



Teledyne DALSA • 880 Rue McCaffrey • St-Laurent, Québec, H4T 2C7 • Canada

<http://www.teledynedalsa.com/imaging/>

GVF-AN002-V1: GigE-V Framework for Linux App Note

Using gstreamer and ROS with GigE-V Framework for Linux

For Teledyne DALSA GigE-Vision cameras

Overview

ROS (The Robot Operating System) is a flexible framework for writing robot software. It provides an extensible set of capabilities for interfacing sensors, actuators, and all things robotic together in a distributed, collaborative environment for building robotic applications.

The addition of video images to a ROS application is provided by a video/imaging plugin for ROS. One such plugin, gscam, utilizes the configurable video streaming mechanism provided by the gstreamer multimedia framework.

Gstreamer is a library for constructing graphs of media-handling components into a pipeline that streams video (and/or audio) information from a source, through run-time configurable processing stages, to a destination (sink) for display, storage, or some transport mechanism for further use.

Utilizing Teledyne DALSA's GigeVision cameras in ROS can be accomplished by using Teledyne DALSA's **tdgigevsrc**, a video source plugin for gstreamer-1.0, which uses our **GigE-V Framework for Linux** software SDK to provide images to ROS via a gstreamer-1.0 pipeline into the gscam plugin.

Quick Start

Teledyne DALSA's GigeVision cameras can be used right away with ROS, provided everything is installed correctly. Proper set up can be quickly checked using the following steps.

Check that ROS is installed :

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ rosversion -d  
melodic  
dalsa@dalsa-1804:~$
```

If the response is "melodic", or "kinetic"(*) then a suitable ROS environment is installed. If the response is "noetic", it is not yet supported. If an error occurs indicating ROS is not installed, please skip to the "Getting Set Up – Installing ROS" section.

(*) Note : ROS Kinetic requires that the gscam driver be rebuilt to support gstreamer-1.0 (rather than the older gstreamer-0.1 version). There are instructions in the "Getting Set Up – Installing ROS" section.

Check that gstreamer-1.0 packages are installed :

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ gst-inspect-1.0 --gst-version  
GStreamer Core Library version 1.14.5  
dalsa@dalsa-1804:~$ pkg-config --variable=plugindir gstreamer-1.0  
/usr/lib/x86_64-linux-gnu/gstreamer-1.0  
dalsa@dalsa-1804:~$ ls -l /usr/include/gstreamer-1.0/gst/video/video.h  
-rw-r--r-- 1 root root 6415 Jul  3 2019 /usr/include/gstreamer-1.0/gst/video/video.h  
dalsa@dalsa-1804:~$
```

If the pkg-config command has an error or the "video.h" file is not found, please skip to the "Getting Set Up – Installing gstreamer-1.0" section.

Check that Teledyne DALSA's gstreamer plugin, "tdgigevsrc", is installed.

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ gst-inspect-1.0 | grep tdgigevsrc  
tdgigevsrc: tdgigevsrc: Teledyne DALSA GigeVsrc  
dalsa@dalsa-1804:~$
```

If tdgigevsrc is not found, please skip to the "Getting Set Up – Installing tdgigevsrc" section.

If it all **checks** out, follow these steps to display an image from a GigeVision camera in the ROS "image_view" plugin.

Start ROS using roscore:

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ roscore &  
[1] 3172  
dalsa@dalsa-1804:~$ ... logging to /home/dalsa/.ros/log/84d807b4-1fa3-11eb-a62c-94de80a47ea6/ro  
slaunch-dalsa-1804-3172.log  
Checking log directory for disk usage. This may take a while.  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://dalsa-1804:41843/  
ros_comm version 1.14.9  
  
SUMMARY  
=====  
  
PARAMETERS  
* /rostdistro: melodic  
* /rosversion: 1.14.9  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [3183]  
ROS_MASTER_URI=http://dalsa-1804:11311/  
  
setting /run_id to 84d807b4-1fa3-11eb-a62c-94de80a47ea6  
process[rosout-1]: started with pid [3194]  
started core service [/rosout]  
dalsa@dalsa-1804:~$
```

Start the gscam plugin (setting the GSCAM_CONFIG environment variable to indicate a mono image):

