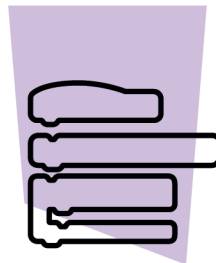
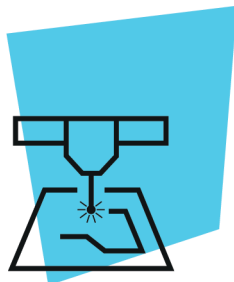
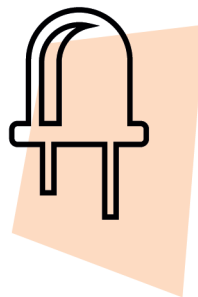
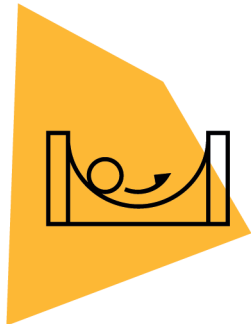
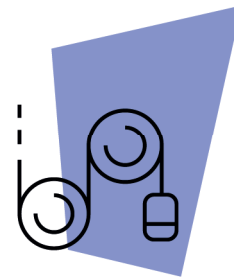


Diretoria de Tecnologia e Inovação

ROBÓTICA

Módulo 1



Kit Chassi 2WD Robô

AULA 32

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Cleres Rozeli Cristófolli Grande

Simone Sinara De Souza

Revisão Textual

Adilson Carlos Batista

Projeto Gráfico e Diagramação

Edna do Rocio Becker

2021



Este trabalho está licenciado com uma Licença Creative Commons
Atribuição NãoComercial - Compartilhável 4.0 Internacional

Aula 01	Por Que Robótica?
Aula 02	Tensão, Corrente e Resistência
Aula 03	Kit de Robótica
Aula 04	Arduino Uno R3
Aula 05	Softwares Arduino IDE e mBlock
Aula 06	Portas Digitais
Aula 07	Circuito Elétrico
Aula 08	LED e Resistor
Aula 09	Semáforo [Carros]
Aula 10	Semáforo [Cruzamento Carros]
Aula 11	Semáforo [Pedestres]
Aula 12	Semáforo [Cruzamento Carros + Pedestres]
Aula 13	Push Button
Aula 14	Feedbacks + Inventário I
Aula 15	Semáforo [Carros + Pedestres com Botão]
Aula 16	Display 7 Segmentos
Aula 17	Fonte DC + Plug P4
Aula 18	Portas PWM
Aula 19	LED Fade-In
Aula 20	LED Fade-Out
Aula 21	Super Máquina 80's
Aula 22	Super Máquina 2008
Aula 23	Potenciômetro
Aula 24	Buzzer Passivo
Aula 25	LED RGB
Aula 26	Arco-Íris
Aula 27	Sensor LDR
Aula 28	Feedbacks + Inventário II
Aula 29	Sensor de Temperatura
Aula 30	Sensor de Obstáculo IR
Aula 31	Controle Motor DC
Aula 32	Kit Chassi 2WD Robô
Aula 33	Seguidor de Linha
Aula 34	Sensor de Distância
Aula 35	Sensor de Estacionamento
Aula 36	Display LCD 16x2
Aula 37	Trena Digital
Aula 38	Robô Sumô [Estrutura]
Aula 39	Robô Sumô [Programação + Treinamento I]
Aula 40	Robô Sumô [Programação + Treinamento II]
Aula 41	Disputa de Sumôs
Aula 42	Feedbacks + Inventário III

Aula 31
Controle Motor DC

Aula 32
Kit Chassi
2WD Robô

Aula 33
Seguidor de Linha



Sumário

Introdução	2
Objetivos desta Aula	2
Competências Gerais Previstas na BNCC	3
Habilidades do Século XXI a Serem Desenvolvidas	4
Lista de Materiais	4
Roteiro da Aula	4
1. Contextualização	4
2. Montagem e Programação	5
3. Feedback e Finalização	10
Videotutorial	11



