

Funding application to improve Debian support of the MNT Reform open hardware laptop

Johannes Schauer Marin Rodrigues

2024-07-28

Preface

In this document I'd like to apply for Debian funds to be spent on an MNT Reform laptop. My goal is to be able to efficiently perform maintenance and development tasks targeting that laptop with the end-goal of being able to install and run vanilla Debian on the device and have it be supported by me during the life-time of future Debian stable releases.

The initial trigger for this funding request was a private conversation with Andreas Tille, originating from his mail to `debian-project` about a potential RISC-V module for the Framework laptop in June 2024¹. Andreas suggested I create a more formal write-up which is what you are now seeing in front of you. Thank you Andreas, for this opportunity.

Why is the MNT Reform relevant for Debian?

I believe that free software needs free hardware. I have been trying to run Debian on free hardware since my first baby steps with Debian during the times of the Openmoko GTA01 Neo 1973 for which I was maintaining custom Debian images (with our own user interface called PyNeo) between 2007 and 2010. This is also what ultimately got me into Debian via emdebian (the bootstrapping and cross-building team back then), into a Google Summer of Code with them in 2012, a master thesis on Debian dependencies in 2013, made me rewrite `multistrap` in 2017 and create `mmdebstrap` the year after. All the while I was accumulating what became a growing graveyard of hardware which was more or less able to run Debian GNU/Linux. It is not hard to find enthusiastic communities of people out there who create their own devices under an open hardware license and with the spirit of development in the open just as we practice it in Debian. What is hard to find is a device which is more than something to tinker with during weekends. What I was seeking was a device which I could use as my main computing platform without looking back. There are manufacturers who produce hardware which is usable as a daily driver but their idea of an open platform is comparable to the open-ness of Android. Yes, you may get CAD files or even schematics and they may even be distributed under an open license but you are



Figure 1: MNT Reform open hardware laptop. Copyright MNT Research GmbH CC BY-SA 2.0

¹ https://lists.debian.org/ZnGtiJAU2jP_Go1x@an3as.eu

still 100% reliant on the manufacturer to produce the hardware for you because its design requires industrial production methods that are not attainable for independent makers.

The MNT Reform is what I think is the only open hardware laptop right now which is both usable as a daily driver and which implements the spirit of open-ness as we are all used to from free software projects: development in the open and the ability to reproduce everything “at home”. The latter has for example been shown by jacqueline who re-created the Reform innards and hand-soldered them on her own custom purple PCBs as can be seen in Figure @ref(fig:jacqueline). If you like, you can also 3D print the main body of the laptop on your 3D printer at home (see Figure @ref(fig:PCo20009)). The former can be seen from how the device is developed by MNT. The 3D models and KiCAD files of all components (in case of the LS1028A or Kintex-7 module) are stored in git. If you like, you can download the altium files and get the CPU module manufactured yourself. All the changes that are made by MNT are immediately visible as git commits. This also holds for hardware that MNT didn’t even release yet. If you are curious about the “Reform Next”, a slimmer version of the MNT Reform, you can download 3D stl models of the current version of its case as well as kicad files of the current revision of its PCBs right now². If I want to submit fixes to their hardware or their software or their documentation, then I can do so by submitting a merge request on their gitlab. This is how I started my journey with MNT: I had a lot to criticize about how they were building their Debian packages and system images and filed merge requests with my improvements.

I do not want to throw shade on companies like Framework. Their releasing PCB schematics and 3D CAD models to the public under an open license is a huge step up compared to the status-quo in the laptop market. I just consider what MNT does yet another step up in terms of open and reproducible development of hardware:

“What works for me is to say, you can clone this thing. If we go out of business, you can still make one.”

— Lukas F. Hartmann³

The fact that the publicly available git repositories of MNT do not only contain everything required to re-create the laptop but are also at the same time the exclusive development repositories by MNT itself reflects the way we are used to develop software in Debian. Instead of irregular dumps of our source code without a real way to contribute and interact as it is done for “open source” projects like Android, we want our main repositories be publicly available and interactable and share all progress we make with the public. To

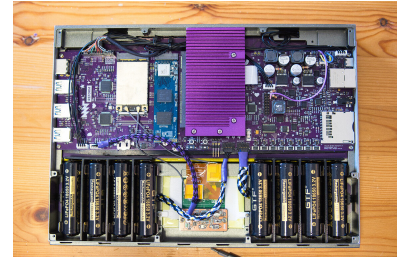


Figure 2: Hand-soldered MNT Reform with custom purple PCBs and USB-C charging. Copyright @jacqueline@chaos.social CCo

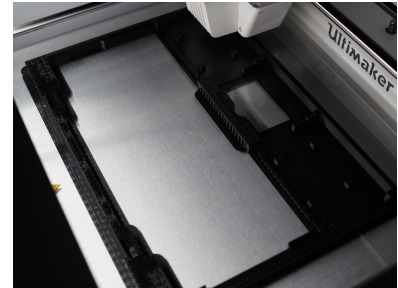


Figure 3: My own 3D print of the MNT Reform main box on an Ultimaker S3

²<https://source.mnt.re/reform/reform-next>

³<https://mntre.com/reform-irc-logs/2024-06-18.log.html#t19:34:19>

my knowledge, the MNT Reform is the only readily available open hardware laptop which is mature enough such that it can be used as a daily driver and which implements this idea of open-ness and hardware freedom. It thus, in my eyes, deserves the attention of the Debian project as the free hardware we want our free operating system to run on perfectly.

