**GStreamer**

**PI INSTALLATION**

Install GStreamer on Pi with:

sudo apt-get install gstreamer1.0-tools

**MAC OS CLIENT INSTALLATION**

On your receiving end, you will also need GStreamer. To install it for Mac first update HomeBrew:

ruby -e "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install>)"

Now we need to grab GStreamer:

brew install GStreamer gst-libav gst-plugins-ugly gst-plugins-base gst-plugins-bad gst-plugins-good

*This will take little time*

**RUNNING ON BOTH ENDS (**tcpserversink**)**

**Now on the Pi enter:**

raspivid -t 999999 -h 720 -w 1080 -fps 25 -hf -b 2000000 -o - | gst-launch-1.0 -v fdsrc ! h264parse ! rtph264pay config-interval=1 pt=96 ! gdppay ! tcpserversink host=192.168.2.44 port=5000

**And on your local machine enter:**

gst-launch-1.0 -v tcpclientsrc host=192.168.2.44 port=5000 ! gdpdepay ! rtph264depay ! avdec\_h264 ! videoconvert ! autovideosink

**RUNNING ON BOTH ENDS (**udp, raspivid**)**

**local machine enter FIRST:**

gst-launch-1.0 -v udpsrc port=5000 ! application/x-rtp, clock-rate=90000,payload=96 ! rtph264depay ! decodebin3 ! videoconvert ! autovideosink

**And then on the Pi enter: (ip of client)**

raspivid -t 999999 -h 720 -w 1080 -fps 25 -hf -b 2000000 -o - | gst-launch-1.0 -v fdsrc ! h264parse ! rtph264pay pt=96 ! udpsink host=192.168.2.10 port=5000

**RUNNING ON BOTH ENDS file (**udp**)**

**Test video generation (resolution should be over 400x400!)**

gst-launch-1.0 -e videotestsrc ! video/x-raw, width=640, height=360 ! x264enc ! filesink location=test.ts

**or**

raspivid -t 999999 -h 800 -w 800 -fps 25 -hf -b 2000000 -o - | gst-launch-1.0 -e fdsrc ! filesink location=rvid.ts

**local machine enter FIRST:**

gst-launch-1.0 -v udpsrc port=5000 ! application/x-rtp,,encoding-name=H264,payload=96 ! rtph264depay ! decodebin3 ! videoconvert ! autovideosink

**And then on the Pi enter: (ip of client)**

gst-launch-1.0 -v filesrc location=long.ts ! h264parse ! rtph264pay pt=96 ! udpsink host=192.168.2.10 port=5000

**MAKE PICS**

raspistill -o png/%04d.png -w 400 -h 400 -t 50000 -e png -tl 0

**TIMELAPS ON LOCAL MACHINE**

gst-launch-1.0 -e multifilesrc location="png/%04d.png" index=0 caps="image/png,framerate=(fraction)10/2,width=160,height=160" ! pngdec ! videoconvert ! videoscale ! video/x-raw,width=160,height=160 ! autovideosink

**Save timelaps video encoded in h264**

gst-launch-1.0 -e multifilesrc location="png/%04d.png" index=0 caps="image/png,framerate=(fraction)5/2,width=160,height=160" ! pngdec ! videoconvert ! videoscale ! video/x-raw,width=160,height=160 ! x264enc ! filesink location=tlaps.ts

Can’t play this timelaps through network

Try play it local

**raspistill -w 160 -h 160 -t 500000 -e jpg -tl 100 -o - | gst-launch-1.0 -v fdsrc ! image/jpg,framerate=(fraction)10/2,width=160,height=160 ! jpegdec ! videoconvert ! videorate ! video/x-raw,width=160,height=160 ! jpegparse ! rtpjpegpay pt=96 ! udpsink host=192.168.2.10 port=5000**

raspistill -w 160 -h 160 -t 500000 -e png -tl 0 -o - | gst-launch-1.0 -v fdsrc caps="image/png,framerate=(fraction)10/2,width=160,height=160" ! pngdec ! videoconvert ! videorate ! videoscale ! video/x-raw,width=160,height=160 ! h264parse ! rtph264pay pt=96 ! udpsink host=192.168.2.10 port=5000

**Stream frames**

gst-launch-1.0 filesrc location=mypic.jpg ! jpegdec ! imagefreeze ! autovideosink

gst-launch-1.0 -e multifilesrc location="png/%04d.png" index=0 caps="image/png,width=100,height=100" ! pngdec ! videoconvert ! videoscale ! video/x-raw,width=160,height=160 ! x264enc ! mpegtsmux ! filesink location=test.ts

***From Pi, test video encoded in 264 and written to file***

gst-launch-1.0 -e videotestsrc ! video/x-raw, width=640, height=360 ! x264enc ! filesink location=test.ts

raspivid -t 999999 -h 720 -w 1080 -fps 25 -hf -b 2000000 -o - | gst-launch-1.0 -e fdsrc ! filesink location=rvid.ts

***On mac***

gst-launch-1.0 filesrc location=test.ts ! decodebin3 ! videoconvert ! autovideosink

On mac standalone encoder decoder test:

gst-launch-1.0 videotestsrc ! video/x-raw, width=640, height=360 ! x264enc ! decodebin3 ! videoconvert ! autovideosink