Building the Stepper Motor
Pi Arduino Music Machine

Kellon Sandall
kgs256@byu.edu

Levi Powell
gideonpowell99@gmail.com

Sterling Smith
sterling.smith.stl@gmail.com

Abstract

We build a stepper motor music machine with a Raspberry Pi and an Arduino.

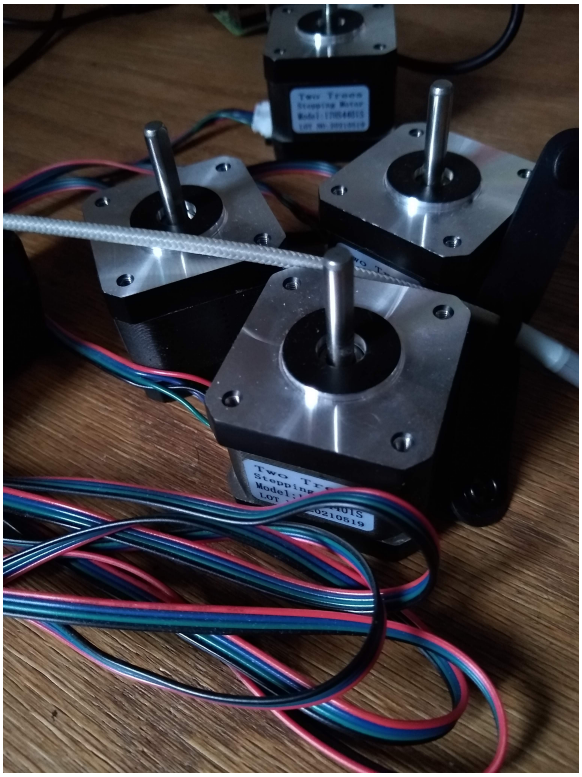
1 Description

Stepper motors are have many interesting properties that set them apart from other types of motors. One of the important properties of stepper motors have is the ability to turn at precise rotations and speeds. If I send the signals to a stepper motor to turn exactly a quarter revolution, it will turn exactly a quarter of a revolution. This makes them especially useful in 3d printers where precision is critical.

A side effect of the motors spinning is vibrations. When a stepper motor is spun at a specified speed, sound is generated. We can take advantage of this and use stepper motors to create music. The main challenge is hooking them together in such a way that we can play music.

2 Beginnings

I first found out about stepper motor music machines through Youtube. They were so intriguing to me that I decided to create my own. I first bought the stepper motors, but I realized that I needed to connect them to some device. I considered using a Raspberry Pi but, after some research, found that an Arduino with a 3d printer CNC shield would be the easiest route. I got an Arduino and CNC shield and hooked them up.



I used some code built by Jonathan Kayne to run the Arduino. The code turned the Arduino into a MIDI device that could accept MIDI signals. I figured out how to open a MIDI port on my PC and connect a MIDI editor to play through that port. Once I had achieved that, I was able to hear the motors, but they didn't have any resonating chamber, so I couldn't hear them well. So I thought for a while how I would fix this.

3 Forming the team

When I heard about this competition, I thought it would be a cool idea to make the machine interactive. My vision was to have a way that people could choose which song they wanted to play with their phones. This started getting complicated, so I thought it would be a good idea to get a team together to be more efficient. So I got my friends Levi and Sterling to join me.

We thought of various ways that we could get phones to communicate with the machine. We came up with ideas of making apps and maybe using bluetooth. I tasked us all with finding the best/easiest way to get the communication established.

4 Coding

Since I had some experience with node js, I decided to try out creating a HTTP server. Since bringing around a laptop everywhere with the stepper motors would be cumbersome, I decided to hook the Arduino up with a Raspberry Pi. I decided to build a centralized server to which people could connect their phones and to which I could could connect the stepper motor machine. This way, I could bring the stepper motor machine wherever I want and people could be wherever they wanted to as well.

This process consisted of much trial and error. I tried using a prebuilt HTTP solution called Express which is fairly popular in the node js community, but I couldn't get the data from the machine to the server and then to the phone like I wanted to. So I decided to build my own HTTP server. I learned the HTTP protocol and was able to create a REST api which could control the stepper motor machine.

