

# Robótica III

## Professores

**Antonio Fernando Traina** – Professor da FATEC – Franca  
Doutor em Física Aplicada Computacional - IFSC-USP,  
[aftraina@gmail.com](mailto:aftraina@gmail.com)



**Roseli Aparecida Romero** – Coordenadora do Curso  
Professora ICMC-USP,  
[rafrance@icmc.usp.br](mailto:rafrance@icmc.usp.br)



**Março -2019**

## Horario da aula:

14:00 – 15:30 – Primeira aula

15:30 – 15:45 – Intervalo

15:45 – 17:00 – Segunda aula

# Agenda do curso

## Introdução

- Conceitos Iniciais
- A Olimpíada Brasileira de Robotica - OBR
- Conceitos de Arduino
- Conceitos de Sensores e atuadores
- Apresentação do Kit básico
- Plataforma e simulação

## Programação para arduino

- Apresentação das Regras da Olimpíada Brasileira de Robotica
- Estrutura da Linguagem do Arduino - linguagem C
- As portas de E/S do Arduino e suas funções em C
- Programando os sensores e atuadores



**Aula de Hoje**

## Desenvolvendo Programação robô seguidor de linha

- Motores
- Sensores claro/escuro
- Sensores de cor
- Sensores de distância
- Aprimoramentos e melhorias

## Competição entre equipes – Organizar.

# Material para a aula de hoje

Item	Descrição	Preço	Quantidade
2	Placa de desenvolvimento Arduino R3	26.00	1
3	Cabo USB A-B	9.90	1
4	jumper macho-femea (10 pares)	15.00	1
5	Bateria 9 volts + cabo	25.25	1
6	Motor Shield Driver Ponte H L298N	23.00	1
7	Mini Protoboard 170 Pontos	9.50	1
	<b>TOTAL</b>	<b>162.10</b>	<b>9</b>

# Robótica III

## Aula II

**Março -23/2019**

# Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha

Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

## Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C) ←

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha

Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

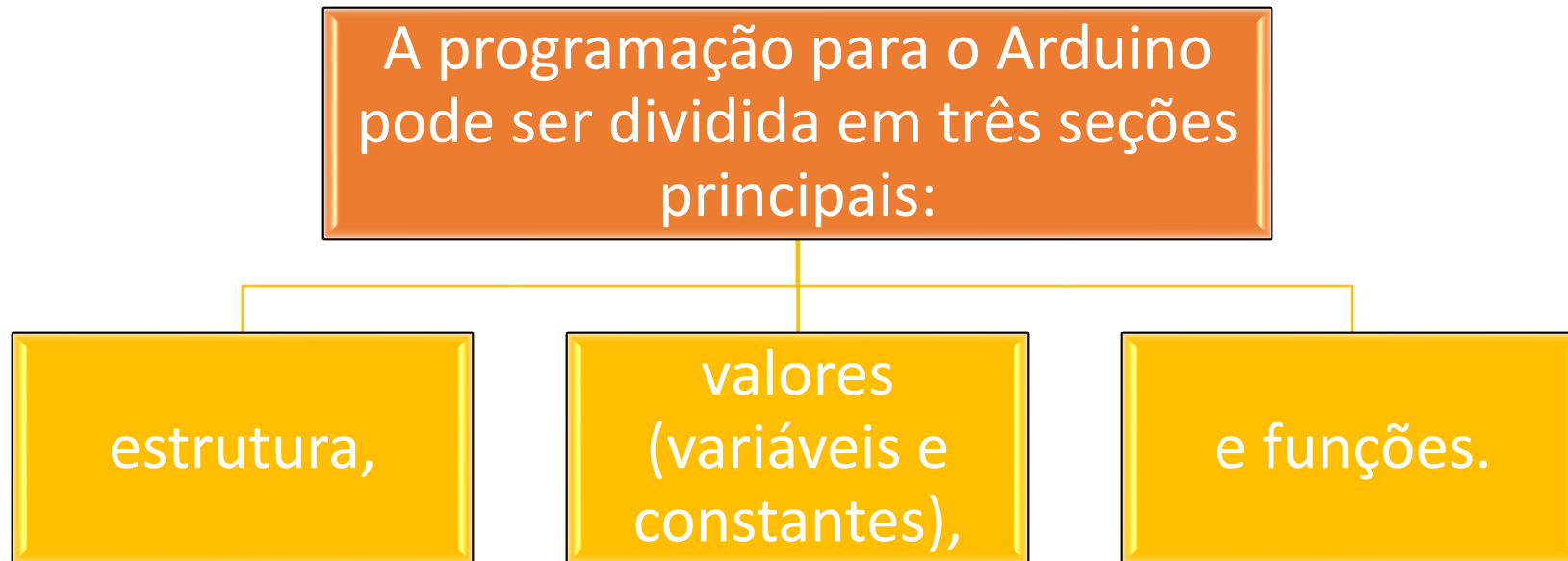
# Estrutura da Linguagem do Arduino (linguagem C)

# Referência

<https://www.arduino.cc/reference/pt/>

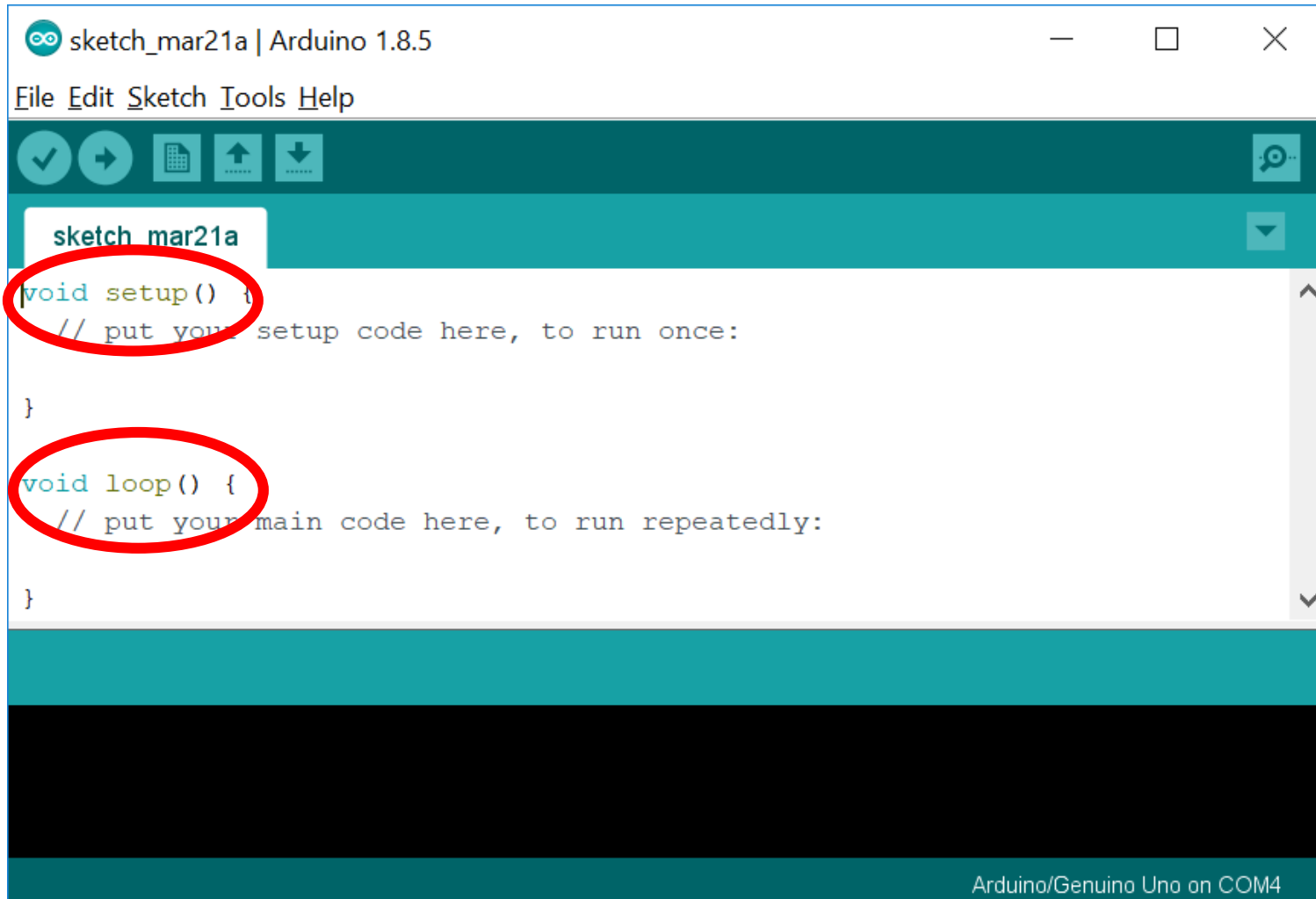


# Referência da linguagem



A linguagem Arduino é baseada em C/C++.

