**Gstreamer getting started**

Before we jump to embedded linux , lets learn some commands , plugins in Ubuntu

<https://gstreamer.freedesktop.org/documentation/installing/on-linux.html>

Install the Gstreamer on Ubuntu

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

Clone the examples

git clone git://anongit.freedesktop.org/gstreamer/gst-docs

Compile and run the example:

gcc basic-tutorial-1.c -o basic-tutorial-1 `pkg-config --cflags --libs gstreamer-1.0`

./basic-tutorial-1

It should open a windows to play the file directly from the internet.

The example are in C.

Gstreamer can run from C code , compile and run

Gstreamer can use gst-launch-1.0

To build and run Gstreamer pipelines

## Description

gst-launch-1.0 is a tool that builds and runs basic GStreamer pipelines.

In its simplest form, a PIPELINE-DESCRIPTION is a list of elements separated by exclamation marks (!). Properties may be appended to elements in the form property=value.

You can download some sample files from

<http://file-examples.com/index.php/sample-video-files/sample-mp4-files/>

Play the mp4 file using playbin plugin

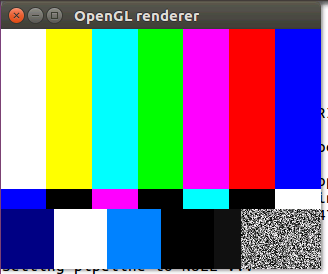
<https://gstreamer.freedesktop.org/data/doc/gstreamer/head/gst-plugins-base-plugins/html/gst-plugins-base-plugins-playbin.html>

gst-launch-1.0 playbin uri=file:/ name of file

http://www.einarsundgren.se/gstreamer-basic-real-time-streaming-tutorial/

Test source:

gst-launch-1.0 videotestsrc ! autovideosink



This command will inject the videtestsrc as we see above

Will encode it than decode it , then show it on screen

gst-launch-1.0 videotestsrc ! x264enc bitrate=256 ! decodebin ! autovideosinc

This will encode the testvideosrce and save it to file 1.mp4

gst-launch-1.0 videotestsrc ! x264enc bitrate=256 ! filesink location=1.mp4

this will play the fila that we just encode

gst-lauch-1.0 playbin uri=file:/home/elia/1.mp4

v4l2 is a video for linux driver where it uses the camera device

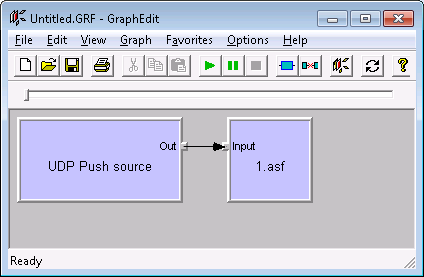
gst-launch v4l2src device=/dev/video0 ! x264enc bitrate=256 ! udpsink host=192.168.21.106 port=8064

We can also send the videotestsrc if we don’t have camera.

