

Protótipo de Robô Autônomo para Mapeamento de Áreas com Terreno Irregular^{*}

Vitor Fernando de Souza Gadelha^{*} Sandro Azevedo da Mota^{*}
Marcos Vinícius Souza da Costa^{*} Almir Kimura Junior^{*}

^{*} *Escola Superior de Tecnologia, Universidade do Estado do Amazonas, Manaus, AM (e-mails: vfsg.eng@uea.edu.br, sam.eai18@uea.edu.br, mvsc.eai18@uea.edu.br, akimura@uea.edu.br)*

Abstract: Mobile robotics is increasingly present in the development of society, from simple tasks such as domestic services to more complex tasks such as rescue missions. Currently, one of the main challenges of mobile robots occurs when a robot is inserted in an unknown environment and with dynamic obstacles, having to recalculate its route several times to avoid collisions. Therefore, this article presents the modeling and construction of a low-cost autonomous terrestrial mobile robot prototype, capable of moving around in environments with uneven terrain and mapping the environment to which it is inserted. The prototype was developed using a *Raspberry Pi*, as well as several sensors capable of detecting obstacles and mapping the area and several actuators, such as DC motors and servomotors, which allow the robot to move in the environment. To make it as cheap as possible, it was decided to 3D print all the chassis parts, allowing greater freedom during the CAD modeling of the prototype and, to facilitate connections, it was decided to develop its own PCB in which the controller was docked, connecting the devices. For programming, we chose to use the ROS (Robot Operating System) to facilitate the connection between the various devices and because it contains packages for implementing the mapping algorithms, such as Gmapping and Hector SLAM. The robot was assembled and the results indicate potential in relation to the mapping and locomotion of the robot in environment with uneven terrain.

Resumo: A robótica móvel está cada vez mais presente no desenvolvimento da sociedade, fazendo parte de tarefas simples como serviços domésticos até tarefas mais complexas como missões de salvamento e resgate. Atualmente, um dos principais desafios dos robôs móveis se dá quando um robô está inserido em um ambiente desconhecido e com obstáculos dinâmicos tendo que recalculá-la sua rota diversas vezes para evitar colisões. Sendo assim, este artigo apresenta a modelagem e construção de um protótipo de robô móvel terrestre autônomo de baixo custo, capaz de se locomover em ambientes com terrenos irregulares e mapear o ambiente ao qual está inserido. O protótipo foi desenvolvido utilizando um *Raspberry Pi*, além de diversos sensores capazes de detectar os obstáculos e fazer o mapeamento da área e vários atuadores, tais como motores CC e servomotores, que possibilitam a locomoção do robô no ambiente. Para deixá-lo com um custo menor possível, optou-se pela impressão 3D de todas as peças do chassi, permitindo uma maior liberdade durante a modelagem CAD do protótipo e para facilitar as conexões, optou-se por desenvolver uma PCB própria em que o controlador fosse encaixado, conectando os dispositivos. Para a programação, optou-se por utilizar o ROS (Robot Operating System) por facilitar a conexão entre os vários dispositivos e conter pacotes para implementação dos algoritmos de mapeamento, tais como o Gmapping e o Hector SLAM. O robô foi montado e os resultados indicam potencial em relação ao mapeamento e locomoção do robô em ambientes irregulares.

Keywords: Mobile Robotics; *Raspberry Pi*; ROS; Mapping ; 3D Printing; CAD .

Palavras-chaves: Robótica móvel; *Raspberry Pi*; ROS; Mapeamento; Impressão 3D; CAD .

1. INTRODUÇÃO

A habilidade de navegar, monitorar ambientes e se locomover por ambientes é de suma importância para o desenvolvimento da sociedade. No entanto, nem sempre é possível fazer esse monitoramento em determinados locais, visto a dificuldade de acesso em determinados ambientes ou a hostilidade do ambiente ao ser humano. Sendo assim, muitas vezes se faz necessária a utilização de tecnologias como robôs móveis para solucionar esse problema.

Matarić (2007) define robô como: "um sistema autônomo que existe no mundo físico, pode sentir o seu ambiente e pode agir sobre ele para alcançar determinado objetivo". Dudek and Jenkin (2010) definem a robótica móvel com uma área de pesquisa que lida com o controle de sistemas autônomos e veículos semiautônomos. A diferença da robótica móvel para outras áreas como a robótica de manipuladores convencional é a ênfase em solucionar problemas que exigem a compreensão do espaço em grande escala, ou seja, regiões do espaço onde não é possível observar de um único ponto.

Um dos principais desafios dos robôs móveis se dá quando um robô está inserido em um ambiente desconhecido e com obstáculos dinâmicos, o que faz com que ele tenha que recalcular sua rota diversas vezes (e rapidamente) para chegar ao seu objetivo, evitando colisões. Como o terreno muitas vezes é desconhecido, se faz necessário encontrar uma forma com que o robô se adapte ao ambiente ao qual está inserido.

