

# Creating Audio Plugins

From MOD Wiki

A plugin is a software extension that adds new features to an existing application. Similarly, an audio plugin can add or enhance audio-related functionality to a device (e.g. MOD Duo). Such functionality may include digital signal processing, sound synthesis or MIDI data processing[1].

## Contents

- 1 LV2 standard
- 2 Creating plugins
  - 2.1 From scratch
  - 2.2 Framework / High Level Language
  - 2.3 Visual Programming

## LV2 standard

MOD devices run LV2 plugins internally. This means that if you want to create your own plugin for using with the MOD, you will need to build an LV2 plugin. LV2 is an open-source plugin specification designed for audio applications which is powerful and extensible.

There are many good reasons why LV2 is the best choice as the audio plugin standard for the MOD devices. Most of them gracefully explained in this page of the LV2 website.

Thanks to the extensibility and the bundle concept which LV2 provides we were able to create beautiful HTML/CSS interfaces for the plugins present in the MOD devices.

## Creating plugins

The idea of this page is not to go deep into the LV2 standard but to show the options you have when creating your own plugin, guide you through the appropriate documentation according the method you choose and finally show how to upload and test it on your device.

Bellow are listed the topics that you might need to know in order to create audio plugins for MOD devices.

- Digital signal processing (DSP)
- Programming
- Basic math

Note that LV2 standard is not listed as requirement because you don't need to know it if you are using Max gen~.

There are 3 ways to create an LV2 plugin:

- Coding everything from scratch
- Using a framework or high level language (e.g.: DPF, Faust)

- Using visual programming software (e.g.: Max Gen~)

Regardless the option you choose it's highly recommended that you use Linux (preferable) or Mac OS for development. Not only all MOD team developers use Linux but most of the LV2 developers does as well. In other words this means that almost all documentation you will find assumes that you are using Linux and in the case you need to ask for support it'll be much easier.

## **From scratch**

The first option, coding everything from scratch, is the hardest one if you are new to programming. Here we will need to know C or C++ to code the plugins and also read the LV2 documentation and examples. Although the learning curve is steep (seeing that you'll need to learn programming first) that's a good choice if you want full control of what you're doing and probably it's the best choice to write well performing plugins.

What to learn:

- C/C++ language
- LV2 standard

To learn about the LV2 standard start reading this [blog post <http://harryhaaren.blogspot.com.br/2012/06/writing-lv2-plugins-lv2-overview.html>] by Harry Van Haaren.

[Programming LV2 Plugins Book <http://lv2plug.in/book/>] from David Robillard. It aims to explain the LV2 by using examples instead of API documentation.

TODO: short explanation of TTL files and links to examples and/or documentation.

TODO: development environment setup (docker only?)

## **Framework / High Level Language**

TODO: Juce and DISTRHO Plugin Framework (DPF), Examples: there are plugins done using DPF, need to find out)

TODO: development environment

## **Visual Programming**

TODO: Max/MSP (MAX Gen~) Examples: shiro plugins

TODO: Puredata (Camomile) Examples?

TODO: development environment

[1] [https://en.wikipedia.org/wiki/Audio\\_plugin](https://en.wikipedia.org/wiki/Audio_plugin)

Retrieved from

"[https://wiki.moddevices.com/index.php?title=Creating\\_Audio\\_Plugins&oldid=11622](https://wiki.moddevices.com/index.php?title=Creating_Audio_Plugins&oldid=11622)"

- 
- This page was last edited on 22 April 2019, at 02:12.
  - Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.

