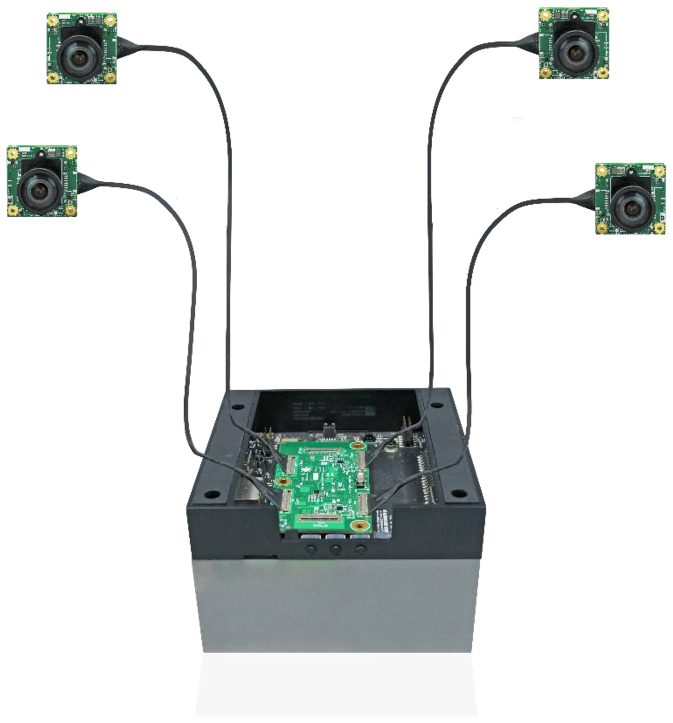
e-CAM56_CUOAGX

GStreamer Usage Guide



Version 1.4

e-con Systems

3/26/2024

e-con Systems

Think Camera. Think e-con.

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 the 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 GStreamer-1.0 on the Jetson AGX Orin™ development kit and use it with the available plugins.

Setup Procedure

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

To know the details of available plugins, please refer to the *Accelerated GStreamer* section in NVIDIA Jetson Linux Developer Guide.

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 AGX Orin™ development kit.

```
$ sudo apt-add-repository universe
$ sudo apt-get update
$ sudo apt-get install gstreamer1.0-tools
gstreamer1.0-alsa gstreamer1.0-plugins-base
gstreamer1.0-plugins-good gstreamer1.0-plugins-bad
gstreamer1.0-plugins-ugly gstreamer1.0-libav
```

Note: If the installation process stops with **could not get lock /var/lib/dpkg/lock** message, run the following command to remove the file and proceed with the installation.

```
$ sudo rm /var/lib/dpkg/lock
```

2. 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 1.16.3
GStreamer 1.16.3
https://launchpad.net/distros/ubuntu/+source/gstreamer
1.0
```

Note: Make sure that e-CAM56_CUOAGX is connected, and the required drivers are loaded.

During booting, the module drivers for e-CAM56_CUOAGX will be loaded automatically in the Jetson AGX Orin™ development kit.

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

```
$ sudo dmesg | grep -i "Detected eimx568 sensor"
```

The output message appears as shown below.

```
Detected eimx568 sensor
```

The output message shows depend upon number of cameras that are properly initialized.

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

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

The output message for four camera setup appears as shown below.

```
/dev/video*
```

Where (*) denotes the number of cameras connected to the Jetson AGX Orin™ development kit.

5. The resolutions supported by e-CAM56_CUOAGX in 4-lane and corresponding frame rates are listed below:

Table 2: Supported Resolutions and Frame Rates in 4-Lane Mode (Jetson AGX Orin)

Lane	Resolution	Frame Rate in 10-bit	Frame Rate in 12-bit
4	2432 x 2048	79	67
4	1920 x 1080	142	121
4	1280 x 720	202	172
4	640 x 480	280	240

Tested GStreamer Examples

This section describes some of the tested GStreamer commands which work on the Jetson AGX Orin™ 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
$ sudo $HOME/max-isp-vi-clks.sh
```

Example 1: Streaming 2048p from a single camera (HW accelerated).

Run the following command to stream 2048p video from a single camera.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> sensor-
mode=<m> ! "video/x-raw(memory:NVMM),
width=(int)2432,height=(int)2048, format=(string)NV12,
framerate=(fraction)<f>/1" ! nv3dsink -e
```

Note:

- Replace **<n>** with the number of video device node, from which you need to stream.
- Replace **<m>** with the sensor mode in which you need to stream. Here 'm' indicates supported sensor modes i.e., 0 to 7. For 2048p, 'm' is either 0 or 1.
- Replace **<f>** with the frame rate supported for the mentioned sensor mode, please refer Table2.

Example 2: Streaming 2048p resized to 1080p from a single camera (HW accelerated).

Run the following command to stream 2048P video from a single camera.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> !
"video/x-raw(memory:NVMM),
width=(int)2432,height=(int)2048, format=(string)NV12" !
nvvidconv ! "video/x-raw(memory:NVMM),
width=(int)1920,height=(int)1080, format=(string)NV12" !
nv3dsink -e
```

Note: Replace **<n>** with the number of video device node, from which you need to stream.

