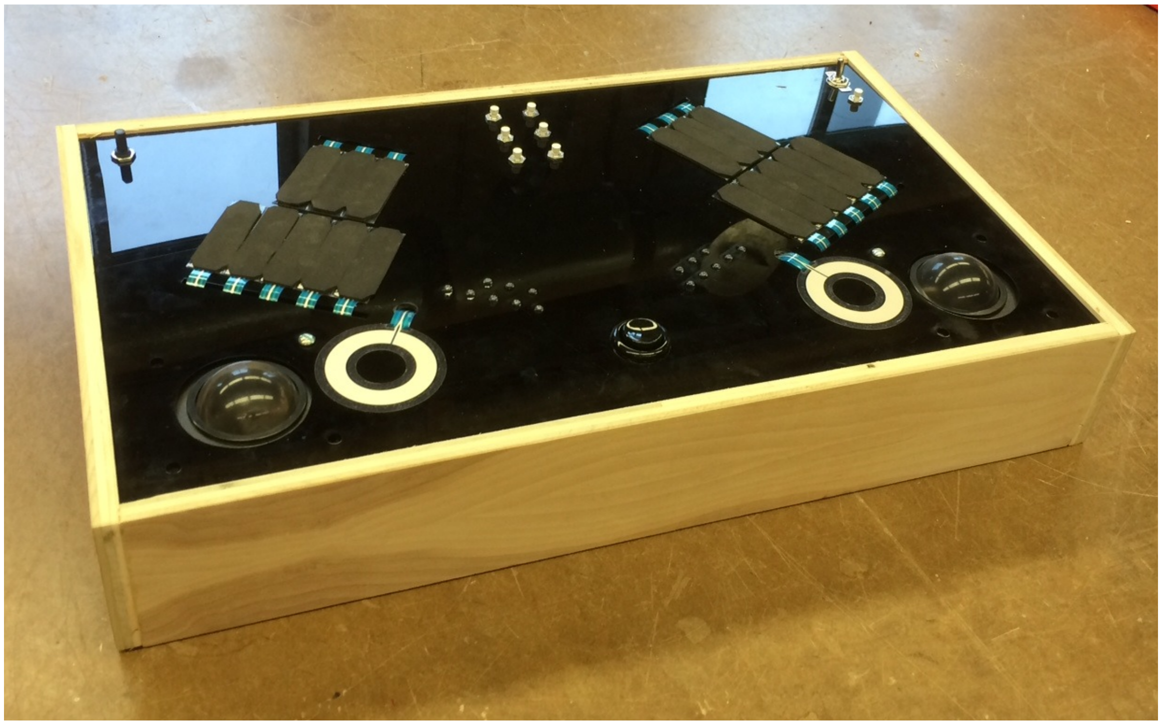
Elias Jarzombek
April 8, 2014
EMID
Final Project Report

Xenomod



Concept

Our goal was to build an instrument that could be played like a MIDI keyboard but with much more control over timbre. Its ergonomic design allows the user to control as many parameters as possible with limited hand movement. Each hand controls eight soft-potentiometers (softpots). Our initial design had the two sides resting on joysticks, but we revised this idea in favor of two trackballs, on which the user's palms could rest and have two axes to control. Unfortunately, only one of the trackballs functions properly, but both rest on FSRs so that pressing down on either of them generates data. The player also controls two rotary softpots with their thumbs. Additional buttons control octave changes and toggle various modes, and the LEDs provide visual feedback when keys are played.

Components

- 16 linear soft potentiometers
- 2 rotary soft potentiometers
- 2 2-axis trackballs
- 8 Buttons
- 2 Arduino Megas

- 8 Position Rotary Switch
- Toggle switch
- 16 LEDs
- 1 binary foot pedal

Construction

Once we finalized the placement of the components, we had a piece of quarter-inch black acrylic laser cut. In order to account for the softpots' unusually sized connectors, Nate devised an adapter that firmly attached them to the wires (appendix 1.1). The softpots come with adhesive on the back, so applying them was very simple. For a more comfortable playing surface, we added 16th inch strips of neoprene on top of each softpot. Underneath the panel, we used a breadboard to organize the wires before feeding them into the Arduinos (appendix 1.2). Nate built a sturdy wooden box to house the apparatus.

