GStreamer Buster Version 1.14.4 on RaspberryPi 4

Installation :

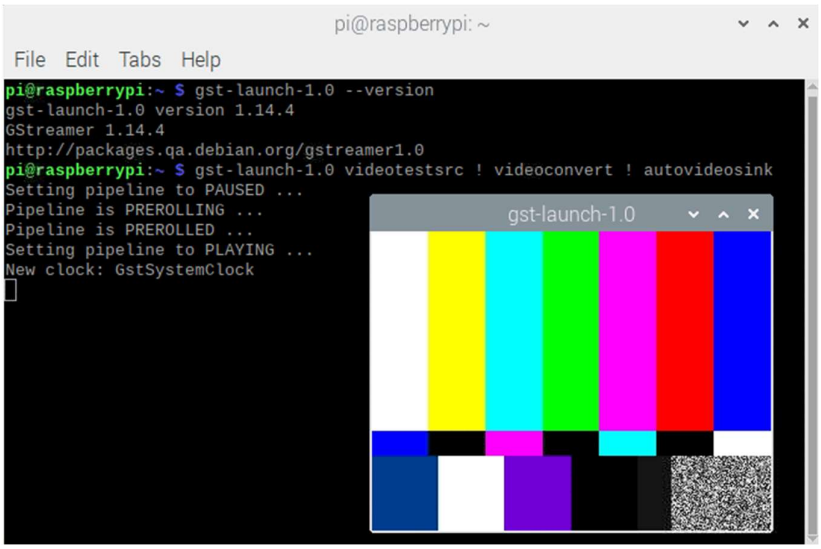
A few of the packages required for the gstreamer are already installed raspberry pi, we need to install some additional plugins

```
# install a missing dependency
$ sudo apt-get install libx264-dev libjpeg-dev
# install the remaining plugins
$ sudo apt-get install libgstreamer1.0-dev \
    libgstreamer-plugins-base1.0-dev                                    \

    libgstreamer-plugins-bad1.0-dev \
    gstreamer1.0-plugins-ugly \
    gstreamer1.0-tools
# install some optional plugins
$ sudo apt-get install gstreamer1.0-gl gstreamer1.0-gtk3
# if you have Qt5 install this plugin
$ sudo apt-get install gstreamer1.0-qt5
# install if you want to work with audio
$ sudo apt-get install gstreamer1.0-pulseaudio
```

Testing Installation :

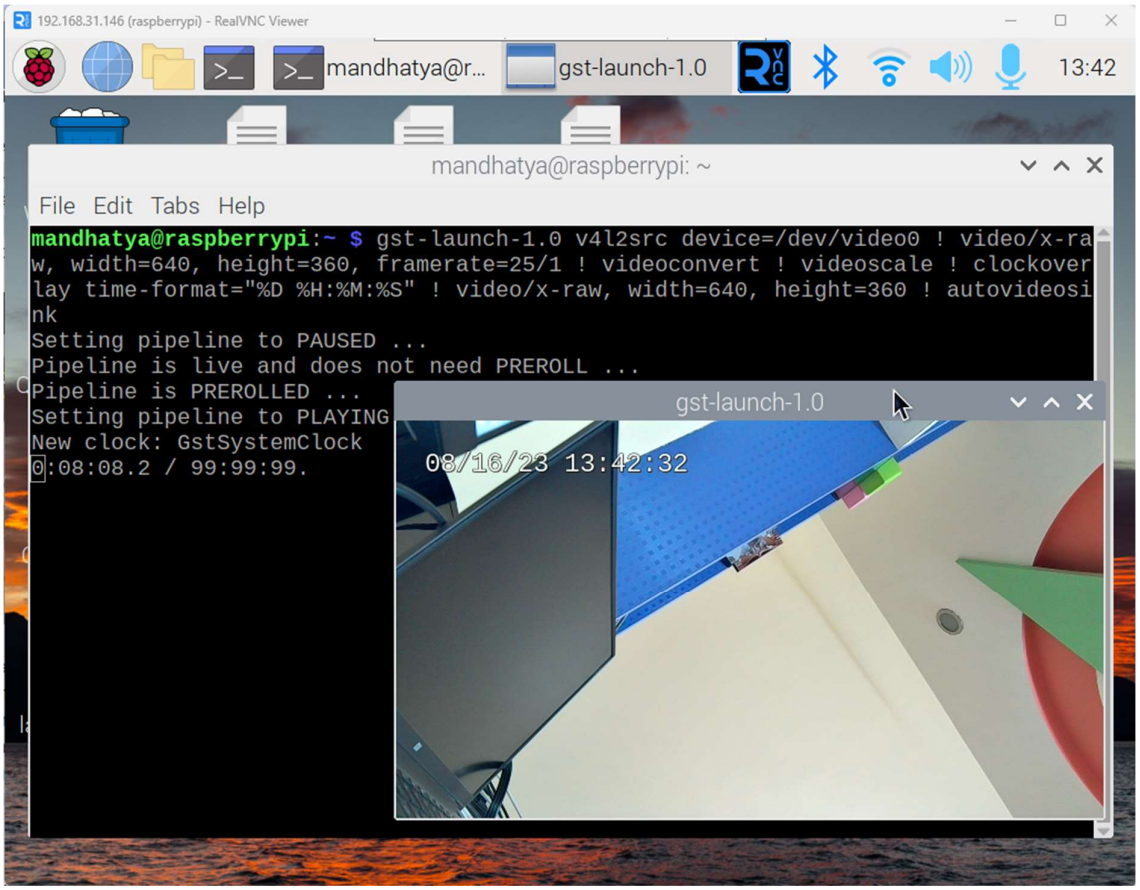
```
$ gst-launch-1.0 videotestsrc ! videoconvert ! autovideosink
```



