CENG 336
ASSIGNMENT III
"Interrupts I"
Rev1.0
Assigned: 6/04/07 & Due: 15/04/07 23:55

Introduction

The purpose of this assignment is to make you familiar with input output operations and basic interrupt features on CengPIC-LAB boards. You will do this by coding imaginary "sms (Short Message Service) typing/displaying" task using push-buttons and LCD Display.

Your Mission

You are expected to simulate a cell phone's sms typing feature by using push buttons and LCD display unit. Unfortunately, you have only 8 buttons. The character and function assignments to these buttons are as follows:

SW00: abcde/ABCDE/12	SW01: efghij/EFGHIJ/3
SW04: klmno/KLMNO/45	SW05: pqrst/PQRST/67
SW08: uvwxyz/UVWXYZ/89	SW09: blank, comma, dot,! ,? , @ / 0
SW12: clear one character	SW13: lower case/Upper case/number

You will be able to type only 16*3=48 characters starting from line 1 up to the end of line 3. The last line of LCD will be used as information line showing available character count in **<xx/48>** form where xx is the available character count, and current typing mode controlled by SW13 in **<abc>**, **<ABC>** or **<123>** form.

- Initially only **<48/48> <abc>** will be shown on the information line and other lines will be all empty (First < character is on the most left position of the line and there is a blank between > and <).
- The message will start from first character of first line.
- SW12 button should clear one character from end of the message if it is pressed less than about one second and the entire message if it is pressed more than about one second.

Example 1: If you want to type **Ceng 336** you will press sw13(Upper Case), sw00-sw00-sw00 ("C"), sw13(Number State), sw13(lower case), sw01 ("e"), sw04-sw04-sw04-sw04 ("n"), sw01-sw01-sw01 ("g"), sw09 (blank), sw13(Upper Case) -sw13 (Number Case), -sw01 ("3"), delay for a while (~1 sec), again -sw01 ("3"), -sw05 ("6"). The information line should be updated continuously while typing and it should be **<40/48> <123>** after typing 6.

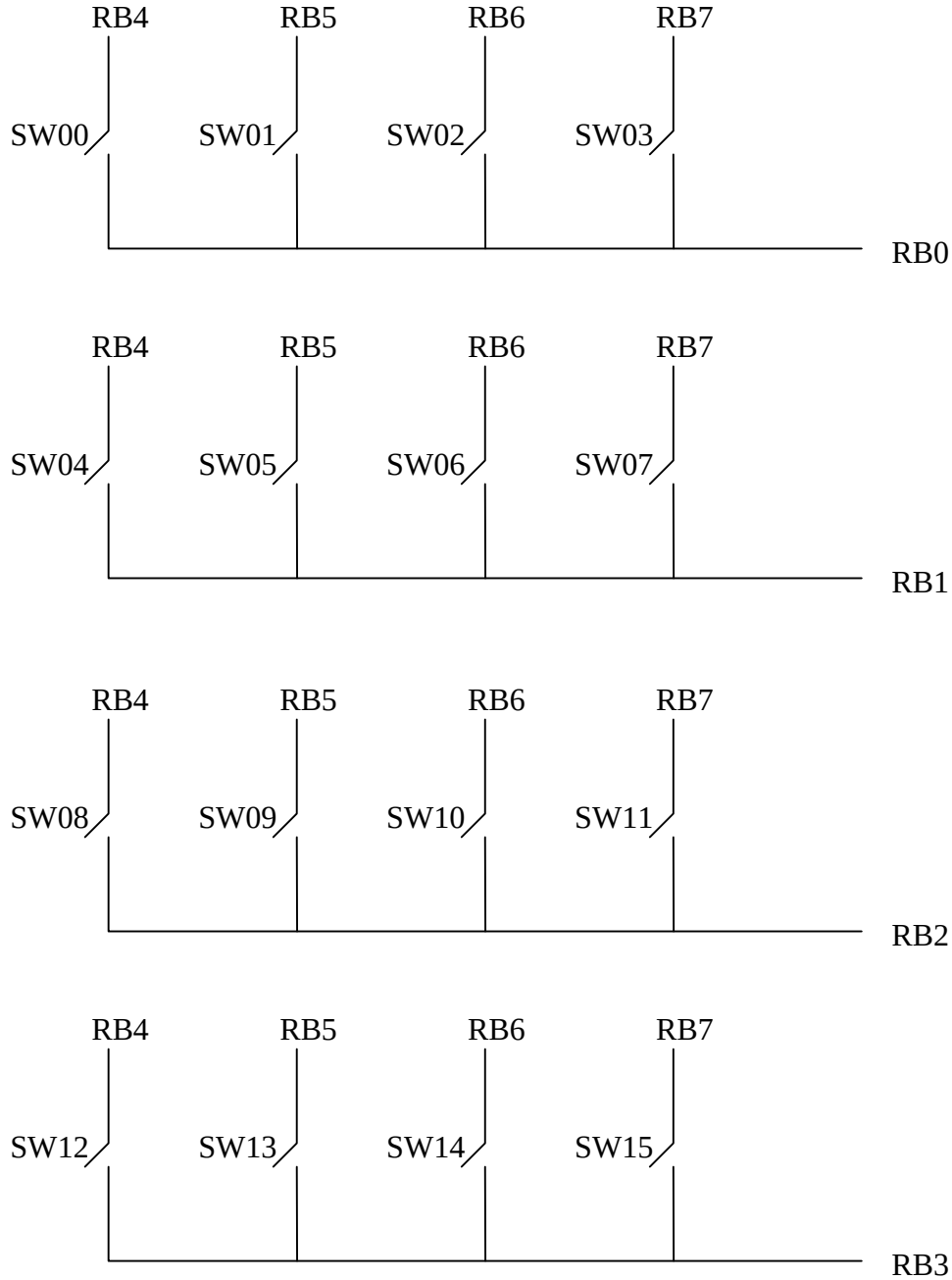
Example 2: If you press sw00, sw01, sw04, sw05, sw08, sw09, sw08, sw09, sw12, sw13, sw01, sw01, delay ~1 sec, sw01, then **aekpu uFE** will be displayed accordingly as in the cell phones. Meanwhile, the last line of LCD will be **<39/48> <ABC>**.

