

# ECE 3710 LAB V

---

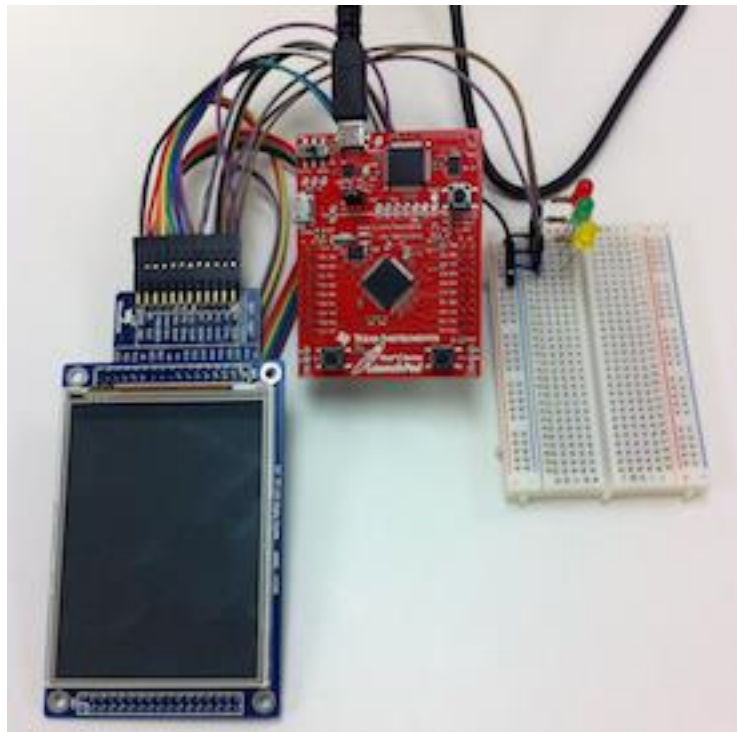
Due Date: Week of November 16<sup>th</sup>, 2015 (20 Points)

## Objectives

The student should be able to interface with the LCD touch panel for input and output in the C programming language.

## Overview

In this lab you will practice programming the microcontroller using C. You will be using the LCD touch panel as a 3-button array to turn on or off corresponding LEDs on your breadboard.



## Preparation: week one

- Look over the ILI9341 datasheet. This is the controller for the LCD display. Here you will find descriptions of the display controller, its command indexes, the display RAM addresses, display on/off sequences, etc.
- Look over the ER-TFTM032 Data sheet. This is the datasheet for the whole LCD module, and it includes valuable information about the pins.
- Review the lecture notes about 8-bit Parallel and LCD interfacing. Remember 8 bit Parallel was utilized in Lab 0, review that if necessary.

- Download the LCD project code (LCD.zip) from the wiki.
- Read about PLL. That is, phase-locked loop, not the hit TV show, Pretty Little Liars.

## Hardware

- Tiva™ C Series TM4C123G LaunchPad Evaluation Board
- ILI9341 / XPT2046 LCD touch screen
- 3 LEDs (1 red, 1 green, 1 yellow)
- Three 220 ohm resistors or 220 ohm resistor bus
- Jumper wires

## Requirements

1. Three buttons must be drawn on the LCD. One red, one green, and one yellow. The buttons will be blank colored outlines initially (like the red and yellow button in figure 1).
2. When a button is touched the button will be filled (like the green button in figure 1). When that button is touched again the button should be return to a blank outline.
3. The red, green, and yellow LEDs should be wired to three available ports for output. The LEDs must be lit up with the corresponding LCD button. If the LCD button is solid, the LED should be on and vice versa.
4. The screen refresh rate must be acceptably fast. What is the maximum clock rate that the ILI9341 controller can handle? PLL is one method to speed up the clock rate, which will result in faster refresh rates.

