

CRÉATION EN RÉSEAU - UJM2017

INTERLOPER OU LE "PUBLICIEN" IMPROVISATEUR

Thomas Ciperre
CIEREC-EA3068



RÉSEAU INTERLOPER

1. OSC
2. Faust
3. Osc-Apps
4. httpd

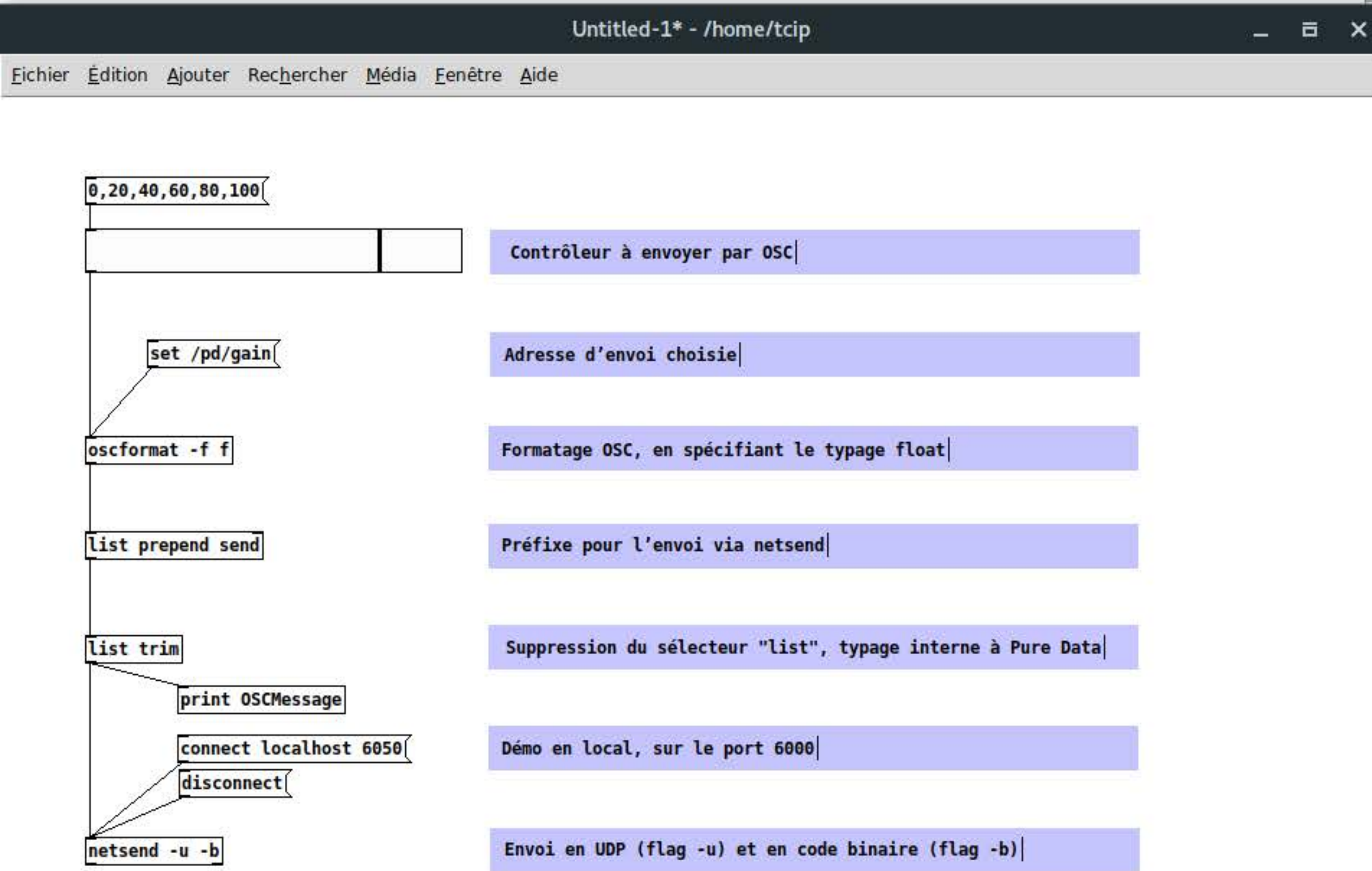
OPEN SOUND CONTROL

OSC

1. CNMAT (Adrian Freed, Matt Wright)
2. format de contrôle temps-réel en réseau (TCP et UDP)
3. Atomic data types
4. Address patterns "/"

