

Is There an If?

for flute and electronics

documentation for electronics

Shu-Yu Lin

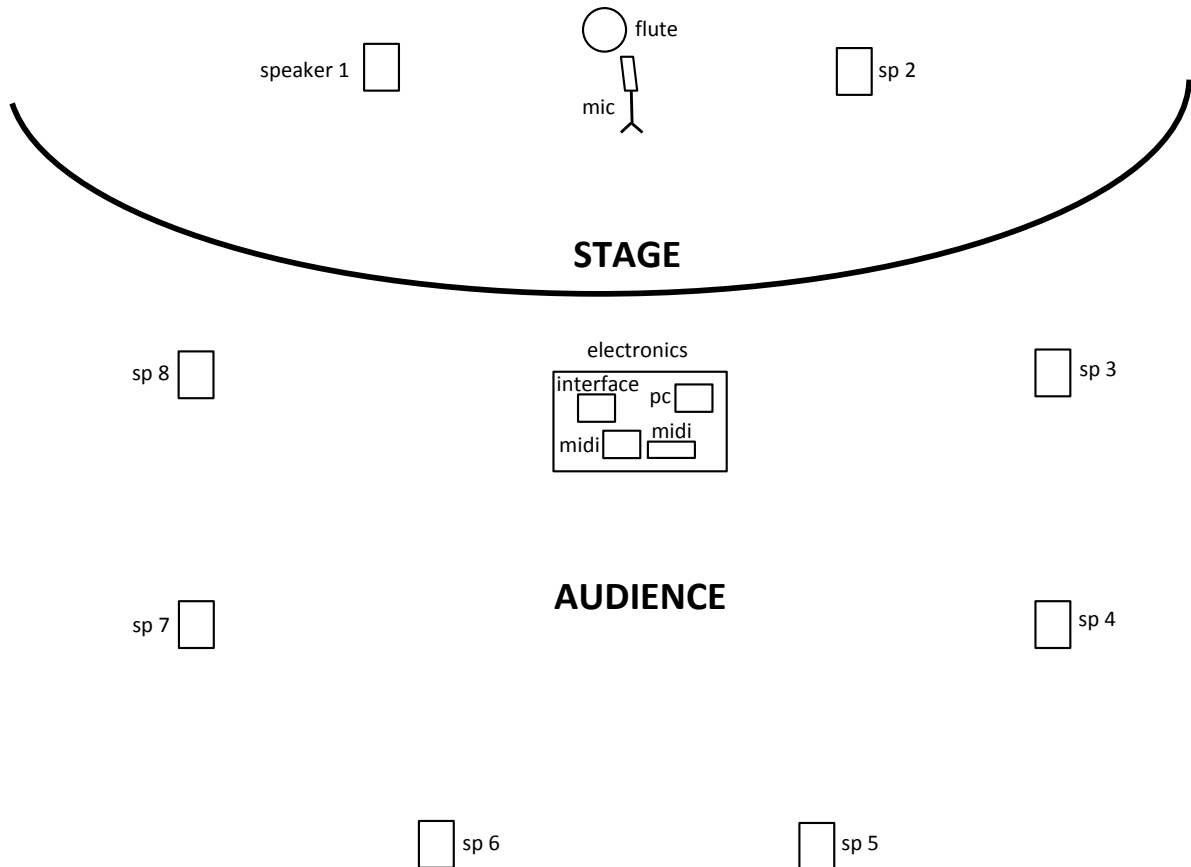
Table of Contents

Technical specification	3
Stage plan	3
Performance instruction	4
Hardware setup	10
SuperCollider code instruction	14
Speaker test	25

Technical specification

- A midi controller with at least 8 turn knobs (eg. Akai LPD 8)
- A push button controller with at least 68 buttons and 8 sliders (eg. Akai APC mini)
- 8 mid-range speakers
- An audio interface with at least 8 outs and 1 in
- 1 dynamic microphones (eg. Shure SM58)
- Laptop computer
- SuperCollider Programming environment

Stage plan



Performance Instruction

The electronics performer operates the SuperCollider script and midi controllers to trigger electronics processing at particular times that indicate on the score.

The Akai APC mini and Akai LPD 8, or controllers of your choice in place of these two, are suggested to be placed on a table with the laptop computer as the electronics operations are controlled using these devices.

Explanatory notes for score

①	<p>Press button 1 on controller to trigger electronics; '2' refers to press button 2 etc. Press it again to stop processing.</p>
(f-1-8)	<p>Press button f-1-8 to amplify flute through speakers 1, 2, 3, 4, 5, 6, 7 and 8. Button f1 refers to amplify flute through speaker 1, f2 through speaker 2 etc. Button f-o refers to amplify through odd number speakers. Button f-e refers to amplify through even number speakers. f1-p amplification consists of panning function which can be controlled using knob 1 (further explanation see below regarding knobs). Press it again to stop amplification.</p>
rec1-17s	<p>Press button rec1-17s to start record flute for 17 seconds; rec1-2s refers to record flute for 2 seconds, rec1-5s refers to record flute for 5 seconds etc. Names of the recording buffers are in sequence. For example, rec1-2s and rec2-2s both record for 2 seconds but into separate buffers. Button will be auto off at the end of the recording. There is no need to press it again to stop.</p>
(p-rec1-17s)	<p>Press button p-rec1-17s to playback recording rec1-17s. Button will be auto off at the end of the playback. There is no need to press it again to stop.</p>
k1	<p>Turn knob 1 (k1) on controller accordingly to modify computer-generated sound; k2 refers to knob 2 etc. See below for further detail on the knob control parameter.</p>
S	<p>Adjust slider on controller accordingly to modify output amplification. See below for further detail on the slider control parameter.</p>

