Mainlining the reMarkable 2 eInk tablet

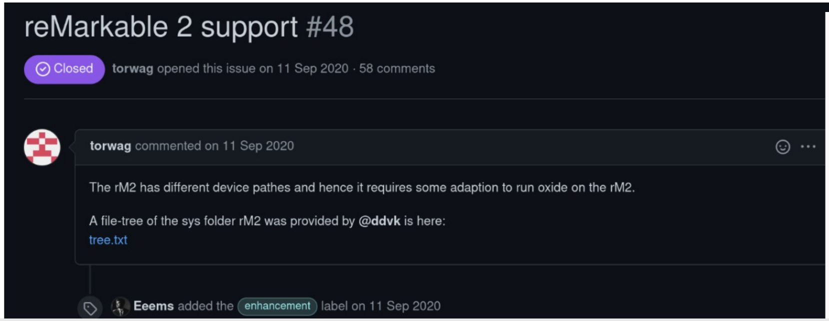
Alistair Francis <alistair@alistair23.me>
@al:alistair23.me

reMarkable 2 (rM2)

- eInk Tablet
- Uses i.MX7 SoC
- Ships with 4.14 vendor kernel

https://remarkable.com/

Existing Community



reMarkable 2 support #48

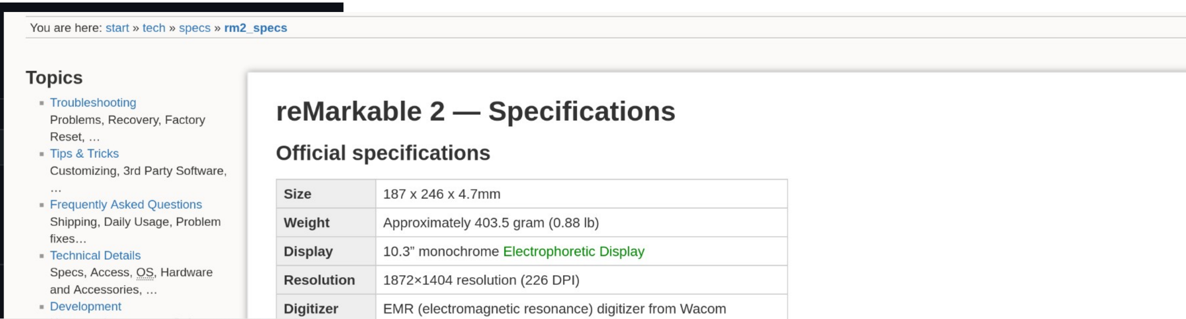
Closed **torwag** opened this issue on 11 Sep 2020 · 58 comments

torwag commented on 11 Sep 2020

The rM2 has different device pathes and hence it requires some adaption to run oxide on the rM2.

A file-tree of the sys folder rM2 was provided by **@ddvk** is here:
tree.txt

Eeems added the enhancement label on 11 Sep 2020



You are here: start » tech » specs » **rm2_specs**

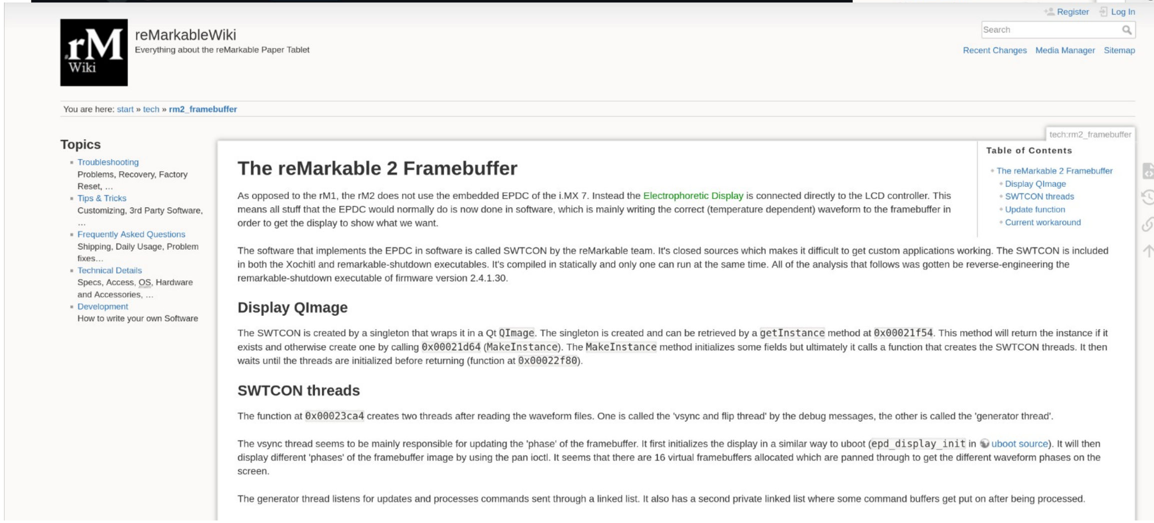
Topics

- Troubleshooting
  Problems, Recovery, Factory Reset, …
- Tips & Tricks
  Customizing, 3rd Party Software, …
- Frequently Asked Questions
  Shipping, Daily Usage, Problem fixes…
- Technical Details
  Specs, Access, OS, Hardware and Accessories, …
- Development

