About us

- ellyq (@elly@donotsta.re)
  - Systems Engineer by day, hardware hacker by night
  - Contributing to upstream Linux kernel, Fedora, postmarketOS, OpenWrt, Coreboot etc.
  - One of three admins in Chrultrabook community, partially managing infra
  - Has a pile of hacked Chromebooks running Linux
  - You might know me from porting Coreboot to a funny desktop motherboard with laptop ES SoC (erying)
- sdomi (@domi@donotsta.re)
  - toyed with coreboot ever since TP X230 was relevant
  - nerdsniped by elly into chromebook hacking
  - you may know me from making a Minecraft server in Bash
  - into making the most cursed computer stuff you’ve ever heard of

What are those ChromeOS devices, really?

- Devices created according to Google’s standards and guidelines
  - Specific Embedded Controllers, Codecs, Amplifiers…
  - Open-source firmware (EC, FPMCU, Platform) maintained by Google

What are those ChromeOS devices, really?

- Devices created according to Google’s standards and guidelines
  - Specific Embedded Controllers, Codecs, Amplifiers…
  - Open-source firmware (EC, FPMCU, Platform) maintained by Google
- Wide range of hardware:
  - Low-end: Cheap and durable machines with low-end SoCs for students
  - Mid-range: General use, with Pentiums or i3’s and eMMC/NVME
  - High-end: Core i7/Ryzen 7, NVME, 16:10, AMOLED…

What are those ChromeOS devices, really?

- Devices created according to Google’s standards and guidelines
  - Specific Embedded Controllers, Codecs, Amplifiers…
  - Open-source firmware (EC, FPMCU, Platform) maintained by Google
- Wide range of hardware:
  - Low-end: Cheap and durable machines with low-end SoCs for students
  - Mid-range: General use, with Pentiums or i3’s and eMMC/NVME
  - High-end: Core i7/Ryzen 7, NVME, 16:10, AMOLED…
- Generally considered “e-waste” by most people
  - Year-old machines can be found for half the price
  - Machines decommisioned from schools are flooding the used market
    - GeminiLake devices can be found for ~50EUR
    - TigerLake/Ryzen devices can be found for ~260EUR

What are those ChromeOS devices, really?

- Devices created according to Google’s standards and guidelines
  - Specific Embedded Controllers, Codecs, Amplifiers…
  - Open-source firmware (EC, FPMCU, Platform) maintained by Google
- Wide range of hardware:
  - Low-end: Cheap and durable machines with low-end SoCs for students
  - Mid-range: General use, with Pentiums or i3’s and eMMC/NVME
  - High-end: Core i7/Ryzen 7, NVME, 16:10, AMOLED…
- Generally considered “e-waste” by most people
  - Year-old machines can be found for half the price
  - Machines decommisioned from schools are flooding the used market
    - GeminiLake devices can be found for ~50EUR
    - TigerLake/Ryzen devices can be found for ~260EUR
- Great debugging tools, a ton of the firmware code is open

What’s ChromeOS?

- Linux kernel w/ patches, but contrary to popular belief,

What’s ChromeOS?

- Linux kernel w/ patches, but contrary to popular belief, **not Gentoo**

What’s ChromeOS?

- Linux kernel w/ patches, but contrary to popular belief, **not Gentoo**
- Mostly 64-bit, except for budget ARM devices

What’s ChromeOS?

- Linux kernel w/ patches, but contrary to popular belief, **not Gentoo**
- Mostly 64-bit, except for budget ARM devices
- Upstart as an init system, SELinux, fully custom userspace

What's ChromeOS?

- Linux kernel w/ patches, but contrary to popular belief, **not Gentoo**
- Mostly 64-bit, except for budget ARM devices
- Upstart as an init system, SELinux, fully custom userspace
- Some cool tech made especially for cros:
  - Crostini (Linux containers)
  - ArcVM (Android compat layer)
  - CRAS (ChromeOS Audio Server), as a replacement for PulseAudio/PipeWire
  - BioD (Userspace fingerprint driver), as a replacement for LibFprint

Enter: coreboot

Coreboot: what’s this?

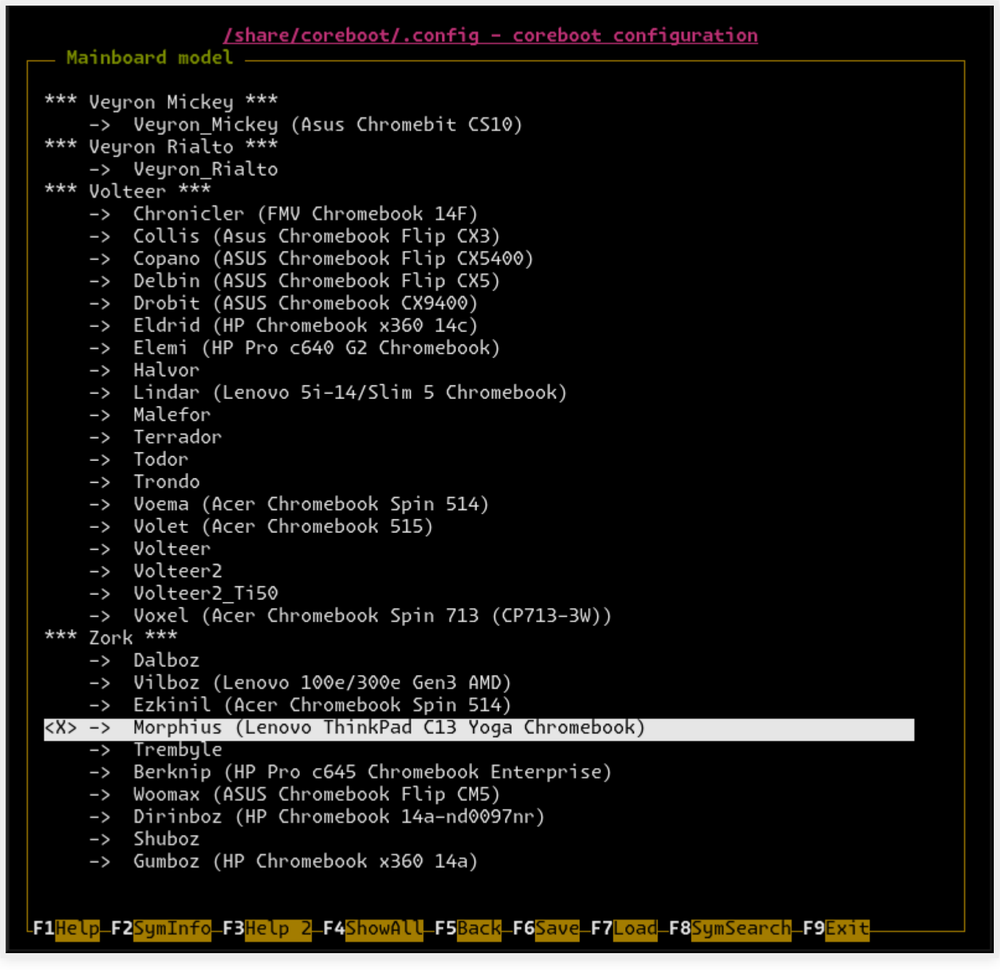
- Open-source firmware implementation
  - similar to U-Boot and EDK2; in fact, it supports embedding them as payloads

Coreboot: what’s this?

- Open-source firmware implementation
  - similar to U-Boot and EDK2; in fact, it supports embedding them as payloads
  - as seen on: obsolete ThinkPads and many, many other devices

Coreboot: what’s this?

- Open-source firmware implementation
  - similar to U-Boot and EDK2; in fact, it supports embedding them as payloads
  - as seen on: obsolete ThinkPads and many, many other devices
- ChromeOS devices ship with a version of coreboot by default!
  - Google upstreams their code! See the “Google” manufacturer!



“but google uses coreboot, what’s the point of your project?”

“but google uses coreboot, what’s the point of your project?”

