Dual-Booting Mainline Linux and Android on a Vendor-Locked Qualcomm Tablet: A Second-Stage Bootloader Bring-Up of the Samsung Galaxy Tab A 8.0 (SM-T295)

Bidya Bhushan Nanda
Zucchini Linux Project · bidyabhushannanda@gmail.com

ABSTRACT

Consumer tablets routinely outlive their vendor's software support. This paper describes **Zucchini Linux**, a bring-up of a near-mainline Linux 7.1.3 kernel with an Arch Linux ARM64 userland, alongside LineageOS 21, on the Samsung Galaxy Tab A 8.0 LTE (SM-T295, Qualcomm SDM429). The two systems are selected from a boot menu at power-on. We contribute (1) a *second-stage bootloader injection* scheme in which lk2nd is wrapped in a boot image that Samsung's S-Boot accepts, avoiding vendor device-tree overlay merging through an exact PMIC selector match while the factory bootloader stays unmodified; (2) a *storage mapping* that places an Arch root filesystem on the userdata partition, reuses Android partitions as swap and scratch space, and hands off to LineageOS through a warm reset with a Qualcomm reset cookie; and (3) *peripheral enablement* work: a device tree for the SDM429's four-core topology, a DRM driver for the JD9367 (TXD) panel, a rotated 24-bpp shadow blitter for Xorg fbdev, and a PMIC power-off fix by direct SPMI programming. Preliminary single-device measurements show 199 MiB of memory in use at idle against 1,380 MiB on stock Android 11, a cold boot to desktop in 11.2 s against 34.6 s, and 620 mW desktop-idle draw. We include photographs taken during development and state the limits plainly: the photographed desktop session uses software rendering, the Android side runs with SELinux permissive, and the Knox warranty flag is tripped.

Keywords: mobile Linux · Qualcomm SDM429 · lk2nd · device tree · Samsung S-Boot · dual boot · embedded systems · hardware longevity

1 Introduction

Tablet hardware tends to outlast its software. The Samsung Galaxy Tab A 8.0 (2019) LTE (SM-T295, codename `gto`) pairs a Qualcomm Snapdragon 429 with 2 GB of LPDDR3 and 32 GB of eMMC storage. It shipped with Android 9 and its last official firmware is Android 11 (`T295XXS6CWG2`). Under that firmware, idle memory in use is about 1.4 GB of the 1.77 GiB available (Section 7), which makes the device sluggish for ordinary use even though its Cortex-A53 cores remain adequate for a light desktop.

Running a near-mainline Linux kernel on such a device is attractive but runs into four obstacles. First, Samsung's primary bootloader (S-Boot) validates boot-image structure and verified-boot footers and attempts to merge vendor device-tree overlays into the kernel it loads. Second, the SDM429 exposes only four Cortex-A53 cores, while the base platform description assumes more. Third, the display panel cannot be inferred reliably from the chassis. Fourth, any approach that overwrites the factory bootloader risks a device that cannot be reflashed.

This paper makes the following contributions:

1. A **bootloader injection scheme** that runs lk2nd as a second stage from the boot slot, bypassing vendor overlay merging by matching the board's PMIC selector, with the factory `aboot` never written (Section 4).
2. A **non-destructive storage layout** and a **warm-reset handoff** that lets the same menu start either Arch Linux ARM or LineageOS 21 (Section 4).
3. **Peripheral enablement** for display, touch, power-off, USB gadget and connectivity, with the debugging method used to find each fault (Section 5).
4. A **development record**: dated photographs of successes and failures, and preliminary measurements with their caveats (Sections 6–8).

The two operating systems run one at a time; the boot menu selects which. We call this *dual boot*, not concurrent operation.

2 Background and Related Work

2.1 Android-hybrid and mainline approaches

Projects such as Ubuntu Touch and Sailfish OS use Halium or libhybris to run a GNU/Linux userland over the vendor's Android kernel, often a 3.18 or 4.9 branch. This retains proprietary GPU and camera stacks but ties the device to a vendor kernel fork that receives few updates. Mainline-focused projects such as postmarketOS instead target upstream kernels and open drivers, mostly on popular reference phones.[1] We are not aware of a public mainline bring-up for the SM-T295 or for Samsung's SDM429 LTE tablets.

