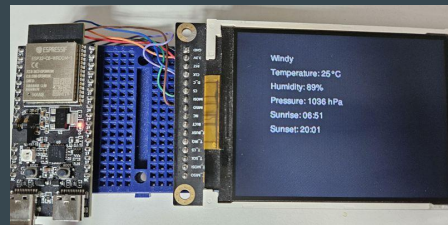


Think Different, Even on Microcontrollers Embedded Swift on ESP32

2025-06-23

Swift Server Meetup #5 - Online

Juraj Michálek
Espressif Systems



Content

Why Embedded Swift might be good fit for IoT solution

Embedded Swift with RISC-V base Espressif MCUs

- examples with LED, RGB LED, display

Embedded Swift Bare Metal

Why Embedded Swift makes sense?

OSes and integration in Production



ESP-IDF (OS based on FreeRTOS) - <https://github.com/espressif/esp-idf>

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

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



Zephyr - <https://zephyrproject.org/>

- <https://developer.espressif.com/tags/zephyr/>
- <https://wokwi.com/projects/325751453458104914>



NuttX - <https://nuttx.apache.org/> (as app, Linux-like OS)

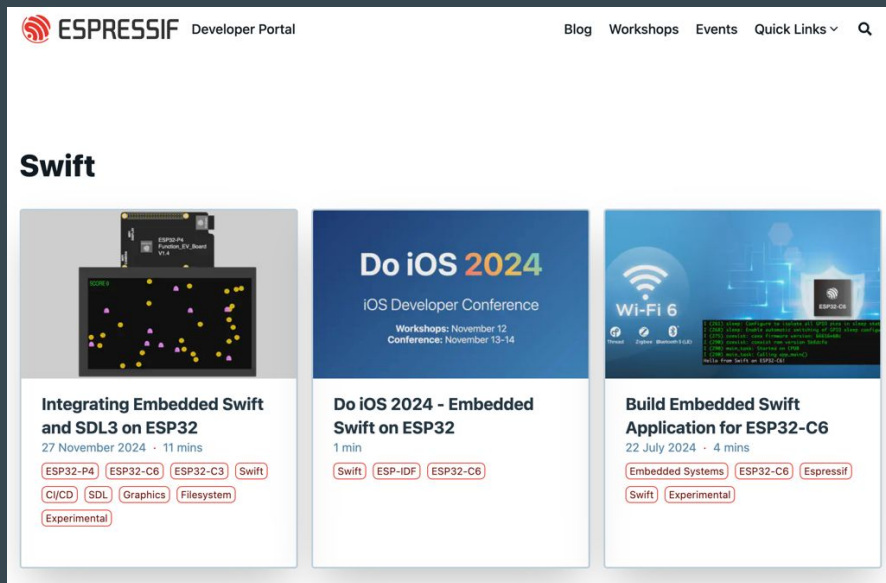
- <https://developer.espressif.com/tags/nuttx/>

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

Cyber Resilience Act - Language Fit

Rust	 Excellent	Memory safety, OTA-ready, ecosystem-ready
Swift	 Promising	Needs maturing runtime/tooling; good safety/readability
C/C++	 High burden	Requires external tools, extensive QA to meet safety standards Poor memory safety - frequent source of CVEs
Python	 Weak fit	Dynamic runtime, opaque interpreter, poor auditability

