

Webcam: Instalación y uso

Por Paco Aldarias Raya

Impreso: 4 de febrero de 2004

Email: pacolinux@inicia.punto.es

Web: <http://pagina.de/pacodebian>

Con Linux Debian. En Valencia (España)

Este documento es de libre reproducción siempre que se cite su fuente.

Realizado con: \LaTeX

Índice

Índice	1
1. Introducción	2
2. Versiones.	3
3. Contexto	3
3.1. Hardware	3
3.2. Software Utilizado	3
4. Instalación del software	3
5. Instalación de los drivers	3
6. Ver que tipo de usb disponemos	4
7. El kernel	4
7.1. USB support	4
7.2. Multimedia devices	5
7.3. Character Devices	5
8. Módulos cargados	5

9. El arranque	6
9.1. usb	6
9.2. video	6
9.3. cam	6
10. Estar seguro que funcionan USB	7
11. Comprobar el dispositivo	8
11.1. /dev/video0	8
11.2. /dev/video	8
11.3. Dar permisos	9
12. El sistema de ficheros: fstab	9
13. Poniéndola en marcha	9
14. Para apagar la cámara	9
15. Script de Arranque y parada	9
16. Ver si la reconoce	10
17. Una foto	11
18. Software para webcam	11
18.1. Instalación	11
18.2. Configuración de webcam	11
18.3. Script para automatizar	12
19. Crear videos: motion	13
20. Bibliografía	14
21. Agradecimientos	14

1. Introducción

Aquí explico como poner en marcha una camara de video.

2. Versiones.

1. 9.11.03. V.1.0. Primera versión.

2. 4.2.04. V.1.1. Explicación de instalación de los drivers.

3. Contexto

3.1. Hardware

1. Video Camara: Philips PCVC 720K USB TOUCAM
2. PC: AMD-K6-II 450MZ 320RAM

3.2. Software Utilizado

1. Linux Debian Sid. Kernel 2.4.20.
2. xawtv Version: 3.82
3. OVCam Drivers for 2.4 and 2.5 kernels: ov511-2.21
4. mesag3-glide2 para que funcion xawtv

4. Instalación del software

```
apt-get install xawtv mesag3-glide2
```

5. Instalación de los drivers

Bajarse el fichero :

```
http://alpha.dyndns.org/ov511/download/2.xx/distros/ov511-2.21.tar.bz2
```

Aquí están todos las versiones:

```
http://alpha.dyndns.org/ov511/download.html
```

1. Como usuario normal descomprimierlo:

```
tar jxvf ov511-2.21.tar.bz2
```
2. Como usuario normal compilarlo:

```
make
```
3. Como root instalarlo:

```
make install
```

6. Ver que tipo de usb disponemos

```
# lspci
00:00.0 Host bridge: VIA Technologies, Inc. VT82C598 [Apollo MVP3] (rev 04)
00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x [Apollo MVP3/Pro133x AG
00:07.0 ISA bridge: VIA Technologies, Inc. VT82C686 [Apollo Super South] (rev 1b
00:07.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B PIPC Bus Master
00:07.2 USB Controller: VIA Technologies, Inc. USB (rev 0e)
00:07.3 USB Controller: VIA Technologies, Inc. USB (rev 0e)
00:07.4 Host bridge: VIA Technologies, Inc. VT82C686 [Apollo Super ACPI] (rev 20
00:08.0 VGA compatible controller: 3Dfx Interactive, Inc. Voodoo Banshee (rev 03
00:09.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139
00:0a.0 Multimedia audio controller: Ensoniq 5880 AudioPCI (rev 02)
00:0b.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139
```

Como se puede ver detecta los usb, y son VIA Technologies

7. El kernel

7.1. USB support

En USB support:

```
<*> Support for USB
    [*]   USB verbose debug messages
    [*]   Preliminary USB device filesystem
    <M>   UHCI Alternate Driver (JE) support
    <M>   OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support
    <M>   USB Audio support
    <M>   USB OV511 Camera support
    <M>   USB Philips Cameras
```