What work will be enabled by granting this request?

My work with the MNT Reform lead to a few new packages in Debian, like for example the video player `clapper` which allows hardware accelerated video decoding or the `box64` emulator which allows one to run x86 binaries on ARM. Even on comparatively slow ARM processors, this allows playback of 1080p@60 h264 video or playing Starcraft (see Figure @ref(fig:P1050059)) and Stardew Valley (40 hours in networked multiplayer with my partner on a Thinkpad), respectively. But maintaining these packages does not require the actual hardware for 99% of their maintenance tasks. On the other hand there are some areas where access to an MNT Reform is not only required, but which require destructive access to it.

For example, Figure @ref(fig:d-i) shows `debian-installer` running on my MNT Reform. This works in principle but I'd like to improve on the current experience, support other CPU modules and make sure that there are no regressions over time. For example, users reported to me, that `d-i` works fine with the kernel from Debian bookworm but with the kernel from `bookworm-backports`, the screen does not work with the `imx8mq`. This is something that I have to investigate on the actual hardware (this is not something that QEMU can emulate).

Furthermore, there was a recent change in `debian-installer` which broke `flash-kernel` from configuring the correct device tree which means that the installed system fails to boot. To test and fix this, I need to actually finish the installation on the device but that would destroy my existing installation, making creating a fix for this very tricky.

To make some of these maintenance tasks more manageable, I created what I call the "MNT Board Reform" in Figure @ref(fig:P2060098). This is my old motherboard (which has a broken audio chip) with the LS1028A CPU module on it. It is powered by an ATX power supply and is switched on by a Raspberry Pi Pico RP2040 which is sending the appropriate commands to the motherboard firmware (this would otherwise be the task of the keyboard). I can use this setup to test installing Debian using `d-i` via a serial connection over UART. But this system does not have a display (so I cannot test the



Figure 4: Playing Blizzard's StarCraft: Terran mission 3: "Desperate Alliance" on the MNT Reform using `box64`

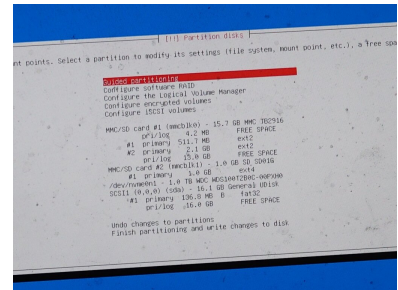


Figure 5: The partition setup of the Debian installer running on the MNT Reform with `imx8mq`

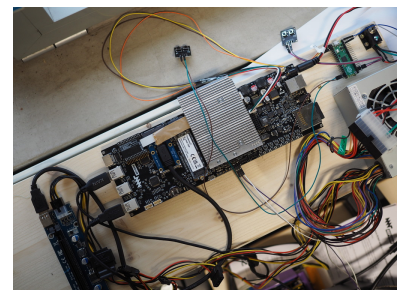


Figure 6: LS1028A powered by ATX PSU and a full-size PCI-E connector to test external graphics cards

problem with the screen not coming on), it does not have keyboard or trackball attached to it (so I cannot test their interaction with it or test new firmware versions), it does not have the battery boards attached (so I cannot for example test the recent regressions of the battery plugin in waybar⁴) and since its audio chip is broken, I cannot do tests involving audio playback either (some custom alsa and pulseaudio configurations are required for sound to work).

⁴<https://github.com/Alexays/Waybar/issues/3482>

The current functionality was mostly created when I still had my old Thinkpad as my main machine and the MNT Reform as a disposable second machine that I could use to tinker with as it contained no relevant data. Now that the Reform became my main laptop, more and more bugs start appearing which I have a hard time fixing as I rely on other users gathering all the data for me. I can no longer reproduce their issues as that would often mean that I have to re-install my machine or perform steps which would risk breaking my installation by accident.

Having a dedicated machine to tinker with would again allow me to improve my work on Debian installer, resume maintaining it long-term, fix the scripts contained in the `reform-tools` package involving flashing firmware or disk images to eMMC or NVMe and create and test new scripts. My concrete TODO list is:

- upload `reform-tools` package to Debian main
- create and upload a `reform-firmware` package to Debian main
- `debian-installer` support in main
- fix the `reform-setup-wizard` package so that it can transition to testing
- integration into `openqa.debian.net` (Phil Hands already helped me with that)
- create a firmware flashing tool using rescue media (Chris Hofstaedtler already assisted me with that)

All of these components above involve destructive low-level operations on the hardware. I would feel much more confident in being able to maintain all of these components long-term and develop new utilities, had I access to a dedicated machine to test all of this on.

