Running Android on the Raspberry Pi

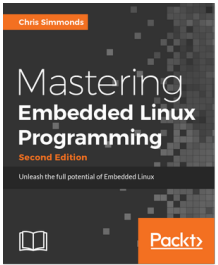
Android Pie meets Raspberry Pi

Chris Simmonds

Android Makers 2019

License

About Chris Simmonds



- Consultant and trainer
- Author of *Mastering Embedded Linux Programming*
- Working with embedded Linux since 1999
- Android since 2009
- Speaker at many conferences and workshops

"Looking after the Inner Penguin" blog at `http://2net.co.uk/`

@2net_software

`https://uk.linkedin.com/in/chrisdsimmonds/`

Why?

- Porting Android to a dev board is a great way to learn about Android

Why?

- Porting Android to a dev board is a great way to learn about Android
- It's a good testing ground for new ideas

Why?

- Porting Android to a dev board is a great way to learn about Android
- It's a good testing ground for new ideas
- It's fun! No, really it is!

What do you need to run Android?

- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties

What do you need to run Android?

- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties
- Running a recent version of Linux kernel (v4.4 or later)

What do you need to run Android?

- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties
- Running a recent version of Linux kernel (v4.4 or later)
- With at least 512 MiB RAM

What do you need to run Android?

- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties
- Running a recent version of Linux kernel (v4.4 or later)
- With at least 512 MiB RAM
- And at least 1 GiB flash storage - e.g. eMMC, SD card

What do you need to run Android?

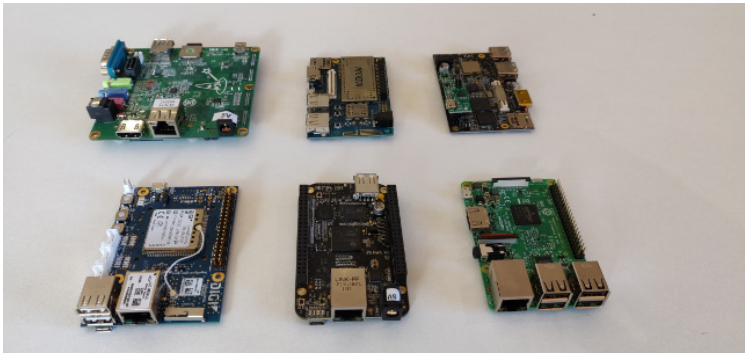
- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties
- Running a recent version of Linux kernel (v4.4 or later)
- With at least 512 MiB RAM
- And at least 1 GiB flash storage - e.g. eMMC, SD card
- Plus a touchscreen or external display - e.g. HDMI

What do you need to run Android?

- Hardware from one of the supported architectures
  - ARM, x86 or MIPS, in 32 or 64 bit varieties
- Running a recent version of Linux kernel (v4.4 or later)
- With at least 512 MiB RAM
- And at least 1 GiB flash storage - e.g. eMMC, SD card
- Plus a touchscreen or external display - e.g. HDMI
- And a GPU with OpenGL ES 2.0 libraries (more about this later)

Android on dev boards

DragonBoard, Hikey, BeagleBone, WandBoard, Raspberry Pi, Digi ConnectCore ...



Why Raspberry Pi?

- It's cheap ($35)

Why Raspberry Pi?

- It's cheap ($35)
- Easy to get hold of

Why Raspberry Pi?

- It's cheap ($35)
- Easy to get hold of
- Hackable

Why Raspberry Pi?

- It's cheap ($35)
- Easy to get hold of
- Hackable
- **Because it is there**

The Raspberry Pi 3B

- BCM2837, Cortex-A53 (ARMv8) 64-bit SoC @ 1.2GHz
- 1 GiB SDRAM
- Micro SD card slot
- 4 full size USB 2.0 A host
- 100 Mbit Ethernet
- WiFi 802.11 a/b/g/n/ac
- Bluetooth 4.2/BLE
- HDMI video output
- 40-pin header for HATs

Hasn't it been done already?

Sure! Here are some notable projects

- **Android RPi**: https://github.com/android-rpi
- **LineageOS**: (unofficial build from KonstaKang) https://konstakang.com/devices/rpi3/LineageOS15.1
- **RTAndroid**: https://embedded.rwth-aachen.de/doku.php?id=en:tools:rtandroid
  - based on research by Igor Kalkov, now merged into emteria.os
- **emteria.os**: https://emteria.com (not open source)
- **Android Things**: https://developer.android.com/things/hardware/raspberrypi (not open source)

What do you need?

