

Developers

From MOD Wiki

The MOD Duo runs a highly-optimized Linux embed system, using JACK as audio server and loading plugins in LV2 format.

Plugin parameters can be controlled via the Duo's built-in knobs and footswitches, via MIDI CC and Pitchbend or via Control Chain devices.

Contents

- 1 Technical information
- 2 Useful links
- 3 mod-plugin-builder
 - 3.1 Quick example plugin
- 4 Deploying plugins
- 5 APIs and Documentation
 - 5.1 Linux
 - 5.2 JACK
 - 5.3 LV2
 - 5.4 Toolchain
 - 5.5 Libraries
- 6 Developer mode
- 7 Unstable plugins
- 8 USB Device Support
- 9 Hardware Specification

Technical information

As of release 1.1 the MOD Duo runs the Linux kernel version 3.4, with real time patch.

ALSA is used for the audio driver. JACK runs inside as audio server backend, with mod-host loading each LV2 plugin as a new (internal) JACK client as mod-ui as the web server.

There's no X11 or any other kind of graphical interface. In fact, X11 libs are not even available.

SSH is open by default.

The CPU is a Cortex A7, running at a constant 912 MHz.

The compiler is GCC 4.9, with the default compilation flags being:

```
-O3 -mcpu=cortex-a7 -mtune=cortex-a7 -mfpv4 -mfloat-abi=hard -mvectorize-with-neon-quad
```

For aggressive optimization, add these:

```
-ffast-math -fprefetch-loop-arrays -funroll-loops -funsafe-loop-optimizations
```

Building binaries for the MOD Duo can be easily done using the mod-plugin-builder.

Apart from programming software in a "traditional" way, it's also possible to generate audio plugins for the MOD from gen~ objects created in Cycling '74 Max. See the Max_gen~ page for more details.

Useful links

These are useful links for general information regarding developing audio plugins and realtime software.

- Real-time audio programming 101: time waits for nothing
- Programming LV2 Plugins

mod-plugin-builder

The mod-plugin-builder is a custom build system that gives developers a similar system to what's available inside the Duo.

We do not recommend the use of a regular Linux armhf system or compiler, it might lead to issues due to mismatching library versions.

If you're running Linux just clone MOD Plugin Builder and follow the instructions.

In summary:

```
┌-----┐
│ $ git clone git://github.com/moddevices/mod-plugin-builder
│ $ cd mod-plugin-builder
│ $ ./bootstrap.sh
└-----┘
```

This process should take at least 1 hour, probably more depending on your CPU.

Assuming your source code is cross-compile friendly (ie, no hardcoded compiler and paths and uses pkg-config to find extra libraries), the compiler setup is as simple as: (adjust as needed)

```
┌-----┐
│ $ source ~/mod-plugin-builder/local.env
│ $ # cmake, waf configure, ./configure, etc
│ $ make
└-----┘
```

The local.env file will setup your Linux compiler environment variables (such as CC, CXX, CFLAGS, etc) to use mod-plugin-builder files.

If everything goes well, the resulting binaries will be ARMv7, MOD Duo compatible.

Quick example plugin

A quick example plugin is available inside mod-plugin-builder in plugins/package/eg-amp-lv2/source/, which works with this cross-compilation setup.

See <https://github.com/moddevices/mod-plugin-builder/tree/master/plugins/package/eg-amp-lv2/source>

Building this example plugin can be done like so:

```
┌-----┐
│ $ source ~/mod-plugin-builder/local.env
│ $ make -C ~/mod-plugin-builder/plugins/package/eg-amp-lv2/source
└-----┘
```

After this the eg-amp.lv2 bundle is ready to be deployed into a MOD unit.

Deploying plugins

You can push an LV2 plugin bundle to the MOD Duo via command-line, using curl, by running this: (adjust as needed)

```
┌-----┐  
│ $ cd ~/mod-plugin-builder/plugins/package/eg-amp-lv2/source/  
│ $ tar cz eg-amp.lv2 | base64 | curl -F 'package=@-' http://192.168.51.1/sdk/install  
└-----┘
```

That's it! Your plugin is now inside the Duo!

To those curious ones, the last command packs the eg-amp.lv2 into a tar.gz file, encodes it as base64 and then sends it to a special URL that the MOD Duo uses to install plugin tarballs.

