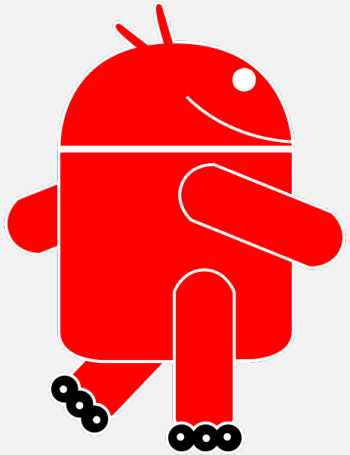
Replicant Keynote



Paul Kocialkowski
paulk@replicant.us

Saturday June 4^{th} 2016

Components, Issues and Freedom

Components and Implementations

Components of a mobile device:

- Processors
  - Primary processor
  - Auxiliary treatment processors
  - Communications processor
- Controllers
- Peripherals

Implementations:

- Hardware design
- Software

Current Situation and Issues

Situation and evolution:

- Amount of **programmed** components
- Treatment offloading
- **Data** storage and **communications**

Issues:

- **Trust** in technology
- **Control** of the devices
- **Knowledge** of the inner-workings, preservation

Degree of complexity

Basic Freedoms and Implementations

Guarantees: basic freedoms

1. Run for any purpose
2. Study and modify
3. Redistribution
4. Redistribution of modifications

Implementations regarding mobile devices:

- Free hardware design
- Free software

Freedom in Mobile Devices

Freedom in Mobile Devices

Free Hardware Design

Free Hardware Design

Technologies:

- Printed circuit board
- Integrated circuits

Liberting the hardware:

- Modifications, *source code* and design
- Formats and toolchains
- Costs and scale
- Infrastructure and trust

Free Hardware Design

Current situation:

- Possible to some limited extent
- Free integrated circuit designs examples:
  OpenRISC, OpenSPARC, LEON, LM32, lowRISC, etc
- Free printed circuit board designs examples:
  Arduino, Freeduino, USB armory, Novena, etc
- Documented hardware, "OpenHardware" confusion
- What about mobile devices?

Never as simple as:

```
$ ./configure
$ make
# make install
```

Freedom in Mobile Devices

Free Software, Constraints and Limitations

Liberating the Software

Liberating the software:

- Manufacturers' positions
  - Economic interest
  - Copyright (hardware blocks), patents
  - Copyleft and releases
  - Quality, maintainability (reference code)
- Reverse engineering work
- Required timed and resources
- Technical capabilities, recurrent limitations
- Long-term interest and obsolescence

Recurrent Limitations when Freeing the Software

Recurrent limitations:

- Hardware documentation, schematic, etc
- Technical knowledge and adapted tools
- Legal constraints (reverse engineering)
- Ability to replace the code:
  Read-only memories, secret interfaces, external access
- Ability to run our own code: verification
- Ability to debug the code

Once the software is free and works:

- Installation cost for users
- Associated risks

Software on Various Components

Various components running software:

- Processors (primary, auxiliary, communications)
- Controllers
- Peripherals

Various types of software:

- Firmwares
- Bootup software
- Trusted environment
- Operating system

Freedom in Mobile Devices

Firmwares

Firmwares and Free Software

Related hardware:

- Auxiliary processors
- Controllers and peripherals

Free software support:

- Specific hardware (Arduino, BusPirate, FX2LA)
- Wi-Fi peripherals (ath9k_htc, AR9170, OpenFWWF)

Mobile devices:

- **Proprietary** firmwares, rarely verified
- Pre-installed or loaded
- Wi-Fi, bluetooth, GPS, multimedia, cameras, ...

Overall, no free firmwares for mobile devices

Freedom in Mobile Devices

Communications Processor (modem)

Free Software for the Communications Processor (modem)

Communications processor (modem):

- Powerful processor
- Full-blown operating system
- Various R/F front-ends

Software for the modem:

- **Proprietary** systems (modern devices)
- Ability to replace the system, verification

Free software support:

- Free GSM stack: **OsmocomBB**
- Limitations: usability, support, certification

Tremendous problem for freedom and privacy/security

Communications Processor (modem) Isolation

Stakes for privacy/security:

- Hardware access
- Ability to spy on the user
- Ability to compromise the main processor?

Modem isolation:

- Workaround, other ways to spy
- Practical verification and trust:
  bad situation proof, integrated modem, schematics and free hardware design
- What about freedom issues?

Pragmatic solution, never fully reliable

Freedom in Mobile Devices

Primary Processor

Free Software on the Primary Processor

Software running on the primary processor:

- Bootup software
- Trusted software environment
- Operating system:
  - Kernel (Linux)
  - Hardware abstraction layers
  - *Frameworks*
  - Applications

Free Bootup Software

Bootup software:

- Bootrom: **proprietary**, read-only memory (hardware-like)
- Verification with **signatures**, chain of trust
  platform-specific, specific models
- Free bootup software: **U-Boot, Coreboot**
- Examples of good platforms:
  i.MX, Allwinner, OMAP (GP), Tegra (non-ODM), Rockchip
- What about mobile devices with these?

