



# Relatório de Atividade

## Robô Desviador de Obstáculos

GIOVANNA M. P. ZAGO, WILLIAN B. SILVA

Petianos responsáveis pelo projeto, em ordem alfabética

---

### Resumo

*O Robô Desviador de Obstáculos foi um projeto proposto pelos alunos do EsColab da Escola Estadual Francisco Menezes Filho no primeiro semestre de 2019. Quando o Robô encontra um obstáculo a menos de 40cm de distância, ele volta para trás e desvia para a sua direita.*

---

### CONTEÚDO

<b>I</b>	<b>Introdução</b>	<b>1</b>
<b>II</b>	<b>Materiais e Métodos</b>	<b>2</b>
<b>III</b>	<b>Resultados</b>	<b>5</b>
<b>IV</b>	<b>Discussão</b>	<b>5</b>

#### I. INTRODUÇÃO

O EsColab é um projeto de extensão voltado ao ensino de robótica para estudantes do ensino médio de escolas públicas. Idealizado pela Enactus UFMG, o projeto tem como objetivos principais a difusão do conhecimento e diminuição da evasão escolar utilizando como ferramenta a multidisciplinaridade da robótica.

Ao longo das oficinas realizadas na Escola Estadual Francisco Menezes Filho, foram desenvolvidas diversas habilidades em áreas necessárias para a confecção de um projeto de robótica, tais quais: Circuitos Elétricos, Eletrônica, Programação e Projeto Técnico.

Nas últimas oficinas ministradas no primeiro semestre de 2019, foi proposto aos alunos a escolha de um projeto a ser desenvolvido que envolvesse alguns conceitos e componentes vistos por eles ao longo das aulas. Sendo assim, o projeto escolhido por eles foi um Robô Desviador de Obstáculos.

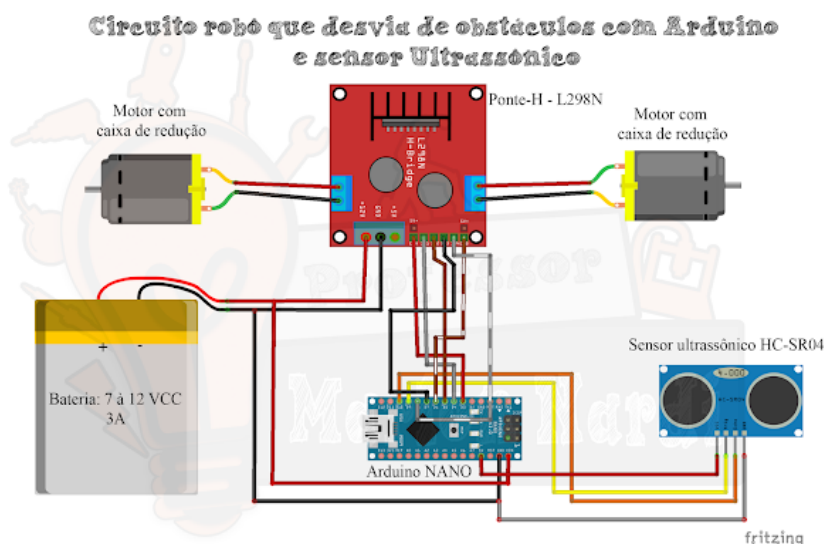
## II. MATERIAIS E MÉTODOS

### Materiais

- 1 Arduino UNO ou NANO
- Bateria 7 a 12V
- Jumpers
- 2 Motores DC com caixa de redução e rodas
- 1 Ponte-H L298N
- 1 Sensor Ultrassônico HC-SR04
- 1 Roda boba
- 1 Chassi

### Metodologia

O Robô Desviador de Obstáculo projetado anda em linha reta e verifica se há objetos a 40cm de distância e, em caso afirmativo, ele volta para trás e, em seguida, desvia para a direita por um determinado tempo, voltando a andar em linha reta caso não encontre nenhum outro obstáculo no caminho. O circuito montado pode ser visto na Figura 1. No diagrama está sendo utilizado o Arduino NANO, porém o mesmo pode ser substituído pelo Arduino UNO mantendo a mesma pinagem.



**Figura 1:** Diagrama da Montagem

O sensor ultrassônico envia uma onda sonora de alta frequência, de forma semelhante ao sonar dos morcegos. O objeto a ser detectado, que reflete essa onda, produz um eco que é

## II MATERIAIS E MÉTODOS

transformado em um sinal a ser lido pelo Arduino.

