

## Raspberry pi diffuser un flux vidéo avec le serveur VLC

Vous trouverez ci-dessous des informations sur l'utilisation de VLC en tant que serveur pour diffuser un flux vidéo.

### 1 installation de vlc

```
pi@raspberrypi3:~ $ sudo apt-get install vlc
```

affichage de la version installée (Aout 2018)

```
pi@raspberrypi3:~ $ cvlc --version
VLC media player 2.2.7 Umbrella (revision 2.2.7-0-g6e32381286)
VLC version 2.2.7 Umbrella (2.2.7-0-g6e32381286)
```

### 2 Test du fonctionnement de la caméra

En dehors des commandes Linux habituelles qui affichent l'état, le Raspberry Pi inclut une commande personnalisée nommée **vcgencmd**, qui permet de signaler les tensions et les températures et autres informations. En utilisant la commande **vcgencmd** (video core general commande)

```
pi@raspberrypi:~ $ vcgencmd get_camera
supported=1 detected=1
```

la commande renvoie le support et la détection de la caméra

### 3 démarrage du streaming en rtsp en UNICAST

RTSP ou Real Time Streaming Protocol (protocole de *streaming* temps-réel) est un protocole applicatif du même genre que HTTP permettant à un client de contrôler des flux multimédia du type audio et vidéo. Il ne fournit pas lui-même le flux qui est à la charge d'un autre protocole RTP.

Entrez la commande suivante pour démarrer le streaming:

```
pi@raspberrypi:~ $ raspivid -o - -t 0 -w 640 -h 360 -fps 25 | cvlc -vvv
stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

**raspivid** est utilisé pour capturer la vidéo

"-o -" provoque l'écriture de la sortie sur stdout

"-t 0" définit le délai d'expiration sur infini

"-w 640" définit la largeur de l'image 640px

"-h 360" définit la hauteur de l'image

"-fps 25" définit le nb d'images par seconde

**cvlc** est le lecteur console de vlc

"-vvv" met vlc en mode debug, ce qui fait qu'il écrira beaucoup d'informations.

**stream:///dev/stdin** spécifie où obtenir le flux vidéo (l'entrée standard)

"-sout" et son argument spécifie où sortir le flux vidéo

**#rtp** Ce module doit être utilisé pour envoyer un flux en utilisant le protocole RTP

**sdp** Cette option permet de définir la manière dont le fichier SDP (Session Description Protocol) correspondant au flux doit être disponible.

**rtsp://** <IP d'interface locale: port / path> pour rendre le fichier SDP disponible à l'aide du protocole RTSP (voir RFC 2326). Remarque L' argument IP de l'interface locale est facultatif. Si ce n'est pas le cas, VLC écouterait toutes les interfaces disponibles.

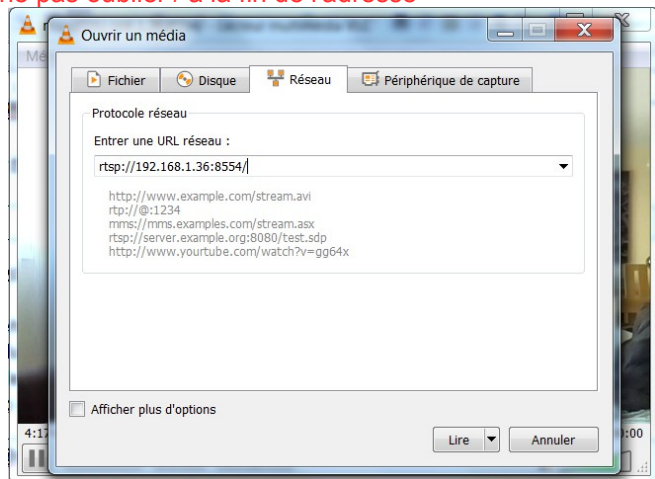
### 4 Affichage du flux RTSP en UNICAST

Vous avez besoin d'un lecteur prenant en charge le protocole RTSP.

Dans Média cliquez sur ouvrir un flux réseau puis saisir l'adresse du serveur

