ENAS 344/MUSI 415 Final Project

Jason Wang

I. Overview

This document covers the design and creation process of my instrument (see figure 1)[1]. The inspiration for the instrument comes from a melodica. I wanted to create an instrument with a similar control mechanism; namely, a wind-based instrument controlled using a keyboard, but I wanted to change the timbre of the melodica to something different. Being inspired by our recorder lab, I decided to use a whistle mouthpiece with a fipple as the sounding mechanism. This will be explained in further detail in the following section.

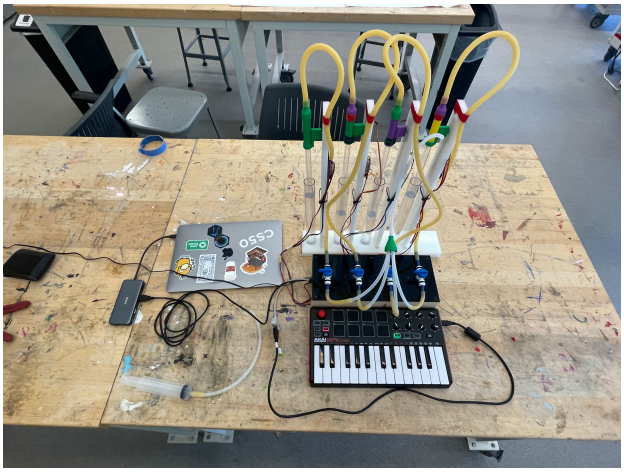

Fig. 1. An image of my instrument. It consists of four parts; a MIDI keyboard, a valve control system, four pipes, and (optionally) a computer as a secondary MIDI input source.

The instrument functions by using a MIDI keyboard (in the version I created, a MPK Mini MK2) to control the position of one of the four pipes. These pipes are numbered 1 through 4, left to right, and pressing pads 1 through 4 on the keyboard switches the control to that corresponding pipe. Pipes 1 and 4 correspond to the "soprano" pipes, and pipes 2 and 3 correspond to the "alto" pipes. Four valves are used to control airflow into each one of the pipes. These valves are then combined into one central mouthpiece, which is blown into to produce a note or chord, depending on how many of the valves are open.

The instrument needs to be tuned, and there are two ways to do so; one way is to by manually adding or removing water to raise or lower the pitch of the pipe. The water levels used for my performance were 8 1/2" above the upper surface of the base for pipes 1 and 4, 5.5" for the other two pipes, and this can be adjusted using a syringe with an attached pipe, as can be seen in the bottom left of Figure 1. The other way to tune the instrument is by using knobs 1 through 4 on the MIDI to sharpen the notes an incremental amount (by lowering the pipe a fixed amount for every note). These knobs can also be used to bend notes and play notes not found in a 12-tone scale.

Furthermore, the instrument can be connected to a computer and accept secondary MIDI input. Notes broadcast through channels 1 through 4 will control pipes 1 through 4, respectively.

II. Specifications

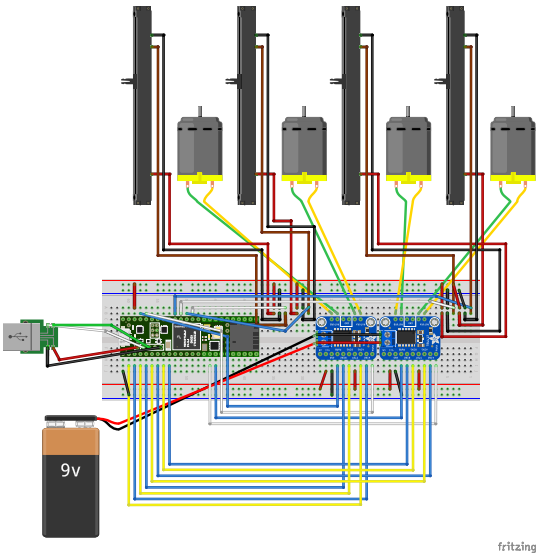

Fig. 2. A wiring diagram of the internals of the instrument. The 9V battery is a placeholder for the wall plug DC power supply used in the real instrument.

The instrument is mainly controlled by a Teensy 4.1 chip, connected to two Adafruit TB6612 motor drivers, each controlling two motorized slide potentiometers, which controls the movement of the pipes. These motorized slide potentiometers consist of two parts, a 10 kΩ slide potentiometer and a 5V to 10V DC motor.[2]

The instrument requires two power sources; a micro-USB port to power the Teensy 4.1, and a 9V DC power supply to power the motorized potentiometers. The micro-USB connection may also double as a connection for secondary MIDI input, as was done during the performance.

