NVIDIA TEGRA LINUX DRIVER PACKAGE

PG_06706-R24 | May 16, 2016 | R24.1 Release

Development Guide

TABLE OF CONTENTS

Overview

Welcome to the *NVIDIA Tegra Linux Driver Package Development Guide*. It is intended for software engineers to help them understand the NVIDIA® Tegra® Linux Driver Package, commonly known as Linux for Tegra (L4T). Use this documentation to learn how to set up L4T, and how to get started developing systems software and applications that target compatible reference hardware from NVIDIA.

The following diagram shows the architecture of L4T and related components.

This documentation is preliminary and subject to change. Consult your NVIDIA representative for additional information and to request documentation updates.

The following topics are included in the *Development Guide*.

Components	Description
Sample Code/Applications	Sample source code for developing embedded applications for the Jetson platform.
V4L2	Tegra V4L2 camera driver bypasses the Tegra ISP and is based on Version 2 of the Linux kernel video capture and output device API and driver framework.
X11	X11 X Window System driver.
libjpeg	C library for reading and writing JPEG image files.
CUDA	NVIDIA® CUDA® parallel computing platform and API for CUDA-enabled GPU.
VisionWorks	NVIDIA® VisionWorks™ software development package for computer vision (CV) and image processing.
EGL	Interface between Khronos rendering APIs such as OpenGL ES or OpenVG and the underlying native platform window system.
OpenGL ES	Cross-platform API for full-function 2D and 3D graphics on embedded systems
cuDNN	NVIDIA® CUDA® Deep Neural Network library.

Package Manifest

Kernel
Boot Loader
NV Tegra

The NVIDIA® Tegra® Linux Driver Package is provided in the following tar file:

```
Tegra<t-arch|ver>_Linux_<release_num>.<version_num>_<release_type>.tbz2
```

The following table lists the directories (denoted by a trailing slash /) and top-level files that are created when you expand the tar file.

Directory or Filename	Description
bootloader/	Boot loader and related components.
bootloader/<platform|ver>/	Platform-specific files.
bootloader/<platform|ver>/BCT/	Platform-specific Boot Configuration Table (BCT) files.
bootloader/<platform|ver>/cfg/	Configuration files for specific `<platform|ver>`.
bootloader/<platform|ver>/<board_and_rev>/ bootloader/p2371-2180-devkit/ bootloader/p2371-2180-devkit-24x7/	TBD files for specific `<board_and_rev>`.
kernel/	Kernel images and kernel modules.
kernel/dtb/	Kernel Device Tree Binary (DTB) files for the particular SoC.
nv_tegra/	NVIDIA drivers and sample applications.
nv_tegra/nv_sample_apps/	NVIDIA sample applications.
rootfs/	Staging directory for the root filesystem.
rootfs/README.txt	README for the root filesystem.
apply_binaries.sh	Script to apply `nv_tegra` components.
elf-get-entry.py	Python script to extract and print entry point address of an ELF-format binary.
flash.sh	Script to flash the boot loader and kernel from the package.
<platform>.conf <board_and_rev>.conf	Configuration file(s) for `flash.sh` specific

p2371-2180-devkit.conf p2371-2180-devkit-24x7.conf	to the development platform represented by `<platform>` or `<board_and_rev>`.
source_sync.sh	Script to download kernel and U-Boot source.

Documentation

Tegra Linux Driver Package (L4T) also includes the following documentation:

- Tegra_Linux_Driver_Package_Release_Notes_`<release>`.pdf
- Tegra_Linux_Driver_Package_Documents_`<release>`.tar

Kernel

Kernel Supplements TBZ2

The `kernel` directory contains the following directories (denoted by a trailing slash /) and files.

Directory or Filename	Description
dtb/	SoC-specific kernel Device Tree Binary (DTB) files.
dtb/*.dtb	DTB files specific to various board types.
dtc	Device-tree-compiler binary.
Image	Kernel binary image.
kernel_headers.tbz2	Kernel header files needed for compiling kernel modules. You can download these headers and sources from the nv_tegra git server.
kernel_supplements.tbz2	Loadable kernel modules specific to the included kernel zImage that was built with the defconfig enabled for the device.
LICENSE	GNU General Public License (GPL).
LICENSE.dtc	GNU General Public License (GPL) for the device-tree-compiler binary.
zImage	Compressed kernel binary image.

Kernel Supplements TBZ2

The following table lists the contents available upon decompressing the `kernel_supplements.tbz2` archive, located at:

```
kernel/kernel_supplements.tbz2
```

Filename	Description
lib/	-
lib/firmware/	-
lib/firmware/nvavp_helper.bin	AVP ucode test binary (aarch64 only).
lib/firmware/tigon/	-
lib/firmware/tigon/tg3.bin	TG3 USB kernel driver file.
lib/firmware/tigon/tg3_tso5.bin	TG3 USB kernel driver file.
lib/firmware/tigon/tg3_tso.bin	TG3 USB kernel driver file.
lib/modules/	-
lib/modules/<kernel_version>/*	Kernel modules.

Boot Loader

The `bootloader` directory contains the following directories (denoted by a trailing slash /) and files.

Directory or Filename	Description
<platform|ver>/	<platform|ver>-specific boot loader directory.
<platform|ver>/BCT/	Platform-specific BCT directory.
<platform|ver>/BCT/*.cfg	Boot Configuration Table (BCT) files for Jetson-TX1.
<platform|ver>/cfg/	Platform-specific configuration directory.
<platform|ver>/cfg/board_config_ers_e2220.xml	Platform-specific configuration file.
<platform|ver>/cfg/board_config_p2595.xml	Platform-specific configuration file.
<platform|ver>/cfg/board_config_p2597.xml	Platform-specific configuration file.
<platform|ver>/cfg/board_config_p2597-devkit.xml	Platform-specific configuration file.
<platform|ver>/cfg/flash_l4t_t210.xml	Platform-specific configuration file.
<platform|ver>/cfg/gnu_linux_fastboot_emmc_full.cfg	Platform-specific configuration file.
<platform|ver>/cfg/gnu_linux_tegraboot_emmc_full.xml	Platform-specific configuration file.

<platform|ver>/<board_and_rev>/	Boot loader <board_and_rev>-specific directory.
<platform|ver>/<board_and_rev>/ extlinux.conf.emmc	<board_and_rev>-specific U-Boot config file for booting off the internal EMMC.
<platform|ver>/<board_and_rev>/ extlinux.conf.nfs	<board_and_rev>-specific U-Boot config file for booting off the nfs root.
<platform|ver>/<board_and_rev>/ extlinux.conf.sdcard	<board_and_rev>-specific U-Boot config file for booting off the SD card.
<platform|ver>/<board_and_rev>/ extlinux.conf.usb	<board_and_rev>-specific U-Boot config file for booting off USB flash storage device.
<platform|ver>/<board_and_rev>/u-boot	<board_and_rev>-specific U-Boot boot loader binary.
<platform|ver>/<board_and_rev>/u-boot.bin	<board_and_rev>-specific U-Boot boot loader binary.
<platform|ver>/<board_and_rev>/u-boot.dtb	<board_and_rev>-specific U-Boot device tree binary.
<platform|ver>/<board_and_rev>/u-boot-dtb.bin	<board_and_rev>-specific U-Boot device tree binary.
<platform|ver>/p2371-2180-devkit/	Boot loader p2371-2180-devkit specific directory used for `jetson-tx1`.
<platform|ver>/p2371-2180-devkit/ extlinux.conf.emmc	p2371-2180-specific U-Boot config file for booting off the internal EMMC.
<platform|ver>/p2371-2180-devkit/ extlinux.conf.nfs	p2371-2180-specific U-Boot config file for booting off the nfs root.
<platform|ver>/p2371-2180-devkit/ extlinux.conf.sdcard	p2371-2180-specific U-Boot config file for booting off the SD card.
<platform|ver>/p2371-2180-devkit/ extlinux.conf.usb	p2371-2180-specific U-Boot config file for booting off USB flash storage device.
<platform|ver>/p2371-2180-devkit-24x7/	Boot loader p2371-2180-devkit-24x7 use case specific directory.
<platform|ver>/p2371-2180-devkit-24x7/ extlinux.conf.emmc	p2371-2180-devkit-24x7 specific U-Boot config file for booting off the internal EMMC.
<platform|ver>/p2371-2180-devkit-24x7/ extlinux.conf.nfs	p2371-2180-devkit-24x7 specific U-Boot config file for booting off the nfs root.
<platform|ver>/p2371-2180-devkit-24x7/ extlinux.conf.sdcard	p2371-2180-devkit-24x7 specific U-Boot config file for booting off the SD card.
<platform|ver>/p2371-2180-devkit-24x7/ extlinux.conf.usb	p2371-2180-devkit-24x7 specific U-Boot

	config file for booting off USB flash storage device.
<platform|ver>/cboot.bin	CPU binary to load the kernel. It also supports Fastboot, charging, and display.
<platform|ver>/LICENSE.cboot	LICENSE file for the cboot.bin binary.
<platform|ver>/nvtboot.bin	Tegra boot specific boot loader binary (AVP bootloader, microboot, miniloader).
<platform|ver>/warmboot.bin	Warm boot binary.
bpmp.bin	Boot and power management firmware.
exec-uboot.sh	Shell script used to load U-Boot into RAM and execute.
gen-tboot-img.py	Script used by the bootlaoder to add an nvtboot-specific header during the flash process.
l4t_initrd.img	L4T initrd image based on minimal Ubuntu environment.
LICENSE	Tegra software license.
LICENSE.bpmp_and_tos-img	License file for bpmp.bin and tos.img.
LICENSE.mkbctpart	License file for mkbctpart.
LICENSE.mkbootimg_and_mkubootscript	License for the `mkbootimg` and `mkbootscript` tools.
LICENSE.mkgpt	License file for the `mkgpt` tool.
LICENSE.mksparse	License file for the `mksparse` tool.
LICENSE.u-boot_and_mkimage	License for `U-Boot` and `mkimage`.
mkbctpart	BCT Partition updating library.
mkbootimg	Tool for img creation.
mkgpt	Tool that encodes both primary and secondary GPT into flashable binary image files.
mkimage	U-Boot tool for `vmlinux.uimg` creation.
mksparse	Sparse image flashing with the boot loader.
mkubootscript	Tool for flashing U-Boot.
nvtboot_cpu.bin	CPU part of Tegraboot for TLK hand over transition.

nvtboot_recovery.bin	AVP bootrom applet binary used by Tegraflash
nvtboot_recovery_cpu.bin	CPU part of Tegraboot used for RCM boot for MODS.
tegrabct	BCT operation helper binary.
tegradevflash	Boot loader device communication library.
tegraflash.py	Script used to flash the board.
tegraflash_internal.py	Helper implementation API script for tegraflash.py.
tegrahost	Boot loader encryption binary.
tegraparser	Parses partition layout, common BCT configuration, fuse bypass configuration and NVIDIA Configuration Table (NCT).
tegrarcm	Bootrom RCM communications binary.
tegrasign	TegraSign creates signature data for PKC operating mode and hash, and encrypted data for SBK operating mode.
tos.img	The monitor binary running in the EL3 exception space on ARMv8 CPUs.

NV Tegra

Nvgstapps TBZ2
Config TBZ2
NVIDIA Drivers TBZ2

The `nv_tegra` directory contains the following directories (denoted by a trailing slash /) and files.

Direcotry or Filename	Description
nv_sample_apps/	NVIDIA sample applications.
nv_sample_apps/LICENSE.gst-openmax	License for the `libgstomx.so`, `libgstnvegl-1.0.so.0`, and `libnvgstjpeg.so` libraries included in `nvgstapps.tbz2`.
nv_sample_apps/LICENSE.gstvideocuda	License for Gstreamer 1.0 CUDA post-processing plugin library.
nv_sample_apps/nvgstapps.tbz2	NVIDIA gstreamer components and applications. For details, see Nvgstapps TBZ2 table (below).

nv_sample_apps/nvgstcapture-<version>_README.txt	Read Me for NVIDIA Gstreamer-based camera capture application (nvgstcapture).
nv_sample_apps/nvgstplayer-<version>_README.txt	Read Me for NVIDIA Gstreamer-based multimedia player (nvgstplayer).
config.tbz2	Configuration files specific to the sample filesystem. For details, see Config TBZ2 table (below).
LICENSE	Tegra software license.
LICENSE.brcm_patchram_plus	License for brcm_patchram_plus.
LICENSE.libargus	License for Argus.
LICENSE.libglvnd	LICENSE file for libglvnd.
LICENSE.libnvcam_imageencoder	License for image encoder.
LICENSE.libscf	License for core camera driver.
nvidia_drivers.tbz2	NVIDIA driver components.
nv_tools.tbz2	The `tegrastats` application, a script for calculations for loads, frequencies, RAM sizes, using existing sysfs nodes. Refer to the *Development Guide* for usage.

Nvgstapps TBZ2

The following table lists the directories (denoted by a trailing slash /) and files available upon decompressing the `nvgstapps.tbz2` archive, located at:

```
nv_tegra/nv_sample_apps/nvgstapps.tbz2
```

Filename	Description
usr/bin/gst-install	Script to build gstreamer from sources. Version can be specified with the --version option (1.6.0 is the default).
usr/bin/nvgstcapture-<version>	Multimedia capture camera application. (Version 0.10 included in hardfp only.)
usr/bin/nvgstplayer-<version>	Multimedia video player application. (Version 0.10 included in hardfp only.)
usr/lib/<ABI_directory>/gstreamer-<version>/	Plug-ins and drivers for gstreamer. (Version 0.10 included in hardfp only.)
usr/lib/<ABI_directory>/gstreamer-<version>/libgstnvcamera.so	gst-plugin library for camera.
usr/lib/<ABI_directory>/gstreamer-1.0/libgstnveglglessink.so	Accelerated Egl based renderer element for gstreamer-1.0.

usr/lib/<ABI_directory>/gstreamer-1.0/ libgstnveglstreamsrc.so	EGLStream Consumer functionality library.
usr/lib/<ABI_directory>/gstreamer-1.0/ libgstnvegltransform.so	NVM buffer conversion to EGLImage plugin library.
usr/lib/<ABI_directory>/gstreamer-1.0/ libgstnvivafilter.so	CUDA post-processing plugin library for gstreamer-1.0.
usr/lib/<ABI_directory>/gstreamer-<version>/libgstnvvidconv.so	NVIDIA proprietary GStreamer conversion plug-in library. (gstreamer 0.10 version included in hardfp only.)
usr/lib/<ABI_directory>/gstreamer-1.0/ libgstnvvideosink.so	GStreamer 1.0 EGLProducer video sink plugin.
usr/lib/<ABI_directory>/gstreamer-<version>/libgstomx.so	OpenMax driver. (gstreamer 0.10 version included in hardfp only.)
usr/lib/<ABI_directory>/gstreamer-1.0/ libgstvideocuda.so	Gstreamer 1.0 CUDA post-processing plugin library.
usr/lib/<ABI_directory>/gstreamer-<version>/libnvgstjpeg.so	Accelerated libjpeg based jpeg decoding and encoding library. (gstreamer 0.10 version included in hardfp only.)
usr/lib/<ABI_directory>/ libgstnvegl-1.0.so.0	Gstreamer EGL API wrapper library.
usr/lib/<ABI_directory>/ libgstnvexifmeta.so	Gstreamer buffer exif metadata library.
usr/lib/<ABI_directory>/ libgstnvivameta.so	Interface library used to add and get gst metadata.
usr/lib/<ABI_directory>/ libsample_process.so	"gst-nvivafilter" sample for cuda post-processing.

Config TBZ2

The following table lists the contents available upon decompressing the `config.tbz2` archive, located at:

```
nv_tegra/config.tbz2
```

Filename	Description
etc/init/nv.conf	NVIDIA-specific initialization script.
etc/init/nvcamera-daemon.conf	nvcamera-daemon service configuration launcher.
etc/init/nvfb.conf	NVIDIA specific first-boot script.
etc/init/nvwifibt.conf	NVIDIA bluetooth/wifi init script.
etc/init/ttyS0.conf	Initialization script for getty on ttyS0.

etc/modprobe.d	Configuration directory/file for modprobe.
etc/modprobe.d/bcmdhd.conf	NVIDIA specific modprobe configuration file for bcmdhd driver loading.
etc/nv/nvfirstboot	Control file used for for first boot.
etc/pulse/daemon.conf	Configuration file for the PulseAudio daemon.
etc/pulse/default.pa.hdmi	PulseAudio configuration file.
etc/pulse/default.pa.orig	PulseAudio configuration file.
etc/sysctl.d/90-tegra-settings.conf	Control file for sysrq.
etc/udev/rules.d/90-alsa-asound-tegra.rules	Rules configuration for proper asound.conf selection.
etc/udev/rules.d/91-xorg-conf-tegra.rules	Rules configuration for proper xorg.conf selection.
etc/udev/rules.d/92-hdmi-audio-tegra.rules	Rules configuration for proper /etc/pulse/default.pa selection.
etc/udev/rules.d/99-nv-wifibt.rules	Rules configuration for Wi-Fi and Bluetooth.
etc/udev/rules.d/99-tegra-devices.rules	Permission setting for Tegra devices.
etc/udev/rules.d/99-tegra-mmc-ra.rules	SD card read_ahead_kb configuration.
etc/X11/xorg.conf	Configuration file for xorg.
etc/X11/xorg.conf.jetson_e	Configuration file for <board_and_rev>-specific xorg.
etc/amixer_settings	Audio configuration.
etc/asound.conf.hdmi	ALSA sound configuration for HDMI audio.
etc/asound.conf.tegrasnd<platform|ver>	ALSA sound configuration for onboard audio.
etc/enctune.conf	Default multimedia encoding parameters for NVIDIA reference platforms.
etc/modules	Lists bluedroid as a supporting module for Bluetooth.
etc/wpa_supplicant.conf	Sample WPA supplicant.

NVIDIA Drivers TBZ2

The following table lists the contents available upon decompressing the `nvidia_drivers.tbz2` archive, located at:

```
nv_tegra/nvidia_drivers.tbz2
```

Filename	Description
etc/ld.so.conf.d/nvidia-tegra.conf	Ldconf file for tegra directories.
etc/nv_tegra_release	Tegra driver versioning file.
lib/firmware/brcm/	BRCM firmware directory.
lib/firmware/brmc/fw_bcmdhd.bin	Firmware for Jetson-tx1 on-board wifi.
lib/firmware/brcm/nvram.txt	File containing tuning parameters for the jetson-tx1 on-board wifi.
lib/firmware/tegra21x/	Firmware files for Jetson TX1and other Tegra X1 devices.
lib/firmware/tegra21x/acr_ucode.bin	High secure mode PMU code.
lib/firmware/tegra12x/fecs.bin	GPU FECS firmware.
lib/firmware/tegra12x/fecs_sig.bin	Signature of FECS microcode.
lib/firmware/tegra12x/gpccs.bin	GPU GPCCS firmware.
lib/firmware/tegra12x/gpmu_ucode.bin	GPU PMU ucode firmware
lib/firmware/tegra12x/gpmu_ucode_desc.bin	Descriptor data for LS PMU.
lib/firmware/tegra12x/gpmu_ucode_image.bin	Low-secure mode PMU code.
lib/firmware/tegra21x/gpu2cde.bin	GPU shader program used for converting GPU compression metadata to be read by VIC and Display.
lib/firmware/tegra12x/NETB_img.bin	GPU device hardware description.
lib/firmware/tegra21x/nvhost_nvdec020.fw	NVDEC firmware file for video decode.
lib/firmware/tegra21x/nvhost_nvdec020_ns.fw	NVDEC firmware that runs without boot loader.
lib/firmware/tegra21x/nvhost_nvdec020_prod.fw	NVDEC firmware.
lib/firmware/tegra21x/nvhost_nvdec_bl020_prod.fw	NVDEC boot loader firmware.
lib/firmware/tegra21x/nvhost_nvenc050.fw	NVENC firmware file for video decode.
lib/firmware/tegra21x/nvhost_nvjpg010.fw	NVJP firmware file for jpeg encode and decode.

lib/firmware/tegra21x/nvhost_tsec.fw	Firmware for TSEC security engine.
lib/firmware/tegra21x/pmu_bl.bin	Boot loader loading acr_ucode.bin.
lib/firmware/tegra21x/pmu_sig.bin	Signature of gpmu_ucode_image.bin.
lib/firmware/tegra21x/vic04_ucode.bin	VIC hardware-specific ucode control firmware.
lib/firmware/tegra21x_xusb_firmware	USB 3.0 firmware.
lib/firmware/bcm4354.hcd	Bluetooth firmware for the BCM4354 chip.
usr/bin/nvidia-bug-report-tegra.sh	NVIDIA bug reporting script. Run for usage tips.
usr/lib/	-
usr/lib/<ABI_directory>/	-
usr/lib/<ABI_directory>/libv4l/plugins/libv4l2_nvvidconv.so	Gstreamer (nv to raw and raw to nv) conversion plugin.
usr/lib/<ABI_directory>/libv4l/plugins/libv4l2_nvvideocodec.so	Video encode/decode libv4l2 plugin library.
usr/lib/<ABI_directory>/tegra/	-
usr/lib/<ABI_directory>/tegra/libargus.so	Argus camera library.
usr/lib/<ABI_directory>/tegra/libcuda.so.1.1	CUDA library.
usr/lib/<ABI_directory>/tegra/libdrm.so.2	Alternative OSS libdrm library.
usr/lib/<ABI_directory>/tegra/libGL.so.1	GL graphics support library.
usr/lib/<ABI_directory>/tegra/libGLdispatch.so.0	OpenGL dispatching and TLS library.
usr/lib/<ABI_directory>/tegra/libglx.so	GLX extension module for X. Module is used by the X server to provide server-side GLX support.
usr/lib/<ABI_directory>/tegra/libnvapputil.so	Host (x86) shared object for application utilities.
usr/lib/<ABI_directory>/tegra/libnvavp.so	User-space interface to the AVP for audio/video acceleration via the nvavp kernel driver.
usr/lib/<ABI_directory>/tegra/libnvcameratools.so	Supporting library for NVIDIA camera utilities.
usr/lib/<ABI_directory>/tegra/libnvcamerautils.so	Supporting library for NVIDIA camera utilities.

usr/lib/<ABI_directory>/tegra/libnvcam_imageencoder.so	Library encodes camera YUV frames to JPEG using the NVIDIA TVMR architecture.
usr/lib/<ABI_directory>/tegra/libnvcamlog.so	Camera runtime tracing and logging helper library.
usr/lib/<ABI_directory>/tegra/libnvcolorutil.so	NvColor utility library.
usr/lib/<ABI_directory>/tegra/libnvdc.so	DC driver file.
usr/lib/<ABI_directory>/tegra/libnvddk_2d_v2.so	DDK 2D.
usr/lib/<ABI_directory>/tegra/libnvddk_vic.so	DDK VIC.
usr/lib/<ABI_directory>/tegra/libnveglstream_camconsumer.so	Argus consumer library.
usr/lib/<ABI_directory>/tegra/libnveglstreamproducer.so	Libray that implements EGLStream Producer functionality.
usr/lib/<ABI_directory>/tegra/libnvexif.so	Helper library to generate exif header.
usr/lib/<ABI_directory>/tegra/libnvfnet.so	OpenGL image postprocessing helper library.
usr/lib/<ABI_directory>/tegra/libnvfnetstorehdfx.so	Memory management utility library used by libdrm.so.2.
usr/lib/<ABI_directory>/tegra/libnvidia-eglcore.so.<EGL|GL_ver>	EGL core library.
usr/lib/<ABI_directory>/tegra/libnvidia-glcore.so.<EGL|GL_ver>	OpenGL core library. This library is implicitly used by libGL and by libglx, and contains the core accelerated 3D functionality.
usr/lib/<ABI_directory>/tegra/libnvidia-glsi.so.<EGL|GL_ver>	OpenGL System Interaction library.
usr/lib/<ABI_directory>/tegra/libnvidia-rmapi-tegra.so.<EGL|GL_ver>	Utility library that implements common code for using kernel-level graphics drivers on Tegra.
usr/lib/<ABI_directory>/tegra/libnvidia-tls.so.<EGL|GL_ver>	NVIDIA tls libraries.
usr/lib/<ABI_directory>/tegra/libnvjpeg.so	Accelerated libjpeg for Tegra.
usr/lib/<ABI_directory>/tegra/libnvll.so	TBD
usr/lib/<ABI_directory>/tegra/libnvmedia.so	Multimedia programming API to access hardware units like encoder, decoder, and video post-processing on Tegra.

usr/lib/<ABI_directory>/tegra/libnvmm.so	NVIDIA Multimedia Framework.
usr/lib/<ABI_directory>/tegra/libnvmm_contentpipe.so	Content pipe implementation (file source abstraction).
usr/lib/<ABI_directory>/tegra/libnvmmlite.so	NVIDIA Multimedia driver.
usr/lib/<ABI_directory>/tegra/libnvmmlite_image.so	NVIDIA Multimedia image driver.
usr/lib/<ABI_directory>/tegra/libnvmmlite_utils.so	NVIDIA Multimedia utilities.
usr/lib/<ABI_directory>/tegra/libnvmmlite_video.so	NVIDIA Multimedia video driver.
usr/lib/<ABI_directory>/tegra/libnvmm_parser.so	NVIDIA Multimedia parser.
usr/lib/<ABI_directory>/tegra/libnvmm_utils.so	Multimedia Framework utilities.
usr/lib/<ABI_directory>/tegra/libnvodm_imager.so	Tegra development platform ODM adaptation for imager.
usr/lib/<ABI_directory>/tegra/libnvomx.so	OpenMAX IL implementation.
usr/lib/<ABI_directory>/tegra/libnvomxilclient.so	OpenMAX IL client.
usr/lib/<ABI_directory>/tegra/libnvos.so	NVIDIA OS abstraction library.
usr/lib/<ABI_directory>/tegra/libnvparser.so	Parser used for NVIDIA NvMMlite.
usr/lib/<ABI_directory>/tegra/libnvrm.so	Resource Manager kernel interface.
usr/lib/<ABI_directory>/tegra/libnvrm_gpu.so	NVIDIA kernel graphics driver abstraction library.
usr/lib/<ABI_directory>/tegra/libnvrm_graphics.so	Resource Manager (NvRM) graphics host, AVP communication library, and graphics drivers.
usr/lib/<ABI_directory>/tegra/libnvtestresults.so	Test results library.
usr/lib/<ABI_directory>/tegra/tegra/libnvtnr.so	Temporal Noise Reduction (TNR) interface.
usr/lib/<ABI_directory>/tegra/libnvtvmr.so	Multimedia Tegra video mixer/renderer.
usr/lib/<ABI_directory>/tegra/libnvv4l2_utils.so	libv4l2 helper library.
usr/lib/<ABI_directory>/tegra/libnvwinsys.so	Winsys library.

usr/lib/<ABI_directory>/tegra/libOpenGL.so	Provides symbols for OpenGL entry point library.
usr/lib/<ABI_directory>/tegra/libscf.so	Core camera driver.
usr/lib/<ABI_directory>/tegra/libtegrav4l2.so	V4L2 driver for Tegra.
usr/lib/<ABI_directory>/tegra/nvidia_icd.json	Vulkan ICD configuration file.
usr/lib/<ABI_directory>/tegra-egl/	-
usr/lib/<ABI_directory>/tegra-egl/ld.so.conf	Ldconf file for tegra-egl directories.
usr/lib/<ABI_directory>/tegra-egl/libEGL.so.1	OpenGL ES driver file.
usr/lib/<ABI_directory>/tegra-egl/libEGL_nvidia.so.0	OpenGL ES driver file.
usr/lib/<ABI_directory>/tegra-egl/libGLESv1_CM.so.1	OpenGL ES driver file.
usr/lib/<ABI_directory>/tegra-egl/libGLESv2.so.2	OpenGL ES driver file.
usr/lib/xorg/	X Windows System libraries and drivers
usr/lib/xorg/modules/	-
usr/lib/xorg/modules/drivers/	-
usr/lib/xorg/modules/drivers/nvidia_drv.so	Tegra X driver.
usr/lib/xorg/modules/extensions/	-
usr/lib/xorg/modules/extensions/libglx.so	Symbolic link pointing to `/usr/lib/<ABI_directory>/tegra/libglx.so`in the rootfs.
usr/sbin/	-
usr/sbin/brcm_patchram_plus	Utility for loading the broadcom bluetooth firmware.
usr/sbin/nvcamera-daemon	Daemon process for using multiple or simultaneous camera instances on L4T platform using core_scf library.
usr/sbin/nvtunerd	Support for image quality tuning tools.
var/nvidia/	-
var/nvidia/nvcam/	-

var/nvidia/nvcam/apps/	-
var/nvidia/nvcam/apps/README.txt	Nvcam application README.txt file.
var/nvidia/nvcam/input/	-
var/nvidia/nvcam/input/ model_frontal.xml	
var/nvidia/nvcam/input/README.txt	Nvcam input README.txt file.
var/nvidia/nvcam/output/	-
var/nvidia/nvcam/output/README.txt	Nvcam output README.txt file.
var/nvidia/nvcam/settings/	-
var/nvidia/nvcam/settings/README.txt	Nvcam settings README.txt file.

Getting Started

To ensure success with NVIDIA® Tegra® Linux Driver Package (L4T), review this topic before you start developing on targeted NVIDIA® Tegra® devices. L4T software drivers require setup and configuration before use.

This guide describes the following L4T functions:

- Setting up L4T on your host system
- Building the kernel
- Flashing binary images
- Installing and testing multimedia
- Bug reporting programs

Consult your board documentation for guidance on setting up and configuring your reference board.

Reference Board Preparation

When developing systems and application software with L4T, you run and test your code on an actual reference platform, such as the NVIDIA® Jetson™ TX1 developer kit. Your code targets this hardware directly, rather than a software simulator or emulator.

Accordingly, you must acquire and set up your reference board before using L4T. Consult your board documentation for guidance on setting up and configuring your board.

Although the reference board supports a variety of peripheral devices, you can start developing on L4T with a board that has the following:

- One of the storage devices specified in Boot Options in this topic.
- A USB cable to plug into the board recovery port.

Boot Options

Boot L4T on the Jetson TX1 reference board from a root file system (rootfs) on integrated, attached, or network-accessible storage. The boot loader must be loaded from the internal eMMC. Root filesystem options include:

- USB stick (formatted to EXT4)
- USB hard disk (formatted to EXT4)
- SD card (formatted to EXT4)
- Internal eMMC
- Network File System (NFS)

Linux Host System Prerequisites

To use L4T on a Linux host system, the following hardware and software prerequisites must be met:

- Host PC running Linux 14.04 or above.
- A kernel image (Image). L4T contains a kernel image for your use. Alternatively, you can download and rebuild the kernel image from source.
- Boot loader. Flashing on a Tegra X1 series (Jetson TX1) developer board requires a boot loader, which is a combination of NVIDIA T-Boot (nvtboot) and U-Boot.
- NFS if you intend to boot L4T on the reference board from your Linux host system or a network-accessible server.
- A USB cable to plug into the recovery port.

Extracting Tegra Linux Driver Package

Use the following procedures to extract your L4T package.

> **Note**: Commands in the examples assume you extracted the release package in ~/.

To extract Tegra Linux Driver Package

- Extract the package manually by executing the following command:

```
$ sudo tar -vxjf Tegra<t-arch|ver>_Linux_<release_num>.<version_num>_<release_type>.tbz
```

Setting Up Your File System

L4T requires a root file system. You must create one the Linux host system and then copy it to your reference board.

Sample Root File System

L4T comes with a pre-built sample root file system created for the Jetson TX1 developer kit. If you wish to create an Ubuntu sample root file system, see https://wiki.ubuntu.com/ARM/RootfsFromScratch.

Setting Up the Root File System

Before booting the target board, you must configure the root file system (rootfs) to:

- Set up the rootfs
- Copy it to the rootfs on the device

Step 1: Set Up the Root File System

This procedure uses the sample file system provided by NVIDIA as the base. If you wish to use your own file system, set the `LDK_ROOTFS_DIR` environment variable to point to the location of your rootfs and skip the steps for setting the root file system.

To set up the rootfs

1. Download the following file to your home directory:

   ```
   Tegra-Linux-Sample-Root-Filesystem_<release_type>.tbz2
   ```

   This file contains the NVIDIA-provided sample root file system.

2. Extract the compressed file as follows:

   - Navigate to the rootfs directory of the extracted NVIDIA driver package with this command:

     ```
     $ cd <your_L4T_root>/Linux_for_Tegra/rootfs
     ```

     Where <your_L4T_root> is your L4T root directory, which is assumed to be your home directory (`~`).

     For more information, see Extracting Tegra Linux Driver Package in this section.

   - Extract the sample file system to the rootfs directory with this command:

     ```
     $ sudo tar jxpf ../../Tegra-Linux-Sample-Root-Filesystem_<release_type>.tbz2
     ```

3. Run the `apply_binaries.sh` script to copy the NVIDIA user space libraries into the target file system:

   ```
   $ cd ..
   $ sudo ./apply_binaries.sh
   ```

4. If you are using a different rootfs, or if you have already configured your rootfs, apply the NVIDIA user space libraries by setting the `LDK_ROOTFS_DIR` environment variable to point to your rootfs. Then run the script, as shown above, to copy the binaries into your target file system.

5. If the `apply_binaries.sh` script installs the binaries correctly, the last message output from the script is “Success!”.

   You have now completed setting up the root filesystem. Proceed to flash the rootfs onto the target Tegra device.

Step 2: Copy the rootfs to the Device

Use these procedures to copy the file system to the Tegra device.

1. Pick a device to place your rootfs.
   - If you are using the internal EMMC, skip ahead to Flashing the Bootloader and Kernel.
2. If you prefer to use an external storage device for the root filesystem, use the following procedure.

To copy the file system to an external storage device

1. Plug your rootfs device into the host system.
2. If your device is not formatted as Ext4, enter the following command to format it with an Ext4 file system:

```
$ sudo mkfs.ext4 /dev/sd<port><device number>
```

   Where:

   - `<port>` is the port to which your device is mounted.
   - `<device_number>` is the device number of the device attached to the port. You can use the `dmesg` command to determine the port.

3. If needed, mount your device with the following command:

```
$ sudo mount /dev/sdX1 <mntpoint>
```

   Where `<mntpoint>` is the mount point on the host system for your rootfs device.

4. Copy the file system. If `LDK_ROOTFS_DIR` is set, execute these commands:

```
$ cd ${LDK_ROOTFS_DIR}
$ sudo cp -a * <mntpoint> && sync
```

5. If it is not set, copy the rootfs directory that is included in the release by executing the following commands:

```
$ cd <your_L4T_root>/Linux_for_Tegra/rootfs
$ sudo cp -a * <mntpoint> && sync
```

6. After copying the content to the external disk or device, unmount the disk and connect it to the target Tegra device.

Proceed to flashing the device with the instructions provided in Flashing the Boot Loader and Kernel.

