

libcin

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>FastCCD Communication Library (libcin)</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	Cin Control Initialization Routines . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Function Documentation . . . . .	9
5.1.2.1	cin_ctl_destroy() . . . . .	9
5.1.2.2	cin_ctl_init() . . . . .	10
5.2	Cin Control Read/Rwrite Routines . . . . .	11
5.2.1	Detailed Description . . . . .	11
5.2.2	Function Documentation . . . . .	11
5.2.2.1	cin_ctl_read() . . . . .	11
5.2.2.2	cin_ctl_stream_write() . . . . .	11
5.2.2.3	cin_ctl_write() . . . . .	12
5.2.2.4	cin_ctl_write_with_readback() . . . . .	12
5.3	CIN Control Bias Routines . . . . .	14

5.3.1	Detailed Description	14
5.4	CIN CONTROL Timing Routines	15
5.4.1	Detailed Description	15
5.5	CIN Data Initialization Routines	16
5.5.1	Detailed Description	16
5.5.2	Function Documentation	16
5.5.2.1	cin_data_init()	16
5.5.2.2	cin_data_stop_threads()	17
5.6	CIN Data Framestore Functions	18
5.6.1	Detailed Description	18
5.6.2	Function Documentation	18
5.6.2.1	cin_data_framestore_disable()	18
5.6.2.2	cin_data_framestore_skip()	18
5.6.2.3	cin_data_framestore_trigger()	19
5.6.2.4	cin_data_framestore_trigger_enable()	19
5.6.2.5	cin_data_get_framestore_counter()	19
<b>6</b>	<b>Class Documentation</b>	<b>21</b>
6.1	cin_ctl Struct Reference	21
6.2	cin_ctl_config Struct Reference	21
6.3	cin_ctl_id Struct Reference	22
6.4	cin_ctl_listener Struct Reference	22
6.5	cin_ctl_pwr_mon_t Struct Reference	22
6.6	cin_ctl_pwr_val Struct Reference	23
6.7	cin_data Struct Reference	23
6.8	cin_data_callbacks Struct Reference	23
6.9	cin_data_frame Struct Reference	24
6.10	cin_data_packet Struct Reference	24
6.11	cin_data_proc Struct Reference	24
6.12	cin_data_stats Struct Reference	25
6.13	cin_data_threads Struct Reference	25
6.14	cin_map_t Struct Reference	25
6.15	cin_port Struct Reference	26
6.16	descramble_map_t Struct Reference	26
6.17	fifo Struct Reference	26

<b>7 File Documentation</b>	<b>27</b>
7.1 src/cin.h File Reference . . . . .	27
7.1.1 Detailed Description . . . . .	32
7.1.2 LICENSE . . . . .	32
7.1.3 DESCRIPTION . . . . .	32
7.2 src/cin_register_map.h File Reference . . . . .	33
7.2.1 Detailed Description . . . . .	37
7.2.2 LICENSE . . . . .	38
7.2.3 DESCRIPTION . . . . .	38
<b>Index</b>	<b>39</b>



# Chapter 1

## FastCCD Communication Library (libcin)

### Introduction

This library, based in C is designed to control the FastCCD detector from Lawrence Berkeley National Laboratory. It controls both camera control functions and data acquisition (frame acquisition). It is separated into two distinct parts, the control part ,[cin\\_ctl](#), and the data (image) part named [cin\\_data](#). It was written in part for use with `areaDetector`.

### Prerequisites

The library relies on the following:

- `libbsd` (Used for string manipulation)
- `libconfig` (Used for nice config files)
- `libpthread` (Used for threading)
- `librt` (Used for time functions)

### Installation

Installation of the library is like most unix based source packages:

```
./make
./make doc
./make test
./make install
```

## TCP/IP Stack Tuning

In order for the CIN data to operate efficiently, the 10G interface on the host computer needs to be tuned. This needs to be done by adding the following to the file `/etc/sysctl.conf`.

```
# Increase the maximum buffer that user programs can request
# 2147483647 = 2048 Mb
net.core.rmem_max=2147483647
net.core.wmem_max=2147483647
# Set a default value 10 times bigger
net.core.rmem_default=1000000
net.core.wmem_default=1000000
# increase the length of the processor input queue
net.core.netdev_max_backlog = 250000
# recommended for hosts with jumbo frames enabled
net.ipv4.tcp_mt看u_probing=1
```

These can be reread by the system without rebooting by entering the command:

```
$sudo sysctl --system
```

## Versioning

For the versions available, see the [tags on this repository](#).

## Authors

- **Stuart B. Wilkins** - [stuwilkins](#)

See also the list of [contributors](#) who participated in this project.

## License

This project is licensed under the BSD License - see the [LICENSE](#) file for details

## Acknowledgments

A huge thanks to Peter Dennes, John Joseph and the detector team at LBNL and the team at Sydor Instruments.



## Chapter 2

# Module Index

### 2.1 Modules

Here is a list of all modules:

Cin Control Initialization Routines . . . . .	9
Cin Control Read/Rwrite Routines . . . . .	11
CIN Control Bias Routines . . . . .	14
CIN Control Timing Routines . . . . .	15
CIN Data Initialization Routines . . . . .	16
CIN Data Framestore Functions . . . . .	18



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">cin_ctl</a>	21
<a href="#">cin_ctl_config</a>	21
<a href="#">cin_ctl_id</a>	22
<a href="#">cin_ctl_listener</a>	22
<a href="#">cin_ctl_pwr_mon_t</a>	22
<a href="#">cin_ctl_pwr_val</a>	23
<a href="#">cin_data</a>	23
<a href="#">cin_data_callbacks</a>	23
<a href="#">cin_data_frame</a>	24
<a href="#">cin_data_packet</a>	24
<a href="#">cin_data_proc</a>	24
<a href="#">cin_data_stats</a>	25
<a href="#">cin_data_threads</a>	25
<a href="#">cin_map_t</a>	25
<a href="#">cin_port</a>	26
<a href="#">descramble_map_t</a>	26
<a href="#">fifo</a>	26



## Chapter 4

# File Index

### 4.1 File List

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

src/ <a href="#">cin.h</a> . . . . .	<a href="#">27</a>
src/ <a href="#">cin_register_map.h</a> . . . . .	<a href="#">33</a>
src/ <b>cinregisters.h</b> . . . . .	??
src/ <b>common.h</b> . . . . .	??
src/ <b>config.h</b> . . . . .	??
src/ <b>control.h</b> . . . . .	??
src/ <b>data.h</b> . . . . .	??
src/ <b>descramble.h</b> . . . . .	??
src/ <b>descramble_map.h</b> . . . . .	??
src/ <b>fclk_program.h</b> . . . . .	??
src/ <b>fifo.h</b> . . . . .	??
src/ <b>report.h</b> . . . . .	??



## Chapter 5

# Module Documentation

### 5.1 Cin Control Initialization Routines

#### Functions

- int [cin\\_ctl\\_init](#) ([cin\\_ctl\\_t](#) \*cin, const char \*ipaddr, const char \*bind\_addr, uint16\_t oport, uint16\_t iport, uint16\_t soport, uint16\_t siport)
- int [cin\\_ctl\\_destroy](#) ([cin\\_ctl\\_t](#) \*cin)

#### 5.1.1 Detailed Description

#### 5.1.2 Function Documentation

##### 5.1.2.1 cin\_ctl\_destroy()

```
int cin_ctl_destroy (
    cin\_ctl\_t * cin )
```

Destroy (close) the cin control library

Close connections, free memory and exit library

#### Parameters

<i>cin</i>	handle to cin library
------------	-----------------------

#### Returns

Returns 0 on success non-zero if error

### 5.1.2.2 cin\_ctl\_init()

```
int cin_ctl_init (
    cin_ctl_t * cin,
    const char * ipaddr,
    const char * bind_addr,
    uint16_t oport,
    uint16_t iport,
    uint16_t soport,
    uint16_t siport )
```

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 receive packets from the CIN.

#### Parameters

<i>cin</i>	handle to cin library
<i>ipaddr</i>	ip address of CIN base address
<i>bind_addr</i>	ip address to bind to
<i>oport</i>	output udp port of cin
<i>iport</i>	input udp port of cin
<i>soport</i>	stream output udp port of cin
<i>siport</i>	stream input udp port of cin

#### Returns

Returns 0 on success non-zero if error



## 5.2 Cin Control Read/Rwrite Routines

### Functions

- int `cin_ctl_read` (`cin_ctl_t` \*cin, uint16\_t reg, uint16\_t \*val, int wait)
- 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, unsigned 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)

### 5.2.1 Detailed Description

### 5.2.2 Function Documentation

#### 5.2.2.1 cin\_ctl\_read()

```
int cin_ctl_read (
    cin_ctl_t * cin,
    uint16_t reg,
    uint16_t * val,
    int wait )
```

Read register from CIN

#### Parameters

<i>cin</i>	handle to cin library
<i>reg</i>	register to read
<i>val</i>	variable to read value of register to
<i>wait</i>	if non-zero, wait a predefined time before read command (for i2c)

#### Returns

Returns 0 on success non-zero if error

#### 5.2.2.2 cin\_ctl\_stream\_write()

```
int cin_ctl_stream_write (
    cin_ctl_t * cin,
    unsigned char * val,
    int size )
```

Write stream data to CIN

**Parameters**

<i>cin</i>	handle to cin library
<i>val</i>	array of values to write
<i>size</i>	size of array pointed to by val

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

**Returns**

Returns 0 on success non-zero if error

**5.2.2.3 cin\_ctl\_write()**

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

Write register to CIN

**Parameters**

<i>cin</i>	handle to cin library
<i>reg</i>	register to write to
<i>val</i>	value to write to register
<i>wait</i>	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 success non-zero if error

**5.2.2.4 cin\_ctl\_write\_with\_readback()**

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

Write register to CIN with readback verification

**Parameters**

<i>cin</i>	handle to cin library
<i>reg</i>	register to write to
<i>val</i>	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 success non-zero if error

## 5.3 CIN Control Bias Routines

### Functions

- 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\_bias\_regs** ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals, int verify)
- int **cin\_ctl\_get\_bias\_regs** ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals)
- int **cin\_ctl\_set\_bias\_voltages** ([cin\\_ctl\\_t](#) \*cin, float \*voltage, int verify)
- int **cin\_ctl\_get\_bias\_voltages** ([cin\\_ctl\\_t](#) \*cin, float \*voltage)

### 5.3.1 Detailed Description

Initialization group

## 5.4 CIN Control Timing Routines

### Functions

- int **cin\_ctl\_set\_timing\_regs** ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals, int vals\_len)
- int **cin\_ctl\_get\_timing\_regs** ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals)

### 5.4.1 Detailed Description

Timing setup group

## 5.5 CIN Data Initialization Routines

### Functions

- int [cin\\_data\\_init](#) ([cin\\_data\\_t](#) \*cin, 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\\_stop\\_threads](#) ([cin\\_data\\_t](#) \*cin)