Bouton et al. (2020) apresentam o design, controle e implementação de um robô que possui rodas sob pernas e é capaz de se adaptar conforme o terreno em que está se locomovendo. As pernas foram desenvolvidas utilizando um sistema de molas, que fazem com que as rodas possam ser conformadas ao solo usando a própria força da gravidade. Já de Sena Filho et al. (2020) apresentam um robô autônomo para ambientes capaz de interpretar placas do Código de Trânsito Brasileiro. Em Köseoğlu et al. (2017), os autores apresentam o design de um robô utilizando o ROS (Robot Operating System), colocando todos os passos da modelagem e da programação.

Neste contexto, este artigo apresenta o processo de modelagem e construção de um robô autônomo de baixo custo e pequeno porte capaz de se locomover em diversos ambientes, cujas rodas se adaptam a pequenas irregularidades no solo. O robô foi desenvolvido utilizando um microcomputador *Raspberry Pi* para o processamento geral dos dados e um microcontrolador ESP32 para acionamento dos motores e sensores. Para diminuição do custo de produção das peças, todas foram projetadas para serem impressas utilizando uma impressora 3D.

Optou-se por utilizar o ROS como principal framework para desenvolvimento dos algoritmos, visto que facilita a comunicação entre os dispositivos e pode ser instalado facilmente no *Raspberry*.

Na próxima seção, será descrito como foi feita a arquitetura física do robô utilizando o software SolidWorks e o projeto da placa eletrônica utilizando o Proteus. A

* Este projeto foi parcialmente financiado pelo Samsung Ocean Manaus.

seção 3 apresentará a impressão 3D das peças. A seção 4 apresentará os testes e análise dos resultados. A seção 5 apresentará a programação dos dispositivos e testes dos algoritmos. Finalmente, as considerações finais e os trabalhos futuros serão abordados na seção 6.

2. MODELAGEM DO PROTÓTIPO

Para o desenvolvimento do robô, inicialmente foi feita a modelagem computacional do mesmo utilizando o software *Solidworks*, fazendo um modelo em que todos os dispositivos ficariam expostos, como na figura 1

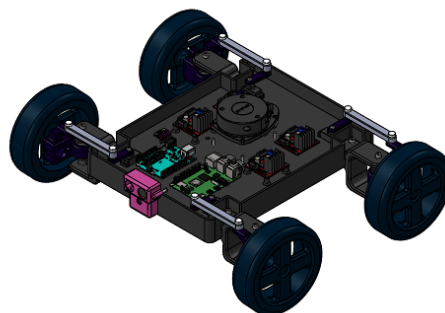


Figura 1. Projeto do Modelo 3D do robô

Após algumas simulações e testes, notou-se que o ideal era ter um robô fechado para que fosse possível seu uso tanto em ambientes externos quanto ambientes internos. Sendo assim, o chassi do robô foi projetado de forma a mantê-lo fechado, no entanto como todas as outras peças já mostravam um bom funcionamento, o design das mesmas foi mantido conforme figura 2.

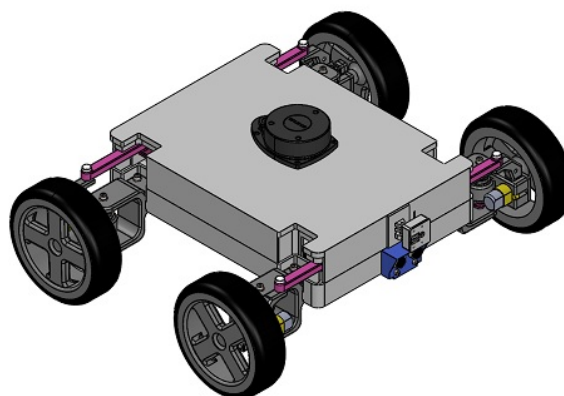


Figura 2. Projeto do Modelo 3D do robô

Todos os componentes foram fixados na base do chassi, feita através de impressão 3D utilizando filamento de ABS. O robô possui quatro rodas independentes o que traz uma maior liberdade de movimentos, utilizando quatro motores CC para a tração e quatro servomotores para controle de direção de cada roda. Para ativação dos motores de tração foram utilizados dois módulos Ponte H L298N, em que cada módulo fornece os níveis de tensão para dois motores diferentes.

2.1 Sistema de Molas

Para fazer com que o robô possa se locomover em terrenos irregulares, foi desenvolvido um sistema de molas, similar ao apresentado por Bouton, Grand e Benamar (2020), em cada roda a fim de diminuir o desnível de suspensão entre cada roda, que pode ser visto na figura 3.

