Oficina de introdução à robótica

Quarta Aula

Instrutores: Pedro Rocha e João Pedro Arruda
Organização: Profa. Regiane Kawasaki e Prof. Victor Santiago

> Revisão Aula Passada

> Sensor de Linha

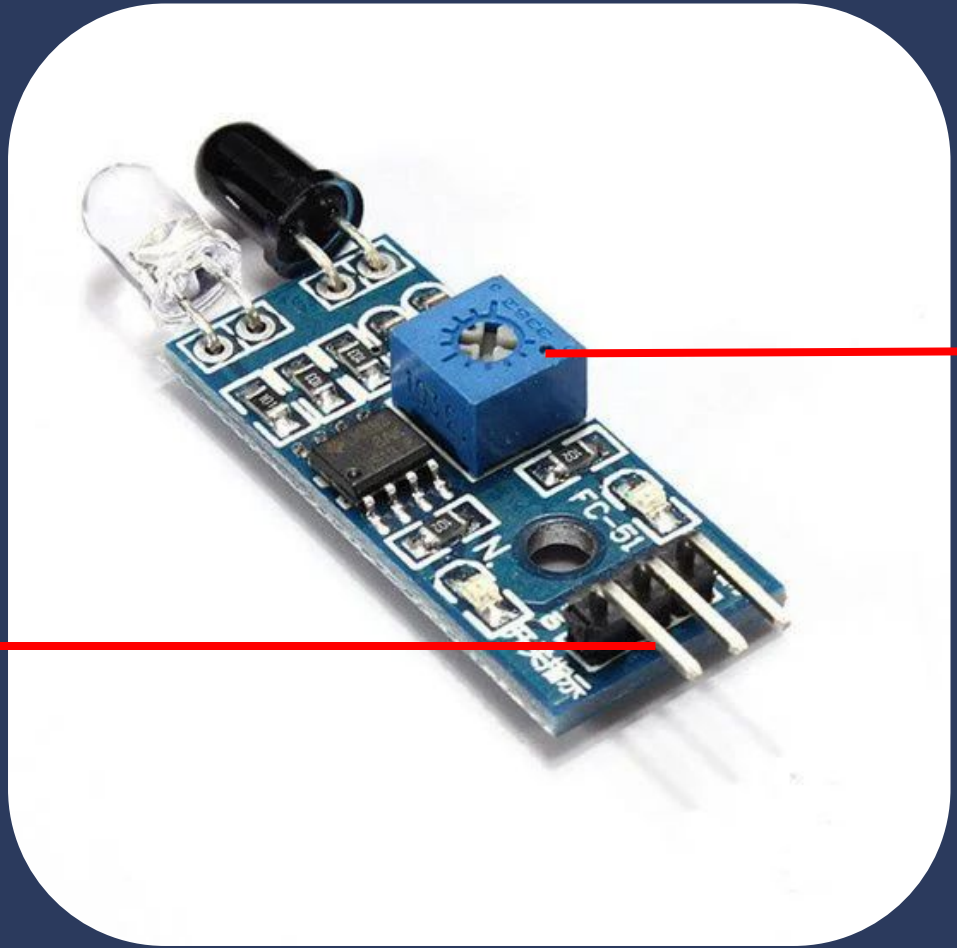
/Sensor de Linha

Funcionamento:
Os sensores de linha são baseados na emissão de luz infravermelha e um fototransistor ou fotodiodo que detecta a luz refletida. Quando o sensor está sobre uma superfície, a luz infravermelha é refletida de volta para o fototransistor ou fotodiodo. O valor da reflexão **depende do contraste entre a linha e a superfície** circundante.

VCC (Voltage Common Collector):
O pino VCC é onde você fornece a tensão de alimentação ao sensor. Este pino deve ser conectado à tensão de alimentação necessária para o funcionamento do sensor. Para a maioria dos sensores, como o sensor de linha QRE1113, é comum usar uma tensão de 5V. Certifique-se de que a tensão aplicada esteja de acordo com as especificações do sensor.

/Sensor de Linha

OUT (Output):
O pino OUT é onde o sensor fornece a saída do sinal. A natureza dessa saída pode variar entre sensores. Pode ser uma saída analógica que fornece uma tensão proporcional à intensidade do reflexo de luz, ou uma saída digital que indica a detecção ou não detecção da linha. A leitura deste pino é a informação que você usará para determinar o status do sensor.