Google’s Coreboot != Coreboot

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left
  - Not really useful for us

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left
  - Not really useful for us
    - depthcharge-tools by alpernebbi makes handling it easier, but it’s still bespoke

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left
  - Not really useful for us
    - depthcharge-tools by alpernebbi makes handling it easier, but it’s still bespoke
- Non-compliant ACPI, resulting in broken USB, audio, input devices, sleep…

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left
  - Not really useful for us
    - depthcharge-tools by alpernebbi makes handling it easier, but it’s still bespoke
- Non-compliant ACPI, resulting in broken USB, audio, input devices, sleep…
- Mainline kernel isn’t tested well, google takes a long time to fix bugs, etc.

- Google uses depthcharge as a payload; it requires ChromeOS partition scheme, kernels…
  - Depthcharge used to be a fork of U-Boot, but there’s not much of the original code left
  - Not really useful for us
    - depthcharge-tools by alpernebbi makes handling it easier, but it’s still bespoke
- Non-compliant ACPI, resulting in broken USB, audio, input devices, sleep…
- Mainline kernel isn’t tested well, google takes a long time to fix bugs, etc.
- Google also offers AltFW (alternative payloads) on x86:
  - SeaBIOS on ApolloLake and older
  - EDK2 on GeminiLake and newer
  - U-Boot on AMD StoneyRidge

    … while neat, AltFW doesn’t fix firmware issues and non-functioning SMMSTORE (NVRAM)

Importance of open firmware

Importance of open firmware

- Your device doesn’t work because of buggy firmware?

Importance of open firmware

- Your device doesn’t work because of buggy firmware? **YOU CAN FIX IT!**

Importance of open firmware

- Your device doesn’t work because of buggy firmware? **YOU CAN FIX IT!**
- Firmware updates have a lag because of the manufacturer -> vendor -> you pipeline (i.e. BIOS updates)

Importance of open firmware

- Your device doesn’t work because of buggy firmware? **YOU CAN FIX IT!**
- Firmware updates have a lag because of the manufacturer -> vendor -> you pipeline (i.e. BIOS updates)
- We can iron out the edge cases, making device support much better than it otherwise would have been

Importance of open firmware

- Your device doesn’t work because of buggy firmware? **YOU CAN FIX IT!**
- Firmware updates have a lag because of the manufacturer -> vendor -> you pipeline (i.e. BIOS updates)
- We can iron out the edge cases, making device support much better than it otherwise would have been
- Add your own reason: so many ways to do new, cool things
  - This project is by no means a finished endeavour: help appreciated :)

Embedded Controller: also “open”!

Embedded Controller: also “open”!

- Open-source, but hard to build; requires Chromium chroot for a specific board
- Recent (2022?) machines use EC Firmware based on ZephyrOS

Embedded Controller: also “open”!

- Open-source, but hard to build; requires Chromium chroot for a specific board
- Recent (2022?) machines use EC Firmware based on ZephyrOS
- We can fix bugs, but the overall experience is not as streamlined as with coreboot
- Somewhat bespoke, lacking good docs

Embedded Controller: also “open”!

- Open-source, but hard to build; requires Chromium chroot for a specific board
- Recent (2022?) machines use EC Firmware based on ZephyrOS
- We can fix bugs, but the overall experience is not as streamlined as with coreboot
- Somewhat bespoke, lacking good docs
- There are other vendors using this Embedded Controller (i.e: Framework)

```
        /* A small number of non-Chromebook/box machines also use the ChromeOS EC */
        {
                /* the Framework Laptop */
                .matches = {
                        DMI_MATCH(DMI_SYS_VENDOR, "Framework"),
                        DMI_MATCH(DMI_PRODUCT_NAME, "Laptop"),
                },
        },
        { /* sentinel */ }
};
MODULE_DEVICE_TABLE(dmi, cros_ec_lpc_dmi_table);
```

Making Chromebooks run alternative OSes - Chrultrabook Project:

Making Chromebooks run alternative OSes - Chrultrabook Project:

“let’s fix google’s stuff!”

Chrultrabook Project

Coreboot stuff:

Chrultrabook Project

Coreboot stuff:

- Patching ACPI tables

Chrultrabook Project

Coreboot stuff:

- Patching ACPI tables
- Patching platform-specific code (i.e PWM backlight, TCSS controller on TGL/ADL)

Chrultrabook Project

Coreboot stuff:

- Patching ACPI tables
- Patching platform-specific code (i.e PWM backlight, TCSS controller on TGL/ADL)
- Power sequencing for touchscreens and other i^2c devices

Chrultrabook Project

Coreboot stuff:

- Patching ACPI tables
- Patching platform-specific code (i.e PWM backlight, TCSS controller on TGL/ADL)
- Power sequencing for touchscreens and other i^2c devices
- Debugging EDK2 (UEFI payload), extending features:
  - Added support for SecureBoot and PXE in release 4.20
  - Debugging NVME/eMMC/USB enumeration
  - GOP (display initialization)

Chrultrabook Project

Coreboot stuff:

- Patching ACPI tables
- Patching platform-specific code (i.e PWM backlight, TCSS controller on TGL/ADL)
- Power sequencing for touchscreens and other i^2c devices
- Debugging EDK2 (UEFI payload), extending features:
  - Added support for SecureBoot and PXE in release 4.20
  - Debugging NVME/eMMC/USB enumeration
  - GOP (display initialization)
- Lots and lots of bugfixing and debugging overall, which also benefits other systems running Coreboot

Chrultrabook Project

Linux kernel stuff:

Chrultrabook Project

Linux kernel stuff:

- Drivers/platform/chrome was missing custom ACPI definitions

Chrultrabook Project

Linux kernel stuff:

- Drivers/platform/chrome was missing custom ACPI definitions
- Google’s code usually wasn’t well tested w/ upstream Linux + we frequently have to fix regressions

Chrultrabook Project

Linux kernel stuff:

- Drivers/platform/chrome was missing custom ACPI definitions
- Google’s code usually wasn’t well tested w/ upstream Linux + we frequently have to fix regressions
- Many small workarounds for specific vendors
  - Intel’s PMC MUX in TGL doesn’t work with Chrome platforms without a workaround; we’re working on upstreaming our fix.

Chrultrabook Project

Linux kernel stuff:

- Drivers/platform/chrome was missing custom ACPI definitions
- Google’s code usually wasn’t well tested w/ upstream Linux + we frequently have to fix regressions
- Many small workarounds for specific vendors
  - Intel’s PMC MUX in TGL doesn’t work with Chrome platforms without a workaround; we’re working on upstreaming our fix.
- CrosEC-specific hooks: sensors, USB-C, i^2c on some platforms

Chrultrabook Project

Linux kernel stuff:

- Drivers/platform/chrome was missing custom ACPI definitions
- Google’s code usually wasn’t well tested w/ upstream Linux + we frequently have to fix regressions
- Many small workarounds for specific vendors
  - Intel’s PMC MUX in TGL doesn’t work with Chrome platforms without a workaround; we’re working on upstreaming our fix.
- CrosEC-specific hooks: sensors, USB-C, i^2c on some platforms
- Fixing audio: all post-2016 machines had broken audio for *years*

chrultrabook: audio

- Most devices use unusual CODECs, DSP chips, etc.

chrultrabook: audio

- Most devices use unusual CODECs, DSP chips, etc.
  - Nowadays: good kernel support, mediocre distro support
    - We’re working with alsa-devel folks on new AVS drivers
    - Most distros are helping by enabling necessary drivers; Ubuntu, Debian and their derivatives are an exception

chrultrabook: audio

- Most devices use unusual CODECs, DSP chips, etc.
  - Nowadays: good kernel support, mediocre distro support
    - We’re working with alsa-devel folks on new AVS drivers
    - Most distros are helping by enabling necessary drivers; Ubuntu, Debian and their derivatives are an exception
  - Devices using DSP (SOF, AVS) require firmware blobs and custom topologies

