



NVIDIA Jetson Linux

Release Notes

Version 35.1 GA

Table of Contents

About this Release	2
Login Credentials	4
What's New	4
Known Issues	8
General System Usability	8
Flashing	14
Camera	15
Multimedia	18
Display	18
Fixed Issues	20
Implementation Details	22
Downloading Jetson Linux Sources	22
Emulation Flash Configurations	24
Camera	25
Device Registration	25
Device Tree Overlay	25
Using the Jetson IO Tool	27
Multimedia	27
UEFI	27

1. About this Release

The NVIDIA® Jetson™ Linux 35.1 General Availability (GA) includes the Linux Kernel 5.10, the UEFI based Bootloader, the Ubuntu 20.04 based root file system, NVIDIA drivers, the necessary firmwares, toolchain, and more. This release supports development with NVIDIA Jetson AGX Orin™ 32GB module, NVIDIA Jetson AGX Orin™ Developer Kit and also with NVIDIA Jetson Xavier™ NX, and NVIDIA Jetson AGX Xavier™ developer kits and modules.

Important: This GA release can be used for production purposes.

Platform and Release Information

Description	Supported version
Host machine Linux distribution for flashing software onto Jetson devices	Ubuntu x64 18.04 or 20.04 (x64 distribution)
Sample rootfs derived from Ubuntu operating system to run on Jetson devices	Ubuntu 20.04
Supported Linux kernel version	5.10 LTS
Supported ARM architecture	aarch64

Description	Supported version
<p>Name of the configuration file used in flashing.</p> <p>Note: When you flash a configuration file with <code>flash.sh</code>, specify the configuration's basename, i.e. the file name without the <code>.conf</code> suffix.</p> <p>For a complete description of supported platforms and configuration names, see the <i>Jetson Modules and Configurations</i> table in Environment Variables.</p>	<p><code>jetson-agx-orin-devkit.conf</code>: Flashes a Jetson AGX Orin module (P3701-0000) or Jetson AGX Orin 32GB module (P3701-0004) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).</p> <p><code>jetson-agx-orin-devkit-as-nx-16gb.conf</code>: Flashes a configuration that emulates a Jetson Orin NX 16GB module on a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).</p> <p><code>jetson-agx-orin-devkit-as-nx-8gb.conf</code>: Flashes a configuration that emulates a Jetson Orin NX 8GB module on a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).</p> <p><code>jetson-agx-orin-devkit-as-jao-32gb.conf</code>: Flashes a configuration that emulates a Jetson AGX Orin 32GB module on a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).</p> <p><code>jetson-agx-xavier-devkit.conf</code>: Flashes a Jetson AGX Xavier module that is attached to a Jetson AGX Xavier Developer Kit reference carrier board.</p> <p><code>jetson-xavier-nx-devkit.conf</code>: Flashes QSPI-NOR and microSD card for Jetson Xavier NX (P3668-0000).</p> <p><code>jetson-xavier-nx-devkit-emmc.conf</code>: Flashes QSPI-NOR and eMMC for Jetson Xavier NX (P3668-0001).</p> <p><code>Jetson-agx-xavier-industrial.conf</code>: Flashes QSPI-NOR and eMMC for Jetson AGX Xavier</p>

Description	Supported version
	Developer Kit with Jetson AGX Xavier Industrial module (P2888-0008).
Board names, module names, and revision numbers	Refer to the Jetson FAQ for a detailed list of Jetson device information.
Toolchain for cross-compilation	Bootlin GCC 9.3 https://developer.nvidia.com/embedded/jetson-linux
Release Tag	jetson_35.1

1.1. Login Credentials

To create your login credentials, follow the system prompts at the first boot.

1.2. What's New

Here is the list of new features in this Jetson Linux 35.1 GA production public release:

- Adds support for Jetson AGX Orin 32 GB production module
- Enables Jetson AGX Orin Developer Kit to emulate performance of Jetson AGX Orin 64GB, Jetson AGX Orin 32GB, Jetson Orin NX 16GB and Jetson Orin NX 8GB modules.

