Lab 3a: LCD Parallel Interface – Direct I/O Control

Revised March 10, 2017
This manual applies to Unit 3, Lab 3a

1　Objectives

1. Identify the PIC32 processor pins used for the LCD interface.
2. Implement LCD control using bit-banging I/O techniques.
3. Understand handshaking requirements for interfacing two systems operating at differing clock speeds.

2　Basic Knowledge

1. How to interpret control flow and data flow diagrams.
2. How to interpret a schematic diagram and electric circuits.
3. How to write a computer program using the C language.
4. How to launch a Microchip MPLAB X project.

3　Equipment List

3.1　Hardware

1. Basys MX3 trainer board
2. Micro USB cable
3. Workstation computer running Windows 10 or higher, MAC OS, or Linux

In addition, we suggest the following instruments:

4. Digilent Analog Discovery 2

3.2　Software

The following programs must be installed on your development workstation:

1. Microchip MPLAB X® v3.35 or higher
2. PLIB Peripheral Library
3. XC32 Cross Compiler
4. WaveForms 2015

4   Project Takeaways

1. How to implement a bi-directional parallel I/O interface.
2. How to use delays to control signal timing.
3. How to manage ASCII text strings.

5   Fundamental Concepts

Microprocessors are the core element in an embedded system. Most embedded applications need resources or capability not available in the microprocessor. The additional capability can be provided by specialty electronic devices. Computer communications is necessary to exchange data between the microprocessor and these devices. The communications interface is provided using serial or parallel inputs and outputs. Parallel I/O implies multiple input/output operations at the same time whereas serial I/O implies single bit data exchanges. Whether the communications interface is serial or parallel, handshaking is required to coordinate the data exchange activity.

6   Problem Statement

Develop and test a collection functions that allow text to be placed at any position on a parallel I/O character LCD following the DFD and CFD models presented in Unit 3. The *LCD_puts* function is required to process the ASCII control characters, which are described in Table 8.1.

7   Background Information

Character Liquid Crystal Displays (LCD) are low-cost electronic devices that are capable of generating visual displays consisting of alphanumeric characters and graphical symbols commonly used for embedded system human machine interfaces (HMI). Many suppliers of LCD devices are based on the Hitachi HD44780 LCD controller, and hence have a common hardware and software interface. Some LCD devices have serial interfaces that use UART, I^{2}C, and SPI communications[1]. These protocols are covered in Unit 4. Other LCD devices use interfaces consisting of four or eight data lines and three control lines[2]. Although the LCDs that use a serial interface require fewer processor I/O pins, the display controller IC is still based on the Hitachi HD44780, which uses 8 or 11 I/O pins. Hence, the serial LCD devices have an on-board microprocessor that translates the serial communications instructions to parallel I/O.

The handshaking required to successfully communicate with the LCD on the Basys MX3 is representative of many common parallel interfaces.

---

[1] PmodCLS: Character LCD with Serial Interface, http://store.digilentinc.com/pmodcls-character-lcd-with-serial-interface/

[2] PmodCLP Character LCD with Parallel Interface, http://store.digilentinc.com/pmodclp-character-lcd-with-parallel-interface/

8   Lab 3a

8.1   Requirements

1. All LCD interface pin control will use *LATx*, *LATxSET*, or *LATxCLR* instructions for outputs and *PORTxBITS* for inputs.
2. Write a library of functions to provide general capability to display strings of text, characters, and numbers at specified positions on the LCD display.
3. All LCD control sequences must meet the strict timing requirements for the following operations:
    a. Initialize the LCD using the recommended sequence of instructions shown in the control flow diagram Fig. B.2. to write specific codes to the LCD at specified intervals.
    b. Position the cursor in the display space (Line 1 or Line 2) where the next character is to be displayed.
    c. Write function(s) that write ASCII control and display data to the Display Data Ram (DDRAM).
    d. Translate the ASCII control characters into one or more C instructions for the LCD, as detailed in Table 8.1.

Table 8.1. LCD controls.

