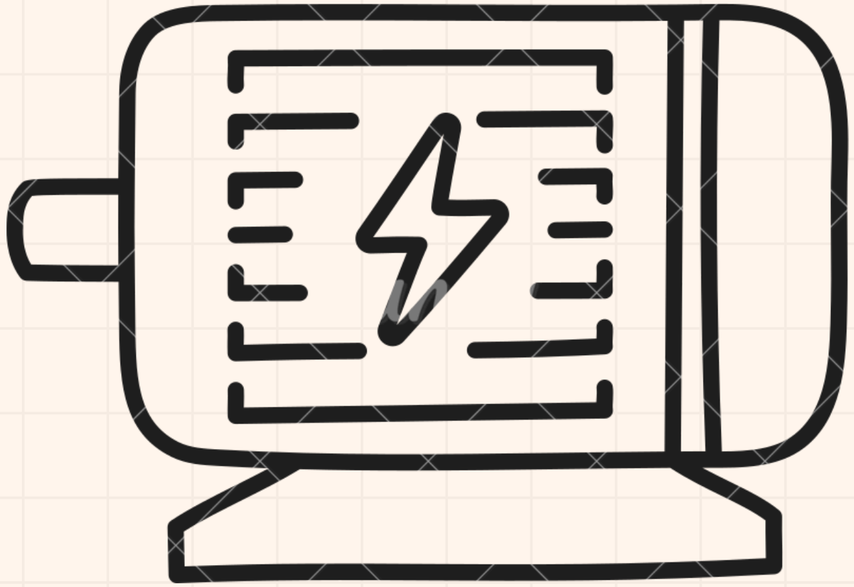
PROGRAMANDO MOTORES EM C/C++ USANDO O ESP8266



QUESTIONÁRIO DA OFICINA:

HTTPS://DOCS.GOOGLE.COM/FORMS/D/E/1FAIPQLSDI9L82FFDA_GVNLYB5EGLXWDG0Z0G52TK3Y88CIE-UISFSOQ/VIEWFORM

INTRODUÇÃO

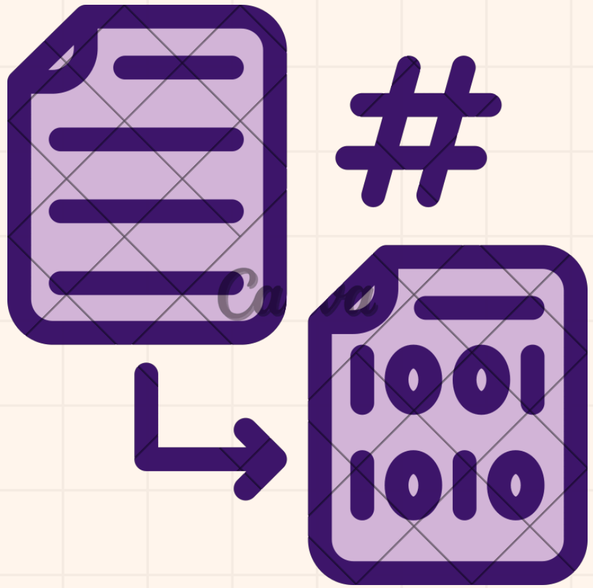
OBJETIVOS:

O objetivo deste minicurso é explicar e exemplificar o básico da programação em C/C++ para utilizar microcontroladores para as diversas funções para as quais serve focando em como controlar os principais motores.

LINK DO GITHUB DO CURSO:

HTTPS://GITHUB.COM/ERICKSEBBA/?????

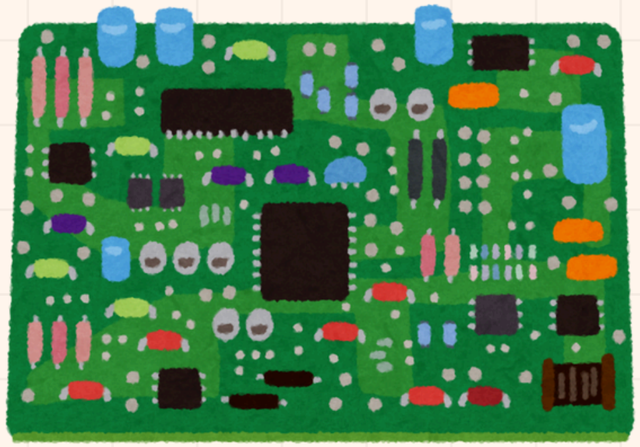
O QUE É UM MICROCONTROLADOR



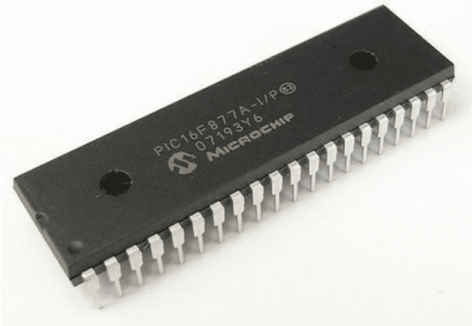
Um microcontrolador é um chip que contém um computador em miniatura, com um processador, memória, periféricos de entrada e saída, entre outros componentes. Ele é responsável por controlar circuitos eletrônicos por meio de algoritmos programáveis.

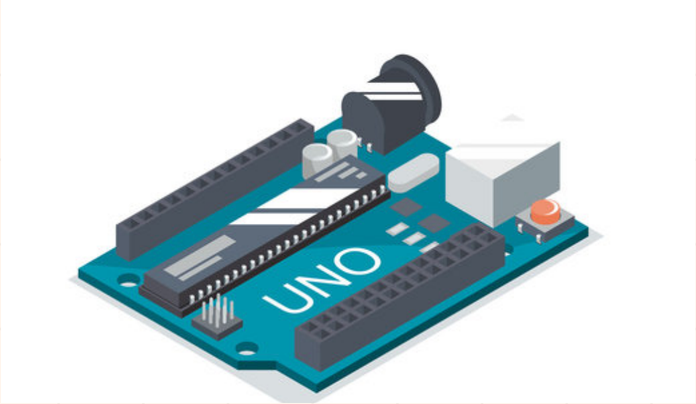
Os microcontroladores são usados em uma grande variedade de produtos e dispositivos, como:

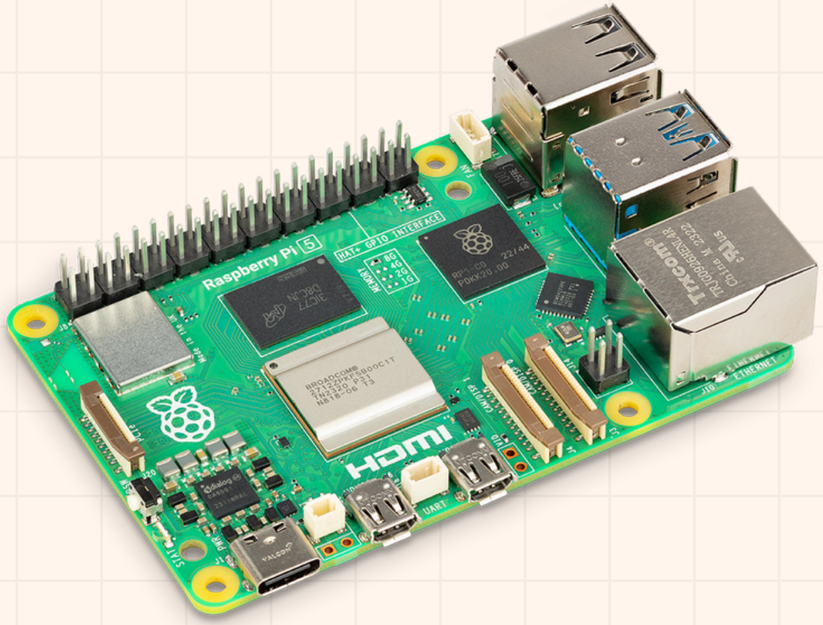
- Controles remotos
- Eletrodomésticos
- Controles de carros
- Brinquedos
- Dispositivos médicos implantáveis
- Máquinas de escritório
- Ferramentas elétricas



O QUE É UM MICROCONTROLADOR



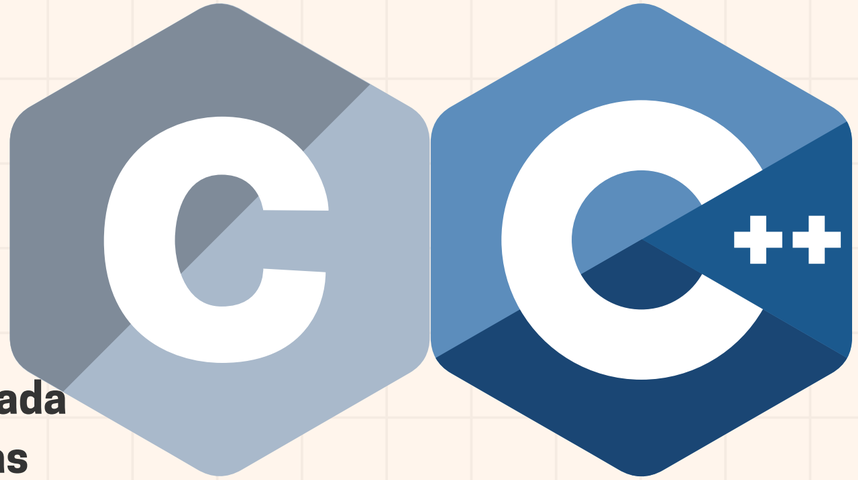




POR QUE PROGAMAR EM C NO MICROCONTROLADOR

Neste Minicurso iremos apresentar as vantagens e como programar microcontroladores em geral utilizando a linguagem C.
Mas primeiro, qual é a vantagem de se programar em C puro, ante as outras linguagens: A resposta é simples, o compilador da linguagem C é tão rápida quanto o Assembly do chip!

POR QUE PROGAMAR EM C NO MICROCONTROLADOR



- Propósito Geral: C é uma linguagem de programação de propósito geral usada para uma ampla gama de desenvolvimentos de software, desde sistemas operacionais até sistemas embarcados.

- Sintaxe: C possui uma sintaxe mais complexa e exige o manuseio explícito de operações de baixo nível, como gerenciamento de memória e interação com hardware.

- Compilador: O código em C é compilado usando um compilador C, que converte o código em código de máquina que um computador ou microcontrolador pode executar.

- Bibliotecas Padrão: C possui um conjunto rico de bibliotecas padrão, mas frequentemente é necessário incluir bibliotecas específicas para interação com hardware

ESTRUTURA

INICIO

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    printf("Hello World\n");
    system("pause");
    return 0;
}
```

FIM

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    int a = 10;
    a=2;
    printf("O valor de a é: %d\n", a);
    system("pause");
    return 0;
}
```

LINGUAGEM ESTRUTURADA

A linguagem C é estruturada, sua principal característica é a sequência de execução do programa

Exemplo de código

```c
#include <stdio.h>

int main()
{
    int nota1, nota2, nota3;
    nota1 = 10;
    nota2 = 8;
    nota3 = 7;

    float media = (nota1 + nota2 + nota3)/3;
    printf("A média das notas é: %.2f", media);
}
```

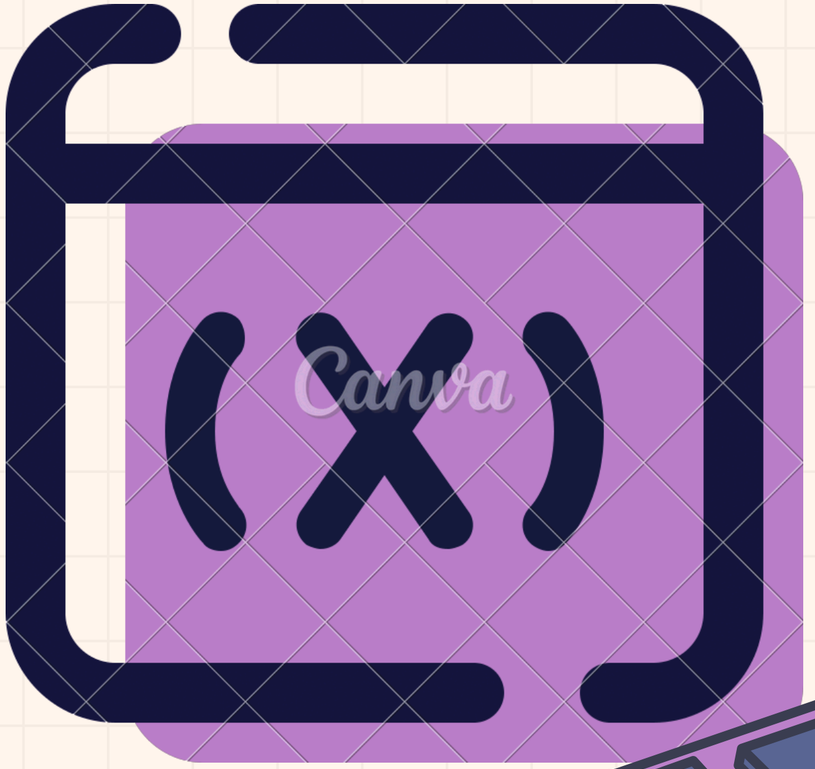
IDENTAÇÃO

Identação é o nome que se dá à técnica de separar graficamente cada bloco de instruções de acordo com seu grau de pertinência.

Os compiladores da linguagem C não necessitam de identação para funcionar corretamente, entretanto ela melhora muito o entendimento do Código.

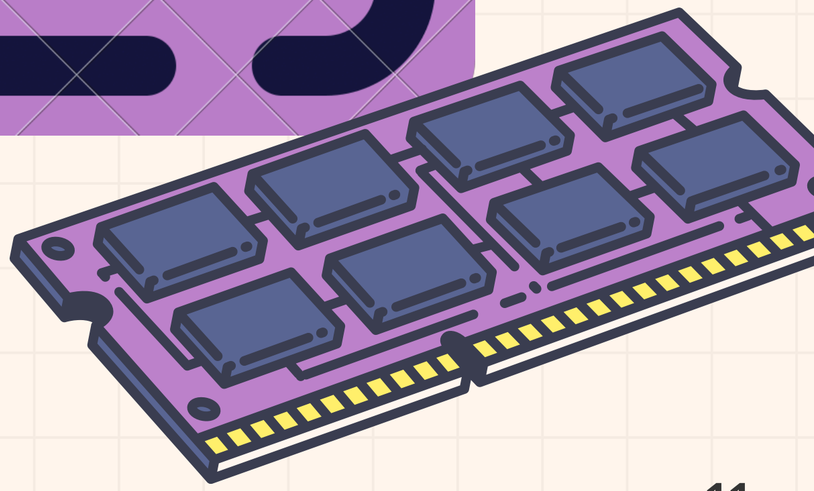
```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    for(int i = 0; i < 10; i++){
        if(i % 2 == 0){
            printf("i eh par e eh: %d\n", i);
        }
        else{
            printf("i eh impar e eh: %d\n", i);
        }
    }
}
```

VARIÁVEIS
E TIPOS DE DADOS



variável é um espaço na memória RAM, alocado para armazenar um determinado valor para uso futuro. Esse valor será modificado e utilizado para efetuar operações, comparações e outras instruções

Uma variável sempre deve ser declarada, ou seja, todas as variáveis utilizadas em uma função precisam ser relacionadas no início desta função. A declaração de variáveis é feita conforme exemplo a seguir:



Tab. 4.2. Tipos de variáveis.

Declaração	Bytes	Variação
`char`	1	-128 a 127
`int`*	2	-32.768 a 32.767
`float`	4	$1,2 \times 10^{-38}$ a $3,4 \times 10^{+38}$
`double`	8	$2,2 \times 10^{-308}$ a $1,8 \times 10^{+308}$

* Depende da arquitetura do processador, sistema operacional e compilador utilizado.

Tab. 4.3 Modificadores de tipo.

Declaração	Bytes	Variação
`signed char`	1	-128 a 127
`unsigned char`	1	0 a 255
`short`	2	-32.768 a 32.767
`short int`	2	-32.768 a 32.767
`signed int`	2	-32.768 a 32.767
`signed short`	2	-32.768 a 32.767
`signed short int`	2	-32.768 a 32.767
`unsigned int`	2	0 a 65.535
`unsigned short`	2	0 a 65.535
`unsigned short int`	2	0 a 65.535
`long`	4	-2.147.438.648 a 2.147.438.647
`long int`	4	-2.147.438.648 a 2.147.438.647
`signed long`	4	-2.147.438.648 a 2.147.438.647
`signed long int`	4	-2.147.438.648 a 2.147.438.647
`unsigned long`	4	0 a 4.294.967.295
`unsigned long int`	4	0 a 4.294.967.295
`long double`*	8	$2,2 \times 10^{-308}$ a $1,8 \times 10^{+308}$

* Dependendo do compilador pode ser atribuído 8, 10 ou 12 bytes.

NOME DE VARIÁVEIS
E PALAVRAS RESERVADAS

A linguagem C17 possui 32 comandos e palavras reservadas, ou seja, palavras que executam funções específicas e que não podem ser utilizadas como nomes de funções ou variáveis, conforme tabela abaixo:

Tab. 4.1. Palavras reservadas da linguagem C.

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

MODIFICADORES
DE TIPOS

Os quatro tipos de dados podem receber os modificadores opcionais de modo a criar um tipo de dado mais adequado para cada aplicação. A instrução short informa ao compilador que o tamanho da variável necessária é menor que o padrão do tipo de dado; de forma análoga, a instrução long informa que o tamanho necessário é maior que o tamanho padrão do tipo de dado, e as instruções unsigned e signed informam que os valores armazenados serão apenas positivos ou positivos e negativos, respectivamente

Tab. 4.3 Modificadores de tipo.

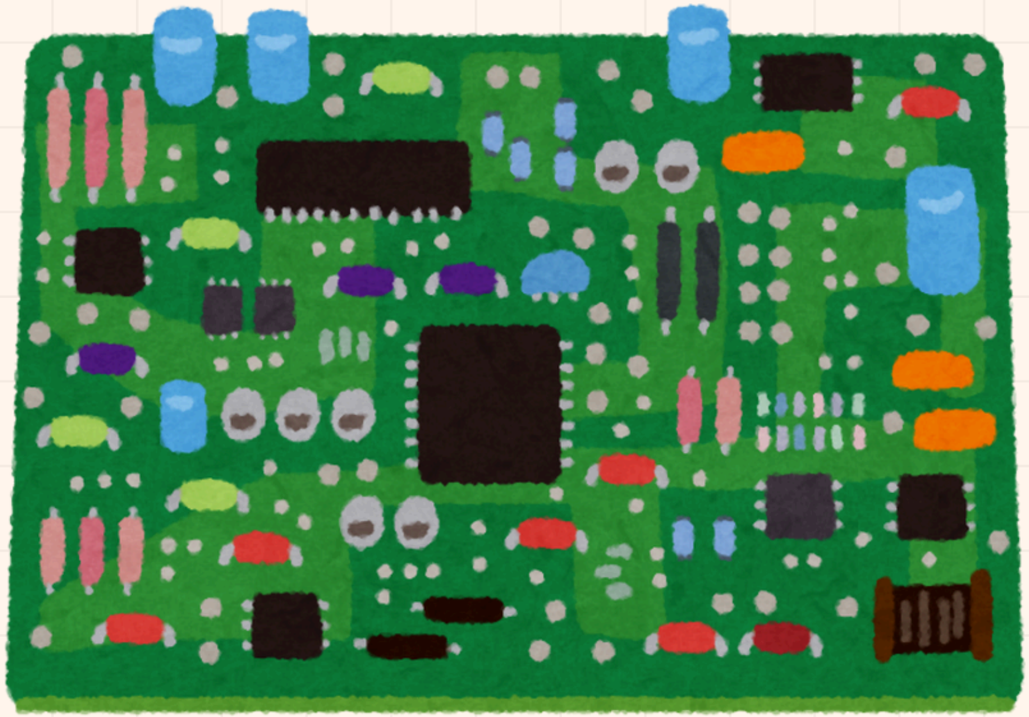
Declaração	Bytes	Variação
signed char	1	-128 a 127
unsigned char	1	0 a 255
short	2	-32.768 a 32.767
short int	2	-32.768 a 32.767
signed int	2	-32.768 a 32.767
signed short	2	-32.768 a 32.767
signed short int	2	-32.768 a 32.767
unsigned int	2	0 a 65.535
unsigned short	2	0 a 65.535
unsigned short int	2	0 a 65.535
long	4	-2.147.438.648 a 2.147.438.647
long int	4	-2.147.438.648 a 2.147.438.647
signed long	4	-2.147.438.648 a 2.147.438.647
signed long int	4	-2.147.438.648 a 2.147.438.647
unsigned long	4	0 a 4.294.967.295
unsigned long int	4	0 a 4.294.967.295
long double*	8	$2{,}2 \times 10^{-308}$ a $1{,}8 \times 10^{+308}$

* Dependendo do compilador pode ser atribuído 8, 10 ou 12 bytes.

DATASHEETS
& CIRCUITOS

Quando se trata de sistemas embarcados e microcontroladores é muito importante ter a experiência com datasheets e os circuitos mais simples.

Nos exemplos de datasheets do PIC16f630 e do DHT11 podemos entender aspectos fundamentais de seu funcionamento. E no que se trata de programação, podemos ver, de forma antecipada, o que o Micro escolhido tem a nos oferecer, além de ver como um sensor qualquer faz a comunicação.

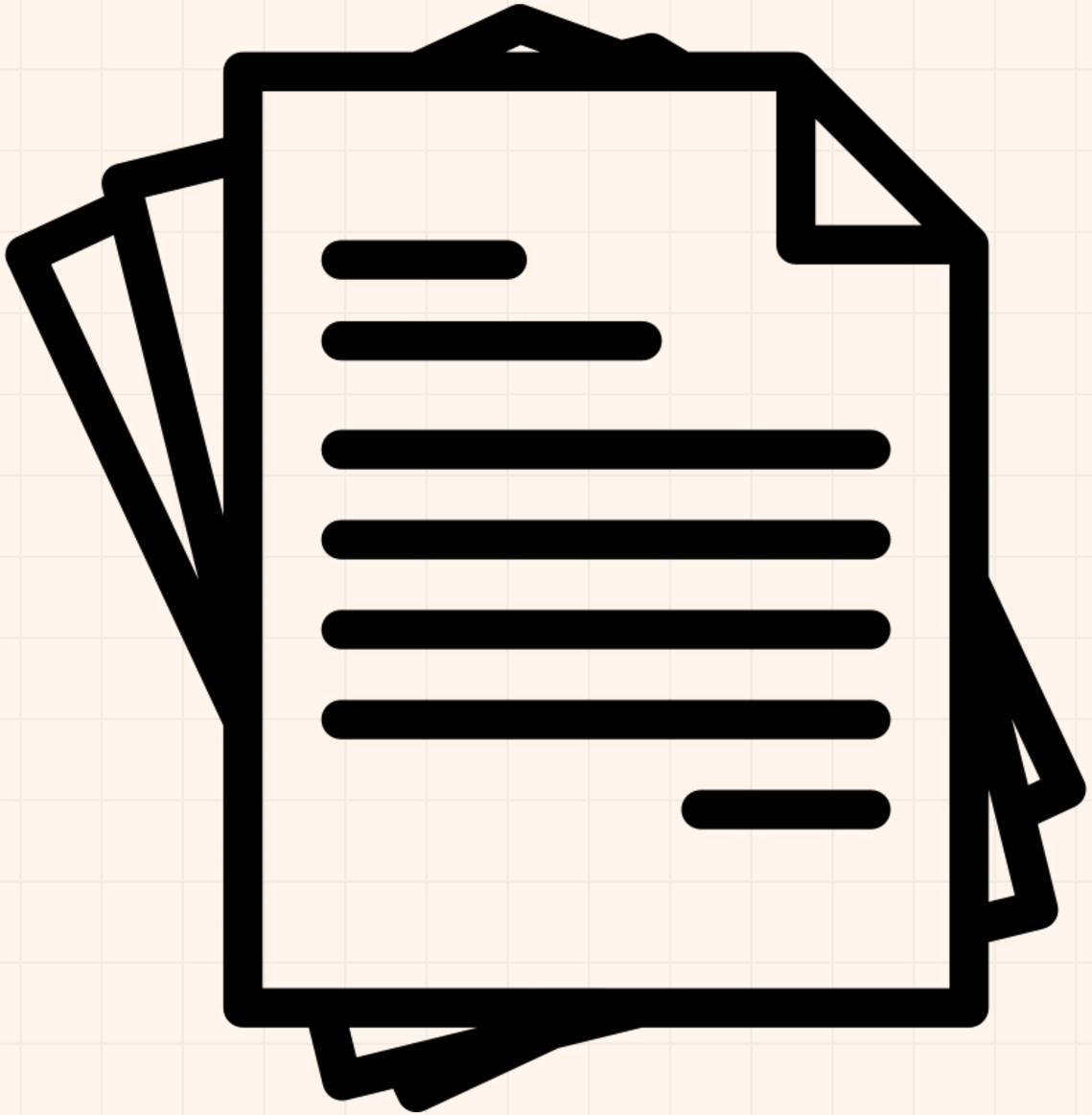


DATASHEETS
& CIRCUITOS

O que é um Datasheet?