- A copy of the Android Open Source Project (AOSP)

What do you need?

- A copy of the Android Open Source Project (AOSP)
- A Linux kernel with Android extensions

What do you need?

- A copy of the Android Open Source Project (AOSP)
- A Linux kernel with Android extensions
- A fair knowledge of the hardware

What do you need?

- A copy of the Android Open Source Project (AOSP)
- A Linux kernel with Android extensions
- A fair knowledge of the hardware
- All the help you can get from existing projects

What do you need?

- A copy of the Android Open Source Project (AOSP)
- A Linux kernel with Android extensions
- A fair knowledge of the hardware
- All the help you can get from existing projects
- A fairly fast computer

What do you need?

- A copy of the Android Open Source Project (AOSP)
- A Linux kernel with Android extensions
- A fair knowledge of the hardware
- All the help you can get from existing projects
- A fairly fast computer
- Time and patience

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS
- My version of the code is at
  https://github.com/csimmonds/a4rpi-local-manifest

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS
- My version of the code is at
  `https://github.com/csimmonds/a4rpi-local-manifest`
- Challenges posed by the Raspberry Pi

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS
- My version of the code is at
  `https://github.com/csimmonds/a4rpi-local-manifest`
- Challenges posed by the Raspberry Pi
  - Bootloader

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS
- My version of the code is at
  `https://github.com/csimmonds/a4rpi-local-manifest`
- Challenges posed by the Raspberry Pi
  - Bootloader
  - Graphics

Putting Android on Raspberry Pi

- What follows is based on Konsta's port of LineageOS
- My version of the code is at
  `https://github.com/csimmonds/a4rpi-local-manifest`
- Challenges posed by the Raspberry Pi
  - Bootloader
  - Graphics
  - Lack of USB OTG port

Getting AOSP

- **AOSP** is all of the open source components of the Android O/S

Getting AOSP

- **AOSP** is all of the open source components of the Android O/S
- Begin by getting the **repo** tool

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

Getting AOSP

- **AOSP** is all of the open source components of the Android O/S
- Begin by getting the **repo** tool

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

- Get the **manifest** of the version you want

```
$ mkdir ~/myandroid
$ cd myandroid
$ repo init -u https://android.googlesource.com/platform/manifest -b android-9.0.0_r35
```

Getting AOSP

- **AOSP** is all of the open source components of the Android O/S
- Begin by getting the **repo** tool

```
$ curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
$ chmod a+x ~/bin/repo
```

- Get the **manifest** of the version you want

```
$ mkdir ~/myandroid
$ cd myandroid
$ repo init -u https://android.googlesource.com/platform/manifest -b android-9.0.0_r35
```

- Copy all the code (*)

```
$ repo sync
```

(*) 70 GiB worth

Customizing AOSP

- Porting Android to new hardware means creating a new **Board Support Package**

Customizing AOSP

- Porting Android to new hardware means creating a new **Board Support Package**
- Consisting of **device configuration** (next slide), **Hardware Abstraction Layer** and other low level packages

Customizing AOSP

- Porting Android to new hardware means creating a new **Board Support Package**
- Consisting of **device configuration** (next slide), **Hardware Abstraction Layer** and other low level packages
- Put these into a **local manifest**, `.repo/local_manifests/default.xml`

Customizing AOSP

- Porting Android to new hardware means creating a new **Board Support Package**
- Consisting of **device configuration** (next slide), **Hardware Abstraction Layer** and other low level packages
- Put these into a **local manifest**, `.repo/local_manifests/default.xml`

```
$ git clone https://github.com/csimmonds/a4rpi-local-manifest .repo/local_manifests -b pie
```

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>
  <remote name="a4rpi" fetch="https://github.com/csimmonds" />
  <remote name="lineage" fetch="https://github.com/lineage-rpi"/>

  <project path="device/rpiorg/rpi3" name="a4rpi-device" revision="pie" remote="a4rpi"/>
[...]
```

Device configuration

- Device specific configuration is in **device/[organisation]/[product]**
  - Example: `device/rpiorg/rpi3`

Device configuration

- Device specific configuration is in **device/[organisation]/[product]**
  - Example: `device/rpiorg/rpi3`
- Select the one you want with command **lunch**