OSC DEMO

```
Pd
Fichier Édition Ajouter Rechercher Média Fenêtre Aide
Log: 2
Audio ON
DSP
connecting to port 6050
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 0 0 0
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 65 160 0 0
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 66 32 0 0
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 66 112 0 0
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 66 160 0 0
OSCMessage: send 47 112 100 47 103 97 105 110 0 0 0 0 44 102 0 0 66 200 0 0
```



```
tcip@tcip:~
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ oscdump
oscdump version 0.28
Copyright (C) 2008 Kentaro Fukuchi

Usage: oscdump <port>
or      oscdump <url>
Receive OpenSound Control messages and dump to standard output.

Description
port      : specifies the listening port number.
url       : specifies the server parameters using a liblo URL.
            e.g. UDP      "osc.udp://:9000"
            Multicast  "osc.udp://224.0.1.9:9000"
            TCP         "osc.tcp://:9000"
```

```
[tcip@tcip ~]$ oscdump 6050
/pd/gain f 0.000000
/pd/gain f 20.000000
/pd/gain f 40.000000
/pd/gain f 60.000000
/pd/gain f 80.000000
/pd/gain f 100.000000
```



```

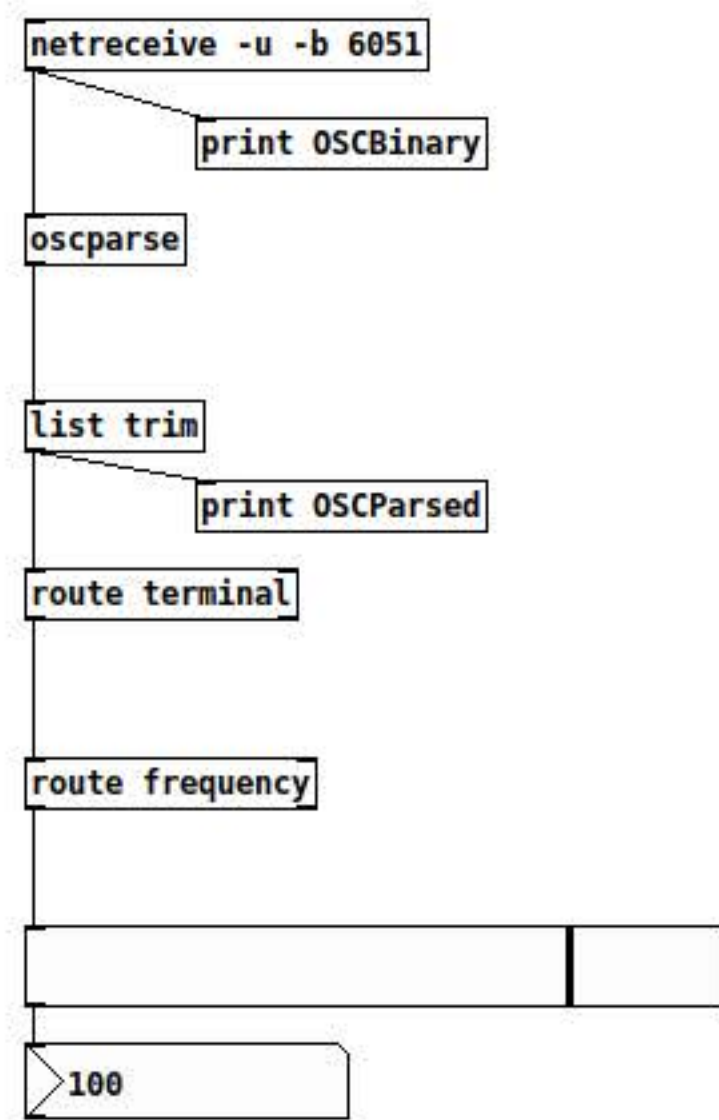
Pd
Fichier Édition Ajouter Rechercher Média Fenêtre Aide
Log: 2
Audio ON
OSCparsed: terminal frequency 0
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 0 0 0
OSCparsed: terminal frequency 20
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 65 160 0 0
OSCparsed: terminal frequency 40
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 66 32 0 0
OSCparsed: terminal frequency 60
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 66 112 0 0
OSCparsed: terminal frequency 80
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 66 160 0 0
OSCparsed: terminal frequency 100
OSCBinary: 47 116 101 114 109 105 110 97 108 47 102 114 101 113 117 101 110 99 121 0 44 102 0 0 66 200 0 0

```

```

Untitled-1* - /home/tcip
Fichier Édition Ajouter Rechercher Média Fenêtre Aide

```



Réception en UDP sur le port 6051

Décodage de l'OSC binaire

Suppression du sélecteur "list"

Routing du premier sélecteur choisi : "terminal"

Routing du deuxième sélecteur choisi : "frequency"

Contrôleur récupérant la valeur numérique routée

```

tcip@tcip:~
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ oscsend
oscsend version 0.28
Copyright (C) 2008 Kentaro Fukuchi

Usage: oscsend hostname port address types values...
or      oscsend url address types values...
Send OpenSound Control message via UDP.

Description
hostname: specifies the remote host's name.
port     : specifies the port number to connect to the remote host.
url      : specifies the destination parameters using a liblo URL.
           e.g. UDP      "osc.udp://localhost:9000"
           Multicast    "osc.udp://224.0.1.9:9000"
           TCP          "osc.tcp://localhost:9000"

address  : the OSC address where the message to be sent.
types    : specifies the types of the following values.
           i - 32bit integer
           h - 64bit integer
           f - 32bit floating point number
           d - 64bit (double) floating point number
           s - string
           S - symbol
           c - char
           m - 4 byte midi packet (8 digits hexadecimal)
           T - TRUE (no value required)
           F - FALSE (no value required)
           N - NIL (no value required)
           I - INFINITUM (no value required)
values   : space separated values.

Example
$ oscsend localhost 7777 /sample/address iTfs 1 3.14 hello
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 0
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 20
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 40
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 60
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 80
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 100
[tcip@tcip ~]$