Midi controllers

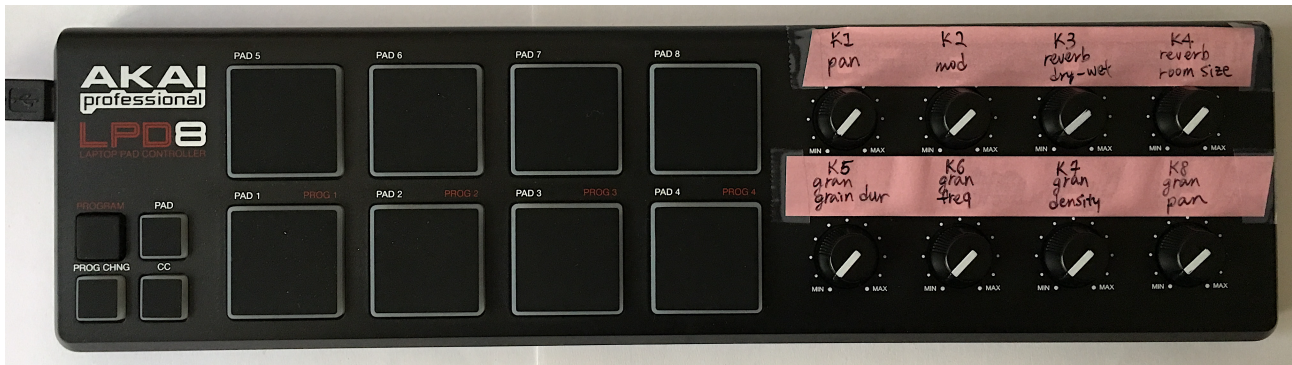
- Akai APC mini

Electronics performers are instructed to press a certain button on the push button controller to begin a particular electronics processing. As indicated in the explanatory notes above, circular shapes enclose button numbers, oval shapes enclose amplification processes, rectangular shapes enclose recording processes and oval shapes enclose playbacks of recordings. Each slider on the APC mini corresponds to the numbers above. For example, the slider 1 (counting from the left) controls the volume of the processing of the buttons f1, f-o, 1, 9, rec1-2s, p-rec1-2s, rec2-4s, p-rec2-4s and 1, and slider 2 controls the volume of the processing of the buttons f2, f-e, 2, 10, rec2-2s, p-rec2-2s, rec3-4s, p-rec3-4s and 2 etc.



- Akai LPD 8

The electronics performer is instructed to turn a specific knob at a particular time according to the score to operate the electronics processing. K1 refers knob 1, K2 refers to knob 2 etc.



Electronics control parameters and preparation

- Knob:

Knobs 1 and 8 should be turned to 0 as the default before the performance.

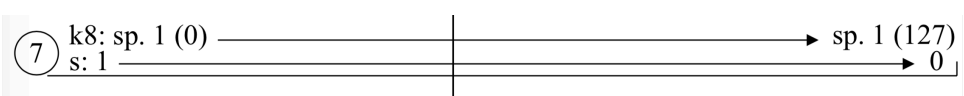
Knobs 3, 4, 5, 6, 7 should be turned to 1/2 (63.5 of 127) as the default. Electronics performer is free to adjust the reverberation according to the characteristics of the performance venue.

Knob 1 (k1) controls the panning of the input signal and knob 8 (k8) controls the panning of the granular synth output. On the score, speaker number is specified with midi output number, which is shown in the post window in SuperCollider script, for panning precision. If the panning midi number specified 71, pan to values between 69-73 is acceptable.

The following example shows the instruction to press the playback button p-rec1-17s with k1 pan to speaker 5 and 6, in which the midi value is 71.

```
p-rec1-17s k1: sp. 5, 6 (71)
```

The next example shows the instruction to press button number 7 with granular panning control k8 with specified starting speaker 1, which the midi number is 0, and pan to speaker 1 to complete a panning cycle, which the midi number is 127.



Knob 3 (k3) controls the dry and wet gain of the reverberation, in which 0 represents the driest, 127 the wettest.

Knob 4 (k4) controls the room size of the reverberation, in which 0 represents the smallest and 127 the largest.

Knob 5 (k5) controls the duration of grain in granular synth. Midi number 0 represents the shortest duration and 127 the longest. Values are written between 0, the minimum, and 1, the maximum. For example, the value 1/4 indicates to adjust k5 to 1/4 of 1 etc.

Knob 6 (k6) controls the frequency of grain in granular synth. Midi number 0 represents the lowest and 127 the highest. Values are written between 0, the minimum, and 1, the maximum. For example, the value 2/4 indicates to adjust k6 to 2/4 of 1 etc.

Knob 7 (k7) controls the cloud density in granular synth. Midi number 0 represents the lightest and 127 the densest. Values are written between 0, the minimum, and 1, the maximum. For example, the value 3/4 indicates to adjust k7 to 3/4 of 1 etc.

The following example demonstrates the instruction to press button 12 and to adjust k6 and k7 to 3/4 and k8 to 127, which is speaker 1.

k6: 3/4 _____
 k7: 3/4 _____
 ⑫ k8: sp. 1 (127) _____

- Slider:

The sliders control the amplitude of the specific processing output. They should be turned to the maximum as default before the performance.

Slider values are written between 0, the minimum, and 1, the maximum. For example, the value 1/4 indicates to adjust slider to 1/4 of 1.

The following example shows the instruction to adjust slider to 3/4 of 1.

④ s: 3/4 _____

Electronics numbering

For reference, the brief explanation of instructions of each button number is provided.

Movement	Electronics button number	Brief description
I	1	Playback pre-recorded tape recording wind_low.wav through speakers 1, 3, 5, 7. Knob controls: k1, k3, k4.
	2	Playback pre-recorded tape recording wind_low.wav through speakers 2, 4, 6, 8. Knob controls: k1, k3, k4.
	3	Playback pre-recorded tape recording wind_high.wav through speakers 1, 3, 5, 7. Knob controls: k1, k3, k4.
	4	Playback pre-recorded tape recording wind_high.wav through speakers 2, 4, 6, 8. Knob controls: k1, k3, k4.
I, II, III	5	Granularize input signal. Knob controls: k3, k4, k5, k7, k8.
I	6	Granularize recording rec1_17s. Knob control: k5, k6, k7, k8.
II	7	Granularize recording rec4_5s. Knob control: k5, k6, k7, k8.
	8	Granularize recording rec5_5s. Knob control: k5, k6, k7, k8.
	9	Granularize recording rec1_4s. Knob control: k5, k6, k7, k8.
	10	Granularize recording rec1_22s. Knob control: k5, k6, k7, k8.
III	11	Granularize recording rec3_2s. Knob control: k5, k6, k7, k8.
	12	Granularize recording rec6_4s. Knob control: k5, k6, k7, k8.

