



Otimização de software de navegação e mapeamento dedicada a robô Micromouse

Navigation and mapping software optimization dedicated to the Micromouse robot

Henrique Siqueira Camara de Souza¹, André Luiz Regis Monteiro², Lucas Ricken Garcia³, Marcio Rodrigues da Cunha⁴

RESUMO

O mercado agrícola vem tornando-se cada vez mais significativo devido ao acréscimo de novas tecnologias. A automação e a eletrônica embarcada tornam-se ainda mais presentes e necessárias. Hoje existem várias tecnologias que visam a automação, não só em plantio, mas também em carros, casas, iluminação pública. Isso indica uma área de pesquisa e atuação extremamente abrangente e com alto nível de conteúdos e estudos de caso. As áreas em que são realizados os plantios seguem padrões, ou seja, existem corredores entre as linhas de plantações para a passagem das máquinas, tornando-se um labirinto, propiciando assim corredores que podem ser mapeados. É possível, desta forma, que se aplique o uso de sensoriamento e algoritmos, eficazes e autônomos, de resolução de labirinto. Faz-se, assim o estudo da lógica de software e hardware para o desenvolvimento de e aplica-la em um protótipo. Embora as questões de economia da área agrícola considerem, principalmente, índices ligados à mão-de-obra e distribuição de alimentos, a contribuição desta pesquisa vem ao encontro de reduções econômicas, minimizando desperdícios através da inovação robótica.

PALAVRAS-CHAVE: Agricultura; Automação; Sensoriamento.

ABSTRACT

The agricultural market has become increasingly significant due to the addition of new technologies. Automation and embedded electronics become even more present and necessary. Today there are several technologies that aim at automation, not only in planting, but also in cars, houses, street lighting. This indicates an extremely comprehensive area of research and activity with a high level of content and case studies. The areas in which the plantings are carried out follow patterns, that is, there are corridors between the rows of plantations for the passage of the machines, becoming a labyrinth, thus providing corridors that can be mapped. It is possible, in this way, to apply the use of sensing and algorithms, effective and autonomous, of maze resolution. Thus, the study of software and hardware logic is done for the development of and apply it in a prototype. Although the issues of economics in the agricultural area consider, mainly, indices related to labor and food distribution, the contribution of this research meets economic reductions, minimizing waste through robotic innovation.

KEYWORDS: Agriculture; Automation; Sensing.

¹ Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: henriquesiqueira@alunos.utfpr.edu.br. ID Lattes: 2504062830178187.

² Docente no Curso de Engenharia Eletrônica. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: marciocunha@utfpr.edu.br. ID Lattes: 0516286832674015.

³ Docente no Curso de Engenharia Eletrônica. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: lucasgarcia@utfpr.edu.br. ID Lattes: 1085422040174691.

⁴ Docente no Curso de Engenharia Eletrônica. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: almonteiro@utfpr.edu.br. ID Lattes: 0961660781313311.

Introdução

Segundo Patrícia Knebel (2021), [8], o Brasil ampliou a produção de grãos em 500%, mas a área semeada aumentou apenas 70%. Segundo ela, essa grande produção se deve a inserção da automação e sensoriamento aliados a internet e a inteligência artificial.

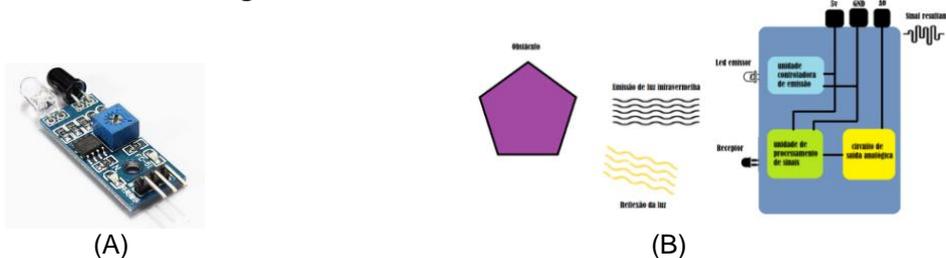
Neste contexto, o projeto proposto, que envolve hardware e software, aplica soluções de automação através de sensoriamento para mapear labirintos, que são os corredores entre as plantações, por onde passam as máquinas agrícolas. Em tal implementação utilizou-se, foi utilizado o microcontrolador ESP-32 junto a IDE Arduíno para programar e controlar os sensores e o chassi de testes convencional 2WD.

