Middle East Technical University **Ceng336**
Computer Engineering Department **Spring 2016**

Take Home Exam 1

Revision 2.0

Due: Sunday, April 3, 2016, 23:55:50hrs

Submission: ***via COW***

The purpose of this assignment is to familiarize you with **basic input/output operations** by implementing a simple single player game.

Any clarifications and revisions to the assignment will be posted to the *metu.ceng.course.336* newsgroup.

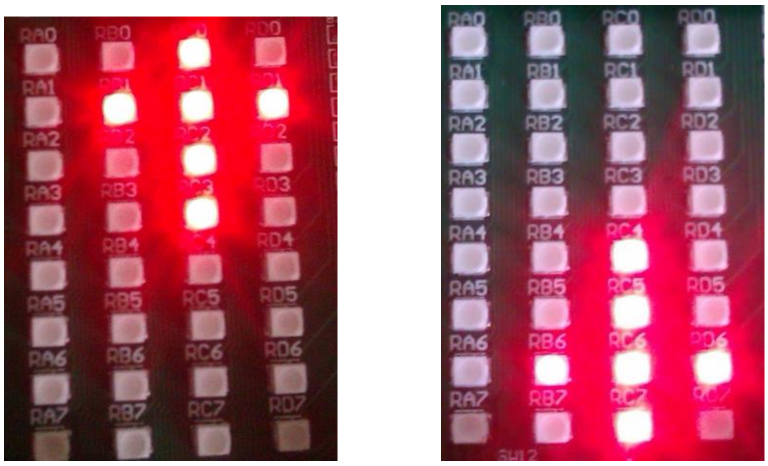
Your Mission

Your mission is to implement a simple game in which the player is expected to guess a number that is randomly generated by the PIC. Detailed specifications of the game are as follows:

Specifications:

- The 7-Segment displays D0, D1, D2, D3 and 18 LEDs (RB1, RB6, RC0-7, RD1, RD6, RE0-5) should be used as output devices while the push buttons such as RF0-5 used as input devices.
- When the PIC is powered up, D3, D2, D1, D0 should show nothing and all LEDs should be turned off until RF5 is pressed and released. After you press RF5, PIC will generate a random number between 0-99, after which the game will start.
- When the game is started by pressing RF5, 6 LEDs which is connected to PORTE should be turned on (from LED0 to LED5), indicating 6 units of initial energy. D1, D0 should show the character "-", and on D2 and D3 number "0" should be displayed.
- After this step, the player is expected to find the randomly generated number by entering his/her own guesses. Number entry will be performed by showing the number using D3 and D2 for a two-digit decimal number. D3 and D2 will represent the tens and units digits of the number, respectively. For the numbers between 0 and 9, the notation will be as "01, 02 … 09". You will use RF4 to increment the number on D3 and RF3 to decrement it. Similarly, RF2 will be used to increment the number on D2 and RF1 to decrement it. After you see the desired number on displays (e.g. for "23" you should see "2" on D3 and "3" and D2), then press the button RF0 to enter or select it. D1, D0 should continue to show "-" during this period.
- When a number is entered, **if it is correct**, D3 and D2, D1 and D0 should flash this number approximately for **three** seconds with a **500 ms time interval**. The correct number should be shown on D1, D0. The number entered by the player should be displayed on D3, D2. By flashing, we mean that the number should appear on D3 and D2, D1 and D0 and then disappear repeatedly, for a total of approximately three seconds.
- If the player enters an **incorrect number**, s/he loses **one** unit of energy. So, you should turn off the top most led (starting from LED0). If the energy of the player becomes 0 (after 6 incorrect attempt), then s/he loses the game. After a loss, the 7-Segment displays (D3, D2) should show the character '-', and the correct number should be displayed on (D1, D0).

- After **an incorrect entry, if the player still has some energy**, a **hint** should appear on the LEDs that leads the player to the correct number. This hint tells the player that if his/her guess is below or above the randomly generated number. If the guess is smaller than the randomly generated number, the LEDs (RB1, RC0-RC3, RD1) shows the character in **Figure 1(a)** (It means that the next guess should be greater than the previous one). If the guess is greater than the randomly generated number, the LEDs (RB6, RC4-7, RD6) shows the character in **Figure 1(b)** (It means that the next guess should be smaller than the previous one). Meanwhile D3 and D2 should keep **the lastly tried number**, also D1 and D0 should display '-' . Then the player enters the next number by modifying the last number on D3 and D2 according to the hint. When the player hits a button to increment or decrement a digit, the hint should disappear and D1 and D0 should show the character “-”.
- To start a new game, the player can use the reset button on the Board.



