NVIDIA Jetson Linux

Release Notes

Version 38.2.1 GA

RN_10698-r38.2.1 | September 2025

Table of Contents

1. About This Release

The NVIDIA® Jetson™ Linux 38.2.1 General Availability (GA) release includes the Linux Kernel 6.8, the UEFI-based Bootloader, the Ubuntu 24.04-based root file system, NVIDIA drivers, the necessary firmware, toolchain, and more. This release supports the NVIDIA Jetson® Thor™ platforms.

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

1.1. Platform and Release Information

Description	Supported Version
Host machine Linux distribution for flashing software on the hardware platforms supported by this release.	Ubuntu 24.04 and 22.04
Sample rootfs that was derived from Ubuntu to run on Tegra devices that are supported in this release.	Ubuntu 24.04
Supported Linux kernel version.	6.8
Supported Arm® architecture.	aarch64
Name of the configuration file used in flashing. **Note:** When you flash a configuration file with `flash.sh`, specify the configuration's basename; for example, the file name without the `.conf` suffix.	For details, refer to Quick Start in the Jetson Linux Developer Guide.
Toolchain for cross-compilation.	GCC 13.2
BSP tag for the source push.	jetson_38.2.1

1.2. Login Credentials

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

1.3. What's New

Jetson Linux 38.2.1 is the latest Jetson Linux release that supports NVIDIA JetPack™ 7.0. With Jetson Linux 38.2.1, Jetson software aligns with the Server Base System Architecture (SBSA), positioning Jetson Thor alongside industry-standard ARM server design. The following are the highlights for this release:

- Minor stability and bug fixes for issues noted in the 38.2 BSP release.

For information about a variety of implementation details, refer to Implementation Details in these release notes.

2. Flashing Support

This section provides information about flashing support.

2.1. Flashing the Complete Jetson Linux Release

1. Extract and configure the Jetson Linux BSP and root file-system.

```
Shell
L4T_RELEASE_PACKAGE=Jetson_Linux_R38.2.0_aarch64.tbz2
SAMPLE_FS_PACKAGE=Tegra_Linux_Sample-Root-Filesystem_R38.2.0_aarch64.tbz2

$ tar xf ${L4T_RELEASE_PACKAGE}
$ sudo tar xpf ${SAMPLE_FS_PACKAGE} -C Linux_for_Tegra/rootfs/
$ cd Linux_for_Tegra/
$ sudo ./tools/l4t_flash_prerequisites.sh
$ sudo ./apply_binaries.sh --openrm
```

2. Place the device in recovery mode.
3. Flash the device.

```
Shell
# Jetson AGX Thor Developer Kit
sudo ./l4t_initrd_flash.sh jetson-agx-thor-devkit internal
```

4. When the flashing is complete, reset the board.

2.2. Flashing Only the Boot Firmware

To flash only the boot firmware, use the following commands:

```
Shell
# Jetson AGX Thor Developer Kit
sudo ./l4t_initrd_flash.sh --no-flash jetson-agx-thor-devkit internal
sudo ./l4t_initrd_flash.sh --qspi-only jetson-agx-thor-devkit internal
```

2.3. Using the apply_binaries.sh Script

Jetson Thor supports the OpenRM driver architecture. When using the `apply_binaries.sh` script, you *must* add the option `--openrm`, as follows:

```
Shell
$ sudo ./apply_binaries.sh --openrm
```

Using the command without any options installs nvgpu drivers by default, which is not supported:

```
Shell
#Installs nvgpu drivers. Do not use for Jetson Thor.

$ sudo ./apply_binaries.sh
```

If `apply_binaries.sh` was run without the `--openrm` option, do one of the following to recover *before* flashing:

- Clean up your local directories and redo the setup instructions.
  or
- Run the following command:

  ```
  sudo ./apply_binaries.sh --openrm --rootfs-tar <path-to-rootfs>
  ```

