

ROBÓTICA EDUCACIONAL COM ARDUINO: FORMAÇÃO CIENTÍFICA E CIDADANIA DIGITAL NA EDUCAÇÃO BÁSICA

EDUCATIONAL ROBOTICS WITH ARDUINO: SCIENTIFIC TRAINING AND DIGITAL CITIZENSHIP IN BASIC EDUCATION

ROBÓTICA EDUCATIVA CON ARDUINO: FORMACIÓN CIENTÍFICA Y CIUDADANÍA DIGITAL EN LA EDUCACIÓN BÁSICA

Matheus Carvalho do Nascimento¹
Geórgia Regina Rodrigues Gomes²
Frederico José de Jesus Coelho³

RESUMO: Este artigo discute as potencialidades da robótica educacional na formação científica e na promoção da cidadania digital na Educação Básica, a partir do desenvolvimento de um carro-robô autônomo baseado na plataforma Arduino. O protótipo, montado sobre chassi Pololu e controlado por uma placa Arduino Romeo, utiliza sensor infravermelho analógico para desvio automatizado de obstáculos. Para além dos aspectos técnicos (microcontroladores, sensores, ponte H e programação em C++), o trabalho analisa as contribuições pedagógicas da atividade para o pensamento computacional, a resolução de problemas e o letramento em Inteligência Artificial. A fundamentação teórica baseia-se em diretrizes da UNESCO (2024) e do Ministério da Educação (BRASIL, 2026), dialogando com o construcionismo e a aprendizagem ativa. Conclui-se que projetos de robótica de baixo custo constituem espaços fundamentais para integrar conhecimentos de física, eletrônica, automação e informática a experiências formativas críticas, investigativas e socialmente contextualizadas.

1

Palavras-chave: Robótica educacional. Pensamento computacional. Arduino.

ABSTRACT: This article discusses the potentials of educational robotics in scientific training and the promotion of digital citizenship in Basic Education, based on the development of an autonomous robot-car based on the Arduino platform. The prototype, assembled on a Pololu chassis and controlled by an Arduino Romeo board, uses an analog infrared sensor for automated obstacle avoidance. Beyond the technical aspects (microcontrollers, sensors, H-bridge, and C++ programming), this study analyzes the pedagogical contributions of the activity toward computational thinking, problem-solving, and Artificial Intelligence literacy. The theoretical framework is based on guidelines from UNESCO (2024) and the Ministry of Education (BRASIL, 2026), dialoguing with constructionism and active learning. It concludes that low-cost robotics projects constitute fundamental spaces to integrate knowledge of physics, electronics, automation, and computer science into critical, investigative, and socially contextualized formative experiences.

Keywords: Educational robotics. Computational thinking. Arduino.

¹Mestre em Ensino pela Universidade Federal.

²Doutora em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio).

³Pós-graduado em tecnologia da informação pela Universidade FAVENI.

RESUMEN: Este artículo analiza las potencialidades de la robótica educativa en la formación científica y la promoción de la ciudadanía digital en la Educación Básica, a partir del desarrollo de un carro-robot autónomo basado en la plataforma Arduino. El prototipo, montado sobre un chasis Pololu y controlado por una placa Arduino Romeo, utiliza un sensor infrarrojo analógico para el desvío automatizado de obstáculos. Más allá de los aspectos técnicos (microcontroladores, sensores, puente H y programación en C++), el trabajo analiza las contribuciones pedagógicas de la actividad para el pensamiento computacional, la resolución de problemas y el alfabetismo en Inteligencia Artificial. La fundamentación teórica se basa en directrices de la UNESCO (2024) y del Ministerio de Educación (BRASIL, 2026), dialogando con el construccionismo y el aprendizaje activo. Se concluye que los proyectos de robótica de bajo costo constituyen espacios fundamentales para integrar conocimientos de física, electrónica, automatización e informática en experiencias formativas críticas, investigativas y socialmente contextualizadas.

Palabras clave: Robótica educativa. Pensamiento computacional. Arduino.

INTRODUÇÃO

A presença crescente das tecnologias digitais, dos sistemas algorítmicos e das aplicações de Inteligência Artificial nos diferentes setores da sociedade tem produzido transformações significativas nas formas de comunicação, trabalho, acesso à informação e produção do conhecimento. Nesse contexto, a escola passa a enfrentar o desafio de ressignificar suas práticas pedagógicas diante de um público cada vez mais conectado, dinâmico e inserido em uma cultura digital (Belloni, 2005; Lévy, 2005). Mais do que incorporar equipamentos tecnológicos ao espaço escolar, torna-se necessário desenvolver propostas educativas capazes de articular formação científica, pensamento crítico e compreensão sobre o funcionamento das tecnologias contemporâneas.

Entre as diferentes possibilidades de integração entre tecnologias e educação, a robótica educacional vem ganhando destaque por seu potencial interdisciplinar e formativo. Trata-se de uma área que articula conhecimentos de mecânica, eletrônica, automação e computação, possibilitando a construção de experiências práticas voltadas à resolução de problemas, à experimentação e ao desenvolvimento do pensamento computacional. Conforme argumenta Valente (1993), o uso das tecnologias na educação não deve limitar-se à informatização de práticas tradicionais de ensino. Em oposição a modelos centrados apenas na transmissão de informações, atividades baseadas na criação, na investigação e na interação com objetos técnicos ampliam as possibilidades de aprendizagem significativa.

Nessa perspectiva, a robótica educacional aproxima-se de concepções construcionistas de aprendizagem, especialmente das contribuições de Papert (1985), ao favorecer situações em

que o estudante deixa de ocupar uma posição passiva e passa a atuar como sujeito ativo na elaboração de soluções, na formulação de hipóteses e na análise crítica dos resultados obtidos. Ao construir e programar sistemas automatizados, estudantes mobilizam conhecimentos diversos e estabelecem conexões entre diferentes áreas do saber, aproximando-se de uma lógica menos fragmentada do conhecimento e mais próxima das múltiplas conexões características da perspectiva rizomática discutida por Deleuze e Guattari (1995) e Gallo (1995).

No cenário contemporâneo, marcado pela expansão da Inteligência Artificial e pela crescente presença de sistemas automatizados no cotidiano social, a robótica educacional deixou de representar apenas uma atividade técnica ou instrumental. Documentos como o Guia para a IA Generativa na Educação e na Pesquisa (UNESCO, 2024), o Referencial para Desenvolvimento e Uso Responsáveis de Inteligência Artificial na Educação (BRASIL, 2026) e o documento orientador IA na Educação Básica (BRASIL, 2026) destacam a necessidade de uma formação que articule competências técnicas, éticas e sociais, possibilitando que crianças e jovens compreendam criticamente as tecnologias que estruturam a vida contemporânea.

Diante desse contexto, este artigo possui dois objetivos articulados. O primeiro, de natureza técnica, consiste em apresentar o processo de construção e programação de um carro-robô autônomo, desenvolvido com componentes acessíveis e utilizando a plataforma Arduino. O segundo, de natureza pedagógica, busca discutir as contribuições desse tipo de atividade para a formação científica, o desenvolvimento do pensamento computacional, o letramento em Inteligência Artificial e a promoção da cidadania digital na Educação Básica.

O texto está organizado em seis seções, além desta introdução. A Seção 2 discute a importância pedagógica da robótica educacional à luz dos documentos oficiais e do referencial teórico adotado. A Seção 3 apresenta os principais dispositivos eletrônicos utilizados no desenvolvimento do protótipo. A Seção 4 detalha a estrutura e a programação do robô. A Seção 5 apresenta o estudo de caso e os ajustes realizados durante os testes. Por fim, a Seção 6 sintetiza as considerações finais e aponta possibilidades para trabalhos futuros.