Testing Streaming with Camera :

```
$ gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw, width=1280, height=720, framerate=30/1 ! videoconvert ! videoscale ! clockoverlay time-format="%D %H:%M:%S" ! video/x-raw, width=640, height=360 ! autovideosink
```

(Follow the instructions below the picture before executing the above command to use the correct configuration available for your camera)



V4l2src device=/dev/video0 is used to capture the video from the camera

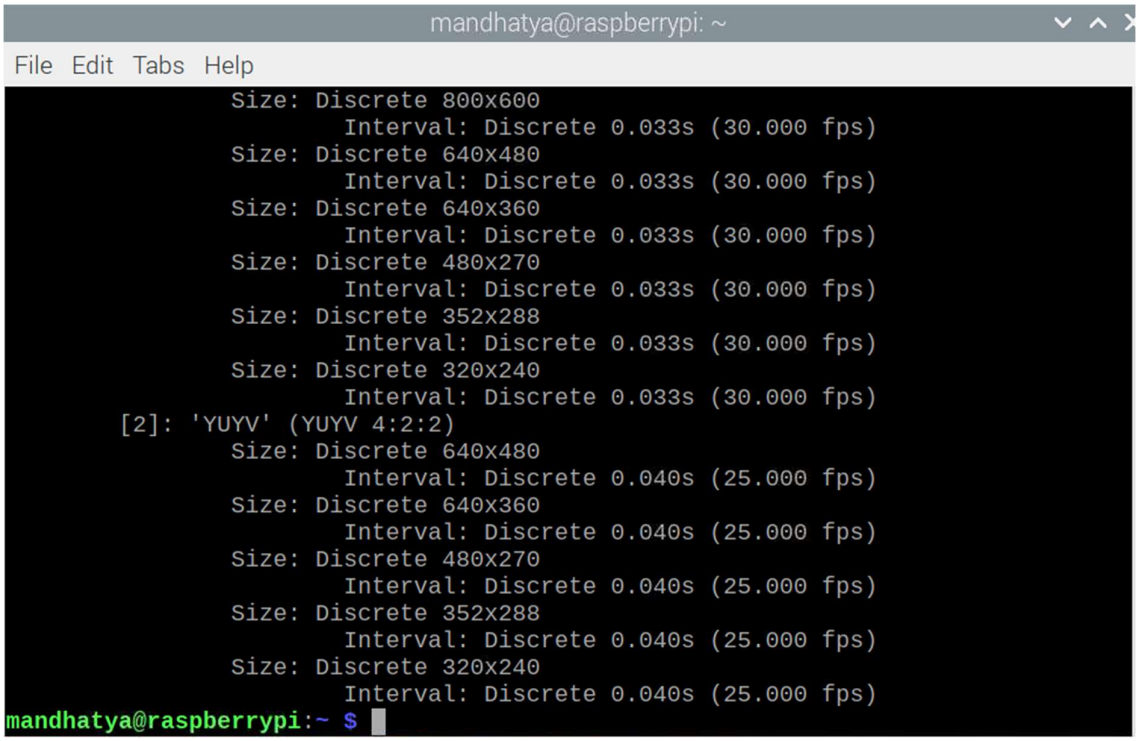
- If you are using raspi camera(cli) then it is video0
- If there is no raspi on the raspberry pi and you are using a usb camera, even then it's video0

If there is raspi camera and there is also a usb camera connected, if you want to stream through raspi its video0, if usb camera its video1.

