2011 ARM® TechCon™
Join the community defining the future

i'm watch the first Android Smartwatch

Nicola La Gloria, Ph.D.

Field Application Engineer

Agenda

- Introduction
- Hardware Design
- OS firmware design
- Enterprise (quick introduction)
- Q&A

Introduction: what is

”The ultimate way to connect to your mobile device”…

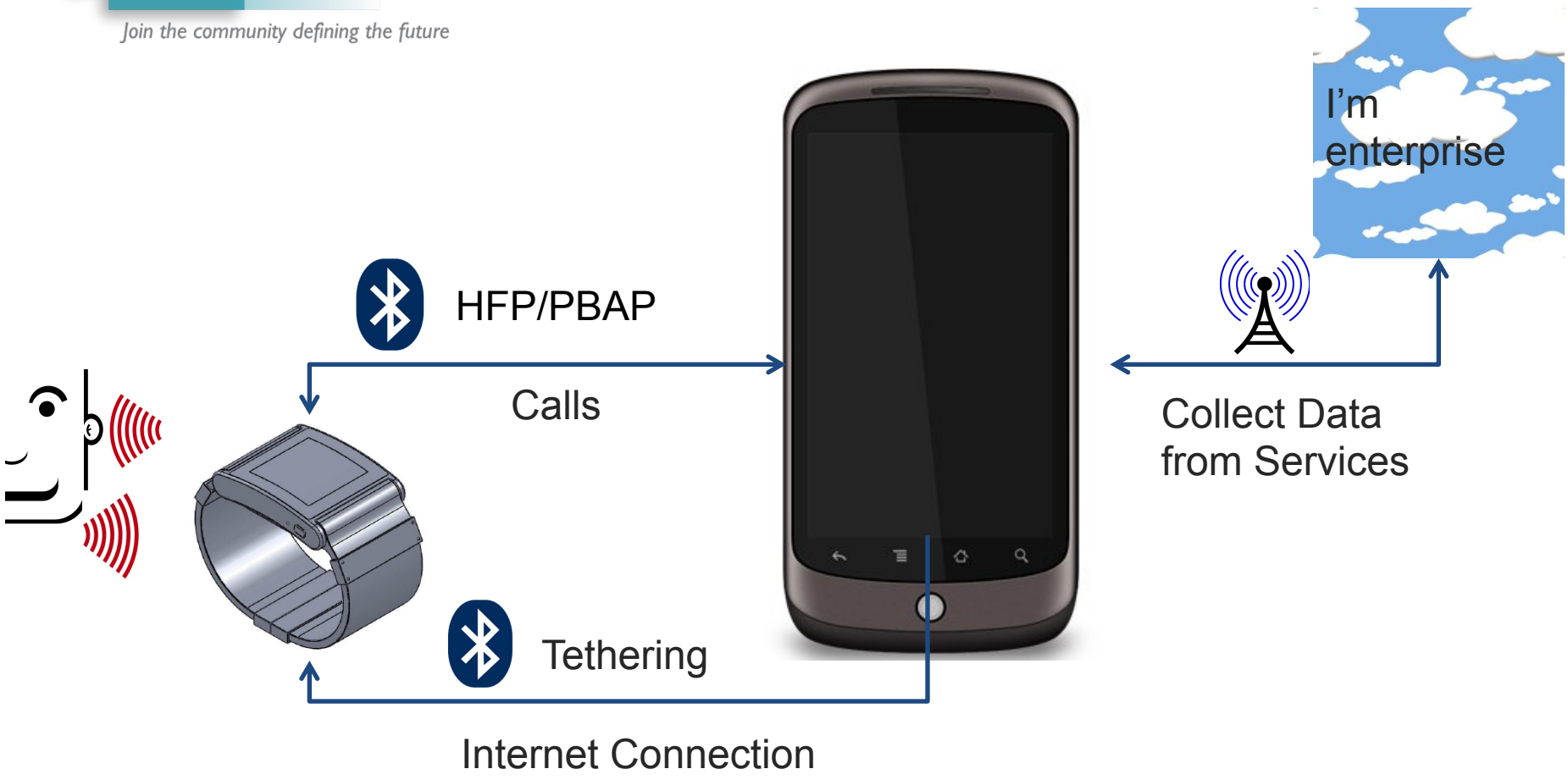
Features:

- **Android** based OS
- Manages services **notifications**
- It’s a Bluetooth **handset** device (HFP, PBAP)
- Features **Apps** as any smartphone
- Plays **multimedia** contents
- **Appealing** design and quality of **manufacture**
- **User** Apps through SDK.