	13	Granularize recording rec1_8s. Knob control: k5, k6, k7, k8.
	14	Granularize recording rec2_8s. Knob control: k5, k6, k7, k8.
I, III	15	Reverb for 12 seconds on input signal through sp. 1, 2, 3, 4, 5, 6, 7, 8.
I	Rec1-2s	Record input signal for 2 seconds.
	Rec2-2s	Record input signal for 2 seconds.
	Rec1-5s	Record input signal for 5 seconds.
	Rec2-5s	Record input signal for 5 seconds.
	Rec3-5s	Record input signal for 5 seconds.
	Rec1_17s	Record input signal for 17 seconds.
II	Rec4-5s	Record input signal for 5 seconds.
	Rec5-5s	Record input signal for 5 seconds.
	Rec1-4s	Record input signal for 4 seconds.
	Rec2-4s	Record input signal for 4 seconds.
	Rec3-4s	Record input signal for 4 seconds.
	Rec4-4s	Record input signal for 4 seconds.
	Rec5-4s	Record input signal for 4 seconds.
	Rec1-22s	Record input signal for 22 seconds.
III	Rec3-2s	Record input signal for 2 seconds.
	Rec6-5s	Record input signal for 5 seconds.
	Rec1-3s	Record input signal for 3 seconds.
	Rec1-8s	Record input signal for 8 seconds.
	Rec2-8s	Record input signal for 8 seconds.
	Rec6-4s	Record input signal for 4 seconds.
I	p-rec1-2s	Playback rec1-2s. Knob control: k5, k7, k8.
	p-rec2-2s	Playback rec2-2s. Knob control: k5, k7, k8.
	p-rec1-5s	Playback rec1-5s. Knob control: k5, k7, k8.
	p-rec2-5s	Playback rec2-5s. Knob control: k5, k7, k8.
	p-rec3-5s	Playback rec3-5s. Knob control: k5, k7, k8.
	p-rec1-17s	Playback rec1-17s. Knob control: k5, k7, k8.
II	p-rec2-4s	Playback rec2-4s. Knob control: k5, k7, k8.
	p-rec3-4s	Playback rec3-4s. Knob control: k5, k7, k8.
	p-rec4-4s	Playback rec4-4s. Knob control: k5, k7, k8.
	p-rec5-4s	Playback rec5-4s. Knob control: k5, k7, k8.
	p-rec1-22sb	Playback rec1-22s backwards. Knob control: k5, k7, k8.
III	p-rec1-3sb	Playback rec1-3s backwards. Knob control: k5, k7, k8.
	p-rec6-5s	Playback rec6-5s. Knob control: k5, k7, k8.
I, II, III	f1	Amplify flute through speaker 1. Knob control: k4, k5.
I, III	f2	Amplify flute through speaker 2. Knob control: k4, k5.
II, III	f3	Amplify flute through speaker 3. Knob control: k4, k5.
I, II, III	f4	Amplify flute through speaker 4. Knob control: k4, k5.
I, III	f5	Amplify flute through speaker 5. Knob control: k4, k5.
I, III	f6	Amplify flute through speaker 6. Knob control: k4, k5.
I, II, III	f7	Amplify flute through speaker 7. Knob control: k4, k5.
II, III	f8	Amplify flute through speaker 8. Knob control: k4, k5.

I	f-o	Amplify flute through speaker 1, 3, 5, 7. Knob control: k4, k5.
I	f-e	Amplify flute through speaker 2, 4, 6, 8. Knob control: k4, k5.
I, II, III	f-1-8	Amplify flute through speaker 1, 2, 3, 4, 5, 6, 7, 8. Knob control: k4, k5.
I, II, III	f1-p	Amplify flute through speaker 1 initially for panning. Knob control: k1, k4, k5.

Source of tape recording

Sound files wind_low.wav and wind_high.wav are made from the original sound file of wind from the Coexistence project (2015) in which recordings are obtained on the Navarino Island of Chile.

Contact

If you have question concerning the code, you may get in touch with Shu-Yu by email at shuyuslin@gmail.com

Hardware setup

Audio interface

1. Connect audio interface to your laptop computer.
2. Choose output to be the audio interface. If are using Mac OS, go to Systems Preferences -> Sound -> Output -> [name of the audio interface].
3. Choose the sample rate of the interface. If you are using a Mac OS, open Audio Devices application. On the left panel choose your interface. Click on Output, and from the Format drop down menu, choose 44,100 Hz as sampling rate. Also choose 44,100 Hz as Input sampling rate by clicking on the Input and select 44,100 Hz.

Speakers

1. Place 2 speakers on the left and the right of the stage and 6 speakers around the performance space evenly.
2. Connect the speakers to your audio interface sequentially. Make sure the speaker number and the interface output correspond. For example speaker 1 connects to interface output 1 etc.
3. Connect power to the 8 speakers.
4. Turn on the speakers.

Microphones

1. Place the dynamic microphone in front of the position of the flutist.
2. Connect microphones to audio interface input 1.

Midi controllers

- Midi controller with at least 8 turn knobs:
 - > If you are using Akai LPD 8:
 1. Label the knobs as follows:
 - K1 knob 'pan' (panning)
 - K2 knob 'mod' (modulation)
 - K3 knob 'reverb dry-wet' (reverberation parameters dry to wet)
 - K4 'reverb room size' (reverberation parameter room size small to large)
 - K5 'gran grain dur' (granular synthesis duration of grain)
 - K6 'gran freq' (granular synthesis frequency of grain)
 - K7 'gran density' (granular synthesis density of grain)
 - K8 'gran pan' (granular synthesis panning of grain)

Each knob label corresponds to the instructions in the electronics part of the score.



2. Connect the device to the computer.

-> If you are using a different device with at least 8 knobs:

1. You may choose to label the 8 knobs on your device the way you wanted as long as it is comprehensible that K1 corresponds to panning function, K2 corresponds to modulation etc. This is because the knob instructions written on the score make use of the labels K1, K2, K3 etc. (See the Electronics control parameters and preparation section on page 6 for reference.)

2. Find and note the midi numbers of each knob. These numbers will be used to modify the midi numbers in the code that was written for LPD 8. The next section (Supercollider code instruction) specifies which parameters at which lines in the code need to be modified.

4. Connect the device to the computer.

- Push button controller:

-> If you are using Akai APC mini:

1. Label the buttons as shown in the image.



Buttons f1 to f8 and f-o, f-e, f1-8 triggers corresponding amplifications.
 Buttons 1 to 15 triggers specific electronics processing.
 Buttons associate with recording begins with 'rec'.
 Buttons associate with playback of recording begins with 'p'.
 Circular buttons directly above the sliders with labels 1 to 8 trigger playback of noise or frequency sweep to test specific speaker where 1 refers to speaker 1, 2 refers to speaker 2.

