

# Introdução à Robótica com Arduino

## Desvio de Obstáculos

Michelle Mendes Santos

**MANUAL DO  
PROFESSOR**



LABORATÓRIO DE **INOVAÇÃO** COLABORATIVO



LABORATÓRIO DE **INOVAÇÃO** COLABORATIVO

Michelle Mendes Santos

# **Introdução à Robótica com Arduino**

## **Desvio de Obstáculos**

1ª Edição

Belo Horizonte  
Instituto Federal de Minas Gerais

2022

© 2022 by Instituto Federal de Minas Gerais

Todos os direitos autorais reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico. Incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização por escrito do Instituto Federal de Minas Gerais.

Pró-reitor de Pesquisa	Fernando Gomes Braga
Coordenador de Inovação	Norimar de Melo Verticchio
Autor do curso	Michelle Mendes Santos
Arte gráfica	Ângela Bacon
Diagramação	Eduardo dos Santos Oliveira

#### FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Índice para catálogo sistemático:

1. Não preencher

Não preencher

2022

Direitos exclusivos cedidos à  
Instituto Federal de Minas Gerais  
Avenida Mário Werneck, 2590,  
CEP: 30575-180, Buritis, Belo Horizonte – MG,  
Telefone: (31) 2513-5157

## Sobre o material

Este curso é autoexplicativo e não possui tutoria. O material didático, incluindo suas videoaulas, foi projetado para que você consiga desenvolver e ministrar esse curso de forma autônoma e suficiente.

Caso opte por imprimir este *e-book*, você não perderá a possibilidade de acessar os materiais multimídia e complementares. Os *links* podem ser acessados usando o seu celular, por meio do glossário de Códigos QR disponível no fim deste manual.

Embora o material passe por revisão, somos gratos em receber suas sugestões para possíveis correções (erros ortográficos, conceituais, *links* inativos etc.). A sua participação é muito importante para a nossa constante melhoria. Para relatar qualquer erro ou sugestão de melhoria, envie um e-mail para [nit@ifmg.edu.br](mailto:nit@ifmg.edu.br), com o seguinte assunto: MELHORIA NO MATERIAL DO LIC

Para saber mais sobre a Inovação do IFMG acesse

[www.ifmg.edu.br](http://www.ifmg.edu.br)



## Apresentação dos Ícones

Os ícones são elementos gráficos para facilitar os estudos, fique atento quando eles aparecem no texto. Veja aqui o seu significado:



**Atenção:** indica pontos de maior importância na atividade e que exige mais atenção do aluno e instrutor.



**Dica do professor:** novas informações ou curiosidades relacionadas ao tema em estudo.



**Mídia digital:** sugestão de recursos audiovisuais para enriquecer a aprendizagem.





## 1.1. O que é a Robótica?

Robótica é um ramo da ciência que engloba mecânica, eletroeletrônica e computação, responsável pelo estudo e pela construção de sistemas capazes de realizar tarefas de maneira autônoma (seguindo uma programação) ou o controle humano. A Robótica refere-se ao estudo dos robôs.

A Robótica se divide em duas grandes áreas: Robótica de Manipuladores e Robótica Móvel. Os manipuladores robóticos são robôs que, normalmente, são fixos em um ponto e, por meio de seus elos e juntas, podem se movimentar para alcançar objetos em torno de si e realizar tarefas específicas em um ambiente estruturado. A Figura 1 apresenta exemplos de robôs manipuladores.



Figura 1: Exemplos de Manipuladores Robóticos

Fonte: <http://www.autoentusiastasclassic.com.br/2013/02/as-leis-da-robotica-e-o-carro-autonomo.html>

A Robótica Móvel se destina ao estudo dos robôs que podem se locomover no ambiente, seja ele externo ou interno. Os robôs móveis podem ser terrestres, aéreos, ou aquáticos e, para cada ambiente, ele deve contar com um sistema de locomoção adequado a esse ambiente. A figura 2 apresenta alguns exemplos de robôs móveis.



Figura 2: Exemplos de Robôs Móveis

Fonte: <http://graduacao.ufabc.edu.br/eiar/conteudo/ensino/disciplinas/Robotica/FundamentosRobotica.html>

O robô que será desenvolvido nesta atividade é um robô móvel terrestre e deverá realizar sua tarefa sem a intervenção humana, seguindo uma programação prévia. Esse tipo de robô é conhecido como robô autônomo.

Você já parou para pensar por que os robôs funcionam? Para que um veículo seja considerado um robô, ele deve possuir alguns sistemas que irão conferir a ele diversas habilidades. Um robô móvel autônomo deve possuir, pelo menos um sistema de locomoção, sensores para conhecer o ambiente e seu próprio estado e um sistema de processamento das informações, que irá interagir com seus componentes e tomar a decisão mais acertada de acordo com sua programação. Para funcionar de maneira autônoma, o robô precisa ser equipado com Sensores, que enviarão informações a um Controlador; este processará a lógica de programação e tomará decisões, enviando comandos aos Atuadores, que farão com que as ações sejam executadas, como seguir em frente, ou virar, por exemplo. Esse esquema pode ser observado na figura 3.

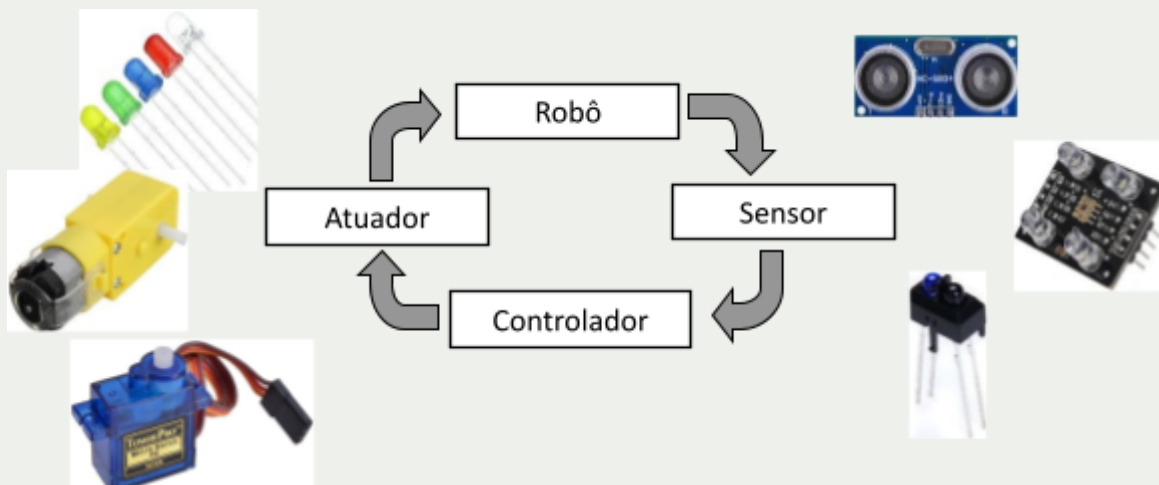
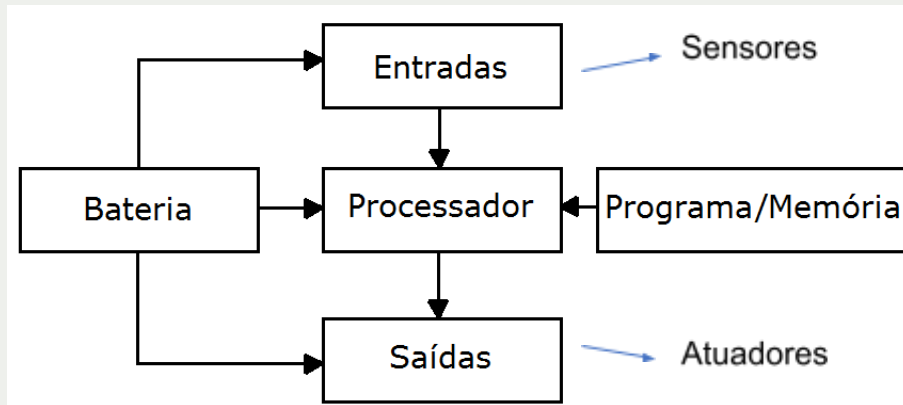