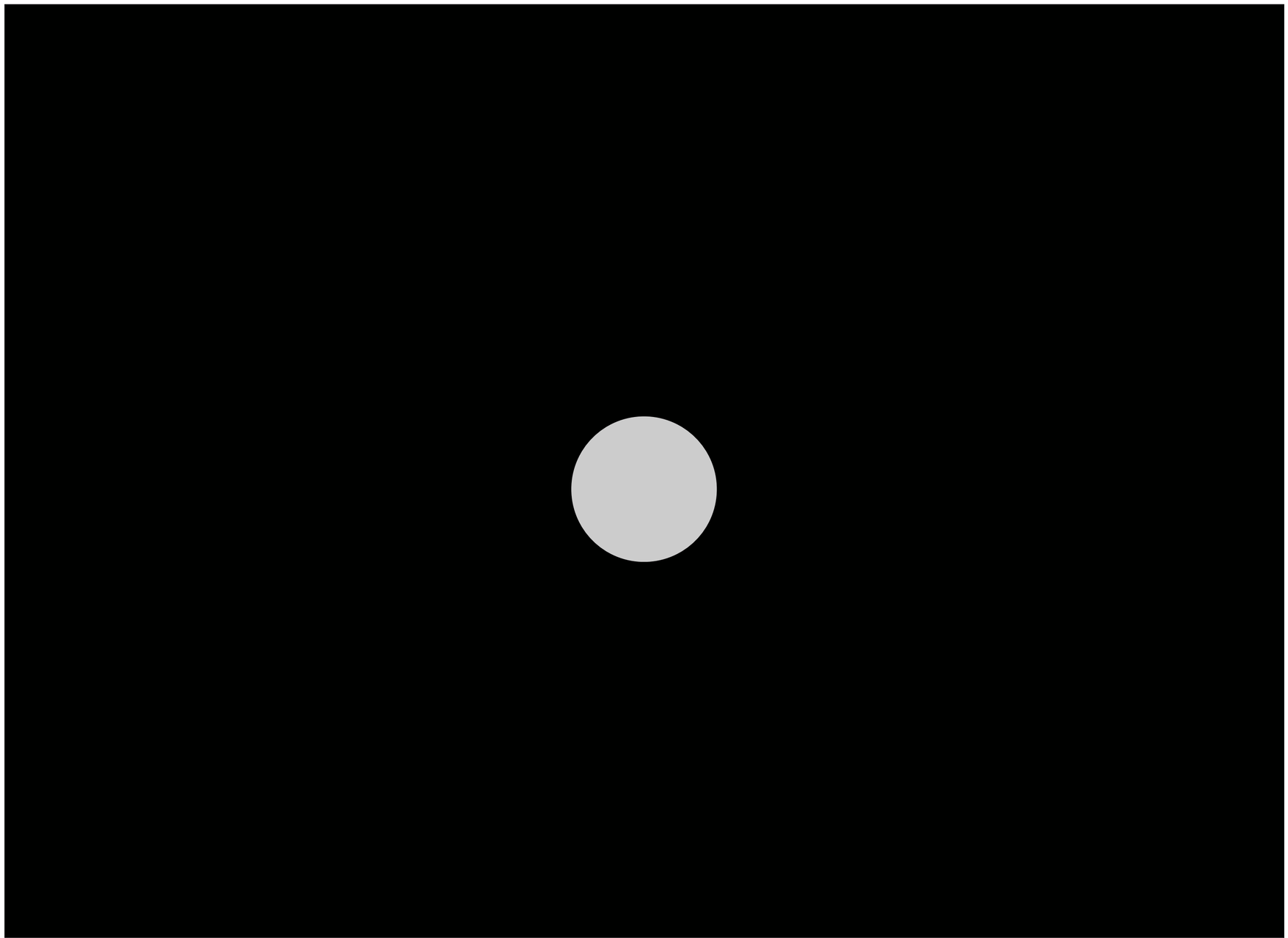
chrultrabook: audio

- Most devices use unusual CODECs, DSP chips, etc.
  - Nowadays: good kernel support, mediocre distro support
    - We’re working with alsa-devel folks on new AVS drivers
    - Most distros are helping by enabling necessary drivers; Ubuntu, Debian and their derivatives are an exception
  - Devices using DSP (SOF, AVS) require firmware blobs and custom topologies
- ALSA Use-Case Manager
  - ChromeOS has their own UCM, but it’s not a drop-in replacement
  - We wrote our own configs (soon to be upstreamed)

chrultrabook: audio

- Most devices use unusual CODECs, DSP chips, etc.
  - Nowadays: good kernel support, mediocre distro support
    - We’re working with alsa-devel folks on new AVS drivers
    - Most distros are helping by enabling necessary drivers; Ubuntu, Debian and their derivatives are an exception
  - Devices using DSP (SOF, AVS) require firmware blobs and custom topologies
- ALSA Use-Case Manager
  - ChromeOS has their own UCM, but it’s not a drop-in replacement
  - We wrote our own configs (soon to be upstreamed)
- AVS limits (speaker safety)
  - Google implemented them in userspace, not firmware
  - We’re trying to do the opposite, as we don’t control userspace
  - This still isn’t finished, but we’re working on it

“meh, it’ll be fine” - AVS w/o limits



It’s not all FOSS: x86_64 edition

- Microcode (specific for SoC stepping)
- FSP (Firmware Support Package), both S and M (silicon, memory)
  - FSP provides well-documented API for controlling variables (at least in Intel’s case)
  - Not open-source, although AMD is working on it.
    - There’s some initial OpenSIL code in Coreboot 4.22
- Signed PSP Verstage (AMD), Management Engine (Intel)
  - PSP FW runs on an ARMv7 LE core
- VBT (Intel), VBIOS (AMD)
- GOP (Graphics Option ROM) - in some cases can be done by FSP

It’s not all FOSS: ARM64 edition

- Mediatek:
  - MT8183, MT8186: DRAM, PCM, SSPM
  - MT8188, MT8192, MT8195: DRAM, DPM, MCUPM, SPM, SSPM
    - DRAM.ELF - memory training, saves results to specified region in SPI flash
    - PM - Power Management
    - MCUPM (codename “PICACHU”) - FreeRTOS 10.1.0
- Runtime FW (i.e SCP on Kukui): Video Decoding/Encoding, Digital Image Processing, Cameras, possibly USB-C DP AltMode

It’s not all FOSS: ARM64 edition

- Qualcomm:
  - AOP (Always-On Processor) - kinda like ME/PSP?
  - DCB - DRAM controller/memory training
  - PMIC, QcLib - PMIC, GPIO, SoC clocks, ROMStage interface
  - QCLib - DDR training data
  - LibQTiSec - AR archive, bunch of ELFs inside. Possibly Secure Monitor?
  - QUP (GSI, I2C, SPI, UART) - GSI seems to be modem-related
  - CPUCP - DCVS (Dynamic Clock Voltage Scaling)
  - SHRM - Yet another DDR “training” - provides QD-UTT interface for debugging?