Refer to [Emulation Flash Configurations](#) for more information.

- Security
 - [Secure Boot](#)¹.
 - [Disk Encryption](#).
 - Secure storage in [OP-TEE](#) using the normal world (REE) file system.
- Over The Air Updates:
 - [A/B Bootloader](#) and A/B Rootfs.
 - [Image Based OTA tools](#) to upgrade Jetson AGX Xavier or Jetson Xavier NX modules running JetPack 4.x releases.

- Bootloader:
 - UEFI support for USB and NVMe.
 - UEFI support for 4K Display.
- Power:
 - Low Power SC7 state.
 - Support for [Power Estimator](#) for Jetson AGX Orin and Jetson Xavier NX series.
- Display:
 - [Display Configuration Blob \(DCB\) tool](#) to configure display outputs.
 - HDMI 4K and Audio.
 - DisplayPort Multi Stream Transport (MTS) on Jetson AGX Orin.
- Kernel:
 - Real Time Kernel Patches.
- Sensor Processing Engine (SPE):
 - Sources for SPE firmware.
 - Support for I2C and Inter VM Communication (IVC).
- Multimedia:
 - AV1 encoding/decoding on Jetson AGX Orin.
 - Support for nvJPEG load balancing to use two nvJPEG engines in Jetson AGX Orin, which provides twice the encoding/decoding performance as compared to JetPack 5.0/5.0.1 Developer Preview releases.
 - Pre-Processing Enhancements (PPE) module.
This module provides a framework for various video encode enhancement algorithms. Currently in this release, Temporal Adaptive Quantization (TAQ) is supported.
 - New NVUtils Multimedia buffer management and buffer transformation API.
- Camera:
 - Argus support for YUV444 and Linear RGB888 output format for Jetson AGX Xavier and Jetson Xavier NX
 - HDR Support for Jetson AGX Orin.
Digital Overlap (DOL) Mode and Piecewise Linear (PWL) are supported.
 - Support for Error Resiliency on Jetson AGX Orin.
 - New Samples:
 - `argus_demosaicOutput` to demonstrate the CVOutput capability.
 - `argus_rawBayerOutput` to demonstrate raw capture by using argus with options available to enable/disable 3A/ISP to converge sensor exposure settings.
 - `argus_userAlternatingAutoExposure` to demonstrate the captures by using alternating exposure.
 - `argus_yuvOneshot` now supports the YUV444 and YUV420 formats.

- Graphics:
 - Support for Vulkan® 1.3 (including the Roadmap 2022 Profile) ¹.

See the [Vulkan 1.3 Announcement](#). Previous Jetson releases supported Vulkan 1.2.

- Support for Vulkan® SC 1.0 ¹.
 - Vulkan SC is a low-level, deterministic, robust API that is based on Vulkan 1.2.

This API enables state-of-the-art GPU-accelerated graphics and computation that can be deployed in safety-critical systems and that are certified to meet industry functional safety standards. Refer to <https://www.khronos.org/vulkansc/> for more information.

- Vulkan SC can also be invaluable for real-time, non safety critical embedded applications.

Vulkan SC is designed to increase determinism, provide predictable execution times, and reduce the application size by shifting the preparation of the run-time application environment offline or to application initialization. This process includes the offline compilation of graphics pipelines that define how the GPU processes data and static memory allocation. These options both enable detailed GPU control that can be rigorously specified and tested. For more details see <https://www.khronos.org/blog/vulkan-sc-overview>.

Note: Jetson support for Vulkan SC is **not** safety certified.

¹ The product is based on a published Khronos Specification and has been submitted to, and is expected to pass, the Khronos Conformance Process. Refer to www.khronos.org/conformance for more information about the current conformance status.

- Support for OpenWF™ Display 1.0.
 - OpenWF Display is a Khronos API for low overhead interaction with the native display driver on Jetson. This API allows interaction with Vulkan SC to display images.

Note: Jetson support for OpenWF Display is **not** safety certified.

