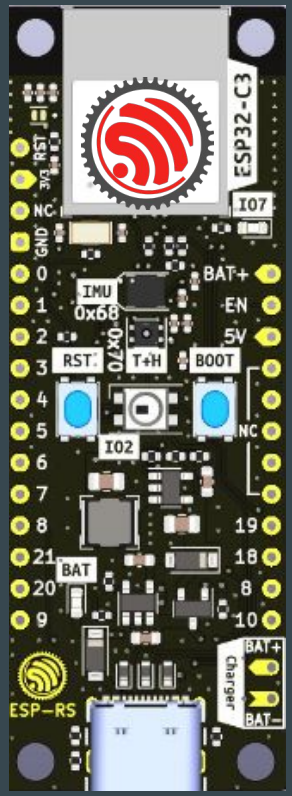
# Rust on ESP32

2025-11-03

42 Hackathon Challenge

Juraj Michálek
Espressif Systems



# https://developer.espressif.com

ESPRESSIF Developer Portal

Blog Workshops Events Products Quick Links

## esp-hal 1.0.0 release announcement

30 October 2025 · 3 mins ·

Esp32 Rust Xtensa RISCV Announcement

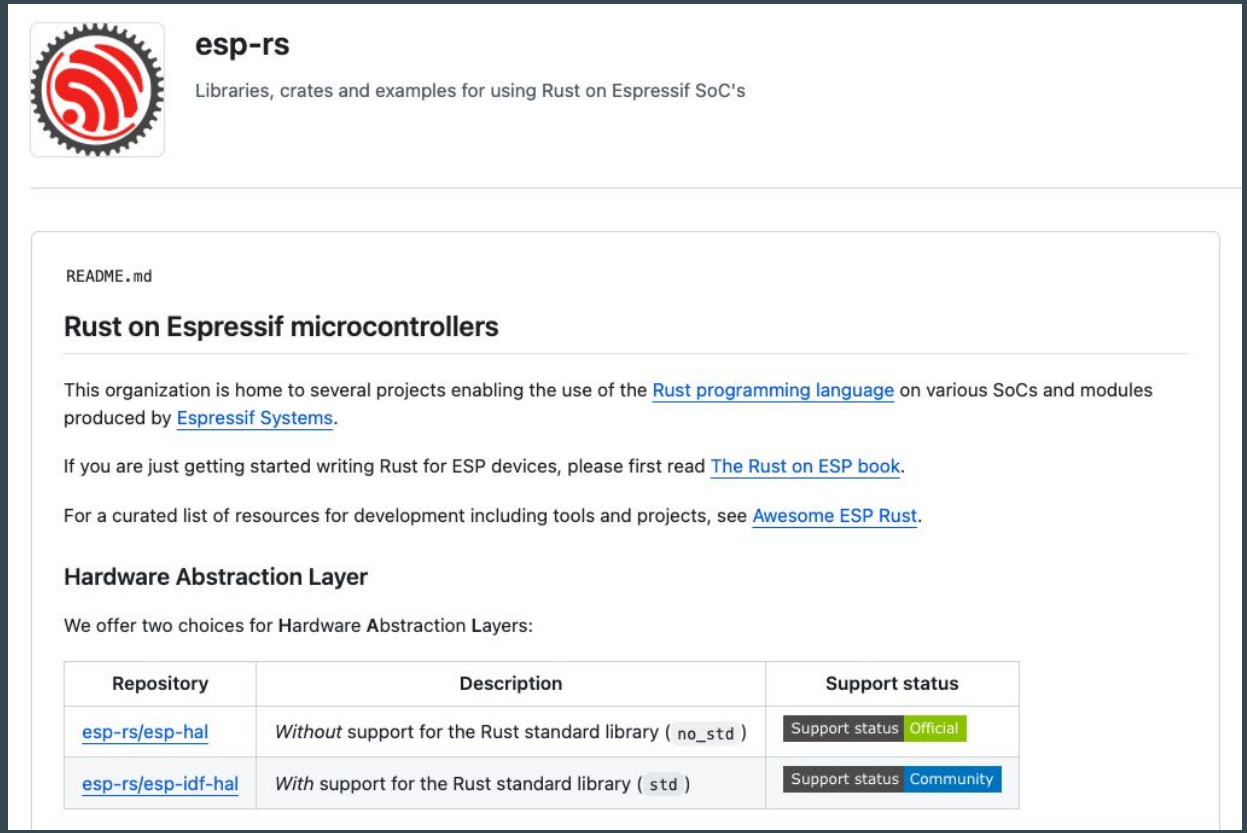
AUTHOR
Scott Mabin

▶ Table of Contents

> ***Announcing esp-hal 1.0, the first Rust SDK for embedded devices.***

In February this year, we announced the first `esp-hal` 1.0 beta release. Since then we've been hard at work, polishing and preparing for the full release. Today, the Rust team at Espressif is excited to announce the official `1.0.0` release for `esp-hal`, the *first* vendor-backed Rust SDK!

# Open Source - https://github.com/esp-rs



esp-rs

Libraries, crates and examples for using Rust on Espressif SoC's

README.md

## Rust on Espressif microcontrollers

This organization is home to several projects enabling the use of the Rust programming language on various SoCs and modules produced by Espressif Systems.

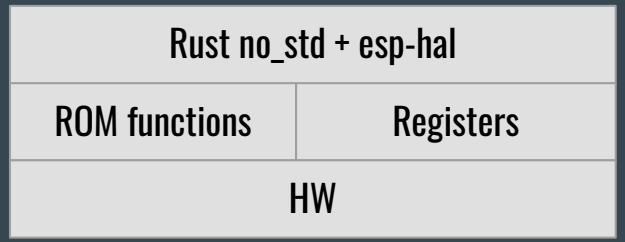
If you are just getting started writing Rust for ESP devices, please first read The Rust on ESP book.

For a curated list of resources for development including tools and projects, see Awesome ESP Rust.

### Hardware Abstraction Layer

We offer two choices for **H**ardware **A**bstraction **L**ayers:

| Repository | Description | Support status |
|---|---|---|
| esp-rs/esp-hal | *Without* support for the Rust standard library ( `no_std` ) | Support status Official |
| esp-rs/esp-idf-hal | *With* support for the Rust standard library ( `std` ) | Support status Community |





https://github.com/esp-rs/esp-hal

Official Espressif + Community Support

Requires Rust compiler

Bleeding edge MCUs and drivers might be missing

https://github.com/esp-rs/esp-idf-hal

Community support

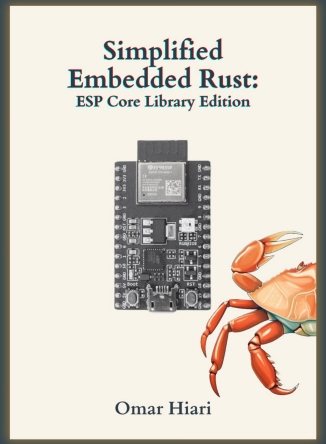
Requires Rust compiler + ESP-IDF

Bleeding edge MCUs and drivers from ESP-DF

# Training

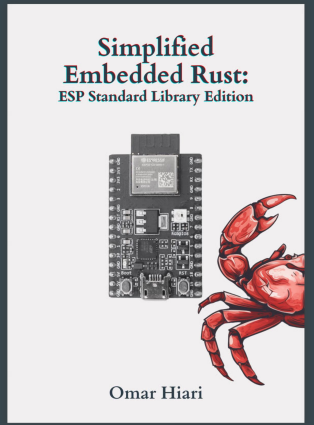
Rust no_std

- https://docs.espressif.com/projects/rust/no_std-training/
- https://github.com/esp-rs/no_std-training
- https://www.theembeddedrustacean.com/c/ser-no-std




Rust std

- https://docs.esp-rs.org/std-training/
- https://github.com/esp-rs/std-training
  - Created in cooperation with Ferrous Systems
- https://www.theembeddedrustacean.com/c/ser-std



# RISC-V vs Xtensa

RISC-V

- newer
- ESP32-C, ESP32-H, ESP32-P
- upstream Rust compiler is sufficient

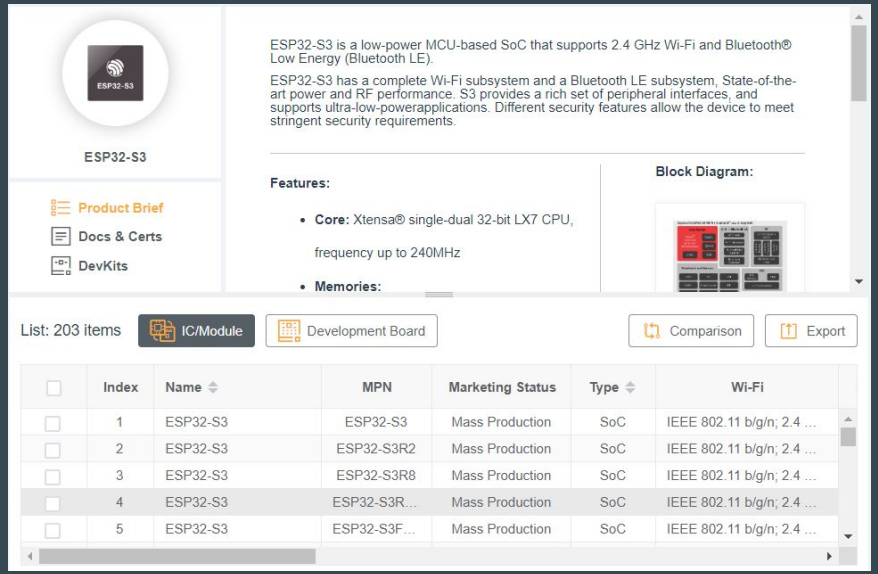
Xtensa

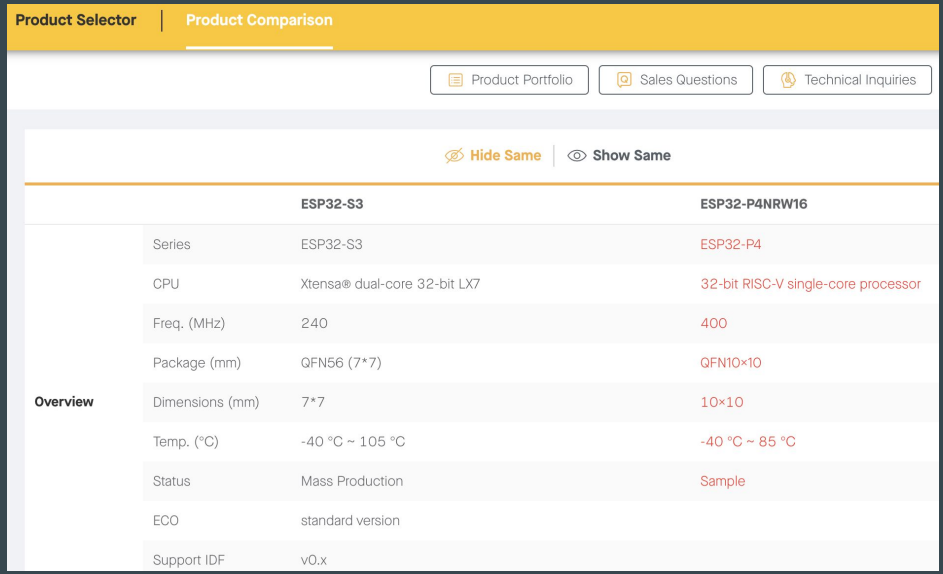
- older
- ESP32, ESP32-S2, ESP32-S3
- requires custom installation
- espup install


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

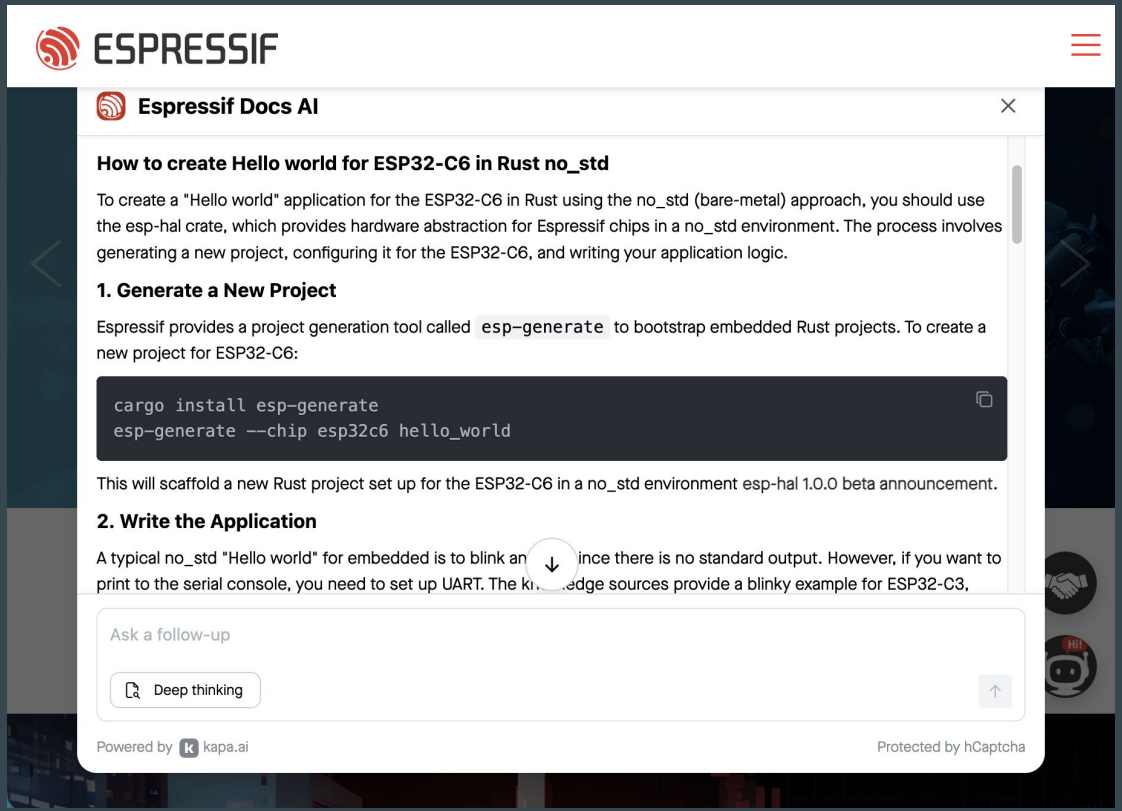
# ESP Product Selector + Product Comparison

