Making Android More Wearable

THE CHALLENGES OF ADDING MULTI-SPORTS SENSORS AND RADIOS

GIL ZHAIEK
RECON INSTRUMENTS
A Little History

- Incorporated January 2008
- Headquarters in Vancouver, Canada
- Operated from the small office lab spaces rented from the University of British Columbia
- Moved to Downtown Vancouver in April 2010
- Originally looked into developing a HUD product for swimming but due to existing patent and form factor challenges – had to refocus their efforts
- Realized the potential for incorporating Heads-up Displays into ski goggles is far greater than swimming HUDs
RTOS Products – Transcend / MOD

- Transcend was for the 2010-2011 ski season
- MOD was for the 2011-2012 ski season
- RTOS was bought from IAR (SEGGER)
- Both used ST ARM9 Chip @96Mhz
- 96KB ram and 256KB flash
• Recon’s first Android Device (Gingerbread)
• TI’s Omap3 single core – running at 800Mhz
• Based on reference design of the BeagleBoard
• On device display – 428x240 pixels (can’t pass CTS)
• Controlled by a Bluetooth LE Remote control (D-Pad)
• 6 hours run time while display and GPS on
• Acc, Gyro, Mag, Alt, Temp, BT, GPS
Snow2 – October 2013

- Android 4.1.2 - Jellybean
- TI’s Omap4 dual core – up to 1Ghz
- Based on reference design of the Blaze board
- 9 hours run time while display and GPS on
- MFI
- WiFi
Jet – Summer 2014

- Superset of Snow2
- Speaker, Microphone, Camera
- ANT+
- Finger Navigation
- Detachable from the Sunglass
HUD App Design

Glanceable information on a 428x240 screen
• General Design
  – Glanceable Information: Design must take into account that the user looks at the screen for only a moment - don’t burden the user with too much information.
  – Limitation: Optical limitation causes the corner of the screen less sharp as the center.
  – Flat Design
• 3D Maps Control using head gesture and remote control:
  – Move head up to look at the horizon and move head down to look at your location.
  – Rotate head left and right to look left and right
  – Remote control is still used to zoom in and out
• Maps are back to be 2D and are provided as a maps service for 3rd party apps
• OpenStreetMaps are now supported: OpenSkiMaps and OpenCycleMaps and future terrain support
Recons Patented HUD Keyboard
Android OS Bringup

Bringup, stabilization and quality
Android Bring Up - Questions

• Android
  – What is Android OS?
  – Where is the documentation?
  – How do we upgrade the OS while development?

• How do we expand Android?
  – How do we add Bluetooth LE to support our remote?

• How do we modify Android?
  – We didn’t have a touchscreen or a keyboard…
Android Customization

• Replaced most of the default resources (strings, layouts) via the overlay folder
• Replaced the entire BlueZ stack with ours (SS1)
• Removed most stock apps, left providers, couldn’t remove system services (we tried)
• Modified Android’s framework (overlay folder wasn’t enough)
• Forced the Bluetooth to be turned on by default during boot
Bluetooth – Android

- Bluetooth Connectivity was basic when the product came out.
- Android phone connectivity was added later by using SPP for trip syncing.
- SMS and Phone calls were easily added by capturing the incoming messages in our Android app and passing it to the HUD via SPP.
- For internet access we supported PAN – this required the user to remember to switch it on the phone.
- An alternative for PAN was Recon’s Web API.
Bluetooth – iOS

- Bluetooth LE (BLE) was added to iPhone 4s
- BLE is very slow 700 bytes per second
- Support for MAP (receiving text messages)
- Support for HFP (Caller ID)
  - Our Snow devices don’t have a speaker
  - Old Version: We had to disconnect the HFP on Answer, monitor the call using our iOS app and transfer call-state via BLE to reconnected when call ended.
  - New Version: We only disconnected the Audio layer of the HFP – Bluetooth certification allows this
- MOD Live: Connectivity through BLE
- Snow2: Connectivity through MFI
Connectivity

• We had to reverse some of our Bluetooth profiles from master to sink
  – HFP: The HUD is also a Bluetooth headset
  – MAP: The HUD can received Text Messages from another device

• WiFi
  – Added in Snow2 and Jet but was disabled by default due to consumption
  – Not really useful – yes we can run with a WiFi hotspot but why not just use Bluetooth
  – Main use case is GoPro remote video support
GPS - Problem