Flashing the Boot Loader and Kernel

This section describes the steps to flash and boot the target Tegra device. It also provides usage information for the `flash.sh` helper script.

Flash Procedure

First, flash the board with the boot loader and kernel, and, optionally, flash the rootfs to internal eMMC.

Prerequisites

The following directories must be present:

- bootloader—boot loader plus flashing tools (NvFlash, CFG, BCT, etc.)
- kernel—a kernel `zImage` /`vmlinux.uimg`, DTB files, and kernel modules
- rootfs—the root file system that you download (This directory starts empty and you populate it with the sample file system.)
- nv_tegra—NVIDIA® Tegra® user space binaries and sample applications

Additionally, before running the following commands, you must have the USB cable connected to the recovery port.

To flash the boot loader and kernel

1. Put the target into reset/recovery mode.
   - Power on the carrier board and hold the RECOVERY button.
   - Then press the RESET button.
2. Run the `flash.sh` script that is in the top level directory of this release. The script must be supplied with the target board (`jetson-tx1`) for the root file system:

   ```
   sudo ./flash.sh <platform> <rootdev>
   ```

   Where `<rootdev>` depends on where the root file system will be:

   - If the root file system will be on the Jetson TX1 internal eMMC, execute the script as follows:

     ```
     sudo ./flash.sh jetson-tx1 mmcblk0p1
     ```

   - If the root file system will be on a USB disk, execute the script as follows:

     ```
     sudo ./flash.sh jetson-tx1 sda1
     ```

     Note: If a SATA device is connected, that device enumerates as `sda1`.

   - If the root file system will be on an SD card, execute the script as follows:

     ```
     sudo ./flash.sh jetson-tx1 mmcblk1p1
     ```

   The above examples are for U-Boot. For Fastboot, add the following argument:

```
-L <PATH_TO_FASTBOOT_BIN_FILE>
```

For example:

```
sudo ./flash.sh –L bootloader/<platform>/fastboot.bin <platform> <rootdev>
```

This loads the boot loader and kernel.

Flash Script Usage

Locate the most up-to-date usage information by running `flash.sh –h` (using the `flash.sh` script included in the release). The basic usage is as follows.

```
sudo ./flash.sh [options] <platform> <rootdev>
```

Where you specify the required parameters and one or more of the options shown in the following table.

Parameters	Description	
<platform>	Is the <platform> for your release.	
<rootdev>	Is one of following:	
	mmcblk0p1	Specifies internal eMMC.
	mmcblk1p1	Specifies external SDCARD.
	sda1	Specifies external USB device (such as, USB memory stick or HDD).
	eth0	Specifies nfsroot via external USB Ethernet interface.
Options	**Description**	
-h	Specifies to print this usage information.	
-b <bct_file>	Specifies the NvFlash Boot Configuration Table (BCT) file.	
-c <cfg_file>	Specifies the NvFlash configuration file.	
-d <dtb_file>	Optionally specifies a device tree file to use instead of the default.	
-e <emmc_file>	Specifies the eMMC size of the target device.	
-f <flashapp>	Specifies the path to flash application: nvflash or tegra-rcm.	
-i	Specifies to pass the user kernel command line to the kernel as-is.	
-k <partition id>	Specifies the kernel partition ID to be updated (minimum = 5).	
-n <nfs args>	Specifies the static NFS network assignments: `<Client IP>:<Server IP>:<Gateway IP>:<Netmask>`	
-o <odmdata>	Specifies the ODM data value.	

-p	Total eMMC HW boot partition size.
-r	Specifies to skip building and reuse existing `system.img`.
-s <ubootscript>	Specifies the boot script file for U-Boot.
-C <cmdline>	Specifies the kernel command line. **Warning**: Each option in this kernel command-line gets higher precedence over the same option from fastboot. In case of NFS booting, this script adds NFS booting related arguments if the -i option is omitted.
-F <flasher>	Specifies the flash server, such as `fastboot.bin`.
-I <initrd>	Specifies `initrd` file. Null `initrd` is the default.
-K <kernel>	Specifies the kernel image, such as zImage.
-L <bootloader>	Specifies the full path to the boot loader, such as `fastboot.bin` or `u-boot.bin`.
-P <end_of_PPT_plus_1>	Specifies the sum of the primary GPT start address, the size of PPT, plus 1.
-R <rootfs_dir>	Specifies the sample rootfs directory.
-N <nfsroot>	Specifies the nfsroot, for example: `<my IP addr>:/my/exported/nfs/rootfs`
-S <size>	Specifies the rootfs size in bytes. This is valid only for internal rootdev. KiB, MiB, GiB style shorthand is allowed. For example, 1GiB signifies 1024 * 1024 * 1024 bytes.
-T <ITS_file>	ITS file name. Valid only for u-boot.

Increasing the Internal Memory Partition for the Root File System

The suggested rootfs partition size for the Jetson TX1 platform is 15 gigabytes (GB) and is specified by default in the `<target_board>.conf` file used by the `flash.sh` script.

The “`-S <size-in-bytes>`” argument to `flash.sh` can be used to change the partition size.

To flash for a larger partition

- Execute the following command:

```
$ sudo ./flash.sh -S <size> <platform> <rootdev>
```

Where:

- `<size>` is the desired size for the partition, such as 8589934592 (or 8 GiB) for 8 GB, if you want to decrease the size of the partition.
- `<rootdev>` is the rootfs partition’s internal memory, for example mmcblk0p1.

Determining the Success of a Driver Update

After updating drivers on a target board, verify that the update completed successfully. You can determine the success or failure of a driver update by using the following commands.

To determine the success of a driver update

- Execute the following command on a booted target device:

```
$ sha1sum –c /etc/nv_tegra_release
```

If the driver update succeeded, the output displays the word *OK* after the file name. A typical success message looks like this:

```
/usr/lib/xorg/modules/drivers/nvidia_drv.so: OK
```

The driver update fails if the file is missing. A typical error message looks like this:

```
sha1sum: /usr/lib/xorg/modules/drivers/nvidia_drv.so: No such file or directory
/usr/lib/xorg/modules/drivers/nvidia_drv.so: FAILED open or read
```

The driver update also fails if the new file is not the same as the existing file, producing an error such as:

```
/usr/lib/xorg/modules/drivers/nvidia_drv.so: FAILED
```

Installing Additional Packages

L4T comes with additional NVIDIA packages, including packages for Ubuntu and Google Chrome.

Installing Additional NVIDIA Packages

Additional NVIDIA packages may be posted alongside the release. To make full use of the features in the release, install these additional packages.

Directly after the `apply_binaries` step in Setting Up the Root File System, you can install the package into the configured rootfs.

Installing Additional Ubuntu Packages

Install additional packages from Ubuntu, using the provided sample file system.

Note: L4T is tested with the provided sample file system Ubuntu packages. Periodic Ubuntu package updates from Canonical are not validated.

To receive notifications

1. Locate and edit the following file:

```
/etc/apt/sources.list
```

2. Add the following line:

```
deb http://ports.ubuntu.com/ubuntu-ports <distribution>-updates main universe
```

Where `<distribution>` is the name of the Ubuntu distribution your rootfs is based on. For example, for a rootfs based on the Trusty Tahr distribution of Ubuntu, add the line:

```
deb http://ports.ubuntu.com/ubuntu-ports trusty-updates main universe
```

Prerequisite

You have attached an Ethernet cable to the device through either the Ethernet port (if available) or through the USB Ethernet adapter.

To install more packages

1. Boot the target device.
2. Verify your Ethernet connection.
3. Update the package list by executing:

```
$ sudo apt-get update
```

Note: Ensure that you run `sudo apt-get update`and not `apt-get upgrade`, which upgrades already installed packages. Do not confuse the two commands.

4. Install packages using `apt-get`. For example, to install `wget` execute this command:

```
$ sudo apt-get install wget
```

Configuring NFS Root on the Linux Host

To boot the target device from NFS, you must provide an NFS root mount point on your Linux host machine. Following are the general steps for configuring an NFS root on the Linux host.

Prerequisites

- An Ethernet connection to install packages on the host.
- An Ethernet connection on the target.

To configure NFS root on the Linux host

1. Install the `nfs` components on your host machine:

```
$ sudo apt-get install nfs-common nfs-kernel-server
```

2. The NFS server must know which directories you want to 'export' for clients. This information is specified in the `/etc/exports` file.
   - Modify `/etc/exports` to look somewhat like this:

```
$ /nfsroot *(rw,nohide,insecure,no_subtree_check,async,no_root_squash)
```

   - After adding the entry, restart using the following command:

```
$ sudo /etc/init.d/nfs-kernel-server restart
```

3. Create an /nfsroot directory on your Linux host machine:

```
$ sudo mkdir /nfsroot
```

4. Copy the file system to the nfsroot directory:

```
$ cd ./rootfs
$ sudo cp –a * /nfsroot
```

5. Export the root point:

```
$ sudo exportfs -a
```

Alternatively, you can export or un-export all directories by using the –a and –u flags. The following command un-exports all directories:

```
$ sudo exportfs -au
```

6. (Optional) If the Ubuntu firewall blocks NFS root access, it must be disabled depending upon your configuration. You can do so with the following command:

```
$ sudo ufw disable
```

7. If there are issues performing the NFS boot, to separately verify everything on the 'host' machine is configured properly, you can perform the following step on a booted target board through USB/SD/internal eMMC. It should be possible to mount the host NFS root point on the target device:

```
$ mkdir rootfs
$ sudo mount -v -o nfsvers=3 <IP-ADDR>:/nfsroot rootfs
```

Where <IP-ADDR> is the IP address of the Linux Host machine as taken from the ifconfig command. This proves that the host configuration is correct.

Note: Prior to executing the mount command on the target machine, you must install the nfs-common package using the following command:

```
$ sudo apt-get install nfs-common
```

To boot the target with the NFS root point, see the Flashing the Boot Loader and Kernel topic in this section and be sure to include the –N option for the nfs root point.

Synchronizing the Kernel Sources

You can manually rebuild the kernel used for this package. Internet access is required.

Prerequisites

- You have installed Git. Install Git with the following command:

```
$ sudo apt-get install git-core
```

- Your system has the default Git port 9418 open for outbound connections.

To rebuild the kernel

1. Get the kernel source by running the source_sync.sh script:

```
$ ./source_sync.sh -k
```

When prompted enter a ‘tag’ name, as provided in the release notes.

—Or—

Manually sync the sources, as follows:

```
$ cd <myworkspace>
$ git clone git://nv-tegra.nvidia.com/linux-<lnx_ver>.git kernel_sources
$ cd kernel_sources
$ git checkout <release_tag>
```

You can sync to any Linux tag you like. However, the tag provided in the release notes syncs the sources to the same source revision the release binary was built from. To see a list of the available release tags, use:

```
$ git tag –l tegra-l4t*
```

Building the NVIDIA Kernel

Follow the steps in this procedure to build the NVIDIA kernel.

Prerequisites

- You have downloaded the kernel source code.

To build the Tegra Kernel

1. Export the following environment variables:

```
$ export CROSS_COMPILE=<crossbin>
$ export CROSS32CC=<cross32bin>gcc
$ export TEGRA_KERNEL_OUT=<outdir>
$ export ARCH=arm64
```

Where:

- `<crossbin>` is the prefix applied to form the path to the tool chain for cross compilation targeting arm64, e.g., `gcc`. For a Linaro tool chain, it will look something like:

```
<linaro_install>/aarch64-unknown-linux-gnu/bin/aarch64-unknown-linux-gnu-
```

Note: This example requires GCC 4.8 or above. See Jetson TX1 Toolchains for information on how to obtain the reference toolchains.

- `<cross32bin>` is the prefix applied to form the path to the tool chain for cross compilation targeting arm32, e.g., `gcc`. For a CodeSourcery tool chain, it will look something like:

```
<csinstall>/arm-2009q1-203-arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-
And CROSS32CC would be:
<csinstall>/arm-2009q1-203-arm-none-linux-gnueabi/bin/arm-none-linux-gnueabi-gcc
```

Note: This example requires GCC 4.7 or above.

- `<outdir>` is the desired destination for the compiled kernel.

2. Execute the following commands to create the `.config`:

```
$ cd <myworkspace>/<kernel_source>
$ mkdir $TEGRA_KERNEL_OUT
```

   Where <kernel_source> directory contains the kernel sources.

   For Tegra X1, Jetson TX1, use:

```
$ make O=$TEGRA_KERNEL_OUT tegra21_defconfig
```

   Where `<myworkspace>` is the parent of the Git root.

3. Execute the following commands to build the kernel:

```
$ make O=$TEGRA_KERNEL_OUT zImage
```

4. Execute the following command to create the kernel device tree components:

```
$ make O=$TEGRA_KERNEL_OUT dtbs
```

5. Execute the following commands to build the kernel modules (and optionally install them)

```
$ make O=$TEGRA_KERNEL_OUT modules
$ make O=$TEGRA_KERNEL_OUT modules_install INSTALL_MOD_PATH=<your_destination>
```

6. Copy both the uncompressed (Image) and compressed (zImage) kernel images over the ones present in the 'kernel' directory of the release.
7. Archive the kernel modules created in Step 4 using the `tar` command and the filename that is used for the kernel modules TAR file in the same kernel directory of the release. When both of those TAR files are present, you can follow the instructions provided in this document to flash and load your newly built kernel.

Building External Kernel Modules

The procedures in this section describe how to build an out-of-tree kernel module against kernel headers included in the BSP.

The kernel headers are installed at: /usr/src/linux-headers-<kernel version>

To compile natively on the target system

1. Determine user space architecture with the following command:

```
getconf LONG_BIT
```

   The command returns `64` for AARCH64 and `32` for ARM.

2. In AARCH64 user space, prepare the kernel headers with the following commands:

```
$ cd /usr/src/linux-headers-`uname -r`
$ sudo make modules_prepare
```

3. Build the kernel module.

Building in ARM user space is not recommended because an AARCH64 cross-compile toolchain is required. The procedure below installs an AARCH64 toolchain but also switches the userspace runtime to AARCH64. This can cause problems for other user space binaries installed from the BSP.

For best results, use the AARCH64 user space or cross-compile the module on an x86/x86_64 system.

To compile natively in ARM user space on the target system

1. Install aarch64 toolchains with the following commands:

```
sudo dpkg --add-architecture arm64
sudo apt-get update
sudo apt-get install libc6:arm64 binutils:arm64 cpp- 4.8:arm64 gcc-4.8:arm64
sudo ln -s /usr/bin/aarch64-linux-gnu-gcc-4.8 /usr/bin/gcc
```

2. Confirm gcc availability with the following command:

```
gcc –v
```

3. Confirm the architecture with the following command:

```
uname –m
```

   Expected output is `aarch64`.

4. Prepare the kernel headers with the following commands:

```
cd /usr/src/linux-headers-`uname –r`
export ARCH=arm64
sudo make modules_prepare
```

5. Build the kernel module.

If the makefile for the module uses `uname –m` to determine the architecture, it incorrectly uses `aarch64` instead of `arm64`. The following error is displayed during build:

```
make[1]: *** No rule to make target `/usr/src/linux-headers-3.10.67-g458d45c/arch/aarch
```

To avoid this issue, modify the makefile to set `ARCH` to `arm64`.

The recommended non-target build system configuration is x86/x86_64, Ubuntu 14.04.

To cross-compile on another system

1. Uncompress the following kernel headers to a local directory:

```
 <top>/Linux_for_Tegra/kernel/kernel_headers.tbz2
```

   with the following commands:

```
cd <local src dir>
tar xpf <top>/Linux_for_Tegra/kernel/kernel_headers.tbz2
```

2. Set up the cross-compile toolchain with the following commands:

```
export CROSS_COMPILE=<crossbin>
export ARCH=arm64
```

Where `<crossbin>` completes path to the tool chain for cross compilation targeting `arm64`, e.g., `gcc`. For a Linaro tool chain, the path is similar to the following:

```
<linaro_install>/aarch64-unknown-linux-gnu/bin/aarch64-unknown-linux-gnu-
```

See Jetson-TX1 Toolchains for how to obtain the reference toolchains.

3. Prepare the kernel headers with the following commands:

```
cd <local src dir>/linux-headers-<kernel version>
export ARCH=arm64
sudo make modules_prepare
```

4. Specify `<local_source_directory>/linux-headers-<kernel version>` as the kernel source path, with the following line in the make file:

```
make -C <local src dir>/linux-headers-<kernel version> M=$(PWD) modules
```

5. Build the kernel module.

OpenGL/EGL Gears Test Application

To run a sample OpenGL/EGL test application, you can run the open-source Gears application.

To install and run Gears test application

1. Boot the target system with an Ethernet connection.
2. Enable package download from the "universe" repository by editing /etc/apt/sources.list as root:

```
$ sudo vi /etc/apt/sources.list
```

3. Uncomment the following line in the file by removing the leading # character:

```
# deb http://ports.ubuntu.com/ubuntu-ports/ trusty universe
```

4. Update the repository:

```
$ sudo apt-get update
```

5. Install the `mesa-utils` and `mesa-utils-extra` packages:

```
$ sudo apt-get install -y mesa-utils
$ sudo apt-get install -y mesa-utils-extra
```

6. At this point you should be able to run the application with the following steps:

```
$ export DISPLAY=:0
$ X&
$ /usr/bin/es2gears
```

GStreamer-based Multimedia Playback (NvGstPlayer)

Use the GStreamer open source multimedia framework and the NvGstPlayer utility for testing multimedia local playback and HTTP/RTSP streaming playback use cases. The NvGstPlayer can be used as a reference implementation.

For more information about the NvGstPlayer application, refer to the ReadMe file included with the release.

Installing GStreamer

Install GStreamer from the Internet directly on the target. The wrapper library, `gst-openmax`, is an interface between GStreamer and OpenMAX. It enables accelerated NVIDIA plug-ins in the GStreamer framework.

For more information about GStreamer, see the following website:

```
http://gstreamer.freedesktop.org
```

NvGstPlayer is a multimedia player test application.

Complete prerequisite steps in the file `nvgstcapture_README.txt` before running the NvGstPlayer and NvGstCapture applications.

Instructions for installing GStreamer are also included in that text file.

Using NvGstPlayer

NvGstPlayer is a command line media file player. It plays audio/video files encapsulated in MP4, 3GP, AVI, ASF, WMA, MKV, M2TS, WEBM, and MOV. NvGstPlayer supports local file playback and playback over RSTP, HTTP, and UDP.

For information about NvGstPlayer runtime commands, default settings, and important notes see the `nvgstplayer_README.txt` file included in the release.

Gstreamer-based Camera Capture (NvGstCapture)

The NvGstCapture application supports GStreamer version 0.10.36 by default. NvGstCapture captures audio and video data using a microphone and camera and encapsulate encoded A/V data in the container file.

For NvGstCapture installation and usage information, see the `nvgstcapture-<VERSION>_README.txt` file included with the release at `~Linux_for_Tegra/nv_tegra/nv_sample_apps`.

NVIDIA Bug Reporting Script

For debugging purposes, attach the log file to communicate issues found with the release. Use the `nvidia-bug-report-tegra.sh` script to generate log files.

To generate a log file for bug reporting

- Log into the target board and enter the following command:

```
$ sudo /usr/bin/nvidia-bug-report-tegra.sh
```

To generate a log file for bug reporting with extended logging mode

- Log into the target board and enter the following command:

```
$ sudo /usr/bin/nvidia-bug-report-tegra.sh –e
```

By default, the logfile generated by both these procedures is located at `$HOME/nvidia-bug-report-tegra.log`.

> **Note:** Attach a log file when reporting any bugs to NVIDIA, whether through email or the forums.

Software Features

NVIDIA® Tegra® Linux Driver Package (L4T) supports the following software features, which provide users a complete package to bring up Linux on targeted NVIDIA® Tegra® X1 devices.

This release supports the NVIDIA® Jetson™ TX1 developer kit and module.

Note:	Always check the *Release Notes* for constraints related to these features.

Boot Loaders

Boot Loader	Feature	Notes
nvboot	Boot Device	eMMC
	2nd Stage Load Device	eMMC
U-Boot	Storage Device Support	eMMC (no CQ), SD card, USB (HS)
	Display: Console	UART
	Display: Splash/Menu	UART
	I/O Bus Support	I2C, USB (HS), USB (device)

Toolchain

Feature	Tool Chains	Notes
Aarch64	gcc-4.8.2-glibc-2.17	For 64-bit Kernel, Userspace, and U-Boot

Hardfp	gcc-4.5.3-glibc-2.11.3	For 32-bit Userspace and U-Boot

Kernel

Interface	Feature	Notes
DSI	DSI Display Support	-
	DSI Ganged Mode	-
	PWM Backlight	-
	DC Continuous Mode	-
	DC Driven Command Mode	-
	Host Write	-
	DSI One-Shot Mode	-
	Dual Display	-
	Run Time Power Management	-
HDMI	EDID Support	-
	Hot-Plug Detection Mechanism	-
	HDMI 1.4	480p, 720p, 1080p, RGB 444 4K @ 30 Hz
	Driver Suspend/Resume for Low Power	-
	HDMI as Primary Display	-
	Dual Display	-
	HDMI: 1.4b compliance	Pending certification
	HDMI: 2.0 compliance	Pending certification
	Audio Support	-
Ethernet	10/100/1000 BASE	-
	MAC Filtering	-
PWM	Speed Control from sysfs	-
	Control from Temperature Variation	-
I2C	Master Mode	-

Wifi	802.11a/b/g/n/ac	BCM4354
Bluetooth	Bluetooth 4.0	BCM4354
Camera support (CSI input support)	V4L2 Media-Controller (V4L2 API bypasses ISP)	CSI0, CSI1, CSI2, CSI3, CSI4, CSI5 Note: The media-controller driver model is adopted in the 24.1 release. the Soc_camera driver is provided, but deprecated.
Peripheral devices	INA support	Current monitoring for: CPU/GPU/VDD_IN
Platform support	Baseboard: P2597 Jetson module: P2180	
Wifi	Multi-Region support	Region Support: # U.S. # Taiwan # Europe # Japan # Korea # Canada # Isreal # Default (lowest-common-denominator)

I/O

I/O Type	Feature	Notes
SPI	Max Bus Speed	SPI4: 65 MHz
		SPI1: 65 MHz
		SPI2: 65 MHz
	Chip Select	SPI4: 0
		SPI1: 0/1
		SPI2: 0/1
	Packed/Unpacked	SPI4, SPI1, SPI2
	Full Duplex Mode	SPI4, SPI1, SPI2
	Both Enable Bit	SPI4, SPI1, SPI2
	Both Enable Byte	SPI4, SPI1, SPI2

	Bi-directional	SPI4, SPI1, SPI2
	Least Significant Bit	SPI4, SPI1, SPI2
	Least Significant Byte First	SPI4, SPI1, SPI2
	Software or Hardware Chip Select Polarity Section	SPI4, SPI1, SPI2
	Supported Modes 1/2/3/4	SPI4, SPI1, SPI2
	Purpose/Client	SPI4: Touch
		SPI1: Audio
		SPI2: Cam/Display
SDMMC	I/O Speeds (Clock speed)	SDMMC1: 204 MHz
		SDMMC4: 200 MHz
		SDMMC (M.2/SDIO): 204 MHz
	Hot Plug Support	SDMMC1
	SD High Speed Mode	SDMMC1, SDMMC (M.2/SDIO)
	SDR50	SDMMC1, SDMMC4, SDMMC (M.2/SDIO)
	SDR104	SDMMC1, SDMMC (M.2/SDIO)
	HS533	SDMMC4
	HS400	SDMMC4
	HS200	SDMMC4
	DDR Mode	SDMMC1, SDMMC4, SDMMC (M.2/SDIO)
	Voltage Switching	SDMMC1, SDMMC (M.2/SDIO)
	Frequency Tuning	SDMMC1, SDMMC4, SDMMC (M.2/SDIO)
	Packed Commands	SDMMC4, SDMMC (M.2/SDIO)
	Cache Control	SDMMC4
	Discard	SDMMC4
	Sanitize	SDMMC4
	RPMB	SDMMC4
	HPI	SDMMC4

	BKOPS	SDMMC4
	Power Off Notification	SDMMC4
	Sleep	SDMMC4
	Field Firmware Upgrade	SDMMC4
	CMD Queuing	-
	Device Life Estimation Type A	SDMMC4
	Device Life Estimation Type B	SDMMC4
	PRE EOL Information	SDMMC4
	Power Management	SDMMC1, SDMMC4, SDMMC (M.2/SDIO)
SATA	Speed	GEN1
		GEN2
	AHCI Mode	1.3.1
	SATA Specification	3.1
	HIPM	-
	DIPM	-
	NCQ	-
	Port Multiplier Support	CBS
	Link Power Management States	Partial
		Slumber
	Device Power Management States	D0
		D1
		D2
	Runtime Time Power Management	-
	S.M.A.R.T	-
	ATA Error Logging	-
I2C	Master	I2C GEN1, I2C GEN2, I2C GEN3, I2C DDC, I2C PWR, I2C6
		Speeds 400 kHz (FM)
		10-bit addressing
		Lost arbitration detect

		Packet mode
		7-bit
		DMA mode
		Bus clear support
USB 2.0	Device Mode	USB0
	OTG Mode	USB0
	Host Mode	USB0, USB1
	Host - Low Speed Devices	USB0
	Host - Full Speed Devices	USB0
	Host - High Speed Devices	USB0, USB1
	Host - Auto Suspend Support	USB0
USB 3.0	Speeds	USB0: HS/480 Mbps
		USB1: SS/5 Gbps
	Lanes	USB1: pex5
	USB 3.0 Support	USB1
	Connector	USB0: Micro AB
		USB1: TYPE A
	USB 2.0 Support	USB0, USB1
	Remote Wakeup Support	USB0: USB 2.0
		USB1: USB 2.0/3.0
	Host - Auto Suspend Support	USB0, USB1
	OTG Support	USB0
	Class Support	Mass storage (USB0, USB1)
		USB video class (USB0, USB1)
		HID (USB0, USB1)
		USB audio class (USB0, USB1)
		MTP (USB0, USB1)
		CDC - NCM/ECM (USB0, USB1)
GPIO	Pinmux Configuration	-

	GPIO Configuration And Programming	-
	GPIO Interrupt Support	-
UART	Speed	UART0: 115200
		UART2: 921600
		UART3: 3000000
	Hardware Flow Control	UART2, UART3
	PIO Mode	UART0, UART2, UART3
	DMA Mode	UART0, UART2, UART3
	FIFO Mode	UART0, UART2, UART3
PCIe	Speed	PCIe 0: Gen1/Gen2
		PCIe 1: Gen1/Gen2
	Lane Width	PCIe 0: x1
		PCIe 1: x1, x2, x4
	Host Controller Features	Lanes Xbar config (X4_X1, X2_X1)
		Extended Config Space
		Hardware Clock Gating
		Deep Power Down (DPD)
	PCI Features	Message Signaled Interrupts
		Vendor Specific Messages
		PCI Express
		MSI-X
	PCIe Device Capabilities	Max Payload
		Extended Tag Field Support
		Role-Based Error Reporting
		Maximum Link Speed; Supports Up to Gen2 Speeds
		Maximum Link Width; Supports Up to X4 Link Width
		ASPM Support (L0s and L1)

		L1 Clock Power Management
		Data Link Layer Link Active Reporting Capable
		Link Bandwidth Notification Capability
	Link Control	Read Completion Boundary
	Root Control	System Error on Correctable Error
		System Error on Non-Fatal Error
		System Error on Fatal Error
		PME Interrupt Enable
	Extended Capabilities	Advanced Error Reporting (AER)
		Latency Tolerance Reporting (LTR)
	L1 PM Substates	L1.1
		L1.2
	Misc Features	Dynamic Voltage Frequency (DVPS)
		Tegra Low Power Mode (LP0)
		Runtime PM
JTAG	JTAG Attach	-
	JTAG Halt/Step/Go	-

CUDA

Feature	Version
CUDA	Version 7.0.74 with FP16 support

Graphics

Graphics APIs	Notes
OpenGL	4.5

OpenGL-ES	3.1
EGL	1.4
API Support	**Notes**
GL + GLX	-
GL + EGL	-
GL-ES + EGL	-
X11 ABI	Through version 19
Display API	Direct Rendering Manager: Compatibility with DRM 2.0
Vulkan	Version 1
64-bit support	Kernel and Userspace

EGL and OpenGL ES Support

EGL is an interface between Khronos rendering APIs, such as OpenGL ES, and the underlying native platform window system. It handles graphics context management, surface/buffer binding, and rendering synchronization. EGL enables high-performance, accelerated, mixed-mode 2D and 3D rendering using other Khronos APIs.

L4T supports the EGL 1.4 specification, Khronos Native Platform Graphics Interface (EGL 1.4 Specification).

The OpenGL ES driver in this release supports the following OpenGL ES specifications:

- OpenGL ES Common Profile Specification 2.0
- OpenGL 4.5

For more information on OpenGL ES, see the Khronos OpenGL ES API Registry.

Video Decoders

Video Decode	Output Formats	Sampling Frequency and Bit rate/Frame rate	Notes
H.264	NV12, NVMM:NV12	3840 x 2160 at 60 fps Up to 120 Mbps	Full-frame, Disable-DPB, Skip-Frames
H.265	NV12, NVMM:NV12	3840 x 2160 at 60 fps Up to 160 Mbps	Decode Support in Gstreamer 1.4.5 and later
JPEG	I420, NVMM:I420	600 MP/sec	-

VP8	NV12, NVMM:NV12	3840 x 2160 at 60 fps Up to 140 Mbps	-
VP9	NV12, NVMM:NV12	3840 x 2160 at 60 fps Up to 120 Mbps	-

Video Encoders

Video Encode	Input Formats	Sampling Frequency and Bit rate/Frame rate	Notes
H.264	I420, NV12, NVMM:1420, NVMM:NV12	3840 x 2160 at 30 fps Up to 120 Mbps	RC-Mode, Bitrate, Iframeinterval, Quality-Level, Low-Latency, Sliceintrarefreshinterval, Bit-Packetization, VBV-Size, Insert-SPS-PPS, No-B-Frames, Slice-Header-Spacing, Profile, Force-IDR
JPEG	I420, NVMM:I420	600 MP/sec	-
H.265	I420, NVMM:I420, NVMM:NV12	3840 x 2160 at 30 fps Up to 100 Mbps	-
VP8	I420, NV12, NVMM:I420, NVMM:NV12	3840 x 2160 at 30 fps Up to 120 Mbps	RC-Mode, Bitrate, Iframeinterval, Quality-Level

Display Outputs

nveglglessink	nvxvimagesink	nvoverlaysink	nvhdmioverlaysink
X11 Window	X11 Window	Panel Overlay	HDMI Overlay
-	-	Overlay	Overlay
-	-	Overlay-Depth	Overlay-Depth
-	-	Overlay-X	Overlay-X
-	-	Overlay-Y	Overlay-Y
-	-	Overlay-W	Overlay-W

-	-	Overlay-H	Overlay-H

Conversion, Scaling, and Rotation Formats

Input Formats	Output Formats	Notes
I420	I420	Flip-Method
UYVY	UYVY	Flip-Method
NV12	NV12	Flip-Method
GRAY8	GRAY8	Flip-Method
NVMM:I420	NVMM:I420	Flip-Method
NVMM:NV12	NVMM:NV12	Flip-Method

CSI and USB Camera Formats

Output Format	Options	Notes
NVMM:I420	Scene-Mode	-
	Color-Effect	-
	Auto-Exposure	-
	Flicker	-
	Contrast	-
	Saturation	-
	TNR-Strength	-
	TNR-Mode	-
	Edge-Enhancement	-
	Intent	Still, Video, Video snapshot, Preview
	Sensor-ID	-
	Enable-EXIF	-

	aeRegion	-
	wbRegion	-
	fpsRange	-
	Exposure-Time	-
	wbManualMode	-
	wbGains	-
	Embedded Metadata	Precision timestamping, DCT-NR, V4L2 interface for sensor driver, Gyro service for L4T for VSTAB and AF
	ARGUS	-
	RAW capture	-
	EGL producer	-
	Face detection	-
	HDFX	-
	Simultaneous Multi-Camera	Pluggable/replacable 3A, 12- and 14-bit sensors, DPCM sensors
	VSTAB support	AF2.8 support, Auto Iris
	Image De-Warping and Distortion Correction	Global Shutter
	Coordinated Multi-Camera Support	-

U-Boot Guide

U-Boot is the default boot loader for NVIDIA® Tegra® Linux Driver Package (L4T). If you used an earlier release of L4T, check that your environment is fully updated for the new boot loader before compiling and flashing the boot loader and the kernel.

Requirements

The software requirements and prerequisites required for Tegra Linux Driver Package (L4T) include:

- Linux-based Host System

  Functionality of the U-Boot build and flashing utilities was validated using an Ubuntu 12.04 host system. Later versions of Ubuntu or alternative Linux distributions may work with host-specific modifications.

- Tegra Linux Driver Package (L4T)

  Download the latest L4T package from the Tegra Developer Zone and follow the installation instructions in the user documentation. You can find L4T on the Tegra Developer Zone:

  ```
  http://developer.nvidia.com/linux-tegra
  ```

- Device Tree Compiler (dtc)

  The Device Tree Compiler (dtc) is used to compile device tree files contained in the U-Boot source tree. Many of the dtc packages available from standard Linux distribution package management systems (like apt) are not yet updated with a version of dtc supporting the features required by the U-Boot makefile. Therefore, an example of building dtc from source is included in this chapter. For the procedure, see Building Device Tree Compiler.

  A pre-built dtc binary is also included in the kernel directory of the release. This binary is built from the kernel sources in this release. The sources are located in the `scripts/dtc` directory. You build dtc by building the kernel `dtbs` target.

- Download the Linaro Aarch64 tool chain
- U-Boot source.

For more information, see Downloading and Building U-Boot in this chapter.

- Kernel source

  For information, see the following sections in the Getting Started chapter:

  - Setting up the Root File System
  - Synchronizing the Kernel Sources
  - Building the NVIDIA Kernel

  Also, see Adding a Compiled Kernel to the Root File System in this chapter.

Downloading and Building U-Boot

Before flashing U-Boot to your reference platform, you must download and build it on your Linux host system. NVIDIA offers a Git repository containing the source code for a U-Boot build suitable for L4T.

Prerequisite

Before copying U-Boot, back up all of the original U-Boot files in:

```
<top>/Linux_for_Tegra/bootloader/<platform>/<board>/
```

Where:

- `<platform|ver>` is the Tegra hardware platform, such as `t210ref`.
- <board> is the individual board, such as `p2371-2180`.

To download and build U-Boot

1. Download the L4T U-Boot source code by executing the following commands:

   ```
   $ git clone -n git://nv-tegra.nvidia.com/3rdparty/u-boot.git
   ```

   Alternatively, you can use the `source_sync.sh` script in the L4T release.

   If you run `source_sync.sh –u` without parameters, the script prompts for the `<TAG_NAME>`, which is provided in the *Release Notes*.