https://products.espressif.com/

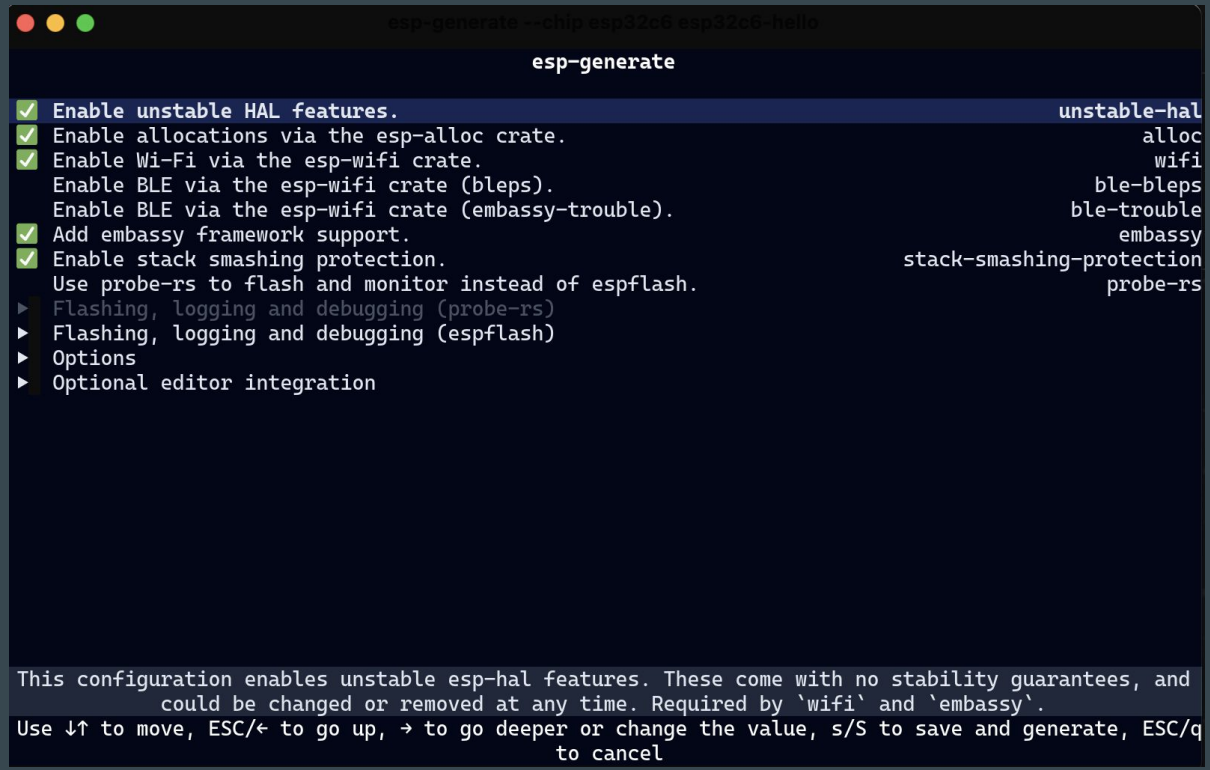


# Quick start with AI at espressif.com



# esp-generate --chip esp32c6

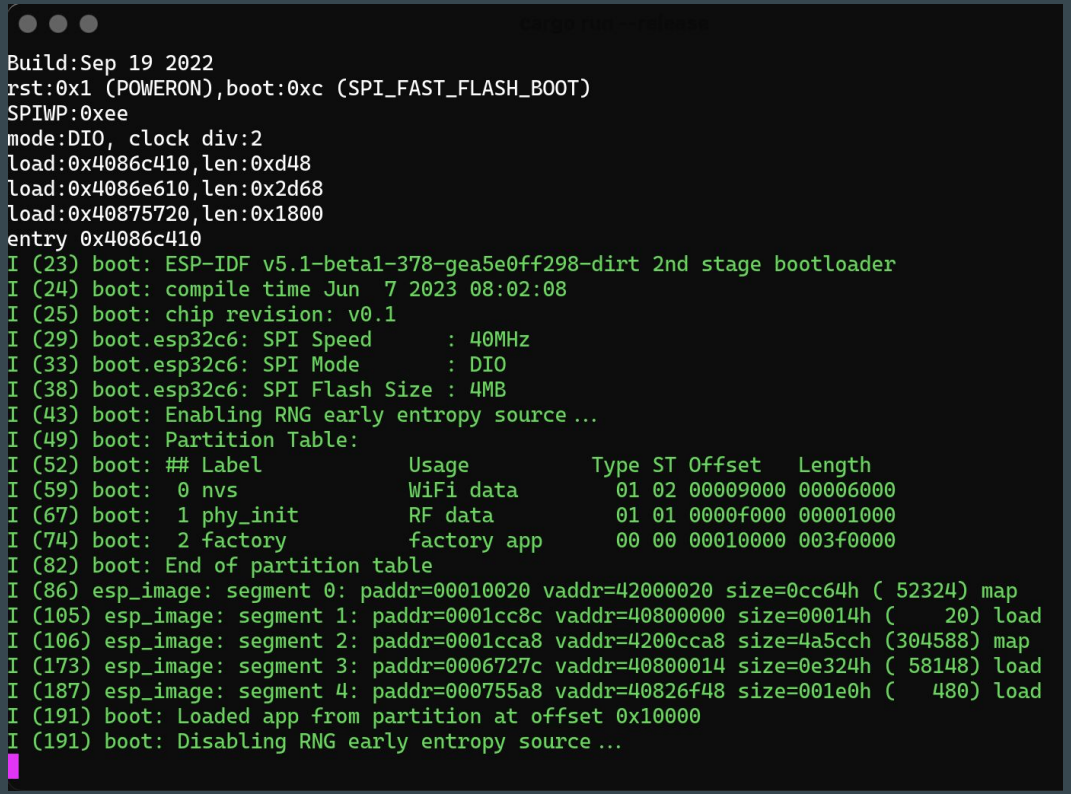


# esp-generate - more options

# Build, flash, monitor

cargo run --release

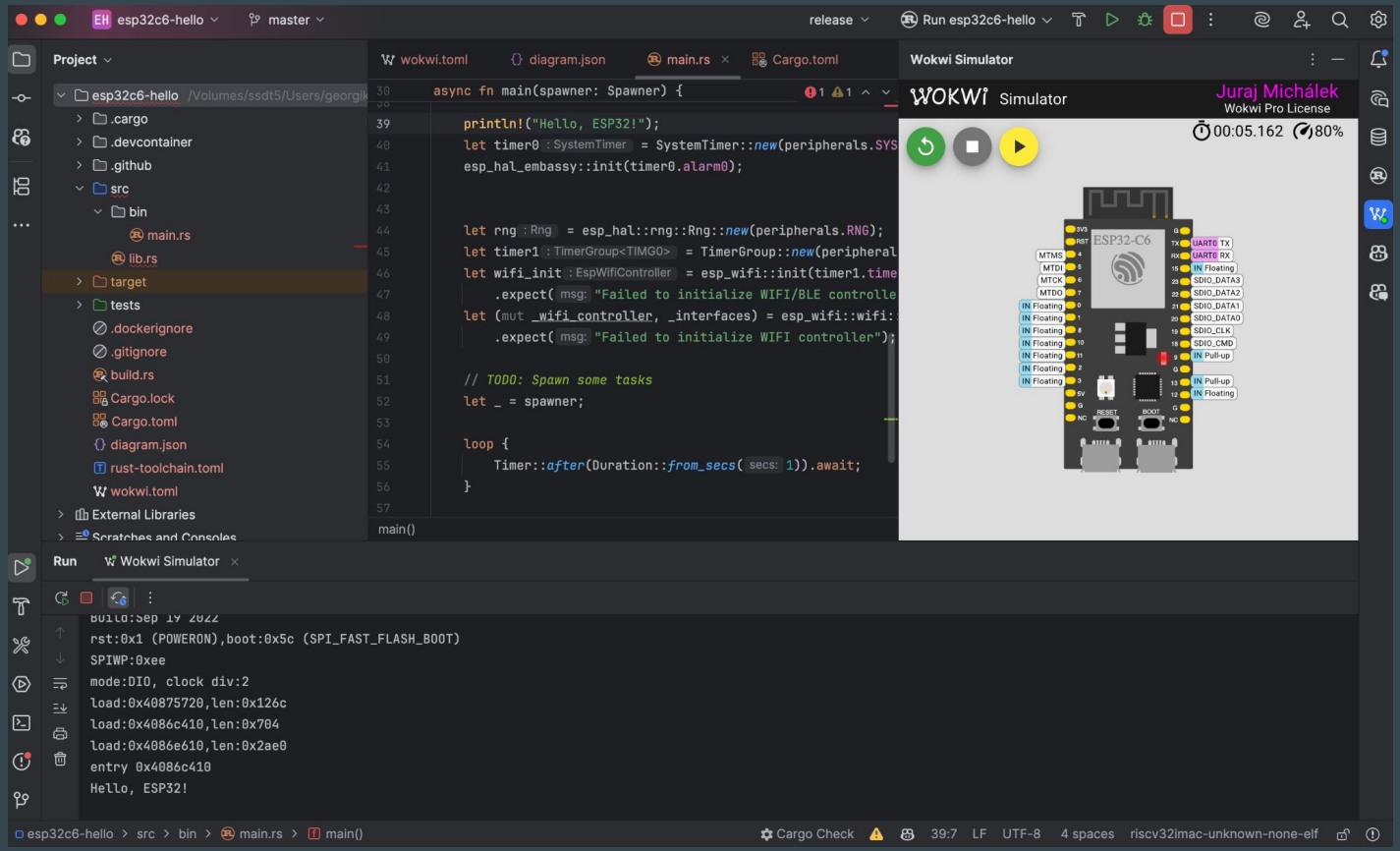
