CONFIDENTIAL

Ethernet AVB Software
AVB Streaming Driver

User's Manual: Software

R-Car H3/M3/M3N/E3/D3 Series

Renesas Electronics
www.renesas.com

Rev.2.30   Dec. 2021

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

    "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

    "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

    Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1   October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas
Electronics Corporation. All trademarks and registered trademarks
are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademark

- Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
- Windows and Windows Media are registered trademarks of Microsoft Corporation in the United States and other countries.
- Other company names and product names mentioned herein are registered trademarks or trademarks of their respective owners.
- Registered trademark and trademark symbols (® and ™) are omitted in this document

How to Use This Manual

- **[Readers]**

  This manual is intended for engineers who develop products which use the R-Car H3/M3/M3N/E3 processor.

- **[Purpose]**

  This manual is intended to give users an understanding of the functions of the R-Car H3/M3/M3N/E3 processor device driver and to serve as a reference for developing hardware and software for systems that use this driver.

- **[How to Read This Manual]**

  It is assumed that the readers of this manual have general knowledge in the fields of electrical
  — engineering, logic circuits, microcontrollers, and Linux.
    → Read this manual in the order of the CONTENTS.
  — To understand the functions of a multimedia processor for R-Car H3/M3/M3N/E3
    → See the R-Car H3/M3/M3N/E3 User's Manual.
  — To know the electrical specifications of the multimedia processor for R-Car H3/M3/M3N/E3
    → See the R-Car H3/M3/M3N/E3 Data Sheet.

- **[Conventions]**

  The following symbols are used in this manual.

  Data significance: Higher digits on the left and lower digits on the right

  Note: Footnote for item marked with Note in the text

  Caution: Information requiring particular attention

  Remark: Supplementary information

  Numeric representation: Binary ... 0b××××, or ××××B

  Decimal ... ××××

  Hexadecimal ... 0x×××× or ××××H

  Data type: Double word ... 64 bits

  Word … 32 bits

  Half word ... 16 bits

  Byte ... 8 bits

CONFIDENTIAL

Table of Contents

LIST OF FIGURES

LIST OF TABLES

# 1.	Overview

## 1.1	Overview

This manual explains the driver module (this module) that function of AVB Streaming in EthernetAVB-IF on R-Car H3/M3/M3N/E3/D3 System Evaluation Board.

## 1.2	Reference

### 1.2.1	Standard

The following table shows the document related to module.

**Table 1.1	Standards**

Number	Issue	Title	Edition	Date
IEEE Std 802.1BA-2011	IEEE STANDARDS ASSOCIATION	IEEE Standard for Local and metropolitan area networks - Audio Video Bridging (AVB) Systems	-	30 September 2011
IEEE Std 802.1Qav-2009	IEEE STANDARDS ASSOCIATION	IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams	-	5 January 2010
IEEE Std 1722-2016	IEEE STANDARDS ASSOCIATION	IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks	-	7 December 2016

### 1.2.2	Related Document

The following table shows the document related to this module.

**Table 1.2	Related documents**

Number	Issue	Title	Edition	Date
-	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	Rev.2.30	Aug 2021
-	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7795SIPB0011S	Rev.1.09	May. 11, 2017
-	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7796SIPB0011S	Rev.0.04	Oct. 3, 2016
-	Renesas Electronics	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Hardware Manual	Rev.2.04	Jul. 17, 2018
-	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D (E3 board 4xDRAM) Hardware Manual	Rev.1.01	Jul. 19, 2018
-	Renesas Electronics	R-CarD3 System Evaluation Board Draak Hardware Manual	Rev.1.20	July. 2017

1.3 Restrictions

Nothing.

1.4 Terminology

The following table shows the terminology related to this module.

Table 1.3 Terminology

Terms	Explanation
AVB	Audio Video Bridging
AVTP	Audio Video Transport Protocol (IEEE Std 1722)
CBS	Credit Based Shaper
DMAC	Direct Memory Access Controller
FQTSS	Forwarding and Queuing Enhancements for Time Sensitive Streams (IEEE Std 802.1Q)
SRP	Stream Reservation Protocol (IEEE Std 802.1Q)
MAC	Media Access Control
MMRP	Multiple MAC Registration Protocol (IEEE Std 802.1Q)
MSRP	Multiple Stream Registration Protocol (IEEE Std 802.1Q)
MVRP	Multiple VLAN Registration Protocol (IEEE Std 802.1Q)
VLAN	Virtual LAN (IEEE Std 802.1Q)

2. Operating Environment

2.1 Hardware Environment

The following table lists the hardware needed to use this module.

Table 2.1 Hardware environment (R-Car H3/M3/M3N/E3/D3)

Name	Version	Manufacture
R-CarH3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarM3-SiP System Evaluation Board Salvator-X	-	Renesas Electronics
R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS	-	Renesas Electronics
R-CarE3 System Evaluation Board Ebisu-4D	-	Renesas Electronics
R-CarD3 System Evaluation Board Draak	-	Renesas Electronics

2.2   Module Configuration

The following figure shows the configuration of this module.
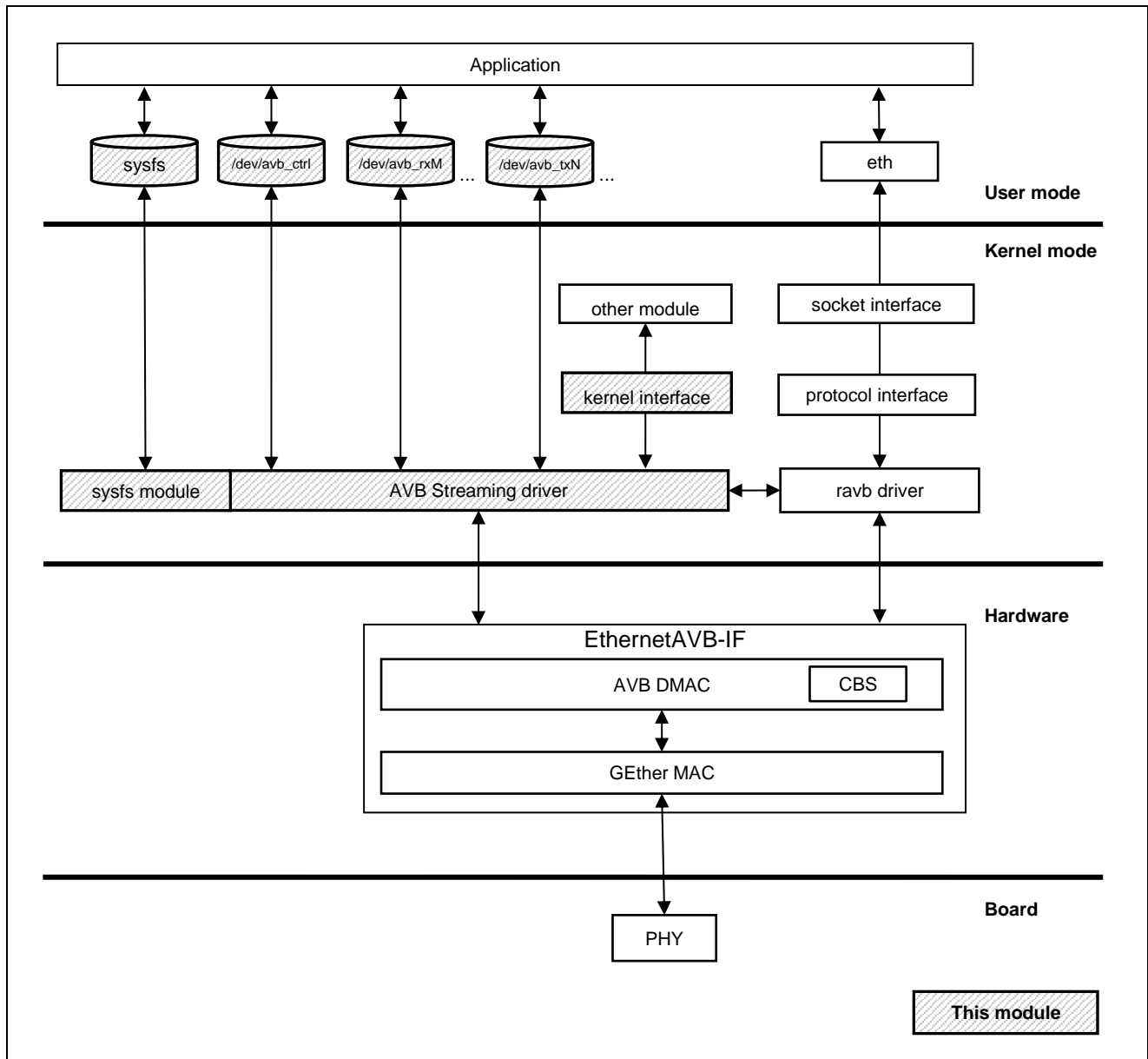
Figure 2.1   Module configuration

3. Function

This module controls the EthernetAVB-IF on the R-Car H3/M3/M3N/E3/D3.

