## **General Notice**

When using this document, keep the following in mind:

- 1. This document is confidential. By accepting this document you acknowledge that you are bound by the terms set forth in the non-disclosure and confidentiality agreement signed separately and /in the possession of SEGA. If you have not signed such a non-disclosure agreement, please contact SEGA immediately and return this document to SEGA.
- 2. This document may include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new versions of the document. SEGA may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time.
- 3. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without SEGA'S written permission. Request for copies of this document and for technical information about SEGA products must be made to your authorized SEGA Technical Services representative.
- 4. No license is granted by implication or otherwise under any patents, copyrights, trademarks, or other intellectual property rights of SEGA Enterprises, Ltd., SEGA of America, Inc., or any third party.
- 5. Software, circuitry, and other examples described herein are meant merely to indicate the characteristics and performance of SEGA's products. SEGA assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples describe herein.
- 6. It is possible that this document may contain reference to, or information about, SEGA products (development hardware/software) or services that are not provided in countries other than Japan. Such references/information must not be construed to mean that SEGA intends to provide such SEGA products or services in countries other than Japan. Any reference of a SEGA licensed product/program in this document is not intended to state or simply that you can use only SEGA's licensed products/programs. Any functionally equivalent hardware/software can be used instead.
- 7. SEGA will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's equipment, or programs according to this document.

NOTE: A reader's comment/correction form is provided with this document. Please address comments to:

SEGA of America, Inc., Developer Technical Support (att. Evelyn Merritt) 150 Shoreline Drive, Redwood City, CA 94065

SEGA may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.



# System Library User's Guide

Doc. # ST-162-062094

CONTENTS
System Program User's Manual 1
SMPC I/F User's Manual 7
CD Communication Interface User's Manual 21

# READER CORRECTION/COMMENT SHEET

## Keep us updated!

If you should come across any incorrect or outdated information while reading through the attached document, or come up with any questions or comments, please let us know so that we can make the required changes in subsequent revisions. Simply fill out all information below and return this form to the Developer Technical Support Manager at the address below. Please make more copies of this form if more space is needed. Thank you.

Genera	al Info	orma	tion:		
Your Na	ame _				Phone
Documo	ent nu	mber	ST-162-062094		Date
Documo	ent naı	me _	System Library User's Guide		
Correc	ctions	<b>:</b> :			
Chpt.	pg.‡	# C	Correction		
Questio	ons/co	mme	nts:		
			Where to send y	our corre	ections:
	F	ax:	(415) 802-3963 Attn: Manager, Developer Technical Support	Mail:	SEGA OF AMERICA Attn: Manager, Developer Technical Support

275 Shoreline Dr. Ste 500 Redwood City, CA 94065

# **REFERENCES**

In translating/creating this document, certain technical words and/or phrases were interpreted with the assistance of the technical literature listed below.

- 1. KenKyusha New Japanese-English Dictionary 1974 Edition
- 2. *Nelson's Japanese-English Character Dictionary* 2nd revised version
- 3. Microsoft Computer Dictionary
- 4. *Japanese-English Computer Terms Dictionary* Nichigai Associates 4th version

(This page is blank in the original Japanese document.)

# **Table of Contents**

SYS	TEM I	PROGRAM USER'S MANUAL	1
1.0	GUI	DE	1
	1.1	Explanation	
2.0	REF	ERENCE	3
2.0	2.1	List of Functions	
	2.2	Function Specifications	
SMI	PC I/F	USER'S MANUAL	7
1.0		DE	
1.0	1.1	Objective	
	1.2	Overview	
	1.3	System Management System	
	1.4	Peripheral Control System	
2.0	REF	ERENCE	11
	2.1	Data Specification List	
	2.2	Data Specifications	11
	2.3	List of Function Specifications	
	2.4	Function Specifications	
CD	COMI	MUNICATION INTERFACE USER'S MANUAL	21
1.0		ERVIEW	
1.0	1.1	Library Configuration	
	1.2	System Configuration	22
	1.3	Functions and Characteristics	23
2.0	BAS	IC FACTS	24
	2.1	Terminology and Symbols	24
	2.2	Disk Layout	
3.0	CON	MMUNICATION WITH THE CD BLOCK	27
	3.1	Communication Register	
	3.2	Communication Method	
	3.3	Response	
	3.4	Data Transfer	

4.0	CD	DRIVE	33
	4.1	CD Drive Status Change	33
	4.2	CD Drive Operation	
	4.3	Subcodes	
5.0	CD	BLOCK CONFIGURATION	41
	5.1	Data Flow	41
	5.2	Stream Process Mechanism	42
	5.3	Selector Configuration	43
	5.4	Sector Data Format	48
	5.5	CD Block Initialization	49
6.0	CD	BLOCK FILE SYSTEM	50
	6.1	File Management in CD-ROM (ISO9660)	50
	6.2	CD Block File System Functions	
7.0	DAT	TA SPECIFICATIONS	55
	7.1	Data List	
	7.2	Data Detailed	
8.0	FUN	NCTION SPECIFICATIONS	73
	8.1	List of Functions	74
	8.2	Function Details	76
Inde	v		104

Pages 1-20 of this document have been archived. The revised pages of the "System Program User's Manual" and the "SMPC I/F User's Manual" can be found in ST-162-R1-092994 entitled "SATURN System Library User's Guide ver.1.0".

# CD Communication Interface User's Manual

## 1.0 OVERVIEW

This document is a collection of CD block functions and CD communication interface external specifications. Functions of CD blocks are classified in Table 1.1 below.

Table 1.1 Classification of CD Block Functions

Classification	Description
CD Part	Communication format, CD drive related, CD buffer related
	CD block file system related
MPEG Part	MPEG Related

CD block functions can be used by a common software interface through the CD communication interface.

## 1.1 Library Configuration

The library configuration of CD relationships is shown in Figure 1.1.

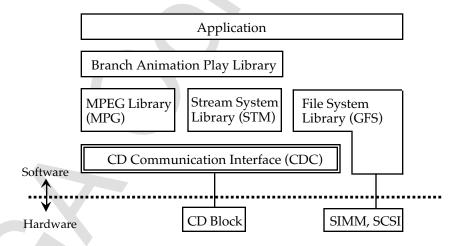
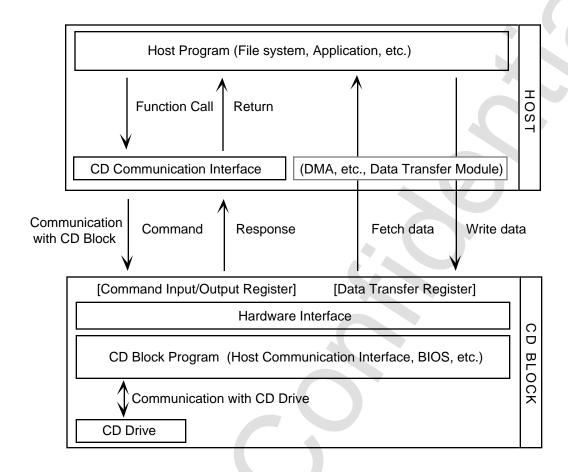


Figure 1.1 Library Structure of CD Relationships

## 1.2 System Configuration

The system configuration of CD relationships is shown in Figure 1.2.



In interfacing with the CD block, the main system is called the host. Moreover, The CD block is one subsystem.

Figure 1.2 System Structure of CD Relationships

## 1) Communication with the CD Block

The host and CD block communicate (command/response, data transfer) using registers on the hardware interface. A command issued from the host to the CD block is called a CD block command; however, this is normally shortened to *command*.

#### 2) CD Drive

The CD block periodically communicates with the CD drive controlling the CD drive with respect to commands from the host. The CD drive accommodates the CD-ROM XA standard and multi-session. It is able to switch between the standard speed as well as double the standard speed. (Standard speed is automatically set when playing CD-DA.)



## 1.3 Functions and Characteristics

Numbers "1" to "3" below are functions of the CD communication interface, and "4" to "7" are CD block functions.

#### 1. CD Block Communication Functions

- (a) Communicates between the host and CD block and is the C Language interface function set.
- (b) Issues commands to the CD block and receives the response.
- (c) Corresponds one to one with the CD block command.

## 2. Data Transfer Auxiliary Functions

Supports auxiliary functions concerning data transfer such as data transfer setup and termination as well as data transfer register address acquisition. Data that is fetched and written in the CD block is sent through the CD block data transfer register. The transfer itself is not through the CD communication interface but is through the DMA data transfer module. (Software transfer, SH-2 DMA, and SCU DMA can be done through the CPU.)

## 3. Register Access Function

Supports functions that access the CD block register.

## 4. CD Drive Function

Playing music (CD-DA) and sector read (CD-ROM) are handled by a common command such as CD play. Stop and pause CD play are seek commands. Resume CD play (cancel pause) is executed by the CD play command.

#### 5. Subcode R ~ W Decode Function

Subcode R ~ W is decoded when playing CD-DA. There is a pack buffer of 23 pack segments for storing decoded pack data.

## 6. Stream Select Function

CD-ROM XA standard sectors and the stream of every type of device (host, MPEG, etc.) are processed uniformly.

- (a) There are 200 sectors (about 460 Kbytes) of CD buffer for storing streams.
- (b) There are 24 independent selectors for selecting (separating, storing, fetching) streams.

## 7. CD Block File System Function (ISO9660)

Directory information is held in the CD block so that files that conform to the ISO9660 standard can be easily accessed.

# 2.0 BASIC FACTS

# 2.1 Terminology and Symbols

## Definition of **T**erms

Terms used are defined in Table 2.1 below.

Table 2.1 Definition of **€**rms

Term	Meaning
Sector	Basic units of data handled by a CD block (2352 bytes fixed). The sector size and configuration follows the CD-ROM format.
Stream	Distinguished by sector headers and subheaders. Flow of logically continuous data.
CD Buffer	Buffer within the CD block that stores sector data.
CD Buffer Size	Size of CD buffer sector units.
Buffer Partition	Segments CD buffer into multiple logical partitions. (Can be considered logical elements getting and storing streams.)
Buffer Partition Size	Size of buffer partition sector units.
Sector Location	Sector locations within buffer partition (location of sector units). Takes the value of 0 to (buffer partition size -1). Sectors of the buffer partition key-in the sector location and accesses.
Filter	Logical element that separates streams by set conditions.
Selector	Composed from the filter and buffer partition, logical element that selects streams. The selector enables the construction of a stream select circuit.
Device	Logical apparatus that causes and absorbs streams such as a CD-ROM and MPEG. The stream select process is enabled by connecting a device to the circuit mentioned above.
Connector	Terminal for connecting a filter, buffer partition, or device.
Frame Address (FAD)	Continually adds numbers in frame units with the absolute time 00:00:00 on the CD as 0. Corresponds 1 to 1 with absolute time. The CD block accesses the frame address as a key and not absolute time. (both CD-ROM, CD-DA)
Logic Sector No. (LSN)	Continually adds numbers in sector (frame) units with the absolute time 00:02:00 on the CD as 0. Logic sector number = frame address - 150 (2 seconds). Used for information within the directory code (ISO9660).
File Information	Directory record information maintained for accessing files. There are the file lead FAD, file size, unit size, gap size, file number, and attributes.
File Identifier	Directly sequence number for identifying files. Takes a value of 0 to (directory record number -1).
	0: Own directory, 1: Parent directory.



## List of Symbols and Codes

Symbols and codes used are defined in Table 2.1 below.

Table 2.2 List of Symbols and Codes

Symbol or Code	Meaning	Description
Adr	address	address
BCD	binary coded decimal	binary decimal number
bn.bufno	buffer no.	buffer partition number
bufnum	buffer numbers	total number of buffer partitions
CI	coding information	coding information
CN	channel no.	channel number
Ctrl	control	control
dst	destination	copy and move destination
fad	frame address	frame address
fasnum	fad sector numbers	frame address sector numbers
fid	file identifier	file identifier
fln, filtno	filter no.	filter number
FN	file no.	file number
LSB	least significant bit	least significant bit
MSB	most significant bit	most significant bit
ply	play parameter	play parameters
pos	position parameter	position parameters
SM	submode	submode
sct	sector	sector
ses	session information	session information
sesno	session no.	session number
snum	sector numbers	(when specifying sector range of partition)
sp. spos	sector position	(when specifying sector range of partition)
src	source	copy / move source
stat	CD status information	CD status information
subh	subheader condition	subheader condition for filter
TNO	track no.	track number (music number)
toc	TOC information	TOC information
txwnum	transfer word numbers	(data transfer size in word units)
X, idx	index	index number
word	word	units 2 bytes (16 bits) in length

## **Numeric Notation**

B is added to the end of a binary number. (Ex: 5 is 101B.)

H is added to the end of a hexadecimal number. (Ex: 255 is FFH.)

A hexadecimal number in a C language program is noted by C language specification (0x).

## **Handling BCD**

BCD should not be used in parameters. Numbers recorded as BCD on a CD track number, for instance, are all converted to binary values.

## C Language Identifier

Use identifiers that begin with CD or Cd in the CD communication interface.

# 2.2 Disk Layout

## **Track Layout**

Figure 2.1 shows the relationship of the track configuration with the access key that indicates the position on the disk.

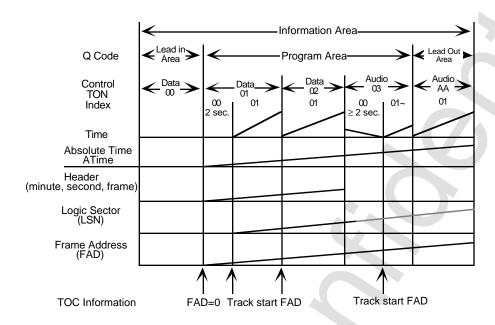


Figure 2.1 Relationship of rack Configuration and ccess Key

#### Multi Session Layout

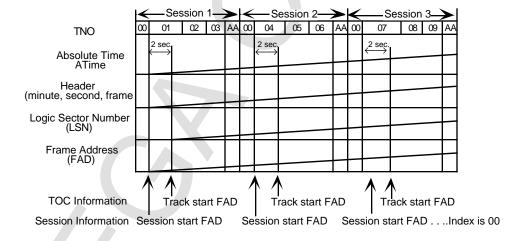


Figure 2.2 Multi Session Layout



## 3.0 COMMUNICATION WITH THE CD BLOCK

## 3.1 Communication Register

Table 3.1 shows the CD block and communication registers and corresponding access functions.

Table 3.1 Communication Registers

Register Name	R/W	Description	Access Function
DATATRNS	R/W	Data Transfer Register	CDC_GetDataPtr
HIRQREQ	R/W	Interrupt Factor Register	CDC_GetHirqReq, CDC_CIrHirqReq
HIRQMSK	R/W	Interrupt Mask Register	CDC_GetHirqMsk, CDC_SetHirqMsk
MPEGRGB	R	MPEG Register (RGB data)	CDC_GetMpegPtr

<sup>\*</sup> All access widths are 16 bits (1 word)

See below for more information about the communication register.

In the figure, a reserve bit is shown by " - " The reserve bit must be specified with a 0. The bit configuration is 15 for the most significant bit and 0 for the least significant bit.