reMarkable 2 — Specifications

Official specifications

Size	187 x 246 x 4.7mm
Weight	Approximately 403.5 gram (0.88 lb)
Display	10.3" monochrome Electrophoretic Display
Resolution	1872×1404 resolution (226 DPI)
Digitizer	EMR (electromagnetic resonance) digitizer from Wacom
ige	8GB
	1024 MB LPDDR3 SDRAM
ry	3000 mAh
	1.2 GHz dualCore ARM CPU



Register Log In

reMarkableWiki
Everything about the reMarkable Paper Tablet

Search

Recent Changes Media Manager Sitemap

You are here: start » tech » **rm2_framebuffer**

Topics

- Troubleshooting
  Problems, Recovery, Factory Reset, …
- Tips & Tricks
  Customizing, 3rd Party Software, …
- Frequently Asked Questions
  Shipping, Daily Usage, Problem fixes…
- Technical Details
  Specs, Access, OS, Hardware and Accessories, …
- Development
  How to write your own Software

tech:rm2_framebuffer

Table of Contents

The reMarkable 2 Framebuffer

As opposed to the rM1, the rM2 does not use the embedded EPDC of the i.MX 7. Instead the Electrophoretic Display is connected directly to the LCD controller. This means all stuff that the EPDC would normally do is now done in software, which is mainly writing the correct (temperature dependent) waveform to the framebuffer in order to get the display to show what we want.

The software that implements the EPDC in software is called SWTCON by the reMarkable team. It's closed sources which makes it difficult to get custom applications working. The SWTCON is included in both the Xochitl and remarkable-shutdown executables. It's compiled in statically and only one can run at the same time. All of the analysis that follows was gotten be reverse-engineering the remarkable-shutdown executable of firmware version 2.4.1.30.

Display QImage

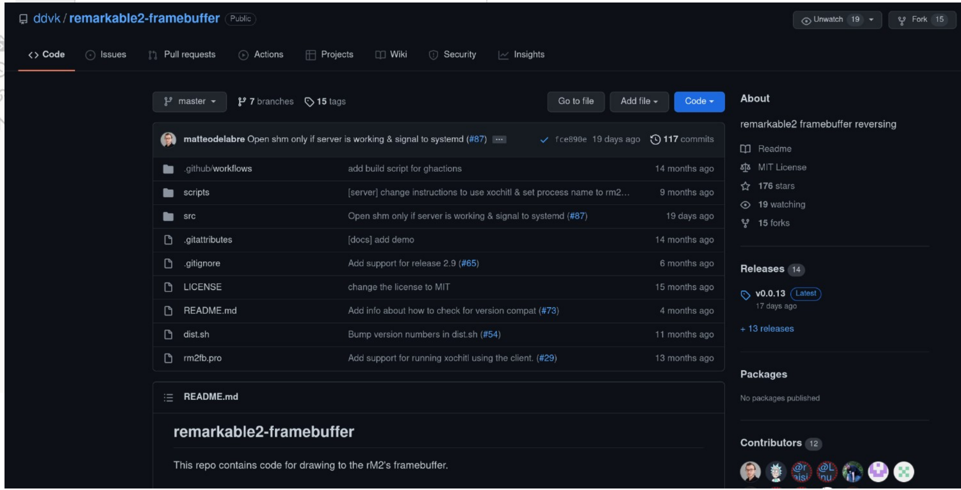
The SWTCON is created by a singleton that wraps it in a Qt `QImage`. The singleton is created and can be retrieved by a `getInstance` method at `0x00021f54`. This method will return the instance if it exists and otherwise create one by calling `0x00021d64` (`MakeInstance`). The `MakeInstance` method initializes some fields but ultimately it calls a function that creates the SWTCON threads. It then waits until the threads are initialized before returning (function at `0x00022f80`).

SWTCON threads

The function at `0x00023ca4` creates two threads after reading the waveform files. One is called the 'vsync and flip thread' by the debug messages, the other is called the 'generator thread'.

The vsync thread seems to be mainly responsible for updating the 'phase' of the framebuffer. It first initializes the display in a similar way to uboot (`epd_display_init` in uboot source). It will then display different 'phases' of the framebuffer image by using the pan ioctl. It seems that there are 16 virtual framebuffers allocated which are panned through to get the different waveform phases on the screen.

