

BÀI 5. NGẮT TIMER

Chuẩn đầu ra của học phần:

L1: Giải thích cấu tạo, hoạt động của vi điều khiển, trình bày một số tài nguyên của PIC và lập trình được cho vi điều khiển

Mục tiêu bài học:

Bài học cung cấp kiến thức về các ngắt timer trên PIC18F4520, cách lập trình sử dụng ngắt timer.

Nội dung bài học trực tuyến:

- (1) Khái quát về ngắt timer
- (2) Các bit điều khiển ngắt timer
- (3) Lập trình mô phỏng sử dụng ngắt Timer.

5.1. Vai trò quan trọng của ngắt timer trong các hệ thống đo lường, điều khiển

Trong các bài toán thực tế, một hệ thống nhúng sử dụng vi điều khiển thường phải thực hiện hàng chục đến hàng trăm tác vụ khác nhau. Trong đó có những tác vụ được thực hiện tuần tự theo tiến trình thời gian, khoảng thời gian để vi điều khiển thực hiện mỗi tác vụ cũng thường khác nhau tùy thuộc vào yêu cầu của từng tác vụ. Ngoài ra, cũng có những tác vụ không thực hiện tuần tự như các tác vụ khác, điển hình là đáp ứng của vi điều khiển mỗi khi một sự kiện quan tâm nào đó xảy ra (ví dụ như hoạt động ngắt ngoài của vi điều khiển).

Tuy nhiên, còn có những tác vụ khác mà cần được định kỳ thực hiện, ví dụ như hoạt động định kỳ kiểm tra thông tin của cảm biến nhiệt độ, cảm biến độ ẩm, cảm biến khí gas, nồng độ bụi... của một thiết bị giám sát môi trường. Để vi điều khiển thực hiện được yêu cầu này mà vẫn đảm bảo thực hiện được các tác vụ khác, như các tác vụ thực hiện tuần tự, đáp ứng với ngắt ngoài, thì sử dụng ngắt timer là lựa chọn hàng đầu.

5.2. Các bit điều khiển ngắt timer

Tương tự như ngắt ngoài, ngoài các bit điều khiển ngắt toàn cục, các ngắt timer có 3 bit điều khiển riêng cho từng timer:

- TMRxIE: bit cho phép ngắt timer
- TMRxIP: bit lựa chọn ưu tiên cho ngắt timer
- TMRxIF: bit cờ báo ngắt timer (cần được xóa bằng phần mềm)

Trong đó 'x' là chỉ số của 0,1,2,3 của 4 bộ timer.

Bảng tổng hợp các bit điều khiển ngắt timer

Timer	Bit cho phép ngắt	Bit ưu tiên ngắt	Cờ ngắt
0	TMR0IE INTCON<5>	TMROIP INTCON2<2>	TMR0IF INTCON<2>

1	TMR1IE PIE1<0>	TMR1IP IPR1<0>	TMR1IF PIR1<0>
2	TMR2IE PIE1<1>	TMR2IP IPR1<1>	TMR2IF PIR1<1>
3	TMR3IE PIE2<1>	TMR3IP IPR2<1>	TMR3IF PIR2<1>

Ngắt timer xảy ra khi nào (TMRxIF được set bằng 1 khi nào)?

Timer0: ngắt timer0 được tạo ra khi xảy ra hiện tượng tràn của thanh ghi chứa số đếm từ 255->0 (chế độ 8 bit) hoặc 65536->0 (chế độ 16 bit). Khi đó, cờ báo ngắt (cờ báo tràn) TMR0IF được tự động set lên mức '1'. Bit TMR0IF cần được xóa bằng phần mềm trong chương trình con phục vụ ngắt.

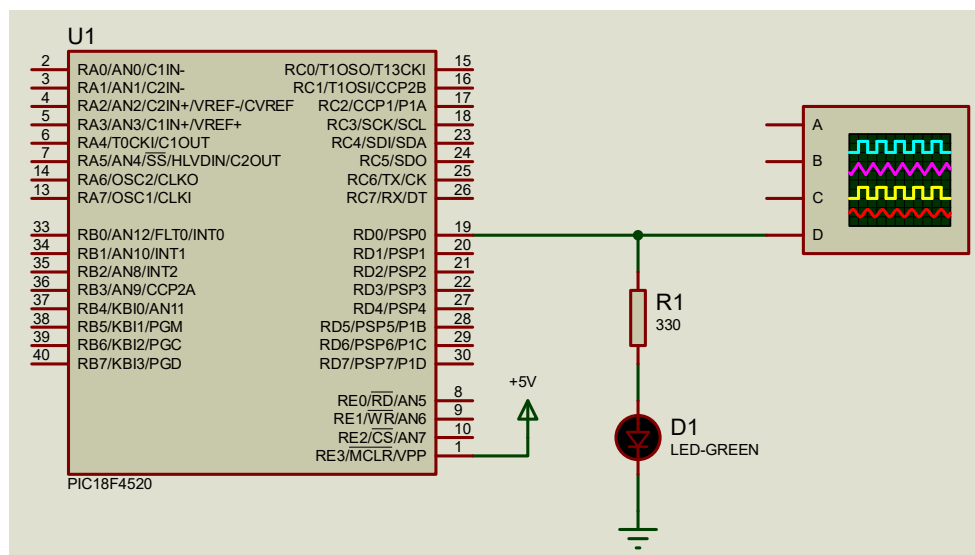
Timer1: ngắt timer1 được tạo ra khi xảy ra hiện tượng tràn của thanh ghi chứa số đếm từ 65536->0. Khi đó, cờ báo ngắt (cờ báo tràn) TMR1IF được tự động set lên mức '1'. Bit TMR1IF cần được xóa bằng phần mềm trong chương trình con phục vụ ngắt.