## **Data Transfer Register (DATATRNS)**

When transferring data, this register inputs and outputs data between the host and CD block. The inner part of the register is FIFO.

Register Name	R/W	15	14	13	12	11	10	9	8
DATATRNS	R/W								

7	6	5	4	3	2	1	0

## Interrupt Factor Register (HIRQREQ)

This is the interrupt factor register from the CD block to the host.

Register Name	R/W	15	14	13	12	11	10	9	8
HIRQREQ	R/W		-	*	*	*	SCDQ	EFLS	ECPY
									_
						-			
		7	6	5	4	3	2	1	0

CMOK (bit 0): Issueable command 1 = command can be issued (response set complete)

DRDY (bit 1) : Data transfer ready 1 = data transfer setup is complete.

CSCT (bit 2) : 1 sector stored 1 = 1 CD-ROM sector is stored or discarded in partition

BFUL (bit 3) : CD buffer full 1 = CD buffer is full

PEND (bit 4) : CD play end 1 = CD play has ended (current FAD is outside play range)

DCHG (bit 5) : Disk exchange 1 = Disk has been exchanged (tray has been opened)

ESEL (bit 6) : Selector set end 1 = soft reset, stop execution of selector set EHST (bit 7) : Host I/O end 1 = stop execution of host input/output

ECPY (bit 8) : Copy end 1 = stop execution of copy and move between buffer

partition

EFLS (bit 9) : File system end 1 = stop execution of CD block file system

(CD drive communication timing)

**★** bit 11 ~ bit 13 : MPEG related interrupt factor (see MPEG part for the meaning of each bit)

(a) Bit write can only be done at 0 (clear), not at 1.

(b) The IRQ output to the host is OR output of all factors. Clears corresponding factor bits during interrupt process.

(c) Names the command end flag of each bit ESEL, EHST, ECPY, EFLS. Reports the end of execution of commands belonging to each flag.

## Interrupt Mask Register (HIRQMSK)

This register is for masking interrupt from the CD block to the host.

Register Name	R/W	15	14	13	12	11	10	9	8
HIRQMSK	R/W	1	l	*	*	*	SCDQ	EFLS	ECPY
			Y						
		7	6	5	4	3	2	1	0
		EHST	ESEL	DCHG	PEND	BFUL	CSCT	DRDY	CMOK

(a) Bit content is the same as the HIRQREQ register.

(1 = interrupt enable, 0 = same mask)

- (b) This occurs if interrupt is masked (corresponding bit = 0) when interrupt occurs.
  - not reflected in IRQ output.
  - reflected in the HIRQREQ register. (Polling process possible)



## MPEG Register (MPEGRGB)

This register is for transferring in the RGB format MPEG frame buffer image data that is in the RGB format to the host.

Register Name	R/W	15	14	13	12	11	10	9	, 8
MPEGRGB	R	*	*	*	*	*	*	*	*
		7	6	5	4	3	2	1	0

**★** bit 0 ~ bit 15 : RGB data (See MPEG part for the meaning of each bit.)

## 3.2 Communication Method

#### **Communication Time**

All interrupts are prohibited during the response time of command/response. Response time may be delayed when commands are continuously issued.

#### Communication Error

A communication time out error occurs when the CD block does not respond (command is not returned) within the prescribed time after a command is issued.

#### **Command Issue Provisions**

Command issue must obey the following provisions.

- (a) A maximum of 30 commands can be issued within 6.7 ms.
- (b) A maximum of 60 commands can be issued within 16.7 ms.
- (c) When acquiring a subcode, a maximum of 34 commands can be issued within 13.3 ms. (subcode R~W is 16 commands + subcode Q is 1 command)
- (d) Commands belonging to the same command end flag wait for execution to end and are issued.

# <Program Guide>

The polling process by continuously issuing commands is prohibited.

Regular response and the interrupt factor register are used in the polling of condition changes. Use of timing through the CSCT and SCDQ flags, and VSYNC issues commands.

#### **Command End Flag**

The end of execution of each command listed below is posted.

Table 3.2 Commands belonging to the Command End Flag

Flag	Command					
ESEL	Set CD device connection source	Set filter frame address range				
	Set filter subheader conditions	Set filter mode				
	Set filter connection destination	Reset selector				
	Calculate actual data size	<ul> <li>Execute frame address find</li> </ul>				
	Set sector length					
EHST	Fetch sector data	Cancel sector data				
	Cancel sector data fetch	Write sector data				
ECPY	Copy sector data	Move sector data				
EFLS	Move directory	Save file information				
	Read file					

- (a) The ESEL flag also posts the end of soft reset execution.
- (b) The EFLS flag is 1 when opening the tray and when stopping file access.

#### Command End Flag Process in CD Communication Interface

When commands belonging to the command end flag are issued, the CD communication interface function does the following.

- (a) Checks the command end flag, and if 0, does not issue a command.
- (b) If 1, the command end flag is cleared to 0 and a command is issued.

The command end flag is processed when the following commands are issued.

Soft reset
 Clears ESEL flag to 0 (no check)

• Gets actual data size Checks ESEL flag (not cleared to 0)

Gets frame address find results

Checks ESEL flag (not cleared to 0)

Stop file access Clears EFLS flag to 0 (no check)



## 3.3 Response

## **Response Contents**

Response is composed of a 1 byte status and 7 byte report. The status are common but the report depends on the command. This most basic of reports is the CD report.

#### **Status**

The following information returns to status.

• REJECT (improper command) : Command itself is invalid due to

mistake in command format.

• WAIT (hold execute) : Cannot be executed under current

conditions, therefore command can

not be received.

• Data transfer request : Shows that data transfer is necessary.

• Periodic response : Shows whether a periodic response or

response to a command.

• CD drive condition : Shows the logical condition of the

CD drive seen from the host.

(a) Response during REJECT is invalid.

(b) Status during WAIT and CD report are effective but other reports are ineffective.

## Periodic Response

Periodic response is a response that returns with the communication timing of the CD block with the CD drive. The host can obtain information (status and CD report) without issuing a command. It is periodically updated by the communication cycle with the CD drive. (Same as the SCDQ flag update timing.)

Periodic response update cycle

At standard playing speed : 13.3 ms

At double playing speed : 6.7 ms

When not playing : 16.7 ms

Update is not done during command/response. After response is read by command, it is updated according to the next CD drive communication timing.

#### 3.4 Data Transfer

#### **Data Transfer Procedure**

When using commands to transfer data, the host must do the data transfer process after the command/response. This procedure is listed below.

- (a) The command needed for transferring data is executed (data fetch/write).
- (b) Data transfer setup wait is executed. (Wait until the DRDY flag is 1).
- (c) Data is input and output through the data transfer register (DATATRNS). Get Data transfer register pointer is executed when getting the address.
- (d) Data transfer end is executed when the necessary data is transferred.

#### **Data Transfer End**

When there is a data transfer request by status, end data transfer must be executed even when no data is actually transferred. This is not necessary for REJECT and WAIT. When an error occurs with the data transfer setup wait, the application does not need to execute data transfer end. Execution is automatically done within the setup wait function.

#### **Data Transfer Errors**

Table 3.3 shows errors that occur while waiting for the setup of data transfer.

Table 3.3 Data Transfer Errors

Data Transfer Error	Description
Data transfer time out	CD block does not complete data transfer setup within the prescribed time.
	(DRDY flag remains at 0 and does not change to 1)
Write sectors cannot be secured	Empty sectors could not be reserved by writing sector data.
	(With the DRDY flag at 0 the EHST flag is 1.)



## 4.0 CD DRIVE

## 4.1 CD Drive Status Change

#### **CD Drive Status**

Table 4.1 CD Drive Status

Status	Explanation
<busy></busy>	Status change in progress
<pause></pause>	Temporarily stopped
<standby></standby>	Drive stop status
<play></play>	CD play in progress
<seek></seek>	Seeking
<scan></scan>	Scanning
<open></open>	Tray is open
<nodisc></nodisc>	There is no disc
<retry></retry>	Read retry in progress
<error></error>	Read data error occurred
<fatal></fatal>	Fatal error occurred (hard reset required)

## **Types of Drive Commands**

Table 4.2 shows the types of commands (drive commands) that change the CD drive status.

Table 4.2 Types of Drive Commands

Drive Commands	Corresponding Functions
Initialize CD block	CDC_CdInit
Open tray	CDC_CdOpen
Play	CDC_CdPlay
Seek	CDC_CdSeek
Pause	CDC_CdSeekd
Stop	CDC_CdSeek
Scan	CDC_CdScan

- (a) When doing a soft reset, CD block initialization does not operate as a drive command.
- (b) Drive commands other than tray open can be done concurrently with the tray close command.

## **Issuing Drive Commands**

- (a) Drive command response must return the <BUSY> status as long as there is no REJECT.
- (b) As a rule, command issued afterwards are issued by priority.

## 4.1.1 Illustrations of CD Drive Status Change

1) Status Change Illustration of a Normal System. Figure 4.1 illustrates the status change of a CD drive in a normal system.

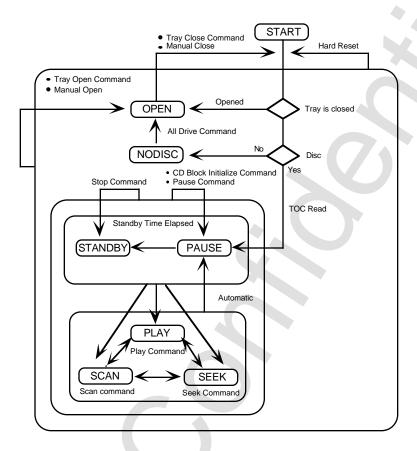


Figure 4.1 Illustration of CD Drive Status Change (Normal System)

- (a) <PAUSE> occurs two seconds after TOC read at frame 0 (FAD=150=96H).
- (b) The <BUSY> status indicates that a status change is in progress (on arrow line).
- (c) The tray close command specifies drive commands other than tray open. The status corresponding to every command changes after the tray is closed.

**Example:** Play command in the <OPEN> status changes to the <PLAY> status after the tray is closed.



2) Status Change Illustration of an Error System. Figure 4.2 illustrates the status change of a CD drive in an error system. Statuses are explained in Table 4.3.

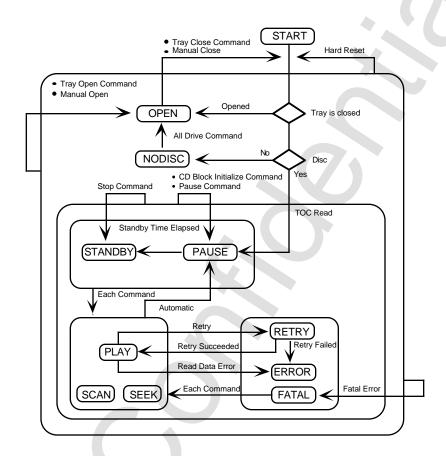


Figure 4.2 Illustration of CD Drive Status Change (Error System)

Table 4.3 Error Relationship Status

Status	Explanation	Internal Status
<retry></retry>	<play> if retry succeeds and <error> if retry fails.</error></play>	<seek></seek>
<error></error>	Status does not change until the next drive command is issued.	<pause></pause>
<fatal></fatal>	Stops the disc motor rotation. Hard reset is required.	<standby></standby>

Internal conditions indicate equal conditions as CD block internal operations.

#### 4.1.2 Table of CD Drive Status Changes

Table 4.4 lists status changes of the CD drive. For example, if the pause command is issued during <PLAY>, <PAUSE> will be displayed.

Table 4.4 Table of CD Drive Status Changes

Operation		Commands						
& Status	Automatic	Initialize	Tray Open	Play	Seek	Pause	Stop	Scan
<busy></busy>	changes	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<pause></pause>	<standby></standby>	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<standby></standby>	_	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<play></play>	<pause></pause>	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<seek></seek>	<pause></pause>	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<scan></scan>	<pause></pause>	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<open></open>	_	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<nodisc></nodisc>	_	<open></open>	<open></open>	<open></open>	<open></open>	<open></open>	<open></open>	<open></open>
<retry></retry>	changes	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<error></error>	_	<pause></pause>	<open></open>	<play></play>	<seek></seek>	<pause></pause>	<standby></standby>	<scan></scan>
<fatal></fatal>	_	undefined	undefined	undefined	undefined	undefined	undefined	undefine

<sup>&</sup>quot;-" indicates no change.

1) Commands issued in the <BUSY> status.

Commands issued in the <BUSY> status can be received but there is noguarantee of immediate execution. For example, While the tray is being opened or closed, a command is not executed until that operation is over. (However, the tray open command is executed immediately.)

# 2) Tray Open/Close.

- (a) As a rule, the tray open command is executed immediately in any mode.
- (b) The tray open/close command when not using the automatic open/close format is <BUSY> until executed manually.
- (c) Commands in the <OPEN> status (except for tray open) change to each status after the tray close process.
- (d) When the tray is closed, <NODISC> occurs if TOC read cannot be done. (Even if there is a disc).

# 3) Status Routings.

- (a) Everything goes though the <BUSY> status when done by command.
- (b) In some cases the routing is by way of <SEEK> when changing to the <PLAY> and <SCAN> statuses.
- 4) In the case of Multi Sessions.
  - (a) Reads the final session of TOC.
  - (b) After TOC is read, the <PAUSE> status is at the 0 frame, 2 seconds from the start position of the final session.



## 4.2 CD Drive Operation

#### <PLAY> Status FrameAddress

The frame address (currently FAD) posted while a CD is playing indicates the sector during the read process. The current FAD sector is not stored in the CD buffer and cannot be retrieved. The host can access up to one sector before the current FAD. (In the case of a CD-ROM).

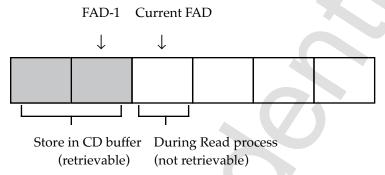


Figure 4.3 Sectors that the current FAD indicates

When play has ended, FAD = play end position + 1. (The read out area is indicated when the play end position concludes at the end of the disc.)

## Change from <PLX> Status and Sector Storage

When changing from <PLAY> to a different status, whether the sector during the read process can be store is undefined. If changing to any status other than <BUSY> the sector that should be stored is defined. When the pause command is issued in the <PLAY> status and the status changes from <PLAY> to <PAUSE>, the storage sector + 1 FAD is posted the same as with 1).

#### Play Range and FrameAddress

By changing play end, seek, and the play range, <PAUSE> occurs when FAD is outside the play range and the PEND flag of the interrupt factor register becomes 1. When FAD is outside the play range, the <PAUSE> status cannot be canceled (restart play). Canceling <PAUSE> when FAD is within the play range changes the status to <PLAY>.

#### **CD Play Repeat Process**

Both the repeat post frequency  $(0H\sim EH)$  and maximum indication frequency  $(0H\sim FH)$  are shown by 4 bits. After the frame of the play end position is played (FAD = play end position + 1), the following process is performed.