ASCII Character	Hex Code	LCD Code	Action
LF	0x0A	0x01	Line Feed: Clears the line where the cursor is positioned and places the cursor at left position on that line
CR	0x0F	0x02	Return: Places the cursor at left position on the line where the cursor is positioned without clearing display
TAB	0x09		Moves cursor position 4 places to the right. If this action places the cursor beyond the end of Line 1, the cursor will be placed at the start of Line 2.
FF	0x0C	0x01	Form Feed: Clears display and places the cursor at left position on top line

    e. Write functions that write ASCII character data to the Display Data Ram (DDRAM).
        i. Write a function that will write a single ASCII encoded character to the display.
        ii. Write a function that will write a string (array) of ASCII encoded characters to the display.
4. Using the following program, develop a test project that will verify that the function *LCD_puts(byte *str);* correctly displays an array of ASCII text and control characters.

```
char test_str[] = " User generated test string";
int main(void)
{
    hardware_setup(void);           // Initialize the LCD IO pins
    LCD_init();                 // Initialize LCD
    while(1)
    {
        LCD_puts(test_str); // Display the text string
        DelayMs(1000);              // Delay 1 second
        LCDWrite(0x01);     // Clear LCD
    }

}
```

5. Write an application program for the Basys MX3 processor board that achieves the following:
    a. Initializes a loop counter to a value of zero.

b. Displays your name on the first line.

c. Repositions the cursor to the beginning of the second line and displays the text "LAB 3."

d. Repositions the cursor to the eighth character position of the second line.

e. Displays the value of the loop counter.

f. Increments the loop counter.

g. Delays one second before repeating steps d through g.

8.2 Design Phase

1. Generate a project in MPLAB X.
2. Copy the config.bits file to the project directory.
3. Open three new C files: main.c, lcd.c, and swdelay.c.
4. Open two new header files: lcd.h and swdelay.h.
5. Generate the contents of the main.c file for the initialization code and the infinite loop.
6. In swdelay.c, generate a function prototype for the *msDelay(unsigned int dly);* function.
   a. Copy the software delay function from Unit 2, Listing 3 to swdelay.c.
7. In lcd.h, generate function prototypes for the five LCD functions shown in Fig. B.1.
8. Generate the code for the five LCD functions in lcd.c.

8.3 Construction Phase

1. Write a problem statement that lists the requirements.
2. Determine what elements of the Basys MX3 board will be used.
3. List the processor hardware resources that you will be using in this design.
4. Generate a data flow diagram that identifies tasks (functions) and the information that needs to be passed between these tasks.
5. Using the DFD and the multiple CFDs shown in Appendix B, develop and test the functions that will initialize the LCD and display single characters and a string of characters. You should use the software delay function used in Lab 1b.

8.4 Testing

1. Demonstrate that the test program developed for requirement 4 executes correctly using the test string "\0141234\0115678\0119ABC\r".
2. Demonstrate that the test program developed for requirement 5 executes correctly.

9 Questions

1. How would the LCD display be affected if the application described above is combined with an application that uses interrupts?
2. How can the blocking semaphore represented by the Busy Flag be eliminated?
3. What is the maximum rate at which characters can be written to the LCD?

10 References

1. Basys MX3 Data sheet.

2. [PIC32MX330/350/370/430/450/470](#) Family Data Sheet.
3. 2 Line 16 Character LCD Data Sheet, http://www.lcd-modules.com.tw/upload/web/SC/SC1602B.pdf.
4. LCD Data sheet, https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf.
5. ASCII Table, http://www.asciitable.com/.

