

i-score: an overview

Linux Audio Conference - Workshop

Jean-Michaël Celerier

May 19, 2017

Context

Foundation: libossia

Goal

Protocols

Interoperability

The sequencer: i-score

Control

Temporal structure

Interactivity

Devices

Audio features

Conclusion

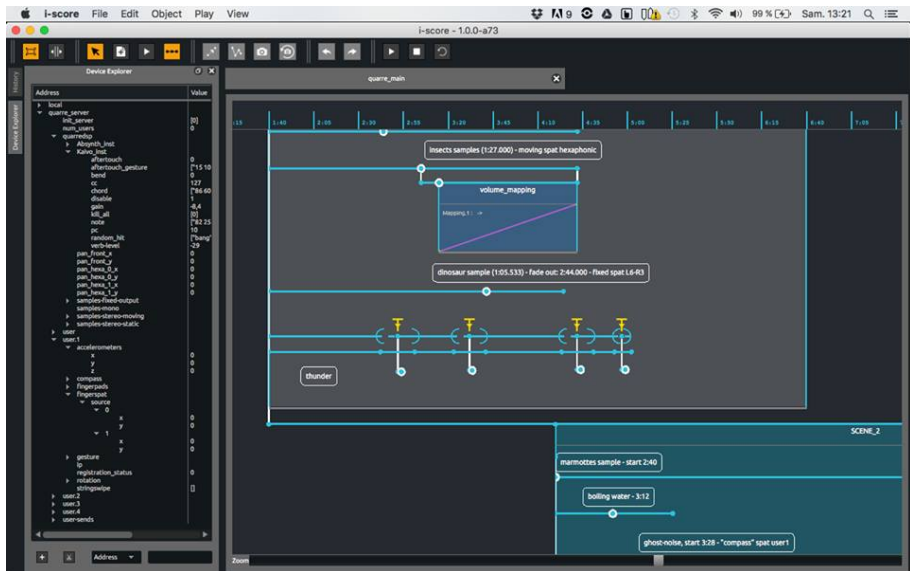
Workshop

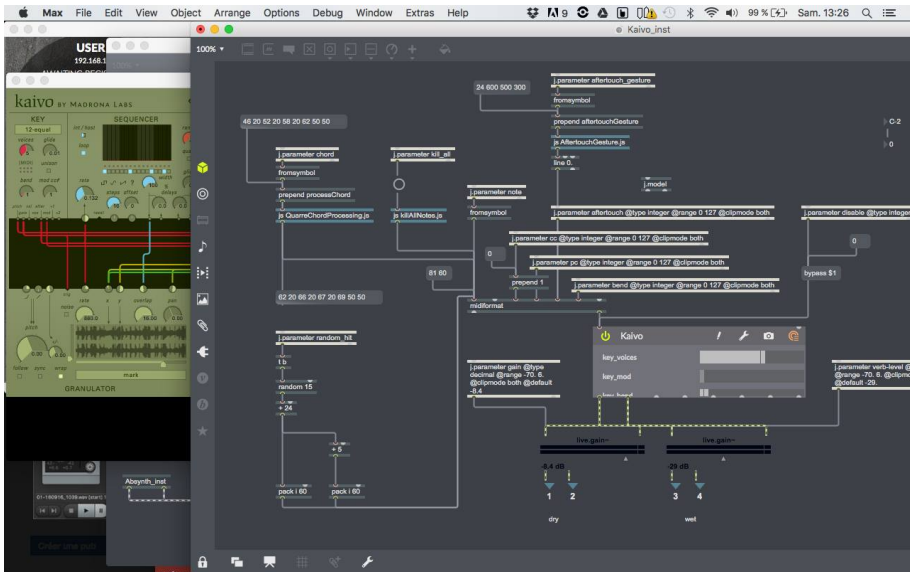
Les Baltazars - Tumbleweed



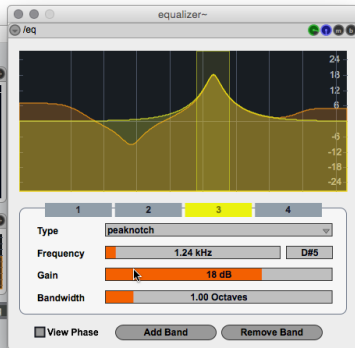
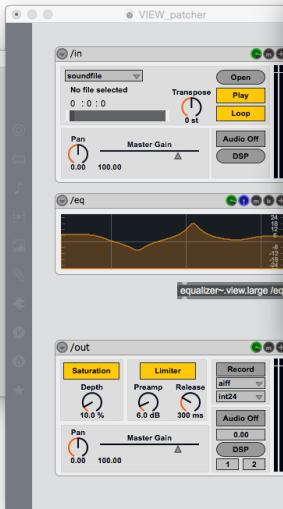
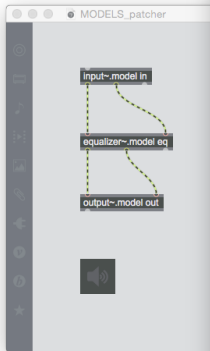
Pierre Cochard - Quarre







- ▶ Digital arts: music, video, transmedia, etc...
- ▶ Temporal structure & interactivity.
- ▶ Interoperability: software, hardware.



libossia: goals

- ▶ Automatic discovery.
- ▶ Shared object model inspired from OSC.
- ▶ Scoring primitives.

libossia: protocols

- ▶ OSC
- ▶ Minuit
- ▶ OSCQuery
- ▶ MIDI
- ▶ HTTP (Requires Qt)
- ▶ WebSocket (Requires Qt)
- ▶ Serial port (Requires Qt)

To come: ArtNet (DMX)

Standard protocols

Address	Value	Get	Set	Min	Max
▼ OSCdevice					
▼ mouse					
move	[0, 0]	✓	✓		
click	[0, 0]	✓	✓		
release	[0, 0]	✓	✓		
▼ particle					
density	0	✓	✓	1	50
radius	0	✓	✓		
color	[0, 0, 0]	✓	✓		
▼ VDMX					
▼ layer.1					
alpha	1	✓	✓	0	1
