

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:

- Chuyển đổi ADC, comparator thông qua xử lý ngắt (ADC, Comparator interrupts) và phân biệt được ưu điểm của việc sử dụng ngắt để chuyển đổi ADC và Comparator so với sử dụng delay và pooling.
- Nhận tín hiệu analog input và hiển thị kết quả trên LCD.
- Sử dụng ngắt comparator với COMPARATOR VOLTAGE REFERENCE.

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: Chương trình ngắt ADC.

Đề bài: *Hãy viết một đoạn chương trình sử dụng ngắt ADC cho vi điều khiển thực hiện nhiệm vụ: Xoay biến trở POT1 thì sẽ hiển thị kết quả ra LCD*

Thực hiện kết nối các chân như sau:

- Kết nối chân POT1 => RA0

Chương trình 3.1:

```
//File "main.h"
#include <16F917.h>
#define adc=8

//#use delay(clock=32768)
//#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

```
// Chương trình chính
#include "C:\Users\An\Desktop\tn vdk\code tn\read_adc\main.h"
#define delay(internal=8M)
#include "SEGMENT_LCD.c"
long adc_data;
#define INT_AD
void ADC_ISR()
{
    display_digit3(adc_data/100);
    display_digit2((adc_data%100)/10);
    display_digit1(adc_data%10);
}
void main()
{
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(sAN0);
    set_adc_channel(0);
    enable_interrupts(INT_AD); // turn on interrupts
    enable_interrupts(GLOBAL);
    delay_us(10);
    lcd_init();
    while (1){
        adc_data=read_adc();
        delay_ms(50);
    }
}
```

3.2. Bài thí nghiệm 2: Chương trình ngắt Comparator.

Đề bài: Hãy viết một đoạn chương trình sử dụng ngắt Comparator cho vi điều khiển thực hiện nhiệm vụ: So sánh điện áp ra của biến trở POT1 với mức điện áp 3.6V nếu nhỏ hơn 3.6 V thì Led D0 sáng, ngược lại Led D0 sẽ tắt.

Thực hiện kết nối các chân như sau:

- Kết nối chân POT1 => RA0
- Kết nối chân RD7 => R36 – led D0

```
#include <16F917.h>
#device adc=8

#use delay(internal=8M)
#bit bitRD7 = 0x08.7
#byte CMCON = getenv("SFR:CMCON0")
int8 temp;
#INT_COMP
void comp_isr() {
    int8 temp;
    if(C1OUT == 1)
        bitRD7 = 1;
    else
        bitRD7 = 0;
    temp = CMCON;
}
VOID main()
{
    SET_TRIS_D(0b00000000); //set RD0 to RD7 as output
    bitRD7 = 0;
    setup_adc_ports(sAN0|sAN1);
    enable_interrupts(INT_COMP); // turn on interrupts
    enable_interrupts(GLOBAL);
    setup_comparator(A0_VR_A1_VR);
    setup_vref(VREF_HIGH|15);
    while(TRUE);
}
```

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

3.3. Thực hiện chương trình sau:

Hãy viết một đoạn chương trình chuyển đổi ADC sử dụng hàm delay(). Xoay biến trở POT1 thì điện áp ra của POT1 sẽ hiển thị kết quả ra LCD. Kiểm tra kết quả thực tế và kết quả xuất ra LCD.

3.4. Thực hiện chương trình sau:

4. *Hãy viết một đoạn chương trình sử dụng ngắt Comparator cho vi điều khiển thực hiện nhiệm vụ: So sánh điện áp ra của biến trở POT1 với mức điện áp 3,6V và 3.4V nếu nhỏ hơn 3.4 V thì Led D0 sáng, ngược lại nếu lớn hơn 3.6V Led D0 sẽ tắt.*

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 1:

Câu hỏi

1. Có bao nhiêu kênh chuyển đổi ADC với PIC16F917?

Giải thích:

2. Câu lệnh `#byte CMCON = getenv("SFR:CMCON0")` để làm gì? Nếu bỏ câu khai báo `#byte CMCON = getenv("SFR:CMCON0")` trong chương trình trên có được không?

Giải thích:

3. Ưu điểm của việc sử dụng ngắt khi chuyển đổi ADC?

Giải thích:

4. Cho biết câu lệnh sau: `setup_vref(VREF_HIGH|15);` tương ứng với mức điện áp nào của CVREF.

Giải thích:

5. Trong 2 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 3.3:

Không chạy ☐

Chạy không hoàn chỉnh ☐

Chạy tốt ☐

Ý kiến khác:

Bài thí nghiệm 3.4:

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.