

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
188 460 ไมโครคอนโทรลเลอร์

Experiment MCL-03: PIC Timer Module

วัตถุประสงค์

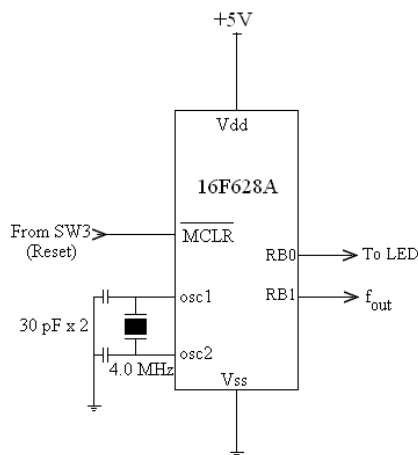
เพื่อศึกษาการทำงานของ Timer module ใน timer mode และ counter mode

อุปกรณ์และเครื่องมือ

1. เครื่องคอมพิวเตอร์พร้อมโปรแกรม MPLAB สำหรับการเขียนโปรแกรมภาษาแอสเซมบลีและ WinPIC800 สำหรับการบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์
2. เครื่องบันทึกโปรแกรมลงในไมโครคอนโทรลเลอร์ ETT EMPIC V2.0 พร้อม power supply และสายเชื่อมต่อกับเครื่องคอมพิวเตอร์
3. PIC MCU 16F628A
4. Crystal Oscillator 4.0 MHz.
5. Ceramic capacitor 30 pF x 2
6. Digital Experiment Box
7. Oscilloscope

ขั้นตอนการทดลอง

1. ต่อดังต่อไปนี้บนกล่องทดลอง Digital box



2. ป้อนและแปลโปรแกรมภาษาแอสเซมบลีต่อไปนี้ใน MPLAB

```
#include "p16f628.inc"
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
    ctr      equ    0x20          ;allocate variable
    org      0x000
    goto     main
    org      0x004
    goto     isr

main:
    movlw   0xff
    banksel TRISB                    ;switch to TRISB reg. bank
    clrf    TRISB                    ;PORTB is output
    movlw   TRISA                     ;PORTA is input
    movlw   b'10000100'
    banksel OPTION_REG               ;set prescaler to TMR0
    movwf   OPTION_REG               ;switch to OPTION reg. bank
    banksel PORTB                     ;switch to PORTB bank
```

```

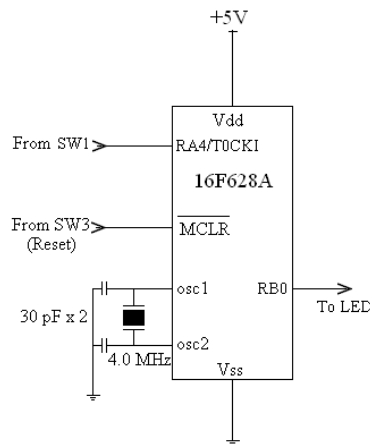
        clrfs PORTB                ;output = 0
        bsf INTCON,T0IE            ;enable TMR0 interrupt
        movlw .96                  ;timer value = 96 (decimal)->(5.088 mS)
        movwf TMR0                ;start timer
        bsf INTCON,GIE            ;enable Global interrupt
        clrfs ctr

loop:
        goto loop                ;wait for timer overflow

isr:
        movlw .96                  ;timer value for 5mS next int.
        movwf TMR0                ;start timer
        bcf INTCON, T0IF          ;clear TMR0 int. flag
        incf ctr,F                ;
        movlw .196                ;196 x 5.088 mS = 1 Sec.
        subwf ctr, W              ;1 sec ?
        btfss STATUS, Z           ;< 1 Sec, return
        retfie
        comf PORTB,F              ;= 1 Sec, toggle port B
        clrfs ctr                ;clear counter
        retfie
end

```

3. ทำการ download hex file ที่ได้จากขั้นตอนที่ 1 ลงในไมโครคอนโทรลเลอร์
4. ทดสอบการทำงาน โดยใช้ oscilloscope วัดสัญญาณที่ออกจาก RA0 และบันทึกความถี่
5. ดัดแปลงวงจรใหม่ดังรูป



6. ป้อนและแปลโปรแกรมภาษาแอสเซมบลีต่อไปใน MPLAB

```

#include "p16f628.inc"
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

org 0x000
goto main
org 0x004
goto isr

main:
        banksel TRISB                ;switch to TRISB reg. bank
        clrfs TRISB                ;PORTB is output
        movlw b'10100001'          ;set prescaler to TMR0
        banksel OPTION_REG          ;switch to OPTION reg. bank
        movwf OPTION_REG           ;set OPTION reg. value
        banksel PORTB              ;switch to PORTB bank
        clrfs PORTB                ;output = 0
        bsf INTCON,T0IE            ;enable TMR0 interrupt
        movlw .255                 ;counter value start at 255 (decimal)
        movwf TMR0                ;start counter
        bsf INTCON,GIE            ;enable Global int

loop:
        goto loop                ;wait for external clock

isr:
        movlw .255                 ;counter value start at 255
        movwf TMR0                ;start counter
        bcf INTCON, T0IF          ;clear TMR0 int. flag
        comf PORTB,F              ;toggle PORTB
        retfie
end

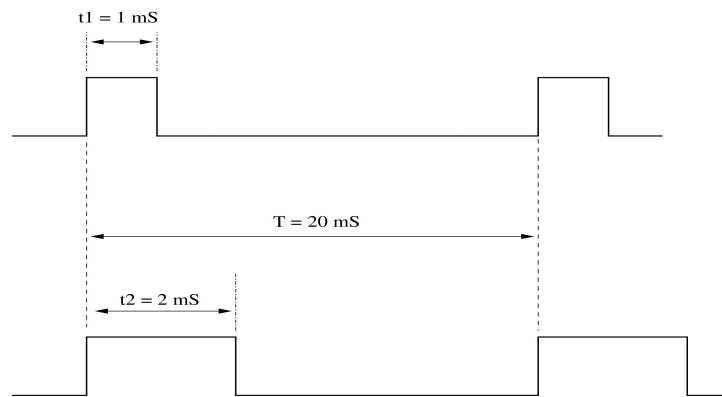
```

7. ทำการ download hex file ที่ได้จากขั้นตอนที่ 6 ลงในไมโครคอนโทรลเลอร์

8. ทดสอบการทำงานโดยกด input switch แล้วสังเกตการเปลี่ยนแปลงสถานะของ LED แสดงผล

คำถามท้ายการทดลอง

ต้องการใช้ PIC 16F628 สร้างสัญญาณ square wave ที่มีคาบเวลา 20 mS โดยมีช่วงเวลา on ที่สามารถเลือกได้ 2 ค่า ได้แก่ 1 mS และ 2 mS ดังภาพข้างล่างนี้ โดยใช้สวิตช์เป็นตัวเลือก



ให้แสดงวงจรที่ออกแบบและอธิบายหลักการคำนวณและการทำงานของซอฟต์แวร์ที่เขียนขึ้น พร้อมทั้งแสดงผลรูปคลื่นที่ได้

**หมายเหตุ: ในการออกแบบ ให้เลือกใช้ Internal oscillator ของ 16F628 ซึ่งมีความถี่ $f_{\text{osc}} = 4.0 \text{ MHz}$ ในการคำนวณ และในโปรแกรม WinPIC800 ให้ใช้ตัวเลือก INTRC I/O ในส่วนของ OSC