Mainline on form-factor devices / Improving AOSP

John Stultz
Topics from Linux Plumbers

- Barriers to running mainline on form-factor devices
- Improving AOSP + Summary of related talks
Mainline on form-factor devices

Why Bother?
Benefits

Validate upstream changes (enable CI)

Allow for more experimentation

Avoid devboard functionality gaps in testing

Enable selfish maintainer interest

Improve collaboration
Hardware

Key requirements
Unlockable bootloader
Most device’s bootloader
Access to the serial UART
Nexus Headphone Debug UART

https://android.googlesource.com/device/google/debugcable/+/master
USB-C to the rescue?

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>GND</td>
<td>Ground return</td>
<td>B12</td>
<td>GND</td>
<td>Ground return</td>
</tr>
<tr>
<td>A2</td>
<td>SSTXp1</td>
<td>SuperSpeed differential pair #1, TX, positive</td>
<td>B11</td>
<td>SSRXp1</td>
<td>SuperSpeed differential pair #1, RX, positive</td>
</tr>
<tr>
<td>A3</td>
<td>SSTXn1</td>
<td>SuperSpeed differential pair #1, TX, negative</td>
<td>B10</td>
<td>SSRXn1</td>
<td>SuperSpeed differential pair #1, RX, negative</td>
</tr>
<tr>
<td>A4</td>
<td>V_{BUS}</td>
<td>Bus power</td>
<td>B9</td>
<td>V_{BUS}</td>
<td>Bus power</td>
</tr>
<tr>
<td>A5</td>
<td>CC1</td>
<td>Configuration channel</td>
<td>B8</td>
<td>SBU2</td>
<td>Sideband use (SBU)</td>
</tr>
<tr>
<td>A6</td>
<td>Dp1</td>
<td>USB 2.0 differential pair, position 1, positive</td>
<td>B7</td>
<td>Dn2</td>
<td>USB 2.0 differential pair, position 2, negative</td>
</tr>
<tr>
<td>A7</td>
<td>Dn1</td>
<td>USB 2.0 differential pair, position 1, negative</td>
<td>B6</td>
<td>Dp2</td>
<td>USB 2.0 differential pair, position 2, positive</td>
</tr>
<tr>
<td>A8</td>
<td>SBU1</td>
<td>Sideband use (SBU)</td>
<td>B5</td>
<td>CC2</td>
<td>Configuration channel</td>
</tr>
<tr>
<td>A9</td>
<td>V_{BUS}</td>
<td>Bus power</td>
<td>B4</td>
<td>V_{BUS}</td>
<td>Bus power</td>
</tr>
<tr>
<td>A10</td>
<td>SSRXn2</td>
<td>SuperSpeed differential pair #2, RX, negative</td>
<td>B3</td>
<td>SSTXn2</td>
<td>SuperSpeed differential pair #2, TX, negative</td>
</tr>
<tr>
<td>A11</td>
<td>SSRXp2</td>
<td>SuperSpeed differential pair #2, RX, positive</td>
<td>B2</td>
<td>SSTXp2</td>
<td>SuperSpeed differential pair #2, TX, positive</td>
</tr>
<tr>
<td>A12</td>
<td>GND</td>
<td>Ground return</td>
<td>B1</td>
<td>GND</td>
<td>Ground return</td>
</tr>
</tbody>
</table>

*USB 2.0 differential pair connects only in one position; position 2 is not physically present in the plug*
Avoiding Binary Blobs
Binary Blobs

QmlZ2VzdCBpc3N1ZSB3LyBHUFUKCldpZmkvQmx1ZXRvb3RoIGZpcm13YXJlIGFsc28gcHJvYmxlbWF0aWMsIHNpbnRvZXMgZWRhc2VydG9yIHVwc3RyZWFtZW1lbnRheCgpOZXQgZWZmZWN0OiBSZXN0cmljdCB1c2FibGUgaGFyZm9yIHRocmFpbmdsZXMgY29udGVudGo=

Linaro

connect

San Francisco 2016
Android Kernel Patches
Decreasing common.git delta
Remaining areas (android-3.18)

15.9% drivers/usb/gadget/
12.7% drivers/video/adf/
13.4% net/netfilter/
  8.7% include/
6.1% drivers/input/
5.4% drivers/staging/android/fiq_debugger/
4.7% drivers/cpufreq/
3.5% arch/arm/
3.3% arch/arm64/
(~25% spread out elsewhere)
Infrastructural deltas

Android kernel and upstream kernel interfaces still differ in places.

