libdc1394

2.1.2

Generated by Doxygen 1.6.3

Tue May 25 14:11:26 2010

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 __dc1394_avt_adv_feature_info_struct Struct Reference

`#include <dc1394/vendor/avt.h>`

3.1.1 Detailed Description

Inquiry of available advanced features - This struct is provided for backwards compatibility. New code should use dc1394_avt_adv_function_inquiry_t instead.

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

- dc1394/vendor/**avt.h**

3.2 __dc1394_avt_csradv_f7mode_mapping_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- uint32_t **m_Inq** [2]
  available extended modes
- uint32_t **m_nMappedModes** [DC1394_AVT_NUM_FORMAT7_MODES]
  mapped format 7 modes

3.2.1 Detailed Description

Extended/mapped format 7 modes

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

- dc1394/vendor/avt_csr_structs_adv.h

3.3 __dc1394_avt_csradv_image_mirror_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- uint32_t **m_MNR**

  preserve compatibility with Marlin series

3.3.1 Detailed Description

Image mirror register definition.

Remarks

- The former name of this register was MISC_FEATURES
- The m_MNR member has been kept for compatibility with older Marlin software versions. Nevertheless the MNR features has been removed from recent Marlin software versions.

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

- dc1394/vendor/avt_csr_structs_adv.h

3.4 __dc1394_avt_csradv_io_inp_debounce_x_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- uint32_t **m_nMinDebTime**
  minimum debounce time
- uint32_t **m_nMaxDebTime**
  maximum debounce time

3.4.1 Detailed Description

Advanced feature input debounce control

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

- dc1394/vendor/avt_csr_structs_adv.h

3.5 __dc1394_avt_csradv_max_isosize_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- union __dc1394_avt_csradv_max_isosize_struct::__tagCsrAdvMaxIsoSizeElems **m_S400**
  ISO packet size S400.
- union __dc1394_avt_csradv_max_isosize_struct::__tagCsrAdvMaxIsoSizeElems **m_S800**
  ISO packet size S800.

3.5.1 Detailed Description

Advanced feature to override the maximum ISO packet size. Because we want speed, we don't obey S100 and S200 here...

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

- dc1394/vendor/avt_csr_structs_adv.h

3.6 __dc1394_avt_csradv_paramlist_info_union Union Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

3.6.1 Detailed Description

Advanced parameter list info

Remarks

GPDATA_BUFFER and PARAMLIST_BUFFER share the same memory!

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

- dc1394/vendor/avt_csr_structs_adv.h

3.7 __dc1394_avt_csradv_paramupd_timing_union Union Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

3.7.1 Detailed Description

Advanced parameter update control This feature controls the update behaviour of the camera between the host and the uC and also between the uC and the FPGA. It controls when and how new parameters become active

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

- dc1394/vendor/avt_csr_structs_adv.h

3.8 __dc1394_avt_csradv_sis_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- uint32_t **m_nUserVal**
  user provided SIS value

3.8.1 Detailed Description

SIS control structure

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

- dc1394/vendor/avt_csr_structs_adv.h

3.9 __dc1394_avt_csradv_whitebal_struct Struct Reference

`#include <dc1394/vendor/avt_csr_structs_adv.h>`

Data Fields

- union __dc1394_avt_csradv_whitebal_struct::__tagAdvWhiteBal1 **m_WhiteBal12**
  white balance control PXGA 1+2
- union __dc1394_avt_csradv_whitebal_struct::__tagAdvWhiteBal1 **m_WhiteBal34**
  white balance control PXGA 3+4

3.9.1 Detailed Description

Advanced white balance - extends the standard white balance feature to 4 independent color controllers

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

- dc1394/vendor/avt_csr_structs_adv.h

3.10 __dc1394_avt_sis_data_struct Struct Reference

`#include <dc1394/vendor/avt.h>`

Data Fields

- uint32_t **FrameCounter**
  frame number
- uint32_t **TriggerCounter**
  number of detected trigger events
- uint16_t **AOILeft**
  AOI left position.
- uint16_t **AOITop**
  AOI top position.
- uint16_t **AOIWidth**
  AOI width.
- uint16_t **AOIHeight**
  AOI height.
- uint32_t **Shutter**
  shutter setting
- uint16_t **Gain**
  gain setting
- uint16_t **Reserved0**
  reserved for future use
- uint8_t **OutputState** [4]
  output pin states - idle:0 active:255
- uint8_t **InputState** [2]
  input pin states - idle:0 active:255
- uint8_t **Reserved1** [2]
  reserved for future use
- uint8_t **SequenceIndex**
  sequence position (sequence mode)
- uint8_t **Reserved2a**
  reserved for future use
- uint8_t **ColorCoding**

IIDC color coding.

- uint8_t **Reserved2b**
  reserved for future use
- uint32_t **SerialNumber**
  camera serial number
- uint32_t **UserValue**
  user defined value
- uint32_t **Offset**: 12
  elapsed time within the current bus cycle, unit 40.69 ns (range 0-3071)
- uint32_t **Cycles**: 13
  elapsed time in firewire bus cycles, unit 125 us, will be reset every second (range 0-7999)
- uint32_t **Seconds**: 7
  elapsed time in seconds

3.10.1 Detailed Description

SIS data structure See also dc1394_avt_set_sis, dc1394_avt_get_sis_data_inquiry and dc1394_avt_get_-sis_data.

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

- dc1394/vendor/**avt.h**

3.11 __dc1394_avt_smart_feature_info_struct Struct Reference

`#include <dc1394/vendor/avt.h>`

3.11.1 Detailed Description

Inquiry of available advanced features - This struct is intended to be used as a replacement for dc1394_-avt_adv_feature_info_t. For initialization see **dc1394_avt_get_smart_feature_inquiry()** (p. 112).

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

- dc1394/vendor/**avt.h**

3.12 __dc1394_camera Struct Reference

`#include <dc1394/camera.h>`

3.12.1 Detailed Description

Camera structure

This structure represents the camera in libdc1394. It contains a number of useful static information, such as model/vendor names, a few capabilities, some ROM offsets, a unique identifier, etc...

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

- dc1394/**camera.h**

3.13 __dc1394_pxl_adv_feature_info_struct Struct Reference

`#include <dc1394/vendor/pixelink.h>`

3.13.1 Detailed Description

Advanced feature inquiry

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

- dc1394/vendor/**pixelink.h**

3.14 __dc1394_pxl_camera_info_struct Struct Reference

`#include <dc1394/vendor/pixelink.h>`

3.14.1 Detailed Description

Camera information

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

- dc1394/vendor/**pixelink.h**

3.15 __dc1394_pxl_gpio_info_struct Struct Reference

`#include <dc1394/vendor/pixelink.h>`

3.15.1 Detailed Description

GPIO Information structure

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

- dc1394/vendor/**pixelink.h**

3.16 __dc1394_video_frame Struct Reference

`#include <dc1394/video.h>`

3.16.1 Detailed Description

Video frame structure.

dc1394video_frame_t is the structure returned by the capture functions. It contains the captured image as well as a number of information.

In general this structure should be calloc'ed so that members such as "allocated size" are properly set to zero. Don't forget to free the "image" member before freeing the struct itself.

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

- dc1394/**video.h**

3.17 __dc1394camera_list_t Struct Reference

`#include <dc1394/camera.h>`

3.17.1 Detailed Description

A list of cameras

Usually returned by dc1394_camera_eumerate().

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

- dc1394/**camera.h**

3.18 __dc1394feature_info_t_struct Struct Reference

`#include <dc1394/control.h>`

3.18.1 Detailed Description

A structure containing all information about a feature.

Some fields are only valid for some features (e.g. trigger, white balance,...)

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

- dc1394/**control.h**

3.19 __dc1394featureset_t Struct Reference

`#include <dc1394/control.h>`

3.19.1 Detailed Description

The list of features

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

- dc1394/**control.h**

3.20 __dc1394format7mode_t Struct Reference

`#include <dc1394/format7.h>`

3.20.1 Detailed Description

A struct containing information about a mode of Format_7, the scalable image format.

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

- dc1394/**format7.h**

3.21 __dc1394format7modeset_t Struct Reference

`#include <dc1394/format7.h>`

3.21.1 Detailed Description

A struct containing the list of Format_7 modes. FIXME: this may become very big if format_7 pages are used in IIDC 1.32. It would be better to use a "num" and an allocated list.

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

- dc1394/**format7.h**

3.22 dc1394basler_dcam_csr_value_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.22.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.23 dc1394basler_dcam_whitebalance_csr_value_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.23.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.24 dc1394basler_sff_chunk_tail_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.24.1 Detailed Description

SFF chunks are read from the end to the beginning (that is backwards), each chunk ends in a chunk-tail, which contains information about the size of the chunk as well as the GUID used to identify which chunk this really is.

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

- dc1394/vendor/**basler_sff.h**

3.25 dc1394basler_sff_crc_checksum_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.25.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.26 dc1394basler_sff_cycle_time_stamp_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.26.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.27 dc1394basler_sff_dcam_values_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.27.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.28 dc1394basler_sff_extended_data_stream_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.28.1 Detailed Description

This structure is used to capture the SFF extended data stream chunk. According to the Basler manuals the extended data stream chunk also contains to members pixel_data and gap of variable size; these members are ignored in this API because they can be obtained from other sources. The pixel_data member which is the actual image frame is all data from the beginning of the frame buffer until width∗height∗bytes_per_-pixel bytes. The gap is required on some cameras for technical reason but not used otherwise. The size of the gap can be computed by computing frame_size - sizeof all chunks - image_size.

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

- dc1394/vendor/**basler_sff.h**

3.29 dc1394basler_sff_frame_counter_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.29.1 Detailed Description

No Docs

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

- dc1394/vendor/**basler_sff.h**

3.30 dc1394basler_sff_guid_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.30.1 Detailed Description

Basler SFF Guid struct, this structure is used to identify chunks and to request features from the camera, is basically just a normal GUID value

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

- dc1394/vendor/**basler_sff.h**

3.31 dc1394basler_sff_t Struct Reference

`#include <dc1394/vendor/basler_sff.h>`

3.31.1 Detailed Description

Data type used by this API to define SFFs and also when iterating

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

- dc1394/vendor/**basler_sff.h**

3.32 dc1394camera_id_t Struct Reference

`#include <dc1394/camera.h>`

3.32.1 Detailed Description

A unique identifier for a functional camera unit

Since a single camera can contain several functional units (think stereo cameras), the GUID is not enough to identify an IIDC camera. The unit number must also be used, hence this struct.

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

- dc1394/**camera.h**

3.33 dc1394color_codings_t Struct Reference

`#include <dc1394/types.h>`

3.33.1 Detailed Description

A struct containing a list of color codings

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

- dc1394/**types.h**

3.34 dc1394feature_modes_t Struct Reference

`#include <dc1394/control.h>`

3.34.1 Detailed Description

List of feature modes

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

- dc1394/**control.h**

3.35 dc1394framerates_t Struct Reference

`#include <dc1394/video.h>`

3.35.1 Detailed Description

List of framerates

dc1394framerates_t (p. 40) contains a list of available framerates for a particular video mode.

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

- dc1394/**video.h**

3.36 dc1394trigger_modes_t Struct Reference

`#include <dc1394/control.h>`

3.36.1 Detailed Description

List of trigger modes

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

- dc1394/**control.h**

3.37 dc1394trigger_sources_t Struct Reference

`#include <dc1394/control.h>`

3.37.1 Detailed Description

List of trigger sources

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

- dc1394/**control.h**

3.38 dc1394video_modes_t Struct Reference

`#include <dc1394/types.h>`

3.38.1 Detailed Description

A struct containing a list of video modes

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

- dc1394/**types.h**

3.39 fw_cdev_add_descriptor Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.39.1 Detailed Description

struct **fw_cdev_add_descriptor** (p. 44) - Add contents to the local node's config ROM : If non-zero, immediate key to insert before pointer : Upper 8 bits of root directory pointer : Userspace pointer to contents of descriptor block : Length of descriptor block data, in bytes : Handle to the descriptor, written by the kernel

Add a descriptor block and optionally a preceding immediate key to the local node's configuration ROM.

The field specifies the upper 8 bits of the descriptor root directory pointer and the and fields specify the contents. The should be of the form 0xXX000000. The offset part of the root directory entry will be filled in by the kernel.

If not 0, the field specifies an immediate key which will be inserted before the root directory pointer.

If successful, the kernel adds the descriptor and writes back a handle to the kernel-side object to be used for later removal of the descriptor block and immediate key.

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

- dc1394/juju/firewire-cdev.h

3.40 fw_cdev_allocate Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.40.1 Detailed Description

struct **fw_cdev_allocate** (p. 45) - Allocate a CSR address range : Start offset of the address range : To be passed back to userspace in request events : Length of the address range, in bytes : Handle to the allocation, written by the kernel

Allocate an address range in the 48-bit address space on the local node (the controller). This allows userspace to listen for requests with an offset within that address range. When the kernel receives a request within the range, an &**fw_cdev_event_request** (p. 54) event will be written back. The field is passed back to userspace in the response event. The field is an out parameter, returning a handle to the allocated range to be used for later deallocation of the range.

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

- dc1394/juju/firewire-cdev.h

3.41 fw_cdev_allocate_iso_resource Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.41.1 Detailed Description

struct **fw_cdev_allocate_iso_resource** (p. 46) - (De)allocate a channel or bandwidth : Passed back to userspace in correponding iso resource events : Isochronous channels of which one is to be (de)allocated : Isochronous bandwidth units to be (de)allocated : Handle to the allocation, written by the kernel (only valid in case of FW_CDEV_IOC_ALLOCATE_ISO_RESOURCE ioctls)

The FW_CDEV_IOC_ALLOCATE_ISO_RESOURCE ioctl initiates allocation of an isochronous channel and/or of isochronous bandwidth at the isochronous resource manager (IRM). Only one of the channels specified in is allocated. An FW_CDEV_EVENT_ISO_RESOURCE_ALLOCATED is sent after communication with the IRM, indicating success or failure in the event data. The kernel will automatically reallocate the resources after bus resets. Should a reallocation fail, an FW_CDEV_EVENT_ISO_RESOURCE_-DEALLOCATED event will be sent. The kernel will also automatically deallocate the resources when the file descriptor is closed.

The FW_CDEV_IOC_DEALLOCATE_ISO_RESOURCE ioctl can be used to initiate deallocation of resources which were allocated as described above. An FW_CDEV_EVENT_ISO_RESOURCE_-DEALLOCATED event concludes this operation.

The FW_CDEV_IOC_ALLOCATE_ISO_RESOURCE_ONCE ioctl is a variant of allocation without automatic re- or deallocation. An FW_CDEV_EVENT_ISO_RESOURCE_ALLOCATED event concludes this operation, indicating success or failure in its data.

The FW_CDEV_IOC_DEALLOCATE_ISO_RESOURCE_ONCE ioctl works like FW_CDEV_IOC_-ALLOCATE_ISO_RESOURCE_ONCE except that resources are freed instead of allocated. At most one channel may be specified in this ioctl. An FW_CDEV_EVENT_ISO_RESOURCE_DEALLOCATED event concludes this operation.

To summarize, FW_CDEV_IOC_DEALLOCATE_ISO_RESOURCE allocates iso resources for the lifetime of the fd or handle. In contrast, FW_CDEV_IOC_ALLOCATE_ISO_RESOURCE_ONCE allocates iso resources for the duration of a bus generation.

is a host-endian bitfield with the most significant bit representing channel 0 and the least significant bit representing channel 63: 1ULL << (63 - c)

is expressed in bandwidth allocation units, i.e. the time to send one quadlet of data (payload or header data) at speed S1600.

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

- dc1394/juju/firewire-cdev.h

3.42 fw_cdev_create_iso_context Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.42.1 Detailed Description

struct **fw_cdev_create_iso_context** (p. 47) - Create a context for isochronous IO : FW_CDEV_ISO_-CONTEXT_TRANSMIT or FW_CDEV_ISO_CONTEXT_RECEIVE : Header size to strip for receive contexts : Channel to bind to : Speed to transmit at : To be returned in &**fw_cdev_event_iso_interrupt** (p. 52) : Handle to context, written back by kernel

Prior to sending or receiving isochronous I/O, a context must be created. The context records information about the transmit or receive configuration and typically maps to an underlying hardware resource. A context is set up for either sending or receiving. It is bound to a specific isochronous channel.

If a context was successfully created, the kernel writes back a handle to the context, which must be passed in for subsequent operations on that context.

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

- dc1394/juju/firewire-cdev.h

3.43 fw_cdev_deallocate Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.43.1 Detailed Description

struct **fw_cdev_deallocate** (p. 48) - Free a CSR address range or isochronous resource : Handle to the address range or iso resource, as returned by the kernel when the range or resource was allocated

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

- dc1394/juju/firewire-cdev.h

3.44 fw_cdev_event Union Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.44.1 Detailed Description

union **fw_cdev_event** (p. 49) - Convenience union of fw_cdev_event_ types : Valid for all types : Valid if .type == FW_CDEV_EVENT_BUS_RESET : Valid if .type == FW_CDEV_EVENT_RESPONSE : Valid if .type == FW_CDEV_EVENT_REQUEST : Valid if .type == FW_CDEV_EVENT_ISO_INTERRUPT : Valid if .type == FW_CDEV_EVENT_ISO_RESOURCE_ALLOCATED or FW_CDEV_EVENT_ISO_-RESOURCE_DEALLOCATED

Convenience union for userspace use. Events could be read(2) into an appropriately aligned char buffer and then cast to this union for further processing. Note that for a request, response or iso_interrupt event, the data[] or header[] may make the size of the full event larger than sizeof(union fw_cdev_event). Also note that if you attempt to read(2) an event into a buffer that is not large enough for it, the data that does not fit will be discarded so that the next read(2) will return a new event.

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

- dc1394/juju/firewire-cdev.h

3.45 fw_cdev_event_bus_reset Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.45.1 Detailed Description

struct **fw_cdev_event_bus_reset** (p. 50) - Sent when a bus reset occurred : See set by FW_CDEV_IOC_-GET_INFO ioctl : See always FW_CDEV_EVENT_BUS_RESET : New node ID of this node : Node ID of the local node, i.e. of the controller : Node ID of the bus manager : Node ID of the iso resource manager : Node ID of the root node : New bus generation

This event is sent when the bus the device belongs to goes through a bus reset. It provides information about the new bus configuration, such as new node ID for this device, new root ID, and others.

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

- dc1394/juju/firewire-cdev.h

3.46 fw_cdev_event_common Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.46.1 Detailed Description

struct **fw_cdev_event_common** (p. 51) - Common part of all fw_cdev_event_ types : For arbitrary use by userspace : Discriminates the fw_cdev_event_ types

This struct may be used to access generic members of all fw_cdev_event_ types regardless of the specific type.

Data passed in the field for a request will be returned in the corresponding event. It is big enough to hold a pointer on all platforms. The ioctl used to set depends on the of event.

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

- dc1394/juju/firewire-cdev.h

3.47 fw_cdev_event_iso_interrupt Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.47.1 Detailed Description

struct **fw_cdev_event_iso_interrupt** (p. 52) - Sent when an iso packet was completed : See set by FW_-CDEV_CREATE_ISO_CONTEXT ioctl : See always FW_CDEV_EVENT_ISO_INTERRUPT : Cycle counter of the interrupt packet : Total length of following headers, in bytes : Stripped headers, if any

This event is sent when the controller has completed an &**fw_cdev_iso_packet** (p. 60) with the FW_-CDEV_ISO_INTERRUPT bit set. In the receive case, the headers stripped of all packets up until and including the interrupt packet are returned in the field.

