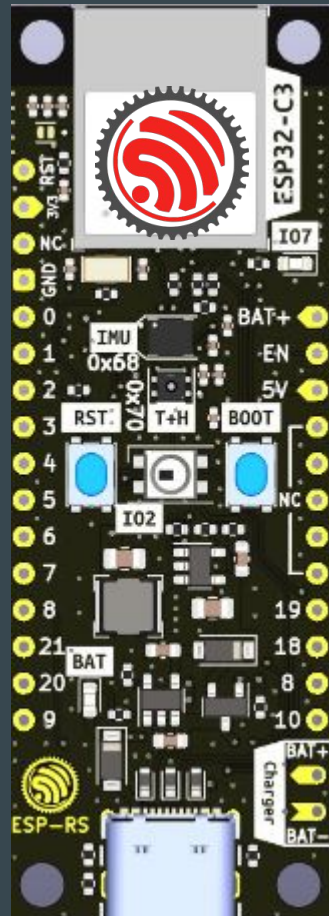


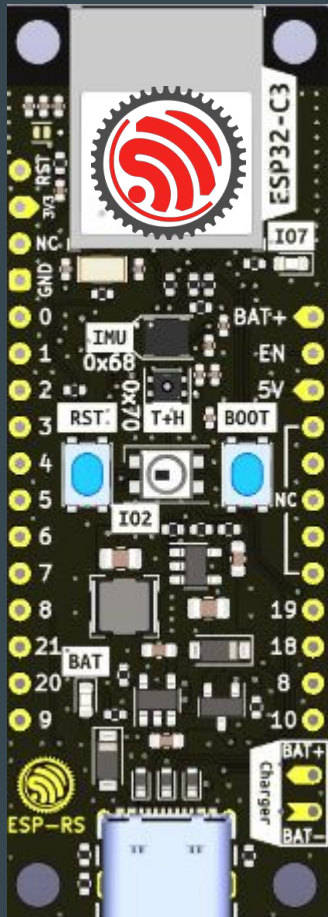
With Open Source Community
and Espressif
made it possible to use
Rust language
for ESP32 development

2022-09-18
OpenAlt - Brno
Juraj Michálek



wokwi.com/rust

Quick start



Context

ESP8266, ESP32 +S family, ESP32-C3
Xtensa, RISC-V

Arduino, C, MicroPython, CircuitPython, Toit, C#
based on ESP-IDF or bare metal

Early days of Rust support

Sep 10, 2019 - 6 min read

Rust on the ESP32



About six months ago, I made a [post on reddit](#) highlighting the launch of Espressif's llvm xtensa fork, not too long after, I had a working `rustc` toolchain capable of generating xtensa assembly. At this point I had to put this project to the side to finish my final year of university. Funnily enough I didn't stray too far, my final year project used Rust to create a '[smartwatch](#)' (I may write about this in the future, if anyone is interested).

<https://mabez.dev/blog/posts/>

mabez - 2019 - 2020

2020

Rust on the ESP32 & ESP8266 - Building an ecosystem

Sep 15

Rust on the ESP32 - SVD's, PAC's and USB flashing

Jan 13

2019

Rust on the ESP32

Sep 10

<https://mabez.dev/blog/posts/>

ctron - 2019 - A tale of Rust, the ESP32 and IoT

Telemetry data for Eclipse Hono



ESP-RS organisation <https://github.com/esp-rs>



esp-rs

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

Challenges in those days

Complex setup

Toolchain not available for all platforms

Just few things working

Espressif - let's make open source better

Building Rust team within Espressif and work with community.

Establishing the 1st Rust ESP32 Community meeting

2021-07-13 - <https://github.com/esp-rs/rust/discussions/53>

Regular meeting every two weeks

2021-09-09 rust-build repo

Going from manual builds to automatic

Open sourcing deployment tooling

Publishing artifact for all major OSes

Installers

std vs no_std

std based on ESP-IDF

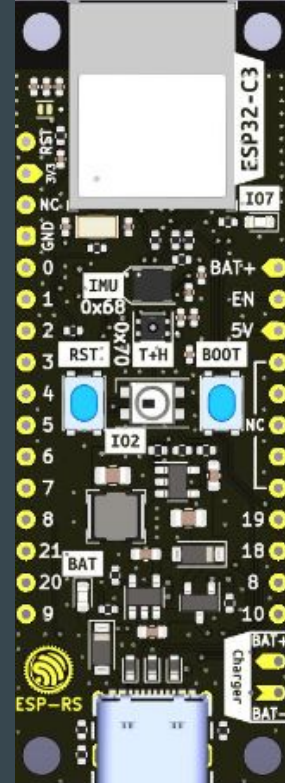
no_std a.k.a bare metal

<https://github.com/espressif/svd>

Designing Open Hardware - esp-rust-board

KiCad templates

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



Training Embedded for ESP32-C3 by Ferrous Systems

Training: <https://ferrous-systems.com/training/#package-espressif-beginner-training>

