Laser Guitar

School of Engineering and Applied Sciences
Harvard University
33 Oxford Street
Cambridge MA.

Description: A guitar that uses lasers and digital processing instead of strings to create music

By: Student A Ontiveros and Student B Levenson

Class: ES50
Instructor: Marko Loncar & Christopher Lombardo

December 2014

ABSTRACT

Dealing with guitar strings can be quite frustrating. Having to retune a guitar and replace strings can be time consuming. Thus, the project seeks to replace the guitar strings with digital components to make music. To replace the strings, lasers and photo transistors were used. The current running out of the phototransistor was then connected to an Arduino Uno microcontroller. Based on certain inputs from the phototransistor, the Arduino Uno could send MIDI notes to a computer, which could read these MIDI notes and replicate the sounds from a guitar.

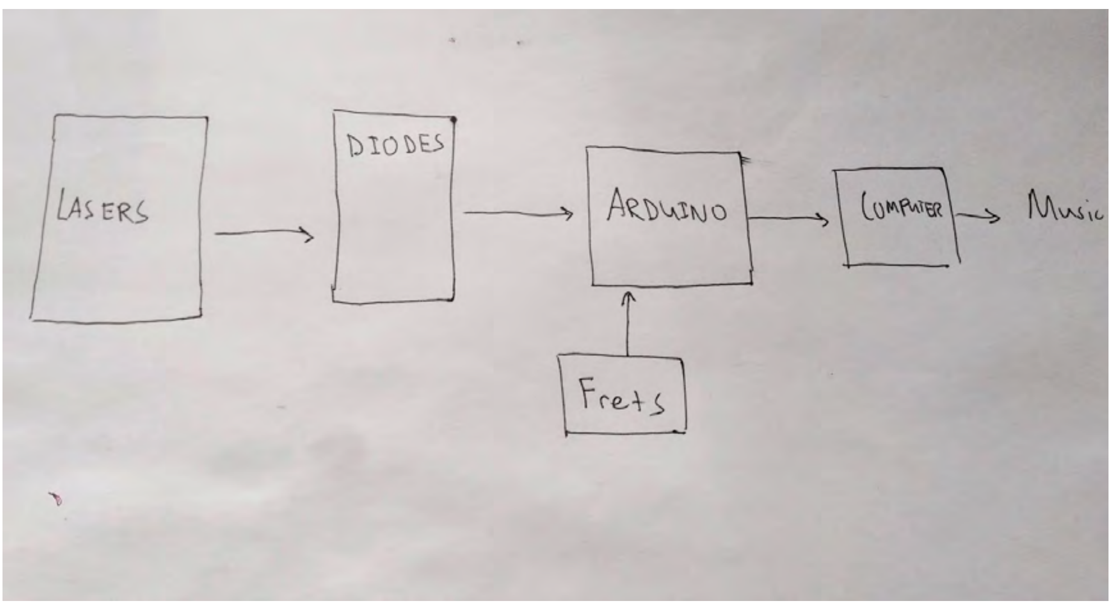
INTRODUCTION

The overall goal of the project is to emulate a real guitar using the concepts we learned in class. The foundation of the project comes from our ability to program and use an Arduino Uno microcontroller. Once we had an idea of how the arduino would take in certain inputs and what it would output, designing the rest of the laser guitar would be somewhat simple considering our knowledge about lasers from Lab 8. We chose this project because we have an interest in music and wanted to learn more about how MIDI works.

DESIGN

1) Preliminary design ideas

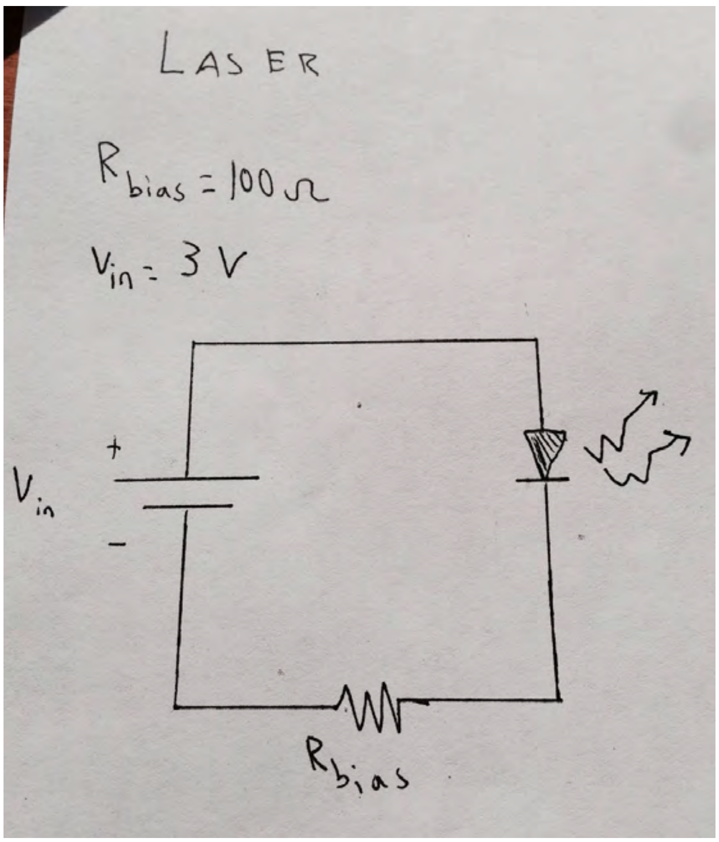
When we first came to the drawing board, we created a basic block diagram of how we wanted each part of the laser guitar to function.



