NVIDIA Jetson Linux

Release Notes

Version 36.2 Developer Preview

RN_10698-r36.2 | December 2023

Table of Contents

1. About this Release

The NVIDIA® Jetson™ Linux 36.2 Developer Preview (DP) 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 also supports all Jetson Orin production modules and Developer Kits.

> **Note**: This release is DP quality and should not be used for production purposes. Refer to Known Issues for a list of the known issues in this release.

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. 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.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) *jetson-agx-orin-devkit-as-nx-16gb.conf:* 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.conf*: 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). *jetson-agx-orin-devkit-as-jao-32gb.conf*: 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).
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_36.2

1.1. Login Credentials

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

1.2. What's New

Jetson Linux 36.2 is the first Jetson Linux release that supports NVIDIA JetPack™ 6. Jetson Linux 36.2 is a developer preview release and includes LTS Kernel 5.15 and the Ubuntu 22.04-based root file system.

Here are the highlights for this release:

- Supports all Jetson Orin modules and developer kits.
- LTS Kernel 5.15.
- Ubuntu 22.04-based root file system.
- Flexibility to bring any upstream Linux Kernel.
- Expanded choices of Linux based distros.

  Refer to the JetPack 6 Developer Preview page for more information about the Linux-based distro options.

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
4185596	Jetson AGX Orin Developer Kit and Jetson AGX Industrial modules could intermittently fail to resume after suspend.
4323950	All Jetson devices sometimes see an intermittent TLP packet Malform issue with the `rtl8822ce` WiFi driver when you see the following messages in the error log: **`[ 163.849288] rtl88x2ce 0001:01:00.0: AER: can't recover (no error_detected callback)`** When this issue occurs, the WiFi will not be functional and ethernet might also get dropped. To recover from this issue, reboot the device.
4191812	On NVIDIA IGX, when the bandwidth request from Display/VI comes before the nvpmodel EMC clock capping request, the request from VI/Display takes precedence because the VI/Display cannot handle the latency of a DRAM switch without a fatal error. When the issue occurs, the EMC clock is uncapped and power might go high. You need to select the nvpmodel power mode with the appropriate EMC clock setting.
4403124	`ramoops` is not enabled in this build.

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 fails with the following error: **`qemu: Unsupported syscall: 293`** To work around the issue, update the `qemu-user-static` package on the Linux host that is used to install to the latest version.
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.

2.3. Camera

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

Issue	Description
4385287	When only a single IMX219 is connected to Jetson Orin NX 16GB/8GB and dual IMX219 overlay is applied, the preview and capture with argus_samples and argus_camera fails. To work around this issue, connect dual IMX219 cameras when applying dual overlay support.
4327644	When IMX219 or IMX477 camera is connected to Jetson Orin NX and Jetson Orin Nano devices through 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.

Issue	Description
4389285	AR1335: YUV: gstreamer fails to stream with `nvv4l2camerasrc` because sensor IOCTL, which is used to get the format and resolution, is failing.
4389380	A half-preview image of IMX477 is observed when you run `argus_camera` for Orin Nano 4GB SKU4.
4264284	On Jetson AGX Orin, images captured using argus cameras with DOL HDR Sensors might show a marginal increase in noise. A fix for this issue will be provided in the next release.

2.4. Multimedia

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

Issue	Description
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`** This issue will be fixed by using `apt update` after the release.
4146738	The deInterlace feature is not supported in this release.
4401805	Starting a `video_drc_drm` with the `disable-video` option leaves the rendered in a bad state. To resolve this issue, reboot the system.

2.5. Display

The following Display issues are noted in this release.

Issue	Description
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.`**
4385047	Hotplugging the display using DP to Jetson AGX Orin after the bootup might lead to a corrupted screen.
4324714	The secondary display that is connected to Jetson AGX Orin and Jetson Orin NX/Nano shows as being connected on the `xrandr` but `gdm` is not rendered on the screen.

2.6. Compute Stack

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

Issue	Description
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.
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.

2.7. Deepstream

Issue	Description
4361621	DRC support in gst pipelines is a work in progress and will require some changes in the Video Codec.
4297071	The new Gstreamer version might have some issues with the RTP stack and might lead to intermittent segmentation faults.

Issue	Description
4325898	The pipeline gets stuck for multifilesrc when using `nvv4l2decoder`. DS developers use the pipeline to run decode and infer jpeg images.

3. Implementation Details

This section provides information about implementation details.

3.1. 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"`.

3.2. 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.

3.2.1. 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.

3.2.2. 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.

3.2.3. Adaptation to the Carrier Board with HDMI for the Orin NX/Nano Modules

If you are using a third-party carrier board that supports HDMI, ensure that the following patch is applied in your `mb2 scr bct` in the `./bootloader/t186ref/BCT/tegra234-mb2-bct-scr-p3767-0000.dts` file:

```
--- a/firewall/tegra234-mb2-bct-scr-p3767-0000.dts
+++ b/firewall/tegra234-mb2-bct-scr-p3767-0000.dts
@@ -5,6 +5,11 @@

 / {
     tfc {
+        reg@322 { /* GPIO_M_SCR_00_0 */
+            exclusion-info = <2>;
+           value = <0x38009696>;
+        };
```

```
+
         reg@5138 { /* CBB_CENTRAL_CBB_FIREWALL_QSPI0_BLF, READ_CTL */
             exclusion-info = <2>;
             value = <0x00100009>;
```

This change is needed because the `GPIO_M_0 GPIO` pin is used for the HDMI hotplug. In Orin, the access to this pin should be limited **only** to the DCE firmware.

3.3. 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