

# DevTitans

Wokwi: um emulador para ESP32

# WOKWI

<https://wokwi.com/>



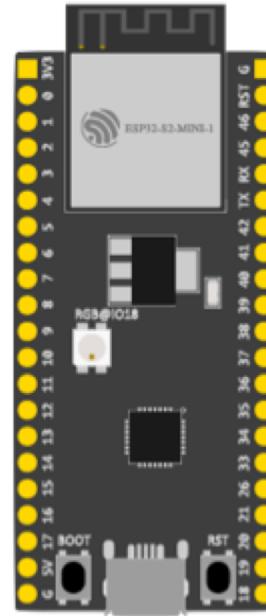
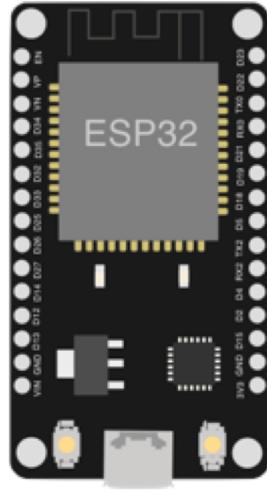
Simulate IoT Projects in Your Browser

# ESP32's do Wokwi

YOUR PROJECTS YOUR LIKES ▾

+ NEW PROJECT

- Arduino Uno
- Arduino Mega
- Arduino Nano
- ATtiny85
- ESP32
- ESP32-S2
- ESP32-C3
- TinyPICO
- MicroPython on ESP32
- Raspberry Pi Pico
- Raspberry Pi Pico (SDK)
- MicroPython on Raspberry Pi Pico
- CircuitPython on Raspberry Pi Pico
- Franzininho
- Franzininho WiFi (ESP32-S2)