Introduction:
Challenges

- **Low Memory**, 64MB RAM for System and Applications
- **Reduced battery** consumption
- **Reduced** space for components placement
- **Curved** capacitive touch technology
- Bluetooth integration for **handset** features
- **MIPI display** technology integration
- Applications **GUI design** and **Accessibility**
- Bluetooth **tethering** (internet connection)

Introduction:
The Concept

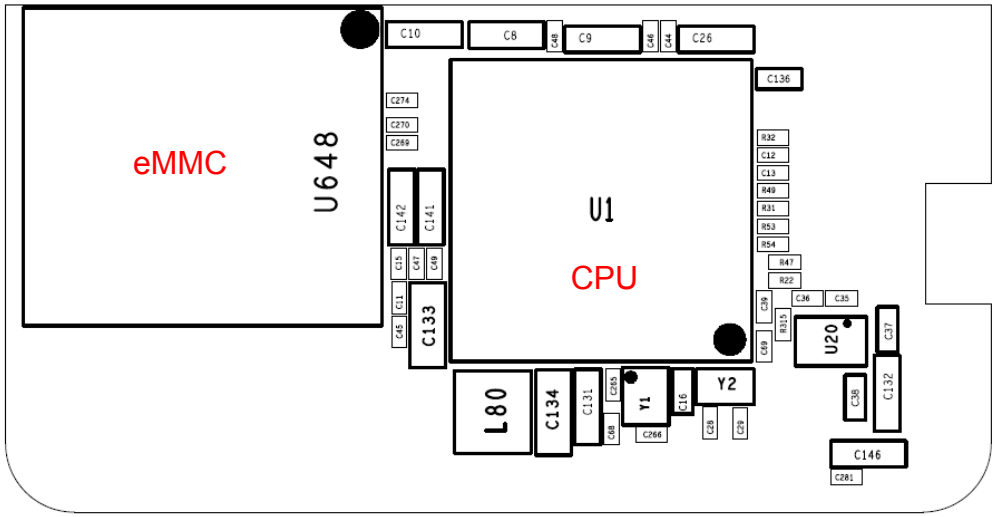


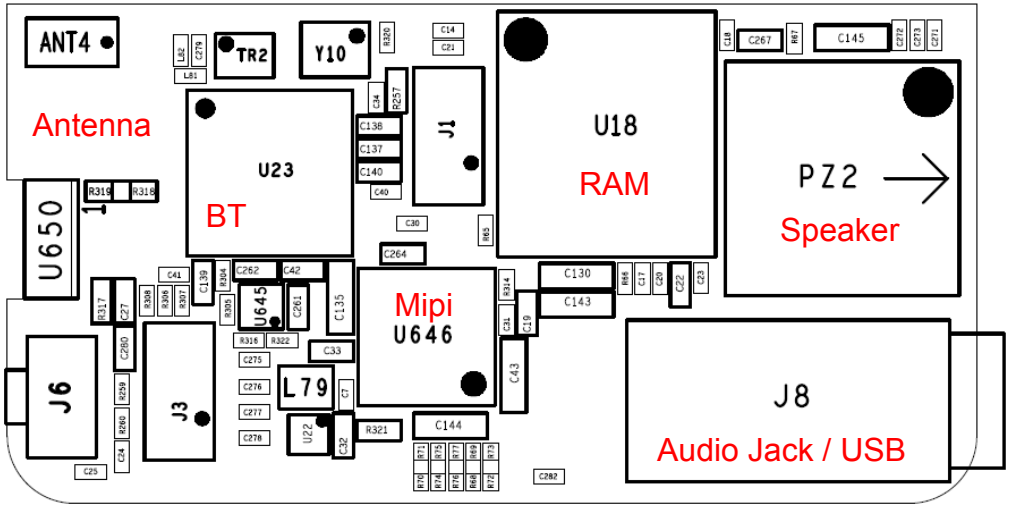
Hardware Design: Hardware Specifications

- i.Mx233 @ 450MHz CPU
- 64MB LPDDR (MT46H32M16LFBF-6L_:C)
- Bluetooth
- Microphone
- Speaker
- 4 GB eMMC
- Jack Stereo Audio Out
- USB OTG
- 450mAh Battery
- 1.54” 240x240 Display (MIPI)
- Curve Capacitive Touch screen
- One stand-by button