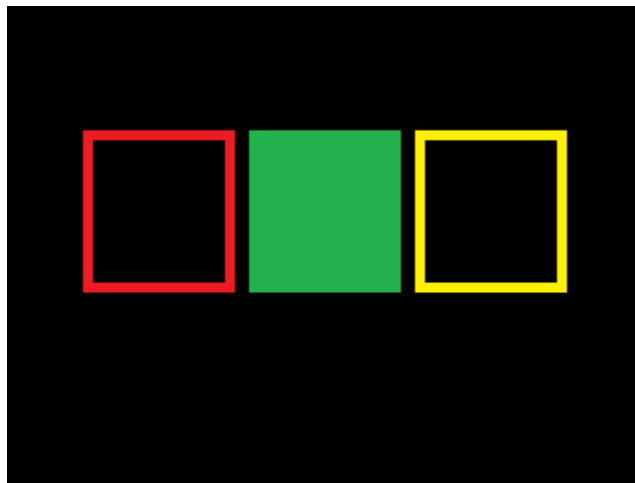


Figure 1: LCD Button Example

## Verify Requirements

Demonstrate to your TA that all buttons work correctly as described in the requirements section. Touching outside the buttons must have no effect on the buttons or LEDs. Debouncing must be properly implemented. Touching and holding a button must not toggle the button state until the button is

released. You must discuss in your report how you handled various touch scenarios including touches outside the button boundaries, partly-in-partly-out touches, touch and holds, debouncing, etc.

## Week One

This is a two-week lab. At the end of the first week you should 1) have a complete interface (software and hardware) implemented for the LCD, and 2) be able to use said interface to draw the filled/unfilled boxes.

We have provided a skeleton project file that includes a number of function prototypes that we feel are useful; you are free to use our function definitions or write your own. The project file (LCD.zip) should work with your board without further modification.

For the LCD Screen, you must use 8-bit Parallel. This will allow you to use SPI for the touchscreen portion, without having to mux the SPI.

For a description of the Pins, see section 4 of the ER\_TFTM032-3 data sheet. An example of how to wire the screen can be seen in Figure 2. The capacitors will not be needed for this project

