Using Android Outside of the Mobile Phone Space

Jason Kridner
Khasim Syed Mohammed
Texas Instruments
Challenges for Android in non-mobile

- Amazing success in phones (and tablets)
- Challenges in embedded devices
- No focus from Google
- Becoming more costly
- Not very scalable
- Losing embedded developer interest
Google not focused on non-mobile segment

- Roadmap and future focused on Google Experience Devices

  “writing code to support hardware other than Google's Nexus model has proven to be a tall order for smart phone makers.”

  Christy Wyatt, senior VP & GM, Motorola Enterprise Business

  Source: http://www.pcmag.com/article2/0,2817,2400023,00.asp

- Android market and Google apps unavailable
- Android software stack caters to phone and tablets
- No guarantee of sources until it gets published
- No direct access to Google
Android may be becoming unaffordable

- Hardware requirement increasing significantly
  - ICS: 3D graphics acceleration a must!
  - High-end processors, more memory, etc.
- Customization R&D investments not community sustained
  - Phone submissions rare
  - Embedded never
- Fear of royalties
  - Price depends on who you are
  - Adds to product cost
- Licensing Google apps
Scalability vs. embedded Linux distros

- Android software stack statically tuned for phones and tablets
- Several configuration options are made difficult
- C library and language bindings
- Remote headless operation and configuration
- Limited shell
- Non-Android package management
- Ultra-fast boot times
Losing interest – community developers never brought into camp

- Developers from open communities haven’t fully migrated to Android yet
- Example: gStreamer developers find Android decoders (3gpp) useless for Set Top Box Transport Stream playback
- Ubuntu on next generation dual and quad cores is much efficient with potential desktop app market
- Product manufacturers still see significant potential and need to be on existing software
- Example: Thin client manufacturers moving from x86 PCs to ARM® for low cost, low power still prefer to be on Windows and Ubuntu than Android
- No control and no insight frustrates developers
Can Android really address the future needs of embedded “non-mobile” devices?
with

The Droids

Stop Look Proceed
Embedded “non-mobile” devices have new Jobs…(with all respect) to:

- Bring in new technology
- Enable innovation
- Support new business model
Things are not same, are changing, have changed:

- User experience demand
- Ecosystem enablement establishment
- Industry’s focus need for change
The pinch effect: User demand

**QNX’s Andy Gryc**, Senior Product Marketing manager, QNX Software Systems says

- He’s seen a trained engineer “forget” how to operate an oscilloscope and attempt to use the pinch-and-spread gesture to zoom into a scope trace.

**Beckhoff’s McAtee** takes it further.

- “[If you] combine [multi-touch] functionality with wide format 24-in. screens, device vendors and machine designers would be able to **remove all physical push buttons** from the panel, allowing the user to manage every machine function directly on the touchscreen. This would permit easy scrolling and zooming through dashboards and menus, beyond the capabilities of conventional touchscreen technology.”

**Fujitsu’s Bruce DeVisser**, product marketing manager for touch input

- Technologies have crossed over into the industrial space. “**Haptic feedback**, embodied as a vibration of the touch panel (like how a cell phone vibrates), is very useful for noisy industrial environments”. A display in black mode (power-saving or screen-saver state) is unappealing to [consumer] users if it is covered with fingerprints.

Source: http://m.controleng.com
Connected devices: Ecosystem establishment

**Shawn DuBravac**, Director of Research, Consumer Electronics Association says

“The demand for **standard television has dropped nearly 20%** over the last 12 months while over the same period demand for tablets and smartphones have jumped 20% and 17% respectively. This is not reflective of the end user wanting to replace the main household TV with a tablet, as demand for **internet-ready TV is up by 14%**.”

**QNX’s Andy Gryc** says

“Similarly, **connectivity demands** are spreading from consumer devices as well,” he says. “We’ve had some unusual requests from the industrial segment lately, such as inquiries about adapting **application store** technology into what would be normally considered an isolated system.”

**Audi’s Johan de Nysschen and Dr. Peter Steiner** share

“Continental is actively looking into bringing Google's mobile OS into the automotive realm. More intriguing, **Audi could offer its own App Store**.”

