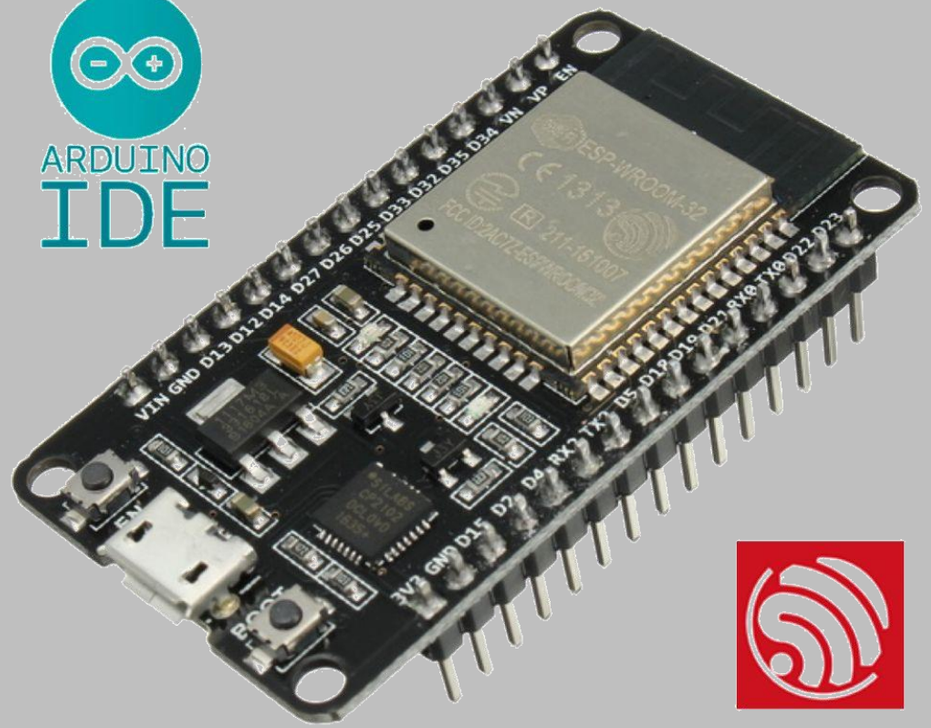
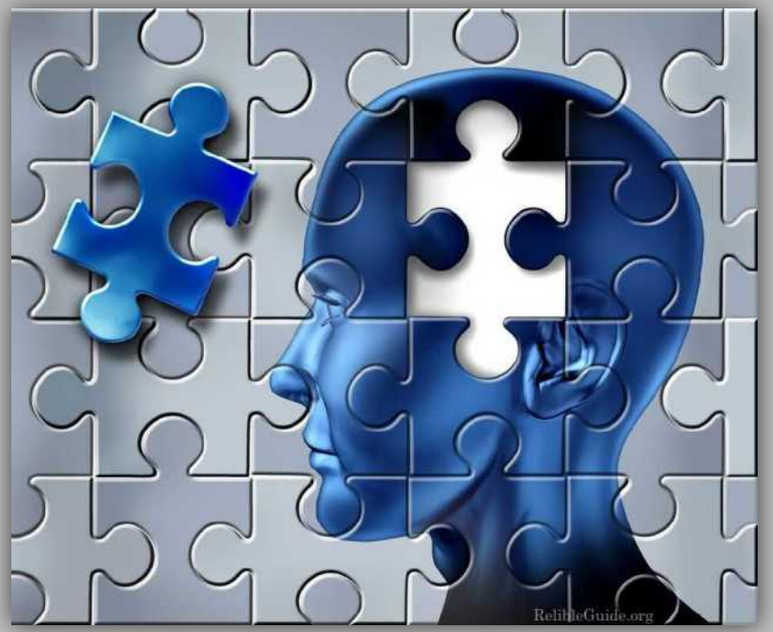
ESP32: Detalhes internos e pinagem





Por Fernando Koyanagi

Intenção dessa aula

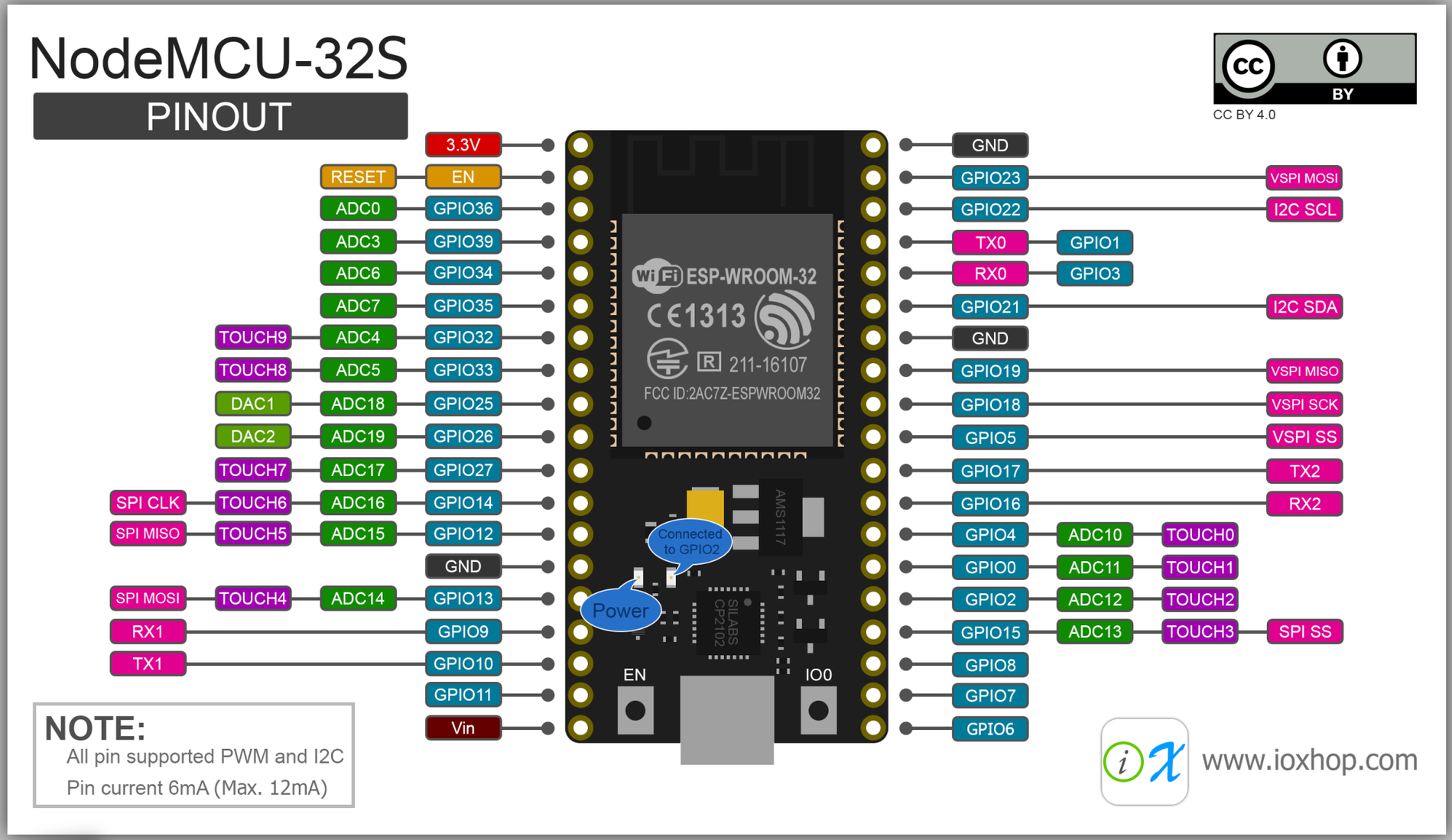
1. Aprender qual a identificação correta dos pinos olhando o datasheet