Developer Portal



developer.espressif.com/tags/swift

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

ESP32 at Do iOS 2024 - Amsterdam



<https://do-ios.com/>

<https://youtu.be/HFnAvDZtW4A?si=NmlpZ53aoVsfAziW>

Swift Embedded Examples

```
idf.py set-target esp32-c6
```

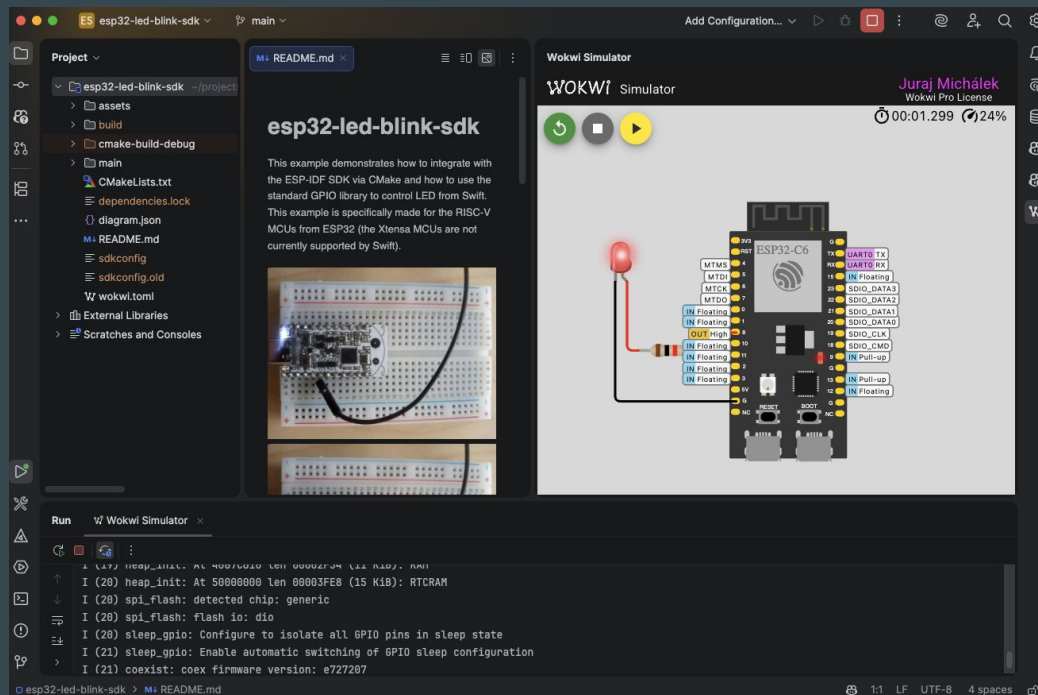
```
idf.py build flash monitor
```

```
Quit monitor: CTRL+]
```

<https://github.com/swiftlang/swift-embedded-examples>

Swift Embedded Examples - esp32-led-blink-sdk

CLion + Wokwi plugin



<https://plugins.jetbrains.com/plugin/23826-wokwi-simulator>

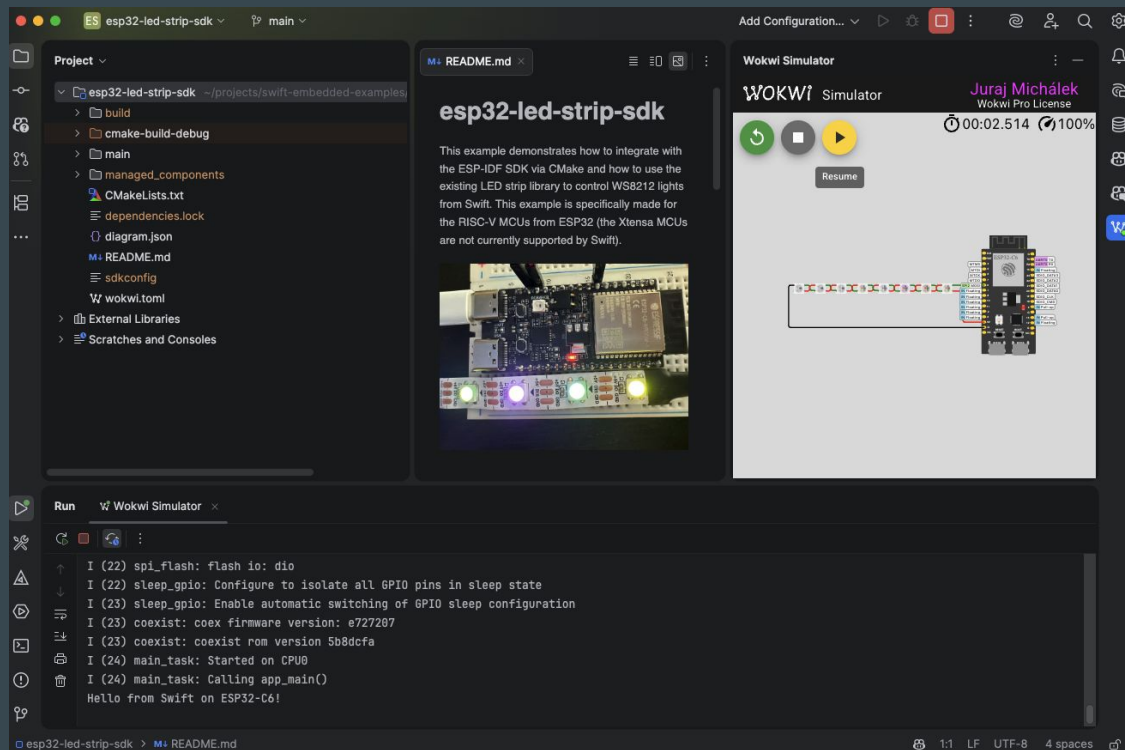
Swift Embedded Examples - esp32-led-strip-sdk