How it is Played? (Appendix 2)

This instrument is played for the most part like any MIDI keyboard, in that it has "keys" that each correspond to a specific note value. In order to achieve the polyphony that we desired, we assigned each softpot to its own channel in Reason. Thus, each key has independent control over its own sound. Touching the softpot generates a note and sliding up and down the key adjusts a parameter selected by the rotary switch (on the top left of the panel). The user selects a root note and scale through the Max interface, and can shift the octaves on both hands by using the bottom four buttons. The scale can also be changed using the foot pedal. A key feature of the device is the ability to link and unlink the two hands using the top left button. When linked, all changes made to the left hand affect the right, one octave up. When unlinked, the player can choose scales and roots independently, for some interesting results. The last button (top right) toggles the velocity mode. When enabled, velocity is tracked along with whatever parameter is active. When disabled, the velocity is set at a constant value (defaulted to 127). The x and y axes on the right trackball control delay dry/wet and feedback on all channels, respectively. Pressure on the right trackball adjusts the parameters of Thor's formant filter, which results in a wah wah effect. The right rotary softpot controls master volume. On the left side, pressure on the trackball controls LFO amount and the rotary softpot controls LFO rate.

Sequencer mode:

The toggle switch at the top right of the panel turns on and off a sequencer mode and the button directly underneath starts playback. This mode enables the user to control a sequencer using all of the available controllers. In this mode, each key controls a step in the sequencer, and the pedal acts as an alt switch. the rotary switch controls which parameter the softpots control. At position 0, they control nothing. At position 1, they control note value from 1-15 in the selected scale. A value of 16 turns the note off. The second and third positions allow the keys to adjust filter frequency and wavetable position. In the center panel, the bottom two left buttons control octave change and the bottom right buttons select major or minor scale. To select a root note, the player presses the alt pedal and presses a key. The right rotary softpot controls note duration or, when the alt is pressed, tempo. The rotary softpot controls portamento. Pressure on the right trackball serves the same function as in the normal mode, whereas pressure on the left trackball adds white noise to the sound. Spinning the right trackball has multiple features in sequencer mode. Moving it normally controls delay time and dry/wet. Spinning it quickly to the right randomizes all note values, down randomizes al frequency values, and up randomizes the wavetable position values. The large black button in the center of the panel resets all parameter values and, when the pedal is pressed, all not values.

Max/Msp patch (appendix 2)

The Max patch has three main parts: reading in and manipulating the data from the softpots, reading and applying the data from the other controllers, and organizing the note values for each key.

We expanded the Arduino code and *Serialreader* subpatch (appendix 2.3) to account for the extra digital pins required for the LEDs. This patch reads in all of the Arduino data and sends each pin to a different value. The main patch receives the data for each of the 16 softpots and sends each to its respective channel and subpatch. This subpatch (appendix 2.4) generates a note, finds the velocity, and handles the CC modulation data based on the position of the rotary switch. It holds the note for as long as you are pressing down and upon release, it holds the last value so that the controller does not jump to zero.

The part of the patch that selects the mode is straightforward. It simply receives the data from the rotary switch pins and bangs the corresponding value to every softpot. The data from the FSRs, trackballs, and softpots is also manipulated and sent to all channels. The final section is the note and key selector. The root note routes to the intervals required for each scale, and applies them to the subsequent notes, which are sent to each softpot. The link connects the two hands using gates.

Sequencer:

Using a metro and a counter as a foundation, we built a custom sequencer for the Xenomod. In one cycle, 16 bangs are sent at intervals dictated by the tempo slider. They send a new note value, frequency value, and wavetable amount value each bang (the sequencer is all on one channel in Reason). The randomizers detect when the trackball is being spun quickly and in what direction. You can also set the range of values within which the randomizer generates notes. The foot pedal operates various gates so that a controller can only affect one parameter at a time (as described earlier).

Reason (appendix 3.1)

Our Reason rack consists of sixteen identical Thor modules that produce a thick, warm sound. Because Thor has so many adjustable parameters, the patch lends itself well to manipulation. The static parameters that can be controlled are velocity, pitchbend, filter 1frequency, and modwheel. The four assignable variables are currently set to Wavetable position, chorus amount, shaper amount and filter 3 frequency mix.

Challenges

We ran into several challenges along the way, some of which we overcame, some of which we did not. The trackballs posed the biggest problem and we were unable to get one of them to function. Initially we encountered a problem with the Arduino, where the pins were extremely noisy and would affect all of the other pins. We overcame this challenge by simply enabling the Arduino's pull up resistors. We also had planned to use two FSRs under the trackballs to add aftertouch. To do this we needed an Arduino expander, but the expander we received was faulty. We solved this problem in the second phase of the project by simply using two Arduinos. In Max, we overcame several obstacles, including sustaining the note for as long as the softpot is pressed, holding the last value to avoid parameters jumping to zero, and resetting parameters when the rotary switch is turned.

