Reva Scharf

3/27/15

EC 450

HW5 Write Up

My design for this homework used a lot of similar concepts that we’ve used in past homework assignments. I started with a lot of the code from the tone4 example. This is how I was able to get my sound to work. Then through trial and error I found the correct half period for the note C5. From this I was able to determine the other note values by scaling them with the factor 21/12. All of the half period values for the notes were put in #define statements to make it easier to code songs. My song scores are coded using three variables each. One is an integer array holding the half period values, one is an integer array holding the note duration values, and the third is an integer holding the length of the previous two arrays, giving us the number of notes in the song. I also defined values for the different notes in #define statements that would scale them mathematically correctly. To change the tempo for the song, there is a speed variable that can be changed. This is how my speed up and slow down buttons are implemented: they increment or decrement speed by 0.5 respectively.

For my design I used four external buttons and the on board red and green LEDs. The green LED is on when Joy to the World is playing. The red one is on when the Jeopardy Theme (my other song choice) is playing. The buttons are hooked to P1.2 for the song select button, P1.4 for the slow down button, P1.5 for the start/pause/reset button, and P1.7 for the speed up button. The start/pause/reset button acts like a toggle between play and pause. It functions as reset when held down for three seconds.

I chose not to use any button interrupt handlers and instead debounced and checked the button values in the watchdog timer. The watchdog timer also acted as the “conductor” for the song, handling pauses and note lengths. The only other interrupt handler I used was for timerA to determine the frequencies played.

I think the define statements in my code are definitely a virtue of the design in that it makes entering new songs much easier. There are definitely some limitations as well. I’d be lying if I said my code was perfect, but it will perform the requested operations. The pause button will, on occasion, hold out the note that it is on when paused rather than stopping all sound. After much debugging to fix this, I instead decided to leave it as is, determining my definition of pause to be slightly different. I decided to let pause mean “keep the state the same as it just was and don’t keep going in the song”. By this definition, my button performs to my standard.

For the functionality to work as coded, all resets should be done straight from a play mode state and not a paused state. The song selection can be switched at any time. The speed up and slow down buttons can be pressed multiple times to achieve desired tempo. The slow down button has a lower cap of 0.5 for the speed multiplier. It will not go below this value. If the song has ended already, double tapping the song select button works a more reliable reset than using the reset button. The reset button is more to be used during the song playback to return to the beginning.

For a schematic, please see the image in the github repository. I have also included the Excel spreadsheet I used to calculate the half period values for specific notes.