

Aula 2

Introdução a

# Microcontroladores

com ESP32 e MicroPython



x x x x

# Sensores e Atuadores

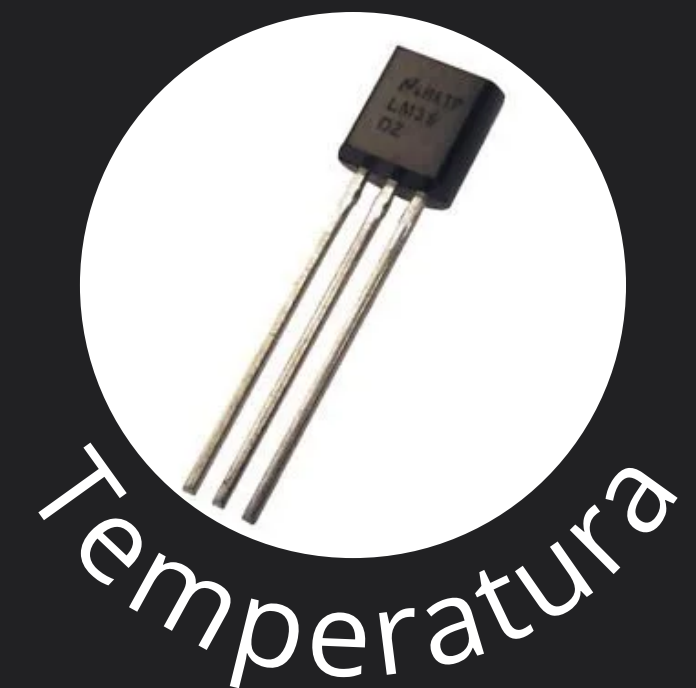
Os sensores e atuadores são componentes eletrônicos e a forma do nosso sistema interagir com o ambiente e suas variáveis.



x x x x

# Sensores

são componentes que medem variáveis do ambiente e convertem essa informação em sinais elétricos, utilizando um transdutor interno, para usar como entrada no controlador.