O propósito é realizar um labirinto com um algoritmo eficiente e autônomo detectando paredes, laterais e frontais, que corresponderiam às plantações e fim de área, respectivamente. Explorando assim a parte de automação das máquinas.

Metodologia

Para a localização das paredes torna-se necessário a utilização de métodos de hardware e software. Para hardware, utilizou-se o sensor infravermelho que tem como principal base teórica a absorção de luz.

Figura 1 – Sensor Reflexivo Infravermelho



Fonte: Eletrogate (2023), Danilo Almeida (2018)

A luz emitida pelo sensor da Figura 1 A é absorvida pelo objeto e também refletida, deste modo a mesma é recebida como é possível ver também na Figura 1 B.

Após este processo, são feitos cálculos e tratamentos no sinal que possibilitam a aferição de distância que é convertida em um sinal analógico, sendo está a ação do software. Porém para este projeto foi utilizado apenas o sinal digital, sendo a distância regulada pelo “trimpot” contido no circuito. A leitura digital foi implementada em código como apresentado a Figura 2.

Figura 2 – Leitura dos sensores

```
bool sensor_esquerdo = digitalRead(sensorEsquerdo);  
bool sensor_frente = digitalRead(sensorFrente);  
bool sensor_direito = digitalRead(sensorDireito);
```

Fonte: Autoria própria (2023)

Para tal leitura foram utilizadas variáveis booleanas, que retornam verdadeiro quando o sinal está em nível lógico alto e falso para nível lógico baixo, presença e não presença de parede respectivamente.

O robô conta com dois motores que são controlados com auxílio do driver ponte h L298n. Para isso foram criados 2 objetos da classe motor, sendo o motor esquerdo e o motor direito, com seus atributos de velocidade e direção. O código utilizado foi baseado no algoritmo feito por Gustavo Nery (2020), [5]. Tal algoritmo é aberto e está disponível para download no site da Eletrogate.



Após isso foram criadas 3 funções básicas, a “forward”, “backward” e “stop”. A junção destas permite a criação de outras, como avançar, virar à esquerda, virar à direita, ir para frente, virar 180°.

Para uma alta eficiência do algoritmo, utilizou-se a máquina de estados para direcionar o funcionamento. Desta forma não foi preciso controlar os motores utilizando o tempo como fator de controle, mas sim uma variação contínua de estados.

