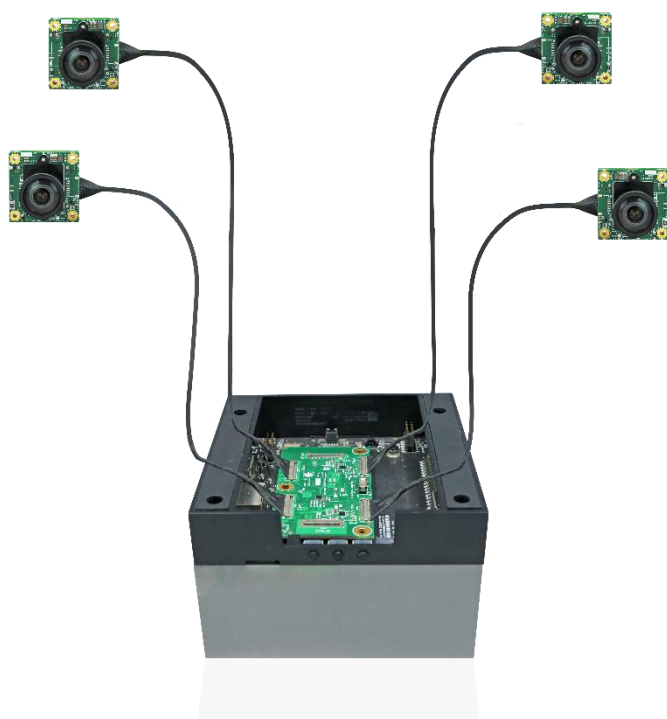


e-CAM56\_CUOAGX

# Developer Guide



### **Disclaimer**

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

# Contents

<b>INTRODUCTION TO E-CAM56_CUOAGX</b>	<b>4</b>
<b>SOFTWARE REQUIREMENTS</b>	<b>4</b>
<b>PREREQUISITES</b>	<b>5</b>
<b>SETTING UP ENVIRONMENT</b>	<b>5</b>
<b>DOWNLOADING REQUIREMENTS</b>	<b>6</b>
<b>EXTRACTING AND PREPARING L4T</b>	<b>7</b>
<b>EXTRACTING RELEASE PACKAGE</b>	<b>8</b>
<b>INSTALLATION PROCEDURE</b>	<b>9</b>
<b>BUILDING FROM SOURCE</b>	<b>9</b>
<b>DOWNLOADING AND CONFIGURING KERNEL</b>	<b>9</b>
<b>BUILDING AND INSTALLING KERNEL</b>	<b>10</b>
<b>MODIFYING ROOTFS</b>	<b>12</b>
<b>FLASHING JETSON AGX ORIN DEVELOPMENT KIT</b>	<b>13</b>
<b>LOADING DRIVERS</b>	<b>16</b>
<b>INSTALLING SAMPLE APPLICATION</b>	<b>16</b>
<b>USING SAMPLE APPLICATIONS WITH E-CAM56_CUOAGX</b>	<b>17</b>
<b>TROUBLESHOOTING</b>	<b>18</b>
<b>FAQ</b>	<b>19</b>
<b>WHAT'S NEXT?</b>	<b>20</b>
<b>GLOSSARY</b>	<b>21</b>
<b>SUPPORT</b>	<b>22</b>

# Introduction to e-CAM56\_CUOAGX

---

e-CAM56\_CUOAGX is a 5 MP MIPI camera from e-con Systems, a company with over two decades of experience in designing, developing, and manufacturing OEM cameras. It supports up to four camera modules in 4-lane mode which are connected to the Jetson AGX Orin™ development kit. The prebuild driver for this camera along with the camera board is provided by e-con Systems.

The NVIDIA® Jetson AGX Orin™ development kit is a full-featured development platform for visual computing. It is ideal for applications requiring high computational performance in a low power envelope. The Jetson AGX Orin™ development kit is pre-flashed with a Linux environment, includes support for many common APIs, and is supported by NVIDIA® complete development toolchain.

e-con Systems also provides the sample applications 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 shown in the below table.

**Table 1: Notation of Colors**

Color	Notation
Blue	Commands running in Host PC
Red	Commands running in Jetson development kit
Green	Output message in Terminal

This document explains how to setup the Jetson AGX Orin™ development kit for using e-CAM56\_CUOAGX camera.

## Software Requirements

The software requirements are as follows:

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

# Prerequisites

---

This section describes the requirements to use e-CAM56\_CUOAGX on the Jetson AGX Orin™ development kit.

The prerequisites are as follows:

- Host PC which runs Ubuntu 18.04 or 20.04 (64-bit).
- NVIDIA® provided L4T release and corresponding sample rootfs for the Jetson AGX Orin™ development kit. Please refer to the *e-CAM56\_CUOAGX\_Release\_Notes\_Rev\_<ver>.pdf* for the compatible Linux Distribution version (L4T version).
- A USB cable to plug into the recovery port of the Jetson AGX Orin™ development kit.

**Note:** USB-C port located on the other side of the board from the power port is used for flashing the board.

Please refer to the *e-CAM56\_CUOAGX\_Release\_Package\_Manifest\_Rev\_<ver>.pdf* to know the directory contents of release package and their description.

## Setting Up Environment

The steps to set up the environments are as follows:

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

```
mkdir top_dir/kernel_out -p
mkdir top_dir/kernel_sources
export TOP_DIR=<absolute path to>/top_dir
export RELEASE_PACK_DIR=$TOP_DIR/e-
CAM56_CUOAGX_JETSON_XAVIER_ORIN_<L4T_version>_<releas
e_date>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export CROSS_COMPILE=aarch64-buildroot-linux-gnu-
export CROSS32CC=arm-linux-gnueabi-hf-gcc
export LOCALVERSION="-tegra"
export TEGRA_KERNEL_OUT=$TOP_DIR/kernel_out
export
NVIDIA_SRC=$TOP_DIR/kernel_sources/Linux_for_Tegra/so
urce/public
```

```
export DISPLAY_SOURCE_DIR=$NVIDIA_SRC/NVIDIA-kernel-  
module-source-TempVersion
```

**Note:** The above steps must be performed in a single terminal till flashing completed, such that exported environmental variables are preserved.

2. Run the following command to copy the e-con Systems release package tar file to the staging directory.

```
mv <location of>/e-  
CAM56_CUOAGX_JETSON_XAVIER_ORIN_<L4T_version>_<releas  
e_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. You can use the default cross compiler toolchain and other tools provided in Ubuntu repositories.

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.

The steps to download the required toolchain and to build the kernel source are as follows:

- a. Download the required toolchain from NVIDIA® website as listed in the below table.

**Table 2: GCC Tool Chain Package**

S.NO	Title	Version	Download link
1	Bootlin Toolchain	9.3	<a href="https://developer.nvidia.com/embedded/jets-on-linux/bootlin-toolchain-gcc-93">https://developer.nvidia.com/embedded/jets-on-linux/bootlin-toolchain-gcc-93</a>

- b. Run the following commands to extract the package in the host PC.

```
mkdir -p $HOME/toolchain  
cd $HOME/toolchain  
tar -xf $HOME/Downloads/aarch64--glibc--stable-  
final.tar.gz
```

- c. Run the following command to export PATH environment for building kernel source.

```
export PATH=<Tool_chain_extract_path>/bin:$PATH
```

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

```
sudo apt-get update
```

```

sudo apt-get install bison
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
sudo apt-get install make

```

3. Download the required L4T release package and sample root filesystem from NVIDIA® website using <https://developer.nvidia.com/embedded/downloads> link.
  - a. Download the packages from the NVIDIA® website as listed in the below table.

**Table 3: Packages for Jetson AGX Orin**

S.NO	Title	Version	Download Link
1	L4T Jetson AGX Orin™ driver Package	35.4.1	<a href="https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/release/jetson_linux_r35.4.1_aarch64.tbz2/">https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/release/jetson_linux_r35.4.1_aarch64.tbz2/</a>
2	L4T Jetson AGX Orin™ sample Rootfs	35.4.1	<a href="https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/release/tegra_linux_sample-root-filesystem_r35.4.1_aarch64.tbz2/">https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/release/tegra_linux_sample-root-filesystem_r35.4.1_aarch64.tbz2/</a>

- b. Run the following commands to copy the downloaded file to staging directory.

```

cp
$HOME/Downloads/Jetson_Linux_R35.4.1_aarch64.tbz2
$TOP_DIR

cp $HOME/Downloads/Tegra_Linux_Sample-Root-
Filesystem_R35.4.1_aarch64.tbz2 $TOP_DIR

```

## Extracting and Preparing L4T

The steps for extracting and preparing L4T are as follows:

**Note:** The following steps must be performed in the host PC.

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
sudo tar -xjpf Jetson_Linux_R35.4.1_aarch64.tbz2

```

**Note:** The folder contains the necessary tools and binaries for modifying the Jetson AGX Orin™ development kit.

2. Run the following commands to extract the sample file system to the rootfs directory which is present inside the Linux\_for\_Tegra directory.

```
cd $LDK_ROOTFS_DIR
sudo tar -xjpf $TOP_DIR/Tegra_Linux_Sample-Root-Filesystem_R35.4.1_aarch64.tbz2
```

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

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

## Extracting Release Package

Run the following commands to extract the e-CAM56\_CUOAGX release package.

```
cd $TOP_DIR
tar -xf e-CAM56_CUOAGX_JETSON_XAVIER_ORIN_<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to *e-CAM56\_CUOAGX-Release\_Package\_Manifest\_Rev\_<ver>.pdf*.

Please refer to Installation Procedure section to use prebuilt files or build kernel with support for e-CAM56\_CUOAGX. The procedure would require flashing the eMMC of the Jetson AGX Orin™ development kit for erasing the pre-existing contents.

Please refer to *e-CAM56\_CUOAGX\_Getting\_Started\_Manual\_Rev\_<ver>.pdf* to upgrade the Jetson AGX Orin™ development kit which is already running L4T version and enable support for e-CAM56\_CUOAGX without flashing the eMMC. The procedure will preserve the existing rootfs of Jetson AGX Orin™ development kit.



# Installation Procedure

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

## Building from Source

You can use the patch file provided by e-con Systems to build your own customized kernel image binary and modules with support to use e-CAM56\_CUOAGX camera on the Jetson AGX Orin™ development kit.

## Downloading and Configuring Kernel

This section describes how you can download and configure the kernel for Jetson AGX Orin™ 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 AGX Orin™ development kit are as follows:

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

**Table 4: Packages for Jetson AGX Orin**

S.NO	Title	Version	Download Link
1	L4T Jetson AGX Orin™ Sources	35.4.1	<a href="https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/sources/public_sources.tbz2/">https://developer.nvidia.com/downloads/embedded/l4t/r35_release_v4.1/sources/public_sources.tbz2/</a>

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

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

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

```
cd $TOP_DIR/kernel_sources
tar -xjpf public_sources.tbz2
cd $NVIDIA_SRC
```

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

```
tar -xjpf kernel_src.tbz2
tar -xjpf nvidia_kernel_display_driver_source.tbz2
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM56_CUOAGX_JETSON_XAVIER_ORIN_<L4T_version>_kernel.
patch --dry-run
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM56_CUOAGX_JETSON_XAVIER
_ORIN_<L4T_version>_kernel.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-
CAM56_CUOAGX_JETSON_XAVIER
_ORIN_<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-
CAM56_CUOAGX_JETSON_XAVIER
_ORIN_<L4T_version>_dtb.patch
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM56_CUOAGX_JETSON_XAVIER
_ORIN_<L4T_version>_module.patch --dry-run
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM56_CUOAGX_JETSON_XAVIER
_ORIN_<L4T_version>_module.patch
```

## Building and Installing Kernel

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

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

```
cd kernel/kernel-5.10/
make ARCH=arm64 O=$TEGRA_KERNEL_OUT tegra_defconfig
make ARCH=arm64 O=$TEGRA_KERNEL_OUT Image -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT dtbs
sudo ARCH=arm64 make O=$TEGRA_KERNEL_OUT
modules_install INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
```

2. Run the following commands to build and install the e-CAM56\_CUOAGX driver code.

```
cd $NVIDIA_SRC/e-CAM56_CUOAGX

make ARCH=arm64 KERNEL_PATH=$TEGRA_KERNEL_OUT e-
con_cam

sudo make -C $TEGRA_KERNEL_OUT M=$PWD
INSTALL_MOD_PATH=$LDK_ROOTFS_DIR modules_install

cd $DISPLAY_SOURCE_DIR

make -C $PWD modules -j $(nproc) TARGET_ARCH=aarch64
ARCH=arm64 CC="{CROSS_COMPILE}gcc"
LD="{CROSS_COMPILE}ld" AR="{CROSS_COMPILE}ar"
CXX="{CROSS_COMPILE}g++"
OBJCOPY="{CROSS_COMPILE}objcopy"
SYSOUT="$TEGRA_KERNEL_OUT"
SYSSRC="$NVIDIA_SRC/kernel/kernel-5.10"

sudo make M=$PWD/kernel-open -C $TEGRA_KERNEL_OUT
INSTALL_MOD_PATH=$LDK_ROOTFS_DIR modules_install
```

3. Run the following commands to copy the kernel to Linux\_for\_Tegra (L4T\_DIR) flashing path.

```
sudo cp $TEGRA_KERNEL_OUT/arch/arm64/boot/Image
$L4T_DIR/kernel/ -f
```

4. Run the following commands to copy the dtb file to Linux\_for\_Tegra (L4T\_DIR) flashing path.

Execute the below command for **four camera setup in Jetson AGX Orin™ 64GB Development Kit.**

```
sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/nvidia/tegra234
-p3701-0005-p3737-0000-camera-4lane-eimx568.dtb
$L4T_DIR/kernel/dtb/tegra234-p3701-0005-p3737-
0000.dtb -f
```

Execute the below command for **four camera setup in Jetson AGX Orin™ 32GB Development Kit.**

```
sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/nvidia/tegra234
-p3701-0000-p3737-0000-camera-4lane-eimx568.dtb
$L4T_DIR/kernel/dtb/tegra234-p3701-0000-p3737-
0000.dtb -f
```

5. Follow the steps in Modifying Rootfs and Flashing Jetson AGX Orin Development Kit sections to make the Jetson AGX Orin™ development kit to run in tegra kernel.

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

## Modifying Rootfs

Run the following commands to modify additional files in the rootfs for the proper functioning of the e-CAM56\_CUOAGX camera on the Jetson AGX Orin™ development kit.

### ISP Libraries for Jetson AGX Orin™

```
sudo cp $RELEASE_PACK_DIR/misc/camera_overrides_jetson-  
orin.isp  
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_overrid  
es.isp -f
```

Then,

```
sudo chmod 664  
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_overrid  
es.isp  
  
sudo chown root:root  
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_overrid  
es.isp
```

**Note:** To achieve the best image quality, the ISP has been tuned by e-con Systems in collaboration with NVIDIA®, specific to e-CAM56\_CUOAGX and the ISP configuration file is **camera\_overrides.isp** file.

### Nvscf Library files for Jetson Orin™

```
sudo cp $RELEASE_PACK_DIR/misc/libnvscf.so  
$LDK_ROOTFS_DIR/lib/aarch64-linux-gnu/tegra/libnvscf.so  
-f
```

Then,

```
sudo chmod 664 $LDK_ROOTFS_DIR/lib/aarch64-linux-  
gnu/tegra/libnvscf.so  
  
sudo chown root:root $LDK_ROOTFS_DIR/lib/aarch64-linux-  
gnu/tegra/libnvscf.so
```

```
sudo cp  
$RELEASE_PACK_DIR/misc/libnvargus_socketserver.so  
$LDK_ROOTFS_DIR/lib/aarch64-linux-  
gnu/tegra/libnvargus_socketserver.so -f
```

Then,

```
sudo chmod 664 $LDK_ROOTFS_DIR/lib/aarch64-linux-  
gnu/tegra/libnvargus_socketserver.so  
  
sudo chown root:root $LDK_ROOTFS_DIR/lib/aarch64-linux-  
gnu/tegra/libnvargus_socketserver.so
```

```
sudo cp
$RELEASE_PACK_DIR/misc/libnvargus_socketclient.so
$LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus_socketclient.so -f
```

**Then,**

```
sudo chmod 664 $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus_socketclient.so

sudo chown root:root $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus_socketclient.so
```

```
sudo cp $RELEASE_PACK_DIR/misc/libnvargus.so
$LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus.so -f
```

**Then,**

```
sudo chmod 664 $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus.so

sudo chown root:root $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvargus.so
```

```
sudo cp $RELEASE_PACK_DIR/misc/libnvmfusacap.so
$LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvmfusacap.so -f
```

**Then,**

```
sudo chmod 664 $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvmfusacap.so

sudo chown root:root $LDK_ROOTFS_DIR/lib/aarch64-linux-
gnu/tegra/libnvmfusacap.so
```

**Copying the deskew firmware file,**

```
sudo cp $RELEASE_PACK_DIR/misc/camera-rtcpu-t234-
rce_35_4_1.img $L4T_DIR/bootloader/camera-rtcpu-t234-
rce.img -f
```

**Run the following command to copy the MCU file to the rootfs,**

