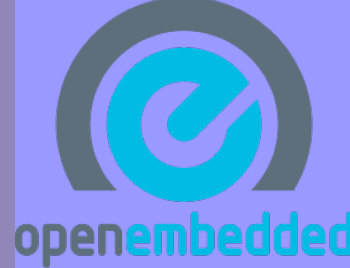




SCaLE 13x – Open Source Hardware



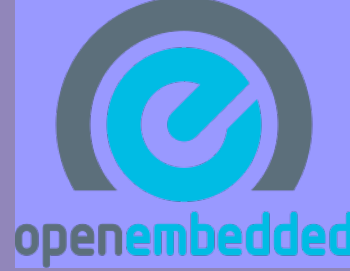
ARM Linux Kernels and Graphics Drivers on Popular "Open" Hardware: Bleeding Edge vs. Vendor Blobs and Kernel Forks - How Much is in Mainline, and How Open is Open?

*Prepared / Presented by
Stephen Arnold, Principal Scientist VCT Labs
Gentoo Linux / OpenEmbedded Developer*





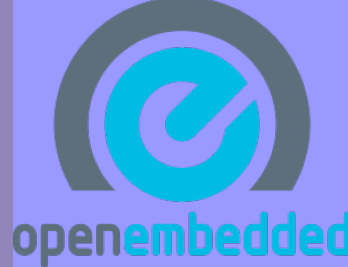
What is ARM/Embedded?



- Small Single Board Computer (SBC) or System on Chip (SoC)
 - Very resource-constrained
 - Zaurus 5000-D – 32 MB RAM, StrongARM SA-1100 (DEC/ARM)
 - Kurobox HG – 128 MB RAM, 256 MB flash, no display (G3, no altivec)
- Modern devices blurring the lines between “embedded” and desktop/server-class hardware
 - Multicore CPUs – 2/4/8 cores
 - Per-core FPUs - VFP3/VFP4, NEON
 - Multicore GPUs – 192-core Cuda on Tegra K1
 - Accelerated HD video processing
 - USB3, 10/100/1000 Ethernet, SATA, HDMI



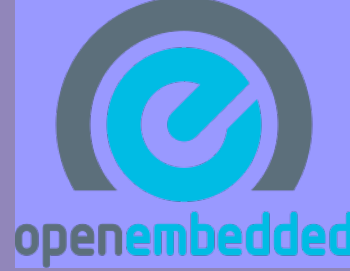
ARM Devices and Graphics Hardware



- ARMv7 HardFloat VFP/NEON
 - Wandboard / udoo / cubox-i - iMX.6 quad core, Vivante GPU
 - Beaglebone black / white - AM335X single core, OMAP3 / SGX GPU, PRUs
 - Sunxi MK802-II 1GB TV stick - Allwinner A10 single core, Mali GPU
 - Samsung Chromebook - Exynos5 dual core, Mali GPU
 - Acer Chromebook / Jetson TK1 – Tegra K1 quad-core, NVIDIA Cuda GPU
 - Genesi SmartBook - Freescale iMX.5 single core, AMD z430 GPU



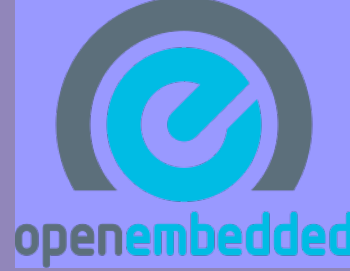
ARM Graphics Hardware cont.



