Ciência & Tecnologia
Fatec-JB

ISSN: 2178-9436

DESENVOLVIMENTO DE UM ROBÔ MÓVEL CONTROLADO POR ARDUINO NA CULTURA *MAKER*

DEVELOPMENT OF AN ARDUINO-CONTROLLED ROBOTIC CART IN MAKER CULTURE

Mateus Tronto [I]
Wilian Rafael Rodrigues de Oliveira [II]
Wagner da Silva Ribeiro [III]
Luis Carlos Geron [IV]
João Paulo Sachetto [V]

RESUMO

Este artigo tem como objetivo apresentar um projeto de desenvolvimento de um carrinho robótico. A robótica tem se mostrado uma ferramenta bastante versátil por sua multidisciplinaridade, uma vez que tem se tornado usual em diversas áreas, movimentando equipamentos variados. Apesar de espaço privilegiado nas indústrias, onde atua no chão de fábrica agilizando as atividades de produção, tem sido foco de muitas pesquisas, a fim de tornar sua atuação ainda mais eficaz. Nos espaços acadêmicos, a robótica tem cooperado na transposição de teorias para a prática, onde o estudante consegue experenciar a construção de robôs manipuladores ou móveis, como é o caso dessa pesquisa. Diante disso, foi construído em ambiente de aprendizagem um protótipo robô móvel, com componentes de fácil aquisição e o Ambiente de Desenvolvimento Integrado Arduino. O veículo foi equipado com motores, sensores e um microcontrolador ESP32 que permite a programação e controle eficaz do carrinho. O sistema foi concebido para demonstrar a integração de *hardware* e *software* em aplicações robóticas, proporcionando uma plataforma versátil e de baixo custo para aprendizado e experimentação robótica. Os resultados obtidos apontam a viabilidade e facilidade da programação de robôs móveis utilizando a plataforma ESP32 e IDE Arduino, aliando a disponibilidade de *hardware* de baixo custo, às quais possibilitam a aplicação e aprimoramento das habilidades em robótica.

Palavras-chave: arduino; controle; ESP32; programação, robótica.

ABSTRACT

This article aims to present a project for developing a robotic cart. Robotics has proven to be a highly versatile tool due to its multidisciplinary nature, becoming increasingly common in various fields, handling diverse equipment. Despite its prominent role in industries, where it operates on factory floors to streamline production activities, robotics remains a focus of extensive research to enhance its effectiveness further. In academic settings, robotics facilitates the transition from theory to practice, allowing students to engage in the construction of manipulator or mobile robots, as is the case with this research. Accordingly, a

[I] Fatec Sertãozinho; E-mail: mateus.tronto@fatec.sp.gov.br
[II] Fatec Sertãozinho; E-mail: wilian.oliveira4@fatec.sp.gov.br
[III] Fatec Sertãozinho; E-mail: wagner.ribeiro10@fatec.sp.gov.br
[IV] Fatec Sertãozinho; E-mail: luis.geron@fatec.sp.gov.br
[V] Fatec Sertãozinho; E-mail: joao.sachetto@fatec.sp.gov.br

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

prototype mobile robot was built in a learning environment using readily available components and the Arduino Integrated Development Environment (IDE). The vehicle was equipped with motors, sensors, and an ESP32 microcontroller enabling effective programming and control of the cart. The system was designed to demonstrate the integration of hardware and software in robotic applications, providing a versatile and cost-effective platform for robotics learning and experimentation. The results indicate the feasibility and ease of programming mobile robots using the ESP32 platform and Arduino IDE, leveraging the availability of low-cost hardware to apply and enhance robotics skills.

Keywords: arduino; control; ESP32; programming; robotics.

Data de submissão do artigo: 13/11/2024.
Data de aprovação do artigo: 29/11/2024.
DOI: 10.52138/citec.v16i1.406

1 INTRODUÇÃO

Os veículos robóticos, segundo Magrin *et al.* (2022) têm desempenhado um papel significativo na pesquisa e desenvolvimento de tecnologias autônomas. O uso de microcontroladores embarcados numa placa de prototipagem, com ambiente de desenvolvimento integrado, como Arduino, tem facilitado a criação e experimentação de robôs móveis. Desta forma, o objetivo geral deste estudo é apresentar a criação de um carrinho robótico que utiliza a plataforma Arduino para controle e programação.

Especificamente, este trabalho de conclusão de curso (TCC) investiga a criação e programação de um carrinho robótico utilizando a linguagem C++. O projeto abrange desde a seleção e montagem de componentes de *hardware,* como sensores e atuadores, até a implementação de algoritmos avançados para navegação autônoma e detecção de obstáculos. A escolha do C++ é justificada por sua eficiência e aplicabilidade em sistemas embarcados Magrin *et al.* (2022).

O trabalho consiste na apresentação dos fundamentos teóricos da robótica, tais como os princípios de cinemática e controle e descrição do processo de integração dos componentes de *hardware*. A seguir, é dada a ênfase na criação de algoritmos para processamento de dados dos sensores e controle dos atuadores. Finalmente, técnicas de processamento de sinais, filtragem de dados e algoritmos de navegação, como mapeamento e planejamento de trajetórias, são exploradas, além de algoritmos para detecção e evitação de obstáculos.

