

## Replicant - Issue #705

Feature # 1539 (New): Graphics acceleration

### Incomplete EGL implementation

11/12/2013 01:43 PM - Paul Kocalkowski

<b>Status:</b>	New	<b>Start date:</b>	11/12/2013
<b>Priority:</b>	Urgent	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Graphics	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Replicant 6.0	<b>Spent time:</b>	0.00 hour
<b>Resolution:</b>		<b>Device:</b>	
<b>Description</b>			
<p>Replicant is currently using the fallback Android EGL implementation, which is incomplete. The missing features of this implementation cause multiple issues:</p> <ul style="list-style-type: none"><li>• Slowness issues on devices with slow CPUs</li><li>• YUV is not properly supported (only black &amp; white), which forces us to use RGB565 as a preview format on the camera modules. This causes several issues: * preview-based detection (such as barecodes, QR codes, etc) doesn't work * panorama feature is broken</li><li>• Screenshots do not work</li><li>• Previews of the windows in the tasks switcher are missing</li><li>• Some applications rely on unimplemented features, such as Firefox, Document Viewer and crash as they are missing</li></ul> <p>There are patches to use llvmpipe with Replicant, and work is ongoing to:</p> <ul style="list-style-type: none"><li>• investigate how to improve the performances of llvmpipe</li><li>• investigate how to integrate llvmpipe and libagl at the same time on the device, in a clean way.</li></ul>			

### History

#### #1 - 11/17/2013 10:01 PM - Rodger Fox

Since slowness is the problem with Mesa, is this bug solvable in a way besides using graphics hardware?

Or should the focus be on contributing to and integrating projects like Lima driver into Replicant?

#### #2 - 11/18/2013 03:21 PM - Paul Kocalkowski

It would be possible to complete the Android EGL implementation so that it is fully-featured and optimize it with NEON and ARM ASM code so that it becomes faster. There are also ways to improve things with hwcomposer and the framebuffer drivers and other per-hardware things.

But the real way to solve this is to have per-GPU free software implementations, such as Lima. Lima is not ready yet, but once it doesn't depend on the proprietary compiler and gets integrated in mesa, we'll certainly try and make it work with Replicant.

#### #3 - 10/16/2014 11:12 PM - My Self

IMHO the Lima development stagnated since the last post, or are there any insider news?

I really want to take screenshots and the app alternative AndroSS:

<https://f-droid.org/repository/browse/?fdfilter=screenshot&fdid=net.tedstein.AndroSS>

doesn't work, too. (It takes screenshots, but the pictures are broken; looking like colored barcode art).

Does anybody knows a workaround for the screenshot feature (on Replicant), please?

**#4 - 10/19/2014 12:27 PM - Paul Kocialkowski**

IMHO the Lima development stagnated since the last post, or are there any insider news?

Quote: *the tamil driver will see work whenever/ifever i am in the mood for working on it again*

Does anybody knows a workaround for the screenshot feature (on Replicant), please?

Well, screenshots work on the browser, so maybe we should look how it's done there. I don't have time to work on graphics acceleration, as usual.

**#5 - 10/25/2014 08:58 AM - My Self**

[I've removed the system browser, so I can't test that].

But I've found a functional workaround to make screenshots over ADB:

<http://blog.shvetsov.com/2013/02/grab-android-screenshot-to-computer-via.html>

**#6 - 01/17/2015 03:18 PM - Paul Kocialkowski**

- Target version changed from 21 to Any version

**#7 - 01/22/2015 05:54 PM - My Self**

IMHO the <incomplete EGL implementation> is an acute security related problem and will become a bigger one in the future. The reason is, that I don't see another chance to get a more secure browser to work on Replicant without OpenGL ES, at the moment. Please have a look at the discussion here: <http://redmine.replicant.us/boards/39/topics/8007>

A working EGL will bring also the following benefit:

- There would be a workaround for "Add support for more video software-decoding codecs, such as Theora" from the Tasks list, because with a working Firefox, I've successfully tested to run a .ogv (Theora) Video file out of the browser.
- This video was also switchable to fullscreen, so this could be another workaround for "Make the embedded video player in the browser work (or make it switch to fullscreen when playback starts)" on the Tasks list.
- You could delete "Write a modern Gallery application that doesn't rely on missing OpenGL features" from the Tasks list.
- Of course that would delete the EGL point itself "Improve the generic Android EGL implementation so that it implements all the currently missing features" from the Tasks list.
- The points above could be the some kind of workaround for "Make software video decoding work on Replicant 4.2" on the Tasks list, too.
- And last but not least, it (probably!) could lead to less battery consumption (<http://redmine.replicant.us/boards/9/topics/7953>) because there would be less (battery draining) software rendering, (but I have to test that first).

