# Lab 5:

**CCP - PWM Mode, Program Memory Paging**

**Name:**\_\_\_\_\_\_\_REZA SHISHEIE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_\_2708062\_\_\_\_\_\_\_\_\_\_  
  
**Give brief answers to the following questions. You can edit this document and insert your answers after each question.**

**Due dates:**

**MW – Wed, Mar 21, beginning of class  
TTH – Tue, Mar 20, beginning of class**

**Circle one: MW or TTH**

1. (1 pt) What tristate register and bit must be cleared in order to use the PWM feature of CCP2?  
     
   **Ans.**

TRISC<1> must be cleared to make CCP2 pin as output

1. (1 pt) CCP1 and CCP2 can be used to obtain two PWM outputs, but the two PWM signals must have the same frequency. Why? (Hint: data sheet)  
     
   **Ans.**The PWMs will have the same frequency and update rate since both of them are based on TMR2 interrupt
2. (1 pt) Considering the PIC’s hardware stack,  
     
   a) How many bits are in each level of the hardware stack?   
     
   **Ans.**

The PIC16F87X family has an 8-level deep x 13-bit wide hardware stack.  
  
b) How many bits do we need to completely specify a program memory address?   
  
**Ans.**   
2^13 = 8192 —> 13 bits needed

1. (1 pt) If we jump across program memory page boundaries with a call statement, do we need to change the page select bits before the return instruction? Why or why not?  
     
   **Ans**.

There are 11 bits for the address in the calland gotoinstructions. (211= 2048). Program memory requires 13 bits. (213= 8192). 11 bits only specify the relativeaddresson each page. There are not enough address bits in the call instruction to include all the bits of the Sub1 address. We need two more bits to specify a 13-bit address –these two bits are called the “page select bits” The page select bits PC<12:11> in the PC register cannot be accessed directly. We must use the PCLATH<4:3> bits. Before a call or goto, the PIC looks at the page select bits to decide which page to transfer to. Instead of using bcf or bsf to set the page select bits in PCLATH, we can use the pagesel directive; for example, pagesel SUB1.

Thus the answer is yes we have to change the page select bits. Otherwise the call will go to a wring location.

1. (1 pt) lab05.asm does not use the entire range of the potentiometer because the PWM duty cycle reaches 100% before the pot output reaches 5 volts. What value should PR2 be set to (in decimal) to use the entire range of the potentiometer output?  
     
   **Ans.**  
   Since the output of ADC goes to a ADRESH and it is an 8-bit long register, RP2 should be set to 255 to be able to cover the whole range from 0 to 255.
2. (1 pt) What should be written to PR2 and T2CON to obtain a PWM signal with the lowest possible frequency? What is the lowest frequency? Use ‘x’ for ‘don’t care’ bits.   
     
   **Ans.**PR2 = 0

T2CON= xxxx x100

TOUTPS3:TOUTPS0: do not care

TMR2ON =1 -> Timer2 on

T2CKPS1:T2CKPS0 = 00 -> prescaler = 1

PWM period = (Timer2 Prescale) (PR2 + 1) (4 Tosc) = (1)(1)(1.085μs) = 1.085μs μs

1. (2 pts ) For the lab05 project, the value in the ADRESH register is controlled by the potentiometer setting.  
     
   a) At what value of ADRESH will the on-time equal one half of the PWM period? Give your answer in binary, hex, and decimal.  
     
   **Ans.**ADRESH (bin) = \_\_1100100\_\_\_  
     
   ADRESH (hex) = \_\_0x64\_\_\_\_\_\_\_   
     
   ADRESH (dec) = \_\_100\_\_\_\_\_\_\_\_  
     
      
   b) Assuming the ADC reference voltages are 0 and 5 volts, what analog voltage does this correspond to at the AN0 pin (to the nearest hundredth of a volt)?

**Ans.**  
maximum voltage equal to 100 duty cycle: 200/255\*5= 3.9215

Half duty cycle = 3.9215/2 = 1.96 v

1. (1 pt) What does the assembly directive “$” stand for in the following code: **goto $ + 3**. What does the instruction do?  
     
