

ROBÓTICA

Módulo 1



Seguidor de Linha

AULA 33

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

Andrea da Silva Castagini Padilha

Cleiton Rosa

Simone Sinara De Souza

Revisão Textual

Adilson Carlos Batista

Projeto Gráfico e Diagramação

Edna do Rocio Becker

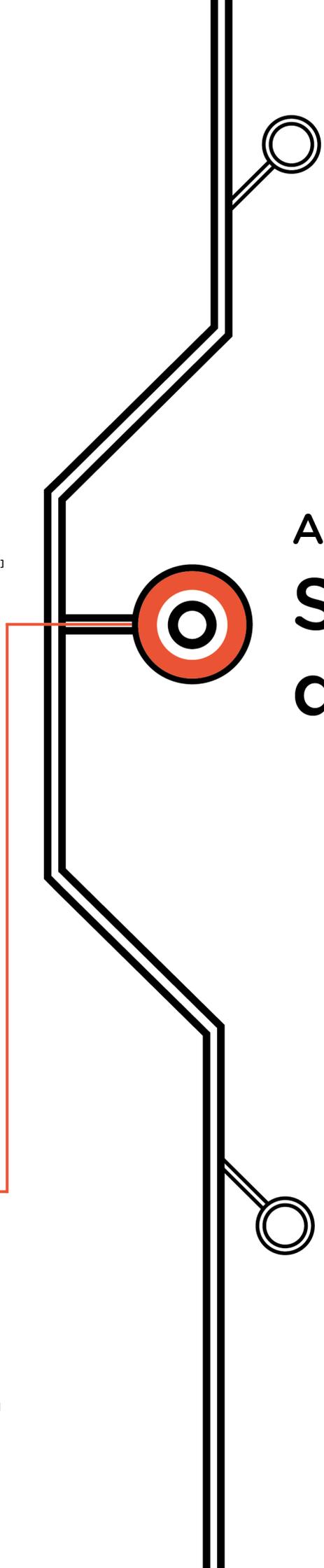
2021

- 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-Iris
- 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 Treina 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 32
**Kit Chassi
 2WD Robô**

Aula 33
**Seguidor
 de Linha**

Aula 34
**Sensor
 de Distância**



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	5
1. Contextualização	5
2. Montagem e Programação	6
3. Feedback e Finalização	15
Videotutorial	16



Introdução

A presença, atuação e envolvimento de robôs já estão sendo percebidos no cotidiano das pessoas, seja para realizar pequenas tarefas domésticas ou em tarefas industriais, nas quais há limitações dos seres humanos.

Nesta aula, você conhecerá o funcionamento de um robô seguidor de linha, a partir da montagem e programação de um protótipo, utilizando componentes presentes no kit de robótica.



Objetivos desta Aula

- Entender o que é um seguidor de linha;
- Identificar os componentes necessários para o funcionamento do seguidor de linha;
- Programar o funcionamento de um seguidor de linha.





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.



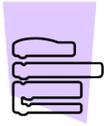
Habilidades do Século XXI a Serem Desenvolvidas

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



Lista de Materiais

- 1 placa Arduino UNO R3;
- 1 cabo USB;
- 1 motor shield L293D driver;
- 1 kit chassi 2WD;
- 2 módulos sensor de obstáculo infravermelho IR;
- 6 jumpers fêmea-fêmea;
- 1 Notebook;
- Software mBlock ou Arduino IDE.



Roteiro da Aula

1. Contextualização (15min):

Você já teve a oportunidade de assistir ou participar de uma competição de robôs? Aquelas, na qual os participantes colocam sobre uma arena, robôs no formato de carrinhos que, com uma programação, movimentam-se em trajetória determinada e delimitada? Você saberia dizer como são chamados esses robôs?

O termo **seguidor de linha** refere-se a um robô que tem a função de detectar, com o uso de sensores de obstáculos infravermelhos, caminhos a percorrer, respeitando uma trajetória predeterminada, via programação.