ROBÓTICA EDUCACIONAL, PENSAMENTO COMPUTACIONAL E CIDADANIA DIGITAL

A presença crescente de sistemas baseados em dados, algoritmos e Inteligência Artificial nos diferentes domínios da vida social tem reposicionado as discussões sobre os currículos da Educação Básica. Mais do que inserir ferramentas digitais nas escolas, torna-se necessário

desenvolver competências que permitam compreender criticamente o funcionamento dessas tecnologias, seus limites, suas implicações sociais e seus impactos sobre os modos contemporâneos de viver, trabalhar e produzir conhecimento (UNESCO, 2024; BRASIL, 2026).

Nesse contexto, o debate sobre tecnologias educacionais ultrapassa perspectivas meramente instrumentais. Conforme argumenta Valente (1993), o uso dos recursos digitais na educação não deve restringir-se à informatização de práticas tradicionais de ensino, nas quais a tecnologia apenas reproduz metodologias centradas na transmissão de informações. Ao contrário, seu potencial pedagógico amplia-se quando esta é utilizada como espaço de criação, experimentação, investigação e resolução de problemas.

O Guia para a IA Generativa na Educação e na Pesquisa (UNESCO, 2024) sustenta que o uso da Inteligência Artificial na educação deve ampliar as capacidades humanas, e não substituí-las. O documento propõe uma abordagem centrada no ser humano, baseada na cooperação entre sujeitos e sistemas inteligentes em contextos de aprendizagem e trabalho. Na mesma direção, o Referencial para Desenvolvimento e Uso Responsáveis de Inteligência Artificial na Educação (BRASIL, 2026) e o guia orientador IA na Educação Básica (BRASIL, 2026) defendem que o letramento em IA deve articular-se aos letramentos digital, midiático e computacional, integrando competências técnicas, éticas e críticas.

4

Entre as aprendizagens fundamentais para a era da IA destacadas pelo MEC (2026), sobressaem aquelas relacionadas à compreensão da linguagem computacional, ao desenvolvimento do raciocínio lógico e à capacidade de relacionar soluções técnicas a problemas humanos e sociais. Tais competências dialogam diretamente com propostas de robótica educacional, especialmente quando estas são estruturadas em torno de atividades práticas e investigativas.

Sob essa perspectiva, projetos de robótica educacional como o aqui descrito ganham relevância por mobilizarem múltiplas dimensões do conhecimento. Ao construir, programar e testar um carro-robô autônomo, o estudante é inserido em uma situação concreta de resolução de problemas, na qual diferentes áreas do saber passam a se articular simultaneamente. Tal dinâmica aproxima-se de uma lógica rizomática do conhecimento (Deleuze; Guattari, 1995; Gallo, 1995), ao favorecer conexões entre conceitos de programação, eletrônica, física, automação e modelagem computacional.

Do ponto de vista pedagógico, atividades dessa natureza também dialogam com perspectivas construcionistas de aprendizagem. Conforme Papert (1985), a construção do

conhecimento ocorre de maneira mais significativa quando o estudante participa ativamente da elaboração de objetos concretos e da experimentação prática. Nesse processo, o erro deixa de representar apenas falha e passa a integrar o próprio movimento investigativo, contribuindo para a formulação de hipóteses, para a revisão de estratégias e para o desenvolvimento da autonomia intelectual.

Ao longo da construção do protótipo descrito neste trabalho, diferentes competências foram mobilizadas simultaneamente:

desenvolvimento do pensamento computacional, por meio da decomposição de problemas em sub-rotinas e algoritmos implementáveis em linguagem C++;

aplicação de conhecimentos de física e eletrônica relacionados a tensão, corrente, atrito, velocidade e controle de motores;

modelagem e abstração computacional, ao representar comportamentos esperados do sistema em termos de condições, estados e respostas;

desenvolvimento de atitudes investigativas e críticas, associadas à análise de erros, validação empírica e reformulação de soluções.

Além disso, práticas de robótica educacional também se aproximam das estratégias de IA desplugada propostas pelo MEC (2026) para contextos escolares com infraestrutura tecnológica limitada. Mesmo na ausência de plataformas avançadas de Inteligência Artificial, torna-se possível trabalhar conceitos relacionados a sensores, dados, automação, regras de decisão e sistemas inteligentes por meio de experiências físicas e tangíveis. Isso contribui para que estudantes compreendam as tecnologias digitais não como “caixas-pretas” inacessíveis, mas como construções sociotécnicas produzidas historicamente.

Nesse sentido, a robótica educacional pode contribuir diretamente para o desenvolvimento do letramento em IA, entendido como um conjunto de conhecimentos, habilidades e atitudes necessários para compreender, utilizar e avaliar criticamente sistemas inteligentes (UNESCO, 2024; BRASIL, 2026). Mais do que formar usuários de tecnologias, trata-se de possibilitar que estudantes compreendam os processos, limitações e implicações éticas envolvidos nos sistemas automatizados presentes na sociedade contemporânea.

Por fim, destaca-se a dimensão de cidadania digital envolvida nessas práticas. Ao projetar sistemas capazes de perceber o ambiente e agir sobre ele, estudantes podem ser levados a refletir sobre questões que ultrapassam o plano estritamente técnico: quais decisões automatizadas são aceitáveis? Quais limites éticos devem orientar sistemas inteligentes? Quem

se responsabiliza pelos impactos e falhas dessas tecnologias? Tais questões ocupam posição central nos debates contemporâneos sobre ética da IA, direitos digitais e formação cidadã (BRASIL, 2026).

Assim compreendida, a robótica educacional não se reduz a uma estratégia de motivação para áreas STEM. Ela constitui uma possibilidade concreta de articulação entre educação científica, cultura digital, pensamento crítico e formação cidadã em uma sociedade cada vez mais mediada por tecnologias inteligentes.

3. DISPOSITIVOS ELETRÔNICOS

Esta seção apresenta os principais componentes utilizados na construção do protótipo. Os conceitos foram organizados de modo a facilitar a compreensão de seu funcionamento e de suas possibilidades de aplicação em contextos de robótica educacional.

3.1 Arduino

O Arduino é uma plataforma de computação física open source, composta por hardware e software, voltada à prototipagem rápida de projetos eletrônicos. Sua placa funciona como um pequeno computador embarcado, capaz de habilitar e desabilitar entradas e saídas conectadas a dispositivos como LEDs, motores, sensores, módulos de comunicação e displays. A programação é realizada em uma IDE (Integrated Development Environment) específica, que utiliza linguagem baseada em C++ (McRoberts, 2011; Banzi, 2011).

Após a edição, o código é compilado e enviado (upload) à placa, que passa a executar as instruções de forma autônoma. Esse modelo permite ao Arduino atuar como elemento controlador do sistema, analisando continuamente os valores obtidos em suas entradas e executando ações em suas saídas conforme a lógica definida no programa embarcado.

As placas Arduino, em geral, integram microcontroladores ATMEL AVR, osciladores responsáveis pela geração de pulsos de frequência específica e reguladores lineares de tensão. Algumas variantes incluem ainda portas USB, entradas e saídas adicionais e shields especializados.

Neste projeto utilizou-se a placa Arduino Romeo, escolhida por integrar recursos de controle lógico, acionamento de motores e interfaces de comunicação em uma única estrutura, característica que favorece aplicações de robótica educacional em contextos de baixo custo e relativa simplicidade de montagem.