```

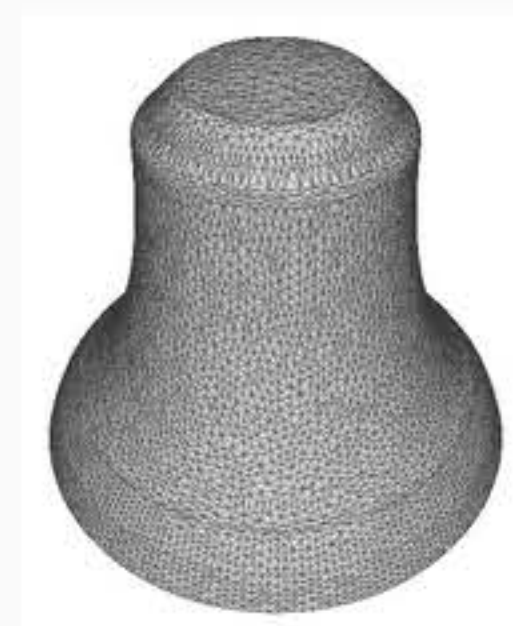
```

[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 0
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 20
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 40
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 60
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 80
[tcip@tcip ~]$ oscsend localhost 6051 /terminal/frequency f 100
[tcip@tcip ~]$

```


FAUST – GRAME(LYON)





FUNCTIONAL AUDIO STREAM

1. FaustLive, FaustPlayground, Faust2Android, etc.
2. Documentation automatique et compilation distante
3. Modèles Physiques et CAD, réseaux de neurones, etc.

FAUST-OSC DEMO


```
oscTest.dsp — ~/Documents/Conferences/e-formes/dev — Atom
File Edit View Selection Find Packages Help
Project oscTest.dsp
1 import("stdfaust.lib");
2 // GUI
3 gainGUI=hslider("gain",0,0,1,0.001):si.smoo;
4 freqGUI=hslider("freq[unit:Hz]",440,30,4000,0.1);
5 // PROCESS
6 process = os.oscsin(freqGUI):(gainGUI);
7
```



```
tcip@tcip:~/Documents/Conferences/e-formes/dev/faust
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ cd Documents/Conferences/e-formes/dev/faust/
[tcip@tcip faust]$ faust2jack -osc oscTest.dsp
oscTest;
[tcip@tcip faust]$ ./oscTest -xmit 1
OSC is on
The sample rate is now 48000/sec
physical input system:capture_1
physical input system:capture_2
physical input system:capture_3
physical input system:capture_4
physical input system:capture_5
physical input system:capture_6
physical input system:capture_7
physical input system:capture_8
physical input system:capture_9
physical input system:capture_10
physical input system:capture_11
physical input system:capture_12
physical output system:playback_1
physical output system:playback_2
physical output system:playback_3
physical output system:playback_4
physical output system:playback_5
physical output system:playback_6
physical output system:playback_7
physical output system:playback_8
physical output system:playback_9
physical output system:playback_10
physical output system:playback_11
physical output system:playback_12
The buffer size is now 1024/sec
Faust OSC version 0.96 application 'oscTest' is running on UDP ports 5510, 5511, 5512
```

```
tcip@tcip:~
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ oscdump 5511
/oscTest/freq f 880.000000
/oscTest/gain f 0.500000
/oscTest/freq f 440.000000
/oscTest/gain f 0.000000
```

```
tcip@tcip:~
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ oscsend localhost 5510 /oscTest/freq f 880
[tcip@tcip ~]$ oscsend localhost 5510 /oscTest/gain f 0.5
[tcip@tcip ~]$ oscsend localhost 5510 /oscTest/freq f 440
[tcip@tcip ~]$ oscsend localhost 5510 /oscTest/gain f 0
[tcip@tcip ~]$
```

