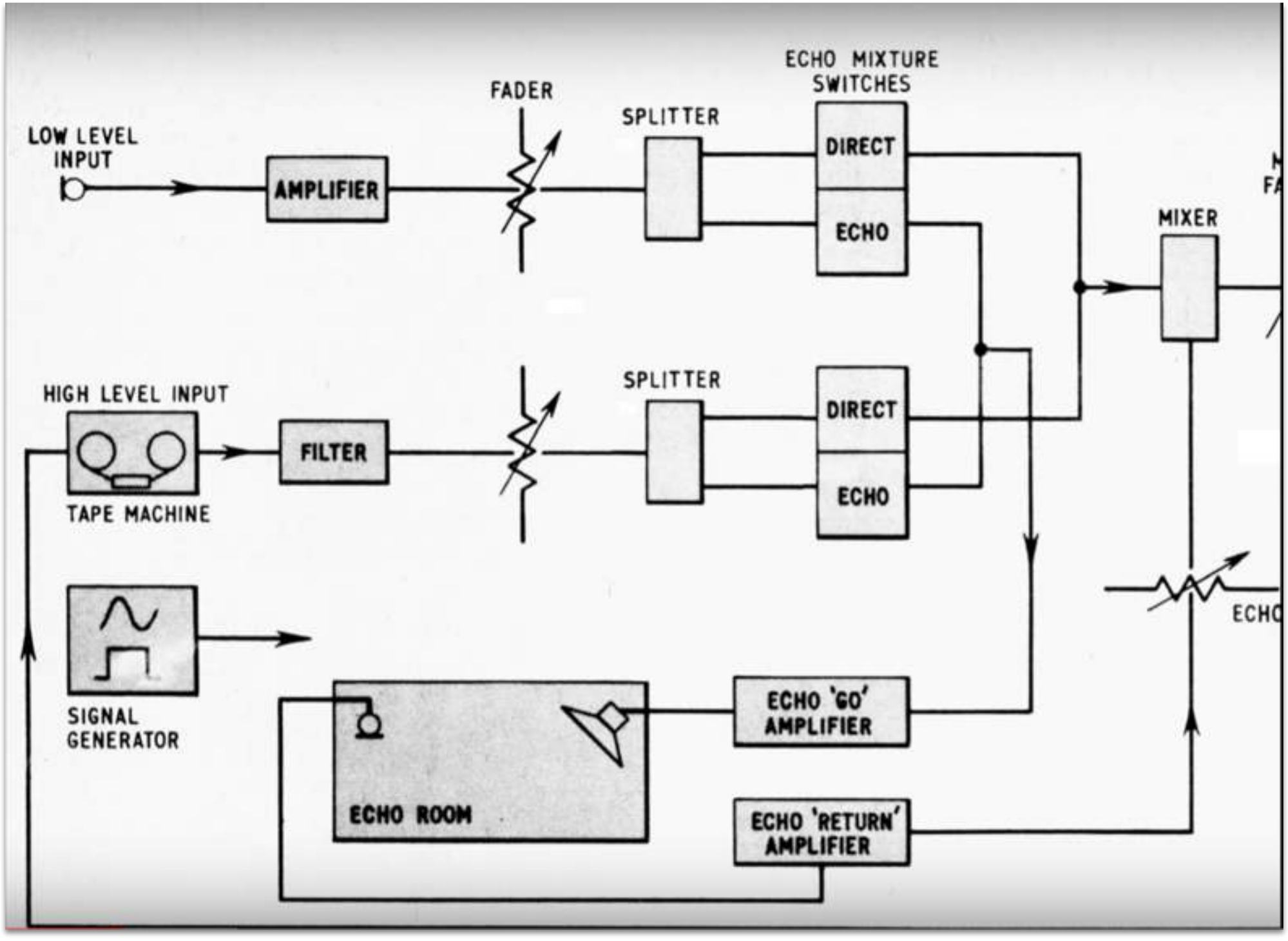