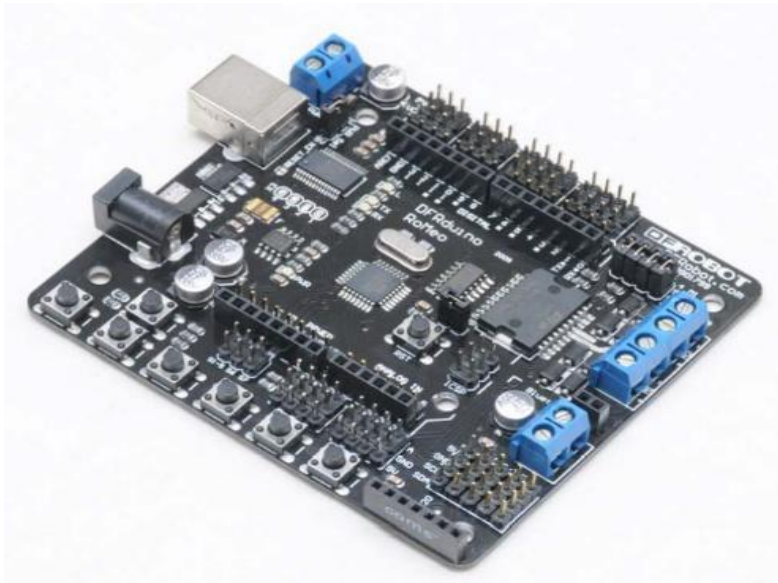


Figura 1: Placa Arduino Romeo.

Fonte: DFRobot.

3.2 Características da placa Romeo

A placa Romeo utilizada apresenta, entre outras, as seguintes especificações técnicas:

Microcontrolador ATmega328;

14 entradas e saídas digitais (6 com operação PWM);

8 canais de entrada/saída analógica com conversor de 10 bits;

Interface USB;

Pinagem ICSP para gravação de programas;

Suporte AREF (referência de tensão para entradas analógicas);

Soquete integrado para módulos APC220 (wireless) e Bluetooth;

5 pinos de interface I²C;

Ponte H integrada para controle bidirecional de até dois motores de 2A;

7 botões de uso geral;

Alimentação via USB ou fonte externa de 7 a 12V;

Dimensões aproximadas de 90 × 80 mm e peso de cerca de 60g.

A integração desses recursos em uma única placa reduz a complexidade de montagem do protótipo e amplia suas possibilidades de utilização em atividades pedagógicas voltadas ao ensino de programação, automação e sistemas embarcados.

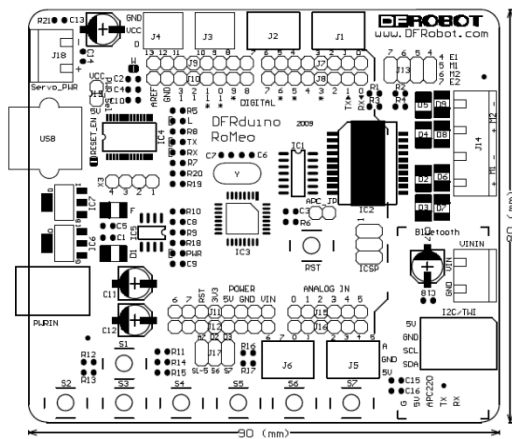


Figura 2: Disposição dos componentes na placa Romeo.

Fonte: dfrobot.com.

3.3 Sensores

Sensores são dispositivos capazes de detectar fenômenos físicos, como luz, temperatura, pressão, distância e cor, convertendo essas informações em sinais elétricos interpretáveis por sistemas de controle (Silveira; Santos, 2002). Quando o fenômeno observado não possui natureza elétrica, utiliza-se um transdutor, responsável pela conversão da grandeza física em sinal elétrico.

Nos sistemas de automação, sensores atuam em conjunto com atuadores, dispositivos responsáveis por executar ações a partir das leituras realizadas. No presente projeto, os motores desempenham o papel de atuadores, alterando seu comportamento conforme os valores identificados pelo sensor frontal.

Sensores digitais operam com níveis discretos de tensão, representados por estados binários (High e Low). Sensores analógicos, por sua vez, podem assumir múltiplos valores intermediários dentro de uma faixa contínua de tensão (Maxwell Bohr, 2012).

Para o desenvolvimento do protótipo, optou-se pelo sensor analógico infravermelho Sharp GP2Y0A21YKoF, capaz de detectar objetos entre aproximadamente 10 e 80 cm de distância. O componente opera com tensão entre 4,5V e 5,5V e apresenta tempo médio de resposta de ± 10 ms. Sua curva característica relaciona a tensão de saída à distância do objeto detectado.

A escolha desse sensor ocorreu em razão de sua compatibilidade com a plataforma Arduino e de sua aplicabilidade em projetos educacionais de automação e robótica de baixo custo.



Figura 3: Sensor analógico de distância Sharp GP2Y0A21YK0F.

Fonte: robocore.net.

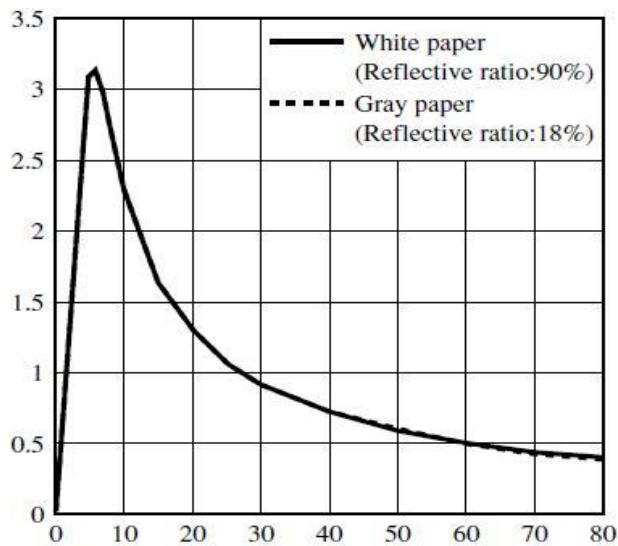


Figura 4: Curva distância × tensão do sensor Sharp.

Fonte: pololu.com.

3.4 Ponte H e chip L298

A ponte H é um circuito eletrônico utilizado para inverter a polaridade da alimentação de motores, permitindo sua rotação em sentidos opostos. Seu nome decorre da disposição de seus componentes, cuja estrutura se assemelha à letra “H”. O acionamento alternado das chaves eletrônicas permite controlar o sentido de rotação do motor sem necessidade de alterações físicas nas conexões.

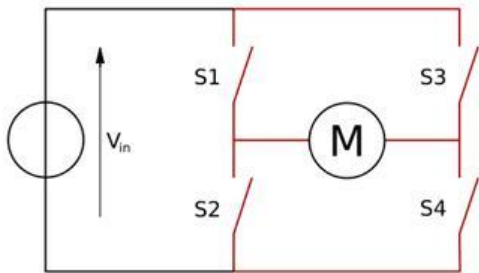


Figura 5: Topologia da ponte H.

Fonte: mecatrons.xpg.com.br.

Na placa Romeo, essa função é desempenhada pelo circuito integrado L298, um chip monolítico capaz de operar com correntes contínuas de até 4A e tensões elevadas, reduzidas neste projeto pelas limitações da plataforma Arduino. O componente aceita níveis lógicos padrão TTL e permite o acionamento de cargas indutivas, como motores CC, relés e solenoides (STMicroelectronics, 2012).

A utilização da ponte H mostrou-se fundamental para a implementação das rotinas de movimentação do robô, possibilitando o controle bidirecional dos motores por meio da programação embarcada.

