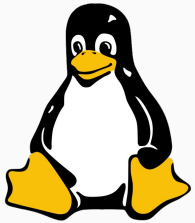
Kernel support for Mobile Linux: The missing 20%

Luca Weiss | FOSDEM 2025

About Me

- **Luca Weiss**
- postmarketOS Core Contributor
- Many patches in Linux Kernel
- OpenRazer Maintainer
- Software Longevity Team at Fairphone



fosstodon.org/@z3ntu

lucaweiss.eu

Mainline Linux on phones in 2025

Phones

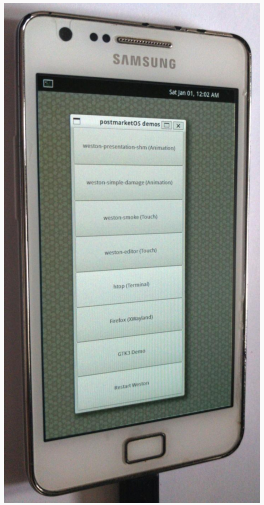
Smartphone

Device	Codename	Architecture	USB Net	Flashing	Touch	Screen	Wifi	FDE	Battery	3D	IMU	Audio	Bluetooth	Camera	GPS	Mobile Data	SMS	Calls	USB-OTG	NFC
BQ Aquaris X5	bq-paella	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	-
Fairphone 4	fairphone-fp4	aarch64	Y	Y	Y	Y	Y	Y	N	Y	Y	N	Y	N	N	N	Y	P	Y	N
Google Pixel 3a	google-sargo	aarch64	Y	Y	Y	Y	P	Y	P	Y	Y	P	Y	P	N	Y	Y	P	N	P
Lenovo A6000	lenovo-a6000	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y			Y	Y	Y	-	-
Lenovo A6010	lenovo-a6010	aarch64	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y			Y	Y	Y	P	-
LG Optimus Vu	lg-p895	armv7	Y	Y	Y	Y	Y	Y	Y	P	Y	Y	Y						P	Y
Motorola Moto G4 Play	motorola-harpia	aarch64	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	N	Y	Y	Y	Y	P	
OnePlus 6	oneplus-enchilada	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	P	Y	Y	P	P	P
OnePlus 6T	oneplus-fajita	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	P	Y	Y	P	P	P
PINE64 PinePhone	pine64-pinephone	aarch64	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	Y	-
PINE64 PinePhone Pro	pine64-pinephonepro	aarch64	Y	-	Y	Y	Y	Y	P	Y	Y	Y	Y	P	Y	Y	Y	Y	N	-
Purism Librem 5	purism-librem5	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	Y	-
Samsung Galaxy A3 (2015)	samsung-a3	aarch64	Y	Y	Y	P	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y
Samsung Galaxy A5	samsung-a5	aarch64	Y	Y	Y	P	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Samsung Galaxy E7	samsung-e7	armv7	Y	Y	Y	P	Y	Y	Y	Y	P	Y	Y	N		Y	Y	Y	Y	Y
Samsung Galaxy Grand Max	samsung-grandmax	armv7	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	Y	N		Y	Y	Y	Y	-
Samsung Galaxy S III	samsung-m0	armv7	P	Y	Y	Y	Y	Y	Y	Y	Y	P	P	P	N	N	N	N	Y	-
Samsung Galaxy S4 Mini Value Edition	samsung-serranove	armv7	Y	Y	Y	P	Y	Y	Y	P	-	Y	Y	N	Y	Y	Y	Y	-	-
SHIFT SHIFT6mq	shift-axolotl	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	N	N
Wileyfox Swift	wileyfox-crackling	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	P	-
Xiaomi Mi A1	xiaomi-tissot	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	P	P	Y	P	Y	-
Xiaomi Mi A2 Lite	xiaomi-daisy	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	P	P	Y	P	Y	-
Xiaomi Mi Note 2	xiaomi-scorpio	aarch64	Y	Y	Y	Y	Y	P	P	Y	N	P	Y	P	N	N	P	N	Y	Y
Xiaomi Pocophone F1 / POCO F1	xiaomi-beryllium	aarch64	Y	Y	Y	Y	Y	Y	P	Y	Y	Y	Y	P	P	Y	Y	Y	N	-
Xiaomi Redmi 2	xiaomi-wt88047	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	-
Xiaomi Redmi 4 Prime	xiaomi-markw	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	P	P	Y	Y	Y	-
Xiaomi Redmi 5 Plus	xiaomi-vince	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	Y	P	P	P	Y	P	Y	-
Xiaomi Redmi Note 4 / 4X	xiaomi-mido	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	P	P	Y	Y	Y	-
Xiaomi Redmi S2 / Redmi Y2	xiaomi-ysl	aarch64	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	P	P	P	Y	Y	Y	-

Let’s start with some history

- No noteworthy mainline support for any phone (2016)
  - arm32 qcom: Nexus 7, Sony Xperia Z, Sony Xperia Z1 (+ SBCs)
  - arm64 qcom: only SBCs or test platforms
  - Nexus 5 with bare functionality upstream: August 2016
  - Freedreno demo with Nexus 7 tablet in 2016

