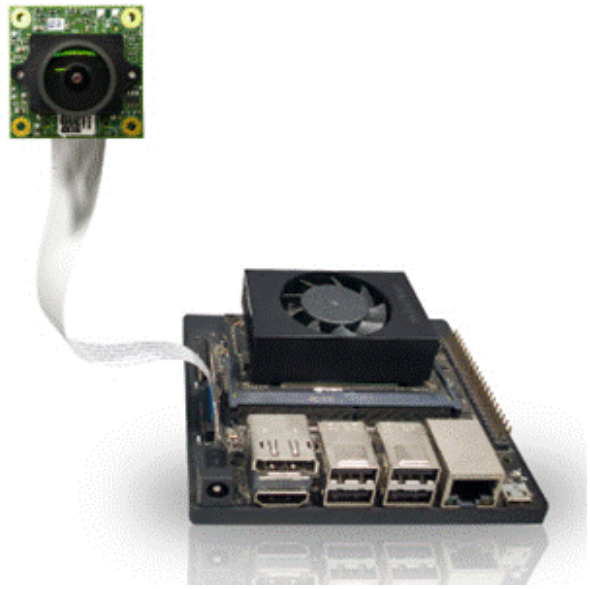
e-CAM24_CUNX

Developer Guide



Version 2.3

e-con Systems

8/19/2021

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

INTRODUCTION TO E-CAM24_CUNX 4

SOFTWARE REQUIREMENTS 5

PREREQUISITES FOR JETSON XAVIER NX 6

SETTING UP THE ENVIRONMENT 6
DOWNLOADING THE REQUIREMENTS 7
EXTRACTING AND PREPARING L4T 8
EXTRACTING THE RELEASE PACKAGE 8

INSTALLATION PROCEDURE FOR JETSON XAVIER NX 9

BUILDING FROM SOURCE 9
DOWNLOADING AND CONFIGURING THE KERNEL 9
BUILDING AND INSTALLING THE KERNEL 10
MODIFYING THE ROOTFS 11
FLASHING THE JETSON XAVIER NX DEVELOPMENT KIT 11

PREREQUISITES FOR JETSON NANO 13

SETTING UP THE ENVIRONMENT 13
DOWNLOADING THE REQUIREMENTS 14
EXTRACTING AND PREPARING L4T 15
EXTRACTING THE RELEASE PACKAGE 15

INSTALLATION PROCEDURE FOR JETSON NANO 16

BUILDING FROM SOURCE 16
DOWNLOADING AND CONFIGURING THE KERNEL 16
BUILDING AND INSTALLING THE KERNEL 17
MODIFYING THE ROOTFS 18
FLASHING THE JETSON NANO DEVELOPMENT KIT 18

LOADING THE DRIVERS 21

INSTALLING THE SAMPLE APPLICATION 22
USING THE SAMPLE APPLICATIONS WITH E-CAM24_CUNX 22

TROUBLESHOOTING 23

FAQ 24

WHAT'S NEXT? 25

GLOSSARY 26

SUPPORT 27

Introduction to e-CAM24_CUNX

e-con Systems is a leading Embedded Product Design Services Company which specializes in advanced camera solutions. e-CAM24_CUNX is a new MIPI camera which uses the AR0234 camera module. It is a 2-Lane module connected to the Jetson Xavier NX™/Jetson Nano™ development kit launched by e-con Systems. The prebuilt driver for this camera along with the camera board is provided by e-con Systems.

The NVIDIA® Jetson Xavier NX™/Jetson 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. Jetson Xavier NX™/Jetson Nano™ development kit supports dual camera.

e-CAM24_CUNX is 2.3 MP custom lens camera module based on 1/2.6" AR0234CS CMOS image sensor from ON Semiconductor®. 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. However, this camera can also be utilized by any V4L2 application.

e-CAM24_CUNX has a 2.3 MP color camera with S-mount (also known as M12 board lens) lens holder. The S-mount is one of the most used small form factor lens mounts for board cameras. e-CAM24_CUNX camera is interfaced to J1 and J9 connectors of the Jetson Xavier NX™ or is interfaced with J13 connector of Jetson Nano™ (A02 revision) or J13 or J49 connectors of the Jetson Nano™ (B01 revision) development kit using the ACC_XVRNX_MIPICAMERA_ADP board.

