MOD – An LV2 host and processor at your feet

Gianfranco Ceccolini
The MOD team
Rua Júlio Rebollo Perez, 488 – cj 3
São Paulo, Brasil
contato@portalmod.com

Leonardo Germani
The MOD team
Rua Júlio Rebollo Perez, 488 – cj 3
São Paulo, Brasil
leogermani@hacklab.com.br

Abstract

MOD is a linux-based LV2 [1] plugin processor and controller. It is accessed by musicians via bluetooth to setup their pedalboards by making internal digital connections between plugins and audio inputs/outputs.

After a pedalboard set is saved, it can be shared with other users at the MOD Social network.

The software components are Open Source, which means you can also run it on any computer using GNU/Linux operating system, not only on MOD hardware.

The presentation aims to introduce the device to the community and discuss how its development may interact with plugins and the LV2 standard's development.

Keywords
LV2, host, hardware, jack, plugins

1 Introduction

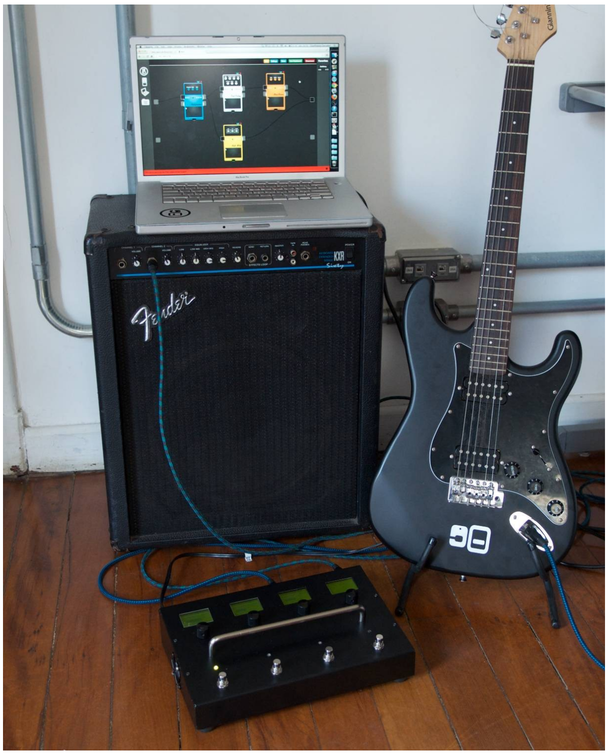
We have been developing MOD for more than three years. From the first draft to a usable prototype, now our first end-user ready units are being produced and we would love to show them at LAC, since MOD is only possible thanks to the work of many people we expect to meet there.

MOD is a LV2 host device at your feet, that can be used on a live performance disconnected from a computer. It has a powerful and intuitive user interface to create setups via tablet or computer and it supports any LV2 plugin.

1.1 Motivation

As old-time users of Jack, LADSPA and all the GNU/Linux pro-audio stack, we always missed an easy, plug-and-play way to use the effects and tools available in our GNU/Linux boxes. MOD was born with this in mind, to provide an accesible and user-friendly device that supports the tools GNU/Linux users are used to and also accesible to more traditional musicians.

2 Overview



2.1 How it works

In a pinch, you plug your instrument and amp to the MOD and play. It processes the pedalboards you have saved in it.

To create or change a pedalboard you connect to the MOD using a tablet or a computer via bluetooth and access it using a browser. You will be presented with a dashboard where you can drag and drop plugins and make connections. At the left side on the screen you will see two fixed sound sources that represents the physical audio inputs, where you will plug in your instruments, and two audio outputs on the right, representing the physical audio outputs.

You can make as many connections as you like from each sound source, being it the physical audio input or a plugin output.

Each plugin is represented by a pedal. Click on it and you can tweak all its parameters. You can also address a plugin parameter to a physical knob, foot switch or expression pedal.

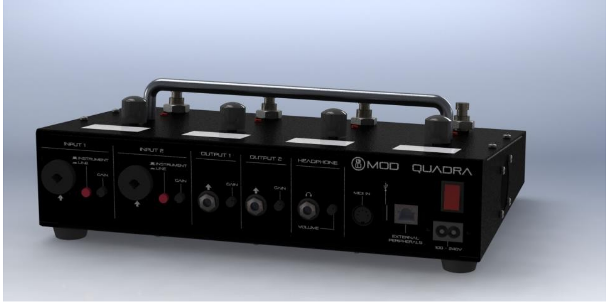
MOD comes with many plugins installed, but you can search and install more plugins on the fly, browsing the MOD repository (see 3.1). Plugins are organized into categories and rated by the users.

Once you are done with your setup, we call it a pedalboard, you can name it and save it. When you have many pedalboards you can organize it in banks so you can easily access it from the MOD menu when you are not using your tablet/computer.

MOD can be used in a standalone mode, which mean you don't need to carry your computer to the gig. You can easily browse the banks and pedalboards you organized from the menu, or you can even address one or two foot-switches to go to the next/previous pedalboard.

