









𝐓𝐢̀𝐦 𝐡𝐢𝐞̂̉𝐮 𝐛𝐨𝐨𝐭𝐥𝐨𝐚𝐝𝐞𝐫 𝐭𝐫𝐞̂𝐧 𝐯𝐢 đ𝐢𝐞̂̀𝐮 𝐤𝐡𝐢𝐞̂̉𝐧 - 𝐏𝐡𝐚̂̀𝐧 𝟏

Chào các bạn, trong chuỗi bài viết này chúng ta sẽ tìm hiểu về ứng dụng của bootloader và cách triển khai bootloader trong 1 số tình huống cụ thể

✔𝟏. 𝐏𝐡𝐮̛𝐨̛𝐧𝐠 𝐩𝐡𝐚́𝐩 𝐜𝐚̣̂𝐩 𝐧𝐡𝐚̣̂𝐭 𝐟𝐢𝐫𝐦𝐰𝐚𝐫𝐞 𝐜𝐡𝐨 𝐯𝐢 đ𝐢𝐞̂̀𝐮 𝐤𝐡𝐢𝐞̂̉𝐧

Hiện nay có 3 cách chủ yếu để cập nhật firmware cho vi điều khiển:

- Sử dụng debugger chuyên dụng như St-linh cho stm32, Jlink, Ulink cho các dòng ARM, Pickit cho pic

+ Ưu điểm: support tận răng, nạp và debug thoải mái

+ Nhược điểm: cồng kền, giá thành cao (Jlink, Ulink hàng real cũng 15 củ khoai)

- Sử dụng chức năng boot của vi điều khiển hay bootloader mặc định (Điển hình như esp32)

+ Ưu điểm: nhỏ gọn, nạp qua UART, I2C, SPI tùy phiên bản, chỉ cần 1 mạch USB-UART là có thể update firmware được

+ Nhược điểm: Chức năng hạn chế trong giới hạn của bootloader mặc định, thường chỉ debug bằng log, khó tùy biến

- Sử dụng bootloader hay second bootloader

+ Ưu điểm: nhỏ gọn, tùy biến cao tùy thuộc theo nhu cầu

+ Nhược điểm: thương chỉ debug bằng log

✔𝟐. 𝐕𝐢𝐞̣̂𝐜 𝐠𝐢̀ 𝐬𝐞̃ 𝐱𝐚̉𝐲 𝐫𝐚 𝐤𝐡𝐢 𝐯𝐢 đ𝐢𝐞̂̀𝐮 𝐤𝐡𝐢𝐞̂̉𝐧 𝐤𝐡𝐨̛̉𝐢 đ𝐨̣̂𝐧𝐠

- Khi đặt vi điền khiển khỏi động , bộ đếm chương trình hoặc program counter được tải với địa chỉ

0x00000000 (địa chỉ đầu tiên này chứa địa chỉ của đỉnh ngăn xếp hoặc giá trị sẽ được tải vào con trỏ ngăn xếp chính.)

- Tải giá trị được lưu tại địa chỉ 0x00000000 vào MSP (Main stack pointer)

- PC di chuyển đến địa chỉ tiếp theo 0x00000004 và giá trị tại địa chỉ 0x00000004 chứa một con trỏ để trỏ tới hàm reset handler

- Chạy chức năng của hàm reset handler (Vô hiệu hóa tất cả ngắt, khởi tạo đoạn dữ liệu, khởi tạo phân đoạn .Bss, khởi tạo

phân đoạn ngăn xếp, kích hoạt ngắt,…)

- Gọi hàm main()

(Các bạn có thể phân tích file start-up.s để biết thêm chi tiết)

✔𝟑. 𝐂𝐡𝐮̛́𝐜 𝐧𝐚̆𝐧𝐠 𝐜𝐮̉𝐚 𝟏 𝐛𝐨𝐨𝐭𝐥𝐨𝐚𝐝𝐞𝐫

- Bootloader là một chương trình cho phép chúng ta tải các chương trình khác thông qua giao diện thuận tiện hơn như cáp USB.

Khi bật nguồn hoặc reset bo mạch vi điều khiển, bootloader sẽ kiểm tra xem có yêu cầu cập nhật firmware (ví dụ như nhấn nút) hay không.

Nếu có, nó sẽ tải lên chương trình mới và ghi nó vào bộ nhớ Flash.

Nếu không, nó sẽ bắt đầu chạy chương trình cuối cùng mà bạn đã tải vào trước đó (Chương trình Application).

- Bootloader thương được đặt tại ví trị bắt đầu 0x00000000 (Với stm32 là 0x08000000).

✔𝟒. 𝐇𝐚̀𝐦 𝐧𝐡𝐚̉𝐲 𝐭𝐨̛́𝐢 𝐜𝐡𝐮̛𝐨̛𝐧𝐠 𝐭𝐫𝐢̀𝐧𝐡 𝐜𝐡𝐢́𝐧𝐡

- Một trong những phần quan trọng nhất của bootloader là hàm nhảy tới chương trình chính. Nó có chức năng nhảy tới vị trí chưa chương trình

chính mà bạn muốn thực thi.

- Chương trình chính thường được đặt tại sector (Nhiều vi điều khiển gọi là Block) tiếp theo so với bootloader (Tùy vào kính thước bootloader)

- Nhìn chung, hàm này thường có các thành phần như: deinit tất cả các ngoại vi đang sử dụng, đặt lại main stack pointer và nhảy tới hàm reset handler.

(Xem ảnh để biết được 1 hàm nhảy tới chương trình chính viết cho stm32)

🥂🥂🥂🥂🥂🥂🥂🥂

𝑇𝑟𝑜𝑛𝑔 𝑝ℎ𝑎̂̀𝑛 𝑠𝑎𝑢, 𝑐ℎ𝑢𝑛𝑔 𝑡𝑎 𝑠𝑒̃ 𝑡ℎ𝑢̛̉ 𝑥𝑎̂𝑦 𝑑𝑢̛̣𝑛𝑔 𝑚𝑜̣̂𝑡 𝑐ℎ𝑢̛𝑜̛𝑛𝑔 𝑡𝑟𝑖̀𝑛ℎ 𝑏𝑜𝑜𝑡𝑙𝑜𝑎𝑑𝑒𝑟 𝑐𝑜̛ 𝑏𝑎̉𝑛 ℎ𝑜̂̃ 𝑡𝑟𝑜̛̣ 𝑐𝑎̣̂𝑝 𝑛ℎ𝑎̂́𝑡 𝑓𝑖𝑟𝑚𝑤𝑎𝑟𝑒 𝑞𝑢𝑎 𝑈𝐴𝑅𝑇

𝑃ℎ𝑎̂̀𝑛 2: (𝑂𝑛-𝑔𝑜𝑖𝑛𝑔)

𝑃ℎ𝑎̂̀𝑛 3: (𝑂𝑛-𝑔𝑜𝑖𝑛𝑔)

𝑃ℎ𝑎̂̀𝑛 4: (𝑂𝑛-𝑔𝑜𝑖𝑛𝑔)