2 DESENVOLVIMENTO DE VEÍCULOS ROBÓTICOS

Essa seção destina-se à apresentação da revisão bibliográfica sobre os veículos robóticos.

2.1 Definindo robótica e conceitos pertinentes

O desenvolvimento de novas tecnologias tem colaborado para o avanço em todas as áreas, com experiências de sucesso e com menor custo do que em outros contextos históricos. Para Silva e Barbosa (2021), a robótica tem se apresentado como uma ferramenta multidisciplinar que favorece a aprendizagem em diversos campos de atuação, como o das engenharias, computação, biologia, entre outras.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

Apesar de muito presente no chão de fábrica, Sato, Santos e Almeida (2015), relatam que a robótica apresenta, ainda, problemas que impõem soluções imediatas e, por isso, tem sido alvo de muitas pesquisas, o que faz com que essa área tenha um desenvolvimento crescente, sendo utilizado atualmente em outros espaços que não o da indústria.

Em busca de maior autonomia e flexibilidade, esta ferramenta tem possibilitado integrar conhecimentos de diversas disciplinas, sendo por isso muito utilizada para que os alunos coloquem em prática teorias e conceitos aprendidos, desenvolvendo protótipos por meio do método científico (Resnick, 2020).

Sato, Santos e Almeida (2015) diferenciam os robôs móveis dos manipuladores, tendo os primeiros, capacidade de locomoção e complexidade mais elevada no âmbito da percepção, localização e navegação, que os robôs manipuladores.

2.2 Diferenças entre a plataforma arduino e ESP32: controle e programação

A plataforma Arduino foi criada no ano de 2005 pelo italiano Massimo Banzi e equipe para auxiliar estudantes de eletrônica. É uma plataforma de código aberto, baseada em *hardware* e *software* flexíveis, disponível para a comunidade, de baixo custo, possibilitando o desenvolvimento de protótipos de objetos ou ambientes de interação (Arduino, 2024).

Movido por sensores, pode interagir em seu entorno, sentindo o estado do ambiente, o que possibilita controlar as luzes, motores e outros atuadores. A linguagem de programação do microcontrolador tem como base a linguagem *Wiring,* dedicada ao controle do *hardware*, e o ambiente baseia-se no *Processing,* que é focado em desenvolvimento de gráficos, podendo desenvolver projetos autônomos ou que se comuniquem com um computador.

O componente *hardware* do Arduino tem uma placa de prototipagem que permite a construção de projetos. Existem vários modelos, como o Arduino Nano, Arduino Mega ADK e o Arduino Uno. A princípio, era a plataforma escolhida para o projeto, porém, algumas limitações obrigaram a observar o ESP32. A principal diferença entre o ESP32 e o Arduíno Uno é o desempenho e as funcionalidades integradas.

O ESP32 possui um processador *dual-core* de 32 bits que pode operar em até 240 MHz, enquanto o Arduino Uno tem um processador de 8 *bits* que chega a 16 MHz (ESPRESSIF, 2018). Ele tem 520kB de memória SRAM e 4 MB de memória *flash*, muito mais que os 2 kB de SRAM e 32 kB de *flash* do Arduino Uno. Possui, ainda, *Wi-fi* e *Bluetooth* integrados, enquanto o Arduino Uno não possui conectividade sem fio.

O ESP 32 possui uma gama mais ampla de periféricos, com 36 pinos GP10, 16 canais PWM, 12 canais ADC de 12 *bits*, interfaces UART, SPI, I2C, I2S e CAN (ESPRESSIF, 2018). Possui modos de baixo consumo de energia, como *deep sleep*, que o torna adequado para aplicações alimentadas por bateria (Miranda, 2019). Diante disso, o ESP32 se destaca por seu desempenho muito superior, mostrando-se mais adequado aos projetos loT e de automação que exigem conexão com a *Internet* ou dispositivos *Bluetooth.*

O microcontrolador ATMEL ATMEGA16U2 faz a interface USB para comunicação com o computador, sendo o principal componente, responsável por gerenciar os processos de comunicação e controlar os sinais analógicos e digitais nos pinos. Ele é um dispositivo de 8 *bits* da família AVR, com arquitetura RISC avançada e com encapsulamento DIP28 e dispõe de 32 kB de *flash*, sendo que 512 *Bytes* são utilizados para o *bootloader*. Tem 2 kB de RAM e 1kB de EEPROM. Os cristais utilizados nesse modelo possuem dimensões bem reduzidas (Arduino, 2024).