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	random Chromebook

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	random Chromebook
Open BIOS firmware	✅	✅

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	random Chromebook
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌
Special tools required to flash	ch341a / Pi Pico / RPi; SOIC-8 / SOIC-16 clip or a soldering iron	worst case: a paperclip; usually nothing

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌
Special tools required to flash	ch341a / Pi Pico / RPi; SOIC-8 / SOIC-16 clip or a soldering iron	worst case: a paperclip; usually nothing
Bonus: available new?	❌ used, aging hardware	✅

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌
Special tools required to flash	ch341a / Pi Pico / RPi; SOIC-8 / SOIC-16 clip or a soldering iron	worst case: a paperclip; usually nothing
Bonus: available new?	❌ used, aging hardware	✅
Bonus: reasonably fast?	❌ (fastest ever X230 has similar performance to some high-end chromebooks, but it gets to 100°C Tdie under load)	✅

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌
Special tools required to flash	ch341a / Pi Pico / RPi; SOIC-8 / SOIC-16 clip or a soldering iron	worst case: a paperclip; usually nothing
Bonus: available new?	❌ used, aging hardware	✅
Bonus: reasonably fast?	❌ (fastest ever X230 has similar performance to some high-end chromebooks, but it gets to 100°C Tdie under load)	✅
Bonus: price?	surprisingly high for a 10+ year old device; corebooted ones go for big $$$	can be had for cheap or free

“Laptop respecting your privacy” comparison

	old ThinkPad X60/X200/X230…	**random Chromebook**
Open BIOS firmware	✅	✅
Open Embedded Controller	❌	✅
Disable Intel ME / AMD PSP	❌ (X60 only)	❌
Special tools required to flash	ch341a / Pi Pico / RPi; SOIC-8 / SOIC-16 clip or a soldering iron	worst case: a paperclip; usually nothing
Bonus: available new?	❌ used, aging hardware	✅
Bonus: reasonably fast?	❌ (fastest ever X230 has similar performance to some high-end chromebooks, but it gets to 100°C Tdie under load)	✅
Bonus: price?	surprisingly high for a 10+ year old device; corebooted ones go for big $$$	can be had for cheap or free
Hacker cred?	✅	Coming soon!

Long story short: you cannot escape from blobs…

Long story short: you cannot escape from blobs…

BUT

Long story short: you cannot escape from blobs…

BUT

…the current situation is actually better than 10+ years ago

I want to hack a Chromebook…
How do I get started?

1. Pick a fun device! There’s lots of them, most can be had for cheap

I want to hack a Chromebook…
How do I get started?

1. Pick a fun device! There’s lots of them, most can be had for cheap
    - Easy mode: x86_64, Intel

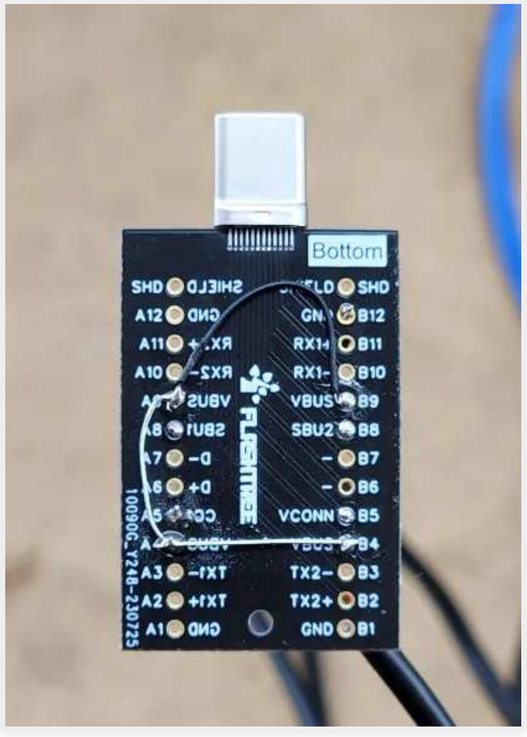
I want to hack a Chromebook…
How do I get started?

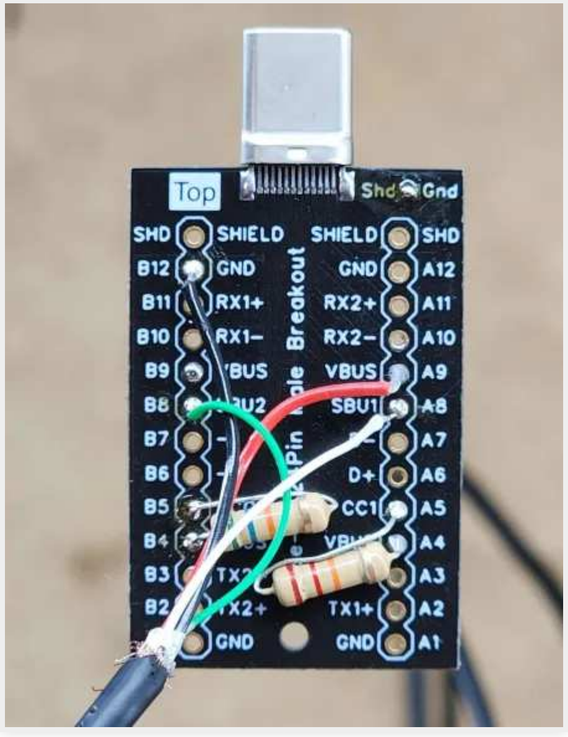
1. Pick a fun device! There’s lots of them, most can be had for cheap
    - Easy mode: x86_64, Intel
    - Hard mode: x86_64, AMD (worse platform support, some things are WiP)

I want to hack a Chromebook…
How do I get started?

1. Pick a fun device! There’s lots of them, most can be had for cheap
    - Easy mode: x86_64, Intel
    - Hard mode: x86_64, AMD (worse platform support, some things are WiP)
    - **Expert mode:** ARM64
        - we’re working on support as we speak :)
        - want to get into firmware development? join us, we’re friendly!

2. Optional: obtain or make a SuzyQ cable
    - highly recommended if you want to do development
    - cheap and easy to make
    - exposes three serial ports
    - superpowers: works with flashrom, making unbricking trivial





(pic credit: Bringus Studios)

SuzyQ cheat sheet

- flashrom -p raiden_spi_debug:target=AP <-- flashing the BIOS through SuzyQ
- /dev/ttyUSB0 - AP (CR50) console (Google H1 security chip)
- /dev/ttyUSB1 - Platform UART
- /dev/ttyUSB2 - Embedded Controller console
  - All UART interfaces use 115200 baud for compatibility reasons
  - SuzyQ as kernel debugger:
    - console=ttyS4,115200n8 on x86_64
    - console=ttyS0,115200n8 on ARM64

- Bonus: With USB-TTL adapter, it can also be used to debug new Google Pixel phones!

3. Unlock flash write protect
    - Older devices (pre-APL) have a WP screw
    - Newer devices require you to open the case and remove the battery
    - Some outliers require bridging two contacts on the motherboard (JasperLake)
    - SuzyQ makes this process a lot easier :)

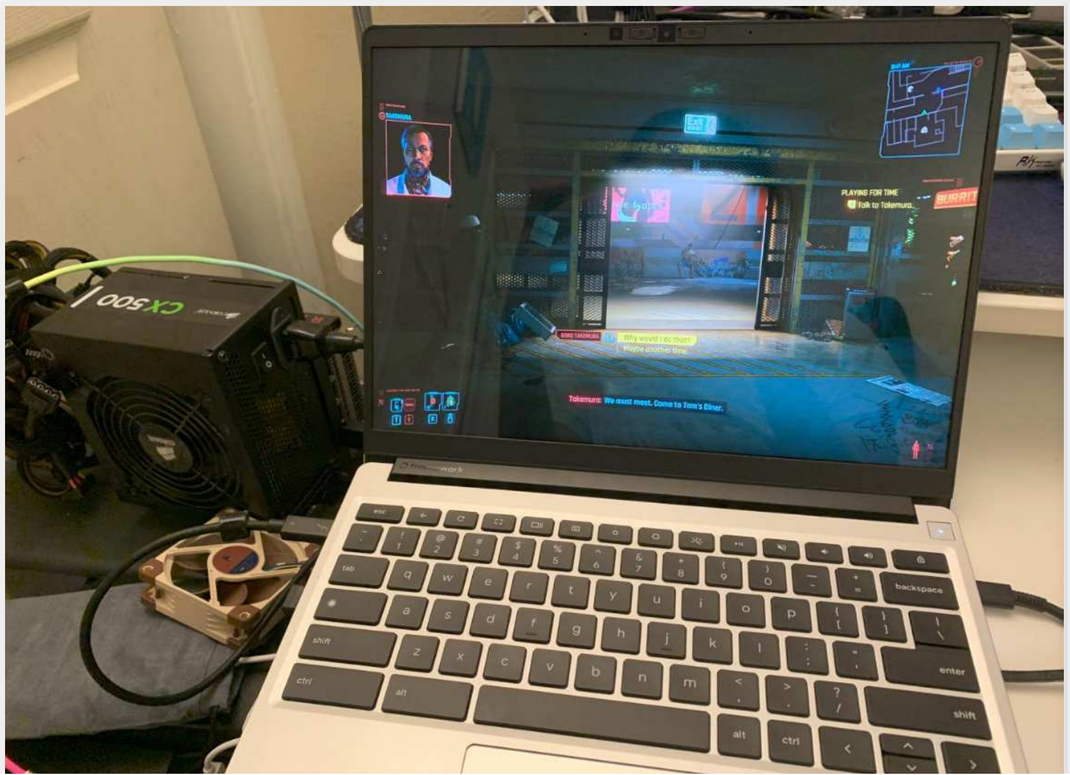
4. Flash and enjoy!