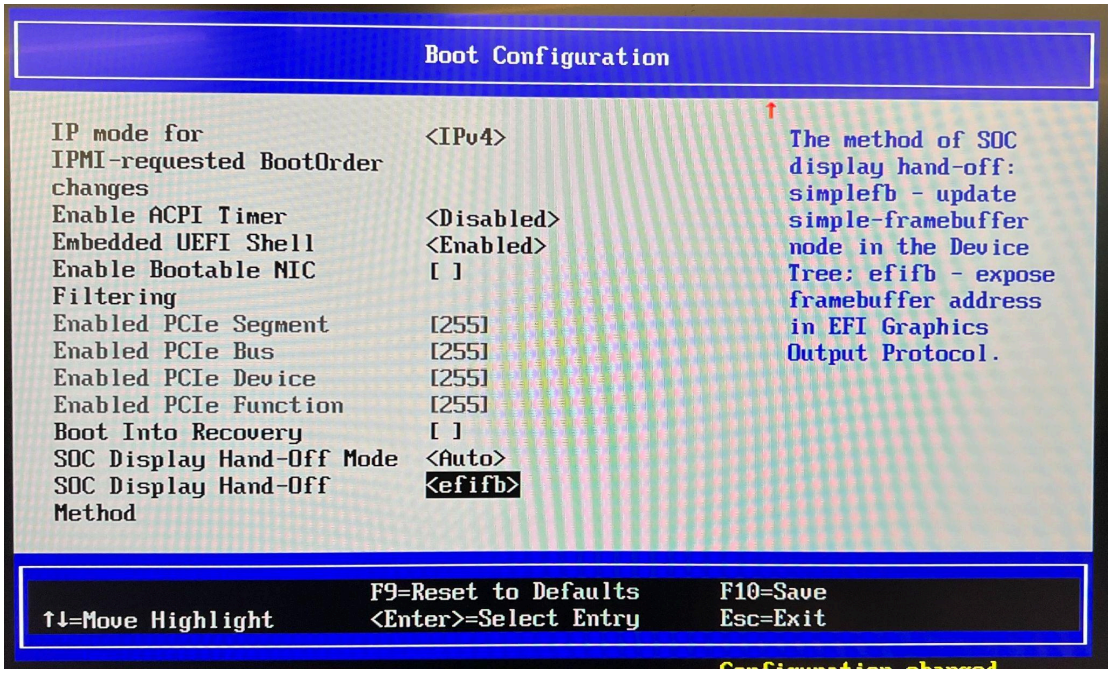
3. Changes for ISO Reinstallation

When re-installing ISO after the QSPI on the unit is updated through the capsule update, you must make the following changes so that the unit can boot up and the display can function.

Note: These steps do not need to be performed when booting the unit for the first time. But any time after the first boot in case of reinstalling the ISO these steps need to be repeated.

1. Press ESC to enter the UEFI menu.
2. Navigate to **Device Manager** > **NVIDIA Configuration** > **Boot Configuration** and make the following changes:

- SOC Display Hand-Off Mode: **Auto**
- SOC Display Hand-Off Method: **efifb**



3. Install the ISO from USB.

4. Known Issues

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

4.1. General System Usability

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

Issue	Description
5437763	When debug cable is removed and re-inserted, minicom becomes inaccessible. Restarting minicom will fix the issue.
5367003	Suspend/resume cycle could sometimes lead to reboot of the unit.
5424568	ISO installation fails when USB with the image is plugged into a USB hub “USB3.0 4-port Portable Hub Model UH400”. Other USB sticks or hubs work as expected.
5437041	During the OEM Config process, navigating back to previous pages to reenter the computer name could throw an error saying that the name is not unique even though the configuration was not yet saved.
5463861	When installing ISO in headless mode without display connected the UEFI console `/dev/ttyACM1` is filled with a lot of output making the ISO install hard. This issue is resolved after QSPI is updated to GA (38.2) release. Issue is also not seen when a display is connected to the target.
5460707	Flashing to target sometimes fails when a Trendnet TU2-ET100 USB-to-Ethernet dongle is connected. We did not notice issues with other USB-to-Ethernet dongles.
5116642	For Jetson Thor, the OC event counter (reported by `/sys/class/hwmon/hwmon*/oc*event_cnt`) starts at 1 even when no OC event has occurred.
5406663	Currently, you can monitor GPU utilization by using `nvidia-smi dmon`. However, because of design changes, integration of GPU utilization in Jetson Power GUI is still under evaluation**.**

Issue	Description
5130696	In Jetson Thor, ASPM for PCIe devices might not be enabled due to the low-power state exit latency of PCIe controllers, specifically Gen5 speed. To avoid the issue, configure the speed of the PCIe controller to lower Gen.

