

NemoTablet project

A DIY Tablet concept with a FOSS operating system

Aleksi Suomalainen

aleksi.suomalainen@protonmail.com

<https://locusf.wordpress.com/>

<https://twitter.com/locusf/>

January 27, 2016

Starters

Nemo Mobile
Glacier UI

SailPi

SailPi tablet and
Hackaday
NemoTabletFreedom aspects
of hardware
adaptationsFuture aspects
and adaptations

Closing words

- Nemo Mobile contributor and community member since 2013 summer
- Part of MeeGo Integration Team 2010-2012
- SailfishOS community member
- Currently employed at Symbio as software engineer
- As such, any opinions or statements made in this presentation are mine and mine only

Nemo Mobile began as a new project from the MeeGo Community Edition, mainly including all the middleware packages like lipstick, back at 2011. This was after Mer Project was launched by David Greaves and Carsten Munk, which had its roots in the MeeGo Core. I became involved at the summer of 2013, when Nemo Mobile spawned a UI project called Glacier UI.

NemoTablet project

Aleksi Suomalainen

Starters

Nemo Mobile
Glacier UI

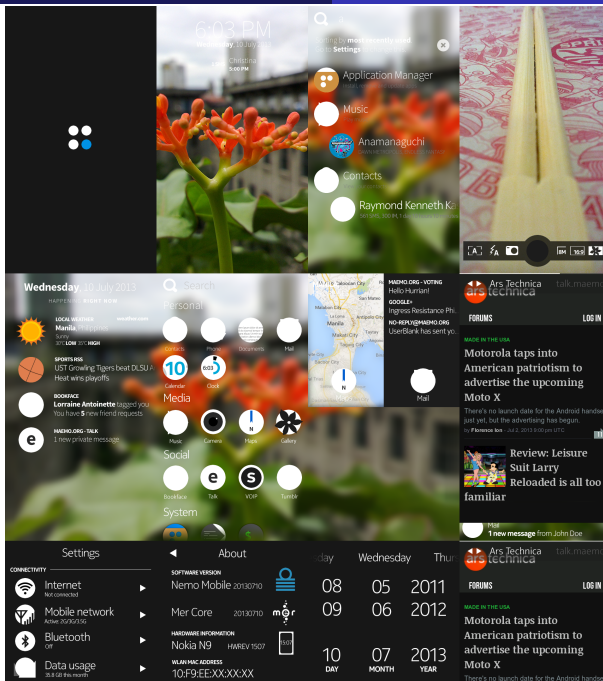
SailPi

SailPi tablet and Hackaday
NemoTablet

Freedom aspects of hardware adaptations

Future aspects and adaptations

Closing words



Starting from Kenneth Kasilags concept art, current Nemo Mobile team started to work on the components needed to create his concept as closely as possible. With Michael Demetriou as design lead, the team proceeded quite fast to create the initial components, like windows, pages, buttons etc. I am part of implementing the components and homescreen, by creating these alongside my masters thesis (unfortunately only available in finnish *here*).

NemoTablet
projectAleksi
Suomalainen

Starters

Nemo Mobile
Glacier UI

SailPi

SailPi tablet and
Hackaday
NemoTabletFreedom aspects
of hardware
adaptationsFuture aspects
and adaptations

Closing words

Glacier UI has been in a bit of a hiatus from spring 2015, but that doesn't mean interest on it has been low. Due to the recent events at Jolla Oy, it has received some attention as a possible alternative, in the situation that Jolla might some day seize existing (which I don't that they would do).

SailPi began quite shortly after FOSDEM 2015, once Raspberry Pi 2 was launched. Matias Lindström, SailPi co-creator, contacted me on the idea of creating a blog about running SailfishOS on Raspberry Pi 2.

SailPi was first adapted to be a DIY phone running SailfishOS, which turned out quite well, except the modem initialization was not simple to do at the startup of ofono. This is because the modem board required specific reset signals, between 2 second intervals and ofono was already started before it was initialized. Also my first touchscreen choice meant that the reset signal GPIO was wired to the backlight of the touchscreen.

