**Pi Camera Live Streaming**

This section teaches you how to start a pi camera stream over RTSP.

1. First, connect the monitor, keyboard, and mouse to raspberry pi. Power on everything.
2. Enable legacy camera using sudo raspi-config. Navigate to Interface options 🡪 Legacy camera. Reboot the Raspberry Pi.

A screenshot of a computer

Description automatically generated

1. Open VLC media player by clicking on the Raspberry Pi icon 🡪 Sound & Video 🡪 VLC Media Player.

A screenshot of a computer

Description automatically generated

1. Execute this command in a terminal to stream pi camera over RTSP.

**raspivid -o - -t 0 -w 640 -h 480 -fps 90 -n | cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8080/}' :demux=h264**

-w is for adjusting width. -h is for adjusting height. -fps is to adjust frame rate per second. After this command is executed, a RTSP URL will be generated in the form of rtsp://<IP\_Address>:8080/ where the IP address can be either eth0, wlan0, usb0 or wwan0.

# A computer screen with a sunset in the background Description automatically generated

1. Go back to VLC media player, click on Media 🡪 Open Network Stream. Enter the RTSP URL and click “Play”.

A computer screen with a message

Description automatically generated

1. You should see a live stream displayed inside the VLC media player.
2. Next, Dial up 5G internet access to both Raspberry Pi and Windows laptop.
3. Run pi camera streaming command again. On your laptop, open the registration.py code in Lab6, replace this following line with the generated RTSP URL.

cap = VideoStream(0).start() 🡪 cap = VideoStream(rtsp://<IP\_Address>:8080/).start()

For the IP address, use the public IP address of wwan0. Then, run the script. If it does not run, it means that the public IP address is wrong, or you did not start mbim network to obtain the public IP address of wwan0.

Do note that this is just one way of streaming the pi camera, you can try to use a simple RTSP servers or even stream it to a webpage to get better quality and lower latency streams.