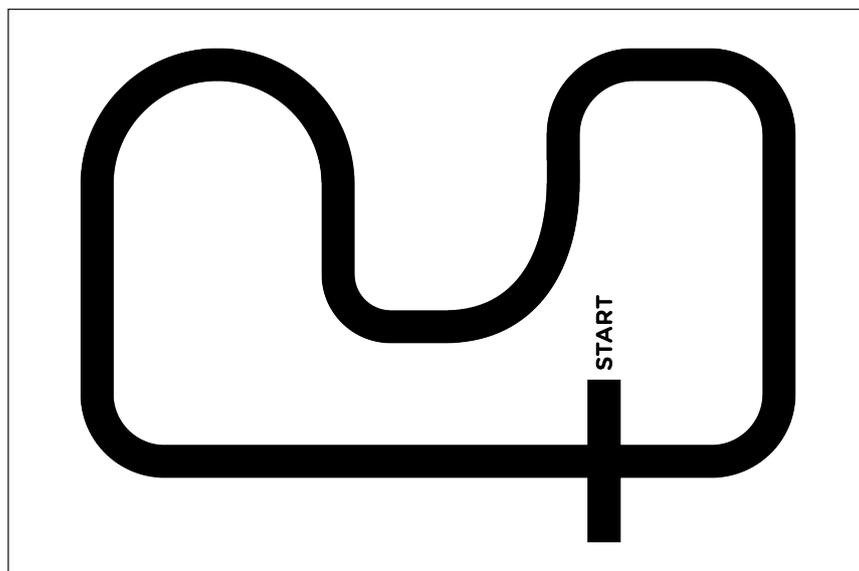
Este modelo de robô é bastante utilizado em competições de robótica, nas quais, vence o robô que finalizar o trajeto em menor tempo. Na indústria, são utilizados para gerenciar estoques, manejar prateleiras de produtos, levando caixas de uma área para outra, por orientação de faixas no chão.

O funcionamento do robô seguidor de linha consiste na presença de sensores de obstáculos infravermelhos IR, dispostos nas laterais dianteiras do robô, os quais têm a função de detectar a presença de faixas limites presentes na superfície, impedindo assim, a saída do robô do trajeto demarcado.

Na **Aula 30 - Sensor de Obstáculo IR**, você aprendeu que este sensor é composto por um emissor, um receptor de radiação infravermelho e um circuito comparador que facilita a conexão com o Arduino Uno. O receptor detecta a radiação infravermelha refletida na superfície e identifica variações de intensidade, portanto, é recomendado que as cores da superfície e das faixas limitantes, sejam pretas e brancas, permitindo assim, o contraste e, conseqüentemente, maior variação de intensidade.

Você pode montar um circuito com o auxílio de fitas adesivas, atentando para que a fita faça contraste à superfície escolhida para melhor funcionamento dos sensores (sugestão - figura 1).

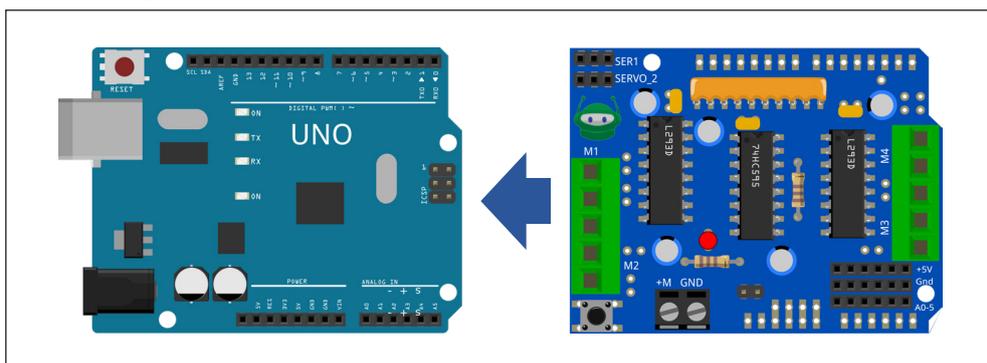
Figura 1 - Sugestão de circuito para o robô seguidor de linha



