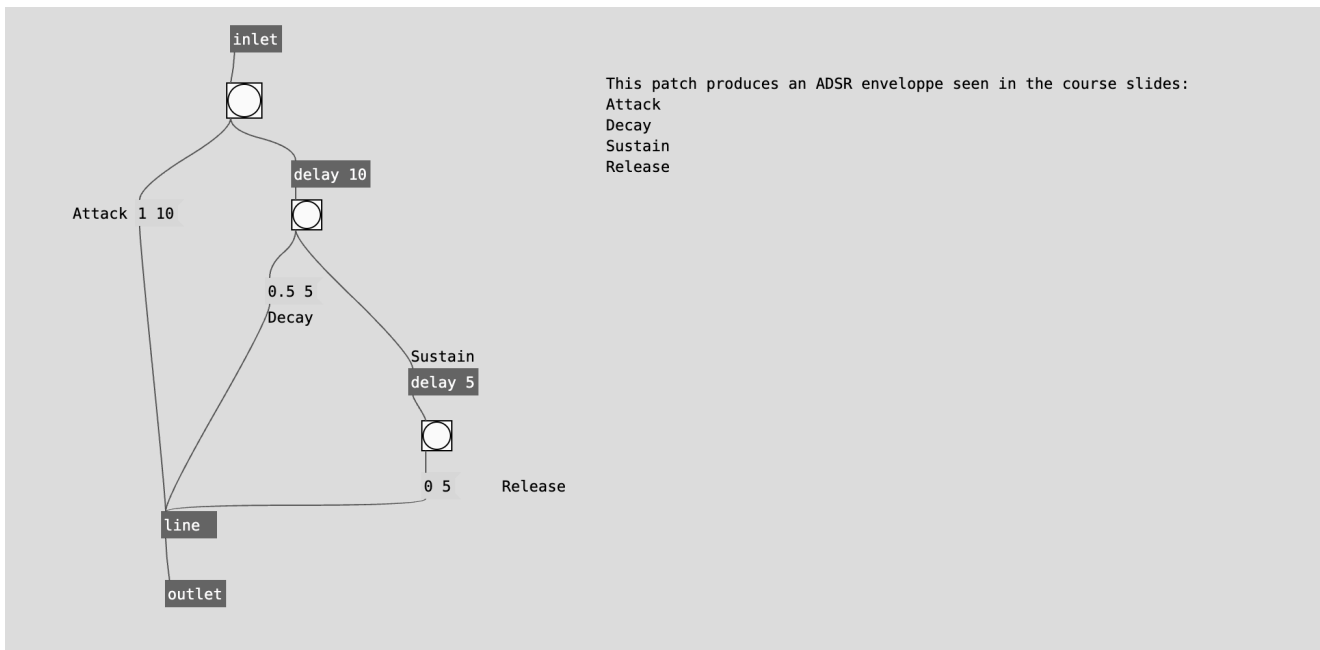


# Lab Report

## Sound Production:

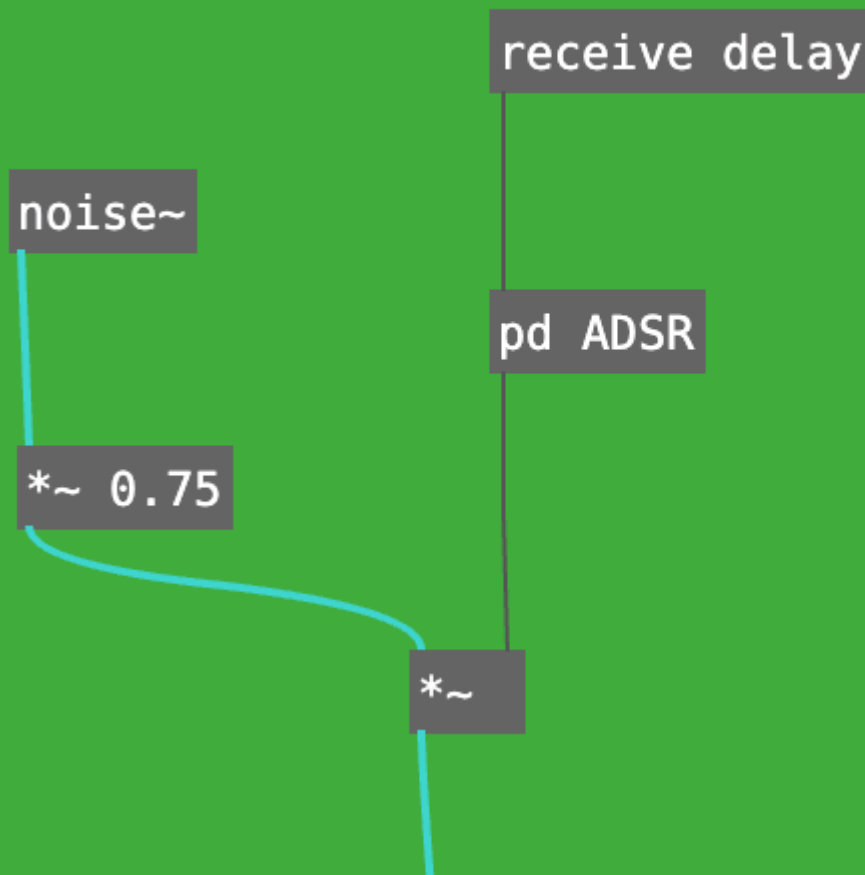
To produce the sounds i am using an [ADSR](#) envelope applied to [white noise](#).

- ADSR envelope:



- Applied to White noise:

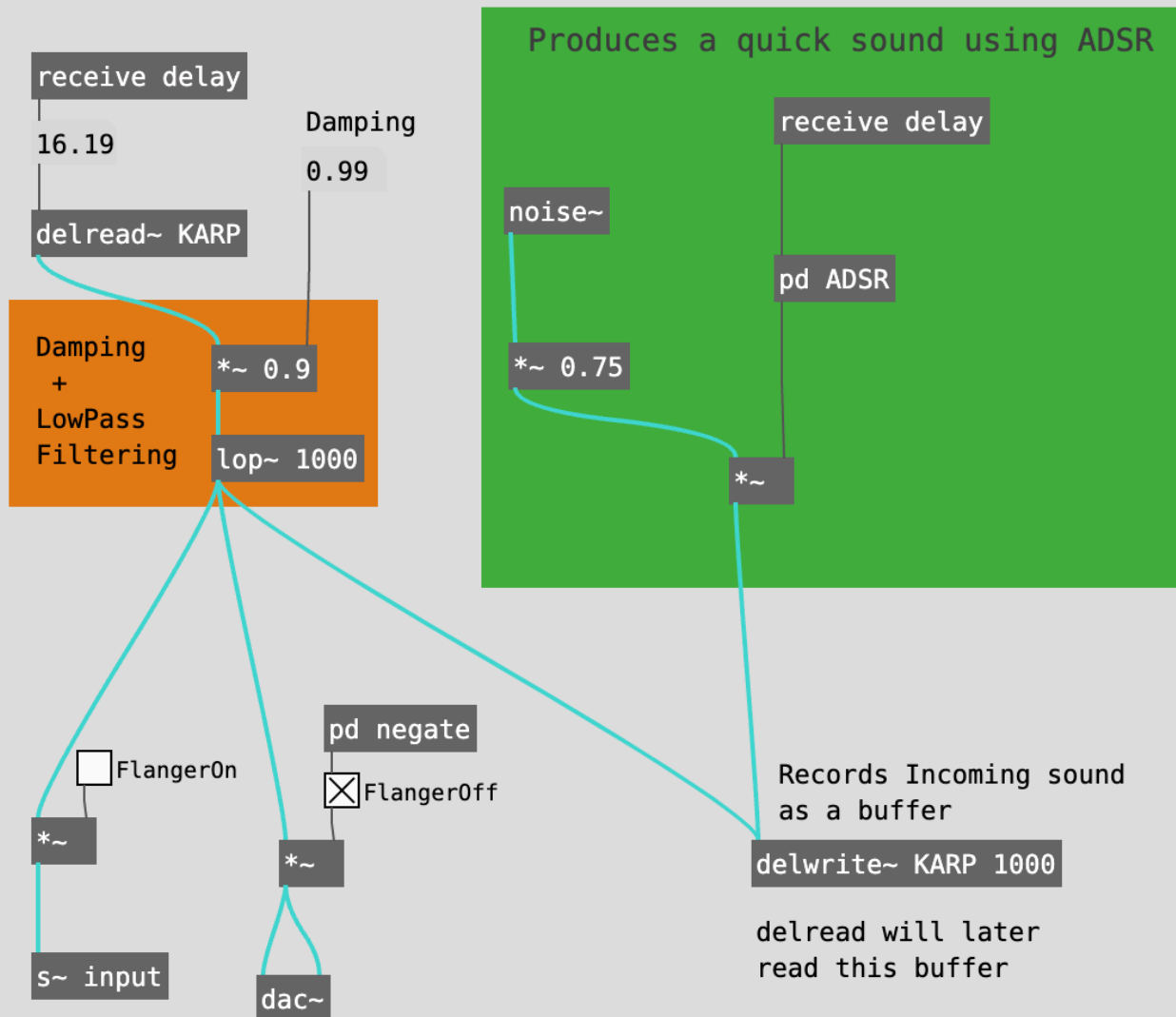
Produces a quick sound using ADSR



## Karplus Strong

This produces a sound burst which is then passed to the [Karplus Strong Algorithm](#) section of the patch.

# Karplus Strong Algorithm



The burst is written to a buffer using `delwrite~` (KARP is the name of the delay and 1000 is the maximum delay amount).

