



## **Objectives**

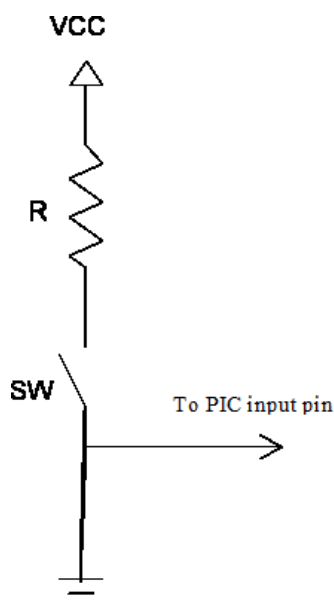
- 1- To become familiar with Pulse Width Modulation in software.
- 2- To demonstrate the use of external interrupts linked with the port B on-change.

## **Pre-lab Preparation:**

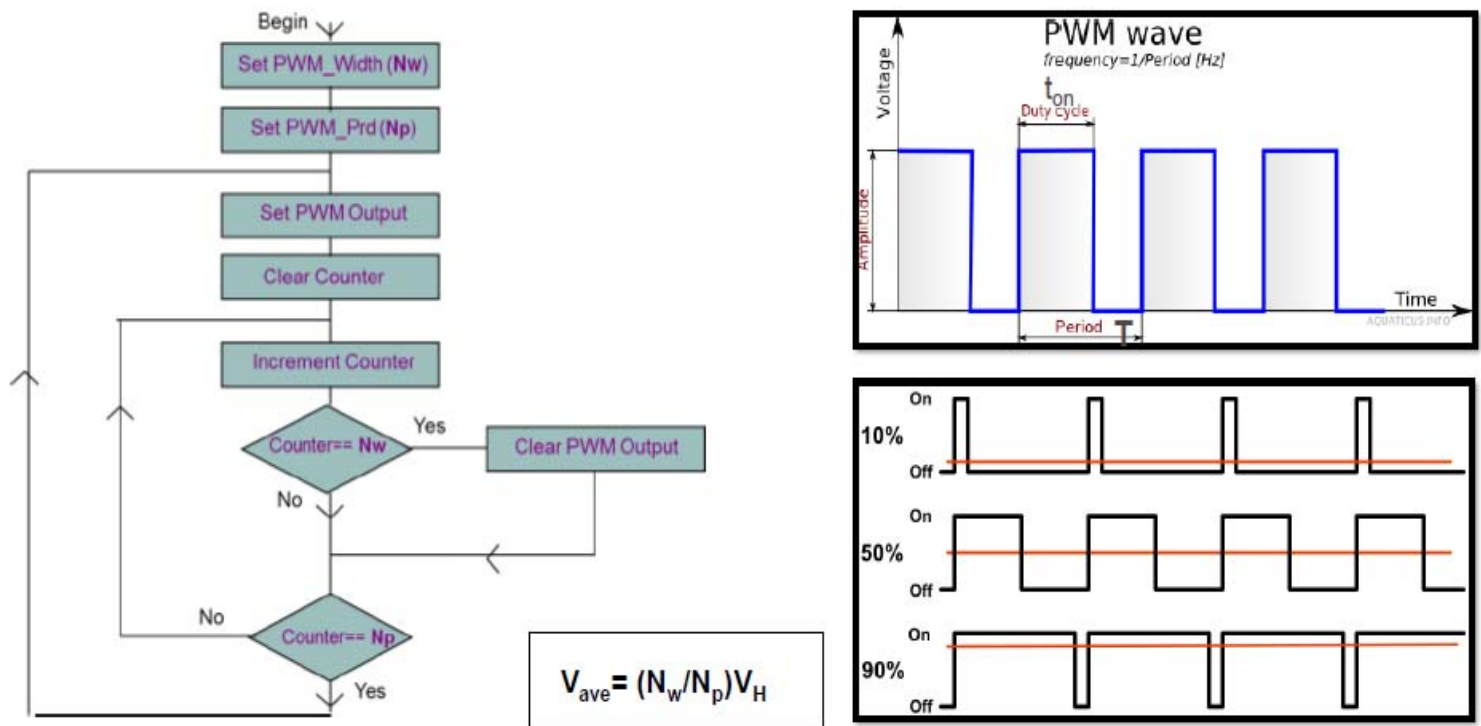
- 1- Review the sections in the book regarding PWM (Chapter 7).
- 2- Review the instruction set of the PIC 16F877.
- 3- Read the assembly program carefully.

## **Procedure:**

In this experiment, we are going to use four pushbuttons on the board connected with the supply voltage to produce logic 1 (5Volt). Each pushbutton will set a different value that used to pulse width modulate, a signal is connected to the light bulb on the board to get a visual indication of the affect of the different values on the PWM signal.