The generator thread listens for updates and processes commands sent through a linked list. It also has a second private linked list where some command buffers get put on after being processed.



ddvk / **remarkable2-framebuffer** Public

7 branches 15 tags

matteodelabre Open shm only if server is working & signal to systemd (#87) … ✓ fce890e 19 days ago 117 commits

.github/workflows	add build script for ghactions	14 months ago
scripts	[server] change instructions to use xochitl & set process name to rm2…	9 months ago
src	Open shm only if server is working & signal to systemd (#87)	19 days ago
.gitattributes	[docs] add demo	14 months ago
.gitignore	Add support for release 2.9 (#65)	6 months ago
LICENSE	change the license to MIT	15 months ago
README.md	Add info about how to check for version compat (#73)	4 months ago
dist.sh	Bump version numbers in dist.sh (#54)	11 months ago
rm2fb.pro	Add support for running xochitl using the client. (#29)	13 months ago

README.md

remarkable2-framebuffer

This repo contains code for drawing to the rM2's framebuffer.

About

remarkable2 framebuffer reversing

Readme
MIT License
176 stars
19 watching
15 forks

Releases 14

v0.0.13 Latest
17 days ago

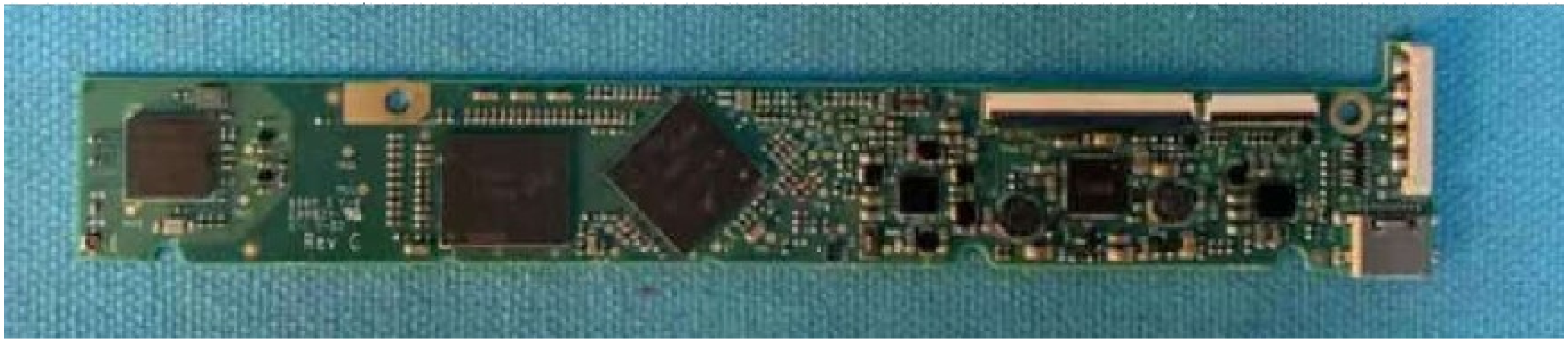
+ 13 releases

Packages

No packages published

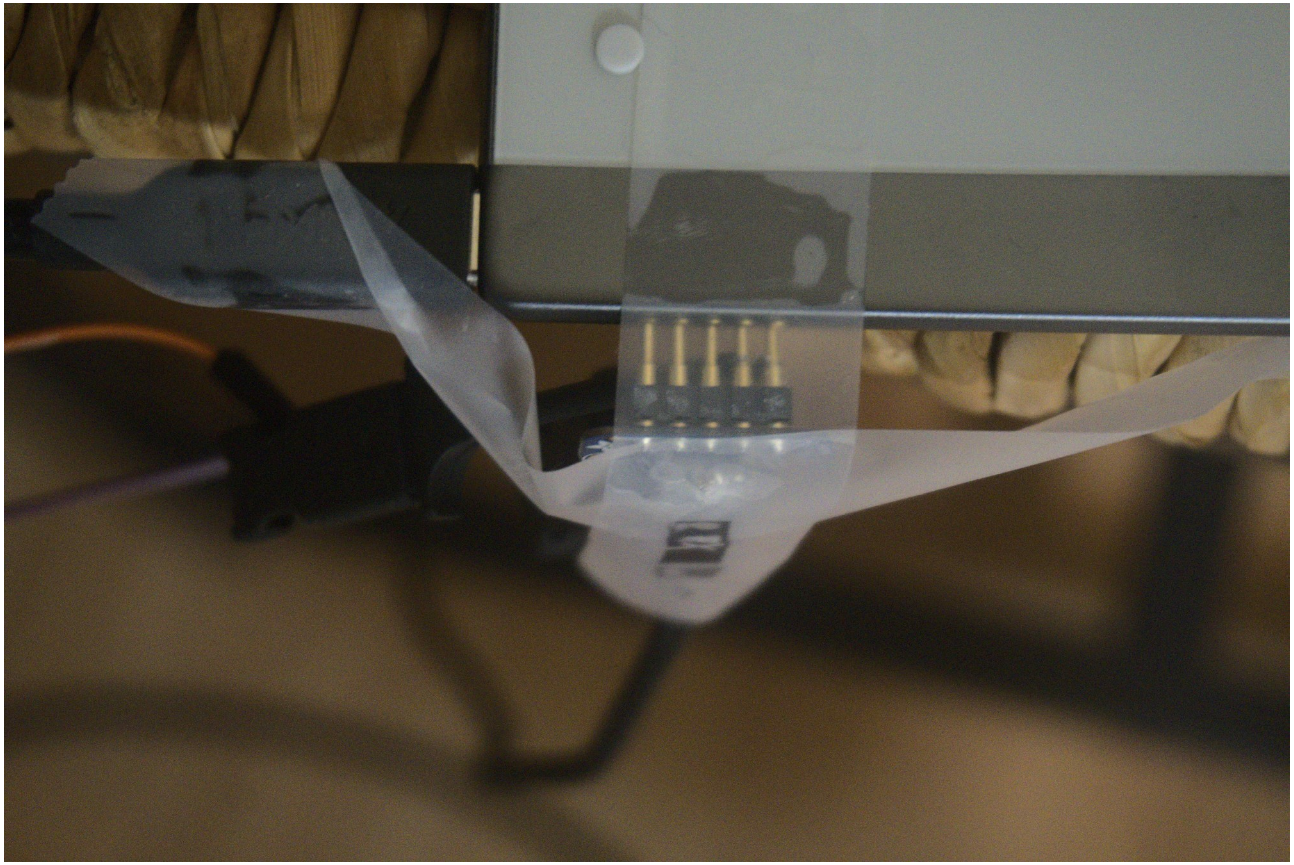
Contributors 12

Initial Investigation

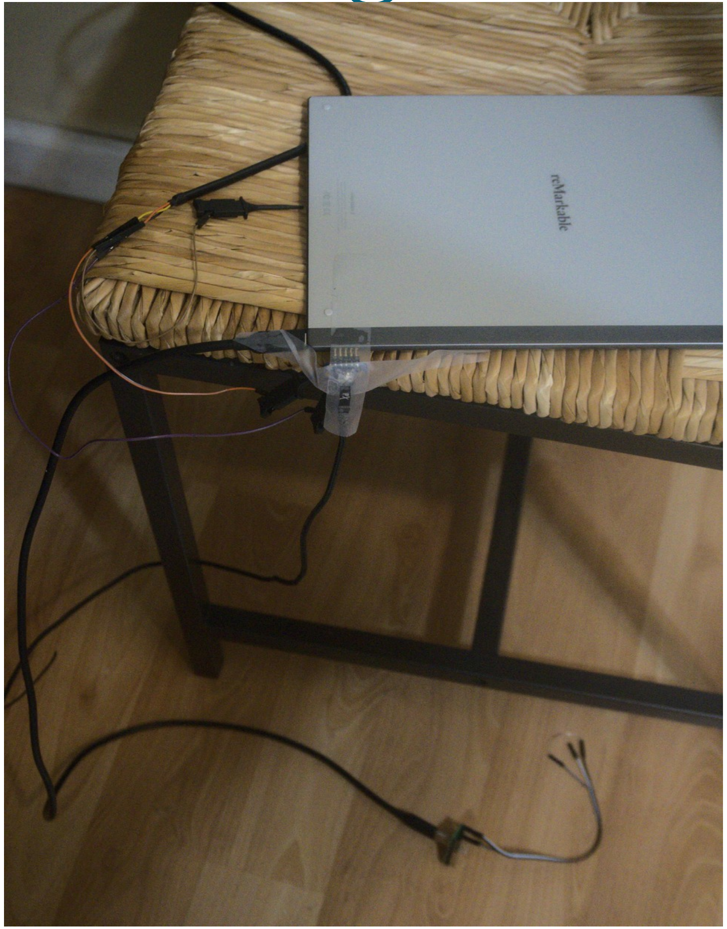


https://fccid.io/2AMK2-RM110/Internal-Photos/Internal-photos-4523931.pdf

Accessing the hardware

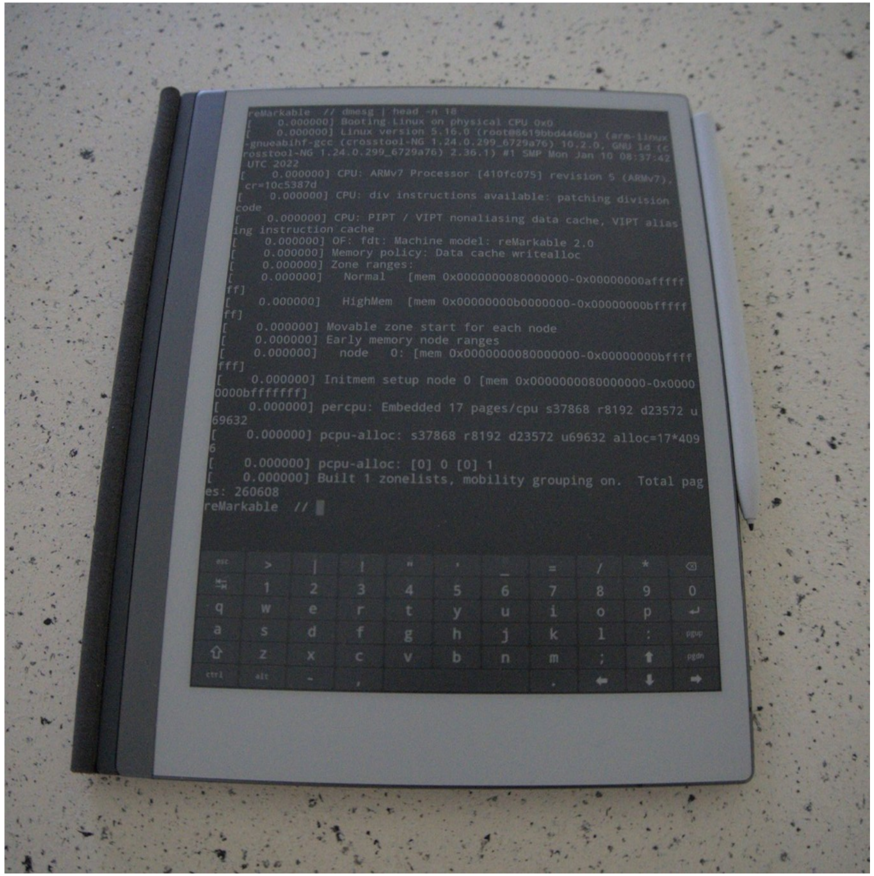


Accessing the hardware



- The pogo pins provide UART TX and debug downloader mode (with uuu)