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 Xavier NX™/Jetson Nano™ development kit for using eCAM24_CUNX.

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 Xavier NX

This section describes the requirements to use e-CAM24_CUNX on the Jetson Xavier NX™ development kit.

The prerequisites are as follows:

- Host PC which runs Ubuntu 18.04 (64-bit).
- Download and install lbzip2 package.
- NVIDIA® provided L4T release and corresponding sample rootfs for Jetson Xavier NX™ development kit.
- A USB cable (micro-USB port) to plug into the recovery port of the Jetson Xavier NX™ development kit.
- Power cable (19V) to power the Jetson Xavier NX™ board.
- Micro SD card must be connected to the respective slot.

Please refer to the *e-CAM24_CUNX_Release_Package_Manifest_<REV>.pdf* to know the contents of release package and their description.

Setting Up the Environment

The steps to setup the environment 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=$TOP_DIR/e-CAM24_CUNX_JETSON
_<L4T_version>_<release_date>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export CROSS_COMPILE=aarch64-linux-gnu-
export CROSS32CC=arm-linux-gnueabihf-gcc
export TEGRA_KERNEL_OUT=$TOP_DIR/kernel_out
export
NVIDIA_SRC=$TOP_DIR/kernel_sources/Linux_for_Tegra/sou
rce/public
```

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

```
mv <location_of>/e-CAM24_CUNX_JETSON_
<L4T_version>_<release_date>_<release_version>.tar.gz
$TOP_DIR
```

Downloading the 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	7.3.1	https://developer.nvidia.com/embedded/dlc/l4t-gcc-7-3-1-toolchain-64-bit

   a. Run the following command to extract the package in Host PC.

```
tar -xf $HOME/Downloads/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
```

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

```
export PATH=<Tool_chain_extract_path>/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/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
```

3. Download the required L4T release package and sample rootfs 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 Xavier NX

S.NO	Title	Version	Download Link
1	L4T Jetson AGX Xavier™, Xavier NX™ and TX2 Driver Package	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/t186/jetson_linux_r32.6.1_aarch64.tbz2
2	L4T Jetson Xavier NX™ Sample Rootfs	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/t186/tegra_linux_sample-root-filesystem_r32.6.1_aarch64.tbz2

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

```
cp $HOME/Downloads/Jetson_Linux_R32.6.1_aarch64.tbz2
$TOP_DIR
cp $HOME/Downloads/Tegra_Linux_Sample-Root-
Filesystem_R32.6.1_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
sudo tar -xjpf Jetson_Linux_R32.6.1_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.

```
cd $LDK_ROOTFS_DIR
sudo tar -xjpf $TOP_DIR/Tegra_Linux_Sample-Root-
Filesystem_R32.6.1_aarch64.tbz2
```

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

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

Extracting the Release Package

Run the following commands to extract the e-CAM24_CUNX release package.

```
cd $TOP_DIR
tar -xaf e-CAM24_CUNX_JETSON_
<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to the *e-CAM24_CUNX_Release_Package_Manifest_<REV>.pdf*.

Installation Procedure for Jetson Xavier 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-CAM24_CUNX camera on the Jetson Xavier NX™ development kit to build the kernel.

Downloading and Configuring the Kernel

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

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

Table 4: Packages for Jetson Xavier NX

S.NO	Title	Version	Download Link
1	L4T Jetson AGX Xavier™ + Jetson Xavier NX™ + TX2 Sources	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/sources/t186/public_sources.tbz2

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
```

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-
CAM24_CUNX_JETSON_XAVIER-NX_<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-
CAM24_CUNX_JETSON_XAVIER-NX_<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-
CAM24_CUNX_JETSON_XAVIER-NX_<L4T_version>_dtb.patch --
dry-run
```

8. 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-
CAM24_CUNX_JETSON_XAVIER-NX_<L4T_version>_dtb.patch
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM24_CUNX_JETSON_XAVIER-NX_<L4T_version>_module.patch
--dry-run
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-
CAM24_CUNX_JETSON_XAVIER-NX_<L4T_version>_module.patch
```

