

LATINOWARE 2024



21º Congresso Latino-americano de
Software Livre e Tecnologias Abertas

Realização



Desenvolvendo seu projeto(produto) IoT com ESP32

Autores:

Fábio Souza

Desenvolvendo seu projeto IoT com ESP32

Fábio Souza

Desenvolvendo seu projeto IoT com ESP32

Por Fábio Souza



CC BY-NC-SA 4.0

Desenvolvendo seu projeto IoT com ESP32 © 2024 by [Fábio Souza](#) is licensed under [CC BY-NC-SA 4.0](#)

Fabio Souza



- Formação/Experiência
 - Especialização em Eletrônica Embarcada
 - Engenharia Elétrica
 - Técnico em Eletroeletrônica
 - Desenvolvimento de projetos eletrônicos para área de automação
 - Ensino de eletrônica e sistemas embarcados
- Atualmente:
 - Diretor de operações do Embarcados
 - Líder do projeto Franzininho
 - Professor, Instrutor de cursos e workshops
- Áreas de Interesse
 - Sistemas Embarcados
 - Sistemas operacionais de tempo Real
 - Internet das Coisas
 - Desenvolvimento de produtos
 - Escrita técnica
 - Movimento Maker
 - Open hardware
 - Educação EAD

[@fabiosouza.io](https://fabiosouza.io)

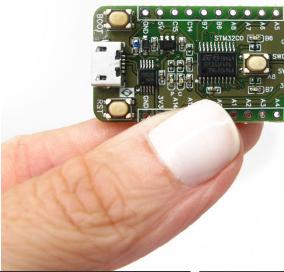
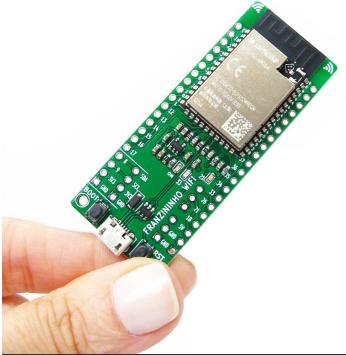


CONTEÚDO SOBRE SISTEMAS
ELETRÔNICOS EMBARCADOS PARA TODOS.
JUNTOS POR UM BRASIL MAIS TECNOLÓGICO.

www.embarcados.com.br



Franzininho - Ferramentas open source desenvolvidas no Brasil



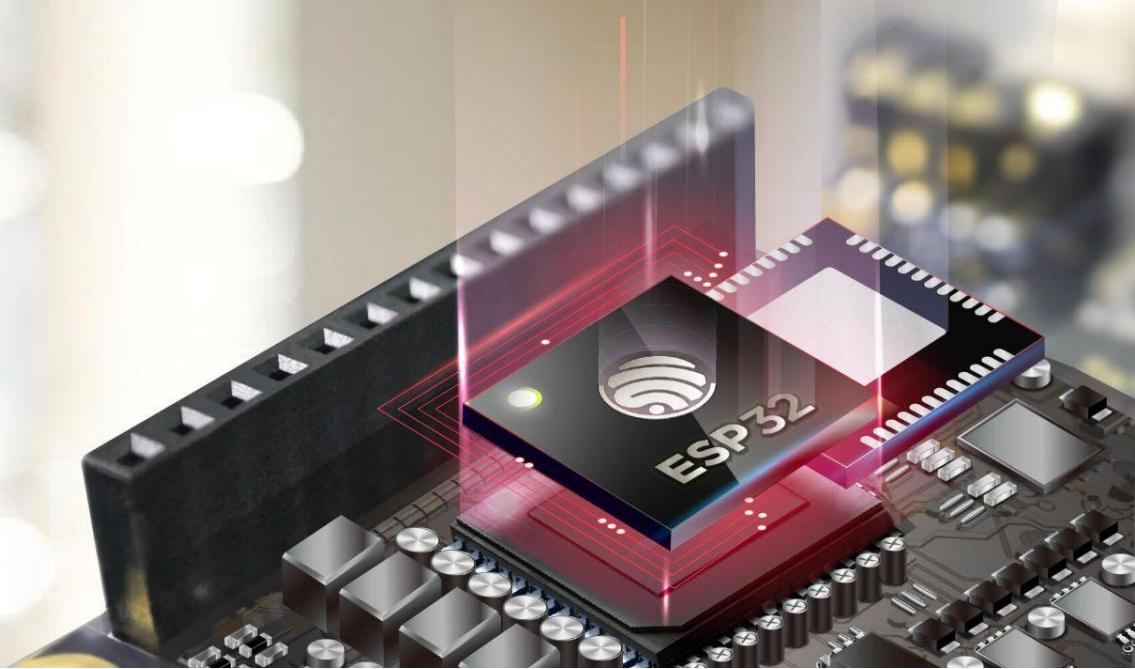
The screenshot shows the BIPES interface. On the left, there's a sidebar with various project categories like Lógica, Laços, Matemática, Texto, Listas, Variáveis, Funções, BIPES, Python, Temporização, Máquina, Displays, Sensores, Saídas e atuadores, Comunicação, Arquivos, Rede e Internet, Cloud IA / Vision, Controle, and micropython. The main area displays a breadboard setup with a central Arduino Uno-like board labeled "Franzininho WiFi LAB". A hand is shown pointing at the breadboard. To the right of the breadboard is a terminal window showing C code for a WiFi LCD I2C library. The code includes definitions for the LCD, network settings, and a function to print the current time.

```
1 // Franzininho WiFi LCD1602 I2C Clock Demo
2 // https://wokwi.com/arduino/projects/323796075549019410
3 // Copyright (C) 2022, URL Shaked
4
5 #include <WiFi.h>
6 #include <LiquidCrystal_I2C.h>
7
8 LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);
9
10 #define NTP_SERVER "pool.ntp.org"
11 #define UTC_OFFSET 0
12 #define UTC_OFFSET_DST 0
13
14 void setup() {
15     static int t_counter = 0;
16     const char* localTime = "00:00:00";
17     LCD.setBacklight(1);
18     LCD.print(glyphs[counter]);
19     if (counter == strlen(glyphs)) {
20         counter = 0;
21     }
22 }
23
24 void printLocalTime() {
25     struct tm* timeinfo;
26     if (!getLocaltime(&timeinfo)) {
27         lcd.setCursor(0, 0);
28     }
29 }
```

Agenda

- O que é o ESP32?
- Escolher SoC ou Módulo?
- Como desenvolver meu próprio Hardware?
- Como programar o ESP32?
- Onde Aprender?

ESP32





O que é o ESP32?

- Família de SoCs da Empresa Espressif Systems
 - Arquiteturas de 32-bit: Xtensa LX6, Xtensa 32-bit LX7 e RISC-V 32-bit
 - Conectividade: Wi-Fi, Bluetooth (LE) (Mesh), IEEE 802.15.4(Thread, Matter e Zigbee)
 - Alto desempenho com foco em AIoT (Inteligência Artificial das Coisas)
- Diversos recursos disponíveis:
 - Recursos para Baixo consumo;
 - Rico conjunto de interfaces periféricas e GPIOs, ideal para vários cenários:
 - SPI, I2S, UART, I2C, LED PWM, LCD interface, camera interface, ADC, DAC, touch sensor, USB Device e Host
 - Recursos de segurança:
 - secure boot, signature verification, integrated AES, SHA and RSA algorithms
- Variedade de opções em suas famílias:
 - ESP32, ESP32-S, ESP32-C, ESP32-H, ESP32-P
 - Todas suportadas pelo mesmo framework: ESP-IDF

O ESP32

- Anunciado em 2015 ([link](#)) como sucessor do ESP8266

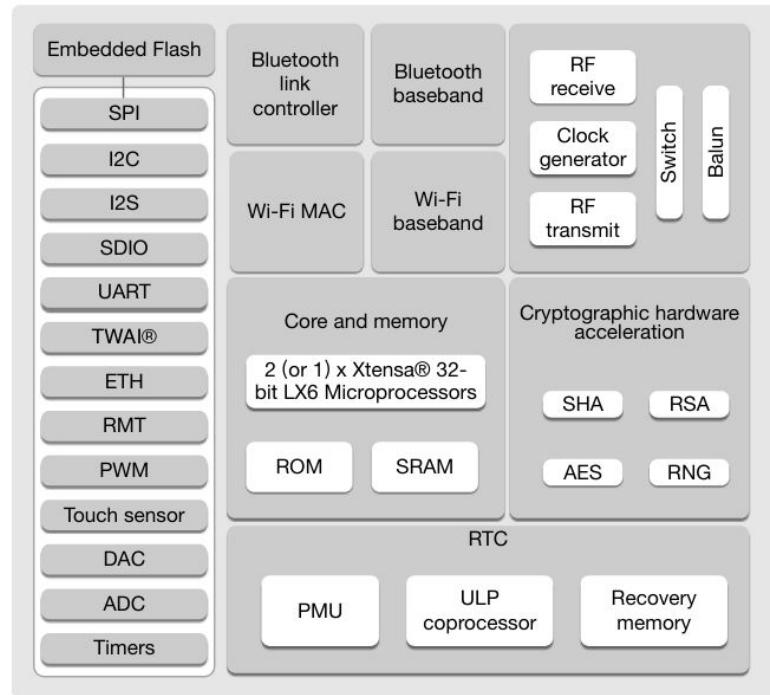
Notícia no Embarcados em 2015

A seguintes características estarão disponíveis no novo SoC:

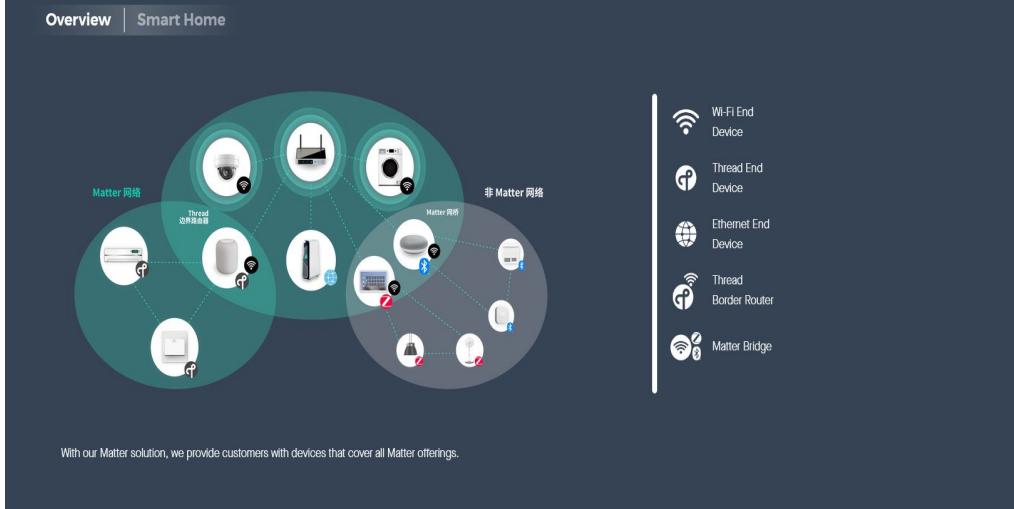
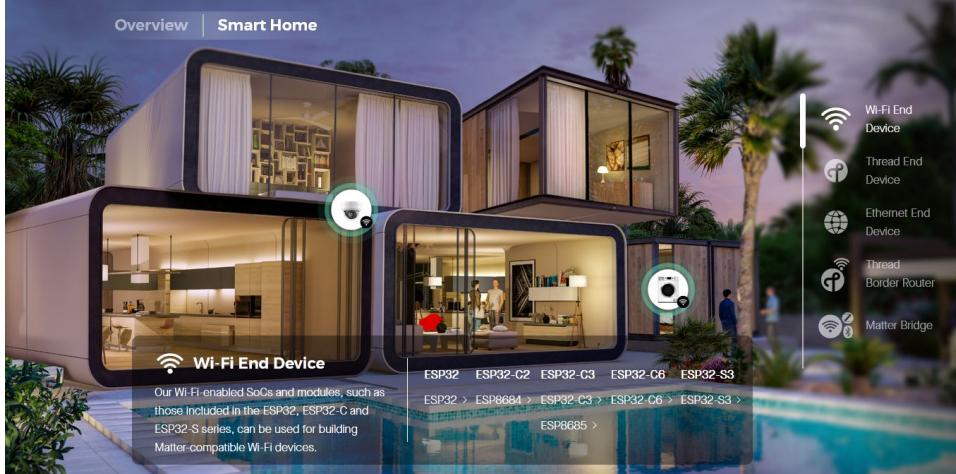
- WiFi mais rápido: O novo WiFi foi melhorado para suportar velocidade HT40 (144,4 Mbps).
- Bluetooth e Bluetooth Low Energy;
- 2 processadores **Tensilica** L108 trabalhando a 160 MHz;
- Low Power: diversos modos de funcionamento para baixo consumo;
- Variedades de periféricos: Touch capacitivo, ADCs, DACs, I2C. UART, SPI, SDIO, I2S, RMII, PWM, mas ainda não terá USB;
- Mais memória RAM: ~400 KB;
- Segurança melhorada: Aceleradores por hardware para AES e SSL, com diversas melhorias;
- APIs simplificadas: a API está sendo melhorada. O desenvolvimento ainda está em progresso e logo estará disponível.

ESP32

- Core: Xtensa® dual-core 32-bit LX6 CPU, frequency up to 240MHz
- Memories:
 - 448 KB of ROM
 - 520 KB of SRAM
 - 16 KB of RTCSRAM
- Working Voltage: 2.3 V to 3.6 V
- Up to 34 GPIOs
- 2*12-bit ADC (up to 18 channels)
- Communication interfaces
 - 2 I2C interfaces
 - 2 I2S interfaces
 - 4 SPI interfaces
 - 3 UART interfaces
- Security:
 - 1024 bit OTP
 - AES, SHA, RSA, ECC, RNG
 - Secure Boot, Flash Encryption
- Extended temperature range: -40 to 125 °C



Aplicações



Aplicações

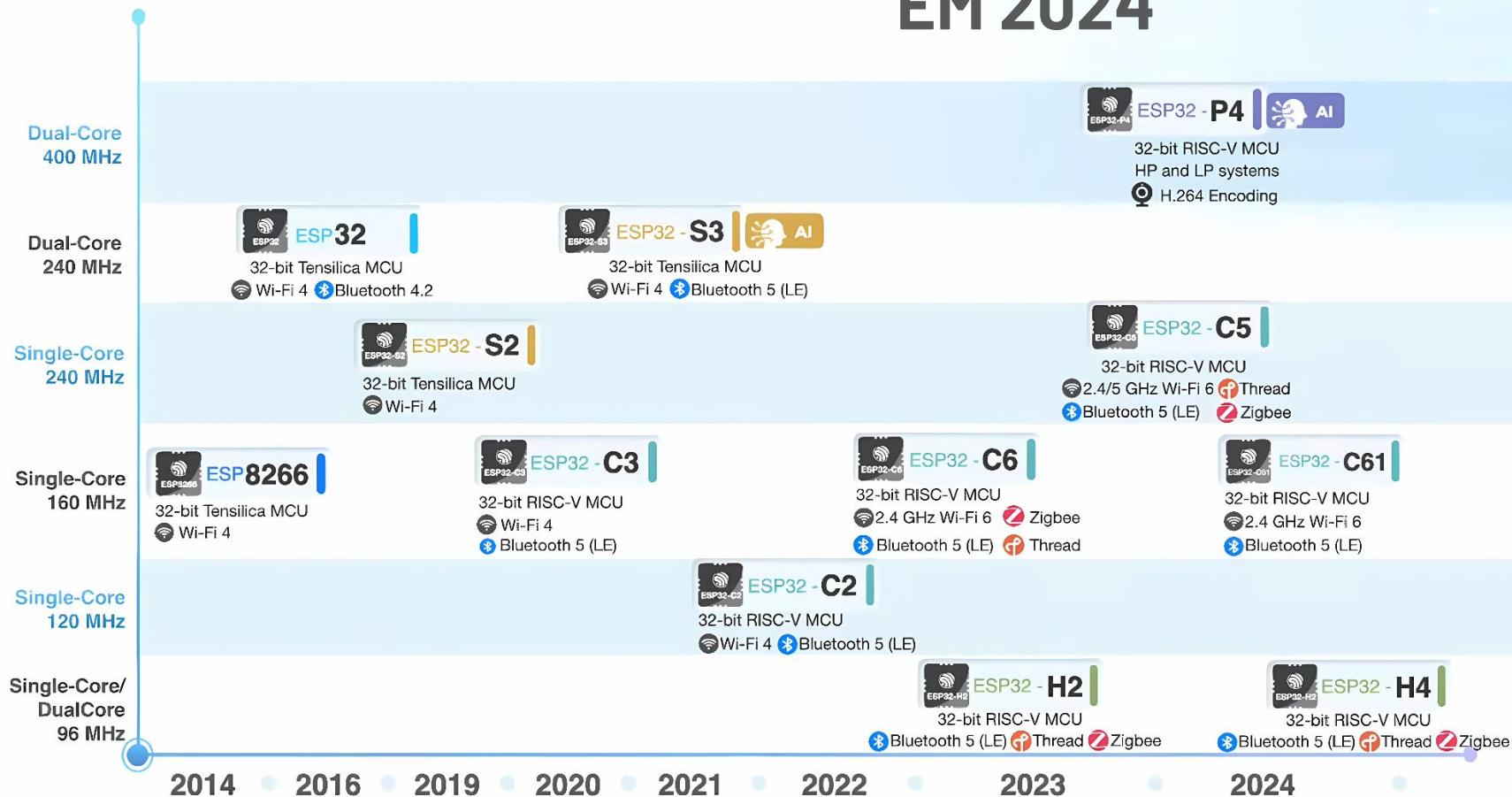
- Automação residencial
- Automação Industrial
- Dispositivos Médicos e de Saúde
- Eletrônicos de Consumo
- Agricultura Inteligente
- Sistemas de Ponto de Venda (POS)
- Robôs e Drones
- Dispositivos de Áudio
- Hubs de Sensores IoT de Baixo Consumo
- Registradores de Dados IoT de Baixo Consumo
- Câmeras para Transmissão de Vídeo
- Reconhecimento de Voz
- Reconhecimento de Imagem
- Detecção de Toque e Proximidade

