User's Manual

RENESAS

CONFIDENTIAL

Ethernet AVB Software Media Streaming Engine

User's Manual: Software

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

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.2.31 Dec. 2023

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/.

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

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Overview

1.1 Overview

This manual explains the Media Streaming Engine module (MSE) that is a middleware 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 MSE.

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 802.1Q-2014	IEEE STANDARDS ASSOCIATION	IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks	-	19 December 2014
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
IEEE P1722-rev1/D13	IEEE STANDARDS ASSOCIATION	IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks	Draft 13	July 2014
IETF RFC 6184	Internet Engineering Task Force	RTP Payload Format for H.264 Video	-	May 2011
IETF RFC 2435	Internet Engineering Task Force	RTP Payload Format for JPEG-compressed Video	-	October 1998
IEC 61883-4	International Electrotechnical Commission	Consumer audio/video equipment - Digital interface - Part 4: MPEG2-TS data transmission	-	August 2004
IEC 61883-6	International Electrotechnical Commission	Consumer audio/video equipment - Digital interface - Part 6: Audio and music data transmission protocol	-	October 2005

1.2.2 Related Document

The following table shows the document related to MSE.

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
-	Renesas Electronics	Linux Interface Specification GStreamer User's Manual: Software	Rev.2.30	Aug. 16, 2021

1.3 Restrictions

- Nothing.

1.4 Terminology

The following table shows the terminology related to this module.

Table 1.3 Terminology

Terms	Explanation
MAC	Media Access Control
VLAN	Virtual LAN (IEEE Std 802.1Q)
RTP	Real-time Transfer Protocol (RFC 3550)
AVB	Audio Video Bridging
CBS	Credit Based Shaper
AVTP	Audio Video Transport Protocol (IEEE Std 1722)
AAF	AVTP Audio Format (IEEE Std 1722)
CVF	Compressed Video Format (IEEE Std 1722)
CRF	Clock Reference Format (IEEE Std 1722)
PTP	Precision Time Protocol (IEEE Std 1588)
gPTP	Generalized Precision Time Protocol (IEEE Std 802.1AS)
H.264	ITU-T H.264, ISO/IEC MPEG-4 AVC
MJPEG	Motion JPEG (RFC 2435)
MPEG2-TS	MPEG2 transport stream (ISO/IEC 13818-1 / ITU-T Recommendation H.222.0)
PCR	Program Clock Reference
PID	Packet Identifier
PCM	Pulse Code Modulation
sysfs	Virtual file system provided by the Linux kernel
ALSA	Advanced Linux Sound Architecture
V4L2	Video for Linux 2
GStreamer	Open source multimedia framework
VSP	Video Signal Processor

2. Operating Environment

2.1 Hardware Environment

The hardware environment premises on "Chapter 1.3 Hardware environment" in Ethernet AVB Software Start-Up Guide.

2.2 Module Configuration

The following figure shows the configuration of this module.

- MSE consist of core and adapters.
- MSE core provides basic function AVB Streaming.
- Adapters provides interfaces of V4L2, ALSA or the Streaming Driver.
- Media clock recovery using Media Clock Recovery Handler (MCH).

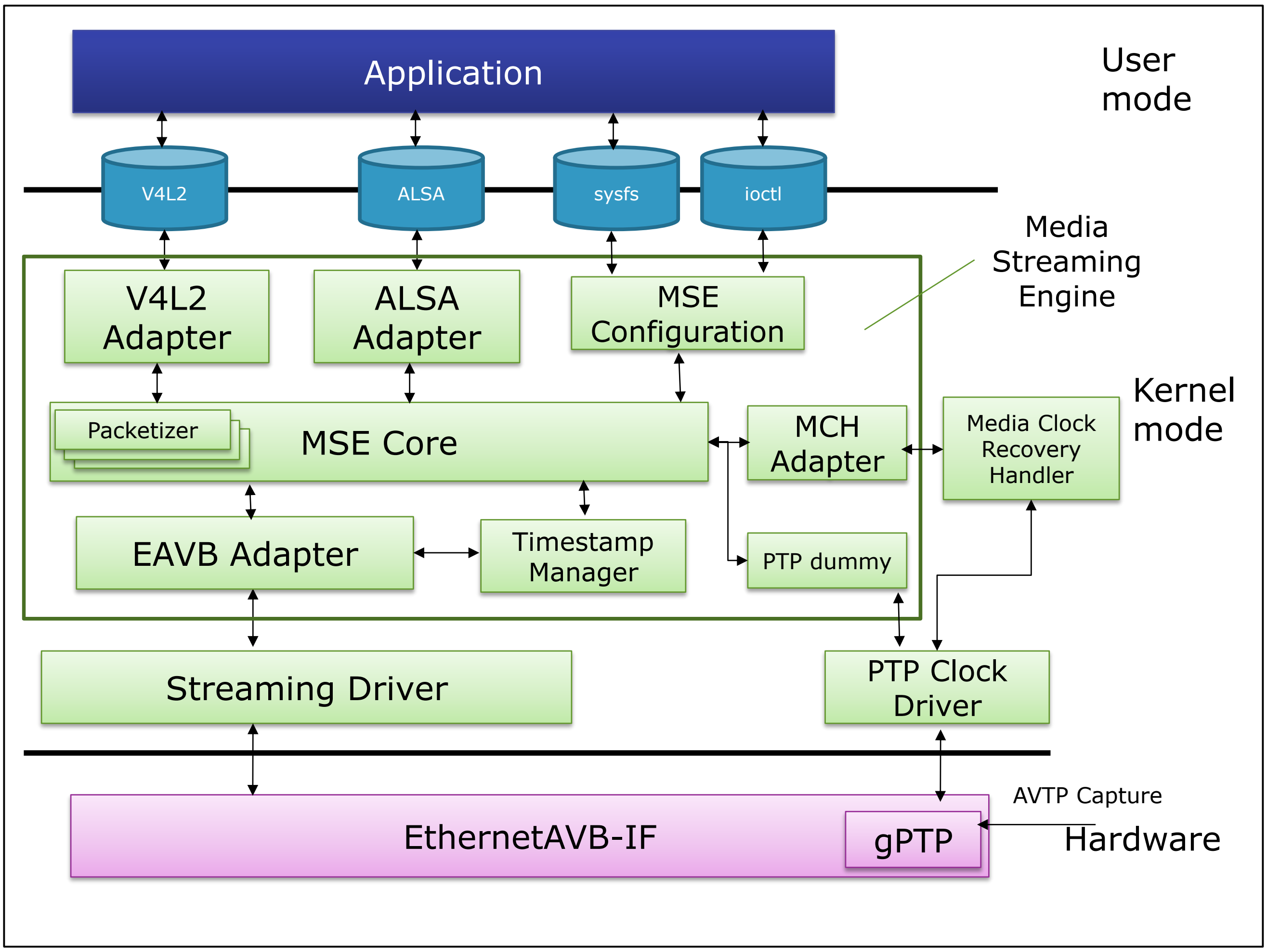


Figure 2-1 Module configuration

3. Function

MSE Core has following functions:

- I/F(API in Kernel) functions for Media Adapter and Network Adapter
- Functions convert data among Adapters (Packetize, De-packetize)
- I/F such as sysfs, ioctl for configuring each functions
- Media clock recovery

Media Adapter has following functions:

- Providing standard Audio I/F on Linux by ALSA Adapter
- Providing standard Video I/F on Linux by V4L2 Adapter

Network Adapter has following functions:

- Sending and receiving AVTP streams by EAVB Adapter linking up with Streaming Driver

MCH Adapter has following functions:

- Providing I/F for gPTP Timer and Media Clock Recovery functions

Note:

- MSE core returns silence data (value 0) when no audio data is received.
- MSE core wait PTP time to output receiving data at AVTP timestamp when using media clock recovery.

Following figures are the relationship diagram of each module.

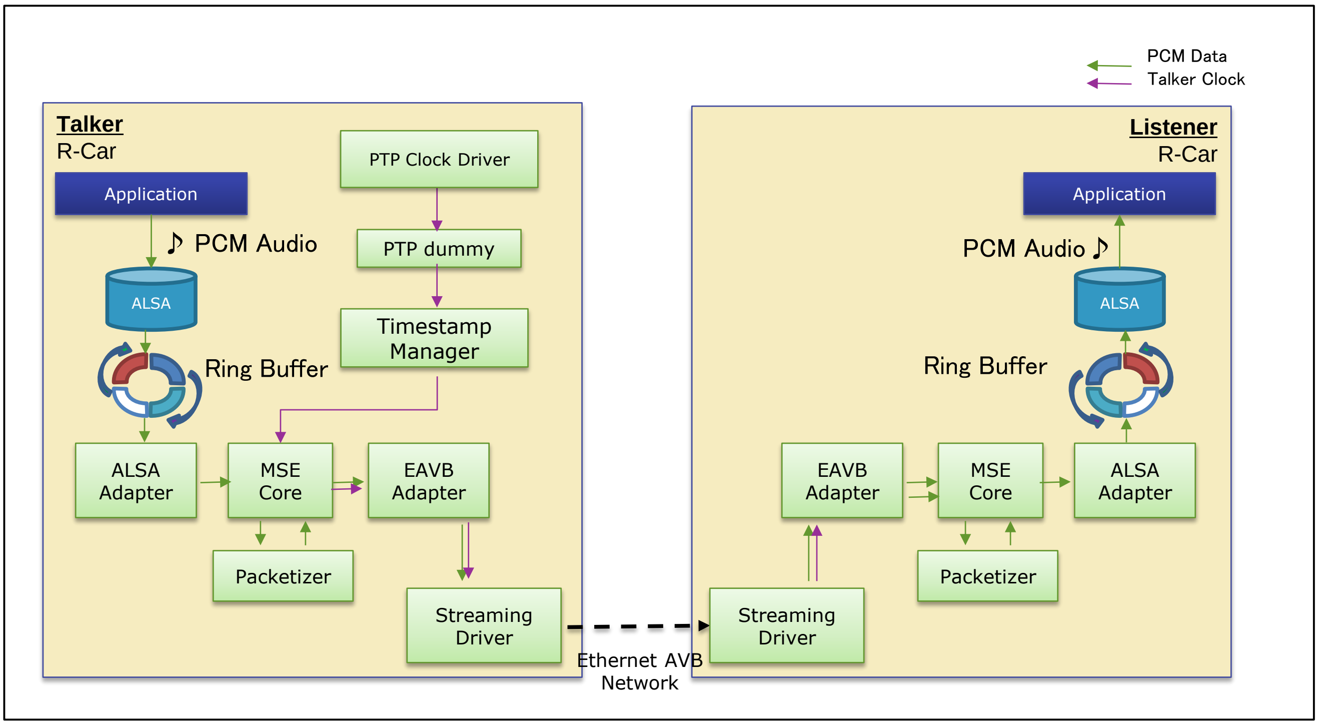


Figure 3-1 Module relationship (ALSA)

Execute Media clock recovery

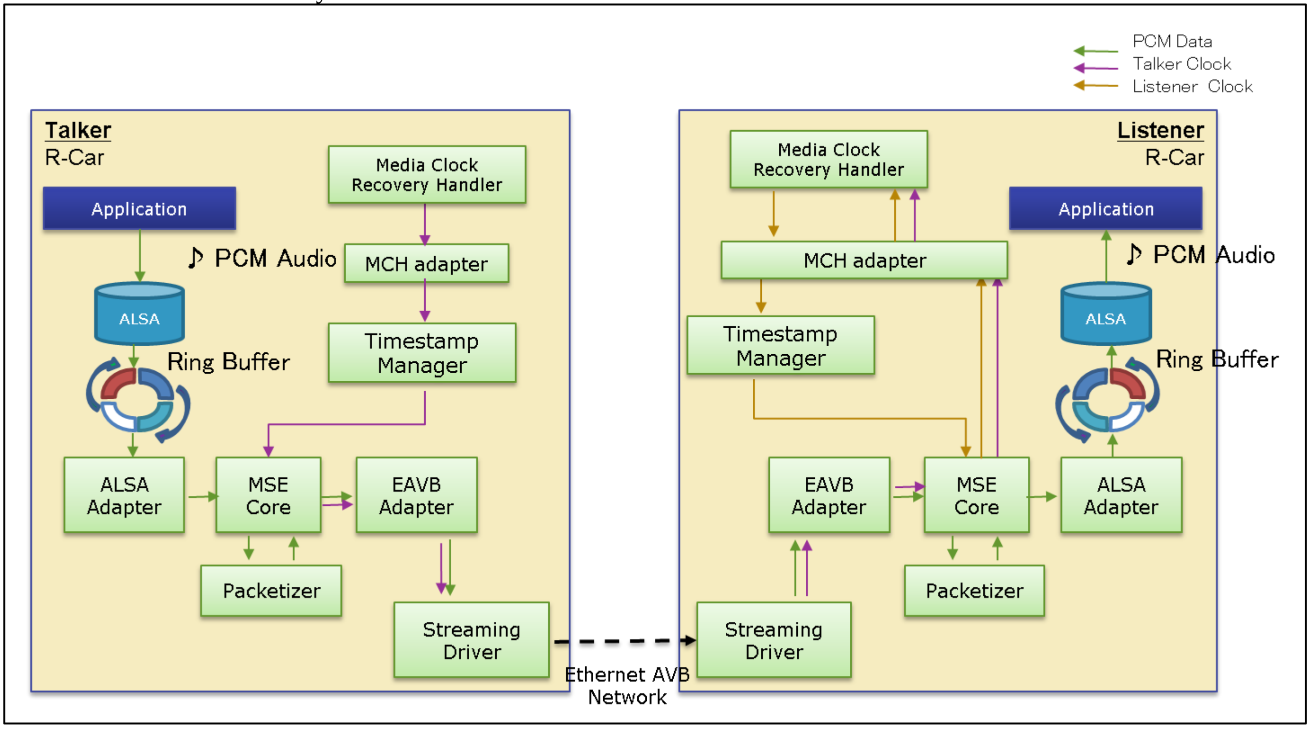


Figure 3-2 Module relationship (ALSA) with MCH

Media clock recovery from listener (using CRF)