Calibrador
Entradas
FC-51

> Servo Motor

/Servo Motor

Funcionamento:
A maioria dos servo motores tem uma faixa limitada de movimento, geralmente de 0 a 180 graus. Isso significa que o eixo do servo pode ser movido apenas dentro desse intervalo. Para conectar um servo motor, é necessário usar uma porta PWM.

Conexões:
Servo motores geralmente funcionam com uma tensão de 4,8 a 6 volts. Eles têm três fios: um para alimentação e fornecimento de energia (VCC), um para aterramento (GND) e um para o sinal de controle que é conectado com um pino PWM.

/Servo Motor

Programação:
Para programar um servo motor, é necessário importar a biblioteca Servo.h, usar myservo.attach para definir o pino usado no setup e usar myservo.write para definir para qual posição o servo irá girar

```
1  void setup() {
2    myservo.attach(9);
3  }
```

```
1  myservo.write(180);
```

/Quarta Aula

/Quarta Aula

Tópicos Principais:

1. Ponte H
2. Motores
3. Random

/Ponte H

/Ponte H

O que é:
Uma ponte H é um circuito eletrônico usado para controlar a direção e a velocidade de motores elétricos, como motores de corrente contínua (DC) e motores de passo. Ela permite que os motores girem em ambas as direções e regula a velocidade por meio de pulsos PWM.

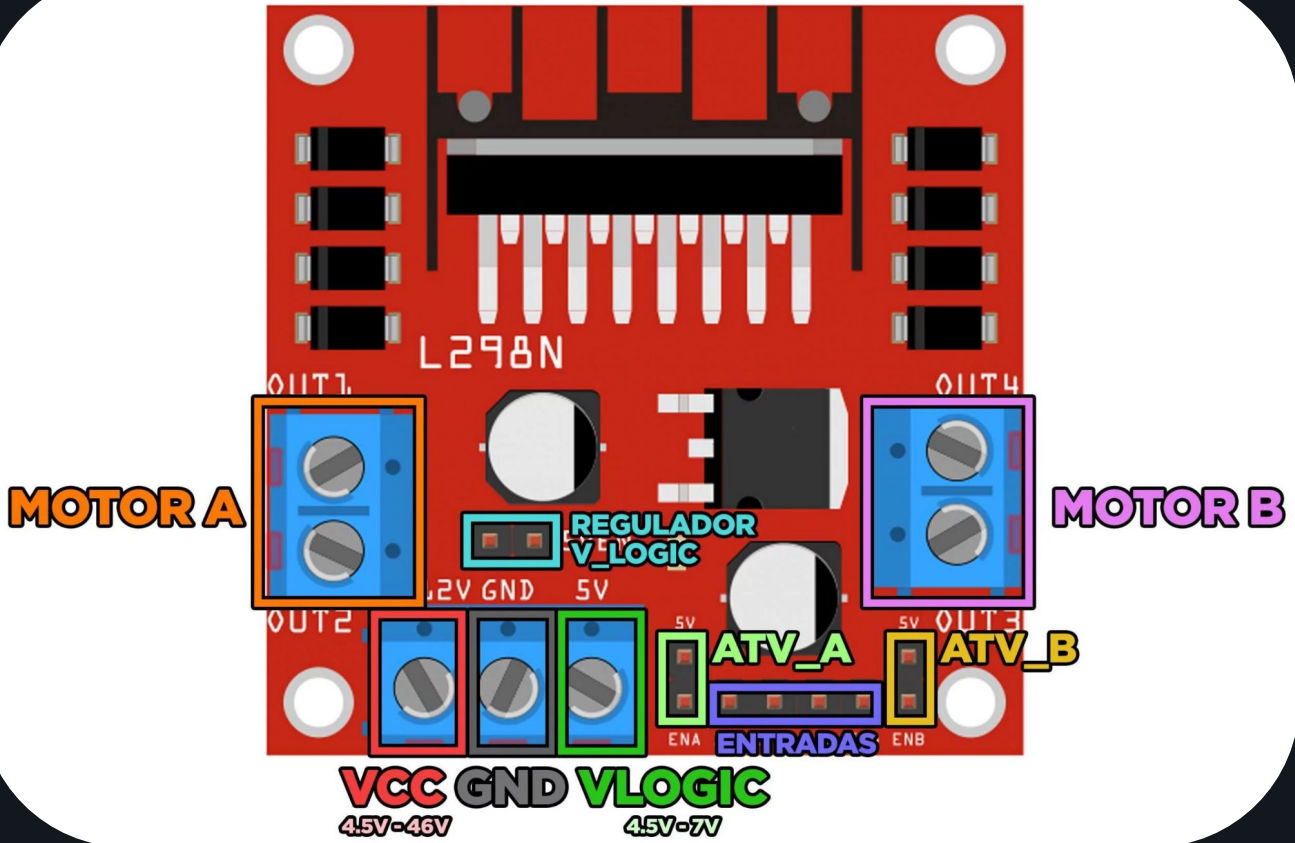
Proteção Elétrica:
A ponte H desempenha um papel crucial na proteção elétrica do Arduino. Ela isola o microcontrolador de picos de tensão e corrente gerados por motores, evitando danos. Essa proteção permite o controle seguro de motores de alta potência em projetos eletrônicos, tornando a ponte H um componente indispensável para garantir a integridade e a longevidade do Arduino.