The width and height are your resolution, the width, height and your framerate are dependent on your camera, to check your camera specifications type :

```
$ v4l2-ctl --device=/dev/video0 -D --list-formats-ext
```

Use only YUVY Specification



UDP Streaming:

We use two Raspberry Pis, both connected to the same home network. However, it could just as easily be an RPi and a laptop on the other side of the world. You need to know the address of the receiving Raspberry Pi on forehand.

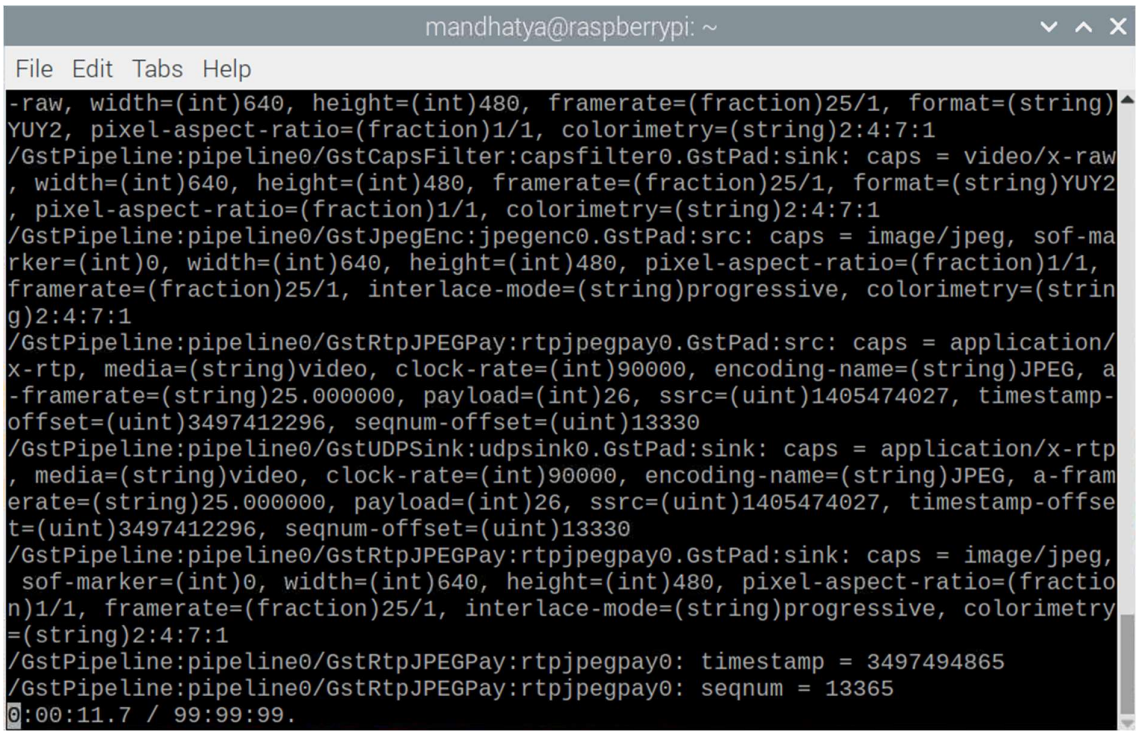
Raspberry Pi 32 or 64-bit OS

```
# get the IP address of the recieving RPi first
$ hostname -I
# start the sender, the one with the camera
$ gst-launch-1.0 -v v4l2src device=/dev/video0 num-buffers=-1 ! video/x-raw, width=640, height=480, framerate=30/1 ! videoconvert ! jpegenc ! rtpjpegpay ! udpsink host=192.168.31.146 port=5200
```