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ export GSCAM_CONFIG="tdgigevsrc bayer-as-mono=1 ! videoconvert "  
dalsa@dalsa-1804:~$ rosrn gscam gscam  
[ INFO] [1604607555.798989017]: Using gstreamer config from env: "tdgigevsrc bayer-as-mono=1 ! videoconvert "  
[ INFO] [1604607555.801338830]: using default calibration URL  
[ INFO] [1604607555.801389484]: camera calibration URL: file:///home/dalsa/.ros/camera_info/camera.yaml  
[ INFO] [1604607555.801436713]: Unable to open camera calibration file [/home/dalsa/.ros/camera_info/camera.yaml]  
[ WARN] [1604607555.801462214]: Camera calibration file /home/dalsa/.ros/camera_info/camera.yaml not found.  
[ INFO] [1604607555.801472653]: Loaded camera calibration from  
[ INFO] [1604607556.830623666]: Time offset: 1604606067.276  
[ INFO] [1604607556.965897474]: Publishing stream...  
[ INFO] [1604607556.966000012]: Started stream.
```

Check that images are available, in another terminal window check for ROS "topics" (aka published items available for use):

```
dalsa@dalsa-1804:~$ rostopic list  
/camera/camera_info  
/camera/image_raw  
/camera/image_raw/compressed  
/camera/image_raw/compressed/parameter_descriptions  
/camera/image_raw/compressed/parameter_updates  
/camera/image_raw/compressedDepth  
/camera/image_raw/compressedDepth/parameter_descriptions  
/camera/image_raw/compressedDepth/parameter_updates  
/camera/image_raw/theora  
/camera/image_raw/theora/parameter_descriptions  
/camera/image_raw/theora/parameter_updates  
/rosout  
/rosout_agg  
dalsa@dalsa-1804:~$
```

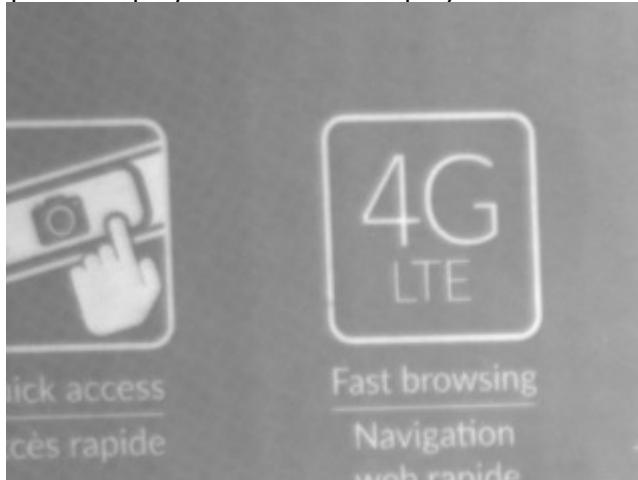
In our case, the "/camera/image_raw" is the item to use.

(Note: If the GigE-V Framework for Linux is not installed – no cameras will be detected and no images will be available).

To view an image with the ROS "image_view" plugin/driver:

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ rosrun image_view image_view image:=/camera/image_raw  
[ INFO] [1604612625.254553901]: Initializing nodelet with 4 worker threads.  
[ INFO] [1604612625.336795712]: Using transport "raw"
```

It will open a display window and display live video:



Color images (via bayer decoding) can be obtained as well. Changing the GSCAM_CONFIG environment variable and restarting the gscam driver gives a

```
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$  
dalsa@dalsa-1804:~$ export GSCAM_CONFIG="tdgigevsrsrc ! bayer2rgb ! videoconvert "  
dalsa@dalsa-1804:~$ rosrun gscam gscam  
[ INFO] [1604610542.361361244]: Using gstreamer config from env: "tdgigevsrsrc ! bayer2rgb ! videoconvert "  
[ INFO] [1604610542.363599133]: using default calibration URL  
[ INFO] [1604610542.363654463]: camera calibration URL: file:///home/dalsa/.ros/camera_info/camera.yaml  
[ INFO] [1604610542.363723703]: Unable to open camera calibration file [/home/dalsa/.ros/camera_info/camera.yaml]  
[ WARN] [1604610542.363739576]: Camera calibration file /home/dalsa/.ros/camera_info/camera.yaml not found.  
[ INFO] [1604610542.363760632]: Loaded camera calibration from  
[ INFO] [1604610543.394179707]: Time offset: 1604606067.276  
[ INFO] [1604610543.528142368]: Publishing stream...  
[ INFO] [1604610543.528236061]: Started stream.
```

bayer decoded / color image:



The ROS gscam driver will provide images on the "/camera/image_raw" topic to any other ROS object that a developer wants to use.

Getting Set Up

Issues encountered in the "Quick Start" (above) can be addressed in this section. The installation of ROS, the possible re-configuration of ROS, the installation of gstreamer-1.0, and the installation of the tdgigevsrc plugin are presented here.