To use the OpenWF Display APIs, use the `openwfd_nvsci_sample` sample.

- Starting with this release, JetPack 5.0.2, the complete GBM APIs are supported.
 - Generic Buffer Management (GBM) is an API that provides a mechanism to allocate buffers for graphics rendering. The handle it creates can be used to pass directly to Weston/Wayland to render or to initialize EGL and create render target buffers.
 - Earlier releases supported a subset of the GBM API called miniGBM. From JP 5.0.2 onwards, all GBM APIs are supported.
- Jetson Linux Sources are now available on Git and the Jetson Linux page.

Refer to [Downloading Jetson Linux Sources](#) for more information.

For more information about the adaptation and bringup for your custom carrier boards, refer to [Jetson Module Adaptation and Bringup](#) for the Bringup and Adaptation Guides for the Jetson AGX Orin, AGX Xavier, and Xavier NX platforms.

Refer to the [Jetson Linux Developer Guide](#) for Jetson Linux Documentation and also [Implementation Details](#) for more information about implementation details that cover a variety of topics.

2. Known Issues

This section provides details about issues that were discovered during development and QA but have not yet been resolved in this release.

2.1. General System Usability

The following general system usability-related issues are noted in this release.

Issue	Description
3697875	<p>If you installed CUDA 11.4.14 from JP 5.0.1 DP and earlier releases, the <code>apt</code> upgrade to JP 5.0 GA will fail. This occurs because in the JP 5.0 GA release, the <code>cuda-nvprof-11-4</code> package has been renamed.</p> <p>After the <code>apt</code> upgrade, to fix this issue, run the following command:</p> <pre>\$ sudo apt install --fix-broken -o Dpkg::Options::="--force-overwrite"</pre>
3693491	<p>On a Jetson system that is configured without <code>oem-config</code>, using the cloning functionality on the APP partition might fail. The issue occurs because during the one time resize action at the initial boot, <code>oem-config</code> adjusts the APP partition size to a multiple of 4096.</p> <p>To work around this issue, use the following patch:</p> <pre>diff --git a/rfs/usr/lib/nvidia/resizefs/nvresizefs.sh b/rfs/usr/lib/nvidia/resizefs/nvresizefs.sh index 37d3ad3..cd36266 100755 --- a/rfs/usr/lib/nvidia/resizefs/nvresizefs.sh +++ b/rfs/usr/lib/nvidia/resizefs/nvresizefs.sh @@ -1,6 +1,6 @@ #!/bin/bash</pre>

Issue	Description
	<pre> -# Copyright (c) 2019-2021, NVIDIA CORPORATION. All rights reserved. +# Copyright (c) 2019-2022, NVIDIA CORPORATION. All rights reserved. # # Redistribution and use in source and binary forms, with or without # modification, are permitted provided that the following conditions @@ -103,13 +103,19 @@ done support_resizefs="\${is_last}" fi + + if ["\${is_last}" == "true"]; then + get_app_size + max_available_app_size + size="+\${max_app_size}M" + fi } function get_app_size() { </pre>

Issue	Description
	<pre> # shellcheck disable=SC2086 partition_size="\$(cat /sys/block/\${block_dev}/\${root_dev}/size)" - echo "\$((partition_size/2/1024))" + cur_app_size="\$((partition_size/2/1024))" } function max_available_app_size() @@ -123,7 +129,7 @@ grep "last usable sector" \ awk '{print \$10}'" - echo "\$(((last_usable_sector - app_start_sector + 1)/2/1024))" + max_app_size="\$(((last_usable_sector - app_start_sector + 1)/2/1024))" } function parse_args() @@ -143,7 +149,7 @@ echo "ERROR: \${script_name} doesn't support this platform." exit 1 fi </pre>