Um datasheet, folha de dados ou folha de especificações é um documento que resume o desempenho e outras características técnicas de um produto, máquina, componente (por exemplo, um componente eletrônico), material, subsistema (por exemplo, fonte de alimentação) ou software em detalhe suficiente para que possa ser usado por um engenheiro de projeto para integrar o componente em um sistema.

wikipédia

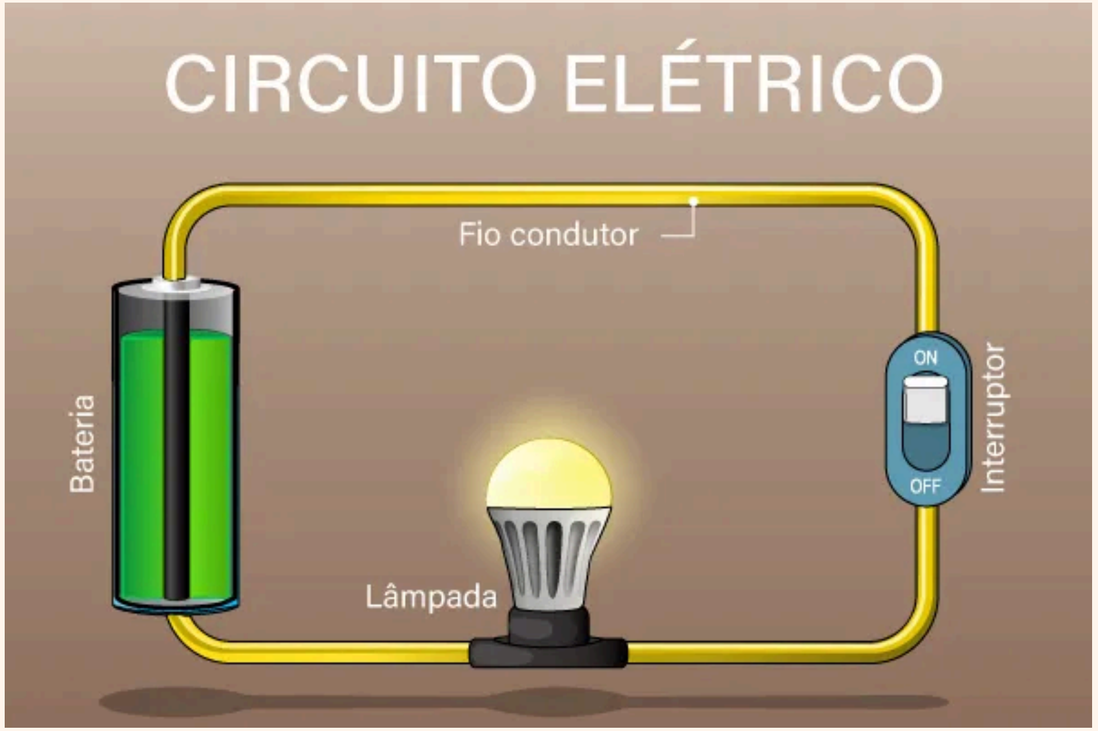


É um documento que reúne informações sobre as características, especificações, funcionamento e aplicação de um produto,

SISTEMA ELÉTRICO

- “Um sistema elétrico é aquele que utiliza a energia elétrica principalmente para realizar trabalho físico, sem processamento ativo de sinais”
  - Baseia-se nas leis da corrente elétrica e da potência
  - Não há manipulação lógica do sinal
  - Uso de componentes passivos

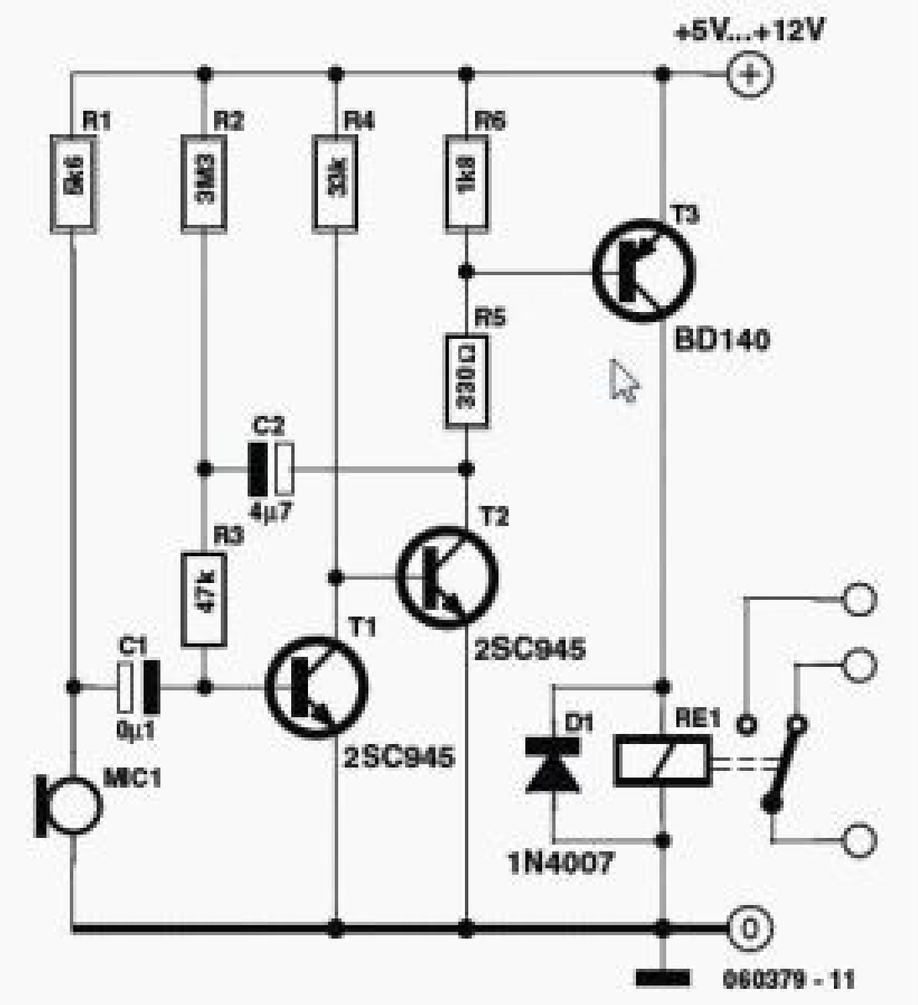
- Componentes típicos:
  - Condutores
  - Resistores
  - Lâmpadas
  - Motores AC/DC



SISTEMA ELETRÔNICO

- “Um sistema eletrônico é aquele que utiliza a eletricidade para manipular, processar, transmitir ou armazenar informações, por meio de sinais elétricos”
  - Baseado no controle do fluxo de elétrons em semicondutores
  - Pode usar sinais analógicos ou digitais
  - Processamento de sinais

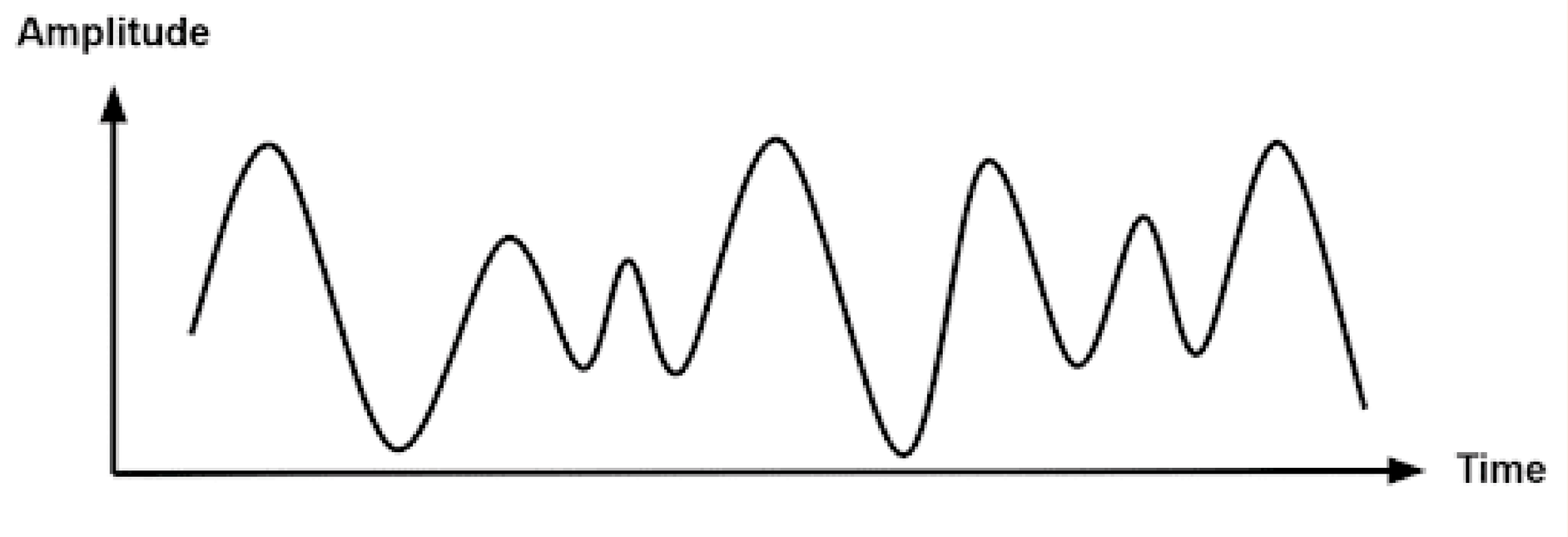
- Componentes típicos:
  - Transistores
  - Capacitores
  - Diodos
  - Amplificadores
  - Microcontroladores



SINAL ANALÓGICO

- “Um sinal analógico é uma representação elétrica contínua de uma grandeza física variável no tempo, onde os valores assumidos podem variar infinitamente dentro de um intervalo definido”

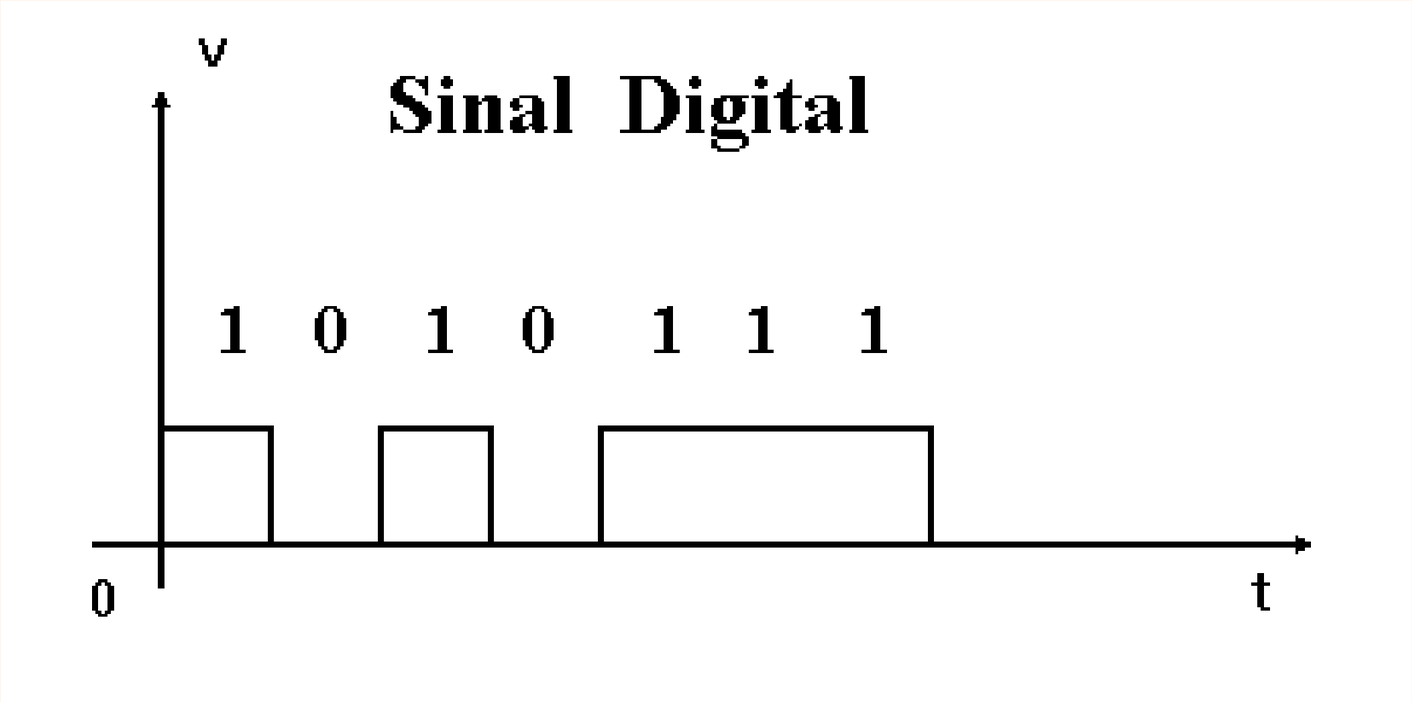
- Imitam o comportamento natural das grandezas físicas contínuas



SINAL DIGITAL

- “Um sinal digital é uma forma de sinal elétrico que assume apenas valores discretos, geralmente dois níveis bem definidos (LOW ou HIGH), representando estados lógicos ou comandos binários”

- O valor da tensão não precisa ser exato, apenas estar dentro da faixa de reconhecimento definida pela eletrônica digital



TENSÃO

- “Representa o quanto de energia elétrica está disponível para mover cargas elétricas (elétrons) de um ponto a outro”
- Medida em Volts (V)
- Diferença de potencial entre dois pontos distintos (DDP)
- Potencial Elétrico: 1 Volt = 1 Joule/1 Coulomb
- O potencial elétrico é dado pela equação:

$$U_{AB} = V_A - V_B$$

Alessandro Volta (1745-1827)

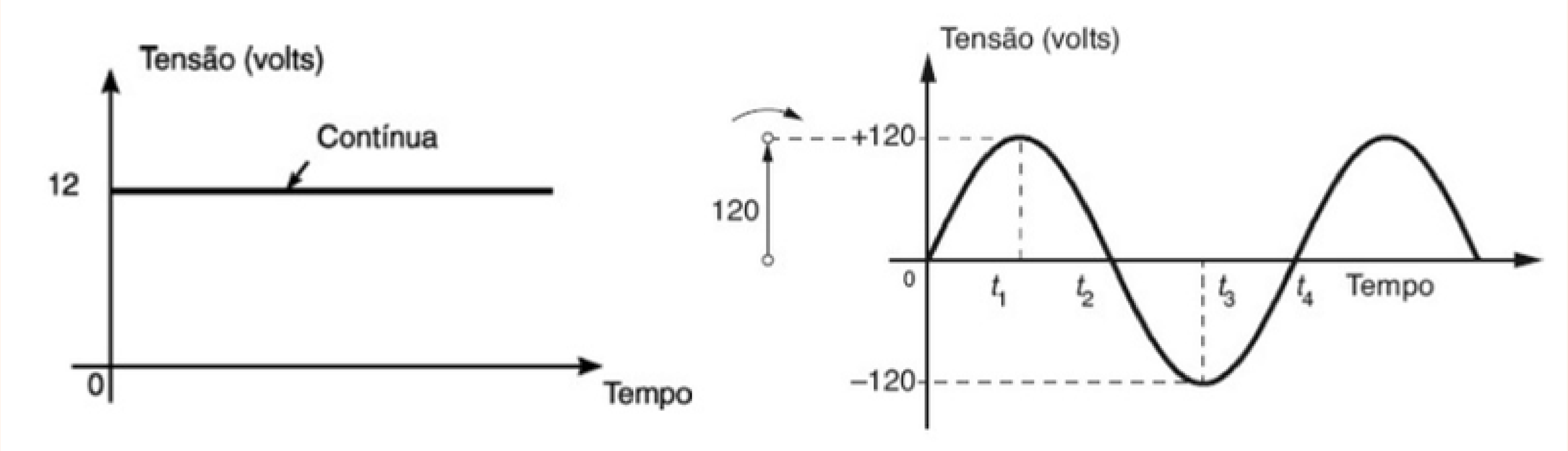
CORRENTE

- “Movimento ordenado dos elétrons em um condutor”
- Medida em Ampères (A)
- Unidade base usada para medir a intensidade da corrente elétrica
- Comportamento da Corrente Elétrica
  - Sentido Real (Verdadeiro)
  - Sentido Convencional
- “A corrente elétrica é composta por elétrons (cargas negativas). Seguindo o sentido real da corrente, ela sempre fluirá do polo de menor potencial elétrico para o polo de maior potencial elétrico”

André-Marie Ampère (1775-1836)

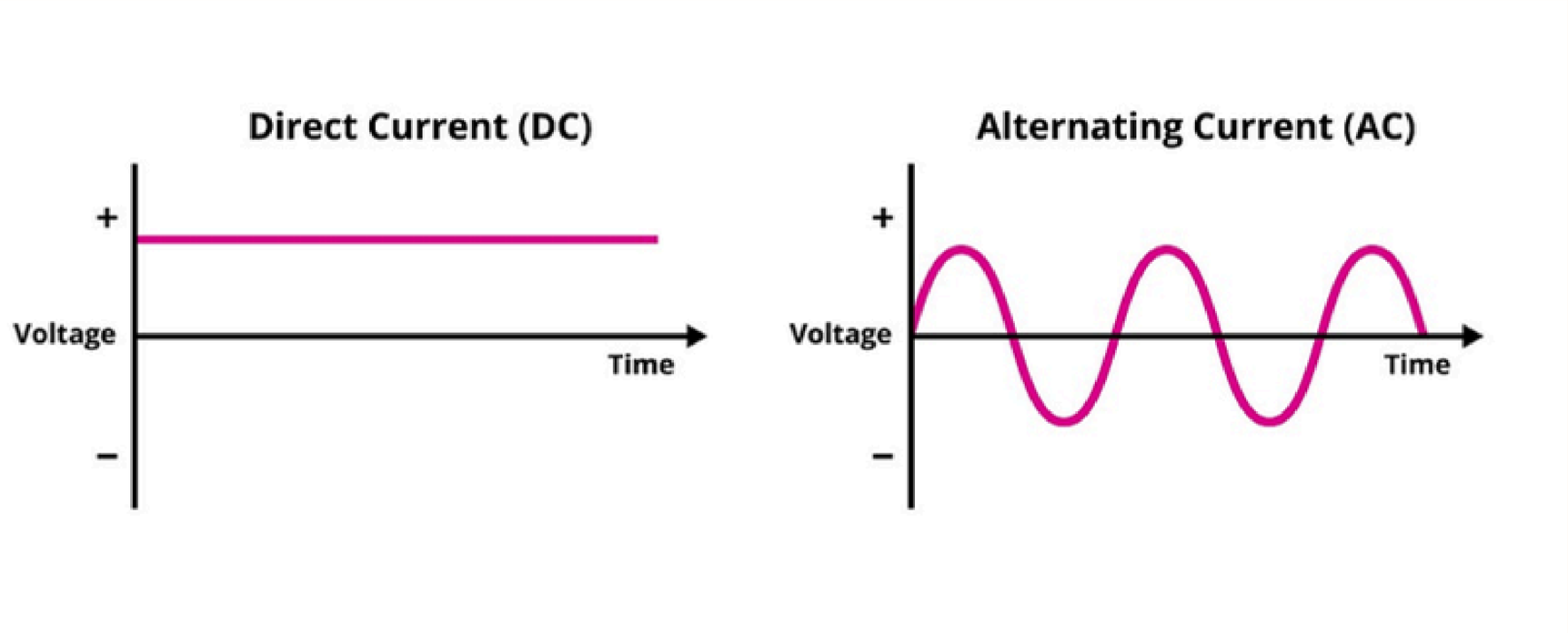
TENSÃO (CONTÍNUAxALTERNADA)

- **Tensão Contínua (DC)**
  - **A tensão não muda de polaridade com o tempo**
  - **O valor da tensão é constante**
- **Tensão Alternada (AC)**
  - **A tensão inverte a polaridade periodicamente**
  - **O valor da tensão alterna entre positiva e negativa**



CORRENTE (CONTÍNUAxALTERNADA)

- Corrente Contínua (DC)
  - Os elétrons fluem sempre na mesma direção (do polo negativo para o positivo)
- Corrente Alternada (AC)
  - O fluxo de elétrons muda de direção junto com a tensão (60 vezes por segundo no Brasil = 60 Hz)



RESISTÊNCIA

- "A resistência elétrica é a oposição que um material oferece à passagem da corrente elétrica"

- Medida em Ohms (Ω)

- Primeira Lei de Ohm: $\mathbf{R = \frac{U}{i}}$

- Fatores que definem a resistência de um condutor:
  - Material
  - Comprimento
  - Área da seção transversal
  - Resistividade
  - Temperatura

- Segunda Lei de Ohm: $\mathbf{R = \rho \frac{L}{A}}$

Georg Simon Ohm (1787-1854)

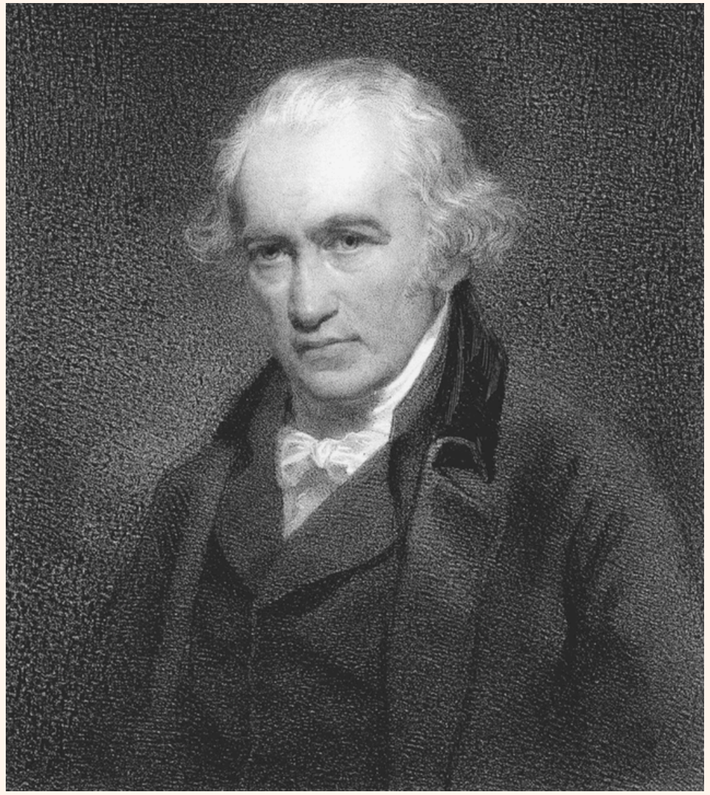
POTÊNCIA ELÉTRICA

- "Quantidade de energia elétrica transformada ou consumida por um dispositivo em um determinado intervalo de tempo"
- Medida em Watts (W)
- Potência: 1 Watt = 1 Joule por segundo (1 J/s)
- A Potência Elétrica é dada pela fórmula:

$$P = V \cdot I$$

- Variações da fórmula pela Lei de Ohm:

$$P = I^2 \cdot R \qquad P = \frac{V^2}{R}$$



James Watt (1736-1819)

