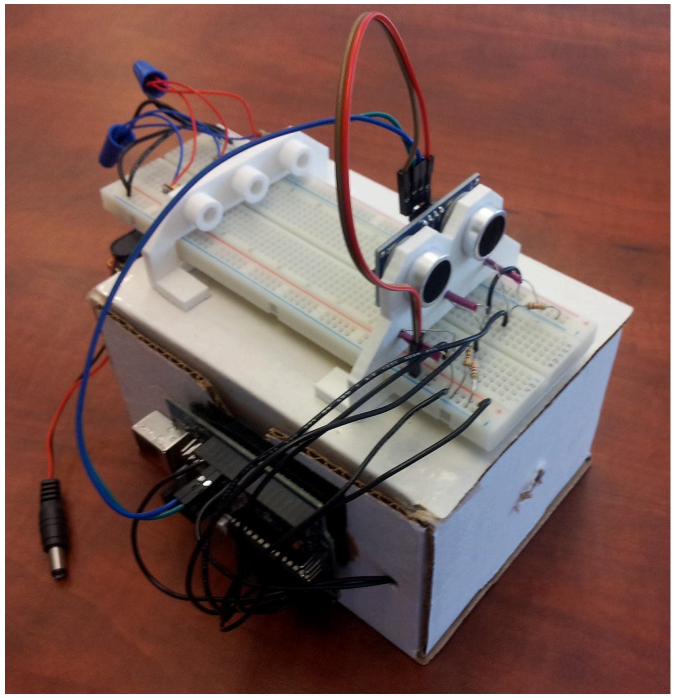
Laser-Guitar Construction Manual



July, 2022

Steve Cox, Northern New Mexico College

Hardware: One Elegoo Super Starter Kit + One Photoresistor + One Speaker + Three Lasers + Two Wire Nuts. More specifically, each guitar uses

1 Arduino Uno with USB cable
1 Full size breadboard
1 Ultrasound Sensor
1 Speaker
3 Photoresistors
3 Lasers
3 1kOhm resistors
2 wire nuts
4 female to male jumpers
2 3D printed parts
Wire

Contents:

1. Overview

This is a construction manual for a Laser Guitar, a riff on the popular laser harp at Meow Wolf

[Meow Wolf Laser harp]

Though our instrument is table-top and has only three lasers, its ultrasonic fretboard will give it a much richer set of tones.

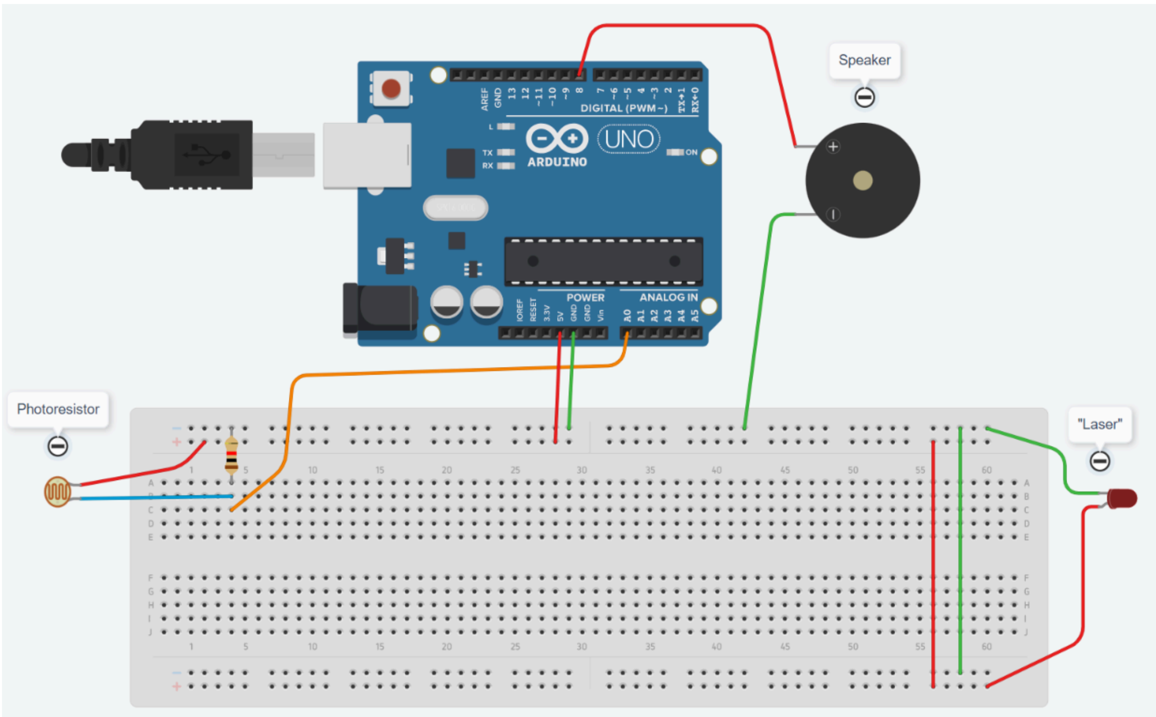
Everything you need to know is here. We presume no previous experience with electronics, lasers, programming, 3d printing or music.

You will need only an Elegoo Super Starter Kit with extra bag-o-parts-and tools, a computer, a Mentor and a group that is open to the notion that electronics, lasers, programming, and 3d printing can make music together.

2. Laser Guitar Part 1

1) We first wire and code our guitar in the cloud at tinkercad.com

Sign-up/in and click **Create New Circuit** and wire up the circuit below.

