Lab Report 6

Nathan Tipton, Partner: Tarrin Rasmussen

November 18,2016

Contents

1 Objectives

The purpose of this lab is to interface with LCD screen with touch panel using C programming language.

2 Overview

For this lab, we will be writing a C program to use the LCD touch panel as a 3-button controller for corresponding LEDS on the breadboard.

2.1 Requirements

1. Three buttons must be drawn on the LCD. One red, one green, and one yellow.
2. When a button is touched the button will be filled. When that button is touched again the button should be returned 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 corresponing 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.

3 Procedure

3.1 Wiring

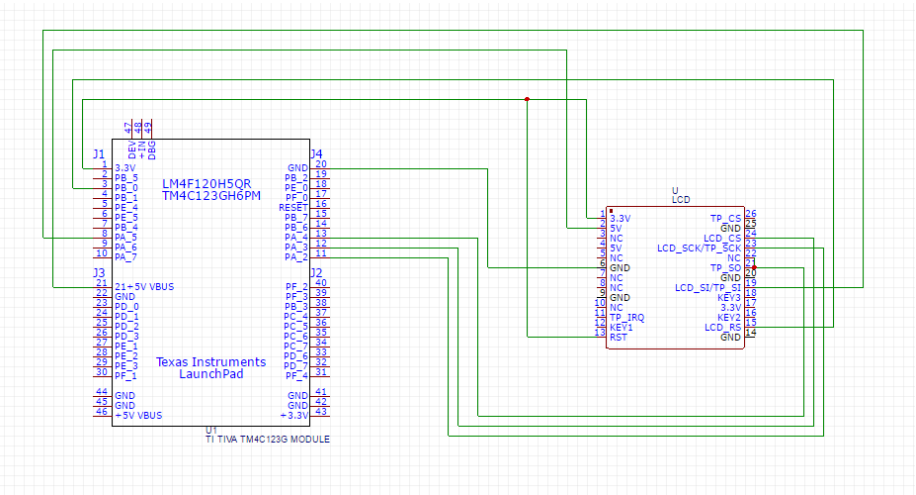

Figure 1: Schematic

3.2 SSI

The microcontroller will use Synchronous Serial Interface (SSI) to communicate with the LCD touch panel. SSI is a synchronous protocol. The clock signal is included when interfacing between devices. SSI uses a master to create the clock and slaves use the clock to transfer data. The following steps are used to configure SSI:

1. Enable the clock to the SSI module.
2. Enable the clock to the GPIO module being used.
3. Enable Alternate function for the GPIO pins. PA[2:5]
4. Enable Digital function for the pins.
5. Disable SSI module by clearing SSE bit.
6. Using the mux control register enable the pins for SSI functionality.
7. Initialize the microcontroller for SSI master.
8. Set the SSI clock source.
9. Configure clock divisor to be used for SSI module.
10. Choose protocol mode. Freescale SPI
11. Set data size.
12. Enable SSI module.

4 Lab Difficulties

We had problem after problem with this lab.

4.1 Wiring

At the beginning of the lab we had considerable difficulty determining how the LCD should be wired to the microcontroller. We were able to eventually solve this problem when more people had LCDs to work with and the TA was able to guide us. We also had a small mishap where the jumper leads broke leaving pieces in the headers of the LCD and Microcontroller.

4.2 Header file

We created our own header file at the start of the lab. We had difficulties using the provided header file due to the instructors code being unavailable. After creating our own custom.h file the LCD was still not flashing red. We eventually were told we were incorrectly using the header file. We were doing things that C++ allows but not C. Our inexperience with C has caused many problems.

4.3 Data Transfer

After trying many different solutions we were able to get the screen to flash red with the provided code. However, once we tried using our own writedat and writecmd commands the screen produced unexpected results. Our code only changed part of the screen. There was a bar on the left of the screen as seen in Fig. 2. No matter what we changed, we were unable to change this section of the screen.

5 Conclusion

This lab was the hardest for us this semester. We were unable to meet the requirements. Nothing seemed to work like it was supposed to. Given more time on this lab I am confident we could work out the bugs and display the required squares. I would do things differently if I started this lab again. First, I would try and use the header file provided in order to prevent small mistakes that are difficult to debug in a custom header file. Second, I would refer to the book more. I found a great deal of useful information on SSI near the end. We tried using the Lecture notes and had difficulty following them in order to configure our SSI and understand what was happening. However, we must have missed something due to our data transfer problem.

