Pulse Width Modulation

Introduction.

Pulse Width Modulation is a technique to provide an output logic one for a period of time and a logic for the balance of the time.

This discussion discusses a few applications and then illustrates a simple algorithm for use with a PIC.

Applications.

When driving motors one frequently desires to change the speed by varying the voltage applied to the winding. However, it is too expensive to use a linear amplifier which provides 0 to 12 Volts at say 2 Amps at 12 Volts.

The technique which is usually used in to turn on a transistor or power FET for a period of time and then turn it off for a period of time. For example, if the voltage is +12 Volts and the transistor is on 25, 50 or 75 percent of the time, the effective voltage is 3.0, 6.0 and 9.0 Volts.

The above applies to lighting a lamp as well.

Another application is to charge a capacitor through a suitable current limiting resistor. Thus, the output assumes either a logic one or zero corresponding to nominally +5.0 and 0.0 volts. However, the average voltage appearing across the capacitor is the duty cylcle times +5.0. For example, if one desires 2.5 Volts, a 50 percent duty cycle might be used.

I have used this technique to display parameters on a 200 mV panel meter. Although this is discussed in the context of a Basic Stamp, the idea is applicable to PICs as well. (See http://www.access.digex.net/~pha/stamp).

Note that in this application, it is important that when the output is not being modulated with a precise duty cycle that the ouput be placed in a high impedance mode so as to avoid charging or discharging the capacitor.

An Algorithm.

Assume an 8-bit variable DUTY has a value in the range of 0 through 255. If it is n, then an output is to be high n/256 of the time and low for the balance.

Consider an accumulator where on each pass ACC = ACC + DUTY and if no carry occurs, the lead is brought low. But if a carry does occur, the lead is brought to a logic one.

Assume the loop is repeated 256 times.

Thus, if DUTY is zero, a carry never occurs and the output is always low. If DUTY is one, a carry will occur when ACC rolls over from FFH to 00H. This will occur once in 256 loops. Thus the output will be at a logic one, 1/256 of the time. Similarly, if DUTY is 2, ACC will roll over twice. Thus, the output will be a logic one, 2/256 of the time. If DUTY is 255, a carry will always occur, except when ACC is zero. Thus, the output will be high 255/256 of the time.

Note that if the process is looped 256 times, the initial value of ACC may be any value. That is, zero added to any number never results in a carry, one added to anything results in a carry on only one pass through the loop, etc.

In the following program, DUTY begins at zero and ramps up to 255 and then down to zero. The program continually loops.

In routine, PWM, the W register is used as the accumulator. On each pass through the loop, W is replaced by W + DUTY and if there was a carry, the output is brought to a logic one. Otherwise, the output is cleared to zero. Note that on entering the routine, PORTB.0 is brought out of the high impedance state and prior to leaving the routine, it is again placed in the high impedance state.

Note that 256 passes through the loop is achieved by initializing PWM_LOOP to zero. Note that the decrement (to 0FFH) occurs before testing as to whether it is zero.

Note that as presented, the time between passes through the loop is seven clock cycles or 7 usecs if using a 4.0 MHz clock. In some cases, it may be desirable to have the output assume the state for a longer period of time and this may be done by inserting a delay at the point noted. Of course, if a routine is called which uses the W register remember to save the ACC value in a temporary variable.

```
; Program PWM.ASM
;
; Illustrates use of the PWM command.  Average voltage at PORTB.0 is
; ramped up from zero and then down from near +5.0.  Process repeats.
;
; Peter H. Anderson, MSU, August 7, '97
;

        LIST p=16c84
#include &ltc:\mplab\p16c84.inc>
        __CONFIG 11h

        CONSTANT  BASE_VAR=0CH

PWM_DUTY EQU BASE_VAR+0
PWM_LOOP EQU BASE_VAR+1

MAIN:
        BSF STATUS, RP0
        BSF TRISB, 0            ; make pin RB.0 an input, high impedance
        BCF STATUS, RP0

        CLRF PWM_DUTY           ; start duty cycle at minimum
MAIN_1:
        CALL PWM
        INCFSZ PWM_DUTY, F      ; and ramp up
        GOTO MAIN_1
MAIN_2:
        CALL PWM                ; now, ramp down
        DECFSZ PWM_DUTY, F
        GOTO MAIN_2
        GOTO MAIN

PWM:
        CLRF PWM_LOOP           ; 256 counter

        BSF STATUS, RP0         ; bank 1
        BCF TRISB, 0            ; make PORTB.0 an output
        BCF STATUS, RP0         ; back to bank 0

PWM_1:
        ADDWF PWM_DUTY, W       ; accum = accum + duty
        BTFSS STATUS, C
        BCF PORTB, 0
        BTFSC STATUS, C
        BSF PORTB, 0
                                ; a delay may be inserted here
        DECFSZ PWM_LOOP, F      ; do this 256 times

        GOTO PWM_1

        BSF STATUS, RP0
        BSF TRISB, 0            ; make PORTB.0 an input
        BCF STATUS, RP0
        RETURN

        END
```