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

- dc1394/juju/firewire-cdev.h

3.48 fw_cdev_event_iso_resource Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.48.1 Detailed Description

struct **fw_cdev_event_iso_resource** (p. 53) - Iso resources were allocated or freed : See set by FW_CDEV_IOC_(DE)ALLOCATE_ISO_RESOURCE(_ONCE) ioctl : FW_CDEV_EVENT_ISO_-RESOURCE_ALLOCATED or FW_CDEV_EVENT_ISO_RESOURCE_DEALLOCATED : Reference by which an allocated resource can be deallocated : Isochronous channel which was (de)allocated, if any : Bandwidth allocation units which were (de)allocated, if any : Last known availability of channels : Last known availability of bandwidth

An FW_CDEV_EVENT_ISO_RESOURCE_ALLOCATED event is sent after an isochronous resource was allocated at the IRM. The client has to check and for whether the allocation actually succeeded.

An FW_CDEV_EVENT_ISO_RESOURCE_DEALLOCATED event is sent after an isochronous resource was deallocated at the IRM. It is also sent when automatic reallocation after a bus reset failed.

is <0 if no channel was (de)allocated or if reallocation failed. is 0 if no bandwidth was (de)allocated or if reallocation failed.

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

- dc1394/juju/firewire-cdev.h

3.49 fw_cdev_event_request Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.49.1 Detailed Description

struct **fw_cdev_event_request** (p. 54) - Sent on incoming request to an address region : See set by FW_CDEV_IOC_ALLOCATE ioctl : See always FW_CDEV_EVENT_REQUEST : Transaction code of the incoming request : The offset into the 48-bit per-node address space : Reference to the kernel-side pending request : Data length, i.e. the request's payload size in bytes : Incoming data, if any

This event is sent when the stack receives an incoming request to an address region registered using the FW_CDEV_IOC_ALLOCATE ioctl. The request is guaranteed to be completely contained in the specified region. Userspace is responsible for sending the response by FW_CDEV_IOC_SEND_RESPONSE ioctl, using the same .

The payload data for requests carrying data (write and lock requests) follows immediately and can be accessed through the field.

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

- dc1394/juju/firewire-cdev.h

3.50 fw_cdev_event_response Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.50.1 Detailed Description

struct **fw_cdev_event_response** (p. 55) - Sent when a response packet was received : See set by FW_-CDEV_IOC_SEND_REQUEST ioctl : See always FW_CDEV_EVENT_RESPONSE : Response code returned by the remote node : Data length, i.e. the response's payload size in bytes : Payload data, if any

This event is sent when the stack receives a response to an outgoing request sent by FW_CDEV_IOC_-SEND_REQUEST ioctl. The payload data for responses carrying data (read and lock responses) follows immediately and can be accessed through the field.

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

- dc1394/juju/firewire-cdev.h

3.51 fw_cdev_get_cycle_timer Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.51.1 Detailed Description

struct **fw_cdev_get_cycle_timer** (p. 56) - read cycle timer register : system time, in microseconds since the Epoch : isochronous cycle timer, as per OHCI 1.1 clause 5.13

The FW_CDEV_IOC_GET_CYCLE_TIMER ioctl reads the isochronous cycle timer and also the system clock. This allows to express the receive time of an isochronous packet as a system time with microsecond accuracy.

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

- dc1394/juju/firewire-cdev.h

3.52 fw_cdev_get_info Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.52.1 Detailed Description

struct **fw_cdev_get_info** (p. 57) - General purpose information ioctl

Version

: The version field is just a running serial number. We never break backwards compatibility, but may add more structs and ioctls in later revisions. : If is non-zero, at most rom_length bytes of configuration ROM will be copied into that user space address. In either case, is updated with the actual length of the configuration ROM. : If non-zero, address of a buffer to be filled by a copy of the device's configuration ROM : If non-zero, address of a buffer to be filled by a &struct **fw_cdev_event_bus_reset** (p. 50) with the current state of the bus. This does not cause a bus reset to happen. : Value of &closure in this and subsequent bus reset events : The index of the card this device belongs to

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

- dc1394/juju/firewire-cdev.h

3.53 fw_cdev_get_speed Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.53.1 Detailed Description

struct **fw_cdev_get_speed** (p. 58) - Query maximum speed to or from this device : Speed code; minimum of the device's link speed, the local node's link speed, and all PHY port speeds between the two links

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

- dc1394/juju/firewire-cdev.h

3.54 fw_cdev_initiate_bus_reset Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.54.1 Detailed Description

struct **fw_cdev_initiate_bus_reset** (p. 59) - Initiate a bus reset : FW_CDEV_SHORT_RESET or FW_-CDEV_LONG_RESET

Initiate a bus reset for the bus this device is on. The bus reset can be either the original (long) bus reset or the arbitrated (short) bus reset introduced in 1394a-2000.

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

- dc1394/juju/firewire-cdev.h

3.55 fw_cdev_iso_packet Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.55.1 Detailed Description

struct **fw_cdev_iso_packet** (p. 60) - Isochronous packet : Contains the header length (8 uppermost bits), the sy field (4 bits), the tag field (2 bits), a sync flag (1 bit), a skip flag (1 bit), an interrupt flag (1 bit), and the payload length (16 lowermost bits) : Header and payload

&struct **fw_cdev_iso_packet** (p. 60) is used to describe isochronous packet queues.

Use the FW_CDEV_ISO_ macros to fill in . The sy and tag fields are specified by IEEE 1394a and IEC 61883.

FIXME - finish this documentation

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

- dc1394/juju/firewire-cdev.h

3.56 fw_cdev_queue_iso Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.56.1 Detailed Description

struct **fw_cdev_queue_iso** (p. 61) - Queue isochronous packets for I/O : Userspace pointer to packet data : Pointer into mmap()'ed payload buffer : Size of packet data in bytes : Isochronous context handle

Queue a number of isochronous packets for reception or transmission. This ioctl takes a pointer to an array of &**fw_cdev_iso_packet** (p. 60) structs, which describe how to transmit from or receive into a contiguous region of a mmap()'ed payload buffer. As part of the packet descriptors, a series of headers can be supplied, which will be prepended to the payload during DMA.

The kernel may or may not queue all packets, but will write back updated values of the , and fields, so the ioctl can be resubmitted easily.

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

- dc1394/juju/firewire-cdev.h

3.57 fw_cdev_remove_descriptor Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.57.1 Detailed Description

struct **fw_cdev_remove_descriptor** (p. 62) - Remove contents from the configuration ROM : Handle to the descriptor, as returned by the kernel when the descriptor was added

Remove a descriptor block and accompanying immediate key from the local node's configuration ROM.

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

- dc1394/juju/firewire-cdev.h

3.58 fw_cdev_send_request Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.58.1 Detailed Description

struct **fw_cdev_send_request** (p. 63) - Send an asynchronous request packet : Transaction code of the request : Length of outgoing payload, in bytes : 48-bit offset at destination node : Passed back to userspace in the response event : Userspace pointer to payload : The bus generation where packet is valid

Send a request to the device. This ioctl implements all outgoing requests. Both quadlet and block request specify the payload as a pointer to the data in the field. Once the transaction completes, the kernel writes an &**fw_cdev_event_response** (p. 55) event back. The field is passed back to user space in the response event.

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

- dc1394/juju/firewire-cdev.h

3.59 fw_cdev_send_response Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.59.1 Detailed Description

struct **fw_cdev_send_response** (p. 64) - Send an asynchronous response packet : Response code as determined by the userspace handler : Length of outgoing payload, in bytes : Userspace pointer to payload : The handle from the &**fw_cdev_event_request** (p. 54)

Send a response to an incoming request. By setting up an address range using the FW_CDEV_IOC_-ALLOCATE ioctl, userspace can listen for incoming requests. An incoming request will generate an FW_CDEV_EVENT_REQUEST, and userspace must send a reply using this ioctl. The event has a handle to the kernel-side pending transaction, which should be used with this ioctl.

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

- dc1394/juju/firewire-cdev.h

3.60 fw_cdev_start_iso Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.60.1 Detailed Description

struct **fw_cdev_start_iso** (p. 65) - Start an isochronous transmission or reception : Cycle in which to start I/O. If is greater than or equal to 0, the I/O will start on that cycle. : Determines the value to wait for for receive packets that have the FW_CDEV_ISO_SYNC bit set : Tag filter bit mask. Only valid for isochronous reception. Determines the tag values for which packets will be accepted. Use FW_CDEV_-ISO_CONTEXT_MATCH_ macros to set . : Isochronous context handle within which to transmit or receive

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

- dc1394/juju/firewire-cdev.h

3.61 fw_cdev_stop_iso Struct Reference

`#include <dc1394/juju/firewire-cdev.h>`

3.61.1 Detailed Description

struct **fw_cdev_stop_iso** (p. 66) - Stop an isochronous transmission or reception : Handle of isochronous context to stop

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

- dc1394/juju/firewire-cdev.h

Chapter 4

File Documentation

4.1 dc1394/camera.h File Reference

Basic system and camera functions.

```
#include <dc1394/log.h>
#include <stdint.h>
#include <dc1394/log.h>
#include <dc1394/video.h>
#include <stdio.h>
```

Data Structures

- struct **__dc1394_camera**
- struct **dc1394camera_id_t**
- struct **__dc1394camera_list_t**

Typedefs

- typedef struct **__dc1394_camera dc1394camera_t**
- typedef struct **__dc1394camera_list_t dc1394camera_list_t**

Enumerations

- enum **dc1394iidc_version_t**
- enum **dc1394power_class_t**
- enum **dc1394phy_delay_t**

Functions

- dc1394_t ∗ **dc1394_new** (void)
- void **dc1394_free** (dc1394_t ∗dc1394)
- dc1394error_t **dc1394_camera_set_broadcast** (**dc1394camera_t** ∗camera, **dc1394bool_t** pwr)

- dc1394error_t **dc1394_reset_bus** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_camera_get_node** (**dc1394camera_t** ∗camera, uint32_t ∗node, uint32_t ∗generation)
- dc1394error_t **dc1394_camera_enumerate** (dc1394_t ∗dc1394, **dc1394camera_list_t** ∗∗list)
- void **dc1394_camera_free_list** (**dc1394camera_list_t** ∗list)
- **dc1394camera_t** ∗ **dc1394_camera_new** (dc1394_t ∗dc1394, uint64_t guid)
- **dc1394camera_t** ∗ **dc1394_camera_new_unit** (dc1394_t ∗dc1394, uint64_t guid, int unit)
- void **dc1394_camera_free** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_camera_print_info** (**dc1394camera_t** ∗camera, FILE ∗fd)

4.1.1 Detailed Description

Basic system and camera functions.

Author

Damien Douxchamps: coding
Peter Antoniac: documentation maintainer

More details soon

4.1.2 Typedef Documentation

4.1.2.1 typedef struct __dc1394camera_list_t dc1394camera_list_t

A list of cameras

Usually returned by dc1394_camera_eumerate().

4.1.2.2 typedef struct __dc1394_camera dc1394camera_t

Camera structure

This structure represents the camera in libdc1394. It contains a number of useful static information, such as model/vendor names, a few capabilities, some ROM offsets, a unique identifier, etc...

4.1.3 Enumeration Type Documentation

4.1.3.1 enum dc1394iidc_version_t

List of IIDC versions

Currently, the following versions exist: 1.04, 1.20, PTGREY, 1.30 and 1.31 (1.32 coming soon) Observing other versions means that there's a bug crawling somewhere.

4.1.3.2 enum dc1394phy_delay_t

Enumeration of PHY delays

This is currently not used in libdc1394.

4.1.3.3 enum dc1394power_class_t

Enumeration of power classes

This is currently not used in libdc1394.

4.1.4 Function Documentation

4.1.4.1 dc1394error_t dc1394_camera_enumerate (dc1394_t * *dc1394*, dc1394camera_list_t ** *list*)

Returns the list of cameras available on the computer. If present, multiple cards will be probed

4.1.4.2 void dc1394_camera_free (dc1394camera_t * *camera*)

Frees a camera structure

4.1.4.3 void dc1394_camera_free_list (dc1394camera_list_t * *list*)

Frees the memory allocated in dc1394_enumerate_cameras for the camera list

4.1.4.4 dc1394error_t dc1394_camera_get_node (dc1394camera_t * *camera*, uint32_t * *node*, uint32_t * *generation*)

Gets the IEEE 1394 node ID of the camera.

4.1.4.5 dc1394camera_t* dc1394_camera_new (dc1394_t * *dc1394*, uint64_t *guid*)

Create a new camera based on a GUID (Global Unique IDentifier)

4.1.4.6 dc1394camera_t* dc1394_camera_new_unit (dc1394_t * *dc1394*, uint64_t *guid*, int *unit*)

Create a new camera based on a GUID and a unit number (for multi-unit cameras)

4.1.4.7 dc1394error_t dc1394_camera_print_info (dc1394camera_t * *camera*, FILE * *fd*)

Print various camera information, such as GUID, vendor, model, supported IIDC specs, etc...

4.1.4.8 dc1394error_t dc1394_camera_set_broadcast (dc1394camera_t * *camera*, dc1394bool_t *pwr*)

Sets and gets the broadcast flag of a camera. If the broadcast flag is set, all devices on the bus will execute the command. Useful to sync ISO start commands or setting a bunch of cameras at the same time. Broadcast only works with identical devices (brand/model). If the devices are not identical your mileage may vary. Some cameras may not answer broadcast commands at all. Also, this only works with cameras on the SAME bus (IOW, the same port).

4.1.4.9 void dc1394_free (dc1394_t * *dc1394*)

Liberates a context. Last function to use in your program. After this, no libdc1394 function can be used.

4.1.4.10 dc1394_t* dc1394_new (void)

Creates a new context in which cameras can be searched and used. This should be called before using any other libdc1394 function.

4.1.4.11 dc1394error_t dc1394_reset_bus (dc1394camera_t * *camera*)

Resets the IEEE1394 bus which camera is attached to. Calling this function is "rude" to other devices because it causes them to re-enumerate on the bus and may cause a temporary disruption in their current activities. Thus, use it sparingly. Its primary use is if a program shuts down uncleanly and needs to free leftover ISO channels or bandwidth. A bus reset will free those things as a side effect.

4.2 dc1394/capture.h File Reference

Capture functions.

```
#include <dc1394/log.h>
#include <dc1394/video.h>
```

Defines

- #define **DC1394_CAPTURE_FLAGS_CHANNEL_ALLOC** 0x00000001U

Enumerations

- enum **dc1394capture_policy_t**

Functions

- dc1394error_t **dc1394_capture_setup** (**dc1394camera_t** ∗camera, uint32_t num_dma_buffers, uint32_t flags)
- dc1394error_t **dc1394_capture_stop** (**dc1394camera_t** ∗camera)
- int **dc1394_capture_get_fileno** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_capture_dequeue** (**dc1394camera_t** ∗camera, **dc1394capture_policy_t** policy, **dc1394video_frame_t** ∗∗frame)
- dc1394error_t **dc1394_capture_enqueue** (**dc1394camera_t** ∗camera, **dc1394video_frame_t** ∗frame)
- **dc1394bool_t dc1394_capture_is_frame_corrupt** (**dc1394camera_t** ∗camera, **dc1394video_frame_t** ∗frame)

4.2.1 Detailed Description

Capture functions.

Author

Damien Douxchamps: coding
Peter Antoniac: documentation maintainer

More details soon

4.2.2 Define Documentation

4.2.2.1 #define DC1394_CAPTURE_FLAGS_CHANNEL_ALLOC 0x00000001U

Capture flags. Currently limited to switching automatic functions on/off: channel allocation, bandwidth allocation and automatic starting of ISO transmission

4.2.3 Enumeration Type Documentation

4.2.3.1 enum dc1394capture_policy_t

The capture policy.

Can be blocking (wait for a frame forever) or polling (returns if no frames is in the ring buffer)

4.2.4 Function Documentation

4.2.4.1 dc1394error_t dc1394_capture_dequeue (dc1394camera_t * *camera*, dc1394capture_policy_t *policy*, dc1394video_frame_t ** *frame*)

Captures a video frame. The returned struct contains the image buffer, among others. This image buffer SHALL NOT be freed, as it represents an area in the memory that belongs to the system.

4.2.4.2 dc1394error_t dc1394_capture_enqueue (dc1394camera_t * *camera*, dc1394video_frame_t * *frame*)

Returns a frame to the ring buffer once it has been used.

4.2.4.3 int dc1394_capture_get_fileno (dc1394camera_t * *camera*)

Gets a file descriptor to be used for select(). Must be called after **dc1394_capture_setup()** (p. 72).

4.2.4.4 dc1394bool_t dc1394_capture_is_frame_corrupt (dc1394camera_t * *camera*, dc1394video_frame_t * *frame*)

Returns DC1394_TRUE if the given frame (previously dequeued) has been detected to be corrupt (missing data, corrupted data, overrun buffer, etc.). Note that certain types of corruption may go undetected in which case DC1394_FALSE will be returned. The ability to detect corruption also varies between platforms. Note that corrupt frames still need to be enqueued with **dc1394_capture_enqueue()** (p. 72) when no longer needed by the user.

4.2.4.5 dc1394error_t dc1394_capture_setup (dc1394camera_t * *camera*, uint32_t *num_dma_buffers*, uint32_t *flags*)

Setup the capture, using a ring buffer of a certain size (num_dma_buffers) and certain options (flags)

4.2.4.6 dc1394error_t dc1394_capture_stop (dc1394camera_t * *camera*)

Stop the capture

4.3 dc1394/control.h File Reference

Diverse controls of camera features.

```
#include <dc1394/log.h>
```

Data Structures

- struct **dc1394feature_modes_t**
- struct **dc1394trigger_modes_t**
- struct **dc1394trigger_sources_t**
- struct **__dc1394feature_info_t_struct**
- struct **__dc1394featureset_t**

Typedefs

- typedef struct **__dc1394feature_info_t_struct dc1394feature_info_t**
- typedef struct **__dc1394featureset_t dc1394featureset_t**

Enumerations

- enum **dc1394trigger_mode_t**
- enum **dc1394feature_t**
- enum **dc1394trigger_source_t**
- enum **dc1394trigger_polarity_t**
- enum **dc1394feature_mode_t**

Functions

- dc1394error_t **dc1394_feature_get_all** (**dc1394camera_t** ∗camera, **dc1394featureset_t** ∗features)
- dc1394error_t **dc1394_feature_get** (**dc1394camera_t** ∗camera, **dc1394feature_info_t** ∗feature)
- dc1394error_t **dc1394_feature_print** (**dc1394feature_info_t** ∗feature, FILE ∗fd)
- dc1394error_t **dc1394_feature_print_all** (**dc1394featureset_t** ∗features, FILE ∗fd)
- dc1394error_t **dc1394_feature_whitebalance_get_value** (**dc1394camera_t** ∗camera, uint32_t ∗u_b_value, uint32_t ∗v_r_value)
- dc1394error_t **dc1394_feature_whitebalance_set_value** (**dc1394camera_t** ∗camera, uint32_t u_b_value, uint32_t v_r_value)
- dc1394error_t **dc1394_feature_temperature_get_value** (**dc1394camera_t** ∗camera, uint32_t ∗target_temperature, uint32_t ∗temperature)
- dc1394error_t **dc1394_feature_temperature_set_value** (**dc1394camera_t** ∗camera, uint32_t target_temperature)
- dc1394error_t **dc1394_feature_whiteshading_get_value** (**dc1394camera_t** ∗camera, uint32_t ∗r_value, uint32_t ∗g_value, uint32_t ∗b_value)
- dc1394error_t **dc1394_feature_whiteshading_set_value** (**dc1394camera_t** ∗camera, uint32_t r_value, uint32_t g_value, uint32_t b_value)
- dc1394error_t **dc1394_feature_get_value** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, uint32_t ∗value)
- dc1394error_t **dc1394_feature_set_value** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, uint32_t value)

