

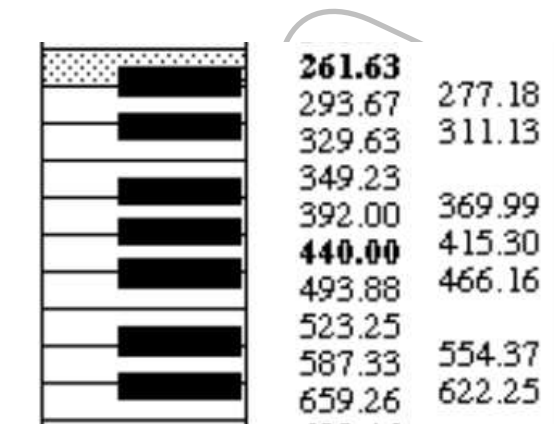
CS452/552 - Real-Time Operating Systems

Lab Assignment #5 Spring 2016

The purpose of this assignment is to get further practice using timers and interrupts.

A musical “note” is a waveform that has a certain fundamental frequency. For example, the “A” above middle C on a piano has a frequency of 440 Hz, and middle C itself is approximately 261.63 Hz (this key is sometimes called C4, with the first C on a piano keyboard being C0). On an equally tempered scale, each of the twelve notes is related to each other by a factor of $2^{\frac{1}{12}}$. After 12 notes (called an octave), the names of the notes repeat; the frequency of a note that is an octave above another is double the frequency of the lower note.

A picture of a piano keyboard, starting at Middle C, is shown below. The “black keys” also have names - the black key between the C and D key is called C#, the key between D and E is called D#, etc.



C4	261.63	
D4	293.67	277.18
E4	329.63	311.13
F4	349.23	
G4	392.00	369.99
A4	440.00	415.30
B4	493.88	466.16
C5	523.25	
D5	587.33	554.37
E5	659.26	622.25

A perfect sine wave consists only of the fundamental frequency, and sounds very “pure,” like a flute. Most waveforms consist of a fundamental and *harmonics* or *overtones*, which are additional frequencies above the fundamental. For instance, a square wave consists of the fundamental and multiples of the odd harmonics (third, fifth, seventh, etc. multiples above the fundamental) in decreasing amplitude. The distinctive sound of a violin is mostly a sawtooth wave (created as the rosin on the bow catches the string, then releases it, all at the microscopic level), with other overtones (called *dissonant overtones*, due to the fact that they are not multiples of the fundamental) created by resonances in the wood.

For this assignment, you are to use interrupts to generate the notes of a musical scale. We will generate square waves by toggling a bit within a port on and off. The notes should be triggered using the keypad you programmed in assignment #3. “Middle C” should play while pressing the ‘0’ key, C# should play while pressing the ‘1’ key, etc.; D# should play on the ‘A’ key, etc. Your program should be able to play at least 13 notes (ie., including C5). Your notes should play within 20 cents (20%) of the correct frequency.

As before, determine the code sizes, and estimate the *percentages* of time your ISR’s and “main” program takes of the total execution time. Also, for each note, estimate its accuracy - that is, how close is the note your program plays to the calculated frequency for the note.