Things I would do Differently

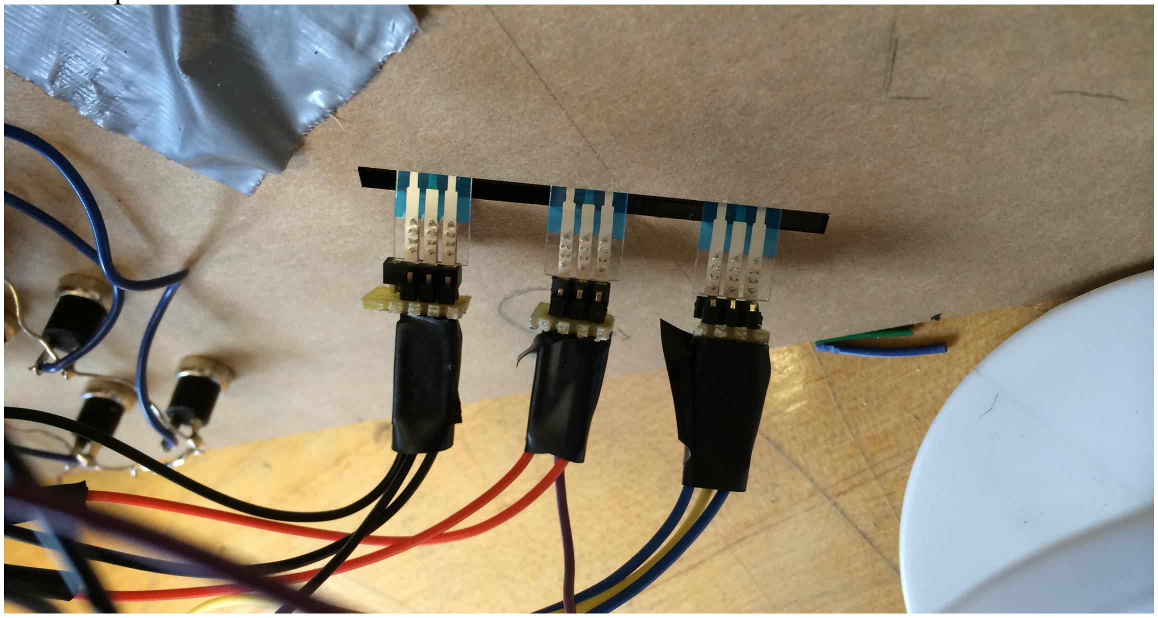
Though it probably would have been infeasible, I would have liked to make the device more ergonomic with two curved surfaces so that it would be easier for hands to reach the keys. I also would have put the buttons much closer to the keys so that they could be pressed without lifting all of your fingers.

Upgrades/Additions in the second iteration

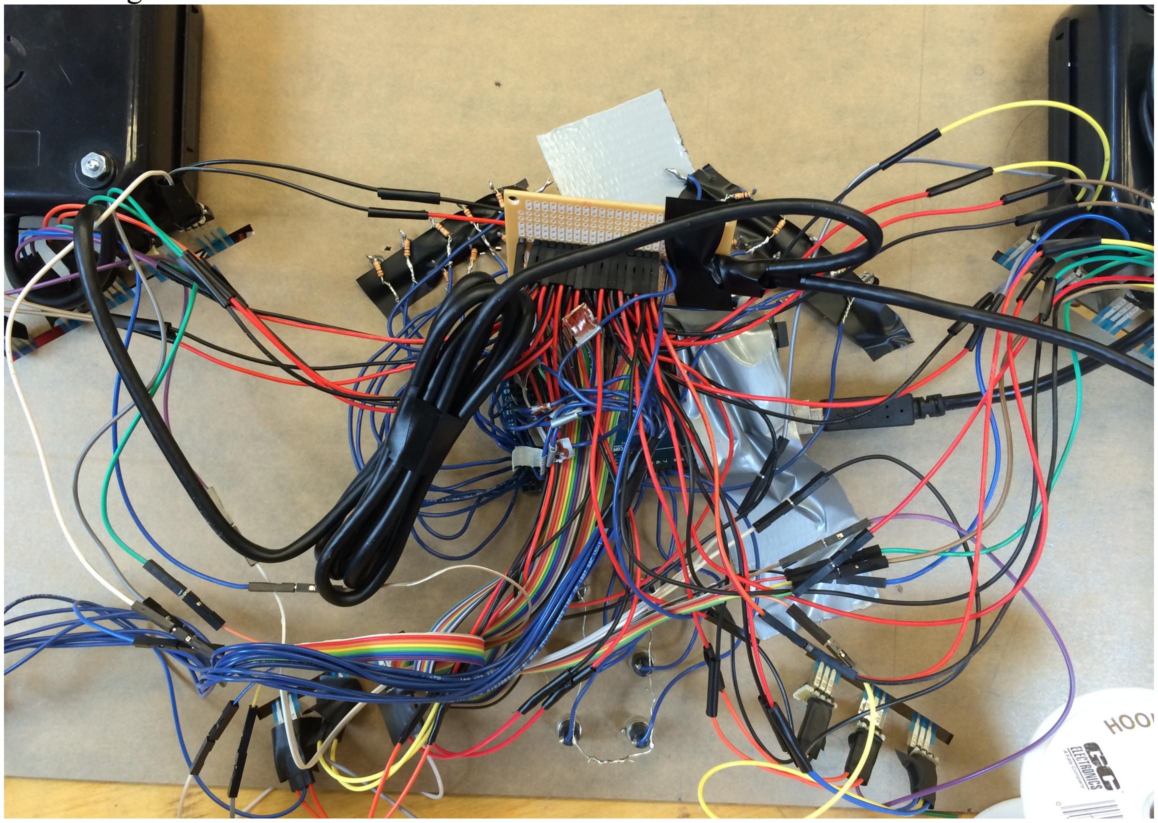
- Added trackball functionality (right side)
- Added two FSRs under trackballs
- Added two rotary softpots
- Implemented Sequencer Mode and its integration
- Constructed wooden casing
- Added reset button
- Added second Arduino
- Added foot pedal
- Fixed bugs/improved Max code

