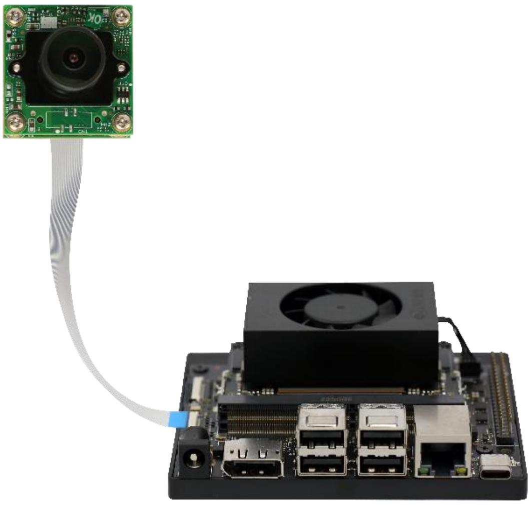
e-CAM25_CUONX

Developer Guide



Version 1.6

e-con Systems

3/24/2025

e-con Systems®

Think Vision. Think e-con.

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

Contents

Introduction to e-CAM25_CUONX

e-CAM25_CUONX is a 2.3 MP MIPI CSI-2 camera from e-con Systems, a company with over two decades of experience in designing, developing, and manufacturing OEM cameras.. The prebuilt driver for this camera along with the camera board is provided by e-con Systems.

The NVIDIA® Orin Nano™ development kit is a small, powerful computer for embedded applications and Artificial Intelligence (AI) Internet of Things (IoT). It is pre-flashed with a Linux environment, includes support for many common APIs, and is supported by NVIDIA® complete development toolchain. Orin NX™/Orin Nano™ platforms supports single camera for 4-lane configuration in cam 1 port only and dual camera support in 2-lane configuration.

e-CAM25_CUONX is 2.3 MP custom lens camera module based on 1/2.6" AR0234CS CMOS image sensor from onsemi™. It is a color camera which supports UYVY image format and provided with S-mount (also known as M12 board lens) lens holder. The S-mount is small form-factor lens mounts for board cameras. e-con Systems provides the sample applications that demonstrates the features of this camera. e-CAM25_CUONX camera is interfaced to J21 connectors of the Orin Nano™ development kit using the ACC-RB-WTB-ADP board. However, this camera can also be utilized by any Video for Linux version 2 (V4L2) application.

e-con Systems also provides ecam_tk1_guvcview sample application that demonstrates the features of this camera. However, this camera can utilize any Video for Linux version 2 (V4L2) application.

The commands and output messages in this manual are represented by different colors as listed in below table.

Table 1: Notation of Colors

Color	Notation
Blue	Commands running in host PC
Red	Output message in host PC
Green	Output message in Jetson development kit
Orange	Commands running in Jetson development kit

This document explains how to setup the Jetson Orin NX™/Orin Nano™ development kit for using eCAM25_CUONX.

Software Requirements

The software requirements are as follows:

- Cross compiler toolchain
- Linux for Tegra (L4T) release package and sample root filesystem (rootfs)

Prerequisites for Jetson Orin NX

This section describes the requirements to use e-CAM25_CUONX on the Jetson Orin Nano™ development kit with Jetson Orin NX™ SOM.

The prerequisites are as follows:

- Host PC which runs Ubuntu 22.04 (64-bit) or Ubuntu 20.04 (64-bit).
- Download and install lbzip2 package.
- NVIDIA® provided L4T release and corresponding sample rootfs for Jetson Orin NX™ platform.
- The USB Type-C cable to plug into the recovery port of the Jetson Orin Nano™ development kit.
- Power cable (19V) to power the Jetson Orin Nano™ development kit.
- 64GB USB boot drive [SD Card/Pendrive] must be connected to the respective slot.

Please refer to the *e-CAM25_CUONX_Release_Package_Manifest_Rev_<ver>.pdf* to know the contents of release package and their description.