Figura 3: Esquema básico de funcionamento de um robô autônomo

Fonte: Autoria própria

É importante lembrar que, para que todos esses componentes possam funcionar corretamente, é necessário que exista uma fonte de energia. No caso de robôs móveis, a

energia deve ser fornecida por meio de baterias, uma vez que eles não podem estar



presos a fios. A figura 4 apresenta um esquema dos principais componentes dos robôs.

Figura 4: Principais componentes de um robô autônomo  
Fonte: Autoria própria

## 1.2. Situação problema

Você já deve ter visto ou se deparado com um robô aspirador de pó, que faz uma trajetória dentro de um ambiente fechado e vai sugando a sujeira para o seu interior. Você também já deve ter assistido a algum vídeo que apresenta carros que “dirigem sozinhos” não é mesmo? São os “carros autônomos”. Já parou para pensar em como esse tipo de robô (autônomo) se locomove dentro dos espaços e, ao encontrar um obstáculo, como um móvel da casa, uma parede, ou até um pedestre, ele muda de direção e continua realizando sua tarefa? Para isso, esse tipo de robô precisa de sensores, que detectam a presença de um obstáculo à sua frente, e de uma lógica de programação, que faz com que ele desvie sua trajetória sempre que encontrar esse obstáculo.

Na atividade prática desta oficina, os estudantes deverão entender o funcionamento de um sensor ultrassônico e como ele detecta a presença de obstáculos à sua frente. Eles devem compreender sua conexão com o controlador Arduino e elaborar uma lógica de controle que comanda as rodas do robô a fim de andar para frente até que algum obstáculo esteja a uma distância pré-determinada e, então, girar no sentido horário ou anti-horário, desviando do mesmo.

## 1.3. Questão motriz






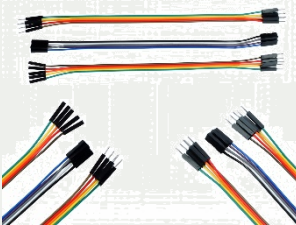
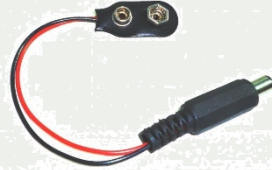

Ao final desta atividade, os estudantes serão capazes de responder à pergunta: **“Como um robô autônomo pode identificar e desviar de um obstáculo em seu caminho?”** e ainda vivenciar na prática os desafios da robótica móvel e dos robôs autônomos.



**Mídia digital:** Introdução à atividade proposta. Os vídeos que serão introduzidos no material, serão gravados durante a aplicação da atividade piloto

# Materiais necessários

Para a realização desta atividade, são necessários os materiais listados na tabela a seguir:

Item	Nome do item	Descrição	Quant.	Foto	Substituição
1	Kit Chassi 2WD ou 4WD	Kit de Chassi em acrílico, MDF ou similar, rodas e motores, parafusos e suportes para montagem do robô	1		
2	Sensor Ultrassônico	Sensor Ultrassônico para Arduino HC-SR04	1		Não há substituto
3	Arduino UNO	Placa Controladora Arduino UNO ou compatível com cabo USB	1		Arduino Mega
4	Bateria 9V	Bateria 9V, preferencialmente recarregável e previamente recarregada	1		Bateria 9V não-recarregável
5	Jumpers	Conjunto de jumpers Macho-fêmea e Macho-macho	1		Pequenos fios feitos com cabos de rede, com as pontas decapadas
6	Clipe Bateria 9V	Clipe para bateria de 9V com pino P4	1		Não há substituto
7	Suporte Sensor Ultrassônico	Suporte em acrílico, MDF, impressão 3D ou similar para encaixar o sensor ultrassônico	1		Não há substituto

8	Módulo Ponte H	Driver L298n de Ponte H dupla	1		Módulo Ponte H Dupla HG7881
---	----------------	-------------------------------	---	--	-----------------------------



**Atenção:** O instrutor da atividade deve montar previamente o chassi do robô, sua estrutura, soldar fios nos motores. Deve, também, antes de cada execução da atividade, conferir o funcionamento de todos os componentes e recarregar as baterias, caso sejam recarregáveis ou verificar a carga de baterias não recarregáveis. Mais detalhes da montagem estão no capítulo 3.



**Mídia digital:** Apresentação dos materiais utilizados na atividade.

## 2.1 O que é o Arduino

Segundo o site oficial do projeto, “Arduino é uma plataforma open-source de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobistas e qualquer pessoa interessada em criar objetos ou ambientes interativos.” (Arduino, 2022) ou seja, o Arduino é uma plataforma formada por dois componentes: A placa, que é o Hardware usado para construir os projetos e a IDE Arduino, que é o Software utilizado para programar o que se deseja que a placa faça.

A placa Arduino a ser utilizada nesta oficina é o UNO, que é o modelo mais utilizado na robótica educacional (figura 5). Ela possui um Microcontrolador, que é o cérebro da placa – é onde a lógica de programação será processada para que as decisões sejam tomadas automaticamente. Possui também pinos digitais e analógicos, que possibilitam a conexão dos sensores e atuadores à placa. É possível utilizar seus pinos de alimentação para fornecer energia aos sensores e atuadores que serão ligados a ela. Para realizar a programação, pode-se utilizar seu conector USB que, por meio de um cabo, será conectado a um computador e a programação será enviada para ser armazenada em sua memória. Esse conector também é capaz de alimentar os componentes da placa, quando conectado a um computador. Porém, quando se necessita de mobilidade, é preciso desconectar a placa do computador e utilizar uma alimentação independente. Para

alimentar o Arduino externamente, existe um conector no modelo P4 para fornecimento de energia.