- dc1394error_t **dc1394_feature_is_present** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394bool_t** ∗value)
- dc1394error_t **dc1394_feature_is_readable** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394bool_t** ∗value)
- dc1394error_t **dc1394_feature_get_boundaries** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, uint32_t ∗min, uint32_t ∗max)
- dc1394error_t **dc1394_feature_is_switchable** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394bool_t** ∗value)
- dc1394error_t **dc1394_feature_get_power** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394switch_t** ∗pwr)
- dc1394error_t **dc1394_feature_set_power** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_feature_get_modes** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394feature_modes_t** ∗modes)
- dc1394error_t **dc1394_feature_get_mode** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394feature_mode_t** ∗mode)
- dc1394error_t **dc1394_feature_set_mode** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394feature_mode_t** mode)
- dc1394error_t **dc1394_feature_has_absolute_control** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394bool_t** ∗value)
- dc1394error_t **dc1394_feature_get_absolute_boundaries** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, float ∗min, float ∗max)
- dc1394error_t **dc1394_feature_get_absolute_value** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, float ∗value)
- dc1394error_t **dc1394_feature_set_absolute_value** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, float value)
- dc1394error_t **dc1394_feature_get_absolute_control** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394switch_t** ∗pwr)
- dc1394error_t **dc1394_feature_set_absolute_control** (**dc1394camera_t** ∗camera, **dc1394feature_t** feature, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_external_trigger_set_polarity** (**dc1394camera_t** ∗camera, **dc1394trigger_polarity_t** polarity)
- dc1394error_t **dc1394_external_trigger_get_polarity** (**dc1394camera_t** ∗camera, **dc1394trigger_polarity_t** ∗polarity)
- dc1394error_t **dc1394_external_trigger_has_polarity** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗polarity_capable)
- dc1394error_t **dc1394_external_trigger_set_power** (**dc1394camera_t** ∗camera, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_external_trigger_get_power** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗pwr)
- dc1394error_t **dc1394_external_trigger_set_mode** (**dc1394camera_t** ∗camera, **dc1394trigger_mode_t** mode)
- dc1394error_t **dc1394_external_trigger_get_mode** (**dc1394camera_t** ∗camera, **dc1394trigger_mode_t** ∗mode)
- dc1394error_t **dc1394_external_trigger_set_source** (**dc1394camera_t** ∗camera, **dc1394trigger_source_t** source)
- dc1394error_t **dc1394_external_trigger_get_source** (**dc1394camera_t** ∗camera, **dc1394trigger_source_t** ∗source)
- dc1394error_t **dc1394_external_trigger_get_supported_sources** (**dc1394camera_t** ∗camera, **dc1394trigger_sources_t** ∗sources)
- dc1394error_t **dc1394_software_trigger_set_power** (**dc1394camera_t** ∗camera, **dc1394switch_t** pwr)

- dc1394error_t **dc1394_software_trigger_get_power** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗pwr)
- dc1394error_t **dc1394_pio_set** (**dc1394camera_t** ∗camera, uint32_t value)
- dc1394error_t **dc1394_pio_get** (**dc1394camera_t** ∗camera, uint32_t ∗value)
- dc1394error_t **dc1394_camera_reset** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_camera_set_power** (**dc1394camera_t** ∗camera, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_memory_busy** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗value)
- dc1394error_t **dc1394_memory_save** (**dc1394camera_t** ∗camera, uint32_t channel)
- dc1394error_t **dc1394_memory_load** (**dc1394camera_t** ∗camera, uint32_t channel)

4.3.1 Detailed Description

Diverse controls of camera features.

Author

Gord Peters: main writer
Chris Urmson: some additions
Damien Douxchamps: some additions
Peter Antoniac: documentation maintainer
Rudolf Leitgeb: documentation writer

This is the main include file of the library. It lists most of the library functions, enumerations and data structures.

4.3.2 Typedef Documentation

4.3.2.1 typedef struct __dc1394feature_info_t_struct dc1394feature_info_t

A structure containing all information about a feature.

Some fields are only valid for some features (e.g. trigger, white balance,...)

4.3.2.2 typedef struct __dc1394featureset_t dc1394featureset_t

The list of features

4.3.3 Enumeration Type Documentation

4.3.3.1 enum dc1394feature_mode_t

Control modes for a feature (excl. absolute control)

4.3.3.2 enum dc1394feature_t

Enumeration of camera features

4.3.3.3 enum dc1394trigger_mode_t

Enumeration of trigger modes

4.3.3.4 enum dc1394trigger_polarity_t

External trigger polarity

4.3.3.5 enum dc1394trigger_source_t

Enumeration of trigger sources

4.3.4 Function Documentation

4.3.4.1 dc1394error_t dc1394_camera_reset (dc1394camera_t * *camera*)

reset a camera to factory default settings

4.3.4.2 dc1394error_t dc1394_camera_set_power (dc1394camera_t * *camera*, dc1394switch_t *pwr*)

turn a camera on or off

4.3.4.3 dc1394error_t dc1394_external_trigger_get_mode (dc1394camera_t * *camera*, dc1394trigger_mode_t * *mode*)

Gets the external trigger mode

4.3.4.4 dc1394error_t dc1394_external_trigger_get_polarity (dc1394camera_t * *camera*, dc1394trigger_polarity_t * *polarity*)

Gets the polarity of the external trigger

4.3.4.5 dc1394error_t dc1394_external_trigger_get_power (dc1394camera_t * *camera*, dc1394switch_t * *pwr*)

Gets the status of the external trigger

4.3.4.6 dc1394error_t dc1394_external_trigger_get_source (dc1394camera_t * *camera*, dc1394trigger_source_t * *source*)

Gets the external trigger source

4.3.4.7 dc1394error_t dc1394_external_trigger_get_supported_sources (dc1394camera_t * *camera*, dc1394trigger_sources_t * *sources*)

Gets the list of available external trigger source

4.3.4.8 dc1394error_t dc1394_external_trigger_has_polarity (dc1394camera_t * *camera*, dc1394bool_t * *polarity_capable*)

Tells whether the external trigger can change its polarity or not.

4.3.4.9 dc1394error_t dc1394_external_trigger_set_mode (dc1394camera_t * *camera*, dc1394trigger_mode_t *mode*)

Sets the external trigger mode

4.3.4.10 dc1394error_t dc1394_external_trigger_set_polarity (dc1394camera_t * *camera*, dc1394trigger_polarity_t *polarity*)

Sets the polarity of the external trigger

4.3.4.11 dc1394error_t dc1394_external_trigger_set_power (dc1394camera_t * *camera*, dc1394switch_t *pwr*)

Switch between internal and external trigger

4.3.4.12 dc1394error_t dc1394_external_trigger_set_source (dc1394camera_t * *camera*, dc1394trigger_source_t *source*)

Sets the external trigger source

4.3.4.13 dc1394error_t dc1394_feature_get (dc1394camera_t * *camera*, dc1394feature_info_t * *feature*)

Stores the bounds and options associated with the feature described by feature->feature_id

4.3.4.14 dc1394error_t dc1394_feature_get_absolute_boundaries (dc1394camera_t * *camera*, dc1394feature_t *feature*, float * *min*, float * *max*)

Gets the absolute boundaries of a feature

4.3.4.15 dc1394error_t dc1394_feature_get_absolute_control (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394switch_t * *pwr*)

Gets the status of absolute control of a feature

4.3.4.16 dc1394error_t dc1394_feature_get_absolute_value (dc1394camera_t * *camera*, dc1394feature_t *feature*, float * *value*)

Gets the absolute value of a feature

4.3.4.17 dc1394error_t dc1394_feature_get_all (dc1394camera_t * *camera*, dc1394featureset_t * *features*)

Collects the available features for the camera described by node and stores them in features.

4.3.4.18 dc1394error_t dc1394_feature_get_boundaries (dc1394camera_t * *camera*, dc1394feature_t *feature*, uint32_t * *min*, uint32_t * *max*)

Gets the boundaries of a feature

4.3.4.19 dc1394error_t dc1394_feature_get_mode (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394feature_mode_t * *mode*)

Gets the current control modes for a feature

4.3.4.20 dc1394error_t dc1394_feature_get_modes (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394feature_modes_t * *modes*)

Gets the list of control modes for a feature (manual, auto, etc...)

4.3.4.21 dc1394error_t dc1394_feature_get_power (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394switch_t * *pwr*)

Gets the power status of a feature (ON or OFF)

4.3.4.22 dc1394error_t dc1394_feature_get_value (dc1394camera_t * *camera*, dc1394feature_t *feature*, uint32_t * *value*)

Gets the value of a feature

4.3.4.23 dc1394error_t dc1394_feature_has_absolute_control (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394bool_t * *value*)

Tells whether a feature can be controlled in absolute mode

4.3.4.24 dc1394error_t dc1394_feature_is_present (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394bool_t * *value*)

Tells whether a feature is present or not

4.3.4.25 dc1394error_t dc1394_feature_is_readable (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394bool_t * *value*)

Tells whether a feature is readable or not

4.3.4.26 dc1394error_t dc1394_feature_is_switchable (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394bool_t * *value*)

Tells whether a feature is switcheable or not (ON/OFF)

4.3.4.27 dc1394error_t dc1394_feature_print (dc1394feature_info_t * *feature*, FILE * *fd*)

Displays the bounds and options of the given feature

4.3.4.28 dc1394error_t dc1394_feature_print_all (dc1394featureset_t * *features*, FILE * *fd*)

Displays the bounds and options of every feature supported by the camera

4.3.4.29 dc1394error_t dc1394_feature_set_absolute_control (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394switch_t *pwr*)

Sets the feature in absolute control mode (ON/OFF)

4.3.4.30 dc1394error_t dc1394_feature_set_absolute_value (dc1394camera_t * *camera*, dc1394feature_t *feature*, float *value*)

Sets the absolute value of a feature

4.3.4.31 dc1394error_t dc1394_feature_set_mode (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394feature_mode_t *mode*)

Sets the current control modes for a feature

4.3.4.32 dc1394error_t dc1394_feature_set_power (dc1394camera_t * *camera*, dc1394feature_t *feature*, dc1394switch_t *pwr*)

Sets the power status of a feature (ON or OFF)

4.3.4.33 dc1394error_t dc1394_feature_set_value (dc1394camera_t * *camera*, dc1394feature_t *feature*, uint32_t *value*)

Sets the value of a feature

4.3.4.34 dc1394error_t dc1394_feature_temperature_get_value (dc1394camera_t * *camera*, uint32_t * *target_temperature*, uint32_t * *temperature*)

Gets the temperature values (target and current)

4.3.4.35 dc1394error_t dc1394_feature_temperature_set_value (dc1394camera_t * *camera*, uint32_t *target_temperature*)

Sets the temperature values (target only) FIXME: COULD BE DROPPED? already in the standard feature_set_value()?

4.3.4.36 dc1394error_t dc1394_feature_whitebalance_get_value (dc1394camera_t * *camera*, uint32_t * *u_b_value*, uint32_t * *v_r_value*)

Gets the whitebalance values

4.3.4.37 dc1394error_t dc1394_feature_whitebalance_set_value (dc1394camera_t * *camera*, uint32_t *u_b_value*, uint32_t *v_r_value*)

Sets the whitebalance values

4.3.4.38 dc1394error_t dc1394_feature_whiteshading_get_value (dc1394camera_t * *camera*, uint32_t * *r_value*, uint32_t * *g_value*, uint32_t * *b_value*)

Gets the white shading values

4.3.4.39 dc1394error_t dc1394_feature_whiteshading_set_value (dc1394camera_t * *camera*, uint32_t *r_value*, uint32_t *g_value*, uint32_t *b_value*)

Sets the white shading values

4.3.4.40 dc1394error_t dc1394_memory_busy (dc1394camera_t * *camera*, dc1394bool_t * *value*)

Download a camera setup from the memory.

4.3.4.41 dc1394error_t dc1394_memory_load (dc1394camera_t * *camera*, uint32_t *channel*)

Tells whether the writing of the camera setup in memory is finished or not.

4.3.4.42 dc1394error_t dc1394_memory_save (dc1394camera_t * *camera*, uint32_t *channel*)

Uploads a camera setup in the memory.

Note that this operation can only be performed a certain number of times for a given camera, as it requires reprogramming of an EEPROM.

4.3.4.43 dc1394error_t dc1394_pio_get (dc1394camera_t * *camera*, uint32_t * *value*)

Gets the current quadlet at the PIO (input)

4.3.4.44 dc1394error_t dc1394_pio_set (dc1394camera_t * *camera*, uint32_t *value*)

Sends a quadlet on the PIO (output)

4.3.4.45 dc1394error_t dc1394_software_trigger_get_power (dc1394camera_t * *camera*, dc1394switch_t * *pwr*)

Gets the state of software trigger

4.3.4.46 dc1394error_t dc1394_software_trigger_set_power (dc1394camera_t * *camera*, dc1394switch_t *pwr*)

Turn software trigger on or off

4.4 dc1394/conversions.h File Reference

functions to convert video formats

`#include <dc1394/log.h>`

Enumerations

- enum **dc1394bayer_method_t**
- enum **dc1394stereo_method_t**

Functions

- dc1394error_t **dc1394_convert_to_YUV422** (uint8_t *src, uint8_t *dest, uint32_t width, uint32_t height, uint32_t byte_order, **dc1394color_coding_t** source_coding, uint32_t bits)
- dc1394error_t **dc1394_convert_to_MONO8** (uint8_t *src, uint8_t *dest, uint32_t width, uint32_t height, uint32_t byte_order, **dc1394color_coding_t** source_coding, uint32_t bits)
- dc1394error_t **dc1394_convert_to_RGB8** (uint8_t *src, uint8_t *dest, uint32_t width, uint32_t height, uint32_t byte_order, **dc1394color_coding_t** source_coding, uint32_t bits)
- dc1394error_t **dc1394_convert_packed12_to_16** (const uint8_t *restrict src, uint16_t *restrict dest, uint32_t width, uint32_t height, uint8_t bits)
- dc1394error_t **dc1394_convert_packed12_to_8** (const uint8_t *restrict src, uint8_t *restrict dest, uint32_t width, uint32_t height, uint32_t bits)
- dc1394error_t **dc1394_deinterlace_stereo** (uint8_t *src, uint8_t *dest, uint32_t width, uint32_t height)
- dc1394error_t **dc1394_bayer_decoding_8bit** (const uint8_t *bayer, uint8_t *rgb, uint32_t width, uint32_t height, **dc1394color_filter_t** tile, **dc1394bayer_method_t** method)
- dc1394error_t **dc1394_bayer_decoding_16bit** (const uint16_t *bayer, uint16_t *rgb, uint32_t width, uint32_t height, **dc1394color_filter_t** tile, **dc1394bayer_method_t** method, uint32_t bits)
- dc1394error_t **dc1394_convert_frames** (**dc1394video_frame_t** *in, **dc1394video_frame_t** *out)
- dc1394error_t **dc1394_debayer_frames** (**dc1394video_frame_t** *in, **dc1394video_frame_t** *out, **dc1394bayer_method_t** method)
- dc1394error_t **dc1394_deinterlace_stereo_frames** (**dc1394video_frame_t** *in, **dc1394video_frame_t** *out, **dc1394stereo_method_t** method)

4.4.1 Detailed Description

functions to convert video formats

Author

Damien Douxchamps: coding
Frederic Devernay: coding
Peter Antoniac: documentation maintainer

More details soon

4.4.2 Enumeration Type Documentation

4.4.2.1 enum dc1394bayer_method_t

A list of de-mosaicing techniques for Bayer-patterns.

The speed of the techniques can vary greatly, as well as their quality.

4.4.2.2 enum dc1394stereo_method_t

A list of known stereo-in-normal-video modes used by manufacturers like Point Grey Research and Videre Design.

4.4.3 Function Documentation

4.4.3.1 dc1394error_t dc1394_bayer_decoding_16bit (const uint16_t * *bayer*, uint16_t * *rgb*, uint32_t *width*, uint32_t *height*, dc1394color_filter_t *tile*, dc1394bayer_method_t *method*, uint32_t *bits*)

Perform de-mosaicing on an 16-bit image buffer For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.2 dc1394error_t dc1394_bayer_decoding_8bit (const uint8_t * *bayer*, uint8_t * *rgb*, uint32_t *width*, uint32_t *height*, dc1394color_filter_t *tile*, dc1394bayer_method_t *method*)

Perform de-mosaicing on an 8-bit image buffer

4.4.3.3 dc1394error_t dc1394_convert_frames (dc1394video_frame_t * *in*, dc1394video_frame_t * *out*)

Converts the format of a video frame.

To set the format of the output, simply set the values of the corresponding fields in the output frame

4.4.3.4 dc1394error_t dc1394_convert_packed12_to_16 (const uint8_t *restrict *src*, uint16_t *restrict *dest*, uint32_t *width*, uint32_t *height*, uint8_t *bits*)

Converts 12 bit packed mono and raw to 16 bit mono and raw, respectively. For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.5 dc1394error_t dc1394_convert_packed12_to_8 (const uint8_t *restrict *src*, uint8_t *restrict *dest*, uint32_t *width*, uint32_t *height*, uint32_t *bits*)

Converts 12 bit packed mono and raw to 8 bit mono and raw, respectively For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.6 dc1394error_t dc1394_convert_to_MONO8 (uint8_t * *src*, uint8_t * *dest*, uint32_t *width*, uint32_t *height*, uint32_t *byte_order*, dc1394color_coding_t *source_coding*, uint32_t *bits*)

Converts an image buffer to MONO8 For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.7 dc1394error_t dc1394_convert_to_RGB8 (uint8_t * *src*, uint8_t * *dest*, uint32_t *width*, uint32_t *height*, uint32_t *byte_order*, dc1394color_coding_t *source_coding*, uint32_t *bits*)

Converts an image buffer to RGB8 For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.8 dc1394error_t dc1394_convert_to_YUV422 (uint8_t * *src*, uint8_t * *dest*, uint32_t *width*, uint32_t *height*, uint32_t *byte_order*, dc1394color_coding_t *source_coding*, uint32_t *bits*)

Converts an image buffer to YUV422. For images with MSB-aligned data (as produced by AVT cameras), 'bits' should be set to 0.

4.4.3.9 dc1394error_t dc1394_debayer_frames (dc1394video_frame_t * *in*, dc1394video_frame_t * *out*, dc1394bayer_method_t *method*)

De-mosaicing of a Bayer-encoded video frame

To set the format of the output, simply set the values of the corresponding fields in the output frame

Parameters

in is a pointer to the bayer video frame that is to be converted

out is a pointer to the frame to be converted to. If there is memory allocated to the image field, then it will be adjusted accordingly by this function. If there is no memory allocated to the image field, then ensure that out->image == NULL and out->allocated_image_bytes == 0

method is the bayer method to interpolate the frame.