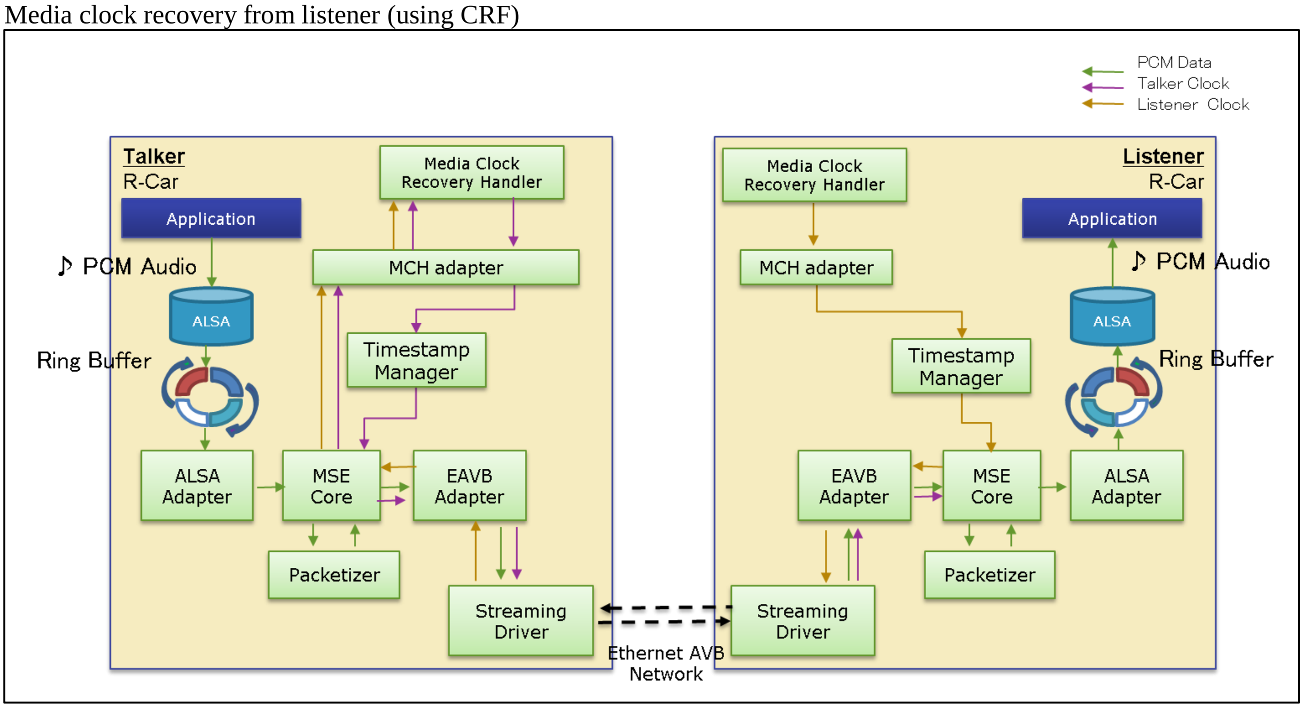


Figure 3-3 Module relationship (ALSA) with CRF stream

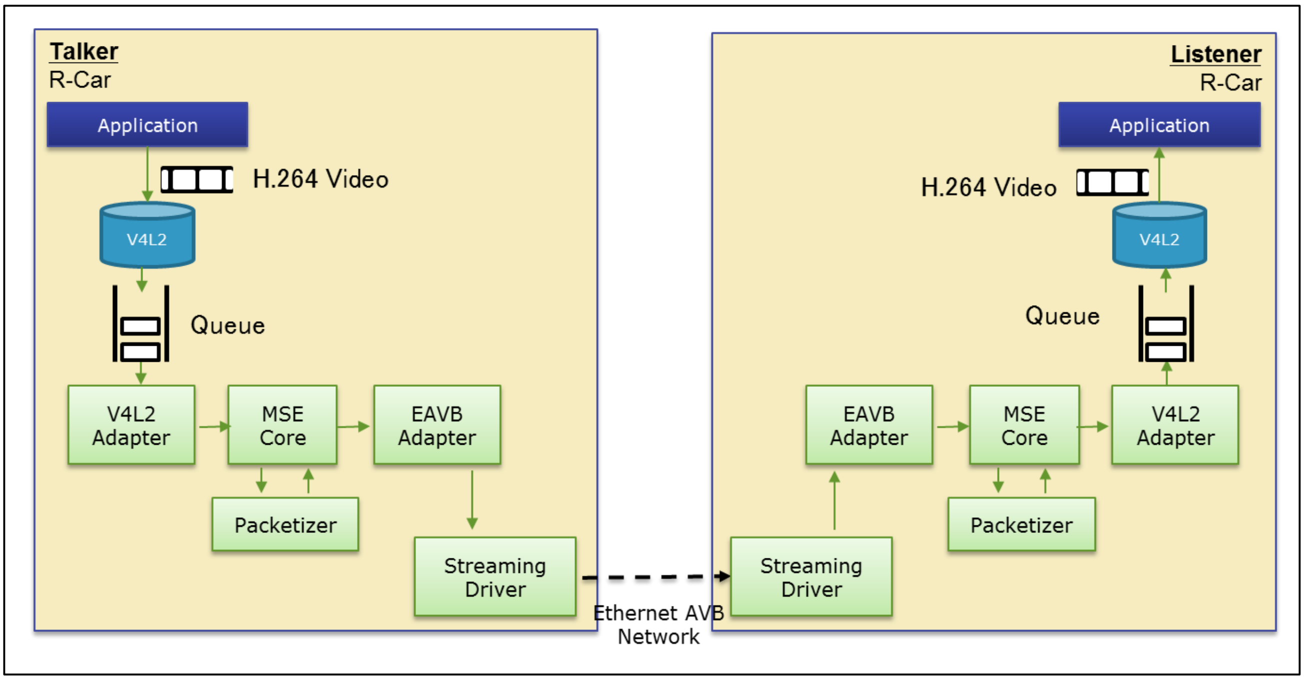


Figure 3-4 Module relationship (V4L2)

3.1 ALSA Adapter

ALSA adapter provides the pseudo driver function that conforms to ALSA. This driver controls streaming process for audio PCM data by Linux standard I/F. ALSA adapter supports following formats.

Table 3.1 ALSA Adapter capabilities

Sample format (depend on packet format)	S16_LE/S16_BE
	S24_LE/S24_BE
	S32_LE/S32_BE
	S18_3LE/S18_3BE
	S20_3LE/S20_3BE
	S24_3LE/S24_3BE
Sampling rate (depend on packet format)	8000 to 192000 Hz (#2)
Channels	1 to 24 (#2)
Buffer size	max 65536 byte (#1)
Period size	64 to 8192 byte (#1)
Number of period	2 to 32

Notes: #1 Buffer size must be able to divide by Period size.

Notes: #2 Cannot set the combination of values that exceeds the maximum AVTP payload size.
AVTP payload size is calculated from format, sampling rate and channels.

3.2 V4L2 Adapter

V4L2 adapter provides the pseudo driver function that conforms to V4L2. This driver controls streaming process for video data by Linux standard I/F. V4L2 adapter supports following formats.
In the case of MPEG2-TS, one buffer should include at least three TS packets, since the V4L2 adapter check the position of sync byte(0x47) in the first three TS packets in order to confirm whether the format is TS or M2TS.

Table 3.2 V4L2 adapter capabilities (H.264)

Video format	H.264 AVC H.264 bytestream
Number of buffers	2 to 8
Picture size	Width: 80 to 3840 (multiple of 2) Height: 80 to 2160 (multiple of 2)
Framerate	1/1 to 60000/1001
Interlaced/ Progressive	Progressive

Table 3.3 V4L2 adapter capabilities (MJPEG)

Video format	MJPEG
Number of buffers	2 to 8
Picture size	Width: 8 to 2040 (multiple of 8) Height: 8 to 2040 (multiple of 8)
Framerate	1/1 to 60000/1001
Interlaced/ Progressive	Progressive

Table 3.4 V4L2 adapter capabilities (MPEG2-TS)

Video format	MPEG2-TS - MPEG-2 Transport Stream - Blu-ray Disc Audio-Video (BDAV) MPEG-2 Transport Stream (including PCR) (#3)
Number of buffers	2 to 8

Notes: #3 MPEG2-TS(TS) and BDAV(M2TS) are able to accept as input data. But, MSE output only MPEG2-TS(TS) regardless of either input.

3.3 EAVB Adapter

EAVB Adapter provides the function for AVTP streaming via EthernetAVB-IF by using Streaming driver.

3.4 MCH Adapter

MCH Adapter provides the clock recovery function via EthernetAVB-IF by using Media Clock Recovery Handler.

3.5 Packetize/De-packetize

Packetize/De-packetize provides the function that converts Audio/Video multimedia data to AVTP format.
Table3.1 shows the list of Converting formats.

Table 3.5 Packetize/De-packetize format

Multimedia data	AVTP Stream
PCM Sound	IEC 61883-6 AM824 format (IEC61883-6)
	PCM AVTP Audio Format (AAF PCM)
H.264 AVC H.264 bytestream	H.264 Compressed Video Format without h264_timestamp field (CVF H264 D13)
	H.264 Compressed Video Format (CVF H264)
Motion JPEG	MJPEG Compressed Video Format (CVF MJPEG)
MPEG2-TS	IEC 61883-4 MPEG2-TS (IEC 61883-4)

3.5.1 Ethernet AVB Packet Format

MSE supports following packet formats.

Table 3.6 Audio Packet format (IEC 61883-6)

Packet format	IEC 61883-6 AM824 format (IEC61883-6) (MBLA: Multi Bit Linear Audio 16bit, 20bit, 24bit)
Sampling rate	32000Hz, 44100Hz, 48000Hz, 88200Hz, 96000Hz, 176400Hz, 192000Hz
Bit depth	16, 20, 24bit integer
Channels	1 to 24

Table 3.7 Audio Packet format (AAF PCM)

Packet format	PCM AVTP Audio Format (AAF PCM) (INT_16, INT_24, INT_32)
Sampling rate	8000Hz, 16000Hz, 32000Hz, 44100Hz, 48000Hz, 88200Hz, 96000Hz, 176400Hz, 192000Hz
Bit depth	16, 24, 32bit integer
Channels	1 to 24

Table 3.8 Video Packet format (H.264)

Packet format	H.264 Compressed Video Format without h264_timestamp field (CVF H264 D13)
	H.264 Compressed Video Format (CVF H264)
Encode format	H.264

Table 3.9 Video Packet format (MJPEG)

Packet format	MJPEG Compressed Video Format (CVF MJPEG)
Encode format	MJPEG (supports only in-band quantization tables that defined in RFC-2435 3.1.4, and not support Restart Marker header defined by RFC-2435 3.1.7)
Picture size	Width: 8 to 2040 (multiple of 8) Height: 8 to 2040 (multiple of 8)

Table 3.10 Video Packet format (MPEG2-TS)

Packet format	IEC61883-4 MPEG2-TS (IEC61883-4)
Encode format	MPEG2-TS

Table 3.11 Control Packet format (CRF)

Packet format	CRF Audio Sampling Timestamp

3.6 Network Traffic Shaping

MSE set parameter of the H/W CBS (Credit Based Shaper) via AVB Streaming Driver.
User is able to set traffic bandwidth for each AVTP stream.

In audio stream:
To set media_audio_config/samples_per_frame using sysfs or ioctl.
It is the number of audio sample in an AVTP frame.
Using default value 0, MSE calculates samples_per_frame automatically from ALSA settings.

In video stream:
To set media_video_config/bitrate using sysfs or ioctl.
It is the max video bitrate of payload data.

In MPEG2-TS stream:
To set media_mpeg2ts_config/bitrate using sysfs or ioctl.
It is the max MPEG2-TS bitrate of payload data.

MSE calculates the bandwidth to use above configuration parameters and overhead of Ethernet and AVTP protocol.

4. External Interface

This chapter shows 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/mse[0-n]	234-254,384-511 (#4)	0-n

Notes: #4 It is dynamically allocated.

4.1 Sysfs Interface

MSE has a sysfs interface that is used for changing operating parameters from a user land application.

Note that directory configuration of sysfs interface has been significantly updated from v0.30 to v0.40. For this reason, tables for comparison with the v0.30 specifications are shown Appendix 8.1.

Configuration parameters of sysfs interface are shown below.

Table 4.2 List of sysfs device node for MSE

Device name	Device node	Description
ravb_mse	/sys/class/ravb_mse/mse[0-n]	The module parameter for MSE

Table 4.3 List of sysfs common configuration parameter for MSE

Name	Value	Default	Description
network_device/module_name	ravb	ravb	Network name of MSE
network_device/device_name_tx	avb_tx0 avb_tx1	avb_tx0	Device name of Streaming Driver (TX)
network_device/device_name_rx	avb_rx[0-15]	avb_rx0	Device name of Streaming Driver (RX)
avtp_tx_param/dst_mac	HEX strings	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	HEX strings	769050000000	Ethernet source MAC address
avtp_tx_param/vlan	0 to 4094	2	VLAN ID
avtp_tx_param/priority	0 to 7	3	AVB Stream priority
avtp_tx_param/uniqueid	0 to 65535	1	AVB Stream unique id
avtp_rx_param/streamid	HEX strings	7690500000000001	AVB Stream id
ptp_config/deviceid	0 to 4294967295	0	gPTP device number

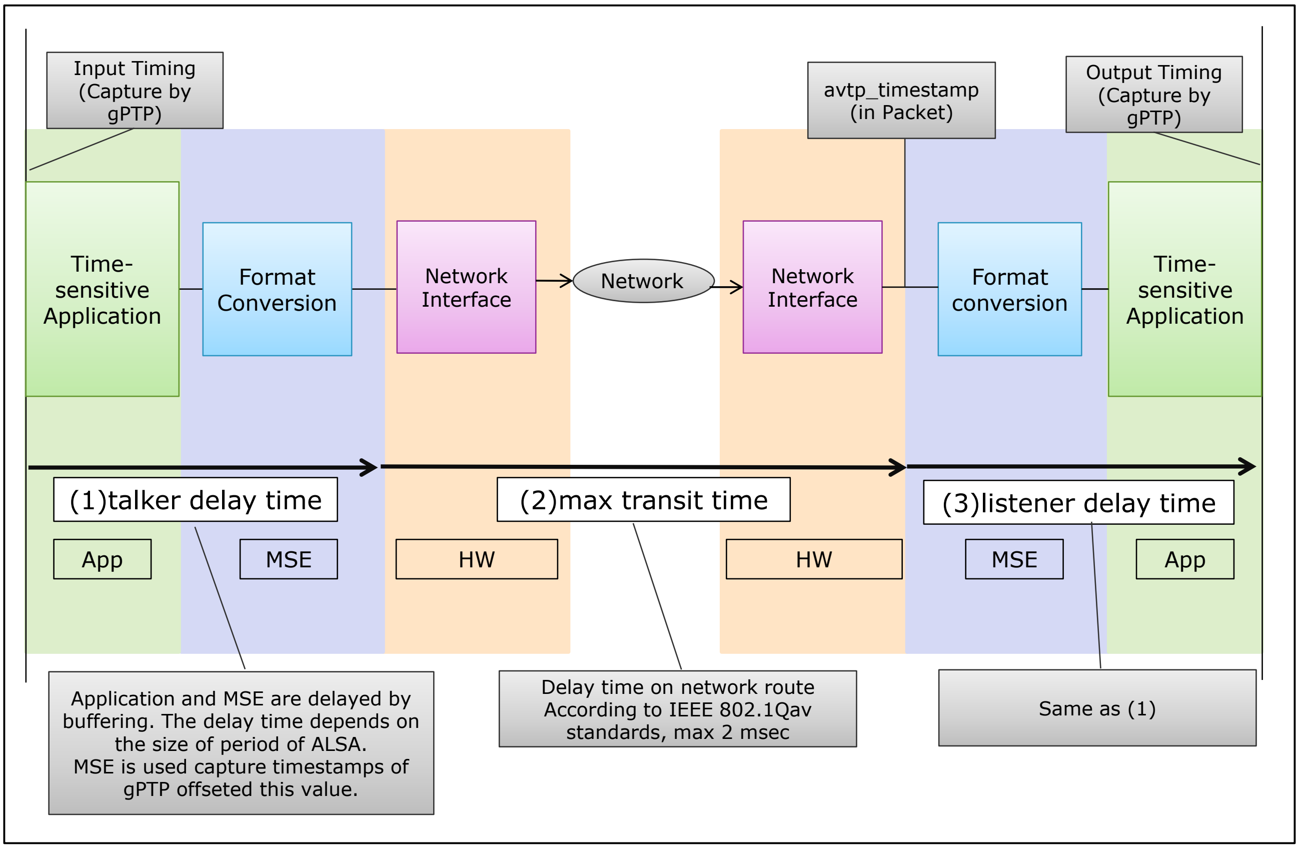

Input Timing (Capture by gPTP)
avtp_timestamp (in Packet)
Output Timing (Capture by gPTP)
Time-sensitive Application
Format Conversion
Network Interface
Network
Network Interface
Format conversion
Time-sensitive Application
(1)talker delay time
(2)max transit time
(3)listener delay time
App
MSE
HW
HW
MSE
App
Application and MSE are delayed by buffering. The delay time depends on the size of period of ALSA. MSE is used capture timestamps of gPTP offseted this value.
Delay time on network route According to IEEE 802.1Qav standards, max 2 msec
Same as (1)

