementação de um sistema de tempo real (STR) no EspeleoRobô é essencial para monitorar continuamente seu consumo de energia e detectar inclinações perigosas da IMU, garantindo respostas rápidas a eventuais anomalias. Assim como em sistemas automotivos e industriais, essa solução deve processar dados dentro de prazos determinados para prevenir falhas críticas (FARINES; FRAGA; OLIVEIRA, 2000, p.4). Em ambientes de difícil acesso como cavernas, onde intervenções físicas são complexas, essa tecnologia se torna indispensável para assegurar a operação segura e eficiente do robô, evitando custos adicionais com reparos ou recuperações.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver e avaliar um sistema embarcado de monitoramento de parâmetros críticos baseado no FreeRTOS para o EspeleoRobô, com foco na identificação de ângulos críticos de inclinação por meio de uma IMU para a prevenção de tombamentos, além do monitoramento da autonomia energética e controle de subsistemas em operações de teleoperação.

- Projetar a arquitetura de software multitarefas em tempo real utilizando o FreeRTOS

para o gerenciamento concorrente da IMU e dos subsistemas do robô;

- Implementar um sistema de monitoramento da inclinação do EspeleoRobô capaz de emitir alertas em tempo real quando o ângulo crítico, definido a partir de ensaios experimentais, for atingido, auxiliando o operador na prevenção de tombamentos;
- Desenvolver um sistema de monitoramento da autonomia das baterias e de controle da intensidade luminosa dos LEDs por PWM, integrando essas funcionalidades ao sistema embarcado do EspeleoRobô;
- Validar experimentalmente o sistema de monitoramento da inclinação por meio da comparação entre a IMU BNO08X e a IMU Xsens MTi-G-710 GNSS/INS, verificando a eficácia da BNO08X para aplicações de monitoramento da inclinação do EspeleoRobô.

1.3 Metodologia

A metodologia deste trabalho foi dividida em quatro etapas. Inicialmente, foi realizada uma revisão bibliográfica sobre robótica móvel, sistemas de tempo real, IMUs e monitoramento de baterias. Em seguida, foi desenvolvido o sistema embarcado proposto para ser utilizado no EspeleoRobô, integrando o monitoramento de inclinação, o monitoramento das baterias e o controle da intensidade luminosa dos LEDs. Posteriormente, foram realizados testes experimentais para validar o funcionamento do sistema e avaliar o desempenho da IMU BNO08X em comparação com a Xsens MTi-G-710 GNSS/INS. Por fim, os resultados obtidos foram analisados para verificar a eficiência da solução proposta.

1.4 Organização e estrutura

Este trabalho está organizado em cinco capítulos. O Capítulo 1 apresentou uma introdução ao tema, abordando diferentes aplicações de robôs móveis na mineração, bem como a justificativa, os objetivos e a metodologia adotada no desenvolvimento do projeto. O Capítulo 2 apresenta a revisão de literatura, destacando os principais conceitos relacionados às IMUs, ao protocolo SMBus e aos sistemas de tempo real, além de exemplos de suas aplicações na robótica. No Capítulo 3 é descrito o desenvolvimento do projeto, incluindo a montagem do sistema, a implementação do software e os principais desafios encontrados durante sua execução. O Capítulo 4 apresenta os resultados obtidos nos testes realizados, incluindo a validação do sistema e a análise dos resultados experimentais. Por fim, o Capítulo 5 apresenta as considerações finais, destacando as principais conclusões do trabalho e sugestões para trabalhos futuros.

2 Revisão de literatura

Este capítulo apresenta a revisão de literatura, abordando pontos essenciais como a funcionalidade do SMBus e da IMU, além da história e evolução dos sistemas de tempo real. A síntese desses componentes destaca sua relevância para aplicações robóticas, demonstrando como o avanço dessas tecnologias viabilizou soluções mais robustas e eficientes.

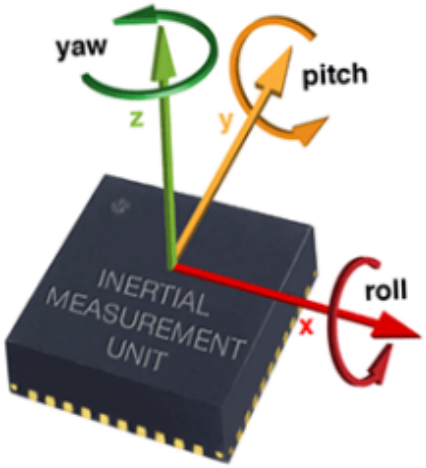
2.1 Inertial Measurement Unit (IMU)

As IMUs são utilizadas principalmente em dispositivos para medir velocidade, orientação e força gravitacional. Inicialmente, essa tecnologia consistia em dois tipos de sensores: acelerômetros e giroscópios. O acelerômetro é responsável por medir a aceleração inercial, enquanto o giroscópio mede a rotação angular. Ambos os sensores possuem três graus de liberdade, permitindo medições ao longo de três eixos. Posteriormente, a tecnologia IMU evoluiu com a incorporação de um terceiro tipo de sensor: o magnetômetro. Esse sensor mede a direção magnética, o que permite aprimorar a precisão das leituras do giroscópio (AHMAD *et al.*, 2013, p.1).

Para uma navegação eficiente de robôs, diversos desafios ainda persistem, como os relacionados ao mapeamento e à técnica de localização e mapeamento simultâneos (SLAM), planejamento de trajetórias e desvio de obstáculos (SAMATAS; PACHIDIS, 2022, p.1). Nesse contexto, a IMU desempenha um papel fundamental na robótica móvel, fornecendo dados essenciais para navegação, localização e controle de robôs. Ao medir parâmetros como aceleração, velocidade angular e orientação, a IMU permite que robôs móveis se movimentem de forma autônoma e precisa, mesmo em ambientes dinâmicos e desafiadores.

Basicamente, a IMU (Figura 5) utiliza acelerômetros, giroscópios e magnetômetros para medir aceleração, rotação e direção em tempo real, proporcionando maior precisão na movimentação do robô. Geralmente, elas são equipadas em sua maioria, por três giroscópios e três acelerômetros dispostos ortogonalmente. A partir do processamento dos dados fornecidos por esses sensores, é possível determinar a posição e a orientação de um dispositivo (WOODMAN, 2007, p.5).

Figura 5 – *Roll*, *Pitch* e *Yaw* de acordo com um sensor IMU.



Fonte: Chopra (2019, p. 38).

2.2 System Management Bus (SMBus)

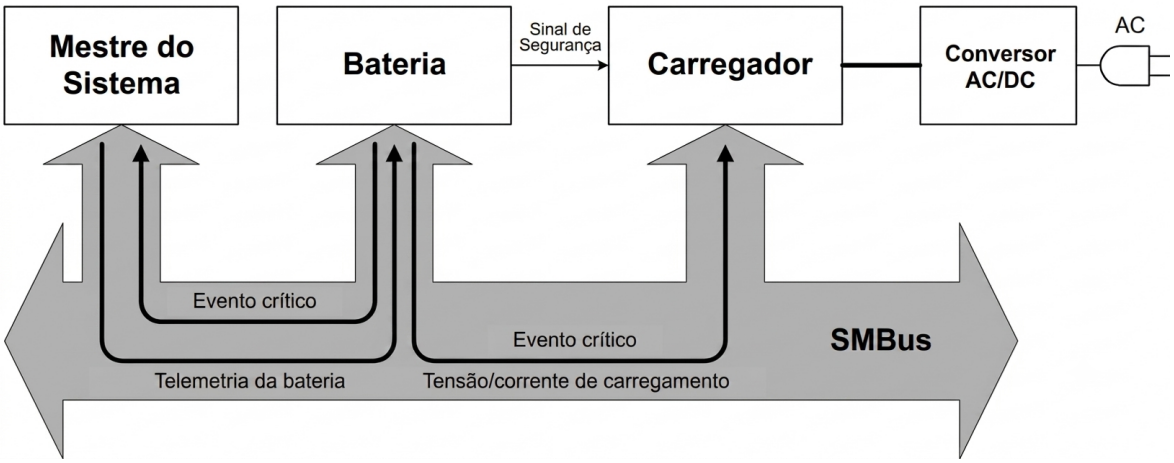
O SMBus foi desenvolvido para expandir a funcionalidade dos *chipsets* sem exigir a adição de mais pinos ou terminais, utilizando barramentos seriais de baixa velocidade para superar limitações físicas. Paralelamente, buscou-se uma solução eficiente e acessível para conectar baterias inteligentes ao sistema. Assim, o SMBus foi definido com base no protocolo I2C, adaptando características elétricas e introduzindo protocolos específicos para atender às necessidades das baterias. Suas especificações iniciais foram lançadas em 1994, enquanto drivers e suporte para sistemas operacionais, facilitaram o acesso e a integração das baterias inteligentes ao sistema. Isso ampliou sua aplicabilidade em dispositivos e sistemas (DUNSTAN, 1997, p.1).

Além de manter a simplicidade e a economia, o SMBus fornece um meio padronizado e de baixa latência para conectar a bateria ao sistema. Essa integração, apresentada na Figura 6, permite monitorar com precisão o estado da bateria, como nível de carga e condições operacionais. Uma bateria inteligente é capaz de monitorar seu próprio estado e se comunicar diretamente com um carregador inteligente, informando a necessidade de carga a qualquer momento. Além disso, ela pode se conectar diretamente a dispositivos inteligentes, como celulares, tablets e notebooks, por meio do SMBus. O uso de baterias inteligentes possibilita monitorar o estado da bateria com precisão e contribui para prolongar sua vida útil (LIU; ZHANG, 2011, p.232).

Nos robôs modernos, a integração crescente de sensores, câmeras e diversos outros componentes tem aumentado significativamente a demanda por maior capacidade de carga. Esse avanço tecnológico traz desafios na gestão energética. As baterias, frequentemente, apresentam variações em seu desempenho, como mudanças na duração de carga e eficiência,

dependendo das condições de uso e temperatura. Além disso, enfrentam problemas como a falta de informações confiáveis sobre seu estado, estimativas imprecisas de capacidade restante e uso inadequado, incluindo carregamento incorreto ou armazenamento em condições inapropriadas. Tais limitações comprometem a operação segura, especialmente em aplicações críticas. Estudos anteriores, como os citados por Lucas *et al.* (2005, p.21), destacam a importância de interfaces inteligentes e soluções avançadas de monitoramento e gerenciamento de baterias para melhorar a eficiência energética, prolongar a vida útil das baterias e garantir o funcionamento adequado de robôs em cenários de alta demanda.

Figura 6 – Comunicação entre dispositivos via SMBus.



Fonte: Adaptado de FORUM (1998).

2.3 Sistemas de tempo real

Os sistemas de tempo real são sistemas computacionais responsáveis por reagir dentro de uma determinada restrição de tempo precisa a um evento no ambiente. O comportamento correto depende não apenas do valor da computação, mas também do momento em que os resultados são produzidos. Caso a reação ocorra com atraso, ela pode se tornar inútil ou até mesmo causar algum tipo de dano (BUTTAZZO, 2011, p.1). Nesse contexto, sistemas de tempo real frequentemente utilizam sistemas operacionais como o FreeRTOS, devido à sua capacidade de gerenciar tarefas e interrupções com alta precisão temporal, assegurando que as respostas do sistema ocorram dentro dos limites de tempo exigidos.

2.3.1 FreeRTOS

O FreeRTOS é um sistema operacional gratuito e de código aberto, desenvolvido em linguagem C, amplamente utilizado em sistemas embarcados, como microcontroladores. Sua distribuição oficial inclui arquivos fonte comuns a todas as combinações de compiladores e arquiteturas suportadas, além de arquivos específicos para cada conjunto de compilador e arquitetura (SPOHN; CHABATURA NETO; FASSINI, 2020, p.2). Atualmente, o FreeRTOS está na versão 202604.00 LTS. O uso de FreeRTOS permite a criação de aplicações multitarefas eficientes, com suporte a tarefas, semáforos, filas, temporizadores e outras funcionalidades essenciais para o desenvolvimento de sistemas em tempo real.

