# e-CAM30_ CUNANO

# Developer Guide

Version 1.2

e-con Systems

6/21/2019

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-CAM30_CUNANO

e-con Systems is a leading Embedded Product Design Services Company which specializes in advanced camera solutions. e-CAM30_CUNANO is a new MIPI camera which uses the AR0330 camera module. It is a 2-Lane module connected to the 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 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.

e-CAM30_CUNANO has a 3.4 MP color camera 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. e-CAM30_CUNANO camera contains 1/3" AR0330 CMOS image sensor from On Semiconductor® and is interfaced to the J13 camera connector of the Jetson Nano™ development kit using the ACC_NANO_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 shown 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 Terminal |
| Orange | Commands running in Jetson Nano development kit |

This document explains how to setup the Jetson Nano™ development kit for using e-CAM30_CUNANO camera.

## Software Requirements

The software requirements are as follows:

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

# Prerequisites

This section describes the requirements to use e-CAM30_CUNANO on the Jetson Nano™ 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 Nano™ development kit.
- A kernel image , device tree blob (DTB) file and module drivers for the e-CAM30_CUNANO camera. The release package contains a kernel binary (Image) ,DTB files and module drivers, which you can download and rebuild from source.
- A jumper pin connected across J48 button header 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 must be connected to the J501 slot.
- A jumper pin must be connected to the pin 3 and pin 4 of J40 button header.

Please refer to the *e-CAM30_CUNANO-Release_Package_Manifest.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
export TOP_DIR=<absolute path to>/top_dir
export RELEASE_PACK_DIR=$TOP_DIR/e-CAM30_CUNANO_JETSON_NANO_<L4T_version>_<release_date>_<release_version>
export L4T_DIR=$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 KERNEL_PATH=$TOP_DIR/kernel_out
```

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

```
mv <location of>/e-
CAM30_CUNANO_JETSON_NANO_<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. 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. Run the following command on the host Linux PC to install the necessary tools.

```
sudo apt-get install gcc-arm-linux-gnueabihf gcc-
aarch64-linux-gnu build-essential
```

2. 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 package to staging directory are as follows:

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

   **Table 2: Packages for Jetson Nano**

   | S.NO | Title | Version |
   |---|---|---|
   | 1 | L4T Jetson Nano™ Driver Package | 32.1 |
   | 2 | L4T Jetson Nano™ Sample Rootfs | 32.1 |

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

```
cp $HOME/Downloads/Jetson-Nano-
Tegra210_Linux_R32.1.0_aarch64.tbz2 $TOP_DIR

cp $HOME/Downloads/Jetson-Nano-Tegra_Linux_Sample-
Root-Filesystem_R32.1.0_aarch64.tbz2 $TOP_DIR
```

## Extracting and Preparing L4T

The steps for extracting and preparing L4T must be performed in host PC 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 -xjf Jetson-Nano-
Tegra210_Linux_R32.1.0_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.

```
cd $LDK_ROOTFS_DIR
sudo tar -xjpf $TOP_DIR/Jetson-Nano-Tegra_Linux_Sample-Root-Filesystem_R32.1.0_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-CAM30_CUNANO release package.

```
cd $TOP_DIR
tar -xaf e-CAM30_CUNANO_JETSON_NANO_<L4T_version>_<release_date>_<release_version>.tar.gz
```

To know more about the release package, please refer to the *e-CAM30_CUNANO-Release_Package_Manifest.pdf*.

Please refer to the *Installation Procedure* section to use prebuilt files or build kernel with support for e-CAM30_CUNANO. The procedure would require flashing the micro SD of the Jetson Nano™ development kit for erasing the pre-existing contents.

Please refer to the *Upgrade Procedure* section to upgrade the Jetson Nano™ development kit which is already running L4T version and enable support for e-CAM30_CUNANO without flashing the micro SD. The procedure will preserve the existing rootfs of Jetson Nano™ development kit.

# Installation Procedure

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

You can choose to use either the prebuilt files  in the release package or build the own kernel with a customized configuration. For using the prebuilt files, please follow the steps in *Using the Prebuilt Files* section. If a customized kernel is required, please follow the steps in *Building from Source* section.

## Using the Prebuilt Files

The prebuilt files present in the e-CAM30_CUNANO release package can be used for making the camera to work quickly.

### Installing the Kernel and Drivers

Run the following commands to replace the NVIDIA® provided kernel image with the prebuilt kernel image  provided in the release package.