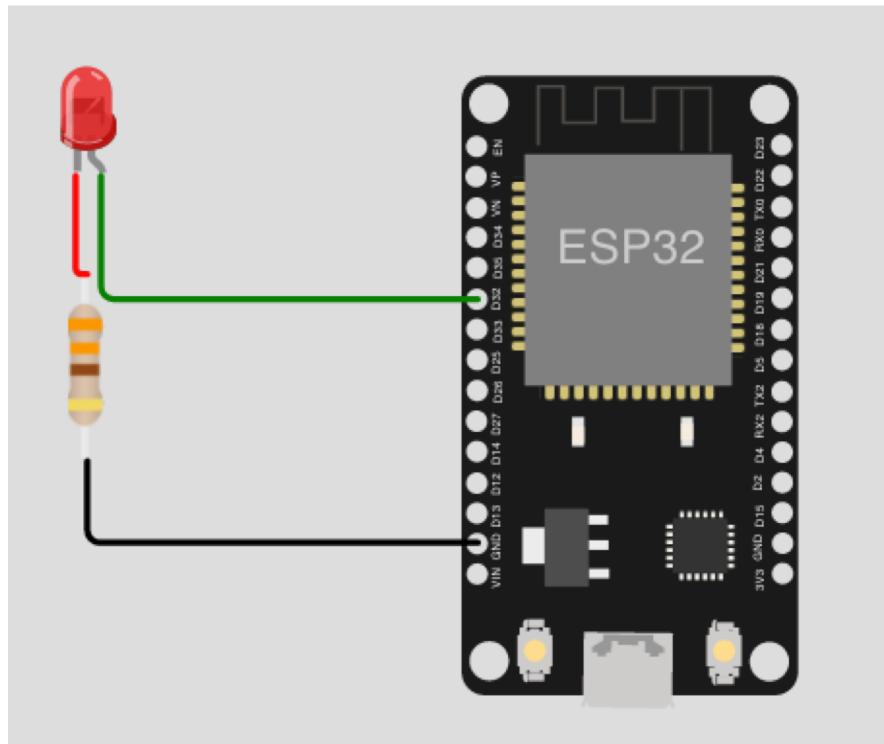


Pisca LED

Saída Digital

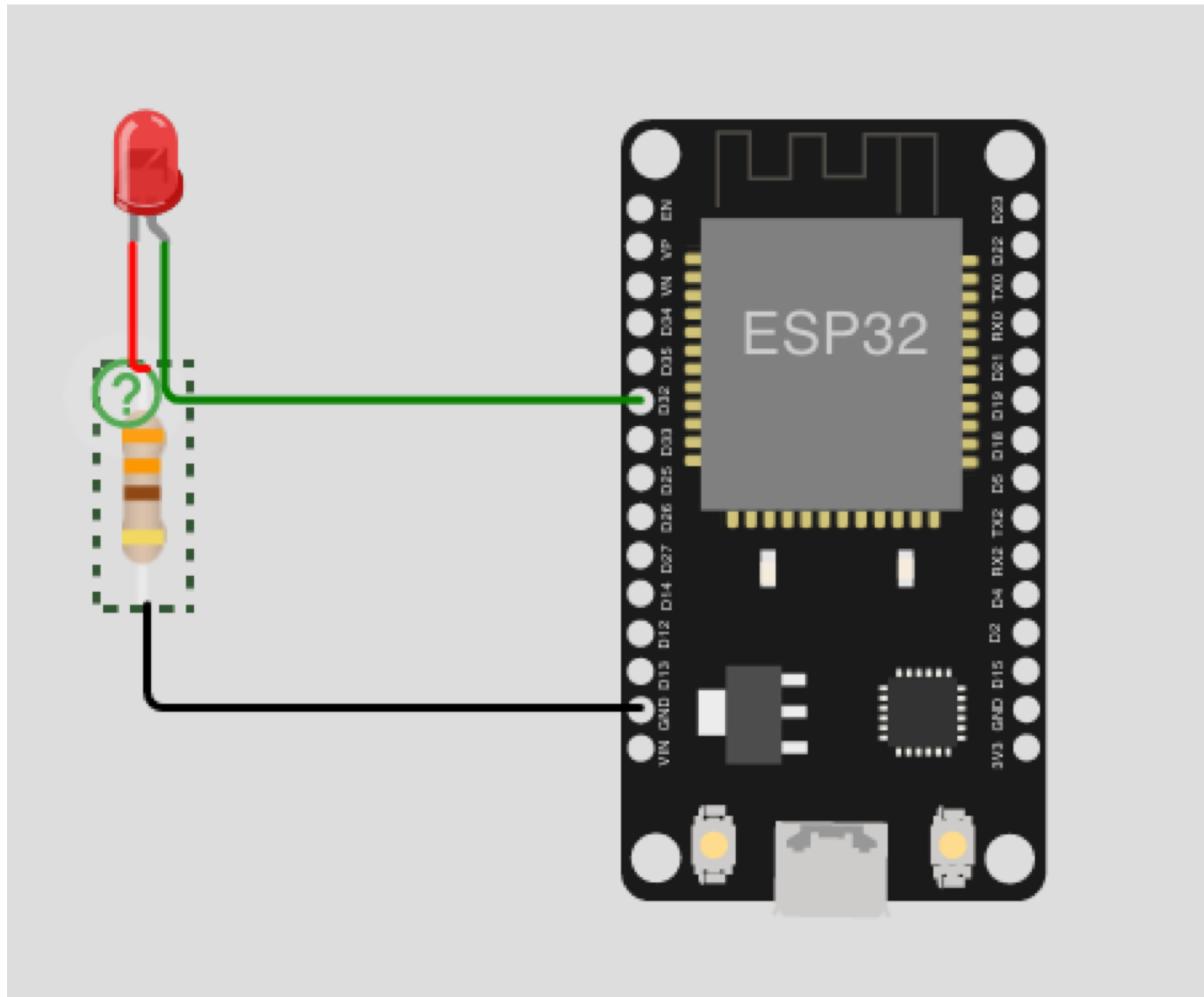
# Círculo LED

Construa o círculo abaixo



Anodo (+) do LED (Pino D32)  
Catodo (-) do LED (Term. Resistor)  
Term. Resistor (Pino GND)

# Documentação (Ajuda)



# Documentação (Ajuda)



Wokwi

Docs

Ainda Português (Brasil) ▾

Blog ↗

Simulador ↗



Procurar

Começando



Guias

Referência do Diagrama

diagram.json

board-franzininiho-wifi

board-ssd1306

wokwi-74hc165

wokwi-74hc595

wokwi-7segment

wokwi-a4988

wokwi-analog-joystick

wokwi-arduino-mega

wokwi-arduino-nano

wokwi-arduino-uno

wokwi-attiny85

wokwi-buzzer

wokwi-dht22

wokwi-dip-switch-8

wokwi-ds1307

wokwi-franzininho

## Referência do wokwi-resistor

Um resistor.