4.2. Camera

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

Issue	Description
5383804	Flashing of FPGA FW might fail if setup is not properly configured. To prevent FPGA FW failure, ensure the following steps are followed: 1. Verify that the camera is connected and that you can ping it. 2. Use `hololink-enumerate` to capture your current FPGA FW version. 3. Ensure that you are using the correct flash script and FW version for the type of camera you are using, per the user guide instructions. 4. When flashing starts, do not interrupt the process.
5449309	The error NOTIF_ERROR_ICP_BAD_INPUT_STREAM might be observed for the first few frames. This is expected because HSB generates sequence numbers that need to be matched and thus skips frames as required.
5456784	Using Jetson IO to configure Jetson Camera Hawk-Owl p3762 module could sometimes lead to the display being stuck.
5461056 5398396	Argus crash can be seen while running preview/capture use cases for E3333[OV5693], E3331[IMX318], or IMX185, due to the opening sequence of streaming only. Use `override.cfg` files for the sensors to avoid this issue.
5500614	VB1940 may generate corrupted images during overnight long-run tests in preview mode.
5461150	Failed to stream with AR1335. nvv4l2camerasrc: Internal data stream error AR1335 uses a discrete framerate not supported by nvv4l2camerasrc. Use v4l2src + nvvidconv to avoid the issue.
5476402	The first two or three frames generated from VB1940 are corrupted. Image capture should be done after that.

4.3. Multimedia

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

Issue	Description
5436810	CUDA BL memory is not supported for encode and decode use cases.
5451745	De-interlace decode is not supported.
5437529	Slice-level encoding is not supported.
5236938	Init time delay for multi-instance decode use cases.
5452012	GStreamer `nvsiplsrc` plug-in requires `nvvidconv` to copy the buffers because it does not work directly with the encoder.
5216744	GStreamer `nvv4l2decoder` plug-in does not support JPEG/MJPEG decoding.
5388059	USB camera preview fails with OSS Cheese application due to conflicts between OSS and NVIDIA JPEG library.
5416070	DVFS is not fully optimal for multimedia use cases
4506985	The `emc_log` tool is not showing correct memory bandwidth for iGPU-based NVJPG HW engine on Jetson Thor.
5466051	The NVIDIA FFMPEG HW acceleration is currently not supported on Jetson Thor. If your applications require FFMPEG, you can use the open source version with the following command: `sudo apt-get install -y ffmpeg`

4.4. Graphics

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

Issue	Description
—	—

4.5. Connectivity

The following connectivity issues are noted in this release.

Issue	Description
5226667 5465140	Wi-Fi 6-GHz routers that use MBSSID are not supported.
5426982	Sometimes in a busy environment, not all Wi-Fi access points are observed due to limited buffer size for scan results. As a workaround, run `wpa_cli set bss_max_count 500` to increase the `wpa_supplicant` buffer size.

4.6. Display

The following display issues are noted in this release.

Issue	Description
5424504	UEFI menu is slow to respond during ISO installation on some 4K monitors. This issue does not affect the functionality.
5380828	Toggling fractional scaling options in display settings on desktop could lead to some flickering on the background.
5437402	When two displays are connected to target one over HDMI and another over DP, the NVIDIA logo gets clipped during boot up and reboot.
5422032	In a dual-monitor setup, enabling desktop scaling may cause window corruption or visual artifacts
5378590	Changing display orientation to Landscape Flipped or Portrait Right or Portrait Left does not work.

4.7. Compute Stack

The following issues related to various compute components are noted in this release.

Issue	Description

5092683	Fisheye python samples fail to run because of OpenCV compatibility issues. To work around this issue, install the following:
	`sudo python3 -m pip install opencv-python==4.8.0.74;`
	`sudo python3 -m pip install pillow numpy==1.26;`

5228336 5462083	(For details, refer to the VPI Release Note.)

5. Fixed Issues

The following issues were fixed in this release.