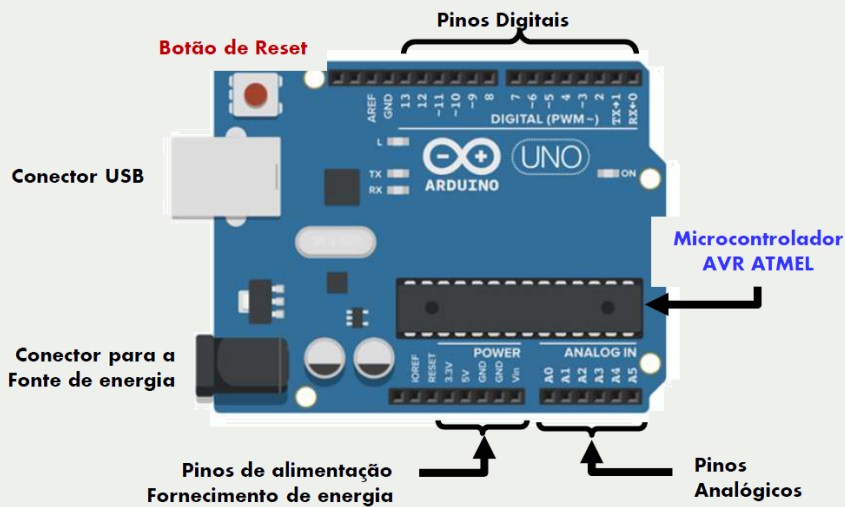


Figura 5: Principais componentes de uma placa Arduino UNO  
 Fonte: Autoria própria

Pra realizar a programação da placa, é necessário utilizar o software Arduino IDE, disponível para download gratuito no site oficial do projeto (<https://www.arduino.cc/>). A interface do Arduino IDE pode ser vista na Figura 6.



Figura 6: Interface do Software Arduino IDE.  
 Fonte: <http://www.professorakeila.com.br/2018/04/entendendo-o-arduino-ide.html>

Nesse software, deve-se escrever a programação, basicamente, em três blocos. O primeiro bloco é aquele em que se escreve aqueles comandos que não ficam dentro de nenhuma função – são definições, configurações, inclusão de bibliotecas, etc. O segundo bloco é a função *void setup*. Nela são definidos os pinos do Arduino que serão utilizados e como serão utilizados (se como entradas ou saídas). Também são escritos nesse bloco aqueles comandos que devem ser executados apenas uma vez ao início da execução do programa. O terceiro bloco é a função *void loop*. Nela são inseridos os comandos que serão executados como um loop, ou seja, continuamente a partir do momento em que a lógica for carregada para a placa. A figura 7 apresenta um exemplo de programação contendo esses três blocos.

<pre> 1  #include &lt;Modbusino.h&gt; 2  ModbusinoSlave modbusino_slave(1); 3  uint16_t tab_reg[1]; 4  #define LED 9 5 6 7  void setup() { 8      // put your setup code here, to run once: 9      modbusino_slave.setup(9600); 10     pinMode(LED, OUTPUT); 11 } 12 13 void loop() { 14     // put your main code here, to run repeatedly: 15     tab_reg[0] = 1; 16     digitalWrite(LED, HIGH); 17     modbusino_slave.loop(tab_reg, 1); 18 } </pre>	<p><b>Bloco 1 - definições</b></p> <p><b>Bloco 2 - configurações iniciais e comandos executados 1 vez</b></p> <p><b>Bloco 3 - comandos a serem repetidos</b></p>
---	--

Figura 7: Exemplo de programa Arduino IDE  
Fonte: Autoria própria

Após realizar a programação na interface, deve-se selecionar o modelo de placa (Fig. 8) e a Porta Serial (Fig. 9), no menu Ferramentas (Tools). Deve-se, então, compilar o código para verificar se há erros de digitação e, caso esteja tudo correto, deve-se carregar o programa na placa.

