

IMCA 221
Programming for Artists
Fall 2024

lee wilkins
l.wilkins@concordia.ca

**Class is on Zoom even in the classroom
for sharing, find the details on Moodle**

<https://moodle.concordia.ca/>

Download the slides!

Inspo

Imagen Heap Gloves



[https://
www.dezeen.com/
2014/03/20/imagen-
heap-funding-drive-
for-gloves-that-turn-
gestures-into-music/](https://www.dezeen.com/2014/03/20/imagen-heap-funding-drive-for-gloves-that-turn-gestures-into-music/)



Victoria Shen
[https://
evicshen.com/
misc](https://evicshen.com/misc)



Laurie Anderson
[https://cdm.link/
2021/09/laurie-
anderson-in-video-
on-the-stories-
behind-her-
custom-built-
instruments/](https://cdm.link/2021/09/laurie-anderson-in-video-on-the-stories-behind-her-custom-built-instruments/)



Janet Cardiff
[https://
www.youtube.co
m/watch?
v=38ORiaia9r8](https://www.youtube.com/watch?v=38ORiaia9r8)

Resources:

Max Documentation: <https://docs.cycling74.com/max8>

Max Cookbook <https://music.arts.uci.edu/dobrian/maxcookbook/>

Andrew Robinson Video Tutorials <https://www.youtube.com/@AndrewRobinson26>



Max Thesaurus

**[https://docs.cycling74.com/max8/
vignettes/thesaurus](https://docs.cycling74.com/max8/vignettes/thesaurus)**

Find the name for objects that might be difficult to find otherwise!

Keyboard Shortcuts

<https://docs.cycling74.com/max8/vignettes/keycommands>

a: attrui.

b: button.

c: comment.

f: floating point number box.

h: briefly highlights a small area around the cursor.

H: A capital letter "H" briefly highlights a larger area around the cursor.

i: integer number box.

j: object box containing "jit." for creating Jitter objects.

l: object box containing "live." for creating Live objects.

m: message.

n: new blank object with the cursor active. Typing the name of any object and pressing enter or clicking outside of the object box will transform it into that object.

r: bring up a list of the most recently created objects, including any arguments and attributes typed in. Selecting an element from the list creates an object with the corresponding text and with the cursor active at the far right. Hitting a carriage return or clicking away from the object instantiates the object.

p: create an object box containing the message newobj @presentation 1 @text and a cursor. When you type the name of an object (e.g. dial) and hit a carriage return, the object will transform itself into a copy of the object whose name you type in, and the object will be automatically added to the Presentation Layer.

s: slider.

t: toggle.

x: shows a menu describing the key commands, including those added by external packages.

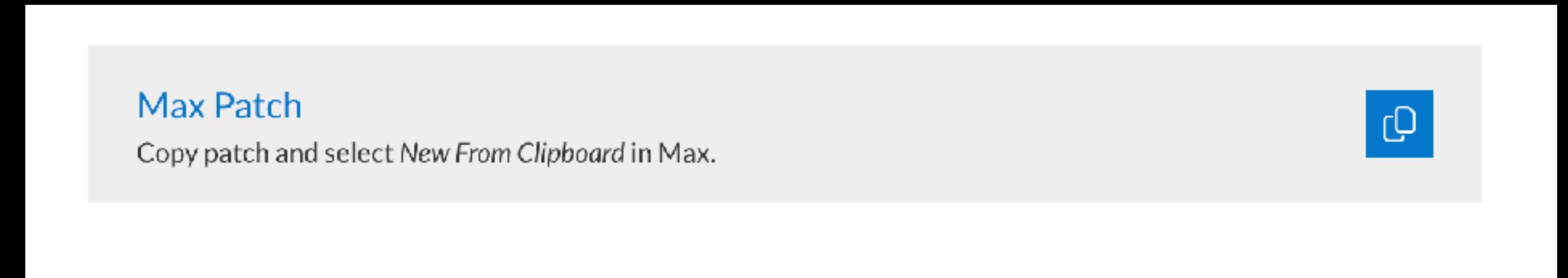
z: zooms the patcher in around the cursor.

Z: A capital letter "Z" zooms the patcher out around the cursor.

You can copy Max code and paste it directly into your patch as text.

CMD + V (paste)

```
{  
  "boxes": [  
    {  
      "box": {  
        "maxclass": "gain~",  
        "patching_rect":  
          [ 385.792366743087769,  
            357.923513054847717, 22.0, 140.0 ],  
        "outlettype": [ "signal", "" ],  
        "multichannelvariant": 0,  
        "id": "obj-114",  
        "parameter_enable": 0,  
        "numinlets": 1,  
        "numoutlets": 0  
      }  
    }  
  ]  
}
```



Lets make a simple sequencer together. We'll learn to COUNT and SELECT,.

Count: Counts up or down

Select: chooses a path that matches the input.

Count & Select

tri~

saw~

Try these!

Set the beat, you can
use tempo if you want

Count from 1 - 4

Select each
number

Show they are working

Set some frequencies

Set the wave

Multiply the
amplitude

Toggle On

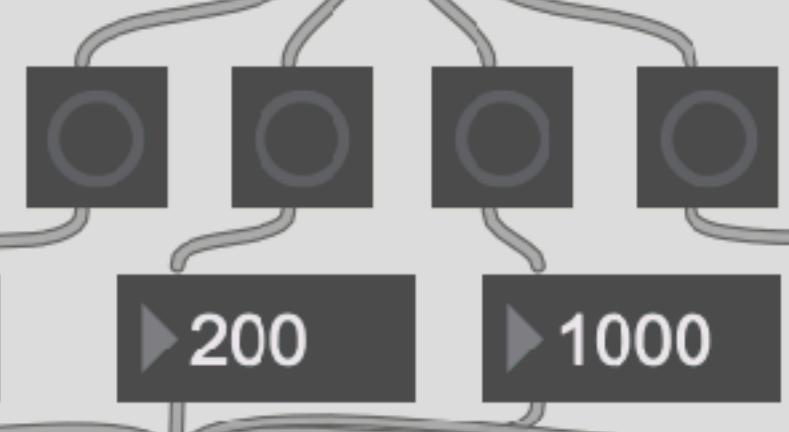
metro 1000

counter 0 1 4

▶ 3

Show the
current
count

sel 1 2 3 4

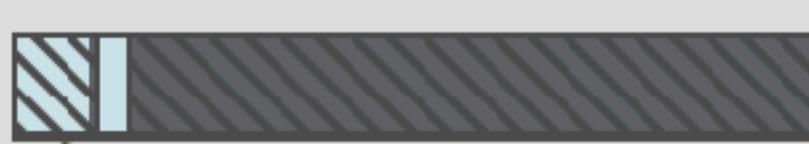


▶ 300 ▶ 200 ▶ 1000 ▶ 400

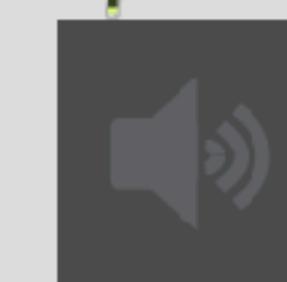
cycle~

Slider to
change
amplitude

*~ 1

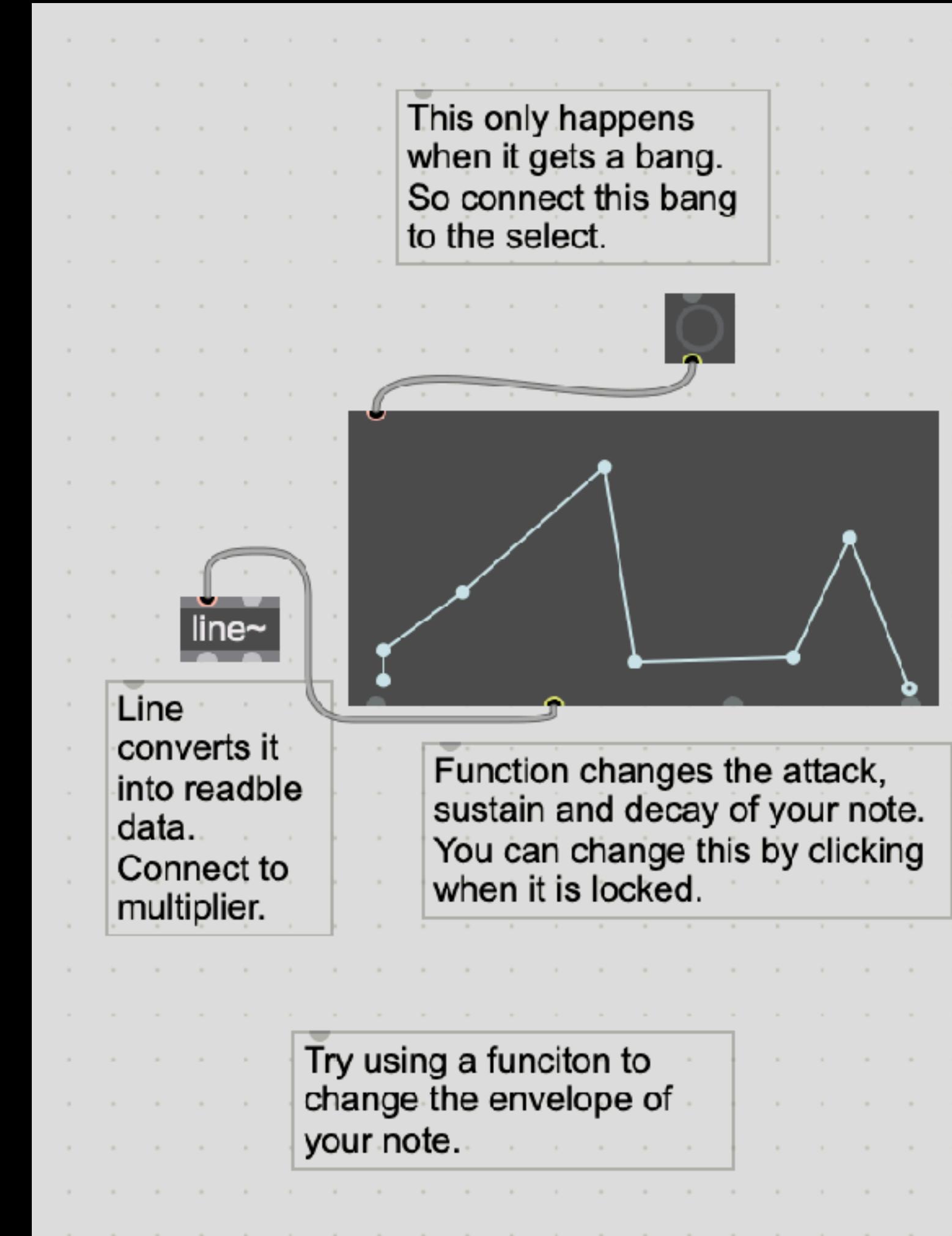


Gain slider

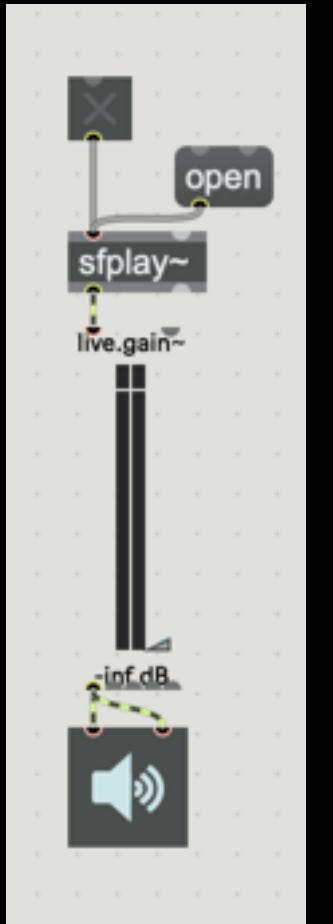


Output

We can modify the notes by using this envelope technique



SFPlay~ can load sound from a buffer or from open and has lots of features



Playlist~ is the automatic object created when dragging. It has UI and can hold lots of files in a list.