   The "-k" option to `source_sync` syncs the kernel sources. A space between the `–u` and `–k` options is allowed. By default, if no option is provided, the script syncs the kernel and u-boot sources.

   Also, you can run the script by passing the `<TAG_NAME>` as follows:

   ```
   $ cd <your_L4T_root>/Linux_for_Tegra
   $ ./source_sync.sh –u <TAG_NAME>
   ```

   This syncs the source to:

   ```
   <source_sync.sh_location>/sources/u-boot_source
   ```

The <uboot_src_dir> directory becomes:

```
<your_L4T_root>/Linux_for_Tegra/sources/u-boot_source
```

Note:	Further instructions assume your current working directory is the U-Boot source tree.

2. Check out the Git tag name:

```
$ git checkout -b <branch_name> <tag_name>
```

Where:

- <branch_name> is the name of your local branch.
- <tag_name> is the release tag name provided in the *Release Notes*.

3. Set the build environment:

```
$ export ARCH=arm
$ export CROSS_COMPILE=<your_toolchain_location>
```

4. Build U-Boot by executing:

```
$ make distclean
$ make <target_board>_defconfig
$ make
```

Flashing U-Boot

You must flash U-Boot to internal eMMC only. At boot time, U-Boot fetches the boot configuration file, kernel and device tree, all of which may reside on one of the following storage devices used for boot:

- Internal eMMC
- An SD card
- A USB storage device
- A TFTP/NFS server

When executing the script that flashes U-Boot, you must specify a command-line option to specify the storage device containing the root filesystem, so that the appropriate boot configuration file is selected. The boot configuration file contains kernel command line parameters that control where the Linux kernel looks for the root filesystem. The following sections describe the script command for each configuration.

To flash U-Boot and mount the root filesystem from internal eMMC

- Use the following command to select a boot configuration file that causes the kernel to mount the root filesystem from internal eMMC:

```
$ sudo ./flash.sh <target_board> mmcblk0p1
```

Note:	Check that your environment is fully updated for this change in boot loader before compiling and flashing the boot loader and the kernel.

To flash U-Boot and mount the root filesystem from an SD card

- Use the following command to select a boot configuration file that causes the kernel to mount the root filesystem from an SD card:

```
$ sudo ./flash.sh <target_board> mmcblk1p1
```

To flash U-Boot and mount the root filesystem from a USB storage device

- Use the following command to select a boot configuration file that causes the kernel to mount the root filesystem from a USB storage device, such as a Pen Drive.

```
$ sudo ./flash.sh <target_board> sda1
```

Note:	**The U-Boot boot loader detects USB external storage. The kernel detects both USB external storage and external SATA storage.**

Use one USB or SATA external storage device at a time. If using more than one external device, a random device may be chosen as the root device.

To flash U-Boot and mount the root filesystem from an IP network

- Use the following command to select a boot configuration file that causes the kernel to mount the root filesystem from a TFTP/NFS server:

```
$ sudo ./flash.sh -N <IPA>:/<nfs directory> [-n <target IPA>:<host IPA>:<gateway IPA>:<
```

Where:

- `<interface name>` is eth0 for RJ45 connector and eth1 for a USB Ethernet dongle.
- `<IPA>` is the NFS server hosting the root filesystem.
- `<nfs_directory>` is the full path name of exported root filesystem.
- `<target IPA>` is the static IP address for the target device.
- `<host IPA>` is the static IP address for the NFS server.
- `<gateway IPA>` is the static IP address for the gateway.
- `<netmask>` is the static netmask for the local network.

Note: The `-n` option is recommended on point-to-point network connections where no DHCP server is configured.

Flashing Just U-Boot

To flash the full L4T image to the reference platform see Flashing U-Boot in this chapter.

To flash U-Boot, proceed as follows.

To copy U-Boot for flashing to the reference platform

- Execute the following on your Linux host system:

```
$ cp <uboot_src_dir>/u-boot{,.bin,.dtb,-dtb.bin} \
```

```
<your_L4T_root>/Linux_for_Tegra/bootloader/<platform>/<board>
```

To flash just new U-Boot

- Execute the following:

```
$ sudo ./flash.sh -k EBT <target_board> mmcblk0p1
```

Changing the eMMC Partition Layout

The following information is based on eMMC hardware and software layout information in the following files:

- `<target_board>.conf`
- `<top>/Linux_for_Tegra/bootloader/<platform>/cfg/gnu_linux_tegraboot_emmc_full.xml`

  Where `<top>` is the L4T root directory.

  Note: L4T U-Boot does not use the kernel partition. The kernel is installed into the filesystem alongside the boot configuration file. Aside from this difference, U-Boot has the same internal eMMC partition layout as that used by cboot.

eMMC IC Parameter

The eMMC IC parameter is defined by 2 variables in the `<target_board>.conf` file. They limit the size of the total usable data area and determine the location of GPT partitions.

- BOOTPARTSIZE: specifies the eMMC boot partition size (boot0 partition size + boot1 partition size)
- EMMCSIZE: specifies the eMMC usable data size (BOOTPARTSIZE + user partition size)

  Note: boot0, boot1, and user partition size can be obtained from the eMMC device datasheet.

Root Filesystem Size

The root filesystem partition is the largest of the partitions, and its size is one of the key factors in partition layout determination. By default, `flash.sh` sets the root filesystem size to 14 GB.

To modify the root filesystem partition size

- Modify the value of the `ROOTFSSIZE` variable in the `<target_board>.conf` file.

Note: The total space used by all partitions cannot exceed `EMMCSIZE`.

GPT Partitions

The `flash.sh` script creates the primary and secondary GPT partitions automatically, based on the internal eMMC partition layout.

- The Protective MBR contains device information to prevent traditional boot loaders from performing destructive actions. It is located at LBA 0.
- The primary GPT partition contains the GUID Partition Table. It is located at LBA 1.
- The secondary GPT partition contains the same information as the primary GPT and serves as the backup. It is located at the last LBA of the boot device.

- The last Logical Block Address (LBA) varies from device to device. Both U-Boot and the kernel are able to obtain the last LBA.

LNX Partition

The LNX partition is not used by U-Boot; however, for compatibility, an empty LNX partition is allocated.

APP Partition

If root filesystem storage is in eMMC, the root filesystem is flashed to this partition. U-Boot expects the boot configuration file, kernel, and device tree files to exist in the `<rootfs>/boot` directory; consequently, `flash.sh` flashes the following kernel files in the APP partition:

- kernel (`Image`)
- device_tree_blob (`tegra210-jetson-cv-base-p2597-2180-a00.dtb`)
- boot configuration file (`extlinux/extlinux.conf`)

Note: The `flash.sh` script treats the root filesystem-on-IP-network configuration as a special case and also flashes these kernel files in the `<APP partition>:/boot` directory.

Example Full Internal eMMC Partition Layout

An eMMC layout configuration file has the following contents. The actual configuration file is named `gnu_linux_tegraboot_emmc_full.xml`.

Note: Under default settings, U-Boot does not use the kernel partition (LNX).

```
<?xml version="1.0"?>

<!-- Nvidia Tegra Partition Layout Version 1.0.0 -->

<partition_layout version="01.00.0000">
    <device type="sdmmc" instance="3">
        <partition name="BCT" id="2" type="boot_config_table">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4194304 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>

        <partition name="NXC" id="3" type="NVCTYPE">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4194304 </size>
```

```
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> NVCFILE </filename>
    </partition>

    <partition name="PPT" id="4" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> PPTSIZE </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
    </partition>

    <partition name="GP1" id="5" type="GP1">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 2097152 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
    </partition>

    <partition name="APP" id="6" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> APPSIZE </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> APPFILE </filename>
    </partition>

    <partition name="TXC" id="7" type="TBCTYPE">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
```

```
            <size> 2097152 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> TBCFILE </filename>
        </partition>

        <partition name="EBT" id="8" type="bootloader">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4194304 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> EBTFILE </filename>
        </partition>

        <partition name="BXF" id="9" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 2097152 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> BPFFILE </filename>
        </partition>

        <partition name="WX0" id="10" type="WB0TYPE">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 6291456 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> WB0FILE </filename>
        </partition>
```

```
<partition name="RP1" id="11" type="data">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 4194304 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <allocation_attribute> 0x8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> DTBFILE </filename>
</partition>


<partition name="TXS" id="12" type="data">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 6291456 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <partition_attribute> 0 </partition_attribute>
    <allocation_attribute> 8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> TOSFILE </filename>
</partition>

<partition name="EXS" id="13" type="data">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
    <size> 2097152 </size>
    <file_system_attribute> 0 </file_system_attribute>
    <partition_attribute> 0 </partition_attribute>
    <allocation_attribute> 8 </allocation_attribute>
    <percent_reserved> 0 </percent_reserved>
    <filename> EKSFILE </filename>
</partition>

<partition name="FX" id="14" type="FBTYPE">
    <allocation_policy> sequential </allocation_policy>
    <filesystem_type> basic </filesystem_type>
```

```
        <size> 2097152 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> FBFILE </filename>
    </partition>

    <partition name="SOS" id="15" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 20971520 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <filename> SOSFILE </filename>
    </partition>

    <partition name="EXI" id="16" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> EFISIZE </size>
        <file_system_attribute> 0 </file_system_attribute>
        <partition_attribute> 0 </partition_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> EFIFILE </filename>
    </partition>

    <partition name="LNX" id="17" type="data">
        <allocation_policy> sequential </allocation_policy>
        <filesystem_type> basic </filesystem_type>
        <size> 67108864 </size>
        <file_system_attribute> 0 </file_system_attribute>
        <allocation_attribute> 0x8 </allocation_attribute>
        <percent_reserved> 0 </percent_reserved>
        <filename> LNXFILE </filename>
    </partition>
```

```
        <partition name="DXB" id="18" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 4194304 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> DTBFILE </filename>
        </partition>

        <partition name="NXT" id="19" type="NCTTYPE">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 2097152 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> NCTFILE </filename>
        </partition>

        <partition name="MXB" id="20" type="MPBTYPE">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 6291456 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> MPBFILE </filename>
        </partition>

        <partition name="MXP" id="21" type="MBPTYPE">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 6291456 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <partition_attribute> 0 </partition_attribute>
```

```
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
            <filename> MBPFILE </filename>
        </partition>

        <partition name="USP" id="22" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 2097152 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>

        <partition name="UDA" id="23" type="data">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 2097152 </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 0x808 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>

        <partition name="GPT" id="24" type="GPT">
            <allocation_policy> sequential </allocation_policy>
            <filesystem_type> basic </filesystem_type>
            <size> 0xFFFFFFFFFFFFFFFF </size>
            <file_system_attribute> 0 </file_system_attribute>
            <allocation_attribute> 8 </allocation_attribute>
            <percent_reserved> 0 </percent_reserved>
        </partition>
    </device>
</partition_layout>
```

Testing Root Filesystem By Device

You should test the root file system location by device. A Y in the output indicates that correct U-Boot initialization and hand-off to the kernel occurred.

Root filesystem Location	Jetson TX1
mmcblk0p1	Y
mmcblk1p1	Y
sda1	Y
eth0	Y
eth1	Y

Building the Device Tree Compiler

Build the Device Tree Compiler (dtc) from source code included in L4T, specifying the features required by the U-Boot makefile.

> **Note**: In the procedure below, if you do not want to pass in dtc as a parameter to the U-Boot environment, ensure a local command path (such as `/usr/local/bin` or another choice) is at the beginning of the shell command path. Furthermore, if you execute (in the last step):
>
> `$ make install`
>
> the dtc makefile installs the binary into the first entry of shell PATH variable. Therefore, it is important that the local command path is at the beginning of the shell PATH variable.

To build DTC from source

1. Download dtc source code by executing the following `git clone` command:

   ```
   $ git clone git://git.kernel.org/pub/scm/utils/dtc/dtc.git
   $ cd dtc
   ```

   Further instructions will assume your current working directory is the dtc source tree.

2. Build dtc by executing:

   ```
   $ make
   ```

3. Install dtc.

To install into the default directory (`$HOME`), execute:

```
$ make install
```

To install to a specific directory, execute:

```
$ make install PREFIX=/usr/local
```

In either case, `bin`, `lib`, and `include` sub-directories will be created below the installation directory.

Adding a Compiled Kernel to the Root File System

Adding a new Kernel

U-Boot requires a kernel image on the root filesystem. First you must configure the file system for U-Boot. Then you add the kernel image to the root filesystem.

Prerequisite

- You have compiled the kernel as described in Getting Started in this guide.

To configure a file system for U-Boot

1. Use the `apply_binaries` script to copy the Image file in the kernel directory into the root filesystem directory in the `/boot` folder.
2. Install the root filesystem directory onto your device.

   For U-Boot to function properly, there must be Image and `dtb` files in the `/boot` directory of the target file system.

   For more information on installing the root filesystem directory onto your device, see Setting Up the Root File System in the Getting Started chapter.
3. If you have already installed your root filesystem onto a device, manually copy the Image file and `dtb` files to the installed root file system.

To configure a file system installed in the internal eMMC

1. Optionally, backup the existing release kernel and dtb files to avoid overwriting.
2. Copy the compiled Image and `dtb` files over the current L4T release kernel directory by executing the following commands:

```
$ cp arch/arm/boot/Image <L4T_path>/Linux_for_Tegra/kernel
$ cp arch/arm/boot/dts/tegra210-jetson-cv-base-p2597-2180-a00.dtb <L4T_path>/Linux_for_
```

   `flash.sh` automatically copies the Image file to the internal eMMC root filesystem.

Adding a new Kernel

After U-Boot has been flashed as the default boot loader, you can replace the kernel. The procedure you should follow depends on the kind of storage device from which your device boots.

To replace the kernel in systems that boot from internal eMMC

1. Boot the Jetson TX1 system and log in.
2. Copy the new kernel files (using `scp`) into the `/boot` directory.
3. Reboot the Jetson TX1 system.

To replace the kernel in systems that boot from an SD Card or USB Pen Drive

1. Connect the SD Card or USB Pen Drive to your host system.

2. Copy the new kernel files to the `/boot` directory on the SD Card or USB Pen Drive.
3. Disconnect the SD Card or USB Pen Drive from the host system.
4. Connect the SD Card or USB Pen Drive to the Jetson TX1 system.
5. Reboot the Jetson TX1 system.

To replace the kernel in systems that boot from a TFTP/NFS server

1. Boot the Jetson TX1 system and log in.
2. On the target system enter the following command:

```
$ sudo mount /dev/mmcblk0p1 /mnt
```

3. Copy the new kernel files (using `scp`) to the `mnt/boot` directory.
4. Reboot the Jetson TX1 system.

Example Sysboot Configuration Files

eMMC Sysboot extlinux.conf File

For external media, you must copy the root filesystem to the device **after** running the `flash.sh` command. Then you attach the device.

The U-Boot functionality includes a default booting scan sequence. It scans bootable devices in the following order:

- External SD Card
- Internal eMMC
- USB Device
- NFS Device

It looks for an `extlinux.conf` configuration file in the following directory of the bootable device:

```
<rootfs>/boot/extlinux
```

Upon finding the `extlinux.conf` file, U-Boot does the following.

- Uses the `sysboot` command to read out boot configuration from `extlinux.conf`,
- Loads kernel Image file and device tree file, and then
- Boots the kernel.

The Image and device tree files are all user-accessible in the `<rootfs>/boot` location after booting. The `extlinux.conf` file is user accessible in the `<rootfs>/boot/extlinux` location. Users can easily change these files to test their own kernel without flashing.

The file `extlinux.conf` is a standard text-format sysboot configuration file that contains all boot information. It indicates the kernel image filename, the device tree blob filename, and the kernel boot command line. There are four example `extlinux.conf` files provided in the L4T release for each supported board:

```
extlinux.conf.emmc
```

```
extlinux.conf.sdcard
extlinux.conf.usb
extlinux.conf.nfs
```

During flashing, `flash.sh` copies the appropriate variant to the following location:

```
<rootfs>/boot/extlinux/extlinux.conf
```

The `extlinux.conf` files are very similar except for different kernel boot command lines. You can find the `extlinux.conf` files in the following location:

```
bootloader/<platform>/<board>
```

Where <platform> is `t210ref` and <board> is `p2371-2180-devkit` for Jetson TX1.

eMMC Sysboot extlinux.conf File

The `extlinux.conf` file has the following contents.

```
TIMEOUT 30
DEFAULT primary

MENU TITLE p2371-2180 eMMC boot options

LABEL primary
      MENU LABEL primary kernel
      LINUX /boot/Image
      FDT /boot/tegra210-jetson-cv-base-p2597-2180-a00.dtb
      APPEND fbcon=map:1 console=tty0 console=ttyS0,115200n8 androidboot.modem=none and
```

Different boot methods have different APPEND strings in the `extlinux.conf` file. Check each file for details.

Note: NFS booting also uses eMMC as boot device. <rootfs>/boot is flashed into to eMMC but kernel mounts NFS device a root filesystem.

Optimizing U-Boot Boot Time

By default, U-Boot includes a default configuration that enables all supported hardware features. It searches the available devices for boot scripts. This enables out-of-the-box support for the widest possible variety of storage devices and boot configurations.

This flexibility delays execution of the final operating system because hardware support takes time to initialize and scanning all attached storage and network devices takes time. In constrained or pre-configured systems, this flexibility may not be necessary. You may know, ahead of time, which storage device contains the required files, or that certain devices don’t need to be initialized by the boot-loader. To optimize boot time, you can configure U-Boot to allow for these constraints and thereby reduce system boot time.

Compile-Time Configuration

U-Boot pro-actively initializes certain types of devices when it starts. However, disabling features at compile time may be useful to reduce system boot time. For example, if PCIe and network support are enabled, the PCIe bus is enumerated at startup which can be a time-consuming process.

The U-Boot compile-time configuration is stored in the following files:

- include/configs/<board>.h
- configs/<board>_defconfig

Where:

- <board> is jetson-tk1 or p2371-2180.

Disabling features reduces the size of the U-Boot binary and thereby reduces the time it takes to load and initialize U-Boot itself.

Disabling PCIe

Since the PCIe bus is enumerated at startup, disabling it reduces overall system boot time.

To disable PCIe

- Remove the definition of the following values from `include/configs/<board>.h`:

```
CONFIG_PCI
CONFIG_PCI_TEGRA
CONFIG_PCI_PNP
CONFIG_CMD_PCI
CONFIG_CMD_PCI_ENUM
CONFIG_RTL8169 (This is a PCIe device and thereby relies on PCIe support)
```

Disabling USB Support

U-Boot delays enumerating and initializing USB devices until the user, or a boot script, explicitly attempts to access a USB device. Consequently, disabling USB support does not reduce boot time.

To disable USB support

1. Remove the following values from `include/configs/<board>.h`:

```
CONFIG_USB_EHCI
CONFIG_USB_EHCI_TEGRA
CONFIG_USB_MAX_CONTROLLER_COUNT
CONFIG_USB_STORAGE
CONFIG_CMD_USB
CONFIG_USB_HOST_ETHER
CONFIG_USB_ETHER_ASIX
```

2. Edit `include/configs/<board>.h` to remove the `#include` of `tegra-common-usb-gadget.h`.
3. Edit `include/configs/tegra-common-post.h` to remove the USB entry from `BOOT_TARGET_DEVICES`.

Environment Configuration

U-Boot runtime behavior is controlled by scripts contained in the U-Boot environment. When U-Boot begins execution, it waits for `<bootdelay>` seconds before executing the automatic boot sequence. During this time, the user may interrupt the boot process to access the U-Boot shell. If not interrupted, U-Boot executes `<bootcmd>` as a shell command. `<bootcmd>` contains a series of commands to search storage devices for boot scripts and execute them.

By modifying the values of these variables at compile or manufacturing time, U-Boot can be directed to boot from a specific device in a specific manner, thereby reducing boot time.

The following table identifies the variables that can be modified.

Variable	Description
bootdelay	Contains the number of seconds that U-Boot pauses to determine whether the user wishes to interrupt the boot sequence. To avoid delay, set to 0. **Note:** Although this value avoids delay, if the user has requested to interrupt the boot process before the U-Boot shell is reached, that request is honored. To avoid a delay and user interruption of the boot process, set `bootdelay` to a negative value.
bootcmd	Contains a sequence of U-Boot shell commands to be executed automatically at boot. For systems with custom requirements, this value can be completely replaced. However, simple customizations,

	such as selecting a specific storage device for booting, does not require editing this value.
boot_targets	Contains a space-separated list of storage devices or network protocols that U-Boot scans to find boot scripts. Valid values include: # mmc0 - the built-in eMMC. # mmc1 – the SD card slot. # usb0 – any attached USB Mass Storage device. # pxe – network, using DHCP to receive an IP address, then PXE to download a syslinux configuration file. # dhcp – network using DHCP to receive an IP address, then TFTP to download a U-Boot boot script. This variable can be set to a single specific device, or a more restrictive list than the default.
Applies to: L4T R23 releases and later scan_dev_for_boot_part	Contains a script to parse the device partition table and determines which partition U-Boot must scan for boot files. If the partition number is known ahead of time, replace the script with a simpler script that hardcodes the value of <devplist> with a single partition number (represented in hexadecimal), and then runs either or both of the following variables: # scan_dev_for_extlinux or # scan_dev_for_boot_scripts
boot_prefixes	Contains a list of filesystem directories to scan for boot scripts or configuration files. File system layouts vary between installations. For example, /boot may be a separate partition containing boot scripts, or part of the root filesystem. By default, U-Boot searches both locations for boot scripts. In constrained cases, the user may set this variable to a single directory name so that U-Boot does not search unnecessary directories. **Note:** All entries in this variable must contain both a leading and a trailing /.
scan_dev_for_extlinux	Contains a script to search for extlinux configuration files. If found, boots the system based on their content. If the system is known not to use extlinux configuration files, replace this script with commands that do nothing. For best results, set this variable to an innocuous value, such as “true”, rather than leaving it empty, so that U-Boot does not complain about attempts to execute an empty variable as a script.
scan_dev_for_scripts	Contains a script to search for U-Boot boot scripts. If found, loads and executes them. If the system is known not to use U-Boot boot scripts, replace this script with commands that do nothing.

	Note: The default boot scripts execute <scan_dev_for_extlinux> prior to executing <scan_dev_for_scripts>. Therefore, modifying this variable does not affect systems that boot using extlinux configuration files because this script will never be executed. For best results, set this variable to an innocuous value, such as “true”, rather than leaving it empty, so that U-Boot does not complain about attempts to execute an empty variable as a script.
boot_scripts	Contains a space-separated list of U-Boot script names for <scan_dev_for_scripts> to search for. If the script name is known ahead of time, set this variable to the desired value rather than the default list.

Setting Environment Variables

You can set environment variable at:

- Compile-Time
- Manufacturing Time
- Flashing Time

Compile-Time

When U-Boot starts executing, it attempts to initialize the environment variables from data stored in flash memory. The location of the data is determined by the U-Boot configuration file on the board. If the data is missing or corrupted, U-Boot uses the default set of values that are built into the U-Boot binary.

The default L4T flashing process does not write this data into flash memory, so the built-in copy is always used initially. The built-in default environment values are set in the following places in the U-Boot source code:

U-Boot Source Code	Variable Value
include/config_distro_bootcmd.h	This file defines the variables and scripts related to the automatic boot process.
include/config_distro_defaults.h	The value of CONFIG_BOOTDELAY determines the default value of <bootdelay>.
include/configs/tegra-common-post.h	The value of BOOT_TARGET_DEVICES determines: # The default value of <boot_targets>. # The set of legal values that can appear in <boot_targets>. Removing entries from this variable prevents the use of those values in <boot_targets> at all. To modify this value at compile-time: 1. Modify include/config_distro_bootcmd.h to avoid setting <boot_targets> if the board-

	specific configuration file has already defined this value. 2. Modify include/config/<board>.h to set <boot_targets>. For example, see how CONFIG_BOOTCOMMAND is defined in L4T R23 and later.

Manufacturing and Flashing Time

You can modify your manufacturing flow to add an extra step that writes a saved copy of the environment variables to flash. The following provides a set of alternatives to use for modifying your manufacturing flow:

- After flashing the board, arrange for the board to execute U-Boot, and cause U-Boot to execute commands that modify and save the environment:
  - Reset the board so that U-Boot runs and sends commands to U-Boot using the serial console. You must write and execute some program on your host system to send the commands to U-Boot.
  - Place the board into USB recovery mode. Download U-Boot into RAM. Package the required commands along with the U-Boot binary. Execute the following from the directory where you extracted L4T on your host system:

    ```
    ./bootloader/exec-uboot.sh <board> "<uboot_script>"
    ```

    Where `<uboot_script>` is a U-Boot script that sets the environment variables according to your needs.

    Note: L4T Release 21 does not contain an equivalent of `exec-uboot.sh`. However, use `tegra-uboot-flasher` instead. See https://github.com/NVIDIA/tegra-uboot-flasher-scripts for details.

  - Execute a sequence of commands similar to the following:

    ```
    env default -f -a; setenv boot_targets mmc0; saveenv
    ```

- Manually edit the U-Boot environment interactively, save the result, and extract the appropriate flash memory region to a file on your host system. Then, modify the L4T partition layout XML file to that the data is written to flash during any subsequent flashing process.

  Alternatively, apply this technique if your manufacturing process programs flash chips directly, rather than using the L4T flashing tools on each board. You must merge the extracted saved environment data into your flash image file, rather than referencing it from the L4T partition layout XML file.

- Generate a set of saved environment data using the U-Boot `fw_env` tool located in the tools/env directory in the U-Boot source code. Then, modify the L4T partition layout XML file so that data is written to flash during any subsequent flashing process.

  Alternatively, apply this technique if your manufacturing process programs flash chips directly, rather than using the L4T flashing tools on each board. You must merge the extracted saved environment data into your flash image file, rather than referencing it from the L4T partition layout XML file.

extlinux.conf Modifications

L4T includes an `extlinux.conf` file that tells U-Boot which kernel and DT filenames to load. It also includes the commandline to pass to the kernel. The file is configured to display a boot menu to the user for 3 seconds before automatically booting. To reduce boot processing time, you can remove this timeout.

To remove extlinux.conf timeout boot menu

- Edit `extlinux.conf` file to remove the lines containing the following keywords:

  ```
  TIMEOUT
  MENU
  ```

  Prior to flashing, this file is located in the L4T flashing directory as:

  ```
  bootloader/<board_dir>/extlinux.conf.emmc
  ```

  Where <board_dir> is:

  - “ardbeg” for L4T Release 21 (Jetson TK1).
  - “t210ref/p2371-2180” or “t210ref/p2371-2180-refkit” for L4T Release 23 (p2371-2180 / Jetson TX1).

Debugging U-Boot Environment

Use these debugging tips to help you debug your U-Boot environment. These examples do not represent a comprehensive listing of U-Boot functionality. For a full list of supported commands and their usage by U-Boot, consult U-Boot documentation and source.

Interrupting U-Boot

You can interrupt U-Boot during boot.

To interrupt U-Boot

- Connect a terminal application to the serial port of the board.
- Power on or reset the board.
- Wait for the U-Boot sign on message to appear. Key presses before this point in time may be ignored.
- Press any key to interrupt the automatic boot process.

Getting Help

On the U-Boot terminal screen, type `help` at any time for the list of supported commands from the U-Boot terminal.

To see the U-Boot Help text

- To see the U-Boot help text enter the following command:

```
# help
```

Listing a Directory Structure

You can list the directory structure of a particular device.

To list the directory structure

- To list the directory structure of eMMC device 0 partition 1, enter the following command:

```
# ls mmc 0:1
```

This command functions correctly on EXT2/3/4 and FAT file systems.

Example output follows:

```
<DIR>        4096 .
<DIR>        4096 ..
<DIR>        4096 bin
<DIR>        4096 boot
<DIR>        4096 dev
<DIR>        4096 etc
<DIR>        4096 home
<DIR>        4096 lib
<DIR>        4096 lost+found
<DIR>        4096 media
<DIR>        4096 mnt
<DIR>        4096 opt
<DIR>        4096 proc
<DIR>        4096 root
<DIR>        4096 sbin
<DIR>        4096 selinux
<DIR>        4096 srv
<DIR>        4096 sys
<DIR>        4096 tmp
<DIR>        4096 usr
<DIR>        4096 var
```

Listing the Contents of a Directory

You can list the contents of any directory.

To list the contents of a directory

- List directory contents with the following command:

```
# ls mmc 0:1 <directory>
```

Where <directory> is a pathname in the filesystem.

For example, to list contents of the /boot directory where the zImage file should be, (as shown in the example output below), use the following command:

```
# ext2ls mmc 0:1 /boot
<DIR>       1024 .
<DIR>       1024 ..
           34642 tegra124-pm375.dtb
             908 extlinux.conf
         5910248 zImage
```

Printing the U-Boot Environment

You can print the entire U-Boot environment.

To print the U-Boot environment

- Execute the following command:

```
# printenv
```

Printing/Setting Environment Variables

You can print and set environment variables.

To print an environment variable

- Execute the following command:

```
# printenv <environment_variable>
```

Where <environment_variable> refers to an environment variable in U-Boot.

For example, to print the list of devices U-Boot sends console output to, execute:

```
# printenv stdout
```

Output can be as follows:

```
stdout=serial
```

To set an environment variable

- Execute the following command:

```
# setenv <environment_variable> <new_value>
```

Where `<environment_variable>` refers to an environment variable in U-Boot and <new_value> is the new value for that variable.

For example, to modify the set of devices that U-Boot sends console output to, execute:

```
# setenv stdout serial
```

To save the modified environment

- Execute the following command:

```
# saveenv
```

The saved modified environment is preserved in case of resets and reboots.

Kernel Boot Time Optimization

NVIDIA® Tegra® Linux Driver Package (L4T) provides a generic boot kernel for development of your product. To decrease kernel boot time, you can customize the provided kernel based on the requirements of your product.

The kernel includes a default configuration that enables all supported hardware features, and searches all available devices for boot scripts. This enables out-of-the box support for the widest possible variety of controllers, features, storage devices, and boot configurations.

This flexibility comes at a cost:

- Some hardware support takes time to initialize
- Enabling all software features, mostly over Advanced Peripheral Bus (APB), takes time
- Scanning all attached storage and network devices takes time thereby delaying execution of the final operating system

In constrained or pre-configured systems, this flexibility may not be necessary; the system designer may know ahead of time which storage device contains the required files, or know that certain devices do not need to be initialized by the kernel. To reduce system boot time, you can configure the kernel to respect these constraints.

For a Jetson Tegra X1 system running L4T R24 with the default configuration, it takes 12 seconds from cold power-on to begin showing the login prompt. When the following optimization techniques are applied, that process can be reduced to approximately three seconds.

Device Tree Nodes

If you are not using any controller from Tegra SoC, disable the Device Tree nodes for those device tree entries.

PCIe

If you are not using PCIe, disable both the ports from the kernel Device Tree Blob (DTB). If you are using one port, disable the second one.

The Device Tree Source (DTS) file in the L4T package is available at:

```
arch/arm64/boot/dts/tegra210-jetson-cv-base-p2597-2180-a00.dts
```

To disable PCIe ports

- Modify the controller and each PCIe port status as shown in the following example.

```
pcie-controller {
```

```
                nvidia,wake-gpio = <&gpio TEGRA_GPIO(A, 2) 0>;
                nvidia,lane-map = <0x14>;
                dvdd-pex-pll-supply = <&max77620_ldo1>;
                l0-dvddio-pex-supply = <&max77620_ldo1>;
                l1-dvddio-pex-supply = <&max77620_ldo1>;
                l2-dvddio-pex-supply = <&max77620_ldo1>;
                l3-dvddio-pex-supply = <&max77620_ldo1>;
                l4-dvddio-pex-supply = <&max77620_ldo1>;
                l5-dvddio-pex-supply = <&max77620_ldo1>;
                l6-dvddio-pex-supply = <&max77620_ldo1>;
                hvdd-pex-pll-e-supply = <&max77620_sd3>;
                l0-hvddio-pex-supply = <&max77620_sd3>;
                l1-hvddio-pex-supply = <&max77620_sd3>;
                l2-hvddio-pex-supply = <&max77620_sd3>;
                l3-hvddio-pex-supply = <&max77620_sd3>;
                l4-hvddio-pex-supply = <&max77620_sd3>;
                l5-hvddio-pex-supply = <&max77620_sd3>;
                l6-hvddio-pex-supply = <&max77620_sd3>;
                vddio-pex-ctl-supply = <&max77620_sd3>;
                status = "disabled";

                pci@1,0 {
                        status = "disabled";
                };

                pci@2,0 {
                        status = "disabled";
                };
        };
```

Pinmux

There is duplicity of pinmux and GPIO configuration in u-boot and the kernel; however you can remove this duplicity modifying the configuration. The default configurations are performed based on customer_pinmux.xlsm sheet at one place (u-boot). Extra configuration can be performed by the specific driver.

To eliminate pinmux duplicity

- Remove the following pinmux-name nodes from the device tree DTS file.

```
/ {
        pinmux: pinmux@700008d4 {
                status = "okay";
pinctrl-names = "default", "drive", "unused";
                pinctrl-0 = <&pinmux_default>;
                pinctrl-1 = <&drive_default>;
                pinctrl-2 = <&pinmux_unused_lowpower>;
```

Real-time Clock

Two Real-time Clocks (RTC) are enabled by default:

- Tegra RTC
- PMCI RTC

Enable one, both, or none, based on your development needs. Be aware of the following differences:

- Tegra RTC is the fastest in response, but cannot work as a backup RTC when the system is turned off.
- PMIC RTC can work as a backup RTC, however it is slow in response because of the transfer average to I2C.

If you do not shutdown the system, of if you are not concerned about backup time, then you do not need to enable RTC. Disabling RTC can speed up boot time.

To disable the RTC

1. Remove the the following configuration from your `tegra21_defconfig` file.

   ```
   CONFIG_TRC_DRV_MAX77620=y
   ```

2. Add the following configuration to the `tegra21_defconfig` file.

   ```
   # CONFIG_RTC_HCTOSYS is not set
   ```

3. Set the Tegra RTC to disabled in the DTS file as follows:

   ```
   rtc {
           compatible = "nvidia,tegra-rtc";
           reg = <0x00000000 0x00000003 0x00000002 0x0000009b>;
           interrupts = <0x00000000 0x00000005 0x00000002>;
           status = "disabled";
       };
   ```

Environment Configuration

You can optimize boot time by modifying the environment configuration in the root file system.

Single Step Boot

If you are not using the Network File System (NFS), you can reduce the kernel boot time by approximately 3 seconds by modify the default initial ramdisk (`initrd`) to disable it from loading the temporary root file system.

By default, `initrd` is set for `rootdev_type` for network and external boot media. For example, SD card, USB stick.

To modify the initrd settings

1. Execute the following command to rename the original initrd file.

   ```
   mv l4t_initrd.img l4t_initrd.img_orig
   ```

2. Execute the following command:

   ```
   touch l4t_initrd.img
   ```

Disable Console over UART

Prints over UART are a major bottleneck in kernel boot time. To reduce this time, you can remove `console=ttyS0` from the `extlinux.conf` configuration file.

