

# Linux on mobile devices

Jozef Mlích

*Department of Computer Graphics and Multimedia  
Brno University of Technology, Faculty of Information Technology  
Božetěchova 2, 612 66 Brno, Czech Republic  
[imlich@fit.vutbr.cz](mailto:imlich@fit.vutbr.cz)*



- Why? Who? How? What?
- Boot
- Filesystem and storage
- Kernel / drivers / api / interfaces
- Inter-process communication (DBUS)
- GUI (libhildon, Qt)
- Compilation,
- Cross compilation
- Remote debugging, Deployment
- Emulation / Virtualization
- Profiling
- Packaging
- Profit

- Lifetime of phone hardware is long
  - ~ 10 years
  - Except of battery
- Lifetime of software is much shorter
  - Android 2-3 years of updates
  - iOS up to 5 years of updates
  - New APIs , Web standards are changing
  - Support for new hardware / for old hardware
  - Security (CVEs)
- Could be old phone used as core of embedded system?
  - Main board? Webcam? HVAC? IoT?
  - Phone
  - ...

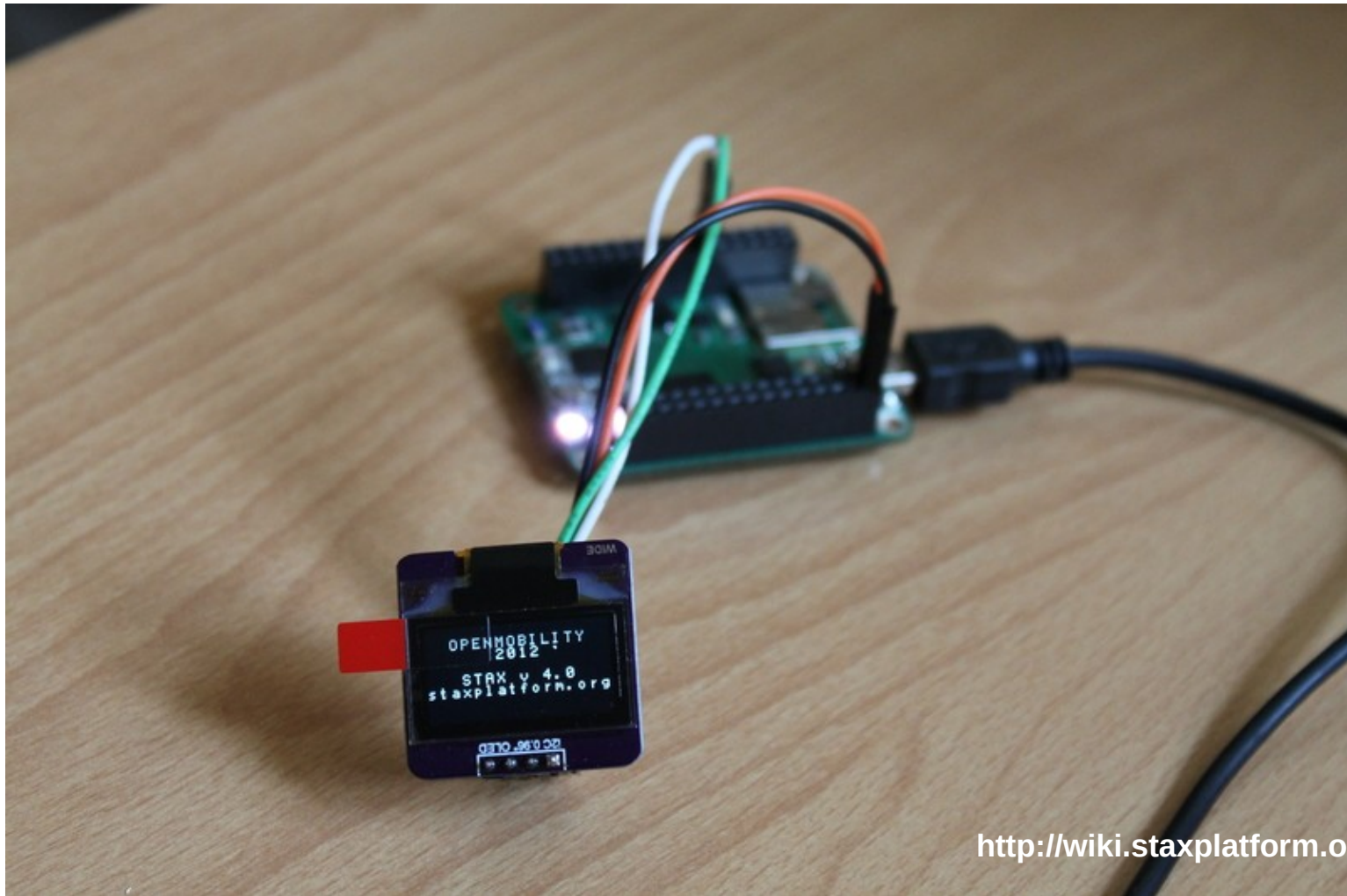
## Cemetery of mobile phones and it's operating systems

- Greenphone
- Openmoko – SHR, Hackable:1, QtMoko
- Symbian
- Nokia N900 – Maemo
- Nokia N9 – MeeGo Harmattan
- Firefox OS
- Tizen
- Blackberry Playbook, BB10
- Ubuntu Phone
- Windows Phone
- WebOS
- PalmOS
- Bada

## Some survivors of mobile phone evolution

- Jolla – SailfishOS - <https://jolla.com/>
- Neo900 - <http://neo900.org/>
- Mer project/Nemo
  - <https://wiki.merproject.org/wiki/Nemo>
  - <https://wiki.merproject.org/wiki/Adaptations/libhybris>
- Plasma mobile - <https://plasma-mobile.org/>
- Librem 5 – Purism - <https://puri.sm/shop/librem-5/>
- Dragonbox Pyra
  - <https://pyra-handheld.com/boards/pages/pyra/>
- Gemini PDA
  - <https://www.indiegogo.com/projects/gemini-pda-android-linux-keyboard-mobile->
- PostmarketOS
  - <https://wiki.postmarketos.org/wiki/Devices>

