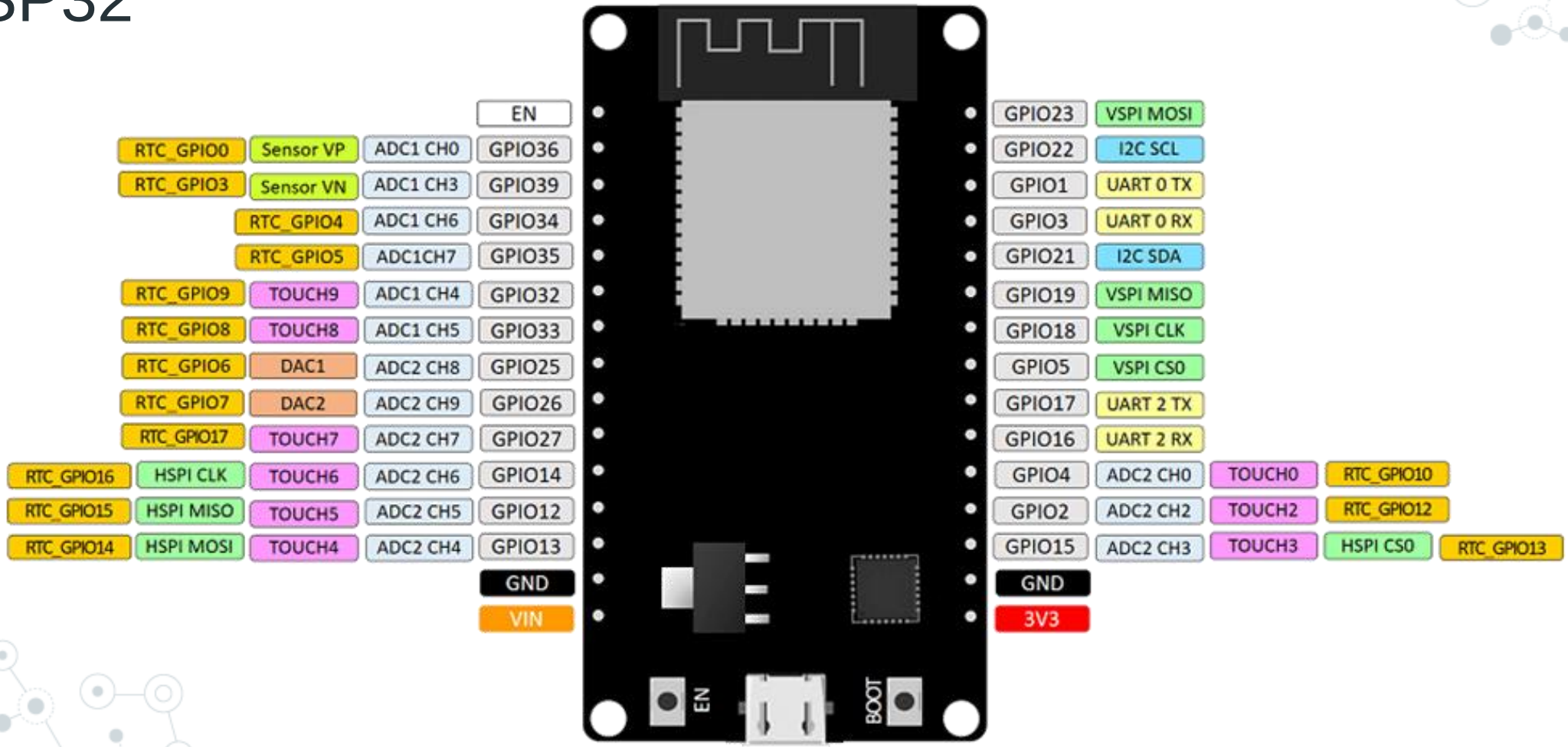
centro universitário FEI

Sistemas Embarcados

Prof. Dr. André Luiz Perin
Prof. Dr. Marco Antonio A. Melo
Prof. Dr. Rudolf T. Bühler

