Lab 3 Reaction Timer

Wadhwani Electronics Lab
Compiled by: Devdatta and Jishnu

Department of Electrical Engineering
Indian Institute of Technology Bombay
February 1, 2016

Problem Statement

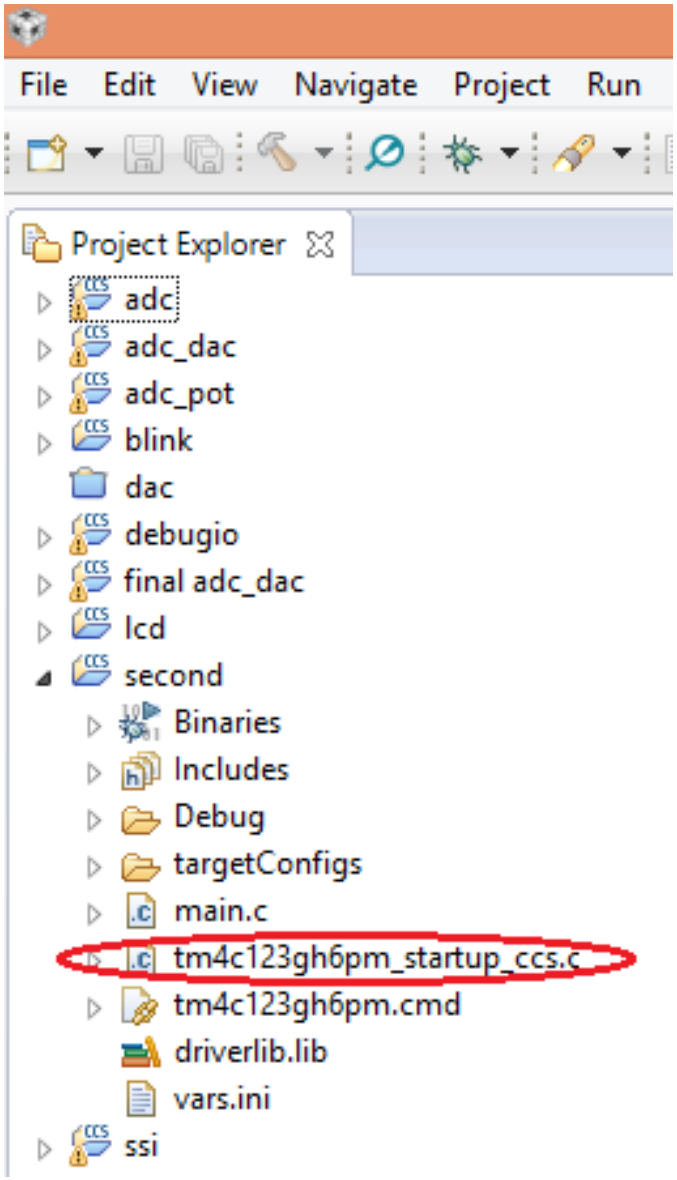
1. The program starts with a message on LCD as Press switch SW1 on the first line and as LED glows on the second line.Turn on a LED.
2. The user presses the switch. The LED is turned OFF.
3. The program is expected to measure time between the instant the LED starts glowing and the instant the switch press is identified.
4. The display on the LCD shows Attempt no # on the first line and the time in ###ms on the second line. This is message is displayed for 30 seconds.
5. The program then goes back to step 1 again and repeats for five user attempts.
6. After five attempts, display Now press SW2 on the first line and for average response time on the second.
7. After five attempts, display Now press SW2 on the first line and for average response time on the second.
8. Note that we need to account for 10ms delay for switch debouncing.

Instructions: 1

- Initialize the LCD. Find detailed procedure in the attached pdf(Tips for Interfacing LCD). And also refer Peripheral Driver Library Guide for System functions(Chapter 27) and Timer functions(Chapter 30).
- Set up the Timer in continuous(periodic) mode. Load in a value so that it takes 1 millisecond. Make sure you enable the timer via *SysCtlPeriphEnable()* and also the Timer interrupt. Do NOT run the timer yet.
  Functions: *TimerConfigure();TimerLoadSet();IntEnable();TimerIntEnable();*
- Change the corresponding Timer interrupt handler in **tm4c123gh6pm_startup_ccs.c** to your own function (say *void Timer2IntHandler()*). Write the prototype of this function with the other prototypes in **tm4c123gh6pm_startup_ccs.c**. *Detailed Instructions in next few slides*
- This function should count(using a global variable(say *times*)) the number of times it is called.

Instructions for Interrupt Usage:1

Open file **tm4c123gh6pm_startup_ccs.c**



Instructions for Interrupt Usage:2

In file **tm4c123gh6pm_startup_ccs.c** add declaration of your own interrupt function as *static void*
(We have added *static void Timer2IntHandler(void);*)

main.c | tm4c123gh6pm_startup_ccs.c

```c
32 void ResetISR(void);
33 void ledToggle(void);
34 static void NmiSR(void);
35 static void FaultISR(void);
36 static void IntDefaultHandler(void);
37 static void Timer2IntHandler(void);
38
39 //*****************************************
```

Instructions for Interrupt Usage:3

In file **tm4c123gh6pm_startup_ccs.c** change the handler corresponding to the Timer to your own function
(We have changed *Timer2IntHandler*)

main.c | tm4c123gh6pm_startup_ccs.c

```
105     IntDefaultHandler,                      // ADC Sequence 3
106     IntDefaultHandler,                      // Watchdog timer
107     IntDefaultHandler,                      // Timer 0 subtimer A
108     IntDefaultHandler,                      // Timer 0 subtimer B
109     IntDefaultHandler,                      // Timer 1 subtimer A
110     IntDefaultHandler,                      // Timer 1 subtimer B
111     Timer2IntHandler,                       // Timer 2 subtimer A
112     IntDefaultHandler,                      // Timer 2 subtimer B
113     IntDefaultHandler,                      // Analog Comparator 0
114     IntDefaultHandler,                      // Analog Comparator 1
115     IntDefaultHandler,                      // Analog Comparator 2
116     IntDefaultHandler,                      // System Control (PLL,
```

Define the function along with the other definitions.

Instructions: 2

- Display MSG1. Wait for switch press. Glow GREEN LED. Display MSG2. Wait for sometime.
- Glow RED LED. Immediately run the timer. Wait in a loop until SW2 is pressed.
- As soon as SW2 is pressed, stop the timer. The value in the global variable *times* is the response time in milliseconds.
- Display response time.
- Repeat similar procedure 5 times. Finally calculate and display the mean response time.

Precautions and Tips:1

- Timing is vitally important for LCD. Refer the manual and give appropriate delays.
- Unlock the switch as you did in previous labs.
- Make functions to separate the digits and convert to char. Bitwise OR of a digit(0 to 9) with 0x30 gives ASCII value of that digit.
- Use *while()* loops to wait for events.
- **Do NOT** call the interrupt function in the program. It is called directly when Timer reaches zero.

Precautions and Tips:2

- The return type and arguments for interrupt function should be *void*
- Calculate Timer load value considering the proper System Clock frequency. System Clock frequency may be obtained by *SysCtlClockGet()*
- Remember that *SysCtlDelay(x)* delays $3x$ clock cycles.
- Make sure you have included "driverlib/interrupt.h" and "driverlib/timer.h"
- In your interrupt handling function make sure you **clear** the interrupt before leaving the function, else it will be called immediately over and over again. Use the *TimerIntClear();* function