Appendix 1: Hardware

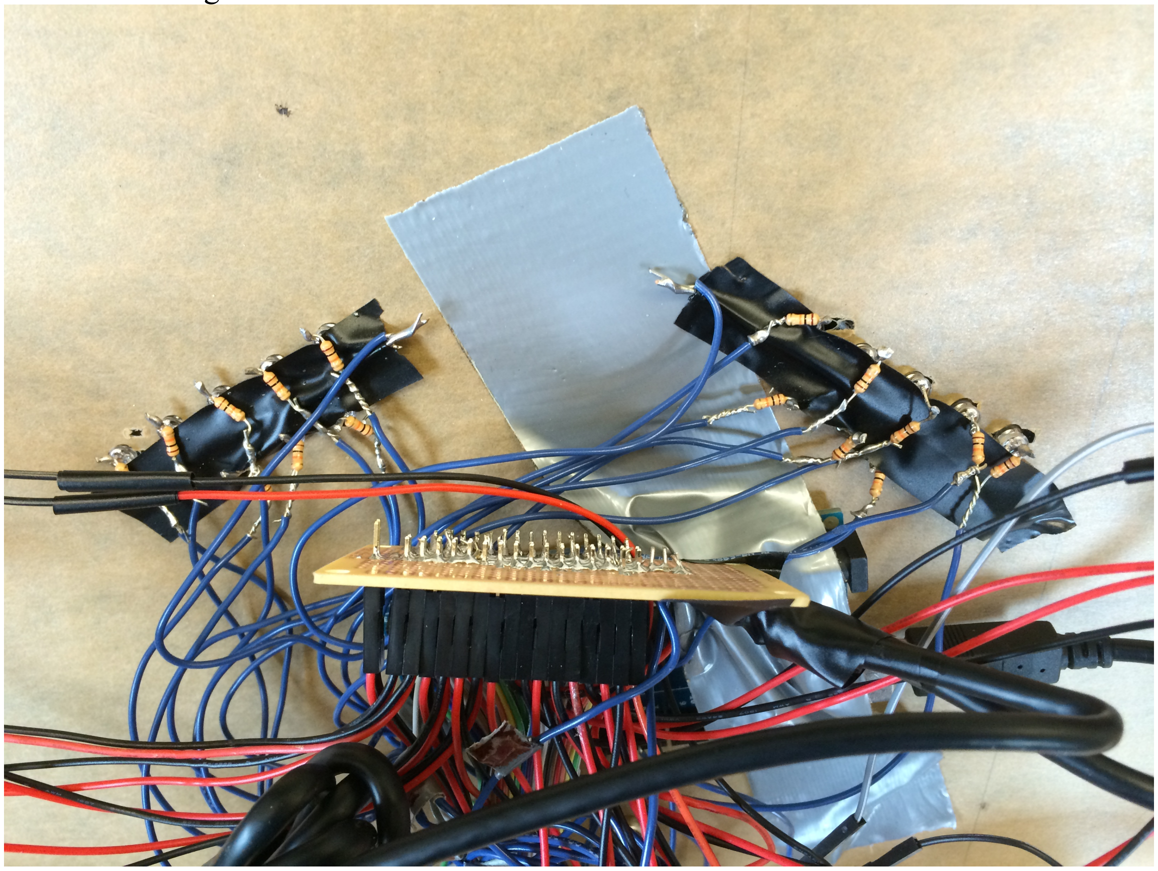
1.1: Softpot connectors



1.2 Wiring



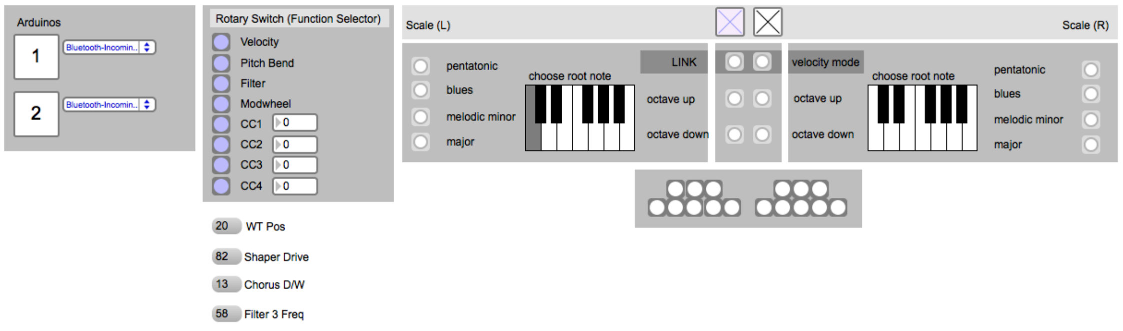
1.3 More Wiring



Appendix 2: Max