(a) (b)

Figure 1. Hints to help the player to enhance its guess. (a) and (b) mean that the next guess should be greater and smaller than the previous one, respectively.

Coding Rules

- You will code your program using PIC assembly language.
- Your program should be written for PIC18F87722 working at 40 MHz
- You are provided with a template .asm file named "the1_##.asm" (## represents your group number). You should add this .asm file as a source file in your MPLAB X IDE project. You are expected to write your code in this file. Moreover, "the1_##.asm" contains required function calls, some initial configurations and some directives. (You can download these files from THE 1 page on the COW).
- Timer0 interrupt is used in the template .asm file to generate a random number, however the use of interrupts other than this library provided interrupt is prohibited in this homework. You have to use polling to control buttons. For delays, you have to calculate the amount of cycles in a code segment (like the delays in recitation sample codes) therefore the amount time spent (See the Hint in THE-Warm-up for how to calculate the time spent by a code segment in simulator.).
- You should be aware of whether the player pushes and releases the RF5 button. If so, you should call the subroutine "assignRandomNumber" as shown in the1_##.asm". Then the random number will be saved in register "023h" called "randNumber".
- While using a push button, the displays and LEDs should still keep showing the characters and their status.
- You should update the characters on displays after **releasing** corresponding buttons.
- If you increase the number 0 to 9 on a 7-segment, and then if you press the increment button again, it should keep displaying the last number 9. Then the player should use the

corresponded decrement button to decrease the number.

- If you decrease the number 9 to 0 on a 7-segment, and then if you press the decrement button again, it should keep displaying the last number 0. Then the player should use the corresponded increment button to increase the number.

Switch Configurations for UNIDS6

- The J13 switch should be on VCC.
- In SW12 switch, the following pins between 1 and 6 should be ON. the others should be OFF.
- The remaining switches should be OFF.

Resources

- PIC 18F87722 Datasheet
- PIC Development Tool User Manual
- PIC Development Tool Programming Manual
- Course web page and newsgroup

7-Segment Displays and LEDs

There are three common cathode 7-Segment displays mounted on the development board. PORTJ and PORTH are used as data bus and selection pins, respectively, as illustrated in Figure 3. Notice that PORTH pins are connected to all displays.

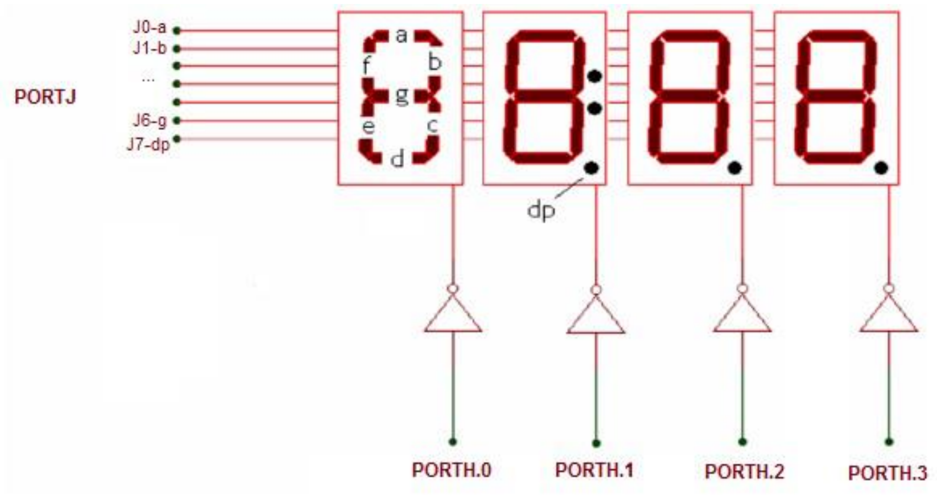


Figure 3. Connections for the 7-Segment displays in the MCDEV board.

