NVIDIA Jetson Linux

Release Notes

Version 36.4.3 GA

RN_10698-r36.4.3 | January 2025

Table of Contents

1. About this Release

The NVIDIA® Jetson™ Linux 36.4.3 General Availability (GA) release includes the Linux Kernel 5.15, the Ubuntu 22.04-based root file system, the UEFI-based Bootloader, NVIDIA drivers, the necessary firmware, toolchain, and more. This release supports all Jetson Orin production modules and Developer Kits.

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 20.04 or 22.04 (x64 distribution)
Sample rootfs derived from the Ubuntu operating system to run on Jetson devices.	Ubuntu 22.04
The supported Linux kernel version.	5.15 LTS
The supported ARM architecture.	aarch64

Description	Supported Version
Name of the configuration file used in flashing. **Note**: When you flash a configuration file with the flash.sh script, specify the configuration's basename, for example, the file name without the .conf suffix. Some of the products require flashing through initrd instead of flash.sh. For a complete description of supported platforms and configuration names, see the *Jetson Modules and Configurations* table in Environment Variables.	*jetson-agx-orin-devkit-industrial.conf*: Flashes the Jetson Orin industrial module (P3701-0008), which is attached to a Jetson Orin reference carrier board (P3737-0000). *jetson-orin-nano-devkit-super.conf*: Flashes one of the following modules that is attached to a Jetson Orin Nano Carrier board (P3768-0000) with boosted performance—25W power mode for Orin Nano modules, 40W power mode for Orin NX modules, and MAXN “Super Mode” for all Orin Nano and Orin NX modules: ● Jetson Orin Nano developer kit module with SD Card (P3767-0005) ● Jetson Orin Nano 8GB module (P3767-0003) ● Jetson Orin Nano 4GB module (P3767-0004) ● Jetson Orin NX 16GB module (P3767-0000) ● Jetson Orin NX 8GB module (P3767-0001) For details on Super mode, refer to Supported Modes and Power Efficiency in the NVIDIA Jetson Linux Developer Guide. *jetson-orin-nano-devkit.conf*: Flashes one of the following modules that is attached to a Jetson Orin Nano Carrier board (P3768-0000): ● Jetson Orin Nano developer kit module with SD Card (P3767-0005) ● Jetson Orin Nano 8GB module (P3767-0003) ● Jetson Orin Nano 4GB module (P3767-0004) ● Jetson Orin NX 16GB module (P3767-0000) ● Jetson Orin NX 8GB module (P3767-0001) *jetson-agx-orin-devkit.conf*: Flashes one of the following modules that is attached to a Jetson AGX Orin Developer Kit (P3730-0000) reference carrier board (P3737-0000): ● Jetson AGX Orin developer kit module (P3701-0000) ● Jetson AGX Orin 32GB module (P3701-0004) ● Jetson AGX Orin 64GB module (P3701-0005)
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 11.3 https://developer.nvidia.com/embedded/jetson-linux
Release Tag.	jetson_36.4.3

1.1. Login Credentials

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

1.2. What's New

Jetson Linux 36.4.3 is the latest production quality Jetson Linux release that supports NVIDIA JetPack™ 6.2. The following are the highlights for this release:

- Support for new reference power modes on Jetson Orin Nano and Jetson Orin NX production modules, delivering up to 2x generative AI performance available with a new flashing configuration: `jetson-orin-nano-devkit-super.conf`).
  - NVIDIA Jetson Orin Nano 4GB: Supports `10W`, `25W`, and `MAXN SUPER`.
  - NVIDIA Jetson Orin Nano 8GB: Supports `15W`, `25W`, and `MAXN SUPER`.
  - NVIDIA Jetson Orin NX 8GB: Supports `10W`, `15W`, `20W`, `40W`, and `MAXN SUPER`.
  - NVIDIA Jetson Orin NX 16GB: Supports `10W`, `15W`, `25W`, `40W`, and `MAXN SUPER`.
- CVE resolutions and minor bug fixes.

Here is some additional information:

- Jetson Linux Sources are available on Git in addition to the Jetson Linux page (refer to Working with Sources for more information).

  For more information about the adaptation and bring up process for your custom carrier boards, refer to Jetson Module Adaptation and Bringup for the Jetson AGX Orin, Orin NX, and Orin Nano platforms.

- Refer to the Jetson Linux Developer Guide for Jetson Linux documentation and 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

4480028	With some USB cables that are shipped with Jetson AGX Orin, you might see the following errors in the logs:
	`usb usb2-port2: Cannot enable. Maybe the USB cable is bad?`
	`usb usb2-port2: config error`
	To stop the error messages, swap the USB cables.

