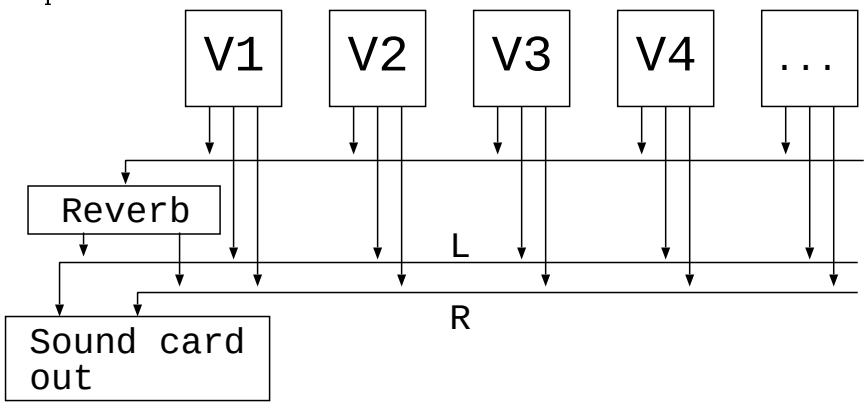
**Iiwusynth <=> LADSPA interface**
Markus Nentwig, 2002
*"Some call it a Manual"*

## Summary

This is a short description of the part of Iiwusynth, which deals with LADSPA plugins[1].
The extension to Iiwusynth allows to route Iiwusynth's output signal through one or more LADSPA plugins. There is no fundamental limit to the number of plugins and the complexity of the resulting effect processor, as long as the signal flow graph includes no loops ('the signal goes only in one direction').

## Signal routing, normal Iiwusynth

When Iiwusynth is started up, the LADSPA extension is disabled (bypassed). It does not require any computation power. The signal is routed as shown in the picture:

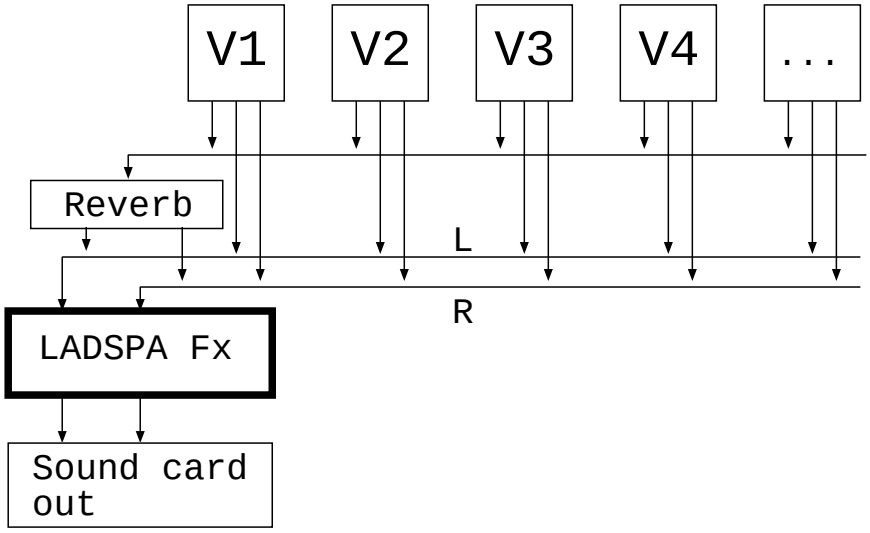



Note that the reverb unit is completely unrelated to the LADSPA plugin link discussed here.

[1] at the time of writing, this is a proposal, which is not included in the CVS repository

## Signal routing, with LADSPA link

When the LADSPA Fx unit is working, the signal is routed as follows:




The box labeled 'LADSPA Fx' contains one or more LADSPA plugins.
In this case, the built-in reverb may not be too useful, since it is in the wrong place (usually a reverb belongs at the end of the signal chain). It is recommended to disable the built-in reverb and use the 'Freeverb' plugin as a last effect in the LADSPA 'box'.

## Inside the 'LADSPA box'

The LADSPA Fx section is wired up using *plugins* (LADSPA) and *nodes*[2]. A plugin has a number of inputs and outputs, here referred to as *ports*.
By default, the Fx section contains four special nodes:

- **Master_L_Synth**
- **Master_R_Synth**: Feed the signal into the Fx unit
- **Master_L_Out**
- **Master_R_Out**: Pass the signal to the soundcard

### Connection rules

- A plugin port connects to a node, in both directions.
- A node *cannot* connect to a node

[2]Internally, a node is a memory location for a real number (control node) or a buffer for a piece of audio data (audio node).

- A plugin port *cannot* connect to a plugin port - that's what nodes are good for...
- *All ports* of all plugins, audio and control, must be connected. Use a *dummy node* for 'open' inputs.
- Signal loops are not allowed

### Creating user nodes

Specifying an unknown node name creates a new node. Its type (audio or control) depends on the connected plugin port. Although any name is accepted, nodes should be named in the same fashion as variables in 'C' (First character a letter, no +-*/ etc).

