26 MAY 2018 / FFMPEG

Stream camera video and audio with FFmpeg

FFmpeg is one of leading frameworks for multimedia processing. Among variety of features, FFmpeg can capture video and audio from your computer's camera and stream them over network to some other remote machine.

Use cases

It becomes more and more common that there is at least one desktop computer or lap top at your home with sole purpose to **collect dust**. Have you ever wondered if you can put it back in action to do something more useful?

Well I have an idea for you: turn it into live **surveillance** device

- (for fresh parents) to monitor baby's cradle and listen if baby started to *complain*

see if soup cooking transformed into volcano

- (for businessmen) to watch TV football match while playing video game
- (for teenagers) to monitor if mommy entered their room to read *secret diaries*

Install FFmpeg

Windows 7 or later

- go to FFmpeg builds web page, choose **Windows 64-bit** architecture, **static** linking and **Download build**.
- create directory `c:\\ffmpeg` and extract zip content into it,
- go to *Control Panel->System->Edit system environment variables* and click on *Environment variables*
- in *System variables* locate and select *Path* row, click *Edit* and **add** value `c:\ffmpeg\bin` .

Linux Ubuntu

Open *Terminal* and execute following

```
$ sudo apt-get install ffmpeg
$ sudo apt-get install v4l-utils
```

Discover camera and microphone system names

Windows

```
ffmpeg -list_devices true -f dshow -i dummy
```

Linux

From terminal type and execute

```
$ v4l2-ctl --list-devices
```

and you will see all video and audio devices available on your system. Something like this

```
ffmpeg version N-90264-g80798e3857 Copyright (c) 2000-2018 the FFmp
[dshow @ 00000128ed43a5c0] DirectShow video devices (some may be bo
[dshow @ 00000128ed43a5c0]  "Integrated Webcam"
[dshow @ 00000128ed43a5c0]     Alternative name "@device_pnp_\\?\us
[dshow @ 00000128ed43a5c0] DirectShow audio devices
[dshow @ 00000128ed43a5c0]  "Microphone (Realtek Audio)"
[dshow @ 00000128ed43a5c0]     Alternative name "@device_cm_{33D9A7
```

Find and note video and audio device names. In this example

- video device name is *Integrated Webcam*
- audio device name is *Microphone (Realtek Audio)*

Discover computer IP address

First discover **IP address** of computer or smart phone where you want to watch stream from camera:

- on linux: execute `ifconfig` from terminal
- on android: go to *Settings->WiFi->Advanced settings*

Usually IP address will be something like `192.168.1.x`. Note it down.

Capture and stream video and audio

Now, on machine with camera, open *Terminal/command prompt* and run following command to start capturing video and audio and generating live stream to another computer

Windows

```
ffmpeg -f dshow -i video="Integrated Webcam":audio="Microphone (Real
```

Linux

On linux, instead of `dshow` (DirectShow), we need to use 2 different drivers for video and audio:

- video capture - `v4l2` (Video For Linux drivers)
- audio capture - `alsa` (ALSA)

```
ffmpeg -f v4l2 -i video="Integrated Webcam" -f alsa -i hw:0 -profile
```

> *Do not forget to replace video and audio devices names and destination IP address with previously discovered values.*

Options demystified

Lets reformat previous command so that we can see used options better

```
ffmpeg -f dshow \
-i video="Integrated Webcam":audio="Microphone (Realtek Audio)" \
-profile:v high -pix_fmt yuvj420p -level:v 4.1 -preset ultrafast -tu
-vcodec libx264 -r 10 -b:v 512k -s 640x360 \
-acodec aac -ac 2 -ab 32k -ar 44100 \
-f mpegts -flush_packets 0 udp://192.168.1.4:5000?pkt_size=1316
```