Figure 4-1 overall outline procedure of time-sensitive application

Table 4.4 List of sysfs audio configuration parameter for MSE

Name	Value	Default	Description
info/type	audio	(fixed)	MSE device type
info/device	ALSA device name	(fixed)	MSE ALSA Card name
network_device/device_name_tx_crf	avb_tx0 avb_tx1	avb_tx0	Device name of Streaming Driver (TX)
network_device/device_name_rx_crf	avb_rx[0-15]	avb_rx0	Device name of Streaming Driver (RX)
packetizer/name	aaf_pcm iec61883-6	aaf_pcm	packet format name
media_audio_config/samples_per_frame	0, 1 to 738	0	samples of payload per frame 0: auto config by MSE
media_audio_config/crf_type	not use, tx, rx	not use	not use: not use CRF tx: send CRF rx: receive CRF (see Table 4.5)
ptp_config/type	ptp capture	ptp	ptp: ptp current time capture: ptp capture time(use device timing) (audio only)
ptp_config/capture_ch	0 to 16	2	gPTP capture channel
ptp_config/capture_freq	0 to 4294967295	300	gPTP capture timestamps per Second
ptp_config/recovery_capture_freq	0, 1	0	1: need to be set when capture frequency is not fixed (ex. using SRC, Audio device)
mch_config/enable	0, 1	0	1: mch enable(see Table 4.5)
delay_time/max_transit_time_ns	0 to 4294967295	2000000	Transit time for AVTP network (nanoseconds) Refer to Figure 4-1 (2).
delay_time/tx_delay_time_ns	0 to 1073741824	2000000	Talker side process time (nanoseconds) Refer to Figure 4-1 (1).
delay_time/rx_delay_time_ns	0 to 1073741824	2000000	Listener side process time (nanoseconds) Refer to Figure 4-1 (3).
avtp_tx_param_crf/dst_mac	HEX strings	91e0f0000e80	Ethernet destination MAC address of CRF
avtp_tx_param_crf/src_mac	HEX strings	769050000000	Ethernet source MAC address of CRF
avtp_tx_param_crf/vlan	0 to 4094	2	VLAN ID of CRF
avtp_tx_param_crf/priority	0 to 7	3	AVB Stream priority of CRF
avtp_tx_param_crf/uniqueid	0 to 65535	1	AVB Stream unique id of CRF
avtp_rx_param_crf/streamid	HEX strings	7690500000000001	AVB Stream id of CRF

Table 4.5 List of available combinations of CRF and MCH

Operation	MCH	CRF	function
Talker	mch_config/enable=1	media_audio_config/crf_type=rx	clock recover by receiving CRF
		media_audio_config/crf_type=tx	not available
		media_audio_config/crf_type=not use	not available
	mch_config/enable=0	media_audio_config/crf_type=not use	clock recover not use
		media_audio_config/crf_type=tx	sending CRF for clock recover other devices.
		media_audio_config/crf_type=rx	not available
Listener	mch_config/enable=1	media_audio_config/crf_type=rx	clock recover by receiving CRF
		media_audio_config/crf_type=not use	clock recover by receiving AVTP Packet
		media_audio_config/crf_type=tx	not available
	mch_config/enable=0	media_audio_config/crf_type=tx	sending CRF for clock recover other devices.
		media_audio_config/crf_type=not use	clock recover not use
		media_audio_config/crf_type=rx	not available

Table 4.6 List of sysfs video configuration parameter for MSE

Name	Value	Default	Description
info/type	video	(fixed)	MSE device type.
info/device	string	(fixed)	MSE device file path (v4l2)
packetizer/name	cvf_h264 cvf_h264_d13 cvf_mjpeg	cvf_h264	packet format name
media_video_config/bytes_per_frame	0, 128 to 1476	0	bytes of payload par per frame (#5,#6) 0: auto config by MSE
media_video_config/fps_denominator	0, 1 to 1001	0	Framerate setting for Video
media_video_config/fps_numerator	0, 1 to 60000	0	Framerate setting for Video
media_video_config/bitrate	1 to 4294967295	50000000	Encoding bitrate for Video
media_video_config/class_interval_frames	100 to 8000	4000	Frames per seconds (#7)
media_video_config/max_interval_frames	1	1	Number of frames per interval (#7)
delay_time/max_transit_time_ns	0 to 4294967295	2000000	Transit time for AVTP network (nanoseconds).
delay_time/tx_delay_time_ns	0 to 1073741824	0	Talker side process time (nanoseconds).
delay_time/rx_delay_time_ns	0 to 1073741824	0	Listener side process time (nanoseconds).

Notes: #5 bytes_per_frame is including H.264 header and FU_A header using H.264, but not including MJPEG header and Q table data using MJPEG.

Notes: #6 The payload of frame may become smaller than setting value of bytes_per_frame, in case of the frame size exceeds maximum value of ethernet.

Note: #7 Configure the CBS by frames per seconds and number of frames per interval. Set media_video_config/bitrate is 0 to make these settings are available.

Table 4.7 List of sysfs mpeg2ts configuration parameter for MSE

Name	Value	Default	Description
info/type	mpeg2ts	(fixed)	MSE device type.
Info/device	string	(fixed)	MSE device file path (v4l2)
packetizer/name	iec61883-4	iec61883-4	packet format name
media_mpeg2ts_config/tspackets_per_frame	1 to 7	7	MPEG2-TS packets of payload per frame
media_mpeg2ts_config/bitrate	1 to 4294967295	50000000	Encoding bitrate for Video
media_mpeg2ts_config/pcr_pid	0 to 8191, 8192	8192	PID of PCR packet 8192: auto (use any PCR)
media_mpeg2ts_config/transmit_mode	string	timestamp	MPEG2-TS transmit mode.
media_mpeg2ts_config/class_interval_frame	100 to 8000	4000	Frames per seconds (#8)
media_mpeg2ts_config/max_interval_frames	1	1	Number of frames per interval (#8)
delay_time/max_transit_time_ns	0 to 4294967295	2000000	Transit time for AVTP network (nanoseconds).
delay_time/tx_delay_time_ns	0 to 1073741824	0	Talker side process time (nanoseconds)
delay_time/rx_delay_time_ns	0 to 1073741824	0	Listener side process time (nanoseconds)

Note: #8 Configure the CBS by frames per seconds and number of frames per interval. Set media_mpeg2ts_config/bitrate is 0 to make these settings are available.

4.2 Ioctl Interface

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

Table 4.8 List of file operation system call for MSE

Chapter	Function name	Description	Remarks
4.2.1	open	Open the control device.	-
4.2.2	close	Close the control device.	-
4.2.3	ioctl(MSE_G_INFO)	Get the current information.	-
4.2.4	ioctl(MSE_S_NETWORK_DEVICE)	Set the network device parameter.	-
4.2.5	ioctl(MSE_G_NETWORK_DEVICE)	Get the current network device parameter.	-
4.2.6	ioctl(MSE_S_PACKETIZER)	Set the packetizer.	-
4.2.7	ioctl(MSE_G_PACKETIZER)	Get the current packetizer.	-
4.2.8	ioctl(MSE_S_AVTP_TX_PARAM)	Set the packet tx parameter.	-
4.2.9	ioctl(MSE_G_AVTP_TX_PARAM)	Get the current packet tx parameter.	-
4.2.10	ioctl(MSE_S_AVTP_RX_PARAM)	Set the packet rx parameter.	-
4.2.11	ioctl(MSE_G_AVTP_RX_PARAM)	Get the current packet rx parameter.	-
4.2.12	ioctl(MSE_S_MEDIA_AUDIO_CONFIG)	Set the audio configuration.	-
4.2.13	ioctl(MSE_G_MEDIA_AUDIO_CONFIG)	Get the current audio configuration.	-
4.2.14	ioctl(MSE_S_MEDIA_VIDEO_CONFIG)	Set the video configuration.	-
4.2.15	ioctl(MSE_G_MEDIA_VIDEO_CONFIG)	Get the current video configuration.	-
4.2.16	ioctl(MSE_S_MEDIA_MPEG2TS_CONFIG)	Set the MPEG2-TS configuration.	-
4.2.17	ioctl(MSE_G_MEDIA_MPEG2TS_CONFIG)	Get the current MPEG2-TS configuration.	-
4.2.18	ioctl(MSE_S_PTP_CONFIG)	Set the PTP configuration.	-
4.2.19	ioctl(MSE_G_PTP_CONFIG)	Get the current PTP configuration.	-
4.2.20	ioctl(MSE_S_MCH_CONFIG)	Set the MCH configuration.	-
4.2.21	ioctl(MSE_G_MCH_CONFIG)	Get the current MCH configuration.	-
4.2.22	ioctl(MSE_S_AVTP_TX_PARAM_CRF)	Set the CRF tx parameter.	-
4.2.23	ioctl(MSE_G_AVTP_TX_PARAM_CRF)	Get the current CRF tx parameter.	-
4.2.24	ioctl(MSE_S_AVTP_RX_PARAM_CRF)	Set the CRF rx parameter.	-
4.2.25	ioctl(MSE_G_AVTP_RX_PARAM_CRF)	Get the current CRF rx parameter.	-
4.2.26	ioctl(MSE_S_DELAY_TIME)	Set the delay time parameter.	-
4.2.27	ioctl(MSE_G_DELAY_TIME)	Get the current delay time parameter.	-

Table 4.9 List of ioctl structure for MSE

Chapter	Structure name	Description	Remarks
4.2.28	struct mse_info	MSE information	-
4.2.29	struct mse_network_device	Network device	-
4.2.30	struct mse_packetizer	Packetizer information	-
4.2.31	struct mse_avtp_tx_param	AVTP tx configuration	-
4.2.32	struct mse_avtp_rx_param	AVTP rx configuration	-
4.2.33	struct mse_media_audio_config	Audio configuration	-
4.2.34	struct mse_media_video_config	Video configuration	-
4.2.35	struct mse_media_mpeg2ts_config	MPEG-2 TS configuration	-
4.2.36	struct mse_ptp_config	PTP configuration	-
4.2.37	struct mse_mch_config	MCH configuration	-
4.2.38	struct mse_delay_time	Delay time parameter	-

4.2.1 open

[Function]	int open(const char *device_name, int flags)	
[Arguments]	device_name	Request the "/dev/mseX".
	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
[Description]	Open the control device.	

4.2.2 close

[Function]	int close(int fd)	
[Arguments]	fd	File descriptor
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	No such device
[Description]	Close the control device.	

4.2.3 ioctl(MSE_G_INFO)

[Function]	int ioctl(int fd, int request, struct mse_info *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_INFO
	argp	Pointer of mse_info structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current MSE device information.	

4.2.4 ioctl(MSE_S_NETWORK_DEVICE)

[Function]	int ioctl(int fd, int request, const struct mse_network_device *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_NETWORK_DEVICE
	argp	Pointer of mse_network_device structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the parameter for network adapter.	

4.2.5 ioctl(MSE_G_NETWORK_DEVICE)

[Function]	int ioctl(int fd, int request, struct mse_network_device *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_NETWORK_DEVICE
	argp	Pointer of mse_network_device structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current device parameter for network adapter.	

4.2.6 ioctl(MSE_S_PACKETIZER)

[Function]	int ioctl(int fd, int request, const struct mse_packetizer *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_PACKETIZER
	argp	Pointer of mse_packetizer structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the packetizer.	

4.2.7 ioctl(MSE_G_PACKETIZER)

[Function]	int ioctl(int fd, int request, const struct mse_packetizer *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_PACKETIZER
	argp	Pointer of mse_packetizer structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current packetizer.	

4.2.8 ioctl(MSE_S_AVTP_TX_PARAM)

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

[Arguments]	fd	File descriptor
	request	MSE_S_AVTP_TX_PARAM
	argp	Pointer of mse_avtp_tx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't transmission type
	EBUSY	Device busy
	EFAULT	Cannot access userspace

[Description] Set the streaming parameter for transmission.

4.2.9 ioctl(MSE_G_AVTP_TX_PARAM)

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

[Arguments]	fd	File descriptor
	request	MSE_G_AVTP_TX_PARAM
	argp	Pointer of mse_avtp_tx_param 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.

4.2.10 ioctl(MSE_S_AVTP_RX_PARAM)

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

[Arguments]	fd	File descriptor
	request	MSE_S_AVTP_RX_PARAM
	argp	Pointer of mse_avtp_rx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't reception type
	EBUSY	Device busy
	EFAULT	Cannot access userspace

[Description] Set the streaming parameter for reception.

4.2.11 ioctl(MSE_G_AVTP_RX_PARAM)

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

[Arguments]	fd	File descriptor
	request	MSE_G_AVTP_RX_PARAM
	argp	Pointer of mse_avtp_rx_param 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.

4.2.12 ioctl(MSE_S_MEDIA_AUDIO_CONFIG)

[Function]	int ioctl(int fd, int request, const struct mse_media_audio_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_MEDIA_AUDIO_CONFIG
	argp	Pointer of mse_media_audio_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the configuration for audio.	

4.2.13 ioctl(MSE_G_MEDIA_AUDIO_CONFIG)

[Function]	int ioctl(int fd, int request, struct mse_media_audio_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_MEDIA_AUDIO_CONFIG
	argp	Pointer of mse_media_audio_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EFAULT	Cannot access userspace
[Description]	Get the current configuration for audio.	

4.2.14 ioctl(MSE_S_MEDIA_VIDEO_CONFIG)

[Function]	int ioctl(int fd, int request, const struct mse_media_video_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_MEDIA_VIDEO_CONFIG
	argp	Pointer of mse_media_video_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't video type.
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the configuration for video.	

4.2.15 ioctl(MSE_G_MEDIA_VIDEO_CONFIG)

[Function]	int ioctl(int fd, int request, struct mse_media_video_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_MEDIA_VIDEO_CONFIG
	argp	Pointer of mse_media_video_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't video type.
	EFAULT	Cannot access userspace
[Description]	Get the current configuration for video.	

4.2.16 ioctl(MSE_S_MEDIA_MPEG2TS_CONFIG)

[Function]	int ioctl(int fd, int request, const struct mse_media_mpeg2ts_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_MEDIA_MPEG2TS_CONFIG
	argp	Pointer of mse_media_mpeg2ts_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't mpeg2ts type.
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the configuration for MPEG2-TS.	

4.2.17 ioctl(MSE_G_MEDIA_MPEG2TS_CONFIG)

[Function]	int ioctl(int fd, int request, struct mse_media_mpeg2ts_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_MEDIA_MPEG2TS_CONFIG
	argp	Pointer of mse_media_mpeg2ts_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't mpeg2ts type.
	EFAULT	Cannot access userspace
[Description]	Get the current configuration for MPEG2-TS.	

4.2.18 ioctl(MSE_S_PTP_CONFIG)

[Function]	int ioctl(int fd, int request, const struct mse_ptp_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_PTP_CONFIG
	argp	Pointer of mse_ptp_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Set the configuration for PTP.	

4.2.19 ioctl(MSE_G_PTP_CONFIG)

[Function]	int ioctl(int fd, int request, struct mse_ptp_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_PTP_CONFIG
	argp	Pointer of mse_ptp_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Get the current configuration for PTP.	

4.2.20 ioctl(MSE_S_MCH_CONFIG)

[Function]	int ioctl(int fd, int request, const struct mse_mch_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_MCH_CONFIG
	argp	Pointer of mse_mch_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the configuration for MCH.	

4.2.21 ioctl(MSE_G_MCH_CONFIG)

[Function]	int ioctl(int fd, int request, struct mse_mch_config *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_MCH_CONFIG
	argp	Pointer of mse_mch_config structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current configuration for MCH.	

4.2.22 ioctl(MSE_S_AVTP_TX_PARAM_CRF)

[Function]	int ioctl(int fd, int request, const struct mse_avtp_tx_param *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_AVTP_TX_PARAM_CRF
	argp	Pointer of mse_avtp_tx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the CRF parameter for transmission.	

4.2.23 ioctl(MSE_G_AVTP_TX_PARAM_CRF)

[Function]	int ioctl(int fd, int request, struct mse_avtp_tx_param *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_AVTP_TX_PARAM_CRF
	argp	Pointer of mse_avtp_tx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EFAULT	Cannot access userspace
[Description]	Get the current CRF parameter for transmission.	

4.2.24 ioctl(MSE_S_AVTP_RX_PARAM_CRF)

[Function]	int ioctl(int fd, int request, const struct mse_avtp_rx_param *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_AVTP_RX_PARAM_CRF
	argp	Pointer of mse_avtp_rx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the CRF parameter for reception.	

4.2.25 ioctl(MSE_G_AVTP_RX_PARAM_CRF)

[Function]	int ioctl(int fd, int request, struct mse_avtp_rx_param *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_AVTP_RX_PARAM_CRF
	argp	Pointer of mse_avtp_rx_param structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EPERM	This device isn't audio type.
	EFAULT	Cannot access userspace
[Description]	Get the current CRF parameter for reception.	

