libwebcam Reference Manual

1.2

Generated by Doxygen 1.3-rc3

Sat May 10 01:01:12 2003

Contents

Chapter 1

libwebcam Compound Index

1.1 libwebcam Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

libwebcam File Index

2.1 libwebcam File List

Here is a list of all files with brief descriptions:

Chapter 3

libwebcam Class Documentation

3.1 BMPHEAD Struct Reference

Definition of a structure specific for this driver (see declaration in camera.h).

Public Attributes

- char id [2]
- long filesize
- short reserved [2]
- long headersize
- long infosize
- long width
- long depth
- short biplanes
- short bits
- long bicompression
- long bisizeimage
- long bixpelspermeter
- long biypelspermeter
- long biclrused
- long biclrimportant

3.1.1 Detailed Description

Definition of a structure specific for this driver (see declaration in camera.h).

3.1.2 Member Data Documentation

3.1.2.1 long BMPHEAD::biclrimportant

3.1.2.2 long BMPHEAD::biclrused

3.1.2.3 long BMPHEAD::bicompression

3.1.2.4 short BMPHEAD::biplanes

3.1.2.5 long BMPHEAD::bisizeimage

3.1.2.6 short BMPHEAD::bits

3.1.2.7 long BMPHEAD::bixpelspermeter

3.1.2.8 long BMPHEAD::biypelspermeter

3.1.2.9 long BMPHEAD::depth

3.1.2.10 long BMPHEAD::filesize

3.1.2.11 long BMPHEAD::headersize

3.1.2.12 char BMPHEAD::id[2]

3.1.2.13 long BMPHEAD::infosize

3.1.2.14 short BMPHEAD::reserved[2]

3.1.2.15 long BMPHEAD::width

The documentation for this struct was generated from the following file:

- camera.c

3.2 camprop Struct Reference

structure qui accueille les parametres.

`#include <camera.h>`

Public Attributes

- COMMON_CAMSTRUCT int imax
- int jmax
- int driver
- int longuepose
- char longueposestart
- char longueposestop
- char webcamDevice [128]

  webcam device (only for Linux).

- char longExposureDevice [128]

  long exposure device.

- int validFrame

  Valid image number (used under Linux).

- int cam_fd

  cam_fd, webcam device file descriptor.

- int long_fd

  long_fd, long exposure device file descriptor.

- unsigned char * rgbBuffer

  Buffer for rgb frame.

- int rgbBufferSize

  rgbBufferSize is size in bytes of rgbBuffer.

- unsigned char * yuvBuffer

  Buffer for yuv frame.

- int yuvBufferSize

  yuvBufferSize is size in bytes of yuvBuffer.

- int shutterSpeed

  shutterSpeed remember the shutter speed.

3.2.1 Detailed Description

structure qui accueille les parametres.

Structure which contains camera's parameters.

- COMMON_CAMSTRUCT - standard parameters, don't change.

3.2.2 Member Data Documentation

3.2.2.1 int camprop::cam_fd

cam_fd, webcam device file descriptor.

3.2.2.2 int camprop::driver

3.2.2.3 COMMON_CAMSTRUCT int camprop::imax

3.2.2.4 int camprop::jmax

3.2.2.5 int camprop::long_fd

long_fd, long exposure device file descriptor.

3.2.2.6 char camprop::longExposureDevice[128]

long exposure device.

default:

- Linux - ”/dev/parport0”
- Windows - ”lpt1”

3.2.2.7 int camprop::longuepose

3.2.2.8 char camprop::longueposestart

3.2.2.9 char camprop::longueposestop

3.2.2.10 unsigned char* camprop::rgbBuffer

Buffer for rgb frame.

Used under Linux for keeping rgb frame

In cam_init memory is allocated and dislocated in cmdCamClose, also video format functions change buffers sizes and allocates new memory for them.

3.2.2.11 int camprop::rgbBufferSize

rgbBufferSize is size in bytes of rgbBuffer.

3.2.2.12 int camprop::shutterSpeed

shutterSpeed remember the shutter speed.

A negative value sets the shutter speed to automatic (controlled by the camera’s firmware). A value of 0..65535 will set manual mode, where the values have been calibrated such that 65535 is the longest

possible exposure time that I could find on any camera model. It is not a linear scale, where a value of '1' is 1/65536th of a second, etc.

Used under Linux.

3.2.2.13 int camprop::validFrame

Valid image number (used under Linux).

Pwc kernel module has some buffers and when you take long exposure you need to find which buffer contains your frame.

This parameter says which frame is your valid frame (how many read() calls you need), if is 0 auto detection is performed (less then 20 read() calls).

- dafault: 3

3.2.2.14 char camprop::webcamDevice[128]

webcam device (only for Linux).

uses pwc and pwcx modules

- default: /dev/video0

3.2.2.15 unsigned char* camprop::yuvBuffer

Buffer for yuv frame.

Used under Linux for keeping yuv frame

3.2.2.16 int camprop::yuvBufferSize

yuvBufferSize is size in bytes of yuvBuffer.