Once the system is ready for deployment, you can remove the UART console logs or review the console logs over the Framebuffer console. The console log is `console=tty1` in the `extlinux.conf` configuration file.

Compile-Time Configuration

To reduce compile-time configuration, examine the generated configuration file to identify which configurations are required. Once the required configurations are defined, identify which ones to boot asynchronously. For those configurations, the drivers probe is executed asynchronously in a separate thread instead of the main initial thread.

Additionally, examine the required configurations and verify that they can be programmed as modules so that the drivers are loaded when it is called for use. When the drivers are not loaded, the kernel image is reduced and more RAM space is available.

The following topics provide examples of each of these conditions.

Asynchronous Probe

The asynchronous probe feature is available, by default, in Kernel Version 3.18. For Kernel Version 3.10, you must cherry-pick the patches from the mainline available at:

```
https://lkml.org/lkml/2015/1/16/576
```

To move the driver to another thread

- Add the `probe_type` in your driver as follows:

```
static struct platform_driver sdhci_tegra_driver = {
        .driver         = {
                .name   = "sdhci-tegra",
                .of_match_table = sdhci_tegra_dt_match,
                .pm     = SDHCI_PLTFM_PMOPS,
                .probe_type = PROBE_PREFER_ASYNCHRONOUS,
        },
        .probe          = sdhci_tegra_probe,
        .remove         = sdhci_tegra_remove,
        .shutdown       = sdhci_tegra_shutdown,
};
```

File System

To decrease boot time the filesystem, modify the following configurations to set them as modules:

```
CONFIG_FUSE_FS=m
CONFIG_VFAT_FS=m
CONFIG_NTFS_FS=m
```

Sound

Audio codec requires some time to initialize, To eliminate this initialization time, disable the audio configurations as follows:

```
# CONFIG_SND_SOC_TEGRA_ALT is not set
# CONFIG_SND_SOC_TEGRA_T210REF_MOBILE_ALT is not set
# CONFIG_SND_SOC_TEGRA_T210REF_MOBILE_ES755_ALT is not set
```

Lauterbach Debugging Scripts

Setting Up the Lauterbach Debugging Scripts Environment

The Lauterbach scripts supplied with this release include:

Script	Description
axi_attach.cmm	Sets the CPU for AXI access
config_coredump.t32	Provides environment variable settings and driver settings
config_cpu.t32	Provides environment variable settings
config_cpu_win.t32	Provides environment variable settings
cpu_attach.cmm	Attaches to CPU on Tegra <platform> BSP for kernel
cpu_boot_attach.cmm	Attaches to CPU on Tegra <platform> BSP for Ethernet boot
cpu_boot_sdram_noload.cmm	Boots CPU with various configurations
cpu_dcc_setup.cmm	Configures DCC for the CPU
cpu_dcc_swi_setup.cmm	Configures DCC using SWI method
cpu_disable_mmu.cmm	Disables the CPU MMU and caches
cpu_kernel_attach.cmm	Attaches to CPU with kernel symbols loaded
cpu_kernel_load.cmm	Loads kernel image via JTAG loader
cpu_menu_setup.cmm	Installs CPU-side menu buttons
cpu_monitor_attach.cmm	Attaches to the CPU with the monitor symbols loaded
cpu_mp_attach.cmm	Sets up CPU for complex core/ multiprocessor settings
cpu_uboot-attach.cmm	Boots CPU on with U-Boot already present in SDRAM
csite_cpu.cmm	Dumps CoreSight CPU apertures
install_customer_scripts	Installs scripts to the `$T32SYS` (Android) `C:\T32` (Windows) directory, then prompts user to customize configuration script
install_scripts	Installs lauterbach scripts

physical_setup.cmm	Reconfigures for boot loader physical addressing mode
setup_customer_environment.cmm	Sets up paths and global environment variables used by other scripts
t32.cmm	Initializes TRACE32
t32_customer.cmm	Default startup program for TRACE32
t32cpu.bat	Specifies TRACE32 instance is CPU for start up
toolbar_setup.cmm	Sets up common toolbar items
user_config_customer.cmm	Sets user-specific parameters, such as script variables
virtual_setup.cmm	Reconfigures virtual addressing mode for kernel
windows.cmm	Provides Windows settings
t21x/t21x_axi_attach.cmm	Sets user-specific parameters for TRACE32
t21x/t21x_cpu_jtag_setup.cmm	Sets user-specific parameters for TRACE32
t21x/t21x_cpu_mp_jtag_setup.cmm	Sets user-specific parameters for TRACE32
t21x/t21x_init_cpu.cmm	Sets user-specific parameters for TRACE32

Setting Up the Lauterbach Debugging Scripts Environment

Four sets of commands must be run to set up the environment to execute the Lauterbach scripts. These are detailed below.

To setup to run Lauterbach

1. Add these variables to `~/.bashrc`:

```
$ export T32SYS=<directory you chose as your Trace32 install directory>
$ export T32TMP=/tmp
$ export T32ID=T32
$ export PATH=$PATH:$T32SYS/bin/pc_linux64:$T32SYS
```

2. In your build directory, set the following:

```
$ export TEGRA_TOP=$(pwd)
$ export TARGET_BOARD=t210ref
$ export TARGET_OS_SUBTYPE=gnu_linux
```

3. Download the tar ball of Lauterbach scripts from the link to them under the "Downloads" button and extract them.

The correct paths for Image and vmlinux are setup in the `user_config_customer.cmm` script.

4. Copy the required files to your t32 directory:

```
$ sudo -E ./install_customer_scripts
$ cp user_config_customer.cmm /opt/t32/user_config.cmm
$ cp ./setup_customer_environment.cmm ./setup_environment.cmm
```

5. Execute the following command on the host system:

```
$ t32cpu-64 &
```

6. Execute the following commands on the device:

```
$ echo 0 > /sys/devices/system/cpu/cpuquiet/tegra_cpuquiet/enable
$ echo 1 > /sys/kernel/debug/cpuidle_t210/fast_cluster_states_enable
$ echo 1 > /sys/kernel/debug/cpuidle_t210/slow_cluster_states_enable
```

Video for Linux User Guide

This document provides information on use of the MIPI Camera Serial Interface (CSI) on NVIDIA® Tegra® X1, using software from the NVIDIA® Tegra® Linux Driver Package (also referred to as L4T). The MIPI CSI protocol, V4L2 API, Tegra X1 system architecture and method of attaching a CSI camera to Jetson TX1 are outside the scope of this document.

The V4L2 software implementation bypasses the Tegra ISP, and is suitable for use when Tegra ISP support is not required, such as with sensors or input devices that provide data in YUV format.

References to additional resources are provided, but the reader should already be familiar with Tegra X1, and have access to the *Tegra Technical Reference Manual* (TRM) and other documentation available at the Jetson Embedded Platform portal:

http://developer.nvidia.com/embedded-computing

V4L2/SOC_CAMERA Overview

V4L2 is the second version of Video4Linux or V4L, a video capture and output device API and driver framework in the Linux kernel. It supports many USB webcams, TV tuners, and other devices and is closely integrated with the Linux kernel. For a description of the APIs, see Linux Media Infrastructure APIs.

`soc-camera` is a set of drivers and a core module, that implement V4L2 functionality on embedded devices; typically a video-enabled embedded device: SoC with a capture interface and video data sources. The `soc-camera` includes host driver such as the Tegra V4L2 camera driver and client drivers (sensor drivers).

Note:	**Software releases after R23: `soc_camera` driver is deprecated and replaced with the media-controller driver. The media-controller driver framework is V4L2-compatible and provides greater functionality than the `soc_camera` driver. Existing V4L2 sensor drivers require minor modifications to be compatible with media-controller. Plan your software development for this transition.**

V4L2 on Jetson TX1

Jetson TX1 is a powerful embedded development board including the NVIDIA® Tegra® X1 processor. Tegra X1 processors have a video input interface (VI) and camera serial interface (CSI), so Tegra X1 can communicate with the external video input sources, such as camera sensor module or other MIPI CSI compatible devices. VI/CSI of Tegra X1 also has two test pattern generators that can generate some data patterns like color bricks for testing purpose. You can find out more about this development board at:

http://elinux.org/Jetson_TX1

On the software side, Linux for Tegra (L4T) latest release R23 provides a Tegra V4L2 camera driver and sample drivers for a camera sensor and a built-in test pattern generator (TPG). With an open source V4L2 user space tool like Yavta, users can capture data from the TPG and sensors. For more information about Yavta, see:

http://git.ideasonboard.org/yavta.git

Test Pattern Generator

The test pattern generator is a configurable resource introduced to improve hardware verification capability for the Tegra CSI. There are two separate test pattern generators that can be configured to provide for the generation of synthetic image data, which is delivered to the PPA and PPB input FIFOs. The image data is multiplexed into the CSI data patch between lane-merging logic and the data FIFOs.

L4T provides a virtual V4L2 `soc_camera` sensor driver for exposing TPG functionality (`soc_camera_platform driver`). It can generate 1280x720 resolution RGBA32 color bricks data. There is no need to rebuild the kernel and the `soc_camera_platform driver` is provided as a loadable module.

To verify the TPG

1. Remove the `nvhost_vi` module, an incompatible non-V4L2 VI driver used for other purposes and outside the scope of this document:

   ```
   $ sudo rmmod nvhost_vi
   ```

2. Install V4L2 driver modules:

   ```
   $ sudo modprobe soc_camera_platform
   $ sudo modprobe tegra_camera tpg_mode=2
   ```

3. Use the yavta application to capture data (other V4L2 applications can be used, if preferred)

   ```
   $ ./yavta /dev/video0 -c1 -n1 -s1280x720 -fRGB32 -Ftpg.rgba
   ```

4. Copy over `tpg.rgba` to host and use ImageMagick to show the picture:

   ```
   $ display -size 1280x720 -depth 8 tpg.rgba
   ```

Example Sensor: OV5693

L4T provides a sample V4L2 sensor driver for the OmniVision OV5693 Bayer sensor. This driver can be used as a reference in creating a custom V4L2 sensor driver. NVIDIA provides a reference OV5693 camera module E3326, with Jetson TX1.

The driver for OV5693 is neither built into the kernel nor built as module. Please try the following steps to test OV5693 in L4T on Jetson TX1.

Hardware setup:

- Jetson TX1
- OV5693 camera module

To test OV5693 in L4T on Jetson TX1

1. Apply patches on top of L4T release.

   L4T 23.1 Release (19 patches)

   ```
   $ tar xfz r23.1_v4l2.tgz

   $ ls r23.1/
   0001-drivers-soc_camera-add-ov23850-OTP-controls.patch
   0002-drivers-soc_camera-sensor-drivers-update.patch
   0003-drivers-media-platform-tegra-get-gpio-pwr-info.patch
   0004-drivers-media-soc_camera-IMX230-driver-to-gain.patch
   0005-drivers-Restructure-GRHOST_VI.patch
   0006-media-v4l2-core-Migration-from-upstream.patch
   0007-media-soc_camera-add-ov13860-v4l2-driver.patch
   0008-drivers-media-platform-ov5693-gpio-warning.patch
   0009-media-tegra_camera-support-YUV-CSI-input.patch
   0010-kernel-changes-for-implementing-sensor-cropping.patch
   0011-drivers-soc_camera-give-default-hdr_en.patch
   0012-media-tegra_camera-fix-syncpoint-time-out-issue.patch
   0013-media-soc-camera-actually-use-default-resolution.patch
   0014-media-soc_camera-add-tc358840-v4l2-driver.patch
   0015-media-soc_camera-vi2-support-gang-mode.patch
   0016-media-camera-common-export-symbols-for-modules.patch
   0017-media-tegra_camera-init-mipi-bias-pad.patch
   0018-ARM64-t210-add-E3326-camera-and-camera_common.patch
   0019-ARM64-adding-OV5693-V4L2-on-E3326-jetson_cv.patch

   $ git am r23.1/*
   ```

   L4T R23.2 Release (1 patch)

   ```
   $ git am 0001-ARM64-adding-OV5693-V4L2-on-E3326-jetson_cv.patch
   ```

2. Enable OV5693 kernel driver and disable `soc_camera_platform`.
   - CONFIG_SOC_CAMERA_OV5693=m
   - CONFIG_VIDEO_TEGRA_VI=m

- Disable CONFIG_SOC_CAMERA_PLATFORM
- Disable CONFIG_SOC_CAMERA_OV13860
- Disable CONFIG_SOC_CAMERA TC358840

3. Build kernel, flash Jetson TX1 and boot the Linux OS.
4. Use Yavta to capture a frame.

```
$ sudo rmmod nvhost-vi
$ sudo modprobe ov5693_v4l2
$ sudo modprobe tegra_camera
$ ./yavta /dev/video0 -c1 -n1 -s1920x1080 -fSRGGB10 -Fov.raw
```

5. Use `raw2bmp` to convert raw data to BMP file.

```
$ ./raw2bmp ov.raw ov.bmp 1920 1080 16 3
```

V4L2 Tegra Driver Overview

As V4L2 is a kernel video input framework, Tegra V4L2 stack contains several components. It controls hardware such as the Tegra VI/CSI hardware controller and external sensors. Additionally, it exports a generic device node named `/dev/video<N>` to user space, where `<N>` is a numeric value. User space applications can use V4L2 standard API to control real hardware via `/dev/video<N>`.

This section focuses on Tegra X1-related drivers and code in L4T kernel source.

Tegra V4L2 Camera Driver

Tegra V4L2 camera driver is a part of soc_camera and acts as a host driver. It directly controls Tegra X1 VI/CSI hardware. Normally users don’t need to modify this driver, but developers should become familiar with it; it may require customization for some use cases.

- Source code

```
drivers/media/platform/soc_camera/Kconfig
drivers/media/platform/soc_camera/Makefile
drivers/media/platform/soc_camera/tegra_camera/*
include/media/tegra_v4l2_camera.h
```

- Kernel config

```
CONFIG_VIDEO_TEGRA=m
```

- `tegra_camera.ko` module

  The module name is `tegra_camera.ko` and it won’t be loaded by default after booting into L4T.

There is another driver named `nvhost_vi.ko` installed by default and is mutually-exclusive with `tegra_camera.ko`, so users must remove the `nvhost_vi.ko` before loading `tegra_camera.ko`.

- Input data format

  Tegra X1 VI/CSI hardware supports 3 major input data formats: YUV, RGB, and Bayer RAW. However in this driver only the following have been tested:

  - RGB888
  - RAW8
  - RAW10
  - YUV422

Note:	Some other formats are also supported by hardware, but software support is not present in the driver. Please refer to the Tegra TRM for details on supported input formats.

Please study the source code then add new input data formats not listed here.

Tegra V4L2 Sensor Driver

V4L2 sensor driver normally is an I2C device driver and in L4T it is also a V4L2 `soc_camera` client driver. It has several I2C register tables for different resolutions like 1920x1080, 1280x720, etc. When a user space application opens `/dev/video<N>`, the sensor driver powers on the sensor hardware and programs it with the register table via I2C.

- Real sensor code

```
drivers/media/i2c/soc_camera/ov5693_v4l2.c
drivers/media/i2c/soc_camera/ov5693_mode_tbls.h
include/media/ov5693.h
drivers/media/i2c/soc_camera/Kconfig
drivers/media/i2c/soc_camera/Makefile
```

- Test Pattern Generator virtual sensor driver source code

```
drivers/media/platform/soc_camera/soc_camera_platform.c
```

  The `soc_camera_platform` driver does not perform any real hardware operations like power control and I2C transactions. It is just a virtual driver to enable use of the TPG for testing.

- Kernel configs

```
CONFIG_SOC_CAMERA_OV5693
CONFIG_SOC_CAMERA_PLATFORM
```

- Power controls

  Each sensor has its own power on/off sequence, clock settings and other hardware specific operations. L4T sensor driver puts these power controls in the sensor driver itself. For more flexible driver design, these power controls must go to board files since each hardware board may have different power controls. Then, the sensor driver itself can be more generic. Normally, power controls include:

- GPIO for sensor reset, power on or power down
- Regulators for sensor power supply
- Clocks for sensor running like mclk or sensor local clock

Board File

Before fully moving to device tree binding, a board file is the only way to describe platform-specific configurations within the Linux kernel. Beginning in L4T R23 releases, most hardware devices use device tree binding, but V4L2 `soc_camera` still uses a board file approach.

- Source code

  ```
  arch/arm64/mach-tegra/board-t210ref-camera.c
  ```

- TPG board configs

  `soc_camera_platform_info` defines data format and resolution which must be matched with our TPG hardware.

  ```
  static struct soc_camera_platform_info t210ref_soc_camera_info = {
          .format_name = "RGB4",
          .format_depth = 32,
          .format = {
                  .code = V4L2_MBUS_FMT_RGBA8888_4X8_LE,
                  .colorspace = V4L2_COLORSPACE_SRGB,
                  .field = V4L2_FIELD_NONE,
                  .width = 1280,
                  .height = 720,
          },
          .set_capture = t210ref_soc_camera_set_capture,
  };
  ```

  `tegra_camera_platform_data` is the most important data structure to describe the sensor connection. `.port` indicates which CSI port the sensor connects to:

  TEGRA_CAMERA_PORT_CSI_A means the sensor uses CIL_A.

  TEGRA_CAMERA_PORT_CSI_B means the sensor uses CIL_B.

  TEGRA_CAMERA_PORT_CSI_C means the sensor uses CIL_C.

  TEGRA_CAMERA_PORT_CSI_D means the sensor uses CIL_D.

  TEGRA_CAMERA_PORT_CSI_E means the sensor uses CIL_E.

TEGRA_CAMERA_PORT_CSI_F means the sensor uses CIL_F.

Tegra X1 internally has 6 CSI channels (CSI_A to CSI_F). CSI A/C/E channel can support 1, 2 and 4 data lane sensors and CSI B/D/F can support 1 and 2 data lane sensors..

```
static struct tegra_camera_platform_data t210ref_camera_platform_data = {
        .flip_v                 = 0,
        .flip_h                 = 0,
        .port                   = TEGRA_CAMERA_PORT_CSI_A,
        .lanes                  = 4,
        .continuous_clk         = 0,
};
```

- OV5693 board file configs

  Real sensors do not require that sensor resolution or data format information be put into the board file like TPG `soc_camera_platform` driver, because that information is in the sensor driver itself.

  - OV5693 connects to port CSI_C via 2 data lanes:

    ```
    static struct tegra_camera_platform_data
    t210ref_ov5693_e3326_camera_platform_data = {
            .flip_v                 = 0,
            .flip_h                 = 0,
            .port                   = TEGRA_CAMERA_PORT_CSI_C,
            .lanes                  = 2,
            .continuous_clk         = 0,
    };
    ```

  - OV5693 uses I2C bus 6 and it’s I2C address is 0x36:

    ```
    static struct camera_common_pdata t210ref_ov5693_e3326_data = {
            .regulators = {
                    .avdd = "vana",
                    .iovdd = "vif",
            },
            .reset_gpio = 148, /* TEGRA_GPIO_PS4 */
            .pwdn_gpio = 151, /* TEGRA_GPIO_PS7 */
    };

    static struct i2c_board_info t210ref_ov5693_e3326_camera_i2c_device = {
            I2C_BOARD_INFO("ov5693_v4l2", 0x36),
    };
    ```

```
static struct soc_camera_link ov5693_e3326_iclink = {
        .bus_id         = 0, /* This must match the .id of tegra_vi01_device */
        .board_info     = &t210ref_ov5693_e3326_camera_i2c_device,
        .module_name    = "ov5693_v4l2",
        .i2c_adapter_id = 6, /* VI2 I2C controller */
        .power          = t210ref_ov5693_power,
        .priv           = &t210ref_ov5693_e3326_camera_platform_data,
        .dev_priv       = &t210ref_ov5693_e3326_data,
};
```

- Register OV5693 soc_camera platform device:

```
platform_device_register(&t210ref_ov5693_e3326_soc_camera_device);
```

Device Tree File

Device tree provides regulator information required by the V4L2 sensor driver. OV5693 sensor driver use 2 regulators: `vana` and `vif`. They are defined in:

```
arch/arm64/boot/dts/tegra210-platforms/tegra210-jetson-e-pmic-p2530-0930-e03.dtsi
arch/arm64/boot/dts/tegra210-platforms/tegra210-jetson-e-power-fixed-p2530-0930-e03.dts
```

Writing and Integrating a Sensor Driver for L4T

Developers can write their own sensor driver for their specific device. Sensor drivers usually have very similar structures but different I2C register tables. Modification of the board file and the device tree file is required for different boards.

Sensor Driver Development

The OV5693 sensor drivers are a good starting point for writing a new sensor driver. The following steps are recommended for developing a new driver:

- Import new I2C register tables

  Sensor vendors provide I2C register settings as tables, which must be added to sensor driver. The following struct is a good example:

```
static const ov5693_reg *mode_table[] = {
        [OV5693_MODE_2592X1944]                 = mode_2592x1944,
        [OV5693_MODE_2592X1458]                 = mode_2592x1458,
```

```
        [OV5693_MODE_1920X1080]                  = mode_1920x1080,
        [OV5693_MODE_1296X972]                   = mode_1296x972,
        [OV5693_MODE_1280X720_120FPS]            = mode_1280x720_120fps,
        [OV5693_MODE_2592X1944_HDR]              = mode_2592x1944_HDR_24fps,
        [OV5693_MODE_1920X1080_HDR]              = mode_1920x1080_HDR_30fps,
        [OV5693_MODE_1296X972_HDR]               = mode_1296x972_HDR_30fps,

        [OV5693_MODE_START_STREAM]               = ov5693_start,
        [OV5693_MODE_STOP_STREAM]                = ov5693_stop,
        [OV5693_MODE_TEST_PATTERN]               = tp_colorbars,
};
```

- Power controls

  Different boards have different sensor power controls. It is better put those power controls into a board file. But it is simpler to implement them in a sensor driver. Please take a look at `OV5693_power_on()` and `OV5693_power_off()` functions.

- `soc_camera` and I2C interface

  The sensor driver implements `soc_camera_ops` functions as well as I2C device probing/removing functions. Normally these are quite similar across different sensor drivers—just reuse them in your driver and use `OV5693_v4l2.c` as an example.

- KConfig and Makefile

  Add a `SOC_CAMERA_OV5693` entry into the `Kconfig` and `Makefile` files.

- Header file `include/media/ov5693.h`

  This header contains some information for non-V4L2 NVIDIA camera stacks. The following structures can be reused if necessary:

```
struct ov5693_regulators {
        const char *avdd;
        const char *dvdd;
        const char *dovdd;
};

struct ov5693_platform_data {
        unsigned cfg;
        unsigned num;
        const char *dev_name;
        unsigned gpio_count; /* see nvc.h GPIO notes */
        struct nvc_gpio_pdata *gpio; /* see nvc.h GPIO notes */
```

```
        struct nvc_imager_static_nvc *static_info;
        bool use_vcm_vdd;
        int (*probe_clock)(unsigned long);
        int (*power_on)(struct ov5693_power_rail *);
        int (*power_off)(struct ov5693_power_rail *);
        const char *mclk_name;
        struct nvc_imager_cap *cap;
        struct ov5693_regulators regulators;
        bool has_eeprom;
        bool use_cam_gpio;
};
```

Board File and Device Tree File Updates

A new project or new hardware board may have a new board file such as board-`t210ref*.c` for Jetson TX1. If so, the new board file must include those settings for sensor drivers. Follow this template in the board file and replace `SENSOR` with your sensor name:

```
#if IS_ENABLED(CONFIG_SOC_CAMERA_SENSOR)
static int t210ref_sensor_power(struct device *dev, int enable)
{
        return 0;
}
// NOTE: power controls can go here instead of sensor driver itself.

struct sensor_platform_data t210ref_sensor_data;

static struct i2c_board_info t210_sensor_camera_i2c_device = {
        I2C_BOARD_INFO("sensor_v4l2_driver_name", sensor_i2c_address),
        // sensor_v4l2_driver_name should match the sensor driver's module name
        .platform_data = &t210ref_sensor_data,
};

static struct tegra_camera_platform_data t210ref_sensor_camera_platform_data = {
        .flip_v                 = 0,
        .flip_h                 = 0,
        .port                   = TEGRA_CAMERA_PORT_CSI_X_for_sensor,
        .lanes                  = number_of_sensor_data_lanes,
        .continuous_clk         = 0,
```

```
};

static struct soc_camera_link sensor_iclink = {
        .bus_id         = 0,
        .board_info     = &t210ref_sensor_camera_i2c_device,
        .module_name    = "sensor_v4l2_driver_name",
        .i2c_adapter_id = sensor_i2c_bus_number,
        .power          = t210ref_sensor_power,
        .priv           = &t210ref_sensor_camera_platform_data,
};

static struct platform_device t210ref_sensor_soc_camera_device = {
        .name   = "soc-camera-pdrv",
        .id     = 0,
        .dev    = {
                .platform_data = &sensor_iclink,
        },
};
#endif
```

- Finally register the platform device in `t210ref_camera_init()`:

```
#if IS_ENABLED(CONFIG_SOC_CAMERA_SENSOR)
        platform_device_register(&t210ref_sensor_soc_camera_device);
#endif
```

- Device tree update

  Find the new device tree file for the new board and update regulator information appropriate to the hardware configuration of the new board. A good example to look at is:

```
arch/arm64/boot/dts/tegra210-platforms/tegra210-jetson-e-pmic-p2530-0930-e03.dtsi
arch/arm64/boot/dts/tegra210-platforms/tegra210-jetson-e-power-fixed-p2530-0930-e03.dts
```

Troubleshooting

The following tips can help you troubleshoot the specified issue.

I2C transaction timeout error

- Is I2C information wrong?

  Check the sensor I2C bus number and the sensor I2C device address in the board file.

- Is sensor power control sequence wrong?

  Check sensor MCLK setting.

  Check regulator operations.

  Check GPIO settings.

Sync point timeout without error

This means Tegra VI/CSI does not receive any data but no error occurs. Verify that the sensor is powered on and streaming data correctly before debugging the Tegra driver.

Change settle time value to see if there if some error shows up. These registers must be configured with the right values to get data from the sensor.

```
cil_regs_write(vi2_cam, chan, TEGRA_CSI_CIL_PHY_CONTROL, 0xA);
```

Sync point timeout with error

Capture the error message and look it up in Tegra X1 TRM for further debugging.

Resources

Good resources for V4L integration are:

- Kernel documentation located in:

  ```
  Documentation/video4linux/
  ```

- Linux TV website:

  http://www.linuxtv.org/

- soc-camera slides:

  http://elinux.org/images/f/f2/Soc-camera.pdf

- Yavta user space V4L2 tool

  http://git.ideasonboard.org/yavta.git

- Jetson Embedded Platform page

  http://developer.nvidia.com/embedded-computing

Tegra ASoC Driver

DISCLAIMER:	This document describes hardware functionality present on Tegra X1 and Jetson TX1. Not all hardware functionality may be supported by the software. Review your software documentation to determine availability of software for audio features.

The NVIDIA® Tegra® ASoC driver is implemented for the Android and Linux operating systems and is intended to work seamlessly with different Tegra devices, using an existing framework called Advanced Linux Sound Architecture (ALSA), which is maintained by the upstream Linux community.

Note:	On Jetson TX1, there is no audio codec. Audio output is supported via HDMI audio. Audio input hardware in the development kit is NOT supported.

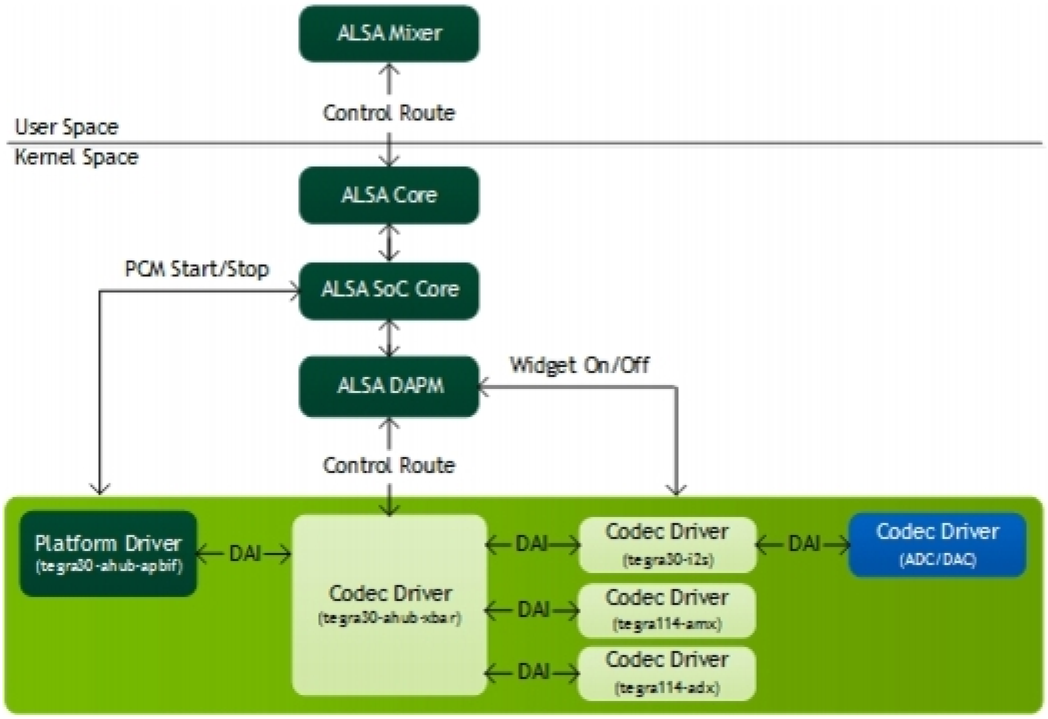
ALSA

The ALSA framework is a part of the Linux kernel that is supported and maintained by the larger Linux community. This makes it feasible to adapt to the framework, designing a driver that leverages NVIDIA audio routing support. ALSA has a very good collection of sound card drivers, including actual codec drivers for platform ADC and DAC support, and can support adding new codec drivers as well.

ALSA also includes libraries and utilities that enable more refined audio control in Linux and Android user space. These include alsamixer, aplay, arecord, tinycap, and others. This enables you to control audio applications without having to interact with kernel space drivers.

The following diagram shows the ALSA software hierarchy.

Tegra ASoC Driver Overview



User space ALSA applications interact with ALSA core (kernel space) through APIs provided by userspace libraries that initialize the actual hardware codecs at the back end of the audio pipeline.

More information on the ALSA framework is available at the following link:

```
http://www.alsa-project.org/main/index.php/Main_Page
```

DAPM

ALSA is designed to support various functionalities including but not limited to dynamic audio routing to the available PCM devices. The component of ALSA core that provides this support is Dynamic Audio Power Management (DAPM). DAPM controls the power flow into and out of various codec blocks in the audio subsystem, thereby minimizing power consumption. DAPM introduces switches or kernel controls in the form of widgets to turn ON/OFF the power of a module and help manipulate the required bit of the specific register dynamically using user space applications such as aplay, arecord, or alsamixer. The widgets are classified into various groups.

More information on the widgets and their applications is available at:

```
http://www.alsa-project.org/main/index.php/DAPM
https://www.kernel.org/doc/Documentation/sound/alsa/soc/DPCM.txt
```

In terms of software hierarchy, DAPM is part of the ALSA core as shown in the Tegra ASoC Driver Overview diagram, helping to manage the codec module power efficiently.

For details on the clocking and power management in Tegra ASoC driver, see Clocking and Power Management topic.

For more information see the Dynamic Audio Routing topic.

Device Tree

The Device Tree is a data structure which describes devices on the platform. It is passed to the operating system at boot time and allows you to avoid hard coding component details in the operating system. In this way it makes it easier to change hardware configurations without rebuilding the kernel.

The data structure contains the name of nodes and properties. Each node can have properties or child nodes, and each property is comprised of a name and more than one value. Device Tree structures must be written in the correct format and according to format rules so that the data structure can be parsed by the operating system.

More details on usage of DTS and its script format can be found at:

```
http://www.devicetree.org/Device_Tree_Usage
```

For a simple device tree example see the Codec Driver Instantiation via Device Tree section of this chapter.

Audio Driver

The Tegra Audio Driver leverages Tegra Audio Hub (AHUB) hardware acceleration in the form of platform and codec drivers. The ARM peripheral bus interface (ADMAIF) is implemented as a platform driver with PCM interfaces for playback/record and the rest of the AHUB modules such as the Audio Cross Bar (XBAR), Audio Multiplexer (AMX), Audio Demultiplexer (ADX) and Inter-IC sound (I2S) implemented as codec drivers. Each of the drivers is connected to XBAR through Digital Audio Interfaces (DAIs), inside a machine driver, forming an audio hub.

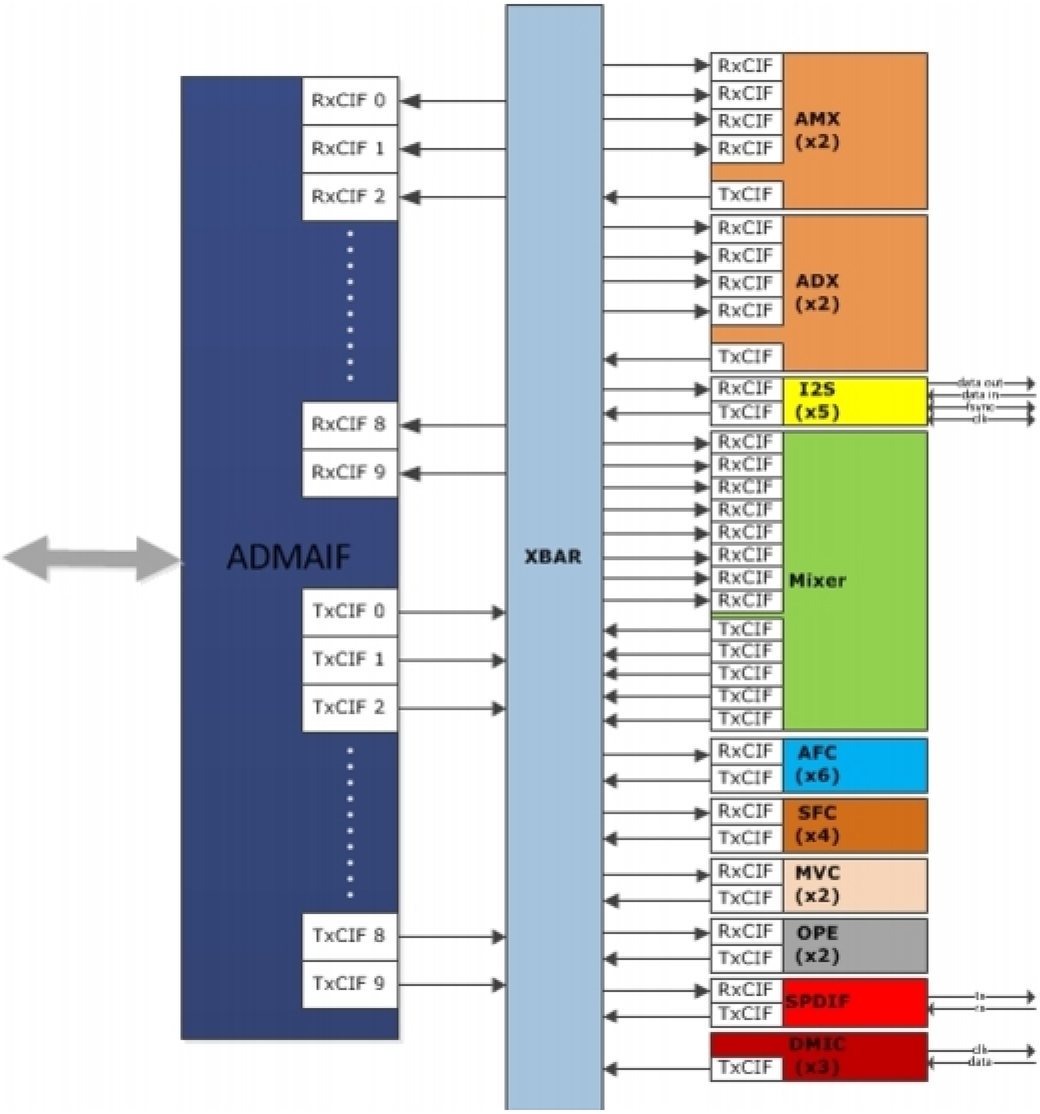
The machine driver probe instantiates the sound card device and registers all the PCM interfaces as exposed by ADMAIF. After booting, before we can use these interfaces to playback or record audio, you must set up the audio paths inside XBAR. By default, XBAR has no routing connections at boot, and no DAPM path is complete to power on the corresponding widgets. The XBAR driver introduces MUX widgets for all the audio components and enables you to create any custom routing through kcontrol from user space using the alsamixer utility. If the audio path is not complete, the DAPM path is not closed. Hardware settings are not applied and you might not hear any audio output.

