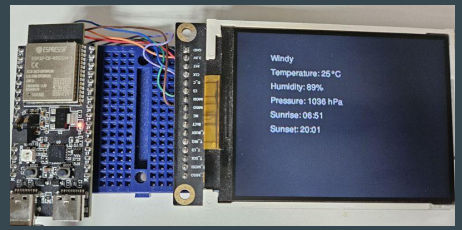
Embedded Swift Comparison of different approaches for ESP32

2025-07-11

Embedded Swift Community Hour

Juraj Michálek
Espressif Systems



Content

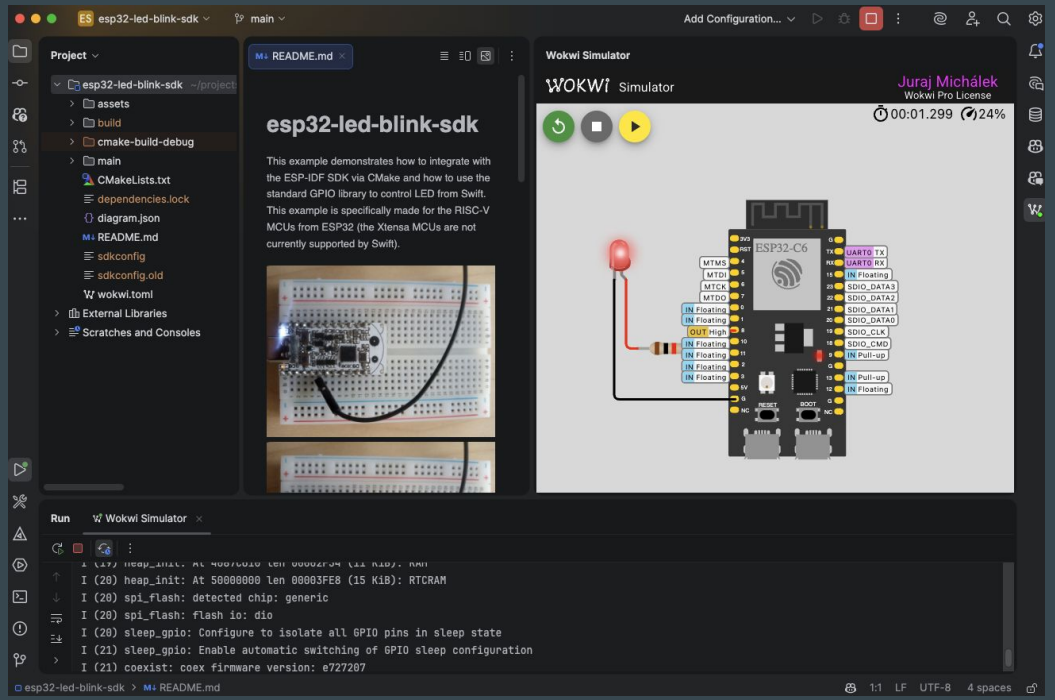
Comparison of approaches:

- Swift + ESP-IDF
- Swift - Bare Metal
- Swift - loaded by OS (ESP-IDF)

Additional resources and links

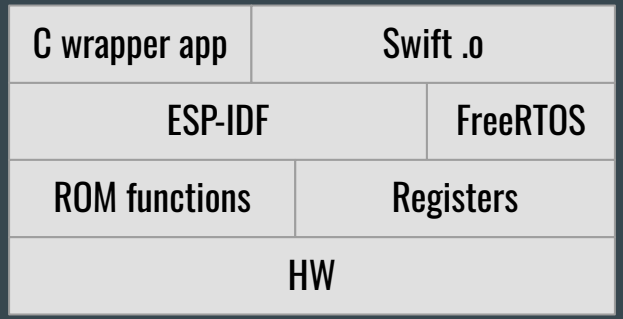
Swift Embedded Examples - esp32-led-blink-sdk

CLion + Wokwi plugin



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

Embedded Swift on top of ESP-IDF (FreeRTOS)

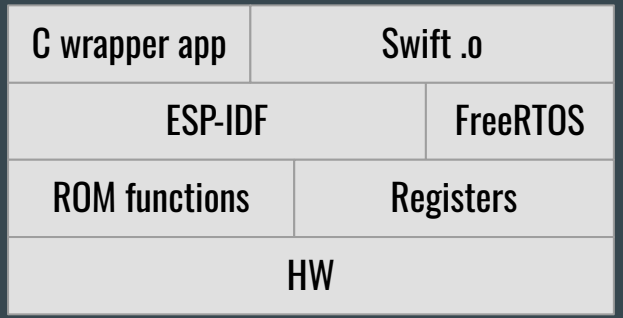


Embedded Swift on top of ESP-IDF (FreeRTOS)

Upside: All functionality of MCU is available including WiFi, well proved and working

