İzmir Katip Çelebi University

Department of Electrical and Electronics Engineering

EEE316 Microprocessors

Spring 2020

Experiment V

Simulating I/O Ports

Pre-Lab Report

- Please study related topics in reference notes.
- Answer the questions under the lab activities. Prepare the report in the specified format. Reports must be completed before coming to the lab.
- Submit your report to CANVAS until April 29, 23:59.

Experimental Work

- Please explain your code step by step to instructors during lab hours.

Lab Objectives

- Embedded programming with C language
- Getting familiar with parallel ports
- To understand how to use 7 segment display

References

- Lecture notes
- Mazidi, McKinlay, Causey “PIC Microcontroller and Embedded Systems,” Chapter 7

Introduction

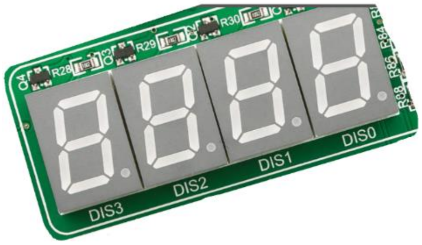


Figure 1 Four 7-Segment Display

There are four seven-segment displays on the application board in Figure 1. These displays are often used to show some numbers to the user. As its name implies, each display unit is composed of seven LEDs. Each LED is attached to one pin of the port D.

To enable digit select lines for the 4-digit 7-segment display you have to turn on **SW4.1, SW4.2, SW4.3** and **SW4.4** switches. Digit select lines are connected to **RA0 – RA3** pins on the microcontroller sockets, while data lines are connected to **RD0 – RD7** pins as shown in Figure 2. Make sure to disconnect other peripherals from the interface lines in order not to interfere with signal/data integrity.

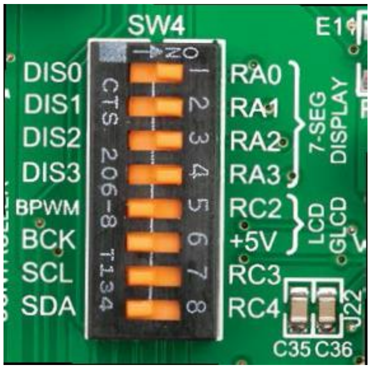


Figure 2 Pins

In order to light up “b” and “c” LEDs to show “1” on the display, we have to send 00000110 through the Port D. “a” is attached to Port D-0 pin, “b” is attached to Port D-1 pin, and so forth (Figure 3).

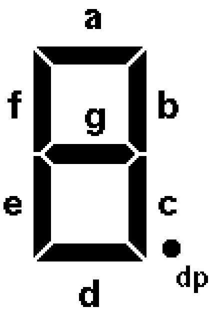


Figure 3 7-Segment Display

If you look at the application board, you will see that there are four seven-segment displays (Figure 4). How will we select the right one? Least significant four of the Port A pins are responsible for selecting one of the displays.

Port A bit 3→ 1^{st} display, named as “DIS3” on the board.
Port A bit 2→ 2^{nd} display, named as “DIS2” on the board.
Port A bit 1→ 3^{rd}display, named as “DIS1” on the board.
Port A bit 0→ 4^{th} display, named as “DIS0” on the board.

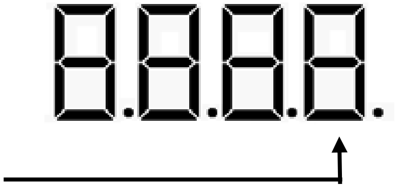


Figure 4 Four 7-Segment Display

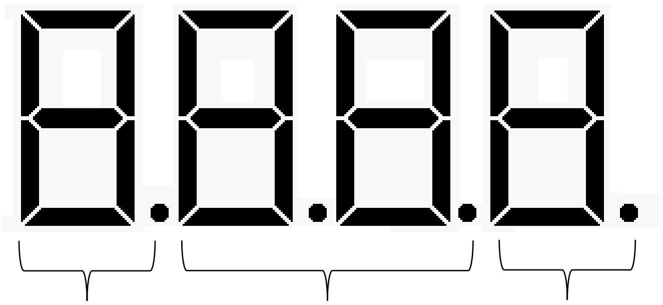
Send “1” to the specified pin of Port A to select the display. The data on Port D will appear by the lighting of “.gfedcba” LEDs.

You are recommended to make a list for hexadecimal corresponding of all numbers (from 0 to 9) in order to display on seven-segment such as 0x3F is for “0”, 0x06 for “1”, 0x5B for “2” and 0x4F for “3”.

Reminder : To let the user enter data in the MPLAB simulator, set a breakpoint to the next line after “ORG 0H”. So when you run the program, it will stop at the beginning of the program. Then you can go to “File Registers” and enter the data to any address. After that, you can run your program step by step, or click “continue” or (press F5) to see the output.

Lab Activities

1. Design a chronometer **in C** consisting of three fields to show minute, second and tenth of a second like shown in Figure 5.



Periods : Minute Second Tenths of a Second

Figure 5 Four 7-Segment Display

It means that;

- Changing period of the left-most 7-segment display is one minute.
- Changing period of middle two 7-segment displays are one second.
- Changing period of right-most 7-segment display is one tenths of a second.

2. Write a scrolling (or moving) text application **in C** which shows your name, surname, student number by moving the letters or numbers four 7-segment.
Check the following web-page for the representation of the characters on 7-segment: https://en.wikipedia.org/wiki/Seven-segment_display_character_representations
Similar examples can be found in the following videos :
https://www.youtube.com/watch?v=I_lnbiihMGg

3. Write a C program for Four 7-segment displays are shown in Figure 6. Consider that two one-byte hex numbers (like AC and BE) are entered via MPLAB’s file register by the user. Before the user enters the numbers, four 7-segment displays should keep showing a shifting

(or moving) text: “Enter your numbers”. The first one-byte hex number needs to be displayed on the first two 7-segments on the left while the second number will be displayed on the last two 7 segments as illustrated in Figure 6.

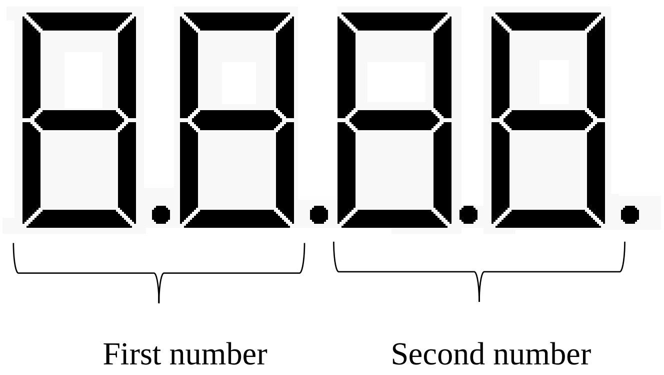


Figure 6 Four 7-segment Display

Next, your code should detect the largest number and subtract the smaller number from the largest.

Before displaying subtraction results on the displays, a shifting text: “Subtraction is completed” needs to be shown on the 7-segments.

Then, multiply these entered two one-byte numbers and display the multiplication result on the displays after a shifting text: “Multiplication is completed” is showed on the 7-segments.

Last, you should design a counter which starts running from subtraction result to multiplication result on four 7-segment display. For example, let’s assume the subtraction result is AA, and the multiplication result is CFAB. Then, your counter starts running from AA to CFAB. After the counter reaches the multiplication result (it is CFAB in this example), the multiplication result (CFAB) should blink five times and a shifting text: “Have fun, goodbye” needs to be shown on the 7-segments.

Note: In each step, texts should be printed by shifting.