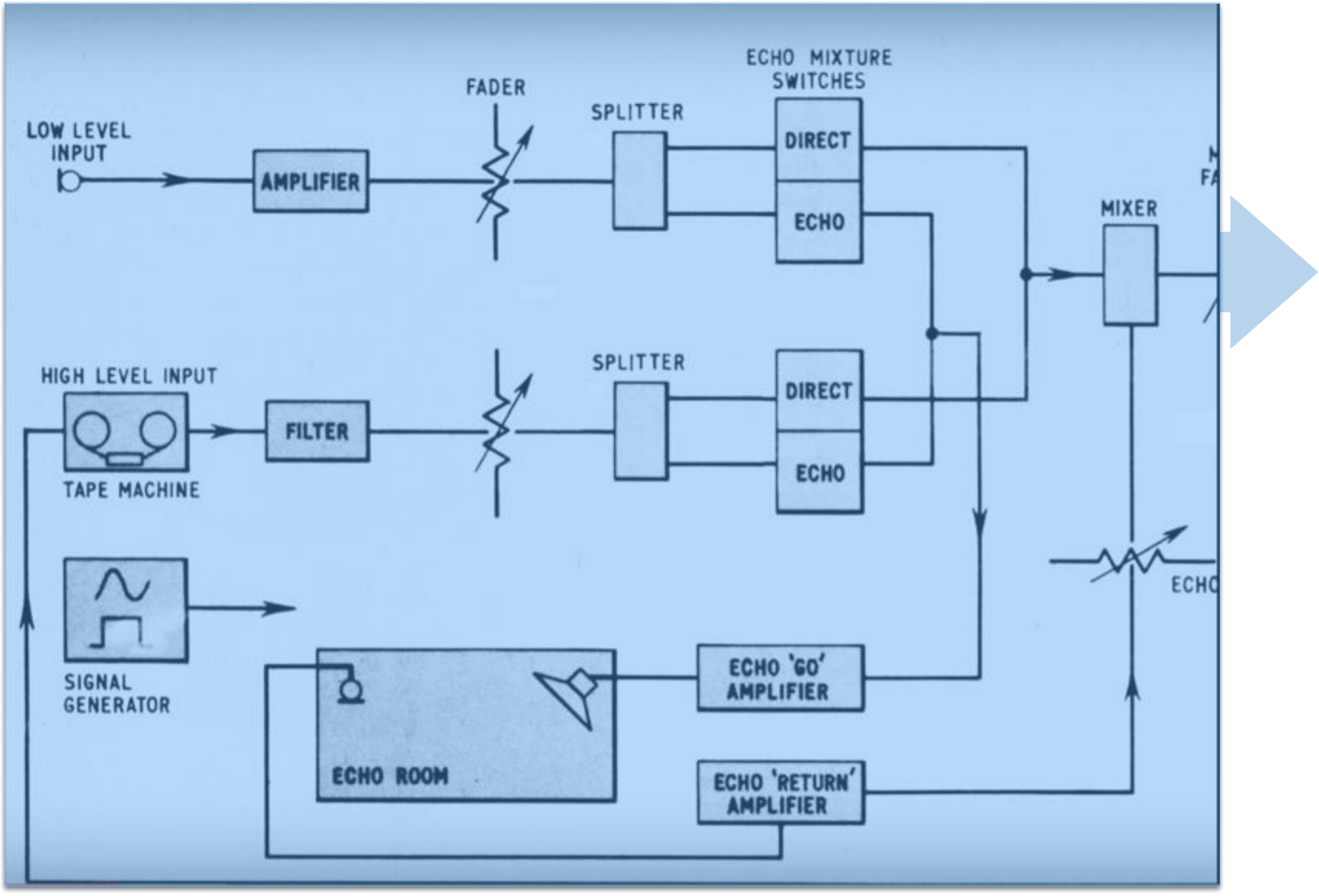