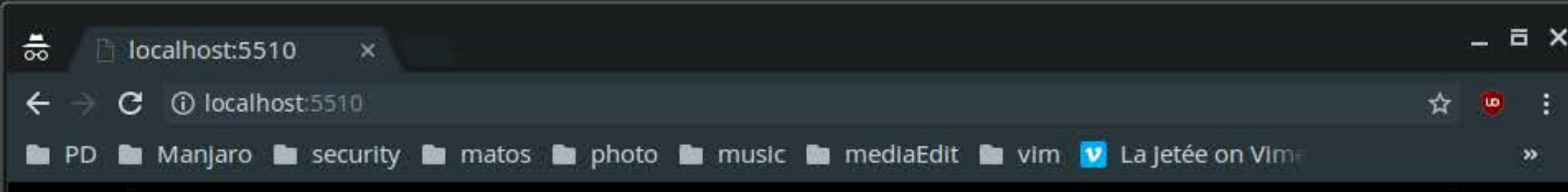



LIBMICROHTTPD



FAUST-HTTPD DEMO


```
oscTest.dsp — ~/Documents/Conferences/e-formes/dev — Atom
File Edit View Selection Find Packages Help
Project oscTest.dsp
dev
  faust
  pd
  router
  interloper.sh
1 import("stdfaust.lib");
2 // GUI
3 gainGUI=hslider("gain",0,0,1,0.001):si.smoo;
4 freqGUI=hslider("freq[unit:Hz]",440,30,4000,0.1);
5 // PROCESS
6 process = os.oscsin(freqGUI):(gainGUI);
7
```



oscTest

freq

833.5Hz

gain

0.624

```
tcip@tcip:~/Documents/Conferences/e-formes/dev/faust
Fichier Édition Affichage Rechercher Terminal Aide
[tcip@tcip ~]$ cd Documents/Conferences/e-formes/dev/faust/
[tcip@tcip faust]$ faust2jack -httpd oscTest.dsp
oscTest;
[tcip@tcip faust]$ ./oscTest
HTTPD is on
The sample rate is now 48000/sec
physical input system:capture_1
physical input system:capture_2
physical input system:capture_3
physical input system:capture_4
physical input system:capture_5
physical input system:capture_6
physical input system:capture_7
physical input system:capture_8
physical input system:capture_9
physical input system:capture_10
physical input system:capture_11
physical input system:capture_12
physical output system:playback_1
physical output system:playback_2
physical output system:playback_3
physical output system:playback_4
physical output system:playback_5
physical output system:playback_6
physical output system:playback_7
physical output system:playback_8
physical output system:playback_9
physical output system:playback_10
physical output system:playback_11
physical output system:playback_12
The buffer size is now 1024/sec
Faust httpd server version 0.73 is running on TCP port 5510

```

oscTest

oscTest

freq

833,5

gain

0,624

INTERLOPER DEMO

Project

makefile

```

dev
├── faust
│   ├── bassGUI
│   ├── bassGUI.dsp
│   ├── delayGUI
│   ├── delayGUI.dsp
│   ├── drumGUI
│   ├── drumGUI.dsp
│   ├── faustDelay.dsp
│   ├── faustSpat.dsp
│   ├── lSynthGUI
│   ├── lSynthGUI.dsp
│   ├── makefile
│   ├── masterGUI
│   ├── masterGUI.dsp
│   ├── ringModGUI
│   ├── ringModGUI.dsp
│   ├── spatGUI
│   ├── spatGUI.dsp
│   ├── sSynthGUI
│   └── sSynthGUI.dsp
├── pd
├── router
└── interloper.sh

```

```

1 all: drumGUI bassGUI lSynthGUI sSynthGUI ringModGUI delayGUI spatGUI masterGUI
2
3
4 drumGUI : drumGUI.dsp
5     faust2jack -httpd -osc drumGUI.dsp
6
7 bassGUI : bassGUI.dsp
8     faust2jack -httpd -osc bassGUI.dsp
9
10 lSynthGUI : lSynthGUI.dsp
11     faust2jack -httpd -osc lSynthGUI.dsp
12
13 sSynthGUI : sSynthGUI.dsp
14     faust2jack -httpd -osc sSynthGUI.dsp
15
16 ringModGUI : ringModGUI.dsp
17     faust2jack -httpd -osc ringModGUI.dsp
18
19 delayGUI : delayGUI.dsp
20     faust2jack -httpd -osc delayGUI.dsp
21
22 spatGUI : spatGUI.dsp
23     faust2jack -httpd -osc spatGUI.dsp
24
25 masterGUI : masterGUI.dsp
26     faust2jack -httpd -osc masterGUI.dsp
27
28 clean:
29     rm drumGUI bassGUI lSynthGUI sSynthGUI ringModGUI delayGUI spatGUI masterGUI
30

```

interloper.sh

```

1 #!/bin/bash
2 #starts all Fausts with correct ports
3 ./faust/drumGUI -port 6001 -xmit 1 -outport 6009 -errport 6010 &
4 ./faust/bassGUI -port 6002 -xmit 1 -outport 6009 -errport 6010 &
5 ./faust/lSynthGUI -port 6003 -xmit 1 -outport 6009 -errport 6010 &
6 ./faust/sSynthGUI -port 6004 -xmit 1 -outport 6009 -errport 6010 &
7 ./faust/ringModGUI -port 6005 -xmit 1 -outport 6009 -errport 6010 &
8 ./faust/delayGUI -port 6006 -xmit 1 -outport 6009 -errport 6010 &
9 ./faust/spatGUI -port 6007 -xmit 1 -outport 6009 -errport 6010 &
10 ./faust/masterGUI -port 6008 -xmit 1 -outport 6009 -errport 6010 &
11 pd -jack -open ./pd/interloper.pd