2. Informar quais os pinos funcionam como OUTPUT/INPUT

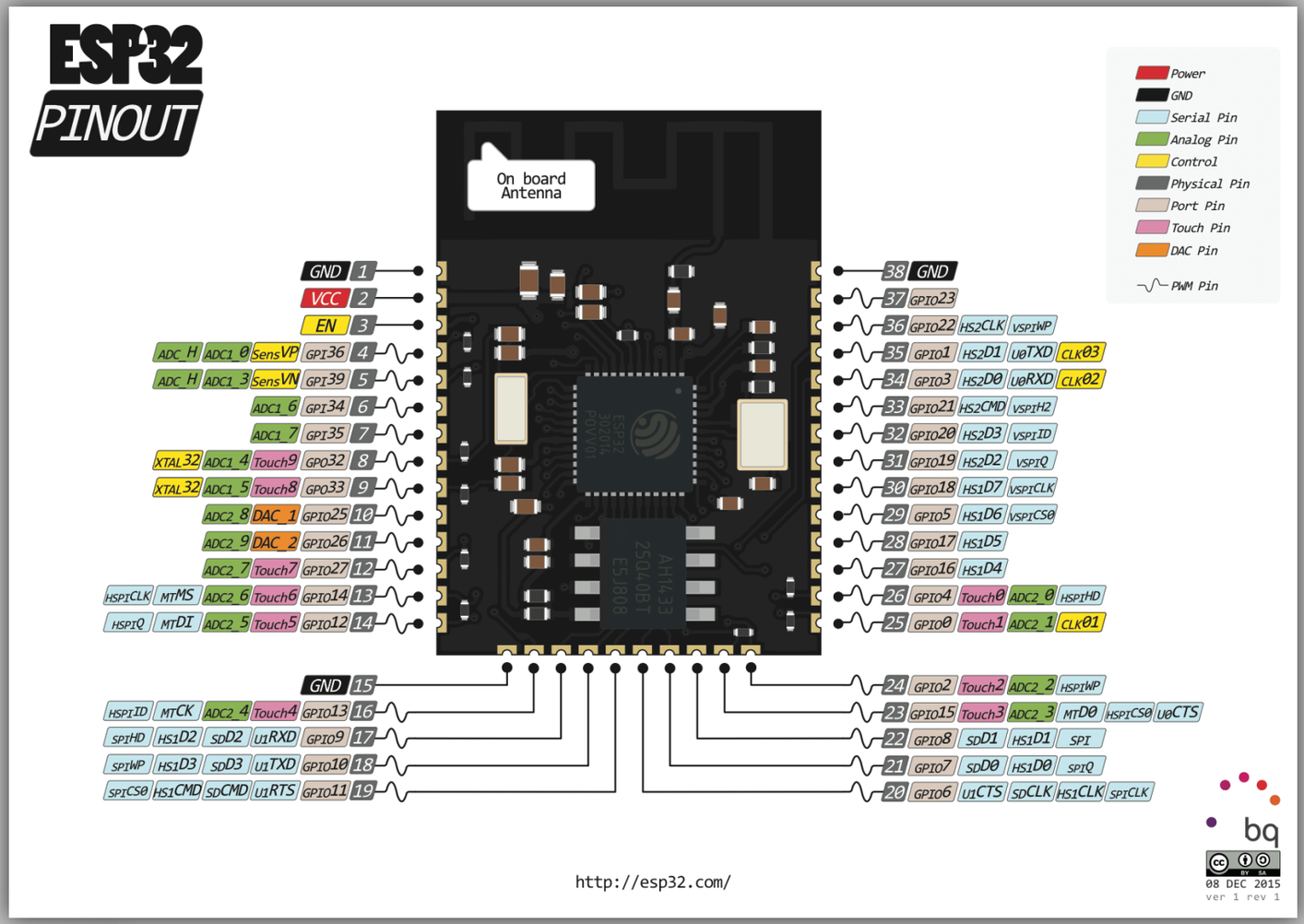
3. Ter uma visão geral sobre os sensores e periféricos que o ESP32 nos oferece