You can use an integer to specific which file to play on a list

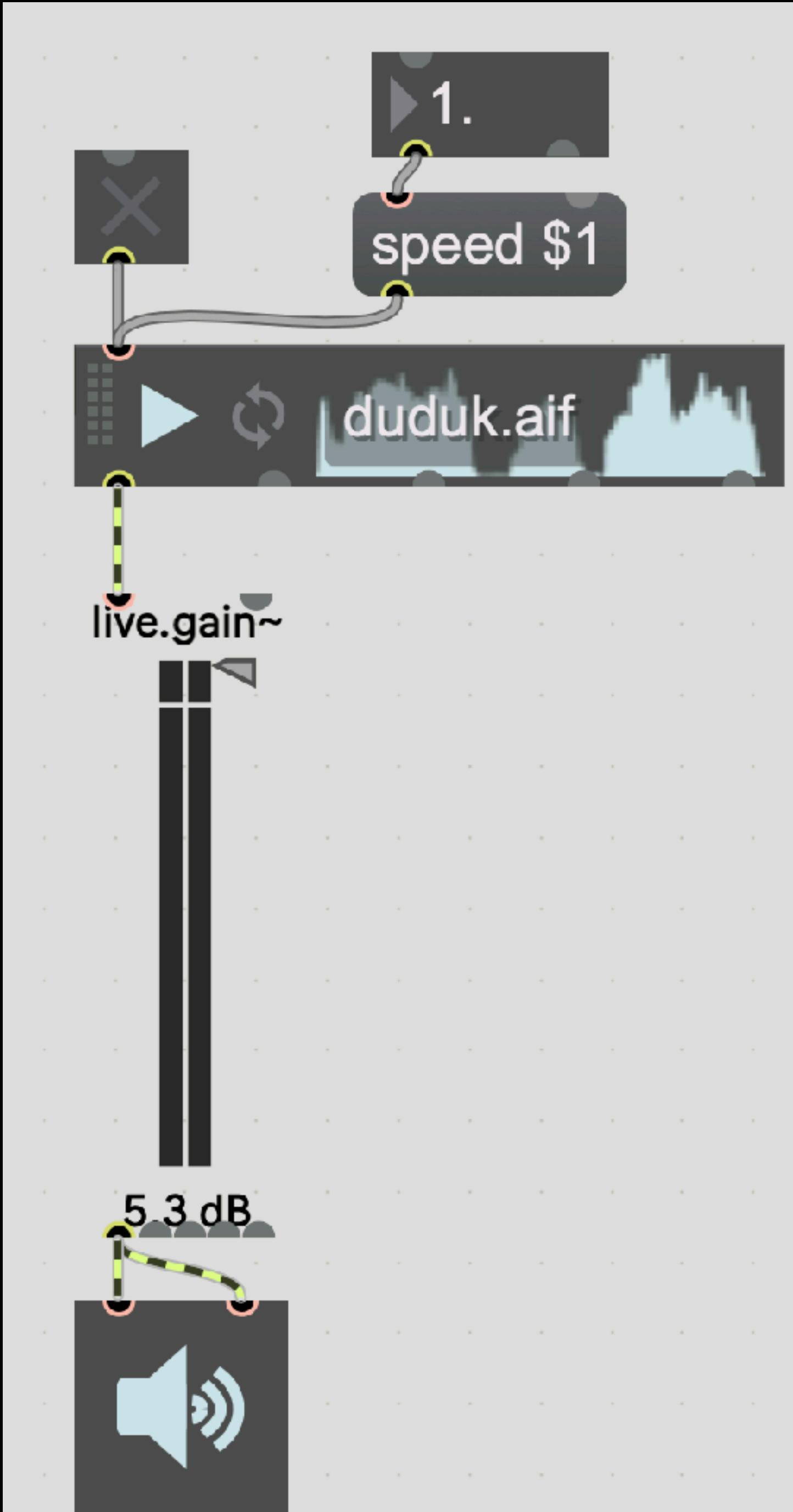


AudioFiles Patch

Attributes can be sent to an object as messages to change its parameters. You can see a full list of them on the documentation for each object online or under Help.

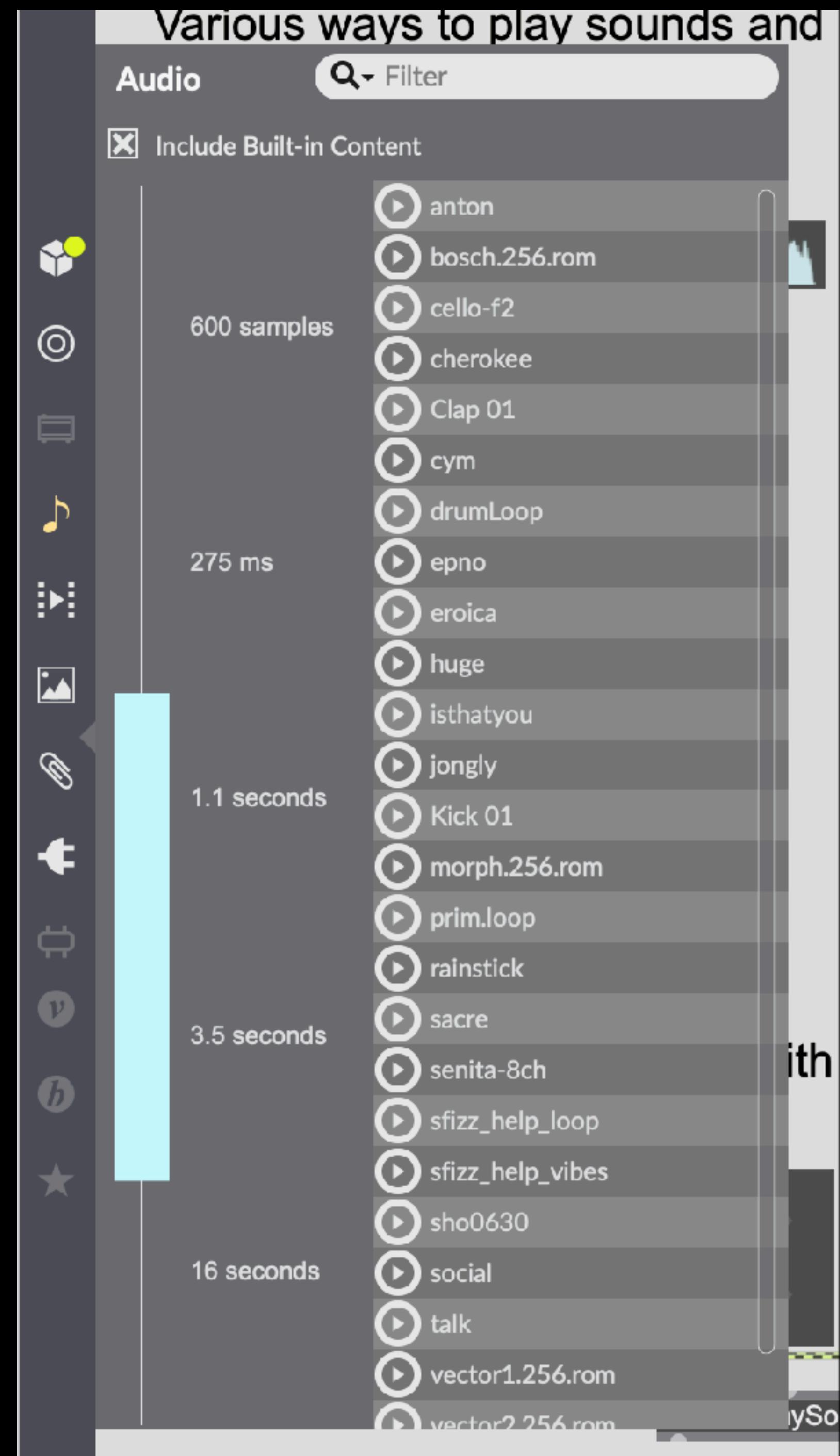
Attributes	
@ accentcolor	- Controls Off Color
@ allowreorder	- Allow the re-ordering of clips
@ basictuning	- Basic tuning (from A)
@ bgcolor	- Background Color
@ channelcount	- Number of audio channels
@ clipheight	- Height allotted for each clip to be displayed
@ color	- Waveform/Controls Color
@ elementcolor	- Clip Divider Color
@ expansion	- Style of accomodation for adding clips
@ followglobaltempo	- Follow global tempo
@ formant	- Formant Scale
@ formantcorrection	- Turn on formant correction
@ loop	- Turn looping on/off
@ looperport	- Loop Report
@ mode	- Timestretching Mode
@ name	- Name
@ originallength	- Original Length
@ originaltempo	- Original Tempo

Here, we are using a variable with \$ to be able to change the speed dynamically.



AudioFiles Patch

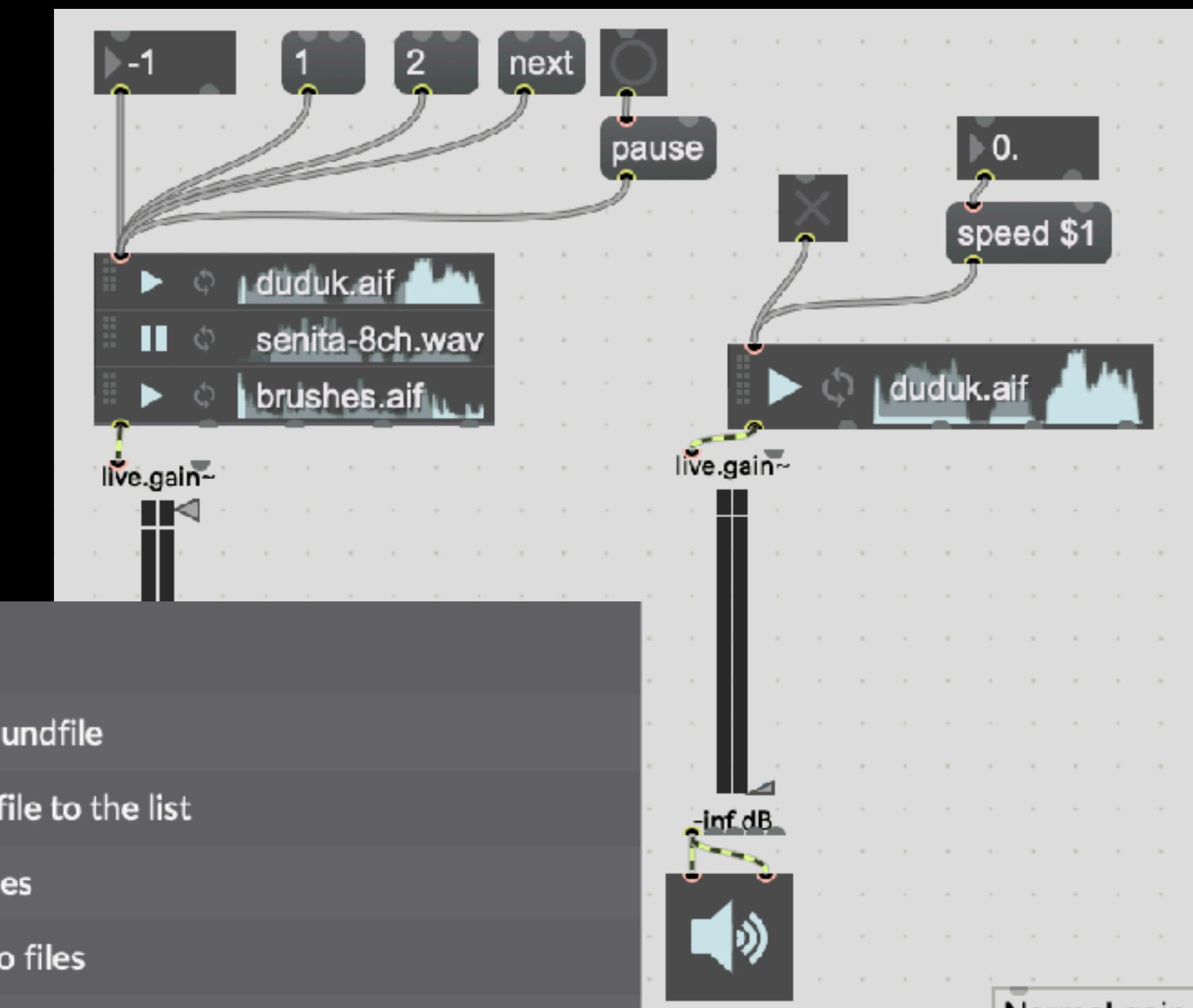
Max comes with some audio files, but you can also add your own. Make sure to put them in the same folder as your patch if you want to access them easily. You can drag and drop these into the patch.



AudioFiles Patch

You can send messages to objects to do actions. You can find them all in the help file under messages. Messages can be triggered using a click, bang, other input. You can also use variables to change things like speed.