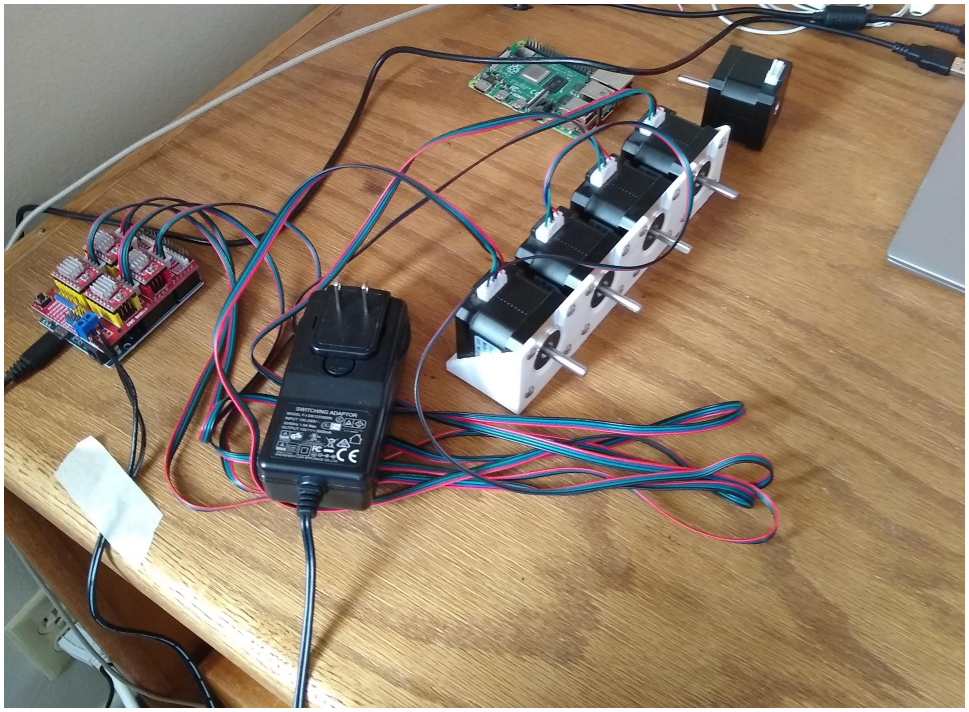
Levi and Sterling weren't able to get another communication solution working, so we went my solution.

One other interesting aspect of the coding of this machine was how to get MIDI signals from the Raspberry Pi to the Arduino. Initially, I thought that it wouldn't be too hard to write code to do this in javascript. However, when I played the MIDI files through javascript with node js, the motors wouldn't spin correctly quite often. I tried a couple different solutions but ultimately settled on using an external pre-written program to send the MIDI signals. Then it ran much smoother.

5 Obtaining materials

Then came the challenge of getting the materials for the project. I happened to find this box which ended up being the perfect fit for the stepper motors.

Originally, I had planned to get a metal beam and drill holes in it to mount the stepper motors on. However, this was proving difficult, so I found a 3d schematic of a stepper motor holder from Jonathan Kayne and used it to print the holder. I got it printed and it worked quite nicely.