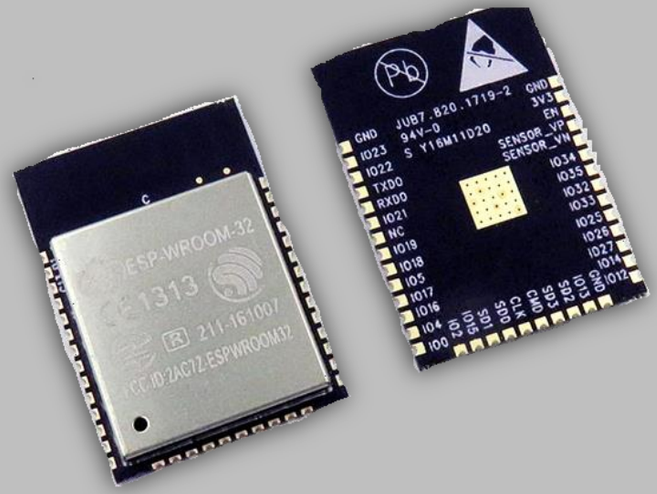
4. Explicar sobre o boot

NodeMCU ESP-WROOM-32

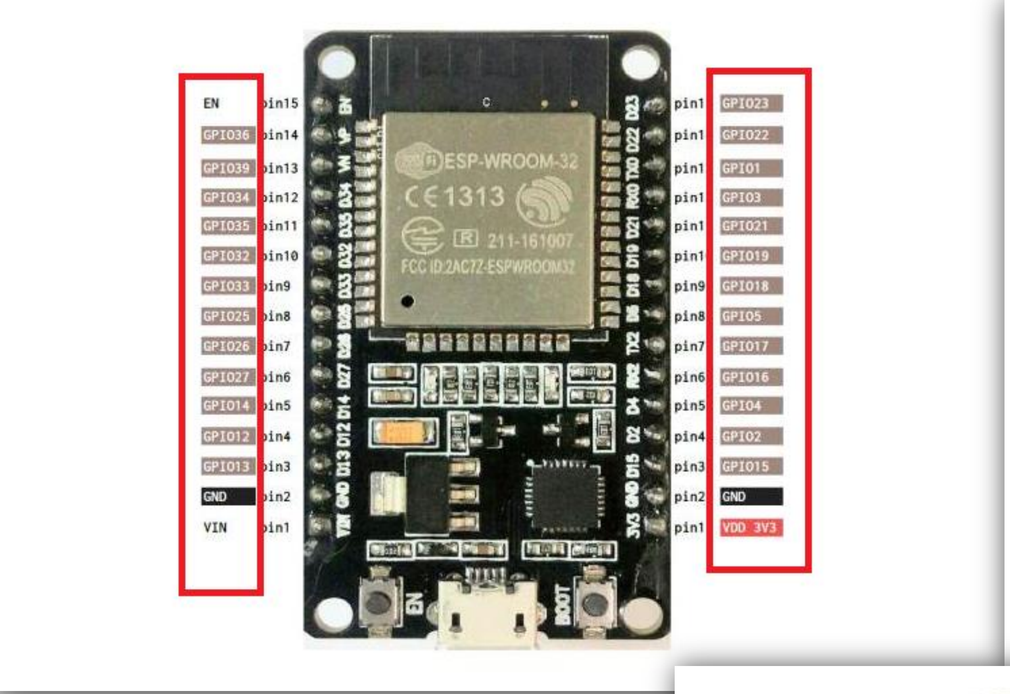


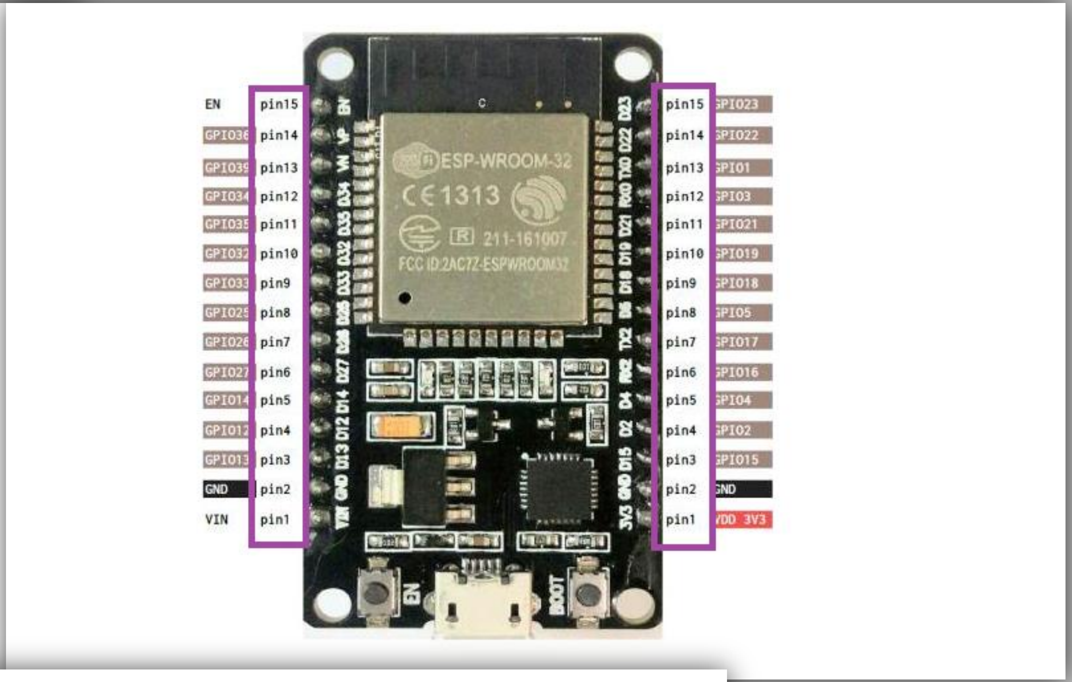
ESP-WROOM-32

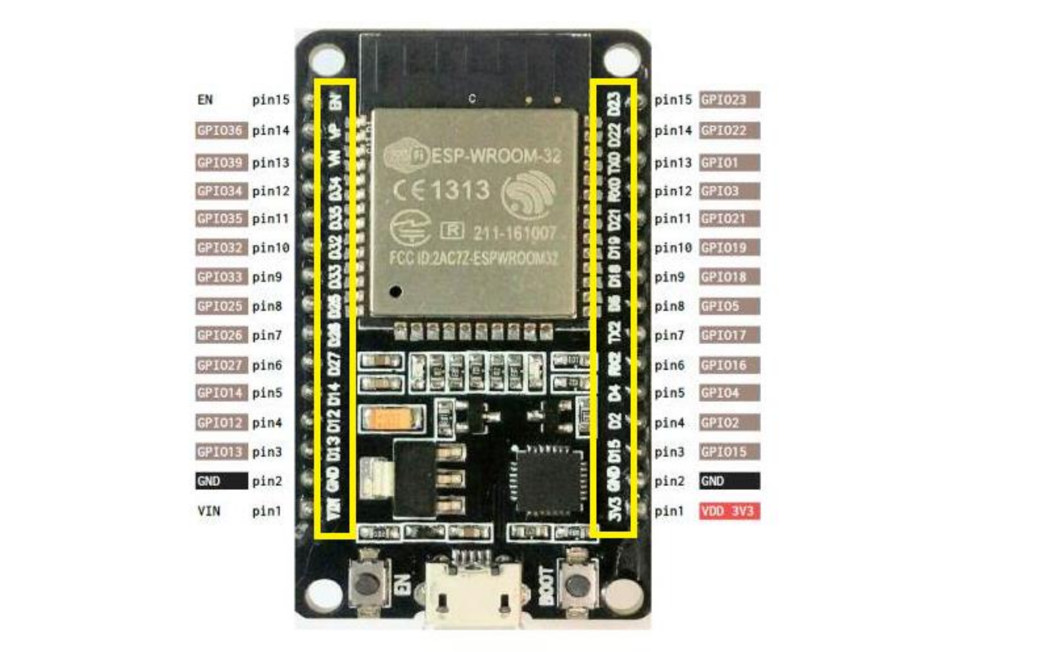

ESP32
PINOUT
On board Antenna
Power
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin
PWM Pin
GND 1
VCC 2
EN 3
ADC_H ADC1_0 SensVP GPI36 4
ADC_H ADC1_3 SensVN GPI39 5
ADC1_6 GPI34 6
ADC1_7 GPI35 7
XTAL32 ADC1_4 Touch9 GPO32 8
XTAL32 ADC1_5 Touch8 GPO33 9
ADC2_8 DAC_1 GPIO25 10
ADC2_9 DAC_2 GPIO26 11
ADC2_7 Touch7 GPIO27 12
HSPICLK MTMS ADC2_6 Touch6 GPIO14 13
HSPIQ MTDI ADC2_5 Touch5 GPIO12 14
GND 15
HSPID MTCK ADC2_4 Touch4 GPIO13 16
SPIHD HS1D2 SDD2 U1RXD GPIO9 17
SPIWP HS1D3 SDD3 U1TXD GPIO10 18
SPICS0 HS1CMD SDCMD U1RTS GPIO11 19
38 GND
37 GPIO23