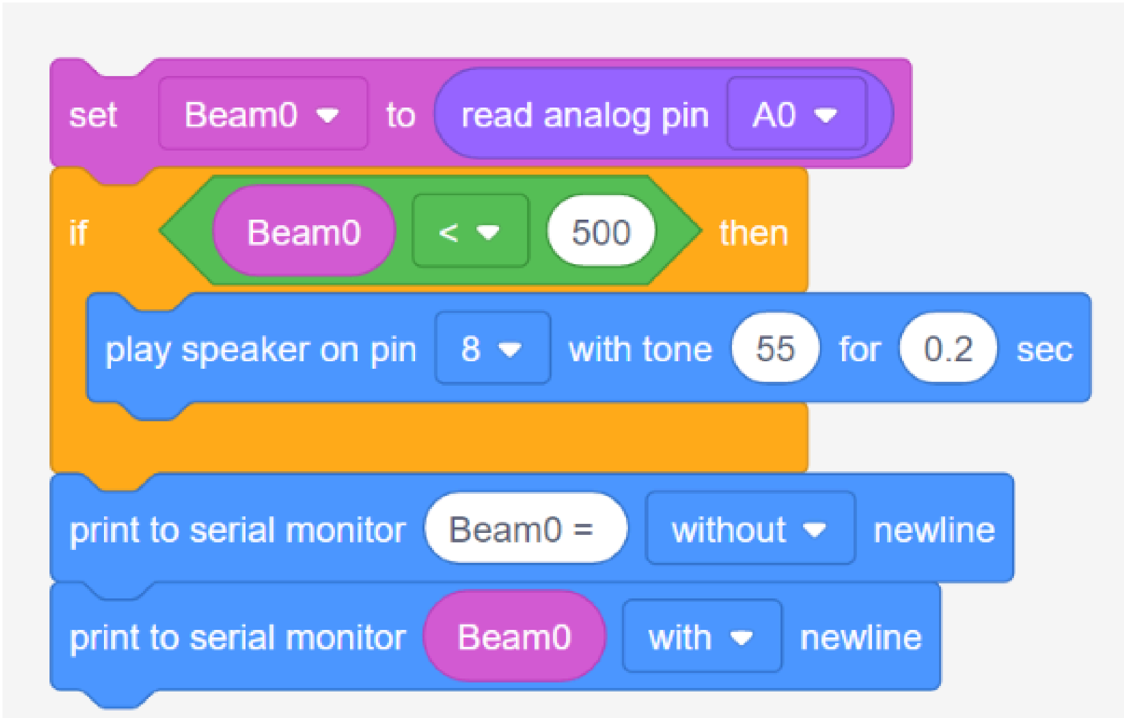


Tinkercad does not have the lasers we need so we've just placed LEDs there to give us a feel for the real thing.

The key here is that tinkercad does have **Photoresistors**, aka Light Dependent Resistors or LDRs. These objects vary their resistance with the amount of incoming light - that we can adjust by simply clicking on a Photoresistor during simulation. [Please watch this demo video]().

2) To get coding, create a variable called **Beam0** that reads the voltage of the top LDR. When this voltage crosses 500 we play a tone.

Enter the code below and play your cloud monochord by clicking on the Photoresistor and turning down the light (as if your finger was blocking the light). Open the Serial Monitor (at the bottom of the code window) to see the actual values of Beam0.



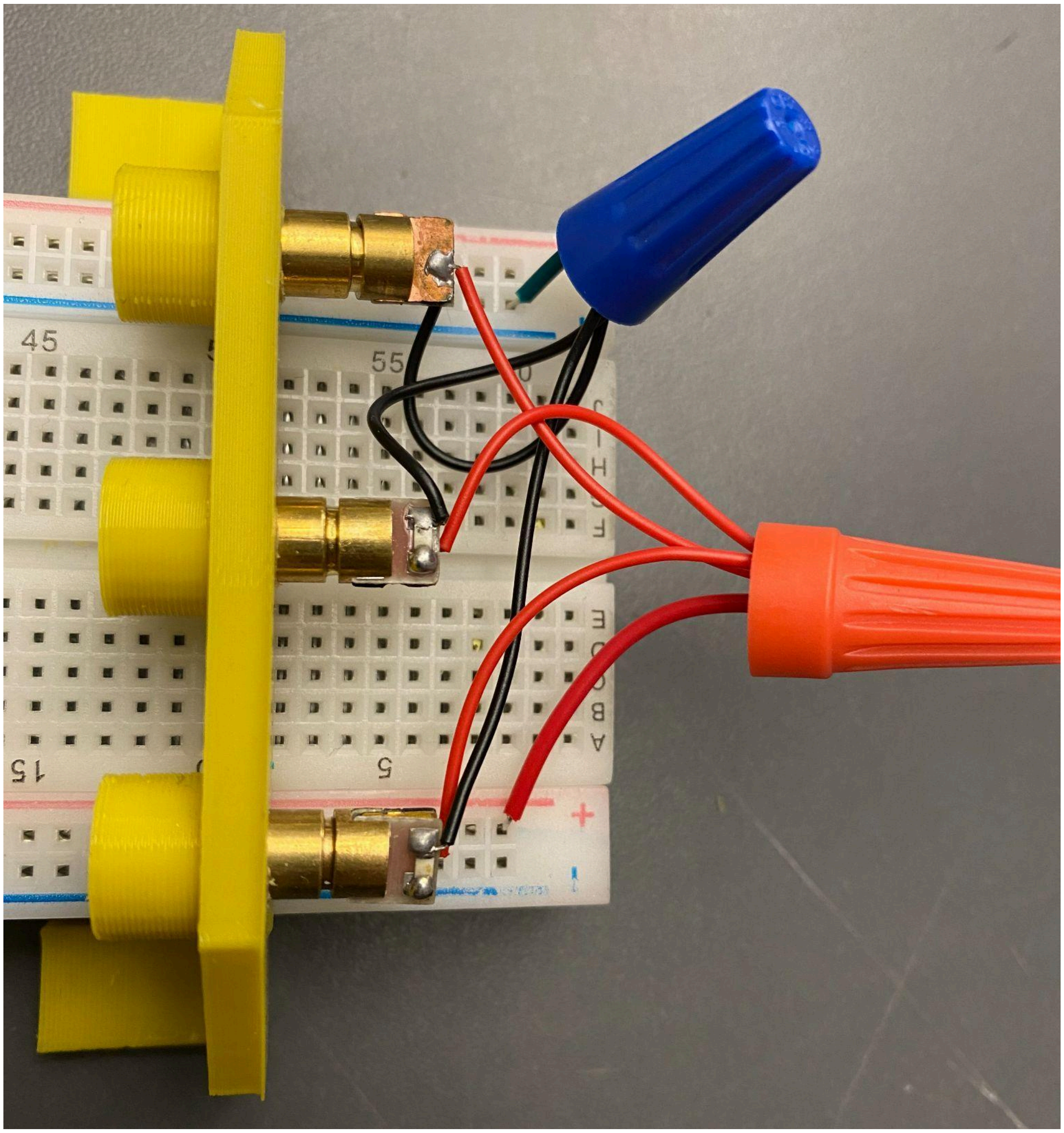
Now try changing the numbers you put in for tone. How does this number affect the guitar? What about when you change 500?