The documentation for this struct was generated from the following file:

- camera.h

3.3 pwc_leds Struct Reference

`#include <pwc-ioctl.h>`

Public Attributes

- int led_on
- int led_off

3.3.1 Member Data Documentation

3.3.1.1 int pwc_leds::led_off

3.3.1.2 int pwc_leds::led_on

The documentation for this struct was generated from the following file:

- pwc-ioctl.h

3.4 pwc_probe Struct Reference

`#include <pwc-ioctl.h>`

Public Attributes

- char name [32]
- int type

3.4.1 Member Data Documentation

3.4.1.1 char pwc_probe::name[32]

3.4.1.2 int pwc_probe::type

The documentation for this struct was generated from the following file:

- pwc-ioctl.h

3.5 pwc_wb_speed Struct Reference

`#include <pwc-ioctl.h>`

Public Attributes

- int control_speed
- int control_delay

3.5.1 Member Data Documentation

3.5.1.1 int pwc_wb_speed::control_delay

3.5.1.2 int pwc_wb_speed::control_speed

The documentation for this struct was generated from the following file:

- pwc-ioctl.h

3.6 pwc_whitebalance Struct Reference

`#include <pwc-ioctl.h>`

Public Attributes

- int mode
- int manual_red
- int manual_blue
- int read_red
- int read_blue

3.6.1 Member Data Documentation

3.6.1.1 int pwc_whitebalance::manual_blue

3.6.1.2 int pwc_whitebalance::manual_red

3.6.1.3 int pwc_whitebalance::mode

3.6.1.4 int pwc_whitebalance::read_blue

3.6.1.5 int pwc_whitebalance::read_red

The documentation for this struct was generated from the following file:

- pwc-ioctl.h

3.7 ScanStruct Struct Reference

`#include <camtcl.h>`

Public Attributes

- char * dateobs
- char * dateend
- ClientData clientData
- Tcl_Interp * interp
- Tcl_TimerToken TimerToken
- int width
- int offset
- int height
- int bin
- float dt
- int y
- unsigned long t0
- unsigned short * pix
- unsigned short * pix2
- int last_delta
- int blocking
- int keep_perfos
- int fileima
- FILE * fima
- int * dts
- unsigned long loopmilli1
- int stop
- double tumoinstl
- double ra
- double dec

3.7.1 Member Data Documentation

3.7.1.1 int ScanStruct::bin

3.7.1.2 int ScanStruct::blocking

3.7.1.3 ClientData ScanStruct::clientData

3.7.1.4 char* ScanStruct::dateend

3.7.1.5 char* ScanStruct::dateobs

3.7.1.6 double ScanStruct::dec

3.7.1.7 float ScanStruct::dt

3.7.1.8 int* ScanStruct::dts

3.7.1.9 int ScanStruct::fileima

3.7.1.10 FILE* ScanStruct::fima

3.7.1.11 int ScanStruct::height

3.7.1.12 Tcl_Interp* ScanStruct::interp

3.7.1.13 int ScanStruct::keep_perfos

3.7.1.14 int ScanStruct::last_delta

3.7.1.15 unsigned long ScanStruct::loopmilli1

3.7.1.16 int ScanStruct::offset

3.7.1.17 unsigned short* ScanStruct::pix

3.7.1.18 unsigned short* ScanStruct::pix2

3.7.1.19 double ScanStruct::ra

3.7.1.20 int ScanStruct::stop

3.7.1.21 unsigned long ScanStruct::t0

3.7.1.22 Tcl_TimerToken ScanStruct::TimerToken

3.7.1.23 double ScanStruct::tumoinstl

3.7.1.24 int ScanStruct::width

3.7.1.25 int ScanStruct::y

The documentation for this struct was generated from the following file:

- camtcl.h

Chapter 4

libwebcam File Documentation

4.1 camcmd.h File Reference

Common commands for all cameras and specific commands for that camera.

Variables

- cmditem cmdlist [ ]

4.1.1 Detailed Description

Common commands for all cameras and specific commands for that camera.

4.1.2 Variable Documentation

4.1.2.1 struct cmditem cmdlist[ ] [static]

Initial value:

```
{

  COMMON_CMDLIST

  {"close", cmdCamClose},
  {"videosource", cmdCamVideoSource},
  {"videoformat", cmdCamVideoFormat},
  {"snap", cmdCamSnap},
  {"snaprgb", cmdCamSnapRgb},
  {"longuepose", cmdCamLonguePose},
  {"longueposeport", cmdCamLonguePosePortAdress},
  {"longueposestartvalue", cmdCamLonguePoseStartValue},
  {"longueposestopvalue", cmdCamLonguePoseStopValue},
```

```
    {NULL, NULL}
}
```

4.2 camera.c File Reference

This file contains camera's "driver".

```
#include "sysexp.h"
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <stdio.h>
#include "camera.h"
#include "util.h"
```

Compounds

- struct BMPHEAD

  Definition of a structure specific for this driver (see declaration in camera.h).

Functions

- void libcam_get_tel_coord (Tcl_Interp *interp, double *ra, double *dec, struct camprop *cam, int *status)
- void libcam_GetCurrentFITSDate (Tcl_Interp *interp, char *s)
- void libcam_GetCurrentFITSDate_function (Tcl_Interp *interp, char *s, char *function)
- int cam_init (struct camprop *cam, int argc, char **argv)
- void cam_start_exp (struct camprop *cam, char *amplionoff)

  Function cam_start_exp - starts the exposure.

- int cam_stop_longexposure (struct camprop *cam)

  cam_stop_longexposure stops long exposure.

- void cam_stop_exp (struct camprop *cam)
- void cam_read_ccd (struct camprop *cam, unsigned short *p)

  cam_read_ccd - reads a frame.

- void cam_shutter_on (struct camprop *cam)
- void cam_shutter_off (struct camprop *cam)
- void cam_ampli_on (struct camprop *cam)
- void cam_ampli_off (struct camprop *cam)
- void cam_measure_temperature (struct camprop *cam)
- void cam_cooler_on (struct camprop *cam)
- void cam_cooler_off (struct camprop *cam)
- void cam_cooler_check (struct camprop *cam)
- void cam_set_binning (int binx, int biny, struct camprop *cam)
- void cam_set_exptime (float exptime, struct camprop *cam)

  Function cam_set_exptim.

