

ULTRON

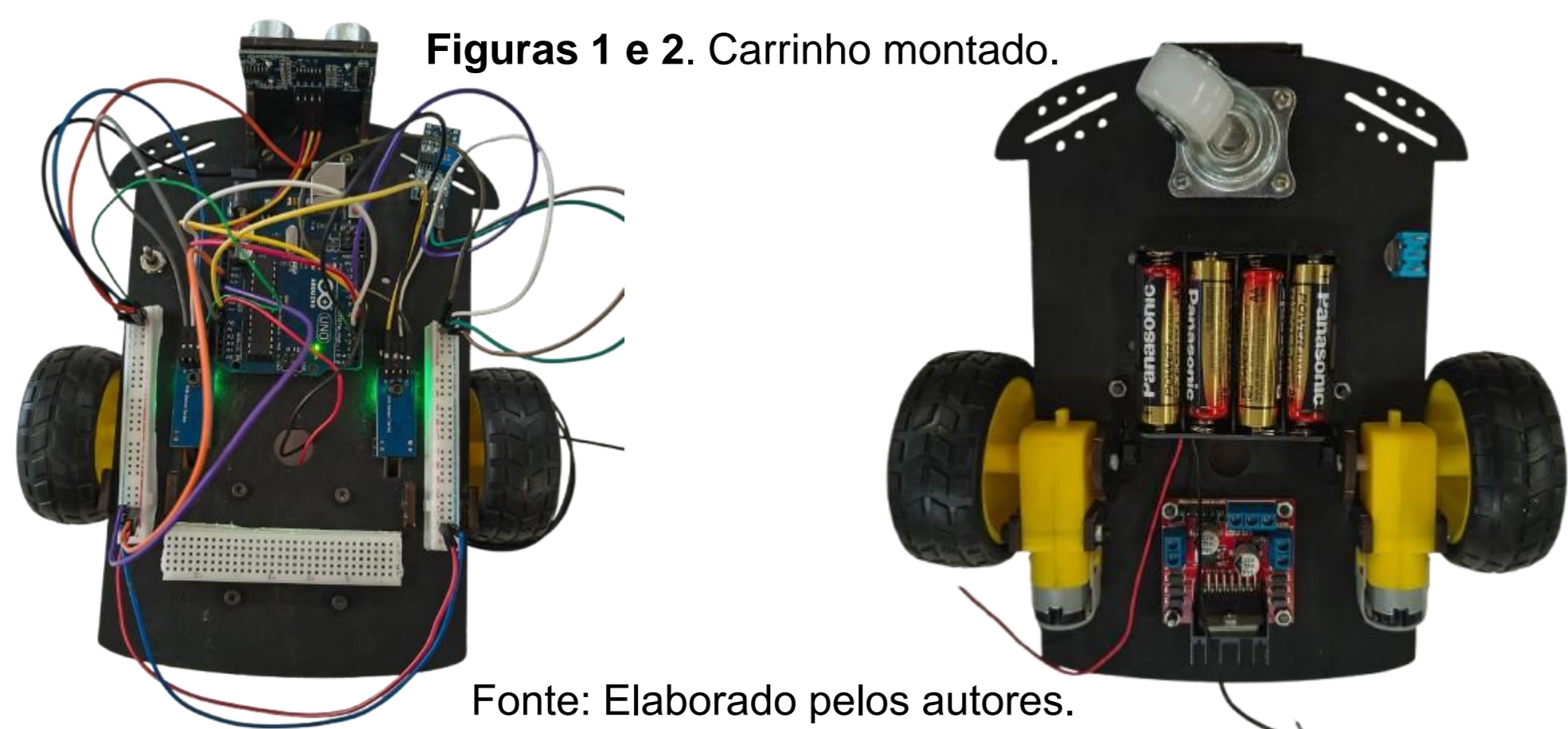
Daniel Benites Ferreira - RA:152023
Eric Schechtel Bergmann - RA: 235356
Larissa Baraldi Terciani - RA: 236239

Luan Andre Pires Correa - RA: 234815
Mirella Cazarin Terenciano - RA:236068

Profº Rodrigo Henrique Geraldo

INTRODUÇÃO

O projeto proposto é a elaboração de um carrinho seguidor de linha, que utilize de programação de Arduino, sensores infravermelhos e sensor ultrassônico para identificação do desenho da linha e para parada ao identificar um obstáculos. Além disso, através da alimentação por pilhas, ou baterias, o motor DC movimenta as rodas automaticamente, pela programação do Arduino.



Figuras 1 e 2. Carrinho montado.

Fonte: Elaborado pelos autores.

