

libdc1394

Generated by Doxygen 1.9.3



<b>1 Data Structure Index</b>	<b>1</b>
1.1 Data Structures . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Data Structure Documentation</b>	<b>5</b>
3.1 __dc1394_camera Struct Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.2 __dc1394_pxl_adv_feature_info_struct Struct Reference . . . . .	5
3.2.1 Detailed Description . . . . .	5
3.3 __dc1394_pxl_camera_info_struct Struct Reference . . . . .	5
3.3.1 Detailed Description . . . . .	6
3.4 __dc1394_pxl_gpio_info_struct Struct Reference . . . . .	6
3.4.1 Detailed Description . . . . .	6
3.5 __dc1394_video_frame Struct Reference . . . . .	6
3.5.1 Detailed Description . . . . .	6
3.6 __dc1394camera_list_t Struct Reference . . . . .	6
3.6.1 Detailed Description . . . . .	7
3.7 __dc1394feature_info_t_struct Struct Reference . . . . .	7
3.7.1 Detailed Description . . . . .	7
3.8 __dc1394featureset_t Struct Reference . . . . .	7
3.8.1 Detailed Description . . . . .	7
3.9 __dc1394format7mode_t Struct Reference . . . . .	7
3.9.1 Detailed Description . . . . .	8
3.10 __dc1394format7modeset_t Struct Reference . . . . .	8
3.10.1 Detailed Description . . . . .	8
3.11 dc1394basler_dcam_csr_value_t Struct Reference . . . . .	8
3.11.1 Detailed Description . . . . .	8
3.12 dc1394basler_dcam_whitebalance_csr_value_t Struct Reference . . . . .	8
3.12.1 Detailed Description . . . . .	8
3.13 dc1394basler_sff_chunk_tail_t Struct Reference . . . . .	9
3.13.1 Detailed Description . . . . .	9
3.14 dc1394basler_sff_crc_checksum_t Struct Reference . . . . .	9
3.14.1 Detailed Description . . . . .	9
3.15 dc1394basler_sff_cycle_time_stamp_t Struct Reference . . . . .	9
3.15.1 Detailed Description . . . . .	9
3.16 dc1394basler_sff_dcam_values_t Struct Reference . . . . .	9
3.16.1 Detailed Description . . . . .	10
3.17 dc1394basler_sff_extended_data_stream_t Struct Reference . . . . .	10
3.17.1 Detailed Description . . . . .	10
3.18 dc1394basler_sff_frame_counter_t Struct Reference . . . . .	10
3.18.1 Detailed Description . . . . .	10

3.19 dc1394basler_sff_guid_t Struct Reference . . . . .	10
3.19.1 Detailed Description . . . . .	11
3.20 dc1394basler_sff_t Struct Reference . . . . .	11
3.20.1 Detailed Description . . . . .	11
3.21 dc1394camera_id_t Struct Reference . . . . .	11
3.21.1 Detailed Description . . . . .	11
3.22 dc1394color_codings_t Struct Reference . . . . .	11
3.22.1 Detailed Description . . . . .	12
3.23 dc1394feature_modes_t Struct Reference . . . . .	12
3.23.1 Detailed Description . . . . .	12
3.24 dc1394framerates_t Struct Reference . . . . .	12
3.24.1 Detailed Description . . . . .	12
3.25 dc1394trigger_modes_t Struct Reference . . . . .	12
3.25.1 Detailed Description . . . . .	12
3.26 dc1394trigger_sources_t Struct Reference . . . . .	13
3.26.1 Detailed Description . . . . .	13
3.27 dc1394video_modes_t Struct Reference . . . . .	13
3.27.1 Detailed Description . . . . .	13
3.28 fw_cdev_add_descriptor Struct Reference . . . . .	13
3.28.1 Detailed Description . . . . .	13
3.29 fw_cdev_allocate Struct Reference . . . . .	14
3.29.1 Detailed Description . . . . .	14
3.30 fw_cdev_allocate_iso_resource Struct Reference . . . . .	14
3.30.1 Detailed Description . . . . .	14
3.31 fw_cdev_create_iso_context Struct Reference . . . . .	15
3.31.1 Detailed Description . . . . .	15
3.32 fw_cdev_deallocate Struct Reference . . . . .	15
3.32.1 Detailed Description . . . . .	16
3.33 fw_cdev_event Union Reference . . . . .	16
3.33.1 Detailed Description . . . . .	16
3.34 fw_cdev_event_bus_reset Struct Reference . . . . .	16
3.34.1 Detailed Description . . . . .	16
3.35 fw_cdev_event_common Struct Reference . . . . .	17
3.35.1 Detailed Description . . . . .	17
3.36 fw_cdev_event_iso_interrupt Struct Reference . . . . .	17
3.36.1 Detailed Description . . . . .	17
3.37 fw_cdev_event_iso_resource Struct Reference . . . . .	18
3.37.1 Detailed Description . . . . .	18
3.38 fw_cdev_event_request Struct Reference . . . . .	18
3.38.1 Detailed Description . . . . .	18
3.39 fw_cdev_event_response Struct Reference . . . . .	19
3.39.1 Detailed Description . . . . .	19

3.40 fw_cdev_get_cycle_timer Struct Reference . . . . .	19
3.40.1 Detailed Description . . . . .	19
3.41 fw_cdev_get_cycle_timer2 Struct Reference . . . . .	19
3.41.1 Detailed Description . . . . .	20
3.42 fw_cdev_get_info Struct Reference . . . . .	20
3.42.1 Detailed Description . . . . .	20
3.43 fw_cdev_initiate_bus_reset Struct Reference . . . . .	20
3.43.1 Detailed Description . . . . .	20
3.44 fw_cdev_iso_packet Struct Reference . . . . .	21
3.44.1 Detailed Description . . . . .	21
3.45 fw_cdev_queue_iso Struct Reference . . . . .	21
3.45.1 Detailed Description . . . . .	21
3.46 fw_cdev_remove_descriptor Struct Reference . . . . .	22
3.46.1 Detailed Description . . . . .	22
3.47 fw_cdev_send_request Struct Reference . . . . .	22
3.47.1 Detailed Description . . . . .	22
3.48 fw_cdev_send_response Struct Reference . . . . .	22
3.48.1 Detailed Description . . . . .	23
3.49 fw_cdev_send_stream_packet Struct Reference . . . . .	23
3.49.1 Detailed Description . . . . .	23
3.50 fw_cdev_start_iso Struct Reference . . . . .	23
3.50.1 Detailed Description . . . . .	23
3.51 fw_cdev_stop_iso Struct Reference . . . . .	24
3.51.1 Detailed Description . . . . .	24
<b>4 File Documentation</b>	<b>25</b>
4.1 dc1394/camera.h File Reference . . . . .	25
4.1.1 Detailed Description . . . . .	26
4.1.2 Typedef Documentation . . . . .	26
4.1.2.1 dc1394camera_list_t . . . . .	26
4.1.2.2 dc1394camera_t . . . . .	26
4.1.3 Enumeration Type Documentation . . . . .	26
4.1.3.1 dc1394iidx_version_t . . . . .	26
4.1.3.2 dc1394phy_delay_t . . . . .	27
4.1.3.3 dc1394power_class_t . . . . .	27
4.1.4 Function Documentation . . . . .	27
4.1.4.1 dc1394_camera_enumerate() . . . . .	27
4.1.4.2 dc1394_camera_free() . . . . .	27
4.1.4.3 dc1394_camera_free_list() . . . . .	27
4.1.4.4 dc1394_camera_get_node() . . . . .	27
4.1.4.5 dc1394_camera_get_platform_string() . . . . .	28
4.1.4.6 dc1394_camera_new() . . . . .	28

4.1.4.7 dc1394_camera_new_unit()	28
4.1.4.8 dc1394_camera_print_info()	28
4.1.4.9 dc1394_camera_set_broadcast()	28
4.1.4.10 dc1394_free()	28
4.1.4.11 dc1394_new()	29
4.1.4.12 dc1394_reset_bus()	29
4.2 dc1394/conversions.h File Reference	29
4.2.1 Detailed Description	30
4.2.2 Enumeration Type Documentation	30
4.2.2.1 dc1394bayer_method_t	30
4.2.2.2 dc1394stereo_method_t	30
4.2.3 Function Documentation	30
4.2.3.1 dc1394_bayer_decoding_16bit()	30
4.2.3.2 dc1394_bayer_decoding_8bit()	31
4.2.3.3 dc1394_convert_frames()	31
4.2.3.4 dc1394_convert_to_MONO8()	31
4.2.3.5 dc1394_convert_to_RGB8()	31
4.2.3.6 dc1394_convert_to_YUV422()	32
4.2.3.7 dc1394_debayer_frames()	32
4.2.3.8 dc1394_deinterlace_stereo()	32
4.2.3.9 dc1394_deinterlace_stereo_frames()	32
4.3 dc1394/dc1394.h File Reference	33
4.3.1 Detailed Description	33
4.4 dc1394/format7.h File Reference	33
4.4.1 Detailed Description	34
4.4.2 Typedef Documentation	35
4.4.2.1 dc1394format7mode_t	35
4.4.2.2 dc1394format7modeset_t	35
4.4.3 Function Documentation	35
4.4.3.1 dc1394_format7_get_color_coding()	35
4.4.3.2 dc1394_format7_get_color_codings()	35
4.4.3.3 dc1394_format7_get_color_filter()	35
4.4.3.4 dc1394_format7_get_data_depth()	36
4.4.3.5 dc1394_format7_get_frame_interval()	36
4.4.3.6 dc1394_format7_get_image_position()	36
4.4.3.7 dc1394_format7_get_image_size()	36
4.4.3.8 dc1394_format7_get_max_image_size()	36
4.4.3.9 dc1394_format7_get_mode_info()	37
4.4.3.10 dc1394_format7_get_modeset()	37
4.4.3.11 dc1394_format7_get_packet_parameters()	37
4.4.3.12 dc1394_format7_get_packet_size()	37
4.4.3.13 dc1394_format7_get_packets_per_frame()	37

4.4.3.14 dc1394_format7_get_pixel_number()	38
4.4.3.15 dc1394_format7_get_recommended_packet_size()	38
4.4.3.16 dc1394_format7_get_roi()	38
4.4.3.17 dc1394_format7_get_total_bytes()	38
4.4.3.18 dc1394_format7_get_unit_position()	38
4.4.3.19 dc1394_format7_get_unit_size()	39
4.4.3.20 dc1394_format7_set_color_coding()	39
4.4.3.21 dc1394_format7_set_image_position()	39
4.4.3.22 dc1394_format7_set_image_size()	39
4.4.3.23 dc1394_format7_set_packet_size()	39
4.4.3.24 dc1394_format7_set_roi()	40
4.5 dc1394/iso.h File Reference	40
4.5.1 Detailed Description	40
4.5.2 Function Documentation	40
4.5.2.1 dc1394_iso_allocate_bandwidth()	40
4.5.2.2 dc1394_iso_allocate_channel()	41
4.5.2.3 dc1394_iso_release_all()	42
4.5.2.4 dc1394_iso_release_bandwidth()	42
4.5.2.5 dc1394_iso_release_channel()	43
4.5.2.6 dc1394_iso_set_persist()	43
4.6 dc1394/capture.h File Reference	43
4.6.1 Detailed Description	44
4.6.2 Macro Definition Documentation	44
4.6.2.1 DC1394_CAPTURE_FLAGS_CHANNEL_ALLOC	44
4.6.3 Typedef Documentation	45
4.6.3.1 dc1394capture_callback_t	45
4.6.4 Enumeration Type Documentation	45
4.6.4.1 dc1394capture_policy_t	45
4.6.5 Function Documentation	45
4.6.5.1 dc1394_capture_dequeue()	45
4.6.5.2 dc1394_capture_enqueue()	45
4.6.5.3 dc1394_capture_get_fileno()	46
4.6.5.4 dc1394_capture_is_frame_corrupt()	46
4.6.5.5 dc1394_capture_set_callback()	46
4.6.5.6 dc1394_capture_setup()	46
4.6.5.7 dc1394_capture_stop()	46
4.7 dc1394/control.h File Reference	46
4.7.1 Detailed Description	48
4.7.2 Typedef Documentation	49
4.7.2.1 dc1394feature_info_t	49
4.7.2.2 dc1394featureset_t	49
4.7.3 Enumeration Type Documentation	49

4.7.3.1 dc1394feature_mode_t . . . . .	49
4.7.3.2 dc1394feature_t . . . . .	49
4.7.3.3 dc1394trigger_mode_t . . . . .	50
4.7.3.4 dc1394trigger_polarity_t . . . . .	50
4.7.3.5 dc1394trigger_source_t . . . . .	50
4.7.4 Function Documentation . . . . .	50
4.7.4.1 dc1394_camera_reset() . . . . .	50
4.7.4.2 dc1394_camera_set_power() . . . . .	50
4.7.4.3 dc1394_external_trigger_get_mode() . . . . .	50
4.7.4.4 dc1394_external_trigger_get_polarity() . . . . .	51
4.7.4.5 dc1394_external_trigger_get_power() . . . . .	51
4.7.4.6 dc1394_external_trigger_get_source() . . . . .	51
4.7.4.7 dc1394_external_trigger_get_supported_sources() . . . . .	51
4.7.4.8 dc1394_external_trigger_has_polarity() . . . . .	51
4.7.4.9 dc1394_external_trigger_set_mode() . . . . .	51
4.7.4.10 dc1394_external_trigger_set_polarity() . . . . .	52
4.7.4.11 dc1394_external_trigger_set_power() . . . . .	52
4.7.4.12 dc1394_external_trigger_set_source() . . . . .	52
4.7.4.13 dc1394_feature_get() . . . . .	52
4.7.4.14 dc1394_feature_get_absolute_boundaries() . . . . .	52
4.7.4.15 dc1394_feature_get_absolute_control() . . . . .	52
4.7.4.16 dc1394_feature_get_absolute_value() . . . . .	53
4.7.4.17 dc1394_feature_get_all() . . . . .	53
4.7.4.18 dc1394_feature_get_boundaries() . . . . .	53
4.7.4.19 dc1394_feature_get_mode() . . . . .	53
4.7.4.20 dc1394_feature_get_modes() . . . . .	53
4.7.4.21 dc1394_feature_get_power() . . . . .	53
4.7.4.22 dc1394_feature_get_value() . . . . .	54
4.7.4.23 dc1394_feature_has_absolute_control() . . . . .	54
4.7.4.24 dc1394_feature_is_present() . . . . .	54
4.7.4.25 dc1394_feature_is_readable() . . . . .	54
4.7.4.26 dc1394_feature_is_switchable() . . . . .	54
4.7.4.27 dc1394_feature_print() . . . . .	54
4.7.4.28 dc1394_feature_print_all() . . . . .	55
4.7.4.29 dc1394_feature_set_absolute_control() . . . . .	55
4.7.4.30 dc1394_feature_set_absolute_value() . . . . .	55
4.7.4.31 dc1394_feature_set_mode() . . . . .	55
4.7.4.32 dc1394_feature_set_power() . . . . .	55
4.7.4.33 dc1394_feature_set_value() . . . . .	55
4.7.4.34 dc1394_feature_temperature_get_value() . . . . .	56
4.7.4.35 dc1394_feature_temperature_set_value() . . . . .	56
4.7.4.36 dc1394_feature_whitebalance_get_value() . . . . .	56



4.7.4.37 dc1394_feature_whitebalance_set_value()	56
4.7.4.38 dc1394_feature_whiteshading_get_value()	56
4.7.4.39 dc1394_feature_whiteshading_set_value()	57
4.7.4.40 dc1394_memory_busy()	57
4.7.4.41 dc1394_memory_load()	57
4.7.4.42 dc1394_memory_save()	57
4.7.4.43 dc1394_pio_get()	57
4.7.4.44 dc1394_pio_set()	57
4.7.4.45 dc1394_software_trigger_get_power()	58
4.7.4.46 dc1394_software_trigger_set_power()	58
4.8 dc1394/log.h File Reference	58
4.8.1 Detailed Description	58
4.8.2 Enumeration Type Documentation	59
4.8.2.1 dc1394error_t	59
4.8.2.2 dc1394log_t	59
4.8.3 Function Documentation	59
4.8.3.1 dc1394_log_debug()	59
4.8.3.2 dc1394_log_error()	60
4.8.3.3 dc1394_log_register_handler()	60
4.8.3.4 dc1394_log_set_default_handler()	60
4.8.3.5 dc1394_log_warning()	60
4.9 dc1394/register.h File Reference	61
4.9.1 Detailed Description	61
4.9.2 Function Documentation	62
4.9.2.1 dc1394_get_absolute_register()	62
4.9.2.2 dc1394_get_adv_control_registers()	62
4.9.2.3 dc1394_get_control_registers()	62
4.9.2.4 dc1394_get_format7_register()	62
4.9.2.5 dc1394_get_PIO_register()	62
4.9.2.6 dc1394_get_registers()	63
4.9.2.7 dc1394_get_SIO_register()	63
4.9.2.8 dc1394_get_strobe_register()	63
4.9.2.9 dc1394_set_absolute_register()	63
4.9.2.10 dc1394_set_adv_control_registers()	63
4.9.2.11 dc1394_set_control_registers()	64
4.9.2.12 dc1394_set_format7_register()	64
4.9.2.13 dc1394_set_PIO_register()	64
4.9.2.14 dc1394_set_registers()	64
4.9.2.15 dc1394_set_SIO_register()	64
4.9.2.16 dc1394_set_strobe_register()	65
4.10 dc1394/types.h File Reference	65
4.10.1 Detailed Description	65