Contenido de .config

```
# cat /usr/src/linux-2.4.20/.config | grep USB
CONFIG_USB=y
CONFIG_USB_DEBUG=y
CONFIG_USB_DEVICEFS=y
CONFIG_USB_UHCI_ALT=m
CONFIG_USB_OHCI=m
CONFIG_USB_AUDIO=m
CONFIG_USB_OV511=m
CONFIG_USB_PWC=m
```

7.2. Multimedia devices

En Multimedia devices:

```
<M> Video For Linux  En Video For Linux
---> [*]  V4L information in proc filesystem
```

Contenido de .config

```
#cat /usr/src/linux-2.4.20/.config | grep VIDEO
CONFIG_VIDEO_DEV=m
CONFIG_VIDEO_PROC_FS=y
CONFIG_VIDEO_SELECT=y
```

7.3. Character Devices

En Character Devices -

```
I2C support  --->
<*> I2C support
<M> I2C device interface
<M> I2C /proc interface (required for hardware sensors)
```

Contenido de .config

```
cat /usr/src/linux-2.4.20/.config | grep CONFIG_I2C
CONFIG_I2C=y
CONFIG_I2C_CHARDEV=m
CONFIG_I2C_PROC=m
```

8. Módulos cargados

En nuestro caso hemos elegido en el kernel q queremos módulos para trabajar con la cámara.

```
# lsmod
Module                Size  Used by    Not tainted
uhci                   25264  0 (unused)
ovfx2                  27444  0 (unused)
pwc                    40968  0 (unused)
ovsensor              14376  1
ov511                  81592  0
videodev              5728  1 [ovfx2 pwc ov511]
```

9. El arranque

9.1. usb

```
# dmesg | grep usb
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
usb.c: registered new driver ov511
usb.c: registered new driver Philips webcam
usb.c: registered new driver ovfx2
usb.c: new USB bus registered, assigned bus number 1
usb.c: kmalloc IF c68af3c0, numif 1
usb.c: new device strings: Mfr=0, Product=2, SerialNumber=1
usb.c: USB device number 1 default language ID 0x0
usb.c: hub driver claimed interface c68af3c0
usb.c: kusbdevfs: /sbin/hotplug add 1
usb.c: kusbdevfs policy returned 0xffffffff
usb.c: new USB bus registered, assigned bus number 2
usb.c: kmalloc IF c68af5a0, numif 1
usb.c: new device strings: Mfr=0, Product=2, SerialNumber=1
usb.c: USB device number 1 default language ID 0x0
usb.c: hub driver claimed interface c68af5a0
usb.c: kusbdevfs: /sbin/hotplug add 1
usb.c: kusbdevfs policy returned 0xffffffff
usb.c: kmalloc IF c68afa60, numif 1
usb.c: new device strings: Mfr=1, Product=2, SerialNumber=0
usb.c: USB device number 2 default language ID 0x409
ov511.c: Device at usb-00:07.2-2 registered to minor 0
usb.c: ov511 driver claimed interface c68afa60
usb.c: kusbdevfs: /sbin/hotplug add 2
usb.c: kusbdevfs policy returned 0xffffffff
```

9.2. video

```
# dmesg | grep video
Linux video capture interface: v1.00
ov511.c: USB OV518 video device found
```

