Fotenie Apostolo
Electronic Musical Instrument Design
December 19, 2022

EMID Final Project Report

Design Goals

Our project is a marble sequencer. It is a sequencer that visually represents the music that is being outputted. This tool can be used by live musicians and in production environments as well as by individuals creating music for fun. The shape is a circle to visually resemble how the measure is repeated in a pattern over and over again unless it is changed or stopped.

The two knobs control the instrument sound and the tempo of each measure. One button controls the start and stop of the music and the other provides a loop option. This allows the user to individually control the baseline and the melody with the sequencer.

The eight holes around the outside of our sequencer represent a beat in two 4/4 measures. There are color sensors located below each hole. The sensors detect the color of the marble which correlates with a different note. If a marble is not present, a note is not played. Each color from the color sensor is represented by a different number ranging from 0-4 in the serial monitor of the Arduino software. This number is then transferred to Max and the Reason patches control the sound output.

Parts Used

Body:

- 2 living hinges, 3 mm birch
- 1 top plate, 3 mm birch
- 1 bottom plate, 3mm black matte acrylic
- 8 3D printed sensor mounts, PLA
- Adhesives (duct tape and epoxy)

Electronics

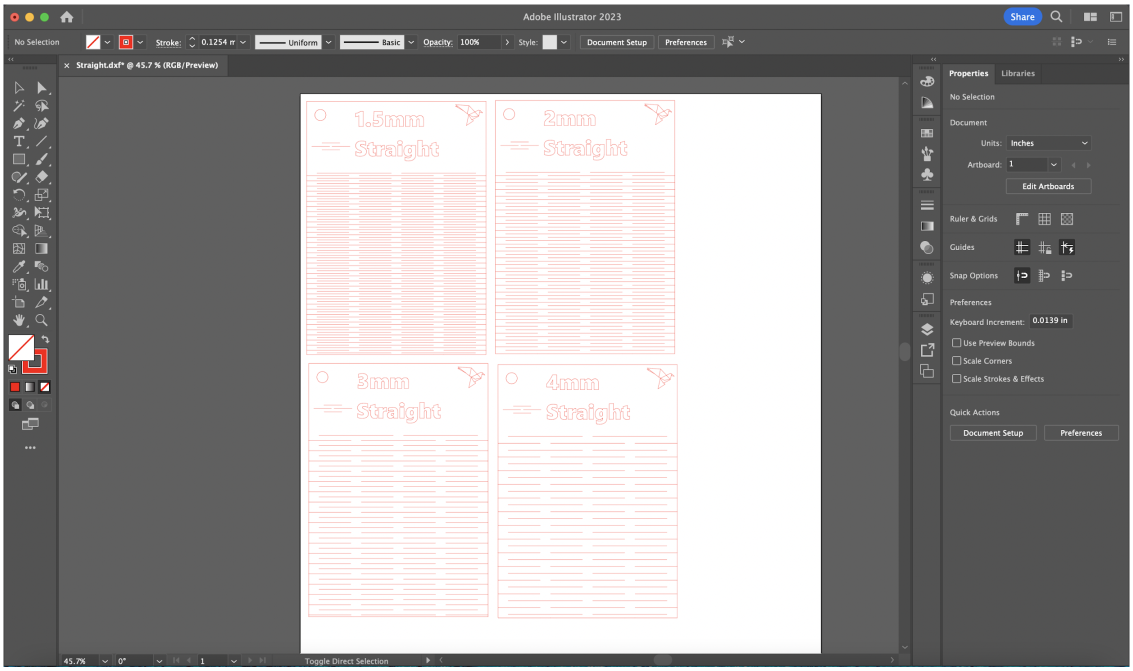
- 8 Adafruit TCS34725 RGB color sensors with white LED
- 2 arcade game buttons
- 1 Sparkfun 10-position switch
- 1 10k linear Adafruit potentiometer with knob
- 1 Arduino MEGA
- 1 TCA9548A I2C Adafruit multiplexer
- 3 resistors of various sizes
- 1 protoboard
- 1 breadboard
- Many wires

Other

- 3 blue marbles
- 3 red marbles
- 3 green marbles
- 3 black marbles
- 3 white marbles
- 1 yellow marble
- Paint, white-out, black sharpie