2.3.2 História e evolução

Sistemas de tempo real são caracterizados por requisitos temporais explícitos, que podem ser determinísticos, com prazos rígidos que sempre devem ser cumpridos, ou probabilísticos, onde os prazos precisam ser atendidos em uma certa porcentagem de vezes (SHA *et al.*, 2004, p.2). O processamento nesses sistemas deve ser preciso e previsível, garantindo que as tarefas sejam concluídas dentro dos prazos especificados. No entanto, nem sempre foi assim: nas décadas de 1970 e 1980, o escalonamento era feito de forma estática, utilizando sistemas cíclicos, que, apesar de funcionais, eram inflexíveis e difíceis de manter. O reconhecimento dessas limitações levou ao lançamento da "*Real-Time Systems Initiative*" (Iniciativa de Sistemas em Tempo Real) pelo Office of Naval Research dos Estados Unidos, liderada por Andre van Tilborg, marcando um avanço crucial no desenvolvimento de infraestruturas para computação em tempo real como teorias de escalonamento baseadas em prioridades fixas, padrões abertos de hardware e software para suportar essas teorias, ferramentas comerciais de análise de escalonabilidade, materiais educacionais e de treinamento profissional (SHA *et al.*, 2004, p.2).

Em 1988, John A. Stankovic identificava grandes desafios no desenvolvimento de sistemas em tempo real. Segundo ele, duas forças principais estavam impulsionando esses sistemas para uma nova geração: a necessidade crescente de incorporar recursos de inteligência artificial e o rápido avanço tecnológico no hardware. Essas forças, contudo, ampliavam os já complexos problemas científicos e de engenharia envolvidos (STANKOVIC, 1988, p.2).

O autor destacava que essas transformações introduziam novas entidades complexas que precisavam ser integradas tanto em aplicativos existentes quanto em projetos futuros. Porém, as técnicas de design, análise e verificação ainda não haviam evoluído na mesma velocidade, gerando um descompasso preocupante. Ele também mencionava que inovações como computação distribuída e multiprocessamento, possibilitadas por avanços no hardware e software, estavam se tornando realidade. Com isso, previa-se o surgimento de redes robustas de multiprocessadores, mas a integração dessas tecnologias permanecia como um

desafio significativo na época (STANKOVIC, 1988, p.2).

Ao longo das últimas duas décadas, o RTOS tem passado por uma evolução constante, impulsionando o desenvolvimento de uma ampla gama de produtos comerciais baseados nesse sistema (HAMBARDE; VARMA; JHA, 2014, p.1). Esse progresso tem permitido sua aplicação em diversas áreas, atendendo demandas cada vez mais complexas e específicas.

Ainda assim, persistem desafios a serem enfrentados no campo dos sistemas de tempo real. A integração de recursos de tempo real em tarefas com comportamento dinâmico, aliada às limitações de custo e recursos disponíveis, gera novos obstáculos que precisam ser superados durante o processo de design desses sistemas em diferentes níveis de arquitetura. A abordagem clássica de projetar para o pior caso, comumente utilizada em sistemas de tempo real rígidos para assegurar respostas pontuais em todas as situações possíveis, tornou-se inadequada em ambientes altamente dinâmicos, pois resulta no desperdício de recursos e em um aumento excessivo dos custos (BUTTAZZO, 2006, p.1).

2.3.3 Aplicações de STR na Robótica

O avanço dos STR revolucionou os mecanismos de detecção de falhas na robótica moderna. Projetos atuais demonstram como a integração de STR permite não apenas maior precisão operacional, mas principalmente a implementação de sistemas de diagnóstico contínuo, capazes de identificar e responder a anomalias em tempo hábil, um fator crítico para operação em ambientes dinâmicos e não estruturados.

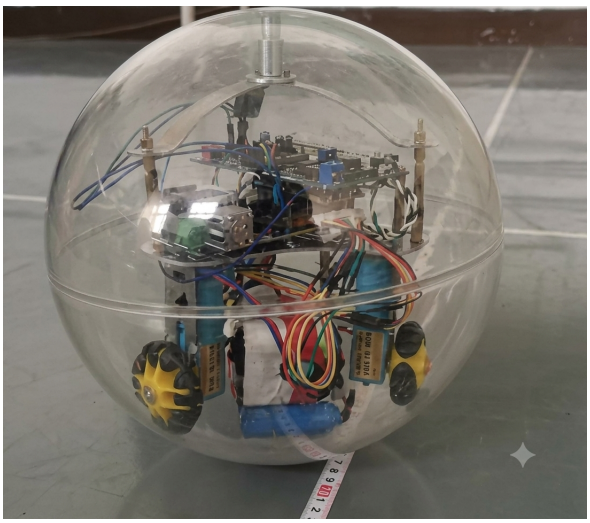
Essa mesma capacidade de processamento em tempo real tem sido fundamental para avanços em navegação autônoma. Zhuang, Du e Wu (2005) investigaram o uso de STR para planejar trajetórias e navegar em ambientes dinâmicos. Os autores destacaram a integração entre planejamento global e local, permitindo que os robôs adaptassem suas trajetórias dinamicamente com base em informações sensoriais processadas em tempo real. Sensores ultrassônicos foram utilizados para identificar obstáculos, enquanto algoritmos otimizados garantiam decisões ágeis. As simulações realizadas evidenciaram como esses sistemas aprimoram significativamente a capacidade dos robôs de operar de forma autônoma em cenários imprevisíveis.

Com o aumento da complexidade nos ambientes operacionais e a crescente integração de robôs em processos industriais críticos, torna-se essencial adotar estratégias avançadas que garantam a continuidade das operações. Sistemas de controle tolerantes a falhas, do inglês *Fault-Tolerant Control* (FTC) são altamente valorizados por sua capacidade de operar em tempo real, assegurando confiabilidade e mitigando impactos na produtividade e nos custos. Milecki e Nowak (2023) destacam que os FTC desempenham um papel fundamental na robótica, garantindo operação estável mesmo em cenários de falhas aceitáveis. Um algoritmo inovador foi implementado para controladores de manipuladores

com sete graus de liberdade, permitindo supervisão contínua e reconfiguração automática em processamento determinístico. Esses sistemas oferecem benefícios expressivos, como a manutenção da produção sem interrupções para manutenção, aumentando a segurança de equipamentos críticos e otimizando os custos operacionais. Além disso, essa abordagem previne paradas imprevistas, contribuindo diretamente para a eficiência e a economia nos processos de produção.

Lu *et al.* (2022) estudaram o uso de sistemas em tempo real no desenvolvimento de um robô esférico projetado para monitoramento de radiação (Figura 7). Utilizando o FreeRTOS como sistema operacional em tempo real, o robô foi capaz de gerenciar múltiplas tarefas, incluindo controle de movimento, detecção de radiação e comunicação, garantindo baixa latência e alta capacidade de resposta. A integração eficiente dos módulos permitiu ao robô adaptar-se rapidamente às mudanças no ambiente enquanto desempenhava suas funções essenciais. Experimentos realizados confirmaram a estabilidade e precisão do sistema, ressaltando sua relevância para aplicações robóticas em cenários críticos e dinâmicos.

Figura 7 – Robô esférico para monitoramento de radiação.



Fonte: Adaptado de Lu *et al.* (2022, p. 1262).

Docekal e Slanina (2017) utilizaram o FreeRTOS no desenvolvimento de um robô móvel projetado para robótica em enxame (Figura 8). Essa abordagem é inspirada no comportamento coletivo da natureza, como o de colônias de formigas ou abelhas, em que múltiplos robôs individuais, de custo acessível e com capacidades limitadas, interagem e colaboram de forma autônoma para resolver problemas complexos e mapear ambientes sem depender de um controle centralizado. Para que cada unidade do enxame consiga se movimentar com precisão e evitar colisões em tempo real, o sistema operacional organiza a aquisição de dados em tarefas (*tasks*) separadas e independentes. O destaque é o uso do

sensor 10-DOF, encarregado de monitorar a orientação espacial, a rotação e a inclinação angular do veículo por meio de giroscópio, acelerômetro e magnetômetro de três eixos. Dentro da arquitetura do FreeRTOS, uma tarefa (*task*) dedicada consulta esse sensor periodicamente e grava as informações em uma memória compartilhada. Em seguida, outra tarefa de alta prioridade acessa esses dados e os transmite ao computador principal (Raspberry Pi), garantindo determinismo e agilidade na tomada de decisões de navegação e na manutenção da estabilidade do robô.

Figura 8 – Robô projetado para robótica em enxame.

Fonte: Docekal e Slanina (2017, p. 435).

3 Desenvolvimento

Neste capítulo é apresentada toda a etapa de desenvolvimento do projeto, abrangendo os materiais utilizados, a implementação física do protótipo e o desenvolvimento do software embarcado no EspeleoRobô.

3.1 Materiais e Métodos

Nesta seção, são apresentados os materiais utilizados no projeto, bem como a implementação do STR no EspeleoRobô. Para facilitar a compreensão, o conteúdo foi dividido em três subseções: Hardware, que descreve os componentes empregados e suas especificações técnicas; Esquema Elétrico, que apresenta as conexões entre os componentes usados no trabalho, e Software, que apresenta a estrutura do pacote de comunicação serial do sistema.

3.1.1 Hardware

- **ESP32 C3 SuperMini:** Este microcontrolador destaca-se por sua capacidade de integrar múltiplos sensores e módulos essenciais ao EspeleoRobô. Ele suporta comunicação via Inter-Integrated Circuit (I2C), utilizada tanto pela IMU quanto pelos sistemas de monitoramento energético que operam sob o protocolo System Management Bus (SMBus). Seu núcleo RISC-V de 32 bits, operando a 160MHz, processa em tempo real os dados dos sensores, garantindo a detecção de anomalias com latência mínima, visto que esse desempenho é fundamental para operações em áreas de mineração e ambientes confinados. A escolha desta plataforma (Figura 9) justifica-se pelo seu excelente custo-benefício e eficiência operacional. Além disso, sua capacidade de 4MB de memória Flash é superior à de diversos microcontroladores da mesma categoria, sendo suficiente para comportar a pilha de execução do projeto; isso é particularmente relevante, visto que implementações de FreeRTOS podem exceder os limites de armazenamento de hardware mais simples. Por fim, o suporte nativo a Wi-Fi e Bluetooth viabiliza futuras expansões de conectividade, tornando-a ideal para os requisitos específicos deste projeto robótico.

Figura 9 – Vista do topo da placa da ESP32 C3 SuperMini.

Fonte: (CURTO CIRCUITO, 2025).

- **Baterias:** Os robôs móveis exigem baterias que aliem eficiência energética, dimensões compactas e autonomia, sendo especialmente úteis em determinados ambientes de inspeção, como os enfrentados pelo EspeleoRobô em locais de difícil acesso, onde pode haver água ou lama, dependendo da situação.

  Para atender a essas necessidades, foram selecionadas as baterias Bren-Tronics BT-70791CG (Figura 10). Essas baterias são recarregáveis, extremamente robustas (sendo utilizadas até em equipamentos militares) com características elétricas compatíveis com os dispositivos embarcados no robô, possui tensão nominal de 28,8V (máxima de 33,0V), capacidade de 294Wh, opera entre -20ºC e +60ºC, e pesa 1,4Kg. Além disso, elas contam com a conexão SMBus, um protocolo de comunicação que permite o gerenciamento inteligente da bateria, incluindo monitoramento de carga, temperatura e status, o que garante maior segurança e eficiência durante o uso.

Figura 10 – Bateria Bren Tronics BT-70791CG.

Fonte: Bren-Tronics (2024).

- **IMU GY-BNO08X:** Este sensor (Figura 11) permite a obtenção de dados de aceleração linear, velocidade angular e campo magnético nos três eixos, por meio da integração de um acelerômetro, um giroscópio e um magnetômetro. Sua utilização no projeto é fundamental, pois fornece dados confiáveis de orientação e movimento com alta taxa de atualização.

  A escolha por este componente ocorreu após a identificação de limitações práticas em testes preliminares com outros sensores. Inicialmente, utilizou-se o módulo MPU-6050. Contudo, quando os motores do robô eram acionados, o sensor sofria interferências e apresentava oscilações nas medições. Esse comportamento comprometia a precisão dos dados obtidos, sendo este o principal motivo para a substituição da MPU pela BNO08X.

  Encapsulado em um pacote Land Grid Array (LGA) de 28 pinos (3,8 mm x 5,2 mm x 1,1 mm), o sensor atua como um *hub* completo, reduzindo a complexidade e os custos de integração. O diferencial do BNO08X é o seu processador ARM Cortex M0+ interno, responsável por executar algoritmos avançados de fusão de sensores (CEVA, INC., 2023, p.1).

  De acordo com o datasheet da IMU BNO08X (CEVA, INC., 2023, p. 50), devem ser considerados três tipos de erro na avaliação do desempenho do sensor. O primeiro corresponde ao erro estático nominal, associado às medições realizadas com o sensor completamente imóvel, cujo valor típico é de 2,0°. O segundo refere-se ao erro dinâmico nominal, observado quando o sensor está em movimento, com valor típico de 3,5°. Por fim, em condições reais de operação, perturbações magnéticas e outros fatores ambientais podem aumentar a incerteza das medições, fazendo com que o erro prático de campo. atinja valores de até 5,0°.

Figura 11 – IMU CEVA GY-BNO08X.

Fonte: Ewald (2026).

- **IMU Xsens (MTi-G-710 GNSS/INS):** Esse sensor (Figura 12) se destaca pela sua precisão de orientação, sendo utilizado no EspeleoRobô em algoritmos de mapeamento que envolvem a fusão de sensores (IMU + SLAM). O manual

do fabricante (XSENS TECHNOLOGIES B.V., 2020, p. 35) especifica a precisão da orientação em função das condições de operação. Para os ângulos de *roll* e *pitch*, o erro típico é de 0,2° em condições estáticas e de 0,3° em condições dinâmicas. Para o ângulo de *yaw*, o erro típico é de 0,8°, considerando a utilização do sistema *Global Navigation Satellite System* (GNSS).

Essa abordagem é relevante para inspeções em ambientes de mineração, pois permite a localização do robô e a construção da representação do ambiente em locais de difícil acesso. Além disso, o sensor também é utilizado no monitoramento da inclinação do robô durante as inspeções, auxiliando na teleoperação.

Entretanto, vale ressaltar mais uma vez que a proposta deste trabalho é substituir a Xsens especificamente no cenário de monitoramento de inclinação do robô, não envolvendo as aplicações relacionadas ao mapeamento. O sensor é equipado com giroscópios imunes a vibrações, acelerômetros, magnetômetros, barômetro para estimativa de altitude e GPS/GNSS. (XSENS TECHNOLOGIES B.V., 2020, p. 38).

Figura 12 – Xsens MTi-G-710 GNSS/INS.

Fonte: Xsens Technologies B.V. (2020, p. 11).

- **Pan-Tilt Unit (PTU D47):** É um mecanismo motorizado, geralmente acoplado a uma câmera, capaz de realizar movimentos de rotação em dois eixos distintos, denominados *pan* (horizontal) e *tilt* (vertical) (KIKUCHI, 2007, p. 18). Neste trabalho, a PTU (Figura 13) foi utilizada como plataforma de movimentação controlada para a realização dos testes comparativos entre os sensores BNO08X e Xsens. Por meio da variação controlada dos ângulos de pan e tilt, foi possível submeter ambos os sensores aos mesmos movimentos, com maior precisão do que seria possível por meio de movimentação manual. Dessa forma, a utilização da PTU permitiu uma comparação mais confiável entre as medições fornecidas pelos dois sensores.

Figura 13 – PTU-D47.

Fonte: FLIR Systems, Inc. (2014).

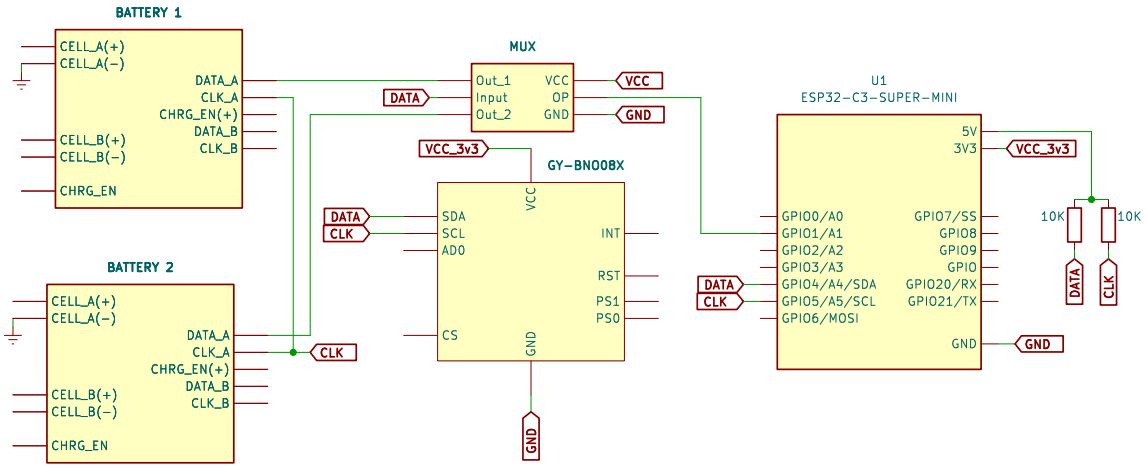
3.1.2 Esquema Elétrico:

A Figura 14 apresenta o diagrama elétrico do sistema desenvolvido, evidenciando as conexões entre as duas baterias, o multiplexador (MUX), o microcontrolador ESP32-C3 Super Mini e a IMU BNO08X.

Como os circuitos internos de gerenciamento das baterias utilizam o mesmo endereço de comunicação (*0x0B*), o qual é fixo e não pode ser alterado, não é possível acessá-los simultaneamente pelo mesmo barramento. Para contornar essa limitação, foi empregado um multiplexador, responsável por selecionar qual dispositivo seria conectado à ESP32 em cada instante. Dessa forma, o microcontrolador realiza o chaveamento entre os canais do MUX, permitindo a leitura alternada dos dados de cada bateria, sem que ocorram conflitos de comunicação.

O diagrama também ilustra a conexão da IMU ao microcontrolador por meio do mesmo barramento I2C. O sensor é alimentado com 3,3V, tensão compatível com suas especificações de operação. Além disso, foram utilizados resistores de *pull-up* de 10kΩ nas linhas *Serial Data* (SDA) e *Serial Clock Line* (SCL), garantindo a integridade dos sinais e o correto funcionamento da comunicação.

Figura 14 – Esquema elétrico das conexões entre as duas baterias, o multiplexador I2C, a IMU BNO08X e o microcontrolador ESP32-C3 Super Mini.



Fonte: O autor.

3.1.3 Software

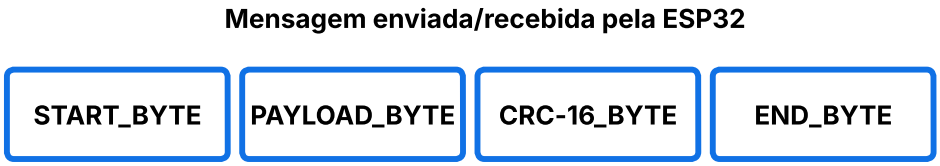
- **Estrutura do pacote de comunicação serial:** Durante o desenvolvimento do sistema, a comunicação entre a ESP32 e o computador embarcado no robô foi implementada por meio da interface serial *Universal Asynchronous Receiver/Transmitter* (UART). Inicialmente, utilizou-se a biblioteca do *rosserial*, que permite a integração entre microcontroladores e o Robot Operating System (ROS). Entretanto, essa abordagem apresentou elevado consumo de memória na ESP32, além de maior latência na transmissão dos dados. Em razão dessas limitações, optou-se pelo desenvolvimento de um protocolo serial próprio, mais leve e eficiente para a aplicação.

  Com o objetivo de garantir a integridade das informações transmitidas, considerando a possibilidade de interferências e ruídos durante a comunicação serial, foi desenvolvido um protocolo baseado em um pacote de dados de tamanho fixo de 24 bytes mostrado na Figura 15. A estrutura da mensagem (*frame*) é composta por um *Start Byte*, um *Payload*, um campo de verificação *Cyclic Redundancy Check* de 16 bits (CRC-16) e um *End Byte*. O Start Byte, definido pelo valor hexadecimal *0xAA*, indica o início de uma nova mensagem e permite a sincronização entre a ESP32 e o computador embarcado.

  O *Payload* possui 20 bytes e contém as informações úteis da comunicação, incluindo os dados de orientação provenientes da IMU, porcentagem luminosa dos LEDs via PWM e as leituras relacionadas ao monitoramento das baterias. Em seguida, são transmitidos 2 bytes correspondentes ao CRC-16, utilizado para detectar possíveis erros durante a transmissão. Ao receber o pacote, o computador embarcado recalcula o CRC a partir dos dados recebidos e compara o resultado com o valor transmitido, validando a integridade da mensagem. Por fim, o pacote é encerrado com o *End*

Byte, definido pelo valor hexadecimal *0x55*, que indica o término da transmissão e indica o fim da mensagem. Dessa forma, o computador embarcado somente processa as informações quando a estrutura da mensagem é válida e a verificação do CRC confirma sua integridade, reduzindo a probabilidade de processamento de mensagens corrompidas durante a transmissão.

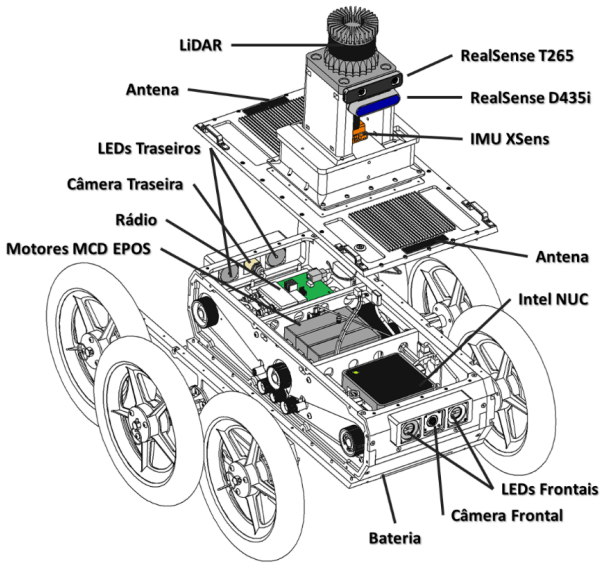
Figura 15 – Organização do frame das mensagens transmitidas pelo protocolo de comunicação serial.



Fonte: O autor.

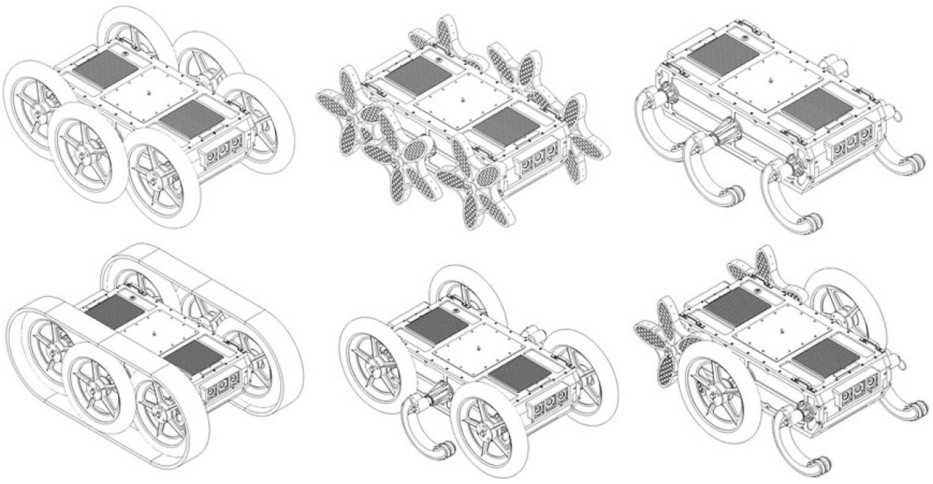
A Figura 16 apresenta a localização de alguns dos componentes mencionados anteriormente no Espeleorobô, como o computador embarcado, os LEDs, a IMU Xsens e as baterias. O robô possui versatilidade em seus modos de locomoção, podendo utilizar rodas, esteiras, patas ou rodas estreladas, de acordo com a configuração adotada como mostrado na Figura 17. Além disso, o robô possui capacidade para operar com até seis motores, embora, atualmente, utilize apenas quatro.

Figura 16 – Apresentação da construção física do Espeleorobô, detalhando seus componentes embarcados e a composição da torre de instrumentação.



Fonte: BARROS (2021, p. 31).

Figura 17 – Diferentes modos de locomoção do EspeleoRobô.



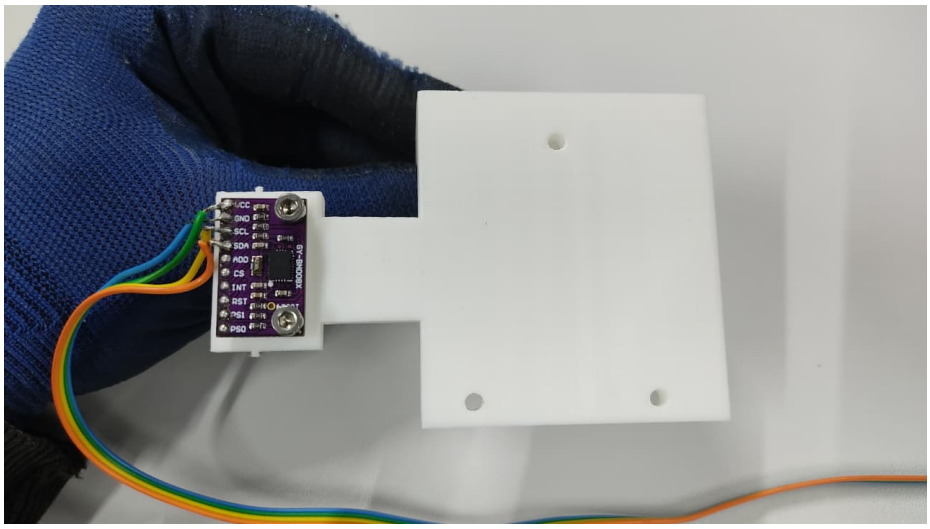
Fonte: Azpúrua *et al.* (2021, p. 3).

3.2 Suporte de Fixação das IMUs Desenvolvido por Impressão 3D

A placa de desenvolvimento foi montada em uma protoboard, na qual foram realizadas todas as conexões necessárias para o funcionamento do sistema. As ligações elétricas entre a ESP32 e a IMU foram cuidadosamente verificadas e testadas, garantindo a correta comunicação entre os dispositivos e minimizando o risco de falhas de funcionamento decorrentes de problemas de conexão. A escolha da placa deve-se ao fato de atender aos requisitos de processamento e comunicação do projeto, além de apresentar dimensões reduzidas, característica importante em razão do espaço interno limitado disponível no EspeleoRobô.

Para a realização dos ensaios experimentais, foi desenvolvido um suporte por meio de impressão 3D, conforme apresentado na Figura 18. O protótipo foi projetado para servir como base de fixação das IMUs BNO08X e Xsens, possibilitando a instalação dos sensores na plataforma de testes de maneira prática e estável.

Figura 18 – Protótipo feito por impressão 3D utilizado para fixação das IMUs.

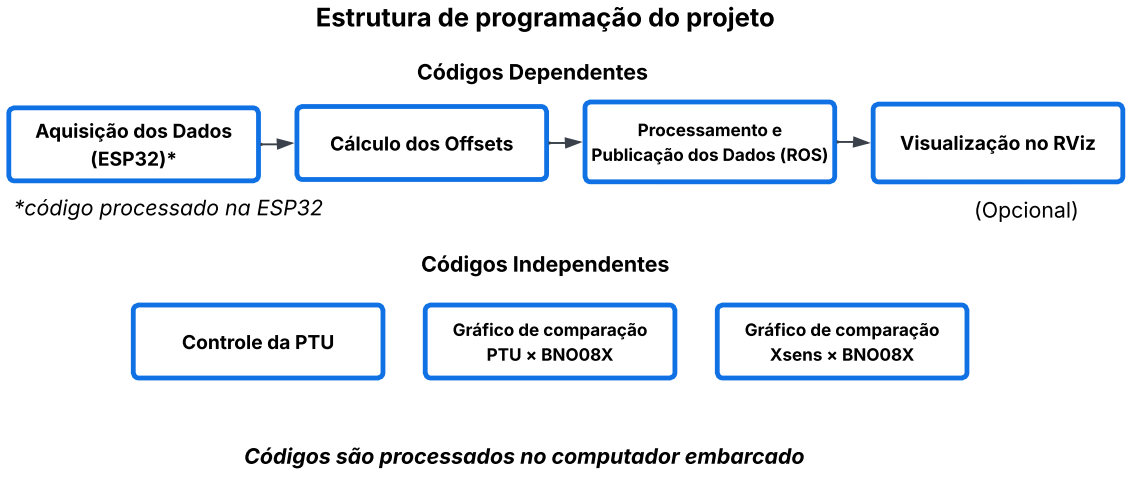


Fonte: O autor.

3.3 Desenvolvimento do código

Os códigos desenvolvidos para o projeto foram organizados em sete arquivos, divididos em dois grupos: quatro códigos dependentes, que devem ser executados em uma sequência específica para a calibração inicial e posterior funcionamento do sistema, e três códigos independentes, utilizados como ferramentas de apoio para testes e análise dos resultados (Figura 19). Todos os arquivos de código utilizados no projeto estão disponíveis publicamente em um repositório no GitHub[1].

Figura 19 – Organização dos códigos desenvolvidos para o sistema.



Fonte: O autor.

[1] https://github.com/igucoimbra/Projeto-Espeleo-

3.3.1 Scripts do Sistema Proposto (Códigos Dependentes)

O primeiro código, desenvolvido em C++ (com algumas simplificações), foi estruturado no formato conhecido como *sketch* e compilado pela IDE do Arduino e processados na ESP32-C3 Super Mini. Esse código é responsável pela leitura dos dados da IMU e da bateria por meio do protocolo I2C, pelo acionamento dos pinos de saída do microcontrolador e pela montagem do pacote de dados contendo o código de verificação CRC, que é enviado continuamente ao computador embarcado.

O escalonamento do sistema foi projetado utilizando o *FreeRTOS*, com a divisão das funcionalidades em três tarefas concorrentes: *TaskSensor*, *TaskBattery* e *TaskLed*. As prioridades foram definidas de acordo com a criticidade temporal de cada atividade. A tarefa *TaskSensor* recebeu a maior prioridade (3), pois realiza a aquisição dos dados do sensor inercial e sua transmissão, sendo diretamente relacionada ao monitoramento da inclinação do robô. A tarefa *TaskBattery* recebeu prioridade intermediária (2), devido à necessidade de realizar periodicamente a leitura das baterias por meio da comunicação I2C. A tarefa *TaskLed* recebeu a menor prioridade (1), uma vez que suas funções de sinalização e controle da iluminação possuem menor criticidade temporal.

Além disso, foi definido um tempo limite de 95ms para a obtenção de dados do sensor por meio da comunicação I2C, configurado na constante *I2C_TIMEOUT_MS*, utilizada tanto na configuração da biblioteca *Wire* por meio da função *Wire.setTimeout()*, quanto na função responsável pela obtenção dos dados da IMU. Esse mecanismo impede que a tarefa permaneça bloqueada indefinidamente em caso de falha ou indisponibilidade do sensor.

Na sequência, esses dados são processados no computador embarcado (Intel NUC) por um segundo código, desenvolvido em Python. Para representar a orientação espacial da IMU, o sistema utiliza *quaternions*, que consistem em uma representação matemática capaz de descrever a rotação de um objeto no espaço tridimensional. Diferentemente dos ângulos de Euler, que representam a orientação por meio de três rotações independentes (*roll, pitch e yaw*) e podem apresentar limitações durante determinados movimentos, os *quaternions* permitem representar essas rotações de forma contínua, evitando perdas de informação durante a movimentação do robô.

Os *quaternions* podem ser interpretados de várias maneiras, dentre elas como um vetor de dimensão quatro ou como um número complexo com três unidades imaginárias, considerando o escalar um (w) e os versores (i, j e k) como a base desse espaço (SILVA; LIMA; TAMAROZZI, 2015, p. 72), conforme apresentado na Equação 3.1:

$$q = w + xi + yj + zk \tag{3.1}$$

(w) representa o componente escalar do *quaternion*, enquanto (x), (y) e (z) corres-

pondem aos componentes vetoriais associados aos eixos de rotação no espaço tridimensional. A partir dessas quatro componentes, é possível determinar a orientação do sensor em relação a um sistema de referência.

Após o recebimento dos dados da IMU, os *quaternions* são corrigidos utilizando os *offsets* previamente calculados durante a etapa de calibração. Em seguida, o *quaternion* corrigido é normalizado, convertido para os ângulos de Euler (*roll, pitch e yaw*) e publicado em tópicos do ROS para utilização pelas demais aplicações do sistema.

3.3.2 *Script* de Calibração

Ainda no grupo de códigos dependentes do sistema, foi desenvolvido um *script* em Python dedicado exclusivamente à calibração, executado apenas durante a montagem e a configuração física do robô.

Nesse procedimento, os *offsets* dos *quaternions* são calculados a partir da média de amostras coletadas com o sensor em repouso. Esses valores funcionam como uma tara, definindo a orientação inicial como referência e compensando pequenos desvios de instalação do sensor. Após a fixação da IMU em um local definitivo, os *offsets* obtidos por esse *script* permanecem válidos e são armazenados para reutilização nas execuções subsequentes do sistema, não sendo necessária uma nova calibração, exceto em casos de reinstalação ou alteração física da posição do sensor.

3.3.3 *Scripts* de Suporte aos Testes e Ensaios Experimentais (Códigos Independentes)

Como ferramenta de apoio aos testes experimentais, o código responsável pelo processamento principal registra em um arquivo no formato *Comma-Separated Values* (CSV) os valores dos ângulos obtidos após a conversão dos *quaternions*. Essa funcionalidade pode ser habilitada opcionalmente por meio de comandos no terminal e é utilizada principalmente durante as etapas de testes experimentais. Dessa forma, os dados coletados podem ser armazenados para análises posteriores e comparação dos resultados obtidos.