4.10.2 Enumeration Type Documentation . . . . .	65
4.10.2.1 dc1394bool_t . . . . .	65
4.10.2.2 dc1394byte_order_t . . . . .	66
4.10.2.3 dc1394color_coding_t . . . . .	66
4.10.2.4 dc1394color_filter_t . . . . .	66
4.10.2.5 dc1394switch_t . . . . .	66
4.10.2.6 dc1394video_mode_t . . . . .	66
4.11 dc1394/utils.h File Reference . . . . .	66
4.11.1 Detailed Description . . . . .	67
4.11.2 Function Documentation . . . . .	67
4.11.2.1 dc1394_checksum_crc16() . . . . .	67
4.11.2.2 dc1394_error_get_string() . . . . .	67
4.11.2.3 dc1394_feature_get_string() . . . . .	68
4.11.2.4 dc1394_framerate_as_float() . . . . .	68
4.11.2.5 dc1394_get_color_coding_bit_size() . . . . .	68
4.11.2.6 dc1394_get_color_coding_data_depth() . . . . .	68
4.11.2.7 dc1394_get_color_coding_from_video_mode() . . . . .	68
4.11.2.8 dc1394_get_image_size_from_video_mode() . . . . .	68
4.11.2.9 dc1394_is_color() . . . . .	69
4.11.2.10 dc1394_is_same_camera() . . . . .	69
4.11.2.11 dc1394_is_video_mode_scalable() . . . . .	69
4.11.2.12 dc1394_is_video_mode_still_image() . . . . .	69
4.12 dc1394/vendor/avt.h File Reference . . . . .	69
4.12.1 Detailed Description . . . . .	71
4.12.2 Function Documentation . . . . .	71
4.12.2.1 dc1394_avt_get_advanced_feature_inquiry() . . . . .	71
4.12.2.2 dc1394_avt_get_aoi() . . . . .	72
4.12.2.3 dc1394_avt_get_auto_gain() . . . . .	72
4.12.2.4 dc1394_avt_get_auto_shutter() . . . . .	72
4.12.2.5 dc1394_avt_get_blemish() . . . . .	72
4.12.2.6 dc1394_avt_get_channel_adjust() . . . . .	72
4.12.2.7 dc1394_avt_get_color_corr() . . . . .	73
4.12.2.8 dc1394_avt_get_deferred_trans() . . . . .	73
4.12.2.9 dc1394_avt_get_dsnu() . . . . .	73
4.12.2.10 dc1394_avt_get_extented_shutter() . . . . .	73
4.12.2.11 dc1394_avt_get_frame_info() . . . . .	73
4.12.2.12 dc1394_avt_get_gpdata_info() . . . . .	74
4.12.2.13 dc1394_avt_get_hsnr() . . . . .	74
4.12.2.14 dc1394_avt_get_io() . . . . .	74
4.12.2.15 dc1394_avt_get_lut() . . . . .	74
4.12.2.16 dc1394_avt_get_lut_info() . . . . .	74
4.12.2.17 dc1394_avt_get_lut_mem_ctrl() . . . . .	75

4.12.2.18 dc1394_avt_get_MaxResolution()	75
4.12.2.19 dc1394_avt_get_mirror()	75
4.12.2.20 dc1394_avt_get_multiple_slope()	75
4.12.2.21 dc1394_avt_get_shading()	75
4.12.2.22 dc1394_avt_get_shading_info()	76
4.12.2.23 dc1394_avt_get_shading_mem_ctrl()	76
4.12.2.24 dc1394_avt_get_test_images()	76
4.12.2.25 dc1394_avt_get_timebase()	76
4.12.2.26 dc1394_avt_get_trigger_delay()	76
4.12.2.27 dc1394_avt_get_version()	76
4.12.2.28 dc1394_avt_print_advanced_feature()	77
4.12.2.29 dc1394_avt_read_gpdata()	77
4.12.2.30 dc1394_avt_read_shading_img()	77
4.12.2.31 dc1394_avt_reset()	77
4.12.2.32 dc1394_avt_reset_frame_info()	77
4.12.2.33 dc1394_avt_set_aoi()	77
4.12.2.34 dc1394_avt_set_auto_gain()	78
4.12.2.35 dc1394_avt_set_auto_shutter()	78
4.12.2.36 dc1394_avt_set_blemish()	78
4.12.2.37 dc1394_avt_set_channel_adjust()	78
4.12.2.38 dc1394_avt_set_color_corr()	78
4.12.2.39 dc1394_avt_set_deferred_trans()	79
4.12.2.40 dc1394_avt_set_dsnu()	79
4.12.2.41 dc1394_avt_set_extented_shutter()	79
4.12.2.42 dc1394_avt_set_hsnr()	79
4.12.2.43 dc1394_avt_set_io()	79
4.12.2.44 dc1394_avt_set_lut()	80
4.12.2.45 dc1394_avt_set_lut_mem_ctrl()	80
4.12.2.46 dc1394_avt_set_mirror()	80
4.12.2.47 dc1394_avt_set_multiple_slope()	80
4.12.2.48 dc1394_avt_set_shading()	80
4.12.2.49 dc1394_avt_set_shading_mem_ctrl()	81
4.12.2.50 dc1394_avt_set_test_images()	81
4.12.2.51 dc1394_avt_set_timebase()	81
4.12.2.52 dc1394_avt_set_trigger_delay()	81
4.12.2.53 dc1394_avt_write_gpdata()	81
4.12.2.54 dc1394_avt_write_shading_img()	81
4.13 dc1394/vendor/basler.h File Reference	82
4.13.1 Detailed Description	82
4.13.2 Function Documentation	82
4.13.2.1 dc1394_basler_sff_check_crc()	82
4.13.2.2 dc1394_basler_sff_chunk_find()	83

4.13.2.3 dc1394_basler_sff_chunk_iterate()	83
4.13.2.4 dc1394_basler_sff_chunk_iterate_init()	83
4.13.2.5 dc1394_basler_sff_feature_enable()	83
4.13.2.6 dc1394_basler_sff_feature_is_available()	83
4.13.2.7 dc1394_basler_sff_feature_is_enabled()	84
4.13.2.8 dc1394_basler_sff_feature_print()	84
4.13.2.9 dc1394_basler_sff_feature_print_all()	84
4.13.2.10 dc1394_basler_sff_is_available()	84
4.14 dc1394/vendor/basler_sff.h File Reference	84
4.14.1 Detailed Description	85
4.14.2 Typedef Documentation	85
4.14.2.1 dc1394basler_dcam_csr_value_t	85
4.14.2.2 dc1394basler_dcam_whitebalance_csr_value_t	85
4.14.2.3 dc1394basler_sff_crc_checksum_t	85
4.14.2.4 dc1394basler_sff_cycle_time_stamp_t	86
4.14.2.5 dc1394basler_sff_dcam_values_t	86
4.14.2.6 dc1394basler_sff_extended_data_stream_t	86
4.14.2.7 dc1394basler_sff_feature_t	86
4.14.2.8 dc1394basler_sff_frame_counter_t	86
4.14.3 Enumeration Type Documentation	86
4.14.3.1 dc1394basler_sff_feature_t	86
4.15 dc1394/vendor/basler_sff_registry.h File Reference	88
4.15.1 Detailed Description	88
4.15.2 Function Documentation	88
4.15.2.1 basler_sff_registry_find_by_chunk_guid()	88
4.15.2.2 basler_sff_registry_find_by_csr_guid()	88
4.15.2.3 basler_sff_registry_find_by_id()	89
4.16 dc1394/vendor/pixelink.h File Reference	89
4.16.1 Detailed Description	90
4.16.2 Typedef Documentation	90
4.16.2.1 dc1394_pxl_adv_feature_info_t	90
4.16.2.2 dc1394_pxl_camera_info_t	90
4.16.2.3 dc1394_pxl_gpio_info_t	90
4.16.3 Enumeration Type Documentation	90
4.16.3.1 dc1394pxl_gpio_mode_t	90
4.16.3.2 dc1394pxl_gpio_polarity_t	90
4.16.4 Function Documentation	91
4.16.4.1 dc1394_pxl_convert_float32_to_quadlet()	91
4.16.4.2 dc1394_pxl_convert_uint32_to_float32()	91
4.16.4.3 dc1394_pxl_get_camera_info()	91
4.16.4.4 dc1394_pxl_get_camera_name()	91
4.16.4.5 dc1394_pxl_get_camera_serial_number()	91

4.16.4.6 dc1394_pxl_get_gpo_config()	92
4.16.4.7 dc1394_pxl_get_gpo_param()	92
4.16.4.8 dc1394_pxl_get_gpo_param_min_max()	92
4.16.4.9 dc1394_pxl_print_camera_info()	92
4.16.4.10 dc1394_pxl_set_gpio_mode_param()	92
4.16.4.11 dc1394_pxl_set_gpo_config()	93
4.17 dc1394/video.h File Reference	93
4.17.1 Detailed Description	94
4.17.2 Typedef Documentation	94
4.17.2.1 dc1394video_frame_t	94
4.17.3 Enumeration Type Documentation	94
4.17.3.1 dc1394framerate_t	94
4.17.3.2 dc1394operation_mode_t	95
4.17.3.3 dc1394speed_t	95
4.17.4 Function Documentation	95
4.17.4.1 dc1394_video_get_bandwidth_usage()	95
4.17.4.2 dc1394_video_get_data_depth()	95
4.17.4.3 dc1394_video_get_framerate()	95
4.17.4.4 dc1394_video_get_iso_channel()	96
4.17.4.5 dc1394_video_get_iso_speed()	96
4.17.4.6 dc1394_video_get_mode()	96
4.17.4.7 dc1394_video_get_multi_shot()	96
4.17.4.8 dc1394_video_get_one_shot()	96
4.17.4.9 dc1394_video_get_operation_mode()	96
4.17.4.10 dc1394_video_get_supported_framerates()	97
4.17.4.11 dc1394_video_get_supported_modes()	97
4.17.4.12 dc1394_video_get_transmission()	97
4.17.4.13 dc1394_video_set_framerate()	97
4.17.4.14 dc1394_video_set_iso_channel()	97
4.17.4.15 dc1394_video_set_iso_speed()	97
4.17.4.16 dc1394_video_set_mode()	98
4.17.4.17 dc1394_video_set_multi_shot()	98
4.17.4.18 dc1394_video_set_one_shot()	98
4.17.4.19 dc1394_video_set_operation_mode()	98
4.17.4.20 dc1394_video_set_transmission()	98



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

__dc1394_camera	5
__dc1394_pxl_adv_feature_info_struct	5
__dc1394_pxl_camera_info_struct	5
__dc1394_pxl_gpio_info_struct	6
__dc1394_video_frame	6
__dc1394camera_list_t	6
__dc1394feature_info_t_struct	7
__dc1394featureset_t	7
__dc1394format7mode_t	7
__dc1394format7modeset_t	8
dc1394basler_dcam_csr_value_t	8
dc1394basler_dcam_whitebalance_csr_value_t	8
dc1394basler_sff_chunk_tail_t	9
dc1394basler_sff_crc_checksum_t	9
dc1394basler_sff_cycle_time_stamp_t	9
dc1394basler_sff_dcam_values_t	9
dc1394basler_sff_extended_data_stream_t	10
dc1394basler_sff_frame_counter_t	10
dc1394basler_sff_guid_t	10
dc1394basler_sff_t	11
dc1394camera_id_t	11
dc1394color_codings_t	11
dc1394feature_modes_t	12
dc1394framerates_t	12
dc1394trigger_modes_t	12
dc1394trigger_sources_t	13
dc1394video_modes_t	13
fw_cdev_add_descriptor	13
fw_cdev_allocate	14
fw_cdev_allocate_iso_resource	14
fw_cdev_create_iso_context	15
fw_cdev_deallocate	15
fw_cdev_event	16
fw_cdev_event_bus_reset	16
fw_cdev_event_common	17

<code>fw_cdev_event_iso_interrupt</code> . . . . .	17
<code>fw_cdev_event_iso_resource</code> . . . . .	18
<code>fw_cdev_event_request</code> . . . . .	18
<code>fw_cdev_event_response</code> . . . . .	19
<code>fw_cdev_get_cycle_timer</code> . . . . .	19
<code>fw_cdev_get_cycle_timer2</code> . . . . .	19
<code>fw_cdev_get_info</code> . . . . .	20
<code>fw_cdev_initiate_bus_reset</code> . . . . .	20
<code>fw_cdev_iso_packet</code> . . . . .	21
<code>fw_cdev_queue_iso</code> . . . . .	21
<code>fw_cdev_remove_descriptor</code> . . . . .	22
<code>fw_cdev_send_request</code> . . . . .	22
<code>fw_cdev_send_response</code> . . . . .	22
<code>fw_cdev_send_stream_packet</code> . . . . .	23
<code>fw_cdev_start_iso</code> . . . . .	23
<code>fw_cdev_stop_iso</code> . . . . .	24



## Chapter 2

# File Index

### 2.1 File List

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

dc1394/ <b>camera.h</b>	
Basic system and camera functions . . . . .	25
dc1394/ <b>capture.h</b>	
Capture functions . . . . .	43
dc1394/ <b>control.h</b>	
Diverse controls of camera features . . . . .	46
dc1394/ <b>conversions.h</b>	
Functions to convert video formats . . . . .	29
dc1394/ <b>dc1394.h</b>	
Main include file, which include all others . . . . .	33
dc1394/ <b>format7.h</b>	
Functions to control Format_7 (aka scalable format, ROI) . . . . .	33
dc1394/ <b>iso.h</b>	
Functions to manually manage the ISO resources (channels and bandwidth) . . . . .	40
dc1394/ <b>log.h</b>	
Functions to log errors, warning and debug messages . . . . .	58
dc1394/ <b>register.h</b>	
Functions to directly access camera registers . . . . .	61
dc1394/ <b>types.h</b>	
Various types that must be defined here . . . . .	65
dc1394/ <b>utils.h</b>	
Utility functions . . . . .	66
dc1394/ <b>video.h</b>	
Functions related to video modes, formats, framerate and video flow . . . . .	93
dc1394/vendor/ <b>avt.h</b>	
No docs yet . . . . .	69
dc1394/vendor/ <b>basler.h</b>	
No docs yet . . . . .	82
dc1394/vendor/ <b>basler_sff.h</b>	
No docs yet . . . . .	84
dc1394/vendor/ <b>basler_sff_registry.h</b>	
No docs yet . . . . .	88
dc1394/vendor/ <b>pixelink.h</b>	
No docs yet . . . . .	89



## Chapter 3

# Data Structure Documentation

### 3.1 `__dc1394_camera` Struct Reference

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

#### 3.1.1 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**

### 3.2 `__dc1394_pxl_adv_feature_info_struct` Struct Reference

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

#### 3.2.1 Detailed Description

Advanced feature inquiry

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

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

### 3.3 `__dc1394_pxl_camera_info_struct` Struct Reference

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

### 3.3.1 Detailed Description

Camera information

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

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

## 3.4 \_\_dc1394\_pxl\_gpio\_info\_struct Struct Reference

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

### 3.4.1 Detailed Description

GPIO Information structure

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

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

## 3.5 \_\_dc1394\_video\_frame Struct Reference

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

### 3.5.1 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**

## 3.6 \_\_dc1394camera\_list\_t Struct Reference

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

### 3.6.1 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`

## 3.7 \_\_dc1394feature\_info\_t\_struct Struct Reference

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

### 3.7.1 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`

## 3.8 \_\_dc1394featureset\_t Struct Reference

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

### 3.8.1 Detailed Description

The list of features

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

- `dc1394/control.h`

## 3.9 \_\_dc1394format7mode\_t Struct Reference

```
#include <dc1394/format7.h>
```

### 3.9.1 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**

## 3.10 `__dc1394format7modeset_t` Struct Reference

```
#include <dc1394/format7.h>
```

### 3.10.1 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**

## 3.11 `dc1394basler_dcam_csr_value_t` Struct Reference

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

### 3.11.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.12 `dc1394basler_dcam_whitebalance_csr_value_t` Struct Reference

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

### 3.12.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.13 dc1394basler\_sff\_chunk\_tail\_t Struct Reference

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

### 3.13.1 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**

## 3.14 dc1394basler\_sff\_crc\_checksum\_t Struct Reference

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

### 3.14.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.15 dc1394basler\_sff\_cycle\_time\_stamp\_t Struct Reference

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

### 3.15.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.16 dc1394basler\_sff\_dcam\_values\_t Struct Reference

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

### 3.16.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.17 dc1394basler\_sff\_extended\_data\_stream\_t Struct Reference

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

### 3.17.1 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**

## 3.18 dc1394basler\_sff\_frame\_counter\_t Struct Reference

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

### 3.18.1 Detailed Description

No Docs

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

- dc1394/vendor/ **basler\_sff.h**

## 3.19 dc1394basler\_sff\_guid\_t Struct Reference

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



### 3.19.1 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**

## 3.20 dc1394basler\_sff\_t Struct Reference

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

### 3.20.1 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**

## 3.21 dc1394camera\_id\_t Struct Reference

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

### 3.21.1 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**

## 3.22 dc1394color\_codings\_t Struct Reference

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

### 3.22.1 Detailed Description

A struct containing a list of color codings

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

- dc1394/ **types.h**

## 3.23 dc1394feature\_modes\_t Struct Reference

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

### 3.23.1 Detailed Description

List of feature modes

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

- dc1394/ **control.h**

## 3.24 dc1394framerates\_t Struct Reference

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

### 3.24.1 Detailed Description

List of framerates

**dc1394framerates\_t** (p. 12) 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**

## 3.25 dc1394trigger\_modes\_t Struct Reference

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

### 3.25.1 Detailed Description

List of trigger modes

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

- dc1394/ **control.h**

## 3.26 dc1394trigger\_sources\_t Struct Reference

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

### 3.26.1 Detailed Description

List of trigger sources

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

- dc1394/ **control.h**

## 3.27 dc1394video\_modes\_t Struct Reference

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

### 3.27.1 Detailed Description

A struct containing a list of video modes

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

- dc1394/ **types.h**

## 3.28 fw\_cdev\_add\_descriptor Struct Reference

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

### 3.28.1 Detailed Description

struct **fw\_cdev\_add\_descriptor** (p. 13) - Add contents to the local node's config ROM @immediate: If non-zero, immediate key to insert before pointer @key: Upper 8 bits of root directory pointer @data: Userspace pointer to contents of descriptor block @length: Length of descriptor block data, in quadlets @handle: 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 @key field specifies the upper 8 bits of the descriptor root directory pointer and the @data and @length fields specify the contents. The @key 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 @immediate field specifies an immediate key which will be inserted before the root directory pointer.

