




Franzininho Na FATEC



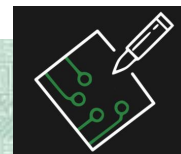


Aula 03: Sinal Analógico



Sinal Analógico

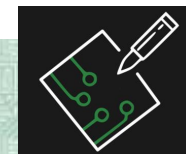
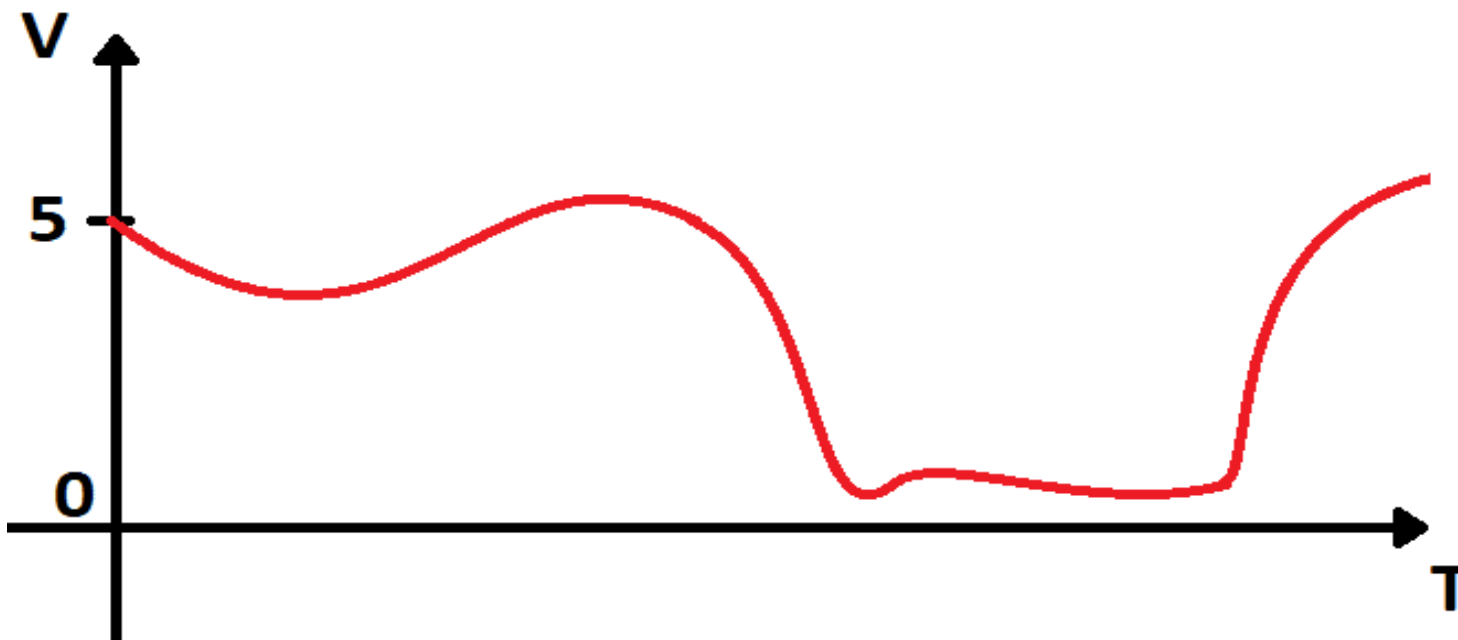
- Até um tempo atrás, o analógico ainda era muito falado. Por exemplo, antes dos relógios digitais e smart watches, as pessoas costumavam utilizar o relógio de ponteiro, também conhecido como relógio analógico.





Sinal Analógico

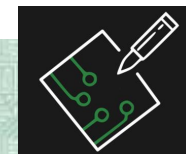
- O sinal analógico pode assumir qualquer valor dentro de um intervalo. Exemplificando, podemos utilizar a temperatura, que pode variar -15°C até 50°C , por se tratar de um valor analógico pode assumir qualquer valor dentro desse intervalo, 3.256°C ou $45,3^{\circ}\text{C}$.