- void cam_update_window (struct camprop *cam)
- short loadbmp24bw (char *nom, unsigned short *buf, struct camprop *cam)
- short loadbmp24rgb (char *nom, unsigned short *bufrgb, struct camprop *cam)
- int snap (struct camprop *cam, int rgb)

  snap - reads frame and stories it in (libaudela) buffer.

- int videoformat (struct camprop *cam, char *formatname)

  videoformat - sets video format.

- void ng_color_yuv2rgb_init (void)

  Init Lookup tables for yuv to rgb conversion.

- void yuv420p_to_rgb24 (unsigned char *yuv, unsigned char *rgb, int width, int height)

  Convert from yuv to rgb.

- int setLongExposureDevice (struct camprop *cam, unsigned char value)

  setLongExposureDevice - writes value *to the parallel port.*

- int initLongExposureDevice (struct camprop *cam)

  initLongExposureDevice - initiates a long exposure device and sets cam->longueposestop.

- int readFrame (struct camprop *cam, unsigned char *rgbBuffer)

  readFrame - reads one frame from webcam and stores it in cam->rgbBuffer.

- int getVideoSource (struct camprop *cam, char *result, int command)

  getVideoSource - returns asked parameters.

- int saveUser (struct camprop *cam)

  saveUser.

- int setPicSettings (struct camprop *cam, int brightness, int contrast, int colour, int whiteness)

  setPicSettings - sets brightness, contrast, colour and whiteness (gamma).

- int setVideoSource (struct camprop *cam, int paramValue, int command)

  setVideoSource - sets some video source parameters.

- int setWhiteBalance (struct camprop *cam, char *mode, int red, int blue)

  setWhiteBalance sets White Balance.

Variables

- camini cam_ini [ ]

  Definition of different cameras supported by this driver (see declaration in libstruc.h).

4.2.1 Detailed Description

This file contains camera's "driver".

Structure "camprop" can be adapted from file camera.h

Ceci est le fichier contenant le driver de la camera.

La structure "camprop" peut etre adaptee dans le fichier camera.h

4.2.2 Function Documentation

4.2.2.1 void cam_ampli_off (struct camprop * *cam*)

4.2.2.2 void cam_ampli_on (struct camprop * *cam*)

4.2.2.3 void cam_cooler_check (struct camprop * *cam*)

4.2.2.4 void cam_cooler_off (struct camprop * *cam*)

4.2.2.5 void cam_cooler_on (struct camprop * *cam*)

4.2.2.6 int cam_init (struct camprop * *cam*, int *argc*, char ** *argv*)

cam_init

- cam_init permet d'initialiser les variables de la
- structure 'camprop'
- specifiques a cette camera.

cam_init

- cam_init initialize variables of structure "camprop" specified for this camera.

under Linux it opens webcam and parallel port

- sets image format on 640 x 480 (max for this camera).

4.2.2.7 void cam_measure_temperature (struct camprop * *cam*)

4.2.2.8 void cam_read_ccd (struct camprop * *cam*, unsigned short * *p*)

cam_read_ccd - reads a frame.

This function store the frame in (unsigned short *)p buffer.

Under:

- Linux rgbBuffer is copied to p,
- Windows @0.bmp file is read and copied to p.

Calling diagram:

”acq” -> cmdCamAcq -> cam_start_exp ...

-> AcqRead -> cam_read_ccd

or

”stop” -> cmdCamStop -> AcqRead -> cam_read_ccd.

should return a value???

4.2.2.9 void cam_set_binning (int *binx*, int *biny*, struct camprop * *cam*)

4.2.2.10 void cam_set_exptime (float *exptime*, struct camprop * *cam*)

Function cam_set_exptim.

Probably never used... ???

4.2.2.11 void cam_shutter_off (struct camprop * *cam*)

4.2.2.12 void cam_shutter_on (struct camprop * *cam*)

4.2.2.13 void cam_start_exp (struct camprop * *cam*, char * *amplionoff*)

Function cam_start_exp - starts the exposure.

Called by command ”acq” (function: cmdCamAcq), after **exptime** TCL calls cam_read_ccd (function: AcqRead).

”acq” -> cmdCamAcq -> cam_start_exp ...

-> AcqRead -> cam_read_ccd

or

”stop” -> cmdCamStop -> AcqRead -> cam_read_ccd.

should return a value???

4.2.2.14 void cam_stop_exp (struct camprop * *cam*)

4.2.2.15 int cam_stop_longexposure (struct camprop * *cam*)

cam_stop_longexposure stops long exposure.

Under Linux this function reads a frame and store it in cam->rgbBuffer, under Windows it saves frame in file ”@0.bmp”.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.16 void cam_update_window (struct camprop * *cam*)

4.2.2.17 int getVideoSource (struct camprop * *cam*, char * *result*, int *command*)

getVideoSource - returns asked parameters.

command is defined by *command* , result is copied to *result* string,

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.18 int initLongExposureDevice (struct camprop * *cam*)

initLongExposureDevice - initiates a long exposure device and sets cam->longueposestop.