2.1 Presentation modes

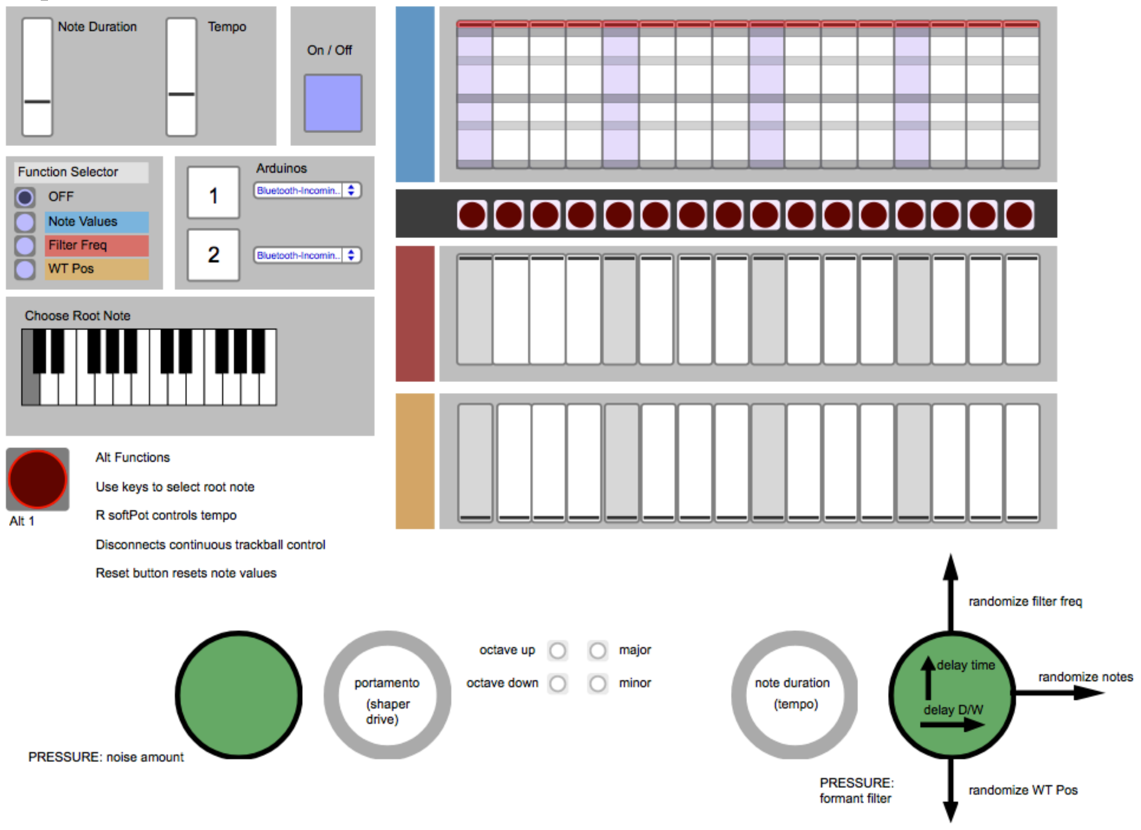
main:



CC1 0
CC2 0
CC3 0
CC4 0

20 WT Pos
82 Shaper Drive
13 Chorus D/W
58 Filter 3 Freq

Sequencer:



Alt Functions

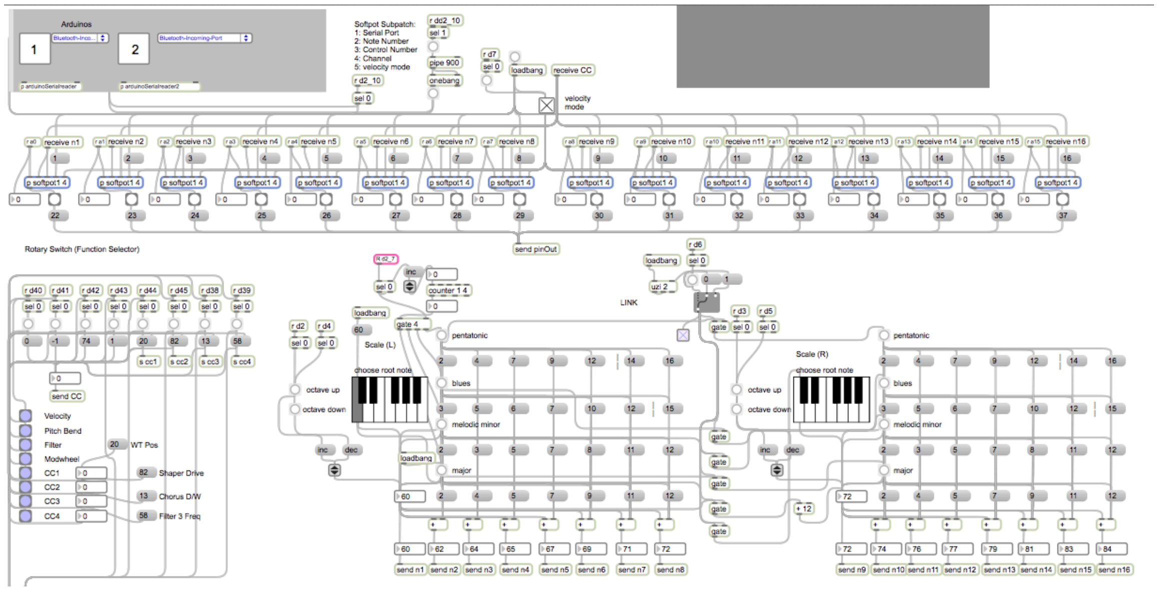
Use keys to select root note

R softPot controls tempo

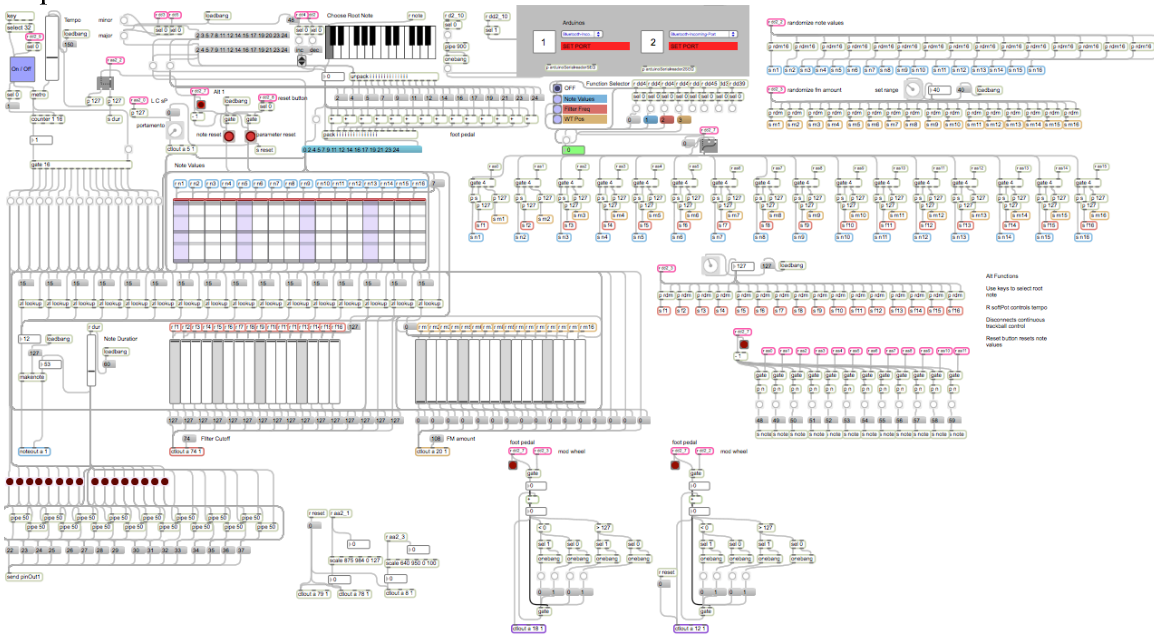
Disconnects continuous trackball control

