CE-2812, Lab Week 6, Play a Tune in the Background

1 Purpose

The purpose of this lab is to explore interrupts and implement an interrupt service routine.

2 Prerequisites

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

3 Activities

The previous lab used a timer/counter peripheral to generate tones in a somewhat automated fashion. Once configured and started, the timer/counter can generate a “fixed” waveform with no software intervention.

However, the sequencing of the notes was accomplished by the “mainline,” or foreground, of our program, and hence, while a song is playing, we are prevented, or “blocked,” from doing anything else. By deploying interrupts, we can move the sequencing of notes to the background and continue to use our menu for other functions when a song is playing. This is the basic requirement of this lab assignment.

Specific requirements:

- Leave all existing menu options / commands intact and add option(s) to play music in the background.
- Background music can last for the duration of the song or play continuously and be stopped with another menu command.
- The previous foreground “play song” commands should refuse to operate if background music is playing with a message to the console stating that music is playing.
- You will need to use file-scope variables to communicate with the timer’s ISR. The ISR will need access to the song’s notes, and will need to indicate when it is playing a song and/or finished. In addition, the ISR will need to keep track of where it is in the song.
- Keep in mind our “API Layout” when deciding what to put where. You should have global functions in your API to start (and optionally stop) background music.
- It is somewhat advised that you use Timer 3 to sequence the song. This is the same timer that is driving the piezo and represents the most efficient usage of on-board peripherals even if it requires a little more complicated implementation. If this proves too difficult, you may use another timer.
- You specifically may not use the SysTick timer to sequence notes in the background as your foreground code may call delay_ms()/delay_us() and thus mess with the song sequencing.

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 6, you will earn a 10% bonus on this lab.
   b. If you demo during lab in Week 7, 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 7, 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.

4.1 GRADING CRITERIA

For full credit, your solution must:

- Use a timer/counter in an appropriate mode to play tones as per previous lab assignment.
- Use the best practice for declaring the note structure.
- Implement an interrupt service routine to automatically sequence notes in a song.
- Add “play background music” options to existing menu structure, when option is selected, menu remains usable for other menu functions while music plays in background.
- 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.