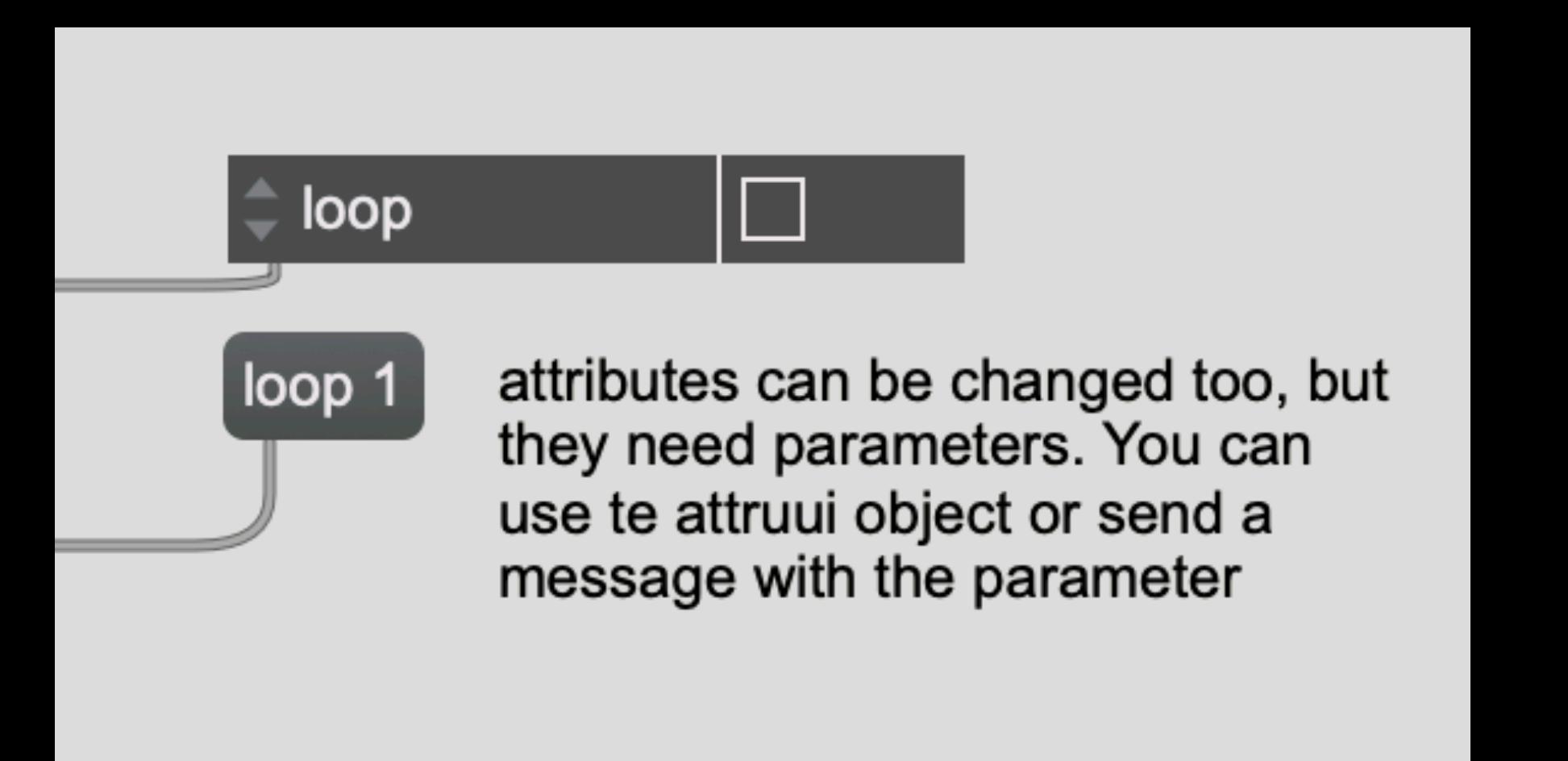
▼ Messages	
M	int - Start/stop playing a soundfile
M	append - Add a new soundfile to the list
M	clear - Remove all sound files
M	(drag) - Drag and drop audio files
M	getcontent - Get the content and behavior of all clips
M	(mouse) - Press play/loop and make selections
M	next - Play the next clip
M	pause - Pause playback
M	remove - Remove a clip
M	resume - Resume playback
M	selection - Select playback endpoints for a clip
M	selectionms - Select playback endpoints for a clip in milliseconds
M	setclip - Set an attribute state for a specific clip
M	signal - Trigger a stored cue with sample accuracy



Normal gain

AudioFiles Patch

You can also modify attributes, but they need a parameter. You can either access them with the attrui object or as a message



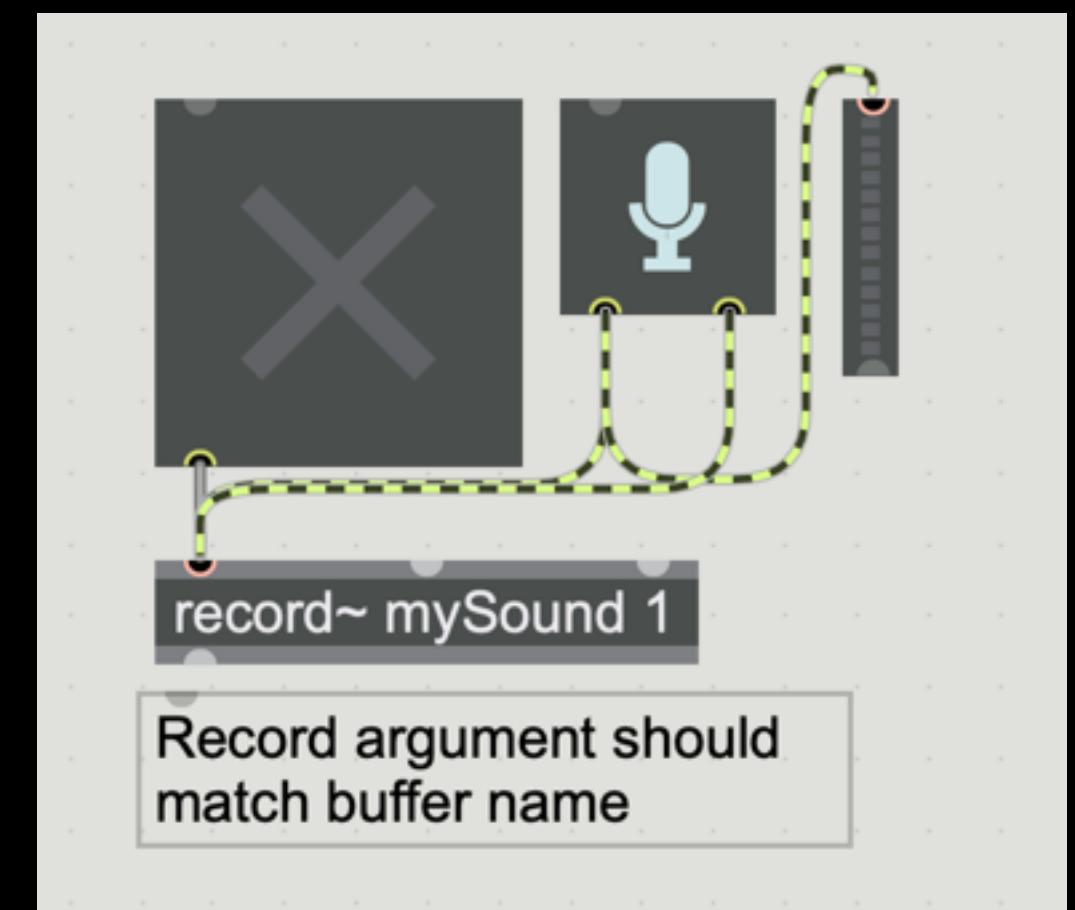
attributes can be changed too, but they need parameters. You can use the attrui object or send a message with the parameter

The Attrui object interface is shown. It has a dark grey header bar with the word 'Attrui' in white. Below the header is a list of attributes, each preceded by a blue '@' symbol and a brief description:

- @ accentcolor - Controls Off Color
- @ allowreorder - Allow the re-ordering of clips
- @ basictuning - Basic tuning (from A)
- @ bgcolor - Background Color
- @ channelcount - Number of audio channels
- @ clipheight - Height allotted for each clip to be displayed
- @ color - Waveform/Controls Color
- @ elementcolor - Clip Divider Color
- @ expansion - Style of accommodation for adding clips
- @ followglobaltempo - Follow global tempo
- @ formant - Formant Scale
- @ formantcorrection - Turn on formant correction
- @ loop - Turn looping on/off
- @ loopreport - Loop Report
- @ mode - Timestretching Mode
- @ name - Name

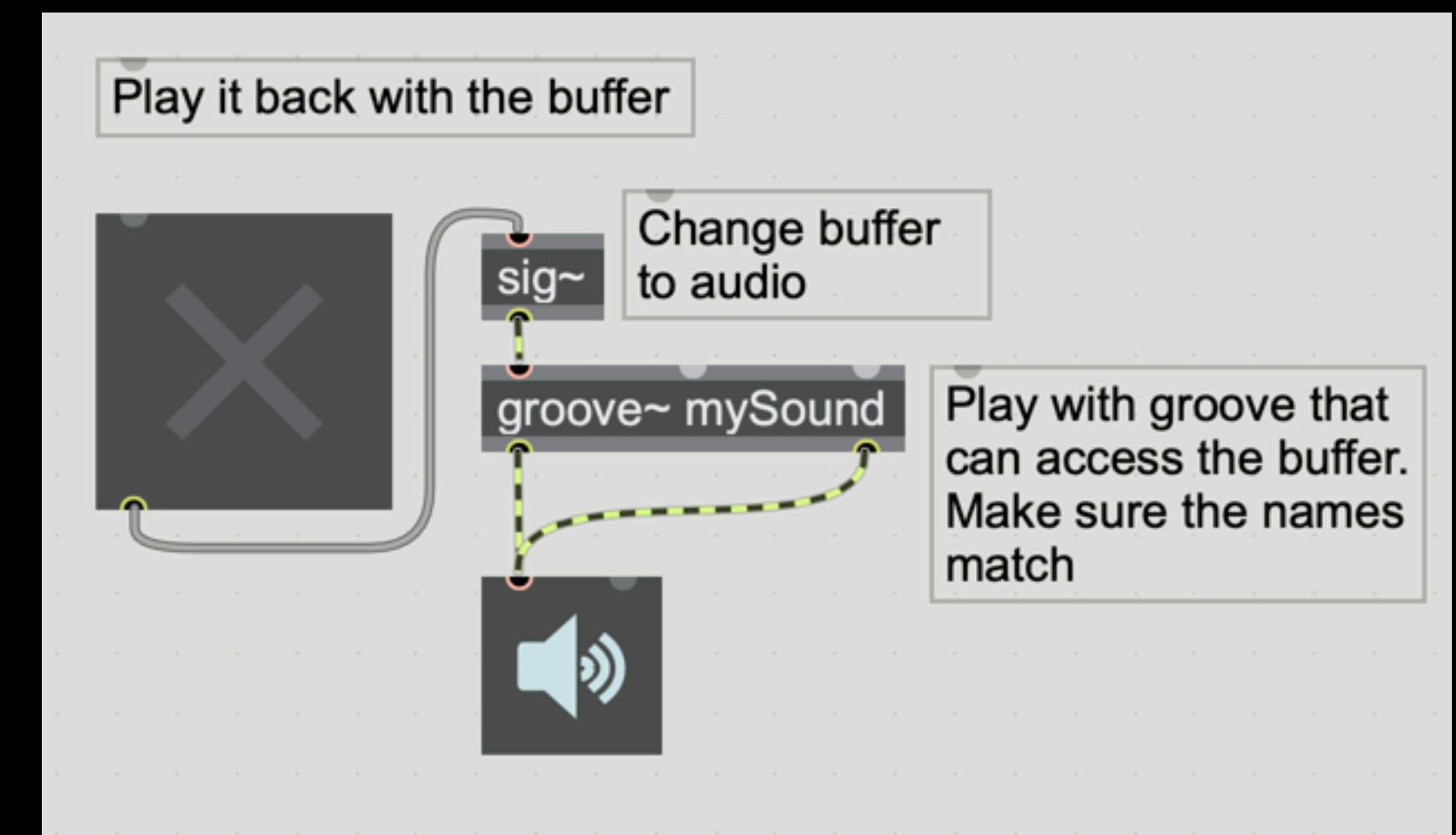
AudioFiles Patch

You can record audio using `record~`. You'll need to create a buffer to hold the audio. You need to give the buffer a name and a size (in milliseconds) and it is not connected to the patch



Buffer keeps mySound stored for 5000 milliseconds
buffer~ mySound 5000

You can then play it back using `groove~` which can play from the buffer. `Sig~` makes the buffer data back into audio.