Example 3: Record 2048p in H.264 format to a video file (HW accelerated).

Run the following command to record 2048p video in H.264 format to a video file.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> sensor-
mode=<m> ! "video/x-raw(memory:NVMM),
width=(int)2432,height=(int)2048, format=(string)NV12" !
nvv4l2h264enc ! avimux ! queue ! filesink
location=<filename>.mkv
```

Note:

- Replace **<n>** with the number of video device node, from which you need to stream.
- Replace **<m>** with the sensor mode in which you need to stream. Here 'm' indicates supported sensor modes i.e., 0 to 7. For 2048p, 'm' is either 0 or 1.
- Change **<filename>** with the name under which the video is recorded.

Example 4: Playback of saved 2048p video file in H.264 format (HW accelerated).

Run the following command to playback the 2048p saved video.

```
$ gst-launch-1.0 filesrc location=<filename>.mkv !
avidemux ! h264parse ! nvv4l2decoder ! nv3dsink
```

Note: Change **<filename>** with the name under which the video is recorded.

Example 5: Record 2048p in H.265 format to a video file (HW accelerated).

Run the following command to record 2048p video in H.264 format to a video file.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> sensor-
mode=<m> ! "video/x-
raw(memory:NVMM),width=(int)2432,height=(int)2048,
format=(string)NV12" ! nvv4l2h265enc qp-
range=20,20:20,20:-1,-1 ! h265parse ! matroskamux ! queue
! filesink location=<filename>.mkv
```

Note:

- Replace **<n>** with the number of video device node, from which you need to stream.
- Replace **<m>** with the sensor mode in which you need to stream. Here 'm' indicates supported sensor modes i.e., 0 to 7. For 2048p, 'm' is either 0 or 1.
- Change **<filename>** with the name under which the video is recorded.

Example 6: Playback of saved 2048p video file in H.265 format (HW accelerated).

Run the following command to playback the 2048p saved video.

```
$ gst-launch-1.0 filesrc location=<filename.mkv> !
matroskademux ! h265parse ! nvv4l2decoder ! nv3dsink
```

Note: Change **<filename>** with the name under which the video is recorded.

Example 7: Network streaming H.264 encoded 2048p video using RTP over UDP (HW accelerated).

Run the following command to stream 2048p video captured by camera connected with the Jetson AGX Orin™ development kit.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> sensor-
mode=<m> ! "video/x-raw(memory:NVMM),
width=(int)2432,height=(int)2048, format=(string)NV12" !
nvvidconv ! "video/x-raw(memory:NVMM),
format=(string)NV12, width=(int)2432, height=(int)2048" !
nvv4l2h264enc insert-vui=1 ! h264parse ! rtph264pay
config-interval=1 ! udpsink host=<ip_address>
port=<port_no>
```

Note:

- Replace **<n>** with the number of video device node, from which you need to stream.
- Replace **<m>** with the sensor mode in which you need to stream. Here 'm' indicates supported sensor modes i.e., 0 to 7. For 2048p, 'm' is either 0 or 1.
- 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).
- Do not close the application in Jetson AGX Orin™ 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)\"Z0JAKJZUA8ARPyzUAAADAQAAAwABAAADADyPCIRq\\
,aM48gA\\=\\=\", payload=(int)96" ! rtph264depay !
h264parse ! decodebin ! videoconvert ! autovideosink
sync=false
```

Note: Replace **<port_no>** with the port number of the Client device (For example, 5000).

Example 8: Capturing 2048p still image.

Run the following command to capture 2048p still image.

```
$ gst-launch-1.0 nvarguscamerasrc sensor-id=<n> num-
buffers=1 sensor-mode=<m> ! "video/x-raw(memory:NVMM),
format=(string)NV12, width=(int)2432, height=(int)2048,
framerate=(fraction)<f>/1" ! nvjpegenc ! filesink
location=<filename>.jpg
```

Note:

- Replace **<n>** with the number of video device node, from which you need to stream.
- Replace **<m>** with the sensor mode in which you need to stream. Here ‘m’ indicates supported sensor modes i.e., 0 to 7. For 2048p, ‘m’ is either 0 or 1.
- Replace **<f>** with the frame rate supported for the mentioned sensor mode, please refer Table2.
- 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, face issues in video quality and frame rate.

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-CAM56_CUOAGX.

- *e-CAM56_CUOAGX Datasheet*
- *e-CAM56_CUOAGX Release Package Manifest*

Glossary

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

IP: Internet Protocol.

RTP: Real-time Transport Protocol.

UDP: User Datagram Protocol.

Support

Contact Us

If you need any support on e-CAM56_CUOAGX 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	22-Aug-2023	Initial Draft	Camera Dev Team
1.1	20-Oct-2023	Updated to L4T35.4.1 and resolution to 2432x2048	Camera Dev Team
1.2	31-Oct-2023	Updated output messages for camera initialization	Camera Dev Team
1.3	16-Nov-2023	Added examples to record a video with h265 format in mkv file and playback the same	Camera Dev Team
1.4	26-Mar-2024	Updated product image in home page. Updated examples by adding framerate to the pipeline. Also, added a table with supported resolutions and framerates.	Camera Dev Team