INTRODUCTION

Pulse Width Modulation (PWM) modules are commonly used in many applications to provide an inexpensive control output method that uses only a few external components. The PWM signal can be used directly as a digital signal to drive switches in a power conversion circuit. Or, it can be filtered using external components to produce an averaged 'analog' signal with an output level that is proportional to the duty cycle. Either way, the duty cycle of the PWM signal determines the level of system output while the frequency remains fixed. The typical PICmicro® PWM module (CCP/ECCP) is ideally designed to support these common types of applications providing high duty cycle resolution for a given fixed frequency.

Variable Frequency, Fixed Duty Cycle Applications

In this application note, we will illustrate a simple technique that allows all PICmicro PWM modules to support a different class of applications, including more specifically several lighting applications, where the duty cycle is required to be constant and it is the output frequency that changes in small increments. In fluorescent and high intensity discharge (HID) electronic ballasts for example, the frequency variation is used to control the impedance of an inductor (the ballast) in series with the lamp. To keep the ballast inductor small (reducing cost and size), the switching frequency must be relatively high, in the typical range of 80kHz to 100kHz. But to allow for an optimal control of the current in the lamp, the frequency is required to be controlled in small increments while maintaining a fixed 50% duty cycle. In other words, these applications require high frequency resolution and fixed duty cycle.

The typical PICmicro MCU CCP and ECCP module is based on the structure represented in Figure 1.

A Technique to Increase the Frequency Resolution of PICmicro® MCU PWM Modules
A prescaler is available to reduce the input clock frequency by three fixed possible ratios of 1:1, 1:4 and 1:16. For the high frequencies required in lighting applications, the 1:1 ratio must be selected and the Period Register PR2 (PR4) is used to control the actual PWM period. The following equation helps determine the correct timer configuration for a given PWM frequency and clock frequency pair:

**EQUATION 1:**

\[ PR2 = \frac{F_{osc}}{4 \cdot \text{Prescaler} \cdot \text{FPWM}} - 1 \]

Given a 40 MHz clock signal and a desired 100 kHz PWM frequency, setting the prescaler to the 1:1 ratio, we obtain \( PR2 = 99 \). Solving Equation 2 for FPWM, we obtain:

**EQUATION 2:**

\[ FPWM = \frac{F_{osc}}{4 \cdot \text{Prescaler} \cdot (PR2 + 1)} \]

By incrementing and decrementing PR2 in small increments around the central period register value, we can observe that the actual frequency resolution (step) provided by the CCP/ECCP module is in the range of 1 kHz.

**TABLE 1: CCP/ECCP FREQUENCY RESOLUTION @ 100 KHZ**

<table>
<thead>
<tr>
<th>PR2</th>
<th>FPWM (Hz)</th>
<th>Step (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>97,087</td>
<td>934</td>
</tr>
<tr>
<td>102</td>
<td>98,039</td>
<td>952</td>
</tr>
<tr>
<td>101</td>
<td>99,009</td>
<td>971</td>
</tr>
<tr>
<td>100</td>
<td>100,000</td>
<td>990</td>
</tr>
<tr>
<td>99</td>
<td>101,010</td>
<td>1010</td>
</tr>
<tr>
<td>98</td>
<td>102,040</td>
<td>1031</td>
</tr>
<tr>
<td>97</td>
<td>103,092</td>
<td>1052</td>
</tr>
</tbody>
</table>

If used in dimmable ballast, this resolution would not be sufficient to provide a smooth dimming effect, especially at the low range of the lamp intensity scale where the human eye is the most sensitive.

**Fractional Frequency Increment**

In order to provide steps of about 60 Hz with a digital PWM peripheral (a commonly used reference value), we would need to increase the clock frequency by a factor of 16 or 640 MHz, a costly and technically challenging proposition. But there is a simpler and inexpensive solution that can be adopted using the interrupt mechanism associated to the CCP/ECCP modules and only a few lines of code. The basic idea consists of considering groups of 16 PWM periods at a time, and alternating between two discrete frequency values (two contiguous values of the PR2 register). For example alternating 8 periods with PR2=100 and 8 periods with PR2 = 99, we will obtain an average frequency of 100,500 Hz. By using other ratios 1:16, 2:16, 3:16...15:16, we will be able to produce 14 intermediate steps equally spaced by about 64 Hz increments, between the 100,000 Hz and the 101,010 Hz values. In a lighting application, the human eye will naturally integrate the luminous output and perceive as if the overall resolution was in fact increased by a factor of 16.

The simplest algorithm suitable to implement such mechanism would utilize a counter and perform a number of cycles, equal to the desired fraction, at the lower frequency (T1), followed by the complementary number of cycles at the higher frequency (T2) as shown in Figure 4.
But this method would add an undesirable strong second harmonic component to the output signal. A better result can be obtained by interspersing periods of the two frequencies as evenly as possible as depicted in Figure 5.

To obtain the evenly spaced distribution of periods, a 4-bit accumulator is used and at each cycle the chosen fractional value (1...15) is added to it. If a carry is generated the following period will be extended (T1), otherwise, it will be of base value (T2).
A demonstration for the PIC18F1220

The example code provided in Appendix A, illustrates the simplicity of the solution as implemented in a general purpose PIC18 microcontroller.

The PIC18F1220 model was chosen as it represents one of the smallest and most inexpensive PIC18 devices available and it features an ECCP module that can produce PWM complementary signals as required to drive a half bridge ("push-pull") output MOSFET stage as typically implemented in several ballast applications.