JUSTIFICATIVA

O projeto do Robô Seguidor de Linha é uma forma didática de integrar eletrônica, programação, automação e controle, componentes essenciais na formação de um engenheiro mecatrônico. Além de ser a porta de entrada prática para a Robótica, expandindo a visão de atuação da profissão e abrangendo mais áreas de interesse.

OBJETIVOS e ODS

Os objetivos do projeto são desenvolver a capacidade de integração dos diferentes componentes curriculares do curso de mecatrônica e aprimorar os conhecimentos de programação, robótica e IoT. Como ODS – Objetivo de Desenvolvimento Sustentável - que podem ser atreladas ao projeto destacam-se as ODS 4 (Educação de qualidade), por aplicar na prática os conceitos aprendidos pelos alunos de mecatrônica, e 9 (Indústria, inovação e infraestrutura), por se tratar de um item de automação e tecnologia.

Figuras 3 e 4. ODS.



Fonte: <https://brasil.un.org/pt-br/sdgs>

PERSPECTIVAS

Como melhorias o grupo almeja aumentar o número de sensores infravermelhos para que seja possível uma programação mais robusta e um carro mais veloz e desenvolver uma carcaça para que os componentes eletrônicos não fiquem a mostra.

AGRADECIMENTOS

Agradecemos o suporte dos professores Rafael da Paz e Denis Borgi ao decorrer das aulas e a orientação do professor Rodrigo Geraldo.

RESULTADOS E VALIDAÇÃO

Para validar a programação desenvolvida, primeiro, foi utilizado o próprio software de programação, o Arduino IDE, para compilar o código e verificar se apresentava algum erro na escrita do código, para posteriormente transferir a programação ao microcontrolador Arduino e assim realizar o teste na pista desenhada no laboratório da FACENS, o Fablab, para verificar se realmente a programação está correta e o carrinho está seguindo os passos desejados, que seria seguir a linha de forma correta, sendo capaz de realizar curvas e parando, caso necessário, ao identificar algum tipo de obstáculo. Além disso, antes de colocar o carrinho para rodar foram alimentados todos os componentes para verificar se estavam todos funcionando adequadamente e para validar as ligações, por meio da identificação visual dos LEDs.

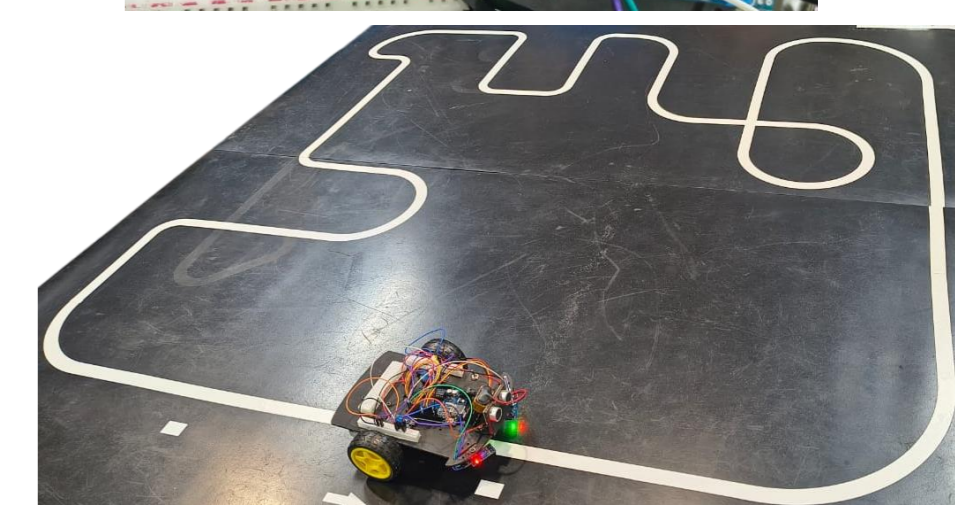
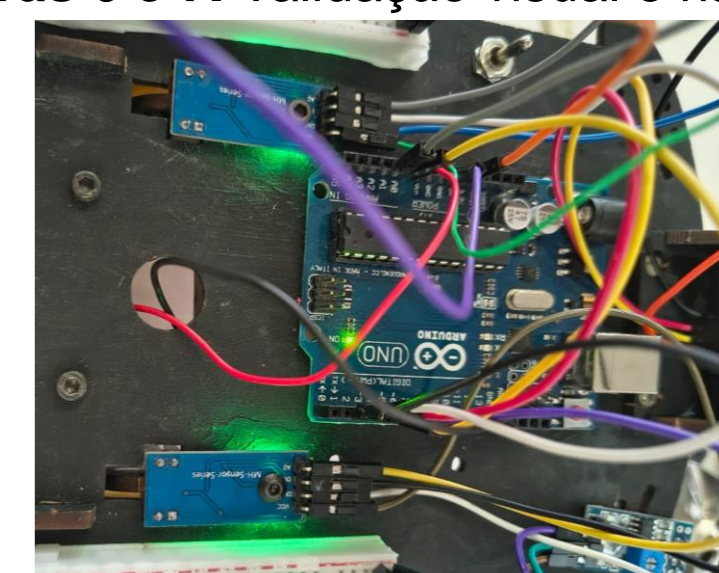
Figuras 6 e 7. Validação visual e na pista.

