MNT REFORM: DIY PORTABLE COMPUTER

By Lukas Hartmann

"I know that there are some who would like to better understand and take control of their device — for reasons of security, curiosity, or the desire for personal customization and hackability."

Suggested citation: Hartmann, Lukas (2018). "MNT Reform: DIY Portable Computer." In *Interface Critique Journal Vol.1*. Eds. Florian Hadler, Alice Soiné, Daniel Irrgang
DOI: 10.11588/ic.2018.0.44735

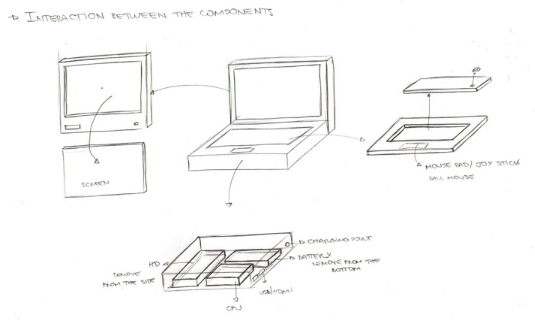
GOALS

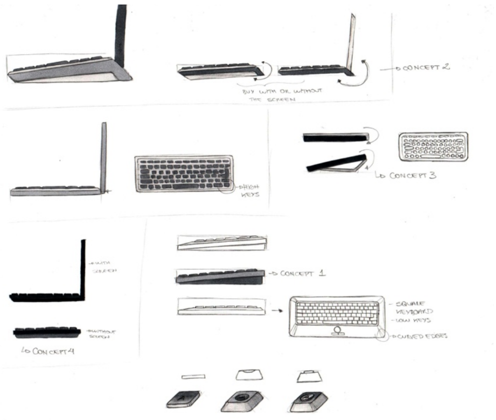
SECURITY, TRANSPARENCY, HACKABILITY

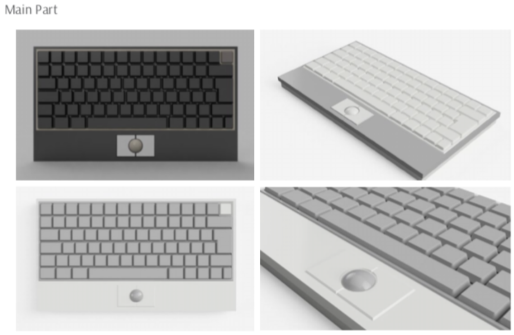
Like many of you, I spend a lot of time using computers: to write, draw, render, make music and program. I understand that most people want a digital appliance to get out of the way and make their lives easier, but I know that there are some who would like to better understand and take control of their device — for reasons of security, curiosity, or the desire for personal customization and hackability.
For a long time, I wanted a portable personal computer which you can…

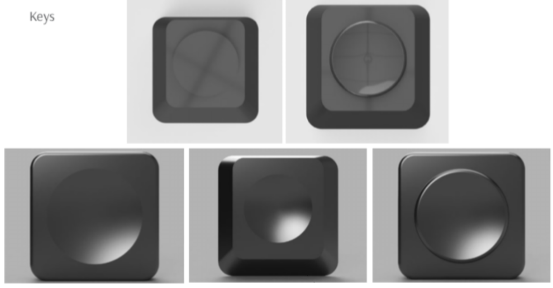
- Repair by yourself with parts from the hardware store or 3D printing
- Thoroughly understand on any level
- Take apart, modify and upgrade without regret
- Adapt to your tastes and use cases, staying with you for many years

In Summer/Autumn 2017, I teamed up with industrial designer Ana Dantas to make this machine happen. Three months later, I can type this article on the first prototype of Reform, our DIY portable computer.