Material: <https://espressif-trainings.ferrous-systems.com/>

GitHub: <https://github.com/ferrous-systems/espressif-trainings/>



ferrous systems



Simulate IoT Projects in Your Browser

[Discord Community](#)

[Facebook Group](#)

Online Embedded Rust Simulator

Use Wokwi to simulate embedded Rust projects and test your hardware projects.

Rust project examples

+ NEW PROJECT

```
//! Blinks an LED
//!
//! This assumes that a LED is connected to GPIO4.
//! Depending on your target board you are using you should change
//! If your board doesn't have a LED, don't forget to add an appropriate
//!
// rebuild

use std::thread;
use std::time::Duration;
```



esp32-blink.rs

```
{
  "version": 1,
  "author": "Sergio Gasquez Arcos",
  "editor": "wokwi",
  "parts": [
    {
      "type": "board",
      "board": "esp32-s2-devkitm-1",
      "pin": "gpio",
      "top": 0,
      "left": 0,
      "attrs": { "builder": "esp32-s2-rust" }
    }
  ]
}
```



esp32s2-blink.rs

```
//! Blinks an LED
//!
//! This assumes that a LED is connected to GPIO4.
//! Depending on your target board you are using you should change
//! If your board doesn't have a LED, don't forget to add an appropriate
//!
// use std::thread;
// use std::time::Duration;

use embedded_hal::digital::blocking::OutputPin;
use esp32c3::peripherals::Peripherals;
```