## What is actually mobile device?



<http://wiki.staxplatform.org>

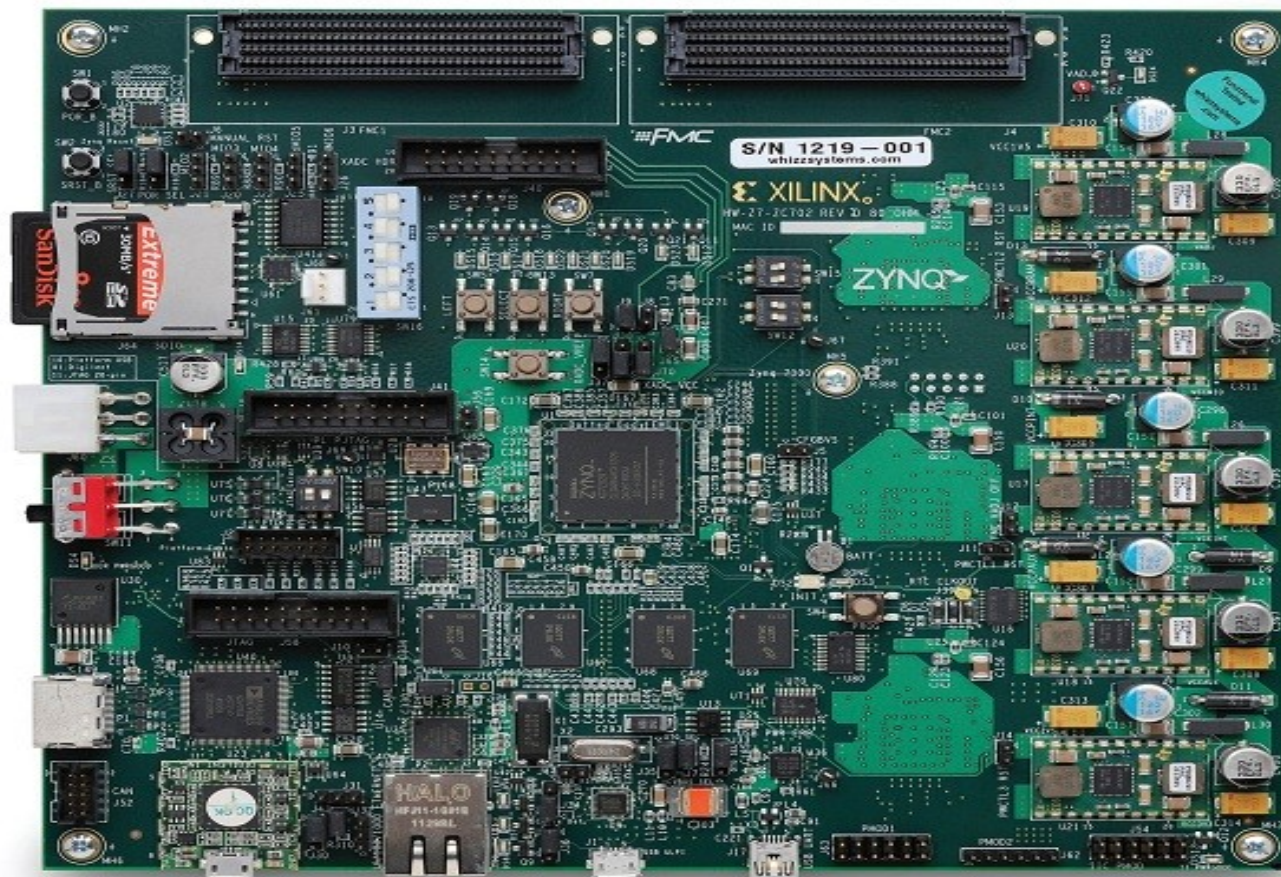




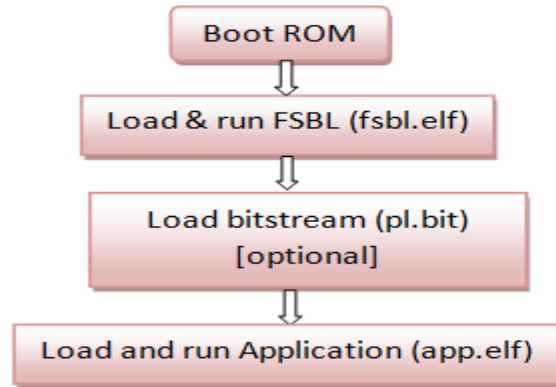
# Raspberry Pi



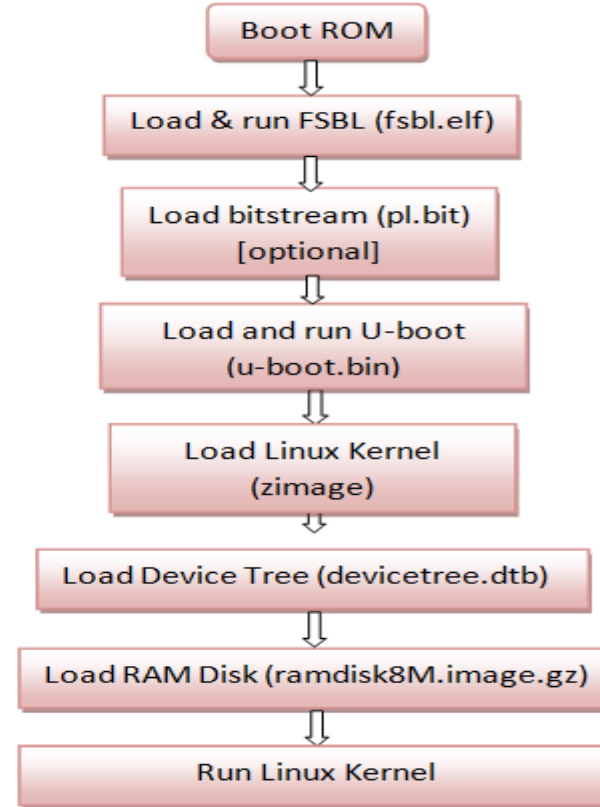




### *Running a Standalone application*

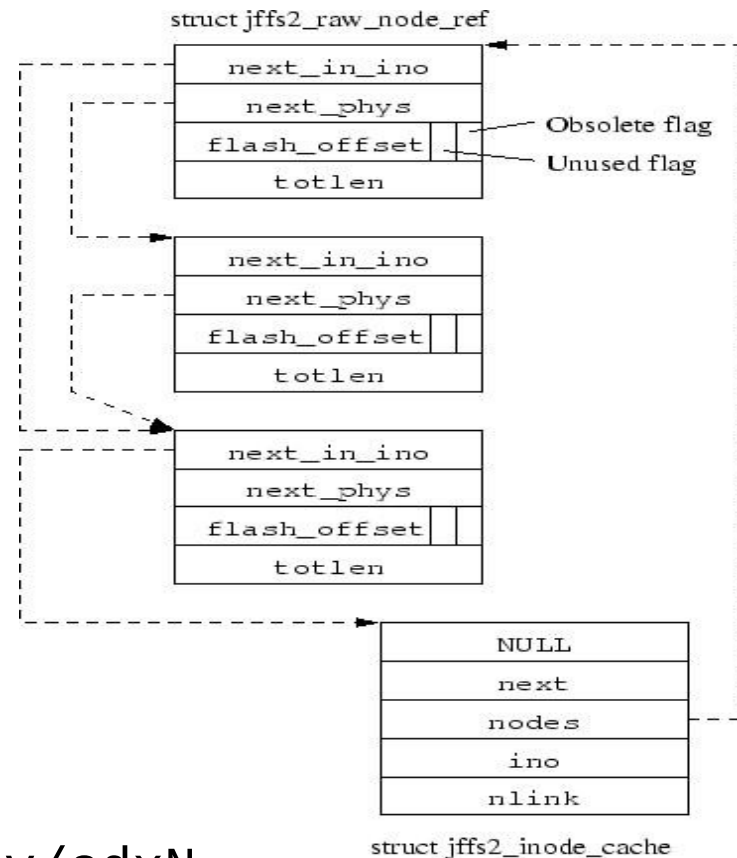


### *Running Linux*



- What we need
  - CPU, memory, storage
  - Display, Touch layer, keyboard
- PC vs. Embedded system
  - BIOS -> primary master (selection/default) -> MBR (grub stage 1) -> VBR (grub stage 2) -> kernel, initramfs, systemd
  - Bootloader (hardcoded address in NAND/NOR), kernel, initramfs, systemd (SYSV init, upstart)
- Bootstrapping
  - For example: We need ext4 driver stored on ext4 driver.
- Bootloader
  - Usually rather simple and written in assembler.
  - u-boot (but also blob, bootldr, redboot, able)
  - What about multiboot?
  - How to control it? Keyboard (+,-,home) vs. touch screen

- NAND/NOR
- FAT
  - Common for cameras,
  - Beware of patents
  - Doesn't support symlinks and posix permissions.
  - Size Limits
- ext2/3/4, reiserFS, XFS
  - Usual for desktops
- JFFS2, YAFFS, UBIFS
  - Journalling Flash File System

