University of Applied Sciences and Arts Northwestern Switzerla
Academy of Music

Musik Akademie Basel

OSPW 2.0 - An Open Source Linux-based DSP Server for Audio Applications

Clemens Fiechter, Research and Development
School of Music, FHNW

Thomas Resch, Research and Development
School of Music, FHNW

Correspondence: Clemens Fiechter, clemens.fiechter@students.fhnw.ch

Introduction

The Open Signal Processing Workstation 2.0 is a Linux-based open software platform, designed for rapid prototyping and the development of digital signal processing audio algorithms and corresponding user interfaces. Since audio interface and computer hardware can be chosen almost completely freely, the system can be easily integrated into any existing audio network and studio environment. Besides the necessary hardware components, OSPW 2.0 consists of the graphical programming environment Pure Data (Pd) for the signal processing, a script for the start-up procedure and initial configuration, and a webserver which generates browserbased UIs for an arbitrary number of remote clients automatically. All connected UI clients are synchronized among each other. This enables the simultaneous operation of applications by multiple users.

Overview

The OSPW software consists of two main parts: the audio backend and the OSPW server. The backend is based on a plain Pd Vanilla installation. The server is a Node.js application which enables the user to control and interact with the running Pd instance. Once the design for an algorithm is completed, the user can transfer the patch to the server, where it is analysed for automatic UI generation and executed. Any device with a browser running in the same network can be used as remote control for the loaded Pd patch.

Automatic UI Generation

For a parameter to appear in the UI, a matching Open Sound Control (OSC) string must be included in the Pd patch. This is done by placing a comment containing the string somewhere in the patch as shown in figure 3. The syntax for the string is:

`/ospw/x/y/widgetType/parameterName/initValue.`

- The string has to start with the keyword *ospw*.
- *x* and *y* are grid coordinates within a scalable, symmetric grid
- *widgetType* defines the generated interface object. Possible values are button, toggle, number, dial, hslider, vslider.
- *parameterName* can be chosen freely and results in the rendered widget label.
- *initValue* initializes the interface object with the entered value.

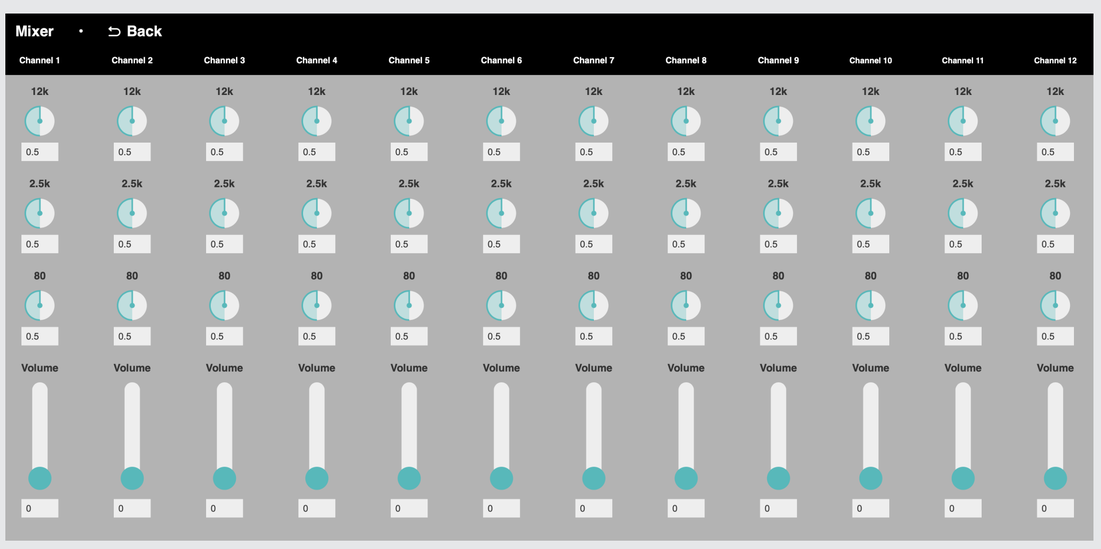

Figure 1: A mixer demo application for OSPW.

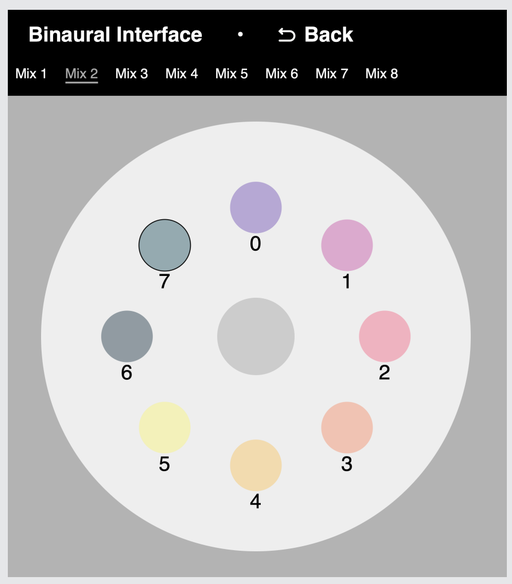

Figure 2: A custom interface for the OSPW binaural mixer demo

```
netreceive -u -b 57120
oscparse
route list
route ospw
        route param1
        0
        s param1
        /ospw/0/0/dial/param1/0.5
route param2
0
s param2
/ospw/0/3/hslider/param2/0.5
```

Figure 3: Pd Patch with two OSPW parameters.

Discussion

OSPW is an easy-to-use open source DSP platform which can be built with off-the-shelf hardware components. The free choice of sound card (as long as it is ALSA compatible) makes the integration in any existing audio environment possible. By using Pd as audio backend, the signal processing can be implemented both in the C programming language and graphically. The graphical access also enables "intermediate" programmers and artists in the field of media technology to use the system. OSPW enables intuitive, network-based access to Pd. Finished patches are simply pushed into the designated folder and can then be selected and operated via remote client. Currently only the most important UI elements (dials, sliders, labels and number boxes) are implemented for automatic interface generation. To ensure intuitive handling for more complex DSP algorithms, future updates should also include more sophisticated UI elements such as multisliders or frequency domain editors.

OSPW Git-Repository: www.github.com/cfiechter/OSPW