SoCs ESP32





F





Espressif's Series of SoCs



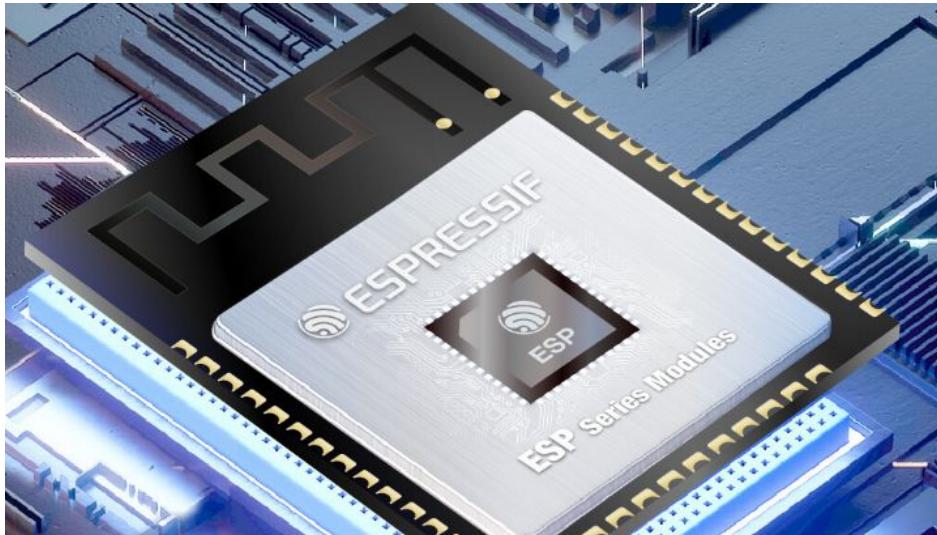
EM 2024

	ESP 8266 Tensilica	ESP 32 Xtensa®	ESP32 S2 Xtensa®	ESP32 S3 Xtensa®	ESP32 P4 RISC-V	ESP32 C2 RISC-V	ESP32 C3 RISC-V	ESP32 C5 RISC-V	ESP32 C6 RISC-V	ESP32 C61 RISC-V	ESP32 H2 RISC-V	ESP32 H4 RISC-V
Connectivity												
CPU	1 x Core @160MHz + 1 x ULP Core @8MHz	2 x HP Core @240MHz	1 x Core @240MHz	2 x HP Core @240MHz + 1 x ULP Core @17.5MHz	2 x HP Core @400MHz + 1 x LP Core @40MHz	1 x Core @120MHz	1 x Core @160MHz	1 x HP Core @240MHz + 1 x LP Core @40MHz	1 x HP Core @160MHz + 1 x LP Core @20MHz	1 x Core @160MHz	1 x Core @96MHz	2 x Core @96MHz
RAM	160KB	520KB	320KB	512KB	768KB	272KB	400KB	384KB	512KB	320KB	320KB	320KB
Optional PSRAM	No	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes	No	Yes
GPIO	17	34	43	45	55	14	22 or 16	29	30 or 22	22 or 18	19	32
Status	MP (NRND)	MP	MP	MP	Sample in Q3 2024	MP	MP	Sample in Q3 2024	MP	Sample in Q3 2024	MP	Sample in Q4 2024

Módulos ESP32



Módulos



<https://www.espressif.com/en/products/modules>

Escolher SoC ou Módulo?



SoC ou Módulo

1. Complexidade de Design
2. Tempo de Desenvolvimento
3. Custo
4. Certificações
5. Produção em Pequena ou Grande Escala

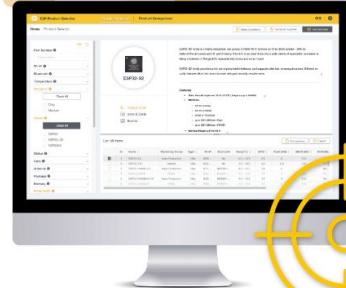
Quando Usar um SoC

- Quando você precisa de controle total sobre o design de hardware e RF.
- Se seu projeto tem requisitos muito específicos de tamanho e desempenho.
- Quando seu volume de produção é grande o suficiente para justificar o investimento em design e certificações.

Quando Usar um Módulo

- Se você deseja reduzir o tempo de desenvolvimento e facilitar o design de hardware.
- Quando o projeto é de pequena ou média escala.
- Se você quer evitar as complicações e os custos associados às certificações de RF e compatibilidade eletromagnética (EMC).
- Se o foco é prototipagem rápida e integração simples.

ESP Product Selector



The screenshot shows the 'Project Overview' screen of the ESP Product Selector. At the top, there's a navigation bar with tabs like 'Project Overview', 'Search', 'Compare', 'List View', and 'Grid View'. Below the navigation is a search bar with placeholder text 'Search by part number or keyword...'. The main area displays a table with columns: 'Part Number', 'Product Type', 'Status', 'Last Update', and 'Actions'. One row is highlighted in yellow, showing 'ESP8266-02' as the part number, 'Module' as the product type, 'Active' as the status, '2019-01-08' as the last update, and a 'View Details' button. To the left of the table, there's a sidebar with sections for 'Part Number', 'Product Type', 'Status', and 'Last Update'. A large yellow call-to-action button labeled 'Start Now' is positioned at the bottom left. The background features a grey gradient with a faint watermark of various ESP chip components.

Choosing the ESP products you need has never been easier!

Start Now

<https://products.espressif.com/#/>

Product Selector

ESP Product Selector | Product Selector | Product Comparison | 中文 | ?

Home / Product Selector

Sales Questions | Technical Inquiries | Get Samples

Part Number

Wi-Fi N/A 2.4 GHz 802.11 b/g/n

Check All

Bluetooth

Temperature

Products SoC Module

Series ESP32-S3 ESP32-C3(including ESP8685) ESP32-S2 ESP32 ESP8266

Status

ESP SOLUTIONS

ESP-WNIC
Espressif's Wireless Network Interface Controller

- Recommended Soc: **ESP32 / ESP32-S / ESP32-C**
- Solution Description: With ESP-WNIC, users can quickly join a wireless network, connect to a Cloud platform, transmit data, and implement remote-control functions. It can be widely used in smart-home applications, consumer electronics, weather monitoring, equipment monitoring,...

ESP-NOW
Espressif's Wireless-Communication Protocol

- Recommended Soc: **ESP32-S / ESP32-C / ESP32 / ESP8266**
- Solution Description: ESP-NOW is an efficient, responsive, and easy-to-develop protocol based on the data-link layer defined by Espressif. ESP-NOW can provide a flexible data transmission that is suitable for connecting "one-to-many" and "many-to-many" devices. It can also be use...

List: 2 items

	Index	Name	MPN	Marketing Status	Type	Wi-Fi	Bluetooth	Temp (°C)	GPIO	Flash (MB)	SRAM (KB)	ROM (KB)	PSRAM (MB)	Freq. (MHz)
<input type="checkbox"/>	1	ESP32-S2	ESP32-S2	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...	N/A	-40 ~ 105	43	0	320	128	0	240
<input type="checkbox"/>	2	ESP32-S2	ESP32-S2R2	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...	N/A	-40 ~ 85	43	0	320	128	2	240

Comparison | **Export**

<https://www.espressif.com/en/contact-us/sales-questions>

<https://products.espressif.com/#/>

Product Comparison

ESP Product Selector | Product Selector | Product Comparison | 中文 | ?

Home / Product Comparison | Sales Questions | Technical Inquiries | Get Samples

SoC | Module

	ESP32-S2	ESP32-D0WDQ6-V3	ESP32-C3	ESP32-S3	
Overview	Series	ESP32-S2	ESP32	ESP32-C3	
	CPU	Xtensa® single-core 32-bit LX7	Xtensa® dual-core 32-bit LX6	32-bit RISC-V single-core processor	Xtensa® dual-core 32-bit LX7
	Core	1	2	1	2
	Freq. (MHz)	240	240	160	240
	Package (mm)	QFN56 (7*7)	QFN48 (6*6)	QFN32 (5*5)	QFN56 (7*7)
	Dimensions (mm)	7*7	6*6	5*5	7*7
	Temp. (°C)	-40 °C ~ 105 °C	-40 °C ~ 125 °C	-40 °C ~ 105 °C	-40 °C ~ 105 °C
	Status	Mass Production	NRND	Mass Production	Mass Production
Wireless	ECO	standard version	ECO V3	standard version	
	Bluetooth	N/A	BR/EDR + Bluetooth LE v4.2	Bluetooth LE v5.0	Bluetooth LE v5.0
	SRAM (KB)	320	520	400	512
	Memory	ROM (KB)	128	448	384
	RTC SRAM (KB)	16	16	8	16
	ADC	2*12-bit ADC, 20 channels	2*12-bit ADC, 18 channels	2*12-bit ADC, 6 channels	2*12-bit ADC, 20 channels
	DAC	2*8-bit DAC	2*8-bit DAC	0	0
	Touch	14	10	0	14
	Total	~	~	~	~