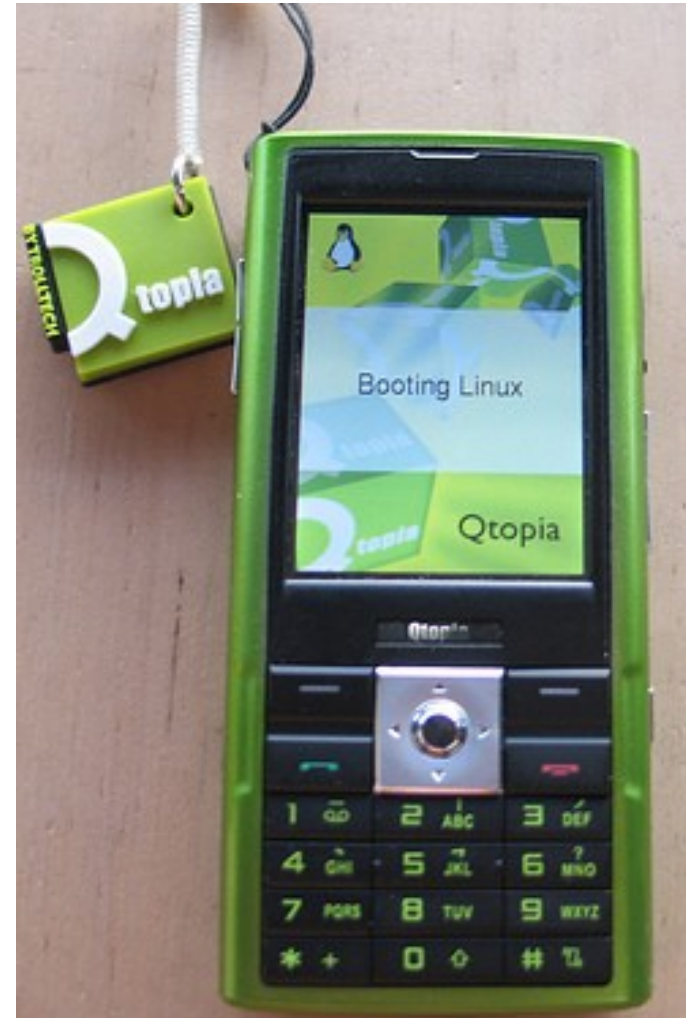
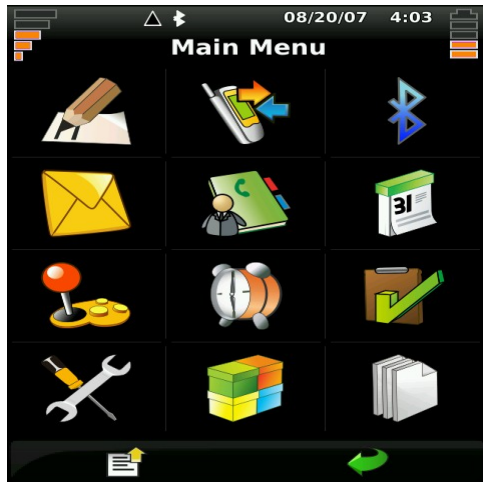


```
tune2fs -o journal_data_writeback /dev/sdxN
```

```
defaults,noatime,data=writeback,barrier=0,nobh,errors=remount-ro
```

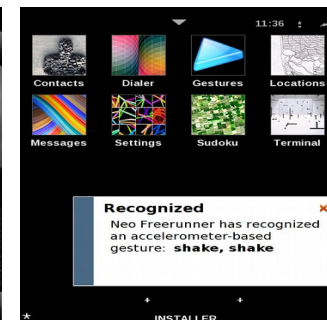
# Greenphone

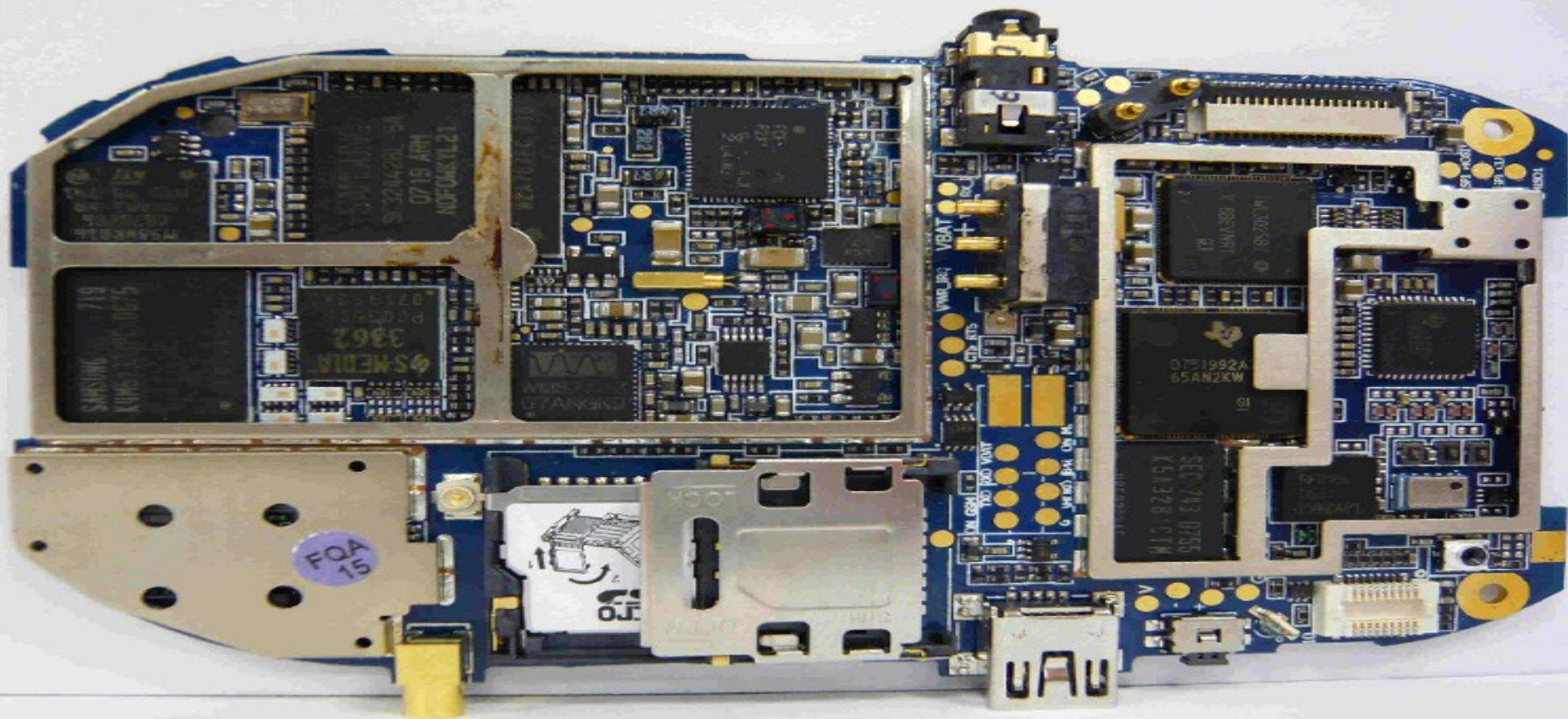
- 2006/9
- Development device
- 300MHz
- 64MB RAM
- 240x320
- Linux, Qtopia (mobile Qt)