36 GPIO22 HS2CLK VSPIWP
35 GPIO1 HS2D1 U0TXD CLK03
34 GPIO3 HS2D0 U0RXD CLK02
33 GPIO21 HS2CMD VSPIH2
32 GPIO20 HS2D3 VSPID
31 GPIO19 HS2D2 VSPIQ
30 GPIO18 HS1D7 VSPICLK
29 GPIO5 HS1D6 VSPICS0
28 GPIO17 HS1D5
27 GPIO16 HS1D4
26 GPIO4 Touch0 ADC2_0 HSPIHD
25 GPIO0 Touch1 ADC2_1 CLK01
24 GPIO2 Touch2 ADC2_2 HSPIWP
23 GPIO15 Touch3 ADC2_3 MTD0 HSPICS0 U0CTS
22 GPIO8 SDD1 HS1D1 SPI
21 GPIO7 SDD0 HS1D0 SPIQ
20 GPIO6 U1CTS SDCLK HS1CLK SPICLK
http://esp32.com/
bq
08 DEC 2015
ver 1 rev 1


ESP-WROOM-32
FCC ID:2AC7Z-ESPWROOM32
JUB7.820.1719-2
94V-0
GND
3V3
EN
SENSOR_VP
SENSOR_VN
IO34
IO35
IO32
IO33
IO25
IO26
IO27
IO14
IO12
IO23
IO22
TXD0
RXD0
IO21
NC
IO19
IO18
IO5
IO17
IO16
IO4
IO0

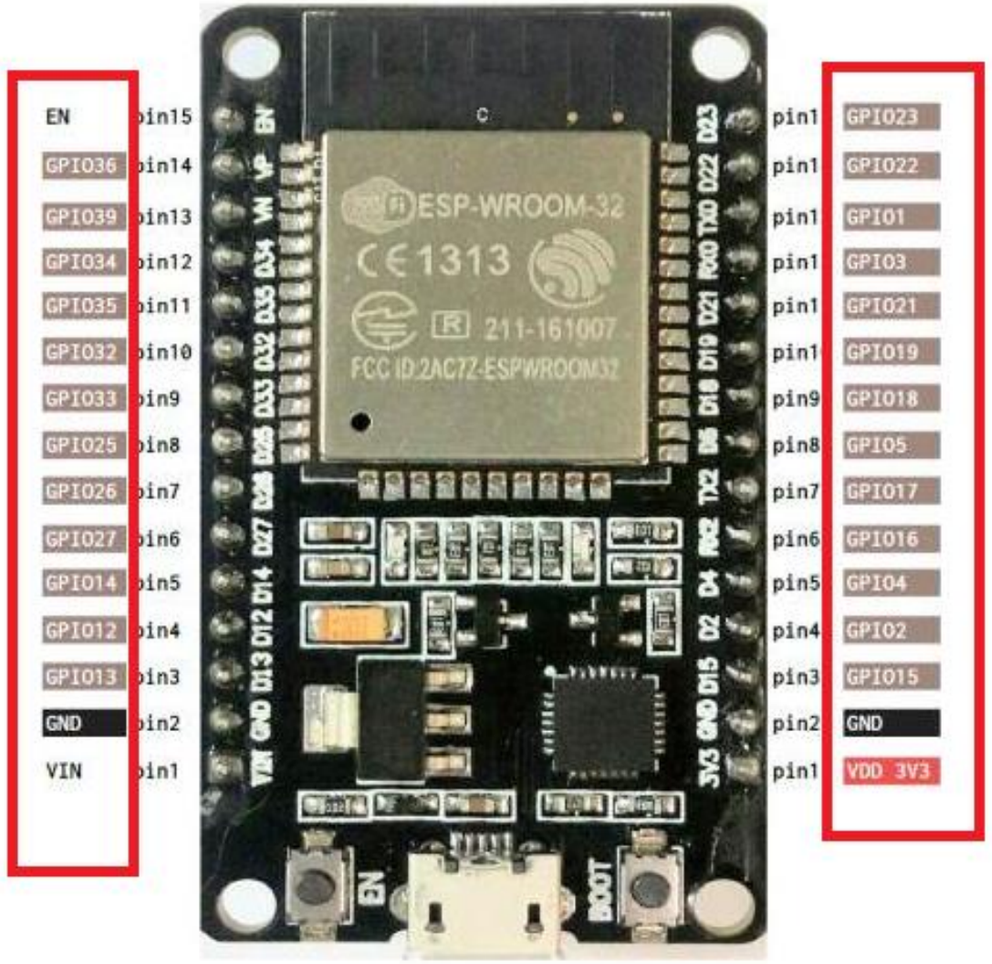
Mas, qual a pinagem correta para eu utilizar meu ESP32?



