

# Motor de passos 28BYJ-48 e driver ULN2003

---

Prof. Me. Hélio Esperidião

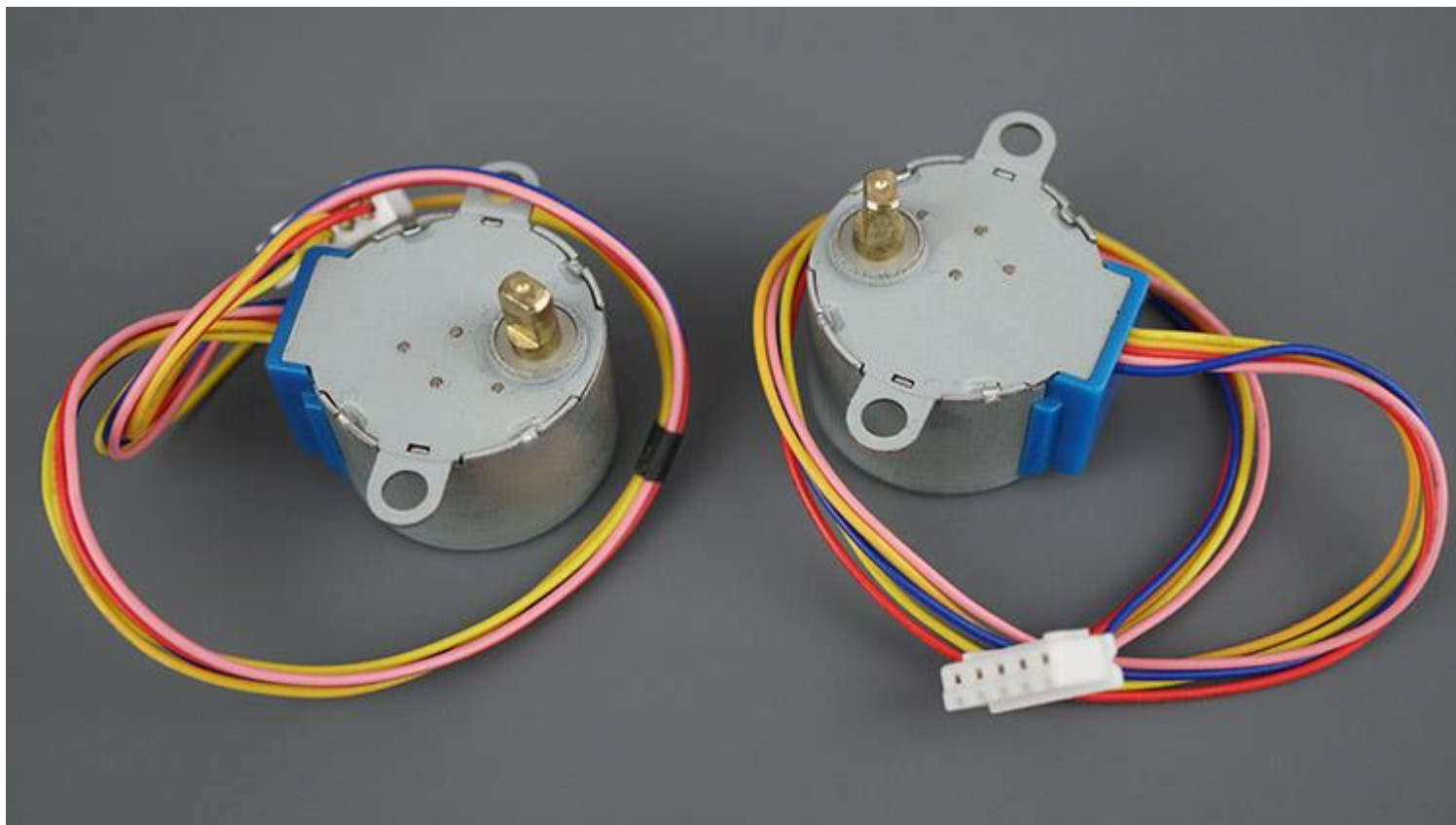
# Motores de Passo

- Um motor de passo é um motor elétrico DC sem escovas que divide uma rotação completa em um número de passos.
- Ele se move um passo de cada vez, sendo que cada passo é do mesmo tamanho.
- Isso nos permite girar o motor por um ângulo preciso até uma posição precisa.
- O motor de passo pode girar no sentido horário ou anti-horário.

