Installing drivers for Red Hat 9 using 2.4 kernal

Source: http://linux.derkeiler.com/Newsgroups/comp.os.linux.setup/2006−02/msg00317.html

- *From*: "Andrew" <AndrewA.Reid@xxxxxxxxx>
- *Date*: 15 Feb 2006 19:28:45 −0800

Hey can anyone here help me install these drivers I have for this
webcam. Obviously I am new to this and I am not familier with the terms

in the directions. So if someone can p lease explain these directions
and why I have to do certain things then that would be flipping
awesome. PLEASE HELP ME!!!! BTW IT COULD HELP IF YOU TOLD ME HOW TO
MAKE IT LOAD AUTOMATICALLY IF IT CAN. THANX IN ADVANCE.

I am using a Logitech QuickCAM Express.

%%%%%%%%%%%% INSTRUCTIONS %%%%%%%%%%%%%%%%%%

QUICKSTART
−−−−−−−−−−
Type "./quickcam.sh" in the current directory and follow the
instructions.

Requirements
−−−−−−−−−−−−
An USB camera with ST Microelectronics bridge chip and either VV6410,
HDCS−1000, HDCS−1020 or Photobit sensor. Usually Logitech QuickCams
qualify,
but check if your camera is supported from
http://qce−ga.sourceforge.net.
Important notice: never trust that if camera named "xxxx" is mentioned
as supported, it really is. There are many different cameras sold
with exactly same name/model, with some working and some not. Examples
are QuickCam Notebook and Labtec cameras, of which some work and some
don't.
VendorId and ProductId, as reported by lsusb, help much determining
whether a camera really is supported, but even they aren't failproof.
The VendorId should be 0x046D and ProductId one of 0x0840, 0x0850, or
0x0870 or the camera definitely is not supported by this driver.
Required kernel: Linux 2.2.x (x >= 18), 2.4.x, or 2.6.x.
x86 is best tested, but the driver might also work on other
architectures

(I got a success report on Alpha and PPC). SMP appears to work too.

Where to get it?
−−−−−−−−−−−−−−−
Well, you're already reading this so you obviously should already have
the driver too. But just in case, here's the important links:
http://qce−ga.sourceforge.net
http://sourceforge.net/projects/qce−ga
http://sourceforge.net/project/showfiles.php?group_id=12924
To get the latest version under development, use anonymous CVS
at http://qce−ga.sourceforge.net.
Here may be also some interesting links and information:
http://www.ee.oulu.fi/~tuukkat/quickcam/quickcam.html

How to install?
−−−−−−−−−−−−−−
The recommended way is now just to run the script "./quickcam.sh" in
the driver source directory. In some cases you might want to run
make manually, as described in the following:
Type "make clean && make all" to compile the driver. The
resulting driver file is "quickcam.o" for kernels 2.2.x and 2.4.x,
or "quickcam.ko" for kernel 2.6.x. Type "make" to get some installation

options. Especially you should consider disabling debugging, it will
make
the driver about 30% smaller and faster (this is now the default). And
if
compressed mode is not supported with your camera, you might want to
compile the driver with "make all USER_OPT=−DCOMPRESS=0", which will
save some more bytes.
Note: /lib/modules/x.y.zz/build should be a symbolic link to
your
kernel source. It is created automatically by "make modules_install" in

kernel source directory, but some distributions (such as Debian) might
not
have it pre−installed correctly. You can make the link manually or by
specifying alternate kernel source directory with
LINUX_DIR=/usr/src/x.y.zz.
You can see the kernel version that you are running with "uname −r".
The kernel source must be configured (with "make menuconfig",
for
example) to have the same options as the kernel you will be running
with the
camera. Also "make dep" must be performed in the kernel source
directory
before the camera can be compiled against it. One way is to download
fresh
kernel source into /usr/src/linux−x.y.zz, copy .config into the

directory,
and then enter "make oldconfig && make dep". Then you can compile the
camera driver against the kernel source using the LINUX_DIR option, and

compile the kernel with "make bzImage && make modules", install the
kernel with "make modules_install" and running lilo, and install the
camera
driver with "MODULE_DIR=/lib/modules/x.y.zz make install", and reboot.
With kernel 2.6.x you must use "make modules_prepare" instead of "make
dep".
You should also compile the kernel and the camera driver with same
versions
of gcc. Check gcc version with "gcc −v" and the version which was used
for compiling the kernel with "cat /proc/version".
Easier is just to install the kernel source package from your
Linux
distribution and compile then the camera driver. For example, in Debian