**#8 - 12/11/2015 01:42 PM - Denis 'GNUtoo' Carikli**

- Parent task set to #1521

**#9 - 12/11/2015 01:42 PM - Denis 'GNUtoo' Carikli**

- Parent task changed from #1521 to #1491

**#10 - 12/11/2015 01:45 PM - Denis 'GNUtoo' Carikli**

- Parent task changed from #1491 to #1539

**#11 - 12/11/2015 02:50 PM - Denis 'GNUtoo' Carikli**

- Device set to Not device specific

**#12 - 01/04/2017 12:16 AM - Wolfgang Wiedmeyer**

- Device deleted (Not device specific)

It seems that TWRP has working screenshots. Their implementation could be useful:

[https://github.com/omnirom/android\\_bootable\\_recovery/blob/b523650c8ecb6751409120a38e52a66a3e48753f/minuitwrp/graphics\\_utils.cpp#L29](https://github.com/omnirom/android_bootable_recovery/blob/b523650c8ecb6751409120a38e52a66a3e48753f/minuitwrp/graphics_utils.cpp#L29)

**#13 - 03/17/2017 12:39 PM - Wolfgang Wiedmeyer**

- Category set to Graphics

- Priority changed from High to Urgent

**#14 - 04/09/2017 06:07 PM - Wolfgang Wiedmeyer**

- Assignee deleted (Paul Kocalkowski)

llvmpipe is available in Replicant 6.0 and can optionally be enabled. It is still too slow to be the default renderer.

llvmpipe can become fast enough if it is optimized for ARM. Looking at optimized code for architectures that are supported by llvmpipe could be a viable strategy to figure out the instructions that are worth to optimize (e.g. by implementing them in ARM assembly).

Making use of [ARM NEON](#) should improve speed as well.

**#15 - 04/22/2017 06:40 PM - Wolfgang Wiedmeyer**

- Target version changed from Any version to Replicant 6.0

**#16 - 05/06/2017 06:23 PM - Wolfgang Wiedmeyer**

llvmpipe in Replicant 6.0 is currently based on Mesa 13.0.3. We use the Mesa sources from Android-x86 and they updated to Mesa 17.0.4:

<https://osdn.net/projects/android-x86/scm/git/external-mesa/commits?branch=marshmallow-x86>

Unfortunately, it's not possible to merge their updates because they seem to rebase. We only need one additional patch, so it's better to switch to a new branch and adapt our patch on top of it. This breaks the current way of how branches are handled. In other repos, there is only one branch per Replicant release. For the Mesa repo, we'd need several branches to preserve the tags.

Maybe a naming scheme for the branches like replicant-6.0-mesa-13.0, replicant-6.0-mesa-17.0 could work.

**#17 - 12/05/2017 09:30 PM - Niclas Hoyer**

Just a quick note: a new Mali-400 driver seems to be in very active development. It is far from complete, but passed a milestone on last week: it runs

the kmscube successfully, so the whole stack basically works: <https://github.com/yuq/mesa-lima/wiki>

This GPU is used in Samsung i9100 and i9300, eventually others.

**#18 - 12/10/2018 04:39 PM - Denis 'GNUtoo' Carikli**

- *Description updated*

**#19 - 01/21/2019 05:53 PM - Denis 'GNUtoo' Carikli**

Some time ago someone gave me that link on IRC: <https://www.mesa3d.org/perf.html> that explains how to increase performances in MESA when using software rendering.

**#20 - 02/01/2019 07:33 PM - Kurtis Hanna**

Niclas Hoyer posted a link a while ago to a new Mali-400 driver that still seems to be in very active development. They said that the whole stack basically worked back then, so I'd be curious to see how many bugs they still have left to fix. The link Niclas provided doesn't seem to be used anymore. You can find the links to the new active git repo here:

<https://gitlab.freedesktop.org/lima>

<https://gitlab.freedesktop.org/lima/web/wikis/home>