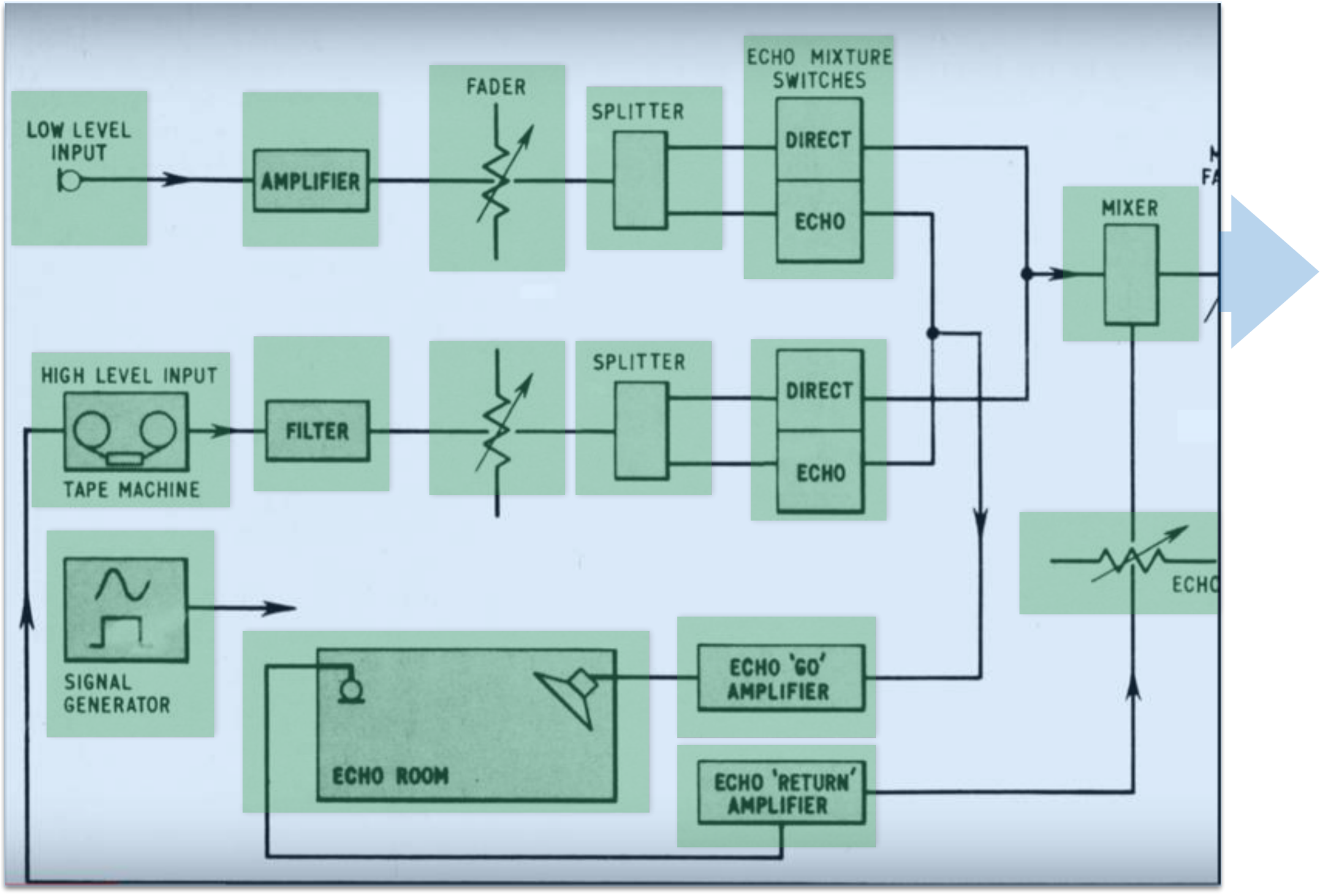
alemangui