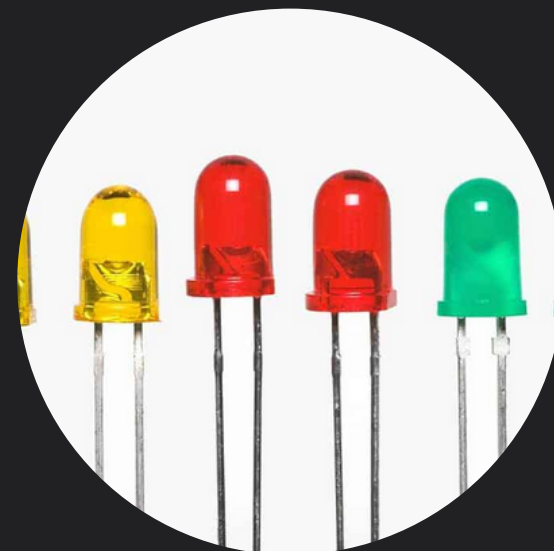


× × × ×

x x x x



Servo motor



LED



Buzzer



Display

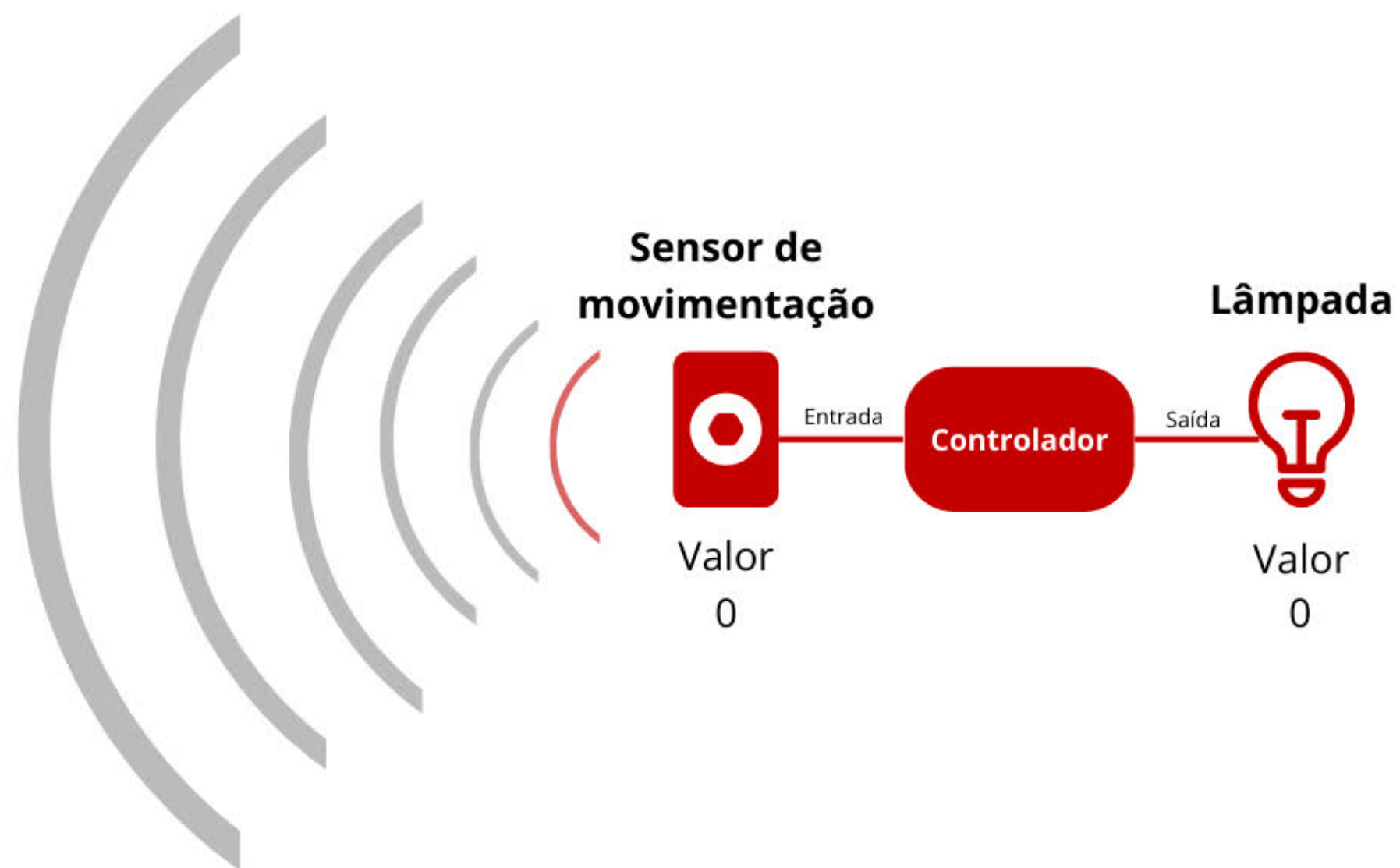
x x x x

# Atuadores

Manipulam as variáveis do ambiente, agindo como uma saída pro controlador. Ou seja, eles transformam os sinais elétricos em outros tipos de energia para criar alguma ação no ambiente.