- (a) If the repeat frequency is less than maximum repeat frequency, CD play will be repeated (return to the start position and continue playing). If the repeat frequency is <EH(14 times)> at that time, the repeat frequency is increased by 1.
- (b) If the repeat frequency is greater than or equal to the maximum repeat frequency, CD play will not repeat (play end position + 1 and pause).

If the play range or maximum repeat frequency changes, the repeat frequency clears to 0. Neither the repeat frequency nor play range rely on the tray open/close or the seek operation while play is in progress.

## Seek the Home Position (Stop)

The following operations are done by seeking the home position.

- (a) The disc motor rotation stops and pickup is moved to the standby position inside the circumference.
- (b) The CD drive status changes to <STANDBY> and report is an invalid value (FFH line).
- (c) When the status changes from <STANDBY> in the home position to <PAUSE>, seek moves to the start of the disc.
- (d) The play range, maximum repeat frequency, and repeat notification frequency which have been saved cannot be changed.

## Pickup Position in the <SANDBY> status

• Moving from the <PAUSE> status: Current position (report also remains

unchanged).

• When seeking the home position: Standby position inside the circumference

(report is an invalid value).

#### CD Read in a Full CD Buffer

If the CD buffer becomes full, a <PAUSE> will occur and the BFUL flag of the interrupt factor register becomes 1. When there is space in the CD buffer, play automatically starts where it left off.



#### 4.3 Subcodes

## Subcode Update and SCDQ Flag

The subcode Q is updated for each frame (one sector). The SCDQ flag becomes one every time subcode Q is updated (same as update timing of periodic response). Subcode R~W can update four packs (one packet) with one frame and be stored in the pack buffer. There is a 23 pack (short of six packets) segment in the CD block.

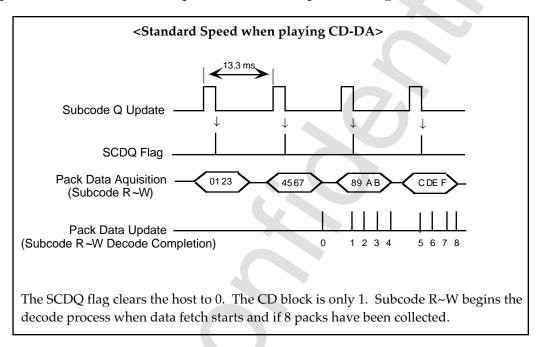


Figure 4.4 Subcode Update and SCDQ Flaginfling

#### Subcode Get Tming

The host issues the Get command by a timing with the SCDQ flag at 1 and gets subcode Q and R~W data. (Polling is prevented by continuously issuing commands.)

#### Getting Subcode Q

The Get command can be executed any time regardless of the CD drive status. But when in the <OPEN> status, ineffective values may return depending on the CD drive status.

## Getting Subcode R~W

Pack data reissues the Get command and gets data until the pack buffer is empty. However, exceeding 16 packs within the time of one frame (13.3 ms) is not allowed. WAIT is returned if the pack buffer is empty and pack data disappears.

#### Subcode R~W Decode

## (a) Decode start and end

Decode On/Off are set by the initialize command of the CD block. To start decoding, set decode to ON then execute CD-DA play.

## (b) Decode Start Timing

Start decode during <PLAY>.

Data actually begins to be retrieved two frames before <PLAY>.

## (c) Pack Buffer Clear Timing

Clear is done when decode begins.

The pack buffer content is saved even when doing seek and pause.

## (d) Decode Conditions

Decode is done only when playing CD-DA.

Decode is not done in other cases (while scanning or playing the CD- ROM area)

## Subcode R~W Error

## (a) Over Run Error

If a pack is not fetched in time, the pack buffer is overwritten and an over run error occurs. After data transfer has ended, the get pointer advances to the extent that the pack is without an overrun error. The host must finish transferring data within the decode time of 1 pack (3.2 ms).

## (b) Pack Data Error

Pack data checks CRC and corrects the data if an error is detected. A pack data error occurs if the data cannot be corrected.



## **5.0** CD BLOCK CONFIGURATION

The characteristics and logical configuration of the CD block as seen from the host is shown below.

#### **Data Format**

The basic unit of the stream handled by the CD buffer is a fixed size (2352 bytes) sector no matter what the device is.

#### Stream Select Circuit

The functions that separate and store streams are attained by a filter and buffer partition design. These functions are considered to be logical elements (selector). The circuit that selects the necessary streams is constructed by combining selectors.

#### Device

The CD and MPEG are considered to be logical devices that produce and absorb streams. The flow of a stream is controlled by connecting a device to the stream select circuit.

#### 5.1 Data Flow

Figure 5.1 shows the entire data flow for a CD block.

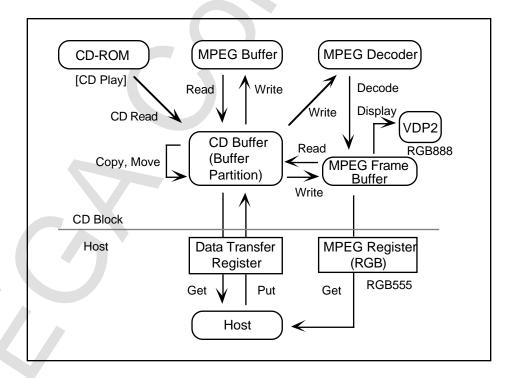


Figure 5.1 CD Block Entire Data Flow

## 5.2 Stream Process Mechanism

Figure 5.2 shows the overall configuration of the CD block from the viewpoint of stream flow.

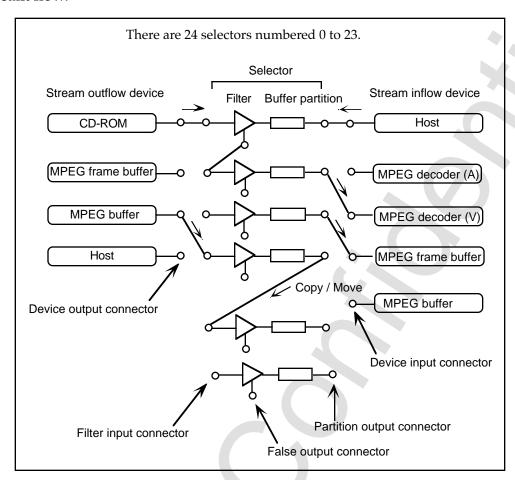


Figure 5.2 CD Block Overall Configuration

[What each part does]

- (a) Device: Produces and absorbs streams (stream inflow and outflow device.)
- (b) Selector: Composed of a filter and buffer partition, used for selecting streams.
- (c) Filter: Separates streams by set conditions (conforms to Yes/No conditions.)
- (d) Buffer Partition: Stores streams, releases or cancels data through outside requests.

There are connectors for connecting every device to a selector. The device and selector input/output connectors are 1 to 1. Connections can be made between selectors as well.

Connections flow regularly in streams by connecting connectors that don't pass through partitions and stop when entering a partition. Sectors that are output from an output connector that is not connected will be canceled.

Devices must be connected through a selector. A device output connector can only connect to a filter input connector, and a device input connector can only connect to a partition output connector.

## 5.3 Selector Configuration

The selector process function does not rely on the parts connection destination or device. It separates and stores the input stream and only outputs by request. Figure 5.3 shows the selector configuration.

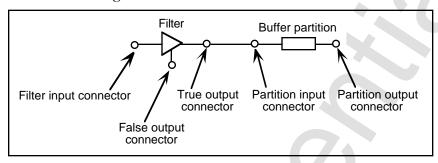


Figure 5.3 Selector Configuration (Initial condition)

In the initial condition of the selector, the filter and partition with the same number are connected through the true output connector and partition input connector. Other connectors are not connected.

#### 5.3.1 Filter

Conditions (FAD range, subheader) of the sector that should pass are set in the filter. Sectors that match with the conditions are output to the true output connector and stored in buffer partition of the connection destination. All other sectors are output to false output connectors.

False output connectors can be connected to other filter input connectors and achieve in secession similar selection processes. Sectors output from unconnected output connectors will be deleted.

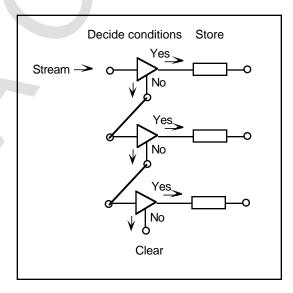


Figure 5.4 Type of Stream Select Process

#### 5.3.2 Buffer Partition

1) Buffer Partition Configuration

The CD buffer is divided into several buffer partitions. Figure 5.5 shows the configuration of a buffer partition.

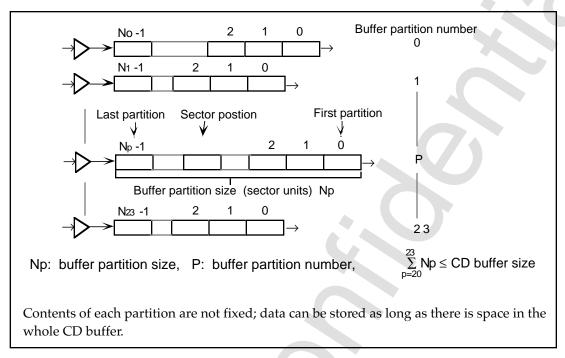


Figure 5.5 Buffer Partition Configuration

- 2) Storing Sectors to the Buffer Partition Sectors entered in partitions are stored in the last partition. The last sector position of the partition is equal to the buffer partition size minus 1.
- 3) Discharging and Canceling Sectors from the Buffer Partition
  Two ways to discharge sectors from partitions are to keep the sectors at the
  partition discharge source or clear the sectors (clear only can also be selected.)
  For example, get to the host corresponds to "Get" and "Get and Clear."
  Flowing in and out between selectors corresponds to "Copy" and "Move."
- 4) CD 1 Sector Interrupt and Store Destination
  One sector interrupt in CD read occurs in each sector at the time the sector is
  stored in the partition. The CD block stores the partition number of the store
  destination. The store destination partition number of the sector read last can
  be retrieved from the host.



#### 5.3.3 Connectors

Connection from Buffer Partition to Filter (Sector Data Copy/Move)
 The partition output connector is connected to the filter input connector and the flow of sector data allows sector data to be copied and moved. The connection and flow I/O process are executed by executing one command.

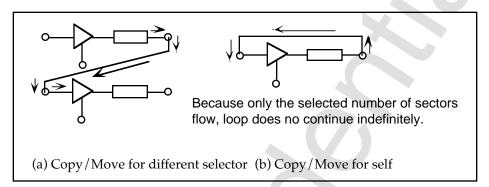


Figure 5.6 Copy/Move Sector Data

Selector Connections by OR Condition (multi to one connection)
 True output connectors of several filters can be connected to the same partition input connector. This enables a sector to be stored in whichever of several conditions it coincides with.
 (OR condition: logical add).

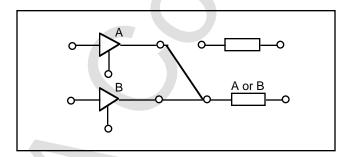


Figure 5.7 Selector Connection by OR Conditions

## 3) Types of Connectors and Connection Destination

Table 5.1 Types of Connectors and Connection Destination

Output/Input	Filter Input	Partition Input	Device Input
Device Output	0	X	X
True Output	X	Δ	X
False Output	0	X	X
Partition Output	0	Х	0

O: connectable (1 to 1), only 1 output connector can connect

to 1 input connector.

 $\Delta$ : connectable (multi to 1), several output connectors can

connect to the same input connector.

(OR condition)

X: not connectable

Whether to do a simple process of connecting (disconnecting) by one command or a complex process collecting connections and flow in/out processes depends on each device. The CD-ROM and MPEG decoder are single processes. Others (host, copy/move, etc.) are complex processes.



#### 5.3.4 Selector Precautions

1) Error in the Stream Process

Except with the CD device, error in the stream process occurs in the following cases.

- Connection separates during operation.
- CD buffer becomes full while reading or writing to the buffer partition.
- 2) Selector Set Effective Timing

FAD with selector settings made effective while a CD is playing is within one FAD from response of the selector set command. The host considers the margin and issues the set command before 10 or more sectors of the target FAD. When the selector setting has ended, the ESEL flag of theinterrupt factor register becomes 1.

3) Sector Data when Changing Connections

When the CD is being read, data is not omitted even if the connection changes during the sector data sequence. In short, the flow of data is guaranteed when temporarily interrupted while switching connections.

- 4) General Access to the Buffer Partition (general command specifying sector position and sector range)
  - Returns to WAIT when the specified sector range doesn't exist like the data number within the partition is exceeded. Always returns to WAIT when the data number within the partition is 0.
- 5) Clear Discharge from Buffer Partition

When a sector is cleared and discharged from a partition, all data of the selected sector range is cleared (get and clear or move sector data) even if the process stops because of an error etc.

#### 5.4 Sector Data Format

Basic Format
 Sector data format basically conforms to the CD-ROM XA sector format.

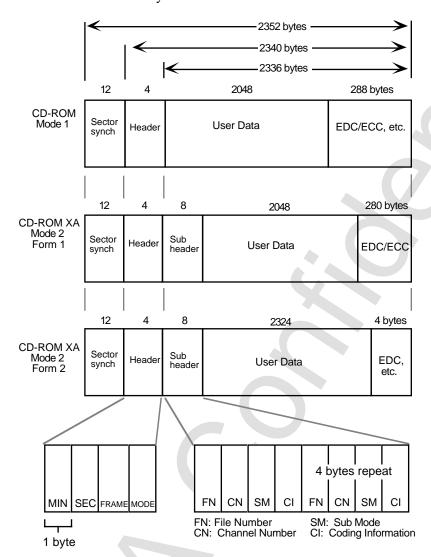


Figure 5.8 CD-ROM XA Sector Format

- 2) Handling Sub header and User Data (2048 bytes)
  - (a) Sub header except for Mode 2 (mode part within header is 02H) is handled as 0.
  - (b) There is user data immediately after the header only for Mode 1 (mode part within header is 01H). Otherwise, user data is considered to exist in the same position as mode 2 form 1.
  - (c) Storing user data within a buffer partition, except data from a CD device, is the same as mode 2 form 1. The first 24 bytes is 0, and data after the user data is undefined.



#### 5.5 CD Block Initialization

#### **TOC/Session Information**

Initialized when the tray is opened, and information is in a non-input condition. Reads TOC when tray is closed (unrelated to the reset of software.)

#### File Information

Initialized if tray is opened or software is reset, and information is in a non-input condition. The host explicitly commands the reading of file information.

#### **Host Information**

Initialized if software is reset (unrelated to opening the tray.)

