



**Linaro
connect**
Vancouver 2018

OpenSDK OE builds in MMWG

Andrey Konovalov
TI assignee
Linaro Multimedia Working Group



OpenSDK OE builds

- Are still based on RPB (even though the support model for Reference Platform Builds has changed) - no plans to change that.
- meta-lhg layer contains the components developed by LHG/MMWG
- The builds are running in jenkins (ci.linaro.org)
- The images are published at
<http://snapshots.linaro.org/openembedded/pre-built/lhg/>
- Some of the LAVA test results are available at
<https://qa-reports.linaro.org/lhg/openembedded-morty/>



OpenSDK OE builds matrix

manifest branch	image to build	boards
morty	rpb-westonchromium-image	am57xx-evm, db410c, hikey, TBD: imx8
rocko	rpb-westonchromium-image	am57xx-evm, db410c, hikey, TBD: imx8
playready/morty	rpb-westonchromium-image	hikey
wpe/morty	lhg-westeros-wpe-image	hikey-32
TBD	TBD	am57xx-evm

Sumo builds will be added after the sumo bsp branches appear.

The current focus is on rocko builds.



The current focus areas

- OpenCDM, Secure Data Path

Key components:

- wayland/weston/westeros
- OP-TEE
- Chromium
- WPE browser
- Linaro OCDM plugin and OCDMi (based on the work by fraunhoferfokus)

- Hardware accelerated video

Key components:

- wayland/weston
- gstreamer + out of tree patches from silicon vendors

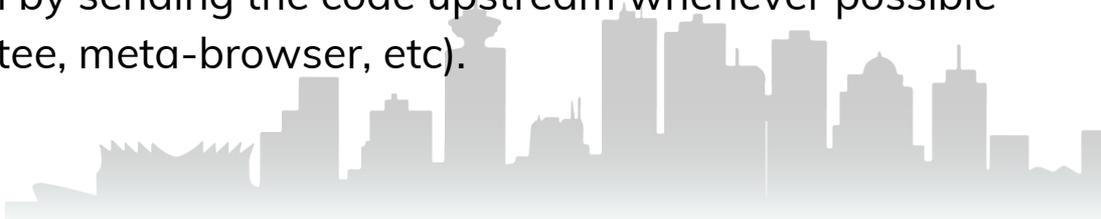


Meta-lhg structure

<https://github.com/linaro-home/meta-lhg>

- meta-lhg layer - components developed / being worked on in LHG:
chromium bbappend to add OpenCDM plugin, CDMi and optee-aes-decryptor recipes, rpb-westonchromium-image, ...
- meta-lhg-integration layer - glue to integrate components external to LHG:
e.g. ti-sgx-ddk-um_%.bbappend, linux-hikey_%.bbappend to enable SDP, ducati and vpe gst plugins from meta-arago, etc
- meta-lhg-wpe layer - WPE browser integration:
lhg-westeros-wpe-image, and wpewebkit_0.1.bbappend

The idea is to keep these layers small by sending the code upstream whenever possible (meta-96boards, meta-rpb, meta-optee, meta-browser, etc).



OP-TEE integration

Boards which have OP-TEE support, and can be used to develop Trusted Apps:

- HiKey - is most actively used inside MMWG, but it is out of production
- am57xx-evm-hs - the sources are public, but the tools to create the secure image are not: the builds in jenkins and the tests in LAVA aren't possible
- imx8m-evk - looks promising (SDP can be implemented); adding it to the MMWG's builds is in progress. In the meanwhile MMWG uses a build based on NXP's GA release, which is based on their own distro, not RPB.



OP-TEE integration: a problem

Each of the three above boards uses its own git tree for optee_[os|client|test] and the optee recipes (or bbappends which overrides almost everything).

	HiKey (meta-linaro.git/meta-optee)	am57xx-evm	imx8m-evk
Morty	2.4.0	2.6.0*	3.0.0***
Rocko	2.5.0	3.1.0**	3.0.0****
Sumo / master	2.6.0 / ?	N.A. / ?	N.A. / ?