you should first check which "kernel−image" package is installed, and
then install the corresponding "kernel−headers" package, which is
correctly preconfigured (at least on x86).
The driver is accessed using either /dev/video or /dev/videoX
device file. If you are using devfs, it should appear automatically
when
the driver is loaded. If not, you can create the file manually with
mknod /dev/video0 c 81 0
mknod /dev/video1 c 81 1
chmod a+rw /dev/video0
ln −s /dev/video0 /dev/video
as root.
If you still don't have a clue, ask from
qce−ga−discuss...@xxxxxxxxxxxxxxxxxxxxxx

How to load?
−−−−−−−−−−−
You need USB and Video4Linux support in your kernel. Check that they
are compiled, if not, reconfigure kernel, compile and install. It took
a long time to find the Video4Linux checkbox in 2.2, it is in
Character Devices / Video4Linux
When you have a supported kernel, just load the necessary drivers with
modprobe videodev
modprobe usb−uhci # Either UHCI or OHCI,
modprobe usb−ohci # loading the another will fail
insmod ./quickcam.o(.ko) compatible=3
(or "modprobe quickcam" if you have already installed it).
If it works, you can install these lines into some startup script
in /etc to load the driver every time you boot up your computer.
If you're using hotplug (package "hotplug" on Debian), the
driver
should be loaded automatically (in 2.4.x and newer kernels) when you

plug
the camera in (after you have installed the driver with "make
install").
For hotplug to work, the UHCI or OHCI driver has to be first loaded,
on Debian typically by inserting the correct driver name into
"/etc/modules".

How to run?
−−−−−−−−−−
After the driver has been loaded, you can run your favorite camera
software
such as Xawtv, Motv, or Mplayer/Mencoder (see file APPLICATIONS for
some
suggestions). If you have multiple video devices, do
cd /proc/video/quickcam
ls −la
to see which devices are allocated for the qc−usb driver. You can
examine
the contents of the video* files in the above directory by using e.g.
"cat".
Specify /dev/video* device file corresponding to the desired camera
to the program what you are using.
There's an utility called "qcset" that you can use to change
driver options on−the−fly. Run "qcset −h" for more info. It is
recommended
that you use qcset to change driver options instead of using module
parameters
when loading the module with insmod. The module options might go away
in
future, and you can set different options for each plugged camera only
with
qcset. Check out the possible module parameters with "qcset −h".
If some piece of program doesn't work, try enabling more compatibility
levels:
qcset compatible=16x,dblbuf
By default all compatibility is disabled, because it is the user
applications
not the driver which should be fixed. Meanwhile, use either
compatible=1
or 3.
Some programs that require compatible=1: anything using the Xvideo
driver.
Some programs that require compatible=3: motion, vic.
Other available options with qcset:
−b and −c Set camera exposure time and gain, when automatic
adaptation
is disabled. When enabled, −b selects the target image
brightness, into which the adaptation algorithm tries
to
settle.

−o and −u Set camera color balance. If the color strength (−o) is

less than half of the maximum (the default), all colors

have the equal gain and hue (−u) has no effect.
Otherwise,
the relative color gains are balanced according to hue
and color strength. These options have effect only with

the HDCS and Photobit sensors, and on Photobit only
when
automatic exposure control has been disabled
(adaptive=n).
−w Set image sharpness. This option has effect only when
image quality is 5 (best). Zero value corresponds
to quality 3 (bilinear). Also, this option has no
effect
when compression is enabled.
−g v Enable lookup−table with the given gamma value
−g rg:gg:bg Specify different gamma value for each channel
−g rg:gg:bg:rw:gw:bw Specify also color gain for each color channel
Gamma and color correction is useful if your sensor
does
not support hardware correction with options −o and −u.