RESISTORES

- Resistores Fixos
  - Possui resistência elétrica fixa, ou seja, constante, não muda com o tempo
  - Limitador de tensão

Resistor fixo de quatro faixas

- Resistores Variáveis
  - Possui resistência variável, permitindo que o valor da resistência seja alterado de forma manual ou automática

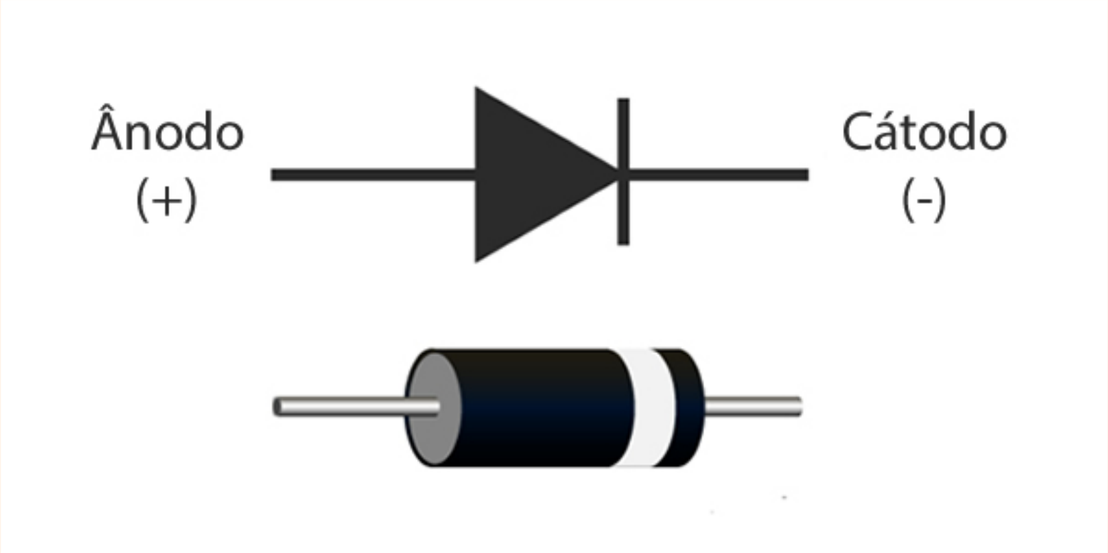
Resistor variável: potenciômetro

DIODOS

- “Componente eletrônico fundamental que permitem o fluxo de corrente elétrica em apenas uma direção”

- Material semicondutor
  - Silício
  - Germânio

- Possuem dois terminais:
  - Ânodo (Positivo)
  - Cátodo (Negativo)

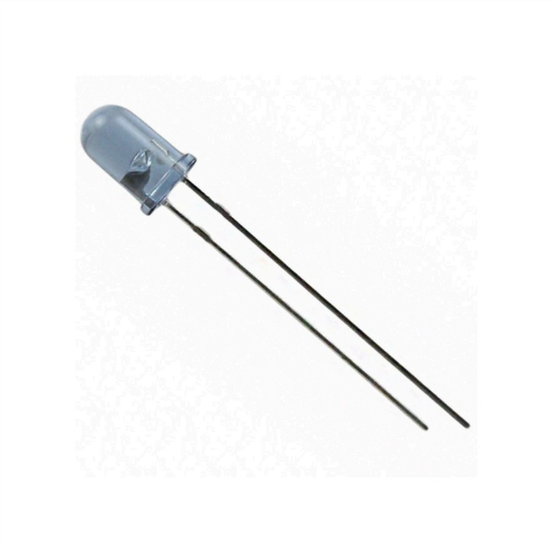


- Característica elétricas:
  - Tensão Direta de Polarização
  - Corrente Direta Máxima
  - Resistência Dinâmica
  - Tensão Reversa Máxima
  - Corrente Reversa de Fuga
  - Tempo de Recuperação

DIODOS

- Diodos possuem diversos tipos, cada qual com o objetivo de atender as necessidades desejas em circuito desenvolvido.

Diodo Retificador



Diodo Emissor de Luz (LED)

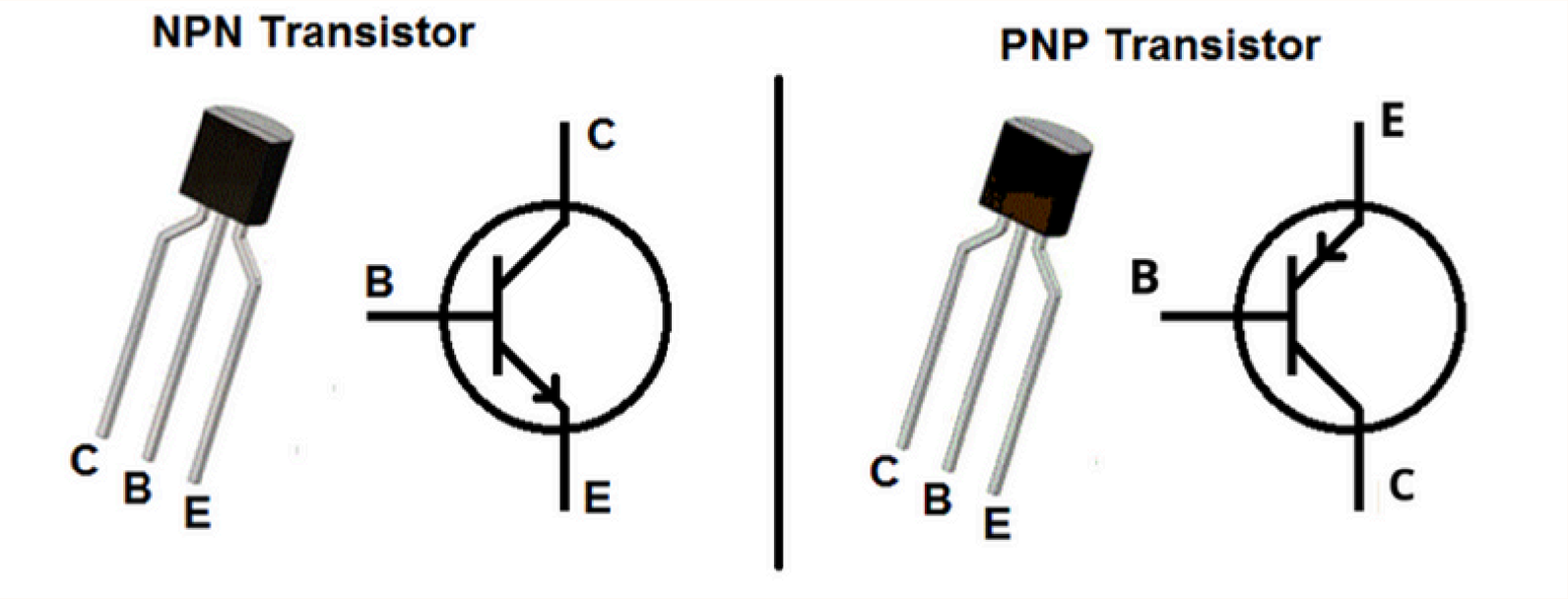
Diodo Zener

TRANSISTORES

- “Semicondutor fundamental usado para amplificar ou controlar o fluxo de corrente elétrica em circuitos eletrônicos”

- Dois tipos comuns:
  - Transistores bipolares
  - Transistores de Efeito de Campo (FET’s)

- Duas principais funcionalidades:
  - Interruptor (chave)
  - Amplificador de sinal



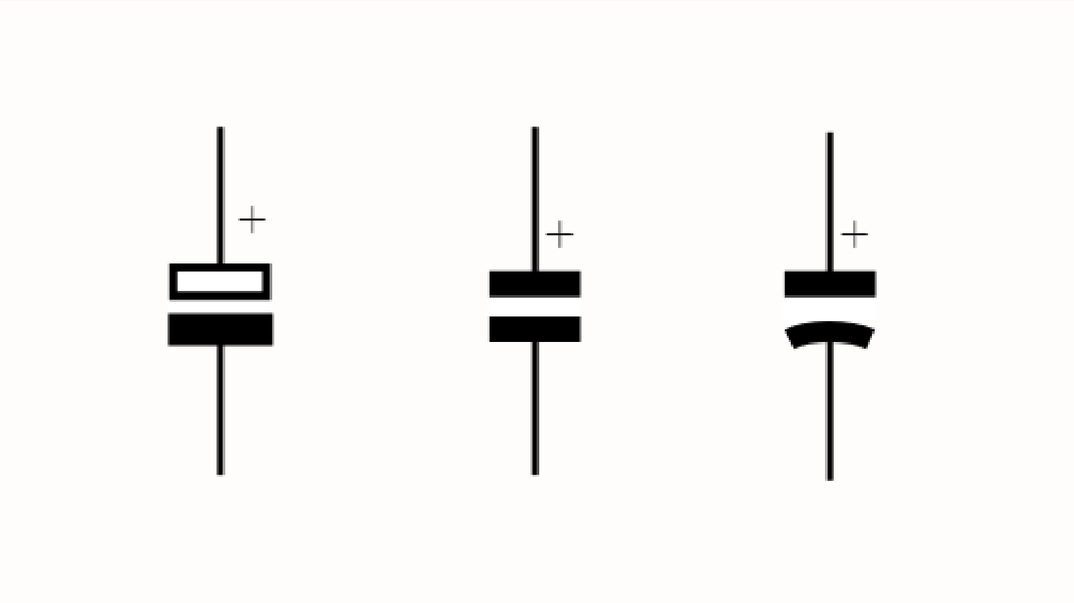
CAPACITORES

- “Componente eletrônico passivo que armazena e liberam carga elétrica de forma temporária em um circuito”

- Amplamente utilizados para:
  - Filtrar de sinais
  - Armazenar energia
  - Acoplar sinais em circuitos

- Princípio da capacitância
  - Quantidade de carga elétrica que o capacitor pode armazenar
  - Unidade de medida: Farad (F)

- Tensão nominal



CAPACITORES

- Capacitores possuem diferentes tipos, cada qual com suas características próprias de armazenar energia e aplicações

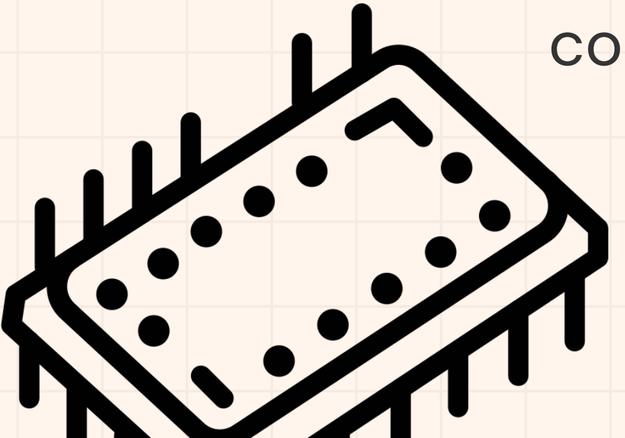
Capacitor Eletrolítico

Capacitor Cerâmico

Capacitor de Poliéster

ESP 8266

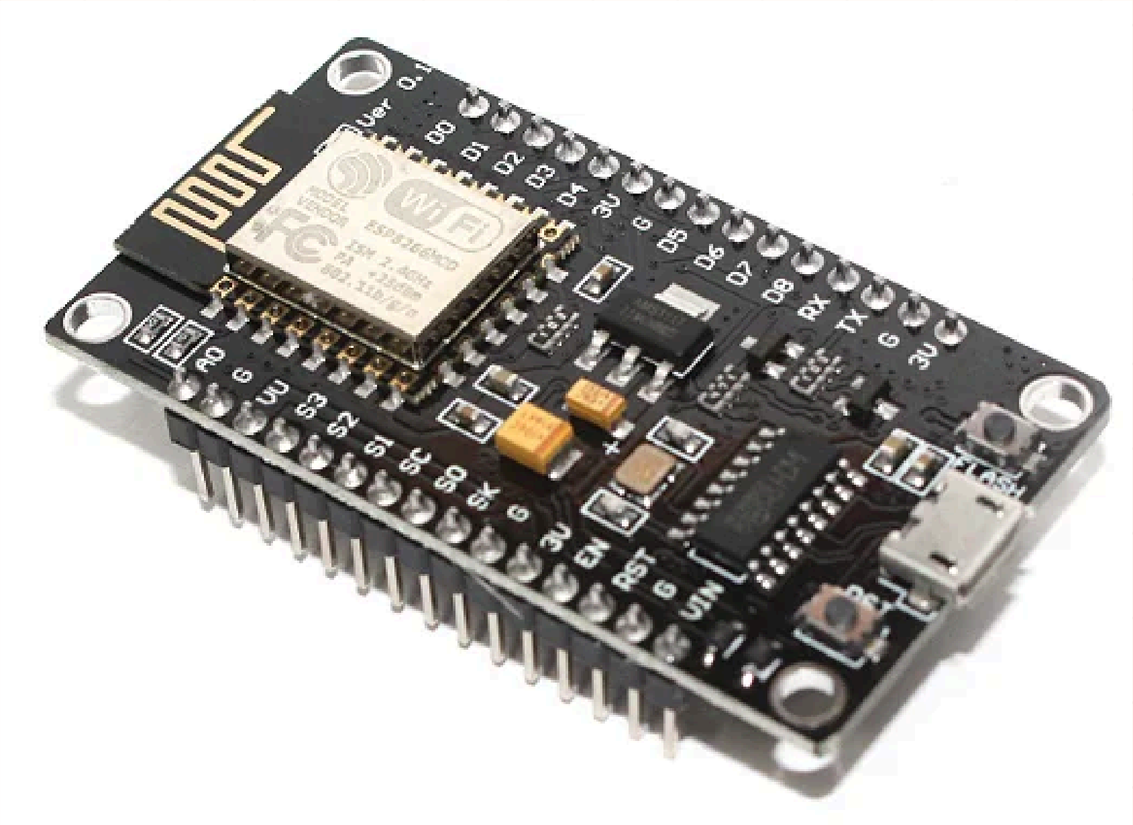
O ESP8266 é um chip microcontrolador desenvolvido pela Expressif para facilitar a conectividade de placas e a criação de soluções dentro do ecossistema da Internet das Coisas. s. O chip possui uma CPU de 32 Bits rodando a 80 MHz e ele traz incorporado os principais recursos necessários para comunicação Wi-Fi.



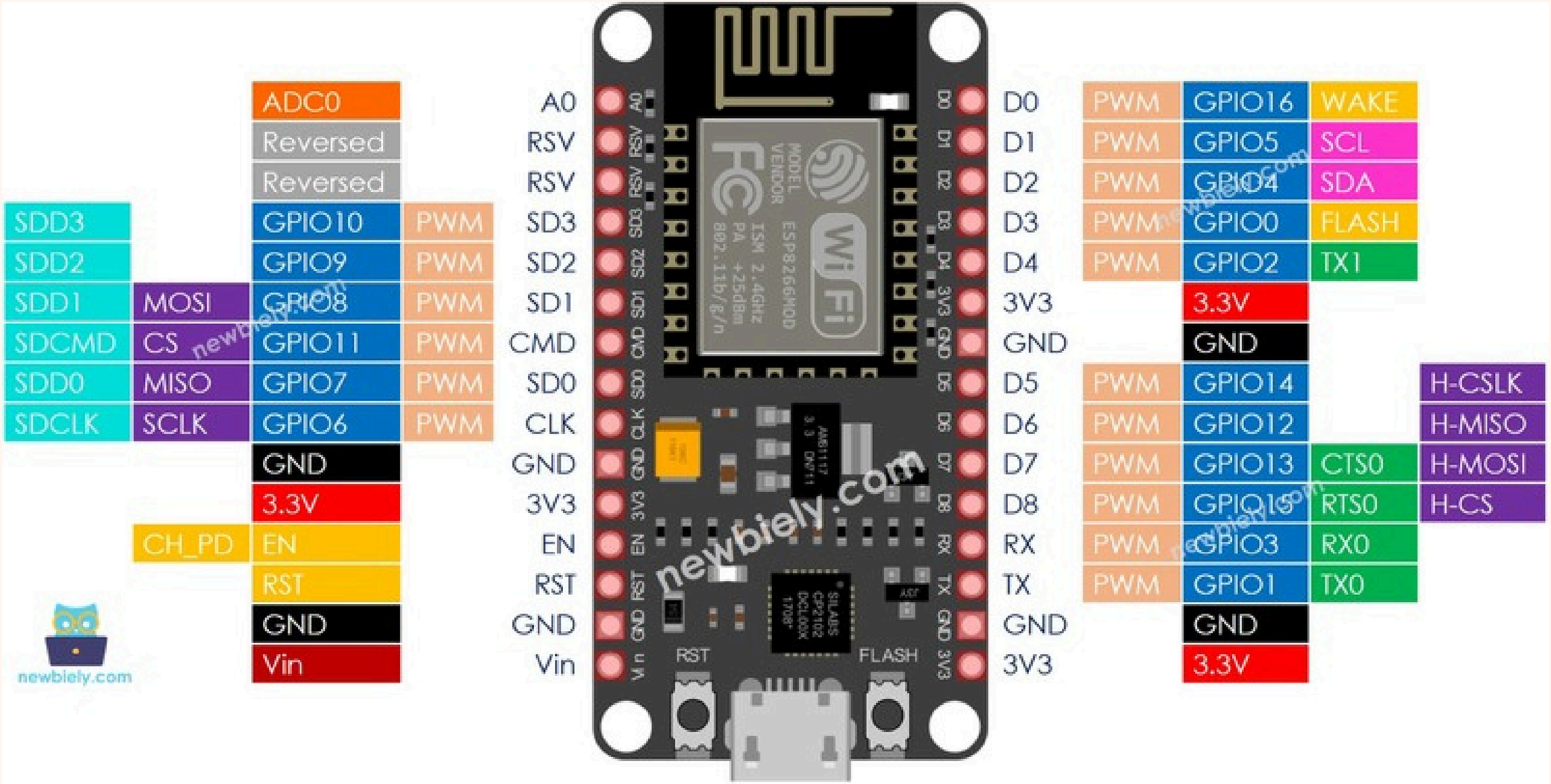
Espressif Logo

ESP 8266
POR QUE?

- Wi-Fi Integrado
- Baixo Custo
- Compatibilidade: A programação pode ser feita via comandos AT, usando a linguagem LUA ou a IDE do Arduino.
- GPIOs Versáteis
- Baixo Consumo de Energia:
- Grande Comunidade



PIN OUT



ESP8266 NodeMCU Pinout

DATASHEET LEITURA

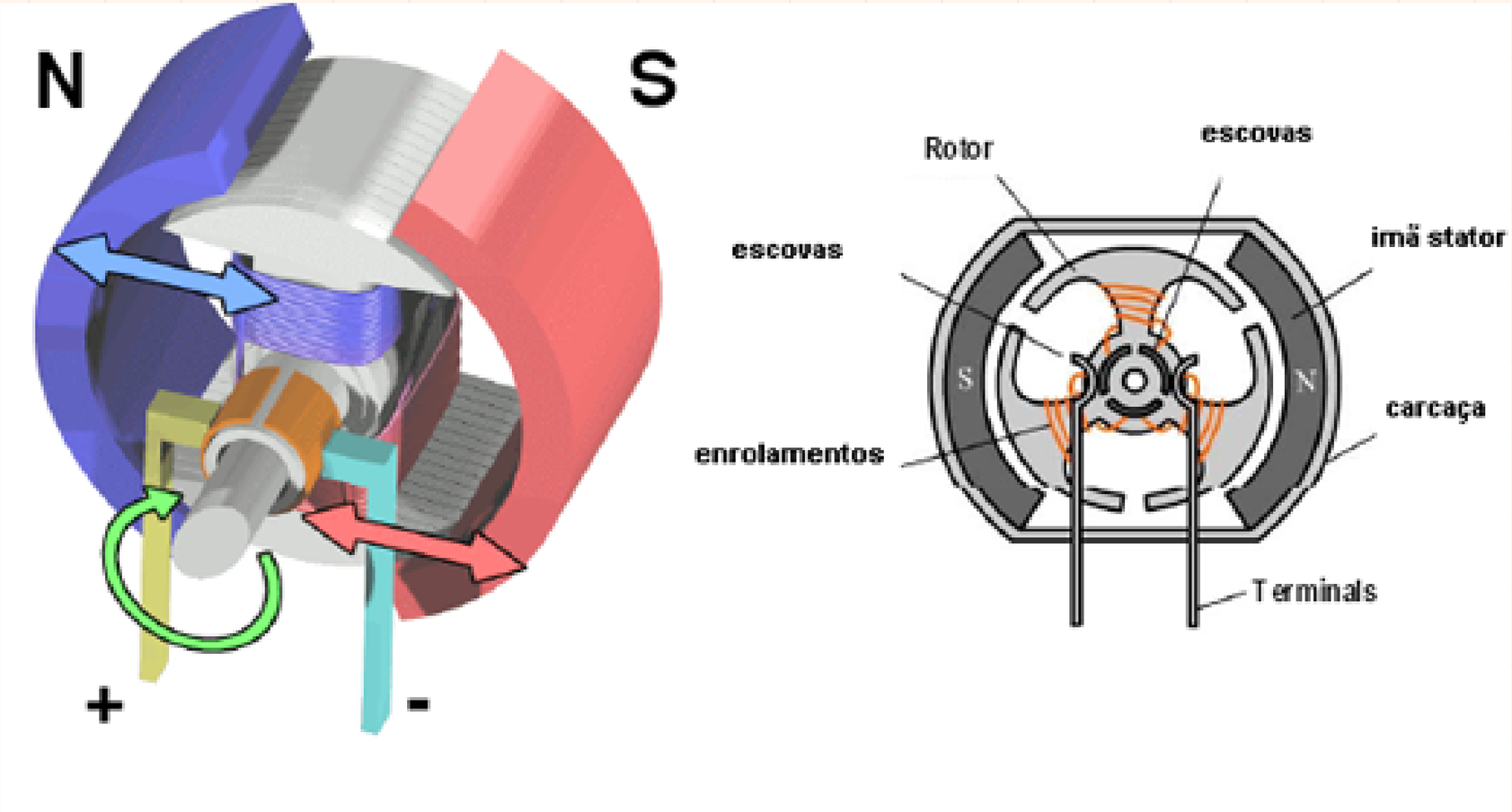
Chip ESP

node mcu

MOTOR DC
COMO?

