
Low Pass Filter

Creating a low pass filter effect for Glove with Flex Sensors

```
s.boot;
(
/*
SerialPort.closeAll;
SerialPort.devices;
s.quit;
*/

//initial setup
~port=SerialPort("/dev/tty.usbmodem1411", 9600);
~val = [];
~min = 620;
~max = 830;
d = Dictionary.new;
d.add(\a -> 0);
d.add(\b -> 0);
d.add(\c -> 0);

Tdef(\getdata, {
  loop{
    var ascii;

    ascii=~port.next;

    case

    {ascii==nil}
    {nil}

    {ascii.asAscii.isAlpha}
    {
      if(
        ~val.size>0,
        {
          d[ascii.asAscii.asSymbol] = ~val.convertDigits;
          ~val = [];
        }
      );
    }

    {ascii.asAscii.isDecDigit}
    {~val=~val.add(ascii.asAscii.digit)}

    {true}{nil};

    //d.postln;
    0.0001.wait;
  };
});
)
```

```

(
s.waitForBoot({
  ~spin= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Spin.wav" );
  ~spin2= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Spin2.wav" );
  ~spin3= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Spin3.wav" );
  ~spin4= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Spin4.wav" );
  ~spin5= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Spin5.wav" );
  ~bow= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Bow.wav" );
  ~bowtwo= Buffer.read(s, "/Users/Casey/Music/MusicGlove/BowTwo.wav" );
  ~bowthree= Buffer.read(s, "/Users/Casey/Music/MusicGlove/BowThree.wav" );
  ~cave= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Cave.wav" );
  ~robo= Buffer.read(s, "/Users/Casey/Music/MusicGlove/Robo.wav" );

  ~spinM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Spin.wav", channels:[0]);
  ~spin2M1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Spin2.wav", channels:[0] );
  ~spin3M1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Spin3.wav", channels:[0] );
  ~spin4M1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Spin4.wav", channels:[0] );
  ~spin5M1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Spin5.wav", channels:[0] );
  ~bowM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Bow.wav", channels:[0] );
  ~bowtwoM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/BowTwo.wav", channels:[0] );
  ~bowthreeM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/BowThree.wav", channels:[0] );
  ~caveM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Cave.wav", channels:[0] );
  ~roboM1= Buffer.readChannel(s, "/Users/Casey/Music/MusicGlove/Robo.wav", channels:[0] );

SynthDef.new(\playsound, {
  arg out=0, buf=0, density=1, cf=20000, rq=1,
  atk=5, rel=5, gate=1;
  var sig, env;
  sig = GrainBuf.ar(
    2,
    Dust.kr(density),
    1,
    buf,
    1,
    LFNnoise0.kr(500).range(0,1),
    2,
    0,
  );
  sig = RLPF.ar(sig, cf, rq);
  env = EnvGen.kr(Env.new([0,1,1,0], [atk,0.01,rel],[1,0,-1],2), gate, doneAction:2);
  sig = sig * env;
  Out.ar(out, sig);
}).add;
});
)

//EVENT FUNCTIONS
(
~events = [];
~events = ~events.add({
  ~roboGrain1 = Synth.new(\playsound, [\out, 0, \buf, ~roboM1.bufnum, \density, 8]);

Tdef(\roboGrain1FX, {
  loop{
    ~roboGrain1.set(\cf, d[\a].linexp(690,830,500,10000));
  }
}
)

```

```

109         0.05.wait;
110     }
111     }).play
112
113 });
114
115 ~events = ~events.add({
116     Tdef(\roboGrain1FX).stop;
117     ~roboGrain1.set(\gate, 0);
118     ~caveGrain1 = Synth.new(\playsound, [\out, 0, \buf, ~caveM1.bufnum, \density, 20]);
119 });
120 )
121
122 Tdef(\getdata).play;
123
124 ~events[0].value;
125 ~events[1].value;
126
127 (
128 w = Window.new.front;
129 ~button = Button.new(w, Rect(50, 50, 300, 300))
130 .states_([[\LowPass, Color.white, Color.black], [\1, Color.black, Color.white], [\2, Color.white,
131 .action_({
132     arg btn;
133     ~events[btn.value-1].value;
134 });
135 )
136
137
138 //test lines
139 ~newsynth = Synth.new(\playsound, [\out, 0, \buf, ~roboM1.bufnum, \density, 8])
140
141 ~newsynth.set(\density, 0.5);
142 ~newsynth.set(\density, 8);
143
144 ~newsynth.set(\cf, 7000, \rq, 1);

```