```

drumGUI.dsp

```

1 initGUI=attach(0,button("v:drum/[0]init"));
2 sampleGUI=attach(0,nentry("v:drum/h:[1]sample/[0]sample",0,0,1,1));
3 playGUI=attach(0,checkbox("v:drum/h:[1]sample/[1]play"));
4 gainGUI=attach(0,hslider("v:drum/[2]gain",0,0,1,0.001));
5 resetSpeedGUI=attach(0,button("v:drum/[3]resetSpeed"));
6 speedGUI=attach(0,hslider("v:drum/[4]speed",0,-2,2,0.01));
7 process = initGUI,sampleGUI,playGUI,gainGUI,resetSpeedGUI,speedGUI:>_;
8

```

bassGUI.dsp

```

1 initGUI=attach(0,button("v:bass/[0]init"));
2 sampleGUI=attach(0,nentry("v:bass/h:[1]sample/[0]sample",0,0,1,1));
3 playGUI=attach(0,checkbox("v:bass/h:[1]sample/[1]play"));
4 gainGUI=attach(0,hslider("v:bass/[2]gain",0,0,1,0.001));
5 resetSpeedGUI=attach(0,button("v:bass/[3]resetSpeed"));
6 speedGUI=attach(0,hslider("v:bass/[4]speed",0,-2,2,0.01));
7 process = initGUI,sampleGUI,playGUI,gainGUI,resetSpeedGUI,speedGUI:>_;
8

```

delayGUI.dsp

```

1 initGUI=attach(0,button("v:delay/[0]init"));
2 timeGUI=attach(0,hslider("v:delay/[1]time",250,1,2000,1));
3 feedbackGUI=attach(0,hslider("v:delay/[2]feedback",0,0,100,0.1));
4 interpolationGUI=attach(0,hslider("v:delay/[3]interpolation",300,1,2000,1));
5 sampleGUI=attach(0,nentry("v:delay/[4]sample",0,0,3,1));
6 process = initGUI,timeGUI,feedbackGUI,interpolationGUI,sampleGUI:>_;
7

```


drumGUI

drum

init

sample

sample 0 play

gain 0,000

resetSpeed

speed 0,00

bassGUI

bass

init

sample

sample 0 play

gain 0,000

resetSpeed

speed 0,00

lSynthGUI

lSynth

init

sample

sample 0 play

gain 0,000

resetSpeed

speed 0,00

sSynthGUI

sSynth

init

sample

sample 0 play

gain 0,000

resetSpeed

speed 0,00

ringModGUI

ringMod

init

freq 120

amount 0,001

sample 0

delayGUI

delay

init

time 250

feedback 0,0

interpolation 300

sample 1

spatGUI

spat

init

depth 0,0

pan 50,0

sample 0

master

DSP

gain 1,000

interloper.pd - /home/tcip/Documents/Conferences/e-formes/dev/.pd

Fichier Édition Ajouter Rechercher Média Fenêtre Aide

The screenshot shows a Pure Data patch window titled 'interloper.pd'. The patch is organized into several sub-patch boxes:

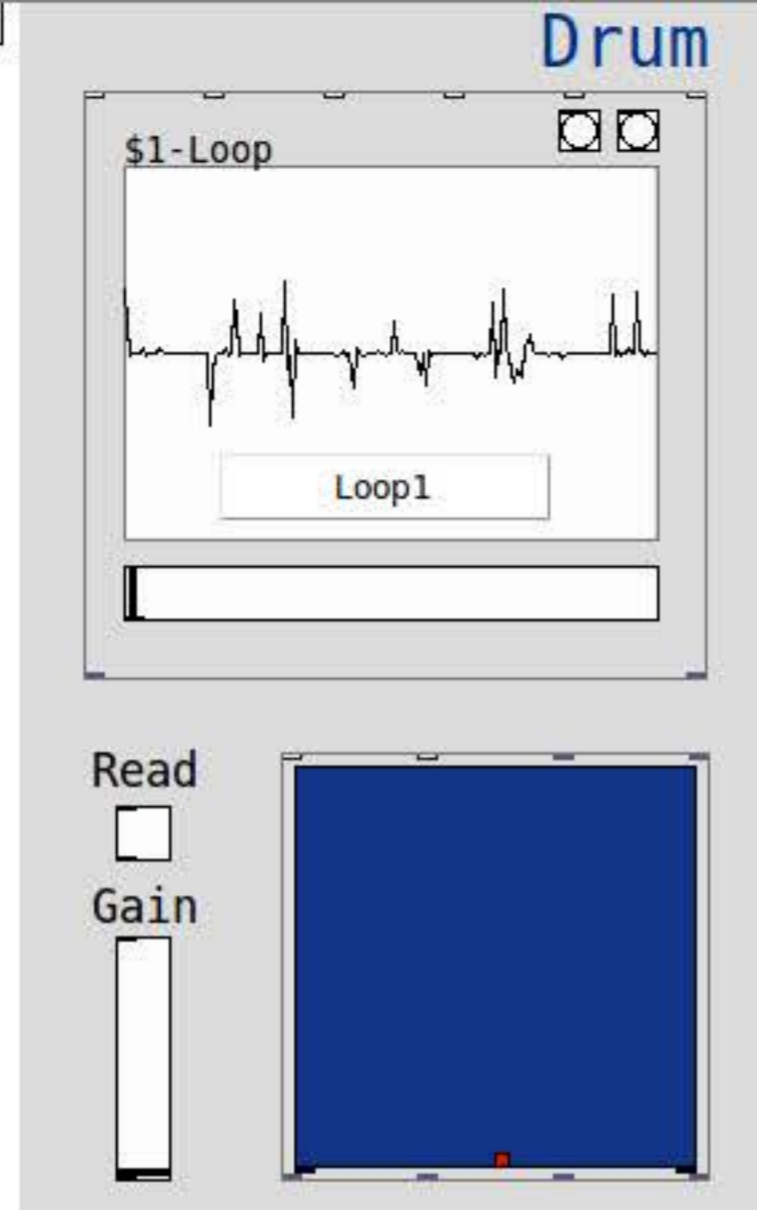
- Top Row:** Four sub-patches for 'drum', 'bass', 'lSynth', and 'sSynth'. Each contains an 'init' button, a 'sample' dropdown (set to 0), a 'play' checkbox (checked), a 'gain' slider (set to 0,000), a 'resetSpeed' button, and a 'speed' slider (set to 0,00).
- Second Row:** A sub-patch for 'ringMod' (freq, amount, sample), a sub-patch for 'delay' (time, feedback, interpolation, sample), and a sub-patch for 'spat' (depth, pan, sample).
- Bottom Row:** A 'master' sub-patch with a 'DSP' checkbox (checked) and a 'gain' slider (set to 1,000).

