libcin

Generated by Doxygen 1.8.6

Thu Jul 6 2017 11:30:19

Contents

1	Mod	ule Inde	ex													1
	1.1	Module	es						 	 	 	 	 	 		1
2	Clas	s Index														3
	2.1	Class I	_ist						 	 	 	 	 	 		3
3	File I	Index														5
	3.1	File Lis	st						 	 	 	 	 	 		5
4	Mod	ule Doc	umentatio	on												7
	4.1	Cin Co	ntrol Routi	ines .					 	 	 	 	 	 		7
		4.1.1	Detailed	Descrip	tion .				 	 	 	 	 	 		8
		4.1.2	Function	Docum	entatio	on			 	 	 	 	 	 		8
			4.1.2.1	cin_ct	l_destr	оу			 	 	 	 	 	 		8
			4.1.2.2	cin_ct	l_init .				 	 	 	 	 	 		8
			4.1.2.3	cin_ct	l_read				 	 	 	 	 	 		9
			4.1.2.4	cin_ct	l_strea	ım_wri	ite		 	 	 	 	 	 		10
			4.1.2.5	cin_ct	l_write				 	 	 	 	 	 		10
			4.1.2.6	cin_ct	l_write	_with_	_readb	ack	 	 	 	 	 	 		10
	4.2	CIN Da	ata Routine	es					 	 	 	 	 	 		11
5	Clas	s Docu	mentation	1												13
	5.1	cin_ctl	Struct Ref	erence					 	 	 	 	 	 		13
	5.2	cin_ctl	_config Str	uct Ref	erence				 	 	 	 	 	 		13
	5.3	cin_ctl	_id Struct I	Referen	ice .				 	 	 	 	 	 		14
	5.4	cin_ctl_	_listener S	truct Re	eferenc	е			 	 	 	 	 	 		14
	5.5	cin_ctl_	_pwr_mon	_t Struc	t Refe	rence			 	 	 	 	 	 		14
	5.6	cin_ctl_	_pwr_val S	Struct R	eferen	ce			 	 	 	 	 	 		14
	5.7	cin_da	ta Struct R	Referenc	е				 	 	 	 	 	 		15
	5.8	cin_da	ta_callbac	ks Struc	ct Refe	rence			 	 	 	 	 	 		15
	5.9	cin_da	ta_frame S	Struct R	eferen	ce			 	 	 	 	 	 		15
	5.10	cin_da	ta_packet	Struct F	Referer	nce .			 	 	 	 	 	 		16
	5 11	cin da	ta proc St	ruct Re	ferenci	Δ										16

iv CONTENTS

	5.12	cin_dat	ta_stats St	ruct Refer	ence .		 	 	 	 		 	 	 16
	5.13	cin_dat	ta_threads	Struct Re	ference		 	 	 	 		 		 17
	5.14	cin_ma	p_t Struct	Reference	e		 	 	 	 		 	 	 17
	5.15	cin_po	rt Struct R	eference			 	 	 	 		 	 	 17
	5.16	descra	mble_map	_t Struct F	Referenc	ce	 	 	 	 		 	 	 18
	5.17	fifo Stru	uct Refere	nce			 	 	 	 		 	 	 18
6	File I	Docume	entation											19
	6.1	src/cin.	h File Ref	erence .			 	 	 	 		 		 19
		6.1.1	Detailed	Descriptio	n		 	 	 	 		 	 	 24
		6.1.2	LICENSE				 	 	 	 		 	 	 24
		6.1.3	DESCRI	PTION .			 	 	 	 		 	 	 24
		6.1.4	Function	Documen	tation .		 	 	 	 		 	 	 24
			6.1.4.1	cin_data	_init		 	 	 	 		 	 	 24
Inc	lex													26

Module Index

_	_	_		_	
4	4		١л	ᆔ	 PS
		- 1	VIC	11	 76

lere is a list of all modules:		
		 1

2 **Module Index**

Class Index

2.1 Class List

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

cin_ctl	3
cin_ctl_config	13
cin_ctl_id	14
cin_ctl_listener	14
cin_ctl_pwr_mon_t	14
<u></u>	14
cin_data	15
	15
cin_data_frame	15
	16
cin_data_proc	16
	16
cin_data_threads	17
	17
<u> </u>	17
descramble_map_t	8
fifo	18

Class Index

File Index

3.1 File List

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

src/cin.h																					
src/cin_register_map.h																					
src/cinregisters.h					 																??
src/common.h					 																??
src/config.h					 																??
src/control.h																					
src/data.h					 																??
src/descramble.h																					
src/descramble_map.h																					
src/fclk_program.h																					
src/fifo.h																					
src/report.h					 																??
src/version.h					 																??