### Constant control nodes

Control inputs need nodes, which give a constant value. To create such a node, use a # character in front of a number (for example "#0" or "#3.4".[3]

### Dummy node

The flow graph check will issue a warning, if a node has no input. This warning can be prevented by starting the node name with an underscore '_'.

## LADSPA related commands

The following sequence of commands is used to configure the effect section (in the given order):

- **ladspa_clear**
- **ladspa_add**
- **ladspa_start**

### ladspa_clear

This command resets the state of the Fx unit to default (restarting iiwusynth does the same). All plugins are released, the LADSPA Fx section is bypassed.

[3]Internal note: If several inputs use exactly the same constant, they will refer to the same node.

**ladspa_add**

Adds exactly one plugin and an arbitrary number of nodes.
Parameters:

- **1:**Plugin library name (aw.so)
- **2:**Plugin identifier (alienwah_stereo)
  Note: A library may contain one or more plugins
- **3:**Plugin port name (Input1)
  Note that any port (but not node) name can be abbreviated by using only the first characters[4].
- **4:** ← or → (arrows consisting of the less-than or greater-than character and a dash).
  ← feeds data *into* the plugin port, → feeds data *out of* the plugin port (into the node).
- **5:** Node name, for example Master_L_Out
- **6:** Second port: Similar to **3**
- **7:** Similar to **4**
- **8:** Similar to **5**

etc.

**ladspa_run**

When the effect section is programmed using ladspa_add, use ladspa_run to load the 'program' into the Fx unit. Now libraries are loaded, plugins are instantiated and connected. The flowgraph is checked.
When no errors are found, then the Fx section is patched into the signal path. It can be bypassed with ladspa_clear, which also starts a new programming cycle.

## Examples

The following examples show, how to wire up some plugins found on the internet. In general, use 'analyzeplugin ./libraryname' to find out details about a plugin. The program comes with the LADSPA SDK (which is not needed for running Iiwusynth, only for compiling).
At the time of writing, Iiwusynth has no 'batch' command to read textfiles. The examples can be copied into iiwusynth using the middle mouse button. Alternatively,

[4]This was necessary, because port names may contain spaces, which Iiwusynth cannot handle. The parser will complain, if ambiguities are detected.

```
tail -f ExampleScript | iiwusynth MySoundFont
```

will do the job[5].
The library files (*.so) are assumed to be in the current working directory.

### Example: Alienwah

The script has been broken into several lines to make it better readable. Please note, that everything between ladspa_add and (not including) ladspa_start belongs into one line! (And remove the backslashes)
The script connects the 'Alienwah' plugin (a single stereo plugin).

```
ladspa_clear
ladspa_add ./aw.so alienwah_stereo \
     Input1 <- Master_L_Synth      \
     Input2 <- Master_R_Synth      \
     Output1 -> Master_L_Out       \
     Output2 -> Master_R_Out       \
     Frequency <- #0.6             \
     Initial <- #0                 \
     Feedback <- #0.5              \
     Delay <- #20
ladspa_start
```

### Example: Single parametric Eq

Again: Broken into several lines.
This script only uses the left channel, the right channel is left open.

```
ladspa_clear
ladspa_add ./single_para_1203.so singlePara \
       Gain <- #30                          \
       Freq <- #200                         \
       Bandw <- #0.5                        \
       Input <- Master_L_Synth              \
       Output -> Master_L_Out
ladspa_start
```

### Example: Alienwah and high pass filters

This script routes the signal through a (stereo) 'alienwah', and then two (mono) high pass filters. The intermediate nodes are called 'Node1' and 'Node2'. Remember to remove the linebreaks :-)

```
ladspa_clear
ladspa_add ./aw.so alienwah_stereo
       Input1 <- Master_L_Synth  \
```

[5] But no further input is possible except Ctrl-C

```
       Input2 <- Master_R_Synth  \
       Output1 -> Node1          \
       Output2 -> Node2          \
       Frequency <- #0.6         \
       Initial <- #0             \
       Feedback <- #0.5          \
       Delay <- #20
ladspa_add ./filter.so hpf       \
       Input <- Node1            \
       Output -> Master_L_Out    \
       Cutoff <- #1000
ladspa_add ./filter.so hpf       \
       Input <- Node2            \
       Output -> Master_R_Out    \
       Cutoff <- #1000
ladspa_start
```

## Further planning

### Realtime controller

A set of predefined nodes MIDI000 - MIDI127, MIDIPITCH, MIDIPROG00-MIDIPROG15, MIDIBANK etc allow real-time control of the Fx section via the MIDI controllers going through Iiwusynth.
Implementation should require no big effort.

## Individual channel outputs

A MIDI channel can be cut off from the standard Pan / Reverb / Chorus routing, and is instead sent to an individual output (INDCHAN00-INDCHAN15). With this feature, an individual insert effect can be used on each voice.

## Additional audio inputs and outputs

In principle, the Fx unit can be extended to an arbitrary number of audio outputs. Also, inputs from a soundcard could be routed into the Fx unit (for Vocoder, to connect external instruments,...)

## Change history

- 02-03-28 Initial version M. Nentwig
  Proposal, for discussion on the Iiwusynth mailing list