4845581	The GPIO `sysfs` node has been deprecated, and `/sys/class/gpio` cannot be used to control the GPIO state. To use GPIO from the userspace, use the new GPIO character device API (for example, `libgpiod`).
4201479	Although we support flashing Jetson-Linux BSP in multiple boot media like USB and NVMe, the media should have the same version of BSP. An attempt to flash different BSP versions to multiple boot media will lead to a system crash in the UEFI because different overlays are flashed in the UEFI partition.
4450559	On the Jetson AGX Orin series of products, enabling the Wake-On-Lan (WOL) feature is not recommended when users connect to these products using SSH or when the target is booting from NFS. In both cases, users will lose connection with the target.

2.2. Flashing

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

Issue	Description
4357750	On the Ubuntu 18.04 Linux host, the `sudo ./apply_binaries.sh` installation step shows the following messages: **`qemu: Unsupported syscall: 293`** Reinstalling the `qemu-user-static package` on the Linux host sometimes eliminates the messages. These messages are typically harmless and everything still functions properly.
4229251	On some Linux hosts during flashing, the following message appears in `dmesg` logs, followed by flashing failures: **`Cannot enable. Maybe the USB cable is bad?`** To work around the issue, try connecting to a different USB port on your host machine. If this step does not help, change the USB cable or reboot the host machine.

4695663	Initrd flash could fail when flashing with high speed USB port (>= 10Gbps); in that case, the flashing will fail when trying to untar the file system like this:
	
	`tar: Read checkpoint 590000`
	`tar: Read checkpoint 600000`
	`Erased 67108864 bytes from address 0x00000000 in flash`
	`Flash failure`
	`Either the device cannot mount the NFS server on the host or a flash command has failed. Debug log saved to /tmp/tmp.QwrriPM9V9.`
	`You can access the target's terminal through "sshpass -p root ssh root@fc00:1:1:0::2"`
	
	To fix this, you can try to put an USB hub between the host and the target or use a lower speed USB port.

2.3. Camera

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

Issue	Description
4845798	For the HAWK Stereo camera [AR0234 sensor], the first captured image is dark when using the `argus_oneshot` or `argus_camera` applications.
4807063	In the latest `v4l2-utils`, the `VIDIOC_G/S_PARM` test fails during the `v4l2-compliance` test suite.
4264284	On Jetson Orin, captured images using `argus_camera` with DOL HDR sensors may show a marginal increase in noise.
4685929	There may be issues while running `argus_camera` with the error "JPEG parameter struct mismatch: library thinks size is 584, caller expects 728". This is due to a mismatch in the JPEG library where a 3rd party JPEG encoding library was being picked up. Users can WAR this issue by PRELOADING the NVIDIA JPEG library before running the argus_camera app: `"export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/nvidia/libnvjpeg.so"`

2.4. Multimedia

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

Issue	Description
4146738	The deinterlace feature is not supported in this release.
4419907	VK rendering is not supported for the AV1 MVC decoder feature.

2.5. Display

The following display issues are noted in this release.

Issue	Description
4385047	Hotplugging the display using DP to Jetson AGX Orin after bootup might lead to a corrupted screen.

Issue	Description
4324714	The secondary display connected to Jetson AGX Orin and Jetson Orin NX/Nano shows as connected in the `xrandr` utility, but `gdm` is not rendered on the screen.
4618846	The CableCreation branded DP-to-HDMI converter does not work as expected with Jetson AGX Orin. We recommend that you use a different DP-to-HDMI converter.
4840276	Blank screen after SC7 on Orin AGX, if the Orin AGX is booted with multi user target mode (ubuntu desktop is disabled and VT terminal is enabled),
5042888	When whitepoint is adjusted with ctm_property, it will not apply to the cursor (if the cursor is enabled).
4796028	Vulkan runs for first time but it fails during re-run after killing the application
4796011	VT switching won't effect when Vulkan D2D apps is running

2.6. Compute Stack

The following issues related to Deep Learning are noted in this release.

Issue	Description
4564075	To run VPI PVA algorithms in a Docker container, the same VPI version has to be installed on the Docker host.

2.7. DeepStream

The following issues related to video streaming are noted in this release.

Issue	Description
4325898	The pipeline gets stuck when using `multifilesrc` with `nvv4l2decoder`. DeepStream developers use this pipeline to run decode and inferJPEG images.

3. Fixed Issues

This section provides details about issues that were noted as issues in the previous releases but fixed in the current release.