6 File Index

Module Documentation

4.1 Cin Control Routines

Functions

```
int cin_ctl_init (cin_ctl_t *cin, const char *ipaddr, uint16_t oport, uint16_t iport, uint16_t siport)
int cin_ctl_destroy (cin_ctl_t *cin)
int cin_ctl_read (cin_ctl_t *cin, uint16_t reg, uint16_t *val)
int cin_ctl_write (cin_ctl_t *cin, uint16_t reg, uint16_t val, int wait)
int cin_ctl_stream_write (cin_ctl_t *cin, char *val, int size)
int cin_ctl_write_with_readback (cin_ctl_t *cin, uint16_t reg, uint16_t val)
```

- int cin_ctl_pwr (cin_ctl_t *cin, int pwr)
- int cin_ctl_fp_pwr (cin_ctl_t *cin, int pwr)
- int cin ctl fo test pattern (cin ctl t *cin, int on off)
- int cin_ctl_load_config (cin_ctl_t *cin, char *filename)
- int cin ctl load firmware (cin ctl t *cin, char *filename)
- int cin_ctl_set_fclk (cin_ctl_t *cin, int clkfreq)
- int cin_ctl_get_fclk (cin_ctl_t *cin, int *clkfreq)
- int cin ctl freeze dco (cin ctl t *cin, int freeze)
- int cin_ctl_get_cfg_fpga_status (cin_ctl_t *cin, uint16_t *_val)
- int cin_ctl_get_id (cin_ctl_t *cin, cin_ctl_id_t *_val)
- void cin_ctl_display_id (FILE *out, cin_ctl_id_t val)
- void cin_ctl_display_fpga_status (FILE *out, uint16_t val)
- int cin_ctl_get_dcm_status (cin_ctl_t *cin, uint16_t *_val)
- void cin_ctl_display_dcm_status (FILE *out, uint16_t *_val)
- double cin ctl current calc (uint16 t val)
- int cin_ctl_get_power_status (cin_ctl_t *cin, int full, int *pwr, cin_ctl_pwr_mon_t *values)
- void cin_ctl_display_pwr (FILE *out, cin_ctl_pwr_mon_t *values)
- void cin_ctl_display_pwr_line (FILE *out, const char *msg, cin_ctl_pwr_val_t val)
- int cin_ctl_calc_vi_status (cin_ctl_t *cin, uint16_t vreg, uint16_t ireg, double vfact, cin_ctl_pwr_val_t *vi)
- int cin ctl get camera pwr (cin ctl t *cin, int *val)
- int cin_ctl_set_camera_pwr (cin_ctl_t *cin, int val)
- int cin ctl set bias (cin ctl t *cin, int val)
- int cin_ctl_get_bias (cin_ctl_t *cin, int *val)
- int cin_ctl_set_clocks (cin_ctl_t *cin, int val)
- int cin ctl get clocks (cin ctl t *cin, int *val)
- int cin_ctl_set_trigger (cin_ctl_t *cin, int val)
- int cin_ctl_get_trigger (cin_ctl_t *cin, int *val)
- int cin_ctl_set_focus (cin_ctl_t *cin, int val)

8 Module Documentation

- int cin_ctl_get_focus (cin_ctl_t *cin, int *val)
- int cin_ctl_get_triggering (cin_ctl_t *cin, int *trigger)
- int cin_ctl_int_trigger_start (cin_ctl_t *cin, int nimages)
- int cin_ctl_int_trigger_stop (cin_ctl_t *cin)
- int cin_ctl_ext_trigger_start (cin_ctl_t *cin, int trigger_mode)
- int cin ctl ext trigger stop (cin ctl t *cin)
- int cin_ctl_set_exposure_time (cin_ctl_t *cin, float e_time)
- int cin_ctl_set_trigger_delay (cin_ctl_t *cin, float t_time)
- int cin ctl set cycle time (cin ctl t *cin, float ftime)
- int cin_ctl_frame_count_reset (cin_ctl_t *cin)
- int cin_ctl_set_mux (cin_ctl_t *cin, int setting)
- int cin_ctl_get_mux (cin_ctl_t *cin, int *setting)
- int cin_ctl_set_fcric_gain (cin_ctl_t *cin, int gain)
- int cin ctl set fabric address (cin ctl t *cin, char *ip)
- int cin_ctl_reg_dump (cin_ctl_t *cin, FILE *fp)
- int cin_ctl_get_bias_voltages (cin_ctl_t *cin, float *voltage)
- int cin_ctl_set_bias_voltages (cin_ctl_t *cin, float *voltage)
- int cin_ctl_set_fcric_clamp (cin_ctl_t *cin, int clamp)