For more details on how to set up the route and how to play or record on the PCM interfaces, see the Audio Playback/Record Examples section of this document.

Tegra Audio Hub

In this section, we provide an overview of the audio hub hardware architecture inside the Tegra SoC and describe the software architecture of the driver.

Tegra Audio Hub Architecture



The audio hub contains the modules I2S, SPDIF, and the Digital MIC Controller (DMIC) that interface with the external world. The audio hub also contains Mixer, Sampling Frequency Converter (SFC), Master Volume Control (MVC), Output Processing Engine (OPE), Audio Multiplexers (AMX), Audio Demultiplexers (ADX) and Audio Flow Controllers (AFC). An Audio Direct Memory Access (ADMA) component is included (ADMAIF-DMA) to communicate with memory. The Crossbar (XBAR) facilitates the routing of audio samples through these modules using a proprietary protocol called Audio Client Interface (ACIF).

The modules in the audio hub support various kinds of audio devices that are expected to interface with the application processor, such as cellular baseband devices, different types of audio CODECs, Bluetooth modules, and A/V receivers. The audio hub is capable of supporting the different interface and signal quality requirements of these devices.

As shown in the Tegra Audio Hub Architecture diagram, each of the AHUB modules has at least one RX port and one TX port, and some have more than one, depending on whether they are for duplex flow such as I2S. The RX ports feed off from XBAR and the TX ports are fed back into XBAR, except in the case of I2S where one end feeds of the actual codec. This configuration makes XBAR a switch where an audio input can be fed to multiple outputs depending on use case.

For dynamic audio routing examples see the Audio Playback/Record Examples topic.

Each ADMAIF has both TX and RX FIFOs that support simultaneous recording and playback. ADMA transfers the data to the ADMAIF FIFO for all audio routing scenarios, as shown in the Tegra Audio Hub Architecture diagram.

For details on hardware configuration of each module, refer to the *Tegra Reference Manual*.

Software Architecture

The software architecture of the Tegra ASoC driver is very similar to the hardware architecture of the AHUB itself, as shown in the Tegra Audio Hub Architecture diagram. This enables you to leverage all the features supported by the hardware and still conform to the ALSA framework. This ALSA System on a Chip (ASoC) driver is comprised of platform, codec and machine drivers.

- Platform driver–This driver is responsible for PCM registration and interfaces with the PCM driver. The ADMAIF is the platform driver.
- Codec driver–Any driver that registers snd_soc_codec_driver structure with the ASoC core can be viewed as the codec driver. The module must have at least one input and one output. In addition, the structure provides a way to define your own DAPM widgets for power management and also kcontrols for register setting from user space. All other modules except ADMAIF are implemented as codec drivers.
- Machine driver–This driver connects one or more codec drivers and a PCM driver together for a given platform.

For details on how to write a machine driver and identify a sound card, see the Tegra Machine Driver topic and the kernel documentation for AsoC framework at:

```
https://www.kernel.org/doc/Documentation/sound/alsa/soc/
```

Tegra Platform Driver

The Tegra platform driver realizes the number of ports of playback and capture possible inside the AHUB. Some or all of these ports can connected to form a full audio routing path. You must complete these paths as described in Audio Playback/Record Examples.

ADMAIF

In the Tegra ASoC driver, ADMAIF is implemented as a platform driver and interfaces with the PCM driver. The PCM driver helps perform DMA operations by overriding the function pointers exposed by the `snd_pcm_ops` structure. The PCM driver is platform agnostic and interacts only with the SOC DMA engine upstream APIs. The DMA engine then interacts with the platform specific DMA driver to get the correct DMA settings. The ADMAIF platform driver defines DAIs and registers the same with ASoC core.

In Tegra X1, there are 10 ADMAIFs. Each ADMAIF is bi-directional, facilitating 10 streams of playback and 10 streams of capture.

Playback Hardware Devices in the Tegra ASoC Driver

```
**** List of PLAYBACK Hardware Devices ****
card 0: tegrasndt210ref [tegra-snd-t210ref], device 0: ADMAIF1 CIF ADMAIF1-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 1: ADMAIF2 CIF ADMAIF2-1 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 2: ADMAIF3 CIF ADMAIF3-2 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 3: ADMAIF4 CIF ADMAIF4-3 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 4: ADMAIF5 CIF ADMAIF5-4 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 5: ADMAIF6 CIF ADMAIF6-5 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 6: ADMAIF7 CIF ADMAIF7-6 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 7: ADMAIF8 CIF ADMAIF8-7 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 8: ADMAIF9 CIF ADMAIF9-8 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 9: ADMAIF10 CIF ADMAIF10-9[]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 93: ADSP PCM ADSP-FE1-93 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Capture Hardware Devices in the Tegra ASoC Driver

```
**** List of CAPTURE Hardware Devices ****
card 0: tegrasndt210ref [tegra-snd-t210ref], device 0: ADMAIF1 CIF ADMAIF1-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 1: ADMAIF2 CIF ADMAIF2-1 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 2: ADMAIF3 CIF ADMAIF3-2 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 3: ADMAIF4 CIF ADMAIF4-3 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 4: ADMAIF5 CIF ADMAIF5-4 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 5: ADMAIF6 CIF ADMAIF6-5 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 6: ADMAIF7 CIF ADMAIF7-6 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 7: ADMAIF8 CIF ADMAIF8-7 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 8: ADMAIF9 CIF ADMAIF9-8 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 9: ADMAIF10 CIF ADMAIF10-9[]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 0: tegrasndt210ref [tegra-snd-t210ref], device 93: ADSP PCM ADSP-FE1-93 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Tegra Codec Driver

A small overview of codec drivers is presented in the Tegra Audio Hub section of this document. In the Tegra ASoC driver implementation, the rest of the AHUB modules, except ADMAIF, are implemented as codec drivers.

Their responsibilities include:

- Interface to other modules by defining DAIs.
- Define DAPM widgets and establish DAPM routes for dynamic power switching.
- Expose additional kcontrols (kernel controls) as needed for user space utilities to dynamically control module behavior.

XBAR

The XBAR codec driver defines RX, TX and MUX widgets for all the interfacing modules such as ADMAIF, AMX, ADX, I2S, SPDIF, DMIC, Mixer, SFC, MVC, OPE and AFC. MUX widgets are permanently routed to the corresponding TX widgets inside the `snd_soc_dapm_route` structure.

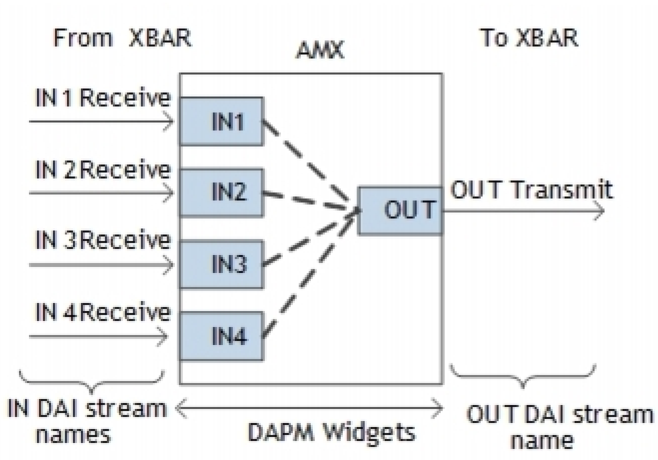
However, the XBAR interconnections are made by connecting any RX widget block to any MUX widget block as needed using the alsamixer utility. The get/put handlers for these widgets are implemented so that audio connections are stored by setting the appropriate bit in the hardware MUX register.

For more information see the Audio Playback/Record Examples topic and the Dynamic Audio Routing topic.

AMX

An Audio Multiplexer (AMX) module can multiplex up to 4 streams of 16 channels, 32 bits each, into one time-division multiplexed (TDM) stream of 16 channels and 32 bits. The 4 RX ports of AMX originate from XBAR and 1 TX port feeds into XBAR. The DAPM widgets exposed are as shown in the following diagram. The DAPM routes established using these widgets are shown in the dotted lines.

AMX Codec Driver Internals

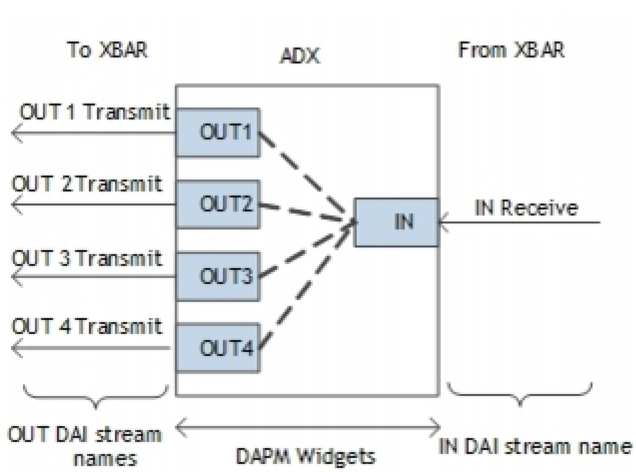


ADX

ADX Codec Driver Internals

An Audio Demultiplexer (ADX) module can de-multiplex a single TDM stream of 16 channels and 32 bits into 4 streams of up to 16 channels, 32 bits each. The 1 RX port of ADX originates from XBAR and 4 TX ports feeds into XBAR. The DAPM widgets exposed are as shown in the following diagram. The DAPM routes established using these widgets are shown in the dotted lines.

ADX Codec Driver Internals

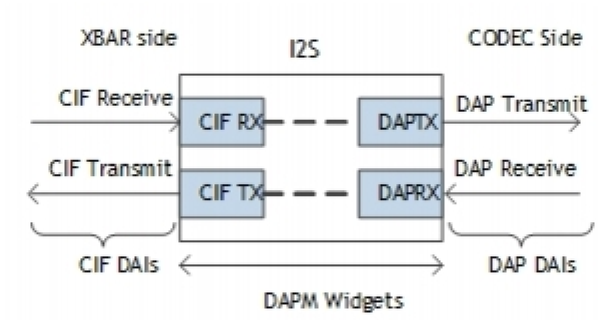


I2S

I2S Codec Driver Internals

An I2S module supports 6 different modes, the most useful of which are I2S and TDM. When I2S is configured in TDM mode, we can use AMX and ADX for multiplexing and demultiplexing audio streams. I2S module can also be in I2S mode if the physical codec interfaced does not support TDM mode of operation. An I2S codec driver supports bidirectional data flow and thus defines CIF and DAP RX/TX widgets as shown in the following diagram. The CIF side of I2S, interfaces with XBAR and DAP side is meant to interface with the physical codec on the given platform. The DAPM routes established using these widgets are shown in the dotted lines. I2S modules also expose kernel control to enable internal I2S loopback.

I2S Codec Driver Internals

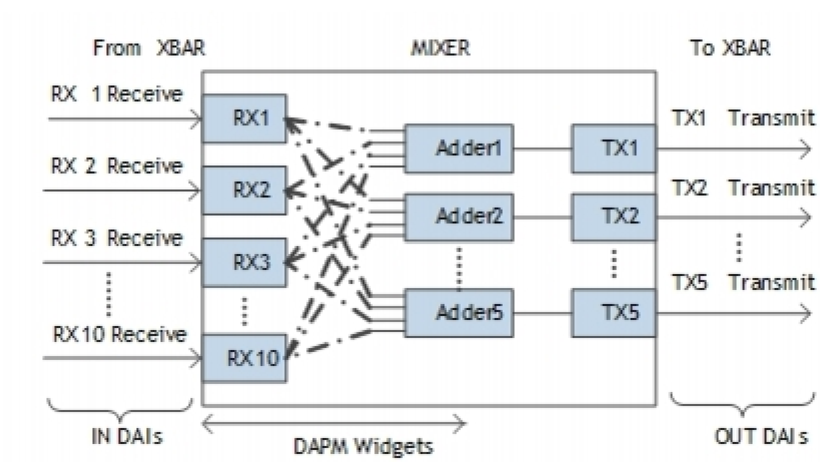


Mixer

Mixer Codec Driver Internals

The Mixer can mix audio streams from any of the 10 XBAR-originating input ports to any of the 5 output ports. The DAPM widgets and routes for Mixer are shown in the following diagram. The Mixer driver also exposes Rx Gain and Mixer Enable as additional kcontrols to set the volume of each input stream and to globally enable or disable the Mixer itself.

Mixer Codec Driver Internals

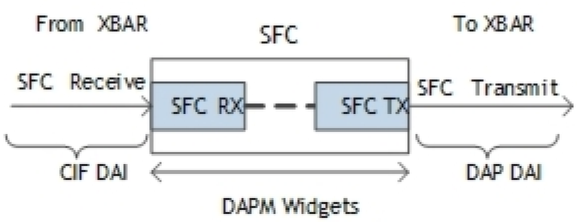


SFC

SFC Codec Driver Internals

The Sampling Frequency Converter(SFC) can convert the input sampling frequency to the required sampling rate. SFC has one input port and one output port which are connected to XBAR. The DAPM widgets and routes for SFC are shown in the following diagram.

SFC Codec Driver Internals

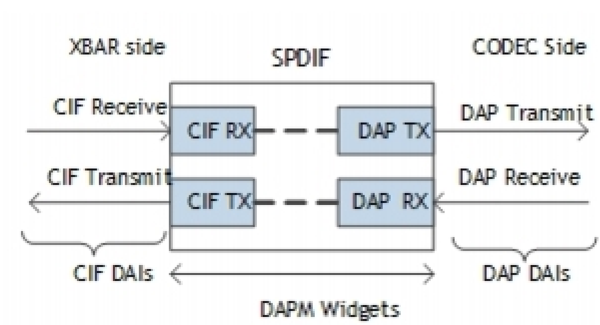


SPDIF

SPDIF Codec Driver Internals

The SPDIF driver is very similar to I2S in terms of the DAPM widgets that are exposed and the routing setup. SPDIF is also bidirectional and adapts to the sampling rate of the incoming data. It is implemented as described in the following diagram.

SPDIF Codec Driver Internals

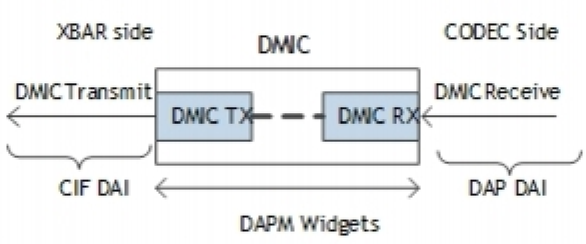


DMIC

DMIC Codec Driver Internals

It is beneficial to interface directly to digital microphones and speakers via pulse-density modulation (PDM), thus avoiding the need for a PDM-capable external codec. The DMIC controller implements a converter to convert PDM (Pulse density modulation) signals to PCM (Pulse code modulation) signals. The DAPM widgets and routes are as shown below.

DMIC Codec Driver Internals

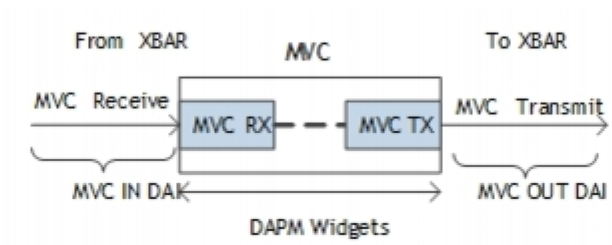


MVC

MVC Codec Driver Internals

MVC provides gain or attenuation to a digital signal path. The digital volume control block is a generic block. It can be used in input or output digital signal path. It can also be used for per-stream volume control and master volume control. The DAPM widgets and routes as shown below.

MVC Codec Driver Internals

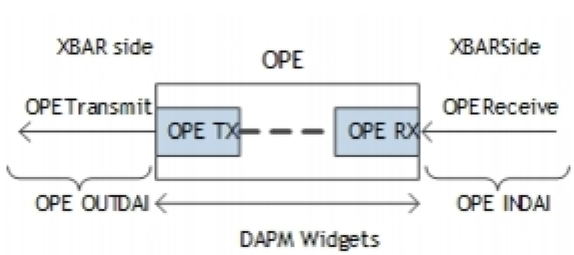


OPE

OPE Codec Driver Internals

The Output Processing Engine (OPE) is a client of AHUB. OPE contains PEQ and MBDRC which have a scalable number of BiQuad stages, support stereo, 5p1 and 7p1 channels, and meet ultra-low power(ULP) audio requirements. The DAPM widgets and routes are shown in the following diagram.

OPE Codec Driver Internals



Tegra Machine Driver

Tegra X1

The Tegra machine driver connects the codec drivers by linking the DAIs exposed by each module described in previous chapter. It defines the `snd_soc_dai_link` structure and instantiates the sound card.

In general, machine driver responsibilities include:

- Populate the `snd_soc_dai_link` structure with appropriate CPU and CODEC DAIs
- Physical codec clock setting (if any) and codec initializations
- Master/slave configurations (if any)
- Define DAPM widgets to route through the physical codec internals and complete DAPM path as needed
- Propagate the runtime sampling frequency to the individual codec drivers as needed

This section describes how to use the generic machine driver for a given platform and instantiate a sound card.

To adapt the generic machine driver template to a given platform, you must initialize the platform data structure in the device tree structure file shown below. It is important to gather some information about the hardware platform. You must identify the physical codecs present on board and the I2S instance they interface with in the hardware. Refer to hardware schematics for the platform to obtain this information.

```
sound_ref {
         compatible = " ";
         nvidia,model = " ";
         nvidia,num-codec-link = < >;
         nvidia,num-amx = < >;
         nvidia,num-adx = < >;
         nvidia,amx-slot-size = < >;
         nvidia,adx-slot-size = < >;
         nvidia,amx-slot-map = < >;
         nvidia,adx-slot-map = < >;

         nvidia,audio-routing = ;
         nvidia,xbar = <&tegra_axbar>;

         nvidia,dai-link-1 {
               link-name = " ";
               cpu-dai = < >;
               codec-dai = < >;
               cpu-dai-name = " ";
               codec-dai-name = " ";
```

```
                tx-mask = < >;
                rx-mask = < >;
                format = " ";
                bitclock-slave;
                frame-slave;
                bitclock-noninversion;
                frame-noninversion;
                bit-format = " ";
                bclk_ratio = < >;
                srate = < >;
                num-channel = < >;
                name-prefix = " ";
          };
    };
```

The `sound` node is added to the DT file for sound card registration and passing platform related data. To bind a device driver to a specific device in a DT file, each node must define a `compatible` property, which is a list of strings. The optional `nvidia,model` property is used for the sound card name. The `nvidia,num-codec-link` property shows the number of dai links exposed outside of the Tegra devices such as I2S, SPDIF or DMIC. This is the same as the number of `nvidia,dai-link-<number>` child nodes.

In Tegra X1 devices, each AMX and ADX has two modules, nvidia,num-amx and nvidia,num-adx. These indicate the total number of AMX and ADX to be registered by the machine driver. The nvidia,amx-slot-size and nvidia,adx-slot-size properties indicate the slot map size of each module, which is related to the size of nvidia,amx-slot-map and nvidia,adx-slot-map. The machine driver uses the default value for each property to support stereo 16 bits while initiating AMX and ADX modules unless the above properties are overridden.

The `nvidia,dai-link-<number>` node is initialized based on I2S and physical codec links as shown above. You can replace the codec with a dummy `spdif-dit.0` codec. In that case, you must program the physical codecs at the required sampling rate and operational modes either through a separate utility or by adding necessary `hw_params` API calls inside the machine driver. The `link-name` property represents each DAI link, and differentiates the codecs connected to the different I2S instances.

The bitclock-slave, frame-slave, format, bitclock-noninverstion and frame-noninversion properties set I2S in I2S/TDM formats and/or master/slave modes. The bclk-ratio property is available to configure the I2S bit clock sample rate. The srate and num-channel properties indicate I2S LRCK and the number of channels in one frame. DAPM routes are initialized as established in nvidia,audio-routing property.

To populate the snd_soc_dai_link structure in the machine driver

1. Initialize all XBAR related DAI links common to any machine driver. Use the APIs in the utility file tegra_asoc_machine_alt.h in the machine driver probe function as shown in the following example:

```
/* get the xbar dai link structure */
tegra_machine_dai_links=
 tegra_machine_get_dai_link();


/* set AMX/ADX dai_init */
```

```
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_AMX1,
 &tegra_t210ref_amx1_dai_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_ADX1,
 &tegra_t210ref_adx1_dai_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_AMX2,
 &tegra_t210ref_amx2_dai_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_ADX2,
 &tegra_t210ref_adx2_dai_init);

/* set sfc dai_init */
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_SFC1_RX,
 &tegra_t210ref_sfc1_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_SFC2_RX,
 &tegra_t210ref_sfc2_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_SFC3_RX,
 &tegra_t210ref_sfc3_init);
tegra_machine_set_dai_init(TEGRA210_DAI_LINK_SFC4_RX,
 &tegra_t210ref_sfc4_init);
```

2. The generic machine driver handles DAI link management based on the initialized platform specific data in the DT file.

   Because this step is platform/machine dependent, it is discussed with an example in subsequent sections of this chapter.

Tegra X1

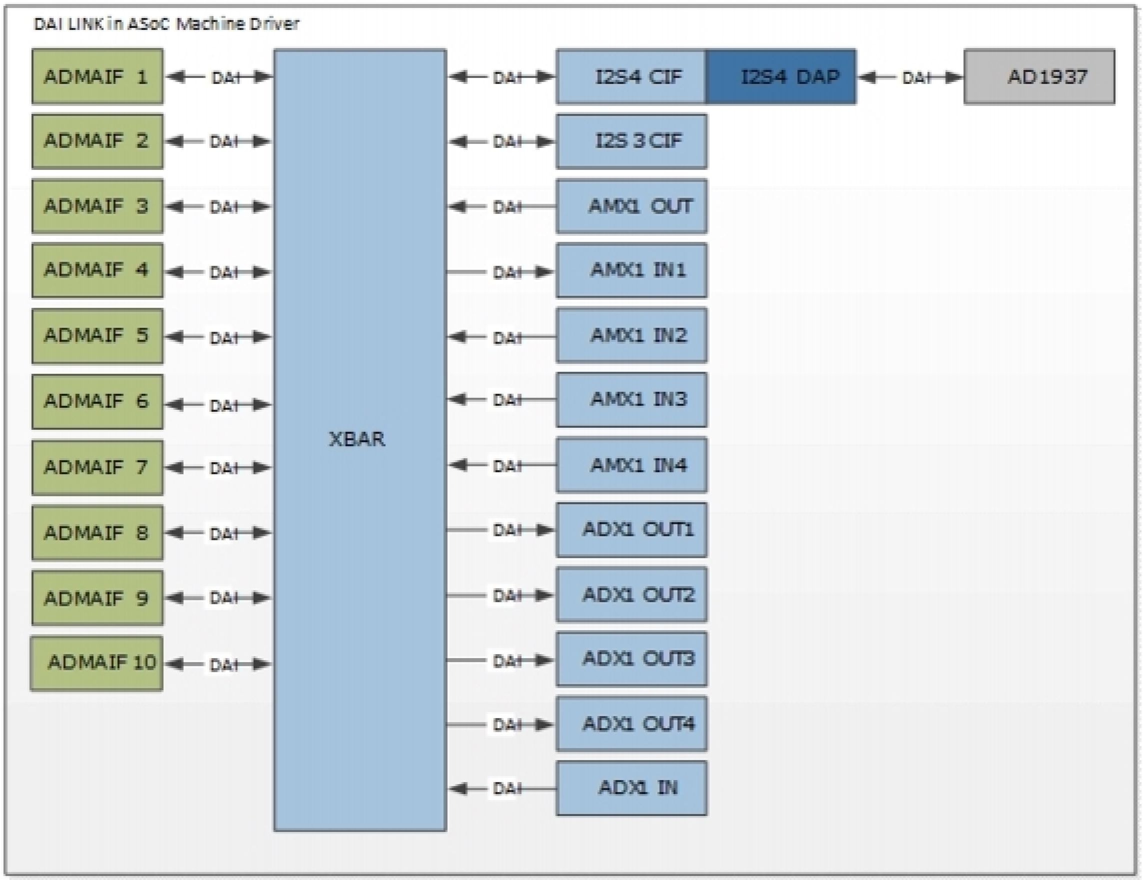
In the Tegra X1 combination of boards, there is one available AD1937s codec. AD1937 supports eight channels in TDM and I2S modes, respectively. The AD1937 codec is connected to I2S4 on P1892.

Machine Specific DAI links

The platform data structure must be initialized in the DT file so that the machine driver can retrieve the device specific information during the platform probe. The illustration below shows the DAI link diagram for PEDP+ with Tegra X1 where the left side of each DAI is CPU DAI and right side of each DAI is codec DAI in snd_soc_dai_link.

Note:	The following diagram shows the DAI LINK for PEDP+ platform; check the Release Notes for platforms supported in your release.

DAI LINK In the Machine Driver of PEDP with Tegra X1




```
sound_ref {
 compatible = "nvidia,tegra-audio-t210ref";
 nvidia,model = "tegra-snd-t210ref";
 nvidia,num-codec-link = <1>;
 nvidia,num-amx = <1>;
 nvidia,num-adx = <1>;
 nvidia,amx-slot-size = <32 32>;
 nvidia,adx-slot-size = <32 32>;
 nvidia,addr-max9485 = <112>;
 nvidia,amx-slot-map = <
  /* jack 0 */
  TDM_SLOT_MAP(0, 0, 0)
  TDM_SLOT_MAP(0, 0, 0)
  TDM_SLOT_MAP(0, 1, 0)
  TDM_SLOT_MAP(0, 1, 1)
  TDM_SLOT_MAP(0, 0, 0)
  TDM_SLOT_MAP(0, 0, 0)
```

```
 TDM_SLOT_MAP(0, 2, 0)
 TDM_SLOT_MAP(0, 2, 1)
 /* jack 1 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(1, 1, 0)
 TDM_SLOT_MAP(1, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(1, 2, 0)
 TDM_SLOT_MAP(1, 2, 1)
 /* jack 2 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(2, 1, 0)
 TDM_SLOT_MAP(2, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(2, 2, 0)
 TDM_SLOT_MAP(2, 2, 1)
 /* jack 3 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(3, 1, 0)
 TDM_SLOT_MAP(3, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(3, 2, 0)
 TDM_SLOT_MAP(3, 2, 1)>;
nvidia,adx-slot-map = <
 /* jack 0 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 1, 0)
 TDM_SLOT_MAP(0, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
```

```
 TDM_SLOT_MAP(0, 2, 0)
 TDM_SLOT_MAP(0, 2, 1)
 /* jack 1 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(1, 1, 0)
 TDM_SLOT_MAP(1, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(1, 2, 0)
 TDM_SLOT_MAP(1, 2, 1)
 /* jack 2 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(2, 1, 0)
 TDM_SLOT_MAP(2, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(2, 2, 0)
 TDM_SLOT_MAP(2, 2, 1)
 /* jack 3 */
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(3, 1, 0)
 TDM_SLOT_MAP(3, 1, 1)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(0, 0, 0)
 TDM_SLOT_MAP(3, 2, 0)
 TDM_SLOT_MAP(3, 2, 1)>;

nvidia,audio-routing =
 "Headphone-z",  "z DAC1OUT",
 "Headphone-z",  "z DAC2OUT",
 "Headphone-z",  "z DAC3OUT",
 "Headphone-z",  "z DAC4OUT",
 "z ADC1IN", "LineIn-z";
```

```
 nvidia,xbar = <&tegra_axbar>;

 nvidia,dai-link-1 {
  link-name = "ad-playback-z";
  cpu-dai = <&tegra_i2s4>;
  codec-dai = <&ad1937z>;
  cpu-dai-name = "I2S4";
  codec-dai-name = "ad193x-hifi";
  tx-mask = <0xFF>;
  rx-mask = <0xFF>;
  format = "dsp_a";
  bitclock-slave;
  frame-slave;
  bitclock-noninversion;
  frame-noninversion;
  bit-format = "s32_le";
  bclk_ratio = <1>;
  srate = <48000>;
  num-channel = <8>;
  name-prefix = "z";
 };
};
```

The `dai_fmt` sets I2S4 in TDM master mode. It also connects codec AD1937 with I2S4 DAPM routes and AMX/ADX slot maps initialized. The sound card is named `tegra-snd-t210ref`.

The DAPM widgets supported in the generic machine driver include Headphone-x, Headphone-y, Headphone-z, Headphone-s, LineIn-x, LineIn-y, LineIn-z, and LineIn-s. Based on how `name_prefix` and `dai_fmt` are initialized, the DAPM route must be set by parsing the IN/OUT DAPM widgets of the physical codec. Without this, the DAPM path is not complete and audio does not function correctly. The Tegra X1 audio map is shown below.

```
nvidia,audio-routing =
  "Headphone-z",  "z DAC1OUT",
  "Headphone-z",  "z DAC2OUT",
  "Headphone-z",  "z DAC3OUT",
  "Headphone-z",  "z DAC4OUT",
  "z ADC1IN", "LineIn-z",
```

Though we can assign any ADMAIF to any AMX via XBAR, we can fix ADMAIF1 to ADMAIF4 as the inputs of AMX1 are connected to I2S4.

The machine driver creates three `snd_soc_dai_link`s for each `dai-link-<number>` node initialized in the DT file and appends them to XBAR DAI links using the utility API. It also initializes the `.dai_ops` for the ADMAIF interfaces accordingly as shown below.

```
/* set ADMAIF dai_ops */
for (i = TEGRA210_DAI_LINK_ADMAIF1;
 i <= TEGRA210_DAI_LINK_ADMAIF10; i++)
 tegra_machine_set_dai_ops(i, &tegra_t210ref_spdif_ops);

for (i = 0; i < machine->num_codec_links; i++) {
 if (tegra_t210ref_codec_links[i].name) {
  if (strstr(tegra_t210ref_codec_links[i].name,
   "ad-playback-z")) {
   for (j = TEGRA210_DAI_LINK_ADMAIF1;
    j <= TEGRA210_DAI_LINK_ADMAIF4; j++)
    tegra_machine_set_dai_ops(j,
     &tegra_t210ref_ad1937_z_ops);
  } else if (strstr(tegra_t210ref_codec_links[i].name,
   "ad-playback-x")) {
   for (j = TEGRA210_DAI_LINK_ADMAIF5;
    j <= TEGRA210_DAI_LINK_ADMAIF8; j++)
    tegra_machine_set_dai_ops(j,
     &tegra_t210ref_ad1937_x_ops);
  } else if (strstr(tegra_t210ref_codec_links[i].name,
   "spdif-playback")) {
   tegra_machine_set_dai_ops(
    TEGRA210_DAI_LINK_ADMAIF9,
    &tegra_t210ref_spdif_ops);
  }
 }
}

/* append t210ref specific dai_links */
card->num_links =
 tegra_machine_append_dai_link(tegra_t210ref_codec_links,
  2 * machine->num_codec_links);
```

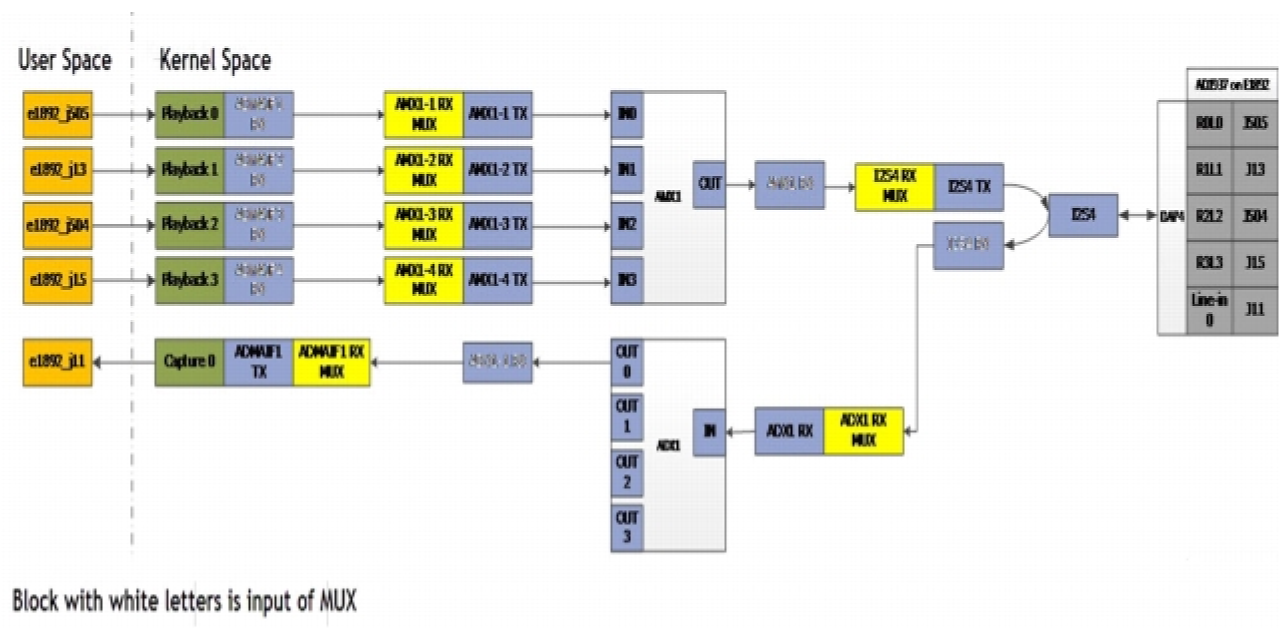
```
tegra_machine_dai_links = tegra_machine_get_dai_link();
card->dai_link = tegra_machine_dai_links;
```

Audio Path

The dai_link connection in the machine driver only connects the platform and codec drivers to XBAR driver. However, to realize any audio output from the physical codecs, you must complete the DAPM path by routing through the internals of XBAR, i.e., connect the RX and MUX widgets.from user space using alsamixer utility as follows.