Parallel port control:

- Linux uses parport, parport_pc and ppdev modules.
- Windows uses "lpt1" printer port (with its handshake), so you will need "null printer" modified plug. If you don't like to use "lpt1" printer port you can define OS_WIN_USE_LPT_OLD_STYLE.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.19 void libcam_get_tel_coord (Tcl_Interp * *interp*, double * *ra*, double * *dec*, struct camprop * *cam*, int * *status*)

4.2.2.20 void libcam_GetCurrentFITSDate (Tcl_Interp * *interp*, char * *s*)

4.2.2.21 void libcam_GetCurrentFITSDate_function (Tcl_Interp * *interp*, char * *s*, char * *function*)

4.2.2.22 short loadbmp24bw (char * *nom*, unsigned short * *buf*, struct camprop * *cam*)

4.2.2.23 short loadbmp24rgb (char * *nom*, unsigned short * *bufrgb*, struct camprop * *cam*)

4.2.2.24 void ng_color_yuv2rgb_init (void)

Init Lookup tables for yuv to rgb conversion.

Code comes from xawtv.

4.2.2.25 int readFrame (struct camprop * *cam*, unsigned char * *rgbBuffer*)

readFrame - reads one frame from webcam and stores it in cam->rgbBuffer.

If longexposure is set, function looks for valid frame.

Function is implemented only for Linux.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.26 int saveUser (struct camprop * *cam*)

saveUser.

This function will write the current brightness, contrast, colour and whiteness (gamma) settings into the camera's internal EEPROM.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.2.2.27 int setLongExposureDevice (struct camprop * *cam*, unsigned char *value*)

setLongExposureDevice - writes *value* to the parallel port.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.28 int setPicSettings (struct camprop * *cam*, int *brightness*, int *contrast*, int *colour*, int *whiteness*)

setPicSettings - sets brightness, contrast, colour and whiteness (gamma).

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.2.2.29 int setVideoSource (struct camprop * *cam*, int *paramValue*, int *command*)

setVideoSource - sets some video source parameters.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.2.2.30 int setWhiteBalance (struct camprop * *cam*, char * *mode*, int *red*, int *blue*)

setWhiteBalance sets White Balance.
Arguments:

- mode - mode name
- red, blue - red and blue levels - valid only when mode is ”manual”

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.2.2.31 int snap (struct camprop * *cam*, int *rgb*)

snap - reads frame and stories it in (libaudela) buffer.

4.2.2.32 int videoformat (struct camprop * *cam*, char * *formatname*)

videoformat - sets video format.
Possible format names:

- VGA - 640 x 480
- CIF - 352 x 288
- SIF - 320 x 240
- SSIF - 240 x 176
- QCIF - 176 x 144
- QSIF - 160 x 120
- SQCIF - 128 x 96.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.2.2.33 void yuv420p_to_rgb24 (unsigned char * *yuv*, unsigned char * *rgb*, int *width*, int *height*)

Convert from yuv to rgb.
Code comes from xawtv, actually it converts to bgr and flips vertically.

4.2.3 Variable Documentation

4.2.3.1 struct camini cam_ini[ ]

Definition of different cameras supported by this driver (see declaration in libstruc.h).

4.3 camera.h File Reference

Adaptation of your preferred camera functions and especially structure ”camprop”.

```
#include "tcl.h"
#include "libstruc.h"
```

Compounds

- struct camprop

  structure qui accueille les parametres.

Defines

- #define OS_WIN_USE_LPT_OLD_STYLE

  If you define OS_WIN_USE_LPT_OLD_STYLE, you will use libcam_out function with your lpt port, this function doesn146t work under WinXP and others WinNT systems.

- #define VALID_FRAME 3

  Default value of cam->validFrame parameter.

- #define REQUIRED_MAX_VALUE 150

  Frame with any pixel > REQUIRED_MAX_VALUE is detected as valid frame (used in autodetection mode).

Typedefs

- typedef float TYPE_PIXELS

  Type of pixels variables.

Functions

- int cam_init (struct camprop *cam, int argc, char **argv)
- void cam_update_window (struct camprop *cam)
- void cam_start_exp (struct camprop *cam, char *amplionoff)

  Function cam_start_exp - starts the exposure.

- void cam_stop_exp (struct camprop *cam)
- void cam_read_ccd (struct camprop *cam, unsigned short *p)

  cam_read_ccd - reads a frame.

- void cam_shutter_on (struct camprop *cam)
- void cam_shutter_off (struct camprop *cam)
- void cam_ampli_on (struct camprop *cam)
- void cam_ampli_off (struct camprop *cam)
- void cam_measure_temperature (struct camprop *cam)

- void cam_cooler_on (struct camprop *cam)
- void cam_cooler_off (struct camprop *cam)
- void cam_cooler_check (struct camprop *cam)
- void cam_set_binning (int binx, int biny, struct camprop *cam)
- void cam_set_exptime (float exptime, struct camprop *cam)

  Function cam_set_exptim.

- int snap (struct camprop *cam, int rgb)

  snap - reads frame and stories it in (libaudela) buffer.

- short loadbmp24bw (char *nom, unsigned short *buf, struct camprop *cam)
- short loadbmp24rgb (char *nom, unsigned short *bufrgb, struct camprop *cam)
- int videoformat (struct camprop *cam, char *formatname)

  videoformat - sets video format.

- void libcam_strupr (char *chainein, char *chaineout)
- void yuv420p_to_rgb24 (unsigned char *yuv, unsigned char *rgb, int width, int height)

  Convert from yuv to rgb.

- void ng_color_yuv2rgb_init (void)

  Init Lookup tables for yuv to rgb conversion.

- int setLongExposureDevice (struct camprop *cam, unsigned char value)

  setLongExposureDevice - writes value *to the parallel port.*

- int initLongExposureDevice (struct camprop *cam)

  initLongExposureDevice - initiates a long exposure device and sets cam->longueposestop.

- int readFrame (struct camprop *cam, unsigned char *rgbBuffer)

  readFrame - reads one frame from webcam and stores it in cam->rgbBuffer.