▼ position					
x	0.5	✓	✓	0	1
y	0.5	✓	✓	0	1
▼ scale					
x	0.5	✓	✓	0	1
y	0.5	✓	✓	0	1

Qt-based protocols

When no tree structure easily makes sense,
let the user script it !

```
function onMessage(message) {
    console.log(message);
    var res = JSON.parse(message);
    console.log(res.value);
    if(res.name === "toto")
    {
        return [ { address: "/toto", value: res.value } ];
    }
    return { };
}

function createTree() {
    return [ {
        name: "tata",
        children: [
            {
                name: "tutu",
                request: "{ \"name\": \"toto\", \"value\"
                type: Ossia.Float,
```

libossia (C++14)

Linux, macOS, Windows, GCC, Clang, MSVC, static, dynamic...

Only header-only dependencies.

```
auto& node = find_or_create_node(device, "/test/my_int");  
auto address = node.create_address(val_type::INT);
```

```
node.set(access_mode_attribute{}, access_mode::GET);  
node.set(bounding_mode_attribute{}, bounding_mode::FOLD);  
node.set(domain_attribute{}, make_domain(2, 14));  
node.set(description_attribute{}, "an_integral_value");
```

```
address->add_callback([] (const auto& val) {  
    std::cout << val << "\n";  
});
```