Downside: Requires also ESP-IDF toolchain

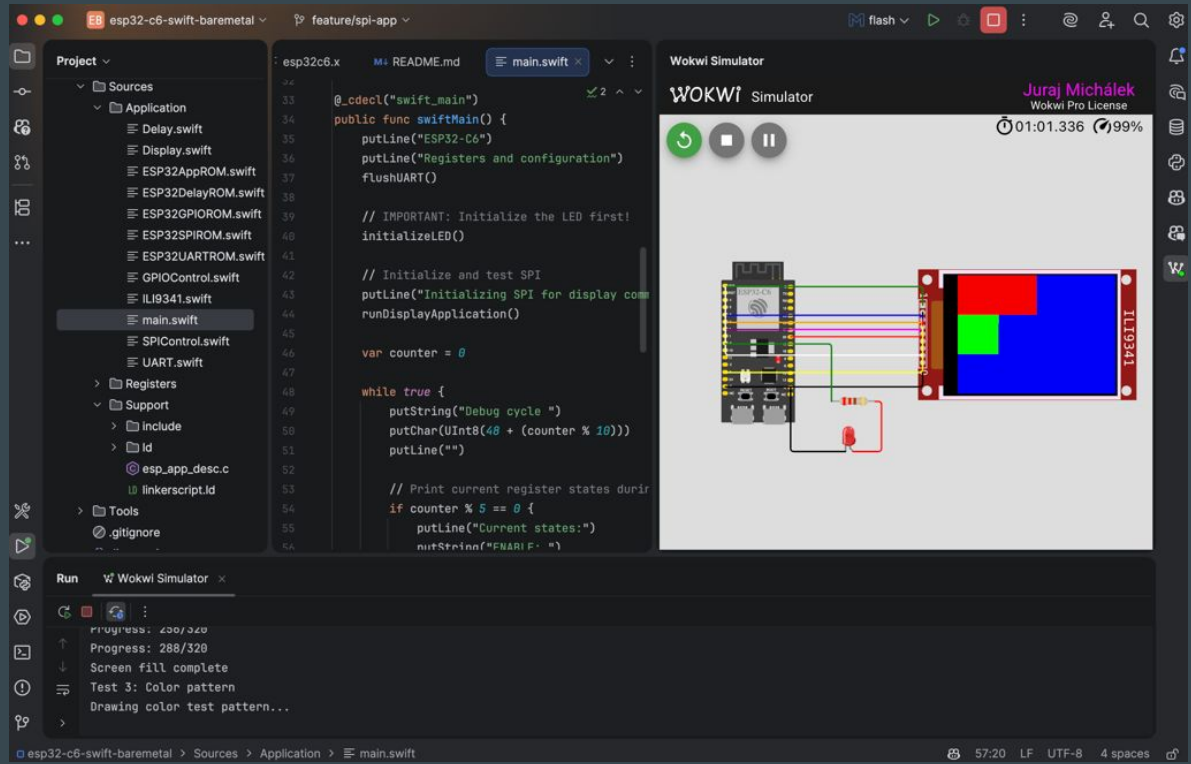
Example: https://github.com/georgik/esp32-sdl3-swift-example



Working example

Showcase:

- GPIO
- UART
- SPI
- Delay



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

Embedded Swift Bare Metal (No OS)

Embedded Swift Bare Metal (No OS)

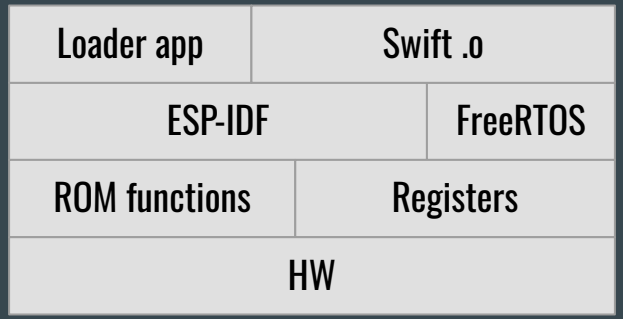
Upside: Total control of the code, no blobs, uses just Swift toolchain

Downside: Requires implementation of everything

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

- working PoC: UART, GPIO, SPI and Delay

Embedded Swift dynamically loaded by ESP-IDF (inception)



Embedded Swift dynamically loaded by ESP-IDF (inception)

Upside: All functions of MCU, uses just Swift toolchain, small flashable Swift binary

Downside: OS binary must be precompiled with everything

Models


C wrapper app
Swift .o
ESP-IDF
FreeRTOS
ROM functions
Registers
HW
Swift
ROM functions
Registers
HW
Loader app
Swift .o
ESP-IDF
FreeRTOS
ROM functions
Registers
HW

What we can learn from Rust?