@immediate, @key, and @data array elements are CPU-endian quadlets.

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.

This ioctl affects the configuration ROMs of all local nodes. The ioctl only succeeds on device files which represent a local node.

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

- dc1394/juju/firewire-cdev.h

## 3.29 fw\_cdev\_allocate Struct Reference

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

### 3.29.1 Detailed Description

struct **fw\_cdev\_allocate** (p. 14) - Allocate a CSR address range @offset: Start offset of the address range @closure: To be passed back to userspace in request events @length: Length of the address range, in bytes @handle: 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. 18) event will be written back. The @closure field is passed back to userspace in the response event. The @handle field is an out parameter, returning a handle to the allocated range to be used for later deallocation of the range.

The address range is allocated on all local nodes. The address allocation is exclusive except for the FCP command and response registers.

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

- dc1394/juju/firewire-cdev.h

## 3.30 fw\_cdev\_allocate\_iso\_resource Struct Reference

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

### 3.30.1 Detailed Description

struct **fw\_cdev\_allocate\_iso\_resource** (p. 14) - (De)allocate a channel or bandwidth @closure: Passed back to userspace in corresponding iso resource events @channels: Isochronous channels of which one is to be (de)allocated @bandwidth: Isochronous bandwidth units to be (de)allocated @handle: 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 @channels 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. An FW\_CDEV\_EVENT\_ISO\_RESOURCE\_DEALLOCATED event concludes this operation.

To summarize, FW\_CDEV\_IOC\_ALLOCATE\_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.

@channels is a host-endian bitfield with the least significant bit representing channel 0 and the most significant bit representing channel 63:  $1ULL \ll c$  for each channel  $c$  that is a candidate for (de)allocation.

@bandwidth 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

## 3.31 fw\_cdev\_create\_iso\_context Struct Reference

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

### 3.31.1 Detailed Description

struct **fw\_cdev\_create\_iso\_context** (p. 15) - Create a context for isochronous IO @type: FW\_CDEV\_ISO\_CONTEXT\_TRANSMIT or FW\_CDEV\_ISO\_CONTEXT\_RECEIVE @header\_size: Header size to strip for receive contexts @channel: Channel to bind to @speed: Speed for transmit contexts @closure: To be returned in &fw\_cdev\_event\_iso\_interrupt (p. 17) @handle: 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.

For receive contexts, @header\_size must be at least 4 and must be a multiple of 4.

Note that the effect of a @header\_size > 4 depends on &fw\_cdev\_get\_info.version, as documented at &fw\_cdev\_event\_iso\_interrupt (p. 17).

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

- dc1394/juju/firewire-cdev.h

## 3.32 fw\_cdev\_deallocate Struct Reference

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

### 3.32.1 Detailed Description

struct **fw\_cdev\_deallocate** (p. 15) - Free a CSR address range or isochronous resource @handle: 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

## 3.33 fw\_cdev\_event Union Reference

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

### 3.33.1 Detailed Description

union **fw\_cdev\_event** (p. 16) - Convenience union of fw\_cdev\_event\_types @common: Valid for all types @bus↵\_reset: Valid if @common.type == FW\_CDEV\_EVENT\_BUS\_RESET @response: Valid if @common.type == FW\_CDEV\_EVENT\_RESPONSE @request: Valid if @common.type == FW\_CDEV\_EVENT\_REQUEST @iso↵\_interrupt: Valid if @common.type == FW\_CDEV\_EVENT\_ISO\_INTERRUPT @iso\_resource: Valid if @common.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

## 3.34 fw\_cdev\_event\_bus\_reset Struct Reference

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

### 3.34.1 Detailed Description

struct **fw\_cdev\_event\_bus\_reset** (p. 16) - Sent when a bus reset occurred @closure: See &fw\_cdev\_event\_↵common; set by FW\_CDEV\_IOC\_GET\_INFO ioctl @type: See &fw\_cdev\_event\_common; always FW\_CDEV\_↵EVENT\_BUS\_RESET @node\_id: New node ID of this node @local\_node\_id: Node ID of the local node, i.e. of the controller @bm\_node\_id: Node ID of the bus manager @irm\_node\_id: Node ID of the iso resource manager @root\_node\_id: Node ID of the root node @generation: 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

## 3.35 fw\_cdev\_event\_common Struct Reference

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

### 3.35.1 Detailed Description

struct **fw\_cdev\_event\_common** (p. 17) - Common part of all fw\_cdev\_event\_ types @closure: For arbitrary use by userspace @type: 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 @closure 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 @closure depends on the @type of event.

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

- dc1394/juju/firewire-cdev.h

## 3.36 fw\_cdev\_event\_iso\_interrupt Struct Reference

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

### 3.36.1 Detailed Description

struct **fw\_cdev\_event\_iso\_interrupt** (p. 17) - Sent when an iso packet was completed @closure: See &fw\_cdev\_event\_common; set by FW\_CDEV\_CREATE\_ISO\_CONTEXT ioctl @type: See &fw\_cdev\_event\_common; always FW\_CDEV\_EVENT\_ISO\_INTERRUPT @cycle: Cycle counter of the interrupt packet @header\_length: Total length of following headers, in bytes @header: Stripped headers, if any

This event is sent when the controller has completed an &fw\_cdev\_iso\_packet (p. 21) 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 @header field. The amount of header data per packet is as specified at iso context creation by &fw\_cdev\_create\_iso\_context.header\_size.

In version 1 of this ABI, header data consisted of the 1394 isochronous packet header, followed by quadlets from the packet payload if &fw\_cdev\_create\_iso\_context.header\_size > 4.

In version 2 of this ABI, header data consist of the 1394 isochronous packet header, followed by a timestamp quadlet if &fw\_cdev\_create\_iso\_context.header\_size > 4, followed by quadlets from the packet payload if &fw\_cdev\_create\_iso\_context.header\_size > 8.

Behaviour of ver. 1 of this ABI is no longer available since ABI ver. 2.

Format of 1394 iso packet header: 16 bits len, 2 bits tag, 6 bits channel, 4 bits tcode, 4 bits sy, in big endian byte order. Format of timestamp: 16 bits invalid, 3 bits cycleSeconds, 13 bits cycleCount, in big endian byte order.

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

- dc1394/juju/firewire-cdev.h

### 3.37 fw\_cdev\_event\_iso\_resource Struct Reference

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

#### 3.37.1 Detailed Description

struct **fw\_cdev\_event\_iso\_resource** (p. 18) - Iso resources were allocated or freed @closure: See &fw\_cdev\_event\_common; set by FW\_CDEV\_IOC\_(DE)ALLOCATE\_ISO\_RESOURCE(\_ONCE) ioctl @type: FW\_CDEV\_EVENT\_ISO\_RESOURCE\_ALLOCATED or FW\_CDEV\_EVENT\_ISO\_RESOURCE\_DEALLOCATED @handle: Reference by which an allocated resource can be deallocated @channel: Isochronous channel which was (de)allocated, if any @bandwidth: Bandwidth allocation units which were (de)allocated, if any

An FW\_CDEV\_EVENT\_ISO\_RESOURCE\_ALLOCATED event is sent after an isochronous resource was allocated at the IRM. The client has to check @channel and @bandwidth 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.

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

### 3.38 fw\_cdev\_event\_request Struct Reference

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

#### 3.38.1 Detailed Description

struct **fw\_cdev\_event\_request** (p. 18) - Sent on incoming request to an address region @closure: See &fw\_cdev\_event\_common; set by FW\_CDEV\_IOC\_ALLOCATE ioctl @type: See &fw\_cdev\_event\_common; always FW\_CDEV\_EVENT\_REQUEST @tcode: Transaction code of the incoming request @offset: The offset into the 48-bit per-node address space @handle: Reference to the kernel-side pending request @length: Data length, i.e. the request's payload size in bytes @data: 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 @handle.

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

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

- dc1394/juju/firewire-cdev.h



## 3.39 fw\_cdev\_event\_response Struct Reference

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

### 3.39.1 Detailed Description

struct **fw\_cdev\_event\_response** (p. 19) - Sent when a response packet was received @closure: See &fw\_cdev\_event\_common; set by FW\_CDEV\_IOC\_SEND\_REQUEST ioctl @type: See &fw\_cdev\_event\_common; always FW\_CDEV\_EVENT\_RESPONSE @rcode: Response code returned by the remote node @length: Data length, i.e. the response's payload size in bytes @data: 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 @data field.

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

- dc1394/juju/firewire-cdev.h

## 3.40 fw\_cdev\_get\_cycle\_timer Struct Reference

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

### 3.40.1 Detailed Description

struct **fw\_cdev\_get\_cycle\_timer** (p. 19) - read cycle timer register @local\_time: system time, in microseconds since the Epoch @cycle\_timer: Cycle Time register contents

The FW\_CDEV\_IOC\_GET\_CYCLE\_TIMER ioctl reads the isochronous cycle timer and also the system clock (CLOCK\_REALTIME). This allows to express the receive time of an isochronous packet as a system time.

@cycle\_timer consists of 7 bits cycleSeconds, 13 bits cycleCount, and 12 bits cycleOffset, in host byte order. Cf. the Cycle Time register per IEEE 1394 or Isochronous Cycle Timer register per OHCI-1394.

In version 1 and 2 of the ABI, this ioctl returned unreliable (non- monotonic) @cycle\_timer values on certain controllers.

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

- dc1394/juju/firewire-cdev.h

## 3.41 fw\_cdev\_get\_cycle\_timer2 Struct Reference

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

### 3.41.1 Detailed Description

struct **fw\_cdev\_get\_cycle\_timer2** (p. 19) - read cycle timer register @tv\_sec: system time, seconds @tv\_nsec: system time, sub-seconds part in nanoseconds @clk\_id: input parameter, clock from which to get the system time @cycle\_timer: Cycle Time register contents

The FW\_CDEV\_IOC\_GET\_CYCLE\_TIMER2 works like FW\_CDEV\_IOC\_GET\_CYCLE\_TIMER but lets you choose a clock like with POSIX' clock\_gettime function. Supported @clk\_id values are POSIX' CLOCK\_REALTIME and CLOCK\_MONOTONIC and Linux' CLOCK\_MONOTONIC\_RAW.

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

- dc1394/juju/firewire-cdev.h

## 3.42 fw\_cdev\_get\_info Struct Reference

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

### 3.42.1 Detailed Description

struct **fw\_cdev\_get\_info** (p. 20) - 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. @rom\_length: If @rom is non-zero, at most rom\_length bytes of configuration ROM will be copied into that user space address. In either case, @rom\_length is updated with the actual length of the configuration ROM. @rom: If non-zero, address of a buffer to be filled by a copy of the device's configuration ROM @bus\_reset: If non-zero, address of a buffer to be filled by a &struct **fw\_cdev\_event\_bus\_reset** (p. 16) with the current state of the bus. This does not cause a bus reset to happen. @bus\_reset\_closure: Value of &closure in this and subsequent bus reset events @card: 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

## 3.43 fw\_cdev\_initiate\_bus\_reset Struct Reference

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

### 3.43.1 Detailed Description

struct **fw\_cdev\_initiate\_bus\_reset** (p. 20) - Initiate a bus reset @type: 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

## 3.44 fw\_cdev\_iso\_packet Struct Reference

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

### 3.44.1 Detailed Description

struct **fw\_cdev\_iso\_packet** (p. 21) - Isochronous packet @control: 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: Header and payload

&struct **fw\_cdev\_iso\_packet** (p. 21) is used to describe isochronous packet queues.

Use the FW\_CDEV\_ISO\_ macros to fill in @control.

For transmit packets, the header length must be a multiple of 4 and specifies the numbers of bytes in @header that will be prepended to the packet's payload; these bytes are copied into the kernel and will not be accessed after the ioctl has returned. The sy and tag fields are copied to the iso packet header (these fields are specified by IEEE 1394a and IEC 61883-1). The skip flag specifies that no packet is to be sent in a frame; when using this, all other fields except the interrupt flag must be zero.

For receive packets, the header length must be a multiple of the context's header size; if the header length is larger than the context's header size, multiple packets are queued for this entry. The sy and tag fields are ignored. If the sync flag is set, the context drops all packets until a packet with a matching sy field is received (the sync value to wait for is specified in the &fw\_cdev\_start\_iso (p. 23) structure). The payload length defines how many payload bytes can be received for one packet (in addition to payload quadlets that have been defined as headers and are stripped and returned in the &fw\_cdev\_event\_iso\_interrupt (p. 17) structure). If more bytes are received, the additional bytes are dropped. If less bytes are received, the remaining bytes in this part of the payload buffer will not be written to, not even by the next packet, i.e., packets received in consecutive frames will not necessarily be consecutive in memory. If an entry has queued multiple packets, the payload length is divided equally among them.

When a packet with the interrupt flag set has been completed, the &fw\_cdev\_event\_iso\_interrupt (p. 17) event will be sent. An entry that has queued multiple receive packets is completed when its last packet is completed.

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

- dc1394/juju/firewire-cdev.h

## 3.45 fw\_cdev\_queue\_iso Struct Reference

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

### 3.45.1 Detailed Description

struct **fw\_cdev\_queue\_iso** (p. 21) - Queue isochronous packets for I/O @packets: Userspace pointer to packet data @data: Pointer into mmap()'ed payload buffer @size: Size of packet data in bytes @handle: 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. 21) structs, which describe how to transmit from or receive into a contiguous region of a mmap()'ed payload buffer. As part of transmit 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 @packets, @data and @size fields, so the ioctl can be resubmitted easily.

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

- dc1394/juju/firewire-cdev.h

## 3.46 fw\_cdev\_remove\_descriptor Struct Reference

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

### 3.46.1 Detailed Description

struct **fw\_cdev\_remove\_descriptor** (p. 22) - Remove contents from the configuration ROM @handle: 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 nodes' configuration ROMs.

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

- dc1394/juju/firewire-cdev.h

## 3.47 fw\_cdev\_send\_request Struct Reference

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

### 3.47.1 Detailed Description

struct **fw\_cdev\_send\_request** (p. 22) - Send an asynchronous request packet @tcode: Transaction code of the request @length: Length of outgoing payload, in bytes @offset: 48-bit offset at destination node @closure: Passed back to userspace in the response event @data: Userspace pointer to payload @generation: 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 @data field. Once the transaction completes, the kernel writes an **&fw\_cdev\_event\_response** (p. 19) event back. The @closure 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

## 3.48 fw\_cdev\_send\_response Struct Reference

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

### 3.48.1 Detailed Description

struct **fw\_cdev\_send\_response** (p. 22) - Send an asynchronous response packet @rcode: Response code as determined by the userspace handler @length: Length of outgoing payload, in bytes @data: Userspace pointer to payload @handle: The handle from the &**fw\_cdev\_event\_request** (p. 18)

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

## 3.49 fw\_cdev\_send\_stream\_packet Struct Reference

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

### 3.49.1 Detailed Description

struct **fw\_cdev\_send\_stream\_packet** (p. 23) - send an asynchronous stream packet @length: Length of outgoing payload, in bytes @tag: Data format tag @channel: Isochronous channel to transmit to @sy: Synchronization code @closure: Passed back to userspace in the response event @data: Userspace pointer to payload @generation: The bus generation where packet is valid @speed: Speed to transmit at

The FW\_CDEV\_IOC\_SEND\_STREAM\_PACKET ioctl sends an asynchronous stream packet to every device which is listening to the specified channel. The kernel writes an &**fw\_cdev\_event\_response** (p. 19) event which indicates success or failure of the transmission.