APIs and Documentation

Linux

Being Linux, developers have access to UNIX and POSIX APIs. This means `<unistd.h>`, `<pthread.h>` and `<stdlib.h>` for example.

For more information see:

- <https://www.kernel.org/doc/man-pages/>
- <http://pubs.opengroup.org/onlinepubs/9699919799/functions/contents.html>

JACK

Also known as Jack Audio Connection Kit, is the audio server running inside the MOD. Developers can create JACK clients to interact with the audio data directly (without being managed by a plugin host).

For more information see:

- <http://jackaudio.org/api/index.html>

NOTE: As of release 1.1, MOD Duo uses JACK2.

LV2

Audio and MIDI plugins in MOD Duo are done in LV2 format. MOD provides custom LV2 extensions (for MOD specific data and HTML interfaces).

For more information see:

- <http://lv2plug.in/pages/developing.html>
- <http://moddevices.com/ns/mod/>
- <http://moddevices.com/ns/modgui/>

Toolchain

As of release 1.1, the MOD Duo uses:

- GCC 4.9
- glibc 2.21
- binutils 2.25

This means support for C11 and C++11.
C++14 or higher is not supported.

Libraries

The following libraries are available inside the Duo:

- acl
- alsa
- armadillo
- audiofile
- boost
- cairo
- dbus
- expat
- fftw/fftwf
- ffmpeg
- flac
- fluidsynth
- fontconfig
- freetype
- gio/giomm
- glib/glibmm
- glibc (v2.21)
- icu
- jack
- libffi
- libgig
- libjpeg/libturbojpeg
- liblo
- libogg
- libpng
- libsamplerate
- libsigc
- libsndfile
- libvorbis
- libxml2
- lilv
- lvtk (v1 and v2)
- mxml
- ncurses
- ne10
- pcre
- pixman
- python (v3.4)
- qt5 (core, network, sql and xml)
- readline
- sdl

- serd
- sord
- sratom
- systemd (v219)
- zita-convolver
- zita-resampler

Developer mode

MOD's web interface has some hidden features, not really useful for users but useful for developers. This includes:

- Buffer size change (128 or 256 frames)
- Install-All button in plugin store
- Network statistics
- Xrun counter

To enable developer mode, open the browser developer console and enter:

```
| enable_dev_mode()
|
```

Starting from v1.3, this setting is saved within the MOD, and developer mode is enabled the next time the page loads.

To disable developer mode, open the browser developer console and enter:

```
| disable_dev_mode()
|
```

Unstable plugins

By default unstable plugins are hidden in the plugin store; however they can be made visible for use within pedalboards:

- For releases 1.5.0 and later, there is an option on the special settings page:
<http://modduo.local/settings> (or <http://192.168.51.1/settings> if you don't have zeroconf working).
- For earlier releases, open the browser development console and enter:

```
| $("#cloud-plugins-stable").parent().show()
|
```

NOTE: Do not report bugs when using unstable plugins.

They have not been curated or tested, so they might corrupt memory or even crash.

Use with caution.

USB Device Support

- Several USB-MIDI devices
- Arduino running MocoLufa firmware (to create midi-controllers)
- PlayStation 3 and 4 Controllers
- Guitar Hero Controller (Red Octane Xbox version)

Hardware Specification

CPU:

- Dual Core ARM A7 1.0GHz
- 4GB Flash Storage, 1GB RAM

AUDIO CODEC:

- Cirrus Logic, 24 bit / 48 kHz AD/DA
- DAC/ADC: 104dB Dynamic Range. -90dB THD+N

CONTROLLER:

- 2 knobs with LCD screen
- 2 foot switches with color LEDs

I/O CONNECTIONS:

- MIDI IN/OUT - Standard DIN 5 pins
- Control Chain - RJ45 connector - for additional controllers
- USB HOST - USB 2.0 Standard-A type, supporting: USB Bluetooth, USB Wifi, USB MIDI
- USB DEVICE - USB 2.0 Standard-B type, supporting: USB Ethernet Connection

Retrieved from "<https://wiki.moddevices.com/index.php?title=Developers&oldid=11563>"

-
- This page was last edited on 9 November 2018, at 22:26.
 - Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.