### ALERTA

Wokwi tem apenas uma simulação de circuito analógico muito básica. Você não será capaz de usar resistores junto com componentes analógicos (por exemplo, potenciômetro ou sensor de temperatura NTC). Você ainda pode usar os resistores como pull-up/pull-down externos.

## Nome dos Pinos

Pino	Descrição
1	Primeiro pino
2	Segundo pino

## Atributos

Nome	Descrição	Valor padrão

Nome dos Pinos

Atributos

Exemplos

Exemplos no simulador

# Codificação

WOKWI

SAVE SHARE

Docs

sketch.ino

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
```

Diagram

Simulation

Diagram

ESP32

D3

# Codificação

```
#define LED_PIN 32
void setup() {
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_PIN, HIGH);
    delay(500);
    digitalWrite(LED_PIN, LOW);
    delay(500);
}
```

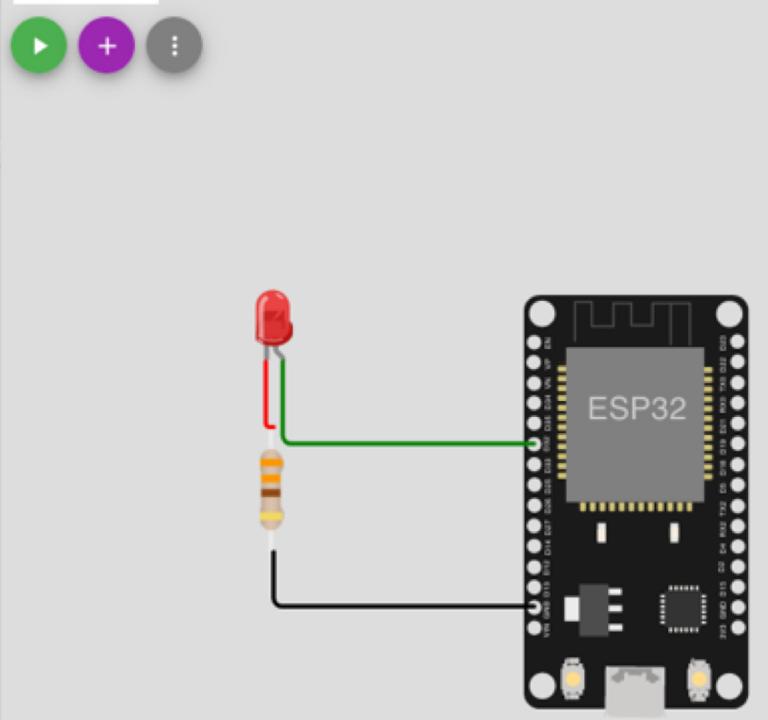
# JSON

WOKWI SAVE SHARE Docs

sketch.ino **diagram.json** Library Manager

Simulation

```
1 {  
2   "version": 1,  
3   "author": "Raimundo Barreto",  
4   "editor": "wokwi",  
5   "parts": [  
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 6, "left": 172.67, "attrs": {} },  
7     { "type": "wokwi-led", "id": "led1", "top": -2.7, "left": 34.83, "attrs": { "color": "red" } },  
8     {  
9       "type": "wokwi-resistor",  
10      "id": "r1",  
11      "top": 89.3,  
12      "left": 23.51,  
13      "rotate": 90,  
14      "attrs": { "value": "330" }  
15    },  
16  ],  
17  "connections": [  
18    [ "esp:TXD", "$serialMonitor:RX", "", [] ],  
19    [ "esp:RXD", "$serialMonitor:TX", "", [] ],  
20    [ "r1:1", "led1:C", "red", [ "h0" ] ],  
21    [ "esp:GND.2", "r1:2", "black", [ "h0" ] ],  
22    [ "led1:A", "esp:D32", "green", [ "v0" ] ]  
23  ]  
24 }
```



# Simulação (HIGH)

WOKWI

SAVE SHARE

Docs

sketch.ino

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
12
```

Simulation

00:16.240 77%

The image shows a screenshot of the Wokwi web-based simulation environment. On the left, the code editor displays a sketch named 'sketch.ino' with the following content:

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
12
```

The right side of the screen shows the simulation interface. At the top, there's a 'Simulation' header with three buttons: a green circle for 'play', a grey square for 'stop', and a grey rectangle with a double vertical bar for 'pause'. Below this is a digital circuit diagram. It features an ESP32 microcontroller at the bottom right. A red LED is connected to pin 32 of the ESP32. A vertical stack of three resistors (brown, orange, yellow) is connected between pin 32 and ground. The LED is currently illuminated, indicating that pin 32 is outputting a HIGH voltage. The simulation status at the top right shows '00:16.240' and '77%'. The overall interface is clean and modern, designed for easy prototyping and testing of microcontroller sketches.

# Simulação (LOW)

WOKWI

SAVE SHARE

Docs

sketch.ino

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
```

Simulation

00:15.657 67%

The screenshot shows the Wokwi simulation interface. On the left, the code for sketch.ino is displayed, defining a constant LED\_PIN as 32 and a loop function that alternates the pin between HIGH and LOW states with a 500ms delay each. On the right, the simulation window shows an ESP32 development board with a red LED connected to its digital pin 32. A 220 ohm resistor is placed in series with the LED. The simulation status bar at the top right indicates the time as 00:15.657 and battery level as 67%.

# Renomear

WOKWI

SAVE SHARE

sketch.ino

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
```

Diagram

Simulation

+

⋮

Raimundo Barreto  
rbarreto@icomp.ufam.edu.br

Discord

My projects

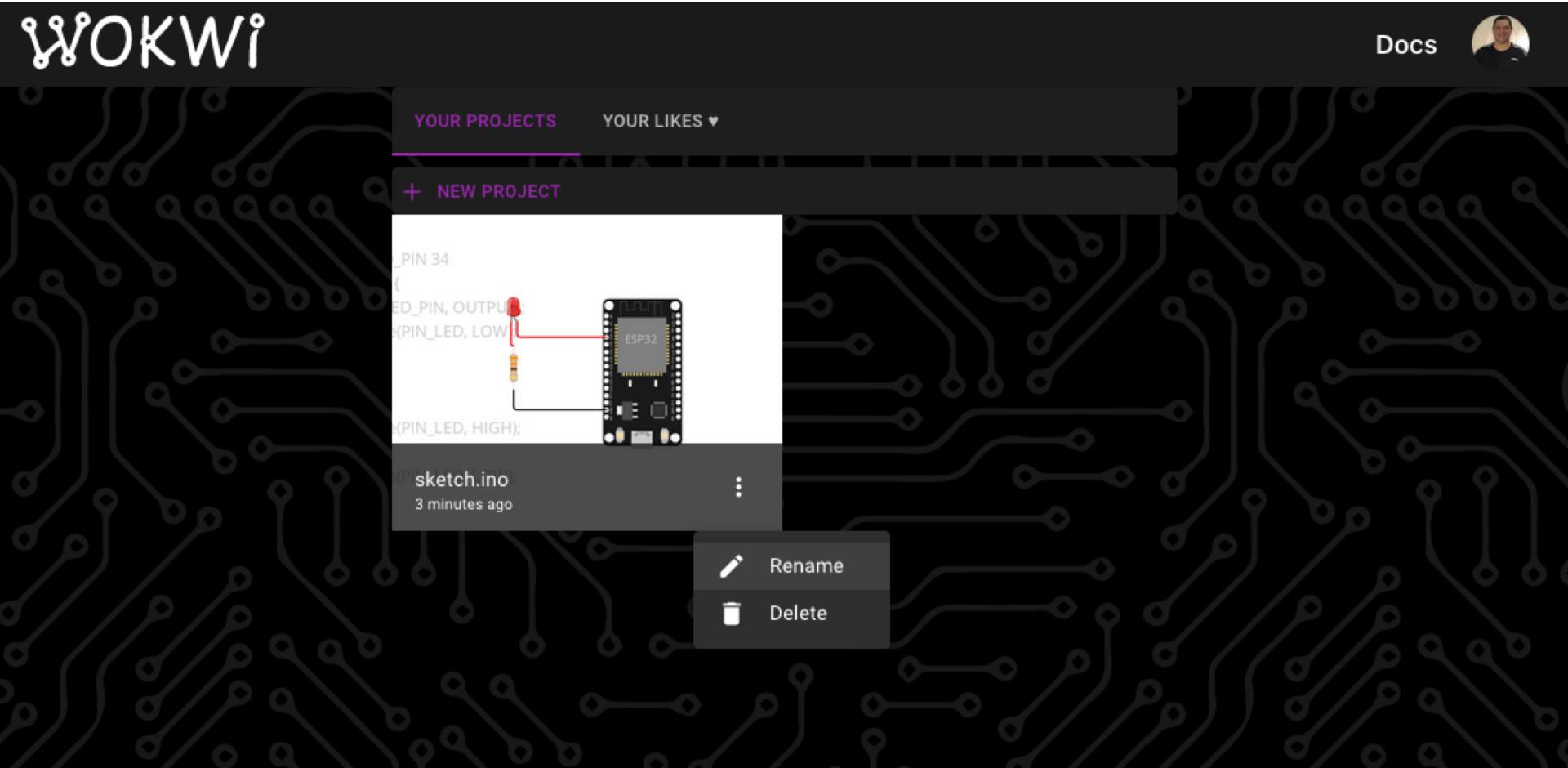
The Club

Feature Roadmap

Logout

ESP32

# Renomear



# Download

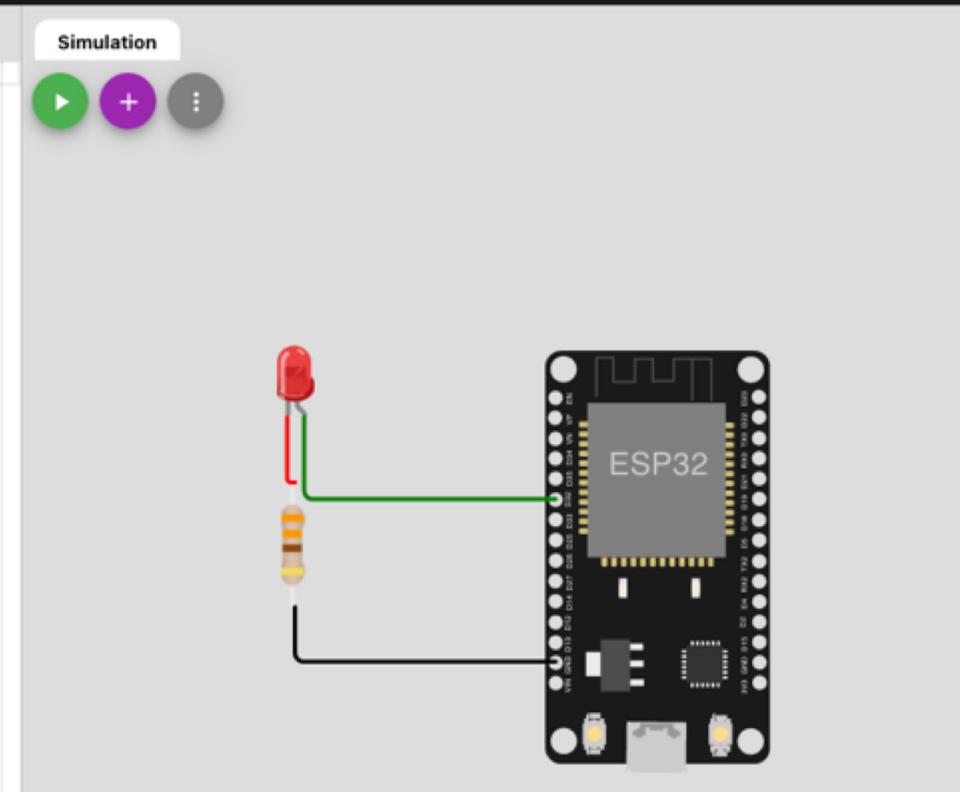
WOKWI | [SAVE](#) [SHARE](#) [SIMULATION](#) Docs

Save a copy  
Lock project  
Unlist project  
**Download project ZIP**

PUT);  
HIGH);  
LOW);  
delay(500);  
}

10  
11  
12

Simulation



# Compartilhando

WOKWI

SAVE SHARE

sketch.ino