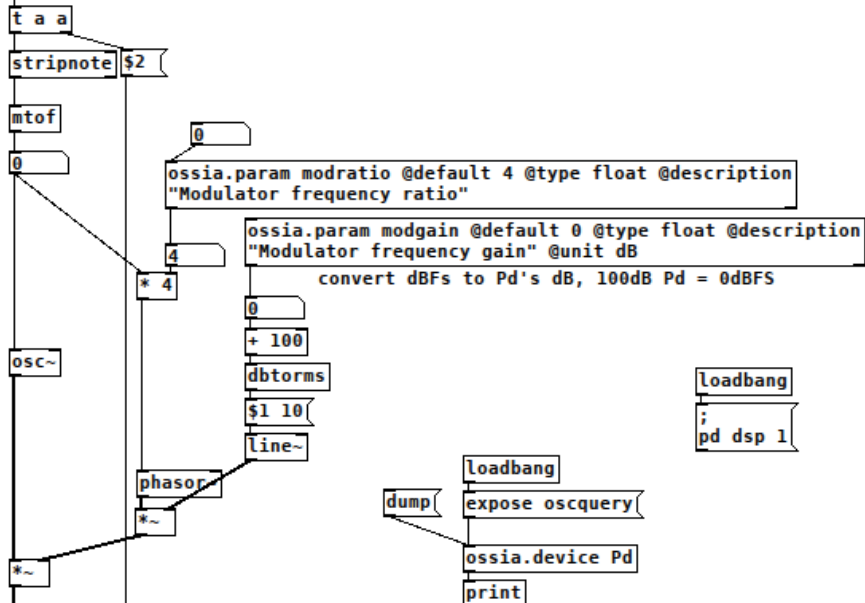
```
address->push_value(5678);
```

Integration with ofParameter, ofParameterGroup

```
ossia :: Parameter<bool> _fill;  
ossia :: Parameter<ofColor> _color;  
ossia :: ParameterGroup _sizeParams;  
...  
_circleParams.setup(_parent_node, "circle");  
  
_sizeParams.setup(_circleParams, "sizeParams");  
_radius.setup(_sizeParams, "radius", 10., 1., 100.);  
_position.setup(_sizeParams,  
    "position",  
    ofVec2f(ofGetWidth() / 2, ofGetHeight() / 2),  
    ofVec2f(0., 0.), // Min  
    ofVec2f(ofGetWidth(), ofGetHeight())); // Max
```


ossia-pd (and ossia-max soon)

```
ossia.param note @type vec2f @description "Midi note and velocity"
```



ossia-python

```
# create a node, create a tuple address and initialize it
tuple_node = local_device.add_node("/test/value/tuple")
tuple_address = tuple_node.create_address(
    ossia.ValueType.Tuple)

tuple_value = ossia.Value([
    ossia.Value(44100),
    ossia.Value("test.wav"),
    ossia.Value(0.9)]
)
tuple_address.push_value(tuple_value)

# attach a callback function to the boolean address
def bool_value_callback(v):
    print(v.get())
    bool_address.add_callback(bool_value_callback)
```

ossia-unity3D (C#)

```
public class Foo : MonoBehaviour
{
    [Ossia.Attribute]
    int foo;
}
```

ossia-qml (Qt QML)

```
Item {  
    Ossia.Node { name: 'test' }  
    AngleSlider {  
        // Reads and writes from /test/angle  
        Ossia.Property on angle {  
            min: -90  
            max: 0  
            bounding: Ossia.Context.Clip  
        }  
    }  
}
```

ossia-C (C99)

OSSIA_EXPORT

```
bool ossia_device_update_namespace(  
    ossia_device_t device);
```

OSSIA_EXPORT

```
ossia_node_t ossia_device_get_root_node(  
    ossia_device_t device);
```

OSSIA_EXPORT

```
const char* ossia_device_get_name(  
    ossia_device_t node);
```

//// Node ////

OSSIA_EXPORT

```
ossia_node_t ossia_node_add_child(  
    ossia_node_t node,  
    const char * name);
```

Demonstration

i-score + PureData + Processing

Sending messages

The screenshot displays a software interface for managing messages, likely for a microcontroller or embedded system. The interface is divided into several panels:

- Top Panel:** Contains a menu bar (File, Edit, Object, Play, Tool, View, Settings, About) and a toolbar with various icons for navigation and editing.
- Device Explorer (Left Panel):** Shows a tree view of the device's memory map. The "OSCdevice" is expanded, showing its address and value.
- Timeline (Center Panel):** A horizontal timeline with a time scale from 0:00.0 to 0:10.0. A blue line represents a message being sent, labeled "Example /". A yellow circle with a plus sign indicates a trigger point at approximately 0:02.500.
- Inspector (Right Panel):** Provides details about the selected object, "laws73dock5". It includes fields for Name, Label, Default date (2 s 234 ms), and a Trigger section with an "Enable trigger" button. Below these are sections for Events, Condition, and State.

