

1. Giới thiệu

Interrupts, thường được gọi là ngắt, là một “tín hiệu khẩn cấp” gửi đến bộ xử lý, yêu cầu bộ xử lý tạm ngừng tức khắc các hoạt động hiện tại để “nhảy” đến một nơi khác thực hiện một nhiệm vụ “khẩn cấp” nào đó, nhiệm vụ này gọi là trình phục vụ ngắt – ISR (Interrupt Service Routine). Sau khi kết thúc nhiệm vụ trong ISR, bộ đếm chương trình sẽ được trả về giá trị trước đó để bộ xử lý quay về thực hiện tiếp các nhiệm vụ đang được thực hiện khi trước khi ngắt xảy ra.

Như vậy, ngắt có mức độ ưu tiên xử lý cao nhất, ngắt thường được dùng để xử lý các sự kiện bất ngờ nhưng không tốn quá nhiều thời gian. Các tín hiệu dẫn đến ngắt có thể xuất phát từ các thiết bị bên trong chip (ngắt báo bộ đếm timer/counter tràn, ngắt báo quá trình gửi dữ liệu bằng RS232 kết thúc...) hay do các tác nhân bên ngoài (ngắt báo có 1 button được nhấn, ngắt báo có 1 gói dữ liệu đã được nhận...)

Trong bài thí nghiệm ta sẽ tập trung vào việc sử dụng module ngắt I/O trên bo mạch thực hành Kit PICDEM.

Để có thể tiến hành việc thí nghiệm, sinh viên cần trang bị các thiết bị sau:

- Bo mạch Kit PICDEM Mechatronics
- Mạch nạp Pickit
- Phần mềm viết code cho vi điều khiển (MPLab + Cxx Compiler, Hi-TechC, CCS-C hoặc MicroC)

Mục tiêu:

- Đọc tín hiệu từ bên ngoài vào vi điều khiển thông qua xử lý ngắt (PortB).
- Hiện thực chương trình thiết lập PIC ở chế độ SLEEP và WAKEUP PIC sử dụng ngắt.

2. Thời gian thực hiện:

Thời lượng: 3 tiết cho mỗi nhóm sinh viên

3. Nội dung thí nghiệm

3.1. Bài thí nghiệm 1: Lập trình với ngắt PORTB

Đề bài: *Hãy viết một đoạn chương trình cho vi điều khiển thực hiện nhiệm vụ: nhấn nút SW3 thì PIC ở chế độ SLEEP, nhấn nút SW2 thì PIC ở chế độ WAKEUP và làm thay đổi trạng thái đèn led D0 (sáng thành tắt, tắt thành sáng)*

Thực hiện kết nối các chân như sau:

- Kết nối chân SW2 => RB5
- Kết nối chân SW3 => RB4
- Kết nối chân RD7 => R36

Chương trình:

```

#include <16F917.h>
#define adc=8
#define delay(internal=8M)
// #include <SEGMENT_LCD.c>
#define bitRB0 = 0x06.0
#define bitRB4 = 0x06.4
#define bitRB5 = 0x06.5
#define bitRD7 = 0x08.7
#define bitRD6 = 0x08.6
short sleep_mode;

//-----
//----- interrupts -----
#define INT_RB
void rb_isr() {
    if(bitRB4 == 0){
        sleep_mode=TRUE;
    }
    if(bitRB5 == 0){
        sleep_mode=FALSE;
    }
    delay_ms(100);
}
//===== end
=====

void main()
{
    SET_TRIS_B( 0b11111111 ); //set RB0 as input, RB1 to RB7 as input
    SET_TRIS_D( 0b00000000 ); //set RD0 to RD7 as output
    sleep_mode=TRUE;          // init sleep flag
    //ext_int_edge(H_TO_L);    // init interrupt triggering for button press
    port_b_pullups (0);
    enable_interrupts(INT_RB4); // turn on interrupts
    enable_interrupts(INT_RB5); // turn on interrupts
    enable_interrupts(GLOBAL);
    while (1)
    {
        if(sleep_mode)        // if sleep flag set
            sleep();           // make processor sleep
        delay_ms(100);
        bitRD7 ^= 1;
    }
}

```

} }

Yêu cầu:

Sinh viên viết lại các đoạn code trên và nạp vào vi điều khiển trả lời các câu hỏi trong phần báo cáo kết quả thí nghiệm

3.2. Thực hiện chương trình sau:

Sinh viên thực hiện kết nối 3 nút nhấn SW2, SW3, SW4 đến 3 chân input RB4-RB6 của vi điều khiển, kết nối 3 chân Output đến các đèn D0,D1,D2. Viết chương trình các ngắt PORTB để nhận tín hiệu cho vi điều khiển:

- Nhấn SW2: D0 sáng. Các đèn khác tắt.
- Nhấn SW3: D1 sáng. Các đèn khác tắt.
- Nhấn SW4: D2 sáng. Các đèn khác tắt.

3.3. Thực hiện chương trình sau:

Sinh viên thực hiện kết nối 3 nút nhấn SW2, SW3, SW4 đến 3 chân input RB4-RB6 của vi điều khiển, kết nối 3 chân Output đến các đèn D0,D1,D2. Viết chương trình thiết lập chế độ luôn luôn SLEEP cho PIC và sẽ WAKEUP PIC mỗi khi có một trong các nút nhấn SW2, SW3, SW4 được nhấn và:

- Nhấn SW2 : D0 sáng, chờ 3s, thiết lập chế độ SLEEP, tắt D0, Bật D1, D2
- Nhấn SW3 : D1 sáng, chờ 3s, thiết lập chế độ SLEEP, tắt D1, Bật D0, D2
- Nhấn SW4 : D2 sáng, chờ 3s, thiết lập chế độ SLEEP, tắt D2, Bật D0, D1

4. Kết quả thực hành, thí nghiệm (*Lưu ý: Sinh viên nộp lại tờ này cho Giảng viên hướng dẫn sau buổi thí nghiệm*)

Bài thí nghiệm 3.1:**Câu hỏi**

1. Có bao nhiêu interrupt PORTB với PIC16F917?

Giải thích:

2. Câu lệnh **port_b_pullups (0);** có ý nghĩa gì? Nếu thiết lập câu lệnh **port_b_pullups (1);** trong chương trình trên có được không?

Giải thích:

3. Những sự kiện nào có thể WAKEUP PIC ở chế độ SLEEP?

Giải thích:

4. Tại sao interrupt PORTB, interrupt ngoài RB0 không cần thiết lập bit PEIE?

Khi đèn Led D0 được bật, sau đó thiết lập chế độ SLEEP cho Pic. Hỏi sau khi SLEEP đèn D0 vẫn được bật hay tắt?

Giải thích:

5. Trong chương trình trên, hãy tìm các đoạn code thừa mà khi bỏ ra không ảnh hưởng đến hoạt động của mạch? Giải thích vì sao?

Giải thích:

Bài thí nghiệm 3.2:

Không chạy ☐

Chạy không hoàn chỉnh ☐

Chạy tốt ☐

Ý kiến khác:

Bài thí nghiệm 3.3:

Không chạy ☐

Chạy không hoàn chỉnh ☐

Chạy tốt ☐

Ý kiến khác:

Họ và tên sinh viên:.....MSSV:..... Nhóm:.....

Ngày thực hành / thí nghiệm:.....Ký tên:

5. Tài liệu tham khảo

[1] PICDEM™ Mechatronics Demonstration Board User's Guide.