The functions on this module are the following list.
- Transmission and Reception multiple AVB Stream.
- Traffic shaping by hardware Credit Based Shaper (CBS) with transmit stream.
- Stream separation using by StreamID at reception IEEE1722 (AVTP) stream.

3.1 Initialization Process for Listener

It is necessary to initialize a streaming device in this module, before starting streaming as a Listener.
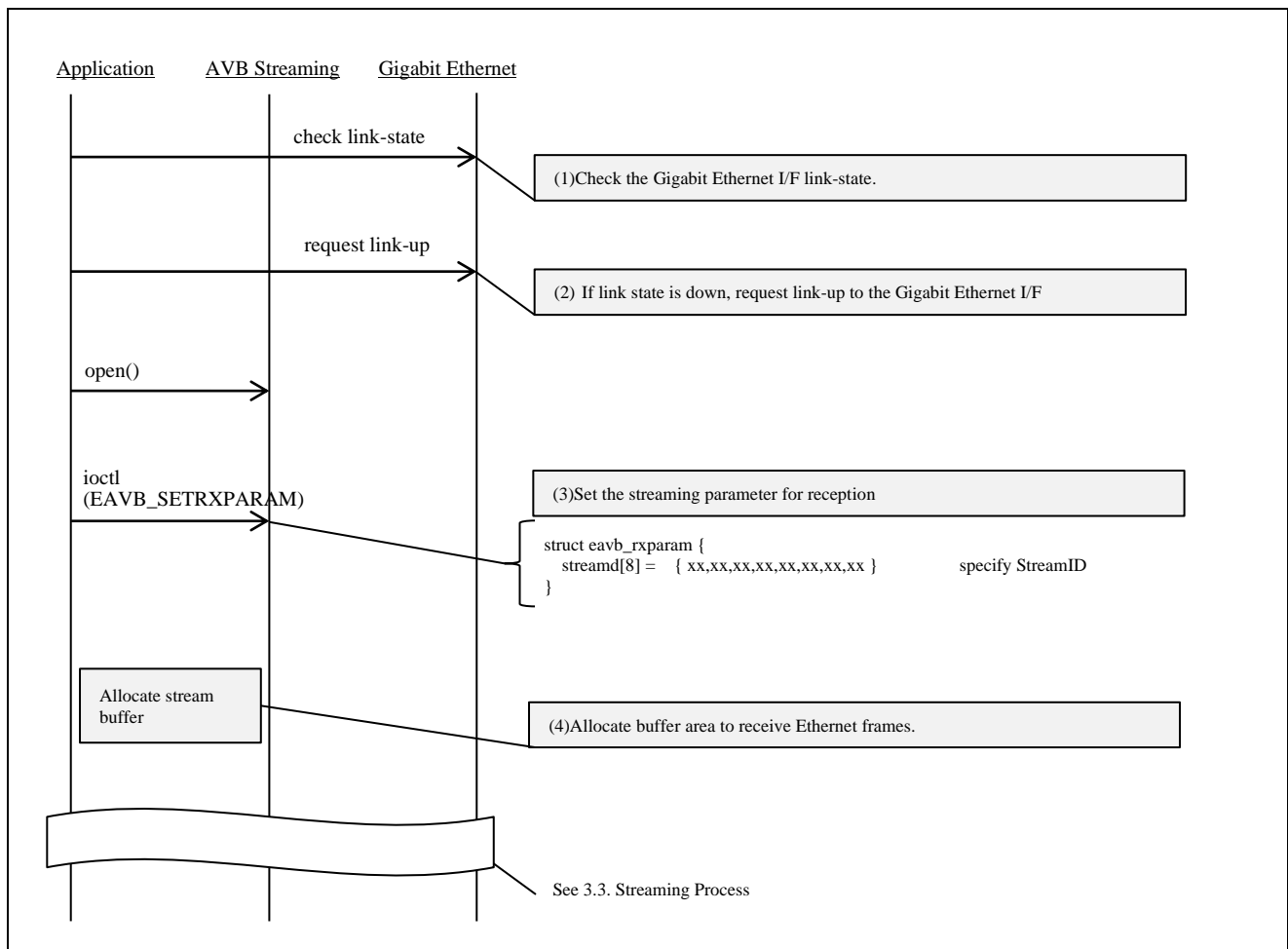
Figure 3.1 Initialization sequence for Listener

3.2   Initialization Process for Talker

It is necessary to initialize a streaming device in this module, before starting streaming as a Talker.
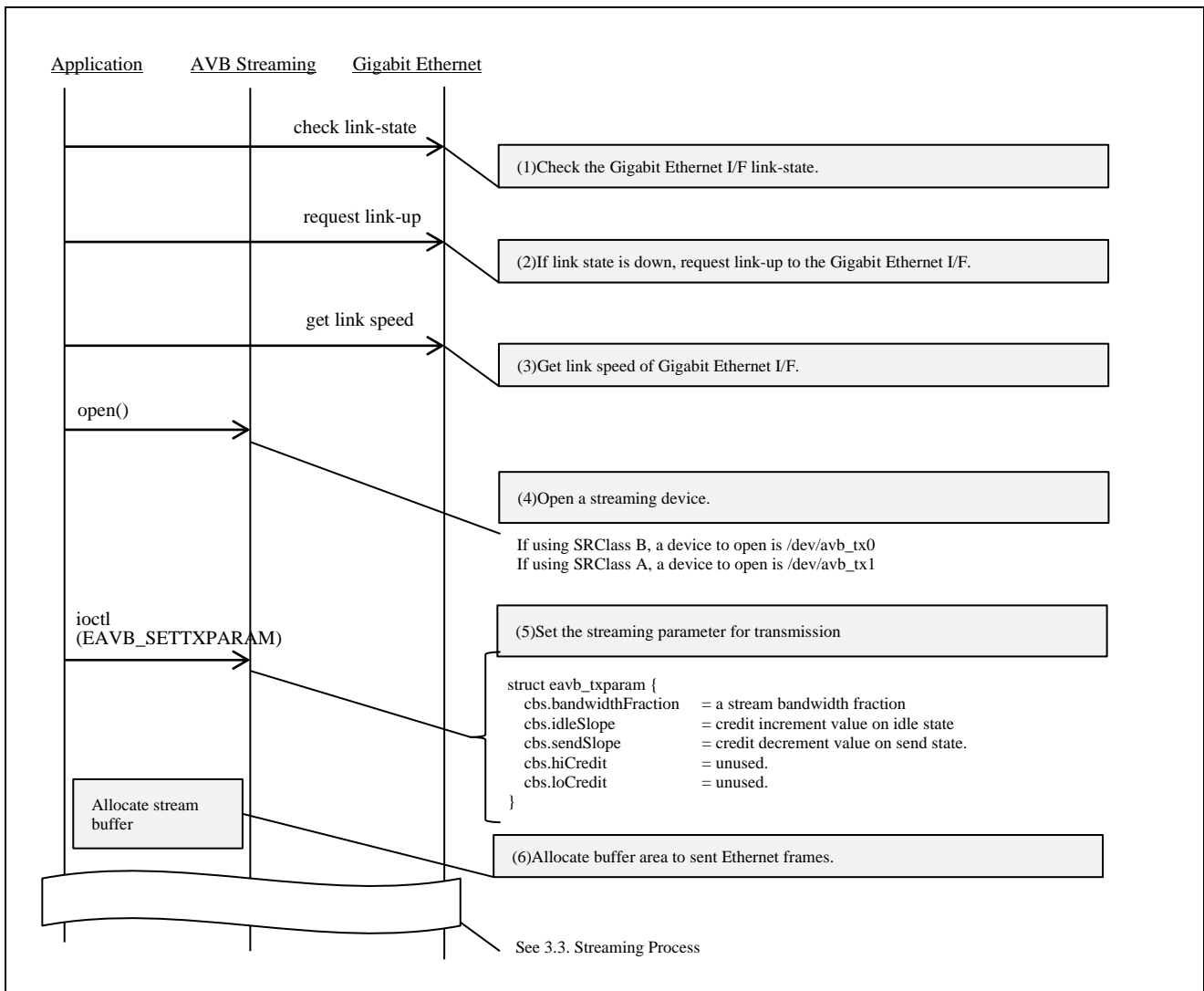
Figure 3.2   Initialization sequence for Talker