Introdução

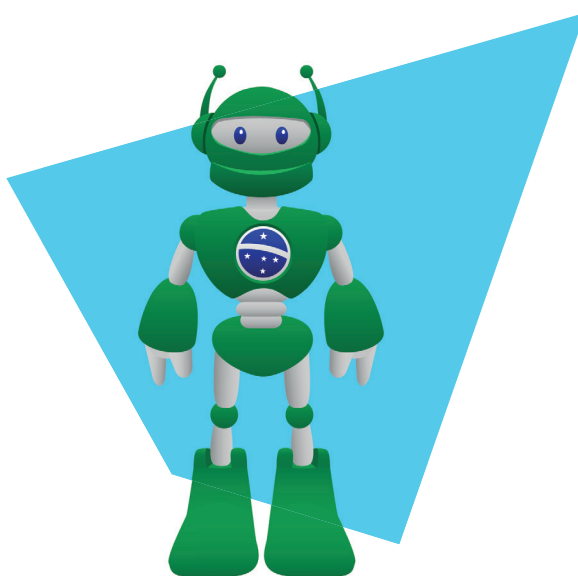
Uma área da robótica que vem apresentando avanços na atualidade é a robótica móvel, um exemplo disso, são os robôs exploradores enviados a outros planetas pela Administração Nacional da Aeronáutica e Espaço (NASA) visando obter informações sobre a superfície e atmosfera desses.

Nesta aula, apresentamos o funcionamento de um robô móvel com trajetória programável, a partir da montagem do Kit Chassi 2WD, presente no kit de robótica.



Objetivos desta Aula

- Entender o funcionamento de um Robô Chassi 2WD (2 rodas);
- Programar o Robô Chassi 2WD para se deslocar em uma trajetória pré-determinada.





Competências Gerais Previstas na BNCC

[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.



Habilidades do Século XXI a Serem Desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de Materiais

- 01 Placa Arduino Uno R3;
- 01 Cabo USB;
- 01 Motor Shield Arduino L293D;
- 01 Kit Chassi 2WD;
- Software Arduino IDE ou mBlock;
- 01 Notebook.



Roteiro da Aula

1. Contextualização (15min):

Quando pensamos em robôs, provavelmente, a primeira imagem que vem a mente é a de robô humanoide, isto é, semelhante à espécie humana, como os apresentados em filmes de ficção científica. Entretanto, há vários formatos de robôs móveis que estão sendo utilizados cada vez mais em nosso cotidiano, principalmente pelo campo industrial e polícia investigativa. Você saberia dizer em que atividades os robôs são utilizados nesses campos?

Os robôs de vários tamanhos e dotados de sensores e atuadores realizam atividades consideradas difíceis ou perigosas para a espécie humana, como, por exemplo, carga e descarga de materiais, desarmamento de bombas e minas terrestres, pintura em recinto fechado, exploração espacial, entre outras funções.

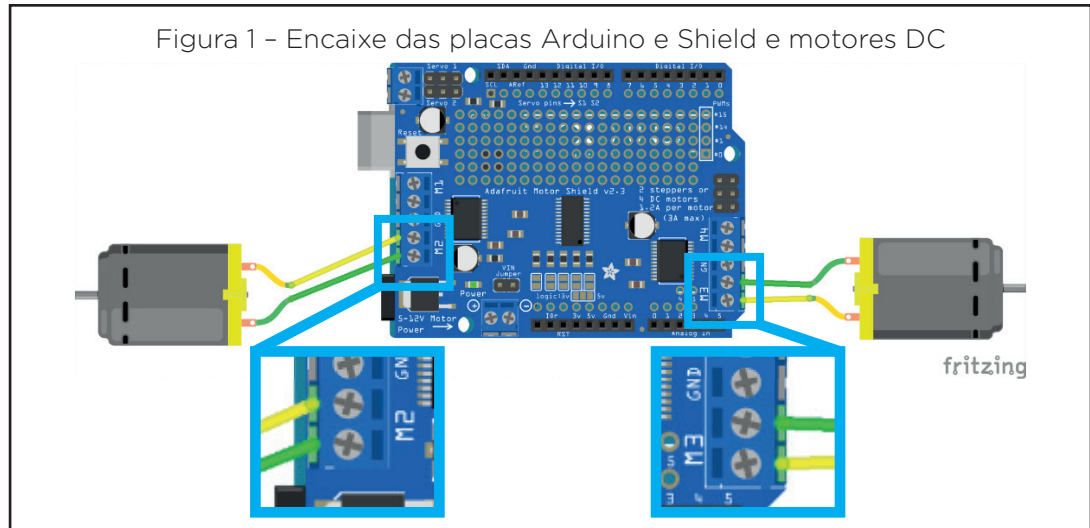
No ensino de robótica, utilizamos os robôs móveis para medir distâncias, testar a velocidade, identificar obstáculos, participar de competições, entre outras atividades. Nesta aula, usaremos o kit Chassi 2WD (2 rodas), presente no kit de robótica, para simular os possíveis movimentos de um robô móvel deste formato.