# Na IDE apresenta essa estrutura pronta



The screenshot shows the Arduino IDE interface. The window title is "sketch\_mar21a | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checkmark, right arrow, grid, up arrow, and down arrow. The main editor area shows the following code structure:

```
sketch_mar21a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

The code is displayed in a white editor area with a teal header bar above it. The text "sketch\_mar21a" is in the header bar. The code is in a monospaced font. The "void setup()" and "void loop()" lines are circled in red. The IDE has a teal header bar and a black footer bar. The footer bar contains the text "Arduino/Genuino Uno on COM4".

# Referência da linguagem

## Estrutura

- void [setup\(\)](#)
  - É chamada no momento em que o programa começa.
  - É usada para inicializar variáveis, definir os modos de entrada ou saída dos pinos, indicar bibliotecas, etc.

# Referência da linguagem

## Estrutura

- void [loop\(\)](#)
  - Esta função é executada após o `setup()` e ela repete-se continuamente permitindo que seu programa funcione dinamicamente.
  - É utilizada para controlar de forma ativa a placa Arduino.

2 - Botão de reset

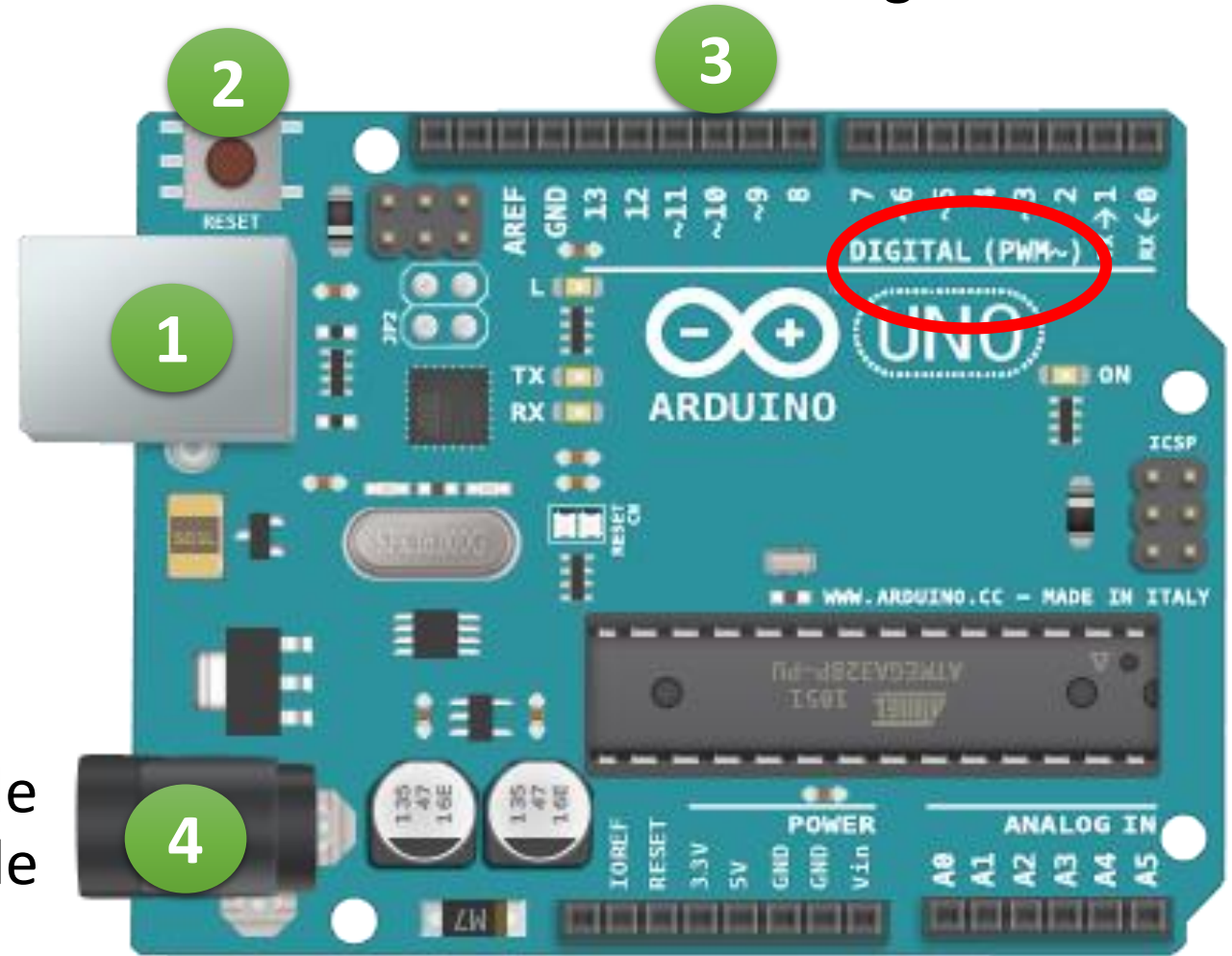
3 - Pinos de entrada e saída digital e PWM

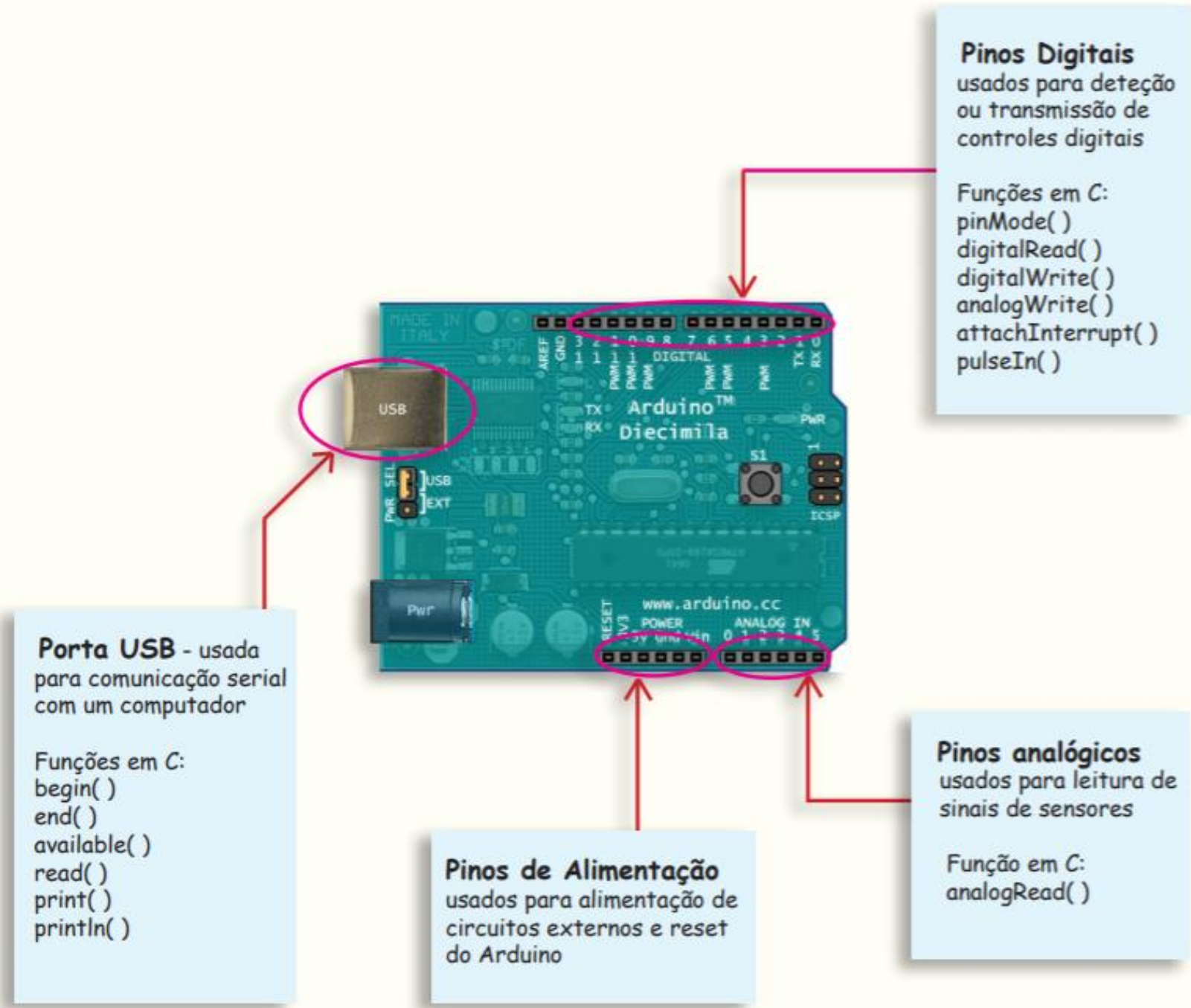
1 - Conector USB para o cabo tipo AB

4 - Conector de alimentação de energia

5 - Pinos de voltagem e terra

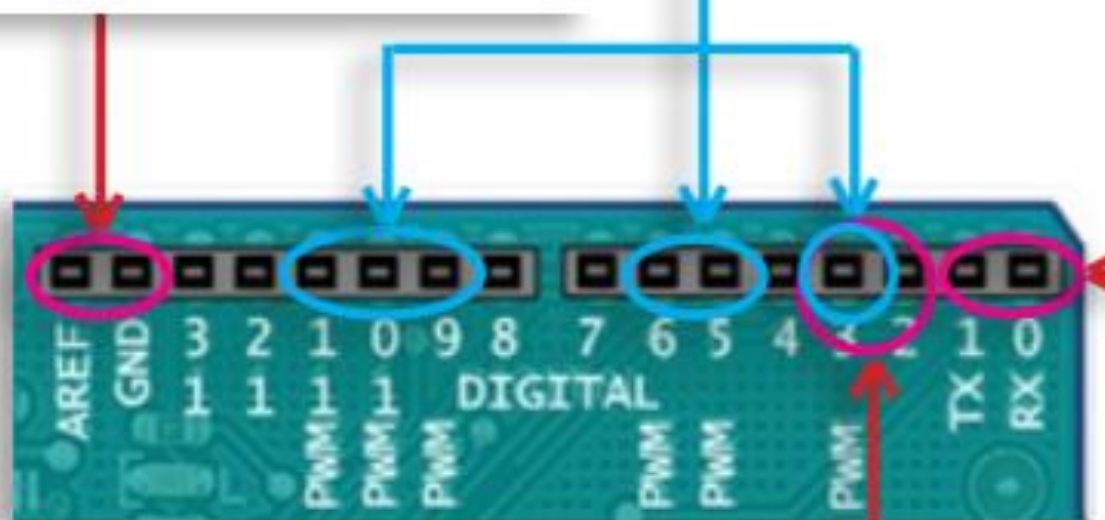
6 - Entradas analógicas





**Pinos AREF e GND:** o pino AREF é a entrada de tensão de referência para o conversor A/D do Arduino; o pino GND é o terra, comum a todos os outros pinos.

**Pinos 3, 5 e 6 e 9 a 11 (PWM):** 6 pinos dos 14 pinos digitais podem ser usados para gerar sinais analógicos com a função `analogWrite( )` utilizando a técnica de Modulação por Largura de Pulso (PWM).



**Pinos 0 e 1:** os dois primeiros pinos digitais são conectados a USART do microcontrolador do Arduino para comunicação serial com um computador.

**Pinos 2 e 3:** pinos que chamam uma ISR (Interrupt Service Routine) para tratar uma interrupção com a função `attachInterrupt( )` nesses pinos.



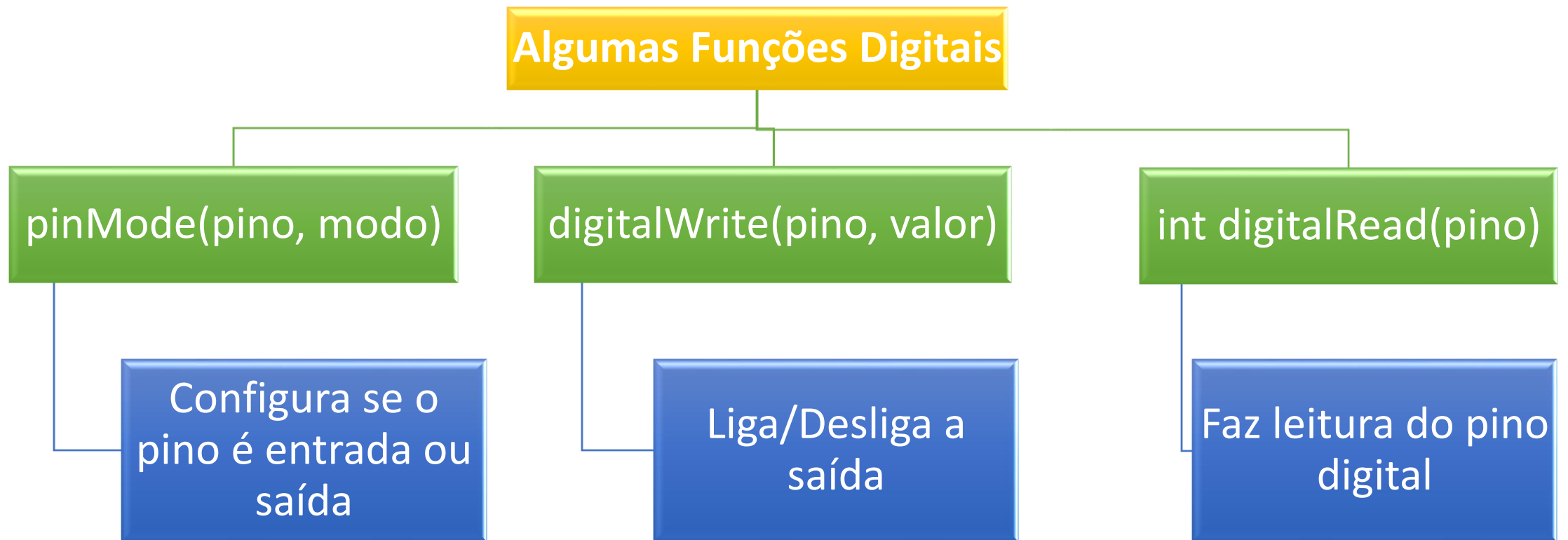
**Pinos 0 a 5:** esses 6 pinos aceitam tensões entre zero e 5 volts CC que vão ao conversor A/D de 10 bits no microcontrolador do Arduino. O pino AREF, numa das barras de pinos digitais, é a entrada de tensão de referência para esse conversor.



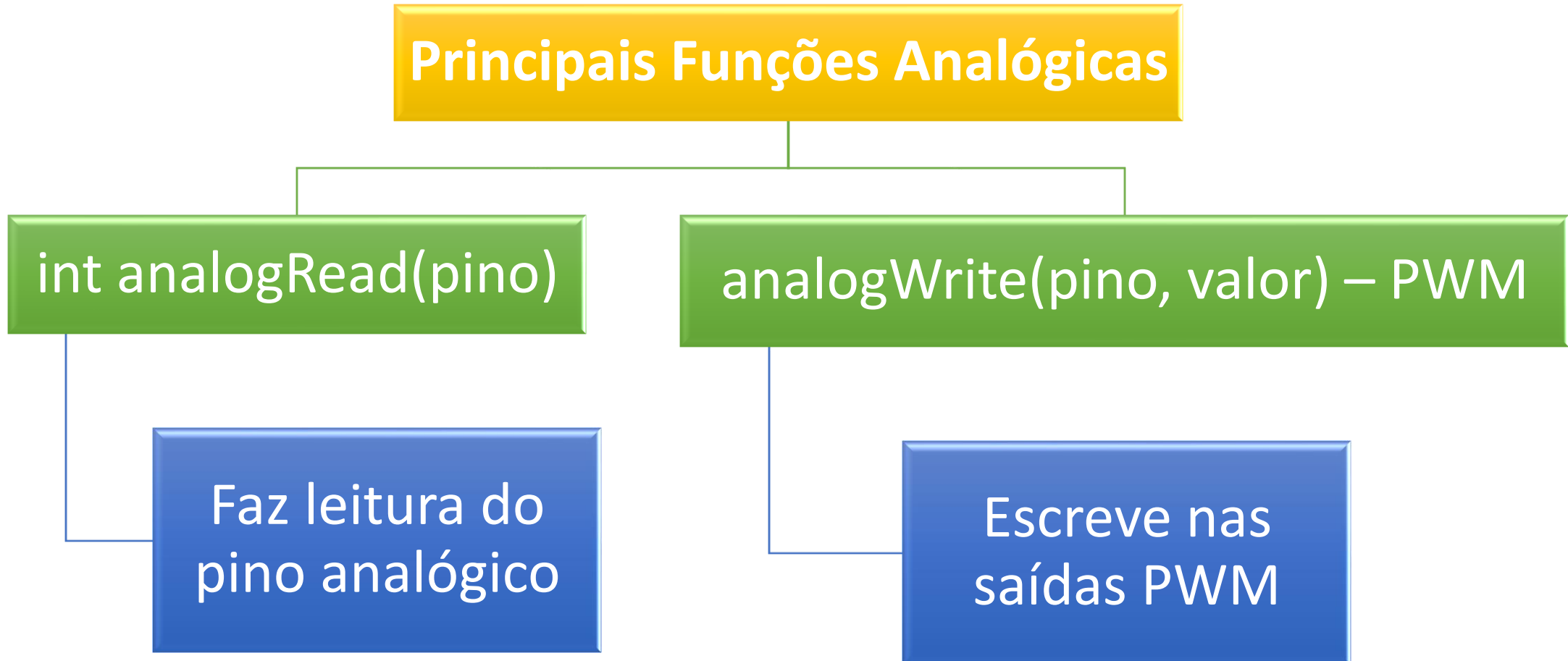


**Pinos 3V3, 5V e Gnd:** dos 6 pinos dessa barra somente os quatro do meio servem para alimentar um circuito externo conectado ao Arduino: o pino de 5V e o terra (os dois pinos Gnd entre 5V e Vin); e o pino 3V3 que disponibiliza essa tensão com uma corrente máxima de 50mA.

# Referência da linguagem



# Referência da linguagem



# Referência da linguagem

## Temporizadores

- `delay(ms)`
  - Pausa por X milissegundos
- `delayMicroseconds(us)`
  - Pausa por X microsegundos

# Referência da linguagem

## Estruturas de controle

- if
- if...else
- for
- switch case
- while

# Referência da linguagem

## Tipos de dados

- boolean
- char
- byte
- int / unsigned int
- long / unsigned long
- float / double
- string
- array
- void

# Exemplo Blink

```
int ledPino = 10;
```

Variável Global

```
void setup() {
```

```
  pinMode(ledPino, OUTPUT);
```

Configura o pino como saída

```
}
```

```
void loop() {
```

```
  digitalWrite(ledPino, HIGH);
```

Liga o Pino 10

```
  delay(1000);
```

Pausa por 1 segundo

```
  digitalWrite(ledPino, LOW);
```

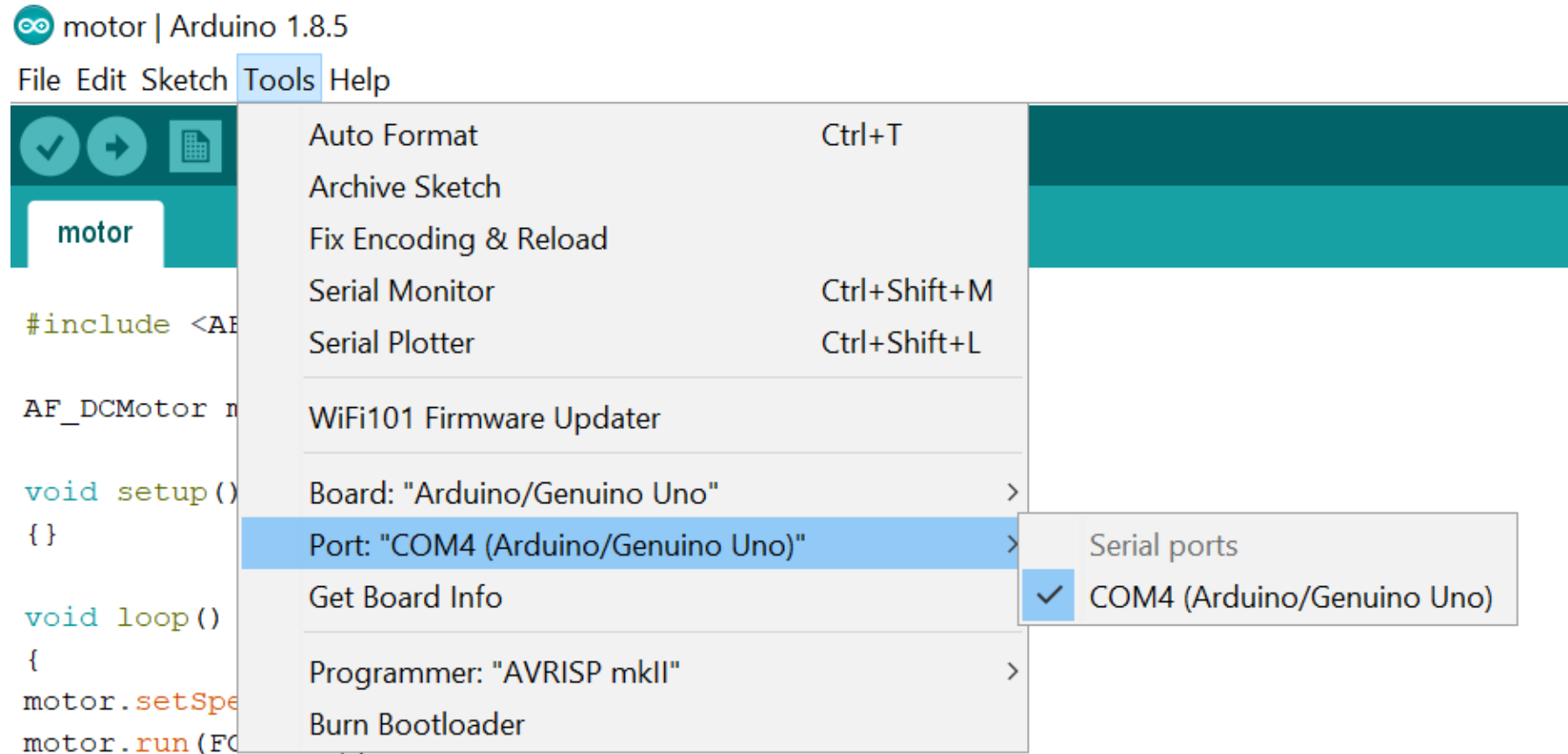
Desliga o Pino 10