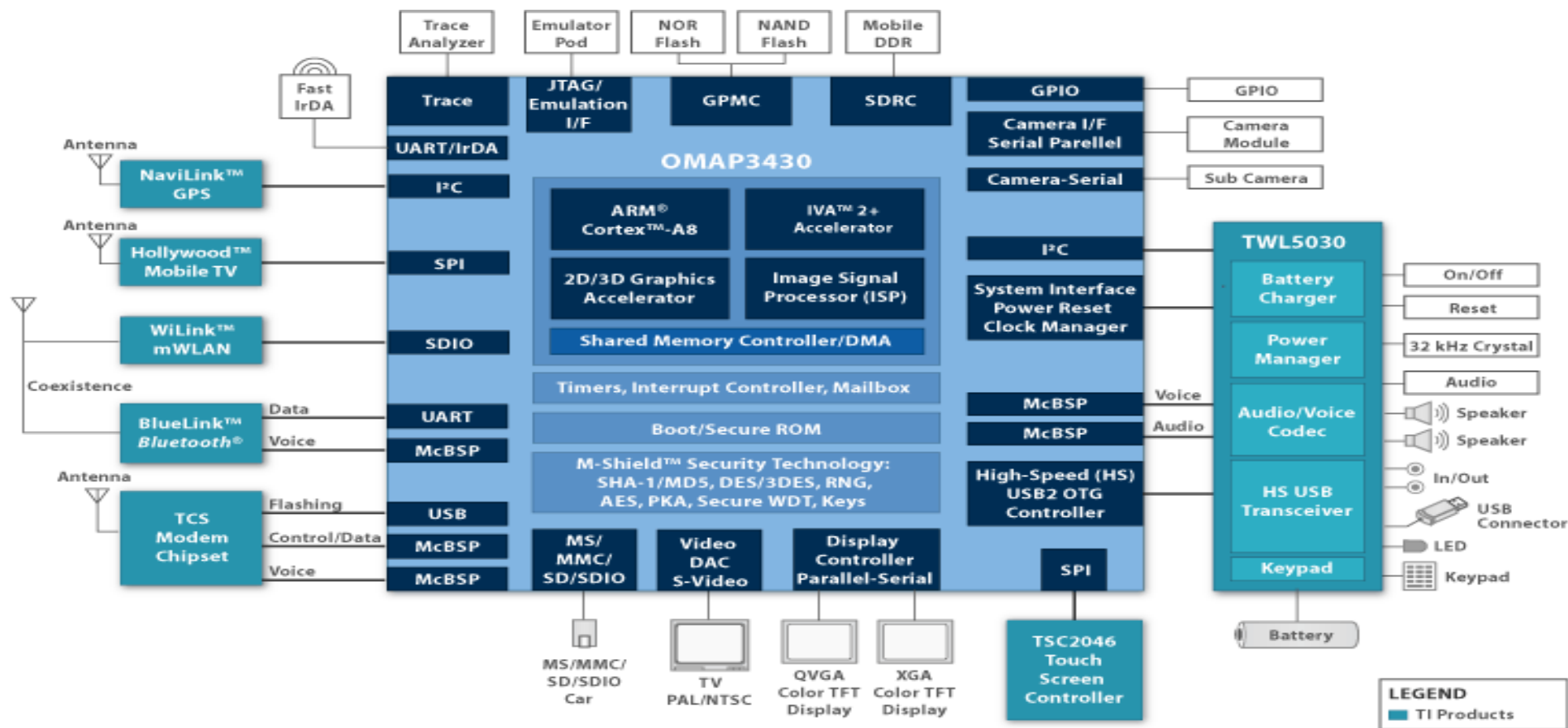
- **2008/6 GTA02**  
(2012/2 GTA04, open phoenix)
- Development device
- 400 MHz (800 MHz)
- SDRAM 128MB (512MB)
- 480x640
- Linux (10+ distributions)
  - i Android (2<sup>nd</sup> device with android word wide, 1<sup>st</sup> in CZE)
  - Qt, Gtk, C, C++, Python, ...
- Have also open source hardware
- ~10 000 units sold
- “phone without phone calls”







# Block diagram



- 2006-2008
- Slow CPU, few memory
- Direct access to hardware
  - Accelerometers, GPS (/dev/tty) – udev, “gpsd”
  - LED (gpio) – udev, sysfs, procfs, ..
  - Network connection (gsm, wifi, bluetooth, ir, nfc, usb, lora) – ~~ifconfig (depraced 2007)~~, ip (iproute2 tools), iptables, network manager, ...
  - Audio (bluetooth, loud audio, headphones, fm receiver/transmitter) – alsa, oss, phonon
  - Software keyboard
- Daemons, service providers
  - Systemd (+cgroups, ulimit)
  - ofono
  - gpsd
  - ConnMan or network manager
  - MCE (mode-control)

```
// active pooling example
#include <linux/input.h>

int main (int argc, char** argv) {
    FILE *eventfp = NULL;
    eventfp = fopen("/dev/input/event3", "rb");
    if (eventfp == NULL) {
        return 1;
    }
    struct input_event buffer[BUFSIZE];

    if (fread(buffer, sizeof (struct input_event), BUFSIZE, eventfp) != BUFSIZE) {
        fprintf(stderr, "fread() failed\n");
        return 2;
    }

    for (int i = 0; i < BUFSIZE; i++) {
        printf("%d %d %d\n",
            buffer[i].type,    // absolute or relative
            buffer[i].code,    // axis X=0/Y=1/Z=2
            buffer[i].value); // rough 0-5g, fine 0-1g
    }
    fclose(eventfp);
    return 0;
}
```

```
struct input_event {
    struct timeval time;
    __u16 type;
    __u16 code;
    __s32 value;
};
```

- LED

```
echo 1 > /sys/devices/platform/gta02-led.0/leds\:gta02-aux\:red/brightness
```

```
cat /sys/devices/platform/i2c_omap.2/i2c-2/2-0032/leds/lp5523\:channel1/brightness
```

```
#!/bin/sh
```

```
echo 0 > /sys/class/gpio/export
```

```
echo out > /sys/class/gpio/gpio0/direction
```

```
while [ true ]; do
```

```
    echo 1 > /sys/class/gpio/gpio0/value
```

```
    sleep 1
```

```
    echo 0 > /sys/class/gpio/gpio0/value
```

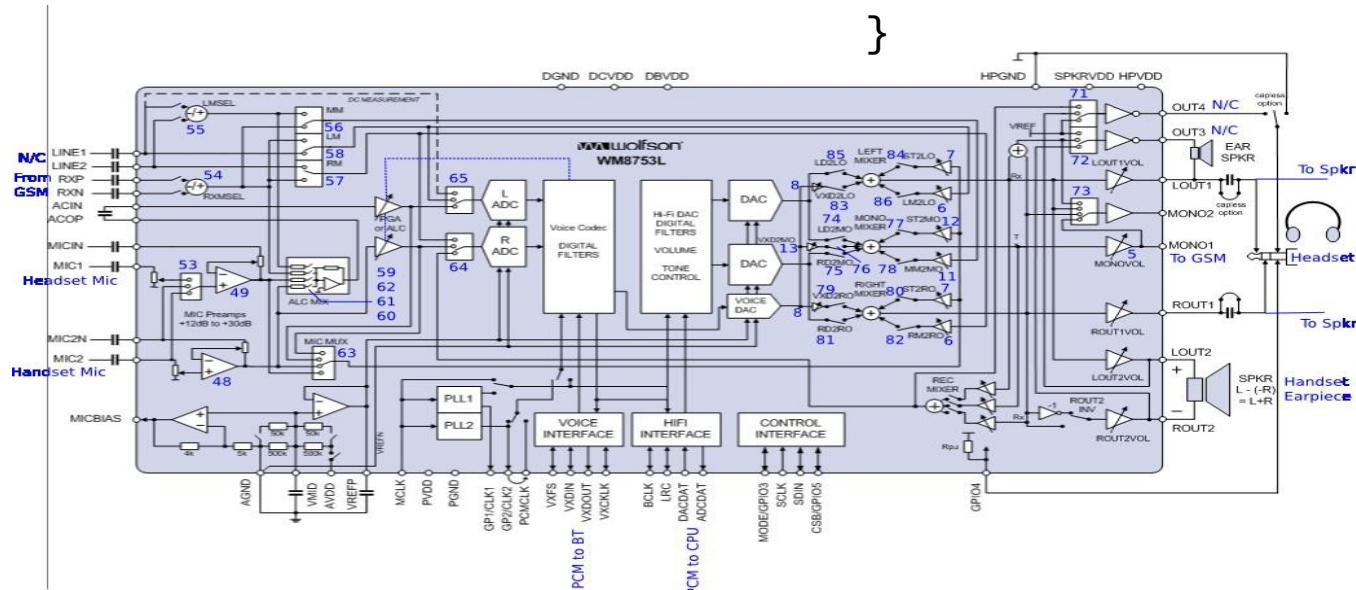
```
    sleep 1
```

```
done
```

```
echo 0 > /sys/class/gpio/unexport
```

- alsamixer
- alsactl
- alsastate

```
control.49 {  
    comment.access 'read write'  
    comment.type INTEGER  
    comment.count 1  
    comment.range '0 - 3'  
    iface MIXER  
    name 'Mic1 Capture Volume'  
    value 0  
}
```