- int cam_stop_longexposure (struct camprop *cam)

  cam_stop_longexposure stops long exposure.

- int getVideoSource (struct camprop *cam, char *result, int command)

  getVideoSource - returns asked parameters.

- int saveUser (struct camprop *cam)

  saveUser.

- int setPicSettings (struct camprop *cam, int brightness, int contrast, int colour, int whiteness)

  setPicSettings - sets brightness, contrast, colour and whiteness (gamma).

- int setVideoSource (struct camprop *cam, int paramValue, int command)

  setVideoSource - sets some video source parameters.

- int setWhiteBalance (struct camprop *cam, char *mode, int red, int blue)

  setWhiteBalance sets White Balance.

4.3.1 Detailed Description

Adaptation of your preferred camera functions and especially structure ”camprop”.

Adapter le contenu de ce fichier a votre camera preferee notamment la structure ”camprop”.

4.3.2 Define Documentation

4.3.2.1 #define OS_WIN_USE_LPT_OLD_STYLE

If you define OS_WIN_USE_LPT_OLD_STYLE, you will use libcam_out function with your lpt port, this function doesn146t work under WinXP and others WinNT systems.

If it is not defined, lpt port will be used like printer port, so you will need ”null printer” modified plug.

4.3.2.2 #define REQUIRED_MAX_VALUE 150

Frame with any pixel > REQUIRED_MAX_VALUE is detected as valid frame (used in autodetection mode).

4.3.2.3 #define VALID_FRAME 3

Default value of cam->validFrame parameter.

4.3.3 Typedef Documentation

4.3.3.1 typedef float TYPE_PIXELS

Type of pixels variables.

Should be the same as in cbuffer.h (libaudela).

4.3.4 Function Documentation

4.3.4.1 void cam_ampli_off (struct camprop * *cam*)

4.3.4.2 void cam_ampli_on (struct camprop * *cam*)

4.3.4.3 void cam_cooler_check (struct camprop * *cam*)

4.3.4.4 void cam_cooler_off (struct camprop * *cam*)

4.3.4.5 void cam_cooler_on (struct camprop * *cam*)

4.3.4.6 int cam_init (struct camprop * *cam*, int *argc*, char ** *argv*)

cam_init

- cam_init permet d’initialiser les variables de la
- structure ’camprop’

- specifiques a cette camera.

cam_init

- cam_init initialize variables of structure ”camprop” specified for this camera.

under Linux it opens webcam and parallel port

- sets image format on 640 x 480 (max for this camera).

4.3.4.7 void cam_measure_temperature (struct camprop * *cam*)

4.3.4.8 void cam_read_ccd (struct camprop * *cam*, unsigned short * *p*)

cam_read_ccd - reads a frame.

This function store the frame in (unsigned short *)p buffer.

Under:

- Linux rgbBuffer is copied to p,
- Windows @0.bmp file is read and copied to p.

Calling diagram:

”acq” -> cmdCamAcq -> cam_start_exp ...

-> AcqRead -> cam_read_ccd

or

”stop” -> cmdCamStop -> AcqRead -> cam_read_ccd.

should return a value???

4.3.4.9 void cam_set_binning (int *binx*, int *biny*, struct camprop * *cam*)

4.3.4.10 void cam_set_exptime (float *exptime*, struct camprop * *cam*)

Function cam_set_exptim.

Probably never used... ???

4.3.4.11 void cam_shutter_off (struct camprop * *cam*)

4.3.4.12 void cam_shutter_on (struct camprop * *cam*)

4.3.4.13 void cam_start_exp (struct camprop * *cam*, char * *amplionoff*)

Function cam_start_exp - starts the exposure.

Called by command ”acq” (function: cmdCamAcq), after **exptime** TCL calls cam_read_ccd (function: AcqRead).

”acq” -> cmdCamAcq -> cam_start_exp ...

-> AcqRead -> cam_read_ccd

or

”stop” -> cmdCamStop -> AcqRead -> cam_read_ccd.

should return a value???

4.3.4.14 void cam_stop_exp (struct camprop * *cam*)

4.3.4.15 int cam_stop_longexposure (struct camprop * *cam*)

cam_stop_longexposure stops long exposure.

Under Linux this function reads a frame and store it in cam->rgbBuffer, under Windows it saves frame in file ”@0.bmp”.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.16 void cam_update_window (struct camprop * *cam*)

4.3.4.17 int getVideoSource (struct camprop * *cam*, char * *result*, int *command*)

getVideoSource - returns asked parameters.

command is defined by *command* , result is copied to *result* string,

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.18 int initLongExposureDevice (struct camprop * *cam*)

initLongExposureDevice - initiates a long exposure device and sets cam->longueposestop.

Parallel port control:

- Linux uses parport, parport_pc and ppdev modules.
- Windows uses ”lpt1” printer port (with its handshake), so you will need ”null printer” modified plug. If you don’t like to use ”lpt1” printer port you can define OS_WIN_USE_LPT_OLD_STYLE.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.19 void libcam_strupr (char * *chainein*, char * *chaineout*)

4.3.4.20 short loadbmp24bw (char * *nom*, unsigned short * *buf*, struct camprop * *cam*)

4.3.4.21 short loadbmp24rgb (char * *nom*, unsigned short * *bufrgb*, struct camprop * *cam*)

4.3.4.22 void ng_color_yuv2rgb_init (void)

Init Lookup tables for yuv to rgb conversion.

Code comes from xawtv.

4.3.4.23 int readFrame (struct camprop * *cam*, unsigned char * *rgbBuffer*)

readFrame - reads one frame from webcam and stores it in cam->rgbBuffer.