Setting Up Environment

The steps to setup the environment are as follows:

1. Run the following commands to setup the required environment variables.

```
mkdir top_dir -p
mkdir top_dir/tool_chain
export TOP_DIR=<absolute_path_to>/top_dir
export RELEASE_PACK_DIR=$TOP_DIR/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_<release_da
te>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export TOOL_CHAIN=$TOP_DIR/tool_chain
export CROSS_COMPILE=$TOOL_CHAIN/aarch64--glibc--
stable-2022.08-1/bin/aarch64-buildroot-linux-gnu-
export NVIDIA_SRC=$L4T_DIR/source/
export TEGRA_KERNEL_OUT=$NVIDIA_SRC/out/nvidia-linux-
header
export KERNEL_HEADERS=$NVIDIA_SRC/kernel/kernel-jammy-
src
export INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
export SENSOR_DRIVER=e-CAM25_CUONX
```

2. Run the following command to copy the release package tar file to the staging directory.

```
mv <location_of>/e-CAM25_CUONX_JETSON_ONX_ONANO_
<L4T_version>_<release_date>_<release_version>.tar.gz
$TOP_DIR
```

Downloading Requirements

For building the kernel, a cross compiler toolchain and other tools necessary for compiling are required.

The steps to download the requirements for building the kernel are as follows:

1. Download the required toolchain from NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

Table 2: Required Toolchain

S.NO	Title	Version	Download link
1	GCC Toolchain	11.3.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v3.0/toolchain/aarch64--glibc--stable-2022.08-1.tar.bz2

a. Run the following commands to extract the package in host PC.

```
mv $HOME/Downloads/aarch64--glibc--stable-2022.08-
1.tar.bz2 $TOOL_CHAIN
cd $TOOL_CHAIN
tar -xf aarch64--glibc--stable-2022.08-1.tar.bz2
```

b. Run the following command to add CROSS_COMPILER to PATH environment variable for building kernel source.

```
export PATH=$TOOL_CHAIN/aarch64--glibc—stable-
2022.08-1/bin:$PATH
```

2. Run the following commands to download the required package for extracting sources.

```
sudo apt-get update
sudo apt-get install qemu-user-static
sudo apt-get install build-essential
sudo apt-get install bc
sudo apt-get install lbzip2
sudo apt-get install python
sudo apt-get install flex
sudo apt-get install openssl
sudo apt-get install libssl-dev
```

3. Download the required L4T release package and sample rootfs from NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

   The steps to download and copy the package to staging directory are as follows:

   a. Download the packages from the NVIDIA® website as listed in below table.

Table 3: Packages for Jetson Orin NX

S.NO	Title	Version	Download Link
1	L4T Jetson Orin NX™, Jetson Orin Nano™	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/release/Jetson_Linux_R36.4.0_aarch64.tbz2
2	L4T Jetson Orin NX™ Sample Rootfs	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/release/Tegra_Linux_Sample-Root-Filesystem_R36.4.0_aarch64.tbz2

   b. Run the following commands to copy the downloaded package to staging directory.

```
cp $HOME/Downloads/Jetson_Linux_R36.4.0_aarch64.tbz2
$TOP_DIR
cp $HOME/Downloads/Tegra_Linux_Sample-Root-
Filesystem_R36.4.0_aarch64.tbz2 $TOP_DIR
```

Extracting and Preparing L4T

The steps for extracting and preparing L4T must be performed in host PC are as follows:

1. Run the following commands to extract the downloaded L4T release package to navigate a folder with the name Linux_for_Tegra.

```
cd $TOP_DIR
tar xf Jetson_Linux_R36.4.0_aarch64.tbz2
```

2. Run the following commands to extract the sample rootfs to the rootfs directory which is present inside the Linux_for_Tegra directory.

```
sudo tar xpf Tegra_Linux_Sample-Root-
Filesystem_R36.4.0_aarch64.tbz2 -C
Linux_for_Tegra/rootfs/
```

3. Run the following commands to set the package to be ready to flash binaries.

```
cd $L4T_DIR
sudo ./tools/l4t_flash_prerequisites.sh
sudo ./apply_binaries.sh
```

Extracting Release Package

Run the following commands to extract the e-CAM25_CUONX release package.

```
cd $TOP_DIR

tar -xaf e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to the *e-CAM25_CUONX_Release_Package_Manifest_Rev_<ver>.pdf*.

e-con Systems®
Think Vision. Think e-con.

Installation Procedure for Jetson Orin NX

This section describes the steps for building and installing the kernel.

Building from Source

You can use the patch files provided by e-con Systems to use your own kernel image binary and modules along with the e-CAM25_CUONX camera on the Jetson Orin NX™ SOM + Jetson Orin Nano™ development kit to build the kernel.

Downloading and Configuring Kernel

This section describes how you can download and configure the kernel for Jetson Orin NX™ SOM.

Download the kernel source code for L4T from the NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

The steps to download and configure the kernel Jetson Orin NX™ SOM + Jetson Orin Nano™ development kit are as follows:

1. Download the packages from the NVIDIA® website as listed in below table.

Table 4: Packages for Jetson Orin NX

S.NO	Title	Version	Download Link
1	L4T Jetson Orin NX™ + Jetson Orin Nano™ Sources	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/sources/public_sources.tbz2

2. Run the following command to copy the downloaded file to staging directory.

```
cp $HOME/Downloads/public_sources.tbz2 $TOP_DIR
```

3. Run the following commands to extract the downloaded kernel source code to any path on the host Linux PC.

```
tar xf public_sources.tbz2
cd $NVIDIA_SRC
```

4. Run the following commands to extract the kernel source code.

```
tar xf kernel_src.tbz2
tar xf kernel_oot_modules_src.tbz2
tar xf nvidia_kernel_display_driver_source.tbz2
```

5. Run the following command to make sure that the patch command is applied properly in the nvidia-oot source.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_oot.patch --dry-run
```

6. Run the following command to apply the patch file to the nvidia-oot source code if there is no error from dry-run command.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_oot.patch
```

7. Run the following command to make sure that the patch command is applied properly in the device tree source.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_dtb.patch --dry-run
```

8. Run the following command to apply the patch file to the device tree source code if there is no error from dry-run command.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_dtb.patch
```

9. Run the following command to make sure that the patch command is applied properly in the module source to build the sensor module.

```
mkdir -p sensor_driver

patch -d sensor_driver -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_module.patch --dry-run
```

10. Run the following command to apply the sensor module patch file to the sensor module source code if there is no error from dry-run command.

```
patch -d sensor_driver -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_module.patch
```

Building and Installing Kernel

The steps to build and install the kernel are as follows:

1. Run the following commands to build and install the kernel image,modules, sensor driver and dtbo to the Jetson Orin NX™ SOM.

```
make -C kernel

make modules

sudo -E make modules_install

make dtbs
```

2. Run the following commands to copy the dtbo file to Linux_for_Tegra (L4T_DIR) flashing path.

   For 2 lane configuration,

```
sudo cp $NVIDIA_SRC/kernel-devicetree/generic-
dts/dtbs/tegra234-p3767-0000-p3768-0000-a0-2lane-
ar0234.dtbo $LDK_ROOTFS_DIR/boot/
```

   For 4 lane configuration,

```
sudo cp $NVIDIA_SRC/kernel-devicetree/generic-
dts/dtbs/tegra234-p3767-0000-p3768-0000-a0-4lane-
ar0234.dtbo $LDK_ROOTFS_DIR/boot/
```

   Note: Even if the image is custom built, the kernel configuration must have module versioning support for the camera driver to work.

Flashing Jetson Orin NX SOM + Jetson Orin Nano Development Kit

The steps to flash the Jetson Orin NX™ are as follows:

1. Power OFF the kit.
2. Connect the Orin NX™ SOM to the Jetson Orin Nano™ development kit.
3. Connect a jumper across FC-REC and GND to enter recovery mode.
4. Connect the Type-C USB cable to the Jetson Orin Nano™ development kit and host PC.

   The top view of Jetson Orin Nano™ development kit is shown below.

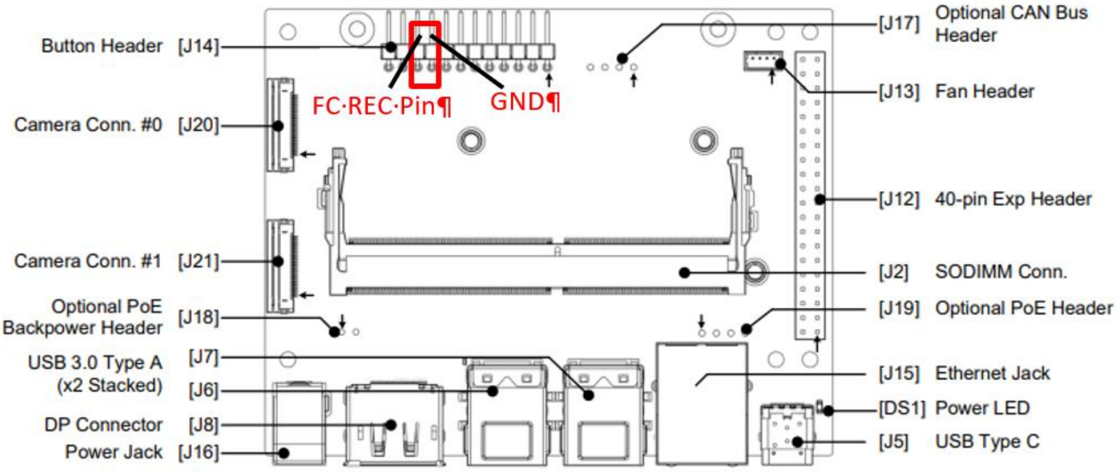


Figure 1: Top View of Jetson Orin Nano Development Kit

5. Power ON the Jetson Orin Nano™ development kit.

   If the Jetson Orin Nano™ development kit is successfully changed to recovery mode, the Jetson Orin NX™ will be enumerated as an USB device to the host PC.

6. Run the following command to verify whether the Jetson Orin Nano™ development kit is in recovery mode.

```
lsusb
```

The output message appears as shown below.

```
Bus 003 Device 023: ID 0955:7323 NVidia Corp.
```

7. Run the following commands to flash Jetson Orin NX™ SOM + Jetson Orin Nano™ development kit from your host PC.

```
cd $L4T_DIR
1.Flash to NVMe
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
external-device nvme0n1p1 -c
tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
--network usb0 jetson-orin-nano-devkit internal
2. Flash to the USB
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
external-device sda1 -c
tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
--network usb0 jetson-orin-nano-devkit internal
3. Flash to SD card
sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
external-device mmcblk0p1 -c
tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
--network usb0 jetson-orin-nano-devkit internal
```

Note: If you are flashing with a USB flash drive as storage, the USB drive must be at least 128GB.

Note: Now, the entire USB drive storage/Non-volatile Memory Express (NVMe) on the Jetson Orin Nano™ development kit will be erased. Flashing will take about 10-30 minutes to complete depending on the host PC configuration.

8. Connect the Jetson Orin Nano™ development kit to a monitor and keyboard.
9. Reboot the Jetson Orin NX™ development kit to complete the OS configuration, once the Jetson Orin NX™ development kit reboots after flashing.

Prerequisites for Jetson Orin Nano

This section describes the requirements to use e-CAM25_CUONX on the Jetson Orin Nano™ development kit.

The prerequisites are as follows:

- Host PC which runs Ubuntu 22.04 (64-bit) or Ubuntu 20.04 (64-bit).
- Download and install lbzip2 package.
- NVIDIA® provided L4T release and corresponding sample rootfs for Jetson Orin Nano™ development kit.
- The USB Type-C cable to plug into the recovery port of the Jetson Orin Nano™ development kit.
- Power cable (19V) to power the Jetson Orin Nano™ development kit.
- 64GB USB boot drive [SD Card/Pendrive] must be connected to the respective slot.

Please refer to the *e-CAM25_CUONX_Release_Package_Manifest_Rev_<ver>.pdf* to know the contents of release package and their description.

Setting Up Environment

The steps to setup the environment are as follows:

1. Run the following commands to setup the required environment variables.

```
mkdir -p top_dir
mkdir top_dir/tool_chain
export TOP_DIR=<absolute_path_to>/top_dir
export RELEASE_PACK_DIR=$TOP_DIR/e-
CAM25_CUONX_JETSON_ONX_ONANO
_<L4T_version>_<release_date>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export TOOL_CHAIN=$TOP_DIR/tool_chain
export CROSS_COMPILE=$TOOL_CHAIN/aarch64--glibc--
stable-2022.08-1/bin/aarch64-buildroot-linux-gnu-
export NVIDIA_SRC=$L4T_DIR/source
export TEGRA_KERNEL_OUT=$NVIDIA_SRC/out/nvidia-linux-
header
export KERNEL_HEADERS=$NVIDIA_SRC/kernel/kernel-jammy-
```

```
src
export INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
export SENSOR_DRIVER=e-CAM25_CUONX
```

2. Run the following command to copy the release package tar file to the staging directory.

```
mv <location_of>/e-CAM25_CUONX_JETSON_ONX_ONANO_
<L4T_version>_<release_date>_<release_version>.tar.gz
$TOP_DIR
```

Downloading Requirements

For building the kernel, a cross compiler toolchain and other tools necessary for compiling are required.

The steps to download the requirements for building the kernel are as follows:

1. Download the required toolchain from NVIDIA® website using link listed in below table.

Table 5: Required Toolchain

S.NO	Title	Version	Download link
1	GCC Toolchain	11.3.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v3.0/toolchain/aarch64--glibc--stable-2022.08-1.tar.bz2

a. Run the following commands to extract the package in host PC.

```
mv $HOME/Downloads/aarch64--glibc--stable-2022.08-
1.tar.bz2 $TOOL_CHAIN
cd $TOOL_CHAIN
tar -xf aarch64--glibc--stable-2022.08-1.tar.bz2
```

b. Run the following command to add CROSS_COMPILER to PATH environment variable for building kernel source.

```
PATH=$TOOL_CHAIN/aarch64--glibc--stable-2022.08-
1/bin:$PATH
```

2. Run the following commands to download the required package for extracting sources.

```
sudo apt-get update
sudo apt-get install qemu-user-static
sudo apt-get install build-essential
sudo apt-get install bc
sudo apt-get install lbzip2
sudo apt-get install python
sudo apt-get install flex
sudo apt-get install openssl
sudo apt-get install libssl-dev
```

3. Download the required L4T release package and sample rootfs from NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

   The steps to download and copy the package to staging directory are as follows:

   a. Download the packages from the NVIDIA® website as listed in below table.

Table 6: Packages for Jetson Orin Nano

S.NO	Title	Version	Download Link
1	L4T Jetson Orin NX™, Jetson Orin Nano™	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/release/Jetson_Linux_R36.4.0_aarch64.tbz2
2	L4T Jetson Orin NX™ Sample Rootfs	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/release/Tegra_Linux_Sample-Root-Filesystem_R36.4.0_aarch64.tbz2

   b. Run the following commands to copy the downloaded package to staging directory.

```
cp $HOME/Downloads/Jetson_Linux_R36.4.0_aarch64.tbz2
$TOP_DIR
cp $HOME/Downloads/Tegra_Linux_Sample-Root-
Filesystem_R36.4.0_aarch64.tbz2 $TOP_DIR
```

Extracting and Preparing L4T

The steps for extracting and preparing L4T must be performed in host PC are as follows:

1. Run the following commands to extract the downloaded L4T release package to navigate a folder with the name Linux_for_Tegra.

```
cd $TOP_DIR
tar -xf Jetson_Linux_R36.4.0_aarch64.tbz2
```

2. Run the following commands to extract the sample rootfs to the rootfs directory which is present inside the Linux_for_Tegra directory.

```
sudo tar xpf Tegra_Linux_Sample-Root-
Filesystem_R36.4.0_aarch64.tbz2 -C
Linux_for_Tegra/rootfs/
```

3. Run the following commands to set the package to be ready to flash binaries.

```
cd $L4T_DIR
sudo ./tools/l4t_flash_prerequisites.sh
sudo ./apply_binaries.sh
```

Extracting Release Package

Run the following commands to extract the e-CAM25_CUONX release package.

```
cd $TOP_DIR
tar -xaf e-CAM25_CUONX_JETSON_ONX_ONANO_
<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to the *e-CAM25_CUONX_Release_Package_Manifest_Rev_<ver>.pdf*.

Installation Procedure for Jetson Orin Nano

This section describes the steps for building and installing the kernel.

Building from Source

You can use the patch files provided by e-con Systems to use your own kernel image binary and modules along with the e-CAM25_CUONX camera on the Jetson Orin Nano™ development kit to build the kernel.

Downloading and Configuring Kernel

This section describes how you can download and configure the kernel for Jetson Orin Nano™ development kit.

