

Lab 7: Let's Make a Little Noise

Introduction

Ever had a piece of a song rattling around in your head, but when you tried to sing it, it just did not sound right? What you need is a little keyboard to explore your thoughts and bring out that tune.

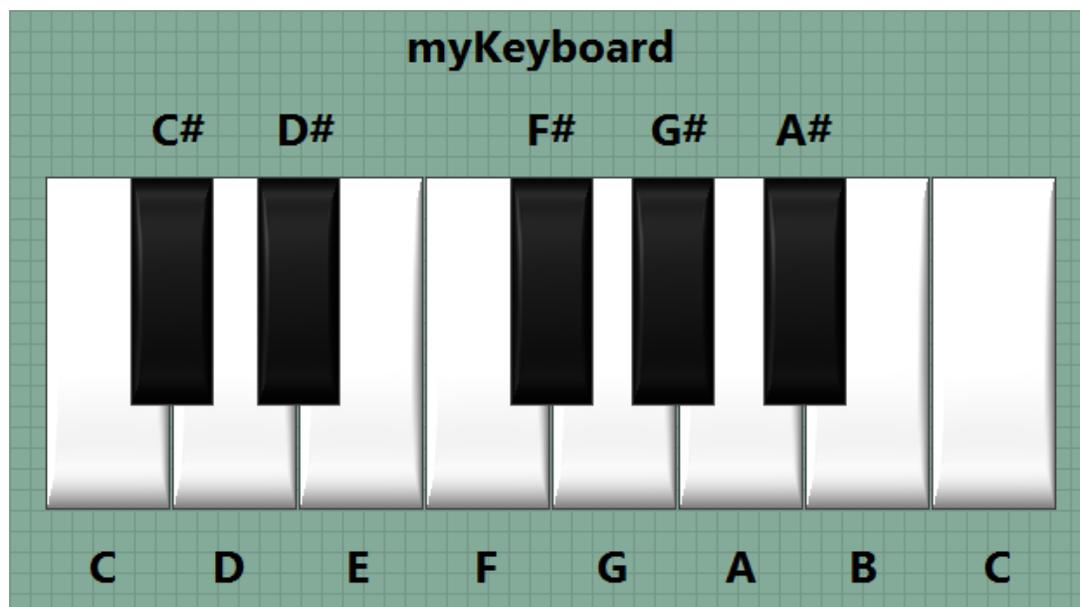


Fig. 1 LabVIEW Rendition of a Single Octave Piano Keyboard

Purpose

This Lab uses the analog output of myDAQ to generate an audio signal from a LabVIEW keyboard (Fig. 1) of one or two octaves. A small electronic audio amplifier adds plenty of volume.

Equipment

- NI myDAQ
- 3.5 mm Sub-Minature Stereo Cable (comes with NI myDAQ)
- iPod or Other Music Source and Earphones
- Stereo Computer Speaker Set (optional)
- LM386 or TI TPA301 Audio Amplifier
- Resistors: 10 k Ω Pot, 10 Ω
- Capacitors: 1 μ F, 1 nF, and 47 nF
- Capacitors (electrolytic): Three 10 μ F and One 100 μ F
- 8 Ω Speaker
- Stereo Audio Sub-Miniature Socket

- Push button switch (SPST)
- 9 V battery
- Solderless Breadboard

Prerequisite Reference Materials

Equipment Check (Analog Output(AO0,AO1) and Analog Input(AI0,AI1)):

<http://decibel.ni.com/content/docs/DOC-12884>

Equipment Check (Audio Output and Analog Input):

<http://decibel.ni.com/content/docs/DOC-12885>

Music Scales:

http://en.wikipedia.org/wiki/Piano_key_frequencies

Audio Amplifiers:

<http://www.techlib.com/electronics/audioamps.html/>

Exercise 7-1: Getting Started on a Piano Keyboard

Audio Check

Connect your iPod output to the myDAQ (Audio In socket). Connect your earphones or external speakers to myDAQ (Audio Out) Socket.

Open and [Run] the LabVIEW program entitled Test Audio.vi.

Start playing a favorite song on the iPod. Sound will tell you that the iPod signal is being streamed from your iPod through a LabVIEW program and back out to the earphones/speakers.

Audio also can be played from a LabVIEW program to the analog output channel 0 (pins AO1 and AGND). Using two alligator cables, connect these two pins to your earphone or speaker cable. If no alligator cables are available, the wires may be manually wrapped around one another to make a connection.

