# 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

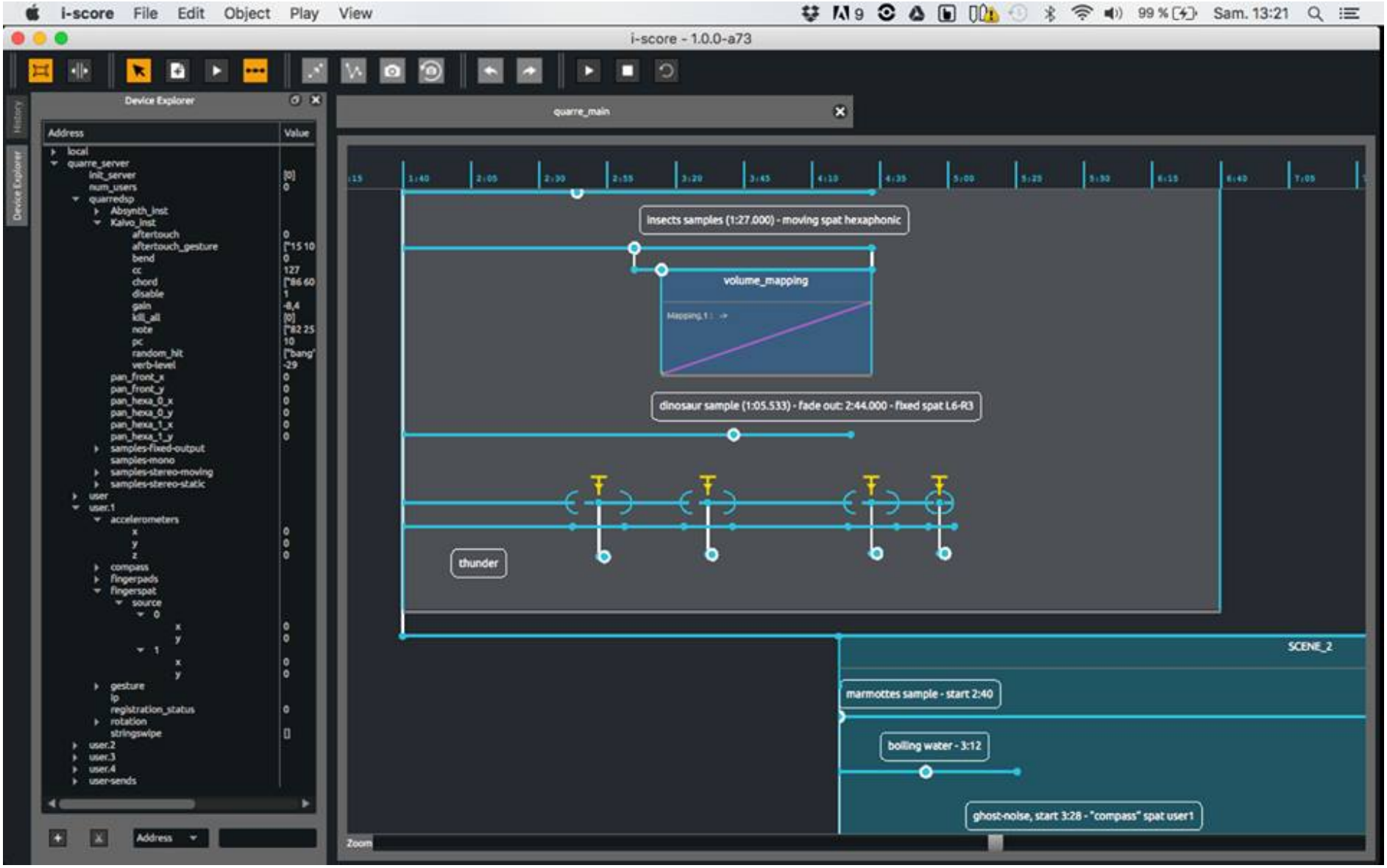
Audio features

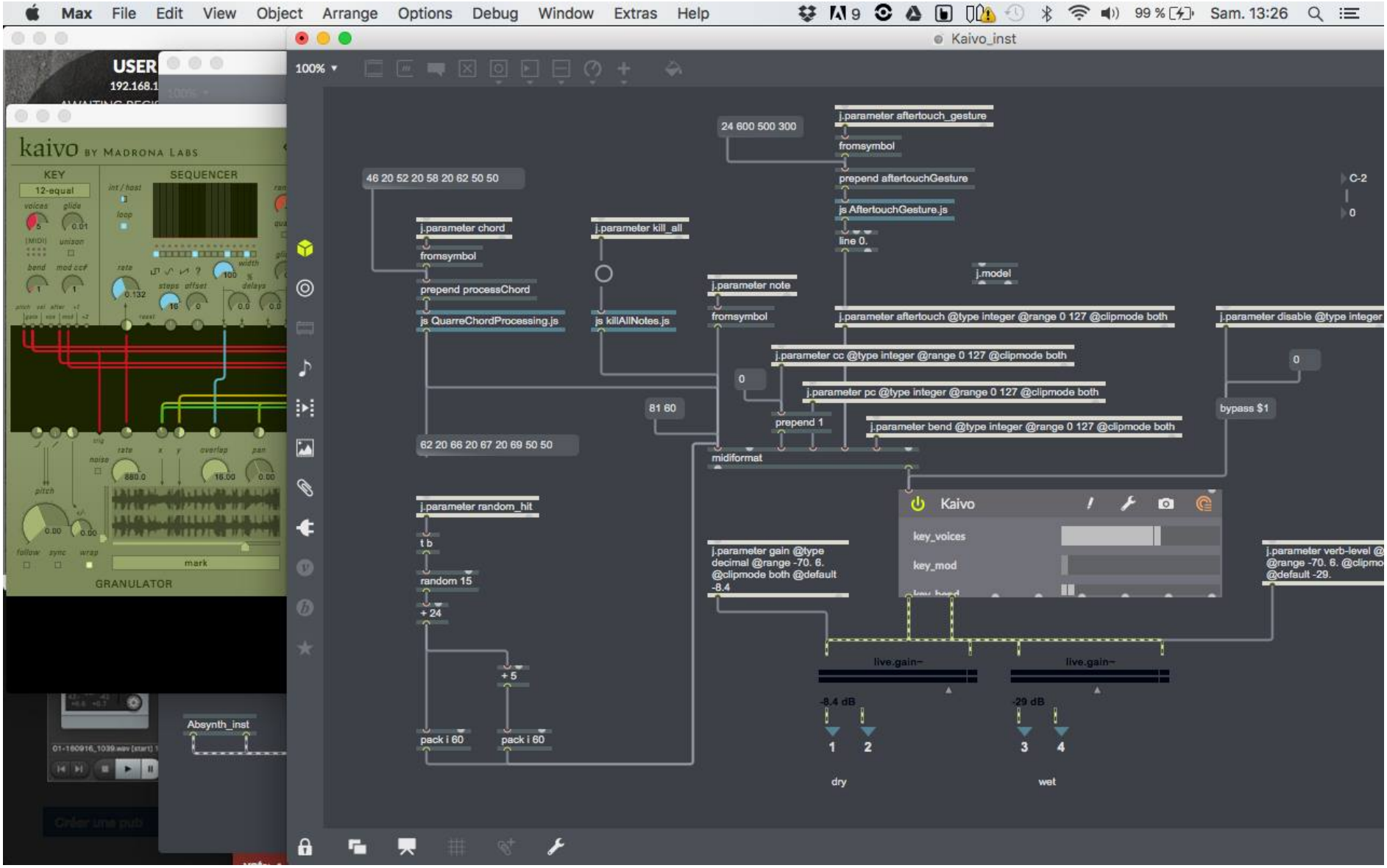
Conclusion

Workshop

# Les Baltazars - Tumbleweed

# Pierre Cochard - Quarrè


i-score File Edit Object Play View
Sam. 13:21
i-score - 1.0.0-a73
Device Explorer
Address
Value
quarre_main
insects samples (1:27.000) - moving spat hexaphonic
volume_mapping
dinosaur sample (1:05.533) - fade out: 2:44.000 - fixed spat L6-R3
thunder
SCENE_2
marmottes sample - start 2:40
boiling water - 3:12
ghost-noise, start 3:28 - "compass" spat user1
Zoom