The explanatory notes for score and the electronics numbering in the performance instruction section describes the number symbols on the score corresponding to the button numbers, and the electronic process of each number.

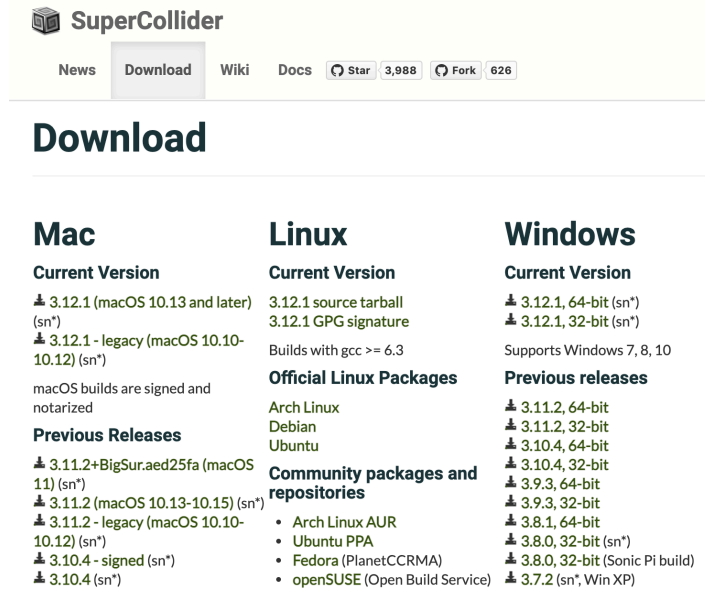
2. Connect the device to your computer.

-> If you are using a different push button midi controller with at least 68 buttons and 8 sliders:

1. Label the buttons with the same label names in the APC image. This is because the button instructions on the score are written using these label names. (See the Electronics control parameters and preparation section for reference.)
2. Find and note the midi numbers of each button. These numbers will be used to modify the midi numbers in the code that was written for APC mini. The next section (Supercollider code instruction) specifies which parameters at which lines in the code need to be modified.
3. Connect the device to your computer.

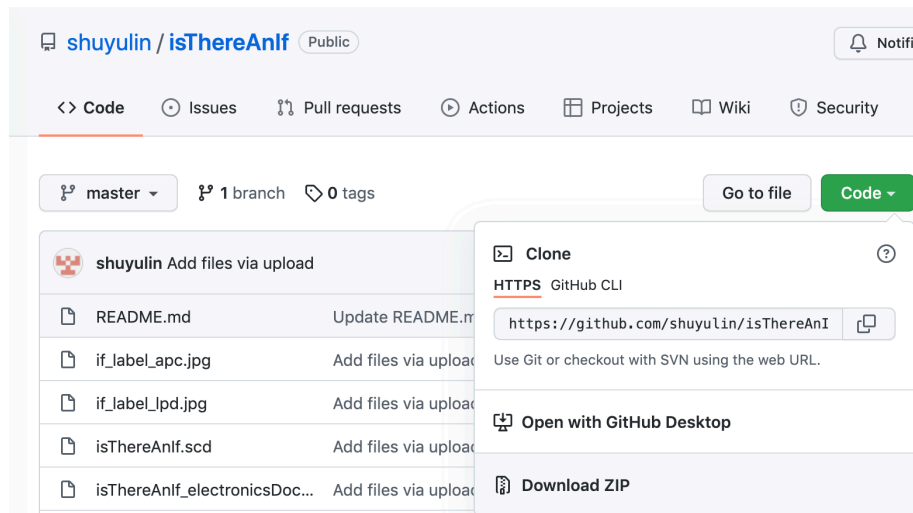
SuperCollider code instruction

1. Download SuperCollider at <https://supercollider.github.io/download> Make sure the version that you download is compatible with your computer's operating system. Click open the downloaded zip file and install SuperCollider.



The screenshot shows the GitHub repository page for SuperCollider. At the top, there is a navigation bar with 'Download' highlighted. Below this, the 'Download' section is titled. It is divided into three columns: Mac, Linux, and Windows. Each column lists the current version and provides instructions for downloading or installing. The Mac section lists versions 3.12.1 and 3.12.1-legacy. The Linux section lists the current version 3.12.1 and provides instructions for building and installing on various distributions. The Windows section lists the current version 3.12.1 and provides instructions for installing on Windows 7, 8, and 10.

2. Download files for electronics from github at <https://github.com/shuyulin/isThereAnIf> Click on 'Code' and select 'Download ZIP'.

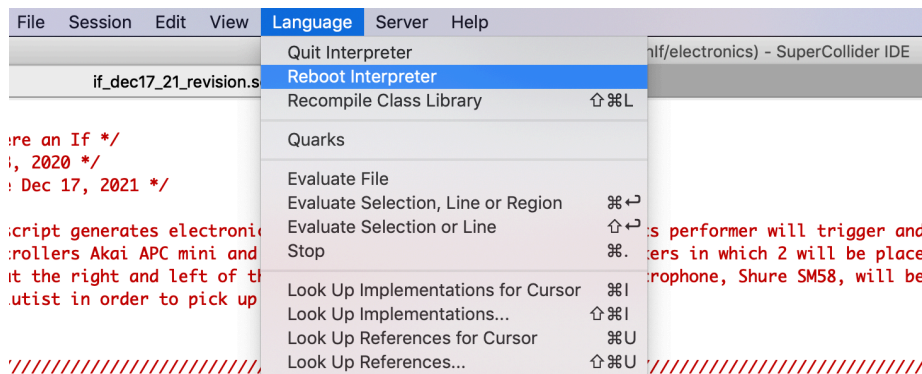


The screenshot shows the GitHub repository page for 'shuyulin/isThereAnIf'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted, and a dropdown menu is open, showing options to clone the repository using HTTPS or GitHub CLI, open it with GitHub Desktop, or download it as a ZIP file. The repository contains several files, including README.md, if_label_apc.jpg, if_label_lpd.jpg, isThereAnIf.scd, and isThereAnIf_electronicsDoc...

3. Open zip and move to a directory where you wanted it to be.

4. Click open the SuperCollider script 'isThereAnIf.scd'.

Note: The Interpreter should indicate Active after you opened the .scd file. Reboot the Interpreter if it is inactive. To do this, go to Language drop down menu and select Reboot Interpreter.