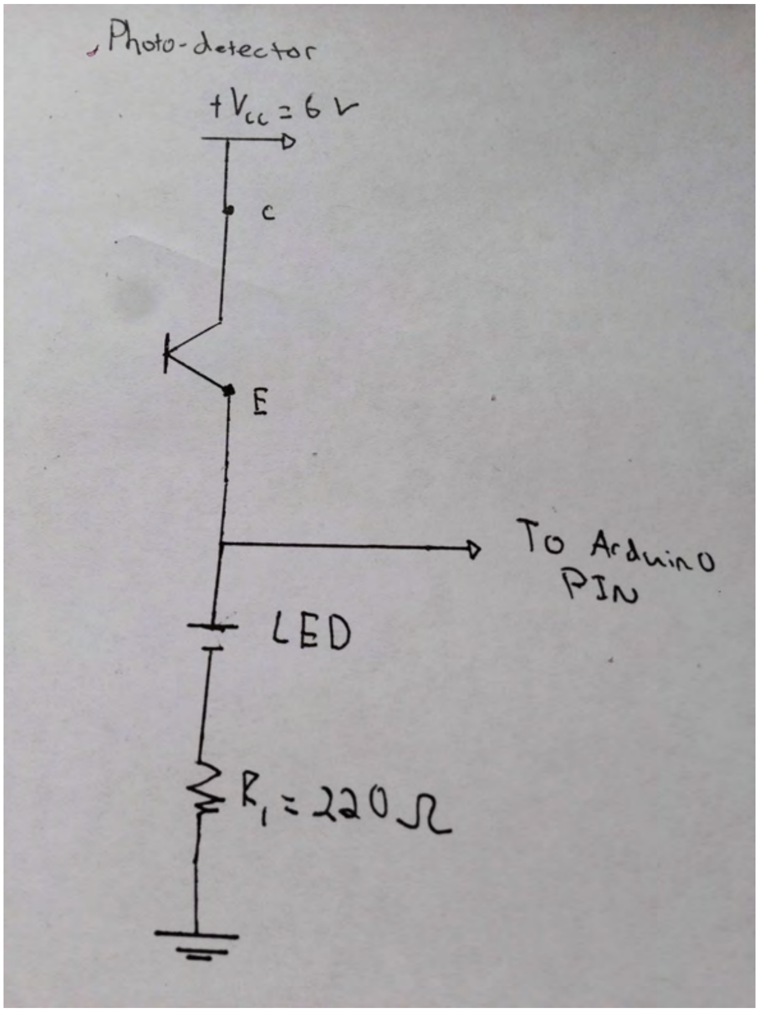
The above diagram is the very basic model for the laser guitar. Basically, the Arduino would take in inputs from the phototransistors and frets (pushbuttons). Based on a given input, the Arduino would output the MIDI notes to the computer. We decided on this initial design because the Arduino had enough inputs for us to work with, so to simplify the project we could just have inputs running from the diodes and frets straight to the Arduino. Even though we had ten inputs (six lasers and four frets) and one output, writing a code for this would have been much simpler and easier to modify than if we had used shift registers or multiplexers since there are several combinations of frets and lasers to produce different sounds. Also, in order to figure out how the fretting system would work, we decided to count in binary. With four pushbuttons for frets, we could use each fret as a 1 or 0 (depending on if the button is pushed or not), which the arduino could process (see Table 1 of the Appendix). In terms of playing guitar chords, the Arduino Uno could not output more than one MIDI note at a time, so we decided to not include chords so we would not have had to use multiple Arduinos.

2) Prototyping

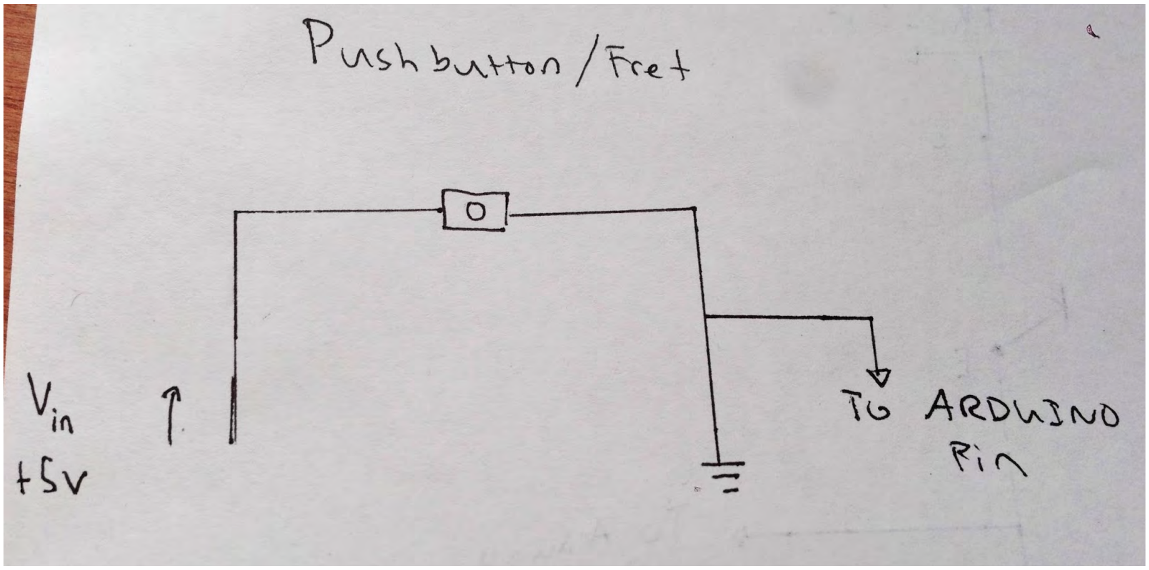
After figuring out a basic block diagram and obtaining the parts we needed, we looked to making a few circuit diagrams for the lasers, phototransistors, frets, and MIDI output.