4.2.26 ioctl(MSE_S_DELAY_TIME)

[Function]	int ioctl(int fd, int request, const struct mse_delay_time *argp)	
[Arguments]	fd	File descriptor
	request	MSE_S_DELAY_TIME
	argp	Pointer of mse_delay_time structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EBUSY	Device busy
	EFAULT	Cannot access userspace
[Description]	Set the delay time for streaming.	

4.2.27 ioctl(MSE_G_DELAY_TIME)

[Function]	int ioctl(int fd, int request, struct mse_delay_time *argp)	
[Arguments]	fd	File descriptor
	request	MSE_G_DELAY_TIME
	argp	Pointer of mse_delay_time structure
[Returns]	0	Success
	-1	Error
[Error number]	EINVAL	Invalid argument
	EFAULT	Cannot access userspace
[Description]	Get the current delay time for streaming.	

4.2.28 struct mse_info

Table 4.10 struct mse_info

Structure name	Member			
	Type	Name	Default value	Description
mse_info	uint8_t	device[32]	-	Device name
	enum MSE_STREAM_TYPE	type	Audio device: MSE_STREAM_TYPE_AUDIO Video device: MSE_STREAM_TYPE_VIDEO MPEG2-TS device: MSE_STREAM_TYPE_MPEG2TS	MSE streaming type MSE_STREAM_TYPE_AUDIO: audio MSE_STREAM_TYPE_VIDEO: video MSE_STREAM_TYPE_MPEG2TS: mpeg2ts

4.2.29 struct mse_network_device

Table 4.11 struct mse_network_device

Structure name	Member			
	Type	Name	Default value	Description
mse_network_device	uint8_t	module_name[32]	ravb	Network adapter name
	uint8_t	device_name_tx[32]	avb_tx0	Device name of tx
	uint8_t	device_name_rx[32]	avb_rx0	Device name of rx
	uint8_t	device_name_tx_crf[32]	avb_tx0	Device name of tx for CRF
	uint8_t	device_name_rx_crf[32]	avb_rx0	Device name of rx for CRF

4.2.30 struct mse_packetizer

Table 4.12 struct mse_packetizer

Structure name	Member			
	Type	Name	Default value	Description
mse_packetizer	enum MSE_PACKETIZER	packetizer	Audio device: MSE_PACKETIZER_AAF_PCM Video device: MSE_PACKETIZER_CVF_H264 MPEG2-TS device: MSE_PACKETIZER_IEC61883_4	Packetizer name MSE_PACKETIZER_AAF_PCM: aaf_pcm MSE_PACKETIZER_IEC61883_6: iec61883-6 MSE_PACKETIZER_CVF_H264: cvf_h264 MSE_PACKETIZER_CVF_H264_D13: cvf_h264_d13 MSE_PACKETIZER_CVF_MJPEG: cvf_mjpeg MSE_PACKETIZER_IEC61883_4: iec61883-4

4.2.31 struct mse_avtp_tx_param

Table 4.13 struct mse_avtp_tx_param

Structure name	Member			
	Type	Name	Default value	Description
mse_avtp_tx_param	uint8_t	dst_mac[6]	0x91, 0xe0, 0xf0, 0x00, 0x0e, 0x80	Destination MAC address The range of the value is from {0x00, 0x00, 0x00, 0x00, 0x00, 0x00} to {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}.
	uint8_t	src_mac[6]	0x76, 0x90, 0x50, 0x00, 0x00, 0x00	Source MAC address The range of the value is from {0x00, 0x00, 0x00, 0x00, 0x00, 0x00} to {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}.
	uint16_t	vlan	2	VLAN-ID The range of the value is from 0 to 4094.
	uint16_t	priority	3	VLAN priority The range of the value is from 0 to 7.
	uint32_t	uniqueid	1	Unique ID for AVTP packet The range of the value is from 0 to 65535.

4.2.32 struct mse_avtp_rx_param

Table 4.14 struct mse_avtp_rx_param

Structure name	Member			
	Type	Name	Default value	Description
mse_avtp_rx_param	uint8_t	streamid[8]	0x76, 0x90, 0x50, 0x00, 0x00, 0x00, 0x00, 0x01	Stream ID as defined AVTP The range of the value is from {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00} to {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}

4.2.33 struct mse_media_audio_config

Table 4.15 struct mse_media_audio_config

Structure name	Member			
	Type	Name	Default value	Description
mse_media_audio_config	uint32_t	samples_per_frame	0	Samples per frame The range of the value is 0 or from 1 to 738.
	enum MSE_CRF_TYPE	crf_type	MSE_CRF_TYPE_NOT_USE	CRF type MSE_CRF_TYPE_NOT_USE: not use CRF, but use AVTP MSE_CRF_TYPE_RX: recv CRF MSE_CRF_TYPE_TX: send CRF (see Table 4.5)

4.2.34 struct mse_media_video_config

Table 4.16 struct mse_media_video_config

Structure name	Member			
	Type	Name	Default value	Description
mse_media_video_config	uint32_t	bytes_per_frame	0	Bytes per frame The range of the value is 0 or from 128 to 1476.
	uint32_t	fps_denominator	0	Seconds of FPS for framerate The range of the value is from 0 to 1001.
	uint32_t	fps_numerator	0	Frames of FPS for framerate The range of the value is from 0 to 60000.
	uint32_t	bitrate	50000000	Bitrate (more than 0)
	uint32_t	class_interval_frames	4000	Frames per second The range of value is 100 to 8000
	uint32_t	max_interval_frames	1	Number of frames per interval

4.2.35 struct mse_media_mpeg2ts_config

Table 4.17 struct mse_media_mpeg2ts_config

Structure name	Member			
	Type	Name	Default value	Description
mse_media_mpeg2ts_config	uint32_t	tspackets_per_frame	7	TS packets per frame The range of the value is from 1 to 7.
	uint32_t	bitrate	50000000	Bitrate (more than 0)
	uint32_t	pcr_pid	8192	PID of PCR packet The range of the value is from 0 to 8192.
	enum MSE_TRANSMIT_MODE	transmit_mode	MSE_TRANSMIT_MODE_TIMESTAMP	MPEG2-TS transmit mode. MSE_TRANSMIT_MODE_BITRATE: bitrate. MSE_TRANSMIT_MODE_PCR: PCR value. MSE_TRANSMIT_MODE_TIMESTAMP: M2TS timestamp value.
	uint32_t	class_interval_frames	4000	Frames per second The range of value is 100 to 8000
	uint32_t	max_interval_frames	1	Number of frames per interval

4.2.36 struct mse_ptp_config

Table 4.18 struct mse_ptp_config

Structure name	Member			
	Type	Name	Default value	Description
mse_ptp_config	enum MSE_PTP_TYPE	type	MSE_PTP_TYPE_CURRENT_TIME	gPTP type MSE_PTP_TYPE_CURRENT_TIME: ptp current time MSE_PTP_TYPE_CAPTURE: ptp capture time
	uint32_t	deviceid	0	gPTP device number The range of the value is from 0 to 4294967295.
	uint32_t	capture_ch	2	gPTP capture channel The range of the value is from 0 to 16.
	uint32_t	capture_freq	300	gPTP capture timestamps per second The range of the value is from 0 to 4294967295.
	enum MSE_RECOVERY_CAPTURE_FREQ	recovery_capture_freq	MSE_RECOVERY_CAPTURE_FREQ_FIXED	Recovery capture frequency type (ex. using SRC, Audio device) MSE_RECOVERY_CAPTURE_FREQ_FIXED: fixed MSE_RECOVERY_CAPTURE_FREQ_NOT_FIXED: not fixed

4.2.37 struct mse_mch_config

Table 4.19 struct mse_mch_config

Structure name	Member			
	Type	Name	Default value	Description
mse_mch_config	bool	enable	false	Use the media clock recovery is true. (see Table 4.5)

4.2.38 struct mse_delay_time

Table 4.20 struct mse_delay_time

Structure name	Member			
	Type	Name	Default value	Description
mse_delay_time	uint32_t	max_transit_time_ns	2000000	Max transit time. The range of the value is from 0 to 4294967295.
	uint32_t	tx_delay_time_ns	2000000 (Audio), 0 (Video/MPEG2-TS)	Process time for transmission The range of the value is from 0 to 1073741824.
	uint32_t	rx_delay_time_ns	2000000 (Audio), 0 (Video/MPEG2-TS)	Process time for reception The range of the value is from 0 to 1073741824.

4.3 Adapter Interface

When a media adapter and network adapter uses MSE, Using an external function. MSE supported external interface are shown below.

Table 4.21 List of media adapter interface for MSE

Chapter	Function name	Description	Remarks
4.3.1	mse_register_adapter_media	Register media adapter.	-
4.3.2	mse_unregister_adapter_media	Unregister media adapter.	-
4.3.3	mse_get_audio_config	Get the current audio configurations	-
4.3.4	mse_set_audio_config	Set the audio configurations	-
4.3.5	mse_get_video_config	Get the current video configurations	-
4.3.6	mse_set_video_config	Set the video configurations	-
4.3.7	mse_get_mpeg2ts_config	Get the current MPEG2-TS configurations	-
4.3.8	mse_set_mpeg2ts_config	Set the MPEG2-TS configurations	-
4.3.9	mse_open	Open the MSE	-
4.3.10	mse_close	Close the MSE	-
4.3.11	mse_start_streaming	Start streaming	-
4.3.12	mse_stop_streaming	Stop streaming	-
4.3.13	mse_start_transmission	Start transmission	-

Table 4.22 List of network adapter interface for MSE

Chapter	Function name	Description	Remarks
4.3.14	mse_register_adapter_network	Register network adapter.	-
4.3.15	mse_unregister_adapter_network	Unregister network adapter.	-

Table 4.23 List of MCH interface for MSE

Chapter	Function name	Description	Remarks
4.3.16	mse_register_mch	Register MCH	-
4.3.17	mse_unregister_mch	Unregister MCH	-
4.3.18	mse_register_ptp	Register external PTP	-
4.3.19	mse_unregister_ptp	Unregister external PTP	-

Table 4.24 List of adapter interface structure for MSE

Chapter	Structure name	Description	Remarks
4.3.20	struct mse_audio_config	Audio configuration	-
4.3.21	struct mse_video_config	Video configuration	-
4.3.22	struct mse_mpeg2ts_config	MPEG2-TS configuration	-
4.3.23	struct mse_packet	DMA buffer information	-
4.3.24	struct mse_adapter_network_ops	Operations for network adapter	-
4.3.25	struct mch_ops	Operations for MCH	-
4.3.26	struct mse_ptp_ops	Operations for external PTP	

4.3.1 mse_register_adapter_media

[Function] int mse_register_adapter_media(enum MSE_TYPE type, char *name, char *device_name);

[Arguments]	type	Type of adapter
	name	Name of adapter
	device_name	Device name
[Returns]	>=0	Success, MSE adapter ID
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	Cannot create config device
	-EBUSY	Device busy

[Description] Register media adapter to MSE.

4.3.2 mse_unregister_adapter_media

[Function] int mse_unregister_adapter_media(int index_media);

[Arguments]	index_media	MSE adapter ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	Device is using

[Description] Unregister media adapter to MSE.

4.3.3 mse_get_audio_config

[Function]	int mse_get_audio_config(int index, struct mse_audio_config *config);	
[Arguments]	index	MSE instance ID
	config	Audio configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	This device isn't audio type.
[Description]	Get audio configuration.	

4.3.4 mse_set_audio_config

[Function]	int mse_set_audio_config(int index, struct mse_audio_config *config)	
[Arguments]	index	MSE instance ID
	config	Audio configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
		This device isn't opened.
	-EPERM	This device isn't audio type.
		Calculation bandwitdth overflow
	-EBUSY	This device is already start streaming.
		Hardware update disabled state.
	-ENOSPC	Insufficient bandwidth.
[Description]	Set audio configuration, but get it first.	

4.3.5 mse_get_video_config

[Function]	int mse_get_video_config(int index, struct mse_video_config *config);	
[Arguments]	index	MSE instance ID
	config	Video configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	This device isn't video type.
[Description]	Get video configuration.	

4.3.6 mse_set_video_config

[Function]	int mse_set_video_config(int index, struct mse_video_config *config)	
[Arguments]	index	MSE instance ID
	config	Video configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument This device isn't opened.
	-EPERM	This device isn't video type. Calculation bandwitdth overflow
	-EBUSY	This device is already start streaming. Hardware update disabled state.
	-ENOSPC	Insufficient bandwidth.
[Description]	Set video configuration, but get it first.	

4.3.7 mse_get_mpeg2ts_config

[Function]	int mse_get_mpeg2ts_config(int index, struct mse_mpeg2ts_config *config);	
[Arguments]	index	MSE instance ID
	config	MPEG2-TS configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	This device isn't mpeg2ts type.
[Description]	Get MPEG2-TS configuration.	

4.3.8 mse_set_mpeg2ts_config

[Function]	int mse_set_mpeg2ts_config(int index, struct mse_mpeg2ts_config *config)	
[Arguments]	index	MSE instance ID
	config	MPEG2-TS configuration
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
		This device isn't opened.
	-EPERM	This device isn't mpeg2ts type.
		Calculation bandwitdth overflow
	-EBUSY	This device is already start streaming.
		Hardware update disabled state.
	-ENOSPC	Insufficient bandwidth.
[Description]	Set MPEG2-TS configuration, but get it first.	

4.3.9 mse_open

[Function]	int mse_open(int index_media, bool tx);	
[Arguments]	index	MSE adapter ID
	tx	Flag of transmission 1: transmission, 0: reception
[Returns]	>=0	Success, instance ID of MSE
	< 0	Error
[Error number]	-EINVAL	Invalid argument Undefined media adapter.
	-ENODEV	Network adapter module is not loaded. PTP clock device cannot open
	-EBUSY	Device busy
	-EPERM	Fail to access to config device Unknown network device.
	-ENOMEM	Cannot allocate memory
	-ENETDOWN	Cannot get link speed
	-EAGAIN	Cannot prepare receive.
[Description]	Open the MSE, and get the MSE instance ID.	

4.3.10 mse_close

[Function]	int mse_close(int index);	
[Arguments]	index	MSE instance ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device is busy
	-EPERM	Operation is not permitted.
[Description]	Close the MSE.	

4.3.11 mse_start_streaming

[Function]	int mse_start_streaming(int index);	
[Arguments]	index	MSE instance ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device is busy
	-EPERM	Operation is not permitted.
[Description]	Start MSE streaming.	

4.3.12 mse_stop_streaming

[Function]	int mse_stop_streaming(int index);	
[Arguments]	index	MSE instance ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	Operation is not permitted.
[Description]	Stop MSE streaming.	

4.3.13 mse_start_transmission

[Function]	int mse_start_transmission(int index, void *buffer, size_t buffer_size, void *priv, int (*mse_completion)(void *priv, int size));	
[Arguments]	index	MSE instance ID
	buffer	Send/Receive buffer
	buffer_size	Buffer size
	priv	Private data
	mse_completion	Callback function pointer
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device is busy
	-EPERM	Operation is not permitted.
	-EAGAIN	No acceptable buffer.

[Description] MSE start transmission.

The buffer_size should be set the unit of the data period that handled by the media format as follows.

Type of device	buffer_size
Audio	Size of audio samples of buffer period.
Video	Size of the one video frame.
MPEG2-TS	Size of MPEG2-TS packets of buffer period.

4.3.14 mse_register_adapter_network

[Function]	int mse_register_adapter_network(struct mse_adapter_network_ops *ops);	
[Arguments]	ops	Adapter operations
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device busy
[Description]	Register network adapter to MSE.	

4.3.15 mse_unregister_adapter_network

[Function]	int mse_unregister_adapter_network(int index);	
[Arguments]	index_media	MSE instance ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EPERM	Device is using
[Description]	Unregister network adapter to MSE.	

4.3.16 mse_register_mch

[Function]	int mse_register_mch(struct mch_ops *ops);	
[Arguments]	ops	MCH operations
[Returns]	>=0	Success, MCH table ID
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device busy
[Description]	Register MCH to MSE.	

4.3.17 mse_unregister_mch

[Function]	int mse_unregister_mch(int index)	
[Arguments]	index	MCH table ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
[Description]	Unregister MCH to MSE.	