No código do Arduino, utilizando a biblioteca Ultrasonic.h, podemos obter a distância entre o objeto e o Robô convertendo o sinal de entrada recebido pelo sensor ultrassônico. Caso essa distância seja maior do que 40cm, o carrinho anda para a frente em linha reta. Caso contrário, o carrinho anda para trás em linha reta por um determinado tempo e, em seguida, desvia para a direita.

A Ponte-H tem como função controlar a velocidade e o sentido de rotação das rodas. Dessa forma, para que o Robô ande para a frente em linha reta, o Arduino deve enviar um sinal para a Ponte-H de maneira a permitir que ambas as rodas girem para frente na mesma velocidade. Em contrapartida, para que ele ande para trás em linha reta, as rodas devem girar com a mesma velocidade, porém ambas para trás. Finalmente, para que o Robô desvie para a direita, é necessário que as rodas girem na mesma velocidade, porém a roda esquerda deve girar para a frente e a direita para trás. É importante ressaltar que as rodas nem sempre giram com a mesma velocidade, sendo necessário realizar testes para ajustar as velocidades definidas no código.

### Código

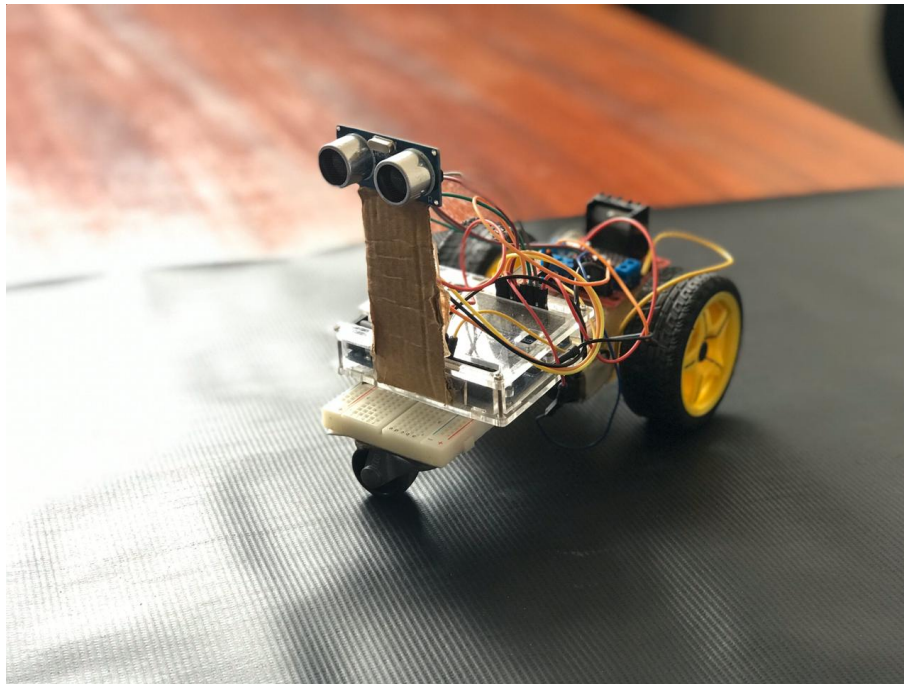
```
1
2 //Incluindo biblioteca Ultrasonic.h
3 #include "Ultrasonic.h"
4
5 //Criando objeto ultrasonic e definindo as portas digitais
6 //do Trigger - 9 - e Echo - 10
7 Ultrasonic SensorUltrassonico1(9, 10);
8
9 long Microsegundos = 0;// Variável para armazenar o valor do tempo da reflexão
10 //do som refletido
11 //pelo objeto fornecido pela biblioteca do sensor
12 float DistanciaemCM = 0;// Variável para armazenar o valor da distância a ser
13 //convertido por uma função da própria biblioteca do sensor
14
15 #define MotorLadoEsquerdo1 7
16 #define MotorLadoEsquerdo2 8
17 #define MotorLadoDireito1 4
18 #define MotorLadoDireito2 5
19 #define VelocidadeMotorLadoEsquerdo 6
20 #define VelocidadeMotorLadoDireito 3
21
22 //Escolhe a velocidade dos motores
23 int ValorVelocidadeMotorLadoEsquerdo = 100;
24 int ValorVelocidadeMotorLadoDireito = 140;
25
26 void setup() {
27 //Definições de entrada e saída
28 pinMode(MotorLadoEsquerdo1, OUTPUT);
29 pinMode(MotorLadoEsquerdo2, OUTPUT);
30 pinMode(MotorLadoDireito1, OUTPUT);
```

```
29 pinMode(MotorLadoDireito2, OUTPUT);
30
31 // Inicia a comunicação serial com velocidade de 115200 bits por segundo
32 Serial.begin(115200);
33
34 // Tempo de espera para inicialização (para dar tempo de por o robô no chão)
35 delay(3000);
36 }
37
38 void loop() {
39 //Convertendo a distância em CM e lendo o sensor
40 DistanciaemCM = SensorUltrassonico1.convert(SensorUltrassonico1.timing(),
41       Ultrasonic::CM);
42 Serial.print(DistanciaemCM);
43 Serial.println(" cm");
44
45 if (DistanciaemCM <= 40) { // Se a distância lida pelo sensor for menor ou igual
46       que 40 centímetros
47 //Velocidade motor lado esquerdo
48 analogWrite( VelocidadeMotorLadoEsquerdo, ValorVelocidadeMotorLadoEsquerdo);
49 //Velocidade motor lado direito
50 analogWrite( VelocidadeMotorLadoDireito, ValorVelocidadeMotorLadoDireito);
51 // Motor lado esquerdo para trás
52 digitalWrite(MotorLadoEsquerdo1, HIGH);
53 digitalWrite(MotorLadoEsquerdo2, LOW);
54 // Motor lado direito para trás
55 digitalWrite(MotorLadoDireito1, HIGH);
56 digitalWrite(MotorLadoDireito2, LOW);
57 delay(700); // Tempo que ficará indo para trás
58 // Motor lado esquerdo para frente
59 digitalWrite(MotorLadoEsquerdo1, LOW);
60 digitalWrite(MotorLadoEsquerdo2, HIGH);
61 // Motor lado direito para trás
62 digitalWrite(MotorLadoDireito1, HIGH);
63 digitalWrite(MotorLadoDireito2, LOW);
64 delay(200); // Tempo que ficará indo para o lado direito
65 }
66
67 else { // Se não, ou seja, se a distância for maior que 40 centímetros
68 //Velocidade motor lado esquerdo
69 analogWrite( VelocidadeMotorLadoEsquerdo, ValorVelocidadeMotorLadoEsquerdo);
70 //Velocidade motor lado direito
71 analogWrite( VelocidadeMotorLadoDireito, ValorVelocidadeMotorLadoDireito);
72 // Motor lado esquerdo para frente
73 digitalWrite(MotorLadoEsquerdo1, LOW);
74 digitalWrite(MotorLadoEsquerdo2, HIGH);
75 // Motor lado direito para frente
76 digitalWrite(MotorLadoDireito1, LOW);
77 digitalWrite(MotorLadoDireito2, HIGH);
78 }
79 }
```

