

e-CAM20\_CUXVR

# Developer Guide



Version 1.8

e-con Systems

3/25/2020



**e-con Systems**

Your Product Development Partner

### **Disclaimer**

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

# Contents

<b>INTRODUCTION TO E-CAM20_CUXVR</b>	<b>3</b>
<b>SOFTWARE REQUIREMENTS</b>	<b>3</b>
<b>PREREQUISITES</b>	<b>4</b>
<b>SETTING UP THE ENVIRONMENT</b>	<b>4</b>
<b>DOWNLOADING THE REQUIREMENTS</b>	<b>5</b>
<b>EXTRACTING AND PREPARING L4T</b>	<b>6</b>
<b>EXTRACTING THE RELEASE PACKAGE</b>	<b>6</b>
<b>INSTALLATION PROCEDURE</b>	<b>7</b>
<b>BUILDING FROM SOURCE</b>	<b>7</b>
DOWNLOADING AND CONFIGURING THE KERNEL	7
BUILDING AND INSTALLING THE KERNEL	8
MODIFYING THE ROOTFS	9
FLASHING THE JETSON AGX XAVIER/ JETSON TX2 DEVELOPMENT KIT	10
<b>LOADING THE DRIVERS</b>	<b>13</b>
<b>INSTALLING THE SAMPLE APPLICATION</b>	<b>14</b>
<b>USING THE SAMPLE APPLICATIONS WITH E-CAM20_CUXVR</b>	<b>14</b>
<b>TROUBLESHOOTING</b>	<b>15</b>
<b>FAQ</b>	<b>16</b>
<b>WHAT'S NEXT?</b>	<b>17</b>
<b>GLOSSARY</b>	<b>18</b>
<b>SUPPORT</b>	<b>19</b>

# Introduction to e-CAM20\_CUXVR

e-con Systems is a leading Embedded Product Design Services Company which specializes in advanced camera solutions. e-CAM20\_CUXVR is a new MIPI camera board which helps to connect one or four IMX290 camera modules in 4-lane mode (in Jetson Xavier™) and 2-lane mode (in Jetson TX2™) to the Jetson AGX Xavier™/TX2™ development kit launched by e-con Systems. The prebuild driver for this camera along with the camera board is provided by e-con Systems.

The NVIDIA® Jetson AGX Xavier™/TX2™ 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 Xavier™/TX2™ development kit is pre-flashed with a Linux environment, includes support for many common APIs, and is supported by NVIDIA® complete development toolchain.

e-CAM20\_CUXVR has a variant of either one or four 2 MP color cameras each with S-mount (also known as M12 board lens) lens holder. The S-mount is one of the most commonly used small form-factor lens mounts for board cameras. Each e-CAM20\_CUXVR camera contains 1/2.8" IMX290 CMOS image sensor from Sony® and is interfaced to the J509 connector on the Jetson AGX Xavier™ or interfaced to the J22 connector on the Jetson TX2™; using the e-CAM30\_HEXCUXVR\_BASE\_BRD board.

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 below table.

**Table 1: Notation of Colors**

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

This document explains how to setup the Jetson AGX Xavier™ development kit for using e-CAM20\_CUXVR 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-CAM20\_CUXVR on the Jetson AGX Xavier™/TX2™ development kit.

The prerequisites are as follows:

- Host PC which runs Ubuntu 16.04 (64-bit).
- NVIDIA® provided L4T release and corresponding sample rootfs for Jetson AGX Xavier™/TX2™. Please refer to the *e-CAM20\_CUXVR\_Release\_Notes.pdf* for the compatible Linux Distribution version (L4T version).
- A USB cable to plug into the recovery port of the Jetson AGX Xavier™/TX2™ 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-CAM20\_CUXVR\_Release\_Package\_Manifest.pdf* to know the contents of release package and their description.

## Setting Up the 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
export TOP_DIR=<absolute path to>/top_dir
export RELEASE_PACK_DIR=$TOP_DIR/e-
CAM20_CUXVR_JETSON_XAVIER_TX2_<L4T_version>_<release_d
ate>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export CROSS_COMPILE=aarch64-linux-gnu-
export CROSS32CC=arm-linux-gnueabi-gcc
export TEGRA_KERNEL_OUT=$TOP_DIR/kernel_out
export KERNEL_PATH=$TOP_DIR/kernel_out
```

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

```
mv <location of>/e-
CAM20_CUXVR_JETSON_XAVIER_TX2_<L4T_version>_<release_d
ate>_<release_version>.tar.bz2 $TOP_DIR
```

**Note:** The above steps must be performed in a single terminal such that the environment variables are preserved.

## Downloading the 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:

3. Download the required toolchain from NVIDIA® website using <https://developer.nvidia.com/embedded/downloads> link.
  - a. Download the required toolchain from NVIDIA® website as listed in below table.

**Table 2: GCC Tool Chain Package**

S.NO	Title	Version
1	GCC Tool Chain Kernel	6.4.1

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

```
tar -xf ~/Downloads/gcc-linaro-6.4.1-2017.08-x86_64_aarch64-linux-gnu.tar.xz
```

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

```
export PATH=<Tool_chain_extract_path>/gcc-linaro-6.4.1-2017.08-x86_64_aarch64-linux-gnu/bin:${PATH}
```

4. 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 below table.

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

S.NO	Title	Version
1	L4T Jetson Driver Package	32.2.1
2	L4T Sample Root Filesystem	32.2.1

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

```
cp $HOME/Downloads/Jetson_Linux_R32.2.1_aarch64.tbz2 $TOP_DIR
cp $HOME/Downloads/Tegra_Linux_Sample-Root-Filesystem_R32.2.1_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 xpf Jetson_Linux_R32.2.1_aarch64.tbz2
```

**Note:** The folder contains the necessary tools and binaries for modifying the Jetson AGX Xavier™/TX2™ 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 xpf $TOP_DIR/Tegra_Linux_Sample-Root-Filesystem_R32.2.1_aarch64.tbz2
```

**Note:** Ensure that lbzip2 is installed in your host PC. If not install lbzip2 using the following command before running apply\_binaries.sh.

```
sudo apt-get install lbzip2
```

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

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

## Extracting the Release Package

Run the following commands to extract the e-CAM20\_CUXVR release package.

```
cd $TOP_DIR
tar -xf e-
CAM20_CUXVR_JETSON_XAVIER_TX2_<L4T_version>_<release_date>
_<release_version>.tar.bz2
```

To know more about the release package, please refer to *e-CAM20\_CUXVR-Release\_Package\_Manifest.pdf*.

Please refer to *Installation Procedure* section to use prebuilt files or build kernel with support for e-CAM20\_CUXVR. The procedure would require flashing the eMMC of the Jetson AGX Xavier™/TX2™ board for erasing the pre-existing contents.

Please refer to *e-CAM20\_CUXVR\_Getting\_Started\_Manual.pdf* to upgrade the Jetson board which is already running L4T version and enable support for e-CAM20\_CUXVR without flashing the eMMC. The procedure will preserve the existing rootfs of Jetson AGX Xavier™/TX2™.

# Installation Procedure

This section describes the steps for building and installing the kernel. You can choose to use own customized kernel.

## Building from Source

You can use the patch file provided by e-con Systems to build your own kernel image binary and modules with support to use e-CAM20\_CUXVR camera on the Jetson AGX Xavier™/TX2™ development kit.

## Downloading and Configuring the Kernel

This section describes how you can download and configure the kernel for Jetson AGX Xavier™/TX2™.

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 Xavier™/TX2™ development kit are as follows:

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

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

Title	Version
L4T Sources	32.2.1

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.

```
cd $TOP_DIR
tar -xjf public_sources.tbz2
cd $TOP_DIR/public_sources
```

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

```
tar -xjf kernel_src.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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_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-
CAM20_CUXVR_JETSON_XAVIER_TX2_L4T32.2.1_module.patch
```

## Building and Installing the Kernel

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

11. Run the following commands to build and install the kernel image and modules to the Jetson AGX Xavier™/TX2™ development kit.

```
cd kernel/kernel-4.9/
make O=$TEGRA_KERNEL_OUT tegra_ecam_isp_defconfig
make O=$TEGRA_KERNEL_OUT Image -j4
make O=$TEGRA_KERNEL_OUT modules -j4
make O=$TEGRA_KERNEL_OUT dtbs
sudo ARCH=arm64 make O=$TEGRA_KERNEL_OUT
modules_install INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
cd ../../e-CAM20_CUXVR
make
sudo mkdir -p
$LDK_ROOTFS_DIR/lib/modules/4.9.140/extra
sudo cp ./imx290.ko
$LDK_ROOTFS_DIR/lib/modules/4.9.140/extra -f
```

```
cd ../kernel/kernel-4.9/

sudo depmod -b $LDK_ROOTFS_DIR 4.9.140

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

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra194-p2888-
0001-p2822-0000-camera-4lane-eimx290.dtb
$L4T_DIR/kernel/dtb/tegra194-p2888-0001-p2822-0000.dtb
-f
```

#### For Jetson TX2™

```
sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra186-quill-
p3310-1000-c03-00-camera-2lane-eimx290.dtb
$L4T_DIR/kernel/dtb/tegra186-quill-p3310-0001-c03-00-
base.dtb -f
```

12. Follow the steps in *Modifying the Rootfs* and *Flashing the Jetson AGX Xavier/ Jetson TX2 Development Kit* sections to make the Jetson AGX Xavier™/TX2™ development kit to run in a custom kernel.

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

### Modifying the Rootfs

Run the following commands to modify additional files in the rootfs for the proper functioning of the e-CAM20\_CUXVR camera on the Jetson AGX Xavier™/TX2™ development kit.

#### ISP Libraries for Jetson Xavier™

```
sudo cp
$RELEASE_PACK_DIR/misc/camera_overrides_eimx290_jetson-
xavier.isp
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_override
s.isp -f
```

#### ISP Libraries for Jetson TX2™

```
sudo cp
$RELEASE_PACK_DIR/misc/camera_overrides_eimx290_jetson-
tx2.isp
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_override
s.isp -f
```

Then,

```
sudo chmod 664
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_override
s.isp
```

```
sudo chown root:root  
$LDK_ROOTFS_DIR/var/nvidia/nvcam/settings/camera_override  
s.isp
```

**Note:** To achieve the best image quality, the ISP has been tuned by e-con Systems in collaboration with NVIDIA®, specific to e-CAM20\_CUXVR and the ISP configuration file is **camera\_overrides.isp** file.

#### Other Files

```
sudo cp $RELEASE_PACK_DIR/misc/max-isp-vi-clks.sh  
$LDK_ROOTFS_DIR/home/max-isp-vi-clks.sh -f  
sudo chmod +x $LDK_ROOTFS_DIR/home/max-isp-vi-clks.sh
```

### Flashing the Jetson AGX Xavier/ Jetson TX2 Development Kit

The steps to flash the Jetson AGX Xavier™/TX2™ development kit are as follows:

1. Connect the USB Type-C cable to the host PC and the USB-C port of Jetson AGX Xavier™ or USB Type-B port of Jetson TX2™ development kit.

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



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

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

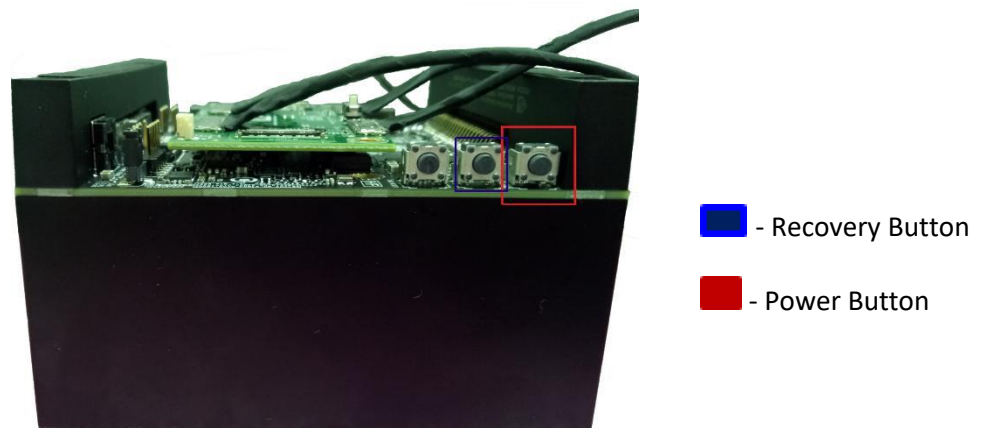
The location of USB-B port on the Jetson TX2™ development kit is shown in below figure.