FIRST SKETCHES AND 3D MODELS

OVERALL STYLE

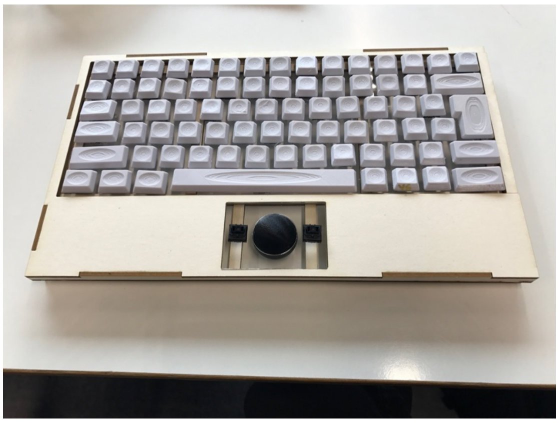
I grew up with home computers like the Sinclair ZX Spectrum, Commodore 64 and Amiga 500. These were quirky and limited in their capabilities, but they had a lot of character and invited to experiment and take apart. It was normal to find complete schematics and programming tutorials in their user manuals.

For the design of Reform, I wanted to incorporate a nod to these and other early personal computers and laptops, so that the look would convey an invitation to hack.

I created a mood board of the approximate style that I had in mind and Ana used this as a reference for the shapes and colors of her first hand sketches. As a product designer, she also came up with several alternatives for dealing with the goal of modularity. While she came up with some more advanced options, we decided to go with simplicity and chose a boxy design for the first prototype.

MODULARITY

Modularity was an important goal and differentiator from the beginning. You should be able to swap in the type of keyboard, pointing device and screen (E-Ink, anyone?) of your choice or even go without an internal screen at your desk.



We decided to include these modules in the first version:

- Chassis: a main body box with mounting facilities for motherboard, battery, SSD storage, input device controllers
- Exchangeable keyboard PCB, swappable by sliding out to the right
- Exchangeable pointing device (we would implement a trackball first, but you can also use a trackpad)
- A slot for SATA SSDs (reminiscent of floppy drive slots)
- Detachable display housing (with standard screws), with a future E-Ink option
- Standard LiPo batteries

ELECTRONICS

TARGET SPECS

The goal for selecting the core hardware was to strike a balance between openness and performance. On the one hand, I wanted to be able to get real work done on the machine, like compiling code, editing images or video/audio and browsing the web. On the other hand it had to be as open and documented as possible, avoiding closed source drivers. I came up with these requirements:

- CPU/SoC parts should be as open, documented, backdoor- and blob-free as possible
- At least 1 GHz and 2 cores
- At least 4 GB of RAM to avoid swapping
- At least SATA-2 for reasonably fast and big storage
- At least HD-ready screen resolution
- PCIe for expandability

SYSTEM ON A CHIP: NXP i.MX6 / i.MX8

After reviewing most available SoCs and SBCs, I settled (like Bunnie in the Novena) on the NXP i.MX6 QuadPlus SoC, which has four ARM Cortex-A9 cores running at up to 1.2 GHz and a Vivante GC2000 GPU for which completely open source drivers are available (etnaviv). Complete documentation for the SoC is available from NXP as PDFs without registration or NDAs. I chose the TinyRex Ultra by Fedevel/Voipac as our evaluation board, because it is a tiny, low-cost system-on-module with 4GB of memory, and it can be plugged into an open source carrier board. For the next iteration, I might go with the open schematics version iMX6 Rex[1].

While the i.MX6 looks a bit dated on paper and can't compete with the Core i7 in my old MacBook Pro, its real-world performance is good enough for the kind of work I wanted to use it for. Also, the much more powerful 64-bit, 6-core i.MX8[2] is around the corner.

[1] http://www.imx6rex.com/

[2] https://www.nxp.com/products/microcontrollers-and-processors/arm-based-processors-and-mcus/i.mx-applications-processors/i.mx-8-processors/i.mx-8-family-arm-cortex-a53-cortex-

With a bit of luck, the second prototype of the computer will coincide with its launch.

CONNECTIVITY: USB, PCIe, SATA, Wi-Fi

While i.MX6QP has an integrated USB2.0 Host Controller, which is fine to connect the input devices and a sound module, USB3.0/3.1 are attractive because of their high bandwidth and charging via USB-C. These could be integrated on the new motherboard and a chip attached to the PCIe bus. I tested the Penguin Wireless N Mini PCIe card[3] in the dev board's Mini PCIe slot and it works great with the open source ath9k drivers.

The system starts U-Boot from a MiniSD card which in turn loads Linux from a 120GB SanDisk SATA SSD. We included a compartment and a slot that accepts 2.5" SATA SSD drives for easy swapping.