We then use `delread~` to output the signal stored in the buffer with the delay it receives from from the keyboard (we'll get back to that later).

The amount of delay is what defines the Pitch of the sound produced by the Karplus Strong Algorithm.

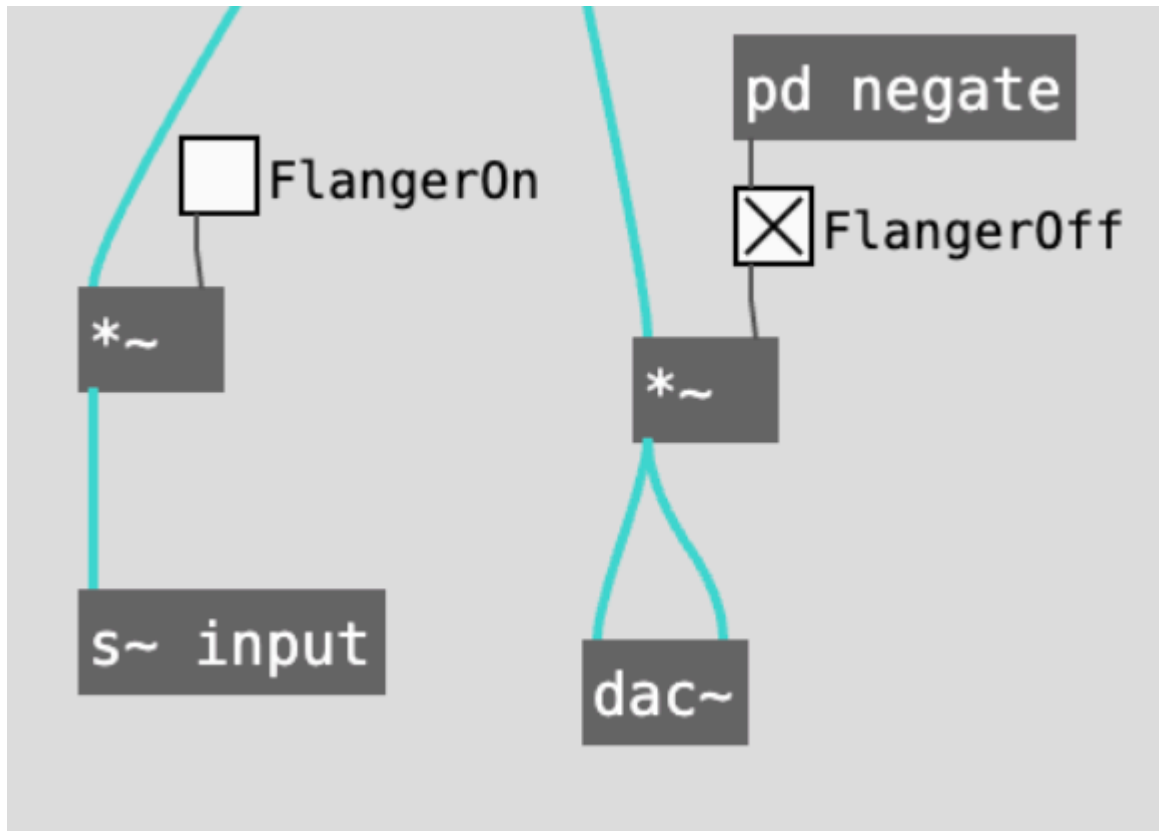
We then dampen and apply a low pass filter to the dampened signal.

