COMP551 – Interfacing

Fall 2017

Lab 10

Interfacing and Programming LCD's

Students:				
Student ID's:				
Section:	**01**	**02**	**03**	**04**

NOTE: Labs are due at the start of the next lab period. Only submit one lab per group of two students.

Lab 10 – Interfacing and Programming LCD’s

10.1 Introduction:

Most character based LCD’s are based on Hitachi’s HD44780 or HD44580 controller. In this tutorial, we will discuss about character based LCDs, their interfacing with PIC microcontrollers, various interfaces (8-bit/4-bit), programming, special stuff and tricks you can do with these simple looking LCDs which can give a new look to your application.

The HD44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

The **EN** line is called “**Enable**.” This control line is used to tell the LCD that you are sending in data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the “**Register Select**” line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which sould be displayed on the screen. For example, to display the letter “T” on the screen you would set RS high.

The **RW** line is the “**Read/Write**” control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction (“Get LCD status”) is a read command. All others are write commands, so RW will almost always be low.

The data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Finally, the power supply pins for the backlight – LED+ and LED-. Some LCD modules come without the backlight. In that case, these pins are not found or are left disconnected. The recommended voltage for LED+ is 4.2V and LED- should be connected to ground (GND). Vary the value of the resistor connected to LED+ will change the brightness of the backlight. Normally, 220 Ohm or 330 Ohm resistor will be used. You can also connect the pin to PWM output and change the brightness in your software by altering the PWM duty cycle.

10.2 Programming Steps:

Before displaying anything on the LCD, it needs to be configured with proper instructions. The following programming steps explain the procedure of configuring the LCD and display a character on it.

Step 1: Initialize the LCD. The LCD must be initialized using pre-defined commands (see table on page 4) of the LCD.

0x38, configure the LCD for 2-line, 5x7 font and 8-bit operation mode
0x0C, display on and cursor off
0x01, clear display screen
0x06, increment cursor
0x80, set cursor position at first block of the first line of LCD.

Step 2: Send the commands to LCD.

Send the command byte to the port connected to the LCD data pins
RS=0, to select command register of LCD
RW=0, to set the LCD in writing mode
EN=1, a high to low pulse to latch command instruction
Delay of 1ms
EN=0

Step 3: Send data to LCD.

Send data to the port which is connected to the LCD data pins
RS=1, register select to select data register of LCD
RW=0, sets the LCD in writing mode
EN=1, a high to low pulse to latch data
Delay of 1ms
EN=0

Step 4: Display character on LCD.

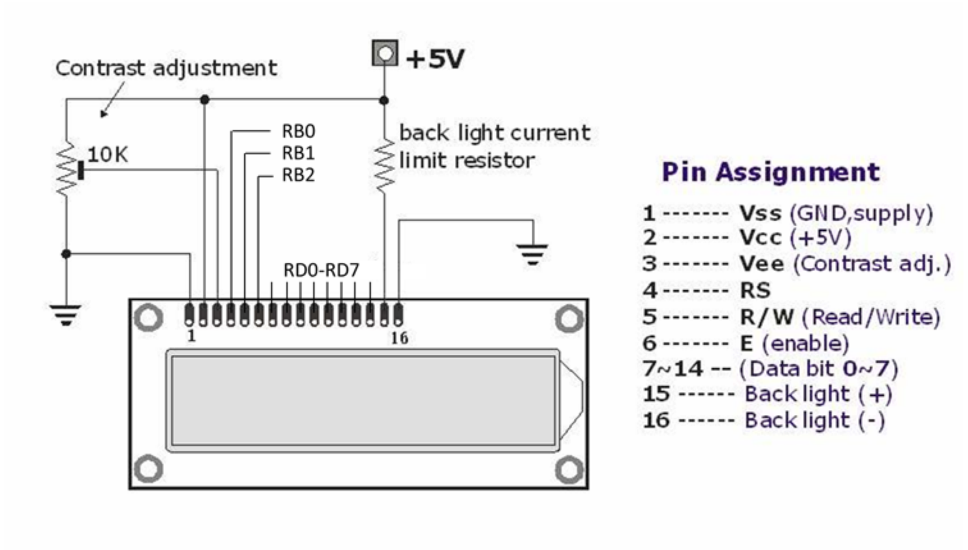
The functions lcdcmd() and lcddata() are user-defined functions. They are used to send a character (E in this case) to be displayed on LCD.

```
lcdcmd(0x38);         // send command 0x38 to LCD
lcddata('E');         // send character E to LCD
```

Exercise:

1. Wire the following LCD circuit on your proto-board, using 8-bit data mode. You will need to solder a header pin to the LCD circuit board so you can plug the LCD into the proto-board. Write a program to continuously display the word '**COMP551**' centred on the top line of the LCD display. Then display the following characters, one at a time; **InterfacIng 2017** centered on the second line of the LCD display, with a .75 second delay between each character. Use one of the Timers to create the .75 second delay. Program the on-board button to stop the count when the button is pressed. Once the letter g is displayed, repeat the cycle.

   Demonstrate the proper operation of the circuit and program to the instructor and submit a hardcopy of your program. Be sure to also include the calculations you made to determine the time delay.



Pin Assignment

1 ------- **Vss** (GND,supply)
2 ------- **Vcc** (+5V)
3 ------- **Vee** (Contrast adj.)
4 ------- **RS**
5 ------- **R/W** (Read/Write)
6 ------- **E** (enable)
7~14 -- (Data bit 0~7)
15 ------ Back light (+)
16 ------ Back light (-)

Command	**Description**
0x01	Clear the display screen
0x02	Return home
0x04	Decrement cursor
0x05	Shift display right
0x06	Increment cursor
0x07	Shift display left
0x08	Display off, Cursor off
0x0a	Display off, Cursor on
0x0c	Display on, Cursor off
0x0e	Display on, Cursor blinking
0x10	Shift cursor position to the left
0x14	Shift cursor position to the right
0x80	Force cursor to beginning of first line
0xc0	Force cursor to beginning of second line
0x38	Using 2 lines, 5x7 matrix and 8-bit mode
0x28	Using 2 lines, 5x7 matrix and 4-bit mode