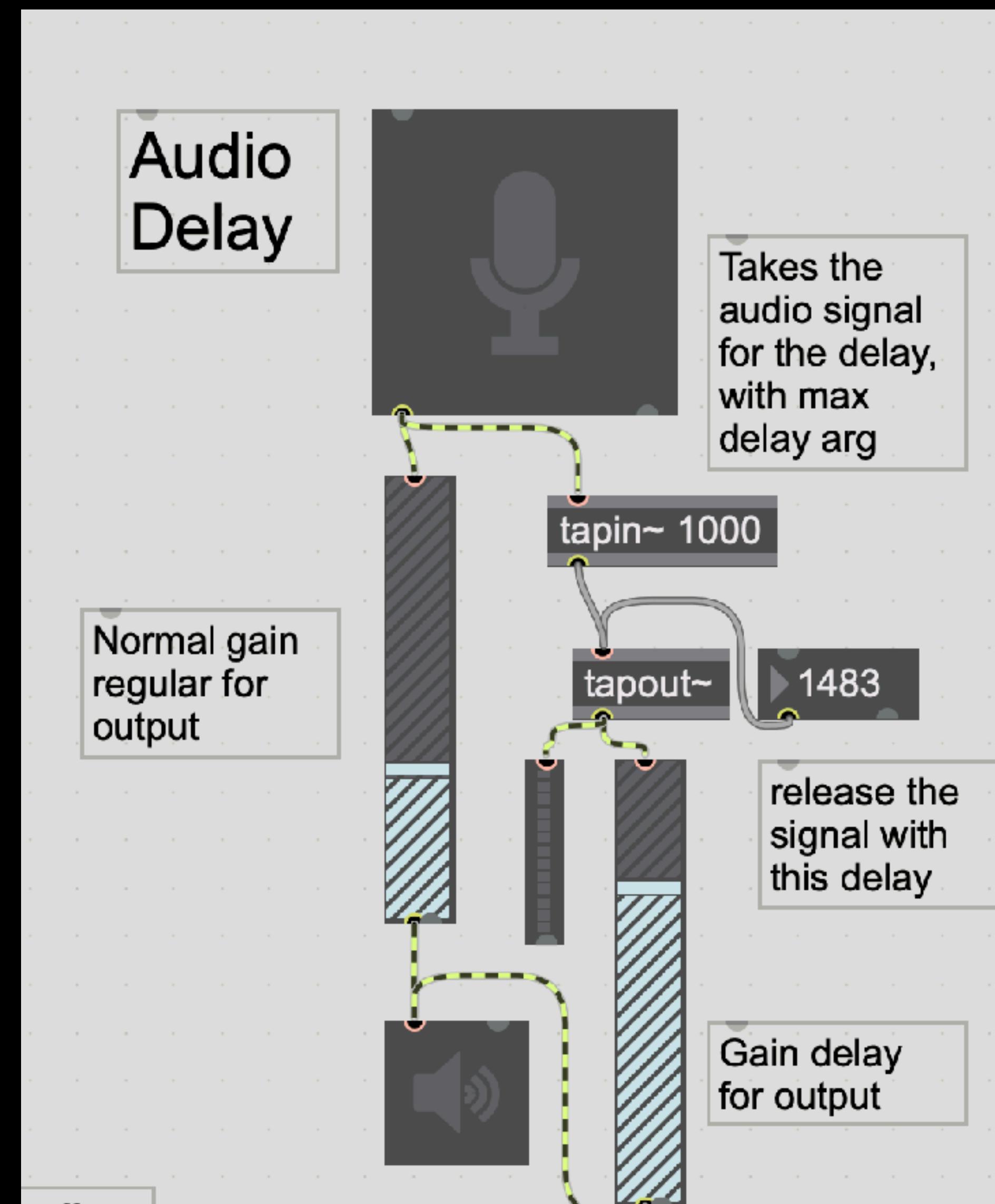


AudioFiles Patch

You can record use live input and feedback with a delay using `tapin~` and `tapout~`.

`Tapin~` has an optional argument for the max delay.

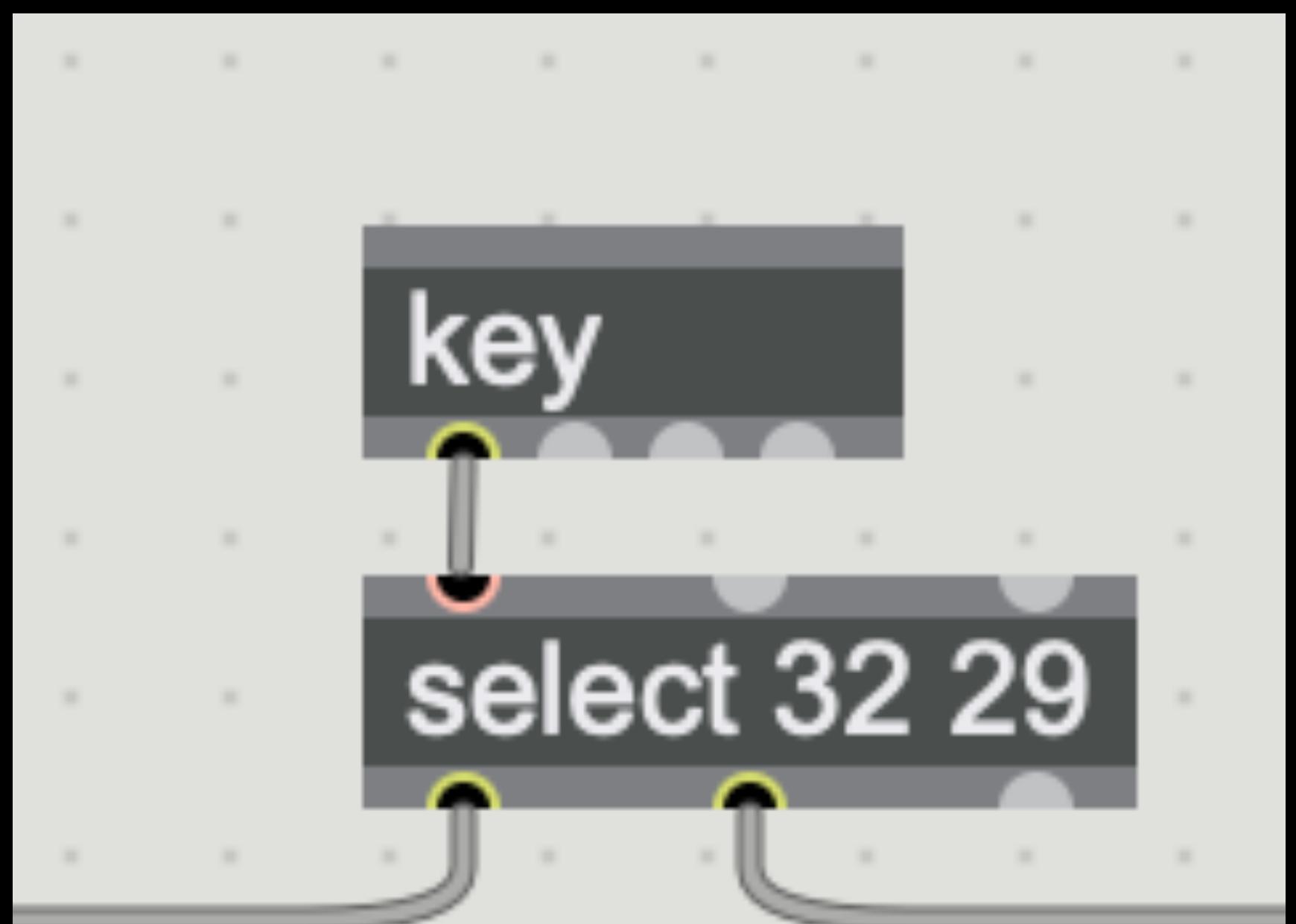
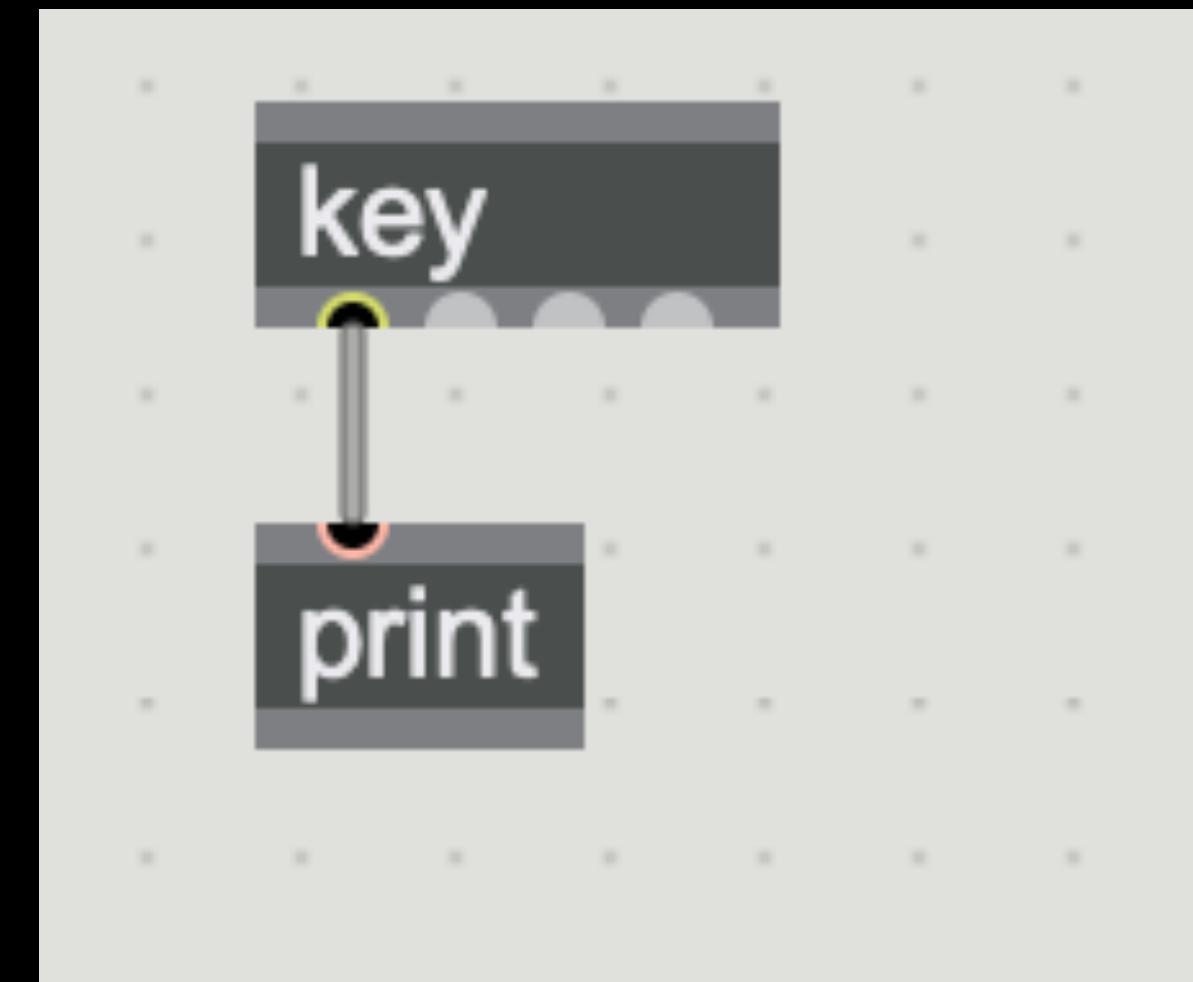
You can pass two audio feeds, one is normal and the other is delayed to create feedback loops.