Issue	Description
	<pre> - get_app_size + echo "\${cur_app_size}" exit 0 ;; -m --max) @@ -151,7 +157,7 @@ echo "ERROR: \${script_name} doesn't support this platform." exit 1 fi - max_available_app_size + echo "\${max_app_size}" exit 0 ;; -s --size) @@ -170,6 +176,8 @@ script_name="\$(basename "\${0}")" support_resizefs="false" size="0" +cur_app_size="0" +max_app_size="0" root_dev="" block_dev="" </pre>

Issue	Description
3586140 3688506	The shutdown request from the Desktop UI on Jetson AGX Orin might take up to a minute to trigger the <code>systemd</code> shutdown.
3538819	When you boot into the Ubuntu desktop, it takes nearly 1 minute for the desktop to display and become operational.
3445976	WiFi attachment points on Jetson Xavier NX are not listed after headless installation with default oem-config options.
3437699	ThunderStrike Bluetooth controller events on Jetson Xavier NX are not reflected in the <code>gstest-gtk</code> application.
3748116	<p>For a rootfs A/B enabled system, if the current rootfs (for example rootfs A), fails to boot up three consecutive times, the next reboot will boot to another rootfs (for example, rootfs B), as expected. However, when the user runs the <code>nvbootctrl</code> command to display rootfs slot status, the following error message is displayed:</p> <pre data-bbox="410 1241 1349 1791"> # nvbootctrl -t rootfs dump-slots-info Last login: Tue Aug 9 10:16:52 UTC 2022 on ttyTCU0 ubuntu@tegra-ubuntu:~\$ sudo nvbootctrl -t rootfs dump-slots-info Current rootfs slot: B Active rootfs slot: A num_slots: 2 slot: 0, retry_count: 0, status: unbootable </pre>

Issue	Description
	<pre data-bbox="410 338 1349 506">slot: 1, retry_count: 3, status: normal ubuntu@tegra-ubuntu:~\$</pre> <p data-bbox="410 573 1333 640">The message in red should display Active rootfs slot: B, but slot A is specified. This error does not have an impact and can be safely ignored.</p> <p data-bbox="410 674 1040 705">This message will be corrected in an upcoming patch.</p>
3747765	<p data-bbox="410 804 1255 871">There is a potential race condition when you access the QSPI controller between the UEFI runtime variable service and the Linux kernel.</p> <p data-bbox="410 905 1349 1005">This issue occurs because the firewall that prevents QSPI from being accessed by the kernel has not yet been enforced in the release. This issue will be fixed in the next release.</p> <p data-bbox="410 1039 1320 1106">To avoid this condition, do not proceed with any kernel access to QSPI where the UEFI runtime variable service will not be called.</p> <p data-bbox="410 1140 1065 1171">Note: This workaround does not apply for OTA updates.</p>
3747765	<p data-bbox="410 1274 1336 1341">The <code>Video_dec_drm</code> sample compilation becomes stuck and displays a blank screen when it is run on Jetson AGX Xavier.</p>
3728816	<p data-bbox="410 1436 1323 1503">Sometimes, the <code>power off</code> command on Jetson AGX Orin can take up to 15 minutes because the Wifi driver does not complete the shutdown callback.</p> <p data-bbox="410 1537 1049 1568">To work around this issue, manually turn off the power.</p>
3694884	<p data-bbox="410 1671 1287 1738">A reboot after an APT update from 34.1.1 to 35.1 might hang with following errors in the logs:</p> <pre data-bbox="410 1772 1141 1803">vdd-ac-bat: Underflow of regulator enable count</pre>

Issue	Description
	<pre>tegra-xusb-padctl 3520000.xusb_padctl: failed to disable port 1 VBUS: -22</pre> <p>To recover from this error, complete a manual reboot.</p>
3648210	<p>In the boot logs for Jetson AGX Orin, the following messages can be safely ignored:</p> <pre>UPHY: UPHY lane info table is empty in MB1 BCT.</pre> <pre>E> FUSE_ALIAS: Fuse alias on production fused part is not supported.</pre>
3742857	<p>Massfuse functionality is not currently supported in the 35.1 GA release. This functionality will be made available soon.</p>

2.2. Flashing

The following flashing-related issues are noted in this release.