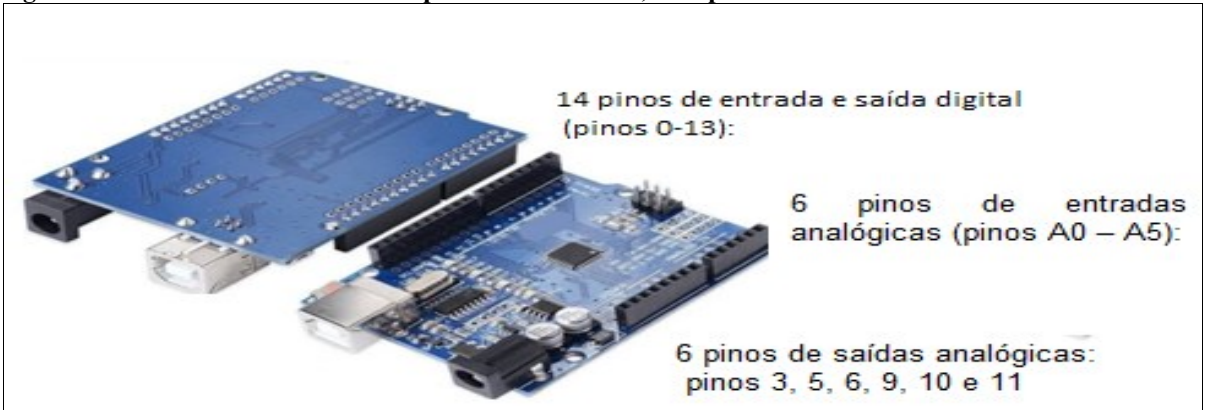
Segundo Miranda (2019), “os microcontroladores são normalmente utilizados em automação e controle de produtos e periféricos, como sistemas de controle de motores

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

automotivos, controles remotos, máquinas de escritório e residenciais, brinquedos, sistemas de supervisão etc.”.

A placa apresenta portas de pinagem enumeradas e identificadas, sendo A0, A1, portas analógicas, 1, 2, 3…, portas digitais, e portas que possuem o símbolo ~, que são portas que podem ser utilizadas, como PWM7 (Silva; Barbosa, 2021), conforme Figura 1

Figura 1 – Vista frontal e traseira da placa Arduino Uno, com pinos



Fonte: Silva e Barbosa (2021, p. 6)

A alimentação da placa ESP32 pode ser feita por USB A/B, conectado a qualquer dispositivo que tenha porta de comunicação USB, ou por fonte externa, como bateria conectada ao seu plug, com um valor de tensão entre 7V e 12V (ESPRESSIF, 2018).

No que se refere ao *software*, a programação e gravação do ESP32 é escrita em linguagem Java - orientada para objetos e disponível para uso em vários sistemas operacionais – e se baseia nas linguagens *Processing*, avr-gcc e outros de código aberto. O ambiente de programação apresenta muitos recursos, entre eles, *verify*, que verifica erros na programação; *upload*, que transfere o código para a placa; *new*, criador de janela em branco; *open*, abre arquivos; *save* que salva as alterações no código (ESPRESSIF, 2018).

Existe uma estrutura para programação específica para utilização em microcontroladores, no caso a linguagem *Wiring*, com exemplos já existentes no ambiente Arduino.

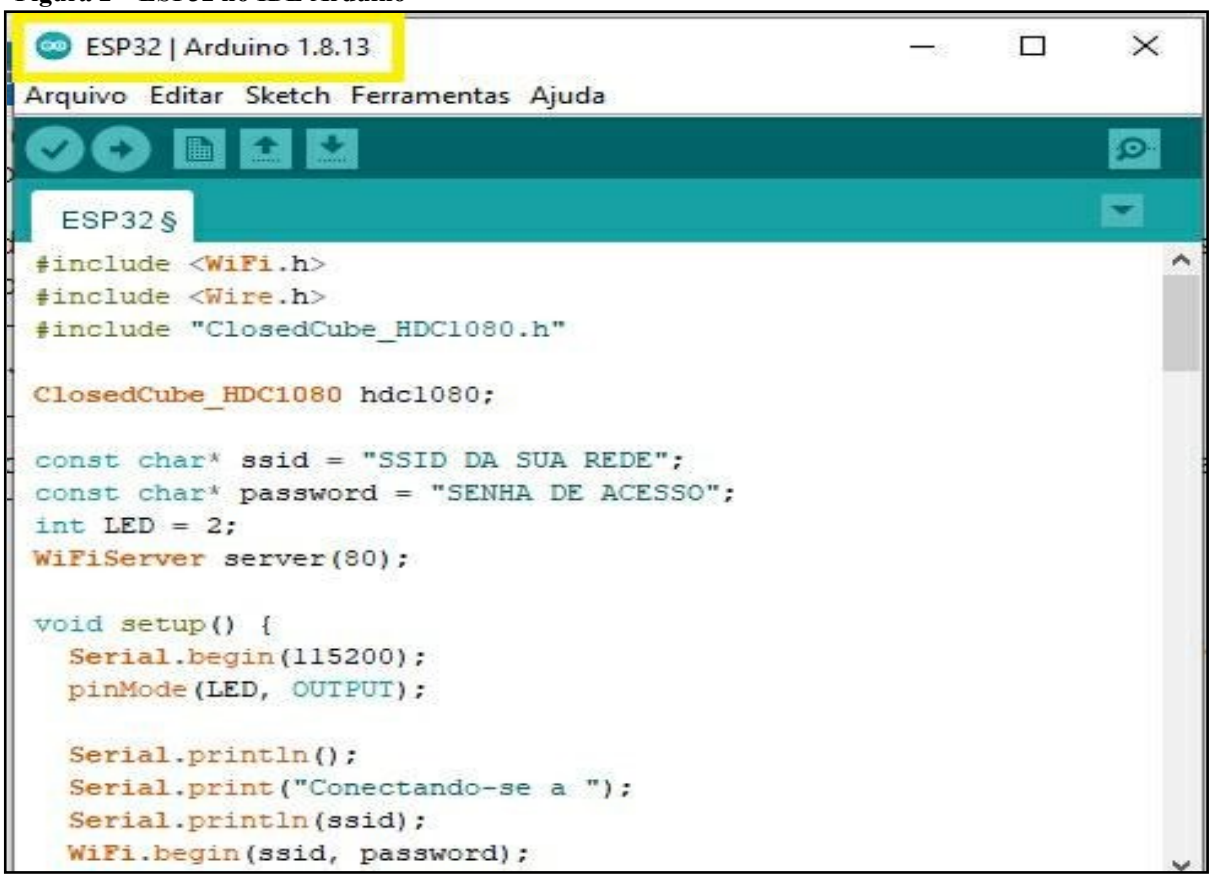
2.3 Desenvolvimento de produtos na IDE Arduino