4.4.3.10 dc1394error_t dc1394_deinterlace_stereo (uint8_t * *src*, uint8_t * *dest*, uint32_t *width*, uint32_t *height*)

changes a 16bit stereo image (8bit/channel) into two 8bit images on top of each other

4.4.3.11 dc1394error_t dc1394_deinterlace_stereo_frames (dc1394video_frame_t * *in*, dc1394video_frame_t * *out*, dc1394stereo_method_t *method*)

De-interlacing of stereo data for cideo frames

To set the format of the output, simply set the values of the corresponding fields in the output frame

4.5 dc1394/dc1394.h File Reference

Main include file, which include all others.

```
#include <dc1394/types.h>
#include <dc1394/camera.h>
#include <dc1394/control.h>
#include <dc1394/capture.h>
#include <dc1394/conversions.h>
#include <dc1394/format7.h>
#include <dc1394/iso.h>
#include <dc1394/log.h>
#include <dc1394/register.h>
#include <dc1394/video.h>
#include <dc1394/utils.h>
```

4.5.1 Detailed Description

Main include file, which include all others. More details soon

4.6 dc1394/format7.h File Reference

Functions to control Format_7 (aka scalable format, ROI).

```
#include <dc1394/log.h>
#include <dc1394/video.h>
```

Data Structures

- struct **__dc1394format7mode_t**
- struct **__dc1394format7modeset_t**

Typedefs

- typedef struct **__dc1394format7mode_t dc1394format7mode_t**
- typedef struct **__dc1394format7modeset_t dc1394format7modeset_t**

Functions

- dc1394error_t **dc1394_format7_get_max_image_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *h_size, uint32_t *v_size)
- dc1394error_t **dc1394_format7_get_unit_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *h_unit, uint32_t *v_unit)
- dc1394error_t **dc1394_format7_get_image_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *width, uint32_t *height)
- dc1394error_t **dc1394_format7_set_image_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t width, uint32_t height)
- dc1394error_t **dc1394_format7_get_image_position** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *left, uint32_t *top)
- dc1394error_t **dc1394_format7_set_image_position** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t left, uint32_t top)
- dc1394error_t **dc1394_format7_get_unit_position** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *h_unit_pos, uint32_t *v_unit_pos)
- dc1394error_t **dc1394_format7_get_color_coding** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, **dc1394color_coding_t** *color_coding)
- dc1394error_t **dc1394_format7_get_color_codings** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, **dc1394color_codings_t** *codings)
- dc1394error_t **dc1394_format7_set_color_coding** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, **dc1394color_coding_t** color_coding)
- dc1394error_t **dc1394_format7_get_color_filter** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, **dc1394color_filter_t** *color_filter)
- dc1394error_t **dc1394_format7_get_packet_parameters** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *unit_bytes, uint32_t *max_bytes)
- dc1394error_t **dc1394_format7_get_packet_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *packet_size)
- dc1394error_t **dc1394_format7_set_packet_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t packet_size)
- dc1394error_t **dc1394_format7_get_recommended_packet_size** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, uint32_t *packet_size)

- dc1394error_t **dc1394_format7_get_packets_per_frame** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, uint32_t ∗ppf)
- dc1394error_t **dc1394_format7_get_data_depth** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, uint32_t ∗data_depth)
- dc1394error_t **dc1394_format7_get_frame_interval** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, float ∗interval)
- dc1394error_t **dc1394_format7_get_pixel_number** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, uint32_t ∗pixnum)
- dc1394error_t **dc1394_format7_get_total_bytes** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, uint64_t ∗total_bytes)
- dc1394error_t **dc1394_format7_get_modeset** (**dc1394camera_t** ∗camera, **dc1394format7modeset_t** ∗info)
- dc1394error_t **dc1394_format7_get_mode_info** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, **dc1394format7mode_t** ∗f7_mode)
- dc1394error_t **dc1394_format7_set_roi** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, **dc1394color_coding_t** color_coding, int32_t packet_size, int32_t left, int32_t top, int32_t width, int32_t height)
- dc1394error_t **dc1394_format7_get_roi** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, **dc1394color_coding_t** ∗color_coding, uint32_t ∗packet_size, uint32_t ∗left, uint32_t ∗top, uint32_t ∗width, uint32_t ∗height)

4.6.1 Detailed Description

Functions to control Format_7 (aka scalable format, ROI). More details soon

4.6.2 Typedef Documentation

4.6.2.1 typedef struct __dc1394format7mode_t dc1394format7mode_t

A struct containing information about a mode of Format_7, the scalable image format.

4.6.2.2 typedef struct __dc1394format7modeset_t dc1394format7modeset_t

A struct containing the list of Format_7 modes. FIXME: this may become very big if format_7 pages are used in IIDC 1.32. It would be better to use a "num" and an allocated list.

4.6.3 Function Documentation

4.6.3.1 dc1394error_t dc1394_format7_get_color_coding (dc1394camera_t ∗ *camera*, dc1394video_mode_t *video_mode*, dc1394color_coding_t ∗ *color_coding*)

Gets the current color coding

4.6.3.2 dc1394error_t dc1394_format7_get_color_codings (dc1394camera_t ∗ *camera*, dc1394video_mode_t *video_mode*, dc1394color_codings_t ∗ *codings*)

Gets the list of color codings available for this mode

4.6.3.3 dc1394error_t dc1394_format7_get_color_filter (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394color_filter_t * *color_filter*)

Gets the current color filter

4.6.3.4 dc1394error_t dc1394_format7_get_data_depth (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *data_depth*)

Gets the data depth (e.g. 12, 13, 14 bits/pixel)

4.6.3.5 dc1394error_t dc1394_format7_get_frame_interval (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, float * *interval*)

Gets the frame interval in float format

4.6.3.6 dc1394error_t dc1394_format7_get_image_position (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *left*, uint32_t * *top*)

Gets the current image position

4.6.3.7 dc1394error_t dc1394_format7_get_image_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *width*, uint32_t * *height*)

Gets the current image size.

4.6.3.8 dc1394error_t dc1394_format7_get_max_image_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *h_size*, uint32_t * *v_size*)

Gets the maximal image size for a given mode.

4.6.3.9 dc1394error_t dc1394_format7_get_mode_info (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394format7mode_t * *f7_mode*)

Gets the properties of a Format_7 mode

4.6.3.10 dc1394error_t dc1394_format7_get_modeset (dc1394camera_t * *camera*, dc1394format7modeset_t * *info*)

Gets the properties of all Format_7 modes supported by the camera.

4.6.3.11 dc1394error_t dc1394_format7_get_packet_parameters (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *unit_bytes*, uint32_t * *max_bytes*)

Get the parameters of the packet size: its maximal size and its unit size. The packet size is always a multiple of the unit bytes and cannot be zero.

4.6.3.12 dc1394error_t dc1394_format7_get_packet_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *packet_size*)

Gets the current packet size

4.6.3.13 dc1394error_t dc1394_format7_get_packets_per_frame (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *ppf*)

Gets the number of packets per frame.

4.6.3.14 dc1394error_t dc1394_format7_get_pixel_number (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *pixnum*)

Gets the number of pixels per image frame

4.6.3.15 dc1394error_t dc1394_format7_get_recommended_packet_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *packet_size*)

Gets the recommended packet size. Ignore if zero.

4.6.3.16 dc1394error_t dc1394_format7_get_roi (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394color_coding_t * *color_coding*, uint32_t * *packet_size*, uint32_t * *left*, uint32_t * *top*, uint32_t * *width*, uint32_t * *height*)

Joint function that fully gets a certain ROI taking all parameters into account.

4.6.3.17 dc1394error_t dc1394_format7_get_total_bytes (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint64_t * *total_bytes*)

Get the total number of bytes per frame. This includes padding (to reach an entire number of packets)

4.6.3.18 dc1394error_t dc1394_format7_get_unit_position (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *h_unit_pos*, uint32_t * *v_unit_pos*)

Gets the unit positions for a given mode. The image position can only be a multiple of the unit position (zero is acceptable).

4.6.3.19 dc1394error_t dc1394_format7_get_unit_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t * *h_unit*, uint32_t * *v_unit*)

Gets the unit sizes for a given mode. The image size can only be a multiple of the unit size, and cannot be smaller than it.

4.6.3.20 dc1394error_t dc1394_format7_set_color_coding (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394color_coding_t *color_coding*)

Sets the current color coding

4.6.3.21 dc1394error_t dc1394_format7_set_image_position (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t *left*, uint32_t *top*)

Sets the current image position

4.6.3.22 dc1394error_t dc1394_format7_set_image_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t *width*, uint32_t *height*)

Sets the current image size

4.6.3.23 dc1394error_t dc1394_format7_set_packet_size (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, uint32_t *packet_size*)

Sets the current packet size

4.6.3.24 dc1394error_t dc1394_format7_set_roi (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394color_coding_t *color_coding*, int32_t *packet_size*, int32_t *left*, int32_t *top*, int32_t *width*, int32_t *height*)

Joint function that fully sets a certain ROI taking all parameters into account. Note that this function does not SWITCH to the video mode passed as argument, it mearly sets it

4.7 dc1394/iso.h File Reference

Functions to manually manage the ISO resources (channels and bandwidth).

```
#include <dc1394/log.h>
```

Functions

- dc1394error_t **dc1394_iso_set_persist** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_iso_allocate_channel** (**dc1394camera_t** ∗camera, uint64_t channels_allowed, int ∗channel)
- dc1394error_t **dc1394_iso_release_channel** (**dc1394camera_t** ∗camera, int channel)
- dc1394error_t **dc1394_iso_allocate_bandwidth** (**dc1394camera_t** ∗camera, int bandwidth_units)
- dc1394error_t **dc1394_iso_release_bandwidth** (**dc1394camera_t** ∗camera, int bandwidth_units)
- dc1394error_t **dc1394_iso_release_all** (**dc1394camera_t** ∗camera)

4.7.1 Detailed Description

Functions to manually manage the ISO resources (channels and bandwidth).

Author

Damien Douxchamps: coding
Peter Antoniac: documentation maintainer

More details soon

4.7.2 Function Documentation

4.7.2.1 dc1394error_t dc1394_iso_allocate_bandwidth (dc1394camera_t ∗ *camera*, int *bandwidth_units*)

dc1394_iso_allocate_bandwidth:

Parameters

camera A camera handle.

bandwidth_units The number of isochronous bandwidth units to allocate.

Allocates isochronous bandwidth. This functions allocates bandwidth _in addition_ to any previous allocations. It may be called multiple times. The bandwidth is automatically re-allocated if there is a bus reset. The bandwidth is automatically released if this camera is freed or the application shuts down prematurely. If the bandwidth needs to persist beyond the lifetime of this application, call ***dc1394_iso_set_persist()*** *(p. 93)* first.

Returns

DC1394_SUCCESS if the operation succeeded. *DC1394_FUNCTION_NOT_SUPPORTED* if the current driver/platform does not allow bandwidth allocation. *DC1394_NO_BANDWIDTH* if there is not enough available bandwidth to support the allocation. In this case, no bandwidth is allocated.

4.7.2.2 dc1394error_t dc1394_iso_allocate_channel (dc1394camera_t * *camera*, uint64_t *channels_allowed*, int * *channel*)

dc1394_iso_allocate_channel:

Parameters

camera A camera handle.

channels_allowed A bitmask of acceptable channels for the allocation. The LSB corresponds to channel 0 and the MSB corresponds to channel 63. Only channels whose bit is set will be considered for the allocation. If *channels_allowed* = 0, the complete set of channels supported by this camera will be considered for the allocation.

channel The allocated channel number is returned here.

Allocates an isochronous channel. This function may be called multiple times, each time allocating an additional channel. The channel is automatically re-allocated if there is a bus reset. The channel is automatically released when this dc1394camera_t is freed or if the application shuts down prematurely. If the channel needs to persist beyond the lifetime of this application, call ***dc1394_iso_set_persist()*** *(p. 93)* first. Note that this function does _not_ automatically program *camera* to use the allocated channel for isochronous streaming. You must do that manually using ***dc1394_video_set_iso_channel()*** *(p. 132)*.

Returns

DC1394_SUCCESS if the operation succeeded. The allocated channel is stored in *channel*. *DC1394_FUNCTION_NOT_SUPPORTED* if the current driver/platform does not allow channel allocation. *DC1394_NO_ISO_CHANNEL* if none of the requested channels are available.

4.7.2.3 dc1394error_t dc1394_iso_release_all (dc1394camera_t * *camera*)

dc1394_iso_release_all:

Parameters

camera A camera handle.

Releases all channels and bandwidth that have been previously allocated for this dc1394camera_t. Note that this information can only be tracked per process, and there is no knowledge of allocations for this camera by previous processes. To release resources in such a case, the manual release functions ***dc1394_iso_release_channel()*** *(p. 93)* and ***dc1394_iso_release_bandwidth()*** *(p. 92)* must be used.

Returns

DC1394_SUCCESS if the operation succeeded. *DC1394_FAILURE* if some resources were not able to be released.

4.7.2.4 dc1394error_t dc1394_iso_release_bandwidth (dc1394camera_t * *camera*, int *bandwidth_units*)

dc1394_iso_release_bandwidth:

Parameters

camera A camera handle.

bandwidth_units The number of isochronous bandwidth units to free.

Releases previously allocated isochronous bandwidth. Each *dc1394camera_t* keeps track of a running total of bandwidth that has been allocated. Released bandwidth is subtracted from this total for the sake of automatic re-allocation and automatic release on shutdown. It is also acceptable for a camera to release more bandwidth than it has allocated (to clean up for another process for example). In this case, the running total of bandwidth is not affected. It is acceptable to release more bandwidth than is allocated in total for the bus. In this case, all bandwidth is released and the function succeeds.

Returns

DC1394_SUCCESS if the operation succeeded. *DC1394_FUNCTION_NOT_SUPPORTED* if the current driver/platform does not allow bandwidth release.

4.7.2.5 dc1394error_t dc1394_iso_release_channel (dc1394camera_t * *camera*, int *channel*)

dc1394_iso_release_channel:

Parameters

camera A camera handle.

channel The channel number to release.

Releases a previously allocated channel. It is acceptable to release channels that were allocated by a different process or host. If attempting to release a channel that is already released, the function will succeed.

Returns

DC1394_SUCCESS if the operation succeeded. *DC1394_FUNCTION_NOT_SUPPORTED* if the current driver/platform does not allow channel release.

4.7.2.6 dc1394error_t dc1394_iso_set_persist (dc1394camera_t * *camera*)

dc1394_iso_set_persist

Parameters

camera A camera handle.

Calling this function will cause isochronous channel and bandwidth allocations to persist beyond the lifetime of this dc1394camera_t instance. Normally (when this function is not called), any allocations would be automatically released upon freeing this camera or a premature shutdown of the application (if possible). For this function to be used, it must be called prior to any allocations or an error will be returned.

Returns

DC1394_SUCCESS if the operation succeeded. *DC1394_FUNCTION_NOT_SUPPORTED* if the current platform/driver does not allow persistent allocations.

4.8 dc1394/register.h File Reference

Functions to directly access camera registers.

Functions

- dc1394error_t **dc1394_get_registers** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_set_registers** (**dc1394camera_t** ∗camera, uint64_t offset, const uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_get_control_registers** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_set_control_registers** (**dc1394camera_t** ∗camera, uint64_t offset, const uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_get_adv_control_registers** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_set_adv_control_registers** (**dc1394camera_t** ∗camera, uint64_t offset, const uint32_t ∗value, uint32_t num_regs)
- dc1394error_t **dc1394_get_format7_register** (**dc1394camera_t** ∗camera, unsigned int mode, uint64_t offset, uint32_t ∗value)
- dc1394error_t **dc1394_set_format7_register** (**dc1394camera_t** ∗camera, unsigned int mode, uint64_t offset, uint32_t value)
- dc1394error_t **dc1394_get_absolute_register** (**dc1394camera_t** ∗camera, unsigned int feature, uint64_t offset, uint32_t ∗value)
- dc1394error_t **dc1394_set_absolute_register** (**dc1394camera_t** ∗camera, unsigned int feature, uint64_t offset, uint32_t value)
- dc1394error_t **dc1394_get_PIO_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value)
- dc1394error_t **dc1394_set_PIO_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t value)
- dc1394error_t **dc1394_get_SIO_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value)
- dc1394error_t **dc1394_set_SIO_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t value)
- dc1394error_t **dc1394_get_strobe_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t ∗value)
- dc1394error_t **dc1394_set_strobe_register** (**dc1394camera_t** ∗camera, uint64_t offset, uint32_t value)

4.8.1 Detailed Description

Functions to directly access camera registers.

Author

Damien Douxchamps: coding
Peter Antoniac: documentation maintainer

More details soon

4.8.2 Function Documentation

4.8.2.1 dc1394error_t dc1394_get_absolute_register (dc1394camera_t * *camera*, unsigned int *feature*, uint64_t *offset*, uint32_t * *value*)

No Docs

4.8.2.2 dc1394error_t dc1394_get_adv_control_registers (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.3 dc1394error_t dc1394_get_control_registers (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.4 dc1394error_t dc1394_get_format7_register (dc1394camera_t * *camera*, unsigned int *mode*, uint64_t *offset*, uint32_t * *value*)

No Docs

4.8.2.5 dc1394error_t dc1394_get_PIO_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*)

No Docs

4.8.2.6 dc1394error_t dc1394_get_registers (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.7 dc1394error_t dc1394_get_SIO_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*)

No Docs

4.8.2.8 dc1394error_t dc1394_get_strobe_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t * *value*)

No Docs

4.8.2.9 dc1394error_t dc1394_set_absolute_register (dc1394camera_t * *camera*, unsigned int *feature*, uint64_t *offset*, uint32_t *value*)

No Docs

4.8.2.10 dc1394error_t dc1394_set_adv_control_registers (dc1394camera_t * *camera*, uint64_t *offset*, const uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.11 dc1394error_t dc1394_set_control_registers (dc1394camera_t * *camera*, uint64_t *offset*, const uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.12 dc1394error_t dc1394_set_format7_register (dc1394camera_t * *camera*, unsigned int *mode*, uint64_t *offset*, uint32_t *value*)

No Docs

4.8.2.13 dc1394error_t dc1394_set_PIO_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t *value*)

No Docs

4.8.2.14 dc1394error_t dc1394_set_registers (dc1394camera_t * *camera*, uint64_t *offset*, const uint32_t * *value*, uint32_t *num_regs*)

No Docs

4.8.2.15 dc1394error_t dc1394_set_SIO_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t *value*)

No Docs

4.8.2.16 dc1394error_t dc1394_set_strobe_register (dc1394camera_t * *camera*, uint64_t *offset*, uint32_t *value*)

No Docs

4.9 dc1394/types.h File Reference

Various types that must be defined here.

```
#include <stdint.h>
```

Data Structures

- struct **dc1394color_codings_t**
- struct **dc1394video_modes_t**

Enumerations

- enum **dc1394video_mode_t**
- enum **dc1394color_coding_t**
- enum **dc1394color_filter_t**
- enum **dc1394byte_order_t**
- enum **dc1394bool_t**
- enum **dc1394switch_t**

4.9.1 Detailed Description

Various types that must be defined here. More details soon

4.9.2 Enumeration Type Documentation

4.9.2.1 enum dc1394bool_t

Yet another boolean data type

4.9.2.2 enum dc1394byte_order_t

Byte order for YUV formats (may be expanded to RGB in the future)

IIDC cameras always return data in UYVY order, but conversion functions can change this if requested.

4.9.2.3 enum dc1394color_coding_t

Enumeration of colour codings. For details on the data format please read the IIDC specifications.

4.9.2.4 enum dc1394color_filter_t

RAW sensor filters. These elementary tiles tesselate the image plane in RAW modes. RGGB should be interpreted in 2D as

RG GB

and similarly for other filters.

4.9.2.5 enum dc1394switch_t

Yet another boolean data type, a bit more oriented towards electrical-engineers

4.9.2.6 enum dc1394video_mode_t

Enumeration of video modes. Note that the notion of IIDC "format" is not present here, except in the format_7 name.

4.10 dc1394/utils.h File Reference

Utility functions.

Functions

- dc1394error_t **dc1394_get_image_size_from_video_mode** (**dc1394camera_t** ∗camera, uint32_t video_mode, uint32_t ∗width, uint32_t ∗height)
- dc1394error_t **dc1394_framerate_as_float** (**dc1394framerate_t** framerate_enum, float ∗framerate)
- dc1394error_t **dc1394_get_color_coding_data_depth** (**dc1394color_coding_t** color_coding, uint32_t ∗bits)
- dc1394error_t **dc1394_get_color_coding_bit_size** (**dc1394color_coding_t** color_coding, uint32_t ∗bits)
- dc1394error_t **dc1394_get_color_coding_from_video_mode** (**dc1394camera_t** ∗camera, **dc1394video_mode_t** video_mode, **dc1394color_coding_t** ∗color_coding)
- dc1394error_t **dc1394_is_color** (**dc1394color_coding_t** color_mode, **dc1394bool_t** ∗is_color)
- **dc1394bool_t dc1394_is_video_mode_scalable** (**dc1394video_mode_t** video_mode)
- **dc1394bool_t dc1394_is_video_mode_still_image** (**dc1394video_mode_t** video_mode)
- **dc1394bool_t dc1394_is_same_camera** (**dc1394camera_id_t** id1, **dc1394camera_id_t** id2)
- const char ∗ **dc1394_feature_get_string** (**dc1394feature_t** feature)
- const char ∗ **dc1394_error_get_string** (dc1394error_t error)
- uint16_t **dc1394_checksum_crc16** (const uint8_t ∗buffer, uint32_t buffer_size)

4.10.1 Detailed Description

Utility functions.