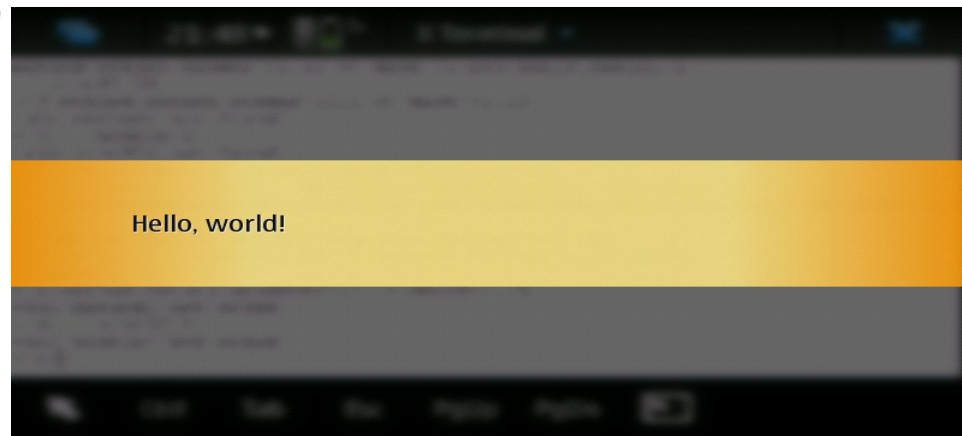
WM8753 Codec – Neo Freerunner

- 770 -> **N900**
- 2005/11 -> **2009/11**
- tablet -> phone for geeks
- 250MHz -> 600 MHz
- 64MB -> 256 MB DDR
- 800x480
- Maemo Linux
  - Based on debian
  - libhildon (gtk), qt
  - C, C++, Python, Bash
- Community repositories (+Nokia Store)



- Inter-process communication library
  - Every daemon publishing own API
  - D-Feet (dbus viewer with UI)
  - Can provide “asynchronous access to gps” (gpsd)

```
$ dbus-send --print-reply \  
  --type=method_call --dest=org.freedesktop.Notifications \  
  /org/freedesktop/Notifications \  
  org.freedesktop.Notifications.SystemNoteDialog \  
  string:'Hello, world!' uint32:0 string:'  
  
method return sender=:1.1 -> dest=:1.15  
uint32 4
```





- Rendering
  - Framebuffer
  - X-Server
  - Wayland
- Libraries
  - (desktop: gtk, qt, efl, fltk, motif, wxwidgets, ..)
  - “skin” (Just decorated standard components. Size of buttons, etc.)
  - libhildon – a little modified gtk (gtk wasn’t enough)
  - Qt Controls (part of Qt)
- What changes are necessary? And Why?
  - Sizes of buttons, layout of components.
  - Touch screen (single touch is click)
  - Multi touch (press and hold gesture, circling)

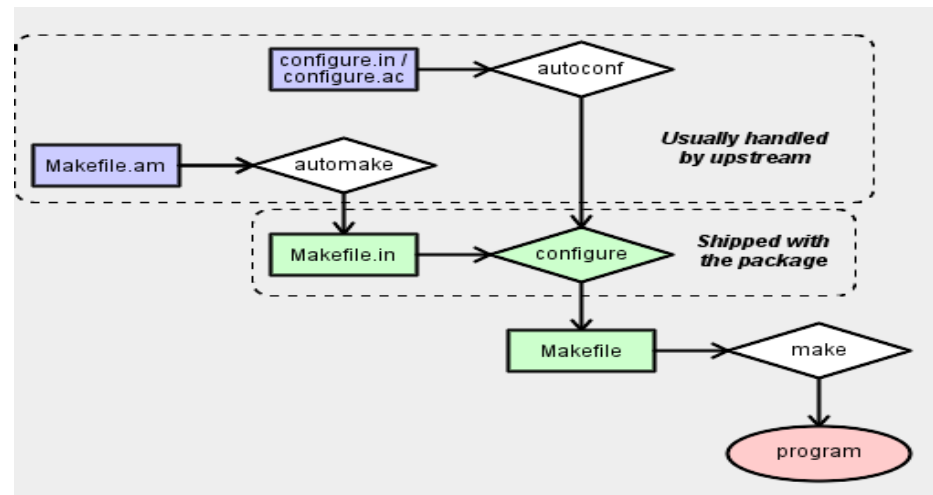
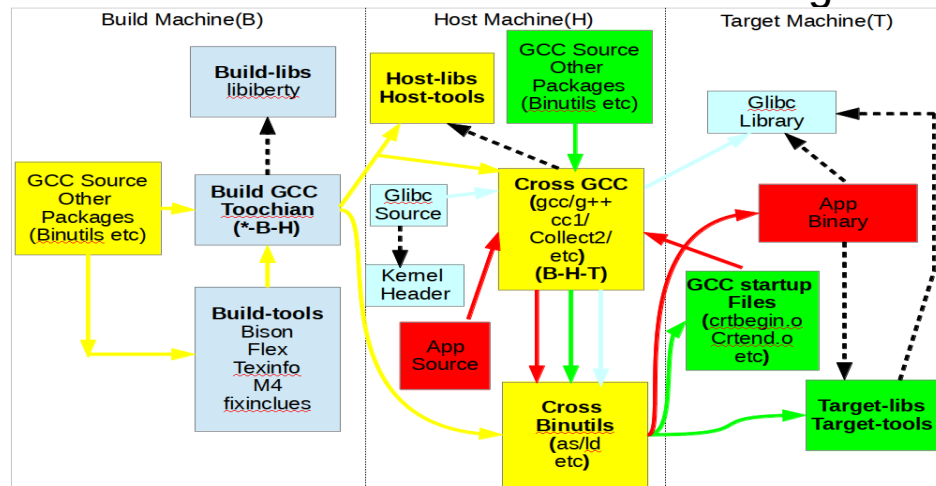
## Nokia N9, Nokia N950

- 2011/9
- Smart phone for normal users
- 1GHz, 1GB RAM, 854x480
- Linux - MeeGo Harmattan
  - deb
  - Qt, QtQuick, C++, QML
- Nokia Store

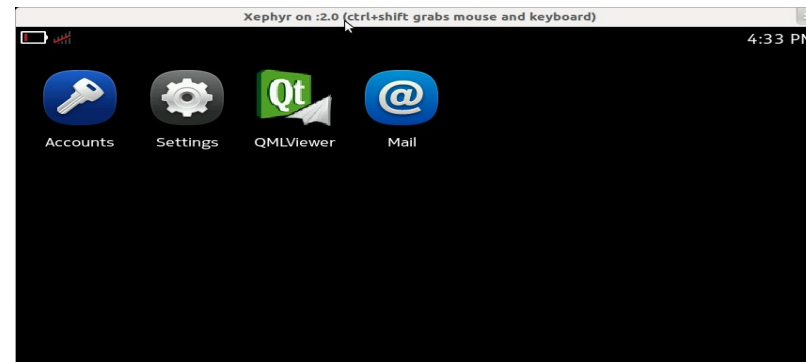


## ■ Toolchain

- Compiler – arm5/arm7/x86, sizeof(int), big/little endian  
export CC=/usr/local/bin/arm-linux-gnueabi-gcc
- Libraries, dependencies  
export CFLAGS=-I /usr/local/arm\_root/include
- Helper tools (e.g. pkg-config)
- qmake myproject.pro -r -spec linux-g++-maemo
- autoreconf && ./configure && make

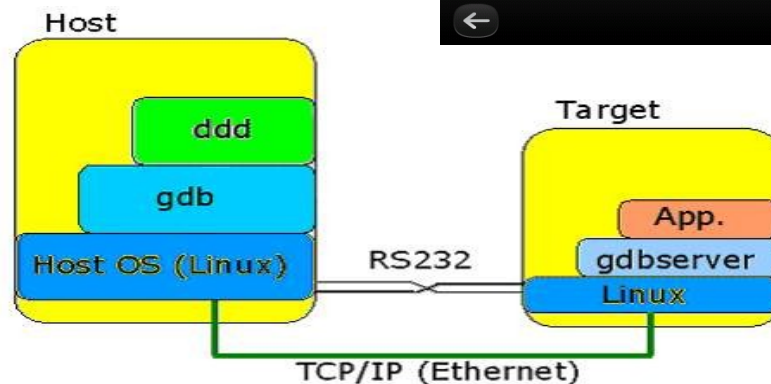
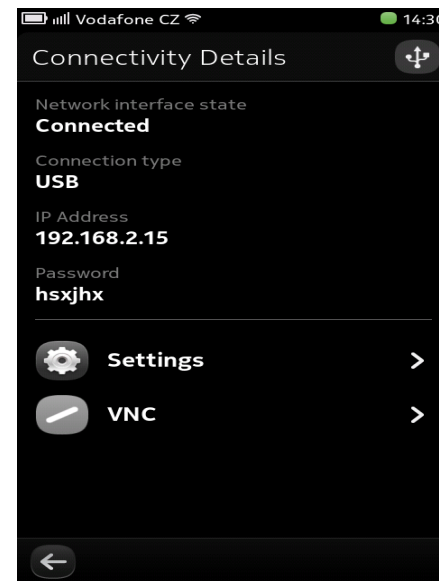