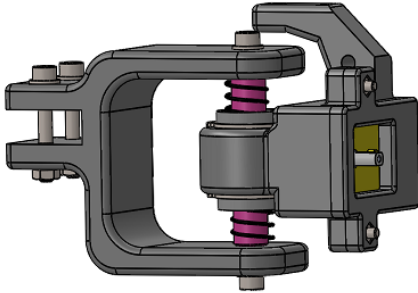


Figura 3. Sistema de molas projetado

2.2 Mecanismo de direção das rodas

Para a direção de cada uma das rodas foi desenvolvido um mecanismo de dois pequenos "braços", sendo um "braço" mais curto e outro mais longo, que fazem a transmissão de movimento do servomotor para o suporte do motor CC, alterando a direção das rodas a eles acoplados, conforme figura 4.

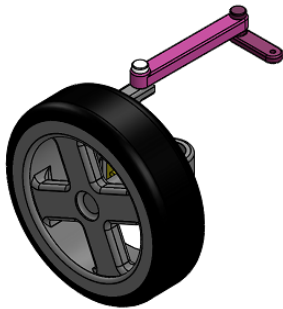


Figura 4. Sistema de direção projetado

2.3 Raspberry Pi

Segundo a *Raspberry Pi Foundation* (2021a), o *Raspberry Pi* é um computador de pequeno porte e que contém todas as funcionalidades de um computador normal, sendo muito utilizado para projetos de prototipagem e aplicações embarcadas. Neste projeto, utilizou-se o modelo 4 B com 4 GB de memória RAM.

Quanto ao software, optou-se por utilizar o Ubuntu Mate 20.04 como sistema operacional para uma melhor utilização do ROS Noetic. Isso faz com que seja mais fácil de se programar os códigos em outros computadores e passando apenas os dados necessários ao *Raspberry*, deixando a imagem do SO mais limpa.

Os dados coletados pelos sensores são transmitidos ao *Raspberry* através de comunicação serial pela porta USB e pelos *GPIO's* (General Purpose Input/Output). Além disso, o *Raspberry* também se comunica com o ESP32 pela comunicação serial, utilizando o pacote *rosserial* do ROS Noetic e com isso, fazendo o acionamento dos servomotores e dos motores CC pelas portas do ESP32.

O módulo de câmera *Raspicam* foi utilizado para que fosse possível gravar o ambiente em que o robô se locomove ao mesmo tempo que, caso necessário, possa ser acessado trazendo uma visão em primeira pessoa ao usuário. Conforme indicado pela *Raspberry Pi Foundation* (2021b), a câmera é capaz de gravar em alta resolução e pode ser controlada via software. Com isso, a câmera foi posicionada na face frontal do robô em um suporte que permite que seja possível girá-la manualmente caso seja necessário.

2.4 ESP32

Segundo o site da *Espressif Systems* (2022), o ESP32 é um microcontrolador de baixo custo e baixo consumo de energia que possui tanto Wi-Fi quanto Bluetooth sendo muito utilizado para projetos de sistemas embarcados que necessitem de uma placa menor e com mais pinos que o Arduino. Devido a grande complexidade do robô, optou-se pelo ESP principalmente pela quantidade de pinos PWM (*Pulse Width Modulation*) que ele pode fornecer além da possibilidade de transmissão de dados pela interface serial utilizando o pacote *rosserial*.

2.5 RPLidar A1M8

O sensor RPLidar A1M8 é um *laser scanner* de baixo custo produzido pela *Open Robotics* que permite a detecção de obstáculos e através de uma interface serial, permite a leitura dos dados de maneira mais simples. Além disso, como o sensor funciona em 360°, facilita o processo de mapeamento pois consegue ter um alcance bem maior do ambiente, sendo acoplado ao topo do robô.

2.6 Ultrassônico HC-SR04

O sensor ultrassônico HC-SR04 é um sensor de baixo custo muito utilizado em projetos simples para detecção de distância. Neste projeto, o sensor foi colocado nas faces frontal e traseira do robô como redundância para obstáculos que o Lidar não consiga detectar, evitando colisões.

2.7 MPU-6050

O sensor MPU-6050 é um módulo IMU (Inertial Measurement Unit) que permite identificar a posição e orientação do robô nos três eixos, permitindo então um melhor controle da dinâmica do mesmo, além de também poder ser conectado usando a interface I2C. O sensor foi posicionado na base do chassi e é conectado diretamente aos *GPIO's* do *Raspberry* evitando assim que haja alguma perda de informação devido as conexões.

2.8 Módulo GPS

Como o robô foi projetado para ser utilizado tanto em ambientes internos quanto externos, utilizou-se um módulo GPS Neo-6M, que permite obter os dados de latitude e longitude utilizando interface serial e podendo ser conectado diretamente ao *Raspberry*.

2.9 Baterias

A alimentação dos motores se dá através de duas baterias LiPo 18650 em série, no entanto como o Raspberry Pi necessita de valores de tensão e correntes bem diferentes, optou-se por se utilizar um Powerbank exclusivamente para isso.