Author

Damien Douxchamps: coding
Peter Antoniac: documentation maintainer

More details soon

4.10.2 Function Documentation

4.10.2.1 uint16_t dc1394_checksum_crc16 (const uint8_t ∗ *buffer*, uint32_t *buffer_size*)

Calculates the CRC16 checksum of a memory region. Useful to verify the CRC of an image buffer, for instance.

4.10.2.2 const char∗ dc1394_error_get_string (dc1394error_t *error*)

Returns a descriptive string for an error code

4.10.2.3 const char∗ dc1394_feature_get_string (dc1394feature_t *feature*)

Returns a descriptive name for a feature

4.10.2.4 dc1394error_t dc1394_framerate_as_float (dc1394framerate_t *framerate_enum*, float * *framerate*)

Returns the given framerate as a float.

4.10.2.5 dc1394error_t dc1394_get_color_coding_bit_size (dc1394color_coding_t *color_coding*, uint32_t * *bits*)

Returns the bit-space used by a pixel. This is different from the data depth! For instance, RGB16 has a bit space of 48 bits, YUV422 is 16bits and YU411 is 12bits.

4.10.2.6 dc1394error_t dc1394_get_color_coding_data_depth (dc1394color_coding_t *color_coding*, uint32_t * *bits*)

Returns the number of bits per pixel for a certain color coding. This is the size of the data sent on the bus, the effective data depth may vary. Example: RGB16 is 16, YUV411 is 8, YUV422 is 8.

4.10.2.7 dc1394error_t dc1394_get_color_coding_from_video_mode (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394color_coding_t * *color_coding*)

Returns the color coding from the video mode. Works with scalable image formats too.

4.10.2.8 dc1394error_t dc1394_get_image_size_from_video_mode (dc1394camera_t * *camera*, uint32_t *video_mode*, uint32_t * *width*, uint32_t * *height*)

Returns the image width and height (in pixels) corresponding to a video mode. Works for scalable and non-scalable video modes.

4.10.2.9 dc1394error_t dc1394_is_color (dc1394color_coding_t *color_mode*, dc1394bool_t * *is_color*)

Tells whether the color mode is color or monochrome

4.10.2.10 dc1394bool_t dc1394_is_same_camera (dc1394camera_id_t *id1*, dc1394camera_id_t *id2*)

Tells whether two IDs refer to the same physical camera unit.

4.10.2.11 dc1394bool_t dc1394_is_video_mode_scalable (dc1394video_mode_t *video_mode*)

Tells whether the video mode is scalable or not.

4.10.2.12 dc1394bool_t dc1394_is_video_mode_still_image (dc1394video_mode_t *video_mode*)

Tells whether the video mode is "still image" or not ("still image" is currently not supported by any cameras on the market)

4.11 dc1394/vendor/avt.h File Reference

Definition of AVT-specific features.

```
#include <stdint.h>
#include <dc1394/types.h>
#include <dc1394/log.h>
```

Data Structures

- struct **__dc1394_avt_smart_feature_info_struct**
- struct **__dc1394_avt_adv_feature_info_struct**
- struct **__dc1394_avt_sis_data_struct**

Typedefs

- typedef struct **__dc1394_avt_smart_feature_info_struct dc1394_avt_smart_feature_info_t**
- typedef struct **__dc1394_avt_adv_feature_info_struct dc1394_avt_adv_feature_info_t**
- typedef struct **__dc1394_avt_sis_data_struct dc1394_avt_sis_data**

Enumerations

- enum **dc1394_avt_family_t**
- enum **dc1394_avt_camera_id_t**

Functions

- dc1394error_t **dc1394_avt_get_version** (**dc1394camera_t** ∗camera, uint32_t ∗UCType, uint32_t ∗Version, uint32_t ∗Camera_ID, uint32_t ∗FPGA_Version)
- dc1394error_t **dc1394_avt_adjust_frames** (uint32_t Camera_ID, **dc1394video_frame_t** ∗frame)
- dc1394error_t **dc1394_avt_get_advanced_feature_inquiry** (**dc1394camera_t** ∗camera, **dc1394_avt_adv_feature_info_t** ∗adv_feature)
- dc1394error_t **dc1394_avt_get_smart_feature_inquiry** (**dc1394camera_t** ∗camera, **dc1394_avt_smart_feature_info_t** ∗smart_feature, int size)
- dc1394error_t **dc1394_avt_print_advanced_feature** (**dc1394_avt_adv_feature_info_t** ∗adv_feature)
- dc1394error_t **dc1394_avt_print_smart_features** (**dc1394_avt_smart_feature_info_t** ∗adv_feature)
- dc1394error_t **dc1394_avt_get_shading** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, **dc1394bool_t** ∗compute, **dc1394bool_t** ∗show, uint32_t ∗frame_nb)
- dc1394error_t **dc1394_avt_get_shading_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off, **dc1394bool_t** ∗build_err, **dc1394switch_t** ∗show, uint32_t ∗frame_nb, uint32_t ∗mem_channel, uint32_t ∗mem_channel_err)
- dc1394error_t **dc1394_avt_set_shading** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, **dc1394bool_t** compute, **dc1394bool_t** show, uint32_t frame_nb)
- dc1394error_t **dc1394_avt_set_shading_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off, **dc1394switch_t** compute, **dc1394switch_t** show, uint32_t frame_nb, uint32_t mem_channel, **dc1394switch_t** mem_clear, **dc1394switch_t** mem_load, **dc1394switch_t** mem_save)

- dc1394error_t **dc1394_avt_get_shading_mem_ctrl** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗en_write, **dc1394bool_t** ∗en_read, uint32_t ∗addroffset)
- dc1394error_t **dc1394_avt_set_shading_mem_ctrl** (**dc1394camera_t** ∗camera, **dc1394bool_t** en_write, **dc1394bool_t** en_read, uint32_t addroffset)
- dc1394error_t **dc1394_avt_get_shading_info** (**dc1394camera_t** ∗camera, uint32_t ∗MaxImageSize)
- dc1394error_t **dc1394_avt_get_shading_correction_info** (**dc1394camera_t** ∗camera, uint32_t ∗MaxImageSize, uint32_t ∗MemChannelCount)
- dc1394error_t **dc1394_avt_get_multiple_slope** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗points_nb, uint32_t ∗kneepoint1, uint32_t ∗kneepoint2, uint32_t ∗kneepoint3)
- dc1394error_t **dc1394_avt_set_multiple_slope** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, uint32_t points_nb, uint32_t kneepoint1, uint32_t kneepoint2, uint32_t kneepoint3)
- dc1394error_t **dc1394_avt_get_timebase** (**dc1394camera_t** ∗camera, uint32_t ∗timebase_id)
- dc1394error_t **dc1394_avt_set_timebase** (**dc1394camera_t** ∗camera, uint32_t timebase_id)
- dc1394error_t **dc1394_avt_get_extented_shutter** (**dc1394camera_t** ∗camera, uint32_t ∗timebase_id)
- dc1394error_t **dc1394_avt_set_extented_shutter** (**dc1394camera_t** ∗camera, uint32_t timebase_id)
- dc1394error_t **dc1394_avt_get_MaxResolution** (**dc1394camera_t** ∗camera, uint32_t ∗MaxHeight, uint32_t ∗MaxWidth)
- dc1394error_t **dc1394_avt_get_auto_shutter** (**dc1394camera_t** ∗camera, uint32_t ∗MinValue, uint32_t ∗MaxValue)
- dc1394error_t **dc1394_avt_set_auto_shutter** (**dc1394camera_t** ∗camera, uint32_t MinValue, uint32_t MaxValue)
- dc1394error_t **dc1394_avt_get_auto_gain** (**dc1394camera_t** ∗camera, uint32_t ∗MinValue, uint32_t ∗MaxValue)
- dc1394error_t **dc1394_avt_set_auto_gain** (**dc1394camera_t** ∗camera, uint32_t MinValue, uint32_t MaxValue)
- dc1394error_t **dc1394_avt_get_trigger_delay** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗DelayTime)
- dc1394error_t **dc1394_avt_set_trigger_delay** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, uint32_t DelayTime)
- dc1394error_t **dc1394_avt_get_mirror** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off)
- dc1394error_t **dc1394_avt_set_mirror** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off)
- dc1394error_t **dc1394_avt_get_dsnu** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗frame_nb)
- dc1394error_t **dc1394_avt_get_dsnu_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off, **dc1394bool_t** ∗build_error, uint32_t ∗frame_nb, **dc1394switch_t** ∗show_image)
- dc1394error_t **dc1394_avt_set_dsnu** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, **dc1394bool_t** compute, uint32_t frame_nb)
- dc1394error_t **dc1394_avt_set_dsnu_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off, **dc1394switch_t** compute_image, uint32_t frame_nb, **dc1394switch_t** show_image, **dc1394switch_t** load_image, **dc1394switch_t** save_image)
- dc1394error_t **dc1394_avt_get_blemish** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗frame_nb)
- dc1394error_t **dc1394_avt_get_blemish_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off, **dc1394bool_t** ∗build_error, uint32_t ∗frame_nb, **dc1394switch_t** ∗show_image)
- dc1394error_t **dc1394_avt_set_blemish** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, **dc1394bool_t** compute, uint32_t frame_nb)
- dc1394error_t **dc1394_avt_set_blemish_correction** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off, **dc1394switch_t** compute_image, uint32_t frame_nb, **dc1394switch_t** show_image, **dc1394switch_t** load_image, **dc1394switch_t** save_image)

- dc1394error_t **dc1394_avt_get_io** (**dc1394camera_t** ∗camera, uint32_t IO, **dc1394bool_t** ∗polarity, uint32_t ∗mode, **dc1394bool_t** ∗pinstate)
- dc1394error_t **dc1394_avt_set_io** (**dc1394camera_t** ∗camera, uint32_t IO, **dc1394bool_t** polarity, uint32_t mode, **dc1394bool_t** pinstate)
- dc1394error_t **dc1394_avt_get_io_pwmout_info** (**dc1394camera_t** ∗camera, uint32_t pwm_output_pin, uint32_t ∗min_period)
- dc1394error_t **dc1394_avt_get_io_pwmout** (**dc1394camera_t** ∗camera, uint32_t pwm_output_pin, uint32_t ∗period, uint32_t ∗pulse_width)
- dc1394error_t **dc1394_avt_set_io_pwmout** (**dc1394camera_t** ∗camera, uint32_t pwm_output_pin, uint32_t period, uint32_t pulse_width)
- dc1394error_t **dc1394_avt_get_io_inp_debounce_info** (**dc1394camera_t** ∗camera, uint32_t debounce_inp_pin, uint32_t ∗min_debounce_time, uint32_t ∗max_debounce_time)
- dc1394error_t **dc1394_avt_get_io_inp_debounce** (**dc1394camera_t** ∗camera, uint32_t debounce_inp_pin, uint32_t ∗debounce_time)
- dc1394error_t **dc1394_avt_set_io_inp_debounce** (**dc1394camera_t** ∗camera, uint32_t debounce_inp_pin, uint32_t debounce_time)
- dc1394error_t **dc1394_avt_reset** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_avt_get_lut** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗lutnb)
- dc1394error_t **dc1394_avt_set_lut** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, uint32_t lutnb)
- dc1394error_t **dc1394_avt_get_lut_mem_ctrl** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗en_write, uint32_t ∗AccessLutNo, uint32_t ∗addroffset)
- dc1394error_t **dc1394_avt_set_lut_mem_ctrl** (**dc1394camera_t** ∗camera, **dc1394bool_t** en_write, uint32_t AccessLutNo, uint32_t addroffset)
- dc1394error_t **dc1394_avt_get_lut_info** (**dc1394camera_t** ∗camera, uint32_t ∗NumOfLuts, uint32_t ∗MaxLutSize)
- dc1394error_t **dc1394_avt_get_lut_extd_info** (**dc1394camera_t** ∗camera, uint32_t ∗NumOfLuts, uint32_t ∗MaxValue, uint32_t ∗NumOfValues, uint32_t ∗MaxLutSize)
- dc1394error_t **dc1394_avt_get_autofunc_aoi_info** (**dc1394camera_t** ∗camera, uint32_t ∗unit_x, uint32_t ∗unit_y)
- dc1394error_t **dc1394_avt_get_aoi** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, int ∗left, int ∗top, int ∗width, int ∗height)
- dc1394error_t **dc1394_avt_get_autofunc_aoi** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off, **dc1394switch_t** ∗show_area, uint32_t ∗left, uint32_t ∗top, uint32_t ∗width, uint32_t ∗height)
- dc1394error_t **dc1394_avt_set_aoi** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, int left, int top, int width, int height)
- dc1394error_t **dc1394_avt_set_autofunc_aoi** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off, **dc1394switch_t** show_area, uint32_t left, uint32_t top, uint32_t width, uint32_t height)
- dc1394error_t **dc1394_avt_get_test_images_info** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗TestImage1, **dc1394bool_t** ∗TestImage2, **dc1394bool_t** ∗TestImage3, **dc1394bool_t** ∗TestImage4, **dc1394bool_t** ∗TestImage5, **dc1394bool_t** ∗TestImage6, **dc1394bool_t** ∗TestImage7)
- dc1394error_t **dc1394_avt_get_test_images** (**dc1394camera_t** ∗camera, uint32_t ∗image_no)
- dc1394error_t **dc1394_avt_set_test_images** (**dc1394camera_t** ∗camera, uint32_t image_no)
- dc1394error_t **dc1394_avt_get_frame_info** (**dc1394camera_t** ∗camera, uint32_t ∗framecounter)
- dc1394error_t **dc1394_avt_get_frame_counter** (**dc1394camera_t** ∗camera, const **dc1394_avt_smart_feature_info_t** ∗feature_info, uint32_t ∗framecounter)
- dc1394error_t **dc1394_avt_reset_frame_info** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_avt_reset_frame_counter** (**dc1394camera_t** ∗camera, const **dc1394_avt_smart_feature_info_t** ∗feature_info)
- dc1394error_t **dc1394_avt_get_trigger_counter** (**dc1394camera_t** ∗camera, uint32_t ∗triggercounter)

- dc1394error_t **dc1394_avt_reset_trigger_counter** (**dc1394camera_t** ∗camera)
- dc1394error_t **dc1394_avt_get_gpdata_info** (**dc1394camera_t** ∗camera, uint32_t ∗BufferSize)
- dc1394error_t **dc1394_avt_get_deferred_trans** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗HoldImage, **dc1394bool_t** ∗FastCapture, uint32_t ∗FifoSize, uint32_t ∗NumOfImages)
- dc1394error_t **dc1394_avt_set_deferred_trans** (**dc1394camera_t** ∗camera, **dc1394bool_t** HoldImage, **dc1394bool_t** FastCapture, uint32_t FifoSize, uint32_t NumOfImages, **dc1394bool_t** SendImage)
- dc1394error_t **dc1394_avt_read_gpdata** (**dc1394camera_t** ∗camera, unsigned char ∗buf, uint32_t size)
- dc1394error_t **dc1394_avt_write_gpdata** (**dc1394camera_t** ∗camera, unsigned char ∗buf, uint32_t size)
- dc1394error_t **dc1394_avt_read_shading_img** (**dc1394camera_t** ∗camera, unsigned char ∗buf, uint32_t size)
- dc1394error_t **dc1394_avt_write_shading_img** (**dc1394camera_t** ∗camera, unsigned char ∗buf, uint32_t size)
- dc1394error_t **dc1394_avt_write_lut** (**dc1394camera_t** ∗camera, uint32_t LutNo, unsigned char ∗buf, uint32_t size)
- dc1394error_t **dc1394_avt_get_channel_adjust** (**dc1394camera_t** ∗camera, int16_t ∗channel_adjust)
- dc1394error_t **dc1394_avt_set_channel_adjust** (**dc1394camera_t** ∗camera, int16_t channel_adjust)
- dc1394error_t **dc1394_avt_get_channel_adjust_offset** (**dc1394camera_t** ∗camera, int16_t ∗channel_adjust)
- dc1394error_t **dc1394_avt_set_channel_adjust_offset** (**dc1394camera_t** ∗camera, int16_t channel_adjust)
- dc1394error_t **dc1394_avt_set_color_corr** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, **dc1394bool_t** reset, int32_t Crr, int32_t Cgr, int32_t Cbr, int32_t Crg, int32_t Cgg, int32_t Cbg, int32_t Crb, int32_t Cgb, int32_t Cbb)
- dc1394error_t **dc1394_avt_get_color_corr** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, int32_t ∗Crr, int32_t ∗Cgr, int32_t ∗Cbr, int32_t ∗Crg, int32_t ∗Cgg, int32_t ∗Cbg, int32_t ∗Crb, int32_t ∗Cgb, int32_t ∗Cbb)
- dc1394error_t **dc1394_avt_get_hsnr** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗on_off, uint32_t ∗grabCount)
- dc1394error_t **dc1394_avt_set_hsnr** (**dc1394camera_t** ∗camera, **dc1394bool_t** on_off, uint32_t grabCount)
- dc1394error_t **dc1394_avt_get_sis** (**dc1394camera_t** ∗camera, const **dc1394_avt_smart_feature_info_t** ∗feature_info, **dc1394switch_t** ∗on_off, int16_t ∗linePos, uint32_t ∗userVal)
- dc1394error_t **dc1394_avt_set_sis** (**dc1394camera_t** ∗camera, const **dc1394_avt_smart_feature_info_t** ∗feature_info, **dc1394switch_t** on_off, int16_t linePos, uint32_t userVal)
- dc1394error_t **dc1394_avt_get_sis_data_inquiry** (const **dc1394_avt_smart_feature_info_t** ∗feature_info, **dc1394_avt_sis_data** ∗sis_data_inquiry)
- dc1394error_t **dc1394_avt_get_sis_data** (const **dc1394_avt_smart_feature_info_t** ∗feature_info, **dc1394video_frame_t** ∗frame, int16_t linePos, **dc1394_avt_sis_data** ∗sis_data)
- dc1394error_t **dc1394_avt_get_max_iso_size** (**dc1394camera_t** ∗camera, uint32_t speed_mode, **dc1394switch_t** ∗on_off, uint32_t ∗max_size)
- dc1394error_t **dc1394_avt_set_max_iso_size** (**dc1394camera_t** ∗camera, uint32_t speed_mode, **dc1394switch_t** on_off, **dc1394switch_t** set_to_max, uint32_t max_size)
- dc1394error_t **dc1394_avt_get_param_upd_timing** (**dc1394camera_t** ∗camera, uint32_t ∗update_timing_mode)
- dc1394error_t **dc1394_avt_set_param_upd_timing** (**dc1394camera_t** ∗camera, uint32_t update_timing_mode)
- dc1394error_t **dc1394_avt_get_low_smear** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off)

- dc1394error_t **dc1394_avt_set_low_smear** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off)
- dc1394error_t **dc1394_avt_get_low_noise_binning** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off)
- dc1394error_t **dc1394_avt_set_low_noise_binning** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off)
- dc1394error_t **dc1394_avt_get_global_res_rel_shutter** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off)
- dc1394error_t **dc1394_avt_set_global_res_rel_shutter** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off)
- dc1394error_t **dc1394_avt_get_user_profile** (**dc1394camera_t** ∗camera, uint32_t ∗profile_id, **dc1394bool_t** ∗error, uint32_t ∗err_code)
- dc1394error_t **dc1394_avt_set_user_profile** (**dc1394camera_t** ∗camera, uint32_t profile_id, **dc1394switch_t** load_profile, **dc1394switch_t** save_profile, **dc1394switch_t** set_default)
- dc1394error_t **dc1394_avt_get_led** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off)
- dc1394error_t **dc1394_avt_set_led** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_off)
- dc1394error_t **dc1394_avt_get_dpc_info** (**dc1394camera_t** ∗camera, uint32_t ∗MinThreshold, uint32_t ∗MaxThreshold, uint32_t ∗MaxSize)
- dc1394error_t **dc1394_avt_get_dpc** (**dc1394camera_t** ∗camera, **dc1394switch_t** ∗on_off, uint32_t ∗threshold, uint32_t ∗mean_value, uint32_t ∗data_size)
- dc1394error_t **dc1394_avt_set_dpc** (**dc1394camera_t** ∗camera, **dc1394switch_t** on_-off, **dc1394switch_t** build_data, **dc1394switch_t** zero_data, **dc1394switch_t** mem_save, **dc1394switch_t** mem_load, uint32_t threshold)
- dc1394error_t **dc1394_avt_read_dpc_data** (**dc1394camera_t** ∗camera, dc1394_avt_dpc_pixel_-position ∗DestBuffer, uint32_t ∗PixelCount, uint32_t BufferSize)
- dc1394error_t **dc1394_avt_write_dpc_data** (**dc1394camera_t** ∗camera, dc1394_avt_dpc_pixel_-position ∗SourceBuffer, uint32_t PixelCount)

4.11.1 Detailed Description

Definition of AVT-specific features. More details soon

4.11.2 Typedef Documentation

4.11.2.1 typedef struct __dc1394_avt_adv_feature_info_struct dc1394_avt_adv_feature_info_t

Inquiry of available advanced features - This struct is provided for backwards compatibility. New code should use dc1394_avt_adv_function_inquiry_t instead.

