Assignment #6 Fall 2011
SYSC-3006 Computer Organization
Department of Systems and Computer Engineering

Due Date: December 5 @ 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 (assign61.asm):

Convert Assignment 5 into an interrupt-driven program. The program will

a. Run the light schedule non-stop
b. If a switch is turned on, that particular light will be set (ignoring the schedule). When the switch is turned off, the light will continue using the schedule.
c. The keyboard should be used to increment/decrement speed or exit the program.

The values in seconds are precise. Do not use timing loops: you should count time using the timer Interrupt only.

The program should cycle until the E key is pressed; at that moment, the program must finish, leaving the computer in a stable status.

You must organize your code using subroutines, using the policies defined in the course.

Part II (assign62.asm):

Extend Part I, allowing a system's user to reprogram the schedule of the lights. To do so, you have to modify your Keyboard ISR. The Keyboard ISR will recognize the following commands:

. A #CONF, #DURATION: it adds, at the end of the schedule, a new light configuration (written in HEX), and the duration of that configuration (in milliseconds).

. M #NUMBER, #CONF, #DURATION: it modifies position #NUMBER in the array, and it puts a new configuration and duration. As above, the new configuration is written in HEX, and the duration of that configuration in milliseconds.

The program should cycle until the E key is pressed; at that moment, the program must finish, leaving the computer in a stable status. The user can accelerate/slow down the display of the lights, change the configuration of the schedule, and flip switches.

The keyboard ISR should interact with the rest of the program. In the case that the reprogramming occurs in the middle of a light cycle, the reprogrammed value can be used immediately or on the next cycle (any of them is correct; simply write a note in your Assignment text specifying which of the two options you have used)

Part III – [BONUS] **Your project here (assign63.asm)**

NOTE: this exercise will be marked by Prof. Wainer.

Now that you know everything about this hardware and how to program: what would you do with it?

The idea of this optional exercise is to allow you to be creative.

You can extend the nightclub light application we built throughout the term, or build a complete application from scratch. You can use any I/O devices.

Mandatory activities:
a. The main program must run in the background, doing not important tasks only. It installs the ISRs, and then executes low priority tasks (for instance, displaying information on screen).
b. You must use the timer and keyboard ISRs. Low priority I/O (for instance, the switches), and could be handled in the main program when no interrupts are running in the foreground.

Your assignment should start explaining, in comments, what the program is doing (in detail). Include any information needed for testing.

Part IV – (assign64.txt) Concept questions

1. What is the PIC? What is it used for? How does it work? Explain in detail.

2. What is a Hardware Interrupt? Explain how Hardware Interrupts provides a mechanism for synchronization between I/O devices and the CPU.

3. What is Interference? What is a Critical Region? How do we implement Critical Regions in the 8086 processor?

4. What is the Timer? How does it work?

Assignment 6 Marking Criteria:

Part I – 12 marks

Part II – 6 marks

Part III – 6 marks

Part IV – 4 marks

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.