Architecture of ESP-IDF project:

Components

- main
- managed_components
- components (local)

main/idf_component.yml



Component Registry

The **ESP** Component Registry

Discover, download and publish components and examples for ESP-IDF



Browse components

ALL Board Support Package

Compatible with ESP-IDF: v5.0 v5.1 v5.2 v5.3

By target: ESP32 ESP32-C2 ESP32-C3 ESP32-C5 ESP32-C6 ESP32-C61 ESP32-H2 ESP32-P4 ESP32-S2 ESP32-S3

Featured

espressif/mdns

v1.4.0

uploaded 2 months ago

mDNS

lvgl/lvgl

v9.2.0

uploaded 1 month ago

LVGL - Light and Versatile Graphics Library

espressif/esp-modbus

v1.0.16

uploaded 1 month ago

ESP-MODBUS is the official Modbus library for Espressif SoCs.

joltwallet/littlefs

v1.14.8

uploaded 3 months ago

LittleFS is a small fail-safe filesystem for micro-controllers.

espressif/arduino-esp32

v3.0.7

uploaded 23 hours ago

Arduino core for ESP32, ESP32-S and ESP32-C series of SoCs

espressif/openai

v1.0.0

uploaded 5 months ago

OpenAI library compatible with ESP-IDF

wolfssl/wolfssl

v5.7.2

uploaded 3 months ago

wolfSSL Embedded SSL/TLS Library

slint/slint

v1.8.0

uploaded 1 month ago

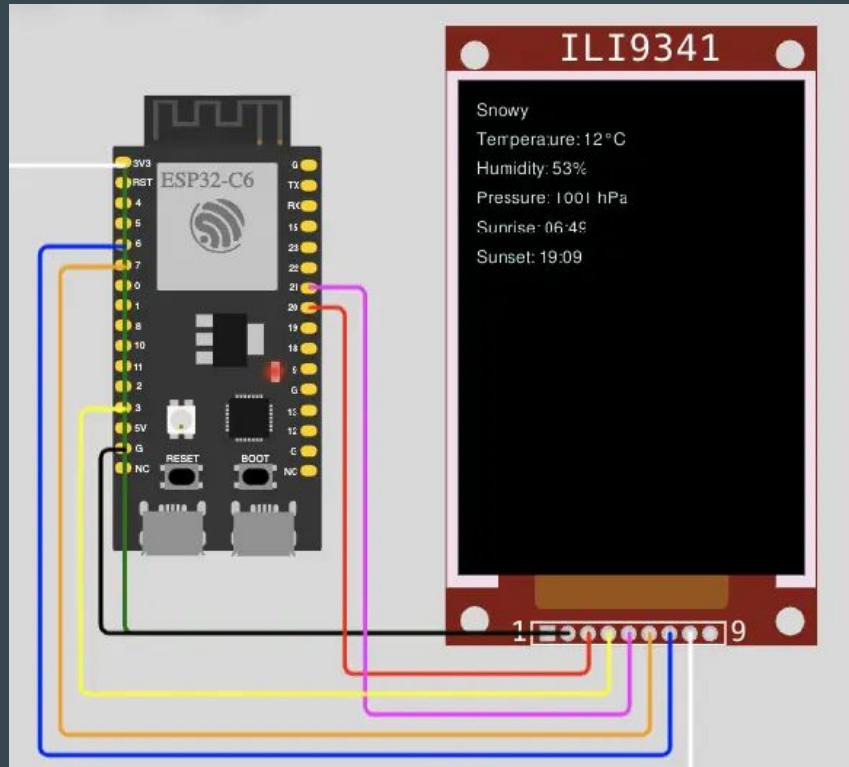
Slint — declarative GUI toolkit

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

ESP32 Weather Display

Display connected via SPI

Board Support Package - using Generic



<https://github.com/georgik/esp32-swift-weather-display>

Online [simulation with Wokwi](#)