Launch the function generator FGEN and set it to Sine, 1000 Hz, and 0.5 Vpp.

Press [Run] and hear the sine wave signal in your speakers or earphone.

Now that the audio channels are set to go, let's build a LabVIEW program to generate some sounds.

The Keyboard

Load the program Piano.vi and look at the front panel (see Fig.1).

A keyboard for one octave is made up of seven white keys and five black keys. The extra white key on the right side has been added so you can go from middle C (C4) to the higher C note (C5). The frequency for C5 is exactly twice the frequency for C4. The keys are just LabVIEW Boolean indicators shaped, colored, and placed into a keyboard arrangement. Whenever a key is pressed via the mouse pointer, a note is played to the analog output pins (AO 0 and AGND).

[Run] the program and play the scale.

Some Music Background

There are 88 keys on a standard piano keyboard. The frequency range is separated into seven and one half octaves. An octave is the frequency interval between the same notes in two successive octaves. For example, the key middle A (A4) is 440.00 Hz. In the next higher octave, the note A5 is 880.00 Hz, and the next lower octave A3 is 220.00 Hz. There are eight white keys and five black keys in an octave where neighboring keys are separated in frequency by the 12th root of 2 (diatonic scale).

For more information and the frequencies on all 88 notes, checkout

Scales:

<http://www.phy.mtu.edu/~suits/scales.html>

Note Frequencies:

<http://www.phy.mtu.edu/~suits/notefreqs.html>

Simple Piano Keyboard

Look at the block diagram for the program Piano.vi, as shown in Fig. 2.

Lab 7. Let's Make a Little Noise

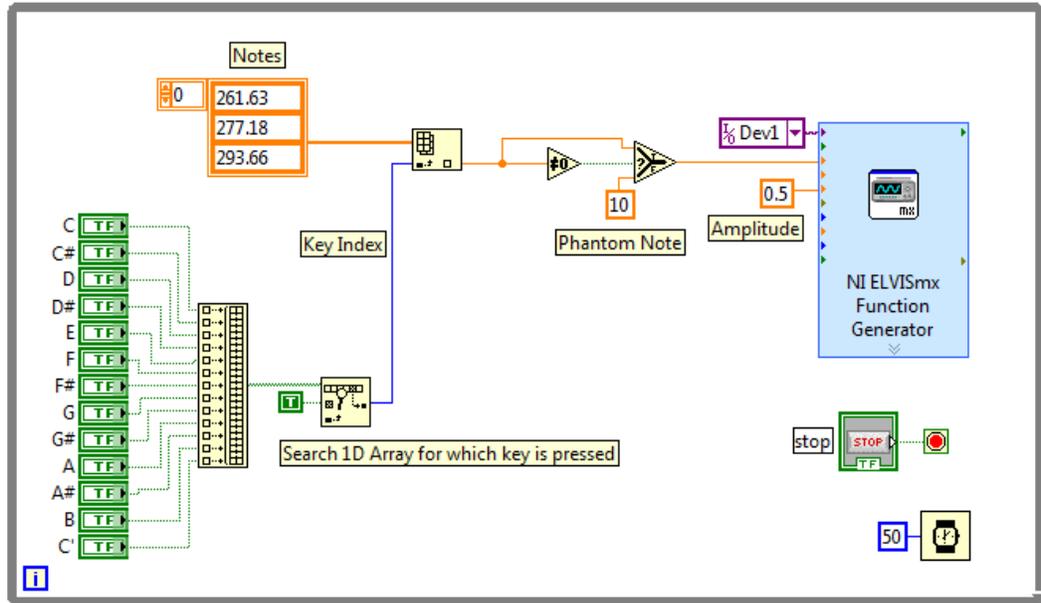


Fig. 2 Block Diagram for LabVIEW Program Piano.vi

The musical note frequencies are stored in an array called 'Notes'. The notes are entered in the array in the same sequence as the keys are laid out on the keyboard.

Example:

Middle C (C4) is element 0 of the array (261.63 Hz).

The keys have a mechanical action of 'Switch Until Released'. The output is 'True' as long as a key is pressed. The keys C, C#, ..., C' are built into a Boolean array whose value is 0 until a key is pressed. The Search 1D Array VI returns a value of -1 (no key pressed) or the integer number related to the key position. This number becomes an index into the 'Notes' array to find the note frequency. The frequency data packet then passes on to the function generator (FGEN), which produces the sound and outputs it to the analog output sockets AO 0 and AGND.