```
  delay(1000);
```

Pausa por 1 segundo

```
}
```

# Configure a porta do computador para o Arduino

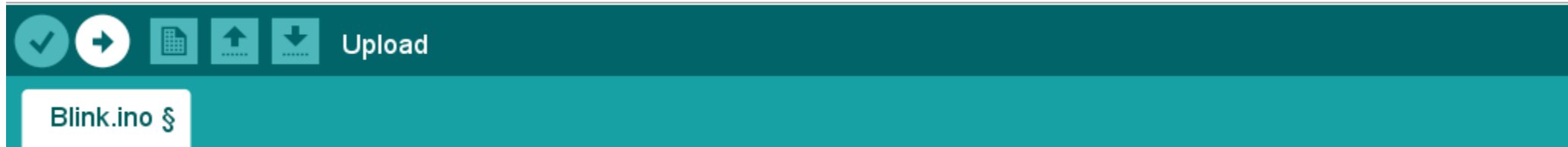




# Na IDE do Arduino

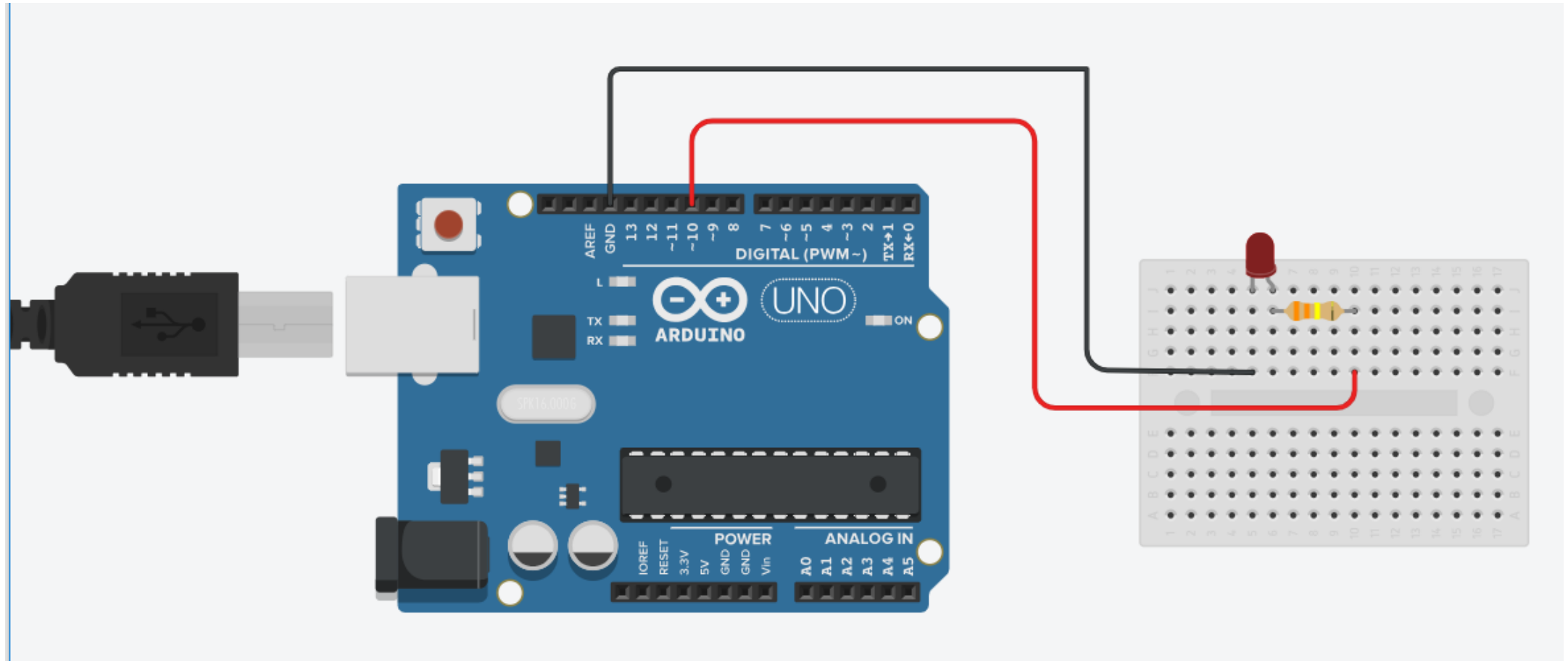
∞ Blink | Arduino 1.8.5

File Edit Sketch Tools Help



```
//Este código de exemplo está no domínio público.  
// a função de configuração é executada uma vez quando você pressiona reset ou liga a placa  
  
int ledPin = 10; // Cria uma variavel inteira com o numero da porta de entrada para  
                //controle do led como sendo a porta 10  
  
void setup() {  
  // inicialize o pino digital ledPin como uma saída.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// a função loop é executada repetidamente para sempre  
void loop() {  
  digitalWrite(ledPin, HIGH); // liga o LED (HIGH é o nível de voltagem)  
  delay(1000);                // espera por um segundo (1000 millessimos)  
  digitalWrite(ledPin, LOW); // desliga o LED fazendo a tensão LOW  
  delay(1000);                // espera por um segundo  
}
```

# Prática: na placa física



## Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C



Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha

Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

# As portas de E/S do Arduino e suas funções em C

# Referência

<https://www.arduino.cc/reference/pt/>

# pinMode(IN1,OUTPUT);

## Descrição

Configura o pino especificado para que se comporte ou como uma entrada (input) ou uma saída (output).

## Sintaxe

```
pinMode(pin, mode)
```

## Parâmetros

pin: o número do pin o qual você deseja predeterminar.

mode: pode ser [INPUT](#) ou [OUTPUT](#)

# analogWrite()

## Descrição

Aciona uma onda PWM em um pino.

A frequência do sinal PWM na maioria dos pinos é aproximadamente 490 Hz.

No Uno e placas similares, pinos 5 e 6 usam uma frequência de aproximadamente 980 Hz.

## Sintaxe

```
analogWrite(pino, valor)
```

## Parâmetros

pino: o pino escolhido. Tipos de dados permitidos: int.

valor: o duty cycle: entre 0 (sempre desligado) and 255 (sempre ligado). Tipos de

# digitalWrite()

## Descrição

Aciona um valor HIGH ou LOW em um pino digital.

Se o pino for configurado como saída (OUTPUT) com a função pinMode(), sua tensão será acionada para o valor correspondente: 5V (ou 3.3V em placas alimentadas com 3.3V como o DUE) para o valor HIGH, 0V (ou ground) para LOW.

Se o pino for configurado como entrada (INPUT), a função digitalWrite() irá ativar (HIGH) ou desativar (LOW) o resistor interno de pull-up no pino de entrada. É recomendado configurar pinMode() com INPUT\_PULLUP para ativar o resistor interno de pull-up.

## Sintaxe