[Host Information] ..... mostly host set information and buffer data

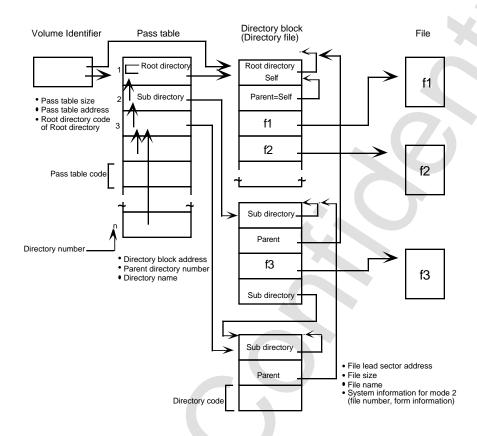
- Initialization information (CD block initialization parameters)
- Play information (play range, maximum number of repeats, current repeat notification number)
- Selector information (filter, buffer partition, etc., host setting for selector)
- Data within buffer, buffer partition size, space size
- Actual data size retained, setting of sector length
- Data transfer control register (initializing ends transfer of data)
- MPEG related information

If the CD block is initialized without resetting the software, information in the TOC/ Session and Host is not initialized.

#### 6.0 CD BLOCK FILE SYSTEM

### 6.1 File Management in CD-ROM (ISO9660)

The CD block file system supports a logic format that conforms to ISO9660.



Volume identifier: Arranged from CD 00:02:16 (FAD = 166 = A6H)
 Pass table: A collection of pass table records. Pass table size and

address are recorded in the volume identifier.

• Pass table record : Corresponds to one directory (root, sub)

Record position is the directory number  $(1 \sim n)$ Directory information is listed in the directory book.

A collection of directory records. A directory block is

considered to be one file. (directory file)

• Directory record : Corresponds to one file (includes directory).

Figure 6.1 File Management Data Configuration in CD-ROM (ISO9660)



Directory block :

## 6.2 CD Block File System Functions

The CD block file system (referred to "file system" hereafter) has the following functions.

- Moves directories and holds file information
   Moves to the specified directory and holds in the CD block file information
   within that directory.
- Gets file information
   Transfers held file information in response to a request from the host.
- 3) Reads files Reads file sector data originating from held file information to the CD buffer. The host is able to extract read sector data by data transfer.
- 4) Stop Accessing Files CD play stops by moving a directory, holding file information, and reading a file. If file access is stopped, the CD block pauses the CD drive.

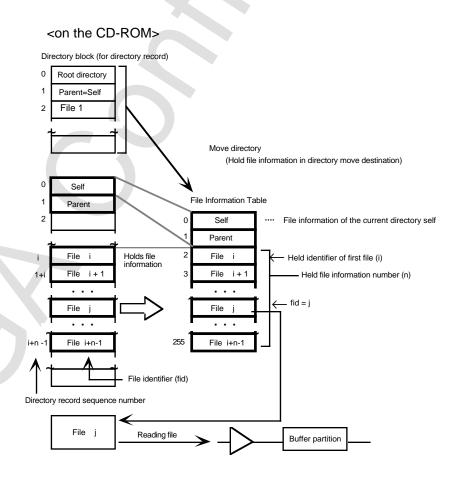


Figure 6.2 CD Block File System Configuration

#### 6.2.1 File Information Holder Method

The CD block provides 256 file information tables that hold file information for accessing. The first two parts in this table always holds current directory (self) and parent directory information. Up to 254 parts of ordinary file information can be held. (The number cannot be designated.)

## 1) Directory Move

If a directory is moved, the directory record is read, and file information is extracted and held in a table. The range that is held is from the start of the directory block.

2) Holding File Information (specifying the hold range)
Specifies a file identifier and can hold file information from any position within the current directory. In this case also, self and parent information is held in the first two parts of the table.

#### 6.2.2 File Access Procedure

### 1) Moving to the Root Directory

In using the file system, first move to the root directory and create a file information table. This is the same when file information is cleared when starting up or changing disks. Neither a table can be created nor the file system used if the disk is not in the ISO9660 format. (REJECT occurs if getting the holding file information range.)

### 2) Holding File Information

The file system can access only files inside the file information table. When accessing a file, the file information must be held beforehand. When moving directories, the target directory must be held.

### 3) Reading Files

File identifier and offset (sector units) are selected and the file read. The read destination selector is selected by the filter number.

## 4) Getting Sector Data

Files read to the buffer partition transfer and fetch data by fetching sector data. Data can be fetched while reading ,and there is no waiting for read to end. Pause occurs if the CD buffer becomes full while reading files. If a space can be created, pause is canceled and the remaining files read.

## 5) Ending File Read

When file read has ended, the EFLS flag of the interrupt factor register becomes "1". Executing the file access stop command stops file read. (Accessing a file is done in the same by moving a directory and holding file information.)



#### 6.2.3 File System Precautions

### 1) Work Selector Designation

When reading file information from the CD-ROM, the CD block requires a work selector. One selector can be designated to move a directory and hold file information. At least one empty sector is required at this time. The selector used is canceled after the process ends.

### 2) Selector Settings

When accessing to a file, the file system does the following settings for the selector.

(a) Connection conditions of indicated filter

• filter input connector : connects with a CD device.

• true output connector : connects with buffer partition with the same

number.

• false output connector : cuts off other selectors.

#### (b) Filter conditions

Table 6.1 Set Filter Condition

Action	Filter Condition	
Move Directory	Sets filter frame address range (FAD range)	
Hold File Information	Makes effective the frame address range of the filter mode	
	Sets filter frame address range.	
Read Files	Sets the file number of the filter subheader conditions	
	Makes effective the filter mode FAD range and file number selection	

# (c) Filter and connected buffer partition

The buffer partition sectors are cleared before files are accessed.

# 3) CD Play Range and Drive Command

When accessing a file, the play range becomes the default value (disk first to last). The drive command (CD play, pause, etc.) is executed unchanged even with the file system in operation. Be aware that operation of the file system can be obstructed. Auto pause/cancel-pause operate normally when the buffer is full.

## 4) File Access End and Stop

When file access ends or is stopped, the EFLS flag becomes 1. File access also stops if the tray is open. (But before the <OPEN> status, EFLS = 1.) While directory move and file information hold are being executed, the file information table is destroyed if file access stops.

5) Multi Session In multi session cases, the file system refers to the volume descriptor of the last session.



# 7.0 DATA SPECIFICATIONS

This chapter lists the constants, data types, access macros, etc. used by the CD communication interface.

#### Format

In the figures, the reserved regions are shown by the "-" mark. A 0 must be specified in the reserve regions.

#### 7.1 Data List

Table 7.1 is the CD communication interface data list.

Table 7.1 Data List

	Data	Data Name	Number
Ва	asic data		1.0
Co	onstants		2.0
TO	OC information & Session information		3.0
St	atus and Report		4.0
Sı	ubcode information		5.0
Data Types			6.0
	CD communication function return code	CdcRet	6.1
	CD status information	CdcStat	6.2
	Hardware information	CdcHw	6.3
	CD position parameters	CdcPos	6.4
	CD play parameters	CdcPly	6.5
	Sub header conditions	CdcSubh	6.6
	Sector information	CdcSct	6.7
	File information	CdcFile	6.8

# 7.2 Data Detailed

## 7.2.1 Basic Data

Title	Data	Data Name	No
Data Specification	Basic Data		1.0

# 1) Basic Data Types

Type Name	Description
Uint 8	1 byte integer without code
Sint 8	1 byte integer with code
Uint 16	2 byte integer without code
Sint 16	2 byte integer with code
Uint 32	4 byte integer without code
Sint 32	4 byte integer with code
Boo 1	logical type 4 byte integer (assumes logic constant as value)

# 2) Logic Constants Used as a logic (Bool) value.

Constant Name	Value	Description	
FALSE	0	Indicates that logic value is false	
TRUE	1	Indicates that logic value is true	
OFF	0	Indicates switch is off (false)	
ON	1	Indicates switch is on (true)	

# 3) Additional

Constant Name	Value	Description
NULL	((void *)0)	NULL pointer



#### 7.2.2 Constants

Title	Data	Data Name	No
Data Specification	Constants		2.0

1) Bit Names of interrupt factor register relationship (HIRQREQ, HIRQMSK).

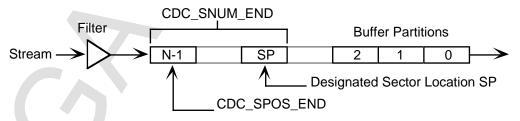
Constant Name	Description
CDC_HIRQ_CMOK	Enable commands to be issued
CDC_HIRQ_DRDY	Prepare data transfer
CDC_HIRQ_CSCT	Store 1 sector
CDC_HIRQ_BFUL	CD buffer is full
CDC_HIRQ_PEND	CD play end
CDC_HIRQ_DCHG	Disk change
CDC_HIRQ_ESEL	End Selector set
CDC_HIRQ_EHST	End host input/output
CDC_HIRQ_ECPY	End copy
CDC_HIRQ_EFLS	End file system
CDC_HIRQ_SCDQ	Subcode Q

2) Special designation of buffer partition sector range (sector location and sector number).

Constant Name	Description	
CDC_SPOS_END	Indicates sector location of last partition	
	Same as (sector no. in partition - 1) designated	
CDC_SNUM_END	Sector number from designated sector location SP to last partition. Same as designated (sector no. in partition - SP)	

Effective with functions that correspond to the buffer partitions below.

CDC\_CalActSiz, CDC\_ExeFadSearch, CDC\_GetSctInfo, CDC\_GetSctData, CDC\_DelSctData, CDC\_GetdelSctData, CDC\_CopySctData, CDC\_MoveSctData



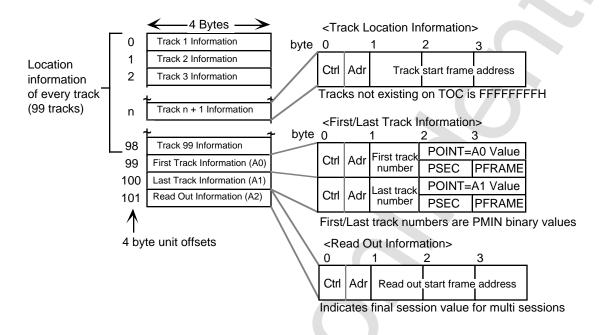
## 3) Other

<b>Constant Name</b>	Description
CDC_NUL_SEL	Special value of selector no. (filter number, buffer partition number)
CDC_NUL_FID	Special value of file identifier

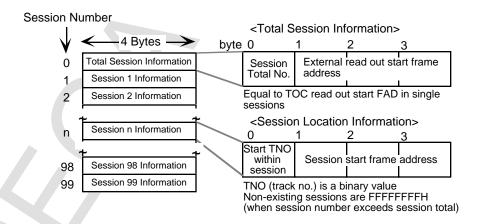
#### 7.2.3 TOC and Session Information

Title	Data	Data Name	No
Data Specification	TOC and Session Information		3.0

#### 1) TOC Information



#### 2) Session Information



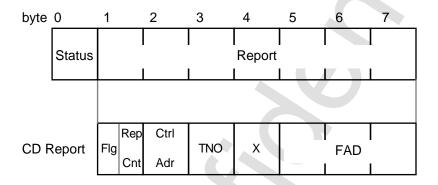


#### 7.2.4 Status and Report

Title	Data	Data Name	No
Data Specification	Status and Report (1/2)		4.0

### 1) Response Format

Command response is in the format below. Whether CD report returns as a report depends on the command. Periodic response absolutely returns CD report.

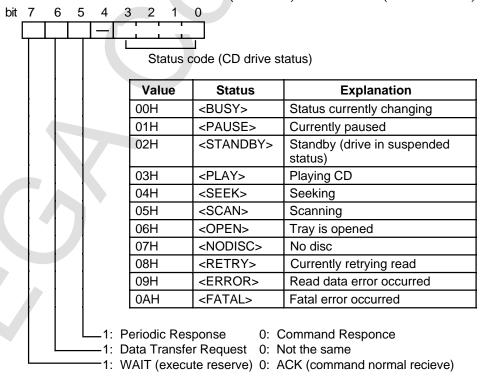


Combined status and CD reports are called CD status information.

#### 2) Status

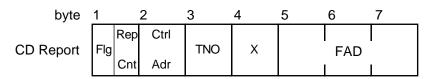
A status of FFH indicates REJECT (abnormal command)

The status code is valid when WAIT (bit 7 = 1). Other bits (bit 4 to bit 6) are 0.

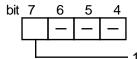


Title	Data	Data Name	No
Data Specification	Status and Report (2/2)		4.0

## 3) CD Report



FLG: 4 bit flag



1 : CD-ROM 0 : CD-DA (or when seeking or scanning)

Rep Cnt : 4 bit repeat frequency. Notification range 0H ~ EH (0 ~ 14 times)

Ctrl Adr : CONTROL/ADR byte of subcode Q

TNO : Track number of subcode Q (binary value, not BCD)
 X : Index number of subcode Q (binary value, not BCD)
 FAD : Frame address (in header information when CD-ROM,

otherwise, based on subcode Q)

4) CD Report content in every status

CD Report is invalid for REJECT and valid for WAIT.

CD drive status and CD report contents are shown below. The FFH rows mean invalid values.

Status	CD flag/repeat	Control/ADR	Track No.	Index No.	Frame address
<busy></busy>	O/FFH	O/FFH	O/FFH	O/FFH	O/FFFFFH
<pause></pause>	0	0	0	0	0
<standby></standby>	value at pause	value at pause	value at pause	value at pause	value at pause
(when seek home)	FFH	FFH	FFH	FFH	FFFFFFH
<play></play>	0	0	0	0	0
<seek></seek>	0	target position	target position	target position	target position
<scan></scan>	0	0	0	0	0
<open></open>	FFH	FFH	FFH	FFH	FFFFFFH
<nodisc></nodisc>	FFH	FFH	FFH	FFH	FFFFFFH
<retry></retry>	0	0	0	0	0
<error></error>	FFH	FFH	FFH	FFH	FFFFFFH
<fatal></fatal>	undefined	undefined	undefined	undefined	undefined

O: Current value. Seek Home: seek (stop) against home position. When the standby time expires and the status changes from <PAUSE> to <STANDBY>, the value when paused is saved.



- 5) When CD report content is an ineffective value
  - (a) The value immediately before the repeat frequency is saved. If changing to a different status, returns to the retaind value.
  - (b) Moves to the start (FAD = 150) of the disk if the status changes from Seek Home to <PAUSE>.
  - (c) The value immediately before <ERROR> is retained. If changing to the <PAUSE> status, returns to the retained value.
  - (d) Midway during the <BUSY> status, the ineffective value returns in some cases when changing to a different status.

#### 7.2.5 Subcode Information

Title	Data	Data Name	No
Data Specification	Subcode Information		5.0

#### 1) Subcode Q Information

#### (a) When Adr = 1

byte	1	2	3	4	5	6	7	8	9	10	
	Ctrl										
		TNO	Х		P_FAD	)	00	(	$Q_{F}AE$	)	CRC
	Adr										

Ctrl Adr : CONTROL/ADR byte

TNO : Track No. (binary value not BCD)X : Index No. (binary value not BCD)

P FAD : Time within track (frame address format with track lead at 0)

Q\_FAD : Absolute time (frame address format with 00:00:00 at 0)

The content in each status is the same as for the CD report. Values in read in and read out are shown in the table below. (Can be acquired by multi-session)

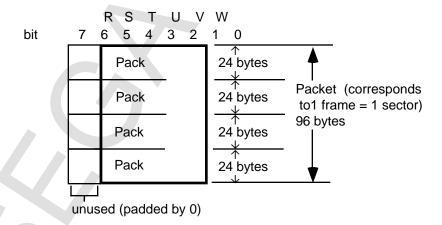
TNO	X	P_FAD
01H ~ 63H (1 ~ 99)	01H ~ 63H	Elapsed FAD within track
AAH (Read Out)	01H ~ 63H	Elapsed FAD within track
00H (Read In)	01H ~ 63H, A0H, A1H, A2H	000000H

Read in of the first session is Q FAD = 000000H.

