

W996[87]CF JPEG USB Dual Mode Camera Chip  
Driver for Linux 2.6 (basic version)

---

- Documentation -

Index

=====

1. Copyright
2. Disclaimer
3. License
4. Overview
5. Supported devices
6. Module dependencies
7. Module loading
8. Module parameters
9. Contact information
10. Credits

1. Copyright

=====

Copyright (C) 2002-2004 by Luca Risolia <luca.risolia@studio.unibo.it>

2. Disclaimer

=====

Winbond is a trademark of Winbond Electronics Corporation.  
This software is not sponsored or developed by Winbond.

3. License

=====

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

4. Overview

=====

This driver supports the video streaming capabilities of the devices mounting Winbond W9967CF and Winbond W9968CF JPEG USB Dual Mode Camera Chips. OV681 based cameras should be supported as well.

w9968cf.txt.txt

The driver is divided into two modules: the basic one, "w9968cf", is needed for the supported devices to work; the second one, "w9968cf-vpp", is an optional module, which provides some useful video post-processing functions like video decoding, up-scaling and colour conversions.

Note that the official kernels do neither include nor support the second module for performance purposes. Therefore, it is always recommended to download and install the latest and complete release of the driver, replacing the existing one, if present.

The latest and full-featured version of the W996[87]CF driver can be found at: <http://www.linux-projects.org>. Please refer to the documentation included in that package, if you are going to use it.

Up to 32 cameras can be handled at the same time. They can be connected and disconnected from the host many times without turning off the computer, if your system supports the hotplug facility.

To change the default settings for each camera, many parameters can be passed through command line when the module is loaded into memory.

The driver relies on the Video4Linux, USB and I2C core modules. It has been designed to run properly on SMP systems as well. An additional module, "ovcamchip", is mandatory; it provides support for some OmniVision image sensors connected to the W996[87]CF chips; if found in the system, the module will be automatically loaded by default (provided that the kernel has been compiled with the automatic module loading option).

## 5. Supported devices

=====

At the moment, known W996[87]CF and OV681 based devices are:

- Aroma Digi Pen VGA Dual Mode ADG-5000 (unknown image sensor)
- AVerMedia AVerTV USB (SAA7111A, Philips FI1216Mk2 tuner, PT2313L audio chip)
- Creative Labs Video Blaster WebCam Go (OmniVision OV7610 sensor)
- Creative Labs Video Blaster WebCam Go Plus (OmniVision OV7620 sensor)
- Lebon LDC-035A (unknown image sensor)
- Ezonics EZ-802 EZMega Cam (OmniVision OV8610C sensor)
- OmniVision OV8610-EDE (OmniVision OV8610 sensor)
- OPCOM Digi Pen VGA Dual Mode Pen Camera (unknown image sensor)
- Pretec Digi Pen-II (OmniVision OV7620 sensor)
- Pretec DigiPen-480 (OmniVision OV8610 sensor)

If you know any other W996[87]CF or OV681 based cameras, please contact me.

The list above does not imply that all those devices work with this driver: up until now only webcams that have an image sensor supported by the "ovcamchip" module work. Kernel messages will always tell you whether this is case.

Possible external microcontrollers of those webcams are not supported: this means that still images cannot be downloaded from the device memory.

Furthermore, it's worth to note that I was only able to run tests on my "Creative Labs Video Blaster WebCam Go". Donations of other models, for additional testing and full support, would be much appreciated.

## 6. Module dependencies

=====

For it to work properly, the driver needs kernel support for Video4Linux, USB and I2C, and the "ovcamchip" module for the image sensor. Make sure you are not actually using any external "ovcamchip" module, given that the W996[87]CF driver depends on the version of the module present in the official kernels.

The following options of the kernel configuration file must be enabled and corresponding modules must be compiled:

```
# Multimedia devices
#
CONFIG_VIDEO_DEV=m

# I2C support
#
CONFIG_I2C=m
```

The I2C core module can be compiled statically in the kernel as well.

```
# OmniVision Camera Chip support
#
CONFIG_VIDEO_OVCAMCHIP=m

# USB support
#
CONFIG_USB=m
```

In addition, depending on the hardware being used, only one of the modules below is necessary:

```
# USB Host Controller Drivers
#
CONFIG_USB_EHCI_HCD=m
CONFIG_USB_UHCI_HCD=m
CONFIG_USB_OHCI_HCD=m
```

And finally:

```
# USB Multimedia devices
#
CONFIG_USB_W9968CF=m
```

## 7. Module loading

=====

To use the driver, it is necessary to load the "w9968cf" module into memory after every other module required.