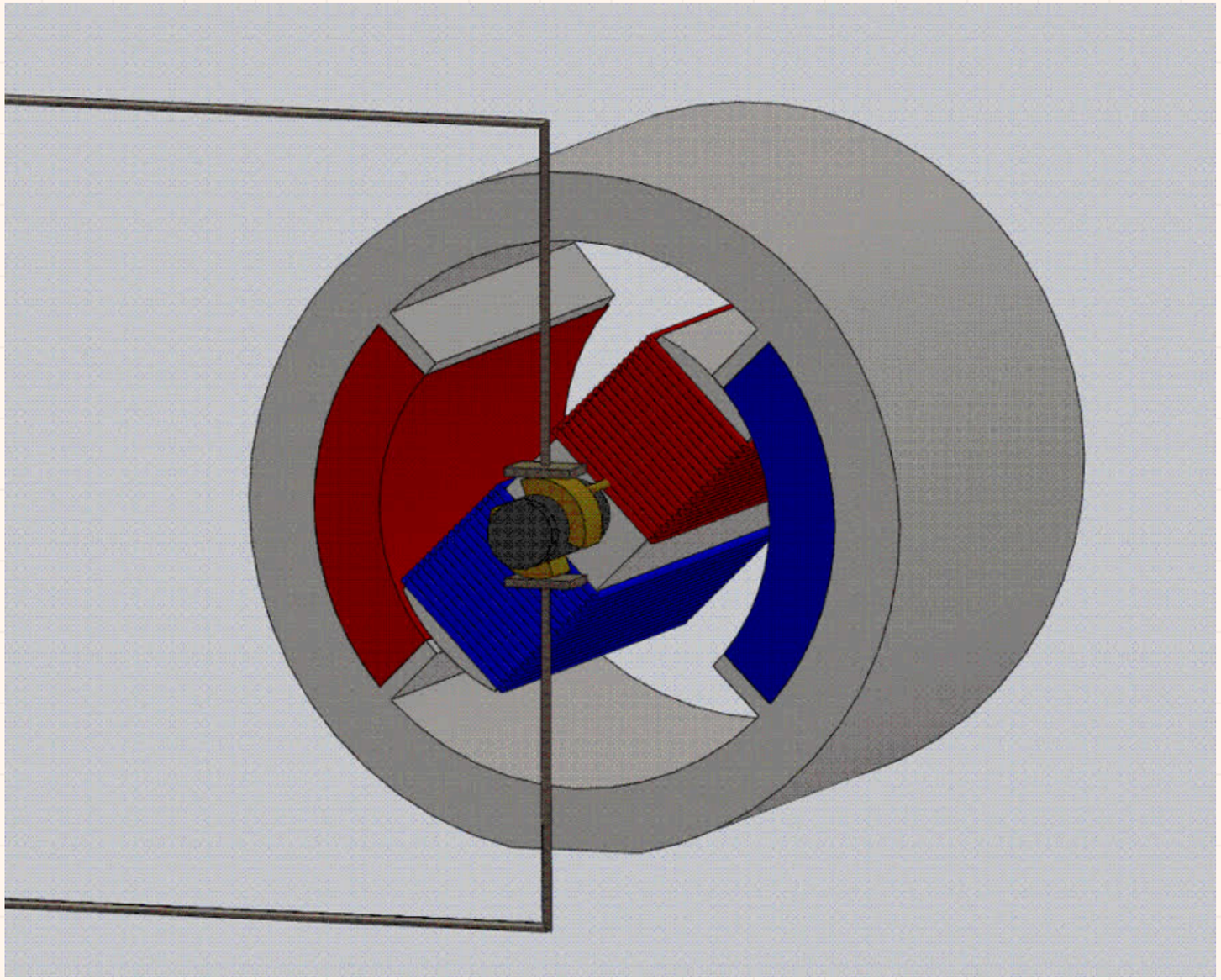
A corrente circula sobre uma bobina imersa em um campo magnético, surgindo uma força perpendicular ao campo após a energização.

Ou seja, a bobina ganha polaridade e o imã impulsiona a bobina que agora é um imã também

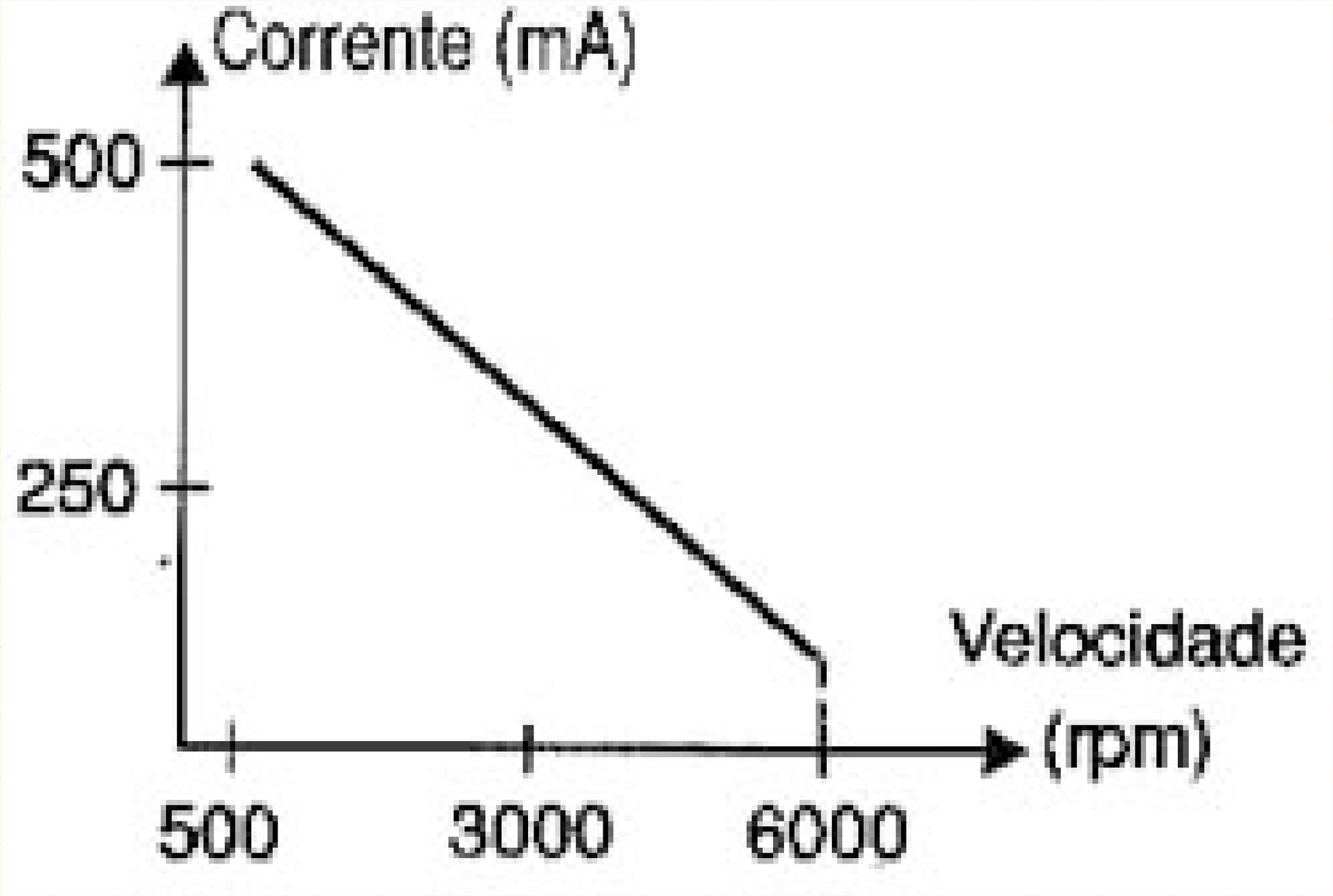


MOTOR DC
COMO?

Após a energização, o motor precisa de torque para vencer a inércia. Esse torque é proporcional à corrente, ao fluxo magnético e à área da bobina. Como ele se inverteria a cada meia volta, a arquitetura do motor utiliza um sistema especial para manter a rotação sempre no mesmo sentido.



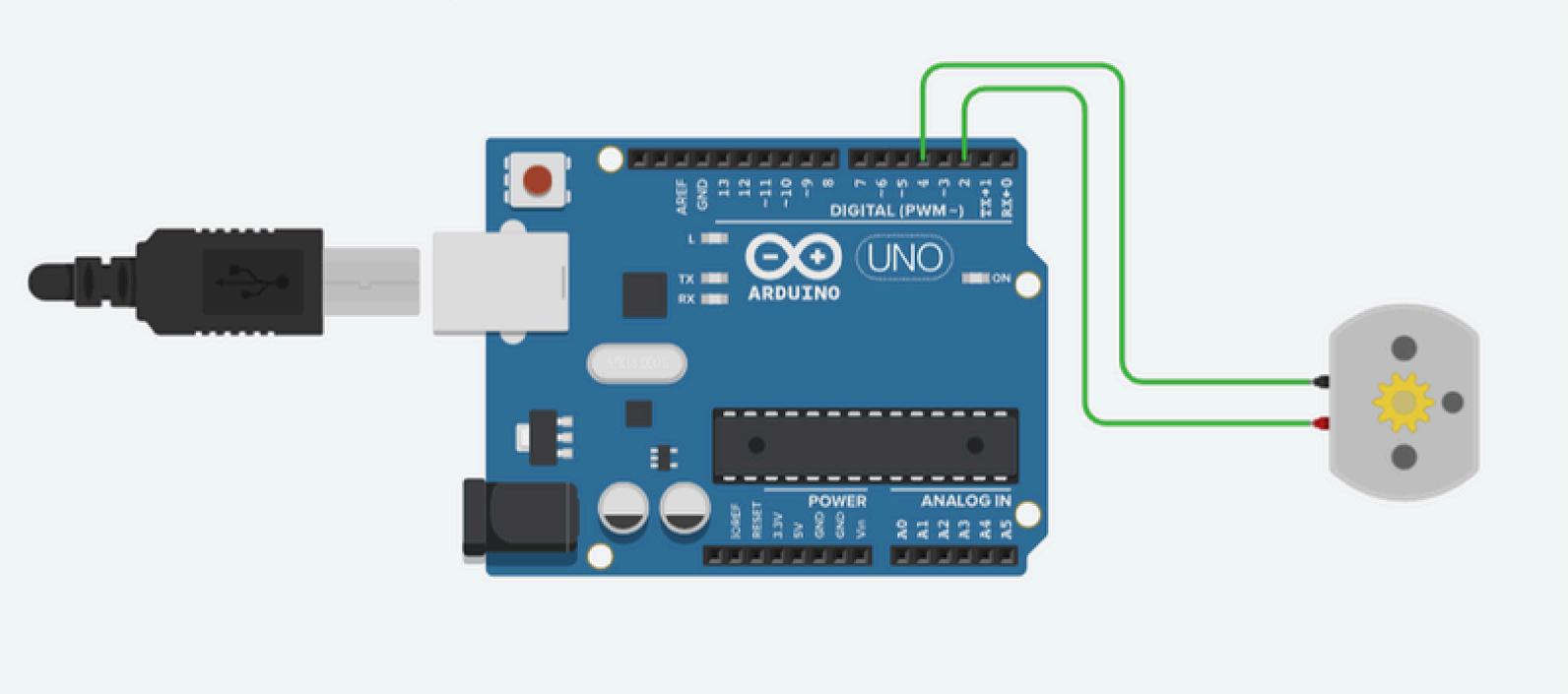
MOTOR DC
COMO?



MOTOR DC
FUNCIONAMENTO

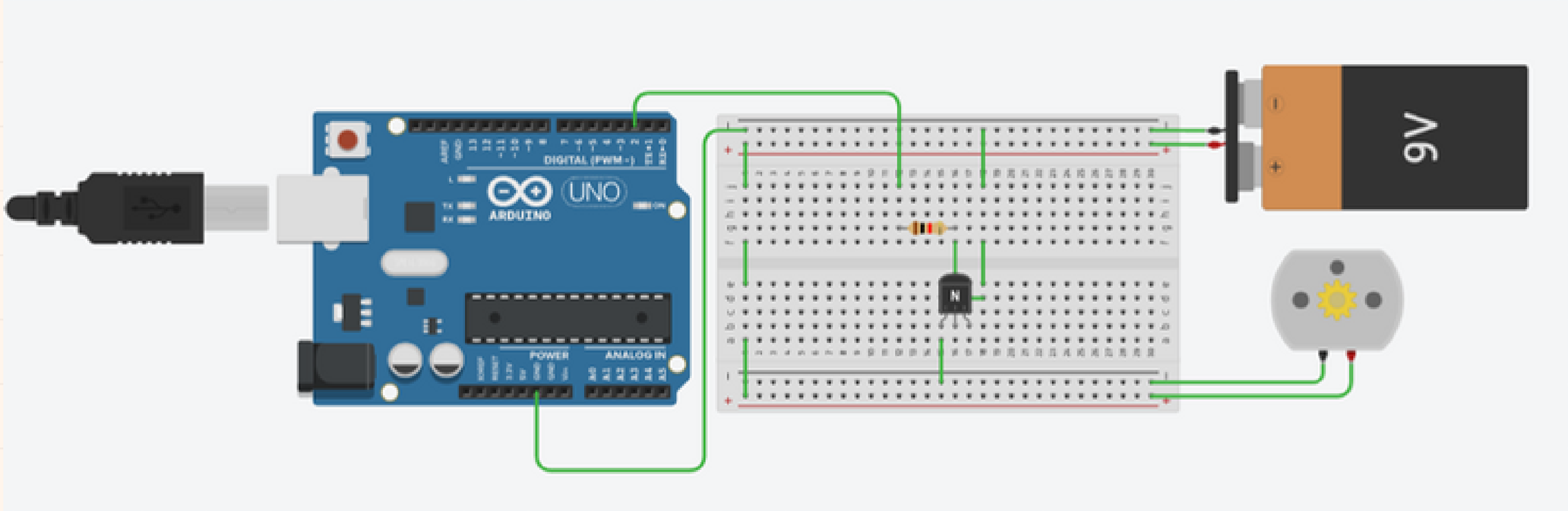
MOTOR A	IN1	IN2
HORÁRIO	5V	GND
ANTI-HORÁRIO	GND	5V
PONTO MORTO	GND	GND
FREIO	5V	5V

MONTAGEM SIMPLES



POR QUE NÃO USAMOS?

SOLUÇÃO SIMPLES

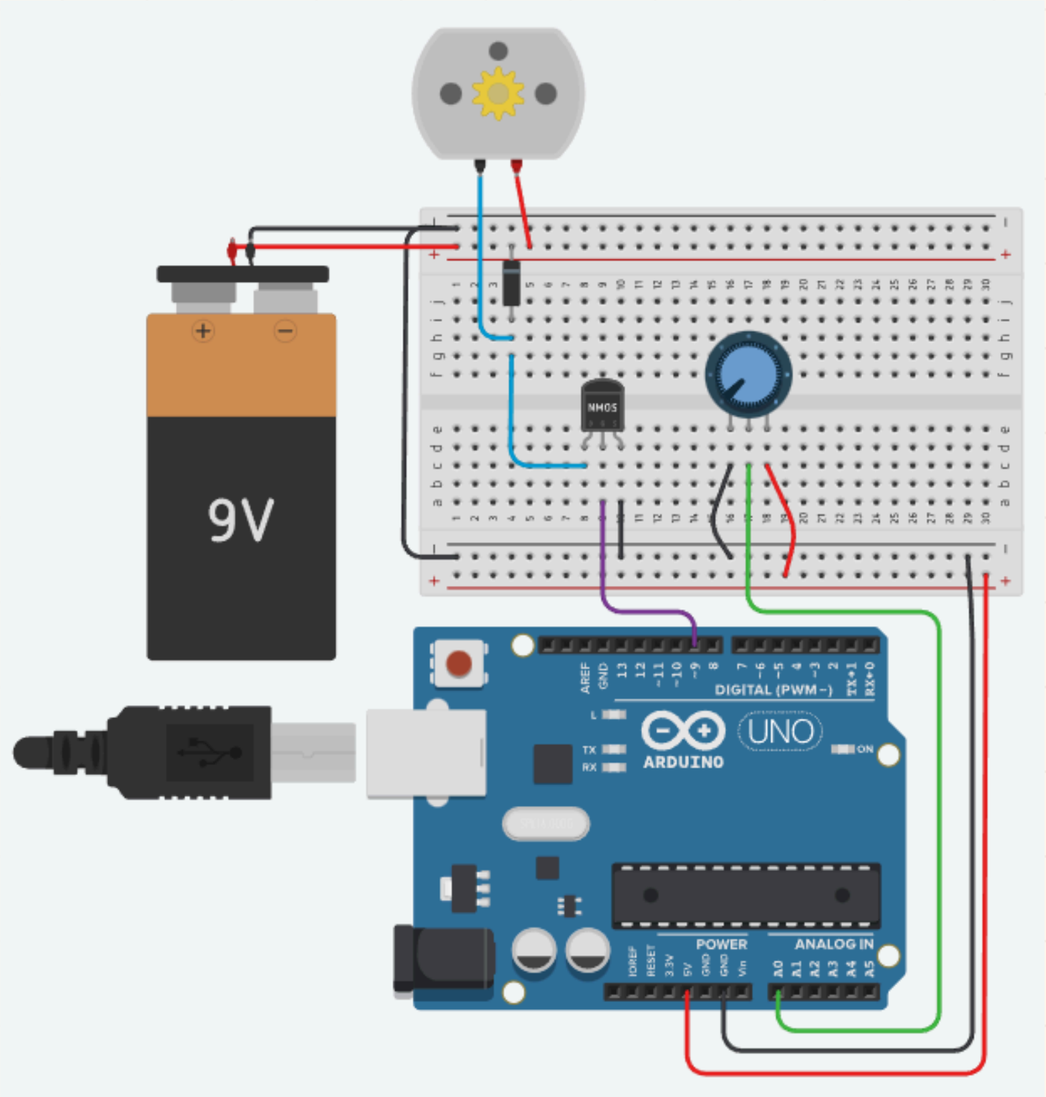


CÓDIGO

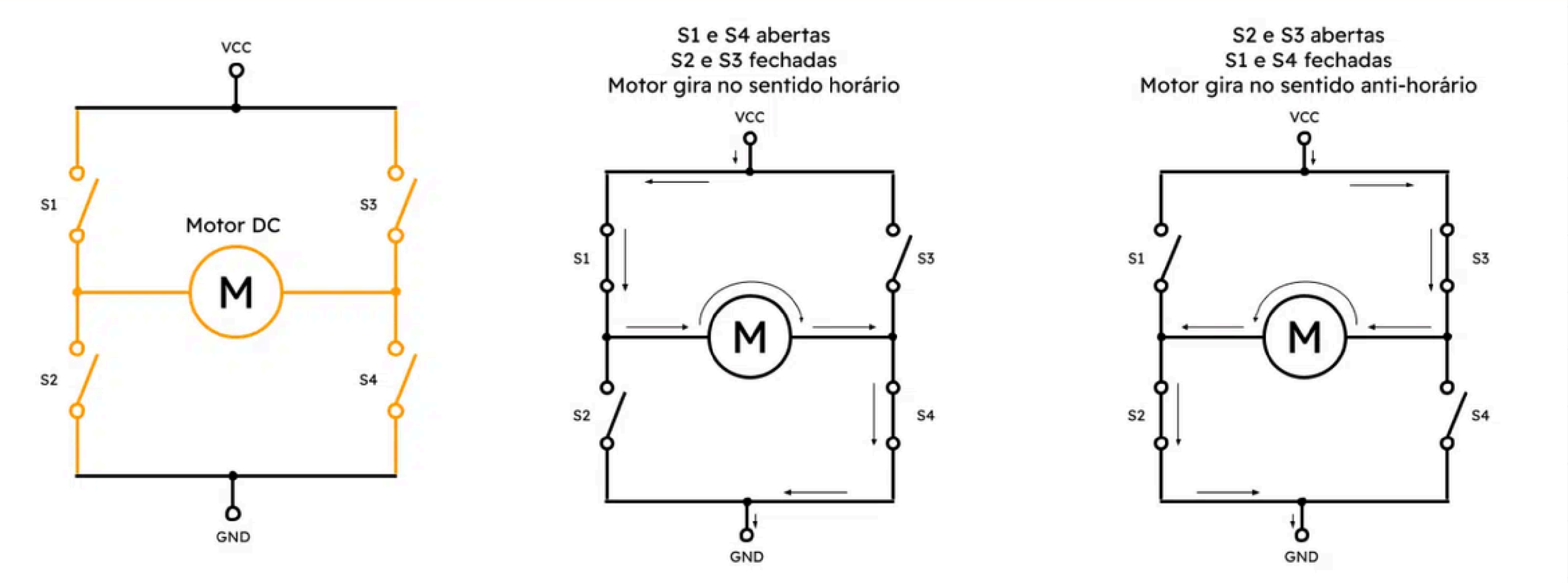
```
#define motor 2

void setup() {
    pinMode(motor, OUTPUT);
    digitalWrite(motor, HIGH);
}

```

MONTAGEM “COMPLEXA”

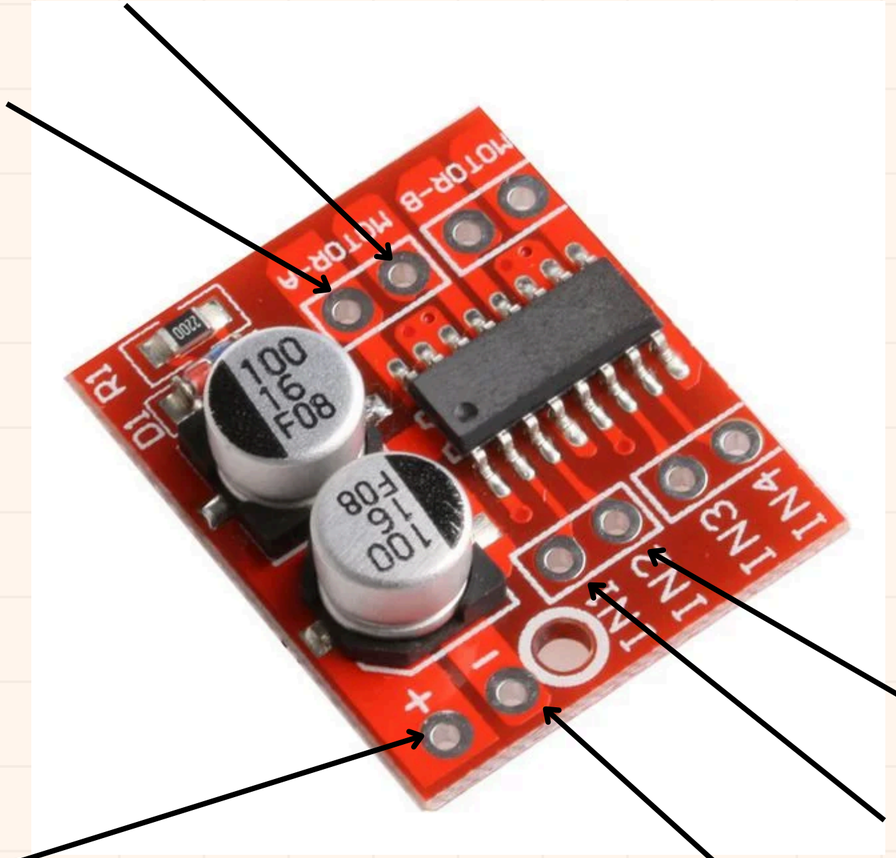
```
#define pot A0
#define motor 9
#define tmp 10

int valpot;

void setup() {
    pinMode(motor, OUTPUT);
    pinMode(pot, INPUT);
    valpot = 0;
}

void loop() {
    valpot = map(analogRead(pot), 0, 1023, 0, 255);

    analogWrite(motor, valpot);

    delay(tmp);
}
```