x x x x

# Exemplo de sistema

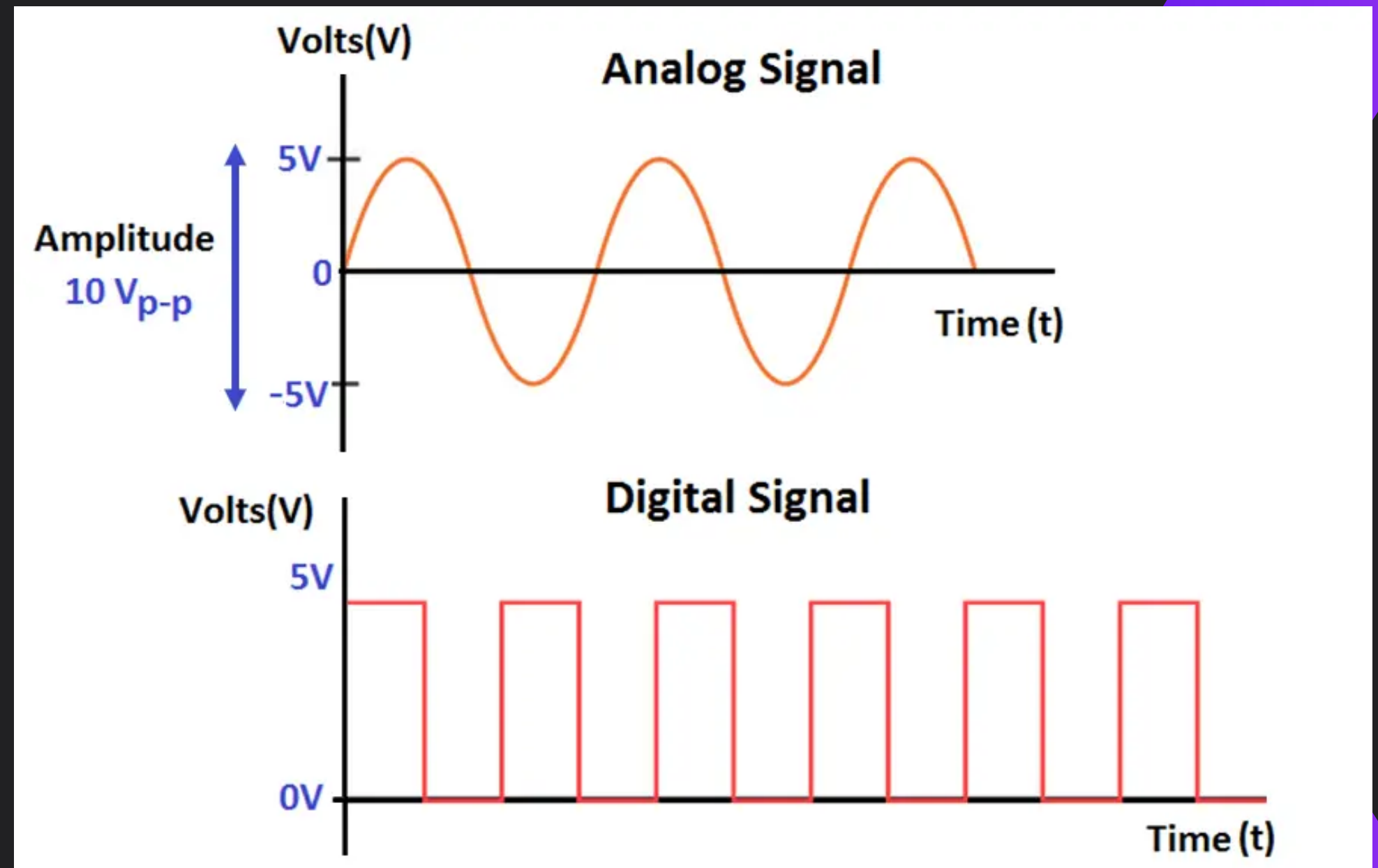


x x x x



# Sinal digital x analógico

Os sinais de entrada e saída podem ser definidos como digitais ou analógicos.



# Sinal digital

O sinal digital pode assumir apenas dois valores no seu sinal, que podem ser interpretados como zero ou um.

**Exemplo:** um sensor de luminosidade que devolve um 1 caso esteja claro e 0 caso não esteja

# Sinal digital – Implementação

```
import machine

# Criando um pin de saída no pino #0
saida = machine.Pin(0, machine.Pin.OUT)

# Mudando o valor
saida.value(0) # Você também pode usar p0.on()
saida.value(1) # Você também pode usar p0.off()

# Criando um pin de entrada no pino #2
entrada = machine.Pin(2, machine.Pin.IN)

# Ler e printar o valor do pino de entrada
print(entrada.value())
```



# Sinal analógico

O sinal analógico pode assumir qualquer valor no seu sinal dentro de uma faixa de operação.