Sinal Analógico

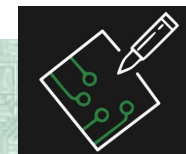
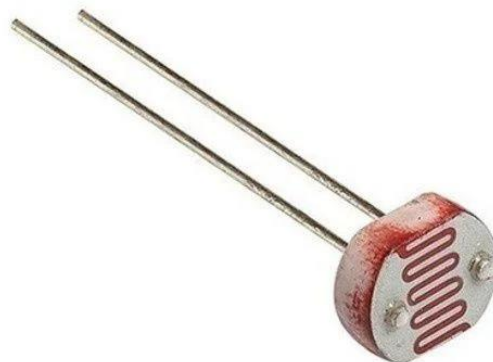
- ADC – Conversão Analógica Digital (ou, Analog Digital Converter, em inglês).
- o ADC é responsável por “transformar” a corrente elétrica do sinal analógico em um sinal digital. A onda gerada tenha a leitura realizada em vários pontos por segundo da onda elétrica gerada, após isso, é gerada uma versão digital desse sinal elétrico, estabelecido através da aproximação de valores da medição.

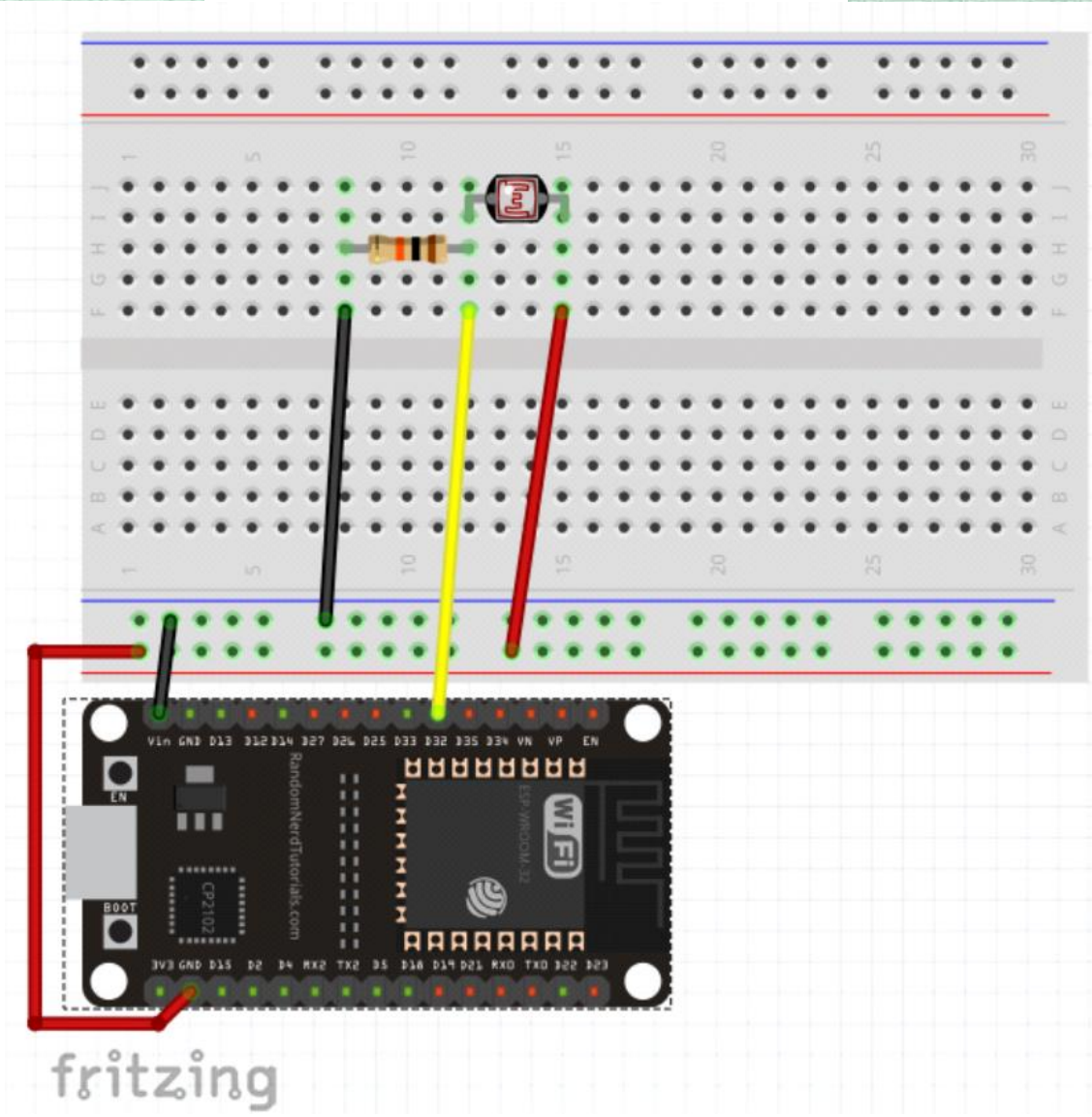




LDR (Light Dependent Resistor)

O LDR (Light Dependent Resistor), é **um componente eletrônico passivo do tipo resistor variável**, mais especificamente, é um resistor cuja resistência varia conforme a intensidade da luz que incide sobre ele. À medida que a intensidade da luz aumenta, a sua resistência diminui.