Max File Edit View Object Arrange Options Debug Window Extras Help
Sam. 13:26
Kaivo_inst
100%
kaivo BY MADRONA LABS
KEY
12-equal
SEQUENCER
GRANULATOR
mark
Absynth_inst
24 600 500 300
j.parameter aftertouch_gesture
fromsymbol
prepend aftertouchGesture
js AftertouchGesture.js
line 0.
46 20 52 20 58 20 62 50 50
j.parameter chord
fromsymbol
prepend processChord
js QuarreChordProcessing.js
j.parameter kill_all
js killAllNotes.js
j.parameter note
fromsymbol
j.model
C-2
0
j.parameter aftertouch @type integer @range 0 127 @clipmode both
j.parameter disable @type integer
j.parameter cc @type integer @range 0 127 @clipmode both
j.parameter pc @type integer @range 0 127 @clipmode both
j.parameter bend @type integer @range 0 127 @clipmode both
prepend 1
81 60
bypass $1
62 20 66 20 67 20 69 50 50
midiformat
Kaivo
key_voices
key_mod
j.parameter random_hit
t b
random 15
+ 24
+ 5
pack i 60
pack i 60
j.parameter gain @type decimal @range -70. 6. @clipmode both @default -8.4
j.parameter verb-level @range -70. 6. @clipmode @default -29.
live.gain~
live.gain~
-8.4 dB
-29 dB
1 2
3 4
dry
wet

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


MODELS_patcher
input~.model in
equalizer~.model eq
output~.model out
VIEW_patcher
/in
soundfile
No file selected
0 : 0 : 0
Transpose
0 st
Open
Play
Loop
Pan
0.00
Master Gain
100.00
Audio Off
DSP
/eq
equalizer~.view.large /eq
/out
Saturation
Depth
10.0 %
Limiter
Preamp
6.0 dB
Release
300 ms
Record
aiff
int24
Audio Off
0.00
DSP
1
2
Pan
0.00
Master Gain
100.00
equalizer~
/eq
24
18
12
6
-6
-12
-18
-24
1
2
3
4
Type
peaknotch
Frequency
1.24 kHz
D#5
Gain
18 dB
Bandwidth
1.00 Octaves
View Phase
Add Band
Remove Band

# 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\": $val
                             }",
                        type: Ossia.Float,
                        unit: "color.rgb"
                    },
```

# libossia (C++14)

Linux, macOS, Windows, GCC, Clang, MSVC, static, dynamic...
Only header-only dependencies.

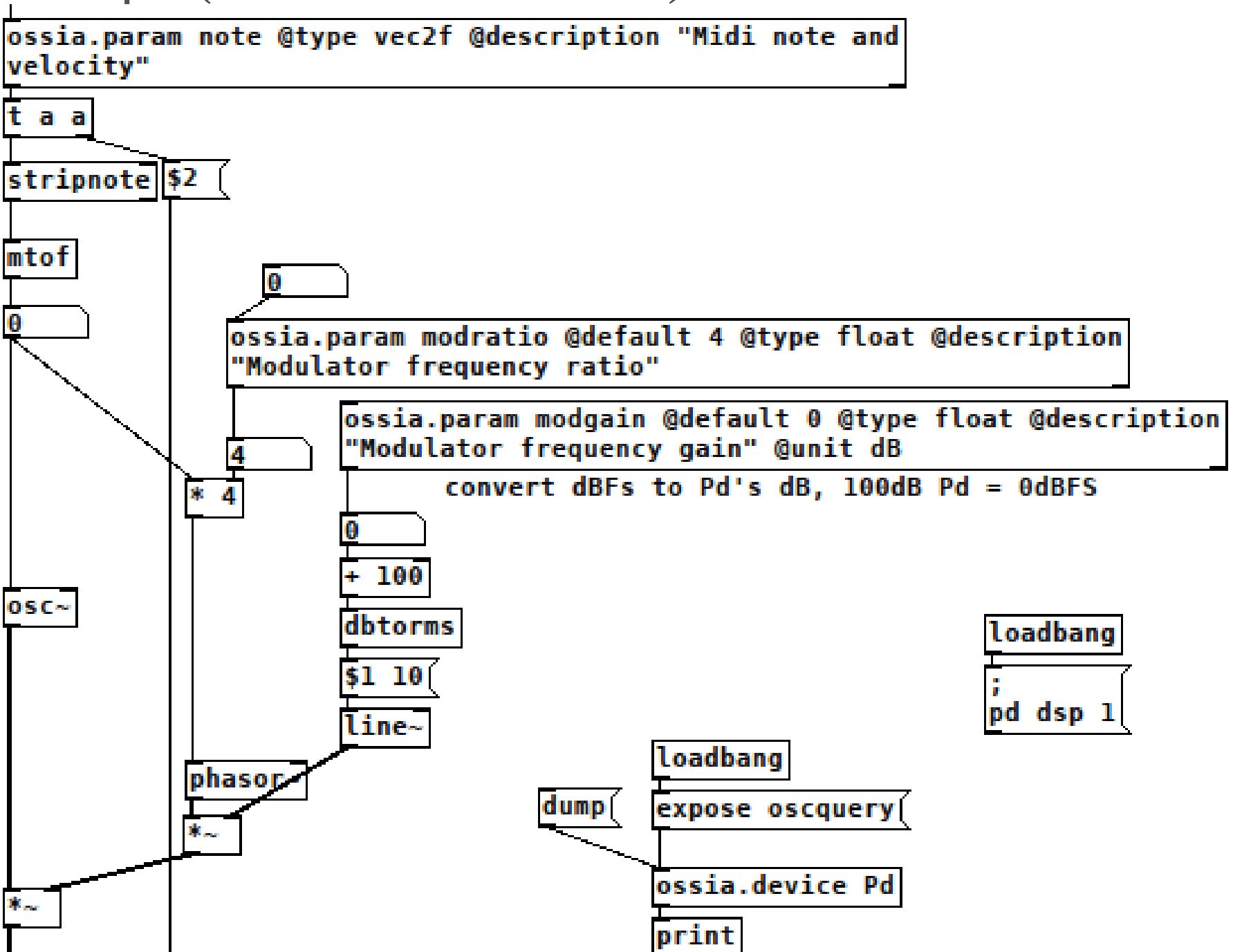
```
auto& node = find_or_create_node(device, "/test/my_int");
auto address = node.create_address(val_type::INT);

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 << " ";
  });

address->push_value(5678);
```

# ofxOssia

## 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-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 : public 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
    }
  }
}
```

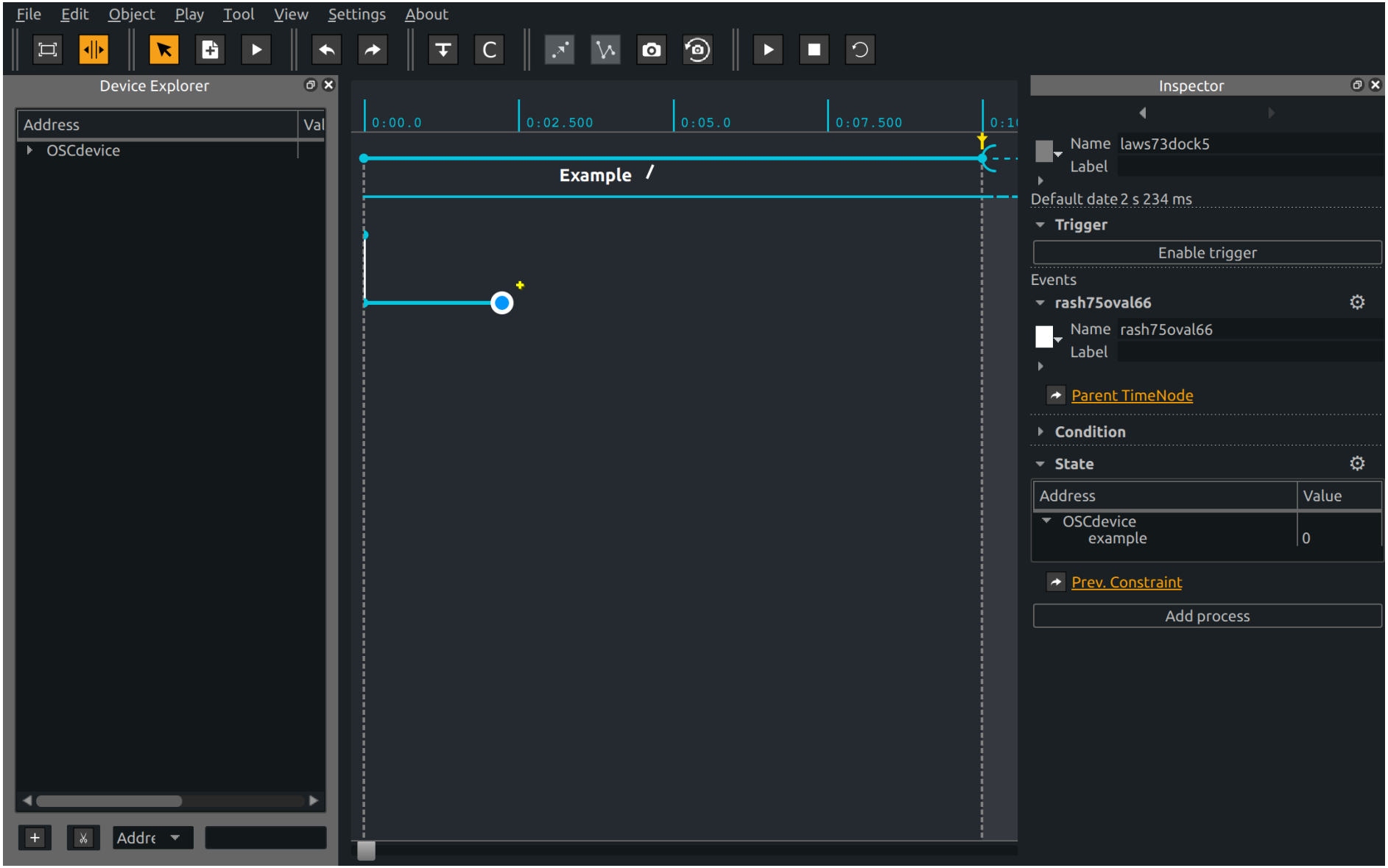
# ossia-C (C99)

```
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);
```