shorter: cargo r -r



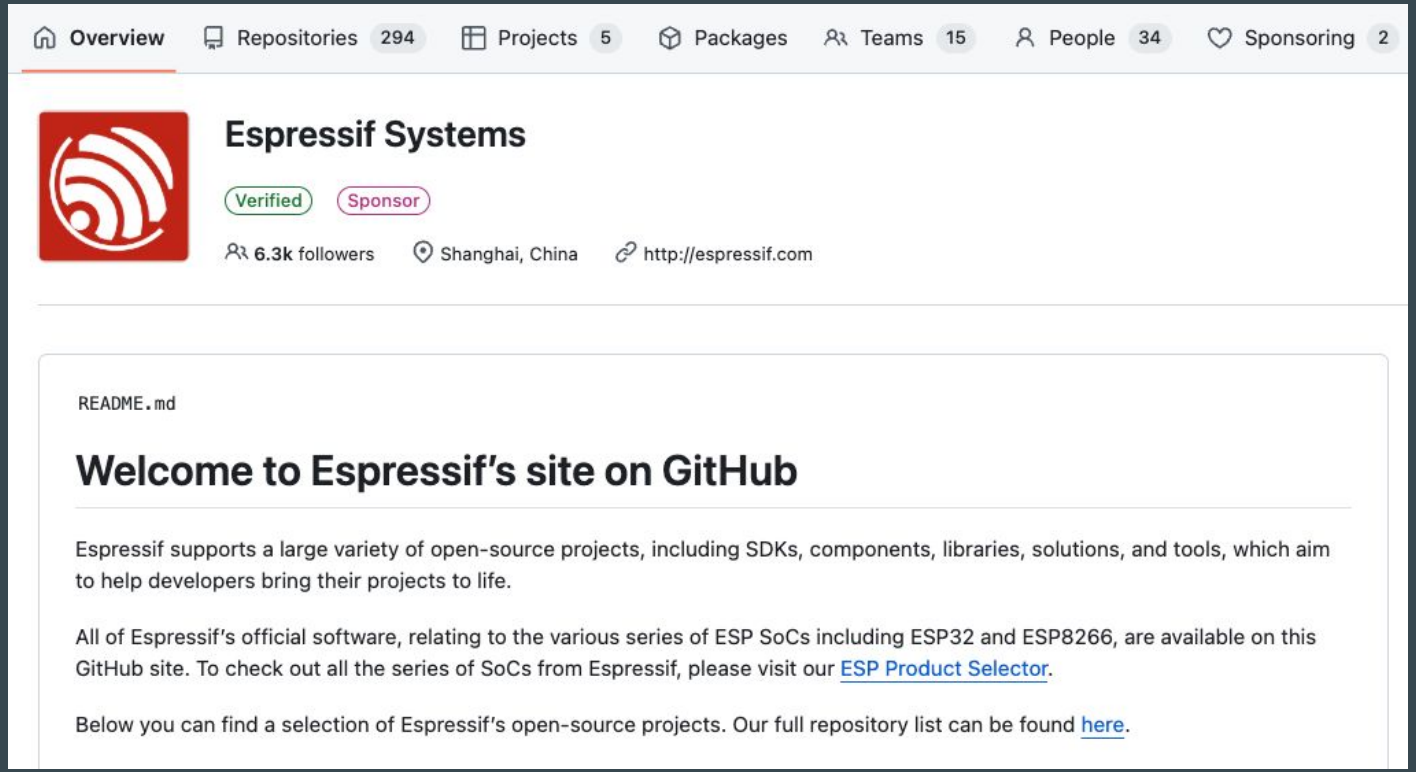
# Simulation: Update wokwi.toml from debug to release

```
[wokwi]

version = 1

gdbServerPort = 3333

#elf = "target/riscv32imac-unknown-none-elf/release/esp32c6-hello"

firmware = "target/riscv32imac-unknown-none-elf/release/esp32c6-hello"
```