Figura 5. Software Arduino IDE.

```

1 // --- PINOS DOS MOTORES (DIREÇÃO) ---
2 const int in1 = 7;
3 const int in2 = 6;
4 const int in3 = 5;
5 const int in4 = 4;
6
7 // --- PINOS DOS MOTORES (VELOCIDADE / PWM) ---
8 // Precisa ser pino com "-" (PWM)
9 const int enA = 9; // ENB da Ponte H (Motor Esquerda)
10 const int enB = 10; // ENB da Ponte H (Motor Direita)
11
12 // --- PINOS DOS SENSORES DE LINHA ---
13 const int sensorEsquerda00 = 2;
14 const int sensorDireita00 = 3;
15
16 // --- PINOS DO SENSOR ULTRASSÔNICO (HC-SR04) ---
17 const int PINO_TRIG = 12;
18 const int PINO_ECHO = 13;
19
20 // --- CONFIGURAÇÕES DE VELOCIDADE (0 a 255) ---
21 const int VELOCIDADE_ALTA = 200; // Velocidade ao andar reto
22 const int VELOCIDADE_BAIXA = 90; // Velocidade da rota de dentro da curva
23 const int VELOCIDADE_ZERO = 0;
24
25 // --- CONFIGURAÇÃO DE PARADA ---
26 const int DISTANCIA_PARADA = 10; // Distância em CM para parar
27
28 void setup() {
29
30   pinMode(in1, OUTPUT);
31   pinMode(in2, OUTPUT);
32   pinMode(in3, OUTPUT);
33   pinMode(in4, OUTPUT);
34   pinMode(enA, OUTPUT);
35   pinMode(enB, OUTPUT);

```



Fonte: Elaborado pelos autores.

Fonte: Elaborado pelos autores.

ORÇAMENTO

Do orçamento apresentado abaixo foi necessária a compra apenas da "roda boba", o conjunto de motor com roda, o módulo encoder, a ponte H, as pilhas com o suporte e o sensor. Portanto o orçamento dos itens foi realizado separado, porém há um kit de robô seguidor de linha pronto que custa em média R\$300,00.

Tabela 1. Orçamento.

COMPONENTE	QTD.	Loja 1	Loja 2	Loja 3	Preço 1	Preço 2	Preço 3	Média
Placa MDF 30x40cm	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 4,96	R\$ 60,00	R\$ 70,00	R\$ 44,99
Conjunto motor DC com redutor e rodas	2	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 29,99	R\$ 34,00	R\$ 15,50	R\$ 26,50
Sensor de Velocidade/Módulo Encoder	2	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 21,99	R\$ 8,90	R\$ 11,00	R\$ 13,96
"Roda boba"	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 20,00	R\$ 7,46	R\$ 9,00	R\$ 12,15
Mini chave ON/OFF	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 18,26	R\$ 4,60	R\$ 14,50	R\$ 12,45
Sensor Ultrassônico HC-SR04	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 19,00	R\$ 12,22	R\$ 25,96	R\$ 19,06
Suporte para pilhas	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 3,52	R\$ 5,33	R\$ 22,00	R\$ 10,28
Ponte H	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 19,90	R\$ 19,99	R\$ 24,90	R\$ 21,60
Protoboard	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 19,00	R\$ 9,75	R\$ 12,00	R\$ 13,58
Sensor de Linha infravermelho	2	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 13,99	R\$ 8,90	R\$ 11,00	R\$ 11,30
Microcontrolador Arduino Uno	1	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 34,27	R\$ 49,26	R\$ 65,00	R\$ 49,51
Pilhas 3A Alcalinas	4	Mercado Livre	Baú da Eletrônica	Cinestec	R\$ 11,40	R\$ 27,00	R\$ 32,00	R\$ 23,47

Fonte: Elaborado pelos autores.

Menor custo
Maior Custo

CONCLUSÃO

Por meio do projeto foi possível aprimorar as habilidades de eletrônica e IoT, como programação do Arduino, solda e programação de sensores, além de abranger a atuação como engenheiros mecatrônicos no campo da robótica.