<https://products.espressif.com/#/>

Compromisso de Longevidade



Our Commitment

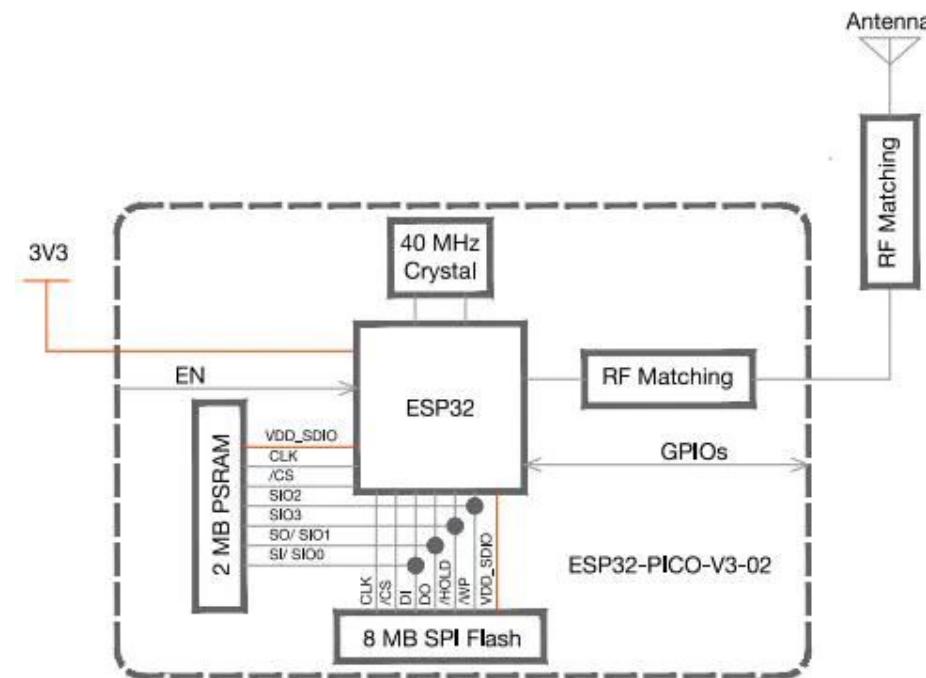
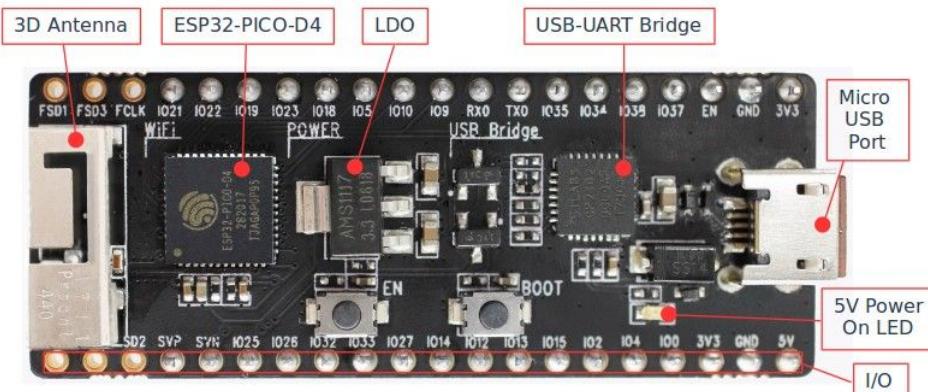
Espressif provides a minimum longevity commitment for the product series listed below.

Series	Longevity Commitment	Start Date
ESP 32-S Series	12 YEARS	from Jan 1, 2020
ESP 32-S2	12 YEARS	from Jan 1, 2021
ESP 32-S3	12 YEARS	from Jan 1, 2021
ESP 32-H Series	12 YEARS	from Jan 1, 2023
ESP 32-H2	12 YEARS	from Jan 1, 2023
ESP 32 Series	15 YEARS	from Jan 1, 2016
ESP 32-C Series	12 YEARS	from Jan 1, 2022
ESP 32-C2	12 YEARS	from Jan 1, 2021
ESP 32-C3	12 YEARS	from Jan 1, 2023
ESP 32-C6	12 YEARS	from Jan 1, 2023
ESP 8266 Series	15 YEARS	from Jan 1, 2014
ESP 8285		
ESP 8089 Series	12 YEARS	from Jan 1, 2014

<https://www.espressif.com/en/products/longevity-commitment>

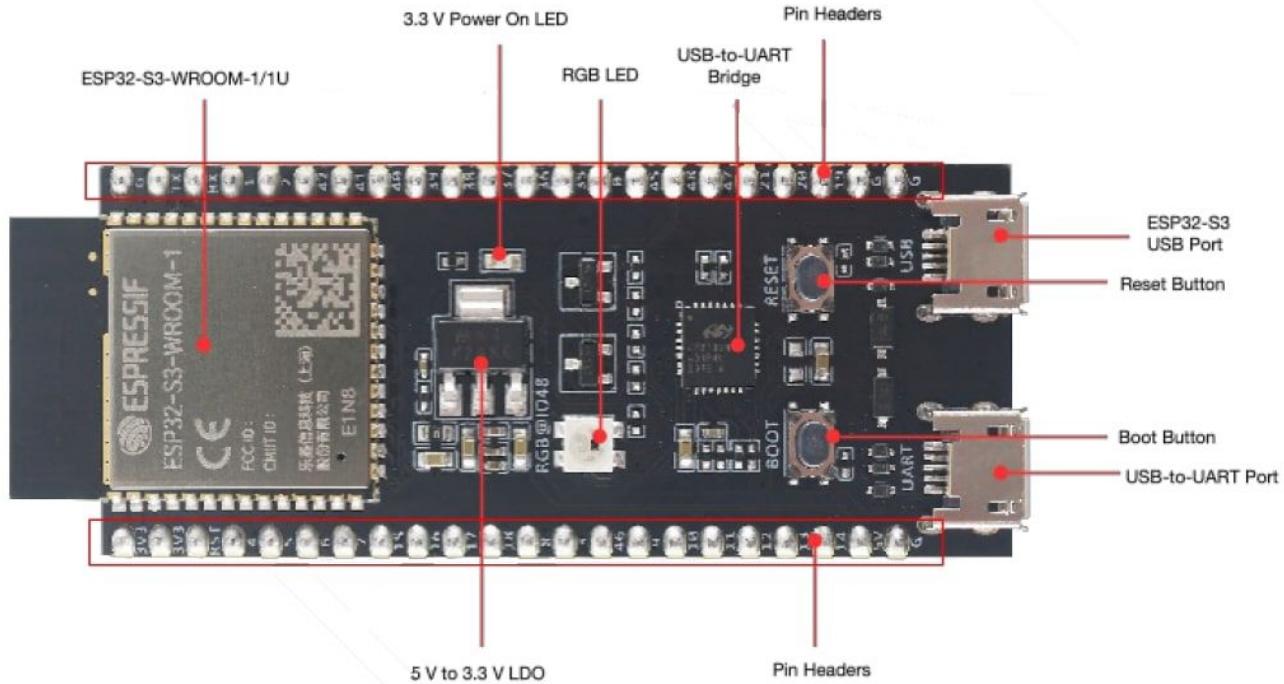
Placas de
desenvolvimento

ESP32-PICO



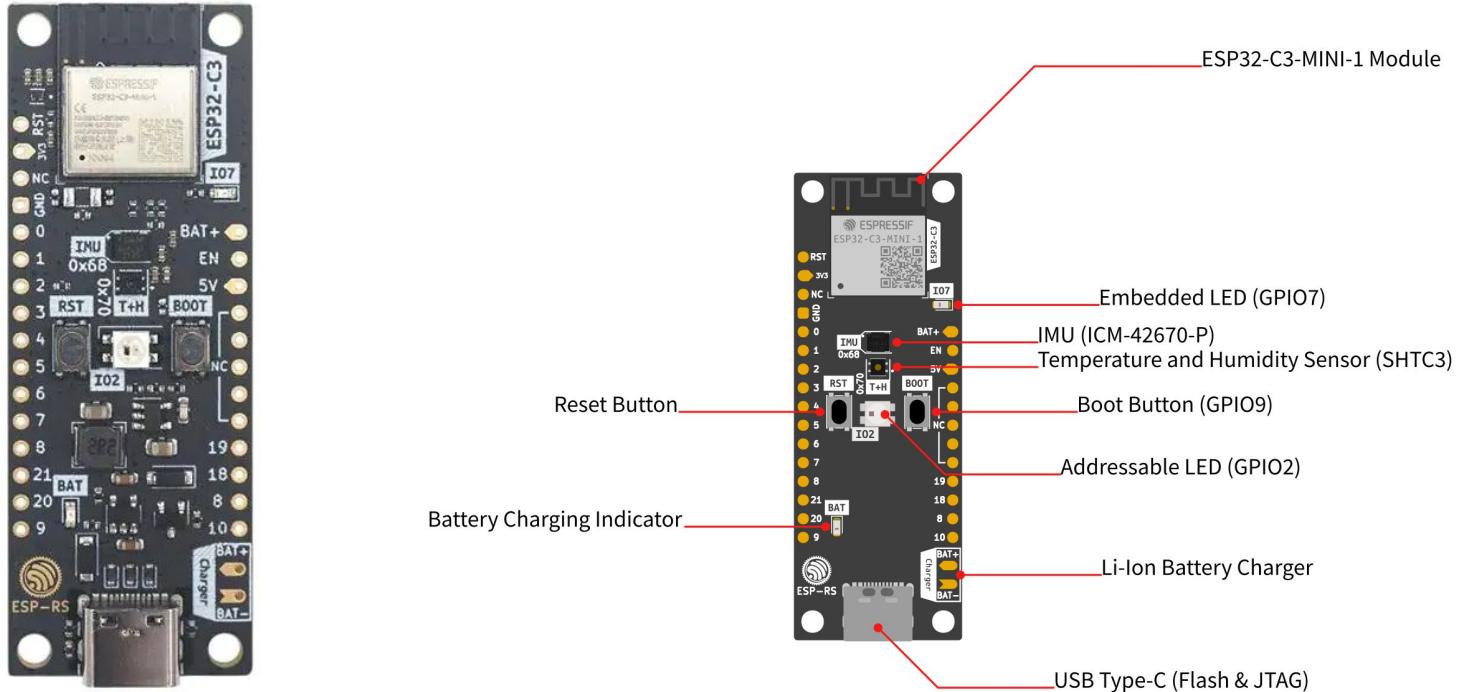
<https://br.mouser.com/new/espressif/espressif-esp32-pico-v3-02-module/>