Building and Installing the 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 Xavier NX™ development kit.

```
cd kernel/kernel-4.9/
make ARCH=arm64 O=$TEGRA_KERNEL_OUT
tegra_ecam_defconfig
make ARCH=arm64 O=$TEGRA_KERNEL_OUT Image -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT dtbs
sudo make ARCH=arm64 O=$TEGRA_KERNEL_OUT
modules_install INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
```

2. Run the following commands to build and install the e-CAM24_CUNX driver code.

```
cd $NVIDIA_SRC/e-CAM24_CUNX

make ARCH=arm64 KERNEL_PATH=$TEGRA_KERNEL_OUT ar0234

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

3. Run the following commands to copy the kernel and dtb file to Linux_for_Tegra (L4T_DIR) flashing path.

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

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra194-p3668-
all-p3509-0000-ar0234.dtb
$L4T_DIR/kernel/dtb/tegra194-p3668-all-p3509-0000.dtb
-f
```

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

Modifying the Rootfs

Run the following commands to modify additional files in the rootfs for the proper functioning of the e-CAM24_CUNX camera on the Jetson Xavier NX™ development kit.

```
sudo cp $RELEASE_PACK_DIR/misc/modules
$LDK_ROOTFS_DIR/etc/modules -f
```

Flashing the Jetson Xavier NX Development Kit

The steps to flash the Jetson Xavier NX™ 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 micro-USB cable to the Jetson Xavier NX™ and host PC.

   The top view of Jetson Xavier NX™ development board is shown below.

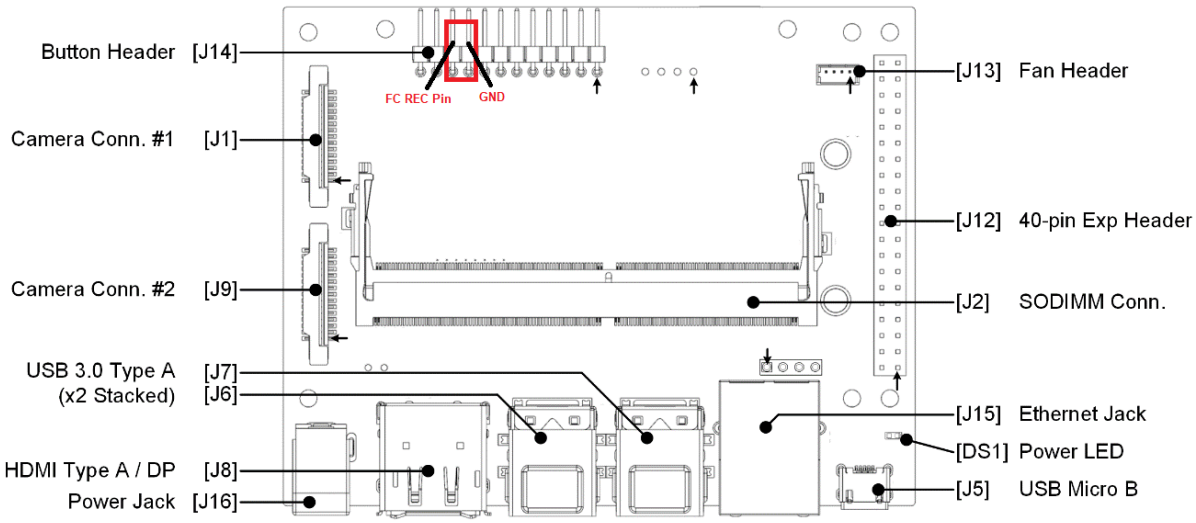


Figure 1: Top View of Jetson Xavier NX Development Board

4. Power ON the Jetson Xavier NX™ development kit.

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

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

```
lsusb
```

   The output message appears as shown below.

```
Bus 001 Device 102: ID 0955:7e19 NVidia Corp
```

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

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

   Note: Now, the entire micro-SD on the Jetson Xavier NX™ development kit will be erased. It will take about 10-30 minutes to complete depending on the host PC configuration.

