How to stream camera over network

How to stream camera over network

Stable: 10.04.2020 - 15:50 / Revision: 10.04.2020 - 15:48

1 Overview

This article will explain how to stream camera content over network thanks to GStreamer application on top of V4L2 Linux® kernel framework.

Capturing compressed JPEG pictures is an efficient way to send camera images to any local or remote player; JPEG pictures require a limited bandwidth while being fully interoperable.

Find below some examples of command lines allowing to capture a continuous JPEG stream while playing it using various multimedia players, either local or remote.

2 Local streaming

Here is an example of a local preview involving V4l2-ctl for JPEG pictures capture and gst-play GStreamer player for JPEG decoding and display. JPEG pictures are sent over a standard Linux pipe.

```
Board $> v4l2-ctl --set-parm=30;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=-1 --stream-to=- 2>/dev/null | gst-play-
1.0 "fd://0"
```

--stream-to=- tells V4l2-ctl to output binary captured content to standard output, which is then sent to pipe |.

Special URI fd://0 tells gst-play GStreamer player to read data from the pipe.

Note the 2>/dev/null right after the V4l2-ctl command to remove the logs from console output.

3 UDP streaming

An internet connection is required, for example by plugging an ethernet cable on the STM32MP157C-EV1 Evaluation board CN3 ethernet connector

Get first the IP address aa.bb.cc.dd of the host PC using ifconfig command:

```
PC $> ifconfig | grep "inet addr"
   inet addr:aa.bb.cc.dd  Bcast:10.201.23.255  Mask:255.255.252.0
   inet addr:127.0.0.1  Mask:255.0.0.0
```

Then fill the host= udpsink property with this IP address on the remote side:

```
Board $> v4l2-ctl --set-parm=30;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=-1 --stream-to=- 2>/dev/null | gst-
launch-1.0 fdsrc ! jpegparse ! rtpjpegpay ! udpsink host=aa.bb.cc.dd port=5000
```

Then play the UDP stream on host PC:

```
PC $> gst-launch-1.0 udpsrc port=5000 ! application/x-rtp, encoding-name=JPEG !
rtpjpegdepay ! jpegparse ! decodebin ! autovideosink
```

A new window will popup on host PC displaying the camera content.

Due to SDP protocol signaling, this solution is not fully interoperable because it needs a dedicated GStreamer command line to be played on host side