Issue	Description
4385287	When only one IMX219 is connected to Jetson Orin NX 16GB/8GB, and a dual IMX219 overlay is applied, the preview and capture with `argus_samples` and `argus_camera` fails. To work around this issue, connect the dual IMX219 cameras when applying dual overlay support.
4327644	When an IMX219 or an IMX477 camera is connected to Jetson Orin NX and Jetson Orin Nano devices through the CAM0 interface, to correctly complete a camera recording, users need to explicitly specify sensor-id=1 instead of the default `sensor-id=0` in the gstreamer `nvarguscamera` element.
4389380	A half-preview image of IMX477 is observed when you run `argus_camera` for Orin Nano 4GB SKU4.
4389112	Multimedia sample compilation fails due to a missing header file, and the following message is displayed: **`NvJpegDecoder.cpp:36:10: fatal error: jpegint.h: No such file or directory`**
4359668	Applying the 3840x2160@30/24Hz resolution on a display that is connected to Jetson Orin NX makes the display go blank and the following error message is displayed: **`No VSI InfoFrame exists on two video fields.`**
4349663	When building TensorRT engines for DLA, there is a known issue where the entire DLA subgraphs that are listed in *Layers Running on DLA* (seen with TensorRT's verbose mode) cannot be built and eventually falls back to the GPU with the following message: **`{ForeignNode[...]} cannot be compiled by DLA, falling back to GPU.`** This has been observed with the following ResNet-based models: • PeopleNet v2.6 • TrafficCamNet from TAO. In both cases, to fix this issue, change TensorRT's default DLA SRAM pool size from **1** MiB to **0.5** MiB. When using `trtexec` to build the TensorRT engine, add the `--memPoolSize=dlaSRAM:0.5` argument. For other TensorRT applications that directly call TensorRT APIs, refer to this code section in `trtexec`.

Issue	Description
4391619	`Resnet-10 batch_size=1` GPU-only inferencing is broken on Jetson AGX Orin. Currently, syncpoint submission is updated to the end of `cudaEventRecord()` after all the operations have finished, so that after syncpoint is complete, there should be no pending work on the marker. However, when the application thread invokes `cudaEventElapsedTime`, but the GPU is still busy and has not yet completed its work, it will fail. To work around this issue and avoid the application execution failure, invoke the `jetson_clocks` utility to max out the SOC clocks and speed up the execution. This workaround only improves the symptom and is not guaranteed to work every time.
4361621	DRC support in `gst` pipelines is a work in progress and will require some changes in the video codec.
4297071	The latest version of GStreamer may have issues with the RTP stack, potentially causing intermittent segmentation faults.

4617111
4620917
	The TSEC CBB errors might be observed on Jetson AGX Orin during bootup, shutdown, suspend, and resume when the target is connected to a display with HDCP 2.2 support. This issue is the result of the missing synchronization between Tsec and DCE/display engines. The HDCP can be disabled by following patch which prevents the CBB errors.
	
	`--- a/nv-soc/tegra234-soc-display-overlay.dtsi`
	`+++ b/nv-soc/tegra234-soc-display-overlay.dtsi`
	`@@ -176,7 +176,6 @@`
	`"dpaux0_reset",`
	`"dsi_core_reset",`
	`"mipi_cal_reset";`
	`- hdcp_enabled;`
	`status = "disabled";`
	`memory-region = <&fb0_reserved>;`
	`nvidia,disp-sw-soc-chip-id = <0x2350>;`

4185596	Waking up from Deep Sleep state (SC7) by USB events is not supported in the NVIDIA JetPack 6.1 GA release for the Jetson Orin Series of products.
4494706	Previous releases of JetPack used the legacy Ethernet interface naming scheme (eth0, eth1, etc.). Starting with this release, it uses the predictable naming scheme for Ethernet interfaces. As part of this improvement, the parameter `net.ifnames=0` has been removed from the kernel command line.
4412365	During the Debian update process on Jetson AGX Orin with an external NVMe connected and flashed using SDK Manager, you may see the following message: **`FuPluginUefiCapsule cannot find default ESP: More than one available ESP`**

Issue	Description

	WARNING: UEFI ESP partition not detected or configured
	See https://github.com/fwupd/fwupd/wiki/PluginFlag:esp-not-found for more information.
	
	To recover from this issue:
	1. Stop the Debian update.
	2. Run the following commands:
	`$ sudo mkdir /boot/efi`
	`$ sudo mount /dev/nvme0n1p10 /boot/efi`
	`$ sudo vim /etc/fwupd/uefi_capsule.conf`
	`change the OverrideESPMountPoint=/boot/efi`
	3. To trigger the OTA again, run the following command:
	`$ sudo apt dist-upgrade`

4641609	ToneMap V3 is enabled for the HAWK AR0234 sensor by default on the ISAAC NOVA platform.
4624028 4599191 4585557	Improved the stability of the `argus_syncstereo` app to avoid crashes in the `nvargus` daemon during heavy CPU load.
4600459	Faded colors that appeared during camera preview with the HAWK [AR0234] sensor were fixed, improving image quality (IQ) using the latest ISP tuning file.
4588292	Improved synchronization and stability in the `argus_syncstereo` app.
4460069	Fixed the detection of the IMX390 [GMSL] camera sensor.
4264284	IMX390 WDR mode is enabled in this release, and issues associated with line corruption issue have been fixed
4262318	These v4l2 compliance tests previously failed or were invalid but have now been fixed in the latest v4l2 utils: 1. `ioctls` 2. `VIDIOC_REQBUFS/CREATE_BUFS/QUERYBUF`
4247835	Line corruption issues associated with v4l2 have been fixed.
4697575	Memory leak issues while running the `gstreamer` app and during encode/decode use cases have been fixed in the Argus library.