2. Montagem e Programação (60min):

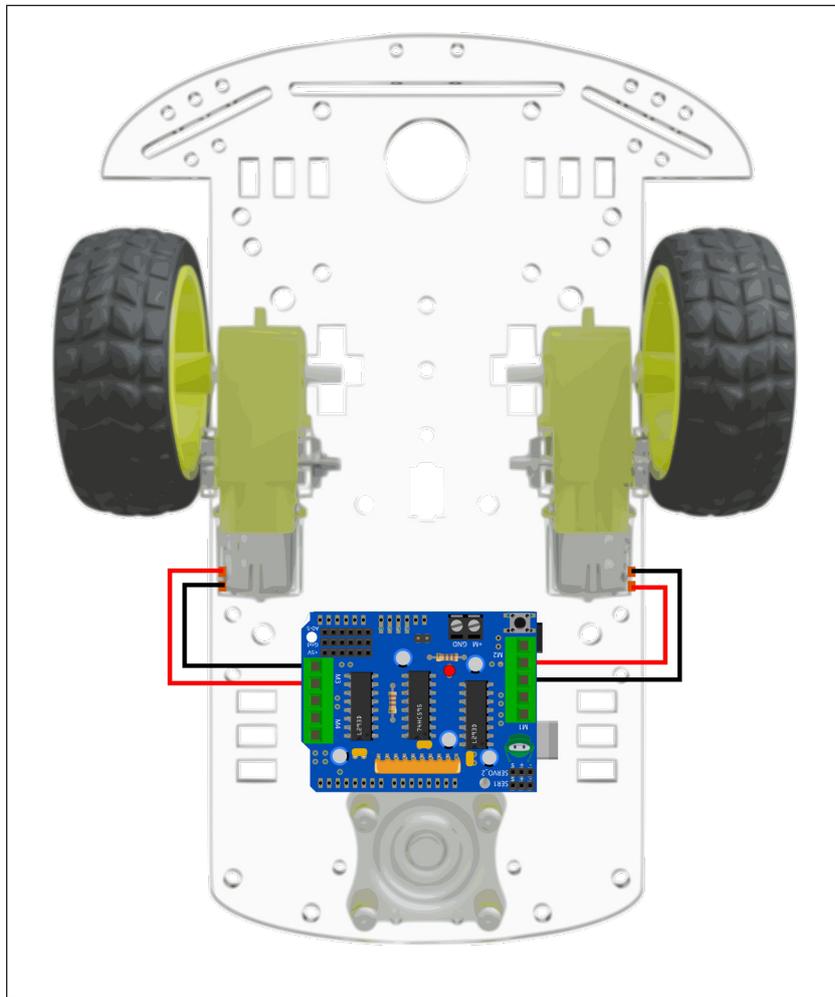
Iniciaremos a montagem dos componentes eletrônicos encaixando a placa motor shield L293D driver sobre o Arduino Uno (figura 2).

Figura 2 - Encaixando a motor shield sobre o Arduino Uno



Em seguida, coloque o conjunto (Arduino + motor shield) sobre o chassi 2WD e conecte os terminais dos motores aos bornes M2 (motor da direita) e M3 (motor da esquerda) (figura 3). Não há necessidade de se preocupar com a ordem dos fios neste momento, pois após a programação, poderemos ajustar a direção de rotação dos motores conforme necessário.