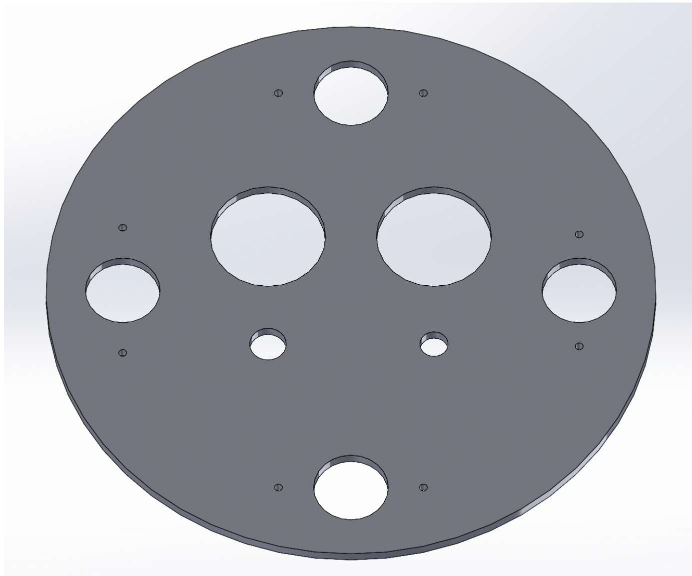
Construction

The construction process began with experimenting with living hinges. A laser cut living hinge is a piece with a ton of intricate cuts that allow the wood to bend in different amounts and different directions based on the cuts. I cut out all of the patterns below. The different thicknesses correlate with how much they can bend. If the lines are closer together, they can bend more.

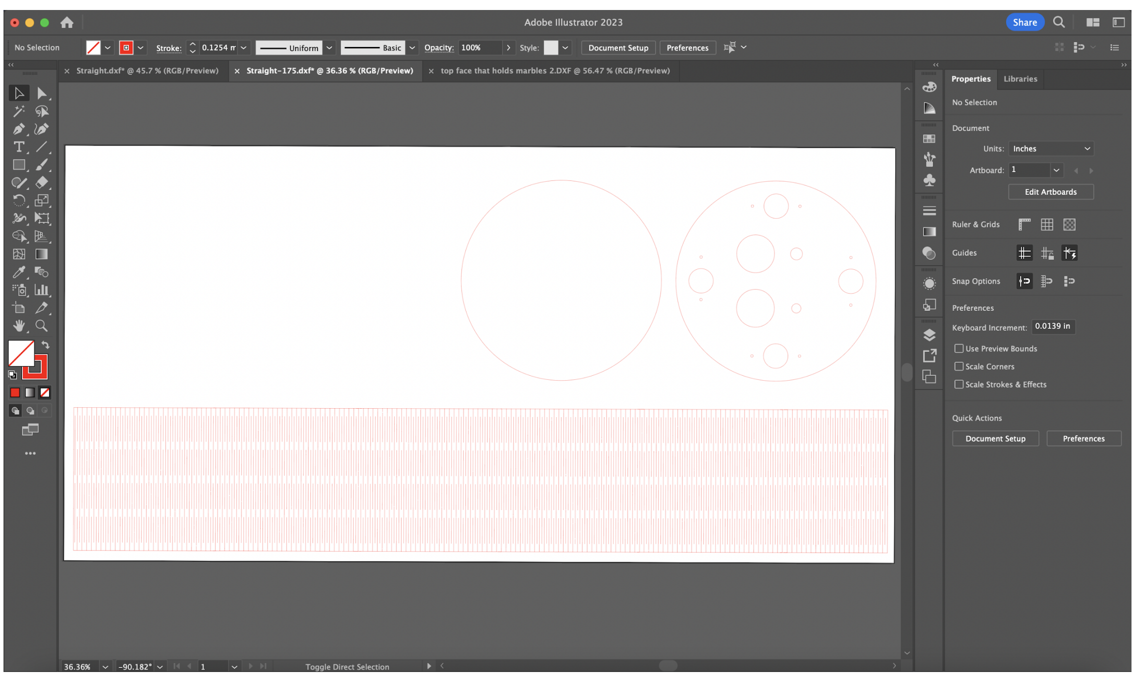


With these tests, I concluded that 1.75 mm thickness was most appropriate for this project because the larger thicknesses did not bend well around the approximate base size and would snap.

I cut out a rough example of a top of our instrument from blue semi-translucent dark blue acrylic for the purpose of testing our sensors with the marble. Once we decided on the sensor that we were going to use that we had the rest of our parts, I created the rest of the body of our instrument. The prototype was created using the laser cutter with 3 mm birch plywood. There were a few things that I had to consider when designing the structure. The first was the width of the hole for the marbles to rest on. They had to be small enough that the marble would not fall through and large enough that the marble would rest nicely and not roll off. I also had to consider the size of the hole and how it impacted how close the marble would be to the color sensor. I also had to use calipers to get the precise measurements of the buttons and switches so that they would fit nicely into their individual holes. The last consideration was the size of the body. We wanted it to be small enough to be portable, but large enough for everything to be spaced out and not cramped. Unfortunately, while the outside did not appear cramped, the inside did not have enough space for all of the wires.