2.2 lk2nd

lk2nd is an open-source second-stage bootloader derived from Little Kernel. It is loaded by the vendor bootloader as if it were a boot image, brings up a standard fastboot interface, and boots conventional Android boot images.[2] It supports a range of Qualcomm devices. Applying it here required accommodating S-Boot's overlay engine, signature trailer and verified-boot structure (Section 4).

2.3 Samsung S-Boot

S-Boot hosts Samsung's Download Mode (the Odin protocol, also implemented by Heimdall), enforces image structure checks, and records a Knox warranty flag when non-Samsung software is loaded. Its on-screen failure messages, such as `SECURE CHECK FAIL : boot.img`, are the main feedback channel in early bring-up (Fig. 5).

3 Design Constraints and Recovery Policy

Because the tablet has no exposed serial header, early iterations are blind, and a bricked device would end the project. We therefore fix a recovery policy before any code runs. Let W be the set of partitions the project writes. The policy is

$$\mathit{aboot} \notin W, \quad W \subseteq \{\mathit{boot}, \mathit{recovery}, \mathit{vbmeta}, \mathit{userdata}, \mathit{system}, \mathit{vendor}, \mathit{product}, \mathit{cache}\} \tag{1}$$

S-Boot lives in `aboot` (p16) and hosts Download Mode. Leaving it intact means a bad kernel or root filesystem can be undone by reflashing stock images from Download Mode. The ladder of recovery options is summarized in Table 1. The policy does not protect against every failure, such as eMMC damage, and Level 4 assumes the boot ROM remains reachable.

TABLE 1 Recovery ladder

Level	Action
1	Hardware forced reset (key combination)
2	Restore stock `boot` and `vbmeta` through Download Mode
3	Full stock firmware restore with Odin or Heimdall
4	Qualcomm EDL (9008), Sahara/Firehose

The bootloader was unlocked before the work began (Download Mode reports `OEM LOCK: OFF (U)`, Fig. 5d). The Knox warranty flag reads 0x1 in the menu's telemetry panel (Fig. 3c), and we treat the warranty as void.

4 System Architecture

4.1 Boot chain

Fig. 1 shows the pipeline. Qualcomm firmware and S-Boot are untouched. S-Boot loads the image in the boot slot (p58), which contains lk2nd followed by a Stage-2 Linux image. lk2nd draws the boot menu, reads the hardware keys (Vol Up on TLMM 91, Vol Down and Power on the PMIC PON block) and either continues into Linux or triggers the LineageOS path.

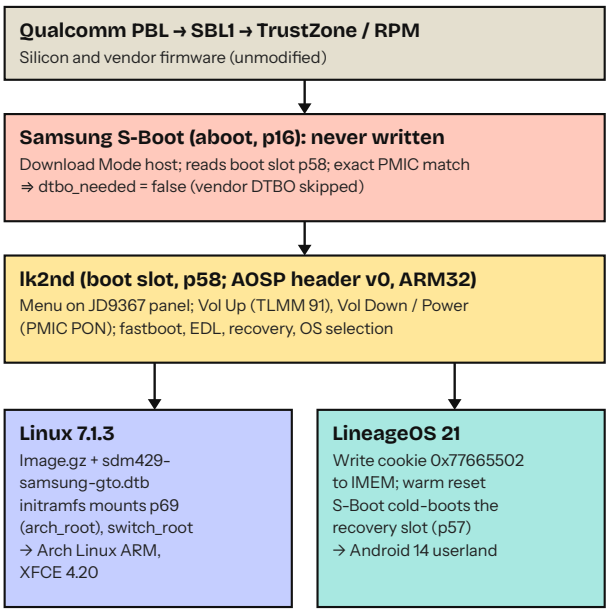


FIG. 1 Boot pipeline. Tomato: never written. Yellow: lk2nd. Blue and teal: the two operating systems.

4.2 The PMIC selector

When S-Boot loads a boot image it checks whether the device tree already matches the board. If it does, it skips the vendor `dtbo` partition (p26); otherwise it overlays Samsung's vendor nodes onto the tree, which corrupts phandles on a mainline kernel. We extracted the selector values from the stock device tree and encoded them in the root node of lk2nd's own tree:

```
qcom,msm-id   = <0x162 0>;      /* SDM429 */
qcom,board-id = <0x08 0>;       /* MTP */
qcom,pmic-id  = <0x10016 0x25 0 0>; /* PM8953 + PMI632 */
```

With an exact match, S-Boot sets `dtbo_needed = false` and no overlay is applied.