```
sudo cp $RELEASE_PACK_DIR/Firmware/imx568_cam_fw.bin
$LDK_ROOTFS_DIR/lib/firmware/imx568_cam_fw.bin -f
```

## Flashing Jetson AGX Orin Development Kit

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

1. Connect the USB Type-C cable to the host PC and the USB-C port of Jetson AGX Orin™ development kit.

The location of USB-C port on the Jetson AGX Orin™ development kit is shown in the below figure.



**Figure 1: Location of USB-C Port on Jetson AGX Orin Development Kit**

**Note:** USB-C port located on the other side of the board from the power port is used for flashing the board.

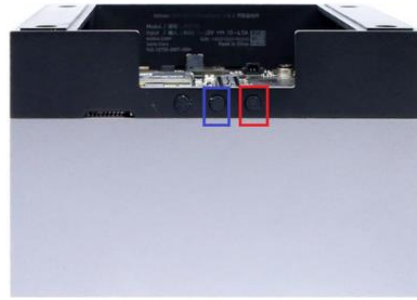
2. Connect the power adaptor to the Jetson AGX Orin™ development kit as shown in the below figure.



**Figure 2: Power Adaptor connected to Jetson Orin Development Kit**

3. Set the board to recovery mode, as mentioned in the below steps:
  - a. Press and hold the **Recovery** button of Jetson AGX Orin™ development kit.
  - b. Press the **Power** button of Jetson AGX Orin™ development kit.

The location of **Recovery** and **Power** buttons on the Jetson AGX Orin™ development kit is shown in the below figure.



 - Recovery Button

 - Power Button

Figure 3: Location of Recovery and Power Buttons on Jetson Orin Development Kit

- c. Release both **Recovery** and **Power** buttons.

If the Jetson AGX Orin™ development kit is successfully changed to recovery mode, **it is enumerated as USB device to the host pc, even not in recovery mode!**

4. Run the following command to verify whether the board is in recovery mode.

```
lsusb
```

The output message appears as shown below.

For Jetson AGX Orin™,

```
Bus 003 Device 006: ID 0955:7023 NVidia Corp.
```

5. Run the following flash.sh scripts to flash the Jetson AGX Orin™ development kit from your host PC.

```
cd $L4T_DIR  
sudo ./flash.sh jetson-agx-orin-devkit mmcblk0p1
```

**Note:** Now, the entire eMMC on the Jetson AGX Orin™ development kit and any files present on the device will be erased. It will take about 10-30 minutes to complete based on the host PC configuration.

6. Reboot the device.

# Loading Drivers

This section describes how to load the drivers, install the sample application and use the sample application with e-CAM56\_CUOAGX.

The module drivers for e-CAM56\_CUOAGX will be loaded automatically in the Jetson AGX Orin™ development kit during booting.

The steps to load the drivers are as follows:

1. Run the following command to check whether all the cameras connected are initialized.

```
sudo dmesg | grep -i "Detected eimx568 sensor"
```

The output message appears as shown below.

```
Detected eimx568 sensor
```

The output message shows depend upon number of cameras that are properly initialized.

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

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

The output message appears as shown below.

```
/dev/video*
```

Where (\*) denotes the number of cameras connected to the Jetson AGX Orin™ development kit.

The number of video node reflect the number of connected cameras. If no other cameras are connected to the Jetson AGX Orin™ development kit. These video nodes can be utilized by any V4L2 application for viewing the camera preview.

The default login credentials of the Jetson AGX Orin™ development kit is shown in the below table.

**Table 5: Default Login Credentials**

Fields	Inputs
Username	nvidia
Password	nvidia

## Installing Sample Application

e-con Systems provides a camera application, called eCAM\_argus\_camera (based on NVIDIA® sample camera application for Jetson AGX Orin™ development kit,



argus\_camera) along with the e-CAM56\_CUOAGX camera. e-con Systems has customized and retained features in NVIDIA® argus\_camera application for e-CAM56\_CUOAGX.

The eCAM\_argus\_camera is a video viewer and capture software for the camera driver on Jetson AGX Orin™ development kit, customized to demonstrate the features of e-CAM56\_CUOAGX.

Please refer to the e-  
*CAM56\_CUOAGX\_eCAM\_Argus\_Camera\_Installation\_Guide\_Rev\_<ver>.pdf* for the procedure to build and install the eCAM\_argus\_camera application.

## **Using Sample Applications with e-CAM56\_CUOAGX**

Please refer to the e-  
*CAM56\_CUOAGX\_eCAM\_Argus\_Camera\_App\_User\_Manual\_Rev\_<ver>.pdf* to use eCAM\_argus\_camera Application.

# Troubleshooting

---

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

1. In case if the sensor is not detected after flashing the kit using top\_dir, please make sure that correct dtb is loaded to the board using the below command.  
**sudo dmesg | grep "dts"**

The above command should display the dtb file which we copied before flashing as shown below.

