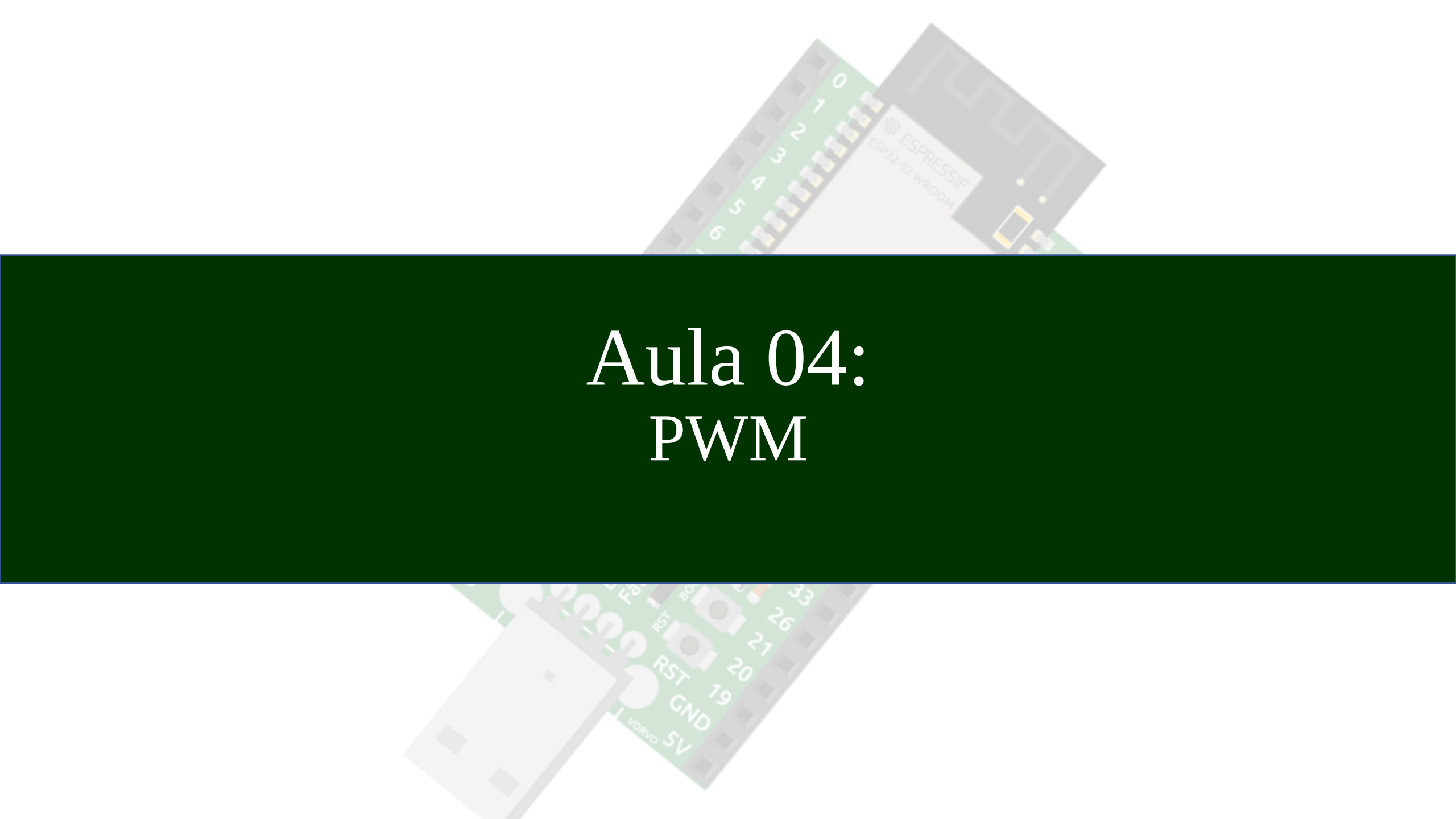
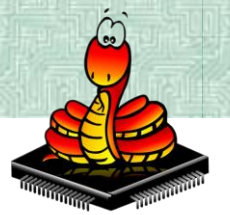
Franzininho
Na
FATEC

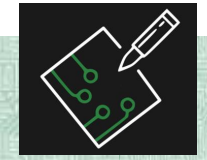


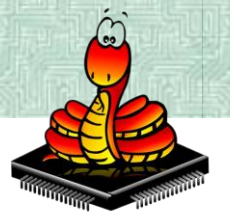
Aula 04:
PWM



Estrutura de Repetição FOR

- Em algumas situações é comum que uma mesma instrução (ou conjunto delas) precise ser executada várias vezes seguidas. Nesses casos, normalmente utilizamos um loop (ou laço de repetição) que permite executar o mesmo bloco de código enquanto uma condição é atendida.

