**1. Tìm hiểu cách viết firmware bằng Rust cho ESP32-C6**

Để viết firmware điều khiển LED RGB cho ESP32-C6 bằng Rust, trước tiên cần hiểu cách Rust hỗ trợ lập trình nhúng (embedded). Rust cho phép lập trình trên vi điều khiển thông qua chế độ no\_std — tức là không sử dụng thư viện tiêu chuẩn std, vốn phụ thuộc vào hệ điều hành. Thay vào đó, Rust cung cấp các thư viện tối giản như core, alloc và một hệ sinh thái phong phú các crate chuyên cho embedded như embedded-hal, esp32c6-hal, panic-halt.

Cụ thể với ESP32-C6, Espressif đã hỗ trợ rất tốt thông qua thư viện esp-hal, trong đó có crate con esp32c6-hal. Thư viện này cung cấp driver điều khiển GPIO, UART, Timer, PWM, SPI, I²C và các peripheral khác. Mã nguồn thường bắt đầu bằng các dòng:

A screen shot of a computer

AI-generated content may be incorrect.

Cấu trúc firmware gồm việc khởi tạo peripheral từ PAC (Peripheral Access Crate), định cấu hình clock, và điều khiển thiết bị ngoại vi như LED. Đây là nền tảng để nhóm có thể điều khiển RGB LED thông qua GPIO hoặc PWM.

**2. Tìm hiểu Toolchain là gì, phân biệt std và no\_std, cách cài toolchain Rust cho ESP32-C6**

**Toolchain** là tập hợp công cụ bao gồm trình biên dịch (rustc), bộ quản lý gói (cargo), target specification (đích biên dịch) và các công cụ liên quan như linker hoặc flasher. Để lập trình Rust cho ESP32-C6, nhóm cần cài đặt công cụ để biên dịch cho nền tảng RISC-V riscv32imc-unknown-none-elf.

Khác biệt giữa std và no\_std là yếu tố quan trọng. Trong môi trường embedded, vi điều khiển không có hệ điều hành và không có bộ nhớ động kiểu heap, nên không thể dùng các thành phần trong std như I/O, thread, hoặc Vec. Thay vào đó, dùng #no\_std] để chỉ sử dụng thư viện core và các chức năng được build riêng cho embedded như thao tác trên GPIO, Timer...

Để cài toolchain Rust cho ESP32-C6, ta thực hiện:

 cargo install cargo-generate ldproxy

rustup toolchain install stable --component rust-src --target riscv32imac-unknown-none-elf

cargo install cargo-espflash espflash

cargo install esp-generate

**3. Tạo thử template code Rust cho ESP32-C6**

Sau khi cài xong toolchain, ta tiến hành tạo project đầu tiên điều khiển LED bằng Rust:

esp-generate --chip=esp32c6 your-project

Sau đó sửa file Cargo.toml để thêm dependency cần thiết. Tiếp đó, ta chỉnh sửa trong file main.rs, viết thử đoạn code nhấp nháy LED.

Dependence:

![A screen shot of a computer program

AI-generated content may be incorrect.

Main.rs:

A computer screen shot of text

AI-generated content may be incorrect.

Cuối cùng, nạp firmware vào board ESP32-C6 bằng:

espflash flash target/riscv32imac-unknown-none-elf/release/hello\_world --chip esp32-c6 --baud 460800 --port COM4

hoặc đơn giản hơn dùng cargo run.

Monitor:

espflash monitor --chip esp32c3 --port COM6 --baud 115200