```
digitalWrite(pino, valor)
```

## Parâmetros

pino: o número do pino

valor: HIGH ou LOW

## Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física



- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha

Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

# Conceitos básicos de Física



# Tópicos

Tensão, corrente & resistência

Componentes básicos

Conhecendo  
os sensores e  
atuadores

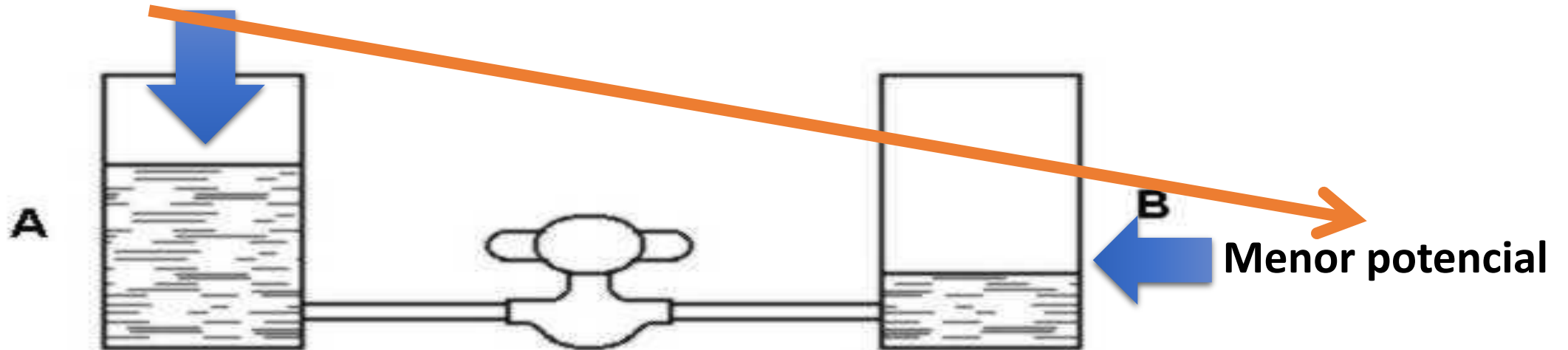
- Motores e Ponte H
- Sensores de Linha

# Tensão, corrente & resistência

# Tensão

- Medida: Volt (V)
- Símbolo em fórmulas:  $V$

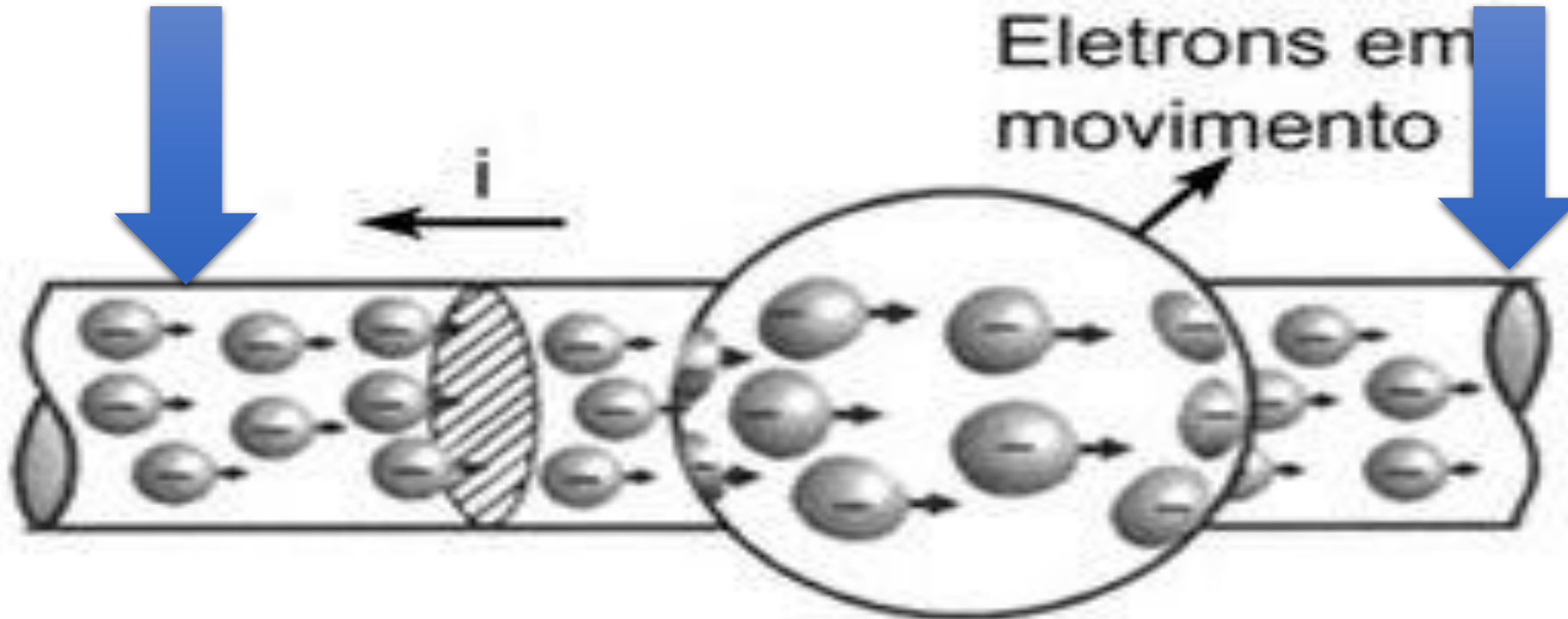
**Maior potencial**



# Tensão

Maior potencial

Menor potencial



movimento ordenado dos eletrons em um fio

# Prática: Medindo tensão

## Polaridade

- Nas medidas de tensões alternadas não é preciso observar a polaridade das pontas de prova.
- No entanto, na medida de tensões contínuas, a polaridade deve ser observada.
- A ponta vermelha (ligada ao +) deve ser colocada no ponto mais positivo do circuito e a preta no mais negativo ou em zero volt (terra),

## Multímetro



# Alternada ou contínua?

## a) Tensões alternadas:

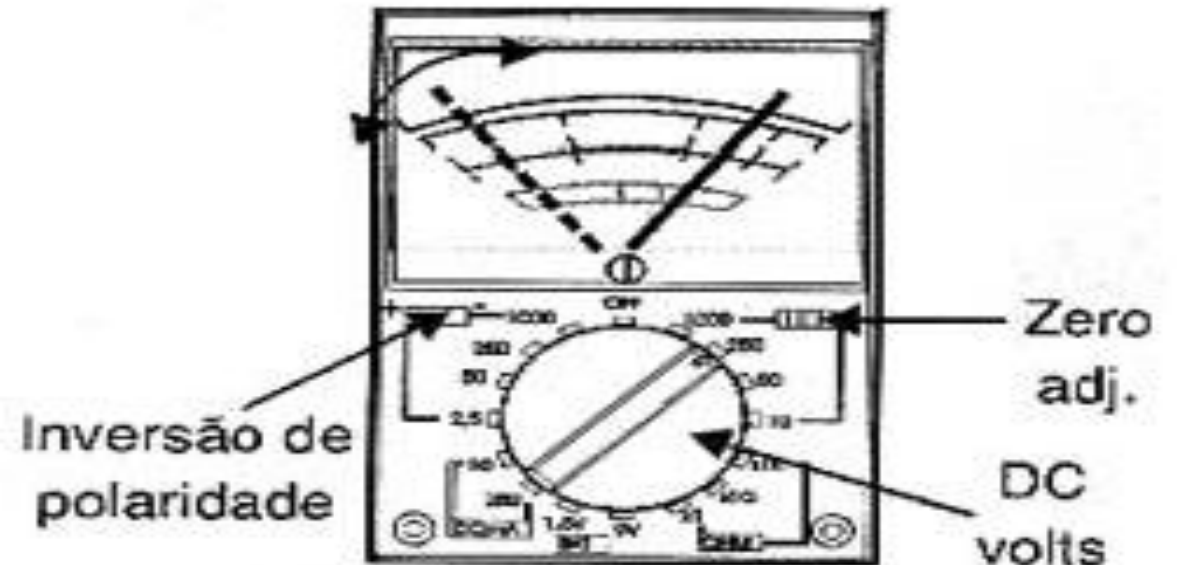
- Na entrada da rede de energia
- Nos interruptores e fusíveis da rede de energia
- Nos enrolamentos primários e secundários dos transformadores de fontes de alimentação ligadas à rede de energia
- Nos circuitos que operam com TRIACS e SCRs ligados à rede de energia

## b) Tensões contínuas

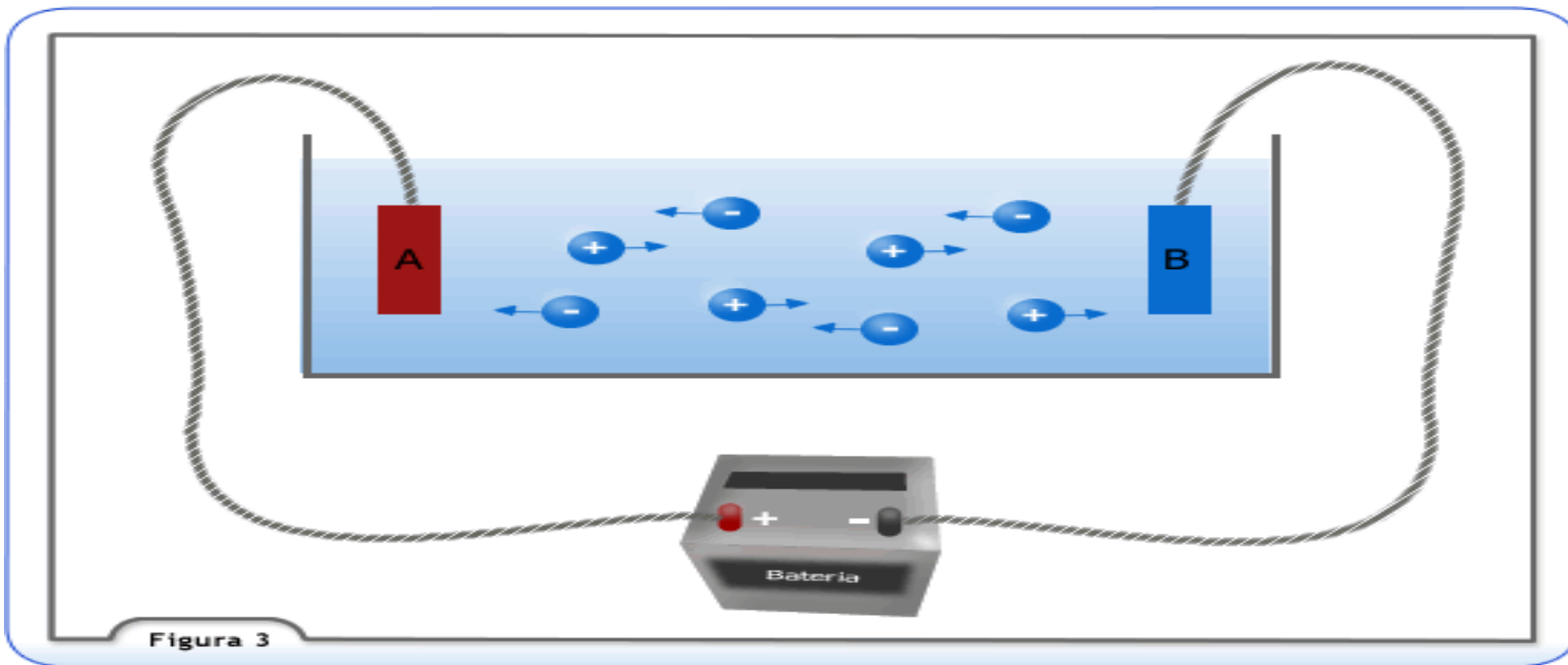
- Depois de diodos retificadores de fontes de alimentação
- Em pilhas e baterias
- Em circuitos com válvulas e transistores nos elementos de polarização
- Na alimentação de **circuitos integrados**
- Nas saídas de eliminadores de pilhas e fontes
- No circuito elétrico de um automóvel

# Polaridade invertida

- Se houver inversão das pontas de prova, nestas condições, a agulha tende a se movimentar em sentido contrário ao desejado e isso pode ser ruim para o instrumento, pois seu mecanismo é forçado.

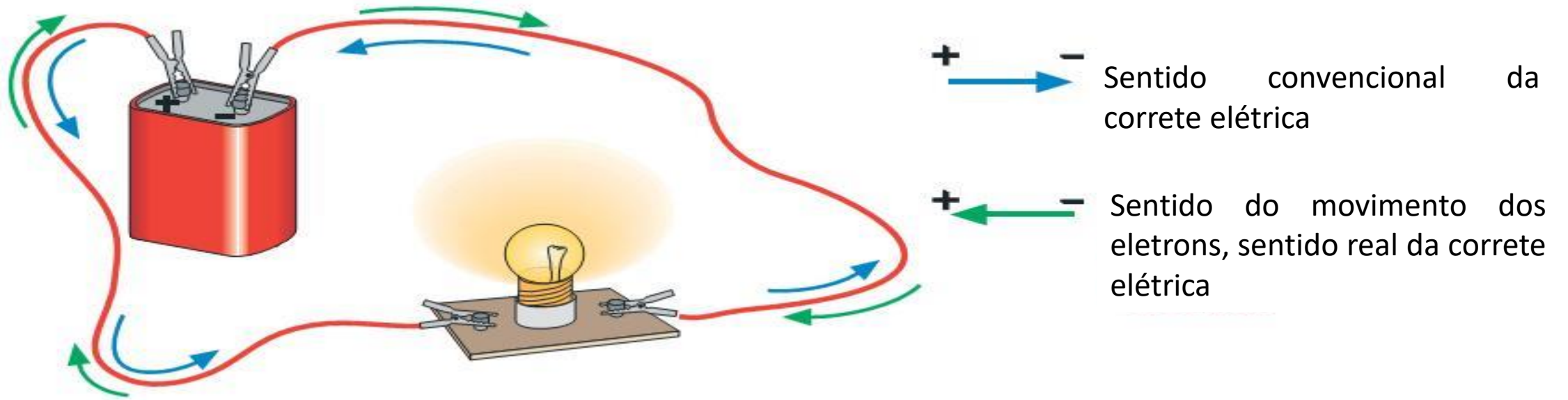


# Fluxo de corrente



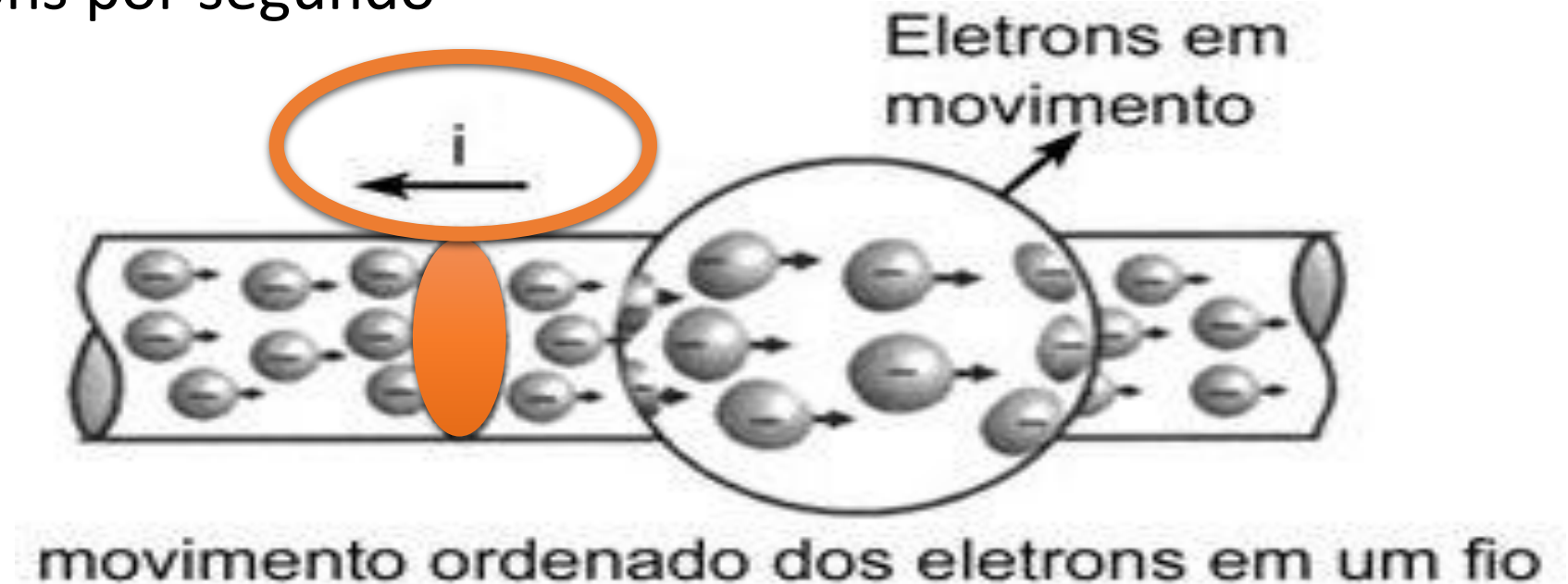


# Corrente elétrica



# Corrente elétrica

- Medida: Ampère (A)
- Símbolo em fórmulas:  $I$
- $6.241 \times 10^{18}$  elétrons por segundo



# Amperagem

**Maior Amperagem**



**Menor Amperagem**

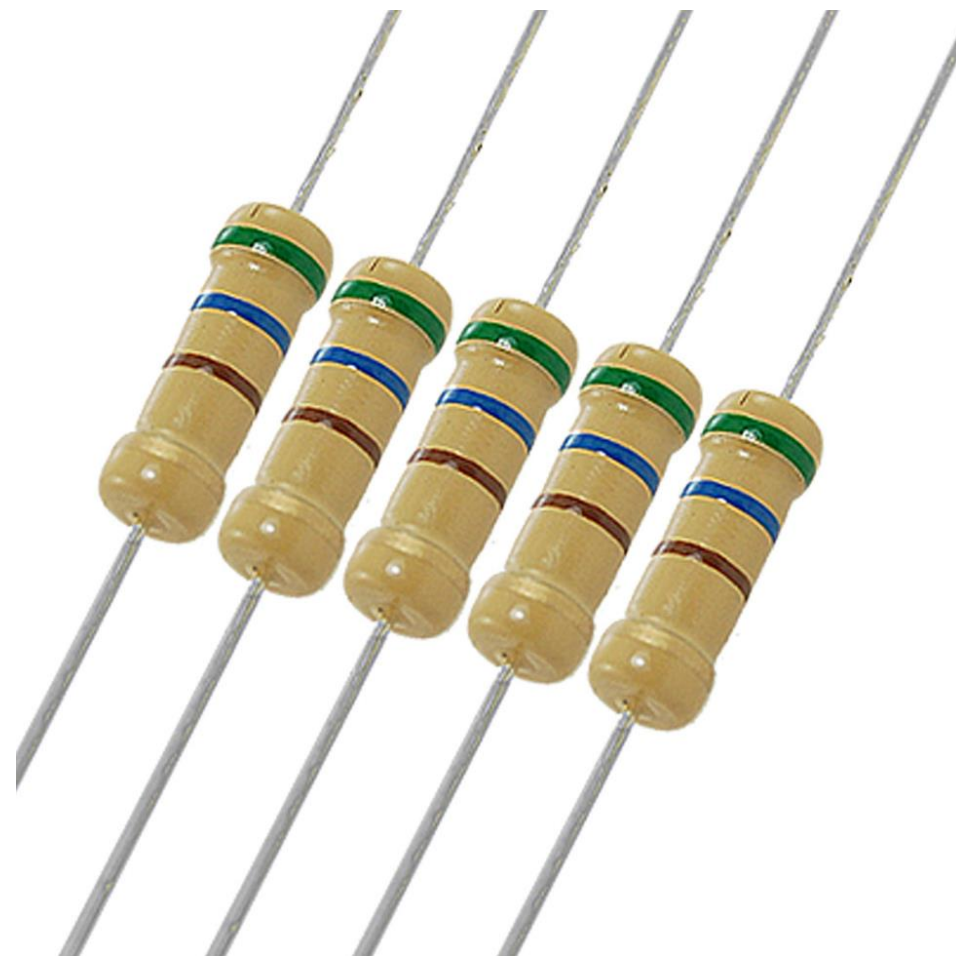
# Resistencia/resistor

É um componente formado por carbono e outros elementos resistentes usados para limitar a corrente elétrica em um circuito.

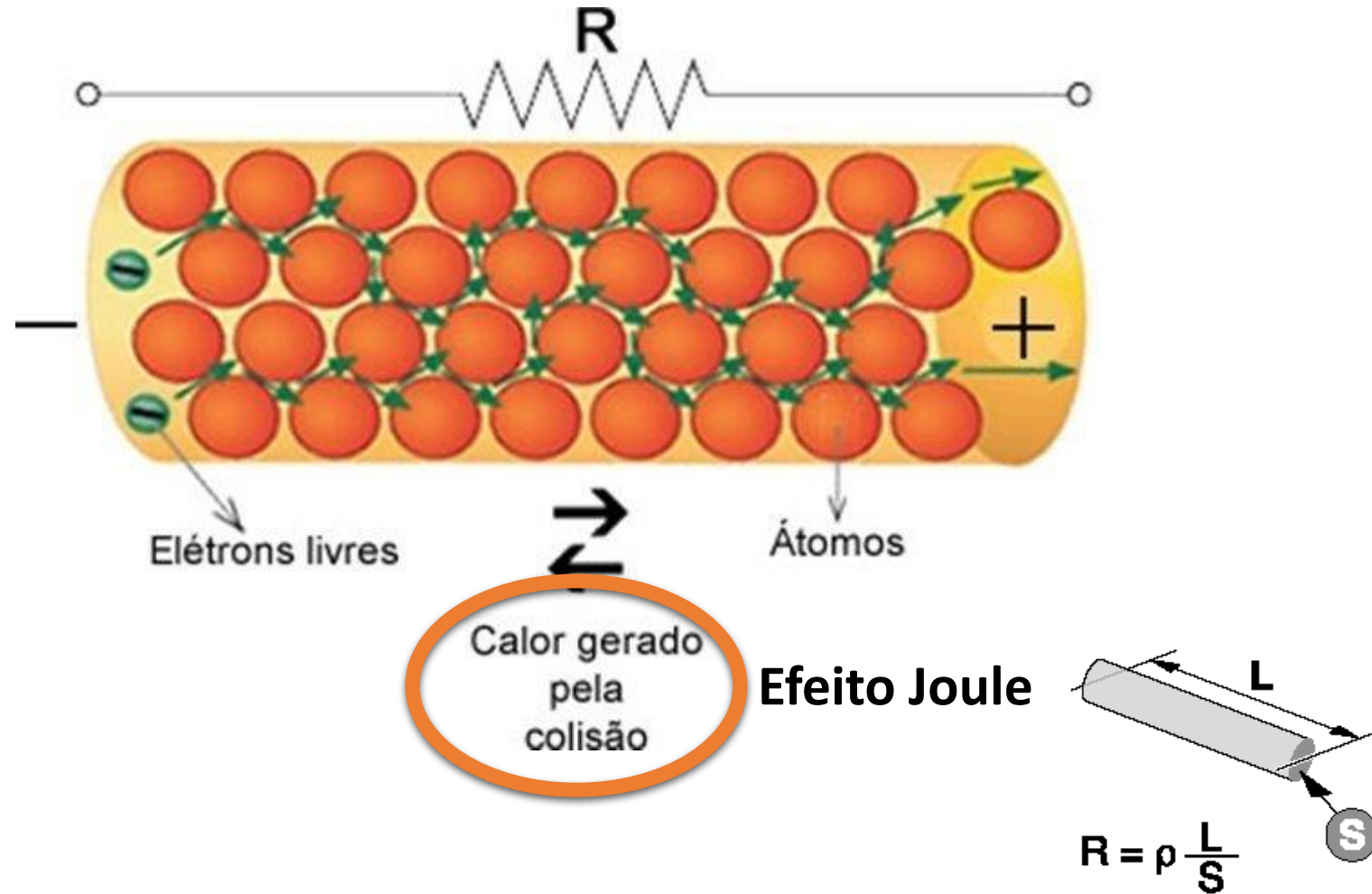
Sem polaridade: pode ser montado em qualquer direção

Valor em Ohms ( $\Omega$ )

Símbolos em esquemas:



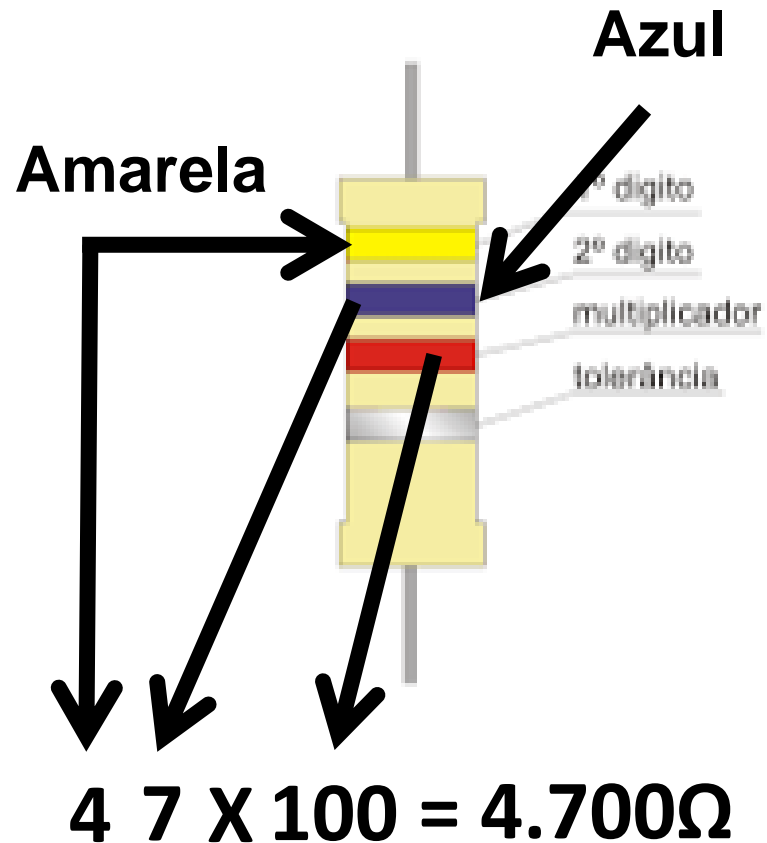
# Resistencia



# Tabela de cores dos resistores

COR	1ºANEL	2ºANEL	3ºANEL	4ºANEL
Preto	-	0	x1	-