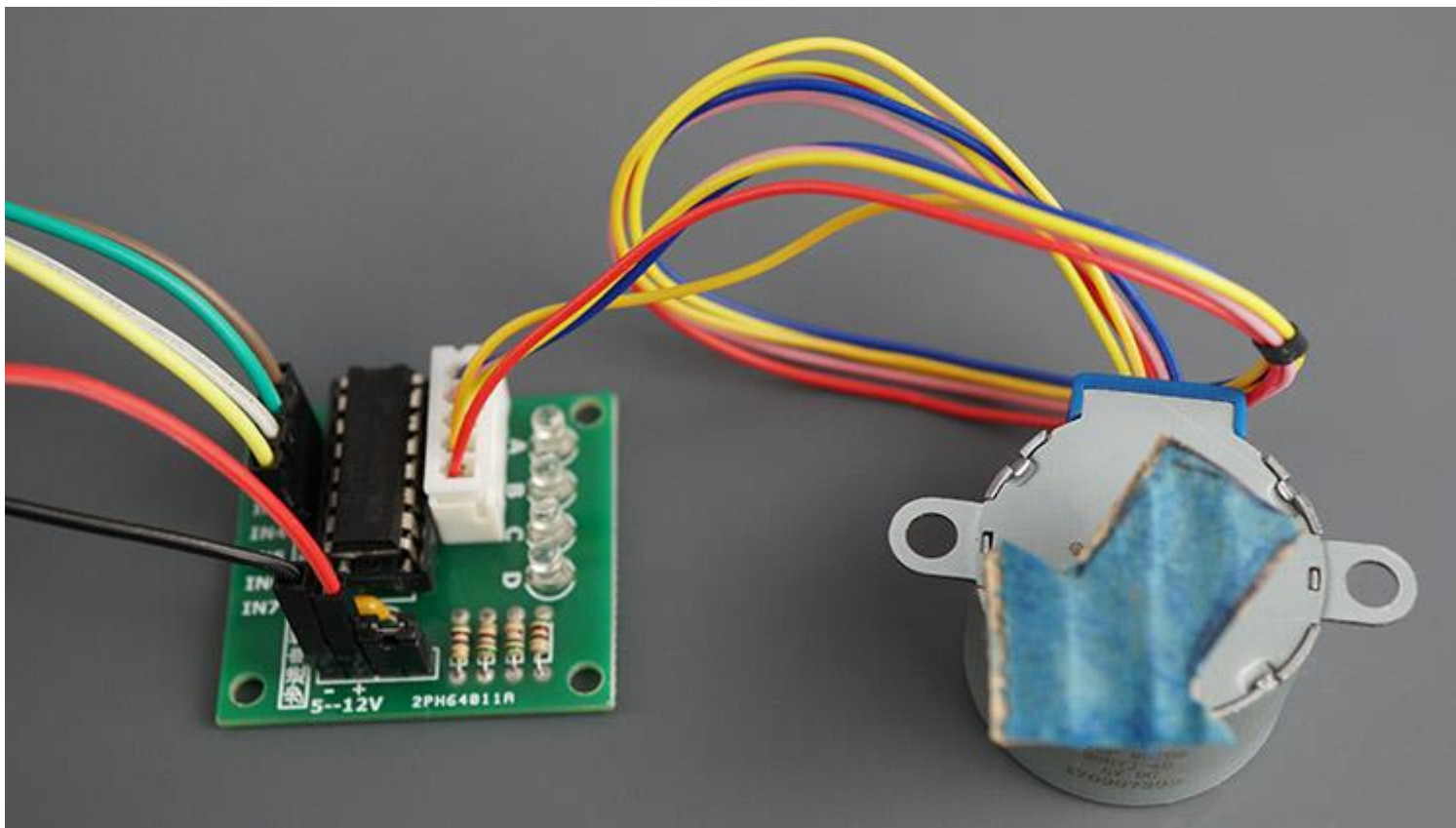
# funcionamento

- Os motores de passo são compostos por bobinas internas que fazem o eixo do motor mover-se em passos em uma direção ou outra quando corrente é aplicada às bobinas de forma específica.
- Existem dois tipos de motores de passo: motores de passo unipolares e bipolares.