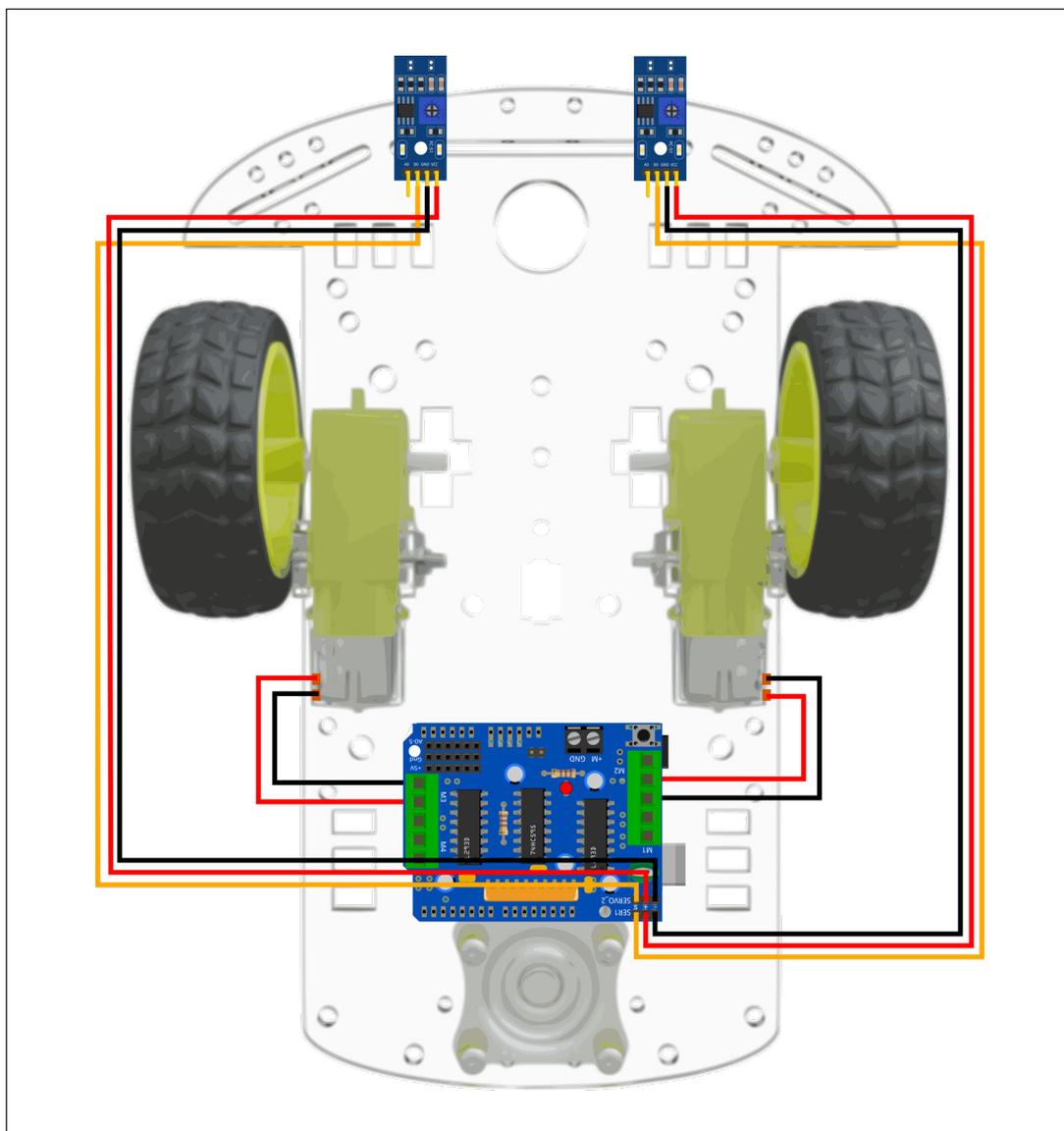
Figura 3 - Conectando os motores na motor shield



Utilizando seis jumpers fêmea-fêmea, conecte:

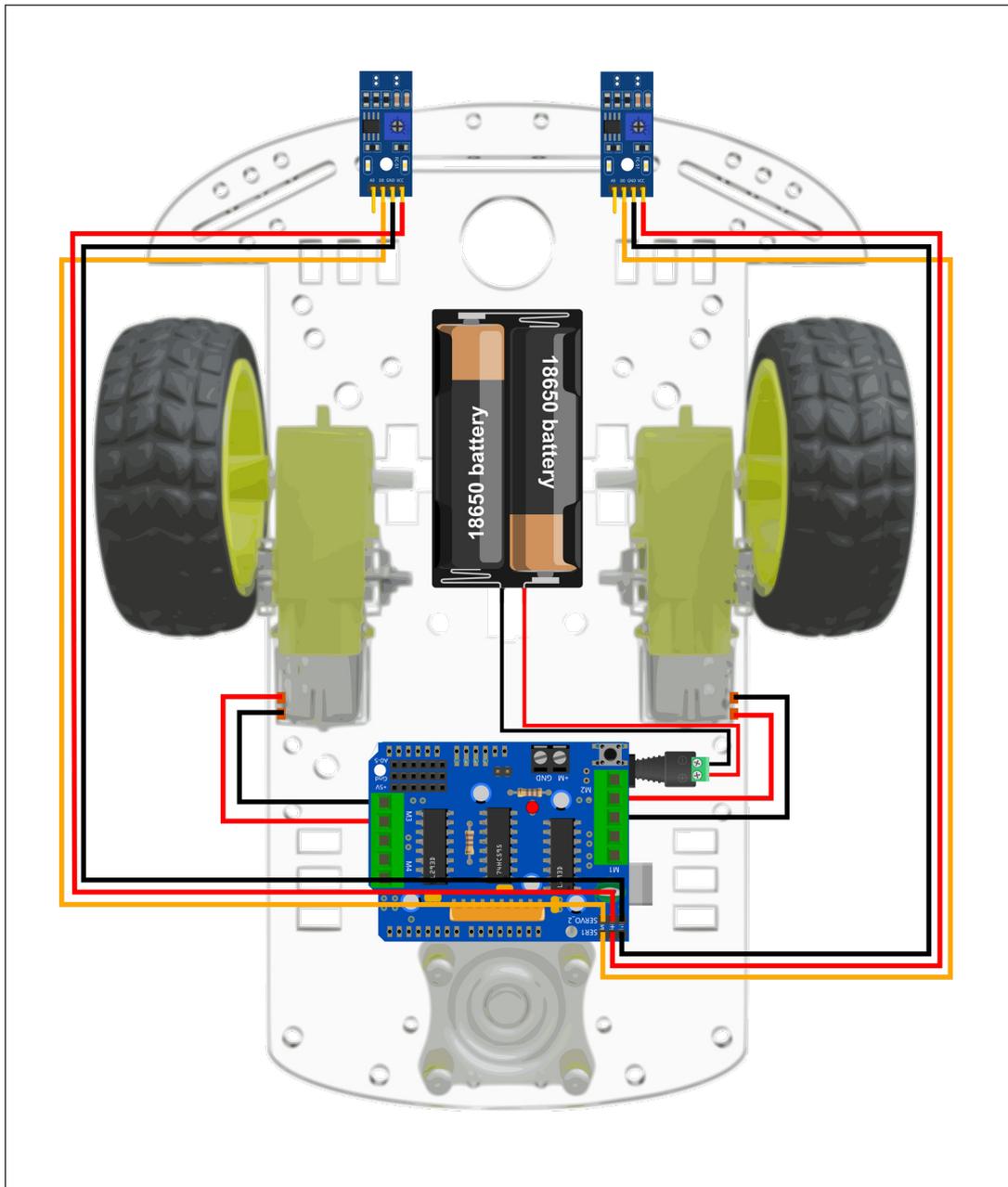
- Os pinos **VCC** dos sensores de obstáculos (pinos da direita) com os pinos positivos (+) dos conectores dos servos na motor shield (pinos centrais);
- Conecte os dois pinos **GND** dos sensores de obstáculos (segundo pino da direita para a esquerda) com os pinos negativos (-) dos conectores dos servos na motor shield (pinos da direita);
- E por fim conecte os dois pinos **DO** dos sensores de obstáculos (terceiro pino da direita para a esquerda) com os pinos **S** (pinos da esquerda) dos conectores dos servos na motor shield, seguindo a ordem: sensor da esquerda, conectado o **S** do **Servo 2** e sensor da direita, conectado com o **S** do **Servo 1** (figura 4).

Figura 4 - Conectando os sensores aos conectores SERVOs da motor shield



Por fim, procederemos à conexão da fonte de alimentação ao motor shield. É crucial estar atento à fonte de energia utilizada, uma vez que os motores demandam um consumo significativo durante o funcionamento. Portanto, é essencial empregar uma fonte capaz de fornecer uma tensão de 6 a 12 Volts. Na Figura 5, demonstramos o uso de um estojo com duas pilhas 18650, no entanto, você tem a liberdade de escolher qualquer outra fonte que satisfaça os requisitos mínimos de operação do seu projeto.

Figura 5 - Conectando a fonte de alimentação do robô à motor shield





Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por codificação e por blocos, o robô seguidor de linha.

i. Linguagem de programação por código

Para esta programação utilizaremos a biblioteca **Adafruit Motor Shield Library V1** desenvolvida por Adafruit, instalada no software Arduino IDE. Na **Aula 05 - Softwares Arduino IDE e mBlock** você encontrará orientações sobre como realizar a instalação desta biblioteca, mas se você estiver utilizando a versão online do Arduino IDE, basta informar no sketch o uso da biblioteca com o comando `#include <AFMotor.h>`.