Hardware Design:
Components placement

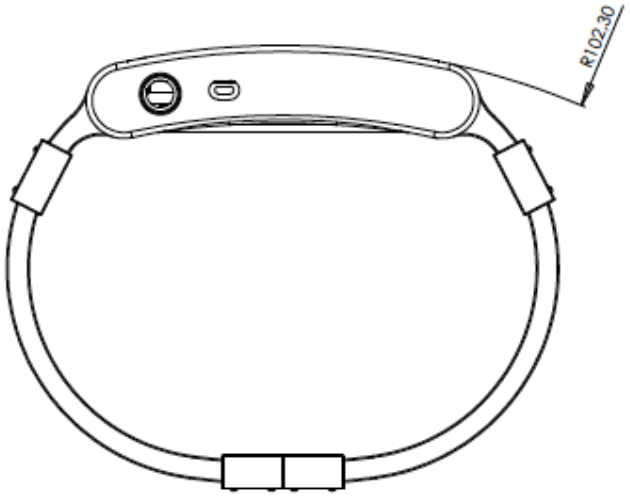
Assembly Bottom

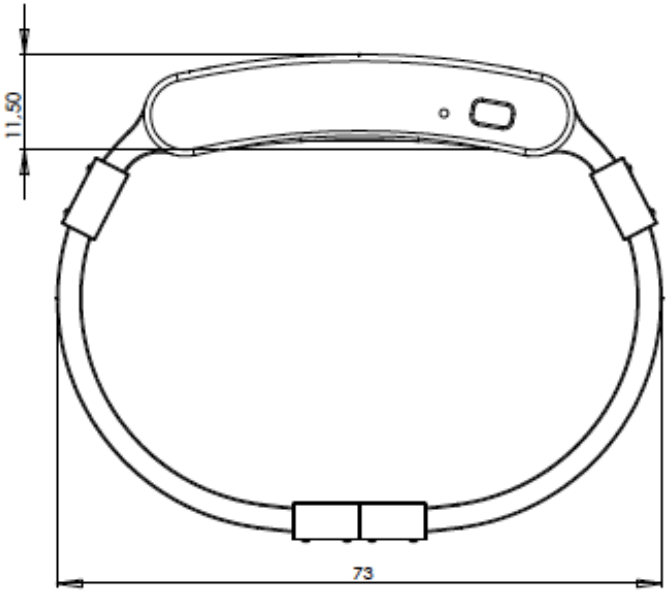


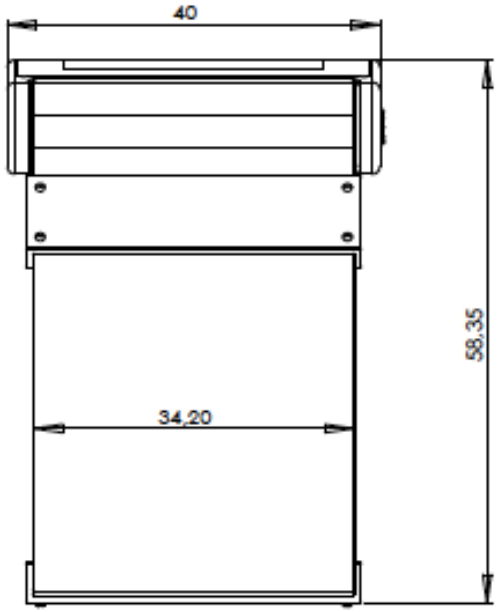


Assembly Top

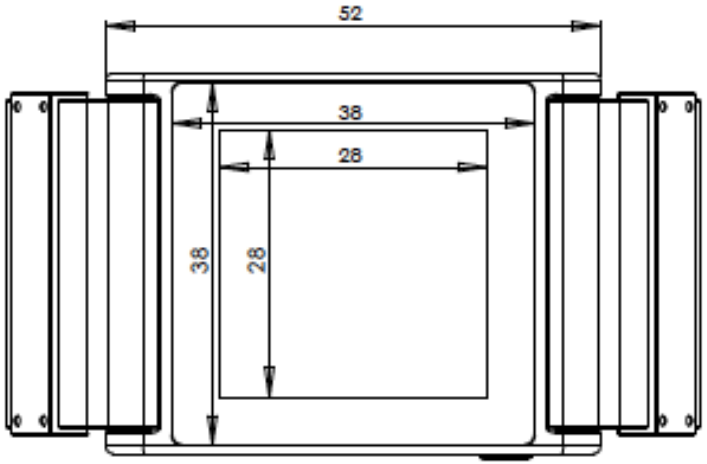
Hardware Design:
Dimensions

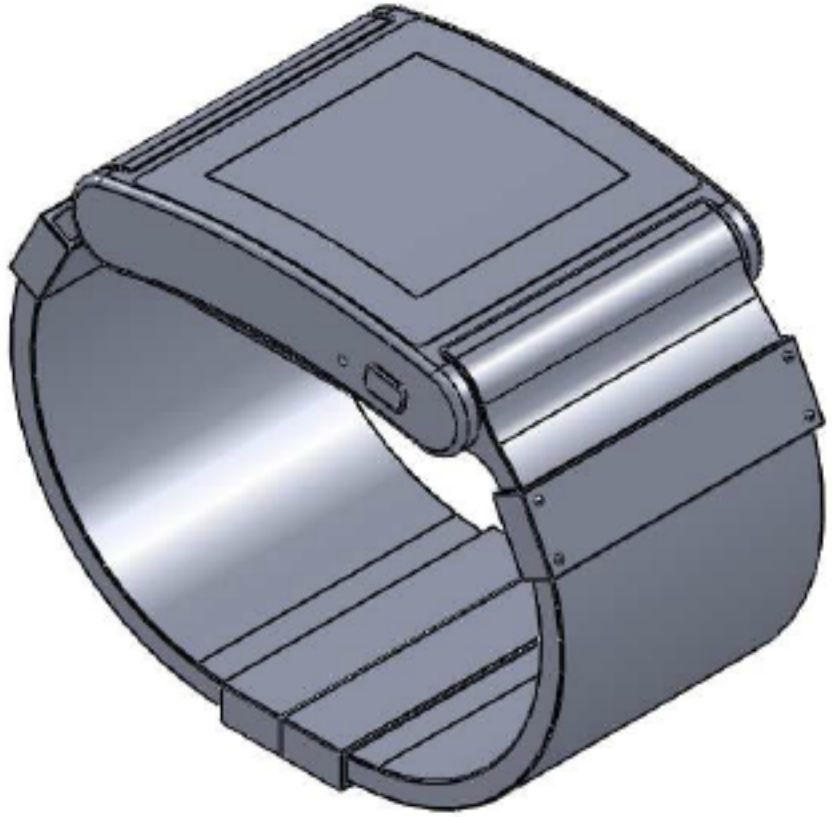




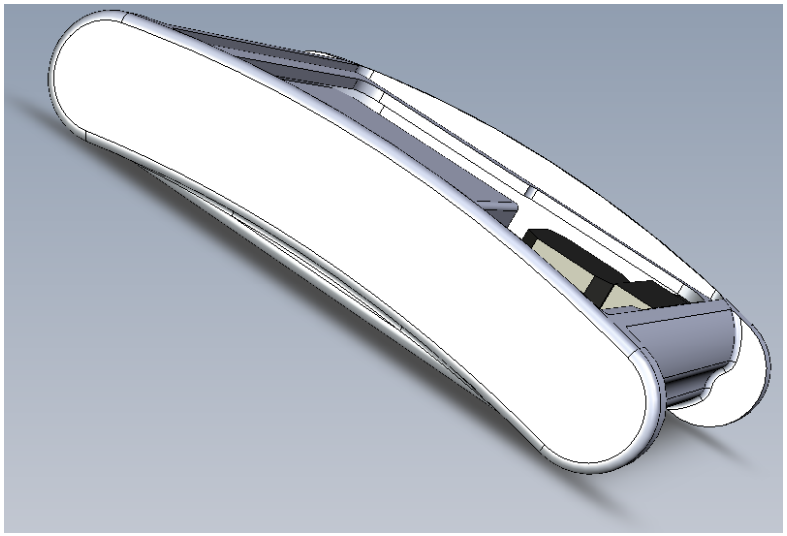


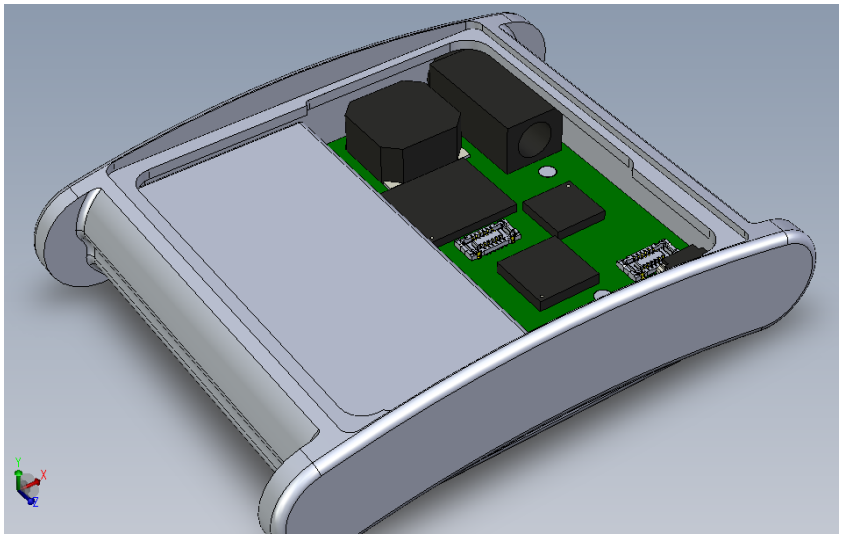
Hardware Design:
Dimensions





Hardware Design:
Mechanical





Rendering

The Operating System

i'm™ DROID

is the i'm watch customized Android OS

- ✓ Donut 1.6 (lower memory requirement)
- ✓ Bug fixes
- ✓ Back-ports from Gingerbread and Froyo
- ✓ Custom native code
- ✓ Expose native methods to APIs (through JNI)

i'm Droid Firmware Design:
the kernel

An official Android Kernel for FSL i.MX233 was not available.

