

Prática 01:

Programando o microcontrolador ESP32

Disciplina: **Introdução à Internet das Coisas - IMD0902**

Prof. Heitor Florencio

Prof. Leonardo Augusto

Aula:
**Prática 01: Programando o
microcontrolador ESP32**

Tópicos

- Placas ESP32
 - ESP-WROOM-32
 - IDE Arduino para ESP's
 - Experimento 01: Acionar LED interno (GPIO2)
 - Experimento 02: Acionar LED externo
-



SoCs

All

ESP32-S

ESP32-S2

ESP32-S3

ESP32-C

ESP32-C2

ESP32-C3

ESP32-C6

ESP32

ESP8266



Modules

All

ESP32-S

ESP32-S2

ESP32-S3

ESP32-C

ESP32-C2

ESP32-C3

ESP32-C6

ESP32

ESP8266



DevKits

All

ESP32-DevKitC

ESP-EYE

ESP Audio DevKits

ESP32-GoogleCloud IoT Kit

ESP32-Azure IoT Kit



SDKs

IoT Development Framework

ESP8266_RTOS_SDK

ESP HomeKit SDK

Audio Development Framework

Mesh Development Framework

ESP32 for Arduino

ESP-AT



ESP SoC & Module Selector

A família de microcontroladores (μ C) ESP

- **Classificado como SOC (system on a chip) que integra:**

- Wi-Fi (banda 2.4 GHz);
- Bluetooth;
- CPU;
- GPIO (General Purpose Input/Output)



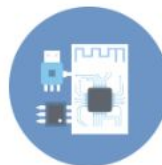
Wi-Fi & Bluetooth Connectivity

This minimum-system development board is powered by an ESP32 module. It integrates Wi-Fi and Bluetooth functions, and provides a rich peripheral set for rapid prototyping!



Rapid Prototyping

ESP32-DevKitC achieves optimal RF performance. You can get right into application design and development, without worrying about RF performance and antenna design. ESP32-DevKitC has your basic system-requirements already covered. Just plug in the USB cable and you are ready to go!



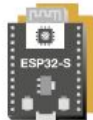
Flexible and Feature-Rich

ESP32-DevKitC contains the entire support circuitry of ESP32-WROOM series, ESP32-WROVER series, and ESP32-SOLO series of modules, also including a USB-UART bridge, reset- and boot-mode buttons, an LDO regulator and a micro-USB connector. Every important GPIO is available to the developer.



Breadboard-Friendly

The ESP32-DevKitC pinout is optimized to enable prototyping on a breadboard. The on-board LDO output is led out for powering up additional off-board electronics. Peripheral outputs are grouped together for hassle-free prototyping.



ESP32-S Series

ESP32-S2 Series

32-bit MCU & 2.4 GHz Wi-Fi

- PC connectivity: USB
- SDK: [ESP-IDF](#) source code and example applications



ESP32-C Series

ESP32-C2 Series

32-bit RISC-V MCU & 2.4 GHz Wi-Fi & Bluetooth LE 5 (LE)

- PC connectivity: USB
- SDK: [ESP-IDF](#) source code and example applications



ESP32 Series

ESP32 DevKits

32-bit MCU & 2.4 GHz Wi-Fi & BT/Bluetooth LE

- PC connectivity: USB
- Power supply options: USB (by default), or 5V/GND header pins, or 3V3/GND header pins
- SDK: [ESP-IDF](#) source code and example applications



ESP8266 Series

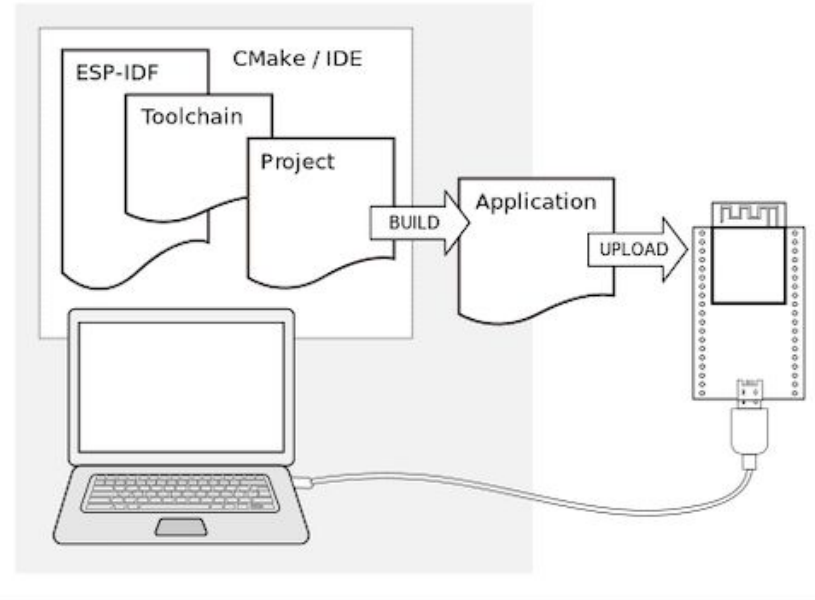
ESP8266 DevKits

32-bit MCU & 2.4 GHz Wi-Fi

- PC connectivity: USB
- SDK: [ESP8266 RTOS](#) SDK source code and example applications