- `-f fshow` - windows system drivers for capturing video and audio
- `-f v4l2` - linux system drivers for capturing video
- `-f alsa` - linux system drivers for capturing audio
- `-i` - ffmpeg option that defines **input**
- `-vcodec libx264` - raw video from camera will be encoded using H264 video codec
- `-r 10` - video FPS (frames per second)
- `-b:v 512k` - video bitrate Kb/s (kilo bits per second)
- `-s 640x360` - video width and height
- `-acodec aac` - raw audio from microphone will be encoded using AAC audio codec
- `-ac 2` - 2 audio channels (stereo)
- `-ab 32k` - audio bitrate in Kb/s
- `-ar 44100` - audio sampling rate 44.1 KHz
- `-f mpegts` - video and audio will be packed into MPEG transport stream (MPEG TS)
- `udp://192.168.1.4:5000` - MPEG transport stream is sent via UDP protocol to computer with IP address 192.168.1.4 on

Play camera stream

Windows/Linux

On destination computer stream can be watched using VLC player

- go to menu Media->Open Network Stream...
- in URL field type `udp://@0.0.0.0:5000` and press Play

Android

Install MX Player, from top menu go to *Network stream*, type `udp://127.0.0.1:5000` in URL field and press OK.

Happy spying !!! :)

Dusan Kovacevic
Versatile software developer in everlasting quest of exploring new technologies

Read More

ALSO ON 4YOUNGPADAWANS

PrimeReact menubar with React Router
3 years ago • 1 comment
React SPA navigation system using PrimeReact Menubar and React Router

End-to-end encryption from React client to ...
2 years ago • 4 comments
Practical example with full source code showing how to encrypt payment ...

Gstreamer real life examples
3 years ago • 5 comments
Nowadays media processing knowledge is becoming kind of ...

Dov ser
3 ye
How serv usin

Adam Kolany • 10 months ago

does not work:
==========
root@kolanybuchse:~# v4l2-ctl --list-devices
Integrated_Webcam_HD: Integrate (usb-0000:00:14.0-5):
/dev/video0

root@kolanybuchse:~#
==========
root@kolanybuchse:~# ffmpeg -f v4l2 -i video="Integrated_Webcam_HD" -f alsa -i hw:0 -profile:v high -pix_fmt yuvj420p -level:v 4.1 -preset ultrafast -tune zerolatency -vcodec libx264 -r 10 -b:v 512k -s 640x360 -acodec aac -strict -2 -ac 2 -ab 32k -ar 44100 -f mpegts -flush_packets 0 udp://192.168.0.101:5000?pkt_size=1316
ffmpeg version 3.4.6-0ubuntu0.18.04.1 Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 7 (Ubuntu 7.3.0-16ubuntu3)
configuration: --prefix=/usr --extra-version=0ubuntu0.18.04.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --enable-gpl --disable-stripping --enable-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite --

Yutaka Yamamoto ➚ Adam Kolany • 10 months ago • edited

Try this,

```
# ffmpeg -f v4l2 -i video="Integrated_Webcam_HD" ~~~ => # ffmpeg -f v4l2 -i /dev/video0 ~~~
***Not "-i video="integrated_Webcam_HD"" but "-i /dev/video0"
```


Arish Shaikh ➚ Adam Kolany • 10 months ago

same problem

Gav ➚ Arish Shaikh • 8 days ago

You have to replace the "Integrated_Webcam_HD" string with the name of your webcam.

Get the name by running "ffmpeg -list_devices true -f dshow -i dummy" at the

video and audio devices)
[dshow @ 0000019ed8bdcf40] "Microsoft® LifeCam Cinema(TM)"

My camera is "Microsoft® LifeCam Cinema(TM)"

Reply • Share ›

Hubery Wang • a year ago
It does help me,thx!

Reply • Share ›

Dmitriy Laguoto • 2 years ago
Great article! Thanks!

JSF

JavaServer Faces overview for beginners

One of the first web oriented frameworks I got interested in was JavaServer Faces known as JSF. This article gives summary of JSF framework and tools necessary to create and run JSF web

8 MIN READ

REACT

PrimeReact lazy table filtering

PrimeReact data table lazy loading, filtering, sorting, column and header templates

7 MIN READ