5. Modify the numbers of channel output if necessary.

-> If you are using 8-speaker setup (original plan for Is There an If?)

The number of output and input channels from lines 25-28 do not need to be changed. If you would want to have exact number of channel correspond to the input and output, modify the number accordingly.

1. Change number of output bus channels to 8 at lines 25 and 27.

```

22 {
23 {
24   /* start up file */
25   Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26   Server.local.options.numInputBusChannels = 1;
27   Server.internal.options.numOutputBusChannels = 8;
28   Server.internal.options.numInputBusChannels = 1;

```

2. Change number of input bus channels to 1 at lines 26 and 28.

```

22 {
23 {
24   /* start up file */
25   Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26   Server.local.options.numInputBusChannels = 1;
27   Server.internal.options.numOutputBusChannels = 8;
28   Server.internal.options.numInputBusChannels = 1;

```

-> If you are using stereo setup with additional speaker at the piano

If 8-speaker setup is not possible at the time to rehearsal, however, 2 speakers are available (stereo), modify the number of input and output channels and their corresponding global variables.

1. Change number of output bus channels to 2 at lines 25 and 27.

```

22 {
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26 Server.local.options.numInputBusChannels = 1;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 1;

```

2. Change number of input bus channels to 1 at lines 26 and 28.

```

22 {
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26 Server.local.options.numInputBusChannels = 1;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 1;

```

3. Modify numbers assigned to the global variables. Change ~ch2, ~ch3 and ~ch4 to 1 for stereo right output. Change ~ch5, ~ch6 and ~ch7 to 0 for stereo left output. Change ~ch_o to [0, 0, 0, 0] for output to stereo left output (speaker 1). Change ~ch_e to [1, 1, 1, 1] for stereo right output (speaker 2). Change ~ch_1_8 to [0, 0, 0, 0, 1, 1, 1, 1] for stereo L R output.

```

120 ///////////////////////////////////////////////////* function */////////////////////////////////////
121 /* global variables for Motu setup */
122 ~ch0 = 0; // channel outputs
123 ~ch1 = 1;
124 ~ch2 = 2;
125 ~ch3 = 3;
126 ~ch4 = 4;
127 ~ch5 = 5;
128 ~ch6 = 6;
129 ~ch7 = 7;
130 ~ch8 = 8; // extra for expansion
131 ~ch_o = [0, 2, 4, 6];
132 ~ch_e = [1, 3, 5, 7];
133 ~ch_1_8 = [0, 1, 2, 3, 4, 5, 6, 7];
134 ~mic1 = 0; // mic input- flute
135 ~mic2 = 1; // extra for expansion
136

```

-> If you are using built-in input and output

If you do not have microphone and external speakers at the time of running the code, you may choose to use the computer's built-in input and outputs.

1. Since no signal will come into input channels 1 and 2, you may indicate the input to be 1 for the single build-in microphone input at lines 26 and 28.

```

22 {
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26 Server.local.options.numInputBusChannels = 1;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 1;

```

2. Change number of output bus channels to 2 at lines 25 and 27.


```

22 {
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26 Server.local.options.numInputBusChannels = 1;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 1;

```

3. Modify numbers assigned to the global variables. Change ~ch2, ~ch3 and ~ch4 to 1 for stereo right output. Change ~ch5, ~ch6 and ~ch7 to 0 for stereo left output. Change ~ch_o to [0, 0, 0, 0] for output to stereo left output (speaker 1). Change ~ch_e to [1, 1, 1, 1] for stereo right output (speaker 2). Change ~ch_1_8 to [0, 0, 0, 0, 1, 1, 1, 1] for stereo L R output.

```

120 ///////////////////////////////////////////////////* function */////////////////////////////////////
121 /* global variables for Motu setup */
122 ~ch0 = 0; // channel outputs
123 ~ch1 = 1;
124 ~ch2 = 2;
125 ~ch3 = 3;
126 ~ch4 = 4;
127 ~ch5 = 5;
128 ~ch6 = 6;
129 ~ch7 = 7;
130 ~ch8 = 8; // extra for expansion
131 ~ch_o = [0, 2, 4, 6];
132 ~ch_e = [1, 3, 5, 7];
133 ~ch_1_8 = [0, 1, 2, 3, 4, 5, 6, 7];
134 ~mic1 = 0; // mic input- flute
135 ~mic2 = 1; // extra for expansion
136

```

6. Indicate audio device.

-> If you are using an external audio interface

1. Find the name of your audio interface. If you are using a Mac OS, open Audio Devices application (Audio MIDI Setup), the names of audio input and output devices are displayed at the left panel.

2. Go to line 29 and modify the name of the audio interface to the one you are using. You may use different input and output audio interface. If so, indicate your input device at line 30 (.inDevice) and indicate your output device at line 31 (.outDevice).

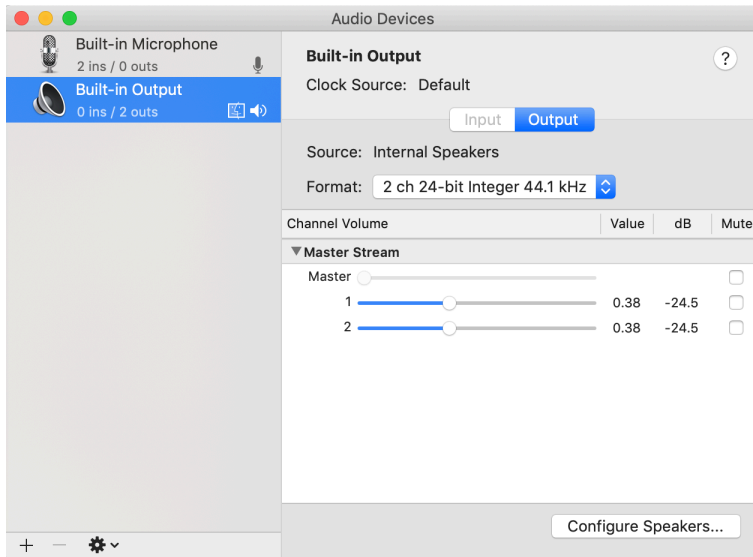
```

29 Server.local.options.device = "MOTU 896mk3 Hybrid";
30 //Server.local.options.inDevice = "Scarlett Zi4 USB"; // other possible audio interfaces
31 //Server.local.options.outDevice = "Orpheus (0979)";
32 //Server.local.options.outDevice = "Built-in Output";
33

```

-> If you are using computer's built-in audio

1. Find the name of the name of the build-in microphone and the name of build-in output. If you are using a Mac OS, open Audio Device application (Audio MIDI Setup), the names of the build-in input and output devices are displayed at the left panel. The name of the built-in input is 'Built-in Microphone' and built-in output is 'Built-in Output'.