• Smart phones use network (WiFi and Cell Towers) to assist for location – AGPS
• Smart phones are used daily so Ephemeris and Almanac is always fresh (no cold start)
• Snow devices don’t really have a problem because of the clear skies and the fact that most user turn on their device while going up a lift
• Jet is used in urban environment after few weeks of no use (Almanac is not fresh) which increases TTFF (cold start)
GPS – Solution

- Innovative AGPS design was implemented in house
- Reduced cold lock from few minutes to few seconds when phone connected
Sensors

• Contains more sensors than your average phones:
  – Accelerometer
  – Gyroscope
  – Magnetometer
  – Pressure
  – Ambient Temperature (located on the remote)

• University of British Columbia helped us with Sensor Fusion:
  – Android Composite Sensors
  – Ability to create a more accurate and grained location points (UBC + R&D)

• Recon provides an SDK to extend Android’s sensors:
  – Jump detection and profile
  – Current altitude (GPS altitude is not accurate enough)
  – Ski run detection (based on altitude change)
• Calibration Process was needed for the “Iron Effect”
• Our fellow Canadians in Dynastream (responsible for ANT) helped us in the bring up – so thank you!
• Hardware limited us to either support BLE or ANT devices by loading a different Firmware when switching a platform
• Testing…
Sensors – FIFO Mode

• Even before KitKat’s batch mode we had to provide support for FIFO mode in JellyBean.
• The reason behind this is consumption. For example, a developer might want to capture sensor data but due to consumption we want to minimize data transfer between layers – Kernel to Native User Space to Java.
• When designing our hardware we chose sensors that have some memory and the kernel pulls the data only when the sensor is almost full.
• User space driver copies a burst of raw sensor data from the kernel driver only when the HW FIFO has filled.
Aggressive Development Cycle

• As a small company we are limited in resources such as time, money, developers - so we need to be efficient
• We outsource what will consume most of the team’s time such as Camera (MMS), Bluetooth stack (StoneStreet1)
• We use the university (UBC) to offload some of the complex algorithms
• We couldn’t move deadlines – the snow fall is not going to wait
• Snow2 is a subset of Jet, just a different form factor
Various Issues

• A small board – everything is crammed together
  – Prepare sensors drivers before the first revision of the board comes – you will need to capture stream of raw data right after the bring-up to validate there is no noise on the line
  – GPS might work but SNR need to be verified quickly

• A small battery
  – The boot up sequence in Android needs to flatten not to cause spikes in consumption that the battery can’t provide
  – Use systrace to validate that the applications are doing what they need to do – for example, how many FPS are in 1 second?

• Bluetooth can be tricky
  – A2DP headset nearby caused our GPS to crash
  – Issues while playing A2DP music and scanning
Various Issues - cont...

- Finger Navigation can be tricky in sunlight
Certifications

- Bluetooth certification will take lots of your team’s time
- Yes, even if you don’t have a Bluetooth logo on the box – you still need to pass certification
- No, you can’t bribe the guy that tests your equipment with free Ski goggles
- WiFi certification is not as bad as Bluetooth Certification
- FCC certification is left mostly to the Hardware team – so we don’t care
- We even needed to pass FDA certification with our Sunglass product (not the Snow)
Upgrade Process

RI Server
- update_1.1_to_1.2.bin
- update_1.0_to_1.2.bin

uBoot.bin
- system patches
- kernel.p

uBoot v1.1
- hard reset

Active System
- Kernel v1.1
- Core System v1.1
- RI Services
- Android Services

recon updater
- Soft reset
- Update system
- Update apps

Snow Apks
- dashlauncher.apk v1.12
- radar.apk v1.14
- ... v1.xx

Factory Partition
- Recovery Kernel v1.0
- boot.img (v1.0)
- snow.zip (v1.0)
- system.img (v1.0)
- mass storage (v1.0)

userdata
- dalvik and dex

Mass Storage
- rbs
- map data
- other

Cache Partition
- update_1.1_to_1.2.bin
Testing MOD Live

How consumers think we test MOD Live

How non-BC Canadian assume I test MOD Live

How my mom thinks I test MOD Live

How I wanna test MOD Live

How the Support team thinks I test MOD Live

How I really test MOD Live
Thank you

Any Questions?

http://www.linkedin.com/in/gilzhaiek