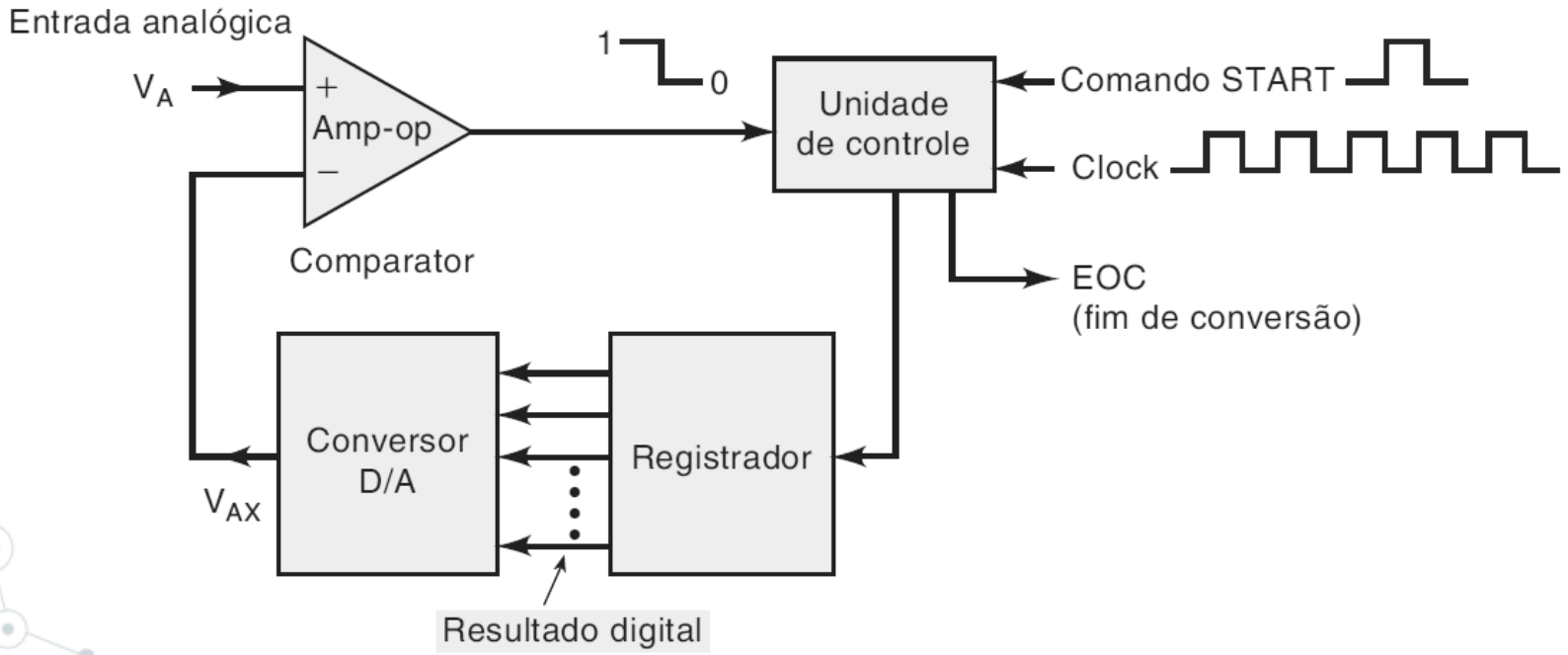
Sistemas Embarcados

- ESP32



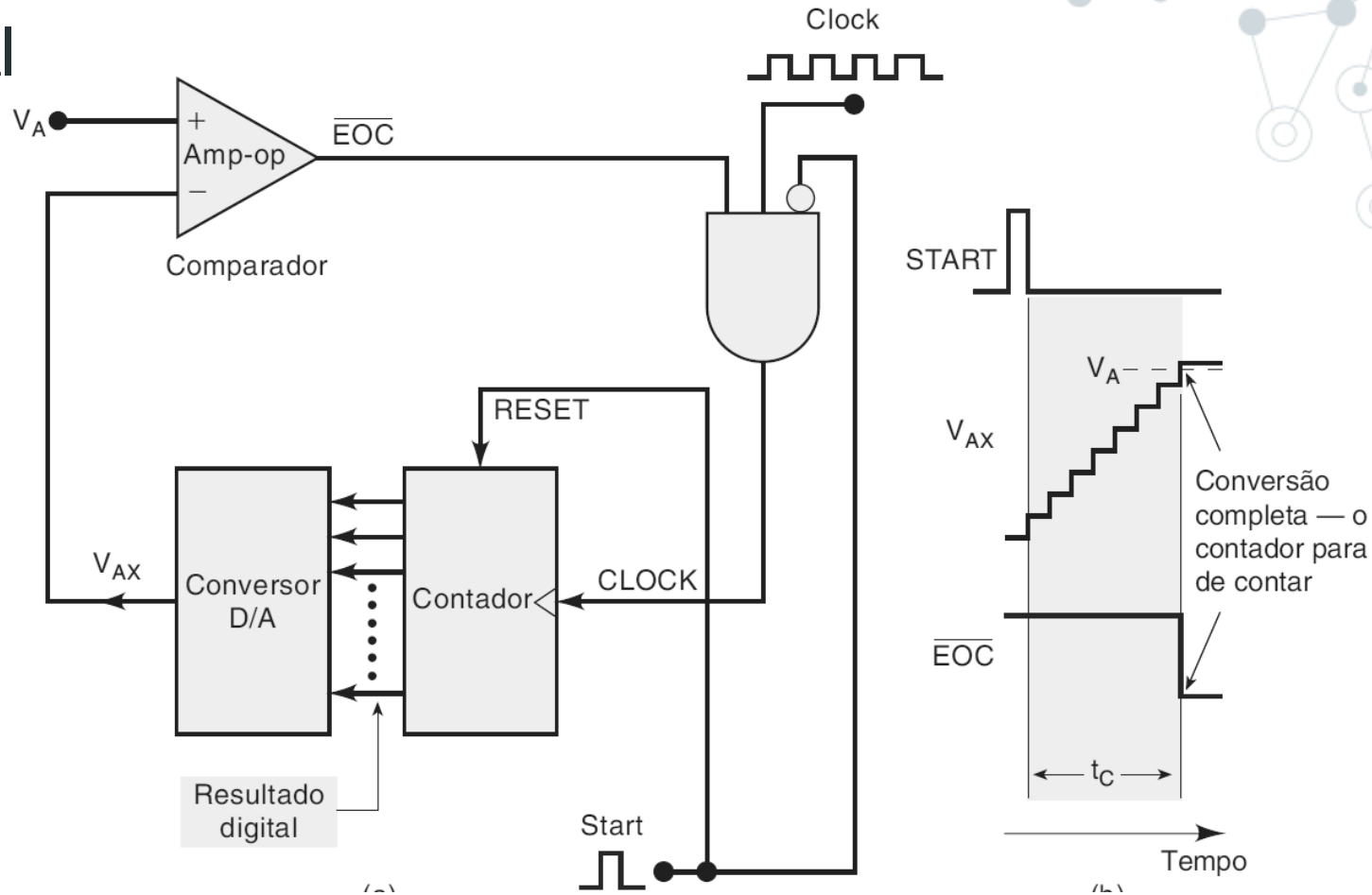
Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital
    - ADC (Analogic Digital Converter)