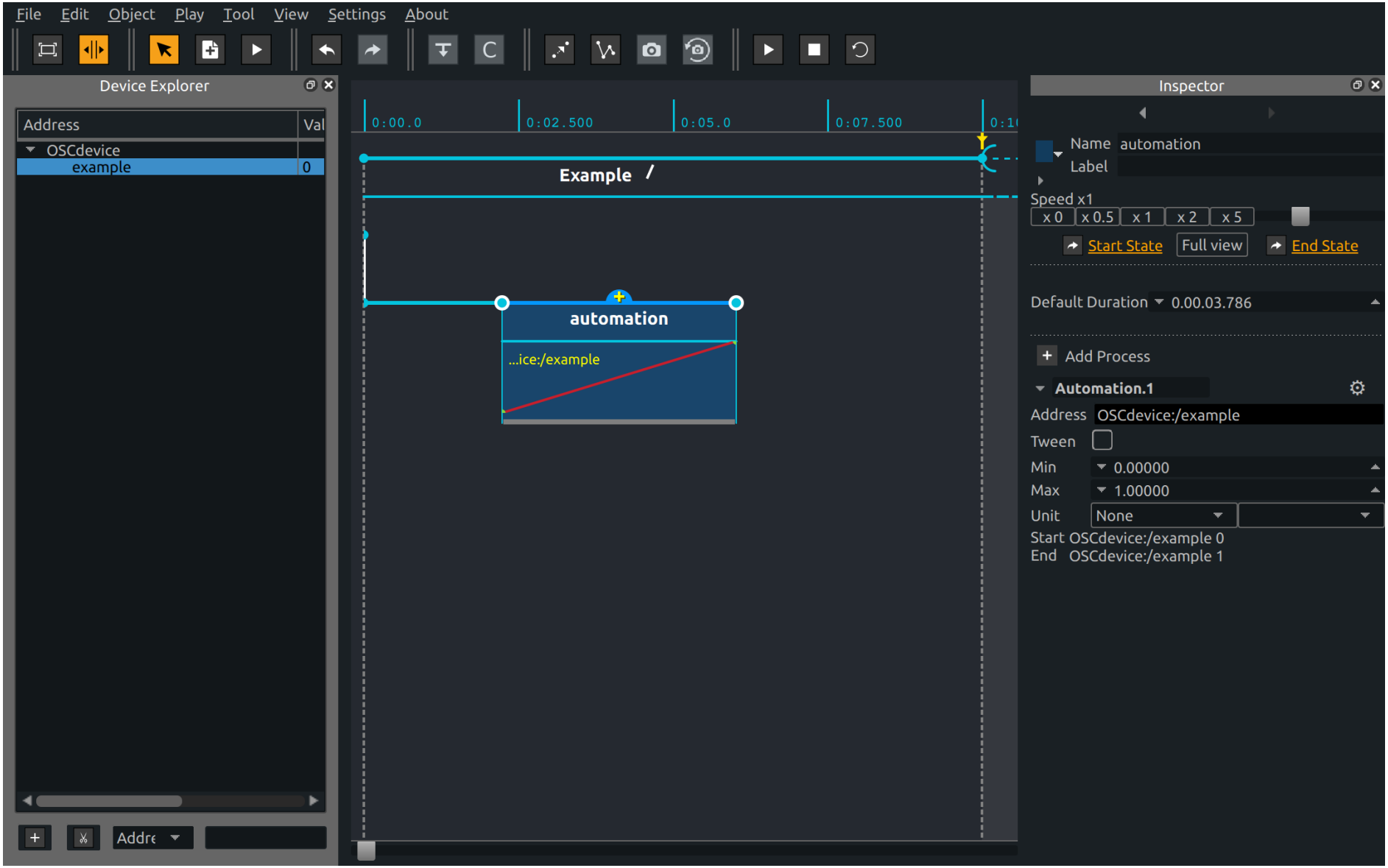
# Demonstration

i-score + PureData + Processing

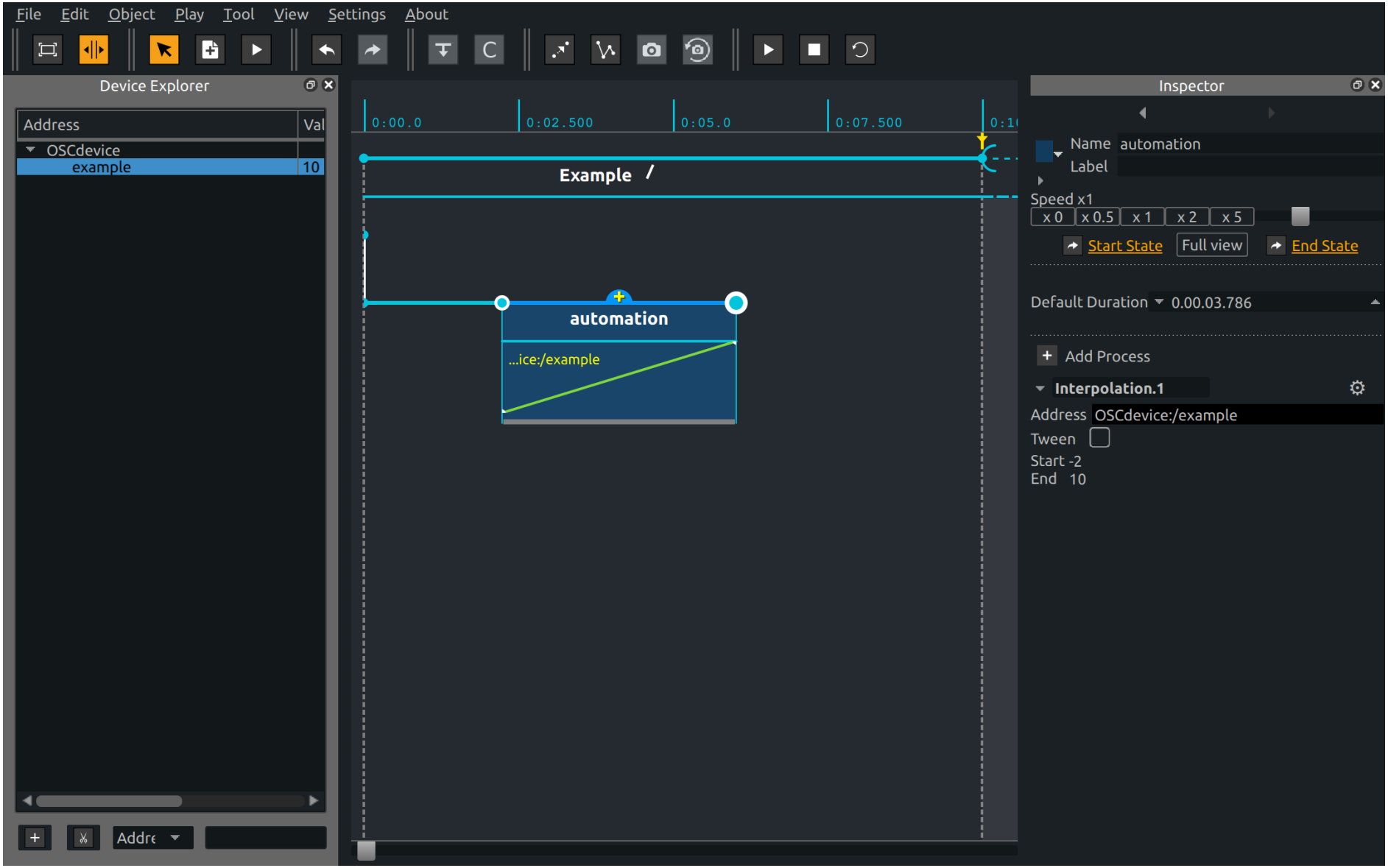
# Sending messages



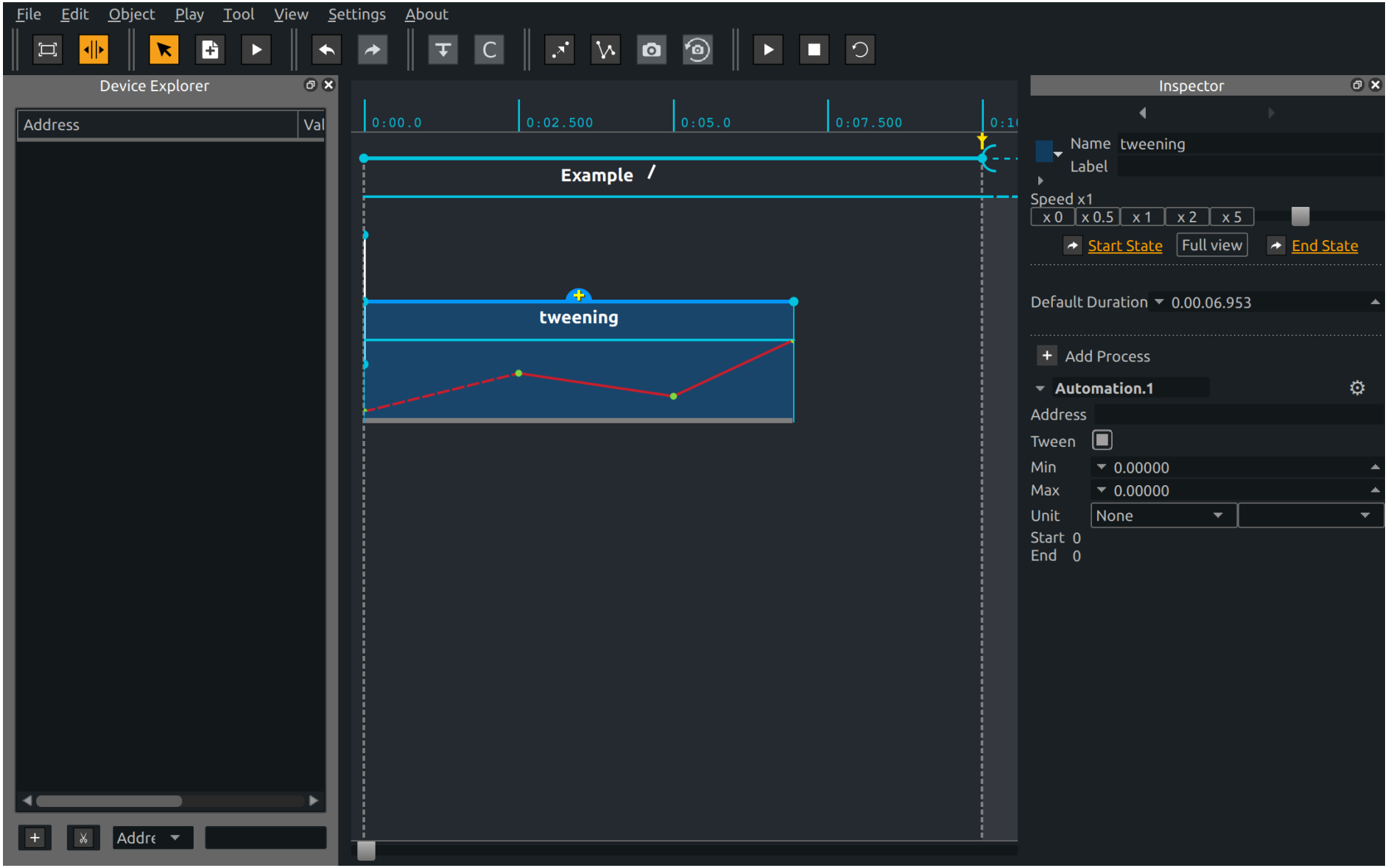
# Automating


File Edit Object Play Tool View Settings About
Device Explorer
Address
Val
OSCdevice
example
0
Addre
0:00.0
0:02.500
0:05.0
0:07.500
Example
automation
...ice:/example
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
Automation.1
Address OSCdevice:/example
Tween
Min 0.00000
Max 1.00000
Unit None
Start OSCdevice:/example 0
End OSCdevice:/example 1