Save your work as Laser Guitar 1.

Now let's build it in the real world.

1. Place the laser holder over one end of the breadboard. Insert the 3 lasers into the three sockets of the laser holder.
2. Strip the 6 laser wires so that about ½ inch of metal is visible. Please be careful not to pull these wires away from the lasers. You may use the wire strippers or just your fingernails.
3. These wires are too fine to insert into our breadboard, so we instead *twist* the 3 blues together with a short jumper (stripped on each end) into a wire nut, and place the open jumper end into the blue rail (-) on the breadboard.

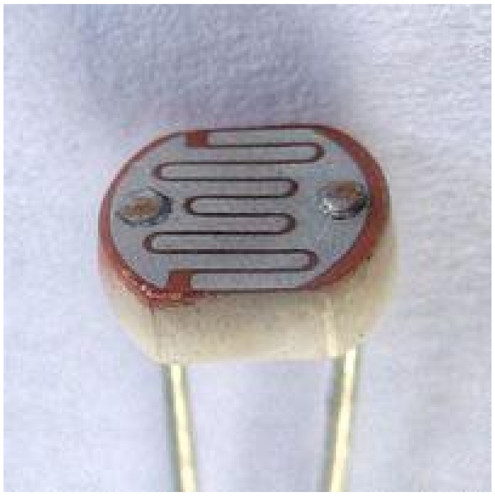
4. *Twist* the 3 reds together with a short jumper (stripped on each end) into a wire nut, and place the open jumper end into the red rail (+) on the breadboard.

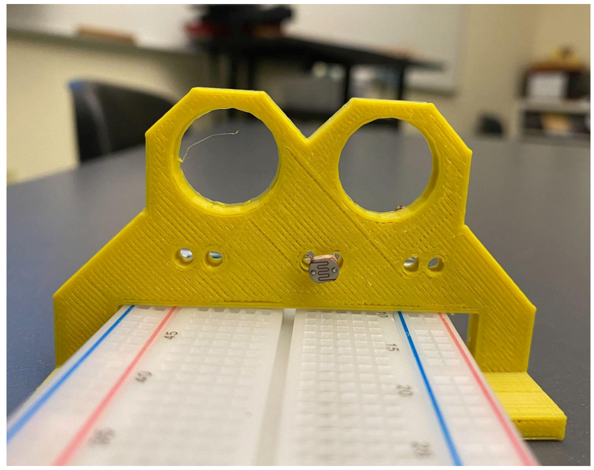


5. Jump the red rail to 5V on the Arduino. Jump the blue rail to GND on the Arduino.
6. Connect your Arduino to your computer and enjoy, *carefully*, the laser light. Then disconnect the battery. Do not shine lasers around the room or in others' eyes.

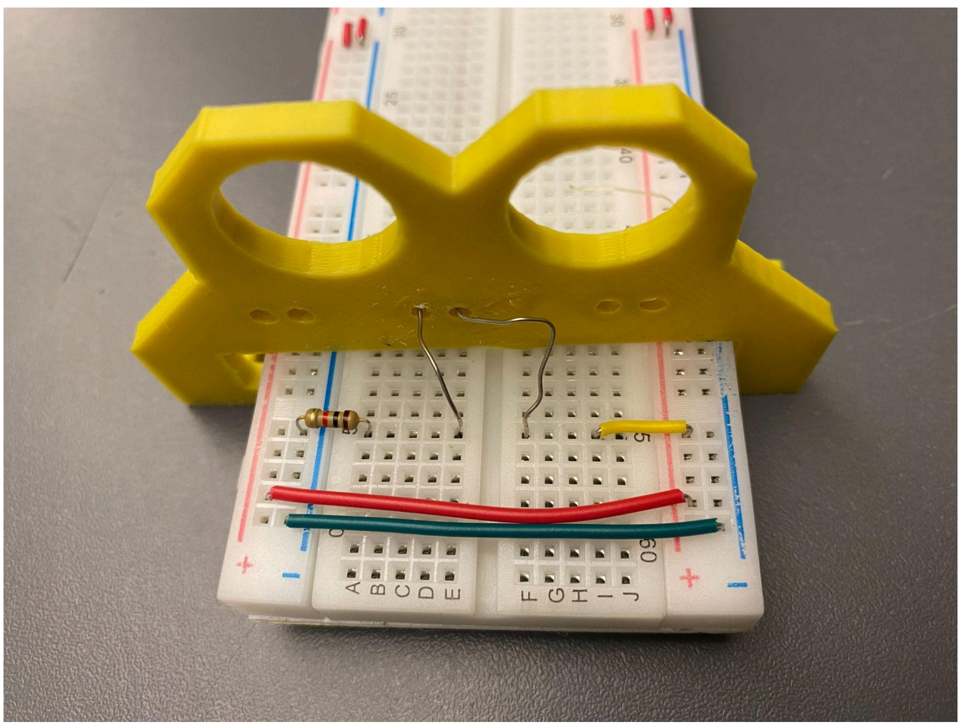
3. Wire up your Light Dependent Resistors and Speaker

1. A Light dependent resistor (LDR) has two legs and a head with a cool snake pattern.





2. Next, we thread one LDRs through the white plastic LDR holder.
3. Next, one leg to a red rail (power). For the middle LDR this requires an extra jumper.
4. The other leg should go to a distinct row, at which it connects to a 1k resistor.
5. The other end of each resistor should go to a blue rail (ground).