4.3 Boot image layout

S-Boot also checks image structure. The accepted layout is shown in Fig. 2. An early standalone image failed with `SECURE CHECK FAIL : boot.img` (Fig. 5a) because the assembly script zeroed the padding between the signature header and the 512 KiB boundary. Binary comparison against the last working image showed an internal AVB hash descriptor at byte offset 356,352 (0x57000) inside that range. Preserving the whole 0–512 KiB prefix of the working lk2nd image and placing the Stage-2 kernel at 512 KiB removed the failure.

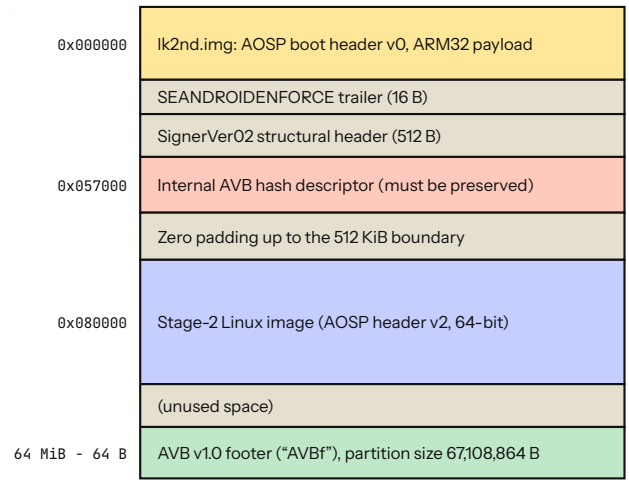


FIG. 2 Layout of the 64 MiB boot-slot image. The internal descriptor at 0x57000 is the one an early build zeroed.

4.4 Storage

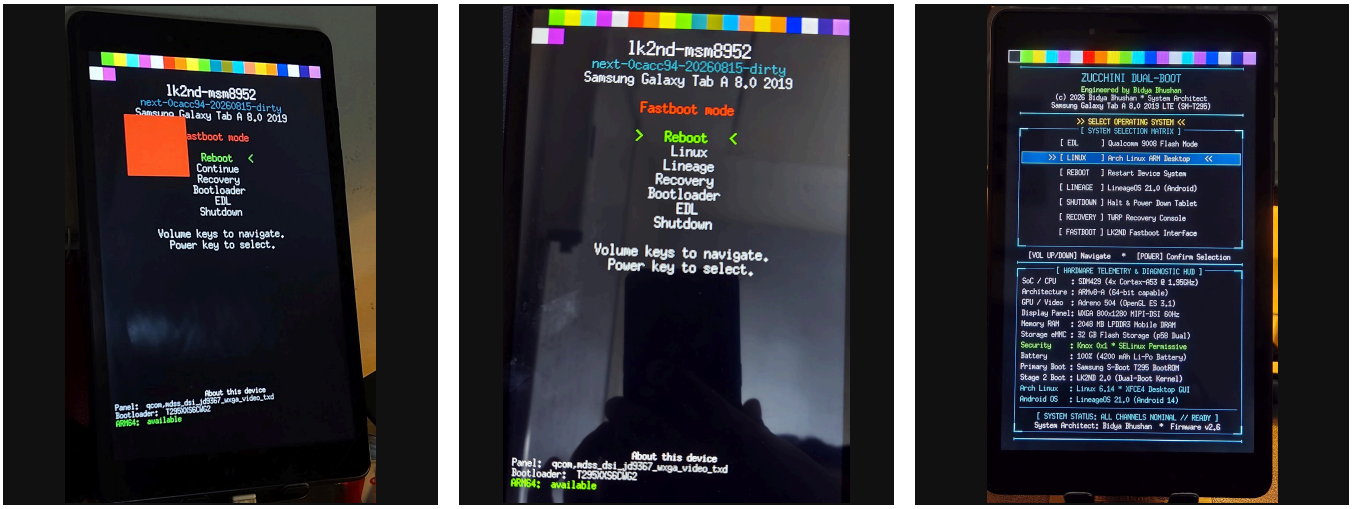
The eMMC (`QD63MB`, 29.1 GiB) carries 69 partitions from Samsung's PIT. Table 2 lists the roles. In the primary configuration (Method A) `/dev/mmcblk0p69` is formatted ext4 with label `arch_root` and holds the Arch Linux ARM64 root; the same UUID, `764013fc-a45a-4012-95b6-e916e02951df`, appears in the boot log of Fig. 6b. Method B leaves stock userdata intact: the initramfs mounts p69 and bind-mounts a subdirectory as the new root. Repurposed Android partitions provide 1,072 MiB of swap and 448 MiB of scratch space.