4.1.1 Detailed Description

4.1.2 Function Documentation

4.1.2.1 int cin_ctl_destroy (cin_ctl_t * cin)

Destroy (close) the cin control library

Close connections, free memory and exit library

Parameters

	cin	handle to cin library
--	-----	-----------------------

Returns

Returns 0 on sucsess non-zero if error

4.1.2.2 int cin ctl init (cin ctl t * cin, const char * ipaddr, uint16 t oport, uint16 t iport, uint16 t soport, uint16 t so

Initialize the cin control library

Initialize the control structures and communications with the CIN via the control interface. This function opens the UDP ports and starts a listening thread to recieve packets from the CIN.

Parameters

cin	handle to cin library
ipaddr	ip address of CIN base address
oport	output udp port of cin
iport	input udp port of cin
soport	stream output udp port of cin
siport	stream input udp port of cin

Returns

Returns 0 on sucsess non-zero if error

4.1 Cin Control Routines 9

4.1.2.3 int cin_ctl_read (cin_ctl_t * cin, uint16_t reg, uint16_t * val)

Read register from CIN

10 Module Documentation

Parameters

cin	handle to cin library
reg	register to read
val	variable to read value of register to

Returns

Returns 0 on sucsess non-zero if error

4.1.2.4 int cin_ctl_stream_write (cin_ctl_t * cin, char * val, int size)

Write stream data to CIN

Parameters

cin	handle to cin library
val	array of values to write
size	size of array pointed to by val

Write stream data to cin in form of 16 bit array.

Returns

Returns 0 on sucsess non-zero if error

4.1.2.5 int cin_ctl_write (cin_ctl_t * cin, uint16_t reg, uint16_t val, int wait)

Write register to CIN

Parameters

cin	handle to cin library
reg	register to write to
val	value to write to register
wait	if non-zero

Write register value to CIN. If wait is non-zero then wait a sleep time of i CIN_CTL_WRITE_SLEEP before releasing the mutex to add flow control to the cin.

Returns

Returns 0 on sucsess non-zero if error

4.1.2.6 int cin_ctl_write_with_readback (cin_ctl_t * cin, uint16_t reg, uint16_t val)

Write register to CIN with readback verification

Parameters

cin	handle to cin library
reg	register to write to
val	value to write to register

Write register value to CIN. Follow write with read of register and compare value. CIN_CTL_WRITE_SLEEP before releasing the mutex to add flow control to the cin.

Returns

Returns 0 on sucsess non-zero if error

4.2 CIN Data Routines

4.2 CIN Data Routines

Data group

12 **Module Documentation**

Class Documentation

5.1 cin_ctl Struct Reference

Public Attributes

- cin_port_t ctl_port
- cin_port_t stream_port
- cin_ctl_config_t config
- cin_ctl_listener_t * listener
- pthread_mutex_t access
- pthread_mutexattr_t access_attr

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

• src/cin.h

5.2 cin_ctl_config Struct Reference

Public Attributes

- char name [CIN_CONFIG_MAX_STRING]
- char firmware_filename [CIN_CONFIG_MAX_STRING]
- int overscan
- int columns
- · int fclk
- uint16_t timing [CIN_CONFIG_MAX_DATA][2]
- int timing_len
- uint16_t fcric [CIN_CONFIG_MAX_DATA][2]
- int fcric_len
- uint16_t bias [CIN_CONFIG_MAX_DATA][2]
- int bias_len