PRICING AND USER PRE-TESTING

Ana encouraged me to set up a user pre-validation test to check if we were on the right track concerning the overall concept, choice of parts and pricing. We put together a fake Amazon page[4] with renderings and a description of our device surrounded by two alternative choices. We positioned our computer in the middle of the price range (500-700 EUR) and focused on offering a complete DIY package (not only a board) and higher specs and customizability than another low-cost project. The mainly positive interviews with pre-filtered testers made us confident that we were on the right path.

All testers were excited about an optional E-Ink display, but this drives up the price by at least 200-300 EUR. Also, controller options were unclear, so we postponed prototyping of this option.

MAKING IT WORK

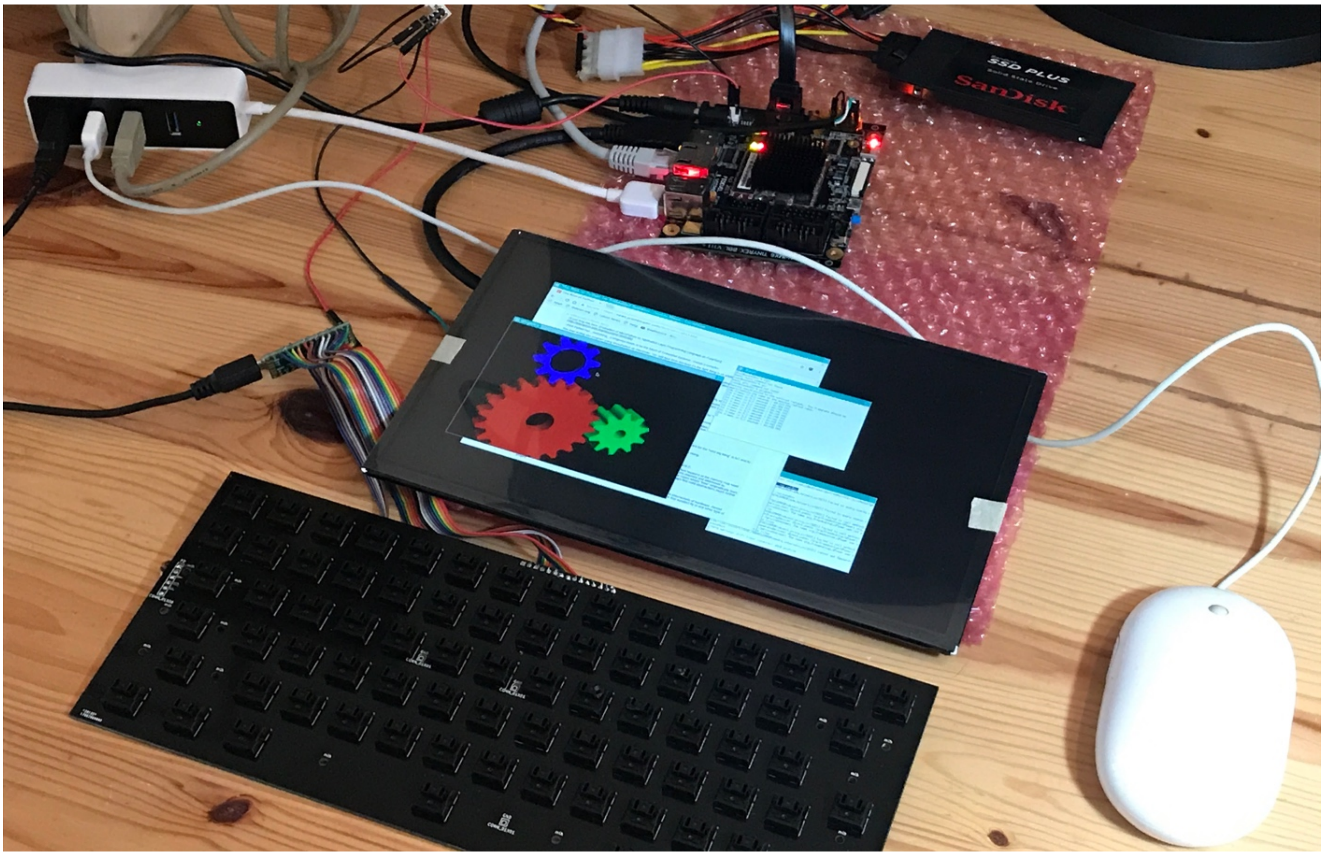
OPERATING SYSTEM VALIDATION

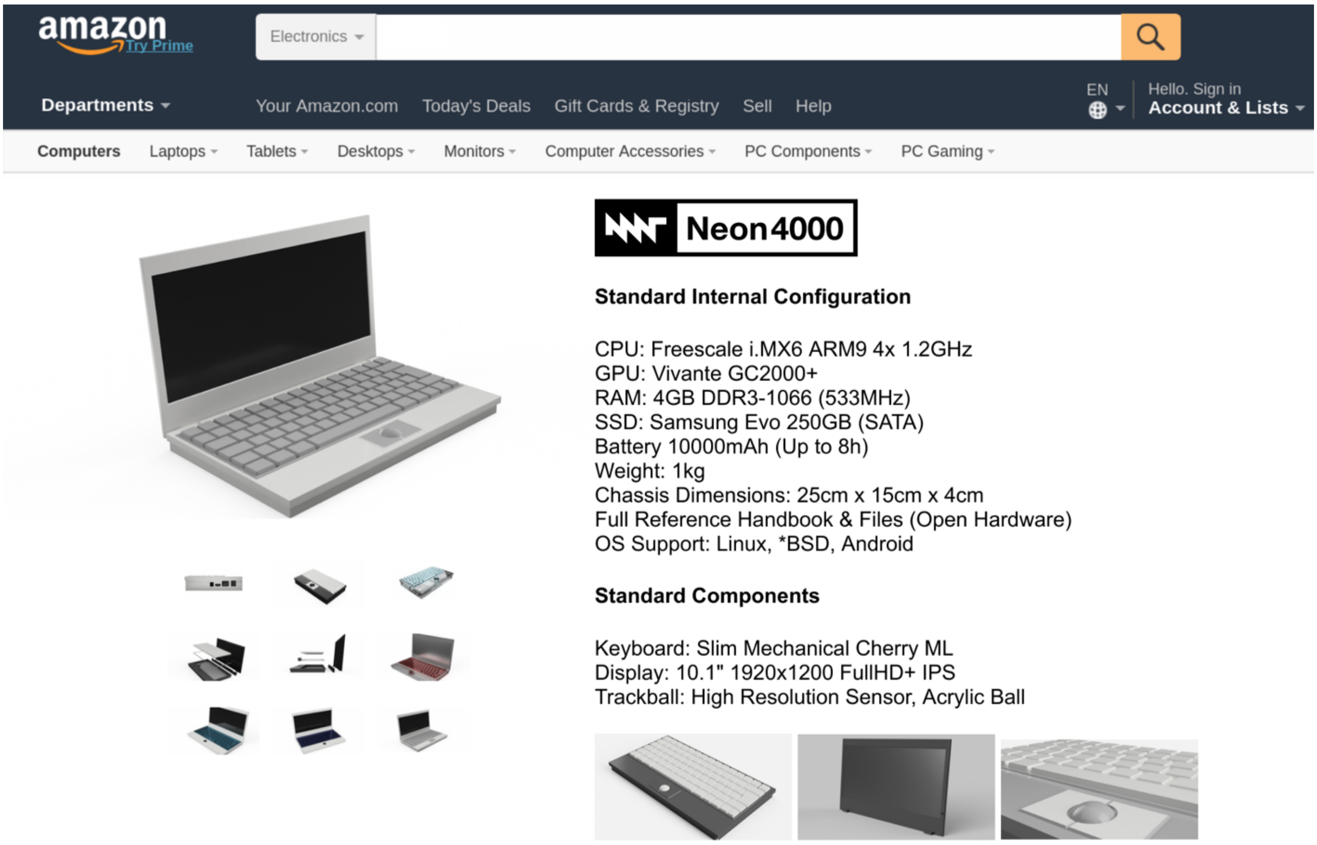
The first thing I wanted to make sure was the possibility of running mainline Linux on the machine, without using any proprietary binary blobs or drivers — while still being able to use basic GPU acceleration. With some important hints from kernel developer Lukas Stach I was able to bring up a usable Linux desktop with Debian userland and OpenGL on my TinyRex development board hooked up to a battery. This validated the main technology decision.