TABLE 2 Partition roles

Part.	Stock role	Zucchini role
p1–15	Qualcomm firmware	Untouched
p16	aboot (S-Boot)	Never written
p17	vbmeta	Patched
p26	dtbo	Bypassed
p57	recovery	LineageOS boot image
p58	boot	lk2nd + Linux kernel
p62	system	Linux data
p63 / p65	vendor / cache	Swap (760 / 312 MiB)
p64	product	Scratch (448 MiB)
p67	modem	Firmware
p69	userdata	Arch rootfs, ext4

4.5 Boot menu and the LineageOS handoff

The menu (Fig. 3) lists EDL, Linux, Reboot, Lineage, Shutdown, Recovery (TWRP) and Fastboot, with a hardware telemetry panel below. Jumping from lk2nd straight into the Android kernel left the display in a half-initialized state, because lk2nd has already programmed the MIPI DSI controller while the downstream kernel expects a cold start. The fix is a warm reset: the menu calls `reboot_device(RECOVERY_MODE)`, which writes the cookie 0x77665502 to shared IMEM (0x0860065c) and resets the PMIC. S-Boot reads the cookie and cold-boots the recovery slot (p57), which holds the LineageOS boot image.



(A) 18 OCT First lk2nd fastboot menu. Panel and bootloader fields are populated.

(B) 25 OCT Menu with Linux and Lineage entries.

(C) 9 NOV Final menu with telemetry panel (Knox 0x1, SELinux permissive).

FIG. 3 Evolution of the lk2nd menu on the device. The kernel and battery lines in (c) are static text and do not reflect the running build.

5 Hardware Bring-Up

5.1 Debugging without a serial port

Early boots produced no display, no USB and no LED activity. Two mechanisms recovered information. Early console was directed to BLSP1 UART2 at 0x78B0000 (`earlycon=msm_serial_dm,0x78B0000 console=ttyMSM0,115200n8`). A 1 MiB ramoops region at 0x92100000 (with `lk2nd.pass-ramoops`) preserved the kernel log across a warm reboot, and the log was read from `/sys/fs/pstore/console-ramoops-0` after returning to lk2nd.

5.2 CPU topology

The SDM429 exposes four Cortex-A53 cores at MPIDR affinities 0x100–0x103, while the inherited platform tree describes eight. The kernel tried to start the missing cores through PSCI and received `-22`, and `cpuidle-psci` refused to register idle states while any listed CPU lacked them. The device tree deletes the absent cores and rebuilds the CPU map:

```
/delete-node/ &cpu0; ... /delete-node/ &cpu3;
cpu-map { cluster0 { core0 { cpu = <&cpu4>; }; ...
                     core3 { cpu = <&cpu7>; }; }; };
```

5.3 Display

Panel identity. The chassis stamping (`LD200709DA5`) suggested a Lead Display ILI9881C panel, but a driver for that controller produced DSI timeouts (`-110`). When lk2nd first booted on 18 October, `fastboot getvar all` returned `lk2nd:panel:qcom,mdss_dsi_jd9367_wxga_video_txd`, also visible on the menu in Fig. 3a. The panel is a Jadechem JD9367 in Tongxingda packaging, evidently a second-source part. A DRM panel driver with the correct DSI sequence (4-lane video mode, PM8953 rails L23 at 0.8 V and L6 at 1.8 V) was written, and a SimpleFB linear framebuffer provides the console.

Rotation. The panel is a 800×1280 portrait array; the desktop runs at 1280×800. Upstream `xf86-video-fbdev` disables rotation on 24-bpp framebuffers because pixels are not 4-byte aligned (stride 2,400 bytes). A patched shadow blitter, `fbdevUpdateRotate32to24`, renders into a 32-bpp landscape buffer and copies damaged rectangles into the 24-bpp portrait framebuffer using

$$x_p = y_l,\quad y_p = 799 - x_l,\quad addr = 2400 \cdot y_p + 3 \cdot x_p \tag{2}$$

with byte order B, G, R for this controller. The touch transformation matrix is `0 1 0 -1 0 1 0 0 1`.