Download the kernel source code for L4T from the NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

The steps to download and configure the kernel for Jetson Orin Nano™ development kit are as follows:

1. Download the packages from the NVIDIA® website as listed in below table.

Table 7: Packages for Jetson Orin Nano

S.NO	Title	Version	Download Link
1	L4T Jetson Orin NX™ + Jetson Orin Nano™ Sources	36.4.0	https://developer.nvidia.com/downloads/embedded/l4t/r36_release_v4.0/sources/public_sources.tbz2

2. Run the following command to copy the downloaded file to staging directory.

```
cp $HOME/Downloads/public_sources.tbz2 $TOP_DIR
```

3. Run the following commands to extract the downloaded kernel source code to any path on the host Linux PC.

```
tar xf public_sources.tbz2
cd $NVIDIA_SRC
```

4. Run the following commands to extract the kernel source code.

```
tar xf kernel_src.tbz2
tar xf kernel_oot_modules_src.tbz2
tar xf nvidia_kernel_display_driver_source.tbz2
```

5. Run the following command to make sure that the patch command is applied properly in the nvidia-oot source.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_oot.patch -
-dry-run
```

6. Run the following command to apply the patch file to the nvidia-oot source code if there is no error from dry-run command.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_oot.patch
```

7. Run the following command to make sure that the patch command is applied properly in the device tree source.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_dtb.patch -
-dry-run
```

8. Run the following command to apply the patch file to the device tree source code if there is no error from dry-run command.

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_dtb.patch
```

9. Run the following command to make sure that the patch command is applied properly in the module to build the sensor module.

```
mkdir -p sensor_driver

patch -d sensor_driver -p1 -i
$RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_module.patc
h --dry-run
```

10. Run the following command to apply the sensor module patch file to the module source code if there is no error from dry-run command.

```
patch -d sensor_driver -p1 -i
$RELEASE_PACK_DIR/Kernel/e-
CAM25_CUONX_JETSON_ONX_ONANO_<L4T_version>_module.patc
h
```

Building and Installing Kernel

The steps to build and install the kernel are as follows:

1. Run the following commands to build and install the kernel image and modules to the Jetson Orin Nano™ development kit.

```
make -C kernel

make modules

sudo -E make modules_install

make dtbs
```

2. Run the following commands to copy the dtbo file to Linux_for_Tegra (L4T_DIR) flashing path.

For 2 lane configuration,

```
sudo cp $NVIDIA_SRC/kernel-devicetree/generic-
dts/dtbs/tegra234-p3767-0000-p3768-0000-a0-2lane-
ar0234.dtbo $LDK_ROOTFS_DIR/boot
```

For 4 lane configuration,

```
sudo cp $NVIDIA_SRC/kernel-devicetree/generic-
dts/dtbs/tegra234-p3767-0000-p3768-0000-a0-4lane-
ar0234.dtbo $LDK_ROOTFS_DIR/boot
```

Note: Even if the image is custom built, the kernel configuration must have module versioning support for the camera driver to work.

Flashing Orin Nano Development Kit

The steps to flash the Jetson Orin Nano™ development kit are as follows:

1. Power OFF the kit.
2. Connect a jumper across FC-REC and GND to enter recovery mode.
3. Connect the Type-C USB cable to the Jetson Orin Nano™ development kit™ and host PC.

   The top view of Jetson Orin Nano™ development kit is shown below.

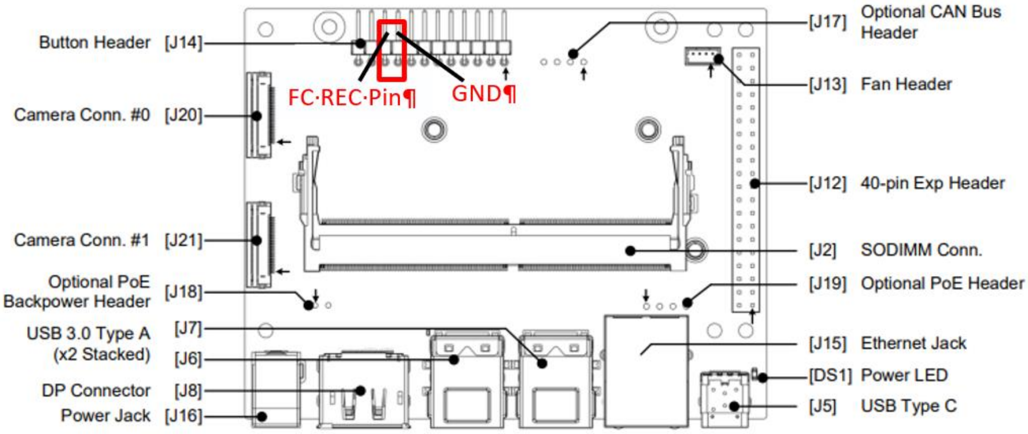


Figure 2: Top View of Jetson Nano Development Kit

4. Power ON the Jetson Orin Nano™ development kit.

   If the Jetson Orin Nano™ development kit is successfully changed to recovery mode, the Jetson Orin Nano™ development kit will be enumerated as an USB device to the host PC.