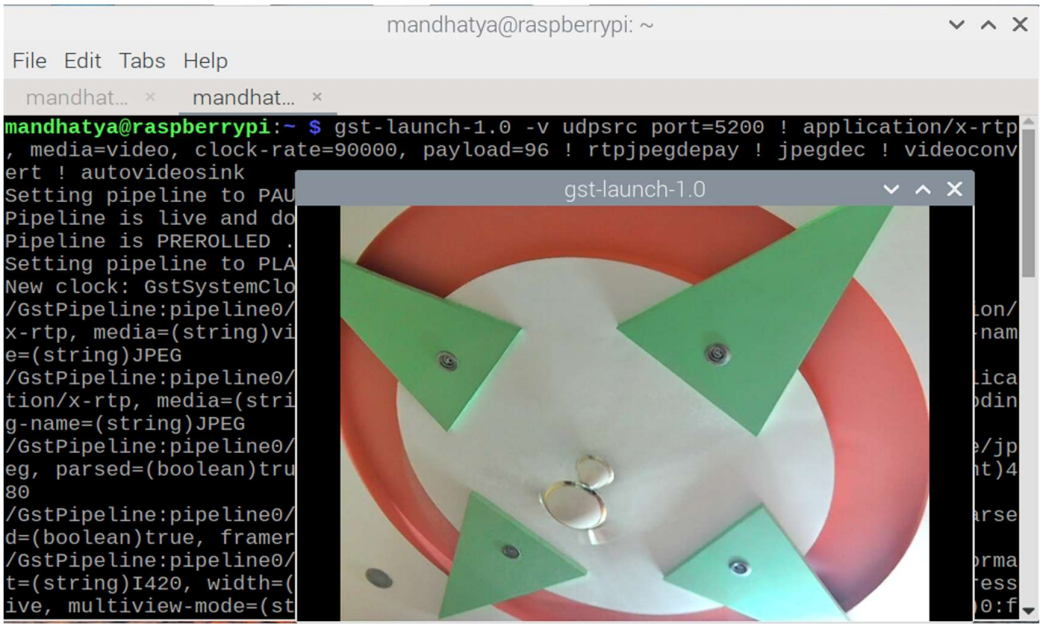
start the reciever, the one with IP 192.168.31.146

$ gst-launch-1.0 -v udpsrc port=5200 ! application/x-rtp, media=video, clock-rate=90000, payload=96 ! rtpjpegdepay ! jpegdec ! videoconvert ! autovideosink

Sender



Receiver



TCP Streaming:

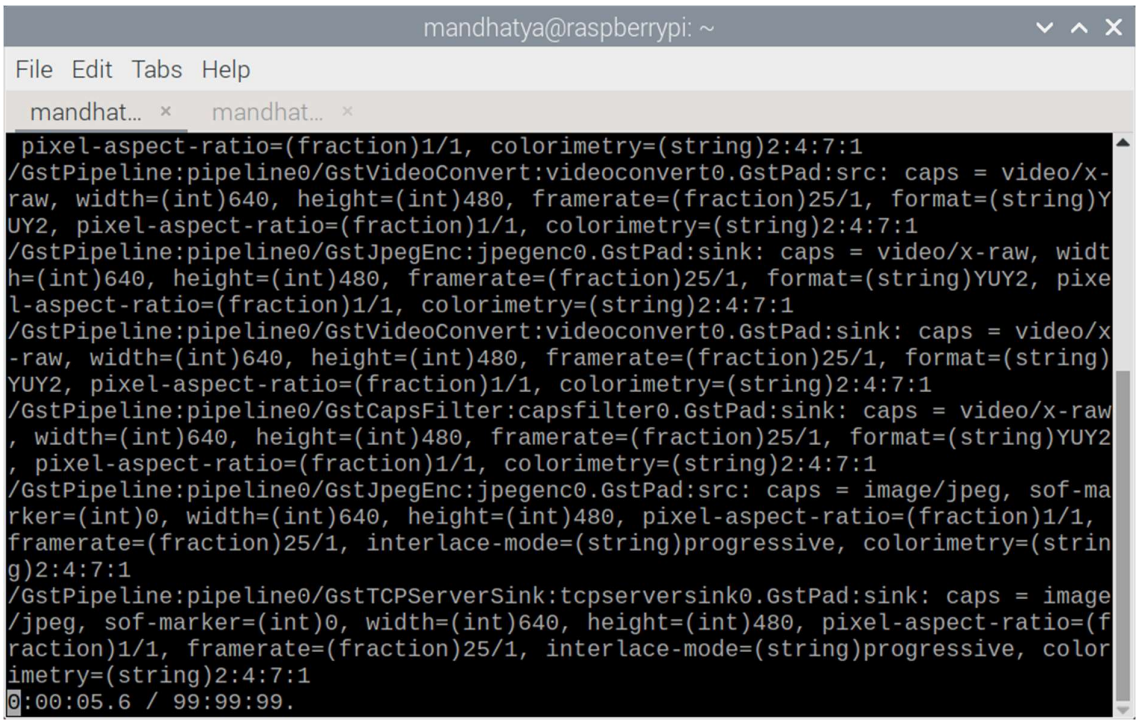
The other method of streaming is with TCP. The difference with UDP is the latency. UDP is faster.