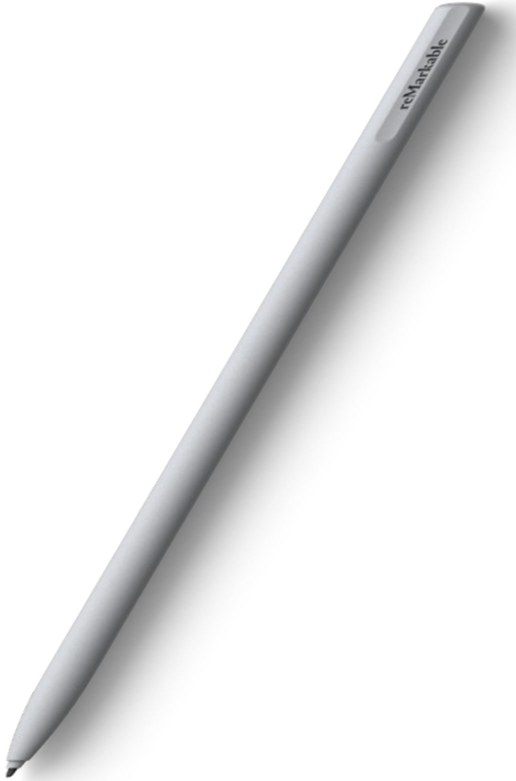
Bringup and WiFi

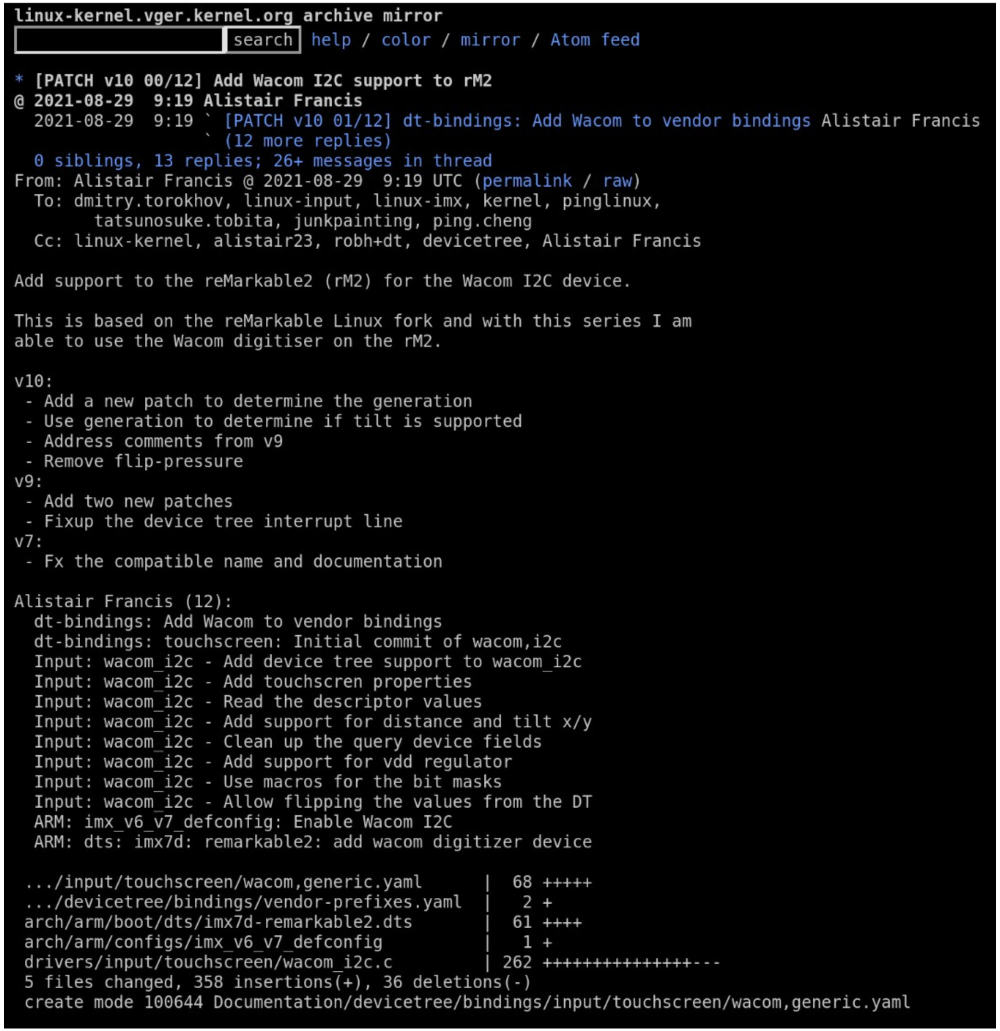


```
reMarkable  // dmesg | head -n 18
[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 5.16.0 (root@8619bbd446ba) (arm-linux
-gnueabihf-gcc (crosstool-NG 1.24.0.299_6729a76) 10.2.0, GNU ld (c
rosstool-NG 1.24.0.299_6729a76) 2.36.1) #1 SMP Mon Jan 10 08:37:42
 UTC 2022
[    0.000000] CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7),
 cr=10c5387d
[    0.000000] CPU: div instructions available: patching division
code
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT alias
ing instruction cache
[    0.000000] OF: fdt: Machine model: reMarkable 2.0
[    0.000000] Memory policy: Data cache writealloc
[    0.000000] Zone ranges:
[    0.000000]   Normal   [mem 0x0000000080000000-0x00000000afffff
ff]
[    0.000000]   HighMem  [mem 0x00000000b0000000-0x00000000bfffff
ff]
[    0.000000] Movable zone start for each node
[    0.000000] Early memory node ranges
[    0.000000]   node   0: [mem 0x0000000080000000-0x00000000bffff
fff]
[    0.000000] Initmem setup node 0 [mem 0x0000000080000000-0x0000
0000bfffffff]
[    0.000000] percpu: Embedded 17 pages/cpu s37868 r8192 d23572 u
69632
[    0.000000] pcpu-alloc: s37868 r8192 d23572 u69632 alloc=17*409
6
[    0.000000] pcpu-alloc: [0] 0 [0] 1
[    0.000000] Built 1 zonelists, mobility grouping on.  Total pag
es: 260608
reMarkable  //
```

- OpenEmbedded RootFS
  - https://github.com/Freescale/meta-freescale-3rdparty/blob/master/conf/machine/imx7d-remarkable2.conf

