

Các khái niệm căn bản lập trình vi điều khiển

Saturday, 30 September 2023 14:45

1. Bản đồ bộ nhớ - Memory Map
 - Linker Script: *.ld - Chỉ ra vùng nhớ ROM, RAM, text, data (bss), heap, stack
 - File định nghĩa các con trỏ tới thanh ghi thiết bị ngoại vi của vi điều khiển - stm32f103c8t6.h
=> Cần sử dụng các từ khóa: define, volatile
2. Bản đồ thanh ghi của 1 thiết bị ngoại vi - gpio.h
 - Có thể dùng struct để định nghĩa bản đồ thanh ghi cho một thiết bị ngoại vi

```
struct gpio_reg {  
    int32_t gpio_in;  
    int32_t gpio_dir;  
    ...  
}
```

3. Trường bit của thanh ghi
 - a. Để thao tác thì dùng các phép toán logic với mặt nạ bit
 - b. Dùng union và struct với bit field để định nghĩa trường bit của thanh ghi

```
union un_gpio_dir {  
    int32_t gpio_dir;  
    struct gpio_dir_bit_field {  
        int32_t dir      :1;  
        int32_t en       :1;  
        int32_t pull_up  :1;  
    }  
}  
  
struct gpio_reg {  
    union un_gpio_dir gpio_in;  
    int32_t gpio_dir;  
    ...  
}  
  
gpioA_regs = ((volatile struct gpio_reg*) GPIO_A_BASE);  
gpioA_regs->gpio_in->gpio_dir_bit_field.dir = GPIO_IN_DIR;
```

4. Startup code
 - a. ROM Boot ở địa chỉ PC đầu tiên 0x0: khởi động CLK, RAM, ROM, FLASH Interface
 - b. Bootloader (file startup.s, crt0.s): khởi tạo và chứa các hàm ngắt, lệnh khởi tạo các hằng số và biến số toàn cục, setup vector ngắt, khởi tạo một số ngoại vi (JTAG, UART, GPIO, PMIC)
 - c. boot.img
 - d. kernel.img
 - e. file.img
 - f. app (main.c)
5. Interrupt & Interrupt vector

Khi có ngắt bộ xử lý sẽ

 - 1) Lưu context: lưu PC, lưu các thanh ghi
 - 2) Đặt PC = int_vec[int_num]
 - 3) Nhảy đến để thực hiện lệnh ở vị trí PC
 - 4) Kết thúc hàm ngắt
 - 5) Khôi phục context