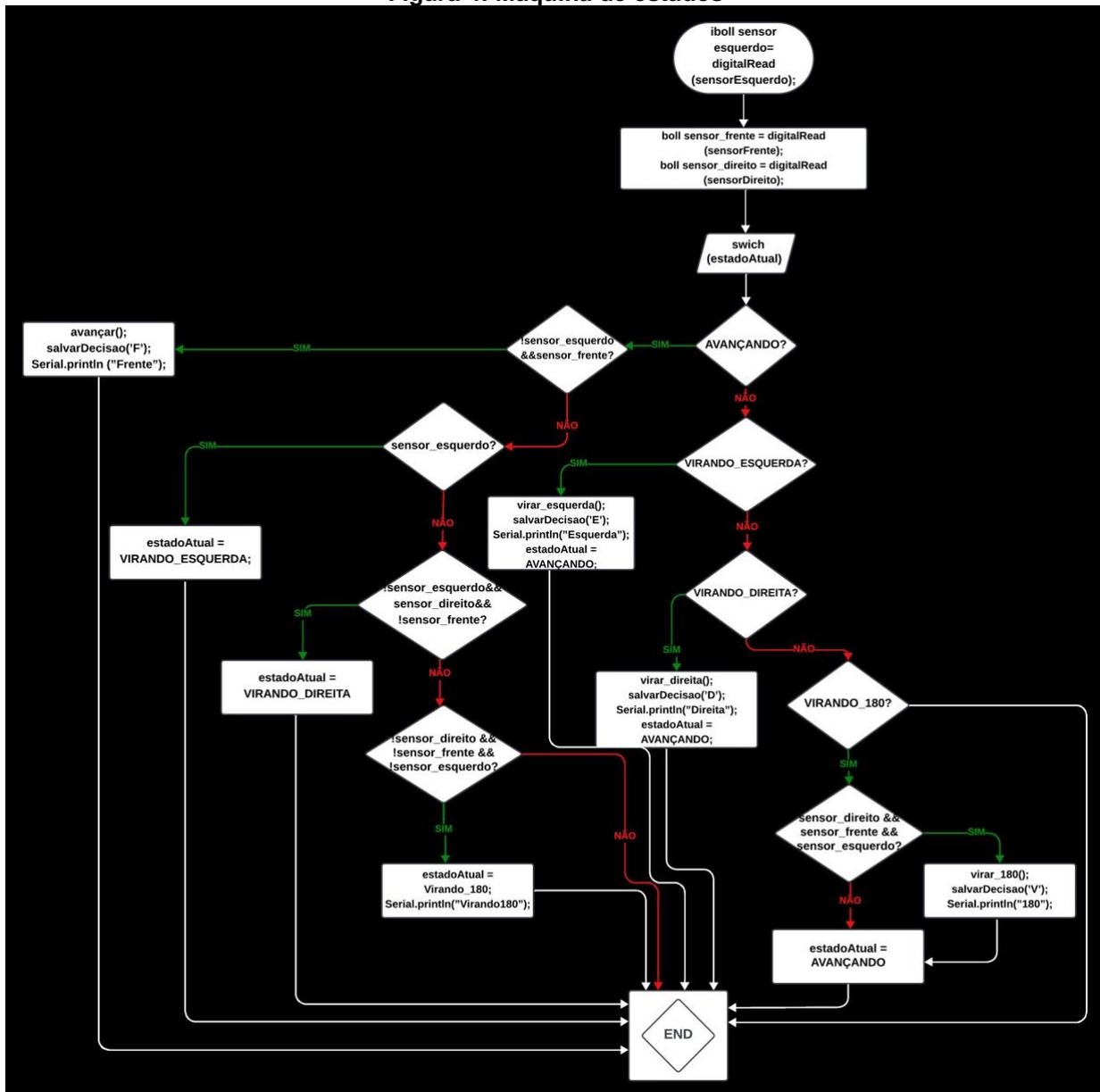
Figura 3: Parâmetros

```
enum Estado { AVANÇANDO, VIRANDO_ESQUERDA, VIRANDO_DIREITA, VIRANDO_180 };  
Estado estadoAtual = AVANÇANDO;
```

Fonte: Autoria própria (2023)

Com base na Figura 3, percebe-se que foi utilizado o tipo de dado “enum”, que define uma serie de valores para o conjunto estado. Define-se também que o estado atual do robô é avançando pois é sua primeira movimentação.

Figura 4: Máquina de estados



Fonte: Autoria própria (2023)