Android manual kernel porting for i.MX28:

- ✓ Official Android Kernel 2.6.35
- ✓ FSL patches for generic Linux i.IMX platform (including i.MX233 patches)
- ✓ Manual conflict resolution

Most efficient:

- ✓ "Git merge" between Android Kernel 2.6.35 and FSL i.MX Kernel 2.6.35
- ✓ Time effective, less conflicts
- ✓ No manual patches
- ✓ Full kernel history and easy update management

i'm Droid Firmware Design:
CPU scaling and standby

Due to strict energy saving policy, the system has to be scalable in terms of CPU speed.

- 262 MHz
- 360 MHz
- 390 MHz
- 454 MHz

To limit battery consumption the user-space Kernel Frequency CPU Governor is used:

- CPU Governor *Conservative*
- CPU Governor *OnDemand*

They scale the CPU frequencies according to the needs. *OnDemand* governor switches to governor increases/decrease frequency immediately, while the *Conservative* step by step.

i'm Droid Firmware Design:
Preliminary Power Consumption Tests

Consumption tests have been done to evaluate the energy saving in the i.MX233 EVK on different set-up:

- Removing/deactivating components (Ethernet, Serial, Memory, USB)
- Varying the CPU Frequency
- Varying the Display Backlight
- Standby/Idle states

As detailed in the specifications of i.MX233 processor, FSL Linux BSP does not support suspend-to-RAM mode. To send properly in low power mode when the screen timeout expires, Android has been forced to call the **standby** mode instead of the suspend-to-RAM.

i'm Droid Firmware Design:
The Touch Interface: TSlib

- ✓ TSlib is an abstraction layer for touchscreen panel events, as well as a filter stack for the manipulation of those events.
- ✓ It was created by **Russell King, of arm.linux.org.uk**
- ✓ TSlib is generally used on embedded devices to provide a common **user-space** interface to touchscreen functionality

To calibrate the touchscreen the TSlib calibration suite has been integrated into Android. They include:

- Porting TSlib for Android (binary build);
- Android's framework integration;
- Application for calibration (TSCalibration for testing).

i'm Droid Firmware Design:
Preliminary Consumption Tests

i.MX233 Consumption Test (Battery 3.6 V)

CPU MHz	Backlight	Status	mA
all	0	Standby	25
454	50	Idle	116
454	100	Idle	164
392	100	Idle	157
392	0	Idle	93
360	100	Idle	154
262	100	Idle	150

Removed: 64MB RAM, Ethernet, Serial

i'm Droid Firmware Design:
Alsa drivers

To enable Alsa Driver you have to configure properly the Kernel

```
[*] SPI Sound devices
<*> ALSA for SoC audio support
->  <*>   SoC Audio for the MXS chips
->  <*>   SoC Audio support for MXS-EVK ADC/DAC
->  <*>   MXS ADC/DAC Audio Interface
```

obtaining the following devices

```
/dev/timer
/dev/controlC0
/dev/pcmC0D0p
/dev/pcmC0D0c
```

```
# cat /proc/asound/cards
0 [EVK            ]: mxs adc/dac - MXS EVK
                     MXS EVK (mxs adc/dac)
# cat /proc/asound/devices
 0: [ 0]   : control
16: [ 0- 0]: digital audio playback
24: [ 0- 0]: digital audio capture
33:        : time
```

i'm Droid Firmware Design:
Set up the system for Alsa Integration

- ✓ Change device permission and device linking into Android init.rc

```
# change permissions for alsa nodes
chown root audio /dev/pcmC0D0c
chown root audio /dev/pcmC0D0p
chown root audio /dev/controlC0
chown root audio /dev/timer
```

&

```
chmod 0660 /dev/pcmC0D0c
chmod 0660 /dev/pcmC0D0p
chmod 0660 /dev/controlC0
chmod 0660 /dev/timer
mkdir /dev/snd
symlink /dev/pcmC0D0c /dev/snd/pcmC0D0c
symlink /dev/pcmC0D0p /dev/snd/pcmC0D0p
symlink /dev/controlC0 /dev/snd/controlC0
symlink /dev/timer /dev/snd/timer
```

- ✓ Get *alsa-lib* and *alsa-utils* from Android Git
- ✓ Make build

```
make BUILD_WITH_ALSA_UTILS=true BOARD_USES_ALSA_AUDIO=true
```