- Chroot + emulated (if required) environment
- “apt-get install” is working inside of chroot
- Switching between toolchains
  - Emulated (arm+QEMU)
  - native (x86)
- UI inside of Xephyr (windowed X Server)



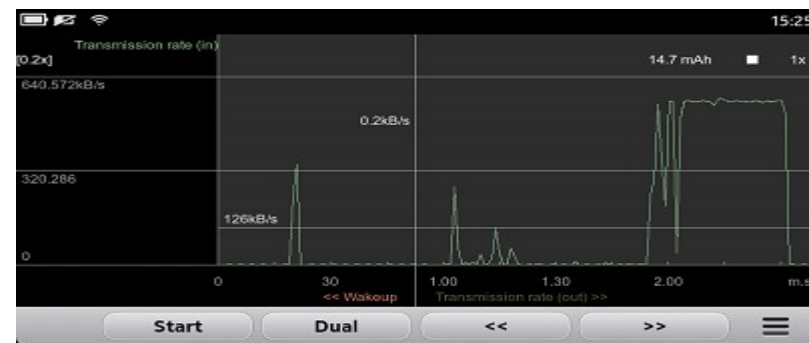
```
$ Xephyr :2 -host-cursor -screen 854x480x16 -dpi 96 -ac +extension Composite &  
$ scratchbox  
[sbox-HARMATTAN_X86: ~] > meego-sb-session start
```

- Workflow
  - Prepare makefile – qmake
  - Compile – make
  - Create package – dh\_make, etc.
  - Install package to sysroot
    - ssh/scp (generic device), madde
  - Debug – gdb, gdbserver, valgrind
- ssh developer@192.168.2.15



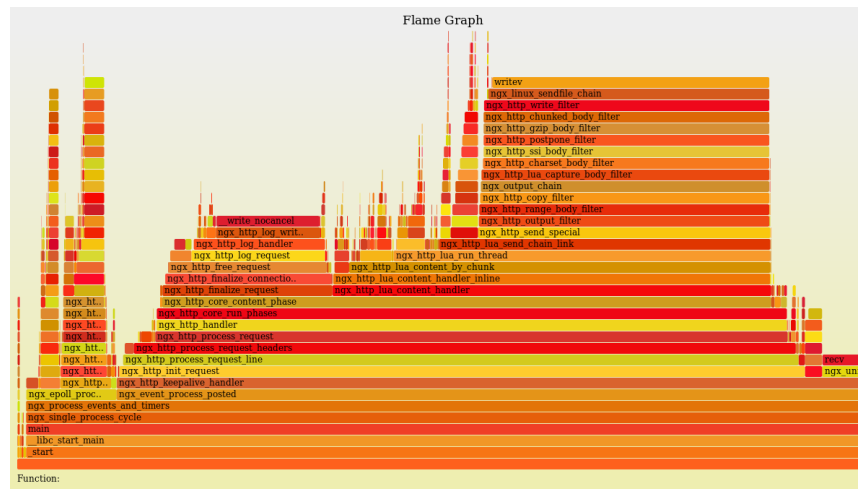
## Profiling, Energy-profiler, powertop

- gprof
- Usually 1 day between charging.
- Harmful application can reduce it to 3 hours
- Monitor
  - CPU load
  - Memory consumption
  - Network traffic (change of IM timeout added +1h)
  - Number of waking up and suspends (C-states)
  - Frequency of CPU
  - Interrupts (IRQs) (or NAPI)



*Pro-tip: try to use “heartbeat” or QSystemAlignedTimer (now obsolete) to spare some CPU ticks.*

*Android AlarmManager.setInexactRepeating(..)*



- \$ perf top # and flame charts
- Observability of system components
  - \$ lsof
  - \$ netstat
  - \$ iostat # iotop
  - \$ sar
  - systemtap
- <http://www.brendangregg.com/linuxperf.html>





### mer-project

- rpm
- Obs (open build service)

### Jolla Sailfish

- 2012/12
- Linux
- harbour
- Continues in tradition of MeeGo  
Harmattan, mer project, nemo mobile

- 2013/Q1
- Tizen SDK 2.0
- Merged with Bada OS
- Linux
  - HTML5, EFL
  - Gtk, Qt is officially not supported



## Blackberry Playbook

- 2011/4
- 1 GHz, 1GB RAM, 1024x600
- QNX
  - C, C++, Java, HTML5, Adobe AIR
  - Qt

## Blackberry 10

- 2013/1
- QNX
  - Realtime unix
  - C, C++, Java, HTML5, Adobe AIR, Cascades SDK (Qt)
- App World



### Blackberry Native SDK

```
$ . ~/bbndk-2.1.0/bbndk-env.sh
```

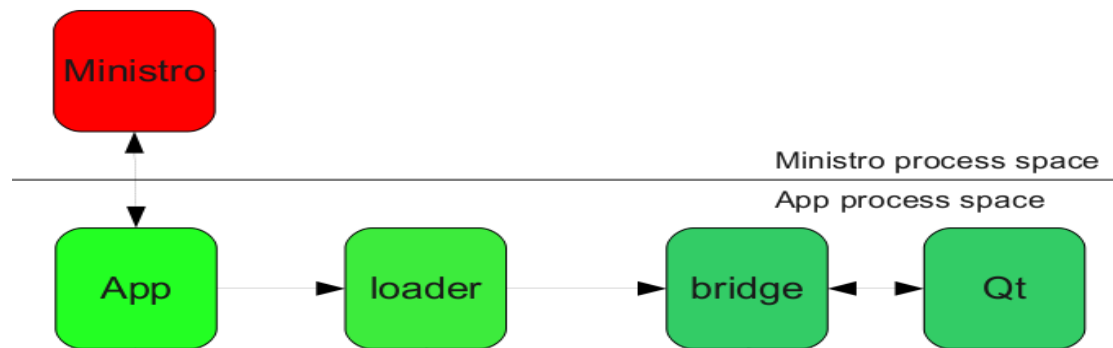
```
$ blackberry-connect -targetHost 169.254.0.1 -devicePassword x
```

```
$ ssh -i ~/.rim/bbt_id_rsa devuser@169.254.0.1
```