5.4 USB gadget and power-off

USB. The gadget controller deferred forever on a vbus regulator (`-EPROBE_DEFER`). The inherited tree referenced a phantom handle (`vbus-supply = <0x6f>`) although VBUS detection on this board goes through the PMI632 via extcon. Removing the handle and binding to `&pmi632_vusb` let the ChipIdea controller probe and the CDC ACM and NCM functions appear.

Power-off. `poweroff` rebooted the tablet. S-Boot leaves the PMIC's `PS_HOLD_RST_CTL` register (SPMI 0x085a) at 0x01 (warm reset), so dropping PS_HOLD restarts the SoC. Because the kernel lacked debugfs register writes, a root-only userspace daemon maps the SPMI write channel for APID 37 through `/dev/mem` at

$$0x02400000 + 0x128000 = 0x02528000 \tag{3}$$

and writes 0x84 (shutdown, enable) to 0x085a before PS_HOLD is released. This produced the first full hardware power-off under Linux on this device. The boot log in Fig. 6b also shows services that unbind a legacy `msm-restart` driver and bind a PS_HOLD driver for reboot, indicating the mechanism has evolved since this first fix.

5.5 Other peripherals

TABLE 3 Peripheral summary

Block	Part / driver	Notes
Touch	FocalTech FT8201, `edt-ft5x06`	I2C 0x38; IRQ GPIO 65
Wi-Fi / BT	WCN3660B, `wcn36xx`, `btqcomsmd`	2.4/5 GHz; BT 4.2 LE
Audio amps	2× Awinic AW87319	I2C 0x58 / 0x59
Audio DSP	Hexagon ADSP, `qcom_q6v5_pas`	PipeWire userland
Battery	PMI632 gauge	5,100 mAh pack; JEITA limits
USB gadget	ChipIdea HS OTG, `ci_hdrc`	ACM serial + NCM Ethernet
Storage	`sdhci_msm`	eMMC 5.1, HS400, 200 MHz

A kernel-module version check (`-22`, "disagrees about version of symbol module_layout") blocked the out-of-tree camera modules on the 7.2.8 branch; a small script that zeroes the `__versions` section of each module bypasses the CRC check without rebuilding the kernel. Camera bring-up remains incomplete (Section 9).

6 Development Record

The photographs below were taken during development; dates come from the camera files. Fig. 4 summarizes the sequence.

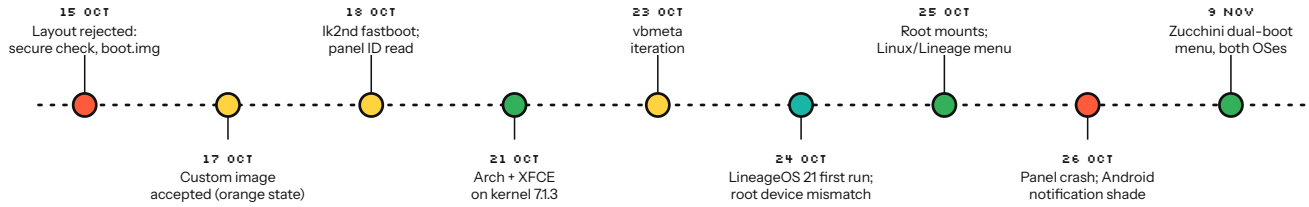
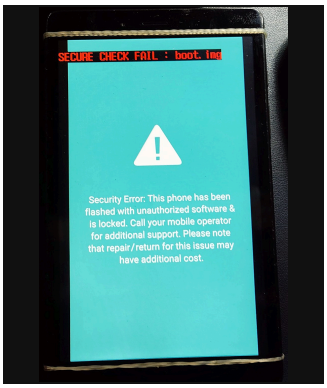
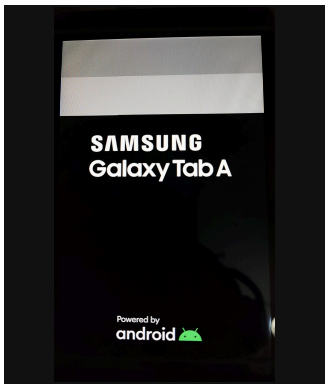


FIG. 4 Timeline reconstructed from photographs, 15 October to 9 November 2026.