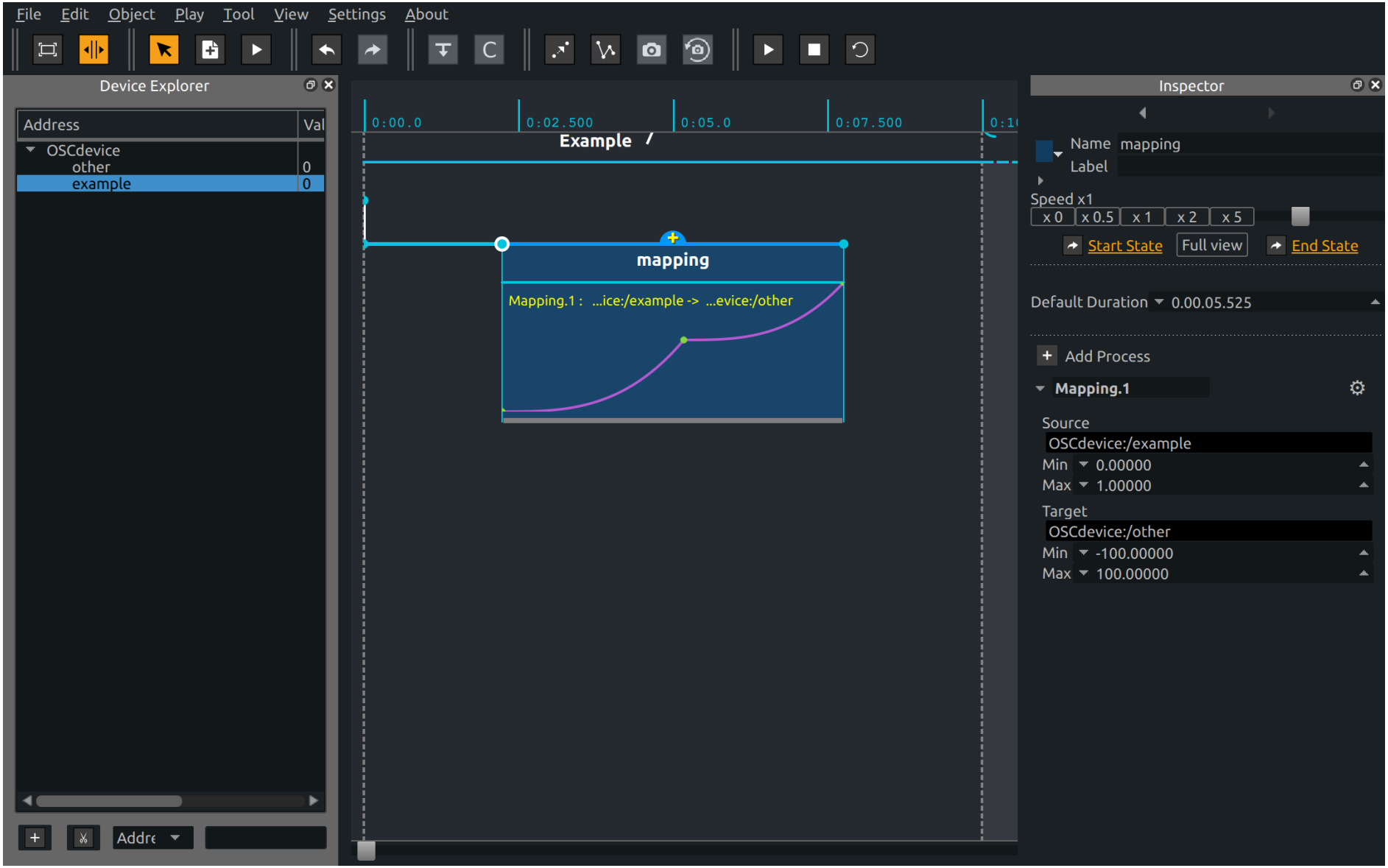
# Interpolating



# Tweening


File
Edit
Object
Play
Tool
View
Settings
About
Device Explorer
Address
Val
Addre
0:00.0
0:02.500
0:05.0
0:07.500
Example
tweening
Inspector
Name tweening
Label
Speed x1
x 0
x 0.5
x 1
x 2
x 5
Start State
Full view
End State
Default Duration 0.00.06.953
Add Process
Automation.1
Address
Tween
Min 0.00000
Max 0.00000
Unit None
Start 0
End 0

# Mapping


File
Edit
Object
Play
Tool
View
Settings
About
Device Explorer
Address
Val
OSCdevice
other
0
example
0
Addre
0:00.0
0:02.500
0:05.0
0:07.500
Example
mapping
Mapping.1 : ...ice:/example -> ...evice:/other
Inspector
Name
mapping
Label
Speed x1
x 0
x 0.5
x 1
x 2
x 5
Start State
Full view
End State
Default Duration
0.00.05.525
Add Process
Mapping.1
Source
OSCdevice:/example
Min
0.00000
Max
1.00000
Target
OSCdevice:/other
Min
-100.00000
Max
100.00000