Free bootup software for a few mobile devices

Trusted Software Environment

Need for trusted software:

- Operating system is flawed
- Privileged operations, hardware access
- Sensitive operations (privacy/security)

Implementing a trusted software environment:

- Cooperation with the chip (**TrustZone**)
- Setup early (by bootup software)
- Privileged mode, Secure Monitor Call (SMC)

Trusted Software Environment and Freedom

Consequences for freedom:

- Not good or bad *per-se*
- Most privileged software
- **Privacy/security** implications
- Free software implementations
- Dependence on the bootloader situation
- **Proprietary** and **verified** implementations (recent devices)
- Known good examples: USB armory with i.MX53, Rockchip

Often problematic on recent devices, but could be done right

Free Software Operating System

Crucial for privacy/security:

- Access to controllers and peripherals
- Access to user data
- Access to user communications
- *Samsung Galaxy Back-door*

On the forefront of free software:

- Direct user interaction
- Understanding, modifications
- New versions, updates

Free Software Operating System

Status of each distinct layer:

- Kernel: Linux, **free**, modified versions, pass-through drivers
- Hardware abstraction layers: mostly **non-free**, mixed
- *Frameworks*: **mostly-free** systems such as:
  Android, FirefoxOS, Ubuntu Touch, OpenWebOS, etc
- Applications: various **free**
  Applications repository: **F-Droid**

Hardware Abstraction Layers

Usually-problematic aspects:

- Graphics acceleration, GPU
- GPS

Platform-specific status for some aspects:

- Cameras (GPU dependency)
- Audio

Dedicated project for complex aspects (GPUs):

- Freedreno (Adreno GPUs)
- Nouveau (nVidia GPUs)
- *Lima* (Mali GPUs)

Proprietary layers: avoid, privileges, access, knowledge

Summary and Possible Improvements

Summary, pragmatic approaches

Overview of the situation of mobile devices:

- Far from perfect situation
- Subject to compromising (data, communications)
- Stakes evaluation, threat model, level of trust
- Limited short term evolution
- Strong need for developers

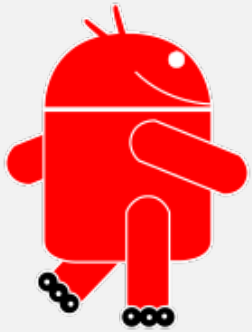
Long and middle-term approaches:

- Liberating the primary processor's operating system
- Selecting good platforms
- Interest in modem isolation
- Producing mobile devices

Summary and Possible Improvements

Free System: Replicant

Replicant: free system for mobile devices



Replicant

Ground ideas and values of the project:

- Fully free mobile system, Android-based
- Distribution and recommendation of solely free software (GNU FSDG)
- Focus on privacy/security
- Working and (somewhat) usable
- Information and documentation

Status of the Replicant Project

Current status of the project:

- **Very few** developers, spare time
- Few (but growing) external contributions (security)
- 12 supported devices:
  Samsung Galaxy, Nexus
- Missing features
- CyanogenMod 10.1, Android 4.2 base
- Financial support: **donations**

Latest images: **Replicant 4.2 0004**

Replicant-Supported Devices

Maintained

- Nexus S (I902x): 2011
- Galaxy S (I9000): 2012
- Galaxy S 2 (I9100): 2012
- Galaxy Note (N7000): 2013
- Galaxy Nexus (I9250): 2012
- Galaxy Tab 2 7.0 (P31xx): 2013
- Galaxy Tab 2 10.1 (P51xx): 2013
- Galaxy S 3 (I9300): 2013
- Galaxy Note 2 (N7100): 2014

Uncompleted

- GTA04: 2012

Unmaintained

- HTC Dream/Magic: 2010
- Nexus One: 2011

Challenges and Goals for the Future

Challenged for the future:

- Trust in upstream: CyanogenMod/OmniROM
- More recent versions, devices support
- Google applications et AOSP

Goals for the future:

- More recent version (6.0)
- Improved documentation
- Privacy/security improvements
- Better devices support

Improved Documentation, Privacy/Security Improvements

Wiki updates:

- Devices evaluations and support information:
  non-free software, privacy/security, modem isolation
- Research about more devices
- Failed projects documentation (GPS, etc)

Privacy/security improvements:

- Security-oriented version of Replicant?
  with loss of functionality
- Better devices support

Summary and Possible Improvements

Better Devices Support

Supported Devices, Freedom and Privacy/Security

Bad modem isolation Proprietary and signed bootloaders

Devices Selection for Replicant

Supporting better devices:

- Free hardware design
- Documented hardware
- Modem-less devices
- Modem isolation
- Free bootloaders
- Known protocols, free drivers

OpenPhoenux community:

- **GTA04**, **Letux** devices
- **Neo900**

Cheap Chinese devices:

- **Allwinner** tablets
- **Rockchip** tablets

Mass-produced devices:

- **Optimus Black** (LG)
- **Kindle Fire** first generation (Amazon)

Other devices **form factors**!

Learn more about Replicant:

- Website: **http://www.replicant.us/**
- Blog: **http://blog.replicant.us/**
- Wiki/tracker: **http://redmine.replicant.us/**

Join the community:

- Forums
- Mailing list
- IRC channel: **#replicant** at **Freenode**

The project needs you!

- Replicant deserves more than **a few developers**
- **Donations** are welcome (devices are expensive)


That's all Folks!