LCD Display A

Objective:
Students will design and write a C language program to interface the LCD display. Students will learn how to utilize specific data sheets and our microcontroller's memory mapped addresses to interface code that will send commands to the LCD and display data on the LCD. Modularity is and suggested good programming techniques are strictly enforced.

Program Development Design worksheet:
Read lab instruction completely, and then complete steps 1-3 for your program planning and outline. You will turn your completed design worksheet in as a well-documented word .docx.

Lab 06 description and requirements:
Functions are to be used liberally. One subroutine should initialize the LCD display I/O ports, another should initialize the display itself, another should write the character passed to it as a command to the display, another should write the character passed to it as a data character to the display, another should delay by the count passed, another should accept a character string as an argument and send it to the subroutine that writes data to the display and finally two more to set the three control lines improperly.

The display functions should be in their own file, separate from the main() routine.

The eight display data lines are connected to Port K. These lines will need to be outputs. Data or commands, that are written to the display, must be written to Port K. Characters for display are written as data and everything else is considered a command. The display will be used in the 8-bit mode.

The control lines to the display are connected to bits 5-7 of Port D. Bit 6 of Port D is the read/write (R/W*) line. This line must be low to write to the display and high to read status from the display (we won't read the status in this laboratory). Bit 5 of Port D is the register select (RS) line. It must be high to write characters (displayed characters) to the display and low to write commands (such as the initialization commands) to the display. The third line, Bit 7 of Port D is connected to the E line. This line must be taken high and then low to latch a command or data into the display. Subroutines should be used to handle the setting for these three control lines. The three control lines must be outputs.

<u>Display Initialization</u>
Initialize the display with a blinking cursor by following the sequence below.

Declare a C language array with the following bytes in the initialization function. This function will send the array bytes, one byte at a time, to the display command register. This sequence will initialize the display. Call your Delay() function with a count of 100 between each byte sent to the display.

0x30, 0x30, 0x30, 0x38, 0x0f, 0x01, 0x06.

When in the lab, if the display does not initialize properly, put a break point in the code where the data is written to the display and verify that the proper data is sent out by examining the register containing the data that is getting written to the display. Also verify the state of the control lines.

To write a command put the command on Port K, take the R/W* line low, the RS line low and then the E line high and then low. This sequence should be put in a subroutine.

Delay Function

The function Delay()is passed an unsigned long value. Use a *for or while* loop in the function that loops the number of times specified by the argument passed.

If you test the overall program in the laboratory using Single-step or Debug-Animate mode, turn off the Delay function with the command sequence: Browse-Function List-Double Click on the On to select Off. Otherwise you will wait approximately forever for the delay loops to run. Alternatively, you can, use the *step out* instead of *single step*.

Displaying Characters

Write a C language function to display a string of characters passed as an argument. This function can pass the character to another subroutine that writes a single character to the display. The function should make use of the zero terminator at the end of the display string. And it will receive, as input, the address of the string (ie, a pointer to the string).

At this point the messages will be 16 characters or less.

The call from main to output a character string will be as follows:

YourSubroutine("My 1st Message");

Of course use a more relevant name for the subroutine to display the string.

Program Structure

Function main should be in the main.C file by itself. The functions to "initialize" and "write strings" to the LCD display should be in another .C file.

A header definitions can be included if you define constants such as the bits. A prototype file must be included and contain all function prototypes.

Demonstrate: your working program to the Instructor or TA.

Before you submit: ensure each of your files includes a commented header, a brief description of the program or file purpose and well organized and commented code.

Submission: Submit your completed well organized design worksheet, main.c, LCD.c defs.h, and protos.h files as well as a zipped version of your entire project to Canvas assignments for Lab 06