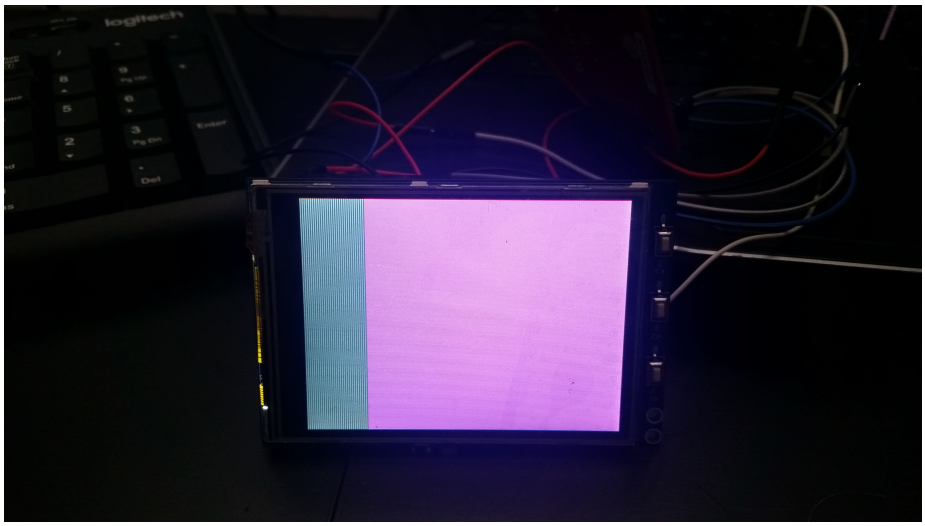

Figure 2: LCD problem

6 Appendix

6.0.1 Code

```
// This function draws three empty boxes. The user touches the box
// to fill the box and turn on a corresponding external light.


#include "LCD.H"
#include "Custom.h"
unsigned char *PA = (unsigned char *) 0x40004000;
unsigned char *PB = (unsigned char *) 0x40005000;
unsigned char *SYSCTL = (unsigned char *) 0x400FE000;
unsigned char *SSI0 = (unsigned char *) 0x40008000;
volatile unsigned int SSI0_SR __attribute__((at(0x4000800C)));



int main(void)
{
        setup_clk();
        setup_SSIO_0(); //setup SPI on PA2-PA5
        setup_PB(); //Setup D/C on PB0

        LCD_Init();
// while (1)
// {
// flash_screen(red);
// }
        tsetArea(0, 239, 0, 319);
        twriteColor(yellow);


    while (1)
```

```
    {

    }
}


#include "Custom.h"



void setup_clk(){
 tSYSCTL->RCC2 |= 0x800;
    tSYSCTL->RCC = (tSYSCTL->RCC &~0x7C0) + 0x540;
    tSYSCTL->RCC2 &= ~0x70;
    tSYSCTL->RCC2 &= ~0x2000;

    tSYSCTL->RCC2 = (tSYSCTL->RCC2 & ~0x1FC00000) + (4<<23);



        __nop();
        __nop();
        __nop();
    tSYSCTL->RCC2 &= ~0x800;
tSYSCTL->RCGCSSI |= 0x1;
tSYSCTL->RCGCGPIO |= 0x3;
                __nop();
        __nop();
        __nop();



};
void setup_PB(){
tGPIOB->DIR |= 0x1;
tGPIOB->AFSEL |= 0;
tGPIOB->PUR |= 0x1;
tGPIOB->DEN |= 0x1;



};
void setup_SSIO_0(){
tGPIOA->DIR |= 0x2C;
tGPIOA->AFSEL |=0x2C;
tGPIOA->DEN |= 0x2C;
        tSSI0->CR1 &= 0xFFFFFFFD;
        tSSI0->CR1 &= 0xFFFFFFFB;
        tSSI0->CPSR = 0x02;
        tSSI0->CR0 = 0x7;
        tSSI0->CR1 |= 0x2;




};
void twriteCmd(unsigned char CMD){
while((tSSI0->SR & 0x11) != 0x01){};
        tGPIOB->DATA =0; //command
        tSSI0->DR=CMD;
        while((tSSI0->SR & 0x4)==0){}
```

```
};
void twriteDat(unsigned char DAT){
while((tSSI0->SR & 0x01) == 0){};
        tGPIOB->DATA=1;
        tSSI0->DR=DAT;
        while((tSSI0->SR & 0x4)==0){}



};
void twriteDat2(unsigned short DAT){
        unsigned short DAT2;
        DAT2=DAT>>8;

                        tGPIOB->DATA=1;
        while((tSSI0->SR & 0x00000001) == 0){};
                tSSI0->DR=DAT;
                while((tSSI0->SR & 0x00000004)==0){}
                        tGPIOB->DATA=1;
                while((tSSI0->SR & 0x000000000951) == 0){};

                tSSI0->DR=DAT2;
                        while((tSSI0->SR & 0x4)==0){}







};
void twriteDat4(unsigned int DAT){



        tGPIOB->DATA = 0x1; // data is 1
        while((tSSI0->SR & 0x01) == 0){};
        tSSI0->DR = DAT >> 24;
        while((tSSI0->SR & 0x2) == 0x0){};
        tSSI0->DR = DAT >> 16;
        while((tSSI0->SR & 0x2) == 0x0){};
        tSSI0->DR = DAT >> 8;
        while((tSSI0->SR & 0x2) == 0x0){};
        tSSI0->DR = DAT;



};


void tsetArea(unsigned short x1, unsigned short x2, unsigned short y1, unsigned short y2){
 // Column 0x2A

        twriteCmd(0x2A);
        twriteDat2(x1);
        twriteDat2(x2);

        // Page 0x2B

        twriteCmd(0x2B);
        twriteDat2(y1);
        twriteDat2(y2);
```

```
};
void twriteColor(unsigned short color){
int i;
        int column;
        int row;
        column = 240;
        row = 320;
        twriteCmd(0x2c);
        for(i=0;i<column*row;i++){
                twriteDat2(color);
        }
};
```