# CLion with Rust plugin or RustRover



# Open Source - https://github.com/espressif



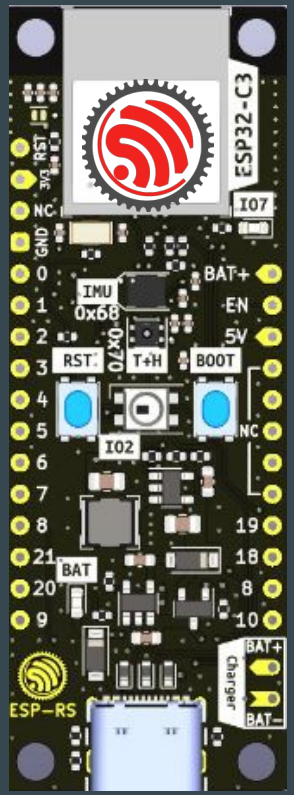
# Open Hardware - esp-rust-board

KiCad templates

https://github.com/esp-rs/esp-rust-board

ESP32-C3-DevKit-RUST-1

https://www.espressif.com/en/products/devkits



slint
GUI toolkit

Slint

Quick prototyping

Free and commercial version

# Example: List of WiFi networks

Rust std

Rust no_std

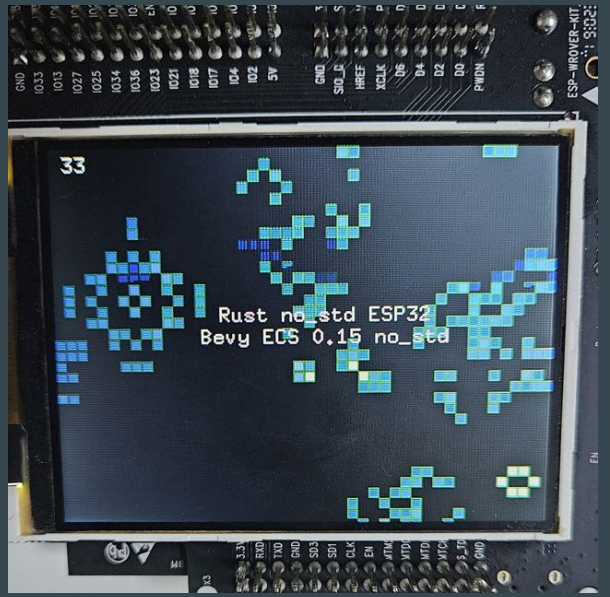
Rust no_std + Embassy

https://github.com/georgik/esp32-list-wifi-networks/

# Bevy ECS (Entity Component System)

One of biggest open source Rust projects

- https://bevyengine.org/




Rust no_std + Bevy ECS

- https://developer.espressif.com/blog/2025/04/bevy-ecs-on-esp32-with-rust-no-std/
- https://github.com/georgik/esp32-spooky-maze-game

# Rust Bare Metal - Embedded Graphics

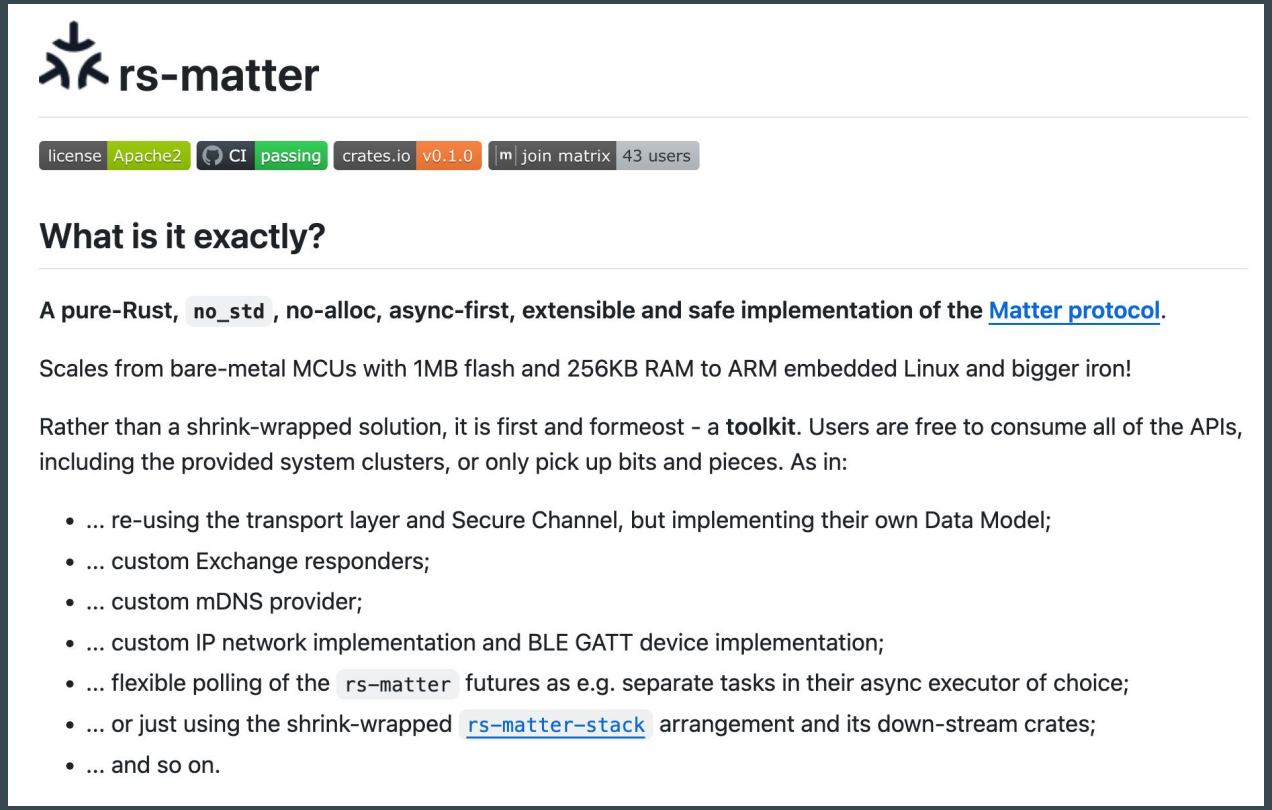

Source: https://github.com/georgik/esp32-spooky-maze-game

Idea: sharing business logic in Rust between multiple targets

Game is using Bevy ECS no_std.