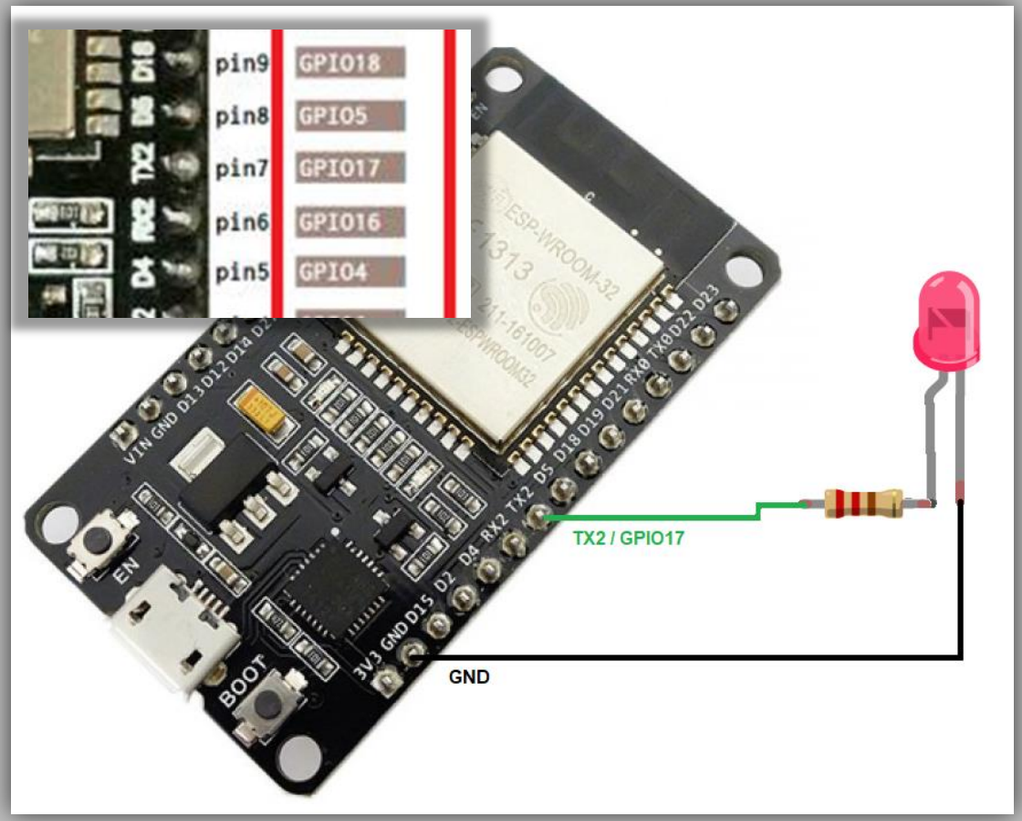




Em um datasheet, são sempre esses destacados a identificação correta dos pinos. Muitas vezes o rótulo no chip não coincide com o número real do pino.



Veja no exemplo uma ligação de um LED no ESP, o modo correto de configurar.




```
const int pinoLED = 17; //pino que o LED foi conectado

void setup() {
   pinMode(pinoLED, OUTPUT); //define o pino 17 como saida
}

void loop() {
  //inverte o estado do LED
  digitalWrite(pinoLED, !digitalRead(pinoLED));
  delay(1000);
}
```

Repare que o rótulo na placa é **TX2**, porém, devemos seguir a identificação correta, como destacado no slide anterior. Portanto, a identificação correta do pino será **17**.

Ex:

- pinMode(17, OUTPUT);
- digitalWrite(17, HIGH);
- digitalWrite(17, LOW);
- digitalRead(17);

INPUT / OUTPUT

Ao realizar testes de INPUT e OUTPUT nos pinos, obtivemos os seguintes resultados:

INPUT não funcionou apenas no **GPIO0**.

OUTPUT não funcionou apenas nos pinos **GPIO34** e **GPIO35**, que são **VDET1** e **VDET2** respectivamente.

*Os pinos VDET, pertencem ao domínio de energia do RTC. Significa que eles podem ser usados como pinos ADC e que o ULP-coprocessador pode lê-los. Podem ser apenas entrada e nunca saídas.