The pipes of the instrument consist of a eight polycarbonate pipes, CNCed plastic blocks, parts 3D-printed with PLA plastic, and surgical tubing to connect it to the rest of the instrument. 3D models of the white plastic parts, as well as an

[1]Larger versions of the figures used here can be found in the "Figures" folder included with this submission.

[2]The motor and the slide potentiometer are drawn separately in the wiring diagram (Figure 2), but they are connected in real life.

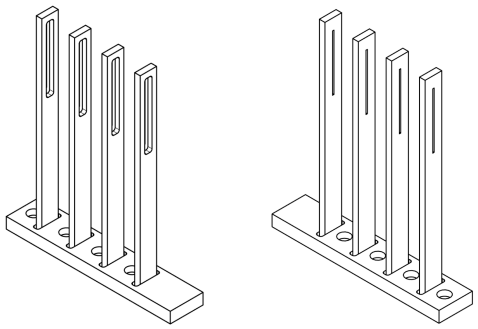

Fig. 3. A diagram of the base and the pipe holders, front (left) and back (right). The motorized potentiometers are inserted in the wider rounded rectangular insets, as seen on the left, so that the slider pokes through the thin rectangular hole, as seen on the right.

OnShape assembly and drawing of how these parts fit together, are included in the files accompanying this write-up as STL files. The white plastic parts consist of a 3"-by-15"-by-1" base and four pipe holders, each with dimensions 1 1/2"-by-17"-by-1/2". The pipe holders were press-fit into four rectangular machines holes in the base. The outer clear polycarbonate tubes, containing the water, had a 1" outer diameter and 7/8" inner diameter, and were cut to 10 1/2" in length. These were then also press-fit, this time into the circular machined holes machined in the base.

The inner clear polycarbonate tubes were metric, with a 16mm outer diameter and a 14mm inner diameter. For pipes 1 and 4, the "soprano" pipes, these were cut to 6 1/2" in length, whereas for pipes 2 and 3, the "alto" pipes, these were cut to 10 1/2" in length. These inner pipes were then fitted with a 3D-printed mouthpiece, then again fitted with an adapter for the surgical tubing and an adapter for the motorized potentiometer, all three of whose STL files are included in the files accompanying this write-up.

The valve control box, also containing the electronics, consists of four press-fit plastic valves centered, glued, and zip-tied 3" apart center-to-center atop a 5"-by-12"-by-1/8" laser-cut black acrylic panel. This panel was then screwed onto two wood planks, which was set atop a piece of cardboard. In the future, the cardboard bottom can be replaced with another piece of acrylic for increased durability.

Each of the motorized potentiometers was press-fit into a pipe holder, with the motor facing down, which was then secured in place with a few dots of hot glue. The wires leading into the potentiometers were also zip-tied onto the corresponding pipe-holder for added stability.

Each of the pipes was connected to its corresponding valve by attaching about 24" of surgical tubing to the adapter on the pipe, then fitting the other end of the tube over about 2" of plastic tubing, which was push-fit in the valve. For added support, the surgical tubing was taped to its corresponding pipe holder about 6" from one end. The plastic tubing used was 1/4" OD, 3/16" ID, and the surgical tubing was about 3/8" OD, 1/4" ID.

Each valve has another approximately 2" of the same plastic tubing on the other end, which was then fit with about 3" of surgical tubing, then fit with another 5" of plastic tubing to connect it to the 3D-printed airway splitter. This can, of course be simplified to one length of plastic tubing per valve, but this reduces a bit of the versatility of the mouthpiece. An STL file for this splitter is included in the accompanying files.

Finally, about 8" of plastic tubing was press-fit into the other end of the splitter for the mouthpiece.

III. Design Challenges/Process

There were quite a few design challenges that were met during the creation process of the instrument.

A. Pitch Control

1) Boundary Conditions: The pitch of a blown instrument, at least those with the same mechanism as a recorder, is inversely proportional to the length of the tube of air used in the instrument. Attempting to control this length is a very interesting challenge; originally, the idea was to use a graphite piston in a glass tube in order to control the length. However, this failed to work, as the condensation from my breath accumulated on the side of the glass tube, creating a nontrivial amount of friction between the piston and the tube.

After a bit of experimentation with a wooden dowel and some taped-together syringe tubes, I observed that even the slightest air leak shifted an expected open-closed boundary condition instrument to something unknown. To solve this issue, water was used, as it provided a perfect seal on one end and allowed for movement without too much friction.

2) Physical Movement: Now, the problem shifted a bit; how was I going to move the pipes up and down, and how would the instrument know the position of the pipe? This was accomplished using a motorized slide potentiometer. The instrument knows where each pipe is using the poteniometer part of the motorized slide potentiometer; the position of the pipe corresponds a resistance between 0 and 10 kΩ, which is then measured by the Teensy and converted to a position value between 0 and 1023.