```
1 #define LED_PIN 32
2 void setup() {
3     pinMode(LED_PIN, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(LED_PIN, HIGH);
8     delay(500);
9     digitalWrite(LED_PIN, LOW);
10    delay(500);
11 }
```

Diagram

Simulation

Share Project

https://wokwi.com/projects/333751514105381459

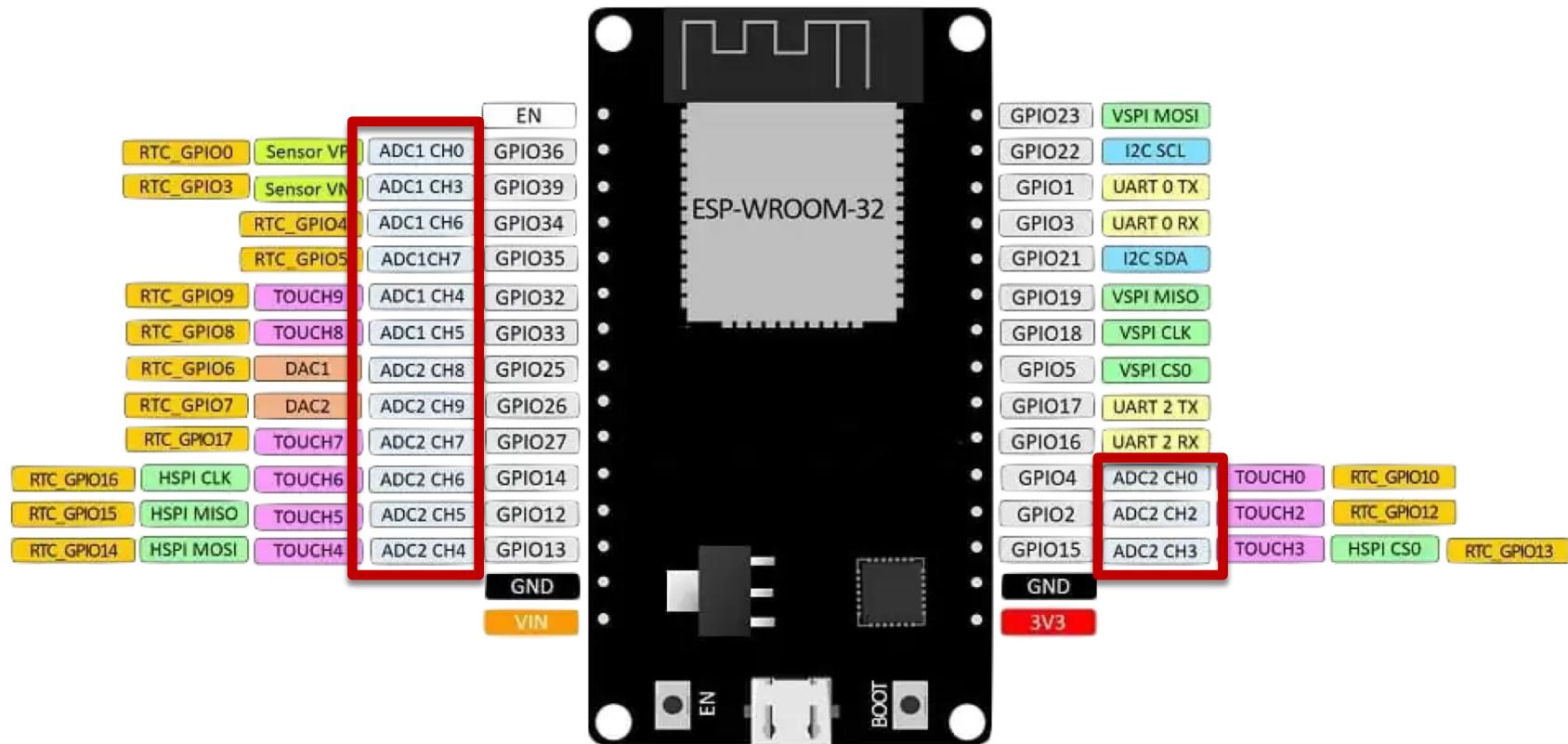
CLOSE

ESP32

# Potenciômetro

Entrada Analógica

# Pinos analógicos (ESP32 DevKit)



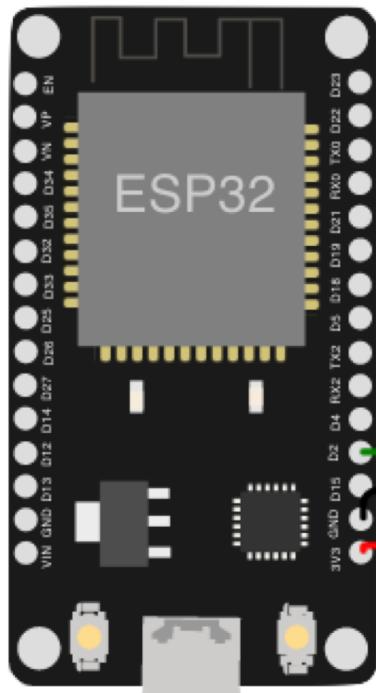
Nem todos os pinos podem ser usados como pinos analógicos

# Entradas Analógicas

- As **entradas analógicas** são úteis para ler valores de sensores analógicos ou potenciômetros
- No ESP32, a ideia é medir **níveis de tensão variáveis** (analogicos) entre 0V e 3,3V
- Como o conversor analógico para digital (ADC) é de **12 bits**, a tensão medida é atribuída a um valor entre 0 e **4095** ( $2^{12}$ ), em que 0 V corresponde a 0 e 3,3 V corresponde a 4095

# Círcito

Construa o círcito abaixo

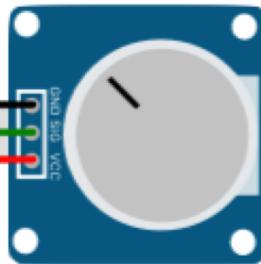


**VCC** (+) do POT (Pino 3V3)

**GND** (-) do POT (GND)

**SIG** do POT (Pino **D2**)

Obs: D2 é um pino que está ligado  
ao ADC2 CH2



# Código

WOKWI

SAVE SHARE

Docs

sketch.ino