4. Implementation Details

This section provides information about implementation details.

4.1. Camera

Here are the camera integration changes compared to previous Jetson Linux 35.x releases:

Note: We recommend that all camera drivers be packages like Loadable Kernel Module (LKM) for JetPack 6 and later.

- Sensor kernel Drivers:
  Driver source code for the supported sensors are located in the `<TOP>/kernel/nvidia-oot/drivers/media/i2c/` directory.
  To obtain a complete understanding of the driver, examine this source file.
- LKM
  The camera and sensor drivers are loadable modules.
  The rebuilt sensor driver will be in the `<OUT>/l4t-generic-release-aarch64/nvidia/kernel-jammy-src/kernel-nvidia-oot/nvidia-oot/drivers/media/i2c/<camera>.ko` directory.
- On root file system, the loadable modules are located in the following directories:
  - Camera sensor driver:
    `/lib/modules/5.15.116-release-tegra/extra/drivers/media/i2c/`.
  - Camera driver:
    `/lib/modules/5.15.116-release-tegra/extra/drivers/media/platform/tegra/camera`.
  - RTCPU driver:
    `/lib/modules/5.15.116-release-tegra/extra/extra/drivers/platform/tegra/rtcpu/`.
  - Nvhost VI driver:
    `/lib/modules/5.15.116-release-tegra/extra/drivers/video/tegra/host/vi/`.
  - Nvhost CSI driver:
    `/lib/modules/5.15.116-release-tegra/extra/drivers/video/tegra/host/nvcsi/`.
- Device Registration
  After driver development is complete, you must add the new device information to the system kernel device tree so it can be registered (instantiated) when the kernel boots.
  To register your device, use one of the following methods:
  - **Device-tree overlay**
    You need to create a device-tree overlay file to register the camera module. If your camera module has on-board EEPROM and is programmed with a valid camera ID,

you can use that to apply the overlay for a specific camera module and update the device-tree entries with proper information at runtime. Using DTB overlays with EEPROM-ID allows a single system image to support multiple camera devices.

To change camera modules, power down the device, replace the camera module, and then reboot. The new module works automatically.

- **Create and apply a DTB Overlay file**
  1. Add your .dtsi file to the camera configuration .dtsi file.
     `tegra234-p3737-camera-modules.dtsi` is for Jetson AGX-orin.
  2. Set the status of your device tree nodes to `disabled`.

```
imx274_cam0: imx274_a@1a {
       status = "disabled";
};
```

  3. Add the overlay information as fragments below to a new .dts file.
     You can also refer to the camera DTB overlay files provided with the current release:
     `<top>/hardware/nvidia/t23x/nv-public/overlay/camera-overlay-file.dts`
  4. Update the .dts file with proper overlay information and a compatible string

```
/{
    overlay-name = "Jetson Camera Dual-IMX274";
   jetson-header-name = "Jetson AGX CSI Connector";
    compatible = "nvidia,p3737-0000+p3701-0000",
   "nvidia,p3737-0000+p3701-0004", "nvidia,p3737-0000+p3701-0005",
   "nvidia,p3737-0000+p3701-0008";

    fragment@0 {
        target-path = "/bus@0/i2c@3180000/tca9546@70/i2c@0/imx274_a@1a";
          board_config {
                ids = "LPRD-dual-imx274-002";
                sw-modules = "kernel";
          };
          __overlay__ {
                status = "okay";
          };
    };
        fragment@1 {
            … … …
        };
};
```

  5. Compile the .dts file to generate a .dtbo file.
  6. Before you flash, move the .dtbo file to `flash_folder/kernel/dtb/`.

7. Add the following lines to the `<board>.conf` file, which is used to flash the device. For Jetson AGX-orin board, the config file is `p3737-0000-p3701-0000.conf`. `OVERLAY_DTB_FILE="${OVERLAY_DTB_FILE},tegra234-p3737-camera-dual-imx274-overlay.dtbo"`.

4.2. 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.

4.2.1. 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_a@1a {
    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 {
                         . . .
                 };

  };
```

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

```
OVERLAY_DTB_FILE="${OVERLAY_DTB_FILE},tegra 194-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.

4.2.2. 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.

4.3. UEFI

For fixes that were made in the UEFI sources after the release, go to 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 NONINFRINGEMENT, 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.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com

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

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com