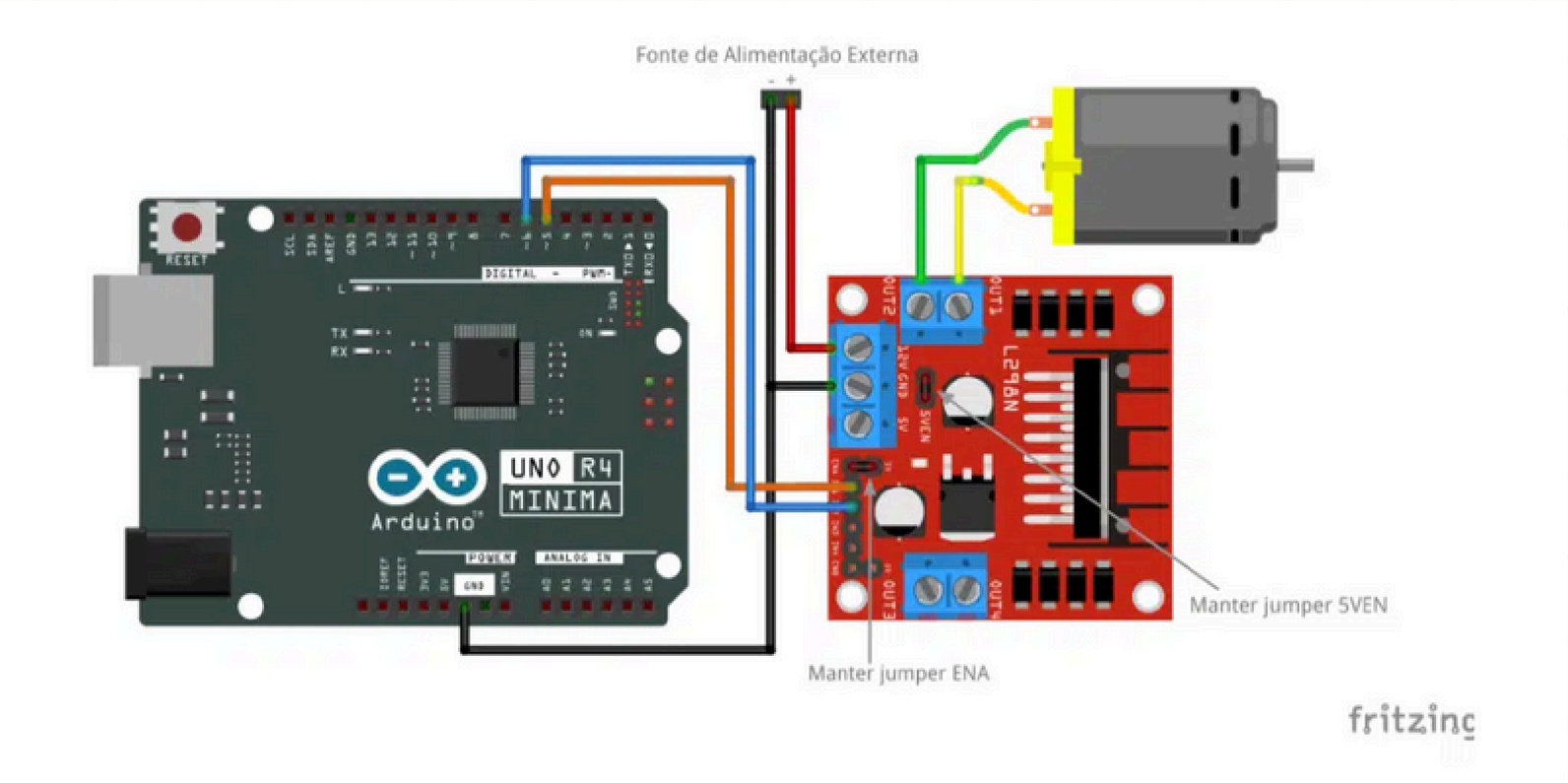
PONTE H


VCC
S1
S3
Motor DC
M
S2
S4
GND
S1 e S4 abertas
S2 e S3 fechadas
Motor gira no sentido horário
VCC
S1
S3
M
S2
S4
GND
S2 e S3 abertas
S1 e S4 fechadas
Motor gira no sentido anti-horário
VCC
S1
S3
M
S2
S4
GND

PONTE H



PONTE H APLICAÇÃO



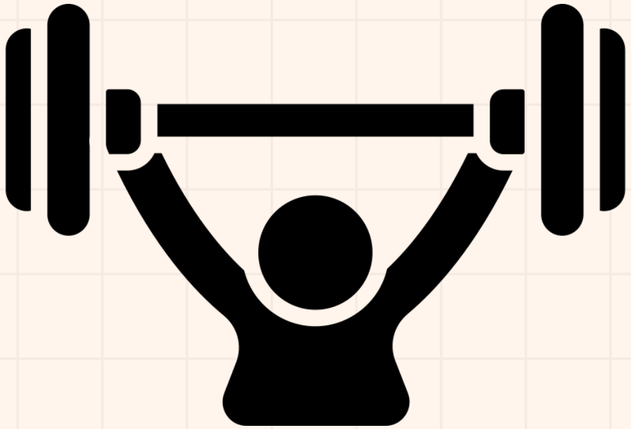
PONTE H CÓDIGO EXEMPLO

```
#define IN1 D1   // GPIO5
#define IN2 D2   // GPIO4
int velocidade = 0;

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
}

void loop() {
  // Gira em um sentido
  for (velocidade = 0; velocidade <= 255; velocidade++) {
    analogWrite(IN1, velocidade);
    digitalWrite(IN2, LOW);
    delay(10);
  }

  delay(500);

  // Gira no outro sentido
  for (velocidade = 255; velocidade >= 0; velocidade--) {
    analogWrite(IN2, velocidade);
    digitalWrite(IN1, LOW);
    delay(10);
  }

  delay(500);
}
```

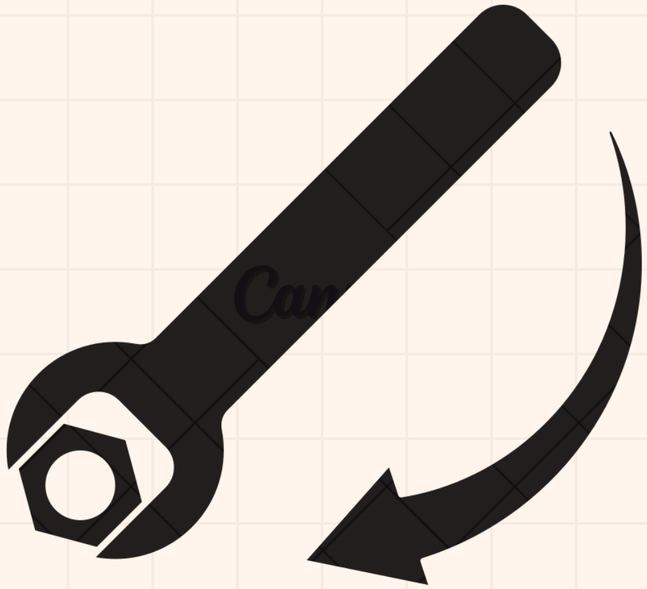
CONCEITOS BÁSICOS
DE MOTOR

MECÂNICA BÁSICA

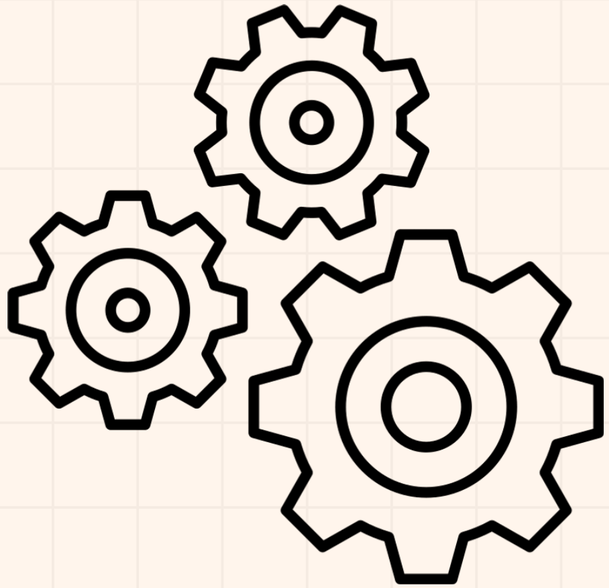
FORÇA



TORQUE



ENGRENAGENS



MECÂNICA BÁSICA

FORÇA

"É o agente da dinâmica responsável por alterar o estado de repouso ou movimento de um corpo. Quando se aplica uma força sobre um corpo, esse pode desenvolver uma aceleração, como estabelecem as leis de Newton, ou se deformar"

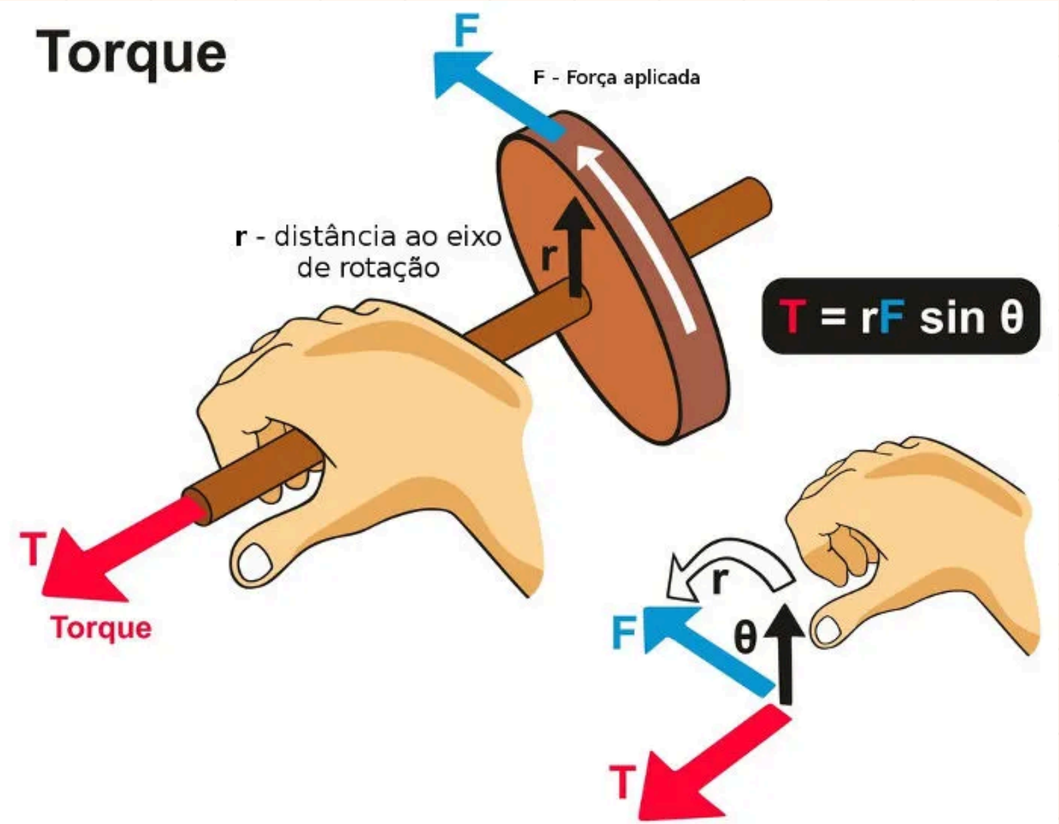
Veja mais sobre "Força" em: https://brasilescola.uol.com.br/fisica/forca.htm

TORQUE

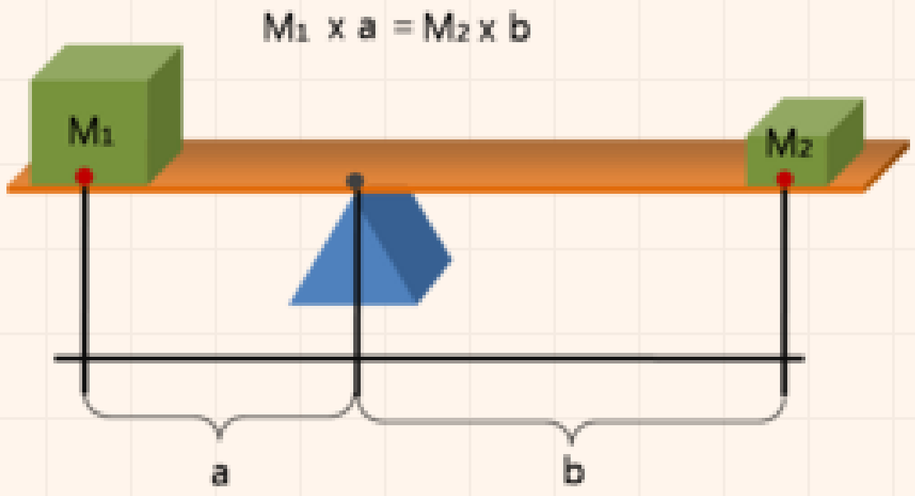
ENGRENAGENS

MECÂNICA BÁSICA

FORÇA **TORQUE** ENGRENAGENS



É a tendência que uma força tem de rotacionar um corpo sobre o qual ela é aplicada.



MECÂNICA BÁSICA

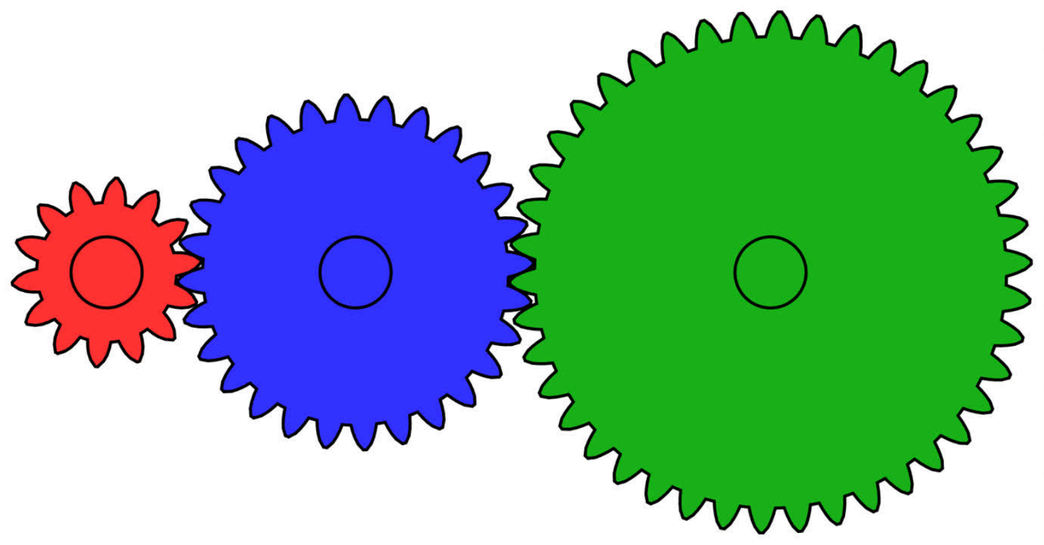
FORÇA TORQUE **ENGRENAGENS**

Engrenagens são elementos mecânicos compostos de rodas dentadas que se ligam a eixos, aos quais imprimem rotação e torque, transmitindo assim potência.
As velocidades rotacionais e os torques de duas engrenagens diferem em proporção aos seus diâmetros. Os dentes das duas rodas de uma engrenagem têm todos o mesmo formato.

ENGRENAGENS

A RAZÃO ENTRE O NÚMERO DE DENTES NAS RODAS É DIRETAMENTE PROPORCIONAL À RAZÃO DE TORQUE E INVERSAMENTE PROPORCIONAL À RAZÃO DAS VELOCIDADES DE ROTAÇÃO. POR EXEMPLO, SE A COROA (A RODA MAIOR) TEM O DOBRO DE DENTES DO PINHÃO, O TORQUE DA ENGRENAGEM É DUAS VEZES MAIOR QUE O DO PINHÃO, AO PASSO QUE A VELOCIDADE DESTE É DUAS VEZES MAIOR QUE A DA COROA.

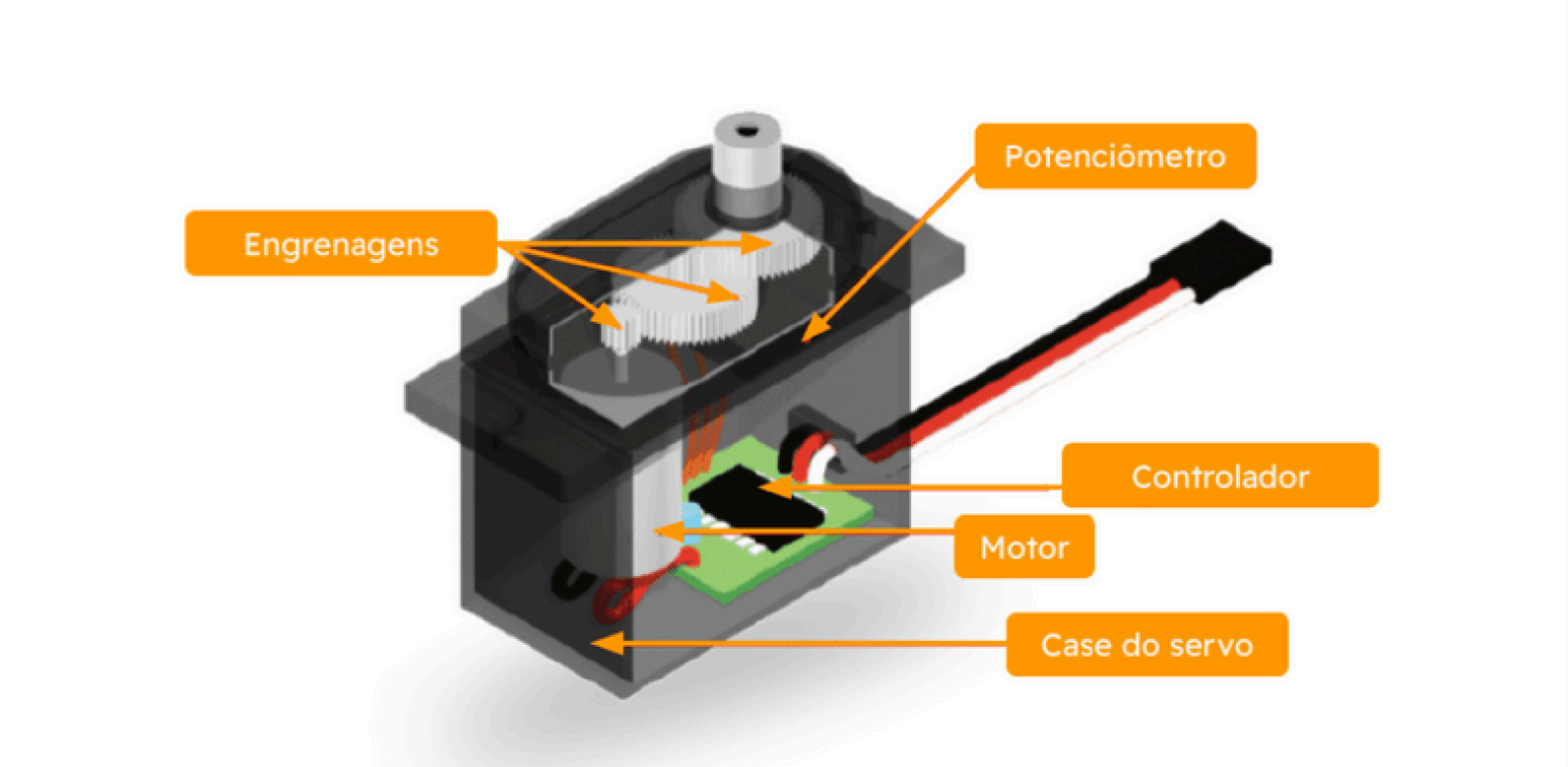
$$\frac{n_1}{n_2} = \frac{z_1}{z_2}$$



14 28 42

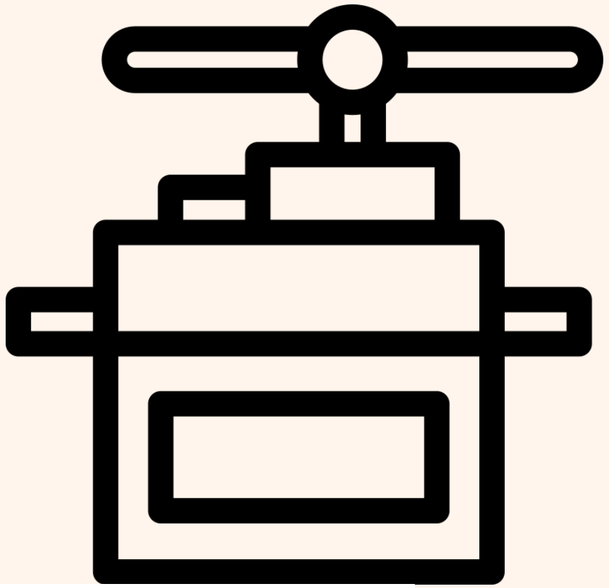
MICRO SERVO MOTOR

O QUE É?



É um motor constituído por um drive, por um motor DC, um potenciômetro e engrenagens.

MOTOR MICRO SERVO



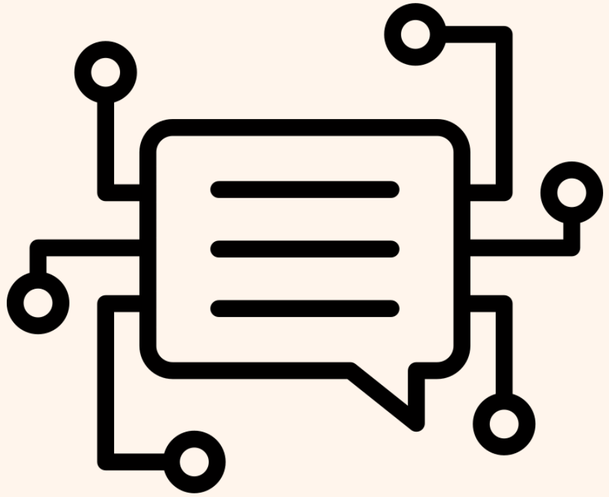
Micro Servo 9g SG90

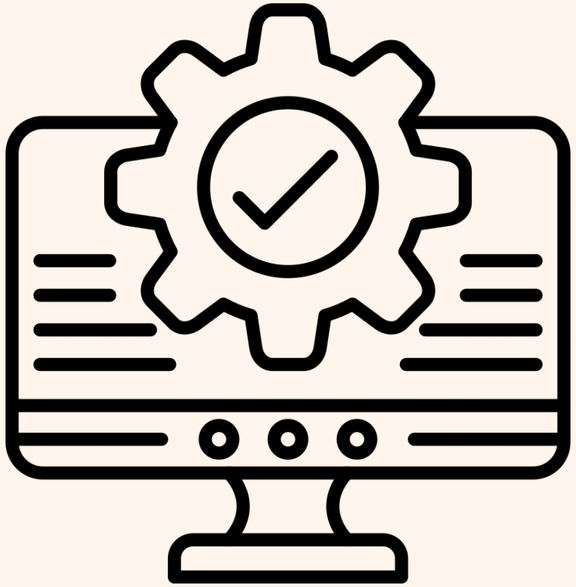
- Torque: ~1,8 kg·cm (4,8 V), suficiente para pequenos mecanismos.
- Tensão de operação: 4,8 – 6,0 V.
- Controle: Sinal PWM (pulso 0,5 ms → 0°, 1,8 ms → 90°, 2,4 ms → 180°).