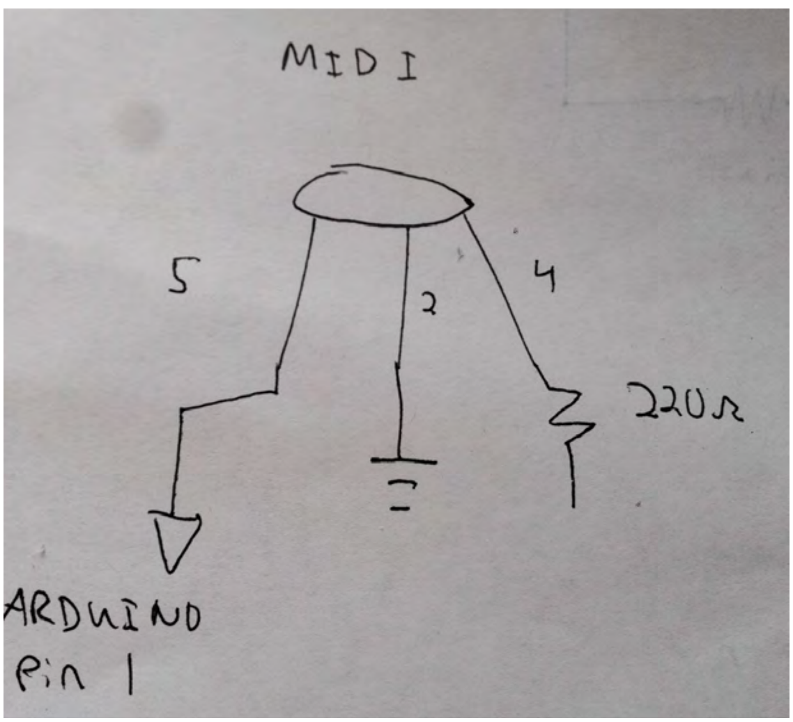
The above diagram is the circuit for each laser. We chose Vin = 3v and Rbias = 100 ohms based on our results from Lab 8. In Lab 8, we measured the current/voltage characteristics of a laser diode under this conditions. See Table 2 in the Appendix for the I-V values we had for the setup. Even though the laser began to emit at 1.8 V and the current dramatically increased at 2.4V, using 3V ensured a strong laser that would definitely turn the phototransistor on. The 3V would come from a battery.



The above diagram represents the photo transistor circuit. The input voltage for the phototransistor is 6V because the Arduino only reads input values as HIGH if the voltage is at least 3V. A 3V battery would therefore not be enough for the Arduino. Also, we put in an LED in order to tell when the laser is hitting the phototransistor. When the LED is dim, the laser is not hitting the phototransistor. The resistor value we chose (220 ohms) was also based on current and voltage values from the output wire to the Arduino pin. The above setup allows the voltage to the pin be below 3v but significantly above when the light hits the phototransistor. The 6V would come from two 3V batteries that are hooked up in series (see Diagram One in the Appendix). Instead of having a 9v battery and a voltage divider to create the necessary 6V, we decided on hooking these batteries in series because having a voltage divider would add extra wires to the breadboard, which can interfere with the diodes.



The above diagram represents the circuit for the frets. Vin is 5v because the Arduino has a 5v out pin that could be easily put on a rail for the breadboard with the frets. 5 volts also ensures the input pins on the Arduino would get the necessarily voltage (minimum 3 volts).



The above diagram is the MIDI output. We used the specifications on the Arduino website. Arduino pin one outputs the MIDI notes, which the MIDI jack receives in MIDI pin five. Note: the 220 ohm resistor is hooked up the 5V pin on the Arduino.

For the design of the body of the guitar, we were thinking about making the laser guitar as realistic as possible by making a full size guitar. However, this presents several logistical problems, such as laser cutting the acrylic pieces large enough for the body. Instead, we decided on making an acrylic base that lays flat. The breadboards for the lasers and phototransistors would be vertical on the base. Using a base greatly decreased the size of the guitar and made it easier to fix than a full size guitar since each breadboard was very accessible. Furthermore, the base design made the guitar relatively easy to play since the frets and lasers are close to one another.

3)Testing/Debugging

Before we soldered everything up, we tested the Arduino code by attaching a laser and fret to different Arduino pins to see if the Arduino would output MIDI notes. This was successful, but we did not test having all the lasers and frets hooked up to the Arduino, which ended up being problematic.