## IV DISCUSSÃO

### III. RESULTADOS

Os resultados apresentados pelo projeto foram satisfatórios e cumpriram os pré requisitos determinados, ou seja, o Robô desvia do obstáculo a uma distância desejada de 40cm e possui uma velocidade equivalente entre as duas rodas, permitindo-o seguir em linha reta ou virar para a direita com eficiência.



**Figura 2:** Robô Desviador de Obstáculos

Por ser um protótipo com baixo nível de complexidade, o Robô Desviador de Obstáculos mostrou ter excelentes fins didáticos. Foi proposta aos alunos do EsColab a mudança do código afim de cumprir alguns desafios, sendo eles: aumentar ou diminuir a distância de percepção do obstáculo, fazer com que o robô parasse quando encontrasse um obstáculo e aumentar a velocidade das rodas.

### IV. DISCUSSÃO

Dependendo do posicionamento do Sensor Ultrassônico, o Robô pode ter problemas para enxergar os obstáculos, uma vez que ele pode acabar identificando o chão. Uma solução para esse problema pode ser a construção de um chassi e suporte do sensor mais robustos.

Além disso, como já dito anteriormente, faz-se necessário o ajuste dos valores das velocidades das rodas, uma vez que, na maioria das vezes, elas não terão a mesma velocidade indicada no código.

REFERÊNCIAS

Como Fazer um Robô que Desvia de Obstáculos com Arduino e Sensor Ultrassônico. Disponível em: <<https://www.marlonnardi.com/p/universo-robos-1-como-fazer-um-robo-que.html>>