```
$ source build/envsetup.sh
$ lunch rpi3-eng
```

Selecting the base product

- Base product is selected in device configuration by `device.mk`

```
$(call inherit-product, $(SRC_TARGET_DIR)/product/aosp_base.mk)
```

- Products in AOSP

aosp_base.mk	Android tablet
aosp_base_telephony.mk	Android phone
atv_base.mk	Android TV
car.mk	Android Automotive

Building AOSP

```
m
```

Building AOSP

```
m
```

Then go and make a cup of tea: it's going to take a couple of hours

When the build is over

- You will find images in `out/target/product/[product name]`

```
out/target/product/rpi3/cache.img
out/target/product/rpi3/ramdisk.img
out/target/product/rpi3/system.img
out/target/product/rpi3/userdata.img
out/target/product/rpi3/vendor.img
```

- Just write these to a micro SD card (Raspberry Pi)
- ... or use **fastboot** to flash them (everything except Raspberry Pi)

But wait, what about Linux?

- Android runs on top of **Linux**

But wait, what about Linux?

- Android runs on top of **Linux**
- So, you need a Linux kernel with the Android kernel patches

But wait, what about Linux?

- Android runs on top of **Linux**
- So, you need a Linux kernel with the Android kernel patches
- Usually provided by the silicon vendor (Qualcomm, NXP, Rockchip, ...)

But wait, what about Linux?

- Android runs on top of **Linux**
- So, you need a Linux kernel with the Android kernel patches
- Usually provided by the silicon vendor (Qualcomm, NXP, Rockchip, ...)
- For Raspberry Pi, use Konsta's kernel:

```
$ git clone https://github.com/lineage-rpi/android_kernel_brcm_rpi3 -b lineage-15.1 rpi3
$ cd rpi3
$ make lineageos_rpi3_defconfig
$ make -j $(nproc) zImage
$ make dtbs
```

Challenge 1: Booting the Raspberry Pi

- Most Android devices have a bootloader that supports the **fastboot** protocol for flashing and booting images

Challenge 1: Booting the Raspberry Pi

- Most Android devices have a bootloader that supports the **fastboot** protocol for flashing and booting images
- Raspberry Pi does not have such a bootloader

Challenge 1: Booting the Raspberry Pi

- Most Android devices have a bootloader that supports the **fastboot** protocol for flashing and booting images
- Raspberry Pi does not have such a bootloader
- So, we have to package the kernel and initial ramdisk and use Broadcom loader to start Linux

Challenge 1: Booting the Raspberry Pi

- Most Android devices have a bootloader that supports the **fastboot** protocol for flashing and booting images
- Raspberry Pi does not have such a bootloader
- So, we have to package the kernel and initial ramdisk and use Broadcom loader to start Linux
- ... To update Android, you need to take out the SD card and reflash ...

Boot files

- The RPi boots from the first partition of the SD card

```
Bootloader:
    bootcode.bin
    fixup.dat
    fixup_x.dat
    start.elf
    start_x.elf
    config.txt