Smooth integration with C libraries - SDL3

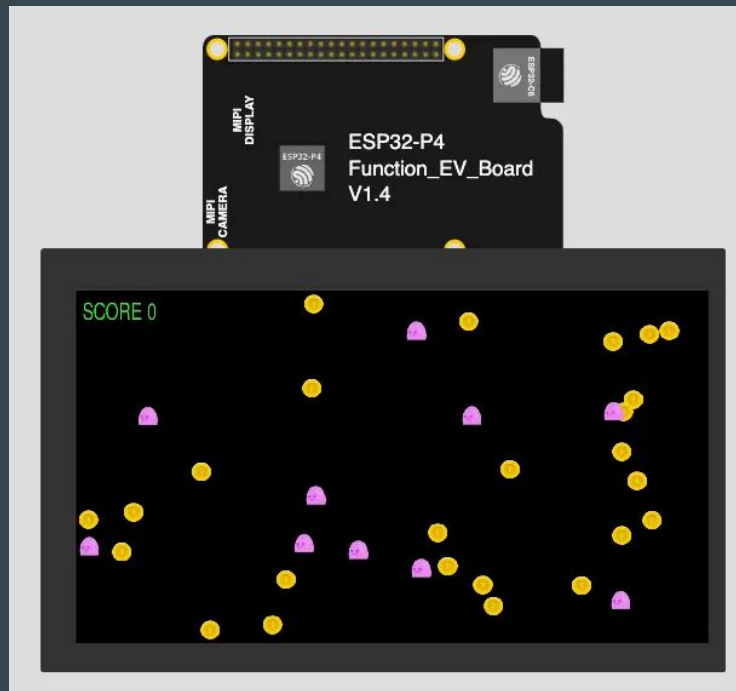
Example with high performance ESP32-P4

RISC-V IMACF

<https://github.com/georgik/esp32-sdl3-swift-example>

Online [simulation with Wokwi](#)

[ESP-IDF SDL3 Component](#)



Integrating with Matter ecosystem - basic level

<https://github.com/swiftlang/swift-matter-examples>

```
let rootNode = Matter.Node()
```

```
rootNode.identifyHandler = {
```

```
    print("identify")
```

```
}
```

```
// (2) Create a "light" endpoint
```

```
let lightEndpoint = Matter.ExtendedColorLight(node: rootNode)
```

```
lightEndpoint.eventHandler = { event in
```

```
    print("lightEndpoint.eventHandler:")
```

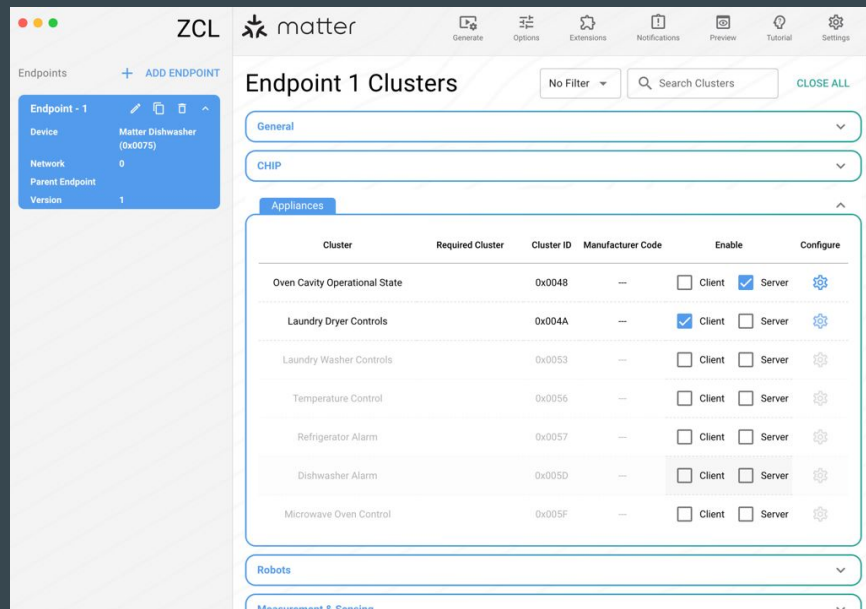
Integrating with Matter ecosystem - pro level