ESP32-S3-DevKitC-1



<https://br.mouser.com/new/espressif/espressif-esp32-pico-v3-02-module/>

ESP32-C3-DevKit-RUST-1



<https://br.mouser.com/new/espressif/espressif-esp32-c3-devkit-rust-1-board/>

ESP32-S3-BOX-3



ESP32-S3-BOX-3

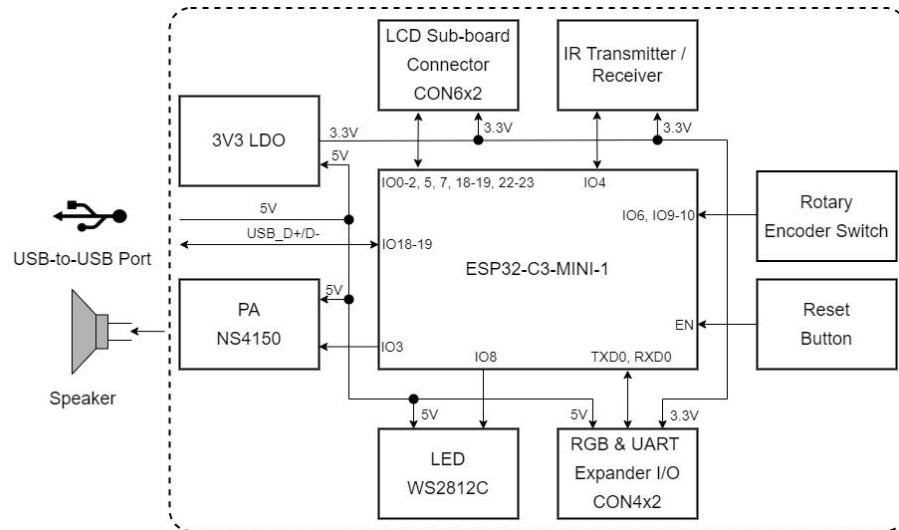
Next-generation AIoT Development Tool



Powered by
ESP32-S3

Microcontroller			
Type:	ESP32-S3	CPU:	Dual-Core Xtensa® 32bit LX7 up to 240 MHz
Memory			
SRAM:	512 KB	ROM:	384 KB
PSRAM:	Octal SPI, 16 MB	PSRAM Speed:	120 MHz (Experimental Feature)
External Flash:	Quad SPI, 16 MB		
AI Feature			
AI Algorithm	Neural Network, Acoustic algorithm, etc.	Computing Acceleration	Vector, Complex number, FFT, etc.
Wireless			
Wi-Fi:	2.4 GHz, IEEE 802.11b/g/n	Bluetooth® LE:	Bluetooth® 5 LE and Bluetooth® mesh
Display			
Display Type:	2.4-inch LCD	Display Resolution:	240 x 320 pixels
Display Interface:	SPI	Interface Speed:	40 MHz
Touch Type:	Capacitive	Touch Points:	10
Driver IC	ILI9342C		
Audio Input			
Microphone Type:	Dual Mic	ADC Model:	ES7210
Mute:	Supported		
Audio Output			
Speaker Model:	8 Ohm 1 W	PA Model:	NS4150
Codec Model:	ES8311		
Sensor			
Sensor Type:	3-axis Gyroscope, 3-axis Accelerometer	Sensor Model:	ICM-42607-P
Interface			
Type:	USB Type-C	Usage:	Power, USB download/JTAG debug, general USB device functions
Type:	Goldfinger	Usage:	I/O Expansion
User Interface			
Onboard Buttons	Reset, Boot, Mute	Onboard LEDs	Power LED, Mute LED
OS / SDK			
Original OS:	FreeRTOS	SDK:	ESP-IDF
Outline			
Dimensions:	61 x 66 x 16.6 mm	Weight:	292 g
Power			
USB-C Power:	5 V - 2.0 A Input	Battery:	N/A

ESP32-C3-LCDkit



Como desenvolver meu
próprio Hardware?



Home / Product Selector

[Sales Questions](#)[Technical Inquiries](#)[Get Samples](#)

Series ?

[Check All](#)

- ESP8266
- ESP32
- ESP32-S2
- ESP32-C3(including ESP8685)
- ESP32-S3
- ESP32-C2(including ESP8684)

Status ?

[Check All](#)

- Mass Production
- NRND
- Sample

Core ?

Antenna ?



ESP32-WROVER-IE(1...

Product Brief

Docs & Certs

DevKits

Documents

- [ESP32-WROVER-E & ESP32-WROVER-IE Datasheet](#)
- [ESP32 Technical Reference Manual](#)
- [ECO and Workarounds for Bugs in ESP32](#)
- [ESP32-WROVER-IE-V1.2 参考设计](#)
- [ESP32-WROVER-IE-V1.2 Reference Design](#)

Certificates

- [ESP32-WROVER-E & ESP32-WROVER-IE & ESP32-WROOM-32E & ESP32-WROOM-32UE BQB Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE KCC Certification](#)
- [ESP32-WROVER-IE NCC Certification](#)
- [ESP32-WROVER-IE SRRC Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE FCC DSS](#)
- [ESP32-WROVER-IE Wi-Fi Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE IC Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE MIC Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE CE Certification](#)
- [ESP32-WROVER-E & ESP32-WROVER-IE FCC DTS](#)

ESP32-WROVER-IE-N16R8

ESPRESSIF



Images are for reference only
See Product Specifications

Mouser #: 356-ESP32WRVIE2864UC

Mfr. #: ESP32-WROVER-IE-N16R8

Mfr.: Espressif Systems

Customer #:

Description: Multiprotocol Modules SMD Module ESP32-WROVER-IE, ESP32-D0WD-V3, 3.3V 64Mbit PSRAM, 16 MB SPI flash, IPEX connector

Datasheet: [ESP32-WROVER-IE-N16R8 Datasheet \(PDF\)](#)

More Information [Learn more about Espressif Systems ESP32-WROVER-IE-N16R8](#)

Compare Product

[Add To Project](#) | [Add Notes](#)

Specifications

In Stock: 29,350

Stock: 29,350 Can Ship Immediately

On Order: 3,250 Expected 08-Jul-22

Factory Lead-Time: 19 Weeks

Enter Quantity: Minimum: 1 Multiples: 1

[Buy](#)

Pricing (USD)

Qty.	Unit Price	Ext. Price
1	\$3.90	\$3.90

Full Reel (Order in multiples of 650)

650	\$3.90	\$2,535.00
-----	--------	------------

FEATURED PRODUCTS

ESPRESSIF



ESP8684-MINI Small Wi-Fi® &
BLUETOOTH® 5 Modules

12.1 Must-Read Documents

The following link provides documents related to ESP32.

- [*ESP32 Datasheet*](#)

This document provides an introduction to the specifications of the ESP32 hardware, including overview, pin definitions, functional description, peripheral interface, electrical characteristics, etc.

- [*ESP32 ECO V3 User Guide*](#)

This document describes differences between V3 and previous ESP32 silicon wafer revisions.

- [*ECO and Workarounds for Bugs in ESP32*](#)

This document details hardware errata and workarounds in the ESP32.

- [*ESP-IDF Programming Guide*](#)

It hosts extensive documentation for ESP-IDF ranging from hardware guides to API reference.

- [*ESP32 Technical Reference Manual*](#)

The manual provides detailed information on how to use the ESP32 memory and peripherals.

- [*ESP32 Hardware Resources*](#)

The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32 modules and development boards.

- [*ESP32 Hardware Design Guidelines*](#)

The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including the ESP32 chip, the ESP32 modules and development boards.

- [*ESP32 AT Instruction Set and Examples*](#)

This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.

- [*ESP Product Selector*](#)

ESP32

Technical Reference Manual

ESP32-WROVER-E & ESP32-WROVER-IE Datasheet

the recommended operating conditions.

Table 5: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	-0.3	3.6	V
I_{output}^1	Cumulative IO output current	-	1,100	mA
T_{store}	Storage temperature	-40	105	°C

1. The module worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground. Please note that pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.
2. Please see Appendix *IO_MUX* in [ESP32 Datasheet](#) for IO's power domain.

6.2 Recommended Operating Conditions

Table 6: Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
VDD33	Power supply voltage	3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T	Operating temperature	-40	-	85	°C

6.3 DC Characteristics (3.3 V, 25 °C)

Table 7: DC Characteristics (3.3 V, 25 °C)

8 Peripheral Schematics

This is the typical application circuit of the module connected with peripheral components (for example, power supply, antenna, reset button, JTAG interface, and UART interface).