Note: FGEN VI cannot handle a negative frequency number (-1) when no key is pressed, so for this case, a 10 Hz frequency is generated. The ear cannot hear this low a frequency.

[Run] program and play a little jingle.

You soon will hear the limitations of having only one octave.

Modify the program for two octaves (C4 to C6).

Exercise 7-2: Building an External Audio Amplifier

Next we will look at building a low-cost audio amplifier that can drive an 8 Ω speaker, as shown in Fig. 3.

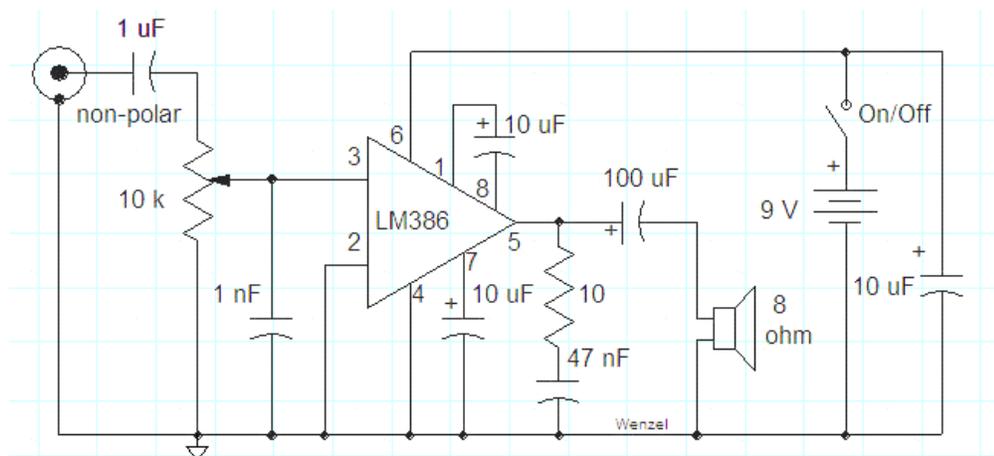


Fig. 3 Audio Amp

Reference: <http://www.techlib.com/electronics/audioamps.html>

You will need the LM386 IC, a few resistors and capacitors, a small speaker, and a power supply. We will use a 9 V battery for the power supply. If no battery is available, myDAQ's analog output port can supply a variable amount of voltage (Measurement & Automation \rightarrow Devices and Interfaces \rightarrow NI myDAQ "DevX" \rightarrow Test Panels \rightarrow Analog Output). Note, however, that the limited amount of current provided by the myDAQ (2 mA) may result in poor sound quality. The input for the amplifier will come from myDAQ analog output sockets AO 0 and AGND. You can use the myDAQ DMM(Ω) software instrument to verify the values of all your chosen resistors. The values of the capacitors are not critical, so the values stamped on the package are good enough. However, if you wish to measure the capacitors, you can use the Bode Analyzer (see Lab 2) and the -3 dB cutoff point (or 45 degree phase point) to calculate the capacitance.

Assemble all of your components on a breadboard, as shown in Fig. 4.