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

- dc1394/juju/firewire-cdev.h

## 3.50 fw\_cdev\_start\_iso Struct Reference

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

### 3.50.1 Detailed Description

struct **fw\_cdev\_start\_iso** (p. 23) - Start an isochronous transmission or reception @cycle: Cycle in which to start I/O. If @cycle is greater than or equal to 0, the I/O will start on that cycle. @sync: Determines the value to wait for for receive packets that have the FW\_CDEV\_ISO\_SYNC bit set @tags: 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 @tags. @handle: 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

## 3.51 fw\_cdev\_stop\_iso Struct Reference

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

### 3.51.1 Detailed Description

struct **fw\_cdev\_stop\_iso** (p. 24) - Stop an isochronous transmission or reception @handle: 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 <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 **dc1394iic\_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)
- **dc1394error\_t** **dc1394\_camera\_get\_platform\_string** ( **dc1394camera\_t** \*camera, const char \*\*platform)

### 4.1.1 Detailed Description

Basic system and camera functions.

#### Author

Damien Douchamps: coding

Peter Antoniac: documentation maintainer

More details soon

### 4.1.2 Typedef Documentation

#### 4.1.2.1 dc1394camera\_list\_t

```
typedef struct __dc1394camera_list_t dc1394camera_list_t
```

A list of cameras

Usually returned by dc1394\_camera\_eumerate().

#### 4.1.2.2 dc1394camera\_t

```
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 dc1394iidx\_version\_t

```
enum dc1394iidx_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 dc1394phy\_delay\_t

```
enum dc1394phy_delay_t
```

Enumeration of PHY delays

This is currently not used in libdc1394.

#### 4.1.3.3 dc1394power\_class\_t

```
enum dc1394power_class_t
```

Enumeration of power classes

This is currently not used in libdc1394.

### 4.1.4 Function Documentation

#### 4.1.4.1 dc1394\_camera\_enumerate()

```
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 dc1394\_camera\_free()

```
void dc1394_camera_free (
    dc1394camera_t * camera )
```

Frees a camera structure

#### 4.1.4.3 dc1394\_camera\_free\_list()

```
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 dc1394\_camera\_get\_node()

```
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 dc1394\_camera\_get\_platform\_string()

```
dc1394error_t dc1394_camera_get_platform_string (
    dc1394camera_t * camera,
    const char ** platform )
```

Returns a pointer to a string identifying the platform for the cameras. Platforms strings are: juju, linux, macosx, windows, usb

#### 4.1.4.6 dc1394\_camera\_new()

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

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

#### 4.1.4.7 dc1394\_camera\_new\_unit()

```
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)

#### 4.1.4.8 dc1394\_camera\_print\_info()

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

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

#### 4.1.4.9 dc1394\_camera\_set\_broadcast()

```
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).

#### 4.1.4.10 dc1394\_free()

```
void dc1394_free (
    dc1394_t * dc1394 )
```

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

#### 4.1.4.11 dc1394\_new()

```
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.

#### 4.1.4.12 dc1394\_reset\_bus()

```
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/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\_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)

## 4.2.1 Detailed Description

functions to convert video formats

### Author

Damien Douxchamps: coding

Frederic Devernay: coding

Peter Antoniac: documentation maintainer

More details soon

## 4.2.2 Enumeration Type Documentation

### 4.2.2.1 dc1394bayer\_method\_t

```
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.

### 4.2.2.2 dc1394stereo\_method\_t

```
enum dc1394stereo_method_t
```

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

## 4.2.3 Function Documentation

### 4.2.3.1 dc1394\_bayer\_decoding\_16bit()

```
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

#### 4.2.3.2 dc1394\_bayer\_decoding\_8bit()

```
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

#### 4.2.3.3 dc1394\_convert\_frames()

```
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

#### 4.2.3.4 dc1394\_convert\_to\_MONO8()

```
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

#### 4.2.3.5 dc1394\_convert\_to\_RGB8()

```
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

#### 4.2.3.6 dc1394\_convert\_to\_YUV422()

```
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

#### 4.2.3.7 dc1394\_debayer\_frames()

```
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

<i>in</i>	is a pointer to the bayer video frame that is to be converted
<i>out</i>	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
<i>method</i>	is the bayer method to interpolate the frame.

#### 4.2.3.8 dc1394\_deinterlace\_stereo()

```
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

#### 4.2.3.9 dc1394\_deinterlace\_stereo\_frames()

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

De-interlacing of stereo data for video frames

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

## 4.3 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>
```

### 4.3.1 Detailed Description

Main include file, which include all others.

More details soon

## 4.4 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)

### 4.4.1 Detailed Description

Functions to control Format\_7 (aka scalable format, ROI)

More details soon



## 4.4.2 Typedef Documentation

### 4.4.2.1 dc1394format7mode\_t

```
typedef struct __dc1394format7mode_t dc1394format7mode_t
```

A struct containing information about a mode of Format\_7, the scalable image format.

### 4.4.2.2 dc1394format7modeset\_t

```
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.

## 4.4.3 Function Documentation

### 4.4.3.1 dc1394\_format7\_get\_color\_coding()

```
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

### 4.4.3.2 dc1394\_format7\_get\_color\_codings()

```
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

### 4.4.3.3 dc1394\_format7\_get\_color\_filter()

```
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

#### 4.4.3.4 dc1394\_format7\_get\_data\_depth()

```
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)

#### 4.4.3.5 dc1394\_format7\_get\_frame\_interval()

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

Gets the frame interval in float format

#### 4.4.3.6 dc1394\_format7\_get\_image\_position()

```
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

#### 4.4.3.7 dc1394\_format7\_get\_image\_size()

```
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.

#### 4.4.3.8 dc1394\_format7\_get\_max\_image\_size()

```
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.

#### 4.4.3.9 dc1394\_format7\_get\_mode\_info()

```
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

#### 4.4.3.10 dc1394\_format7\_get\_modeset()

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

Gets the properties of all Format\_7 modes supported by the camera.

#### 4.4.3.11 dc1394\_format7\_get\_packet\_parameters()

```
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.

#### 4.4.3.12 dc1394\_format7\_get\_packet\_size()

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

Gets the current packet size

#### 4.4.3.13 dc1394\_format7\_get\_packets\_per\_frame()

```
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.

#### 4.4.3.14 dc1394\_format7\_get\_pixel\_number()

```
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

#### 4.4.3.15 dc1394\_format7\_get\_recommended\_packet\_size()

```
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.

#### 4.4.3.16 dc1394\_format7\_get\_roi()

```
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.

#### 4.4.3.17 dc1394\_format7\_get\_total\_bytes()

```
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)

#### 4.4.3.18 dc1394\_format7\_get\_unit\_position()

```
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).

#### 4.4.3.19 dc1394\_format7\_get\_unit\_size()

```
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.

#### 4.4.3.20 dc1394\_format7\_set\_color\_coding()

```
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

#### 4.4.3.21 dc1394\_format7\_set\_image\_position()

```
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

#### 4.4.3.22 dc1394\_format7\_set\_image\_size()

```
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

#### 4.4.3.23 dc1394\_format7\_set\_packet\_size()

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

Sets the current packet size

#### 4.4.3.24 dc1394\_format7\_set\_roi()

```
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 merely sets it

## 4.5 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)

#### 4.5.1 Detailed Description

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

##### Author

Damien Douxchamps: coding

Peter Antoniac: documentation maintainer

More details soon

#### 4.5.2 Function Documentation

##### 4.5.2.1 dc1394\_iso\_allocate\_bandwidth()

```
dc1394error_t dc1394_iso_allocate_bandwidth (
    dc1394camera_t * camera,
    int bandwidth_units )
```

dc1394\_iso\_allocate\_bandwidth:

## Parameters

<i>camera</i>	A camera handle.
<i>bandwidth_units</i>	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. 43) 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.

## 4.5.2.2 dc1394\_iso\_allocate\_channel()

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

dc1394\_iso\_allocate\_channel:

## Parameters

<i>camera</i>	A camera handle.
<i>channels_allowed</i>	A bitmask of acceptable channels for the allocation. The LSB corresponds to channel 0 and the MSB corresponds to channel  1. Only channels whose bit is set will be considered for the allocation. If <i>channels_allowed</i> = 0, the complete set of channels supported by this camera will be considered for the allocation.
<i>channel</i>	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. 43) 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. 97).

## 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.

#### 4.5.2.3 dc1394\_iso\_release\_all()

```
dc1394error_t dc1394_iso_release_all (
    dc1394camera_t * camera )
```

dc1394\_iso\_release\_all:

##### Parameters

<i>camera</i>	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. 42) and **`dc1394_iso_release_bandwidth()`** (p. 42) must be used.

##### Returns

*DC1394\_SUCCESS* if the operation succeeded. *DC1394\_FAILURE* if some resources were not able to be released.

#### 4.5.2.4 dc1394\_iso\_release\_bandwidth()

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

dc1394\_iso\_release\_bandwidth:

##### Parameters

<i>camera</i>	A camera handle.
<i>bandwidth_units</i>	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.



#### 4.5.2.5 dc1394\_iso\_release\_channel()

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

dc1394\_iso\_release\_channel:

##### Parameters

<i>camera</i>	A camera handle.
<i>channel</i>	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.

#### 4.5.2.6 dc1394\_iso\_set\_persist()

```
dc1394error_t dc1394_iso_set_persist (
    dc1394camera_t * camera )
```

dc1394\_iso\_set\_persist

##### Parameters

<i>camera</i>	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.

## 4.6 dc1394/capture.h File Reference

Capture functions.

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

## Macros

- `#define DC1394_CAPTURE_FLAGS_CHANNEL_ALLOC 0x00000001U`

## Typedefs

- `typedef void(* dc1394capture_callback_t) ( dc1394camera_t *, void *)`

## 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)`
- `void dc1394_capture_set_callback ( dc1394camera_t *camera, dc1394capture_callback_t callback, void *user_data)`

### 4.6.1 Detailed Description

Capture functions.

#### Author

Damien Douxchamps: coding

Peter Antoniac: documentation maintainer

More details soon

### 4.6.2 Macro Definition Documentation

#### 4.6.2.1 DC1394\_CAPTURE\_FLAGS\_CHANNEL\_ALLOC

```
#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

### 4.6.3 Typedef Documentation

#### 4.6.3.1 dc1394capture\_callback\_t

```
typedef void(* dc1394capture_callback_t) ( dc1394camera_t *, void *)
```

typedef for the callback param for dc1394\_capture\_set\_callback

### 4.6.4 Enumeration Type Documentation

#### 4.6.4.1 dc1394capture\_policy\_t

```
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)

### 4.6.5 Function Documentation

#### 4.6.5.1 dc1394\_capture\_dequeue()

```
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.

#### 4.6.5.2 dc1394\_capture\_enqueue()

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

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

#### 4.6.5.3 dc1394\_capture\_get\_fileno()

```
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. 46).

#### 4.6.5.4 dc1394\_capture\_is\_frame\_corrupt()

```
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. 45) when no longer needed by the user.

#### 4.6.5.5 dc1394\_capture\_set\_callback()

```
void dc1394_capture_set_callback (
    dc1394camera_t * camera,
    dc1394capture_callback_t callback,
    void * user_data )
```

Set a callback if supported by the platform (OS X only for now).

#### 4.6.5.6 dc1394\_capture\_setup()

```
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)

#### 4.6.5.7 dc1394\_capture\_stop()

```
dc1394error_t dc1394_capture_stop (
    dc1394camera_t * camera )
```

Stop the capture

## 4.7 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\_t** polarity\_t polarity)
- **dc1394error\_t dc1394\_external\_trigger\_get\_polarity** ( **dc1394camera\_t** \*camera, **dc1394trigger\_t** 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\_t** mode\_t mode)
- **dc1394error\_t dc1394\_external\_trigger\_get\_mode** ( **dc1394camera\_t** \*camera, **dc1394trigger\_t** mode\_t \*mode)
- **dc1394error\_t dc1394\_external\_trigger\_set\_source** ( **dc1394camera\_t** \*camera, **dc1394trigger\_t** source\_t source)
- **dc1394error\_t dc1394\_external\_trigger\_get\_source** ( **dc1394camera\_t** \*camera, **dc1394trigger\_t** 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)

#### 4.7.1 Detailed Description

Diverse controls of camera features.

#### Author

Gord Peters: main writer  
Chris Urmson: some additions  
Damien Douchamps: 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.

## 4.7.2 Typedef Documentation

### 4.7.2.1 dc1394feature\_info\_t

```
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,...)

### 4.7.2.2 dc1394featureset\_t

```
typedef struct __dc1394featureset_t dc1394featureset_t
```

The list of features

## 4.7.3 Enumeration Type Documentation

### 4.7.3.1 dc1394feature\_mode\_t

```
enum dc1394feature_mode_t
```

Control modes for a feature (excl. absolute control)

### 4.7.3.2 dc1394feature\_t

```
enum dc1394feature_t
```

Enumeration of camera features

#### 4.7.3.3 dc1394trigger\_mode\_t

```
enum dc1394trigger_mode_t
```

Enumeration of trigger modes

#### 4.7.3.4 dc1394trigger\_polarity\_t

```
enum dc1394trigger_polarity_t
```

External trigger polarity

#### 4.7.3.5 dc1394trigger\_source\_t

```
enum dc1394trigger_source_t
```

Enumeration of trigger sources

### 4.7.4 Function Documentation

#### 4.7.4.1 dc1394\_camera\_reset()

```
dc1394error_t dc1394_camera_reset (
    dc1394camera_t * camera )
```

reset a camera to factory default settings

#### 4.7.4.2 dc1394\_camera\_set\_power()

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

turn a camera on or off

#### 4.7.4.3 dc1394\_external\_trigger\_get\_mode()

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

Gets the external trigger mode



#### 4.7.4.4 dc1394\_external\_trigger\_get\_polarity()

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

Gets the polarity of the external trigger

#### 4.7.4.5 dc1394\_external\_trigger\_get\_power()

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

Gets the status of the external trigger

#### 4.7.4.6 dc1394\_external\_trigger\_get\_source()

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

Gets the external trigger source

#### 4.7.4.7 dc1394\_external\_trigger\_get\_supported\_sources()

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

Gets the list of available external trigger source

#### 4.7.4.8 dc1394\_external\_trigger\_has\_polarity()

```
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.

#### 4.7.4.9 dc1394\_external\_trigger\_set\_mode()

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

Sets the external trigger mode

#### 4.7.4.10 dc1394\_external\_trigger\_set\_polarity()

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

Sets the polarity of the external trigger

#### 4.7.4.11 dc1394\_external\_trigger\_set\_power()

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

Switch between internal and external trigger

#### 4.7.4.12 dc1394\_external\_trigger\_set\_source()

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

Sets the external trigger source

#### 4.7.4.13 dc1394\_feature\_get()

```
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

#### 4.7.4.14 dc1394\_feature\_get\_absolute\_boundaries()

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

Gets the absolute boundaries of a feature

#### 4.7.4.15 dc1394\_feature\_get\_absolute\_control()

```
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

#### 4.7.4.16 dc1394\_feature\_get\_absolute\_value()

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

Gets the absolute value of a feature

#### 4.7.4.17 dc1394\_feature\_get\_all()

```
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.

#### 4.7.4.18 dc1394\_feature\_get\_boundaries()

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

Gets the boundaries of a feature

#### 4.7.4.19 dc1394\_feature\_get\_mode()

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

Gets the current control modes for a feature

#### 4.7.4.20 dc1394\_feature\_get\_modes()

```
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...)

#### 4.7.4.21 dc1394\_feature\_get\_power()

```
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)

#### 4.7.4.22 dc1394\_feature\_get\_value()

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

Gets the value of a feature

#### 4.7.4.23 dc1394\_feature\_has\_absolute\_control()

```
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

#### 4.7.4.24 dc1394\_feature\_is\_present()

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

Tells whether a feature is present or not

#### 4.7.4.25 dc1394\_feature\_is\_readable()

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

Tells whether a feature is readable or not

#### 4.7.4.26 dc1394\_feature\_is\_switchable()

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

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

#### 4.7.4.27 dc1394\_feature\_print()

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

Displays the bounds and options of the given feature

#### 4.7.4.28 dc1394\_feature\_print\_all()

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

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

#### 4.7.4.29 dc1394\_feature\_set\_absolute\_control()

```
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)

#### 4.7.4.30 dc1394\_feature\_set\_absolute\_value()

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

Sets the absolute value of a feature

#### 4.7.4.31 dc1394\_feature\_set\_mode()

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

Sets the current control modes for a feature

#### 4.7.4.32 dc1394\_feature\_set\_power()

```
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)

#### 4.7.4.33 dc1394\_feature\_set\_value()

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

Sets the value of a feature

#### 4.7.4.34 dc1394\_feature\_temperature\_get\_value()

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

Gets the temperature values (target and current)

#### 4.7.4.35 dc1394\_feature\_temperature\_set\_value()

```
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()? ↩

#### 4.7.4.36 dc1394\_feature\_whitebalance\_get\_value()

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

Gets the whitebalance values

#### 4.7.4.37 dc1394\_feature\_whitebalance\_set\_value()

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

Sets the whitebalance values

#### 4.7.4.38 dc1394\_feature\_whiteshading\_get\_value()

```
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

#### 4.7.4.39 dc1394\_feature\_whiteshading\_set\_value()

```
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

#### 4.7.4.40 dc1394\_memory\_busy()

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

Download a camera setup from the memory.

#### 4.7.4.41 dc1394\_memory\_load()

```
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.