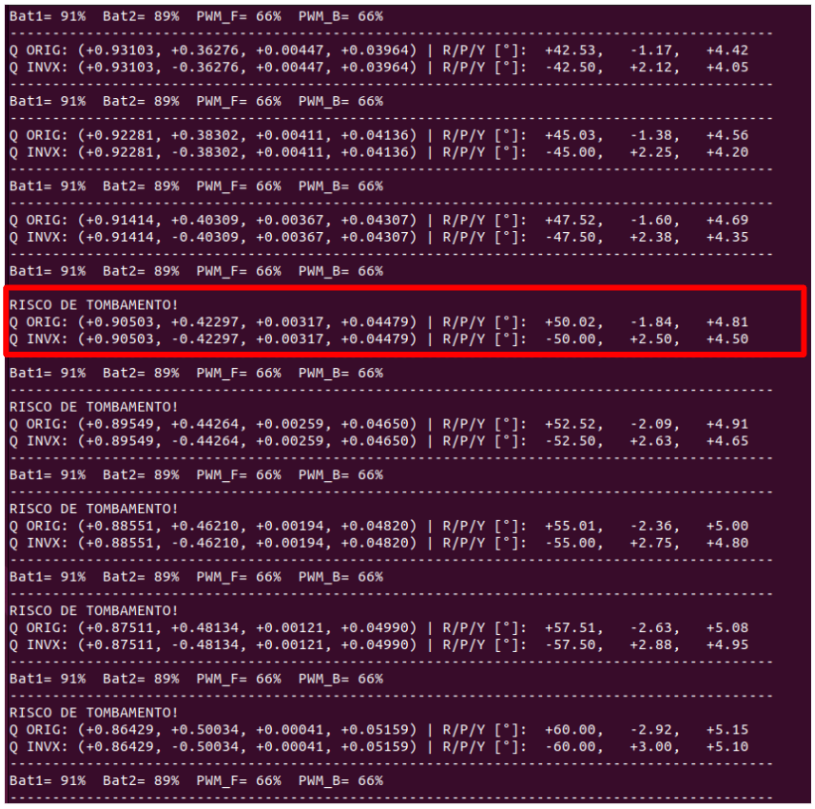
O segundo grupo é composto por três *scripts* independentes em Python, desenvolvidos exclusivamente para dar suporte aos testes e à análise dos resultados experimentais. O primeiro é responsável pelo controle da PTU, permitindo definir os ângulos de *pan* e *tilt* durante a execução dos ensaios. Os outros dois *scripts* são destinados ao pós-processamento dos dados coletados, realizando a geração de gráficos e a comparação matemática entre as medições da PTU e da IMU BNO08X, bem como entre as IMUs BNO08X e Xsens.

3.4 Emissão de alerta no terminal do sistema

A Figura 20 apresenta o instante em que a inclinação excedeu o valor de 50°. Esse valor foi definido de forma conservadora, uma vez que não foram realizados ensaios experimentais específicos para determinar o ângulo exato de tombamento do EspeleoRobô. Com base nas características construtivas da plataforma e na experiência da equipe com sua operação, considerou-se que, a partir dessa inclinação, o robô estaria em uma condição de elevado risco de tombamento, justificando a emissão do alerta.

Esse resultado evidencia que o STR foi capaz de processar os dados da IMU de forma eficiente, possibilitando a emissão dos alertas sempre que a inclinação atingiu uma condição crítica. Embora os testes tenham sido realizados utilizando a PTU, espera-se que, após a integração do sistema ao EspeleoRobô, o comportamento observado seja equivalente, permitindo a emissão dos alertas com a mesma eficiência durante sua operação em campo.

Figura 20 – Alerta de risco de tombamento monstrado no terminal.



Fonte: O autor.

3.5 Leitura dos dados das baterias no sistema

Os dados das baterias, são exibidos no terminal, fornecendo informações da autonomia em tempo real. A disponibilização contínua desses parâmetros, em conjunto com FreeRTOS, contribui para um monitoramento mais eficiente do estado das baterias durante a operação do EspeleoRobô. Dessa forma, o operador pode acompanhar continuamente as condições de funcionamento do sistema de alimentação, aumentando a segurança da inspeção e permitindo a identificação antecipada de possíveis anomalias que possam comprometer nas atividades de inspeção do robô.

Além dos dados referentes à bateria, o sistema também realiza o controle da iluminação do robô por meio de duas saídas PWM. O ciclo de trabalho pode ser ajustado de 0% a 100%, permitindo controlar a intensidade luminosa dos LEDs embarcados na plataforma.

4 Resultados

Este capítulo apresenta os resultados obtidos nos testes e na validação das funcionalidades desenvolvidas para o sistema embarcado do EspeleoRobô. Inicialmente, são apresentados os testes realizados com a IMU BNO08X, incluindo a validação da estimativa de orientação no ambiente de visualização do ROS (RVIZ), os ensaios com a plataforma PTU-D47 e a comparação de suas medições com os dados fornecidos pela IMU Xsens MTi-G-710, utilizada como referência. Em seguida, são discutidos os erros obtidos nas medições de *roll* e *pitch* e a adequação do desempenho da BNO08X para o monitoramento de inclinação do robô.

4.1 Testes realizados

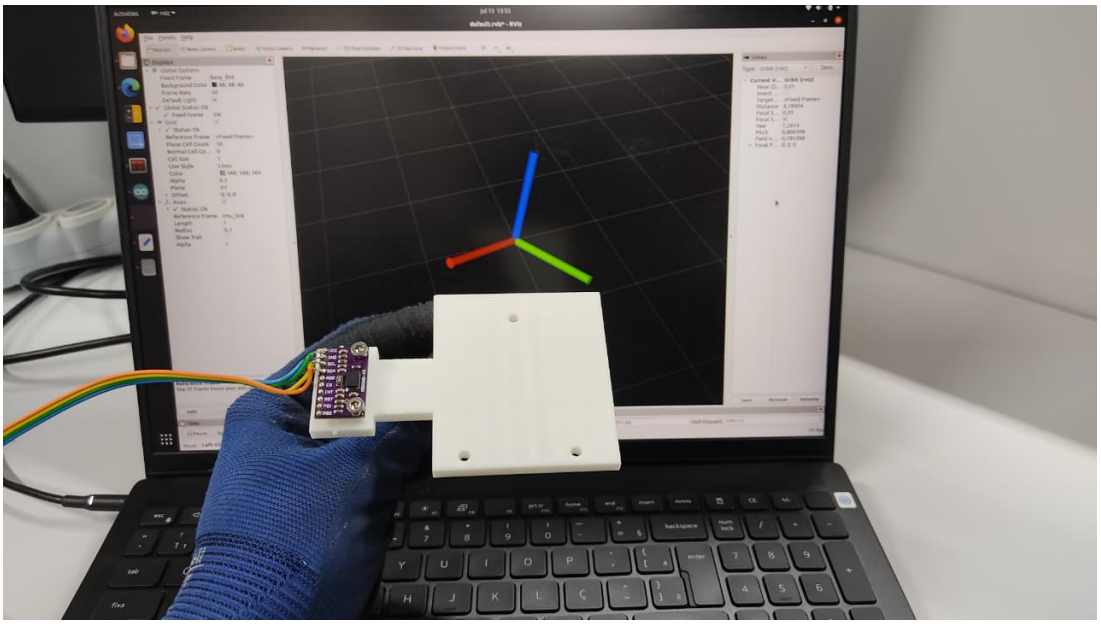
Com o objetivo de validar o sistema desenvolvido e verificar o desempenho da IMU BNO08X, foram realizados diferentes testes. Inicialmente, o funcionamento da estimativa de orientação foi avaliado no ambiente de visualização do ROS (RVIZ), permitindo identificar e corrigir limitações relacionadas às medições. Em seguida, foram realizados testes utilizando uma *Pan-Tilt Unit* (PTU), possibilitando a aplicação de movimentos angulares controlados. Por fim, as medições obtidas pela IMU BNO08X foram comparadas com as fornecidas pela IMU Xsens, utilizada como sensor de referência, a fim de avaliar a precisão e a confiabilidade do sistema proposto.

4.1.1 Testes no ROS Visualization (RVIZ)

O RVIZ é uma ferramenta integrada ao ROS que permite visualizar e analisar o comportamento dos ângulos medidos pelo sensor de forma gráfica, tornando-se um recurso importante para a validação e simulação do sistema. Por meio dessa ferramenta, é possível acompanhar o comportamento da IMU durante diferentes movimentos e verificar se as medições estão de acordo com os deslocamentos realizados (Figura 21).

Nessa etapa, observou-se, durante os primeiros testes de leitura dos ângulos da IMU no RVIZ, que o eixo *roll* apresentava sentido de rotação invertido em relação ao movimento realizado. Diante dessa inconsistência, foram realizados ajustes no código para corrigir a orientação desse eixo, garantindo que as leituras fornecidas pela (BNO08X) representassem corretamente os movimentos executados e aumentando a confiabilidade dos dados obtidos.

Figura 21 – Simulação no RVIZ nos três eixos *Yaw*, *Pitch* e *Roll*.

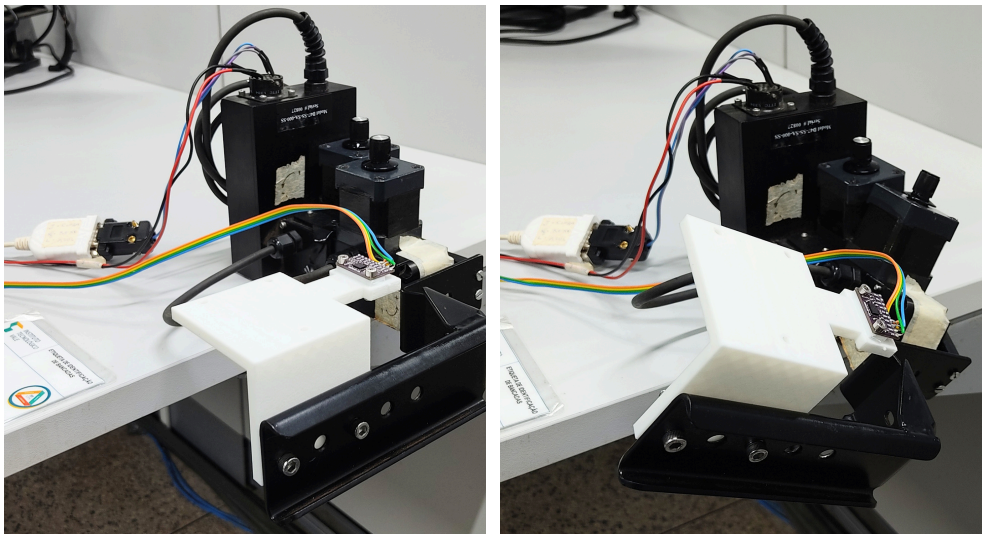


Fonte: O autor.

4.1.2 Teste com a Pan-Tilt Unit (PTU)

Nesta etapa, foram realizados testes utilizando uma PTU. Conforme apresentado na Figura 22, o sensor foi fixado à estrutura móvel da plataforma, garantindo que ambos estivessem submetidos aos mesmos deslocamentos angulares. Dessa forma, foi possível comparar diretamente as medições realizadas pelo sensor inercial com os valores de referência fornecidos pelo sistema de movimentação, permitindo avaliar sua precisão e confiabilidade. Como os dois dispositivos compartilham a mesma orientação ao longo dos movimentos, a comparação dos resultados apresenta elevada consistência, contribuindo para a validação do método proposto.

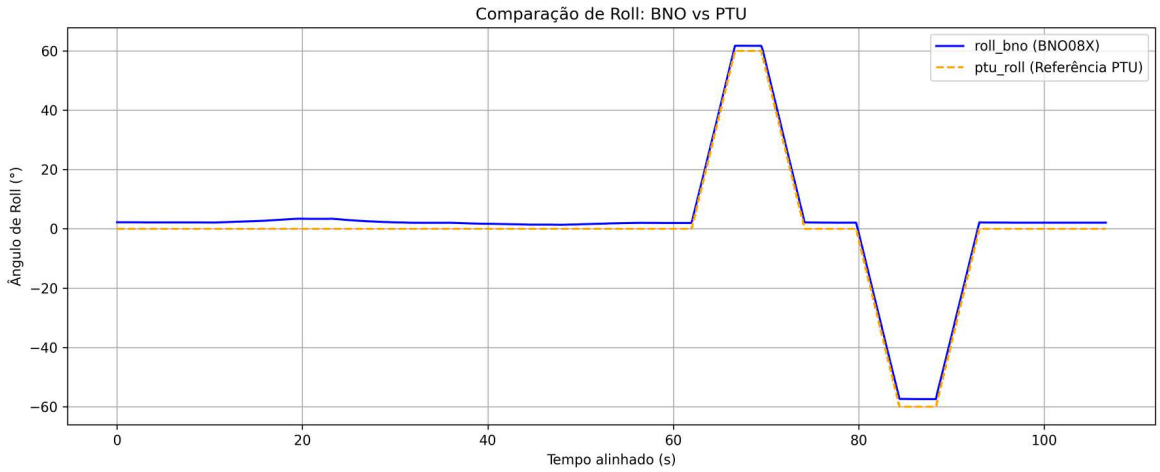
Figura 22 – Teste em diferentes ângulos com a PTU e a IMU.