2. Go to line 30 and 31 and modify the names of the in and out devices to the names of the built-in input and built-in output.

7. Setup midi controllers

-> If you are using the same midi controllers as the ones used during the composition of *Is There an If?*

You do not need to modify the midi setups. But make sure the labels on your controllers correspond to the images of the controllers in the previous instruction, and make sure the devices are properly connected to your computer.

-> If you are using different midi devices.

1. Modify the names of the midi controllers in the code from lines 747 to 754.

- To change the name of the push button controller with at least 68 buttons and 8 sliders, go to lines 748 and 750 and replace APC MINI to the name of your device.

- To change the name of the midi controller with at least 8 turn knobs, go to lines 752 and 754 and replace LPD8 to the name of your device.

```

746      /* get midi controller IDs */
747      MIDIClient.sources.do{|item|
748          if(item.name == "APC MINI", {~apcID = item.uid}); }; // pad controller
749      MIDIClient.destinations.do{|item|
750          if(item.name == "APC MINI", {~apcOutID = item.uid}); };
751      MIDIClient.sources.do{|item|
752          if(item.name == "LPD8", {~lpdID = item.uid}); }; // keyboard controller
753      MIDIClient.destinations.do{|item|
754          if(item.name == "LPD8", {~lpdOutID = item.uid}); };
755

```

* Do not change the name of the global variables eg. ~apcID because these are the names of the containers. Each contains the name of your device and is used in other parts of the code. Assigning the names of your controllers to these global variables (however the

name could be) should be sufficient to link your device to the code. If you want the names of the global variables to correspond to your devices, make sure you change all the names of the corresponding global variables in the code.

2. Change the midi numbers of each device.

- For a push button controller with at least 68 buttons and 8 sliders (in place of Akai APC mini), go to line 762 where commented `/* APC slider controls */`, modify the midi number associate with each slider in each `MIDIdef.cc` from lines 762-857. The midi number is in purple before `srcID`.

```

762      /* APC slider controls */
763      /* slider 1 */
764      MIDIdef.cc(\slider1_button01, {
765          arg num, id, chan, src;
766          //[num, id, chan, src].postln;
767          Ndef(\sweep1).set(\amp, num.linlin(0, 127, 0, 0.1));
768          Ndef(\f1).set(\vol, num.linlin(0, 127, 0, 1));
769          Ndef(\f_o).set(\vol, num.linlin(0, 127, 0, 1));
770          Ndef(\p_rec1_2s).set(\amp, num.linlin(0, 127, 0, 1));
771          Ndef(\p_rec2_4s).set(\amp, num.linlin(0, 127, 0, 1));
772          Ndef(\granularize_rec1_4s).set(\vol, num.linlin(0, 127, 0, 1));
773          Ndef(\p_rec_wind_low_1357).set(\vol, num.linlin(0, 127, 0, 1));
774      }, 48, srcID: ~apcID);
775
776      /* slider 2 */
777      MIDIdef.cc(\slider2_button02, {
778          arg num, id, chan, src;
779          //[num, id, chan, src].postln;
780          Ndef(\sweep2).set(\amp, num.linlin(0, 127, 0, 0.1));
781          Ndef(\f2).set(\vol, num.linlin(0, 127, 0, 1));
782          Ndef(\f_e).set(\vol, num.linlin(0, 127, 0, 1));
783          Ndef(\p_rec2_2s).set(\amp, num.linlin(0, 127, 0, 1));
784          Ndef(\p_rec3_4s).set(\amp, num.linlin(0, 127, 0, 1));
785          Ndef(\granularize_rec1_22s).set(\vol, num.linlin(0, 127, 0, 1));
786          Ndef(\p_rec_wind_low_2468).set(\vol, num.linlin(0, 127, 0, 1));
787      }, 49, srcID: ~apcID);

```

Go to line 1033 where commented `/* trigger sound; APC */`, modify the midi number associate with each button in each `MIDIdef.noteOn` from lines 1119 to 3406 and lines 3412 to 3552. The midi number is in purple before the `srcID`.

```

1118      //////////////// button f1 ////////////////////////// bar X
1119      MIDIdef.noteOn(\apcButton56, {
1120          arg vel, nn, chan, src;
1121          /* amplify flute through speaker 1 */
1122          if (~countApcf1 == 0, {
1123              ~f1 = Ndef(\f1, ~amplifyFluteThru1);
1124              m.noteOn(0, 56, 01); //light on
1125              "on 56 button f1".postln;
1126              ~countApcf1 = 1;
1127          }, {
1128              ~f1.release(1);
1129              m.noteOn(0, 56, 0); //light off
1130              "off 56 button f1".postln;
1131              ~countApcf1 = 0;
1132          });
1133      }, 56, srcID: ~apcID);

```

```

3411 ////////////////////////////////////////////////// sweep/noise sp 1 ////////////////////////////////////////// button 01
3412 MIDIdef.noteOn(\apcButton64, {
3413   arg vel, nn, chan, src;
3414   if (~countApc01 == 0, {
3415     ~sweep1 = Ndef(\sweep1, ~sweep_sp1);
3416     //Ndef(\sweep1).set(\bufnum, ~bufTestSweep);
3417     Ndef(\sweep1).set(\bufnum, ~bufTestNoise);
3418     m.noteOn(0, 64, 01); //light on
3419     "on 64 button 01 sweep/noise speaker 1".postln;
3420     ~countApc01 = 1;
3421   }, {
3422     ~sweep1.release(1);
3423     m.noteOn(0, 64, 0); //light off
3424     "off 64 button 01".postln;
3425     ~countApc01 = 0;
3426   });
3427 }, 64, srcID: ~apcID);