4.11.2.2 typedef struct __dc1394_avt_sis_data_struct dc1394_avt_sis_data

SIS data structure See also dc1394_avt_set_sis, dc1394_avt_get_sis_data_inquiry and dc1394_avt_get_-sis_data.

4.11.2.3 typedef struct __dc1394_avt_smart_feature_info_struct dc1394_avt_smart_feature_info_t

Inquiry of available advanced features - This struct is intended to be used as a replacement for dc1394_-avt_adv_feature_info_t. For initialization see **dc1394_avt_get_smart_feature_inquiry()** (p. 112).

4.11.3 Enumeration Type Documentation

4.11.3.1 enum dc1394_avt_camera_id_t

Enumeration of AVT camera IDs, as returned by dc1394_avt_get_version.

4.11.3.2 enum dc1394_avt_family_t

Enumeration of AVT camera family IDs. Value is stored in bits 24-31 of the cameras serial number.

4.11.4 Function Documentation

4.11.4.1 dc1394error_t dc1394_avt_adjust_frames (uint32_t *Camera_ID*, dc1394video_frame_t * *frame*)

Adjust the frame's color code (in some situations where raw is declared as mono). It is safe to call this function for any received frame.

4.11.4.2 dc1394error_t dc1394_avt_get_advanced_feature_inquiry (dc1394camera_t * *camera*, dc1394_avt_adv_feature_info_t * *adv_feature*)

Retrieve a list with supported features This function is provided for backwards compatibility. New code should call dc1394_avt_get_smart_feature_inquiry instead.

4.11.4.3 dc1394error_t dc1394_avt_get_aoi (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, int * *left*, int * *top*, int * *width*, int * *height*)

Get Autofunction AOI configuration (on/off and area). This function is provided for backwards compatibility - new code should call dc1394_avt_get_autofunc_aoi instead.

4.11.4.4 dc1394error_t dc1394_avt_get_auto_gain (dc1394camera_t * *camera*, uint32_t * *MinValue*, uint32_t * *MaxValue*)

Get min and max gain values for Autogain

4.11.4.5 dc1394error_t dc1394_avt_get_auto_shutter (dc1394camera_t * *camera*, uint32_t * *MinValue*, uint32_t * *MaxValue*)

Get min and max shutter values for Autoshutter

4.11.4.6 dc1394error_t dc1394_avt_get_autofunc_aoi (dc1394camera_t * *camera*, dc1394switch_t * *on_off*, dc1394switch_t * *show_area*, uint32_t * *left*, uint32_t * *top*, uint32_t * *width*, uint32_t * *height*)

Get Autofunction AOI configuration (on/off state, show area mode and area)

4.11.4.7 dc1394error_t dc1394_avt_get_autofunc_aoi_info (dc1394camera_t * *camera*, uint32_t * *unit_x*, uint32_t * *unit_y*)

Get Autofunction AOI unit sizes. Only multiples of these units are allowed for area position and size.

4.11.4.8 dc1394error_t dc1394_avt_get_blemish (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *frame_nb*)

Get Blemish mode and num of frames used for correction data computation This function is provided for backwards compatibility - new code should call dc1394_avt_get_blemish_correction instead.

4.11.4.9 dc1394error_t dc1394_avt_get_blemish_correction (dc1394camera_t * *camera*, dc1394switch_t * *on_off*, dc1394bool_t * *build_error*, uint32_t * *frame_nb*, dc1394switch_t * *show_image*)

Get Blemish Correction configuration and error status of the last correction data computation

4.11.4.10 dc1394error_t dc1394_avt_get_channel_adjust (dc1394camera_t * *camera*, int16_t * *channel_adjust*)

Channel Balance: Read channel (gain) adjust (AVT Pike)

4.11.4.11 dc1394error_t dc1394_avt_get_channel_adjust_offset (dc1394camera_t * *camera*, int16_t * *channel_adjust*)

Channel Balance: Read channel (offset) adjust (AVT Pike)

4.11.4.12 dc1394error_t dc1394_avt_get_color_corr (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, int32_t * *Crr*, int32_t * *Cgr*, int32_t * *Cbr*, int32_t * *Crg*, int32_t * *Cgg*, int32_t * *Cbg*, int32_t * *Crb*, int32_t * *Cgb*, int32_t * *Cbb*)

Get Color Correction + Coefficients

4.11.4.13 dc1394error_t dc1394_avt_get_deferred_trans (dc1394camera_t * *camera*, dc1394bool_t * *HoldImage*, dc1394bool_t * *FastCapture*, uint32_t * *FifoSize*, uint32_t * *NumOfImages*)

Get the fifo control mode

4.11.4.14 dc1394error_t dc1394_avt_get_dpc (dc1394camera_t * *camera*, dc1394switch_t * *on_off*, uint32_t * *threshold*, uint32_t * *mean_value*, uint32_t * *data_size*)

Get configuration for Defect Pixel Correction: on/off, threshold for computation, mean value and size of computed data

4.11.4.15 dc1394error_t dc1394_avt_get_dpc_info (dc1394camera_t * *camera*, uint32_t * *MinThreshold*, uint32_t * *MaxThreshold*, uint32_t * *MaxSize*)

Get info for feature Defect Pixel Correction: valid range for threshold, maximum size of defect pixel data

4.11.4.16 dc1394error_t dc1394_avt_get_dsnu (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *frame_nb*)

Get DSNU mode and num of frames used for correction data computation. This function is provided for backwards compatibility - new code should call dc1394_avt_get_dsnu_correction instead.

4.11.4.17 dc1394error_t dc1394_avt_get_dsnu_correction (dc1394camera_t * *camera*, dc1394switch_t * *on_off*, dc1394bool_t * *build_error*, uint32_t * *frame_nb*, dc1394switch_t * *show_image*)

Get DSNU correction configuration and error status of the last correction data computation.

4.11.4.18 dc1394error_t dc1394_avt_get_extented_shutter (dc1394camera_t * *camera*, uint32_t * *timebase_id*)

Get the Extented Shutter value in us

4.11.4.19 dc1394error_t dc1394_avt_get_frame_counter (dc1394camera_t * *camera*, const dc1394_avt_smart_feature_info_t * *feature_info*, uint32_t * *framecounter*)

Frame Counter: Get the number of captured frames feature_info is needed to point to a properly filled info struct - see dc1394_avt_get_smart_feature_inquiry

4.11.4.20 dc1394error_t dc1394_avt_get_frame_info (dc1394camera_t * *camera*, uint32_t * *framecounter*)

Get the number of captured frames This function is provided for backwards compatibility - new code should call dc1394_avt_get_frame_counter instead.

4.11.4.21 dc1394error_t dc1394_avt_get_global_res_rel_shutter (dc1394camera_t * *camera*, dc1394switch_t * *on_off*)

Get Global Reset Release Shutter

4.11.4.22 dc1394error_t dc1394_avt_get_gpdata_info (dc1394camera_t * *camera*, uint32_t * *BufferSize*)

Get the size of the GP Data buffer

4.11.4.23 dc1394error_t dc1394_avt_get_hsnr (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *grabCount*)

Get HSNR configuration (on/off, num of frames used for averaging)

4.11.4.24 dc1394error_t dc1394_avt_get_io (dc1394camera_t * *camera*, uint32_t *IO*, dc1394bool_t * *polarity*, uint32_t * *mode*, dc1394bool_t * *pinstate*)

Get the polarity, the mode, the state of the IO. Parameter 'IO' specifies the Input-/Output Pin and should be set to DC1394_AVT_IO_INP_CTRLx or DC1394_AVT_IO_OUTP_CTRLx

4.11.4.25 dc1394error_t dc1394_avt_get_io_inp_debounce (dc1394camera_t * *camera*, uint32_t *debounce_inp_pin*, uint32_t * *debounce_time*)

Get Debounce time for a certain input pin. debounce_inp_pin should be set to DC1394_AVT_IO_INP_-DEBOUNCE_CTRLx

4.11.4.26 dc1394error_t dc1394_avt_get_io_inp_debounce_info (dc1394camera_t * *camera*, uint32_t *debounce_inp_pin*, uint32_t * *min_debounce_time*, uint32_t * *max_debounce_time*)

Get valid Debounce time range for a certain input pin. debounce_inp_pin should be set to DC1394_AVT_-IO_INP_DEBOUNCE_CTRLx

4.11.4.27 dc1394error_t dc1394_avt_get_io_pwmout (dc1394camera_t * *camera*, uint32_t *pwm_output_pin*, uint32_t * *period*, uint32_t * *pulse_width*)

Get timebase period and pulse width for PWM (Pulse-Width Modulation) output output_pin should be set to DC1394_AVT_IO_OUTP_PWM_CTRLx

4.11.4.28 dc1394error_t dc1394_avt_get_io_pwmout_info (dc1394camera_t * *camera*, uint32_t *pwm_output_pin*, uint32_t * *min_period*)

Get minimum PWM (Pulse-Width Modulation) period in us. output_pin should be set to DC1394_AVT_-IO_OUTP_PWM_CTRLx

4.11.4.29 dc1394error_t dc1394_avt_get_led (dc1394camera_t * *camera*, dc1394switch_t * *on_off*)

Get LED configuration - controls 'SW Feature - LED' functionality to blank the camera's status LEDs

4.11.4.30 dc1394error_t dc1394_avt_get_low_noise_binning (dc1394camera_t * *camera*, dc1394switch_t * *on_off*)

Get Low Noise Binning

4.11.4.31 dc1394error_t dc1394_avt_get_low_smear (dc1394camera_t * *camera*, dc1394switch_t * *on_off*)

Get Low Smear configuration

4.11.4.32 dc1394error_t dc1394_avt_get_lut (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *lutnb*)

Get LUT (look-up table) configuration (on/off and the index of the current LUT)

4.11.4.33 dc1394error_t dc1394_avt_get_lut_extd_info (dc1394camera_t * *camera*, uint32_t * *NumOfLuts*, uint32_t * *MaxValue*, uint32_t * *NumOfValues*, uint32_t * *MaxLutSize*)

Get num of LUTs (look-up tables) present, the maximum value for lut entries, the number of values per lut and the lut size in bytes.

4.11.4.34 dc1394error_t dc1394_avt_get_lut_info (dc1394camera_t * *camera*, uint32_t * *NumOfLuts*, uint32_t * *MaxLutSize*)

Get num of LUTs (look-up tables) present and the max size This function is provided for backwards compatibility - new code should call dc1394_avt_get_lut_extd_info instead.

4.11.4.35 dc1394error_t dc1394_avt_get_lut_mem_ctrl (dc1394camera_t * *camera*, dc1394bool_t * *en_write*, uint32_t * *AccessLutNo*, uint32_t * *addroffset*)

Get memory access mode of LUT (look-up table) data

4.11.4.36 dc1394error_t dc1394_avt_get_max_iso_size (dc1394camera_t * *camera*, uint32_t *speed_mode*, dc1394switch_t * *on_off*, uint32_t * *max_size*)

Get 'Max Iso Size' settings. This feature overrides the maximum packet size for isochronuos transfers for a certain FireWire speed mode. The following speed modes are supported: 2: S400 setting 3: S800 setting 4: S1600 setting (future use) 5: S3200 setting (future use)

4.11.4.37 dc1394error_t dc1394_avt_get_MaxResolution (dc1394camera_t * *camera*, uint32_t * *MaxHeight*, uint32_t * *MaxWidth*)

Get the Max achievable resolution

4.11.4.38 dc1394error_t dc1394_avt_get_mirror (dc1394camera_t * *camera*, dc1394bool_t * *on_off*)

Get 'Mirror Image' configuration (horizontal)

4.11.4.39 dc1394error_t dc1394_avt_get_multiple_slope (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *points_nb*, uint32_t * *kneepoint1*, uint32_t * *kneepoint2*, uint32_t * *kneepoint3*)

Get HDR mode (multiple slope) configuration (on/off, the nb of kneepoints used and kneepoints values)

4.11.4.40 dc1394error_t dc1394_avt_get_param_upd_timing (dc1394camera_t * *camera*, uint32_t * *update_timing_mode*)

Get Parameter Update Timing. The following modes are supported: 0: Standard Parameter Update Timing 2: Quick Format Change Mode - A running image integration will be stopped when new settings have to be applied

4.11.4.41 dc1394error_t dc1394_avt_get_shading (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, dc1394bool_t * *compute*, dc1394bool_t * *show*, uint32_t * *frame_nb*)

Get Shading Correction settings. This function is provided for backwards compatibility - new code should call dc1394_avt_get_shading_correction instead.

4.11.4.42 dc1394error_t dc1394_avt_get_shading_correction (dc1394camera_t * *camera*, dc1394switch_t * *on_off*, dc1394bool_t * *build_err*, dc1394switch_t * *show*, uint32_t * *frame_nb*, uint32_t * *mem_channel*, uint32_t * *mem_channel_err*)

Get Shading Correction settings

4.11.4.43 dc1394error_t dc1394_avt_get_shading_correction_info (dc1394camera_t * *camera*, uint32_t * *MaxImageSize*, uint32_t * *MemChannelCount*)

Retrieve the max size of a Shading Correction image and the number of supported memory channels to store shading images inside the camera.

4.11.4.44 dc1394error_t dc1394_avt_get_shading_info (dc1394camera_t * *camera*, uint32_t * *MaxImageSize*)

Retrieve the max size of a Shading Correction image This function is provided for backwards compatibility - new code should call dc1394_avt_get_shading_correction_info instead.

4.11.4.45 dc1394error_t dc1394_avt_get_shading_mem_ctrl (dc1394camera_t * *camera*, dc1394bool_t * *en_write*, dc1394bool_t * *en_read*, uint32_t * *addroffset*)

Get the current access mode (none/read/write) for Shading Correction image

4.11.4.46 dc1394error_t dc1394_avt_get_sis (dc1394camera_t * *camera*, const dc1394_avt_smart_feature_info_t * *feature_info*, dc1394switch_t * *on_off*, int16_t * *linePos*, uint32_t * *userVal*)

Get SIS configuration feature_info is needed to point to a properly filled info struct - see dc1394_avt_-get_smart_feature_inquiry. Parameter userVal is only supported for newer camera families like Pike and Stingray and may be set to NULL.

4.11.4.47 dc1394error_t dc1394_avt_get_sis_data (const dc1394_avt_smart_feature_info_t * *feature_info*, dc1394video_frame_t * *frame*, int16_t *linePos*, dc1394_avt_sis_data * *sis_data*)

Retrieve SIS data from image data. 'linePos' needs to reflect the setting of the corresponding SIS parameter at the time the image was aquired. Not all elements of sis_data are necessarily filled with valid data, depending on the camera model. See dc1394_avt_get_sis_data_inquiry for supported elements. feature_info is needed to point to a properly filled info struct - see dc1394_avt_get_smart_feature_inquiry

4.11.4.48 dc1394error_t dc1394_avt_get_sis_data_inquiry (const dc1394_avt_smart_feature_info_t * *feature_info*, dc1394_avt_sis_data * *sis_data_inquiry*)

Get supported SIS elements for a certain camera. Supported elements of sis_data_inquiry will be set to '1', '0' otherwise. Elements marked as 'reserved' are generally unsupported. feature_info is needed to point to a properly filled info struct - see dc1394_avt_get_smart_feature_inquiry

4.11.4.49 dc1394error_t dc1394_avt_get_smart_feature_inquiry (dc1394camera_t * *camera*, dc1394_avt_smart_feature_info_t * *smart_feature*, int *size*)

Retrieve a list with supported features Parameter 'size' is expected to reflect the size of struct smart_feature in bytes. It is recommended to pass 'sizeof(dc1394_avt_smart_feature_inquiry_t)' as size.

4.11.4.50 dc1394error_t dc1394_avt_get_test_images (dc1394camera_t * *camera*, uint32_t * *image_no*)

Get Test Image configuration (Index 0 disables the feature)

4.11.4.51 dc1394error_t dc1394_avt_get_test_images_info (dc1394camera_t * *camera*, dc1394bool_t * *TestImage1*, dc1394bool_t * *TestImage2*, dc1394bool_t * *TestImage3*, dc1394bool_t * *TestImage4*, dc1394bool_t * *TestImage5*, dc1394bool_t * *TestImage6*, dc1394bool_t * *TestImage7*)

Get supported Test Images

4.11.4.52 dc1394error_t dc1394_avt_get_timebase (dc1394camera_t * *camera*, uint32_t * *timebase_id*)

Get the timebase used for 'shutter' feature. Possible values: 0: 1us 1: 2us 2: 5us 3: 10us 4: 20us 5: 50us 6: 100us 7: 200us 8: 500us 9: 1000us

4.11.4.53 dc1394error_t dc1394_avt_get_trigger_counter (dc1394camera_t * *camera*, uint32_t * *triggercounter*)

Trigger Counter: Get the number detected trigger events

4.11.4.54 dc1394error_t dc1394_avt_get_trigger_delay (dc1394camera_t * *camera*, dc1394bool_t * *on_off*, uint32_t * *DelayTime*)

Get Trigger Delay configuration (on/off and the actual delay)

4.11.4.55 dc1394error_t dc1394_avt_get_user_profile (dc1394camera_t * *camera*, uint32_t * *profile_id*, dc1394bool_t * *error*, uint32_t * *err_code*)

Get User Profile settings. This feature is an AVT-specific extension of the standard IIDC memory feature.