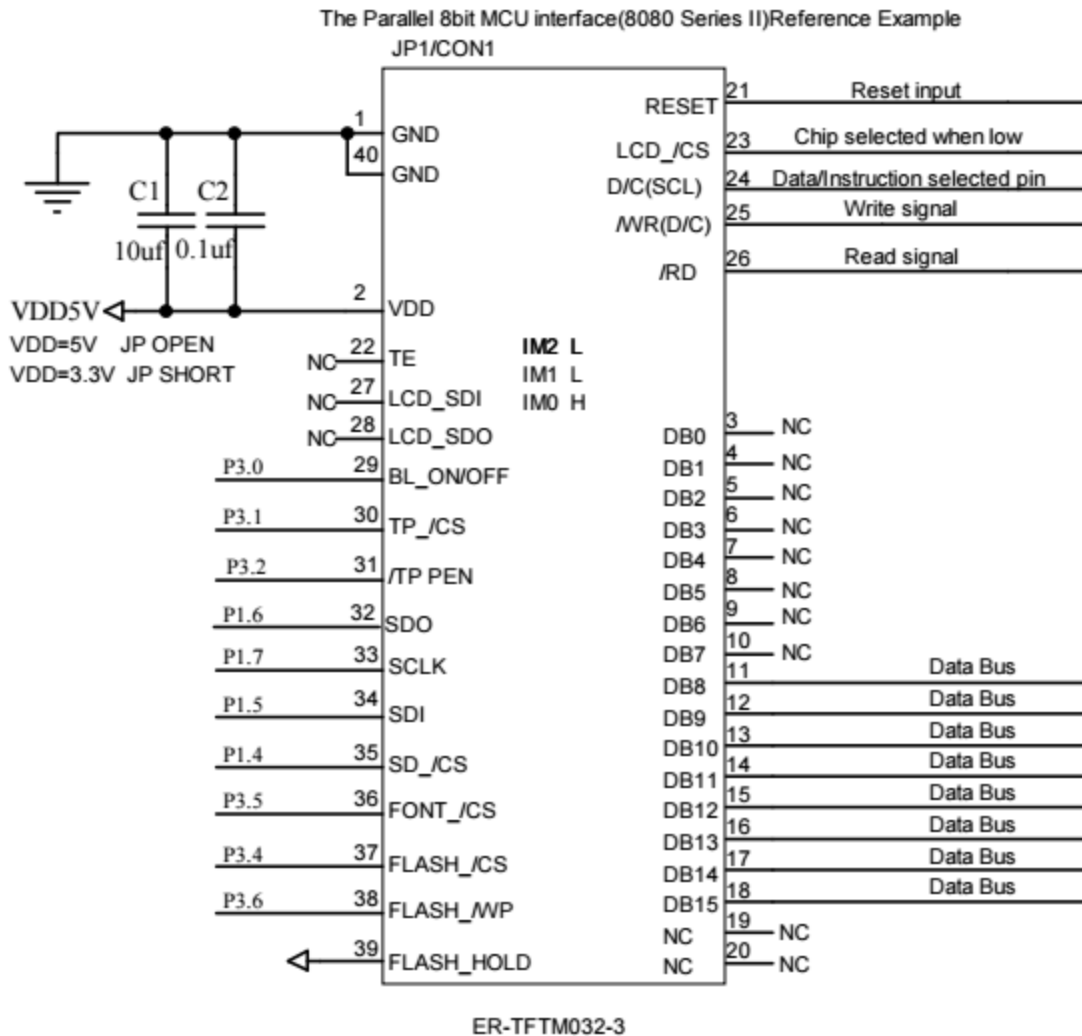


Figure 2 Screen Pins

We provided a binary file of functioning LCD communication. The workingLCD.h will link to a library file which will flash your screen red. All you have to do is write the commented out “setup” functions in main. The code will work if DB8-11 are set up to be PC4-7, and DB12-15 are set up to be PE0-3. /WR and /RD should be PA6 and PA7 respectively. PB6 is TB\_CS (Touch Screen Chip select), PB4 is the LCD\_CS and PB1 is D/C. **This is optional, you may configure your own ports and write your own test.** This is just to check you wiring and help you understand the LCD screen.

Next week's phase of this lab will be discussed in detail after this week.

## Notes and Tips: week one

- You will want to consult chapter seven of the LCD datasheet to ensure that your uC provides the correct timing. Without doing this, your code will not work. Chapter seven also gives a graphical representation of the state of the pins during a write cycle that you may find useful.

- The 5V pins must have a 5V source or the LCD screen will not light up.
- Use the logic analyzer to help you debug your code.

# ECE 3710 LAB V WEEK II

---

Due Date: Week of November 16<sup>th</sup>, 2015 (20 Points)

## Preparation: week two

- Finalize last week's lab if you have not done so. Note the dimensions of your buttons displayed on the LCD.
- Read V1.Ch8.3 & V2.Ch7.5 of your textbook.
- Read the Wikipedia article on Serial Peripheral Interface (SPI). Note SPI is referred to as Synchronous Serial Interface (SSI) in TM4C123GH6PM data sheet.
- Skim through the section on SSI, which is section 15 (Page.949), of the TM4C123GH6PM data sheet. Pay careful attention to section 15.3.4 (Frame Format); section 15.4 (SSI Initialization and Configuration); section 15.5 and 15.6 (SSI Register Map and Register Description).
- Look over the XPT2046 datasheet. This is the controller for the resistive touch screen. You might want to read it a few times to make sense out of it.

## Week Two

Last week you were given code that initialized the LCD display controller. This week, you are responsible for proper setup and initialization of the touch panel controller from scratch. Be sure to look over the appropriate datasheets to figure out how to do this. Upon successfully initializing the touch panel controller, you will then be required to write function that will read data from and write commands to it. Come up with a way to test these functions out before diving into the full lab assignment. Once you have your touch panel working properly, use your code for the LCD display interface and combine these to create a program that accomplishes the requirements.

## Notes and Tips: week two

- The Tiva™ C Series TM4C123G LaunchPad Evaluation Board has four SSI modules, select the appropriate GPIO pins to configure as alternate functions for SSI module.
- Read through the XPT2046 document a few times to get a clear understanding of how to communicate with the module.
- The XPT2046 uses the Touch Panel Interrupt Request (TP \_\_ PEN) pin to inform the user that it is receiving data. It is recommended to use an interrupt to handle this request.
- All the resources mentioned in the "Preparations" section of this document will be very useful.
- Debouncing in this case generally refers to small finger movements unnoticed by humans but picked up by the touch panel. Consider getting multiple x, y positions from the touch panel for one touch gesture and average them.

- For more accurate results, consider writing a calibration function by prompting the user to touch multiple known points. Store the prompted coordinates and the actual user's touch coordinates in a matrix to be referenced for all future touches. This is not required.
- Focus on basic functionality first then refine your program later. In engineering, “good enough” is often desirable/cheaper.