If longexposure is set, function looks for valid frame.

Function is implemented only for Linux.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.24 int saveUser (struct camprop * *cam*)

saveUser.

This function will write the current brightness, contrast, colour and whiteness (gamma) settings into the camera's internal EEPROM.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.3.4.25 int setLongExposureDevice (struct camprop * *cam*, unsigned char *value*)

setLongExposureDevice - writes *value* to the parallel port.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.26 int setPicSettings (struct camprop * *cam*, int *brightness*, int *contrast*, int *colour*, int *whiteness*)

setPicSettings - sets brightness, contrast, colour and whiteness (gamma).

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.3.4.27 int setVideoSource (struct camprop * *cam*, int *paramValue*, int *command*)

setVideoSource - sets some video source parameters.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.3.4.28 int setWhiteBalance (struct camprop * *cam*, char * *mode*, int *red*, int *blue*)

setWhiteBalance sets White Balance.

Arguments:

- mode - mode name
- red, blue - red and blue levels - valid only when mode is ”manual”

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

Function implemented for Linux.

4.3.4.29 int snap (struct camprop * *cam*, int *rgb*)

snap - reads frame and stories it in (libaudela) buffer.

4.3.4.30 int videoformat (struct camprop * *cam*, char * *formatname*)

videoformat - sets video format.

Possible format names:

- VGA - 640 x 480

- CIF - 352 x 288
- SIF - 320 x 240
- SSIF - 240 x 176
- QCIF - 176 x 144
- QSIF - 160 x 120
- SQCIF - 128 x 96.

Returns value:

- 0 when success.
- no 0 when error occurred, error description in cam->msg.

4.3.4.31 void yuv420p_to_rgb24 (unsigned char * *yuv*, unsigned char * *rgb*, int *width*, int *height*)

Convert from yuv to rgb.

Code comes from xawtv, actually it converts to bgr and flips vertically.

4.4 camtcl.c File Reference

Functions C-Tcl specifics for this camera.

```
#include "sysexp.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "camera.h"
#include "libcam.h"
#include "camtcl.h"
#include "util.h"
```

Functions

- int cmdCamClose (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamClose.
- int cmdCamVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamVideoSource.
- int cmdCamGetVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamGetVideoSource - returns specified camera settings.
- int cmdCamSetVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSetVideoSource - sets specified camera settings.
- int cmdCamVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamVideoFormat - Réglage des paramètres de la caméra.
- int cmdCamSetVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSetVideoFormat - implemented under Linux.
- int cmdCamGetVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamGetVideoFormat - implemented under Linux.
- int cmdCamSnap (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSnap - grabs black and white frame and store it in buffer.
- int cmdCamSnapRgb (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSnapRgb - grabs colour frame and store it in buffer.
- int cmdCamLonguePose (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamLonguePose - Réglage du mode longue pose.

- int cmdCamLonguePosePortAdress (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])

  cmdCamLonguePosePortAdress.

- int cmdCamLonguePoseStartValue (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])

  cmdCamLonguePoseStartValue - définition du caracter de debut de pose.

- int cmdCamLonguePoseStopValue (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])

  cmdCamLonguePoseStopValue - définition du caracter de fin de pose.

- int cmdCamValidFrame (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])

  cmdCamValidFrame - set valid frame number.

Variables

- ScanStruct * TheScanStruct = NULL

4.4.1 Detailed Description

Functions C-Tcl specifics for this camera.

Fonctions C-Tcl specifiques a cette camera. A programmer.

4.4.2 Function Documentation

4.4.2.1 int cmdCamClose (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamClose.

Ferme la caméra

close the camera.

4.4.2.2 int cmdCamGetVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamGetVideoFormat - implemented under Linux.

It returns actual video format.

4.4.2.3 int cmdCamGetVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamGetVideoSource - returns specified camera settings.

Implemented for Linux, use with many options.

4.4.2.4 int cmdCamLonguePose (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePose - Réglage du mode longue pose.

Declare if use long or normal exposure, with no parameters returns actual setting.

4.4.2.5 int cmdCamLonguePosePortAdress (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePosePortAdress.

Change or returns the long exposure port name (long exposure device).

4.4.2.6 int cmdCamLonguePoseStartValue (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePoseStartValue - définition du caracter de debut de pose.

4.4.2.7 int cmdCamLonguePoseStopValue (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePoseStopValue - définition du caracter de fin de pose.

4.4.2.8 int cmdCamSetVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSetVideoFormat - implemented under Linux.

Sets image format, argument must be format name (one of):

- SQCIF - 128x96
- QSIF - 160x120
- QCIF - 176x144
- SSIF - 240x176
- SIF - 320x240
- CIF - 352x288
- VGA - 640x480.

4.4.2.9 int cmdCamSetVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSetVideoSource - sets specified camera settings.

Implemented for Linux, use with many options.

4.4.2.10 int cmdCamSnap (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSnap - grabs black and white frame and store it in buffer.

Acqisition monocoup.

4.4.2.11 int cmdCamSnapRgb (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSnapRgb - grabs colour frame and store it in buffer.

Acqisition monocoup trois plans de couleurs.

4.4.2.12 int cmdCamValidFrame (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamValidFrame - set valid frame number.

Possible arguments:

- no argument - returns actual setting,
- number > 0 - set valid frame on number,
- number = 0 - set auto detection mode.

4.4.2.13 int cmdCamVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamVideoFormat - Réglage des paramètres de la caméra.

Under Linux it shows a window dialog where you can chose image format.