Appendix A: Basys MX3 Schematic Drawings
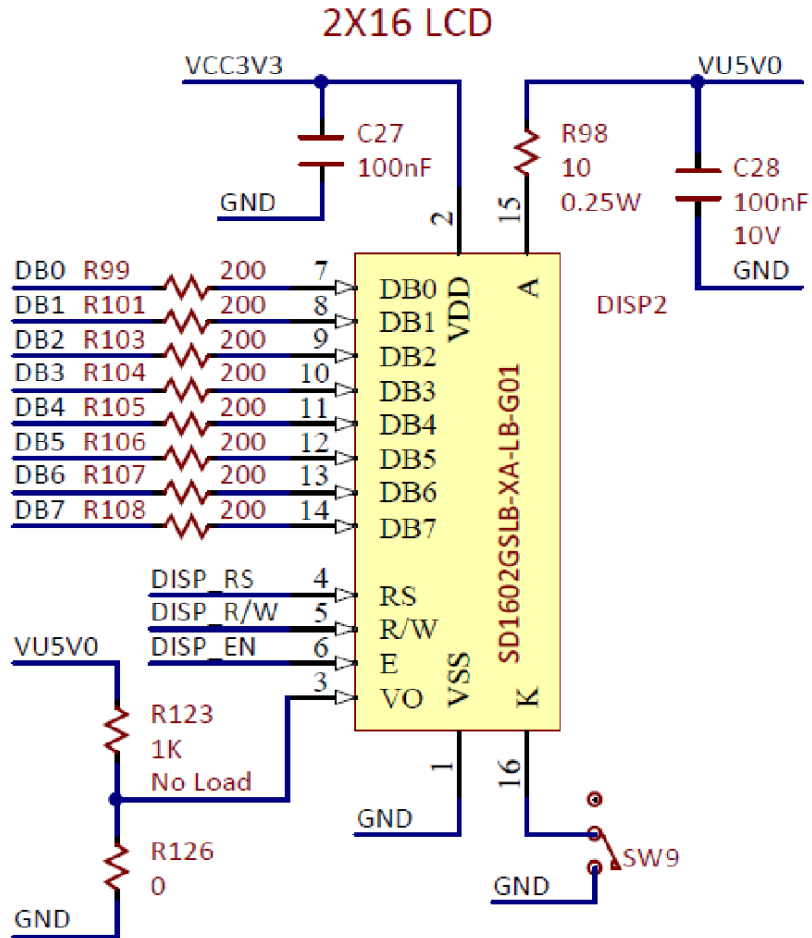
Figure A.1. LCD connection schematic diagram for Labs 3a and 3b.

Appendix B: LCD Code Concept Maps
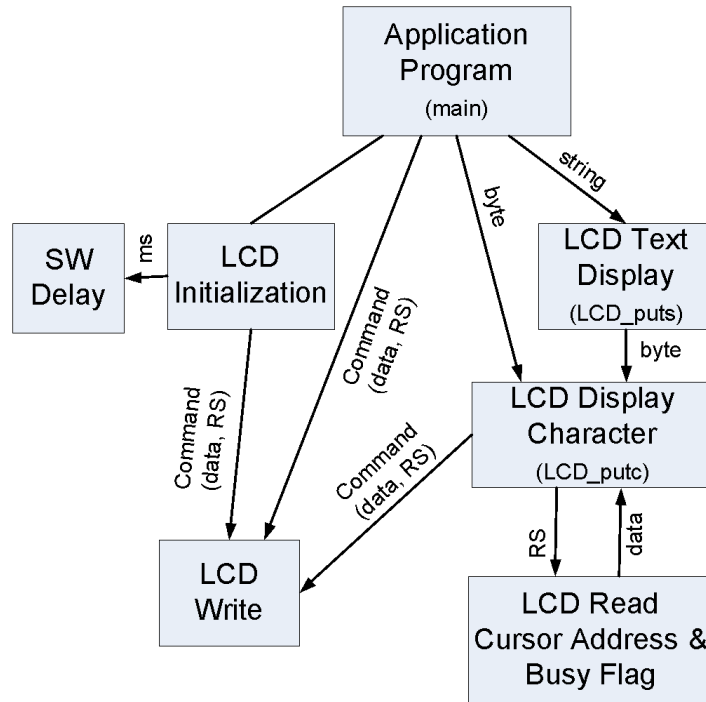
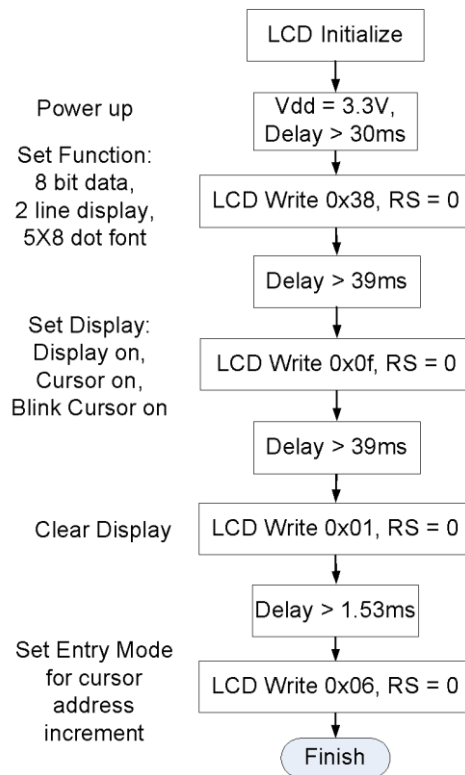*Figure B.1. Data flow diagram for the LCD control.*