Reset button resets note values

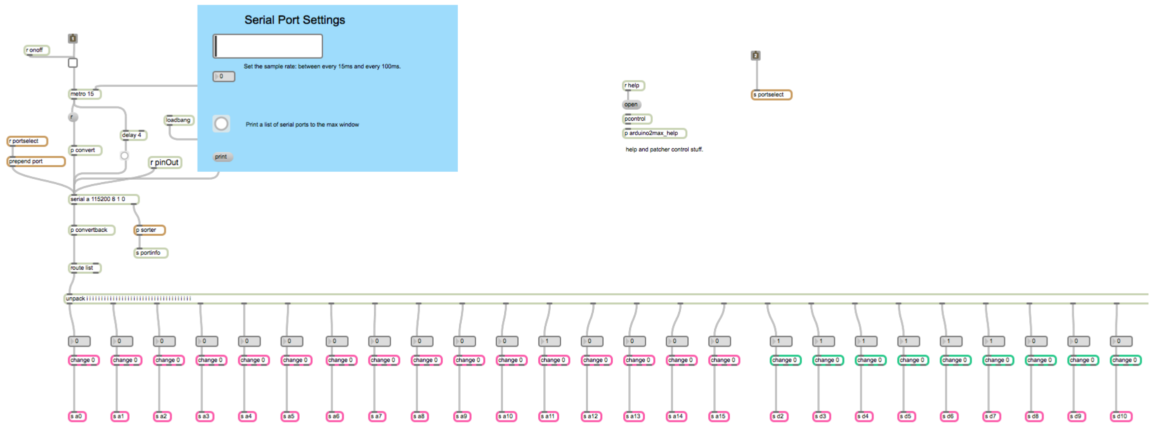
2.2 Main Patch



sequencer:



2.3 SerialReader subpatch



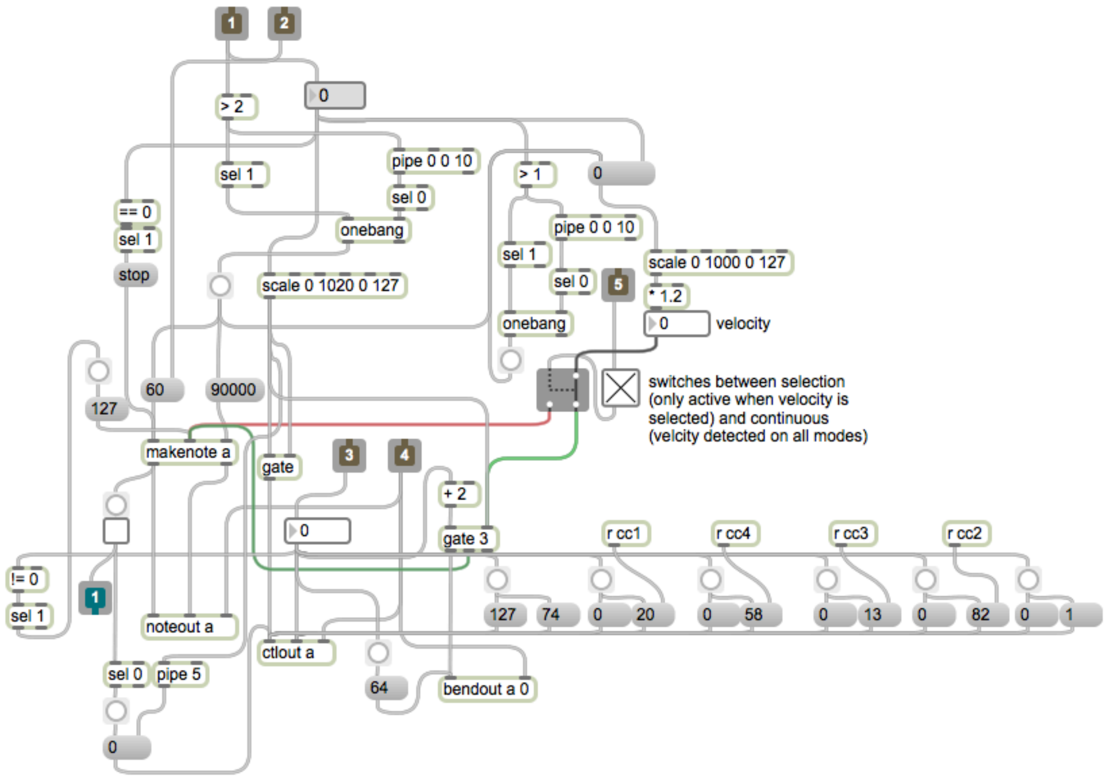
Serial Port Settings

Set the sample rate: between every 15ms and every 100ms.