Programando o μ C ESP32: software ESP-IDF

- Existem diversas plataformas para programar os microcontroladores ESP's:
 - ESP-IDF (pacotes da própria Espressif)
 - IDE Arduino (linguagem C)
 - Espruino (Javascript);



Programando o μ C ESP32: IDE Arduino



ARDUINO IDE



Last Minute
ENGINEERS.com



ESP32





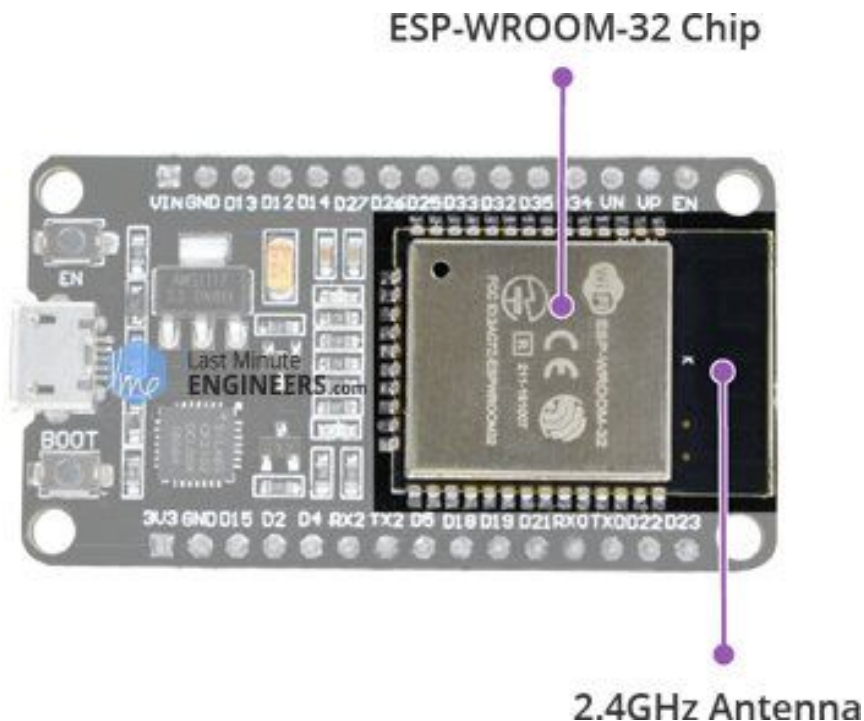
Placa de desenvolvimento:
ESP32-DevKit

Módulo: **ESP-WROOM-32**

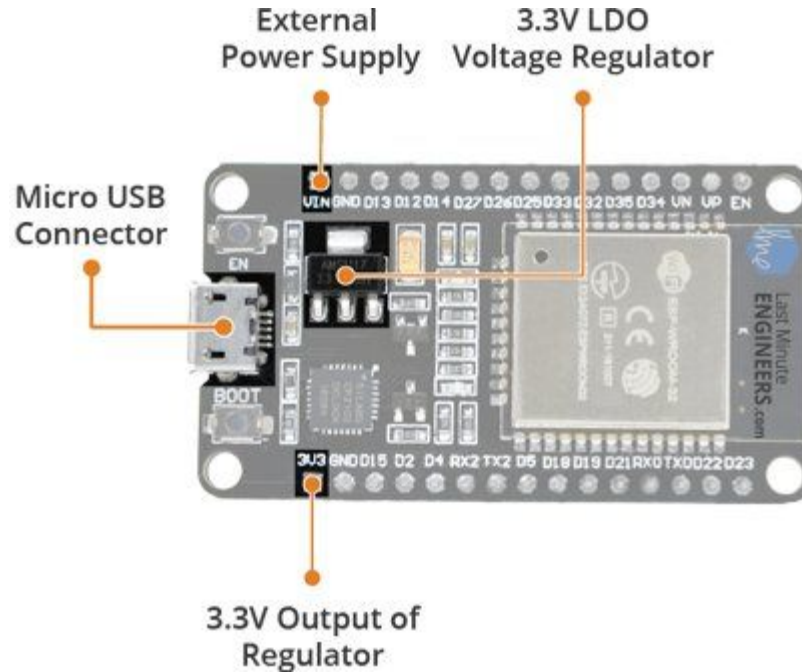
ESP-WROOM-32

Módulo ESP-WROOM-32:

- Microprocessador Tensilica Xtensa® Dual-Core 32-bit LX6
- Frequência de clock até 240 MHz
- RAM interna de 520 kB
- Flash de 4MB
- 802.11b/g/n Wi-Fi
- Bluetooth 4.2/BLE



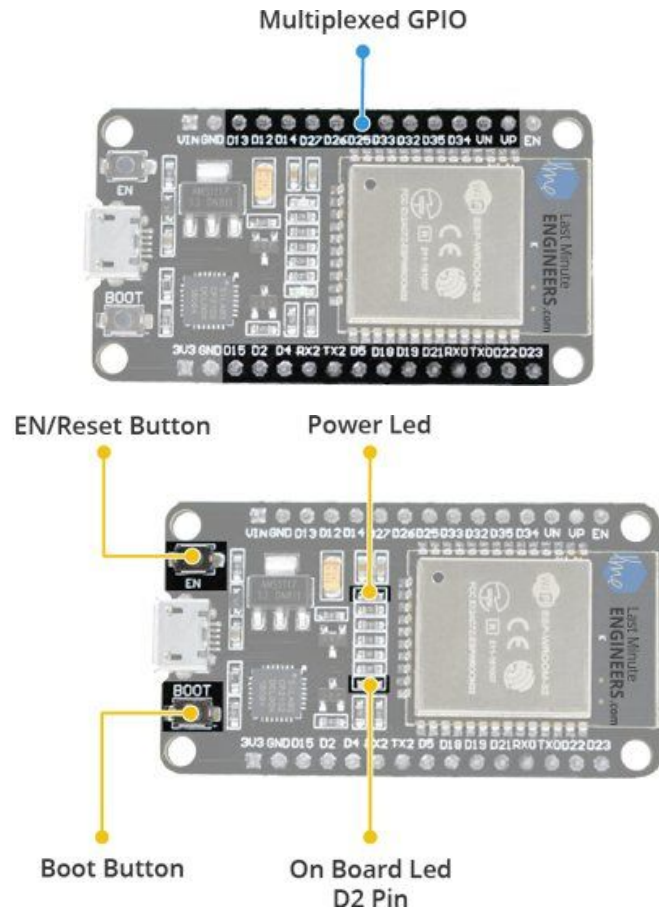
Alimentação do ESP-WROOM-32



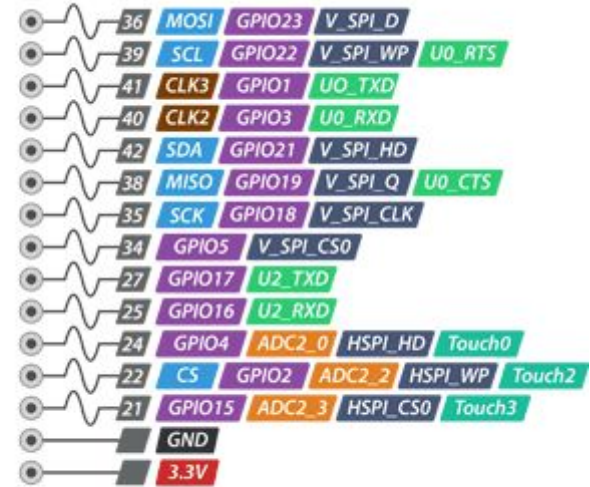
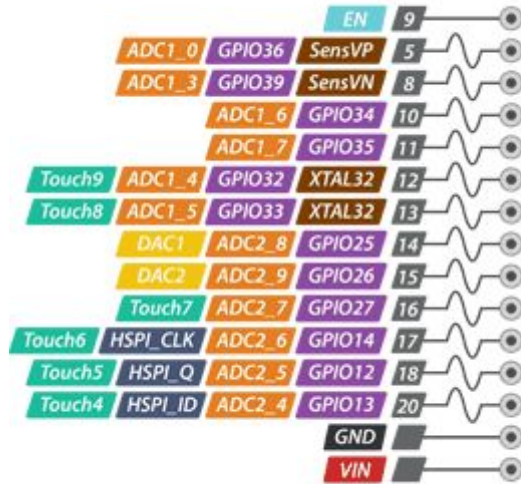
Interfaces do ESP-WROOM-32