- ✓ Copy the libraries and executables

```
/system/lib/libasound.so
/system/bin/alsa_amixer
/system/bin/alsa_aplay
/system/bin/alsa_ctl
```

- ✓ Configure properly `/system/etc/asound.conf`

```
ctl.AndroidOut {
                type hw
                card 0
}
ctl.AndroidIn {
                type hw
                card 0
}
```

i'm Droid Firmware Design:
Alsa Android Integration

- ✓ Get *alsa-sound* (Audioflinger backend) from Android Git
- ✓ To build the system with Alsa support remember to deactivate the GENERIC_AUDIO flag

```
make BUILD_WITH_ALSA_UTILS=true BOARD_USES_ALSA_AUDIO=true BOARD_USES_GENERIC AUDIO=false
```

- ✓ Copy all the libraries and binaries in /system/…

```
• libasound.so
• libaudio.so
• libaudioflinger.so
• libsystem_server.so
• libandroid_servers.so
• hw/alsa.default.so →hw/alsa.freescale.so
• hw/acoustics.default.so → hw/acoustics.freescale.so

• mediaserver
• system_server
```

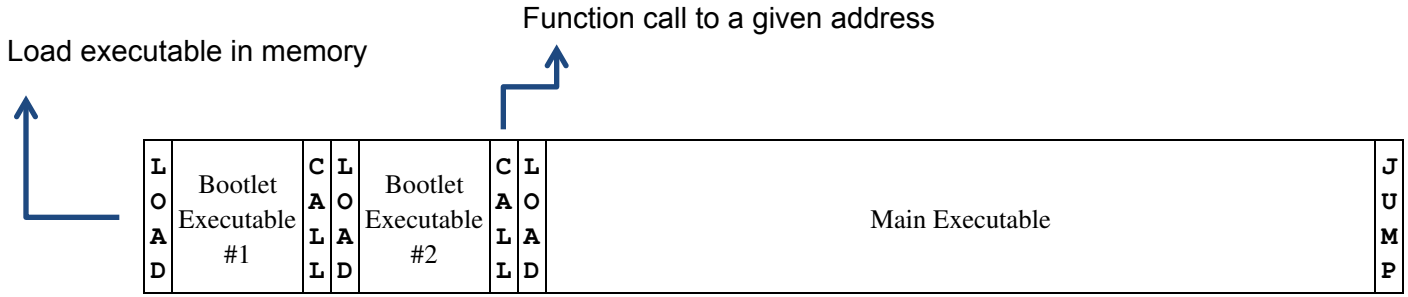
- ✓ Verify in `logcat`

```
D/AudioHardwareInterface: Creating Vendor Specific AudioHardware
```

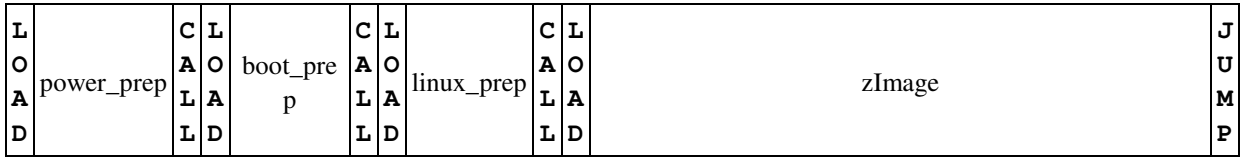
- ✓ Ready to Play!

i'm Droid Firmware Design:
Bootloader

The ROM of i.MX233 reads the boot mode pins to discover the boot source and negotiates the *boot stream*, a stream of byte in SB format.

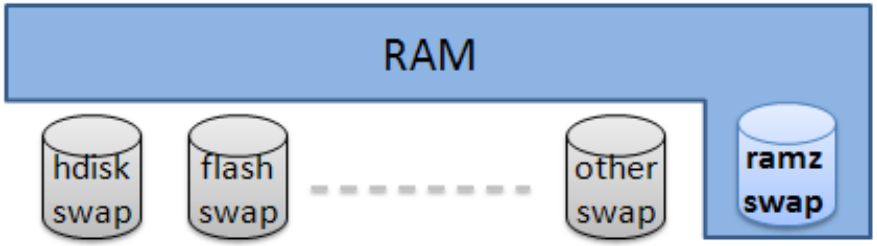


I'm watch features an eMMC on BGA, that is the system non-volatile memory. In particular, it stores the kernel which is bundled in a boot-stream:



i'm Droid Firmware Design:
Memory Optimization

A strategy to increase the amount of memory available is to compress/decompress transparently the data. This type of approach is slower than writing directly to RAM,
(it requires the use of the CPU for comp/decomp), but it's still faster and less power consuming than writing to disk. **CompCache** puts into practice this strategy by making a swap partition that can be mapped to RAM.