Print a list of serial ports to the max window

help and patcher control stuff.

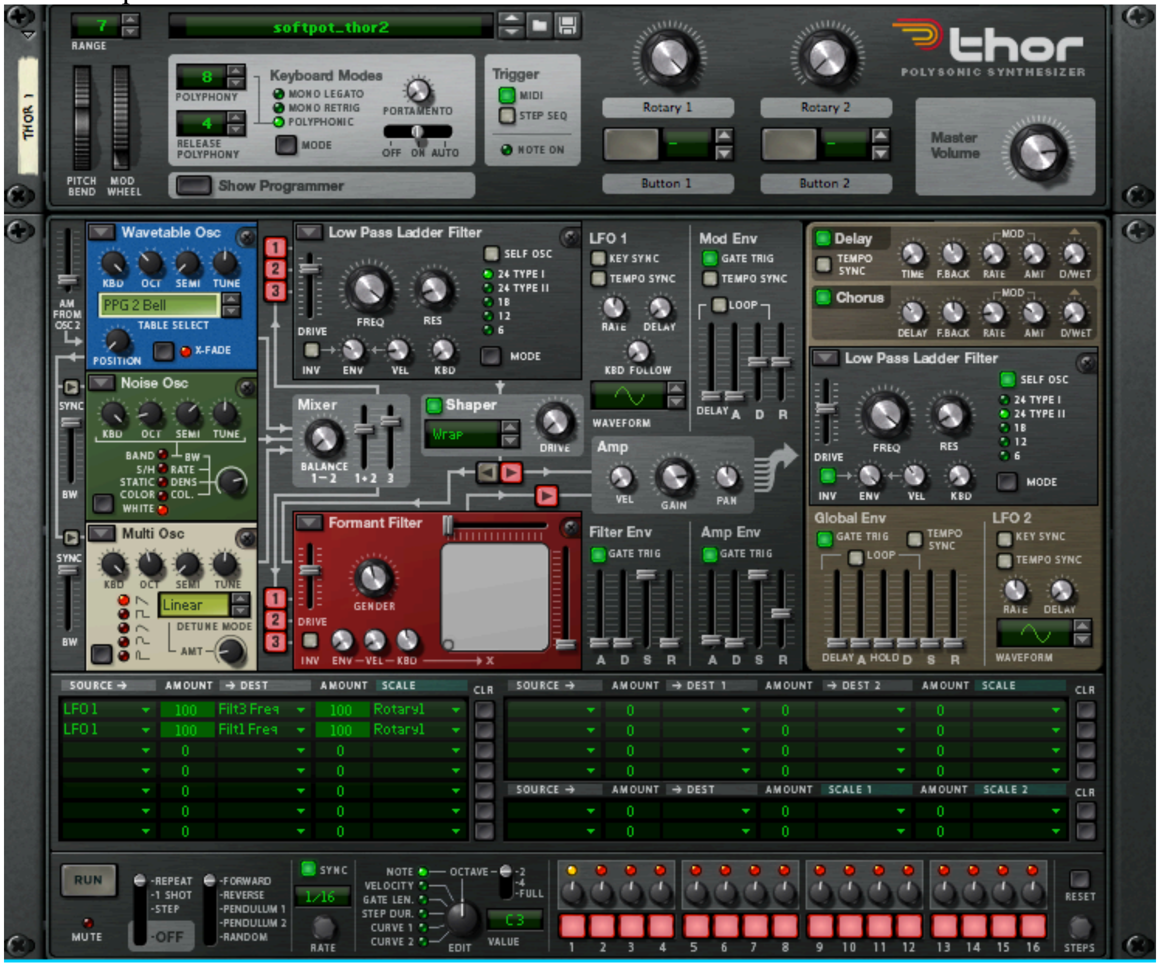
2.4 Softpot subpatch



switches between selection (only active when velocity is selected) and continuous (velcity detected on all modes)

Appendix 3: Reason

3.1 Thor patch



softpot_thor2

Polyphony: 8; Release Polyphony: 4; Range: 7

Wavetable Osc: PPG 2 Bell

Shaper: Wrap

Multi Osc: Linear

SOURCE →	AMOUNT	→ DEST	AMOUNT	SCALE
LFO 1	100	Filt3 Freq	100	Rotary1
LFO 1	100	Filt1 Freq	100	Rotary1
	0		0	
	0		0	
	0		0	
	0		0	
	0		0	

Step sequencer: Sync 1/16; Value C3; OFF