Pinos GPIO:

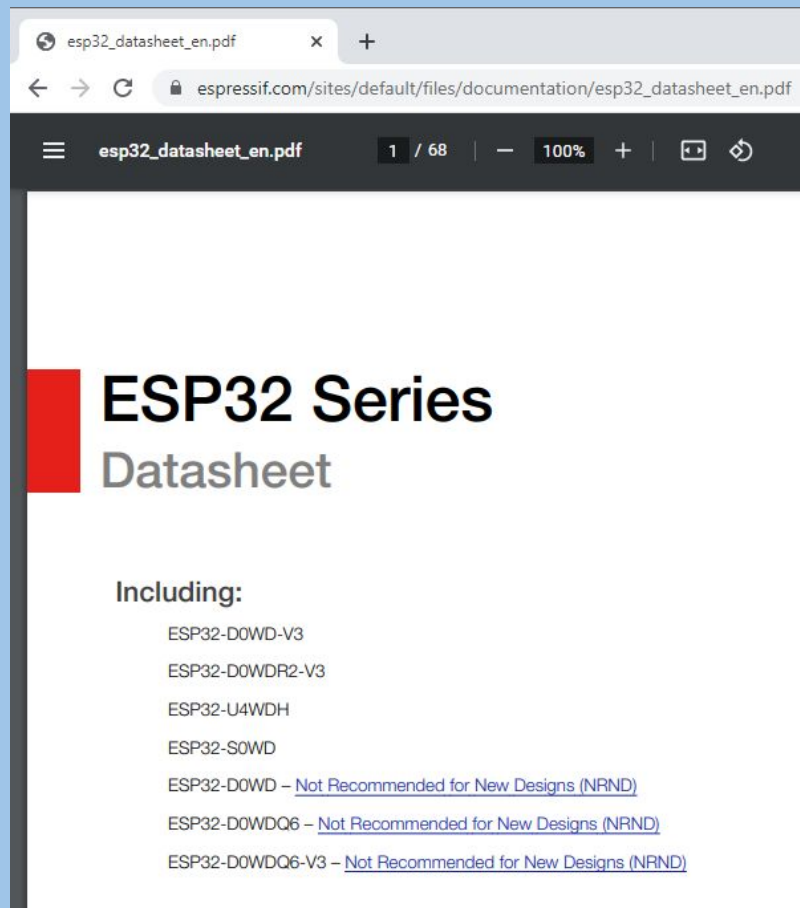
- 15 pinos ADC:
 - 12-bit; sinais 0-1V, 0-1.4V, 0-2V, or 0-4V;
- 2 pinos UART;
- 25 saídas PWM;
- 2 pinos DAC;
- Interfaces SPI, I2C e I2S;
- Pinos para touch pads;



ESP-WROOM-32: pinout



Mais informações no Datasheet





Ambiente de desenvolvimento:
Arduino IDE

Download IDE

- <https://www.arduino.cc/en/software>



Arduino IDE 2.0.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits

Windows MSI installer

Windows ZIP file

Linux AppImage 64 bits (X86-64)

Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits

macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

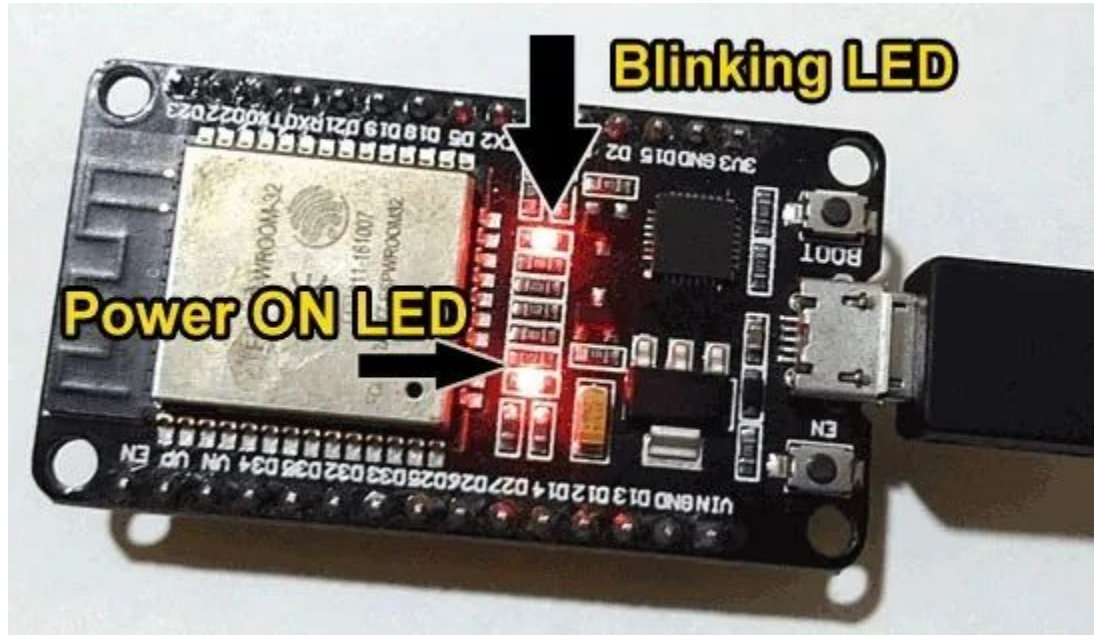
Instalação do pacote ESP32

- Abra a Arduino IDE e acesse Arquivos -> Preferências;
- Cole em “URLs adicionais para Gerenciadores de Placas” a seguinte URL:
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
- Clique em OK;
- Vá agora no menu Ferramentas > Placa > Gerenciador de Placas:
- Ao abrir, procure por ESP32, e clique em instalar:
- Aguarde a instalação e clique depois em fechar.

Experimento 01: Acionar LED interno (GPIO2)

Experimento 01: Acionar LED GPIO2

- Objetivo: Acionar um LED a partir do pino GPIO2 do ESP32.



Experimento 01: Acionar LED GPIO2

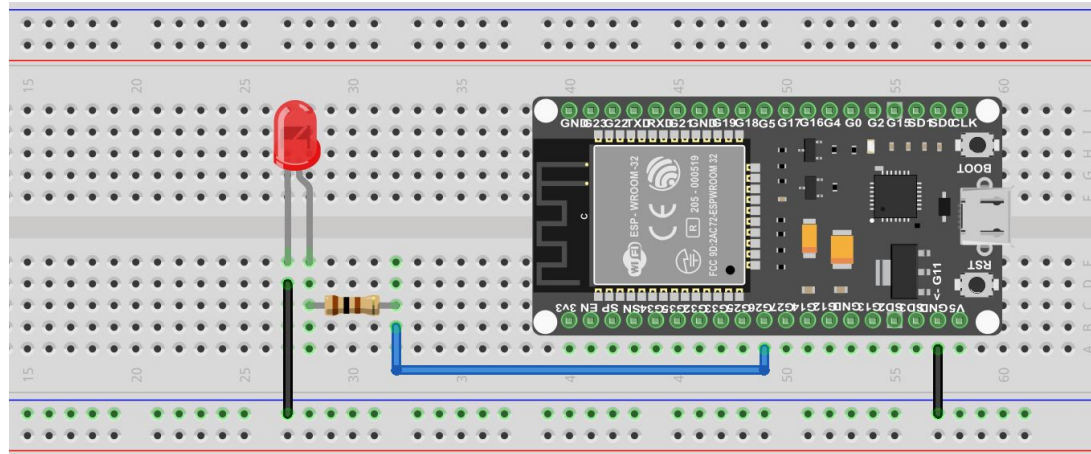
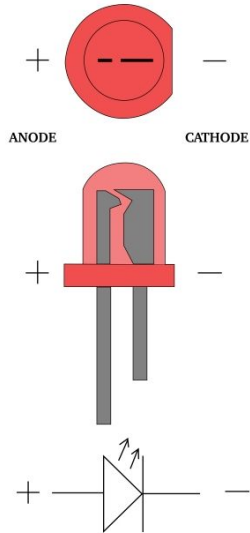
- Código:

```
sketch_pratica01_aciona-led$  
  
int pinLED = 2;  
  
void setup() {  
    pinMode(pinLED, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(pinLED, HIGH);  
    delay(200);  
    digitalWrite(pinLED, LOW);  
    delay(200);  
}
```

Experimento 02: Acionar LED externo

Experimento 01: Acionar LED

- Objetivo: Acionar um LED a partir do pino GPIO26 do ESP32.
- Requisitos funcionais:
 - O microcontrolador ESP32 deve manter o LED ativado por 100 ms e, em seguida, desativado por 100 ms. Esse ciclo de ativação e desativação deve ser repetido.



Experimento 01: Acionar LED

- Código:



```
sketch_pratica01_aciona-led | Arduino 1.8.13
Arquivo  Editar  Sketch  Ferramentas  Ajuda

sketch_pratica01_aciona-led

int led = 26;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);
}
```

Dúvidas?

Prof Heitor Florencio
IMD/UFRN
Sala 103 - nPITI/IMD
heitorm@imd.ufrn.br