2.10 Ponte H L298N

O módulo Ponte H *L298N* é um CI de ponte H que permite o controle de dois motores CC independentes ou um motor de passo. Sendo assim, optou-se por utilizar esse módulo para facilitar o controle de velocidade de cada uma das rodas do robô independentemente. Cada uma das entradas do módulo é conectada a um dos pinos do ESP que faz o controle do módulo.

2.11 Projeto da PCB

Para facilitar a ligação dos componentes, foi desenvolvida uma PCB que permite o encaixe do ESP32 e facilite a passagem de cabos por dentro do chassi. O projeto da placa foi feito utilizando o software *Proteus* e o design pode ser visto nas figuras 5, 6 e 7. A placa produzida pode ser vista na figura 8

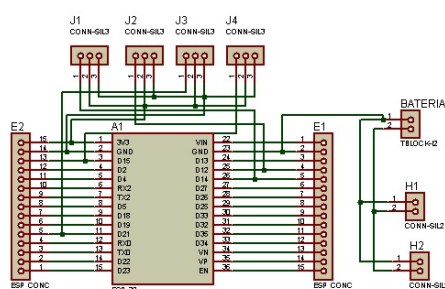


Figura 5. Esquema elétrico da placa projetada

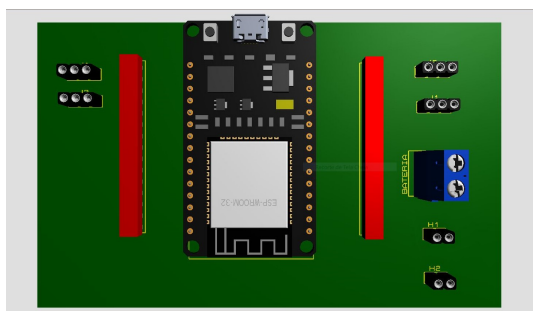


Figura 6. Face superior da placa projetada

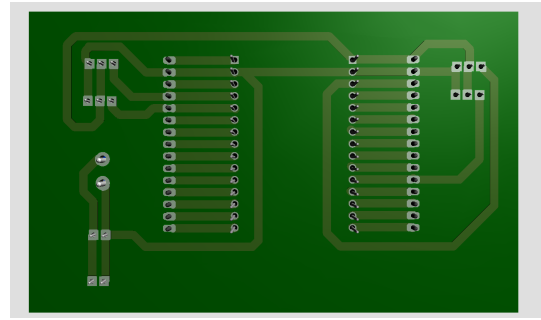


Figura 7. Face inferior da placa com as respectivas trilhas

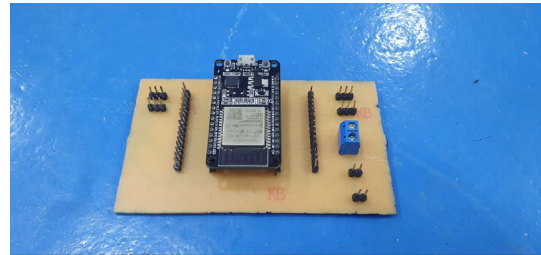


Figura 8. Placa finalizada e montada

2.12 Arquitetura dos componentes do Sistema

Para uma melhor visualização do sistema, temos o diagrama de blocos da figura 9 que indica as principais conexões da parte de hardware do robô, em que as setas vermelhas indicam a alimentação do dispositivo, as setas verdes indicam comunicação via GPIO e as setas azuis indicam comunicação serial.

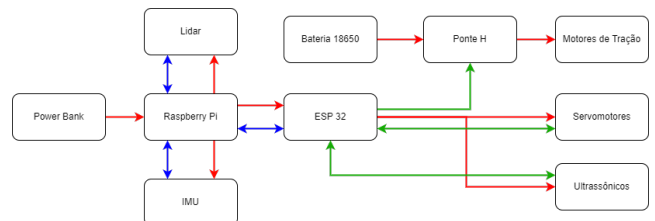


Figura 9. Arquitetura do Sistema

3. IMPRESSÃO DAS PEÇAS

De acordo com os websites da 3DLab (2019), Mecalux (2021) e Wishbox (2015), a manufatura aditiva pode ser descrita como o conjunto de tecnologias que utiliza desenhos assistidos por computador (CAD) e produz objetos 3D a partir de uma impressora 3D, por exemplo. Com ela, é possível produzir peças complexas otimizando recursos, sendo que cada tipo de impressão 3D tem uma forma característica de se moldar o objeto desejado.

Neste projeto, todas as peças que compõem a parte física do robô foram impressas usando filamentos de PLA e ABS, que são facilmente encontrados no mercado, e utilizando o software Simplify 3D. Devido às dimensões do chassi serem muito grandes, as peças da base e da tampa foram impressas separadamente. Optou-se também por imprimir as rodas do chassi para uma maior liberdade na construção da mesma. A quantidade de material de impressão de cada peça pode ser vista na tabela 1.