      - Diagrama geral de uma classe de ADCs



Sistemas Embarcados

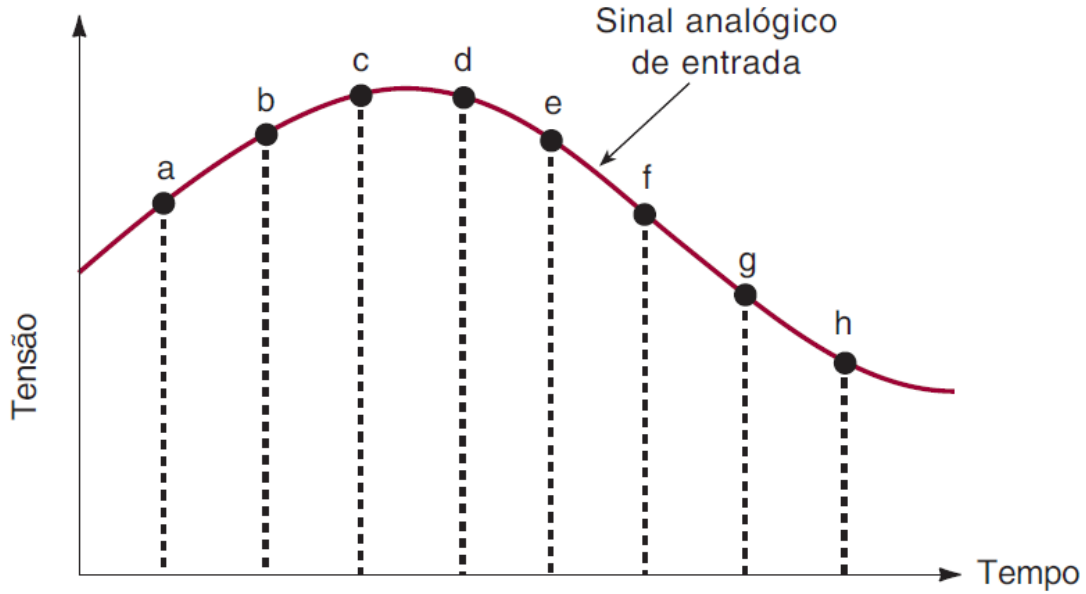
- Node MCU
  - Conversor Analógico Digital
    - ADC (Analogic Digital Converter)
      - ADC de rampa digital