**Exemplo:** um sensor de luminosidade que devolve um valor para a luminosidade do local

Há algumas implementações gerais sinais analógicos no ESP32 com Micropython:

- ADC (para sensores)
- DAC (para atuadores)
- PWM (para atuadores)

x x x x

# ADC

## (Analog to Digital Converter)

Nossa microcontroladora é um "computadorzinho" que só entende 0 e 1. Quando temos um sensor que pega um valor analógico (que pode assumir qualquer valor numa faixa específica), é necessário converter esse valor analógico em 0 e 1 para que a placa entenda

Sensores

Sinal analógico -> Digital

Pinos: 0, 2, 4, 12, 13, 14, 15, 25, 26, 27, 32, 33, 34, 35, 36, 39

x x x x



x x x x

# ADC

## (Analog to Digital Converter)

```
import machine

# Criando um pin de entrada ADC no pino #12
entrada = machine.ADC(machine.Pin(12))

# Ler e printar o valor analógico do pino de entrada
print(entrada.read())
```

x x x x



x x x x

# DAC

## (Digital to Analog Converter)

Este é o contrário: nosso atuador precisa agir no ambiente com um valor analógico. Para que a controladora, que só se expressa em 0 e 1, envie o sinal analógico que o atuador precisa, é necessário converter!

Atuadores

Sinal digital -> Analógico

Pinos: 25 e 26

x x x x

x x x x

# DAC

## (Digital to Analog Converter)

```
import machine

# Criando um pin de saída DAC no pino #25
saida = machine.DAC(machine.Pin(25))

# Mudar o valor de sinal do pino (entre 0 e 255)
saida.write(150)
```

x x x x

# PMW

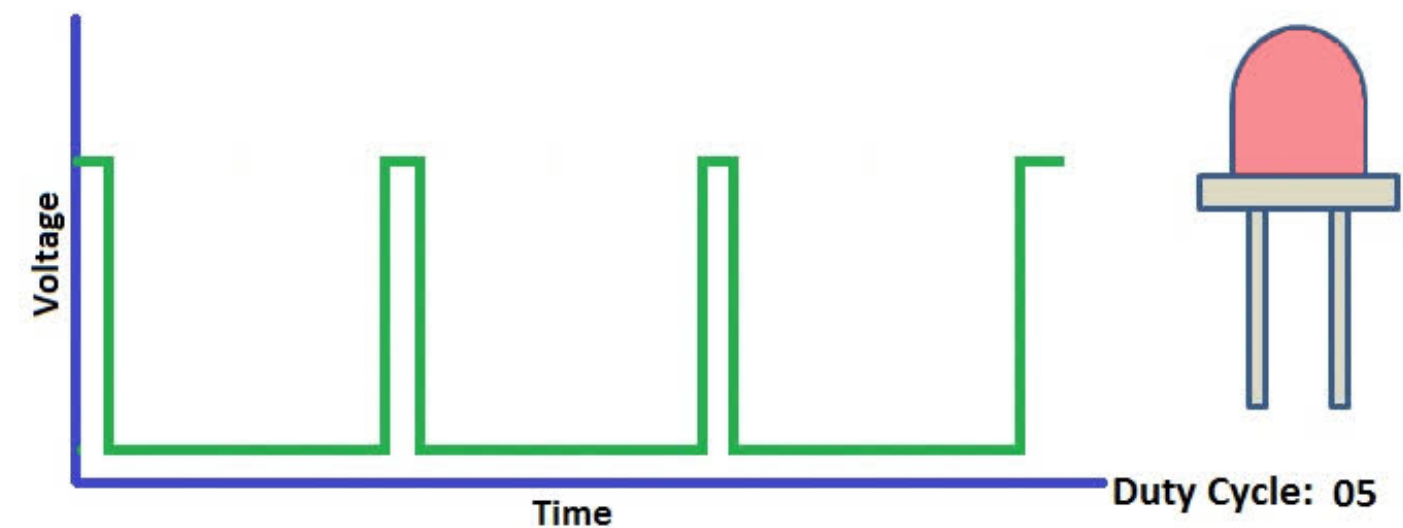
## (Pulse Width Modulation)

Para criar uma saída analógica para um atuador em um pino digital, podemos usar o PMW. Consiste em oscilar o sinal digital muito rápido, dando a impressão de um sinal analógico. É possível utilizá-lo em atuadores digitais, como os LEDs.