6. The LDR-resistor junction needs a read-out jumper over to the Arduino, to A2.
7. Finally, it remains only to add a jumper between the two red rails and the two blue rails at the Laser end and to connect a speaker to your Arduino at pins GND and 8

4. Upload Code Play your single string guitar.

You will download your tinkercad code to the Downloads folder (or a folder of your own creation) on your computer.

If you have a **laptop**, you will open this code in your Arduino IDE and then from there upload the code to your physical Arduino. Here is a video that steps you through this process.

If you have a **chromebook** please follow this link Arduino Create app, to install the free app and set up your free Arduino Create account. Now you can

- Download your Arduino Code from the tinkercad cloud to your chromebook
- Import the chromebook Arduino code to the Arduino Create Web Editor
- Upload this code from the Web Editor to your physical Arduino
- Here is a video that steps you through this process.

1. Turn on your laptop and connect the USB cable from your laptop to your Arduino.
2. Align your lasers and LDRs so that each laser strikes the center of its LDR.
3. Select Tools from the top menu, then scroll down to ensure that Board reads “Arduino/Genuino Uno” and Port that says Arduino Uno next to it.
4. Select the right arrow (upload) from the second row menu.
5. Select Tools from the top menu and Serial Monitor.
6. As you interrupt the A0 beam with your finger, the number should go way down and you should hear a tone. If not, adjust your code based on what the serial monitor is reading.
7. Experiment with the tone by changing its frequency (880) and/or duration (200) in the code. Upload your changed code and enjoy.

I guarantee things will sound infinitely better when we come down from the cloud.

You are undoubtedly curious about our choice of **500** for a threshold and **50** for a tone. What do they signify? Our answers rely on powers and roots of 2!

threshold: The Arduino senses the Analog world and converts this reading into a number using a 10-bit Analog-to-Digital Converter. As such it converts the world into whole numbers between 0 and 2^{10}-1, i.e., between 0 and 1023. With this knowledge do you see that our 500 is close to the middle of the range?

tone: The western 12-tone scale is revealed by the beautiful connection between piano key number and the frequency it produces

$$\text{(piano)} \qquad \text{frequency} = 440 \times 2^{(\text{key}-49)/12} \text{ Hertz}$$

This 440 corresponds to the 49th key, and is called concert pitch (the A above middle C). Now the tinkercad play speaker block does something very similar. It takes your tone number and generates

$$\text{(tinkercad)} \qquad \text{frequency} = 440 \times 2^{(\text{tone}-57)/12} \text{ Hertz}$$

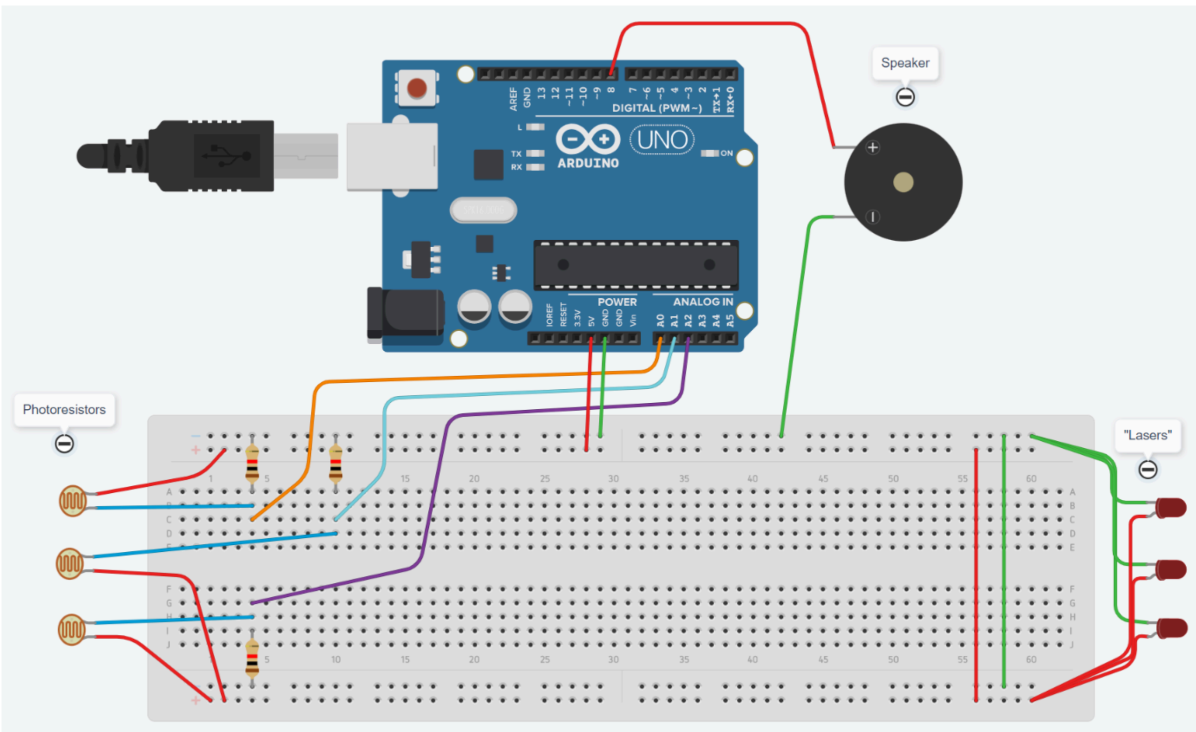
So, what frequency are we hearing when tone = 50? First do the math then look under the tinkercad hood by choosing **block + text** in your code window. Did you get it right?

