



Forum: [8-bit PIC](#)

Topic: PIC16F1503 timer 計數中斷 與 外部中斷無法同時使用

Subject: PIC16F1503 timer 計數中斷 與 外部中斷無法同時使用

作者: ken.wu

2020年05月21日 16:18:15

請教前輩，我原本有使用timer1設定每1ms中斷一次當做時間計算給LED1使用，後來我要增加另一個LED2(PWM控制點燈)，所以打算用外部中斷弁端A可以用彈跳開關做為兩各LED的切換，但加入中斷後，按彈跳開關都沒有反應，我是設定負緣觸發的時候產生中斷，示波器量測確實有負緣信號產生，但卻沒有進行切換，請問是我的設定哪邊有問題嗎？

以下是我的程式code，請不吝指教，

```
=====
#include<pic.h>

__CONFIG(FOSC_INTOSC&WDTE_OFF&PWRTE_ON&MCLRE_OFF&CP_OFF&BOREN_OFF);

#define LED1 RC2
#define LED2 RC1 //PWM

volatile unsigned int ms=0;
volatile unsigned char step=0;
volatile unsigned char SW=0;

void system_initial(void)
{
    OSCCONbits.IRCF = 0b1110; //內部振盪器頻率設為8MHz
    OSCCONbits.SCS = 0b11; //內部振盪器模塊
    INTCONbits.GIE = 1; //允釧嶺釧陵臚刈_
    INTCONbits.PEIE = 1; //允釧嶺釧陵艦~設中斷
    INTCONbits.INTE = 1; //允蛆NT外部中斷
    TRISA = 0b00000100; //RA2 input
    TRISC = 0b00000000;
    ANSELA = 0b00000000; //RA0~RA4 Digital I/O
    ANSELC = 0b00000000; //RC0~RC3 Digital I/O
    LED1 = 0;
    LED2 = 0;
}

void PWM_Init(void)
{
    PWM4CON = 0b00000000; //Clear PWM2CON
    PR2 = 0b10100110; //Configure the Timer2 period - decimal 166
    PWM4CON = 0b11000000; //Enable PWM Module, Module Output
    PWM4DCH = 0b00000000; //Clear duty cycle registers
    PWM4DCL = 0b00000000;
    TMR2IF = 0; //Clear the timer 2 interrupt flag
}
```

```

    T2CON  = 0b00000001;    //Set prescaler to 4
    TMR2ON  = 1;            //Enable timer 2
    PWM4DCH = 0b00010000;   //Pulse Width:33.5µs, Duty = 10%
    PWM4DCL = 0b00000011;
}

void INT_Init(void)
{
    OPTION_REGbits.INTEDG = 0; //下降沿中斷
    INTF = 0;    //Clear INTF
}

void TIMER1_Init(void)
{
    TMR1H=0xFF;
    TMR1L=0x06;
    T1CONbits.T1CKPS = 0b11;
    T1CONbits.TMR1CS = 0b00;
    PIE1bits.TMR1IE = 1;
    T1CONbits.TMR1ON = 1;
    PIR1bits.TMR1IF = 0;
}

void interrupt SystemISR(void)
{
    if(TMR1IF && TMR1IE)
    {
        TMR1IF = 0;
        TMR1H = 0xFF;
        TMR1L = 0x06;
        mS++;
    }

    if(INTF && INTE)
    {
        INTF = 0;    //Clear INTF
        SW = 1;
    }
}

void main(void)
{
    system_initial();
    TIMER1_Init();
    PWM_Init();

    while(1)
    {
        switch(step)

```

```

{
  case 0:
  {
    PWM4CON = 0b01000000; //disable PWM Module
    LED2=0;

    if(SW==1)
      {
        LED1=0;
        step=1;
      }

    if(ms==60)
      LED1=1;

    if(ms==120)
      LED1=0;

    if(ms==180)
      LED1=1;

    if(ms==240)
      LED1=0;

    if(ms==1000)
      ms=0;

    break;
  }

  case 1: //LED2 PWM mode
  {
    if(SW==1)
      {
        ms=0;
        step=0;
      }

    LED1=0;
    PWM4CON = 0b11000000; //Enable PWM Module

    break;
  }
}
}

```