9.3. cam

```
dmesg | grep cam
```

ovsensor.c: v2.21 : OV camera chip I2C driver

pwc Philips PCA645/646 + PCVC675/680/690 + PCVC730/740/750 webcam module version

pwc Also supports the Askey VC010, Logitech Quickcam 3000 Pro, Samsung MPC-C10 a

usb.c: registered new driver Philips webcam

10. Estar seguro que funcionan USB

```
cat /proc/bus/usb/devices
```

```
T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
```

```
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
```

```
D: Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
```

```
P: Vendor=0000 ProdID=0000 Rev= 0.00
```

```
S: Product=USB UHCI-alt Root Hub
```

```
S: SerialNumber=d800
```

```
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
```

```
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
```

```
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Iv1=255ms
```

```
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2
```

```
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0
```

```
D: Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
```

```
P: Vendor=0000 ProdID=0000 Rev= 0.00
```

```
S: Product=USB UHCI-alt Root Hub
```

```
S: SerialNumber=d400
```

```
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
```

```
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
```

```
E: Ad=81(I) Atr=03(Int.) MxPS= 8 Iv1=255ms
```

```
T: Bus=01 Lev=01 Prnt=01 Port=01 Cnt=01 Dev#= 2 Spd=12 MxCh= 0
```

```
D: Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
```

```
P: Vendor=05a9 ProdID=0518 Rev= 1.01
```

```
S: Manufacturer=OmniVision Technologies, Inc.
```

```
S: Product=USB Camera
```

```
C:* #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=500mA
```

```
I: If#= 0 Alt= 0 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
```

```
E: Ad=81(I) Atr=01(Isoc) MxPS= 0 Iv1=1ms
```

```
I: If#= 0 Alt= 1 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
```

```
E: Ad=81(I) Atr=01(Isoc) MxPS= 129 Iv1=1ms
```

```
I: If#= 0 Alt= 2 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
```

```
E: Ad=81(I) Atr=01(Isoc) MxPS= 257 Iv1=1ms
```

```
I: If#= 0 Alt= 3 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
```

```
E: Ad=81(I) Atr=01(Isoc) MxPS= 385 Iv1=1ms
```

```
I: If#= 0 Alt= 4 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
E: Ad=81(I) Atr=01(Isoc) MxPS= 513 Iv1=1ms
I: If#= 0 Alt= 5 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
E: Ad=81(I) Atr=01(Isoc) MxPS= 641 Iv1=1ms
I: If#= 0 Alt= 6 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
E: Ad=81(I) Atr=01(Isoc) MxPS= 769 Iv1=1ms
I: If#= 0 Alt= 7 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=ov511
E: Ad=81(I) Atr=01(Isoc) MxPS= 897 Iv1=1ms
```

```
#cat /proc/bus/usb/drivers
usbdevfs
hub
ov511
Philips webcam
ovfx2
```

11. Comprobar el dispositivo

11.1. /dev/video0

Comprobar que existe /dev/video0.

Si no existe crealo con:

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

```
# ll /dev/video0
crw-rw----  1 paco  paco  81,  0 2003-02-12 02:40 /dev/video0
```

11.2. /dev/video

Algunas aplicaciones buscan el dispositivo /dev/video . En algunos sistemas (ejemplo. RedHat 7.1), /dev/video debe ser un directorio y no un nodo o un enlace simbólico.

Deberemos especificar manualmente el dispositivo, /dev/video0 cuando uses tus aplicaciones de video

Es recomendable hacer este enlace:

```
ln -s /dev/video0 /dev/video
```

```
ll /dev/video
lrwxrwxrwx  1 root  root  6 2003-02-12 02:40 /dev/video -> video0
```


11.3. Dar permisos

Dar a los usuarios permisos para usar la cámara. Este requiere permiso de lectura y escritura a `/dev/video0`.

Puedes dar privilegios con :

```
chmod 666 /dev/video0
```

Dar privilegios a tu usuario para usar la cámara:

Edita como root y añade tu usuario al grupo video

```
nano /etc/group
```

```
video:x:44:usuario
```

12. El sistema de ficheros: fstab

Añadir la línea en `/etc/fstab`

```
none /proc/bus/usb usbdevfs defaults,rw,auto 0 0
```

13. Poniéndola en marcha

Como root:

```
modprobe videodev
modprobe ov511
modprobe ovfx2
modprobe uhci
modprobe ovsensor
```

Como usuario:

```
xawtv
```

14. Para apagar la cámara

```
modprobe -r uhci
```

15. Script de Arranque y parada

```
# cat cam.sh
#!/bin/bash
```

```
case "$1" in
start)
insmod videodev
insmod ov511
insmod ovfx2
insmod uhci
insmod ovsensor
;;
stop)
rmmod -r -s videodev
rmmod -r -s ov511
rmmod -r -s ovfx2
rmmod -r -s uhci
rmmod -r -s ovsensor
;;
*)      echo "Uso: cam.sh start|stop"
        exit 1
        ;;
esac
```