### \*.bar packages

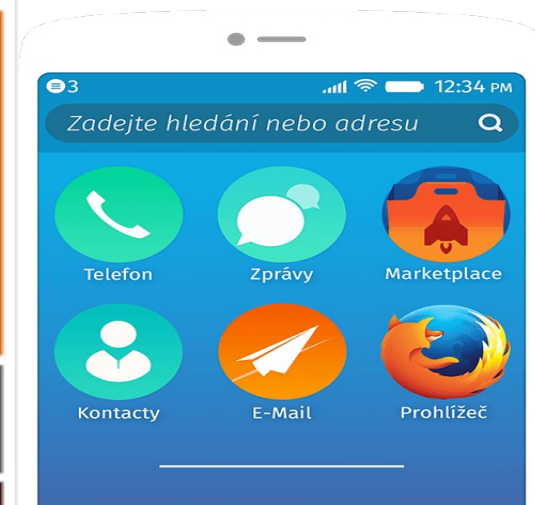
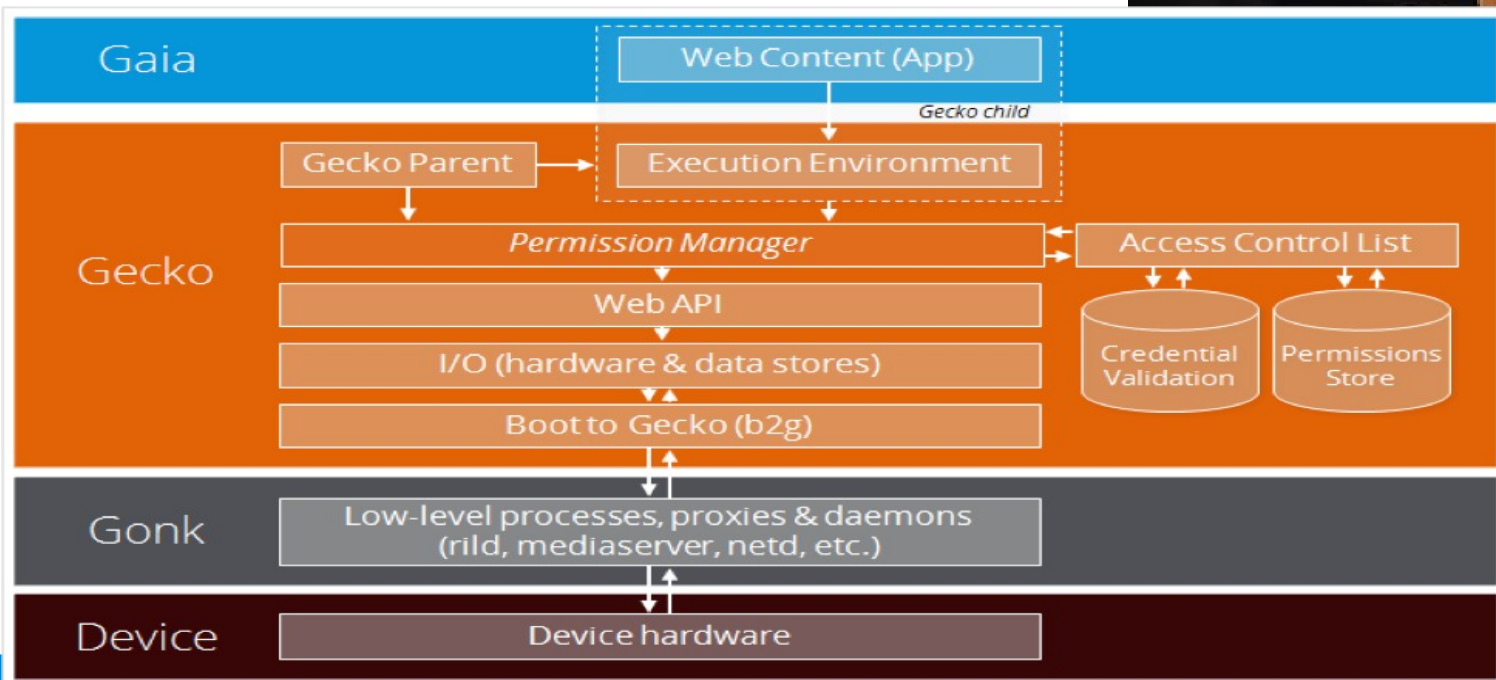
- Zip
- META-INF/MANIFEST.MF
- bar-descriptor.xml
  - (not \*.desktop, see freedesktop.org)
  - Environment variables
  - Name, description, icon

- 2008/9 – 2017 ...
- ~ 19 000+ different devices
- Linux
  - No vanilla kernel -);
  - C++ with NDK, but ...
  - Dalvik/ART, something only via JN
  - Bionic vs. Libc
  - IPC via binder, ashmem
  - Qt
  - Python for Android, etc.



# FirefoxOS

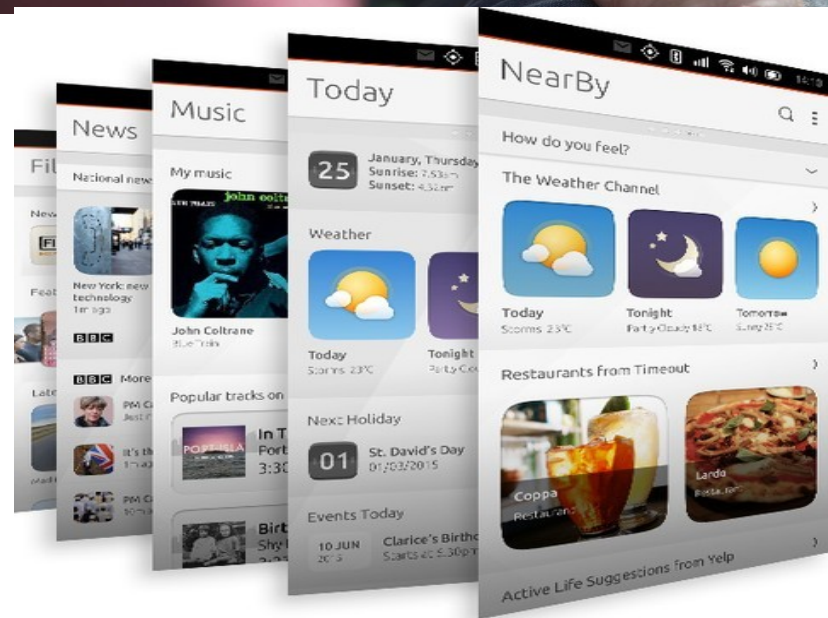
- Android Kernel
- Only HTML5 applications
- >15 devices
  - Mostly low cost (alcatel one touch fire ~ 3 000 CZK)





# Ubuntu Phone

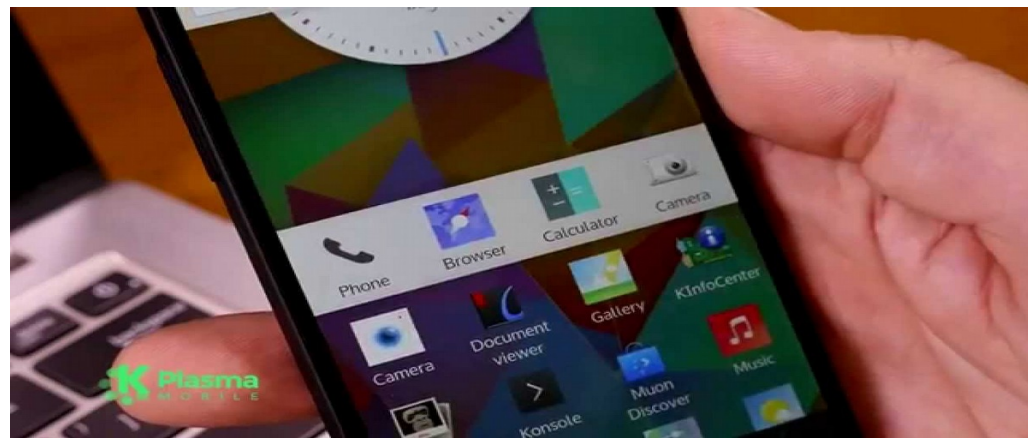
- Dead / Community driven
- Android kernel
- Mostly in QML
- Few application
- Supported devices
  - BQ Aquaris
  - Meizu MX4
  - <https://docs.ubuntu.com/phone/en/devices/devices>



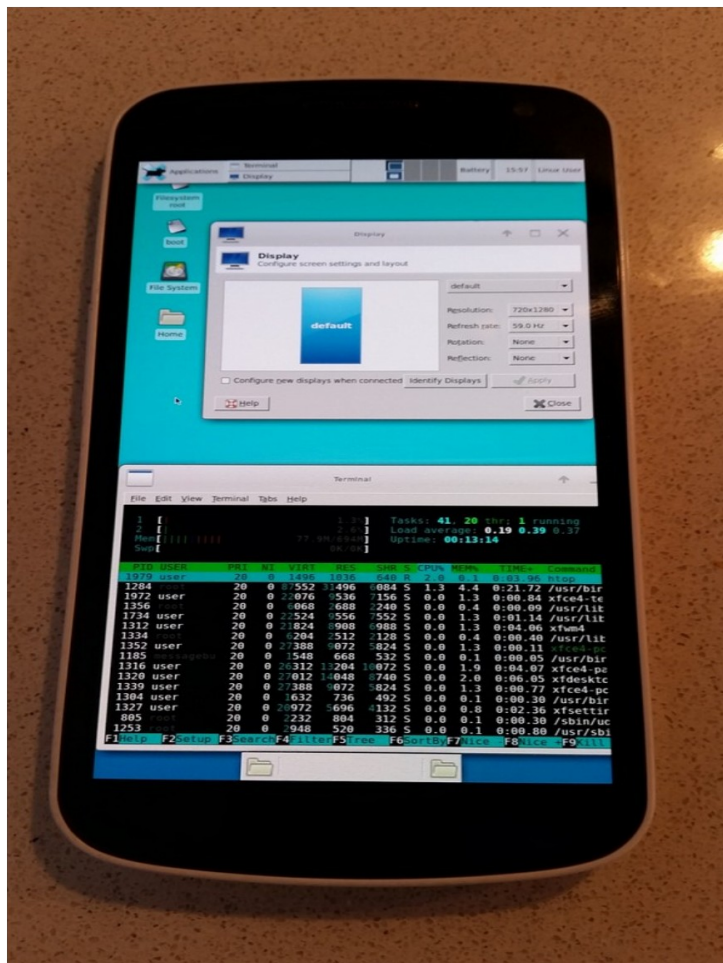


## Plasma mobile

- Based on mer
- prototype
- <https://www.youtube.com/watch?v=auuQA0Q8qpM>
- Used technologies
  - libHybris
  - Direct Rendering Manager
  - Voicecall
  - Ofono
  - OHM
  - Telepathy
  - Pulseaudio