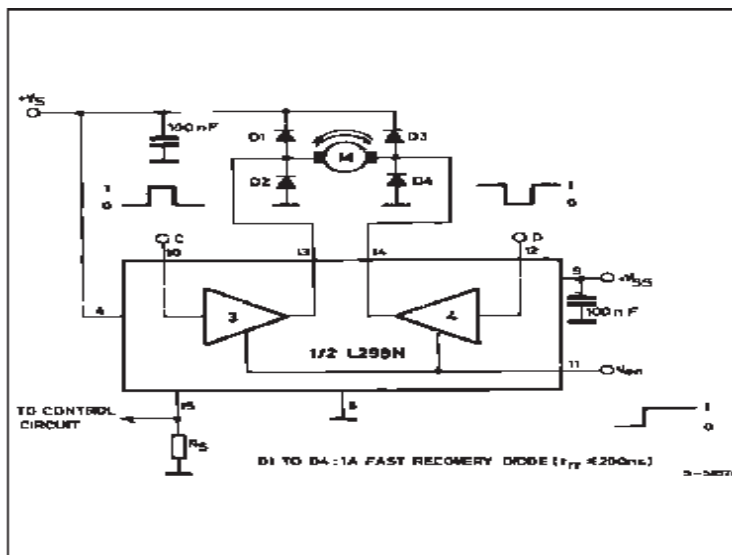


Figura 6: Diagrama de controle bidirecional do motor com o L298.

Fonte: alldatasheet.com.

3.5 Microcontroladores e o ATmega 328

Microcontroladores são circuitos integrados que reúnem, em um único chip, CPU, memória e circuitos de entrada e saída, funcionando como pequenos computadores dedicados. Diferentemente dos computadores de uso geral, esses dispositivos são projetados para executar tarefas específicas com baixo consumo energético e reduzido custo operacional (HowStuffWorks, 2012).

O ATmega 328, núcleo da placa Romeo, é um microcontrolador CMOS de 8 bits baseado em arquitetura RISC, composto por 32 registradores de propósito geral conectados diretamente à unidade lógica e aritmética (ULA). Seu desempenho aproxima-se de 1 MIPS por MHz, oferecendo equilíbrio entre capacidade de processamento e eficiência energética.

Sua ampla utilização em projetos educacionais e de prototipagem rápida decorre da simplicidade de programação, da estabilidade operacional e da compatibilidade com diferentes sensores e módulos eletrônicos.

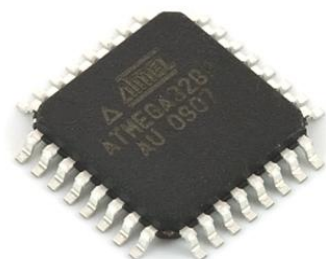


Figura 7: Microcontrolador ATmega 328.

Fonte: hobbytronics.co.uk.

3.6 Micro-motores

O protótipo emprega micro-motores Pololu de 6V com caixa de redução integrada na proporção 50:1. Esses motores fornecem torque aproximado de 0,14 N·m no eixo de saída, operando com tensões entre 3V e 9V e corrente de stall próxima de 1,6A.

A utilização de motores de corrente contínua mostrou-se adequada ao contexto educacional do projeto, considerando fatores como baixo custo, simplicidade de controle e compatibilidade com plataformas de prototipagem rápida. Além disso, seu peso reduzido favorece aplicações em pequenos protótipos móveis.



Figura 8: Micro-motor Pololu com caixa de redução.

Fonte: pololu.com.

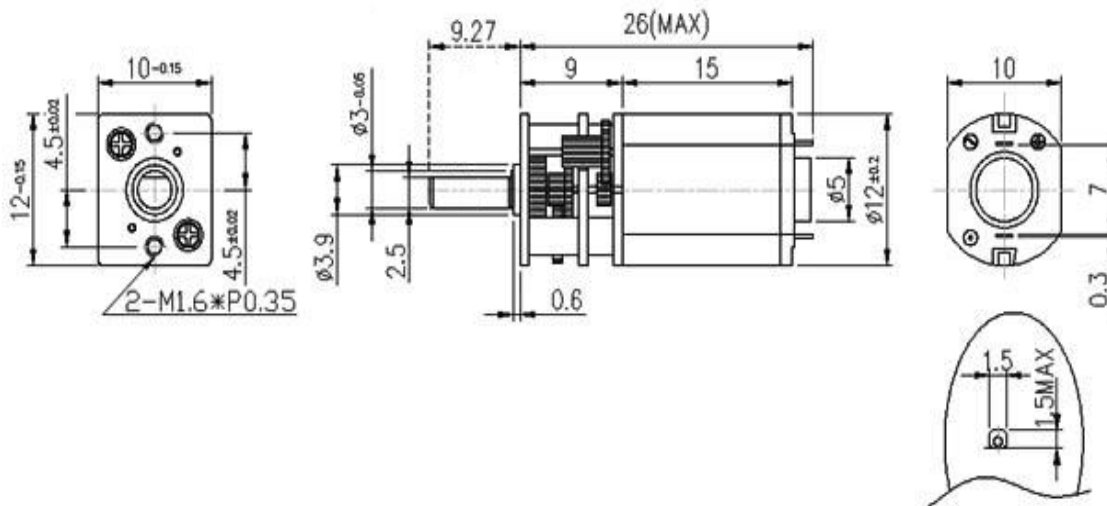


Figura 9: Dimensões do micro-motor.

Fonte: pololu.com.

4. DEFINIÇÃO E PROGRAMAÇÃO DO ROBÔ

A estrutura física do protótipo foi desenvolvida a partir de um chassi Pololu circular, escolhido em razão de sua leveza, simplicidade de montagem e compatibilidade com projetos de robótica educacional de baixo custo. Sobre essa base foram fixados dois micro-motores de corrente contínua com caixas de redução acopladas às rodas do robô.

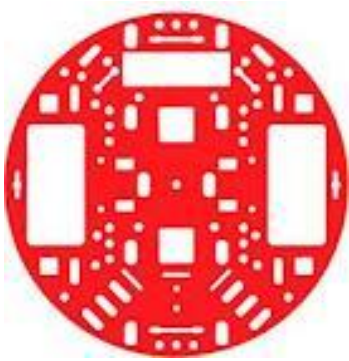


Figura 10: Chassi Pololu utilizado no projeto.

Fonte: robocore.net.



Figura 11: Rodas utilizadas no projeto.

Fonte: robocore.net.

Os motores são conectados à placa Arduino Romeo por meio dos terminais M1 (+/-) e M2 (+/-), responsáveis pela alimentação elétrica e pelo controle de rotação. A inversão da polaridade elétrica aplicada aos motores permite alterar o sentido de rotação, fazendo com que estes operem em sentido horário ou anti-horário.

Como os motores foram montados em posições opostas no chassi, tornou-se necessário inverter a polaridade de conexão de um deles na própria placa, garantindo que ambos executassem movimentos coerentes durante a locomoção do robô. Essa configuração permitiu simplificar a lógica de programação utilizada no controle de deslocamento.

O trecho de código responsável pela definição dos motores é apresentado a seguir:

```
int M1 = 4;
```

```
int M2 = 7;
```

Os valores 4 e 7 correspondem aos pinos digitais utilizados para controle do sentido de rotação dos motores direito e esquerdo, respectivamente.

Para determinar o sentido de movimento, utilizam-se os comandos LOW e HIGH, responsáveis por controlar a polaridade aplicada aos motores através da ponte H integrada à placa Romeo. Dessa forma, a alteração do estado lógico enviado ao motor modifica seu sentido de rotação.

Além da direção de movimento, também foi necessário definir a velocidade individual de cada motor. Para isso, foram utilizados os pinos PWM da placa:

```
int E1 = 5;
```

```
int E2 = 6;
```

Os valores 5 e 6 correspondem aos pinos responsáveis pelo controle de velocidade dos motores direito e esquerdo.