https://thonny.org

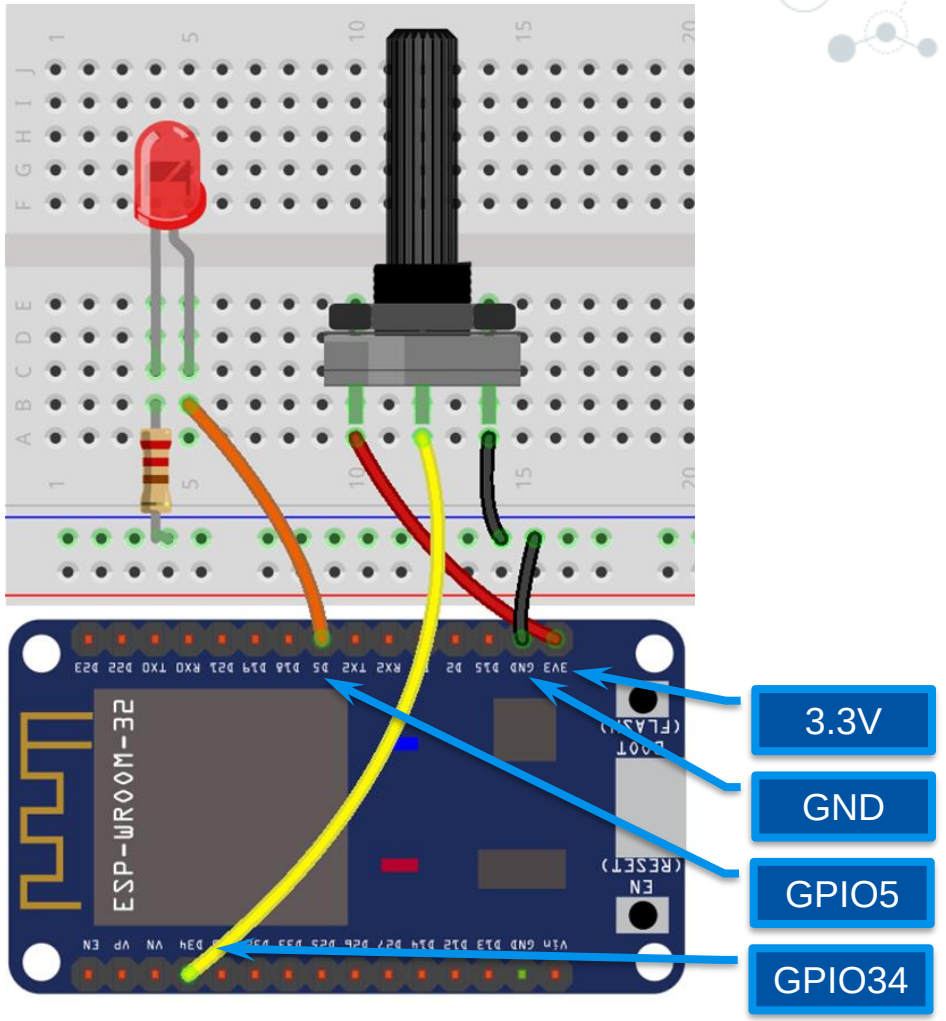
Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital
    - ADC (Analogic Digital Converter)
      - Digitalização de um sinal analógico



