# Assignment #5 Fall 2011
# SYSC-3006 Computer Organization
# Department of Systems and Computer Engineering

---

## Due Date: November 18 @ 7:00 PM. (electronic submission)

***You must submit the files identified below using the electronic Submit application. The submission process will be cancelled at the deadline. No assignments will be accepted via email or on disk.***

***THIS ASSIGNMENT MUST BE DONE IN TEAMS OF TWO STUDENTS.* Only ONE student must submit the assignment, clearly stating who the members of the team are.**
This Assignment should be done using the Pair Programming technique. In order to guarantee that this technique has been used, DETAILED QUESTIONS on this Assignment will be included in the Final Exam. This is part of the evaluation of the assignment; students not being able to answer the questions asked during the Final exam will receive a mark of 0 (zero).

Important instructions about submissions:

- Submit ONLY ONE ASSIGNMENT PER TEAM.
- Each file in your Assignment should include a Header with the names/numbers of BOTH students
- Students submitting TWO copies of the assignment will receive the LOWER of the two marks and will receive a demerit of 20% of the final mark.
- Students who, by mistake submit more than one copy, should contact the instructor BEFORE the deadline to fix the problem.

In this assignment, we will complete building the simple controller for a nightclub. There are five lights you will control (each of which connected to the same circuit in the LED box in our lab). The lab's switch box will be eventually changed by the actual lights (i.e., when a LED turns on, the corresponding light will be turned on).

You can reuse any routines employed in earlier assignments.

**Part I [1.5 marks] (assign51.asm, assign51.exe):** programming the LEDs, subroutines and policies.

We will re-program the previous assignment results using the course policies for subroutines.

In summary, you must:
a. Write a routine that will read the values of the switches. The routine must debounce the switches until you are sure the value obtained is stable
b. Whenever you change a switch, the corresponding LEDs are set.
c. The main program should cycle through each element in the array of lights 4 times and finish. On each cycle, it should call a subroutine that sets the corresponding LEDs (DO NOT WORRY

FOR THE DURATION; MAKE THE DURATION OF EVERY LIGHT CONFIGURATION APPROXIMATELY 1 SECOND).

In order to make the organization of your code easier, you must define and use the following subroutines:

; void setLEDs (unsigned byte bitmap)
; Set all the LEDs according to the given bitmap: xxx54321

; unsigned byte bitmap readSwitches (void)
; Reads the current value of the switches

; void debounce(unsigned byte bitmap)
; checks if the bitmap has changed to debounce the switches

; void traverseLightArray(unsigned byte &lightArray, unsigned word delayInMS; unsigned byte ArraySize);
; traverses the *lightArray* and calls the *setLEDs* function for each element in the array. The
; *delayInMS* tells the delay between each of the lights in the array (use 1000 ms for this
; assignment). *ArraySize* tells the number of elements in the array.

; void delay(void)
; waits for a fraction of second using a for loop (you can use your Assign 4 routine here)

Write the subroutines and a main program that uses them to provide the desired functionality.

**Use the policies defined for subroutines in the course.**

**Part II.A. [1 mark] (assign52.asm, assign52.exe):** Programming the Timer

Modify part I and use the timer to compute timing information precisely. The *delay* subroutine must NOT be used in this assignment, and you must compute the actual *delayInMS* and use it. Use a global variable containing the number of milliseconds since the beginning of your program.

The main program should only be responsible for installing the timer interrupt service routine (saving the previous one), setup the timer and the PIC, and then it must check the current time, convert it to seconds from the beginning of your program, and display that number on screen (you should see a sequentially increasing number once a second).

After 15 seconds, the program must finish, leaving the computer in a stable status.

**Part II.b. BONUS [1 mark] (bonus51.asm; bonus51.exe)**

Extend your solution from Part I and combine it with Part II. The main program will now continuously read the values of the switches. Every time a switch changes, the scheduled information is overridden, and the corresponding switch is turned on.

**Part III [1 mark] (assign53.asm, assign53.exe):** programming the Keyboard ISR

We will now program the keyboard to control the light's update speed. To do so, you have to write a new Keyboard ISR that will recognize:

- 3 commands:
  . +: To make the lights change faster
  . -: To make the lights change slower
  . E to exit from your program.

The first step will be to create an Interrupt Service Routine to interact with the keyboard. In this assignment, the ISR should:

- detect the key pressed
- the main program must display the key pressed on screen.
- If a wrong key is pressed, display an error message

This is independent from Part II.

The main program should be responsible for installing the keyboard interrupt service routine (saving the previous one), and to setup the PIC. The program will then loop waiting for commands. Every time a command is received, it must be displayed on screen, and the main program will also insert a CR/LF to change to the following line.

Commands are not case-sensitive (i.e., you can write upper/lowercase characters).

When the E command is received, the program must finish, leaving the computer in a stable status.

**Assignment 6 Marking Criteria (5 marks + 2 bonus):**

Part I:
2 marks: the program is organized adequately using subroutines. The policies are followed.
2 marks: the program is reading switches and setting/clearing LEDs properly.
2 marks: the program works properly and traverses the array properly.

Part II.a:
2 marks: ISR is installed properly, program exits clean
2 marks: the program displays the time from the start of execution in seconds once per second

Part II.b:
2 marks: Part II.a. is integrated with the switch/led functionality properly
2 marks: the timing information works precisely

Part III
3 marks: your ISR is properly installed and you can capture the keystrokes
1 mark: the main program displays the results properly

No marks will be given if the assignment is late or does not assemble. You may submit incomplete assignments if there are some working portions, but the submission must assemble and run for those portions AND you must clearly identify which parts are complete with comments.

50% of the marks will be deducted if you do not stick to the course policies for defining subroutines, including lack of comments.