   **Ans.**   
   It's a jump, relative to the program counter. If the current counter is at 100 goto will fo to 100+3 = 103 in the next step
2. (2 pt) For the instruction goto 0x18DC.   
     
   a) Write the opcode in binary.  
     
   **Ans.**  
   GOTO = 10 1kkk kkkk kkkk

0x18DC = 1 1000 1101 1100

goto 0x18DC = 10 11101 1101 1100  
  
b) Which page contains the address 0x18DC?   
  
**Ans.**   
page number = 11b = page3 or the fourth page  
  
c) Suppose PCLATH<4:3> = 10. To which program memory address (in hex) on which page will the PC jump after the goto 0x18DC instruction?  
  
**Ans.**   
0x18DC = 1 1000 1101 1100

address = 1 0000 1101 1100 = 0x10DC  
  
d) Which two assembly code instructions should you execute before the goto 0x1800 instruction to insure that the PC jumps to the correct address? Which assembler directive could you use in place of those two instructions?   
  
**Ans.**   
0x1800 = 11 000 0000 0000:

bsf PCLATH, 4

bsf PCLATH, 3

the following code can be used instead:

pagesel 0x1800

1. (1 pt) What are the minimum and maximum supply voltage of a commercial PIC16F877 with a 4 MHz XC oscillator? (Hint: Section 15.1 in the data sheet)  
     
   **Ans.**  
   FIGURE 15-2: page 150 = 2.0 ~5.5
2. (1 pt) What is the limit to which *VDD* can be lowered without losing the RAM data?  
     
   **Ans.**

1.5 v

1. (1 pt) If brown-out reset is enabled, under what two conditions will the PIC reset? (Hint: Section 12.7 in the data sheet)  
     
   **Ans.**  
   If VDD falls below VBOR (parameter D005, about 4V) for longer than TBOR (parameter #35, about 100μS), the brown-out situation will reset the device.
2. (1 pt) The device can wake-up from SLEEP through any one of the three events. What are they?  
     
   **Ans.**  
   The user can wake-up from SLEEP through external RESET, Watchdog Timer Wake-up, or through an interrupt.
3. (1 pt) Modify the lab05.asm code by adding the directive org .7000 on the line before the WaitForConversion label. Assemble the code.  
     
   a) What does this directive do?  
     
   **Ans.** ???   
   org set the PC to the new value and advances the PC to the new value  
     
   b) What is the address (in hex and decimal) in program memory of the WaitForConversion label?  
     
   **Ans.**

The address without org is 0x21 or D33 = 00 000 0010 0001

If the org .7000 is added the address would be:

1 1011 0101 1001:­ 0x1B59 = .7001

1. (1 pt) Modify the lab05.asm code so that pressing the RB0 button puts the PIC into sleep mode. If the button is not pressed, what is the voltage at pin RB0? Why? Is the button circuit an active high or active low circuit? Why?  
     
   **Ans.**

The modified code at the end of the main loops is:

btfss PORTB, 0

sleep

if portb finds that bit0 is set to high it skips the next command and since it is active low it gets activated when the button is pressed and the pin is pulled down.

1. (10 pts) Program the PIC for standalone operation. (Note that sleep mode might not work in debug mode.) When the PIC is in sleep mode, the pot will not change the intensity of the LED. Test your code change by setting the ADC input voltage to about 2.5 V and pushing the button to enter the sleep mode. Power the PIC on and off 10 times, each time entering the sleep mode with the 2.5 V setting. Record how many times out of the 10 trials that the LED remains on when you enter the sleep mode and how many times it turns of. Demonstrate your modified lab05 and explain the results to the TA or instructor.

The reason is duty cycle and thus sometimes it is high and sometime low on the duty cycle.

On – on – off – on – off – off – off – off – on – off

On: 4

Off: 6

**Number of times LED remains on out of 10 trials \_\_\_\_\_\_\_\_\_\_**

**Student Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructor/TA Signature:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Date:** \_\_\_\_\_\_\_\_\_\_\_