So a fair amount of userspace may need to change
Lagging Upstream SoC Support
Device Tree conversions

Adding support upstream using device tree has been painful for moving 3.4 board-file era devices forward.

Recent devices have been shipped using DT with 3.10 (still 11+ revisions behind).
Achievement unlocked (Xperia Z3 + mainline + a few patches). Now we need to get those patches merged.
Credits!

Björn Andersson
Stephen Boyd
Rob Clark
Kumar Gala
Lina Iyer
Ivan T. Ivanov
Srinivas Kandagatla
Amit Pundir
Sumit Semwal
Vinay Simha
Archit Taneja

And everyone else at Qualcomm, Sony, Inforce, and Linaro who have helped get code upstream.
Currently ~25 kernel patches (v4.3-rc1)

DeviceTree changes
- hw_rng
- usb gadget
- pinctrl / gpio-keys

Real features
- mmc > 8 partitions
- MTP usb gadget
- “reboot bootloader”
- pm8921 pmic gpio

Build helpers
- flo_defconfig
- Android.mk
- ATAG MEM fixup

Hacks for now
- virtual fb
- mmc wp gpio hack
- allow broken gcc 4.8
<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display panel</td>
<td>Bluetooth</td>
</tr>
<tr>
<td>GL acceleration</td>
<td>Sensors</td>
</tr>
<tr>
<td>USB hotplug</td>
<td>Audio</td>
</tr>
<tr>
<td>Battery charger</td>
<td>Cameras</td>
</tr>
<tr>
<td>Power Management</td>
<td>NFC</td>
</tr>
<tr>
<td>Wifi</td>
<td>Slimport</td>
</tr>
</tbody>
</table>
Reproduce it yourself!

$ repo init -u https://android.googlesource.com/platform/manifest -b android-5.1.1_r6

$ git clone https://git.linaro.org/people/john.stultz/AOSP/flo-mainline/manifest.git .
repo/local_manifests

$ repo sync

<fetch & install the (lmy48g) firmware blobs from
   https://developers.google.com/android/nexus/drivers >

$ build/envsetup.sh
$ lunch aosp_flo-userdebug
$ make -j24
Already seeing benefits

Making clear what code is most critical to upstream, helping find pain points

Using as a test platform to validate transition to ConfigFS gadget

Targeting for cenalloc proof of concept
Wishing and hoping

$199 Nexus-like tablet released on “decently upstreamed” SoC

Standard USB-C alternative UART mode.
Improving AOSP
Admit we have a problem
Problem areas

- Targeting multiple devices
- Vendor & reference HALs
- Missing best practices
- Collaborative limits of AOSP
Targeting multiple devices

- Android One
- Intel’s IRDA platform
- Project Ara
### Android One - Partition time

<table>
<thead>
<tr>
<th>Directory</th>
<th>Mapping</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/system</td>
<td>-&gt;</td>
<td>Architecture</td>
</tr>
<tr>
<td>/vendor</td>
<td>-&gt;</td>
<td>SoC</td>
</tr>
<tr>
<td>/odm</td>
<td>-&gt;</td>
<td>Device</td>
</tr>
<tr>
<td>/oem</td>
<td>-&gt;</td>
<td>OEM/Carrier</td>
</tr>
</tbody>
</table>
IRDA - Boot time

- Check BIOS/Bootloader tables at boot
- Bind-mount HAL driver directories
- Fuse Filesystem for config data
Project Ara - Run time

- Create standard device classes
- Hotplug aware HAL layers
- Slowly adapt framework
Multiple solutions

- Partition based customization
- BIOS specified bind mounting and fuse filesystems
- Standard device classes & Hotplus aware HALs
- Likely to see these approaches combined!
Ideas for build improvements

- Reducing duplication in device/ dirs
- Common configuration infrastructure
- Kconfig for AOSP?
Vendor & Reference HALs

- HAL unification effort
- Improving Reference HALs
- KMS/DRM based HWComposer
Best Practices

- Better reference HALs
- Improving documentation
- Providing tutorials
- Advising vendors ship custom applications as apks via store
Collaboration w/ AOSP

- Develop community for shared HALs
- Community reviews on Gerrit
- Delegated maintenance
- Delegated device validation