PHYS364

Brandon Joel Gonzalez, Grace Ringlein

Final Project - 5/14/19

PennKey: bgonz, gring

The Premise

Good music is pretty great. Flashy lights are pretty fun. So why not put two and two together and make your own awesome custom dance party! We set out to create an audio filter system that could pick apart the treble and the bass in music and flash up lights accordingly. You see this a lot at concerts, so we thought it'd be cool to create this and understand the principles behind it!

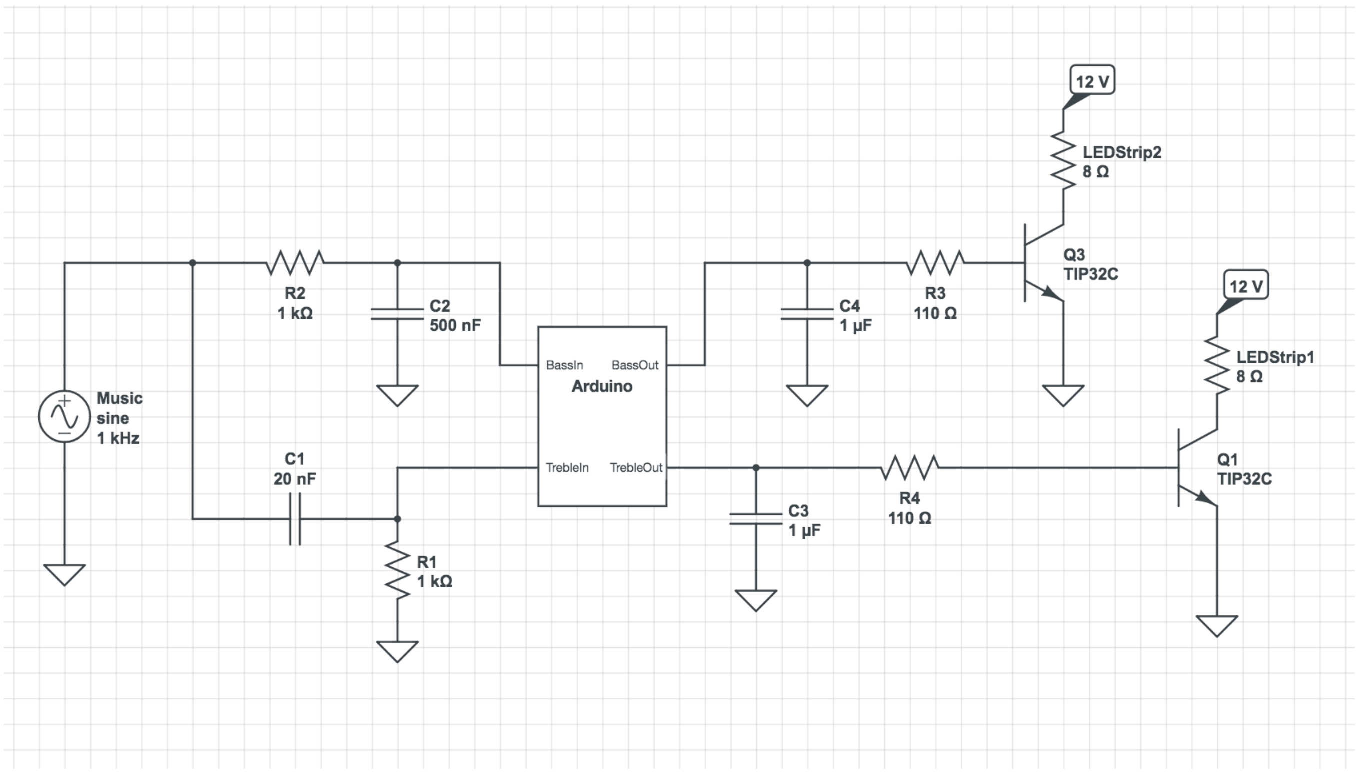
The Implementation

We used an audio splitter to feed the audio from a computer (or phone) into both our circuit and a speaker or headphones. The audio was then fed into both a high pass filter and a low pass filter. We used the low pass filter to select bass frequencies, which are approximately 300Hz and lower frequency. For this purpose, we chose an f_{3db} of 300Hz and thus selected a capacitor of .5 μF and a resistor of 1 kΩ. We used a high pass filter to select treble frequencies, of approximately 8kHz and higher, by choosing f_{3db} and then 20 nF and a resistor of 1 kΩ.

The two outputs of the filters were then fed into an Arduino. The outputs were HIGH (+5V) when there was a signal from the respective filter. This method was used to amplify the signal from the filters to properly drive the LED strips. These outputs were fed into a resistor and decoupling capacitor combination (capacitor of 1 μF, resistor of 110 Ω), and then into the base of a BJT. The audio signal from the playback device was about .3V which wouldn't be enough to reliably turn the BJT on, so we found it necessary to output a strong 5V signal.

When the bass or the treble in the song was present, the Arduino would output this high signal to the base of the BJT, which would switch it on. This would let current flow from the collector to the emitter. Right above the collector was an LED strip with its negative terminal connected to the collector and its positive terminal connected to a 12V source. Thus, when the BJT was on, current flowed from the 12V source, through the LED strip, and then through the transistor. Thus, we could light the LED strips based on the presence of bass and treble frequencies in our audio signal.

Here's a schematic of our circuit:



And here's the code we ran on our Arduino:

```
int bassInput = A0;
int trebleInput = A1;

int bassOutput = 2;
int trebleOutput = 3;

void setup()
{
  pinMode(bassInput, INPUT);
  pinMode(trebleInput, INPUT);

  pinMode(bassOutput, OUTPUT);
  pinMode(trebleOutput, OUTPUT);

  Serial.begin(9600);
}

void loop()
{
  // clear outputs
  digitalWrite(bassOutput, LOW);
  digitalWrite(trebleOutput, LOW);

  // check bass and treble input voltages
  int bassIn = analogRead(bassInput);
  int trebleIn = analogRead(trebleInput);

  Serial.print("Bass voltage: ");
  Serial.println(bassIn);
  Serial.print("Treble voltage: ");
  Serial.println(trebleIn);

  // trigger LEDs if bass or treble is high
  if (bassIn >= 5)
  {
    digitalWrite(bassOutput, HIGH);
  }

  if (trebleIn >= 5)
  {
    digitalWrite(trebleOutput, HIGH);
  }
}
```

The Results

Our circuit produced the desired output, where high frequency treble beats corresponded with LEDs illuminating on the lower strip, and low frequency bass beats corresponded with LEDs illuminating on the upper strip. Overall, we were quite satisfied with the results of our project!

Below are some photos of our circuit: