

# Servo Motor

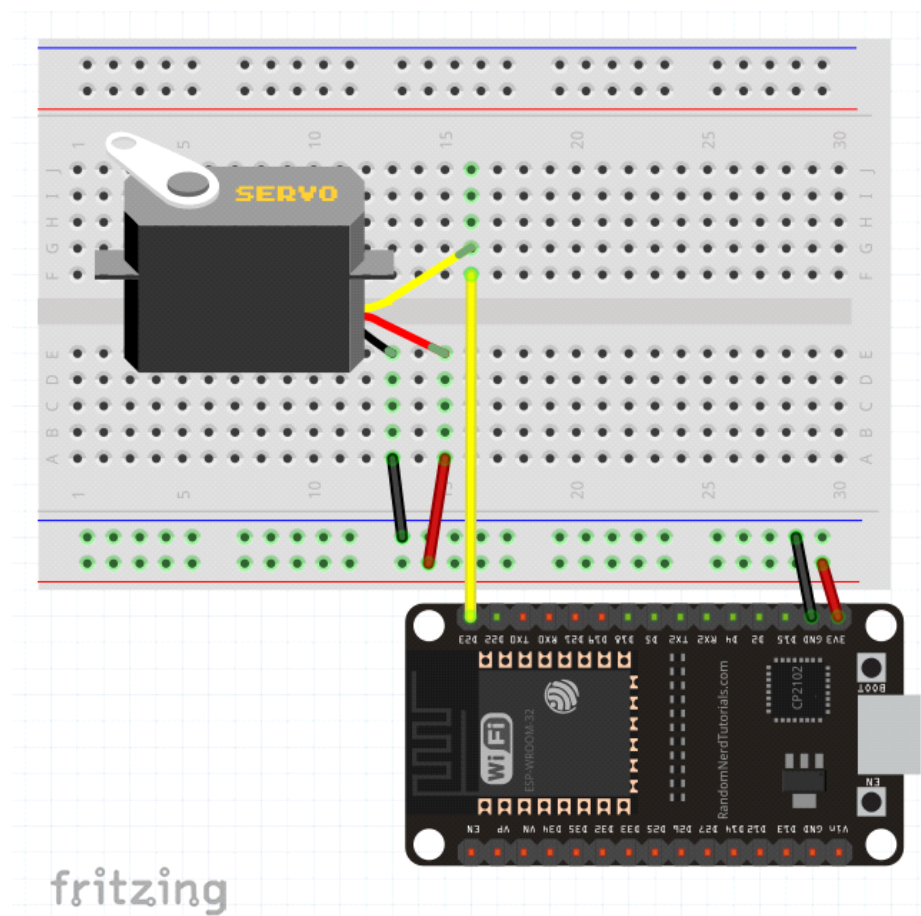
Um servo motor é um pequeno motor de corrente contínua. Diferentemente dos demais motores DC, onde apenas controlamos a velocidade de giro, conseguimos controlar a posição de um servo, através do PWM.

O servo que vamos utilizar nesse exemplo, tem a resolução de meia volta, ou seja, de 0° a 180°.

Para essa aula, você vai precisar de...

- 1 Placa Franzininho WiFi com CircuitPython;
- 1 Micro servo 9g;
- 1 Protoboard;
- 1 potenciômetro de 10 k $\Omega$ ;
- Jumpers.

Com os materiais em mãos, realize a montagem a seguir...



## Código

Nesse programa, vamos aprender a como usar um servo motor e posicioná-lo no ângulo que desejamos. Para isso, escreva o código a seguir:

```
1 from machine import Pin, PWM
2 from time import sleep
3 import math
4
5 def converter(x, in_min,in_max,out_min,out_max):
6     conta = math.trunc((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)
7     return conta
8
9 def angulo(x):
10     angulo = converter(x, 0,180,20,120)
11     servo.duty(angulo)
12
13 servo = PWM(Pin(23), freq = 50) #frequência de servomecanismo
14
15 while True:
16     angulo(0)
17     sleep(2)
18     angulo(45)
19     sleep(2)
20     angulo(90)
21     sleep(2)
22     angulo(135)
23     sleep(2)
24     angulo(180)
25     sleep(2)
```

Agora, execute-o!

## Explicação

Começaremos impontando as funções Pin e PWM do módulo machine, que nos permitira acessar os pinos da placa e configurá-los:

```
1 from machine import Pin, PWM
```

Também, necessitaremos da função sleep da biblioteca time que, utilizaremos como para intervalos de temporização:

```
2 from time import sleep
```

E, a biblioteca nativa do Python, math. Que traz diversas funções matemáticas e usaremos a de arredondamento:

```
3 import math
```

Agora, criaremos uma função chamada converter onde necessitaremos de 5 parâmetros: valor que desejamos converter, entrada mínima, entrada máxima, saída mínima e saída máxima. Dentro da função, vamos realizar o cálculo de conversão de valores, armazenar o valor na variável conta e retornar esse valor:

```
5 def converter(x, in_min,in_max,out_min,out_max):
6     conta = math.trunc((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)
7     return conta
```

A segunda função, criaremos com o intuito de enviar o ângulo desejado para o servo motor. Talvez, você esteja se perguntado o porque não enviar o valor direto para o servo. Isso ocorre, porque o intervalo de valores do servo (confirme não documentação), não está de acordo com os ângulos. No servo utilizado no experimento, o intervalo vai de 20 a 120. Essa função, pede apenas um parâmetro, o ângulo que desejamos posicionar o nosso servo motor. Dentro de ângulo, chamaremos converter e enviaremos esse valor para o servo:

```
9 def angulo(x):  
10     angulo = converter(x, 0,180,20,120)  
11     servo.duty(angulo)
```

Configuraremos o pino como uma saída PWM, nota que definiremos uma frequência de 50Hz nesse caso, pois é a frequência padrão de servomecanismo:

```
13 servo = PWM(Pin(23), freq = 50) #frequência de servomecanismo
```

Por fim, vamos para o laço de repetição infinita. Onde a cada dois segundos, trocaremos a posição do servo:

```
15 while True:  
16     angulo(0)  
17     sleep(2)  
18     angulo(45)  
19     sleep(2)  
20     angulo(90)  
21     sleep(2)  
22     angulo(135)  
23     sleep(2)  
24     angulo(180)  
25     sleep(2)
```

## Exercícios

- 1) Assim como fizemos no ultimo capítulo com a intensidade luminosa do led, movimente o servo de 0° a 180° e depois decrescente o valor.
- 2) Acrescente um potenciômetro a sua montagem e, controle o ângulo do servo de acordo com a posição do potenciômetro.