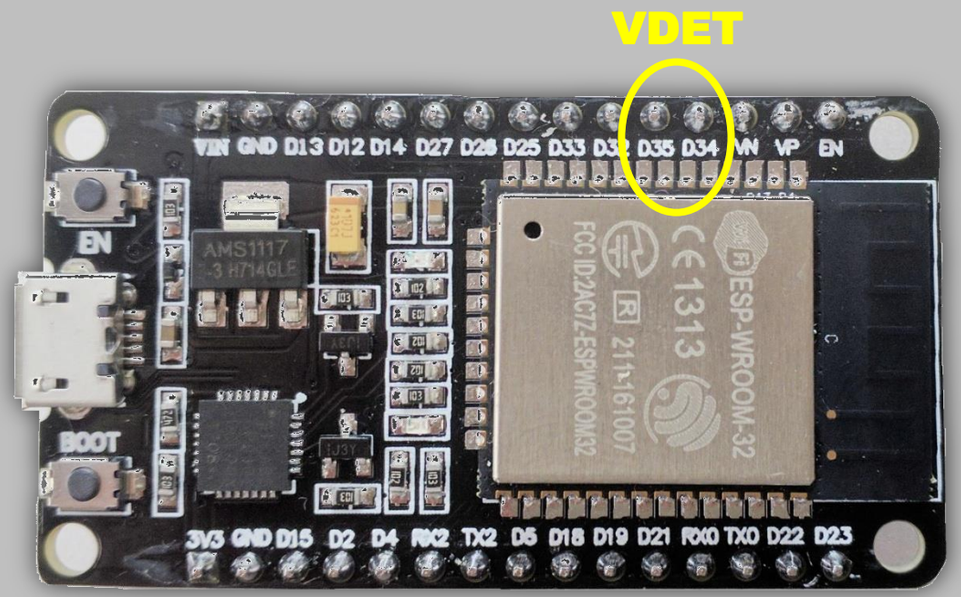
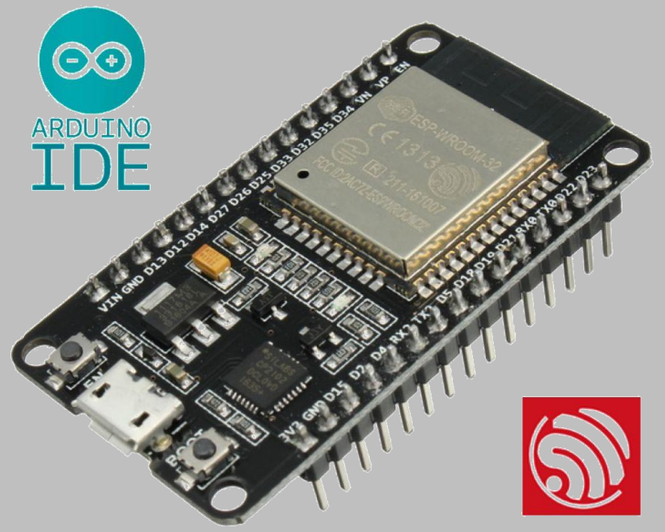
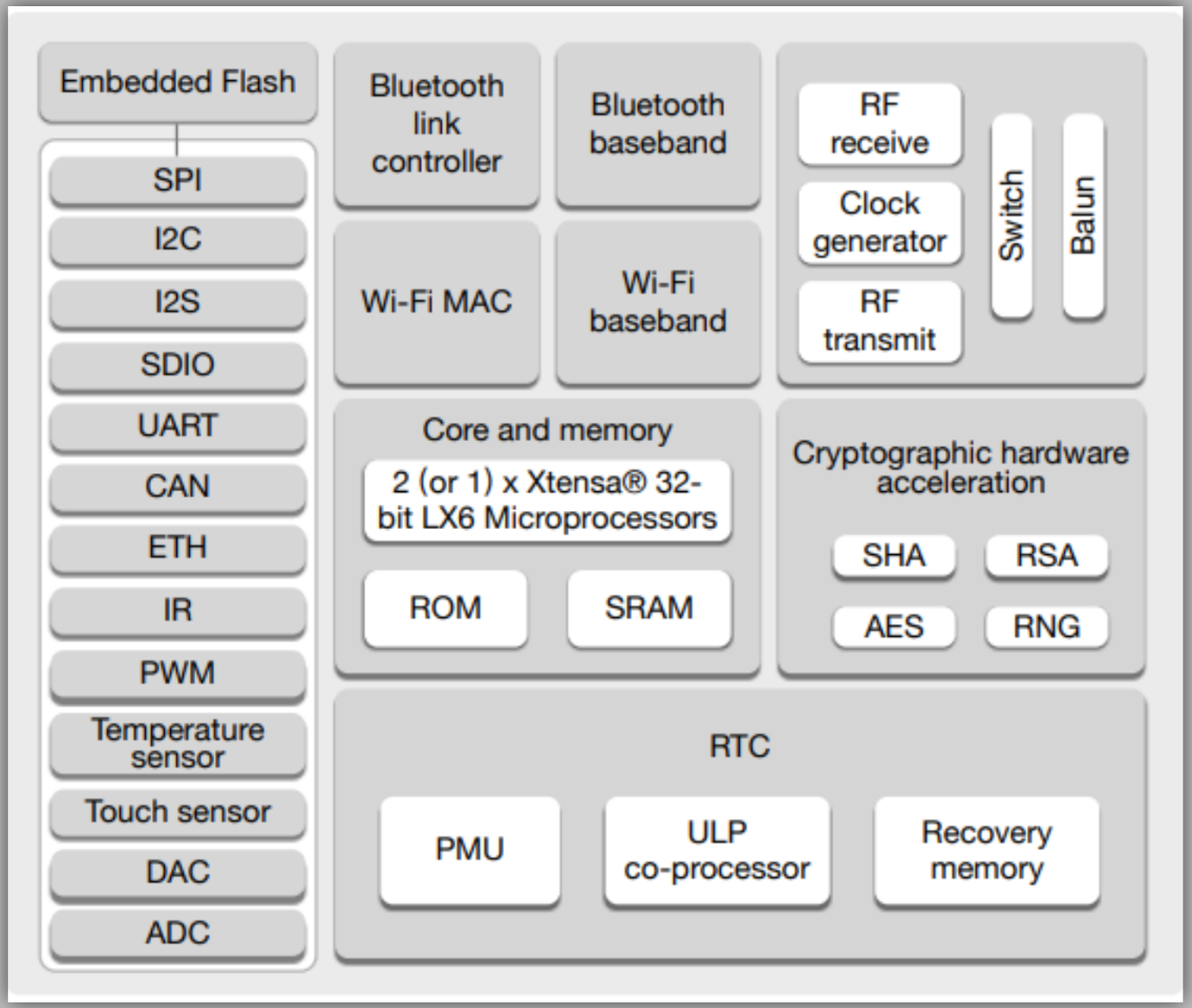
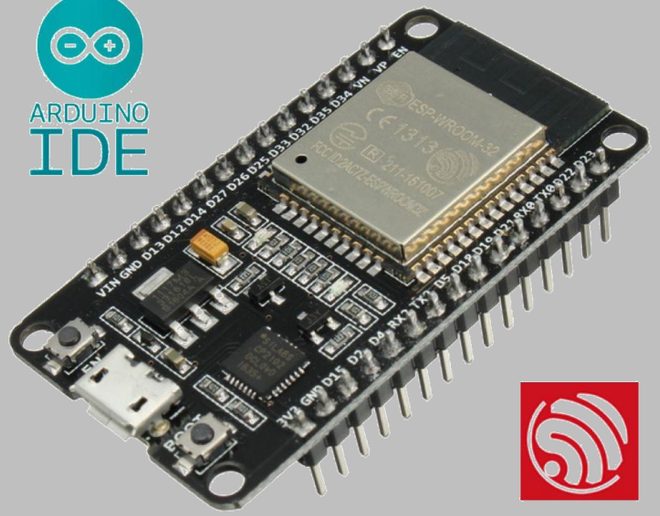


Diagrama de Blocos





Periféricos e Sensores

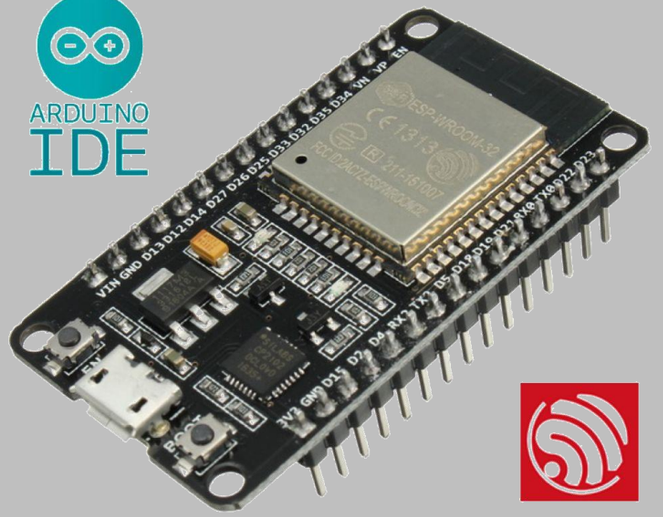


O ESP32 tem 34 pinos GPIO que podem ser atribuídos a várias funções.
Existem vários tipos de GPIOs:

- Digital-only
- Analog-enabled (podem ser configurados como digital)
- Capacitive-touch-enabled (podem ser configurados como digital)
- Entre outros

A maioria dos GPIOs digitais pode ser configurada como pull-up ou pull-down interno, ou configurado para alta impedância. Quando configurados como entrada (input), o valor pode ser lido através do registro.