(Normally cannot be acquired).

(b) In other cases: the subcode Q value unchanged (10 bytes except CRC)

#### 2) Subcode R~W Information



Subcode R~W is decoded in 1 pack units (24 bytes = 12 words). The CD block has an internal pack buffer of 23 packs (short of 6 packets).



#### 7.2.6 Data Type

Title	Data	Data Name	No
Data Specification	CD Communication Function Return Code	CdcRet	6.1

## 1) Return Code

CdcRet ret ..... Be sure ret is not the pointer.

Access Macro	Type	Explanation
CDC_RET_ERR (ret)	Sint 16	Error Code
CDC_RET_STATUS (ret)	Uint 8	Status

#### 2) Error Code

Constant Name	Explanation
CDC_ERR_OK	Normal (process continuation)
CDC_ERR_CMDBUSY	Command end flag is not 1
CDC_ERR_CMDNG	CMOK flag is not 1 when command is issued
CDC_ERR_TMOUT	Time out (wait for response, wait for data transfer ready)
CDC_ERR_PUT	Data transfer read wait by sector data write. Could not reserve empty sectors.
CDC_ERR_REJECT	REJECT for command response
CDC_ERR_WAIT	WAIT for command response
CDC_ERR_TRNS	Abnormal data transfer size
CDC_ERR_PERI	Not a periodic response
CDC_ERR_PTYPE	Abnormal play, seek position type

## 3) Status

The retaining status of the CD communication interface is substituted in the return code status together with all functions. Response status returns when response is acquired by acquiring periodic response or issuing commands. The holding status returns unchanged when a response cannot be acquired by not issuing a command or error.

Title	Data	Data Name	No	
Data Specification	CD Status Information	CdcStat	6.2	

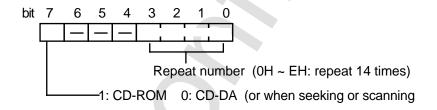
Combine the CD block status and CD report to create CD status information.

## 1) CD Status Information

CdcStat \*stat

Access Macro	Туре	Explanation
CDC_STAT_STATUS (stat)	Uint 8	Status
CDC_STAT_FLGREP (stat)	Uint 8	CD report CD flag and repeat no.
CDC_STAT_CTLADR (stat)	Uint 8	CD report CONTROL/ADR byte
CDC_STAT_TNO (stat)	Uint 8	CD report track no. (binary value)
CDC_STAT_IDX (stat)	Uint 8	CD report index no. (binary value)
CDC_STAT_FAD (stat)	Sint 32	CD report frame address

# 2) CD Report CD flag and Repeat Count



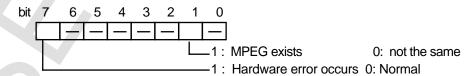
Title	Data	Data Name	No
Data Specification	Hardware Information	CdcHw	6.3

## 1) Hardware Information

CdcHw \*hw

Access Macro	Туре	Explanation
CDC_HW_HFLAG (hw)	Uint 8	Hardware flag
CDC_HW_VER (hw)	Uint 8	CD block version information
CDC_HW_MPVER (hw)	Uint 8	MPEG version information
CDC_HW_DRV (hw)	Uint 8	CD drive information
CDC_HW_REV (hw)	Uint 8	CD block revision information

# 2) Hardware Flag





Title	Data	Data Name	No
Data Specification	CD Position Parameters (1/2)	CdcPos	6.4

Designates the CD range of play (start and end positions) and is used in designating the seek position.

## 1) Position Type

Default value, frame address, track/index, and no change can be designated at position parameters. The constants (position type) below are used to select the parameter.

Constant Name	Explanation	
CDC_PTYPE_DFL	Designates default values	
	For CD play: indicates disk start and end.	
	For seek: indicates home position.	
CDC_PTYPE_FAD	Designates frame address	
CDC_PTYPE_TNO	Designates track/index	
CDC_PTYPE_NOCHG	Designates no change (no change in set values)	

Beginning of disc: FAD = 150(96H) position (2 seconds, 0 frame with absolute time) End of disc: FAD = readout start frame address (TOC information) - 1 position.

## 2) Position Paramters

### CdcPos \*pos

Access Macro	Туре	Explanation
CDC_POS_PTYPE (pos)	Sint 32	Position type
CDC_POS_FAD (pos)	Sint 32	Frame address, FAD sector no.
CDC_POS_TNO (pos)	Uint 8	Track number
CDC_POS_IDX (pos)	Uint 8	Index number

## 3) Setting Method

(a) Designating the default value

(b) Designating the frame address

$$CDC_POS_PTYPE$$
 (&pos) =  $CDC_PTYPE_FAD$ ;

CDC POS FAD (&pos) = fad; 
$$/*$$
 frame address  $*/$ 

The end position is designated by sector number (FAD sector

number) from the starting FAD

$$CDC_POS_PTYPE(\&pos) = CDC_PTYPE_TNO;$$

CDC POS IDX (&pos) = 
$$x$$
; /\* index number \*/

(d) Designating No Change

Title	Data	Data Name	No
Data Specification	CD Position Parameters (2/2)	CdcPos	6.4

## 4) Execptions to Frame Address

The frame address is explained by the conditions below.

Condition	Start and Seek Positions	End Position
FAD < disc start	FAD = disc start (150)	FAD = 150+FAD sector no 1 (when starting FAD <150)
FAD > disc end	FAD = disc end + 1 (equal to read out)	FAD = disc end

Relationship of start and end positions:

End FAD = Start FAD + FAD sector number - 1 (The held contents of the play range are the Start FAD and End FAD.)

## 5) Execptions to Track/Index

The track/index is explained by the conditions below.

Condition	Start and Seek Positions	End Position
TNO = 0 (track default value)	TNO = disc start track	TNO = disc last track
X = 0 (index default value)	X =1	X = 99(63H)
TNO < disc start track TNO = disc start track		TNO = disc start track
	X = 1 (same as $X = 0$ )	X = 99 (same as $X = 0$ )
TNO > disc end track	TNO = disc end track	TNO = disc end track
	X = 1 (same as $X = 0$ )	X = 99 (same as $X = 0$ )
X does not exist	(from the next track)	(until track ends)

TNO: track number, X: index number

X = 0 means designation of the track only. (track start/last index) TNO = X = 0 is the default value of the position parameter. (Start position: start of disc, End position: end of disc, Seek postion: home postion)

# 6) Exceptions to the Play Range

The play range is retained if the end postion is smaller than the start position, but the CD will not play. (For designating both frame address and track/index)

# 7) Meanings of Special Designations

• Specifies seek position default value: used as a stop command.

(seeks home position)

• Specifies seek position no change: used as a pause command.

Specifies play range no change: used with pause cancel (retry CD play).

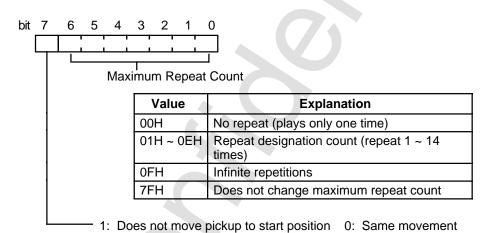


Title	Data	Data Name	No
Data Specification	CD Play Parameters	CdcPly	6.5

Used to designate the play range and play mode when playing a CD.

### 1) Play Mode

When playing a CD, the maximum repeat count of the play segment and the existence of pickup movement (play position) to the start position are designated. The designation range of the maximum repeat count is 4 bits. The default value is 0.



CDC\_PM\_DFL Play mode default value (no repeat, moves pickup)

CDC\_PM\_REP\_NOCHG No change in maximum repeat count

CDC\_PM\_PIC\_NOMOV Does not move pickup to start position
(<PAUSE> status when current position is outside play range)

CDC\_PM\_NOCHG Play mode does not change (can change play range only)

Equal to (CDC\_PM\_REP\_NOCHG: CDC\_PM\_PIC\_NOMOV)

# 2) Play Parameters

# CdcPly \*ply

	Access Macro	Туре	Explanation
CI	DC_PLY_START (ply)	CdcPos	Play start position parameters
	CDC_PLY_STYPE (ply)	Sint 32	Start position type
	CDC_PLY_SFAD (ply)	Sint 32	Start position frame address
	CDC_PLY_STNO (ply)	Uint 8	Start position track number
	CDC_PLY_SIDX (ply)	Uint 8	Start position index number
CI	DC_PLY_END (ply)	CdcPos	Play end position parameters
	CDC_PLY_ETYPE (ply)	Sint 32	End position type
	CDC_PLY_EFAS (ply)	Sint 32	End position frame address sector number
	CDC_PLY_ETNO (ply)	Uint 8	End position track number
	CDC_PLY_EIDX (ply)	Uint 8	End position index number
CDC_PLY_PMODE(ply)		Uint 8	Play mode (repeat designation, pickup movement)



- 3) Method for setting play parameters
  Play parameters designate the play range and play mode. Play range is
  designated by a combination of the position parameters of the start and end
  positions.
  - (a) Track designation and frame address designation cannot be combined in the play range. In such cases, REJECT is returned. All other play range combinations are possible.
  - (b) All play range and play mode combinations are possible.
  - (c) The play range and maximum repeat count is held within the CD block and are effective until reset.
  - (d) Default value: start of disc to end of disc, no repeat, moves pickup.

Examples of settings are shown below.

No.	Play Method	Start Position	End Position	Comments
1	Track designation	tno 1, x1	tno 2, x2	
2	Frame address designation	fad	fasnum	note: 1
3	Play from first of disc	CDC_PTYPE_DFL	tno 2, x2 fasnum	
4	Play until end of disc	tno 1, x1 fad	CDC_PTYPE_DFL	
5	End position does not change. (only the start position changes)	tno 1, x1 fad	CDC_PTYPE_NOCHG	note: 2
6	Start position does not change. (only the end position changes)	CDC_PTYPE_NOCHG	tno 2, x2 fasnum	note: 2
7	Play is retried from the current position without changing the play range and play mode.	CDC_PTYPE_NOCHG	CDC_PTYPE_NOCHG	CDC_PM_NOC HG is designated in the Play mode.

<KEY> **fad**: frame address **fasnum**: frame address sector number

tno 1: start track numberx1: start index numberx2: end track numberx2: end index number

**Note 1:** fad + fasnum -1 is retained as the end postion.

**Note 2:** Track and frame address designation cannot be combined even if there is no change on the one hand.

4) Execptions to the Play Range

No play if the end postion is less than the start position. Becomes <PAUSE> status same as play range exterior. The play range is held and the play mode is also effective. (Pickup moves to the start position.)

Title	Data	Data Name	No
Data Specification	Sub header Conditions	CdcSubh	6.6

Used when setting/getting sub header conditions for a filter.

### 1) Sub header Conditions

dcSubh \*subh

Access Macro	Туре	Explanation
CDC_SUBH_FN (subh)	Uint 8	File Number
CDC_SUBH_CN (subh)	Uint 8	Channel Number
CDC_SUBH_SMMSK (subh)	Uint 8	Sub mode mask pattern
CDC_SUBH_SMVAL (subh)	Uint 8	Sub mode comparison value
CDC_SUBH_CIMSK (subh)	Uint 8	Coding information mask pattern
CDC_SUBH_CIVAL (subh)	Uint 8	Coding information comparison value

• Mask pattern : Comparison object bit pattern.

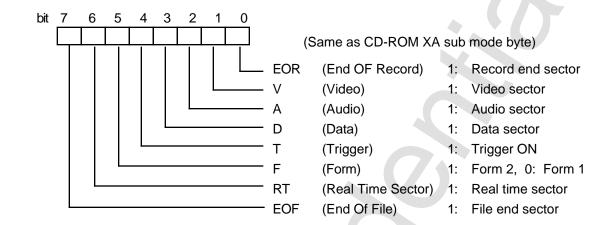
• Comparision Value : Comparison value of the bit designated

by the mask pattern.

Whether the submode and coding information is equal to the comparison value is decided after the logical product of their repective mask patterns are taken.



## 2) Sub Mode



Constant Name	Туре	Explanation
CDC_SM_EOR	01H	Record end sector
CDC_SM_VIDEO	02H	Video sector
CDC_SM_AUDIO	04H	Audio sector
CDC_SM_DATA	08H	Data sector
CDC_SM_TRIG	10H	Trigger bit
CDC_SM_FORM	20H	Form bit (0: Form 1, 1: Form 2)
CDC_SM_RT	40H	Real time sector
CDC_SM_EOF	80H	File end sector

Title	Data	Data Name	No
Data Specification	Sector Information	CdcSct	6.7

#### 1) Sector Information

### CdcSct \*sct

Access Macro	Type	Explanation
CDC_SCT_FAD (sct)	Sint 32	Frame address
CDC_SCT_FN (sct)	Uint 8	File Number
CDC_SCT_CN (sct)	Uint 8	Channel Number
CDC_SCT_SM (sct)	Uint 8	Sub mode
CDC_SCT_CI (sct)	Uint 8	Coding information

Title	Data	Data Name	No
Data Specification	File Information	CdcFile	6.8

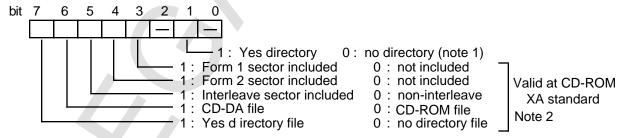
### 1) File Information

#### CdcFile \*file

Access Macro	Туре	Explanation
CDC_FILE_FAD (file)	Sint 32	File start frame address
CDC_FILE_SIZE (file)	Sint 32	File size (byte number)
CDC_FILE_UNIT (file)	Uint 8	File unit size
CDC_FILE_GAP (file)	Uint 8	Gapsize
CDC_FILE_FN (file)	Uint 8	File number *
CDC_FILE_ATR (file)	Uint 8	File attributes

When there is no directory record system information, the file number becomes 0.

#### 2) File Attributes



**Note 1:** bit 1 is a directory bit of the file flag within the directory record and valid when there is no system information (discriminant gives bit 7 priority)

**Note 2:** bits 3 to 7 is attribute information conforming to CD-ROM XA standards, and bits 3 to 7 are 0 when there is no system information.



# 8.0 Function Specifications

Function specifications of the CD Communication Interface are described.

#### 1) Format

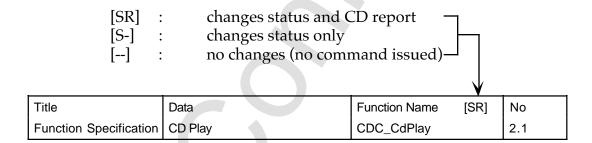
Reserved areas with the figure are indicated by a "-" mark. Reserved areas must be designated with a 0.

