

Targets Evaluation

This page contains information on the freedom of various device, along with the ability to be ported to Replicant. It can be used for many different things:

- It can be used to evaluate potential target devices
- It can be used to denounce very vicious attacks on users freedom, for instance if the kernel is signed.
- It can be used to track if new devices have removable batteries

To track the upstream status of various devices, we also have the [Upstream](#) wiki page. Here we really need information on the upstream status of smartphones and tablets, even if the devices are worse than the ones currently supported as it helps plan the future of Replicant. For instance we could use the data to understand how to build virtuous circles with various upstream project that care less about freedom.

That page was previously used to evaluate potential target devices to which we may port Replicant, but the information on the devices Replicant would not want to support turned out to be useful as well.

Also note that adding support for a new device in Replicant takes some time and effort.

Because of that, adding new devices in that page doesn't guarantee at all that the device will be supported in Replicant, even if that device has significantly less freedom issues than the ones currently supported by Replicant, as some people still need to step up and do the work.

Other resources:

- There is also a [forum section](#) about devices evaluations.
- The mailing list can also be used to discuss about device evaluations

The [developer guide](#) explains the terminology used on this page. See also the [website](#) for background information. The [porting guides](#) provide information how the following requirements can be evaluated.

Improving data on devices

It might be a good idea to investigate how to use Wikidata to store the data and be able to automatically generate some tables from that.

This could also enable us to share the work with other communities. However the main issue with doing that would be to ensure that the data quality is good enough to our standards.

The inventaire project has exactly the same issues and did [a presentation](#) on how they deal with it.

It's also possible to build compatibility tables with [sparql queries](#) but much more research needs to be done on how to best interface with wikidata data.

Other data sources

- LineageOS generates its wiki compatibility page from [yaml data](#). We can also use the same data to find devices that might be interesting to support in Replicant.
- Wikidata had or has already a community of people involved in adding smartphones in wikidata.
- The gsmarena website also has some information on smartphones.
- Wikipedia has a lot of information on smartphones, including specifications.

Replicant even has a tool: "https://git.replicant.us/replicant/vendor_replicant-scripts/tree/research" to automatize that

Evaluation criteria for accepting new devices in Replicant

Minimal requirements

List of the bare minimum requirements for a device to be ported to Replicant:

- Users must not be required to use download, install or use any nonfree software to install Replicant.
- The bootloader must not prevent users from installing modified kernel versions.

- Replicant will need to be usable on the device without having to redistribute any nonfree software.
- Users or automatic programs are not expected to download and install nonfree software to make the device usable.
- The device also need to have enough resources to run Android. For instance, at the time of writing, it's unrealistic to port devices with 128M of RAM to Replicant.
- User replaceable batteries:
 - The batteries can be removed and replaced with basic tools like a screwdriver and stamp tweezer to open the back cover, without damaging too much the device.
 - Requiring to use a heat gun to open the case in a way that damages it makes it too complicated for Replicant developers, users, and vendors selling devices with Replicant pre-installed, to do it without damaging the device.
 - Not being able to remove batteries results in the inability to easily find the device second hand and replace the battery. Developers also have to rush for supporting the device, as people prefer to buy it new as the battery will last longer this way.
- A somewhat isolated modem:
 - A modem connected with shared memory or in the same SOC is not OK anymore, unless:
 - The modem can run fully free software that is under the user control, which Replicant will use and redistribute.
 - An IOMMU that is setup correctly. As IOMMU are complicated to setup correctly, and to really isolate from the modem they would require:
 - Complete documentation, including documentation that shows that the modem cannot re-configure the IOMMU
 - To make sure that the IOMMU is setup correctly, before the modem is booted. For instance:
 - If the modem boot is not controlled by Replicant, you would need to make sure that at boot, the IOMMU is setup before the modem link is initialized. This typically requires upstream support for the device in a fully free bootloader.
 - If instead, the modem boot is controlled by Replicant, you could just do the IOMMU setup in the kernel and upstream the code to be sure it works as intended.