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

• src/cin.h

14 Class Documentation

5.3 cin_ctl_id Struct Reference

Public Attributes

- · uint16_t board_id
- uint16_t serial_no
- uint16_t fpga_ver

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

• src/cin.h

5.4 cin_ctl_listener Struct Reference

Public Attributes

- struct cin_port * cp
- · fifo ctl fifo
- · pthread_t thread_id

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

• src/cin.h

5.5 cin_ctl_pwr_mon_t Struct Reference

Public Attributes

- cin_ctl_pwr_val_t bus_12v0
- cin ctl pwr val t mgmt 3v3
- cin_ctl_pwr_val_t mgmt_2v5
- cin_ctl_pwr_val_t mgmt_1v2
- cin_ctl_pwr_val_t enet_1v0
- cin_ctl_pwr_val_t s3e_3v3
- cin_ctl_pwr_val_t gen_3v3
- cin_ctl_pwr_val_t gen_2v5
- cin_ctl_pwr_val_t v6_0v9
- cin_ctl_pwr_val_t v6_1v0
- cin_ctl_pwr_val_t v6_2v5
- cin_ctl_pwr_val_t fp

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

• src/cin.h

5.6 cin_ctl_pwr_val Struct Reference

Public Attributes

• double i

double v

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

src/cin.h

5.7 cin_data Struct Reference

Public Attributes

```
fifo * packet_fifo
fifo * frame_fifo
fifo * image_fifo
cin_data_threads_t listen_thread
cin_data_threads_t assembler_thread
cin_data_threads_t descramble_thread
pthread_mutex_t listen_mutex
pthread_mutex_t assembler_mutex
pthread mutex t descramble mutex
```

- pthread_mutex_t stats_mutex
- cin_data_callbacks_t callbacks
- cin_port_t dp
- · struct timespec framerate
- · unsigned long int dropped_packets
- · unsigned long int mallformed_packets
- uint16_t last_frame
- descramble_map_t map

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

• src/cin.h

5.8 cin_data_callbacks Struct Reference

Public Attributes

```
    void *(* push )(cin_data_frame_t *)
    void *(* pop )(cin_data_frame_t *)
    cin_data_frame_t * frame
```

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

src/cin.h

5.9 cin_data_frame Struct Reference

Public Attributes

- uint16_t * data
- uint16_t number
- struct timespec timestamp

16 Class Documentation

- int size_x
- int size_y
- void * usr_ptr

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

• src/cin.h

5.10 cin_data_packet Struct Reference

Public Attributes

- unsigned char * data
- int size
- · struct timespec timestamp

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

· src/data.h

5.11 cin_data_proc Struct Reference

Public Attributes

- void *(* input_get)(void *, int)
- void *(* input_put)(void *, int)
- void * input_args
- int reader
- void *(* output_put)(void *)
- void *(* output_get)(void *)
- void * output_args
- cin_data_t * parent

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

· src/data.h

5.12 cin_data_stats Struct Reference

Public Attributes

- int last_frame
- · double framerate
- · double datarate
- double packet_percent_full
- · double frame_percent_full
- · double image_percent_full
- · long int packet_overruns
- long int frame_overruns
- long int image_overruns
- · long int packet_used

- · long int frame_used
- · long int image_used
- · long int dropped_packets
- · long int mallformed_packets

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

• src/cin.h

5.13 cin_data_threads Struct Reference

Public Attributes

- · pthread_t thread_id
- · int started

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

• src/cin.h

5.14 cin_map_t Struct Reference

Public Attributes

- char * name
- uint16_t reg

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

· src/cinregisters.h

5.15 cin_port Struct Reference

Public Attributes

- char * srvaddr
- char * cliaddr
- uint16_t srvport
- uint16_t cliport
- · int sockfd
- struct timeval tv
- · struct sockaddr in sin srv
- struct sockaddr_in sin_cli
- socklen_t slen
- · int rcvbuf
- int rcvbuf_rb

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

• src/cin.h

18 Class Documentation

5.16 descramble_map_t Struct Reference

Public Attributes

- uint32_t * map
- int size_x
- int size_y
- int overscan
- int rows

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

• src/cin.h

5.17 fifo Struct Reference

Public Attributes

- void * data
- void * head
- void * tail [FIFO_MAX_READERS]
- void * end
- · int readers
- long int size
- int elem_size
- int full
- long int overruns
- pthread_mutex_t mutex
- pthread_cond_t signal

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

• src/cin.h

File Documentation

6.1 src/cin.h File Reference

```
#include <stdint.h>
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <sys/time.h>
#include <pthread.h>
```

