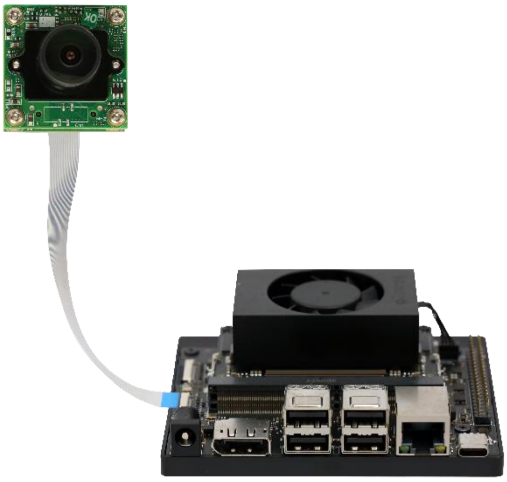
e-CAM25_CUONX

GStreamer Usage Guide



Version 1.4

e-con Systems

3/24/2025

Disclaimer

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

Contents

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

This document explains how to install and use Gstreamer-1.0 on the Jetson Orin NX™/Jetson Orin Nano™ development kit.

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_Rev_<ver>.pdf* from Jetson Download Center.

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

1. Run the following commands to install GStreamer-1.0 binaries and libraries on the Jetson Orin Nano™/Orin NX™ development kit.

```
$ sudo apt-add-repository universe
$ sudo apt-get update
$ sudo apt-get install nvidia-l4t-gstreamer
```

2. Run the following command to check the Gstreamer-1.0 version.

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

Output will be shown as below for Jetson Orin Nano:

```
gst-inspect-1.0 version 1.20.3
GStreamer 1.20.3
https://launchpad.net/distros/ubuntu/+source/gstreamer1.0
```

Note: Make sure that e-CAM25_CUONX is connected, and the required drivers are loaded.

During booting, the module drivers for e-CAM25_CUONX will be loaded automatically in the Jetson Orin™ development kit.

3. Run the following command to confirm whether the camera is initialized.

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

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

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

The output message indicates that the camera is initialized properly.

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

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

The output message appears as shown below.

```
/dev/video<X>
```

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 for Jetson Orin NX

This section describes some of the tested GStreamer commands which work on the Jetson Orin NX™ 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 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 nvv4l2camerasrc device=/dev/video<X> !
"video/x-raw(memory:NVMM), format=(string)UYVY,
width=(int)1920, height=(int)1200" ! nvvidconv !
"video/x-raw(memory:NVMM), format=(string)I420,
width=(int)1920, height=(int)1080" ! nv3dsink sync=false
```

Example 2: Streaming FHD video (HW accelerated)

Run the following command to stream FHD video.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X> !
"video/x-raw(memory:NVMM), format=(string)UYVY,
width=(int)1920, height=(int)1080" ! nvvidconv !
"video/x-raw(memory:NVMM), format=(string)I420,
width=(int)1920, height=(int)1080" ! nv3dsink 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 -e nvv4l2camerasrc device=/dev/video<x>
! "video/x-raw(memory:NVMM),
format=(string)UYVY,width=(int)1920,height=(int)1200,fram
erate=<fps>/1" ! nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420, width=(int)1920,height=(int)1200" !
nvv4l2h264enc ! h264parse ! 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 -e nvv4l2camerasrc device=/dev/video<x>
! "video/x-raw(memory:NVMM),
format=(string)UYVY,width=(int)1920,height=(int)1080,fram
erate=<fps>/1" ! nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420, width=(int)1920,height=(int)1080" !
nvv4l2h264enc ! h264parse ! 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=file.mkv !
matroskademux ! h264parse ! nvv4l2decoder ! nv3dsink
```

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

Run the following command to stream the H.264 encoded FHD video captured by camera connected with Jetson Orin™ development kit.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X>  !
"video/x-raw(memory:NVMM), format=(string)UYVY,
width=(int)1920, height=(int)1080" ! nvvidconv !
"video/x-raw(memory:NVMM), format=(string)NV12" !
nvv4l2h264enc control-rate=1 ! h264parse ! rtph264pay
mtu=1400 ! udpsink clients=<ip_address>:<port_no>
sync=false buffer-size=100000
```

Note: Do not close the application in Jetson Orin NX™ 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, encoding-
name=H264, sprop-parameter-
sets=(string)\"Z0JAKJZUA8ARPyo\\=\\,aM48gA\\=\\=\",
payload=(int)96" ! rtph264depay ! h264parse ! decodebin !
videoconvert ! 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 2.3 MP still image

Run the following command to capture 2.3 MP still image.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X>
num-buffers=1 ! "video/x-raw(memory:NVMM),
format=(string)UYVY, width=(int)1920, height=(int)1080" !
nvvidconv ! nvjpegenc ! filesink location=<filename>.jpg
```

Note: Change **<filename>** with a name (For example, Capture_1), to which the image will get stored.