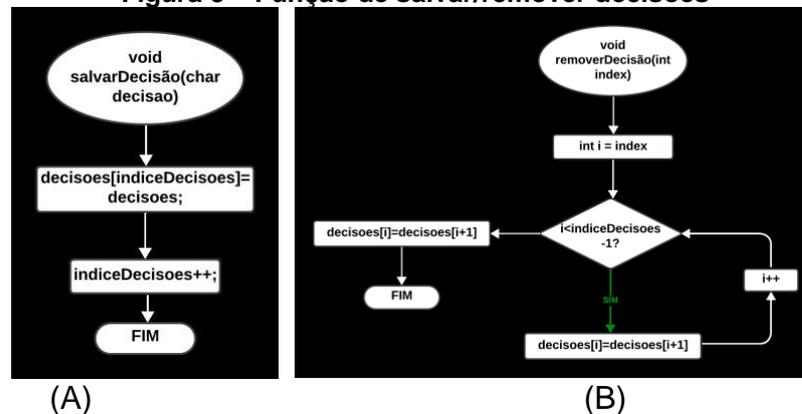


Para construir a máquina de estados é necessário o uso do “switch case”, que define um comportamento baseado nos parâmetros de entrada, que neste caso é o estado atual do protótipo. Para a resolução, utiliza-se um certo padrão, a “Regra da mão esquerda”.

O primeiro caso é “AVANCANDO”, o estado inicial, como pode ser visto na Figura 4, neste modo, é feita a verificação dos sensores para a tomada de decisão com a utilização de condicionais interligadas, ou seja, decide da maior para a menor prioridade. A decisão tomada é salva em um vetor para ser feita e devida otimização e também o estado atual é mudado. Com as mudanças de estado, o algoritmo segue pela mesa mapeando-a, nunca deixando de verificar os estados e as leituras dos sensores.

Para a otimização do caminho é necessário armazená-lo, para isso foi criado um vetor de caracteres que salva o caminho com sua inicial. Quando um dado é armazenado neste vetor, seu índice altera na própria função, como pode ser visto na Figura 5 A.

Figura 5 – Função de salvar/remover decisões



(A) Fonte: Autoria própria (2023)

Para serem feitas as devidas otimizações, torna-se necessário um algoritmo capaz de sobrescrever as decisões redundantes, tal algoritmo é apresentado na Figura 5 B.

Após serem salvos tais dados, apresenta-se o problema principal deste projeto, a otimização do caminho. A lógica para isso é que há decisões que podem ser eliminadas, como a de virar 180°, isso indica que o robô entrou em um caminho que não há saída, ou seja, ele não deve entrar nesta condição, é isso que a função apresentada na Figura 6 indica.

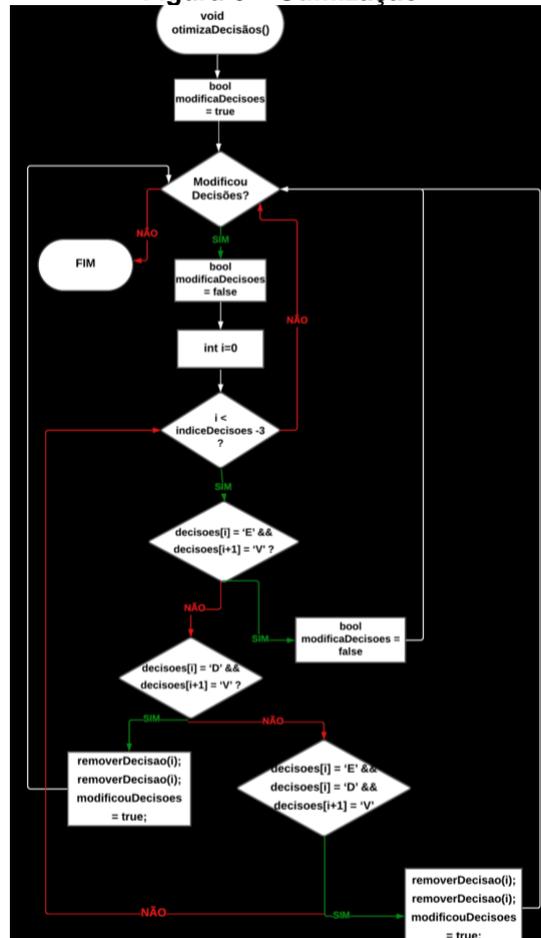
Incrementados tais algoritmos, foram realizados os testes de bancada. Primeiramente foi preciso verificar se a movimentação estava sendo realizada, com o robô sem contato com o solo, aferindo as condições manualmente.