Getting Set Up – Installing ROS

As of this writing, there are 3 version of ROS in use.

- 1) Kinetic Kame (typically associated with Ubuntu 16.04 LTS)
- 2) Melodic Morenia (typically associated with Ubuntu 18.04 LTS)
- 3) Noetic Ninjemys (typically associated with Ubuntu 20.04 LTS)

The required gscam driver is not readily available for Noetic Ninjemys and needs to be re-compiled to support gstreamer-1.0 (as opposed to gstreamer-0.1) for Kinetic Kame. This recompile will be discussed following the information on installing for Melodic Morenia (and Ubuntu 18.04).

The site : "<http://wiki.ros.org/melodic/Installation/Ubuntu>" contains really well done instructions for installing ROS on Ubuntu 18.04. Copy and paste the instructions into a shell window and, following a remarkably large download, ROS will be installed.

The list of instructions is repeated here (for reference):

```
> sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
/etc/apt/sources.list.d/ros-latest.list'

> sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

> sudo apt update

> sudo apt install ros-melodic-desktop-full

> echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

> source ~/.bashrc
```

To use **Kinetic Kame** :

The gscam driver in Kinetic Kame is built for use with the older gstreamer-0.1 version. To use tdgigevsrc, it must be re-built for gstreamer-1.0 as follows:

Following installation of Kinetic Kame (similar to Melodic shown above) on Ubuntu 16.04, get the catkin build tools and create a catkin workspace:

```
sudo apt-get install python-catkin-tools
mkdir -p catkin_ws/src
catkin_make
cd catkin_ws/src
```

Get the gscam source code (from github) :

```
git clone https://github.com/ros-drivers/gscam
```

Build it for gstreamer-1.0 and install it into the catkin workspace tree overlaid onto the Kinetic file tree:

```
cd ..
catkin_make -DGSTREAMER_VERSION_1_X=On -DCMAKE_INSTALL_PREFIX=/opt/ros/kinetic
catkin_make install
source install/setup.bash
```

Kinetic should now function the same as Melodic did.

More detailed information on ROS set up can be found at:

<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment>

(Note: Kinetic, as with Ubuntu 16.04, goes End-Of-Life in April 2021)

For **Noetic Ninjemys**:

While an official solution for Noetic Ninjemys is pending, please check the ROS wiki page at <http://wiki.ros.org/gscam> for news on updated official support for gscam.

Check also on github :

<https://github.com/ros-drivers/gscam>

or

<https://github.com/hap1961/gscam/tree/noetic-devel>)

Getting Set Up – Installing gstreamer-1.0

For full information on how to install the packages required for building against gstreamer-1.0 refer to :

<https://gstreamer.freedesktop.org/documentation/installing/on-linux.html?gi-language=c>

The minimum required steps are :

```
sudo apt-get install libgstreamer1.0-0 gstreamer1.0-plugins-base
                        gstreamer1.0-plugins-good gstreamer1.0-plugins-bad
                        gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-doc
                        gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-gl
                        gstreamer1.0-gtk3 gstreamer1.0-qt5
```

and

```
sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
```

Getting Set Up – Installing tdgigevsrc

Teledyne DALSA's tdgigevsrc plugin for gstreamer-1.0 will detect and use the currently installed version of the **GigE-V Framework for Linux**. If the GigE-V Framework for Linux is not installed, tdgigevsrc will still install but will not be able to provide images from cameras.

Refer the to users manual for **Teledyne DALSA's GigE-V Framework for Linux** for information on installing it.

The tdgigevsrc plugin is available as a self-extracting installation script. To use it, copy it to your home directory and do the following:

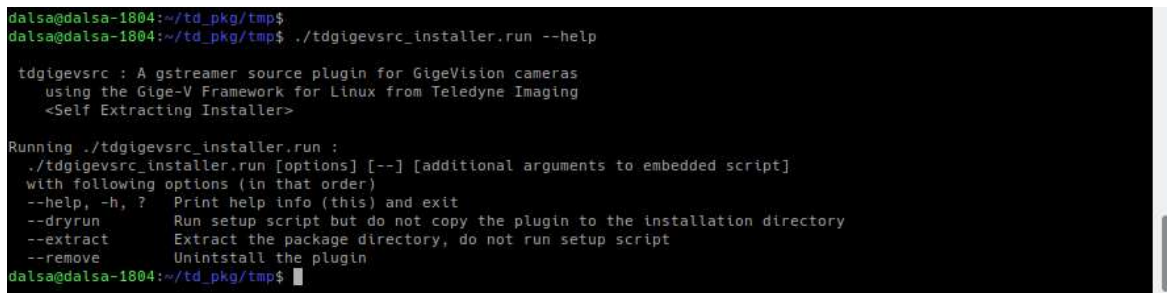
```
chmod u+x ./tdgigevsrc_installer.run
```