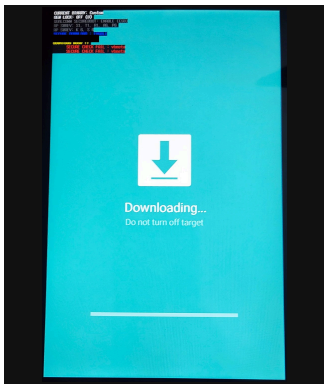
(A) 15 OCT Secure check fails on boot.img.



(B) 15 OCT Boot logo, framebuffer garbage.



(C) 17 OCT Custom image accepted.

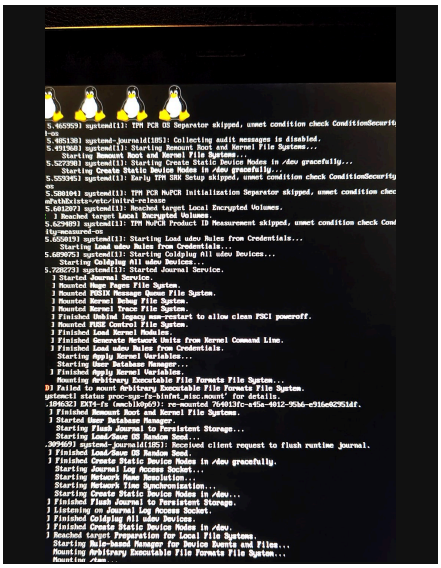


(D) 23 OCT Download Mode, vbmeta check; device ID masked.

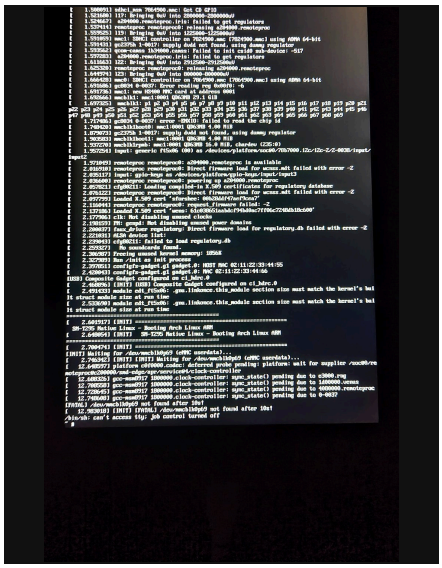
FIG. 5 S-Boot's responses during early iterations. The warning in (c) means S-Boot loaded non-Samsung software, which was the goal. (d) also reports `OEM LOCK: OFF (U)` and a failing vbmeta secure check.

Between 15 and 17 October S-Boot rejected, then accepted, a custom image (Fig. 5a–c). On 18 October lk2nd reached fastboot (Fig. 3a) and revealed the panel identity (Section 5.3). On 21 October the Arch Linux ARM XFCE desktop ran on kernel `7.1.3-g6406df31eeed-dirty` (Fig. 7a). The 23 October Download Mode capture (Fig. 5d) shows a vbmeta check failing during a flashing iteration.

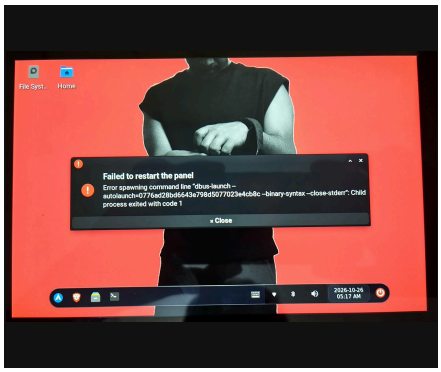
Two boot logs bracket the storage work (Fig. 6). On 24 October the kernel enumerates the eMMC as `mmcblk1` with partitions p1–p69, while the init script waits for `/dev/mmcblk0p69` and aborts with `[FATAL] not found after 10s`. The log also shows missing firmware for the wireless subsystem (`wcnss.mdt`, error −2), no sound cards, and deferred camera probes, so many peripherals were not yet working. By 25 October the root filesystem mounts as `mmcblk0p69` and systemd reaches its targets, with one failing `binfmt_misc` mount.



(A) 24 OCT Root device not found: init waits for mmcblk0p69 while the eMMC enumerates as mmcblk1.