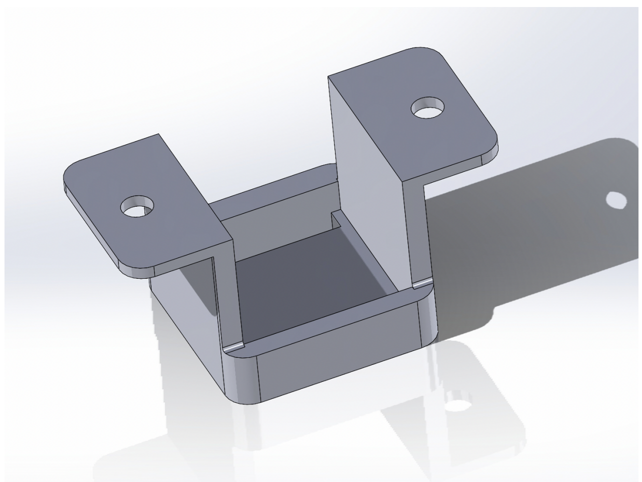


For the living hinge component, once I settled on the 1.75 mm distance between the lines, I just made the pattern as long as the board (~2 ft) so that I would not have to worry about it being too short to go around the board. Below is the final cut that I completed with the laser cutter in Nolop on 3 mm birch plywood.



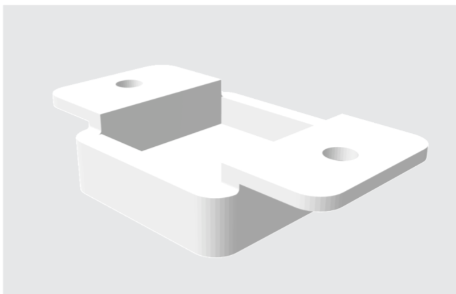
The body fit together nicely and I trimmed the living hinge to be the correct length using a saw. I also put the living hinge back in the laser cutter to cut out a box for the wires to exit or instrument and connect to the computer. I assembled the structure with two-part epoxy and waited for it to dry. I installed the sensor mounts with M1 screws, bolts, and washers into the holes that I cut out with the dimensions provided by measurements I took with calipers.

As with everything else I designed, I also designed the sensor mount in Solidworks. This took awhile because I didn't only have to measure the diameter of the object, I had to measure all components of it. I wanted the sensor to fit without needing to be secured so I designed a mount that was a press fit. The first design worked but some components made the sensor slide out easily so I went back to my CAD. In the next iteration, I re-measured some components and created the model that you see below. This worked very well because you had to use some force to get the sensor in but once it was in, it was not going to leave unless you made focused efforts to do so. I have a hole in the bottom for the pins to stick out and be easily wired to the Arduino. I also added arm-like attachments for mounting the sensor to the top of the instrument. It turns out that the arms should have been shorter but there was not enough time to print out more iterations by the time that we had put everything together. The mounts were printed out in Nolop using PLA material.

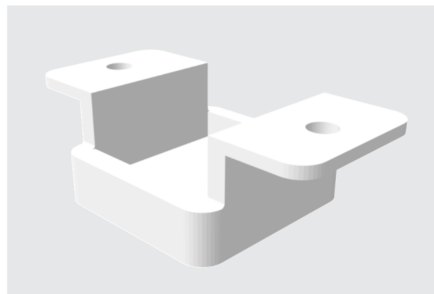


Below is the design of our prototype of the instrument.

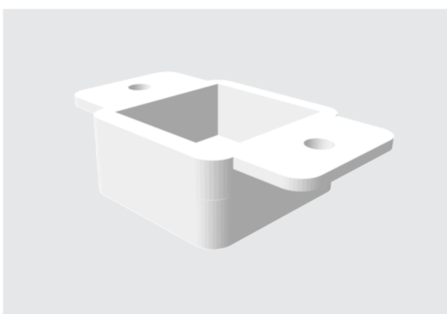
With the gained knowledge from our prototype, two main edits were made to the structure. The first was the sensor mounts. We suspected that the reason for the color sensors not getting accurate numbers was that the distance was too far between the marble and the sensor. I printed out 4 new mounts with varying distances between the sensor and the top plate of our instrument. These distances were 2.5mm, 5mm, and 7.5mm. I also suspected that there was possibly too much light from the sides affecting the readings so we also printed out a mount with walls that cover the side.