Atuadores

Sinal digital que SIMULA analógico

Pinos: Qualquer saída analógica





# PMW

## (Pulse Width Modulation)

- **Frequência:** Pode ser um valor entre 0 e 78125. A frequência de 5000 Hz geralmente é a mais comum e pode ser usada para controlar o brilho de um LED.
- **Ciclo de trabalho (valor):** É um valor entre 0 e 1023, no qual 1023 corresponde a 100% do ciclo de trabalho (1 - ligado) e 0 corresponde a 0% do ciclo de trabalho (0 - desligado)

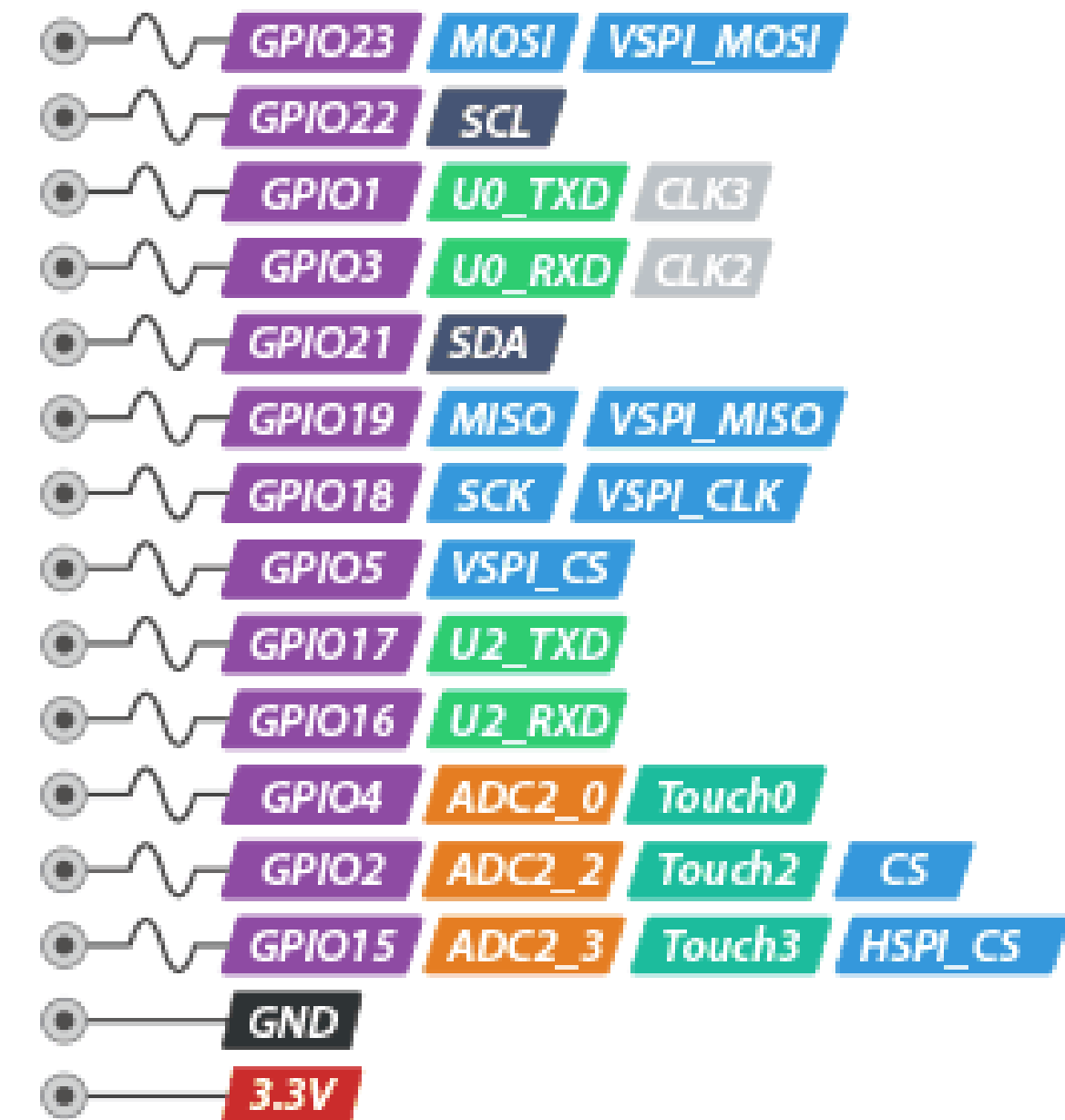
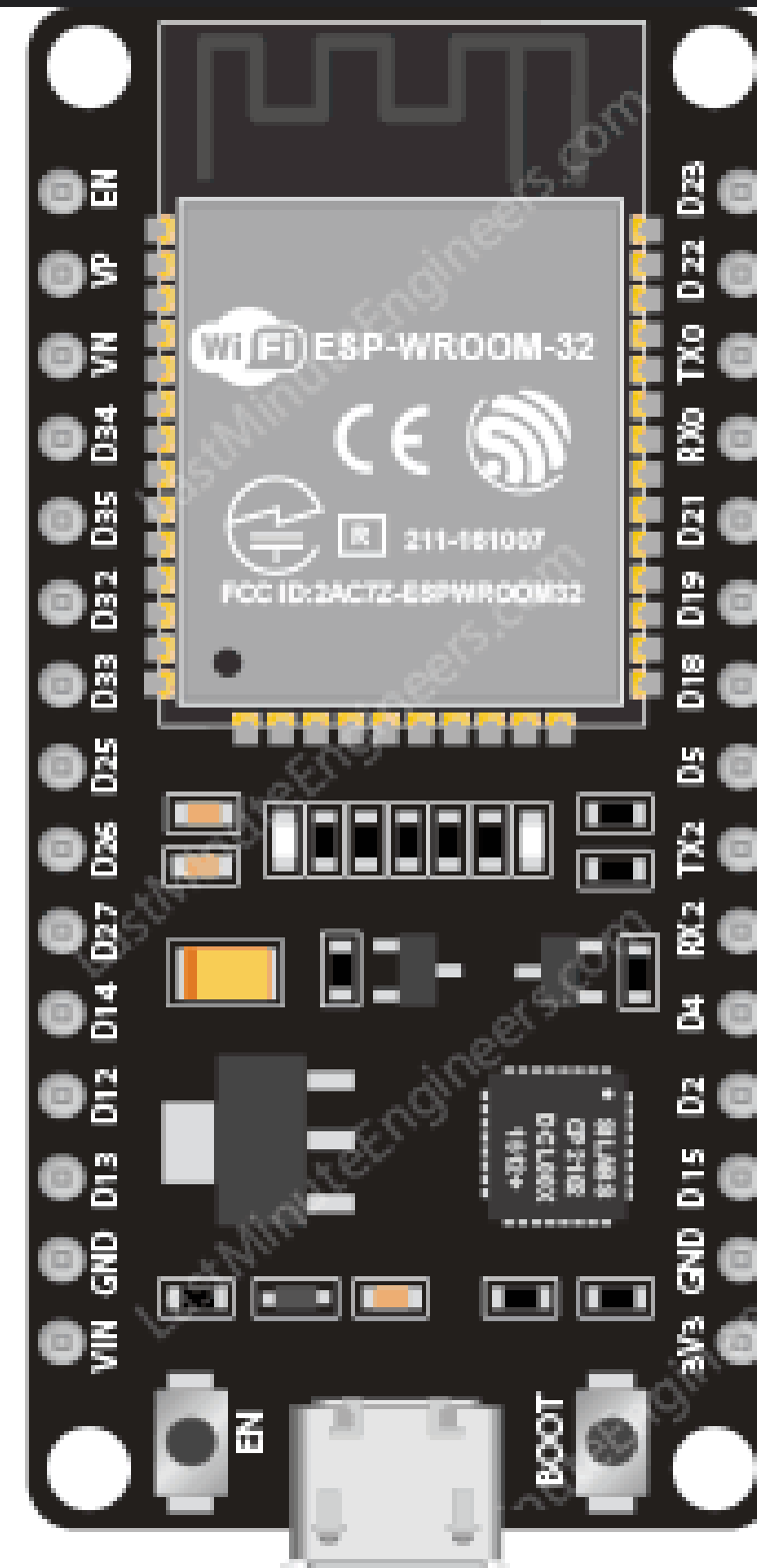
```
import machine

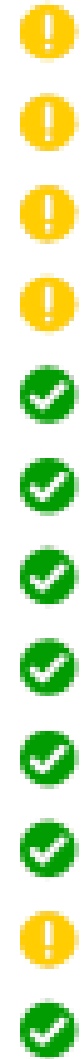
# Criando um pin de entrada PWM no pino #0
# O primeiro parâmetro é o Pin, o segundo é a frequência e o terceiro é o ciclo de trabalho
saida = machine.PWM(machine.Pin(0), freq=5000, duty_u16=32768)

# Mudar o valor do ciclo de trabalho
saida.duty(valor)
```