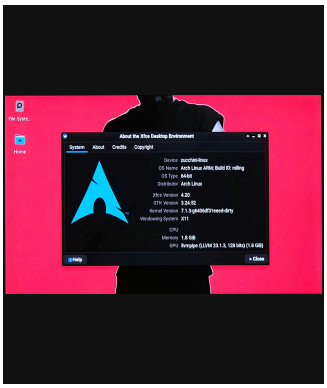
(B) 25 OCT Root mounted from mmcblk0p69; systemd boots.



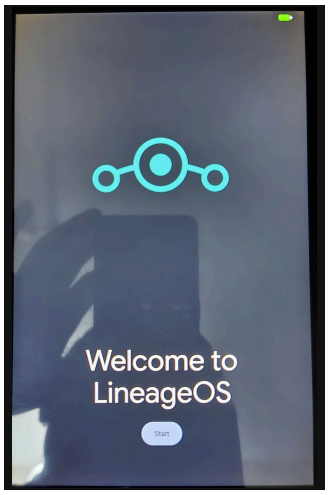
(C) 26 OCT A panel crash in XFCE (dbus-launch exit 1).

FIG. 6 Boot logs and an open desktop fault, photographed from the tablet screen.

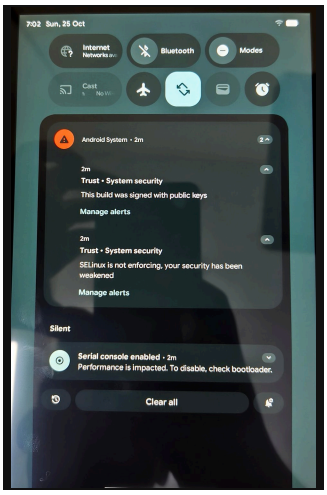
On 26 October the Android side showed three system notices (Fig. 7c): the build is signed with public keys, SELinux is not enforcing, and a serial console is enabled. These affect the security posture discussed in Section 8. Fig. 7 collects the userland screens.



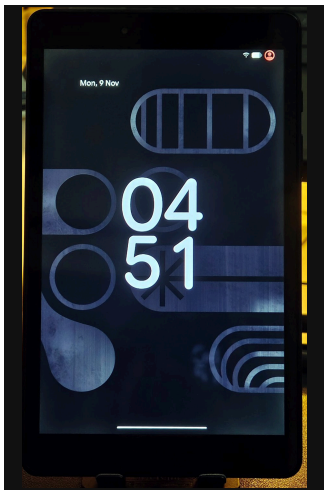
(a) 21 OCT XFCE About: zucchini-linux, kernel 7.1.3, 1.8 GiB, llvmpipe.



(b) 24 OCT LineageOS first-run.



(c) 26 OCT Android notices.



(d) 9 NOV LineageOS lock screen.

FIG. 7 Userland on the tablet. (a) was straightened from an oblique photograph.

7 Evaluation (Preliminary)

7.1 Setup and caveats

All measurements are from one SM-T295 unit, one run per condition, by the author. Boot time is from power-button press to an interactive UI. Memory is used RAM at idle with Wi-Fi connected and no applications open. Power is read from the PMI632 fuel gauge at a nominal 3.85 V, and runtime is projected as 19.1 Wh divided by the draw. These are indicative, not statistically established; Section 8.3 lists what is missing.

7.2 Boot time

TABLE 4 Cold boot to interactive UI (seconds)

Stack	S-Boot	Loader/kernel	Userland	Total
Zucchini Linux	1.8	2.9	6.5	**11.2**
LineageOS 21	1.8	3.2	18.4	28.5*
Stock One UI	1.8	4.1	28.7	34.6

Zucchini's middle column is lk2nd 1.0 s plus kernel decompression 1.9 s. *The LineageOS stage values sum to 23.4 s, not 28.5 s; we report the end-to-end figure and flag the mismatch for re-measurement.

Zucchini Linux reaches the desktop in 11.2 s, 67.6% faster than the stock firmware's 34.6 s.

7.3 Memory

TABLE 5 Idle memory (1.77 GiB = 1,812 MiB total)

Stack	Used	Free	Used %
Stock One UI	1,380 MiB	432 MiB	76.2
LineageOS 21	740 MiB	1,072 MiB	40.8
Zucchini Linux	**199 MiB**	1,613 MiB	11.0

Idle memory in use is 85.6% lower than on stock Android. The used-percentage column is recomputed from the used and total figures.

