

Lab 6 Deliverables

ECE341, University of Idaho

Spring 2015

The following specifications and clarifications apply to the deliverables for Lab 6. Unless otherwise stated, the following should be included in addition to the standard requirements for the lab deliverables.

Demonstration

All LCD-related code *must* be contained within two files: `LCDlib.h` and `LCDlib.c`. The first file, `LCDlib.h` should only contain LCD-specific `#define` statements and function prototypes for the function definitions in `LCDlib.c`. `LCDlib.c` should only `#include` `LCDlib.h`, `plib.h`, and `Cerebot32MX7cK.h`¹ and implement (at a minimum) the following library functions:

- `void initLCD(); //Initialize the PMP and LCD`
- `void LCD_putc(char c); //Write an ASCII character`
- `void LCD_puts(char * str); //Write a null-terminated string`
- `void LCD_delay(unsigned ms); //Delay for LCD timing (Copy the hardware-assisted delay code)`

The function prototypes listed above must match your function definitions exactly; this is the library interface. The implementation of the interface is your choice however it will be helpful to define several helper functions (below). These functions aren't part of the interface definition so you are free to choose the function return value, name, and parameters to suit your particular design.

- `int busyLCD(); //Return the LCD busy flag`
- `void writeLCD(int addr, char c); //Write any byte`
- `char readLCD(int addr); //Read a byte`

Furthermore, you are free to develop additional functions for your library that you find useful (such as clearing the display, or writing a string to a specific line). You will use the library functions in your other source file (typically named `Lab6.c` or `Project6.c`) to display the two lines of text (using `LCD_puts`) specified in the "Project Tasks" section of the lab handout in the `while(1)` loop (use `LCD_delay` for the delay).

¹Only if necessary

Report

- In the background section:
 - Give a brief overview of the PMP peripheral. What sort of devices is the PMP peripheral intended for? Give a couple specific examples (other than an LCD).
 - Why do we interface with the LCD via a controller chip? What purpose does the controller chip serve?
- In the implementation section:
 - Include CFDs for each function in `LCDlib.c`². The DFD, however, is not required for this report.
 - Justify your choice for the number of wait states (beginning, middle and end of a transmission) for the handshaking signals.
- In the testing and verification section:
 - Prove that your handshaking signals satisfy the timing constraints specified in the PmodCLP reference manual (pg 3). Copy the table below and enter the measured values along with a reference to the oscilloscope capture(s) used for the measurement.

Parameter	Symbol	Min	Max	Measurement	Reference
Enable cycle time	tc	500 ns			
Enable high pulse width	tw	220 ns			
RS, R/W setup time	tsu	40 ns			
RS, R/W hold time	th	10 ns			

- In the conclusion:
 - What is the maximum rate at which the LCD can receive characters? Using this rate, how much time would it take to completely erase and rewrite the visible display area?
 - Is using the PMP peripheral more efficient than bit-banging the LCD? Why or why not?

²Take advantage of hierarchy!