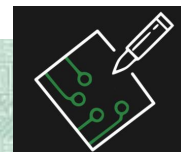




Código-fonte

- Como aprendemos antes, vamos importar a função Pin do módulo machine. Porém, dessa vez vamos necessitar de uma função chamada ADC.
- Também, precisaremos da função time para o delay.

```
3  ✓ from machine import Pin, ADC  
4    from time import sleep
```

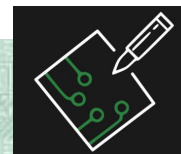




Código-fonte

- Agora, vamos configurar o pino como analógico.
- Por conseguinte, vamos “regular” o pino. Usaremos a função `ldr.atten`, que declara qual o intervalo de tensão, no nosso caso, de 0 a 3.3V:

```
6  ldr = ADC(Pin(32))  
7  ldr.atten(ADC.ATTN_11DB)
```

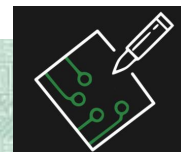




Código-fonte

- Por fim, vamos criar o nosso laço de repetição infinita, while True.

```
9  while True:
10     ldr_leitura = ldr.read() #associar o valor recebido a variável a ldr_leitura
11     print(ldr_leitura)      #imprimir valor monitor
12     sleep(2)
```





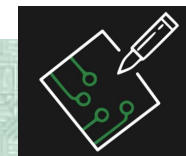
```
1  #Aula 3 - Sinal Analógico
2
3  from machine import Pin, ADC
4  from time import sleep
5
6  ldr = ADC(Pin(32))
7  ldr.atten(ADC.ATTN_11DB)
8
9  while True:
10     ldr_leitura = ldr.read()
11     print(ldr_leitura)
12     sleep(2)
```

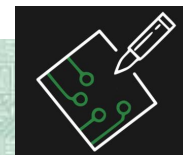
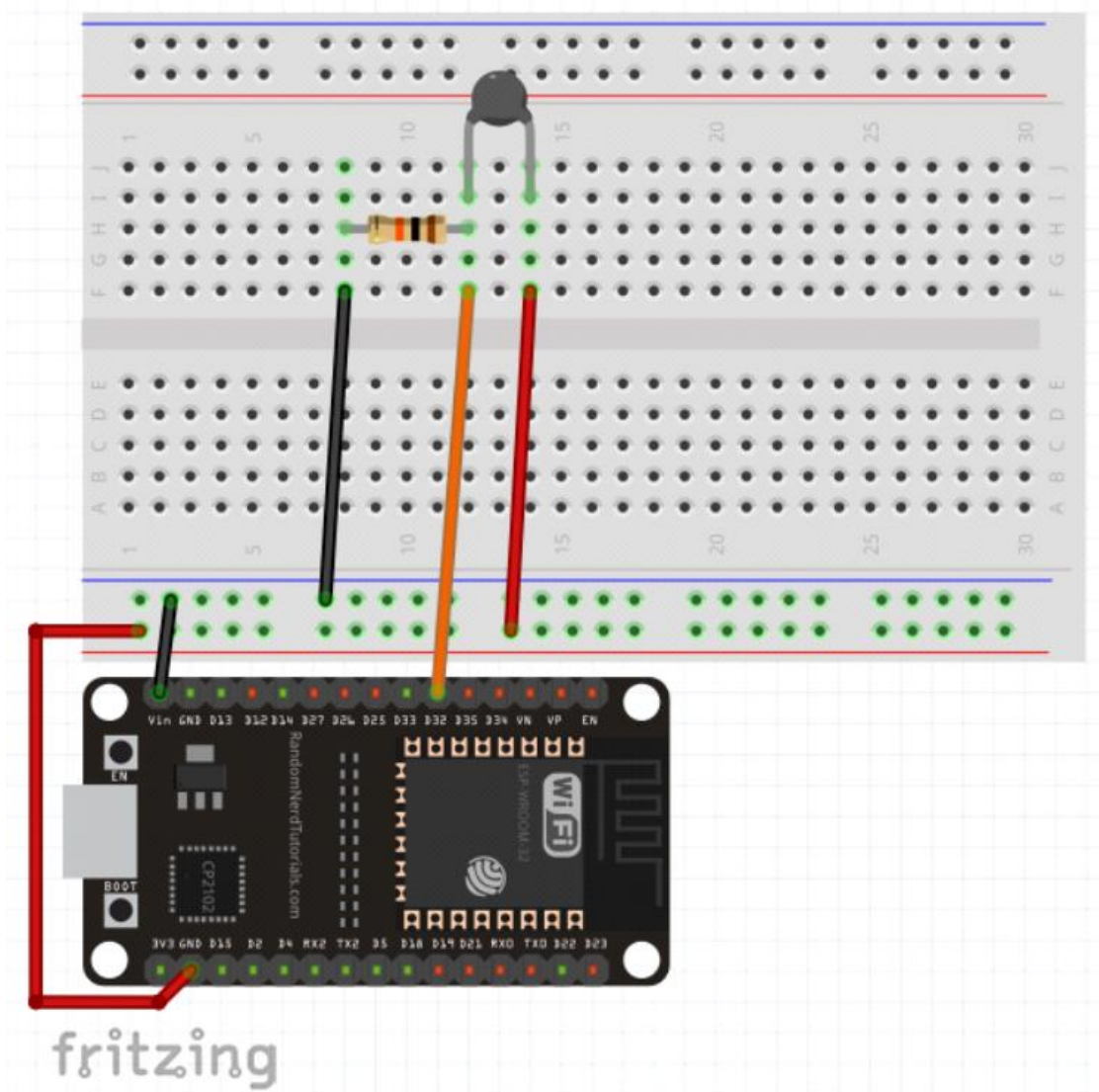