5. Run the following command to verify whether the Jetson Orin Nano™ development kit is in recovery mode.

```
lsusb
```

The output message appears as shown below.

```
Bus 003 Device 023: ID 0955:7523 NVidia Corp.
```

6. Run the following commands to flash the Jetson Nano™ SOM from Nano™ development kit from your host PC.

   ```
   cd $L4T_DIR
   1.Flash to NVMe
   sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
   external-device nvme0n1p1 -c
   tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
   bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
   --network usb0 jetson-orin-nano-devkit internal
   2. Flash to the USB
   sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
   external-device sda1 -c
   tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
   bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
   --network usb0 jetson-orin-nano-devkit internal
   3. Flash to SD card
   sudo ./tools/kernel_flash/l4t_initrd_flash.sh --
   external-device mmcblk0p1 -c
   tools/kernel_flash/flash_l4t_t234_nvme.xml -p "-c
   bootloader/generic/cfg/flash_t234_qspi.xml" --showlogs
   --network usb0 jetson-orin-nano-devkit internal
   ```

   Note: If you are flashing with a USB flash drive as storage, the USB drive must be at least 128GB.

7. **Note**: Now, the entire USB drive storage/NVMe on the Jetson Orin Nano™ development kit will be erased. Flashing will take about 10-30 minutes to complete depending on the host PC configuration.
8. Connect the Jetson Orin Nano™ development kit to a monitor and keyboard.
9. Reboot the Jetson Orin Nano™ development kit to complete the OS configuration, once the Jetson Orin Nano™ development kit reboots after flashing.

Loading Drivers

This section describes how to load the drivers, install the sample application, and use the sample application with e-CAM25_CUONX.

The module drivers for e-CAM25_CUONX will be loaded automatically in the Jetson Orin™ development kit during booting.

The steps to load the drivers are as follows:

1. Run the following command to select e-CAM25_CUONX device tree overlay file for 2 lane.

```
sudo /opt/nvidia/jetson-io/config-by-hardware.py -n
2="AR0234 Sensor 2lane"
```

the output message appears as shown below,

```
Modified /boot/extlinux/extlinux.conf to add following
DTBO entries:
/boot/tegra234-p3767-0000-p3768-0000-a0-2lane-
ar0234.dtbo
Reboot system to reconfigure.
```

2. Run the following command to select e-CAM25_CUONX device tree overlay file for 4 lane.

```
sudo /opt/nvidia/jetson-io/config-by-hardware.py -n
2="AR0234 Sensor 4lane"
```

the output message appears as shown below,

```
Modified /boot/extlinux/extlinux.conf to add following
DTBO entries:
/boot/tegra234-p3767-0000-p3768-0000-a0-4lane-
ar0234.dtbo
Reboot system to reconfigure.
```

2. Run the following command to check whether the camera is initialized.

```
sudo dmesg | grep -i "ar0234"
```

The output message appears as shown below for Jetson Orin NX™/Orin Nano™ development kit.

```
subdev ar0234 10-0042 bound
```

The output message indicates that the camera is initialized properly for Jetson Orin NX™/Orin Nano™ development kit.

3. Run the following command to check the presence of video node.

```
ls /dev/video*
```

The output message appears as shown below.

```
/dev/video0
```

where (*) denotes the number of cameras connected to the Jetson™ development kit.

Note: Only one camera node will be created for Jetson Orin NX™/Orin Nano™ development kit since only the CAM1 port on the Jetson Orin Nano™ development kit is supported for the 4-lane configuration.

This video node can be utilized by any V4L2 application for viewing the camera preview.

The login credentials of the Jetson Orin™ development kit are fully configurable on the first boot, and the sample login credentials are listed in below table.

Table 8: Sample Login Credentials

Fields	Inputs
Username	nvidia
Password	nvidia

Installing Sample Application

The e-CAM25_CUONX guvcviewer or ecam_tk1_guvcview is a simple GTK+ interface for capturing and viewing video from the devices supported on the Jetson Orin™ development kit.