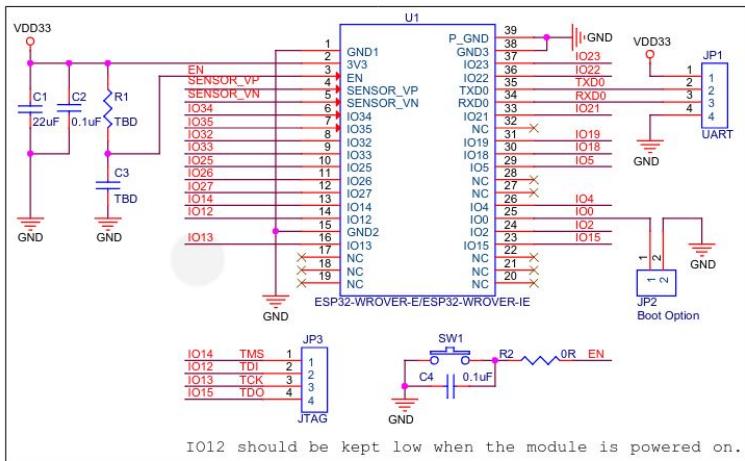
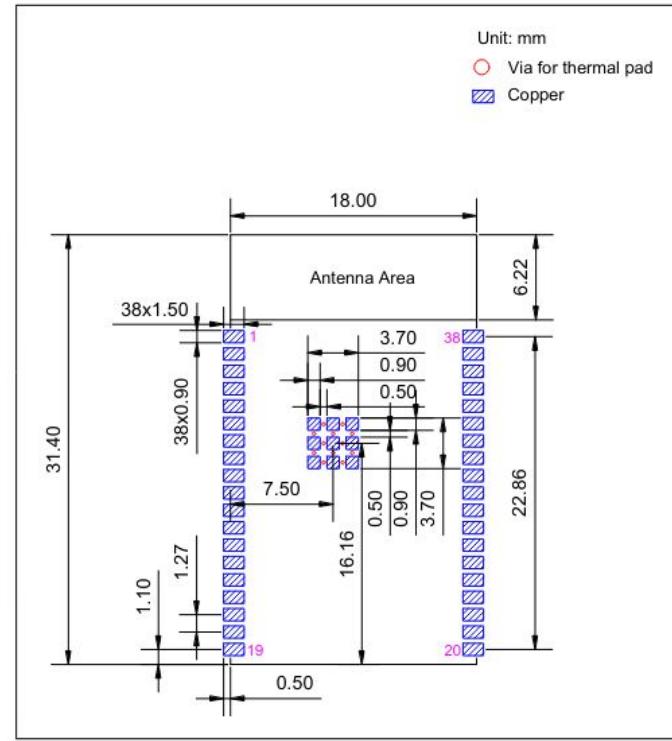
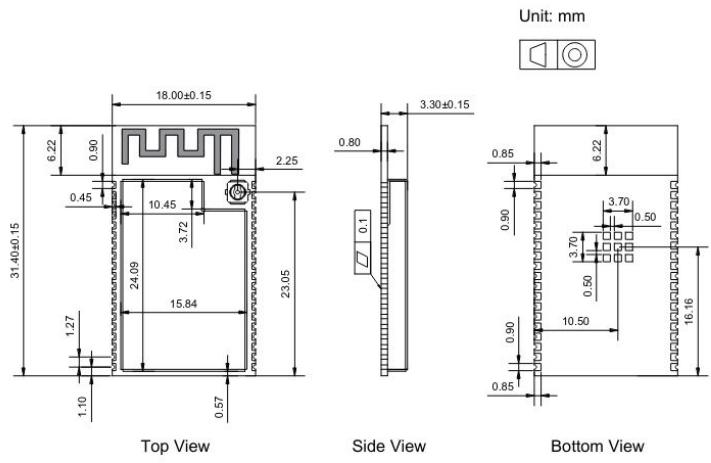


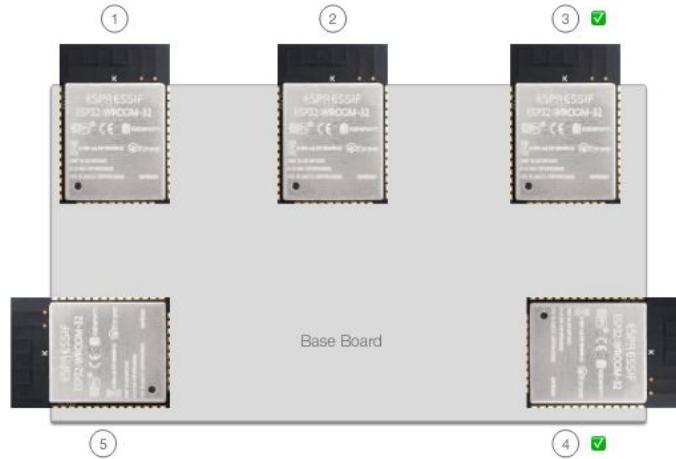
Figure 7: Peripheral Schematics



ESP32

Hardware Design Guidelines

recommended.



Antenas



PCB onboard antenna



Rod antenna



FPC antenna



Ceramic antenna



3D metal antenna

Kicad Library

The screenshot shows the GitHub repository page for `espressif/kicad-libraries`. The repository is public and contains 56 commits from `pedrominatel`. The repository has 3 branches and 2 tags. The code tab is selected. The repository description is "KiCad libraries for Espressif chips and modules". It includes links to Readme, View license, and statistics like 165 stars, 11 watching, and 32 forks. The Releases section shows 2 tags. The Packages section indicates no packages have been published.

Search or jump to... / Pull requests Issues Marketplace Explore

espressif / kicad-libraries Public

Watch 11 Fork 32 Starred 165

Code Issues 2 Pull requests 1 Actions Projects Wiki Security Insights

main 3 branches 2 tags Go to file Add file Code

pedrominatel Fixed the modules descriptions (#47) e862365 6 days ago 56 commits

3d Added ESP8685 symbol and ESP32-C3-WROOM-02U footprint and 3D... 4 months ago

footprints/Espressif.pretty ESP32-S2-DevKitM (#37) 3 months ago

libraries Fixed the modules descriptions (#47) 6 days ago

.gitignore Added ESP32-S3 modules symbols 12 months ago

LICENSE.md Added first libraries (symbols and footprints) and 3d models 12 months ago

README.md ESP32-S2-DevKitM (#37) 3 months ago

README.md

Espressif KiCad Library

About

KiCad libraries for Espressif chips and modules

Readme View license 165 stars 11 watching 32 forks

Releases 2 tags

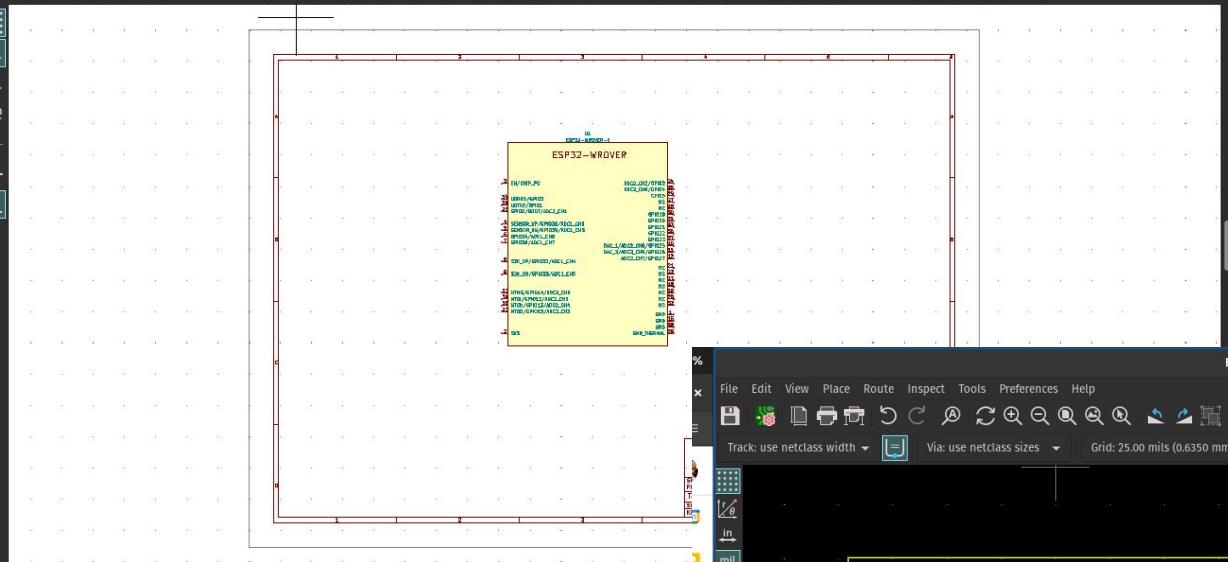
Packages

No packages published

<https://github.com/espressif/kicad-libraries>

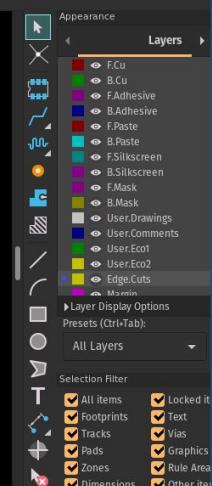
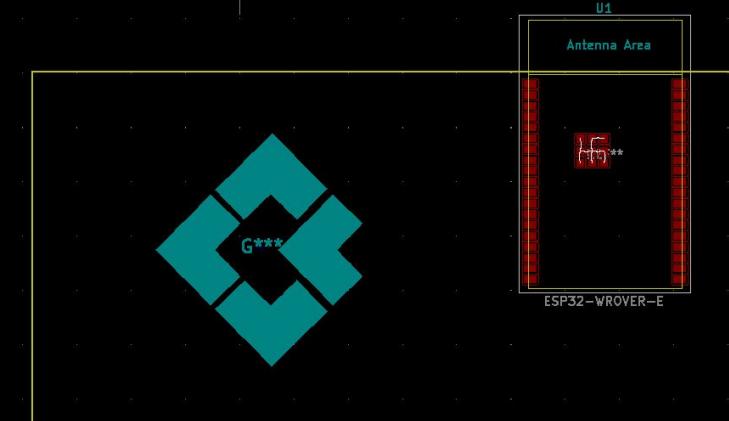
*ESP32-Board [ESP32-Board/] — Schematic Editor