- postmarketOS started in 2017 - first Linux distribution for smartphones
  - 2 phones supported with downstream (v3.0 & v3.4) kernel
  - ‘Weston’ Wayland reference compositor with touchscreen input
  - Weston demo & GTK3 demo apps

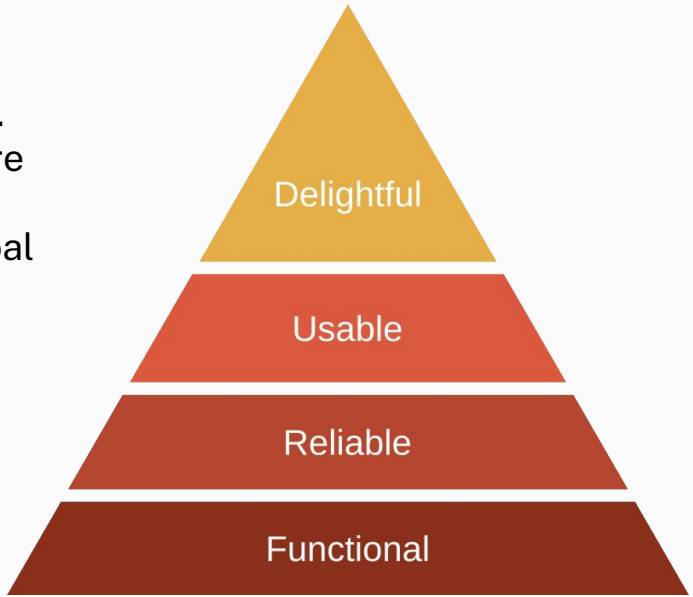


- Mainline Linux support for has been rapidly expanding and improving since!
  - arm32 qcom: 34 boards
  - arm64 qcom: ~140 boards
  - And more out-of-tree!

- Many phones that can be used as daily driver in some scenarios (e.g. podcasting machine)

But it sorta works!

- Two pillars: Functionality & Reliability
  - Functionality: Does it work?
  - Reliability: Does it work **well** and **consistently**?
- Device bringup primarily focuses on functionality
- Reliability is what the user will see - and get annoyed about.
- Testing reliability is hard - and getting it reliable is even more
- A phone with “just” reliably working phone calls is difficult
- *We* also need to figure out what we want to achieve - end goal
  - If *we* want to make a fully functional & reliable phone running mainline, we need to really focus on that
  - Everything’s done voluntarily without central coordination and everybody works on their own pet project - which is awesome!



Design Hierarchy of Needs

Functionality

- ≥ 80% of functionality is working for lots of devices
- What's commonly missing?
  - Cameras – lots of progress is happening here lately!
  - Battery fuel gauge driver
  - Phone calls / mobile data
  - A subsection of essential functionality
- Bringup is quite straightforward for recent devices
  - Lots of ground work is already done
  - SoCs are only some variations of each other
  - After some days of work: new SoC can work reasonably well
- Getting some functionality upstream is difficult
  - Camera sensor driver – lack of docs and knowledge
  - Vibration motor driver – lack of docs
  - Touchscreen hacks for 3rd party panels
  - Display stack hacks
  - Some patches can block others patches from going upstream
  - And mostly: motivation and time!
  - □ A kernel fork will be necessary for the foreseeable future

Reliability

- Bringup is for functionality
- Reliability is often ignored then - if it mostly works it's good enough
- Devs don't use devices over extended periods of time
  - E.g. multiple suspend cycles and longer uptime
- How to test? Dogfooding?
- Ensuring no regressions appear is energy draining
  - One person supporting multiple devices - all tests should be done on every device
  - Requires lots of time and persistence

What do *we* need?

- Who is “*we*” anyways?
  - Hypothetical group wanting to get one device working perfectly - doesn’t matter which device in particular. Doesn’t matter if goal is that techies or non-techies will use the device
- Multiple people need to come together with different roles (kernel devs, testers, project managers)
- **Kernel devs** explore/brainstorm ways to resolve functionality and reliability issues
- **Testers** test new releases, ensure they work as expected, catch regressions. Regressions need to be bisected ⇒ needs more dev and tester time
- **Project managers** hold everything together and decide what needs to be prioritized
- One person cannot do everything
  - Especially for fixing issues that devs (including myself) lack create approaches to resolve some issues
- Junior kernel devs can maintain stable branches (e.g. merging linux-stable)
  - Anybody just knowing git and willing to learn a bit can do that - easy way to get started!
  - Also simple rebasing on new kernel releases (and e.g. testing -rc) is simple enough
- Update, build, test cycle repeats at least every ~9 weeks
- Motivate each other - it’s both a technical and a mind problem
  - Demotivating when nothing is progressing - and people are complaining

The 5 year outlook

- A lot of progress has been made in the last 5-10 years!
  - Going from not much to quite good
- The next 5 years look promising
  - But it won't be done tomorrow
  - Getting closer every day!

Thank You!

Any questions?


lucaweiss.eu


fosstodon.org/@z3ntu