After soldering all the necessary components to breadboards and assembling the laser guitar, we tested the laser guitar and found a few problems with our previous design. First, the fret system would not work even after we looked over the Arduino code several times. The problem with our initial design circuit is hooking up the Arduino 5v pin and ground pin to the fret breadboard would cause all the current to flow from the Arduino to the circuit that was created from the pushed button. This effectively turned the Arduino off because all its power would go through this new circuit. Unfortunately, we did not have enough time left to take apart the laser guitar and fix this problem, so we decided to use each fret as a switch. Whenever one of the lasers would be broken, the Arduino would sustain the note. Since the fret would effectively turn off the Arduino, we could use the frets to cut off the notes. Second, we noticed that lining up the lasers with the phototransistors was also difficult. When soldering the the lasers and phototransistors, we made the phototransistors flush with the breadboard, but we had trouble making the lasers flush. When originally soldering the lasers, we had clipped the laser wires too short, which made it difficult to solder the lasers to make them flush. To fix this, we took out the old lasers and resoldered new lasers, except we fed more of the wire through the breadboard to make the lasers flush. Third, we experienced a lot of lag. When we broke a laser, the note would not play for a few seconds. To fix this, we simplified the Arduino code by taking out the fretting system we previously had.

PARTS LIST

Part	Price Per Unit	Quantity	Total Cost	Website Link
Laser Diode (Lab)	2	6	12	http://www.amazon.com/Ciamlir-Laser-Module-Sight-Pointer/dp/B00MYUAJ8M/ref=sr_1_1?ie=UTF8&qid=1418497570&sr=8-1&keywords=laser+diode
MRD360 phototransistor (Lab)	2.89	6	17.34	http://www.alltronics.com/cgi-bin/item/MRD360/63/Motorola-Darlington-MRD360-Photo-Transistor
Arduino Uno (Lab)	14.41	1	14.41	http://www.amazon.com/Arduino-UNO-board-DIP-ATmega328P/dp/B006H06TVG/ref=sr_1_1?ie=UTF8&qid=1418497709&sr=8-1&keywords=arduino+uno
LED’s (Lab; any color is fine)	0.17	6	1.02	http://www.amazon.com/microtivity-IL142-Diffused-Blue-Resistors/dp/B0059H5Z5O/ref=sr_1_4?ie=UTF8

				&qid=1418497769&sr=8-4&keywords=blue+LED&pebp=1418497777326
Solderable Breadboard	**3.99**	**3**	**11.97**	**http://www.amazon.com/Vktech-Double-Side-Prototype-Universal-Printed/dp/B00COGNWXA/ref=sr_1_sc_4?ie=UTF8&qid=1418497831&sr=8-4-spell&keywords=solderablebreadboard**
12 X 12 Acryllic 1/8th inch thick	**8.99**	**2**	**17.98**	**http://www.amazon.com/Source-One-LLC-Acrylic-Plexiglass/dp/B004DYW31I/ref=sr_1_2?ie=UTF8&qid=1418497886&sr=8-2&keywords=acrylic+sheet&pebp=1418497887251**
MIDI USB cable	**4.60**	**1**	**4.60**	**http://www.amazon.com/HDE-Cable-Converter-Music-Keyboard/dp/B001LJUVO4/ref=sr_1_1?ie=UTF8&qid=1418497959&sr=8-1&keywords=Midi+cable+usb&pebp=14184979584**

				97
Female MIDI jack	**5.69**	**1**	**5.69**	**http://www.amazon.com/Neutrik-NYS325-Female-Chassis-Connector/dp/B00MEI42PU/ref=sr_1_1?ie=UTF8&qid=1418497998&sr=8-1&keywords=Midi+jack&pebp=1418498000056**
Pushbuttons (lab)	**0.46**	**4**	**1.84**	**http://www.amazon.com/microtivity-IM206-6x6x6mm-Tact-Switch/dp/B004RXKWI6/ref=sr_1_3?ie=UTF8&qid=1418498064&sr=8-3&keywords=push+button&pebp=1418498066185**
100 Ohm Resistor (lab)	**0.05**	**6**	**0.3**	**http://www.amazon.com/E-Projects-Resistors-Watt-100R-Pieces/dp/B00BVCCA90/ref=sr_1_1?ie=UTF8&qid=1418498152&sr=8-1&keywords=100+ohm&pebp=1418498152093**
220 Ohm Resistor (Lab)	**0.05**	**6**	**0.3**	**http://www.amazon.com/E-Pr**

				ojects-Resistors-Watt-220R-Pieces/dp/B00BVCCCS4/ref=sr_1_1?ie=UTF8&qid=1418498215&sr=8-1&keywords=220+ohm
3 Volt Battery holder (lab)	**1.37**	**3**	**4.12**	**http://www.amazon.com/1-5V-Battery-Holder-Black-Leads/dp/B00E0IXRUS/ref=sr_1_1?ie=UTF8&qid=1418502619&sr=8-1&keywords=battery+holder+AA&pebp=1418502621676**
AA batteries (Lab)	**2**	**6**	**12**	**http://www.amazon.com/s/ref=nb_sb_noss_1?url=search-alias%3Daps&field-keywords=AA%20batteries**

Grand Total (Lab parts + Ordered Parts) = 103.57

PROJECT IMPLEMENTATION

Step 1: Testing

After creating our block diagram and circuit diagrams, we decided to hook up the circuits to a solderless breadboard. The first part we wanted to test was the MIDI jack. Use alligator clips to hook up the wires to the MIDI pins; trying to feed the wires through the pins can be quite difficult. When testing the MIDI, make sure the ends of the wires are not touching the other pins. This can occur quite easily. For software, use Garageband for Mac or Ableton Live 9 (free 10 day trial) for Windows. These programs can easily let you test your MIDI input and offer several different instruments to replicate (useful if you want to create another laser instrument).