a72-virtualization-vision-3d-graphics-4k-video:i.MX8

[3] https://www.thinkpenguin.com/gnu-linux/penguin-wireless-n-mini-pcie

[4] http://dump.mntmn.com/neonmock/product.html





Neon4000

Standard Internal Configuration

CPU: Freescale i.MX6 ARM9 4x 1.2GHz
GPU: Vivante GC2000+
RAM: 4GB DDR3-1066 (533MHz)
SSD: Samsung Evo 250GB (SATA)
Battery 10000mAh (Up to 8h)
Weight: 1kg
Chassis Dimensions: 25cm x 15cm x 4cm
Full Reference Handbook & Files (Open Hardware)
OS Support: Linux, *BSD, Android

Standard Components

Keyboard: Slim Mechanical Cherry ML
Display: 10.1" 1920x1200 FullHD+ IPS
Trackball: High Resolution Sensor, Acrylic Ball

KEYBOARD, SWITCHES AND CAPS

For the first variant of the keyboard, I was looking for DIY- and typing-friendly key switches. The Cherry ML switch[5] sits in a sweet spot between full-size mechanical keys and laptop chiclet keys, but I couldn't get them at Digikey or Mouser. Calling Cherry's headquarters in Munich was fruitful: they pointed me to a partner company that would then sell me a minimum quantity of 2000 switches. I made a KiCAD component for them, drew up a keyboard PCB and ordered it at Dangerous Prototypes. Hooking the assembled board up to a Teensy LC yielded a usable laptop keyboard.

Because our layout was custom and injection molding is too expensive for small hardware projects, Ana suggested printing our keycaps in a Formlabs Form 2 SLA resin printer[6]. This approach resulted in surprisingly pleasant-to-touch, slim caps for the MLs that we could easily customize. A proper wire mechanism to stabilize the space bar and other long keys is still on the to-do list, but the keyboard already works well enough for first application testing.

POINTING DEVICE

Ana also used the Form 2 for printing a custom designed housing for the PWM3360 motion sensor[7] that would form the base of our trackball module. The ball itself is a standard acrylic sphere from Modulor, and the left and right buttons are Cherry ML as well.

Both the Keyboard and the pointing device are driven by Teensy LC[8] Cortex M0+ boards that speak USB HID. To connect these internally to the SoC, I chose the incredibly small USB NanoHub board[9] by Muxtronics.

DISPLAY

For the first display prototype, I compromised and chose a 10" 1920x1200 panel with a corresponding HDMI to dual LVDS adapter by Chalkboard[10]. I felt that although this approach requires a little dangling HDMI cable on the outside of the prototype, it would save considerable development time to quicker reach a functioning system for further user testing. In the next iteration, this cable will be eliminated.

CHASSIS

For printing the chassis, encompassing the main box and the display assembly, we chose to go with Shapeways[11] with good results, although at a later stage these might be candidates for resin cast-

[5] http://cherryamericas.com/product/ml-series/

[6] https://formlabs.com/3d-printers/form-2

[7] https://www.tindie.com/products/jkicklighter/pmw3360-motion-sensor/

[8] https://www.pjrc.com/store/teensylc.html

[9] https://www.tindie.com/products/mux/nanohub-tiny-usb-hub-for-hacking-projects/

[10] https://www.chalk-elec.com/?page_id=1280#!/10-FullHD+-LCD-with-HDMI-interface/p/41737268/category=3094859