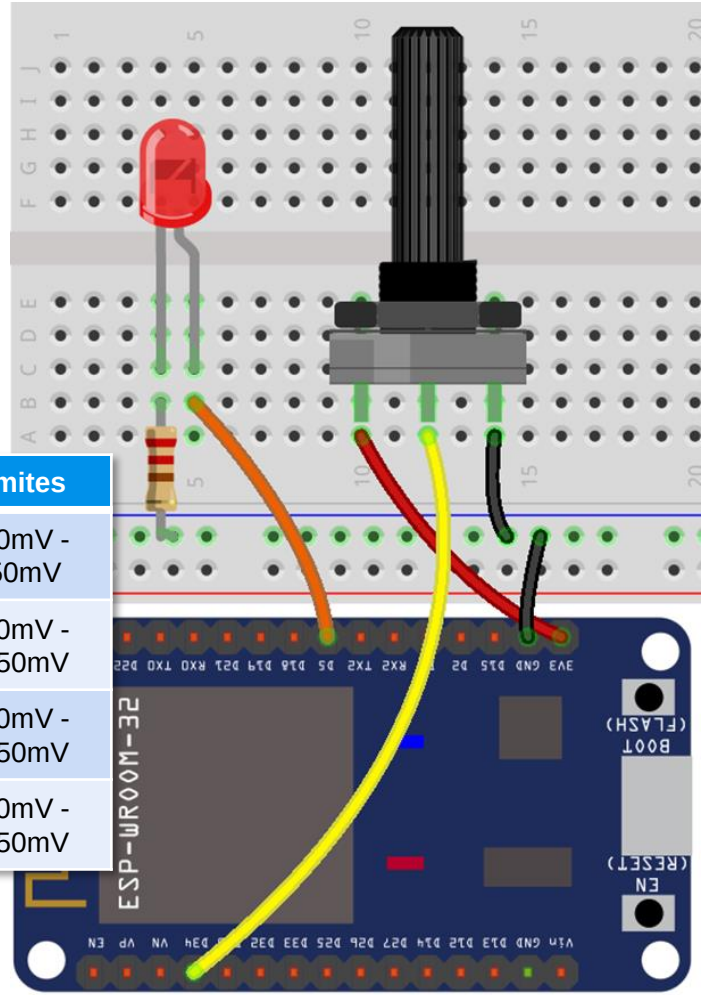
Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital
    - Materiais
      - Node MCU
      - 1 Resistor 220 Ω
      - 1 LED
      - 1 Potenciômetro 10 $k\Omega$
      - 1 Protoboard
      - Fios e jumpers



Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital



Parâmetro	Descrição	Limites
ADC.ATTN_0DB	Sem atenuação	100mV - 950mV
ADC.ATTN_2_5DB	Atenuação de 2.5dB	100mV - 1250mV
ADC.ATTN_6DB	Atenuação de 6dB	150mV - 1750mV
ADC.ATTN_11DB	Atenuação de 11dB	150mV - 2450mV

Frequência regulada pelo potenciômetro

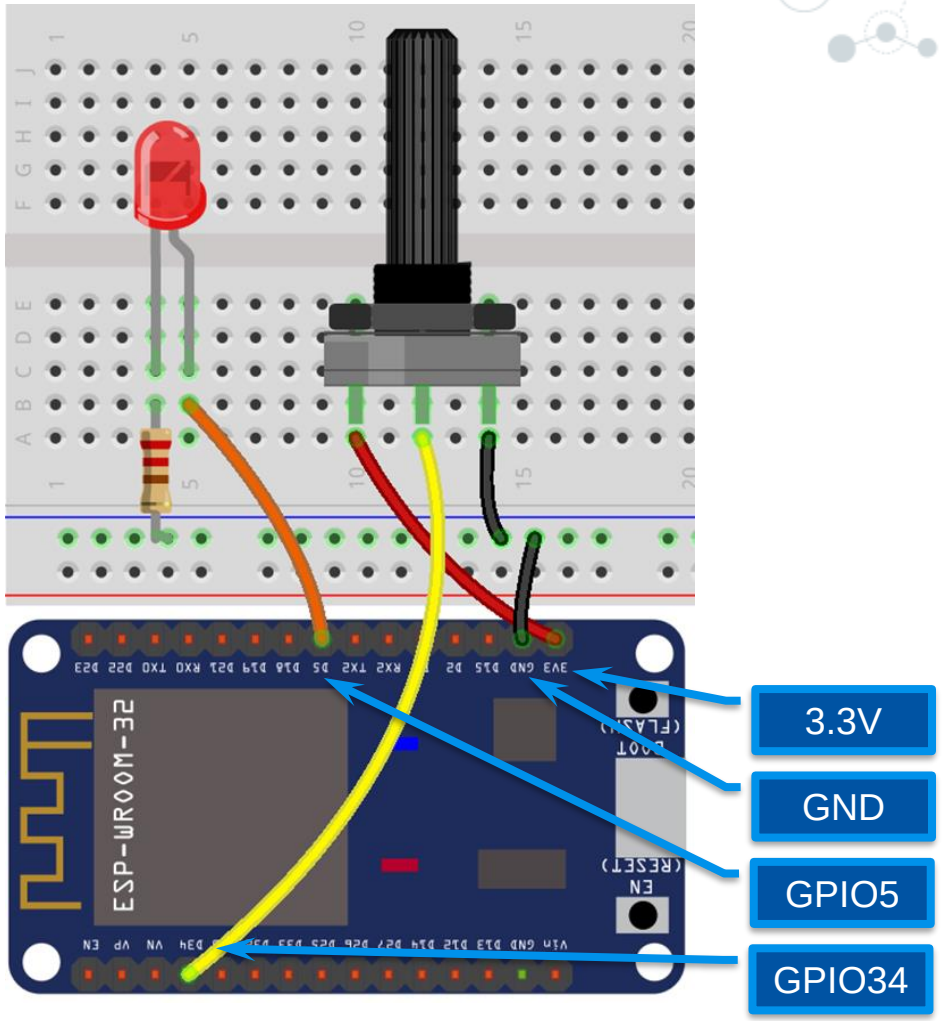
```
from time import sleep_ms
from machine import ADC, Pin

led = Pin(5, Pin.OUT)
pot = ADC(Pin(34))
pot.atten(ADC.ATTN_11DB)
estado = 0

try:
    while True:
        led.value(estado)
        estado = not(estado)
        sleep_ms( int(pot.read()/4) )
except KeyboardInterrupt:
    led.value(0)
```