NemoTablet
projectAleksi
Suomalainen

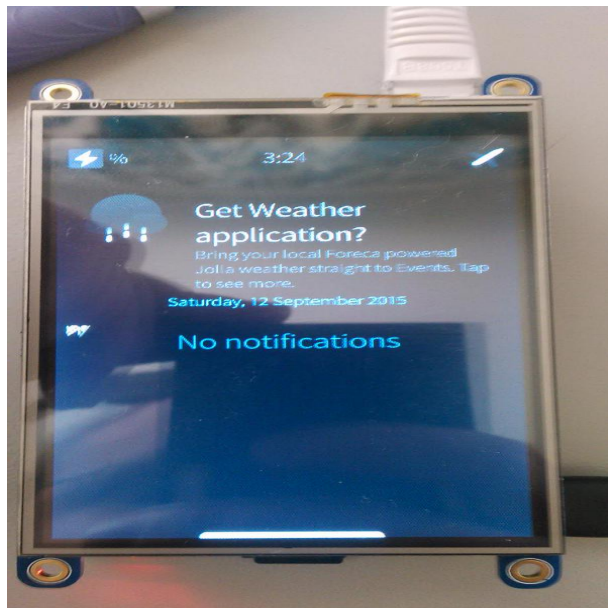
Starters

Nemo Mobile
Glacier UI

SailPi

SailPi tablet and
Hackaday
NemoTabletFreedom aspects
of hardware
adaptationsFuture aspects
and adaptations

Closing words



SailPi was limited to only use Glacier homescreen at first, since unlike other hardware adaptations, it didn't require a special shader which turned eg. an application upside down. I had to patch Glacier homescreen to do this, but I couldn't freely do the same to SailfishOS, due to the QML being copyrighted. Albeit I know that patches are allowed but still I couldn't possibly have touched something nonfree and then ship a patch as something like that should be freely customizable in the first place.

After receiving a working touchscreen from Adafruit, it was possible to use framebuffer copying technique to reflect the UI from the HDMI framebuffer to the SPI touchscreen. This was just a proof of concept, since it had quite slow refresh rate, due to the SPI bus being limited to CPU.

SailPi project proceeded to go into a hibernation as Jolla was implementing SailfishOS 2.0 user interface. This change meant that eventually, SailPi didn't have to rely on Glacier homescreen as primary UI and Jollas UI was used instead. This was due to the shader problem being fixed, although it is still unknown what was the actual fix.

SailPi tablet was created in September of 2015, when Raspberry Pi foundation released their officially supported touchscreen for the Raspberry Pi product family. At the Mikkeli EU codeweek, I presented the SailPi tablet for the first time and also submitted the article tip to Hackaday. It was presented on the site October 18, 2015 and received some attention worldwide. SailPi blog still has the most visitors to date from that day.

After SailPi tablet success, my attention turned to making the adaptation of Nemo Mobile, which SailPi project was first tested conceptually. Having been thinking about this for a some time since SailPi tablet, I thought it would make for a nice presentation. Initially, the touchscreen didn't work at all on the tablet adaptation. Reason was left unknown for a while, as I didn't have time to check what was changed between the six month period of the Mer/Nemo merger, which meant that a lot had been going on the middleware side of things.

Latest developments have since fixed this issue and the touchscreen works very well. Only problem is that the mce (mode control entity) does not support the power saving features that would turn it off, meaning that the touchscreen stays on all the time. This has significant issues to power management.

As Nemo Mobile is entirely FOSS, it is to be made clear that the adaptation on which it runs, has the possibility to have nonfree parts on it. Nemo Mobile adaptations started from the N9xx devices, all of which the adaptations contained proprietary drivers from both SGX and TI. Usually, the degree of freedom of the adaptations varies quite much.

In NemoTablet case, using Raspberry Pi 2 has its problems, with regards to freedom. Raspberry Pi 2 has blobs in its very bootup sequence and graphics drivers are nonfree as well. The most "free as in freedom" hardware, that Nemo Mobile has run on, is the Minnowboard Max, which was adapted during the sprint of 2015.

NemoTablet
projectAleksi
Suomalainen

Starters

Nemo Mobile
Glacier UI

SailPi

SailPi tablet and
Hackaday
NemoTabletFreedom aspects
of hardware
adaptationsFuture aspects
and adaptations

Closing words

Future adaptations will include Fairphone 2, UP board and possible PINE 64 board. Any help regarding the porting to these boards are very welcome and deeply appreciated. Some of these hardware boards are questionable, when it comes to the freedom of their hardware, but I see the trend towards more openness of the hardware and designs.

Other adaptations, for example libhybris adaptations, usually use Cyanogenmod as their base operating system. Replicant has been tried by me late 2014, but at that moment it turned out that graphics acceleration needed to make QtWayland to work didn't function. Replicant still remains the target for base system on the adaptation side, but in reality, the only feasible options are either AOSP or Cyanogenmod. This fact digs some holes under the freedom of Nemo Mobile.

As closing words, I'd like to argue that Nemo Mobile is still free software. It will also remain free software. This means that, along with the freedoms that free software provides to its users for days to come, it will also provide a perhaps lesser known concept of free software.

This is the concept of ownerless software. Unlike the freedoms that free software provides to its users, ownerless software is a concept that is a property of software itself. Ownerless software means that it is independent of its authors, which may come as companies or individual contributors. It has more potential to continue its existence due to this fact. (See Richard Stallmans essays, Why Software Should Not Have Owners)

Questions?