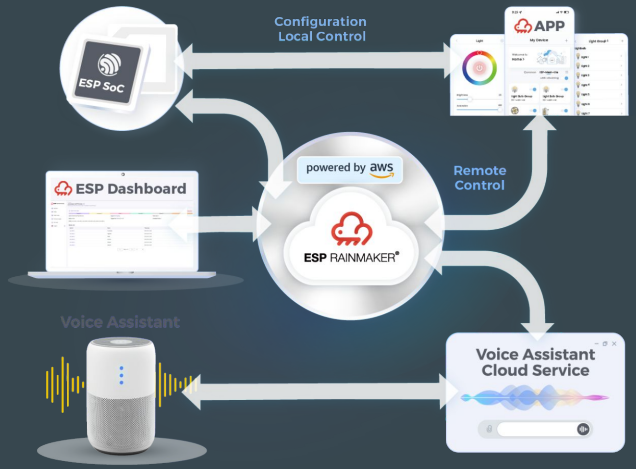

https://github.com/espressif/esp-box

# rs-matter




https://github.com/project-chip/rs-matter

# rainmaker-rs




## Rust Implementation of ESP Rainmaker

### A cross-platform implementation of ESP Rainmaker for ESP32 products and Linux using Rust.

- ESP RainMaker is end-to-end IoT development platform which enables development of IoT applications which can be controled remotely.
- However, the C based ESP RainMaker SDK(which can be found here) only supports execution on Espressif's ESP32 SOCs.
- This crate tries to implment similar functionalities for Linux platform along with ESP32(which can further be extended to other microcontrollers).

### What is working

- [x] WiFi Provisioning*:
  Providing WiFi for a new device. No need to hardcode WiFi credentials!
- [x] Remote Control:
  Controlling Devices connected to a node over internet using phone application.
- [x] User Node Association:
  Associating a specific node to a user account for control.
- [x] Device sharing:
  Share access to a device with multiple members.
- [x] Contol using Home Assistants:
  RainMaker devices can be added to and controlled using Amazon Alexa / Google Home. More details here

* Currently only supported on ESP32

https://github.com/rainmaker-rs/rainmaker

# OSes and integration

Rust no_std - https://github.com/esp-rs/esp-hal - official support

Rust std - https://github.com/esp-rs/esp-idf-hal - community support

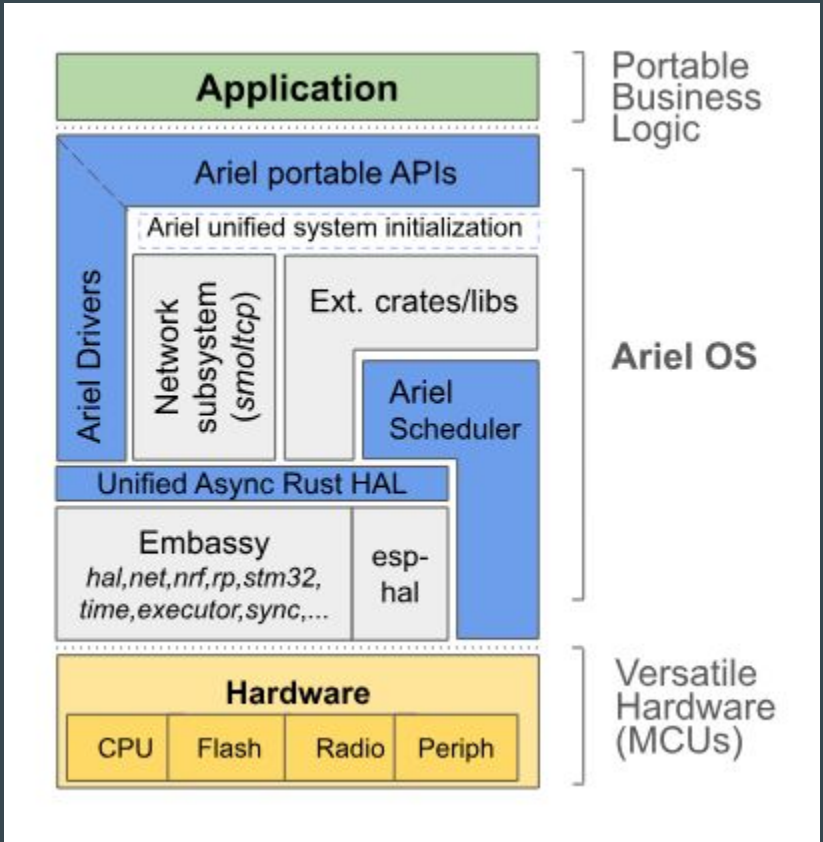
Ariel OS - https://ariel-os.github.io/ariel-os/dev/docs/book/


Zephyr - https://zephyrproject.org/

NuttX - https://nuttx.apache.org/

SVD files: https://github.com/espressif/svd

# Ariel OS




https://ariel-os.github.io/ariel-os/dev/docs/book/

# IDE for Rust development

Supported by Espressif:

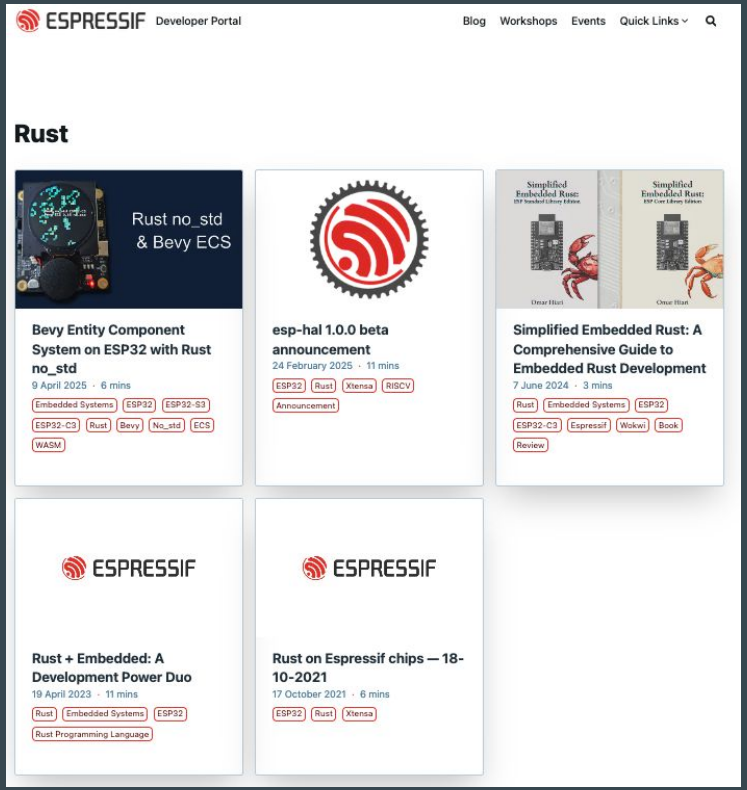
- VS Code with Espressif Extension - https://developer.espressif.com/tags/vscode/