- ARMv7 HardFloat VFP (no NEON)
 - Trimslice Diskless - NVIDIA Tegra 2 dual core CPU/GPU
- ARMv6 HardFloat VFP (no NEON)
 - Raspberry Pi - Broadcom SoC single core, VideoCore IV GPU



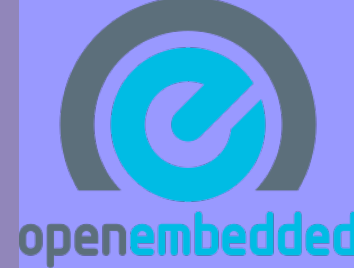
The State of ARM Graphics



- (mostly) Current Vendor Blobs
 - Cubox-i4Pro (iMX.6)
 - RaspberryPi (VideoCore IV)
 - Allwinner (Mali)
 - ChromeOS K1 (Tegra124)
 - TI (OMAP/SGX)
- Open Source Graphics
 - Tegra/Nouveau – opentegra/grate, nouveau w/firmware
 - Broadcom/VideoCore IV – weston/wayland, fbturbo
 - Mali – lima, fbturbo
 - OMAP – omapfb, omap3
 - Vivante – etna-viv, fbturbo
 - Adreno – freedreno (2D/3D, xorg)



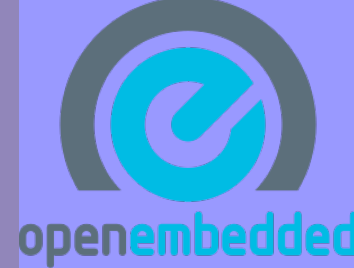
Vendor Kernel Forks



- Typically a single (older) kernel branch with lots of patches
 - Minimal backporting (maybe none)
 - Forwardporting to new branch can take a long time...
- Versions range from 2.6.31.14.x to 3.14.x and later
- Configuration can be brittle
 - Device Tree vs. Legacy Driver Model
 - Warnings vs. Errors
 - Modules vs. Built-in
 - Missing/Incorrect Config Options
 - General Config Changes
- Firmware
 - Network, graphics/drm, audio, camera



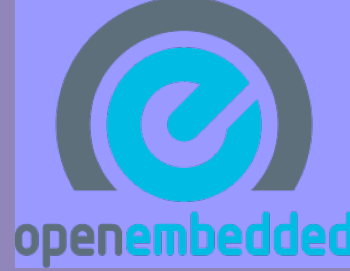
Vendor U-boot Forks



- U-boot forks tend to proliferate like kernel forks
 - Upstream convergence (DENX) somewhat better
 - Fewer vendor/device-specific patches for newer devices
 - Google Chrome devices use “secure” bootloader and EFI partition scheme
 - Jetson K1 can use fastboot or U-boot
 - Common deployment scenario to SDCard is 2 partitions either ext2/ext4 or vfat/ext4 for boot/root
 - Copy U-boot image to /boot partition
 - Load kernel/dtb from /boot directory on root partition
 - Custom boot options via uEnv.txt or boot.scr
 - Board / device-specific differences by vendor even with common SoC



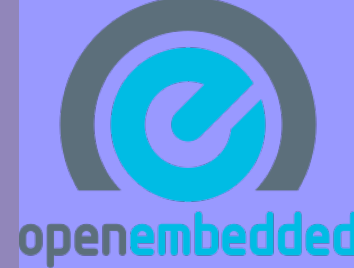
Current State: Kernel



- http://elinux.org/Device_Tree
- <https://eewiki.net/display/linuxonarm/Home>
- Device Trees are main difference between vendor and mainline
 - Kernel support varies by device
 - `<linux_src>/arch/arm/boot/dts`
 - DRM/Graphics interfaces determined by kernel version (more or less) and libdrm/mesa/xorg
- Vendor device tree support an ongoing process
 - sunxi-linux and raspberrypi on github
 - Freescale, TI, Atmel, Allwinner on eewiki (only specific devices)