*Fig.1: - Switch connecting to PIC*



*Fig.2: - PWM Flow Chart and Concept*

```

;*****
; Lab11.asm
;This program operates a car lighting system using 4 pushbuttons
;Each pushbutton will set a different value that used to pulse width modulate
;pushbuttons are connected to the pins of port B (RB4-RB7).
; light bulb is connected to 6.
; The program uses a PIC16F877A running at crystal oscillator of frequency 4MHz.
;*****
    include "p16f877A.inc"
;*****
; Macro definitions
push    macro
    movwf    WTemp        ; WTemp must be reserved in all banks
    swapf    STATUS,W      ; store in W without affecting status bits
    banksel   StatusTemp    ; select StatusTemp bank
    movwf    StatusTemp    ; save STATUS
endm
pop     macro
    banksel   StatusTemp    ; point to StatusTemp bank
    swapf    StatusTemp,W   ; unswap STATUS nibbles into W
    movwf    STATUS        ; restore STATUS
    swapf    WTemp,F        ; unswap W nibbles
    swapf    WTemp,W        ; restore W without affecting STATUS
endm
;*****
Sec_1    equ    D'10'        ; Number of centiseconds in a second
CountOuter0 equ    D'10'
CountInner0 equ    D'250'
;*****
; User-defined variables
    cblock    0x20            ; bank 0 assignments
                WTemp        ; WTemp must be reserved in all banks
                StatusTemp

```

```

        PWM_Width
        PWM_Period
        Counter
        BLNKCNT
        CountOuter
        CountInner

    endc

cblock    0x0A0        ; bank 1 assignnments
        WTemp1        ; bank 1 WTemp

    endc

cblock    0x120        ; bank 2 assignnments
        WTemp2        ; bank 2 WTemp

    endc

cblock    0x1A0        ; bank 3 assignnments
        WTemp3        ; bank 3 WTemp

    endc

;*****
; Start of executable code

    org        0x000

    nop

    goto       Initial

;*****

; Interrupt vector

    org        0x0004

    goto       INT_SVC        ; jump to the interrupt service routine

;*****

; Initial Routine

Initial

    banksel    PORTC

    clrf       PORTC        ;Clear PORTC

    bsf        INTCON,GIE    ;Enable Global Interrupt

    bsf        INTCON,RBIE   ;Enable RB Port Change Interrupt

```

```

    banksel    TRISC
    clrf       TRISC           ; All of the PORTC bits are outputs
    movlw     0xF0
    movwf     TRISB           ;Set port B pins (RB0-RB3 outputs, RB4-Rb7 inputs)
    banksel    ADCON0         ; Select register bank 0
    clrf      PWM_Width
    clrf      PWM_Period
;*****
; Main Routine
Main
    sleep
    comf      PWM_Period
L1
    bsf       PORTC,RC6       ;Set PWM signal to RC6
    clrf      Counter
L2
    incf      Counter,F
    movf      PWM_Width,w
    subwf     Counter,w
    btfsc     STATUS,Z
    bcf       PORTC,RC6       ;clear PWM signal from RC6
    movf      PWM_Period,w
    subwf     Counter,w
    btfsc     STATUS,Z
    goto      L1
    goto      L2
;*****
; Interrupt Service Routine
INT_SVC
    push
    call     Duty_Select
    pop
    retfie

```

```

;*****
; Duty_Select Routine
Duty_Select
    btfsc    PORTB,RB4
    goto    Duty_25
    btfsc    PORTB,RB5
    goto    Duty_50
    btfsc    PORTB,RB6
    goto    Duty_75
    btfsc    PORTB,RB7
    goto    Duty_100
    goto    Cont

Duty_25
    movlw d'64'
    movwf PWM_Width
    goto    Cont

Duty_50
    movlw d'128'
    movwf PWM_Width
    goto    Cont

Duty_75
    movlw d'192'
    movwf PWM_Width
    goto    Cont

Duty_100
    movlw d'255'
    movwf PWM_Width

Cont
    call    Delay
    movf    PORTB,w
    bcf     INTCON,RBIF
    return

```

;;;;;;;;; Delay subroutine ;;;;;;;;;;;;;;;;;;;;;;;;;;  
; This subroutine to get a delay with 100 mSec.  
Delay

Sec    movlw Sec\_1  
      movwf BLNKCNT

TenMs  
      movlw CountOuter0  
      movwf CountOuter

DecO  
      movlw CountInner0  
      movwf CountInner

DecI  
      nop  
      decfsz CountInner, F  
      goto    DecI  
      decfsz CountOuter, F  
      goto    DecO  
      decfsz BLNKCNT, F  
      goto    TenMs  
      Return

.\*\*\*\*\*  
,  
      end