INTERLOPER

Thomas CIPIERRE

MUSIC
NINE INCH NAILS
The Greater Good
(2007 - Year Zero)

Drum



\$1-Loop

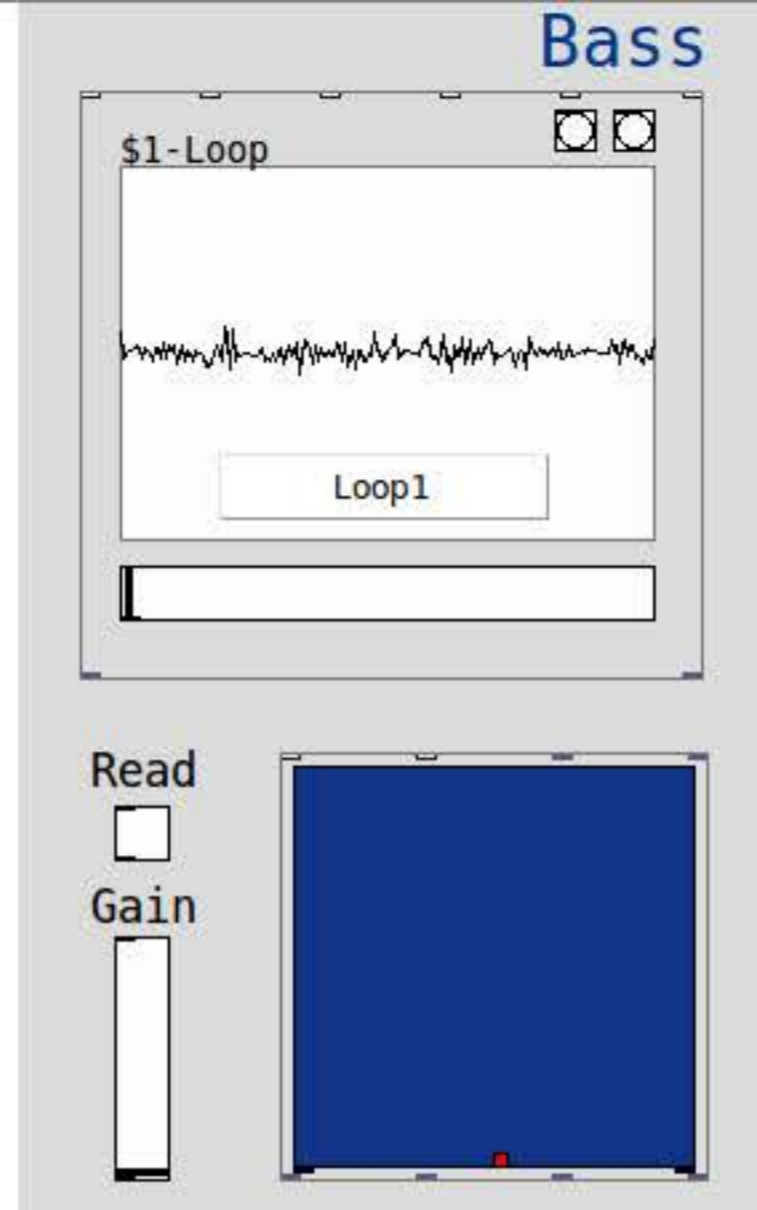
Loop1

Read

Gain

A digital audio workstation interface for a drum track. It features a waveform display labeled '\$1-Loop' with a 'Loop1' button below it. Below the waveform are two vertical sliders for 'Read' and 'Gain', both currently at the bottom position.