Tested GStreamer Examples for Jetson Orin Nano

This section describes some of the tested GStreamer commands which work on the Jetson Orin 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 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 nvv4l2camerasrc device=/dev/video<X>  !
"video/x-raw(memory:NVMM),
format=(string)UYVY,width=(int)1920, height=(int)1200" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)I420,width=(int)1920, height=(int)1080" !
nv3dsink sync=false
```

Example 2: Streaming FHD video (HW accelerated)

Run the following command to stream FHD video.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X> !
"video/x-raw(memory:NVMM), format=(string)UYVY,
width=(int)1920, height=(int)1080" ! nvvidconv !
"video/x-raw(memory:NVMM), format=(string)I420,
width=(int)1920, height=(int)1080" ! nv3dsink 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 (SW accelerated)

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

```
$ gst-launch-1.0 -e nvv4l2camerasrc device=/dev/video<x>
! "video/x-
raw(memory:NVMM),format=(string)UYVY,width=(int)1920,heig
ht=(int)1200,framerate=<fps>/1" ! nvvidconv ! "video/x-
raw,format=I420,width=(int)1920,height=(int)1200" !
x264enc bitrate=10000 speed-preset=fast tune=zerolatency
! h264parse ! matroskamux ! filesink location=file.mkv
```

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

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

```
$ gst-launch-1.0 -e nvv4l2camerasrc device=/dev/video<x>
! "video/x-
raw(memory:NVMM),format=(string)UYVY,width=(int)1920,heig
ht=(int)1080,framerate=<fps>/1" ! nvvidconv ! "video/x-
raw,format=I420,width=(int)1920,height=(int)1200" !
x264enc bitrate=10000 speed-preset=fast tune=zerolatency
! h264parse ! matroskamux ! 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=file.mkv !
matroskademux ! h264parse ! nvv4l2decoder ! nv3dsink
```

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

Run the following command to stream the H.264 encoded FHD video captured by camera connected with Jetson Orin™ development kit.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X>  !
"video/x-raw(memory:NVMM), format=(string)UYVY,
width=(int)1920, height=(int)1080" ! nvvidconv !
"video/x-raw,format=I420" ! x264enc ! h264parse !
rtph264pay mtu=60000 ! udpsink
clients=<ip_address>:<port_no> sync=false
```

Note: Do not close the application in Jetson Orin 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, encoding-
name=H264, sprop-parameter-
sets=(string)\"Z0JAKJZUA8ARPyo\\=\\,aM48gA\\=\\=\",payloa
d=(int)96" ! rtph264depay ! h264parse ! decodebin !
videoconvert ! 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 2.3 MP still image

Run the following command to capture 2.3 MP still image.

```
$ gst-launch-1.0 nvv4l2camerasrc device=/dev/video<X>
num-buffers=1 ! "video/x-raw(memory:NVMM),
format=(string)UYVY, width=(int)1920, height=(int)1080" !
nvvidconv ! nvjpegenc ! 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 with video quality and frame rate. How to overcome this issue?

This is a known 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.

During network streaming and recoding the video in Jetson Orin Nano™ development kit, I face issues with video quality. How to overcome this issue?

This is a known issue. Video quality is depending on the encoder used. Jetson Orin Nano™ development kit doesn’t have a hardware encoder. If you are using a software encoder, the video quality will get affected.

What's Next?

After understanding the usage of GStreamer application, you can refer to the following documents to understand more about e-CAM25_CUONX.

- *e-CAM25_CUONX Datasheet*
- *e-CAM217_CUMI0234_MOD Datasheet*
- *e-CAM25_CUONX Lens Datasheet*
- *e-CAM25_CUONX 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.

V4L2: Video4Linux2 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-CAM25_CUONX 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-Apr-2023	Initial Draft	Camera Dev Team
1.1	24-May-2024	Updated gstreamer version for L4T36.3.0	Camera Dev Team
1.2	23-Jan-2025	Updated gStreamer version for L4T36.4.0	Camera Dev Team
1.3	24-Feb-2025	Updated v4l2src to nvv4l2camerasrc	Camera Dev Team
1.4	24-Mar-2024	Updated gStreamer recording commands	Camera Dev Team