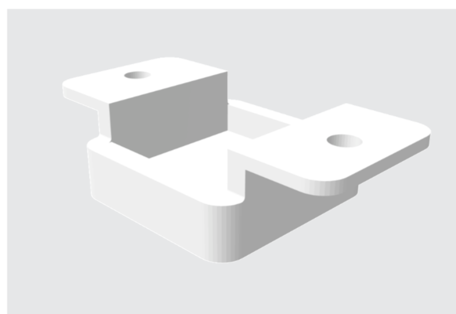
2.5 mm



7.5 mm



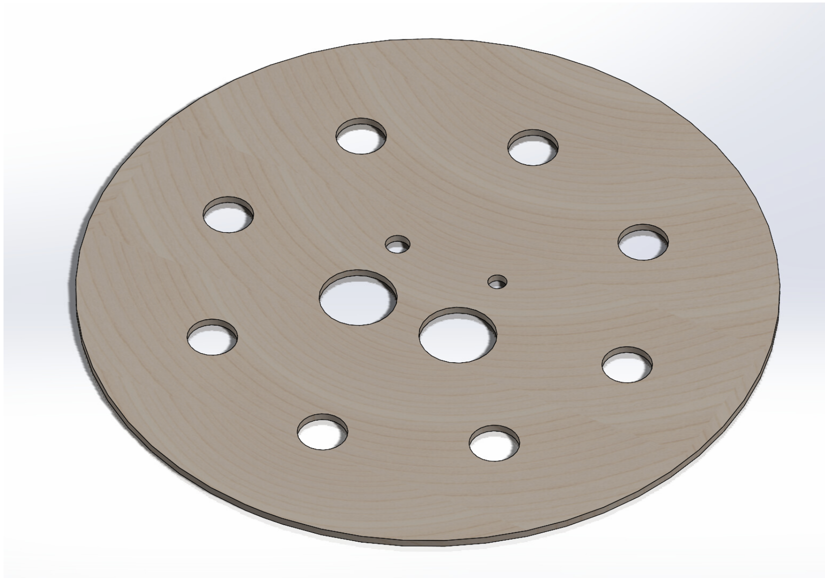
5 mm with side walls



5 mm

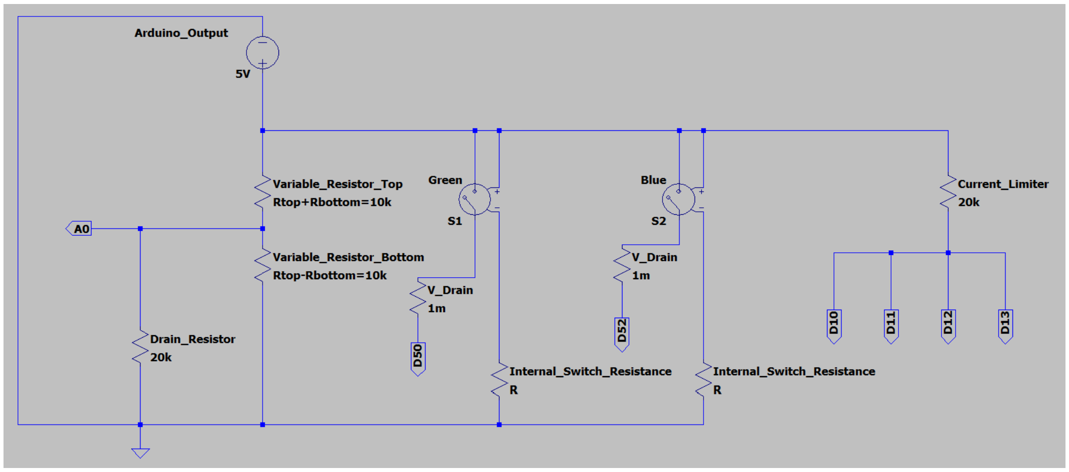
After several tests, we concluded that the 2.5mm distance sensors are best because we get the most dependable readings. The issue with these sensor mounts being form fit is that due the the width of the extrusion of PLA, the inside of the mounts needed to be filed down a bit to be able to perfectly fit the color sensor.

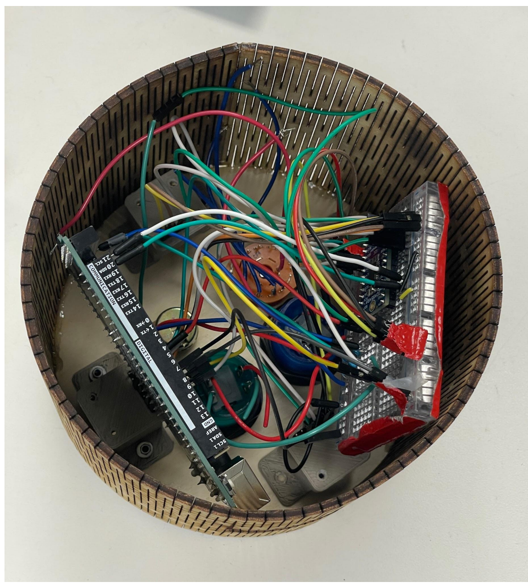
The second large change that was made was that we redesigned the top face of our instrument. I added four more holes so that we could have two 4/4 measures. I also increased the diameter to leave more room for easy installation of the wires. The final design is shown below. The top plate as well as the side of the instrument is 3mm birch that was lasercut in Nolop. The bottom plate is cut out of black matte acrylic and is a plain circle.