Fonte: O autor.

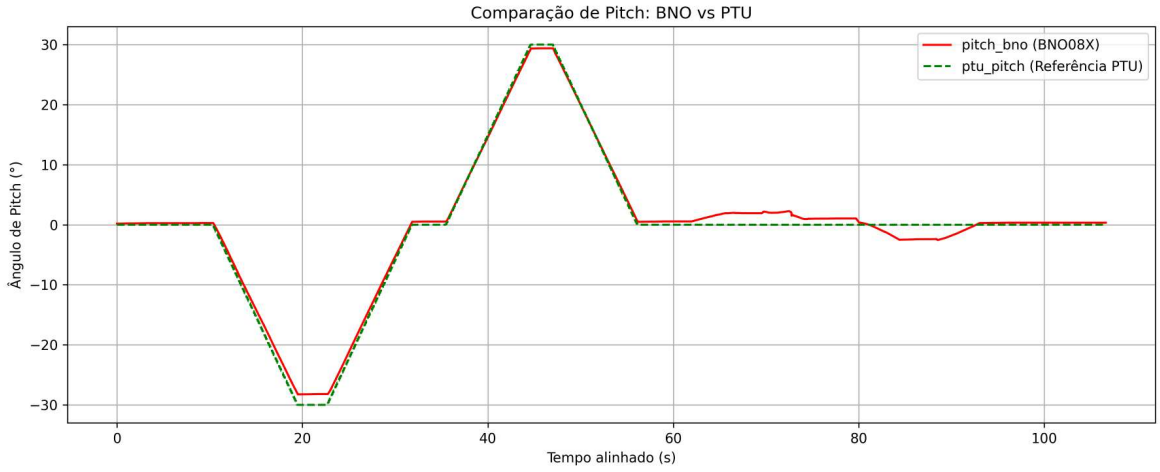
A análise comparativa indica que os dados de *roll* e *pitch* medidos pela IMU BNO08X se mantiveram bastante próximos aos de referência da PTU. No entanto, pequenas divergências são visíveis: a Figura 23 apresenta uma leve oscilação próximo aos 20 segundos, enquanto a Figura 24 mostra uma perturbação mais acentuada entre os instantes de 65 e 90 segundos, causada por vibrações da PTU que afetaram a precisão da leitura. Isso demonstra que, embora o sensor seja suscetível a ruídos mecânicos transitórios durante movimentações ou trepidações mais intensas da plataforma, ele apresenta excelente capacidade de recuperação. Nota-se que, assim que a vibração cessa (como após o instante de 90 segundos), as curvas voltam a se sobrepor quase perfeitamente, e isso demonstra a eficiência da fusão de sensores interno da IMU, que impede que o sensor acumule erros de leitura ao longo do tempo (fenômeno conhecido como *drift*).

Figura 23 – Comparação das medições do ângulo de roll entre a IMU BNO08X (azul) e a PTU (laranja).



Fonte: O autor.

Figura 24 – Comparação das medições do ângulo de pitch entre a IMU BNO08X (vermelho) e a PTU (verde).

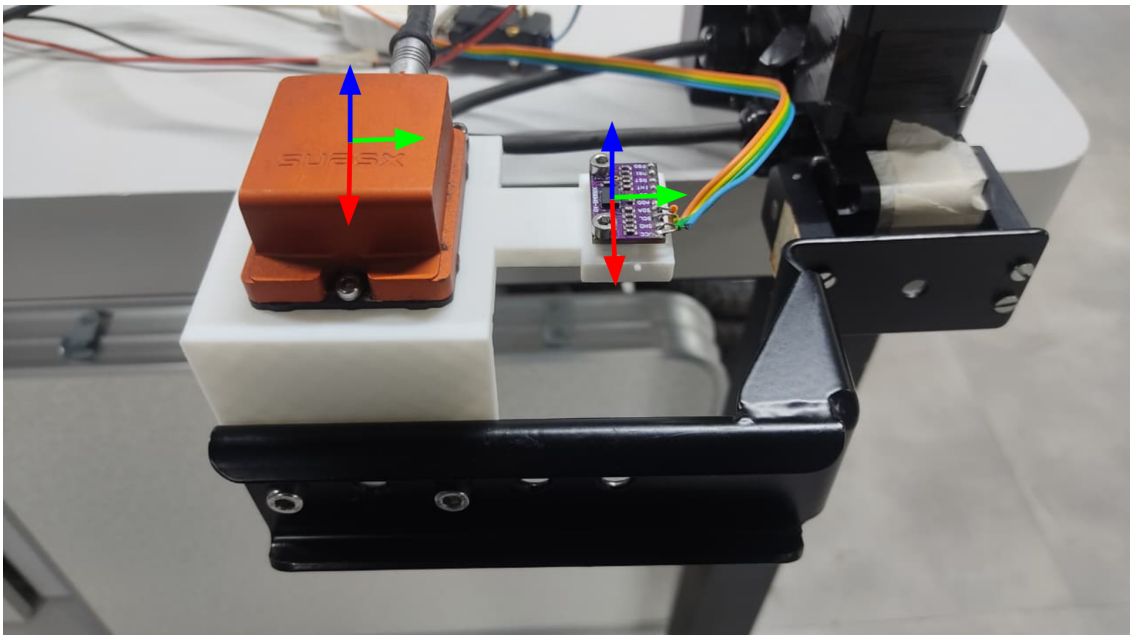


Fonte: O autor.

4.1.3 Comparação entre a Xsens e a BNO08X

Nesta etapa, foram realizados testes comparativos entre a IMU Xsens MTi-G-710 GNSS/INS e a IMU BNO08X utilizando a base da PTU. Ambos os sensores foram fixados na estrutura móvel da PTU por meio de um suporte desenvolvido e fabricado em impressão 3D, permitindo que as duas IMUs fossem instaladas lado a lado e com seus eixos alinhados. A Figura 25 apresenta a montagem utilizada, na qual os eixos de *yaw*, *pitch* e *roll* de ambas as IMUs foram posicionados na mesma orientação. Essa configuração permitiu que os dois sensores fossem submetidos simultaneamente aos mesmos movimentos controlados nos eixos de *pan* e *tilt*, garantindo condições idênticas durante os ensaios.

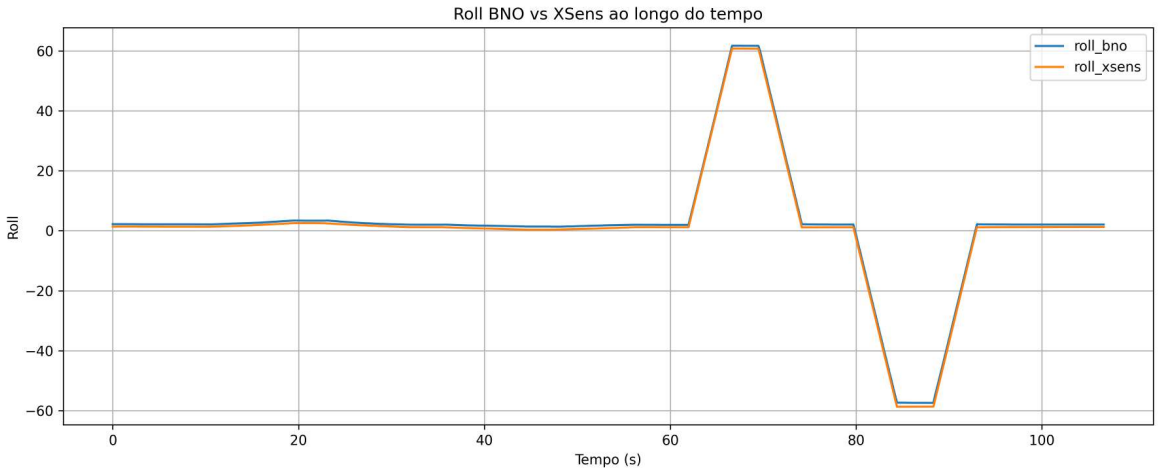
Figura 25 – Posição de *Yaw*, *Pitch* e *Roll* da Xsens e da IMU.



Fonte: O autor.

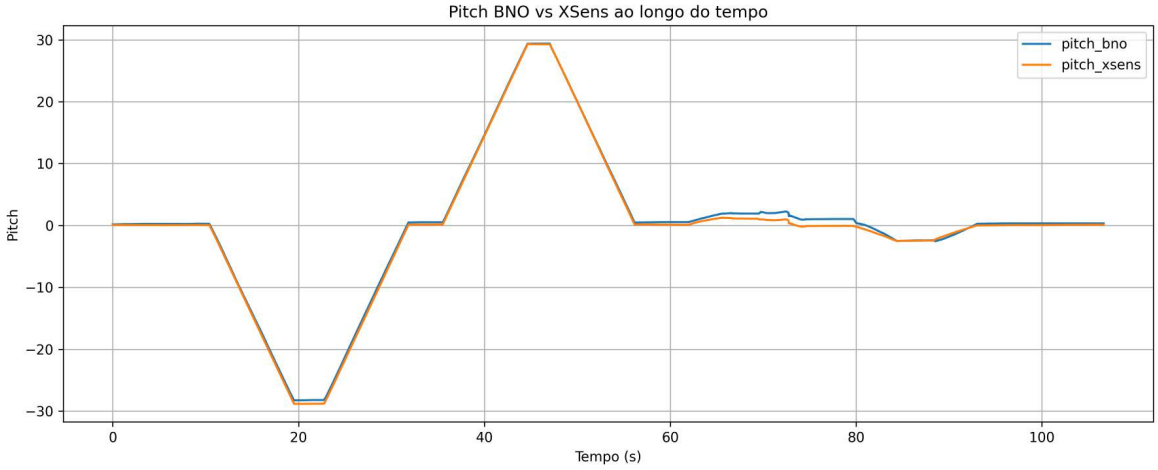
As Figuras 26 e 27 apresentam a comparação entre os ângulos medidos pelos dois sensores ao longo dos testes. Observa-se que as curvas obtidas pelo BNO08X acompanham de forma bastante próxima as medições fornecidas pela Xsens nos eixos roll e pitch, apresentando comportamento semelhante durante as variações de inclinação.

Figura 26 – Comparação das medições do ângulo de roll entre a IMU BNO08X (azul) e o sensor Xsens (laranja).



Fonte: O autor.

Figura 27 – Comparação das medições do ângulo de pitch entre a IMU BNO08X (azul) e o sensor Xsens (laranja).



Fonte: O autor.

Embora pequenas diferenças possam ser observadas em alguns instantes, os resultados indicam que a IMU BNO08X é capaz de fornecer medições com boa precisão e confiabilidade quando comparada a Xsens, que é um sensor de referência de maior custo. Dessa forma, a utilização da BNO08X representa uma alternativa viável para o projeto, permitindo reduzir significativamente o custo do sistema sem comprometer de forma relevante a qualidade das medições de orientação.

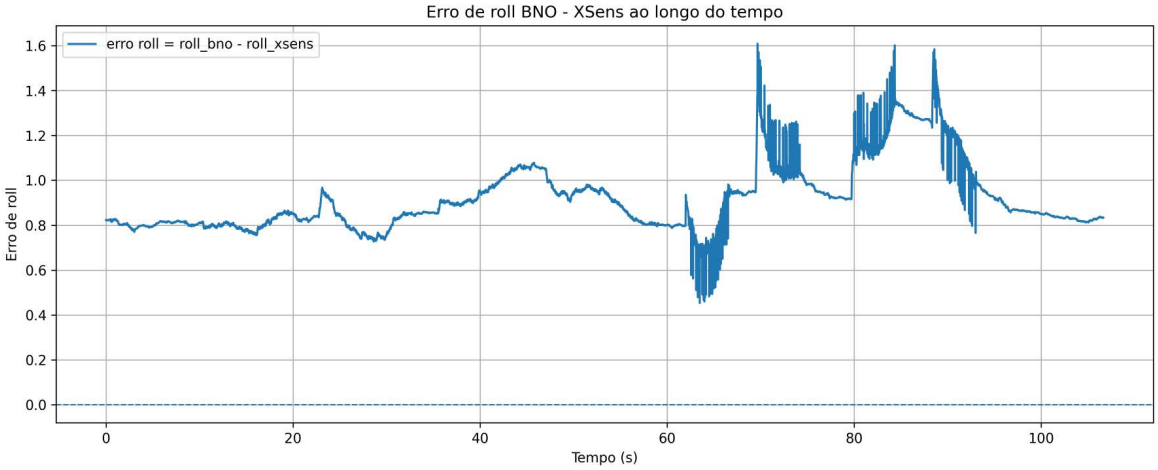
4.2 Discussão em relação aos erros

Como observado na Seção 3.1, relacionada aos materiais utilizados no trabalho, observa-se que a Xsens MTi-G-710 apresenta especificações de precisão significativamente superiores às da BNO08X, motivo pelo qual foi utilizada como sensor de referência nos ensaios experimentais realizados neste trabalho.

As Figuras 28 e 29 apresentam, respectivamente, os erros temporais obtidos para os ângulos de *roll* e *pitch* através da diferença direta entre as leituras da BNO08X e da Xsens MTi-G-710. A análise do comportamento dinâmico de ambos os gráficos revela que os dados apresentaram uma baixa dispersão de erro ao longo de todo o período ensaiado.

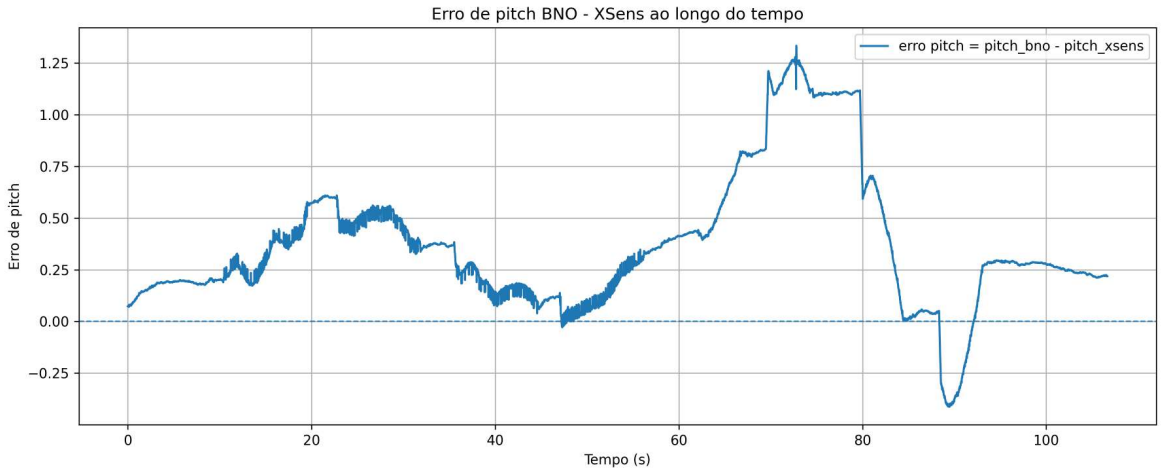
Para o ângulo de *roll* (Figura 28), observa-se que o erro oscilou aproximadamente entre 0,5° e 1,6°. Os picos transitórios e as variações mais bruscas visíveis no gráfico decorrem do ruído mecânico durante a movimentação da estrutura da PTU, mantendo-se, contudo, em uma amplitude bastante controlada. Comportamento semelhante é observado na análise de *pitch* (Figura 29), cujo erro médio também se posicionou dentro de uma faixa de oscilação muito próxima a zero, sofrendo flutuações equivalentes apenas nos instantes de transição de movimento do suporte motorizado.

Figura 28 – Gráfico de erro no eixo *roll* comparando a IMU Xsens com a IMU BNO08X.



Fonte: O autor.

Figura 29 – Gráfico de erro no eixo *pitch* comparando a IMU Xsens com a IMU BNO08X.



Fonte: O autor.

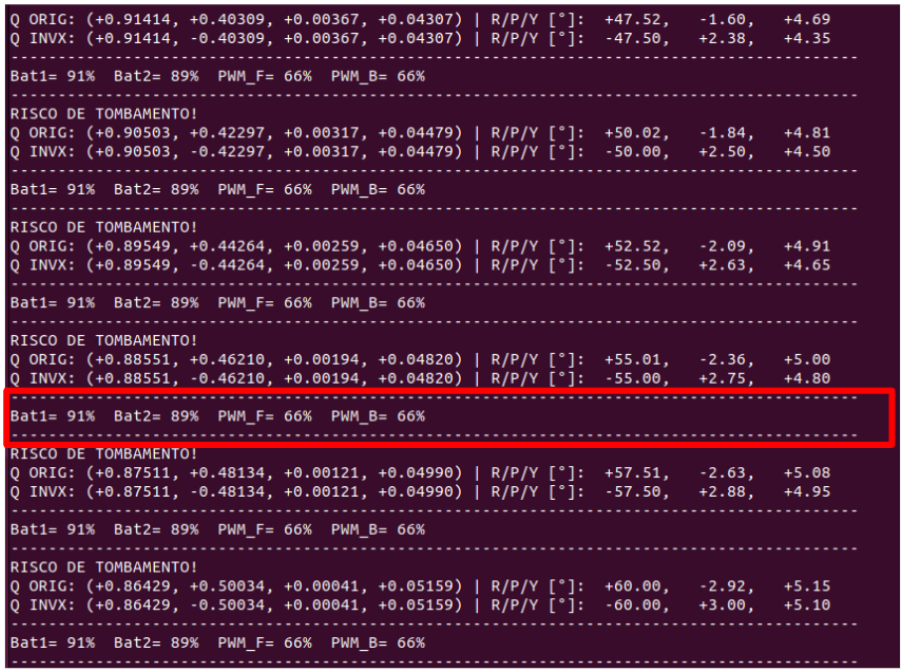
Tais resultados experimentais demonstram que, mesmo operando em ambiente real e sujeito a perturbações dinâmicas e vibrações mecânicas ativas, o sensor BNO08X apresentou um excelente desempenho prático, mantendo seu desvio médio substancialmente abaixo do limite máximo de 5,0° previsto em seu próprio *datasheet*. Assim, confirma-se a alta precisão e a viabilidade técnica da IMU BNO08X para aplicações de estimação de atitude em tempo real.

4.3 Resultados do Monitoramento de Alimentação e Controle de Iluminação

Os testes em bancada validaram o envio e a exibição em tempo real das telemetrias do robô na interface do terminal Linux. Conforme observado no terminal de telemetria destacado na Figura 30, o sistema exibe o nível percentual de carga individual de cada bateria embarcada no EspeleoRobô, apresentando valores de 91% para a Bateria 1 (Bat1= 91%) e 89% para a Bateria 2 (Bat2= 89%), o que garante a verificação do balanço de carga entre os módulos do sistema de alimentação.

Simultaneamente, o estado das saídas PWM para o sistema de iluminação frontal e traseiro é reportado operando ambos com um ciclo de trabalho de 66% (PWM_F= 66% e PWM_B= 66%), confirmando a atuação do controle de brilho conforme configurado pelo operador.

Figura 30 – Dados das baterias e as porcentagens luminosa dos LEDs do robô monstrado no terminal.



Fonte: O autor.

A consolidação dessas informações em uma linha de leitura de telemetria valida a eficácia do FreeRTOS no gerenciamento simultâneo das tarefas de leitura de sensores e envio das informações de status ao operador, assegurando a confiabilidade necessária para as expedições de inspeção em ambientes confinados.

5 Considerações finais

Este trabalho apresentou o desenvolvimento e a implementação de STR voltado à detecção de falhas e ao monitoramento preditivo para o EspeleoRobô. A utilização do sistema operacional FreeRTOS mostrou-se fundamental para gerenciar as restrições temporais do projeto, garantindo o processamento eficiente e determinístico dos dados da IMU BNO08X, das baterias e do controle de iluminação.

Os resultados experimentais demonstraram que os objetivos propostos foram alcançados. A comparação entre a IMU BNO08X e a Xsens MTi-G-710 GNSS/INS mostrou que a BNO08X apresentou desempenho compatível com a aplicação de monitoramento de inclinação, mantendo erros reduzidos em relação ao sensor de referência. Dessa forma, verificou-se que esse sensor constitui uma alternativa tecnicamente viável e de menor custo para aplicações voltadas ao monitoramento da inclinação do EspeleoRobô, sem a finalidade de substituir a Xsens em aplicações de navegação e mapeamento.

Em relação ao principal objetivo deste trabalho, o sistema desenvolvido permitiu identificar ângulos críticos de inclinação e emitir alertas em tempo real sempre que o robô se aproximava de uma condição de risco de tombamento. Essa funcionalidade representa um importante recurso de apoio à teleoperação, principalmente em ambientes confinados, onde o operador normalmente possui visão limitada do robô. Dessa forma, o monitoramento contínuo da inclinação contribui para aumentar a segurança operacional, auxiliando na prevenção de tombamentos e reduzindo os riscos durante as atividades de inspeção.

Além do monitoramento da inclinação, o sistema desenvolvido possibilitou o acompanhamento em tempo real dos parâmetros das baterias por meio do protocolo SMBus, permitindo ao operador acompanhar continuamente informações como autonomia e estado de funcionamento do sistema de alimentação. Também foi implementado o monitoramento da intensidade luminosa dos LEDs, ampliando os recursos disponíveis durante a operação do robô.

Durante o desenvolvimento, alguns desafios precisaram ser superados, principalmente relacionados ao entendimento do funcionamento da IMU BNO08X, à implementação da calibração, às limitações das bibliotecas disponíveis e ao gerenciamento das tarefas utilizando o FreeRTOS. A definição adequada das prioridades das tarefas e o tratamento da comunicação I2C foram fundamentais para garantir a estabilidade e o funcionamento confiável do sistema.

Por fim, conclui-se que o sistema desenvolvido atende aos objetivos propostos, fornecendo uma solução capaz de monitorar continuamente a inclinação e a autonomia energética do EspeleoRobô, emitindo alertas em tempo real para auxiliar a prevenção de

tombamentos durante operações de inspeção. Espera-se que este trabalho sirva de base para futuras melhorias no EspeleoRobô, contribuindo para o aumento da segurança, da confiabilidade e da eficiência das inspeções realizadas pela equipe do ITV-RoC.

5.1 Sugestões de trabalhos futuros

Como continuidade deste trabalho, recomenda-se a ampliação de tarefas do STR do EspeleoRobô, por meio da integração de novos sensores e da incorporação dos sensores já existentes na plataforma, aumentando a capacidade de monitoramento e detecção de anomalias em campo.

Recomenda-se a instalação da IMU BNO08X como componente padrão para medição de inclinação do EspeleoRobô, fornecendo os dados de inclinação essenciais para o monitoramento e o auxílio à teleoperação. Já para cenários mais avançados, nos quais haja necessidade de mapeamento do ambiente e navegação autônoma, sugere-se a utilização da IMU Xsens MTi-G-710 GNSS/INS.

Outra possibilidade consiste na incorporação dos alertas desenvolvidos à interface web do robô, acessível por meio de um endereço de rede. Dessa forma, as informações sobre risco de tombamento e o estado do sistema poderão ser apresentadas diretamente ao operador em uma interface gráfica, proporcionando maior praticidade durante a teleoperação.

Por fim, recomenda-se o desenvolvimento de uma placa de circuito impresso (PCI) para integrar os componentes eletrônicos do sistema embarcado como mostrado no capítulo 3, relacionado aos materiais na seção 3.1. Essa solução contribuirá para uma montagem mais compacta e organizada, além de aumentar a confiabilidade elétrica e a robustez do hardware, facilitando sua integração definitiva ao EspeleoRobô e a realização de futuras manutenções e expansões da plataforma.