3.3 Streaming Process

This module handles entry as an Ethernet frame. The following figure shows the sequence of streaming using this module.
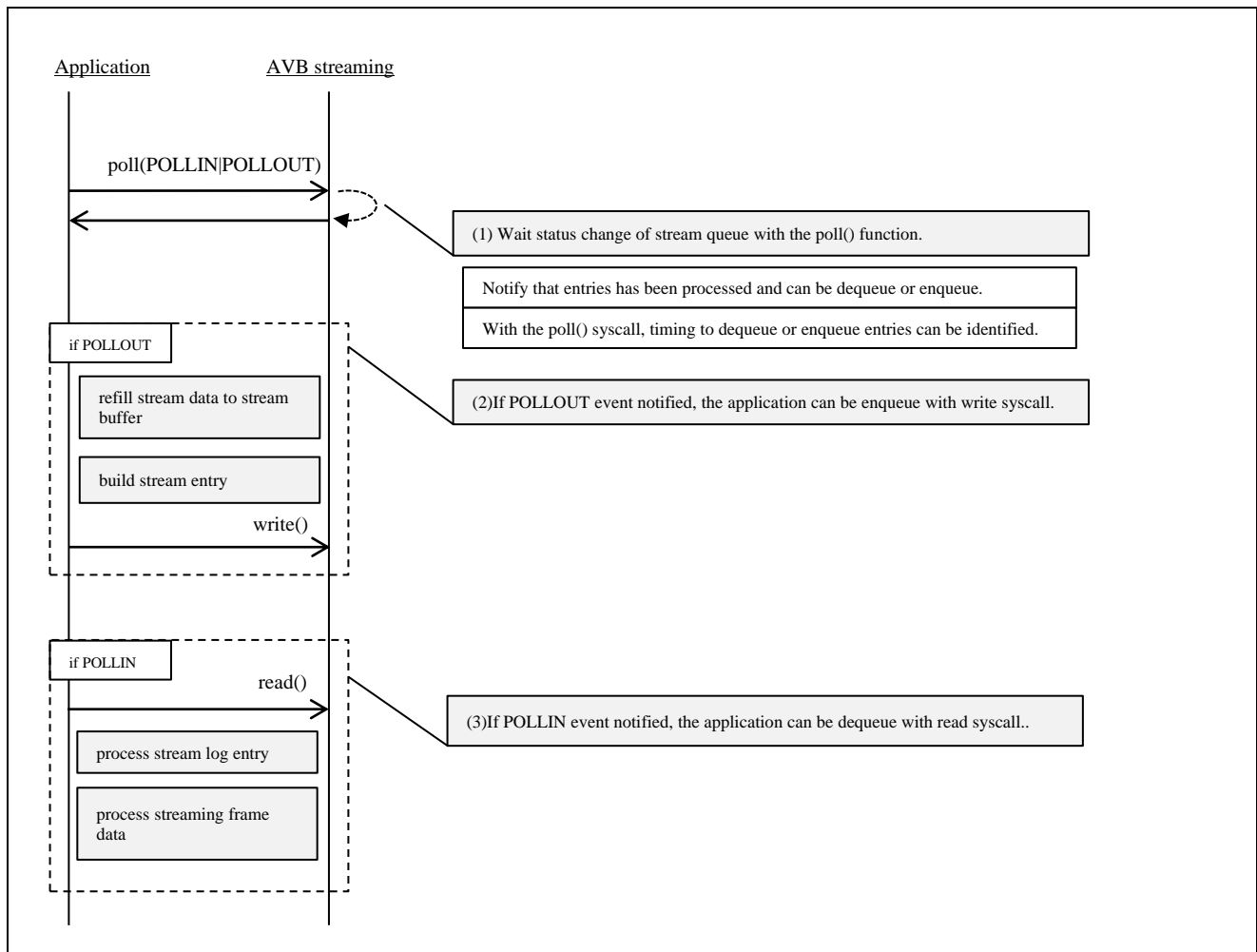
Figure 3.3 Streaming sequence

3.4   Finalization Process

It is necessary to finalize a streaming device, before stopped streaming.
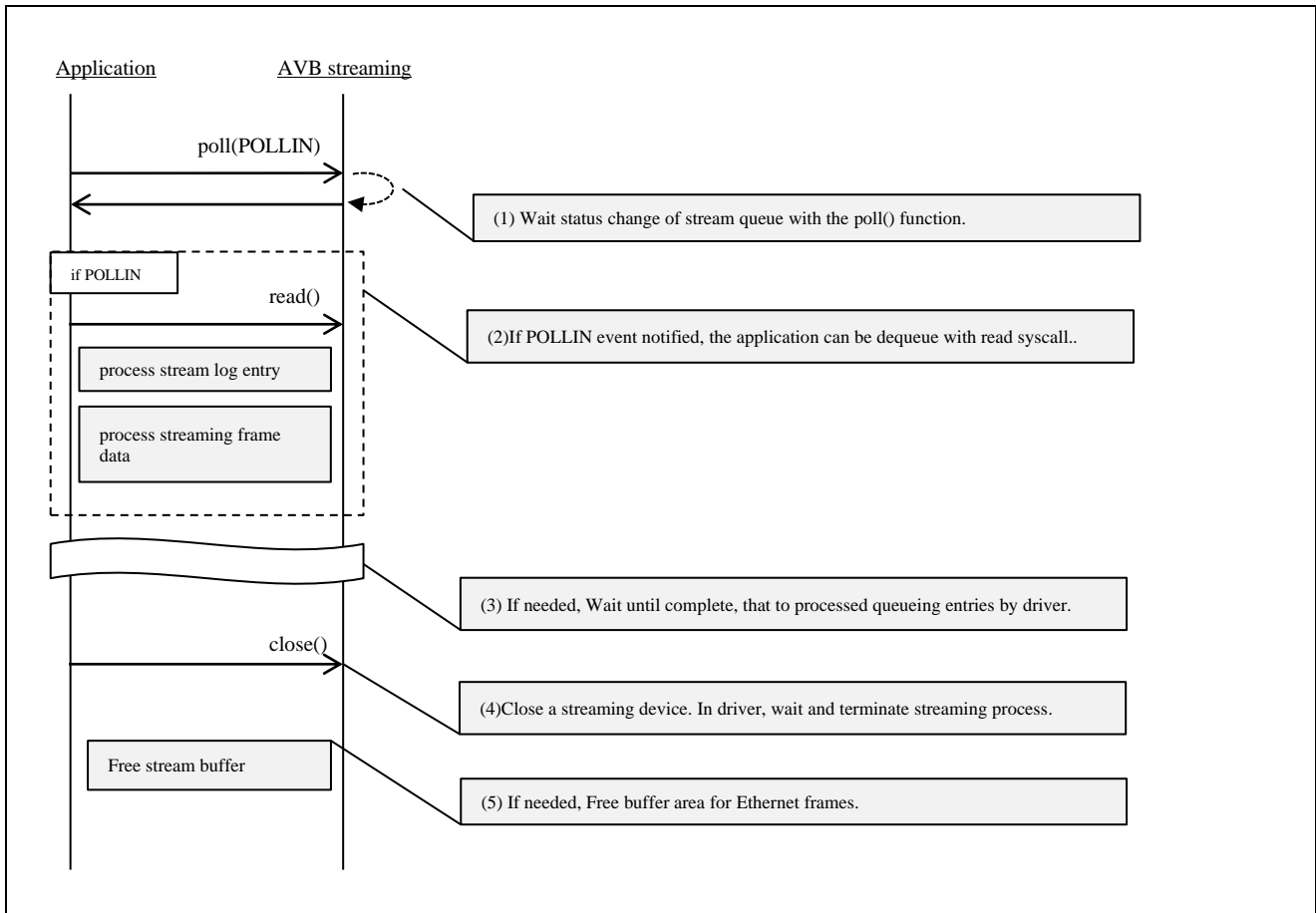
Figure 3.4   Finalization sequence

4. External Interface

The external interface of this module. Device node of this module is shown below.