### 5.5.1 Detailed Description

Initialization group

### 5.5.2 Function Documentation

#### 5.5.2.1 cin\_data\_init()

```
int cin_data_init (
    cin\_data\_t * cin,
    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 handling routines and start the threads for listening.

#### Parameters

<i>cin</i>	Handle to cin data library
<i>packet_buffer_len</i>	Length of packet buffer fifo (in units number of packets)
<i>frame_buffer_len</i>	Length of frame (assembler) buffer fifo (in units of number of frames)
<i>ipaddr</i>	IP-Address to bind to (if NULL binds to 0.0.0.0)
<i>port</i>	UDP Port of host
<i>cin_ipaddr</i>	IP-Address of cin (if NULL defaults to standard)
<a href="#">cin_port</a>	UDP Port of CIN
<i>rcvbuf</i>	TCP/IP Kernel receive buffer size
<i>push_callback</i>	This function is called when a data structure is needed
<i>pop_callback</i>	This function is called when an image has been processed
<i>usr_ptr</i>	Pointer passed to callback functions

### 5.5.2.2 cin\_data\_stop\_threads()

```
void cin_data_stop_threads (
    cin_data_t * cin )
```

Stop all threads and wait

Stop all the processing threads and join them to the main thread. This function blocks until all threads have joined the main thread (program). This should be called to clean up the library before the program is exited

#### Parameters

<i>cin</i>	Handle to cin data library
------------	----------------------------

## 5.6 CIN Data Framestore Functions

### Functions

- 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)

### 5.6.1 Detailed Description

Framestore Group

### 5.6.2 Function Documentation

#### 5.6.2.1 [cin\\_data\\_framestore\\_disable\(\)](#)

```
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

<i>cin</i>	Handle to the cin library
------------	---------------------------

#### 5.6.2.2 [cin\\_data\\_framestore\\_skip\(\)](#)

```
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

<i>count</i>	frames from the first images to be read. This is usually done to stop the first few frames from being over exposed.
<i>cin</i>	handle to the <a href="#">cin_data</a> library

5.6.2.3 `cin_data_framestore_trigger()`

```
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 indefinitely after the trigger has occurred.

## Parameters

<i>cin</i>	handle to the <a href="#">cin_data</a> library
<i>count</i>	number of frames to trigger

5.6.2.4 `cin_data_framestore_trigger_enable()`

```
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

<i>cin</i>	Handle to the cin library
------------	---------------------------

5.6.2.5 `cin_data_get_framestore_counter()`

```
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

<i>cin</i>	handle to the <a href="#">cin_data</a> library
------------	--

#### Returns

Number of frames to go in trigger

## Chapter 6

# Class Documentation

### 6.1 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](#)

### 6.2 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](#)

## 6.3 cin\_ctl\_id Struct Reference

### Public Attributes

- uint16\_t **base\_board\_id**
- uint16\_t **base\_serial\_no**
- uint16\_t **base\_fpga\_ver**
- uint16\_t **fabric\_board\_id**
- uint16\_t **fabric\_serial\_no**
- uint16\_t **fabric\_fpga\_ver**

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

- [src/cin.h](#)

## 6.4 cin\_ctl\_listener Struct Reference

### Public Attributes

- struct [cin\\_port](#) \* **cp**
- [fifo](#) **ctl\_fifo**
- pthread\_t **thread\_id**
- pthread\_barrier\_t **barrier**

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

- [src/cin.h](#)

## 6.5 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](#)

## 6.6 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](#)

## 6.7 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 **malformed\_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](#)

## 6.8 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](#)

## 6.9 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](#)

## 6.10 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

## 6.11 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

## 6.12 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 **malformed\_packets**

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

- src/[cin.h](#)

## 6.13 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](#)

## 6.14 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

## 6.15 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](#)

## 6.16 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](#)

## 6.17 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](#)



# Chapter 7

## File Documentation

### 7.1 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\_RCVBUF** 10
- #define **CIN\_CTL\_MAX\_READ\_TRIES** 10
- #define **CIN\_CTL\_MAX\_WRITE\_TRIES** 5
- #define **CIN\_CTL\_WRITE\_SLEEP** 100
- #define **CIN\_CTL\_STREAM\_CHUNK** 256
- #define **CIN\_CTL\_STREAM\_SLEEP** 10
- #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\_OVERRIDE** 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\_VFCLK1** 0x0005
- #define **CIN\_CTL\_MUX1\_VFCLK2** 0x0006
- #define **CIN\_CTL\_MUX1\_VFCLK3** 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\_VFCLK1** 0x0050

- #define **CIN\_CTL\_MUX2\_VFCLK2** 0x0060
- #define **CIN\_CTL\_MUX2\_VFCLK3** 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(0x0000FFFFFFFFFFFF00)
- #define **CIN\_DATA\_TAIL\_MAGIC\_PACKET** UINT64\_C(0x010DF0ADDEF2F1F0)
- #define **CIN\_DATA\_TAIL\_MAGIC\_PACKET\_MASK** UINT64\_C(0xFFFFFFFFFFFFFFFF)
- #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\_CTL\_NUM\_BIAS\_VOLTAGE** 20
- #define **CIN\_CTL\_BIAS\_POSH** 0
- #define **CIN\_CTL\_BIAS\_NEGH** 1
- #define **CIN\_CTL\_BIAS\_POSRG** 2
- #define **CIN\_CTL\_BIAS\_NEGRG** 3
- #define **CIN\_CTL\_BIAS\_POSSW** 4
- #define **CIN\_CTL\_BIAS\_NEGSW** 5
- #define **CIN\_CTL\_BIAS\_POSV** 6
- #define **CIN\_CTL\_BIAS\_NEGV** 7
- #define **CIN\_CTL\_BIAS\_POSTG** 8
- #define **CIN\_CTL\_BIAS\_NEGTG** 9
- #define **CIN\_CTL\_BIAS\_POSVF** 10
- #define **CIN\_CTL\_BIAS\_NEGVF** 11