Marrom	1	1	x10	1%
Vermelho	2	2	x100	2%
Laranja	3	3	x1000	3%
Amarelo	4	4	x10000	4%
Verde	5	5	x100000	-
Azul	6	6	x1000000	-
Violeta	7	7	-	-
Cinza	8	8	-	-
Branco	9	9	-	-
Prata	-	-	x0,01	10%
Dourado	-	-	x0,1	5%

# Exemplo



**(quarto mil e setecentos ohms)**

Neste exemplo, temos:

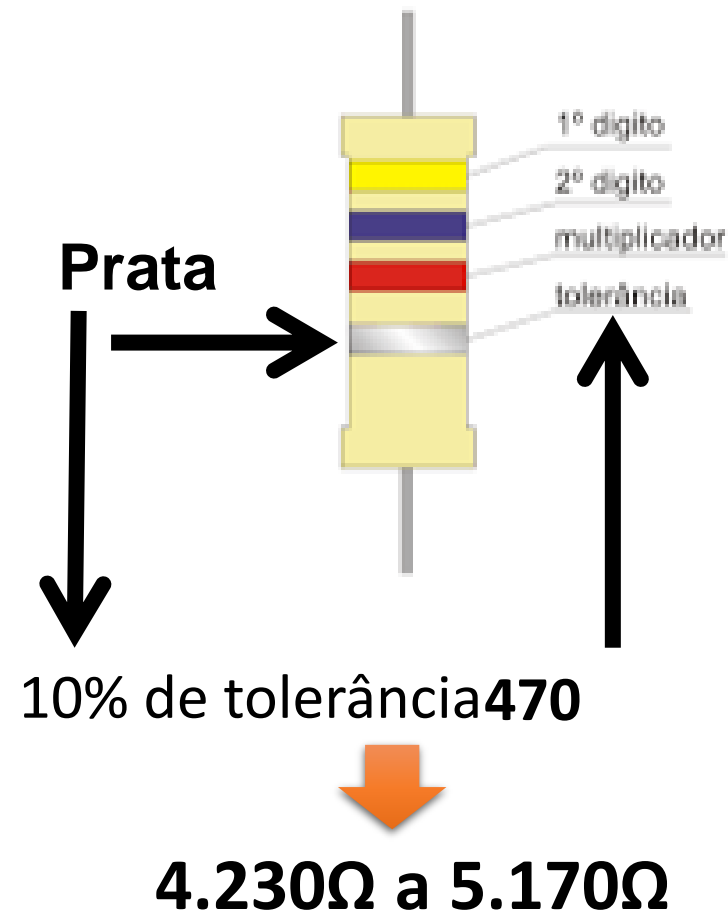
primeira cor o amarelo, que corresponde ao numero 4.

A segunda faixa é de cor violeta, que corresponde ao numero 7, formando assim os dígitos 47.

A terceira faixa é a multiplicadora, e é vermelha, então o valor 47 é multiplicado por 100 obtendo o valor da resistência de 4.700Ω (quarto mil e setecentos ohms).

# Exemplo

- A quarta e última faixa é da tolerância, ou seja, quanto o valor real do **resistor** pode variar para mais ou para menos de acordo com o valor real.
- O **resistor** de exemplo tem o valor de  $4.700\Omega$  e a faixa de tolerância é prata correspondendo a 10% de tolerância.
- Ao se medir esse resistor para saber sua resistência, ele pode apresentar um valor 10% maior ou 10% menor que o valor teórico.
- 10% de 4.700 é 470, então o resistor que indica o valor de  $4.700\Omega$  quando medido pode apresentar um valor entre  $4.230\Omega$  a  $5.170\Omega$ .





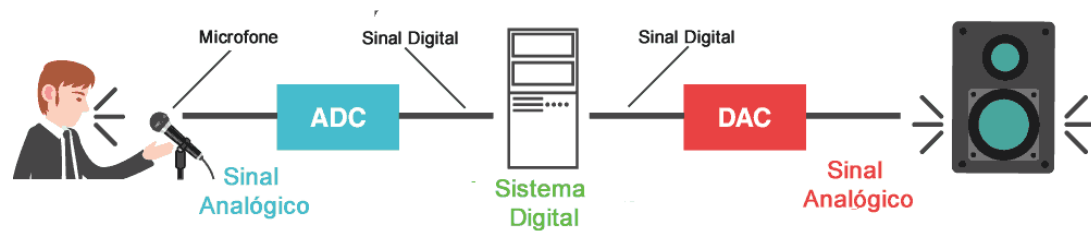
# Exercícios



0	$\times 10^0$
1	$\times 10^1$
2	$\times 10^2$
3	$\times 10^3$
4	$\times 10^4$
5	$\times 10^5$
6	$\times 10^6$
7	$\times 10^7$
8	$\times 10^8$
9	$\times 10^9$

# Componentes básicos

# Sinais analógicos e digitais



Um sinal analógico é aquele que possui o seu valor variando continuamente com resolução infinita em tempo e magnitude.

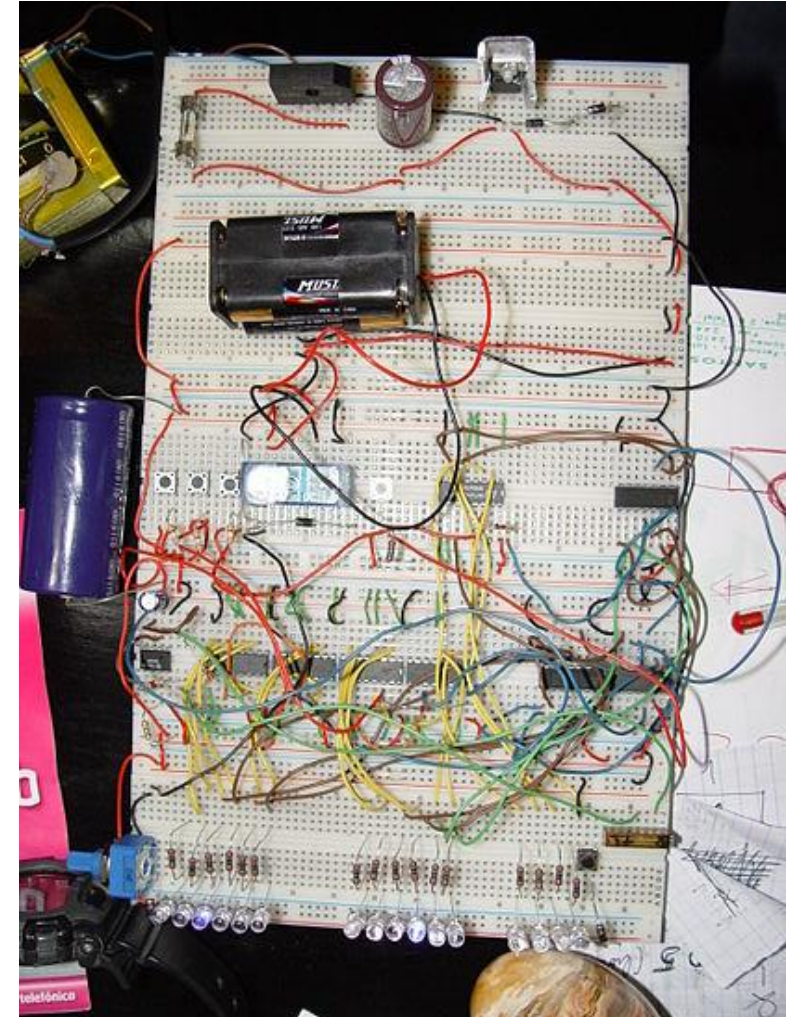
Os sinais analógicos são distinguíveis dos sinais digitais pois estes últimos sempre tomam valores de um conjunto finito de possibilidades predeterminadas, como o conjunto (0V, 5V). Neste caso, ou está ligado (5V) ou desligado (0V).

# Protoboard ou breadboard

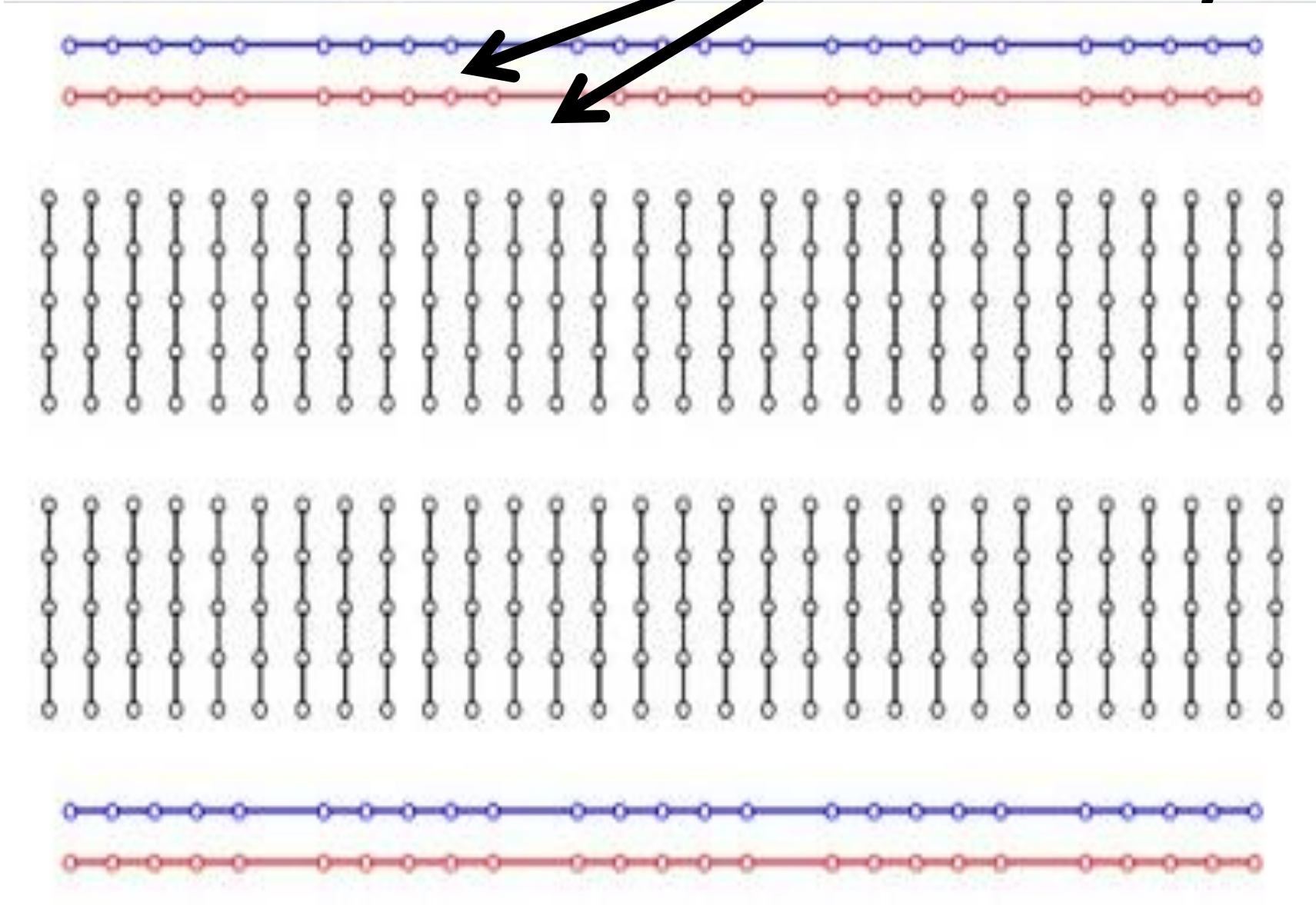
É uma placa de ensaio ou matriz de contato.

Placa com furos e conexões condutoras para montagem de circuitos elétricos experimentais.

Desvantagem é o seu "mau-contato".



linhas de alimentação

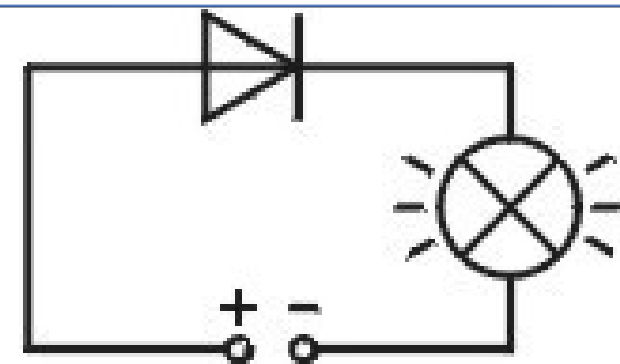
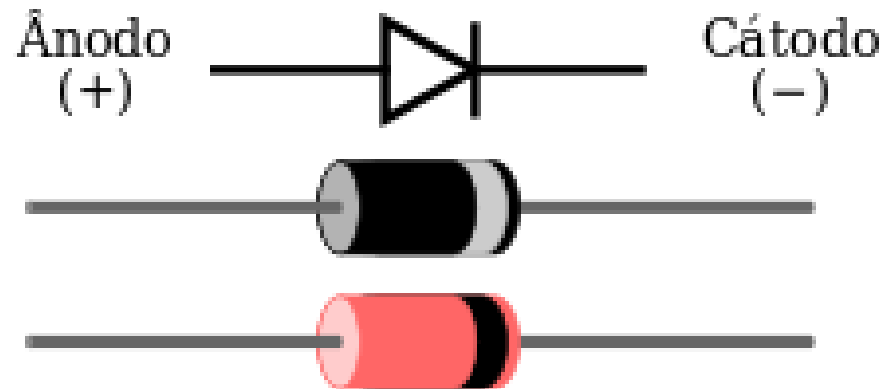


# Diodo

**Cátodo** é o eletrodo no qual há redução (ganho de elétrons). É o pólo **positivo** da pilha.

**Ânodo** é o eletrodo no qual há oxidação (perda de elétrons). É o pólo **negativo** da pilha.

Os elétrons saem do **ânodo** (pólo **negativo**) e entram no **cátodo** (pólo **positivo**) da pilha.



Polarização Direta

# LED: light-emitting diode

Diodo emissor de luz

Componente polarizado: tem direção certa para ligar

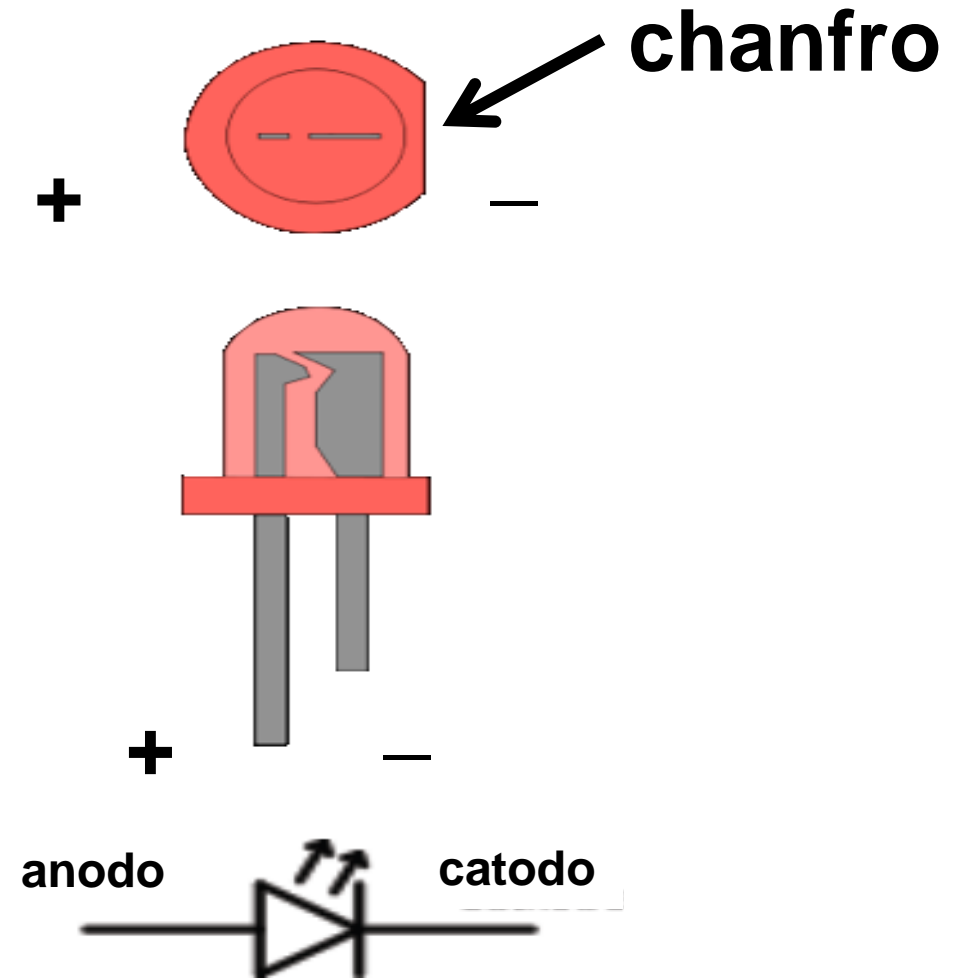


# Noções de eletrônica básica

Pino + (mais longo) ligado na fonte

Pino - ligado no terra