Issue	Description
3601114 3601261	<p>The minimum recommended SD card size for flashing the complete JetPack SDK to an SD card for Jetson Xavier NX Developer Kit is 64 GB.</p>
3586898	<p>A complete JetPack installation on the Jetson Xavier NX production module on the 16GB EMMC fails.</p> <p>Workaround</p> <p>Starting with release 35.1, the SDK Manager offers the following options:</p>

	<ul style="list-style-type: none"> • A complete JetPack installation. • An installation of only the JetPack runtime components <p>The JetPack runtime installation does not include samples and documentation and is helpful for Jetson modules with limited storage and during production.</p>
--	--

2.3. Camera

The following camera-related issues are noted in this release.

Issue	Description
3727769	<p>In Orin with IMX 274, the sensor might have higher noise for an indoor or a low-light environment,</p> <p>This issue can be reduced by using a lower sharpness strength.</p>
3711513	<p>The CVOutput in RGB linear format, which is taken directly after demosaic, is not supported on Jetson Orin and Jetson Orin NX.</p>
3590329	<p>With Orin + OV5693, there might be purple saturation in some portions of the scene.</p>
3692128 3447132	<p>The E3331 (Cphy - IMX318) sensor fails to load and probe.</p> <p>To resolve this issue, before you flash, remove the camera-related DTBO filenames from the <code>OVERLAY_DTB_FILE</code> string in the corresponding <code><boardname>.conf</code> file.</p> <p>This file is used to flash the device by using the <code>sudo ./flash.sh <boardname> mmcblk0p1</code> command.</p>

Issue	Description
3643516	<p>By default, the IMX185 camera module has a <code>pca9570</code> GPIO expander that sets the daymode. If you do not enable the expander, the IR cut filter in the sensor is disabled, and daylight preview might have a pinkish tint.</p> <p>To enable the expander, run the following command:</p> <pre>\$ sudo modprobe pca9570</pre>
3589917	<p>The IMX390 GMSL sensor fails to probe.</p> <p>To enable the IMX390 sensor DT nodes for probing on Jetson AGX Xavier and Jetson AGX Orin:</p> <ol style="list-style-type: none"> 1. In the <code><board.conf></code> file, remove the <code>IMX185 DTBO</code> filename from the <code>OVERLAY_DTB_FILE</code> string. <p>This file is used in the command for flashing the device. For example, <code>jetson-xavier.conf</code> is used in the <code>sudo ./flash.sh jetson-xavier mmcblk0p1</code> command to flash galen.</p> <ol style="list-style-type: none"> 2. Flash the device by using the corresponding command for the board. 3. Use the <code>jetson-io</code> tool to detect the IMX390 sensor module and reboot the device. 4. Check the <code>uart</code> logs and the <code>/dev/video##</code> nodes to determine whether IMX390 gets probed.

Issue	Description
3586900	<p>: When using the camera sensor on Jetson AGX Xavier early boards (board ID lower than 2822-0000-500), the reset-GPIO pin, <code>TEGRA_MAIN_GPIO(H, 3)</code>, might not work. This causes the corresponding sensor to fail probe and streaming.</p> <p>Board IDs 2822-0000-500 and later do not have this problem. To check your device's board ID, run <code>cat /proc/device-tree/chosen/ids</code>.</p> <p>Workaround: Update the device-tree nodes for the sensor by replacing the original GPIO pin with <code>TEGRA_MAIN_GPIO(H, 4)</code>. For an example, see the DTB overlay for dual-imx274 sensor: <code>tegra194-p2822-camera-dual-imx274-overlay.dts</code>.</p>

2.4. Multimedia

The following issues are noted in this release related to multimedia.

Issue	Description
3740825	Vulkan applications X11 might run at higher fps than expected on Jetson AGX Xavier

2.5. Display

The following TensorRT-related issues are noted in this release.