4.11.4.56 dc1394error_t dc1394_avt_get_version (dc1394camera_t * *camera*, uint32_t * *UCType*, uint32_t * *Version*, uint32_t * *Camera_ID*, uint32_t * *FPGA_Version*)

Retrieve the firmware version, FPGA version and the camera ID

4.11.4.57 dc1394error_t dc1394_avt_print_advanced_feature (dc1394_avt_adv_feature_info_t * *adv_feature*)

Print the list of supported features See also dc1394_avt_get_advanced_feature_inquiry

4.11.4.58 dc1394error_t dc1394_avt_print_smart_features (dc1394_avt_smart_feature_info_t * *adv_feature*)

Print the list of supported features See also dc1394_avt_get_smart_feature_inquiry

4.11.4.59 dc1394error_t dc1394_avt_read_dpc_data (dc1394camera_t * *camera*, dc1394_avt_dpc_pixel_position * *DestBuffer*, uint32_t * *PixelCount*, uint32_t *BufferSize*)

Download Defect Pixel Correction data to PC. PixelCount will return the number of pixels that have been read. BufferSize indicates the size of DestBuffer in Pixels

4.11.4.60 dc1394error_t dc1394_avt_read_gpdata (dc1394camera_t * *camera*, unsigned char * *buf*, uint32_t *size*)

Read size number of bytes from GPData buffer

4.11.4.61 dc1394error_t dc1394_avt_read_shading_img (dc1394camera_t * *camera*, unsigned char * *buf*, uint32_t *size*)

Read Shading Correction image from camera into buffer

4.11.4.62 dc1394error_t dc1394_avt_reset (dc1394camera_t * *camera*)

Reset the bus and the fpga

4.11.4.63 dc1394error_t dc1394_avt_reset_frame_counter (dc1394camera_t * *camera*, const dc1394_avt_smart_feature_info_t * *feature_info*)

Reset Frame Counter feature_info is needed to point to a properly filled info struct - see dc1394_avt_get_-smart_feature_inquiry

4.11.4.64 dc1394error_t dc1394_avt_reset_frame_info (dc1394camera_t * *camera*)

Reset Frame Counter This function is provided for backwards compatibility - new code should call dc1394_avt_reset_frame_counter instead.

4.11.4.65 dc1394error_t dc1394_avt_reset_trigger_counter (dc1394camera_t * *camera*)

Reset trigger counter

4.11.4.66 dc1394error_t dc1394_avt_set_aoi (dc1394camera_t * *camera*, dc1394bool_t *on_off*, int *left*, int *top*, int *width*, int *height*)

Set Autofunction AOI configuration (on/off and area). This function is provided for backwards compatibility - new code should call dc1394_avt_set_autofunc_aoi instead.

4.11.4.67 dc1394error_t dc1394_avt_set_auto_gain (dc1394camera_t * *camera*, uint32_t *MinValue*, uint32_t *MaxValue*)

Set min and max gain values for Autogain

4.11.4.68 dc1394error_t dc1394_avt_set_auto_shutter (dc1394camera_t * *camera*, uint32_t *MinValue*, uint32_t *MaxValue*)

Set min and max shutter values for Autoshutter

4.11.4.69 dc1394error_t dc1394_avt_set_autofunc_aoi (dc1394camera_t * *camera*, dc1394switch_t *on_off*, dc1394switch_t *show_area*, uint32_t *left*, uint32_t *top*, uint32_t *width*, uint32_t *height*)

Set Autofunction AOI configuration (on/off state, show area mode and area)

4.11.4.70 dc1394error_t dc1394_avt_set_blemish (dc1394camera_t * *camera*, dc1394bool_t *on_off*, dc1394bool_t *compute*, uint32_t *frame_nb*)

Set Blemish mode, num of frames used for correction data computation and launch the the computation of correction data. This function is provided for backwards compatibility - new code should call dc1394_-avt_set_blemish_correction instead.

4.11.4.71 dc1394error_t dc1394_avt_set_blemish_correction (dc1394camera_t * *camera*, dc1394switch_t *on_off*, dc1394switch_t *compute_image*, uint32_t *frame_nb*, dc1394switch_t *show_image*, dc1394switch_t *load_image*, dc1394switch_t *save_image*)

Configure Blemish Correction. (On/Off, number of frames used for computation, launch computation, enable show_image to receive correction data instead of images, load or save data from/to flash)

4.11.4.72 dc1394error_t dc1394_avt_set_channel_adjust (dc1394camera_t * *camera*, int16_t *channel_adjust*)

Channel Balance: Write channel (gain) adjust (AVT Pike)

4.11.4.73 dc1394error_t dc1394_avt_set_channel_adjust_offset (dc1394camera_t * *camera*, int16_t *channel_adjust*)

Channel Balance: Write channel (offset) adjust (AVT Pike)

4.11.4.74 dc1394error_t dc1394_avt_set_color_corr (dc1394camera_t * *camera*, dc1394bool_t *on_off*, dc1394bool_t *reset*, int32_t *Crr*, int32_t *Cgr*, int32_t *Cbr*, int32_t *Crg*, int32_t *Cgg*, int32_t *Cbg*, int32_t *Crb*, int32_t *Cgb*, int32_t *Cbb*)

Set Color Correction + Coefficients

4.11.4.75 dc1394error_t dc1394_avt_set_deferred_trans (dc1394camera_t * *camera*, dc1394bool_t *HoldImage*, dc1394bool_t *FastCapture*, uint32_t *FifoSize*, uint32_t *NumOfImages*, dc1394bool_t *SendImage*)

Set the fifo control mode

4.11.4.76 dc1394error_t dc1394_avt_set_dpc (dc1394camera_t * *camera*, dc1394switch_t *on_off*, dc1394switch_t *build_data*, dc1394switch_t *zero_data*, dc1394switch_t *mem_save*, dc1394switch_t *mem_load*, uint32_t *threshold*)

Set configuration for Defect Pixel Correction: on/off, threshold for computation, mean value and size of computed data

4.11.4.77 dc1394error_t dc1394_avt_set_dsnu (dc1394camera_t * *camera*, dc1394bool_t *on_off*, dc1394bool_t *compute*, uint32_t *frame_nb*)

Set DSNU mode, number of frames used for correction data computation and launch the the computation of the dsnu frame This function is provided for backwards compatibility - new code should call dc1394_-avt_set_dsnu_correction instead.

4.11.4.78 dc1394error_t dc1394_avt_set_dsnu_correction (dc1394camera_t * *camera*, dc1394switch_t *on_off*, dc1394switch_t *compute_image*, uint32_t *frame_nb*, dc1394switch_t *show_image*, dc1394switch_t *load_image*, dc1394switch_t *save_image*)

Configure DSNU correction. (Enable/Disable, number of frames used for correction data computation, launch the computation of correction data, enable show_image to receive correction data instead of images,

load or save data from/to flash)

4.11.4.79 dc1394error_t dc1394_avt_set_extented_shutter (dc1394camera_t * *camera*, uint32_t *timebase_id*)

Set the Extented Shutter value in us

4.11.4.80 dc1394error_t dc1394_avt_set_global_res_rel_shutter (dc1394camera_t * *camera*, dc1394switch_t *on_off*)

Set Global Reset Release Shutter

4.11.4.81 dc1394error_t dc1394_avt_set_hsnr (dc1394camera_t * *camera*, dc1394bool_t *on_off*, uint32_t *grabCount*)

Set HSNR configuration (on/off, num of frames used for averaging)

4.11.4.82 dc1394error_t dc1394_avt_set_io (dc1394camera_t * *camera*, uint32_t *IO*, dc1394bool_t *polarity*, uint32_t *mode*, dc1394bool_t *pinstate*)

Set the polarity, the mode and the state of the IO Parameter 'IO' specifies the Input-/Output Pin and should be set to DC1394_AVT_IO_INP_CTRLx or DC1394_AVT_IO_OUTP_CTRLx

4.11.4.83 dc1394error_t dc1394_avt_set_io_inp_debounce (dc1394camera_t * *camera*, uint32_t *debounce_inp_pin*, uint32_t *debounce_time*)

Set Debounce time for a certain input pin. debounce_inp_pin should be set to DC1394_AVT_IO_INP_-DEBOUNCE_CTRLx

4.11.4.84 dc1394error_t dc1394_avt_set_io_pwmout (dc1394camera_t * *camera*, uint32_t *pwm_output_pin*, uint32_t *period*, uint32_t *pulse_width*)

Set period and pulse width for PWM (Pulse-Width Modulation) output. output_pin should be set to DC1394_AVT_IO_OUTP_PWM_CTRLx

4.11.4.85 dc1394error_t dc1394_avt_set_led (dc1394camera_t * *camera*, dc1394switch_t *on_off*)

Set LED configuration - controls 'SW Feature - LED' functionality to blank the camera's status LEDs

4.11.4.86 dc1394error_t dc1394_avt_set_low_noise_binning (dc1394camera_t * *camera*, dc1394switch_t *on_off*)

Set Low Noise Binning

4.11.4.87 dc1394error_t dc1394_avt_set_low_smear (dc1394camera_t * *camera*, dc1394switch_t *on_off*)

Set Low Smear configuration

4.11.4.88 dc1394error_t dc1394_avt_set_lut (dc1394camera_t * *camera*, dc1394bool_t *on_off*, uint32_t *lutnb*)

Set LUT (look-up table) configuration (on/off and the index of the current LUT)

4.11.4.89 dc1394error_t dc1394_avt_set_lut_mem_ctrl (dc1394camera_t * *camera*, dc1394bool_t *en_write*, uint32_t *AccessLutNo*, uint32_t *addroffset*)

Set memory access mode of LUT (look-up table) data

4.11.4.90 dc1394error_t dc1394_avt_set_max_iso_size (dc1394camera_t * *camera*, uint32_t *speed_mode*, dc1394switch_t *on_off*, dc1394switch_t *set_to_max*, uint32_t *max_size*)

Set 'Max Iso Size' settings. This feature overrides the maximum packet size for isochronuos transfers for a certain FireWire speed mode. The following speed modes are supported: 2: S400 setting 3: S800 setting 4: S1600 setting (future use) 5: S3200 setting (future use) Parameter 'set_to_max' configures the maximum iso size possible.

Caution: Activating this feature leads to bus utilization beyond FireWire Spec. Higher framerates can be achieved, but asynchronuos transfers may be delayed.

4.11.4.91 dc1394error_t dc1394_avt_set_mirror (dc1394camera_t * *camera*, dc1394bool_t *on_off*)

Set 'Mirror Image' configuration (horizontal)

4.11.4.92 dc1394error_t dc1394_avt_set_multiple_slope (dc1394camera_t * *camera*, dc1394bool_t *on_off*, uint32_t *points_nb*, uint32_t *kneepoint1*, uint32_t *kneepoint2*, uint32_t *kneepoint3*)

Set HDR mode (multiple slope) configuration (on/off, the nb of kneepoints used and kneepoints values)

4.11.4.93 dc1394error_t dc1394_avt_set_param_upd_timing (dc1394camera_t * *camera*, uint32_t *update_timing_mode*)

Set Parameter Update Timing. The following modes are supported: 0: Standard Parameter Update Timing 2: Quick Format Change Mode - A running image integration will be stopped when new settings have to be applied

4.11.4.94 dc1394error_t dc1394_avt_set_shading (dc1394camera_t * *camera*, dc1394bool_t *on_off*, dc1394bool_t *compute*, dc1394bool_t *show*, uint32_t *frame_nb*)

Set Shading Correction feature This function is provided for backwards compatibility - new code should call dc1394_avt_set_shading_correction instead.

4.11.4.95 dc1394error_t dc1394_avt_set_shading_correction (dc1394camera_t * *camera*, dc1394switch_t *on_off*, dc1394switch_t *compute*, dc1394switch_t *show*, uint32_t *frame_nb*, uint32_t *mem_channel*, dc1394switch_t *mem_clear*, dc1394switch_t *mem_load*, dc1394switch_t *mem_save*)

Set Shading Correction feature

4.11.4.96 dc1394error_t dc1394_avt_set_shading_mem_ctrl (dc1394camera_t * *camera*, dc1394bool_t *en_write*, dc1394bool_t *en_read*, uint32_t *addroffset*)

Set access mode (none/read/write) for Shading Correction image

4.11.4.97 dc1394error_t dc1394_avt_set_sis (dc1394camera_t * *camera*, const dc1394_avt_smart_feature_info_t * *feature_info*, dc1394switch_t *on_off*, int16_t *linePos*, uint32_t *userVal*)

Set SIS configuration feature_info is needed to point to a properly filled info struct - see dc1394_avt_-get_smart_feature_inquiry. Changing linePos during image acquisition may lead to synchronization issues with dc1394_avt_get_sis_data.

4.11.4.98 dc1394error_t dc1394_avt_set_test_images (dc1394camera_t * *camera*, uint32_t *image_no*)

Set Test Image configuration (Index 0 disables the feature)

4.11.4.99 dc1394error_t dc1394_avt_set_timebase (dc1394camera_t * *camera*, uint32_t *timebase_id*)

Set the timebase used for 'shutter' feature. See dc1394_avt_get_timebase for possible values.

4.11.4.100 dc1394error_t dc1394_avt_set_trigger_delay (dc1394camera_t * *camera*, dc1394bool_t *on_off*, uint32_t *DelayTime*)

Set Trigger Delay configuration (on/off and the actual delay)

4.11.4.101 dc1394error_t dc1394_avt_set_user_profile (dc1394camera_t * *camera*, uint32_t *profile_id*, dc1394switch_t *load_profile*, dc1394switch_t *save_profile*, dc1394switch_t *set_default*)

Set User Profile settings. This feature is an AVT-specific extension of the standard IIDC memory feature. The number of supported user profiles is indicated by the max_mem_channel member of dc1394camera_t.

4.11.4.102 dc1394error_t dc1394_avt_write_dpc_data (dc1394camera_t * *camera*, dc1394_avt_dpc_pixel_position * *SourceBuffer*, uint32_t *PixelCount*)

Upload Defect Pixel Correction data to camera. PixelCount indicates the number of pixels to be written.

4.11.4.103 dc1394error_t dc1394_avt_write_gpdata (dc1394camera_t * *camera*, unsigned char * *buf*, uint32_t *size*)

Write size number of bytes to GPData buffer

4.11.4.104 dc1394error_t dc1394_avt_write_lut (dc1394camera_t * *camera*, uint32_t *LutNo*, unsigned char * *buf*, uint32_t *size*)

Write LUT (look-up table) data from buffer to camera

4.11.4.105 dc1394error_t dc1394_avt_write_shading_img (dc1394camera_t * *camera*, unsigned char * *buf*, uint32_t *size*)

Write Shading Correction image from buffer to camera

4.12 dc1394/vendor/basler.h File Reference

No docs yet.

```
#include "basler_sff.h"
#include <stdint.h>
```

Functions

- dc1394error_t **dc1394_basler_sff_is_available** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗available)
- dc1394error_t **dc1394_basler_sff_feature_is_available** (**dc1394camera_t** ∗camera, **dc1394basler_sff_feature_t** feature_id, **dc1394bool_t** ∗available)
- dc1394error_t **dc1394_basler_sff_feature_enable** (**dc1394camera_t** ∗camera, **dc1394basler_sff_feature_t** feature_id, **dc1394switch_t** on_off)
- dc1394error_t **dc1394_basler_sff_feature_is_enabled** (**dc1394camera_t** ∗camera, **dc1394basler_sff_feature_t** feature_id, **dc1394bool_t** ∗is_enabled)
- **dc1394bool_t dc1394_basler_sff_check_crc** (const uint8_t ∗frame_buffer, uint32_t frame_size)
- dc1394error_t **dc1394_basler_sff_chunk_iterate_init** (**dc1394basler_sff_t** ∗chunk, void ∗frame_buffer, uint32_t frame_size, **dc1394bool_t** has_crc_checksum)
- dc1394error_t **dc1394_basler_sff_chunk_iterate** (**dc1394basler_sff_t** ∗chunk)
- dc1394error_t **dc1394_basler_sff_chunk_find** (**dc1394basler_sff_feature_t** feature_id, void ∗∗chunk_data, void ∗frame_buffer, uint32_t frame_size, **dc1394bool_t** has_crc_checksum)
- dc1394error_t **dc1394_basler_sff_feature_print** (**dc1394camera_t** ∗camera, **dc1394basler_sff_feature_t** feature_id, FILE ∗fd)
- dc1394error_t **dc1394_basler_sff_feature_print_all** (**dc1394camera_t** ∗camera, FILE ∗fd)

4.12.1 Detailed Description

No docs yet. More details soon

4.12.2 Function Documentation

4.12.2.1 dc1394bool_t dc1394_basler_sff_check_crc (const uint8_t ∗ *frame_buffer*, uint32_t *frame_size*)

Checks the crc checksum of the supplied frame

4.12.2.2 dc1394error_t dc1394_basler_sff_chunk_find (dc1394basler_sff_feature_t *feature_id*, void ∗∗ *chunk_data*, void ∗ *frame_buffer*, uint32_t *frame_size*, dc1394bool_t *has_crc_checksum*)

Finds a specific SFF chunk in the frame buffer

4.12.2.3 dc1394error_t dc1394_basler_sff_chunk_iterate (dc1394basler_sff_t ∗ *chunk*)

Iterates over the available SFF chunks in the frame buffer

4.12.2.4 dc1394error_t dc1394_basler_sff_chunk_iterate_init (dc1394basler_sff_t * *chunk*, void * *frame_buffer*, uint32_t *frame_size*, dc1394bool_t *has_crc_checksum*)

Initializes an iteration

4.12.2.5 dc1394error_t dc1394_basler_sff_feature_enable (dc1394camera_t * *camera*, dc1394basler_sff_feature_t *feature_id*, dc1394switch_t *on_off*)

Enables or disables a specific feature

4.12.2.6 dc1394error_t dc1394_basler_sff_feature_is_available (dc1394camera_t * *camera*, dc1394basler_sff_feature_t *feature_id*, dc1394bool_t * *available*)

Tests whether the camera supports the specified SFF feature

4.12.2.7 dc1394error_t dc1394_basler_sff_feature_is_enabled (dc1394camera_t * *camera*, dc1394basler_sff_feature_t *feature_id*, dc1394bool_t * *is_enabled*)

checks if a feature has been enabled

4.12.2.8 dc1394error_t dc1394_basler_sff_feature_print (dc1394camera_t * *camera*, dc1394basler_sff_feature_t *feature_id*, FILE * *fd*)

prints info about one feature

4.12.2.9 dc1394error_t dc1394_basler_sff_feature_print_all (dc1394camera_t * *camera*, FILE * *fd*)

prints info about all features

4.12.2.10 dc1394error_t dc1394_basler_sff_is_available (dc1394camera_t * *camera*, dc1394bool_t * *available*)

Tests whether the camera supports Basler SFF

4.13 dc1394/vendor/basler_sff.h File Reference

No docs yet.

```
#include <stdint.h>
```

Data Structures

- struct **dc1394basler_sff_extended_data_stream_t**
- struct **dc1394basler_sff_frame_counter_t**
- struct **dc1394basler_sff_cycle_time_stamp_t**
- struct **dc1394basler_dcam_csr_value_t**
- struct **dc1394basler_dcam_whitebalance_csr_value_t**
- struct **dc1394basler_sff_dcam_values_t**
- struct **dc1394basler_sff_crc_checksum_t**

Typedefs

- typedef enum **dc1394basler_sff_feature_t dc1394basler_sff_feature_t**
- typedef struct **dc1394basler_sff_extended_data_stream_t dc1394basler_sff_extended_data_stream_t**
- typedef struct **dc1394basler_sff_frame_counter_t dc1394basler_sff_frame_counter_t**
- typedef struct **dc1394basler_sff_cycle_time_stamp_t dc1394basler_sff_cycle_time_stamp_t**
- typedef struct **dc1394basler_dcam_csr_value_t dc1394basler_dcam_csr_value_t**
- typedef struct **dc1394basler_dcam_whitebalance_csr_value_t dc1394basler_dcam_whitebalance_csr_value_t**
- typedef struct **dc1394basler_sff_dcam_values_t dc1394basler_sff_dcam_values_t**
- typedef struct **dc1394basler_sff_crc_checksum_t dc1394basler_sff_crc_checksum_t**

Enumerations

- enum **dc1394basler_sff_feature_t** { ,

  DC1394_BASLER_SFF_EXTENDED_DATA_STREAM = 0, **DC1394_BASLER_SFF_FRAME_COUNTER**, **DC1394_BASLER_SFF_CYCLE_TIME_STAMP**, **DC1394_BASLER_SFF_DCAM_VALUES**,

  DC1394_BASLER_SFF_CRC_CHECKSUM, **DC1394_BASLER_SFF_TEST_IMAGES**, **DC1394_BASLER_SFF_EXTENDED_VERSION_INFO**, **DC1394_BASLER_SFF_LOOKUP_TABLE**,

  DC1394_BASLER_SFF_TRIGGER_FLAG_AND_COUNTER, **DC1394_BASLER_SFF_OUTPUT_PORT_0_CONFIGURATION**, **DC1394_BASLER_SFF_OUTPUT_PORT_1_CONFIGURATION**, **DC1394_BASLER_SFF_OUTPUT_PORT_2_CONFIGURATION**,

  DC1394_BASLER_SFF_OUTPUT_PORT_3_CONFIGURATION }

4.13.1 Detailed Description

No docs yet. More details soon

4.13.2 Typedef Documentation

4.13.2.1 typedef struct dc1394basler_dcam_csr_value_t dc1394basler_dcam_csr_value_t

No Docs

4.13.2.2 typedef struct dc1394basler_dcam_whitebalance_csr_value_t dc1394basler_dcam_whitebalance_csr_value_t

No Docs

4.13.2.3 typedef struct dc1394basler_sff_crc_checksum_t dc1394basler_sff_crc_checksum_t

No Docs

4.13.2.4 typedef struct dc1394basler_sff_cycle_time_stamp_t dc1394basler_sff_cycle_time_stamp_t

No Docs

4.13.2.5 typedef struct dc1394basler_sff_dcam_values_t dc1394basler_sff_dcam_values_t

No Docs

4.13.2.6 typedef struct dc1394basler_sff_extended_data_stream_t dc1394basler_sff_extended_data_stream_t

This structure is used to capture the SFF extended data stream chunk. According to the Basler manuals the extended data stream chunk also contains to members pixel_data and gap of variable size; these members are ignored in this API because they can be obtained from other sources. The pixel_data member which is the actual image frame is all data from the beginning of the frame buffer until width∗height∗bytes_per_pixel bytes. The gap is required on some cameras for technical reason but not used otherwise. The size of the gap can be computed by computing frame_size - sizeof all chunks - image_size.