Back to your computer, you can now share your pedalboards with other MOD users and exchange ideas and tones.

2.2 Hardware



The MOD is enclosed in a robust folded sheet-metal steel enclosure with spill-proof top face.

Inside it there’s an embedded Intel Atom dual core running at 1.8GHz with an SSD drive for storage.

It sports a 2 x 2 24bit / 48kHz audio interface with very low noise (>100dB). The input section consists of two pre-amped XLR + P10 combo jacks with selectable line/instrument level and adjustable gain. The output section consists of two balanced P10 connectors with adjustable gain plus a Headphone jack.

The controller is composed of four incremental encoders with LCD displays for parameter changing and visualization plus four heavy duty foot-switches. It is equipped with an ARM processor for parameters treatment and displaying.

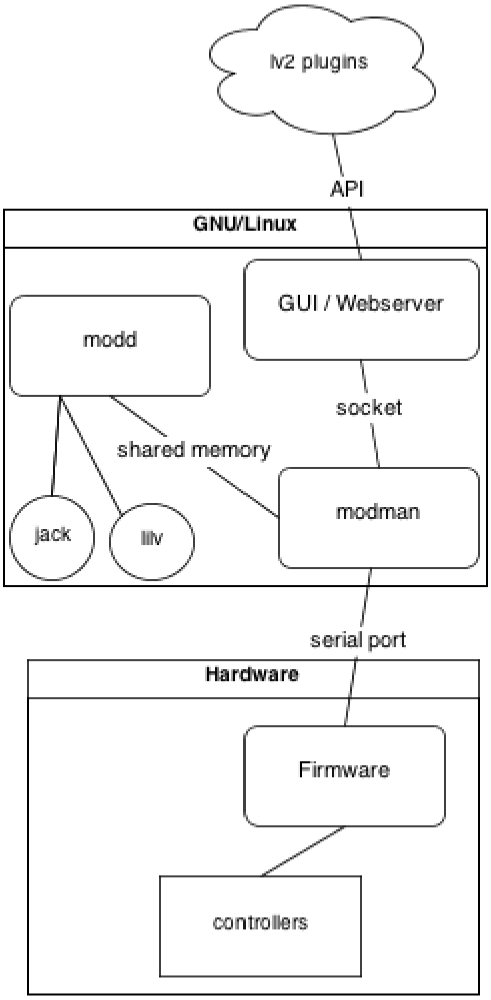
There’s also an internal full range power supply - 100V-240V - to deliver adequate power to all internal components.

2.3 Software

MOD software is divided into an LV2 host, the firmware, a web-based GUI and the manager - the software responsible for the communication between all the other pieces.

All software, with the exception of the firmware, is Free Software and licensed under the GPLv3 license [2].

3 Software Components



Picture 1: Software components

3.1 Operating System

MOD runs a customized ArchLinux [3] based distro, currently running the kernel 3.6.11 tuned for real-time and some other performance patches.

It uses systemd and all daemons rely on it for the start/stop/dependency/keep alive actions. The software is packaged as pacman [4] packages and are published on MOD arch repository which is used to manage the dependencies between packages and to keep the OS stable/frozen.

The communication between the hardware and the OS/software is done through a real serial port.

3.2 LV2 host / Jack Client - modd

The host is called modd and is really simple, so far it supports all the basic / core lv2 functionalities and also the following lv2 features/extensions:

- uri-map
- uridmap
- MIDI events (event and atom)

Support to the time extension is being developed, but at first it will not have transport support, being frame and tempo aware only, so the user can specify a BPM (via knobs or using the tap-tempo foot-switch) and syncing time/bpm/hz parameters.

Modd is mostly written in C using David Robillard's lilv [5] library and is our main jack [6] client. It runs with real-time priority.

3.3 MOD Manager - modman

The manager is responsible for handling the communication between the GUI, the Controller Firmware, the Host and the OS. It also have some core responsibilities for example when saving user data, turning on/off the bluetooth and other system related tasks.

It's also written in C and it communicates with modd using a shared struct (shared memory). It uses a serial port to send messages to the firmware and a socket to the Python [7] based web server and JavaScript [8] GUI. All the communication uses a simple protocol specified by MOD’s team. The protocol describes all the functionalities from adding/removing plugins to assigning control ports to hardware encoders or setting/reading control port values.