Inicie a programação através de um cabo USB, conectando a placa Arduino Uno ao computador, 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.

Atenção: Nas linhas 25 e 26, você deverá escolher a cor da linha que seu robô deverá seguir (preta ou branca), para isto, você apenas precisa descomentar uma das linhas correspondente a sua pista (removendo o `//` do início da linha).

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

```
//*****  
/* Aula 33 - Robô Seguidor de Linha */  
/* Programação: Nesta programação é realizado o */  
/* controle dos motores a partir das informações obtidas dos */  
/* dois sensores de obstáculos IR. */  
/* Quando os dois sensores não detectarem a linha, enviarão */  
/* o nível lógico 0 para o Arduino, o qual fará os dois */  
/* motores seguirem em frente. Quando os dois sensores */  
/* detectarem a linha, enviarão o nível lógico 1 para o */  
/* Arduino, o qual fará os motores pararem. Quando apenas um */  
/* sensor (direita ou esquerda) detectar a linha, enviará */  
/* níveis lógicos alternados 0 e 1 ou 1 e 0 ao Arduino, o */  
/* qual fará o controle da direção que o robô deverá tomar */
```

```

/* para centralizar o chassi à linha. */
/* Links para obtenção da biblioteca by Adafruit */
/* http://librarymanager/All#Adafruit#Motor#V1 */
/* https://github.com/adafruit/Adafruit-Motor-Shield-library */
/*****

/*=====ATENÇÃO!===== ATENÇÃO! =====ATENÇÃO!=====*/
/* Antes de compilar/carregar o código, você deverá */
/* selecionar qual a cor da linha que o robô deverá */
/* seguir, para isto, descomente (removendo o //) de */
/* uma das duas linhas abaixo. */
//#define CorLinha 1 /* Linha PRETA */
//#define CorLinha 0 /* Linha BRANCA */
/*=====ATENÇÃO!===== ATENÇÃO! =====ATENÇÃO!=====*/

/* Configura o modo da detecção da linha de acordo com a cor */
/* escolhida acima. */
#if CorLinha == 1
boolean detectou = 1;
boolean naodetectou = 0;
#else
boolean detectou = 0;
boolean naodetectou = 1;
#endif

/* Inclui a Biblioteca AFMotor para controle do motor. */
#include <AFMotor.h>
/* Variáveis que armazenam a leitura dos sensores de */
/* obstáculo IR. */
int Sensor_Direita;
int Sensor_Esquerda;
int velocidade = 150;
/* Define os pinos 9 e 10 para os sensores IR. */
#define Pin_Sensor_Esquerda 9
#define Pin_Sensor_Direita 10
/* Define a saída M2 da motor shield para o Motor_Direita. */
AF_DCMotor Motor_Direita(2);
/* Define a saída M3 da motor shield para o Motor_Esquerda. */
AF_DCMotor Motor_Esquerda(3);
void setup() {
    Serial.begin(9600);
    pinMode(Pin_Sensor_Direita, INPUT);
    pinMode(Pin_Sensor_Esquerda, INPUT);
    /* Inicia com os motores parados. */
    Motor_Esquerda.run(RELEASE);
    Motor_Direita.run(RELEASE);
}

