
Readme for Linux device driver for the OmniVision OV511 USB to camera bridge IC

Author: Mark McClelland

Homepage: <http://alpha.dyndns.org/ov511>

INTRODUCTION:

This is a driver for the OV511, a USB-only chip used in many "webcam" devices. Any camera using the OV511/OV511+ and the OV6620/OV7610/20/20AE should work. Video capture devices that use the Philips SAA7111A decoder also work. It supports streaming and capture of color or monochrome video via the Video4Linux API. Most V4L apps are compatible with it. Most resolutions with a width and height that are a multiple of 8 are supported.

If you need more information, please visit the OV511 homepage at the above URL.

WHAT YOU NEED:

- If you want to help with the development, get the chip's specification docs at <http://www.ovt.com/omniusbp.html>
- A Video4Linux compatible frame grabber program (I recommend vidcat and xawtv)
vidcat is part of the w3cam package: <http://mpx.freeshell.net/>
xawtv is available at: <http://linux.bytesex.org/xawtv/>

HOW TO USE IT:

Note: These are simplified instructions. For complete instructions see:
<http://alpha.dyndns.org/ov511/install.html>

You must have first compiled USB support, support for your specific USB host controller (UHCI or OHCI), and Video4Linux support for your kernel (I recommend making them modules.) Make sure "Enforce bandwidth allocation" is NOT enabled.

Next, (as root):

```
modprobe usbcore
modprobe usb-uhci <OR> modprobe usb-ohci
modprobe videodev
modprobe ov511
```

If it is not already there (it usually is), create the video device:

```
mknod /dev/video0 c 81 0
```

Optionally, symlink /dev/video to /dev/video0

You will have to set permissions on this device to allow you to read/write from it:

```
chmod 666 /dev/video
chmod 666 /dev/video0 (if necessary)
```

Now you are ready to run a video app! Both vidcat and xawtv work well for me

ov511.txt.txt

at 640x480.

[Using vidcat:]

```
vidcat -s 640x480 -p c > test.jpg
xview test.jpg
```

[Using xawtv:]

From the main xawtv directory:

```
make clean
./configure
make
make install
```

Now you should be able to run xawtv. Right click for the options dialog.

MODULE PARAMETERS:

You can set these with: `insmod ov511 NAME=VALUE`
There is currently no way to set these on a per-camera basis.

NAME: autobright
TYPE: integer (Boolean)
DEFAULT: 1
DESC: Brightness is normally under automatic control and can't be set manually by the video app. Set to 0 for manual control.

NAME: autogain
TYPE: integer (Boolean)
DEFAULT: 1
DESC: Auto Gain Control enable. This feature is not yet implemented.

NAME: autoexp
TYPE: integer (Boolean)
DEFAULT: 1
DESC: Auto Exposure Control enable. This feature is not yet implemented.

NAME: debug
TYPE: integer (0-6)
DEFAULT: 3
DESC: Sets the threshold for printing debug messages. The higher the value, the more is printed. The levels are cumulative, and are as follows:
0=no debug messages
1=init/detection/unload and other significant messages
2=some warning messages
3=config/control function calls
4=most function calls and data parsing messages
5=highly repetitive msgs

NAME: snapshot
TYPE: integer (Boolean)
DEFAULT: 0
DESC: Set to 1 to enable snapshot mode. `read()/VIDIOCSYNC` will block until the snapshot button is pressed. Note: enabling this mode disables

ov511.txt.txt
/proc/video/ov511/<minor#>/button

NAME: cams
TYPE: integer (1-4 for OV511, 1-31 for OV511+)
DEFAULT: 1
DESC: Number of cameras allowed to stream simultaneously on a single bus.
Values higher than 1 reduce the data rate of each camera, allowing two or more to be used at once. If you have a complicated setup involving both OV511 and OV511+ cameras, trial-and-error may be necessary for finding the optimum setting.

NAME: compress
TYPE: integer (Boolean)
DEFAULT: 0
DESC: Set this to 1 to turn on the camera's compression engine. This can potentially increase the frame rate at the expense of quality, if you have a fast CPU. You must load the proper compression module for your camera before starting your application (ov511_decomp or ov518_decomp).

NAME: testpat
TYPE: integer (Boolean)
DEFAULT: 0
DESC: This configures the camera's sensor to transmit a colored test-pattern instead of an image. This does not work correctly yet.

NAME: dumppix
TYPE: integer (0-2)
DEFAULT: 0
DESC: Dumps raw pixel data and skips post-processing and format conversion. It is for debugging purposes only. Options are:
0: Disable (default)
1: Dump raw data from camera, excluding headers and trailers
2: Dumps data exactly as received from camera

NAME: led
TYPE: integer (0-2)
DEFAULT: 1 (Always on)
DESC: Controls whether the LED (the little light) on the front of the camera is always off (0), always on (1), or only on when driver is open (2). This is not supported with the OV511, and might only work with certain cameras (ones that actually have the LED wired to the control pin, and not just hard-wired to be on all the time).

NAME: dump_bridge
TYPE: integer (Boolean)
DEFAULT: 0
DESC: Dumps the bridge (OV511[+] or OV518[+]) register values to the system log. Only useful for serious debugging/development purposes.