Concluído que o mapeamento estava funcionando, foi preciso comprovar a eficácia do código de otimização, para isso foi utilizado um vetor de teste, sendo ele:

- char decisoes[8] = {'F','E','V','F','F','E','D','V'};



Figura 6 – Otimização



Fonte: Autoria própria (2023)

Resultados e Discussões

Os testes manuais e em solo demonstraram um bom resultado, mapeando de forma correta e concomitantemente se movendo conforme o algoritmo por máquina de estados explicita. Tais testes foram realizados cerca de 50 vezes, demonstrando 100% de assertividade. Para o teste de otimização de caminho, o vetor de teste foi submetido ao algoritmo, obtendo um resultado extremamente satisfatório, que é seguir apenas para frente, podendo ser comprovado na leitura do monitor serial na Figura 8.

Figura 8 – Vetor otimizado



Fonte: Autoria própria (2023)

Conclusão

Durante todo o processo foram encontrados problemas, que foram sendo solucionados, tais problemas trouxeram consigo uma grande evolução no algoritmo, o principal deles foi a solução por máquina de estados, não necessitando usar tempo para controlar o motor, tendo assim a máxima eficiência do hardware e um software bem condicionado.



Este projeto demonstrou de maneira prática a viabilidade do uso do modelo de máquina de estados para, por exemplo, uma máquina de plantio. Conta-se que o robô não foi testado de fato em um labirinto oficial, porém, todas as funcionalidades para fazer o mesmo foram testadas, condicionadas e concluídas. Para um futuro trabalho, existe a possibilidade de evolução, como alocação dinâmica de memória e trabalhar com matrizes para uma maior assertividade e exatidão.

Disponibilidade de código

Link para o código: https://github.com/HHSiqueira/Iniciacao_Cientifica

Conflito de interesse

Não há conflito de interesse.

REFERÊNCIAS

- [1] Bertoleti, Pedro. Máquina de Estado. **Embarcados - Sua Fonte de Informações Sobre Sistemas Embarcados**, 7 ago. 2015. Disponível em: <embarcados.com.br/maquina-de-estado/>. Acesso em: 10 de agosto de 2023.
- [2] CASAVELLA, Eduardo. Intelectual. **Linguagemc**, 2022. Disponível em: <linguagemc.com.br/enum-em-c/>. Acesso em: 20 de agosto de 2023.
- [3] De Oliveira Luz, Luiza; Rodrigo Cunha. Robótica Na Educação: Aplicação de Sensor Infravermelho Em Competições Robótica na Educação. **SEI – SICITE 2021**, Guarapuava, 2021.
- [4] Dias, Mateus. Pinagem ESP32: Saiba tudo sobre o ESP. **Lobo Da Robótica**, 11 de setembro de 2020. Disponível em: <lobodarobotica.com/blog/esp32-pinout/> Acesso em: 5 de março de 2023.
- [5] Nery, Gustavo. Guia Definitivo de Uso da Ponte H L298N. **Blog Eletrogate**, 5 de março de 2020. Disponível em: <blog.eletrogate.com/guia-definitivo-de-uso-da-ponte-h-l298n/>. Acesso em: 11 de março de 2023.
- [6] Souza, Fábio. Como programar o ESP32 no Arduino IDE? **Embarcados - Sua Fonte de Informações Sobre Sistemas Embarcados**, 28 de junho de 2022. Disponível em: <embarcados.com.br/como-programar-o-esp32-na-arduino-ide/>. Acesso em: 10 de março de 2023.
- [7] SCHNEIDER, Sergio *et al.* Os efeitos da pandemia da Covid-19 sobre o agronegócio e a alimentação. **Estudos Avançados** (online), São Paulo, v. 34, p. 167-188, 2020.
- [8] Knebel, Patricia. Mundo Conectado: Como a Internet das Coisas está revolucionando os negócios, as cidades e a vida das pessoas. Porto Alegre: Simplíssimo, 2021.