Rust std (build on top of ESP-IDF) - works well, deployed in production

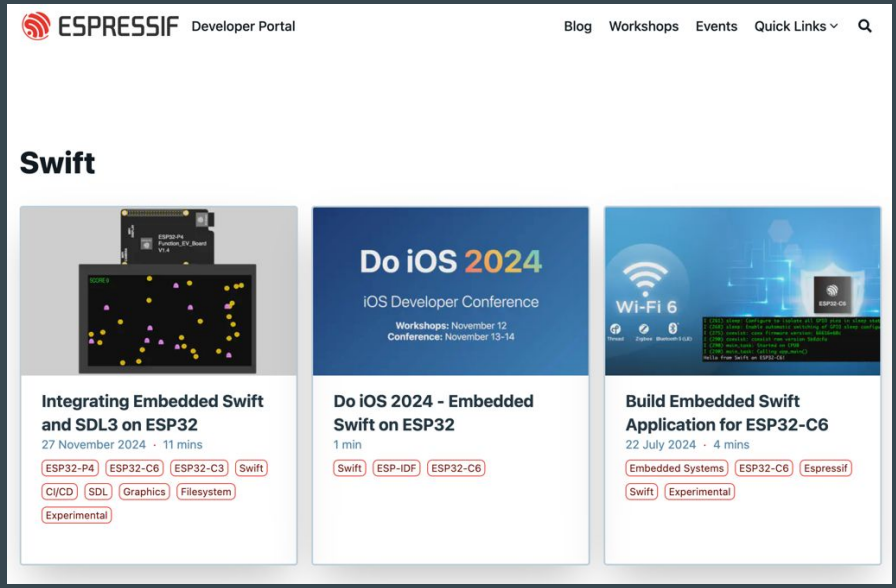
- Swift has advantage of simpler integration with C libraries of ESP-IDF and C components
- very long build (compiling ESP-IDF C code + Rust code)
- only community support

Rust no_std - Bare Metal - options:

- WiFi using blobs
  - works well and deployed in production, officially supported by Espressif
  - short build times, no C code involved
  - requires a lot of integration to “emulate” FreeRTOS for WiFi blobs (metaphor: Wine)
- Pure Rust WiFi driver
  - Ferris on Air - experimentally working on ESP32 and ESP32-S2 (older generation of MCUs)
  - no blobs
  - possibility creating Pure Swift WiFi driver by community - caveat: details about WiFi registers are not available

Resources related to Embedded Swift

Developer Portal



developer.espressif.com/tags/swift

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

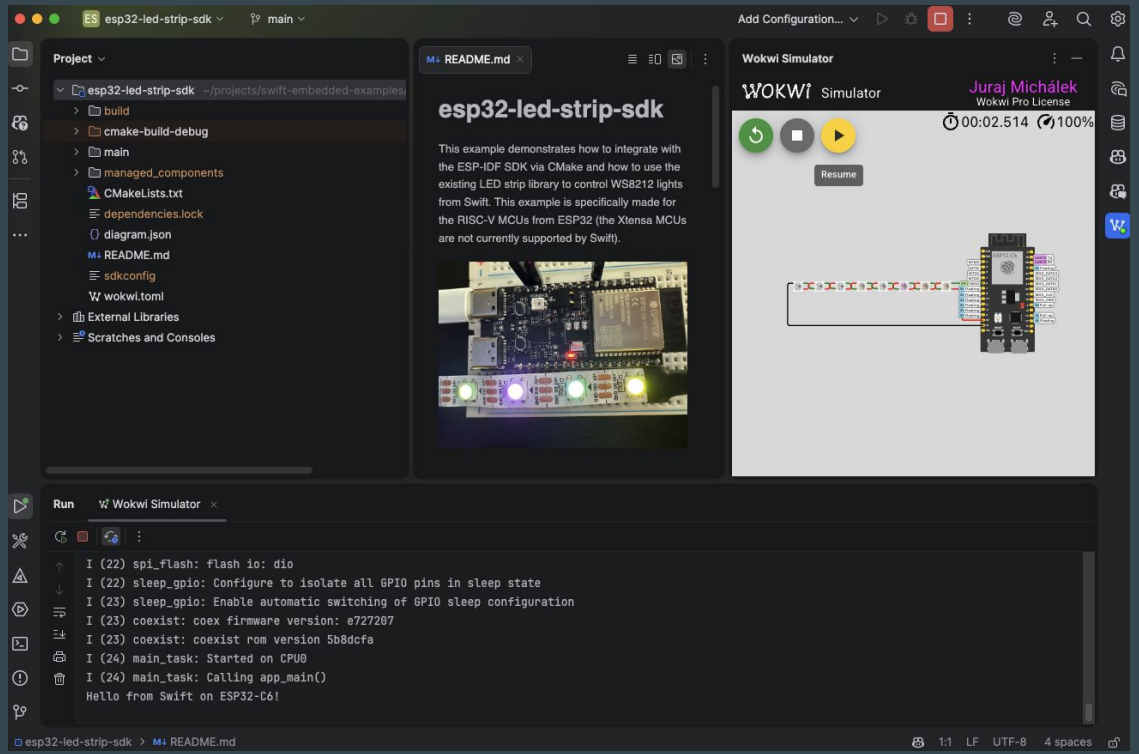
Swift Embedded Examples - esp32-led-strip-sdk

