

# Thoughts about AOSP on Raspberry Pi 4

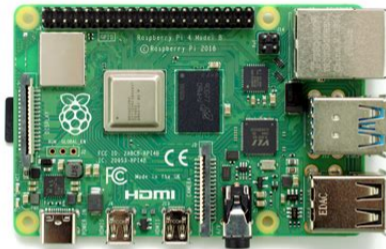
Chris Simmonds, 2net

# Thoughts about AOSP on Raspberry Pi 4

- It would be nice to have a platform for building and testing AOSP on real hardware
- Why RPi4?
  - Raspberry Pi is a standard
  - Raspberry Pi Organization tend to support hardware for a long time
  - Cheap
  - (usually) easy to get hold of

# The Raspberry Pi 4B

- CPU: BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 1, 2, **4**, **8** GiB SDRAM
- Storage: Micro SD card
- USB: 4 x full size USB A host connectors: 2 x USB 3.0; 2 x USB 2.0
- Ethernet: 1 x Gbit
- WiFi: 2.4 GHz and 5 GHz 802.11 ac
- Bluetooth: 5.0, BLE
- Graphics: 2 x HDMI video output
- CSI and DSI connectors for camera and LCD
- 40-pin header for HATs



# Hasn't it been done already?

Sure! Here are some notable projects

- **Android RPi:** <https://github.com/android-rpi>
- **Snapp Automotive:**  
[https://github.com/snappautomotive/firmware-device\\_snappautomotive\\_rpi](https://github.com/snappautomotive/firmware-device_snappautomotive_rpi)
- **LineageOS:** (unofficial build from KonstaKang)  
<https://konstakang.com/devices/rpi4/LineageOS20/>
- There is even a Google group: <https://groups.google.com/g/android-rpi>

But, none do the full AOSP thing

# Project aims

- Clean AOSP build for tablet and Automotive (and maybe TV?)
- SELinux in enforcing mode
- Pass CTS and VTS
- Super partition
- A/B partition slots and working OTA
- Working recovery mode
- GKI kernel
- dm-verity/AVB enabled
- ADB over USB (optional)
- Fastboot over USB (optional)

# Project

`https://github.com/aospandaaos/a4rpi-local-manifest`

## Status

It builds, boots U-Boot and Linux, loads Android but cannot start framework

Some issues with HALs

# Things to consider

- Booting
- Kernel
- Storage layout
- AOSP device configuration
- Graphics

# Booting

- RPi bootloader does not have any Android integration
- So, RPi boot -> U-Boot -> Linux -> AOSP
- U-Boot provides
  - Boot Control Block (misc partition) - needed for Recovery and OTA
  - A/B slot handling
  - boot.img handling - so we can update the kernel and ramdisk
  - AVB
  - fastboot

RPi boot files From <https://github.com/raspberrypi/firmware.git> (-b master)

fixup4.dat, start4.elf

U-Boot From <https://gitlab.denx.de/u-boot/u-boot> (-b v2022.10)



# Booting: status

## Done

- U-Boot builds - using external toolchain
- Bootscript to load kernel image from bootloader partition

## To do

- load boot.img - need to build in AOSP; need to load boot.img in U-Boot script
- recovery mode - U-Boot needs to read BCB; AOSP needs to build recovery
- A/B slot handling
- Android Verified Boot
- Fastboot

# Kernel

## Now

- Kernel from Android rpi
- 5.15, based on up-stream RPi kernel, but integrated into recent android-kernel build (using build.sh)

## Future

- ARM64 GKI
- all RPi specific code moved into vendor modules
- (non trivial amount of work)

# Storage layout

## Now

1	64 MiB	bootloader	FAT32
2	2048 MiB	system	EXT4
3	256 MiB	vendor	EXT4
4	512 MiB	userdata	EXT4

## Future

1	64 MiB	bootloader	FAT32
2	1 MiB	misc	
3	64 MiB	boot_a	
4	64 MiB	boot_b	
5	64 KiB	vbmata	
6	4096 MiB	super	
7	2048 MiB	userdata	F2FS

There is a script to write to SD card in the "current" format:

```
$AOSP/scripts/write-sdcard-rpi4.sh
```

# AOSP device configuration

- Where to start?
  - cuttlefish - example of AOSP done the Google way
  - yukawa/vim3 - example of modern AOSP on dev board
  - Android RPi - RPi specifics
- So far, there is one device, `$AOSP/device/a4rpi/rpi4`, with two products
  - tablet: `rpi4_tablet`
  - car: `rpi4_auto`

# Graphics

- Raspberry Pi has VC4 / V3D video controller and GPU
- Supported by **mesa3d**
  - but not the version shipped in `$AOSP/external/mesa3d`
- Android-rpi project hosts a working version (currently v22.2) in `external_mesa3d`

# ADB over USB

- ADB over network is fine, but not 100% reliable in my experience
- ADB over USB is the more natural way
- RPi4 has USB OTG on the USB C port, but ...
  - usually the USB C port supplied the power (c. 2A)
  - most PC USB ports supply 500 mA
  - so, need extra power via GPIO, or PoE (?)

# Fastboot

- Why?
  - no longer need to keep swapping SD cards round
  - it's the right thing to do
- work involved ... not assessed

# Next steps

- All help, pull requests, issue reports gratefully accepted