Aplicações:

- Robótica leve
- Aeromodelismo
- Braços robóticos compactos
- Modelos de aviões e drones mini
- Sistemas de microajuste em protótipos

MAS COMO SE COMUNICAR COM O SERVO





PWM
ONDAS E DUTYCYCLE

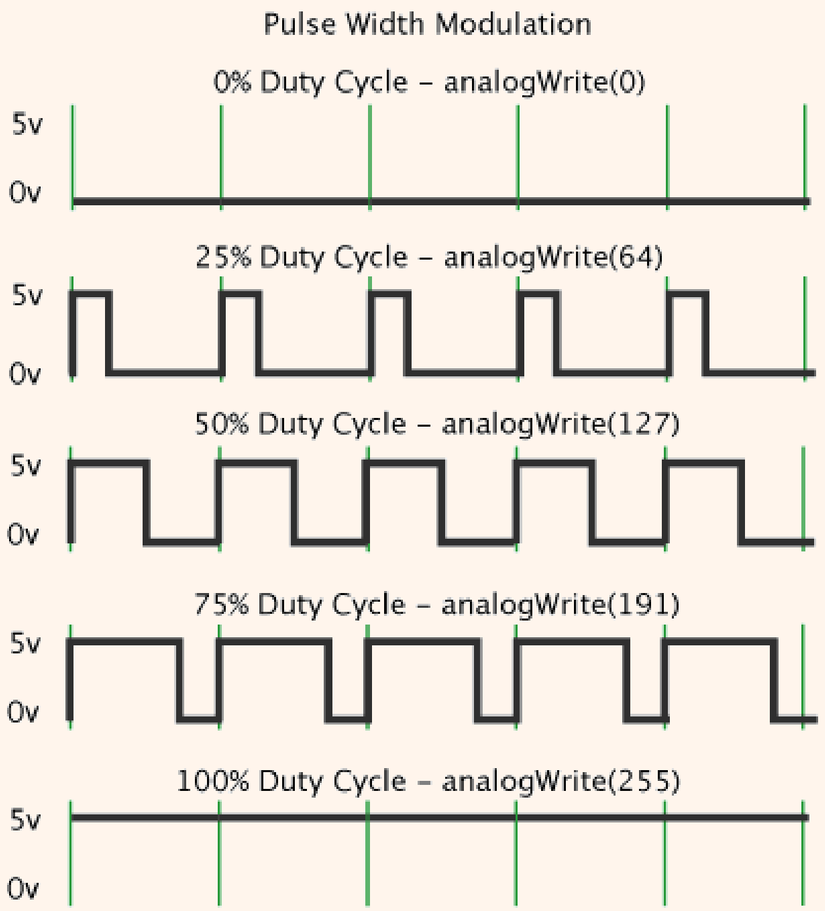
Caracteristicas do PWM ou Pulso de Largura Modulada:

- **Frequência típica: 50 Hz (período de 20 ms).**
- **Largura do pulso define o ângulo:**

  0,5 ms → 0°

  1,8 ms → 90°

  2,4 ms → 180°
- **Duty Cycle varia de 2,5% (0,5/20 ms) a 12% (2,4/20 ms).**



DUTY CYCLE

MATEMÁTICAMENTE

PODEMOS DEFINIR DESTA MANEIRA:

$$D = \frac{t_{on}}{T}$$

Onde :

- t_{on} é o tempo em que o sinal está em high;
- T é a diferença de t_{on} e t_{off};

PWM
MATEMÁTICAMENTE

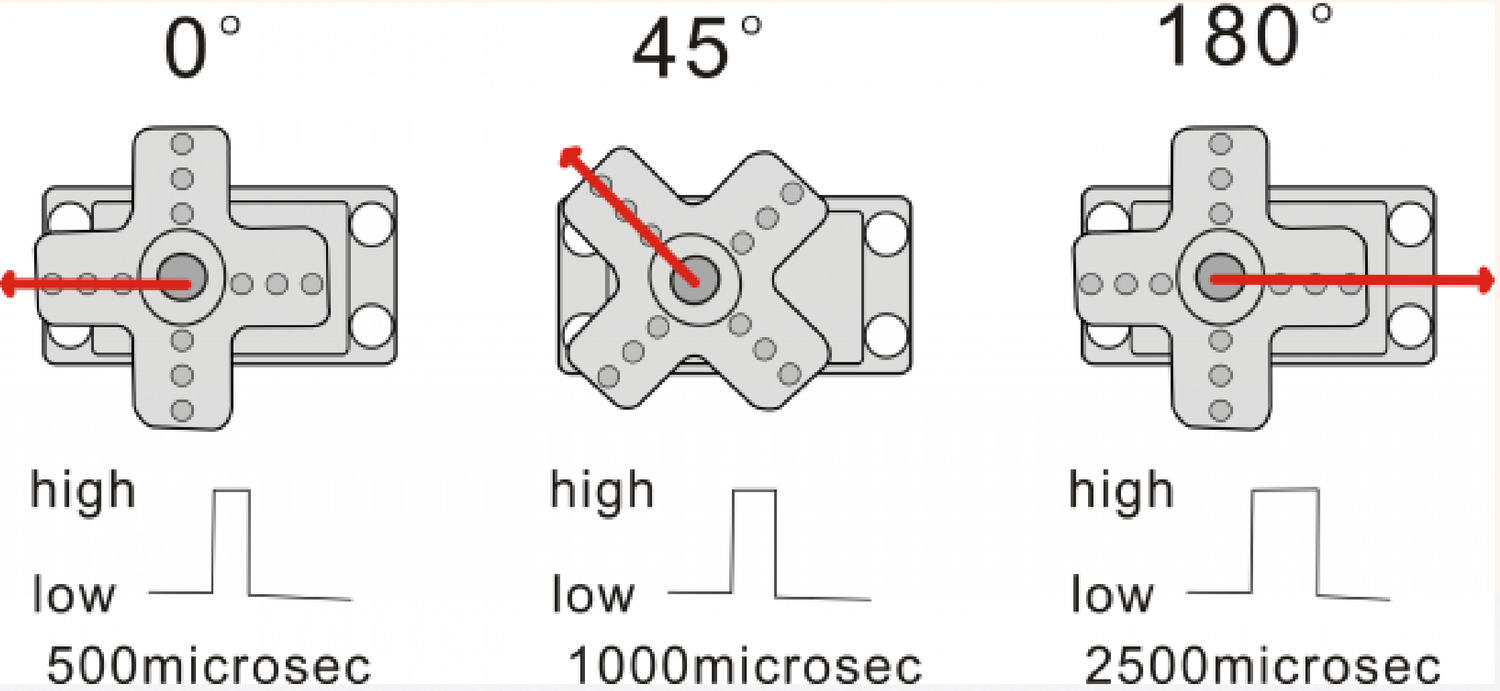
Podemos Definir o PWM desta maneira:

$$\mathbf{PWM} = \begin{cases} \mathbf{A, \text{ se } t \textit{ mod } T < D * T} \\ \mathbf{0, \text{ caso contrário}} \end{cases}$$

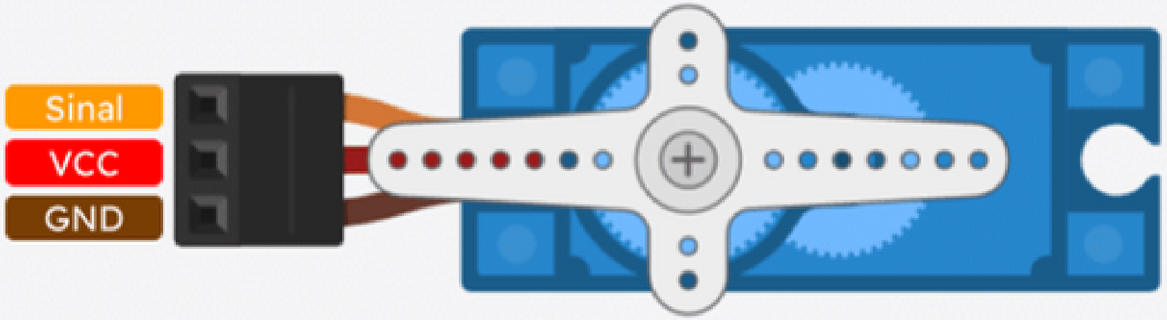
Onde:

- t é o tempo;
- T é o periodo;
- D é o Duty cycle;
- A é a Amplitude
- t *mod T é o tempo no ciclo*

MOTOR MICRO SERVO
ENTRADAS E ANGULOS:

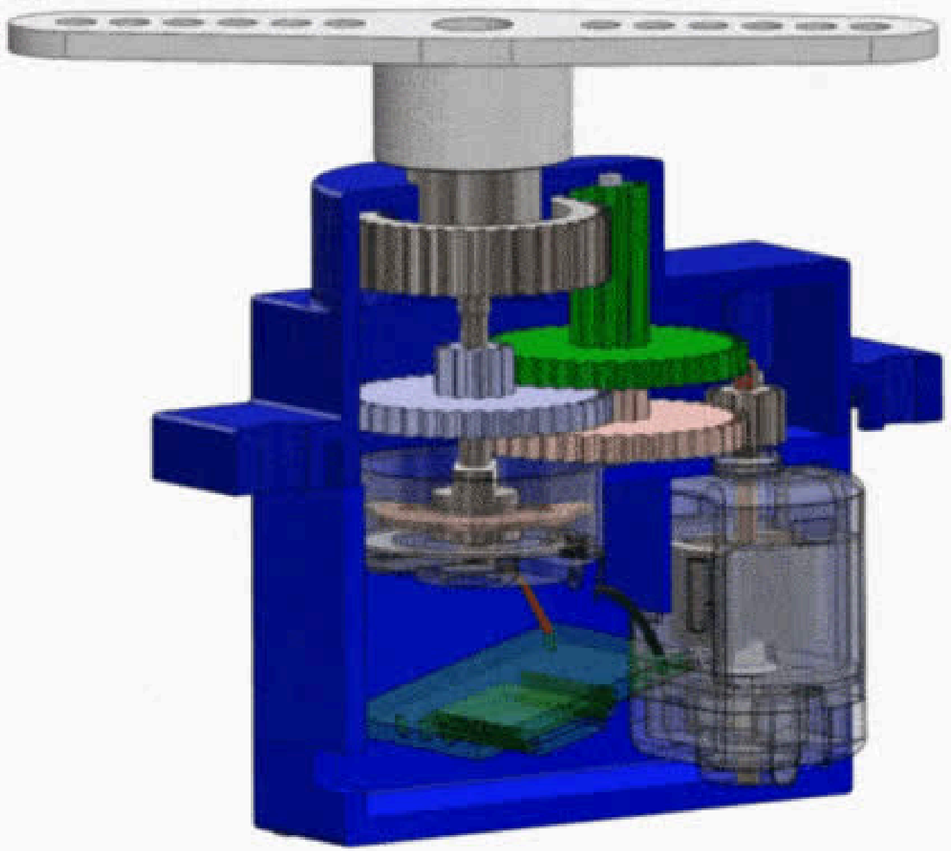


PWM	Angulo
500	0°
1000	45°
1500	90°
2000	135°
2500	180°

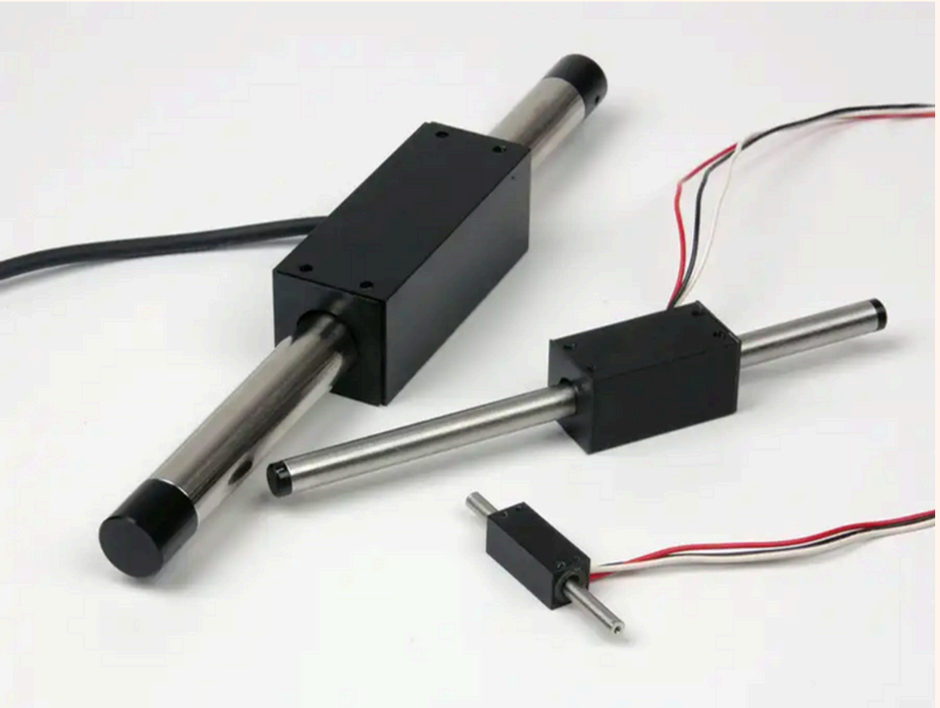


MOTOR MICRO SERVO

EXEMPLO DE FUNCIONAMENTO:



SERVO MOTOR
TIPOS:

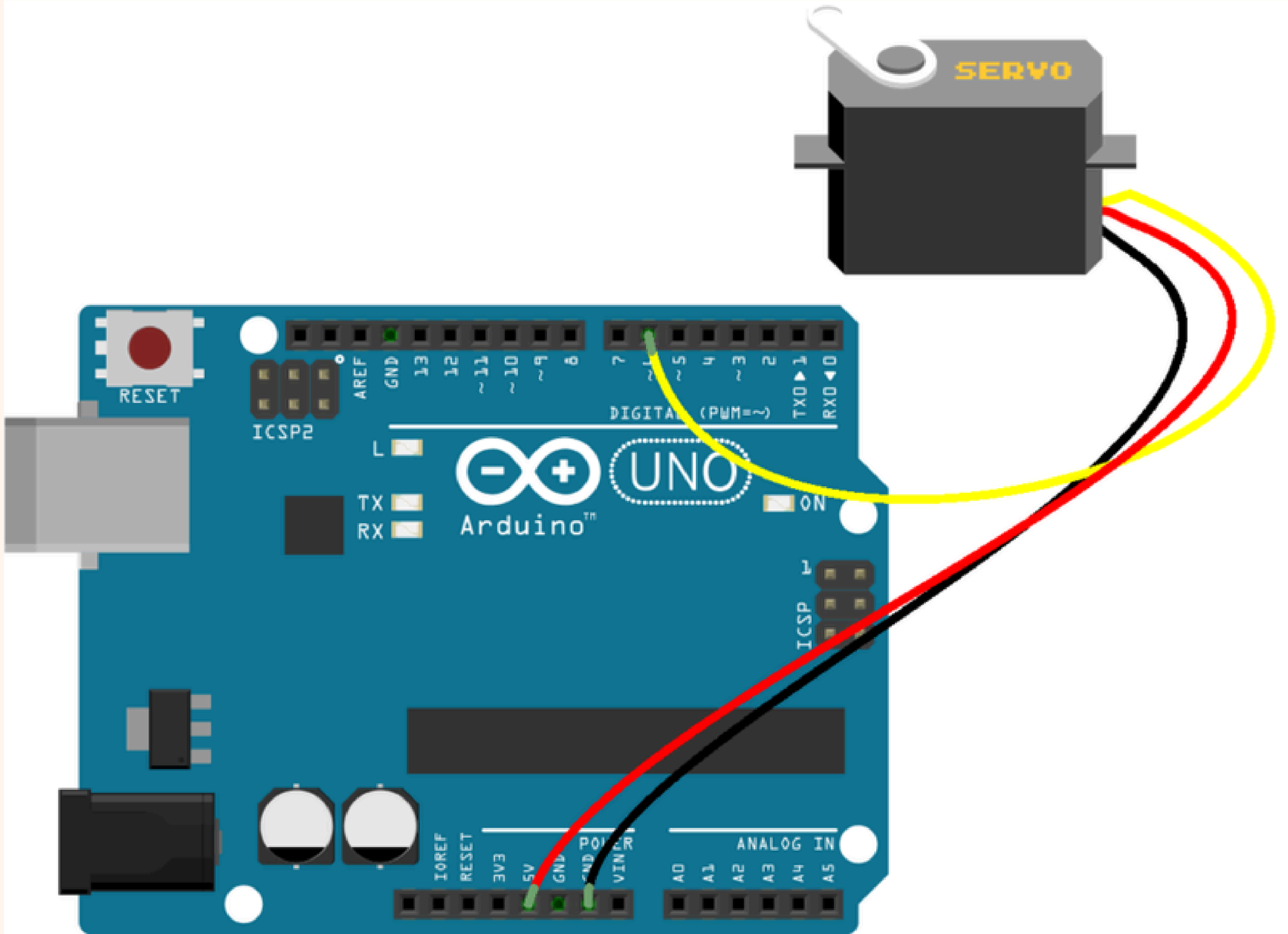


SERVO MOTOR LINEAR

SERVO MOTOR DIGITAL

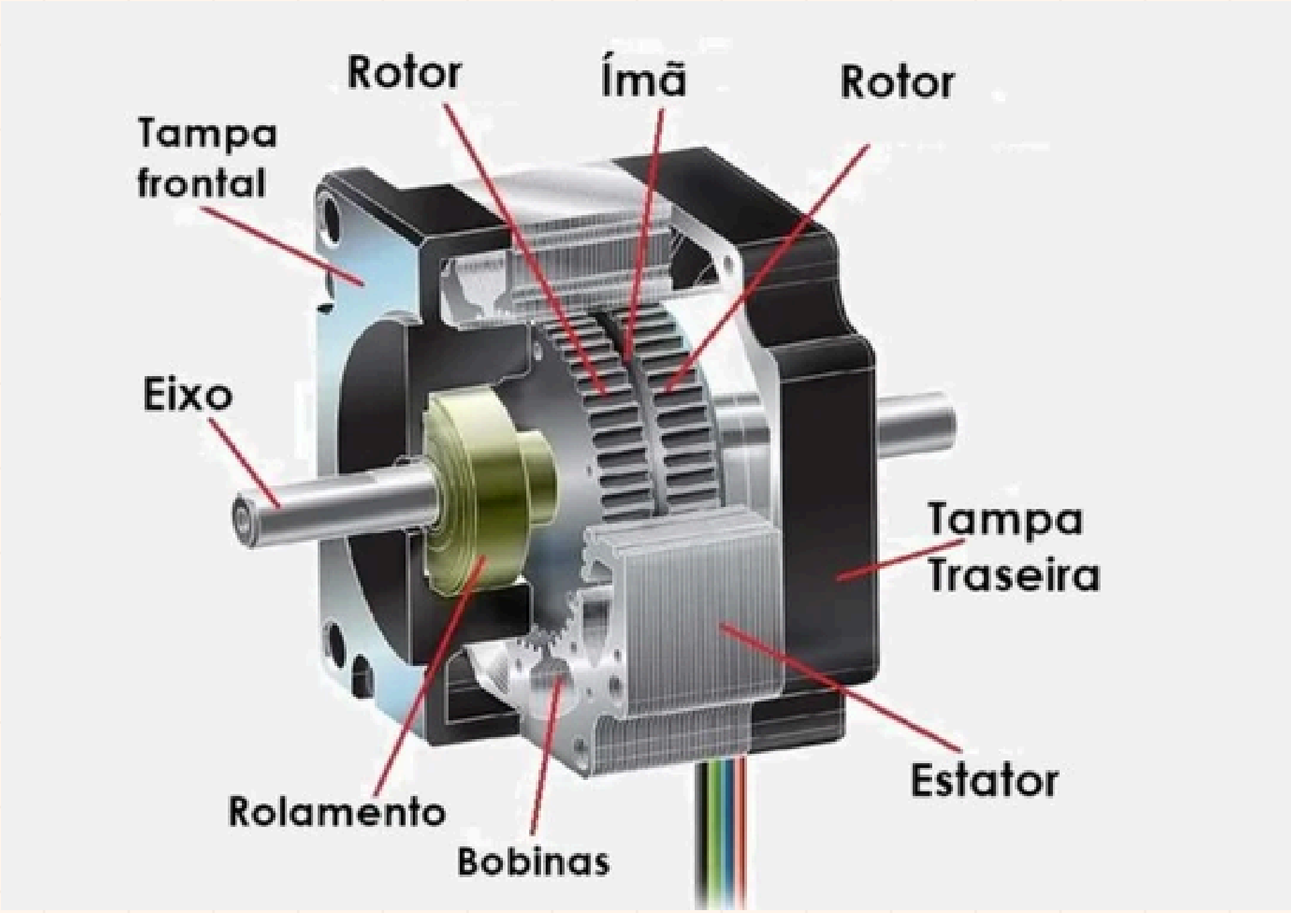
SERVO MOTOR DE CORRENTE ALTERNADA

MONTAGEM


SERVO
RESET
ICSP2
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX0
RX0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

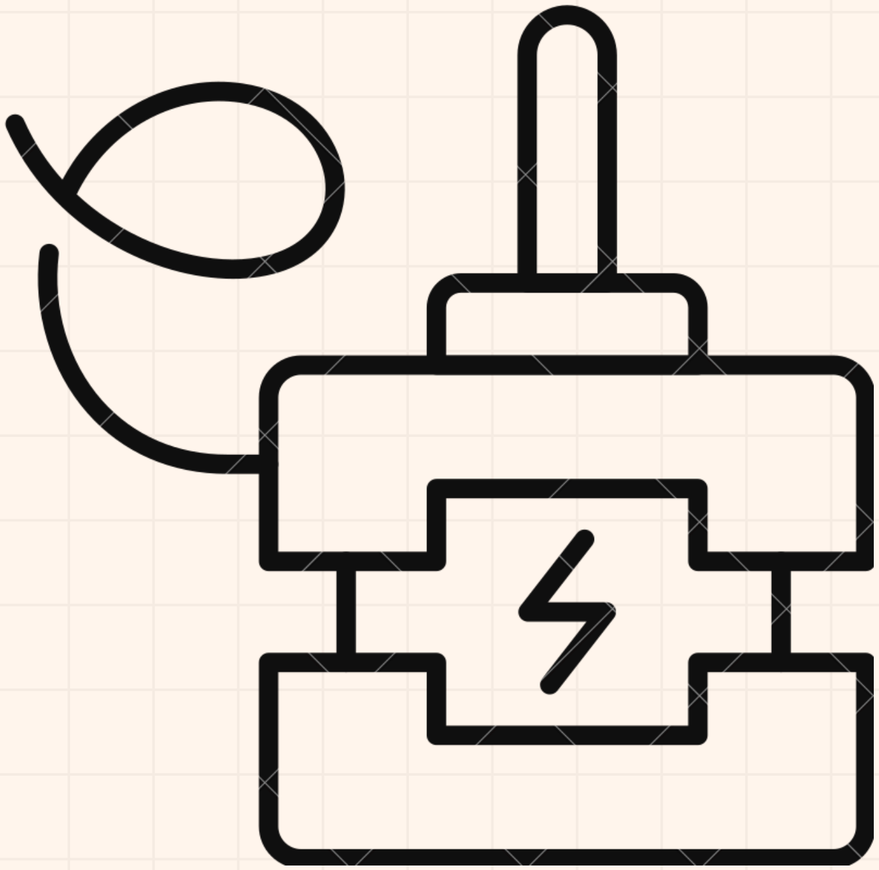
MOTORES DE PASSO

Um motor de passo é um tipo de motor elétrico projetado para mover em incrementos discretos, ou “**passos**”, ao invés de operar de forma contínua como os motores de corrente contínua (DC) ou de corrente alternada (AC). Cada rotação completa do motor é dividida em um número específico de passos, permitindo um controle preciso da posição do eixo.



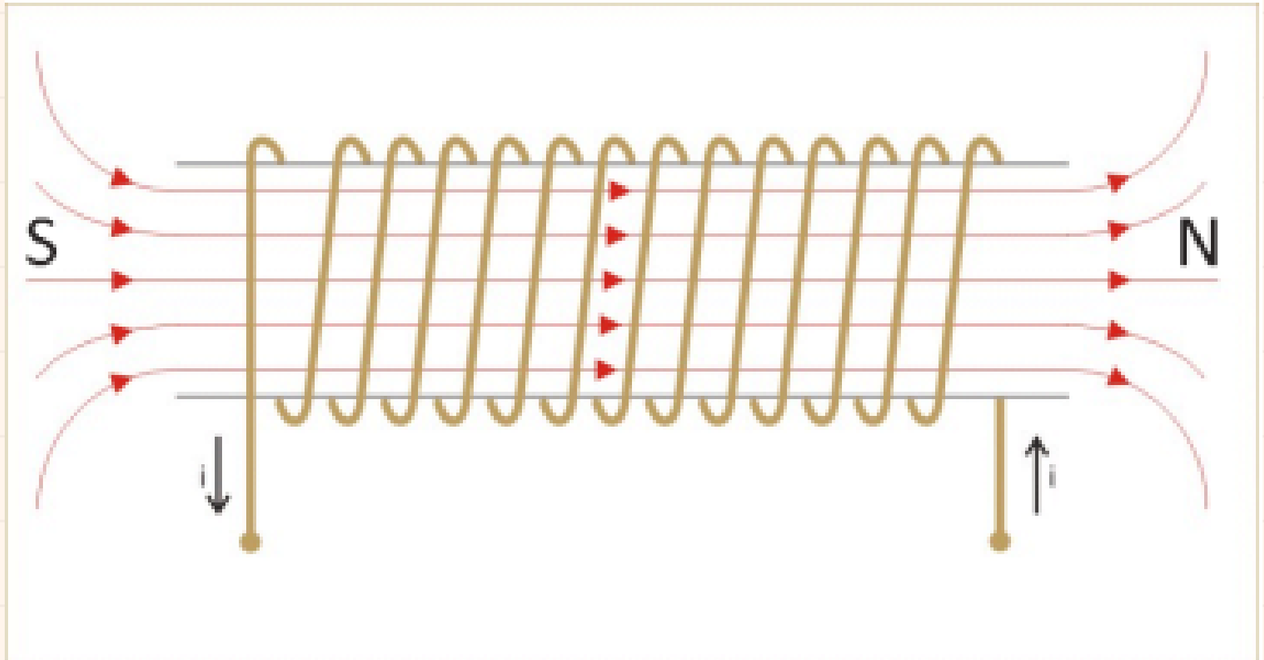
MOTORES DE PASSO

- Vantagens:
  - Movimentos Discreto (Passos)
  - Controle tão angular quanto os passos permitem
  - controle de velocidade
- Desvantagens
  - Velocidade Reduzida em relação aos motores comuns



COMO FUNCIONA?