A plataforma arduino possibilita o desenvolvimento de robôs móveis para várias finalidades. Por isso, é importante, de acordo com Magrin *et al.*(2022), que os estudantes possam desenvolver competências em tecnologia, matemática e áreas relacionadas que utiliza kits educacionais para a criação de protótipos com foco em resolver problemas do cotidiano. Desta forma, estará apto a criar carrinhos robóticos que possam auxiliar na execução de tarefas acadêmicas, domésticas ou profissionais.

No chão de fábrica, esses robôs têm um papel significativo para a manutenção da produção e a aprendizagem do processo, estimulante para profissionais voltados para a cultura *maker*. Resnick (2020) explica que o usuário do ESP32 pode utilizar o IDE Arduino para criação de produtos, programados em linguagem C ++ (Figura 2) e utilizando ainda algumas definições que já estão programadas na plataforma, sendo elas: *analogRead, digitalRead, analogWrite, digitalWrite, pinMode,* entre outras.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

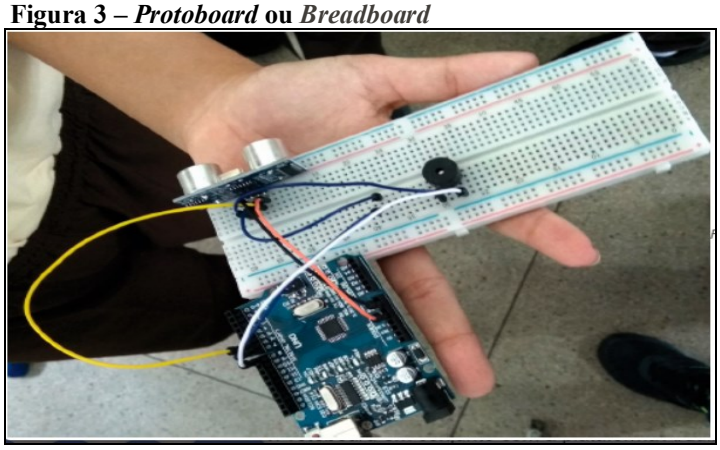
Figura 2 – ESP32 no IDE Arduino



Fonte: Neri (2014, p. 26)

A construção de robôs e autômatos, de acordo com Silva e Barbosa (2021), tem sido muito comum nas indústrias e ambientes automatizados. Para tanto, é necessário sensores, atuadores e peças eletrônicas para a montagem do *Protoboard* ou *Breadboard*, que é uma estrutura para montagem de protótipos, servindo para testes de prototipagem, permitindo conexões não definitivas (Figura 3).

Figura 3 – *Protoboard* ou *Breadboard*



Fonte: Silva e Barbosa (2021, p. 7)

A aprendizagem criativa é defendida por Resnick (2020) e o *protoboard*, junto ao ESP32, o IDE Arduino é o espaço em que isso ocorre, pois é propício para desenvolver as

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

habilidades de projeção, criação, experimento e inovação, já que possibilita a interdisciplinaridade de conhecimentos várias áreas de forma quase lúdica.

Em vista de outros kits robóticos, segundo Miranda (2019), o ESP32 pode se mostrar mais complexo em questões relativas à sua montagem e ao desenvolvimento de *software*, porém há facilitadores, como o programa *Scratch*, que utiliza blocos de encaixe parecidos com o LEGO para a construção de produtos.

