CoE 115 Lab 02

2SAY1617

Objectives

- To construct a LCD interface to the PIC microcontroller
- To implement a simple input-output handling

References Needed

- PIC Instruction
- LCD Dot Matrix User Manual

Overview

Dot-matrix character LCD display modules allow users to output data in a more understandable format. This module has different operating modes which can be set during its initialization phase. Reading and writing data from and to the module is allowed depending on the signals fed to its pins.

Listed below are the signals used by the LCD display module.

- Enable (E) - This is an input signal which will dictate if an operation will start or stop
- Read or Write (R/W) - This is an input signal which will dictate the direction of the operation. This should be set to "0" for writing to the LCD, and "1" for reading from the module.
- Register Select (RS) - This is an input signal which will determine if the input to the LCD corresponds to a command or to a data value. This should be "0" when the input is intended to be a command, and "1" if it's a data.
- Data lines [7:4] - These are input/output lines which corresponds to the upper half of a byte. These are the only lines used when the LCD is operating in 4-bit mode.
- Data lines [3:0] - These are input/output lines which corresponds to the lower half of a byte. These are not used when the LCD is operating in 4-bit mode.

Display

Each 5 pixel x 8 pixel character is controlled by an address in the Display Data RAM (DDRAM) of the LCD module. A value in the DDRAM corresponds to either a predetermined character, or a custom character saved in the Character Generator RAM (CGRAM) of the LCD module. The character table can be seen in Table 5.3 of the uploaded LCD Dot Matrix User Manual in UVLe.

Initialization

Section 2.2.2.2 of the LCD Dot Matrix User Manual shows the sequence on how to initialize the LCD for a 4-bit operation. For CoE 115, we are going to use the following modes of the LCD:

- 4-bit mode
- 2-line display
- 16 character through 40 character
- blinking cursor should be seen

Timing requirements in the sequence and of each instruction should be strictly followed (See Table 3.1 of the LCD Manual for list of instructions and corresponding execution time). Since 4-bit mode will be used, note that the upper half byte (data/instruction[7:4]) of the data or instruction should be sent first before the lower half (data/instruction[3:0]).

Hardware Connections

- Data Lines [7:4] should be connected to RB [11:8]
- Read/Write pin should be connected to 0 or ground
- Register Select should be connected to RB[13]
- Enable should be connected to RB[14]

Seatwork

- Initialize the LCD to the mode detailed in the Initialization section
- After initialization, the LCD should show your surname on the first line and your given name in the second line.
- The display should then swap the two lines for every press in the push button.
- The flow of your code should be something similar to the code below.

```
; after starting

; initialize stack pointer
; initialize GPIO settings

call delay15ms

mov #0x3, W0
call send4ToLCD
call delay4_1ms

call send4ToLCD
call delay100us

call send4ToLCD
mov #0x2, W0
call send4ToLCD

mov #0x28, W0
call send8ToLCD

mov #0x04, W0
call send8ToLCD

mov #0x01, W0
call send8ToLCD

mov #0x06, W0
call send8ToLCD

mov #0x0F, W0
call send8ToLCD

.
.
.

send4ToLCD:
; W0[3:0] are the data bits values
; W0[4] is the RS bit
; set/reset RS bit
; set E bit
; set DB[7:4] bits
; clear E bit

send8ToLCD:
; W0[7:4] are the upper half bits
; W0[3:0] are the lower half bits
; W0[8] is the RS bit
; set/reset RS bit
; set E bit
; send DB[7:4] bits
; reset E bit
; set E bit
; send DB[3:0] bits
; reset E bit
```