Ligar com resistor em série para proteção 30





## Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha



Montagem de um exemplo de carrinho segue linha

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

# Conhecendo os sensores e atuadores

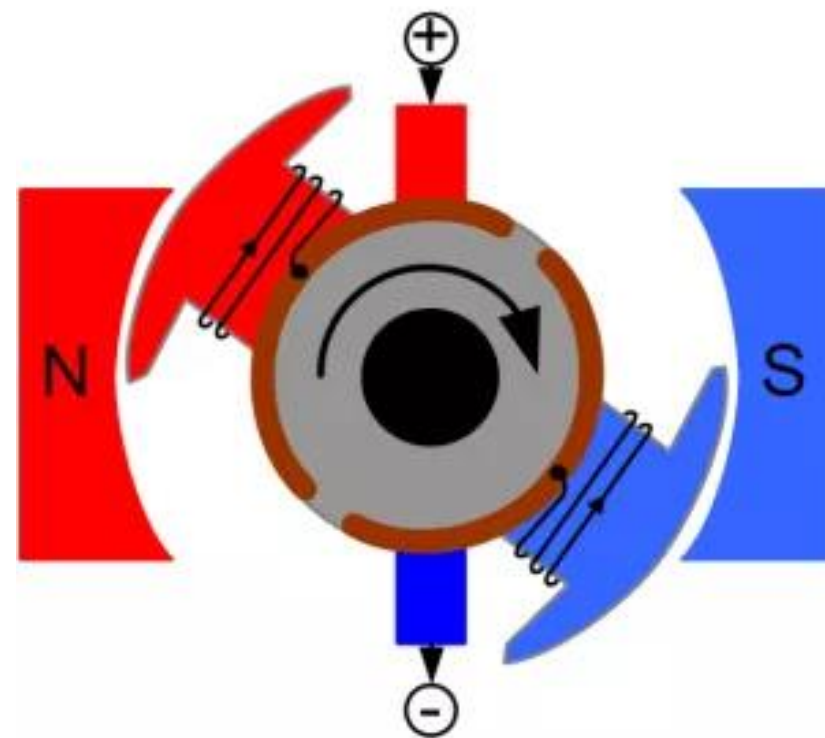
**Motores e Ponte H**  
**Sensores de Linha**

# Motores e Ponte H

# Como controlar a velocidade de um motor?

Um motor DC gira baseado em campos magnéticos gerados pela corrente que passa em suas bobinas.

Para variar a velocidade do motor podemos alterar essa corrente que é diretamente proporcional a tensão sobre elas.



# Como controlar a velocidade de um motor?

Dessa forma, com a mudança da tensão em cima do motor, teremos uma alteração de velocidade.

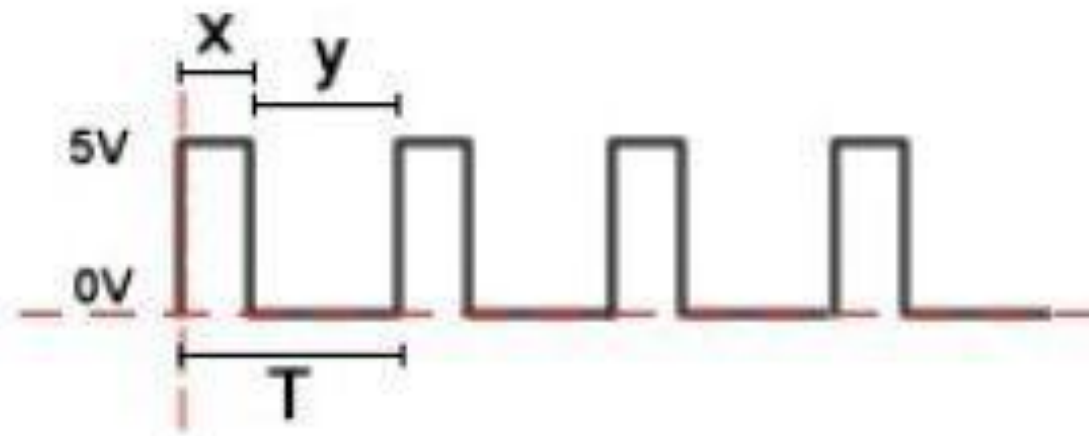
**Mas como podemos fazer isso usando o Arduino e a Ponte H?**

A solução é simples e eficiente e se chama PWM.

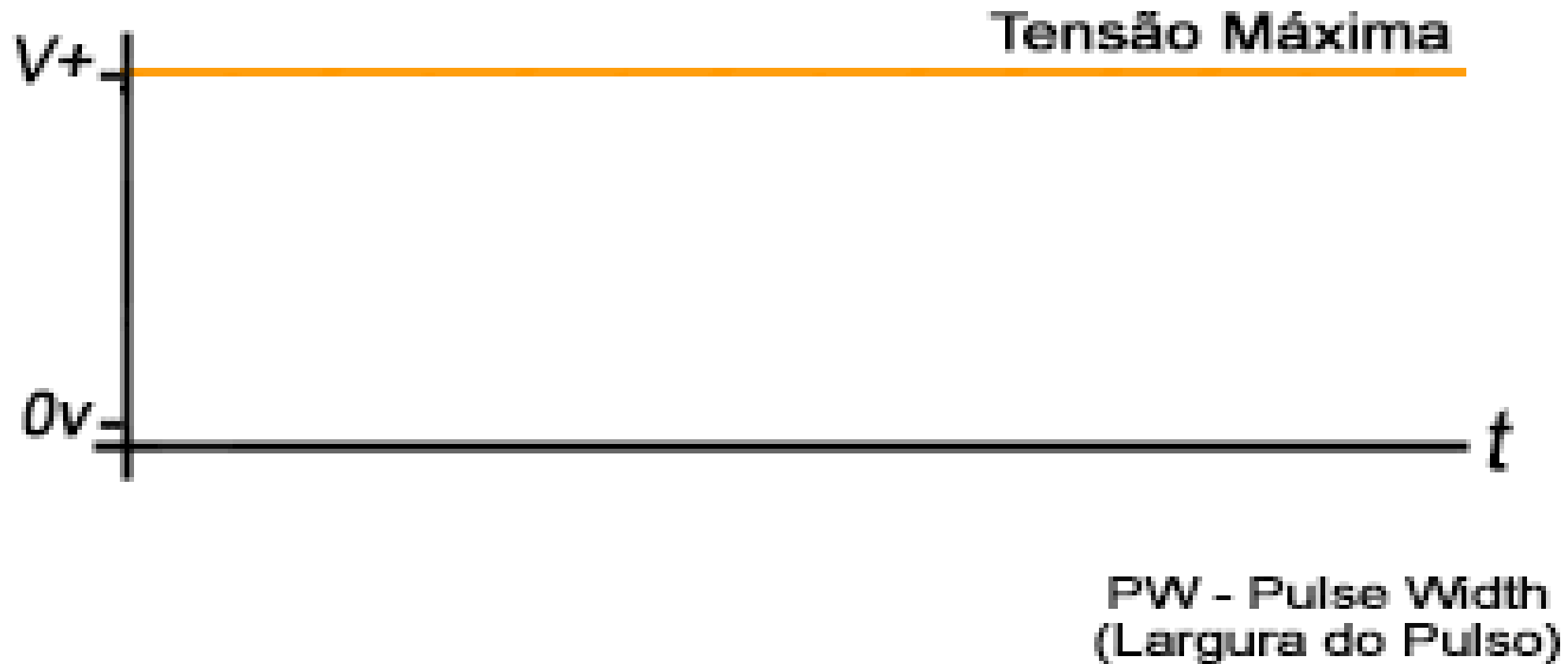
# O que é PWM?

## PWM (Pulse Width Modulation)

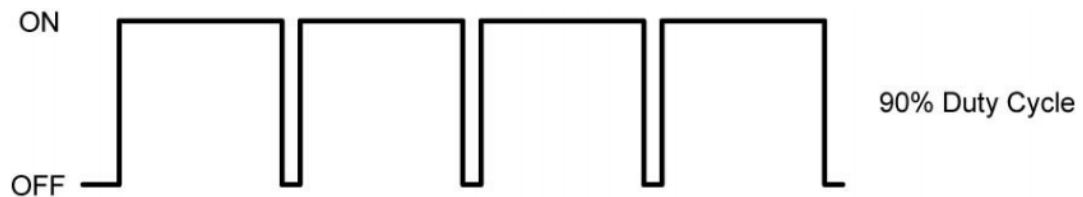
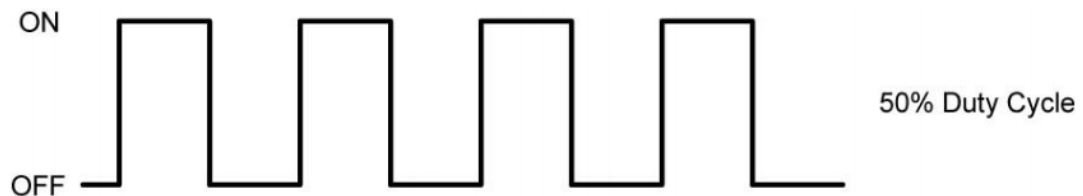
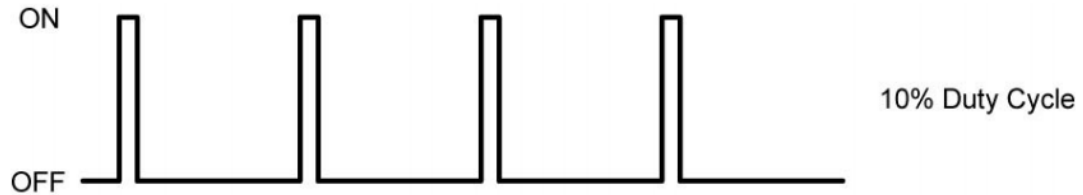
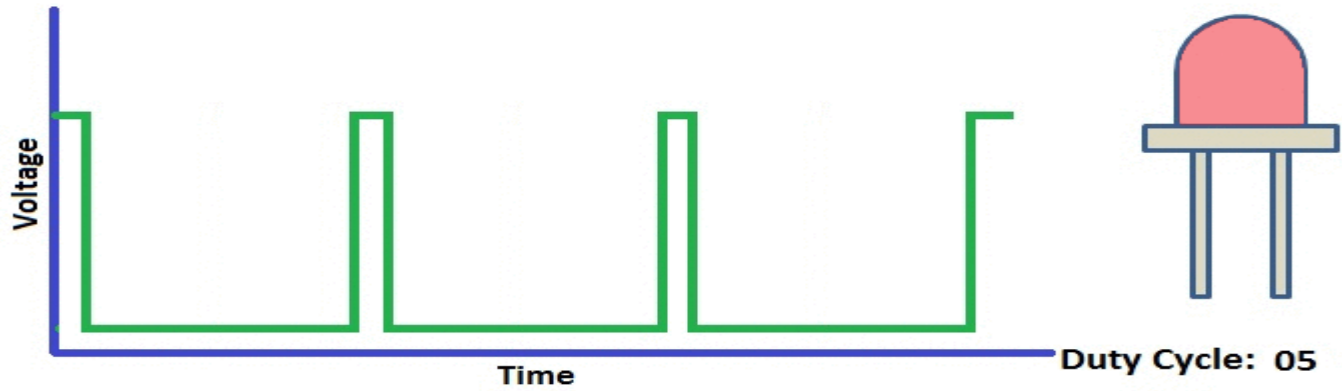
Essa técnica consiste na geração de uma onda quadrada em uma frequência muito alta em que pode ser controlada pela porcentagem do tempo em que a onda permanece em nível lógico alto.



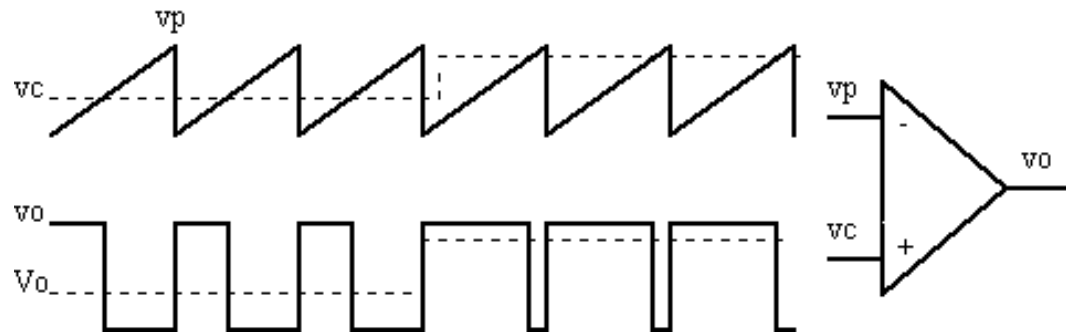
# Modulação



**Brilho do LED  
variando a largura  
do pulso.**



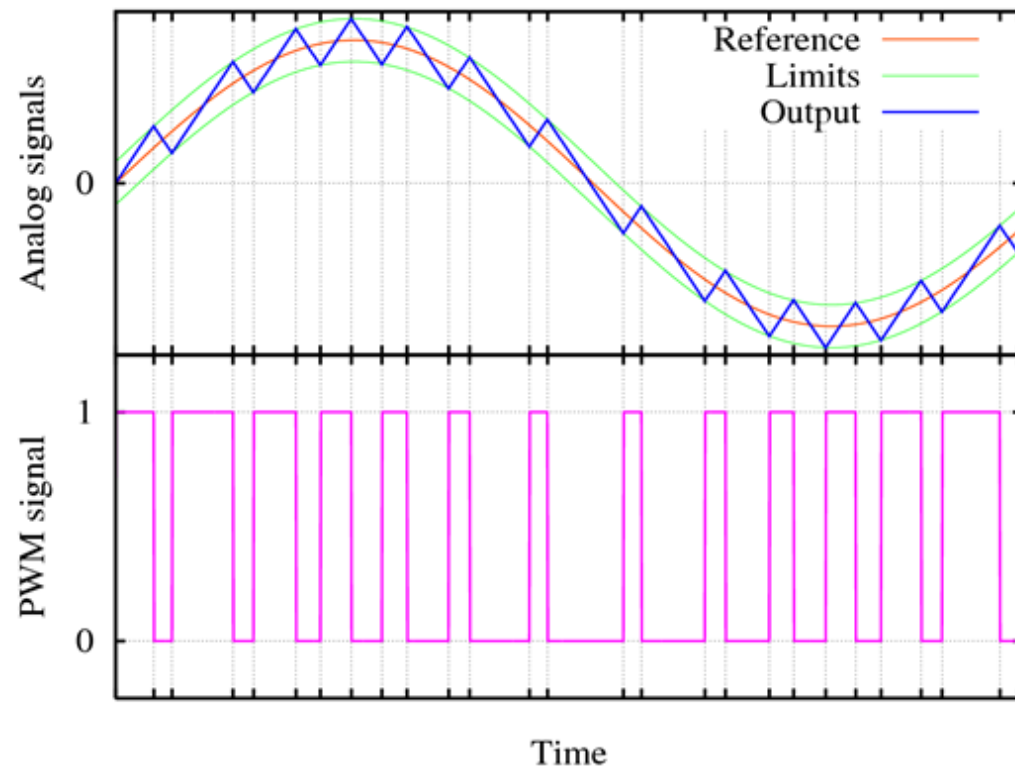
# Modulação



O sinal de comando é obtido, geralmente, pela comparação de um sinal de controle (modulante) com uma onda periódica (portadora) como, por exemplo, uma "dente-de-serra".

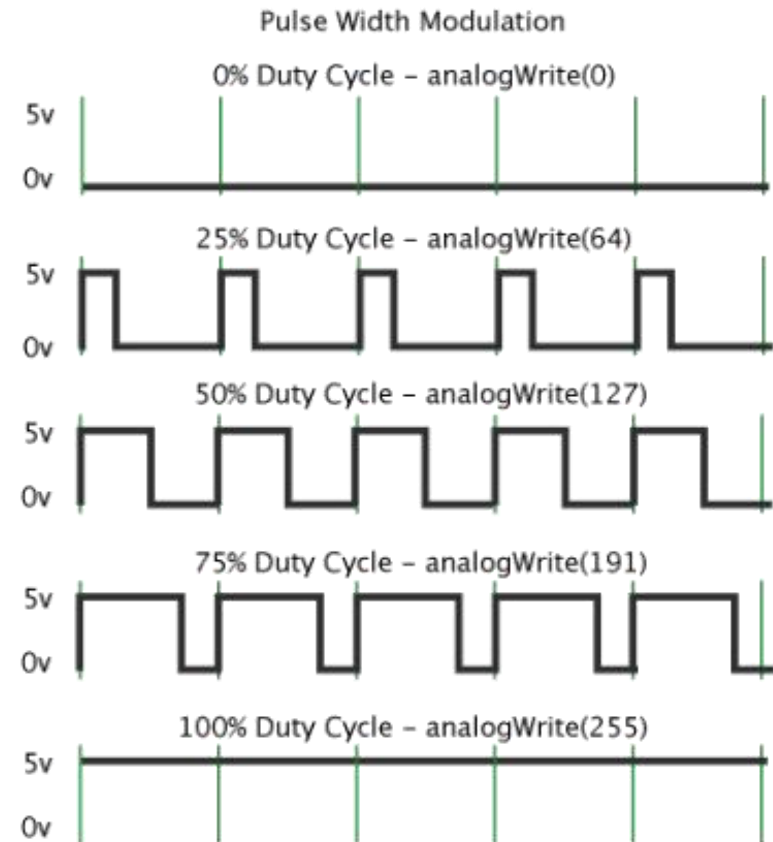


# PWM

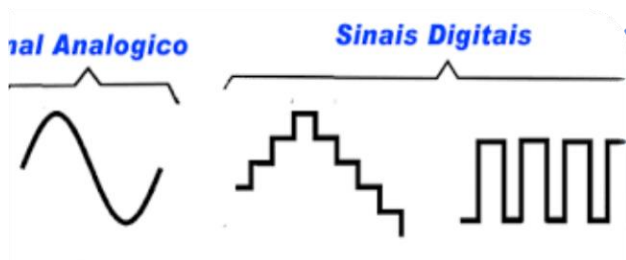


Os sinais PWM também podem ser usados para gerar sinais analógicos que variam no tempo e em conjunto a filtros passa-baixa podemos obter sinais analógicos fieis a um sinal que seja de fato analógico.

Esse tempo é chamado de Duty Cycle(Ciclo de trabalho) e sua alteração provoca mudança no valor médio da onda, indo desde 0V (0% de Duty Cycle) a 5V (100% de Duty Cycle) no caso do Arduino.



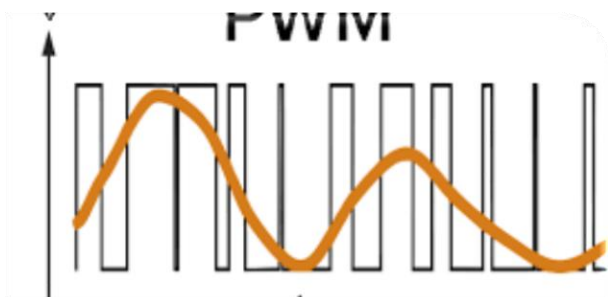
# PWM - Arduino



É usado para obter resultados analógicos por meios digitais



Essa técnica será a responsável pelo controle de velocidade do nosso carrinho.



Os valores de PWM variam de 0 (parado) até 255 (velocidade máxima ou 5 V).

# Ponte H



Este é um *shield* que integra tensão, corrente e controle de 4 canais em uma só placa.

O chip L293D é conhecido como um tipo de Ponte H que é tipicamente um circuito elétrico que permite uma tensão ser aplicada em uma carga em qualquer direção para uma saída, como por exemplo um motor.

É baseado no chip L293D e com ele é possível controlar até 4 Motores DC, 2 Servos ou 2 Motores de Passo.