```

```
void loop() {

    /* Sensores realizam a leitura. */
    Sensor_Direita = digitalRead(Pin_Sensor_Direita);
    Sensor_Esquerda = digitalRead(Pin_Sensor_Esquerda);
    /* Nenhum sensor detectou a linha. */
    if (Sensor_Esquerda == naodetectou && Sensor_Direita ==
naodetectou) {
        /* Define as velocidades dos motores. */
        Motor_Esquerda.setSpeed(velocidade);
        Motor_Direita.setSpeed(velocidade);
        /* Robô move para frente. */
        Motor_Esquerda.run(FORWARD);
        Motor_Direita.run(FORWARD);
        Serial.println("Nenhum sensor detectou a linha.");
    }
    /* Sensor da direita detectou a linha. */
    if ((Sensor_Esquerda == detectou) && (Sensor_Direita ==
naodetectou)) {
        /* Define as velocidades dos motores. */
        Motor_Esquerda.setSpeed(velocidade);
        Motor_Direita.setSpeed(velocidade);
        /* Robô vira à direita. */
        Motor_Esquerda.run(BACKWARD);
        Motor_Direita.run(FORWARD);
        Serial.println("Somente o sensor da Esquerda detectou a
linha.");
    }
    /* Sensor da esquerda detectou a linha. */
    if ((Sensor_Esquerda == naodetectou) && (Sensor_Direita ==
detectou)) {
        /* Define as velocidades dos motores. */
        Motor_Esquerda.setSpeed(velocidade);
        Motor_Direita.setSpeed(velocidade);
        /* Robô vira à esquerda. */
        Motor_Esquerda.run(FORWARD);
        Motor_Direita.run(BACKWARD);
        Serial.println("Somente o sensor da Direita detectou a
linha.");
    }
    /* Os dois sensores detectaram a linha. */
    if (Sensor_Esquerda == detectou && Sensor_Direita == detectou)
    {
        /* Robô permanece parado. */
        Motor_Esquerda.run(RELEASE);
        Motor_Direita.run(RELEASE);
        Serial.println("Ambos sensores detectaram a linha.");
    }
}
```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão Verificar (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 a placa de Arduino. Para tal, pressione o botão **Carregar** (botão com uma seta apontando para a direita).

Após a transferência do programa para o Arduino, o robô executará seu movimento sobre a linha, mantendo-se na trajetória (figura 1) através dos dados emitidos pelos dois sensores de obstáculos.

ii. Linguagem de programação por blocos

Outra forma de programar o robô seguidor de linha 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 variáveis que auxiliarão na estrutura do nosso programa e a extensão **Motor Shield L293D** (para recordar como criar variáveis e instalar uma extensão no mBlock, consulte a **Aula 05 - Softwares Arduino IDE e mBlock**).

Uma vez realizada a instalação da extensão necessária, monte os blocos, arrastando e soltando, de acordo com a programação dos movimentos do robô seguidor de linha (figura 8).

Observe que o parâmetro informado no segundo bloco, será necessário informar a cor da linha que o robô deverá seguir: **0** para seguir uma linha **BRANCA** ou **1** para seguir uma linha **PRETA**.

Figura 8 - Programação em blocos para controle do robô seguidor de linha

```
quando o Arduino Uno começar
definir Cor da Linha para 1
repetir para sempre
  se Cor da Linha = 1 então
    definir detectado para 1
    definir não detectado para 0
  senão
    definir detectado para 0
    definir não detectado para 1
  definir Sensor_Esquerda para ler pino digital 9
  definir Sensor_Direita para ler pino digital 10
  se Sensor_Direita = não detectado e Sensor_Esquerda = não detectado então
    Move forward motor M2 at power 80 %
    Move forward motor M3 at power 80 %
    escrever Nenhum sensor detectou a linha. na porta serial
  se Sensor_Direita = detectado e Sensor_Esquerda = não detectado então
    Move backward motor M2 at power 80 %
    Move forward motor M3 at power 80 %
    escrever Somente o sensor da DIREITA detectou a linha. na porta serial
  se Sensor_Direita = não detectado e Sensor_Esquerda = detectado então
    Move forward motor M2 at power 80 %
    Move backward motor M3 at power 80 %
    escrever Somente o sensor da ESQUERDA detectou a linha. na porta serial
  se Sensor_Direita = detectado e Sensor_Esquerda = detectado então
    Stop motor M2
    Stop motor M3
    escrever AMBOS sensores detectaram a linha. na porta serial
```

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 **Conectar**, 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 seguidor de linha estará pronto para percorrer a trajetória delimitada no circuito (figura 1).



Desafios:

- i. Faça as alterações necessárias para seu robô se mover para trás, mantendo sobre a linha.
- ii. Que tal introduzir um sensor em seu robô para que ele pare de se mover ao se deparar com um obstáculo, faça o desvio e retorne à sua trajetória sobre a linha?



E se...

- i. Caso o projeto não funcione, 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. Caso o(s) motor(es) esteja(m) girando no sentido contrário, inverta o(s) cabo(s) no(s) borne(s) da motor shield.

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.



Videotutorial

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



<https://rebrand.ly/a33robotica>

Acesse, também, pelo QRCode:



DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edgar Cavalli Junior
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Orlando de Macedo Junior
Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