NAME: dump_sensor
TYPE: integer (Boolean)
DEFAULT: 0
DESC: Dumps the sensor register values to the system log. Only useful for serious debugging/development purposes.

NAME: printph

ov511.txt.txt

TYPE: integer (Boolean)

DEFAULT: 0

DESC: Setting this to 1 will dump the first 12 bytes of each isoc frame. This is only useful if you are trying to debug problems with the isoc data stream (i.e.: camera initializes, but vidcat hangs until Ctrl-C). Be warned that this dumps a large number of messages to your kernel log.

NAME: phy, phuv, pvy, pvuv, qhy, qhuv, qvy, qvuv

TYPE: integer (0-63 for phy and phuv, 0-255 for rest)

DEFAULT: OV511 default values

DESC: These are registers 70h - 77h of the OV511, which control the prediction ranges and quantization thresholds of the compressor, for the Y and UV channels in the horizontal and vertical directions. See the OV511 or OV511+ data sheet for more detailed descriptions. These normally do not need to be changed.

NAME: lightfreq

TYPE: integer (0, 50, or 60)

DEFAULT: 0 (use sensor default)

DESC: Sets the sensor to match your lighting frequency. This can reduce the appearance of "banding", i.e. horizontal lines or waves of light and dark that are often caused by artificial lighting. Valid values are:
0 - Use default (depends on sensor, most likely 60 Hz)
50 - For European and Asian 50 Hz power
60 - For American 60 Hz power

NAME: bandingfilter

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Enables the sensor's banding filter exposure algorithm. This reduces or stabilizes the "banding" caused by some artificial light sources (especially fluorescent). You might have to set lightfreq correctly for this to work right. As an added bonus, this sometimes makes it possible to capture your monitor's output.

NAME: fastset

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Allows picture settings (brightness, contrast, color, and hue) to take effect immediately, even in the middle of a frame. This reduces the time to change settings, but can ruin frames during the change. Only affects OmniVision sensors.

NAME: force_palette

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Forces the palette (color format) to a specific value. If an application requests a different palette, it will be rejected, thereby forcing it to try others until it succeeds. This is useful for forcing greyscale mode with a color camera, for example. Supported modes are:
0 (Allows all the following formats)
1 VIDEO_PALETTE_GREY (Linear greyscale)
10 VIDEO_PALETTE_YUV420 (YUV 4:2:0 Planar)
15 VIDEO_PALETTE_YUV420P (YUV 4:2:0 Planar, same as 10)

NAME: backlight

ov511.txt.txt

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Setting this flag changes the exposure algorithm for OmniVision sensors such that objects in the camera's view (i.e. your head) can be clearly seen when they are illuminated from behind. It reduces or eliminates the sensor's auto-exposure function, so it should only be used when needed. Additionally, it is only supported with the OV6620 and OV7620.

NAME: unit_video

TYPE: Up to 16 comma-separated integers

DEFAULT: 0,0,0... (automatically assign the next available minor(s))

DESC: You can specify up to 16 minor numbers to be assigned to ov511 devices. For example, "unit_video=1,3" will make the driver use /dev/video1 and /dev/video3 for the first two devices it detects. Additional devices will be assigned automatically starting at the first available device node (/dev/video0 in this case). Note that you cannot specify 0 as a minor number. This feature requires kernel version 2.4.5 or higher.

NAME: remove_zeros

TYPE: integer (Boolean)

DEFAULT: 0 (do not skip any incoming data)

DESC: Setting this to 1 will remove zero-padding from incoming data. This will compensate for the blocks of corruption that can appear when the camera cannot keep up with the speed of the USB bus (eg. at low frame resolutions). This feature is always enabled when compression is on.

NAME: mirror

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Setting this to 1 will reverse ("mirror") the image horizontally. This might be necessary if your camera has a custom lens assembly. This has no effect with video capture devices.

NAME: ov518_color

TYPE: integer (Boolean)

DEFAULT: 0 (off)

DESC: Enable OV518 color support. This is off by default since it doesn't work most of the time. If you want to try it, you must also load ov518_decomp with the "nouv=0" parameter. If you get improper colors or diagonal lines through the image, restart your video app and try again. Repeat as necessary.

WORKING FEATURES:

- o Color streaming/capture at most widths and heights that are multiples of 8.
- o Monochrome (use force_palette=1 to enable)
- o Setting/getting of saturation, contrast, brightness, and hue (only some of them work the OV7620 and OV7620AE)
- o /proc status reporting
- o SAA7111A video capture support at 320x240 and 640x480
- o Compression support
- o SMP compatibility

HOW TO CONTACT ME:

You can email me at mark@alpha.dyndns.org . Please prefix the subject line with "OV511: " so that I am certain to notice your message.

ov511.txt.txt

CREDITS:

The code is based in no small part on the CPiA driver by Johannes Erdfelt, Randy Dunlap, and others. Big thanks to them for their pioneering work on that and the USB stack. Thanks to Bret Wallach for getting camera reg IO, ISOC, and image capture working. Thanks to Orion Sky Lawlor, Kevin Moore, and Claudio Matsuoka for their work as well.