- `#define CIN_CTL_BIAS_NEDGE 12`
- `#define CIN_CTL_BIAS_OTG 13`
- `#define CIN_CTL_BIAS_VDDR 14`
- `#define CIN_CTL_BIAS_VDD_OUT 15`
- `#define CIN_CTL_BIAS_BUF_BASE 16`
- `#define CIN_CTL_BIAS_BUF_DELTA 17`
- `#define CIN_CTL_BIAS_SPARE1 18`
- `#define CIN_CTL_BIAS_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__, __LINE__, __func__); }`
- `#define ERROR_COMMENT(fmt) if(_error_print_flag) { fprintf(stderr, "%s:%d:%s(): " fmt, __FILE__, __LINE__, __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, const char \*bind\_addr, uint16\_t oport, uint16\_t iport, uint16\_t soport, 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 wait)
- 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, unsigned 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)
- int `cin_ctl_load_firmware_file` (`cin_ctl_t` \*cin, char \*filename)

- int [cin\\_ctl\\_load\\_firmware\\_data](#) ([cin\\_ctl\\_t](#) \*cin, unsigned char \*data, int data\_len)
- int [cin\\_ctl\\_get\\_fclk](#) ([cin\\_ctl\\_t](#) \*cin, int \*clkfreq)
- int [cin\\_ctl\\_set\\_fclk](#) ([cin\\_ctl\\_t](#) \*cin, int clkfreq)
- 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)
- int [cin\\_ctl\\_get\\_dcm\\_status](#) ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*\_val)
- int [cin\\_ctl\\_get\\_power\\_status](#) ([cin\\_ctl\\_t](#) \*cin, int full, int \*pwr, [cin\\_ctl\\_pwr\\_mon\\_t](#) \*values)
- 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\\_bias\\_regs](#) ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals, int verify)
- int [cin\\_ctl\\_get\\_bias\\_regs](#) ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals)
- int [cin\\_ctl\\_set\\_bias\\_voltages](#) ([cin\\_ctl\\_t](#) \*cin, float \*voltage, int verify)
- int [cin\\_ctl\\_get\\_bias\\_voltages](#) ([cin\\_ctl\\_t](#) \*cin, float \*voltage)
- int [cin\\_ctl\\_set\\_timing\\_regs](#) ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals, int vals\_len)
- int [cin\\_ctl\\_get\\_timing\\_regs](#) ([cin\\_ctl\\_t](#) \*cin, uint16\_t \*vals)
- 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\\_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\\_clamp](#) ([cin\\_ctl\\_t](#) \*cin, int clamp)
- 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\\_config\\_read\\_file](#) ([cin\\_ctl\\_t](#) \*cin, const char \*file)
- int [cin\\_data\\_init](#) ([cin\\_data\\_t](#) \*cin, 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\\_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)

## 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**
- uint16\_t **cin\_config\_timing []**
- int **cin\_config\_timing\_len**
- unsigned char **cin\_config\_firmware []**
- unsigned **cin\_config\_firmware\_len**
- uint16\_t **cin\_config\_bias []**
- int **cin\_config\_bias\_len**
- uint16\_t **cin\_config\_fcric\_200 []**
- int **cin\_config\_fcric\_200\_len**

### 7.1.1 Detailed Description

#### Author

Stuart B. Wilkins [swilkins@bnl.gov](mailto:swilkins@bnl.gov)

### 7.1.2 LICENSE

Copyright (c) 2014, Brookhaven Science Associates, Brookhaven National Laboratory All rights reserved.

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

1. 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.