The user can control the value of damping with a slider next to the keyboard.

We then feed the output back into the loop by inputting it to `delwrite~`.

The custom effect i am using is the flanger effect.

This section controls whether or not the effect is active:

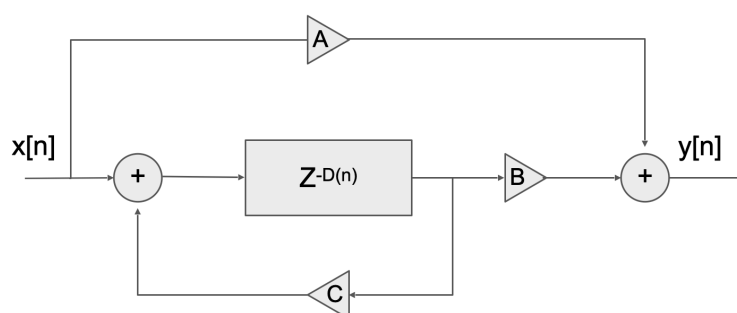


## The Flanger Effect:

The [flanger](#) effect works by duplicating the audio signal, applying a variable delay (usually a short time, like 1-10 milliseconds), and gradually changing the delay amount. The delayed signal is then mixed back with the original signal, creating a “swooshing” or “jet plane” sound as the phases of the signals interact, causing constructive and destructive interference.