7. Connect the Jetson Xavier NX™ board to a monitor and keyboard.
8. Reboot the board to complete the OS configuration, once the board reboots after flashing.

Prerequisites for Jetson Nano

This section describes the requirements to use e-CAM24_CUNX on the Jetson Nano™ development kit.

The prerequisites are as follows:

- Host PC which runs Ubuntu 18.04 (64-bit).
- Download and install lbzip2 package.
- NVIDIA® provided L4T release and corresponding sample rootfs for Jetson Nano™ development kit.
- A jumper pin across J48 button header (for A02 revision) and across J50 button header (for B01) to enable DC power.
- A USB cable (micro-USB port) to plug into the recovery port of the Jetson Nano™ development kit.
- Power cable (5V-4A) to power the Jetson Nano™ board.
- Micro SD card (16GB or higher) must be connected to the respective slot.

Please refer to the *e-CAM24_CUNX_Release_Package_Manifest_<REV>.pdf* to know the contents of release package and their description.

Setting Up the Environment

The steps to setup the environment 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=$TOP_DIR/e-CAM24_CUNX_JETSON
_<L4T_version>_<release_date>_<release_version>
export L4T_DIR=$TOP_DIR/Linux_for_Tegra
export LDK_ROOTFS_DIR=$TOP_DIR/Linux_for_Tegra/rootfs
export ARCH=arm64
export CROSS_COMPILE=aarch64-linux-gnu-
export CROSS32CC=arm-linux-gnueabihf-gcc
export TEGRA_KERNEL_OUT=$TOP_DIR/kernel_out
export
NVIDIA_SRC=$TOP_DIR/kernel_sources/Linux_for_Tegra/sou
rce/public
```

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

```
mv <location_of>/e-CAM24_CUNX_JETSON_
<L4T_version>_<release_date>_<release_version>.tar.gz
$TOP_DIR
```

Downloading the 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 5: Required Toolchain

S.NO	Title	Version	Download Link
1	GCC Toolchain	7.3.1	https://developer.nvidia.com/embedded/dlc/l4t-gcc-7-3-1-toolchain-64-bit

   a. Run the following command to extract the package in Host PC.

```
tar -xf $HOME/Downloads/gcc-linaro-7.3.1-2018.05-
x86_64_aarch64-linux-gnu.tar.xz
```

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

```
export PATH=<Tool_chain_extract_path>/gcc-linaro-
7.3.1-2018.05-x86_64_aarch64-linux-gnu/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
```

3. Download the required L4T release package and sample rootfs 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 6: Packages for Jetson Nano

S.NO	Title	Version	Download Link
1	L4T Jetson Nano™ Driver Package	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/t210/jetson-210_linux_r32.6.1_aarch64.tbz2

2	L4T Jetson Nano™ Sample Rootfs	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/t210/tegra_linux_sample-root-filesystem_r32.6.1_aarch64.tbz2

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

```
cp $HOME/Downloads/Jetson-210_Linux_R32.6.1_aarch64.tbz2 $TOP_DIR
cp $HOME/Downloads/Tegra_Linux_Sample-Root-Filesystem_R32.6.1_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
sudo tar -xjpf Jetson-210_Linux_R32.6.1_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.

```
cd $LDK_ROOTFS_DIR
sudo tar -xjpf $TOP_DIR/Tegra_Linux_Sample-Root-Filesystem_R32.6.1_aarch64.tbz2
```

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

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

Extracting the Release Package

Run the following commands to extract the e-CAM24_CUNX release package.

```
cd $TOP_DIR
tar -xaf e-CAM24_CUNX_JETSON_<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to the *e-CAM24_CUNX_Release_Package_Manifest_<REV>.pdf*.

Installation Procedure for Jetson 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-CAM24_CUNX camera on the Jetson Nano™ development kit to build the kernel.

Downloading and Configuring the Kernel

This section describes how you can download and configure the kernel for Jetson 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 Nanodevelopment kit are as follows:

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

Table 7: Packages for Jetson Nano

S.NO	Title	Version	Download Link
1	L4T Jetson Nano™ + TX1 Sources	32.6.1	https://developer.nvidia.com/embedded/l4t/r32_release_v6.1/sources/t210/public_sources.tbz2

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
```