Select Audio Patch

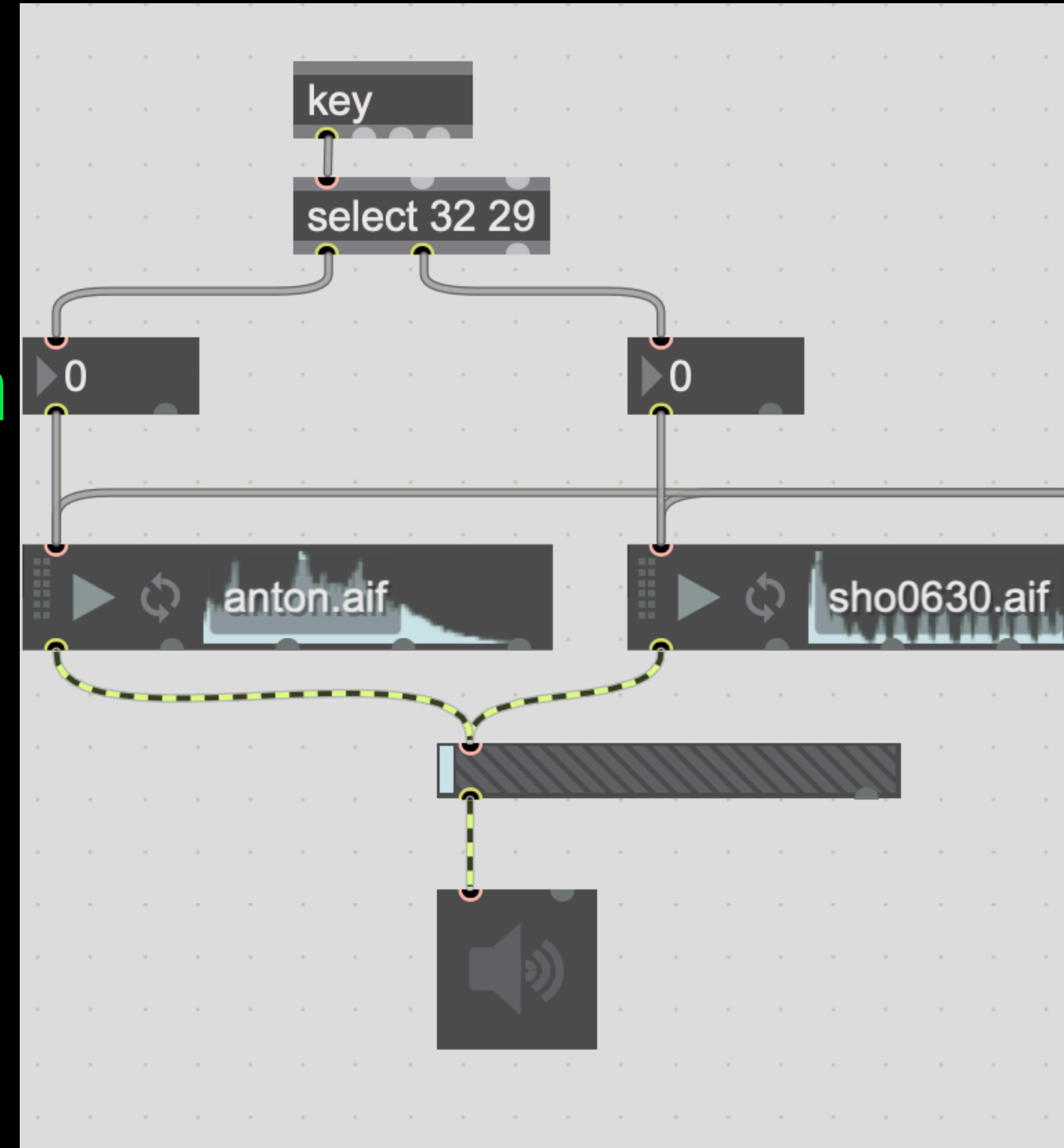
You can use key to detect which key is pressed on the keyboard.

Keys are defined by ASCII codes. You can see them in print, number OR find a chart here. <https://theasciicode.com.ar/>



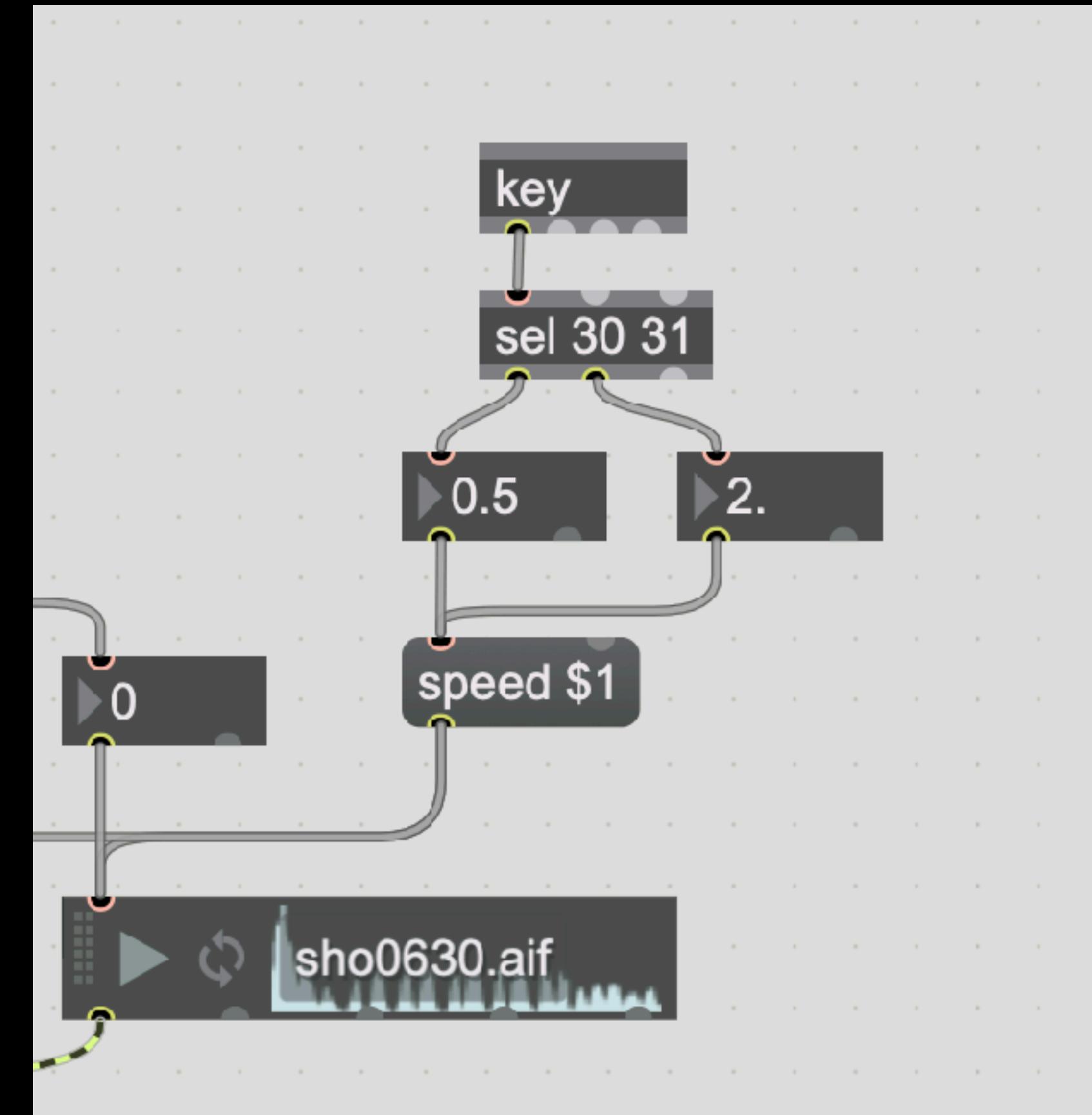
Select Patch

We can use select (or sel) to select different outputs from key presses. Add them as arguments. Here, I am using 32 and 29, space bar and right arrow. When the sel object receives them, it sends a bang through the appropriate outlet.



Select Patch

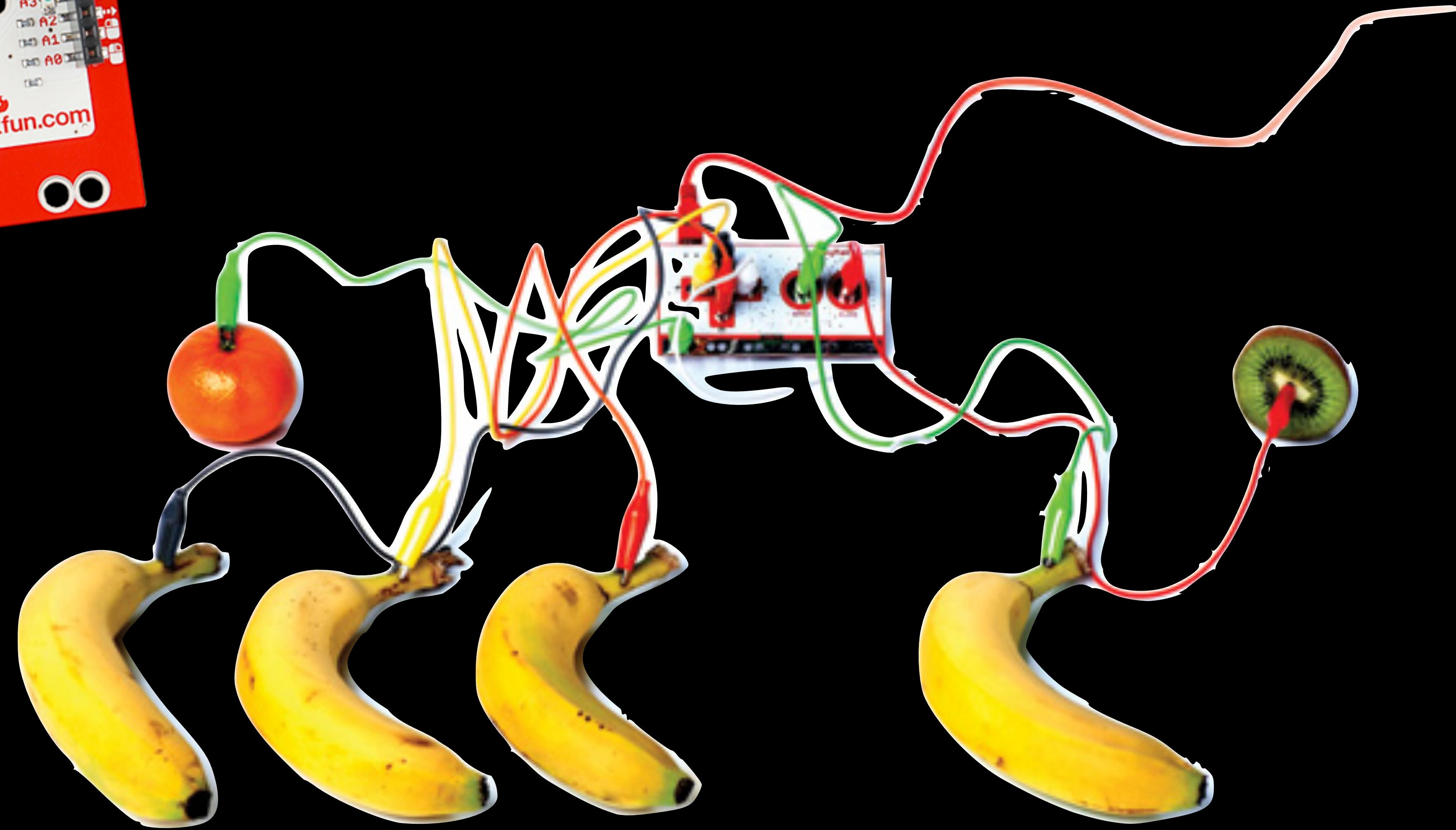
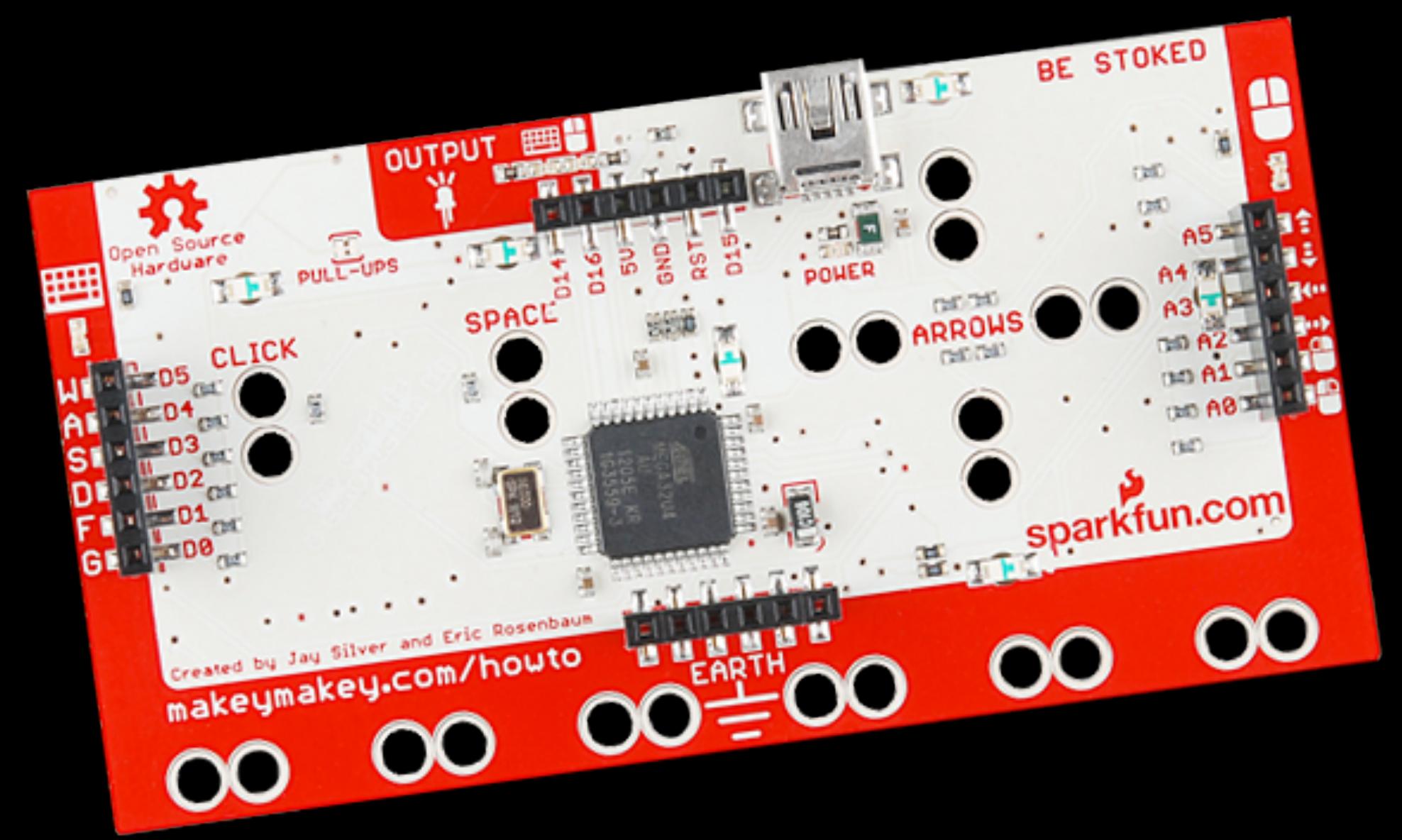
I can use the same concept to change the speed of the playback. Here, we are holding the incoming value in \$1 and then sending it as a speed message.