Matter Data Model Interpreter

<https://github.com/espressif/matter-data-model-interpreter/>

<https://github.com/project-chip/connectedhomeip>

<https://github.com/project-chip/zap>



Supported MCUs

All modern RISC-V: ESP32-C*, ESP32-H*, ESP32-P* family

- easy use of nightly upstream toolchain

More complex setup: ESP32, ESP32-S* family

- requires build of custom toolchain with Xtensa support
- this MCUs are older

Embedded Swift - Bare Metal

CMSIS System View Description (CMSIS-SVD)

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

Patched version used in Rust no_std

- <https://github.com/esp-rs/esp-pacs/>
- <https://github.com/apple/swift-mmio>

SVD2Swift \

--input ../esp-pacs/esp32c6/svd/esp32c6.svd \

--output Sources/Registers \

--access-level public \

--peripherals GPIO UART0 SPI0

ESP32-C6 Embedded Swift Bare Metal

- Sources/Registers from MMIO
- Call of ROM functions
 - `// ESP32-C6 ROM UART functions`
 - `@_silgen_name("esp_rom_uart_putc")`
 - `func esp_rom_uart_putc(_ char: UInt8)`

Linker Scripts

- Sources/Support/Ld/esp32c6/linkall.x
- Minimal size of section 64KB

ESP App Descriptor for Bootloader

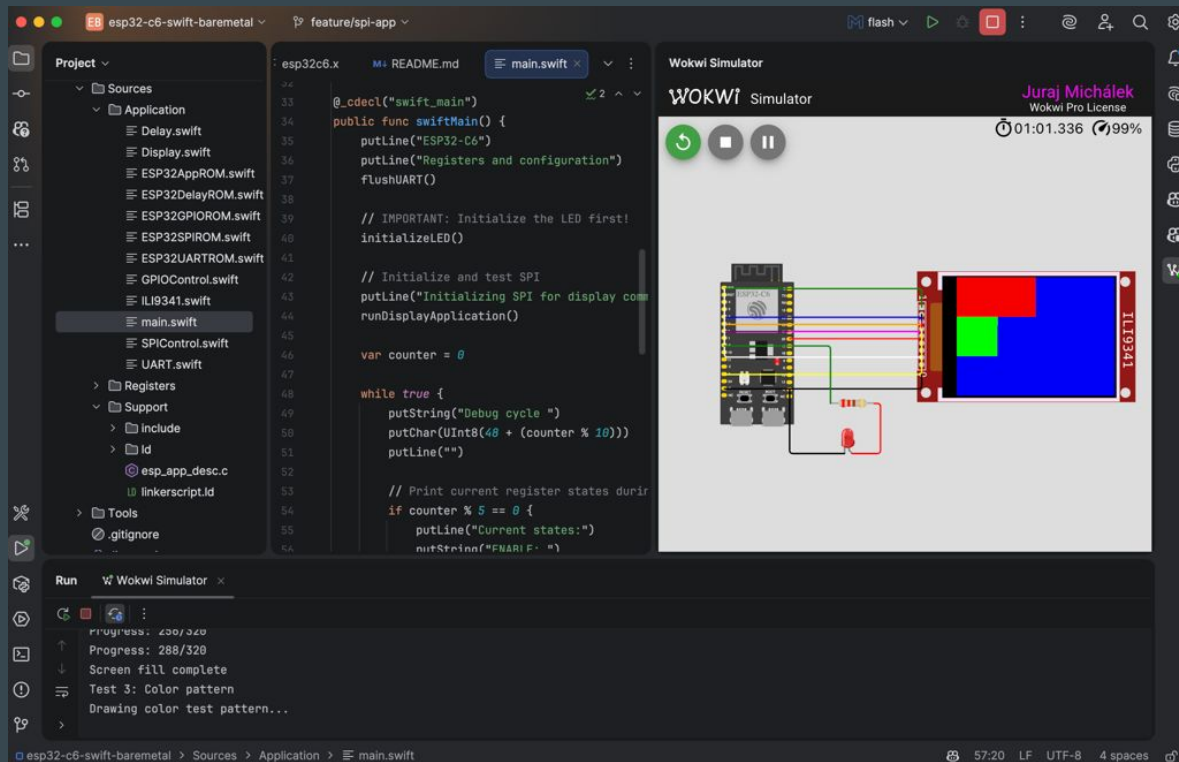
Header required by default boot loader

- https://github.com/georgik/esp32-c6-swift-baremetal/blob/main/Sources/Support/esp_app_desc.c#L21

Working example

Showcase:

- GPIO
- UART
- SPI
- Delay



<https://github.com/georgik/esp32-c6-swift-baremetal>

ESP Product Selector + Product Comparison

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



ESP32-S3

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:



Product Brief

Docs & Certs

DevKits

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 ...

