

# Run upstream coreboot on an ARM Chromebook

Paul Menzel

October 26, 2017

# Who am I?

- ▶ (Economic) Mathematician by studies at TU Berlin
- ▶ Free Software enthusiast
- ▶ Active in coreboot since 2005 (still LinuxBIOS back then)
- ▶ System architect at Max Planck Institute for Molecular Genetics

# Google Chromebooks

x86

ARM

# Samsung Chromebook Plus (RK3399)

See thread *Current, BLOB free laptop available Europe?* on coreboot mailing list

- ▶ Device with Rockchip RK3399, but only available in the USA
- ▶ No BLOBs in firmware

## Linux support

BLOBs required for

- ▶ hardware video decoding
- ▶ Wi-Fi and Bluetooth
- ▶ GPU support

## Acer Chromebook R 13

- ▶ On October 24th, 2017, 384 € at [notebooksbilliger.de](http://notebooksbilliger.de)

## Specifications

---

Processor	Mediatek MT8173C 4x 2.10 GHz
Cache	1 MB
RAM	4 GB LPDDR3, PC3L-12800 (1600MHz)
Format	2in1 Convertible
Display size	33 cm (13,3")
Display	Acer CineCrystal™ Multi-Touch Full-HD IPS Display with
Resolution	1920 x 1080 Pixel (Full HD)
IGD:	PowerVR GX6250
eMMC	32 GB
Dimensions	326 x 228 x 15,5 mm (B x T x H)
Weight	1,49 kg
Battery time	up to 12 hours
Capacity	4.670 mAh

---

# BLOB status

## No BLOBs

- ▶ Embedded Controller

## BLOBs

- ▶ PCM firmware in ARM Trusted Firmware
- ▶ Maybe USB C device
- ▶ Maybe ANX7688: PD + HDMI→DP converter

*It contains a firmware that we update from the AP-FW, at boot time, which is the only reason to have a driver for it in depthcharge.*

See commit 9859ac55 (anx7688: Add support for ANX7688) in Depthcharge.

## Linux support

## Mediatek device and coreboot

- ▶ Google Oak reference design
- ▶ Acer Chromebook R 13 is Google Elm variant of Google Oak



# TLDR

```
$ make crossgcc-arm crossgcc-aarch64 CPUS=160  
$ make menuconfig
```

Select Google Elm, Chrome OS, and Depthcharge

```
$ make
```

Copy to Chromebook, deactivate write protection.

```
$ flashrom -p internal -w coreboot.rom
```

# Libettereboot

- ▶ New build system for Libreboot written by Paul K.

```
$ git clone https://git.code.paulk.fr/libettereboot.git
$ cd libettereboot
$ for project in coreboot depthcharge vboot arm-trusted-firmware; do
$ # gelöschte Firmware in arm-trusted-firmware wieder einbauen
$ ./libreboot cook coreboot depthcharge elm
```

## Board status

- ▶ Upload to board status repository

Longer version

# Developer mode and write protection

## Developer mode

1. Key combination
2. Ctrl + d
3. Data is deleted

Now type `shell` in Crosh Shell to get GNU Bash.

## Write protection

1. Open device
2. Remove screw

Look at shipped image

# Components

1. Vboot2
2. ARM Trusted Firmware

# ARM Trusted Firmware

*ARM Trusted Firmware provides a reference implementation of secure world software for ARMv8-A, including a Secure Monitor executing at Exception Level 3 (EL3). It implements various ARM interface standards, such as:*

- ▶ *The Power State Coordination Interface (PSCI)*
- ▶ *Trusted Board Boot Requirements (TBBR, ARM DEN0006C-1)*
- ▶ *SMC Calling Convention*
- ▶ *System Control and Management Interface*

*As far as possible the code is designed for reuse or porting to other ARMv8-A model and hardware platforms.*

*ARM will continue development in collaboration with interested parties to provide a full reference implementation of Secure Monitor code and ARM standards to the benefit of all developers working with ARMv8-A TrustZone technology.*



# License

- ▶ BSD-3-Clause

# Operating system

1. Chrome OS
2. Arch Linux
3. Debian GNU/Linux

# Chrome OS

- ▶ Ships Linux 3.18
- ▶ Boot in 10 seconds with shipped firmware

# Depthcharge

- ▶ Configure default boot

## FMAP regions and fallback

- ▶ Goal: Similar setup to shipped image with fallback

Questions?

My notes