**Lab 8 - Timers**

In this lab, we are going to load and run a starter program named “lab8\_reflex\_timer\_startercode” which uses the timer. The provided program functions as a human reflex tester: an LED will assert and the user must push the button as fast as possible to stop the timer and establish a score – faster is better. The reflex time will be displayed on the back channel terminal window.

Your task for this lab will be to modify this code to design a Real Time Clock which will display the time in Hours:Minutes:Seconds on the back channel serial terminal.

**Notes about Timers:**

**Refer to the MSP430 user guide chapter 12.1** on the timers in order to understand how Timer\_A works and the registers available to configure and control this peripheral. Timer\_A is a 16-bit timer, which means it can increment to 0xFFFF (65536 cycles) before rolling over. The two identical timers (0 and 1) on the MSP430 both have capture and compare functionality. There are three timer channels in Timer\_A which can be independently configured to either mode. Capture functionality is used to time events, for example, the time between two events such as a wheel completing each rotation in order to determine vehicle speed. In this scenario, the timer runs until one of these two events occurs at which point the current value of the timer is stored in a capture register and an interrupt is generated. Software can then query the values and calculate a difference. Each increment is the duration of the period of the selected clock.

The other mode which these timers support is a compare mode, which is a standard use for a timer and can be used to provide Pulse Width Modulation like the last lab or a periodic millisecond interrupt as we will use in this week’s lab. It is called compare mode because the timer value is compared against the a programmed register which is assigned by software. When the two registers match, the time has expired and an interrupt is generated or the output pulse width modulation is toggled. If the timer is configured as a periodic timer, it will restart the cycle again. The timer module has three modes one of which must be selected for a specific compare application:

* Up mode: timer will start with a value of zero and increment until a software defined value
* Continuous mode: timer will start at zero and increment until it rolls over at 0xFFFF
* Up/Down mode: timer will start at zero, increment until a defined value, and the start decrementing back to zero

In the case of this lab, we will be using the timer in Up mode because we want to define an interval at which is a minimum timer resolution for our application (1 mS).

**Lab 8 Reading Assignment:**

Please read chapter 12 relating to the timers from page number 369 to 373.

**LAB 8 Instructions**

1. The starter code is the reflex timer. The code given to you enters a periodic timer interrupt and turns on the green led periodically. You must press the button as soon as you see a red light. You can check your reflex time in the terminal program. **Remember, you must switch the TX and RX jumpers on the MSP430, the same way as in the serial communications lab.**

2. Create a new CCS file and modify the starter code to design a real time clock. The code already has the timer setup and serial communication setup. Display the time on the terminal program. It must be in this format **hours : minutes : seconds.**

Remember to add (#include "serial\_msp.h") and import the “serial\_msp.h” along with “serial\_msp.c” file from lab 5 (serial\_communication).

**TA Demonstration**

Show the TA your code, functional board and answer questions posed. 20% credit for

functional code and 80% credit for correct answers about the code. You will be expected to explain every line of code for full credit.