### 7.1.3 DESCRIPTION

header file for CIN communications

## 7.2 src/cin\_register\_map.h File Reference

### Macros

- #define **REG\_COMMAND** 0x0001
- #define **REG\_READ\_ADDRESS** 0x0002
- #define **REG\_STREAM\_TYPE** 0x0003
- #define **CMD\_FCLK\_125** 0xB000
- #define **CMD\_FCLK\_200** 0x7000
- #define **CMD\_FCLK\_250** 0x3000
- #define **REG\_IF\_MAC0** 0x0010
- #define **REG\_IF\_MAC1** 0x0011
- #define **REG\_IF\_MAC2** 0x0012
- #define **REG\_IF\_IP0** 0x0013
- #define **REG\_IF\_IP1** 0x0014
- #define **REG\_IF\_CMD\_PORT\_NUM** 0x001A
- #define **REG\_IF\_STREAM\_IN\_PORT\_NUM** 0x001C
- #define **REG\_IF\_STREAM\_OUT\_PORT\_NUM** 0x001D
- #define **REG\_ETH\_RESET** 0x0020 /\* Reset Eth Hardware 1=Rx, 2=Tx, 3=Both \*/
- #define **REG\_ETH\_ENABLE** 0x0021 /\* Enable Eth Hardware 1=Rx, 2=Tx, 3=Both \*/
- #define **REG\_PHY1\_MDIO\_CMD** 0x0022 /\* Start(1), RnW(1), WDRd(1), PHY Addr(5), REG Addr(5) \*/
- #define **REG\_PHY1\_MDIO\_CMD\_DATA** 0x0023
- #define **REG\_PHY1\_MDIO\_STATUS** 0x0024
- #define **REG\_PHY1\_MDIO\_RD\_ADDR** 0x0025
- #define **REG\_PHY1\_MDIO\_RD\_DATA** 0x0026
- #define **REG\_MAC\_CFG\_VECTOR1** 0x0027 /\* Ethernet Hardware Conf \*/
- #define **REG\_PHY2\_MDIO\_CMD** 0x0028
- #define **REG\_PHY2\_MDIO\_CMD\_DATA** 0x0029
- #define **REG\_PHY2\_MDIO\_STATUS** 0x002A
- #define **REG\_PHY2\_MDIO\_RD\_ADDR** 0x002B
- #define **REG\_PHY2\_MDIO\_RD\_DATA** 0x002C
- #define **REG\_MAC\_CFG\_VECTOR2** 0x002D /\* Ethernet Hardware Conf \*/
- #define **CMD\_PS\_ENABLE** 0x0021 /\* Enable Selected Power Modules \*/
- #define **CMD\_PS\_POWERDOWN** 0x0022 /\* Start power down sequence \*/
- #define **REG\_PS\_ENABLE** 0x0030 /\* Power Supply Enable: \*/
- #define **REG\_PS\_SYNC\_DIV0** 0x0031 /\* 2.5V Gen \*/
- #define **REG\_PS\_SYNC\_DIV1** 0x0032 /\* 3.3V Gen \*/
- #define **REG\_PS\_SYNC\_DIV2** 0x0033 /\* 2.5V Frame \*/
- #define **REG\_PS\_SYNC\_DIV3** 0x0034 /\* 0.9V Frame \*/
- #define **REG\_PS\_SYNC\_DIV4** 0x0035 /\* 5.0V FP \*/
- #define **CMD\_PROGRAM\_FRAME** 0x0041
- #define **REG\_FRM\_RESET** 0x0036 /\* Frame Reset \*/
- #define **REG\_FRM\_10GbE\_SEL** 0x0037; /\* 10GbE Link Select \*/
- #define **CMD\_ENABLE\_CLKS** 0x0031 /\* Enable selected Frame FPGA clock crystals \*/
- #define **CMD\_DISABLE\_CLKS** 0x0032 /\* Disable Frame FPGA clock crystals \*/
- #define **REG\_CLOCK\_EN\_REG** 0x0038 /\* Clock Enable Register \*/
- #define **REG\_SI570\_REG0** 0x0039
- #define **REG\_SI570\_REG1** 0x003A
- #define **REG\_SI570\_REG2** 0x003B
- #define **REG\_SI570\_REG3** 0x003C
- #define **CMD\_MON\_STOP** 0x0011 /\* Stop voltage and current monitor \*/
- #define **CMD\_MON\_START** 0x0012 /\* Start voltage and current monitor \*/
- #define **REG\_VMON\_ADC1\_CH1** 0x0040 /\* V12P\_BUS Voltage Monitor \*/
- #define **REG\_IMON\_ADC1\_CH0** 0x0041 /\* V12P\_BUS Current Monitor \*/

- #define **REG\_VMON\_ADC0\_CH5** 0x0042 /\* V3P3\_MGMT Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH5** 0x0043 /\* V3P3\_MGMT Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH4** 0x0044 /\* V3P3\_S3E Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH4** 0x0045 /\* V3P3\_S3E Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH7** 0x0046 /\* V2P5\_MGMT Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH7** 0x0047 /\* V2P5\_MGMT Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH6** 0x0048 /\* V1P8\_MGMT Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH6** 0x0049 /\* V1P8\_MGMT Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH2** 0x004A /\* V1P2\_MGMT Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH2** 0x004B /\* V1P2\_MGMT Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH3** 0x004C /\* V1P0\_ENET Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH3** 0x004D /\* V1P0\_ENET Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH8** 0x004E /\* V3P3\_GEN Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH8** 0x004F /\* V3P3\_GEN Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CH9** 0x0050 /\* V2P5\_GEN Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CH9** 0x0051 /\* V2P5\_GEN Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CHE** 0x0052 /\* V0P9\_V6 Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CHE** 0x0053 /\* V0P9\_V6 Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CHD** 0x0054 /\* V2P5\_V6 Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CHD** 0x0055 /\* V2P5\_V6 Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CHB** 0x0056 /\* V1P0\_V6 Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CHB** 0x0057 /\* V1P0\_V6 Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CHC** 0x0058 /\* V1P2\_V6 Voltage Monitor \*/
- #define **REG\_IMON\_ADC0\_CHC** 0x0059 /\* V1P2\_V6 Current Monitor \*/
- #define **REG\_VMON\_ADC0\_CHF** 0x005A /\* V5P0\_FP Voltage Monitor (1/2) \*/
- #define **REG\_IMON\_ADC0\_CHF** 0x005B /\* V5P0\_FP Current Monitor (1/2) \*/
- #define **REG\_DCM\_STATUS** 0x0080
- #define **REG\_FPGA\_STATUS** 0x0081
- #define **REG\_BOARD\_ID** 0x008D
- #define **REG\_HW\_SERIAL\_NUM** 0x008E
- #define **REG\_FPGA\_VERSION** 0x008F
- #define **REG\_SANDBOX\_REG00** 0x00F0
- #define **REG\_SANDBOX\_REG01** 0x00F1
- #define **REG\_SANDBOX\_REG02** 0x00F2
- #define **REG\_SANDBOX\_REG03** 0x00F3
- #define **REG\_SANDBOX\_REG04** 0x00F4
- #define **REG\_SANDBOX\_REG05** 0x00F5
- #define **REG\_SANDBOX\_REG06** 0x00F6
- #define **REG\_SANDBOX\_REG07** 0x00F7
- #define **REG\_SANDBOX\_REG08** 0x00F8
- #define **REG\_SANDBOX\_REG09** 0x00F9
- #define **REG\_SANDBOX\_REG0A** 0x00FA
- #define **REG\_SANDBOX\_REG0B** 0x00FB
- #define **REG\_SANDBOX\_REG0C** 0x00FC
- #define **REG\_SANDBOX\_REG0D** 0x00FD
- #define **REG\_SANDBOX\_REG0E** 0x00FE
- #define **REG\_SANDBOX\_REG0F** 0x00FF
- #define **REG\_FRM\_COMMAND** 0x8001
- #define **REG\_FRM\_READ\_ADDRESS** 0x8002
- #define **REG\_FRM\_STREAM\_TYPE** 0x8003
- #define **CMD\_SEND\_SYNC\_PULSE** 0x0100 /\*\* ISSUES A SYNC PULSE \*/
- #define **CMD\_SYNC\_DETECTOR2READOUT** 0x0101 /\* COMMAND TO SYNC DETECTOR AND READOUT (SEE IMAGE PROCESSING) \*/
- #define **CMD\_WR\_CCD\_BIAS\_REG** 0x0102 /\*\* WRITE CCD BIAS REGISTERS \*/
- #define **CMD\_WR\_CCD\_CLOCK\_REG** 0x0103 /\*\* WRITE CCD CLOCK REGISTER \*/