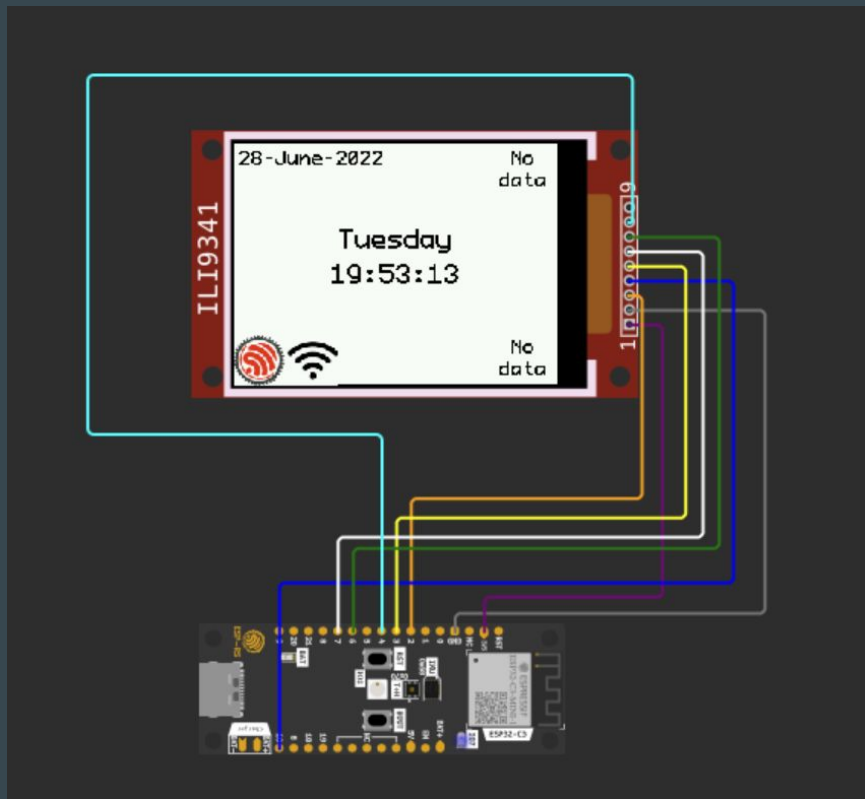
esp32c3-blink.rs

<https://wokwi.com/rust>
<https://github.com/wokwi>

GitPod.io + Wokwi.com

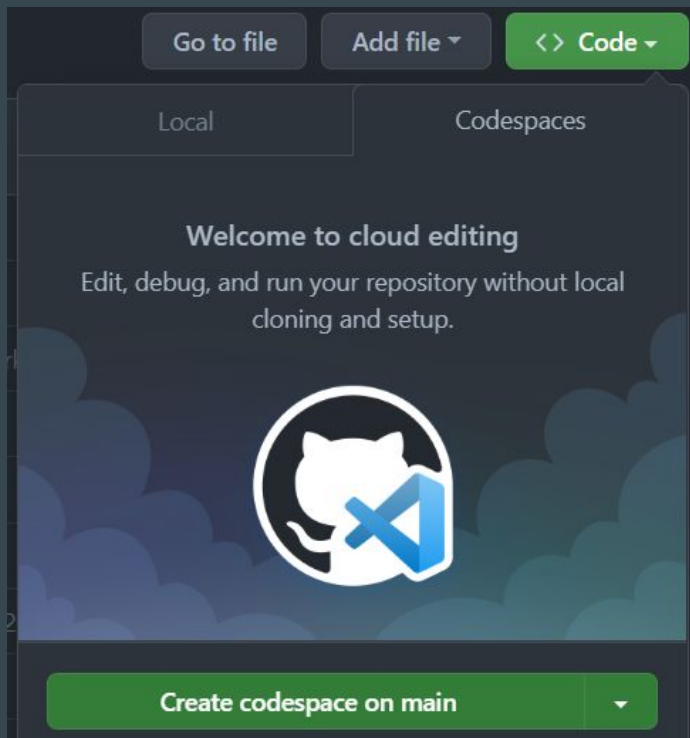


Open in Gitpod



<https://github.com/playfulFence/esp-clock#dev-containers>

GitHub - CodeSpaces

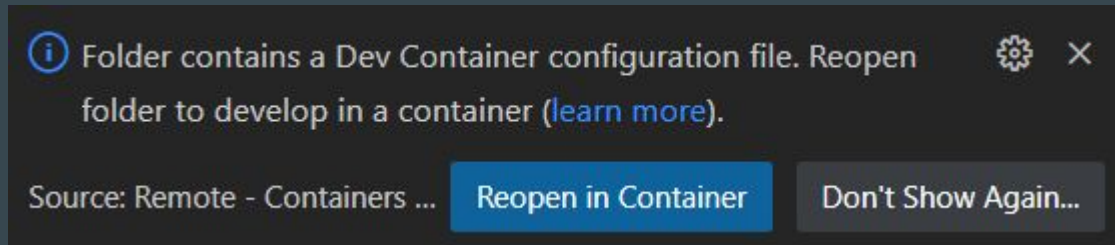


Development Containers



podman

Works Podman, Docker, Lima



Note: Some technologies might have limitations

cargo generate - create new project

cargo generate <https://github.com/esp-rs/esp-template>

cargo generate --vcs none --git https://github.com/esp-rs/esp-idf-template cargo

Some links

The Rust on ESP Book - <https://esp-rs.github.io/book/>

Awesome ESP Rust - <https://github.com/esp-rs/awesome-esp-rust>

- submit your project :-)