Table 4.1 Device Node

Device node	Major number	Minor number
/dev/avb_tx0	234-254, 384-511 (#1)	0
/dev/avb_tx1	234-254, 384-511 (#1)	1
/dev/avb_rx[0-15]	234-254, 384-511 (#1)	2-17
/dev/avb_ctrl	234-254, 384-511 (#1)	127

Notes: #1   It is dynamically allocated.

4.1 sysfs interface

The sysfs interface of this module.

The interface for operating some parameters from a user land is AVB Streaming driver.

Device node of this module is shown below.

Table 4.2 List of sysfs device node for AVB Streaming driver

Device Class	Device node	Description
avb	/sys/class/avb/avb_tx0	The hardware queue device for SR Class B transmission.
avb	/sys/class/avb/avb_tx1	The hardware queue device for SR Class A transmission.
avb	/sys/class/avb/avb_rx[0-15]	The hardware queue device for reception.
avb	/sys/class/avb/avb_ctrl	The virtual device for driver control.

4.2   List of file operation system call for AVB Streaming driver

When an application uses Streaming driver, Using a file operation system call. Streaming driver supported system calls are shown below.

Table 4.3   List of file operation system call for AVB Streaming driver

Chapter	Function name	Description	remark
4.4.1	open	Open the streaming device.	-
4.4.2	close	Close the streaming device.	-
4.4.3	read	Dequeue streaming log entries from stream queue.	-
4.4.4	write	Enqueue streaming entries to stream queue.	-
4.4.5	poll	Wait stream queue status change by events.	-
4.4.6	mmap	Create a mapping for pages allocated by MAPPAGE.	#2
4.4.7	Ioctl(EAVB_SETTXPARAM)	Set the streaming parameter for transmission.	-
4.4.8	Ioctl(EAVB_GETTXPARAM)	Get the current streaming parameter for transmission.	-
4.4.9	Ioctl(EAVB_SETRXPARAM)	Set the streaming parameter for reception.	-
4.4.10	Ioctl(EAVB_GETRXPARAM)	Get the current streaming parameter for reception.	-
4.4.11	Ioctl(EAVB_GETCBSINFO)	Get the current CBS information.	-
4.4.12	Ioctl(EAVB_SETOPTION)	Set the option parameters.	-
4.4.13	Ioctl(EAVB_GETOPTION)	Get the option parameters.	-
4.4.14	Ioctl(EAVB_MAPPAGE)	Allocate a memory page.	#2
4.4.15	Ioctl(EAVB_UNMAPPAGE)	Free a memory page.	#2
4.4.16	Ioctl(EAVB_GDRVINFO)	Get the driver information.	#3
4.4.17	Ioctl(EAVB_GRINGPARAM)	Get the Rx/Tx ring parameters.	#3
4.4.18	Ioctl(EAVB_GSSET_INFO)	Get the number of streaming set.	#3
4.4.19	Ioctl(EAVB_GSTRINGS)	Get an array of the string set.	#3
4.4.20	Ioctl(EAVB_GSTATS)	Get the statistics.	#3
4.4.21	Ioctl(EAVB_GCHANNELS)	Get the number of Rx/Tx channel the driver support.	#3

Notes: #2   for Debug and Testing.

Notes: #3   for avbtool

4.3   List of kernel driver interface for AVB Streaming driver

To provide a function for using the Streaming driver from kernel driver interface. That is shown below.

Table 4.4   List of kernel driver interface for AVB Streaming driver

Chapter	Function name	Description	remark
4.5.1	ravb_streaming_open_stq_kernel	Open the streaming device.	-
4.5.2	ravb_streaming_release_stq_kernel	Close the streaming device.	-
4.5.3	read	Dequeue streaming log entries from stream queue.	#4
4.5.4	write	Enqueue streaming entries to stream queue.	#4
4.5.5	set_txparam	Set the streaming parameter for transmission.	#4
4.5.6	get_txparam	Get the current streaming parameter for transmission.	#4
4.5.7	set_rxparam	Set the streaming parameter for reception.	#4
4.5.8	get_rxparam	Get the current streaming parameter for reception.	#4
4.5.9	set_option	Set the option parameters.	#4
4.5.10	get_option	Get the option parameters.	#4
4.5.11	get_entrynum	Get the number of entry.	#4
4.5.12	get_linkspeed	Get the linkspeed of network interface.	#4
4.5.13	blocking_cancel	Cancel the wait by blocking.	#4

Notes: #4   definition of callback function at ravb_streaming_kernel_if structure

4.4   File operation system call specification for AVB Streaming driver

4.4.1   open

[Function]	int open(const char *device_name, int flags)	
[Arguments]	device_name	Request the "/dev/avb_ctrl", "/dev/avb_rxM" or "/dev/avb_txN".
	flags	Access modes, this argument should be O_RDWR
[Returns]	more than 0	Success
	-1	Error
[Error number]	ENODEV	No such device
	EBUSY	Already open device
	ENOMEM	Cannot allocate memory
[Description]	Open the streaming device.	
	If O_SYNC flag is set, the driver not to do cache sync operation to Ethernet frame buffer.	

4.4.2   close

[Function]	int close(int fd)	
[Arguments]	fd	File descriptor
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	No such device
[Description]	Close the streaming device.	
	When you call this function, driver promise waiting for completion and terminate of the streaming process.	

RENESAS

4.4.3   read

[Function]	ssize_t read(int fd, void *buf, size_t count)	

[Arguments]	fd	File descriptor
	buf	Base address of entry buffer
	count	Bytes of entry buffer size

[Returns]	0 or more	Success, the number of bytes read
	-1	Error

[Error number]	EAGAIN	A stream entry queue is empty, if O_NONBLOCK mode
	EINVAL	Invalid argument Count is not multiple of the entry size
	EFAULT	Cannot access userspace
	ENOMEM	Count is greater than internal buffer, if EAVB_BLOCK_WAITALL mode
	EINTR	Interrupted

[Description]	Dequeue streaming log entries from a stream queue.

4.4.4   write

[Function]	ssize_t write(int fd, const void *buf, size_t count)	

[Arguments]	fd	File descriptor
	buf	Base address of entry buffer
	count	Bytes of entry buffer size

[Returns]	0 or more	Success, the number of bytes write
	-1	Error

[Error number]	EAGAIN	A stream entry queue was full, if O_NONBLOCK mode
	EINVAL	Invalid argument Count is not multiple of the entry size
	EFAULT	Cannot access userspace
	EINTR	Interrupted

[Description]	Enqueue streaming entries to a stream queue.

4.4.5   poll

[Function]	int poll(struct pollfd *fds, nfsds_t nfds, int timeout)	
[Arguments]	fds	To be monitoring fds information
	nfds	Number of items in the fds array
	timeout	Wait time, in milliseconds. Specifying a negative value means an infinite.
[Returns]	more than 0	Success, in which case have nonzero revents fields
	0	Timeout
	-1	Error
[Description]	Enqueue streaming entries to a stream queue.	

4.4.6   mmap

[Function]	void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset)	
[Arguments]	addr	this argument should be NULL
	length	this argument should be PAGE_SIZE
	prot	this argument should be PROT_READ	PROT_WRITE
	flags	this argument should be MAP_SHARED
	fd	File descriptor
	offset	Physical address
[Returns]	more than 0	Success, returns a pointer to the mapped area.
	-1	Error
[Error number]	EAGAIN	Virtual memory mapping faild.
	EINVAL	Invalid argument
[Description]	Creates a new mapping in the virtual address space of the calling process.	
	This functions acceptable physical address is only pages allocated by EAVB_MAPPAGE.	
	If O_SYNC flag is set with open function, mapped virtual address will be uncached area.	
	This function is debug uses.	

4.4.7 ioctl(EAVB_SETTXPARAM)

[Function]	int ioctl(int fd, int request, const struct eavb_txparam *argp)	

[Arguments]	fd	File descriptor
	request	EAVB_SETTXPARAM
	argp	Pointer of eavb_txparam structure

[Returns]	0	Success
	-1	Error

[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't transmission type
	ENOSPC	Total bandwidth fraction over 75%
	EFAULT	Cannot access userspace
	EBUSY	Hardware update disabled state

[Description]	Set the streaming parameter for transmission.
	This function can control to the following parameters.
	- Hardware CBS parameters

4.4.8 ioctl(EAVB_GETTXPARAM)

[Function]	int ioctl(int fd, int request, struct eavb_txparam *argp)	

[Arguments]	fd	File descriptor
	request	EAVB_GETTXPARAM
	argp	Pointer of eavb_txparam structure

[Returns]	0	Success
	-1	Error

[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't transmission type
	EFAULT	Cannot access userspace

[Description]	Get the current streaming parameter for transmission.