Linux:
    cmdline.txt
    bcm2710-rpi-3-b.dtb
    overlays/
    zImage

Android:
    ramdisk.img
```

Challenge 2: Graphics

- We need OpenGL ES 2.0 libraries **with Android extensions**

Challenge 2: Graphics

- We need OpenGL ES 2.0 libraries **with Android extensions**
- Three options

Challenge 2: Graphics

- We need OpenGL ES 2.0 libraries **with Android extensions**
- Three options
  - Get binary OpenGLES libraries from the vendor, if they exist (they don't for Broadcom BCM2708/2835)

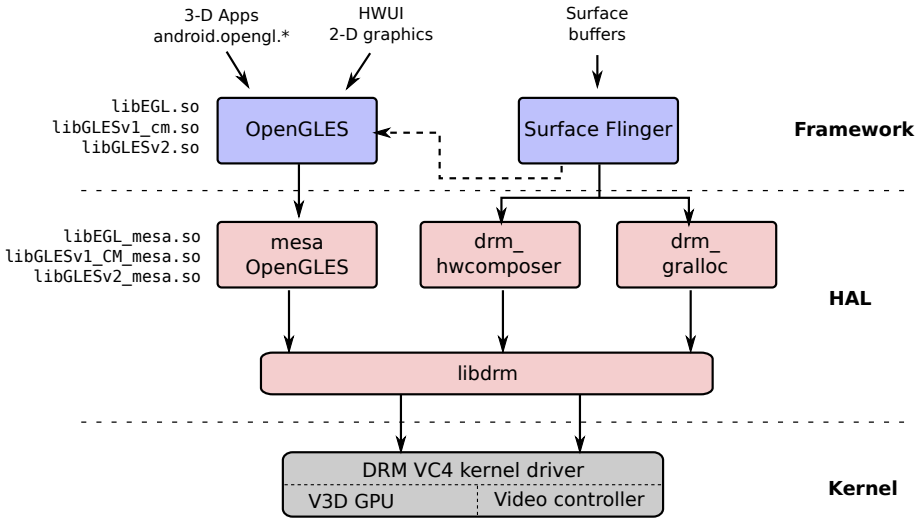
Challenge 2: Graphics

- We need OpenGL ES 2.0 libraries **with Android extensions**
- Three options
  - Get binary OpenGLES libraries from the vendor, if they exist (they don't for Broadcom BCM2708/2835)
  - Use open source drivers, **Mesa** and **DRM**, if they exist (they do for BCM2708/2835)

Challenge 2: Graphics

- We need OpenGL ES 2.0 libraries **with Android extensions**
- Three options
  - Get binary OpenGLES libraries from the vendor, if they exist (they don't for Broadcom BCM2708/2835)
  - Use open source drivers, **Mesa** and **DRM**, if they exist (they do for BCM2708/2835)
  - Use Soft GPU, **Swiftshader**

Graphics: Mesa



Mesa

- Mesa 3D library: https://www.mesa3d.org/

Mesa

- Mesa 3D library: https://www.mesa3d.org/
- Supports OpenGL ES 1/2/3 and EGL

Mesa

- Mesa 3D library: https://www.mesa3d.org/
- Supports OpenGL ES 1/2/3 and EGL
- Drivers for mobile GPUs include:

Driver	SoC
freedreno	Qualcomm MSM
nouveau	NVidea Tegra
VC4	Broadcom BCM2708 (Raspberry Pi)
etnaviv	NXP i.MX6
lima/panfrost	ARM Mali 4xx, Txxx and Gxx
softpipe	soft GPU

DRI/DRM

- **DRI** (Direct Rendering Infrastructure): user-space wrappers for DRM, including **libdrm**

DRI/DRM

- **DRI** (Direct Rendering Infrastructure): user-space wrappers for DRM, including **libdrm**
- **DRM** (Direct Rendering Manager): kernel device driver to access GPU and video controller hardware blocks

DRI/DRM

- **DRI** (Direct Rendering Infrastructure): user-space wrappers for DRM, including **libdrm**
- **DRM** (Direct Rendering Manager): kernel device driver to access GPU and video controller hardware blocks