3 METODOLOGIA

Para a construção do carrinho robótico, um planejamento foi realizado, resultando na lista de materiais mostrada na Tabela 1

Tabela 1 – Lista de Materiais do projeto

Item	Custo unitário	Quantidade	Custo total
Placa ESP32 30 pinos	49,90	1	49,90
Módulo Driver Ponte H	27,90	1	27,90
Kit Chassi + Motores DC	54,42	1	54,42
Sensor ultrassônico	15,99	1	15,99
Bateria	14,90	1	14,90
Protoboard	8,98	1	8,98
Cabos flexíveis	14,99	1	14,99
Total	187,08	7	187,08

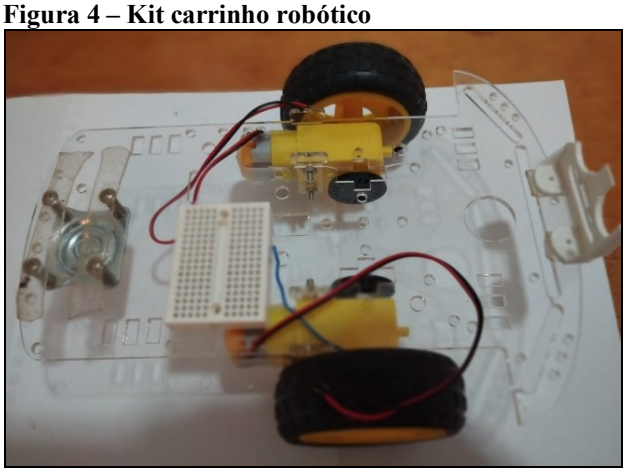
Fonte: autoria própria (2024)

Com os componentes já organizados, a equipe deu início à construção do carrinho robótico.

3.1 Procedimentos

Os componentes para a produção de um carrinho robótico foram recebidos, conforme Figura 4 a 8:

Figura 4 – Kit carrinho robótico



Fonte: Arquivo *image downloader* (2024)

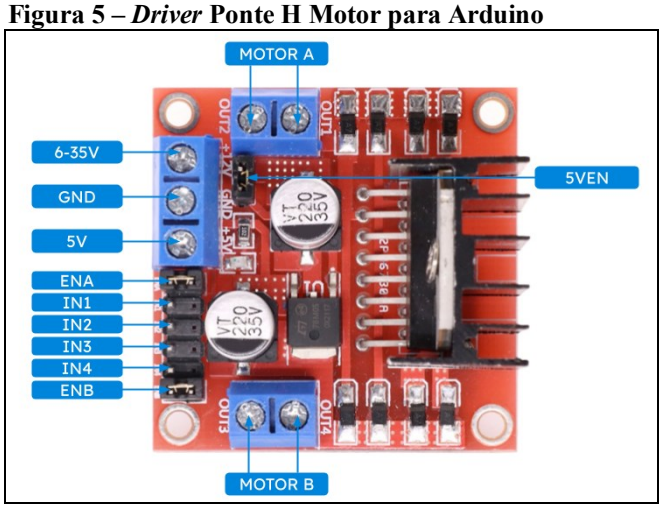
Utilizado como acessório de placas microcontroladoras como ESP32, o *Driver* Ponte H L298M (Figura 5) ajuda a controlar cargas indutivas, sendo possível controlar de modo

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

independente a velocidade e rotação de motores de passo. A ponte H L298N é um driver de motor comumente utilizado para controlar motores de corrente contínua (DC) e motores de passo em projetos de eletrônica e robótica, baseado no circuito integrado L298N (*datasheet*), que permite o controle simultâneo de dois motores DC ou de um motor de passo, possibilitando ajustes tanto na direção de rotação, quanto na velocidade.

O Motor A é ativado por meio dos pinos IN1 e IN2, enquanto o mesmo procedimento pode ser aplicado aos pinos IN3 e IN4, que controlam o Motor B. A alimentação é independente, reduzindo riscos de problemas elétricos, pois a demanda da corrente que ocorre pode ser igual ou maior que aquela permitida no do Arduíno. Além de permitir o acionamento dos motores, permite a variação das velocidades, dentro de um limite estipulado se ligado aos pinos PWM 2 do Arduino

Figura 5 – *Driver* Ponte H Motor para Arduino



Fonte: Arquivo *image Downloader* (2024)

O sensor ultrassônico (Figura 6) foi fixado na parte frontal do chassi. Ele foi utilizado por causa da sua atuação precisa e leituras estáveis, não sendo afetado pela luz. Possui um circuito pronto com emissor e receptor acoplados e 4 pinos (VCC, Trigger, ECHO, GND) para medição.