In particular, the fractional counter technique is implemented in only 12 instructions contained in the interrupt service routine:

```
isr
    bcf CCPICON,DC1B1
    bcf PIR1,TMR2IF      ; clear the interrupt flag
    movf FRAC,W
    addwf FACC,W         ; add the FRAC to the accumulator
    movf PERIOD,W        ; get the base period value in W
    btfss FACC,4          ; if there was a carry in the fractional accumulator
    goto setpr2
    incf WREG,W          ; increase the period by 1
    bsf CCPICON,DC1B1    ; increase duty by 2xTq to keep it 50%
setpr2
    movwf PR2            ; update the next period value

; <<<< for demonstration only
    btfsc FACC,4         ; signal longer period (T1)
    btfss FACC,4         ; signal shorter period (T2)
;
; >>>> for demonstration only
    bcf FACC,4           ; clear the carry bit
    isr
    retfie 1             ; return (fast) restoring the shadow registers
```

Four additional instructions have been added to drive one extra output pin (RB0) and help visualize the alternating sequence of T1 and T2 periods. Pin RB0 is toggled each time the period of the output signal is changed as a timing reference.

The graph in Figure 7 has been recorded using the MPLAB SIM simulator and taking a snapshot of the Logic Analyzer window.
Figure 6: Snapshot of MPLAB SIM Logic Analyzer Window 16 Cycles Group

The ECCPA waveform represents the ECCP module output. Since a ratio of 5:16 was chosen for the demonstration, we can count 5 x T1 periods of 101 cycles each (marked by RB0 high) and eleven x T2 periods of 100 cycles each for every group of 16 PWM periods. The grand total adds up exactly to 1,605 cycles.

Figure 7: Measuring a T2 Period
FIGURE 8: MEASURING A T1 PERIOD

SUMMARY

This application note shows how to generate a variable frequency digital signal with good frequency resolution using a combination of on-chip hardware and software. The provided code example generates a 100 kHz signal that can be adjusted in steps of 64 Hz, while using only 13% of the available CPU cycles thanks to the use of the PIC18 shadow registers fast interrupt context save features.

The code presented here can easily be modified to be utilized on PIC16 (mid-range) microcontrollers although with a slightly higher CPU overhead and/or to produce higher frequency resolutions by working on larger cycle groups.
APPENDIX A: DEMONSTRATION CODE FOR PIC18F1220

```assembly
PROCESSOR PIC18F1220
RADIX HEX

; Enhancing the CCP/ECCP frequency resolution for lighting applications
;
; This technique allows a very high frequency PWM (100kHz) signal to be generated
; while providing extremely small frequency increments (60Hz)
;
INCLUDE "p18f1220.inc"

;-----------------------------------------------
; timing definitions in kHz
#define CLOCK .10000 ; Tcy = 40MHz/4 = 10MHz
#define NFREQ .100 ; nominal frequency 100kHz
#define IPERIOD (CLOCK/NFREQ)-1 ; calculating the base period
#define IFRAC .5 ; 4 bit(0-15) augmented resolution
;-----------------------------------------------

; RAM allocation
CBLOCK 0
PERIOD ; integer period
FRAC ; fractional period (0-15)
FAcc ; fractional accumulator
ENDC

;-----------------------------------------------
; port definitions
#define OUT PORTB,0 ; for demonstration only
;-----------------------------------------------

ORG 0 ; reset vector
resetv
goto init

;-----------------------------------------------
ORG 08 ; high priority interrupt vector
isr
bcf CCP1CON,DC1B1
bcf PIR1,TMR2IF ; clear the interrupt flag
movf FRAC,W
addwf FAcc,F ; add the FRAC to the accumulator
movf PERIOD,W ; get the base period value in W
btfss FAcc,4 ; if there was a carry in the fractional accumulator
goto setpr2
incf WREG,W ; increase the period by 1
bsf CCP1CON,DC1B1 ; increase duty by 2xTq to keep it 50%
setpr2
movf PR2 ; update the next period value
```

---

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the “Company”) is intended and supplied to you, the Company’s customer, for use solely and exclusively with products manufactured by the Company. The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
; <<< for demonstration only
btfsc FACC,4
bsf OUT ; signal longer period (T1)
btfss FACC,4
bcf OUT ; signal shorter period (T2)
; >>> for demonstration only
bcf FACC,4 ; clear the carry bit
isre
retfie 1 ; return (fast) restoring the shadow registers

; total ISR time = 13 cycles or 13% MCU load @100kHz/40MHz
;----------------------------------------------------------------------

setPWM
; save the required PWM period value
movwf PERIOD
; set the initial period register value PR2
movwf PR2
; set the duty cycle to 50%
incf WREG,W ; PERIOD+1 is the actual total cycle count
bcf STATUS,C ; divide by 2
rrcf WREG,F ; shifting right
movwf CCPR1L ; set the duty cycle
return
;----------------------------------------------------------------------

init
; init the output port
movlw b'00000000'
movwf TRISB
; disable analog inputs
setf ADCON1
; set CCP module in PWM mode
movlw b'00001100'
movwf CCP1CON
; set the tmr2 to generate the desired frquency and 50% duty
movlw b'00000100' ; prescale 0, postscale 0, tmr2 ON
movwf T2CON
; init the period value
movlw IPERIOD
call setPWM ; set the PWM and duty cycle
; then init the FRACTIONAL divider for the demo
movlw IFRAC
movwf FRAC ; init the fractional period part
; clear the factional accumulator
clrf FACC ; clear the accumulator
; then init the interrupt on CCP1/TMR2
bcf PIR1,TMR2IF
bsf PIE1,TMR2IE
; init global and peripheral interrupts
bsf INTCON,PEIE
bsf INTCON,GIE
;----------------------------------------------------------------------
main
goto main

end
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rFIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, Mindi, MiWi, MPASM, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rFLAB, rFICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2006, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

printed on recycled paper.