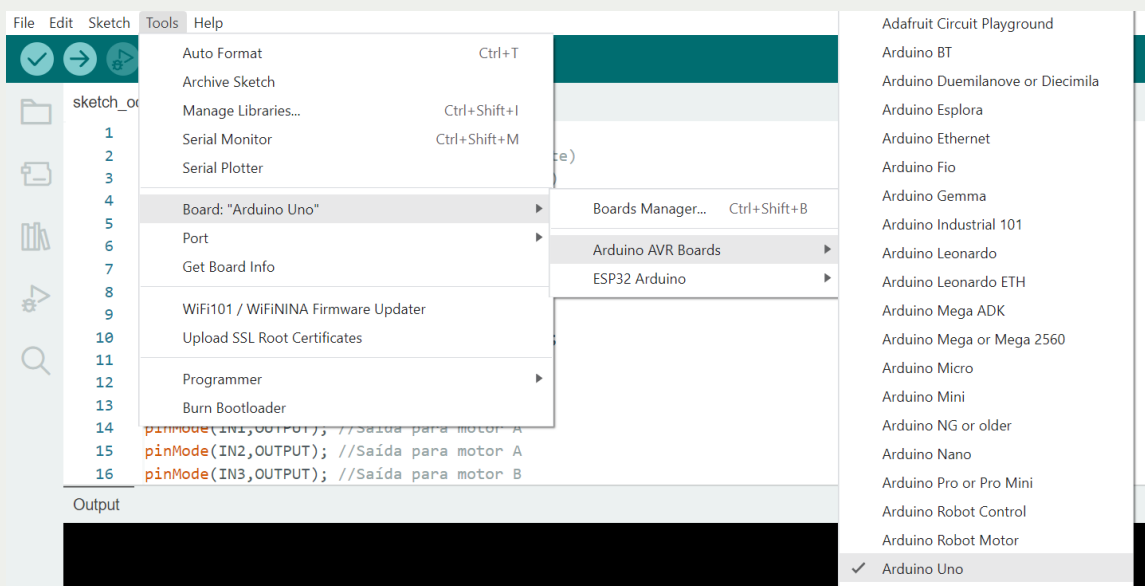




Figura 8: Seleção do Modelo de Placa Arduino UNO  
Fonte: Autoria própria

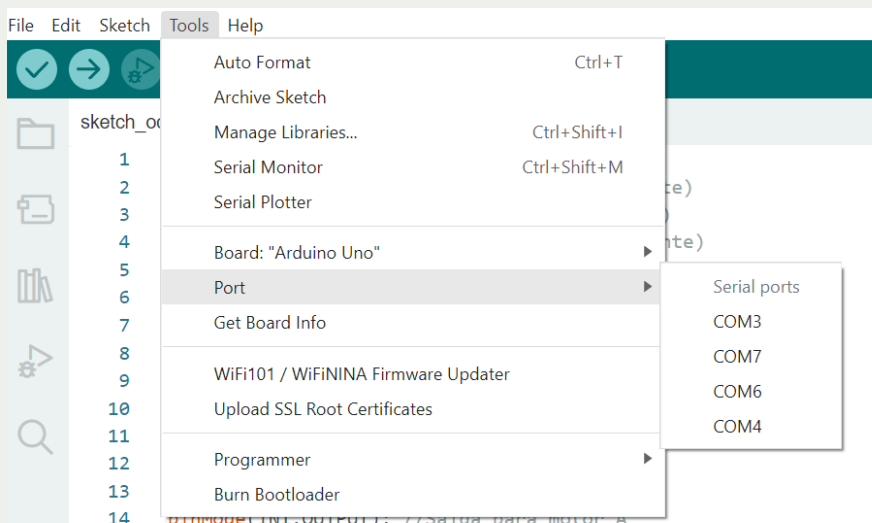


Figura 9: Seleção da Porta Serial  
Fonte: Autoria própria

Caso esteja interessado em se aprofundar um pouco mais no estudo do Arduino, existem diversos sites que trazem tutoriais e apostilas de como começar a programar nessa plataforma. Algumas referências podem ser encontradas em (Mota, 2021).



**Mídia digital:** Links de vídeos disponíveis na internet com uma introdução ao Arduino.

- Curso “Introdução ao Arduino, o básico para começar”, da UFRGS

(<https://lumina.ufrgs.br/course/view.php?id=30>)

- Canal “Brincando com Ideias”, no YouTube

(<https://www.youtube.com/c/BrincandocomIdeias>)

## 2.2 Módulo Ponte H

O acionamento de um motor de corrente contínua, ou motor cc, é feito de maneira muito simples: quando energizado com uma polaridade, o motor gira em um sentido e ao inverter a polaridade da alimentação, ele gira no sentido contrário. Dessa maneira, para fazer as rodas do robô girarem para frente e para trás, bastaria inverter a polaridade do motor ao longo do movimento, correto? Sim! Mas como trocar os fios do motor de lugar enquanto ele se movimenta? E como fazer isso automaticamente, sem realizar mudanças físicas no robô? É nessa tarefa que entra o módulo Ponte H.

Uma Ponte H funciona como um conjunto de chaves eletrônicas, que irão abrir ou fechar de acordo com um comando que o controlador irá enviar. Dessa maneira, o sentido do motor pode ser invertido pelo acionamento dessas chaves. A figura 10 apresenta o circuito que esquematiza o funcionamento de uma Ponte H. Nessa figura, pode-se observar que quando as chaves 1 e 4 estão conectadas, a alimentação positiva está do lado esquerdo do motor e a negativa está do lado direito. Quando as chaves 2 e 3 estão fechadas, a alimentação vista pelo motor está invertida, com o pólo positivo do lado direito do motor e o pólo negativo do lado esquerdo. Dessa forma, pode-se inverter o sentido de giro do motor apenas abrindo e fechando as chaves corretas.

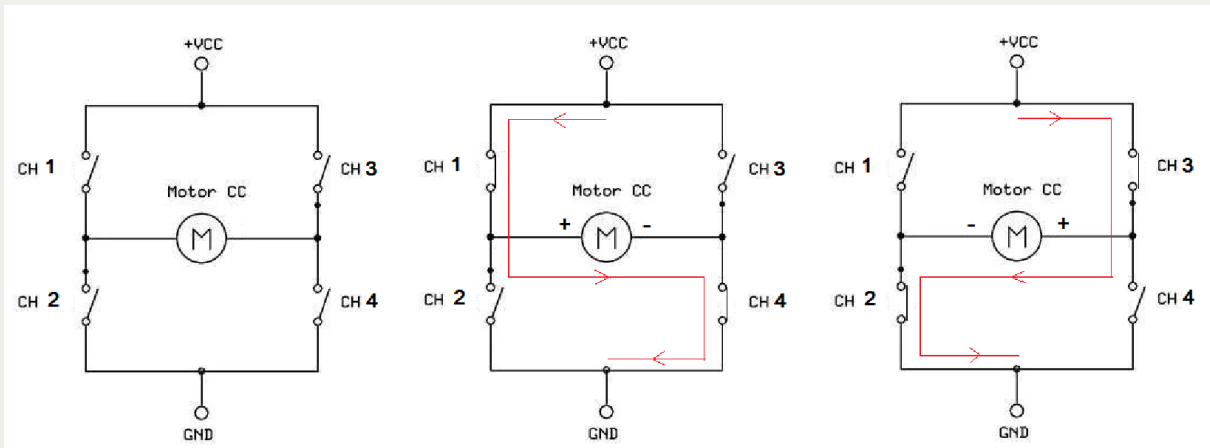


Figura 10: Circuito esquemático do funcionamento da ponte H.  
Fonte: Autoria própria

O módulo Ponte H dupla utilizado nesta aplicação possui dois circuitos de chaves eletrônicas conectadas em Ponte H, o que significa que se pode acionar dois motores e inverter suas polaridades com apenas um módulo. A figura 11 apresenta uma foto do Módulo Ponte H utilizado nesta atividade.

Nesse módulo, é necessário conectar os motores do robô aos bornes destacados como “MOTOR A” e “MOTOR B”, ligar a alimentação nos bornes correspondentes, dependendo de seu valor, e conectar as entradas do módulo aos pinos digitais do Arduino.

O detalhamento das conexões necessárias nesta atividade são encontrados nas seções 3.1 e 3.2 deste caderno.

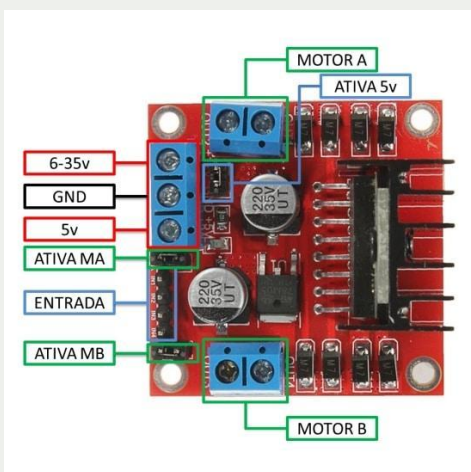


Figura 11: Circuito esquemático do funcionamento da ponte H.  
Fonte: <https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/>

## 3.1 Pré-montagem

Para realizar esta atividade, o instrutor da oficina deve realizar a montagem da estrutura do robô e oferecer aos estudantes o hardware completo, de maneira que a tarefa tenha foco apenas na conexão dos jumpers e programação.

Para iniciar a montagem, verifique se os motores a serem utilizados já possuem fios soldados em seus terminais. Caso contrário, corte e desencape 2 pares de fios (vermelho e preto) de aproximadamente 20 cm e solde aos terminais dos motores, conforme a figura 12.