Issue	Description
5451920	GStreamer `nv3dsink` plug-in hangs in DRC use cases.
5461913	Image captures with IMX274 in WDR mode and SDR mode are blurry at times. This is due to tuning of noise parameters that is required in this mode. The issue is fixed with this release.
5504026 5470960	Issues with Multi HSB streaming are fixed. Can now support up to a maximum of 16 cameras (through 8x HSBs).
5445933 5472056 5446087	Intermittent failures could be observed when pinging cameras over Ethernet. When running SIPL example applications, random I2C errors and capture failures are seen because of UDDF driver header incompatibility. Beginning with this release, the following WAR is no longer needed. 1. `sudo cp /usr/lib/nvsipl_uddf/libnvuddf_eagle_library.so /tmp/libnvuddf_eagle_library.so_orig` 2. `cd /usr/src/jetson_sipl_api/sipl/` 3. `sudo mkdir build` 4. `cd build` 5. `sudo cmake ..`

Issue	Description
	6. `sudo make nvsipl_coe_camera` 7. `sudo cp ./samples/coe_camera/nvsipl_coe_camera ~/` 8. `sudo cp /tmp/libnvuddf_eagle_library.so_orig /usr/lib/nvsipl_uddf/libnvuddf_eagle_library.so` 9. Run the app `nvsipl_coe_camera`.
5435196	Jetson Thor can sometimes get stuck during continuous boot stress cycles and will need a cold reset to recover.
5455397	Fix segmentation fault in nvsipl_query_sample.

5454877	The `nvsipl_coe_camera` application fails to run inside a Docker container. Use the following workaround to fix the problem.
	
	In the file `/etc/nvidia-container-runtime/host-files-for-container.d/drivers.csv`, add the following lines:
	
	`lib, /usr/lib/nvsipl_uddf/libnvuddf_eagle_library.so`
	`lib, /usr/lib/nvsipl_drv/libnvsipl_qry_vb1940.so`
	`lib, /var/nvidia/nvcam/settings/sipl/vb1940.nito`

4755448	Fix PCIe completion timeout issues during the PTP operation.
5406304	Fix issues related to UART where standard baudrates were not supported.
5440020	Bluetooth re-connections for LE keyboard and mice do not work on the initial pair. They will work after device reboot or Bluetooth reboot or toggling the connected device from settings page.

6. Implementation Details

This section provides information about implementation details.

6.1. Camera Serial Interface

The following are the camera integration changes compared to previous Jetson Linux 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 `<OUT>/l4t-generic-release-aarch64/nvidia//kernel-noble/kernel-nvidia-oot/nvidia-oot/drivers/media/i2c/<camera>.ko`.

  On root file system, the loadable modules are located in the following directories:

  - Camera sensor driver: `/lib/modules/6.8.12-tegra/updates/drivers/media/i2c/`.
  - Camera driver: `/lib/modules/6.8.12-tegra/updates/drivers/media/platform/tegra/camera/`.
  - RTCPU driver: `/lib/modules/6.8.12-tegra/updates/drivers/platform/tegra/rtcpu/`.
  - Nvhost VI driver: `/lib/modules/6.8.12-tegra/updates/drivers/video/tegra/host/vi/`.
  - Nvhost CSI driver: `/lib/modules/6.8.12-tegra/updates/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. For Jetson AGX Orin, the file is `tegra264-p3971-camera-modules.dtsi`.

  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/t264/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,p3971-0089+p3834-0008",
     "nvidia,p3971-0050+p3834-0005", "nvidia,p3971-0080+p3834-0008",
     "nvidia,p3834-0008", "nvidia,tegra264";

         / {
             overlay-name = "Jetson Camera Dual-IMX274";
             jetson-header-name = "Jetson AGX CSI Connector";
             compatible = JETSON_COMPATIBLE;
             fragment@0 {
                  target-path = "/";
                  board_config {
                      ids = "LPRD-dual-imx274-002";
                      sw-modules = "kernel";
                  };
     ```

```
        __overlay__ {
...
```

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 line to the `<board>.conf` file, which is used to flash the device. For the Jetson AGX Orin board, the config file is `p3834-0008-p3971-0000.conf`.

   ```
   OVERLAY_DTB_FILE="${OVERLAY_DTB_FILE},tegra264-p3971-camera-dual-imx274-overlay.dtbo"
   ```

6.2. 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. This section describes how to register a new device.

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

Because UEFI boot is enabled in this release, the plug-in 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`.

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

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

   ```
   <top>/hardware/nvidia/t264/nv-public/overlay/tegra264-p3971-camera-over
   lay-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 CSI Connector";
           compatible = "nvidia,p3971-0089+p3834-0008";

   fragment@0 {
        target-path = "/";
        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},tegra264-p3971-camera-dual-imx274-overlay
   .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.

6.3. CoE Camera Use Cases and SIPL Example Applications

All Implementation details for CoE Camera use cases are available in Camera Development using CoE in the Jetson Linux Developer Guide.

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