2. Montagem e Programação (60min):

O kit Chassi 2WD vem desmontado no kit de Robótica. Caso seja o primeiro a utilizar este kit, você, provavelmente, terá que montá-lo. Para tanto, utilize o **Manual de Montagem do kit Chassi 2WD Robô**, disponível no kit de robótica. Mas, atenção! Uma vez montado o kit Chassi e, após sua utilização na aula, ele deverá ser guardado montado, com os demais componentes do kit de Robótica.

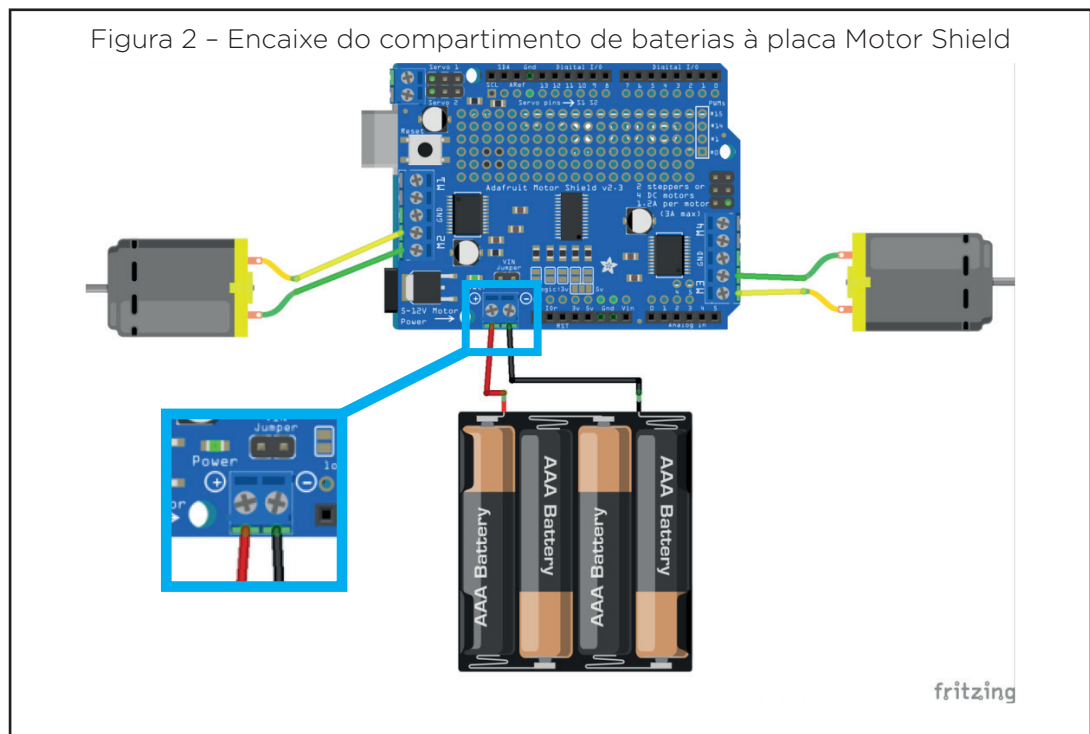
Com o kit Chassi 2WD montado, vamos montar os demais componentes eletrônicos que serão utilizados nesta aula. Encaixe a placa Motor Shield Arduino L293D sobre a placa Arduino, fique atento à posição correta do encaixe e, então, fixe o conjunto, com auxílio de fitas adesivas ou parafusos, sobre o chassi, mantendo os bornes dos motores voltados para as laterais. A seguir, conecte os dois motores DC nas portas M2 e M3 da placa Motor Shield, conforme mostra a figura 1.

Figura 1 - Encaixe das placas Arduino e Shield e motores DC



Insira as baterias de 1,5V no compartimento de baterias e o encaixe à placa Motor Shield, conforme apresentado na figura 2. Caso não possua baterias 1,5V, poderá substituir o conjunto por uma bateria 9V juntamente com o clipe de bateria que acompanha o Kit.