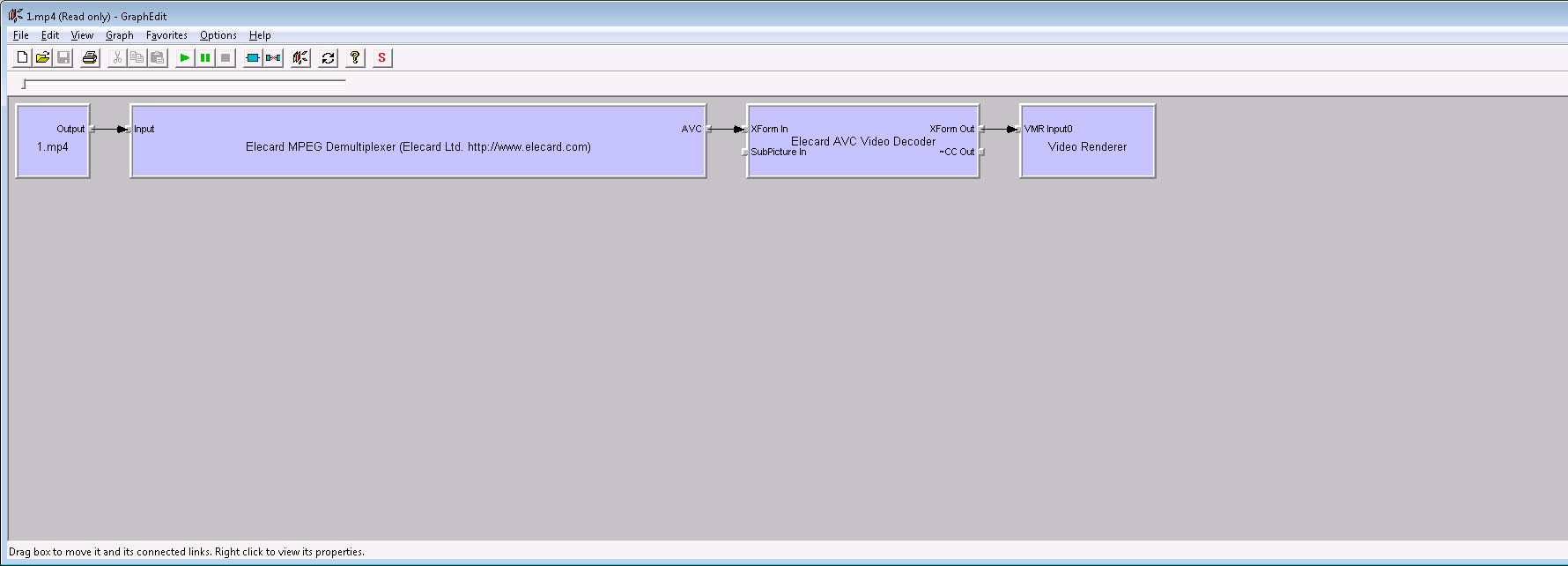
Sending the videotestsrc without compressing it to H264 will result with error that message is too big.

Now it time to write a direct show filter in windows to see if we can play the file

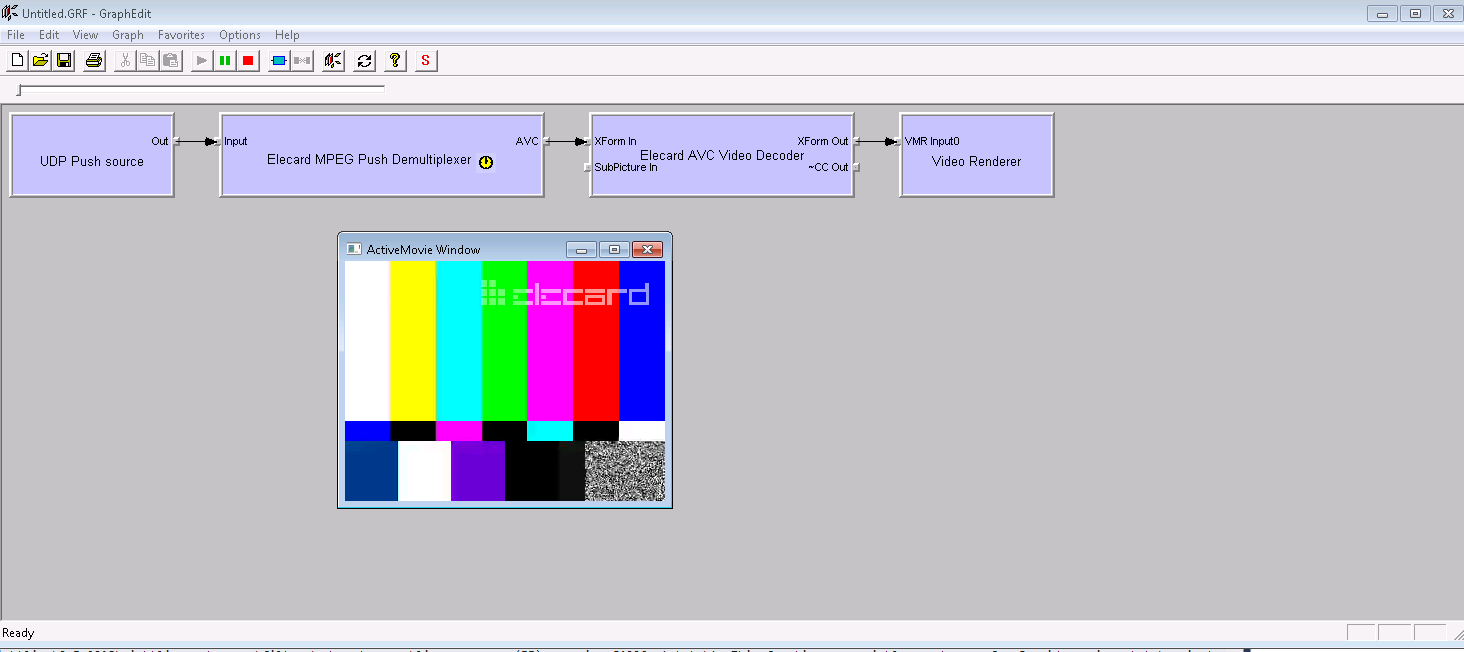
The UDP will now connect to dump to save the mp4 file that being broadcast.