```

- For a midi controller with at least 8 turn knobs (in place of Akai LPD 8), go to /* LPD control mapping */ at line 860. Modify the midi number of each knob in each MIDIdef.cc to the corresponding midi number of the knob of your choice. The midi number is in purple before the srcID.

```

860 /* LPD control mapping */
861 /* LPD knob 1 control panning of incoming signal */
862 MIDIdef.cc(\lpdKnob1_panflute_f1_p, {
863   arg num, nn, chan, src;
864   //[num, nn, chan, src].postln;
865   [num, nn].postln;
866   Ndef(\f1_p).set(\pos, num.linlin(0, 127, 0, 2));
867   Ndef(\p_rec1_2s).set(\pos, num.linlin(0, 127, 0, 2));
868   Ndef(\p_rec2_2s).set(\pos, num.linlin(0, 127, 0, 2));
869   Ndef(\p_rec1_3s_b).set(\pos, num.linlin(0, 127, 0, 2));
870   Ndef(\p_rec1_5s).set(\pos, num.linlin(0, 127, 0, 2));
871   Ndef(\p_rec2_5s).set(\pos, num.linlin(0, 127, 0, 2));
872   Ndef(\p_rec3_5s).set(\pos, num.linlin(0, 127, 0, 2));
873   Ndef(\p_rec1_17s).set(\pos, num.linlin(0, 127, 0, 2));
874   Ndef(\p_rec2_4s).set(\pos, num.linlin(0, 127, 0, 2));
875   Ndef(\p_rec3_4s).set(\pos, num.linlin(0, 127, 0, 2));
876   Ndef(\p_rec4_4s).set(\pos, num.linlin(0, 127, 0, 2));
877   Ndef(\p_rec5_4s).set(\pos, num.linlin(0, 127, 0, 2));
878   Ndef(\p_rec1_22s_b).set(\pos, num.linlin(0, 127, 0, 2));
879   Ndef(\p_rec6_5s).set(\pos, num.linlin(0, 127, 0, 2));
880   Ndef(\p_rec_wind_low_1357).set(\panPos, num.linlin(0, 127, 0, 2));
881   Ndef(\p_rec_wind_low_2468).set(\panPos, num.linlin(0, 127, 0, 2));
882   Ndef(\p_rec_wind_high_1357).set(\panPos, num.linlin(0, 127, 0, 2));
883   Ndef(\p_rec_wind_high_2468).set(\panPos, num.linlin(0, 127, 0, 2));
884   }, 1, srcID: ~lpdID);
885

```

-> If you do not want to connect midi controllers.

When testing certain parts of the code, for example, testing the input signals from microphones, you may want to run the script without connecting midi devices. To do this, comment out the part of code that associates with midi by putting a /* at line 745 and */ at line 3564. Once you put both the /* and */ at the appropriate lines, the part you commented out should appear in red.

```

744 "func and def done".postln;
745 /*
746 /* get midi controller IDs */
747 MIDIClient.sources.do{|item|
748   if(item.name == "APC MINI", {~apcID = item.uid}); }; // pad controller
749 MIDIClient.destinations.do{|item|
750   if(item.name == "APC MINI", {~apcOutID = item.uid}); };
751 MIDIClient.sources.do{|item|
752   if(item.name == "LPD8", {~lpdID = item.uid}); }; // keyboard controller
753 MIDIClient.destinations.do{|item|
754   if(item.name == "LPD8", {~lpdOutID = item.uid}); };

3561 ~midiMapApc.C);
3562
3563 "midiMap done".postln;
3564 */
3565 "done!".postln;
3566 };
3567 }.fork

```

If you comment out the midi part of the code, you may not be able to use any midi device to control sound processes or run any part of the sound processes, unless you modify the code in the way that suits your habit of sound testing and performance without using midi but other devices.

* If you do not comment out the midi part of the code when you do not have midi devices connected to your computer, and proceed to run the script, an error will show in the post window with the message, 'ERROR: Message 'uid' not understood.'

```

func and def done
ERROR: Message 'uid' not understood.
RECEIVER:
  nil

```

-> If you want to connect only one controller

When testing certain parts of the code that associate with a particular midi controllers, you may comment out the parts that do not associate with the particular devices.

1. Comment out the associated MIDIClient.sources, MIDIClient.destinations and MIDIOut from lines 747 to 758. In the example, the midi controller been tested is APC mini and the part of the code that do not associate with APC is commented out.

```

745
746      /* get midi controller IDs */
747      MIDIClient.sources.do{|item|
748          if(item.name == "APC MINI", {~apcID = item.uid}); }; // pad controller
749      MIDIClient.destinations.do{|item|
750          if(item.name == "APC MINI", {~apcOutID = item.uid}); };
751      /*MIDIClient.sources.do{|item|
752          if(item.name == "LPD8", {~lpdID = item.uid}); }; // keyboard controller
753      MIDIClient.destinations.do{|item|
754          if(item.name == "LPD8", {~lpdOutID = item.uid}); }; */
755
756      /* set midi outputs */
757      m = MIDIOut(0, ~apcOutID);
758      /*l = MIDIOut(0, ~lpdOutID);*/
759      "MIDIOut done".postln;
760

```

2. Comment out the lines that do not associate with your midi controller from lines 762 to 3563. In the example, the controller being tested is APC, the lines that do not associate with APC are commented out.

```

856      Ndef(\granularize_rec5_5s).set(\vol, num.linlin(0, 127, 0, 1));
857      }, 55, srcID: ~apcID);
858
859      /*
860      /* LPD control mapping */
861      /* LPD knob 1 control panning of incoming signal */
862      MIDIDef.cc(\lpdKnob1_panflute_f1_p, {
863          arg num, nn, chan, src;
864          //[num, nn, chan, src].postln;
865          [num, nn].postln;
866          Ndef(\f1_p).set(\pos, num.linlin(0, 127, 0, 2));
867
868
869
870
871
872
873      Ndef(\granularize_rec6_4s).set(\shiftPan, num.linlin(0, 127, 0, 2));
874      Ndef(\granularize_rec1_8s).set(\shiftPan, num.linlin(0, 127, 0, 2));
875      Ndef(\granularize_rec2_8s).set(\shiftPan, num.linlin(0, 127, 0, 2));
876      Ndef(\granularize_in0).set(\shiftPan, num.linlin(0, 127, 0, 2));
877      }, 8, srcID: ~lpdID);
878      */
879

```

8. Run the code.

After you have setup hardware and modified the code according to your hardware setup, proceed to run the code.