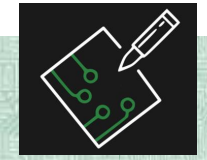


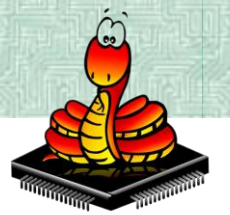


Estrutura de Repetição FOR

- O laço **for** nos permite percorrer os itens de uma coleção e, para cada um deles, executar o bloco de código declarado no loop. Sua sintaxe é a seguinte:

```
1 for variavel in lista
2 comandos
```

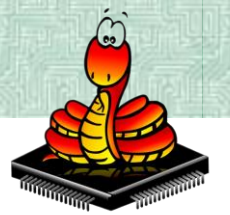




Funçaõ Range()

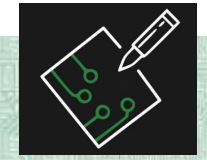
- A função range() retorna uma série numérica no intervalo enviado como argumento. A série retornada é um objeto iterável tipo range e os elementos contidos serão gerados sob demanda.

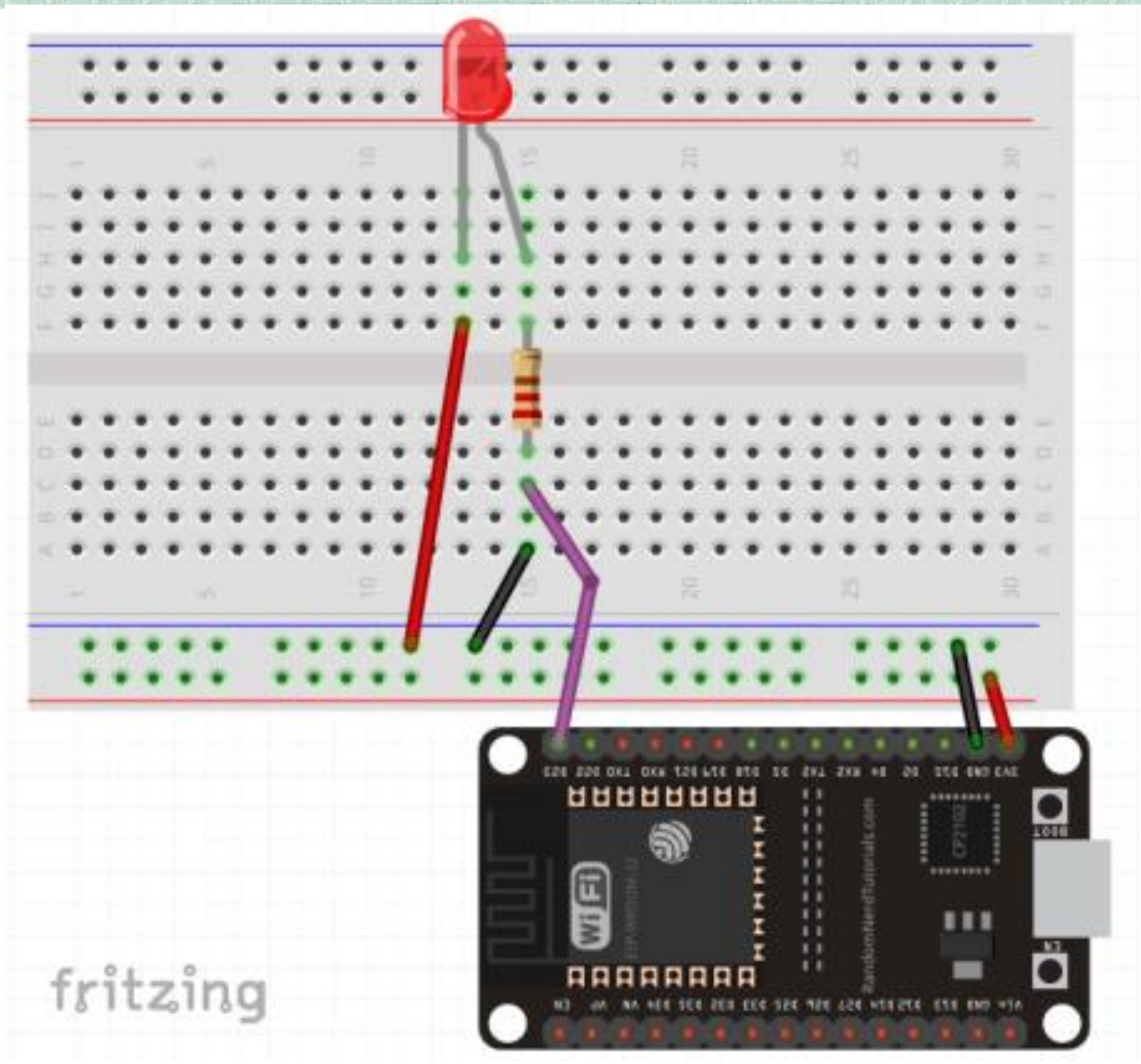
- É comum o uso da função range() com a estrutura for loop. Desta formta temos que a cada ciclo o próximo elemento da sequência será utilizado de tal forma que é possível partirmos de um ponto e ir incrementando, decrementando x unidades.