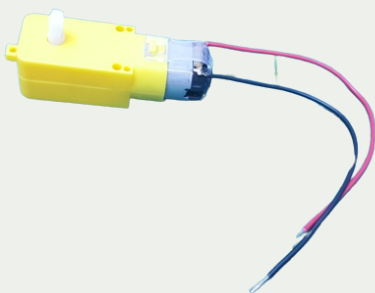


Figura 12: Motor com os fios soldados em seus terminais  
Fonte: Autoria própria.

Em seguida, verifique se o clipe de bateria 9V disponível já possui pino P4 para conexão com o Arduino e se o mesmo possui uma derivação para alimentação da Ponte H. Se não possuir pino P4 ou se não houver a derivação, providencie essas duas conexões, para que o clipe fique conforme a Figura 13.

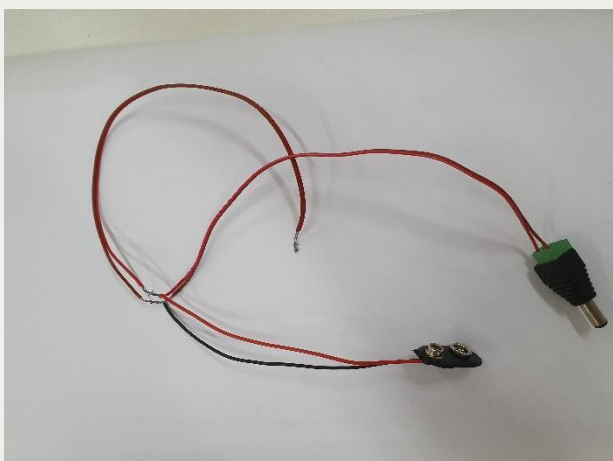


Figura 13: Clipe de bateria 9V preparado para a atividade.  
Fonte: Autoria própria



**Atenção:** Não se esqueça de utilizar fita isolante ou espaguete termo-retrátil para isolar os terminais e fios expostos.

Fixe os motores no chassi, de acordo com o modelo disponível no LIC (2WD ou 4WD) e encaixe as rodas nos eixos dos motores. A Figura 14 apresenta a montagem de um chassi 4WD com 2 andares (duas placas de acrílico para se dispor os componentes). Para esta atividade, apenas um andar é necessário, pois a quantidade de componentes utilizada é pequena, mas você pode optar por montar as duas placas, dando maior estabilidade ao conjunto.

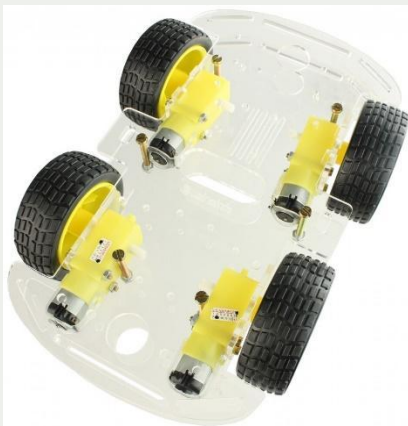


Figura 14: Montagem de um chassi 4WD.

Fonte:

<http://www.usinainfo.com.br/kit-robotica/carrinho-arduino-carro-robo-4wd-80rpm-acrilico-3mm-kit-chassi-2296.html>

Após a montagem do chassi, posicione os seguintes componentes sobre ele: o Arduino UNO, o módulo Ponte H e o suporte para o sensor ultrassônico, utilizando parafusos, elásticos, abraçadeiras plásticas ou material similar que permita que os componentes fiquem presos ao chassi, mas possibilite sua remoção, quando for necessário.

Conecte os fios da derivação realizada no clipe de bateria aos pinos de alimentação do módulo Ponte H, conforme a figura 15.

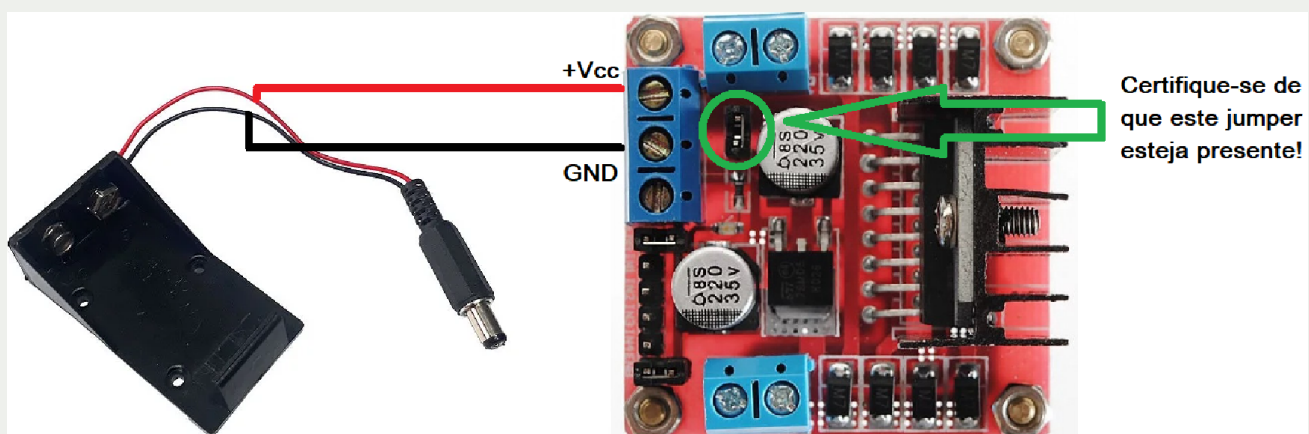


Figura 15: Alimentação do Módulo Ponte H.  
Fonte: Autoria própria

Após a conexão da alimentação, você deve conectar os motores aos bornes laterais do módulo Ponte H. Lembre-se de que, caso o chassi seja do modelo 4WD, ele deve ter os dois motores que se encontram do mesmo lado conectados ao mesmo borne da Ponte H, ou seja, cada borneira da Ponte H vai acionar 2 motores simultaneamente, conforme apresentado na Figura 16.