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-
CAM24_CUNX_JETSON_NANO_<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-CAM24_CUNX_JETSON_NANO_<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-CAM24_CUNX_JETSON_NANO_<L4T_VERSION>_dtb.patch --dry-run
```

8. 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-CAM24_CUNX_JETSON_NANO_<L4T_VERSION>_dtb.patch
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM24_CUNX_JETSON_NANO_<L4T_VERSION>_module.patch --dry-run
```

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

```
patch -p1 -i $RELEASE_PACK_DIR/Kernel/e-CAM24_CUNX_JETSON_NANO_<L4T_VERSION>_module.patch
```

Building and Installing the 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 Nano™ development kit.

```
cd kernel/kernel-4.9/
make ARCH=arm64 O=$TEGRA_KERNEL_OUT tegra_ecam_defconfig
make ARCH=arm64 O=$TEGRA_KERNEL_OUT Image -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules -j4
make ARCH=arm64 O=$TEGRA_KERNEL_OUT dtbs
sudo make ARCH=arm64 O=$TEGRA_KERNEL_OUT modules_install INSTALL_MOD_PATH=$LDK_ROOTFS_DIR
```

2. Run the following commands to build and install the e-CAM24_CUNX driver code.

```
cd $NVIDIA_SRC/e-CAM24_CUNX
make ARCH=arm64 KERNEL_PATH=$TEGRA_KERNEL_OUT ar0234
```

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

3. Run the following commands to copy the kernel and dtb file to Linux_for_Tegra (L4T_DIR) flashing path.

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

//For A02 revision carrier board

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra210-p3448-
0000-p3449-0000-a02-camera-ar0234.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0000-p3449-0000-
a02.dtb -f

//For B01 revision carrier board

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra210-p3448-
0000-p3449-0000-b00-dual-camera-ar0234.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0000-p3449-0000-
b00.dtb -f

//For NANO 2GB revision carrier board

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra210-p3448-
0003-p3542-0000-camera-ar0234.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0003-p3542-0000.dtb
-f
```

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

Modifying the Rootfs

Run the following commands to modify additional files in the rootfs for the proper functioning of the e-CAM24_CUNX camera on the Jetson Nano™ development kit.