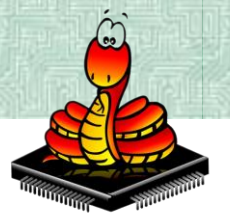


PWM – Pulse Width Modulation

- Essa é uma técnica onde, uma saída digital é usada para simular uma saída analógica. Isso acontece por meio de mudanças constantes de sinal, que são imperceptíveis e, dessa maneira o componente conectado a esse pino apenas tira uma média de quanto tempo está em estado alto e baixo.



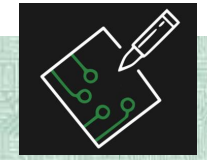

fritzing

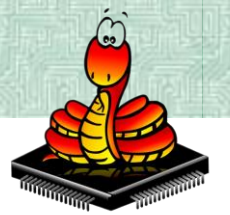


PWM – Pulse Width Modulation

- Para iniciar, vamos importar as funções Pin e PWM do módulo machine:
- Também, precisaremos a função sleep para intervalos de temporização:

```
3  from machine import Pin, ADC, PWM
4  from time import sleep
5
6  led = PWM(Pin(23))
```





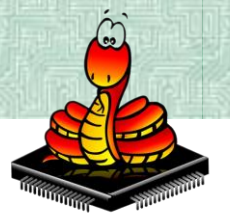
PWM – Pulse Width Modulation

- Agora, iremos para o nosso while True. Aqui, usaremos as funções for, uma estrutura de repetição e a função range, que gera um intervalo de valores.

PWM – Pulse Width Modulation

- No primeiro laço de repetição, aumentaremos gradativamente a luminosidade do led. Para enviarmos um valor para a saída digital usamos a função led.duty( ) e, o valor enviado será i, a variável de contagem que estamos utilizando no laço for.

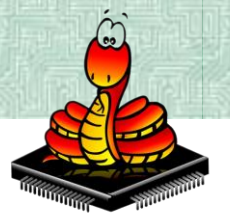
```
8   while True:
9       #for --> estrutura de repetição
10      #range --> intervalo de valores com icremento
11
12      for i in range(0, 1023, 5):
13          led.duty(i)  #enviar o valor para o LED
14          sleep(0.05)
```



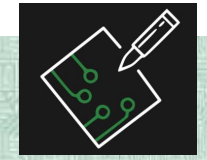
PWM – Pulse Width Modulation

- No segundo, vamos repetir o mesmo processo, porém, agora diminuiremos o brilho:

```
15
16      for i in range(1023,0, -5):
17          led.duty(i)  #enviar o valor para o LED
18          sleep(0.05)
```