Referências

AHMAD, Norhafizan *et al.* Reviews on various inertial measurement unit (IMU) sensor applications. *International Journal of Signal Processing Systems*, v. 1, n. 2, p. 256–262, 2013. Citado 1 vez na página 17.

AZPÚRUA, Héctor *et al.* Towards Semi-autonomous Robotic Inspection and Mapping in Confined Spaces with the EspeleoRobô. *Journal of Intelligent Robotic Systems*, v. 101, abr. 2021. DOI: 10.1007/s10846-021-01321-5. Citado 1 vez na página 31.

BARROS, L. G. D. de. *Desenvolvimentos e aplicações de dispositivos robôticos em ambientes de mineração : proposta de sistema de vedação para o EspeleoRobô II e ferramenta de remoção de cobertura de borracha para emenda de correias*. 2021. Disponível em: http://www.repositorio.ufop.br/jspui/handle/123456789/14054. Citado 1 vez na página 30.

BERNER, Michael; SIFFERLINGER, Nikolaus August. H2020 – ROBOMINERS. *BHM Berg- und Hüttenmännische Monatshefte*, v. 166, n. 2, p. 59–63, fev. 2021. ISSN 1613-7531. DOI: 10.1007/s00501-020-01074-y. Disponível em: https://doi.org/10.1007/s00501-020-01074-y. Citado 1 vez na página 12.

BREN-TRONICS. *BB-2590/U, 294 Wh*. USA, 2024. Acessado em: 19 mar. 2025. Disponível em: https://www.bren-tronics.com/bt-70791cg.html. Citado 1 vez na página 25.

BUTTAZZO, Giorgio. Research trends in real-time computing for embedded systems. *SIGBED Rev.*, Association for Computing Machinery, New York, NY, USA, v. 3, n. 3, p. 1–10, jul. 2006. DOI: 10.1145/1164050.1164052. Disponível em: https://doi.org/10.1145/1164050.1164052. Citado 1 vez na página 21.

BUTTAZZO, Giorgio C. *Hard Real-Time Computing Systems*. New York, NY, USA: Springer US, 2011. Citado 1 vez na página 19.

CEVA, INC. *BNO08X Data Sheet*. 2023. https://cdn.sparkfun.com/assets/2/b/9/0/6/DS-14686-BNO080.pdf. Acessado em: 9 de abril de 2026. Citado 2 vezes na página 26.

CHOPRA, Hetarth. *Programming Of Remote Surveillance Robot And Odometry Sensor Calibration Protocol For CMR's*. Jul. 2019. f. 100. Tese (Doutorado) – Thapar Institute of Engineering Technology. DOI: 10.13140/RG.2.2.19813.73446. Citado 1 vez na página 18.

CUNHA, Igor Novais Alves da. *Controle de um manipulador robótico para tarefas de inspeção de transportadores de correia*. 2022. Projeto de Graduação – Universidade Federal do Rio de Janeiro (UFRJ), Escola Politécnica, Engenharia de Controle e Automação, Rio de Janeiro, Brasil. Orientador: Fernando Cesar Lizarralde. Disponível em: http:

//www.repositorio.poli.ufrj.br/monografias/projpoli10038462.pdf. Citado 1 vez na página 13.

CURTO CIRCUITO. *Placa Super Mini ESP32-C3*. Acessado em: 6 jul. 2026. 2025. Disponível em: https://curtocircuito.com.br/placa-super-mini-esp32-c3.html. Acesso em: 6 jul. 2026. Citado 1 vez na página 25.

DOCEKAL, Tomas; SLANINA, Zdenek. Control system based on FreeRTOS for data acquisition and distribution on swarm robotics platform. *In:* 2017 18th International Carpathian Control Conference (ICCC). Maio 2017. p. 434–439. DOI: 10.1109/CarpathianCC.2017.7970439. Citado 2 vezes nas páginas 22, 23.

DUNSTAN, Robert A. SMBus Architecture and Implementation Overview, 1997. Acesso em: [20 mar. 2025]. Disponível em: https://www.sbs-forum.org. Citado 1 vez na página 18.

EWALD, Wolfgang. *BNO08x – 9-DoF-IMUs*. 2026. https://wolles-elektronikkiste.de/en/bno08x-9-dof-imus. Acesso em: 7 jul. 2026. Citado 1 vez na página 26.

FARINES, Jean-Marie; FRAGA, Joni da Silva; OLIVEIRA, RS de. Sistemas de tempo real. *Escola de Computação*, v. 2000, p. 201, 2000. Citado 1 vez na página 15.

FLIR SYSTEMS, INC. *PTU-D47: Fully-Integrated, Computer-Controlled Positioning*. Wilsonville, OR, USA, 2014. Datasheet. Updated October 8, 2014. Disponível em: https://www.sustainable-robotics.com/reference/PTU/ptu_d47_datasheet.pdf. Acesso em: 6 jul. 2026. Citado 1 vez na página 28.

FORUM, SMART BATTERY SYSTEM. *Smart Battery Data Specification*. 1998. Acessado em: 20 mar. 2025. Disponível em: http://sbs-forum.org/specs/sbdat110.pdf. Citado 1 vez na página 19.

HAMBARDE, Prasanna; VARMA, Rachit; JHA, Shivani. The Survey of Real Time Operating System: RTOS. *In:* 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies. 2014. p. 34–39. DOI: 10.1109/ICESC.2014.15. Citado 1 vez na página 21.

KIKUCHI, Davi Yoshinobu. Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência. *In:* disponível em: https://api.semanticscholar.org/CorpusID:62114193. Citado 1 vez na página 27.

LIU, Peide; ZHANG, Xiujuan. The Design of Smart Battery Management Systems. *J. Comput.*, Citeseer, v. 6, n. 11, p. 2484–2490, 2011. Citado 1 vez na página 18.

LU, Jing *et al.* Design and Implementation of a Spherical Robot for Radiation Monitoring. *In:* 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI). 2022. p. 1261–1266. DOI: 10.1109/PRAI55851.2022.9904103. Citado 2 vezes na página 22.

LUCAS, Nestor *et al.* RoBM2: measurement of battery capacity in mobile robot systems. *GI/ITG KuVS Fachgespräch Energiebewusste Systeme und Methoden, Erlangen, Germany*, p. 13–18, 2005. Citado 1 vez na página 19.

MEDEIROS, Lucas da Silva. Habilitando processamento de tempo real em sistemas embarcados ROS2., 2017. Citado 1 vez na página 15.

MILECKI, Andrzej; NOWAK, Patryk. Review of Fault-Tolerant Control Systems Used in Robotic Manipulators. *Applied Sciences*, v. 13, fev. 2023. DOI: 10.3390/app13042675. Citado 1 vez na página 21.

PAZOS, Fernando. *Automação de sistemas & robótica*. Rio de Janeiro: Axcel Books, 2002. Citado 1 vez na página 12.

RESENDE, Wlisses Moreira *et al.* IMPACTO DA AUTOMATIZAÇÃO NA SEGURANÇA DO TRABALHO: IMPLEMENTAÇÃO CRESCENTE DE ROBÔS E MÁQUINAS AUTOMATIZADAS. *Anais da Semana Universitária e Encontro de Iniciação Científica (ISSN: 2316-8226)*, v. 1, n. 1, 2023. Citado 1 vez na página 12.

ROBOMINERS. *Robominers field trials in Estonia audio visual coverage*. Estônia, 2023. Acessado em: 19 mar. 2025. Disponível em: https://robominers.eu/2023/09/11/robominers-field-trials-in-estonia-audio-visual-coverage/. Citado 1 vez na página 12.

ROCHA, Filipe *et al.* Análise e Comparaçao de Mobilidade de um Robô com Locomoçao Reconfigurável, 2019. Citado 1 vez na página 14.

SAMATAS, Gerasimos G.; PACHIDIS, Theodore P. Inertial Measurement Units (IMUs) in Mobile Robots over the Last Five Years: A Review. *Designs*, v. 6, n. 1, 2022. ISSN 2411-9660. DOI: 10.3390/designs6010017. Disponível em: https://www.mdpi.com/2411-9660/6/1/17. Citado 1 vez na página 17.

SHA, Lui *et al.* Real Time Scheduling Theory: A Historical Perspective. *Real-Time Systems*, v. 28, n. 2, p. 101–155, 2004. ISSN 1573-1383. DOI: 10.1023/B:TIME.0000045315.61234.1e. Disponível em: https://doi.org/10.1023/B:TIME.0000045315.61234.1e. Citado 2 vezes na página 20.

SHAFI, Uferah *et al.* Vehicle Remote Health Monitoring and Prognostic Maintenance System. *Journal of Advanced Transportation*, v. 2018, p. 1–10, jan. 2018. DOI: 10.1155/2018/8061514. Citado 1 vez na página 15.

SILVA, Amanda Santos; LIMA, Juliano Ferreira de; TAMAROZZI, Antônio Carlos. Quaternions e as rotações no espaço. *Colloquium Exactarum*, Universidade do Oeste Paulista (Unoeste), v. 7, n. 4, p. 71–77, out./dez. 2015. DOI: 10.5747/ce.2015.v07.n4.e142. Citado 1 vez na página 33.

SIMONOVIĆ, Mirela; SARANOVAC, Lazar. Power management implementation in FreeRTOS on LM3S3748. *Serbian Journal of Electrical Engineering*, v. 10, n. 1, p. 199–208, 2013. Citado 1 vez na página 14.

SPOHN, Marco Aurélio; CHABATURA NETO, Felipe; FASSINI, Leonardo Tironi. Uma avaliação do sistema operacional freertos na plataforma arduino uno. *Revista de Sistemas e Computação-RSC*, v. 9, n. 2, 2020. Citado 1 vez na página 20.

STANKOVIC, J.A. Misconceptions about real-time computing: a serious problem for next-generation systems. *Computer*, v. 21, n. 10, p. 10–19, 1988. DOI: 10.1109/2.7053. Citado 2 vezes nas páginas 20, 21.

TORRES, Mário César Delunardo. *Desenvolvimento e implementação da eletrônica embarcada em um dispositivo robótico móvel para inspeção: espeleorobô*. 2022. Monografia – Universidade Federal de Ouro Preto, Escola de Minas, Ouro Preto. 68 f. Citado 1 vez na página 13.

WOODMAN, Oliver J. *An introduction to inertial navigation*. Ago. 2007. DOI: 10.48456/tr-696. Disponível em: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf. Citado 1 vez na página 17.

XSENS TECHNOLOGIES B.V. *MTi User Manual: MTi 10-series and MTi 100-series 5th Generation*. Revision 2020.A. Fev. 2020. Document MT0605P. Disponível em: https://www.xsens.com/hubfs/Downloads/usermanual/MTi_usermanual.pdf. Citado 3 vezes na página 27.

YU, Huahuang; WANG, Tao. A Method for Real-Time Fault Detection of Liquid Rocket Engine Based on Adaptive Genetic Algorithm Optimizing Back Propagation Neural Network. *Sensors*, v. 21, n. 15, 2021. ISSN 1424-8220. DOI: 10.3390/s21155026. Disponível em: https://www.mdpi.com/1424-8220/21/15/5026. Citado 1 vez na página 15.

ZHUANG, Hui-Zhong; DU, Shu-Xin; WU, Tie-Jun. Real-Time Path Planning for Mobile Robots. *In:* 2005 International Conference on Machine Learning and Cybernetics. 2005. v. 1, p. 526–531. DOI: 10.1109/ICMLC.2005.1527001. Citado 1 vez na página 21.

APÊNDICE A – Declaração de uso de IA

Durante a preparação deste trabalho, o autor utilizou Gemini versão Gemini 3.6 Flash para modificar as Figuras 6 e 7, que não estavam com uma boa qualidade. Após o uso, o conteúdo foi revisado e editado, assumindo o autor total responsabilidade pelo texto final.