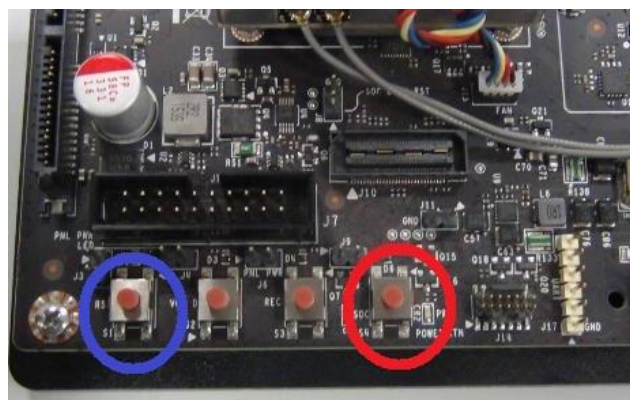
**Figure 2: Location of USB-C Port on Jetson TX2 Development Kit**

2. Set the board to recovery mode, as mentioned in below steps:
  - a. Press and hold the **Recovery** button of Jetson AGX Xavier™/ TX2™ development kit.
  - b. Press the **Power** button of Jetson AGX Xavier™/ TX2™ development kit.

The location of **Recovery** and **Power** buttons on the Jetson AGX Xavier™/ TX2™ development kit is shown in below figure.



**Figure 3: Location of Recovery and Power Buttons on Jetson AGX Xavier Development Kit**



**Figure 4: Location of Recovery and Power Buttons on Jetson TX2 Development Kit**

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

If the board is successfully changed to recovery mode, the Jetson AGX Xavier™/ TX2™ development kit will be enumerated as an USB device to the host PC.

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

```
lsusb
```

The output message appears as shown below.

For Jetson AGX Xavier™

```
Bus 001 Device 102: ID 0955:7020 NVidia Corp.
```

For Jetson TX2™

```
Bus 003 Device 006: ID 0955:7c18 NVidia Corp.
```

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

```
cd $L4T_DIR  
sudo ./flash.sh jetson-xavier mmcblk0p1
```

**for Jetson TX2**

```
sudo ./flash.sh jetson-tx2 mmcblk0p1
```

**To update the dtb alone, use**

```
sudo ./flash.sh -k kernel-dtb -d  
<path_to_dtb>/<dtb_name>.dtb jetson-xavier mmcblk0p1
```

Replace 'jetson-xavier' with 'jetson-tx2' for flashing dtb on TX2 development kit.

**Note:** Now, the entire eMMC on the Jetson AGX Xavier™/TX2™ 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.

4. Reboot the device.

# Loading the Drivers

This section describes how to load the drives, install the sample application and use the sample application with e-CAM20\_CUXVR.

The module drivers for e-CAM20\_CUXVR will be loaded automatically in the Jetson AGX Xavier™/TX2™ 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.

```
dmesg | grep "Detected IMX290 sensor"
```

The output message appears as shown below.

```
Detected imx290 sensor
Detected imx290 sensor
Detected imx290 sensor
Detected imx290 sensor
```

The output message indicates that all cameras connected are initialized properly.

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

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

The output message for quad camera setup appears as shown below.

```
/dev/video0
/dev/video1
/dev/video2
/dev/video3
```

The number of video node reflect the number of connected cameras. If no other cameras are connected to the Jetson AGX Xavier™/TX2™ 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 Xavier™/TX2™ development kit is shown in below table.

**Table 5: Default Login Credentials**

Fields	Inputs
Username	nvidia
Password	nvidia

## Installing the Sample Application

e-con Systems shares a camera application, called eCAM\_argus\_camera (based on NVIDIA® sample camera application for Jetson AGX Xavier™/TX2™ boards, argus\_camera) along with the e-CAM20\_CUXVR camera. e-con Systems has customized and retained only features applicable to e-CAM20\_CUXVR from NVIDIA® argus\_camera application, in eCAM\_argus\_camera.

The eCAM\_argus\_camera is a video viewer and capture software for the camera driver on Jetson AGX Xavier™/TX2™, customized to demonstrate the features of e-CAM20\_CUXVR.