The commands as listed below. Note the different IP addresses. With TCP streaming, you use the server address, the sender, instead of the receiver, as we saw with the UDP streaming.

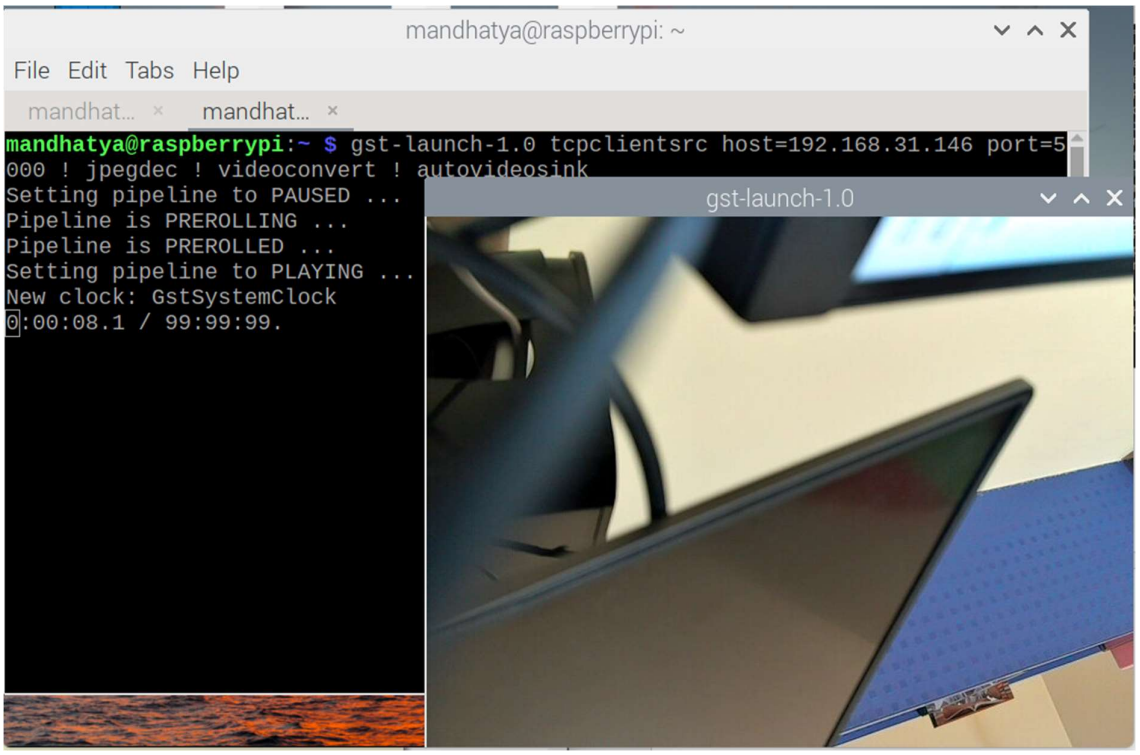
Raspberry Pi 32 or 64-bit OS

```
# get the IP address of the sending RPi first
$ hostname -I
# start the sender, the one with the Raspicam and IP 192.168.178.32
$ gst-launch-1.0 -v v4l2src device=/dev/video0 num-buffers=-1 ! video/x-raw,width=640,height=480, framerate=30/1 ! videoconvert ! jpegenc ! tcpserversink  host=192.168.31.146 port=5000
# start the reciever and connect to the server with IP 192.168.178.32
$ gst-launch-1.0 tcpclientsrc host=192.168.31.146 port=5000 ! jpegdec ! videoconvert ! autovideosink
```

Sender



Receiver



Reference Link :

https://qengineering.eu/install-gstreamer-1.18-on-raspberry-pi-4.html