# Especificações do chip: 293D

Pode controlar 4 Motores DC, 2 Motores de Passo ou 2 Servos.

Tensão de saída: 4,5-36V

Corrente de saída: 600mA por canal

Até 4 motores DC bi-direcional com seleção individual de velocidades de 8 bits (cerca de 0,5% de resolução).

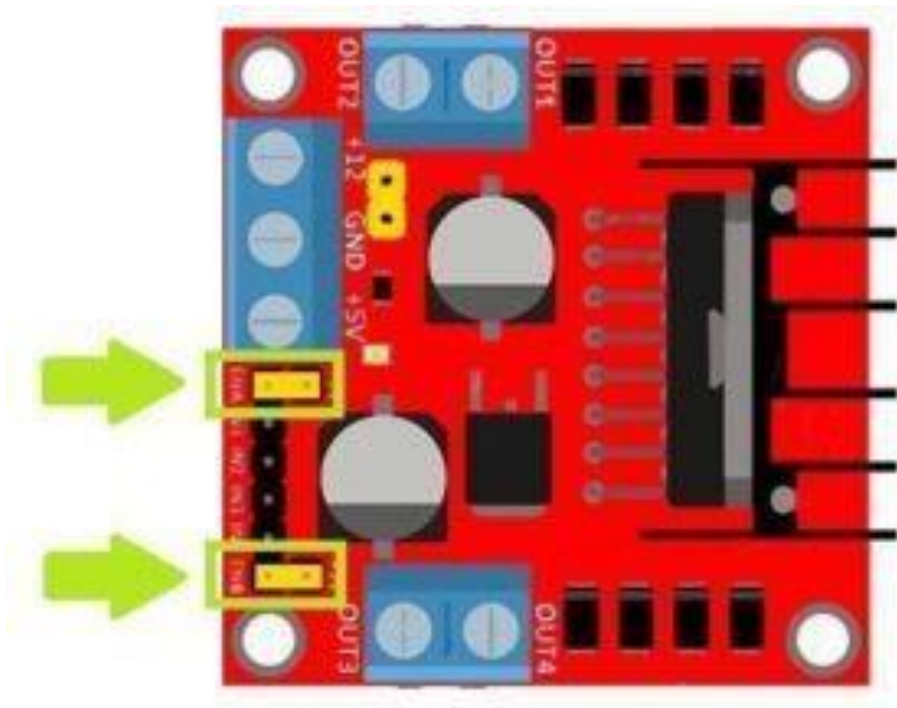
Até 2 Motores de Passo (Unipolar ou Bipolar) com bobina unica, dupla ou passos interlaçados.

4 Pontes H: 0,6A por Ponte (1,2A de pico) com proteção térmica e diodos de proteção contra retro-alimentação.

Resistores Pull Down mantem motores desativados durante a entrada de alimentação.

Terminais em bloco de 2 pinos e jumper para conexão de alimentação externa.

# L298N



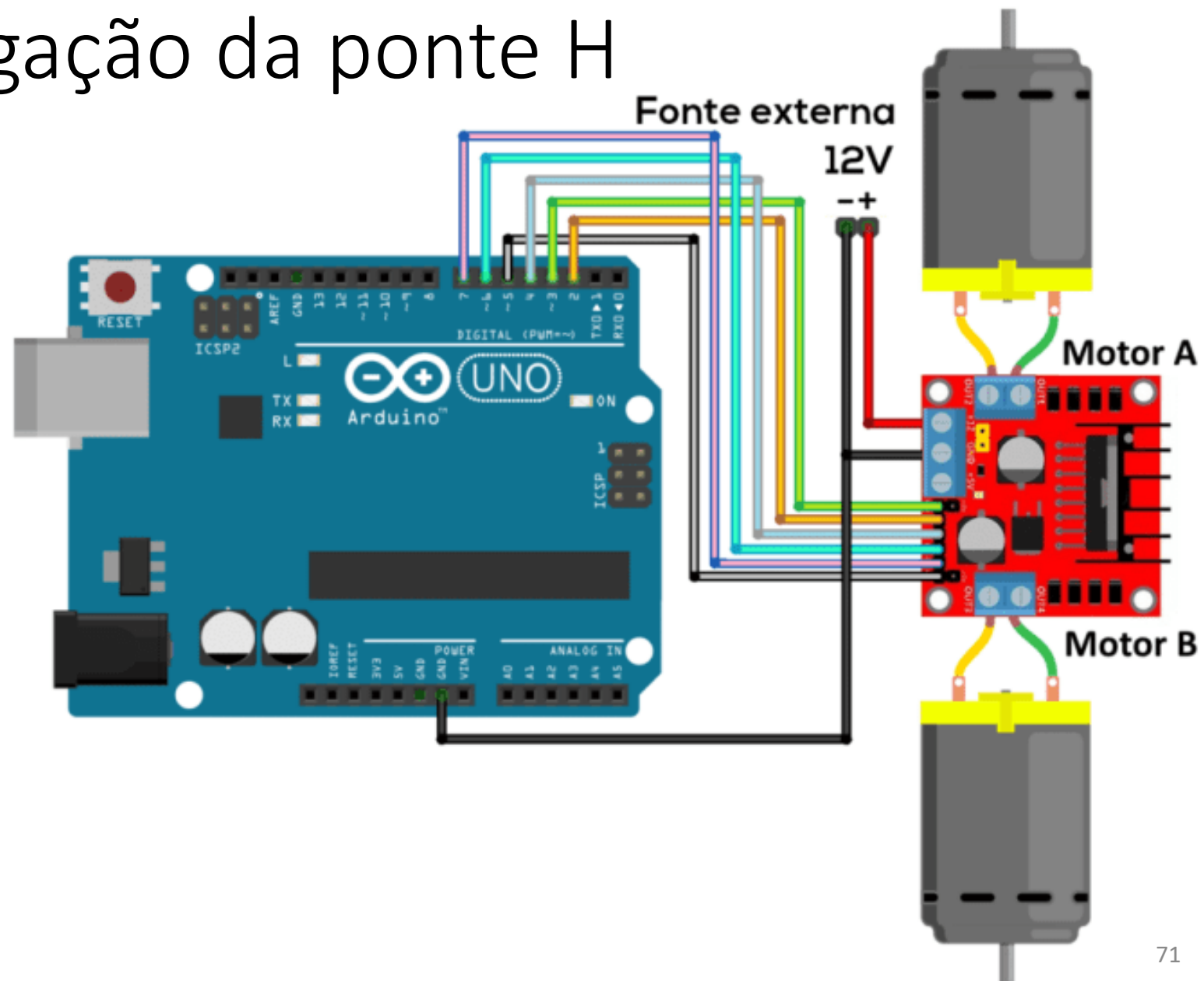
No módulo Ponte H com CI L298N cada ponte H possui um pino que ativa ou não a ponte H.

Caso tenha um sinal de 5V inserido nele, a ponte é ligada, caso seja 0V a ponte esta desligada.

Como temos 2 pontes H, temos o Enable A (Ativa A) e o Enable B (Ativa B).

Normalmente o Enable A e B fica em curto com um sinal de 5V da placa através de um jumper.

# Exemplo de ligação da ponte H



## Agenda de hoje

Estrutura da Linguagem do Arduino (linguagem C)

As portas de E/S do Arduino e suas funções em C

Conceitos básicos de Física

- Tensão, corrente & resistência
- Componentes básicos
- Conhecendo os sensores e atuadores
  - Motores e Ponte H
  - Sensores de Linha

Montagem de um exemplo de carrinho segue linha ←

- A importância do robô seguidor de linha
- Como o carrinho de um robô móvel funciona?
- Motor DC com Driver Ponte H L298N
- Sensor segue linha TCRT5000

# Montagem de um exemplo de carrinho segue linha





# Tópicos

Introdução ao  
Robô seguidor  
de linha

A importância do robô seguidor de linha  
Como o carrinho de um robô móvel funciona?

Prática –  
Desenvolvendo  
o robô  
seguidor de  
linha

**Motor DC com Driver Ponte H L298N**  
**Sensor segue linha**  
**Colocando para funcionar**

# A importância do robô seguidor de linha



Muitas vezes é primeiro projeto de robótica móvel dos iniciantes.



Em todo o mundo, competições de robótica possuem modalidades únicas para o seguidor de linha.



Brasil a OBR (Olimpíada Brasileira de Robótica).  
<http://www.obr.org.br/>

# A importância do robô seguidor de linha



O projeto envolve conceitos de dimensionamento de energia, controle de motores, programação de embarcados, entre outros.

Esses mesmos conceitos são levados para projetos maiores e mais complexos, tanto acadêmicos quanto industriais.

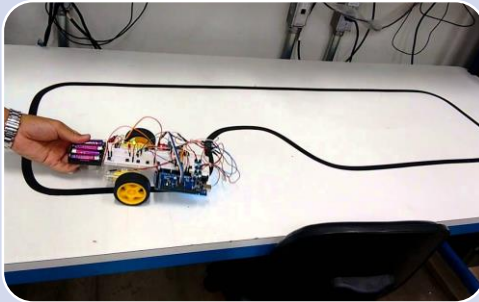
# A importância do robô seguidor de linha

As aplicações industriais são bem diversificadas:

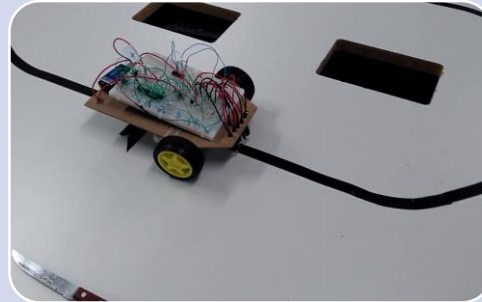
- Os robôs que gerenciam o estoque da Alibabá
- Os robôs manejavam as prateleiras de produtos dentro do estoque por meio de orientação de faixas no chão.