#### 4.7.4.42 dc1394\_memory\_save()

```
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.

#### 4.7.4.43 dc1394\_pio\_get()

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

Gets the current quadlet at the PIO (input)

#### 4.7.4.44 dc1394\_pio\_set()

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

Sends a quadlet on the PIO (output)

#### 4.7.4.45 dc1394\_software\_trigger\_get\_power()

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

Gets the state of software trigger

#### 4.7.4.46 dc1394\_software\_trigger\_set\_power()

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

Turn software trigger on or off

## 4.8 dc1394/log.h File Reference

Functions to log errors, warning and debug messages.

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

### Enumerations

- enum **dc1394error\_t**
- enum **dc1394log\_t**

### Functions

- **dc1394error\_t dc1394\_log\_register\_handler** ( **dc1394log\_t** type, void(\*log\_handler)( **dc1394log\_t** type, const char \*message, void \*user), void \*user)
- **dc1394error\_t dc1394\_log\_set\_default\_handler** ( **dc1394log\_t** type)
- void **dc1394\_log\_error** (const char \*format,...)
- void **dc1394\_log\_warning** (const char \*format,...)
- void **dc1394\_log\_debug** (const char \*format,...)

#### 4.8.1 Detailed Description

Functions to log errors, warning and debug messages.

##### Author

Damien Douxchamps: coding

Rudolf Leitgeb: coding

Peter Antoniac: documentation maintainer

More details soon



## 4.8.2 Enumeration Type Documentation

### 4.8.2.1 dc1394error\_t

enum **dc1394error\_t**

Error codes returned by most libdc1394 functions.

General rule: 0 is success, negative denotes a problem.

### 4.8.2.2 dc1394log\_t

enum **dc1394log\_t**

Types of logging messages

Three types exist:

- ERROR for real, hard, unrecoverable errors that will result in the program terminating.
- WARNING for things that have gone wrong, but are not requiring a termination of the program.
- DEBUG for debug messages that can be very verbose but may help the developers to fix bugs.

## 4.8.3 Function Documentation

### 4.8.3.1 dc1394\_log\_debug()

```
void dc1394_log_debug (  
    const char * format,  
    ... )
```

dc1394\_log\_debug: logs a debug statement to the registered facility This function shall be invoked if a debug statement is to be logged. The message passed as argument is delivered to the registered debug reporting function registered before ONLY IF the environment variable DC1394\_DEBUG has been set before the program starts.

#### Parameters

in	<i>format</i> ,...	debug statement to be logged, multiple arguments allowed (printf style)
----	--------------------	---

#### 4.8.3.2 dc1394\_log\_error()

```
void dc1394_log_error (
    const char * format,
    ... )
```

dc1394\_log\_error: logs a fatal error condition to the registered facility This function shall be invoked if a fatal error condition is encountered. The message passed as argument is delivered to the registered error reporting function registered before.

##### Parameters

in	<i>format</i> ,...	error message to be logged, multiple arguments allowed (printf style)
----	--------------------	---

#### 4.8.3.3 dc1394\_log\_register\_handler()

```
dc1394error_t dc1394_log_register_handler (
    dc1394log_t type,
    void(*) ( dc1394log_t type, const char *message, void *user) log_handler,
    void * user )
```

dc1394\_log\_register\_handler: register log handler for reporting error, warning or debug statements Passing NULL as argument turns off this log level.

##### Parameters

in	<i>log_handler</i>	pointer to a function which takes a character string as argument type: the type of log
in	<i>type</i>	message type ( <i>debug</i> , <i>err</i> or <i>warning</i> )
in	<i>message</i>	log message

#### 4.8.3.4 dc1394\_log\_set\_default\_handler()

```
dc1394error_t dc1394_log_set_default_handler (
    dc1394log_t type )
```

dc1394\_log\_set\_default\_handler: set the log handler to the default handler At boot time, debug logging is OFF (handler is NULL). Using this function for the debug statements will start logging of debug statements usng the default handler.

#### 4.8.3.5 dc1394\_log\_warning()

```
void dc1394_log_warning (
    const char * format,
    ... )
```

dc1394\_log\_warning: logs a nonfatal error condition to the registered facility This function shall be invoked if a nonfatal error condition is encountered. The message passed as argument is delivered to the registered warning reporting function registered before.

## Parameters

in	<i>format,...</i>	warning message to be logged, multiple arguments allowed (printf style)
----	-------------------	---

## 4.9 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)

### 4.9.1 Detailed Description

Functions to directly access camera registers.

#### Author

Damien Douxchamps: coding

Peter Antoniac: documentation maintainer

More details soon

## 4.9.2 Function Documentation

### 4.9.2.1 dc1394\_get\_absolute\_register()

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

No Docs

### 4.9.2.2 dc1394\_get\_adv\_control\_registers()

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

No Docs

### 4.9.2.3 dc1394\_get\_control\_registers()

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

No Docs

### 4.9.2.4 dc1394\_get\_format7\_register()

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

No Docs

### 4.9.2.5 dc1394\_get\_PIO\_register()

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

No Docs

#### 4.9.2.6 dc1394\_get\_registers()

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

No Docs

#### 4.9.2.7 dc1394\_get\_SIO\_register()

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

No Docs

#### 4.9.2.8 dc1394\_get\_strobe\_register()

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

No Docs

#### 4.9.2.9 dc1394\_set\_absolute\_register()

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

No Docs

#### 4.9.2.10 dc1394\_set\_adv\_control\_registers()

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

No Docs

#### 4.9.2.11 dc1394\_set\_control\_registers()

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

No Docs

#### 4.9.2.12 dc1394\_set\_format7\_register()

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

No Docs

#### 4.9.2.13 dc1394\_set\_PIO\_register()

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

No Docs

#### 4.9.2.14 dc1394\_set\_registers()

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

No Docs

#### 4.9.2.15 dc1394\_set\_SIO\_register()

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

No Docs

#### 4.9.2.16 dc1394\_set\_strobe\_register()

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

No Docs

## 4.10 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**

#### 4.10.1 Detailed Description

Various types that must be defined here.

More details soon

#### 4.10.2 Enumeration Type Documentation

##### 4.10.2.1 dc1394bool\_t

```
enum dc1394bool_t
```

Yet another boolean data type

#### 4.10.2.2 `dc1394byte_order_t`

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.

#### 4.10.2.3 `dc1394color_coding_t`

enum `dc1394color_coding_t`

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

#### 4.10.2.4 `dc1394color_filter_t`

enum `dc1394color_filter_t`

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

RG GB

and similarly for other filters.

#### 4.10.2.5 `dc1394switch_t`

enum `dc1394switch_t`

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

#### 4.10.2.6 `dc1394video_mode_t`

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.

## 4.11 `dc1394/utils.h` File Reference

Utility functions.



## Functions

- **dc1394error\_t dc1394\_get\_image\_size\_from\_video\_mode** ( **dc1394camera\_t** \*camera, **dc1394video\_mode\_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)

### 4.11.1 Detailed Description

Utility functions.

Author

Damien Douxchamps: coding

Peter Antoniac: documentation maintainer

More details soon

### 4.11.2 Function Documentation

#### 4.11.2.1 dc1394\_checksum\_crc16()

```
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.

#### 4.11.2.2 dc1394\_error\_get\_string()

```
const char * dc1394_error_get_string (
    dc1394error_t error )
```

Returns a descriptive string for an error code

#### 4.11.2.3 dc1394\_feature\_get\_string()

```
const char * dc1394_feature_get_string (
    dc1394feature_t feature )
```

Returns a descriptive name for a feature

#### 4.11.2.4 dc1394\_framerate\_as\_float()

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

Returns the given framerate as a float.

#### 4.11.2.5 dc1394\_get\_color\_coding\_bit\_size()

```
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.

#### 4.11.2.6 dc1394\_get\_color\_coding\_data\_depth()

```
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.

#### 4.11.2.7 dc1394\_get\_color\_coding\_from\_video\_mode()

```
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.

#### 4.11.2.8 dc1394\_get\_image\_size\_from\_video\_mode()

```
dc1394error_t dc1394_get_image_size_from_video_mode (
    dc1394camera_t * camera,
    dc1394video_mode_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.

#### 4.11.2.9 dc1394\_is\_color()

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

Tells whether the color mode is color or monochrome

#### 4.11.2.10 dc1394\_is\_same\_camera()

```
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.

#### 4.11.2.11 dc1394\_is\_video\_mode\_scalable()

```
dc1394bool_t dc1394_is_video_mode_scalable (
    dc1394video_mode_t video_mode )
```

Tells whether the video mode is scalable or not.

#### 4.11.2.12 dc1394\_is\_video\_mode\_still\_image()

```
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)

## 4.12 dc1394/vendor/avt.h File Reference

No docs yet.

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

## 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\_get\_advanced\_feature\_inquiry** ( **dc1394camera\_t** \*camera, dc1394\_avt\_adv\_feature\_info\_t \*adv\_feature)
- **dc1394error\_t dc1394\_avt\_print\_advanced\_feature** (dc1394\_avt\_adv\_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\_set\_shading** ( **dc1394camera\_t** \*camera, **dc1394bool\_t** on\_off, **dc1394bool\_t** compute, **dc1394bool\_t** show, uint32\_t frame\_nb)
- **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\_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\_set\_dsnu** ( **dc1394camera\_t** \*camera, **dc1394bool\_t** on\_off, **dc1394bool\_t** compute, uint32\_t frame\_nb)
- **dc1394error\_t dc1394\_avt\_get\_blemish** ( **dc1394camera\_t** \*camera, **dc1394bool\_t** \*on\_off, uint32\_t \*frame\_nb)
- **dc1394error\_t dc1394\_avt\_set\_blemish** ( **dc1394camera\_t** \*camera, **dc1394bool\_t** on\_off, **dc1394bool\_t** compute, uint32\_t frame\_nb)
- **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\_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\_aoi** ( **dc1394camera\_t** \*camera, **dc1394bool\_t** \*on\_off, int \*left, int \*top, int \*width, int \*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\_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\_reset\_frame\_info** ( **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\_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\_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)

## 4.12.1 Detailed Description

No docs yet.

More details soon

## 4.12.2 Function Documentation

### 4.12.2.1 dc1394\_avt\_get\_advanced\_feature\_inquiry()

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

Retrieve the supported features

#### 4.12.2.2 dc1394\_avt\_get\_aoi()

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

Get on/off and area

#### 4.12.2.3 dc1394\_avt\_get\_auto\_gain()

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

Get min and max gain values for autogain

#### 4.12.2.4 dc1394\_avt\_get\_auto\_shutter()

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

Get min and max shutter values for autoshutter

#### 4.12.2.5 dc1394\_avt\_get\_blemish()

```
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 computing the correction

#### 4.12.2.6 dc1394\_avt\_get\_channel\_adjust()

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

Read channel adjust (AVT Pike)

#### 4.12.2.7 dc1394\_avt\_get\_color\_corr()

```
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

#### 4.12.2.8 dc1394\_avt\_get\_deferred\_trans()

```
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

#### 4.12.2.9 dc1394\_avt\_get\_dsnu()

```
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 computing dsnu correction

#### 4.12.2.10 dc1394\_avt\_get\_extented\_shutter()

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

Get the extented shutter value in us

#### 4.12.2.11 dc1394\_avt\_get\_frame\_info()

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

Get the number of captured frames

#### 4.12.2.12 dc1394\_avt\_get\_gpdata\_info()

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

Get the size of the buffer

#### 4.12.2.13 dc1394\_avt\_get\_hsnr()

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

Get HSNR

#### 4.12.2.14 dc1394\_avt\_get\_io()

```
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

#### 4.12.2.15 dc1394\_avt\_get\_lut()

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

Get on/off and the num of the current lut loaded

#### 4.12.2.16 dc1394\_avt\_get\_lut\_info()

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

Get num of luts present and the max size



#### 4.12.2.17 dc1394\_avt\_get\_lut\_mem\_ctrl()

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

Get access mode of a lut

#### 4.12.2.18 dc1394\_avt\_get\_MaxResolution()

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

Get the Max reachable resolution

#### 4.12.2.19 dc1394\_avt\_get\_mirror()

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

Get mirror mode

#### 4.12.2.20 dc1394\_avt\_get\_multiple\_slope()

```
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 )
```

Retrieve if on/off, the nb of kneepoints used and the kneepoints values

#### 4.12.2.21 dc1394\_avt\_get\_shading()

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

Retrieve if shading is on and the number of frames used to compute the shading reference frame

#### 4.12.2.22 dc1394\_avt\_get\_shading\_info()

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

Retrieve the max size of a shading image

#### 4.12.2.23 dc1394\_avt\_get\_shading\_mem\_ctrl()

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

Retrieve write and read access mode of the shading reference frame

#### 4.12.2.24 dc1394\_avt\_get\_test\_images()

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

Get current test image

#### 4.12.2.25 dc1394\_avt\_get\_timebase()

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

Get the timebase value with an Id. See Manual for correspondance

#### 4.12.2.26 dc1394\_avt\_get\_trigger\_delay()

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

Get if trigger delay on and the trigger delay

#### 4.12.2.27 dc1394\_avt\_get\_version()

```
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

#### 4.12.2.28 dc1394\_avt\_print\_advanced\_feature()

```
dc1394error_t dc1394_avt_print_advanced_feature (
    dc1394_avt_adv_feature_info_t * adv_feature )
```

Print the supported features requested

#### 4.12.2.29 dc1394\_avt\_read\_gpdata()

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

Read size number of bytes from GPData buffe

#### 4.12.2.30 dc1394\_avt\_read\_shading\_img()

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

Read shading image from camera into buffer

#### 4.12.2.31 dc1394\_avt\_reset()

```
dc1394error_t dc1394_avt_reset (
    dc1394camera_t * camera )
```

Reset the bus and the fpga

#### 4.12.2.32 dc1394\_avt\_reset\_frame\_info()

```
dc1394error_t dc1394_avt_reset_frame_info (
    dc1394camera_t * camera )
```

Reset frame counter

#### 4.12.2.33 dc1394\_avt\_set\_aoi()

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

Set on/off and area

**4.12.2.34 dc1394\_avt\_set\_auto\_gain()**

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

Set min and max gain values for autogain

**4.12.2.35 dc1394\_avt\_set\_auto\_shutter()**

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

Set min and max shutter values for autoshutter

**4.12.2.36 dc1394\_avt\_set\_blemish()**

```
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 computing and launch the the computation of the blemish correction

**4.12.2.37 dc1394\_avt\_set\_channel\_adjust()**

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

Write channel adjust (AVT Pike)

**4.12.2.38 dc1394\_avt\_set\_color\_corr()**

```
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

#### 4.12.2.39 dc1394\_avt\_set\_deferred\_trans()

```
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

#### 4.12.2.40 dc1394\_avt\_set\_dsnu()

```
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 computing and launch the the computation of the dsnu frame

#### 4.12.2.41 dc1394\_avt\_set\_extented\_shutter()

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

Set the extented shutter value in us

#### 4.12.2.42 dc1394\_avt\_set\_hsnr()

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

Set HSNR

#### 4.12.2.43 dc1394\_avt\_set\_io()

```
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

#### 4.12.2.44 dc1394\_avt\_set\_lut()

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

Set on/off and the num of the current lut to loa

#### 4.12.2.45 dc1394\_avt\_set\_lut\_mem\_ctrl()

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

Set access mode of a lut

#### 4.12.2.46 dc1394\_avt\_set\_mirror()

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

Set mirror mode

#### 4.12.2.47 dc1394\_avt\_set\_multiple\_slope()

```
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 on/off, the nb of kneepoints to use and the kneepoints values

#### 4.12.2.48 dc1394\_avt\_set\_shading()

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

Set the shading to on/off and the number of frames used to compute the shading reference frame

#### 4.12.2.49 dc1394\_avt\_set\_shading\_mem\_ctrl()

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

Set write and read access mode of the shading reference frame

#### 4.12.2.50 dc1394\_avt\_set\_test\_images()

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

Set num of test image

#### 4.12.2.51 dc1394\_avt\_set\_timebase()

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

Set the timebase value with an Id. See Manual for correspondance

#### 4.12.2.52 dc1394\_avt\_set\_trigger\_delay()

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

Set trigger delay on/off and the trigger delay value

#### 4.12.2.53 dc1394\_avt\_write\_gpdata()

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

Write size number of bytes to GPData buffer

#### 4.12.2.54 dc1394\_avt\_write\_shading\_img()

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

Write shading image from buffer to camera

## 4.13 dc1394/vendor/basler.h File Reference

No docs yet.

```
#include "basler_sff.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)

### 4.13.1 Detailed Description

No docs yet.

More details soon

### 4.13.2 Function Documentation

#### 4.13.2.1 dc1394\_basler\_sff\_check\_crc()

```
dc1394bool_t dc1394_basler_sff_check_crc (
    const uint8_t * frame_buffer,
    uint32_t frame_size )
```

Checks the crc checksum of the supplied frame



#### 4.13.2.2 dc1394\_basler\_sff\_chunk\_find()

```
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

#### 4.13.2.3 dc1394\_basler\_sff\_chunk\_iterate()

```
dc1394error_t dc1394_basler_sff_chunk_iterate (
    dc1394basler_sff_t * chunk )
```

Iterates over the available SFF chunks in the frame buffer

#### 4.13.2.4 dc1394\_basler\_sff\_chunk\_iterate\_init()

```
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

#### 4.13.2.5 dc1394\_basler\_sff\_feature\_enable()

```
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

#### 4.13.2.6 dc1394\_basler\_sff\_feature\_is\_available()

```
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

#### 4.13.2.7 dc1394\_basler\_sff\_feature\_is\_enabled()

```
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

#### 4.13.2.8 dc1394\_basler\_sff\_feature\_print()

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

prints info about one feature

#### 4.13.2.9 dc1394\_basler\_sff\_feature\_print\_all()

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

prints info about all features

#### 4.13.2.10 dc1394\_basler\_sff\_is\_available()

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

Tests whether the camera supports Basler SFF

## 4.14 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** {}

### 4.14.1 Detailed Description

No docs yet.