**Fanvil’s Robbin Feng, general manager** says

“**[Android] can offer good expandability and more application service**. Comparing with other OS, it needs to be more standard and open.”
Changing Trend vs. Industry needs

Portable consumer

Home consumer

Application

Portable enterprise

Automotive

Industrial
Final: Look

Smart phone sales increases compared to PCs and Laptops

- As smart phone users increase, user experience changes from that of desktop
- Pre-enabled ecosystem of smartphones is easier and established
- Industry’s rapidly changing trends need to be met
Android is OPEN, FREE to consume, distribute and productize

Android dominates smartphones

Android leads mobile OS

Table 2

| Worldwide Smartphone Sales to End Users by Operating System in 3Q11 (Thousands of Units) |
|-----------------------------------------------|-----------------|-----------------|-----------------|
| Operating System                            | 3Q11 Units      | 3Q11 Market Share (%) | 3Q10 Units      | 3Q10 Market Share (%) |
| Android                                       | 60,490.4        | 52.5             | 20,544.0        | 25.3              |
| Symbian                                       | 19,500.1        | 16.9             | 29,480.1        | 36.3              |
| iOS                                           | 17,295.3        | 15.0             | 13,484.4        | 16.8              |
| Research In Motion                            | 12,701.1        | 11.0             | 12,500.0        | 15.4              |
| Bada                                          | 2,479.5         | 2.2              | 920.6           | 1.1               |
| Microsoft                                     | 1,701.9         | 1.5              | 2,203.9         | 2.7               |
| Others                                        | 1,018.1         | 0.9              | 1,991.3         | 2.5               |
| Total                                         | 115,185.4       | 100              | 81,132.6        | 100               |

Source: Gartner (November 2011)
How to use Android for non-mobile phones

Leverage the brand “Android” for your products
- Focus less on Google’s emphasis on embedded segments

Customize Android as required using NDK
- It’s static but customizable with options to reduce porting headaches

Keep Android products affordable
- Right device, right peripherals and right R&D investment

Join the right community forum
- Leverage the Android Embedded developer community for support
Brand “Android” is important for products

Focus less on Google’s emphasis on embedded segments
- Brand Android adds value; leverage it
- Getting non-phone product out that runs on Android will give much hype and visibility

Don’t rush on a new dessert unless mandatory
- New features for phone generally don’t add value to non-phone products
- Example: Versatile camera capabilities on ICS doesn’t benefit Android PND
- Allow few major brands to release products based on the new version, let stack mature, let community get settled to answer your queries.

Android market and Google apps – you get it if you want it
- With the tablet segment picking up, it’s much easier now compared to phone days
- Mark your product as tablet if you need market and Google apps; focus on CTS compliance
- If the product has significant potential and volume, get it manufactured through Android ODMs; they will get you license for Google apps easily
- In future, major players will own or have their own custom app market

Leverage Android engineering community
- It’s spread across multiple forums like Android porting, arowboat, Linaro, etc
- Stay on older version; it will get enough support and help
Customizing Android is easy

Android is static, customizable and easy to program and manage

**Static** Android is your first source base line
- Android *doesn’t come with everything* that you need on a “native” stack. It gives access to instant apps to prototype, quick demo and benchmark
- **Covers majority of requirements** with pre-integrated connectivity, security, 3D UI and multimedia frameworks

**Customize** as you need
- The **Android software stack is customizable** – add new framework or device-specific pipeline architectures.
- Customization can be achieved with **hardware** – Accessories Dev Kit.

**Program and manage** it easily
- Android is on **Linux** and apps are on **Java** – easy to learn and program
- Reduce porting effort using **SDK and Linux standard APIs** – get isolated from constant changes in drivers and Android stack.
- All the standard advantages that come with Android will continue to remain – **Get best of both worlds**
Software customization example

- **USER APPLICATIONS**
- **CUSTOMER APPLICATION AND UI**
- **CUSTOMER APPLICATION AND UI**
- **ANDROID APPLICATION FRAMEWORK**
- **JNI (JAVA NATIVE INTERFACE) FOR CUSTOM APPLICATION**
- **CUSTOM STACK HOOK TO ANDROID FRAMEWORK IF REQUIRED**
- **ANDROID LIBRARIES**
- **ANDROID Framework**
- **CUSTOM STACK W/ HW ACCELERATION**
- **NDK (NATIVE DEV KIT)**
- **ANDROID LIBRARIES**
- **ANDROID PLUG-INS & HAL**
- **CUSTOM STACK W/ HW ACCELERATION**
- **LINUX KERNEL**
- **STANDARD ANDROID APPLICATIONS**
- **Android native SOLUTION W/ HW ACCELERATION**
- **CUSTOM STACK W/ HW ACCELERATION**