4.3.18 mse_register_ptp

[Function]	int mse_register_ptp(struct mse_ptp_ops *ops);	
[Arguments]	ops	PTP operations
[Returns]	>=0	Success, PTP table ID
	< 0	Error
[Error number]	-EINVAL	Invalid argument
	-EBUSY	Device busy
[Description]	Register external PTP to MSE.	

4.3.19 mse_unregister_ptp

[Function]	int mse_unregister_ptp(int index)	
[Arguments]	index	PTP table ID
[Returns]	0	Success
	< 0	Error
[Error number]	-EINVAL	Invalid argument
[Description]	Unregister external PTP to MSE.	

4.3.20 struct mse_audio_config

Table 4.25 struct mse_audio_config

Structure name	Member			
	Type	Name	Default value	Description
mse_audio_config	int	sample_rate	0	Sampling rate The range of the value is from 8000 to 192000.
	int	channels	0	Channels The range of the value is from 1 to 24.
	int	period_size	0	Period size The range of the value is from 64 to 8192.
	int	bytes_per_sample	0	Samples per frame The range of the value is from 2 to 96.
	enum MSE_AUDIO_BIT	sample_bit_depth	MSE_AUDIO_BIT_INVALID	Sample bit depth MSE_AUDIO_BIT_16: 16bit MSE_AUDIO_BIT_18: 18bit MSE_AUDIO_BIT_20: 20bit MSE_AUDIO_BIT_24: 24bit MSE_AUDIO_BIT_32: 32bit
	bool	is_big_endian	0	Sample endian 1: big endian, 0: little endian
	int	samples_per_frame	0	Samples per frame is data number in 1 ether frame. The range of the value is 0 or from 1 to 738.

4.3.21 struct mse_video_config

Table 4.26 struct mse_video_config

Structure name	Member			
	Type	Name	Default value	Description
mse_video_config	enum MSE_VIDEO_FORMAT_TYPE	format	MSE_VIDEO_FORMAT_H264_BYTE_STREAM	Video format MSE_VIDEO_FORMAT_H264_BYTE_STREAM, MSE_VIDEO_FORMAT_H264_AVC, MSE_VIDEO_FORMAT_MJPEG
	int	bitrate	50000000	Bitrate (more than 0)
	struct	fps	-	Framerate
	int	numerator	0	Numerator The range of the value is from 0 to 60000.
	int	denominator	0	Denominator The range of the value is from 0 to 1001.
	int	bytes_per_frame	0	Bytes per frame is data size in 1 ether frame. The range of the value is 0 or from 128 to 1476.

4.3.22 struct mse_mpeg2ts_config

Table 4.27 struct mse_mpeg2ts_config

Structure name	Member			
	Type	Name	Default value	Description
mse_mpeg2ts_config	int	bitrate	50000000	Bitrate (more than 0)
	int	tspackets_per_frame	7	TS packets per frame is packet number in 1 ether frame. The range of the value is from 1 to 7.
	int	pcr_pid	8192	PID of PCR packet The range of the value is from 0 to 8192.
	enum MSE_MPEG2TS_TYPE	mpeg2ts_type	MSE_MPEG2TS_TYPE_TS	MPEG2-TS type MSE_MPEG2TS_TYPE_TS:TS format MSE_MPEG2TS_TYPE_M2TS:M2TS format
	enum MSE_TRANSMIT_MODE	transmit_mode	MSE_TRANSMIT_MODE_TIMESTAMP	MPEG2-TS transmit mode. MSE_TRANSMIT_MODE_BITRATE: bitrate. MSE_TRANSMIT_MODE_PCR: PCR value. MSE_TRANSMIT_MODE_TIMESTAMP: M2TS timestamp value.

4.3.23 struct mse_packet

Table 4.28 struct mse_packet

Structure name	Member			
	Type	Name	Default value	Description
mse_packet	unsigned int	len	1526	Packet size
	dma_addr_t	paddr	0	Physical address for DMA
	void *	vaddr	NULL	Virtual address for driver

4.3.24 struct mse_adapter_network_ops

Table 4.29 struct mse_adapter_network_ops

Structure name	Member		
	Type	Name	Description
mse_adapter_network_ops	char *	name	The network adapter name
	enum MSE_TYPE	type	The network adapter type
	struct module *	owner	The module owner information.
	int (*)(char *name)	open	The callback function pointer of "open" at the kernel driver interface.
	int (*)(int index)	release	The callback function pointer of "release" at the kernel driver interface.
	int (*)(int index, struct eavb_cbsparam *cbs)	set_cbs_param	The callback function pointer of "set_cbs_param" at the kernel driver interface.
	int (*)(int index, u8 streamid[8])	set_streamid	The callback function pointer of "set_streamid" at the kernel driver interface.
	int (*)(int index, struct mse_packet *packets, int num_packets)	send_prepare	The callback function pointer of "send_prepare" at the kernel driver interface.
	int (*)(int index, struct mse_packet *packets, int num_packets)	send	The callback function pointer of "send" at the kernel driver interface.
	int (*)(int index, struct mse_packet *packets, int num_packets)	receive_prepare	The callback function pointer of "receive_prepare" at the kernel driver interface.
	int (*)(int index, int num_packets)	receive	The callback function pointer of "receive" at the kernel driver interface.
	int (*)(int index)	cancel	The callback function pointer of "cancel" at the kernel driver interface.
	int (*)(int index)	get_link_speed	The callback function pointer of "get_link_speed" at the kernel driver interface.

4.3.25 struct mch_ops

Table 4.30 struct mch_ops

Structure name	Member		
	Type	Name	Description
mch_ops	struct module *	owner	The module owner information.
	void *(*)(void)	open	The callback function pointer of "open" at the kernel driver interface.
	int (*)(void *mch)	close	The callback function pointer of "close" at the kernel driver interface.
	int (*)(void *mch, u32 ns)	set_interval	The callback function pointer of "set_interval" at the kernel driver
	int (*)(void *mch, struct mch_timestamp *ts, int count);	send_timestamps	The callback function pointer of "send_timestamps" at the kernel driver interface.
	int (*)(void *mch, int *value)	get_recovery_value	The callback function pointer of "get_recovery_value" at the kernel driver interface.

4.3.26 struct mse_ptp_ops

Table 4.31 struct mse_ptp_ops

Structure name	Member		
	Type	Name	Description
mse_ptp_ops	struct module *	owner	The module owner information.
	int (*)(u64 *ns)	get_time	The callback function pointer of "get_time" at the kernel driver interface.
	void *(*)(void)	open	The callback function pointer of "open" at the kernel driver interface.
	int (*)(void *ptp_handle)	close	The callback function pointer of "close" at the kernel driver interface.
	int (*)(void *ptp_handle, int ch, int max_count)	capture_start	The callback function pointer of "capture_start" at the kernel driver interface.
	int (*)(void *ptp_handle)	capture_stop	The callback function pointer of "capture_stop" at the kernel driver interface.
	int (*)(void *ptp_handle, int req_count, u64 *timestamps)	get_timestamps	The callback function pointer of "get_timestamps" at the kernel driver interface.
	void *(*)(u32 (*handler)(void *), void *priv)	timer_open	The callback function pointer of "timer_open" at the kernel driver interface.
	int (*)(void *timer_handle)	timer_close	The callback function pointer of "timer_close" at the kernel driver interface.
	int (*)(void *timer_handle, u32 start)	timer_start	The callback function pointer of "timer_start" at the kernel driver interface.
	int (*)(void *timer_handle)	timer_cancel	The callback function pointer of "timer_cancel" at the kernel driver interface.

4.3.27 mse_err, mse_warn, mse_info, mse_debug

Table 4.32 mse_err, mse_warn, mse_info, mse_debug

Macro	Paramaters	Level	Description
mse_err(fmt, ...)	fmt: Format string like printk ...: Other paramaters	KERN_ERR	Send a message to the kernel trace buffer according to kernel log level
mse_warn(fmt, ...)		KERN_WARNING	
mse_info(fmt, ...)		KERN_INFO	
mse_debug(fmt, ...)		KERN_DEBUG	

4.3.28 mse_compare_param_key, mse_name_strlcpy

Table 4.33 mse_compare_param_key, mse_name_strlcpy

Macro	Parameters	Description
mse_compare_param_key(val, key)	val: compare value key: compare key	Compare parameter function for MSE
mse_name_strlcpy(dst, src)	det: copy destination src: copy source	Copy name function for MSE

5. Packet Format

5.1 Ethernet Frame Format

The format of an Ethernet frame is shown below. It is based on the Ethernet frame format with VLAN Tag of IEEE Std 802.1Q.

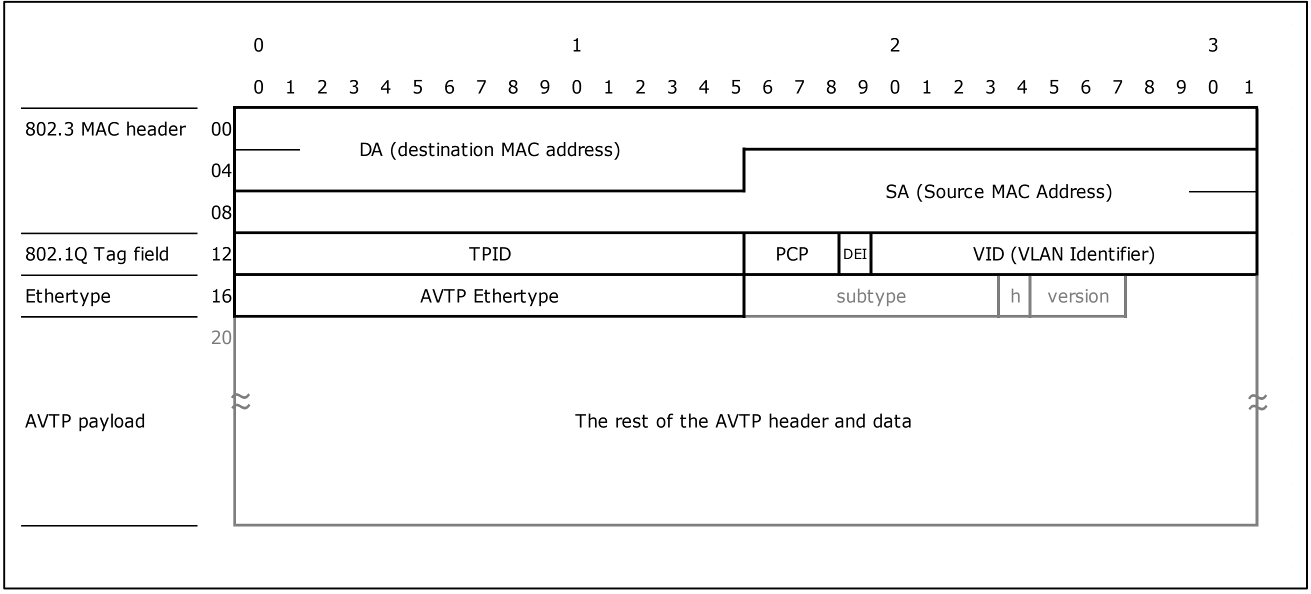


Figure 5-1 Ethernet header format

Table 5.1 Description of fields in Ethernet header

Standard	Field name	Description	Value	Configuration
IEEE 1722 C.2.1.1	DA (destination MAC address)	Destination Address	6 bytes	sysfs (dst_mac)
IEEE 1722 C.2.1.2	SA (Source MAC Address)	Source Address	6 bytes	sysfs (src_mac)
IEEE 1722 C.2.1.3	TPID	Tagged Protocol Identifier	0x8100	-
IEEE 1722 C.2.1.4	PCP	Priority Code Point	0-7	sysfs (priority)
IEEE 1722 C.2.1.5	DEI	Drop Eligible bit	0	-
IEEE 1722 C.2.1.6	VID (VLAN Identifier)	VLAN ID	1-4095	sysfs (vlan)
IEEE 1722 C.2.1.7	AVTP Ethertype	AVTP Ethertype	0x22F0	-

5.2 AVTP IEC61883-4 MPEG2-TS Format

The format of AVTP IEC61883-4 MPEG2-TS is shown below. It is basing on the IEEE Std 1722 Clause 6 IEC 61883/IIDC Format.

5.2.1 AVTP IEC61883-4 MPEG2-TS format

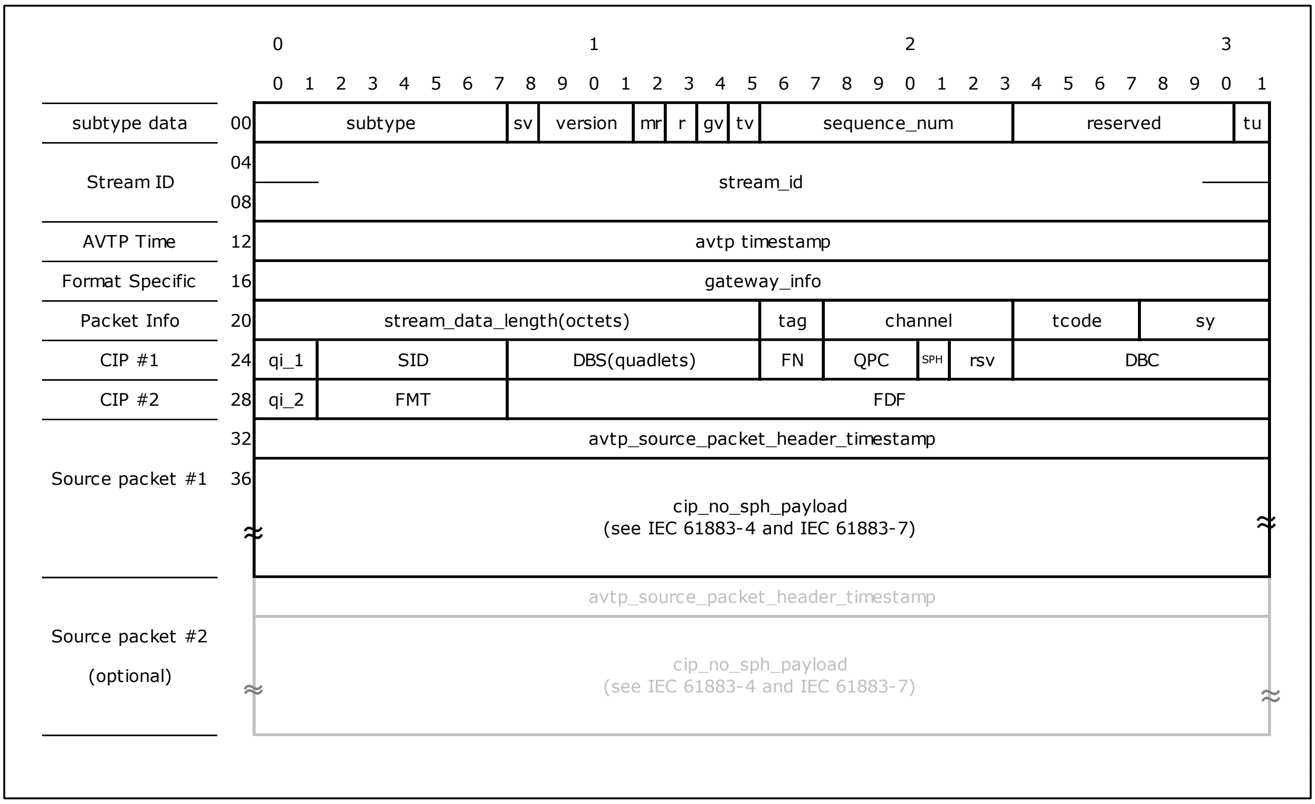

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
subtype data 00 subtype sv version mr r gv tv sequence_num reserved tu
Stream ID 04 08 stream_id
AVTP Time 12 avtp timestamp
Format Specific 16 gateway_info
Packet Info 20 stream_data_length(octets) tag channel tcode sy
CIP #1 24 qi_1 SID DBS(quadlets) FN QPC SPH rsv DBC
CIP #2 28 qi_2 FMT FDF
32 avtp_source_packet_header_timestamp
Source packet #1 36 cip_no_sph_payload (see IEC 61883-4 and IEC 61883-7)
avtp_source_packet_header_timestamp
Source packet #2 (optional) cip_no_sph_payload (see IEC 61883-4 and IEC 61883-7)