Termistor

- Agora, vamos aprender como usar um termistor, que varia sua tensão de acordo com temperatura. A montagem é mesma, assim como a configuração do pino. A única diferença é que usaremos uma função que converterá a resistência em graus Celsius:



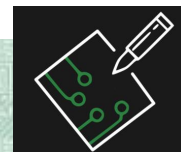




Código-fonte

- Além das bibliotecas e funções já usadas, aqui necessitaremos da biblioteca padrão do python, math:

```
1  #Exemplo Thermistor
2
3  import math
4  from machine import Pin, ADC
5  from time import sleep
```

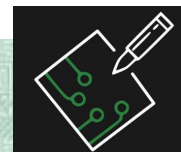




Código-fonte

- Agora, vamos configurar o pino, da mesma maneira que fizemos com o ldr:

```
7  thermistor = ADC(Pin(32))  
8  thermistor.atten(ADC.ATTN_11DB) #Atenuação -->
```

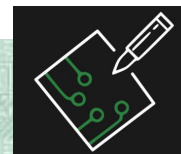




Código-fonte

- Vamos criar a função, que receberá um parâmetro, ou seja, a resistência que será convertida. Depois, vamos definir o fator, a quantia máxima de valores que serão enviados.

```
10 def temperature(r):  
11     fator = 4096  
12     TempK = math.log(10000.0*(fator / r - 1))  
13     TempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * TempK * TempK))* TempK)  
14     TempC = TempK - 273.15 #Convert Kelvin to Celsius  
15     return TempC
```

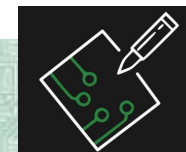




Código-fonte

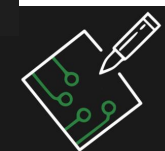
- Finalmente, em nosso loop, realizaremos a leitura do Termistor e depois chamaremos a função `temperature()`

```
17  while True:
18      adc = thermistor.read()
19      temperatura = temperature(adc)
20      print(temperatura, '°C', adc)
21      sleep(1)
```





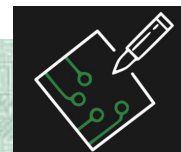
```
1  #Exemplo Thermistor
2
3  import math
4  from machine import Pin, ADC
5  from time import sleep
6
7  thermistor = ADC(Pin(32))
8  thermistor.atten(ADC.ATTN_11DB) #Atenuação -->
9
10 def temperature(r):
11     fator = 4096
12     TempK = math.log(10000.0*(fator / r - 1))
13     TempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * TempK * TempK))* TempK)
14     TempC = TempK - 273.15 #Convert Kelvin to Celsius
15     return TempC
16
17 while True:
18     adc = thermistor.read()
19     temperatura = temperature(adc)
20     print(temperatura, '°C', adc)
21     sleep(1)
```