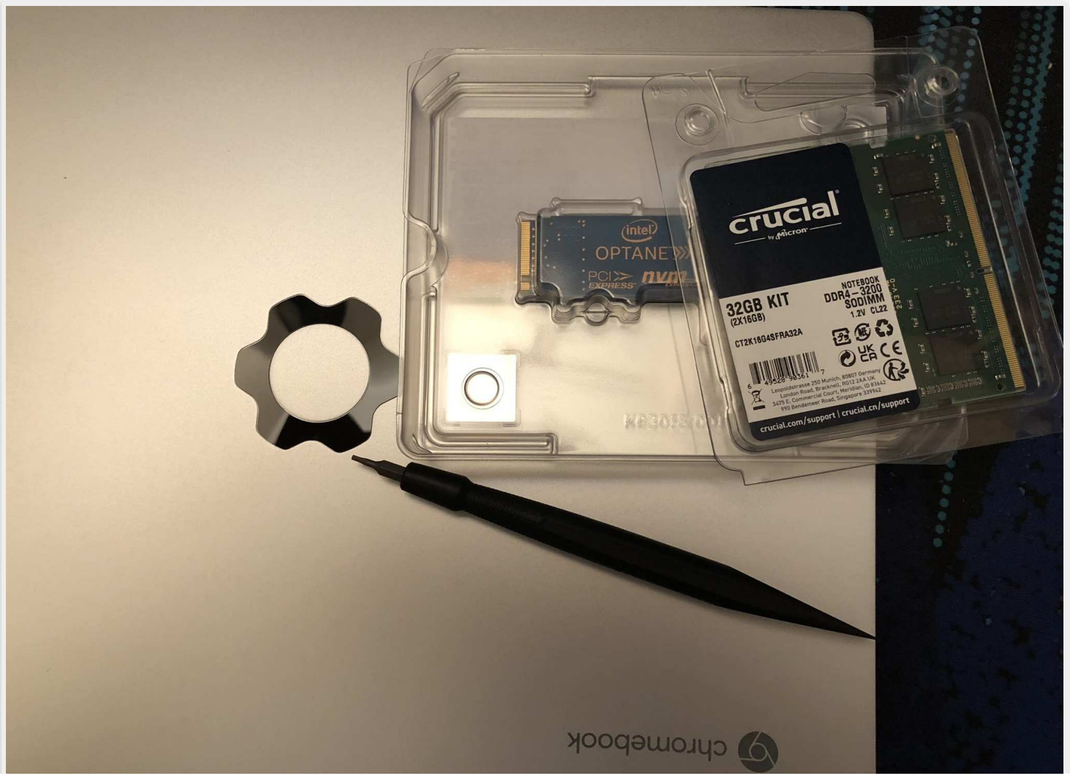
- We provide a wide variety of working, pre-built images
- Everything is open source - compile your own! :3

Cool devices!

BANSHEE (Framework Chromebook); it has no business being a Chromebook

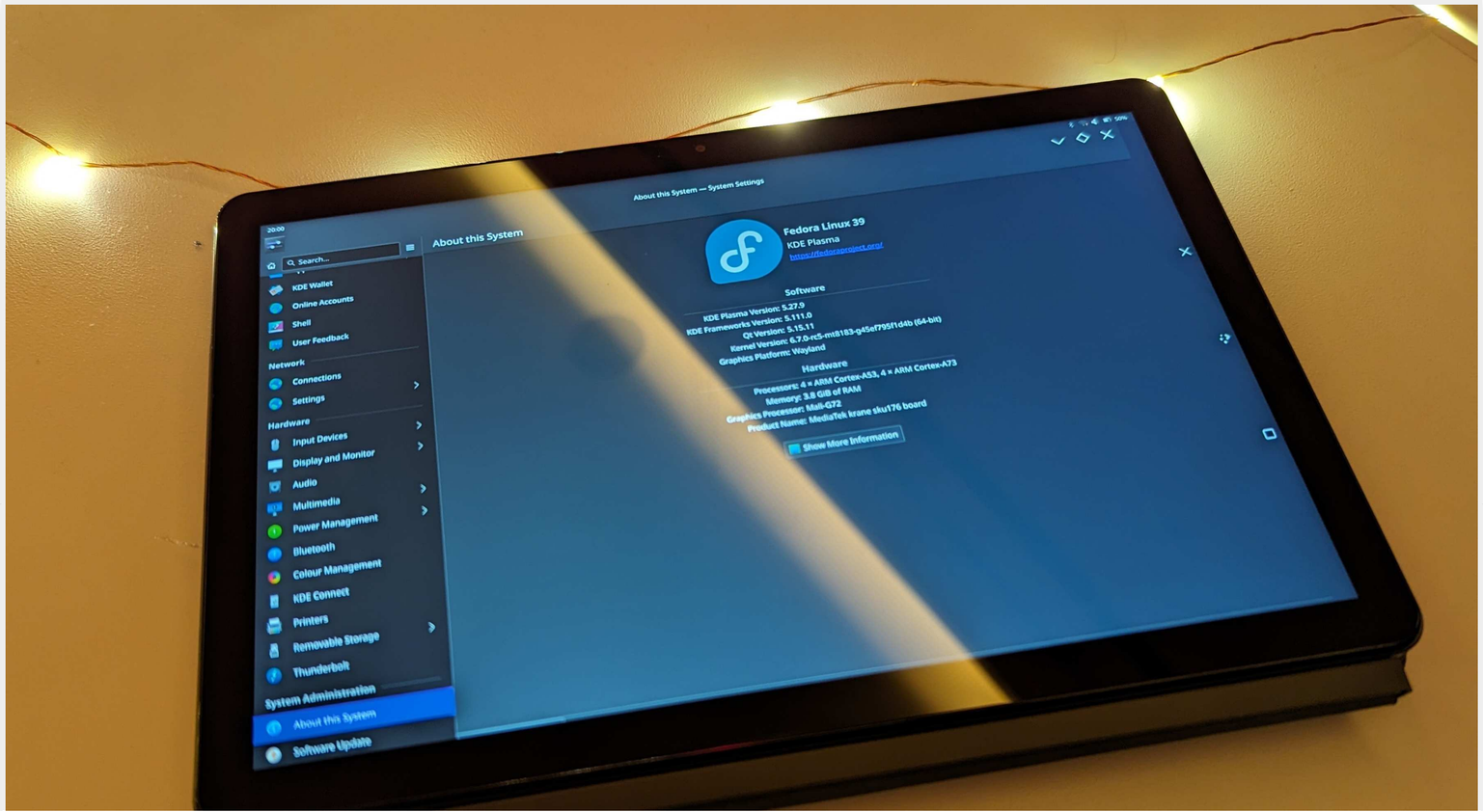
- Core i5-1240P
- Up to 64GB DDR4 SO-DIMM RAM
- 3:2 100% sRGB display
- Thunderbolt 4
- We’ve patched Coreboot and adjusted BAR size…