Classes

- struct cin_ctl_config
- struct fifo
- struct cin_ctl_listener
- struct cin_port
- struct cin_ctl
- · struct cin data frame
- struct cin_data_stats
- struct cin_data_threads
- · struct cin_data_callbacks
- struct descramble_map_t
- struct cin_data
- struct cin_ctl_id
- struct cin_ctl_pwr_val
- struct cin_ctl_pwr_mon_t

Macros

- #define CIN_CTL_IP "192.168.1.207"
- #define CIN_CTL_SVR_PORT 49200
- #define CIN_CTL_CLI_PORT 50200
- #define CIN_CTL_SVR_FRMW_PORT 49202
- #define CIN_CTL_CLI_FRMW_PORT 50202
- #define CIN_CTL_MAX_READ_TRIES 10
- #define CIN_CTL_MAX_WRITE_TRIES 5
- #define CIN_CTL_WRITE_SLEEP 2000

20 File Documentation

- #define CIN CTL POWER ENABLE 0x001F
- #define CIN CTL POWER DISABLE 0x0000
- #define CIN_CTL_FP_POWER_ENABLE 0x0020
- #define CIN CTL DCM LOCKED 0x0001
- #define CIN CTL DCM PSDONE 0x0002
- #define CIN_CTL_DCM_STATUS0 0x0004
- #define CIN CTL DCM STATUS1 0x0008
- #define CIN_CTL_DCM_STATUS2 0x0010
- #define CIN_CTL_DCM_TX1_READY 0x0020
- #define CIN CTL DCM TX2 READY 0x0040
- #define CIN CTL DCM ATCA ALARM 0x0080
- #define CIN CTL TRIG INTERNAL 0x0000
- #define CIN_CTL_TRIG_EXTERNAL_1 0x0001
- #define CIN CTL TRIG EXTERNAL 2 0x0002
- #define CIN_CTL_TRIG_EXTERNAL_BOTH 0x0003
- #define CIN CTL FOCUS BIT 0x0002
- #define CIN CTL FCLK 125 0x0000
- #define CIN CTL FCLK 200 0x0001
- #define CIN CTL FCLK 250 0x0002
- #define CIN_CTL_FCLK_125_C 0x0003
- #define CIN_CTL_FCLK_200_C 0x0004
- #define CIN_CTL_FCLK_250_C 0x0005
- #define CIN_CTL_FCLK_156_C 0x0006
- #define CIN_CTL_FPGA_STS_CFG 0x8000
- #define CIN_CTL_FPGA_STS_FP_PWR 0x0008
- #define CIN_CTL_DCM_STS_ATCA 0x0080
- #define CIN_CTL_DCM_STS_LOCKED 0x0001
- #define CIN_CTL_DCM_STS_OVERIDE 0x0800
- #define CIN_CTL_MUX1_VCLK1 0x0001
- #define CIN_CTL_MUX1_VCLK2 0x0002
- #define CIN_CTL_MUX1_VCLK3 0x0003
- #define CIN_CTL_MUX1_ATG 0x0004
- #define CIN_CTL_MUX1_VFSCLK1 0x0005
- #define CIN_CTL_MUX1_VFSCLK2 0x0006
- #define CIN_CTL_MUX1_VFSCLK3 0x0007
- #define CIN_CTL_MUX1_HCLK1 0x0008
- #define CIN_CTL_MUX1_HCLK2 0x0009
- #define CIN_CTL_MUX1_OSW 0x000A
- #define CIN_CTL_MUX1_RST 0x000B
- #define CIN CTL MUX1 CONVERT 0x000C
- #define CIN_CTL_MUX1_SHUTTER 0x000D
- #define CIN_CTL_MUX1_SWTRIGGER 0x000E
- #define CIN_CTL_MUX1_TRIGMON 0x000F
- #define CIN_CTL_MUX1_EXPOSE 0x0000
- #define CIN_CTL_MUX2_VCLK1 0x0010
- #define CIN_CTL_MUX2_VCLK2 0x0020
- #define CIN_CTL_MUX2_VCLK3 0x0030
- #define CIN CTL MUX2 ATG 0x0040
- #define CIN_CTL_MUX2_VFSCLK1 0x0050
- #define CIN_CTL_MUX2_VFSCLK2 0x0060
- #define CIN CTL MUX2 VFSCLK3 0x0070
- #define CIN_CTL_MUX2_HCLK1 0x0080
- #define CIN_CTL_MUX2_HCLK2 0x0090
- #define CIN_CTL_MUX2_HCLK3 0x00A0
- #define CIN CTL MUX2 OSW 0x00B0