- Purism Librem 5  
<https://puri.sm/shop/librem-5/>



- Based on Alpine Linux
  - sandboxing
- <https://postmarketos.org/>
- /etc/deviceinfo
- Cross-compile all armhf-packages
- Effective caching out of the box (ccache)
- Flasher abstraction
  - Fastboot
  - heimdall
- Security
- UI (Plasma, Hildon aka. N900, LuneOS ui XFCE4, Libweston minimal ui)

## Some random bits (-;

- Packaging
  - Deb, Rpm, Bar, Jar
  - Open Build Service
  - Personal Package Archive
- DSP
  - <http://sourceforge.net/projects/dspgateway/>
  - [http://elinux.org/images/9/9f/DSPGateway\\_Celf\\_Jamboree6.ppt](http://elinux.org/images/9/9f/DSPGateway_Celf_Jamboree6.ppt)
- Realtime Linux
  - RTLinux, RTAI (Application Interface)
  - <https://rt.wiki.kernel.org/>
- User Interface Experience
  - UI matters! Application should look native.
  - <http://harmattan-dev.nokia.com/docs/ux/>
  - <https://developer.blackberry.com/design/>
- Security
  - sandboxing (messaging via d-bus interfaces, ..)
  - aegis (MeeGo Harmattan)
  - SELinux
  - UEFI - Secure boot vs. Vendor lock-in

# Security deserves more attention

- Intel Management Engine and others
- Isn't a non-free baseband firmware a privacy issue?

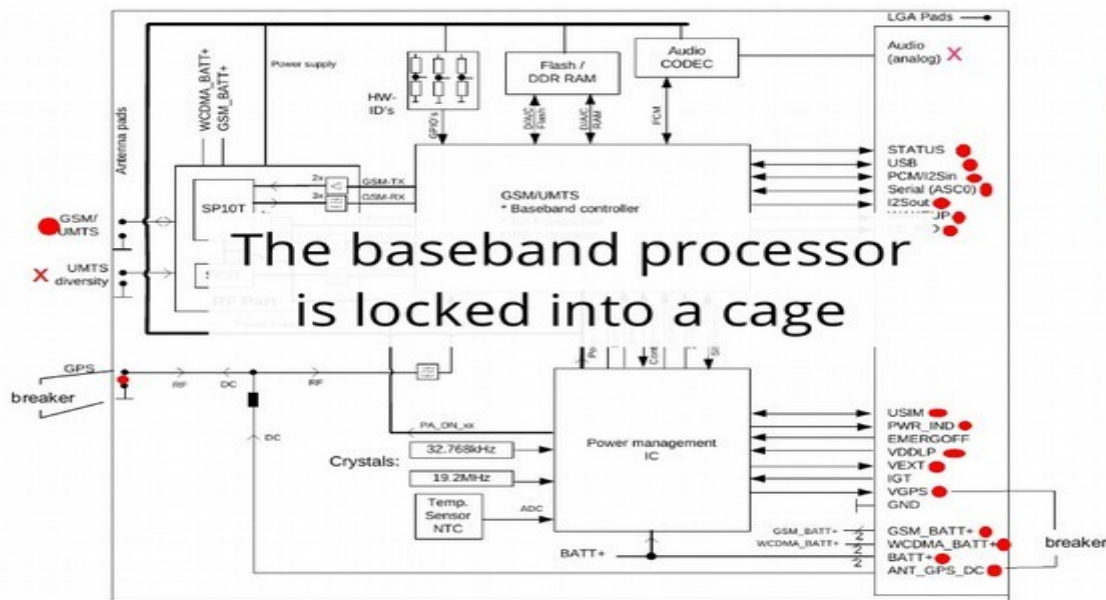


Figure 2: PHS8-P/PHS8-K block diagram

- Baseband is often tightly integrated with rest of the system
  - Direct connection to microphone
  - Direct memory access

## Threats

- Tracking (trilateration, gps-a)
- Eavesdropping
- Data leakage
- Security bugs in firmware, SIM cards
- Direct access to main RAM

<http://neo900.org/faq#privacy>

[http://git.gnumonks.org/laforge-slides/plain/2016/cellular\\_modems\\_33c3/33c3-modems.html](http://git.gnumonks.org/laforge-slides/plain/2016/cellular_modems_33c3/33c3-modems.html)

- Why would you prefer using of common linux based mobile system rather than Android?
- What do you need to compile C/C++ code for linux phone?
- How we can debug application directly on the device?
- How we can access hardware?
- What requirements do have file system for mobile phone?
- How we can reduce power consumption?

- Why would you prefer using of common linux based mobile system rather than Android?
  - ...
- What do you need to compile C/C++ code for linux phone?
  - Toolchain (SDK) – Compiler, Libraries, Chroot, IDE
- How we can debug application directly on the device?
  - scp (madde, dpkg -i), gdb+gdbserver, valgrind, printf
- How we can access hardware?
  - API (Qt), DevFS, SysFS, ProcFS, D-Bus
- What requirements do have file system for mobile phone
  - Life time of NAND, security (stealing), safety (damage), speed, easy of use, patents, compatibility with desktop, ..
- How we can reduce power consumption?
  - Profiling (CPU), Energy profiling, Use sleep and avoid pooling, reduce frequency of accessing to hardware (gps, networking), Use bursts processing if possible.



## References

- <http://elinux.org/>
- Karim Yaghmour: Building Embedded Linux Systems, 1st ed., O'Reilly Media, 2003.
- Zucker, Daniel, and Ray Rischpater. Beginning Nokia Apps Development: Qt and HTML5 for Symbian and MeeGo. Apress, 2010.
- Beekmans, Gerard. Linux from Scratch. Iuniverse Inc, 2000.<http://www.linuxfromscratch.org/>
- [https://www.developer.nokia.com/Community/Wiki/Code\\_snippets\\_table\\_for\\_common\\_use\\_cases](https://www.developer.nokia.com/Community/Wiki/Code_snippets_table_for_common_use_cases)
- Fitzek, Frank H. P., Tommi Mikkonen, and Tony Torp. Qt for Symbian. John Wiley and Sons, 2010.
- Hoque, M.A., M. Siekkinen, and J.K. Nurminen. “On the Energy Efficiency of Proxy-based Traffic Shaping for Mobile Audio Streaming.” In 2011 IEEE Consumer Communications and Networking Conference (CCNC), 891 –895, 2011.
- CCC Chaos Communication Congress  
<https://fahrplan.events.ccc.de/congress/2016/Fahrplan/events.html>

- Alternative Androids
  - <https://copperhead.co/>
  - <https://lineageos.org/>
  - <https://www.replicant.us/>
- OSHW Munich
  - [https://www.youtube.com/channel/UCP\\_U91ipVWsF\\_2EiG3s6BKA/videos?shelf\\_id=0&view=0&sort=dd](https://www.youtube.com/channel/UCP_U91ipVWsF_2EiG3s6BKA/videos?shelf_id=0&view=0&sort=dd)
- Chaos Computer Club
  - [https://media.ccc.de/v/28c3-4735-en-reverse\\_engineering\\_a\\_qualcomm\\_baseband](https://media.ccc.de/v/28c3-4735-en-reverse_engineering_a_qualcomm_baseband)