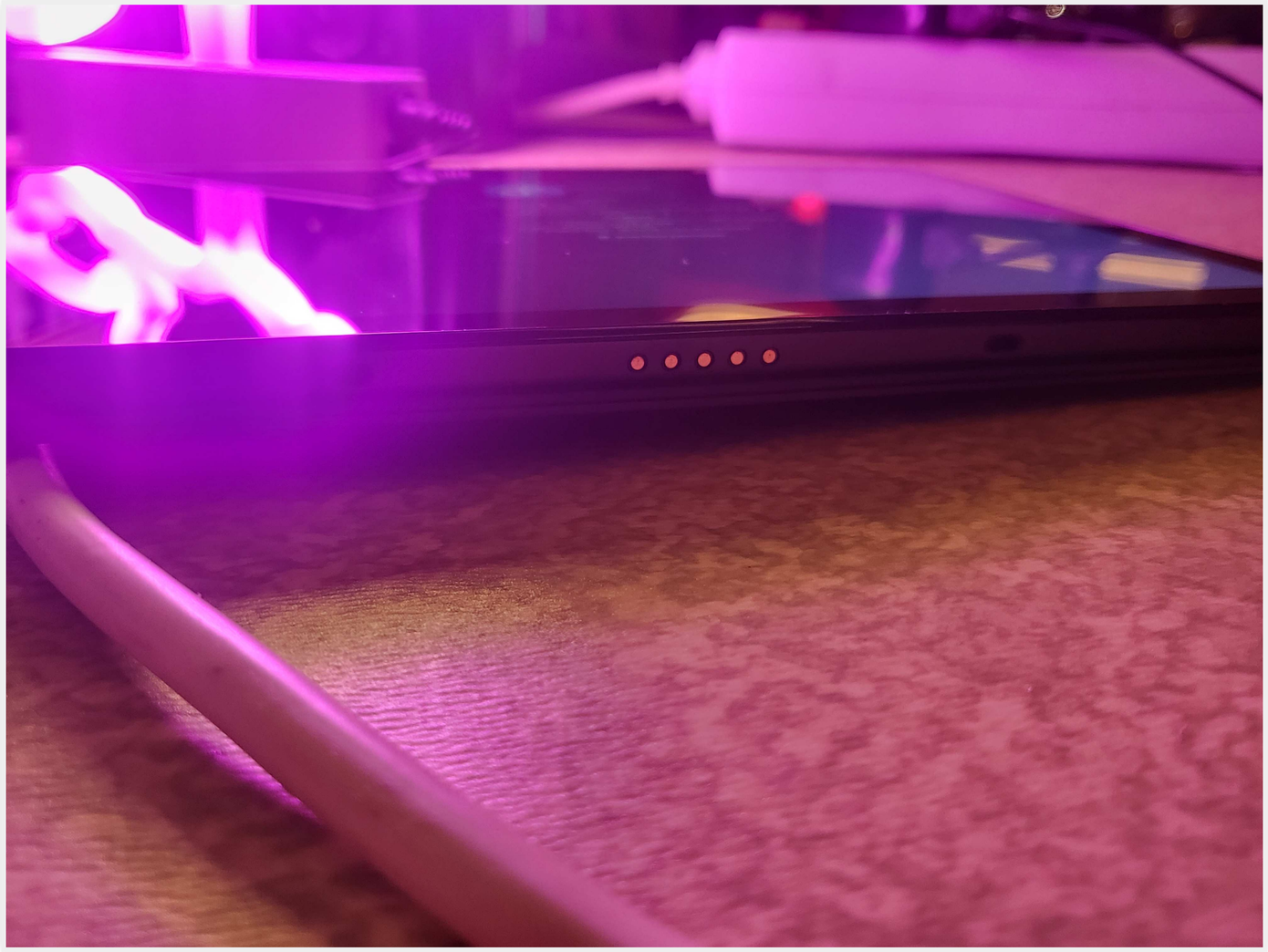


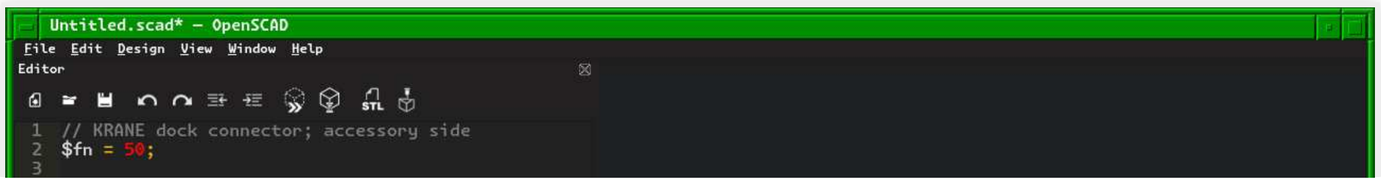


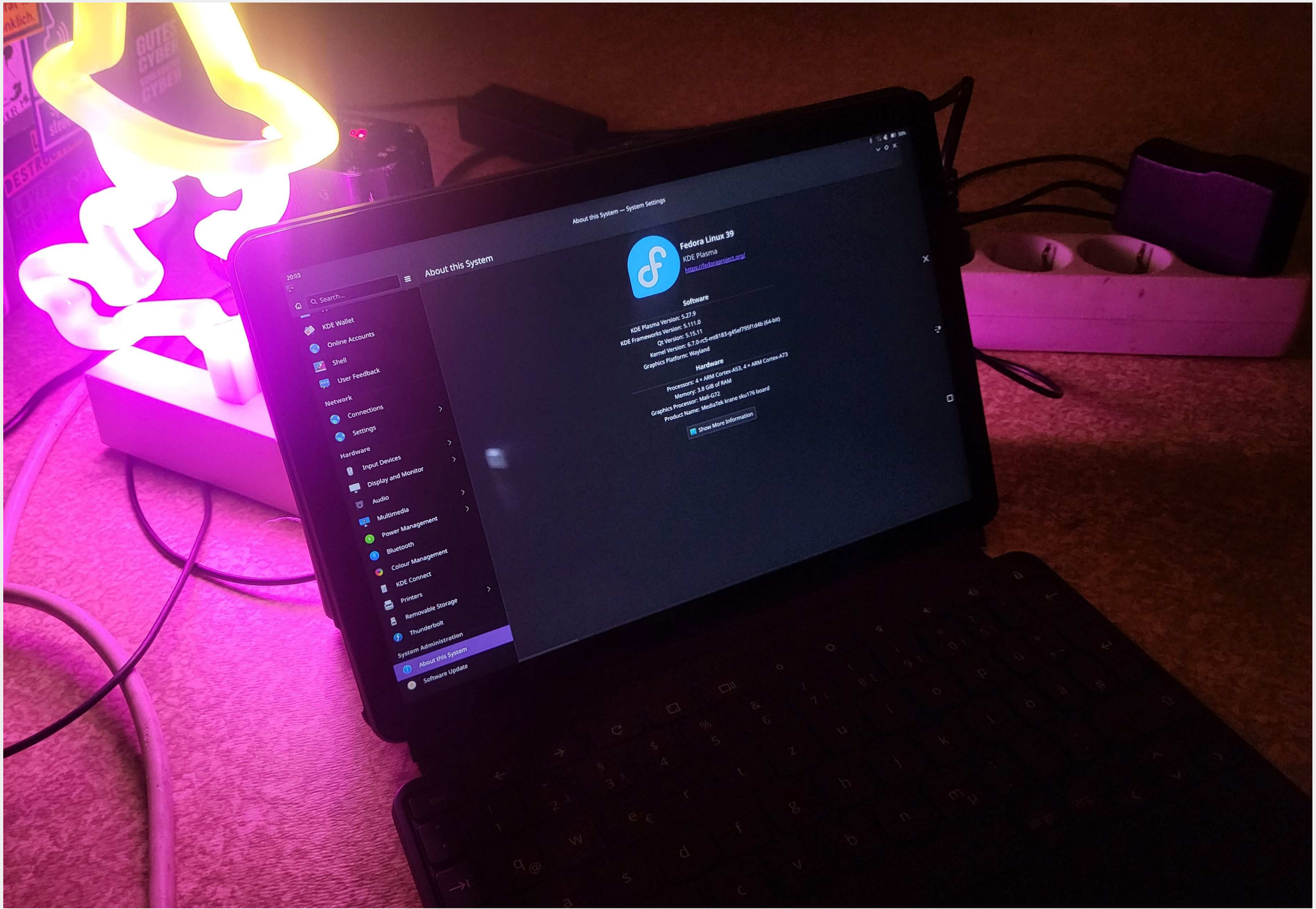
KRANE (Lenovo Duet); cheapo ARM64 tablet with a detachable keyboard and USI stylus support