I don’t know why tinkercad changed the piano’s 49 to 57? Ideas?

That Hertz? Hertz is the 19th century German scientist that pioneered our study of sound. Hertz is German for “Heart,” and one Hertz means one “beat per second.” The human ear can hear as low as 20 Hertz and as high as 20,000 Hertz.

5. Laser Guitar Part 2

1. Duplicate your project and rename it "Laser Guitar Part 2"
2. In tinkercad wire up two more photoresistors to create 2 more 'strings' on your guitar. Make sure to add your read-out jumper wires at A1 and A2.
3. Now it's time to code the strings. Create two new variables, **Beam1** and **Beam2**, and duplicate twice the code above to arrive at a three laser cloud harp. I recommend **tones** of 60 and 70 for these "higher" strings.
4. Start your simulation and make sure you have a working 3 string laser guitar in the cloud.



6. Build Laser Guitar Part 2 in person.

1. Time to wire up the additional 2 photoresistors. Remember one leg goes to ground through a resistor and the other leg goes to 5V.
2. Download your code from Tinkercad and upload it to your arduino.
3. Now you should have 3 working strings on your guitar.

Now, let's all tune our guitar beams to the frequencies

196, 220, and 247,

corresponding to the notes

G3, A3 and B3.

Upload and play Merrily We Roll Along (a dash means a pause, no note)

BAGA BBB- AAA- BBB-

BAGA BBB- AABA G

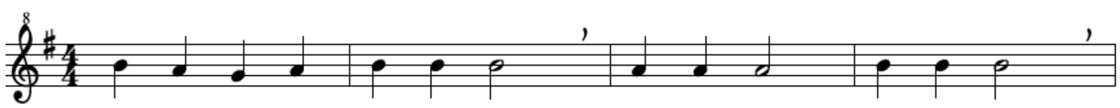
Some groups should tune down an octave to

G2, A2 and B2 at frequencies 98, 110, 124

while other groups tune up to

G4, A4, B4 at frequencies 392, 440, 494.

Can your harps play in unison? Record a video of your concert.



Learn and Play these other 3 note songs.

Hot Cross Buns

|B- A- |G---|B- A- |G---|G G G G |A A A A | B- A- |G---|

French Folk Song

|G G G A |B- A- |G B A A |G---|

Gently Sleep

|G A B- | A G A-| G A B- |A B G- |

Practice, practice, and record videos of solos, duets, and ensembles.

7. Laser Guitar Part 3.

When playing a standard guitar, one hand plucks while another frets. To fret is to push down on a string until it hits the fretboard. By doing this one changes the length of the string and so the sound of the note (shorter strings have higher frequencies).

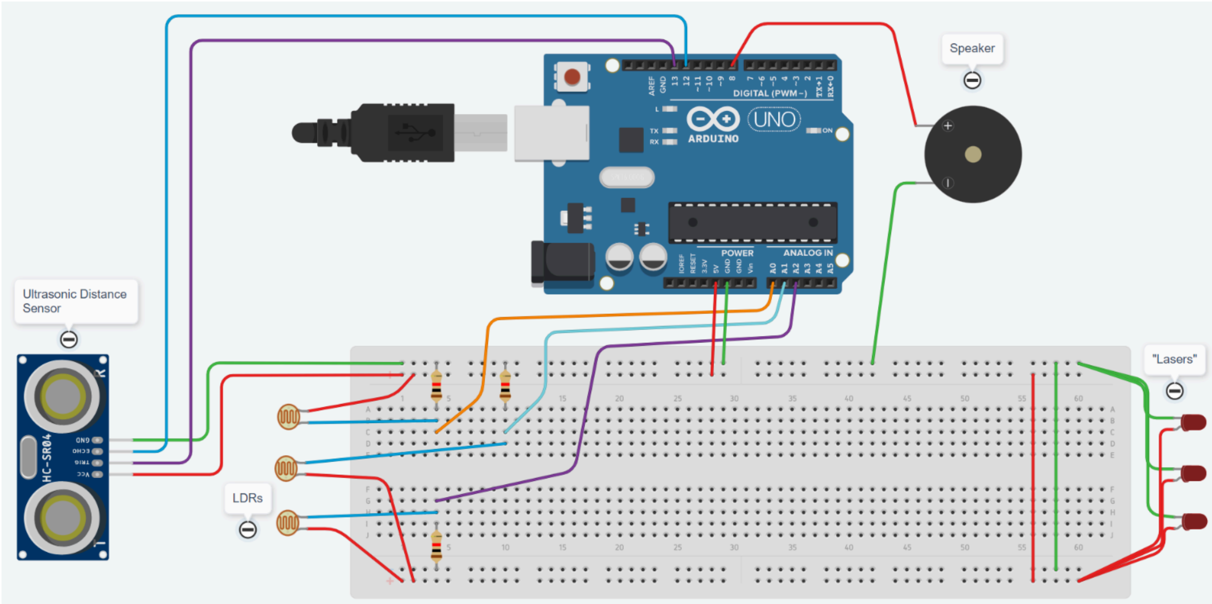
In keeping with our theme of a touchless guitar, let us see that we can use our ultrasound device to measure the distance of our fret hand from the guitar. **Duplicate your project and name your copy Laser Guitar Part 3.**

Tinkercad has two Ultrasound Devices in its toolbox (select Components, All) and we will use the HC-SR04 because that is what is in your Elegoo Super Starter Kit. This one has four pins. This device works by generating a wave (at a fixed ultrasound frequency (> 20 kHz)) and then measuring the time it takes to receive a reflected version of that wave. It finally transforms this time into a distance to the reflector - so long as the reflector is in the active cone (shading) region of the ultrasound device.