### 2) Initial Values of Parameters

Initial values of parameters held in the CD block are equal to the default values designated by the host as long as they are not specially specified.

## 3) CD Status Information (Status and CD Report) Hold

The CD communication interface holds status and CD reports when a CD block command is issued. Information is not held by a periodic response. The intial value of all holding contents is 0. The functions that change the holding status/CD report are listed below by symbol.



Nothing changes when there is an error (When the error code is not CDC\_ERR\_OK).

#### 4) Data Transfer Inside Functions

Among the information get functions, data transferred by the function interior enters the function name as Tget. The method of data transfer is software transfer by the CPU.

**Example:** Get TOC information (CDC\_TgetToc)

These functions call the following functions internally.

• Data transfer waiting for start (CDC\_DataReady)

Data transfer end (CDC\_DataEnd)

# 8.1 List of Functions

Table 8.1 is a list of functions of the CD communication interface.

Table 8.1 List of CD Communication Interface Functions (1)

	Function	Name	Number
CI	D Block Common		1.0
	Get current CD status information	CDC_GetCurStat	1.1
	Get previous CD status information	CDC_GetLastStat	1.2
	Get periodic CD status information	CDC_GetPeriStat	1.3
	Get hardware information	CDC_GetHwInfo	1.4
	Get TOC information	CDC_TgetToc	1.5
	Get session information	CDC_GetSes	1.6
	Initialize CD block	CDC_CdInit	1.7
	Open tray	CDC_CdOpen	1.8
	Data transfer ready	CDC_DataReady	1.9
	Data transfer end	CDC_DataEnd	1.10
CI	D Drive		2.0
	CD play	CDC_CdPlay	2.1
	Seek play position	CDC_CdSeek	2.2
	Scan	CDC_CdScan	2.3
Sı	ubcode		3.0
	Get subcode Q	CDC_TgetScdQch	3.1
	Get Subcode R ~ W	CDC_TgetScdRwch	3.2
CI	D-ROM Device		4.0
	Set connection point of CD device	CDC_CdSetCon	4.1
	Get connection point of CD device	CDC_CdGetCon	4.2
	Get buffer partition of last read sector	CDC_CdGetLastBuf	4.3
Se	elector		5.0
	Set filter frame address range	CDC_SetFiltRange	5.1
	Get filter frame address range	CDC_GetFiltRange	5.2
	Set filter subheader conditions	CDC_SetFiltSubh	5.3
	Get filter subheader conditions	CDC_GetFiltSubh	5.4
	Set filter mode	CDC_SetFiltMode	5.5
	Get filter mode	CDC_GetFiltMode	5.6
	Set filter connection point	CDC_SetFiltCon	5.7
	Get filter connection point	CDC_GetFiltCon	5.8
l	Reset selector (filter, partition)	CDC_ResetSelector	5.9



Table 8.1 List of CD Communication Interface Functions (2)

	Function	Name	Number
Buf	fer Information		6.0
	Get CD buffer size	CDC_GetBufSiz	6.1
	Get buffer partition sector number	CDC_GetSctNum	6.2
	Calculate actual data size	CDC_CalActSiz	6.3
	Get actual data size	CDC_GetActSiz	6.4
	Get sector information	CDC_GetSctInfo	6.5
	Execute frame address search	CDC_ExeFadSearch	6.6
	Get frame address search results	CDC_GetFadSearch	6.7
Buf	fer Input/Output		7.0
ſ	Set sector length	CDC_SetSctLen	7.1
Ī	Get sector data	CDC_GetSctData	7.2
Ī	Delete sector data	CDC_DelSctData	7.3
Ī	Get and delete sector data	CDC_GetdelSctData	7.4
Ī	Write sector data	CDC_PutSctData	7.5
	Copy sector data	CDC_CopySctData	7.6
Ī	Move sector data	CDC_MoveSctData	7.7
Ī	Get sector data copy/move error	CDC_GetCopyErr	7.8
CD	Block File System		8.0
ſ	Change directory	CDC_ChgDir	8.1
ſ	Hold file information	CDC_ReadDir	8.2
Ī	Get holding file information range	CDC_GetFileScope	8.3
ſ	Get holding file information	CDC_TgetFileInfo	8.4
Γ	Read file	CDC_ReadFile	8.5
Ī	Stop file access	CDC_AbortFile	8.6
Re	gister Access		9.0
Ī	Get data transfer register pointer	CDC_GetDataPtr	9.1
ſ	Get interrupt factor register value	CDC_GetHirqReq	9.2
ſ	Clear interrupt factor register	CDC_ClrHirqReq	9.3
Ī	Get interrupt mask register value	CDC_GetHirqMsk	9.4
	Set interrupt mask register	CDC_SetHirqMsk	9.5
	Get MPEG register pointer	CDC_GetMpegPtr	9.6

#### 8.2 Function Details

#### 8.2.1 CD Block Common

Title	Function	Function Name	[SR]	No.
Function	Get current CD status information	CDC_GetCurStat		1.1
Specifications				

Format: CdcRet CDC\_GetCurStat (CdcStat \*stat)

Input: None

**Output:** stat : CD status information

**Function value:** Returns the return code

**Function:** Issues the get command of CD status information for the

CD block and gets current CD status information (status

and CD report).

Title	Function	Function Name []	No.
Function	Get previous CD status information	CDC_GetLastStat	1.2
Specifications			

Format: CdcRet CDC\_GetLastStat (CdcStat \*stat)

**Input:** None

Output: stat : CD status information

**Function value:** Returns the return code.

**Function:** Gets CD status information for the previous CD block command.

**Remarks:** If the CD communication interface issues a CD block

command, the status and CD report of that response is held.

This function returns the value being held.

Title	Function	Function Name	[]	No.
Function	Get periodic CD status information	CDC_GetPeriStat		1.3
Specifications				

Format: CdcRet CDC\_GetPeriStat (CdcStat \*stat)

Input: None

**Output:** stat : CD status information

**Function value:** Returns the return code.

**Function:** Gets CD status information through periodic response. **Remarks:** Since the command for CD block is not issued, the load is

small and the polling process is possible.



Title	Function	Function Name	[S-]	No.
Function	Get hardware information	CDC_GetHwInfo		1.4
Specifications				

Format: CdcRet CDC\_GetHwInfo (CdcHw \*hw)

**Input:** None

Output: hw : hardware information

**Function value:** Returns the return code.

**Function:** Gets CD block hardware information.

Title	Function	Function Name [S-]	No.
Function	Get TOC information	CDC_TgetToc	1.5
Specifications			

Format: CdcRet CDC\_TgetToc (Uint32 \*toc)

Input: None

Output: toc : TOC information (204 words)

**Function value:** Returns the return code.

**Function:** Gets all TOC information (102 peices, 408 bytes)

Waits in the <OPEN> status and <NODISC> status

**Example:** Uint 32 toc [102]; /\* reserves TOC information store region \*/

ret = CDC\_TgetToc(toc); /\* get TOC information

Title	Function	Function Name	[S-]	No.
Function	Get session information	CDC_GetSes		1.6
Specifications				

Format: CdcRet CDC\_GetSes (Sint32 sesno, Uint32 \*ses)

**Input:** sesno : session number  $(0 \sim 99)$ 

Output: ses : session information (2 words)

**Function value:** Returns the return code.

**Function:** Gets session information (4 bytes) of the designed session.

Waits in the <OPEN> status and <NODISC> status.

Title	Function	Function Name	[SR]	No.
Function	Initialize CDblock (1/2)	CDC_CdInit		1.7
Specifications				

Format: CdcRet CDC\_CdInit (Sint32 iflag, Sint32 stnby, Sint32 ecc,

Sint32 retry)

**Input:** iflag : initializing flag (insignificant 8 bits are effective)

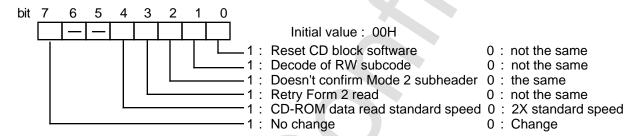
stnby: standby time (insignificant 8 bits are effective)
ecc: ECC frequency (insignificant 8 bits are effective)
retry: retry frequency (insignificant 8 bits are effective)

Output: none

**Function value:** Returns the return code. Function: Initial setting of CD block.

### 1) Initializing flag

Designates each type of setting flag (8 bits) to the CD block. When software is reset, all other initialized parameters are ignored and this becomes the initial value.



- (a) Interrupt factor register ESEL flag becomes 1 when reset of software ends.
- (b) Recognition of the Mode 2 subheader is used only in the correcting process during CD-ROM decode. The selector always recognizes the subheader. (decides conditions of the filter, selector length, etc.)
- (c) Data is output even if there is an error when Form 2 read is not retried.
- (d) CD-ROM data read speed switches when changing to the <PAUSE> status. When the CD-DA area is played, it automatically plays at standard speed. The prior speed returns if the CD-ROM area is entered.

## 2) Standby Time

Designates the transition time from <PAUSE> to <STANDBY>. Changes to <STANDBY> when the standby time of <PAUSE> expires.

Set Value	Description
0000H	180 secondsinitial value
0001H ~ FFFEH	Transition time (in seconds)
FFFFH	No change in settings



Title	Function	Function Name [SR]	No.
Function Specifications	Initialize CD block (2/2)	CDC_CdInit	1.7

## 3) ECC Frequency

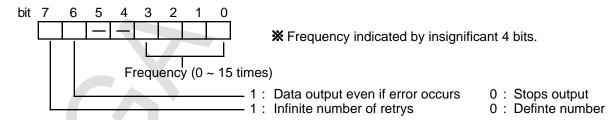
Designates the number of repetitions of the ECC process (PQ once each) during CD-ROM decode.

Set Value	Description	
00H	ECC is processed a maximum of only 1 time in real time····· nitial value.	
01H ~ 05H	When an error exists after the ECC process, maximum number of repetitions (total of 2 to 6 times).	
80H	No ECC process	
FFH	No change in settings	

## 4) Retry Frequency

Designates the number of retries of the same sector if an error occurs during CD-ROM decode.

Set Value	Description	
00H	Without retrying, stops data output (becomes <error>) ····· initial value.</error>	
01H~0FH ( <b>※</b> )	Retries designated number of times, data output stops if there is an error (becomes <error>).</error>	
40H	Data is output without retrying	
41H ~ 4FH ( <b>※</b> )	Retries designated number of times, data is output even if there is an error.	
80H	Infinite number of retries	
FFH	No change in settings	



## 5) Tray Open/Close

Closes the Tray when in the <OPEN> status.

- When automatic format (front loading): CD block automatically closes tray.
- When manual format (top loading): <BUSY> status until closed by manual. Tray open when in the <NODISC> status. (See Tray Open)

Tray open and close operates the same for CD play, seek play position, and scan.

**Remarks:** All command issues are prohibited while the software is being reset.

Title	Function	Function Name [SR]	No.
Function	Open tray	CDC_CdOpen	1.8
Specifications			

Format: CdcRet CDC\_CdOpen (void)

Input: none Output: none

**Function value:** Returns the return code.

**Function:** Stops the CD drive and opens the tray

• When automatic format (front loading): CD block automatically

opens tray.

• When manual format (top loading): <BUSY> status until opened

by manual.

**Remarks:** When the tray is opened, DCHG of the interrupt factor register

and EFLS flag become 1. The timing is before the <OPEN>status.

(Same for when opening manually.)

Title	Function	Function Name [S-]	No.
Function Specifications	Data transfer ready	CDC_DataReady	1.9

Format: CdcRet CDC\_DataReady (Sint32 dtype)

**Input:** dtype : transfer ready type

Output: none

**Function value:** Returns the return code.

**Function:** Waits until data transfer ready is enabled. Data can be transfered

after ready is finished. If data is transfered, data transfer end must be

executed.

### 1) Transfer Ready Type

Value	Description
CDC_DRDY_GET	Designated when getting data. (CD block → host)
CDC_DRDY_PUT	Designated when writing data. (host→ CD block)

### 2) Transfer Ready Process

Waits until DRDY of the interrupt factor register becomes 1, and clears 0 if it becomes 1. If empty sectors can not be reserved when writing, the EHST flag of the interrupt factor register becomes 1. In this case, CDC\_ERR\_PUT error is returned.

**Remarks:** When an error is caused by ready, there is no need to execute data

transfer end. It will be automatically executed inside the ready function.



Title	Function	Function Name	[S-]	No.
Function	Data transfer end	CDC_DataEnd		1.10
Specifications				

Format: CdcRet CDC\_DataEnd (Sint32 \*cdwnum)

**Input:** none

Output: cdwnum : CD block transfer word number

**Function value:** Returns the return code.

**Function:** Instructs the end of data transfer for the CD block. Data transfer

can be stopped while in process.

#### 1) CD Block Transfer Word Number

Returns the word number of effective data transferred by the CD block. The normal word number can not be exceeded (entire number of words that should be transferred).

Constant Name	Description
CDC_DEND_ERR	Error occurred by data transfer
	Data transfer end is executed even though it isn't time for transfer.

### 2) Relationship of Transfer Word Number

Data transfer conditions can be checked by comparing the host transfer word number (word number transfered by the host until data transfer end is executed) and the CD block transfer word number.

Transfer Conditions	Host Transfer Word Number	CD Block Transfer Word Number (cdwnum)		
	(Host Word Number)	When gettng (CD $\rightarrow$ host)	When writing (host $\rightarrow$ CD)	
Transfer interrupt	host word no. < normal word no.	cdwnum > host word number   cdwnum = host word num		
Transfer all	host word no. = normal word no.	cdwnum = normal word number = host word number		
Excess transfer	host word no. > normal word no.	cdwnum = normal word number < host word number		

• Transfer interrupt : Stops data transfer while in progress.

• Transfer all : All data is transfered.

• Excess transfer : Normal word number is exceeded and transfered (Excess part is transfered as

dummy data).

#### 8.2.2 CD Drive

For tray open/close in the <OPEN> and <NODISC> statuses see Initialize CD block.

Title	Function	Function Name	[SR]	No.
Function Specifications	CD play	CDC_CdPlay		2.1

Format: CdcRet CDC\_CdPlay (CdcPly \*ply)
Input: ply : play parameters

Output: none

**Function value:** Returns the return code.

**Function:** Plays CDs according to the designated play parameters.

Plays music in the CD-DA area. Reads sector data in the

CD-ROM area.

**Remarks:** In playing CD-DA, mute is canceled 4 frames prior to the start

position to prevent the start of a song from being cut off.