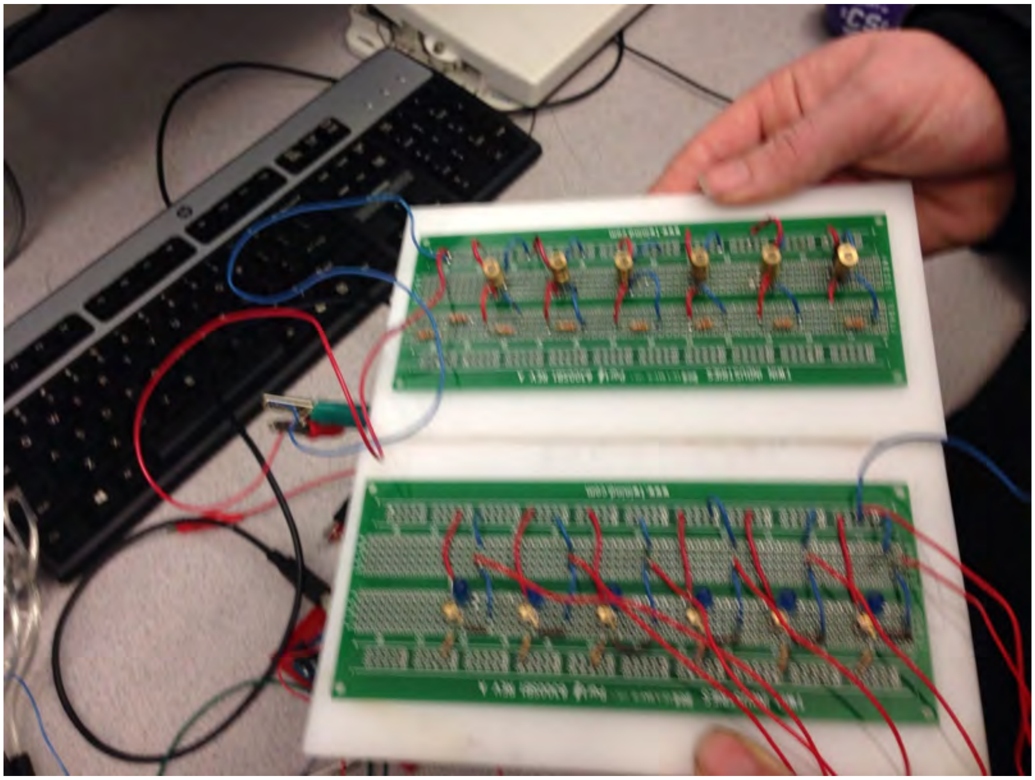
After getting the MIDI working, we tested the other circuits. The other circuits did not need to be amended. However, try to use the multimeter to make sure current is getting through the phototransistor when the laser is shined on the phototransistor. The photo transistor has specific pins that need to be hooked up to specific parts of the circuit. Always double check that the right pin is attached to the right part of the circuit.

During the testing stage, we also wrote the Arduino code and tested it, making changes whenever bugs appeared. For the Arduino, we noticed that uploading the code successfully requires no wires to be connected to the Arduino pin.

Step 2: Assembly

After testing came the assembly of the laser guitar. The first part of assembly is soldering all the circuits to breadboards. The first breadboard we soldered was the lasers. After feeding the wires and resistors through the holes, we tested the circuit when we applied voltage across the circuit. After soldering, we tested again. Make sure the lasers are flush as possible with the breadboard, as described in the previous sections. After soldering, test again. We also made sure the wires were as short as possible because long wires can stick up on the breadboard and interfere with the lasers. Finally, use the rails for the Vin and ground.

The second breadboard we soldered was the phototransistors. When putting the phototransistors through the holes in the breadboard, we had to make sure they were lined up with the lasers. Make sure the wires are short as possible in order to prevent the wires from breaking the laser. Test that there is current through the phototransistor by using a digital multimeter. Also, the wires to the Arduino need to be somewhat long in order to reach the Arduino when everything else is assembled. We accidently cut a few wires too short, so we had to solder additional wire to the short wire. The first two breadboards look like:

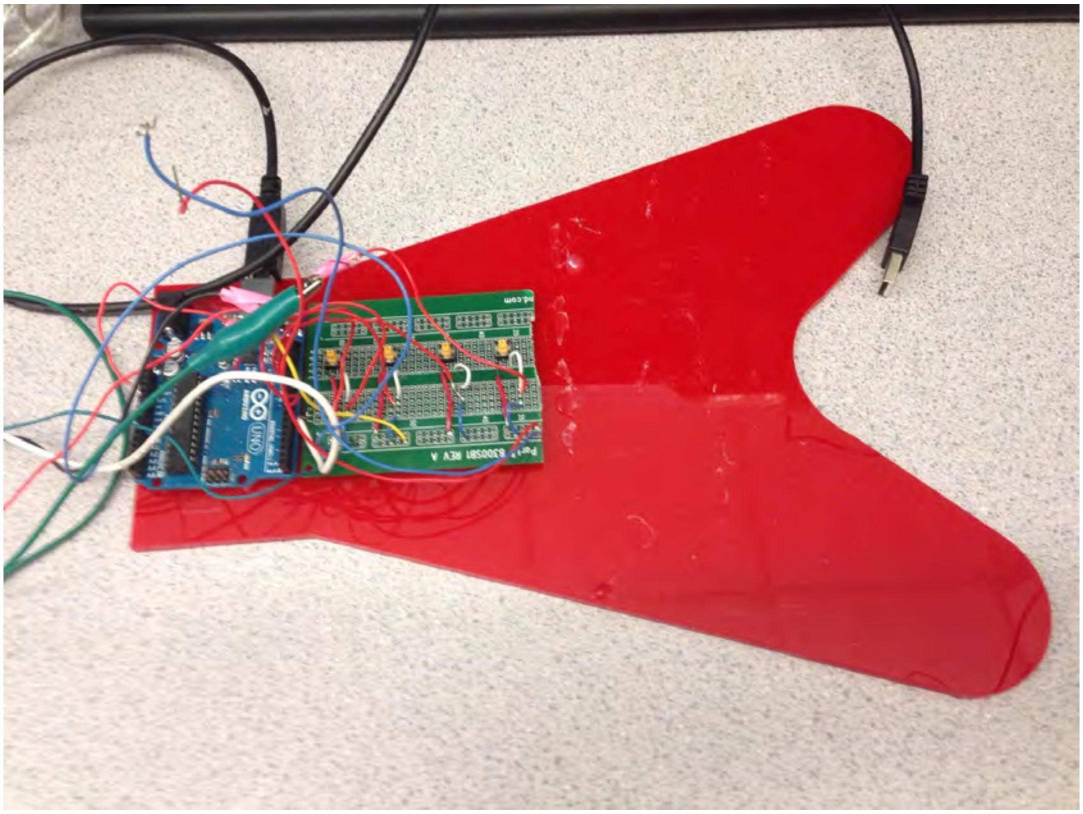


Alligator clips are useful for connecting the Vin/Ground wires to the batteries as opposed to soldering.

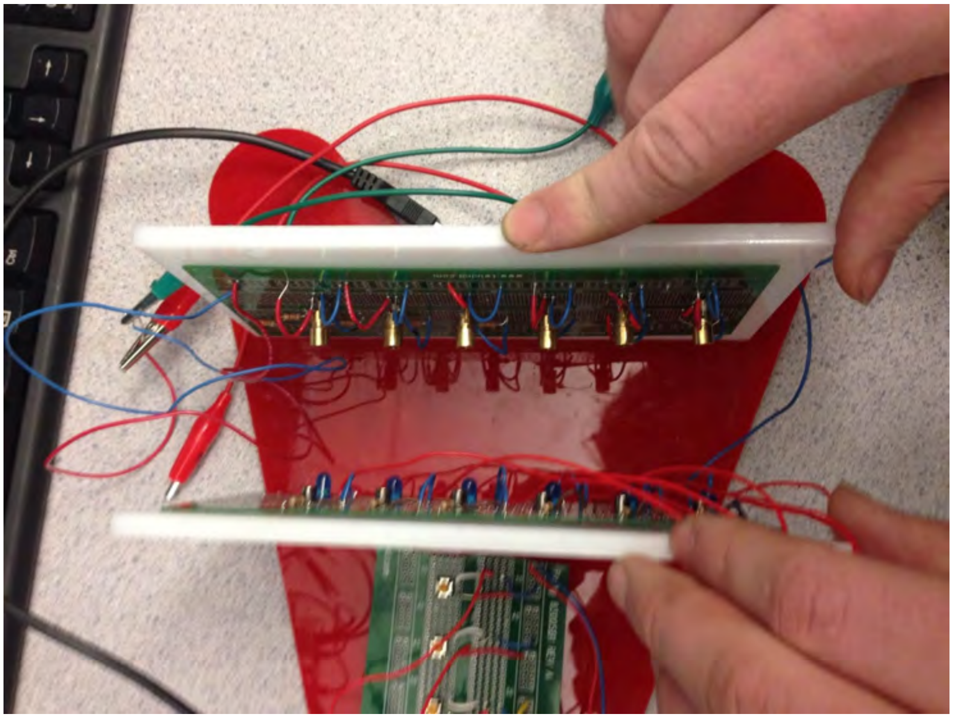
Next, we soldered the fretboard using the previous circuit diagram. After soldering this, we made sure all the wires from the phototransistors and the frets could be put into the Arduino easily. Make all of these input pin wires a little longer than necessary so the wires can be trimmed after everything else is put in place.

After all the soldering was done, we lasercut the the acrylic pieces for the body and breadboards. Ultimately, we decided on having a base and the breadboards vertical on the base. We had to cut three acrylic pieces: one for the base and two for each breadboard. Make sure the acrylic for the base is long enough to fit everything. For attaching the breadboards to the rectangular acrylic pieces, we decided on double sided tape since we could detach the breadboards easier than if we used hot glue. After placing one breadboard to the acrylic, try to line up the lasers/phototransistors as well as possible when attaching the breadboard to the other acrylic.