The **State** section contains a table with the following data:

Address	Value
OSCdevice example	0

Below the table, there is a "Prev. Constraint" button and an "Add process" button.

Automating

The screenshot displays a software interface for automating OSC device control, featuring a timeline, a device explorer, and an inspector panel.

Device Explorer: A table listing OSC devices and their values.

Address	Val
OSCdevice	
example	0

Timeline: A horizontal timeline with time markers at 0:00.0, 0:02.500, 0:05.0, 0:07.500, and 0:10.0. A blue line labeled "Example /" spans the duration. A blue box labeled "automation" is positioned on the timeline, containing the text "...Ice/example".

Inspector: A panel on the right side of the interface showing the properties of the selected automation object.

Name: automation

Label:

Speed x1: A dropdown menu with options x0, x0.5, x1, x2, and x5. The x1 option is selected.

Start State: A button labeled "Start State".

Full view: A button labeled "Full view".

End State: A button labeled "End State".

Default Duration: 0.00.03.786

Add Process: A button labeled "Add Process".

Automation.1: A section containing the following properties:

- Address:** OSCdevice/example
- Tween:** ☐
- Min:** 0.00000
- Max:** 1.00000
- Unit:** None
- Start:** OSCdevice/example 0
- End:** OSCdevice/example 1

Interpolating

The screenshot displays a software interface for creating and editing automation. The main workspace shows a timeline with a clip labeled "Example /" containing an "automation" object. The automation object's graph shows a linear interpolation from a value of 10 at the start to 2 at the end. The Inspector panel on the right provides detailed settings for the selected "Interpolation.1" process.

Device Explorer

Address	Val
OSCdevice	
example	10

Inspector

Name automation

Label

Speed x1
x 0 | x 0.5 | x 1 | x 2 | x 5

Start State Full view End State

Default Duration 0.00.03.786

+ Add Process

Interpolation.1

Address OSCdevice/example

Tween ☐

Start -2

End 10

Tweening

The screenshot displays a software interface for creating and editing animations, specifically focusing on a 'tweening' process. The interface is divided into several panels:

- Top Panel:** Contains a menu bar (File, Edit, Object, Play, Tool, View, Settings, About) and a toolbar with various icons for navigation and editing.
- Device Explorer:** Located on the left, it shows a table with columns 'Address' and 'Val'.
- Timeline:** The central area features a horizontal timeline with markers at 0:00.0, 0:02.500, 0:05.0, 0:07.500, and 0:10.0. A blue bar labeled 'Example /' spans the timeline. Below it, a blue bar labeled 'tweening' is shown, with a red line indicating a value change over time.
- Inspector:** On the right, it provides configuration options for the selected 'tweening' process.
 - Name:** tweening
 - Label:** (empty)
 - Speed:** x1 (with options x0, x0.5, x1, x2, x5)
 - Buttons:** Start State, Full view, End State
 - Default Duration:** 0.00.06.953
 - Add Process:** (+)
 - Automation.1:** (expanded)
 - Address:** (empty)
 - Tween:** ☒
 - Min:** 0.00000
 - Max:** 0.00000
 - Unit:** None
 - Start:** 0
 - End:** 0

Mapping

The screenshot displays a software interface for managing mappings, likely for a MIDI or audio application. The interface is divided into several panels:

- Device Explorer:** Located on the left, it shows a tree view of devices. Under "OSCdevice", there are two entries: "other" and "example". The "example" entry is selected, and its value is 0.
- Timeline:** The central area shows a timeline with a blue bar representing a process. The timeline has markers at 0:00.0, 0:02.500, 0:05.0, 0:07.500, and 0:10.0. A blue box labeled "mapping" is positioned on the timeline, starting at 0:02.500 and ending at 0:07.500. Inside this box, a yellow text label reads "Mapping.1: ...ice:/example -> ...vice:/other".
- Inspector:** Located on the right, it provides details about the selected "mapping" process. It shows the "Name" as "mapping" and the "Label" as "Label". Below this, there are controls for "Speed x1" with a slider and buttons for "Start State", "Full view", and "End State". The "Default Duration" is set to 0.00.05.525. Under the "Mapping.1" section, the "Source" is "OSCdevice:/example" and the "Target" is "OSCdevice:/other". The "Min" value is 0.00000 and the "Max" value is 1.00000.