LOW LEVEL INPUT
AMPLIFIER
FADER
SPLITTER
ECHO MIXTURE SWITCHES
DIRECT
ECHO
MIXER
HIGH LEVEL INPUT
FILTER
TAPE MACHINE
SPLITTER
DIRECT
ECHO
SIGNAL GENERATOR
ECHO ROOM
ECHO 'GO' AMPLIFIER
ECHO 'RETURN' AMPLIFIER
ECHO

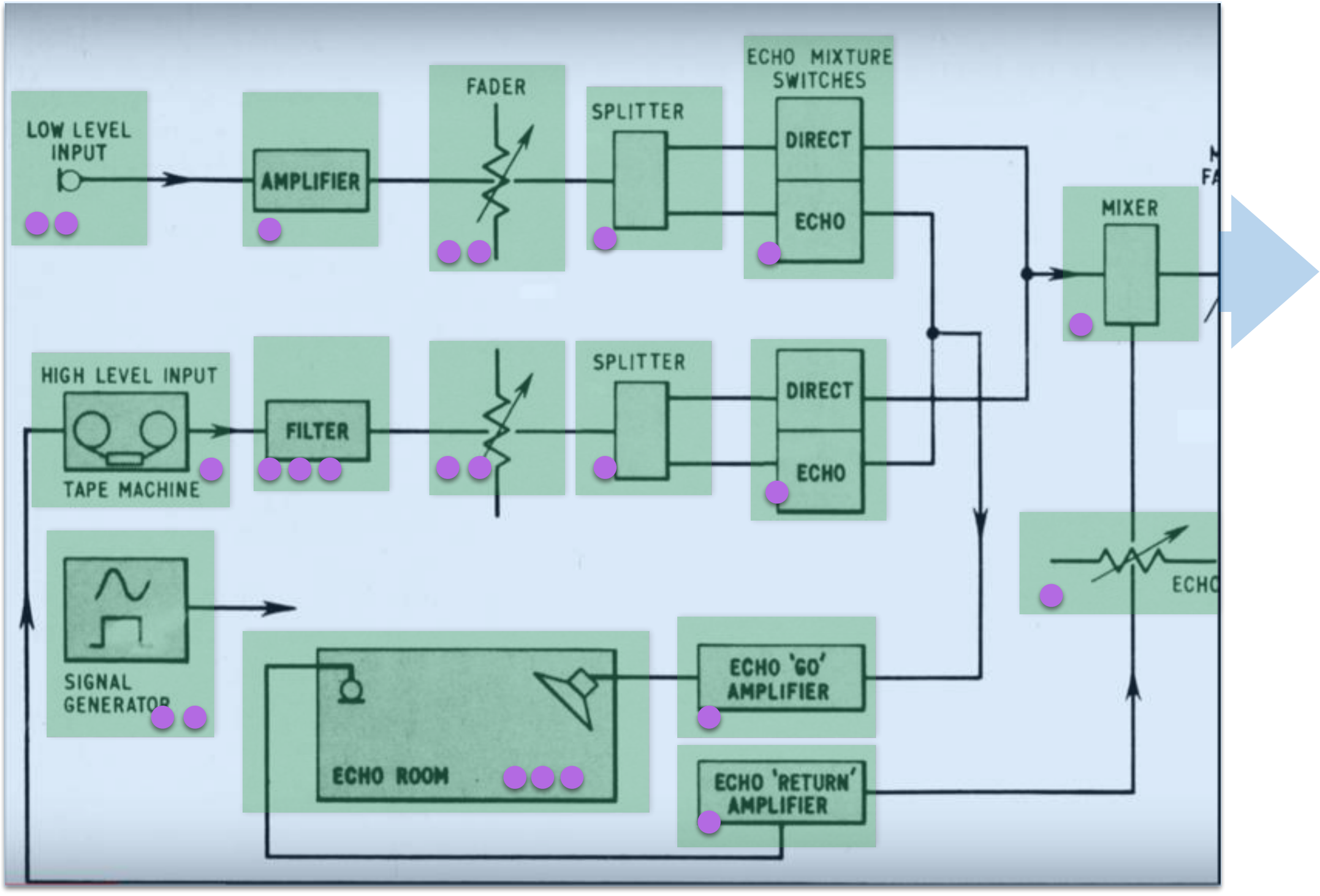
Audio context



Audio context Nodes



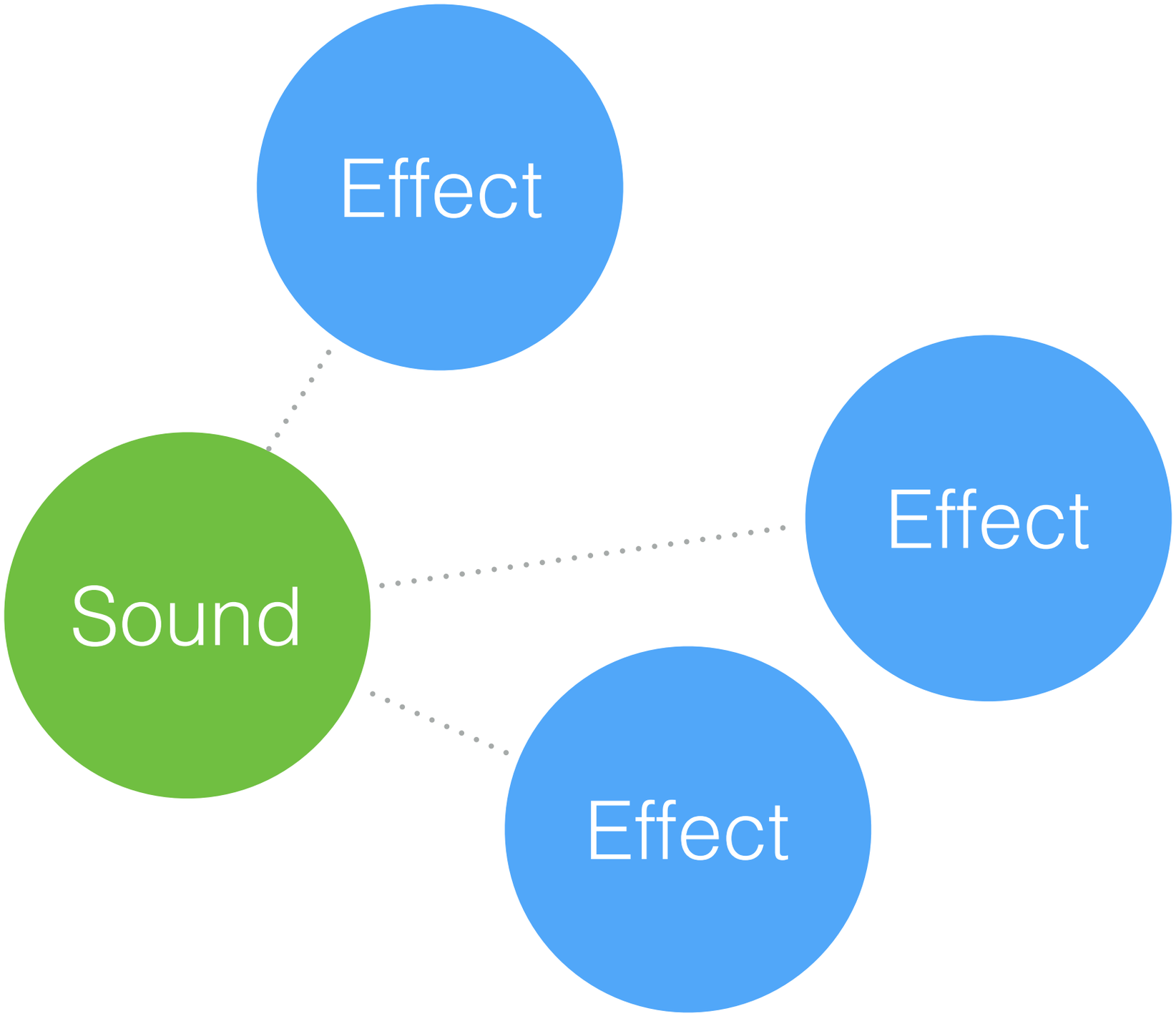
Audio context Nodes Parameters



Hacking Time

10

CHECK
MODE
LEVEL
FUZZ
BOOST
OUTPUT
INPUT
Fuzz
FZ-5
CHECK
TONE
LEVEL
DRIVE
OUTPUT
INPUT
SUPER
Over Drive
SD-1
CHECK
DECAY
MANUAL
SENS
MODE
OUTPUT
GUITAR IN
EXP/CTL
BASS IN
Dynamic Wah
AW-3
CHECK
REVERB LEVEL
BODY
TOP
MODE
LINE OUT
G.AMP OUT
INPUT
Acoustic
Simulator
AC-3