As an example, if you want to show number "4" on leftmost display (D0), you should first select it by setting RH0 and clearing RH1, RH2 and RH3, then send binary \01100110" to PORTJ. Hence, a, d, e segments on D0 will be turned off, and b, c, f, g segments will be turned on. Note that RJ7 pin is used for dp of displays. Also note that there is no dp on D0 (leftmost 7-segment display) and there are 3 dp on D1 which run simultaneously.

If you want to show, for instance some value(1234) on the displays, you should select only D0 by using PORTH, write the byte necessary to turn on segments to PORTJ, and wait for a while. Then you should do the same for D1, D2 and D3. This is illustrated in figure below. If you adjust "on" times properly and repeat on-off cycles continuously, you show your values on the displays in a smooth manner without flicker.

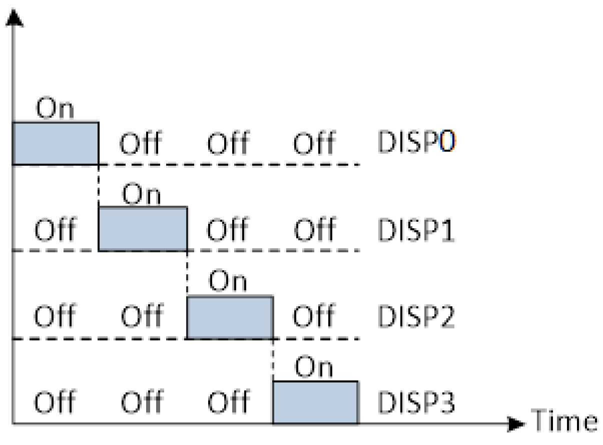


Figure 5. Graphical illustration to show characters 7-Segment displays simultaneously.

Hand In Instructions

- You should submit your code as a single file named as `the1_##.asm` through COW where ## represents your group number. Do not forget replace ## with your group number.
- At the top of `the1_##.asm`, you should write ID, name and surname of **both group members and group number** as commented out.

Only one of the group members should submit the code. Please pay attention to this, since if both members make submission, speed of grading process will be negatively affected.

Grading

You should include **brief but descriptive comments** in your code, including a larger comment in the beginning of your file describing the structure of your program with subroutine descriptions (For example random number generator function usage, button controls etc.) and their relations to one another. Your grade will also depends on the quality of your design, not just its functional correctness. If there is an unclear part in your code, we may ask any of the group member to describe that code segment. Also, group members may get different grades. We reserve the right to evaluate some or all of the groups to determine the contribution of each group member to the exam. For grading we will compile and load your program to the development board and run it for the following checks.

- Proper displaying the characters on 7-segment display. Too low/high brightness and flicker will reduce your points.
- Proper control of push buttons. Only RF0-5 will be pressed during grading. You do not have to manage other buttons.
- Proper usage of the LEDs that is responsible for energy levels and the hint given by your program.
- Functional correctness of the game and its ease of use.

Group members may get **different** grades. We reserve the right to evaluate some or all of the groups to determine the contribution of each group member to the assignment.

Cheating

We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.

Cheating Policy: Students/Groups may discuss the concepts among themselves or with the instructor or the assistants. However, when it comes to doing the actual work, it must be done by the student/group alone. As soon as you start to write your solution or type it, you should work alone. In other words, if you are copying text directly from someone else - whether copying files or typing from someone else's notes or typing while they dictate - then you are cheating (committing plagiarism, to be more exact). This is true regardless of whether the source is a classmate, a former student, a website, a program listing found in the trash, or whatever. Furthermore, plagiarism even on a small part of the program is cheating. Also, starting out with code that you did not write, and modifying it to look like your own is cheating. Aiding someone else's cheating also constitutes cheating. Leaving your program in plain sight or leaving a computer without logging out, thereby leaving your programs open to copying, may constitute cheating depending upon the circumstances. Consequently, you should always take care to prevent others from copying your programs, as it certainly leaves you open to accusations of cheating. We have automated tools to determine cheating. Both parties involved in cheating will be subject to disciplinary action. [Adapted from http://www.seas.upenn.edu/~cis330/main.html]