Install it simply using

```
./tdgigevsrc_installer.run
```

If it does not install correctly, follow the help to instructions to extract it and compile it manually to see if the problem can be fixed. Help is provided by running it as

```
./tdgigevsrc_installer.run -help
```



```
dalsa@dalsa-1804:~/td_pkg/tmp$
dalsa@dalsa-1804:~/td_pkg/tmp$ ./tdgigevsrc_installer.run --help

tdgigevsrc : A gstreamer source plugin for GigeVision cameras
              using the Gige-V Framework for Linux from Teledyne Imaging
              <Self Extracting Installer>

Running ./tdgigevsrc_installer.run :
./tdgigevsrc_installer.run [options] [--] [additional arguments to embedded script]
with following options (in that order)
--help, -h, ?   Print help info (this) and exit
--dryrun        Run setup script but do not copy the plugin to the installation directory
--extract       Extract the package directory, do not run setup script
--remove        Uninstall the plugin
dalsa@dalsa-1804:~/td_pkg/tmp$
```

Using **tdgigevsrc**:

Issuing the command "gst-inspect-1.0 tdgigevsrc" will output a detailed summary of the tdgigevsrc plugin and the parameters that can be passed to it.

These parameters are summarized here:

cam-index:	The index of the camera to select (as reported by lsgev)
cam-sn:	The serial number (as a string) of the camera to select.
cam-ip:	The ip address of the camera to select (as a string a:b:c:d)
cam-name:	The "DeviceName" string (feature) of the camera to select
cfg-file:	A file of "FeatureName Value" pairs (1 per line) to apply to the camera.
bayer-as-mono:	Provide bayer pixel formats as Mono (GRAY8/GRAY16) video.
nfrm:	Number of internal frames to buffer in the GigE-V Framework.

Examples of strings for the GSCAM_CONFIG environment variable are:

To output native images (either Mono or RGB color):

```
"tdgigevsrc bayer-as-mono=1 ! videoconvert "
```

Outputs the default image format from camera index 0 and converts it to whatever the ROS gscam driver requests it to be. (Images from a Bayer format camera will be treated as Monochrome).

```
"tdgigevsrc cam-index=1 bayer-as-mono=1 ! videoconvert "
```

Same as above but from the camera with index #1 (reported by the lsgev tool)

```
"tdgigevsrc cam-ip=169:254:10:2 bayer-as-mono=1 ! videoconvert "
```

Same as above but from the camera with ip address 169:254:10:2 (reported by lsgev)

To output color images from a camera outputting Bayer format pixels:

```
"tdgigevsrc ! bayer2rgb ! videoconvert "
```

Outputs the default image format from camera index 0 and converts it to RGB for deliver to the ROS gscam driver)

```
"tdgigevsrc cam-index=1 ! bayer2rgb ! videoconvert "
```

Same as above but from the camera with index #1 (reported by the lsgev tool)

```
"tdgigevsrc cam-ip=169:254:10:2 ! bayer2rgb ! videoconvert "
```

Same as above but from the camera with ip address 169:254:10:2 (reported by lsgev)

To change the camera settings before outputting images, a cfg-file can be specified. A text file named *camera.txt* containing :

```
Width 1024
Height 512
```

will change the camera output to 1024x512 and the line

```
"tdgigevsrc cfg-file=camera.txt ! bayer2rgb ! videoconvert "
```

will provide that file to the tdgigevsrc plugin and output a 1024x512 RGB image.

(Note : gstreamer requires that the "stride" (number of bytes per image line) be a multiple of 4. The tdgigevsrc plugin tries to handle this but will fail if an incorrect image width is passed to it.)

The **tdgigevsrc** gstreamer plugin package for streaming from GigeVision cameras works with GigE0V Framework for Linux versions 2.10 and higher and can be downloaded from the device-drivers section of the Teledyne DALSA web site:
<https://www.teledynedalsa.com/en/support/downloads-center/device-drivers/>

Teledyne DALSA's **GigE-V Framework for Linux SDK** and can be downloaded from the Teledyne DALSA web site:
<https://www.teledynedalsa.com/imaging/support/downloads/sdks/>