Supported by JetBrains:

- CLion - https://developer.espressif.com/blog/clion/
- RustRover - https://www.jetbrains.com/rust/


# Developer Portal

developer.espressif.com




GitHub: https://github.com/espressif/developer-portal/

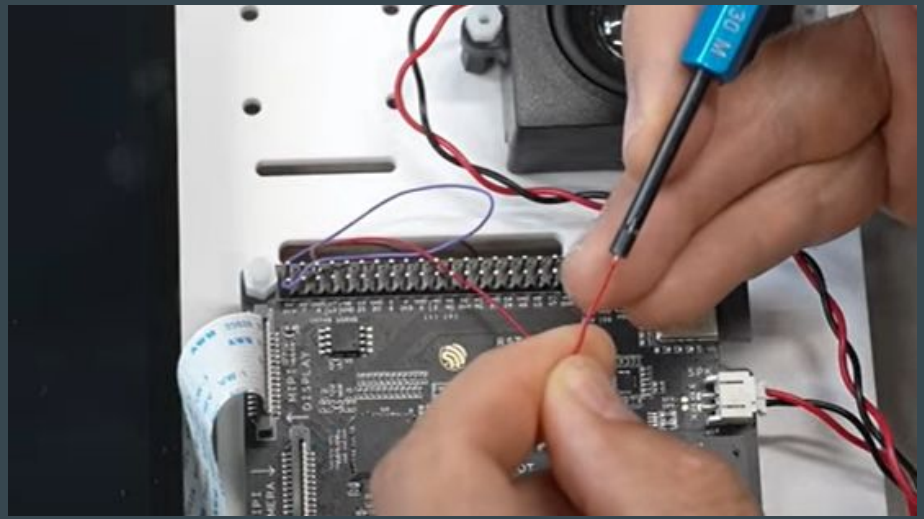
# How to “unbrick” ESP32

Many people incorrectly thinks that ESP32 is bricked, in many cases the custom firmware just does not receive UART or USB signals.

Solution: hold BOOT button and press RESET to switch to boot mode

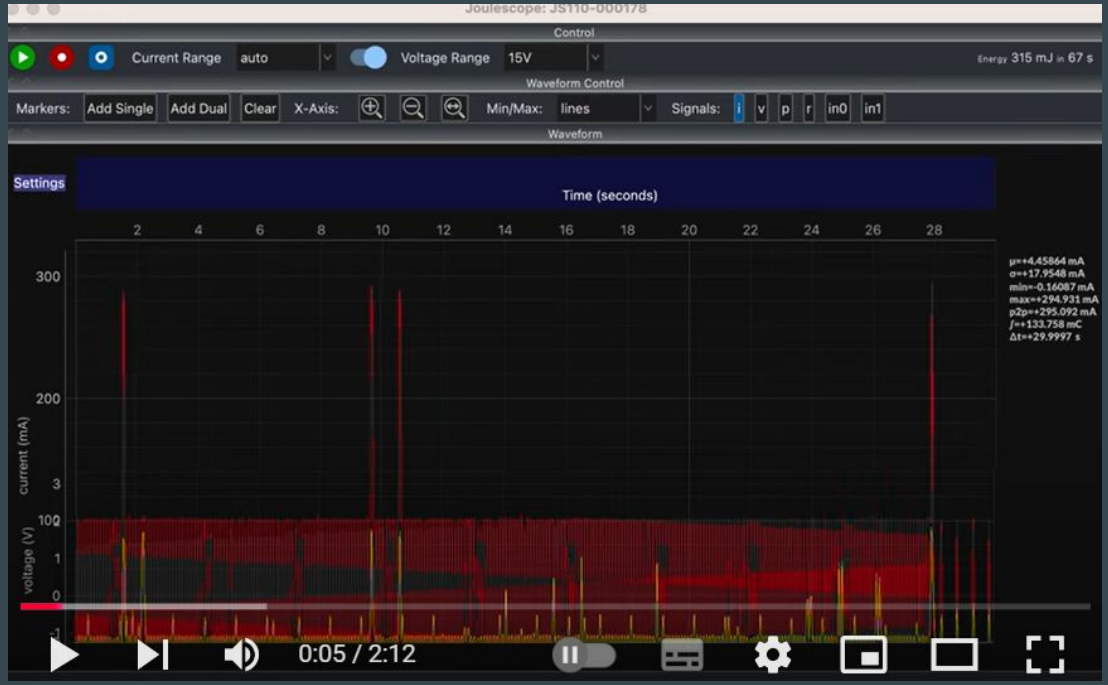
After flashing, just press RESET to return to normal mode.

# Wire Wrap Tool



Espressif DevCon 2024 - Tips and Tricks

# ESP32-C6 TWT




https://www.youtube.com/watch?v=FA1jqZLig4s

# Espressif Developer Conference 2022-2024 - recording


https://www.youtube.com/@EspressifSystems

https://devcon.espressif.com/

# Embedded World 2026

Meet us in Nuremberg, Germany


# Elektor Mag

## Special Guest Edition

https://www.elektor.com/products/elektor-special-espressif-guest-edition-2023-pdf-en

# Espressif in Brno

Vlněna Office Park

Espressif Systems (Czech) s.r.o.

Přízova 3, 602 00 Brno

Czechia, Europe