File Edit View Place Inspect Tools Preferences Help



ESP32-Board — PCB Editor

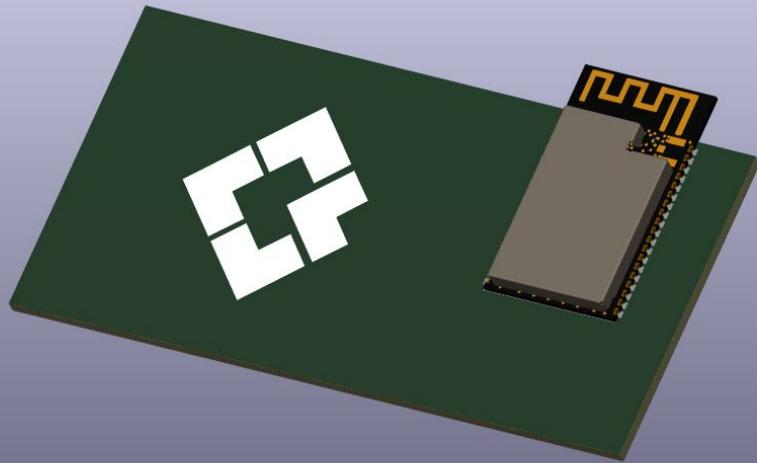
File Edit View Place Route Inspect Tools Preferences Help



File '/home/fabio/Documents/ESP32-Board...' Z 2.81 X 3750.00 Y 1825.00 dx 3750.00 dy 1825.00 dist 4170.51 grid X 25.00 Y 25.00 mils Select item(s)

3D Viewer

File Edit View Preferences Help



dx 0.00

dy 0.00

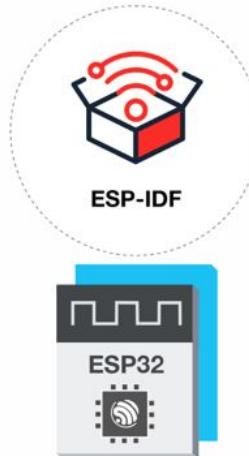
Last render time 2 ms

Como programar o ESP32?

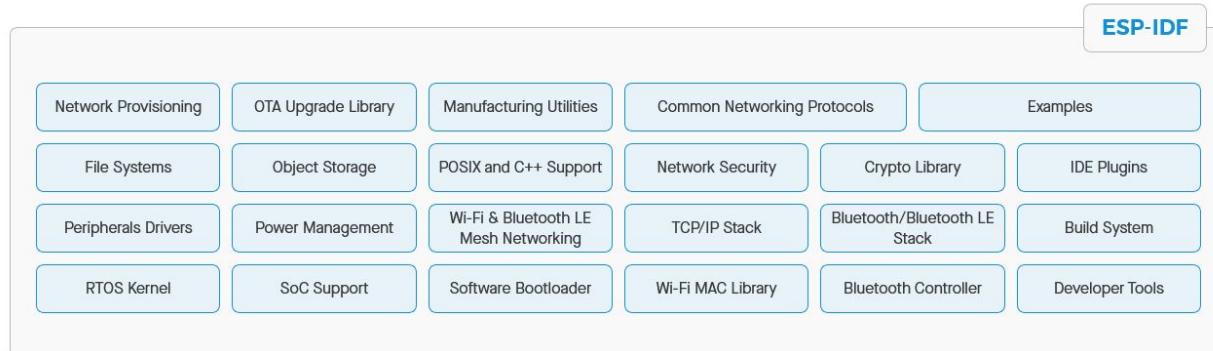


ESP-IDF

- O ESP-IDF é o framework oficial de desenvolvimento IoT da Espressif para as séries de SoCs ESP32.
- Ele oferece um SDK completo para o desenvolvimento de aplicações utilizando linguagens de programação como C e C++.



Software Components and Features



<https://www.espressif.com/en/products/sdks/esp-idf>

Espressif Frameworks and Libraries

ESP-ADF

Audio Development Framework.



[Learn More >](#)

ESP-MDF

Mesh Development Framework for self-forming, self-healing Wi-Fi Mesh applications.



[Learn More >](#)

ESP-IoT-Solution

Various application examples and peripheral drivers for third-party peripherals.



[Learn More >](#)

ESP HomeKit SDK

HomeKit SDK supporting Apple HomeKit-certified accessory development.



[Learn More >](#)

Cloud Connectivity Agents

Support for AWS IoT Core, Azure IoT and Google IoT Core cloud connectivity libraries.



[ESP-AWS-IoT >](#) [ESP-Azure-IoT >](#)
[ESP-Google-IoT >](#)

ESP-Jumpstart

A framework and a step-by-step tutorial on building production-ready applications.



[Learn More >](#)

ESP RainMaker

A complete device firmware + cloud service + mobile app solution for makers.



[Learn More >](#)

ESP-Arduino

Arduino IDE and libraries-based development support for ESP32 SoC.



[Learn More >](#)

AI and DSP Libraries

Libraries for building AI and DSP-based applications.



[ESP-WHO >](#) [ESP-Skainet >](#)
[ESP-DSP >](#)

<https://www.espressif.com/en/products/sdks/esp-idf>

Arduino

Supported Chips

Here are the ESP32 series supported by the Arduino-ESP32 project:

SoC	Stable	Development	Datasheet
ESP32	Yes	Yes	ESP32
ESP32-S2	Yes	Yes	ESP32-S2
ESP32-C3	Yes	Yes	ESP32-C3
ESP32-S3	Yes	Yes	ESP32-S3
ESP32-C6	Yes	Yes	ESP32-C6
ESP32-H2	Yes	Yes	ESP32-H2

 Note

ESP32-C2 is also supported by Arduino-ESP32 but requires rebuilding the static libraries. This is not trivial and requires a good understanding of the ESP-IDF build system. For more information, see the [Lib Builder documentation](#).

<https://github.com/espressif/arduino-esp32>

Supported Peripherals

Currently, the Arduino ESP32 supports the following peripherals with Arduino APIs.

Peripheral	ESP32	S2	C3	S3	C6	H2	Notes
ADC	Yes	Yes	Yes	Yes	Yes	Yes	
BT Classic	Yes	N/A	N/A	N/A	N/A	N/A	
BLE	Yes	N/A	Yes	Yes	Yes	Yes	
DAC	Yes	Yes	N/A	N/A	N/A	N/A	
Ethernet	Yes	N/A	N/A	N/A	N/A	N/A	(*)
GPIO	Yes	Yes	Yes	Yes	Yes	Yes	
Hall Sensor	N/A	N/A	N/A	N/A	N/A	N/A	
I2C	Yes	Yes	Yes	Yes	Yes	Yes	
I2S	Yes	Yes	Yes	Yes	Yes	Yes	
LEDC	Yes	Yes	Yes	Yes	Yes	Yes	
Motor PWM	No	N/A	N/A	N/A	N/A	N/A	
Pulse Counter	No	No	No	No	No	No	
RMT	Yes	Yes	Yes	Yes	Yes	Yes	
SDIO	No	No	No	No	No	No	
SDMMC	Yes	N/A	N/A	Yes	N/A	N/A	
Timer	Yes	Yes	Yes	Yes	Yes	Yes	
Temp. Sensor	N/A	Yes	Yes	Yes	Yes	Yes	
Touch	Yes	Yes	N/A	Yes	N/A	N/A	
TWAI	No	No	No	No	No	No	
UART	Yes	Yes	Yes	Yes	Yes	Yes	
USB	N/A	Yes	Yes	Yes	Yes	Yes	(**)
Wi-Fi	Yes	Yes	Yes	Yes	Yes	N/A	

<https://docs.espressif.com/projects/arduino-esp32/en/latest/libraries.html>

Outras Alternativas

- [Zephyr RTOS](#)
- [NuttX](#)
- MicroPython
- CircuitPython

Por que escolher o
ESP-IDF?

ESP-IDF

- SDK principal da Espressif para toda a ESP32
- Suporte para os novos dispositivos que forem lançados
- Base para diversos outros frameworks da Espressif
- Diversos exemplos e documentação disponíveis
- Rico ecossistema de softwares e ferramentas
- Política de lançamento e suporte de longo prazo
- Open Source: sob a licença Apache 2.0. Componentes de terceiros estão disponíveis sob uma licença permissiva compatível

<https://www.espressif.com/en/products/sdks/esp-idf>

ESP-IDF Release



<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/versions.html>

Onde Aprender?