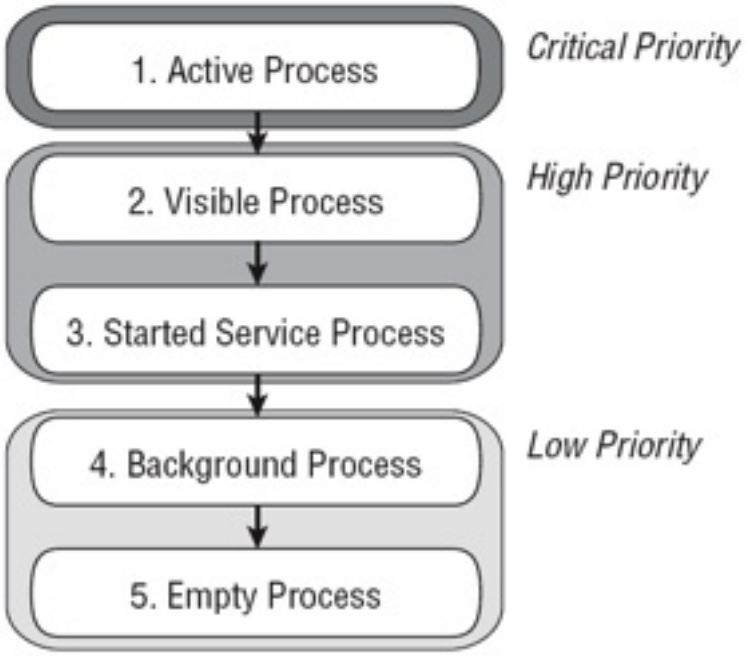

```
ramzswap.ko (virtual block device driver)
rzscontrol  (userspace utility to setup individual ramzswap devices)
```

```
rzscontrol /dev/block/ramzswap0 --init
```

This will initialize (default) the virtual device with a size equal to **25%** of the uncompressed data.
With 64MB, the ***ramz*** device will be initialized to 16MB of uncompressed data.
(in practice one more visible application)

i'm Droid Firmware Design:
Low memory killer

- ✓ Android has an ad-hoc mechanism to select the process to be closed in case of out of memory.
- ✓ The processes are grouped into categories and for each there's a "threshold" expressed in "pages"
- ✓ (1 page = 4KB)
- ✓ When the amount of free memory falls below this threshold, the *lowmemorykiller* module starts to close processes belonging to that category.
- ✓ Parameters tuning is very useful.



i'm Droid Firmware Design:
Low memory killer

```
# Define the oom_adj values for the
# classes of processes that can be
# killed by the kernel.

setprop ro.FOREGROUND_APP_ADJ 0
setprop ro.VISIBLE_APP_ADJ 1
setprop ro.SECONDARY_SERVER_ADJ 2
setprop ro.BACKUP_APP_ADJ 2
setprop ro.HOME_APP_ADJ 2
setprop ro.HIDDEN_APP_MIN_ADJ 7
setprop ro.CONTENT_PROVIDER_ADJ 14
setprop ro.EMPTY_APP_ADJ 15
```

```
# Define the memory thresholds at which the
# above process classes will
# be killed.  These numbers are in pages (4k).

setprop ro.FOREGROUND_APP_MEM 1536
setprop ro.VISIBLE_APP_MEM 2048
setprop ro.SECONDARY_SERVER_MEM 4096
setprop ro.BACKUP_APP_MEM 4096
setprop ro.HOME_APP_MEM 4096
setprop ro.HIDDEN_APP_MEM 5120
setprop ro.CONTENT_PROVIDER_MEM 5632
setprop ro.EMPTY_APP_MEM 6144
```

6 parameters Linux Kernel (low memory killer module), 8 parameters Android (Java)

i'm Droid Firmware Design:
Applications startup and memory

Estimate performance and memory (PSS) consumption [ActivityManager.getMemoryInfo()]

- RAM, 64 MB and 128 MB
- Screen resolution 240x240 and 640x480, 160 dpi

	640x480		240x240		640x480 vs 240x240	
App	Start up (s)	Mem (KB)	Start up (s)	Mem (KB)	Start up (%)	Mem (%)
Radiotime	2,4	6299	1,9	5197	-5,00	-17,49
Mp3 Player	1,3	4096	1,0	3518	-23,08	-14,11
Settings	1,4	4556	1,2	4534	-14,29	-16,73
News	1,9	7181	1,3	4946	-18,75	-31,12
Weather	4,9	5253	4,7	4206	-4,08	-19.93
Mail	1,3	3958	1,0	3876	-23,08	-2,07
Photos	1,1	5837	0,9	4343	-18,18	-19,56
Launcher		8103		6779		-16,34