[11] https://www.shapeways.com/

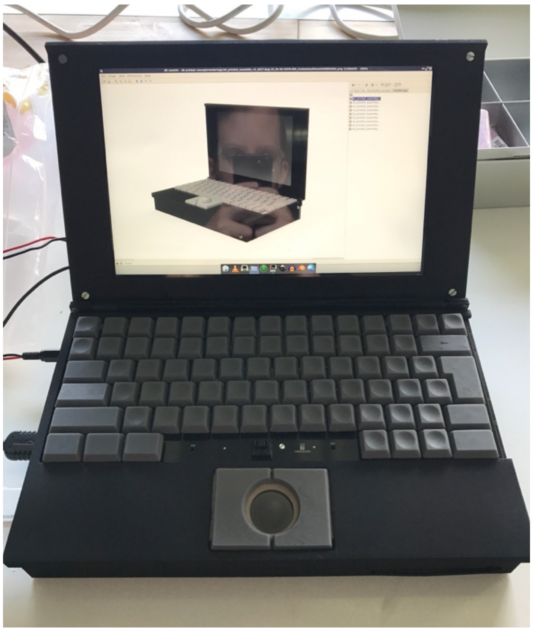
ing or CNC milling, or other technologies that are available to a user that wants to customize the housing.

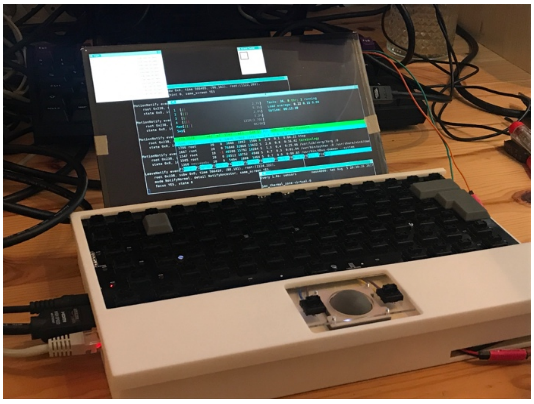
ASSEMBLY

After around 8 weeks of working on all the pieces, we met in FabLab Berlin[12] to assemble the first working system over an intense course of around five hours. Many little things went wrong — from wrongly placed holes to bad solder joints to pieces of 3D printed material breaking off — but because we were surrounded by all necessary production equipment and Ana's never ending supply of improvising techniques, we made it happen and our DIY laptop booted for the first time.

SIZE, WEIGHT, BATTERY LIFE

The first prototype of Reform is quite a brick. While it is only 28cm wide and 17.5cm deep, its complete height including the display adds up to 5.5cm to accommodate for all the connectors of the development board and to allow room for experimentation before shrinking everything down. Its 1.5kg including the battery feel OK, though. The battery is an off-the-shelf RC 7.4V LiPo battery with a modest 3000 mAh capacity. Running a full linux desktop on full LCD brightness clocks in at around 1.8A, 5V, which yields 2.5 hours of continuous usage on this battery. Doubling that number would be a good target.





APPLICATIONS

I successfully tested software packages like LibreOffice, Blender, GIMP, Inkscape and Audacity on the laptop. I managed to run Quake 3 Arena using the GPU, but I had to manipulate the OpenGL version number, which leads to some glitches that should be fixed. It can also play H.264 movies with mplayer and surf the web with QupZilla or Chromium (JavaScript heavy websites drive up the core

[12] https://fablab.berlin/en/

temperature, though). Because they are connected and flashable internally via USB, you can even update the keyboard and trackball firmware using the Arduino tools.

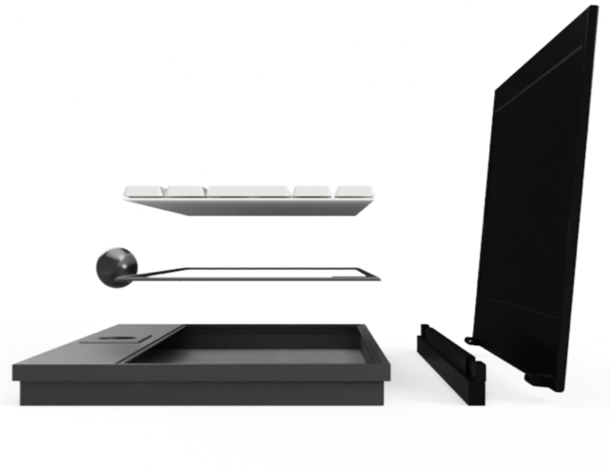
MAKING IT AVAILABLE

BUY KITS OR PRINT YOUR OWN

All in all, although it has a number of rough edges, we're very happy with how this first prototype turned out. Of course we want to make it better, thinner, get rid of external cables, publish schematics and documentation and set up an online shop where you can order DIY kits or single parts. And with the 3D files, you will be able to print chassis parts in any color or modified shape you want. We want to encourage experimentation and tinkering, and bring back some hacking fun to mobile personal computing.