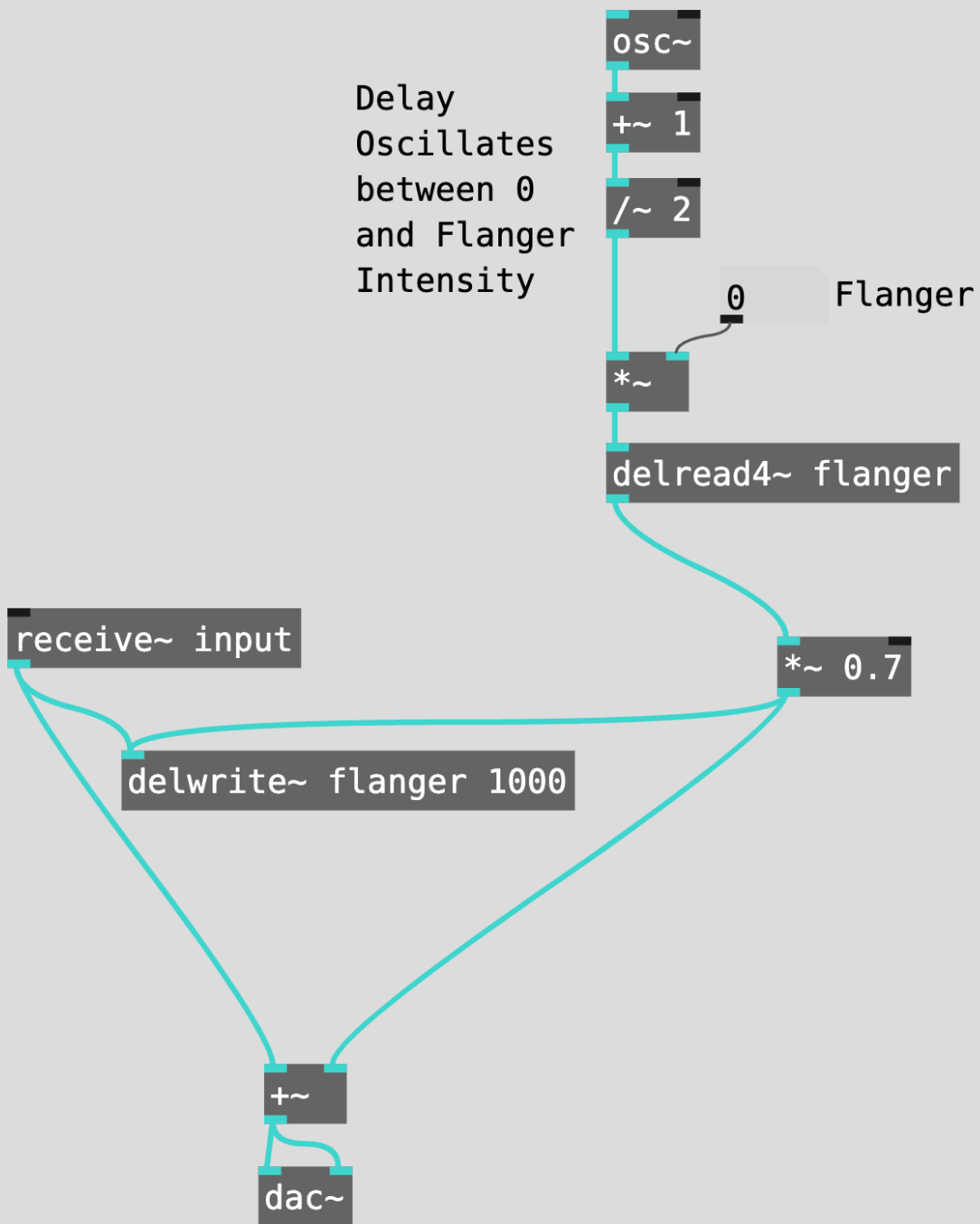
Effects in the space-time domain

- Flanger:  $A = 1$ ,  $B = 0.7$ ,  $C = -0.7$ ,  $0 \leq \text{time} \leq 10\text{ms}$ 
  - the purpose of the flanger is to superimpose a dynamically delayed signal on the source signal, whose delay remains within the ear's integration period. A delay line of 1ms is fine for many applications.

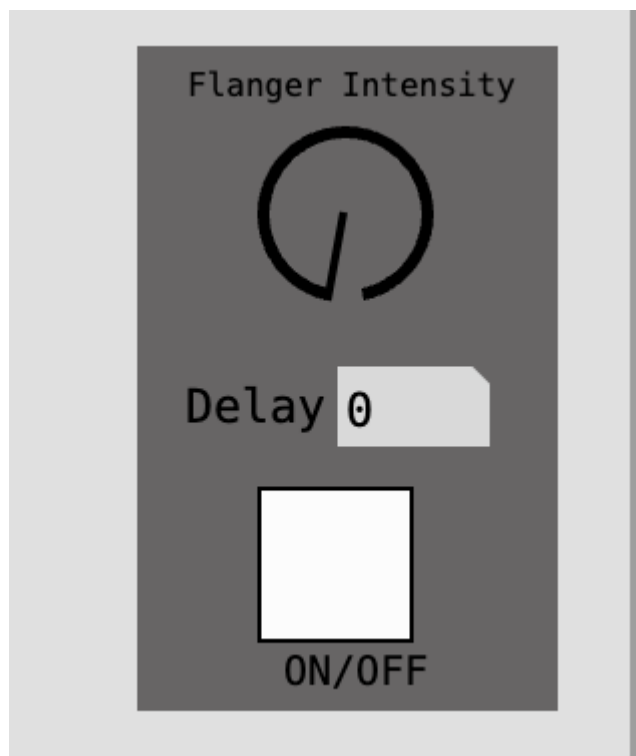


The implementation in PureData is as follows:

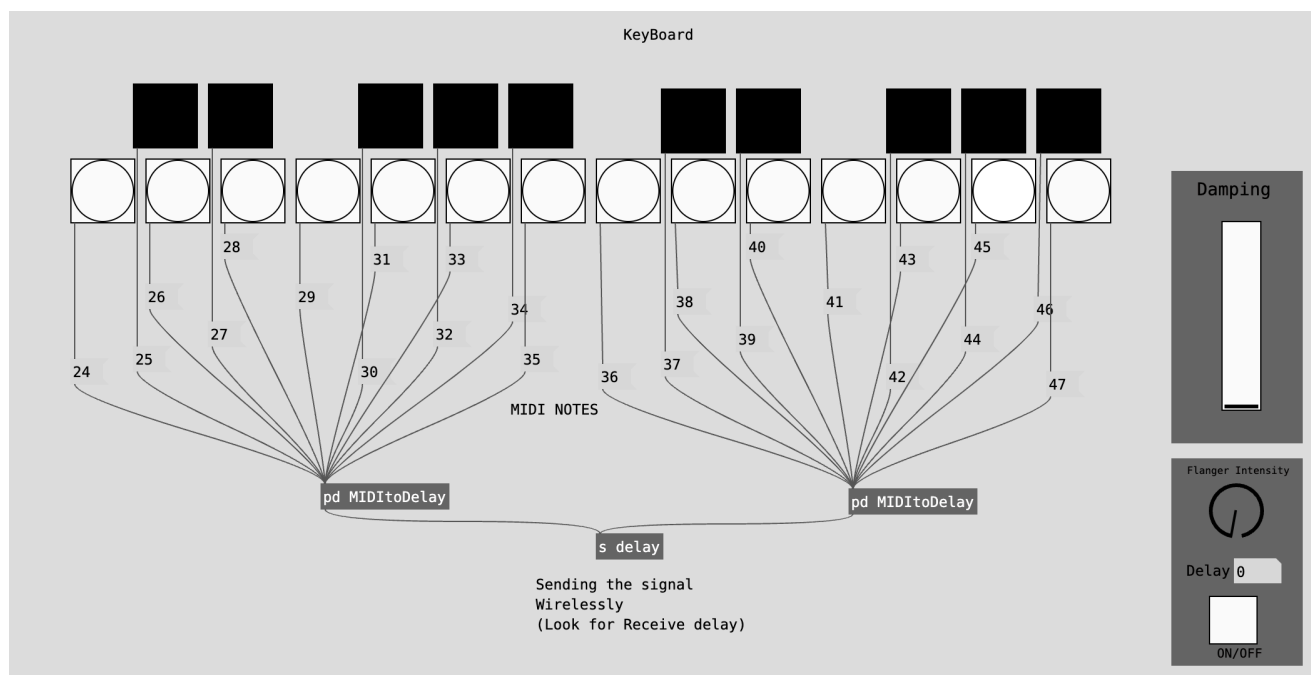
# Flanger Effect



The user can turn on and control the intensity of the effect using a toggle and a knob.