Scripting

The screenshot displays a software interface for scripting, likely for a game engine or animation tool. The interface is divided into several panels:

- Top Panel:** Contains a menu bar (File, Edit, Object, Play, Tool, View, Settings, About) and a toolbar with various icons for navigation and editing.
- Device Explorer:** Located on the left, it shows a list of objects (Address, Val) and a timeline with markers at 0:00.0, 0:02.500, 0:05.0, 0:07.500, and 0:10.0.
- Script Editor:** The central area shows a script titled "script" with the following code:

```
1 (function(t) {  
2   var obj = new Object;  
3   obj["address"] = 'OSCdevice:/millumin/layer/x/instance';  
4   obj["value"] = t + iscore.value('OSCdevice:/millumin/layer/y/instance');  
5   return [ obj ];  
6 });
```
- Inspector:** Located on the right, it shows the properties of the selected object (Example). The properties include:
 - Name:** Example
 - Label:** Label
 - Speed x1:** A slider with values x0, x0.5, x1, x2, x5.
 - Trigger:** A dropdown menu set to False.
 - Start State:** A button labeled "Start State".
 - Full view:** A button labeled "Full view".
 - End State:** A button labeled "End State".
 - Default Duration:** A dropdown menu set to 0.00.10.000.
 - Min Duration:** A dropdown menu set to 0.00.10.000.
 - Max Duration:** A dropdown menu set to Infinity.
 - Add Process:** A button labeled "Add Process".
 - Scenario:** A dropdown menu set to Scenario.1.

Racks

The screenshot displays the Racks software interface, which is used for creating and editing audio racks and mappings.

Device Explorer: Located on the left, it shows a list of devices. The selected device is `OSCdevice`, with a sub-device `example` highlighted. The value for `example` is 0.

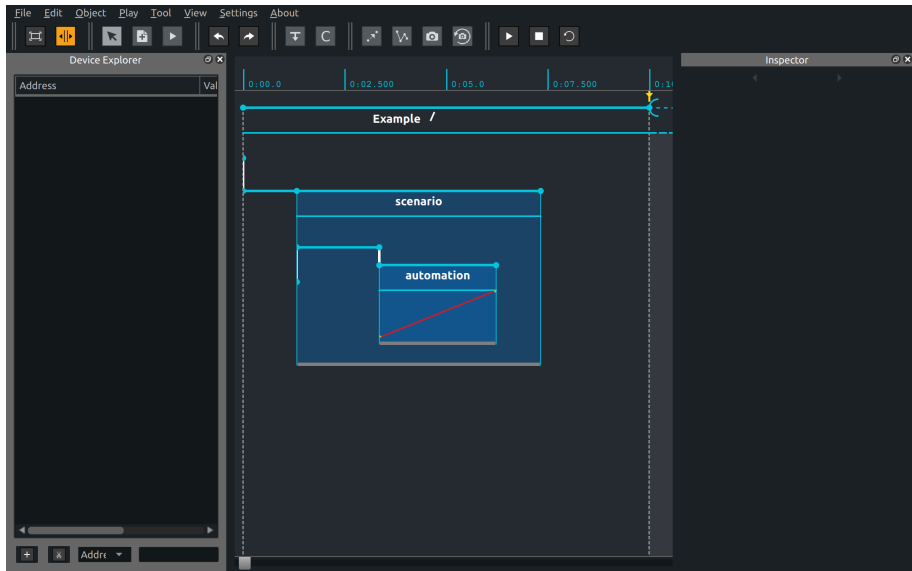
Timeline: The top of the main workspace shows a timeline with time markers at 0:00.0, 0:02.500, 0:05.0, 0:07.500, and 0:10.0. A blue line represents the current state of the rack over time.