# 28BYJ-48



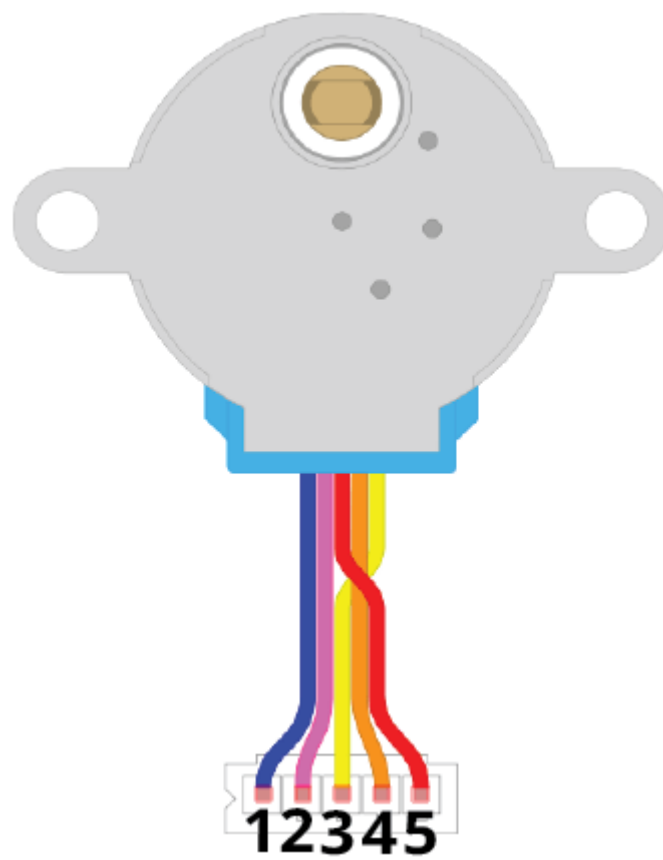
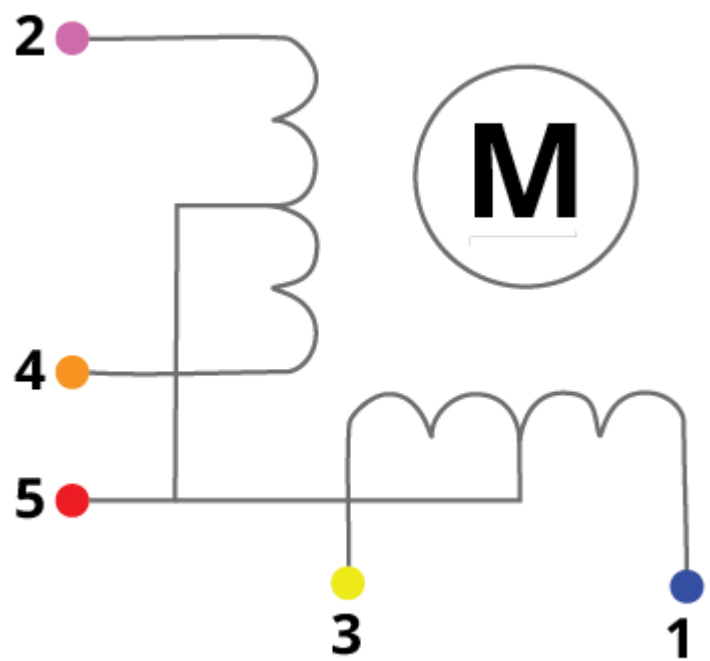
# 28BYJ-48



# Características: 28BYJ-48

- Tensão nominal: 5V DC
  - A "tensão nominal" refere-se à voltagem esperada para operar o motor de passo de forma eficiente
- Número de fases: 4
  - Número de fases indica quantos conjuntos de bobinas o motor possui, o que está relacionado à sua capacidade de realizar movimentos precisos.
- Razão de variação de velocidade: 1/64
  - isso significa que a velocidade do motor pode ser variada em até 1/64 da velocidade máxima do motor
- Ângulo de passo:  $5.625^{\circ}/64$ 
  - é a medida angular entre cada passo do motor, indicando a precisão do movimento que ele pode realizar
- Frequência: 100Hz
  - axa de repetição dos pulsos elétricos que controlam o motor, influenciando na velocidade máxima que ele pode atingir