Below is the final physical prototype. The sensor mounts were installed using duct tape so that they could be adjusted if necessary after testing.

Circuitry

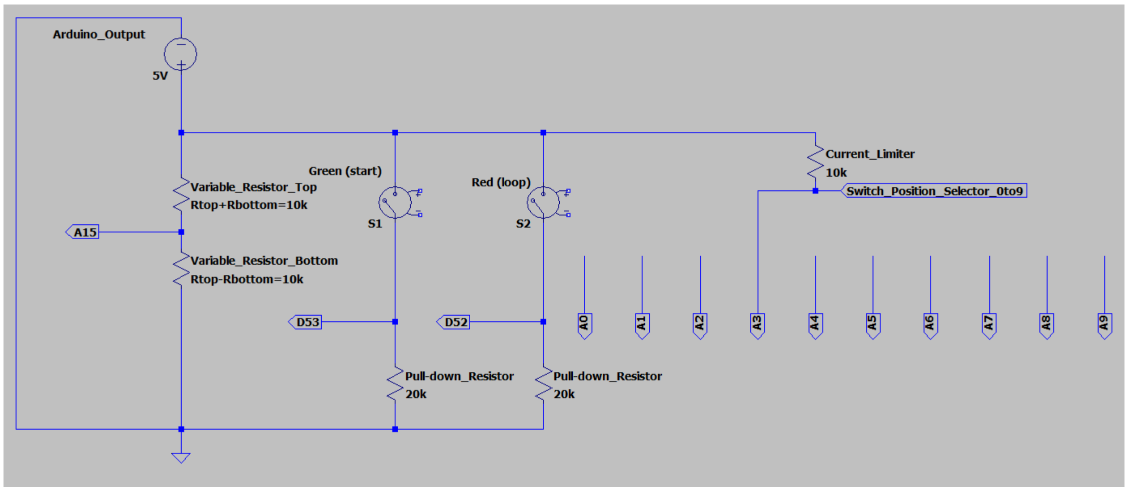


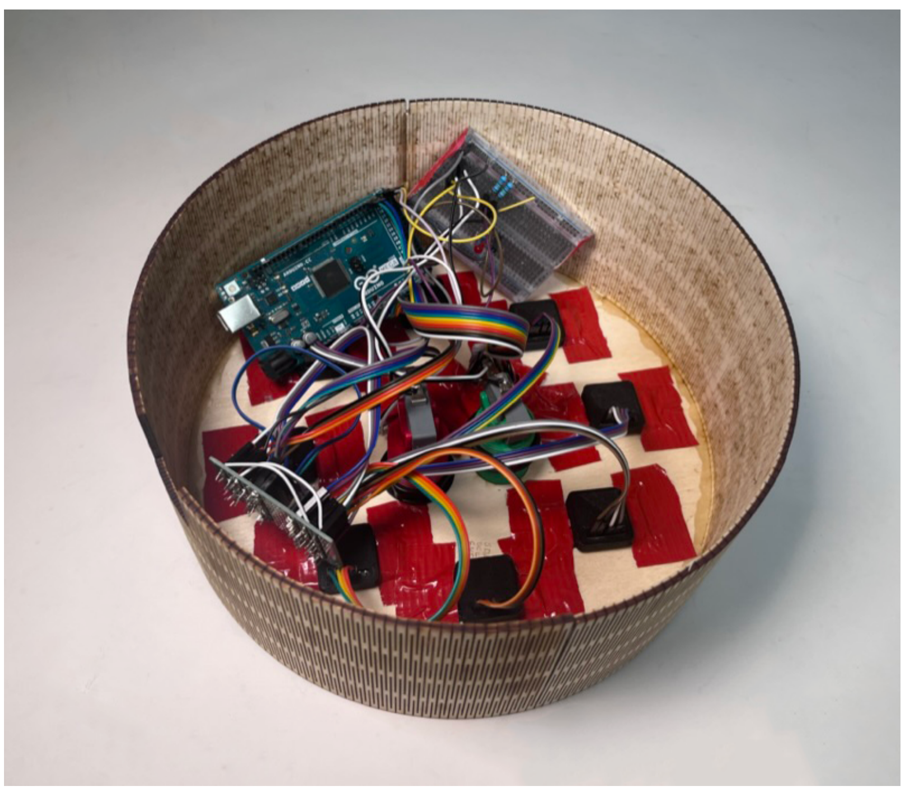


Above is the circuitry for the prototype

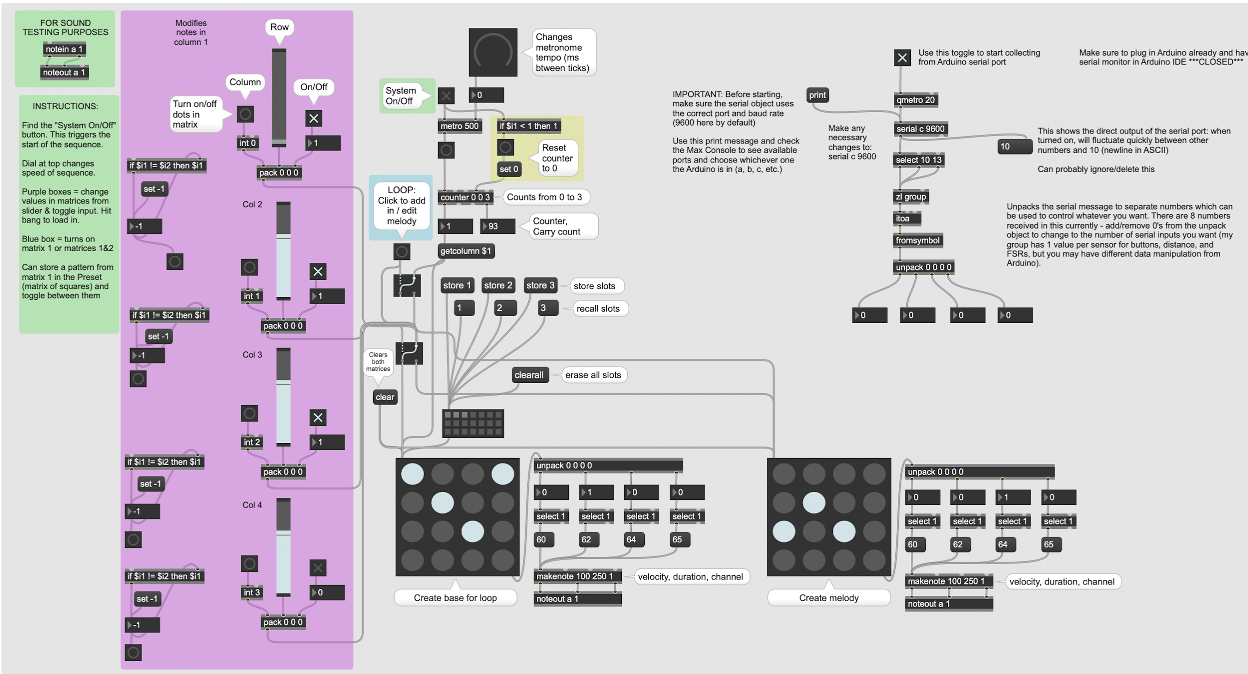
We began working on our project by ordering two color sensors. One color sensor was from Adafruit and the other was from SparkFun. We spent a lot of time testing out both sensors and figured out that the Adafruit sensor provided better results because it had an integrated LED. The switch and potentiometer are supposed to control the instrument sound and the tempo of each measure. The two buttons are supposed to control the start and stop of the music and the looping function. For the color sensors, it was difficult to implement all of them at the same time so we had to order a multiplexer which set us back on the timeline of our project. The multiplexer allows multiple connections to the SDA and SCL ports on the Arduino. While the color sensors could identify the colors in our testing phase, once we put the structure together only the green marble was reliably detected. We were able to get the potentiometer working but were unsuccessful in making the switch work due to space constraints. Also, we had difficulty getting the buttons to work, likely because there was an integrated LED which led to the buttons drawing in too much current. Because the buttons had internal capacitance, they did not output correct data and adding several resistors to attempt to stop this problem did not work. We ran out of time to get more of the electronics working on the prototype but in the future we will go through more design iterations to ensure that all of our electronics work.

After we presented our prototype, we reassessed and changed the electronics a bit although most stayed the same. We switched out the arcade buttons to ones that did not have LEDs and therefore internal capacitance. All of the pins for the electronics except for the buttons are analog. The buttons have pull-up resistors that help ensure the known state of the signal. The potentiometer is also technically inverted but an increase in the value increases the amount of time between the ticks of the metronome. Below is the final circuitry.