MOTHERBOARD PRODUCTION

Fedevel[13] and Voipac[14], the Designers/Manufacturers of TinyRex, agreed to help with the production SoC/SOM, and Voipac offered an upgrade path to the next generation i.MX8 SoC that will be launched in the coming 3-6 months. My job will be to design a slimmed-down base board, an integrated charger/power brick and work with Ana on making the interconnect between chassis, display and input devices clean and easy. Ultimately, we will launch a crowdfunding or pre-order campaign. The latter worked well to make my previous hardware project, the VA2000 Amiga FPGA Graphics Card a reality.



[13] http://fedevel.com/

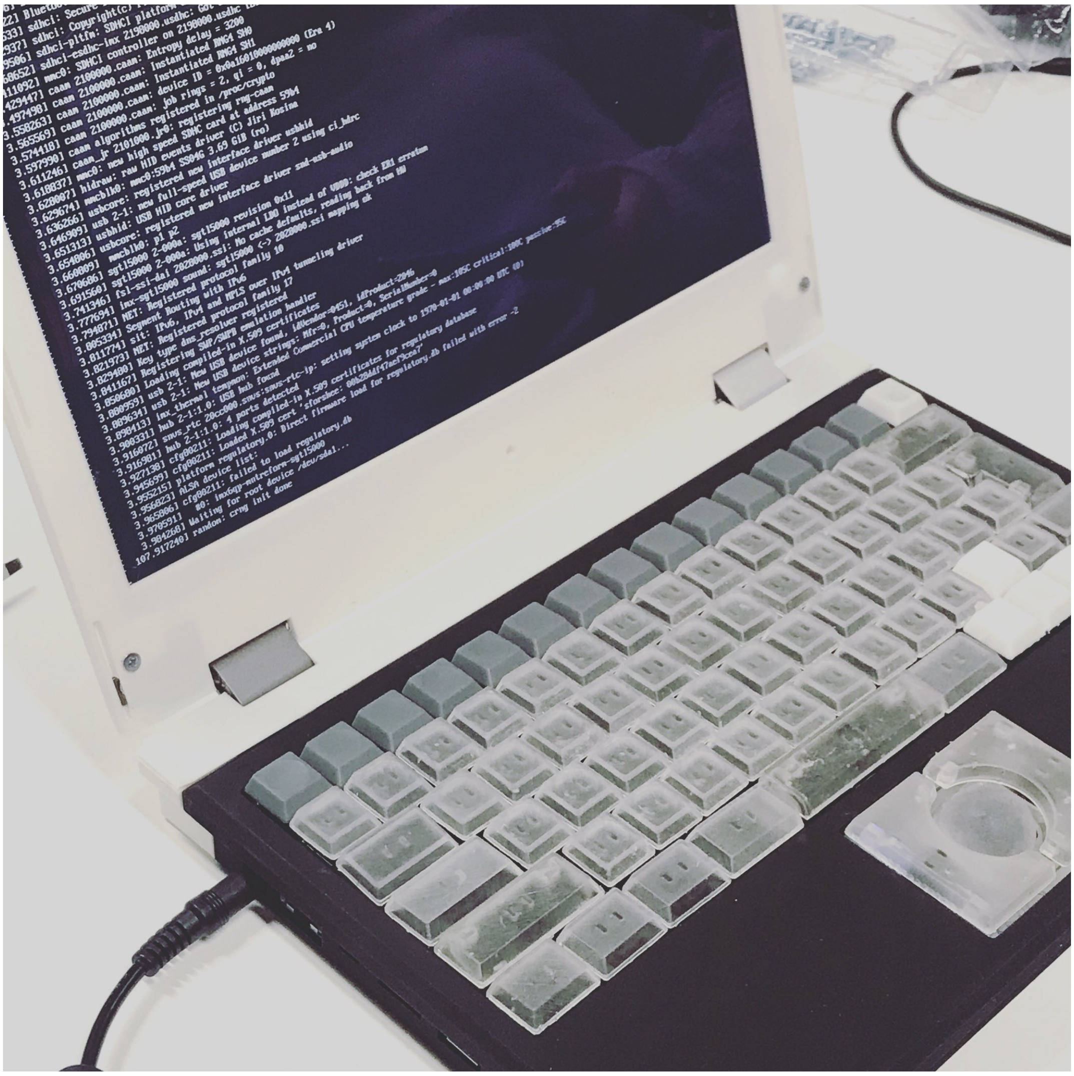
[14] http://www.voipac.com/

YOUR OPINION

At this point, we're eager to hear your early feedback about Reform. What would you like to see in an open, portable computer system? What did we miss? Send your thoughts to lukas@mntmn.com.
Or subscribe to the Reform newsletter[15] and we will update you on further progress of the project.

Check out other MNT Projects[16]

Related DIY laptop projects:
Novena[17], Olimex Teres-1[18]



```
22] Bluet
533] sdhci: Secure
937] sdhci: Copyright(c)
9506] sdhci-pltfm: SDHCI platform
68652] sdhci-esdhc-imx 2190000.usdhc: Got
11092] mmc0: SDHCI controller on 2190000.usdhc
429447] caam 2100000.caam: Entropy delay = 3200
.497498] caam 2100000.caam: Instantiated RNG4 SH0
3.558263] caam 2100000.caam: Instantiated RNG4 SH1
3.565569] caam 2100000.caam: device ID = 0x0a16010000000000 (Era 4)
3.574418] caam 2100000.caam: job rings = 2, qi = 0, dpaa2 = no
3.597990] caam algorithms registered in /proc/crypto
3.611246] caam_jr 2101000.jr0: registering rng-caam
3.618837] mmc0: new high speed SDHC card at address 59b4
3.628007] hidraw: raw HID events driver (C) Jiri Kosina
3.629674] mmcblk0: mmc0:59b4 SS04G 3.69 GiB (ro)
3.636266] usbcore: registered new interface driver usbhid