4.4.9 ioctl(EAVB_SETRXPARAM)

[Function]	int ioctl(int fd, int request, const struct eavb_rxparam *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_SETRXPARAM
	argp	Pointer of eavb_rxparam structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't reception type Invalid hardware state
	EBUSY	Hardware update disabled state
	EAGAIN	Hardware updatable state. However, it could not be updated
	EFAULT	Cannot access userspace

[Description]   Set the streaming parameter for reception.

This function can control to the following parameters.

- StreamID for Hardware separation filter

4.4.10 ioctl(EAVB_GETRXPARAM)

[Function]	int ioctl(int fd, int request, struct eavb_rxparam *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GETRXPARAM
	argp	Pointer of eavb_rxparam structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't reception type
	EFAULT	Cannot access userspace

[Description]   Get the current streaming parameter for reception.

4.4.11   ioctl(EAVB_GETCBSINFO)

[Function]   int ioctl(int fd, int request, struct eavb_cbsinfo *argp)

[Arguments]   fd   File descriptor

request   EAVB_GETCBSINFO

argp   Pointer of eavb_cbsinfo structure

[Returns]   0   Success

-1   Error

[Error number]   EFAULT   Cannot access userspace

[Description]   Get the current CBS information.

4.4.12 ioctl(EAVB_SETOPTION)

[Function]	int ioctl(int fd, int request, const struct eavb_option *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_SETOPTION
	argp	Pointer of eavb_option structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Set the option parameters that specified by eavb_option.id.	

This function can control to the following option parameters.

- Blocking mode

Blocking mode (EAVB_OPTIONID_BLOCKMODE)

This option can select the following mode.

- EAVB_BLOCK_NOWAIT (default)
  In this mode, the read() will unblock as soon as possible.
- EAVB_BLOCK_WAITALL
  In this mode, the read() will unblock when the specified buffer is full.

4.4.13 ioctl(EAVB_GETOPTION)

[Function]	int ioctl(int fd, int request, struct eavb_option *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GETOPTION
	argp	Pointer of eavb_option structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current option parameters that specified by eavb_option.id.	

4.4.14   ioctl(EAVB_MAPPAGE)

[Function]   int ioctl(int fd, int request, struct eavb_dma_alloc *argp)

[Arguments]   fd   File descriptor

                    request   EAVB_MAPPAGE

                    argp   Pointer of eavb_dma_alloc structure

[Returns]   0   Success

                    -1   Error

[Error number]   ENOMEM   Can't allocate a memory page

                    EFAULT   Cannot access userspace

[Description]   Allocate a memory page.

                    This function is debug uses.

4.4.15   ioctl(EAVB_UNMAPPAGE)

[Function]   int ioctl(int fd, int request, struct eavb_dma_alloc *argp)

[Arguments]   fd   File descriptor

                    request   EAVB_UNMAPPAGE

                    argp   Pointer of eavb_dma_alloc structure

[Returns]   0   Success

                    -1   Error

[Error number]   EINVAL   Specifying a page isn't allocate by AVB Streaming driver

                    EFAULT   Cannot access userspace

[Description]   Free a memory page.

                    This function is debug uses.

4.4.16 ioctl(EAVB_GDRVINFO)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_drvinfo *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GDRVINFO
	argp	Pointer of eavb_avbtool_drvinfo structure
[Returns]	0	Success
	-1	Error
[Error number]	EFAULT	Cannot access userspace

[Description]   Get the driver information.

This function can get to the following information.

- The driver name
- The driver verison
- The number of statistics counters

This function is recommended to be used via avbtool.

4.4.17 ioctl(EAVB_GRINGPARAM)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_ringparam *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GRINGPARAM
	argp	Pointer of eavb_avbtool_ringparam structure
[Returns]	0	Success
	-1	Error
[Error number]	EFAULT	Cannot access userspace

[Description]   Get the Rx/Tx ring parameters.

This function can get to the following information.

- The maximum number of pending Rx ring entries
- The maximum number of pending Tx ring entries.

This function is recommended to be used via avbtool.

4.4.18　ioctl(EAVB_GSSET_INFO)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_sset_info *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GSSET_INFO
	argp	Pointer of eavb_avbtool_sset_info structure
[Returns]	0	Success
	-1	Error
[Error Numbers]	ENOMEM	Cannot allocate memory
	EFAULT	Cannot access userspace
[Description]	Get the number of streaming set.	
	This function is recommended to be used via avbtool.	

4.4.19　ioctl(EAVB_GSTRINGS)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_gstrings *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GSTRINGS
	argp	Pointer of eavb_avbtool_gstrings structure
[Returns]	0	Success
	-1	Error
[Error number]	EOPNOTSUPP	Operation not supported.
	EFAULT	Cannot access userspace
	ENOMEM	Cannot allocate memory
[Description]	Get an array of the string set.	
	This function is recommended to be used via avbtool.	

4.4.20   ioctl(EAVB_GSTATS)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_stats *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GSTATS
	argp	Pointer of eavb_avbtool_stats structure
[Returns]	0	Success
	-1	Error
[Error number]	ENOMEM	Cannot allocate memory
	EFAULT	Cannot access userspace
[Description]	Get the statistics.	
	This function is recommended to be used via avbtool.	

4.4.21   ioctl(EAVB_GCHANNELS)

[Function]	int ioctl(int fd, int request, struct eavb_avbtool_channels *argp)	
[Arguments]	fd	File descriptor
	request	EAVB_GCHANNELS
	argp	Pointer of eavb_avbtool_channels structure
[Returns]	0	Success
	-1	Error
[Error number]	EFAULT	Cannot access userspace
[Description]	Get the number of Rx/Tx channel the driver support.	

This function can get to the following parameter.

- The maximum number of Rx channel the driver support.
- The maximum number of Tx channel the driver support.
- Current valid number of Rx channel.
- Current valid number of Tx channel.

This function is recommended to be used via avbtool.

4.5   Kernel driver interface Specification for AVB Streaming driver

4.5.1   ravb_streaming_open_stq_kernel

[Function]	int ravb_streaming_open_stq_kernel(enum AVB_DEVNAME dev_name, struct ravb_streaming_kernel_if *kif, unsigned int flags)

[Arguments]	dev_name	Request the AVB_DEVNAME_RXm or AVB_DEVNAME_TXn".
	kif	Pointer of ravb_streaming_kernel_if structure
	flags	flags can be specified O_SYNC and O_NONBLOCK

[Returns]	0	Success
	-EINVAL	Invalid argument
	-ENODEV	No such device
	-EBUSY	Already open device
	-ENOMEM	Cannot allocate memory

[Description]	Open the streaming device and store handle and pointer of callback functions in kif.
	If O_SYNC flag is set, the driver not to do cache sync operation to Ethernet frame buffer.

4.5.2   ravb_streaming_release_stq_kernel

[Function]   int ravb_streaming_release_stq_kernel(void *handle)

[Arguments]   handle   handle of ravb_streaming_kernel_if structure

[Returns]   0   Success

-EINVAL   Invalid argument

[Description]   Close the streaming device.

When you call this function, driver promise waiting for completion and terminate of the streaming process.

4.5.3   read

[Function]	int read(void *handle, struct eavb_entry *buf, unsigned int num)	

[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	buf	Base address of entry buffer
	num	Number of entry

[Returns]	0 or more	Success, the number of entry read
	-EINVAL	Invalid argument
	-ENOMEM	Count is greater than internal buffer, if EAVB_BLOCK_WAITALL mode
	-EAGAIN	A stream entry queue is empty, if O_NONBLOCK mode
	-EINTR	Interrupted

[Description]	Dequeue streaming log entries from a stream queue.

4.5.4   write

[Function]	int write(void *handle, struct eavb_entry *buf, unsigned int num)	

[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	buf	Base address of entry buffer
	num	Number of entry

[Returns]	0 or more	Success, the number of entry write
	-EINVAL	Invalid argument
	-EAGAIN	A stream entry queue is empty, if O_NONBLOCK mode
	-EINTR	Interrupted

[Description]	Enqueue streaming entries to a stream queue.