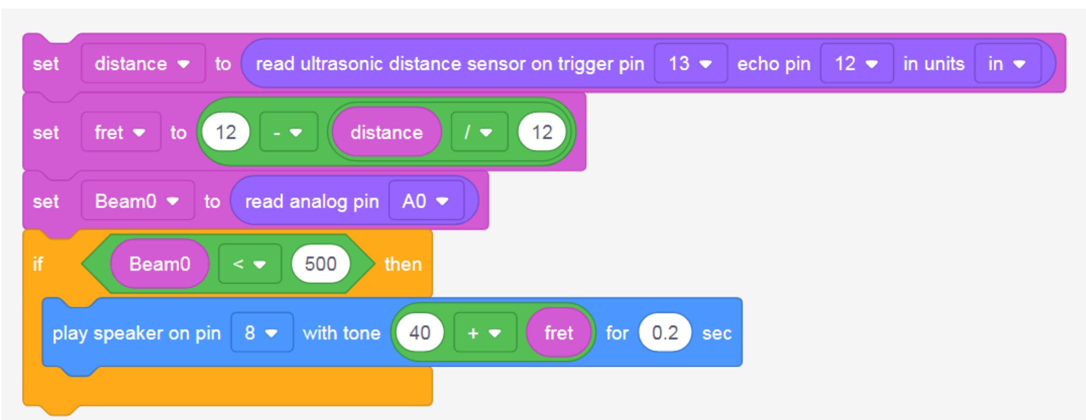
The device has 4 terminals,

- VCC (power)
- TRIG (trigger the outgoing wave)
- ECHO (receive its echo from a reflector)
- GND (ground)

The VCC must connect to the 5V terminal of the Arduino, and the GND must connect to the GND of the Arduino. The TRIG and ECHO pins can connect to any digital pins (other than 0 and 1). We have chosen 12 for the echo and 13 for the trigger in our figure.



As one frets closer to the guitar we expect higher notes. To mimic this we create variables called **distance** and **fret** and use them to vary the tone in our old guitar code.



This “string” has an **open tone** of 40 to which we add 12-(distance/12).

Please duplicate this logic for your other two beams - by choosing different **open tones** but using this same **fret function. Remember you already have most of this code- you just need to modify what you already have.**

5. Wire up your Fretboard

We follow our tinkercad work and use 4 female-to-male wires to connect our UltraSonicDistanceSensor (USDS) to our breadboard and Arduino.

1. Connect Vcc on the USDS to a red rail on the breadboard
2. Connect Gnd on the USDS to a blue rail on the breadboard
3. Connect Echo on the USDS to pin 12 on the Arduino
4. Connect Trig on the USDS to pin 13 on the Arduino

and then download our Laser Guitar 2 code from tinkercad to our chromebook, then open this in our Arduino Create App, then upload this code to our physical Arduino.

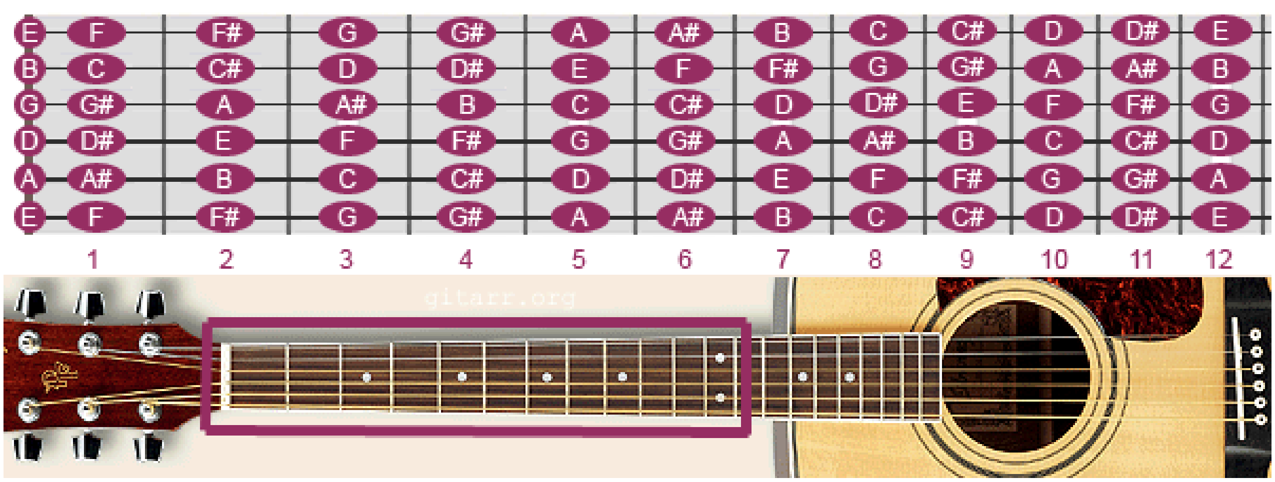
That code has a fretboard that was 12 feet long! We must tune this to arm's length to get a useful guitar.

To accomplish this we position our fretboard so that fret 12 has distance = 1 inch while fret 1 has distance = 12 inches from our ultrasound device (and all other frets are 1 inch apart). We capture this scenario in the simple formula

$$\text{fret} = 13 - \text{distance}$$

so long as distance is within the range of our fretboard, i.e.,

$$0 < \text{distance} < 13$$



The notes corresponding to the 12 frets on the 6 strings of a guitar. From https://www.guitar-chord.org/fretboard.html

To begin, we tie all of our desired notes back to keys 20 through 42 on the **piano** and their associated frequencies.

key	note	frequency (Hz)	
42	D4	293.665	String 4, Fret 12
41	C#4	277.183	
40	C4	261.626	
39	B3	246.942	
38	A#3	233.082	
37	A3	220.000	String 5, Fret 12
36	G#3	207.652	
35	G3	195.998	
34	F#3	184.997	
33	F3	174.614	
32	E3	164.814	String 6, Fret 12
31	D#3	155.563	
30	**D3**	**146.832**	**String 4, open**
29	C#3	138.591	
28	C3	130.813	
27	B2	123.471	
26	A#2	116.541	
25	**A2**	**110.000**	**String 5, open**
24	G#2	103.826	
23	G2	97.9989	
22	F#2	92.4986	
21	F2	87.3071	
20	**E2**	**82.4069**	**String 6, open**