Em qual pino  
conecto?



















Os pinos verdes são seguros para uso. Os amarelos podem apresentar comportamento imprevisível e é recomendado evitar os vermelhos

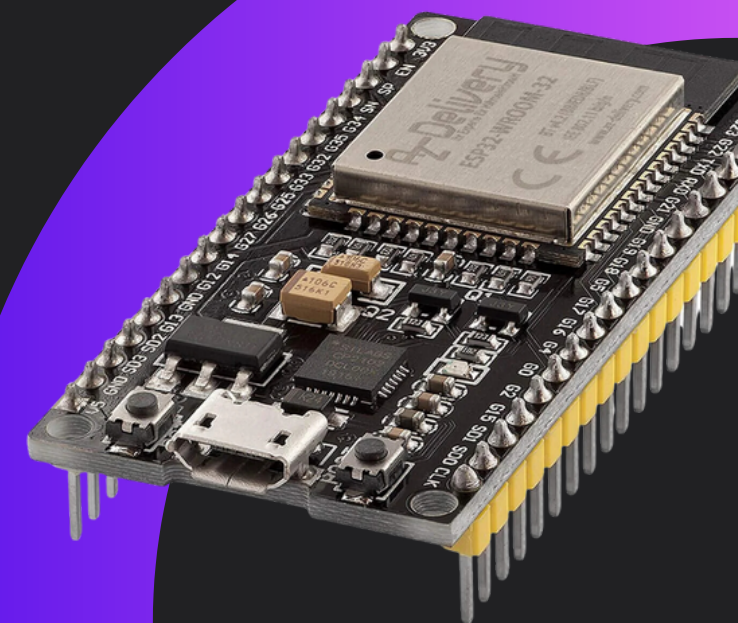
Como usar  
novos atuadores  
e sensores?

# Como escolher e comprar um sensor/atuador?

1. Vá em um site de compra de componentes eletrônicos
2. Procure pela ação que você precisa (por exemplo: sensor de iluminação ou LED azul)
3. Existem diferentes componentes com a mesma função, leia a descrição para encontrar o melhor para seu projeto!