```
sudo cp $RELEASE_PACK_DIR/misc/modules
$LDK_ROOTFS_DIR/etc/modules -f
```

Flashing the Jetson Nano Development Kit

The steps to flash the Jetson Nano™ A02 revision kit in recovery mode are as follows:

1. Ensure a jumper is connected across J48 button header to enable DC power.
2. Connect the micro USB cable to the Jetson Nano™ and host PC.
3. Identify the J40 button header in Jetson Nano™ development kit for A02 carrier board as shown below.

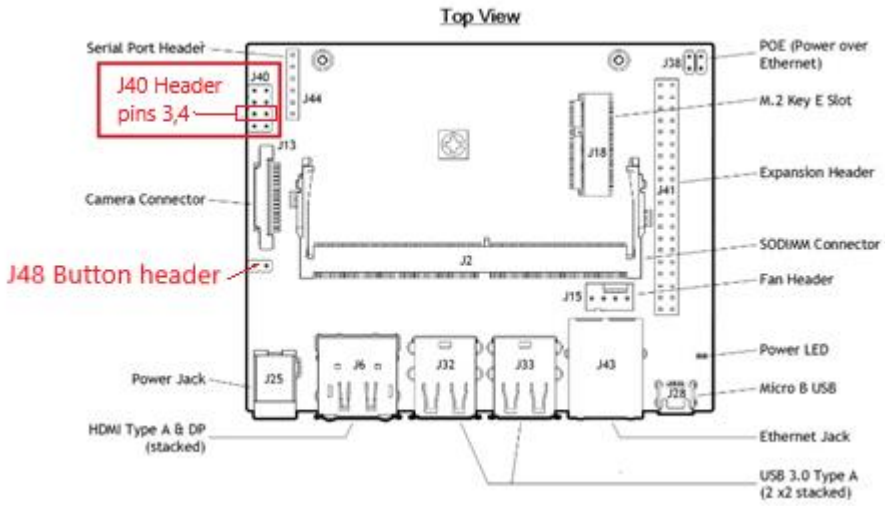


Figure 2: Top View of Jetson Nano Development Board (A02 Revision)

4. Set the board to recovery mode, as mentioned in below steps:

   a. Power off the board.
   b. Connect the jumper pin to the pin 3 and pin 4 of the J40 button header.
   c. Power on the Jetson Nano™ development kit.

The steps to set the Jetson Nano™ B01 revision kit in recovery mode is as follows:

1. Power OFF the kit.
2. Iderntify the J50 button header as shown below.

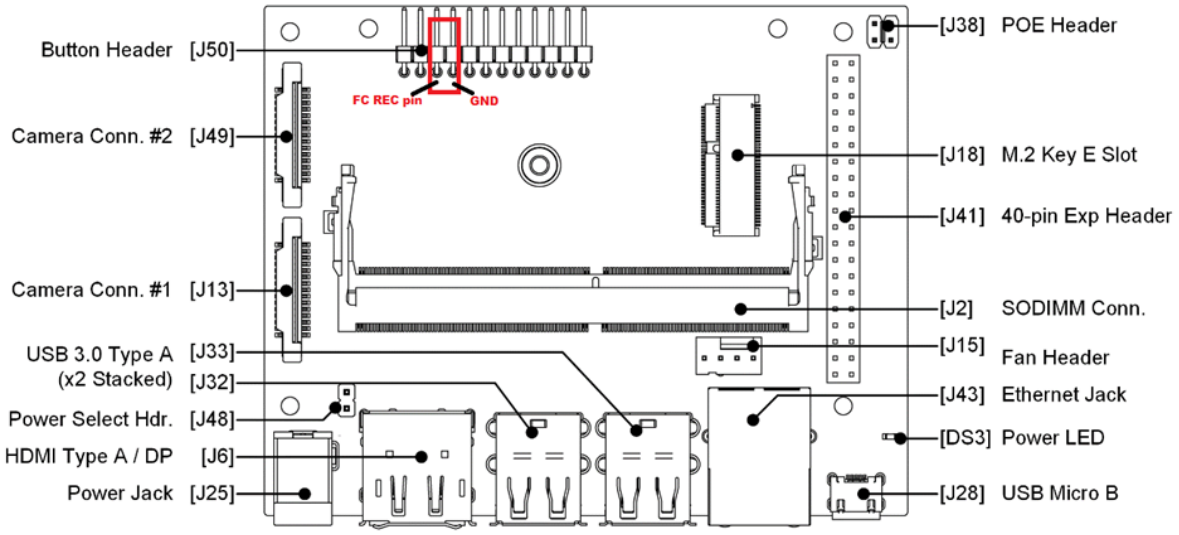


Figure 3: Top View of Jetson Nano (B01) Development Board

3. Place the Jumper across FC_REC and GND to enter recovery mode.
4. Power ON the Jetson Nano ™ development kit.

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

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

```
lsusb
```

The output message appears as shown below.

```
Bus 001 Device 102: ID 0955:7f21 NVidia Corp.
```

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

```
cd $L4T_DIR
sudo ./flash.sh jetson-nano-qspi-sd mmcblk0p1
 // for NANO 2GB
sudo ./flash.sh jetson-nano-2gb-devkit mmcblk0p1
```

Note: Now, the entire micro-SD on the Jetson Nano™ development kit will be erased. It will take about 10-30 minutes to complete depending on the host PC configuration.

7. Connect the Jetson Nano™ board to a monitor and keyboard.
8. Once the board is rebooted after flashing, complete the OS configuration.

Loading the Drivers

This section describes how to load the drivers, install the sample application, and use the sample application with e-CAM24_CUNX.

The module drivers for e-CAM24_CUNX will be loaded automatically in the Jetson™ development kit during booting.

The steps to load the drivers are as follows:

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

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

The output message appears as shown below for Jetson Xavier NX™ development kit.

```
subdev ar0234 9-0042 bound
subdev ar0234 10-0042 bound
```

The output message indicates that the camera is initialized properly for Jetson Xavier NX™.

The output message appears as shown below for A02 revision carrier board.

```
subdev ar0234 6-0042 bound
```

The output message indicates that the camera is initialized properly for Jetson Nano™ A02 carrier board

The output message appears as shown below for B01 revision carrier board.

```
subdev ar0234 7-0042 bound
subdev ar0234 8-0042 bound
```

The output message indicates that the camera is initialized properly for the Jetson Nano™ B01 carrier board

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™ development kit.

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

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

Table 8: Default Login Credentials

Fields	Inputs
Username	nvidia
Password	nvidia

Installing the Sample Application

The e-CAM24_CUNX guvcviewer or ecam_tk1_guvcview is a simple GTK+ interface for capturing and viewing video from the devices supported on the Jetson™ 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>.pdf* for the procedure to build and install ecam_tk1_guvcview application on the Jetson™ development kit.

Using the Sample Applications with e-CAM24_CUNX

To use the ecam_tk1_guvcview application with e-CAM24_CUNX, please refer to the *e-CAM24_CUNX_Linux_App_User_Manual_<REV>.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.

After flashing, the Jetson Xavier NX™/Jetson Nano™ board is not booting, or the display is blank. How to solve this issue?

To solve this issue, you can perform the following:

- Use the correct command with sudo permission whenever needed.
- Use the PC with Ubuntu 18.04 64-bit for flashing.
- Maintain enough free space in hard disk before flashing.

Flashing Jetson™ 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 python2.7.

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

2. Retry flashing the Jetson™ development kit.

1. **I have flashed L4T32.6.1 in Jetson™ development kit already. What are the steps to install the binaries?**

   Refer to the *e-CAM24_CUNX_Getting_Started_Manual_<REV>.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.

3. **After following the developer guide procedure and flashing image to SD-Card of size >= 16GB, the SD card size shrinks to 14GB. How to overcome this?**

   This is a known issue. Run the following commands after following the developer guide till flashing step. For more information, click this link.

```
cd $L4T_DIR/tools
/* for A02 board*/
sudo ./jetson-disk-image-creator.sh -o sd-blob_A02.img
-b jetson-nano -r 200
/* for B01 board*/
sudo ./jetson-disk-image-creator.sh -o sd-blob-B01.img
-b jetson-nano -r 300
/* for Xavier NX board */
sudo ./jetson-disk-image-creator.sh -o sd-blob_NX.img
-b jetson-xavier-nx-devkit
/* Flash the created image to desired SD-card */
/* <X> is the sd card detected by host pc */
sudo dd if=sd-blob_<rev>.img of=/dev/sd<X> bs=1M
oflag=direct status=progress
```

What's Next?

After understanding how to setup the Jetson Xavier NX™/Jetson Nano™ development kit using e-CAM24_CUNX MIPI camera, you can refer to the following documents to understand more about e-CAM24_CUNX.

- *e-CAM24_CUNX Release Notes*
- *e-CAM24_CUNX Release Package Manifest*
- *e-CAM_TK1-GUVCView Build and Install Guide*
- *e-CAM24_CUNX Linux App User Manual*

Glossary

API: Application Programming Interface.

CMOS: Complementary Metal Oxide Semiconductor.

DTB: Device Tree Blob.

Micro SD: micro-Secure Digital.

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.

OS: Operating Systems.

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-CAM24_CUNX 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	15-Dec-2020	Initial draft	Camera Dev Team
2.0	30-Dec-2020	Nano Support Added	Camera Dev Team
2.1	09-Mar-2021	Updated the Building from source section	Camera Dev Team
2.2	05-Jul-2021	Updated to L4T 32.5.1	Camera Dev Team
2.3	19-Aug-2021	Updated to L4T 32.6.1	Camera Dev Team