Figure 5-2 AVTP IEC61883-4 MPEG2-TS format

Table 5.2 Description of fields in AVTP IEC61883-4 MPEG2-TS format

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	0 (61883_ IIDC)	
IEEE 1722 5.4.4.1	sv	stream_id valid field	1	-
IEEE 1722 5.4.2.3	version	version field	0	-
IEEE 1722 5.4.3.2	mr	media clock restart field	0	-
-	r	reserved	0	-
IEEE 1722 6.2.1	gv	gateway_info valid field	0	-
IEEE 1722 6.4.3.1	tv	avtp_timestamp valid field	1	-
IEEE 1722 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	reserved	0	-
IEEE 1722 5.4.3.6	tu	timestamp uncertain field	0	-
IEEE 1722	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac),

5.4.3.7				Last 2 bytes: sysfs (unique_id)
IEEE 1722 6.4.3.2	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
IEEE 1722 6.2.2	gateway_info	gateway_info field	0	-
IEEE 1722 5.4.3.9	stream_data_length (octets)	stream data length field	0-1500	Add by MSE
IEEE 1722 6.2.3	tag	tag field	1	-
IEEE 1722 6.2.4	channel	channel field	0x1F	-
IEEE 1722 6.2.5	tcode	type code field	0xA	-
IEEE 1722 6.2.6	sy	sy field(application specific)	0	-
IEEE 1722 6.4.2.1	qi_1	quadlet indicator field	0b00	-
IEEE 1722 6.4.2.2	SID	source identifier field	0x3F	-
IEC61883-4 5.1	DBS (quadlets)	data block size	6	-
IEC61883-4 5.1	FN	fraction number field	3	-
IEC61883-4 5.1	QPC	quadlet padding count field	0	-
IEC61883-4 5.1	SPH	source packet header field	1	-
-	rsv	reserved	0	-
IEC61883-4 5.2	DBC	data block count field (Increment by the number of source packet)	-	Add by MSE
IEEE 1722 6.4.2.8	qi_2	quadlet indicator field	0b10	-
IEC61883-4 5.1	FMT	stream format field	0x20 (MPEG 2-TS)	-
IEC61883-4 5.3	FDF	format dependent field	0 (not time-shifted)	
IEEE 1722 6.4.4.2	avtp_source_packet_header_timestamp	AVTP Timesamp for each source packet	-	Add by MSE

5.3 AVTP IEC61883-6 AM824 Format

The format of AVTP IEC61883-6 AM824 is shown below. It is basing on the IEEE Std 1722 Clause 6 IEC 61883/IIDC Format.

5.3.1 AVTP IEC61883-6 Audio format

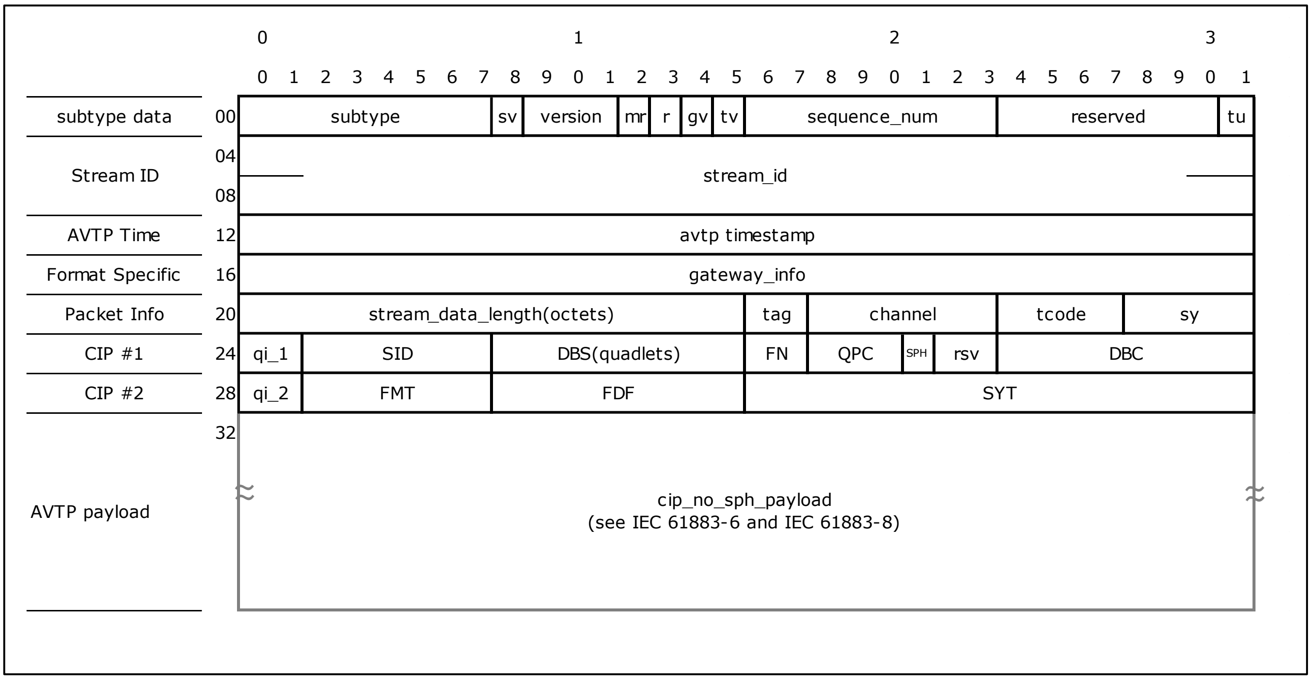


Figure 5-3 AVTP IEC61883-6 Audio format

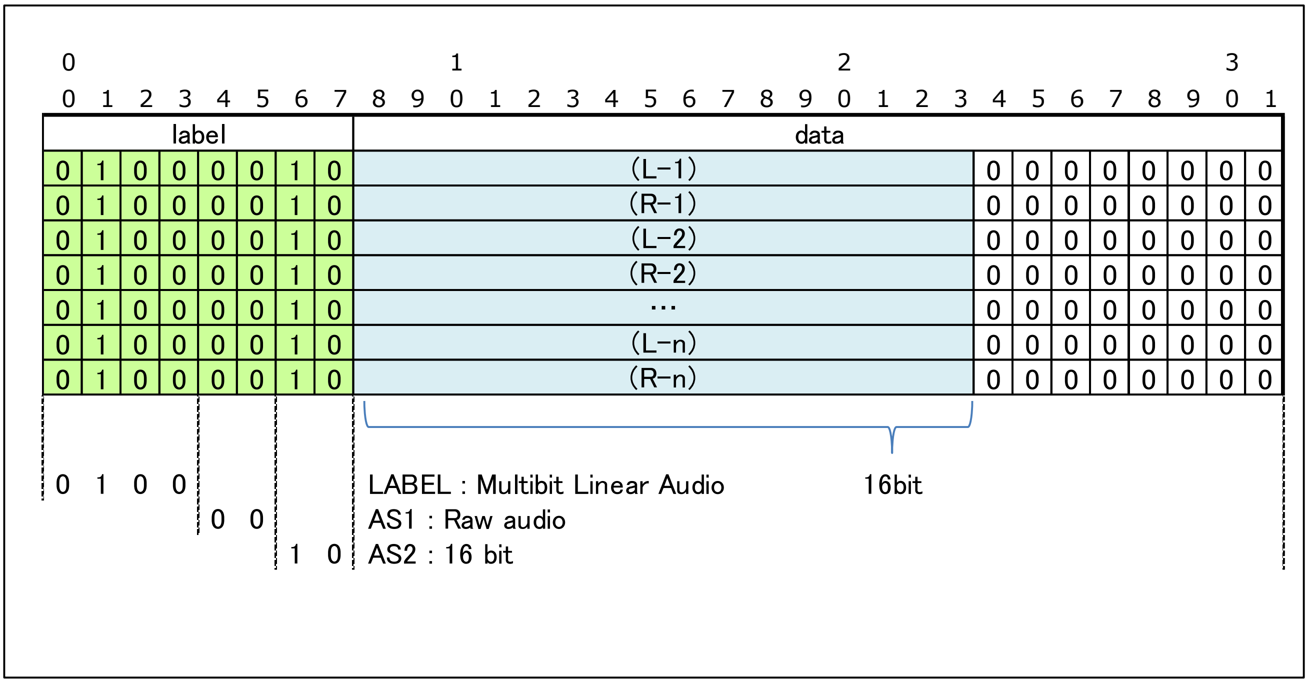
Table 5.3 Description of fields in AVTP IEC61883-6 Audio format

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	0 (61883_ IIDC)	
IEEE 1722 5.4.4.1	sv	stream_id valid field	1	-
IEEE 1722 5.4.2.3	version	version field	0	-
IEEE 1722 5.4.3.2	mr	media clock restart field	0	-
-	r	reserved	0	-
IEEE 1722 6.2.1	gv	gateway_info valid field	0	-
IEEE 1722 6.4.3.1	tv	avtp_timestamp valid field	1	-
IEEE 1722 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	reserved	0	-
IEEE 1722 5.4.3.6	tu	timestamp uncertain field	0	-
IEEE 1722 5.4.3.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
IEEE 1722 6.4.3.2	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
IEEE 1722 6.2.2	gateway_info	gateway_info field	0	-

IEEE 1722 5.4.3.9	stream_data_length (octets)	stream data length field	0-1500	Add by MSE
IEEE 1722 6.2.3	tag	tag field	1	-
IEEE 1722 6.2.4	channel	channel field	0x1F	-
IEEE 1722 6.2.5	tcode	type code field	0xA	-
IEEE 1722 6.2.6	sy	sy field(application specific)	0	-
IEEE 1722 6.4.2.1	qi_1	quadlet indicator field	0b00	-
IEEE 1722 6.4.2.2	SID	source identifier field	0x3F	-
IEEE 1722 6.4.2.3	DBS (quadlets)	data block size (Number of channels)	-	ALSA Setting
IEEE 1722 6.4.2.4	FN	fraction number field	0	-
IEEE 1722 6.4.2.5	QPC	quadlet padding count field	0	-
IEEE 1722 6.4.2.6	SPH	source packet header field	0	-
-	rsv	reserved	0	-
IEEE 1722 6.4.2.7	DBC	data block count field (Increment by the number of samples)	-	ALSA Setting
IEEE 1722 6.4.2.8	qi_2	quadlet indicator field	0b10	-
IEEE 1722 6.4.2.9	FMT	stream format field	0x10 (Audio and music)	-
IEEE 1722 6.4.3.3	FDF	format dependent field (Sampling rate)	-	ALSA Setting
IEEE 1722 6.4.3.4	SYT	synchronization timing field	0xFFFF	-

5.3.2 Payload format of the AVTP IEC61883-6 AM824 format

The format of payload is shown below. A packet format is AM824 format of IEC61883-6. Audio data is stored in MBLA 16bit.



label								data								
0	1	0	0	0	0	1	0	(L-1)	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	(R-1)	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	(L-2)	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	(R-2)	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	…	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	(L-n)	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	(R-n)	0	0	0	0	0	0	0	0

Figure 5-4 AM824 MBLA 16bit format

5.4 PCM AVTP Audio Format (AAF PCM)

The format of PCM AVTP Audio is shown below. It is basing on the IEEE Std 1722 Clause 8 AVTP Audio Format.

5.4.1 AVTP Audio Format (AAF)

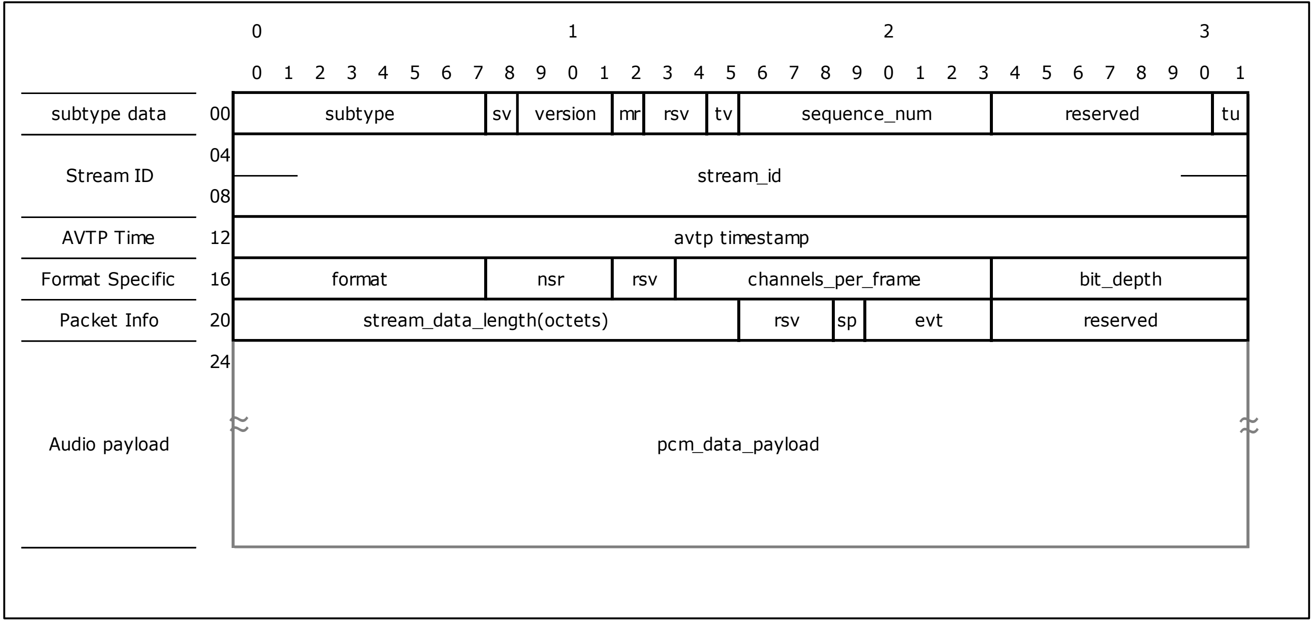


Figure 5-5 AVTP Audio Format (AAF)

Table 5.4 Description of fields in AVTP Audio Format (AAF)

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	2 (AAF)	-
IEEE 1722 5.4.4.1	sv	stream_id valid field	1	-
IEEE 1722 5.4.2.3	version	version field	0	-
IEEE 1722 5.4.3.2	mr	media clock restart field	0	-
-	rsv	reserved	0	
IEEE 1722 5.4.3.4	tv	avtp_timestamp valid	1	-
IEEE 1722 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	reserved	0	-
IEEE 1722 5.4.3.6	tu	timestamp uncertain field	0	-
IEEE 1722 5.4.3.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
IEEE 1722 5.4.3.8	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
IEEE 1722 8.2.2	format	format field	2-4	ALSA Setting
IEEE 1722 8.3.1	nsr	nominal sample rate (Sampling rate)	-	ALSA Setting
-	rsv	reserved	0	-
IEEE 1722 8.3.2	channels_par_frame	number of channels field	-	ALSA Setting
IEEE 1722	bit_depth	bit_depth field	16, 24, 32	ALSA Setting

8.3.3				
IEEE 1722 5.4.3.9	stream_data_length(octets)	stream data length field	0-1500	Add by MSE
-	rsv	reserved	0	-
IEEE 1722 8.2.3	sp	sparse timestamp field	0	-
IEEE 1722 8.2.4	evt	evt field	0	-
-	reserved	reserved	0	-

5.4.2 Payload format of the AVTP audio format

The format of payload is shown below. A packet format is AAF. Data is stored in INT_16.

