OK, so here’s how the scanline synthesis patch works. You load a jpg file into a buffer. Brightness/contrast/saturation of the color can be edited, and then it’s converted to luminosity values (greyscale 0 = black 1 = white) from the RGBA format, and the number of planes is stripped from 4 to 1. As this is happening, three preview screens display the original, contrast edited, and black/white versions. The luma file’s X/Y dimensions are summed and used to control a scanning wrapper that reads the value of each pixel in sequence into a matrix. Note that this can be a huge number of pixels. A 640x480 video is around 179200 individual values, give or take an extra. We’ll come back to the buffer in a minute.

MIDI is taken in and broken down into pitch/velocity pairs. Pitch is split into two streams – the first controls a sawtooth ramp that cycles through the matrix buffer reading each value in sequence (either interpolated or non-interpolated) at a frequency rate equivalent to the MIDI note played. It’s basically using the luma matrix as a giant waveform. This audio is normalized and split, with half of it going to a delay to add a sharper attack. This is controlled by the split of the MIDI note from before, with the pitch converted into milliseconds to control the delay time. After that, it goes through a resonant lowpass filter, with an interesting feature.

Remember how I mentioned that there were three preview screens? If you hover your mouse on them, you can click to select a pixel from any of the screens and output it’s RGBA values. These are scaled and normalized to control the cutoff and resonance of the delay network’s filter. The output of the delay network and the original audio stream are both sent through an ADSR controlled volume envelope (which is controlled by the velocity from MIDI), before being summed, then run though a variable mode filter to shape the sound, then the output.

The result is gnarly, electronic, pitched noise. Lower frequencies are wider and sound great with a filter sweep. Higher frequencies are piercing and insect-like and would do really well in the GRM Space Grains plugin.

Next step is to make it polyphonic and fully MPE compatible.