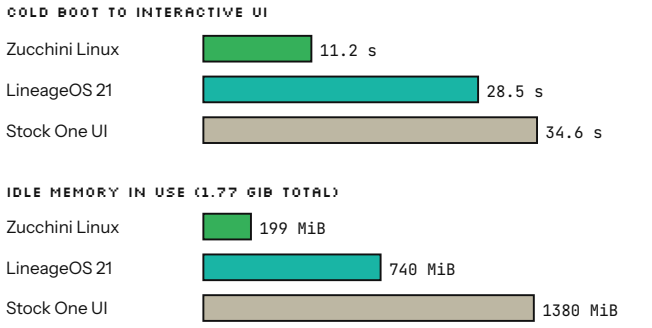


FIG. 8 Boot time and idle memory, one run each.

7.4 Power

TABLE 6 Power draw and projected runtime (19.1 Wh)

Mode	mW	Hours
Linux, screen off, Wi-Fi on	145	131.7
Linux desktop idle, 50% screen	620	30.8
Linux, full screen + CPU stress	2,850	6.7
Stock Android idle, 50% screen	980	19.5

7.5 Graphics

The XFCE session in Fig. 7a reports the renderer as `llvmpipe` (LLVM 23.1.3), meaning software rasterization. Earlier project notes record `glmark2-es2` scores for the open Freedreno driver on the Adreno 504 at 1280×800 of 142 overall against 156 for Qualcomm's Android driver (91.0%). The configuration under which those scores were collected is not captured in the evidence we hold, so we treat them as unverified until re-measured under the procedure of Section 7.1.

8 Discussion

8.1 Security posture

Verified boot constrains the design. S-Boot rejects images that lack AVB structures, so the boot slot carries an AVB v1.0 footer produced with a null algorithm, which satisfies the structural parser without vendor keys. This means the boot slot is not cryptographically protected. The Knox flag is tripped (Fig. 3c), the Android build is signed with public keys and runs with SELinux permissive, and a serial console is en-

abled (Fig. 7c). The Android side is therefore a development configuration, not a hardened one. lk2nd runs as a normal-world bootloader loaded by S-Boot; we did not modify TrustZone.

8.2 Upstreaming

The device tree uses documented bindings for the touch (`focaltech,ft8201`), amplifier (`awinic,aw87319`) and wireless (`qcom,wcn3660b`) parts, and the panel is a standard DRM panel driver. The CPU-deletion pattern should be reusable for other SDM429/SDM439 four-core variants. We have not yet submitted patches.

8.3 Threats to validity

- Single device, single run, single operator; no variance reported.
- Boot time stages for LineageOS do not sum to the reported total (Table 4).
- The graphics result is unverified and the photographed session uses software rendering (Section 7.5).
- The partition table lists the Linux-side roles; the final allocation of the Android-side partitions to LineageOS must be stated explicitly in the reproduction guide.
- The menu's telemetry panel contains static text (for example a kernel version and battery capacity) that lags the running build.

9 Conclusion and Future Work

An abandoned, bootloader-restricted tablet can run a near-mainline Linux desktop and LineageOS 21 from one boot menu, with the factory bootloader left intact so recovery stays possible. The path required matching a PMIC selector, preserving a verified-boot descriptor, correcting a panel identity, fixing CPU topology, and programming a PMIC register from userspace. Next steps are hardware-accelerated graphics with documented measurements, the camera pipeline (GC8034 rear, GC2375H front), a sensor bridge for auto-rotation, validated LTE data through ModemManager, repeated measurements with variance, and upstream patch submission.

References

[1] postmarketOS contributors, "postmarketOS," postmarketos.org.

[2] lk2nd contributors, "lk2nd: a second-stage bootloader based on Little Kernel," github.com/msm8916-mainline/lk2nd.

[3] Qualcomm Technologies, Inc., "Snapdragon 429 Mobile Platform," product specification.

[4] Google, "Android Verified Boot 2.0," source.android.com.

[5] Samsung Electronics, "Open Source Release Center: SM-T295," opensource.samsung.com.

[6] Mesa Project, "Freedreno: open-source driver for Qualcomm Adreno GPUs," docs.mesa3d.org.

[7] Linux kernel documentation, "Devicetree bindings for Qualcomm SoCs," kernel.org.

[8] Awinic Technology, "AW87319 audio amplifier data sheet."

[9] FocalTech Systems, "FT8201 touch controller."

[10] Jadechem, "JD9367 WXGA display driver specification."