Also keep in mind that:

- The device must have at least one usable Linux kernel with with complete corresponding source code. If this is not the case, it might still be possible to obtain the complete corresponding source code from the device manufacturer, or to port the Linux kernel to the device yourself before starting to work on Replicant.
- Some devices might have CPU architectures not supported by the latest Android version, like MIPS. Replicant may refuse to support such devices due to concern over long term maintenance of the code.
- Replicant doesn't support CDMA yet. If your device has CDMA, adding support for CDMA may or may not be a lot of work depending on the device.
- As Replicant 9 will be based on upstream Linux, it's a good idea to have the some support for the device in upstream Linux and work with upstream to add as much support as possible, in order to limit the maintenance of such code.

If you still want to add support for devices that don't meet the minimal requirements, we still have some options left:

- We are open to collaborate with people wanting to add support for devices (that Replicant doesn't want to support) in other upstream projects. For instance, Replicant 9 will use a kernel that is very closely based on upstream Linux. Thanks to that, the source code might be useful to run Android on various devices with upstream Linux kernel.
- We are open to upstreaming the code made by Replicant in other projects as well. We already upstreamed several libraries in CyanogenMod or LineageOS.
- We are also open to collaborate in helping you fork Replicant if there is no suitable upstream for supporting your device.
- If you or someone else has some idea to add here, feel free to request a wiki account to add it, or to suggest it on IRC, the mailing list, in a bug report, etc.

Good target

What makes a good target for Replicant:

- Bootloader must be ready to flash images, or be very easy to unlock
- Free user-space implementation for most of the hardware already available
- Standard or well-known protocols used in the hardware (V4L2/ALSA/NMEA, etc)
- Proprietary loadable firmwares only needed for Wi-Fi/Bluetooth, and not camera, or other peripherals etc
- Freedom-compliant hardware design: Modem isolation (no shared mem, GPS, audio)
- It should be possible to deactivate the modem, either by hardware or software.
- To make the port easier, and to amount lower the maintenance work it would also be a good idea if the chosen device meets or exceed with some margin the minimum [HardwareRequirements](#) of the latest Android version.

Ideal target

An ideal target for Replicant would be:

- Free bootloaders that allow easy flashing for the user
- All the hardware using standard protocols, kernel-drivers and no user-space binary blobs

- Components do not require loadable non-free firmware
- There are ways to update the internal firmwares of the chips (for instance to allow a free software modem implementation)
- Has total modem isolation (doesn't control audio, nor GPS, mem, NAND, etc)
- not SIM locked

Known to be bad targets

Here is a list of the bad targets for Replicant and reasons why:

- **Qualcomm MSM/QSD** devices are known to have bad hardware design where the modem controls GPS, audio (including mic), RAM, NAND.
However, some MSM/QSD SoCs don't ship with a modem embedded, so they may be good targets. See [the page on Qualcomm System On a Chip](#) for more details on this issue.
- **Nvidia Tegra 2** devices are known to be too slow with Replicant, since there is no NEON nor any free hwcomposer/gralloc. Perhaps newer Tegra SoCs (which include NEON) would be faster on Replicant.
- **Motorola** devices manufactured after the Milestone (inclusive). These have non-free locked bootloaders that check kernel signatures. **sbfl_flash**, a utility capable of flashing Motorola devices, is non-free.

Devices evaluation

Android devices

Manufacturer	Device	RAM size	Upstream	Flash method	SoC	Graphics	Sound	Modem	Camera	GPS	Sensors	Removable battery	nonfree	Freedom issues	Verdict
Asus	Nexus 7	?	CM 10 (Android 4.1)	Fastboot	Tegra 3	?	Free module	3G version modem uses AT	Non-free tegra module, perhaps not even v4l2	BCM 4751: secret protocol	?	?	?	?	Port possible but unlikely to succeed
Geek sPhone	One	?	CM 7.2 (Android 2.3)	Recovery	MSM 7K	Should be fast	Should work	Non-standard protocol but AT seems possible, bad design for freedom	MSM undocumented mechanism	Probably NMEA from modem, bad design for freedom	?	?	?	Qualcomm, hardware is BAD for freedom	
Geek sPhone	Zero	?	CM 7.2 (Android 2.3)	Fastboot	MSM 7227	Should be fast	May need non-free firmwares	?	MSM undocumented mechanism	Probably NMEA from modem, bad design for freedom	?	?	?	Qualcomm, hardware is BAD for freedom	
LG	Nexus	?	CM	Fastb	Qualc	?	?	Unkn	?	?	?	?		Really BAD: has too	

