

VANIER COLLEGE – Computer Engineering Technology – Autumn 2017

Introduction to Microprocessors (247-302)

Lab 4

Subroutine call & MPLAB Debugger

NOTE:

To be completed in ONE lab sessions of 3 hrs.

One report has to be submitted **not later than one week** after the last lab session.

This exercise is to be done **individually** except where specified in the procedure. **Each** student must submit a lab report with original design, observations and conclusions.

OBJECTIVES:

After completing this lab, the student will be able to:

1. Use subroutine call in PIC16F887 assembly programming
2. Familiarized with basic debugging features of MPLAB Debugger

THEORY:

Refer to theory notes of week 7 related to stack and branching, on the principle of subroutine calls in PIC16F887.

Refer to section 4.17 – 4.21 of MPLAB X IDE User's Guide

(<http://ww1.microchip.com/downloads/en/DeviceDoc/52027B.pdf>) on more details about debugging features of MPLAB X IDE.

Lab procedures

PART A: Using subroutine

Note: This lab is based on your PIC16F887 prototype board built in lab 3. Make sure your circuit is correctly designed and connected before you proceed with this lab.

1. Start a new project in MPLABX and name the project *Lab4a*. This lab will be based on your code written for Lab 3, Part B.
2. Modify your code so that the nested loop is now a subroutine call. Example :

```
.
.
.
mloop
    incf    PORTA           ; increment bin_ctr
    call    sDELAY
    goto    mloop

;***** subroutines *****
sDELAY
    clrf    counter1
dloop
    movlw   h'02'
    movwf   nested_ctr
    .
    .
    .
```

3. Compile and download your code to your circuit. The code should give the same results as before.
4. Now, modify your code to work as an auto increment 4-bit binary counter, which displays the 4 bits binary on the 4 LEDs. Compile and download your code to your circuit.

5. Look for the list file of your code (.lst). List file is generated primarily for documentation and includes contents of course file with comments, hex code, memory addresses etc. Attach a copy of your list file in the lab report, carefully examine the list file and answer the following questions:
 - a) Is there any memory space assigned to the "include" directive? Why?
 - b) What is the meaning of the hex code at the left column, beside the variables definition in "cblock" directive?
 - c) What is the implication of directive "ORG"?
 - d) List the address of all the labels used in your program.
 - e) What is the total size of your program code?

PART B: Introduction to MPLAB Debugging features

In Part B, you will explore features of MPLAB Debugger using your code developed from Part A.

6. Refer to section 4.17 of the User's Guide on how to start the debugging of running code.
7. Explore various features of debugger as explained in section 4.18 to 4.21. Describe the functions of each of these debugger tools:



8. Describe the approaches that you used to achieve the following debugging scenario: (example, where should the breakpoint(s) be set, should you step over or step into the code, how can you monitor the value of the register/variables etc?)
 - a) You want to find out if the value of your PORTA has been correctly updated.
 - b) You want to find out what is the changes in PCL and PCLATH when a program execute a subroutine call, and when it returns from subroutine call.

Does the change in these 2 SFRs match your understanding about subroutine call? Explain.
 - c) You want to monitor the change of your counter variables used in delay loop.
9. Include a copy of all your source codes (all the parts), full detailed schematic, and answer all the question in your report. Also includes any observation, analysis if any.