CS/ECE 5780/6780 Lab #5
Spring 2010
Davis, Atcitty, Lee

LAB #5: Stopwatch using Interrupts

The Lab write-up is due to your TA at the beginning of your next scheduled lab. Don't put this off to the last minute! There is pre-lab work to complete before the start of the next lab. **NO LATE LAB REPORTS WILL BE ACCEPTED.**

1 Objectives

- Design the software interface for a stopwatch controller using interrupts.

2 Background

This lab demonstrates how a microcomputer can be used to control a simple stopwatch timer. Your software should be implemented using interrupts when keypads of the stopwatch are pushed. Your software should also make use of the free running timer available on the microcontroller.

The stopwatch should display elapsed time on the LCD screen using the following format.

"hh:mm:ss:th"

h - hours
m - minutes
s - seconds
t - tenths a second
h - hundredths of a second

Note: that 'th' can be anywhere from 0 to 99 but minutes and seconds will
need to be limited to the range 00 to 59

You may have to implement your design to possibly display the hours and minutes on one line of the LCD screen and the seconds and fractional seconds on the bottom line of the screen. As long as all values are visible, should be sufficient.

The stop watch will have 4 buttons taken as input from the keypad. Those buttons are:

START - Starts the timer running again if the timer or display was not showing a running/changing elapsed time. There are three scenarios here depending on the previous button pushed:
1) If the previous button pushed was RESET then the timer starts counting from 0.
2) If the previous button pushed was SPLIT then the timer is still counting and the START push resumes the display of the running elapsed time.
3) If the previous button pushed was STOP then both the counter AND display are stopped at some time value and START will resume the counter from the current time and the display will continue to display the elapsed time. If the display is running and showing the current elapsed time, hitting start again will have no effect.

STOPS - Stops the timer and display shows the total elapsed time. Hitting stop when the timer is stopped has no effect, which can only happen if the previous button push was either STOP or RESET.

RESET - Has no effect if the timer if the previous button pushed wasn't a STOP. If the previous push was a STOP then the timer stays stopped and the elapsed time value on the display is set to 00:00:00:00

SPLIT - When the split button is pushed the display stops counting BUT the timer keeps running. If the previous button push was START then the display freezes on the elapsed time that was in effect when the SPLIT button was pushed, BUT the timer continues counting. If the previous push was a SPLIT then the display will show the time between the previous and this SPLIT push.

Note that there are several variables that make sense:

DISPLAY_VALUE - value being displayed which varies as described above.
ET_VALUE - elapsed time value
LAST_SPLIT_VALUE - elapsed time when the previous SPLIT push happened
THIS_SPLIT_VALUE - elapsed time when the current SPLIT push happened
LAST_PUSH_TYPE - keeps track of which button was pushed last

This implementation requires the design of some external hardware as our microcontrollers only contain a limited number of external capable interrupt pins. We highly recommend that you implement logic to trigger a single interrupt routine to read input, and we highly recommend using IRQ as the input pin to trigger the interrupt. This can be done by consistently driving the inputs to the keypad to VDD, then tie all three outputs in say a three-way OR gate to IRQ, so when a button is pressed the interrupt is triggered and within the routine you can drive individual rows high to determine exactly which button was pressed. You can purchase external IC chips from the stockroom such as NOR, gates, etc… to use in your designs.

Also, it does not matter which keypad buttons you use for which stopwatch function. You may use keypad button 1 for start, button 2 for stop, etc… but you must take input from the keypad in this lab.

3 Pre-lab

Complete task 1 before coming to lab.

4 Tasks

1. Implement a simple ISR routine that displays a message to LCD screen when a keypad button is pressed.
2. Implement the 4 button stopwatch using interrupts and check-off your working circuit with your TA.

5 Writeup

Include the following items:

1. A description of your design and why you chose to implement it that way, in particular which interrupt method you used and why.

2. A printout of your well-commented code.