	s 4		10.1 (Android 4.2)	oot	omm Snapdragon S4 Pro APQ8064			own protocol							many proprietary things and the hardware is bad for freedom: modem controls the GPS (included in the modem chip), audio and perhaps NAND too
LG	Optimus L9	?	CM 10.1 (Android 4.2)	?	OMAP 4430	?	?	?	?	?	?	?	?	?	?
Motola	Defy	?	CM 7.2 (Android 2.3)	Recovery	OMAP 3630	Should work	?	?	Non-free	?	?	?	?		Kernel is signed, making a Replicant port impossible
Samsung	Galaxy S II, I9100 G variant	1G	?	?	OMAP 4430	?	?	XMM 6260 MIPI	?	?	?	?	?	?	Looks really good: * The modem looks isolated * The modem probably use the samsung-ipc protocol * Has a free software loader * Unsigned bootloader
Samsung	Galaxy Note 2 (LTE)	?	LineageOS 14.1	heimdall	Exynos 4412	1280x720	?	Qualcomm MDM 9X15 HSIC	?	?	?	?	?	?	?
Samsung	Galaxy Note 3 (HSPA+ version)	?	LineageOS 14.1	heimdall	Exynos 5420	1080x1920	?	XMM 6360 HSIC	?	?	?	Yes	?	?	?

Samsung	Galaxy Note 4 (LTE Cat.4 & South Korea LTE Cat.6 & LTE Cat.9 (S-LTE))	?	?	?	Exynos 5433	?	?	?	?	?	?	Yes	?	?	?
Samsung	Samsung Galaxy Note Edge (South Korea version)	?	?	?	Exynos 5433	?	?	?	?	?	?	Yes	?	?	?
Samsung	Galaxy Note 5	?	?	?	Exynos 7420	?	?	?	?	?	?	No	?	?	?
Samsung	Galaxy S4 (GT-I9500)	?	?	?	Exynos 5410	?	?	?	?	?	?	Yes	?	?	?
Samsung	Galaxy S5 (SM-G900H)	?	LineageOS 14.1	heimdall	Exynos 5422	1080x1920 screen	?	XMM6360 HSIC	?	?	?	Yes	proprietary-files.txt	?	?
Samsung	Galaxy S5 Neo (SM-G903F, SM-G903M, SM-G903W)	?	LineageOS 16.0	heimdall	Exynos 7580	Mali-T720 MP2	?	Shared memory	?	?	?	Yes	s5neolte/proprietary-files.txt universal7580-common/proprietary-file-s.txt universal7580-common/proprietary-file-s-bsp.txt	Has at least a mode m that is not isolated	not isolated mode m => too bad for freedom to support in Repllicant
Samsung	Galaxy	?	?	?	Exynos	?	?	?	?	?	?	No	?	?	?

ung	y_S6				os 7420										
Samsung	Galaxy S7	?	LineageOS 14.1	heimdall	Exynos 8890	2560x1440	?	?	?	?	?	?	?	?	?
Samsung	Galaxy S7 Edge	?	LineageOS 14.1	heimdall	Exynos 8890	2560x1440	?	?	?	?	?	?	?	?	?
Samsung	Galaxy S8 international	?	?	?	Exynos 8895	?	?	?	?	?	?	No	?	?	?
Samsung	Galaxy Tab 8.9	?	CM 9.0 (Android 4.0)	Heimdall	Tegra 2	?	Free module	Samsung IPC XMM 6260	Non-free tegra module, perhaps not even v4l2	BCM 4751: secret protocol	?	?	?	?	PaulK had one but failed to port to it because of Tegra slowdown issues
Samsung	Galaxy Xcover 3	?	?	?	Marvell Armada PXA1908	Vivante GPU	?	?	?	?	?	Yes	?	?	?
Samsung	Galaxy Xcover 4	?	?	?	Exynos 7570	?	?	?	?	?	?	Yes	?	?	?
Samsung	Nexus 10 - codenamed Mantana	?	Downstream LineageOS 16.0 ; Downstream AOSP 9.0 ; PostmarketOS wiki and repo ; Devices with same SoC that have	Fastboot	Exynos 5250 , also called the Exynos 5 Dual. 1.7 GHz dual-core ARMv7 Cortex-A15 User manual	Free gralloc, hwcomposer, non-free 3d, beware: ultra high resolution (could be slow), display may need non-free firmware,	May need non-free firmware	No mode m	Free module	BCM 4751: secret protocol	Free module	No, but ifixit battery replacement is ranked as moderate, taking only 10 to 15 minutes.	?	?	Port of Replicant 6.0 was planned but never pursued