More details soon

### 4.14.2 Typedef Documentation

#### 4.14.2.1 dc1394basler\_dcam\_csr\_value\_t

```
typedef struct dc1394basler_dcam_csr_value_t dc1394basler_dcam_csr_value_t
```

No Docs

#### 4.14.2.2 dc1394basler\_dcam\_whitebalance\_csr\_value\_t

```
typedef struct dc1394basler_dcam_whitebalance_csr_value_t dc1394basler_dcam_whitebalance_csr_value_t↔
```

No Docs

#### 4.14.2.3 dc1394basler\_sff\_crc\_checksum\_t

```
typedef struct dc1394basler_sff_crc_checksum_t dc1394basler_sff_crc_checksum_t
```

No Docs

#### 4.14.2.4 dc1394basler\_sff\_cycle\_time\_stamp\_t

```
typedef struct dc1394basler_sff_cycle_time_stamp_t dc1394basler_sff_cycle_time_stamp_t
```

No Docs

#### 4.14.2.5 dc1394basler\_sff\_dcam\_values\_t

```
typedef struct dc1394basler_sff_dcam_values_t dc1394basler_sff_dcam_values_t
```

No Docs

#### 4.14.2.6 dc1394basler\_sff\_extended\_data\_stream\_t

```
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.

#### 4.14.2.7 dc1394basler\_sff\_feature\_t

```
typedef enum dc1394basler_sff_feature_t dc1394basler_sff_feature_t
```

SFF feature IDs

#### 4.14.2.8 dc1394basler\_sff\_frame\_counter\_t

```
typedef struct dc1394basler_sff_frame_counter_t dc1394basler_sff_frame_counter_t
```

No Docs

### 4.14.3 Enumeration Type Documentation

#### 4.14.3.1 dc1394basler\_sff\_feature\_t

```
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 <code>dc1394basler_sff_extended_data_stream</code> 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 <code>dc1394basler_sff_frame_counter_t</code> (p. 10) 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 <code>dc1394basler_sff_cycle_time_stamp</code> 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 <code>dc1394basler_sff_dcam_values_t</code> (p. 9) 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 <code>dc1394_basler_validate_checksum()</code> 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. <b>This feature is not implemented</b>
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. <b>This feature is not implemented</b>
DC1394_BASLER_SFF_LOOKUP_TABLE	<b>This feature is not implemented</b>
DC1394_BASLER_SFF_TRIGGER_FLAG_AND_↔ COUNTER	<b>This feature is not implemented</b>
DC1394_BASLER_SFF_OUTPUT_PORT_0_↔ CONFIGURATION	<b>This feature is not implemented</b>
DC1394_BASLER_SFF_OUTPUT_PORT_1_↔ CONFIGURATION	<b>This feature is not implemented</b>
DC1394_BASLER_SFF_OUTPUT_PORT_2_↔ CONFIGURATION	<b>This feature is not implemented</b>

## Enumerator

DC1394_BASLER_SFF_OUTPUT_PORT_3_↔ CONFIGURATION	This feature is not implemented
--	---------------------------------

## 4.15 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)`

#### 4.15.1 Detailed Description

No docs yet.

More details soon

#### 4.15.2 Function Documentation

##### 4.15.2.1 basler\_sff\_registry\_find\_by\_chunk\_guid()

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

Returns a sff feature descriptor by CHUNK guid

##### 4.15.2.2 basler\_sff\_registry\_find\_by\_csr\_guid()

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

Returns a sff feature descriptor by CSR guid

#### 4.15.2.3 basler\_sff\_registry\_find\_by\_id()

```
const sff_feature * basler_sff_registry_find_by_id (
    dc1394basler_sff_feature_t feature_id )
```

Returns a sff feature descriptor by id

## 4.16 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 \*)
- `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)

### 4.16.1 Detailed Description

No docs yet.

More details soon

### 4.16.2 Typedef Documentation

#### 4.16.2.1 dc1394\_pxl\_adv\_feature\_info\_t

```
typedef struct __dc1394_pxl_adv_feature_info_struct dc1394_pxl_adv_feature_info_t
```

Advanced feature inquiry

#### 4.16.2.2 dc1394\_pxl\_camera\_info\_t

```
typedef struct __dc1394_pxl_camera_info_struct dc1394_pxl_camera_info_t
```

Camera information

#### 4.16.2.3 dc1394\_pxl\_gpio\_info\_t

```
typedef struct __dc1394_pxl_gpio_info_struct dc1394_pxl_gpio_info_t
```

GPIO Information structure

### 4.16.3 Enumeration Type Documentation

#### 4.16.3.1 dc1394pxl\_gpio\_mode\_t

```
enum dc1394pxl_gpio_mode_t
```

No Docs

#### 4.16.3.2 dc1394pxl\_gpio\_polarity\_t

```
enum dc1394pxl_gpio_polarity_t
```

No Docs



## 4.16.4 Function Documentation

### 4.16.4.1 dc1394\_pxl\_convert\_float32\_to\_quadlet()

```
dc1394error_t dc1394_pxl_convert_float32_to_quadlet (
    double ,
    uint32_t * )
```

No Docs

### 4.16.4.2 dc1394\_pxl\_convert\_uint32\_to\_float32()

```
dc1394error_t dc1394_pxl_convert_uint32_to_float32 (
    uint32_t ,
    double * )
```

No Docs

### 4.16.4.3 dc1394\_pxl\_get\_camera\_info()

```
dc1394error_t dc1394_pxl_get_camera_info (
    dc1394camera_t * ,
    dc1394_pxl_camera_info_t * )
```

No Docs

### 4.16.4.4 dc1394\_pxl\_get\_camera\_name()

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

No Docs

### 4.16.4.5 dc1394\_pxl\_get\_camera\_serial\_number()

```
dc1394error_t dc1394_pxl_get_camera_serial_number (
    dc1394camera_t * ,
    uint32_t * )
```

No Docs

#### 4.16.4.6 dc1394\_pxl\_get\_gpo\_config()

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

No Docs

#### 4.16.4.7 dc1394\_pxl\_get\_gpo\_param()

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

No Docs

#### 4.16.4.8 dc1394\_pxl\_get\_gpo\_param\_min\_max()

```
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 * )
```

No Docs

#### 4.16.4.9 dc1394\_pxl\_print\_camera\_info()

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

No Docs

#### 4.16.4.10 dc1394\_pxl\_set\_gpio\_mode\_param()

```
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

## 4.16.4.11 dc1394\_pxl\_set\_gpo\_config()

```
dc1394error_t dc1394_pxl_set_gpo_config (
    dc1394camera_t * ,
    uint32_t ,
    uint32_t )
```

No Docs

## 4.17 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)`

### 4.17.1 Detailed Description

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

More details soon

### 4.17.2 Typedef Documentation

#### 4.17.2.1 dc1394video\_frame\_t

```
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.

### 4.17.3 Enumeration Type Documentation

#### 4.17.3.1 dc1394framerate\_t

```
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)

### 4.17.3.2 dc1394operation\_mode\_t

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.

### 4.17.3.3 dc1394speed\_t

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).

## 4.17.4 Function Documentation

### 4.17.4.1 dc1394\_video\_get\_bandwidth\_usage()

```
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.

### 4.17.4.2 dc1394\_video\_get\_data\_depth()

```
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,...)

### 4.17.4.3 dc1394\_video\_get\_framerate()

```
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.

#### 4.17.4.4 dc1394\_video\_get\_iso\_channel()

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

Gets the current ISO channel

#### 4.17.4.5 dc1394\_video\_get\_iso\_speed()

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

Gets the current ISO speed.

#### 4.17.4.6 dc1394\_video\_get\_mode()

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

Gets the current video mode.

#### 4.17.4.7 dc1394\_video\_get\_multi\_shot()

```
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.

#### 4.17.4.8 dc1394\_video\_get\_one\_shot()

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

Gets the status of the one-shot mode.

#### 4.17.4.9 dc1394\_video\_get\_operation\_mode()

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

Gets the current operation mode.

#### 4.17.4.10 dc1394\_video\_get\_supported\_framerates()

```
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.

#### 4.17.4.11 dc1394\_video\_get\_supported\_modes()

```
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.

#### 4.17.4.12 dc1394\_video\_get\_transmission()

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

Gets the status of the video transmission

#### 4.17.4.13 dc1394\_video\_set\_framerate()

```
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.

#### 4.17.4.14 dc1394\_video\_set\_iso\_channel()

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

Sets the current ISO channel

#### 4.17.4.15 dc1394\_video\_set\_iso\_speed()

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

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

#### 4.17.4.16 dc1394\_video\_set\_mode()

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

Sets the current video mode.

#### 4.17.4.17 dc1394\_video\_set\_multi\_shot()

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

Turns multishot mode on or off

#### 4.17.4.18 dc1394\_video\_set\_one\_shot()

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

Turns one-shot mode on or off

#### 4.17.4.19 dc1394\_video\_set\_operation\_mode()

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

Sets the current operation mode.