- #define CIN_CTL_MUX2_RST 0x00C0
- #define CIN_CTL_MUX2_CONVERT 0x00D0
- #define CIN_CTL_MUX2_SAVE 0x00E0
- #define CIN CTL MUX2 HWTRIG 0x00F0
- #define CIN CTL MUX2 EXPOSE 0x0000
- #define CIN_CTL_FO_REG1 0x821D
- #define CIN_CTL_FO_REG2 0x821E
- #define CIN_CTL_FO_REG3 0x821F
- #define CIN_CTL_FO_REG4 0x8001
- #define CIN_CTL_FO_REG5 0x8211
- #define CIN_CTL_FO_REG6 0x8212
- #define CIN_CTL_FO_REG7 0x8213
- #define CIN_DATA_IP "10.0.5.207"
- #define CIN DATA PORT 49201
- #define CIN_DATA_CTL_PORT 49203
- #define CIN DATA MAX MTU 9000
- #define CIN DATA UDP HEADER 8
- #define CIN DATA MAGIC PACKET UINT64 C(0x0000F4F3F2F1F000)
- #define CIN_DATA_MAGIC_PACKET_MASK UINT64_C(0x0000FFFFFFFF00)
- #define CIN_DATA_TAIL_MAGIC_PACKET UINT64_C(0x010DF0ADDEF2F1F0)
- #define CIN_DATA_DROPPED_PACKET_VAL 0x2000
- #define CIN DATA DATA MASK 0x1FFF
- #define CIN_DATA_CTRL_MASK 0xE000
- #define CIN DATA SIGN MASK 0x1000
- #define CIN_DATA_GAIN_8 0xC000
- #define CIN_DATA_GAIN_4 0x4000
- #define CIN DATA PACKET LEN 8184
- #define CIN DATA MAX PACKETS 542
- #define CIN_DATA_RCVBUF 100
- #define CIN_DATA_MAX_FRAME_X 1152
- #define CIN_DATA_MAX_FRAME_Y 2050
- #define CIN_DATA_MAX_STREAM 2400000
- #define CIN_DATA_CCD_COLS 96
- #define CIN_DATA_CCD_COLS_PER_CHAN 10
- #define CIN_DATA_PIPELINE_FLUSH 1344
- #define CIN_DATA_MODE_CALLBACK 0x01
- #define NUM_BIAS_VOLTAGE 20
- #define pt_posH 0
- #define pt_negH 1
- #define pt posRG 2
- #define pt_negRG 3
- #define pt_posSW 4
- #define pt_negSW 5
- #define pt_posV 6
- #define pt_negV 7#define pt_posTG 8
- #define pt_negTG 9
- #define **pt_posVF** 10
- #define pt_negVF 11
- #define pt_NEDGE 12
- #define pt OTG 13
- #define pt_VDDR 14
- #define pt VDD OUT 15
- #define pt_BUF_Base 16

22 File Documentation

- #define pt BUF Delta 17
- #define pt_Spare1 18
- #define pt_Spare2 19
- #define DEBUG_PRINT(fmt,...) if(_debug_print_flag) { fprintf(stderr, "%s:%d:%s(): " fmt, __FILE__, __LINE-__, __func__, __VA_ARGS__); }
- #define DEBUG_COMMENT(fmt) if(_debug_print_flag) { fprintf(stderr, "%s:%d:%s(): " fmt, __FILE__, __LI-NE__, __func__); }
- #define ERROR_COMMENT(fmt) if(_error_print_flag) { fprintf(stderr, "%s:%d:%s(): " fmt, __FILE__, __LIN-E__, __func__); }
- #define ERROR_PRINT(fmt,...) if(_error_print_flag) { fprintf(stderr, "%s:%d:%s(): " fmt, __FILE__, __LINE__, __func__, __VA_ARGS__); }
- #define CIN CONFIG MAX STRING 256
- #define CIN CONFIG MAX DATA 5000
- #define FIFO MAX READERS 10