> **Note**: The following commands are specific to Tegra X1 and represent the audio path shown in the following diagram. This diagram shows the DAI LINK for PEDP+ platform; check the *Release Notes* for platforms supported in your release.

Tegra X1 Audio Path



XBAR Route Setting forTegra X1

```
amixer -c 0 sset 'I2S4 Mux' 'AMX1'
amixer -c 0 sset 'AMX1-1 Mux' 'ADMAIF1'
amixer -c 0 sset 'AMX1-2 Mux' 'ADMAIF2'
amixer -c 0 sset 'AMX1-3 Mux' 'ADMAIF3'
amixer -c 0 sset 'AMX1-4 Mux' 'ADMAIF4'
amixer -c 0 sset 'ADX1 Mux' 'I2S4'
amixer -c 0 sset 'ADMAIF1 Mux' 'ADX1-1'
```

Dynamic Audio Routing

Instantiation of the sound card after boot indicates that all codec drivers and the platform driver are interconnected using the machine driver. The remaining step before obtaining the audio output on the physical codecs involves routing through the XBAR internals via MUX widgets to complete the DAPM path using the alsamixer utility. To support audio routing dynamically between AHUB modules, this route setting step is performed from user space. This provides flexibility for complicated use cases.

For example, `[amixer –c 0 sset ‘I2S1 Mux’ ‘I2S1’]` realizes the internal AHUB path I2S1 RX -> XBAR -> I2S1 TX. Similarly, the following are use cases that emphasize dynamic audio route control from user space.

Case 1: Internal AHUB TDM Path

Path: I2S -> ADX -> AMX -> I2S

Commands:

```
amixer –c 0 sset ‘ADX1 Mux’ ‘I2S4’
amixer –c 0 sset ‘AMX1-1 Mux’ ‘ADX1-1’
amixer –c 0 sset ‘AMX1-2 Mux’ ‘ADX2-2’
amixer –c 0 sset ‘AMX1-3 Mux’ ‘ADX3-3’
amixer –c 0 sset ‘AMX1-4 Mux’ ‘ADX4-4’
amixer -c 0 sset 'I2S4 Mux' 'AMX1'
```

Modify Case 1 to Record on I2S3 (I2S Mode) And Output On I2S4 (TDM Mode)

Path: [I2S -> ADX -> AMX -> I2S] + [I2S3 -> AMX -> I2S4]

Initially audio records from I2S4 and outputs on I2S4.

Commands:

```
amixer –c 0 sset ‘AMX1-1 Mux’ ‘None’
amixer –c 0 sset ‘AMX1-1 Mux’ ‘I2S3’
```

After this step, audio records from I2S3 and outputs on I2S4

Codec Driver Instantiation via Device Tree

Based on architecture, the Makefile in the following directory conditionally compiles the required DTS files into DTB files:

```
$KERNEL_TOP/arch/arm64/boot/dts/
```

When the kernel is flashed, the flash script chooses the required DTS file for parsing during boot, and the ASoC codecs listed in DTS are instantiated. To add any new module instantiation as a requirement, identify and edit the Device Tree script as reported in the `dmesg` log on the target as shown in the following example:

```
<6>[    0.000000] Tegra reserved memory:
<6>[    0.000000] LP0:                    f25ff000 - f25fffff
<6>[    0.000000] Bootloader framebuffer: 00000000 - 00000000
<6>[    0.000000] Bootloader framebuffer2: 00000000 - 00000000
<6>[    0.000000] Framebuffer:            ef800000 - f09fffff
<6>[    0.000000] 2nd Framebuffer:        f0a00000 - f19fffff
<6>[    0.000000] Carveout:               00000000 - 00000000
<6>[    0.000000] Vpr:                    f4600000 - ffffffff
<6>[    0.000000] Tsec:                   00000000 - 00000000
<7>[    0.000000] On node 0 totalpages: 980992
<7>[    0.000000]   DMA32 zone: 6244 pages used for memmap
<7>[    0.000000]   DMA32 zone: 0 pages reserved
<7>[    0.000000]   DMA32 zone: 456704 pages, LIFO batch:31
<7>[    0.000000]   Normal zone: 7168 pages used for memmap
<7>[    0.000000]   Normal zone: 524288 pages, LIFO batch:31
<6>[    0.000000] psci: probing function IDs from device-tree
<6>[    0.000000] DTS File Name: /home/juskim/git/android/t210_main/kernel/arch/arm64/b
<6>[    0.000000] Tegra21: Speedo/IDDQ fuse revision 0
```

To add new devices for instantiation

- Add the device name with the base address and status as “okay”, as shown in the following example.

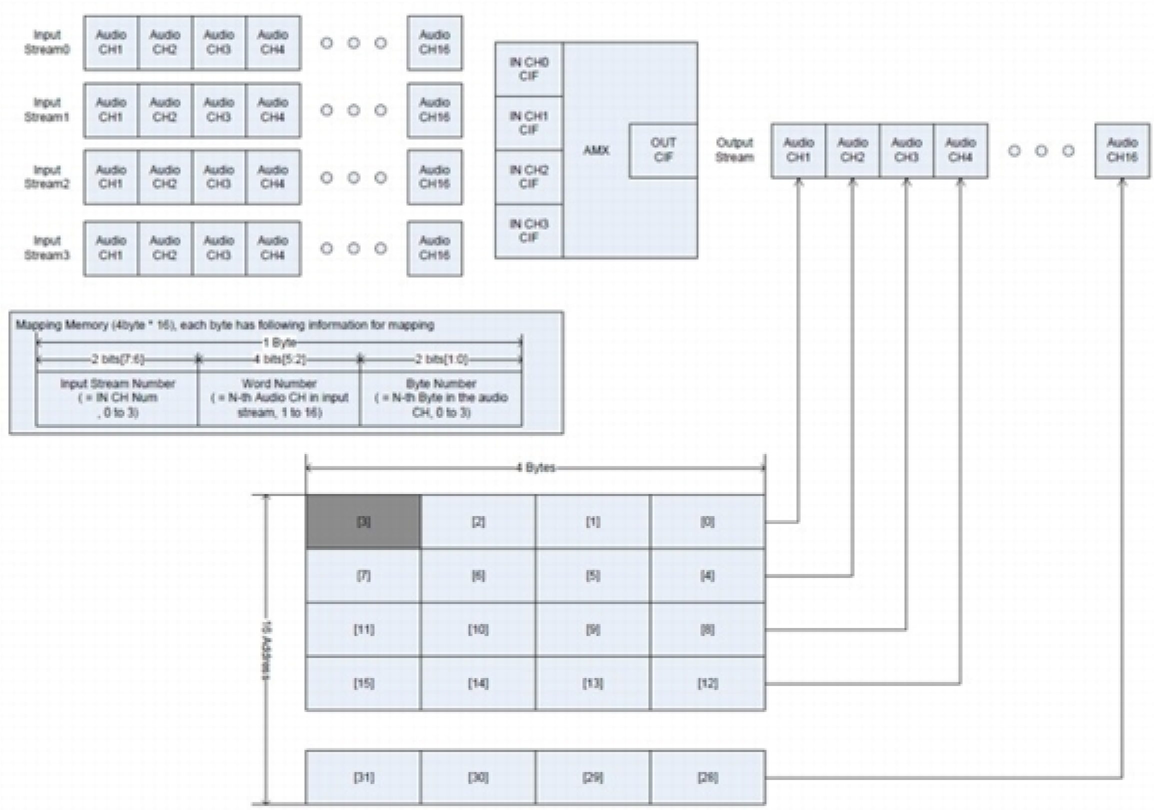
```
ahub {
 status = "okay";
 i2s@702d1000 {
  pinctrl-names = "dap_active", "dap_inactive";
  pinctrl-0 = <>;
  pinctrl-1 = <>;
  status = "okay";
 };
};
```

TDM Slot Mapping

The Tegra X1 AHUB consists of 2 hardware audio multiplexers (AMX) and demultiplexers (ADX) as described in Tegra Audio Hub. Initialize slot mapping in the platform data as shown inMachine Specific DAI links. Without any initialization, the machine driver defaults the slot map for AMX inputs and ADX outputs at 16-bit stereo and

8-channel, 32 bit TDM output. To change the slot mapping of hardware TDMs, adjust the slot map structures in the DT file.

As shown in below, AMX can multiplex up to 4 input streams, each of which can be up to 16 channels, 32 bits, into 1 single output stream of 16 channels, 32 bits to form a TDM signal. The capability of ADX is the reverse of AMX.



The slot map is an integer array of size 32 that controls the mapping of these input streams to the output stream. The following macro is available:

```
#define TDM_SLOT_MAP(stream_id, nth_channel, nth_byte)     \
        ((stream_id << 16) | (nth_channel << 8) | (nth_byte))
```

where `stream_id` can vary from 0 to 3, `nth_channel` can vary from 1 to 16, and `nth_byte` can vary from 0 to 3.

Clocking and Power Management

The clock tree of the Tegra ASoC driver in the idle state, when no audio record or playback is in progress, is as shown below.

```
clock                                   state   ref div   rate
----------------------------------------------------------------------
i2s4_sync                               on      1         24000000
i2s3_sync                               on      1         24000000
```

```
 *audio3                            off    0        24000000
i2s2_sync                           on     1        24000000
i2s1_sync                           on     1        24000000
 *audio1                            off    0        24000000
i2s0_sync                           on     1        24000000
spdif_in_sync                       on     1        24000000
 *audio2_dmic                       off    0        24000000
 *audio1_dmic                       off    0        24000000
 *audio0_dmic                       off    0        24000000
 *audio                             off    0        24000000
 *audio_2x                        off    0  x2    48000000
 *audio4                            off    0        24000000
 *audio2                            off    0        24000000
 *audio0                            off    0        24000000
osc                                 on     3        38400000
  pll_ref                           on     7  1.0   38400000
   pll_a                          on     1  x9.5  368639844
  pll_a_out0                 on     1  15.0  24575990
     extern1                 on     3  1.0   24575990
     clk_out_1          on     2  1.0   24575990
     d_audio                 off    0  2.0   12287995
    *dmic3                   off    0  11.0  2234181
    *dmic2                   off    0  11.0  2234181
    *dmic1                   off    0  11.0  2234181
     spdif_out               off    0  21.0  1170286
     i2s4                    off    0  5.5   4468362
     i2s3                    off    0  5.50  4468362
     i2s1                    off    0  5.50  4468362
     i2s0                    off    0  10.50 2340571
     pll_p                          on     15 x10.6 408000000
  *spdif_in                  off    0  8.50  48000000
  ape                        off    0  2.0   204000000
     xbar.ape                off    0        204000000
     adsp.ape                off    0        204000000
     adma.ape                off    0        204000000
```

The clocks of the individual modules, AMX, ADX, AFC, SFC, MIXER, and others, are internally handled by the APE clock. The clock for the codec drivers I2S and XBAR are switched OFF in idle. They are turned ON when audio playback or recording is in progress. The `idle_bias_off` option provided by ASoC core (`snd_soc_codec_driver`) is set to 1 in the individual codec drivers for dynamic audio power management. With this option, the ASoC core calls the appropriate `resume()` and `suspend()` functions in the codec driver that was registered using `SET_RUNTIME_PM_OPS` during platform probe.

The suspend and resume APIs are called when playback or record stops or starts. The suspend API caches the register map and disables the clock. The resume API enables the clock and then syncs the register map from the cache.

Note: Volatile registers are not synced in this process. Those registers must be reprogrammed after the clock is re-enabled. The internal mapping RAM FIFOs, if any, are cleared during every suspend/resume cycle. The RAM configuration must be restored as well during this process.

Audio Playback/Record Examples

To test audio playback and record scenarios, make sure the XBAR route is set up using alsamixer commands, based on the platform. The following provides the sample testing commands for audio playback and record scenarios.

Command	Result
aplay –D AUDIO_OUT0 sample.wav	Audio output from AUDIO_OUT0 on e1892.
aplay –D AUDIO_OUT1 sample.wav	Audio output from AUDIO_OUT1 on e1892.
aplay –D AUDIO_OUT2 sample.wav	Audio output from AUDIO_OUT2 on e1892.
aplay –D AUDIO_OUT3 sample.wav	Audio output from AUDIO_OUT3 on e1892.
arecord -D AUDIO_IN -f dat -d 30 record_result.wav	Audio recorded as input from AUDIO_IN on E1892 and saved to file record_result.wav.

Troubleshooting

The following issues can be overcome using the described solutions.

Issue 1: No Sound Cards Found

- Identify the ASoC error that lead to no sound card detection with the dmesg [dmesg | grep ASoC] command. Output is similar to the following:

```
[4.874720] tegra-audio-t210ref tegra-audio-t210ref.0: ASoC: no source widget found for
[4.874724] tegra-audio-t210ref tegra-audio-t210ref.0: ASoC: Failed to add route x OUT->
[4.874736] tegra-audio-t210ref tegra-audio-t210ref.0: ASoC: no sink widget found for x
[4.874739] tegra-audio-t210ref tegra-audio-t210ref.0: ASoC: Failed to add route LineIn-
```

  In this case, `x OUT` and `x IN` are the widgets of the `spdif-dit` dummy codec. It might have not been instantiated in ASoC. Confirm that with the following command:

```
cat /sys/kernel/debug/asoc/codecs
```

if `spdif-dit` is instantiated. The `spdif` device must be instantiated using `platform_register` in the board-specific file.

- If output from the `dmesg [dmesg | grep ASoC]` command is similar to the following:

```
[4.874720] tegra-audio-t210ref tegra-audio-t210ref.0: ASoC: CPU DAI DAP not registered
```

In this case, "DAP" is the CPU DAI for the I2S to codec dai link. The I2S codec may not be instantiated in ASoC. Confirm that with the following command:

```
cat /sys/kernel/debug/asoc/codecs
```

if tegra30-i2s is instantiated.

Identification of the DAI link at the point of failure would give a clue on the I2S instance number that failed to instantiate. Accordingly, the I2S codec driver can be instantiated by providing a suitable entry point in DTS file as described in Codec Driver Instantiation via Device Tree.

Issue 2: Sound Not Audible

1. Confirm if the DAPM path is completed. Check the path using alsamixer utility as described in Audio Path.
2. Confirm the pinmux setting for I2S master/slave mode. Check the following files.

```
arch/arm64/boot/dts/tegra210-foster-e-p2530-common.dtsi
arch/arm64/boot/dts/tegra210-platforms/tegra210-foster-e-pinmux-p2530-0930-e00.dtsi
```

3. Confirm the `status` property of i2s is set to `okay` in the following file:

```
arch/arm64/boot/dts/tegra210-common.dtsi
```

4. Confirm the clock settings for I2S master/slave mode. Probe the Frame sync (FS) and bit clock (BCLK) of I2S using a scope.
5. If I2S is configured in TDM mode, please check whether the AMX slot map is configured correctly based on the description in TDM Slot Mapping.
6. You can also form an internal DAPM path connecting I2S with the same I2S instance using the following command:

```
amixer –c 0 sset 'I2S Mux' 'I2S'
```

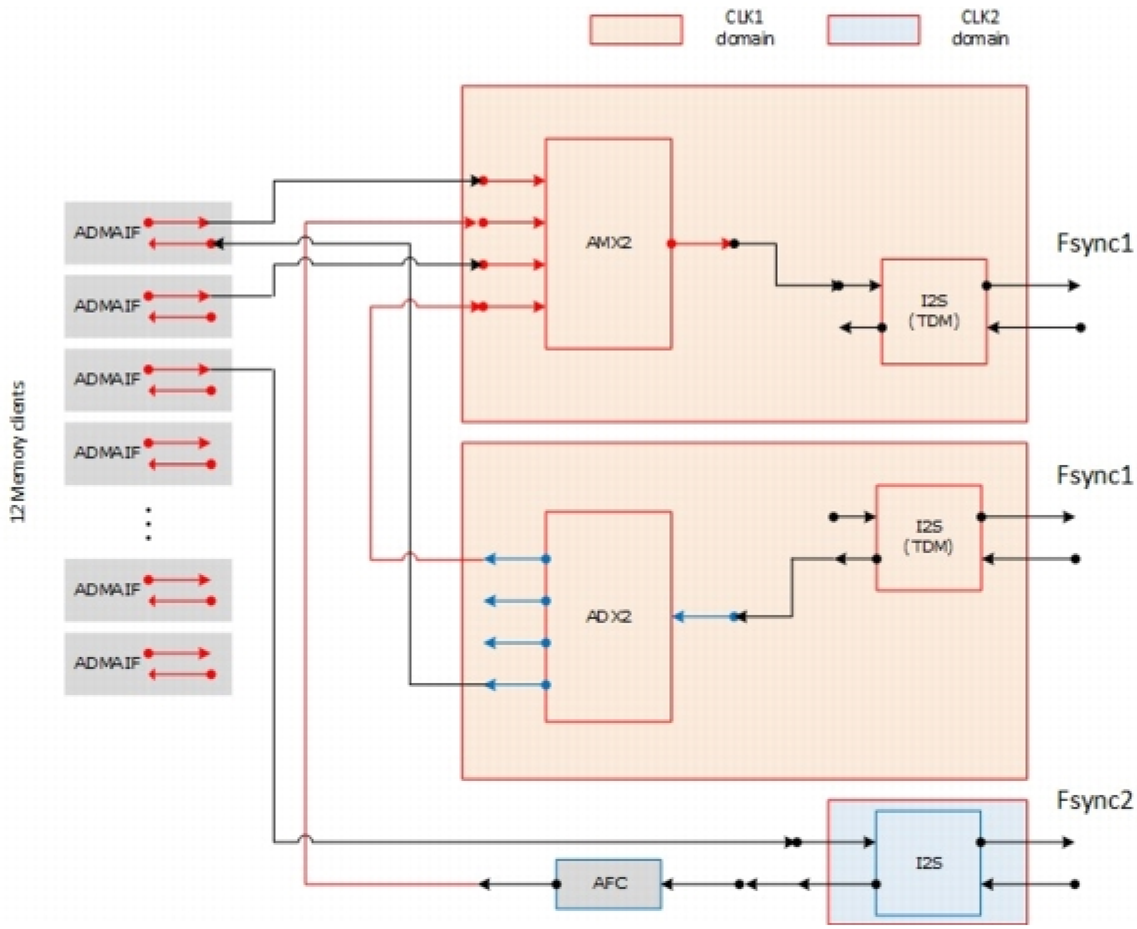
   If the codec record/playback functions correctly, that confirms I2S and onboard codecs are configured correctly. It also isolates the problem to rest of the AHUB path.

Miscellaneous Examples

This section describes setting up the XBAR internal audio path from user space for complicated Tegra use cases.

Simple Internal Audio Path

Routing Commands Testing Commands



Routing Commands

- AMX connection:

```
amixer -c 0 sset 'AMX2-1 Mux' 'ADMAIF5'
amixer -c 0 sset 'AMX2-2 Mux' 'AFC1'
amixer -c 0 sset 'AMX2-3 Mux' 'ADMAIF6'
amixer -c 0 sset 'AMX2-4 Mux' 'ADX2-1'
amixer -c 0 sset 'I2S3 Mux' 'AMX2'
```

- ADX connection:

```
amixer -c 0 sset 'ADX2 Mux' 'I2S3'
amixer -c 0 sset 'ADMAIF5 Mux' 'ADX2-4'
```

- AFC connection:

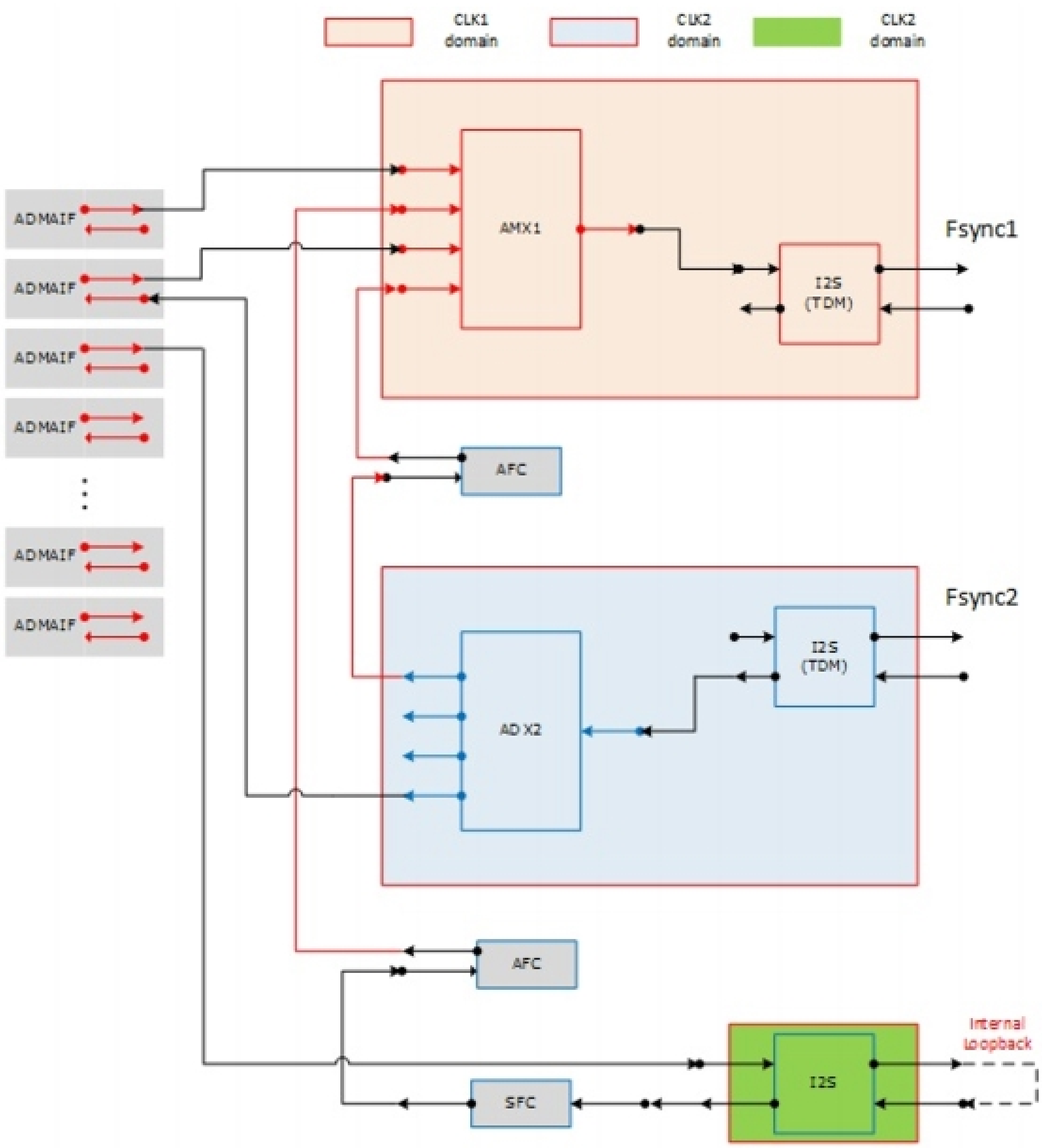
```
amixer -c 0 sset 'AFC1 Mux' 'I2S4'
amixer -c 0 sset 'I2S4 Mux' 'ADMAIF7'
```

Testing Commands

```
aplay -c 0 -D hw:0,4 -f S16_LE -r 48000 sample1.wav&
aplay -c 0 -D hw:0,5 -f S16_LE -r 48000 sample2.wav&
aplay -c 0 -D hw:0,6 -f S16_LE -r 48000 sample3.wav&
arecord -c 0 -D hw:0,4 -f wav -c 2 -r 48000 -b 16 sample4.wav &
```

I2S-x and I2S-y Under Same Clock Domain

If I2S-x and I2S-y are under the same clock domain for the sake of case study, AFC1 insertion between ADX1-1 and AMX1-4 does not help. However, the audio routing path can still be constructed as illustrated below and described in Routing Commands and Testing commands.



Routing

Commands

- AMX connection:

```
amixer -c 0 sset 'AMX1-1 Mux' 'ADMAIF1'
amixer -c 0 sset 'AMX1-2 Mux' 'AFC1'
amixer -c 0 sset 'AMX1-3 Mux' 'ADMAIF2'
amixer -c 0 sset 'AMX1-4 Mux' 'AFC2'
amixer -c 0 sset 'I2S4 Mux' 'AMX1'
```

- ADX connection:

```
amixer -c 0 sset 'ADX2 Mux' 'I2S3'
amixer -c 0 sset 'ADMAIF2 Mux' 'ADX2-4'
```

```
amixer -c 0 sset 'AFC1 Mux' 'ADX2-1'
```

- AFC/SFC connection:

```
amixer -c 0 sset 'SFC1 Mux' 'I2S5'
amixer -c 0 sset 'I2S5 Mux' 'ADMAIF3'
amixer -c 0 sset 'AFC2 Mux' 'SFC1'
amixer -c 0 cset iface=MIXER,name='output rate' '48kHz'
amixer -c 0 sset 'I2S5 Loopback' 1
```

Testing Commands

```
aplay -c 0 -D hw:0,0 -f S16_LE -r 48000 sample1.wav&
aplay -c 0 -D hw:0,1 -f S16_LE -r 48000 sample2.wav&
aplay -c 0 -D hw:0,2 -f S16_LE -r 48000 sample3.wav&
arecord -c 0 -D hw:0,1 -f wav -c 2 -r 48000 -b 16 sample4.wav &
```

Building Hardfp Crosstool-ng Toolchain and glibc

The NVIDIA® Tegra® Linux Driver Package contains the source code for the Crosstool-NG toolchain suite version 4.5.3 and the glibc suite. The Cross-NG toolchain suite resembles the toolchain NVIDIA uses to produce the L4T binaries.

This topic describes how to build Crosstool-NG and glibc on your Ubuntu host system.

Note: For a sample Crosstool-NG configuration file, see Appendix: Crosstool-NG Configuration File..

Toolchain Information

The toolchain contains following components:

- Crosstool-NG reference (http://crosstool-ng.org/)
- Cross Toolchain Version : 4.5.3
- glibc Version : 2.11

Host System Requirements

System requirements for the Ubuntu host systems includes:

- Ubuntu 10.04 32-bit distribution (64-bit distribution is not supported for building the toolchain)
- Fast host CPU such as Core 2 Duo (to reduce build time)
- 1GB Free space on HDD
- 2GB SDRAM

Dependent Packages

The Ubuntu host system must have the following packages installed:

- mercurial
- bison
- flex
- gperf
- texinfo
- m4
- libtool

- automake

Verify that the host system is connected to the internet, and run the following command to install the packages:

```
$ sudo apt-get install mercurial bison flex gperf texinfo m4 libtool automake
```

Building the Toolchain Suite

To build the toolchain you must:

- Set the `TOP_DIR` environment variable and create a directory tree
- Install autoconf-2.68
- Configure crosstool-NG
- Invoke the build

To set the TOP_DIR environment variable and create directories

1. To set the `TOP_DIR` variable to `${HOME}/crosstool` enter the following command:

```
$ export TOP_DIR="${HOME}/crosstool"
```

2. In the `${TOP_DIR}` directory, create the following subdirectories:

```
$ mkdir depends
$ mkdir crosstool-ng
$ cd depends
$ mkdir src
$ mkdir install
$ cd src
$ mkdir autoconf
$ mkdir ct-ng
```

To install autoconf-2.68

1. Change to the `autoconf` directory. Then download `autoconf-2.68.tar.bz2` by executing the following commands:

```
$ cd ${TOP_DIR}/depends/src/autoconf
$ wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.68.tar.bz2
```

2. Extract and configure `autoconf-2.68`:

```
$ tar xf autoconf-2.68.tar.bz2
$ cd autoconf-2.68
$ ./configure --prefix=${TOP_DIR}/depends/install/autoconf_install/autoconf-2.68-instal
```

3. Make and install `autoconf-2.68`:

```
$ make
```

```
$ make install
```

To configure crosstool-NG

1. Change to the `ct-ng` directory:

   ```
   $ cd ${TOP_DIR}/depends/src/ct-ng
   ```

2. Add the autoconf-2.68-install directory to your path:

   ```
   $ export PATH=${TOP_DIR}/depends/install/autoconf_install/autoconf-2.68-install/bin:${P
   ```

3. Clone the `crosstool-ng` repository:

   ```
   $ hg clone http://crosstool-ng.org/hg/crosstool-ng
   ```

4. Configure `crosstool-ng`:

   ```
   $ cd crosstool-ng
   $ ./bootstrap
   $ ./configure --prefix=${TOP_DIR}/depends/install/ct-ng_install/crosstool-ng-hg-install
   ```

5. Make and install `crosstool-ng`:

   ```
   $ make
   $ make install
   ```

6. Create the `${TOP_DIR}/crosstool-ng/src` directory for locally saving downloaded packages:

   ```
   mkdir ${TOP_DIR}/crosstool-ng/src
   ```

To invoke the build

1. Change to the `/crosstool-ng-hg-install/bin` directory:

   ```
   $ cd ${TOP_DIR}/depends/install/ct-ng_install/crosstool-ng-hg-install/bin
   ```

2. Copy the following content of `.config` from the Sample Crosstool-ng Configuration File appendix to this guide to a file called `.config`.

   `Note: .config is a hidden file. After creating it, confirm it exists in the correct location by running ls -a in the directory.`

3. Build `ct-ng` using 8 parallel paths:

   ```
   $./ct-ng oldconfig
   $./ct-ng build.8
   ```

   This will build the complete suite and install the binary components in `${TOP_DIR}/crosstool-ng/install`.

Verifying the Build

After a successful build, the `${TOP_DIR}/crosstool-ng/install` directory contains the following tree structure, as reported by the `tree` application (where available):

```
$ tree -L 2

|-- arm-cortex_a9-linux-gnueabi
|   |-- bin
|   |-- debug-root
|   |-- include
|   |-- lib -> sysroot/lib
|   |-- lib32 -> lib
|   |-- lib64 -> lib
|   `-- sysroot
|-- bin
|   |-- arm-cortex_a9-linux-gnueabi-addr2line
|   |-- arm-cortex_a9-linux-gnueabi-ar
|   |-- arm-cortex_a9-linux-gnueabi-as
|   |-- arm-cortex_a9-linux-gnueabi-c++
|   |-- arm-cortex_a9-linux-gnueabi-cc -> arm-cortex_a9-linux-gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-c++filt
|   |-- arm-cortex_a9-linux-gnueabi-cpp
|   |-- arm-cortex_a9-linux-gnueabi-ct-ng.config
|   |-- arm-cortex_a9-linux-gnueabi-g++
|   |-- arm-cortex_a9-linux-gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-gcc-4.5.3
|   |-- arm-cortex_a9-linux-gnueabi-gccbug
|   |-- arm-cortex_a9-linux-gnueabi-gcov
|   |-- arm-cortex_a9-linux-gnueabi-gprof
|   |-- arm-cortex_a9-linux-gnueabi-ld
|   |-- arm-cortex_a9-linux-gnueabi-ldd
|   |-- arm-cortex_a9-linux-gnueabi-nm
|   |-- arm-cortex_a9-linux-gnueabi-objcopy
|   |-- arm-cortex_a9-linux-gnueabi-objdump
|   |-- arm-cortex_a9-linux-gnueabi-populate
|   |-- arm-cortex_a9-linux-gnueabi-ranlib
|   |-- arm-cortex_a9-linux-gnueabi-readelf
|   |-- arm-cortex_a9-linux-gnueabi-size
|   |-- arm-cortex_a9-linux-gnueabi-strings
|   `-- arm-cortex_a9-linux-gnueabi-strip
|-- build.log.bz2
```

```
|-- include
|-- lib
|   |-- gcc
|   |-- ldscripts
|   `-- libiberty.a
|-- libexec
|   `-- gcc
`-- share
    `-- gcc-4.5.3
```

Building AARCH 64 Crosstool-ng Toolchain and glibc

The NVIDIA® Tegra® Linux Driver Package contains the script and patches used to create the set of toolchains that NVIDIA uses to produce the AARCH64 L4T binaries. This topic describes how to build these toolchains.

Additionally, a Crosstool-NG toolchain used to produce the ARM L4T binaries is described in the Building Crosstool-ng Toolchain and glibc chapter of this guide.

Toolchain Information

There are two toolchains required for AARCH64 builds. Both are built from the same component sources and patches, but they are configured differently, as the following:

- aarch64-unknown-linux-gnu
- arm-uknown-linux-gnueabi

The ARM toolchain is needed for a small compatibility component of the `arm64` kernel.

Each toolchain contains the following components:

- GCC version: 4.8.2
- Glibc Version : 2.17

Building the Toolchain

The toolchain build script and patches are provided in the Jetson TX1 Toolchain Build package.

To build the toolchain

1. Download the Jetson TX1 Toolchain Build package `jetson-tx1-toolchain-build.tbz2`.
2. Uncompress `jetson-tx1-toolchain-build.tbz2`. Output from `tar` is similar to the following:

   ```
   $ tar xpf jetson-tx1-toolchain-build.tbz2


   This unpacks into two folders, one for each toolchain:
   toolchain-build-aarch64
   toolchain-build-armhf
   ```

   Each toolchain folder contains the following:

- README instructions for setting up the build system and building the toolchain.
- A script that builds the toolchain.
- A patches folder containing patches to the toolchain sources.

3. Follow the build instructions in the README for each toolchain. Build products are located in the `install` directory.

Troubleshooting

Each step in the build process logs its progress and errors to a file named for that stage:

Build Process Step	Log File
Downloading sources	get_src.log
Applying patches	apply_patches.log
Build binutils	build_binutils.log
Installing Linux headers	install_linux_headers.log
Building GCC stage 1	build_gcc_stage1.log
Building Glibc stage 1	build_glibc_stage1.log
Building final Glibc	build_final_glibc.log

If the build succeeds, the script prints “Success!” at the end. Otherwise check log files for the last step shown to find the error.

The following are two common build errors:

- The build system configuration is different from what is suggested in the README. There may be support packages needed for your particular system.