Loading can be done this way, from root:

```
[root@localhost home]# modprobe usbcore
[root@localhost home]# modprobe i2c-core
[root@localhost home]# modprobe videodev
[root@localhost home]# modprobe w9968cf
```

At this point the pertinent devices should be recognized: "dmesg" can be used to analyze kernel messages:

```
[user@localhost home]$ dmesg
```

There are a lot of parameters the module can use to change the default settings for each device. To list every possible parameter with a brief explanation about them and which syntax to use, it is recommended to run the "modinfo" command:

```
[root@localhost home]# modinfo w9968cf
```

## 8. Module parameters

Module parameters are listed below:

---

|              |   |
|--------------|---|
| Name:        | ovmod_load  |
| Type:        | bool  |
| Syntax:      | <0 1>   |
| Description: | Automatic 'ovcamchip' module loading: 0 disabled, 1 enabled. If enabled, 'insmod' searches for the required 'ovcamchip' module in the system, according to its configuration, and loads that module automatically. This action is performed as once soon as the 'w9968cf' module is loaded into memory. |
| Default:     | 1   |

---

|              |  |
|--------------|--|
| Name:        | simcams  |
| Type:        | int  |
| Syntax:      | <n>  |
| Description: | Number of cameras allowed to stream simultaneously. n may vary from 0 to 32. |
| Default:     | 32   |

---

|              |  |
|--------------|--|
| Name:        | video_nr   |
| Type:        | int array (min = 0, max = 32)  |
| Syntax:      | <-1 n[,...]>   |
| Description: | Specify V4L minor mode number.<br>-1 = use next available<br>n = use minor number n<br>You can specify up to 32 cameras this way.<br>For example:<br>video_nr=-1,2,-1 would assign minor number 2 to the second recognized camera and use auto for the first one and for every other camera. |
| Default:     | -1   |

---

|              |  |
|--------------|--|
| Name:        | packet_size  |
| Type:        | int array (min = 0, max = 32)  |
| Syntax:      | <n[,...]>  |
| Description: | Specify the maximum data payload size in bytes for alternate settings, for each device. n is scaled between 63 and 1023. |
| Default:     | 1023   |

---

|       |             |
|-------|-------------|
| Name: | max_buffers |
|-------|-------------|

---

Type: int array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: For advanced users.  
 Specify the maximum number of video frame buffers to allocate  
 for each device, from 2 to 32.  
 Default: 2

---

Name: double\_buffer  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Hardware double buffering: 0 disabled, 1 enabled.  
 It should be enabled if you want smooth video output: if you  
 obtain out of sync. video, disable it, or try to  
 decrease the 'clockdiv' module parameter value.  
 Default: 1 for every device.

---

Name: clamping  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Video data clamping: 0 disabled, 1 enabled.  
 Default: 0 for every device.

---

Name: filter\_type  
 Type: int array (min = 0, max = 32)  
 Syntax: <0|1|2[,...]>  
 Description: Video filter type.  
 0 none, 1 (1-2-1) 3-tap filter, 2 (2-3-6-3-2) 5-tap filter.  
 The filter is used to reduce noise and aliasing artifacts  
 produced by the CCD or CMOS image sensor.  
 Default: 0 for every device.

---

Name: largeview  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Large view: 0 disabled, 1 enabled.  
 Default: 1 for every device.

---

Name: upscaling  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Software scaling (for non-compressed video only):  
 0 disabled, 1 enabled.  
 Disable it if you have a slow CPU or you don't have enough  
 memory.  
 Default: 0 for every device.  
 Note: If 'w9968cf-vpp' is not present, this parameter is set to 0.

---

Name: decompression  
 Type: int array (min = 0, max = 32)  
 Syntax: <0|1|2[,...]>  
 Description: Software video decompression:  
 0 = disables decompression  
 (doesn't allow formats needing decompression).  
 1 = forces decompression  
 (allows formats needing decompression only).  
 2 = allows any permitted formats.

# w9968cf.txt.txt

Formats supporting (de)compressed video are YUV422P and YUV420P/YUV420 in any resolutions where width and height are multiples of 16.

Default: 2 for every device.

Note: If 'w9968cf-vpp' is not present, forcing decompression is not allowed; in this case this parameter is set to 2.

---

Name: force\_palette  
Type: int array (min = 0, max = 32)  
Syntax: <0|9|10|13|15|8|7|1|6|3|4|5[,...]>  
Description: Force picture palette.  
In order:  
0 = Off - allows any of the following formats:  
9 = UYVY 16 bpp - Original video, compression disabled  
10 = YUV420 12 bpp - Original video, compression enabled  
13 = YUV422P 16 bpp - Original video, compression enabled  
15 = YUV420P 12 bpp - Original video, compression enabled  
8 = YUVY 16 bpp - Software conversion from UYVY  
7 = YUV422 16 bpp - Software conversion from UYVY  
1 = GREY 8 bpp - Software conversion from UYVY  
6 = RGB555 16 bpp - Software conversion from UYVY  
3 = RGB565 16 bpp - Software conversion from UYVY  
4 = RGB24 24 bpp - Software conversion from UYVY  
5 = RGB32 32 bpp - Software conversion from UYVY  
When not 0, this parameter will override 'decompression'.  
Default: 0 for every device. Initial palette is 9 (UYVY).  
Note: If 'w9968cf-vpp' is not present, this parameter is set to 9.

