

## LAB #05

Week of October 8

**Objective:**

To become familiar with the use of timers in computer applications. Specifically, to implement a piano like keyboard program using the ATmega2560 timer1 in Normal mode and an external speaker.

**Procedure:**

1. Write a C program for the Arduino Mega SBC which will monitor the keyboard for inputs of A, B, C, D, E, F, and G. Depending upon which key was pressed **use the timer** to generate a square wave on PortB.6 of the ATmega2560 (Arduino Digital pin 12) of the appropriate frequency for the middle octave of that musical note (see Table). The note should last until another key is pressed, at which time the new note will be played. If the 'q' key is pressed the note will stop playing. **For debugging purposes, use of the Arduino Serial Library is permitted for this lab. Use the file serial\_echo\_example.ino as an example of serial I/O.**

**YOUR C PROGRAM MUST CONTAIN A FUNCTION WHICH TAKES THE FREQUENCY AS AN INPUT, AND THEN CONFIGURES AND RUNS THE TIMER. THIS FUNCTION MUST WORK FOR ALL NOTES LISTED IN THE TABLE, IT DOES NOT HAVE TO WORK FOR ALL POSSIBLE FREQUENCIES.**

2. Connect the lab speakers to PortB.6 and ground to actually play the notes so you can hear them.

**BE SURE TO ALWAYS DISCONNECT SBC POWER BEFORE CONNECTING OR DISCONNECTING ANYTHING ELSE TO IT.**

3. Connect a sine wave signal generator to one of the lab speakers, set it to one of the musical note frequencies, and listen to it. Compare the sound of this note with the same frequency note generated from the SBC speaker. Explain why they sound different.

NOTE	FREQUENCY
A	440 Hz
A#	466 Hz
B	494 Hz
middle C	523 Hz
C#	554 Hz
D	587 Hz
D#	624 Hz
E	659 Hz
F	698 Hz
F#	740 Hz
G	784 Hz
G#	831 Hz

The sharp notes are included in the table for completeness. If you choose to implement all of the notes including sharps you will need to decide on an interface protocol which will distinguish between regular and sharp notes. For example, you might define regular notes as upper or lower case and sharps as the opposite.

**If you choose to implement all of the notes you will receive up to 50 points (on a scale of 100) extra credit on this lab.**

HINT: You can probably write a single subroutine which outputs the notes and checks for input. Then, you can call this single subroutine with different timer "load" values for the different notes.