4.5.5 set_txparam

[Function]   long set_txparam(void *handle, struct eavb_txparam *txparam)

[Arguments]   handle   handle of ravb_streaming_kernel_if structure

                  txparam   Pointer of eavb_txparam structure

[Returns]   0   Success

                  -EINVAL   Invalid argument

                  -EPERM   This device isn't transmission type

                  -ENOSPC   Total bandwidth fraction over 75%

                  -EBUSY   Hardware update disabled state

[Description]   Set the streaming parameter for transmission.

                  This function can control to the following parameters.

                  -    Hardware CBS parameters

4.5.6 get_txparam

[Function]   long get_txparam(void *handle, struct eavb_txparam *txparam)

[Arguments]   handle   handle of ravb_streaming_kernel_if structure

                  txparam   Pointer of eavb_txparam structure

[Returns]   0   Success

                  -EINVAL   Invalid argument

                  -EPERM   This device isn't transmission type

[Description]   Get the current streaming parameter for transmission.

4.5.7   set_rxparam

[Function]	long set_rxparam(void *handle, struct eavb_rxparam *rxparam)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	rxparam	Pointer of eavb_rxparam structure
[Returns]	0	Success
	-EINVAL	Invalid argument
	-EPERM	This device isn't reception type
		Invalid hardware state
	-EBUSY	Hardware update disabled state
	-EAGAIN	Hardware updatable state. However, it could not be updated

[Description]   Set the streaming parameter for reception.

This function can control to the following parameters.

- StreamID for Hardware separation filter

4.5.8   get_rxparam

[Function]	long get_rxparam(void *handle, struct eavb_rxparam *rxparam)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	rxparam	Pointer of eavb_rxparam structure
[Returns]	0	Success
	-EINVAL	Invalid argument
	-EPERM	This device isn't reception type

[Description]   Get the current streaming parameter for reception.

4.5.9   set_option

[Function]	long set_option(void *handle, struct eavb_option *option)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	option	Pointer of eavb_option structure
[Returns]	0	Success
	-EINVAL	Invalid argument

[Description]   Set the option parameters that specified by eavb_option.id.

This function can control to the following option parameters.

-    Blocking mode

Blocking mode (EAVB_OPTIONID_BLOCKMODE)

This option can select the following mode.

-    EAVB_BLOCK_NOWAIT (default)
     In this mode, the read() will unblock as soon as possible.

-    EAVB_BLOCK_WAITALL
     In this mode, the read() will unblock when the specified buffer is full.

4.5.10   get_option

[Function]	long get_option(void *handle, struct eavb_option *option)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	option	Pointer of eavb_option structure
[Returns]	0	Success
	-EINVAL	Invalid argument

[Description]   Get the current option parameters that specified by eavb_option.id.

4.5.11   get_entrynum

[Function]	long get_entrynum(void *handle, struct eavb_entrynum *entrynum)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
	entry_num	Pointer of eavb_entrynum structure
[Returns]	0	Success
	-EINVAL	Invalid argument

[Description]   Get the number of entrynum.

This function can get to the following parameters

-    accepted, that is accepted entries.

-    processed, that is stream queue entries.

-    completed, that is readable entries.

4.5.12   get_linkspeed

[Function]	long get_linkspeed(void *handle)	
[Arguments]	handle	handle of ravb_streaming_kernel_if structure
[Returns]	0	Success, but network interface is stopped
	Greater than 0	Success, the link speed of network interface at Mbps
	-EINVAL	Invalid argument

[Description]   Get the link speed via network driver.

4.5.13   blocking_cancel

[Function]   long blocking_cancel(void *handle)

[Arguments]   handle   handle of ravb_streaming_kernel_if structure

[Returns]   0   Success

                    -EINVAL   Invalid argument

[Description]   Cancel the wait by blocking at read() and write().
                    If O_NONBLOCK flag is set at ravb_streaming_open_stq_kernel(), This function is do
                    nothing.

4.6   Structure

The structure defined in ravb_eavb.h is explained.

4.6.1   eavb_entryvec

Table 4.5   struct eavb_entryvec

Structure name	Member		
	Type	**Name**	**Description**
eavb_entryvec	uint32_t	base	The physical address value for Ethernet frame buffer. This value should set to 4-byte align.
	uint32_t	len	Length of Ethernet frame buffer.

4.6.2   eavb_entry

Table 4.6   struct eavb_entry

Structure name	Member		
	Type	**Name**	**Description**
eavb_entry	uint32_t	seq_no	The sequence number of entry. This member should set to unique number for each entry.
	struct eavb_entryvec	vec[EAVB_ENTRYVECNUM]	The array of entry vectors.

4.6.3   eavb_cbsparam

Table 4.7   struct eavb_cbsparam

Structure name	Member		
	Type	**Name**	**Description**
eavb_cbsparam	uint32_t	bandwidthFraction	A stream bandwidth fraction value among the total bandwidth. This member should specify as a mapped value from 0x0 to 0xFFFFFFFF. (e.g. 0.0=0x0, 1.0=0xFFFFFFFF)
	uint32_t	idleSlope	The fraction value of credit increment on idle state. This value update only by the driver. The specified value from the user will be ignored. This value as mapped value from 0x0 to 0xFFFF. (e.g. 0.0=0x0, 1.0=0xFFFF)
	uint32_t	sendSlope	The fraction value of credit decrement on send state. This value update only by the driver. The specified value from the user will be ignored. This value as mapped value from 0x0 to 0xFFFF. (e.g. 0.0=0x0, 1.0=0xFFFF)
	uint32_t	hiCredit	The hiCredit value as defined in FQTSS. This value update only by the driver. The specified value from the user will be ignored.
	uint32_t	loCredit	The loCredit value as defined in FQTSS. This value update only by the driver. The specified value from the user will be ignored.

4.6.4   eavb_txparam

Table 4.8   struct eavb_txparam

Structure name	Member		
	Type	**Name**	**Description**
eavb_txparam	struct eavb_cbsparam	cbs	The struct of CBS parameters.

4.6.5   eavb_rxparam

Table 4.9   struct eavb_rxparam

Structure name	Member		
	Type	**Name**	**Description**
eavb_rxparam	uint8_t	streamid[8]	Specify StreamID as defined in AVTP.

4.6.6 eavb_cbsinfo

Table 4.10 struct eavb_cbsinfo

Structure name	Member		
	Type	**Name**	**Description**
eavb_cbsinfo	uint32_t	bandwidthFraction	The current fraction value total of bandwidth for each streaming device among the total bandwidth.
	struct eavb_cbsparam	param[EAVB_CLASS_MAX]	The current CBS parameters each of SR classes.

4.6.7 eavb_option

Table 4.11 struct eavb_option

Structure name	Member		
	Type	**Name**	**Description**
eavb_option	enum eavb_optionid	id	Specify the Option ID.
	uint32_t	param	The parameter of the selected Option ID.

4.6.8 eavb_entrynum

Table 4.12 struct eavb_entrynum

Structure name	Member		
	Type	**Name**	**Description**
eavb_entrynum	uint32_t	accepted	The number of accepted entries.
	uint32_t	processed	The number of stream queue entries.
	uint32_t	completed	The number of readable log entries.

4.6.9   ravb_streaming_kernel_if

Table 4.13   struct ravb_streaming_kernel_if

Structure name	Member		
	Type	**Name**	**Description**
ravb_streaming_ kernel_if	void *	handle	The streaming driver handle.
	int (*) (void *, struct eavb_entry *, unsigned int)	read	The callback function pointer of "read" at the kernel driver interface.
	int (*) (void *, struct eavb_entry *, unsigned int)	write	The callback function pointer of "write" at the kernel driver interface.
	long (*) (void *, struct eavb_txparam *)	set_txparam	The callback function pointer of "set_txparam" at the kernel driver interface.
	long (*) (void *, struct eavb_txparam *)	get_txparam	The callback function pointer of "get_txparam" at the kernel driver interface.
	long (*) (void *, struct eavb_rxparam *)	set_rxparam	The callback function pointer of "set_rxparam" at the kernel driver interface.
	long (*) (void *, struct eavb_rxparam *)	get_rxparam	The callback function pointer of "get_rxparam" at the kernel driver interface.
	long (*) (void *, struct eavb_option *)	set_option	The callback function pointer of "set_option" at the kernel driver interface.
	long (*) (void *, struct eavb_option *)	get_option	The callback function pointer of "get_option" at the kernel driver interface.
	long (*) (void *, struct eavb_entrynum *)	get_entrynum	The callback function pointer of "get_entrynum" at the kernel driver interface.
	long (*)(void *)	get_linkspeed	The callback function pointer of "get_linkspeed" at the kernel driver interface.
	long (*)(void *)	blocking_cancel	The callback function pointer of "blocking_cancel" at the kernel driver interface.