/Ponte H

Alimentação:
A alimentação externa de uma ponte H é a fonte de energia que fornece a corrente necessária para os motores. Ela é separada da alimentação do circuito de controle (geralmente alimentado pelo Arduino) para evitar picos de corrente que possam danificar o controle, garantindo que os motores funcionem eficazmente e com segurança. Isso permite o uso de motores com maior potência do que o Arduino pode fornecer sozinho, tornando a alimentação externa essencial em muitos projetos.

/Ponte H

Conectores:
Uma ponte H pode possuir uma porção demasiada de conectores, entretanto, existem três que são os principais, que são: **Conectores para motores**, que permitem a alimentação necessária para o funcionamento dos motores, **Conector de Alimentação Externa**, que são os conectores que permitem a alimentação externa da própria ponte H, **Conectores de Controle**, que permitem a comunicação do arduino com os motores que irão ser conectados com a ponte H.


L298N
OUT1
OUT4
OUT2
OUT3
MOTOR A
MOTOR B
REGULADOR
V_LOGIC
12V GND 5V
5V
5V
ATV_A
ATV_B
ENA
ENTRADAS
ENB
VCC GND VLOGIC
4.5V - 46V
4.5V - 7V

/Motores

/Motores

O que são:
Motores DC (Direct Current) são dispositivos que convertem energia elétrica fornecida em rotação mecânica. Eles são comuns em projetos de robótica e automação.

Motores e Ponte H:
Para controlar motores DC com segurança, é importante usar um driver de motor ou uma ponte H, pois esses circuitos protegem o Arduino de picos de corrente gerados pelos motores, o que poderia danificar a placa. Além disso, você deve alimentar os motores DC com uma fonte de energia separada, não diretamente do Arduino, para evitar sobrecarregar a placa.

/Motores

Controles de direção:
Além do controle de velocidade, você também pode controlar a direção dos motores DC usando o Arduino Uno. Isso geralmente é feito usando duas saídas digitais do Arduino para acionar o driver de motor ou ponte H. Invertendo o estado das saídas digitais, você pode fazer o motor girar para frente ou para trás.

Controle de velocidade:
Com as entradas PWM do Arduino Uno, é possível controlar a velocidade dos motores DC. Para fazer isso, você pode conectar os fios do motor a um driver de motor ou ponte H, que é um circuito que permite controlar a direção e a velocidade do motor.