We then used double sided tape to adhere the Arduino and the fretboard to the "neck" of the base.

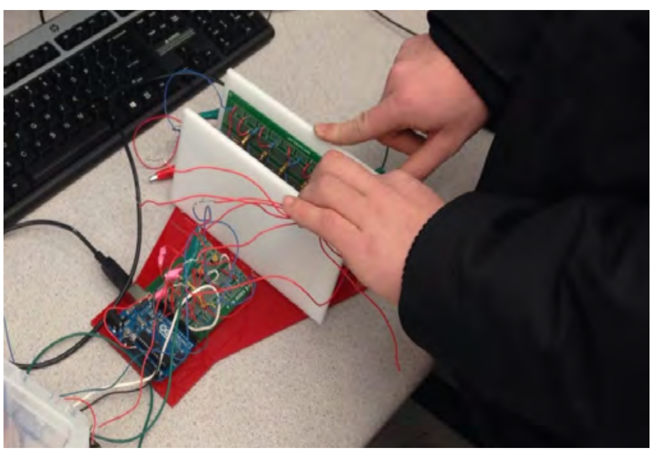


After this, we hot glued the acrylic pieces with the breadboards to the base. We decided on hot glue since it is easy to apply and somewhat stable. After hot gluing the pieces, one of the pieces fell over because we did not hold the acrylic in place to let the glue dry. Here is a closeup of the vertical breadboards:



We had to make sure the breadboards were close enough such that the lasers could reach the phototransistors, but not too close such that playing the guitar would be difficult.

The final product looked like this:



Step 3: Debugging and Fixing
After testing everything, one of the LED's burnt out. This was an easy fix because we could take off the breadboard, solder a new LED, then retape the breadboard back onto the acrylic. Most of the debugging came with the code, which we could fix on the computer.

TEAM MANAGEMENT

During the early stages of the project, both of us came together and thought out the basic ideas behind our project, which include how the code would work, the circuits, and the overall design of the project. After the basic design was created, Student B focused on the coding (CS concentrator) and Student A focused on building/soldering the various circuits. Doing this allowed each person to focus on one part of the project as opposed to both of us trying to do everything. During the debugging process, we came together and worked on the code and hardware together. We found our team management to be quite efficient, and one person did not have to do all the work.

OUTLOOK AND POSSIBLE IMPROVEMENTS

There are three improvements we would have made if we had more time/resources:

1 Fixing the frets

If we had more time for the project, we could have revisited the circuits for the frets and made the fretting system to work. We do not think the code was the problem, but the circuit design. This would require us to resolder a new breadboard and test it.

2 Creating a MIDI software program

This is a more ambitious and CS related project, but it would be interesting to have a program that can play WAV files of guitar sounds based on the MIDI input. The sounds produced in Ableton 9 were very synthetic, which was fine for the time/resources allocated to us.

3 Implementing guitar effects

Popular guitar effects include distortion and wah-wah. This could be implemented by having another circuit whose input is the sound from the computer and provides output to a speaker.

ACKNOWLEDGEMENTS:

1) Marko/Chris - They were very helpful with proposing new ideas for our design and for debugging.
2) Xuan - Able to find a lot of parts we needed but did not order
3) Jessica provided guidance during the early stages of our project
4) Will was helpful with debugging

DISCLAIMER

It is ok to share this information.

REFERENCES:

1) **For Arduino-MIDI setup.: http://arduino.cc/en/tutorial/midi**
2) **Software: Ableton Live 9 https://www.ableton.com/en/live/new-in-9/**
3) **For Arduino-MIDI coding:https://itp.nyu.edu/physcomp/labs/labs-serial-communication**

/lab-midi-output-using-an-arduino/

4) **Lab 8 for laser circuits and info about phototransistor**

APPENDIX

1) **Table One**

Example of implementing frets for low e-string on the guitar:

Note: 0= no current going to an Arduino digital pin. 1= current going to Arduino digital pin.