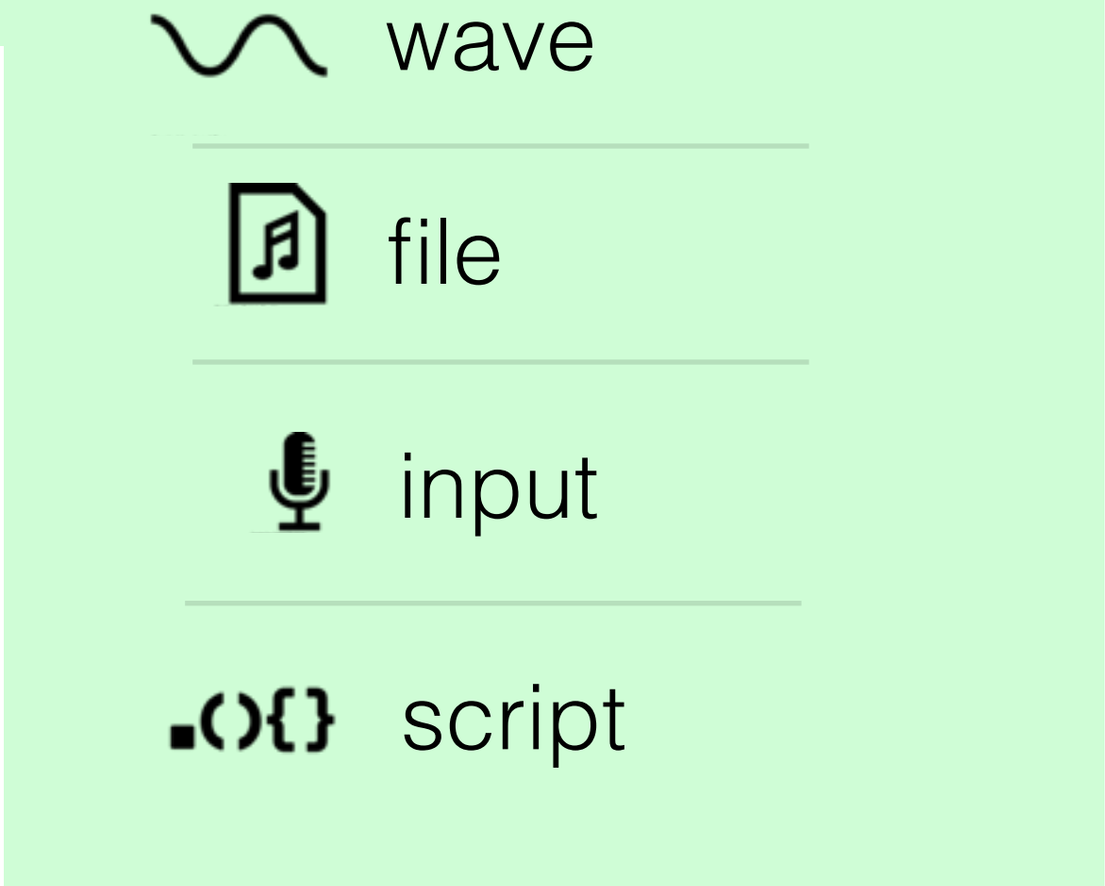

Effect
Sound
Effect
Effect

PIZZICATO.JS

Sound

```
var sound = new Pizzicato.Sound({
    source: "wave",
    options: {
        type: "sine",
        frequency: 220
    }
});
```

```
var sound = new Pizzicato.Sound({
    source: "wave",
    options: {
        type: "sine",
        frequency: 220
    }
});
```



```
var sound = new Pizzicato.Sound({
    source: "wave",
    options: {
        type: "sine",
        frequency: 220
    }
});
```

```
var sound = new Pizzicato.Sound({
    source: "wave",
    options: {
        type: "sine",
        frequency: 220
    }
});
```

sound.**play**()

sound.**stop**()

sound.**pause**()