/Motores

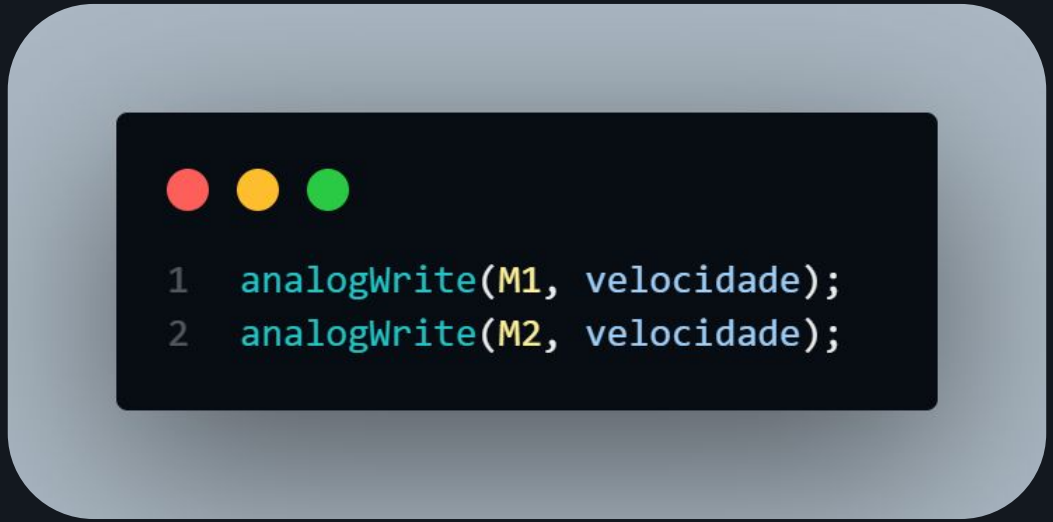
Programação:
Conecte os pinos de controle da ponte H (geralmente IN1, IN2, IN3, IN4) aos pinos digitais do Arduino (por exemplo, IN1 a IN4 para pinos 2, 3, 4, 5 no Arduino Uno)

```
#define M1 9
#define M2 11
#define dir1 8
#define dir2 10
#define pin_S2 6
```

```
pinMode(M1, OUTPUT);
pinMode(M2, OUTPUT);
pinMode(dir1, OUTPUT);
pinMode(dir2, OUTPUT);
```

/Motores

Programação:
Ajuste a velocidade do motor com a sua preferência, usando `analogWrite()`, por se tratar de uma entrada PWM.

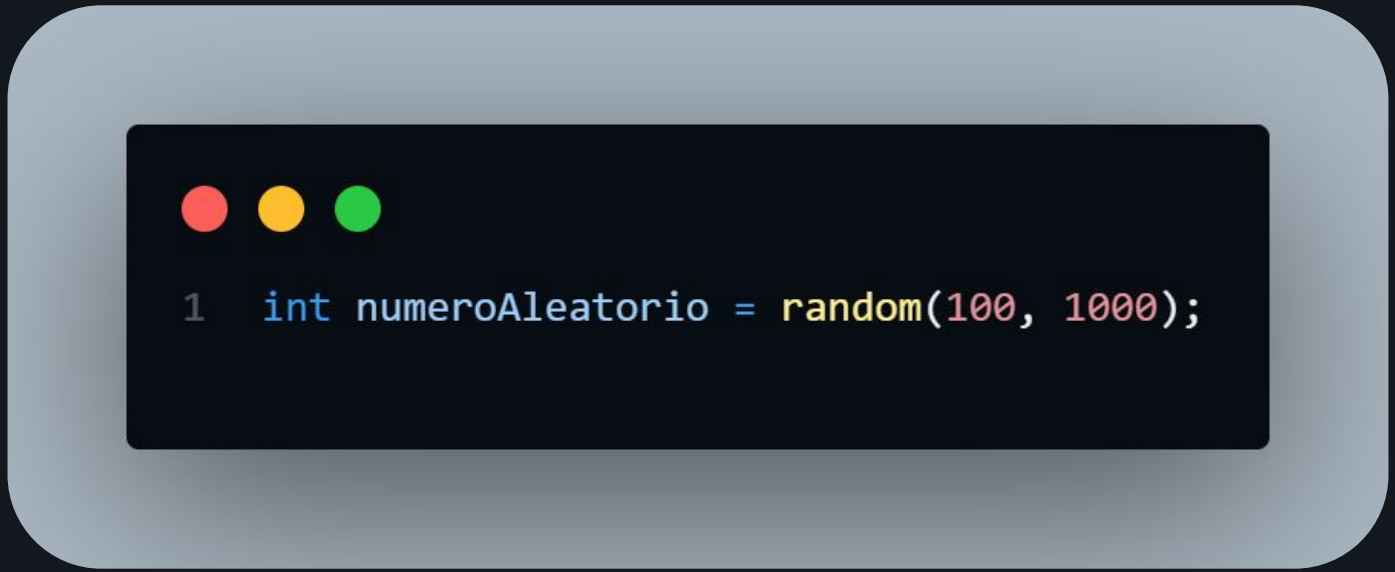


/Função Random

/Função Random

O que é:
A função random() no Arduino é uma função que gera números pseudo-aleatórios. Ela é frequentemente usada para criar variações aleatórias em programas e projetos Arduino. Aqui estão alguns pontos importantes sobre a função random():