Architecture of ESP-IDF project:

Components

- main
- managed_components
- components (local)

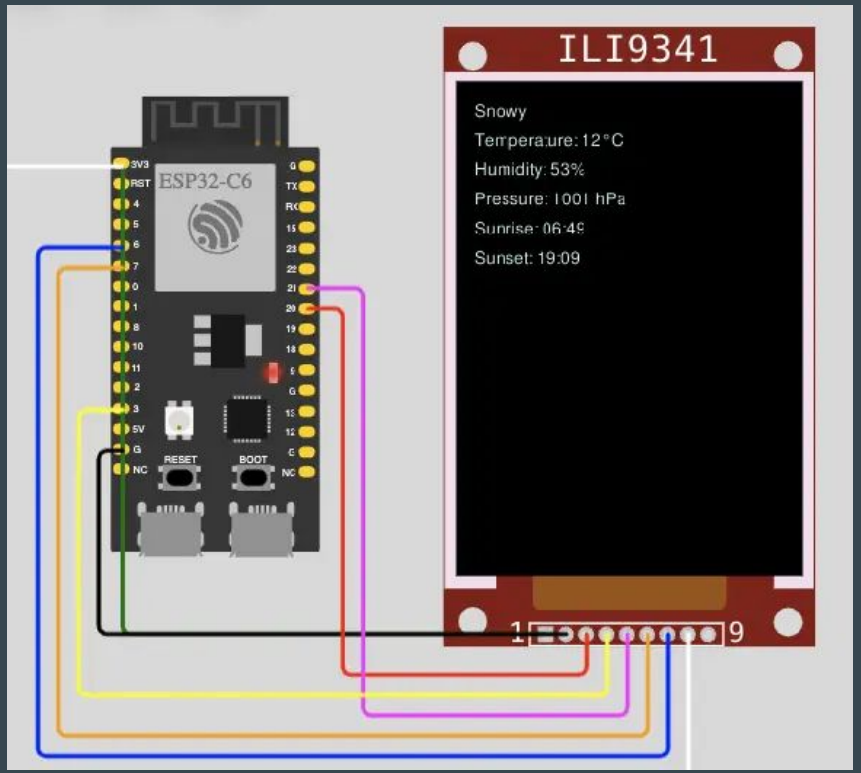
main/idf_component.yml



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

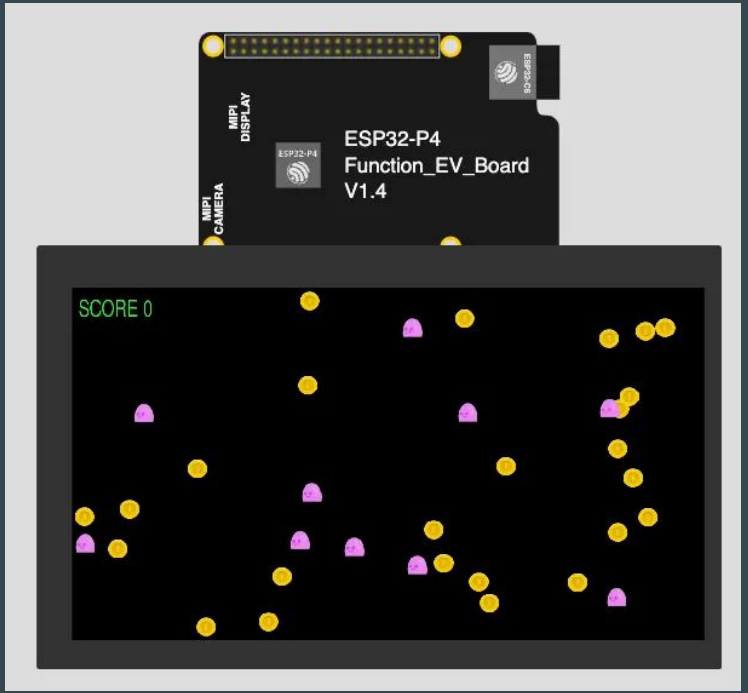
Example with high performance ESP32-P4

RISC-V IMACF

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

Online simulation with Wokwi

ESP-IDF SDL3 Component



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

Espressif in Brno

Vlněna Office Park

Espressif Systems (Czech) s.r.o.

Přízova 3, 602 00 Brno

Czechia, Europe