Figura 6 – Módulo ESP32 38 Pinos e Sensor ultrassônico

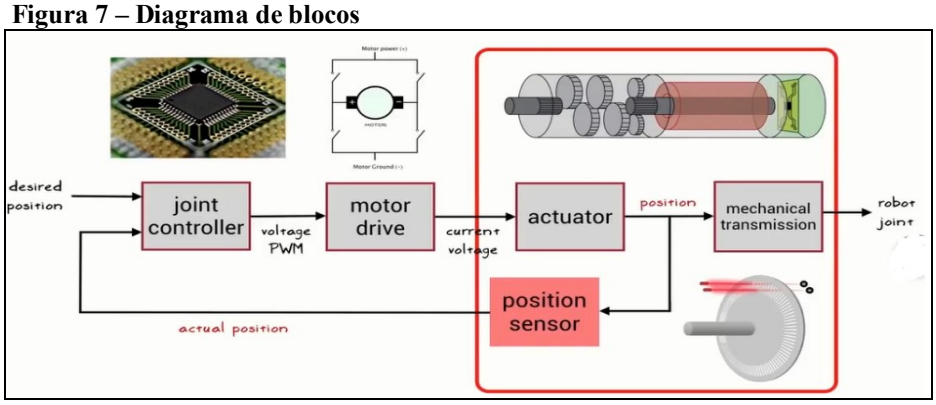


Fonte: Arquivo *image Downloader* (2024)

Já a placa de desenvolvimento *Wifi* e *Bluetooth* ESP32 38 Pinos (Figura 6) é uma plataforma robusta, ideal para aplicações de *Internet* das Coisas (IoT), com controle e monitoramento sem fio. A princípio, o robô seria autônomo, porém foi necessário utilizar um programa para seguir um caminho pré-determinado, no que o ESP32 foi muito útil (Figura 7).

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

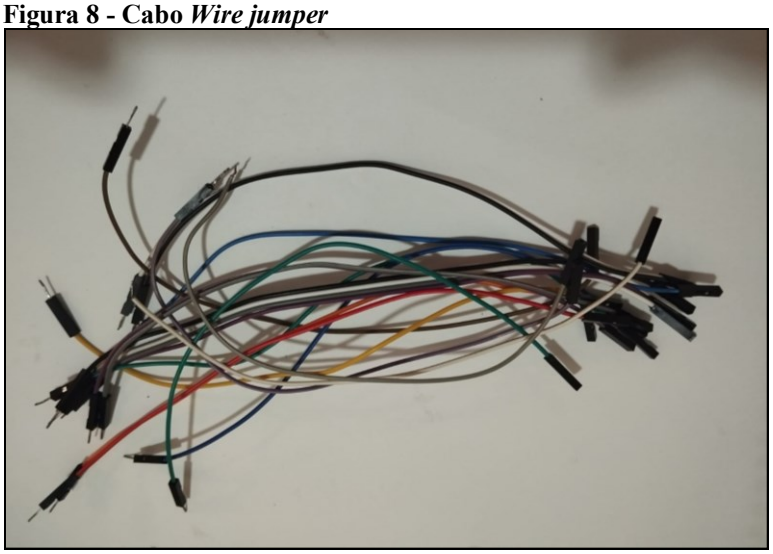
Figura 7 – Diagrama de blocos



Fonte: Arquivo *image Downloader* (2024)

O cabo *Wire Jumper* foi utilizado para fazer a ligação entre os sensores *bricks* e módulos externos às placas Arduino, apresentado na Figura 8.

Figura 8 - Cabo *Wire jumper*



Fonte: Arquivo *image Downloader* (2024)

O desenvolvimento da programação respeitou os seguintes passos:

Foi desenvolvida no ESP32 uma aplicação em C/C++, utilizando-se a IDE Arduino, que é um Ambiente de Desenvolvimento Integrado. Enquanto a linguagem C é procedural, ou seja, estruturada, a linguagem C++ é multiparadigmal, isso é orientada para objetos, possibilitando construir programas mais complexos e flexíveis a aproximando o manuseio de estruturas ao manuseio de objetos do mundo real.

A finalidade do programa (Figura 9) é controlar a movimentação do carrinho, fazer as leituras de distância do sensor ultrassônico e a detecção de objetos próximos correspondentes às informações obtida pelo sensor infravermelho.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

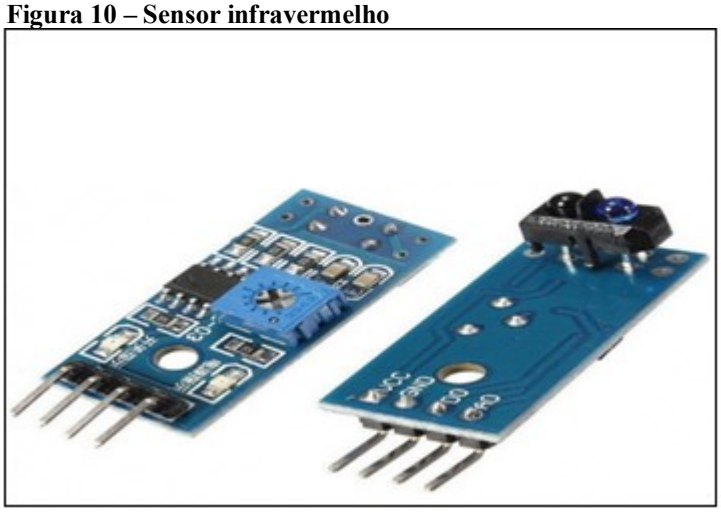
Figura 9 – Exemplo de linguagem C++ no Arduíno

```c
#include <stdio.h>
int main(){
    int a=8;
    float b=7.5;
    int resultado1;
    resultado1= a*a-b;
    printf("%d\n", resultado1);
    return 0;
}
```

Fonte: arquivo dos autores (2024)

O módulo sensor infravermelho Ir TCrt5000 Arduino *Raspberry* funciona com LED emissor Ir e um fototransístor Ir, lado a lado, detectando a presença de objetos (Figura 10).