#### 4.17.4.20 dc1394\_video\_set\_transmission()

```
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

- \_\_dc1394\_camera, 5
- \_\_dc1394\_pxl\_adv\_feature\_info\_struct, 5
- \_\_dc1394\_pxl\_camera\_info\_struct, 5
- \_\_dc1394\_pxl\_gpio\_info\_struct, 6
- \_\_dc1394\_video\_frame, 6
- \_\_dc1394camera\_list\_t, 6
- \_\_dc1394feature\_info\_t\_struct, 7
- \_\_dc1394featureset\_t, 7
- \_\_dc1394format7mode\_t, 7
- \_\_dc1394format7modeset\_t, 8

## avt.h

- dc1394\_avt\_get\_advanced\_feature\_inquiry, 71
- dc1394\_avt\_get\_aoi, 71
- dc1394\_avt\_get\_auto\_gain, 72
- dc1394\_avt\_get\_auto\_shutter, 72
- dc1394\_avt\_get\_blemish, 72
- dc1394\_avt\_get\_channel\_adjust, 72
- dc1394\_avt\_get\_color\_corr, 72
- dc1394\_avt\_get\_deferred\_trans, 73
- dc1394\_avt\_get\_dsnu, 73
- dc1394\_avt\_get\_extented\_shutter, 73
- dc1394\_avt\_get\_frame\_info, 73
- dc1394\_avt\_get\_gpdata\_info, 73
- dc1394\_avt\_get\_hsnr, 74
- dc1394\_avt\_get\_io, 74
- dc1394\_avt\_get\_lut, 74
- dc1394\_avt\_get\_lut\_info, 74
- dc1394\_avt\_get\_lut\_mem\_ctrl, 74
- dc1394\_avt\_get\_MaxResolution, 75
- dc1394\_avt\_get\_mirror, 75
- dc1394\_avt\_get\_multiple\_slope, 75
- dc1394\_avt\_get\_shading, 75
- dc1394\_avt\_get\_shading\_info, 75
- dc1394\_avt\_get\_shading\_mem\_ctrl, 76
- dc1394\_avt\_get\_test\_images, 76
- dc1394\_avt\_get\_timebase, 76
- dc1394\_avt\_get\_trigger\_delay, 76
- dc1394\_avt\_get\_version, 76
- dc1394\_avt\_print\_advanced\_feature, 76
- dc1394\_avt\_read\_gpdata, 77
- dc1394\_avt\_read\_shading\_img, 77
- dc1394\_avt\_reset, 77
- dc1394\_avt\_reset\_frame\_info, 77
- dc1394\_avt\_set\_aoi, 77
- dc1394\_avt\_set\_auto\_gain, 77
- dc1394\_avt\_set\_auto\_shutter, 78
- dc1394\_avt\_set\_blemish, 78
- dc1394\_avt\_set\_channel\_adjust, 78
- dc1394\_avt\_set\_color\_corr, 78

- dc1394\_avt\_set\_deferred\_trans, 78
- dc1394\_avt\_set\_dsnu, 79
- dc1394\_avt\_set\_extented\_shutter, 79
- dc1394\_avt\_set\_hsnr, 79
- dc1394\_avt\_set\_io, 79
- dc1394\_avt\_set\_lut, 79
- dc1394\_avt\_set\_lut\_mem\_ctrl, 80
- dc1394\_avt\_set\_mirror, 80
- dc1394\_avt\_set\_multiple\_slope, 80
- dc1394\_avt\_set\_shading, 80
- dc1394\_avt\_set\_shading\_mem\_ctrl, 80
- dc1394\_avt\_set\_test\_images, 81
- dc1394\_avt\_set\_timebase, 81
- dc1394\_avt\_set\_trigger\_delay, 81
- dc1394\_avt\_write\_gpdata, 81
- dc1394\_avt\_write\_shading\_img, 81

## basler.h

- dc1394\_basler\_sff\_check\_crc, 82
- dc1394\_basler\_sff\_chunk\_find, 82
- dc1394\_basler\_sff\_chunk\_iterate, 83
- dc1394\_basler\_sff\_chunk\_iterate\_init, 83
- dc1394\_basler\_sff\_feature\_enable, 83
- dc1394\_basler\_sff\_feature\_is\_available, 83
- dc1394\_basler\_sff\_feature\_is\_enabled, 83
- dc1394\_basler\_sff\_feature\_print, 84
- dc1394\_basler\_sff\_feature\_print\_all, 84
- dc1394\_basler\_sff\_is\_available, 84

## basler\_sff.h

- DC1394\_BASLER\_SFF\_CRC\_CHECKSUM, 87
- DC1394\_BASLER\_SFF\_CYCLE\_TIME\_STAMP, 87
- DC1394\_BASLER\_SFF\_DCAM\_VALUES, 87
- DC1394\_BASLER\_SFF\_EXTENDED\_DATA\_STREAM, 87
- DC1394\_BASLER\_SFF\_EXTENDED\_VERSION\_INFO, 87
- DC1394\_BASLER\_SFF\_FRAME\_COUNTER, 87
- DC1394\_BASLER\_SFF\_LOOKUP\_TABLE, 87
- DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_0\_CONFIGURATION, 87
- DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_1\_CONFIGURATION, 87
- DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_2\_CONFIGURATION, 87
- DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_3\_CONFIGURATION, 88
- DC1394\_BASLER\_SFF\_TEST\_IMAGES, 87
- DC1394\_BASLER\_SFF\_TRIGGER\_FLAG\_AND\_COUNTER, 87

- dc1394basler\_dcam\_csr\_value\_t, 85
- dc1394basler\_dcam\_whitebalance\_csr\_value\_t, 85
- dc1394basler\_sff\_crc\_checksum\_t, 85
- dc1394basler\_sff\_cycle\_time\_stamp\_t, 85
- dc1394basler\_sff\_dcam\_values\_t, 86
- dc1394basler\_sff\_extended\_data\_stream\_t, 86
- dc1394basler\_sff\_feature\_t, 86
- dc1394basler\_sff\_frame\_counter\_t, 86
- basler\_sff\_registry.h
  - basler\_sff\_registry\_find\_by\_chunk\_guid, 88
  - basler\_sff\_registry\_find\_by\_csr\_guid, 88
  - basler\_sff\_registry\_find\_by\_id, 88
- basler\_sff\_registry\_find\_by\_chunk\_guid
  - basler\_sff\_registry.h, 88
- basler\_sff\_registry\_find\_by\_csr\_guid
  - basler\_sff\_registry.h, 88
- basler\_sff\_registry\_find\_by\_id
  - basler\_sff\_registry.h, 88
- camera.h
  - dc1394\_camera\_enumerate, 27
  - dc1394\_camera\_free, 27
  - dc1394\_camera\_free\_list, 27
  - dc1394\_camera\_get\_node, 27
  - dc1394\_camera\_get\_platform\_string, 27
  - dc1394\_camera\_new, 28
  - dc1394\_camera\_new\_unit, 28
  - dc1394\_camera\_print\_info, 28
  - dc1394\_camera\_set\_broadcast, 28
  - dc1394\_free, 28
  - dc1394\_new, 28
  - dc1394\_reset\_bus, 29
  - dc1394camera\_list\_t, 26
  - dc1394camera\_t, 26
  - dc1394iadc\_version\_t, 26
  - dc1394phy\_delay\_t, 26
  - dc1394power\_class\_t, 27
- capture.h
  - dc1394\_capture\_dequeue, 45
  - dc1394\_capture\_enqueue, 45
  - DC1394\_CAPTURE\_FLAGS\_CHANNEL\_ALLOC, 44
  - dc1394\_capture\_get\_fileno, 45
  - dc1394\_capture\_is\_frame\_corrupt, 46
  - dc1394\_capture\_set\_callback, 46
  - dc1394\_capture\_setup, 46
  - dc1394\_capture\_stop, 46
  - dc1394capture\_callback\_t, 45
  - dc1394capture\_policy\_t, 45
- control.h
  - dc1394\_camera\_reset, 50
  - dc1394\_camera\_set\_power, 50
  - dc1394\_external\_trigger\_get\_mode, 50
  - dc1394\_external\_trigger\_get\_polarity, 50
  - dc1394\_external\_trigger\_get\_power, 51
  - dc1394\_external\_trigger\_get\_source, 51
  - dc1394\_external\_trigger\_get\_supported\_sources, 51
  - dc1394\_external\_trigger\_has\_polarity, 51
  - dc1394\_external\_trigger\_set\_mode, 51
  - dc1394\_external\_trigger\_set\_polarity, 51
  - dc1394\_external\_trigger\_set\_power, 52
  - dc1394\_external\_trigger\_set\_source, 52
  - dc1394\_feature\_get, 52
  - dc1394\_feature\_get\_absolute\_boundaries, 52
  - dc1394\_feature\_get\_absolute\_control, 52
  - dc1394\_feature\_get\_absolute\_value, 52
  - dc1394\_feature\_get\_all, 53
  - dc1394\_feature\_get\_boundaries, 53
  - dc1394\_feature\_get\_mode, 53
  - dc1394\_feature\_get\_modes, 53
  - dc1394\_feature\_get\_power, 53
  - dc1394\_feature\_get\_value, 53
  - dc1394\_feature\_has\_absolute\_control, 54
  - dc1394\_feature\_is\_present, 54
  - dc1394\_feature\_is\_readable, 54
  - dc1394\_feature\_is\_switchable, 54
  - dc1394\_feature\_print, 54
  - dc1394\_feature\_print\_all, 54
  - dc1394\_feature\_set\_absolute\_control, 55
  - dc1394\_feature\_set\_absolute\_value, 55
  - dc1394\_feature\_set\_mode, 55
  - dc1394\_feature\_set\_power, 55
  - dc1394\_feature\_set\_value, 55
  - dc1394\_feature\_temperature\_get\_value, 55
  - dc1394\_feature\_temperature\_set\_value, 56
  - dc1394\_feature\_whitebalance\_get\_value, 56
  - dc1394\_feature\_whitebalance\_set\_value, 56
  - dc1394\_feature\_whiteshading\_get\_value, 56
  - dc1394\_feature\_whiteshading\_set\_value, 56
  - dc1394\_memory\_busy, 57
  - dc1394\_memory\_load, 57
  - dc1394\_memory\_save, 57
  - dc1394\_pio\_get, 57
  - dc1394\_pio\_set, 57
  - dc1394\_software\_trigger\_get\_power, 57
  - dc1394\_software\_trigger\_set\_power, 58
  - dc1394feature\_info\_t, 49
  - dc1394feature\_mode\_t, 49
  - dc1394feature\_t, 49
  - dc1394featureset\_t, 49
  - dc1394trigger\_mode\_t, 49
  - dc1394trigger\_polarity\_t, 50
  - dc1394trigger\_source\_t, 50
- conversions.h
  - dc1394\_bayer\_decoding\_16bit, 30
  - dc1394\_bayer\_decoding\_8bit, 30
  - dc1394\_convert\_frames, 31
  - dc1394\_convert\_to\_MONO8, 31
  - dc1394\_convert\_to\_RGB8, 31
  - dc1394\_convert\_to\_YUV422, 31
  - dc1394\_debayer\_frames, 32
  - dc1394\_deinterlace\_stereo, 32
  - dc1394\_deinterlace\_stereo\_frames, 32
  - dc1394bayer\_method\_t, 30
  - dc1394stereo\_method\_t, 30

dc1394/camera.h, 25  
dc1394/capture.h, 43  
dc1394/control.h, 46  
dc1394/conversions.h, 29  
dc1394/dc1394.h, 33  
dc1394/format7.h, 33  
dc1394/iso.h, 40  
dc1394/log.h, 58  
dc1394/register.h, 61  
dc1394/types.h, 65  
dc1394/utils.h, 66  
dc1394/vendor/avt.h, 69  
dc1394/vendor/basler.h, 82  
dc1394/vendor/basler\_sff.h, 84  
dc1394/vendor/basler\_sff\_registry.h, 88  
dc1394/vendor/pixelink.h, 89  
dc1394/video.h, 93  
dc1394\_avt\_get\_advanced\_feature\_inquiry  
    avt.h, 71  
dc1394\_avt\_get\_aoi  
    avt.h, 71  
dc1394\_avt\_get\_auto\_gain  
    avt.h, 72  
dc1394\_avt\_get\_auto\_shutter  
    avt.h, 72  
dc1394\_avt\_get\_blemish  
    avt.h, 72  
dc1394\_avt\_get\_channel\_adjust  
    avt.h, 72  
dc1394\_avt\_get\_color\_corr  
    avt.h, 72  
dc1394\_avt\_get\_deferred\_trans  
    avt.h, 73  
dc1394\_avt\_get\_dsnu  
    avt.h, 73  
dc1394\_avt\_get\_extented\_shutter  
    avt.h, 73  
dc1394\_avt\_get\_frame\_info  
    avt.h, 73  
dc1394\_avt\_get\_gpdata\_info  
    avt.h, 73  
dc1394\_avt\_get\_hsnr  
    avt.h, 74  
dc1394\_avt\_get\_io  
    avt.h, 74  
dc1394\_avt\_get\_lut  
    avt.h, 74  
dc1394\_avt\_get\_lut\_info  
    avt.h, 74  
dc1394\_avt\_get\_lut\_mem\_ctrl  
    avt.h, 74  
dc1394\_avt\_get\_MaxResolution  
    avt.h, 75  
dc1394\_avt\_get\_mirror  
    avt.h, 75  
dc1394\_avt\_get\_multiple\_slope  
    avt.h, 75  
dc1394\_avt\_get\_shading  
    avt.h, 75  
dc1394\_avt\_get\_shading\_info  
    avt.h, 75  
dc1394\_avt\_get\_shading\_mem\_ctrl  
    avt.h, 76  
dc1394\_avt\_get\_test\_images  
    avt.h, 76  
dc1394\_avt\_get\_timebase  
    avt.h, 76  
dc1394\_avt\_get\_trigger\_delay  
    avt.h, 76  
dc1394\_avt\_get\_version  
    avt.h, 76  
dc1394\_avt\_print\_advanced\_feature  
    avt.h, 76  
dc1394\_avt\_read\_gpdata  
    avt.h, 77  
dc1394\_avt\_read\_shading\_img  
    avt.h, 77  
dc1394\_avt\_reset  
    avt.h, 77  
dc1394\_avt\_reset\_frame\_info  
    avt.h, 77  
dc1394\_avt\_set\_aoi  
    avt.h, 77  
dc1394\_avt\_set\_auto\_gain  
    avt.h, 77  
dc1394\_avt\_set\_auto\_shutter  
    avt.h, 78  
dc1394\_avt\_set\_blemish  
    avt.h, 78  
dc1394\_avt\_set\_channel\_adjust  
    avt.h, 78  
dc1394\_avt\_set\_color\_corr  
    avt.h, 78  
dc1394\_avt\_set\_deferred\_trans  
    avt.h, 78  
dc1394\_avt\_set\_dsnu  
    avt.h, 79  
dc1394\_avt\_set\_extented\_shutter  
    avt.h, 79  
dc1394\_avt\_set\_hsnr  
    avt.h, 79  
dc1394\_avt\_set\_io  
    avt.h, 79  
dc1394\_avt\_set\_lut  
    avt.h, 79  
dc1394\_avt\_set\_lut\_mem\_ctrl  
    avt.h, 80  
dc1394\_avt\_set\_mirror  
    avt.h, 80  
dc1394\_avt\_set\_multiple\_slope  
    avt.h, 80  
dc1394\_avt\_set\_shading  
    avt.h, 80  
dc1394\_avt\_set\_shading\_mem\_ctrl  
    avt.h, 80  
dc1394\_avt\_set\_test\_images

avt.h, 81  
 dc1394\_avt\_set\_timebase  
   avt.h, 81  
 dc1394\_avt\_set\_trigger\_delay  
   avt.h, 81  
 dc1394\_avt\_write\_gpdata  
   avt.h, 81  
 dc1394\_avt\_write\_shading\_img  
   avt.h, 81  
 dc1394\_basler\_sff\_check\_crc  
   basler.h, 82  
 dc1394\_basler\_sff\_chunk\_find  
   basler.h, 82  
 dc1394\_basler\_sff\_chunk\_iterate  
   basler.h, 83  
 dc1394\_basler\_sff\_chunk\_iterate\_init  
   basler.h, 83  
 DC1394\_BASLER\_SFF\_CRC\_CHECKSUM  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_CYCLE\_TIME\_STAMP  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_DCAM\_VALUES  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_EXTENDED\_DATA\_STREAM  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_EXTENDED\_VERSION\_INFO  
   basler\_sff.h, 87  
 dc1394\_basler\_sff\_feature\_enable  
   basler.h, 83  
 dc1394\_basler\_sff\_feature\_is\_available  
   basler.h, 83  
 dc1394\_basler\_sff\_feature\_is\_enabled  
   basler.h, 83  
 dc1394\_basler\_sff\_feature\_print  
   basler.h, 84  
 dc1394\_basler\_sff\_feature\_print\_all  
   basler.h, 84  
 DC1394\_BASLER\_SFF\_FRAME\_COUNTER  
   basler\_sff.h, 87  
 dc1394\_basler\_sff\_is\_available  
   basler.h, 84  
 DC1394\_BASLER\_SFF\_LOOKUP\_TABLE  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_0\_CONFIGURATION  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_1\_CONFIGURATION  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_2\_CONFIGURATION  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_OUTPUT\_PORT\_3\_CONFIGURATION  
   basler\_sff.h, 88  
 DC1394\_BASLER\_SFF\_TEST\_IMAGES  
   basler\_sff.h, 87  
 DC1394\_BASLER\_SFF\_TRIGGER\_FLAG\_AND\_COUNTER  
   basler\_sff.h, 87  
 dc1394\_bayer\_decoding\_16bit  
   conversions.h, 30  
 dc1394\_bayer\_decoding\_8bit  
   conversions.h, 30  
 dc1394\_camera\_enumerate  
   camera.h, 27  
 dc1394\_camera\_free  
   camera.h, 27  
 dc1394\_camera\_free\_list  
   camera.h, 27  
 dc1394\_camera\_get\_node  
   camera.h, 27  
 dc1394\_camera\_get\_platform\_string  
   camera.h, 27  
 dc1394\_camera\_new  
   camera.h, 28  
 dc1394\_camera\_new\_unit  
   camera.h, 28  
 dc1394\_camera\_print\_info  
   camera.h, 28  
 dc1394\_camera\_reset  
   control.h, 50  
 dc1394\_camera\_set\_broadcast  
   camera.h, 28  
 dc1394\_camera\_set\_power  
   control.h, 50  
 dc1394\_capture\_dequeue  
   capture.h, 45  
 dc1394\_capture\_enqueue  
   capture.h, 45  
 DC1394\_CAPTURE\_FLAGS\_CHANNEL\_ALLOC  
   capture.h, 44  
 dc1394\_capture\_get\_fileno  
   capture.h, 45  
 dc1394\_capture\_is\_frame\_corrupt  
   capture.h, 46  
 dc1394\_capture\_set\_callback  
   capture.h, 46  
 dc1394\_capture\_setup  
   capture.h, 46  
 dc1394\_capture\_stop  
   capture.h, 46  
 dc1394\_checksum\_crc16  
   utils.h, 67  
 dc1394\_convert\_frames  
   conversions.h, 31  
 dc1394\_convert\_to\_MONO8  
   conversions.h, 31  
 dc1394\_convert\_to\_RGB8  
   conversions.h, 31  
 dc1394\_convert\_to\_YUV422  
   conversions.h, 31  
 dc1394\_debayer\_frames  
   conversions.h, 32  
 dc1394\_deinterlace\_stereo  
   conversions.h, 32  
 dc1394\_deinterlace\_stereo\_frames  
   conversions.h, 32  
 dc1394\_error\_get\_string  
   utils.h, 67  
 dc1394\_external\_trigger\_get\_mode

- control.h, 50
- dc1394\_external\_trigger\_get\_polarity
  - control.h, 50
- dc1394\_external\_trigger\_get\_power
  - control.h, 51
- dc1394\_external\_trigger\_get\_source
  - control.h, 51
- dc1394\_external\_trigger\_get\_supported\_sources
  - control.h, 51
- dc1394\_external\_trigger\_has\_polarity
  - control.h, 51
- dc1394\_external\_trigger\_set\_mode
  - control.h, 51