Tabela 1. Quantidade de material e tempo de impressão das peças do robô

Peça	Quant. Material	Tempo de Impressão
Tampa do Chassi	444,92g	1145 min
Base do Chassi	477,79g	1178 min
Roda	146,84g	352 min
Suporte do Motor	26,48g	83 min
”Braço”Longo	6,49g	18 min
”Braço”Curto	2,06g	8 min
Pinos de Fixação	0,57g	7 min
Encaixe dos Motores	34,16	105 min
Suporte Ultrassônico	10,47g	42 min
Suporte Câmera	14,07g	50min

Vale ressaltar que todos os valores presentes na tabela 1 são para produção individual, sendo assim, pode-se identificar através da tabela 2, a quantidade de cada peça produzida desconsiderando impressões que falharam por problemas de queda de energia, entupimento do bico da impressora, entre outros.

Tabela 2. Quantidade de peças produzidas

Peça	Quant. Peças Produzidas
Tampa do Chassi	1
Base do Chassi	1
Roda	1
Suporte do Motor	4
”Braço”Longo	4
”Braço”Curto	4
Pinos de Fixação	8
Encaixe dos Motores	4
Suporte Ultrassônico	2
Suporte Câmera	1

Diversas peças tiveram que ser impressas mais de uma vez pelos problemas citados anteriormente, no entanto as peças apresentaram uma resistência melhor que o esperado suportando diversos esforços mecânicos. Devido muitas peças serem de encaixe, tiveram que ser feitos diversos ajustes com ferramentas manuais a fim de deixá-las com uma pequena folga para evitar que o encaixe force demais as peças e acabe rompendo-as. Um exemplo de peça sendo impressa e que foi necessário o ajuste pode ser visto na figura 10

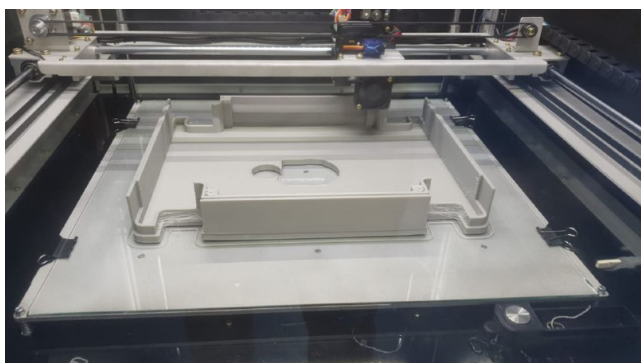


Figura 10. Tampa do chassi sendo impressa

4. MONTAGEM E CONSTRUÇÃO DO PROTÓTIPO

Com todas as peças do protótipo montadas, o robô começou a ser montado inicialmente ajustando a tampa e base do chassi para o encaixe, conforme figura 11.

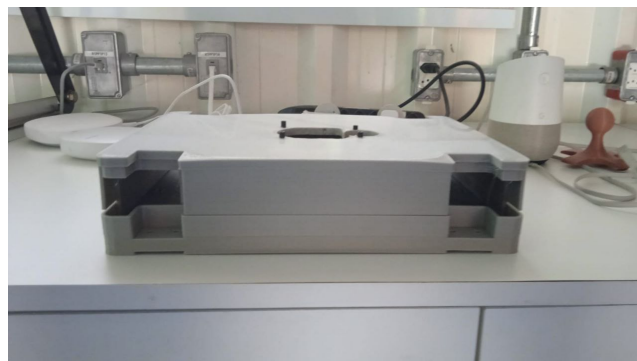


Figura 11. Montagem do chassi e ajuste do encaixe entre as peças

Com o chassi montado, partiu-se então para a montagem do sistema de direção, encaixando os braços nos espaços previamente determinados conforme o modelo 3D. Alguns ajustes tiveram que ser feitos nas peças para um melhor efeito dos ”braços” como uma fixação melhor para os pinos de encaixe e um ajuste no eixo da roda para melhor encaixe. O sistema montado pode ser visto na figura 12.

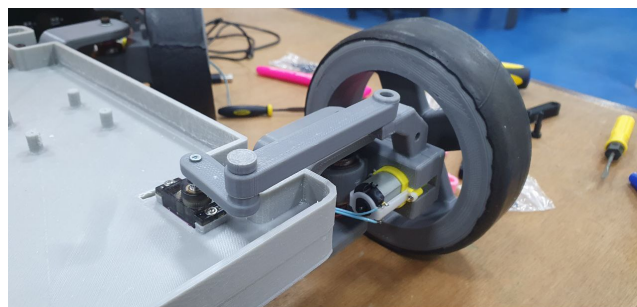


Figura 12. Montagem do sistema de direção das rodas

Como a roda também foi impressa, a superfície ficou lisa demais não conseguindo ter aderência ao solo. Sendo assim, utilizou-se uma borracha na superfície externa da roda para que fosse possível tracionar o robô no chão. Com isso, todas as peças estavam encaixadas e o robô se tornou funcional, conforme a figura 13.