We associate

- Beam0 with String 4 and so open note at key 30
- Beam1 with String 5 and so open note at key 25
- Beam2 with String 6 and so open note at key 20

and resort to the classical piano formula

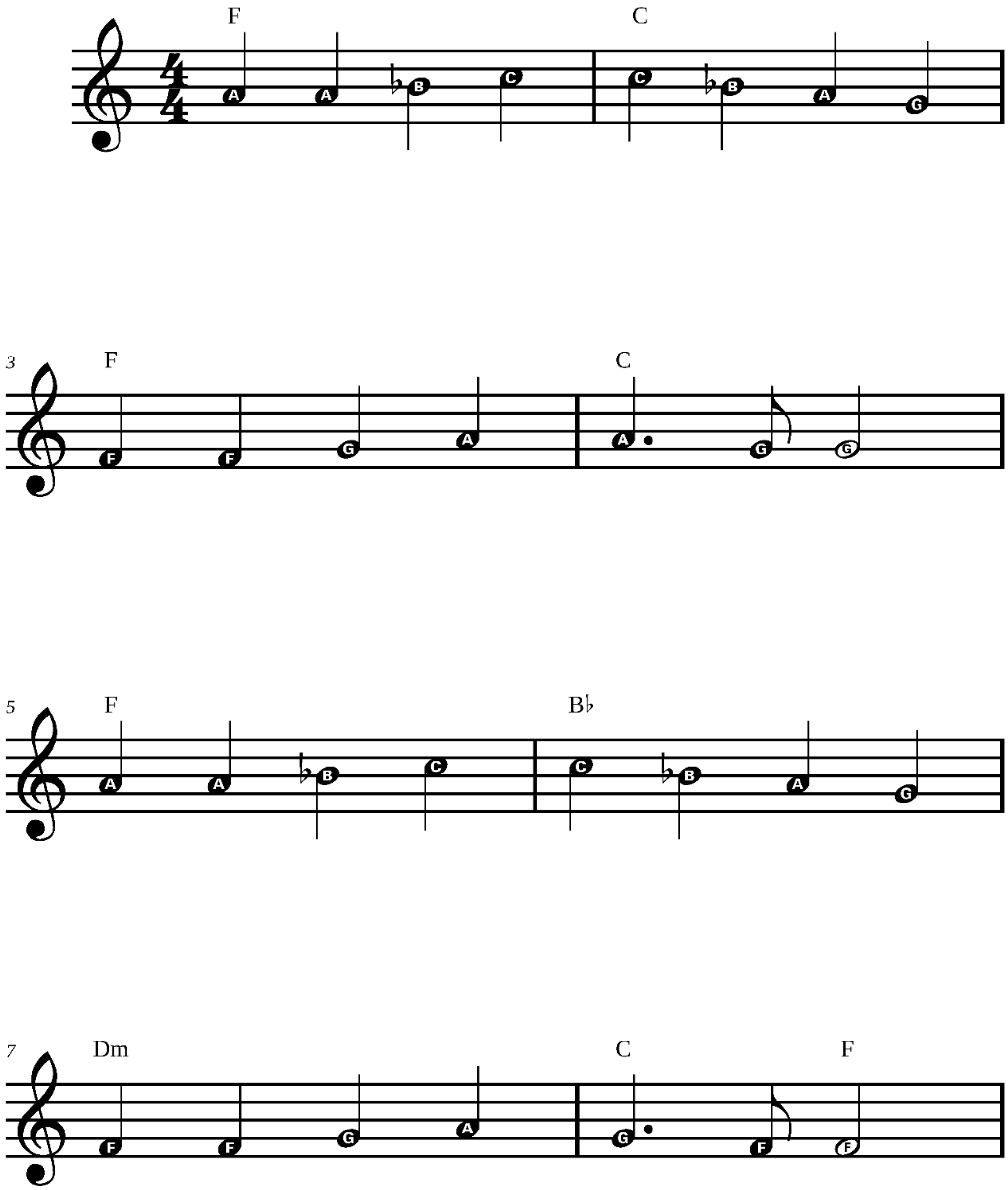
$$\text{frequency} = 440 \times 2^{(\text{key}-49)/12}$$