Timer2: Khi timer2 hoạt động, giả sử số lần TMR2 bằng PR2 là X, ngắt timer2 được tạo ra khi X bằng giá trị Postscaler (xác định bởi 4 bit T2OUTPS3:T2OUTPS0). Khi đó, cờ báo ngắt TMR2IF được tự động set lên mức '1'. Bit TMR2IF cần được xóa bằng phần mềm trong chương trình con phục vụ ngắt.

Timer3: tương tự timer1.

5.3. Lập trình tạo khoảng thời gian xác định sử dụng ngắt Timer0

Cho mạch điện như hình vẽ, lập trình điều khiển để LED D1 sáng nhấp nháy với chu kỳ 1 giây (giả sử vi điều khiển hoạt động ở tần số Fosc=4MHz) sử dụng ngắt Timer0



Bài làm:

Phân tích yêu cầu bài toán và tính chọn tham số

➤ B1: Viết khung chương trình có sử dụng ngắt

➤ B2: Viết các lệnh của chương trình chính:

- ✓ Khởi tạo các PORT: RD0 PORTD chiều ra
- ✓ Khởi tạo thanh ghi T0CON: chế độ 16 bit, nguồn xung clock nội, hệ số chia tần 32 (sinh viên tự thực hiện).
- ✓ Set các bit cho phép ngắt toàn cục: do ta sẽ lựa chọn mức ưu tiên cao cho ngắt timer trong bài toán này, ta cần set giá trị các bit cho phép ngắt toàn cục như sau:

GIE/GIEH = 1.

PEIE/GIEL = 0 hoặc 1.

- ✓ Lựa chọn ưu tiên ngắt: Cho phép ưu tiên ngắt RCONbits.IPEN = 1; lựa chọn ngắt Timer0 ưu tiên cao

INTCON2bits.TMR0IP=1.

- ✓ Cho phép ngắt Timer0:

INTCONbits.TMR0IE = 1.

- ✓ B3: Tính toán giá trị nạp lại cho 2 thanh ghi TMR0H và TMR0L (sinh viên tự thực hiện): số xung cần đếm của timer tới khi tràn là 15625.

➤ B3: Viết các lệnh của chương trình con phục vụ ngắt: trình tự được sử dụng thông dụng như sau

- ✓ Dừng Timer: TMR0ON=0;
- ✓ Nạp lại giá trị cho thanh ghi chứa số đếm
- ✓ Xóa cờ ngắt: TMR0IF=0;
- ✓ Bật Timer: TMR0ON=1;
- ✓ Các lệnh theo đề bài (ví dụ: xung==~xung;)

Code chương trình đầy đủ:

```
#include<p18f4520.h>
```

```
#pragma config OSC = HS // xung hệ thống ở chế độ HS
```

```
#pragma config WDT = OFF // tắt chế độ watchdog
```

```
#pragma config MCLRE = ON // cho phép Reset trên chân MCLR
```

```
#define xung PORTDbits.RD0
```

```

// khai bao ten chuong trinh con ngat
void ngat_ngoai(void);

// ngat uu tien cao
#pragma      code uu_tien_cao = 0x08
void ngat_cao(void)
{
    ngat_ngoai();
}

#pragma code
#pragma      interrupt ngat_ngoai
// chuong trinh con phuc vu ngat
void ngat_ngoai(void)
{
    T0CONbits.TMR0ON = 0;    // dung Timer
    TMR0H = (65536-15625)/256;
    TMR0L = (65536-15625)%256;
    xung=~xung;
    INTCONbits.TMR0IF = 0;    // xoa co tran
    T0CONbits.TMR0ON = 1;
}

void main()
{
    T0CON = 0b00000100; // Timer0 16 bit, chia truoc 32
    ADCON1=0x0f; // chon tat cac cac chan I/O la vao ra so
    TRISD = 0x00;    // thiet lap dau ra cho PORTD
    // Cấu hình các bit điều khiển ngắt
    RCONbits.IPEN = 1; // bit cho phép ưu tiên ngắt
    INTCON2bits.TMR0IP = 1;    // Timer0 ưu tiên cao
    INTCON = 0b11100000; // GIE/GIEH = 1, PEIE/GIEL = 1, TMR0IE = 1,
    // Ghi giá trị khởi tạo cho thanh ghi chứa số đếm của timer0
    TMR0H = (65536-15625)/256; // chia 256 lấy phần nguyên
    TMR0L = (65536-15625)%256; // chia 256 lấy phần dư
    T0CONbits.TMR0ON = 1;    // bat Timer0
    while(1)

```

```
{  
    // Các tác vụ thực hiện trong chương trình chính  
}  
}
```

- ***Yêu cầu: Sinh viên vẽ mạch mô phỏng trên Proteus, tạo project, viết code theo ví dụ và chạy mô phỏng để kiểm chứng kết quả.***