The player for the file using Elecard:



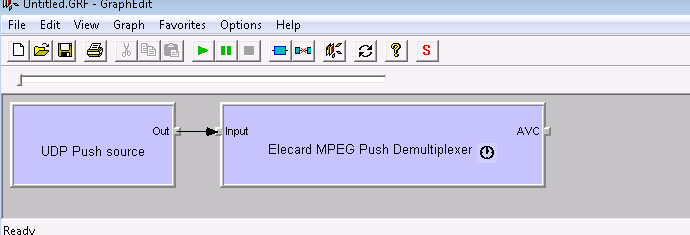
And final , the direct show is now playing live stream from Gstreamer Linux Ubuntu.



How to connect:

Elecard is very smart filter

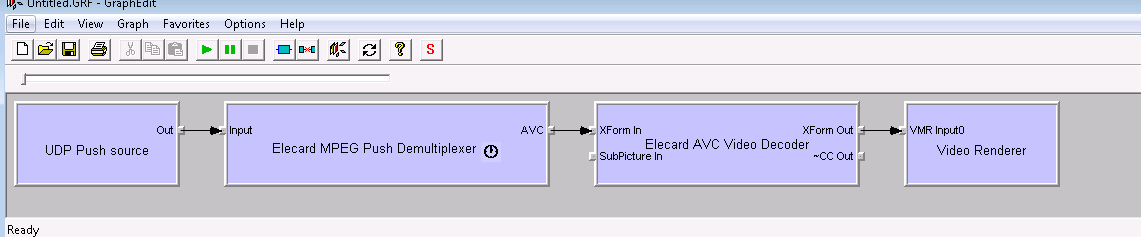
Connect the Elecard MPEG Push Demuliplexer filer



The UDP is always streaming , even if we did not push the run button.

After few seconds, the AVC pin will appear , elecard demux will detect it.

Now you can render elecard AVC pin , before pressing run ofcourse



# Secure Real-time Transport Protocol

SRTP

<http://read.pudn.com/downloads91/sourcecode/windows/comm/349336/srtp-1.0.6/doc/libsrtp.pdf>

<https://github.com/cisco/libsrtp>

# [GStreamer STRP for full HD H.264 video](https://raspberrypi.stackexchange.com/questions/68252/gstreamer-strp-for-full-hd-h-264-video)

https://raspberrypi.stackexchange.com/questions/68252/gstreamer-strp-for-full-hd-h-264-video?rq=1