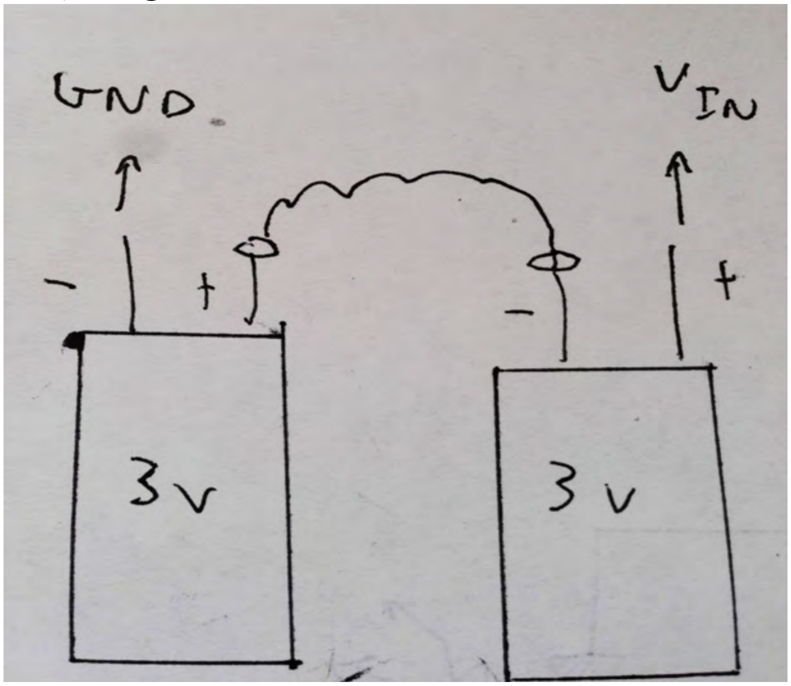
Fret Four	Fret Three	Fret Two	Fret One	Equivalent Guitar Fret	Note (1 indicates lower octave)
0	0	0	0	Open String	E1
0	0	0	1	One	F1
0	0	1	0	Two	F#1
0	0	1	1	Three	G1
0	1	0	0	Four	G#1
0	1	0	1	Five	A1
0	1	1	0	Six	A#1
0	1	1	1	Seven	B1
1	0	0	0	Eight	C1
1	0	0	1	Nine	C#1
1	0	1	0	Ten	D1
1	0	1	1	Eleven	D#1
1	1	0	0	Twelve	E2
1	1	0	1	Thirteen	F2
1	1	1	0	Fourteen	F#2

1	**1**	**1**	**1**	**Fifteen**	**G2**

2) Table Two

O.1V , 0.004 micro Amps
0.2V: 0.004 micro Amps
0.3 v: 0.004 micro Amps
1.1 v: 0.007 micro Amps
1.20 v: 0.02 micro Amps
1.3 V:0.11 micro amps
1.4 v: 0.69 micro amps
1.5 v: 4.14 micro amps
1.6 v: 17.9 micro amps
1.7 v: 53.2 micro amps
1.8 v: 0.114 mili amps
1.9 v: 0.320 mili amps
2.0v: 0.543 mili amps
2.1v: 0.789 mili amps
2.2v: 1.045 mili amps
2.3v: 2.81 mili amps
2.4 v: 3.43 mili amps
2.5 V: 4.05 mili amps
2.6 v: 4.67 mili amps
2.7 v: 5.03 mili amps
2.8 v: 5.94 mili amps
2.9 v: 6.58 mili amps
3 v: 7.22 miliamps

3) **Diagram One**



4) Arduino Code

```
int laserone = 2; //low e string
int lasertwo = 3; //a string
int laserthree = 4; //d string
int laserfour = 5; //g string
int laserfive = 6; //b string
int lasersix = 7; //high e string

void setup(){
 Serial.begin(31250);
 pinMode(laserone,INPUT);
 pinMode(lasertwo,INPUT);
 pinMode(laserthree,INPUT);
 pinMode(laserfour,INPUT);
 pinMode(laserfive,INPUT);
 pinMode(lasersix,INPUT);
}

void loop(){
 int estring = 0;
 estring = digitalRead(laserone);
 int astring = 0;
 astring = digitalRead(lasertwo);
 int dstring = 0;
 dstring = digitalRead(laserthree);
 int gstring= 0;
 gstring = digitalRead(laserfour);
 int bstring = 0;
 bstring = digitalRead(laserfive);
 int Estring = 0;
 Estring = digitalRead(lasersix);

 if (estring == LOW && astring == HIGH && dstring == HIGH && gstring == HIGH &&
bstring == HIGH && Estring == HIGH) {
	int note = 0x34;
	noteon(0x90, note, 0x45);
	delay(100);
	noteon(0x90, note, 0x00);
	delay(100);
 }
 else if (estring == HIGH && astring == LOW && dstring == HIGH && gstring ==
HIGH && bstring == HIGH && Estring == HIGH) {
	int note = 0x39;
	noteon(0x90, note, 0x45);
	delay(100);
```

```
      noteon(0x90, note, 0x00);
      delay(100);
  }
  else if (estring == HIGH && astring == HIGH && dstring == LOW && gstring ==
HIGH && bstring == HIGH && Estring == HIGH) {
      int note = 0x3E;
      noteon(0x90, note, 0x45);
      delay(100);
      noteon(0x90, note, 0x00);
      delay(100);
  }
  else if (estring == HIGH && astring == HIGH && dstring == HIGH && gstring ==
LOW && bstring == HIGH && Estring == HIGH) {
      int note = 0x43;
      noteon(0x90, note, 0x45);
      delay(100);
      noteon(0x90, note, 0x00);
      delay(100);
  }
  else if (estring == HIGH && astring == HIGH && dstring == HIGH && gstring ==
HIGH && bstring == LOW && Estring == HIGH){
      int note = 0x47;
      noteon(0x90, note, 0x45);
      delay(100);
      noteon(0x90, note, 0x00);
      delay(100);

  }
  else if (estring == HIGH && astring == HIGH && dstring == HIGH && gstring ==
HIGH && bstring == HIGH && Estring == LOW){
      int note = 0x4C;
      noteon(0x90, note, 0x45);
      delay(100);
      noteon(0x90, note, 0x00);
      delay(100);
  }

}

void noteon(int cmd,  int pitch, int velocity) {
   Serial.write(cmd);
   Serial.write(pitch);
   Serial.write(velocity);
}
```