```
sudo cp $RELEASE_PACK_DIR/Kernel/Binaries/Image
$L4T_DIR/kernel/ -f

sudo tar jxpmf
$RELEASE_PACK_DIR/Kernel/Binaries/kernel_supplements.tar.
bz2 -C $LDK_ROOTFS_DIR/

sudo cp $RELEASE_PACK_DIR/Kernel/Binaries/tegra210-p3448-
0000-p3449-0000-a02-camera-ar0330.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0000-p3449-0000-
a02.dtb -f
```

### Modifying the Rootfs

Run the following command to modify additional files in the rootfs for the proper functioning of the e-CAM30_CUNANO camera on the Jetson Nano™ development kit.

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

### Flashing the Jetson Nano Development Kit

The steps to flash the Jetson Nano™ development kit 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 by referring the below picture.

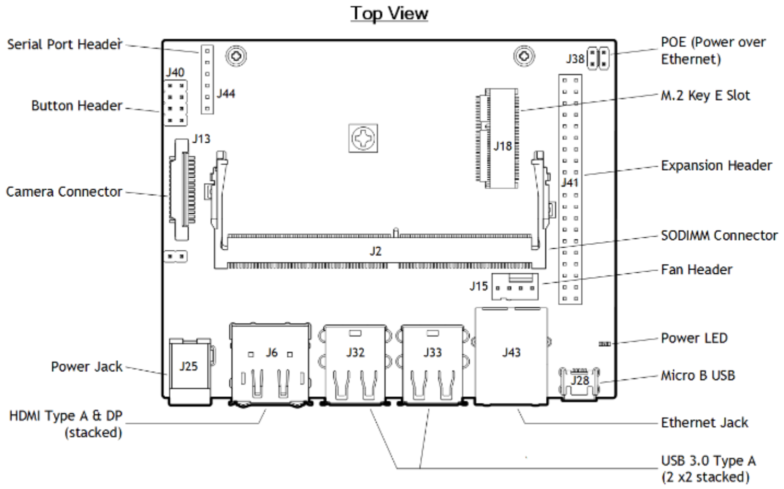

**Figure 1: Mechanical Diagram of Jetson Nano Development Kit**

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.

   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.

   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.
   ```

5. 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
   ```

   **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.

6. Reboot and connect the Jetson Nano™ board to a monitor and keyboard to complete the OS configuration, once flashing is completed.

## Building from Source

You can use a patch file provided by e-con Systems to use your own kernel image binary and modules along with the e-CAM30_CUNANO 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 Nano™ development kit are as follows:

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

**Table 3: Packages for Jetson Nano**

| Title | Version |
|---|---|
| L4T Sources | 32.1 |

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

```
cp $HOME/Downloads/Jetson-Nano-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 Jetson-Nano-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/Source/e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_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/Source/e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_kernel.patch
```

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

```
patch -p2 -i $RELEASE_PACK_DIR/Kernel/Source/e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_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 -p2 -i $RELEASE_PACK_DIR/Kernel/Source/e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_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/Source/ e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_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/Source/e-
CAM30_CUNANO_JETSON_NANO_L4T32.1.0_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 $TOP_DIR/public_sources/kernel/kernel-4.9/

make O=$TEGRA_KERNEL_OUT tegra_ecam_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 $TOP_DIR/public_sources/e-CAM30_CUNANO

make

sudo make -C $TEGRA_KERNEL_OUT
M=$TOP_DIR/public_sources/e-CAM30_CUNANO
INSTALL_MOD_PATH=$LDK_ROOTFS_DIR modules_install

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

sudo cp
$TEGRA_KERNEL_OUT/arch/arm64/boot/dts/tegra210-p3448-
0000-p3449-0000-a02-camera-ar0330.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0000-p3449-0000-
a02.dtb -f
```

2. Follow the steps in *Modifying the Rootfs* and *Flashing the Jetson Nano Development Kit* sections to make the Jetson Nano™ 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 to work.

# Upgrade Procedure

This section describes the steps for upgrading the kernel image, supplements, and DTB file.

**Note**: The procedure in this section is followed when you have already flashed the standard **L4T_R32.1** package for Jetson Nano™ development kit.

The existing kernel image and the DTB file in the Jetson Nano™ development kit will be replaced by the prebuilt kernel image and dtb file given with the release package. The existing rootfs in the Jetson Nano™ development kit will be preserved by following the procedures described in this section. You must backup the existing kernel image and DTB file before proceeding the procedures.

**Note**: Make sure that the Jetson Nano™ development kit is booted. The kernel image and DTB file needs to be flashed from the host PC.

## Upgrading Kernel Image and Supplements

The steps to upgrade kernel image and supplements are as follows:

**Note**: You must setup the environment in a terminal window for Jetson Nano™ development kit.

1. Copy the release package from staging directory to the board.
2. Run the following commands to extract the release package from the Jetson Nano™ development kit.

```
mkdir top_dir/ -p
export TOP_DIR=<absolute path to>/top_dir
export RELEASE_PACK_DIR=<absolute_path_to>/top_dir/e-CAM30_CUNANO_JETSON_NANO_<L4T_version>_<release_date>_<release_version>
cp e-CAM30_CUNANO_JETSON_NANO_<L4T_version>_<release_date>_<release_version>.tar.gz $TOP_DIR/
cd $TOP_DIR
tar -xaf e-CAM30_CUNANO_JETSON_NANO_<L4T_version>_<release_date>_<release_version>.tar.gz
```

3. Run the following commands to extract and copy the e-CAM30_CUNANO camera driver and other driver modules given with the release package.