Com essas variáveis declaradas, foram implementadas duas funções básicas de locomoção: Advance, responsável pelo movimento para frente, e Reverse, utilizada para deslocamento em marcha ré.

```
void Advance(char a, char b) {  
    analogWrite(E1, a);  
    digitalWrite(M1, LOW);  
    analogWrite(E2, b);  
    digitalWrite(M2, LOW);  
}  
void Reverse(char a, char b) {  
    analogWrite(E1, a);  
    digitalWrite(M1, HIGH);  
    analogWrite(E2, b);  
    digitalWrite(M2, HIGH);  
}
```

As variáveis a e b determinam a velocidade aplicada individualmente a cada motor. Essa estratégia permitiu controlar curvas e rotações sem necessidade de funções específicas para giro à direita (Right), giro à esquerda (Left) ou parada (Stop). Bastava modificar os valores atribuídos aos motores durante a execução das funções Advance e Reverse.

Essa solução contribuiu para simplificar a estrutura do código e reduzir redundâncias na programação do sistema de locomoção.

Além dos motores, o protótipo utiliza um sensor analógico infravermelho Sharp para detecção de obstáculos posicionados à frente do robô. O sensor é conectado à placa Romeo por meio de três fios: o fio branco transmite os valores lidos pelo sensor para uma entrada analógica da placa; o fio preto conecta-se ao GND (terra); e o fio vermelho é ligado à alimentação de 5V. O trecho de código responsável pela leitura do sensor é apresentado abaixo:

```
int sensorPin = 0;  
val = analogRead(sensorPin);  
if (val < 350) {  
    Advance(50,55);  
}  
if (val > 350) {  
    ...  
    val = analogRead(sensorPin);  
}
```

Inicialmente, define-se a variável `sensorPin` com valor zero, correspondente à entrada analógica utilizada para leitura do sensor. O comando `analogRead()` realiza a captura do valor detectado e o armazena na variável `val`.

Quando o valor lido é inferior a 350, interpreta-se que o obstáculo está distante, permitindo que o robô continue seu deslocamento para frente. Quando o valor ultrapassa esse limite, entende-se que existe um obstáculo próximo ao protótipo, acionando as rotinas de desvio implementadas no segundo bloco condicional.

Essa lógica de controle permitiu integrar leitura sensorial, tomada de decisão automatizada e acionamento dos motores em tempo real, aproximando o funcionamento do protótipo de conceitos fundamentais presentes em sistemas embarcados e aplicações introdutórias de automação e Inteligência Artificial.

4.1 IDE (Ambiente Integrado de Desenvolvimento)

A comunicação entre o computador e a placa Arduino é realizada por meio da plataforma IDE (Integrated Development Environment), ambiente oficial de desenvolvimento utilizado para programação das placas Arduino. A ferramenta apresenta interface intuitiva e recursos voltados à prototipagem rápida, facilitando sua utilização mesmo por usuários em primeiros contatos com programação embarcada.

Além da edição de códigos em linguagem baseada em C++, a IDE oferece exemplos prontos, documentação técnica, ferramentas de compilação e recursos de depuração, contribuindo para a aprendizagem de conceitos introdutórios de programação e automação.

A plataforma é utilizada em diferentes modelos de placas Arduino e possui como objetivo padronizar e simplificar o desenvolvimento de projetos eletrônicos educacionais e de prototipagem.

Para utilização do software, é necessário inicialmente realizar o download da IDE no site oficial do Arduino.



Figura 12: Site oficial do Arduino.

Fonte: arduino.cc.

Na página de download, o usuário deve selecionar o sistema operacional correspondente ao computador utilizado. O pacote disponibilizado inclui também os drivers necessários para o reconhecimento da placa pelo sistema operacional.

Após o download, basta descompactar os arquivos em uma pasta local, pois a IDE não exige processo convencional de instalação. O acesso ao sistema é realizado diretamente pelo executável `arduino.exe`.

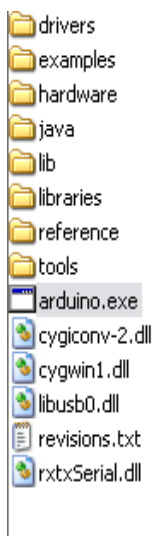


Figura 13: Arquivos e pastas extraídos da IDE Arduino.

Fonte: Autores (2021)

Depois da inicialização da plataforma, a placa pode ser conectada ao computador por meio da interface USB. Em seguida, torna-se necessário configurar a porta serial correspondente e selecionar o modelo da placa utilizada. Essas configurações são realizadas no menu “Tools” da IDE.

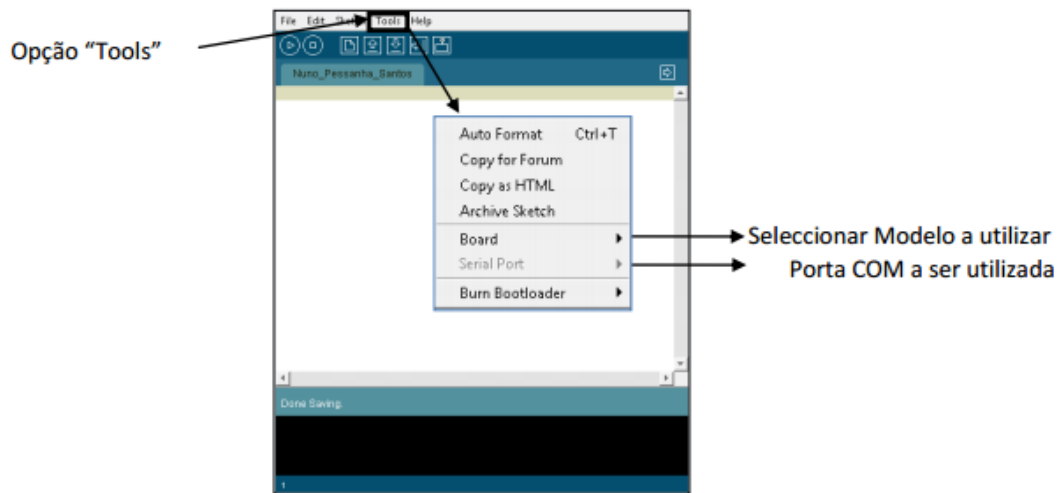


Figura 14: Configuração da IDE Arduino.

Fonte: Autores (2021)

Após a configuração, o usuário pode editar o código-fonte, realizar a compilação e efetuar o envio (upload) do programa para a placa Arduino.

O ciclo básico de desenvolvimento na plataforma pode ser resumido em quatro etapas principais:

edição do código;

compilação;

envio para a placa;

execução do programa embarcado.

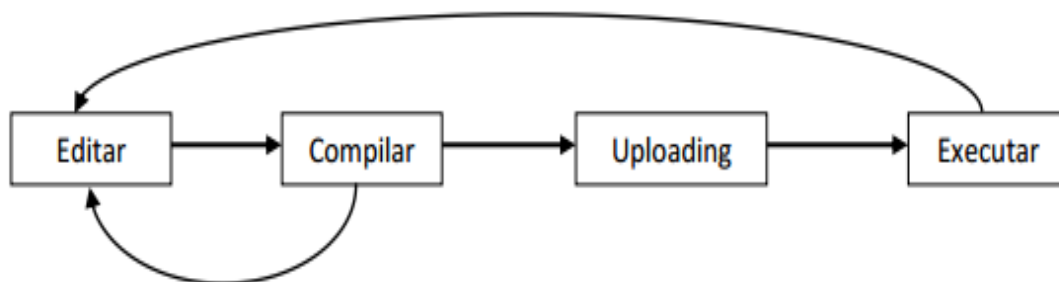


Figura 15: Fases do ciclo de desenvolvimento na IDE Arduino.

Fonte: Autores (2021)

É importante destacar que, após o envio do código para a placa, não é possível recuperar integralmente o programa diretamente do microcontrolador. Dessa forma, recomenda-se manter cópias do código-fonte armazenadas no computador, possibilitando futuras alterações e aperfeiçoamentos.