Bass



\$1-Loop

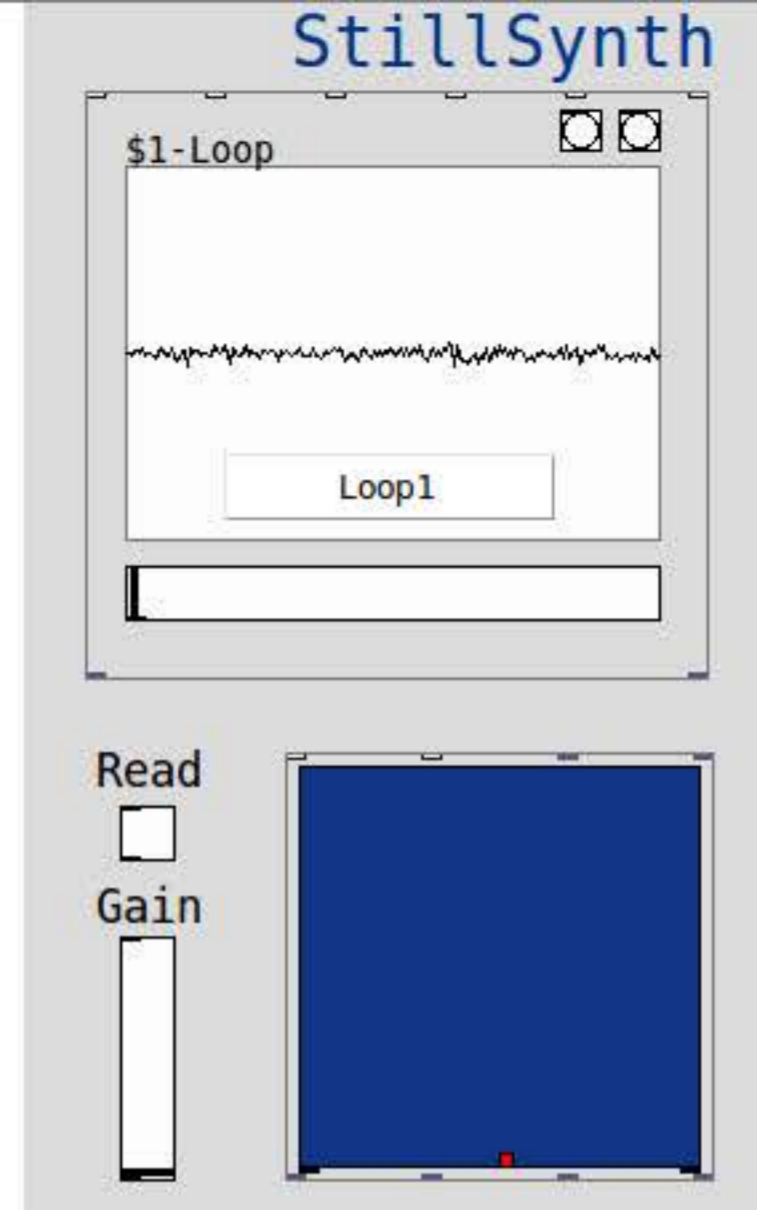
Loop1

Read

Gain

A digital audio workstation interface for a bass track. It features a waveform display labeled '\$1-Loop' with a 'Loop1' button below it. Below the waveform are two vertical sliders for 'Read' and 'Gain', both currently at the bottom position.