Issue	Description
3695925	On Jetson AGX Orin, the display might intermittently go blank during the boot.
3734664	<p>For an MST configuration on Jetson AGX Orin, hotplugging the primary MST monitor will work the first time but additional hotplugs of the primary monitor will fail and the display will not appear. The only way to recover is to reboot the device.</p> <p>This issue is specific to the primary MST monitor, and there are no issues with hotplugging the secondary MST monitor. In a daisy-chained MST configuration, for example, the <i>primary</i> monitor refers to the monitor that is directly connected to the board.</p>

Issue	Description
3724559	<p>HDMI 4K@60Hz does not work on ACER Predator X27 Monitor connected to Jetson AGX Xavier.</p> <p>To work around this issue, change the resolution to 4K@30 or a lower resolution.</p>
3666376	<p>By default, after launching X Window, the display does not appear on the Asus XG279Q monitor.</p> <p>To work around this issue, set the mode to 2560x1440 by running the <code>xrandr --output DP-0 --mode 2560x1440</code> command.</p>

3. Fixed Issues

This section provides details about the issues that were resolved in this release.

Issue	Description
3660805	The SPE's IVC channel does not work with Jetson AGX Orin.
3657961	After an <code>apt upgrade</code> , the Xavier NVME SSD failed to boot.
3623353	Flashing Jetson Xavier NX 16GB fails on Jetpack 5.0.
3605453	Flashing the Jetson Orin Developer Kit with a custom carrier board with no EEPROM fails.
3603552	Flashing a custom carrier board with the Jetson Xavier modules fails.
3573905	Monitors with a DP++ interface are not supported.
3570293	Jetson Xavier NX: Jetson-IO might fail to configure the IMX477 sensor.
3499398	GPIO configuration utility does not work as expected in the Linux 5.10 kernel that is included in the Jetpack 5.0 DP release.
3712616	<code>nvdsp-init</code> does not support SBK/PKC-fused boards, so there will be no boot splash displayed on the screen.

Issue	Description
3692886	The Display MST does not work on Jetson AGX Orin and will be addressed in an upcoming release.
3431695	Watchdog nodes are not enabled on Jetson AGX Xavier Industrial.
3420652	Display does not resume after SC7 suspend/resume cycle.
3447132 3574718	Wake-on-Lan is not supported in release 35.1.
3689332	Jetson IO is not currently supported on Jetson AGX Orin 32GB.
3591721	Jetson AGX Orin: The NVIDIA logo that is displayed during shutdown is corrupted.
3591557	Jetson Xavier NX: The SD card image only works with Jetpack 5.0 Developer Preview Bootloader.

4. Implementation Details

4.1. Downloading Jetson Linux Sources

You can now sync sources that are related to Jetson Linux from the [NVIDIA](#) Git server and download the sources from the Jetson Linux page. To sync the sources from the Git server, select **one** of the following options:

- Use git clone to clone individual git repos locally. Check the table below for the URL information of the repos.

```
git clone <URL of the git repo>
```

For example, `git clone "https://nv-tegra.nvidia.com/r/linux-5.10"`.

- Use the `source_sync.sh` script in the Jetson Linux package to download the available sources.
 - a. To download all kernel related sources use the `-k` option.

```
./source_sync.sh -k [-t <release tag>]
```

- b. To download all sources other than kernel you can use the `-o` option.

```
./source_sync.sh -o [-t <release tag>]
```

If you do not include the `-k` or `-o` options, the available sources for that release tag will be synced.

Here are the Git repo URLs that can be used with the `git clone` command and their corresponding sources tar files.