Au Code

Get sound from file

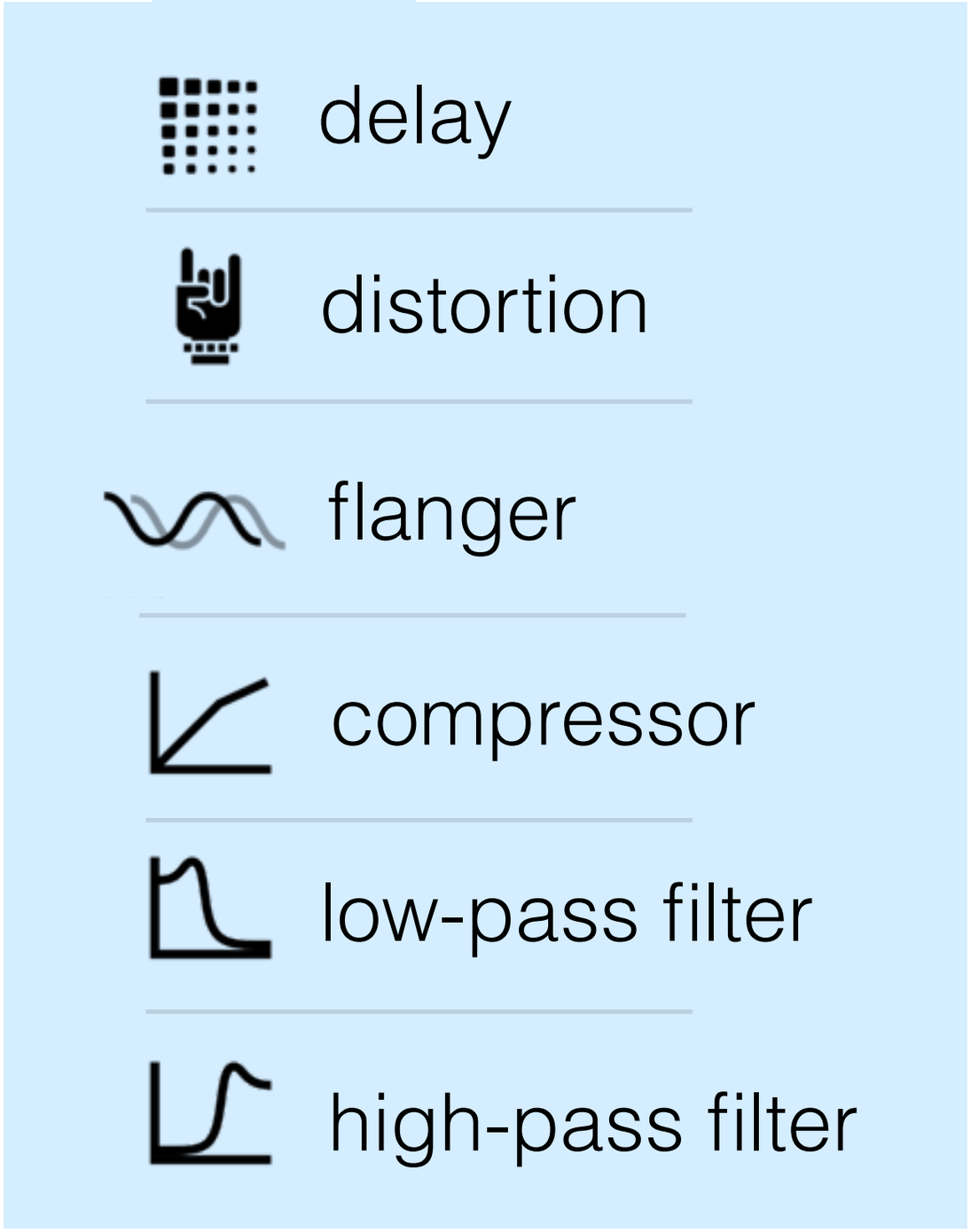
```
var sound = new Pz.Sound({
    source: 'file',
    options: {
        path: './file.mp4'
    }
}, function() {
   console.log('loaded!');
});
```

```
var Ctx = window.AudioContext ||
          window.webkitAudioContext;
var ctx = new Ctx();

var req = new XMLHttpRequest();
req.open('GET', './file.mp4', true);
req.responseType = 'arraybuffer';

req.onload = function(e) {
  ctx.decodeAudioData(e.target.response,(function(b) {
    var node = ctx.createBufferSource();
    node.buffer = b;
    node.connect(ctx.destination);
    console.log('loaded!');
  });
};
```