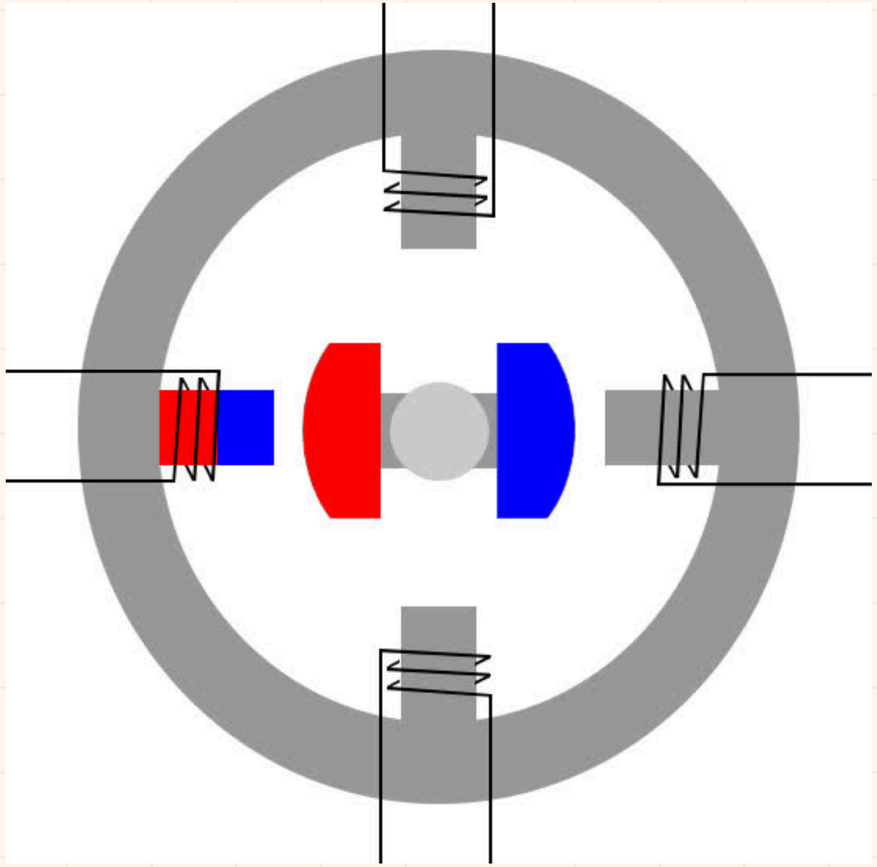
Tal como a grande maioria dos motores elétricos, o motor de passo trabalha pela interação entre campos eletromagnéticos. Quando um solenoide é percorrido por uma corrente constante, dá-se como resultado a produção de um campo magnético uniforme no seu interior.
Invertendo o sentido da corrente elétrica inverte-se o sentido do campo magnético.



COMO FUNCIONA

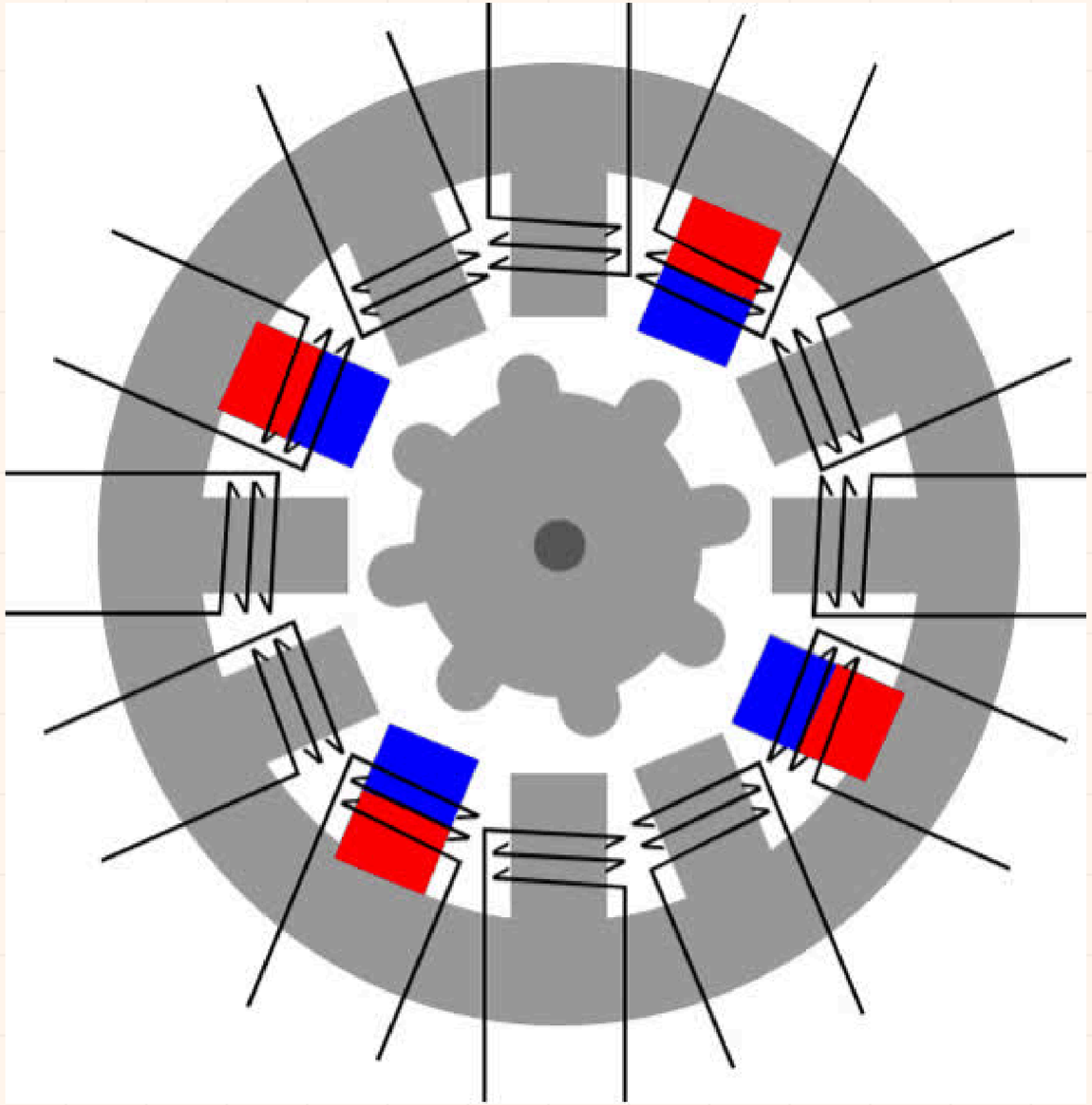
Combinando vários solenóides, e um ímã permanente posicionado entre os mesmos, pode-se controlar a posição do ímã permanente

Assim, movemos o imã permanente sempre com o mesmo ângulo, tendo assim mais precisão sobre que ângulo temos no motor.



COMO FUNCIONA?

- Esse alinhamento ocorre mediante combinação binária na entrada
- A quantidade de solenoides/bobinas definem a quantidade de passo (e os passos definem precisão angular do motor)
- Além disso podemos manter o motor parado em uma posição deixando a bobina fixa em um estado

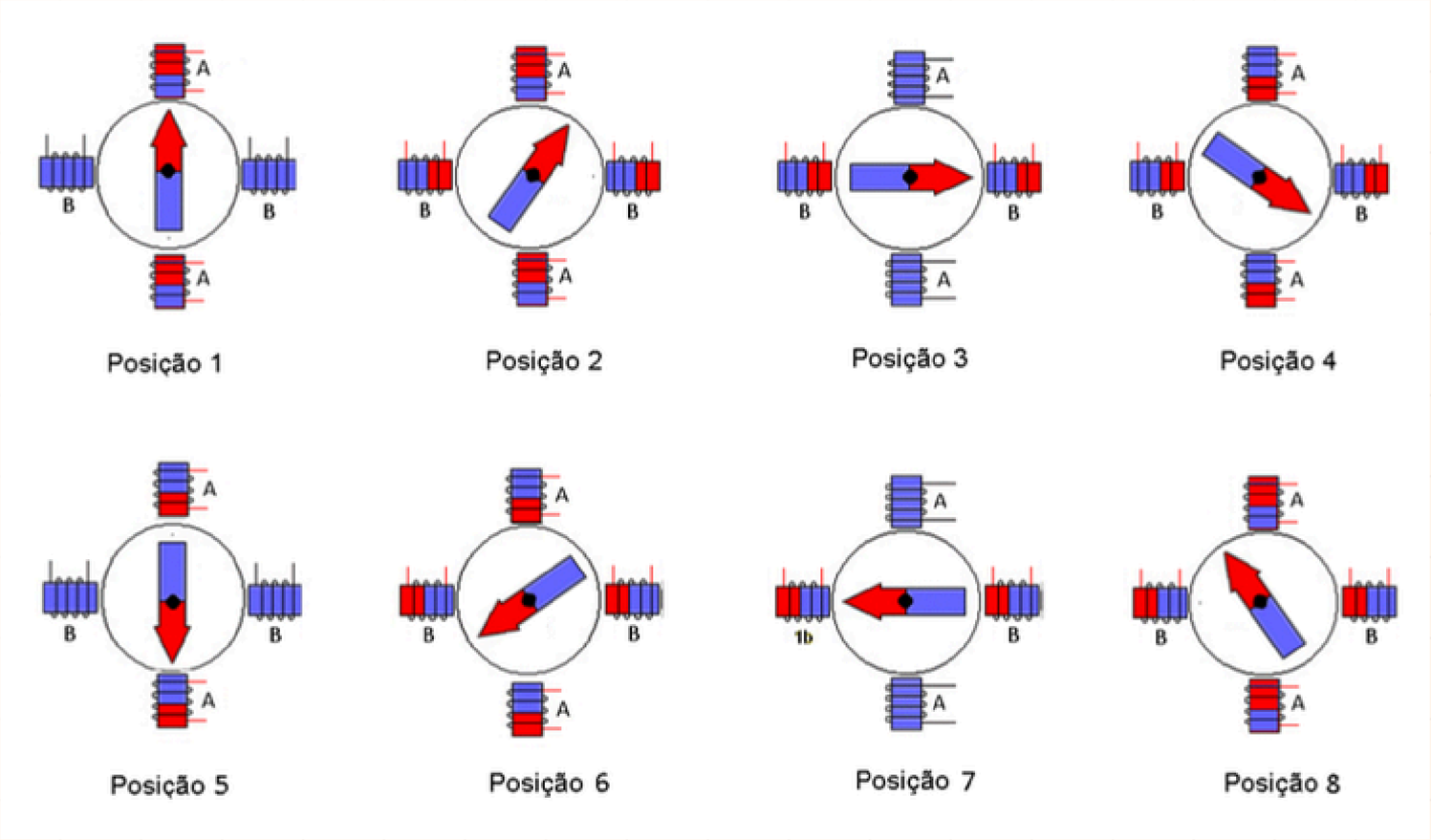


O QUE É
O PASSO?

Caso o motor tivesse um rotor com somente dois polos, como ilustrado na figura seriam necessários 8 comandos sequencias para realizar uma volta completa

$$\frac{360(graus)}{8(passos)} = 45^\circ$$

logo o seria soma dos estados possíveis. sendo diretamente proporcional ao angulo de rotação .

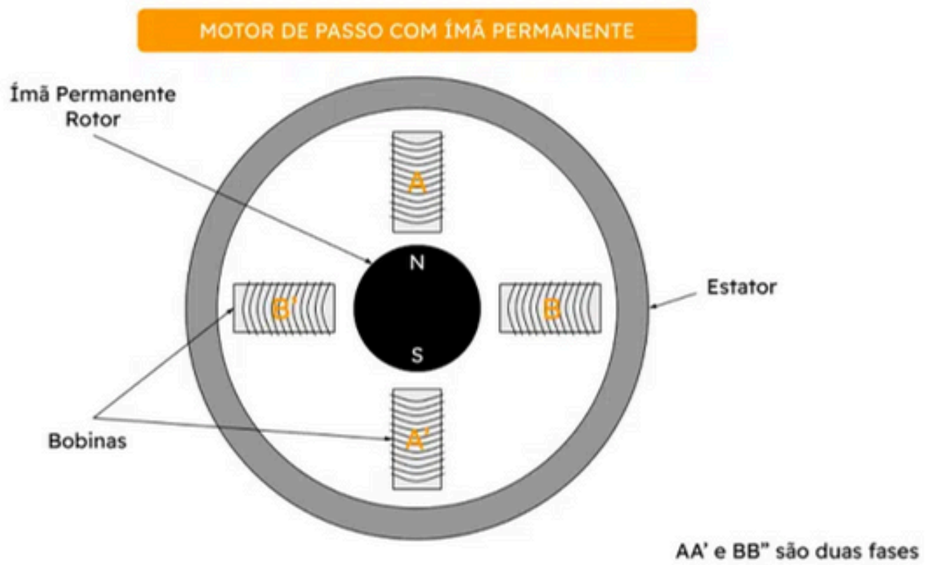


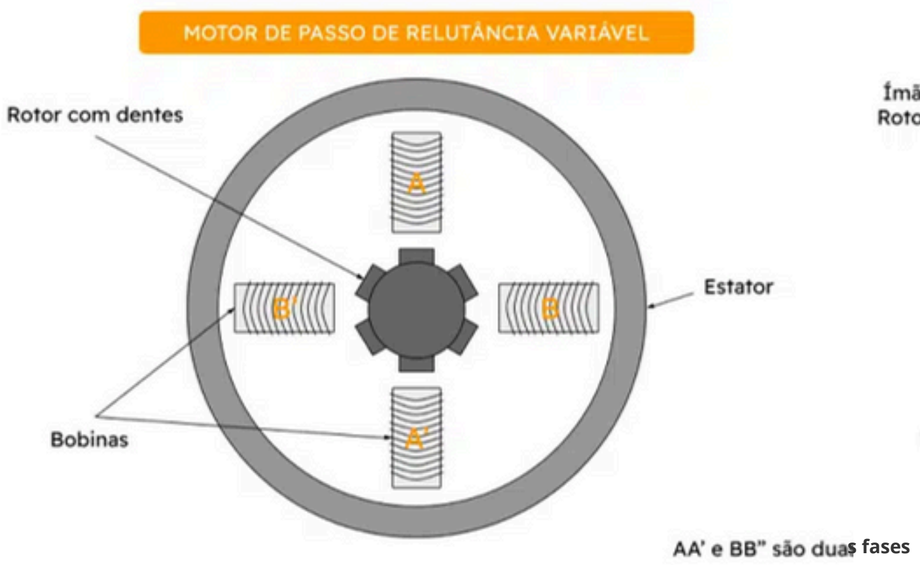
TIPOS DE MOTORES
DE PASSO

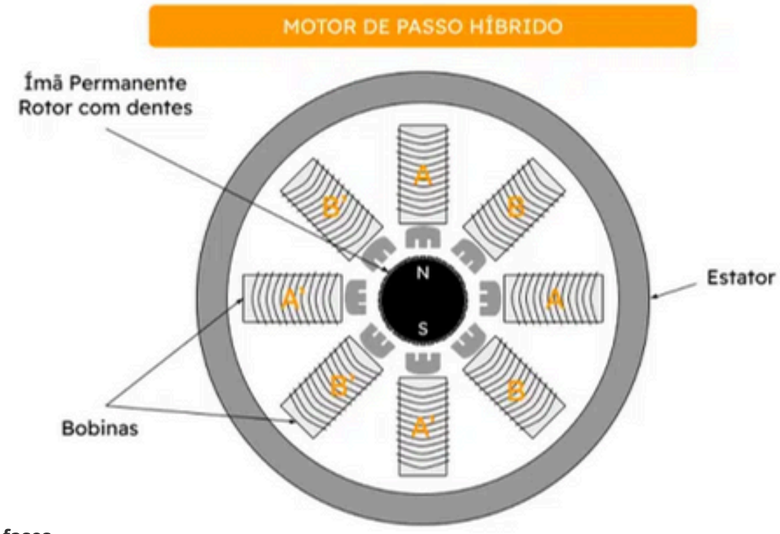
IMÃ PERMANENTE

RELUTÂNCIA VARIÁVEL

MOTORES HÍBRIDOS







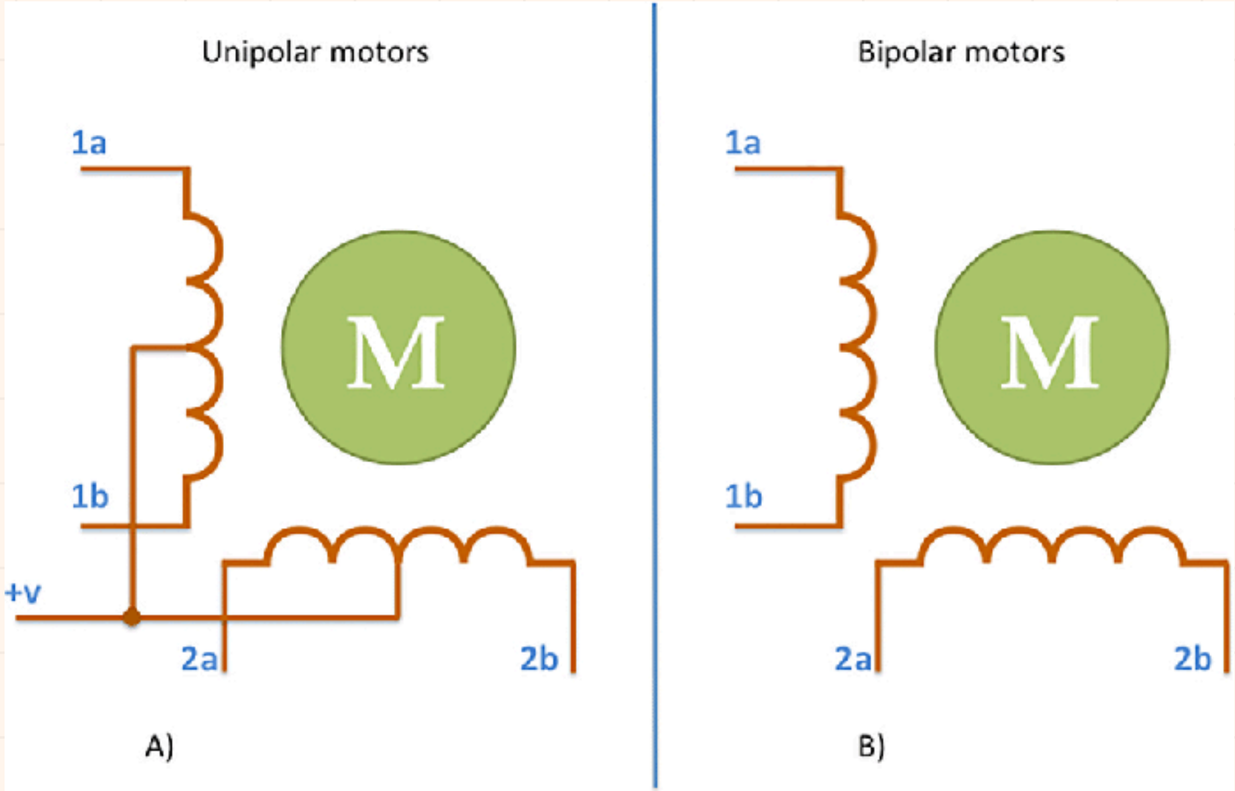
TIPOS DE MOTORES DE PASSO

Característica	Ímã Permanente (PM)	Relutância Variável (VR)	Híbrido
Torque de Retenção	Alto	Baixo	Alto
Precisão	Moderada	Baixa	Alta
Complexidade	Moderada	Simples	Alta
Custo	Médio	Baixo	Alto
Aplicações	Impressoras, dispositivos ópticos, eletrônica	Sistemas de posicionamento simples	Máquinas CNC, impressoras 3D

FORMA DE LIGAÇÃO

UNIPOLAR

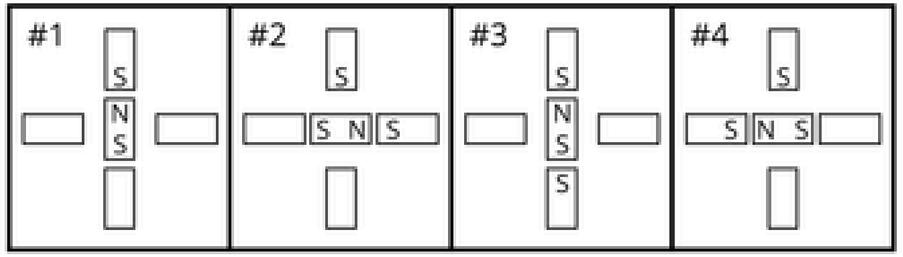
- Derivação ligada ao vcc
- controle entre "a" e "b" através de um driver



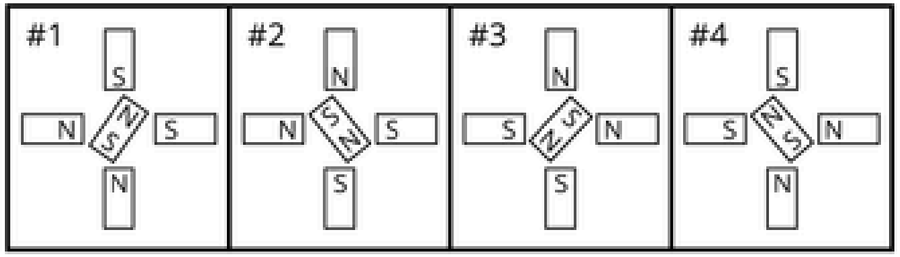
BIPOLAR

- Sem derivação central
- a cada 2 passos o polo deve ser invertido
- as duas bobinas devem ser energizadas juntas