4.13.2.7 typedef enum dc1394basler_sff_feature_t dc1394basler_sff_feature_t

SFF feature IDs

4.13.2.8 typedef struct dc1394basler_sff_frame_counter_t dc1394basler_sff_frame_counter_t

No Docs

4.13.3 Enumeration Type Documentation

4.13.3.1 enum dc1394basler_sff_feature_t

SFF feature IDs

Enumerator:

DC1394_BASLER_SFF_EXTENDED_DATA_STREAM SFF Extended data stream, this feature must be enabled in order to use any other smart feature. Please refer to struct dc1394basler_sff_extended_data_stream for more information.

DC1394_BASLER_SFF_FRAME_COUNTER The frame counter feature numbers images sequentially as they are captured, the counter starts at 0 and wraps at 2^32-1. The counter increments by one for each captured frame. Whenever the camera is powered off, the counter resets to 0. Please refer to struct **dc1394basler_sff_frame_counter_t** (p. 34) for more information.

DC1394_BASLER_SFF_CYCLE_TIME_STAMP The cycle time stamp feature adds a chunk to each image frame containing the value of the IEEE1394 bus cycle timer. These counters are sampled at the start of exposure of each image. Please refer to struct dc1394basler_sff_cycle_time_stamp for more information.

DC1394_BASLER_SFF_DCAM_VALUES The DCAM values smart features adds a chunk to each image containing the current settings for some standard DCAM features. The settings are sampled at the start of exposure of each image. Please refer to struct **dc1394basler_sff_dcam_values_t** (p. 32) for more information.

DC1394_BASLER_SFF_CRC_CHECKSUM The CRC checksum feature adds a chunk to each image frame containing a 16bit CRC checksum computed using the Z-modem algorithm. The checksum is computed for all the image data and all other SFF chunks except the CRC checksum chunk. Please refer to the function dc1394_basler_validate_checksum() for more information

DC1394_BASLER_SFF_TEST_IMAGES The test images feature is used to check the camera's basic functionality and its ability to transmit an image via the video data cable. The test image can be used for service purposes and for failure diagnostics. In test mode the image is generated with a software program and the camera's digital devices and does not use the optics the pixel array or the ADCs.

This feature is not implemented

DC1394_BASLER_SFF_EXTENDED_VERSION_INFO Basler cameras include a register that contains version numbers for the camera's internal software. For troubleshooting purposes, Basler technical support may ask you to read this register and to supply the results.

This feature is not implemented

DC1394_BASLER_SFF_LOOKUP_TABLE **This feature is not implemented**

DC1394_BASLER_SFF_TRIGGER_FLAG_AND_COUNTER **This feature is not implemented**

DC1394_BASLER_SFF_OUTPUT_PORT_0_CONFIGURATION **This feature is not implemented**

DC1394_BASLER_SFF_OUTPUT_PORT_1_CONFIGURATION **This feature is not implemented**

DC1394_BASLER_SFF_OUTPUT_PORT_2_CONFIGURATION **This feature is not implemented**

DC1394_BASLER_SFF_OUTPUT_PORT_3_CONFIGURATION **This feature is not implemented**

4.14 dc1394/vendor/basler_sff_registry.h File Reference

No docs yet.

```
#include "basler_sff.h"
```

Functions

- const sff_feature * **basler_sff_registry_find_by_id** (**dc1394basler_sff_feature_t** feature_id)
- const sff_feature * **basler_sff_registry_find_by_csr_guid** (**dc1394basler_sff_guid_t** *csr_guid)
- const sff_feature * **basler_sff_registry_find_by_chunk_guid** (**dc1394basler_sff_guid_t** *csr_guid)

4.14.1 Detailed Description

No docs yet. More details soon

4.14.2 Function Documentation

4.14.2.1 const sff_feature* basler_sff_registry_find_by_chunk_guid (dc1394basler_sff_guid_t * *csr_guid*)

Returns a sff feature descriptor by CHUNK guid

4.14.2.2 const sff_feature* basler_sff_registry_find_by_csr_guid (dc1394basler_sff_guid_t * *csr_guid*)

Returns a sff feature descriptor by CSR guid

4.14.2.3 const sff_feature* basler_sff_registry_find_by_id (dc1394basler_sff_feature_t *feature_id*)

Returns a sff feature descriptor by id

4.15 dc1394/vendor/pixelink.h File Reference

No docs yet.

```
#include <dc1394/log.h>
#include <dc1394/types.h>
```

Data Structures

- struct **__dc1394_pxl_gpio_info_struct**
- struct **__dc1394_pxl_camera_info_struct**
- struct **__dc1394_pxl_adv_feature_info_struct**

Typedefs

- typedef struct **__dc1394_pxl_gpio_info_struct dc1394_pxl_gpio_info_t**
- typedef struct **__dc1394_pxl_camera_info_struct dc1394_pxl_camera_info_t**
- typedef struct **__dc1394_pxl_adv_feature_info_struct dc1394_pxl_adv_feature_info_t**

Enumerations

- enum **dc1394pxl_gpio_polarity_t**
- enum **dc1394pxl_gpio_mode_t**

Functions

- dc1394error_t **dc1394_pxl_convert_float32_to_quadlet** (double, uint32_t ∗)
- dc1394error_t **dc1394_pxl_convert_uint32_to_float32** (uint32_t, double ∗)
- dc1394error_t **dc1394_pxl_get_camera_name** (**dc1394camera_t** ∗, char ∗, uint32_t)
- dc1394error_t **dc1394_pxl_get_camera_info** (**dc1394camera_t** ∗, **dc1394_pxl_camera_info_t** ∗)
- dc1394error_t **dc1394_pxl_get_camera_serial_number** (**dc1394camera_t** ∗, uint32_t ∗)
- dc1394error_t **dc1394_pxl_get_gpo_param** (**dc1394camera_t** ∗, uint32_t, uint32_t ∗, uint32_t ∗, uint32_t ∗)
- dc1394error_t **dc1394_pxl_get_gpo_param_min_max** (**dc1394camera_t** ∗, uint32_t, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗, uint32_t ∗)
- dc1394error_t **dc1394_pxl_get_gpo_config** (**dc1394camera_t** ∗, uint32_t, uint32_t ∗)
- dc1394error_t **dc1394_pxl_set_gpo_config** (**dc1394camera_t** ∗, uint32_t, uint32_t)
- dc1394error_t **dc1394_pxl_set_gpio_mode_param** (**dc1394camera_t** ∗, uint32_t, **dc1394pxl_gpio_polarity_t**, **dc1394pxl_gpio_mode_t**, double, double, double)
- dc1394error_t **dc1394_pxl_print_camera_info** (**dc1394camera_t** ∗, FILE ∗fd)

4.15.1 Detailed Description

No docs yet. More details soon

4.15.2 Typedef Documentation

4.15.2.1 typedef struct __dc1394_pxl_adv_feature_info_struct dc1394_pxl_adv_feature_info_t

Advanced feature inquiry

4.15.2.2 typedef struct __dc1394_pxl_camera_info_struct dc1394_pxl_camera_info_t

Camera information

4.15.2.3 typedef struct __dc1394_pxl_gpio_info_struct dc1394_pxl_gpio_info_t

GPIO Information structure

4.15.3 Enumeration Type Documentation

4.15.3.1 enum dc1394pxl_gpio_mode_t

No Docs

4.15.3.2 enum dc1394pxl_gpio_polarity_t

No Docs

4.15.4 Function Documentation

4.15.4.1 dc1394error_t dc1394_pxl_convert_float32_to_quadlet (double, uint32_t *)

No Docs

4.15.4.2 dc1394error_t dc1394_pxl_convert_uint32_to_float32 (uint32_t, double *)

No Docs

4.15.4.3 dc1394error_t dc1394_pxl_get_camera_info (dc1394camera_t *, dc1394_pxl_camera_info_t *)

No Docs

4.15.4.4 dc1394error_t dc1394_pxl_get_camera_name (dc1394camera_t *, char *, uint32_t)

No Docs

4.15.4.5 dc1394error_t dc1394_pxl_get_camera_serial_number (dc1394camera_t *, uint32_t *)

No Docs

4.15.4.6 dc1394error_t dc1394_pxl_get_gpo_config (dc1394camera_t *, uint32_t, uint32_t *)

No Docs

4.15.4.7 dc1394error_t dc1394_pxl_get_gpo_param (dc1394camera_t *, uint32_t, uint32_t *, uint32_t *, uint32_t *)

No Docs

4.15.4.8 dc1394error_t dc1394_pxl_get_gpo_param_min_max (dc1394camera_t *, uint32_t, uint32_t *, uint32_t *, uint32_t *, uint32_t *, uint32_t *, uint32_t *, uint32_t *, uint32_t *)

No Docs

4.15.4.9 dc1394error_t dc1394_pxl_print_camera_info (dc1394camera_t *, FILE * *fd*)

No Docs

4.15.4.10 dc1394error_t dc1394_pxl_set_gpio_mode_param (dc1394camera_t *, uint32_t, dc1394pxl_gpio_polarity_t, dc1394pxl_gpio_mode_t, double, double, double)

No Docs

4.15.4.11 dc1394error_t dc1394_pxl_set_gpo_config (dc1394camera_t *, uint32_t, uint32_t)

No Docs

4.16 dc1394/video.h File Reference

Functions related to video modes, formats, framerate and video flow.

```
#include <dc1394/log.h>
```

Data Structures

- struct **dc1394framerates_t**
- struct **__dc1394_video_frame**

Typedefs

- typedef struct **__dc1394_video_frame dc1394video_frame_t**

Enumerations

- enum **dc1394speed_t**
- enum **dc1394framerate_t**
- enum **dc1394operation_mode_t**

Functions

- dc1394error_t **dc1394_video_get_supported_modes** (**dc1394camera_t** *camera, **dc1394video_modes_t** *video_modes)
- dc1394error_t **dc1394_video_get_supported_framerates** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode, **dc1394framerates_t** *framerates)
- dc1394error_t **dc1394_video_get_framerate** (**dc1394camera_t** *camera, **dc1394framerate_t** *framerate)
- dc1394error_t **dc1394_video_set_framerate** (**dc1394camera_t** *camera, **dc1394framerate_t** framerate)
- dc1394error_t **dc1394_video_get_mode** (**dc1394camera_t** *camera, **dc1394video_mode_t** *video_mode)
- dc1394error_t **dc1394_video_set_mode** (**dc1394camera_t** *camera, **dc1394video_mode_t** video_mode)
- dc1394error_t **dc1394_video_get_operation_mode** (**dc1394camera_t** *camera, **dc1394operation_mode_t** *mode)
- dc1394error_t **dc1394_video_set_operation_mode** (**dc1394camera_t** *camera, **dc1394operation_mode_t** mode)
- dc1394error_t **dc1394_video_get_iso_speed** (**dc1394camera_t** *camera, **dc1394speed_t** *speed)
- dc1394error_t **dc1394_video_set_iso_speed** (**dc1394camera_t** *camera, **dc1394speed_t** speed)
- dc1394error_t **dc1394_video_get_iso_channel** (**dc1394camera_t** *camera, uint32_t *channel)
- dc1394error_t **dc1394_video_set_iso_channel** (**dc1394camera_t** *camera, uint32_t channel)
- dc1394error_t **dc1394_video_get_data_depth** (**dc1394camera_t** *camera, uint32_t *depth)
- dc1394error_t **dc1394_video_set_transmission** (**dc1394camera_t** *camera, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_video_get_transmission** (**dc1394camera_t** *camera, **dc1394switch_t** *pwr)
- dc1394error_t **dc1394_video_set_one_shot** (**dc1394camera_t** *camera, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_video_get_one_shot** (**dc1394camera_t** *camera, **dc1394bool_t** *is_on)

- dc1394error_t **dc1394_video_set_multi_shot** (**dc1394camera_t** ∗camera, uint32_t numFrames, **dc1394switch_t** pwr)
- dc1394error_t **dc1394_video_get_multi_shot** (**dc1394camera_t** ∗camera, **dc1394bool_t** ∗is_on, uint32_t ∗numFrames)
- dc1394error_t **dc1394_video_get_bandwidth_usage** (**dc1394camera_t** ∗camera, uint32_t ∗bandwidth)

4.16.1 Detailed Description

Functions related to video modes, formats, framerate and video flow. More details soon

4.16.2 Typedef Documentation

4.16.2.1 typedef struct __dc1394_video_frame dc1394video_frame_t

Video frame structure.

dc1394video_frame_t is the structure returned by the capture functions. It contains the captured image as well as a number of information.

In general this structure should be calloc'ed so that members such as "allocated size" are properly set to zero. Don't forget to free the "image" member before freeing the struct itself.

4.16.3 Enumeration Type Documentation

4.16.3.1 enum dc1394framerate_t

Enumeration of video framerates

This enumeration is used for non-Format_7 modes. The framerate can be lower than expected if the exposure time is longer than the requested frame period. Framerate can be controlled in a number of other ways: framerate feature, external trigger, software trigger, shutter throttling and packet size (Format_7)

4.16.3.2 enum dc1394operation_mode_t

Operation modes

Two operation modes exist: the legacy and most common 1394a, and the newer 1394B. The latter allows speeds over 400Mbps, but can also be used at other speeds.

4.16.3.3 enum dc1394speed_t

Enumeration of iso data speeds

Most (if not all) cameras are compatible with 400Mbps speed. Only older cameras (pre-1999) may still only work at sub-400 speeds. However, speeds lower than 400Mbps are still useful: they can be used for longer distances (e.g. 10m cables). Speeds over 400Mbps are only available in "B" mode (DC1394_-OPERATION_MODE_1394B).

4.16.4 Function Documentation

4.16.4.1 dc1394error_t dc1394_video_get_bandwidth_usage (dc1394camera_t * *camera*, uint32_t * *bandwidth*)

Gets the bandwidth usage of a camera.

This function returns the bandwidth that is used by the camera *IF* ISO was ON. The returned value is in bandwidth units. The 1394 bus has 4915 bandwidth units available per cycle. Each unit corresponds to the time it takes to send one quadlet at ISO speed S1600. The bandwidth usage at S400 is thus four times the number of quadlets per packet. Thanks to Krisitian Hogsberg for clarifying this.

4.16.4.2 dc1394error_t dc1394_video_get_data_depth (dc1394camera_t * *camera*, uint32_t * *depth*)

Gets the current data depth, in bits. Only meaningful for 16bpp video modes (RAW16, RGB48, MONO16,...)

4.16.4.3 dc1394error_t dc1394_video_get_framerate (dc1394camera_t * *camera*, dc1394framerate_t * *framerate*)

Gets the current framerate. This is meaningful only if the video mode is not scalable.

4.16.4.4 dc1394error_t dc1394_video_get_iso_channel (dc1394camera_t * *camera*, uint32_t * *channel*)

Gets the current ISO channel

4.16.4.5 dc1394error_t dc1394_video_get_iso_speed (dc1394camera_t * *camera*, dc1394speed_t * *speed*)

Gets the current ISO speed.

4.16.4.6 dc1394error_t dc1394_video_get_mode (dc1394camera_t * *camera*, dc1394video_mode_t * *video_mode*)

Gets the current vide mode.

4.16.4.7 dc1394error_t dc1394_video_get_multi_shot (dc1394camera_t * *camera*, dc1394bool_t * *is_on*, uint32_t * *numFrames*)

Gets the status of the multi-shot mode.

4.16.4.8 dc1394error_t dc1394_video_get_one_shot (dc1394camera_t * *camera*, dc1394bool_t * *is_on*)

Gets the status of the one-shot mode.

4.16.4.9 dc1394error_t dc1394_video_get_operation_mode (dc1394camera_t * *camera*, dc1394operation_mode_t * *mode*)

Gets the current operation mode.

4.16.4.10 dc1394error_t dc1394_video_get_supported_framerates (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*, dc1394framerates_t * *framerates*)

Gets a list of supported video framerates for a given video mode. This function only works with non-scalable formats.

4.16.4.11 dc1394error_t dc1394_video_get_supported_modes (dc1394camera_t * *camera*, dc1394video_modes_t * *video_modes*)

Gets a list of video modes supported by the camera.

4.16.4.12 dc1394error_t dc1394_video_get_transmission (dc1394camera_t * *camera*, dc1394switch_t * *pwr*)

Gets the status of the video transmission

4.16.4.13 dc1394error_t dc1394_video_set_framerate (dc1394camera_t * *camera*, dc1394framerate_t *framerate*)

Sets the current framerate. This is meaningful only if the video mode is not scalable.

4.16.4.14 dc1394error_t dc1394_video_set_iso_channel (dc1394camera_t * *camera*, uint32_t *channel*)

Sets the current ISO channel

4.16.4.15 dc1394error_t dc1394_video_set_iso_speed (dc1394camera_t * *camera*, dc1394speed_t *speed*)

Sets the current ISO speed. Speeds over 400Mbps require 1394B.

4.16.4.16 dc1394error_t dc1394_video_set_mode (dc1394camera_t * *camera*, dc1394video_mode_t *video_mode*)

Sets the current vide mode.

4.16.4.17 dc1394error_t dc1394_video_set_multi_shot (dc1394camera_t * *camera*, uint32_t *numFrames*, dc1394switch_t *pwr*)

Turns multishot mode on or off

4.16.4.18 dc1394error_t dc1394_video_set_one_shot (dc1394camera_t * *camera*, dc1394switch_t *pwr*)

Turns one-shot mode on or off

4.16.4.19 dc1394error_t dc1394_video_set_operation_mode (dc1394camera_t * *camera*, dc1394operation_mode_t *mode*)

Sets the current operation mode.

4.16.4.20 dc1394error_t dc1394_video_set_transmission (dc1394camera_t * *camera*, dc1394switch_t *pwr*)

Starts/stops the isochronous data transmission. In other words, use this to control the image flow.

Index