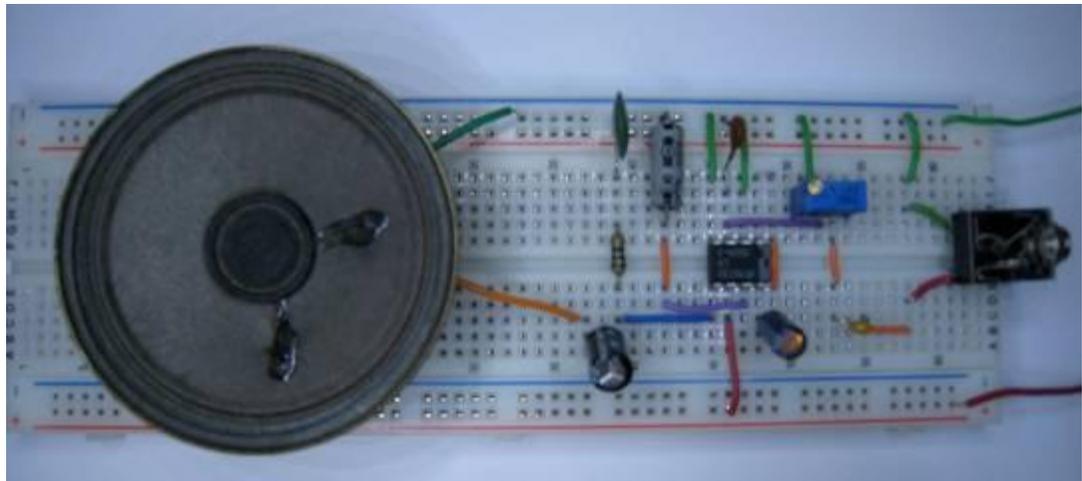


Fig. 4 Breadboard of Low-Cost Audio Amplifier (Single Channel)

Try exploiting the myDAQ instruments to make useful measurements to test and debug the circuit.

Here are some suggestions:

- DC measurements use myDAQ digital voltmeter DMM(V).
- AC measurements use myDAQ function generator FGEN and SCOPE.
- Output signal AO 0 can be connected to both the circuit input and the SCOPE input AI 0.
- The amplifier output pin 5 (LM386) can be connected to SCOPE input AI 1.

Inject a small signal (0.2 Vpp) into the circuit input from the FGEN. Adjust the 10 k Ω pot for a comfortable output. Measure the circuit gain, and compare it with the expected gain.

Note: With no components connected between pins 1 and 8 of the LM386 IC, the amplifier gain is set to 20. This implies that the input signal level (FGEN) should be no more than 0.75 V.

Once the amplifier is tested, then it is time to replace the FGEN signal with the program Piano.vi and make some noise.

Exercise 7-3: More Music Concepts—Quality

Suppose a trumpet, an oboe, and an acoustic guitar were to play the same note (frequency). They all sound quite different. How would you describe the sounds? You might say the trumpet sounds brassy, the oboe sounds haunting, and the guitar sounds mellow. Each note has a fundamental frequency and a set of harmonics. All musical instruments, including your voice, have a unique set of harmonics that make up their sound signatures.

A sine wave has no harmonics, just $\sin(\omega t)$. A composite waveform, however, can have any number of harmonics. Take, for example, a 'triangular' waveform, which has a fundamental frequency and two harmonics:

$$\sin(\omega t) + \sin(2\omega t) + \sin(3\omega t).$$

Let's use myDAQ to explore this idea. The FGEN can only output on one channel, AO 0. However, the iPod connector has two outputs (stereo) with a separate channel for each ear. The arbitrary waveform generator (ARB) has two outputs. Let's put a pure sine wave into one channel (left ear) and our composite waveform into the other channel (right ear). Now it will be easy to hear the difference.

Note: If you are not familiar with the ARB, check out the reference <http://Decibel.ni.com/content/docs/DOC-12941>.

Launch the Arbitrary Waveform Generator

Channel [AudioOutput Left] is enabled and Sine1.wdt entered in the Waveform Name box.

Channel [AudioOutput Right] is enabled and Sine3.wdt entered in the Waveform Name box.

Note: Use the folder icon to navigate to where your .wdt files are stored.

The update rate is 20 k samples per second (S/s) and run continuously.

Connect your stereo speakers or earphone to the myDAQ [AUDIO OUT] socket.

Lab 7. Let's Make a Little Noise

Click on [Run] and explore the differences. Can you hear which channel is the sine wave and which is the composite wave? Try changing the update rate. Any difference?

Compare the wave file pairs:

Sine1.wdt and Sine3.wdt ($1.0 \sin\omega t$, $1.0 \sin\omega t + 0.25 \sin 3\omega t$)
Sine1.wdt and Triangle.wdt
Sine1.wdt and Square.wdt

What can you conclude?

(The larger the discontinuities in the waveform, the harsher the signal sounds.)

Historical Note: Fourier Series Theorem

Any periodic waveform can be generated by an infinite number of sine and cosine waveforms harmonically related.

Look at these examples:

Triangle waveform = $\sin(\omega t) - 1/9 \sin(3\omega t) - 1/25 \sin(5\omega t) + \dots$
Square waveform = $\sin(\omega t) + 1/3 \sin(3\omega t) + 1/5 \sin(5\omega t) + \dots$

Our ability to view these harmonics is the key to many scientific measurements. In the next Lab we will look at techniques for viewing the harmonic content of a periodic signal, or see graphically its sound signature.