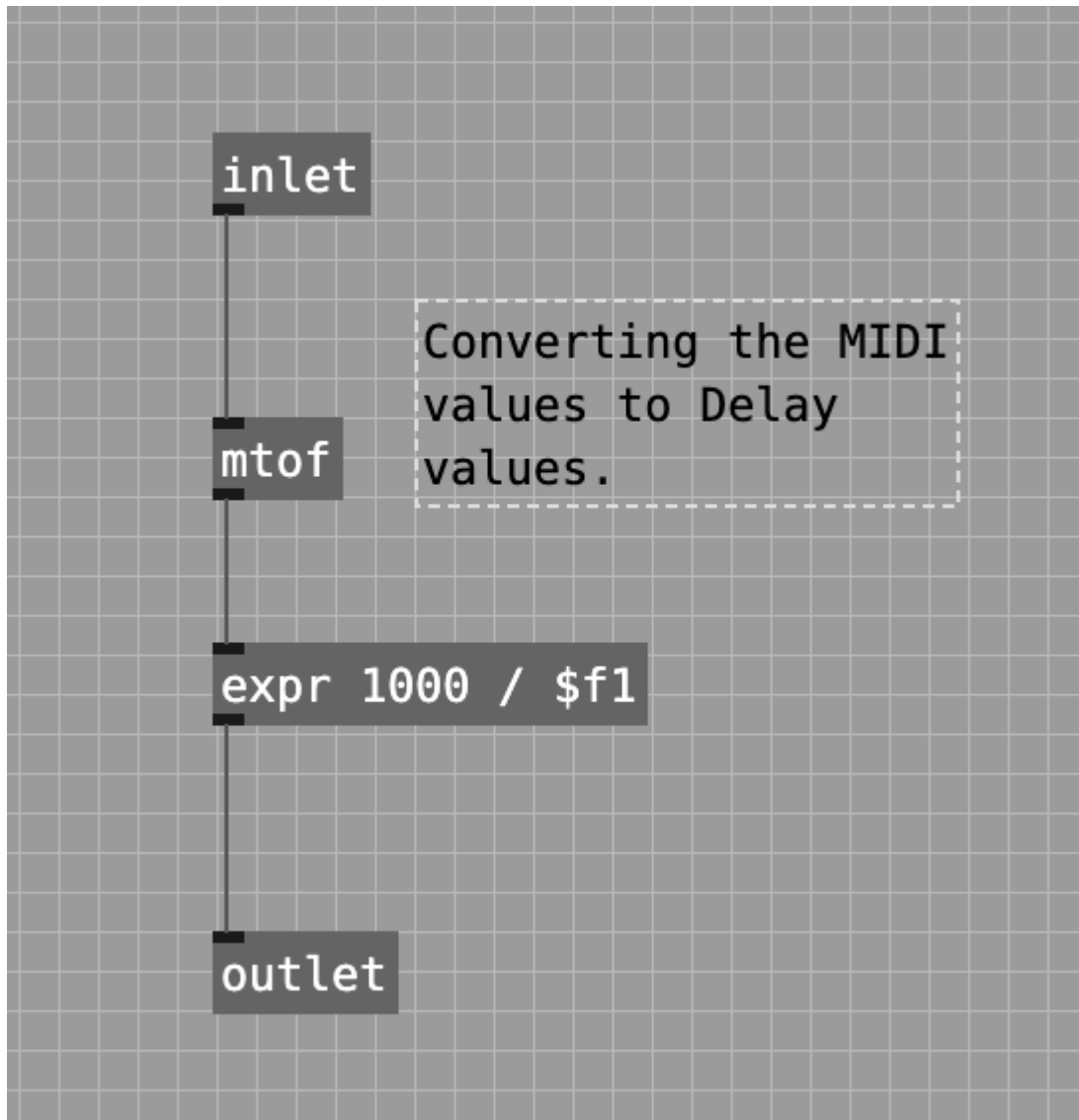


## The Keyboard:



I created a piano keyboard in Pure Data where each key is linked to a “bang” corresponding to a specific [MIDI](#) number.

This MIDI number is then converted into a delay value that determines the pitch of the note.



The delay is fed into the Karplus Strong algorithm, which simulates the sound of a plucked string by processing the delay with feedback and a short noise burst.

The keyboard can be controlled using the laptop's QWERTY keyboard, allowing for easy interaction. This setup generates the appropriate sound for each note.