```
rtsp://XXX.XXX.XXX.XXX:8554/
```

Attention à ne pas oublier / à la fin de l'adresse



```
[73d00508] core input debug: Buffering 66%
[73d00508] core input debug: Buffering 80%
[73d00508] core input debug: Buffering 93%
[73d00508] core input debug: Stream buffering done (320 ms in 167 ms)
[73d00508] core input debug: Decoder wait done in 0 ms
[73b008c8] core stream out debug: net: connecting to [192.168.1.26]:59824
[73b008c8] core stream out debug: net: connecting to [192.168.1.26]:59825 from
[192.168.1.36]:49064
```

Les connections des clients sont affichées dans la console avec l'adresse et les n° de port utilisés.

## 5 Flux video en HTTP

VLC propose d'autres formes de streaming (HTTP) Entrez la commande suivante pour démarrer le streaming en http.

**#standard** Ce module enregistre le flux dans un fichier ou l'envoie sur un réseau

**access=http** Cette option permet de définir le support ou envoyer le flux http

**dst=:8080/stream** option pour définir le port 8080 et le fichier virtuel stream:

**mux=ts** définit le format de conteneur MPEG-TS (Transport Stream)

Pour le serveur :

```
pi@raspberrypi:~ $ raspivid -o - -t 0 -w 640 -h 360 -fps 25|cvlc
-vvv stream:///dev/stdin --sout
'#standard{access=http, mux=ts, dst=:8080/stream}' :demux=h264
```

Dans le client VLC Entrez l'URL réseau: pour lire le fichier virtuel stream

```
http://192.168.1.36:8080/stream
```

Autre possibilité : Pour le serveur Port écouté 65000

```
pi@raspberrypi3:~ $ raspivid -o - -t 0 -w 640 -h 360 -fps 25|cvlc -vvv
stream:///dev/stdin --sout '#http {mux=ffmpeg {mux=flv}, dst=:65000/}'
:demux=h264
```

Dans le client VLC Entrez l'URL réseau du serveur: pour lire le flux vidéo

```
http://192.168.1.36:65000
```

## 6 Autre possibilité avec la syntaxe #http

```
pi@raspberrypi3:~ $ raspivid -o - -t 0 -w 640 -h 360 -fps 25|cvlc
-vvv stream:///dev/stdin --sout '#http {mux=ts, dst=:65000/essai.mpg}'
:demux=h264
```

Dans le client VLC Entrez l'URL réseau: pour lire le flux

```
http://192.168.1.36:65000/essai.mpg
```

```
raspivid -o - -t 0 -w 640 -h 360 -fps 25|cvlc -vvv stream:///dev/stdin --sout '#http
{mux=ps, dst=:8080/essai.mjpeg}' :demux=h264
```

[https://wiki.videolan.org/Documentation:Streaming\\_HowTo/Advanced\\_Streaming\\_Using\\_the\\_Command\\_Line/](https://wiki.videolan.org/Documentation:Streaming_HowTo/Advanced_Streaming_Using_the_Command_Line/)

## *7 streaming avec V4l2*

```
cvlc v4l2:// :v4l2-dev=/dev/video0 :v4l2-width=640 :v4l2-height=360
:v4l2-fps=25 -
sout="#transcode{vcodec=h264, fps=25, vb=150, scale=1} :rtp{sdp=rtsp://:85
54}" &
```

ajouter & pour mettre le processus en arrière plan.

```
cvlc v4l2:// :v4l2-dev=/dev/video0 :v4l2-width=640 :v4l2-height=360
:v4l2-fps=25 --sout="#transcode{vcodec=h264, vb=150, fps=25, scale=1}
:standard{access=http, mux=ts, dst=:8080/stream}" &
```