A Figura 16 apresenta o ambiente principal da IDE Arduino e alguns de seus recursos básicos de desenvolvimento.

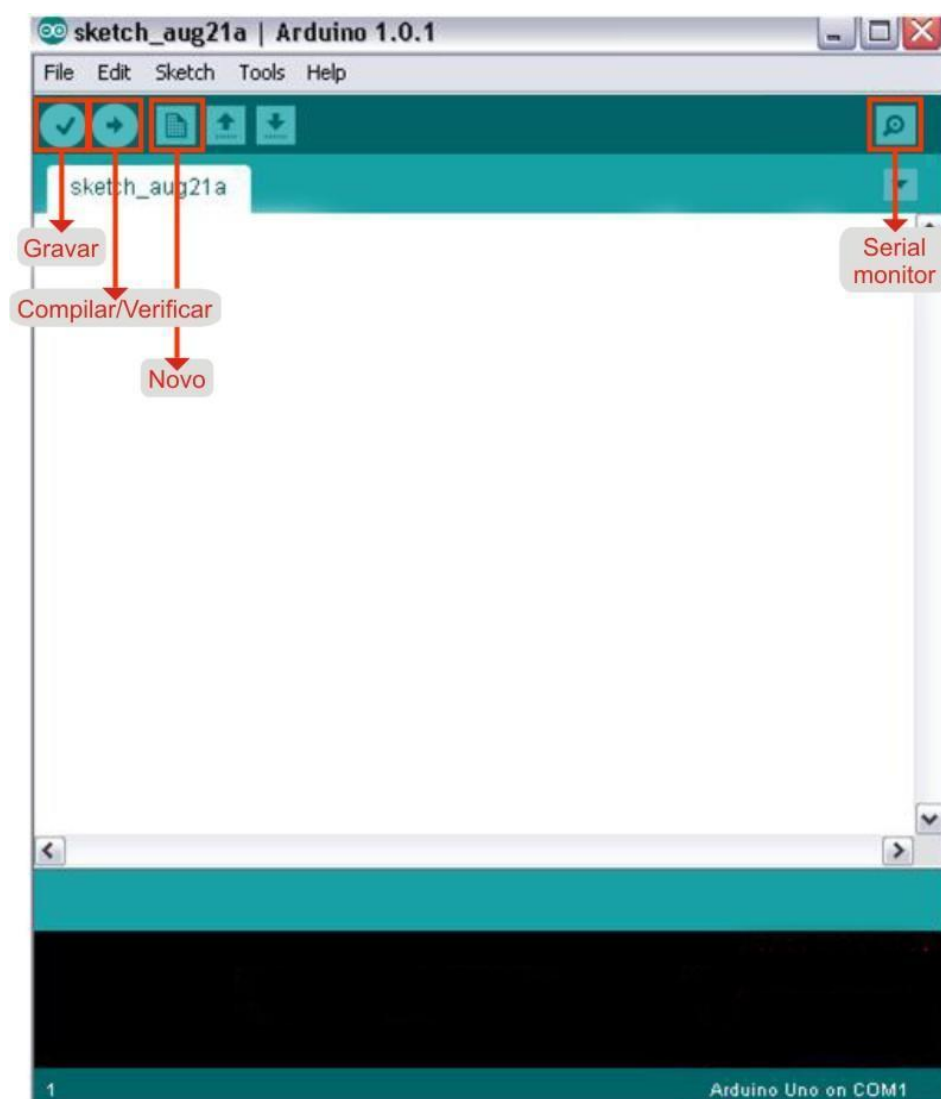


Figura 16: Ambiente de desenvolvimento da IDE Arduino.

Fonte: Autores (2021)

5. ESTUDO DE CASO: AJUSTES E APRENDIZAGENS A PARTIR DOS TESTES

Durante o processo de montagem e realização dos testes do carro-robô, tornou-se evidente a influência das condições físicas do ambiente sobre o funcionamento do protótipo. Aspectos como atrito, peso, distribuição de carga e desgaste mecânico interferiram diretamente no desempenho do sistema, exigindo sucessivos ajustes tanto na estrutura física quanto na programação embarcada.

Do ponto de vista pedagógico, esse processo evidencia uma das principais potencialidades da robótica educacional: a possibilidade de articular teoria e prática em situações reais de resolução de problemas. Conforme discutido por Papert (1985), a aprendizagem torna-se mais significativa quando o estudante participa ativamente da experimentação, da identificação de falhas e da reformulação de soluções. Nesse contexto, o erro deixa de representar apenas um resultado negativo e passa a integrar o próprio movimento investigativo.

Uma das primeiras dificuldades identificadas relacionou-se ao atrito entre o robô e a superfície utilizada nos testes. Inicialmente, os motores operavam com velocidade elevada, utilizando os valores Advance(80,80). Em superfícies de baixo atrito, o robô deslocava-se rapidamente, percorrendo em poucos instantes a área de atuação do sensor infravermelho. Como consequência, o tempo de resposta tornou-se insuficiente para realização adequada das manobras de desvio.

A partir de sucessivos testes práticos, observou-se a necessidade de reduzir a velocidade de deslocamento do protótipo. Essa etapa exigiu análise empírica do comportamento do robô e sucessivos refinamentos nos parâmetros utilizados na programação.

Durante os experimentos também foi identificado que um dos motores necessitava de tensão ligeiramente superior para atingir velocidade semelhante à do outro motor. Após análises, verificou-se desgaste nas engrenagens internas da caixa de redução, ocasionando travamentos intermitentes e fazendo com que o robô passasse a se locomover em círculos após alguns segundos de funcionamento.

Embora o problema não impedisse a continuidade dos estudos envolvendo sensores e atuadores, tornou-se necessária a substituição do componente defeituoso. Após a troca do motor, foi possível restabelecer maior estabilidade no deslocamento do protótipo.

Posteriormente, a inclusão de duas baterias acopladas ao chassi (uma destinada à alimentação da placa e outra ao acionamento dos motores) aumentou significativamente o peso total do robô. Como consequência, o atrito e a resistência ao deslocamento também aumentaram, exigindo novos ajustes nos parâmetros de velocidade.

Após diferentes testes, definiu-se que os valores Advance(50,50) e Reverse(50,50) apresentavam melhor equilíbrio entre velocidade de deslocamento e tempo de resposta do sensor. Essa configuração permitiu ao robô identificar obstáculos e executar manobras de recuo antes da colisão.

Em outro momento dos experimentos, uma queda do protótipo ocasionou novo desalinhamento mecânico em um dos motores, fazendo com que este voltasse a operar com velocidade inferior à do outro. Para compensar essa diferença, optou-se pela utilização de valores assimétricos no controle dos motores, empregando o comando Advance(50,55). Essa solução permitiu restabelecer a estabilidade da trajetória do robô.

Com o funcionamento adequado da locomoção, iniciou-se a etapa de desenvolvimento das estratégias de desvio de obstáculos. A primeira abordagem testada consistiu em reduzir gradativamente a velocidade de um dos motores à medida que o robô se aproximava do obstáculo. Para isso, utilizou-se uma expressão baseada na leitura do sensor, ajustando dinamicamente a rotação de um dos motores conforme o aumento do valor da variável val.

Durante os testes práticos, verificou-se que o robô iniciava corretamente a curva ao detectar o obstáculo. Contudo, assim que o sensor deixava de identificar o objeto, o sistema retomava imediatamente o deslocamento frontal, ocasionando colisões laterais em determinadas situações.