- Huge potential for mods!
  - We’ve reverse-engineered pogo pins: USB, 3v3 VCC, GND and a resistor-based sense pin
  - Making our own accessories (i.e dock) would be pretty cool!
- Really small, almost pocketable
- Great for drawing!
- 100-130 EUR used; sometimes even less
- Cons: only 4GB of RAM :(



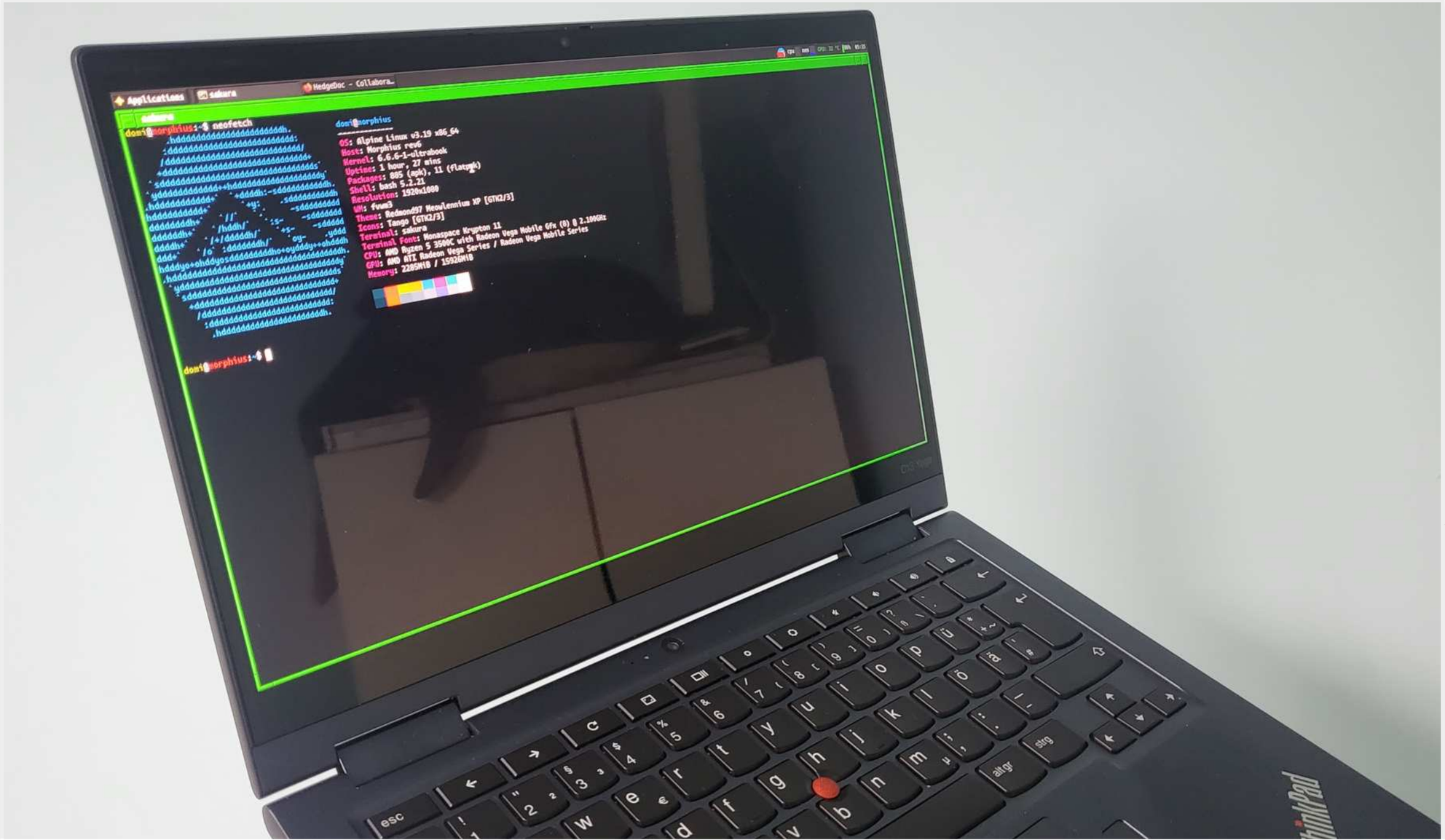


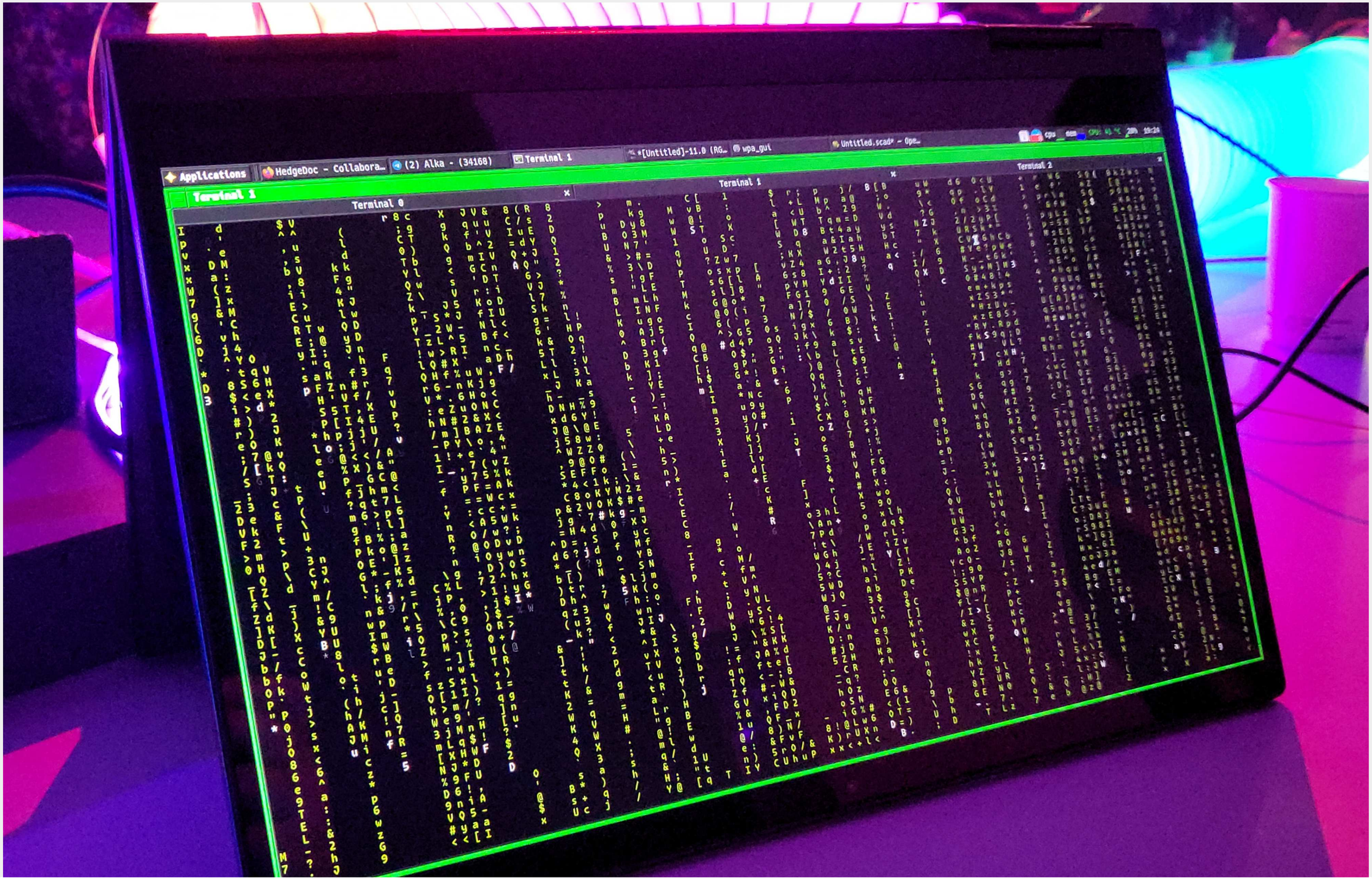

Untitled.scad* – OpenSCAD
File Edit Design View Window Help
Editor
STL
1 // KRANE dock connector; accessory side
2 $fn = 50;
3


About this System — System Settings
About this System
Search...
KDE Wallet
Online Accounts
Shell
User Feedback
Network
Connections
Settings
Hardware
Input Devices
Display and Monitor
Audio
Multimedia
Power Management
Bluetooth
Colour Management
KDE Connect
Printers
Removable Storage
Thunderbolt
System Administration
About this System
Software Update
Fedora Linux 39
KDE Plasma
https://fedoraproject.org/
Software
KDE Plasma Version: 5.27.9
KDE Frameworks Version: 5.111.0
Qt Version: 5.15.11
Kernel Version: 6.7.0-rc5-mt8183-g45ef795f1d4b (64-bit)
Graphics Platform: Wayland
Hardware
Processors: 4 × ARM Cortex-A53, 4 × ARM Cortex-A73
Memory: 3.8 GiB of RAM
Graphics Processor: Mali-G72
Product Name: MediaTek krane sku176 board
Show More Information

MORPHIUS (ThinkPad C13 Yoga)

- on this list because sdomi likes TrackPoints a bit too much
- ~350EUR new for a 16GB RAM/256GB M.2 NVMe model
- touchscreen w/ USI stylus support
