ToscA

Ircam / Centre Pompidou
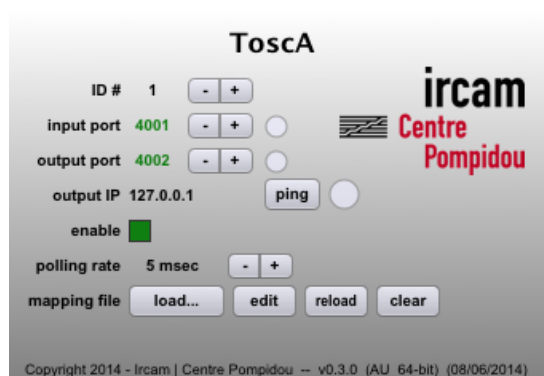
User Manual

ircam
Centre
Pompidou

IRCAM – CNRS UMR STMS
1 place Igor-Stravinksy,
75004, Paris, France
http://www.ircam.fr

Updated : November 18, 2014, Thibaut Carpentier

1 Description

ToscA stands for Thibaut's OpenSoundControl Automation.

ToscA plugin can be inserted into a DAW (Digital Audio Workstation) in order to send/receive parameters' automation. It is based on the OSC (Open Sound Control) protocol (http://www.opensoundcontrol.org). The plugin does not process any audio (the audio signals are just bypassed actually).

- "ID" : this is the ID of the current instance of the plugin. This determines the OSC pattern of incoming/outgoing OSC messages. (cf Section 4)

- "input port" : this is the (UDP) port for incoming OSC messages.

- "output port" : this is the (UDP) port for sending OSC messages.

- "output IP" : this is the IP address of the destination of the outgoing OSC messages. For local communication, you could use "127.0.0.1" or "localhost".

- "enable" : this toggle will enable or disable the OSC send/receive.

- "polling rate" : this is the refresh rate of a background thread polling for incoming/outgoing messages.

- "mapping file" : this is a textual mapping file for parameters' names. (cf Section 3)

Multiple instances of the plugin can live in parallel into the DAW. The ID number is displayed in red when there is a conflict with another instance of the plugin having the same ID. It is displayed in black otherwise.

The input port number is displayed in green when the plugin is properly bound to the port. It is displayed in red if something went wrong. There is also a green led that blinks when OSC messages are received.

The output port number is displayed in green when the plugin is properly connected to the destination (IP and port). It is displayed in red if something went wrong. There is also a green led that blinks when OSC messages are sent.

Please note that all plugin instances share the same input port, output port and output IP address, and mapping file. If you change the input port of a given instance, it will be changed accordingly in all other instances.

All these attributes (ID, input port, output port, output IP, enable toggle, path to the mapping file) are saved with the plugin state i.e. they are stored with your DAW session. And they are automatically recalled when the session is restored.

2 Installation

Copy the plugin bundle into your "plugin folder".

- "ToscA.component" is the AudioUnits version. You can copy it into : "~/Library/Audio/Plug-Ins/Components" or "/Library/Audio/Plug-Ins/Components"

- "ToscA.vst" is the VST version. You can copy it into : "~/Library/Audio/Plug-Ins/VST" or "/Library/Audio/Plug-Ins/VST"

- "ToscA.vst3" is the VST3 version. You can copy it into : "~/Library/Audio/Plug-Ins/VST" or "/Library/Audio/Plug-Ins/VST"

- "ToscA.aaxplugin" is the AAX version. You can copy it into : "/Library/Application Support/Avid/Audio/Plug-Ins"

3 OSC mapping

- The plugin exposes 32 parameters to automation. The parameters' names and range can be tweaked thanks to mapping file.

- If no mapping file is provided, *ToscA* will use the default settings, i.e. the parameters' names will be "param1", "param2", "param3", etc, and their range will be [0.0 - 1.0].

- Mapping file should be a XML file with ".xml" extension.

- The same mapping file is used by all instances of the *ToscA* plugin.

Below is an example of a mapping file, setting up two parameters called "azim" and "elev". The range of each parameter is described by its "min" and "max" attributes. In the host DAW, the automation parameters are always stored in the range [0.0 - 1.0]; however *ToscA* plugin applies a scaling of the incoming/outgoing OSC messages such that "azim" parameter is scaled to [-180. 180.] and "elev" to [-90. 90.]. Currently only linear scaling is implemented.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<tosca version="0.4">
    <parameter index="1" name="azim" min="-180" max="180" scaling="linear"/>
```