# Scripting

File Edit Object Play Tool View Settings About

Device Explorer

Address | Val

Addre

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

Example

script

```
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

Name Example

Label

Speed x1

x 0 | x 0.5 | x 1 | x 2 | x 5

Trigger

False

Start State | Full view | End State

Default Duration 0.00.10.000

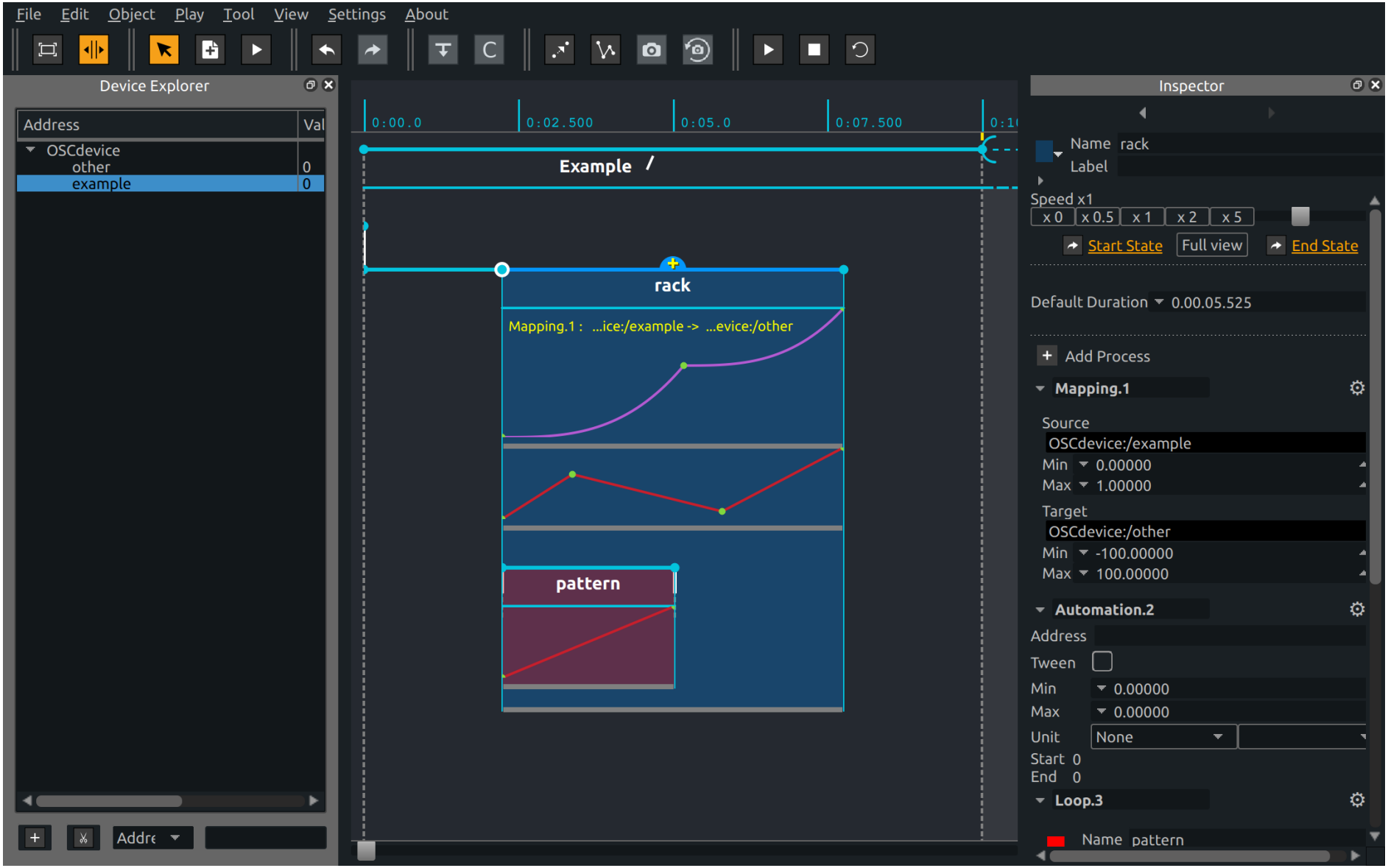
Min Duration 0.00.10.000

Max Duration Infinity

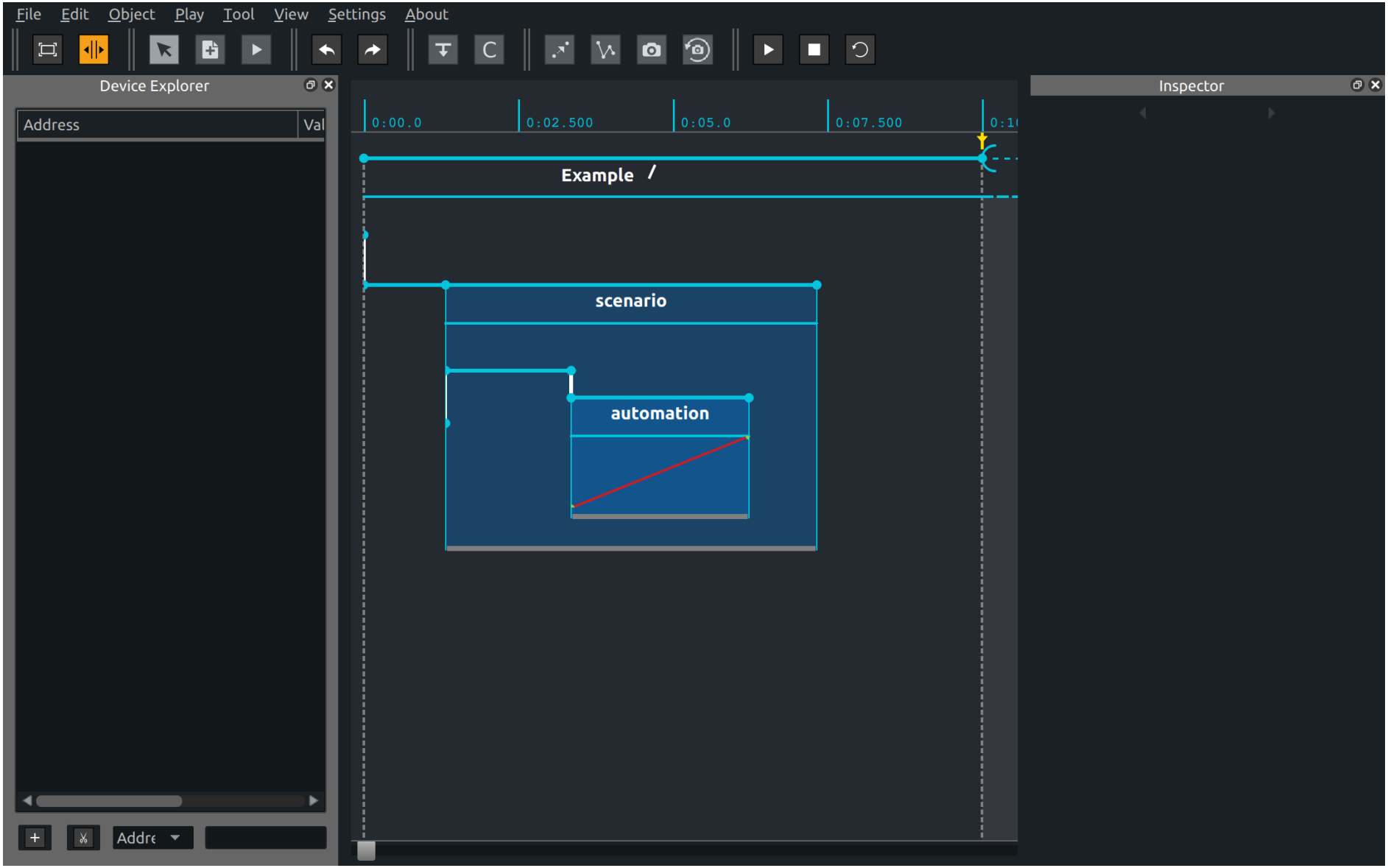
Add Process

Scenario.1

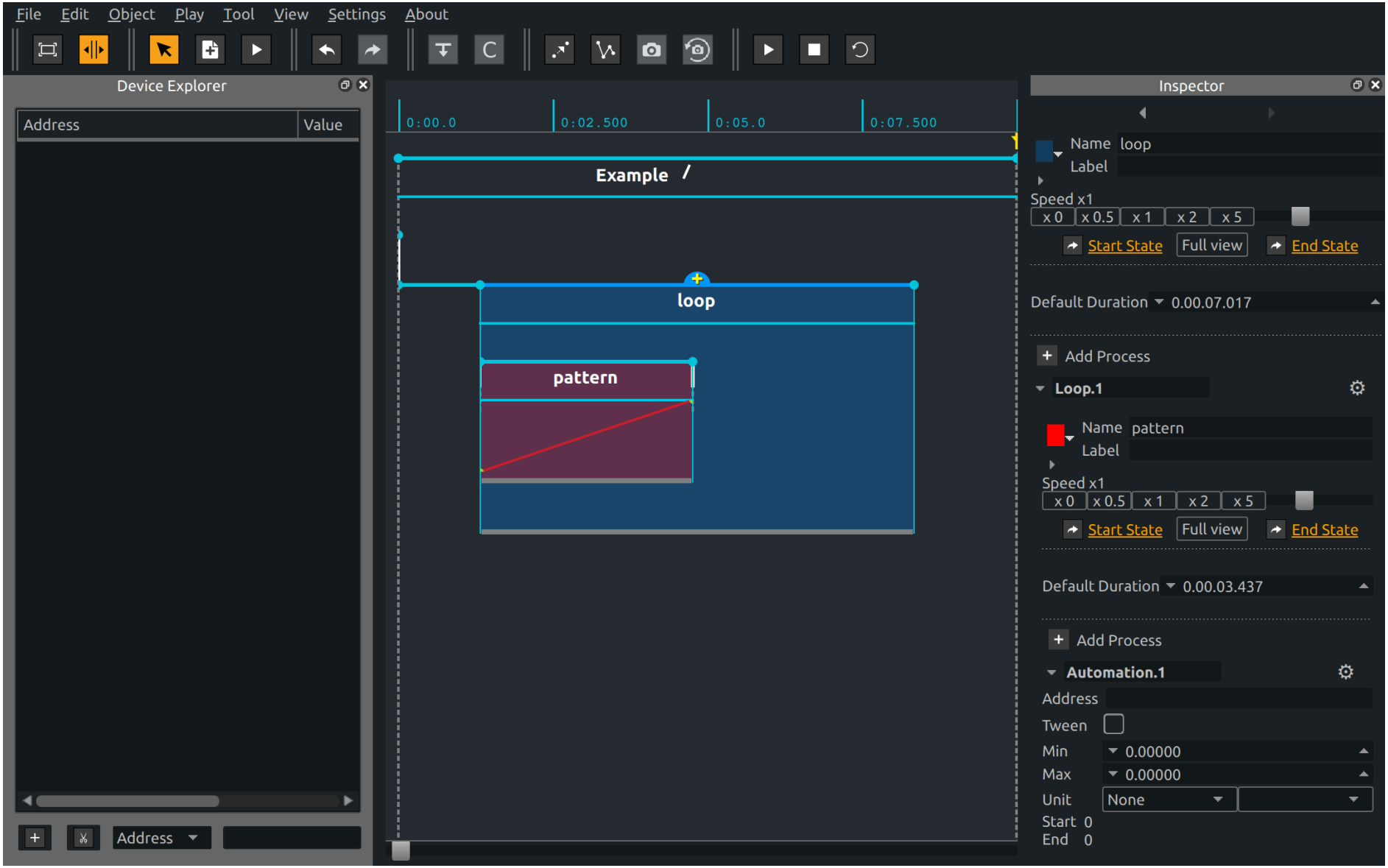
# Racks


File Edit Object Play Tool View Settings About
Device Explorer
Address
Val
OSCdevice
other 0
example 0
Addre
0:00.0
0:02.500
0:05.0
0:07.500
Example /
rack
Mapping.1 : ...ice:/example -> ...evice:/other
pattern