4.4.2.14 int cmdCamVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamVideoSource.

Réglage des paramètres de la caméra

Under Linux it calls **::confCam::confVideoSource** command and shows VideoSource window dialog.

4.4.3 Variable Documentation

4.4.3.1 ScanStruct* TheScanStruct = NULL

4.5 camtcl.h File Reference

Functions C-Tcl specifics for this camera.

Compounds

- struct ScanStruct

Functions

- int cmdCamClose (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamClose.
- int cmdCamVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamVideoSource.
- int cmdCamVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamVideoFormat - Réglage des paramètres de la caméra.
- int cmdCamSnap (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSnap - grabs black and white frame and store it in buffer.
- int cmdCamSnapRgb (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSnapRgb - grabs colour frame and store it in buffer.
- int cmdCamLonguePose (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamLonguePose - Réglage du mode longue pose.
- int cmdCamLonguePosePortAdress (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamLonguePosePortAdress.
- int cmdCamLonguePoseStartValue (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamLonguePoseStartValue - définition du caracter de debut de pose.
- int cmdCamLonguePoseStopValue (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamLonguePoseStopValue - définition du caracter de fin de pose.
- int cmdCamValidFrame (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamValidFrame - set valid frame number.
- int cmdCamSetVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSetVideoFormat - implemented under Linux.
- int cmdCamGetVideoFormat (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamGetVideoFormat - implemented under Linux.

- int cmdCamGetVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamGetVideoSource - returns specified camera settings.
- int cmdCamSetVideoSource (ClientData clientData, Tcl_Interp *interp, int argc, char *argv[ ])
  cmdCamSetVideoSource - sets specified camera settings.
- void ScanCallback (ClientData clientData)
- void ScanLibereStructure ()
- void ScanTerminateSequence (ClientData clientData, int camno, char *reason)
- void ScanTransfer (ClientData clientData)

4.5.1 Detailed Description

Functions C-Tcl specifics for this camera.

Fonctions C-Tcl specifiques a cette camera. A programmer.

4.5.2 Function Documentation

4.5.2.1 int cmdCamClose (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamClose.

Ferme la caméra

close the camera.

4.5.2.2 int cmdCamGetVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamGetVideoFormat - implemented under Linux.

It returns actual video format.

4.5.2.3 int cmdCamGetVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamGetVideoSource - returns specified camera settings.

Implemented for Linux, use with many options.

4.5.2.4 int cmdCamLonguePose (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePose - Réglage du mode longue pose.

Declare if use long or normal exposure, with no parameters returns actual setting.

4.5.2.5 int cmdCamLonguePosePortAdress (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePosePortAdress.

Change or returns the long exposure port name (long exposure device).

4.5.2.6 int cmdCamLonguePoseStartValue (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePoseStartValue - définition du caracter de debut de pose.

4.5.2.7 int cmdCamLonguePoseStopValue (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamLonguePoseStopValue - définition du caracter de fin de pose.

4.5.2.8 int cmdCamSetVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSetVideoFormat - implemented under Linux.

Sets image format, argument must be format name (one of):

- SQCIF - 128x96
- QSIF - 160x120
- QCIF - 176x144
- SSIF - 240x176
- SIF - 320x240
- CIF - 352x288
- VGA - 640x480.

4.5.2.9 int cmdCamSetVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSetVideoSource - sets specified camera settings.

Implemented for Linux, use with many options.

4.5.2.10 int cmdCamSnap (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSnap - grabs black and white frame and store it in buffer.

Acqisition monocoup.

4.5.2.11 int cmdCamSnapRgb (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamSnapRgb - grabs colour frame and store it in buffer.

Acqisition monocoup trois plans de couleurs.

4.5.2.12 int cmdCamValidFrame (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamValidFrame - set valid frame number.

Possible arguments:

- no argument - returns actual setting,
- number > 0 - set valid frame on number,
- number = 0 - set auto detection mode.

4.5.2.13 int cmdCamVideoFormat (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamVideoFormat - Réglage des paramètres de la caméra.

Under Linux it shows a window dialog where you can chose image format.

4.5.2.14 int cmdCamVideoSource (ClientData *clientData*, Tcl_Interp * *interp*, int *argc*, char * *argv*[ ])

cmdCamVideoSource.

Réglage des paramètres de la caméra

Under Linux it calls **::confCam::confVideoSource** command and shows VideoSource window dialog.

4.5.2.15 void ScanCallback (ClientData *clientData*)

4.5.2.16 void ScanLibereStructure ()

4.5.2.17 void ScanTerminateSequence (ClientData *clientData*, int *camno*, char * *reason*)

4.5.2.18 void ScanTransfer (ClientData *clientData*)

4.6 libname.h File Reference

Defines

- #define CAM_ENTRYPOINT Webcam_Init

  Nom du point d'entree de la librairie, doit etre Xx_Init pour une librairie libxx (la majuscule est importante pour permettre un chargement par load libxx).

- #define CAM_LIBNAME "libwebcam"

  Informations sur le driver, le nom est celui qui apparait quand on fait "package names" et la version apparait avec la commande Tcl "package require libxx".

- #define CAM_LIBVER "1.0"
- #define CAM_DRIVNAME "webcam"

  Initialisation d'informations indispensables pour la librairie xx.

4.6.1 Define Documentation

4.6.1.1 #define CAM_DRIVNAME "webcam"

Initialisation d'informations indispensables pour la librairie xx.

4.6.1.2 #define CAM_ENTRYPOINT Webcam_Init

Nom du point d'entree de la librairie, doit etre Xx_Init pour une librairie libxx (la majuscule est importante pour permettre un chargement par load libxx).

