

# Phần 2: "Hello World!!!"

Giống như lập trình phần mềm. Lập trình nhúng cũng có những bài "Hello world!!!".

Trong phần này ngoài thực hiện log lên màn hình dòng "Hello world!!!" chúng ta sẽ thêm một chút liên quan đến phần cứng là điều khiển đèn led trên ESP32

## Cấu hình Dự án

Để cấu hình chúng ta sử dụng file `platformio.ini`

```
[env:esp32-c3-devkitm-1]
platform = espressif32           ; Khai báo n?n t?ng s? d?ng
board = esp32-c3-devkitm-1       ; Khai báo board, thay ??i v?i nh?ng board khác nhau
framework = arduino              ; Khai báo framework
lib_deps =                       ; Khai báo th? vi?n
  SPI                           ; Vi "hello world" thì ch?a c?n ??n th? vi?n nào
  SD                            ; Có t? xóa b? lib_deps n?u ch?a dùng ??n
monitor_speed = 115200           ; Khai báo t?c ?? giao ti?p Serial Monitor
build_flags =                    ; T?ng t? nh? Scripting Define Symbols trong Unity
  -D ARDUINO_USB_MODE=1         ; ?n 2 báo 2 tr??ng t??ng t? nh? trong ví d?
  -D ARDUINO_USB_CDC_ON_BOOT=1 ; ?? có th? xem ???c log
```

## Các hàm cơ bản

```
// M?t file main.cpp c? b?n khi làm vi?c v?i ESP32

#include <Arduino.h> // Khai báo th? vi?n

// T?ng t? nh? Awake trong Unity.
// Hàm này ???c g?i 1 l?n khi ESP32 kh?i ??ng.
// Dùng ?? kh?i t?o, setup...
void setup()
{
}

// T?ng t? nh? hàm Update trong Unity
// L?p l?i m?i khi các l?nh bên trong loop ???c th?c hi?n h?t
void loop()
{
}
```

## "Hello World!!!"

```
#include <Arduino.h>
```

```

// Khai báo chân giao tiếp với đèn led
const int LED_PIN = 8;

void setup()
{
    // Setup chân led là output
    pinMode(LED_PIN, OUTPUT);

    // Khi nào serial có tốc độ trùng với tốc độ đã khai báo ? platformio.ini
    Serial.begin(115200);
    // Chờ đến khi serial khi nào xong hoặc hết thời gian chờ
    unsigned long start = millis();
    while (!Serial && millis() - start < 2000)
    {
        delay(1);
    }

    // Delay 1s để chờ s? Serial Monitor kịp bắt lên và nhìn thấy log
    // Nếu không có thì code vẫn chạy bình thường c?a s? Serial Monitor bắt lên sau s? không nhìn thấy log
    delay(1000);

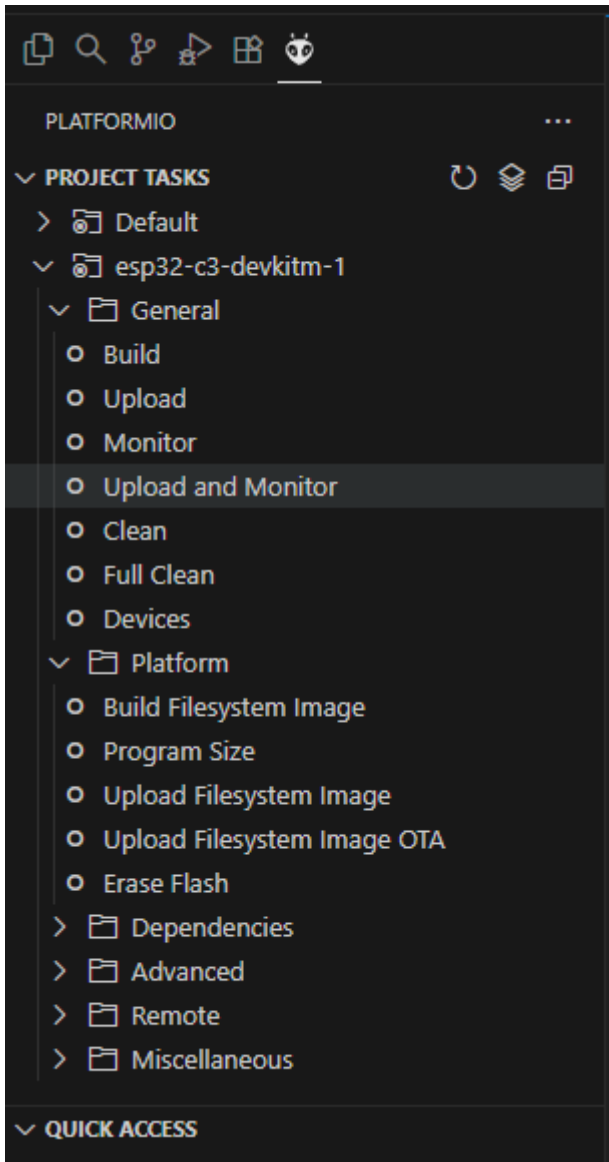
    // Hello world này rồi (=))
    Serial.println("Hello world!!!");
}

void loop()
{
    digitalWrite(LED_PIN, LOW);    // Bật đèn LED
    delay(100);                    // Chờ 0.1 giây
    digitalWrite(LED_PIN, HIGH);   // Tắt đèn LED
    delay(100);                    // Chờ 0.1 giây
}