The gamma values should be between 0.5...1.0 and the
balance weights around 0.7−1.3. You can but probably
should
not use equalization simultaneously with software
lookup−table,
because the advantage is dubious.
Software lookup table will have no effect when
compression is
enabled (compress=y).
−e filename.pnm Compute and enable static equalization from given
PNM/PPM
image file. Similar to "equalize=y", except that the
equalization lookup table is calculated from given
image
file only once, and not constantly. This avoids
problems
with flat surfaces or other views that do not contain
balanced colors (e.g. if you would point the camera at
full−red car, equalization would make it not to look
red
at all).
How to generate own image for equalization:
− qcset equalize=n quality=bilinear −g −
− xawtv −noxv −noscale (disabling Xvideo is required)
− Point the camera into something that contains good
balance

of all primary colors, red, green, and blue.
Something like
pale walls are not good, try a bookshelf or
something.
− Run GIMP, select "File/Acquire/Screen Shot".
Enable "Single Window", disable "Decorations", click
OK.
− Click on the Xawtv window. If you get completely blue

image instead of what xawtv displays, then you didn't

disable Xvideo.
− Click on the GIMP image window with right mouse
button,
select "File/Save As...", enter filename.pnm, select
raw PNM file.
− qcset quality=best −e filename.pnm
The equalization image is sensor and lighting−specific,
so if
you change the camera or go indoors/outdoors, redo the
procedure above.
There may be some pre−made equalization files in
http://www.ee.oulu.fi/~tuukkat/quickcam/eq−*.pmn.
This option can not be use simultaneously with the "−g"
option.
You can disable static equalization with "−g −".
Software lookup table will have no effect when
compression is
enabled (compress=y).
−g + / −e + Enable software−based lookup−table for correcting
colors
−g − / −e − Disable lookup−table
−g '?' / −e '?' Display whether lookup−table is enabled and the table
contents
qcdebug Enable selected debug messages. You can write symbolic
list
of error message types or bitmask, for example
"qcdebug=errors,common" displays debug messages related
to
various errors and some common messages as "frame
lost".
The kernel messages are logged typically into file
/var/log/kern.log.
Note that if the driver was not compiled with debugging

enabled ("USER_OPT=−DDEBUG") this option will not do
much.
debug Same as 'qcdebug', for compatibility with older
versions.
keepsettings Keep image settings over camera uses. When this option
is disabled, the image settings (brightness etc.)

are reset to the default values each time when a
program
opens the camera.
settle Specify the maximum number of frames to wait for image
brightness to settle before starting capturing. Has
effect only when adaptation is enabled. For example,
if settle=30, the camera driver waits maximum of 30
frames until automatic adaptation (brightness control)
has stabilized the image brightness and only then gives

the first image. Some programs, e.g. xawtv, don't like
if the driver pauses for a long time.
This option is particularly useful if you take single
image snapshots but the images are too dark or bright.
subsample Increases framerate but decreases image resolution.
May work or not to work depending on camera sensor
type.
Works on HDCS−1000. To get high frame rate, you must
use
short exposure time. Either disable adaptation and set
contrast into small value or use enough light so that
the
adaptation routine uses short exposure time.
compress Enable MJPEG compression with QuickCam Web and LEGO
cameras.
On other cameras, compression is automatically
disabled.
When using compression, image size is fixed to 320*240
and frame rate is about doubled to 15 frames per
second.
frameskip Lower frame rate by skipping frames. Decreases CPU
power requirement. For slow computers.
quality Select image conversion routine which is used for
converting Bayer CFA image into RGB image. Has no
effect when compression is enabled.
adaptive Enable or disable automatic brightness control.
Use −b to set the target brightness.
equalize Equalizes the image. May improve colors hugely
sometimes.
userlut Enables or disables the default whether to use software

lookup−table color correction in the driver. Affects
only
cameras which are plugged in after setting this
setting.
For already plugged−in cameras, use −g and −e options.
retryerrors If there are problems with communicating with the
camera,
and retryerrors is enabled, the driver tries to retry
a few times instead of immediately failing and
returning

error to application.
compatible Enable various compatibility levels:
16x: force image width and height be multiple of
sixteen.
dblbuf: lie applications about the number of frame
buffers.
torgb: convert BGR values into RGB.
video_nr Select the video device number to use. Can not be set
with qcset.
You can query the named parameters by adding a question mark, for
example
qcset 'qcdebug?'
displays the current debugging selection.
With kernel 2.6.x you can run qcset simultaneously while running other
applications which use the camera. With older kernels, you must close
other applications before qcset can be used.

.