Rack: The central workspace shows a rack with a blue background. It contains a mapping labeled `Mapping.1: ...ice:/example -> ...evice:/other`. The mapping is visualized with a purple curve and a red line. A `pattern` box is also visible within the rack.

Inspector: Located on the right, it provides detailed information about the selected object. The selected object is `Name rack`. The inspector shows the following settings:

- Name:** rack
- Label:** Label
- Speed:** x1, x0, x0.5, x1, x2, x5
- Start State:** Start State
- Full view:** Full view
- End State:** End State
- Default Duration:** 0.00.05.525
- Add Process:** Add Process
- Mapping.1:**
 - Source:** OSCdevice:/example
 - Min:** 0.00000
 - Max:** 1.00000
 - Target:** OSCdevice:/other
 - Min:** -100.00000
 - Max:** 100.00000
- Automation.2:**
 - Address:** Address
 - Tween:** ☐
 - Min:** 0.00000
 - Max:** 0.00000
 - Unit:** None
 - Start:** 0
 - End:** 0
- Loop.3:** Loop.3
- Name:** pattern

Hierarchy



Loops

The screenshot displays a software interface for creating and editing loops and patterns. The main workspace shows a timeline with a blue bar labeled "Example /" and a nested blue bar labeled "loop". Inside the "loop" bar is a purple bar labeled "pattern". A red line connects the start of the "pattern" bar to the end of the "loop" bar, indicating a loop connection. The timeline has time markers at 0:00.0, 0:02.500, 0:05.0, and 0:07.500.

The interface includes a menu bar (File, Edit, Object, Play, Tool, View, Settings, About) and a toolbar with various icons. On the left, there is a "Device Explorer" panel with columns for "Address" and "Value". On the right, there is an "Inspector" panel showing the properties of the selected object, "loop".

The "Inspector" panel for "loop" shows the following properties:

- Name: loop
- Label: (empty)
- Speed x1: x0 | x0.5 | x1 | x2 | x5
- Buttons: Start State, Full view, End State
- Default Duration: 0.00.07.017
- + Add Process
- Loop.1
 - Name: pattern
 - Label: (empty)
 - Speed x1: x0 | x0.5 | x1 | x2 | x5
 - Buttons: Start State, Full view, End State
 - Default Duration: 0.00.03.437
 - + Add Process
 - Automation.1
 - Address: (empty)
 - Tween: ☐
 - Min: 0.000000
 - Max: 0.000000
 - Unit: None
 - Start: 0
 - End: 0

Interactivity: conditions

The screenshot displays a visual scripting environment with a timeline and an Inspector panel.

Timeline: The timeline at the top shows a duration from 0:00.0 to 0:10.0. A blue line labeled "Condition /" spans the entire duration. A yellow circle highlights a node on the timeline at approximately 0:02.500.

Inspector: The Inspector panel on the right shows the configuration for the selected node:

- Name:** bibb54hymn24
- Label:**
- Default date:** 4 s 809 ms
- Trigger:** Enable trigger
- Events:**
 - skim8bogs97
 - alto6axon42
 - Name: alto6axon42
 - Label:
 - Parent TimeNode
- Condition:**
 - a/b \geq 3
- Offset behaviour:** True
- State:** (multiple instances)
- flex20rung34**

Interactivity: trigger points

The screenshot displays a software interface for managing interactive elements, likely a game engine or animation tool. The main workspace is a timeline editor with a dark background and a light blue timeline axis. The timeline is divided into segments with time markers: 0:00.0, 0:02.500, 0:05.0, and 0:07.500. A label "Example /" is visible above the timeline. Several trigger points are marked on the timeline, each represented by a blue dot with a yellow "T" and a plus sign. These points are connected by dashed lines, indicating a sequence of events. The interface includes a top menu bar with options: File, Edit, Object, Play, Tool, View, Settings, and About. Below the menu is a toolbar with various icons for editing and playback. On the left, there is a "Device Explorer" panel with a table showing "Address" and "Val". On the right, there is an "Inspector" panel showing the properties of the selected object, "rips75scout16". The inspector includes a "Label" field, a "Default date" of 3 s 869 ms, a "Trigger" section with a dropdown menu set to "foo/bar", a "Condition" section, and a "State" section with a table showing "Address" and "Value".

