

## **INTRODUÇÃO AO ESP32 – WROOM-32U**

O **ESP32-WROOM-32U** é um módulo de desenvolvimento baseado no microcontrolador **ESP32** , fabricado pela **Espressif Systems** . Ele é uma variação do **ESP32-WROOM-32** , com a principal diferença sendo o conector para antena externa, o que melhora a recepção de sinais Wi-Fi e Bluetooth em aplicações críticas.

### **PRINCIPAIS APLICAÇÕES:**

- Dispositivos IoT (Internet das Coisas)
- Automação residencial e industrial
- Sensoriamento remoto e comunicação sem fio
- Robótica e projetos embarcados avançados

### **CARACTERÍSTICAS:**

- Módulo ESP32 Wifi e Bluetooth ESP-WROOM-32U
- Compatível com a IDE do Arduino
- Conectividade WiFi e conexão Bluetooth
- Aplicação em projetos eletrônicos
- Ideal para projetos da Internet das Coisas IOT

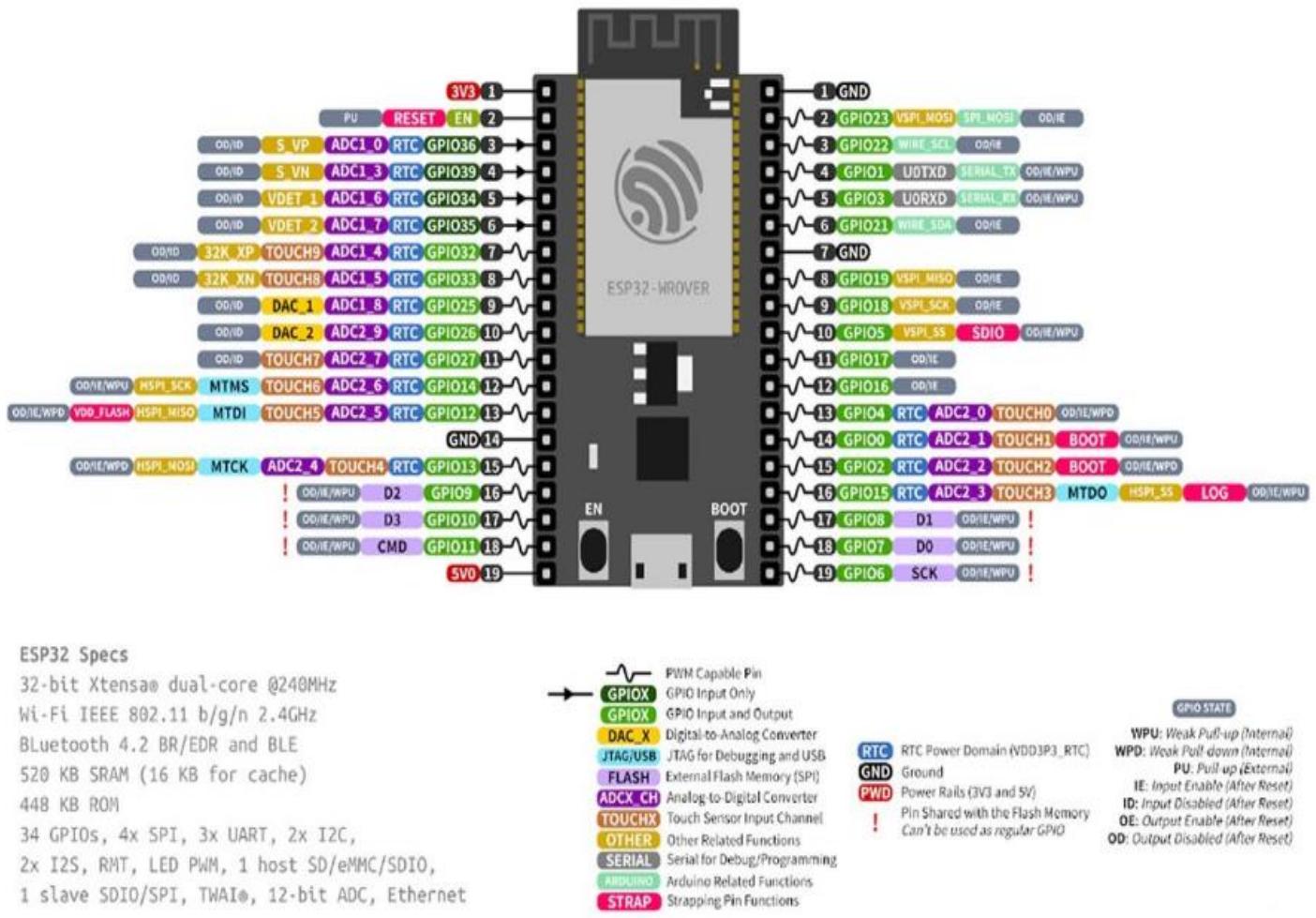
### **ESPECIFICAÇÕES:**

- Módulo: ESP-WROOM-32U;
- Processador: Xtensa 32-Bit LX6 Dual Core;
- Memória SRAM: 520Kb;
- Memória Flash Externa: 32-Bit de acesso e 4Mb;
- Tensão de Alimentação: 2,7 à 3,6 VDC;
- Tensão de nível lógico: 3,3VDC (não tolera 5V);
- Corrente de consumo: 80mA (típica);
- Corrente de consumo: 500mA (máxima);
- WiFi: 802.11 b/g/n: 2.4 à 2.5 GHz;
- Segurança WiFi: WPA / WPA2 / WPA2-Enterprise / WPS;
- Bluetooth: 4.2 BR / EDR e BLE ( Bluetooth Low Energy);
- Suporte: Arduino IDE, Lua e MicroPython, Código C básico;
- Interfaces: UART, SPI, SDIO, I2C, I2S, IR, PWM;
- Temperatura de trabalho: -40° à +85° C;

### **EXEMPLOS DE APLICAÇÕES:**

- **Automação Residencial:** Controle de luzes, sensores e dispositivos inteligentes.
- **Monitoramento Remoto:** Sensores ambientais, nocivos e telemétricos.
- **Robótica:** Controle de motores e sistemas independentes com comunicação sem fio.

Abaixo uma visão superior das pinagens do ESP32 – WROOM32U. Os rótulos dos pinos estão na parte inferior da placa.



## Instalação e Configuração do Thonny IDE

Para programar o Raspberry Pi Pico usando o Micropython, usaremos o Thonny IDE. Podemos baixar o Thonny no seguinte endereço <https://thonny.org/>. Escolha o seu sistema operacional. Baixe o arquivo e execute o mesmo e siga o procedimento de instalação (use todas as configurações padrão).

Uma vez instalado a IDE Thonny, é necessário instalar o Firmware do Micropython na placa de desenvolvimento Raspbrry PI PICO (**UF2 bootloader**). Para este processo, podemos seguir o tutorial no link a seguir:

<https://randomnerdtutorials.com/getting-started-raspberry-pi-pico-w/>

Ou

<https://www.robocore.net/tutoriais/programacao-raspberry-pi-pico-python>

Uma vez instalado o firmware do micropython na placa esta ficará pronta para uso.

## Executando seu primeiro script

Para familiarizá-lo com o processo de escrever um arquivo e executar código na placa Raspberry Pi Pico, faremos upload de um novo script que simplesmente pisca o LED integrado.

## ESP32

```
from machine import Pin  
from time import sleep  
led = Pin(2, Pin.OUT)  
while True:  
    led.value(not led.value())  
    sleep(0.5)
```

## Executando o script

Para executar esse script na sua placa, basta clicar no ícone executar no Thonny IDE.

O LED começará a piscar.

Para interromper a execução do programa, você pode pressionar o botão STOP ou simplesmente pressionar **CTRL + C**.

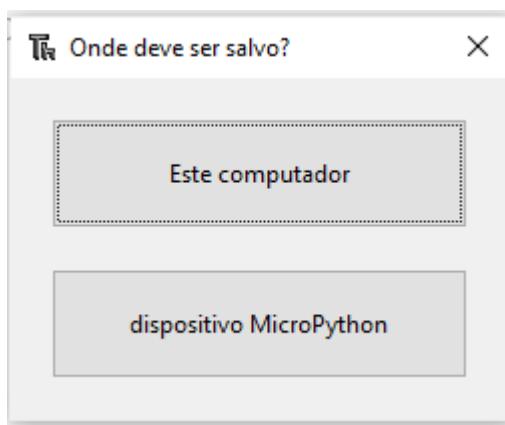


**Nota importante :** apenas executar o arquivo com o Thonny não o copia permanentemente para o sistema de arquivos da placa. Isso significa que se você o desconectar do seu computador e aplicar energia à placa, nada acontecerá porque ela não tem nenhum arquivo Python salvo em seu sistema de arquivos. A função Thonny IDE *Run* é útil para testar o código, mas se você quiser carregá-lo permanentemente para sua placa, você precisa criar e salvar um arquivo no sistema de arquivos da placa.

## Salvando o script na placa ESP32

Pare a execução do programa anterior clicando no botão **Parar**, caso ainda não o tenha feito.

Com o código copiado para o arquivo, clique no ícone **Save**. Então, selecione dispositivo MicroPython.



Salve o arquivo com o seguinte nome: main.py.

**Nota :** Quando você nomeia um arquivo main.py, o ESP32 executará esse arquivo automaticamente na inicialização. Se você o chamar de um nome diferente, ele ainda será salvo no sistema de arquivos da placa, mas não será executado automaticamente na inicialização.

Por fim, clique em **OK** para prosseguir.

Agora, você pode remover e aplicar energia novamente à placa, ou pode até mesmo alimentá-la usando uma fonte de alimentação diferente que não seja seu computador. Você notará que a placa começará a piscar o LED automaticamente quando iniciar.