Figura 10 – Sensor infravermelho

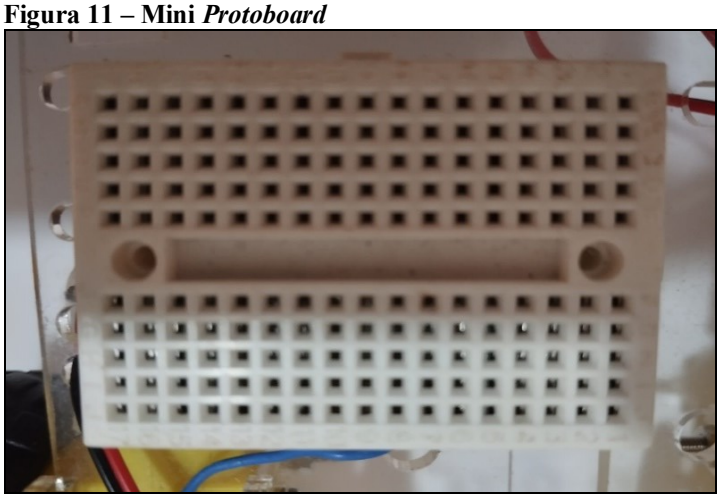


Fonte: Arquivo *image Downloader* (2023)

Foi utilizado um mini *protoboard breadboard* 170 pontos/furos/pinos ESP32 (Figura 11) porque permite conexões não definitivas, sem a necessidade de soldagem, apenas com encaixe das conexões condutivas, interligando os pontos desejados na placa, em linha horizontal.

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

Figura 11 – Mini *Protoboard*



Fonte: autoria própria (2024)

A montagem física ocorreu de forma tranquila, pois todos os componentes se encaixavam perfeitamente e o manual dispunha de explicação adequada. Na mecânica, o chassi do robô móvel foi montado com os motores e rodas acopladas, juntamente com o módulo driver de motor para alimentar os motores a partir do sinal de controle produzido na programação da velocidade e da direção.

Quanto à parte eletrônica, os dois motores de corrente contínua (Figura 12) foram fixados na base inferior da plataforma e quando acionados pelos sensores, que detectam os obstáculos perto do robô móvel, tomam a decisão necessária para evitar a colisão, executando movimentos de giro, para frente e para trás, e movimentos laterais (direita e esquerda).

Figura 12 – Montagem física do carrinho



Fonte: autoria própria (2024)

No que se refere ao sinal PWM, este controla a variação da tensão média aplicada ao motor que, por sua vez, controla a velocidade de rotação do robô, conforme o sentido da corrente elétrica. Os sensores demonstraram boa sensibilidade, pois eram acionados a uma distância de dois centímetros. Os comandos de controle do dispositivo e o recebimento de informações de medições dos sensores são enviados pelo módulo *bluetooth*, acionado por meio de *smartphone.*

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

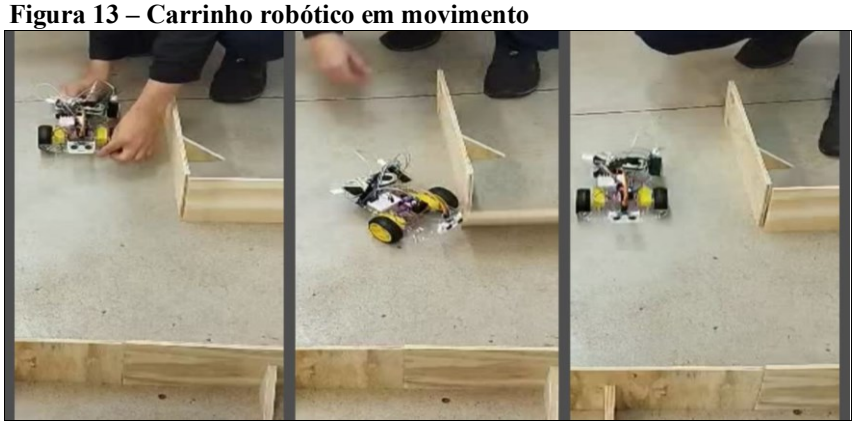
4 RESULTADOS

O teste prático ocorreu de forma satisfatória, pois esperava-se que apresentasse mais problemas para seu funcionamento, já que era a primeira experiência prática com robô móvel. No entanto, problemas como ordem de encaixe dos componentes, detecção dos sensores e leitura destes, ocorreram no decorrer do projeto e foram solucionados gradativamente, à medida que se criava uma intimidade com o IDE Arduíno.

Embora não tenha funcionado num primeiro momento, os ajustes necessários foram sendo previstos e realizados para que a movimentação do robô móvel ocorresse a contento. O robô controlado pelo ESP32 foi capaz de se movimentar em diferentes direções (Figura 13), evitando os obstáculos detectados pelo sensor ultrassônico.