Figura 2 - Encaixe do compartimento de baterias à placa Motor Shield





Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por codificação e por blocos, os movimentos do robô.

i. Linguagem de programação por código

Para essa programação, será necessária a biblioteca **Adafruit Motor Shield Library**, instalada no Software Arduino IDE. Para realizar a instalação, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**.

Inicie a programação conectando a placa Arduino ao computador através de um cabo USB, para que ocorra a comunicação entre a placa microcontroladora e o software Arduino IDE.

No software IDE, escreva ou copie e cole o código-fonte de programação, conforme apresentado no quadro 1.

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/* Programa: Movimentos do chassi 2WD com Arduino e Ponte H
L293D */

#include <AFMotor.h> /* Inclui a Biblioteca AFMotor.h */

AF_DCMotor motor2(2); /* Define motor2 como posição 2 de
controle para motor */
AF_DCMotor motor3(3); /* Define motor3 como posição 3 de
controle para motor */

void setup() {
  motor2.run(RELEASE); /* Inicia com os motores parados */
  motor3.run(RELEASE);
}

void loop() {
  motor2.setSpeed(180); /* Define a potência para os motores
(0-255) */
  motor3.setSpeed(180);

  /* Chama a função frente por 1000 milissegundos */
  frente(1000);
  /* pausa de meio segundo */
  delay(500);
  /* Chama a função ré por 1000 milissegundos */
  re(1000);
  /* pausa de meio segundo */
  delay(500);
}
```

```
/* Chama a função girar em torno do centro no sentido anti-
horário por 1000 milissegundos */
giro_centro_antihorario(1000);
/* pausa de meio segundo */
delay(500);
/* Chama a função girar em torno do centro no sentido horário
por 1000 milissegundos */
giro_centro_horario(1000);
/* pausa de meio segundo */
delay(500);
/* Chama a função girar em torno da roda no sentido anti-
horário por 1000 milissegundos */
giro_roda_antihorario(1000);
/* pausa de meio segundo */
delay(500);
/* Chama a função girar em torno da roda no sentido horário por
1000 milissegundos */
giro_roda_horario(1000);
/* pausa de cinco segundos */
delay(5000);
}

/* Função que move para frente pelo tempo definido */
void frente(int tempo)
{
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  delay(tempo);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
}
/* Função que move para trás pelo tempo definido */
void re(int tempo)
{
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  delay(tempo);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
}
/* Função que gira em torno do centro no sentido anti-horário
pelo tempo definido */
void giro_centro_antihorario(int tempo)
{
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  delay(tempo);
}
```

