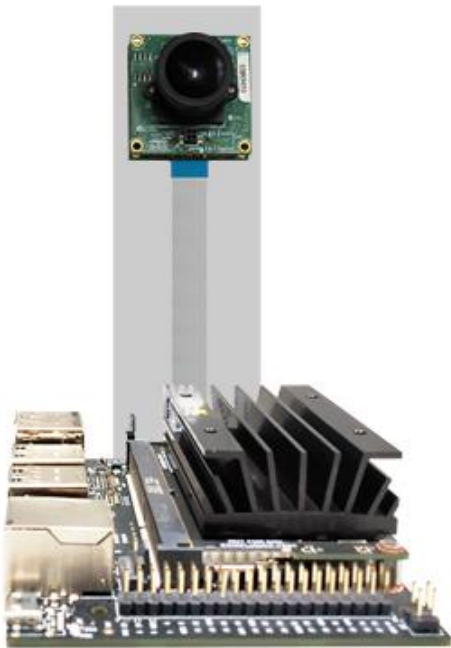


e-CAM30\_  
CUNANO

# Gstreamer Usage Guide



Version 1.1

e-con Systems

5/9/2019

### **Disclaimer**

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

# Contents

<b>INTRODUCTION TO GSTREAMER</b>	<b>3</b>
<b>SETUP PROCEDURE</b>	<b>4</b>
<b>TESTED GSTREAMER EXAMPLES</b>	<b>5</b>
<b>TROUBLESHOOTING</b>	<b>7</b>
<b>WHAT'S NEXT?</b>	<b>8</b>
<b>GLOSSARY</b>	<b>9</b>
<b>SUPPORT</b>	<b>10</b>

# Introduction to Gstreamer

---

Gstreamer is a powerful and flexible multimedia framework with a lot of capabilities. It supports various features such as capturing, displaying, encoding, and decoding of image data. While using Gstreamer, the possibilities of manipulating data are limitless.

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 development board
Red	Output message in development board
Orange	Commands running in Client PC

This document explains how to install Gstreamer-1.0 on the Jetson Nano™ development kit and use it with the available plugins.

# Setup Procedure

---

This section describes how to verify the setup before start testing Gstreamer pipelines.

To know the details of available plugins, please refer to the *L4T32.1\_Accelerated\_Gstreamer\_User\_Guide.pdf* from [Jetson Download Center](#).

The steps to verify the setup before testing Gstreamer pipelines are as follows:

1. Run the following commands to check the Gstreamer-1.0 version.

```
$ gst-inspect-1.0 --version
$ gst-inspect-1.0 version 1.14.1
$ GStreamer 1.14.1
```

**Note:** Make sure that e-CAM30\_CUNANO is connected and the required drivers are loaded.

During booting, the module drivers for e-CAM30\_CUNANO will be loaded automatically in the Jetson Nano™ development kit.

2. Run the following command to confirm 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.

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

```
$ ls /dev/video0
```

The output message appears as shown below.

```
video0
```

# Tested Gstreamer Examples

---

This section describes some of the tested Gstreamer commands which work on the Jetson Nano™ development kit.

**Note:** Please run the following commands to change the power mode to maximum for better performance and to get maximum frame rate.

```
$ sudo nvpmodel -m 0
$ sudo jetson_clocks
```

## Example 1: Streaming 3.4 MP is resized to 1080P (HW accelerated)

Run the following command to resize the streaming video from 3.4 MP to 1080P resolution.

```
$ gst-launch-1.0 v4l2src ! "video/x-raw,
format=(string)UYVY, width=(int)2304, height=(int)1536" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420, width=(int)1920, height=(int)1080" !
nvoverlaysink overlay-w=1920 overlay-h=1080 sync=false
```

## Example 2: Streaming FHD video (HW accelerated)

Run the following command to stream FHD video.

```
$ gst-launch-1.0 v4l2src ! "video/x-raw,
format=(string)UYVY, width=(int)1920, height=(int)1080" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420" ! nvoverlaysink sync=false
```

## Example 3: Record 3.4 MP in H.264 format to a video file (HW accelerated)

Run the following command to record 3.4 MP video in H.264 format to a video file.

```
$ gst-launch-1.0 v4l2src ! "video/x-raw,
format=(string)UYVY, width=(int)2304, height=(int)1536" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420" ! omxh264enc qp-range=20,20:20,20:-
1,-1 ! matroskamux ! queue ! filesink location=<file.mkv>
```

## Example 4: Record FHD in H.264 format to a video file (HW accelerated)

Run the following command to record FHD video in H.264 format to a video file.

```
$ gst-launch-1.0 v4l2src ! "video/x-raw,
format=(string)UYVY, width=(int)1920, height=(int)1080" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420" ! omxh264enc qp-range=20,20:20,20:-
1,-1 ! matroskamux ! queue ! filesink location=<file.mkv>
```

## Example 5: Playback of saved video file (HW accelerated)

Run the following command to playback the saved video file.

```
$ gst-launch-1.0 filesrc location=/home/nvidia/<file.mkv>
! matroskademux ! h264parse ! omxh264dec ! nvoverlaysink
```

### Example 6: Network streaming H.264 encoded FHD video using RTP over UDP (HW accelerated)

Run the following command to stream the video captured by camera connected with Jetson Nano™ development kit.

```
$ gst-launch-1.0 v4l2src device=/dev/video0 ! "video/x-raw, format=(string)UYVY, width=(int)1920, height=(int)1080" ! nvvidconv ! "video/x-raw(memory:NVMM), format=(string)I420" ! omxh264enc qp-range=35,35:35,35:-1,-1 ! rtph264pay mtu=60000 ! udpsink clients=<ip_address>:<port_no> sync=false
```

**Note:** Do not close the application in Jetson Nano™ development kit.

Run the following command in the Client device to view the video.

```
$ gst-launch-1.0 udpsrc port=<port_no> caps="application/x-rtp, media=(string)video, clock-rate=(int)90000, encoding-name=(string)H264, sprop-parameter-sets=(string)\\"Z0JAKpWgHgCJ+VA\\=\\",aM48gA\\=\\" payload=(int)96" ! rtph264depay ! decodebin ! autovideosink sync=false
```

**Note:** Replace **<ip\_address>** with the IP address of the Client device (For example, 192.168.6.100) and **<port\_no>** with the port number of the Client device (For example, 5000).

### Example 7: Capturing 3.4 MP still image

Run the following command to capture 3.4 MP still image.

```
$ gst-launch-1.0 v4l2src num-buffers=1 ! "video/x-raw, format=(string)UYVY, width=(int)2304, height=(int)1536" ! jpegenc ! filesink location=<filename>.jpg
```

**Note:** Change **<filename>** with a name (For example, Capture\_1), to which the image will get stored.

# Troubleshooting

---

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

**During network streaming, I face issues in video quality and frame rate. How to overcome this issue?**

Network streaming quality and frame rate depends on the network bandwidth and receiver decode capability. In the receiver end, if you use software decoder, the frame rate and quality will get affected. Therefore, avoid using software decoder in the receiver end to resolve this issue.



# What's Next?

---

After understanding the usage of Gstreamer application, you can refer to the following documents to understand more about e-CAM30\_CUNANO.

- *e-CAM30\_CUNANO Datasheet*
- *e-CAM30A\_CUMI0330\_MOD Datasheet*
- *e-CAM30\_CUNANO Lens Datasheet*
- *e-CAM30\_CUNANO Release Package Manifest*

# Glossary

---

**FHD:** Full HD (Industry name for 1920 x 1080 resolution).

**HD:** High Definition (Industry name for 1280 x 720 resolution).

**IP:** Internet Protocol.

**RTP:** Real-time Transport Protocol.

**UDP:** User Datagram Protocol.

# 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	09-May-2019	Updated jetson_clocks command instead of jetson_clocks.sh script	Camera Dev Team