- dc1394\_external\_trigger\_set\_polarity
  - control.h, 51
- dc1394\_external\_trigger\_set\_power
  - control.h, 52
- dc1394\_external\_trigger\_set\_source
  - control.h, 52
- dc1394\_feature\_get
  - control.h, 52
- dc1394\_feature\_get\_absolute\_boundaries
  - control.h, 52
- dc1394\_feature\_get\_absolute\_control
  - control.h, 52
- dc1394\_feature\_get\_absolute\_value
  - control.h, 52
- dc1394\_feature\_get\_all
  - control.h, 53
- dc1394\_feature\_get\_boundaries
  - control.h, 53
- dc1394\_feature\_get\_mode
  - control.h, 53
- dc1394\_feature\_get\_modes
  - control.h, 53
- dc1394\_feature\_get\_power
  - control.h, 53
- dc1394\_feature\_get\_string
  - utils.h, 67
- dc1394\_feature\_get\_value
  - control.h, 53
- dc1394\_feature\_has\_absolute\_control
  - control.h, 54
- dc1394\_feature\_is\_present
  - control.h, 54
- dc1394\_feature\_is\_readable
  - control.h, 54
- dc1394\_feature\_is\_switchable
  - control.h, 54
- dc1394\_feature\_print
  - control.h, 54
- dc1394\_feature\_print\_all
  - control.h, 54
- dc1394\_feature\_set\_absolute\_control
  - control.h, 55
- dc1394\_feature\_set\_absolute\_value
  - control.h, 55
- dc1394\_feature\_set\_mode
  - control.h, 55
- dc1394\_feature\_set\_power
  - control.h, 55
- dc1394\_feature\_set\_value
  - control.h, 55
- dc1394\_feature\_temperature\_get\_value
  - control.h, 55
- dc1394\_feature\_temperature\_set\_value
  - control.h, 56
- dc1394\_feature\_whitebalance\_get\_value
  - control.h, 56
- dc1394\_feature\_whitebalance\_set\_value
  - control.h, 56
- dc1394\_feature\_whiteshading\_get\_value
  - control.h, 56
- dc1394\_feature\_whiteshading\_set\_value
  - control.h, 56
- dc1394\_format7\_get\_color\_coding
  - format7.h, 35
- dc1394\_format7\_get\_color\_codings
  - format7.h, 35
- dc1394\_format7\_get\_color\_filter
  - format7.h, 35
- dc1394\_format7\_get\_data\_depth
  - format7.h, 35
- dc1394\_format7\_get\_frame\_interval
  - format7.h, 36
- dc1394\_format7\_get\_image\_position
  - format7.h, 36
- dc1394\_format7\_get\_image\_size
  - format7.h, 36
- dc1394\_format7\_get\_max\_image\_size
  - format7.h, 36
- dc1394\_format7\_get\_mode\_info
  - format7.h, 36
- dc1394\_format7\_get\_modeset
  - format7.h, 37
- dc1394\_format7\_get\_packet\_parameters
  - format7.h, 37
- dc1394\_format7\_get\_packet\_size
  - format7.h, 37
- dc1394\_format7\_get\_packets\_per\_frame
  - format7.h, 37
- dc1394\_format7\_get\_pixel\_number
  - format7.h, 37
- dc1394\_format7\_get\_recommended\_packet\_size
  - format7.h, 38
- dc1394\_format7\_get\_roi
  - format7.h, 38
- dc1394\_format7\_get\_total\_bytes
  - format7.h, 38
- dc1394\_format7\_get\_unit\_position
  - format7.h, 38
- dc1394\_format7\_get\_unit\_size
  - format7.h, 38
- dc1394\_format7\_set\_color\_coding
  - format7.h, 39
- dc1394\_format7\_set\_image\_position

- format7.h, 39
- dc1394\_format7\_set\_image\_size
  - format7.h, 39
- dc1394\_format7\_set\_packet\_size
  - format7.h, 39
- dc1394\_format7\_set\_roi
  - format7.h, 39
- dc1394\_framerate\_as\_float
  - utils.h, 68
- dc1394\_free
  - camera.h, 28
- dc1394\_get\_absolute\_register
  - register.h, 62
- dc1394\_get\_adv\_control\_registers
  - register.h, 62
- dc1394\_get\_color\_coding\_bit\_size
  - utils.h, 68
- dc1394\_get\_color\_coding\_data\_depth
  - utils.h, 68
- dc1394\_get\_color\_coding\_from\_video\_mode
  - utils.h, 68
- dc1394\_get\_control\_registers
  - register.h, 62
- dc1394\_get\_format7\_register
  - register.h, 62
- dc1394\_get\_image\_size\_from\_video\_mode
  - utils.h, 68
- dc1394\_get\_PIO\_register
  - register.h, 62
- dc1394\_get\_registers
  - register.h, 62
- dc1394\_get\_SIO\_register
  - register.h, 63
- dc1394\_get\_strobe\_register
  - register.h, 63
- dc1394\_is\_color
  - utils.h, 68
- dc1394\_is\_same\_camera
  - utils.h, 69
- dc1394\_is\_video\_mode\_scalable
  - utils.h, 69
- dc1394\_is\_video\_mode\_still\_image
  - utils.h, 69
- dc1394\_iso\_allocate\_bandwidth
  - iso.h, 40
- dc1394\_iso\_allocate\_channel
  - iso.h, 41
- dc1394\_iso\_release\_all
  - iso.h, 41
- dc1394\_iso\_release\_bandwidth
  - iso.h, 42
- dc1394\_iso\_release\_channel
  - iso.h, 42
- dc1394\_iso\_set\_persist
  - iso.h, 43
- dc1394\_log\_debug
  - log.h, 59
- dc1394\_log\_error
  - log.h, 59
- dc1394\_log\_register\_handler
  - log.h, 60
- dc1394\_log\_set\_default\_handler
  - log.h, 60
- dc1394\_log\_warning
  - log.h, 60
- dc1394\_memory\_busy
  - control.h, 57
- dc1394\_memory\_load
  - control.h, 57
- dc1394\_memory\_save
  - control.h, 57
- dc1394\_new
  - camera.h, 28
- dc1394\_pio\_get
  - control.h, 57
- dc1394\_pio\_set
  - control.h, 57
- dc1394\_pxl\_adv\_feature\_info\_t
  - pixelink.h, 90
- dc1394\_pxl\_camera\_info\_t
  - pixelink.h, 90
- dc1394\_pxl\_convert\_float32\_to\_quadlet
  - pixelink.h, 91
- dc1394\_pxl\_convert\_uint32\_to\_float32
  - pixelink.h, 91
- dc1394\_pxl\_get\_camera\_info
  - pixelink.h, 91
- dc1394\_pxl\_get\_camera\_name
  - pixelink.h, 91
- dc1394\_pxl\_get\_camera\_serial\_number
  - pixelink.h, 91
- dc1394\_pxl\_get\_gpo\_config
  - pixelink.h, 91
- dc1394\_pxl\_get\_gpo\_param
  - pixelink.h, 92
- dc1394\_pxl\_get\_gpo\_param\_min\_max
  - pixelink.h, 92
- dc1394\_pxl\_gpio\_info\_t
  - pixelink.h, 90
- dc1394\_pxl\_print\_camera\_info
  - pixelink.h, 92
- dc1394\_pxl\_set\_gpio\_mode\_param
  - pixelink.h, 92
- dc1394\_pxl\_set\_gpo\_config
  - pixelink.h, 92
- dc1394\_reset\_bus
  - camera.h, 29
- dc1394\_set\_absolute\_register
  - register.h, 63
- dc1394\_set\_adv\_control\_registers
  - register.h, 63
- dc1394\_set\_control\_registers
  - register.h, 63
- dc1394\_set\_format7\_register
  - register.h, 64
- dc1394\_set\_PIO\_register

register.h, 64  
dc1394\_set\_registers  
  register.h, 64  
dc1394\_set\_SIO\_register  
  register.h, 64  
dc1394\_set\_strobe\_register  
  register.h, 64  
dc1394\_software\_trigger\_get\_power  
  control.h, 57  
dc1394\_software\_trigger\_set\_power  
  control.h, 58  
dc1394\_video\_get\_bandwidth\_usage  
  video.h, 95  
dc1394\_video\_get\_data\_depth  
  video.h, 95  
dc1394\_video\_get\_framerate  
  video.h, 95  
dc1394\_video\_get\_iso\_channel  
  video.h, 95  
dc1394\_video\_get\_iso\_speed  
  video.h, 96  
dc1394\_video\_get\_mode  
  video.h, 96  
dc1394\_video\_get\_multi\_shot  
  video.h, 96  
dc1394\_video\_get\_one\_shot  
  video.h, 96  
dc1394\_video\_get\_operation\_mode  
  video.h, 96  
dc1394\_video\_get\_supported\_framerates  
  video.h, 96  
dc1394\_video\_get\_supported\_modes  
  video.h, 97  
dc1394\_video\_get\_transmission  
  video.h, 97  
dc1394\_video\_set\_framerate  
  video.h, 97  
dc1394\_video\_set\_iso\_channel  
  video.h, 97  
dc1394\_video\_set\_iso\_speed  
  video.h, 97  
dc1394\_video\_set\_mode  
  video.h, 97  
dc1394\_video\_set\_multi\_shot  
  video.h, 98  
dc1394\_video\_set\_one\_shot  
  video.h, 98  
dc1394\_video\_set\_operation\_mode  
  video.h, 98  
dc1394\_video\_set\_transmission  
  video.h, 98  
dc1394basler\_dcam\_csr\_value\_t, 8  
  basler\_sff.h, 85  
dc1394basler\_dcam\_whitebalance\_csr\_value\_t, 8  
  basler\_sff.h, 85  
dc1394basler\_sff\_chunk\_tail\_t, 9  
dc1394basler\_sff\_crc\_checksum\_t, 9  
  basler\_sff.h, 85  
dc1394basler\_sff\_cycle\_time\_stamp\_t, 9  
  basler\_sff.h, 85  
dc1394basler\_sff\_dcam\_values\_t, 9  
  basler\_sff.h, 86  
dc1394basler\_sff\_extended\_data\_stream\_t, 10  
  basler\_sff.h, 86  
dc1394basler\_sff\_feature\_t  
  basler\_sff.h, 86  
dc1394basler\_sff\_frame\_counter\_t, 10  
  basler\_sff.h, 86  
dc1394basler\_sff\_guid\_t, 10  
dc1394basler\_sff\_t, 11  
dc1394bayer\_method\_t  
  conversions.h, 30  
dc1394bool\_t  
  types.h, 65  
dc1394byte\_order\_t  
  types.h, 65  
dc1394camera\_id\_t, 11  
dc1394camera\_list\_t  
  camera.h, 26  
dc1394camera\_t  
  camera.h, 26  
dc1394capture\_callback\_t  
  capture.h, 45  
dc1394capture\_policy\_t  
  capture.h, 45  
dc1394color\_coding\_t  
  types.h, 66  
dc1394color\_codings\_t, 11  
dc1394color\_filter\_t  
  types.h, 66  
dc1394error\_t  
  log.h, 59  
dc1394feature\_info\_t  
  control.h, 49  
dc1394feature\_mode\_t  
  control.h, 49  
dc1394feature\_modes\_t, 12  
dc1394feature\_t  
  control.h, 49  
dc1394featureset\_t  
  control.h, 49  
dc1394format7mode\_t  
  format7.h, 35  
dc1394format7modeset\_t  
  format7.h, 35  
dc1394framerate\_t  
  video.h, 94  
dc1394framerates\_t, 12  
dc1394iidx\_version\_t  
  camera.h, 26  
dc1394log\_t  
  log.h, 59  
dc1394operation\_mode\_t  
  video.h, 94  
dc1394phy\_delay\_t  
  camera.h, 26

- dc1394power\_class\_t
  - camera.h, 27
- dc1394pxl\_gpio\_mode\_t
  - pixelink.h, 90
- dc1394pxl\_gpio\_polarity\_t
  - pixelink.h, 90
- dc1394speed\_t
  - video.h, 95
- dc1394stereo\_method\_t
  - conversions.h, 30
- dc1394switch\_t
  - types.h, 66
- dc1394trigger\_mode\_t
  - control.h, 49
- dc1394trigger\_modes\_t, 12
- dc1394trigger\_polarity\_t
  - control.h, 50
- dc1394trigger\_source\_t
  - control.h, 50
- dc1394trigger\_sources\_t, 13
- dc1394video\_frame\_t
  - video.h, 94
- dc1394video\_mode\_t
  - types.h, 66
- dc1394video\_modes\_t, 13
- format7.h
  - dc1394\_format7\_get\_color\_coding, 35
  - dc1394\_format7\_get\_color\_codings, 35
  - dc1394\_format7\_get\_color\_filter, 35
  - dc1394\_format7\_get\_data\_depth, 35
  - dc1394\_format7\_get\_frame\_interval, 36
  - dc1394\_format7\_get\_image\_position, 36
  - dc1394\_format7\_get\_image\_size, 36
  - dc1394\_format7\_get\_max\_image\_size, 36
  - dc1394\_format7\_get\_mode\_info, 36
  - dc1394\_format7\_get\_modeset, 37
  - dc1394\_format7\_get\_packet\_parameters, 37
  - dc1394\_format7\_get\_packet\_size, 37
  - dc1394\_format7\_get\_packets\_per\_frame, 37
  - dc1394\_format7\_get\_pixel\_number, 37
  - dc1394\_format7\_get\_recommended\_packet\_size, 38
  - dc1394\_format7\_get\_roi, 38
  - dc1394\_format7\_get\_total\_bytes, 38
  - dc1394\_format7\_get\_unit\_position, 38
  - dc1394\_format7\_get\_unit\_size, 38
  - dc1394\_format7\_set\_color\_coding, 39
  - dc1394\_format7\_set\_image\_position, 39
  - dc1394\_format7\_set\_image\_size, 39
  - dc1394\_format7\_set\_packet\_size, 39
  - dc1394\_format7\_set\_roi, 39
  - dc1394format7mode\_t, 35
  - dc1394format7modeset\_t, 35
- fw\_cdev\_add\_descriptor, 13
- fw\_cdev\_allocate, 14
- fw\_cdev\_allocate\_iso\_resource, 14
- fw\_cdev\_create\_iso\_context, 15
- fw\_cdev\_deallocate, 15
- fw\_cdev\_event, 16
- fw\_cdev\_event\_bus\_reset, 16
- fw\_cdev\_event\_common, 17
- fw\_cdev\_event\_iso\_interrupt, 17
- fw\_cdev\_event\_iso\_resource, 18
- fw\_cdev\_event\_request, 18
- fw\_cdev\_event\_response, 19
- fw\_cdev\_get\_cycle\_timer, 19
- fw\_cdev\_get\_cycle\_timer2, 19
- fw\_cdev\_get\_info, 20
- fw\_cdev\_initiate\_bus\_reset, 20
- fw\_cdev\_iso\_packet, 21
- fw\_cdev\_queue\_iso, 21
- fw\_cdev\_remove\_descriptor, 22
- fw\_cdev\_send\_request, 22
- fw\_cdev\_send\_response, 22
- fw\_cdev\_send\_stream\_packet, 23
- fw\_cdev\_start\_iso, 23
- fw\_cdev\_stop\_iso, 24
- iso.h
  - dc1394\_iso\_allocate\_bandwidth, 40
  - dc1394\_iso\_allocate\_channel, 41
  - dc1394\_iso\_release\_all, 41
  - dc1394\_iso\_release\_bandwidth, 42
  - dc1394\_iso\_release\_channel, 42
  - dc1394\_iso\_set\_persist, 43
- log.h
  - dc1394\_log\_debug, 59
  - dc1394\_log\_error, 59
  - dc1394\_log\_register\_handler, 60
  - dc1394\_log\_set\_default\_handler, 60
  - dc1394\_log\_warning, 60
  - dc1394error\_t, 59
  - dc1394log\_t, 59
- pixelink.h
  - dc1394\_pxl\_adv\_feature\_info\_t, 90
  - dc1394\_pxl\_camera\_info\_t, 90
  - dc1394\_pxl\_convert\_float32\_to\_quadlet, 91
  - dc1394\_pxl\_convert\_uint32\_to\_float32, 91
  - dc1394\_pxl\_get\_camera\_info, 91
  - dc1394\_pxl\_get\_camera\_name, 91
  - dc1394\_pxl\_get\_camera\_serial\_number, 91
  - dc1394\_pxl\_get\_gpo\_config, 91
  - dc1394\_pxl\_get\_gpo\_param, 92
  - dc1394\_pxl\_get\_gpo\_param\_min\_max, 92
  - dc1394\_pxl\_gpio\_info\_t, 90
  - dc1394\_pxl\_print\_camera\_info, 92
  - dc1394\_pxl\_set\_gpio\_mode\_param, 92
  - dc1394\_pxl\_set\_gpo\_config, 92
  - dc1394pxl\_gpio\_mode\_t, 90
  - dc1394pxl\_gpio\_polarity\_t, 90
- register.h
  - dc1394\_get\_absolute\_register, 62
  - dc1394\_get\_adv\_control\_registers, 62
  - dc1394\_get\_control\_registers, 62



- dc1394\_get\_format7\_register, 62
- dc1394\_get\_PIO\_register, 62
- dc1394\_get\_registers, 62
- dc1394\_get\_SIO\_register, 63
- dc1394\_get\_strobe\_register, 63
- dc1394\_set\_absolute\_register, 63
- dc1394\_set\_adv\_control\_registers, 63
- dc1394\_set\_control\_registers, 63
- dc1394\_set\_format7\_register, 64
- dc1394\_set\_PIO\_register, 64
- dc1394\_set\_registers, 64
- dc1394\_set\_SIO\_register, 64
- dc1394\_set\_strobe\_register, 64
- dc1394speed\_t, 95
- dc1394video\_frame\_t, 94

#### types.h

- dc1394bool\_t, 65
- dc1394byte\_order\_t, 65
- dc1394color\_coding\_t, 66
- dc1394color\_filter\_t, 66
- dc1394switch\_t, 66
- dc1394video\_mode\_t, 66

#### utils.h

- dc1394\_checksum\_crc16, 67
- dc1394\_error\_get\_string, 67
- dc1394\_feature\_get\_string, 67
- dc1394\_framerate\_as\_float, 68
- dc1394\_get\_color\_coding\_bit\_size, 68
- dc1394\_get\_color\_coding\_data\_depth, 68
- dc1394\_get\_color\_coding\_from\_video\_mode, 68
- dc1394\_get\_image\_size\_from\_video\_mode, 68
- dc1394\_is\_color, 68
- dc1394\_is\_same\_camera, 69
- dc1394\_is\_video\_mode\_scalable, 69
- dc1394\_is\_video\_mode\_still\_image, 69

#### video.h

- dc1394\_video\_get\_bandwidth\_usage, 95
- dc1394\_video\_get\_data\_depth, 95
- dc1394\_video\_get\_framerate, 95
- dc1394\_video\_get\_iso\_channel, 95
- dc1394\_video\_get\_iso\_speed, 96
- dc1394\_video\_get\_mode, 96
- dc1394\_video\_get\_multi\_shot, 96
- dc1394\_video\_get\_one\_shot, 96
- dc1394\_video\_get\_operation\_mode, 96
- dc1394\_video\_get\_supported\_framerates, 96
- dc1394\_video\_get\_supported\_modes, 97
- dc1394\_video\_get\_transmission, 97
- dc1394\_video\_set\_framerate, 97
- dc1394\_video\_set\_iso\_channel, 97
- dc1394\_video\_set\_iso\_speed, 97
- dc1394\_video\_set\_mode, 97
- dc1394\_video\_set\_multi\_shot, 98
- dc1394\_video\_set\_one\_shot, 98
- dc1394\_video\_set\_operation\_mode, 98
- dc1394\_video\_set\_transmission, 98
- dc1394framerate\_t, 94
- dc1394operation\_mode\_t, 94