O carrinho tem por condição andar em linha reta enquanto o sensor não detecta nenhum obstáculo. Quando isso ocorre, o primeiro passo, conforme programação do sensor, é dar um giro de 90° para a esquerda ou para a direita. Encontrando novo obstáculo, o carrinho girará 180° para a direita e, assim, consecutivamente.

Figura 13 – Carrinho robótico em movimento



Fonte: arquivo dos autores (2024)

A integração bem-sucedida entre *hardware* e *software* propiciou uma movimentação eficaz do veículo e possibilitou entender as muitas possibilidades desse incremento, uma vez que essa dinâmica pode ser empregada para criar outros objetos, não só como forma de aprendizado, mas de utilização pública.

5 CONCLUSÃO

Este estudo demonstrou a viabilidade da programação de um carrinho robótico utilizando a plataforma ESP32 e IDE Arduíno. A combinação de *hardware* e *software* de custo acessível e a flexibilidade de programação oferecida pelo Arduino possibilita a criação de projetos inovadores e aprimoramento das habilidades em robótica, com baixo custo e que pode ser facilmente reproduzido na elaboração de outros produtos.

Desta forma, pode-se concluir que os objetivos foram atingidos, uma vez que foi possível desenvolver um produto a partir da utilização da Plataforma de Aprendizagem Arduino e que se mostrou apto em todas as suas funcionalidades, propiciando aprendizagem e

Futuros trabalhos podem explorar a expansão das funcionalidades e a aplicação em contextos mais complexos, as possibilidades de alinhamento das aprendizagens teóricas às respectivas práticas e a viabilização do desenvolvimento de dispositivos robóticos funcionais. Além da conectividade apresentada nesse projeto e da integração de sensores, outras

Ciência & Tecnologia
Fatec-JB
ISSN: 2178-9436

funcionalidades podem ser acrescidas, tais como a integração de algoritmos de aprendizado de máquina para que o robô possa aprender a partir de experiências passadas e adaptar seu comportamento com base em dados coletados e projeção do robô com arquitetura modular o que permitirá a adição ou troca de componentes facilmente, possibilitando a experimentação e a personalização.

REFERÊNCIAS

ARDUINO**.** 2024. **O que é arduíno**. Disponível em: https://www.arduino.cc. Acesso em: 12 out. 2023.

ESPRESSIF. ESP32: **A diferente IoT and Power Performance**. 2018. Disponível em https://www.espressif.com/en/products/hardware/esp32/overview. Acesso em: 26 jun. 2024.

MAGRIN, C.E.; RIBAS, I.C. L.; RIBAS, T. A.; VITOLA, M. S.; TREAQUIN, P.; SURDI, J. J.; MAGRIN, C. S.; TODT, E. Promovendo a aprendizagem da robótica nas escolas com metodologias ativas e o desenvolvimento de um robô móvel acessível para redução das desigualdades sociais**. XIII Computer on the Beach**. Itajaí, SC, Brasil. v. 13, 2022. Disponível em: https://periodicos.univali.br/index.php/acotb/article/view/18811. Acesso em: 10 out. 2024.

MIRANDA, L. A. V. **Monitoramento de parâmetros ambientais de um leito hospitalar utilizando ESP32**. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2019. Disponível em: http://repositorioinstitucional.uea.edu.br/bitstream/riuea/3468/1/Monitoramento%20de%20par%C3%A2metros%20ambientais%20de%20um%20leito%20hospitalar%20utilizando%20ESP32.pdf. Acesso em: 28 jun. 2024.

NERI, H.G. F. **Utilização da plataforma Arduino para controle de experimentos remotos de física.** Trabalho de Conclusão de Curso. Universidade Federal de Uberlândia Instituto de Física. Campus Santa Mônica. Uberlândia, 2014. Disponível em: https://repositorio.ufu.br/bitstream/123456789/26520/1/UtilizacaoPlataformaArduino.pdf. Acesso em: 18 out. 2023.

RESNICK, M. **Jardim de Infância para a Vida Toda**: por uma aprendizagem criativa, mão na massa e relevante para todos. Tradução: Mariana Casetto Cruz e Lívia Rulli Sobral.Porto Alegre: Penso, 2020.

SATO, E. M.; SANTOS, L. A.; ALMEIDA, N. **Hivebots:** interação entre robôs para execução de tarefas. Curitiba: UTFPR, 2015. 15 p. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/200024/PFC%20Volnei%20Fontana%20Junior_2018-2.pdf?sequence=1&isAllowed=y. Acesso em: 19 out. 2023.

SILVA, M. P. DA.; BARBOSA, F. DA C. Matemática e Física em experiências de Robótica Livre: explorando o sensor ultrassônico. **Texto Livre**, v. 14, n. 3, p. e29629, 2021. Disponível em: https://www.scielo.br/j/tl/a/6kcn8mRzBtrpVM5F78kx6mD/?format=pdf. Acesso em: 18 out. 2023.