## **Example:**

(1) Frame Address Designation

```
CdcPly ply;
                                              /* definition of play parameter variable */
                        = CDC_PTYPE_FAD;
CDC_PLY_STYPE (&ply)
                                              /* start frame address
                                                                                 */
CDC_PLY_SFAD (&ply)
                         = fad;
CDC_PLY_ETYPE (&ply)
                         = CDC_PTYPE_FAD;
                                              /* frame address sector number
CDC_PLY_EFAS (&ply)
                         = fasnum;
                                                                                 */
CDC_PLY_PMODE (&ply) = pmode;
                                              /* play mode
                                                                                 */
ret = CDC_CdPlay (&ply);
                                              /* play start
```

(2) Track/Index Designation

```
CDC_PLY_STYPE (&ply) = CDC_PTYPE_TNO;
                                             /* start track number
                                                                              */
CDC_PLY_STNO (&ply)
                        = tno1;
                                                                              */
CDC_PLY_SIDX (&ply)
                                             /* start index number
                        = x1;
CDC_PLY_ETYPE (&ply)
                       = CDC_PTYPE TNO;
                                             /* end track number
                                                                              */
CDC_PLY_ETNO (&ply)
                        = tno2;
CDC _PLY_EIDX (&ply)
                                             /* end index number
                                                                              */
                        = x2;
CDC _PLY_PMODE (&ply) = pmode;
                                             /* play mode
                                                                              */
```

(3) Default Value Designation

```
CDC_PLY_STYPE (&ply) = CDC_PTYPE_DFL; /* start position is first of disc */
CDC_PLY_ETYPE (&ply) = CDC_PTYPE_DFL; /* end position is end of disc */
CDC_PLY_PMODE (&ply) = CDC_PM_DFL; /* no repeat, move pickup */
```

(4) CD Play Restart (unchanged selection: play from current position)

```
···· cancel pause
```

```
CDC_PLY_STYPE (&ply) = CDC_PTYPE_NOCHG;
CDC_PLY_ETYPE (&ply) = CDC_PTYPE_NOCHG;
CDC_PLY_PMODE (&ply) = CDC_PM_NOCHG;
```



Title	Function	Function Name [SR]	No.
Function	Seek play position	CDC_CdSeek	2.2
Specifications			

Format: CdcRet CDC\_CdSeek (CdcPos \*pos)
Input: pos : position parameters

Output: none

**Function value:** Returns the return code.

**Function:** Moves (seeks) the play position (pickup) according to the designated

position parameters.

#### **Example:**

1. Frame Address Designation

CdcPos pos; /\* definition of position parameter variable \*/
CDC\_POS\_PTYPE (&pos) = CDC\_PTYPE\_FAD;
CDC\_POS\_FAD (&pos )= fad; /\* frame address \*/
ret = CDC\_CdSeek (&pos); /\* seek start \*/

2. Track/Index Designation

 CDC\_POS\_PTYPE (&pos)
 =
 CDC\_PTYPE\_TNO;

 CDC\_POS\_TNO (&pos)
 =
 tno; /\* track number
 \*/

 CDC\_POS\_IDX (&pos);
 =
 x; /\* index number
 \*/

- CD Play Stop (Default vlaue designation: Seek home position) Stop CDC\_POS\_PTYPE (&pos) = CDC\_PTYPE\_DFL;
- 4. Pause CD Play (No changed selection: seek current position) ······Pause CDC\_POS\_PTYPE (&pos) = CDC\_PTYPE\_NOCHG;

Title	Function	Function Name [SR]	No.
Function	Scan	CDC_CdScan	2.3
Specifications			

Format: CdcRet CDC\_CdScan (Sint32 scandir)

**Input:** scandir : Scan direction

Output: none

**Function value:** Returns the return code.

**Function:** Scans the CD.

Scanning continues until a CD drive command (CD play, seek, etc.) is

issued. Scan is canceled and a pause occurs if the play range is

exceeded. The pause position at that time is undefined.

#### 1) Scan Direction

Constant Name	Description	
CDC_SCAN_FWD	Fast forward scan (forward direction)	
CDC_SCAN_RVS	Fast reverse scan (reverse direction)	

## 2) Sound Output

If scanning from the <PLAY> status, -12 dB sounds are output in the CD-DA region. Sound is muted if scanning in the CD-ROM area or in the <PAUSE> status. If the scan direction in near the play range boundary is reversed, the sound may not be output.

**Remarks:** When the CD-ROM area is scanned, sector data will not be read.



#### 8.2.3 Subcode

Title	Function	Function Name	[S-]	No.
Function	Get subcode Q	CDC_TgetScdQch		3.1
Specifications				

Format: CdcRet CDC\_TgetScdQch (Uint16 \*qcode)

**Input:** none

Output: qcode : Subcode Q information (5 words)

**Function value:** Returns the return code.

**Function:** Gets subcode Q channel information

(10 bytes other than CRC)

Title	Function	Function Name [S-]	No.
Function	Get subcode R ~ W	CDC_TgetScdRwch	3.2
Specifications			

Format: CdcRet CDC\_TgetScdRwch (Uint16 \*rwcode, Sint32 \*scdflag)

**Input:** none

Output: rwcode : Subcode R ~ W information (12 words)

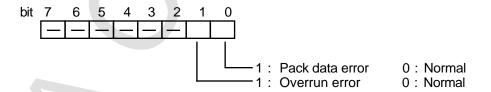
scdflag : Subcode flag (effective bits are insignificant)

**Function value:** Returns the return code.

**Function:** Gets subcode R~W channel information (1 pack of 24 bytes).

# (1) Subcode Flag

Shows error conditions of retrieved subcode R ~ W information (pack data).



## **Remarks:**

Pack data can not exceed retrieval of 16 packs in one time frame (13.3 ms). Returns to WAIT if the pack buffer is empty and there is no pack data.

#### 8.2.4 CD-ROM Device

Title	Function	Function Name	[SR]	No.
Function	Set connection destination of CD device	CDC_CdSetCon		4.1
Specifications				

Format: CdcRet CDC\_CdSetCon (Sint32 filtno)

**Input:** filtno : Connection destination filter number

(CDC\_NUL\_SEL: display non-connection)

Output: none

**Function value:** Returns the return code.

**Function:** Sets the connection destination (filter) of the CD device.

Connection is separated if CDC\_NUL\_SEL is selected.

Title	Function	Function Name [S-]	No.
Function	Get connection destination of CD device	CDC_CdGetCon	4.2
Specifications			

Format: CdcRet CDC\_CdGetCon (Sint32 \*filtno)

**Input:** none

Output: filtno : Connection destination filter number

(CDC\_NUL\_SEL : displays non-connection)

**Function value:** Returns the return code.

**Function:** Gets the connection destination (filter) of the CD device.

Returns CDC\_NUL\_SEL if there is no connection.

Title	Function	Function Name [S-]	No.
Function	Get buffer partition of last read sector	CDC_CdGetLastBuf	4.3
Specifications			

Format: CdcRet CDC\_CdGetLastBuf (Sint32 \*bufno)

**Input:** none

Output: bufno : Storage destination buffer partition number

(CDC\_NUL\_SEL: displays non-storage)

**Function value:** Returns the return code.

**Function:** Gets the buffer partition number of the storage destination of the

sector last read from the CD.



#### 8.2.5 Selector

See selector reset for the initial value of the selector set parameters.

Title	Function	Function Name [SR]	No.
Function Specs	Set filter frame address range	CDC_SetFiltRange	5.1

Format: CdcRet CDC\_SetFiltRange (Sint32 filtno, Sint32 fad,

Sint32 fasnum)

**Input:** filtno : filter number

fad : starting frame address

fasnum : frame address sector number

(0: effective FAD range does not exist)

Output: none

**Function value:** Returns the return code.

**Function:** Sets the frame address range for the filter. When fashum = 0, all

sectors are output to false output connectors if the FAD range of

the filter mode is effective.

Title	Function	Function Name [S-]	No.
Function Spec	Get filter frame address range	CDC_GetFiltRange	5.2

Format: CdcRet CDC\_GetFiltRange (Sint32 filtno, Sint32 \*fad,

Sint32 \*fasnum)

**Input:** filtno : filter number

**Output:** fad : starting frame address

fasnum : frame address sector number

**Function value:** Returns the return code.

**Function:** Gets the frame address range for the filter.

Title	Function	Function Name	[SR]	No.
Function Specs	Set filter subheader conditions	CDC_SetFiltSubh		5.3

Format: CdcRet CDC\_SetFiltSubh (Sint32 filtno, CdcSubh \*subh)

**Input:** filtno : filter number

subh : subheader conditions

Output: none

**Function value:** Returns the return code.

**Function:** Sets subheader conditions for the filter.

Title	Function	Function Name	[S-]	No.
Function Specs	Get filter subheader conditions	CDC_GetFiltSubh		5.4

Format: CdcRet CDC\_GetFiltSubh (Sint32 filtno, CdcSubh \*subh)

**Input:** filtno : filter number

Output: subh : subheader conditions

**Function value:** Returns the return code.

**Function:** Gets subheader conditions for the filter.

Title	Function	Function Name [S	SR]	No.
Function Specs	Set filter mode	CDC_SetFiltMode		5.5

Format: CdcRet CDC\_SetFiltMode (Sint32 filtno, Sint32 fmode)

**Input:** filtno : filter number

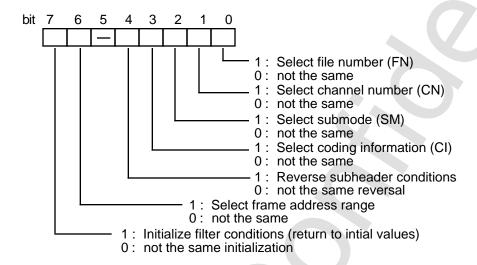
fmode : filter mode (insignificant 8 bits are effective)

Output: none

**Function value:** Returns the return code.

**Function:** Sets the filter mode for the filter.

# 1) Filter mode



## 2) Reverse Subheader Conditions

If subheader conditions are reversed, the output destination will be the opposite of what is normal for subheader conditions. In short, sectors that correspond to subheader conditions are output to false output connectors, and sectors that do not correspond to subheader conditions are output to true output connectors (conditions below.)

• Output to true output connector: filter FAD range coincides and

subheader conditions do not coincide.

• Output to false output connector: filter FAD range does not coincide

or subheader conditions coincide.

# 3) Initialing Filter Conditions

When filter conditions are initialized, other bit selections are ignored and the initial values below are set.

• Frame address range : Starting FAD = 0, FAD sector number = 0

• Subheader conditions: FN, CN, SMMSK, SMVAL, CIMSK, CIVAL are all 0

• Filter mode : All bits = 0



Title	Function	Function Name [S-]	No.
Function	Get filter mode	CDC_GetFiltMode	5.6
Specifications			

Format: CdcRet CDC\_GetFiltMode (Sint32 filtno, Sint32 \*fmode)

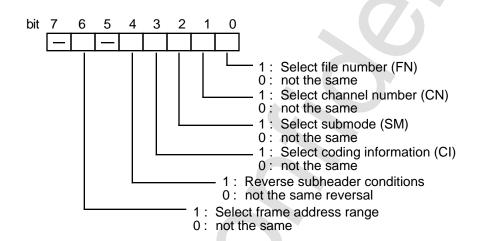
**Input:** filtno : filter number

Output: fmode : filter mode (insignificant 8 bits are effective)

**Function value:** Returns the return code.

**Function:** Gets the filter mode for the filter.

### Filter mode



Title	Function	Function Name	[SR]	No.
Function Specs	Set filter connection destination	CDC_SetFiltCon		5.7

Format: CdcRet CDC\_SetFiltCon (Sint32 filtno, Sint32 cflag, Sint32 bufno, Sint32 flnout)

**Input:** filtno : filter number

cflag : filter connection flag (insignificant 8 bits are effective) bufno : buffer partition no.of true output connector connection

destination (CDC\_NUL\_SEL: disconnected)

flnout : filter number of false output connector connection

destination (CDC\_NUL\_SEL: disconnected)

Output: none

**Function value:** Returns the return code.

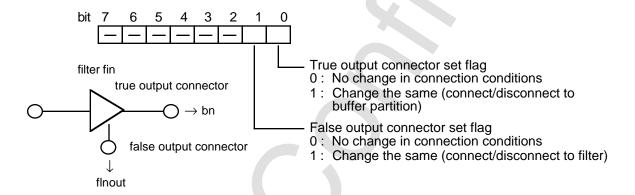
**Function:** Sets the connection destination below for the filter. Connections are

disconnected if CDC\_NUL\_SEL is specified.

• true output connector and buffer partition input connector

• false output connector and other filter input connectors

### **Filter Connection Flag**



Title	Function	Function Name [S-]	No.
Function Specs	Get filter connection destination	CDC_GetFiltCon	5.8

Format: CdcRet CDC\_GetFiltCon (Sint32 filtno, Sint32 \*bufno, Sint32 flnout)

**Input:** filtno : filter number

Output: bufno : buffer partition no. of true output connector connection

destination (CDC\_NUL\_SEL : not connected)

flnout : filter number of false output connector connection

destination (CDC\_NUL\_SEL : not connected)

**Function value:** Returns the return code.

**Function:** Gets the filter connection destination. When not connected returns

CDC NUL SEL.



Ī	Title	Function	Function Name [SF	₹]	No.
l	Function Specs	Resert selector (filter, partition)	CDC_ResetSelector		5.9

Format: CdcRet CDC\_ResetSelector (Sint32 rflag, Sint32 bufno)
Input: rflag : reset flag (insignificant 8 bits are effective)

bufno : buffer partition number

Output: none

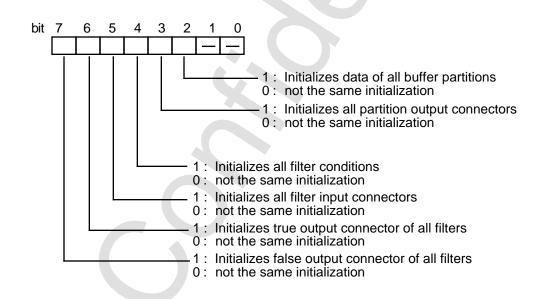
**Function value:** Returns the return code.

**Function:** Initializes selector set conditions according to the reset flag value.

•When reset flag = 0 : All specified buffer partition data is cleared

•When reset flag  $\neq 0$  : Buffer partition number is ignored

# Reset Flag



# 2) Initial Values

The following initial values are set for all selectors by initialization.

• Buffer partition data : All cleared (all data within the CD buffer is

cleared)

• Partition output connector : Status: All unconnected

• Filter conditions : Similar to initialization by "set filter mode"

• Filter input connector : Status: All unconnected

• True output connector : buffer partition and same number companion

filters are connected

• False output connector : Status: All unconnected

#### 8.2.6 Buffer Information

Title	Function	Function Name	[S-]	No.
Function	Get CD buffer size	CDC_GetBufSiz		6.1
Specifications				

Format: CdcRet CDC\_GetBufSiz (Sint32 \*totalsiz, Sint32 \*bufnum, Sint32 \*freesiz)

**Input:** none

Output: totalsiz : total buffer size (sector units, 200 sectors = C8H)

bufnum : total buffer partition number (24 partitions = 18H)

freesiz : empty buffer size (sector units)

**Function value:** Returns the return code.

**Function:** Gets the Cd buffer total size and buffer partition number (both are

fixed values), also the current space size. The selector number (or

filter number) is equal to the total buffer partiton number.

**Remarks:** Even if the space size is 0, the buffer does not need to be full since

there are sectors being used by the system.

