

# How to capture video using v4l2 and streaming video using ffmpeg

method adapted from boneCV:

boneCV: <https://github.com/derekmolloy/boneCV>

<http://derekmolloy.ie/beaglebone/beaglebone-video-capture-and-image-processing-on-embedded-linux-using-opencv/>

<http://derekmolloy.ie/udp-video-streaming-beaglebone-black/>

useful previous cmpt433 how-to guides:

[http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2015-student-](http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2015-student-howtos/RecordingWebcamVideos.pdf)

[howtos/RecordingWebcamVideos.pdf](http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2015-student-howtos/RecordingWebcamVideos.pdf)

<http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2014-student-howtos/WebCam.pdf>

<http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/2016-student-howtos/WebCamVideoOpenCV.pdf>

## How to capture video

1. BBG comes with v4l2, to check

# whereis v4l2-ctl

v4l2-ctl: /usr/bin/v4l2-ctl

Some v4l2 commands(after connect camera to BBG):

# v4l2-ctl --list-formats

...

# v4l2-ctl --list-device

UVC Camera (046d:0825) (usb-musb-hdrc.1.auto-1):

/dev/video0

# v4l2-ctl --list-formats

ioctl: VIDIOC\_ENUM\_FMT

Index : 0

Type : Video Capture

Pixel Format: 'YUYV'

Name : YUYV 4:2:2

Index : 1

Type : Video Capture

Pixel Format: 'MJPG' (compressed)

Name : Motion-JPEG

# v4l2-ctl --get-priority

Priority: 2

# v4l2-ctl -D

Driver Info (not using libv4l2):

Driver name : uvcvideo

Card type : UVC Camera (046d:0825)

```
Bus info      : usb-musb-hdrc.1.auto-1
Driver version: 4.4.9
Capabilities  : 0x84200001
                Video Capture
                Streaming
                Extended Pix Format
                Device Capabilities
Device Caps   : 0x04200001
                Video Capture
                Streaming
                Extended Pix Format
```

## 2. how to capture video

1. # apt-get install libv4l-dev

2. store revised capture.c to NFS server

3. enter the NFS directory

```
# cd /mnt/remote/...
```

compile capture.c on BBG

```
# make
```

or

```
# gcc capture.c -lv4l2 -o capture
```

capture video and store it in output.raw

```
# ./captureVideo
```

or

```
# ./capture -F -c 300 -o > output.raw
```

-F force format to YUYV (after revision)

-c | --count Number of frames to grab [100] - use 0 for infinite

-o | --output Outputs stream to stdout

-h | --help

4. after capturing, a output.raw file will be produced. Convert raw file to mp4

```
#./raw2mpg4
```

or

```
# ffmpeg -f rawvideo -vcodec rawvideo -s 320x240 -r 25 -pix_fmt yuv420p -i output.raw -c:v libx264 -preset ultrafast -qp 0 output.mp4
```

## How to stream video using udp:

1. install ffmpeg on BBG

First, we need to add backports to our sources.list because we are running debian 8.4 on BBG

```
# cd /etc/apt/sources.list.d/
```

modify sources.list and add the following line to the end of the file

```
deb http://ftp.debian.org/debian jessie-backports main
```

then ffmpeg can be installed using apt-get

```
# apt-get update
```

```
# apt-get install ffmpeg
```

2. install VLC media player on host

```
$ sudo apt update
```

```
$ sudo apt install vlc
```

3. on BBG

```
# ./streamVideoUDP
```

or

```
# ./capture -F -o -c0|ffmpeg -f rawvideo -vcodec rawvideo -s 640x480 -r 25 -pix_fmt yuv420p -i pipe:0  
-f mjpeg udp://192.168.7.1:1234
```

on host, open VLC, Media->open Network Stream-> Network-> Network Protocol->Please enter a network URL:

```
udp://@:1234
```

then click play