Inspector
Name rack
Label
Speed x1
x 0
x 0.5
x 1
x 2
x 5
Start State
Full view
End State
Default Duration 0.00.05.525
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
Tween
Min 0.00000
Max 0.00000
Unit None
Start 0
End 0
Loop.3
Name pattern

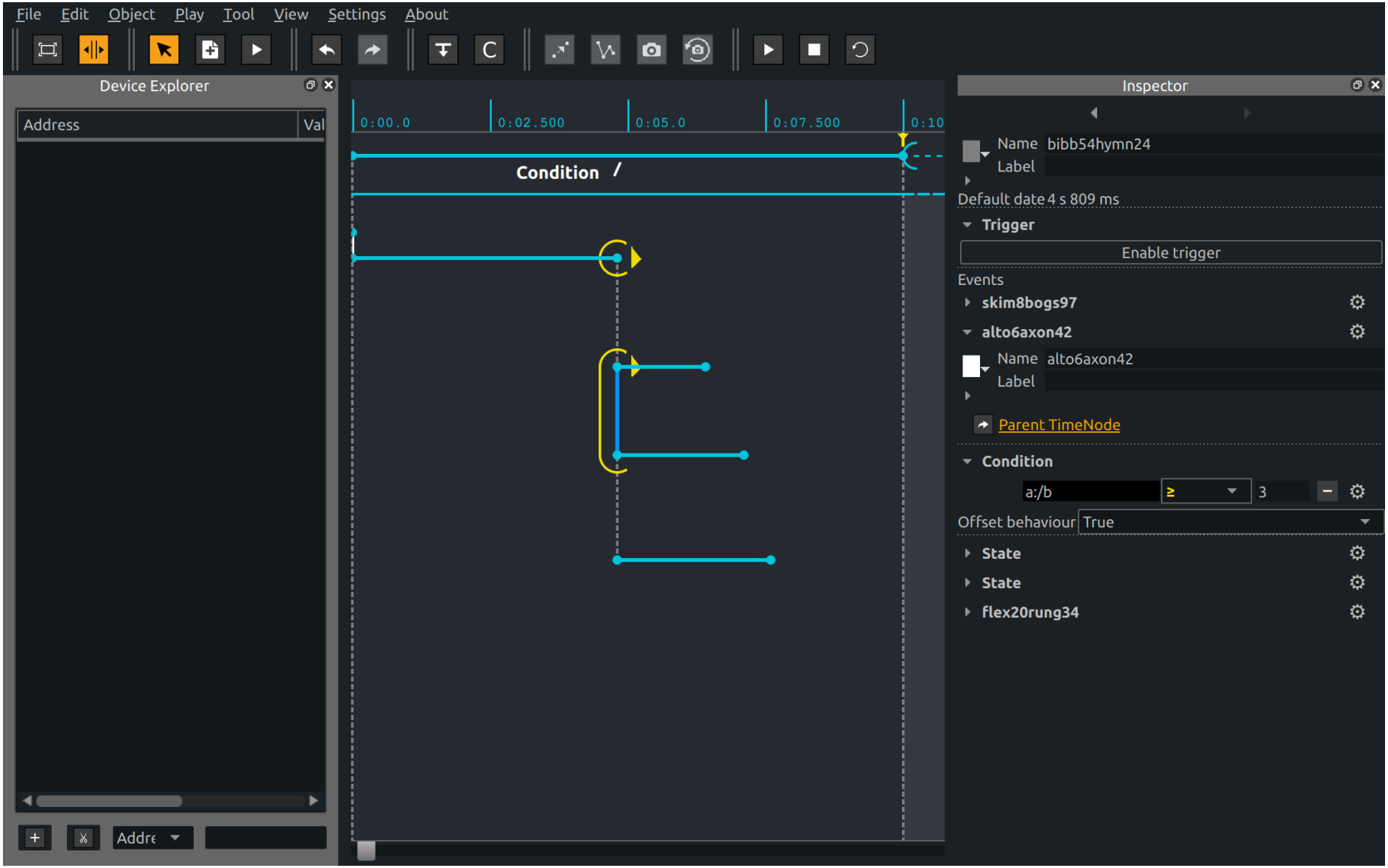
# Hierarchy


File
Edit
Object
Play
Tool
View
Settings
About
Device Explorer
Address
Val
Addre
0:00.0
0:02.500
0:05.0
0:07.500
Example
scenario
automation
Inspector

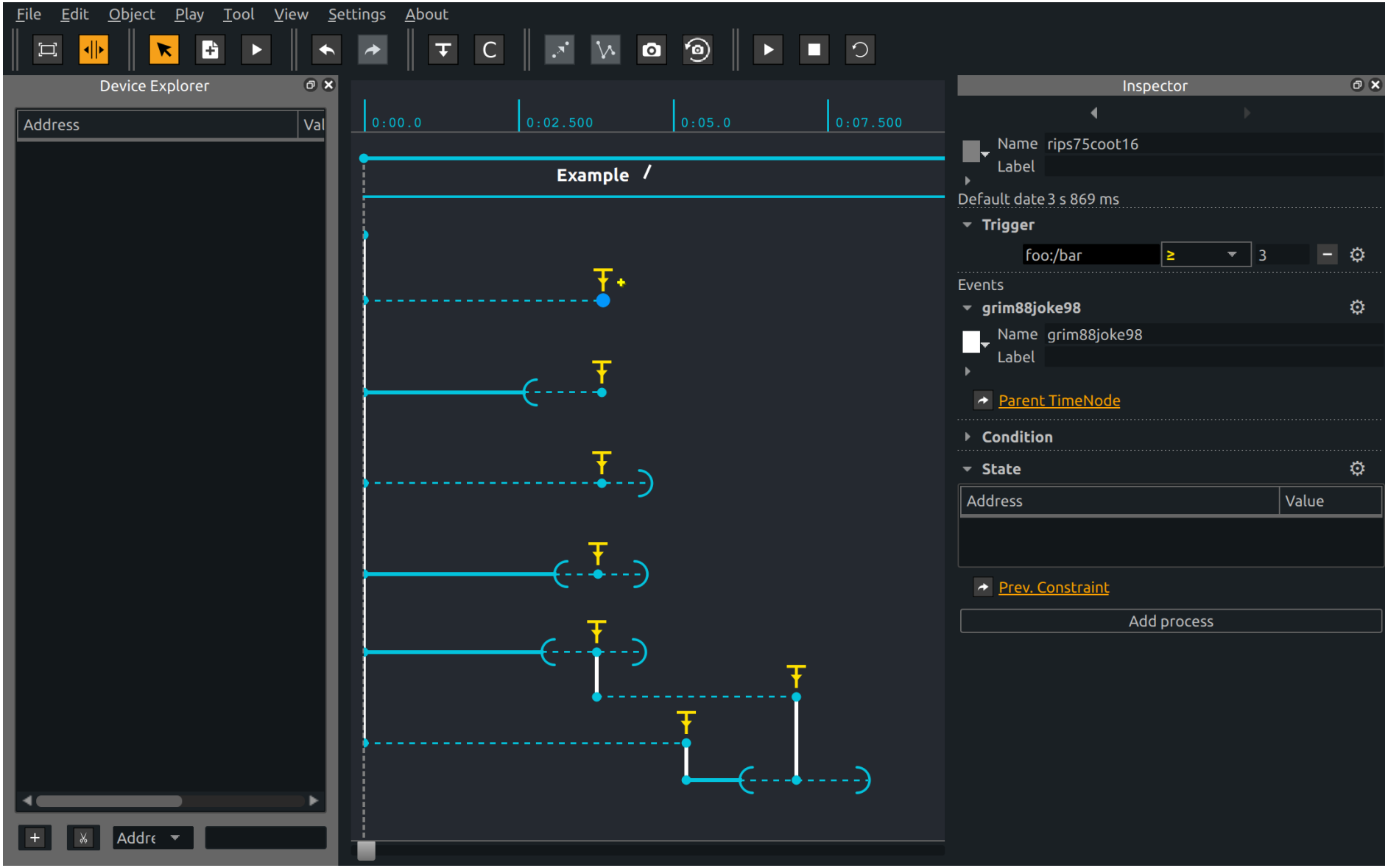
# Loops


File
Edit
Object
Play
Tool
View
Settings
About
Device Explorer
Address
Value
Address
0:00.0
0:02.500
0:05.0
0:07.500
Example
loop
pattern
Inspector
Name loop
Label
Speed x1
x 0
x 0.5
x 1
x 2
x 5
Start State
Full view
End State
Default Duration 0.00.07.017
Add Process
Loop.1
Name pattern
