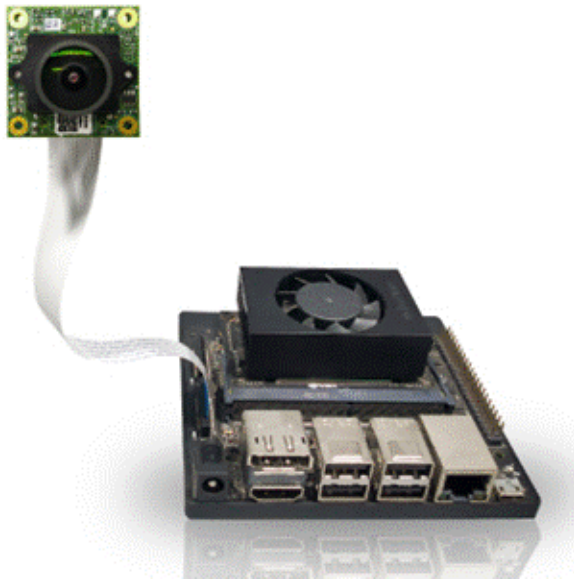


e-CAM24_CUNX

Gstreamer Usage Guide



e-con Systems

Your Product Development Partner

Version 2.3

e-con Systems

3/10/2022

Disclaimer

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

Contents

INTRODUCTION TO GSTREAMER	3
SETUP PROCEDURE	4
TESTED GSTREAMER EXAMPLES	6
TROUBLESHOOTING	8
WHAT'S NEXT?	9
GLOSSARY	10
SUPPORT	11

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 Xavier NX™/ 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 *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 command to check the Gstreamer-1.0 version.

```
$ gst-inspect-1.0 --version
```

The output message appears as shown below.

```
gst-inspect-1.0 version 1.14.5
GStreamer 1.14.5
```

Note: Make sure that e-CAM24_CUNX is connected and the required drivers are loaded.

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

2. Run the following command to confirm 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.

The output message appears as shown below for Jetson Nano development kit

```
//For A02 revision kit
subdev ar0234 6-0042 bound
//For B01 revision kit
subdev ar0234 7-0042 bound
subdev ar0234 8-0042 bound
```

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

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

The output message appears as shown below.

```
/dev/video
```

where (*) depends on the number for cameras connected to the Jetson development kit. The number of video nodes displayed depends on the number of cameras connected.

Tested Gstreamer Examples

This section describes some of the tested Gstreamer commands which work on the Jetson™ 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 nvpmode1 -m 0
$ sudo jetson_clocks
```

Example 1: Streaming 2.3 MP is resized to 1080P (HW accelerated)

Run the following command to resize the streaming video from 2.3 MP to 1080P resolution.

```
$ gst-launch-1.0 v4l2src device=/dev/video<X> ! "video/x-raw, format=(string)UYVY, width=(int)1920, height=(int)1200" ! 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 device=/dev/video<X> ! "video/x-raw, format=(string)UYVY, width=(int)1920, height=(int)1080" ! nvvidconv ! "video/x-raw(memory:NVMM), format=(string)I420" ! nvoverlaysink sync=false
```

Note: Replace **X** in **/dev/videoX** with appropriate number to stream from one or second camera.

Example 3: Record 2.3 MP in H.264 format to a video file (HW accelerated)

Run the following command to record 2.3 MP video in H.264 format to a video file.

```
$ gst-launch-1.0 v4l2src device=/dev/video<X> ! "video/x-raw, format=(string)UYVY, width=(int)1920, height=(int)1200" ! 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 device=/dev/video<X> ! "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™ development kit.

```
$ gst-launch-1.0 v4l2src device=/dev/video<X> ! "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™ 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 !
videoconvert ! glimagesink -e
```

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 2.3 MP still image

Run the following command to capture 2.3 MP still image.

```
$ gst-launch-1.0 v4l2src device=/dev/video<X> num-
buffers=1 ! "video/x-raw, format=(string)UYVY,
width=(int)1920, height=(int)1200" ! 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 commonly occurring issue and its troubleshooting step.

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-CAM24_CUNX.

- *e-CAM24_CUNX Datasheet*
- *e-CAM217_CUMI0234_MOD Datasheet*
- *e-CAM24_CUNX Lens Datasheet*
- *e-CAM24_CUNX 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.

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	10-Mar-2021	Updated the Device enumeration message	Camera Dev Team
2.2	30-Mar-2021	Updated the Sample Commands	Camera Dev Team
2.3	10-Mar-2022	Updated the Sample Commands	Camera Dev Team