```

    motor2.run(RELEASE);
    motor3.run(RELEASE);
}
/* Função que gira em torno do centro no sentido horário pelo
tempo definido */
void giro_centro_horario(int tempo)
{
    motor2.run(BACKWARD);
    motor3.run(FORWARD);
    delay(tempo);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
}
/* Função que gira em torno da roda no sentido anti-horário
pelo tempo definido */
void giro_roda_antihorario(int tempo)
{
    motor2.run(FORWARD);
    motor3.run(RELEASE);
    delay(tempo);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
}
/* Função que gira em torno da roda no sentido horário pelo
tempo definido */
void giro_roda_horario(int tempo)
{
    motor2.run(RELEASE);
    motor3.run(FORWARD);
    delay(tempo);
    motor2.run(RELEASE);
    motor3.run(RELEASE);
}

```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verify** (botão com sinal de tique) para verificar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para o Arduino. Para tal, pressione o botão **Upload** (botão com uma seta apontando para a direita).

Após a transferência do programa para o Arduino, o robô executará uma série de movimentos programados, dentre eles: mover para frente, mover para trás, girar nos sentidos horário e anti-horário e parar.

ii. Linguagem de programação por blocos

Outra forma de programar o robô é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Vamos utilizar o software mBlock.

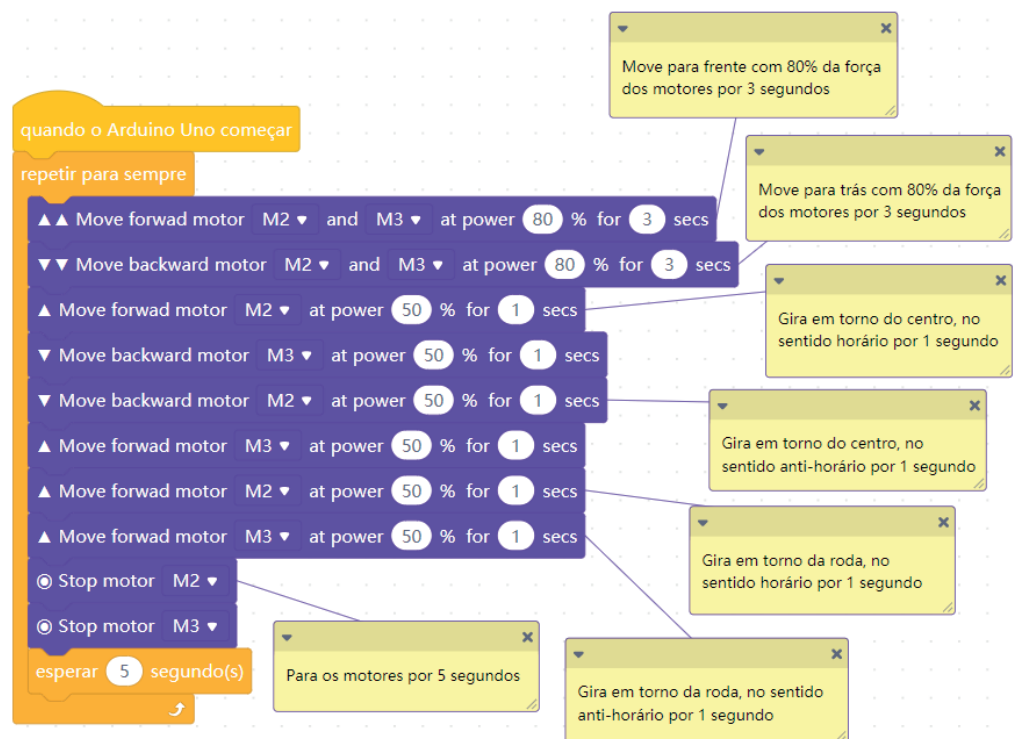
Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

Uma vez selecionado, o Arduino Uno é visualizado no campo **Dispositivos** do mBlock e já é possível iniciar a programação em blocos.

Nesta programação, utilizaremos a extensão **Motor Shield L293D**, que auxiliará na estrutura do nosso programa. Para lembrar como instalar uma extensão no mBlock, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**.

Monte os blocos, arrastando e soltando, de acordo com a programação dos movimentos do robô, como mostra a figura 4.

Figura 4 - Programação em blocos para controle do robô



Assim que os blocos estiverem montados, clique no botão **Conectar** para iniciar a comunicação entre o software mBlock com a placa de Arduino Uno. Ao clicar sobre o botão, aparecerá um *Tooltip* solicitando a confirmação da conexão entre os dois dispositivos.

Uma vez realizada a conexão entre os dispositivos, será ativado, na interface do mBlock, o botão **Upload**, o qual ao ser clicado, o software verificará se não há erros na estrutura do programa e, então, compilará para enviar o programa à placa Arduino.

Com a transferência do código para o dispositivo Arduino Uno, o robô executará movimentos, conforme programação.



Desafios:

I. Faça as alterações necessárias para seu robô se mover numa trajetória quadrada pelo chão.

II. Que tal introduzir um sensor em seu robô para que ele pare de se mover ao se deparar com um obstáculo?



E se... ?

I. O projeto não funcionar, se atente a alguns dos possíveis erros:

1. Confira se a Motor Shield foi conectada corretamente sobre a placa Arduino;

2. Verifique se os motores estão conectados nas mesmas portas da Motor Shield (M2 e M3) que foram utilizadas na programação;

3. Feedback e Finalização (15min):

a. Você e seus colegas conseguiram realizar o desafio? Compartilhem suas impressões e as novas ideias.

b. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.

AULA

32



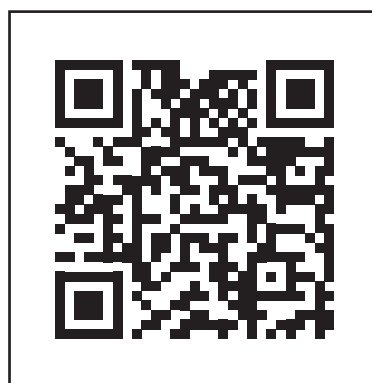
Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/a32robotica>

Acesse, também, pelo QRCode:



12

Robótica