- quite good I/O: 2x USB-C, 2x USB-A, 3.5mm jack, microSD reader

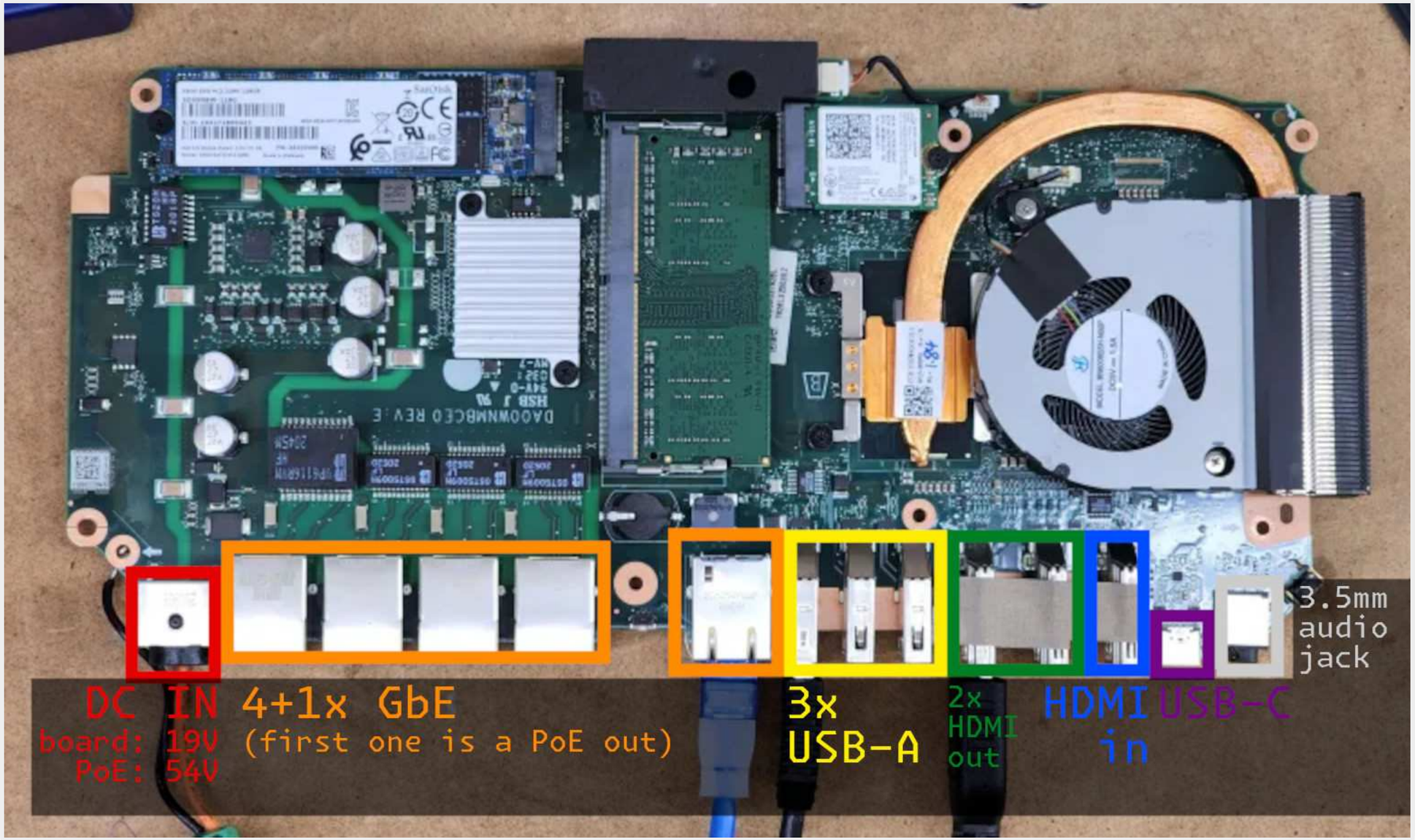



Applications
HedgeDoc - Collabora...
(2) Alka - (34168)
Terminal 1
*[Untitled]-11.0 (AG...
wpa_gui
Untitled.scad* - Ope...
Terminal 1
Terminal 0
Terminal 1
Terminal 2

ENDEAVOUR ((Lenovo) Google Meet Series One)

- all-in-one meeting room “solution”
- inside: 2x DDR4 SODIMM, 2x M.2 slots
- quite rare, ~~add to your mental “looking for” list~~
- hard to get into, but has some *awesome* I/O

(pic credit: Bringus Studios)



(pic credit: Bringus Studios)

Big thanks to:

- FyraLabs
  - Willing to maintain support for devices with stock firmware
  - Packaging our UCMs and other things that haven’t been upstreamed (yet!)
  - Collaboration on Submarine project (LinuxBoot-based bootloader for devices with stock firmware)
- Collabora
  - AngeloGioacchino Del Regno
  - Nícolas F. R. A. Prado
  - Everyone working on Panfrost project! :)
- Chromium, Linaro, U-Boot
  - Simon Glass
  - Ilias Apalodimas
  - Heinrich Schuchardt
- Our community and contributors!
- Everyone who helped us with the presentation, gave feedback, etc ;)

Contributions welcome!

/(sorry|yes)/, we're on discord		https://discord.com/invite/n3gM92XR7r
chrultrabook docs		https://chrultrabook.github.io/docs/
slides		https://md.sakamoto.pl/p/PczFNGGkj
additional presentations about the topic	 	http://tinyurl.com/meowcoreboot, https://coolstar.org/mirror/osfc-sunnyvale-2023.pdf