## Sugestões de lojas:

- Baú da Eletrônica
- Curto circuito

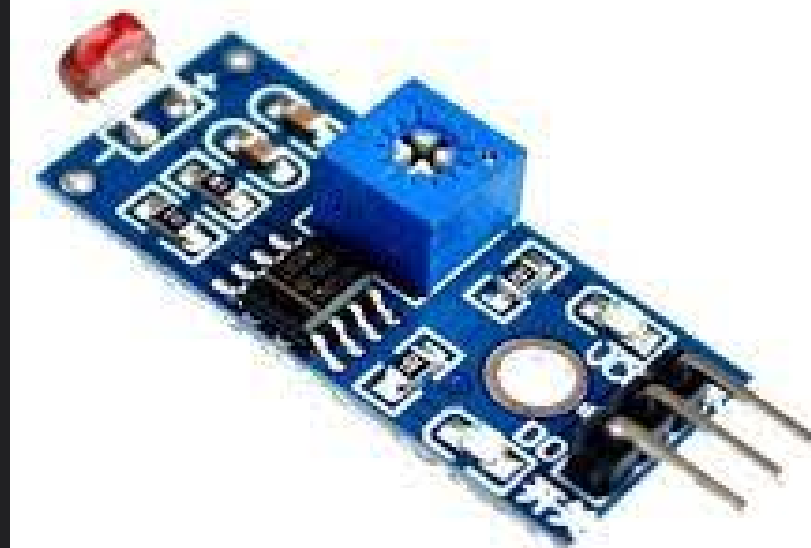




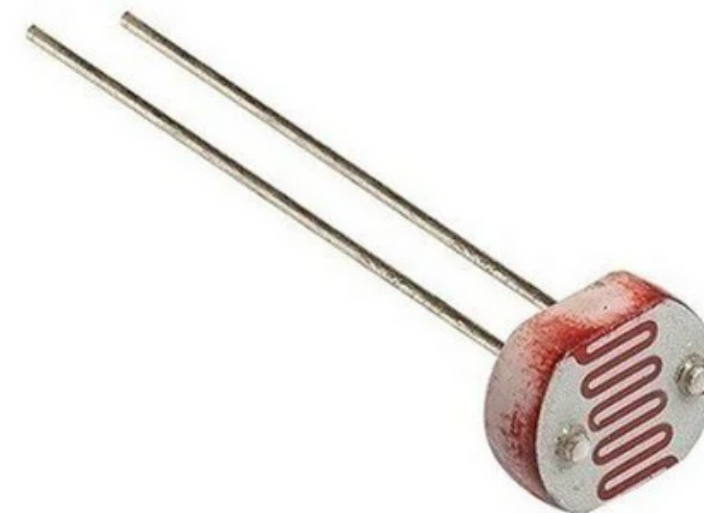
# Módulo ou sem módulo

Alguns componentes podem ter uma versão original, que é apenas o sensor, ou versão com módulo, no qual há um pequeno circuito para facilitar ações como definir valores para funcionamento e conexão no seu circuito.

LDR com módulo



LDR sem módulo



Como saber as informações desse sensor/atuator? (O que ele faz, como montar o circuito, especificações...)

# Datasheet

O datasheet é um documento que reúne dados do componentes, tais como:

- Nome e descrição da funcionalidade do componente
- Circuitos padrões ou exemplos de aplicações
- Valores máximos e mínimos de trabalho
- Outras informações técnicas que podem ser úteis sobre o uso do componente

## Encontre-o:

- Pesquisando no google o nome do componente + datasheet
- Aproveite para ver sites com outros exemplos mais entendíveis
- Veja no site em que comprou o componente (resumo)

Como programar esse  
sensor/atuador com  
MicroPython no ESP32?

# Sozinho...

Agora que você sabe qual componente tem em mão, como programá-lo? **Usando a documentação!**

Caso tenha dúvidas sobre funções, módulos, protocolos e drivers do ESP32 com MicroPython, veja a [documentação do MicroPython para ESP32](#)

Caso não resolva seu problema, procure por guias online utilizando o mesmo sensor, vendo seu funcionamento e se é necessário módulos e protocolos diferentes também. [RandomNerdTutorials](#) possuem guias de passo-a-passo de vários componentes no ESP32 com Micropython, mas fique à vontade para explorar outros :)

# Mãos na massa!

Bora definir quais componentes você vai usar.

Veja circuitos e códigos de diversos sensores e atuadores no nosso [Github](#), na pasta da aula 2!

Caso não encontre o que quer lá, não tenha vergonha de pesquisar :)