Name of the corresponding tar file	Git Repo URL
Linux_for_Tegra/source/public/kernel_src.tbz2	https://nv-tegra.nvidia.com/r/linux-5.10
Linux_for_Tegra/source/public/nvidia_kernel_display_driver_source.tbz2	https://nv-tegra.nvidia.com/r/tegra/kernel-src/nv-kernel-display-driver.git
Linux_for_Tegra/source/public/libn1-3.5.0.tbz2	https://nv-tegra.nvidia.com/r/3rdparty/libn1/3.5.0.git
Linux_for_Tegra/source/public/nvsample_cudaprocess_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/cuda-src/nvsample_cudaprocess.git
Linux_for_Tegra/source/public/nvidia-x	https://nv-tegra.nvidia.com/r/tegra/gf

Name of the corresponding tar file	Git Repo URL
config_src.tbz2	x-src/nv-xconfig.git
Linux_for_Tegra/source/public/gstegl_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-egl.git
Linux_for_Tegra/source/public/gstjpeg_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-jpeg.git
Linux_for_Tegra/source/public/gst-nvarguscamera_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nvarguscamera.git
Linux_for_Tegra/source/public/gst-nvcompositor_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nvcompositor.git
Linux_for_Tegra/source/public/gst-nvtee_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nvtee.git
Linux_for_Tegra/source/public/gst-nv4l2camera_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nv4l2camera.git
Linux_for_Tegra/source/public/gst-nvvidconv_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nvvidconv.git
Linux_for_Tegra/source/public/gst-nvvideo4linux2_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/gst-nvvideo4linux2.git
Linux_for_Tegra/source/public/nvgstapps_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/nvgstapps.git
Linux_for_Tegra/source/public/libgstnvdrmvideosink_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/libgstnvdrmvideosink.git
Linux_for_Tegra/source/public/libgstnvvideosinks_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/libgstnvvideosinks.git
Linux_for_Tegra/source/public/libnvgstreamer_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/gst-src/libnvgstreamer.git
Linux_for_Tegra/source/public/nvsci_headers.tbz2	https://nv-tegra.nvidia.com/r/tegra/nv-sci-src/nvsci_headers.git
Linux_for_Tegra/source/public/atf_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/optee-src/atf.git
Linux_for_Tegra/source/public/nvidia-jetson-optee-source.tbz2	https://nv-tegra.nvidia.com/r/tegra/optee-src/nv-optee.git
Linux_for_Tegra/source/public/v4l2_libs_src.tbz2	https://nv-tegra.nvidia.com/r/tegra/v4l2-src/v4l2_libs.git

Refer to <https://github.com/NVIDIA/edk2-nvidia/wiki> for more information about the UEFI sources and compilation details for this release.

4.2. Emulation Flash Configurations

The Jetson AGX Orin Developer Kit can be used to emulate the performance of Jetson AGX Orin 32GB, Jetson AGX Orin 64GB, Jetson Orin NX 16GB, and Jetson Orin NX 8GB production modules. Emulation support helps to significantly reduce the time to market by kick starting development for any Jetson Orin production module on the Jetson AGX Orin Developer Kit.

During emulation, the GPU, CPU, and other hardware accelerators are configured according to the supported `nvpmodels` for the target platform. Refer to the [Jetson Linux Developer Guide](#) for a list of `nvpmodels` that are supported for a Jetson module.

For emulation, you can flash the Jetson AGX Orin Developer Kit by using the appropriate configuration file from the following table. For example, to emulate Jetson Orin NX 16GB on Jetson AGX Orin Developer Kit, use following command:

```
$ sudo ./flash.sh jetson-agx-orin-devkit-as-nx-16gb mmcblk0p1
```

Flash Config	Description
jetson-agx-orin-devkit-as-jao-32gb	Flashes a configuration that emulates a Jetson AGX Orin 32GB module on a Jetson AGX Orin module (P3701-0000), which is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).
jetson-agx-orin-devkit	Jetson AGX Orin Developer Kit module (P3701-0000), which has 32GB of memory, provides complete performance similar to the Jetson AGX Orin 64GB module. As a result, no separate emulation configuration is required. Users can use the config that flashes a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).
jetson-agx-orin-devkit-as-nx-16gb	Flashes a configuration that emulates a Jetson Orin NX 16GB module on a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).
jetson-agx-orin-devkit-as-nx-8gb	Flashes a configuration that emulates a Jetson Orin NX 8GB module on a Jetson AGX Orin module (P3701-0000) that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000).