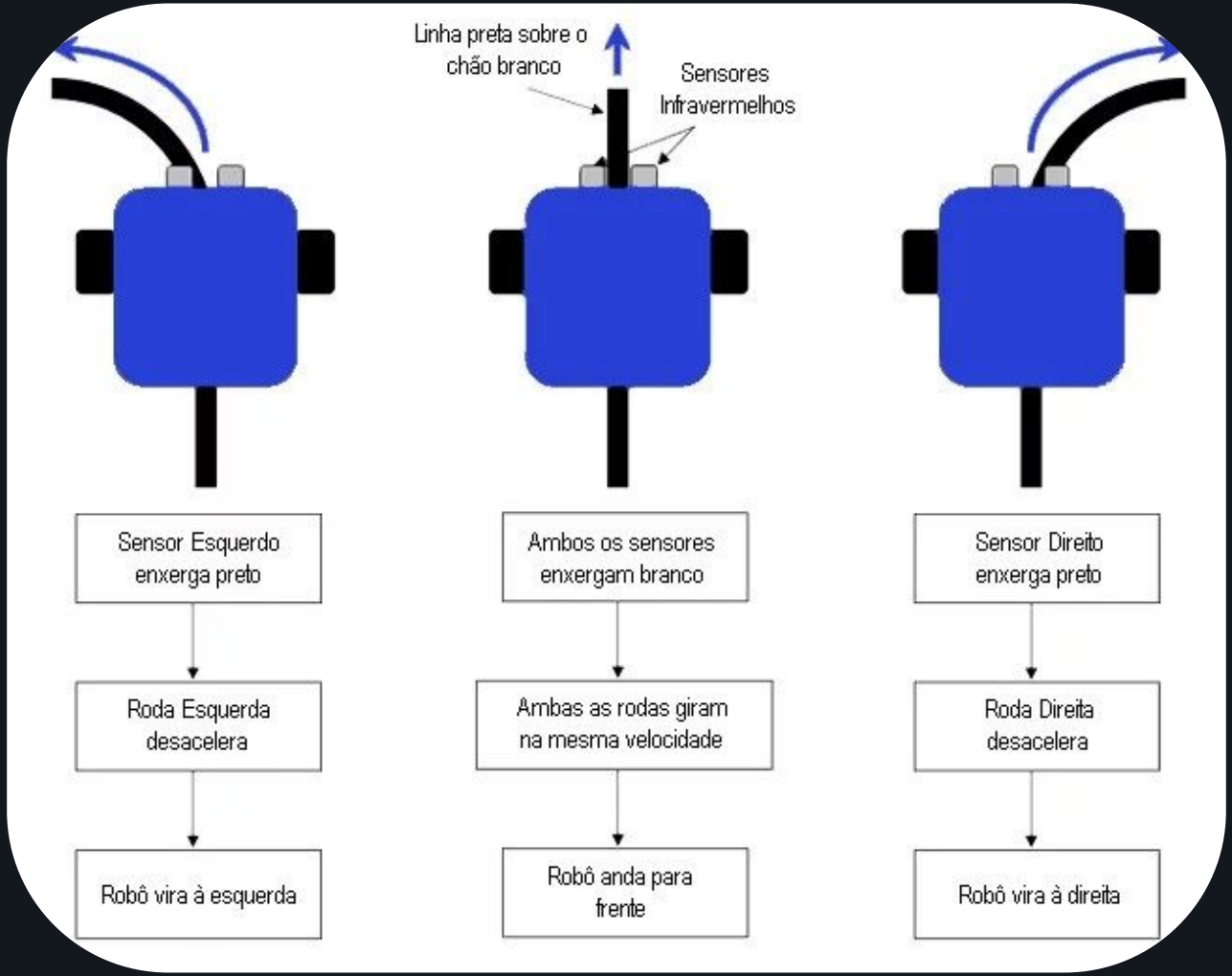
Programação:
Para usar essa função de forma simples, basta chamar a função com a palavra reservada random e colocar como argumento o intervalo desejado