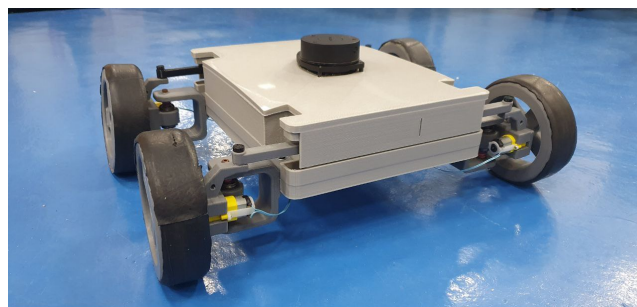


Figura 13. Robô montado com ajustes no encaixe

5. PROGRAMAÇÃO DOS DISPOSITIVOS

A parte de software do projeto foi dividida em três etapas: programação do ESP32 com as funções para movimentação das rodas e comunicação com o *Raspberry Pi*, programação do URDF (*Universal Robot Description Format*) do robô para identificação das posições e orientações de cada peça e por fim, os algoritmos de mapeamento e controle.

5.1 Programação do ESP32

Como neste projeto o ESP32 é utilizado apenas para controle dos motores e sensores conectados a ele, a sua programação pôde ser simplificada, utilizando apenas a Arduino IDE e bibliotecas próprias do ESP para os dispositivos utilizados.

Os testes com o ESP foram feitos de forma manual em que eram enviadas informações para acionamento de cada motor de forma a identificar possíveis erros ainda na fase inicial dos testes. Os códigos para acionamento automático dos motores serão feitos posteriormente ao final da programação dos algoritmos de mapeamento.

5.2 URDF do robô

Neste projeto, utilizou-se o plugin *Solidworks to URDF Exporter*, que facilita a exportação do código do URDF do robô a partir da modelagem CAD no Solidworks. No entanto, como o robô é de pequeno porte e as peças foram impressas, foram feitos alguns ajustes manualmente no código gerado.

O resultado final pode ser visto na figura 14, que mostra o protótipo no Rviz com os respectivos sistemas de coordenadas.

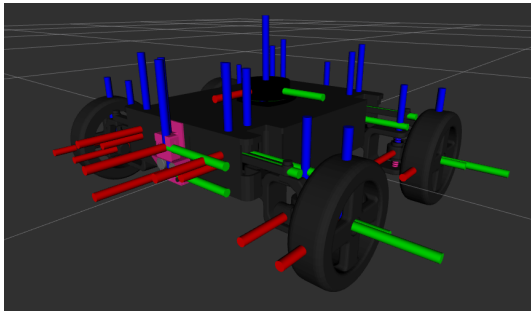


Figura 14. URDF com sistema de coordenadas

5.3 Algoritmos de Mapeamento

Os algoritmos de mapeamento implementados fez o uso de pacotes já existentes no universo do ROS. Durante a pesquisa, foi utilizado o pacote *Hector SLAM* que permite que o robô possa gerar o mapa através de um Lidar acoplado a ele, além de permitir que a odometria do robô seja feita utilizando o mesmo sensor.

Inicialmente, para delimitação da pesquisa, os testes foram realizados primeiramente em salas pequenas e com terreno regular para que depois o robô pudesse ser colocado em ambientes com superfícies irregulares. Por isso, optou-se por utilizar um salão do laboratório em que não houvessem tantos obstáculos e o ambiente tivesse uma forma mais quadrada como visto na planta baixa da figura 15.

O robô foi capaz de fazer o mapeamento da área levando em torno de 15 minutos para detecção de todas as paredes e identificação da rota em que podia percorrer. O acionamento do algoritmo de mapeamento junto com o de direção autônoma foi feito acessando o *Raspberry* via *ssh*.

Os resultados do primeiro teste podem ser vistos em 16. Nota-se que no canto inferior direito do mapa, há um

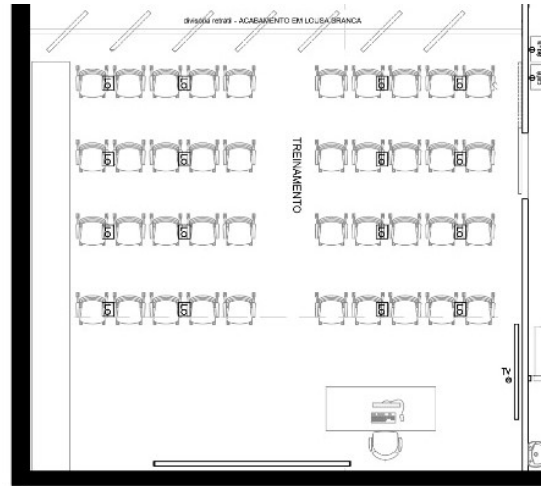


Figura 15. Trecho da planta baixa do salão mapeado

grande desnível em relação as paredes, que ocorre devido a estrutura de som presente no local, indicando que o mapa gerado apresentou bons resultados conforme visto na tabela 3, que compara a área do mapa com a área da planta baixa do local no Autocad indicando uma precisão de 91,68%.