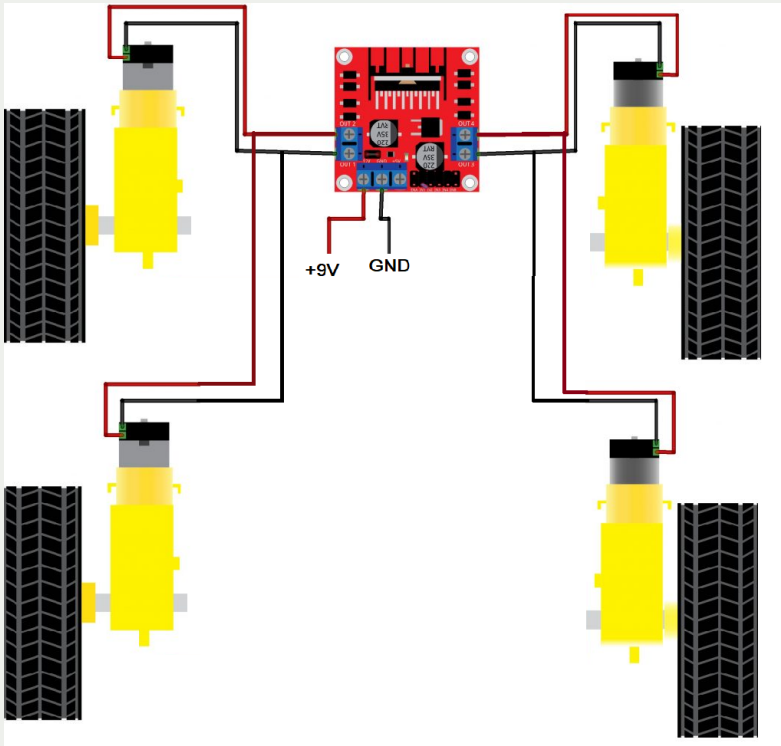


Figura 16: Conexão dos motores.  
Fonte: Autoria própria

Após finalizada esta etapa, o chassi está pronto para utilização na atividade da oficina. A partir deste ponto, o restante da conexão dos fios aos sensores, Arduino e pinos de comando deve ser realizada pelos estudantes.

### 3.2 Conexão física dos componentes

Para dar início à conexão dos componentes do robô, o instrutor deve introduzir uma explicação sobre o funcionamento do Arduino, do Sensor Ultrassônico, e do Módulo Ponte H, principais componentes desta atividade. Ele deve utilizar como base as seções 2.1 a 2.3 para conduzir essa explicação.

Para a parte prática, o estudante deve seguir os passos descritos nesta seção.

**1** – Com a utilização de jumpers macho-fêmea, conectar os pinos do Arduino aos pinos do sensor ultrassônico, conforme a Figura 17.

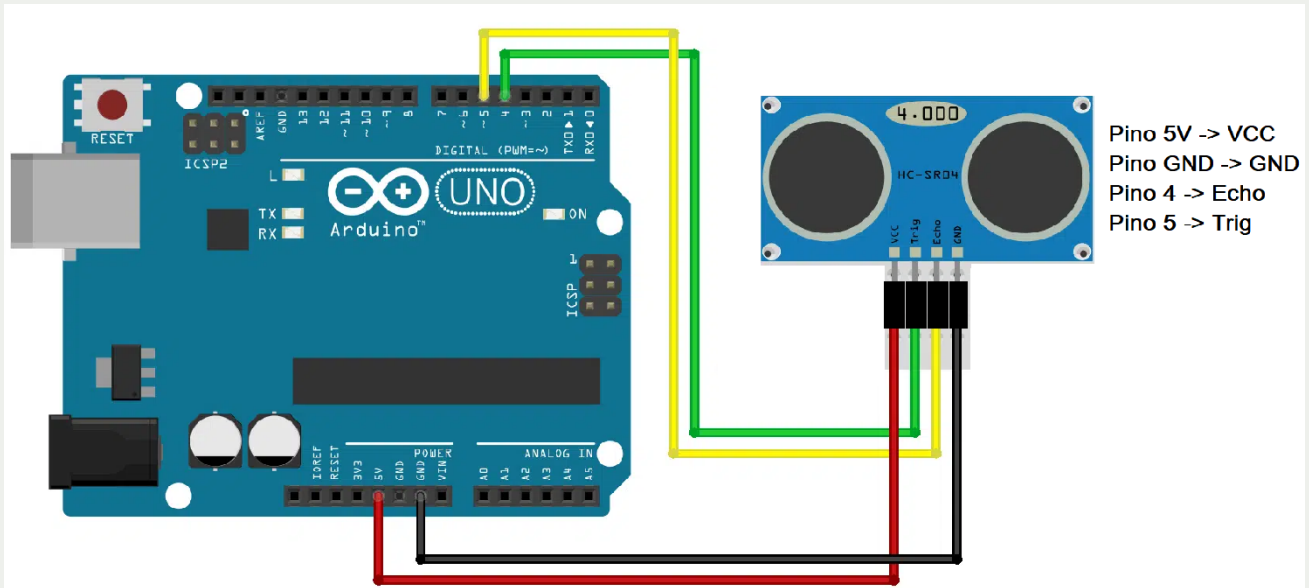


Figura 17: Conexão dos motores.

Fonte: Adaptado de <http://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>

2- Utilizando jumpers, conectar os pinos do módulo Ponte H ao Arduino de acordo com a Figura 18 e a tabela a seguir:

Pino no Arduino	Pino no Módulo Ponte H	Comando dos motores
Digital 6	IN1	Gira Motores da Direita para frente
Digital 7	IN2	Gira Motores da Direita para trás
Digital 8	IN3	Gira Motores da Esquerda para frente
Digital 9	IN4	Gira Motores da Esquerda para trás
Digital 10	ENA	Controla velocidade dos motores da direita
Digital 11	ENB	Controla velocidade dos motores da esquerda

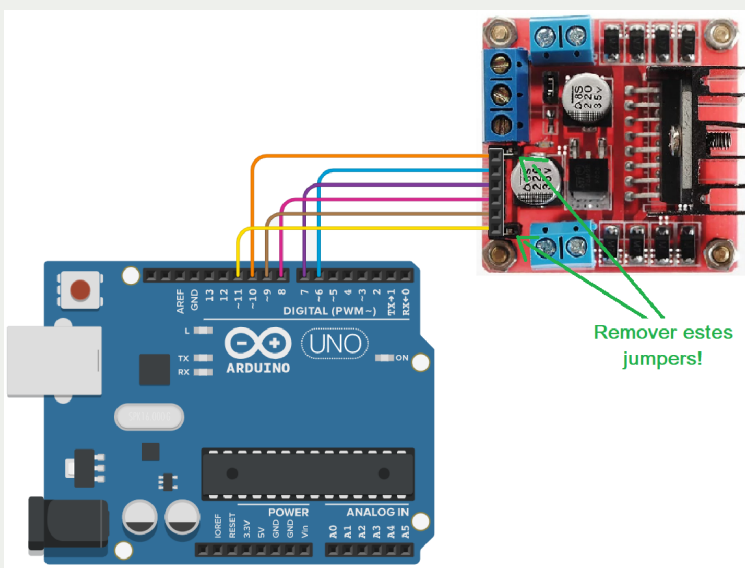


Figura 18: Conexão entre Arduino e Módulo Ponte H.

Fonte: Autoria própria

Concluída a montagem e conexão das partes, o próximo passo é a programação do robô.

### 3.3 Programação do Controlador Arduino – Experimento 1 – Ligar os motores

Para realizar esta atividade, o estudante irá programar utilizando o Arduino IDE, software gratuito para programação do Arduino. Relembre sua interface apresentada no capítulo 2, na Figura 6.

