CE-2812, Lab Week 9, Last Lab, Multitasking

1 Purpose

The purpose of this lab is to explore multitasking on a microcontroller.

2 Prerequisites

- The Nucleo-F446RE board had been mounted onto the Computer Engineering Development Board.

3 Activities

3.1 Background

We saw recently how we can run multiple concurrent tasks simply by swapping the contents of the registers, with proper setup, of course. Seeing it in lecture and doing are two different things, so, let's do it.

3.2 Basic Requirements

Set up your current program such that the Knight Rider Lights (KRL) (from Lab Week 2) runs as a "concurrent" task with your latest console program. You may use any and all information presented in lecture, of course, but, please do not just copy-and-paste code and hope that it works. Develop an understanding of what every line of code is doing as you build your program.

When finished, your board should start up the KRL task and the LEDs should begin sweeping back and forth. You should then be able to interact with your console program as always, and all features of the console program (r/w/dump memory, music, background music, measure frequency, generate sinewave) should all continue to work unchanged.

Now, of course, you will need to make some tweaks, items to consider:

- KRL might use delay_ms() using the SysTick timer, so, you may need to use another timer to invoke task swapping.
- Assuming you use a new timer for task swapping, freeing SysTick for use in delay_ms() does not solve all of your problems because if KRL is using SysTick for delay, your console program cannot also use SysTick. You will need to come up with a way to create a delay without interacting directly with hardware. A good foundation for this is somewhat described by the "non-blocking timer" example in Lecture 6-1.
- Only one task should interact with the console – should not be a problem as KRL (as specified) did not use the console.

3.3 Hints and Suggestions

- Running the two tasks outlined above is the basic/minimal requirement. You may choose to run additional tasks if you wish.
- Switching tasks every 10 ms or so is probably a good place to start. You may wish to experiment with longer and shorter timeslices to see how your application is affected by this property.
- As suggested in lecture, it would be neat to have multiple instances of a task running the same code with different "settings" passed in through the entry point. Absolutely not required, just a thought for a more challenging project.

- The simple two-task setup using an ISR written is assembly is a good choice for this project. You may also choose to write the ISR in C as shown in the later lecture example. You can write the ISR in C without the "fancy" task switcher.
- To fix the delay issue noted above, you might consider having the SysTick fire every 1 ms, for example, and incrementing a static variable. Every 10 ticks, you also swap tasks. The delay routine would just look at the static variable and wait for it to increment the desired delay amount.
- Be sure to test all of your console application features after implementing the task switcher. Be sure to note any differences in behavior in your comments submitted with the lab submission.

3.4 Alternate Assignment

If there is some feature you wish to explore on our platform, you may "design your own lab." You must propose a set of requirements to the instructor by the end of the lab period in Week 9, and be able to demonstrate your lab by the end of the lab period in Week 10. The complexity of the proposed project must be commensurate with the regular assignment lab.

4 Deliverables

When completed:

1. Submit to Canvas a **single pdf** printout of your completed source code to Canvas. **Include in a comment block at the top of your code a summary of your experience with this project.**
2. Ask to demo your lab to instructor. You can do this via writing your name on the whiteboard.
    a. If you demo during lab in Week 9, you will earn a 10% bonus on this lab.
    b. If you demo during lab in Week 10, you will be eligible for full credit.

- Demos are ONLY accepted during lab periods. If you are unable to demo by the end of lab in Week 10, you lose the 10% of the assignment attributed to the demo (per syllabus).

- Demos must be ready a reasonable amount of time before the end of the lab period. If you write your name on the board at 9:45 and lab ends at 9:50, and there are five names in front of yours, you will be unlikely to complete your demo by the end of lab and hence lose the bonus or demo points.

- No labs will be accepted for credit after the final exam period. Period.

4.1 Grading Criteria

For full credit, your solution must:

- Minor errors usually result in a deduction of ~ 3 points (three such errors results in ~ a letter grade reduction)
- Major errors, such as not achieving a requirement, usually result in a deduction of 5 to 10 points.