16. Ver si la reconoce

```
$ xawtv -hwscan
This is xawtv-3.82, running on Linux/i586 (2.4.20)
dlopen failed for /usr/lib/libquicktime/lqt_vorbis.so: libvorbisenc.so.0: no se
puede abrir el fichero del objeto compartido: No existe el fichero o el
directoriolooking for available devicesport 83-83
  type : Xvideo, image scaler
  name : 3dfx Video Overlay

/dev/video0: OK                               [ -device /dev/video0 ]
  type : v4l
  name : OV518 USB Camera
  flags: capture
```



Figura 1: foto

17. Una foto

18. Software para webcam

18.1. Instalación

```
apt-get install webcam ftp
```

18.2. Configuración de webcam

Se entiende q tenermos un servidor ftp y web ya funcionando.

Debemos crear en la carpeta del usuario el fichero:

```
nano .webcamrc
```

```
[grab]  
device = /dev/video0
```

```
text = "webcam linux %Y-%m-%d %H:%M:%S"
infofile = filename
#fg_red = 255
#fg_green = 255
#fg_blue = 255
#width = 320
#height = 240
delay = 0
wait = 0
#input = composite1
norm = pal
rotate = 0
top = 0
left = 0
bottom = -1
right = -1
quality = 75
trigger = 0
once = 1
```

```
[ftp]
host = mi_ip
user = "mi_usuario"
pass = "mi_password"
dir = public_html/webcam
file = webcam.jpeg
tmp = uploading.jpeg
passive = 1
debug = 0
auto = 0
local = 0
ssh = 0
```

18.3. Script para automatizar

Este script permite hacer una foto cada x segundos y guardarlos en una carpeta para luego poderse ver con el servidor web.

Debemos crear en la carpeta del usuario paco el fichero:

```
nano webcam.sh
```

```
#!/bin/sh
```

```
# Por Paco Aldarias
# 31-3-03
# Echo Script para webcam
# Crea la carpeta del dia actual
# Copia cada $1 (parametros de entrada) la imagen en la carpeta del dia

d1=public_html/webcam
d2=$d1/$(date +%d-%m-%Y)
mkdir $d2

while :
do
  d3=$d2/$(date +%d-%m-%Y-%k-%M).jpeg
  echo copiando $d1/webcam.jpeg en $d3
  cp $d1/webcam.jpeg $d3
  webcam
  sleep $1
done
```

Le damos permisos de ejecución:
chmod 700 webcam
Asi se llama:
./webcam.sh 120
Podremos ver el resultado asi:
<http://localhost/~paco/webcam/webcam.jpeg>

19. Crear videos: motion

Motion es un programa q captura imagenes y crea un video. Es ideal como camara de seguridad, ya q solo graba los cambios q se produzcan en la imagen. El programa crea carpetas segun el dia, y tambien un video con formato mpeg q recoge todas las imagenes obtenidas.

Instalarlo:
apt-get install motion
Versión actual:
motion 3.0.4-2 V4L Capture Program supporting Movement Dete
Uso:

```
motion -t public_html/motion/ -a 300
```

Esto crea dentro de la carpeta:

```
$HOME/public_html/motion/
```

el video cada 300 segundos comprueba si hay cambios para generar una imagen..

20. Bibliografía

1. Web Oficial de Philips:
www.pcstuff.philips.com
2. Drivers de la cámara
<http://alpha.dyndns.org/ov511/>
3. Usb:
<http://www.linux-usb.org>
<http://bulmalug.net/body.phtml?nIdNoticia=1356>
4. Esta página:
<http://www.iespana.es/pacodebian/webcam.html>
<http://www.iespana.es/pacodebian/webcam.pdf>
<http://www.iespana.es/pacodebian/webcam.tex>
5. Drivers para cámaras : Logitech Quickcam Express Logitech Quickcam
Web LegoCam Dexxa Webcam Labtec Webcam
<http://qce-ga.sourceforge.net/>

21. Agradecimientos

Quisiera agradecer la ayuda a:

1. Mark McClelland. <mailto://mark@alpha.dyndns.org>