Mini How-to

1. **Push Buttons:** You will use only 8 push buttons (SW00, SW01, SW04, SW05, SW08, SW09, SW12 and SW13), namely left two columns of buttons, and **control**

the buttons only using interrupt on-change feature of PORTB. You will not use RB6 and RB7 pins, hence SW06, SW10, or SW15 etc.

Important note: After setting TRISB to 0xF0 and enabling the pull-up resistors of PORTB, RB6 and RB7 is read as "logic zero" not "logic high". This is due to the hardware design of the board.



2. **LCD:** You will use 4-line, 16-column/character LCD display on CengPIC-Lab boards. You are supplied with LCD library files (`lcd.h` and `lcd.c`) including some basic routines written in SDCC. You are free to make any changes in the files according to your needs. You can use the information given in recitation 7 slides for LCD driving specifications. You may also use any Hitachi HD44780 datasheet for the detailed information about LCD.

The functions in the LCD library are as follows:

`void pulse(void);`
This function generates a pulse on the enable pin of the LCD.

`void lcdInit(void);`
This function initializes the ports needed for LCD and LCD itself.

`void lcdClear(void);`
This function clears the lcd.

`void lcdGoto(unsigned char);`
This function carries the cursor to the given position.

`void lcdWrite(unsigned char);`
This function writes the given character to the LCD.

The character table for the LCD is shown in below figure.

You must include "`lcd.h`" file to your solution as in the given example file, "`lcd_demo.c`".

In order to compile your solution first give

```
Ø sdcc –c –mpic14 –p16f877 lcd.c
```

command to create "`lcd.o`" object file. This is needed only for the first time or after making any change in the "`lcd.h`" and "`lcd.c`" files.

After creating "`lcd.o`" object file give

```
Ø sdcc –mpic14 –p16f877 hw3.c lcd.o
```

command in order to compile your solution and get your "`hw3.hex`" file which you will send to the PIC using WinPIC (for windows) or Odyssey (for linux) as in HW2.

Evaluation

You will **certainly** use SDCC compiler. In addition, you are **NOT** allowed to control the buttons without using interrupt on-change feature of PORTB. Any solution that does not conform to these specifications will **NOT** be accepted.

Total of the lab is worth **100** points.

- **30** points for proper displaying LCD.
- **70** points for push buttons.

Hand In Instructions

- You will submit a single file, "`hw3.tar.gz`", including "`hw3.c`", "`lcd.h`" and "`lcd.c`" files.
- **DO NOT** send email attached your hand in file.
- Make sure you have included your group ID (the number on your card) and the names and IDs of group members in a comment at the top of your "hw3.c" file.
- You will submit "`hw3.tar.gz`" file using http://cow.ceng.metu.edu.tr
- No late submission is allowed.
- **Demo** days will be announced.
- For any clarification please follow the newsgroup.