Sistemas Embarcados

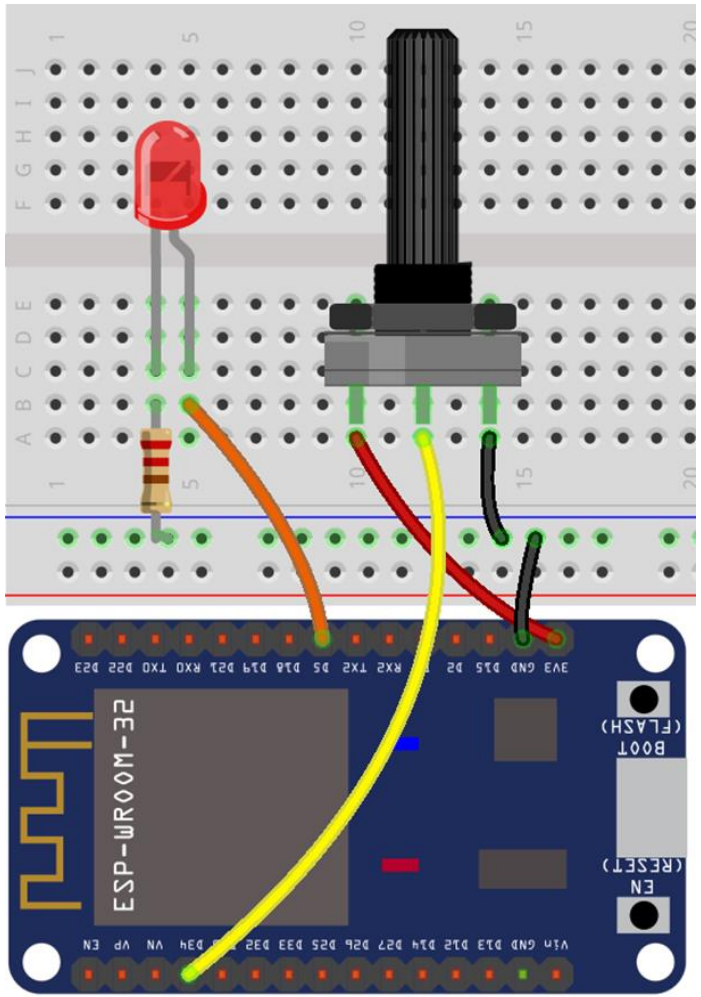
- Node MCU
  - Conversor Analógico Digital
    - Materiais
      - Node MCU
      - 1 Resistor 220 Ω
      - 1 LED
      - 1 Potenciômetro 10 $k\Omega$
      - 1 Protoboard
      - Fios e jumpers



Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital

Parâmetro	Descrição	Limites
ADC.ATTN_0DB	Sem atenuação	100mV - 950mV
ADC.ATTN_2_5DB	Atenuação de 2.5dB	100mV - 1250mV
ADC.ATTN_6DB	Atenuação de 6dB	150mV - 1750mV
ADC.ATTN_11DB	Atenuação de 11dB	150mV - 2450mV



Intensidade regulada pelo potenciômetro

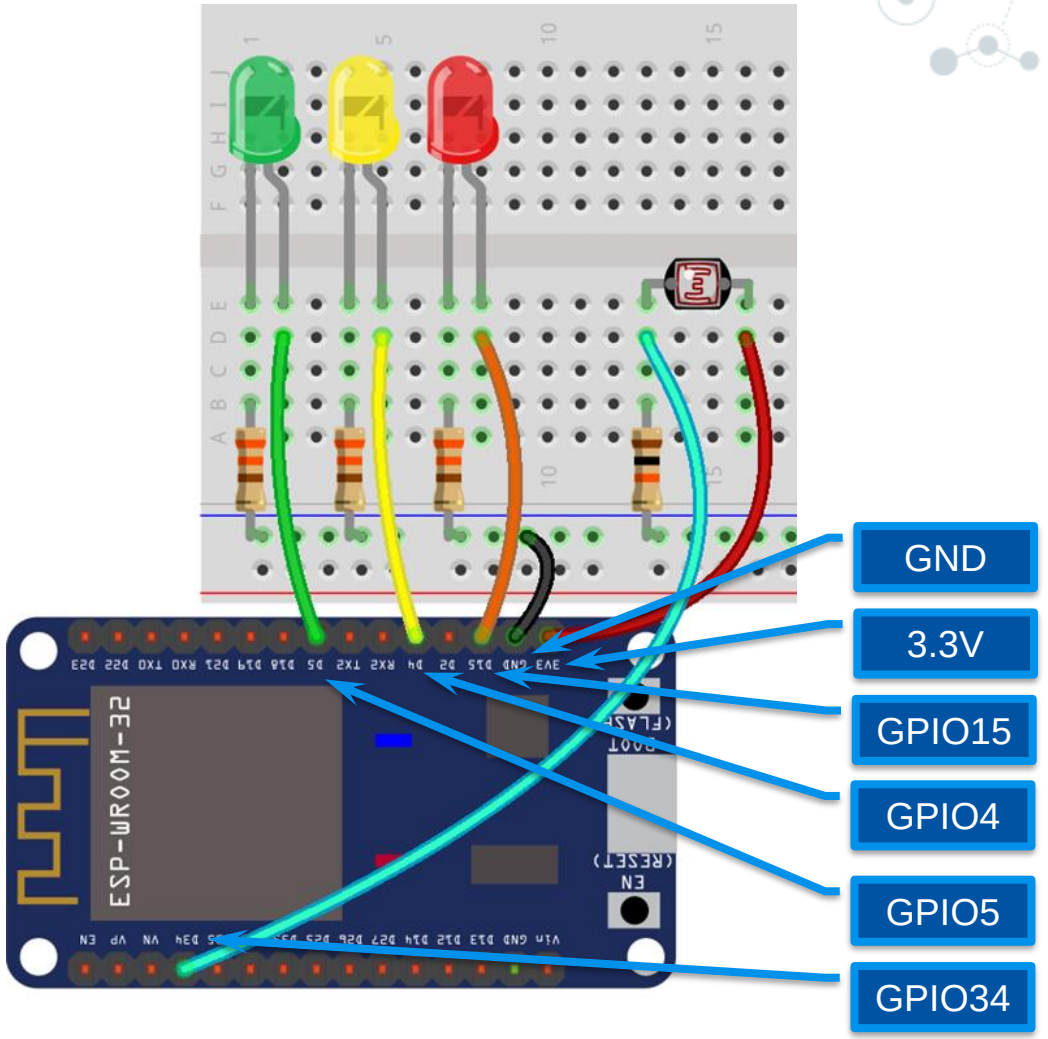
```
from time import sleep
from machine import ADC, Pin, PWM

led = PWM(Pin(5), freq=20000, duty = 0)
pot = ADC(Pin(34))
pot.atten(ADC.ATTN_11DB)
estado = 0

try:
   while True:
      led.duty( int(pot.read()/4) )
      sleep(0.01)
except KeyboardInterrupt:
   led.value(0)
```