StillSynth



\$1-Loop

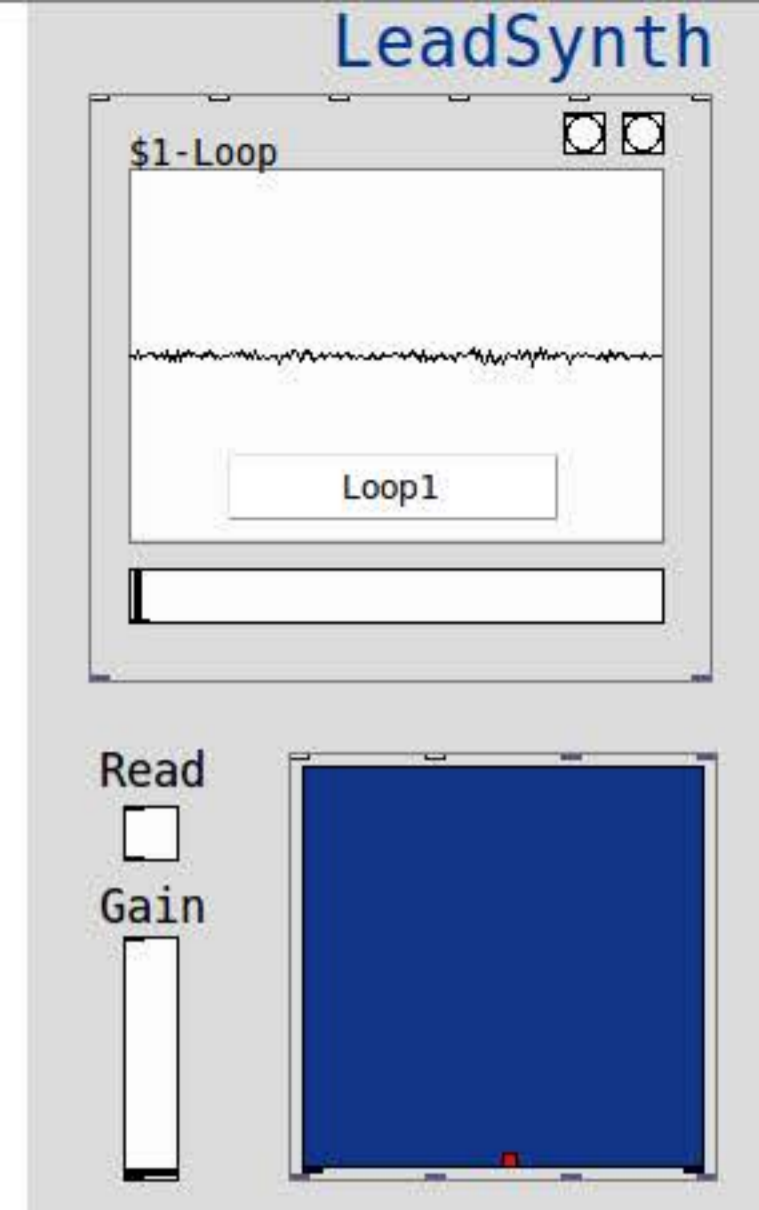
Loop1

Read

Gain

A digital audio workstation interface for a StillSynth track. It features a waveform display labeled '\$1-Loop' with a 'Loop1' button below it. Below the waveform are two vertical sliders for 'Read' and 'Gain', both currently at the bottom position.

LeadSynth



\$1-Loop

Loop1

Read

Gain

A digital audio workstation interface for a LeadSynth track. It features a waveform display labeled '\$1-Loop' with a 'Loop1' button below it. Below the waveform are two vertical sliders for 'Read' and 'Gain', both currently at the bottom position.

DELAY

On

Left_Time Right_Time

Feedback

A control panel for a delay effect. It includes an 'On' checkbox, two time sliders for 'Left_Time' and 'Right_Time' (both set to 0), and a 'Feedback' slider (set to 0).

DELAY

On

Left_Time Right_Time

Feedback

A control panel for a delay effect. It includes an 'On' checkbox, two time sliders for 'Left_Time' and 'Right_Time' (both set to 0), and a 'Feedback' slider (set to 0).

DELAY

On

Left_Time Right_Time

Feedback

A control panel for a delay effect. It includes an 'On' checkbox, two time sliders for 'Left_Time' and 'Right_Time' (both set to 0), and a 'Feedback' slider (set to 0).

DELAY

On

Left_Time Right_Time

Feedback

A control panel for a delay effect. It includes an 'On' checkbox, two time sliders for 'Left_Time' and 'Right_Time' (both set to 0), and a 'Feedback' slider (set to 0).

AMPLITUDE_MODULATION

Bass

On Amount

StillSynth

On Amount

LeadSynth

On Amount

A control panel for amplitude modulation. It has three sections: 'Bass', 'StillSynth', and 'LeadSynth'. Each section has an 'On' checkbox and an 'Amount' slider (all set to 0).

AMPLITUDE_MODULATION

Drum

On Amount

StillSynth

On Amount

LeadSynth

On Amount

A control panel for amplitude modulation. It has three sections: 'Drum', 'StillSynth', and 'LeadSynth'. Each section has an 'On' checkbox and an 'Amount' slider (all set to 0).

AMPLITUDE_MODULATION

Drum

On Amount

Bass

On Amount

LeadSynth

On Amount

A control panel for amplitude modulation. It has three sections: 'Drum', 'Bass', and 'LeadSynth'. Each section has an 'On' checkbox and an 'Amount' slider (all set to 0).

AMPLITUDE_MODULATION

Drum

On Amount

Bass

On Amount

StillSynth

On Amount

A control panel for amplitude modulation. It has three sections: 'Drum', 'Bass', and 'StillSynth'. Each section has an 'On' checkbox and an 'Amount' slider (all set to 0).

INTERLOPER
Thomas CIPIERRE

Master



192.168.0.100:5550

INTERLOPER
Thomas CIPIERRE

Drum



192.168.0.100:5530

INTERLOPER
Thomas CIPIERRE

Bass



192.168.0.100:5534

INTERLOPER
Thomas CIPIERRE

LeadSynth



192.168.0.100:5542

INTERLOPER
Thomas CIPIERRE

StillSynth



192.168.0.100:5538