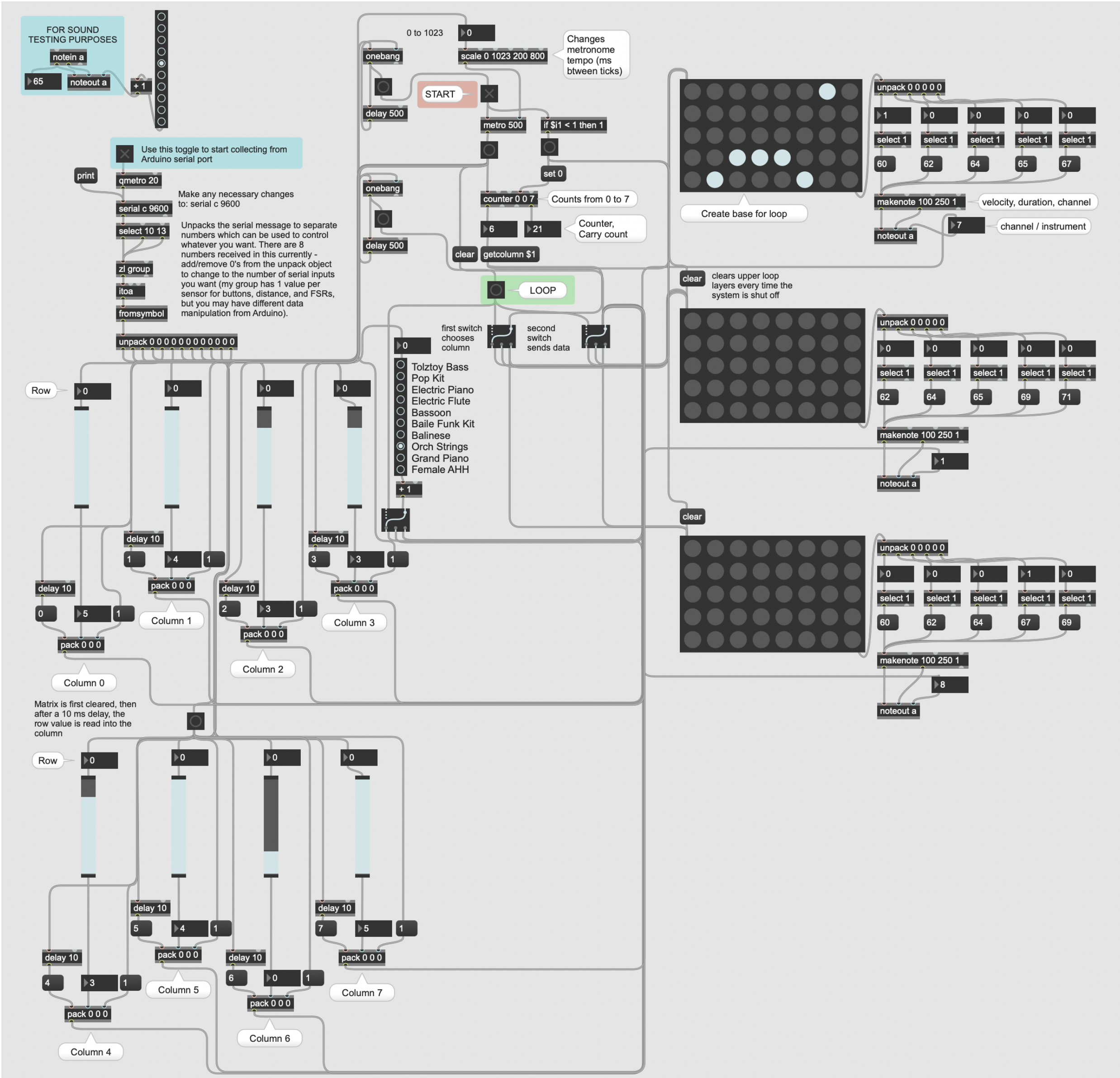


Functionality of Max Patch



Above is the Max patch for the prototype

The Max patch takes in the input (numbers from 0-7) from the Arduino output and sends it to Reason. When a person presses play, a metronome begins in Max. For the prototype, we were unable to figure out how to transfer the data from the Arduino to the Max patch despite having a similar setup to another group who was successful. We later learned that it was due to the fact that the different software needed to be opened in a specific order to allow for the data to transfer properly. Once the code is imported, it is interpreted and it is set to the matrices located at the bottom of the photo above. When the start button is pressed, the metronome begins to tick. Once the metronome is ticking, an instrument can be chosen using the switch on the instrument. There are 10 instrument choices.In the matrix, the columns represent each marble position and the rows represent each possible color of the marbles. At each beat from the metronome, a new column of the matrix would collect the new note to play. Depending on which note is selected, Max outputs the note to Reason to be played. The patch also offers a looping opportunity so that if you set up a base, you can switch to creating the harmony and then a melody which all play in a loop. Below is the final Max patch.