```
    <parameter index="2" name="elev" min="-90" max="90" scaling="linear"/>
</tosca>
```

The remaining parameters use their default values i.e. the next parameter is named "param3" with default range of [0.0 - 1.0].

4 OSC messages

- The OSC mapping is used for determining the pattern of the OSC input/output messages. For instance the messages for the "y" parameter of the plugin with ID #3 are : "/3/y 0.0" where "0.0" is the current value of the "y" parameter.

- *ToscA* can be used for reading/writing automation from a remote controller. It can also be used for "touching" an automation track from a remote controller. To do so, the remote controller must notify the DAW host that the parameter is being touched. The OSC syntax for touching the "y" of the instance with ID #3 is : "/3/y touch 1" for beginning the touch. "/3/y touch 0" for ending the touch.

5 Setup

- Install the plugin.

- Create your mapping file with a ".xml" extension.

- Make sure the plugin folder is properly set in your DAW search path.

- Insert the plugin onto an audio track. The track should be mono or stereo.

- Configure the input/output port and address.

- Load your mapping file.

- Edit the automation tracks and start the playback engine.

- Send or receive OSC data in a remote OSC-enabled application or device.

6 Miscellaneous informations / Known bugs

- In Digital Performer, you must disable the "Pre-gen mode when possible" option in the "Plug-in real-time preferences" menu of Digital Performer.

- In Apple Logic Pro : you should have some audio data on the track(s) otherwise the automation is treated in economy mode.
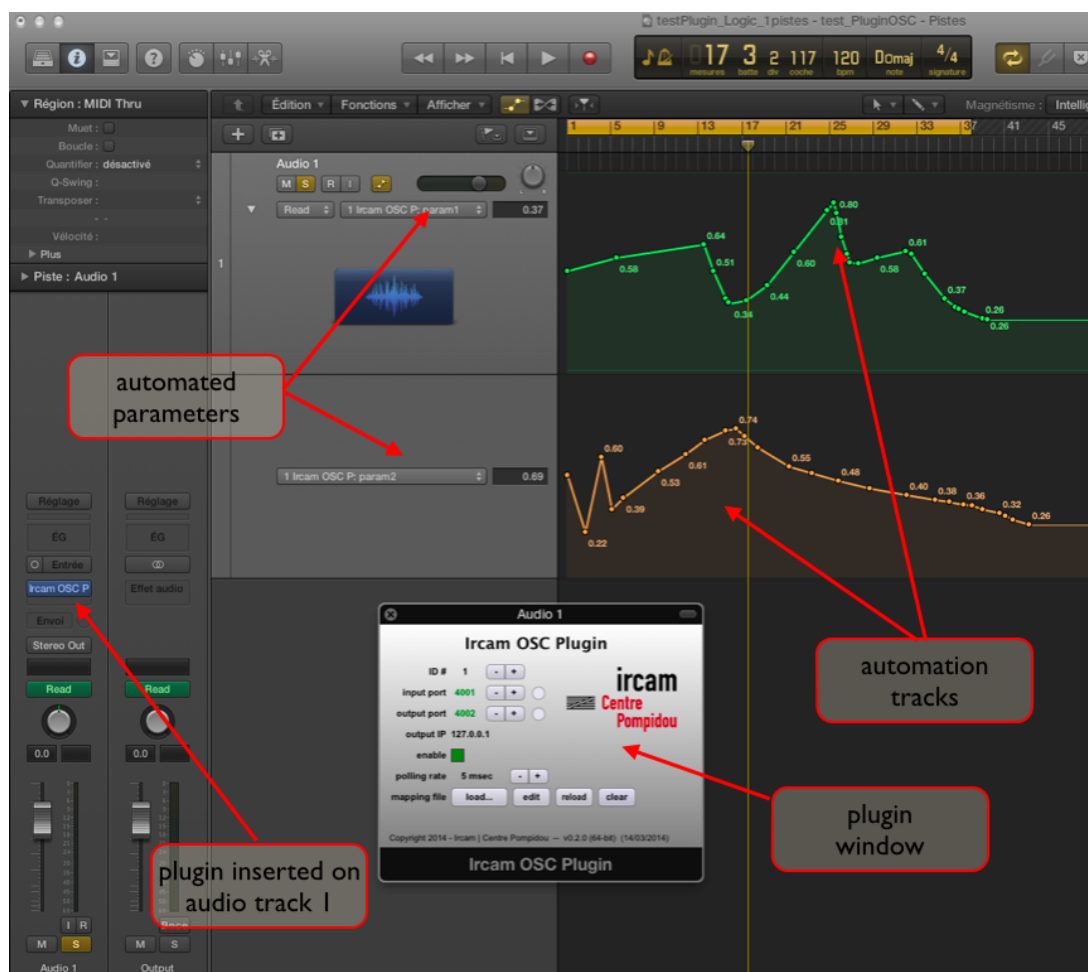
- In Ableton Live VST : the remote automation doesnt work. Use AU instead.

Figure 1: Example in Apple Logic