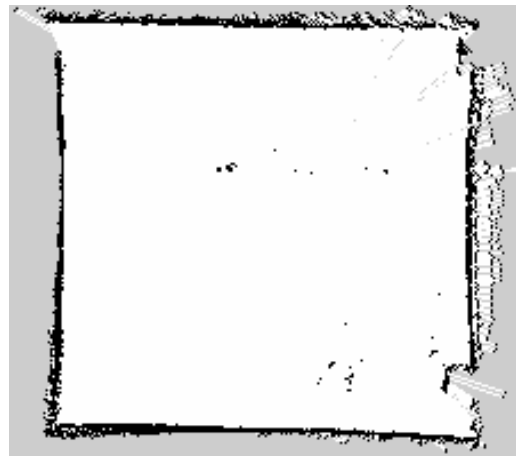


Figura 16. Mapa do laboratório gerado pelo robô

Tabela 3. Área calculada no *AutoCAD* do Auditório do Ocean

Imagem	Área (m ²)
Planta Baixa projetada	83,72
Mapa gerado pelo robô	76,7548

Um segundo teste foi realizado em uma loja de material esportivo, cujo terreno também é regular, no entanto a loja apresentava um balcão grande, que funcionaria como teste para o algoritmo de navegação autônoma. A planta baixa da loja pode ser vista na figura 17.

O resultado do mapeamento pode ser visto na figura 18. Nota-se que no parte superior direita, o robô não conseguiu detectar bem o espaço devido aos vários obstáculos na frente do sensor nessa área. Além disso, no canto inferior direito o robô detectou uma angulação, não conseguindo "fechar" o mapa. Isso ocorre por a porta da loja ser de vidro,

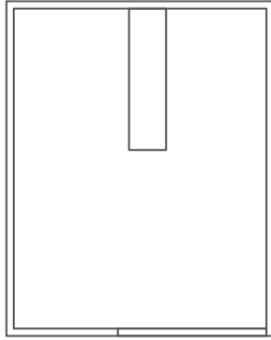


Figura 17. Planta baixa da loja mapeada

o que fez com que o sensor não conseguisse detectar com acurácia as bordas. Na tabela 4, pode ser visto o resultado da comparação entre a área do mapa com a área da planta baixa do local no Autocad, indicando uma precisão de 98,96%.

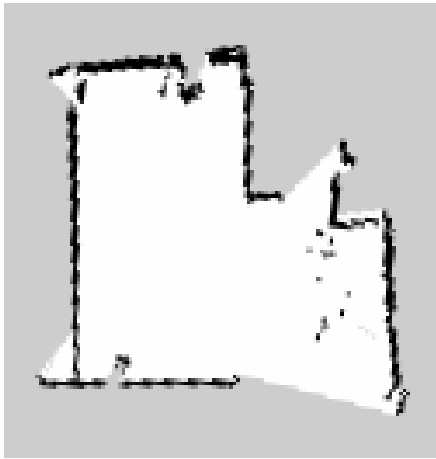


Figura 18. Mapa da loja gerado pelo robô

Tabela 4. Área calculada no *AutoCAD* da Loja mapeada

Imagem	Área (m^2)
Planta Baixa projetada	11,105
Mapa gerado pelo robô	10,9897

Por fim, para um local maior com maior quantidade de obstáculos, optou-se pelo Laboratório de Fabricação Digital dentro da Universidade, cuja planta baixa pode ser vista na figura 19.

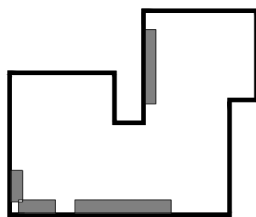


Figura 19. Planta baixa do laboratório

O resultado do mapeamento pode ser visto na figura 20. Nota-se que o robô conseguiu fazer um mapa muito

próximo ao visto na planta baixa projetada. No entanto no lado direito do mapa, pode-se notar que houver muitas falhas de detecção, que ocorreram devido a presença de alguns puffs que estavam posicionados no local. Por se tratar de uma área maior, o processo todo de mapeamento levou em torno de 20 minutos. O resultado comparativo entre o mapa e a planta baixa pode ser visto na tabela 5, indicando uma precisão de 90,38%.