Inicialmente, o estudante deve ser orientado a fazer um programa mais simples, em que ligue os motores para frente o tempo todo. Para isso, ele deve:

**1- Definir os pinos que serão utilizados para acionamento dos motores**

```
#define IN1 6 //Pino motor A (Direita - Frente)
#define IN2 7 //Pino motor A (Direita - Trás)
#define IN3 8 //Pino motor B (Esquerda - Frente)
#define IN4 9 //Pino motor B (Esquerda - Trás)
```

**2- Definir os pinos de controle de velocidade dos motores**

```
#define ENA 10 //Pino velocidade motor A (Enable A)
#define ENB 11 //Pino velocidade motor B (Enable B)
```

**3- Configurar a função dos pinos como SAÍDAS na função “void setup”**

```
void setup() {
  pinMode(IN1,OUTPUT); //Saída para motor A
  pinMode(IN2,OUTPUT); //Saída para motor A
  pinMode(IN3,OUTPUT); //Saída para motor B
  pinMode(IN4,OUTPUT); //Saída para motor B
  pinMode(ENA,OUTPUT); //Controle velocidade motor A
  pinMode(ENB,OUTPUT); //Controle velocidade motor B
```

**4- Ainda dentro da função setup, deve-se configurar a velocidade do motor, pois isso pode ser executado apenas uma vez. Dessa maneira, a escolha da velocidade que o motor vai girar não precisa ser alterada o tempo todo. Sugere-se o valor 100 para a velocidade neste experimento, pois dessa maneira, os motores devem receber um valor de tensão de aproximadamente 3,5V, que está acima do valor mínimo (3V) e é uma velocidade baixa para começar os testes.**

```
analogWrite(ENA,100); //Controle PWM do motor A (0 a 255)
analogWrite(ENB,100); //Controle PWM do motor A (0 a 255)
delay(1000); //Aguarda 1s antes de iniciar
} //fim do void setup
```

**5- Após realizadas as configurações, deve-se escrever o código da função “void loop”, que será executada indefinidamente, até que o Arduino seja desligado. Para que os motores sejam ligados para frente, deve-se acionar os pinos IN1 e IN3 em nível lógico ALTO e desligar os pinos IN2 e IN4, conforme a lógica a seguir:**

```
void loop() //loop principal
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
```



}

- 6- Finalizada a programação, o código deve ser compilado e enviado ao Arduino. Mas antes, é importante selecionar o modelo da placa e a porta serial, no menu “Ferramentas”. Para compilar e gravar, clique no botão “Verificar” e depois “Carregar”.



**Atenção:** Todos os procedimentos listados neste tópico devem ser executados com a bateria desconectada do Arduino e sua alimentação será recebida pela porta USB do computador. Assim, oriente aos estudantes que mantenham o pino P4 do clipe de bateria desconectado do Arduino. O mesmo só deverá ser conectado após desconectar o cabo USB, quando forem realizados testes longe do computador.

Após concluir a programação e carregar o programa no Arduino, os motores já devem ligar de maneira que as rodas se movimentem para frente. Caso as rodas estejam se movimentando em sentido diferente ou para trás, inverta a conexão no módulo Ponte H do motor que está girando no sentido contrário.

Com os motores testados e a compreensão de como eles são acionados no código, é possível explicar aos estudantes o conceito de funções e deixar o código mais elegante, transportando o código que está no loop principal para uma função chamada “frente”. Dessa maneira, na função principal haverá apenas a chamada à função “frente”. O código ficará como se segue:

```
void loop() //loop principal
{
  frente();
}

void frente() //função para mover o robô para frente
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
} //fim da função frente
```

Após realizar essas mudanças, o programa deve ser novamente compilado e carregado no Arduino.

Experimente, agora, definir uma função para parar o robô, enviando o nível lógico baixo para as 4 saídas digitais que comandam os motores.

```
void parar() //função para mover o robô para frente
{
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);
}
```

```
} //fim da função parar
```

No loop principal, faça com que o robô ande para frente por 1 segundo e pare por 1 segundo repetidamente.

```
void loop() //loop principal
{
    frente();
    delay(1000);
    parar();
    delay(1000);
}
```

### 3.4 Programação do Controlador Arduino – Experimento 2 - Girar

Agora que o robô já possui uma função para seguir em frente, o estudante está apto a desenvolver funções para fazer o robô girar para a esquerda e para a direita. Nesta parte da atividade, ele deve construir as funções que rotacionam os motores a fim de reproduzir esses movimentos.

Para girar para a direita, os motores da direita devem girar para trás e os motores da esquerda devem girar para frente, ou seja, as saídas IN2 e IN3 devem ser ativadas, enquanto que as saídas IN1 e IN4 devem ser desativadas. Assim, o instrutor deve orientar os estudantes a construírem uma função “direita”, como a seguir:

```
void direita() //função para girar o robô para direita
{
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
} //fim da função direita
```

Para girar para a esquerda, os motores da direita devem girar para frente e os motores da esquerda devem girar para trás, ou seja, as saídas IN1 e IN4 devem ser ativadas, enquanto que as saídas IN2 e IN3 devem ser desativadas. Assim, o instrutor deve orientar os estudantes a construírem uma função “esquerda”, como a seguir:

```
void esquerda() //função para girar o robô para esquerda
{
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,HIGH);

} //fim da função esquerda
```

Para testar as funções, o instrutor deve solicitar aos estudantes que façam a chamada das mesmas na função principal. Sugere-se realizar a sequência: frente, parar, direita, parar, esquerda, parar, cada uma sendo executada por um segundo. A lógica, então ficará:

```
void loop() //loop principal
```

```

{
  frente();
  delay(1000);
  parar();
  delay(1000);
  direita();
  delay(1000);
  parar();
  delay(1000);
  esquerda();
  delay(1000);
  parar();
  delay(1000);
}

```

Ao final desses dois experimentos, o estudante possui quase todo o ferramental necessário para desenvolver a lógica de desvio de obstáculos, restando apenas a utilização do sensor ultrassônico e a programação da tarefa de desvio, que serão realizados na próxima seção.

### 3.5 Programação do Controlador Arduino – Experimento Final – Desviar de obstáculos

Agora que o robô já possui as funções para se locomover em qualquer direção, é hora de programa-lo para detectar obstáculos. Essa detecção é feita por meio do sensor ultrassônico. Para utilizá-lo, a biblioteca Ultrasonic.zip (disponível em: <http://downloads.arduino.cc/libraries/github.com/ErickSimoies/Ultrasonic-3.0.0.zip>).

Portanto, deve-se baixar o arquivo e incluí-lo na IDE do Arduino. Para isso, abra a IDE do Arduino, clique em “Sketch” > “Incluir Biblioteca” > “Adicionar Biblioteca .ZIP” e selecione o arquivo “Ultrasonic.zip”.