Note: The Emulation mode does not emulate over the current throttling limits.

4.3. Camera

Because UEFI boot is enabled in this release, Camera Auto Detection will not work if the EEPROM ID is not configured for a camera sensor.

4.3.1. Device Registration

After you complete the driver development, you **must** add the new device's information to the system kernel device tree so it can be registered (instantiated) when the kernel boots. The following sections describe ways to register a new device.

Before you begin, ensure that you obtain the kernel source files.

4.3.2. Device Tree Overlay

Because UEFI boot is enabled in this release, the plugin manager is no longer supported. You must create a device tree overlay (DTB overlay or `.dtbo`) file to register the camera module.

If your camera module has an on-board EEPROM, and is programmed with a valid camera ID, at runtime, you can use the device tree overlay file to apply the overlay for a specific camera module and update the device tree entries with proper information. Using a device tree overlay with an EEPROM ID allows a system image to support multiple camera devices. To select a different camera, power down the device, replace the camera module, and reboot. The new module works automatically.

To create and apply a device tree overlay file:

1. Add the `.dtsi` file to the camera configuration `.dtsi` file.
2. Set the status of your device tree nodes to `disabled`.

```
imx185_cam0: imx185_a01a {
    status = "disabled";
};
```

3. Add the overlay information as fragments to a new `.dts` file.

```
<top>/hardware/nvidia/platform/t19x/common/kernel-dts/t19x-common-modules/tegra194-camera-overlay-file.dts
```

You can also see the camera DTB overlay files that are provided with the current release for examples.

4. Update the `.dts` file with the correct overlay information and a compatible string.

```
/ {
    overlay-name = "Jetson Camera Dual-IMX274";
    jetson-header-name = "Jetson AGX Xavier CSI Connector";
    compatible = "nvidia,p2822-0000+p2888-0001";

    fragment@0 {
        target= "<&imx185_cam0>";
        board_config {
            ids = "LPRD-dual-imx274-002" ;
            sw-modules = "kernel";
        };
        __overlay__ {
            status = "okay";
        };
        fragment@1 {
            . . .
        };
    };
};
```

5. To generate a `.dtbo` file, compile the `.dts` file.
6. **Before flashing**, move the `.dtbo` file to `flash_folder/kernel/dtb/`.
7. Add the following line to the `<board>.conf` file, which is used to flash the device.

```
OVERLAY_DTB_FILE="${OVERLAY_DTB_FILE},tegra194-camera-overlay-file.dtbo";
```

This line causes the following tasks to completed:

- If a specific camera board is found when the kernel boots, the override data is applied to that camera board's tree nodes.
- The tree nodes are made available for the system to use.

4.3.3. Using the Jetson IO Tool

If your camera module does not have an on-board EEPROM, you can use the same DTB overlay file to statically configure the board for the attached camera.

1. After you attach the camera module, apply the camera module's DTB overlay using the Jetson-IO tool, and reboot.

The new module will work immediately after Jetson Linux starts.

Note: You might have to delete the `board_config{}` node from the fragments in the DTB overlay file.

2. After you compile the `.dts` file to generate a `.dtbo` file, move the `.dtbo` file to `/boot` on the Jetson device, so that the Jetson-IO tool can recognize it.
3. Launch the Jetson-IO tool and configure the DTB overlay.

4.4. Multimedia

Buffer management and buffer transform, which are related to the API based on `nvbuf_utils` and defined in `nvbuf_utils.h`, have been deprecated in this release. From Jetson Linux release 34.0.1 EA and later, NVIDIA recommends that you use the new API based on `nvutils`, which is defined in `nvbufsurface.h` and `nvbufsurftransform.h`.

Refer to the *nvbuf_utils to NvUtils Migration Guide* for information about facilitating your migration to the new API.

4.5. UEFI

For fixes that were made in the UEFI sources after the release, go to the [UEFI GitHub](#).

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA, the NVIDIA logo, are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

Arm

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.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Copyright

© 2022 NVIDIA Corporation & Affiliates. All rights reserved.