```

Sau khi code xong thì thực hiện

- Kết nối ESP32 với PC bằng cáp USB
- Upload and Monitor (trong VSCode > tab PlatformIO trên Activity Bar > Upload and Monitor)



Chờ đến khi code được nạp xong. Nhìn cửa sổ terminal và mong chờ kết quả:

- Dòng text "Hello world!!!" được in ra 1 lần
- Trên board ESP32 led xanh nháy liên tục

## Hành trình fix bug đầu tiên

Vấn đề ??

Mặc dù đã nạp code thành công. Đèn nguồn có sáng nhưng tôi không nhận được bất kỳ dòng log nào dù có viết `Serial.println("Hello world!!!");` ở `setup()` hay `loop()`

Quá trình debug

Kiểm tra chip

Để đảm bảo ESP32 này không chết trong lúc mình hàn. Nên mình đã dùng code điều khiển led để kiểm tra xem nó có còn hoạt động bình thường không.

Rất mừng là nó còn sống. Trước đây vừa chết một con xong (T.T)

### => Vấn đề có thể do giao tiếp Serial qua USB

#### Kiểm tra các lỗi cụ thể

Đảm bảo tốc độ `monitor_speed = 115200` là trùng nhau trong `platformio.ini` và `main.cpp`

Thêm delay trước khi khởi tạo `Serial.begin(115200);`

Nhưng vấn đề vẫn không được giải quyết.

### => Có thể vấn đề do driver

#### Kiểm tra các môi trường khác

Chuyển sang macOS và sử dụng CLion với PlatformIO => Vẫn không được!!!

Dùng Arduino IDE để code và nạp code => Code đã chạy và monitor đã hiện log.

### => Vấn đề có thể do cấu hình

#### Mô tả

So sánh 2 cách cấu hình thì trên Arduino IDE có `USB CDC On Boot: "Enabled"`.

Tương tự thêm `build_flags -D ARDUINO_USB_CDC_ON_BOOT=1` vào `platformio.ini` thì sẽ gặp lỗi biên dịch của Serial. Cần phải thêm cả cờ `-D ARDUINO_USB_MODE=1`.

**Lúc này vấn đề đã được giải quyết.**

## Giải thích

Mạch ESP32 - C3 SuperMini đang dùng sử dụng Native USB. Nghĩa là con chip sẽ tự mình xử lý giao tiếp USB. Để tính năng này hoạt động đúng thì trình biên dịch cần thực hiện 2 việc:

- `-D ARDUINO_USB_MODE=1`: Chuyển chế độ hoạt động của cổng USB sang chế độ CDC
- `-D ARDUINO_USB_CDC_ON_BOOT=1`: Kích hoạt chế độ CDC ngay khi khởi động.

---

Revision #3

Created 2025-10-11 10:43:37 UTC by ThanhDV

Updated 2025-10-29 17:00:45 UTC by ThanhDV