GPIO



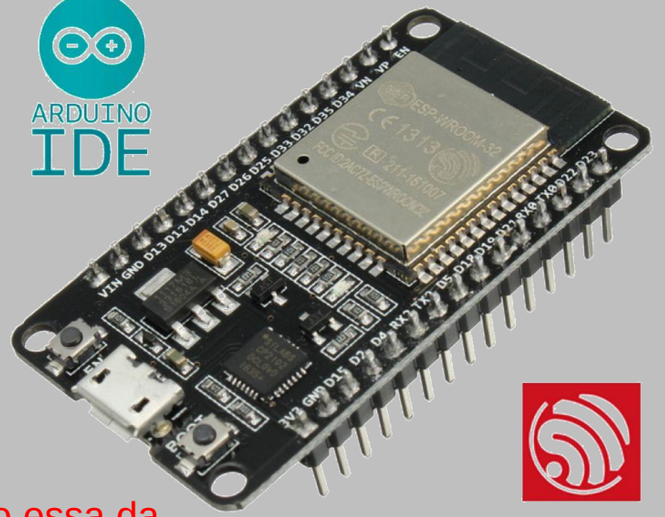
Analog-to-Digital Converter (ADC)

O Esp32 integra ADCs de 12 bits e suporta medições em 18 canais (analog-enabled pins). O ULP-coprocessador no ESP32 também é projetado para medir as tensões enquanto opera em modo sleep, que permite o baixo consumo de energia. A CPU pode ser despertada por uma configuração de limite e/ou através de outros gatilhos.

Digital-to-Analog Converter (DAC)

Dois canais DAC de 8 bits podem ser usados para converter dois sinais digitais em duas saídas de tensão analógica. Estes DAC duplos suportam a fonte de alimentação como referência de tensão de entrada e pode conduzir outros circuitos. Os canais duplos suportam conversões independentes.

Sensores



Touch Sensor

O ESP32 tem 10 GPIO de detecção capacitiva, que detectam variações induzidas ao tocar ou aproximar de um GPIO com um dedo ou outros objetos.

Capacitive-sensing signal name	Pin name
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	MTDO GPIO15
T4	MTCK GPIO13
T5	MTD1 GPIO12
T6	MTMS GPIO14
T7	GPIO27 GPIO27
T8	32K_XN GPIO33
T9	32K_XP GPIO32

Algumas placas (como essa da imagem, escondem o GPIO0)

O ESP32 ainda possui um sensor de Temperatura e um Sensor Hall interno, porém, para trabalhar com eles, deve-se mudar as configurações dos registradores. Para mais detalhes acesse manual técnico através do link:

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

Watchdog

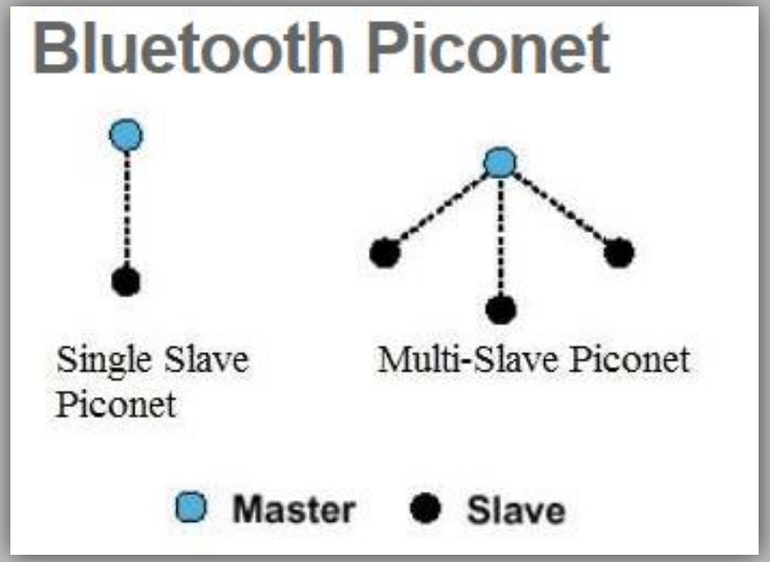
O ESP32 tem três temporizadores de vigilância: um em cada um dos dois módulos de temporizador (chamado o Temporizador de Watchdog Principal, ou MWDT) e um no módulo RTC (chamado RTC Watchdog Timer ou RWDT).

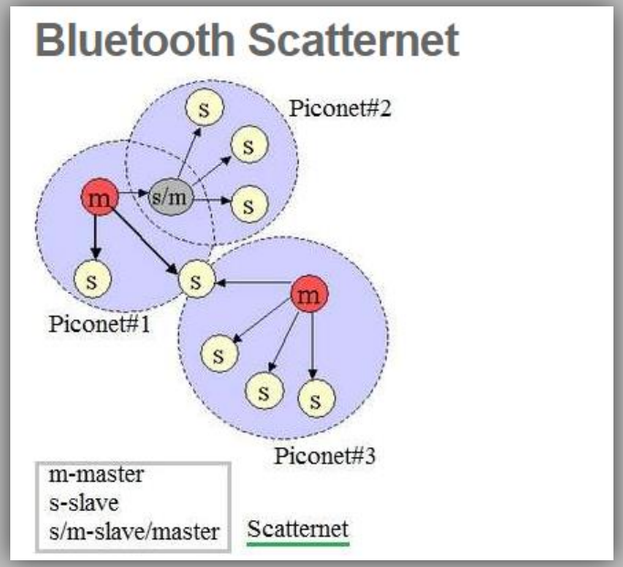
Bluetooth

Interface Bluetooth v4.2 BR/EDR e Bluetooth LE (low energy)

O ESP32 integra um controlador de ligação Bluetooth e Bluetooth baseband, que executam os protocolos de banda base e outras rotinas de links de baixo nível, como modulação / desmodulação, processamento de pacotes, processamento de fluxo de bits, saltos de frequência, etc.

O controlador de ligação opera em três estados principais: standby, connection e sniff. Permite múltiplas conexões, e outras operações, como inquiry, page e secure simple-pairing, e, portanto, permite a Piconet e Scatternet.