Typedefs

- typedef struct cin_ctl_config cin_ctl_config_t
- typedef struct cin_ctl_listener cin_ctl_listener_t
- typedef struct cin port cin port t
- typedef struct cin ctl cin ctl t
- typedef struct cin data frame cin data frame t
- typedef struct cin_data_stats cin_data_stats_t
- typedef struct cin_data_threads cin_data_threads_t
- typedef struct cin_data_callbacks cin_data_callbacks_t
- typedef struct cin data cin data t
- typedef void(* cin data callback)(cin data frame t *)
- typedef struct cin_ctl_id cin_ctl_id_t
- typedef struct cin_ctl_pwr_val cin_ctl_pwr_val_t

Functions

- void cin set debug print (int debug)
- void cin set error print (int error)
- void cin_report (FILE *fp, int details)
- int cin_ctl_init (cin_ctl_t *cin, const char *ipaddr, uint16_t oport, uint16_t iport, uint16_t siport)
- int cin_ctl_destroy (cin_ctl_t *cin)
- int cin_ctl_read (cin_ctl_t *cin, uint16_t reg, uint16_t *val)
- int cin_ctl_write (cin_ctl_t *cin, uint16_t reg, uint16_t val, int wait)
- int cin ctl stream write (cin ctl t *cin, char *val, int size)
- int cin_ctl_write_with_readback (cin_ctl_t *cin, uint16_t reg, uint16_t val)
- int cin_ctl_pwr (cin_ctl_t *cin, int pwr)
- int cin_ctl_fp_pwr (cin_ctl_t *cin, int pwr)
- int cin_ctl_fo_test_pattern (cin_ctl_t *cin, int on_off)
- int cin ctl load config (cin ctl t *cin, char *filename)
- int cin_ctl_load_firmware (cin_ctl_t *cin, char *filename)
- int cin ctl set fclk (cin ctl t *cin, int clkfreq)
- int cin_ctl_get_fclk (cin_ctl_t *cin, int *clkfreq)
- int cin_ctl_freeze_dco (cin_ctl_t *cin, int freeze)
- int cin ctl get cfg fpga status (cin ctl t *cin, uint16 t * val)
- int cin_ctl_get_id (cin_ctl_t *cin, cin_ctl_id_t *_val)
- void cin_ctl_display_id (FILE *out, cin_ctl_id_t val)
- void cin_ctl_display_fpga_status (FILE *out, uint16_t val)

6.1 src/cin.h File Reference 23

```
• int cin_ctl_get_dcm_status (cin_ctl_t *cin, uint16_t *_val)

    void cin_ctl_display_dcm_status (FILE *out, uint16_t *_val)

· double cin ctl current calc (uint16 t val)

    int cin_ctl_get_power_status (cin_ctl_t *cin, int full, int *pwr, cin_ctl_pwr_mon_t *values)

    void cin_ctl_display_pwr (FILE *out, cin_ctl_pwr_mon_t *values)

• void cin_ctl_display_pwr_line (FILE *out, const char *msg, cin_ctl_pwr_val_t val)

    int cin_ctl_calc_vi_status (cin_ctl_t *cin, uint16_t vreg, uint16_t ireg, double vfact, cin_ctl_pwr_val_t *vi)

    int cin_ctl_get_camera_pwr (cin_ctl_t *cin, int *val)

• int cin ctl set camera pwr (cin ctl t *cin, int val)
• int cin ctl set bias (cin ctl t *cin, int val)

    int cin_ctl_get_bias (cin_ctl_t *cin, int *val)

• int cin ctl set clocks (cin ctl t *cin, int val)
• int cin_ctl_get_clocks (cin_ctl_t *cin, int *val)

    int cin_ctl_set_trigger (cin_ctl_t *cin, int val)

    int cin ctl get trigger (cin ctl t *cin, int *val)

• int cin ctl set focus (cin ctl t *cin, int val)

    int cin_ctl_get_focus (cin_ctl_t *cin, int *val)

• int cin ctl get triggering (cin ctl t *cin, int *trigger)

    int cin_ctl_int_trigger_start (cin_ctl_t *cin, int nimages)

    int cin_ctl_int_trigger_stop (cin_ctl_t *cin)

• int cin_ctl_ext_trigger_start (cin_ctl_t *cin, int trigger_mode)

    int cin ctl ext trigger stop (cin ctl t *cin)

• int cin ctl set exposure time (cin ctl t *cin, float e time)
• int cin_ctl_set_trigger_delay (cin_ctl_t *cin, float t_time)
• int cin_ctl_set_cycle_time (cin_ctl_t *cin, float ftime)

    int cin_ctl_frame_count_reset (cin_ctl_t *cin)

int cin_ctl_set_mux (cin_ctl_t *cin, int setting)
int cin_ctl_get_mux (cin_ctl_t *cin, int *setting)

    int cin_ctl_set_fcric_gain (cin_ctl_t *cin, int gain)

• int cin ctl set fabric address (cin ctl t *cin, char *ip)

    int cin_ctl_reg_dump (cin_ctl_t *cin, FILE *fp)

• int cin_ctl_get_bias_voltages (cin_ctl_t *cin, float *voltage)
• int cin_ctl_set_bias_voltages (cin_ctl_t *cin, float *voltage)

    int cin ctl set fcric clamp (cin ctl t *cin, int clamp)

    int cin_config_read_file (cin_ctl_t *cin, const char *file)

• int cin_data_init (cin_data_t *cin, int mode, int packet_buffer_len, int frame_buffer_len, char *ipaddr, uint16-
  _t port, char *cin_ipaddr, uint16_t cin_port, int rcvbuf, cin_data_callback push_callback, cin_data_callback
 pop_callback, void *usr_ptr)

    void cin_data_wait_for_threads (cin_data_t *cin)

    void cin_data_stop_threads (cin_data_t *cin)

    struct cin_data_frame * cin_data_get_next_frame (cin_data_t *cin)

    void cin data release frame (cin data t *cin, int free mem)

    struct cin_data_frame * cin_data_get_buffered_frame (void)

    void cin_data_release_buffered_frame (void)

• void cin_data_compute_stats (cin_data_t *cin, cin_data_stats_t *stats)

    void cin_data_show_stats (FILE *fp, cin_data_stats_t stats)

    void cin_data_reset_stats (cin_data_t *cin)
```