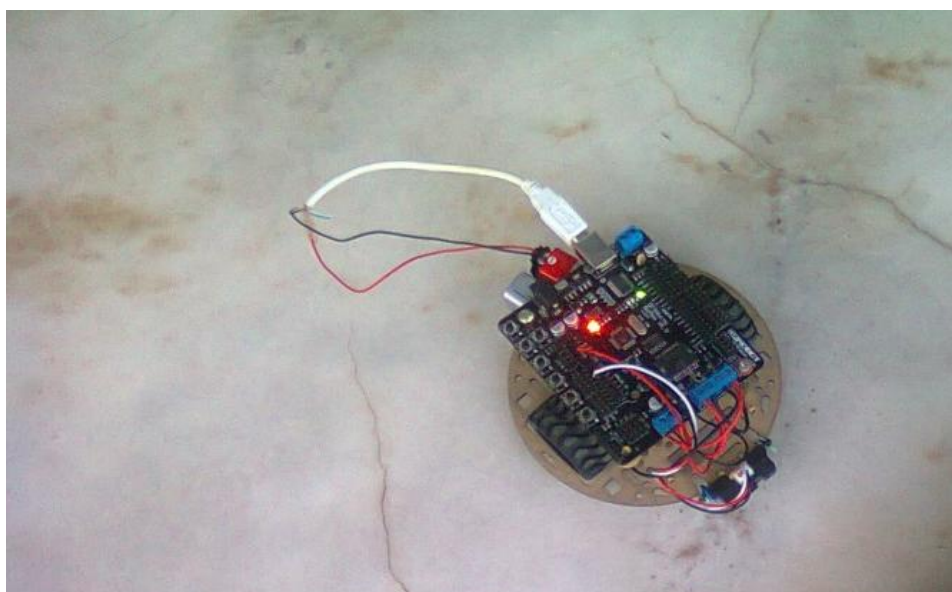


Figura 17: Teste com o robô.

Fonte: Autores (2021)

Diante dessa limitação, optou-se pela reformulação da lógica de desvio. Em vez de apenas reduzir a velocidade de um dos motores, o robô passou a executar uma sequência composta por recuo temporário seguido de rotação sobre uma das rodas. Nessa estratégia, ao detectar um obstáculo próximo, o sistema interrompia momentaneamente o deslocamento

frontal, executava uma marcha ré por alguns milissegundos e, em seguida, realizava um giro aproximado de 90° utilizando apenas um dos motores.

Essa solução apresentou resultados significativamente mais estáveis durante os testes, reduzindo colisões laterais e ampliando a capacidade de navegação do protótipo em ambientes com obstáculos.

Do ponto de vista formativo, o percurso realizado durante os testes evidencia como atividades de robótica educacional mobilizam competências relacionadas à observação, formulação de hipóteses, validação empírica e resolução de problemas. Mais do que reproduzir procedimentos técnicos, os estudantes são inseridos em práticas investigativas que se aproximam do próprio método científico, articulando programação, eletrônica, automação e pensamento computacional em experiências concretas de aprendizagem.

5.1 Variações do sensor e ajustes no sistema de desvio

Conforme discutido anteriormente, o comportamento prático do sensor infravermelho apresentou diferenças em relação às especificações teóricas fornecidas pelo fabricante, sobretudo em função das interferências do ambiente físico e das limitações do campo de detecção. Esse comportamento exigiu sucessivas adaptações na lógica de programação utilizada para o desvio de obstáculos.

21

Nos primeiros testes, foi utilizada uma estratégia baseada na redução gradual da velocidade de um dos motores à medida que o robô se aproximava do obstáculo. Para isso, empregou-se o comando:

*Advance(60-ir*0.07,80)*

Nessa lógica, o valor lido pelo sensor era multiplicado por 0.07 e subtraído de 60, reduzindo progressivamente a velocidade de um dos motores enquanto o outro permanecia com velocidade constante. O objetivo era fazer com que o robô realizasse curvas suaves ao identificar obstáculos próximos.

Embora a estratégia funcionasse parcialmente, os testes práticos demonstraram limitações importantes. Em muitos casos, o robô iniciava corretamente a curva, mas retomava imediatamente o deslocamento frontal assim que o sensor deixava de detectar o obstáculo. Como consequência, a parte lateral do chassi acabava colidindo com o objeto.

Na tentativa de minimizar esse problema, foi proposta a utilização de três sensores digitais de fim de curso, posicionados nas laterais e na parte traseira do robô.



Figura 18: Sensor de fim de curso.

Fonte: robocore.net.

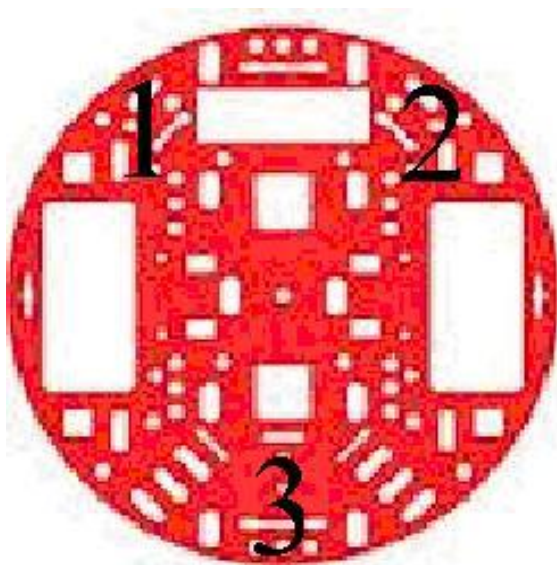


Figura 19: Localização dos sensores no chassi.

Fonte: Autores (2021)

A proposta consistia em utilizar os sensores digitais como mecanismo complementar de segurança. Dessa forma, eventuais falhas no campo de detecção do sensor infravermelho poderiam ser compensadas pelos sensores de contato físico instalados no robô.

Nessa configuração:

os sensores laterais seriam responsáveis por identificar colisões durante curvas;

o sensor traseiro monitoraria obstáculos durante movimentos de ré.

A Figura 20 apresenta a configuração do protótipo com os sensores instalados.

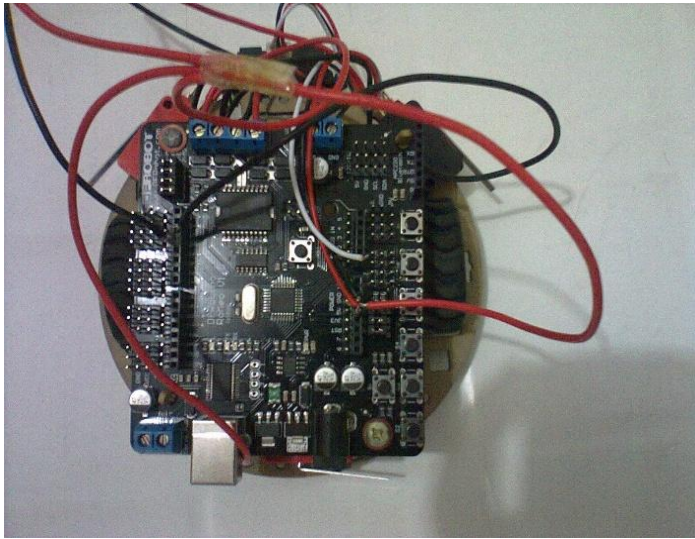


Figura 20: Robô com sensores sensíveis ao toque.

Fonte: Autores (2021)

Os trechos de código utilizados para leitura dos sensores digitais são apresentados a seguir.

```
Sensor direito
void setup() {
  Serial.begin(9600);
  pinMode(0, INPUT);
}
void loop() {
  int ir = digitalRead(2);
  Serial.println(ir, DEC);
  delay(10);
}
Sensor esquerdo
void setup() {
  Serial.begin(9600);
  pinMode(0, INPUT);
}
void loop() {
  int ir = digitalRead(3);
  Serial.println(ir, DEC);
  delay(10);
}
Sensor traseiro
void setup() {
  Serial.begin(9600);
  pinMode(0, INPUT);
}
void loop() {
  int ir = digitalRead(0);
  Serial.println(ir, DEC);
  delay(10);
}
```

Quando os sensores digitais não estavam pressionados, retornavam valor lógico 0. Quando acionados por contato físico, retornavam valor lógico 1, permitindo que o sistema executasse rotinas específicas de correção de movimento.