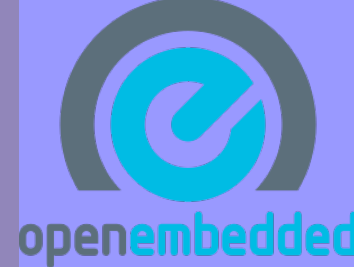
Current State: Graphics



- https://en.wikipedia.org/wiki/Free_and_open-source_graphics_device_driver
- Software Architecture in Flux
 - Xorg / DRM / OpenGL/GLX, EGL/GLES, OpenCL, etc
 - Legacy driver model going away
 - Gallium3D architecture is new model
- Tested Hardware
 - Tegra20/30 – Full mainline kernel, DRM/Mesa/Xorg (git)
 - Tegra K1 – L4T (legacy) and Nvidia SDK (r21.2)
 - Vivante – **etna-viv**, recent kernel, DRM/Mesa/Xorg (git)
 - Efikamx (AMD) – Closed source only, ancient kernel
 - OMAP/PowerVR – TI / open source OMAP framebuffer
 - RPi – BCM firmware, fbturbo/fbdev
 - Mali – arm-soc, **lima**, fbturbo/fbdev



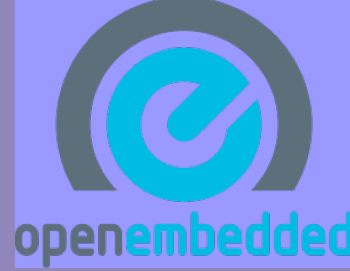
Vendor OS Options



- Typically at least one Linux distribution and Android
 - Legacy kernels, binary blobs, (almost) no device tree support
- Some vendors have several (mostly) current options
 - RaspberryPi – n00bs (allows install of various distros including non-Linux RiscOS)
 - BeagleBone – Yocto/OE (several flavors), Debian, Ubuntu, Gentoo
 - Chromebooks (various) – Chrubuntu script, Debian, Gentoo, Arch, Fedora
 - Udoo – udoobuntu, Android, XBMC / OpenElec, and various other distros



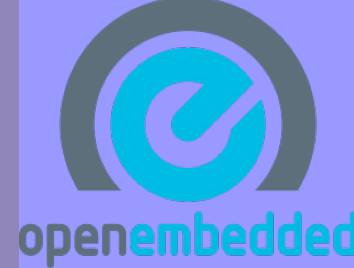
Build / Bootstrap Options



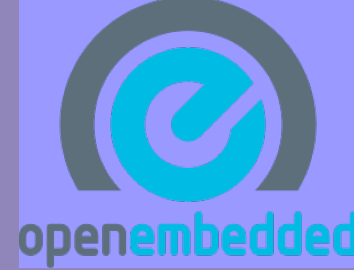
- Yocto – if there's a vendor BSP that contains your “machine” (if not you can create one)
 - Can build on modest desktop machine
 - Install build deps, clone poky repo, build beaglebone
- Gentoo – ARM stage3 builds every few weeks
 - Build native (the Gentoo Way)
 - Build in a (faster) chroot
 - Build in a VM
 - Some Gentoo-dev hardware: <http://bit.ly/1rLmyDz>
- Debian/Ubuntu – If you can boot it, you can dbootstrap it
 - The ubiquitous blog post: How to install debian on a...
 - Debian-kit for Android (in the App Store!)



Deploy Steps



- Find or build a bootable disk image
- Mount / chroot from device / vendor OS
- Get a bootloader (usually u-boot), kernel, rootfs
 - Make an SDCard -or- setup a tftp/nfs boot server
- Many newer devices boot from SDCard and/or flash
 - TV Stick, Chromebook, RPi, Beaglebone, Cubox, etc
- Basic steps are: 2-partition card format (root, boot), bootloader image in boot (or dd'd to card), with kernel image and device tree blob in /boot dir
 - See eewiki for several devices running on mainline:
 - <https://eewiki.net/display/linuxonarm/Home>
 - Maintained by Robert C. Nelson



This work is an original work by Stephen Arnold <sarnold@vctlabs.com>
<<http://www.vctlabs.com>>

Portions copyright 2014 Stephen L Arnold. Some rights reserved.

The Gentoo Linux logo is Copyright 2006 Gentoo Foundation, used with permission.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/1.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.



Please contact Stephen Arnold <sarnold@vctlabs.com>
for commercial uses of this work.