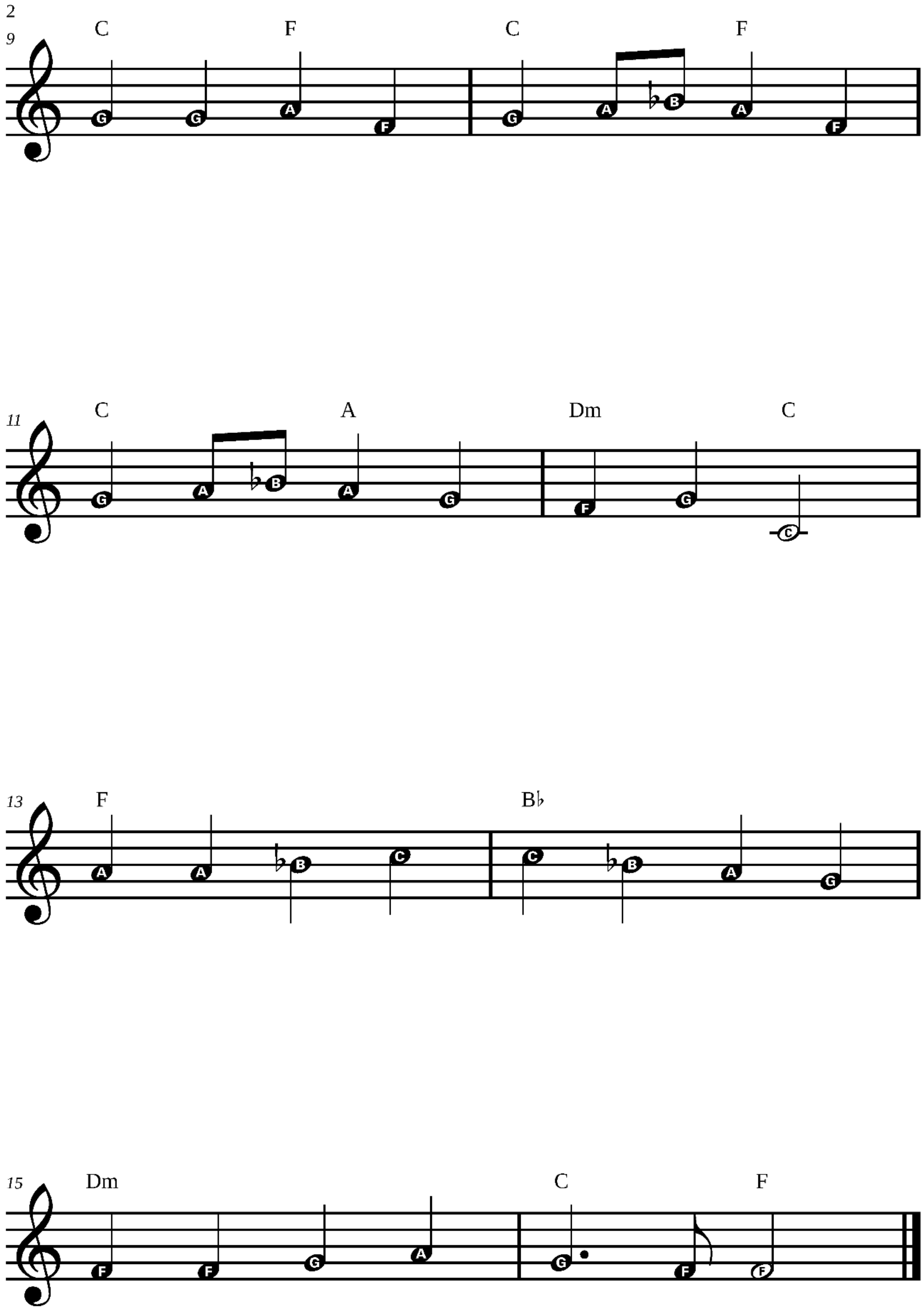
To translate these ideas into code, open your **Laser Guitar 2** code in your Arduino App and revise your code to call playfreq(Beam#) when Beam# is "plucked." For example if Beam0 is below threshold we call playfreq(0). The playfreq function is below, and should be entered after the final } in your existing code.

```
void playfreq(int Beam) //
{
  int key = 30 - Beam*5;
  distance = 0.006783 * readUltrasonicDistance(13, 12);
  if ( distance > 0 && distance < 13 ) {
    int fret = 13 - distance;
    key = key + fret;
  }
  int freq = 440*pow(2,(key-49.0)/12.0);
  tone(8, freq, 300);
}
```

If you’d like to tune your beams to other strings, I recommend that you consult this cool page.

You now have an instrument that can play 39 notes. Like any musical instrument it will take some time to learn. You may wish to add a marked cardboard fretboard before attempting Beethoven’s Ode to Joy.

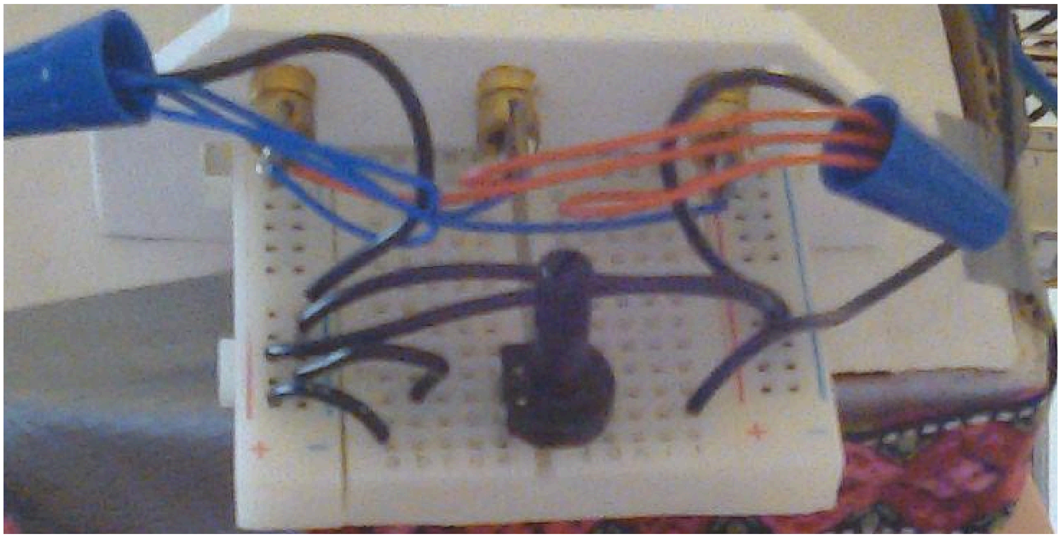

F
C
3
F
C
5
F
B♭
7
Dm
C
F


2
9
C F C F
11
C A Dm C
13
F B♭
15
Dm C F

and De Colores or perhaps Leonard Cohen's Hallelujah or perhaps a composition of your own. Please improve on my recording.

6. Variable Threshold

Depending on the light conditions and the conditions of our lasers and batteries we may need to change the threshold we have been using to detect a "pluck." To free the performer from having to change any code we add a "knob" with a variable resistor to set the threshold.



The two pins at its left are tied to the + and - rails while its important "sweeper" pin, at right, is attached to A3 on the arduino. To make use of it please use the code fragment below in place of the old code, and extend this idea to the other beams.

```
int pot = analogRead(A3);
int Beam0 = analogRead(A0);
if (Beam0 < pot)     {
      playfreq(0);
}
```

Please adjust the knob to a threshold that works for you.

Finally, if you wish to go deeper into the electronics or into using your own custom sounds please check out this Wave Laser Guitar Manual.