Sistemas Embarcados

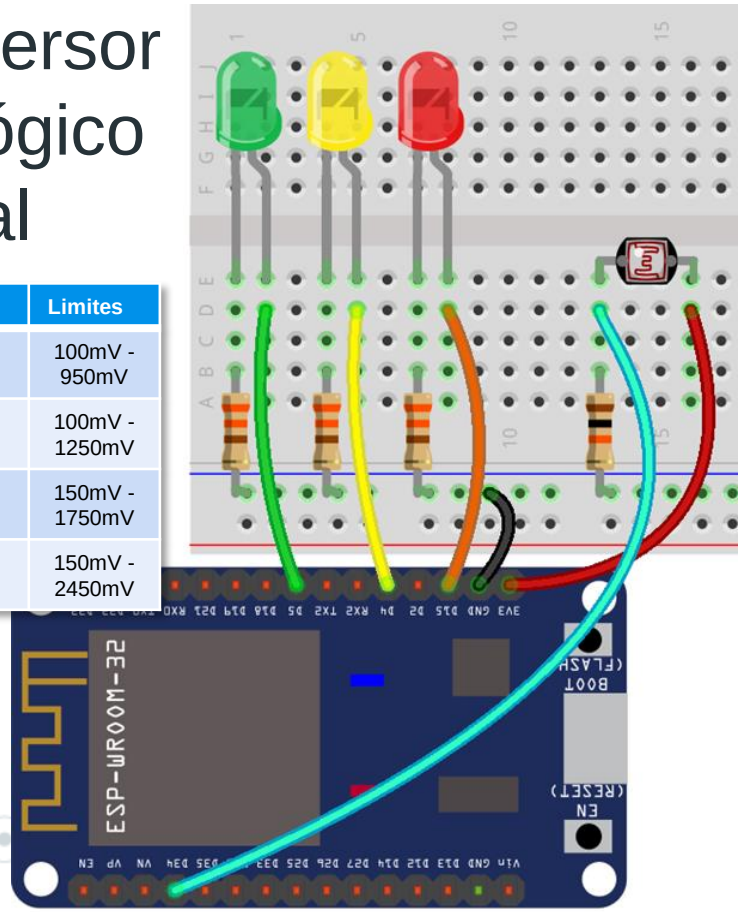
- Node MCU
  - Conversor Analógico Digital
    - Materiais
      - Node MCU
      - 3 Resistores 220 Ω
      - 1 Resistor 1 $k\Omega$
      - 1 LED Verde
      - 1 LED Amarelo
      - 1 LED Vermelho
      - 1 LDR
      - 1 Protoboard
      - Fios e jumpers



Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital

Parâmetro	Descrição	Limites
ADC.ATTN_0DB	Sem atenuação	100mV - 950mV
ADC.ATTN_2_5DB	Atenuação de 2.5dB	100mV - 1250mV
ADC.ATTN_6DB	Atenuação de 6dB	150mV - 1750mV
ADC.ATTN_11DB	Atenuação de 11dB	150mV - 2450mV



Medidor de intensidade luminosa

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

def mapear(x, in_min, in_max,
out_min, out_max):
  return (x - in_min) *
(out_max - out_min) / (in_max -
in_min) + out_min

led_vd = Pin(5, Pin.OUT)
led_am = Pin(4, Pin.OUT)
led_vm = Pin(15, Pin.OUT)

ldr = ADC(Pin(34))
ldr.atten(ADC.ATTN_11DB)

try:
  while True:
    valor_ldr = ldr.read()
    valor = int( mapear(
valor_ldr, 0, 4095, 0, 4))
    print ("ADC: ", valor_ldr,
"Mapear:", valor)
    if valor == 0:
      led_vd.value(0)
      led_am.value(0)
      led_vm.value(0)
    elif valor == 1:
      led_vd.value(1)
      led_am.value(0)
      led_vm.value(0)
    elif valor == 2:
      led_vd.value(1)
      led_am.value(1)
      led_vm.value(0)
    elif valor == 3:
      led_vd.value(1)
      led_am.value(1)
      led_vm.value(1)
    sleep(0.1)
except:
  led_vm.value(0)
  led_am.value(0)
  led_vd.value(0)
```

Sistemas Embarcados

- Node MCU
  - Conversor Analógico Digital