Boot

Em muitas placas de desenvolvimento com USB / Serial incorporado, isso é feito para você, o esptool.py pode redefinir automaticamente a placa para o modo de inicialização.

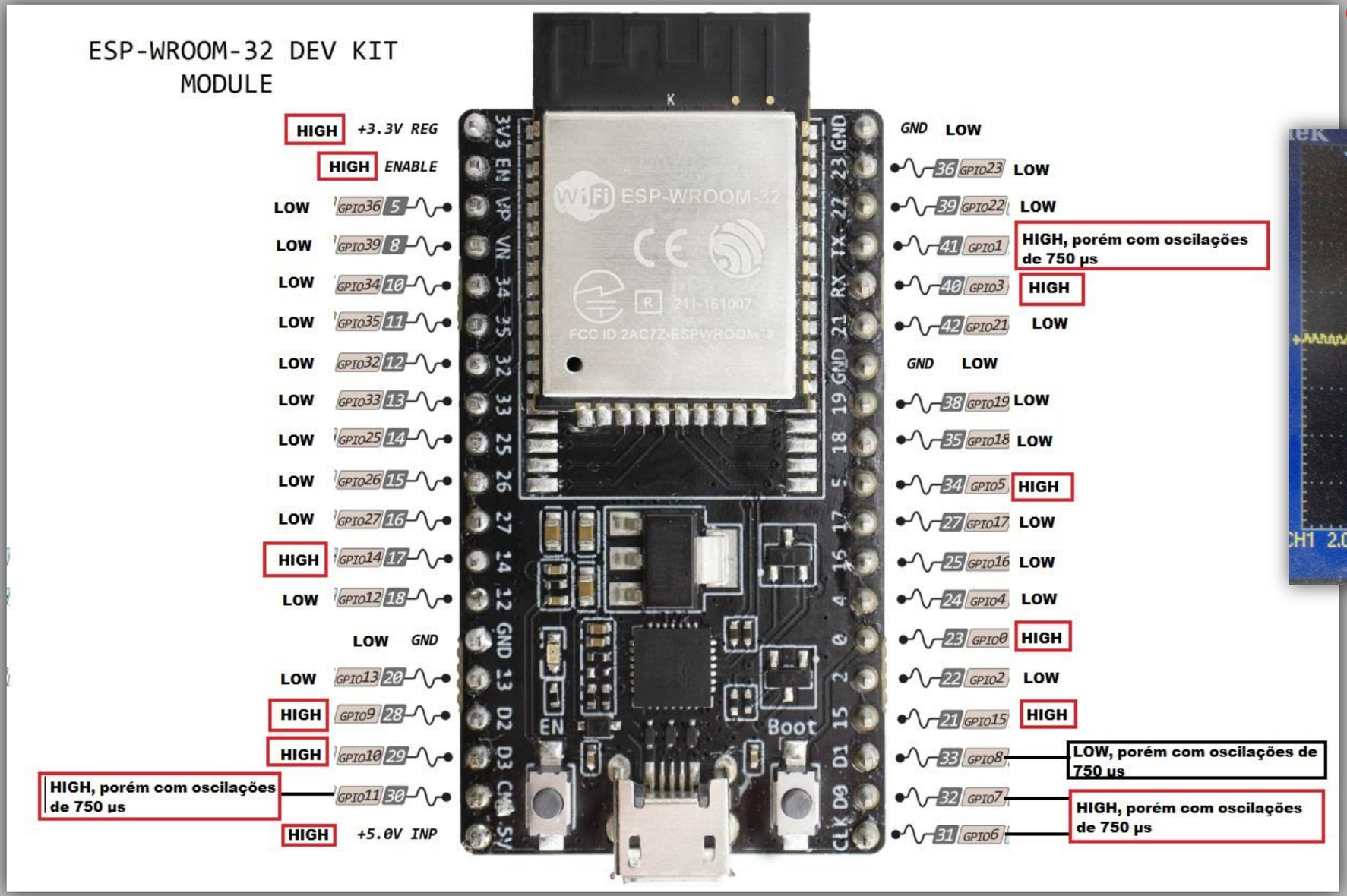
O ESP32 entrará no carregador de inicialização serial quando o GPIO0 for mantido baixo na reinicialização. Caso contrário, ele executará o programa em flash.

GPIO0 Input	Mode
Low/GND	ROM serial bootloader for esptool.py
High/VCC	Normal execution mode

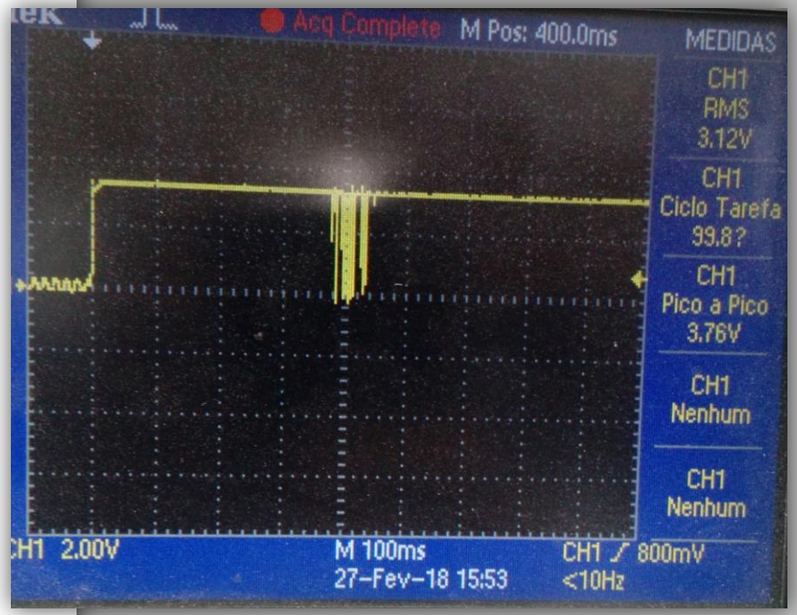
GPIO0 tem um resistor interno pullup, então se ele estiver sem conexão irá setar para alto. Muitas placas usam um botão marcado como “Flash” (ou “BOOT” em algumas placas de desenvolvimento Espressif) que seta o GPIO0 para baixo quando pressionado.

O GPIO2 também deve ser deixado sem conexão/flutuante.

Boot



Observe o comportamento de cada pino quando o ESP32 é iniciado (boot).



Essa é a imagem da oscilação de 750 µs que acontece nos pinos:

GPIO1, GPIO6, GPIO7, GPIO8 e GPIO11

Em www.fernandok.com

Download arquivos PDF e INO do código fonte