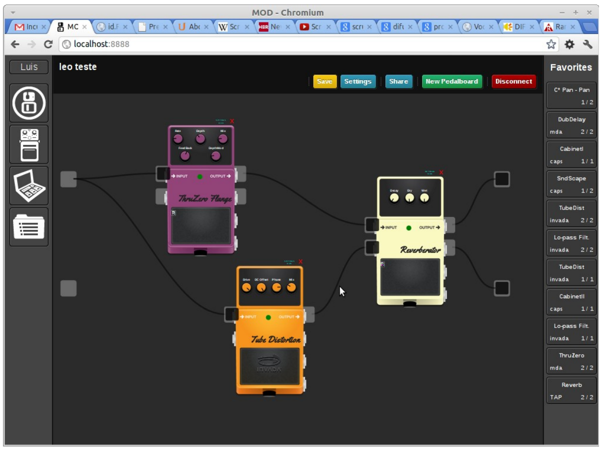
3.4 Bluetooth

The bluetooth host is written in Python using Dbus [9] and Bluez [10]. It simply pairs, authorizes and waits for new bluetooth network connections, whenever a connection is created (using the bnep0 interface) an udev [11] rule is matched and executes a script that will set the IP address for the interface and run dnsmasq [12] on that interface. dnsmasq will then provide a dhcp and dns server for clients connected through bluetooth, that way if the user's computer or tablet tries to resolve the name "mod" it will resolve to the device's IP address.

3.5 Web GUI

The GUI is mostly written in JavaScript and Python using the Tornado framework. It uses the commands described by the MOD protocol to communicate with the manager (modman) through a socket, modman will then route the messages to the other components.

The MOD device is never connected directly to the Internet, the GUI JavaScript code will detect if the user's computer/tablet is connected and, if it is, it will be able to access the cloud and download new plugins, upgrade older plugins or upgrade the whole system, including the firmware.



Picture 2: The Beta GUI

3.6 Firmware

The firmware that drives the controller manages all it’s hardware : displays, encoders and foot-switches.

It is also responsible for:

- managing the exchange of data between the actuators and modman;
- dealing with controlwise parameters properties, such as using logarithmic scales or setting a tap-tempo chronometer when a time-related parameter is addressed to a footswitch;

4 MOD and LV2

4.1 Plugins repository

To make MOD’s users life easier, we are packaging and hosting LV2 plugins in a repository, just as linux distributions do.

We have ported some LADSPA [13] plugins to LV2, such as the CAPS [14] and TAP [15] packages and we will have around sixty plugins packaged on the ongoing release date.

From the user interface users can only install plugins from this repository, but we are already working to setup a “contrib” repository and also the possibility to install any LV2 plugin you have downloaded from anywhere.

4.2 MOD LV2 extension

Any LV2 plugin will work well on MOD. But since we are focusing on user experience, we want each plugin to have its own look.

Part of it is supported by the LV2, but another part of it is specific to MOD. For instance, on the MOD dashboard, each plugin has its icon, usually a pedal icon. It would be nice that the plugin developer could choose the icon he/she wanted. When a parameter is addressed to a physical control, it has a label that appears on the LCD screen, this could be customized too. Also, the configuration GUI, that LV2 allows developer to build, could only be used in MOD if it was written in HTML/javascript.

For those reasons we are extending the LV2 standard, adding some specific features that MOD supports. Nevertheless, we are not touching the standard, and we are supporting any plugin that does not have this extensions with the use of default options.

4.2.1 SDK

In the future we want to release a SDK to help plugin developers adapt their plugin to the MOD extensions. It will give them layout templates and a set of icons to choose from.

5 Conclusions

Being based on open standards we believe MOD can become a much better and versatile multi-effect processor than any other found on the market. We want to improve the social functionalities and support the LV2 standard so new developers get interested and new plugins are developed.

6 Acknowledgments

MOD started because of the LAD and LAU community, because of the softwares and concepts that originated there, so we would like to thank this community of users and developers. We also want to thank the estudiolivre.org community (a Brazilian community focused on the use of free and open-source software for multimedia production).

7 References

[1] LV2, a plugin standard for audio systems http://lv2plug.in/trac/

[2] GPL, the GNU General Public License http://www.gnu.org/licenses/gpl.html

[3] ArchLinux, a simple and lightweight GNU/Linux distribution http://www.archlinux.org

[4] PACMAN, the ArchLinux package manager https://wiki.archlinux.org/index.php/Pacman

[5] LILV, a library to make the use of LV2 plugins as simple as possible http://drobilla.net/software/lilv

[6] JACK Audio Connection Kit http://www.jackaudio.org

[7] Python Programming Language http://www.python.org/

[8] JavaScript, the scripting language of the Web http://www.w3schools.com/js/default.asp

[9] Dbus, a message bus system for applications to talk to one another http://www.freedesktop.org/wiki/Software/dbus

[10] BlueZ, the official Linux bluetooth protocol stack http://www.bluez.org/

[11] UDEV, Linux dynamic device management https://www.kernel.org/pub/linux/utils/kernel/hotplug/udev/udev.html

[12] DNSMASQ, a lightweight DNS forwarder and DHCP server http://www.thekelleys.org.uk/dnsmasq/

[13] LADSPA, Linux Audio Developer's Simple Plugin API http://www.ladspa.org

[14] CAPS, the C* Audio Plugin Suite http://quitte.de/dsp/caps.html

[15] TAP, Tom's Audio Processing plugins http://tap-plugins.sourceforge.net/