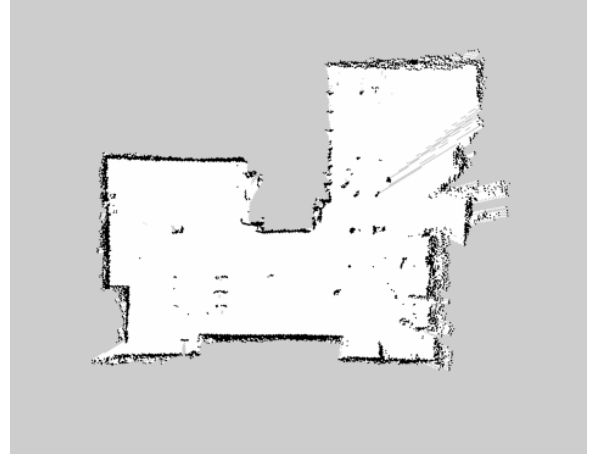


Figura 20. Mapa do laboratório gerado pelo robô

Tabela 5. Área calculada pelo *AutoCAD* do Laboratório de Fabricação Digital

Imagem	Área (m^2)
Planta Baixa projetada	97,8583
Mapa gerado pelo robô	88,4507

6. CONSIDERAÇÕES FINAIS

Este artigo apresentou o desenvolvimento de um robô autônomo projetado para locomoção em ambientes com terreno irregular e mapeamento do ambiente ao qual está inserido. Os testes de locomoção do robô em ambiente interno e controlado apresentaram um bom resultado, garantindo uma boa movimentação das rodas e do protótipo como um todo mesmo em terrenos levemente irregulares. Os testes de mapeamento em ambiente controlado e com terreno regular apresentaram um resultado satisfatório e como o robô ainda não foi testado em ambientes com terreno irregular, o robô ainda não foi finalizado, seguindo para testes.

No âmbito da pesquisa, pode-se citar o estímulo ao desenvolvimento de algoritmos e desenvolvimento de sistemas embarcados. Além disso, o projeto pode ser utilizado para divulgação da robótica nas universidades, a fim de buscar novos projetos tanto na parte de hardware quanto de software para novas soluções referentes a utilização de robôs em novas tarefas.

Como trabalhos futuros, será realizado o ajuste da programação dos dispositivos de forma a torná-lo totalmente funcional em ambientes com terreno irregular. Ao decorrer do processo de integração, novas funcionalidades podem ser acrescentadas e, ao final do processo, o robô será testado em ambiente externo para avaliar a eficácia do sistema de mapeamento do ambiente.

AGRADECIMENTOS

O presente artigo é decorrente do projeto de Pesquisa e Desenvolvimento (P&D) Projeto Samsung Ocean 2.0, que conta com financiamento da Samsung, usando recursos da Lei de Informática para a Amazônia Ocidental (Lei Federal nº 8.387/1991), estando sua divulgação de acordo com o previsto no artigo 39.º do Decreto nº 10.521/2020.

REFERÊNCIAS

- 3DLab (2019). Manufatura aditiva: saiba o que é e o que ela representa. <https://3dlab.com.br/>. Acessado em: 30 de novembro de 2021.
- Bouton, A., Grand, C., and Benamar, F. (2020). Design and control of a compliant wheel-on-leg rover which conforms to uneven terrain. *IEEE/ASME Transactions on Mechatronics*, 5(5), 2354–2363.
- de Sena Filho, E., da Terra C. Júnior, A.A., Cordeiro, I.G.D., de Souza, J.P.G., Diniz, M.B., Avila, J.A.C., Pádua, F.L.C., and Batista, N.C. (2020). Desenvolvimento de um robô autônomo em cenário educativo. *Anais do XXIII Congresso Brasileiro de Automática*, 2(1).
- Dudek, G. and Jenkin, M. (2010). *Principles of Mobile Robotics*. Cambridge University Press.
- Foundation, R.P. (2021a). Raspberry pi documentation. URL <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-4-model-b>. Acessado em: 27 de março 2022.
- Foundation, R.P. (2021b). Raspberry pi documentation. URL <https://www.raspberrypi.com/documentation/accessories/camera.html>. Acessado em: 27 de março 2022.
- Köseoğlu, M., Çelik, O.M., and Pektaş, (2017). Design of an autonomous mobile robot based on ros. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1–5.
- Matarić, M.J. (2007). *The Robotics Primer*. MIT Press.
- Mecalux (2021). Manufatura aditiva: a impressão 3d digitaliza a manufatura. <https://www.mecalux.com.br/blog/manufatura-aditiva>. Acessado em: 30 de novembro de 2021.
- Systems, E. (2022). Esp32 documentation. URL <https://www.espressif.com/en/products/socs/esp32>. Acessado em: 29 de março 2022.
- Wishbox (2015). Manufatura aditiva: entenda o que é e como ela funciona. <https://www.wishbox.net.br/blog/o-que-e-manufatura-aditiva/>. Acessado em: 30 de novembro de 2021.