Title	Function	Function Name	[S-]	No.
Function	Get buffer partiton sector number	CDC_GetSctNum		6.2
Specifications				

Format: CdcRet CDC\_GetSctNum (Sint32 bufno, Sint32 \*snum)

**Input:** bufno : buffer partition number

Output: snum : sector number within buffer partition

(buffer partition size)

**Function value:** Returns the return code.

**Function:** Gets current sector number (buffer partition size) of buffer partition.



Title	Function	Function Name [SR]	No.
Function	Calculation of actual data size	CDC_CalActSiz	6.3
Specifications			

Format: CdcRet CDC\_CalActSiz (Sint32 bufno, Sint32 spos, Sint32 snum)

**Input:** bufno : buffer partition number

spos : sector postion (CDC\_SPOS\_END : shows the

partition's last sector)

snum : sector number (CDC\_SNUM\_END : shows the

sector number from spos to the last partition)

Output: none

**Function value:** Returns the return code.

**Function:** Calculates the actual data size (word number when fetched to host) of

the buffer partition designated sector range, and holds that result. Executes the setting of the sector length to set the length of 1 sector.

**Remarks:** Connectors already connected to the partition output are disconnected.

Executes "get actual data size" to get the calculation result of the actual

data size.

Title	Function	Function Name	[S-]	No.
Function	Get actual data size	CDC_GetActSiz		6.4
Specifications				

Format: CdcRet CDC\_GetActSiz (Sint32 \*actwsiz)

**Input:** none

Output: actwsiz : actual data size (word units)

**Function value:** Returns the return code.

**Function:** Gets the actual data size (initial value of 0) held by the CD block.

Title	Function	Function Name	[S-]	No.
Function	Get sector information	CDC_GetSctInfo		6.5
Specifications				

Format: CdcRet CDC\_GetSctInfo (Sint32 bufno, Sint32 spos, CdcSct \*sct)

**Input:** bufno : buffer partition number

spos : sector position (CDC SPOS END : shows the

partition's last sector)

Output: sct : sector information

**Function value:** Returns the return code.

**Function:** Gets header information (FAD) and subheader information of buffer

partition designated sectors.

Title	Function	Function Name [SR]	No.
Function Specs	Execute frame address search	CDC_ExeFadSearch	6.6

Format: CdcRet CDC\_ExeFadSearch (Sint32 bufno, Sint32 spos, Sint32 fad)

**Input:** bufno : buffer partition number

spos : sector position (CDC\_SPOS\_END : shows the

partition's last sector)

fad : frame address

Output: none

**Function value:** Returns the return code.

**Function:** Starting at the designated sector position in the buffer partition, with the frame address as the key, the sector is retrieved and the result

retained.

1) Direction of retrieval

Retrieves in the direction of the buffer partition (from the

smaller to larger sector position.)

2) Retrieval result

Contents to be held as retrieval results include the buffer partition number, retrieved sector position, and the frame address of that sector.

3) Retrieval conditions

When a FAD that coincides does not exist, the nearest sector not exceeding the designated FAD is used. Therefore, retrieval resultant FAD is the greatest FAD that satisfies FAD≤ designated FAD. When there is no sector that satisfies this condition or when retrieval can not be done because of an error, the sector position of the retrieval results is CDC SOPS END.

**Remarks:** Connector that is already connected to the partition output is

disconnected. The retrieval result executes to get the frame address

retrieval result.

Title	Function	Function Name [S-]	No.
Function Specs	Get frame address search results	CDC_GetFadSearch	6.7

Format: CdcRet CDC GetFadSearch (Sint32 \*bufno, Sint32 \*spos, Sint32 \*fad)

**Input:** none

Output: bufno : buffer partition number

spos : sector position (CDC\_SPOS\_END : no applicable

sector shown)

fad : frame address

**Function value:** Returns the return code.

**Function:** Gets frame address search results (initial values all 0) held by the CD

block.



## 8.2.7 Buffer Input/Output

Each buffer input/output function (get sector data, delete, write, copy, move) includes the connector connection process. In other words, by issuing 1 command, the device and selector connections and flow in and out complex processes are done. Connectors connected to the partition output connector are disconnected.

Title	Function	Function Name	[SR]	No.
Function	Set sector length	CDC_SetSctLen		7.1
Specifications				

Format: CdcRet CDC\_SetSctLen (Sint32 getslen, Sint32 putslen)

**Input:** getslen : sector length when fetching

putslen : sector length when writing

Output: none

**Function value:** Returns the return code.

**Function:** Sets the length of one sector in the next cases.

• Gets sector data (CD buffer to host)

• Writes sector data (host to CD buffer)

# Sector Length

Constant Name	Description
CDC_SLEN_2048	2048 bytes (user data) ······ initial value when fetched, 2324 bytes if mode 2 form 2
CDC_SLEN_2336	2336 bytes (up to subheader)
CDC_SLEN_2340	2340 bytes (up to header)
CDC_SLEN_2352	2352 bytes (entire sector)
CDC_SLEN_NOCHG	No change

**Remarks:** The set "sector length when fetching" is also reflected in "Calculation

of actual data size."

Title	Function	Function Name	[SR]	No.
Function Specs	Get sector data	CDC_GetSctData		7.2

Format: CdcRet CDC\_GetSctData (Sint32 bufno, Sint32 spos, Sint 32 snum)

**Input:** bufno: buffer partition number

spos : sector position (CDC\_SPOS\_END : shows the

partition's last sector)

snum: sector number (CDC\_SNUM\_END: shows the sector

number from spos to the last partition)

Output: none

**Function value:** Returns the return code.

**Function:** Gets sector data from the designated sector range of the buffer partition. **Remarks:** After executing the function, data must be transferred and fetched.

Title	Function	Function Name [S	R]	No.
Function Specs	Delete sector data	CDC_DelSctData		7.3

Format: CdcRet CDC\_DelSctData (Sint32 bufno, Sint32 spos, Sint 32 snum)

**Input:** bufno : buffer partition number

spos : sector position (CDC\_SPOS\_END : shows the

partition's last sector)

snum: sector number (CDC SNUM END: shows the

sector number from spos to the last partition)

Output: none

**Function value:** Returns the return code.

**Function:** Deletes sector data of the designated sector range of the buffer partition.

**Remarks:** Sector positions after data is deleted advance in order.

		/	
Title	Function	Function Name [SR]	No.
Function Specs	Get and delete sector data	CDC_GetdelSctData	7.4

Format: CdcRet CDC\_GetdelSctData (Sint32 bufno, Sint32 spos, Sint 32 snum)

**Input:** bufno : buffer partition number

spos : sector position (CDC\_SPOS\_END : shows the

partition's last sector)

snum: sector number (CDC SNUM END: shows the

sector number from spos to the last partition)

Output: none

**Function value:** Returns the return code.

**Function:** Gets data from the designaed sector range of the buffer partition.

Sector data of the designated sector range is deleted from the buffer

partition.

**Remarks:** After the function is executed, data must be transferred and fetched.

Even if all data is transferred without being fetched, all data of the

designated sector range is deleted.



Title	Function	Function Name	[S-]	No.
Function Specs	Write sector data	CDC_PutSctData		7.5

Format: CdcRet CDC\_PutSctData (Sint32 filtno, Sint32 snum)

**Input:** filtno : filter number

snum: sector number

Output: none

**Function value:** Returns the return code.

**Function:** Writes sector data to the designated filter.

**Remarks:** Data must be transferred and written after the function is executed.

Data of the designated sector number is written even if data transfer is interrupted. Values except for transferred data are not fixed.

TitleFunctionFunction Name[SR]No.Function SpecsCopy sector dataCDC\_CopySctData7.6

Format: CdcRet CDC\_CopySctData (Sint32 srcbn, Sint32 spos, Sint32 snum, Sint32 dstfln)

**Input:** srcbn : copy source buffer partition number

spos : sector postion (CDC\_SPOS\_END : shows the

partition's last sector)

snum : sector number (CDC\_SNUM\_END : shows the

sector number from spos to the last partition)

dstfln : copy destination filter number

Output: none

**Function value:** Returns the return code.

**Function:** Copies sector data from designated sector range of the buffer

partition. (Copies in between selectors.)

Title Function Function Specs Move sector data Function Name [SR] No. CDC\_MoveSctData 7.7

Format: CdcRet CDC\_MoveSctData (Sint32 srcbn, Sint32 spos, Sint32 snum, Sint32 dstfln)

**Input:** srcbn : move source buffer partition number

spos : sector postion (CDC\_SPOS\_END : shows the

partition's last sector)

snum : sector number (CDC\_SNUM\_END : shows the

sector number from spos to the last partition)

dstfln: copy destination filter number

Output: none

**Function value:** Returns the return code.

**Function:** Moves sector data from designated sector range of the buffer

partition. (Moves in between selectors.)

Title	Function	Function Name	[SR]	No.
Function	Get sector data copy/move error	CDC_GetCopyErr		7.8
Specifications				

CdcRet CDC\_GetCopyErr (Sint32 \*cpyerr) Format:

Input:
Output:
Function value:

cpyerr: copy/move error information Returns the return code. Gets sector data copy/move error information. **Function:** 

Copy/Move Error Information

Constant Name	Description
CDC COPY OK	Normal end ····· initial value
CDC_COPY_NG	Error is caused
	Selector is disconnected during copy/move
	No more space in buffer
CDC_COPY_BUSY	Copy/Move process is in progress



## 8.2.8 CD Block File System

Title	Function	Function Name	[SR]	No.
Function	Change directory	CDC_ChgDir		8.1
Specifications				_ `

Format: CdcRet CDC\_ChgDir (Sint32 filtno, Sint32 fid)

**Input:** filtno : filter number

fid : file identifier of the directory file

(CDC\_NUL\_FID : root directory)

Output: none

**Function value:** Returns the return code.

**Function:** Moves the current directory to the directory shown by the

designated file identifier. Moves to the root directory if

CDC\_NUL\_FID is specified in the file identifier.

When moved, it reads directory record, and retains a maximum of 254 files worth of data from the start (256 files if including parent and own). Returns REJECT when directory is not designated by

the file identifier.

**Remarks:** The CD block file system uses the selector in the operation. The

operation selector designates through the filter number.

Title	Function	Function Name	[SR]	No.
Function	Hold file information	CDC_ReadDir		8.2
Specifications				

Format: CdcRet CDC\_ReadDir (Sint32 filtno, Sint32 fid)

**Input:** filtno : filter number

fid : file identifier

Output: none

**Function value:** Returns the return code.

**Function:** Reads the directory record of the current directory and holds up

to 254 pieces of file information after the designated file identifier

(own and parent are always retained).

**Remarks:** The CD block file system uses the selector in the operation. The

operation selector designates through the filter number.

Title	Function	Function Name	[S-]	No.
Function	Get holding file information range	CDC_GetFileScope		8.3
Specifications				

CdcRet CDC GetFileScope (Sint32 \*fid, Sint32 \*infnum) Format:

Input: none

: first file identifier being held Output: fid

: number of file information being held infnum

**Function value:** Returns the return code.

**Function:** Returns the file information range held by the current CD block. Since the self and parent directories are always held, they are not **Remarks:** 

included in the file information number.

Title	Function	Function Name	[S-]	No.
Function	Get holding file information	CDC_TgetFileInfo		8.4
Specifications				

CdcRet CDC\_TgetFileInfo (Sint32 fid, CdcFile \*file) Format: : file identifier (CDC\_NUL\_FID : shows entire file Input:

information)

Output: file : file information **Function value:** Returns the return code.

**Function:** Obtains file information (12 bytes) of the indicated file

identifier. Obtains all file information (up to 254 pieces) held in

the CD block if CDC\_NUL\_FID is designated in the file

identifier.

Example: CdcFile file: /\* Reserves information storage area of 1 file\*/

ret = CDC\_TgetFileInfo (fid, &file);

CdcFile file[256]; /\* Reserves information storage area of all files \*/

ret = CDC\_TgetFileInfo (CDC\_NUL\_FID, file);

Title	Function	Function Name [SR]	No.
Function	Read file	CDC_ReadFile	8.5
Specifications			

Format: CdcRet CDC\_ReadFile (Sint32 filtno, Sint32 fid, Sint32 offset)

filtno : filter number Input:

: file identifier fid

offset: logical offset from first of file (sector units)

Output: none

**Function value:** Returns the return code.

**Function:** Reads the designated file. The read destination selector is

designated by the filter number.

Title	Function	Function Name	[SR]	No.
Function	Stop file access	CDC_AbortFile		8.6
Specifications				

Format: CdcRet CDC\_AbortFile (void)

Input: none Output: none

**Function value:** Returns the return code.

**Function:** Stops file access by each function (directory move, file

information hold, and file read) of the CD block file system. Pauses the CD drive, and the EFLS flag of the interrupt factor

register becomes 1.

**Remarks:** Even if file access is stopped, the buffer partition is not cleared

and the selector is not initialized. File information being held is destroyed if directory-move and file-information-hold are stopped. File access is also stopped when the tray is opened.

# 8.2.9 Register Access

Title	Function	Function Name	[]	No.
Function Specs	Get data transfer register pointer	CDC_GetDataPtr		9.1

Format: Uint16 \*CDC\_GetDataPtr (void)

Input: none Output: none

Function value: Register pointer

**Function:** Gets the data transfer register (DATATRNS) pointer.

Title	Function	Function Name []	No.
Function Specs	Get interrupt factor register value	CDC_GetHirqReq	9.2

Format: Uint16 CDC\_GetHirqReq (void)

Input: none Output: none

**Function value:** Register value

**Function:** Gets the interrupt factor register (HIRQREQ) value.

	Title	Function	Function Name []	No.
ĺ	Function Specs	Clear interrupt factor register	CDC_ClrHirqReq	9.3

Format: void CDC\_ClrHirqReq (Uint16 bitpat)

**Input:** bitpat : cleared bit pattern (bit to clear is 0, other bits are 1)

Output: none Function value: none

**Function:** Clears the interrupt factor by the designated bit pattern.

(Only 0 is written.)

**Remarks:** Since the CD communication interface uses the interrupt factor

register CMOK flag, clearing by the application program is

prohibited.

Title	Function	Function Name []	No.
Function Specs	Get interrupt mask register value	CDC_GetHirqMsk	9.4

Format: Uint 16 CDC\_GetHirqMsk (void)

Input: none Output: none

**Function value:** register value

**Function:** Gets the interrupt mask register (HIRQMSK) value.

Title	Function	Function Name []	No.
Function	Set interrupt mask register	CDC_SetHirqMsk	9.5
Specifications			

Format: void CDC\_SetHirqMsk (Uint 16 bitpat)

**Input:** bitpat: mask bit pattern (bit to mask is 0, other bits are 1)

Output: none Function value: none

**Function:** Masks interrupt factor by designated bit pattern.

Title	Function	Function Name	[]	No.
Function	Get MPEG register pointer	CDC_GetMpegPtr		9.6
Specifications				

Format: Uint 16 \*CDC\_GetMpegPtr (void)

Input: none Output: none

Function value: register pointer

**Function:** Gets MPEG register (MPEGRGB) pointer.