- #define **CMD\_SEND\_FCRIC\_CONFIG** 0x0105 /\*\* SEND CONFIG DATA TO FRIC \*/
- #define **CMD\_RESET\_FRAME\_COUNT** 0x0106 /\*\* RESET STATISTICS/DEBUG COUNTERS \*/
- #define **REG\_IF\_MAC\_FAB1B0** 0x8010
- #define **REG\_IF\_MAC\_FAB1B1** 0x8011
- #define **REG\_IF\_MAC\_FAB1B2** 0x8012
- #define **REG\_IF\_IP\_FAB1B0** 0x8013
- #define **REG\_IF\_IP\_FAB1B1** 0x8014
- #define **REG\_IF\_CMD\_PORT\_NUM\_FAB1B** 0x8015
- #define **REG\_IF\_STREAM\_IN\_PORT\_NUM\_FAB1B** 0x8016
- #define **REG\_IF\_STREAM\_OUT\_PORT\_NUM\_FAB1B** 0x8017
- #define **REG\_XAUI\_FAB1B** 0x8018
- #define **REG\_MAC\_CONFIG\_VEC\_FAB1B0** 0x8019
- #define **REG\_MAC\_CONFIG\_VEC\_FAB1B1** 0x801A
- #define **REG\_MAC\_STATS1\_FAB1B0** 0x801B
- #define **REG\_MAC\_STATS1\_FAB1B1** 0x801C
- #define **REG\_MAC\_STATS2\_FAB1B0** 0x801D
- #define **REG\_MAC\_STATS2\_FAB1B1** 0x801E
- #define **REG\_IF\_MAC\_FAB2B0** 0x8020
- #define **REG\_IF\_MAC\_FAB2B1** 0x8021
- #define **REG\_IF\_MAC\_FAB2B2** 0x8022
- #define **REG\_IF\_IP\_FAB2B0** 0x8023
- #define **REG\_IF\_IP\_FAB2B1** 0x8024
- #define **REG\_IF\_CMD\_PORT\_NUM\_FAB2B** 0x8025
- #define **REG\_IF\_STREAM\_IN\_PORT\_NUM\_FAB2B** 0x8026
- #define **REG\_IF\_STREAM\_OUT\_PORT\_NUM\_FAB2B** 0x8027
- #define **REG\_XAUI\_FAB2B** 0x8028
- #define **REG\_MAC\_CONFIG\_VEC\_FAB2B0** 0x8029
- #define **REG\_MAC\_CONFIG\_VEC\_FAB2B1** 0x802A
- #define **REG\_MAC\_STATS1\_FAB2B0** 0x802B
- #define **REG\_MAC\_STATS1\_FAB2B1** 0x802C
- #define **REG\_MAC\_STATS2\_FAB2B0** 0x802D
- #define **REG\_MAC\_STATS2\_FAB2B1** 0x802E
- #define **REG\_SRAM\_COMMAND** 0x8030
- #define **REG\_SRAM\_START\_ADDR1** 0x8031
- #define **REG\_SRAM\_START\_ADDR0** 0x8032
- #define **REG\_SRAM\_STOP\_ADDR1** 0x8033
- #define **REG\_SRAM\_STOP\_ADDR0** 0x8034
- #define **REG\_SRAM\_FRAME\_DATA\_OUT1** 0x8035
- #define **REG\_SRAM\_FRAME\_DATA\_OUT0** 0x8036
- #define **REG\_SRAM\_FRAME\_DATA\_IN1** 0x8037
- #define **REG\_SRAM\_FRAME\_DATA\_IN0** 0x8038
- #define **REG\_SRAM\_FRAME\_DV** 0x8039
- #define **REG\_SRAM\_STATUS1** 0x803A
- #define **REG\_SRAM\_STATUS0** 0x803B
- #define **CMD\_FCLK\_COMMIT** 0x0012 /\*\* Start I2C Write/Read \*/
- #define **REG\_FCLK\_I2C\_ADDRESS** 0x8040 /\*\* [ Slave Address(7), RD/WRn(1), Reg Address(8) ] Slave address Hx58 -> HxB when shifted up by 1 \*/
- #define **REG\_FCLK\_I2C\_DATA\_WR** 0x8041 /\*\* [ Clock Select(2), Clock Enable (1), 0(5), Write Data (8) ] \*/
- #define **REG\_FCLK\_I2C\_DATA\_RD** 0x8042 /\*\* [ Read Failed (1), Write Failed(1), Toggle bit(1), 0(5), Read Data (8) ] \*/
- #define **REG\_TRIGGERSELECT\_REG** 0x8050
- #define **REG\_TRIGGERMASK\_REG** 0x8051 /\*\* [00]==SW Trigger, [01]==FP TrigIn2, [10]==FP TrigIn1, [11]==FP TrigIn1OR2 \*/
- #define **REG\_CCDCLKSELECT\_REG** 0x8052
- #define **REG\_CDCLKDISABLE\_REG** 0x8053