---

Name: force\_rgb  
Type: bool array (min = 0, max = 32)  
Syntax: <0|1[,...]>  
Description: Read RGB video data instead of BGR:  
1 = use RGB component ordering.  
0 = use BGR component ordering.  
This parameter has effect when using RGBX palettes only.  
Default: 0 for every device.

---

Name: autobright  
Type: bool array (min = 0, max = 32)  
Syntax: <0|1[,...]>  
Description: Image sensor automatically changes brightness:  
0 = no, 1 = yes  
Default: 0 for every device.

---

Name: autoexp  
Type: bool array (min = 0, max = 32)  
Syntax: <0|1[,...]>  
Description: Image sensor automatically changes exposure:  
0 = no, 1 = yes  
Default: 1 for every device.

---

Name: lightfreq  
Type: int array (min = 0, max = 32)  
Syntax: <50|60[,...]>  
Description: Light frequency in Hz:  
50 for European and Asian lighting, 60 for American lighting.

Default: 50 for every device.

---

Name: bandingfilter  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Banding filter to reduce effects of fluorescent lighting:  
 0 disabled, 1 enabled.  
 This filter tries to reduce the pattern of horizontal light/dark bands caused by some (usually fluorescent) lighting.  
 Default: 0 for every device.

---

Name: clockdiv  
 Type: int array (min = 0, max = 32)  
 Syntax: <-1|n[,...]>  
 Description: Force pixel clock divisor to a specific value (for experts):  
 n may vary from 0 to 127.  
 -1 for automatic value.  
 See also the 'double\_buffer' module parameter.  
 Default: -1 for every device.

---

Name: backlight  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Objects are lit from behind:  
 0 = no, 1 = yes  
 Default: 0 for every device.

---

Name: mirror  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: Reverse image horizontally:  
 0 = no, 1 = yes  
 Default: 0 for every device.

---

Name: monochrome  
 Type: bool array (min = 0, max = 32)  
 Syntax: <0|1[,...]>  
 Description: The image sensor is monochrome:  
 0 = no, 1 = yes  
 Default: 0 for every device.

---

Name: brightness  
 Type: long array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: Set picture brightness (0-65535).  
 This parameter has no effect if 'autobright' is enabled.  
 Default: 31000 for every device.

---

Name: hue  
 Type: long array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: Set picture hue (0-65535).  
 Default: 32768 for every device.

---

Name: colour

Type: long array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: Set picture saturation (0-65535).  
 Default: 32768 for every device.

---

Name: contrast  
 Type: long array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: Set picture contrast (0-65535).  
 Default: 50000 for every device.

---

Name: whiteness  
 Type: long array (min = 0, max = 32)  
 Syntax: <n[,...]>  
 Description: Set picture whiteness (0-65535).  
 Default: 32768 for every device.

---

Name: debug  
 Type: int  
 Syntax: <n>  
 Description: Debugging information level, from 0 to 6:  
           0 = none (use carefully)  
           1 = critical errors  
           2 = significant informations  
           3 = configuration or general messages  
           4 = warnings  
           5 = called functions  
           6 = function internals  
           Level 5 and 6 are useful for testing only, when only one  
           device is used.  
 Default: 2

---

Name: specific\_debug  
 Type: bool  
 Syntax: <0|1>  
 Description: Enable or disable specific debugging messages:  
           0 = print messages concerning every level <= 'debug' level.  
           1 = print messages concerning the level indicated by 'debug'.  
 Default: 0

---

## 9. Contact information

---

I may be contacted by e-mail at <luca.risolio@studio.unibo.it>.

I can accept GPG/PGP encrypted e-mail. My GPG key ID is 'FCE635A4'.  
 My public 1024-bit key should be available at your keyserver; the fingerprint  
 is: '88E8 F32F 7244 68BA 3958 5D40 99DA 5D2A FCE6 35A4'.

## 10. Credits

---

The development would not have proceed much further without having looked at  
 the source code of other drivers and without the help of several persons; in  
 particular:



w9968cf.txt.txt

- the I2C interface to kernel and high-level image sensor control routines have been taken from the OV511 driver by Mark McClelland;
- memory management code has been copied from the bttv driver by Ralph Metzler, Marcus Metzler and Gerd Knorr;
- the low-level I2C read function has been written by Frederic Jouault;
- the low-level I2C fast write function has been written by Piotr Czerczak.