Introducing: Makey makey

This is a tool that simulates a keyboard using capacitive touch. You can essentially turn almost any object into a keyboard. It shows up as a keyboard on your computer.





Makey Makey Exploration Time:
In groups, make something with the
MakeyMakey

MIDI Samples patch

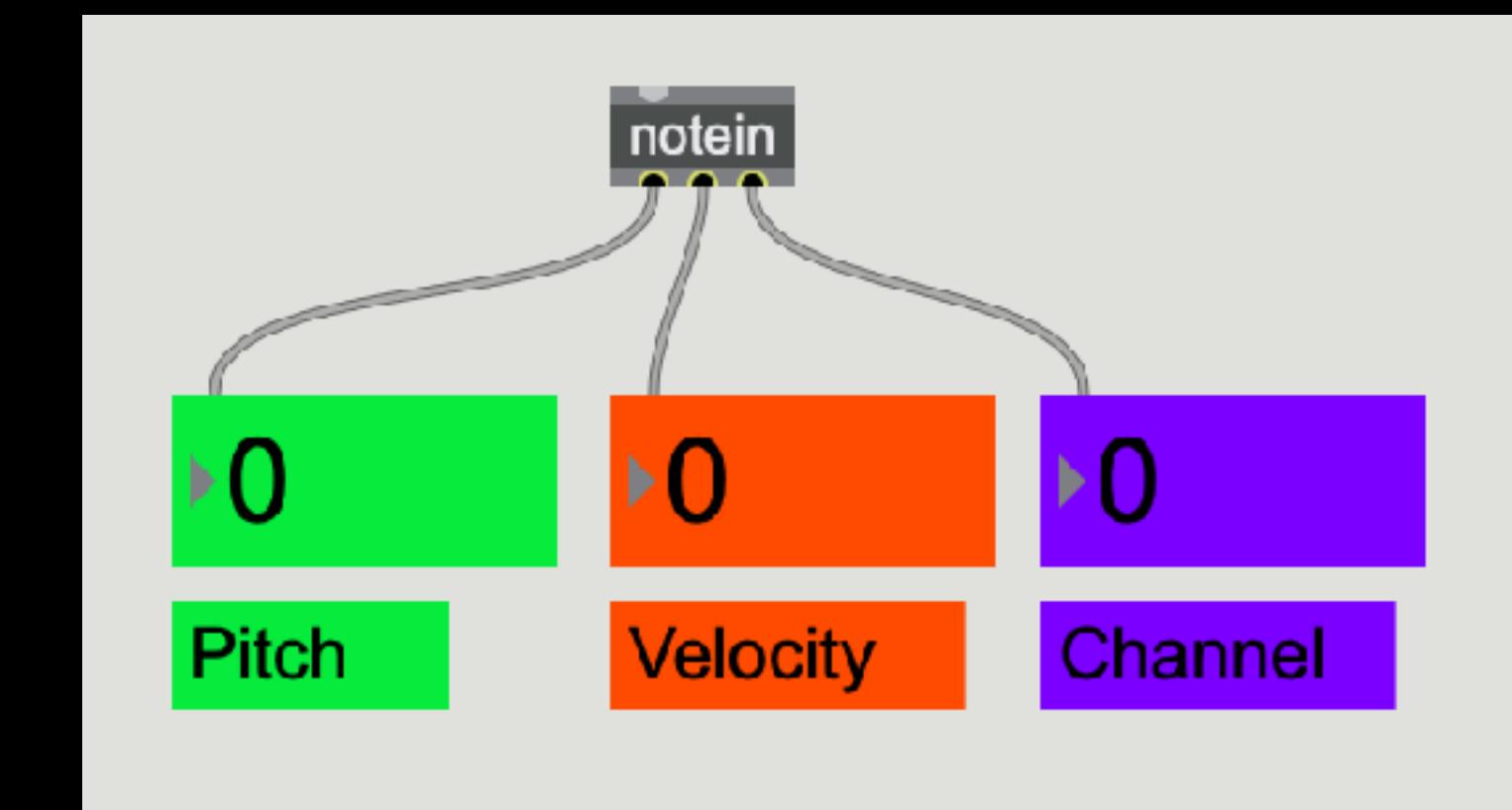
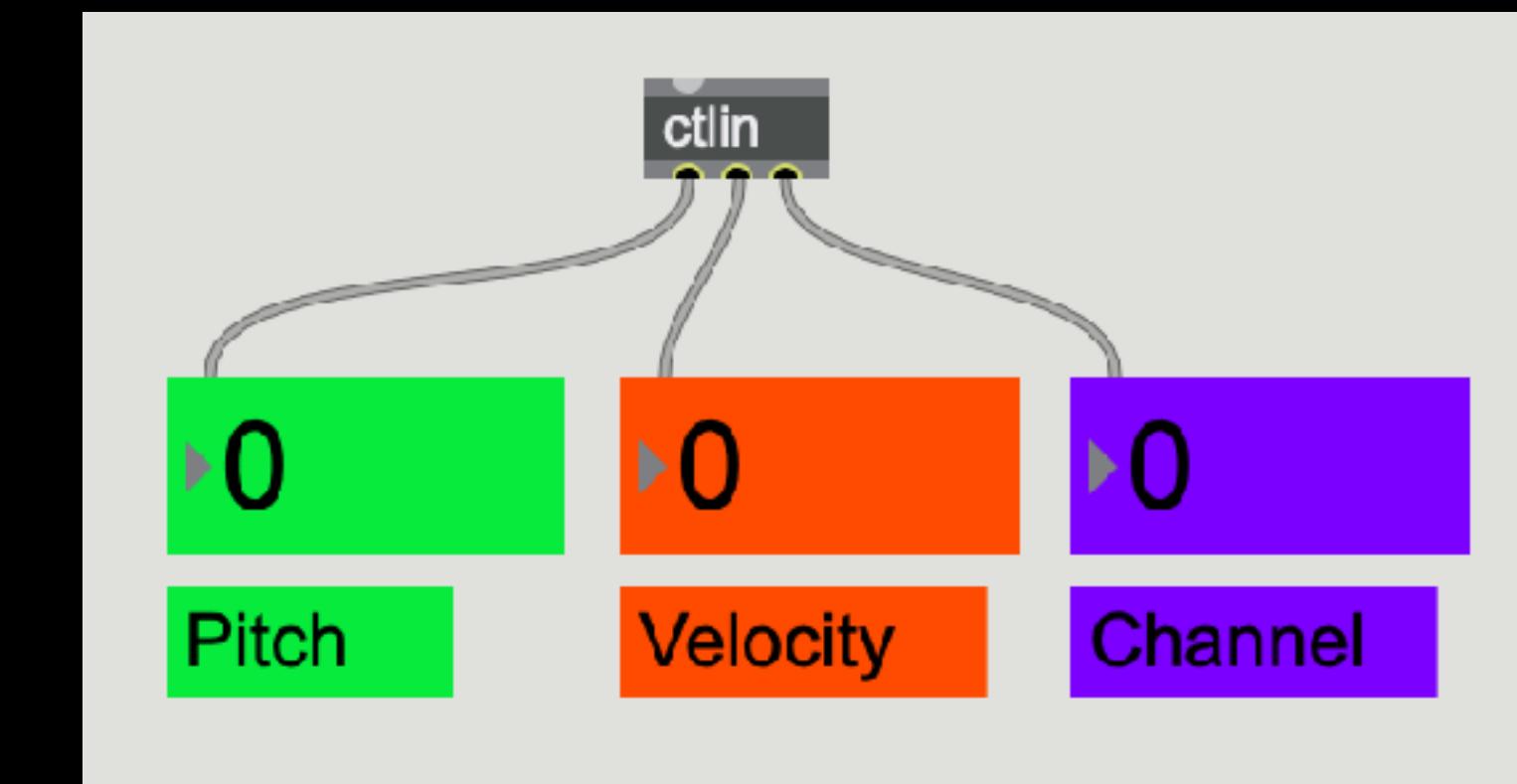
Each midi note contains:

Pitch: key/note/value

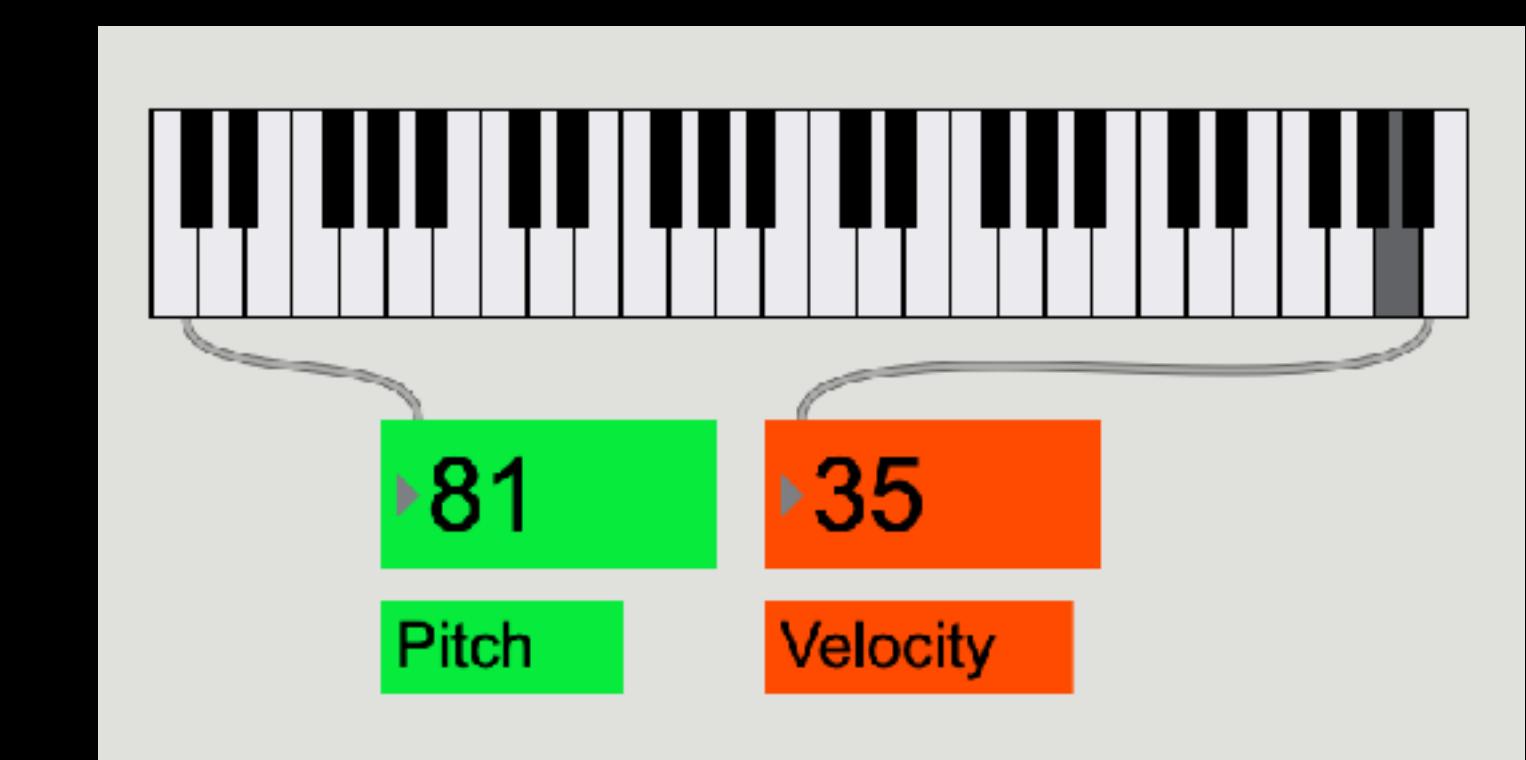
Velocity: intensity

Channel: channel

You can use notein or ctln to get notes or control knobs

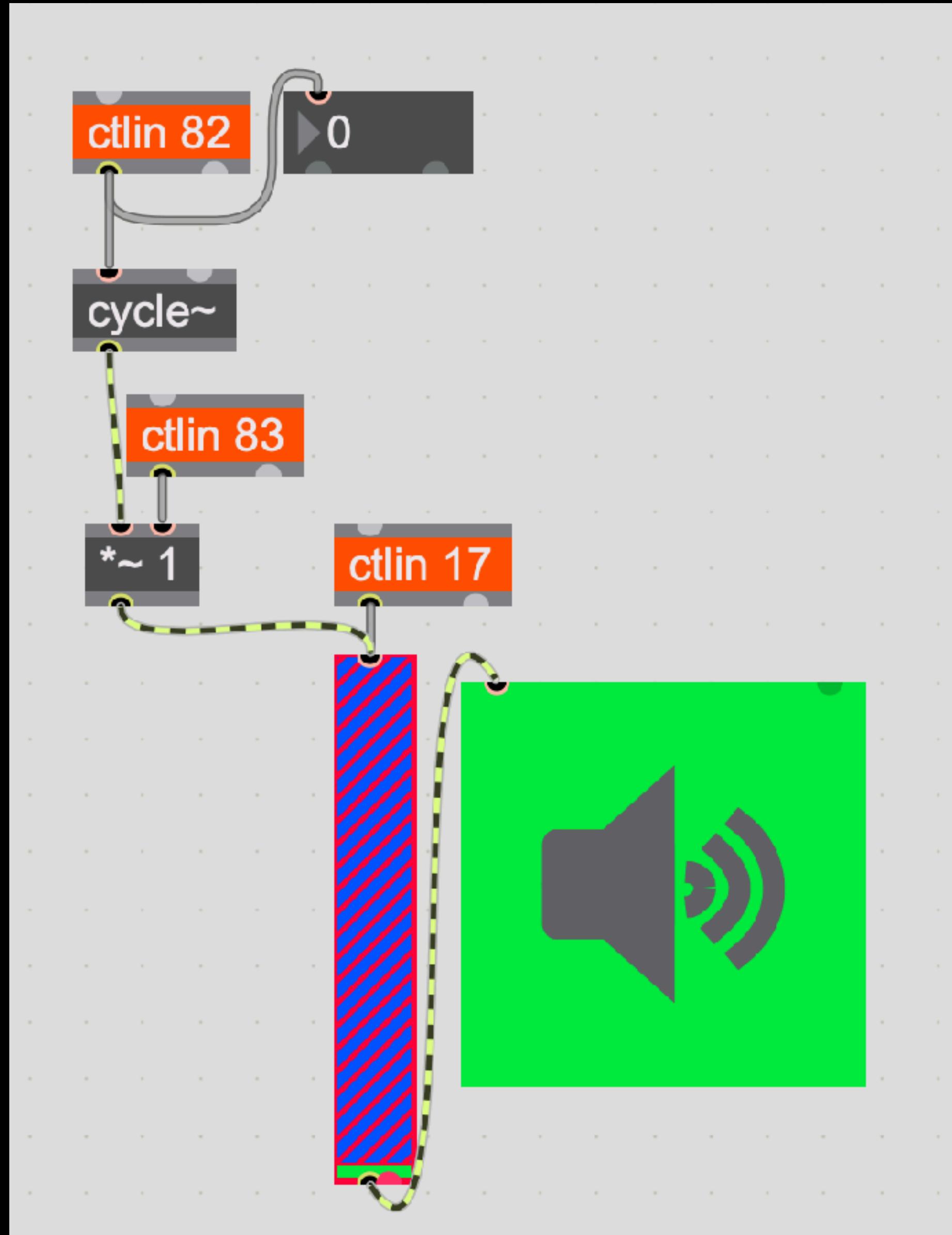


KSlide: This is just a UI. It can be numbers, it can be used to simulate your keyboard, but it is just a visual, nothing special. kslide uses velocity from the height of the key press on a key



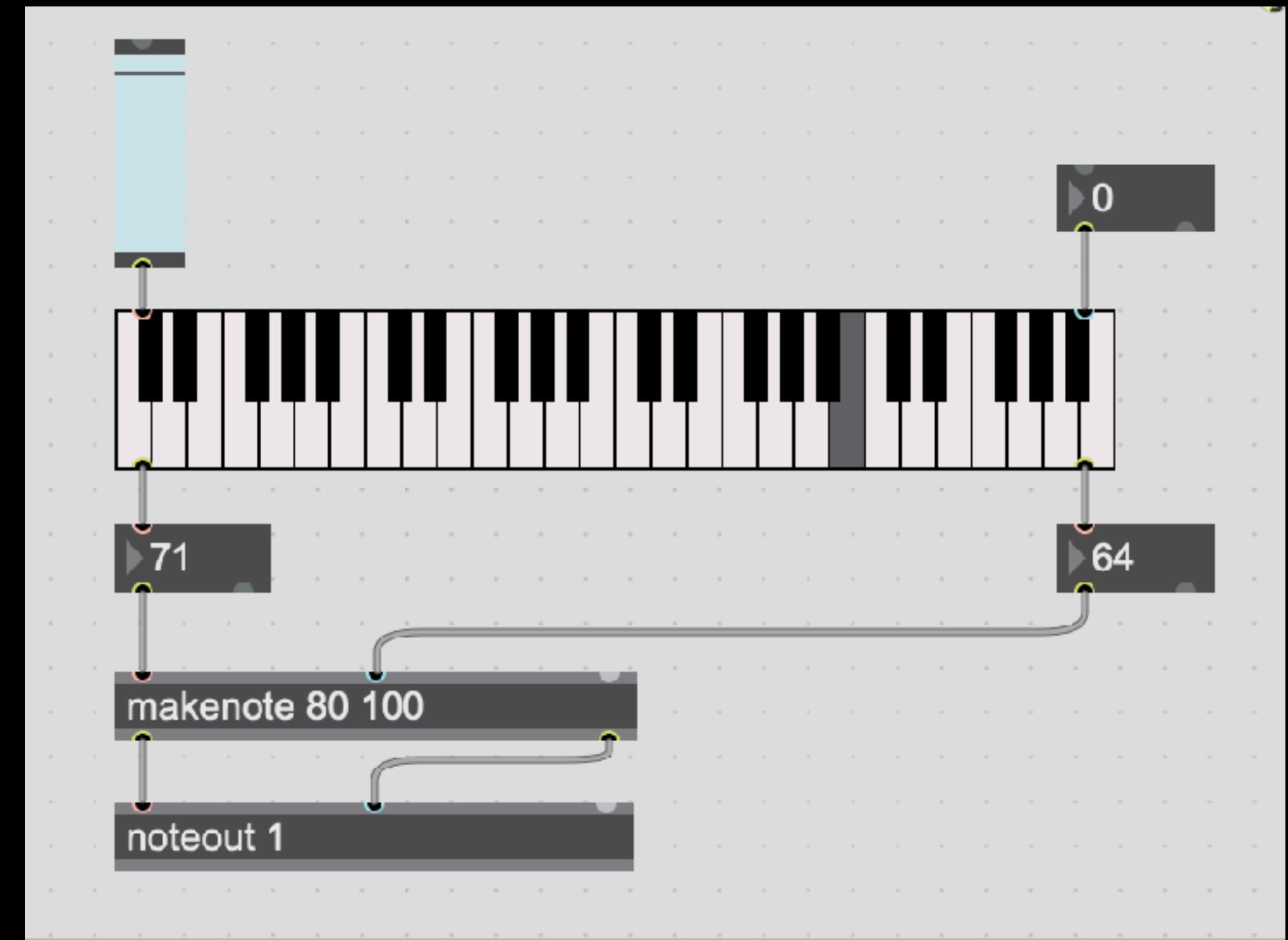
MIDI Samples patch

You can use ctlin to define a channel to listen for and control an oscillator with various knobs on your midi controller . This example uses various sliders on my controller to change the volume, amplitude and frequency of this cycle~

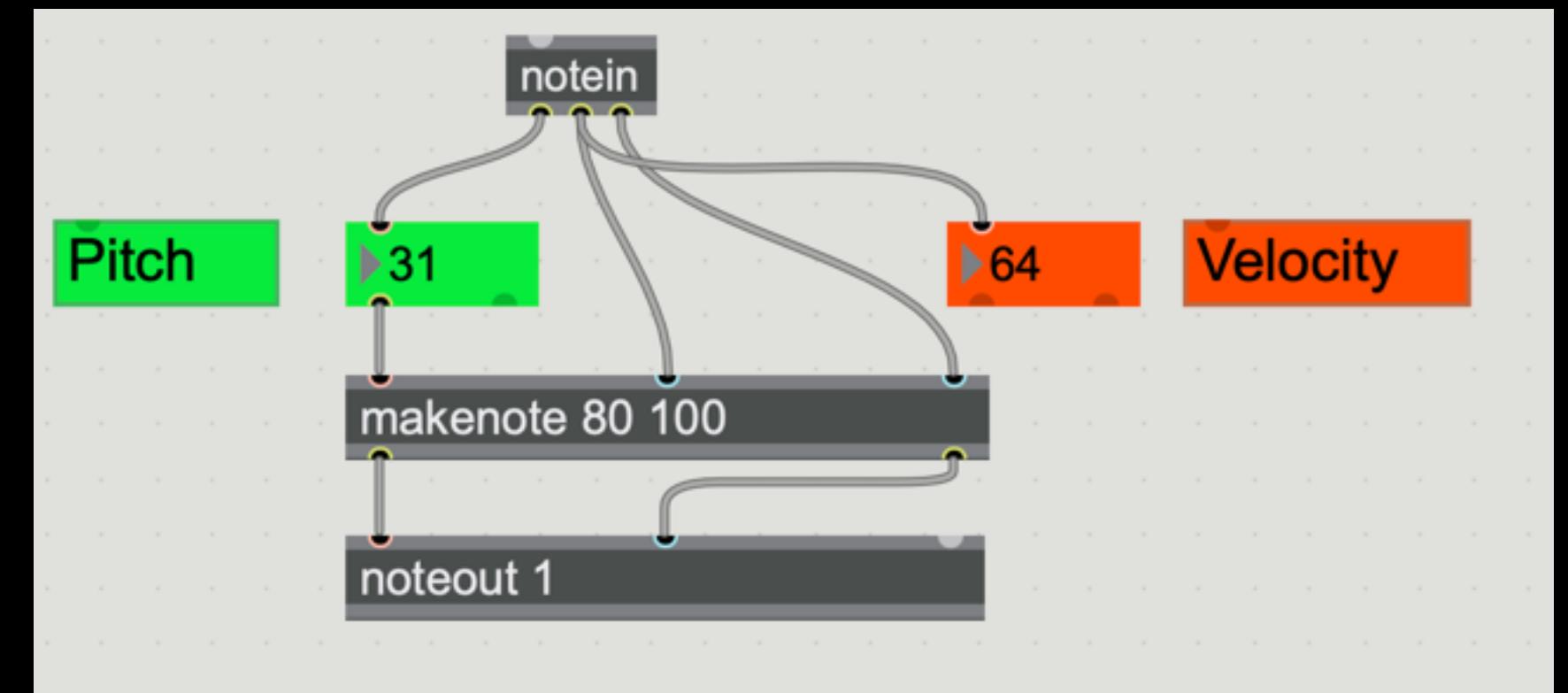


MIDI Samples patch

Notes only happen when you use **makenote** and **noteout**. You can use midi data to do anything, but if you want to produce a midi note you need to use these.