some
upstr
eam
suppo
rt :
Sams
ung
SMD
K525
0 eval
board
:
[kernel
u-boo
t](#);
[Sams
ung
Chro
mebo
ok
Serie
s 3
XE30
3](#) -
coden
amed
snow
:
[kernel
u-boo
t](#) can
use
nv-U-
Boot
to
[boot
unveri
fied
kernel
tutori
al to
install
upstr
eam
kernel
at
boot
time
ifixit
teard
own
XDA
threa
d
about
portin
g
AOS
P/Cya
noge
nMod
to this
Chro
mebo
ok
Tutori
al on](#)

Mali
T604
GPU
can
run
[panfr
ost](#)

			running Android on this Chromebook Tutorial on bootloader unlock on this Chromebook ; Arndale board : wiki kernel u-boot it supports non-secure JTAG debugging ; HP Chromebook 11 G1 - codenamed spring : kernel u-boot												
SHIFTPH ONE	SHIF T5me	?	?	?	MT6797X Helio X27	?	?	?	?	?	?	?	?	Modem in the same SOC	
SHIFTPH ONE	SHIF T6m	?	?	?	MT6797X Helio X27	?	?	?	?	?	?	?	?	Modem in the same SOC	
SHIFTPH ONE	SHIF T6mq	?	?	?	Qualcomm snapdragon 845	?	?	?	?	?	?	?	?	Modem in the same SOC	
SHIFTPH ONE	SHIF Tmu	?	?	?	Qualcomm snapdragon 865	?	?	?	?	?	?	?	?	Modem in the same SOC	
Wiko	Highway	?	?	?	Tegra	?	?	?	?	?	?	?	?	TOD O:	

4G																			Check if the bootloader is signed
--------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----------------------------------

Non-android devices

Device	Manufacturer	Android port	Bootloader	SoC	Standardness	Blobs	Firmwares	Modem	Notes
Neo Freerunner	Openmoko	AoE	u-boot, flash with dfu-utils	Samsung S3C2442, CPU @400Mhz	Audio: ALSA, GPS: NMEA, Modem: AT	No	No	Total isolation	Hardware is too old to be supported by Replicant: * armv4t * Only 128M of RAM
N900	Nokia	Nitdroid	partially signed, chainloaded u-boot, flash with 0xff	OMAP3430, CPU @600Mhz	Voice calls depend on cmt_speech pulseaudio plugin, modem protocol is non-standard but implemented in ofono	Nitdroid uses blobs	Nitdroid uses non-free firmwares (Wi-Fi, bluetooth, camera)	GPS is controlled by the modem, modem transport is over high-speed serial	* Has only 256M of RAM
iPhone	Apple	iDroid	iDroid bootloader seems free	iPhone 3G: Similar to Samsung S5PC110	iPhone 3G: Audio: ALSA, Modem: AT, Wi-Fi: libertas	Probably needs some	Probably needs some	Known to never sleep	already supported by the iDroid project , however it has a really bad status : it can't suspend to ram...
FLOW G1.55	GizmoForYou	Stock	xloader + u-boot	OMAP3	Audio: ALSA, Modem: AT	?	Wi-Fi/Bluetooth	Needs to be checked, could be OK	Instructions to build Android 2.2 for it are provided: http://www.gizmoforyou.net/wiki/index.php/Building_G155_Android Unfortunately that phone isn't cheap.

PengPod1 000/700	PengPod	? Android development forum	u-boot	Allwinner A10	?	both 8192cu and rtl8188eu Wi-Fi chips need non-free firmware	Some have rtl8192cu Wi-Fi, some have rtl8188eu	N/A	More focused on running GNU/Linux than Android. Wiki
---------------------	---------	---	------------------------	------------------	---	---	---	-----	--

There are a small number of do-it-yourself phones where the user is expected to assemble the phone. Examples of this include the [odroid](#) phones which comes with android support. However, the issue is that not all users want or are able to assemble their own phones.