```
1 #define POT_PIN 2
2 void setup() {
3     pinMode(POT_PIN, INPUT);
4     Serial.begin(115200);
5 }
6
7 void loop() {
8     Serial.println(analogRead(POT_PIN));
9     delay(10);
10}
11
```

Simulation

# Código

```
#define POT_PIN 2
void setup() {
    pinMode(POT_PIN, INPUT);
    Serial.begin(115200);
}

void loop() {
    Serial.println(analogRead(POT_PIN));
    delay(10);
}
```

# JSON

WOKWI SAVE SHARE Docs Profile

sketch.ino **diagram.json** Library Manager

```
1  {
2    "version": 1,
3    "author": "Raimundo Barreto",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 12.66, "left": 12.66 },
7      { "type": "wokwi-potentiometer", "id": "pot1", "top": 122.46, "left": 240.33, "rotate": 90, "attrs": {} },
8    ],
9    "connections": [
10      [ "esp:TX0", "$serialMonitor:RX", "", [] ],
11      [ "esp:RX0", "$serialMonitor:TX", "", [] ],
12      [ "esp:3V3", "pot1:VCC", "red", [ "v0" ] ],
13      [ "pot1:GND", "esp:GND.1", "black", [ "h0" ] ],
14      [ "pot1:SIG", "esp:D2", "green", [ "h-38.83", "v-19.09", "h-35.52" ] ]
15    ]
16  }
```

Simulation

The screenshot shows the WOKWI web-based simulation environment. At the top, there's a navigation bar with 'WOKWI' logo, 'SAVE', 'SHARE', 'Docs', and a user profile icon. Below the navigation bar, the file tabs are 'sketch.ino' and 'diagram.json'. The 'diagram.json' tab is active, displaying a JSON configuration for a project. The JSON code defines parts (an ESP32 DevKit v1 and a potentiometer) and their connections. In the simulation pane, an ESP32 DevKit v1 board is shown with its pins labeled. A potentiometer component is connected to the board. The connections are: 3V3 to VCC, GND to GND.1, and SIG to D2. The simulation interface includes a play button, a plus sign for adding components, and a three-dot menu.

# Simulação

WOKWI

SAVE SHARE

sketch.ino

```
1 #define POT_PIN 2
2 void setup() {
3     pinMode(POT_PIN, INPUT);
4     Serial.begin(115200);
5 }
6
7 void loop() {
8     Serial.println(analogRead(POT_PIN));
9     delay(10);
10}
```

Diagram.json

Library Manager

Simulation

00:10.265 56%

Docs

ESP32

1545  
1545  
1545  
1545  
1545  
1545  
1545