Documentação IDF

[ESP-IDF Programming Guide](#)

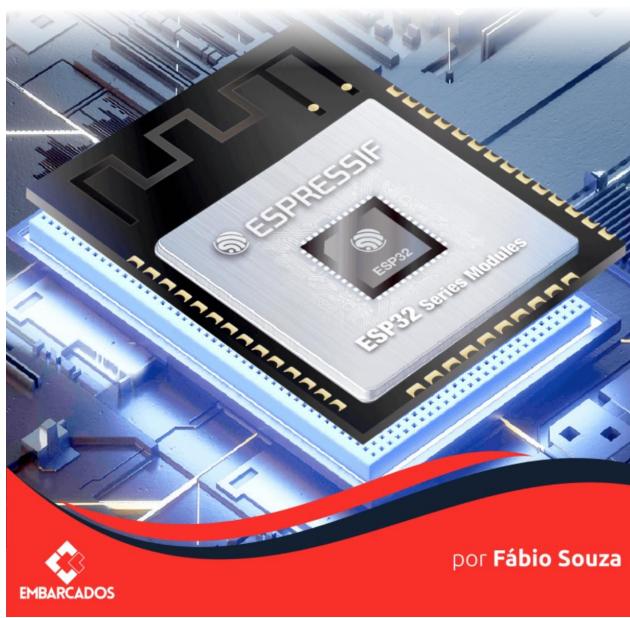
<https://github.com/espressif/esp-idf>

Ebook Gratuito

Explorando o Potencial do

ESP32

Guia de Iniciação ao **ESP-IDF 5**



<https://embarcados.com.br/e-books/e-book-explorando-o-potencial-do-esp32-guia-de-inicao-ao-esp-idf-5/>

Artigos no Embarcados



ESP32-S2 - Interrupção automática no ADC via DMA.
[Threshold]

10/01/2022 · José Moreira



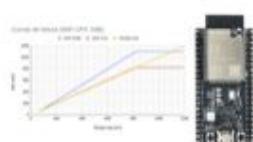
ESP32-S2: Sensor de temperatura Interno.

10/01/2022 · José Moreira



Scaling Governor - Gerenciando o clock da CPU no Linux

10/01/2022 · José Moreira



ESP32-S2 - Analisando o novo e melhor ADC

10/01/2022 · José Moreira



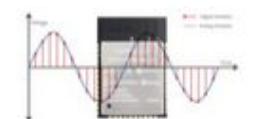
Systand - Adicionando scripts na Inicialização do Linux

10/01/2022 · José Moreira



ESP32 - Watchdog timer

10/01/2022 · José Moreira



ESP32 - Analisando e corrigindo o ADC Interno

10/01/2022 · José Moreira



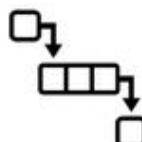
RTOS - Segurança e proteção da flash

10/01/2022 · José Moreira



RTOS: Software Timer no FreeRTOS

10/01/2022 · José Moreira



RTOS: Uso de Queue para sincronização e comunicação de tarefas

10/01/2022 · José Moreira



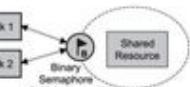
RTOS: Uso de grupo de eventos para sincronização de tarefas

10/01/2022 · José Moreira



RTOS: Semafóros para sincronização de tarefas

10/01/2022 · José Moreira



Documentação Franzininho

The image shows two side-by-side screenshots. On the left is the Franzininho documentation website, featuring a sidebar with links like 'Placa Franzininho WiFi', 'Wokwi', 'ESP-IDF', and 'Hello World! ESP-IDF'. The main content area displays a page titled 'Hello World! ESP-IDF' with text about getting started with the framework. On the right is a screenshot of a GitHub search results page for 'franzininho wifi', showing four repositories related to the Franzininho WiFi board.

Franzininho Documentação

Site ▾ Blog ▾ Contato ▾ Franzininho WiFi Wokwi ESP-IDF Hello World! ESP-IDF Entrada digital Entrada Analógica PWM com LEDC Primeiros passos com a AWS CircuitPython Arduino Placa Franzininho WiFi dev Edition Franzininho WiFi Lab01 Franzininho C0(STM32C0) Franzininho DIY FranzMaKey

Hello World! ESP-IDF

O artigo proposto dará início a uma série de projetos utilizando o framework oficial da Espressif, o IDF. O objetivo deste artigo é apresentar um exemplo de projeto que sirva como base para aqueles que estão dando seus primeiros passos na família de microcontroladores ESP32.

Como é de praxe na programação sempre que iniciamos uma nova linguagem ou quando aprendemos sobre um novo processador o primeiro projeto a ser criado é o "Olá Mundo", caso contrário teremos um eterno azar, então o que acha de começarmos a estudar e apresentar nosso projeto ao Mundo?

Ao final desse exemplo você saberá como é a estrutura básica de um programa no ESP-IDF e estará pronto para os próximos passos.

Recursos Necessários

Para esse exemplo você precisará dos seguintes recursos:

- Placa Franzininho WiFi;
- Protoboard;

Overview Repositories 62 Discussions Projects 1 Packages Teams 3 People 10

Type Language Sort

franzininho wifi

16 results for all repositories matching franzininho wifi sorted by last updated

Franzininho-WiFi-LAB01-CircuitPython Public

Exemplos de códigos baseados em CircuitPython para a Franzininho WiFi LAB01

Python 0 ⚡ 0 ⚪ 0 ⚪ 0 Updated 2 weeks ago

Franzininho-WiFi-LAB01 Public

An expansion board for the Franzininho WiFi board. This board has features such as a display, keys, and so

0 ⚡ 0 ⚪ 0 ⚪ 0 Updated 2 weeks ago

Franzininho-WiFi-LAB01-Arduino Public

Exemplos de códigos baseados em Arduino para a Franzininho WiFi LAB01

C++ Apache-2.0 0 ⚡ 0 ⚪ 0 ⚪ 0 Updated on Sep 9

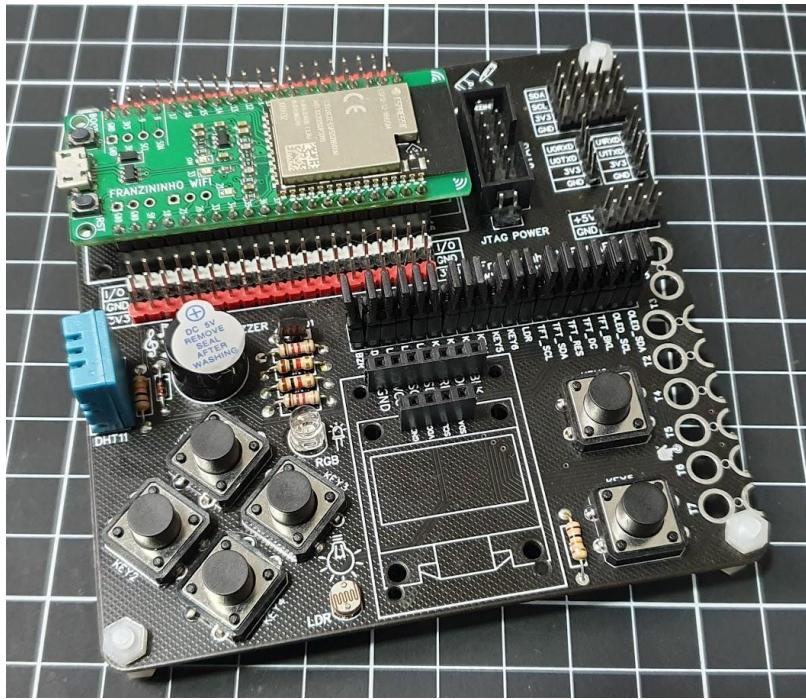
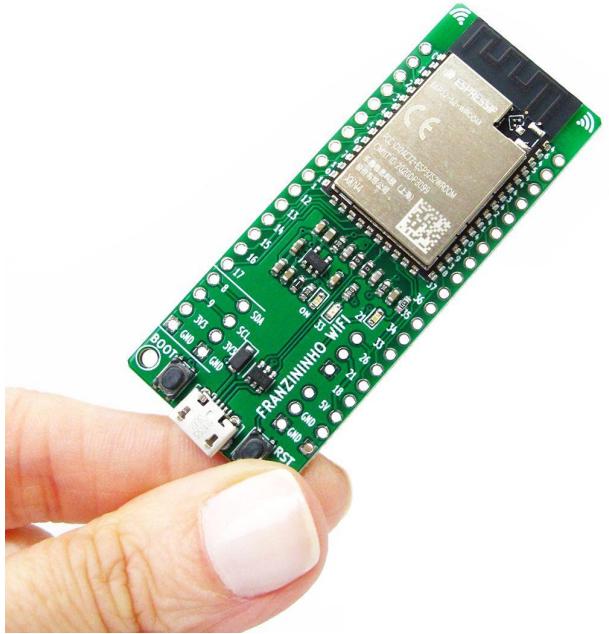
Franzininho-wifi-exemplos-esp-idf Public

Exemplos para Franzininho WiFi com ESP-IDF

C 5 ⚡ 0 ⚪ 0 ⚪ 0 Updated on Aug 23

<https://docs.franzininho.com.br>

Franzininho WiFi

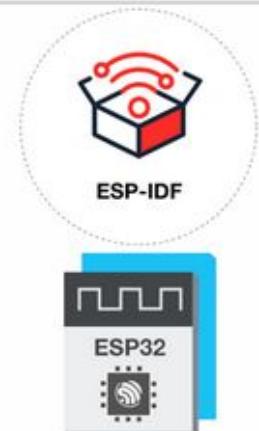


Cursos no Embarcados



Academia ESP32 Profissional

Fabio Souza



Curso: Programe o ESP32 com ESP-IDF

Fabio Souza



Curso: Programe o ESP32 com Arduino

Fabio Souza

<https://cursos.embarcados.com.br/cursos/>

Obrigado!

Contato:

- Instagram: @fabiosouza.io
- <https://linktr.ee/fabiosouza.io>