```
nvidia@nvidia-desktop:~$ sudo dmesg | grep "dts"
[sudo] password for nvidia:
[  0.004718] DTS File Name: /home/varunkumar/top_dir_bharathi/kernel_sources/L
inux_for_Tegra/source/public/kernel/kernel-5.10/arch/arm64/boot/dts/../../../../
../../../../hardware/nvidia/platform/t23x/concord/kernel-dts/tegra234-p3701-0000-p3737
-0000-camera-2lane-eimx568.dts
```

# FAQ

1. Is it possible to install the camera binaries without flashing the entire package?

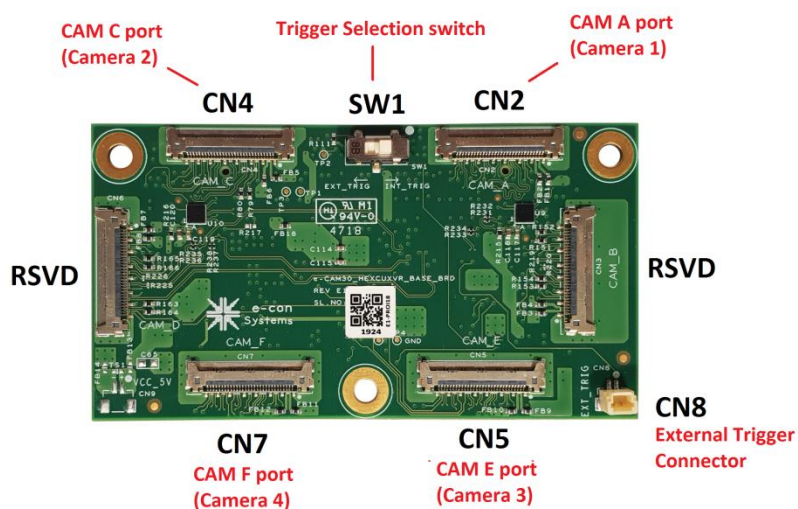
Yes, please refer *e-CAM56\_CUOAGX\_Getting\_Started\_Manual\_Rev\_<ver>.pdf* to upgrade the modules, kernel image and device tree.

2. Is the provided camera driver binary to or ko file compatible with all L4T version?

No, it is not compatible with all L4T version. Please refer to Downloading Requirements section to know about the compatible L4T version.

3. I bought either one or four cameras, can I connect cameras in any of available six connectors?

No, you must connect the camera(s) to CAM A, CAM C, CAM E and CAM F ports for Jetson AGX Orin™ development kit. To connect the cameras to the respective ports, you can refer the following figure.



# What's Next?

---

After understanding on how to setup the Jetson AGX Orin™ development kit for using e-CAM56\_CUOAGX MIPI camera, you can refer to the following documents to understand more about e-CAM56\_CUOAGX.

- *e-CAM56\_CUOAGX Release Notes*
- *e-CAM56\_CUOAGX Release Package Manifest*
- *e-CAM56\_CUOAGX eCAM\_Argus\_Camera Build and Installation Guide*
- *e-CAM56\_CUOAGX eCAM\_Argus\_Camera App User Manual*

# Glossary

---

**API:** Application Programming Interface.

**DTB:** Device Tree Blob.

**eMMC:** Embedded Multi-Media Controller.

**GUI:** Graphical User Interface.

**L4T:** Linux for Tegra.

**MIPI:** Mobile Industry Processor Interface.

**Rootfs:** Root Filesystem.

**USB:** Universal Serial Bus.

**V4L2:** Video for Linux version2 is a collection of device drivers and API for supporting real-time video capture on Linux systems.

## **Contact Us**

If you need any support on e-CAM56\_CUOAGX 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	22-Aug-2023	Initial Draft	Camera Dev Team
1.1	19-Oct-2023	Updated to L4T35.4.1 and 2-lane configuration to support six cameras	Camera Dev Team
1.2	31-Oct-2023	Updated the dtb files for 32GB and 64GB and troubleshooting	Camera Dev Team
1.3	30-Jan-2024	Added Nvidia libraries and service files to rootfs to fix some issue in L4T35.4.1	Camera Dev Team
1.4	08-Mar-2024	Removed the 2-lane support	Camera Dev Team
1.5	21-Mar-2024	Updated the Product image in home page	Camera Dev Team