File Edit Object Play Tool View Settings About

Device Explorer

Address Val

0:00.0 0:02.500 0:05.0 0:07.500

Example /

Inspector

Name rips75scout16

Label

Default date 3 s 869 ms

Trigger

foo/bar ≥ 3

Events

grim88joke98

Name grim88joke98

Label

Parent TimeNode

Condition

State

Address Value

Prev_Constraint

Add process

Interactivity: execution speed

The screenshot displays a software interface for managing execution speed, likely for a game engine or animation tool. The interface is divided into several panels:

- Top Panel:** Contains a menu bar (File, Edit, Object, Play, Tool, View, Settings, About) and a toolbar with various icons for navigation and editing.
- Device Explorer (Left Panel):** A panel with a table showing device information. The table has two columns: "Address" and "Val".
- Timeline (Center Panel):** A horizontal timeline with a vertical playhead. The timeline is divided into segments labeled "0:00.0", "0:10.0", "0:20.0", and "0:30.0". A blue line represents the execution path, and a green line indicates the current state. A label "Example /" is visible above the timeline.
- Inspector (Right Panel):** A panel showing the properties of the selected object. It includes a "Name" field (jilt95bbs52), a "Label" field, and a "Speed x5" dropdown menu with options: x0, x0.5, x1, x2, x5. Below the speed dropdown are buttons for "Start State", "Full view", and "End State". The "Default Duration" is set to 0.00.33.259. A button labeled "+ Add Process" is at the bottom.

Working with external devices

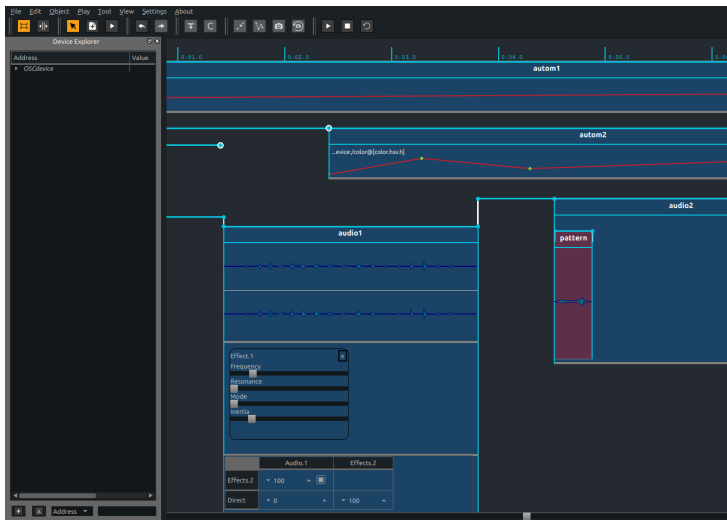
- ▶ Manual entry
- ▶ Loading
- ▶ Learning
- ▶ Automatic discovery

Demonstration

MIDI control surface and WebSockets



Audio

Hierarchical mixing, sounds, effects (Faust, LV2), sends...



The whole audio sequencing part is implemented with the
plug-in API.

What's missing

- ▶ Multichannel operation.
- ▶ Displaying LV2 UIs...
- ▶ Musical time structures (bars, metronome, etc).
- ▶  Packaging for distros 

Work-in-progress

- ▶ Embedded score player.
- ▶ Network operation.
- ▶ Full-fledged audiograph.
- ▶ Ongoing work on UI.

Workshop

- ▶ Building scores.
- ▶ Experimenting with your favorite environments.
- ▶ Gather your remarks and advices !