4.6.1.3 #define CAM_LIBNAME "libwebcam"

Informations sur le driver, le nom est celui qui apparait quand on fait "package names" et la version apparait avec la commande Tcl "package require libxx".

4.6.1.4 #define CAM_LIBVER "1.0"

4.7 pwc-ioctl.h File Reference

Compounds

- struct pwc_leds
- struct pwc_probe
- struct pwc_wb_speed
- struct pwc_whitebalance

Defines

- #define PWC_FPS_SHIFT 16
- #define PWC_FPS_MASK 0x00FF0000
- #define PWC_FPS_FRMASK 0x003F0000
- #define PWC_FPS_SNAPSHOT 0x00400000
- #define PWC_WB_INDOOR 0
- #define PWC_WB_OUTDOOR 1
- #define PWC_WB_FL 2
- #define PWC_WB_MANUAL 3
- #define PWC_WB_AUTO 4
- #define VIDIOCPWCRUSER _IO('v', 192)
- #define VIDIOCPWCSUSER _IO('v', 193)
- #define VIDIOCPWCFACTORY _IO('v', 194)
- #define VIDIOCPWCSCQUAL _IOW('v', 195, int)
- #define VIDIOCPWCGCQUAL _IOR('v', 195, int)
- #define VIDIOCPWCPROBE _IOR('v', 199, struct pwc_probe)
- #define VIDIOCPWCSAGC _IOW('v', 200, int)
- #define VIDIOCPWCGAGC _IOR('v', 200, int)
- #define VIDIOCPWCSSHUTTER _IOW('v', 201, int)
- #define VIDIOCPWCSAWB _IOW('v', 202, struct pwc_whitebalance)
- #define VIDIOCPWCGAWB _IOR('v', 202, struct pwc_whitebalance)
- #define VIDIOCPWCSAWBSPEED _IOW('v', 203, struct pwc_wb_speed)
- #define VIDIOCPWCGAWBSPEED _IOR('v', 203, struct pwc_wb_speed)
- #define VIDIOCPWCSLED _IOW('v', 205, struct pwc_leds)
- #define VIDIOCPWCGLED _IOR('v', 205, struct pwc_leds)
- #define VIDIOCPWCSCONTOUR _IOW('v', 206, int)
- #define VIDIOCPWCGCONTOUR _IOR('v', 206, int)
- #define VIDIOCPWCSBACKLIGHT _IOW('v', 207, int)
- #define VIDIOCPWCGBACKLIGHT _IOR('v', 207, int)
- #define VIDIOCPWCSFLICKER _IOW('v', 208, int)
- #define VIDIOCPWCGFLICKER _IOR('v', 208, int)
- #define VIDIOCPWCSDYNNOISE _IOW('v', 209, int)
- #define VIDIOCPWCGDYNNOISE _IOR('v', 209, int)

4.7.1 Define Documentation

4.7.1.1 #define PWC_FPS_FRMASK 0x003F0000

4.7.1.2 #define PWC_FPS_MASK 0x00FF0000

4.7.1.3 #define PWC_FPS_SHIFT 16

4.7.1.4 #define PWC_FPS_SNAPSHOT 0x00400000

4.7.1.5 #define PWC_WB_AUTO 4

4.7.1.6 #define PWC_WB_FL 2

4.7.1.7 #define PWC_WB_INDOOR 0

4.7.1.8 #define PWC_WB_MANUAL 3

4.7.1.9 #define PWC_WB_OUTDOOR 1

4.7.1.10 #define VIDIOCPWCFACTORY _IO('v', 194)

4.7.1.11 #define VIDIOCPWCGAGC _IOR('v', 200, int)

4.7.1.12 #define VIDIOCPWCGAWB _IOR('v', 202, struct pwc_whitebalance)

4.7.1.13 #define VIDIOCPWCGAWBSPEED _IOR('v', 203, struct pwc_wb_speed)

4.7.1.14 #define VIDIOCPWCGBACKLIGHT _IOR('v', 207, int)

4.7.1.15 #define VIDIOCPWCGCONTOUR _IOR('v', 206, int)

4.7.1.16 #define VIDIOCPWCGCQUAL _IOR('v', 195, int)

4.7.1.17 #define VIDIOCPWCGDYNNOISE _IOR('v', 209, int)

4.7.1.18 #define VIDIOCPWCGFLICKER _IOR('v', 208, int)

4.7.1.19 #define VIDIOCPWCGLED _IOR('v', 205, struct pwc_leds)

4.7.1.20 #define VIDIOCPWCPROBE _IOR('v', 199, struct pwc_probe)

4.7.1.21 #define VIDIOCPWCRUSER _IO('v', 192)

4.7.1.22 #define VIDIOCPWCSAGC _IOW('v', 200, int)

4.7.1.23 #define VIDIOCPWCSAWB _IOW('v', 202, struct pwc_whitebalance)

4.7.1.24 #define VIDIOCPWCSAWBSPEED _IOW('v', 203, struct pwc_wb_speed)

4.7.1.25 #define VIDIOCPWCSBACKLIGHT _IOW('v', 207, int)

4.7.1.26 #define VIDIOCPWCSCONTOUR _IOW('v', 206, int)

4.7.1.27 #define VIDIOCPWCSCQUAL _IOW('v', 195, int)

4.7.1.28 #define VIDIOCPWCSDYNNOISE _IOW('v', 209, int)

4.7.1.29 #define VIDIOCPWCSFLICKER _IOW('v', 208, int)

4.7.1.30 #define VIDIOCPWCSLED _IOW('v', 205, struct pwc_leds)

Index