---

*OSCON open source convention*
Hardware customization example

Accessory Development Kit

- Use existing Android platforms
- Design your customization as accessories
- Get your solutions to market faster
Reduce fragmentation

- **Android fragmentation is inevitable**; impact can be reduced drastically

- **Linux foundation’s help** will reduce fragmentation in kernel
  - At the recent kernel summit in Prague, there was a discussion about re-evaluating some of the Android-specific features that are found in Google’s Android kernel, and looking at whether and how to incorporate them into the mainline (kernel.org) kernel - *From Tim Bird*
  - Today we apply around 300+ patches to mainline kernel

- Google’s serious about keeping **Android less fragmented**
  - Honeycomb was resulting in fragmentation of Android
  - Google took a vice decision of holding Honeycomb tablet edition release and merging the code base into ICS

- Silicon providers are **playing safe**
  - Silicon entitlement for non-mobile frameworks is built on NDK framework
  - Also avoids legal and licensing hurdles (if any) with Android’s apache 2.0

- **OEM/ODM keep Android and Kernel versions separate**
  - Few are using FroYo with Kernel 3.2 to leverage R&D investments for features they added

- Community forums like Linaro, 0xlab and arowboat.org are **here to help**
  - Talk about fragmentation, they will guide you on maintaining and managing customization in a community-friendly way
Keep Android products affordable

- **Android success makes high end technology affordable to non-mobiles**
  - Peripherals like DDR, 4” and 7” LCDs, capacitive touch panels are now affordable as they are mass manufactured
  - Multicore, high-end processor road map is moving at lighting speed
    - Scalable TI Sitara™ AM335x ARM® Cortex™-A8 processor (800MHz with SGX and Neon acceleration) with DDR3 controller

- **Reduce R&D investments; leverage large pool of trained engineers**
  - Android is Linux-powered – the engineering community beyond imagination
  - Android is based on the Java programming language giving instant access to trained app developers

- **Open development and debug tools**
  - Eclipse offers a dedicated plug-in for Android (ADT)
  - Allows setting up Android projects, create application specific UI, add components, debug, and then export .apks – IDEs are open, free and affordable

- **Infinite and healthier ecosystem compared to any other OS distro**
  - Android has one of the largest developer community, driving not only application layer content (more than 300,000 applications are available), but also Android middleware components

- **Microsoft royalties are unknown**
  - Speculations say $8 to $10 per unit
  - Still less when compared to Windows7 for $20-30
Join and contribute to right community

- **Android is built on open-source software and open hardware**
  - Android leverages existing open-source projects, maintains and manages the sources
  - Many hardware-component vendors feed in source code for specific drivers

- **Contribute to Android software stack**
  - Every Linux developer is contributing directly or indirectly to the success of Android
  - Push your changes to respective repositories, it will get in Android some time
  - Example: Kernel mainline

- **Join the appropriate community forums**
  - There is definitely an answer to your non-mobile problem
Call for action

- **Android is “clopen”:** 100% closed and 100% open
  - Use the latest open version without waiting for next one

- **Fragmentation**
  - Get Android kernel differences onto mainline, work with Linux Foundation

- **Community collaboration** to help non-mobile developers
  - Bring your solutions to silicon vendors and ODMs anxious to address non-mobile
    - Example: arowboat.org
  - Seed Android knowledge in non-mobile focused community forums
    - Example: beagleboard.org
  - Involve and motivate domain experts to participate in Android

- **Make Android affordable**
  - Find ways to get rid of Microsoft royalties?

- **Reduce R&D efforts and time to market on latest Android version**
  - Work with open hardware communities to enable low cost Android evaluation kits on GED processors
    - Example: pandaboard.org
  - Help Linaro migration of latest AOSP to non-GED processors
    - Example: BeagleBone
Thank you