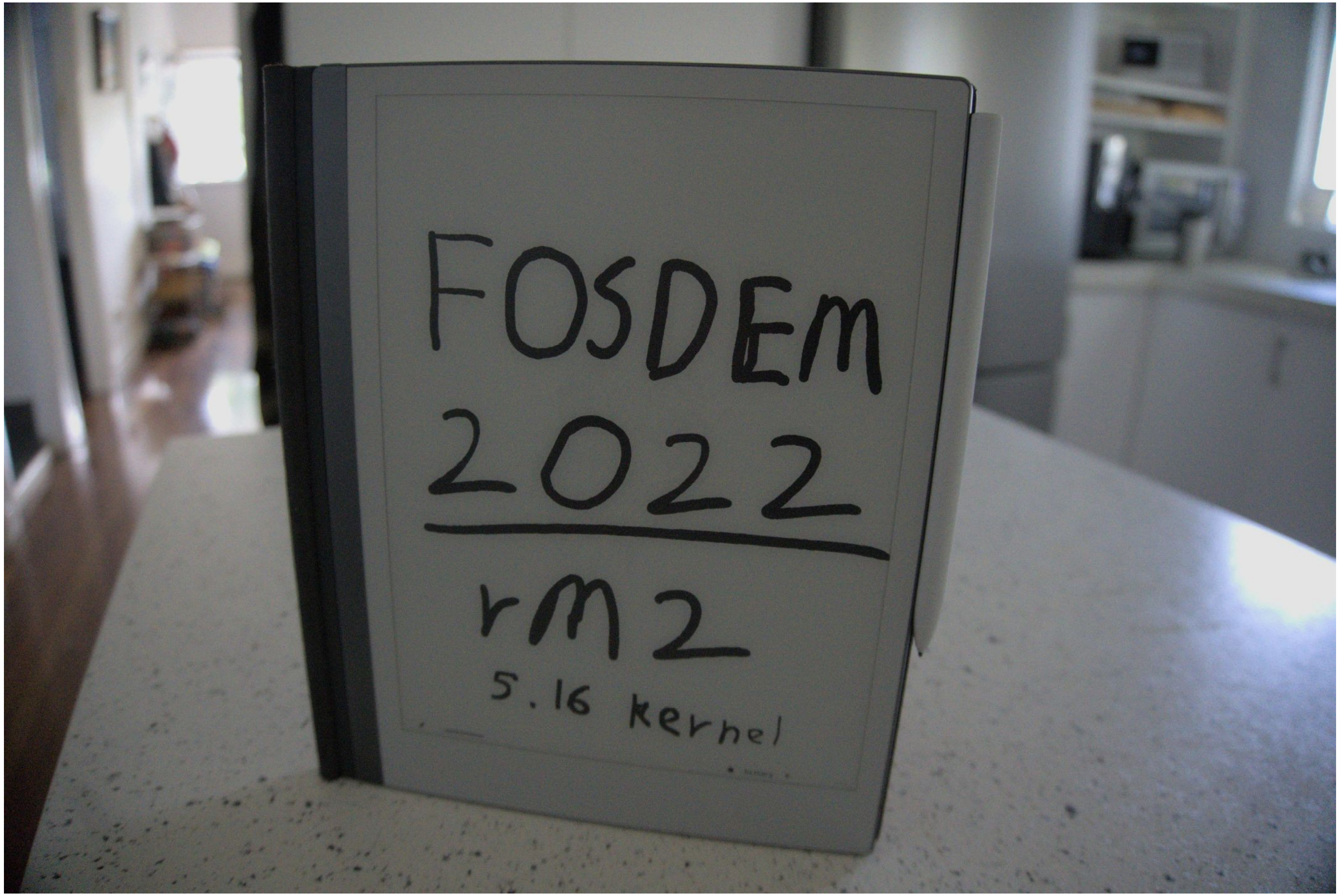
I2C Wacom





```
linux-kernel.vger.kernel.org archive mirror
                    search  help / color / mirror / Atom feed

* [PATCH v10 00/12] Add Wacom I2C support to rM2
@ 2021-08-29  9:19 Alistair Francis
  2021-08-29  9:19 ` [PATCH v10 01/12] dt-bindings: Add Wacom to vendor bindings Alistair Francis
                   ` (12 more replies)
  0 siblings, 13 replies; 26+ messages in thread
From: Alistair Francis @ 2021-08-29  9:19 UTC (permalink / raw)
  To: dmitry.torokhov, linux-input, linux-imx, kernel, pinglinux,
        tatsunosuke.tobita, junkpainting, ping.cheng
  Cc: linux-kernel, alistair23, robh+dt, devicetree, Alistair Francis

Add support to the reMarkable2 (rM2) for the Wacom I2C device.

This is based on the reMarkable Linux fork and with this series I am
able to use the Wacom digitiser on the rM2.

v10:
 - Add a new patch to determine the generation
 - Use generation to determine if tilt is supported
 - Address comments from v9
 - Remove flip-pressure
v9:
 - Add two new patches
 - Fixup the device tree interrupt line
v7:
 - Fx the compatible name and documentation

Alistair Francis (12):
  dt-bindings: Add Wacom to vendor bindings
  dt-bindings: touchscreen: Initial commit of wacom,i2c
  Input: wacom_i2c - Add device tree support to wacom_i2c
  Input: wacom_i2c - Add touchscren properties
  Input: wacom_i2c - Read the descriptor values
  Input: wacom_i2c - Add support for distance and tilt x/y
  Input: wacom_i2c - Clean up the query device fields
  Input: wacom_i2c - Add support for vdd regulator
  Input: wacom_i2c - Use macros for the bit masks
  Input: wacom_i2c - Allow flipping the values from the DT
  ARM: imx_v6_v7_defconfig: Enable Wacom I2C
  ARM: dts: imx7d: remarkable2: add wacom digitizer device

 .../input/touchscreen/wacom,generic.yaml      |  68 +++++
 .../devicetree/bindings/vendor-prefixes.yaml  |   2 +
 arch/arm/boot/dts/imx7d-remarkable2.dts       |  61 ++++
 arch/arm/configs/imx_v6_v7_defconfig          |   1 +
 drivers/input/touchscreen/wacom_i2c.c         | 262 +++++++++++++++---
 5 files changed, 358 insertions(+), 36 deletions(-)
 create mode 100644 Documentation/devicetree/bindings/input/touchscreen/wacom,generic.yaml
```