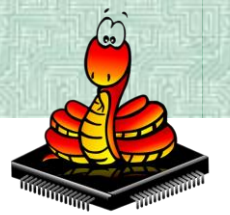


```python
#Aula 4 - PWM

from machine import Pin, ADC, PWM
from time import sleep

led = PWM(Pin(23))

while True:
    #for --> estrutura de repetição
    #range --> intervalo de valores com icremento

    for i in range(0, 1023, 5):
        led.duty(i)  #enviar o valor para o LED
        sleep(0.05)

    for i in range(1023,0, -5):
        led.duty(i)  #enviar o valor para o LED
        sleep(0.05)
```



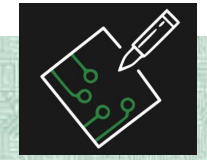
Led RGB

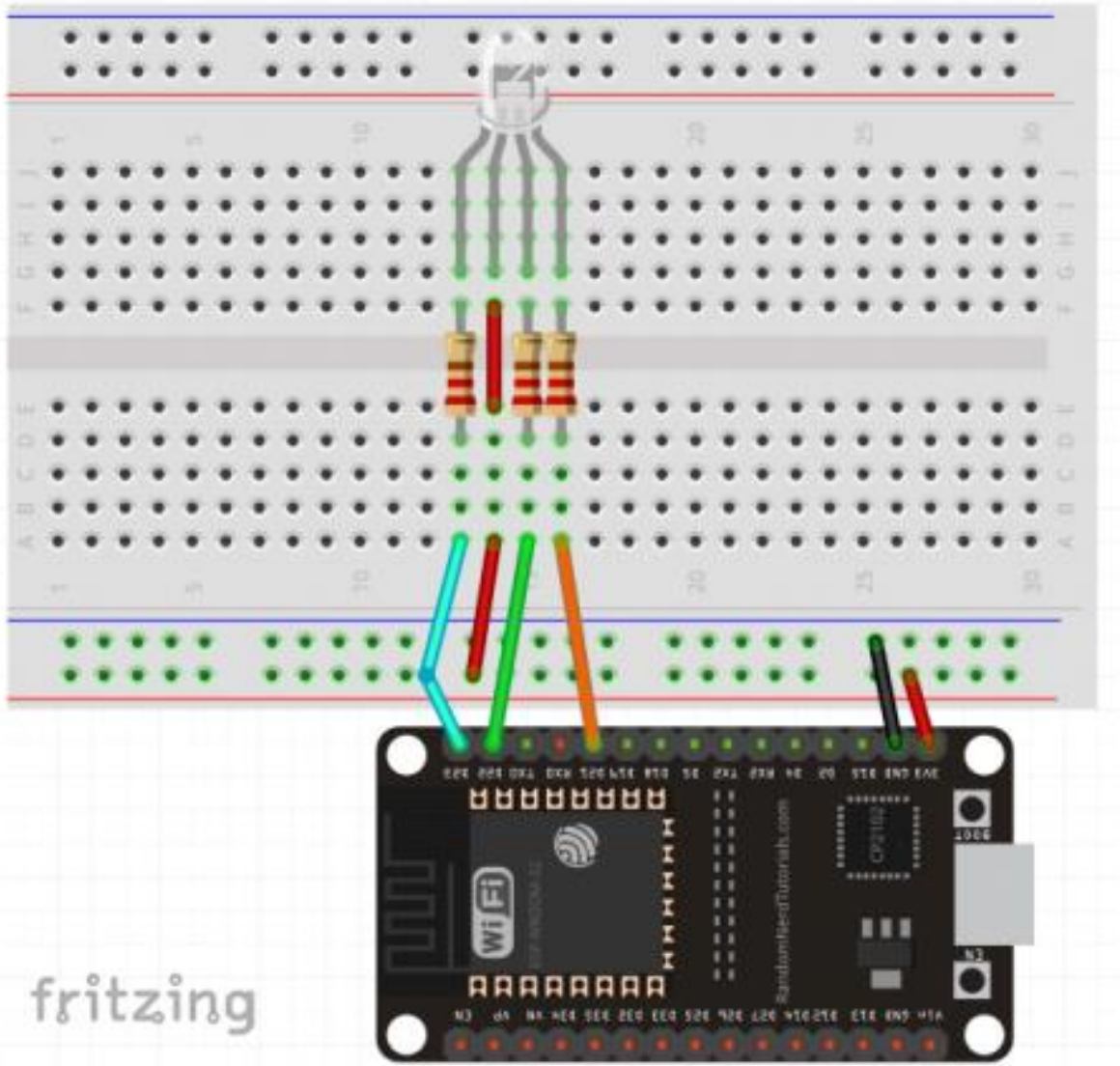
- Agora, vamos utilizar um led RGB. Esse, é um componente que contêm três LEDs encapsulados, sendo eles, vermelho, verde e azul. A partir dessas cores, podemos criar uma grande variedade de combinações. Além disso, ele funciona utilizado PWM.