Entretanto, durante os testes, verificou-se que os sensores de fim de curso apresentavam baixa sensibilidade para o contexto do projeto. Como o protótipo possuía baixo peso e velocidade reduzida, muitas colisões não geravam força suficiente para acionar adequadamente os sensores mecânicos.

A partir dessa limitação, concluiu-se que essa estratégia seria mais adequada em projetos com motores mais robustos, maior velocidade de deslocamento ou estruturas físicas mais pesadas.

Diante disso, optou-se por abandonar a lógica baseada em sensores de contato e adotar uma nova estratégia de desvio fundamentada apenas na leitura do sensor frontal infravermelho. Nessa abordagem, ao detectar um obstáculo próximo, o robô deveria:

interromper momentaneamente o deslocamento;
executar uma marcha ré;
realizar uma rotação para alterar sua trajetória.

O trecho de código implementado foi:

```
if (val > 350) {  
  Advance(0,0);  
  delay(500);  
  
  Reverse(60,60);  
  delay(900);  
  
  Advance(60,0);  
  delay(900);  
  
  val = analogRead(sensorPin);}
```

Nesse algoritmo, quando o sensor identifica um obstáculo a aproximadamente 15 cm de distância, o robô interrompe o movimento por 0,5 segundos, executa uma marcha ré durante 0,9 segundo e, em seguida, realiza uma rotação para a direita durante o mesmo intervalo de tempo.

Os resultados obtidos mostraram-se mais eficientes do que as estratégias anteriores, reduzindo significativamente as colisões laterais e ampliando a estabilidade do sistema de navegação.

Do ponto de vista formativo, os ajustes realizados ao longo do projeto evidenciam como a robótica educacional favorece processos de aprendizagem baseados em experimentação, análise crítica e reformulação contínua de hipóteses. Mais do que reproduzir soluções prontas,

os estudantes são inseridos em práticas investigativas próximas do trabalho científico e do desenvolvimento tecnológico contemporâneo.

6. CONSIDERAÇÕES FINAIS

Este trabalho apresentou o processo de construção e programação de um carro-robô autônomo baseado na plataforma Arduino, desenvolvido com o objetivo de detectar obstáculos e executar manobras automatizadas de desvio. Ao longo do projeto, foram estudados conceitos relacionados à programação embarcada, sensores, atuadores, microcontroladores e controle de motores, integrando conhecimentos de eletrônica, automação e computação em uma experiência prática de robótica educacional.

Durante os testes realizados, tornou-se possível identificar diferentes fatores externos capazes de interferir diretamente no desempenho do protótipo, como atrito, peso, distribuição de carga, sensibilidade dos sensores e limitações mecânicas dos motores utilizados. As adaptações implementadas ao longo do processo evidenciaram a necessidade de constantes ajustes entre estrutura física e programação, aproximando o desenvolvimento do robô de dinâmicas reais de investigação tecnológica e resolução de problemas.

Os resultados obtidos demonstraram que o objetivo inicial do projeto foi alcançado, uma vez que o robô apresentou capacidade satisfatória de detectar obstáculos e executar rotinas automatizadas de desvio. A definição adequada da relação entre velocidade, peso e tempo de resposta do sensor permitiu maior estabilidade no deslocamento e redução significativa de colisões durante os testes realizados.

Para além da dimensão técnica, o trabalho também evidenciou o potencial pedagógico da robótica educacional como espaço de formação científica, desenvolvimento do pensamento computacional e promoção da cidadania digital. Conforme discutido ao longo do artigo, atividades dessa natureza favorecem processos de aprendizagem baseados em experimentação, investigação, análise crítica e reformulação contínua de hipóteses, aproximando-se de perspectivas construcionistas de aprendizagem (Papert, 1985; Valente, 1993).

Em um contexto marcado pela crescente presença de sistemas algorítmicos e aplicações de Inteligência Artificial na vida social, experiências práticas com robótica educacional podem contribuir para que estudantes compreendam criticamente os processos tecnológicos que estruturam a cultura digital contemporânea. Mais do que utilizar tecnologias, trata-se de

possibilitar que crianças e jovens compreendam seus funcionamentos, limitações, impactos sociais e implicações éticas.

Nesse sentido, a robótica educacional não deve ser compreendida apenas como estratégia de incentivo às áreas STEM, mas também como possibilidade concreta de articulação entre educação científica, pensamento crítico e formação cidadã em contextos escolares contemporâneos.

REFERÊNCIAS

- ALLDATASHEET. **L298 Datasheet**. Disponível em: <https://www.alldatasheet.com>. Acesso em: 18 jul. 2012.
- ARDUINO. **Official Arduino Website**. Disponível em: <https://www.arduino.cc>. Acesso em: 30 jul. 2012.
- BANZI, Massimo. **Getting Started with Arduino**. 2. ed. Sebastopol: O'Reilly Media, 2011.
- BELLONI, Maria Luiza. **O que é mídia-educação**. Campinas: Autores Associados, 2005.
- BRASIL. Ministério da Educação. **IA na Educação Básica: documento orientador**. Brasília: MEC, 2026.
- BRASIL. Ministério da Educação. **Referencial para Desenvolvimento e Uso Responsáveis de Inteligência Artificial na Educação**. Brasília: MEC, 2026.
- DELEUZE, Gilles; GUATTARI, Félix. **Mil platôs: capitalismo e esquizofrenia**. v. 1. Rio de Janeiro: Editora 34, 1995.
- DFROBOT. **Arduino Romeo Board**. Disponível em: <https://www.dfrobot.com>. Acesso em: 20 jul. 2012.
- GALLO, Sílvio. **Conhecimento, transversalidade e educação: para além da interdisciplinaridade**. Campinas: Papirus, 1995.
- HOBBYTRONICS. **ATmega328 Microcontroller**. Disponível em: <https://www.hobbytronics.co.uk>. Acesso em: 22 jul. 2012.
- HOWSTUFFWORKS. **How Microcontrollers Work**. Disponível em: <http://www.howstuffworks.com>. Acesso em: 20 jul. 2012.
- LÉVY, Pierre. **Cibercultura**. 3. ed. São Paulo: Editora 34, 2005.
- MAXWELL BOHR. **Arduino: técnicas de projetos e conceitos**. São Paulo: Érica, 2012.
- MICROBERTS, Michael. **Arduino Básico**. São Paulo: Novatec, 2011.
- PAPERT, Seymour. **Logo: computadores e educação**. São Paulo: Brasiliense, 1985.
- POLOLU. **Pololu Robot Chassis and Motors**. Disponível em: <https://www.pololu.com>. Acesso em: 25 jul. 2012.
- ROBOCORE. **Componentes para robótica educacional**. Disponível em: <https://www.robocore.net>. Acesso em: 25 jul. 2012.
- SILVEIRA, Paulo Rogério da; SANTOS, Winderson Eugenio dos. **Automação e controle discreto**. São Paulo: Érica, 2002.

STMICROELECTRONICS. **L298 Dual Full-Bridge Driver Datasheet**. Disponível em: <http://www.st.com>. Acesso em: 18 jul. 2012.

UNESCO. **Guia para a IA Generativa na Educação e na Pesquisa**. Paris: UNESCO, 2024.

VALENTE, José Armando. **Computadores e conhecimento: repensando a educação**. Campinas: UNICAMP/NIED, 1993.