```
sudo tar xjpmf $RELEASE_PACK_DIR/Kernel/Binaries/kernel_supplements.tar.bz2 -C /
sudo cp $RELEASE_PACK_DIR/Rootfs/modules /etc/modules
-f
```

4. Run the following command to copy the e-CAM30_CUNANO kernel image to the boot directory.

```
sudo cp $RELEASE_PACK_DIR/Kernel/Binaries/Image
/boot/Image -f
```

## Upgrading DTB File by Flashing from Host PC

This section describes the steps for upgrading kernel image and DTB file by flashing from host PC for Jetson Nano™ development kit.

**Note**: Please refer to the *Setting Up the Environment* to *Extracting the Release Package* sections to prepare the flash environment in host PC. Ensure that the environmental values are preserved.

The steps for flashing the DTB to Jetson Nano™ development kit from the host PC are as follows:

1. Run the following command to copy e-con Systems prebuilt DTB file to the Linux_for_Tegra folder in the same terminal.

```
sudo cp $RELEASE_PACK_DIR/Kernel/Binaries/tegra210-p3448-
0000-p3449-0000-a02-camera-ar0330.dtb
$L4T_DIR/kernel/dtb/tegra210-p3448-0000-p3449-0000-
a02.dtb -f
```

2. Run the following command to navigate to the Linux_for_Tegra directory.

```
cd $L4T_DIR
```

3. Run the following command to check the presence of the system.img file which is required to flash the DTB file.

```
ls ./bootloader/system.img
```

   If there is no **system.img** file in the directory, run the following command to create this file and wait till the **system.img** file is created (this file will not get flashed to the board).

```
sudo ./flash.sh --no-flash jetson-nano-qspi-sd
mmcblk0p1
```

4. Ensure a jumper is connected across J48 button header to enable DC power.
5. Connect the micro USB cable between the host PC and the micro USB port of Jetson Nano™ development kit.
6. Set the Jetson Nano™ development kit 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.

If the board is successfully changed to recovery mode, the Jetson Nano™ development kit will be enumerated as a 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.

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

7. Run the following command to flash the DTB file to the Jetson Nano™ development kit from your host PC.

```
sudo ./flash.sh -r -k DTB jetson-nano-qspi-sd
mmcblk0p1
```

If the flash is successful, the output message appears as shown below.

```
*** The [kernel-dtb] has been flashed successfully.
***
```

8. Reboot the Jetson Nano™ development kit.

The Jetson Nano™ development kit will now be running the latest binaries.

# Loading the Drivers

This section describes how to load the drivers, install the sample application and use the sample application with e-CAM30_CUNANO.

The module drivers for e-CAM30_CUNANO will be loaded automatically in the Jetson Nano™ 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 "ar0330"
```

   The output message appears as shown below.

```
subdev ar0330 6-0042 bound
```

   The output message indicates that the camera is initialized properly.

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

```
ls /dev/video0
```

   The output message appears as shown below.

```
video0
```

   If no other cameras are connected to the Jetson Nano™ development kit. This video node can be utilized by any V4L2 application for viewing the camera preview.

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

**Table 4: Default Login Credentials**

| Fields | Inputs |
|---|---|
| Username | nvidia |
| Password | nvidia |

## Installing the Sample Application

The e-CAM30_CUNANO guvcviewer or ecam_tk1_guvcview is a simple GTK+ interface for capturing and viewing video from the devices supported on the Jetson Nano™ 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 setting 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.pdf* for the procedure to build and install ecam_tk1_guvcview application on the Jetson Nano™ development kit.

## Using the Sample Applications with e-CAM30_CUNANO

To use the ecam_tk1_guvcview application with e-CAM30_CUNANO, please refer to the *e-CAM30_CUNANO_Linux_App_User_Manual.pdf* for the procedure to use ecam_tk1_guvcview respectively.

# Troubleshooting

In this section, you can view the commonly occurring issue and their troubleshooting step.

**I have flashed the Jetson Nano™ board with quick start package. After flashing, the board is not booting, or the display is blank. How to solve this issue?**

To solve this issue, please follow these steps:

- Use the correct command with sudo permission whenever needed to extract the package.
- Use the PC with Ubuntu 16.04 64-bit for flashing.
- Maintain enough free space in hard disk before flashing.

# FAQ

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

   Yes, please refer to the *Upgrade Procedure* section to upgrade the modules, kernel image and device tree.

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 Nano™ development kit using e-CAM30_CUNANO MIPI camera, you can refer to the following documents to understand more about e-CAM30_CUNANO.

- *e-CAM30_CUNANO Release Notes*
- *e-CAM30_CUNANO Release Package Manifest*
- *e-CAM_TK1-GUVCView Build and Install Guide*
- *e-CAM30_CUNANO 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-CAM30_CUNANO 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 | 25-Apr-2019 | Initial draft | Camera Dev Team |
| 1.1 | 08-May-2019 | Updated buiilding from source and upgrade procedure sections. | Camera Dev Team |
| 1.2 | 21-June-2019 | Updated Installation Procedure Section | Camera Dev Team |