- O motor de passo 28BYJ-48 possui um total de quatro bobinas. Uma extremidade das bobinas está conectada a 5V, correspondendo ao fio vermelho do motor.
- A outra extremidade das bobinas corresponde aos fios de cores azul, rosa, amarelo e laranja.
- Energizar as bobinas em uma sequência lógica faz o motor mover-se um passo em uma direção ou na outra.





# Características

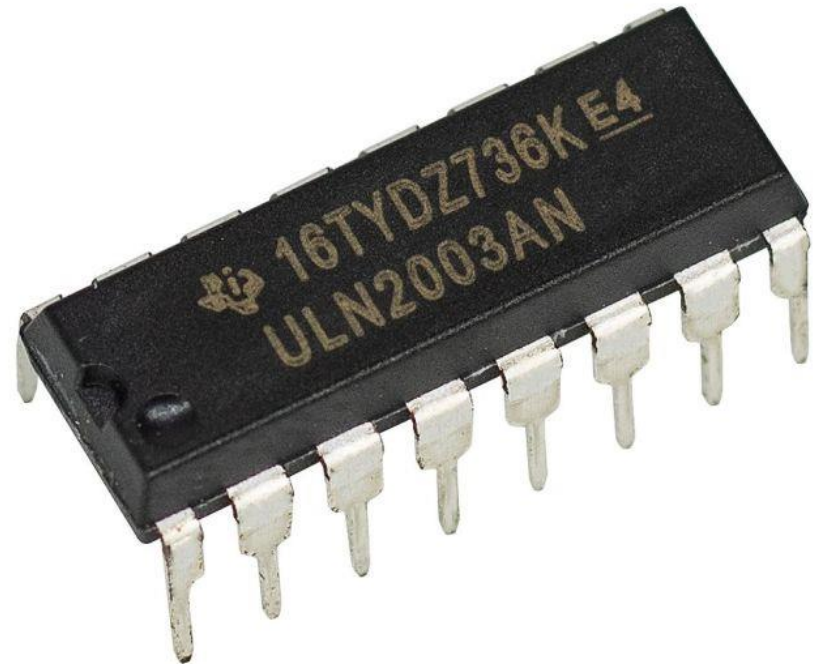
- O Motor de Passo 28BYJ-48 tem um ângulo de passo de  $5.625^\circ/64$  no modo de meio passo.
- Isso significa que o motor tem um ângulo de passo de  $5.625^\circ$ , então ele precisa de  $360^\circ/5.625^\circ = 64$  passos no modo de meio passo.
- No modo de passo completo:  $64/2 = 32$  passos para completar uma rotação.
- O eixo de saída é acionado por meio de uma relação de engrenagens de 64:1.
  - O eixo (visível fora do motor) completará uma rotação se o motor dentro girar 64 vezes.
  - O motor terá que realizar  $32 \times 64 = 2048$  passos para que o eixo complete uma rotação completa.
  - Portanto você terá uma precisão de  $360^\circ/2048$  passos =  $0.18^\circ/\text{step}$ .

