Lab Exercise #1

Objectives

- Refresh knowledge of and use of the Keil uVision development environment
- Determine the frequency of clock signals inside the microcontroller
- Review use of the PIT timer
- Measure a time interval
- Have a little fun

References

- [2] NXP Kinetis KL25 processor sub-family reference manual (pdf) on class web page.

Lab 1 Problem Statement

A timer is needed to measure time between a first and second pressing of a push button activated by a person's finger. Time is to be counted in milliseconds. Switch bounce occurring in the first 10 milliseconds after the button is pressed should be ignored. If the time between first button press and second button press is greater than 1000 milliseconds, the error light (LED1) will come on and remain on until reset is pressed. When a proper time interval has been measured, i.e. one that is equal to or less than 1000 ms, the number of milliseconds will be displayed on the LCD in either decimal or hex notation. If the time is greater than 1000 ms then Pressing reset will zero the display and set the program up to again look for a button press.

Use a state machine approach to organizing your software and monitoring button presses (which are inputs to state machine), start time counting when the button is pressed, and determine if the allowed time interval is exceeded.

Specific input/output resources to use:

sw1 will be the button that the user presses twice
sw2 will be the reset button
led1 will be the error indicator
LCD display will show the elapsed time or error

Design Flow

The general design flow for today's lab is:

Design a program per problem definition
- Confirm understanding of the problem
- Create a state diagram showing program operation

Code your design
Debug

To Turn In

- In the "comment header" of your main.c file report success, failure, or other observations
- Submit your main.c file to a D2L drop box
- Zip up your complete lab 1 uVision project and submit to the D2L drop box