- **#define REG\_FCLK\_SET0** 0xB007
- **#define REG\_FCLK\_SET1** 0xB008
- **#define REG\_FCLK\_SET2** 0xB009
- **#define REG\_FCLK\_SET3** 0xB00A
- **#define REG\_FCLK\_SET4** 0xB00B
- **#define REG\_FCLK\_SET5** 0xB00C
- **#define REG\_FRM\_DCM\_STATUS** 0x8080
- **#define REG\_FRM\_FPGA\_STATUS** 0x8081
- **#define REG\_FRM\_BOARD\_ID** 0x808D
- **#define REG\_FRM\_HW\_SERIAL\_NUM** 0x808E
- **#define REG\_FRM\_FPGA\_VERSION** 0x808F
- **#define REG\_FRM\_SANDBOX\_REG00** 0x80F0
- **#define REG\_FRM\_SANDBOX\_REG01** 0x80F1
- **#define REG\_FRM\_SANDBOX\_REG02** 0x80F2
- **#define REG\_FRM\_SANDBOX\_REG03** 0x80F3
- **#define REG\_FRM\_SANDBOX\_REG04** 0x80F4
- **#define REG\_FRM\_SANDBOX\_REG05** 0x80F5
- **#define REG\_FRM\_SANDBOX\_REG06** 0x80F6
- **#define REG\_FRM\_SANDBOX\_REG07** 0x80F7
- **#define REG\_FRM\_SANDBOX\_REG08** 0x80F8
- **#define REG\_FRM\_SANDBOX\_REG09** 0x80F9
- **#define REG\_FRM\_SANDBOX\_REG0A** 0x80FA
- **#define REG\_FRM\_SANDBOX\_REG0B** 0x80FB
- **#define REG\_FRM\_SANDBOX\_REG0C** 0x80FC
- **#define REG\_FRM\_SANDBOX\_REG0D** 0x80FD
- **#define REG\_FRM\_SANDBOX\_REG0E** 0x80FE
- **#define REG\_FRM\_SANDBOX\_REG0F** 0x80FF
- **#define REG\_DETECTOR\_REVISION\_REG** 0x8100
- **#define REG\_DETECTOR\_CONFIG\_REG1** 0x8101
- **#define REG\_DETECTOR\_CONFIG\_REG2** 0x8102
- **#define REG\_DETECTOR\_CONFIG\_REG3** 0x8103
- **#define REG\_DETECTOR\_CONFIG\_REG4** 0x8104
- **#define REG\_DETECTOR\_CONFIG\_REG5** 0x8105
- **#define REG\_DETECTOR\_CONFIG\_REG6** 0x8106
- **#define REG\_DETECTOR\_CONFIG\_REG7** 0x8107
- **#define REG\_DETECTOR\_CONFIG\_REG8** 0x8108
- **#define REG\_IMG\_PROC\_REVISION\_REG** 0x8120
- **#define REG\_IMG\_PROC\_CONFIG\_REG1** 0x8121
- **#define REG\_IMG\_PROC\_CONFIG\_REG2** 0x8122
- **#define REG\_IMG\_PROC\_CONFIG\_REG3** 0x8123
- **#define REG\_IMG\_PROC\_CONFIG\_REG4** 0x8124
- **#define REG\_IMG\_PROC\_CONFIG\_REG5** 0x8125
- **#define REG\_IMG\_PROC\_CONFIG\_REG6** 0x8126
- **#define REG\_IMG\_PROC\_CONFIG\_REG7** 0x8127
- **#define REG\_IMG\_PROC\_CONFIG\_REG8** 0x8128
- **#define REG\_BIASANDCLOCKREGISTERADDRESS** 0x8200
- **#define REG\_BIASANDCLOCKREGISTERDATA** 0x8201
- **#define REG\_CLOCKREGISTERDATAOUT** 0x8202
- **#define REG\_BIASREGISTERDATAOUT** 0x8203
- **#define REG\_BIASCONFIGREGISTER0\_REG** 0x8204
- **#define REG\_CLOCKCONFIGREGISTER0\_REG** 0x8205
- **#define REG\_BIASPARAM\_READ\_START** 0x3000
- **#define REG\_EXPOSURETIMEMSB\_REG** 0x8206
- **#define REG\_EXPOSURETIMELSB\_REG** 0x8207
- **#define REG\_ALTEXPOSURETIMEMSB\_REG** 0x8306