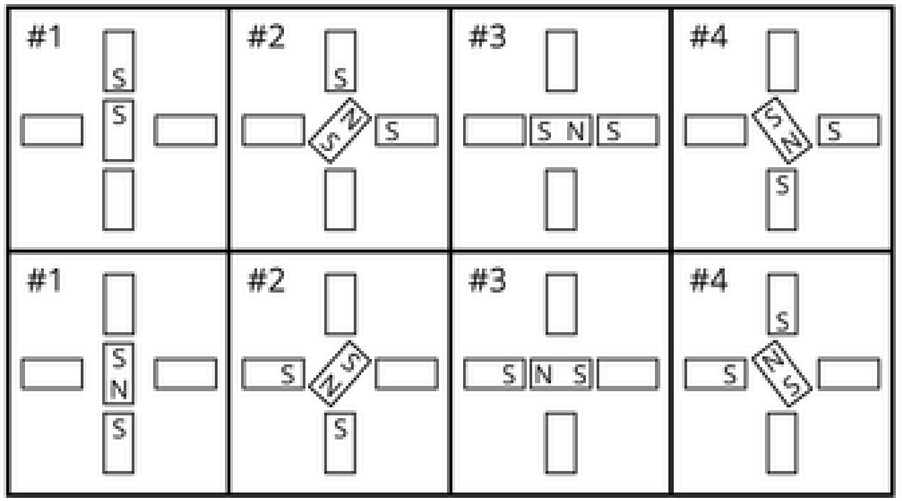
MOTOR DE PASSO
BIPOLAR E UNIPOLAR



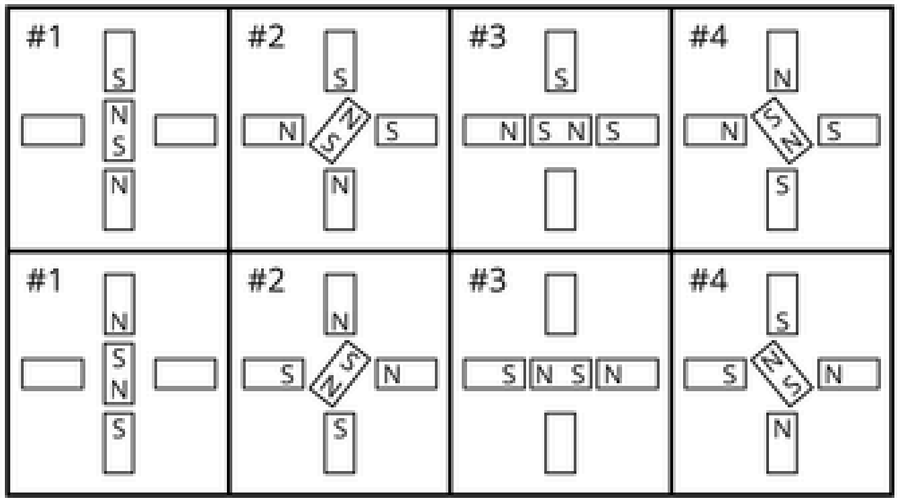
a Motor unipolar de passo inteiro



b Motor bipolar de passo inteiro



c Motor unipolar de médio passo



d Motor bipolar de médio passo

TABELA VERDADE

Motor	Ligação
4 Fios	Bipolar
5 Fios	Unipolar
6 Fios	Unipolar/Bipolar(série)
7 Fios	Unipolar/Bipolar(série/paralelo)

passo	1a	2a	1b	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1

Unipolar – Passo completo
Torque normal

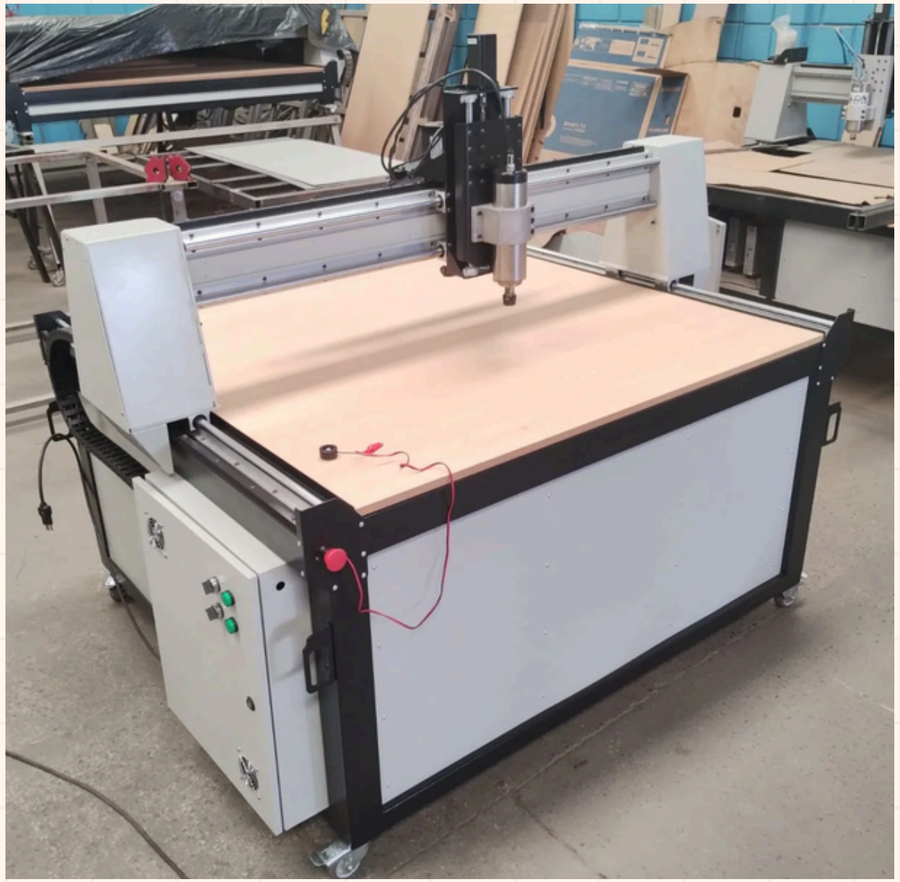
passo	1a	2a	1b	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

Unipolar – Passo completo
1,5x do Torque normal

passo	1a	2a	1b	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

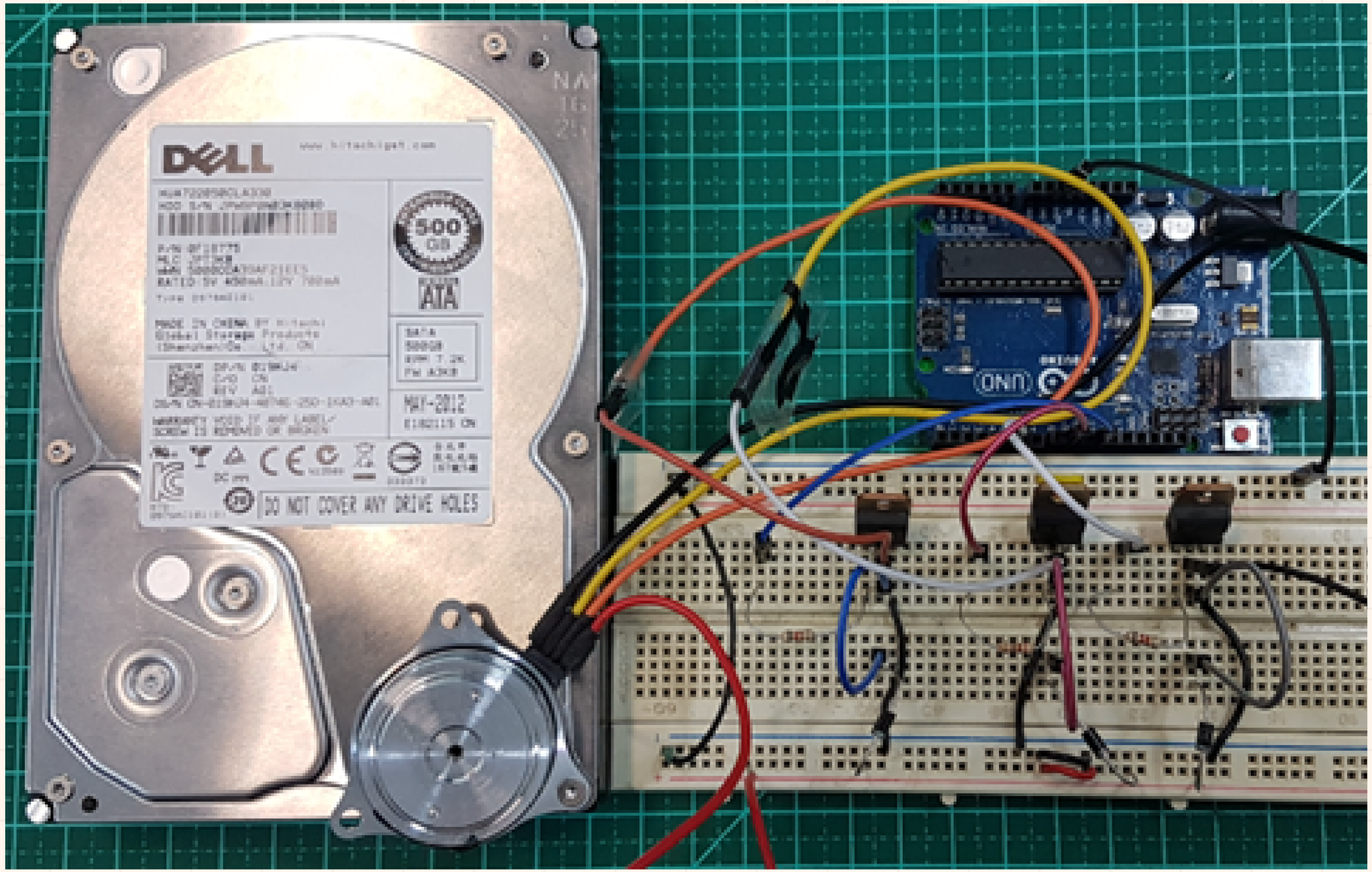
Unipolar – Meio-passo
Torque normal

APLICAÇÕES DE
MOTORES DE PASSO



APLICAÇÕES DE
MOTORES DE PASSO

APLICAÇÕES DE
MOTORES DE PASSO



MAS É POSSIVEL DIMINUIR
O PASSO SEM AUMENTAR
AS BOBINAS?

12V 3RPM
15Kgf.cm

50:1

100:1

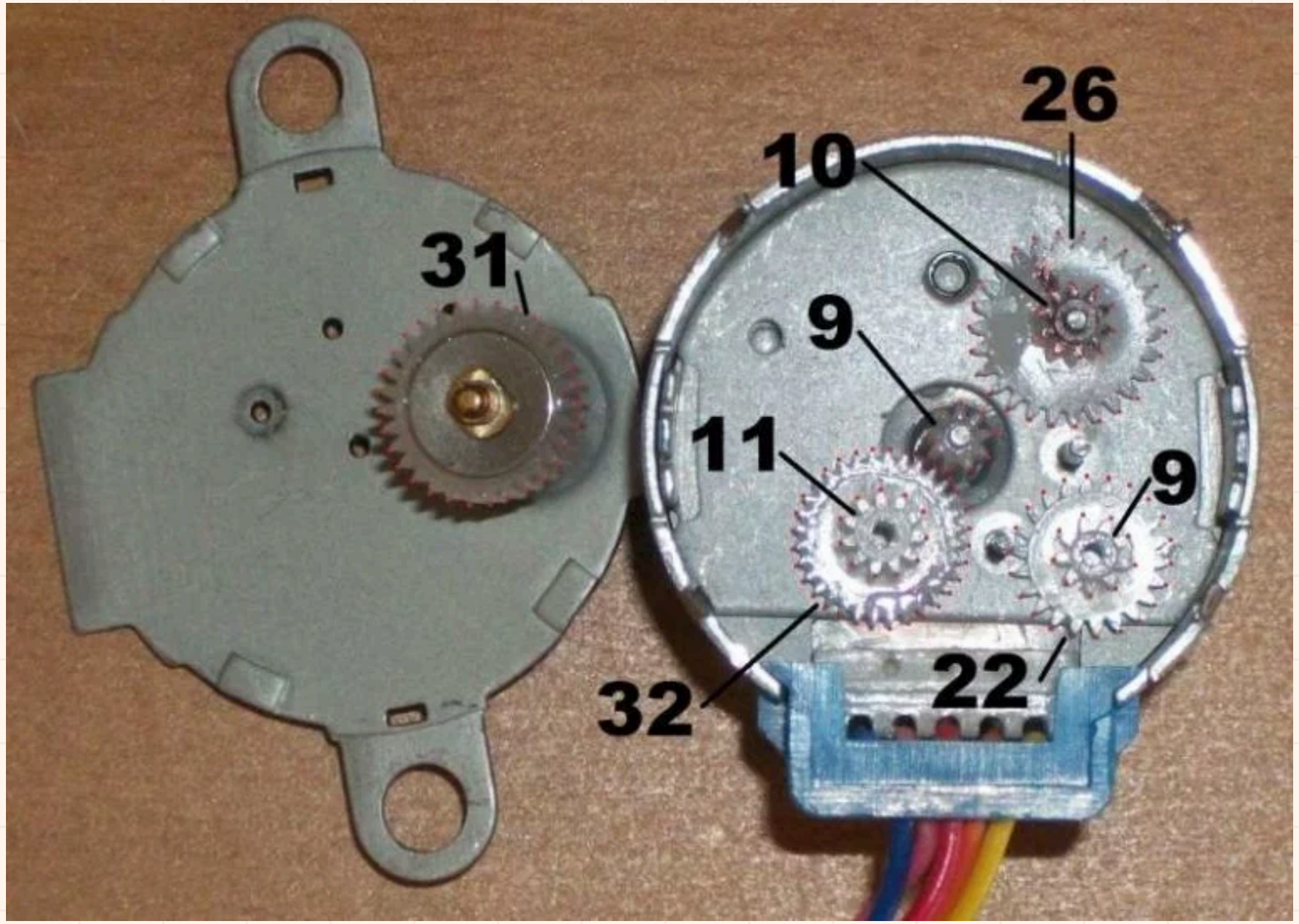
MOTOR DE PASSO
28BYJ-48

Como podem ver, a caixa tem várias engrenagens (31 dentes, 32 dentes, 26 dentes, 22 dentes, 11 dentes, 10 dentes, e mais duas com 9 dentes).

O pessoal fez alguns cálculos para determinar com maior precisão qual era a redução.

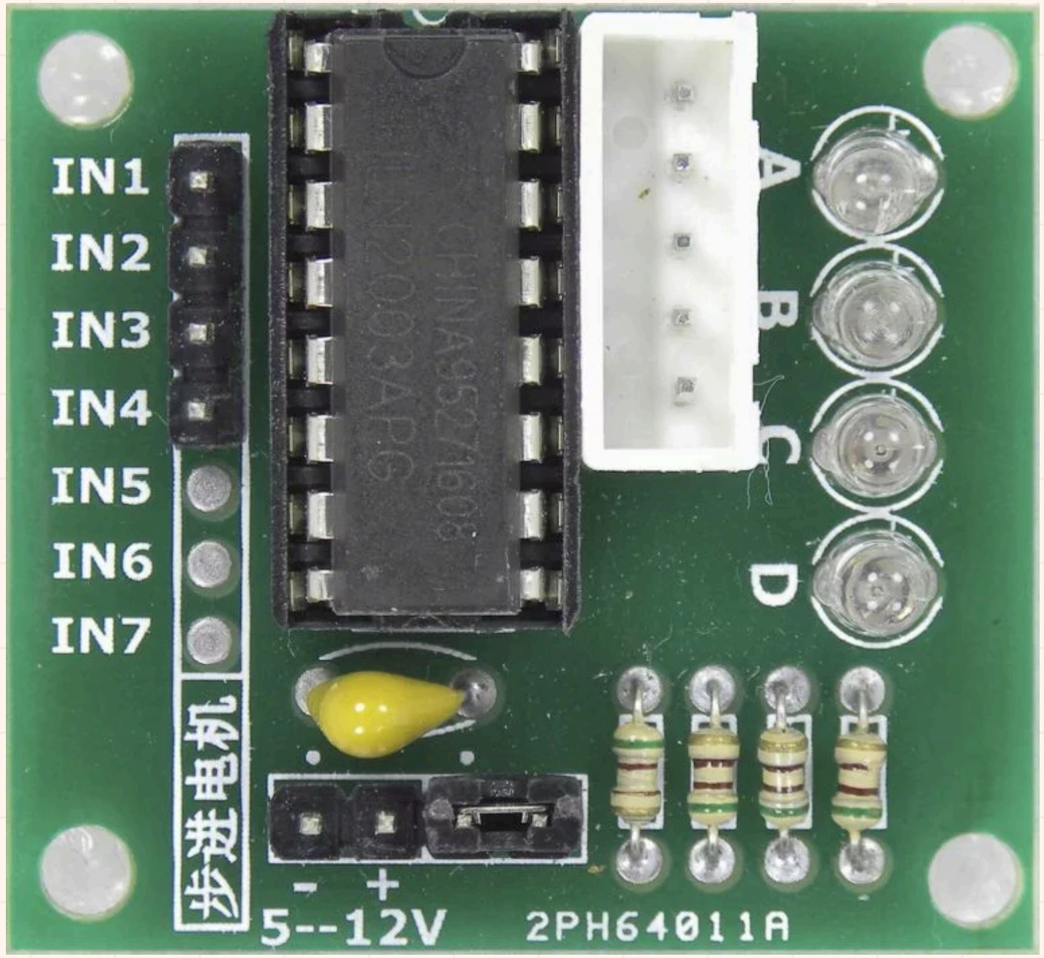
(31*32*26*22)/(11*10*9*9) = 283712/4455 = 25792/405 = 63,68395...

Na especificação do fabricante, ele aproximou o valor para 64

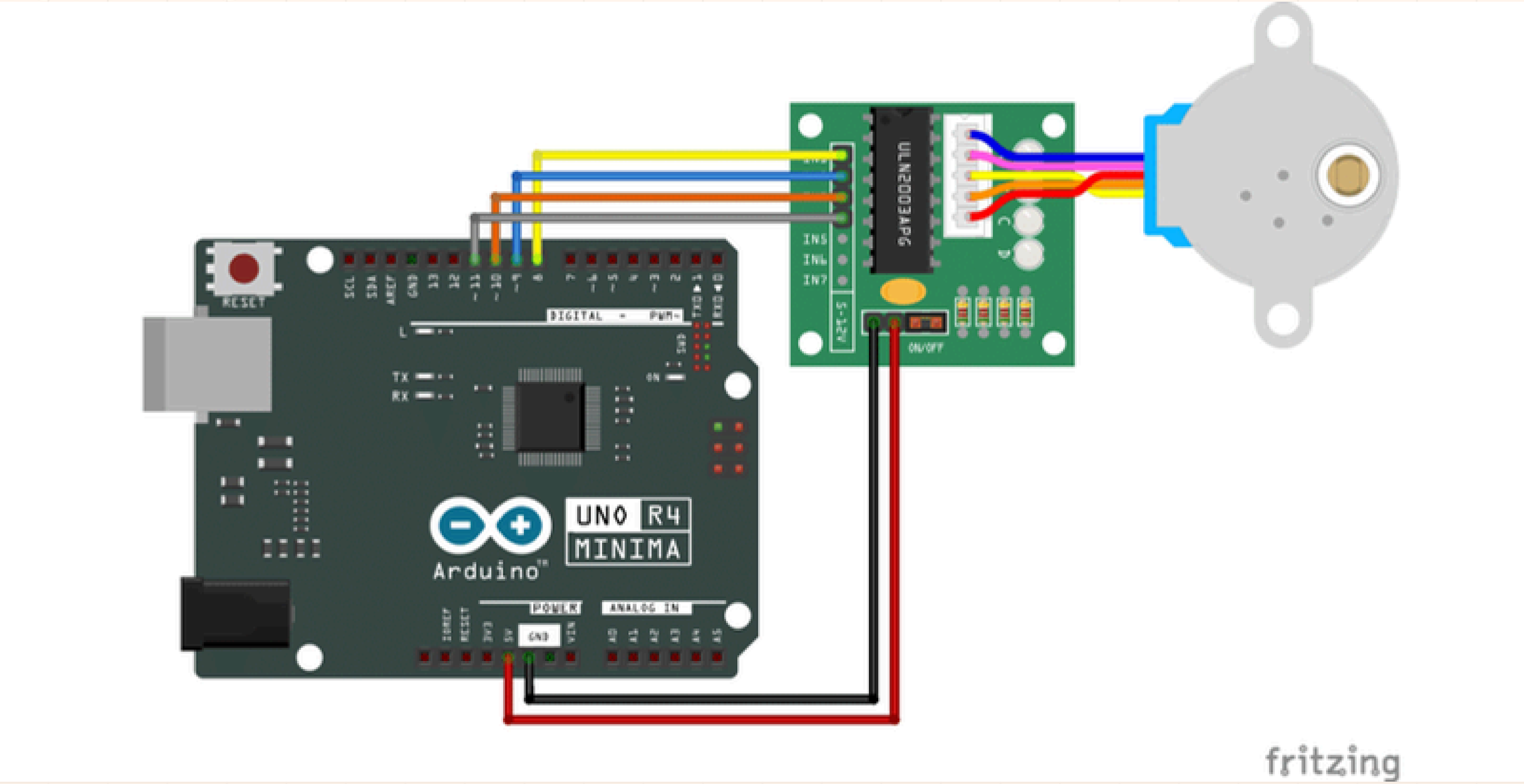


MOTOR DE PASSO
28BYJ-48

O pino mais à esquerda (-) no módulo é o terra (veja foto abaixo). Esse terra tem que ser conectado ao terra da fonte e ao terra do Arduino. O pino (+) no caso do Motor 28BYJ-48, tem que ser conectado ao positivo de uma fonte de 5V (preferencialmente de 1 Ampére). Não recomendo que conecte no 5V do Arduino, pois poderá sobrecarregar o regulador de tensão do mesmo.



MOTOR DE PASSO
MONTAGEM



MOTOR DE PASSO
CODIGO

```
#include <Stepper.h>

int passosPorVolta = 32;

Stepper mp(passosPorVolta, 8, 9, 10, 11);

void setup() {
  mp.setSpeed(500);
  Serial.begin(115200);
}

void loop() {
  mp.step(2048);
  delay(2000);
  mp.step(-2048);
  delay(2000);
}
```

MOTOR DE PASSO
CODIGO

```
#include <AccelStepper.h>

#define MotorIN1 D1
#define MotorIN2 D2
#define MotorIN3 D5
#define MotorIN4 D6

AccelStepper motor(AccelStepper::HALF4WIRE, MotorIN1, MotorIN3, MotorIN2,
MotorIN4);

void setup() {
  Serial.begin(115200);
  motor.setMaxSpeed(1000);
  motor.setSpeed(200);
}

void loop() {
  motor.runSpeed();
}
```

QUESTIONÁRIO DA OFICINA:

HTTPS://DOCS.GOOGLE.COM/FORMS/D/E/1FAIPQLSEKDGP7MHMJ6ZAUX88LWZMQOQE8_QGER3KMPGDDSBXVVFMFZQ/VIEWFORM

REFERENCIAS

https://www.mundodaeletrica.com.br/motor-dc-caracteristicas-funcionamento

https://www.mundodaeletrica.com.br/motor-dc-caracteristicas-funcionamento

https://www.newtoncbraga.com.br/robotica-e-mecatronica/5174-mec104.html

https://www.makerhero.com/blog/motor-dc-arduino-ponte-h-l298n/?srsltid=AfmBOoos9fpfDuFGu1Ng8uCfpu_unYLNGP7EBX3zlWZq48L7EzI359aU

https://www.eletrogate.com/mini-ponte-h-dupla-l298n?srsltid=AfmBOop_A26osgrYCamy-zyqYf0_k6UBQOehPGz9jI04lNGg1DQosf8_

https://www.eletrogate.com/mini-ponte-h-dupla-l298n?srsltid=AfmBOop_A26osgrYCamy-zyqYf0_k6UBQOehPGz9jI04lNGg1DQosf8_

https://www.youtube.com/watch?v=ekLcxNzjcsA

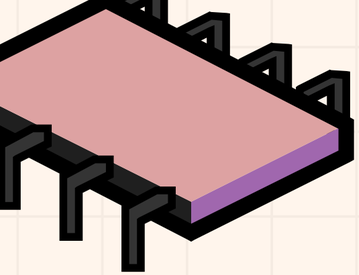
https://www.youtube.com/watch?v=qzerOP9rGOI&t=128s

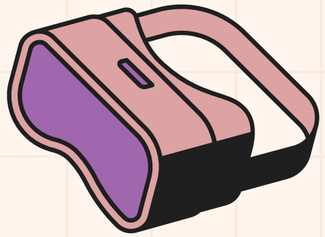
https://bluetechs.wordpress.com/wp-content/uploads/2013/05/step_motor_control_with_arduino.pdf

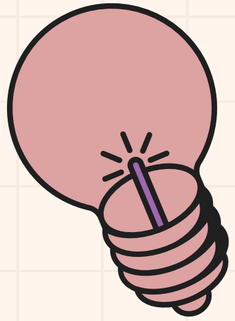
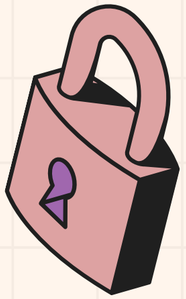
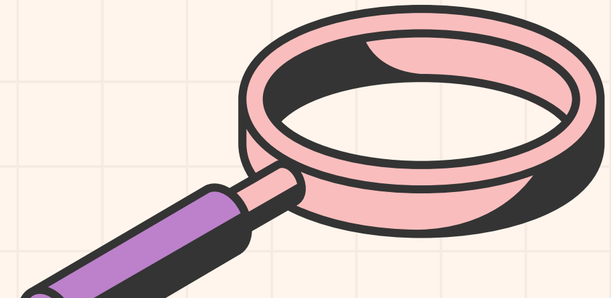
https://materialpublic.imd.ufrn.br/curso/disciplina/1/58/2/16

https://www.researchgate.net/figure/Figura-1-Esquema-de-ligacao-das-bobinas-dos-motores-de-passo-A-Motor-Unipolar-possui_fig1_332512199

https://blog.eletrogate.com/guia-completo-do-motor-de-passo-28byj-48-driver-uln2003







OBRIGADO