  If using Ubuntu 12.04/14.04, install the necessary packages, in addition to the ones mentioned in the README, with the following commands:

  ```
  $ sudo apt-get install gawk
  $ sudo apt-get install texinfo
  $ sudo apt-get install automake
  $ sudo apt-get install libtool
  $ sudo apt-get install g++
  ```

- The build script downloads sources for the toolchain components from well-known locations. These URLs may become outdated over time. Make sure that the URLs in the `get_src()` function are up-to-date.

Host System Requirements

System requirements for the Ubuntu host systems includes:

- Ubuntu 10.04 32-bit distribution (64-bit distribution is not supported for building the toolchain)
- Fast host CPU such as Core 2 Duo (to reduce build time)
- 1GB Free space on HDD
- 2GB SDRAM

Dependent Packages

The Ubuntu host system must have the following packages installed:

- mercurial
- bison
- flex
- gperf
- texinfo
- m4
- libtool
- automake

Verify that the host system is connected to the internet, and run the following command to install the packages:

```
$ sudo apt-get install mercurial bison flex gperf texinfo m4 libtool automake
```

Building the Toolchain Suite

To build the toolchain you must:

- Set the `TOP_DIR` environment variable and create a directory tree
- Install autoconf-2.68
- Configure crosstool-NG
- Invoke the build

To set the TOP_DIR environment variable and create directories

4. To set the `TOP_DIR` variable to `${HOME}/crosstool` enter the following command:

```
$ export TOP_DIR="${HOME}/crosstool"
```

5. In the `${TOP_DIR}` directory, create the following subdirectories:

```
$ mkdir depends
$ mkdir crosstool-ng
$ cd depends
$ mkdir src
$ mkdir install
```

```
$ cd src
$ mkdir autoconf
$ mkdir ct-ng
```

To install autoconf-2.68

1. Change to the `autoconf` directory. Then download `autoconf-2.68.tar.bz2` by executing the following commands:

```
$ cd ${TOP_DIR}/depends/src/autoconf
$ wget http://ftp.gnu.org/gnu/autoconf/autoconf-2.68.tar.bz2
```

2. Extract and configure `autoconf-2.68`:

```
$ tar xf autoconf-2.68.tar.bz2
$ cd autoconf-2.68
$ ./configure --prefix=${TOP_DIR}/depends/install/autoconf_install/autoconf-2.68-instal
```

3. Make and install `autoconf-2.68`:

```
$ make
$ make install
```

To configure crosstool-NG

1. Change to the `ct-ng` directory:

```
$ cd ${TOP_DIR}/depends/src/ct-ng
```

2. Add the autoconf-2.68-install directory to your path:

```
$ export PATH=${TOP_DIR}/depends/install/autoconf_install/autoconf-2.68-install/bin:${P
```

3. Clone the `crosstool-ng` repository:

```
$ hg clone http://crosstool-ng.org/hg/crosstool-ng
```

4. Configure `crosstool-ng`:

```
$ cd crosstool-ng
$ ./bootstrap
$ ./configure --prefix=${TOP_DIR}/depends/install/ct-ng_install/crosstool-ng-hg-install
```

5. Make and install `crosstool-ng`:

```
$ make
$ make install
```

6. Create the `${TOP_DIR}/crosstool-ng/src` directory for locally saving downloaded packages:

```
mkdir ${TOP_DIR}/crosstool-ng/src
```

To invoke the build

1. Change to the `/crosstool-ng-hg-install/bin` directory:

```
$ cd ${TOP_DIR}/depends/install/ct-ng_install/crosstool-ng-hg-install/bin
```

2. Copy the following content of `.config` from the Sample Crosstool-ng Configuration File appendix to this guide to a file called `.config`.

   ```
   Note: .config is a hidden file. After creating it, confirm it exists in the
   correct location by running ls -a in the directory.
   ```

3. Build `ct-ng` using 8 parallel paths:

   ```
   $./ct-ng oldconfig
   $./ct-ng build.8
   ```

   This will build the complete suite and install the binary components in `${TOP_DIR}/crosstool-ng/install`.

Verifying the Build

After a successful build, the `${TOP_DIR}/crosstool-ng/install` directory contains the following tree structure, as reported by the `tree` application (where available):

```
$ tree -L 2


|-- arm-cortex_a9-linux-gnueabi
|   |-- bin
|   |-- debug-root
|   |-- include
|   |-- lib -> sysroot/lib
|   |-- lib32 -> lib
|   |-- lib64 -> lib
|   `-- sysroot
|-- bin
|   |-- arm-cortex_a9-linux-gnueabi-addr2line
|   |-- arm-cortex_a9-linux-gnueabi-ar
|   |-- arm-cortex_a9-linux-gnueabi-as
|   |-- arm-cortex_a9-linux-gnueabi-c++
|   |-- arm-cortex_a9-linux-gnueabi-cc -> arm-cortex_a9-linux-gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-c++filt
|   |-- arm-cortex_a9-linux-gnueabi-cpp
|   |-- arm-cortex_a9-linux-gnueabi-ct-ng.config
|   |-- arm-cortex_a9-linux-gnueabi-g++
|   |-- arm-cortex_a9-linux-gnueabi-gcc
|   |-- arm-cortex_a9-linux-gnueabi-gcc-4.5.3
```

```
|   |-- arm-cortex_a9-linux-gnueabi-gccbug
|   |-- arm-cortex_a9-linux-gnueabi-gcov
|   |-- arm-cortex_a9-linux-gnueabi-gprof
|   |-- arm-cortex_a9-linux-gnueabi-ld
|   |-- arm-cortex_a9-linux-gnueabi-ldd
|   |-- arm-cortex_a9-linux-gnueabi-nm
|   |-- arm-cortex_a9-linux-gnueabi-objcopy
|   |-- arm-cortex_a9-linux-gnueabi-objdump
|   |-- arm-cortex_a9-linux-gnueabi-populate
|   |-- arm-cortex_a9-linux-gnueabi-ranlib
|   |-- arm-cortex_a9-linux-gnueabi-readelf
|   |-- arm-cortex_a9-linux-gnueabi-size
|   |-- arm-cortex_a9-linux-gnueabi-strings
|   `-- arm-cortex_a9-linux-gnueabi-strip
|-- build.log.bz2
|-- include
|-- lib
|   |-- gcc
|   |-- ldscripts
|   `-- libiberty.a
|-- libexec
|   `-- gcc
`-- share
    `-- gcc-4.5.3
```

Watchdog Timer

If an application terminates or hangs, a Watchdog timer eventually expires, triggering a CPU reset, and enabling the system to recover without user intervention. The NVIDIA® Tegra® Linux Driver Package implements a watchdog timer WDT0, allocated to CPU0 of cluster0 or the shadow CPU of cluster1.

> **Note**: For information about the available Tegra Watchdog Timers and configurations, see the "Watchdog Timers (WDTs)" section of the *Tegra Technical Reference Manual* (*TRM*) for your chip.

WDT0 is not enabled in the Linux kernel by default; to enable, see To enable WDT0 from the Linux kernel or To enable WDT0 from user space. This hardware, when turned on, has a timer that starts decrementing. The default timeout value is 120 seconds. For Linux for Tegra (L4T), WDT0 is configured to use TIMER7; therefore, TIMER7 must not be used for any other purpose. When the timeout condition occurs, the WDT0 hardware sends a reset signal to the CPU that causes it to reset.

You can enable WDT0 from the kernel or from user space. If WDT0 is enabled in the kernel, during kernel boot, the kernel loads the WDT0 driver and then starts resetting, or "kicking" WDT0. This prevents the device restarting under normal operation.

If you already enabled the default WDT0 driver from the Linux kernel, your applications in the user space do not need to kick WDT0.

Alternatively, applications can manually enable WDT0 from user space using standard Linux system calls and then by kicking the watchdog periodically. For more information, see the sample code in To enable WDT0 from user space.

Normally, enabling WDT0 enablement is sufficient for system monitoring. If you need to enable Watchdog on other CPUs or AVP, you must modify the WDT driver.

To enable WDT0 from the Linux kernel

1. Go to the kernel configuration file:

   ```
   arch/arm/configs/tegra12_defconfig
   ```

2. Add the following 2 lines under `CONFIG_WATCHDOG_NOWAYOUT=y`:

   ```
   CONFIG_TEGRA_WATCHDOG=y
   CONFIG_TEGRA_WATCHDOG_ENABLE_ON_PROBE=y
   ```

To modify the WDT0 timeout value

1. Go to the WDT kernel driver:

   ```
   drivers/watchdog/tegra_wdt.c
   ```

2. Modify the heartbeat value. The default value is 120 seconds. The example below changes the timeout value to 60 seconds:

   ```
   -static int heartbeat = 120;
   +static int heartbeat = 60;
   ```

To enable WDT0 from user space

1. Go to the kernel configuration file:

```
arch/arm/configs/tegra12_defconfig
```

2. Add the following line under `CONFIG_WATCHDOG_NOWAYOUT=y`:

```
CONFIG_TEGRA_WATCHDOG=y
```

The WDT0 device node is `/dev/watchdog0`. The following user-space sample code shows opening, enabling, obtaining and specifiying the timeout value, and kicking the watchdog timer.

```
int fd, ret;
int timeout = 0;

/* open WDT0 device (WDT0 enables itself automatically) */
fd = open("/dev/watchdog0", O_RDWR);
if(fd < 0) {
    fprintf(stderr, "Open watchdog device failed!\n");
    return -1;
}
/* WDT0 is counting now,check the default timeout value */
ret = ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
if(ret) {
    fprintf(stderr, "Get watchdog timeout value failed!\n");
    return -1;
}
fprintf(stdout, "Watchdog timeout value: %d\n", timeout);

/* set new timeout value 60s */
/* Note the value should be within [5, 1000] */
timeout = 60;
ret = ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
if(ret) {
    fprintf(stderr, "Set watchdog timeout value failed!\n");
    return -1;
}
fprintf(stdout, "New watchdog timeout value: %d\n", timeout);

/*Kick WDT0, this should be running periodically */
ret = ioctl(fd, WDIOC_KEEPALIVE, NULL);
if(ret) {
```

```
    fprintf(stderr, "Kick watchdog failed!\n");
    return -1;
}
```

Licenses

This section provides license information for the NVIDIA® Tegra® Linux Driver Package. Different components of NVIDIA® Tegra® Linux Driver Package involve different licenses, granted variously by NVIDIA, other vendors, and GPL copyright holders. Please read the following carefully so you understand your rights and obligations under these licenses.

NVIDIA Software

Note: This software license applies to software in the "nv_tegra" directory (NVIDIA binary drivers and supporting software), the files in the “nv_tegra/nv_sample_apps” directory: “nvgstcapture”, “nvgstplayer”, and “libgstnvvidconv.so” (those files included in the nvgstapps.tbz2 file), and the files “bootloader/mkgpt”, “bootloader/mkbootimg”, and “bootloader/mkubootscript”.

License For Customer Use of NVIDIA Software

IMPORTANT NOTICE -- READ CAREFULLY: This License For Customer Use of NVIDIA Software ("LICENSE") is the agreement which governs use of the software of NVIDIA Corporation and its subsidiaries ("NVIDIA") downloadable, including computer software and associated printed materials ("SOFTWARE"). By downloading, installing, copying, or otherwise using the SOFTWARE, you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, do not download the SOFTWARE.

RECITALS

Use of NVIDIA's products requires three elements: the SOFTWARE, the hardware on a graphics controller board, and a personal computer. The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is not sold, and instead is only licensed for use, strictly in accordance with this document. The hardware is protected by various patents, and is sold, but this agreement does not cover that sale, since it may not necessarily be sold as a package with the SOFTWARE. This agreement sets forth the terms and conditions of the SOFTWARE LICENSE only.

1. DEFINITIONS

1.1 Customer.

Customer means the entity or individual that downloads the SOFTWARE.

2. GRANT OF LICENSE

2.1 Rights and Limitations of Grant.

NVIDIA hereby grants Customer the following non-exclusive, non-transferable right to use the SOFTWARE, with the following limitations:

2.1.1 Rights.

Customer may install and use one copy of the SOFTWARE on a single computer, and except for making one back-up copy of the Software, may not otherwise copy the SOFTWARE. This LICENSE of SOFTWARE may not be shared or used concurrently on different computers.

2.1.2 Linux/FreeBSD Exception.

Notwithstanding the foregoing terms of Section 2.1.1, SOFTWARE designed exclusively for use on the Linux or FreeBSD operating systems, or other operating systems derived from the source code to these operating systems, may be copied and redistributed, provided that the binary files thereof are not modified in any way (except for unzipping of compressed files).

2.1.3 Limitations.

No Reverse Engineering. Customer may not reverse engineer, decompile, or disassemble the SOFTWARE, nor attempt in any other manner to obtain the source code.

No Separation of Components. The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one computer, nor otherwise used separately from the other parts.

No Rental. Customer may not rent or lease the SOFTWARE to someone else.

3. TERMINATION

This LICENSE will automatically terminate if Customer fails to comply with any of the terms and conditions hereof. In such event, Customer must destroy all copies of the SOFTWARE and all of its component parts.

Defensive Suspension. If Customer commences or participates in any legal proceeding against NVIDIA, then NVIDIA may, in its sole discretion, suspend or terminate all license grants and any other rights provided under this LICENSE during the pendency of such legal proceedings.

4. COPYRIGHT

All title and copyrights in and to the SOFTWARE (including but not limited to all images, photographs, animations, video, audio, music, text, and other information incorporated into the SOFTWARE), the accompanying printed materials, and any copies of the SOFTWARE, are owned by NVIDIA, or its suppliers. The SOFTWARE is protected by copyright laws and international treaty provisions. Accordingly, Customer is required to treat the SOFTWARE like any other copyrighted material, except as otherwise allowed pursuant to this LICENSE and that it may make one copy of the SOFTWARE solely for backup or archive purposes.

5. APPLICABLE LAW

This agreement shall be deemed to have been made in, and shall be construed pursuant to, the laws of the State of California.

6. DISCLAIMER OF WARRANTIES AND LIMITATION ON LIABILITY

6.1 No Warranties.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE IS PROVIDED "AS IS" AND NVIDIA AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

6.2 No Liability for Consequential Damages.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NVIDIA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. MISCELLANEOUS

The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed. If any provision of this LICENSE is inconsistent with, or cannot be fully enforced under the law, such provision will be construed as limited to the extent necessary to be consistent with and fully enforceable under the law. This agreement is the final, complete and exclusive agreement between the parties relating to the subject matter hereof, and supersedes all prior or contemporaneous understandings and agreements relating to such subject matter, whether oral or written. Customer agrees that it will not ship, transfer or export the SOFTWARE into any country, or use the SOFTWARE in any manner, prohibited by the United States Bureau of Export Administration or any export laws, restrictions or regulations. This LICENSE may only be modified in writing signed by an authorized officer of NVIDIA.

Sample File System

The sample root file system is derived from Ubuntu Linux, version 14.04 for the hardware floating point (hardfp) release. Information on re-creating the root file system is provided in the T*egra Linux Driver Package Developers' Guide*. The license agreement for each software component is located in the software component's source code, made available from the same location from which this software was downloaded, or by request to oss-requests@nvidia.com.

GST OpenMAX

GNU LESSER GENERAL PUBLIC LICENSE
How to Apply These Terms to Your New Libraries

The software listed below is licensed under the terms of the LGPLv2.1 (see below). To obtain source code, contact oss-requests@nvidia.com.

gst-openmax (libgstomx.so, libgstegl-1.0.so.0, and libnvgstjpeg.so)

Version 2.1, February 1999

USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights. We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library. To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete

corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.) Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.

Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications. You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.
>
> <signature of Ty Coon>, 1 April 1990
>
> Ty Coon, President of Vice

That's all there is to it!

GST EGL

GStreamer EGL/GLES Sink

GStreamer EGL/GLES Sink

Copyright (C) 2012 Collabora Ltd.

> @author: Reynaldo H. Verdejo Pinochet <reynaldo@collabora.com>
>
> @author: Sebastian Dröge <sebastian.droege@collabora.co.uk>

Copyright (c) 2014, NVIDIA CORPORATION. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Linux Kernel

GNU GENERAL PUBLIC LICENSE

The Linux kernel in this release is licensed under the terms of the GPLv2 (see below). The revision of Linux kernel source code used to build this binary can be retrieved by running the 'source_sync.sh' script or by request to oss-requests@nvidia.com.

The device-tree-compiler (dtc) binary located in the “kernel” directory was built from the Linux kernel source code provided by this release. It also is licensed under the terms of the GPLv2 (see below). The revision of the Linux kernel source code which was used to build this dtc binary can be retrieved by request to oss-requests@nvidia.com.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

    c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
>
> Copyright (C) <year> <name of author>
>
> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) year name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

mkbootimg and mkubootscript

Apache License
Copyright: WIDE Project
GNU General Public License
Copyright: Regents of the University of California

mkbootimg and mkubootscript are provided under the following terms:

Copyright 2007, The Android Open Source Project

Apache License

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the

License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this fi

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under t
```

Copyright: WIDE Project

Copyright (C) 1995, 1996, 1997, and 1998 WIDE Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

FIPS pub 180-1: Secure Hash Algorithm (SHA-1)
based on: http://csrc.nist.gov/fips/fip180-1.txt
implemented by Jun-ichiro itojun Itoh <itojun@itojun.org>

GNU General Public License

For the complete text of this license, see GNU GENERAL PUBLIC LICENSE in this section.

Copyright: Regents of the University of California

Copyright (c) 1992, 1993
The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED

U-Boot and mkimage

U-Boot is Free Software. It is copyrighted by Wolfgang Denk and many others who contributed code (see the actual source code for details). You can redistribute U-Boot and/or modify it under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation. Most of it can also be distributed, at your option, under any later version of the GNU General Public License -- see individual files for exceptions.

NOTE! This license does *not* cover the so-called "standalone" applications that use U-Boot services by means of the jump table provided by U-Boot exactly for this purpose - this is merely considered normal use of U-Boot, and does *not* fall under the heading of "derived work" -- see file Licenses/Exceptions for details.

Also note that the GPL and the other licenses are copyrighted by the Free Software Foundation and other organizations, but the instance of code that they refer to (the U-Boot source code) is copyrighted by me and others who actually wrote it.

-- Wolfgang Denk

Like many other projects, U-Boot has a tradition of including big blocks of License headers in all files. This not only blows up the source code with mostly redundant information, but also makes it very difficult to generate License Clearing Reports. An additional problem is that even the same licenses are referred to by a number of slightly varying text blocks (full, abbreviated, different indentation, line wrapping and/or white space, with obsolete address information, ...) which makes automatic processing a nightmare.

To make this easier, such license headers in the source files will be replaced with a single line reference to Unique License Identifiers as defined by the Linux Foundation's SPDX project [1]. For example, in a source file the full "GPL v2.0 or later" header text will be replaced by a single line:

SPDX-License-Identifier: GPL-2.0+

Ideally, the license terms of all files in the source tree should be defined by such License Identifiers; in no case a file can contain more than one such License Identifier list.

If a "SPDX-License-Identifier:" line references more than one Unique License Identifier, then this means that the respective file can be used under the terms of either of these licenses, i. e. with

SPDX-License-Identifier: GPL-2.0+ BSD-3-Clause

you can chose between GPL-2.0+ and BSD-3-Clause licensing.

We use the SPDX Unique License Identifiers here; these are available at [2].

[1] http://spdx.org/

[2] http://spdx.org/licenses/

Full Name	SPDX ID	OSI approved	File Name	URI
GNU General Public License v2.0 only	GPL-2.0	Y	gpl-2.0.txt	http://www.gnu.org/licenses/gpl-2.0.txt
GNU General Public License v2.0 or later	GPL-2.0+	Y	gpl-2.0.txt	http://www.gnu.org/licenses/gpl-2.0.txt
GNU Library General Public License v2 or later	LGPL-2.0+	Y	lgpl-2.0.txt	http://www.gnu.org/licenses/old-licenses/lgpl-2.0.txt
GNU Lesser General Public License v2.1 or later	LGPL-2.1+	Y	lgpl-2.1.txt	http://www.gnu.org/licenses/old-licenses/lgpl-2.1.txt
eCos license version 2.0	eCos-2.0		eCos-2.0.txt	http://www.gnu.org/licenses/ecos-license.html
BSD 2-Clause License	BSD-2-Clause	Y	bsd-2-clause.txt	http://spdx.org/licenses/BSD-2-Clause
BSD 3-clause "New" or "Revised" License	BSD-3-Clause	Y	bsd-3-clause.txt	http://spdx.org/licenses/BSD-3-Clause#licenseText
IBM PIBS (PowerPC Initialization and Boot Software) license	IBM-pibs	-	ibm-pibs.txt	

ISC License	ISC	Y	isc.txt	https://spdx.org/licenses/ISC
X11 License	X11	-	x11.txt	https://spdx.org/licenses/X11.html

GNU GENERAL PUBLIC LICENSE, Version 2, June 1991

For the complete text of this license, see GNU GENERAL PUBLIC LICENSE in this section.

mkbctpart

The mkbctpart library is covered by the License For Customer Use of NVIDIA Software. For the complete text of that license, see License For Customer Use of NVIDIA Software in this section.

brcm_patchram_plus

The brcm_patchram_plus is installed at the following location:

```
/usr/sbin/brcm_patchram_plus
```

Copyright (C) 2009-2011 Broadcom Corporation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For the complete text of the Apache license, see Apache License in this section.

libnvcam_imageencoder.so

Copyright (c) 2015, NVIDIA CORPORATION. All rights reserved.

NVIDIA CORPORATION and its licensors retain all intellectual property and proprietary rights in and to this software, related documentation and any modifications thereto. Any use, reproduction, disclosure or distribution of this software and related documentation without an express license agreement from NVIDIA CORPORATION is strictly prohibited.

This product incorporates software provided under the following terms:

==

NOTICE file in this case for the JPEGcode.

==

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.

All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

See also RECITALS in NVIDIA Software.

libscf

ZLIB DATA COMPRESSION LIBRARY

libscf is covered by the NVIDIA license. For the complete text of this license, see License For Customer Use of NVIDIA Software in this section.

Portions of this software are provided under the following terms:

This folder contains libraries and headers of a few very popular still image codecs used by highgui module. The libraries and headers are preferably to build Win32 and Win64 versions of OpenCV. On UNIX systems all the libraries are automatically detected by configure script. In order to use these versions of libraries instead of system ones on UNIX systems you should use BUILD_<library_name> CMake flags (for example, BUILD_PNG for the libpng library).

----

libjpeg 8d (8.4) - The Independent JPEG Group's JPEG software.

Copyright (C) 1991-2012, Thomas G. Lane, Guido Vollbeding.

See IGJ home page http://www.ijg.org for details and links to the source code

HAVE_JPEG preprocessor flag must be set to make highgui use libjpeg. On UNIX systems configure script takes care of it.

----

libpng 1.5.12 - Portable Network Graphics library.

Copyright (c) 2004, 2006-2012 Glenn Randers-Pehrson. See libpng home page http://www.libpng.org for details and links to the source code

HAVE_PNG preprocessor flag must be set to make highgui use libpng. On UNIX systems configure script takes care of it.

----

libtiff 4.0.2 - Tag Image File Format (TIFF) Software

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

See libtiff home page http://www.remotesensing.org/libtiff/ for details and links to the source code

HAVE_TIFF preprocessor flag must be set to make highgui use libtiff. On UNIX systems configure script takes care of it. In this build support for ZIP (LZ77 compression) is turned on.

----

zlib 1.2.7 - General purpose LZ77 compression library

Copyright (C) 1995-2012 Jean-loup Gailly and Mark Adler.

See zlib home page http://www.zlib.net for details and links to the source code

No preprocessor definition is needed to make highgui use this library - it is included automatically if either libpng or libtiff are used.

----

jasper-1.900.1 - JasPer is a collection of software (i.e., a library and application programs) for the coding and manipulation of images. This software can handle image data in a variety of formats. One such format supported by JasPer is the JPEG-2000 format defined in ISO/IEC 15444-1.

Copyright (c) 1999-2000 Image Power, Inc.

Copyright (c) 1999-2000 The University of British Columbia

Copyright (c) 2001-2003 Michael David Adams

The JasPer license can be found in src/libjasper.

OpenCV on Windows uses pre-built libjasper library (lib/libjasper*). To get the latest source code, please, visit the project homepage: http://www.ece.uvic.ca/~mdadams/jasper/

----

openexr-1.7.1 - OpenEXR is a high dynamic-range (HDR) image file format developed by Industrial Light & Magic for use in computer imaging applications.

Copyright (c) 2006, Industrial Light & Magic, a division of Lucasfilm Entertainment Company Ltd. Portions contributed and copyright held by others as indicated. All rights reserved.

The project homepage: http://www.openexr.com

----

ffmpeg-0.8.0 - FFmpeg is a complete, cross-platform solution to record, convert and stream audio and video. It includes libavcodec - the leading audio/video codec library, and also libavformat, libavutils and other helper libraries that are used by OpenCV (in highgui module) to read and write video files.

The project homepage: http://ffmpeg.org/

==

== LICENSE file ==

== ==

== in this case for the Open Computer Vision code. ==

===

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license.

If you do not agree to this license, do not download, install, copy or use the software.

License Agreement for Protocol Buffers

This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in:

  ```
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  ```

  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in

  ```
  src/google/protobuf/stubs/atomicops_internals_power.h.
  ```

  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

License Agreement for Open Source Computer Vision Library

Copyright (C) 2000-2008, Intel Corporation, all rights reserved.

Copyright (C) 2008-2011, Willow Garage Inc., all rights reserved.

Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the copyright holders may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

License Agreement for OpenCV Tutorial Library

```
/*                           License Agreement
 *                      For OpenCV Tutorial Library
 *                    http://computer-vision-talks.com
 *
 * Third party copyrights are property of their respective owners.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * Redistribution's of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * Redistribution's in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 *
 * The name of the copyright holders may not be used to endorse or
 * promote products derived from this software without specific prior
 * written permission.
 *
 * This software is provided by the copyright holders and contributors
 * "as is" and any express or implied warranties, including, but not
```

```
* limited to, the implied warranties of merchantability and fitness
* for a particular purpose are disclaimed.
* In no event shall the Intel Corporation or contributors be liable
* for any direct, indirect, incidental, special, exemplary, or
* consequential damages (including, but not limited to, procurement of
* substitute goods or services; loss of use, data, or profits; or
* business interruption) however caused and on any theory of
* liability, whether in contract, strict liability, or tort (including
* negligence or otherwise) arising in any way out of the use of this software, even if advised
*/
```

===

NOTICE file in this case for the Open Computer Vision code.

===

IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.

By downloading, copying, installing or using the software you agree to this license. If you do not agree to this license, do not download, install, copy or use the software.

License Agreement for Open Source Computer Vision Library

Copyright (C) 2000-2008, Intel Corporation, all rights reserved.
Copyright (C) 2008-2011, Willow Garage Inc., all rights reserved.
Third party copyrights are property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of the copyright holders may not be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Intel Corporation or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

===

NOTICE file in this case for the OpenCV Tutorial code.

===

License Agreement for OpenCV Tutorial Library

```
/*                           License Agreement
 *                       For OpenCV Tutorial Library
 *                     http://computer-vision-talks.com
 *
 * Third party copyrights are property of their respective owners.
 *
 *  Redistribution and use in source and binary forms, with or without
 *  modification,are permitted provided that the following conditions
 *  are met:
 *
 *  Redistribution's of source code must retain the above copyright
 *  notice, this list of conditions and the following disclaimer.
 *
 *  Redistribution's in binary form must reproduce the above copyright
 *  notice, this list of conditions and the following disclaimer in the
 *  documentation and/or other materials provided with the
 *  distribution.
 *
 *  The name of the copyright holders may not be used to endorse or
 *  promote products derived from this software without specific prior
 *  written permission.
 *
 *  This software is provided by the copyright holders and contributors
 *  "as is" and any express or implied warranties, including, but not
 *  limited to, the implied warranties of merchantability and fitness
 *  for a particular purpose are disclaimed.
 *  In no event shall the Intel Corporation or contributors be liable X
 *  limited to, the implied indirect, incidental, special, exemplary,
 *  or consequential damages (including, but not limited to,
 *  procurement of substitute goods or services;
 *  loss of use, data, or profits; or business interruption) however
 *  caused and on any theory of liability, whether in contract, strict
 *  liability,or tort (including negligence or otherwise) arising in
 *  any way out of the use of this software, even if advised of the
 *  possibility of such damage.
```

```
*/
```

This package was debianized by Christopher L Cheney <ccheney@debian.org> on Fri, 22 Aug 2003 01:33:34 -0500.

The current maintainer is Roland Stigge <stigge@antcom.de>

It was downloaded from http://www.ece.uvic.ca/~mdadams/jasper/

Upstream Author: Michael Adams <mdadams@ece.uvic.ca>

License:

JasPer License Version 2.0

Copyright (c) 1999-2000 Image Power, Inc.

Copyright (c) 1999-2000 The University of British Columbia

Copyright (c) 2001-2003 Michael David Adams

All rights reserved.

Permission is hereby granted, free of charge, to any person (the "User") obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. The above copyright notices and this permission notice (which includes the disclaimer below) shall be included in all copies or substantial portions of the Software.
2. The name of a copyright holder shall not be used to endorse or promote products derived from the Software without specific prior written permission.

THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY. EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

JasPer License Version 2.0

This package was debianized by Christopher L Cheney <ccheney@debian.org> on

Fri, 22 Aug 2003 01:33:34 -0500.

The current maintainer is Roland Stigge <stigge@antcom.de>

It was downloaded from http://www.ece.uvic.ca/~mdadams/jasper/

Upstream Author: Michael Adams <mdadams@ece.uvic.ca>

License:

JasPer License Version 2.0

The Independent JPEG Group's JPEG software

==

README for release 6b of 27-Mar-1998

====================================

This distribution contains the sixth public release of the Independent JPEG Group's free JPEG software. You are welcome to redistribute this software and to use it for any purpose, subject to the conditions under LEGAL ISSUES, below.

Serious users of this software (particularly those incorporating it into larger programs) should contact IJG at jpeg-info@uunet.uu.net to be added to our electronic mailing list. Mailing list members are notified of updates and have a chance to participate in technical discussions, etc.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group.

IJG is not affiliated with the official ISO JPEG standards committee.

DOCUMENTATION ROADMAP

This file contains the following sections:

OVERVIEW General description of JPEG and the IJG software.

LEGAL ISSUES Copyright, lack of warranty, terms of distribution.

REFERENCES Where to learn more about JPEG.

ARCHIVE LOCATIONS Where to find newer versions of this software.

RELATED SOFTWARE Other stuff you should get.

FILE FORMAT WARS Software *not* to get.

TO DO Plans for future IJG releases.

Other documentation files in the distribution are:

User documentation:

install.doc How to configure and install the IJG software.

usage.doc Usage instructions for cjpeg, djpeg, jpegtran, rdjpgcom, and wrjpgcom.

*.1 Unix-style man pages for programs (same info as usage.doc).

wizard.doc Advanced usage instructions for JPEG wizards only.

change.log Version-to-version change highlights.

Programmer and internal documentation:

libjpeg.doc How to use the JPEG library in your own programs.

example.c Sample code for calling the JPEG library.

structure.doc Overview of the JPEG library's internal structure.

filelist.doc Road map of IJG files.

coderules.doc Coding style rules --- please read if you contribute code.

Please read at least the files install.doc and usage.doc. Useful information can also be found in the JPEG FAQ (Frequently Asked Questions) article. See ARCHIVE LOCATIONS below to find out where to obtain the FAQ article.

If you want to understand how the JPEG code works, we suggest reading one or more of the REFERENCES, then looking at the documentation files (in roughly the order listed) before diving into the code.

OVERVIEW

This package contains C software to implement JPEG image compression and decompression. JPEG (pronounced "jay-peg") is a standardized compression method for full-color and gray-scale images. JPEG is intended for compressing "real-world" scenes; line drawings, cartoons and other non-realistic images are not its strong suit. JPEG is lossy, meaning that the output image is not exactly identical to the input image. Hence you must not use JPEG if you have to have identical output bits. However, on typical photographic images, very good compression levels can be obtained with no visible change, and remarkably high compression levels are possible if you can tolerate a low-quality image. For more details, see the references, or just experiment with various compression settings.

This software implements JPEG baseline, extended-sequential, and progressive compression processes. Provision is made for supporting all variants of these processes, although some uncommon parameter settings aren't implemented yet. For legal reasons, we are not distributing code for the arithmetic-coding variants of JPEG; see LEGAL ISSUES. We have made no provision for supporting the hierarchical or lossless processes defined in the standard.

We provide a set of library routines for reading and writing JPEG image files, plus two sample applications "cjpeg" and "djpeg", which use the library to perform conversion between JPEG and some other popular image file formats. The library is intended to be reused in other applications.

In order to support file conversion and viewing software, we have included considerable functionality beyond the bare JPEG coding/decoding capability; for example, the color quantization modules are not strictly part of JPEG decoding, but they are essential for output to colormapped file formats or colormapped displays. These extra functions can be compiled out of the library if not required for a particular application. We have also included "jpegtran", a utility for lossless transcoding between different JPEG processes, and "rdjpgcom" and "wrjpgcom", two simple applications for inserting and extracting textual comments in JFIF files.

The emphasis in designing this software has been on achieving portability and flexibility, while also making it fast enough to be useful. In particular, the software is not intended to be read as a tutorial on JPEG. (See the REFERENCES section for introductory material.) Rather, it is intended to be reliable, portable, industrial-strength code. We do not claim to have achieved that goal in every aspect of the software, but we strive for it.

We welcome the use of this software as a component of commercial products. No royalty is required, but we do ask for an acknowledgement in product documentation, as described under LEGAL ISSUES.

LEGAL ISSUES

============

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)

2. You can use this software for whatever you want. You don't have to pay us.

3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.

All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

REFERENCES

==========

We highly recommend reading one or more of these references before trying to understand the innards of the JPEG software.

The best short technical introduction to the JPEG compression algorithm is

Wallace, Gregory K. "The JPEG Still Picture Compression Standard",

Communications of the ACM, April 1991 (vol. 34 no. 4), pp. 30-44.

(Adjacent articles in that issue discuss MPEG motion picture compression, applications of JPEG, and related topics.) If you don't have the CACM issue handy, a PostScript file containing a revised version of Wallace's article is available at ftp://ftp.uu.net/graphics/jpeg/wallace.ps.gz. The file (actually a preprint for an article that appeared in IEEE Trans. Consumer Electronics) omits the sample images that appeared in CACM, but it includes corrections and some added material. Note: the Wallace article is copyright ACM and IEEE, and it may not be used for commercial purposes.

A somewhat less technical, more leisurely introduction to JPEG can be found in "The Data Compression Book" by Mark Nelson and Jean-loup Gailly, published by M&T Books (New York), 2nd ed. 1996, ISBN 1-55851-434-1. This book provides good explanations and example C code for a multitude of compression methods including JPEG. It is an excellent source if you are comfortable reading C code but don't know much about data compression in general. The book's JPEG sample code is far from industrial-strength, but when you are ready to look at a full implementation, you've got one here...