4.6.10 eavb_avbtool_drvinfo

Table 4.14 struct eavb_avbtool_drvinfo

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_drvinfo	uint8_t	driver[32]	The driver name.
	uint8_t	version[32]	The driver version string.
	uint8_t	fw_version[32]	Reserved. This value is always return 0 filled.
	uint8_t	bus_info[32]	Bus info for this interface.
	uint8_t	reserved1[32]	Reserved. This value is always return 0 filled.
	uint8_t	reserved2[12]	Reserved. This value is always return 0 filled.
	uint32_t	n_priv_flags	Reserved. This value is always return 0.
	uint32_t	n_stats	The number of statistics counters.
	uint32_t	testinfo_len	Reserved. This value is always return 0.
	uint32_t	eedump_len	Reserved. This value is always return 0.
	uint32_t	regdump_len	Reserved. This value is always return 0.

4.6.11 eavb_avbtool_ringparam

Table 4.15 struct eavb_avbtool_ringparam

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_ringparam	uint32_t	rx_max_pending	The maximum number of pending Rx ring entries the driver.
	uint32_t	rx_mini_max_pending	Reserved. This value is always return 0.
	uint32_t	rx_jumbo_max_pending	Reserved. This value is always return 0.
	uint32_t	tx_max_pending	The maximum number of pending Tx ring entries the driver.
	uint32_t	rx_pending	Reserved. This value is always return 0.
	uint32_t	rx_mini_pending	Reserved. This value is always return 0.
	uint32_t	rx_jumbo_pending	Reserved. This value is always return 0.
	uint32_t	tx_pending	Reserved. This value is always return 0.

4.6.12   eavb_avbtool_channels

Table 4.16   struct eavb_avbtool_channels

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_ch annels	uint32_t	max_rx	Maximum number of Rx channels the driver support.
	uint32_t	max_tx	Maximum number of Tx channels the driver support.
	uint32_t	max_other	Reserved. This value is always return 0.
	uint32_t	max_combined	Reserved. This value is always return 0.
	uint32_t	rx_count	Current valid number of Rx channel.
	uint32_t	tx_count	Current valid number of Tx channel.
	uint32_t	other_count	Reserved. This value is always return 0.
	uint32_t	combined_count	Reserved. This value is always return 0.

4.6.13   eavb_avbtool_gstrings

Table 4.17   struct eavb_avbtool_gstrings

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_gs trings	uint32_t	string_set	The string set id.
	uint32_t	len	The number of strings in the string set.
	uint8_t	data[0]	An array of string set. (variable length)

4.6.14   eavb_avbtool_sset_info

Table 4.18   struct eavb_avbtool_sset_info

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_ss et_info	uint64_t	sset_mask	The each bit selects a string set to do query and returned string set.
	uint32_t	data[0]	The number of string set values (variable length).

4.6.15   eavb_avbtool_stats

Table 4.19   struct eavb_avbtool_stats

Structure name	Member		
	Type	**Name**	**Description**
eavb_avbtool_st ats	uint32_t	n_stats	Number of statistics counters.
	uint64_t	data[0]	Statistics count value. (variable length)

4.6.16   eavb_dma_alloc

Table 4.20   struct eavb_dma_alloc

Structure name	Member		
	Type	**Name**	**Description**
eavb_dma_alloc	uint32_t	dma_paddr	The physical address value of a memory page.
	void *	dma_vaddr	The virtual address pointer of a memory page.
	unsigned int	mmap_size	The size of a memory page.

4.7   Global Variables and Constants

4.7.1   Global Variables

There are no global variables for this module.

4.7.2   Global Constants

The following table shows the global constants for user application.

Table 4.21   List of Global constants

Name	Value	Remark
EAVB_ENTRYVECNUM	2	-
EAVB_TXSTREAMNUM	16	-
EAVB_RXSTREAMNUM	16	-
EAVB_GSTRING_LEN	32	-

4.8 Debug facility

4.8.1 Trace Events

The following table shows the trace events for kernel function tracer.

Table 4.22 List of Trace events

Name	Description
avb_state	The tracing event of the states of AVB Streaming driver.
avb_event	The tracing event of the events of AVB Streaming driver.
avb_entry	The tracing event of the packet entry data in the AVB Streaming driver.
avb_entry_accept	The tracing event of the packet entry data in the AVB Streaming driver.
avb_desc	The tracing event of the DMA descriptor.
avb_lock	The tracing event of the lock control in the AVB Streaming driver.

5.   Integration

5.1   Directory Configuration

The directory configuration is shown below.

```
─── avb-streaming ─────── Kconfig       : Kernel configuration file

                      ─── Makefile      : Make file

                      ─── ravb_streaming_main.c     : AVB Streaming source file

                      ─── ravb_streaming.h          : AVB Streaming header file

                      ─── ravb_streaming_avbtool.c  : avbtool source file

                      ─── ravb_streaming_avbtool.h  : avbtool header file

                      ─── ravb_streaming_version.h  : driver version file

                      ─── ravb_streaming_sysfs.c    : sysfs source file

                      ─── ravb_streaming_sysfs.h    : sysfs header file

                      ─── ravb_streaming_trace.h    : ftrace header file

                      ─── ravb_proc.c          : procfs source file

                      ─── ravb_eavb.h          : AVB Streaming header file
```

Figure 5.1   Directory configuration

5.2 Integration Procedure

5.2.1 Build kernel module only

Following operation is to build loadable modules.

step1. Build Streaming driver as linux loadable module.

```
# cd avb-streaming
# make KERNEL_SRC=/path/to/linux
```

step2. Copy kernel modules. Next step integration, show 5.2.3

```
# cp ravb_streaming.ko ${ROOTFS_PATH}
# cp ravb_proc.ko ${ROOTFS_PATH}
```
${ROOTFS_PATH} rootfs of target

5.2.2 Build Streaming driver with the linux kernel

Following operation is to build Streaming driver with Linux Kernel.

step1. Copy source file to the linux kernel source.

```
# cp -r ${MODULE_SOURCE}/ ${KERNEL_SOURCE}/drivers/staging/
```

${MODULE_SOURCE} Streaming driver source directory
${KERNEL_SOURCE} linux kernel source directory

step2. Edit ${KERNEL_SOURCE}/drivers/staging/Kconfig

```
menuconfig STAGING
        bool "Staging drivers"
        default n
        ---help---
  ...

  source "drivers/staging/vc04_services/Kconfig"                    add this line
  source "drivers/staging/avb-streaming/Kconfig"

  endif # STAGING
```

step3. Edit ${KERNEL_SOURCE}/drivers/staging/Makefile

```
# Makefile for staging directory

  obj-y                              += media/
  obj-$(CONFIG_SLICOSS)              += slicoss/
...                                                      add this line
  obj-$(CONFIG_BCM2708_VCHIQ)        += vc04_services/
  obj-$(CONFIG_RAVB_STREAMING)       += avb-streaming/
```

step4. To enable the function of this module, make the following setting with Kernel Configuration. Then, rebuild kernel.

```
Device Drivers    --->
        [*] Staging drivers --->
                [*] AVB Streaming --->
                        <*> Ethernet AVB Streaming API support (#5)
                        [ ]    Enable descriptor tracing in AVB Streaming API (#6)
                        [ ]    Enable lock tracing in AVB Streaming API (#6)
                        <*>    Enable extended statistics (#5)
```

Notes: #5　　　　　This example, generate a kernel built-in object. If select the <M>, build as loadable module..
Notes: #6　　　　　These configuration are depends on FUNCTION_TRACER=y.

step5. If select <M> with kernel configuration, to copy kernel modules. Next step integration, show 5.2.3

```
# cd ${KERNEL_SOURCE}/drivers/staging/avb-streaming/
# cp ravb_streaming.ko ${ROOTFS_PATH}
# cp ravb_proc.ko ${ROOTFS_PATH}
```

${ROOTFS_PATH} rootfs of target

5.2.3　Integration device

Following operation is to integrate Streaming driver on target boards, and to check the integrated devices.