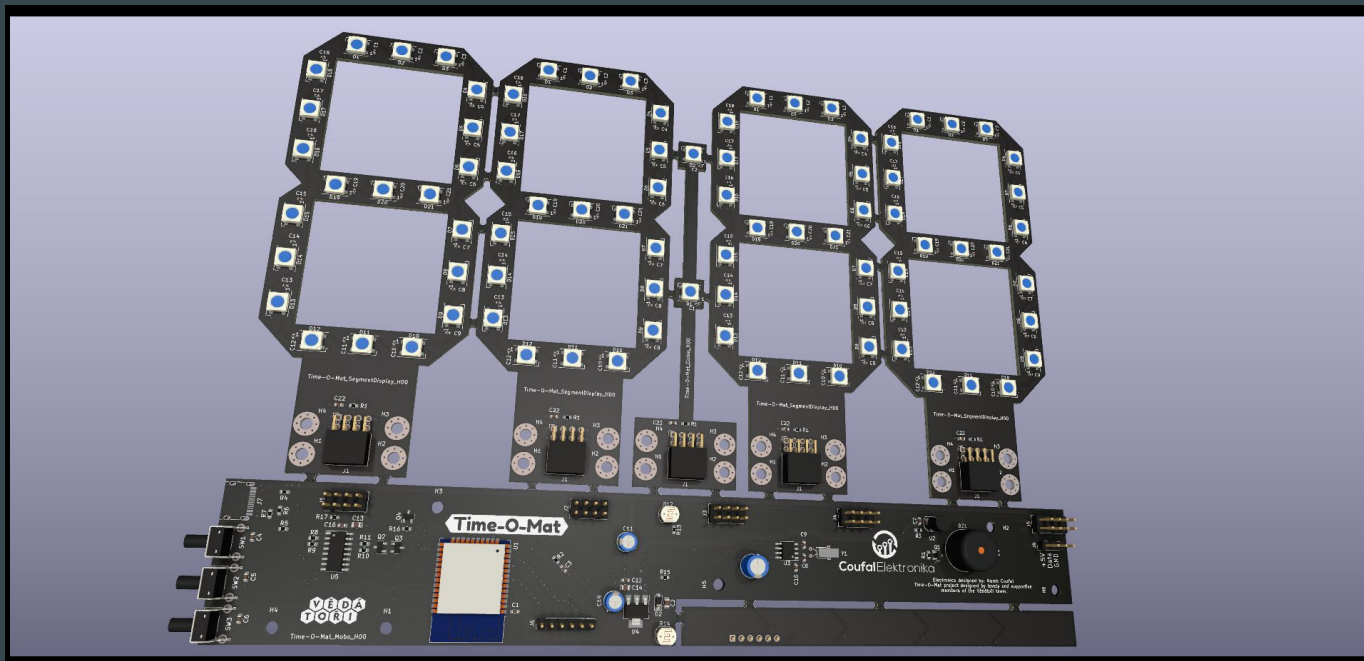
ESP-IDF-HAL and ESP-HAL examples:

- <https://github.com/esp-rs/esp-hal/tree/main/esp32-hal/examples>

**Some Inspiration
(not necessary in Rust)**

Time-O-Mat

<https://github.com/vedatori/Time-O-Mat>



Grafana



<https://grafana.com/blog/2020/06/17/how-to-monitor-a-sourdough-starter-with-grafana/>

<https://github.com/grafana/diy-iot> - Arduino now. We're not Rust yet :)

ESP32-S3-BOX



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

Many chips, many boards - quick help

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



ESP32-S3

 Product Brief

 Docs & Certs

 DevKits

ESP32-S3 is a low-power MCU-based SoC that supports 2.4 GHz Wi-Fi and Bluetooth® Low Energy (Bluetooth LE).

ESP32-S3 has a complete Wi-Fi subsystem and a Bluetooth LE subsystem, State-of-the-art power and RF performance. S3 provides a rich set of peripheral interfaces, and supports ultra-low-power applications. Different security features allow the device to meet stringent security requirements.

Features:

- **Core:** Xtensa® single-dual 32-bit LX7 CPU, frequency up to 240MHz
- **Memories:**

Block Diagram:



List: 203 items

 IC/Module

 Development Board

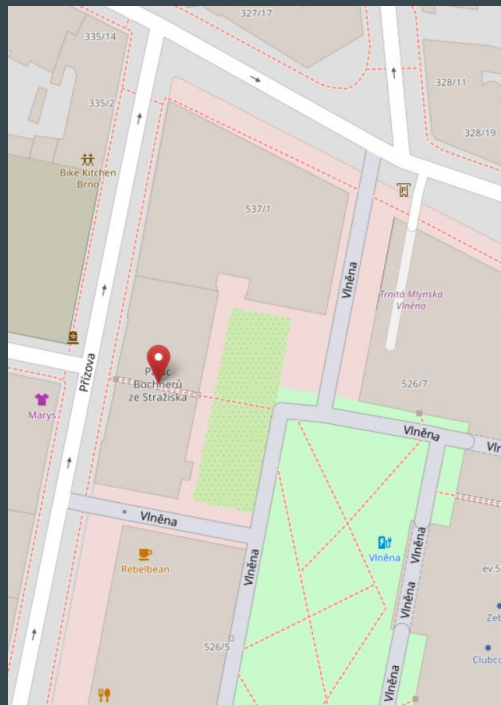
 Comparison

 Export

☐	Index	Name	MPN	Marketing Status	Type	Wi-Fi
☐	1	ESP32-S3	ESP32-S3	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...
☐	2	ESP32-S3	ESP32-S3R2	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...
☐	3	ESP32-S3	ESP32-S3R8	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...
☐	4	ESP32-S3	ESP32-S3R...	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...
☐	5	ESP32-S3	ESP32-S3F...	Mass Production	SoC	IEEE 802.11 b/g/n; 2.4 ...

Where you can find us?

Prízova 3, 602 00 Brno



<https://makerfaire.cz/brno/>

