LAB #6: DIGITAL STOPWATCH DESIGN WITH PROGRAMMABLE TIMERS

INTRODUCTION

Real-time systems often require precise timing of data sampling and control operations. This is usually achieved through the use of programmable timer circuits that periodically interrupt the main processor at precise time intervals. To study timed interrupt-driven operation, this lab will require you to design and implement a digital stopwatch using one of the microcontroller's programmable timers to interrupt the CPU.

Two sets of four LEDs in the *Waveforms Static I/O Instrument* are to be used to display time in seconds and tenths of a second. The two digits are to be displayed in BCD format, with time incrementing from 0.0 to 9.9 seconds, and then rolling over to 0.0 and repeating. The stopwatch is to be controlled with two buttons on the keypad interfaced in the previous lab.

Button 0: start/stop timing
Button 1: clear the display to 0

(If you are still resolving keypad issues, you may use push buttons in the *Waveforms Static I/O Instrument* for this exercise – but you will need the keypad in subsequent labs.)

The keypad interrupt routine should determine the pressed button, and decide when to start the timer, stop the timer, and reset (clear) the display. One of the microcontroller's timer modules, set up to interrupt the CPU periodically, is to provide the time base. Whenever the stopwatch is running, a timer interrupt service routine should update the display every 0.1 second. When the stopwatch is stopped, the display should freeze at the last time value. If the clear button is pressed while the stopwatch is stopped, the display should reset to 0.0. The clear button should have no effect while the stopwatch is running. When the stopwatch is restarted, the time should continue from the displayed value.

PROGRAMMABLE TIMER FUNCTIONS

For an overview of programmable timer functions in computer systems, refer to the Monday lab lecture slides, the *STM32L100 Reference Manual (Chapter 18 – General-Purpose Timers TIM9/TIM10/TIM11)*, your course notes from ELEC 2220, and the Cady or Valvano texts. In this lab, a programmable timer is to be used for triggering periodic interrupts to produce a precise time base for executing a set of operations (updating the stopwatch display.)

PRE-LAB ASSIGNMENT

Reading

Study the Monday lab lecture slides, *STM32L100 Reference Manual (Chapter 18 – General-Purpose Timers TIM9/TIM10/TIM11),* and Chapter 14 of the Cady text, 2nd edition, which describes the HCS12 timers, or Chapter 9 of the Valvano text, which describes the Cortex-M and LM3S/TM4C microcontroller timers. Links to ELEC 2220 lectures on STM32 microcontroller programmable timers and interrupt operation are available on Prof. Nelson's ELEC 3040/3050 web page.

Software Design

Building on your previous projects, design a C program that initializes timer module TIM10, initializes the stopwatch to 0.0, and then enters a "do-nothing" loop to wait for a keypad interrupt. Stopwatch operation is to be as follows.

1. When the start/stop button is pressed for the first time, the timer and timer interrupts should be enabled and the stopwatch should begin running, updating the display every 0.1 seconds. ***The timing must be precise. You must measure it to verify the correct time base.***
2. When the start/stop button is pressed again, the timer and/or timer interrupts should be disabled, effectively stopping the timer and freezing the display at the time corresponding to the instant at which the button was pressed.
3. Pressing the start/stop button again should resume timing at the time shown on the display; the time should be reset to 0.0 only if the display has been cleared.
4. Pressing the clear button should reset the display to 0.0, but only if the stopwatch is stopped. The clear button should be ignored if the stopwatch is running.

In your laboratory notebook, record the following ***prior to lab***.

1. flowcharts for program and the two interrupt service routines (keypad and timer)
2. draft program and the two interrupt service routines (or directions to content on H:drive)
3. a plan for testing the stopwatch

LAB PROCEDURE

1. Implement your hardware design, using connections from previous labs. This project requires the keypad (GPIO pins PB7-PB0 and PA1) and two four-bit displays (GPIO pins PC7-PC0). You might also find it useful to connect additional LEDs to unused GPIO pins to display debugging information.

2. Compile, download and test your program. To begin, verify that you can start and stop the timer and trigger timer interrupts, and then verify that the clear operation works. Accurate timing should be verified using the oscilloscope, and correct

stopwatch functionality should be verified with the logic analyzer. Demonstrate a working system to lab instructor.

DELIVERABLES THIS WEEK

Lab notebooks are to be submitted to the GTA at the end of your lab session.