A conexão do sensor ultrassônico foi feita aos pinos 4 e 5 do Arduino. Por isso, é preciso definir os pinos e a instância para esse sensor no início do código:

```

#define TRIGGER_PIN 4 //define o pino de trigger como sendo o pino 4
#define ECHO_PIN 5 //define o pino de echo como sendo o pino 5
Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN); //chama a biblioteca

```

Agora é o momento de escrever uma função que fará com que a leitura do sensor ultrassônico seja convertida em centímetros:

```

float distancia() //função que mede a distância em cm
{
  float distanciaObj; //declara a variável que vai receber a distância do
  sensor ao objeto
  distanciaObj = ultrasonic.read(); //faz a leitura da distância do
  sensor ao objeto

  return(distanciaObj); // retorna o valor em cm para a função;
} //fim da função distancia

```

Neste momento chegou a hora de reformular a função void loop para que o robô possa seguir sempre em frente e, ao se aproximar de um obstáculo a 20 centímetros, ele deverá parar e escolher um sentido para girar. Sugere-se que seja escolhido um sentido de giro para que o robô sempre se movimente após encontrar o obstáculo.

Uma possível solução para esse programa seria:

```
void loop()
{
  frente();
  float dist; //declara uma variável para receber a distância medida
  dist = distancia(); //chama a função "distancia" e armazena o valor na
                    //variável dist

  if(dist < 20)
  {
    parar();
    delay(500);
    esquerda();
    delay(400);
    parar();
    delay(500);
  }
  delay(100);
}
```

Após compilar e carregar essa solução, havendo tempo, pode-se orientar os estudantes a mudar o sentido de giro, a definir uma lógica para alternar o sentido a cada obstáculo encontrado ou utilizar um número aleatório para que a decisão sobre o sentido de giro seja aleatória a cada obstáculo encontrado.

O código final para essa programação fica como a seguir:

```
#include <Ultrasonic.h>
#define TRIGGER_PIN 4
#define ECHO_PIN 5
Ultrasonic ultrasonic(TRIGGER_PIN, ECHO_PIN);
#define IN1 6
#define IN2 7
#define IN3 8
#define IN4 9
#define ENA 10
#define ENB 11

void setup()
{
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  analogWrite(ENA, 120);
  analogWrite(ENB, 120);
  delay(1000);
}
```

```
} //end setup

void loop()
{
  frente();
  float dist;
  dist = distancia();
  if(dist < 20)
  {
    parar();
    delay(500);
    esquerda();
    delay(400);
    parar();
    delay(500);
  }
  delay(100);
}
float distancia()
{
  float distanciaObj;
  distanciaObj = ultrasonic.read
  return(distanciaObj);
}

void frente()
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
}

void direita()
{
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
}

void esquerda()
{
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
}

void parar()
{
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
}
```

```
digitalWrite(IN4,LOW);  
}
```

## 3.1 Problemas de Hardware

Neste capítulo o instrutor encontra possíveis problemas que podem ocorrer e quais as possíveis causas, bem como a possível solução. A apresentação desses problemas será realizada por meio da tabela a seguir.

Problema	Possível Causa	Como solucionar?
O Arduino não liga	Se conectado à bateria, ela pode estar descarregada	Recarregue ou substitua a bateria
	Se conectado à USB do computador, o cabo pode estar com defeito ou a porta pode estar danificada	Substitua o cabo e troque a porta USB no computador
Os motores não giram	Os fios conectados aos terminais dos motores podem estar frouxos	Confira se todos os fios conectados aos terminais dos motores estão bem firmes e, caso contrário, refaça a solda dos fios
	A bateria pode estar descarregada	Recarregue ou substitua a bateria
	Os jumpers do módulo Ponte H podem estar conectados de maneira incorreta	Revise a conexão dos cabos e jumpers de acordo com as orientações da seção 3.1
	O módulo Ponte H pode estar danificado	Substitua o módulo Ponte H por outro novo, refazendo as conexões
O sensor ultrassônico não retorna nenhum valor	Os jumpers podem estar mal conectados ou com mal contato	Verifique se os jumpers de conexão estão firmes e posicionados corretamente
	O sensor pode estar danificado	Substitua o sensor

## 3.2 Problemas de Software

Problema	Possível Causa	Como solucionar?
O Arduino IDE não abre	A instalação não foi bem sucedida	Realize o download da versão mais atual, desinstale a versão presente no computador e reinstale a versão baixada.
O programa não compila	O modelo de Arduino selecionado pode estar incorreto	Selecione o modelo de Arduino correto e refaça a compilação
	Pode haver algum erro de sintaxe	Revise o seu código procurando erros de digitação,

		falta de “;” ao final das linhas de comando, nomes de variáveis diferentes da declaração, etc.
Não é possível carregar o programa para a placa	O modelo de placa ou a porta COM podem estar incorretos	Vá ao menu Ferramentas e confira se a placa selecionada é a mesma que está conectada ao computador. Verifique também se a porta COM selecionada é a mesma que está conectada a placa
As rodas do robô giram em sentido oposto ao esperado	Os fios dos motores podem estar invertidos	Verifique se a ligação dos fios dos motores é idêntica à apresentada no material.
	Os jumpers que conectam os sinais da Ponte H ao Arduino podem estar trocados	Confira a pinagem do módulo e da placa e revise se os jumpers não estão trocados
	Os motores da esquerda e da direita podem estar conectados ao lado oposto do módulo ponte H	Certifique-se de que o Motor A seja o da direita e o Motor B seja o da esquerda no módulo.



**Mídia digital:** Apresentação dos principais erros e as possíveis soluções.



## Referências

ARDUINO, 2022. Site oficial do projeto Arduino. Disponível em: <http://www.arduino.cc/>. Acesso em out/2022.

MOTA, 2021 – O que é o Arduino e Como Funciona? Disponível em: <https://portal.vidadesilicio.com.br/>. Acesso em out/2022.



## Currículo do autor



Possui mestrado em Engenharia Elétrica pela Universidade Federal de Minas Gerais (UFMG), na linha de Pesquisa Visão Computacional e Robótica, especialização em Educação Profissional e Tecnológica pela Faculdade Unina e bacharelado em Engenharia de Controle e Automação pela UFMG. Atua em docência desde 2007 e é professora efetiva do Instituto Federal de Minas Gerais desde 2012, tendo atuado como docente no *Campus* Formiga (nos cursos de Engenharia Elétrica e Ciência da Computação), e atualmente é docente da área de Automação Industrial e Tecnologia da Informação no Campus Betim. Desenvolve projetos de pesquisa, ensino e extensão na área de Robótica Móvel, Robótica Educacional, Manipuladores Robóticos, Ensino de Programação e Inclusão

Digital. Atuou como Coordenadora de Curso no Curso Técnico Integrado em Automação Industrial no Campus Betim e coordena Projetos Integradores desde 2017.

Currículo Lattes: <http://lattes.cnpq.br/8358169414937606>

Feito por (professor-autor)	Data	Revisão de <i>layout</i>	Data	Versão
Michelle Mendes Santos	17/10/2022	Norimar de Melo Verticchio	02/12/2022	1.0