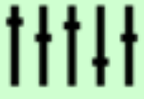
Effects

```
var effect = new Pz.Effects.Delay({
    feedback: 0.8,
    time: 0.22,
    mix: 0.75
});

sound.addEffect(effect);
```

```
var effect = new Pz.Effects.Delay({
    feedback: 0.8,
    time: 0.22,
    mix: 0.75
});

sound.addEffect(effect);
```



```
var effect = new Pz.Effects.Delay({
    feedback: 0.8,
    time: 0.22,
    mix: 0.75
});
sound.addEffect(effect);
```



effect-specific

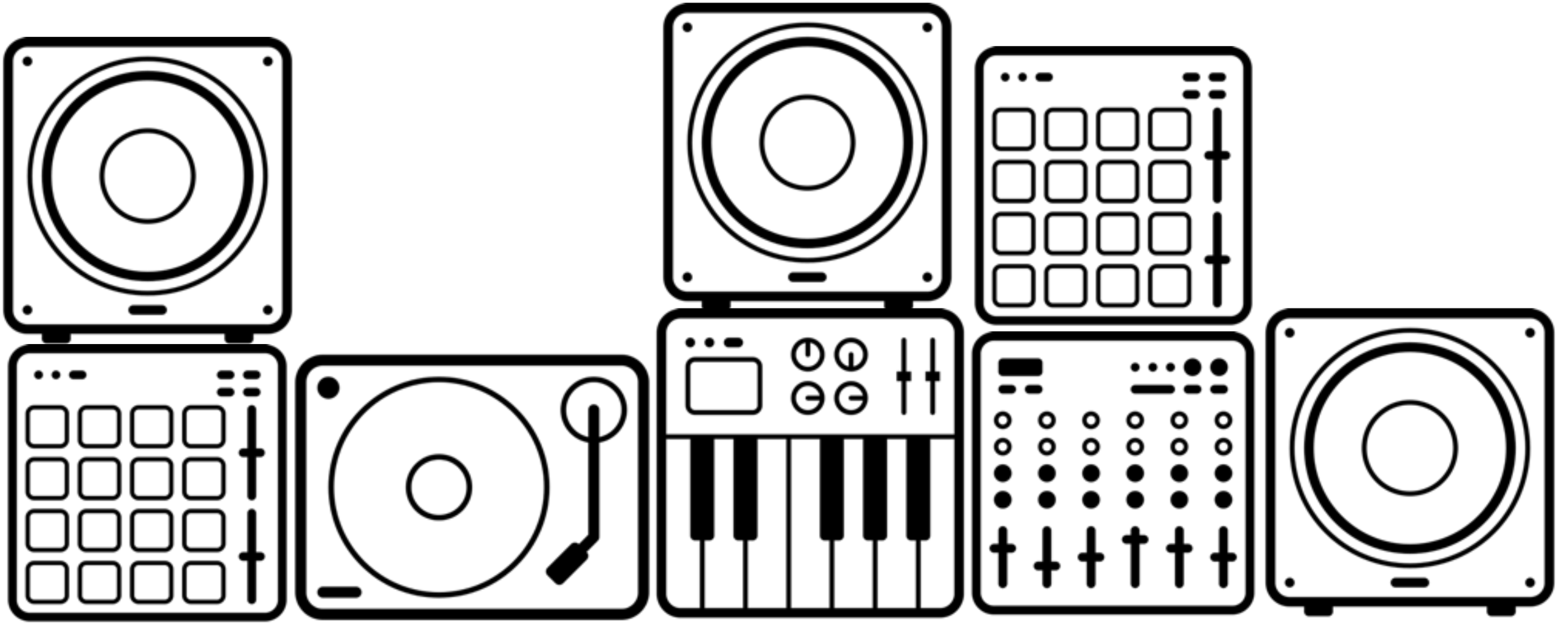
Aux Demos

PIZZICATO.JS

alemangui.github.io/pizzicato/

github.com/alemangui/**pizzicato**

github.com/alemangui/**web-audio-resources**



alemangui