3.646909] usb 2-1: new full-speed USB device number 2 using ci_hdrc
3.651313] usbhid: USB HID core driver
3.654806] usbcore: registered new interface driver snd-usb-audio
3.660809]  mmcblk0: p1 p2
3.670686] sgtl5000 2-000a: sgtl5000 revision 0x11
3.691560] sgtl5000 2-000a: Using internal LDO instead of VDDD: check ER1 erratum
3.741346] fsl-ssi-dai 2028000.ssi: No cache defaults, reading back from HW
3.777694] imx-sgtl5000 sound: sgtl5000 <-> 2028000.ssi mapping ok
3.794871] NET: Registered protocol family 10
3.805334] Segment Routing with IPv6
3.811774] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
3.821973] NET: Registered protocol family 17
3.829480] Key type dns_resolver registered
3.841167] Registering SWP/SWPB emulation handler
3.850680] Loading compiled-in X.509 certificates
3.880959] usb 2-1: New USB device found, idVendor=0451, idProduct=2046
3.889634] usb 2-1: New USB device strings: Mfr=0, Product=0, SerialNumber=0
3.898413] imx_thermal tempmon: Extended Commercial CPU temperature grade - max:105C critical:100C passive:95C
3.900331] hub 2-1:1.0: USB hub found
3.916072] snvs_rtc 20cc000.snvs:snvs-rtc-lp: setting system clock to 1970-01-01 00:00:00 UTC (0)
3.916981] hub 2-1:1.0: 4 ports detected
3.927138] cfg80211: Loading compiled-in X.509 certificates for regulatory database
3.945699] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
3.955215] platform regulatory.0: Direct firmware load for regulatory.db failed with error -2
3.956823] ALSA device list:
3.965806] cfg80211: failed to load regulatory.db
3.970591]   #0: imx6qp-mntreform-sgtl5000
3.984268] Waiting for root device /dev/sda1...
107.917240] random: crng init done
```

[15] http://letters.mntmn.com/lists/reform/join

[16] http://mntmn.com/

[17] https://www.crowdsupply.com/sutajio-kosagi/novena

[18] https://olimex.wordpress.com/2017/02/01/teres-i-do-it-yourself-open-source-hardware-and-software-hackers-friendly-laptop-is-complete/