

## Quiz #5a

LabVIEW

Dr. Pogo

Assignment is due in class on Thursday, October 30, 2025

---

### Quiz #4: RS-232 Communication and Sound

|                  |                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Inputs:</b>   | A “radio button” for 49 piano keys<br>A volume control and a tone control<br>A stop button                                       |
| <b>Outputs:</b>  | An “Instrument Missing” warning light, and indicators for frequency, note, and octave.<br>Four “tone selection” indicator lights |
| <b>Hardware:</b> | One 33120A function generator and one speaker (with wires), one null modem cable.                                                |
| <b>Subvi:</b>    | FindCOM (Detect Serial Devices)                                                                                                  |

You may start with the template provided on the course homepage.

- Using a for loop, generate an array of 49 values in the following way  
The first value is 110 (which is the frequency of the leftmost note on the keyboard shown below, an “A”). Every subsequent value is equal to the previous value  $\times 2^{1/12}$ . Subsequent notes should choose sequential elements in the array.  
These values correspond to the frequencies of sequential “even tempered” musical notes in the western scale.
- If the device is not detected, the program should stop without an error.
- The VI should play the single note chosen by the user. The program should output the chosen frequency to the function generator, which will activate the speaker. You will need to learn to use the ‘APPLY’ command for the function generator.  
LabVIEW should not send any information to the function generator unless the user has made a *change* to their selection. Hint: spaces in text commands matter!  
When the user selects “none”, the function generator should play a 1 Hz signal (which is inaudible). Note that kHz is NOT the same as Hz!
- The volume control should vary the amplitude from 0.5 to 3.0 Volts.
- The tone control should choose between *sine* waves, *square* waves, *triangular* waves, and *ramp* waves. Since these wave types have inherently different volumes, the program should automatically make the louder types quieter so that the last 3 types are comparably loud to the sine wave.
- The “Note” indicator should indicate the name of the note being played, and should read “none” if no note is being played (which is how it should initialize).
- The “Octave” indicator should be a number from 0 through 4. On the keyboard shown, there are 4 “A” notes, 3 “B” notes, etc. If the user plays the left-most version of any note, then the Octave should read 0. If the player chooses the next closest version to the right, the Octave indicator should increase by 1. So, for example, in the image shown, the selected key is the second E from the left, so the Octave indicator reads “1”.
- When the program is stopped, all sound should also stop. You can do this by applying a DC voltage.