However, these motors were not as strong as would be optimal, so some assistance to the motors were provided via taping the connecting surgical tubing to the pipe holders for support.

After this, the problem became figuring out which notes mapped to which positions on the potentiometer. Although it is theoretically possible to calculate which positions correspond to which notes by using lumped element analysis, a bit of pitch bend occurs when the instrument is blown into, as the air pushes into the water, making it not particularly reliable, and so I manually found the which positions correspond to which notes and stored these values as an array in the code. Then, the knobs (which send a control change signal with a value from 0 to 127) add a value between 0 and 255 to each one of the values for the array, essentially tuning the instrument sharp by a certain amount.

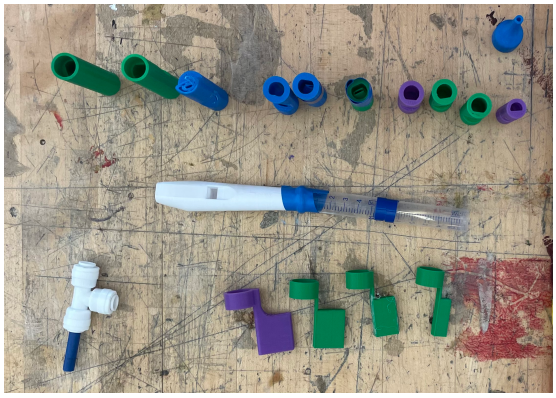

Fig. 4. An image of all prototypes of mouthpiece and adapters used in the instrument.

B. Mouthpiece

I went through about 10 iterations of the mouthpiece, as well as a few more iterations of adapters, before settling on the parts included in the files and used in the actual instrument. The main challenge in developing a mouthpiece that worked was having a long enough gap between the fipple and the other end of the mouthpiece, while also making sure it was still short enough to sound. Images of these prototypes can be seen in Figure 4 and Figure 5.

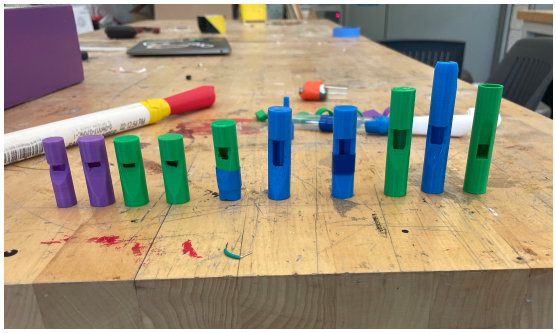

Fig. 5. A side profile of all the mouthpiece prototypes.

C. Polyphony

Polyphony presented me with two issues: how do I control which notes sound, and how do I control these notes?

As for the first of these challenges, there were two clear options given the time constraints for the project; always play all four notes, or use physical valves to control the airflow. The former option still presents us with interesting musical ideas, but the airflow required to blow all four pipes at once made it less-than-ideal. Therefore, I decided to implement a valve system, which was easy enough; the only part that required iterative design was the part to split the air into four different channels. Originally, I used a combination of plastic piping and T junctions, as seen on the bottom-left of Figure 4, but this was later replaced with a 3D-printed splitter, which ended up working first try. Again, the STL file for this splitter can be found in the files accompanying this write-up.

For the second of these challenges, there were also two choices; automatically control the pipe corresponding to which notes were played based on which notes were pressed, or manually select the pipe being controlled. I decided on the second choice, since there were two immediately obvious drawbacks to the first choice. One of these was that it not immediately trivial to implement, and the other was that it would also obfuscate which pipe would be controlled once a note was pressed, making it more difficult to figure out which valves to turn on and off. Implementing the second choice was relatively straight-forward; pads 1 through 4 on the MPK Mini 2 corresponded to four notes not playable by the instrument, so the code simply listens for one of these MIDI inputs to change the current pipe being controlled.

IV. Future Improvements

There are several improvements to the design of this instruments that can be made in addition to the ones mentioned before. One potential major improvement is with the air control; while opening and closing valves isn't terribly inconvenient, having this automated by either pressing a button or automatically opening a valve if a note is to be played would be much more convenient and could be implement via a series of solenoid valves.

Another potential improvement is with the mouthpieces; although the designs used in the final version worked, they could still be improved via even more time spent iterating and redesigning. I also have a hunch that increasing the diameter of the pipes would result in a stronger, fuller sound due to increasing airflow, although admittedly the instrument was already pushing the limits for airflow with four pipes of the small-diameter pipes used in the final version.