• int cin data set descramble params (cin data t *cin, int rows, int overscan)

• void cin data get descramble params (cin data t *cin, int *rows, int *overscan, int *xsize, int *ysize)

24 File Documentation

Variables

- const char * cin build git time
- const char * cin_build_git_sha
- const char * cin_build_version
- int debug print flag
- int _error_print_flag

6.1.1 Detailed Description

Author

Stuart B. Wilkins swilkins@bnl.gov

6.1.2 LICENSE

Copyright (c) 2014, Stuart B. Wilkins All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

6.1.3 DESCRIPTION

header file for CIN communications

6.1.4 Function Documentation

6.1.4.1 int cin_data_init (cin_data_t * cin, int mode, int packet_buffer_len, int frame_buffer_len, char * ipaddr, uint16_t port, char * cin_ipaddr, uint16_t cin_port, int rcvbuf, cin_data_callback push_callback, cin_data_callback pop_callback, void * usr_ptr)

Initialize the cin data library

Initialize the data handeling routines and start the threads for listening. mode should be set for the desired output. The packet_buffer_len in the length of the packet FIFO in number of packets. The frame_buffer_len is the number of data frames to buffer.

Parameters

cin Handle to cin data library

Index

```
CIN Data Routines, 11
Cin Control Routines, 7
     cin_ctl_destroy, 8
     cin_ctl_init, 8
     cin_ctl_read, 8
     cin_ctl_stream_write, 10
     cin_ctl_write, 10
     cin_ctl_write_with_readback, 10
cin.h
     cin data init, 24
cin ctl, 13
cin_ctl_config, 13
cin_ctl_destroy
     Cin Control Routines, 8
cin_ctl_id, 14
cin_ctl_init
     Cin Control Routines, 8
cin_ctl_listener, 14
cin_ctl_pwr_mon_t, 14
cin_ctl_pwr_val, 14
cin_ctl_read
     Cin Control Routines, 8
cin ctl stream write
     Cin Control Routines, 10
cin_ctl_write
     Cin Control Routines, 10
cin_ctl_write_with_readback
     Cin Control Routines, 10
cin_data, 15
cin_data_callbacks, 15
cin data frame, 15
cin_data_init
     cin.h, 24
cin_data_packet, 16
cin_data_proc, 16
cin_data_stats, 16
cin_data_threads, 17
cin_map_t, 17
cin_port, 17
descramble_map_t, 18
fifo, 18
src/cin.h, 19
```