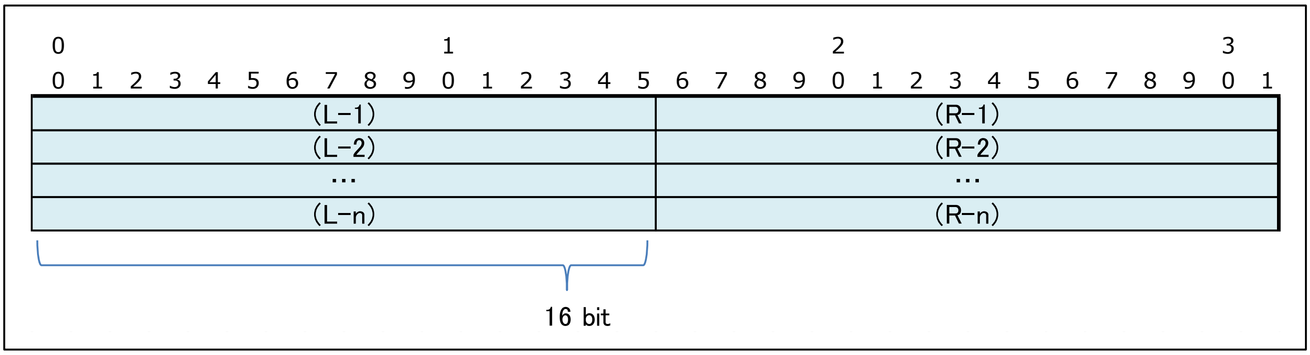


Figure 5-6 AAF Payload 16-bit integer format

5.5 H.264 Compressed Video Format without h264_timestamp field (CVF H264 D13)

The format of H.264 Compressed Video Format without h264_timestamp field is shown below. It is basing of the IEEE P1722/D13 Clause 9 Compressed Video Format.

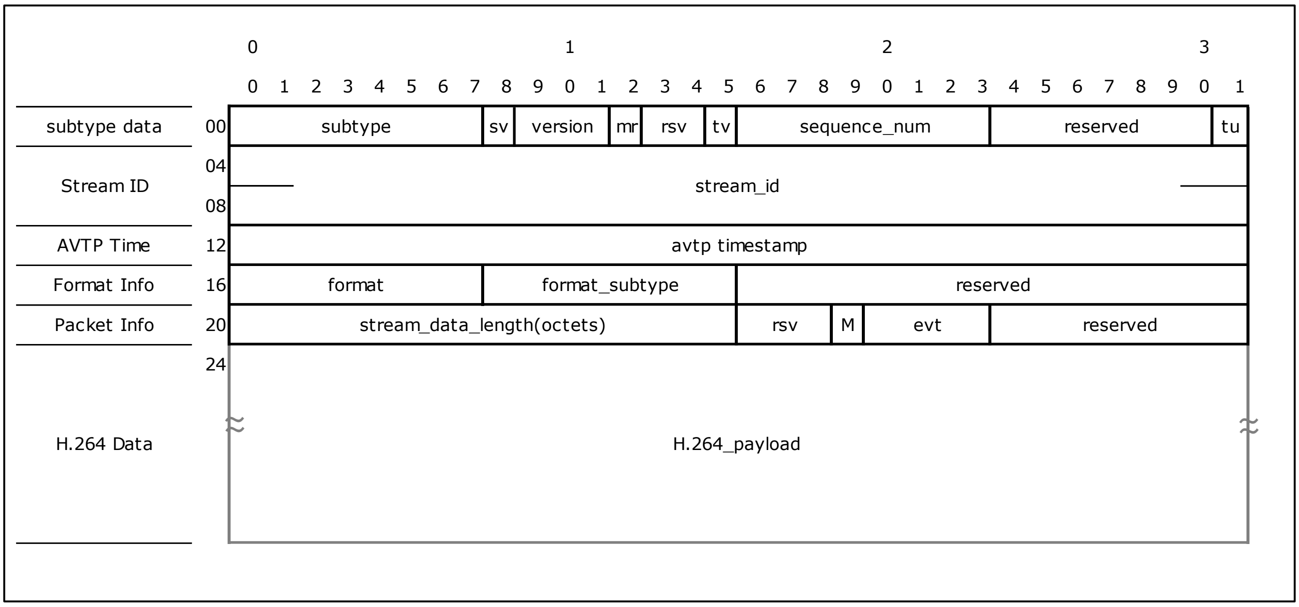


Figure 5-7 H.264 Compressed Video Format without h264_timestamp field

Table 5.5 Description of fields in H.264 Compressed Video Format without h264_timestamp field

Standard	Field name	Description	Value	Configuration
P1722/D13 5.4.2.1	subtype	subtype field	3 (CVF)	-
P1722/D13 5.4.4.1	sv	stream_id Valid field	1	-
P1722/D13 5.4.2.3	version	version field	0	-
P1722/D13 5.4.3.2	mr	media clock restart field	0	-
-	rsv	reserved	0	
P1722/D13 5.4.3.4	tv	avtp_timestamp valid field	1	-
P1722/D13 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	-	0	-
P1722/D13 5.4.3.6	tu	timestamp uncertain field	0	-
P1722/D13 5.4.3.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
P1722/D13 9.5.1.1	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
P1722/D13 9.3.1	format	format field	2 (RFC)	-
P1722/D13 9.3.2	format_subtype	format_subtype field	1 (H.264)	-
-	reserved	reserved	0	-
P1722/D13 5.4.3.9	stream_data_length(octets)	stream data length field	0-1500	Add by MSE
-	rsv	reserved	0	-
P1722/D13 9.5.1.2	M	M field	0 or 1	Set by MSE for the very last packet of the access unit.

P1722/D13 9.3.4	evt	evt field	0	-
-	reserved	reserved	0	-

5.6 H.264 Compressed Video Format (CVF H264)

The format of H.264 Compressed Video Format is shown below. It is basing of the IEEE Std 1722 Clause 9 Compressed Video Format.

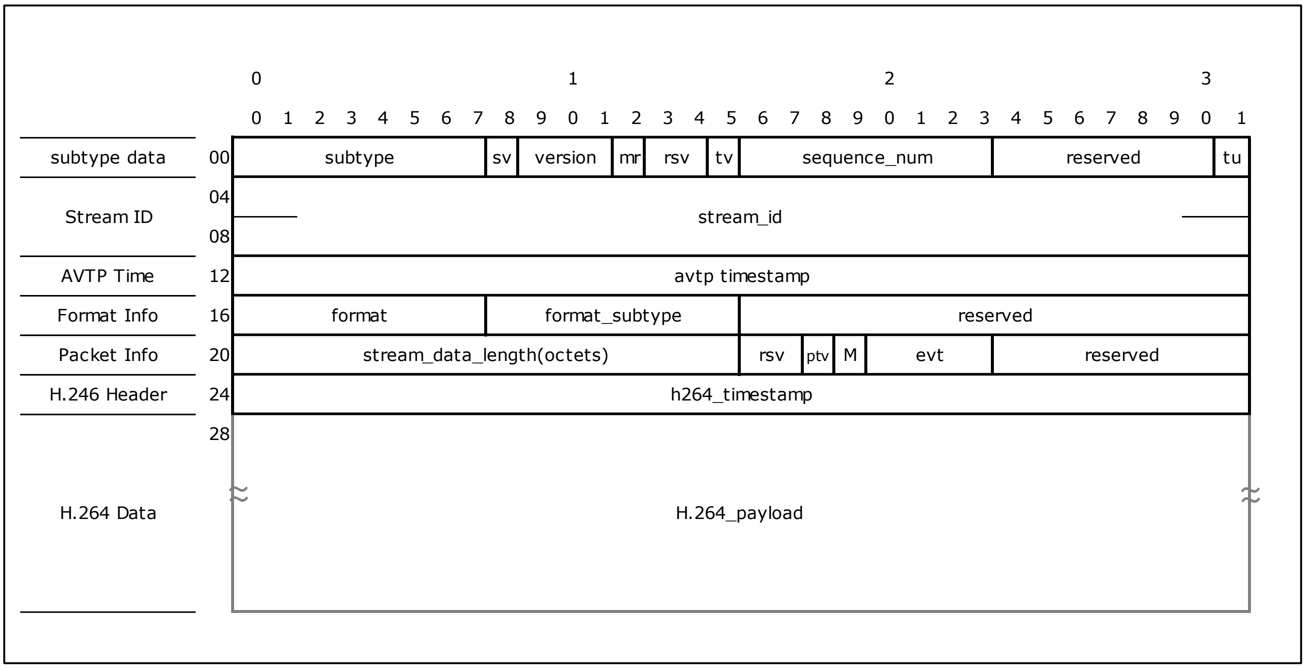


Figure 5-8 H.264 Compressed Video Format

Table 5.6 Description of fields in H.264 Compressed Video Format

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	3(CVF)	-
IEEE 1722 5.4.3.1	sv	stream_id Valid field	1	-
IEEE 1722 5.4.2.3	version	version field	0	-
IEEE 1722 5.4.3.2	mr	media clock restart field	0	-
-	rsv	reserved	0	
IEEE 1722 5.4.3.4	tv	avtp_timestamp valid field	1	-
IEEE 1722 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	-	0	-
IEEE 1722 5.4.3.6	tu	timestamp uncertain field	0	-
IEEE 1722 5.4.3.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
IEEE 1722 9.5.1.1	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
IEEE 1722 9.3.1	format	format field	2 (RFC)	-
IEEE 1722 9.3.2	format_subtype	format_subtype field	1 (H.264)	-
-	reserved	reserved	0	-
IEEE 1722 5.4.3.9	stream_data_length (octets)	stream data length field	0-1500	Add by MSE
-	rsv	reserved	0	-

IEEE 1722 9.5.1.2	ptv	ptv field	0	-
IEEE 1722 9.5.1.3	M	M field	0 or 1	Set by MSE for the very last packet of the access unit.
IEEE 1722 9.3.4	evt	evt field	0	-
-	reserved	reserved	0	-
IEEE 1722 9.5.2.1	h264_timestamp	h264_timestamp field	0	-

5.7 MJPEG Compressed Video Format (CVF MJPEG)

The format of MJPEG Compressed Video Format is shown below. It is basing of the IEEE Std 1722 Clause 9 Compressed Video Format.

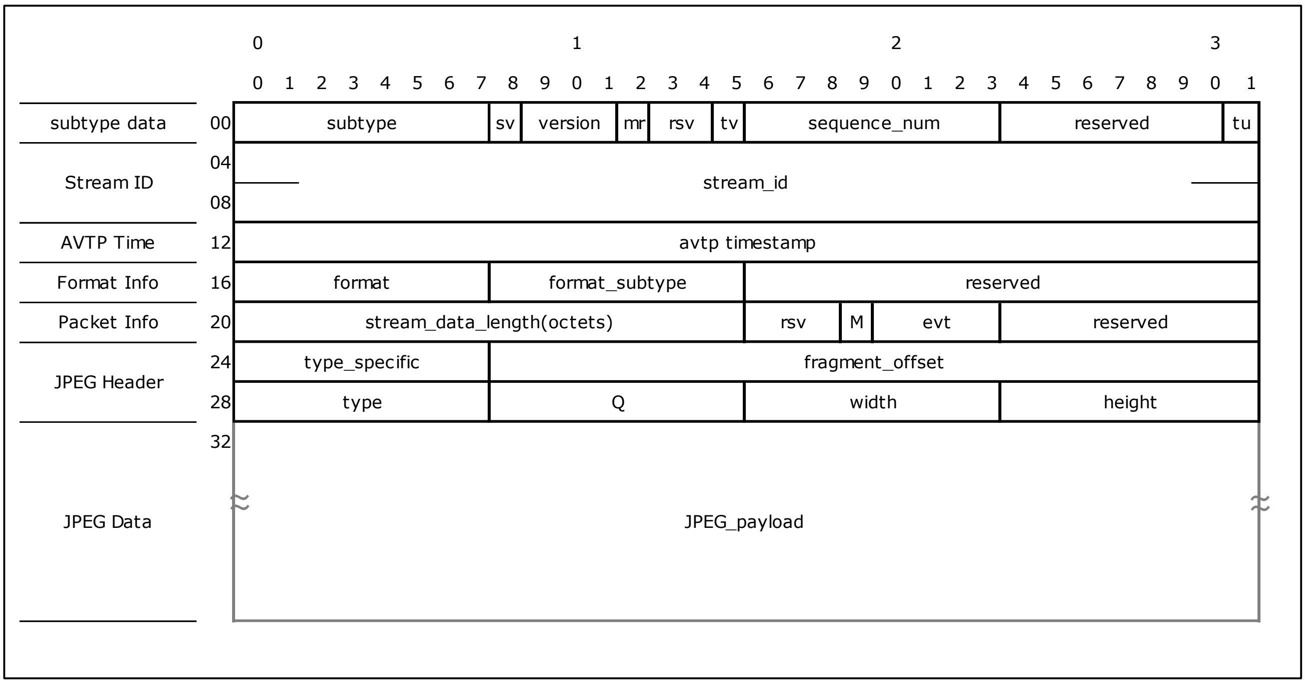


Figure 5-9 MJPEG Compressed Video Format

Table 5.7 Description of fields in MJPEG Compressed Video Format

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	3(CVF)	-
IEEE 1722 5.4.3.1	sv	stream_id Valid field	1	-
IEEE 1722 5.4.2.3	version	version field	0	-
IEEE 1722 5.4.3.2	mr	media clock restart field	0	-
-	rsv	reserved	0	
IEEE 1722 5.4.3.4	tv	avtp_timestamp valid field	1	-
IEEE 1722 5.4.3.5	sequence_num	sequence number field	0-255	Add by MSE
-	reserved	-	0	-
IEEE 1722 5.4.3.6	tu	timestamp uncertain field	0	-
IEEE 1722 5.4.3.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
IEEE 1722 9.4.1.1	avtp timestamp	AVTP Timestamp	4 bytes	Add by MSE
IEEE 1722 9.3.1	format	format field	2 (RFC)	-
IEEE 1722 9.3.2	format_subtype	format_subtype field	0 (MJPEG)	-
-	reserved	reserved	0	-
IEEE 1722	stream_data_length	stream data length field	0-1500	Add by MSE

5.4.3.9	(octets)			
-	rsv	reserved	0	-
IEEE 1722 9.4.1.2	M	M field	0 or 1	Set by MSE for the very last packet of the access unit.
IEEE 1722 9.3.4	evt	evt field	0	-
-	reserved	reserved	0	-
IEEE 1722 9.4.2.1	type_specific	type_specific field	0	-
IEEE 1722 9.4.2.2	fragment_offset	fragment_offset field	0 - 2^24	-
IEEE 1722 9.4.2.3	type	type field	0, 1, 64, 65	-
IEEE 1722 9.4.2.4	Q	Q field	0 - 99, 255	-
IEEE 1722 9.4.2.5	width	width field	1-255	-
IEEE 1722 9.4.2.6	height	height field	1-255	-

5.8 Clock Reference Format (CRF)

The format of Clock Reference Format is shown below. It is basing of the IEEE Std 1722 Clause 11 Clock Reference Format.