This is an example that uses the KSlide UI, and another that just uses the notein data.



MIDI Sequencer patch

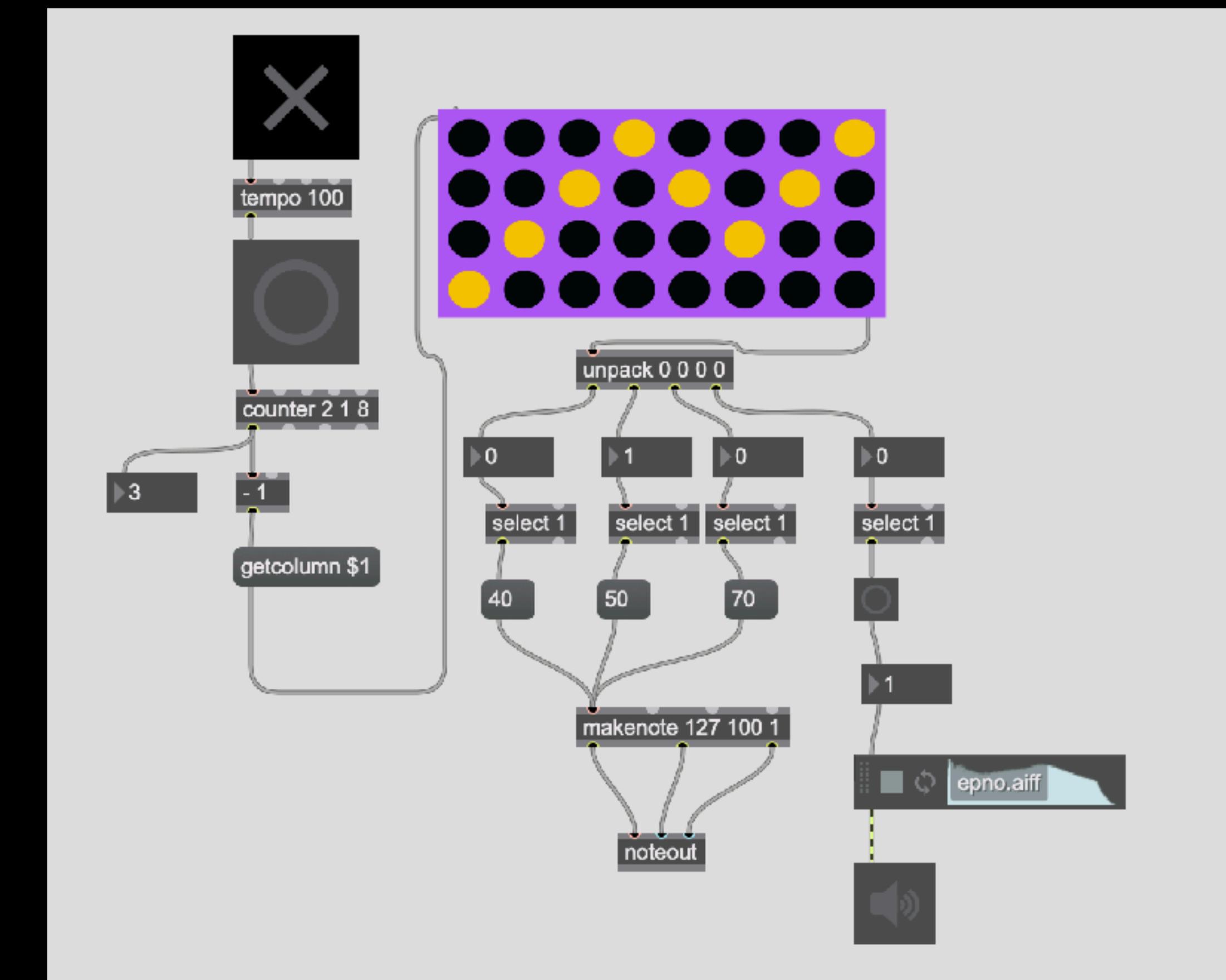
This is a basic sequencer. It uses a variety of tools :

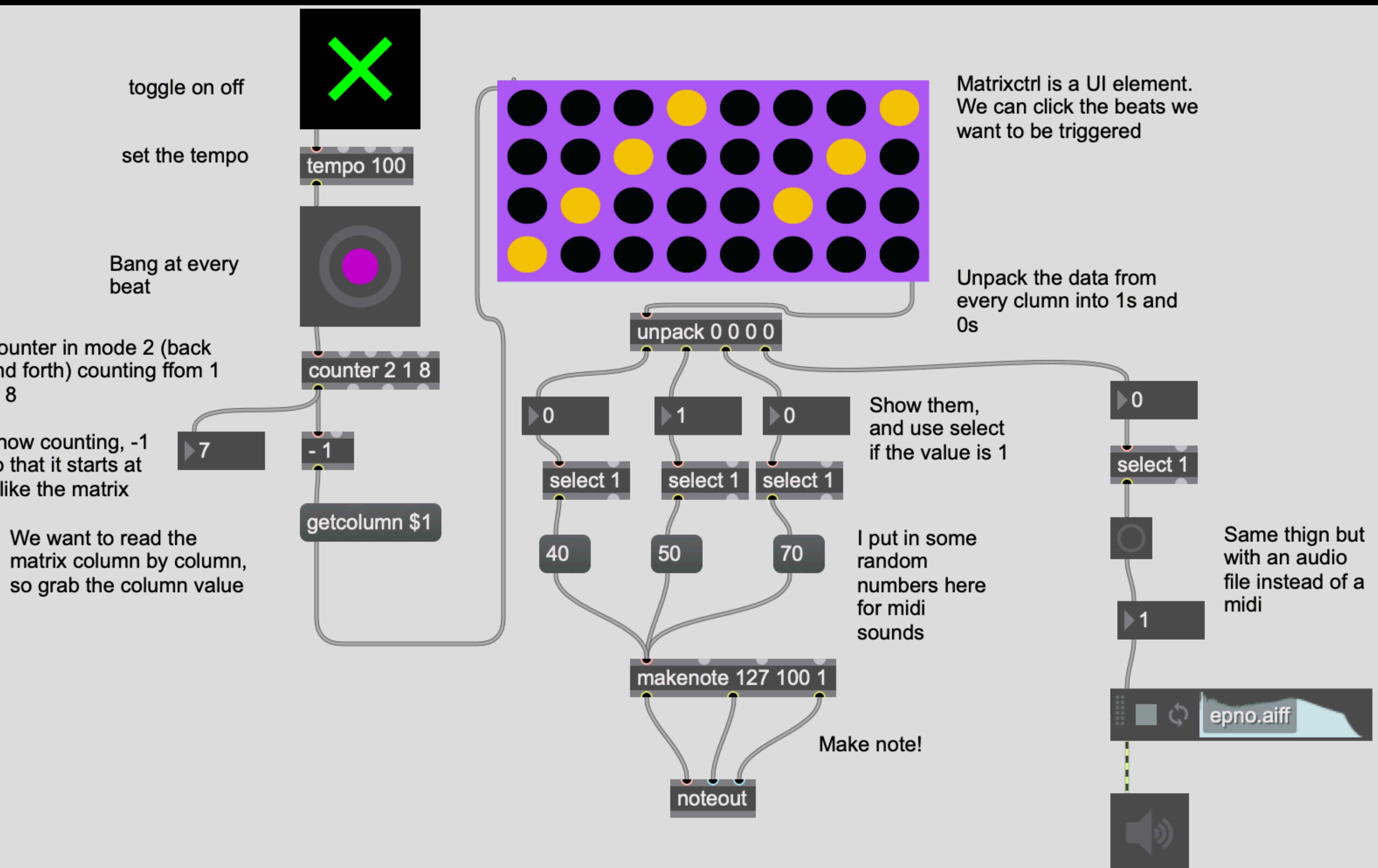
Tempo: Set the BPM

Counter: counts between a range in a variety of ways

Matrixctrl: A UI for control

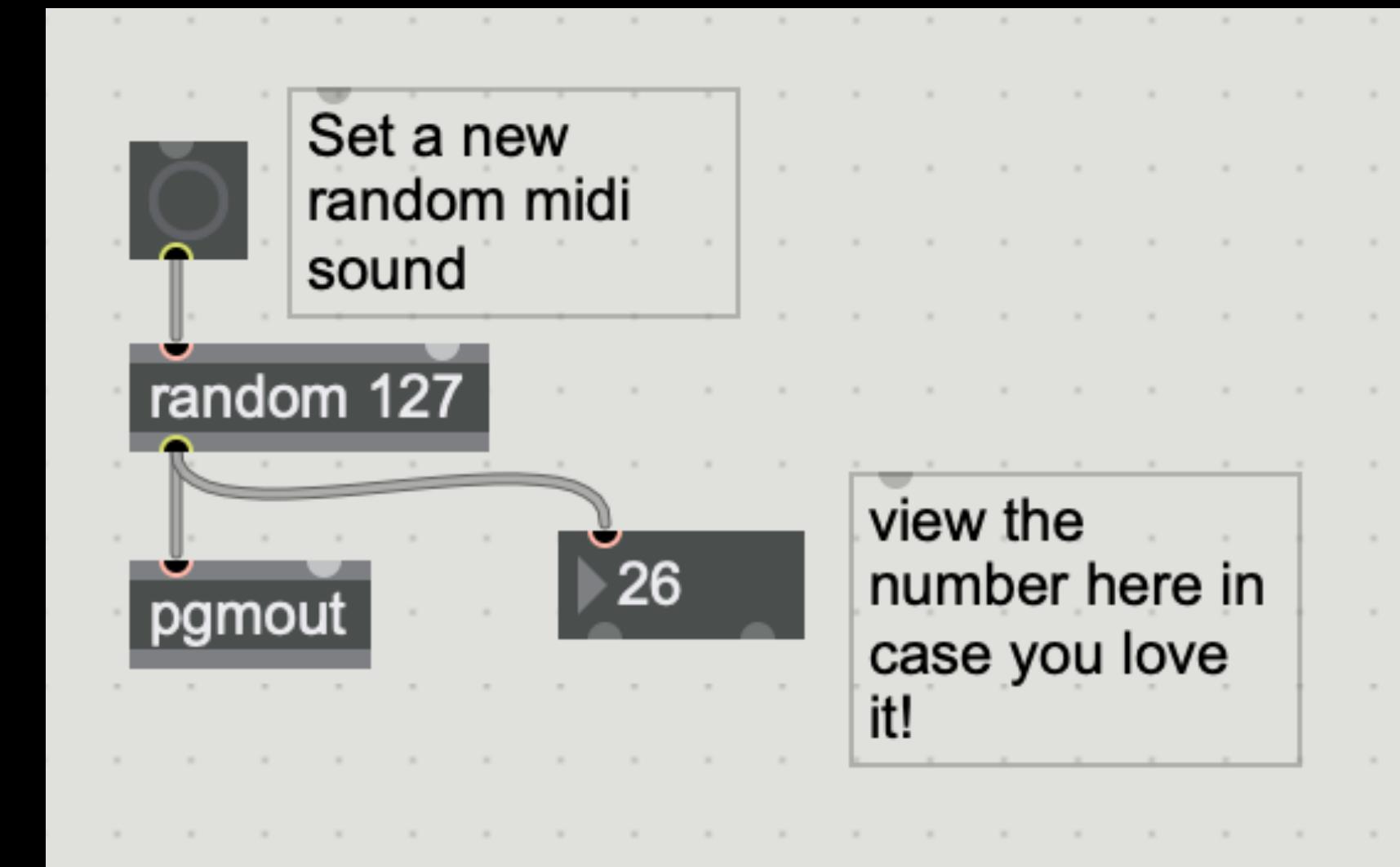
Unpack: Shows complex data from an output





MIDI Sequencer patch

Pgmout sets a new random midi value, you can use this snippet to click the button and load a new one.



**Download todays patch
on Moodle**

Handing in projects

Each project should include a .zip file that contains:

- A saved file (File > Save as Project) .maxpatch
- A screen capture of your max patch
- A video or audio recording of your project working (can be a video or a link to a private video on YouTube, Vimeo or Google)
- A 50-100 word explanation of your project inside your maxpatch

Be sure to name files properly (no untitled-1.zip)

All files are expected to be cleaned up and arranged in a reasonable, legible way. Videos should be clear, well light and show your project working.

Homework:

Work on Audio Experiment (due next week)

Begin to form teams for mid term (max 3 people, 2 is ideal)