Led RGB

- O LED RGB, pode ser de anodo comum (acende em nível baixo) ou catodo comum (acende em nível alto). Nesse caso, é um LED RGB anodo comum, por isso o conectamos ao 3V da placa.

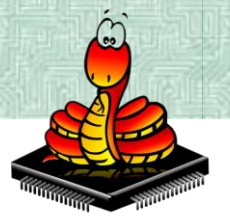



fritzing

Led RGB

- Primeiramente, vamos importar os módulos e funções já utilizados em aulas anteriores – machine, time e math:

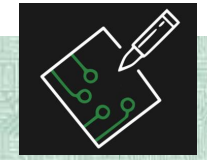
```
1   from machine import Pin, PWM
2   from time import sleep
3   import math
```

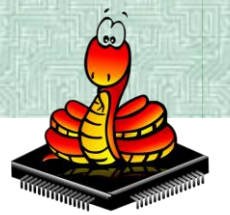


Led RGB

- Após isso, vamos configurar os pinos que utilizaremos. Repare, que assim como leds individuais cada led necessita de um pino:

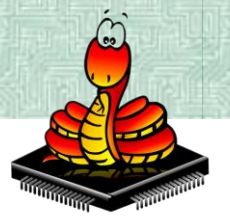
```
5   red = PWM(Pin(21))
6   green = PWM(Pin(22))
7   blue = PWM(Pin(23))
```



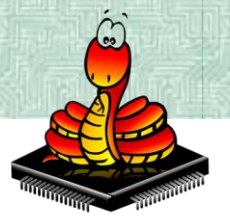


Led RGB

- Criaremos uma função converter, onde enviaremos dois parâmetros, primeiro o valor que desejamos converter se é um anodo ou um catodo comum
- Após isso, verificaremos qual opção foi escolhida.



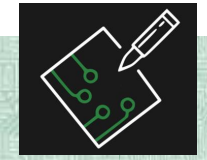
```
9   def converter(x, comum):
10      if comum.lower() == 'catodo':
11          in_min=0
12          in_max=255
13          out_min=0
14          out_max=1023
15      elif comum.lower() == 'anodo':
16          in_min=0
17          in_max=255
18          out_min=1023
19          out_max=0
20      conta = math.trunc((x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)
21      return conta
```

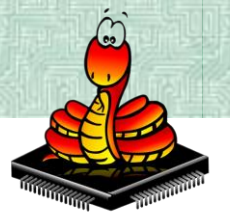


Led RGB

- Por conseguinte, criaremos uma função que facilitará o envio dos valores para o RGB.

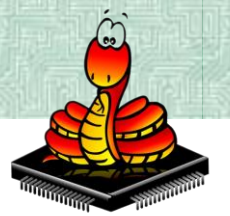
```
23 def rgb(r, g, b):
24     red.duty(converter(r, 'anodo'))
25     green.duty(converter(g, 'anodo'))
26     blue.duty(converter(b, 'anodo'))
```



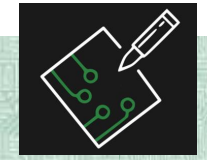


Led RGB

- Por fim, vamos criar o laço de repetição infinita. Aqui vamos char a função rgb e a cada 3 segundos trocar a cor:

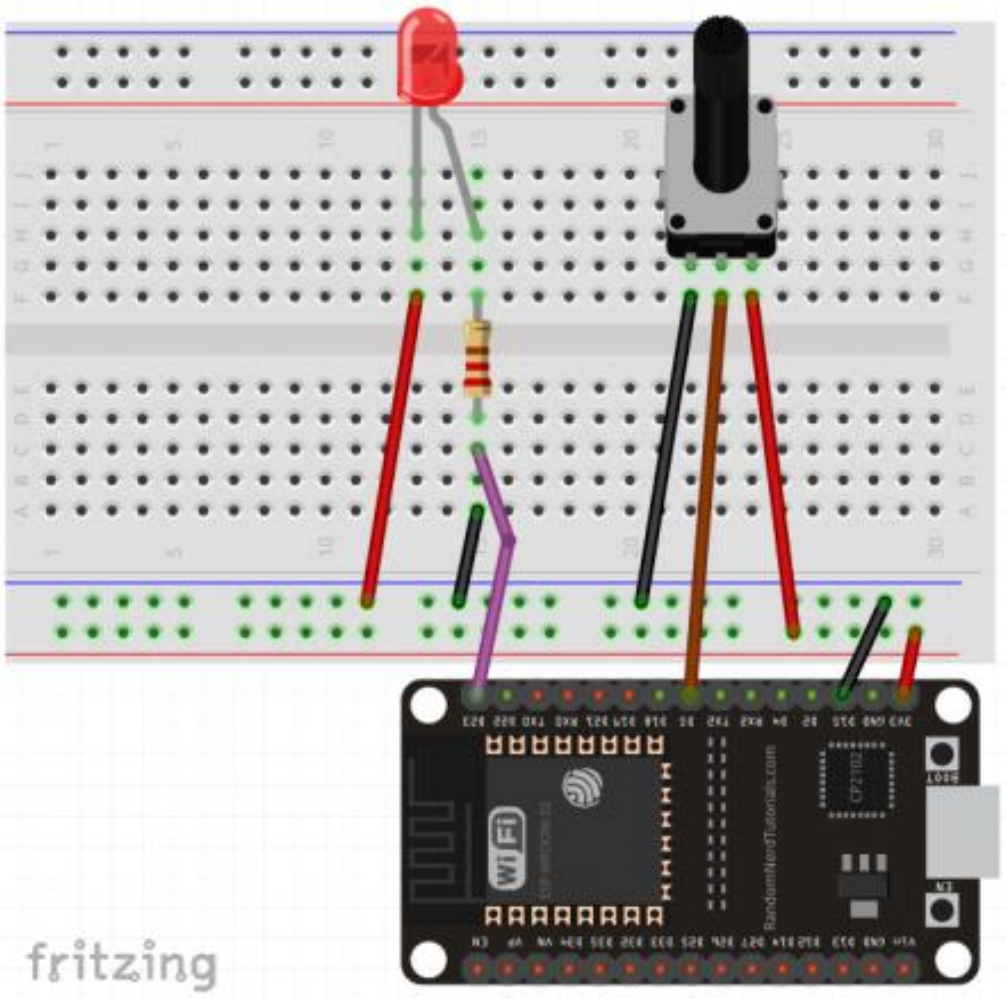


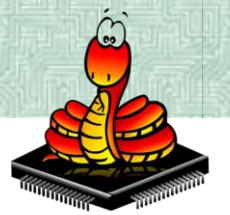
```
28  while True:
29      #Vermelho
30      rgb(255, 0, 0)
31      sleep(3)
32      #Verde
33      rgb(0, 255, 0)
34      sleep(3)
35      #Azul
36      rgb(0, 0, 255)
37      sleep(3)
38      #Amarelo
39      rgb(255, 255, 0)
40      sleep(3)
41      #Rosa
42      rgb(255, 20, 147)
43      sleep(3)
44      #Azul Claro
45      rgb(173,216,230)
46      sleep(3)
47      rgb(0,0,0)
```



Exercícios

- Realize a montagem a abaixo.





Utilizando o potenciômetro, varie a intensidade luminosa do led de acordo com o valor da leitura.

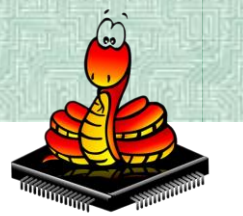
Para isso, você vai necessitar da função a seguir:

```
11  def map(leitura):
12      tensao = leitura * (3.3 / 4095)
13      tensao = int(tensao * 310) #Tensão máxima(3.3)
14      # 3.3 * 310 == 1023
15      return tensao
```

- Utilizando um led RGB, realize as seguintes transições:

Verde-Azul-Vermelho.

Dica: utilize o laço de repetição for e a função range, vista em aulas anteriores.



Referências

- https://docs.franzininho.com.br/docs/franzininho-wifi/exemplos-circuitpython/pwm