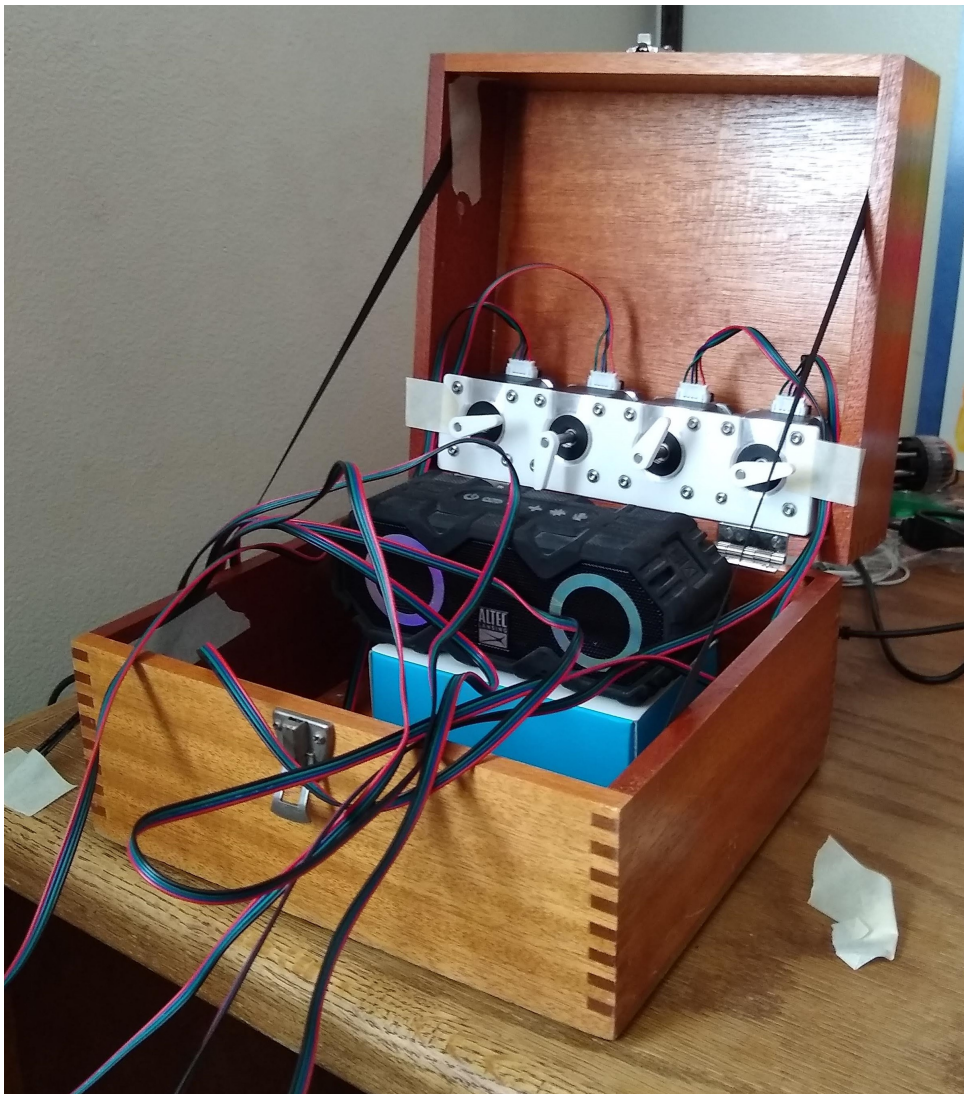
Now that I had these key components, it was time to put everything together.

6 Bringing everything together

A couple of critical issues arose when I tried to run the stepper motors. When I initially played songs, it would start well, but after waiting for a while, the stepper motors would produce this high pitched whine when starting again. It took some digging, but eventually I found the solution. Other people had had the same problem and I used some code that they created to fix it.

Another problem that arose was that at certain frequencies, the stepper motors would "stutter" and not move correctly. This also took some digging around, but eventually I found a solution which consisted of "microstepping" the motors. Basically, this means making the motors move at more precise rotations. After this fix, they moved much better.

After overcoming these problems, it was time to put everything into the box.



I decided the best place for the motors would be right inside the lid of the box. This would help me avoid drilling a hole through the box which would have had to have been very large to fit plugs through.

Then with lots of tape and faith, I got everything to fit inside.

I got an external server running to accept connections from phones and the machine and got the machine script to run on startup and now we have what you see today. Overall, this project taught me how to get things communicating with each other, how different protocols work, and how to interpret different signals that are sent.

References

[1] Kayne, Jonathan (2018, April 22). Arduino MIDI Stepper Synth. Arduino Project Hub. https://projecthub.arduino.cc/JonJonKayne/162864dd-a994-448c-ad28-624104fe28e9