# resumo

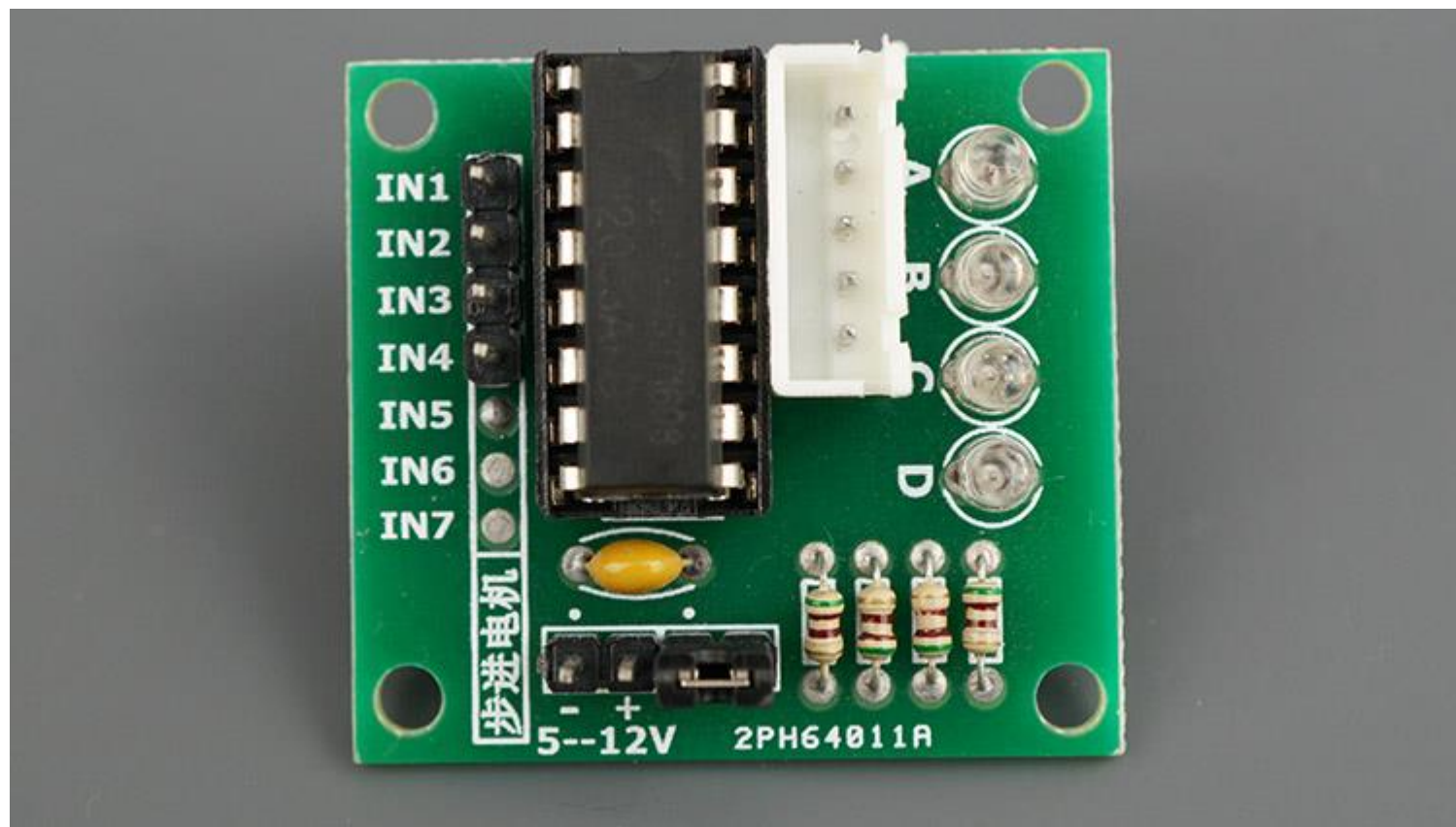
- Passos totais por revolução = 2048 passos
- Ângulo do passo =  $0.18^\circ/\text{step}$
- Se estiver usando um motor de passo diferente, consulte o datasheet.

# ULN2003

- Para conectar o motor de passo ao ESP32, usaremos o driver de motor ULN2003.
- O motor de passo 28BYJ-48 é frequentemente vendido junto com o driver de motor ULN2003.

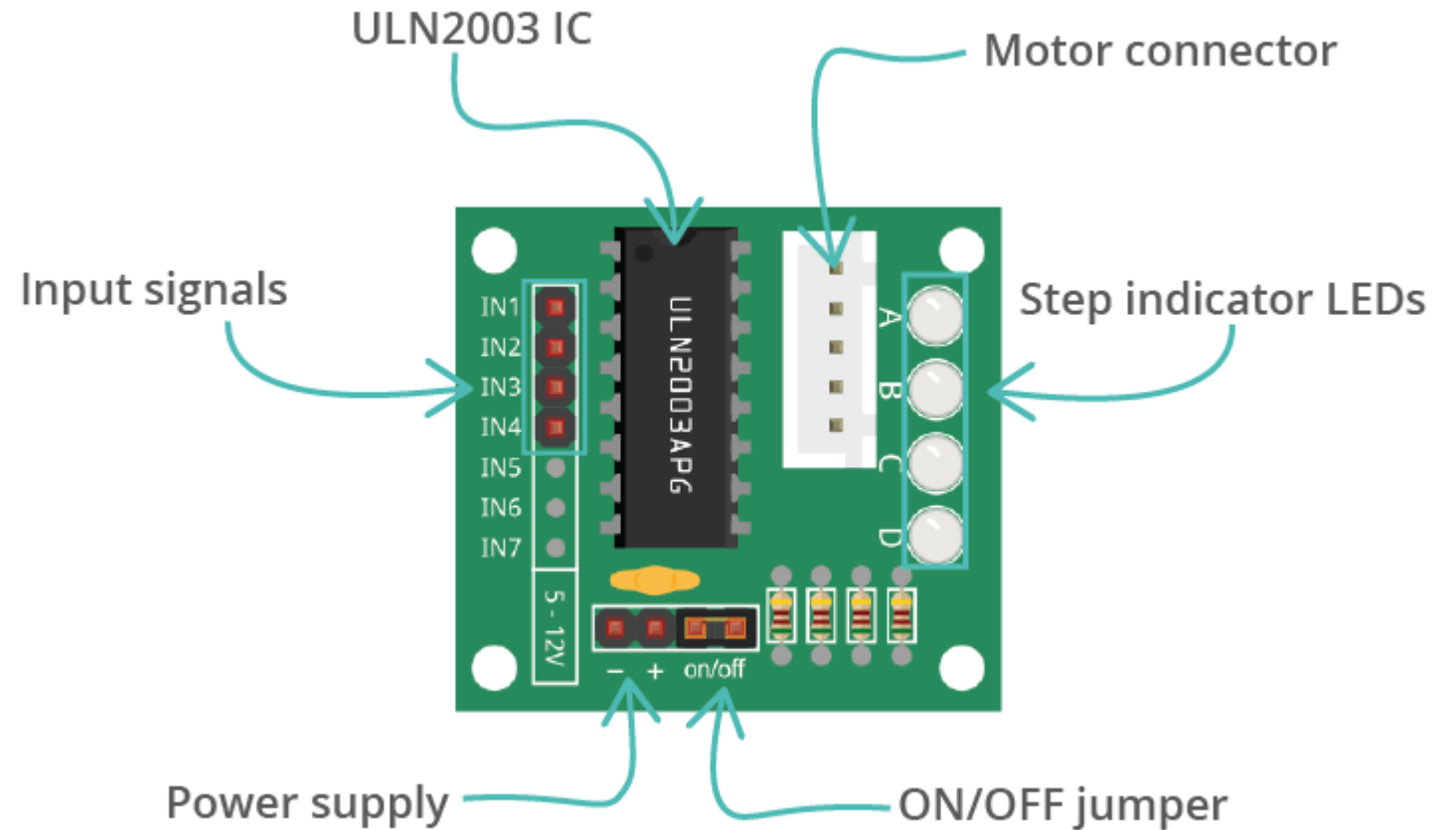


kit



# Esquema

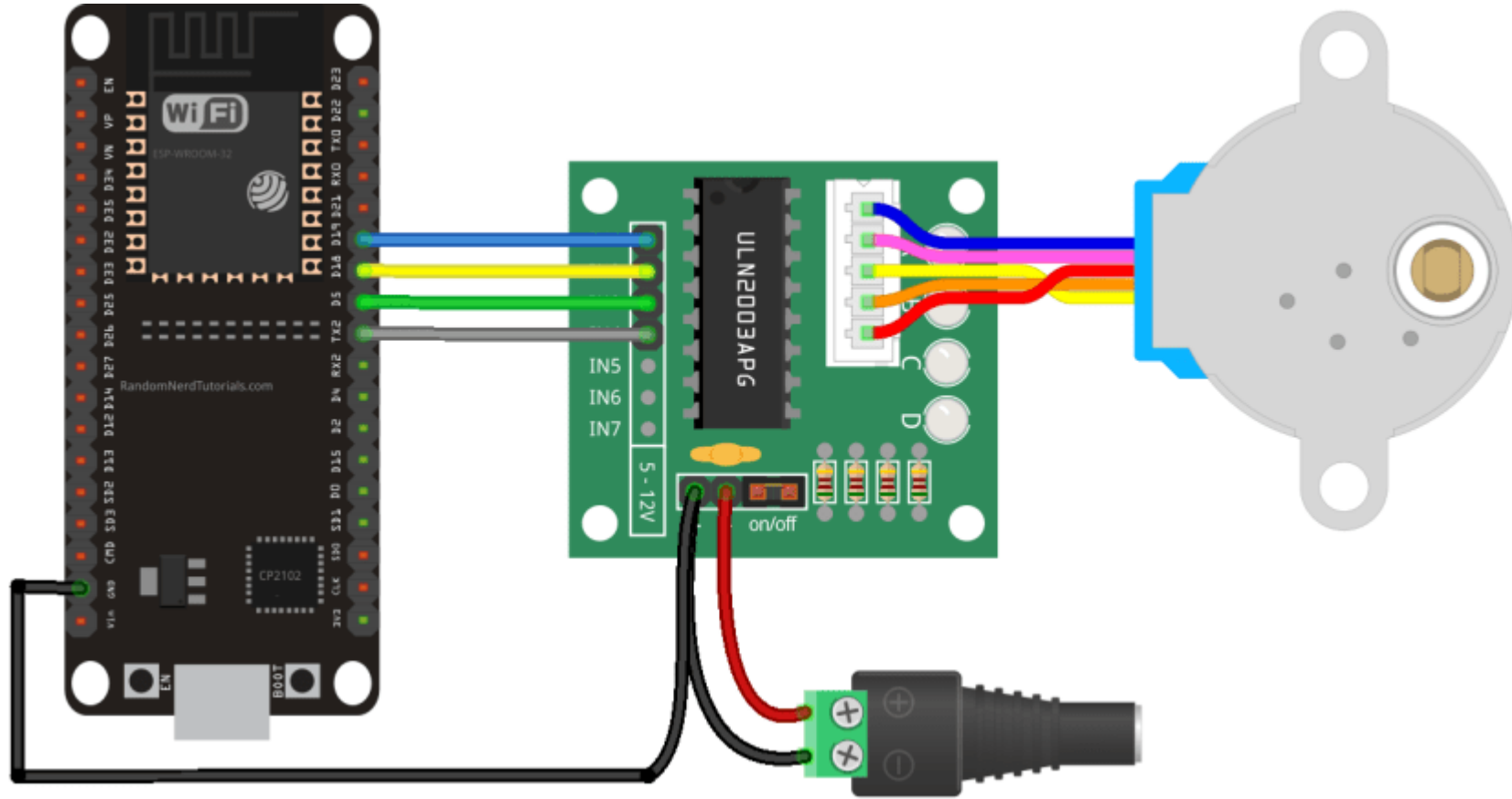
- Se você remover o jumper, não haverá energia chegando ao motor. Você pode usar esses pinos para conectar um interruptor físico.



# Pinout ULN2003

- IN1, IN2, IN3 e IN4
  - Para controlar o motor, você pode conectar um pino digital do microcontrolador ao ULN2003. Isso permite que você envie sinais de controle ao motor, como passos e direção, para controlar seu movimento. Cada pino se refere a uma bobina

esquema



# Ligação com o ESP32

## Motor Driver

IN1

IN2

IN3

IN4

## ESP32

GPIO 19

GPIO 18

GPIO 5

GPIO 17



# Codificação

```
#include <Stepper.h>
```

```
const int stepsPerRevolution = 2048;
```

```
#define IN1 19
```

```
#define IN2 18
```

```
#define IN3 5
```

```
#define IN4 17
```

```
Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);
```

# Setup()

```
void setup() {  
    myStepper.setSpeed(5);  
    Serial.begin(115200);  
}
```

# Loop()

```
void loop() {  
  Serial.println("horario");  
  myStepper.step(stepsPerRevolution);  
  delay(1000);  
  
  Serial.println("antihorario");  
  myStepper.step(-stepsPerRevolution);  
  delay(1000);  
}
```