

CPE 187L Embedded Systems Design

Praveen Meduri

Section #2

Lab 4 Report

Xavier Howell

Introduction

The purpose of this lab is to go through a simple debug process. The program and code is already setup so we will need to figure out why our expected output is not correct. Below is the table of expected outcomes when we press switches on the board.

Switch Input	LED Output
Both switches SW1 and SW2 are pressed	The LED should be blue
Just SW1 switch is pressed	The LED should be red
Just SW2 switch is pressed	The LED should be green
Neither SW1 or SW2 is pressed	The LED should be off

Table 2.1. Specifications for Lab 4. SW1 is on PF4 and SW2 is on PF0.

Debug In Simulation

Switch Input	Desired LED Output	Actual LED Output
Both are pressed	Blue	Off
Just SW1	Red	Green
Just SW2	Green	Red
Neither is pressed	Off	Blue

The results show that our SW1 and SW2 are mixed up. This may also be causing the both and neither inputs to be backwards. My hypothesis is that the switches are declared and initialized incorrectly or the output values are signed improperly.

After reviewing the main code it seems the switches and ports are initialized correctly. The color code for the LEDs are giving at the bottom, but its possible this could be wrong. For a simple fix I recorded the color outputs for each condition and reassigned them to the correct condition.

BLUE = 0x04 → 0x00

RED = 0x02 → 0x08

GREEN = 0x08 → 0x02

Off = 0x00 → 0x04

The screenshot displays the uVision IDE interface for a Texas Instruments lab 4. The main window shows the C code for the program, which initializes port F pins for input and output. The code is in C, but the disassembly view shows the assembly instructions. The code is as follows:

```

41 unsigned long Out; // outputs to PF3,PF2,PF1 (multicolor LED)
42
43 // Function Prototypes
44 void PortF_Init(void);
45 void Delay(void);
46 void EnableInterrupts(void);
47
48
49 // 3. Subroutines Section
50 // MAIN: Mandatory for a C Program to be executable
51 int main(void){
52     TExa5_Init(SW_PIN,PF4,LED_PIN,PF32);
53     // TExa5_Init initializes the Real board grader for lab 4
54     PortF_Init(); // Call initialization of port PF4, PF3, PF2, PF1, PF0
55     EnableInterrupts(); // The grader uses interrupts
56     while(1){
57         SW1 = GPIO_PORTF_DATA_R40x10; // read PF4 into SW1
58         SW2 = GPIO_PORTF_DATA_R40x01; // read PF0 into SW2
59         if((SW1&SW2)){ // both pressed
60             GPIO_PORTF_DATA_R = 0x00; // LED is blue
61         } else{
62             if(SW1&(!SW2)){ // just SW1 pressed
63                 GPIO_PORTF_DATA_R = 0x08; // LED is red
64             } else{
65                 if(!SW1&SW2){ // just SW2 pressed
66                     GPIO_PORTF_DATA_R = 0x02; // LED is green
67                 } else{
68                     GPIO_PORTF_DATA_R = 0x04; // LED is off
69                 }
70             }
71         }
72     }
73 }
74 // Subroutine to initialize port F pins for input and output
75 // PF4 and PF0 are input SW1 and SW2 respectively
76 // PF3,PF2,PF1 are outputs to the LED

```

The screenshot also shows the 'Port F Hardware' window with a circuit diagram of the LED and switches. The 'Port F Registers' window shows the current state of the registers: DATA: 0x11, DIR: 0x0E, DEN: 0x1F, PUR: 0x11, POR: 0x00, RCGC2: 0x00000028, LOCK: 0x00, CR: 0x1F. The 'Grading Controls' window shows a score of 100. The 'Watch' window shows the values of SW1 (0x00000010) and SW2 (0x00000001). The 'Command' window shows the output of the test: 'Both switches released test: - LED is off, Test6 PASSED Done grading. Score is 100'.

Debug On The Real Board

The code was already corrected on the simulation so to run it on the real board all we need to do switch the options to debug on the hardware, build the target again, flash it to the board, and run the debugger.

The screenshot displays the Keil uVision4 IDE interface. The main window shows the disassembly of the `main` function, which includes comments and assembly instructions. The `Registers` window on the left shows the state of the Cortex-M3 registers. The `Properties` window on the right shows the configuration for the target device, including the mode, course, lab, grade, and number of LEDs. The `Command` window at the bottom shows the command line, and the `Watch` window shows the values of the SW1 and SW2 variables.

Registers

Register	Value
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000710
R14 (LR)	0x00000000
R15 (PC)	0x0000026c
PCSRN	0x00000000

Disassembly

```
277:      B      _main
278:
279:      ; *****
280:      ;
281:      ; This is the code that gets called when the processor receives a NMI. This
282:      ;
283:      ;
284:      ;
285:      ; *****
286:      NMI_Handler PROC
287:      EXPORT NMI_Handler
288:      B      .
289:      ENDP
290:
291:      ; *****
292:      ;
293:      ; This is the code that gets called when the processor receives a fault
294:      ; interrupt. This simply enters an infinite loop, preserving the system state
295:      ; for examination by a debugger.
296:      ;
297:      ; *****
298:      HardFault_Handler PROC
299:      EXPORT HardFault_Handler
300:      B      .
301:      ENDP
```

Properties

Property	Value
Mode	Done
Course	UT6.03x
Lab	4
Grade	100
NumFromEdX	0
CopyThisToEdX	IplimaGim
ActionMsg	Release both switches
IntroMsg	Checking both released
OKMsg	Grader sees LED off
ErrMsg	

Ports

Port	Value
InputPort	Inputs are PF4 and PF0
OutputPort	Outputs are PF3, PF2, and PF1

Mode bits

Mode bits	Grading	Done	Error
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Voltmeter

Voltage	Value
	0.306 volts

Command

```
WS 1, 'SW1
WS 1, 'SW2
```

Watch

Name	Value	Type
SW1	0x00000010	unsigned long
SW2	0x00000001	unsigned long

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display

Questions

0) Download and open the data sheet for the TM4C123 microcontroller

1) Look at the Tiva TM4C123GH6PM Microcontroller High Level Block Diagram to see how much SRAM and Flash ROM are available. (10 Points)

The Microcontroller has 32KB of SRAM and 256 KB of Flash ROM.

2) Lookup in the relevant chapter to see how many GPIO (General Purpose Input/Output) pins are supported. (10 Points)

There are 43 GPIO pins.