- Hoje o Alibabá evoluiu os robôs se comunicam, e se localizam, por meio de IA.



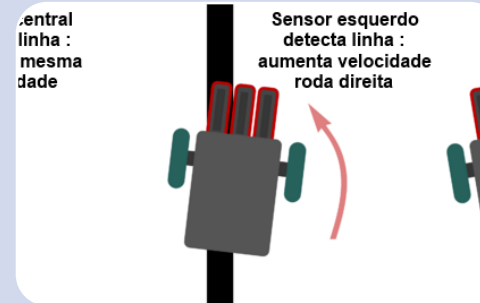
# Como o carrinho de um robô móvel funciona ?



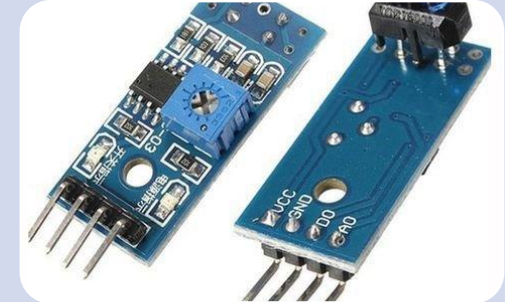
O robô seguidor de linha funciona de maneira bem simples.



Ele deve seguir andando por cima de uma linha de cor preta (fita isolante) ou branca.



Os circuitos podem alternar entre as cores do campo e da faixa, no nosso caso iremos assumir que a pista é branca e a faixa é preta.



Usaremos 2 sensores infravermelho que detectam a presença ou não desta faixa. De acordo com a combinação dos sensores, o carrinho irá para frente ou virar para um dos lados.

# Como o carrinho de um robô móvel funciona ?

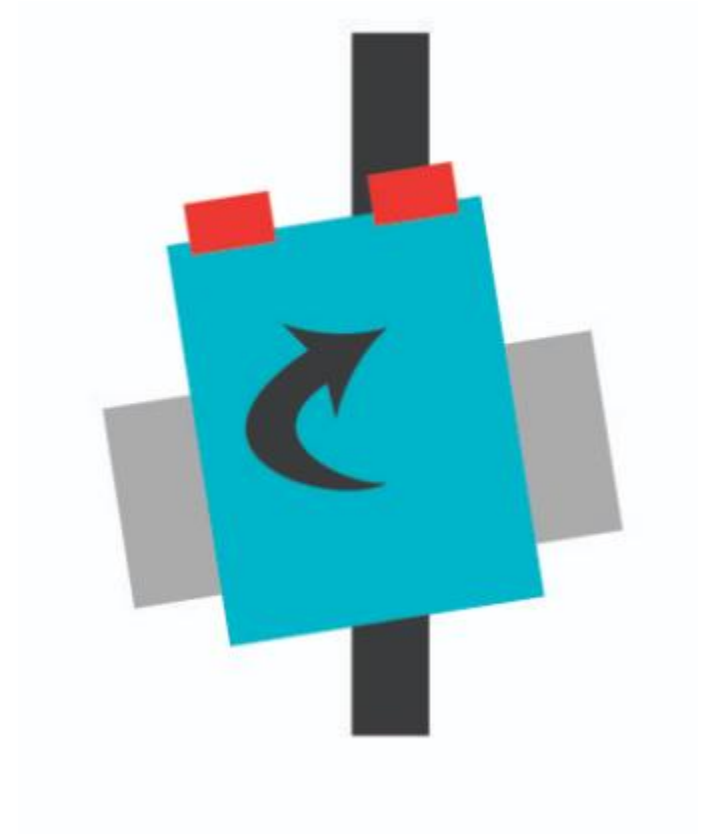
A faixa  
preta  
absorve a  
luz do led

Sensor da  
direita em  
nível baixo

O carrinho  
deve virar  
para direita

O motor da  
esquerda  
deve girar  
menos que  
o da direita

... ou o da  
direita deve  
girar mais  
rápido que  
o da  
esquerda...



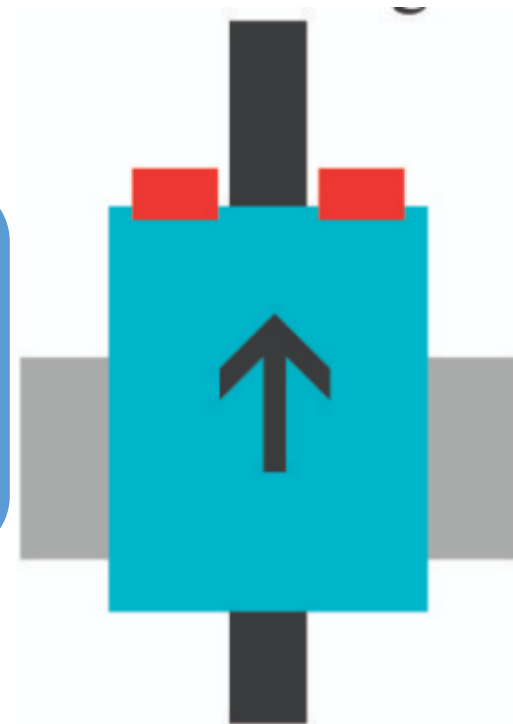
# Como o carrinho de um robô móvel funciona ?

A faixa  
branca  
reflete a  
luz do led

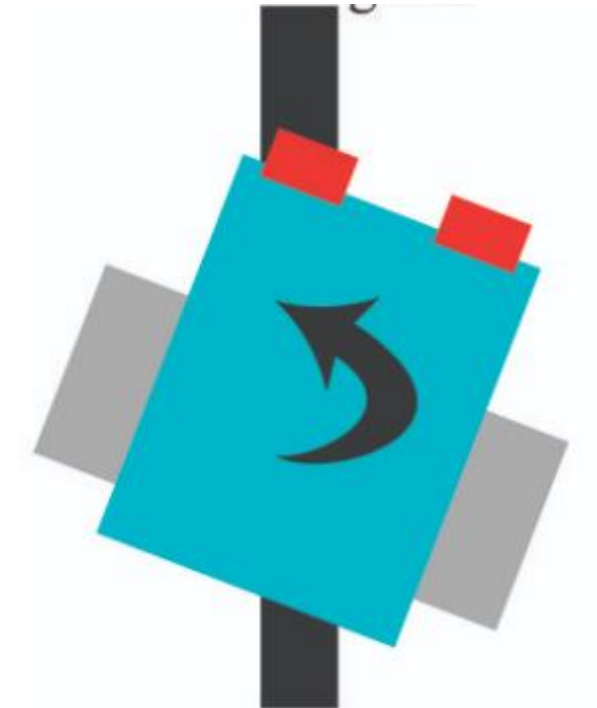
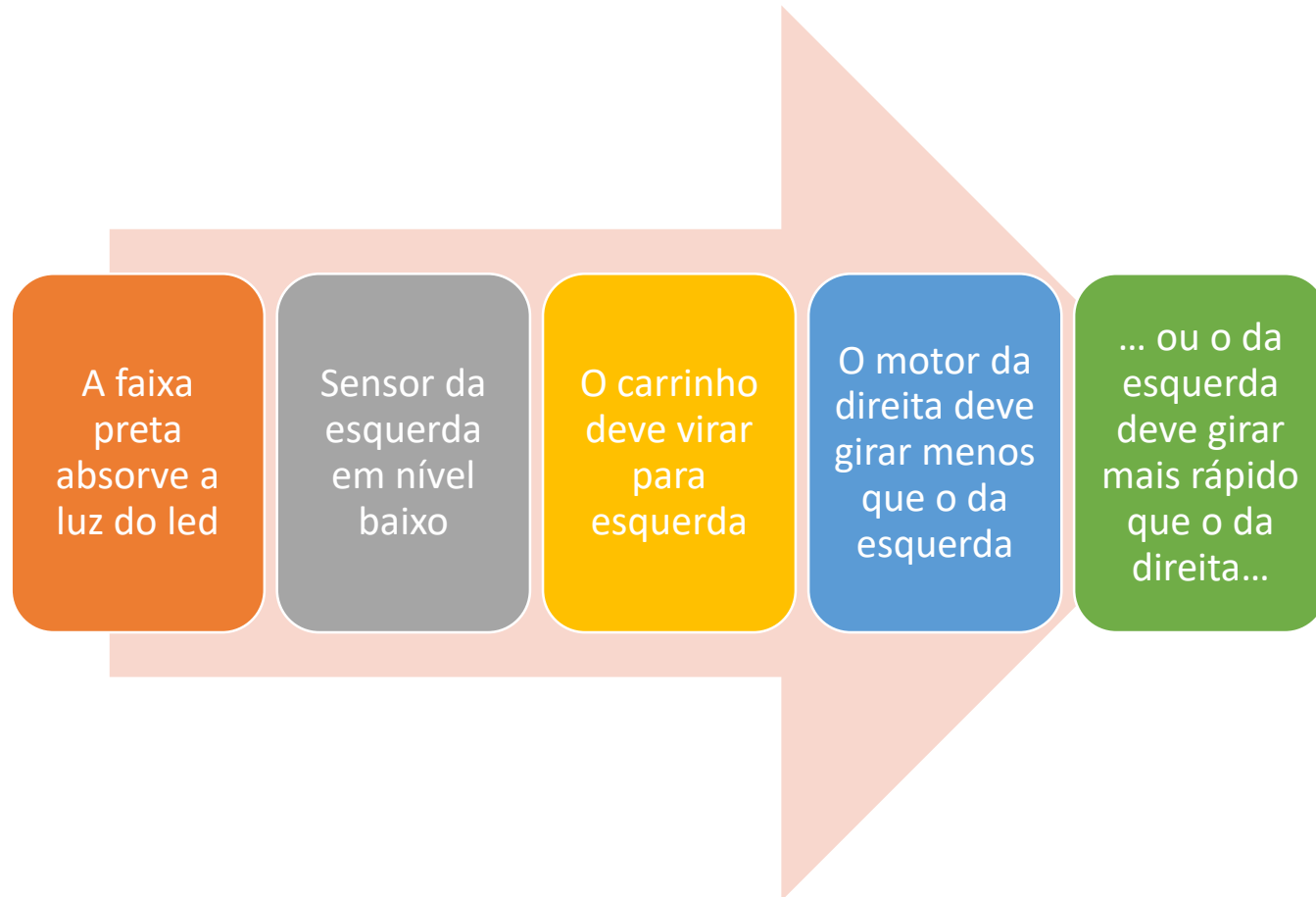
Sensores  
em em  
nível alto

O carrinho  
deve seguir  
em frente

Os motores  
devem ter  
velocidades  
iguais



# Como o carrinho de um robô móvel funciona ?





← → ↻ 🔒 https://www.tinkercad.com 🔍 ☆



TIN KER CAD AUTODESK® TINKERCAD™

Galeria Blog Aprenda Ensinar Entrar INSCREVER-SE

FLATAS BORING

From mind to design in minutes

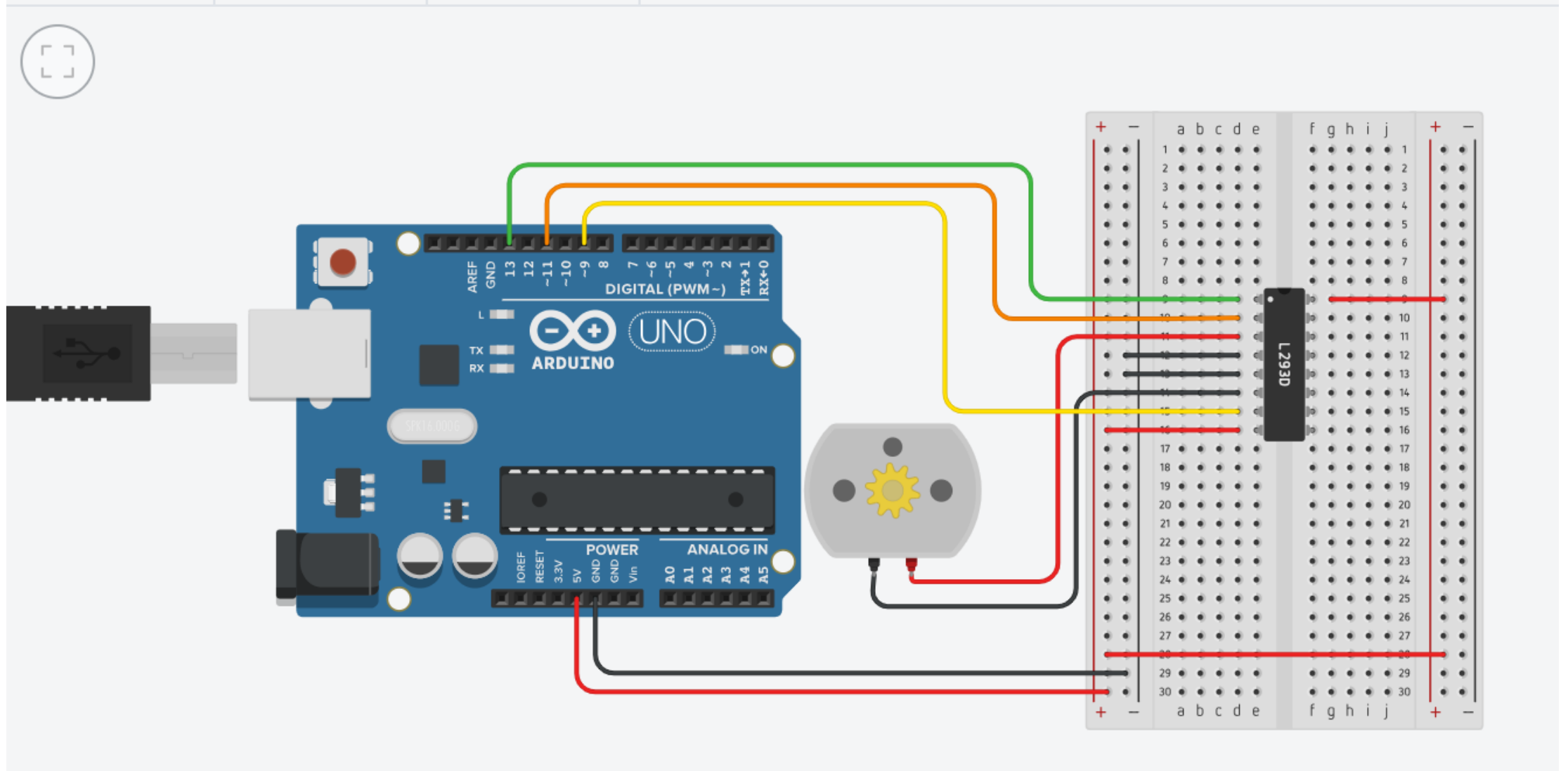
Vamos, primeiramente usar o simulador tinkercad para evitar causar algum dano na nossa ponte H...

# Componentes usados na experiencia

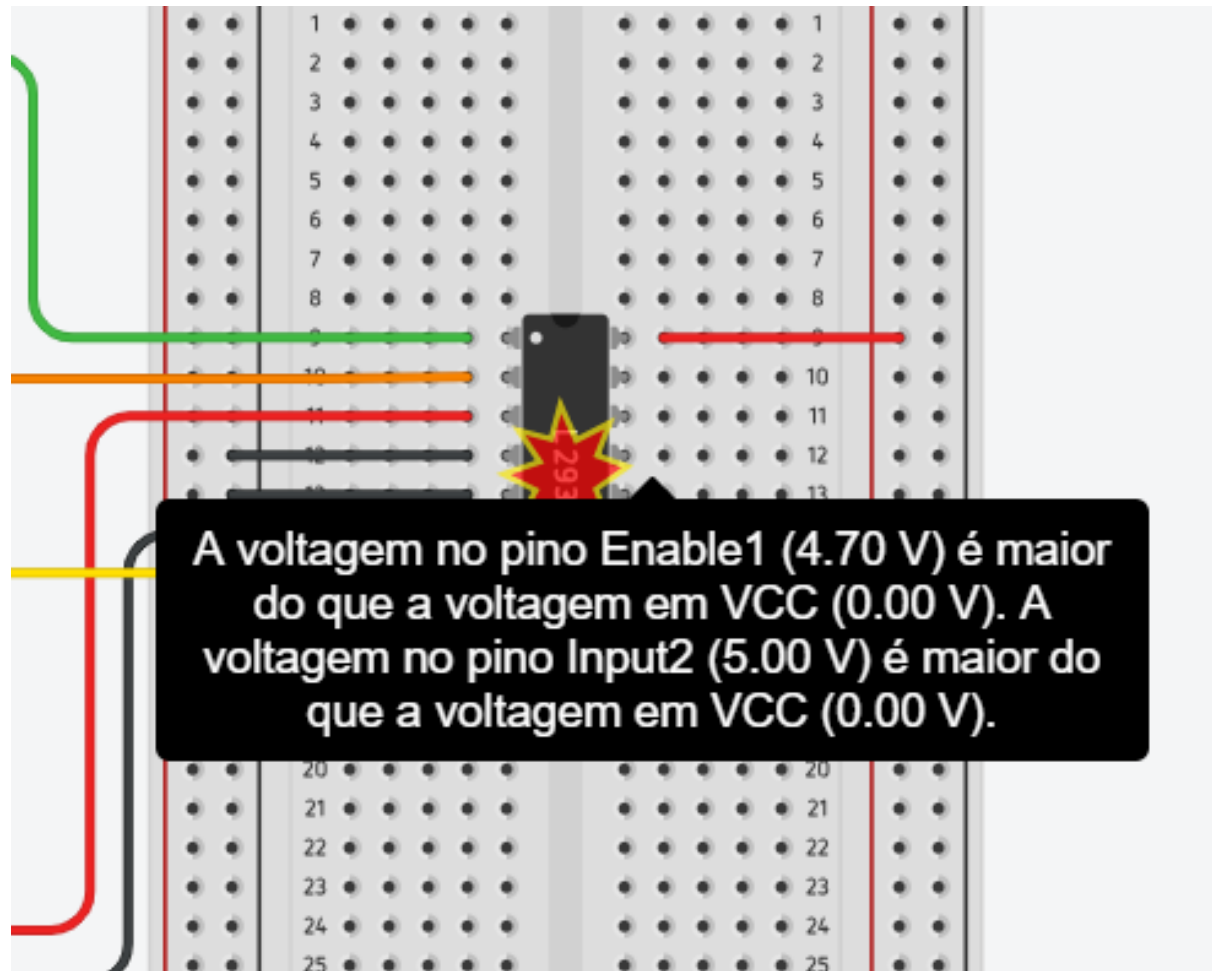
## Lista de componentes

Nome	Quantidade	Componente
U3	1	Arduino Uno R3
M1	1	Motor CC
UPonte H	1	Acionador de motor de ponte H

# No tinkercad



# Cuidado



# Código setup

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(9, OUTPUT);
  digitalWrite(13, HIGH);
  //coloque o pino 13 em HIGH para habilitar o driver do motor
}
```

# Código loop

```
void loop()
{
    digitalWrite(11, HIGH);
    digitalWrite(9, LOW);
    // Use o seguinte bloco de código para executar o motor CC no sentido anti-horário
    // Use somente se for necessário
    delay(2000);
    digitalWrite(11, LOW);
    digitalWrite(9, HIGH);
    delay(2000);
}
```

# Dicas

# Fita dupla face





# Referencias

Motor DC com Driver Ponte H L298N - FelipeFlop

- <https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>

Robô seguidor de linha com sensor infravermelho e PWM - Vida de silício)

- <https://portal.vidadesilicio.com.br/robo-seguidor-de-linha-sensor-infravermelho-e-pwm/>

**Fim da Aula II**