128 MB RAM

i'm Droid Firmware Design:
Applications startup and memory

Estimate performance and memory (PSS) consumption [ActivityManager.getMemoryInfo()]

- RAM, 64 MB and 128 MB
- Screen resolution 240x240 and 640x480, 160 dpi

	640x480		240x240		640x480 vs 240x240	
App	Start up (s)	Mem (KB)	Start up (s)	Mem (KB)	Start up (%)	Mem (%)
Radiotime	2,0	6354	2,3	5897	-4,17	-7,19
Mp3 Player	1,2	4973	1,1	4460	-8,33	-10,32
Settings	1,7	5180	1,2	4901	-17,65	-5,39
News	1,4	7755	1,5	6898	-21,05	-11,05
Weather	6,0	6514	5,5	5814	-8,33	-10,75
Mail	1,3	5148	1,2	4823	-0,00	-6,31
Photos	1,0	5237	0,9	4642	-10,00%	-11,36
Launcher		9219		9172		-0,51

64 MB RAM

i'm Droid Firmware Design:
Applications startup and memory

Estimate performance and memory (PSS = f[sh,pm]) consumption (procrank)

- RAM 128/64 MB
- Screen resolution 240 x 240,160 dpi

	128 MB		64 MB		128 MB vs 64 MB	
App	Start up (s)	Mem (KB)	Start up (s)	Mem (KB)	Start up (%)	Mem (%)
Radiotime	1,9	5197	2,3	5897	+17,39	+11,87
Mp3 Player	1,0	3518	1,1	4460	+9,09	+21,12
Settings	1,2	4534	1,2	4901	+14,28	+22,58
News	1,3	4946	1,5	6898	+13,33	+28,29
Weather	4,7	4206	5,5	5814	+14,54	+27,65
Mail	1,0	3876	1,2	4823	+16,66	+19,63
Photos	0,9	4343	0,9	4642	+0,00	+6,4
Launcher		6779		9172		+26,09

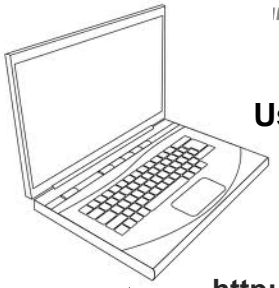
Hypothesis:
- More shared memory
- Increase of shared Memory per process

64 MB RAM

i'm Droid Firmware Design:
Bluetooth Application Interface

- ✓ Bluez 4 native code back-port from Android 2.3 (external/bluetooth/)
- ✓ Bluez 3 (JNI and Java) removal from a Android 1.6 to avoid conflicts with Bluez 4 integration
- ✓ JNI Bluez 4 back-port from Android 2.3 (frameworks/base/core/jni/android_bluetooth_*)
- ✓ Java API Bluez 4 back-port from Android 2.3 (frameworks/base/core/java/android/bluetooth/)
- ✓ OBEX Java code back-port from Android 2.3 (frameworks/base/obex)
- ✓ OPP Service and application back-port from Android 2.3 (packages/apps/Bluetooth).

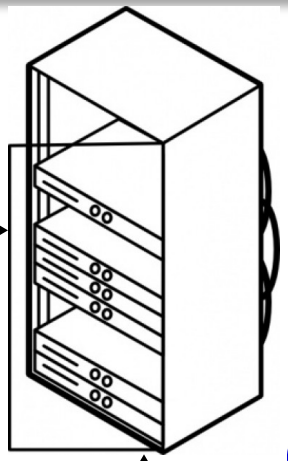
i'm Enterprise



User

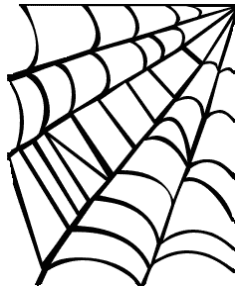
http://im.com

User access to web platform
To manage services
and configuration



Enterprise
Services
Infrastructure
(Data Package Build)

- Scalability
- Many User management
- Backups



Connectivity

3G/UMTS
Connection

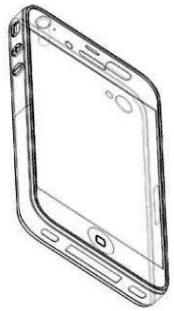
Third Party Services
Data Retriving

- Facebook Notifications
- Twitter Notifications
- Mail Notifications
- News
- Weather
- **Music Providers**
- **Market**
- …

User

Remote Data Package Retrieve

Tethering through Bluetooth



Smartphone

Fine

Thank you…..

Meet you in the Si14's Booth (707) to see the i'm watch prototype.