Integrate builded modules on target boards (both talker and listener boards).

```
# insmod ravb_streaming.ko
# insmod ravb_proc.ko
```

To check devices of Streaming driver, executes following commands.

```
# lsmod
Modules              Size       Used by
ravb_proc            12335      0
ravb_streaming       44523      1
```

5.3   Option Setting

5.3.1   Module Parameters

The module parameter that is defined in this module is shown below. This file is created under /sys of the file system.

Table 5.1   Module parameters

Module parameters	Explanation
/sys/module/ravb_streaming/parameters/major	Current device major number of the streaming devices. An initial value is 0 (dynamic allocate).
/sys/module/ravb_streaming/parameters/interface	Ethernet interface for streaming driver. An initial value is "eth0".

5.3.2   Kernel Parameters

There are no kernel parameters.

6.   Appendix

6.1   avbtool

The avbtool that is helper application in this module. How to use avbtool show below.

avbtool [option*] [device] [suboption]

*show table5.1

Table 6.1   avbtool option

Option	Explanation

-r , --rxparams	Set/Show Rx parameters.
	To show separation filtetr is below.
	    $ avbtool -r [device] streamid
	To set separation filter is below.
	    $ avbtool -r [device] streamid [xx:xx:xx:xx:xx:xx:xx:xx]

-g , --show-ring	Query Rx/Tx ring parameters.
-i , --driver	Show driver information.
-S , --statistics	Show statistics.
-l , --show-channels	Query channels
--version	Show version.
-h , --help	Show help.

CONFIDENTIAL

REVISION HISTORY	Ethernet AVB Software AVB Streaming Driver User's Manual: Software		

Rev.	Date	Description	
		Page	Summary
0.10	Oct. 26, 2015	-	First edition issued
0.20	Apr. 22, 2016	11	4.2 List of file operation system call for AVB Streaming driver Change chapter title to "4.2 List of file operation system call for AVB Streaming driver"
		12	4.3 List of kernel driver interface for AVB Streaming driver Add new chapter
		13	4.4.1 open Add error number - ENOMEM: Cannot allocate memory
		13	4.4.2 close Add error number - EINVAL: No such device
		14	4.4.3 read Add error number - EINVAL: Invalid argument / Count is not multiple of the entry size - EFAULT: Cannot access userspace - ENOMEM: Count is greater than internal buffer,if EAVB_BLOCK_WAITALL mode
		14	4.4.4 write Change error number - EINVAL: Invalid argument / Count is not multiple of the entry size Add error number - EFAULT: Cannot access userspace - EINTR: Interrupted Delete error number - ENOMEM
		15	4.4.6 mmap Add error number - EFGAIN: Virtual memory mapping faild. - EINVAL: Invalid argument
		16	4.4.7 ioctl(EAVB_SETTXPARAM) Change error number - EINVAL: Invalid argument Add error number - EPERM: This device isn't transmission type - EFAULT: Cannot access userspace - EBUSY: Hardware update disabled state
		16	4.4.8 ioctl(EAVB_GETTXPARAM) Change error number - EINVAL: Invalid argument Add error number - EPERM: This device isn't transmission type - EFAULT: Cannot access userspace
		17	4.4.9 ioctl(EAVB_SETRXPARAM) Change error number - EINVAL: Invalid argument - EPERM: This device isn't reception type / Invalid hardware state Add error number - EFAULT: Cannot access userspace
		17	4.4.10 ioctl(EAVB_GETRXPARAM) Change error number - EINVAL: Invalid argument Add error number - EPERM: This device isn't reception type - EFAULT: Cannot access userspace
		18	4.4.11 ioctl(EAVB_GETCBSINFO) Add error number

C - 1

			- EFAULT: Cannot access userspace
		19	4.4.12 ioctl(EAVB_SETOPTION) Change error number - EINVAL: Invalid argument Add error number - EFAULT: Cannot access userspace
		19	4.4.13 ioctl(EAVB_GETOPTION) Change error number - EINVAL: Invalid argument Add error number - EFAULT: Cannot access userspace
		20	4.4.14 ioctl(EAVB_MAPPAGE) Add error number - EFAULT: Cannot access userspace
		20	4.4.15 ioctl(EAVB_MAPPAGE) Add error number - EFAULT: Cannot access userspace
		21	4.4.16 ioctl(EAVB_GDRVINFO) Add error number - EFAULT: Cannot access userspace
		21	4.4.17 ioctl(EAVB_GRINGPARAM) Add error number - EFAULT: Cannot access userspace
		22	4.4.18 ioctl(EAVB_GSSET_INFO) Change error number - ENOMEM: Cannot allocate memory Add error number - EFAULT: Cannot access userspace
		22	4.4.19 ioctl(EAVB_GSTRINGS) Add error number - EFAULT: Cannot access userspace - ENOMEM: Cannot allocate memory
		23	4.4.20 ioctl(EAVB_GSTATS) Change error number - ENOMEM: Cannot allocate memory Add error number - EFAULT: Cannot access userspace
		23	4.4.21 ioctl(EAVB_GCHANNELS) Add error number - EFAULT: Cannot access userspace
		24 - 30	4.5 Kernel driver interface specification for AVB Streaming driver Add new chapter
		33 - 34	4.6 Structure - Add struct eavb_entry_num - Add struct ravb_streaming_kernel_if
		40	5.1 Directory Configuration Update file list
		41 - 42	5.2 Integration Procedure - Add new chapter 5.2.1 Build kernel module only - Procedure of build Streaming driver with linux kernel, which is a conventional way, move to chapter 5.2.2 - Add new chapter 5.2.3 Integration device
		43	5.3.1 Module Parameters Update of Table 5.1 - Add new parameter "interface", that is Ethernet interface for streaming driver
		44	6 Appendix Add new chapter
0.30	May. 20, 2016	1	1.1 Overview Add R-Car M3 support
		1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.

		4	2.1 Hardware Environment Add R-Car M3 support in Table 2.1
0.40	Aug. 24, 2016	1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		4	2.1 Hardware Environment Add R-CarM3-SiP Board support in Table 2.1
0.50	Dec. 22, 2016	12	4.3 List of kernel driver interface for AVB Streaming driver Update of Table 4.4 - Add new kernel driver interface : get_linkspeed.
		30	4.5 Kernel driver interface Specification fot AVB Streaming driver Add chapter - 4.5.12 get_linkspeed.
		35	4.6 Structure Update of Table 4.13 - Add member of structure : get_linkspeed.
0.60	Mar. 15, 2017	1	1.2.1 Standard Update of Table 1.1 - Update the version of IEEE Std 1722
		1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		2	1.3 Restrictions Remove restriction that is already fixed.
		42	5.2.2 Build Streaming driver with the linux kernel Fixes copy and paste problem of example of the command below "cp -r ${MODULE_SOURCE}/ ${KERNEL_SOURCE}/drivers/staging"
		45	6.1 avbtool Update of Table 6.1 - Fixes copy and paste problem of example of the command below "avbtool -r [device] streamed"
1.00	Jul. 12, 2017	1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		42	5.2.2 Build Streaming driver with the Linux kernel - Update examples of the Kconfig and Makefile.
1.10	Nov. 14, 2017	All	Add R-Car M3N support.
		1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
1.20	Jan. 29, 2018	-	Update Arm trademark
		1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
1.30	Feb. 22, 2018	All	Add R-Car E3 support.
		10	4 External Interface Update of Table 4.1 Device Node - Change the range of Major number 240-254 to 234-254, 384-511.
1.40	Sep. 26, 2018	1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
2.00	Dec. 25, 2018	1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		4	2.1 Hardware Environment Update of Table 2.1 - Add Salvator-XS Board support. - Update name of R-CarE3 evaluation board from Ebisu Board to Ebisu-4D Board.
2.10	Apr. 06, 2021	-	Update AddressList
2.20	Aug. 16, 2021	1,4	Add R-Car D3 support
2.30	Dec. 01, 2021	-	Update Notice page

			1	Table 1.2 Related documents: Update version of "R-Car Series, 3rd Generation User's Manual: Hardware": From v1.00 to v2.30 (Date: Aug 2021)

CONFIDENTIAL

RENESAS

Ethernet AVB Software
AVB Streaming Driver