Label
Speed x1
x 0
x 0.5
x 1
x 2
x 5
Start State
Full view
End State
Default Duration 0.00.03.437
Add Process
Automation.1
Address
Tween
Min 0.00000
Max 0.00000
Unit None
Start 0
End 0

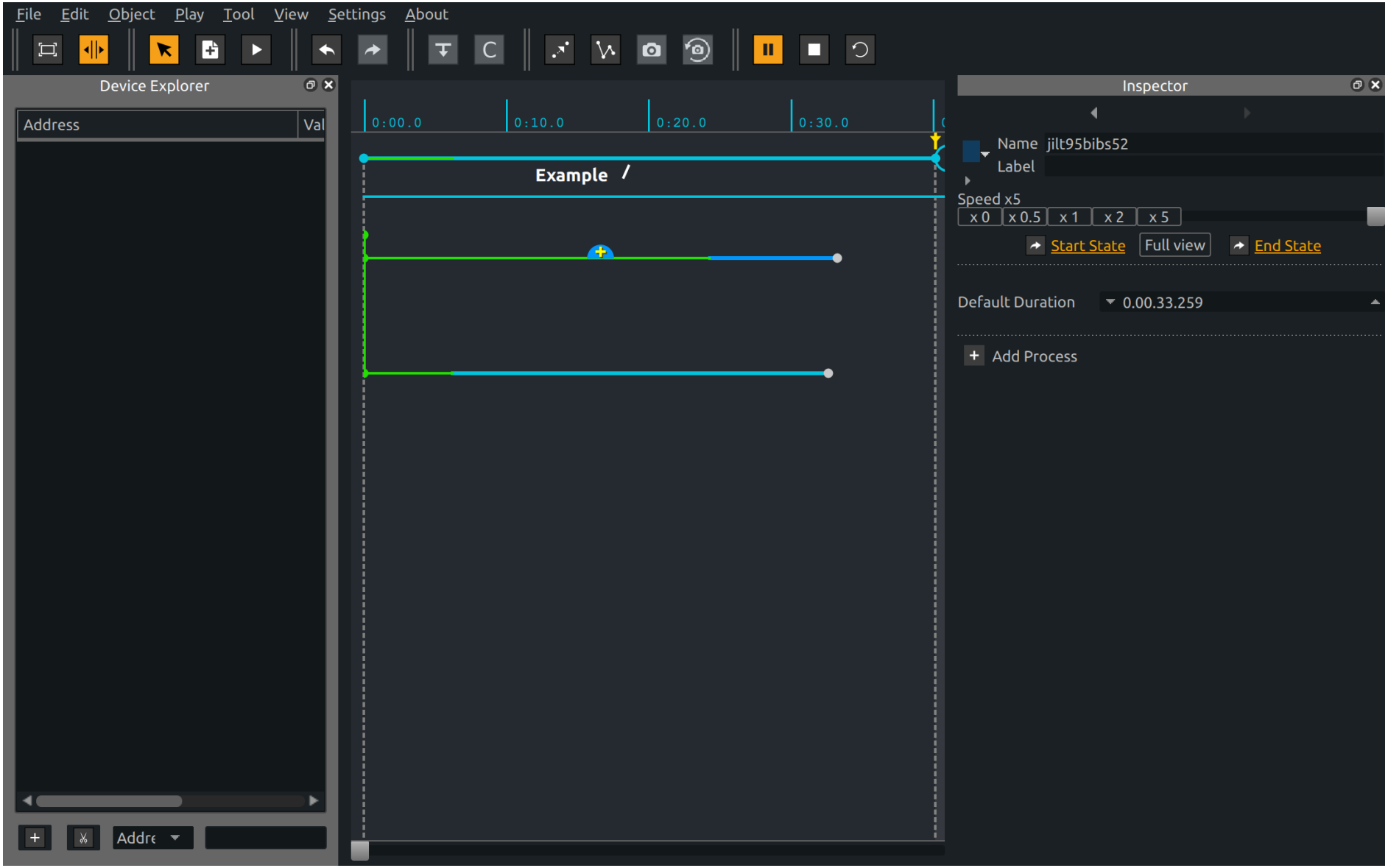
# Interactivity: conditions



# Interactivity: trigger points



# Interactivity: execution speed



# Working with external devices

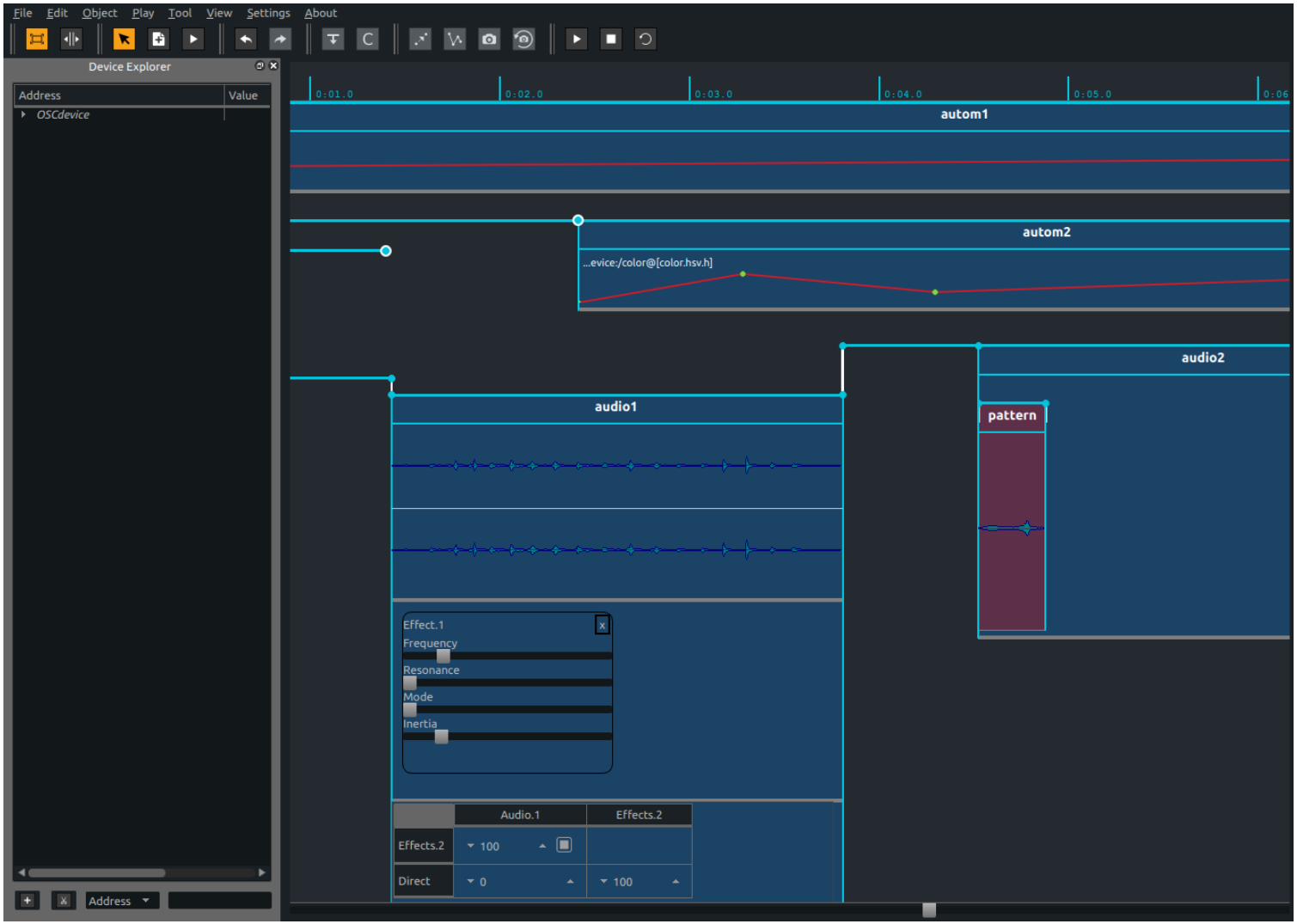
- Manual entry
- Loading
- Learning
- Automatic discovery

# 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: put your score in an Android app, a Pd patch, ...
- Network edition, execution and control, of both editors and players.
- Full-fledged audiograph.
- Ongoing work on UI.
- Need to test live edition more.

# Workshop

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