The best full description of JPEG is the textbook "JPEG Still Image Data Compression Standard" by William B. Pennebaker and Joan L. Mitchell, published by Van Nostrand Reinhold, 1993, ISBN 0-442-01272-1. Price US $59.95, 638 pp. The book includes the complete text of the ISO JPEG standards (DIS 10918-1 and draft DIS 10918-2). This is by far the most complete exposition of JPEG in existence, and we highly recommend it.

The JPEG standard itself is not available electronically; you must order a paper copy through ISO or ITU. (Unless you feel a need to own a certified official copy, we recommend buying the Pennebaker and Mitchell book instead; it's much cheaper and includes a great deal of useful explanatory material.) In the USA, copies of the standard may be ordered from ANSI Sales at (212) 642-4900, or from Global Engineering Documents at (800) 854-7179. (ANSI doesn't take credit card orders, but Global does.) It's not cheap: as of 1992, ANSI was charging $95 for Part 1 and $47 for Part 2, plus 7% shipping/handling. The standard is divided into two parts, Part 1 being the actual specification, while Part 2 covers compliance testing methods. Part 1 is titled "Digital Compression and Coding of Continuous-tone Still Images, Part 1: Requirements and guidelines" and has document numbers ISO/IEC IS 10918-1, ITU-T T.81. Part 2 is titled "Digital Compression and Coding of

Continuous-tone Still Images, Part 2: Compliance testing" and has document numbers ISO/IEC IS 10918-2, ITU-T T.83.

Some extensions to the original JPEG standard are defined in JPEG Part 3, a newer ISO standard numbered ISO/IEC IS 10918-3 and ITU-T T.84. IJG currently does not support any Part 3 extensions.

The JPEG standard does not specify all details of an interchangeable file format. For the omitted details we follow the "JFIF" conventions, revision 1.02. A copy of the JFIF spec is available from:

Literature Department

C-Cube Microsystems, Inc.

1778 McCarthy Blvd.

Milpitas, CA 95035

phone (408) 944-6300, fax (408) 944-6314

A PostScript version of this document is available by FTP at ftp://ftp.uu.net/graphics/jpeg/jfif.ps.gz. There is also a plain text version at ftp://ftp.uu.net/graphics/jpeg/jfif.txt.gz, but it is missing the figures.

The TIFF 6.0 file format specification can be obtained by FTP from ftp://ftp.sgi.com/graphics/tiff/TIFF6.ps.gz. The JPEG incorporation scheme found in the TIFF 6.0 spec of 3-June-92 has a number of serious problems. IJG does not recommend use of the TIFF 6.0 design (TIFF Compression tag 6). Instead, we recommend the JPEG design proposed by TIFF Technical Note #2 (Compression tag 7). Copies of this Note can be obtained from ftp.sgi.com or from ftp://ftp.uu.net/graphics/jpeg/. It is expected that the next revision of the TIFF spec will replace the 6.0 JPEG design with the Note's design. Although IJG's own code does not support TIFF/JPEG, the free libtiff library uses our library to implement TIFF/JPEG per the Note. libtiff is available from ftp://ftp.sgi.com/graphics/tiff/.

ARCHIVE LOCATIONS

=================

The "official" archive site for this software is ftp.uu.net (Internet address 192.48.96.9). The most recent released version can always be found there in directory graphics/jpeg. This particular version will be archived as ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz. If you don't have direct Internet access, UUNET's archives are also available via UUCP; contact help@uunet.uu.net for information on retrieving files that way.

Numerous Internet sites maintain copies of the UUNET files. However, only ftp.uu.net is guaranteed to have the latest official version.

You can also obtain this software in DOS-compatible "zip" archive format from the SimTel archives (ftp://ftp.simtel.net/pub/simtelnet/msdos/graphics/), or on CompuServe in the Graphics Support forum (GO CIS:GRAPHSUP), library 12 "JPEG Tools". Again, these versions may sometimes lag behind the ftp.uu.net release.

The JPEG FAQ (Frequently Asked Questions) article is a useful source of general information about JPEG. It is updated constantly and therefore is not included in this distribution. The FAQ is posted every two weeks to Usenet newsgroups comp.graphics.misc, news.answers, and other groups. It is available on the World Wide Web at http://www.faqs.org/faqs/jpeg-faq/ and other news.answers archive sites, including the official news.answers archive at rtfm.mit.edu: ftp://rtfm.mit.edu/pub/usenet/news.answers/jpeg-faq/. If you don't have Web or FTP access, send e-mail to mail-server@rtfm.mit.edu with body send usenet/news.answers/jpeg-faq/part1

send usenet/news.answers/jpeg-faq/part2

RELATED SOFTWARE

================

Numerous viewing and image manipulation programs now support JPEG. (Quite a few of them use this library to do so.) The JPEG FAQ described above lists some of the more popular free and shareware viewers, and tells where to obtain them on Internet.

If you are on a Unix machine, we highly recommend Jef Poskanzer's free PBMPLUS software, which provides many useful operations on PPM-format image files. In particular, it can convert PPM images to and from a wide range of other formats, thus making cjpeg/djpeg considerably more useful. The latest version is distributed by the NetPBM group, and is available from numerous sites, notably ftp://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/. Unfortunately PBMPLUS/NETPBM is not nearly as portable as the IJG software is; you are likely to have difficulty making it work on any non-Unix machine.

A different free JPEG implementation, written by the PVRG group at Stanford, is available from ftp://havefun.stanford.edu/pub/jpeg/. This program is designed for research and experimentation rather than production use; it is slower, harder to use, and less portable than the IJG code, but it is easier to read and modify. Also, the PVRG code supports lossless JPEG, which we do not. (On the other hand, it doesn't do progressive JPEG.)

FILE FORMAT WARS

================

Some JPEG programs produce files that are not compatible with our library. The root of the problem is that the ISO JPEG committee failed to specify a concrete file format. Some vendors "filled in the blanks" on their own, creating proprietary formats that no one else could read. (For example, none of the early commercial JPEG implementations for the Macintosh were able to exchange compressed files.)

The file format we have adopted is called JFIF (see REFERENCES). This format has been agreed to by a number of major commercial JPEG vendors, and it has become the de facto standard. JFIF is a minimal or "low end" representation. We recommend the use of TIFF/JPEG (TIFF revision 6.0 as modified by TIFF Technical Note #2) for "high end" applications that need to record a lot of additional data about an image. TIFF/JPEG is fairly new and not yet widely supported, unfortunately.

The upcoming JPEG Part 3 standard defines a file format called SPIFF. SPIFF is interoperable with JFIF, in the sense that most JFIF decoders should be able to read the most common variant of SPIFF. SPIFF has some technical advantages over JFIF, but its major claim to fame is simply that it is an official standard rather than an informal one. At this point it is unclear whether SPIFF will supersede JFIF or whether JFIF will remain the de-facto standard. IJG intends to support SPIFF once the standard is frozen, but we have not decided whether it should become our default output format or not. (In any case, our decoder will remain capable of reading JFIF indefinitely.)

Various proprietary file formats incorporating JPEG compression also exist. We have little or no sympathy for the existence of these formats. Indeed, one of the original reasons for developing this free software was to help force convergence on common, open format standards for JPEG files. Don't use a proprietary file format!

TO DO

=====

The major thrust for v7 will probably be improvement of visual quality. The current method for scaling the quantization tables is known not to be very good at low Q values. We also intend to investigate block boundary smoothing, "poor man's variable quantization", and other means of improving quality-vs-file-size performance without sacrificing compatibility.

In future versions, we are considering supporting some of the upcoming JPEG Part 3 extensions --- principally, variable quantization and the SPIFF file format.

As always, speeding things up is of great interest.

Please send bug reports, offers of help, etc. to jpeg-info@uunet.uu.net.

Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler

Kevin Bracey

Sam Bushell

Magnus Holmgren

Greg Roelofs

Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger

Dave Martindale

Guy Eric Schalnat

Paul Schmidt

Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png_get_copyright" function is available, for convenient use in "about" boxes and the like:

printf("%s",png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson

glennrp at users.sourceforge.net

July 11, 2012

USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

== LICENSE file ==

== ==

== in this case for the Threading Building Blocks code ==

==

GNU GENERAL PUBLIC LICENSE

For the complete text of this license, see GNU GENERAL PUBLIC LICENSE in this section.

Threading Building Blocks

The source code of Threading Building Blocks is distributed under version 2 of the GNU General Public License, with the so-called "runtime exception," as follows (or see any header or implementation file):

As a special exception, you may use this file as part of a free software library without restriction. Specifically, if other files instantiate templates or use macros or inline functions from this file, or you compile this file and link it with other files to produce an executable, this file does not by itself cause the resulting executable to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the executable file might be covered by the GNU General Public License.

ZLIB DATA COMPRESSION LIBRARY

zlib 1.2.7 is a general purpose data compression library. All the code is thread safe. The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950 (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).

All functions of the compression library are documented in the file zlib.h (volunteer to write man pages welcome, contact zlib@gzip.org). A usage example of the library is given in the file test/example.c which also tests that the library is working correctly. Another example is given in the file test/minigzip.c. The compression library itself is composed of all source files in the root directory.

To compile all files and run the test program, follow the instructions given at the top of Makefile.in. In short "./configure; make test", and if that goes well, "make install" should work for most flavors of Unix. For Windows, use one of the special makefiles in win32/ or contrib/vstudio/ . For VMS, use make_vms.com.

Questions about zlib should be sent to <zlib@gzip.org>, or to Gilles Vollant <info@winimage.com> for the Windows DLL version. The zlib home page is http://zlib.net/ . Before reporting a problem, please check this site to verify that you have the latest version of zlib; otherwise get the latest version and check whether the problem still exists or not.

PLEASE read the zlib FAQ http://zlib.net/zlib_faq.html before asking for help.

Mark Nelson <markn@ieee.org> wrote an article about zlib for the Jan. 1997 issue of Dr. Dobb's Journal; a copy of the article is available at http://marknelson.us/1997/01/01/zlib-engine/ .

The changes made in version 1.2.7 are documented in the file ChangeLog.

Unsupported third party contributions are provided in directory contrib/ .

zlib is available in Java using the java.util.zip package, documented at http://java.sun.com/developer/technicalArticles/Programming/compression/ .

A Perl interface to zlib written by Paul Marquess <pmqs@cpan.org> is available at CPAN (Comprehensive Perl Archive Network) sites, including http://search.cpan.org/~pmqs/IO-Compress-Zlib/ .

A Python interface to zlib written by A.M. Kuchling <amk@amk.ca> is available in Python 1.5 and later versions, see http://docs.python.org/library/zlib.html .

zlib is built into tcl: http://wiki.tcl.tk/4610 .

An experimental package to read and write files in .zip format, written on top of zlib by Gilles Vollant <info@winimage.com>, is available in the contrib/minizip directory of zlib.

Notes for some targets:

- For Windows DLL versions, please see win32/DLL_FAQ.txt

- For 64-bit Irix, deflate.c must be compiled without any optimization. With -O, one libpng test fails. The test works in 32 bit mode (with the -n32 compiler flag). The compiler bug has been reported to SGI.

- zlib doesn't work with gcc 2.6.3 on a DEC 3000/300LX under OSF/1 2.1 it works when compiled with cc.

- On Digital Unix 4.0D (formerly OSF/1) on AlphaServer, the cc option -std1 is necessary to get gzprintf working correctly. This is done by configure.

- zlib doesn't work on HP-UX 9.05 with some versions of /bin/cc. It works with other compilers. Use "make test" to check your compiler.

- gzdopen is not supported on RISCOS or BEOS.

- For PalmOs, see http://palmzlib.sourceforge.net/

Acknowledgments:

The deflate format used by zlib was defined by Phil Katz. The deflate and zlib specifications were written by L. Peter Deutsch. Thanks to all the people who reported problems and suggested various improvements in zlib; they are too numerous to cite here.

Copyright notice:

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler

jloup@gzip.org madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes. Please read the FAQ for more information on the distribution of modified source versions.

gstvideocuda

Copyright (c) 2015, NVIDIA CORPORATION. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

bpmp and tos-img

The bpmp and tos-img libraries are covered by the following copyrights and license. This information is also provided in the `LICENSE.bpmp_and_tos-img` file.

> Copyright (c) 2013-2015, NVIDIA CORPORATION. All rights reserved.
>
> NVIDIA CORPORATION and its licensors retain all intellectual property and proprietary rights in and to this software, related documentation and any modifications thereto. Any use, reproduction, disclosure or

Appendix: Crosstool-NG Configuration File

The following is a sample `.config` file for the Crosstool-NG toolchain. For more information, see Building Crosstool-ng Toolchain and glibc in this guide.

```
# Automatically generated make config: don't edit
# crosstool-NG hg+-11c23aa9c9f9 Configuration
# Tue Aug 21 15:05:23 2012
#
CT_CONFIGURE_has_xz=y
CT_CONFIGURE_has_cvs=y
CT_CONFIGURE_has_svn=y
CT_MODULES=y

#
# Paths and misc options
#

#
# crosstool-NG behavior
#
# CT_OBSOLETE is not set
CT_EXPERIMENTAL=y
# CT_DEBUG_CT is not set

#
# Paths
#
CT_LOCAL_TARBALLS_DIR="${TOP_DIR}/crosstool-ng/src"
CT_SAVE_TARBALLS=y
CT_WORK_DIR="${TOP_DIR}/crosstool-ng/work"
CT_PREFIX_DIR="${TOP_DIR}/crosstool-ng/install"
CT_INSTALL_DIR="${CT_PREFIX_DIR}"
CT_RM_RF_PREFIX_DIR=y
CT_REMOVE_DOCS=y
CT_INSTALL_DIR_RO=y
CT_STRIP_ALL_TOOLCHAIN_EXECUTABLES=y
```

```
#
# Downloading
#
# CT_FORBID_DOWNLOAD is not set
# CT_FORCE_DOWNLOAD is not set
CT_CONNECT_TIMEOUT=10
# CT_ONLY_DOWNLOAD is not set
# CT_USE_MIRROR is not set

#
# Extracting
#
# CT_FORCE_EXTRACT is not set
CT_OVERIDE_CONFIG_GUESS_SUB=y
# CT_ONLY_EXTRACT is not set
CT_PATCH_BUNDLED=y
# CT_PATCH_LOCAL is not set
# CT_PATCH_BUNDLED_LOCAL is not set
# CT_PATCH_LOCAL_BUNDLED is not set
# CT_PATCH_BUNDLED_FALLBACK_LOCAL is not set
# CT_PATCH_LOCAL_FALLBACK_BUNDLED is not set
# CT_PATCH_NONE is not set
CT_PATCH_ORDER="bundled"

#
# Build behavior
#
CT_PARALLEL_JOBS=1
CT_LOAD=0
CT_USE_PIPES=y
CT_EXTRA_FLAGS_FOR_HOST=""
# CT_CONFIG_SHELL_SH is not set
# CT_CONFIG_SHELL_ASH is not set
CT_CONFIG_SHELL_BASH=y
# CT_CONFIG_SHELL_CUSTOM is not set
CT_CONFIG_SHELL="${bash}"
```

```
#
# Logging
#
# CT_LOG_ERROR is not set
# CT_LOG_WARN is not set
# CT_LOG_INFO is not set
CT_LOG_EXTRA=y
# CT_LOG_ALL is not set
# CT_LOG_DEBUG is not set
CT_LOG_LEVEL_MAX="EXTRA"
# CT_LOG_SEE_TOOLS_WARN is not set
CT_LOG_PROGRESS_BAR=y
CT_LOG_TO_FILE=y
CT_LOG_FILE_COMPRESS=y

#
# Target options
#
CT_ARCH="arm"
CT_ARCH_SUPPORTS_BOTH_MMU=y
CT_ARCH_SUPPORTS_BOTH_ENDIAN=y
CT_ARCH_SUPPORTS_32=y
CT_ARCH_SUPPORTS_WITH_ARCH=y
CT_ARCH_SUPPORTS_WITH_CPU=y
CT_ARCH_SUPPORTS_WITH_TUNE=y
CT_ARCH_SUPPORTS_WITH_FLOAT=y
CT_ARCH_SUPPORTS_WITH_FPU=y
CT_ARCH_SUPPORTS_SOFTFP=y
CT_ARCH_DEFAULT_HAS_MMU=y
CT_ARCH_DEFAULT_LE=y
CT_ARCH_DEFAULT_32=y
CT_ARCH_ARCH="armv7-a"
CT_ARCH_CPU="cortex-a9"
CT_ARCH_TUNE="cortex-a9"
CT_ARCH_FPU=""
# CT_ARCH_BE is not set
CT_ARCH_LE=y
```

```
CT_ARCH_32=y
CT_ARCH_BITNESS=32
CT_ARCH_FLOAT_HW=y
# CT_ARCH_FLOAT_SW is not set
CT_TARGET_CFLAGS=""
CT_TARGET_LDFLAGS=""
# CT_ARCH_alpha is not set
CT_ARCH_arm=y
# CT_ARCH_avr32 is not set
# CT_ARCH_blackfin is not set
# CT_ARCH_m68k is not set
# CT_ARCH_mips is not set
# CT_ARCH_powerpc is not set
# CT_ARCH_s390 is not set
# CT_ARCH_sh is not set
# CT_ARCH_sparc is not set
# CT_ARCH_x86 is not set
CT_ARCH_alpha_AVAILABLE=y
CT_ARCH_arm_AVAILABLE=y
CT_ARCH_avr32_AVAILABLE=y
CT_ARCH_blackfin_AVAILABLE=y
CT_ARCH_m68k_AVAILABLE=y
CT_ARCH_mips_AVAILABLE=y
CT_ARCH_powerpc_AVAILABLE=y
CT_ARCH_s390_AVAILABLE=y
CT_ARCH_sh_AVAILABLE=y
CT_ARCH_sparc_AVAILABLE=y
CT_ARCH_x86_AVAILABLE=y

#
# Generic target options
#
# CT_MULTILIB is not set
CT_ARCH_USE_MMU=y
CT_ARCH_ENDIAN="little"

#
```

```
# Target optimisations
#
# CT_ARCH_FLOAT_SOFTFP is not set
CT_ARCH_FLOAT="hard"

#
# arm other options
#
CT_ARCH_ARM_MODE="arm"
CT_ARCH_ARM_MODE_ARM=y
# CT_ARCH_ARM_MODE_THUMB is not set
# CT_ARCH_ARM_INTERWORKING is not set
CT_ARCH_ARM_EABI=y

#
# Toolchain options
#

#
# General toolchain options
#
CT_FORCE_SYSROOT=y
CT_USE_SYSROOT=y
CT_SYSROOT_NAME="sysroot"
CT_SYSROOT_DIR_PREFIX=""
CT_WANTS_STATIC_LINK=y
CT_STATIC_TOOLCHAIN=y
CT_TOOLCHAIN_PKGVERSION=""
CT_TOOLCHAIN_BUGURL=""

#
# Tuple completion and aliasing
#
CT_TARGET_VENDOR="cortex_a9"
CT_TARGET_ALIAS_SED_EXPR=""
CT_TARGET_ALIAS=""
```

```
#
# Toolchain type
#
# CT_NATIVE is not set
CT_CROSS=y
# CT_CROSS_NATIVE is not set
# CT_CANADIAN is not set
CT_TOOLCHAIN_TYPE="cross"

#
# Build system
#
CT_BUILD=""
CT_BUILD_PREFIX=""
CT_BUILD_SUFFIX=""

#
# Misc options
#
# CT_TOOLCHAIN_ENABLE_NLS is not set

#
# Operating System
#
CT_KERNEL_SUPPORTS_SHARED_LIBS=y
CT_KERNEL="linux"
CT_KERNEL_VERSION="2.6.36.4"
# CT_KERNEL_bare_metal is not set
CT_KERNEL_linux=y
CT_KERNEL_bare_metal_AVAILABLE=y
CT_KERNEL_linux_AVAILABLE=y
# CT_KERNEL_V_3_5 is not set
# CT_KERNEL_V_3_4_7 is not set
# CT_KERNEL_V_3_3_8 is not set
# CT_KERNEL_V_3_2_25 is not set
# CT_KERNEL_V_3_1_10 is not set
# CT_KERNEL_V_3_0_39 is not set
```

```
# CT_KERNEL_V_2_6_39_4 is not set
# CT_KERNEL_V_2_6_38_8 is not set
# CT_KERNEL_V_2_6_37_6 is not set
CT_KERNEL_V_2_6_36_4=y
# CT_KERNEL_V_2_6_33_20 is not set
# CT_KERNEL_V_2_6_32_59 is not set
# CT_KERNEL_V_2_6_31_14 is not set
# CT_KERNEL_V_2_6_27_62 is not set
# CT_KERNEL_LINUX_CUSTOM is not set
CT_KERNEL_mingw32_AVAILABLE=y

#
# Common kernel options
#
CT_SHARED_LIBS=y

#
# linux other options
#
CT_KERNEL_LINUX_VERBOSITY_0=y
# CT_KERNEL_LINUX_VERBOSITY_1 is not set
# CT_KERNEL_LINUX_VERBOSITY_2 is not set
CT_KERNEL_LINUX_VERBOSE_LEVEL=0
CT_KERNEL_LINUX_INSTALL_CHECK=y

#
# Binary utilities
#
CT_ARCH_BINFMT_ELF=y

#
# GNU binutils
#
# CT_BINUTILS_V_2_22 is not set
# CT_BINUTILS_V_2_21_53 is not set
# CT_BINUTILS_V_2_21_1a is not set
CT_BINUTILS_V_2_20_1a=y
```

```
# CT_BINUTILS_V_2_19_1a is not set
# CT_BINUTILS_V_2_18a is not set
CT_BINUTILS_VERSION="2.20.1a"
CT_BINUTILS_2_20_or_later=y
CT_BINUTILS_2_19_or_later=y
CT_BINUTILS_2_18_or_later=y
CT_BINUTILS_HAS_HASH_STYLE=y
CT_BINUTILS_GOLD_SUPPORTS_ARCH=y
CT_BINUTILS_HAS_PKGVERSION_BUGURL=y
CT_BINUTILS_FORCE_LD_BFD=y
CT_BINUTILS_LINKER_LD=y
CT_BINUTILS_LINKERS_LIST="ld"
CT_BINUTILS_LINKER_DEFAULT="bfd"
CT_BINUTILS_EXTRA_CONFIG_ARRAY=""
# CT_BINUTILS_FOR_TARGET is not set

#
# C compiler
#
CT_CC="gcc"
CT_CC_VERSION="4.5.3"
CT_CC_gcc=y
# CT_CC_GCC_SHOW_LINARO is not set
# CT_CC_V_4_7_1 is not set
# CT_CC_V_4_7_0 is not set
# CT_CC_V_4_6_3 is not set
# CT_CC_V_4_6_2 is not set
# CT_CC_V_4_6_1 is not set
# CT_CC_V_4_6_0 is not set
CT_CC_V_4_5_3=y
# CT_CC_V_4_5_2 is not set
# CT_CC_V_4_5_1 is not set
# CT_CC_V_4_5_0 is not set
# CT_CC_V_4_4_7 is not set
# CT_CC_V_4_4_6 is not set
# CT_CC_V_4_4_5 is not set
# CT_CC_V_4_4_4 is not set
```

```
# CT_CC_V_4_4_3 is not set
# CT_CC_V_4_4_2 is not set
# CT_CC_V_4_4_1 is not set
# CT_CC_V_4_4_0 is not set
# CT_CC_V_4_3_6 is not set
# CT_CC_V_4_3_5 is not set
# CT_CC_V_4_3_4 is not set
# CT_CC_V_4_3_3 is not set
# CT_CC_V_4_3_2 is not set
# CT_CC_V_4_3_1 is not set
# CT_CC_V_4_2_4 is not set
# CT_CC_V_4_2_2 is not set
CT_CC_GCC_4_2_or_later=y
CT_CC_GCC_4_3_or_later=y
CT_CC_GCC_4_4_or_later=y
CT_CC_GCC_4_5=y
CT_CC_GCC_4_5_or_later=y
CT_CC_GCC_HAS_GRAPHITE=y
CT_CC_GCC_HAS_LTO=y
CT_CC_GCC_HAS_PKGVERSION_BUGURL=y
CT_CC_GCC_HAS_BUILD_ID=y
CT_CC_GCC_USE_GMP_MPFR=y
CT_CC_GCC_USE_MPC=y
CT_CC_GCC_USE_LIBELF=y
# CT_CC_LANG_FORTRAN is not set
CT_CC_SUPPORT_CXX=y
CT_CC_SUPPORT_FORTRAN=y
CT_CC_SUPPORT_JAVA=y
CT_CC_SUPPORT_ADA=y
CT_CC_SUPPORT_OBJC=y
CT_CC_SUPPORT_OBJCXX=y

#
# Additional supported languages:
#
CT_CC_LANG_CXX=y
# CT_CC_LANG_JAVA is not set
```

```
# CT_CC_LANG_ADA is not set
# CT_CC_LANG_OBJC is not set
# CT_CC_LANG_OBJCXX is not set
CT_CC_LANG_OTHERS=""

#
# gcc other options
#
CT_CC_ENABLE_CXX_FLAGS=""
CT_CC_CORE_EXTRA_CONFIG_ARRAY="--with-float=hard"
CT_CC_EXTRA_CONFIG_ARRAY="--with-float=hard"
CT_CC_STATIC_LIBSTDCXX=y
# CT_CC_GCC_SYSTEM_ZLIB is not set

#
# Optimisation features
#
# CT_CC_GCC_USE_GRAPHITE is not set
CT_CC_GCC_USE_LTO=y

#
# Settings for libraries running on target
#
CT_CC_GCC_ENABLE_TARGET_OPTSPACE=y
# CT_CC_GCC_LIBMUDFLAP is not set
# CT_CC_GCC_LIBGOMP is not set
# CT_CC_GCC_LIBSSP is not set

#
# Misc. obscure options.
#
CT_CC_CXA_ATEXIT=y
# CT_CC_GCC_DISABLE_PCH is not set
CT_CC_GCC_SJLJ_EXCEPTIONS=m
CT_CC_GCC_LDBL_128=m
# CT_CC_GCC_BUILD_ID is not set
```

```
#
# C-library
#
CT_LIBC="glibc"
CT_LIBC_VERSION="2.11"
# CT_LIBC_eglibc is not set
CT_LIBC_glibc=y
# CT_LIBC_uClibc is not set
CT_LIBC_eglibc_AVAILABLE=y
CT_LIBC_glibc_AVAILABLE=y
CT_LIBC_GLIBC_TARBALL=y
# CT_LIBC_GLIBC_V_2_14_1 is not set
# CT_LIBC_GLIBC_V_2_14 is not set
# CT_LIBC_GLIBC_V_2_13 is not set
# CT_LIBC_GLIBC_V_2_12_2 is not set
# CT_LIBC_GLIBC_V_2_12_1 is not set
# CT_LIBC_GLIBC_V_2_11_1 is not set
CT_LIBC_GLIBC_V_2_11=y
# CT_LIBC_GLIBC_V_2_10_1 is not set
# CT_LIBC_GLIBC_V_2_9 is not set
# CT_LIBC_GLIBC_V_2_8 is not set
CT_LIBC_mingw_AVAILABLE=y
CT_LIBC_newlib_AVAILABLE=y
CT_LIBC_none_AVAILABLE=y
CT_LIBC_uClibc_AVAILABLE=y
CT_LIBC_SUPPORT_THREADS_ANY=y
CT_LIBC_SUPPORT_NPTL=y
CT_THREADS="nptl"

#
# Common C library options
#
CT_THREADS_NPTL=y
CT_LIBC_XLDD=y
CT_LIBC_GLIBC_MAY_FORCE_PORTS=y
CT_LIBC_glibc_familly=y
CT_LIBC_GLIBC_EXTRA_CONFIG_ARRAY=""
```

```
CT_LIBC_GLIBC_CONFIGPARMS=""
CT_LIBC_GLIBC_EXTRA_CFLAGS=""
CT_LIBC_EXTRA_CC_ARGS=""
# CT_LIBC_ENABLE_FORTIFIED_BUILD is not set
# CT_LIBC_DISABLE_VERSIONING is not set
CT_LIBC_OLDEST_ABI=""
CT_LIBC_GLIBC_FORCE_UNWIND=y
CT_LIBC_GLIBC_USE_PORTS=y
CT_LIBC_ADDONS_LIST=""
# CT_LIBC_LOCALES is not set
# CT_LIBC_GLIBC_KERNEL_VERSION_NONE is not set
CT_LIBC_GLIBC_KERNEL_VERSION_AS_HEADERS=y
# CT_LIBC_GLIBC_KERNEL_VERSION_CHOSEN is not set
CT_LIBC_GLIBC_MIN_KERNEL="2.6.36.4"

#
# glibc other options
#

#
# WARNING !!!
#

#
#   For glibc >= 2.8, it can happen that the tarballs
#

#
#   for the addons are not available for download.
#

#
#   If that happens, bad luck... Try a previous version
#

#
#   or try again later... :-(
```

```
#

#
# Debug facilities
#
# CT_DEBUG_dmalloc is not set
# CT_DEBUG_duma is not set
# CT_DEBUG_gdb is not set
# CT_DEBUG_ltrace is not set
# CT_DEBUG_strace is not set

#
# Companion libraries
#
CT_COMPLIBS_NEEDED=y
CT_GMP_NEEDED=y
CT_MPFR_NEEDED=y
CT_MPC_NEEDED=y
CT_LIBELF_NEEDED=y
CT_COMPLIBS=y
CT_GMP=y
CT_MPFR=y
CT_MPC=y
CT_LIBELF=y
# CT_GMP_V_5_0_2 is not set
# CT_GMP_V_5_0_1 is not set
CT_GMP_V_4_3_2=y
# CT_GMP_V_4_3_1 is not set
# CT_GMP_V_4_3_0 is not set
CT_GMP_VERSION="4.3.2"
# CT_MPFR_V_3_1_0 is not set
# CT_MPFR_V_3_0_1 is not set
# CT_MPFR_V_3_0_0 is not set
CT_MPFR_V_2_4_2=y
# CT_MPFR_V_2_4_1 is not set
# CT_MPFR_V_2_4_0 is not set
CT_MPFR_VERSION="2.4.2"
```

```
# CT_MPC_V_0_9 is not set
# CT_MPC_V_0_8_2 is not set
CT_MPC_V_0_8_1=y
# CT_MPC_V_0_7 is not set
CT_MPC_VERSION="0.8.1"
CT_LIBELF_V_0_8_13=y
# CT_LIBELF_V_0_8_12 is not set
CT_LIBELF_VERSION="0.8.13"

#
# Companion libraries common options
#
# CT_COMPLIBS_CHECK is not set

#
# Companion tools
#

#
# READ HELP before you say 'Y' below !!!
#
# CT_COMP_TOOLS is not set

#
# Test suite
#
# CT_TEST_SUITE_GCC is not set
```

FAQ

Linux FAQs

This section provides answers to frequently asked questions about your release. Use it as the first step in troubleshooting problems. You can also try searching the Index in this document, contacting your support engineer, or filing a bug.

Linux FAQs

Are ARMv7 binaries compatible with aarch64 binaries?

No, while the kernel supports both ARMv7 (32bi) and aarch64 binaries, distros currently are exclusively aarch64 or ARMv7. ARMv7 binaries are not compatible in an aarch64 distro. The NVIDIA PDK supports aarch64 distro where ARMv7 binaries are not compatible.

How do I use display mode and resolution configuration with the X RandR application?

You can use the X Resize, Rotate and Reflect Extension (RandR) extension to manipulate and configure the attached displays (both the internal panel and any externally connected HDMI panel). The `xrandr(1)` utility is the most common way to do this.

You can find a tutorial on xrandr on the following website:

```
http://www.thinkwiki.org/wiki/Xorg_RandR_1.2
```

Are there generated ssh host keys for the sample file system?

There are no keys in the `/etc/ssh` directory of the provided sample file system. For information about creating the ssh host keys, see the `ssh-keygen` man page.

How do I determine the X driver ABI of the X server used in the root file system?

All `tegra_drv.abi*.so` files are in the driver package. By default the `apply_binaries.sh` script creates a sym-link from `tegra_drv.so` to the X ABI driver compatible with the provided sample file system.

How do I prevent the system display from blanking out?

Linux kernel 3.1 added a power saving feature that may blank the display of an idle system even when applications are running. The feature is called console blank (screen saver). It is defined as:

```
consoleblank= [KNL]
```

Where `[KNL]` is the console blank (screen saver) timeout in seconds. This defaults to `10*60 = 10 mins`. A value of `0` disables the blank timer.

By passing arguments to the kernel command line, you can:

- Disable this feature, or
- Set the timeout to a longer interval.

With the `flash.sh` script, you can override the kernel command line options passed from fastboot to the kernel.

To disable the console blank (screen saver) from the kernel command line

1. In the grub configuration add the following line to the kernel parameters:

```
consoleblank=0
```

2. View the current `consoleblank` value with the following command:

```
$ cat /sys/module/kernel/parameters/consoleblank
```

To disable the console blank feature with an escape sequence

- Enter the following escape sequence:

```
$ echo -ne "\033[9;0]"
```

To change the console blank timeout value with an escape sequence

- Enter the following escape sequence:

```
$ echo -ne "\033[9;<timeout>]"
```

  Where `<timeout>` is the timeout in seconds.

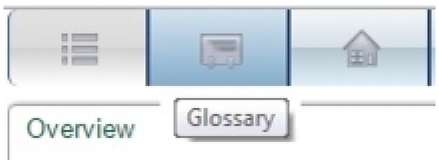
For more information on this escape sequence, see the `console_codes(4)` man page documents. For information on the input/output controls that provide some of the same functionality, see the `console_ioctl(4)` man page.

Glossary

This section provides definitions to frequently used terms in this release.

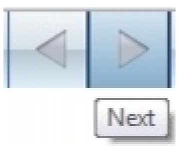
To browse the glossary by keyword

- Click the Glossary button on the tool bar (shown below).



To browse the glossary page-by-page

- Click the NEXT button (right arrow) on the toolbar.



Legal Information

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OR CONDITION OF TITLE, MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE AND ON-INFRINGEMENT, ARE HEREBY EXCLUDED TO THE MAXIMUM EXTENT PERMITTED BY LAW.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, CUDA, Tegra, and Vibrante are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

The Android robot is reproduced or modified from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

ARM, AMBA, and ARM Powered are registered trademarks of ARM Limited. Cortex, MPCore and Mali are trademarks of ARM Limited. All other brands or product names are the property of their respective holders. "ARM" is used to represent ARM Holdings plc; its operating company ARM Limited; and the regional subsidiaries ARM Inc.; ARM KK; ARM Korea Limited.; ARM Taiwan Limited; ARM France SAS; ARM Consulting (Shanghai) Co. Ltd.; ARM Germany GmbH; ARM Embedded Technologies Pvt. Ltd.; ARM Norway, AS and ARM Sweden AB.

Copyright

Index