* - optee_os fork is based on upstream 2.6.0-3-g639e5b8, the diff is 4 files changed, 13 insertions, 36 deletions.

** - optee_os is the snapshot of the current upstream; no out of tree patches.

*** - is the upstream tree at 3.0.0-35-gb14416d plus 24 patches on top of it

**** - is the upstream tree at 3.0.0-35-gb14416d plus 45 patches on top of it

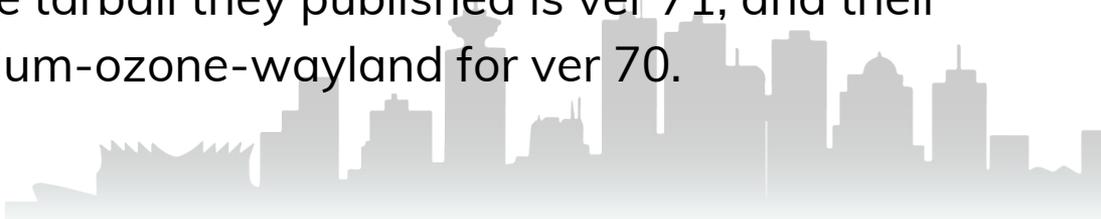
Chromium integration

In MMWG buids for Wayland version of Chromium we use:

- morty: version 53 (pinned commit from upstream meta-browser)
- rocko: version 65 (chromium source tarball with modifications by Igalia)

Problem: in OSSystems/meta-browser chromium-ozone-wayland is stuck at 64.0.3274.0.r517731.igalia.1.bb (since Dec. 2017), while chromium-x11 is updated every week or two.

Igalia is actively working on their “Chromium on Wayland” project. Currently, the most recent Chromium source tarball they published is ver 71, and their fork of meta-browser has chromium-ozone-wayland for ver 70.



Linaro OpenCDM and Chromium v65

OpenCDM has been updated to account for two major changes in Chromium v65:

- Now GN build system is used to build Chromium
- External Clear Key CDM in Chromium was changed to use a more recent CDM interface.

To address this, a new branch has been created in linaro-mmwg's OpenCDM repository at github. Currently it misses the unification with WPE, and only Clear Key CDM is supported.

MMWG's rocko build has been updated to enable OpenCDM for Chromium. The morty builds will continue using v53.



Linaro OpenCDM and Chromium v65 (continued)



TBD: port forward from the v53 OpenCDM branch:

- the possibility to build debug version of OpenCDM plugin while building the release version of Chromium
- PlayReady support
- unification with WPE



Status of the Westeros/WPE build

The current jenkins build for WPE at ci.linaro.org is `lhg-oe-wpe-build`.

The current target boards: hikey.

It is based on morty release, and uses the `chromium-53.0.2785.143` branch of Linaro OpenCDM which supports both Chromium (v53) and WPE.

This version of OpenCDM supports EME v0.1b, and this makes us to use an older version of WPE.



Testing the builds

The current builds are run 3 times a week (both morty, and rocko).

The jenkins job submits a test job to run a set of tests in LAVA.

The tests use the Robotframework. A dummy HDMI dongle is to be connected to the device under test.

The current test cases are:

- OP-TEE: xtest (benchmark and regression tests)
- Weston: weston-simple-egl
- Gstreamer: plays back a video file
- Chromium: starts the browser; plays a video from youtube; plays a morphtargets/horse example from threejs.org (webgl); runs a ClearKey EME test (uses OpenCV to recognize the expected image)



Testing the builds - next steps

- WIP: new build in jenkins which is triggered by commits to OpenCDM and OpenCDMi
- TODO: use more tests (gst-validate, ocdm_encrypted_media_supported_types_browserstest.cc, etc)
- TODO: find out a better way to display the test results at qa-reports.l.o
- TODO: <https://www.chromium.org/chromium-os/testing/chamelium> could be used to verify the output to the HDMI connector vs taking the screenshot on the board under test (Linaro have got couple chamelium boards)



Useful links

<https://github.com/linaro-home/lhg-oe-manifests>

The README.md there has the build instructions, and sometimes the instructions to run the image. Just select the proper branch (morty, rocko, wpe/morty).