eInk Panel



Debugging: Sleep reset

https://www.embarcados.com.br/servindo-watchdog-adequadamente/

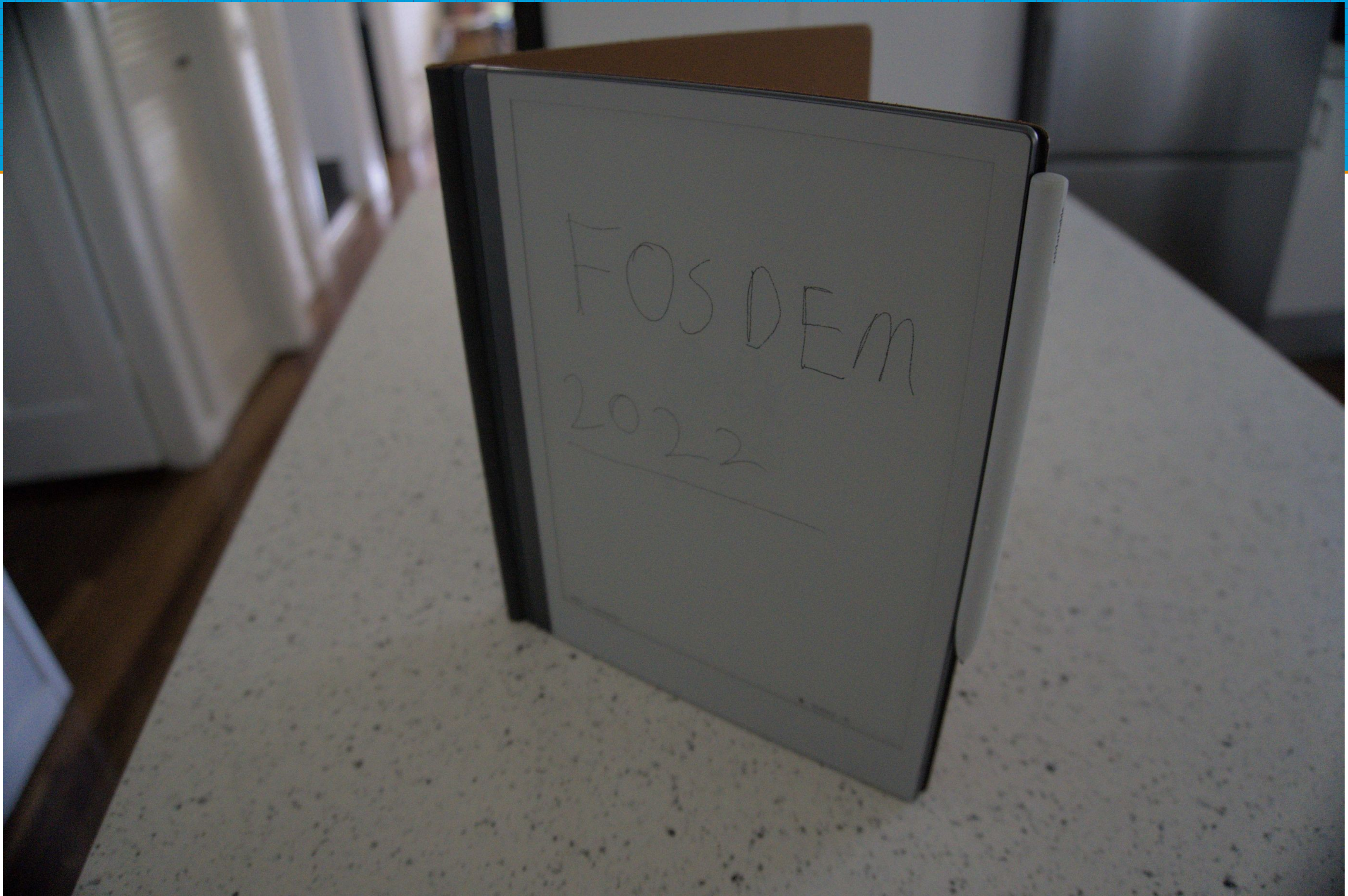
Mainline Status

- 5.17 kernel
  - Full boot support
  - WiFi support
  - Wacom touchscreen

- Patches pending
  - silergy,sy7636a MFD
  - LCD support
  - Cyprus Cyttsp5 touchscreen support
  - i.MX Watchdog ping support

Next steps

- Get current patches on list merged
- Get the Toltec package merged
  - https://github.com/toltec-dev/toltec/pull/488
- Get low power mode working
- Fix up paper cut bugs
- Reverse engineer eInk driver
  - https://github.com/matteodelabre/waved


FOSDEM
2022