Move cursor to the beginning of line 22. This will highlight lines 22 to 3569. Run the lines by pressing command + return. This will boot the Server and store the written variables and functions for sound processes.

```

21
22
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change number of input and output channels
26 Server.local.options.numInputBusChannels = 2;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 2;
29 Server.local.options.device = "MOTU 896mk3 Hybrid";
30 //Server.local.options.inDevice = "Scarlett 2i4 USB"; // other possible audio interfaces
31 //Server.local.options.outDevice = "Orpheus (0979)";
32 //Server.local.options.outDevice = "Built-in Output";
33
34 {s.meter;}defer; // make sure number of channels is correct
35 MIDIClient.init; // start midi
36 MIDIIn.connectAll; // connect all midi devices
37
38 s.waitForBoot { // go through before booting server
39
40 "buffers".postln;

```

If SuperCollider is successfully booted, the parameters of Server at the lower right of the environment will show in green. If buffers are successfully allocated, functions and midiMaps are successfully stored, the post window will show the messages 'buffers done', 'func and def done' and 'midiMap done'.

Shared memory server interface initialized
buffers

```

buffers done
func and def done
midiMap done
done!

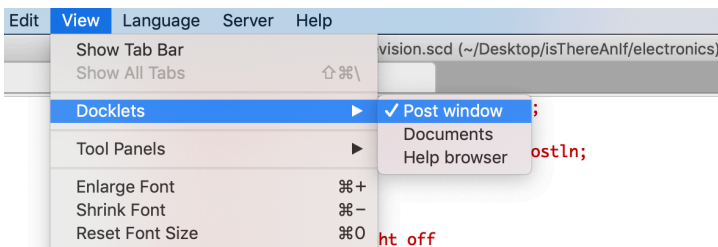
```

```

Interpreter: Active Server: 0.14% 0.19% 14u 3s 3g 111d 0.0dB

```

If you do not see the post window, go to View, select Docklets, and select Post window.

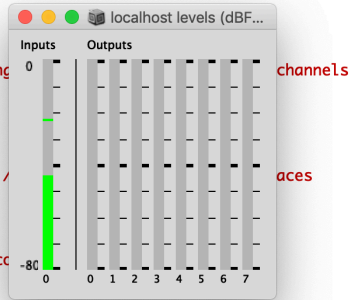


A meter should pop up unless the s.meter (line 34) wasn't run. If there is no meter, go to line 34 to see if the line was commented out accidentally, or trouble shoot lines 22 to 34 to see what the problem is. The number of the input and output on the meter corresponds to the numbers of the input and output channels that you indicate at lines 25-28.

```

21
22 {
23 {
24 /* start up file */
25 Server.local.options.numOutputBusChannels = 8; // change
26 Server.local.options.numInputBusChannels = 1;
27 Server.internal.options.numOutputBusChannels = 8;
28 Server.internal.options.numInputBusChannels = 1;
29 //Server.local.options.device = "MOTU 896mk3 Hybrid";
30 //Server.local.options.inDevice = "Scarlett 2i4 USB"; //
31 //Server.local.options.outDevice = "Orpheus (0979)";
32 Server.local.options.outDevice = "Built-in Output";
33
34 {s.meter;}.defer; // make sure number of channels is correct
35 MIDIClient.init; // start midi
36 MIDIIn.connectAll; // connect all midi devices

```



9. Test speakers, rehearse and perform the electronics part of the *Is There an If?*.

After you have successfully ran lines 22 to 3569, and the post window indicates the messages 'buffers done', 'func and def done', 'midiMap done' and 'done!', proceed to the speaker test. The instruction for speaker test is outlined in the next section. Practice the electronics part by following the written instructions on the score.

10. To close the script, click on the x at upper left of the tab (or command + w). A pop up window will appear asking if you would want to save the changes, click on 'Save' so you do not need to modify the code the next time you open the script. Click on the x at the upper left of the program window to close SuperCollider program (or command + q).

Tips

1. If you wish to stop a process before its end, press command + . (period).
2. If you are unable to run certain processes for unknown reason, recompile class library, go to Language and click on Recompile Class Library or press shift + command + l. After doing this, you will need to run lines 22 to 3569 again to store variables and functions.

Speaker test

1. Speaker test by noise and frequency sweep.

If you are using the 8-speaker setup, and using APC mini controller, you can playback the noise sound file by pressing the circular buttons 1-8 on APC where button 1 corresponds to speaker 1, button 2 corresponds to speaker 2 etc. Pressing buttons 1 will playback noise sound file 'test-noise.wav' through speaker 1. Pressing button 2 will playback noise sound file 'test-noise.wav' through speaker 2.

You may test the speakers by frequency sweep. To do this, modify the global variables that store the noise sound files to the global variable that store the sweep sound files from lines 3412 to 3552. For example, to change sound file for speaker 1 test, uncomment line 3416 and comment out line 3417. Do this also for speakers 2-8. Alternately, you may simply modify the names of the global variables from ~bufTestNoise to ~bufTestSweep at line 3417. Do this also for speakers 2-8.

```
3411 ////////////////////////////////////////////////// sweep/noise sp 1 ////////////////////////////////////////// button 01
3412 MIDIdef.noteOn(\apcButton64, {
3413   arg vel, nn, chan, src;
3414   if (~countApc01 == 0, {
3415     ~sweep1 = Ndef(\sweep1, ~sweep_sp1);
3416     //Ndef(\sweep1).set(\bufnum, ~bufTestSweep);
3417     Ndef(\sweep1).set(\bufnum, ~bufTestNoise);
3418     m.noteOn(0, 64, 01); //light on
3419     "on 64 button 01 sweep/noise speaker 1".postln;
3420     ~countApc01 = 1;
3421   }, {
3422     ~sweep1.release(1);
3423     m.noteOn(0, 64, 0); //light off
3424     "off 64 button 01".postln;
3425     ~countApc01 = 0;
3426   });
3427 }, 64, srcID: ~apcID);
3428
```

You may substitute different sound files for speaker tests by changing the buffer directories from lines 109 to 111.

```
108 s.sync;
109 ~bufTestNoise = Buffer.read(s, "/Users/shuyulin/Desktop/isThereAnIf/test-noise.wav"); // noise for testing speakers
110 s.sync;
111 ~bufTestSweep = Buffer.read(s, "/Users/shuyulin/Desktop/isThereAnIf/test-sweep.wav");
112 s.sync;
```

Depending on the position of the speaker and the type (or brand) of speaker (if you are using various brands/models), noise and sweep may sound slightly different from each other.

2. Adjust volume accordingly.