Product Selector | **Product Comparison**

Product Portfolio

Sales Questions

Technical Inquiries

Hide Same | **Show Same**

	ESP32-S3	ESP32-P4NRW16
Series	ESP32-S3	ESP32-P4
CPU	Xtensa® dual-core 32-bit LX7	32-bit RISC-V single-core processor
Freq. (MHz)	240	400
Package (mm)	QFN56 (7*7)	QFN10*10
Dimensions (mm)	7*7	10*10
Temp. (°C)	-40 °C ~ 105 °C	-40 °C ~ 85 °C
Status	Mass Production	Sample
ECO	standard version	
Support IDF	v0.x	

IDE

Supported by Espressif:



- VS Code with Espressif Extension - <https://developer.espressif.com/tags/vscode/>
- Espressif IDE - <https://developer.espressif.com/tags/espressif-ide/>

Supported by JetBrains:

- CLion - <https://developer.espressif.com/blog/clion/>



Tips

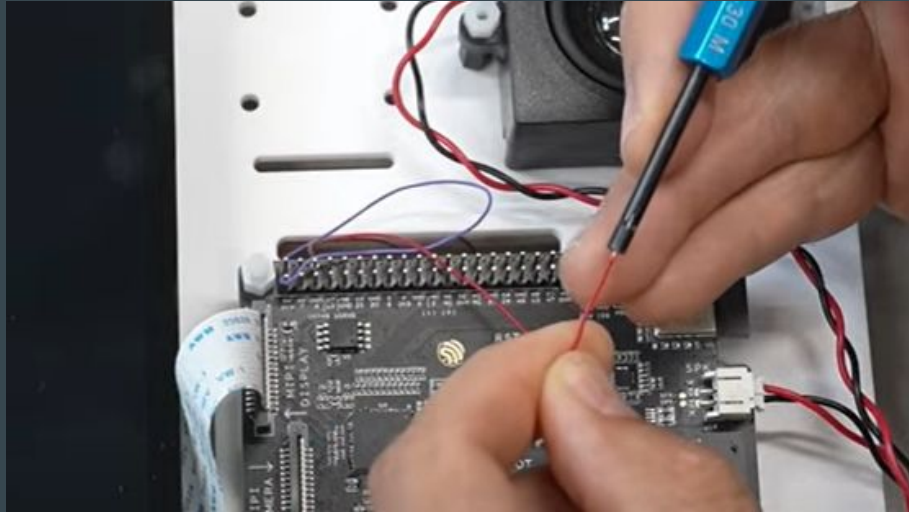
How to “unbrick” ESP32

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

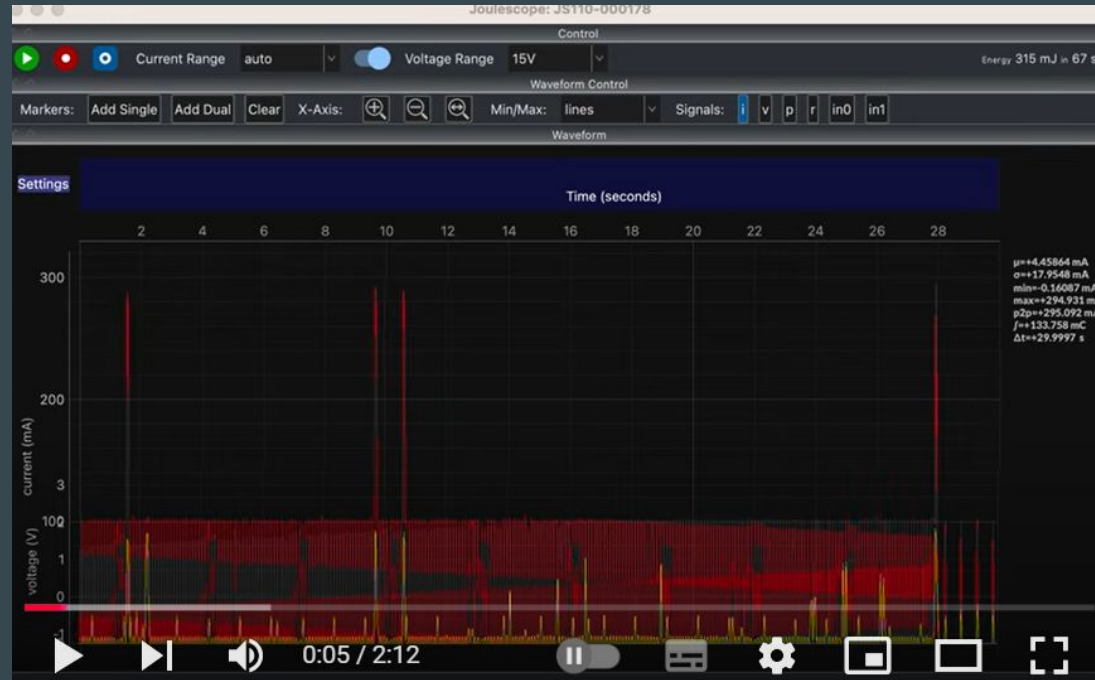
After flashing, just press RESET to return to normal mode.