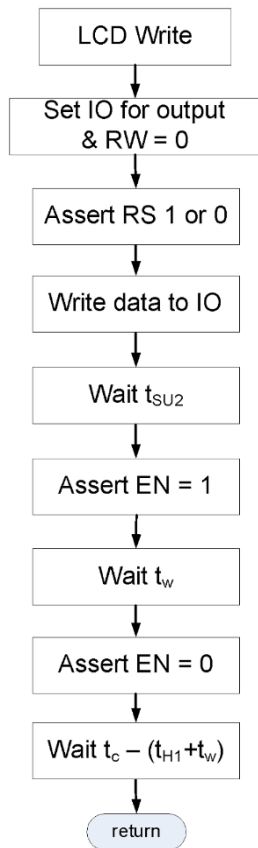*Figure B.2. Control flow diagram for the LCD initialization.*

Figure B.3. Control flow diagram for LCD Write operation using bit-banging.
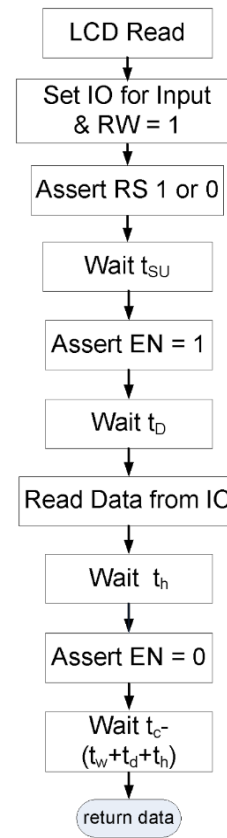

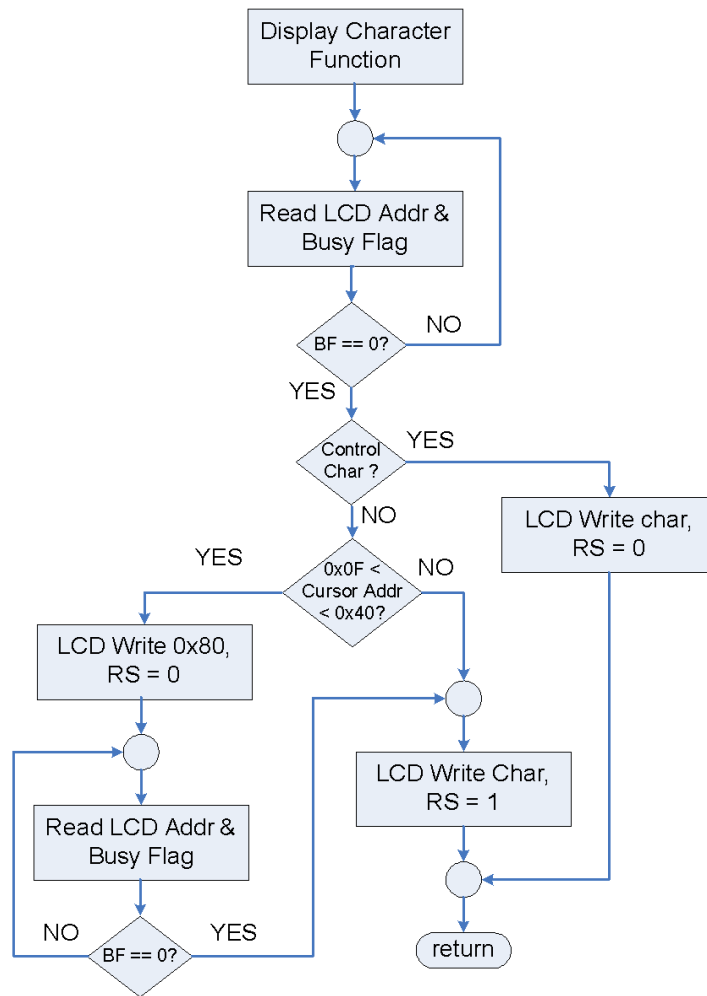*Figure B.4. Control flow diagram for LCD Read operation using bit-banging.*

Figure B.5. LCD display character control flow diagram.