```
# ls -l /dev/dri/*
crw-rw-rw- 1 root graphics 226,   0 1970-01-01 00:00 /dev/dri/card0
crw-rw-rw- 1 root graphics 226, 128 1970-01-01 00:00 /dev/dri/renderD128
```

Open source graphics go mainstream

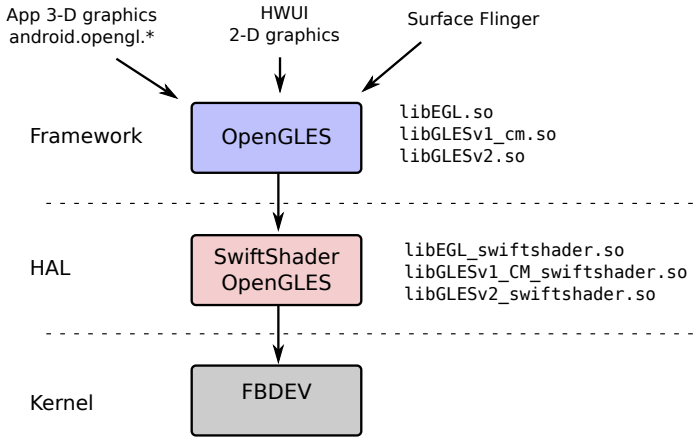
- Google is encouraging vendors to move to mesa/DRM

Open source graphics go mainstream

- Google is encouraging vendors to move to mesa/DRM
- For example
  - Pixel 3 has a Qualcomm 845 SoC with Adreno 630 GPU
  - Uses Mesa/DRM with **freedreno** driver

See Alistair Strachan's presentation at Linux Plumber's conference, November 2018, **DRM/KMS for Android**
https://linuxplumbersconf.org/event/2/contributions/229/attachments/53/60/10._DRM_KMS_for_Android_v1.pdf

Graphics: Swiftshader



Challenge 3: ADB

- Raspberry Pi only has USB **host** ports, but ADB needs a USB **peripheral** port

Challenge 3: ADB

- Raspberry Pi only has USB **host** ports, but ADB needs a USB **peripheral** port
  - Usually provided by a dual mode USB "On The Go" (OTG) port

Challenge 3: ADB

- Raspberry Pi only has USB **host** ports, but ADB needs a USB **peripheral** port
  - Usually provided by a dual mode USB "On The Go" (OTG) port
  - (Actually, the BCM283x has OTG hardware but it is used internally to bridge the USB host controller, Ethernet, and so on)

Challenge 3: ADB

- Raspberry Pi only has USB **host** ports, but ADB needs a USB **peripheral** port
  - Usually provided by a dual mode USB "On The Go" (OTG) port
  - (Actually, the BCM283x has OTG hardware but it is used internally to bridge the USB host controller, Ethernet, and so on)
- But, we can use ADB over Ethernet instead

```
$ adb connect Android.local
connected to Android.local:5555
```

Challenge 3: ADB

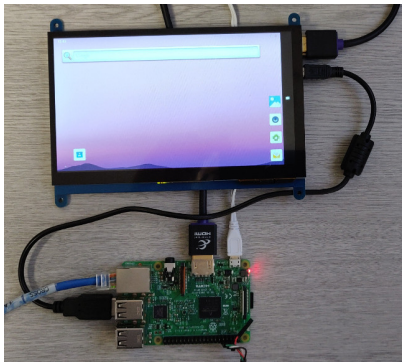
- Raspberry Pi only has USB **host** ports, but ADB needs a USB **peripheral** port
  - Usually provided by a dual mode USB "On The Go" (OTG) port
  - (Actually, the BCM283x has OTG hardware but it is used internally to bridge the USB host controller, Ethernet, and so on)
- But, we can use ADB over Ethernet instead

```
$ adb connect Android.local
connected to Android.local:5555
```

```
$ adb shell
rpi3:/ #
```

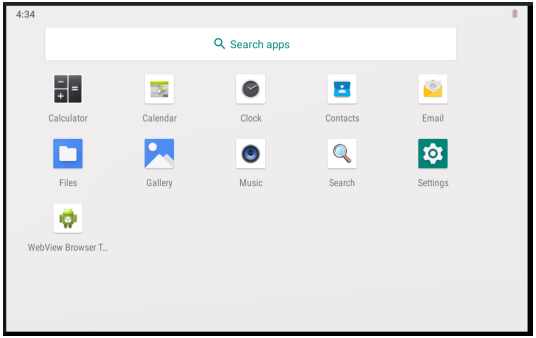
Current status

- Code is on github (URL at the end)
- Based on Android Pie 9.0 r 30
- Using SwiftShader
- Works:
  - HDMI touch screen
  - Ethernet
  - WiFi (probably)
  - Bluetooth (possibly)
- Not working:
  - Audio
  - Streaming video
  - Lots of other things...
- Early stages: still many things to do



Apps

- Has standard AOSP apps
- Install **Gapps** if you want PlayStore, etc
  `https://opengapps.org`
  - Note: this does not come with a license from Google



Delving deeper

- If you would like to discover more about building Android platforms, visit `http://www.2net.co.uk/training.html` and enquire about training classes for your company
  - 2net training is available world-wide

Relevant links:

Android 4 RPi
https://github.com/csimmonds/a4rpi-local-manifest

Slides on SlideShare
https://www.slideshare.net/chrissimmonds/
running-android-on-the-raspberry-pi-android-pie-meets-raspberry-pi

@2net_software http://www.2net.co.uk

Any questions?