Using guvcviewer or ecam_tk1_guvcview application, you can perform the following:

- Enumerate and list all the video devices connected.
- Display properties of video renderer.
- Change resolution and color space or compression for video stream if different resolutions are supported by the device.
- Display currently configured values of preview.
- Capture the still images and set the path where still images will be saved.
- Display the average frame rate.

All the above listed properties can be configured by attractive and easy to use Graphical User Interface (GUI).

Please refer to the *e-CAM_TK1-GUVCView_Build_and_Install_Guide_Rev_<ver>.pdf* for the procedure to build and install ecam_tk1_guvcview application on the Jetson Orin™ development kit.

Using Sample Applications with e-CAM25_CUONX

To use the ecam_tk1_guvcview application with e-CAM25_CUONX, please refer to the *e-CAM25_CUONX_Linux_App_User_Manual_Rev_<ver>.pdf* for the procedure to use ecam_tk1_guvcview respectively.

Troubleshooting

In this section, you can view the commonly occurring issues and their troubleshooting steps.

Flashing the Jetson development kit fails with the error, ERROR: might be timeout in USB write. How can this issue be solved?

To solve this issue, please follow these steps:

- First, try changing the USB port
- Move to a different USB port, if available.
- Power cycle the development kit and retry flashing.

If that doesn’t work, try disabling auto suspend:

- To disable autosuspend on your host's USB ports, run the following

command.

```
sudo bash -c 'echo -1 >
/sys/module/usbcore/parameters/autosuspend'
```

Power cycle the development kit and retry flashing.

Flashing Jetson Orin™ development kit fails with the error, "python: No such file or directory".

To solve this issue, please follow these steps:

1. Run the following commands to install python 2.7.

```
sudo apt-get update
sudo apt-get install python
```

2. Retry flashing the Jetson Orin™ development kit.

FAQ

1. **I have flashed L4T 36.4.0 in Jetson Orin™ development kit already. What are the steps to install the binaries?**

   Refer to the *e-CAM25_CUONX_Getting_Started_Manual_Rev_<ver>.pdf* to upgrade the latest binaries.

2. **How can I get the updated package?**

   Please login to the Developer Resources website and download the latest release package.

What's Next?

After understanding how to setup the Jetson Orin NX™/Orin Nano™ development kit using e-CAM25_CUONX, you can refer to the following documents to understand more about e-CAM25_CUONX.

- *e-CAM25_CUONX Release Notes*
- *e-CAM25_CUONX Release Package Manifest*
- *e-CAM_TK1-GUVCView Build and Install Guide*
- *e-CAM25_CUONX Linux App User Manual*

Glossary

AI: Artificial Intelligence.

API: Application Programming Interface.

CMOS: Complementary Metal Oxide Semiconductor.

DTB: Device Tree Blob.

IoT: Internet of Things.

Micro SD: Micro-Secure Digital.

MIPI: Mobile Industry Processor Interface.

NVMe: Nonvolatile Memory Express.

GIMP: GNU Image Manipulation Program.

GNU: GNU's Not Unix.

GTK: GIMP Toolkit.

GUI: Graphical User Interface.

L4T: Linux for Tegra.

MIPI: Mobile Industry Processor Interface.

Rootfs: Root Filesystems.

USB: Universal Serial Bus.

V4L2: Video for Linux version 2 is a collection of device drivers and API for supporting real-time video capture on Linux systems.

Support

Contact Us

If you need any support on e-CAM25_CUONX product, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

Revision History

Rev	Date	Description	Author
1.0	18-Apr-2023	Initial draft	Camera Dev Team
1.1	10-Nov-2023	Updated to L4T35.4.1 and supported 2 lane configuration	Camera Dev Team
1.2	06-Dec-2023	Flash command for SD card updated for Jetson Orin Nano Development kit	Camera Dev Team
1.3	24-May-2024	Updated to L4T36.3.0	Camera Dev Team
1.4	26-Jul-2024	Updated to fix vi-capture corruption issue	Camera Dev Team
1.5	23-Jan-2025	Updated to L4T36.4.0	Camera Dev Team
1.6	24-Mar-2025	Updated troubleshooting section	Camera Dev Team