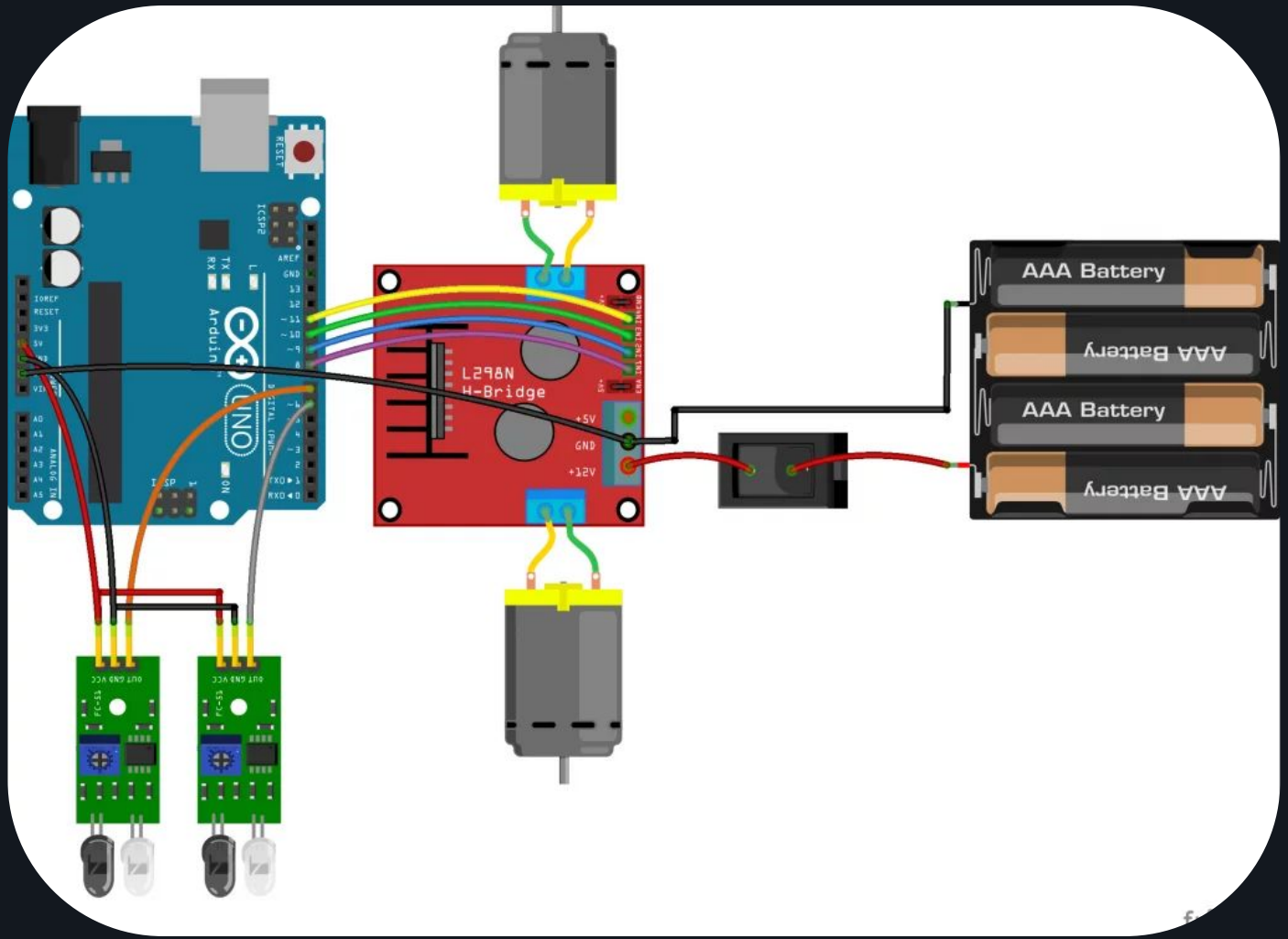

1 int numeroAleatorio = random(100, 1000);

/Hora do Desafio Final

/Desafio Final

Boa Sorte


Linha preta sobre o chão branco
Sensores Infravermelhos
Sensor Esquerdo enxerga preto
Roda Esquerda desacelera
Robô vira à esquerda
Ambos os sensores enxergam branco
Ambas as rodas giram na mesma velocidade
Robô anda para frente
Sensor Direito enxerga preto
Roda Direita desacelera
Robô vira à direita


L298N
H-Bridge
+5V
GND
+12V
AAA Battery
AAA Battery
AAA Battery
AAA Battery