ECE 455 Embedded Software

Michael Daniel Werezak - 20303777

Lab Assignment 1- Simple Timing

Two approaches to timing were evaluated by using them to display the number of elapsed minutes and seconds on an LCD display, and then comparing the results of a 10 minute run with a stopwatch application running on an android-based cellphone.

Software Delay

A software delay timer was implemented using a pair of for loops, the global SystemFrequency variable and a calibration constant. The outer for loop counted the elapsed seconds, updated the LCD, and called a software_delay() function to delay for 1000 ms. The software_delay() function contained the inner for loop which did nothing other than increment a variable i. The number of times i is incremented was based on the global SystemFrequency variable (the details can be found in the source code). A first run was performed to determine the value of the calibration constant, and a second run was performed to evaluate the timer. The results are below.

Run	Calibration Constant	Software Delay Clock	Stopwatch
1	1.00	10:00 (600 s)	7:07 (427 s)
2	1.40	10:00 (600 s)	9:45 (585 s)

Hardware Timer Interrupt

A hardware interrupt based timer was implemented using one of the 32-bit timer/counters on the LPC 1768. The timer was configured to generate an interrupt every second and reset, before entering a loop that updated the LCD screen when a global flag was raised. This was done to avoid updating the LCD screen inside an interrupt handler. The match register for the timer was set to SystemFrequency/4, based on the peripheral clock configuration info found in the system_LPC17xx configuration wizard in uVision4. The interrupt handler would then set the IR register to clear the interrupt condition, before incrementing a counter tracking the number of elapsed seconds and raising the flag to update the LCD screen. The results are below:

Hardware Interrupt Timer Clock	Stopwatch
10:00 (600 s)	9:59 (599 s)

Comparison and Conclusion

The hardware interrupt based timer performed much better than software delay based timing. The 1 second discrepancy in the HW results can be attributed to the difficulty of starting the stopwatch and the HW timer at the same time.

The inaccuracies in the SW timer on the other hand, is likely due to the processor needing to update the LCD screen and possibly to perform other tasks, during which the SW delay counter is suspended. This source of inaccuracy is very problematic since if more functionality is added to the timer, or if the microprocessor is required to perform additional tasks, the calibration constant will likely need to be recalibrated. Furthermore, the discrepancy of 15 s even after calibration suggests the presence of a timing skew that varies between runs of the timer. Not only does this mean that the SW timer is less accurate, but it suggests that further attempts to improve its accuracy may be challenging.