- #define REG\_ALTEXPOSURETIMELSB\_REG 0x8307
- #define REG\_TRIGGERREPETITIONTIMEMSB\_REG 0x8208
- #define REG\_TRIGGERREPETITIONTIMELSB\_REG 0x8209
- #define REG\_DELAYTOEXPOSUREMSB\_REG 0x820A
- #define REG\_DELAYTOEXPOSURELSB\_REG 0x820B
- #define REG\_NUMBEROFEXPOSURE\_REG 0x820C
- #define REG\_SHUTTERTIMEMSB\_REG 0x820D
- #define REG\_SHUTTERTIMELSB\_REG 0x820E
- #define REG\_DELAYTOSHUTTERMSB\_REG 0x820F
- #define REG\_DELAYTOSHUTTERLSB\_REG 0x8210
- #define REG\_FCRIC\_MASK\_REG1 0x8211
- #define REG\_FCRIC\_MASK\_REG2 0x8212
- #define REG\_FCRIC\_MASK\_REG3 0x8213
- #define REG\_LVDS\_OVERFLOW\_ERROR\_REG1 0x8214
- #define REG\_LVDS\_OVERFLOW\_ERROR\_REG2 0x8215
- #define REG\_LVDS\_OVERFLOW\_ERROR\_REG3 0x8216
- #define REG\_LVDS\_PARITY\_ERROR\_REG1 0x8217
- #define REG\_LVDS\_PARITY\_ERROR\_REG2 0x8218
- #define REG\_LVDS\_PARITY\_ERROR\_REG3 0x8219
- #define REG\_LVDS\_STOP\_BIT\_ERROR\_REG1 0x821A
- #define REG\_LVDS\_STOP\_BIT\_ERROR\_REG2 0x821B
- #define REG\_LVDS\_STOP\_BIT\_ERROR\_REG3 0x821C
- #define REG\_FCRIC\_WRITE0\_REG 0x821D
- #define REG\_FCRIC\_WRITE1\_REG 0x821E
- #define REG\_FCRIC\_WRITE2\_REG 0x821F
- #define REG\_FCRIC\_READ0\_REG 0x8220
- #define REG\_FCRIC\_READ1\_REG 0x8221
- #define REG\_FCRIC\_READ2\_REG 0x8222
- #define REG\_DEBUGVIDEO0\_REG 0x8223
- #define REG\_DEBUGVIDEO1\_REG 0x8224
- #define REG\_DEBUGVIDEO2\_REG 0x8225
- #define REG\_DEBUGVIDEO3\_REG 0x8226
- #define REG\_DEBUGVIDEO4\_REG 0x8227
- #define REG\_DEBUGVIDEO5\_REG 0x8228
- #define REG\_DEBUGVIDEO6\_REG 0x8229
- #define REG\_DEBUGVIDEO7\_REG 0x822A
- #define REG\_DEBUGVIDEO8\_REG 0x822B
- #define REG\_DEBUGVIDEO9\_REG 0x822C
- #define REG\_DEBUGVIDEO10\_REG 0x822D
- #define REG\_DEBUGVIDEO11\_REG 0x822E
- #define REG\_DEBUGCOUNTER00\_REG 0x822F
- #define REG\_DEBUGCOUNTER01\_REG 0x8230
- #define REG\_DEBUGCOUNTER02\_REG 0x8231
- #define REG\_DEBUGCOUNTER03\_REG 0x8232
- #define REG\_DEBUGCOUNTER04\_REG 0x8233
- #define CMD\_READ\_REG 0x0001 /\* Read Register \*/

### 7.2.1 Detailed Description

#### Author

Stuart B. Wilkins [swilkins@bnl.gov](mailto:swilkins@bnl.gov)

## 7.2.2 LICENSE

Copyright (c) 2014, Brookhaven Science Associates, Brookhaven National Laboratory All rights reserved.

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

1. 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.

## 7.2.3 DESCRIPTION

Control and Frame FPGA Register Map

# Index

- CIN COntrl Timing Routines, [15](#)
- CIN Control Bias Routines, [14](#)
- CIN Data Framestore Functions, [18](#)
  - cin\_data\_framestore\_disable, [18](#)
  - cin\_data\_framestore\_skip, [18](#)
  - cin\_data\_framestore\_trigger, [19](#)
  - cin\_data\_framestore\_trigger\_enable, [19](#)
  - cin\_data\_get\_framestore\_counter, [19](#)
- CIN Data Initialization Routines, [16](#)
  - cin\_data\_init, [16](#)
  - cin\_data\_stop\_threads, [17](#)
- Cin Control Initialization Routines, [9](#)
  - cin\_ctl\_destroy, [9](#)
  - cin\_ctl\_init, [9](#)
- Cin Control Read/Rwrite Routines, [11](#)
  - cin\_ctl\_read, [11](#)
  - cin\_ctl\_stream\_write, [11](#)
  - cin\_ctl\_write, [12](#)
  - cin\_ctl\_write\_with\_readback, [12](#)
- cin\_ctl, [21](#)
- cin\_ctl\_config, [21](#)
- cin\_ctl\_destroy
  - Cin Control Initialization Routines, [9](#)
- cin\_ctl\_id, [22](#)
- cin\_ctl\_init
  - Cin Control Initialization Routines, [9](#)
- cin\_ctl\_listener, [22](#)
- cin\_ctl\_pwr\_mon\_t, [22](#)
- cin\_ctl\_pwr\_val, [23](#)
- cin\_ctl\_read
  - Cin Control Read/Rwrite Routines, [11](#)
- cin\_ctl\_stream\_write
  - Cin Control Read/Rwrite Routines, [11](#)
- cin\_ctl\_write
  - Cin Control Read/Rwrite Routines, [12](#)
- cin\_ctl\_write\_with\_readback
  - Cin Control Read/Rwrite Routines, [12](#)
- cin\_data, [23](#)
- cin\_data\_callbacks, [23](#)
- cin\_data\_frame, [24](#)
- cin\_data\_framestore\_disable
  - CIN Data Framestore Functions, [18](#)
- cin\_data\_framestore\_skip
  - CIN Data Framestore Functions, [18](#)
- cin\_data\_framestore\_trigger
  - CIN Data Framestore Functions, [19](#)
- cin\_data\_framestore\_trigger\_enable
  - CIN Data Framestore Functions, [19](#)
- cin\_data\_get\_framestore\_counter
  - CIN Data Framestore Functions, [19](#)
- cin\_data\_init
  - CIN Data Initialization Routines, [16](#)
- cin\_data\_packet, [24](#)
- cin\_data\_proc, [24](#)
- cin\_data\_stats, [25](#)
- cin\_data\_stop\_threads
  - CIN Data Initialization Routines, [17](#)
- cin\_data\_threads, [25](#)
- cin\_map\_t, [25](#)
- cin\_port, [26](#)
- descramble\_map\_t, [26](#)
- fifo, [26](#)
- src/cin.h, [27](#)
- src/cin\_register\_map.h, [33](#)