Please refer to the *e-CAM20\_CUXVR-eCAM\_ArgusCamera\_Installation\_Guide.pdf* for the procedure to build and install the eCAM\_argus\_camera application.

To stream all connected cameras (maximum 4 cameras) in a single window with synchronization, e-con Systems provides a fully customized libargus application named eCAM\_Argus\_MultiCamera. Please refer to the *e-CAM20\_CUXVR-eCAM\_Argus\_MultiCamera\_App\_User\_Manual.pdf* for the procedure to build and use the eCAM\_Argus\_MultiCamera application.

## Using the Sample Applications with e-CAM20\_CUXVR

To use the sample applications with e-CAM20\_CUXVR, please refer to the *e-CAM20\_CUXVR\_Linux\_App\_User\_Manual.pdf* and *e-CAM20\_CUXVR-eCAM\_Argus\_MultiCamera\_App\_User\_Manual.pdf* for the procedure to use eCAM\_argus\_camera and eCAM\_Argus\_MultiCamera applications respectively.

# Troubleshooting

---

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

**Unloading the camera drivers causes a kernel crash.** This is a known issue.

This is a known issue. Please write to [techsupport@e-consystems.com](mailto:techsupport@e-consystems.com) to get immediate support on other issues.



# FAQ

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

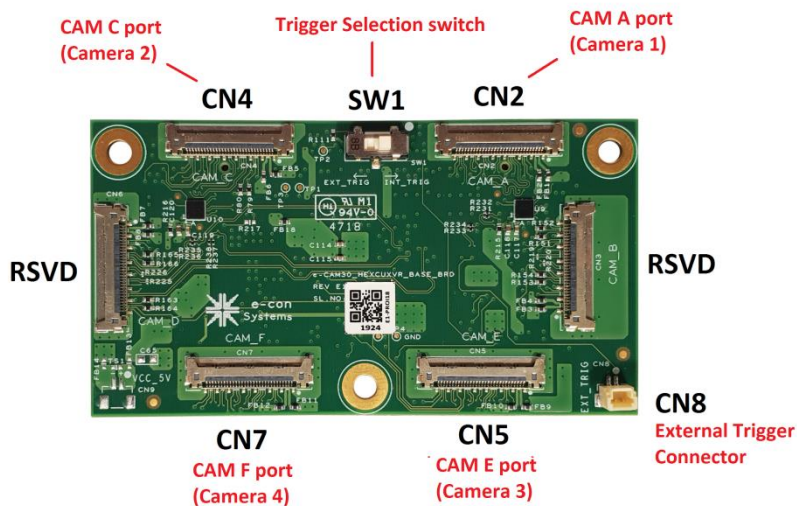
Yes, please refer [e-CAM20\\_CUXVR\\_Getting\\_Started\\_Manual.pdf](#) to upgrade the modules, kernel image and device tree.

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

No, it is not compatible with all L4T version. Please refer to *Downloading the 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 Xavier. 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 Xavier™/TX2™ development kit for using e-CAM20\_CUXVR MIPI camera, you can refer to the following documents to understand more about e-CAM20\_CUXVR.

- *e-CAM20\_CUXVR Release Notes*
- *e-CAM20\_CUXVR Release Package Manifest*
- *e-CAM20\_CUXVR eCAM ArgusCamera Installation Guide*
- *e-CAM20\_CUXVR Linux App User Manual*
- *e-CAM20\_CUXVR eCAM Argus MultiCamera App User Manual*

# Glossary

---

**API:** Application Programming Interface.

**CMOS:** Complementary Metal Oxide Semiconductor.

**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-CAM20\_CUXVR 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	19-May-2019	Initial Draft	Camera Dev Team
1.1	07-June-2019	Updated gcc toolchain	Camera Dev Team
1.2	25-June-2019	Updated Build Steps and upgrade procedure	Camera Dev Team
1.3	02-july-2019	New extract command for L4T	Camera Dev Team
1.4	09-July-2019	Modified for four camera setup	Camera Dev Team
1.5	29-july-2019	Added four camera setup support for Jetson TX2™	Camera Dev. Team
1.6	13-November-2019	Changes for new Release package structure	Camera Dev. Team
1.7	27-January-2020	Rootfs modifications steps update	Camera Dev. Team
1.8	25-March-2020	Misc. command corrections & steps update for new L4T Packages	Camera Dev. Team