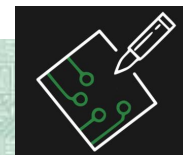
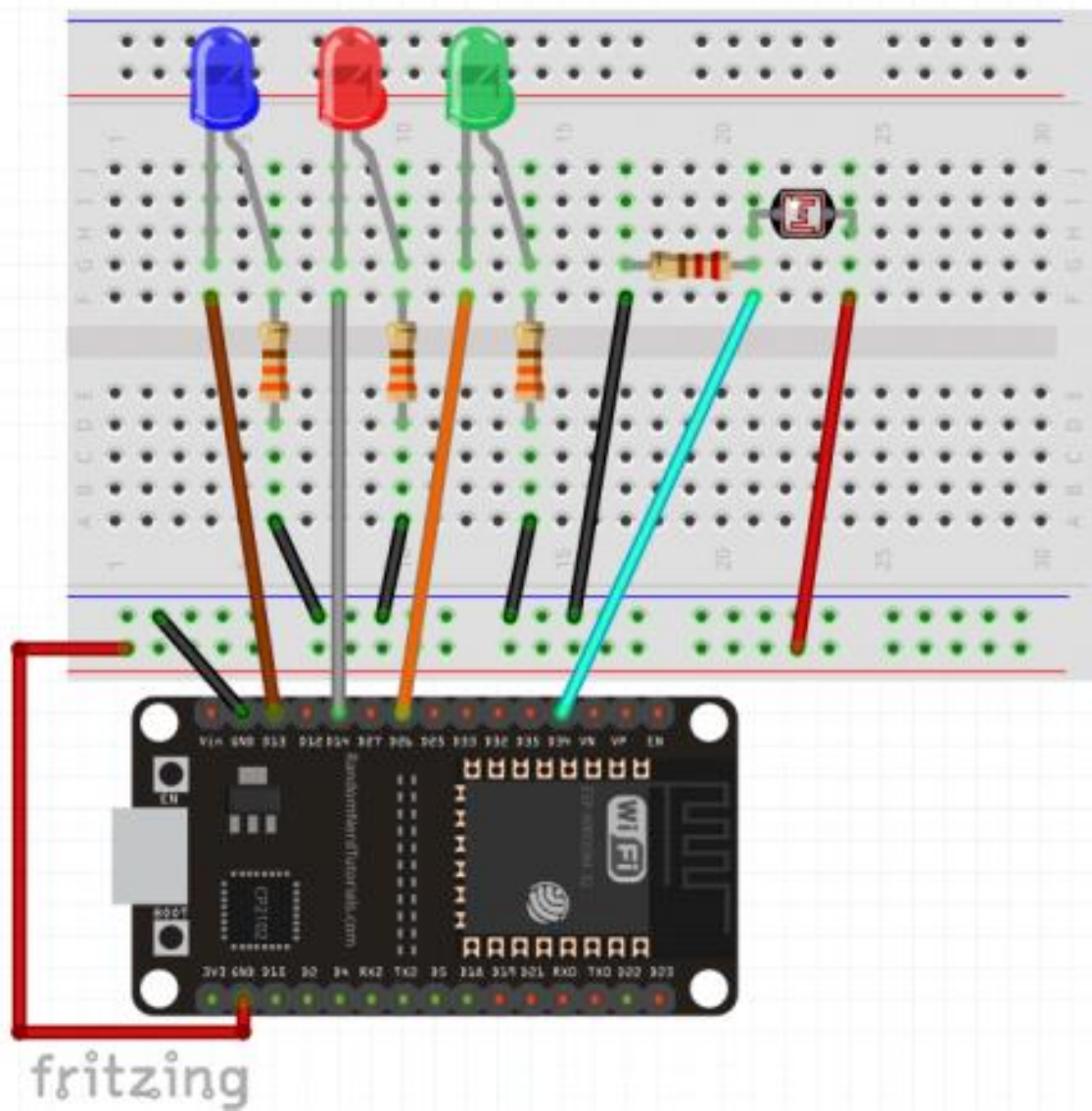




Exercícios

- Aqui vamos fazer uma espécie de termômetro para luminosidade, quanto mais escuro, mais leds ligados. Realiza a montagem abaixo:







Siga o seguinte algoritmo:

- Se a leitura for maior ou igual a 0 e menor ou igual ao valor máximo (4095) dividido por 3, apenas um led ligado;
- Caso seja, maior que $4095/3$ e menor ou igual a 4095, dois leds ligados;
- E, se for maior que $4095/2$ e menor igual ao valor máximo, 3 leds ligados.