Wire Wrap Tool



Espressif DevCon 2024 - Tips and Tricks

ESP32-C6 TWT



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

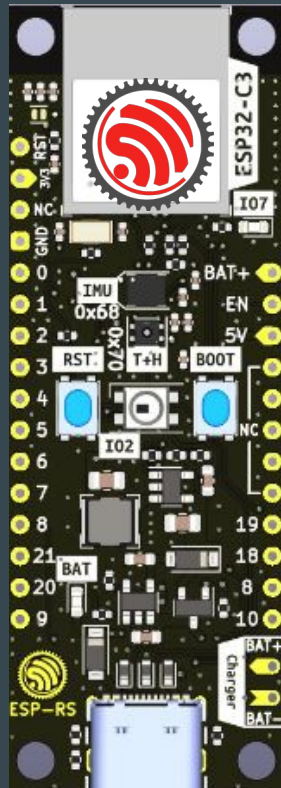
Designing Open Hardware - esp-rust-board

KiCad templates

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

ESP32-C3-DevKit-RUST-1 (available at Mouser, AliExpress)

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

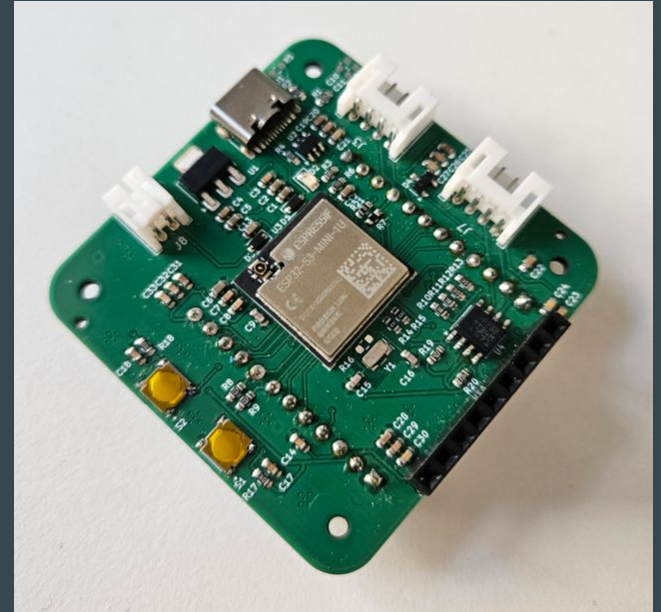


FRI STACK

M5Stack compatible Open Hardware stack

Developed by Technical Cybernetics UNIZA

- https://github.com/andsim04/FRI_STACK_HARDWARE



Espressif in Brno

Vlněna Office Park

Espressif Systems (Czech) s.r.o.

Přízova 3, 602 00 Brno

Czechia, Europe



Espressif Developer Conference 2022-2024 - recording



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

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

Embedded World 2026

Meet us in Nuremberg, Germany



embeddedworld

Exhibition&Conference

Old games ported to ESP32-P4

OpenTyrian - <https://github.com/georgik/OpenTyrian/>

Doom / FreeDoom - <https://github.com/espressif/esp32-doom>

Quake 1 / LibreQuake - <https://github.com/espressif/esp32-quake>