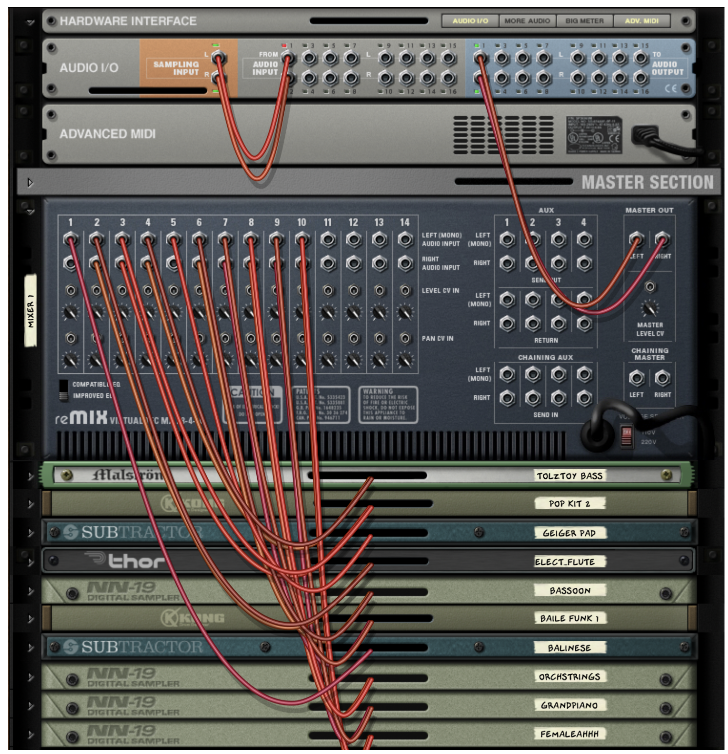


FOR SOUND TESTING PURPOSES
notein a
noteout a
Use this toggle to start collecting from Arduino serial port
print
qmetro 20
serial c 9600
Make any necessary changes to: serial c 9600
select 10 13
Unpacks the serial message to separate numbers which can be used to control whatever you want. There are 8 numbers received in this currently - add/remove 0's from the unpack object to change to the number of serial inputs you want (my group has 1 value per sensor for buttons, distance, and FSRs, but you may have different data manipulation from Arduino).
zl group
itoa
fromsymbol
unpack 0 0 0 0 0 0 0 0 0 0 0 0 0
0 to 1023
onebang
scale 0 1023 200 800
Changes metronome tempo (ms btween ticks)
START
delay 500
metro 500
if $i1 < 1 then 1
set 0
counter 0 0 7
Counts from 0 to 7
Counter, Carry count
clear
getcolumn $1
LOOP
first switch chooses column
second switch sends data
Tolztoy Bass
Pop Kit
Electric Piano
Electric Flute
Bassoon
Baile Funk Kit
Balinese
Orch Strings
Grand Piano
Female AHH
Create base for loop
unpack 0 0 0 0 0
select 1
makenote 100 250 1
velocity, duration, channel
channel / instrument
clears upper loop layers every time the system is shut off
Row
delay 10
pack 0 0 0
Column 0
Column 1
Column 2
Column 3
Matrix is first cleared, then after a 10 ms delay, the row value is read into the column
Column 4
Column 5
Column 6
Column 7

Reason Rack

Our Reason patch was fairly simple for this part of the project. We spent a lot of time concentrating on trying to get the connection between the Arduino and Max to work so we did not put a lot of effort into our Reason patch. The Reason patch for our prototype that we created was based on the NN19 Orchestra Strings sound. Anna expanded on the base sound to give it a short attack and decay time so that it would produce a strong sound.

Our final Reason rack has ten instruments that can be played by the sequencer. These instruments are Tolztoy Bass, Pop Kit 2, Geiger Pad, Electric Flute, Bassoon, Baile Funk 1, Balinese, OrchStrings, Grand Piano, and FemaleAHHH. Several of these instruments came from the NN19 rack but not all of them. These instruments can be changed using the switch. The different channels in the loop can be played by different instruments.



Discussion

We originally planned to have 16 color sensors and therefore beats but we scaled it down to 4 for the prototype and 8 for the final project. In the future, it would be great to expand the project with double the amount of beats and color sensors so that we could have 4 measures. In our final product, we made a stronger effort to make our instrument user friendly by indicating the order of the sequence with dots around the places where the marbles rest. We would also take steps so that the player does not have to look at the computer screen when playing our instrument. To do this, we would integrate LEDs for the buttons so that there would be visual cues on the instrument. An additional improvement would be to add more loops in the Max patch and the functionality to go back and forth between the loops as opposed to only being able to go through them in one direction. We would also like to have a total of 12 marble colors that could be recognized so that we can cover a full western scale.