## libcin

Generated by Doxygen 1.8.6

Thu Jul 6 2017 19:09:18

# **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	umentati	on														7
	4.1	Cin Co	ntrol Rout	ines						 	 	 	 			 		7
		4.1.1	Detailed	Des	criptio	n				 	 	 	 			 		8
		4.1.2	Function	Doc	ument	ation				 	 	 	 			 		8
			4.1.2.1	cin	_ctl_d	estroy	,			 	 	 	 			 		8
			4.1.2.2	cin	_ctl_in	nit				 	 	 	 			 		8
			4.1.2.3	cin	_ctl_re	ead .				 	 	 	 			 		9
			4.1.2.4	cin	_ctl_st	tream	_write	e		 	 	 	 			 		10
			4.1.2.5	cin	_ctl_w	rite .				 	 	 	 			 		10
			4.1.2.6	cin	_ctl_w	rite_w	vith_r	eadb	ack	 	 	 	 			 		10
5	Clas	s Docu	mentatior	1														11
	5.1	cin_ctl	Struct Re	feren	ice					 	 	 	 			 		11
	5.2	cin_ctl	_config St	ruct I	Refere	nce .				 	 	 	 			 		11
	5.3	cin_ctl	_id Struct	Refe	rence					 	 	 	 			 		12
	5.4	cin_ctl	_listener S	Struct	Refer	ence				 	 	 	 			 		12
	5.5	cin_ctl	_pwr_mon	ı_t St	truct R	lefere	nce			 	 	 	 			 		12
	5.6	cin_ctl	_pwr_val S	Struc	t Refe	rence				 	 	 	 			 		12
	5.7	cin_da	ta Struct F	Refer	ence .					 	 	 	 			 		13
	5.8	cin_da	ta_callbac	ks S	truct F	lefere	nce			 	 	 	 			 		13
	5.9		ta_frame S															14
	5.10		_ ta_packet															14
	5.11		 ta_proc St															14
	5.12	cin da	ta stats S	truct	Refer	ence					 	 						14

iv CONTENTS

	5.13	cin_dat	ta_threads	s Struct Reference	15
	5.14	cin_ma	p_t Struct	t Reference	15
	5.15	cin_po	rt Struct R	deference	15
	5.16	descra	mble_map	o_t Struct Reference	16
	5.17	fifo Stru	uct Refere	ence	16
6	File	Docume	entation		17
	6.1	src/cin.	h File Ref	ference	17
		6.1.1	Detailed	Description	22
		6.1.2	LICENSE	E	22
		6.1.3	DESCRI	PTION	22
		6.1.4	Function	Documentation	22
			6.1.4.1	cin_data_framestore_disable	22
			6.1.4.2	cin_data_framestore_skip	23
			6.1.4.3	cin_data_framestore_trigger	23
			6.1.4.4	cin_data_framestore_trigger_enable	23
			6.1.4.5	cin_data_get_framestore_counter	23
			6.1.4.6	cin_data_init	24
Ind	ex				25

# **Module Index**

1.1 Modules	
-------------	--

Here is a list of all module	s:																		
Cin Control Routines																			7

2 **Module Index** 

# **Class Index**

## 2.1 Class List

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

cin_ctl	11
cin_ctl_config	11
cin_ctl_id	12
	12
	12
*** <u>-**</u> -******************************	12
cin_data	13
	13
	14
	14
	14
	14
	15
	15
	15
	16
fifo	16

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

## **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

12 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\_0v9cin\_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
- pthread\_mutex\_t framestore\_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
- int framestore\_mode
- · struct timespec framestore\_trigger
- int framestore\_counter

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

14 Class Documentation

### 5.9 cin\_data\_frame Struct Reference

#### **Public Attributes**

- uint16\_t \* data
- uint16\_t number
- struct timespec timestamp
- 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

16 Class Documentation

- socklen\_t slen
- · int rcvbuf
- · int rcvbuf\_rb

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

• src/cin.h

## 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

18 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(0x0000FFFFFFFFF00)
- #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

20 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 21

```
• 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)

    void cin_data_framestore_trigger (cin_data_t *cin, int count)

• void cin data framestore skip (cin data t *cin, int count)

    int cin data get framestore counter (cin data t *cin)

    void cin_data_framestore_disable (cin_data_t *cin)

    void cin_data_framestore_trigger_enable (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)

22 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 void cin\_data\_framestore\_disable ( cin\_data\_t \* cin )

Disable the framestore modes

This function disables the framestore modes (software trigger and skip). If the camera is hardware triggering then the images will start to be processed.

#### **Parameters**

cin	Handle to the cin library
-----	---------------------------

#### 6.1.4.2 void cin\_data\_framestore\_skip ( cin\_data\_t \* cin, int count )

#### Enable framestore skip mode

Enable the framestore skip mode. This function should be called before hardware triggering the camera. This causes the data processing to skip

#### **Parameters**

count	frames from the first images to be read. This is usually done to stop the first few frames from
	being over exposed.
cin	handle to the cin_data library

#### 6.1.4.3 void cin\_data\_framestore\_trigger ( cin\_data\_t \* cin, int count )

#### Send a framestore (software) trigger

Send a software trigger to the CIN by timestamping the request time and allow images to be processed when recieved after this time. The count option sets the number of frames to trigger. A value of -1 indicated that the trigger should not count images but run indefinately after the trigger has occured.

#### **Parameters**

cin	handle to the cin_data library
count	number of frames to trigger

#### 6.1.4.4 void cin\_data\_framestore\_trigger\_enable ( cin\_data\_t \* cin )

#### Enable the framestore trigger mode

This function enables the framestore trigger mode. It cases the images to not be processed pending a call to the function to (software) trigger the camera.

#### **Parameters**

cin	Handle to the cin library

#### 6.1.4.5 int cin\_data\_get\_framestore\_counter ( cin\_data\_t \* cin )

#### Get the value of the framestore counter

Return the number of frames in the framestore counter. In trigger mode, this returns the number of frames to go. In skip mode, this returns the number of frames that have to be skipped.

#### **Parameters**

cin	handle to the cin_data library

#### Returns

Number of frames to go in trigger

24 File Documentation

6.1.4.6 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 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_framestore_disable, 22
cin_data_framestore_skip, 23
cin_data_framestore_trigger, 23
cin_data_framestore_trigger_enable, 23
cin_data_get_framestore_counter, 23
cin_data_init, 23
cin_ctl, 11
cin_ctl_config, 11
cin_ctl_destroy
Cin Control Routines, 8
cin_ctl_id, 12
cin ctl init
Cin Control Routines, 8
cin_ctl_listener, 12
cin_ctl_pwr_mon_t, 12
cin_ctl_pwr_val, 12
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, 13
cin_data_callbacks, 13
cin_data_frame, 14
cin_data_framestore_disable
cin.h, 22
cin_data_framestore_skip
cin.h, 23
cin_data_framestore_trigger
cin.h, 23
cin_data_framestore_trigger_enable
cin.h, 23
cin_data_get_framestore_counter
cin.h, 23
cin_data_init
cin.h, 23
cin_data_packet, 14
cin_data_packet, 14
οπ_σαία_ρίου, 1 <del>-1</del>

```
cin_data_stats, 14
cin_data_threads, 15
cin_map_t, 15
cin_port, 15
descramble_map_t, 16
fifo, 16
src/cin.h, 17
```