Why should I be the recipient of this funding?

As may be apparent by now, I'm a massive fan of the Reform and how MNT, the company behind it, is approaching the subject of open hardware, open development and interaction with the community. Figure @ref(fig:PA302061) shows my MNT Reform with the `imx8mq` and several modifications like an integrated USB UART adapter, a radio transceiver, a UMTS modem and an additional TRRS jack. After



Figure 7: My MNT Reform with customizations

looking for an open hardware device which can run Debian and which can be my daily driver instead of just a device to tinker on for over 17 years, the MNT Reform is what I have been dreaming of and more. According to my NVMe drive, my MNT Reform has been in operation for a total of 13171 hours or 549 days. I cannot imagine to lose interest in this platform for the years to come.

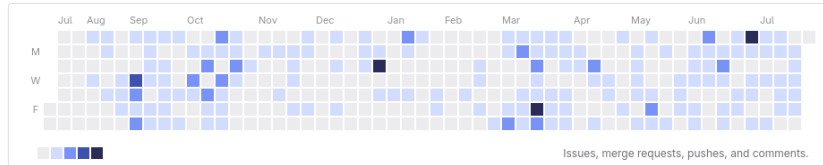


Figure 8: My activity on the <https://source.mnt.re> gitlab instance of MNT over the past year

I have essentially taken over the Debian integration and development of related tools for the MNT Reform which is also reflected in the credit section of the latest version of the MNT Reform handbook⁵. Since I started using the Reform at the end of 2021 I am the main contributor for the `reform-tools` repository (62% of all commits), the `reform-debian-packages` repository (71% of all commits) and the `reform-system-image` repository (44% of all commits). I have direct commit access to the MNT repositories and can only think of positive interaction with MNT regarding my contributions. My regular tasks include bug fixes, rebasing of the Linux kernel patch stack each time a new version gets uploaded to unstable by the Debian kernel team, adding support for new platforms to the tooling and (as all the software is built against Debian unstable) file bugs in the Debian bts once they affect us. One major example for this was a `binutils` bug which made the Linux kernel unbootable but only on `arm64`. This bug was first found by the MNT community⁶.

⁵ <https://mntre.com/reform2/handbook/credits.html>

I am the sole maintainer of `reform.debian.net` (hosting is kindly provided by the provided by the `debian.net` team⁷) which provides a Debian package repository, bootable disk images and `debian-installer` images for the MNT Reform but (in contrast to the content provided by MNT) built against Debian stable (and backports) and signed with my GPG key. All my work is bit-by-bit reproducible, all scripts producing the content can be run locally without requiring superuser privileges (making use of my packages `mmdebstrap` and `sbuild`) and are cross-buildable from `amd64`. My work thus directly benefits other Debian teams like the `cross-building` and `reproducible-builds` team. For example as part of my work for `reform.d.n` I found a bug which violates the principle that the package name/version/architecture triplet uniquely identifies a package in Debian⁸.

⁶ <https://bugs.debian.org/1074112>

⁷ <https://wiki.debian.org/Teams/DebianNet>

I am also deeply involved with the community on the forums⁹ where I have so far read over three thousand posts and created over

⁸ <https://bugs.debian.org/1072205>

⁹ <https://community.mnt.re/>

470 posts of my own (compared to about 500 posts from Lukas), receiving over 500 “likes” for my content. Same goes for IRC where, according to the logs¹⁰ I have contributed 22% of all messages (compared to 23% posts from Lukas). At Mini DebConf Berlin 2024 I gave a lightning talk¹¹ about my software work for the MNT Reform and met other Debian Reform owners (Martin Borgert) as well as plenty Reform-curious folks.

Remaining Challenges

Some of the CPU modules available for the Reform (like the imx8mq) require proprietary firmware. Unfortunately, the license of these firmware files makes them even non-distributable by nonfree¹². I have contacted the Linux NXP team about this issue and hope that NXP can grant a license exception to Debian¹³.

Funding Application

I hereby ask for funding of an MNT Reform laptop with a Rockchip RK3588 module in it with the purpose of carrying out above outlined development tasks over the course of the next Debian stable release and hopefully beyond. The cost for that is 1,400.00 EUR¹⁴.

¹⁰ <https://mntre.com/reform-irc-logs/>

¹¹ <https://mister-muffin.de/p/J1VD.pdf>

¹² <https://bugs.debian.org/1035055>

¹³ <https://lore.kernel.org/r/all/172079266731.2928366.7031846226118749814@localhost/>

¹⁴ <https://shop.mntre.com/products/mnt-reform>