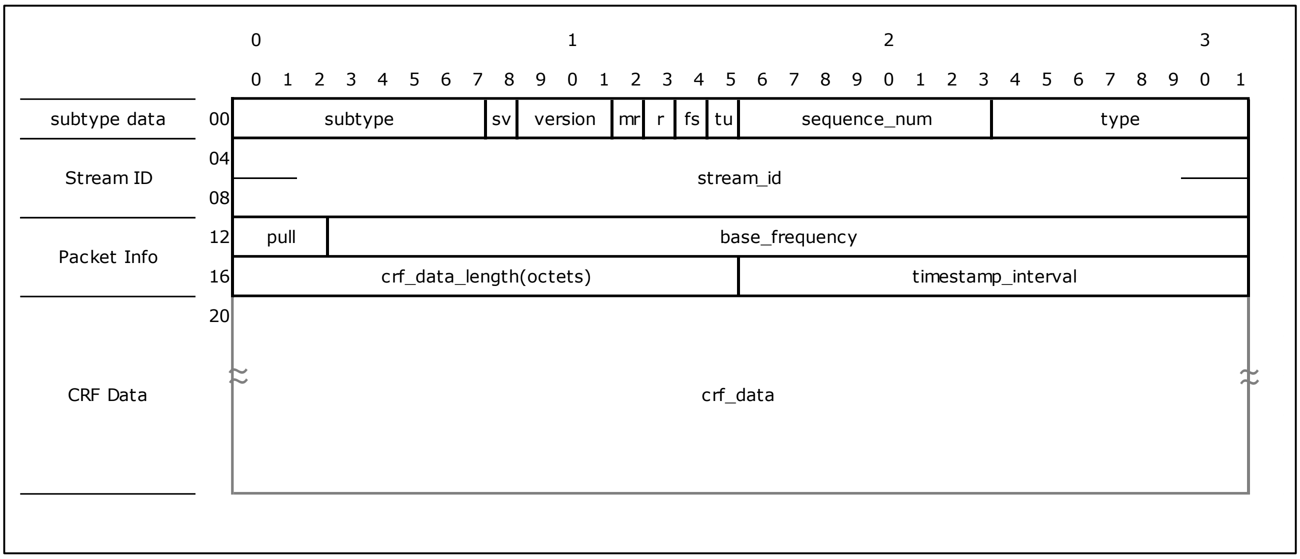


Figure 5-10 Clock Reference Format

Table 5.8 Description of fields in Clock Reference Format

Standard	Field name	Description	Value	Configuration
IEEE 1722 5.4.2.1	subtype	subtype field	4 (CRF)	-
IEEE 1722 11.2.1	sv	stream_id Valid field	1	-
IEEE 1722 5.4.2	version	version field	0	-
IEEE 1722 11.2.2	mr	media clock restart field	0	-
-	r	reserved	0	
IEEE 1722 11.2.3	fs	frame sync	0	
IEEE 1722 11.2.4	tu	timing uncertain field	0	-
IEEE 1722 11.2.5	sequence_num	sequence number field	0-255	Add by MSE
IEEE 1722 11.2.6	type	type field	1 (Audio sample)	-
IEEE 1722 11.2.7	stream_id	stream id field	8 bytes	First 6 bytes: sysfs (src_mac), Last 2 bytes: sysfs (unique_id)
IEEE 1722 11.2.8	pull	pull field	0	
IEEE 1722 11.2.9	base_frequency	base frequency field	29bit	ALSA Setting
IEEE 1722 11.2.10	crf_data_length(octets)	length (in octets) of the crf_data	0-1500	Add by MSE
IEEE 1722 11.2.11	timestamp_interval	timestamp interval field	1-65535	Add by MSE

6. Integration

6.1 Directory Configuration

Directory structure of this module is shown below

```
|---avb-mse
        Kconfig                              : Kernel configuration file
        Makefile                             : Make file
        avtp.c                               : common AVTP source file
        avtp.h                               : common AVTP header file
        jpeg.c                               : common JPEG source file
        jpeg.h                               : common JPEG header file
        ravb_mse_kernel.h                    : MSE public header file for kernel land
        ravb_mse.h                           : MSE public header file for user land
        mse_core_main.c                      : MSE core source file
        mse_packet_ctrl.c                    : MSE core packet control source file
        mse_packet_ctrl.h                    : MSE core packet control header file
        mse_config.c                         : MSE configuration source file
        mse_config.h                         : MSE configuration header file
        mse_sysfs.c                          : MSE sysfs source file
        mse_sysfs.h                          : MSE sysfs header file
        mse_ioctl.c                          : MSE ioctl source file
        mse_ioctl_local.h                    : MSE ioctl header file
        mse_ptp_dummy.c                      : MSE PTP dummy source file
        mse_ptp.h                            : MSE PTP header file
        mse_adapter_alsa.c                   : ALSA adapter source file
        mse_adapter_eavb.c                   : EAVB adapter source file
        mse_adapter_v4l2.c                   : V4L2 adapter source file
        mse_adapter_mch.c                    : MCH adapter source file
        mse_packetizer.h                     : Packetizer header file
        mse_packetizer.c                     : Packetizer common source file
        mse_packetizer_aaf.c                 : AVTP AAF Packetizer source file
        mse_packetizer_iec61883_4.c          : AVTP IEC61883-4 Packetizer source file
        mse_packetizer_iec61883_6.c          : AVTP IEC61883-6 Packetizer source file
        mse_packetizer_cvf_h264.c            : AVTP CVF H.264 Packetizer source file
        mse_packetizer_cvf_mjpeg.c           : AVTP CVF MJPEG Packetizer source file
        mse_packetizer_crf.c                 : AVTP CRF Packetizer source file
```

Figure 6-1 Directory structure

6.2 Integration Procedure

This section describes a process of build and integration.

6.2.1 Build kernel modules only

Following operation is to build loadable modules.

step1. Build MSE as Linux loadable module

```
# cd avb-mse
# make KERNEL_SRC=/path/to/linux INCSHARED=/path/to/avb-streaming,/path/to/avb-mch
```

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

```
# cp *.ko ${ROOTFS_PATH}
```

${ROOTFS_PATH} rootfs of target.

6.2.2 Build MSE modules with the Linux kernel

Following operation is to build MSE modules with Linux Kernel.
At the same time, set up the Streaming Driver and MCH according to the operation of each user's manuals.

step1. Copy source files to the Linux kernel source.

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

${MODULE_SOURCE} MSE modules 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"
source "drivers/staging/avb-mse/Kconfig"

endif # STAGING
```

Add this line

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

```
# Makefile for staging directory

obj-y                           += media/
obj-$(CONFIG_SLICOSS)           += slicoss/
...
obj-$(CONFIG_BCM2708_VCHIQ)     += vc04_services/
obj-$(CONFIG_AVB_MSE)           += avb-mse/
```

Add this line

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

```
Device Drivers  --->
     [*]Staging drivers  --->
          [*] AVB MSE   --->
               <*>   MSE Core module (#6)
               [*]     MSE ioctl interface for configuration     (#8)
               [*]     MSE sysfs interface for configuration     (#8)
               [*]     MSE Packetizer AAF                        (#8)
               [*]     MSE Packetizer IEC61883_4                (#8)
               [*]     MSE Packetizer IEC61883_6                (#8)
               [*]     MSE Packetizer CVF_H264                   (#8)
               [*]     MSE Packetizer CVF_H264 Single NAL format (#8)
               [*]     MSE Packetizer CVF_MJPEG
               <*>     MSE EAVB Adapter (#7)
               <*>     MSE ALSA Adapter (#7)
               <*>     MSE V4L2 Adapter (#7)
               <*>     MSE MCH Adapter  (#7)
```

Notes: #7 This example, generate a kernel built-in object, if select the <M>, build as loadable module.
Notes: #8 If select the <N>, not build the function.

step5. If select <M> with kernel configuration, to copy modules. Next step integration, show 6.2.3.

```
# cd ${KERNEL_SOURCE}/drivers/staging/avb-mse/
# cp *.ko ${ROOTFS_PATH}
```

${ROOTFS_PATH} rootfs of target.

6.2.3 Integration devices

When you use the functions related to video, please execute following commands to use GStreamer with vspfilter.
Refer to the User's Manual "Linux Interface Specification GStreamer" for the information of setting GStreamer.

```
# modprobe -a mmngr mmngrbuf vspm vspm_if vsp2 uvcs_drv
```

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

Integrate the module of Streaming Driver on target boards (both talker and listener boards).

```
# insmod avb-streaming.ko
```

Integrate the module of MCH on target boards (both talker and listener boards).

```
# insmod mch_core.ko
```

Integrate build modules of MSE.

```
# insmod mse_core.ko
# insmod mse_adapter_eavb.ko
# insmod mse_adapter_alsa.ko
# insmod mse_adapter_v4l2.ko
# insmod mse_adapter_mch.ko
```

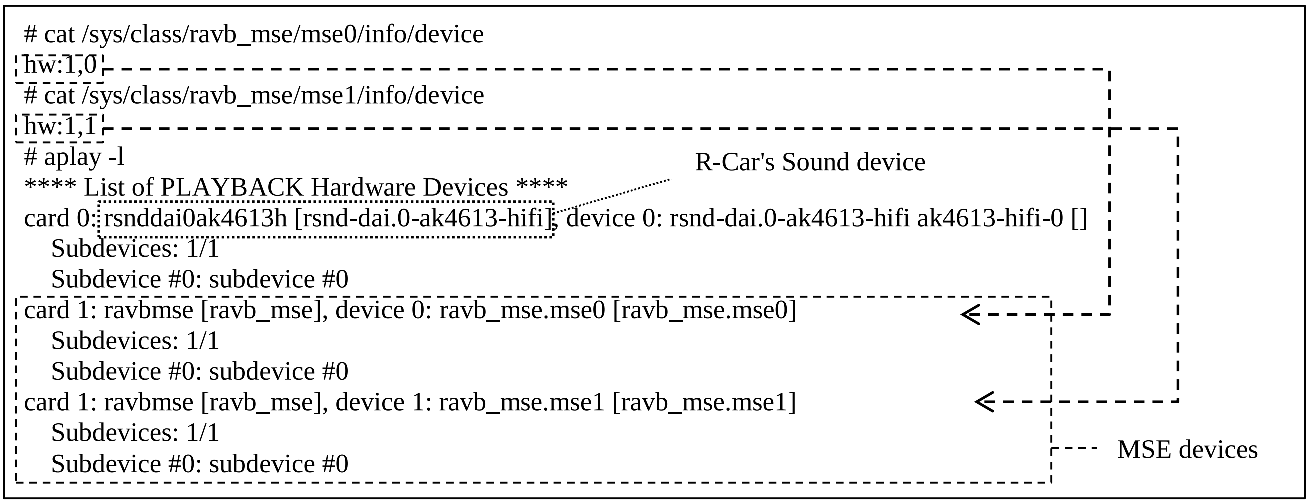
To check devices of MSE, executes following commands.
Note that following points:

- Sysfs interface names of MSE (/sys/class/ravb_mse/mseN) are defined by the order of installing the modules, mse_adapter_alsa.ko and mse_adapter_v4l2.ko.
- PCM Device number of Audio (hw:X,0, hw:X,1...) and Video device number (/dev/videoYY) might be changed by the order of installing other modules. Please execute check command (cat /sys/class/ravb_mse/mseN/device) before execute the commands introduced above.

To check MSE devices

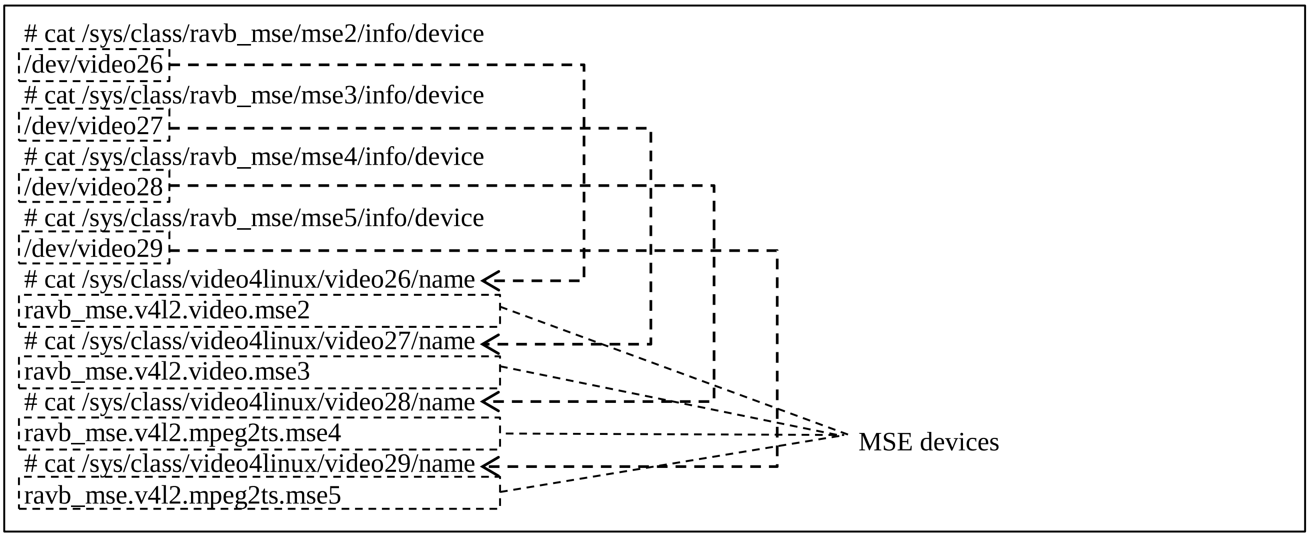
```
# ls -l /sys/class/ravb_mse/mse*
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse0 -> ../../devices/virtual/ravb_mse/mse0
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse1 -> ../../devices/virtual/ravb_mse/mse1
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse2 -> ../../devices/virtual/ravb_mse/mse2
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse3 -> ../../devices/virtual/ravb_mse/mse3
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse4 -> ../../devices/virtual/ravb_mse/mse4
lrwxrwxrwx 1 root root 0 Jan   1 00:00 /sys/class/ravb_mse/mse5 -> ../../devices/virtual/ravb_mse/mse5
```

To check ALSA devices.



```
# cat /sys/class/ravb_mse/mse0/info/device
hw:1,0
# cat /sys/class/ravb_mse/mse1/info/device
hw:1,1
# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rsnddai0ak4613h [rsnd-dai.0-ak4613-hifi], device 0: rsnd-dai.0-ak4613-hifi ak4613-hifi-0 []
   Subdevices: 1/1
   Subdevice #0: subdevice #0
card 1: ravbmse [ravb_mse], device 0: ravb_mse.mse0 [ravb_mse.mse0]
   Subdevices: 1/1
   Subdevice #0: subdevice #0
card 1: ravbmse [ravb_mse], device 1: ravb_mse.mse1 [ravb_mse.mse1]
   Subdevices: 1/1
   Subdevice #0: subdevice #0
```

To check V4L2 devices.



```
# cat /sys/class/ravb_mse/mse2/info/device
/dev/video26
# cat /sys/class/ravb_mse/mse3/info/device
/dev/video27
# cat /sys/class/ravb_mse/mse4/info/device
/dev/video28
# cat /sys/class/ravb_mse/mse5/info/device
/dev/video29
# cat /sys/class/video4linux/video26/name
ravb_mse.v4l2.video.mse2
# cat /sys/class/video4linux/video27/name
ravb_mse.v4l2.video.mse3
# cat /sys/class/video4linux/video28/name
ravb_mse.v4l2.mpeg2ts.mse4
# cat /sys/class/video4linux/video29/name
ravb_mse.v4l2.mpeg2ts.mse5
```

6.3 Option Setting

6.3.1 Module Parameters

The module parameter that is defined in this module is shown below.

Table 6.1 Module parameters

Module	Module parameters	Default value	Range	Explanation
mse_core.ko	major	0(dynamic allocate)	0 - 511	Current device major number of the MSE ioctl devices. If it is out of range, module cannot be installed.
mse_adapter_alsa.ko	alsa_devices	2	1 - 8	Number of ALSA device created MSE. If it is out of range, module cannot be installed.
mse_adapter_v4l2.ko	v4l2_video_devices	2	0 to 10 (#9)	Number of V4L2 device created MSE to use video. If it is out of range, module cannot be installed.
	v4l2_mpeg2ts_devices	2	0 to 10 (#9)	Number of V4L2 device created MSE to use MPEG2-TS. If it is out of range, module cannot be installed.

Notes: #9 The totals of v4l2_video_devices and v4l2_mpeg2ts_devices must be at least 1 and max 10.

6.3.2 Kernel Parameters

There are no kernel parameters.

7. Example of how to use the MSE

This chapter describes some example of how to run applications.

7.1 How to send audio data

This chapter shows several how to transport audio data via the Ethernet AVB network.

7.1.1 How to send speaker test tone using MSE

This example shows how to transport a generated speaker test tone via the Ethernet AVB network.

Set configuration parameters for talker, see Table 7.1.
Parameter does not present in Table 7.1 are set default.

Table 7.1 List of configuration for how to send audio using MSE in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name

Notes: #10 Check own MAC address using ifconfig command and set it.

Set configuration parameters for listener, see Table 7.2.
Parameter does not present in Table 7.2 are set default.

Table 7.2 List of configuration for how to send audio using MSE in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name

Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Set up Ethernet MAC addresses, priority and unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0        Link encap:Ethernet   HWaddr 76:90:50:00:00:00 ------------- MAC Address
             inet addr:192.168.0.6    Bcast:192.168.0.255    Mask:255.255.255.0
             UP BROADCAST RUNNING MULTICAST    MTU:1500    Metric:1
             ...
# echo "91e0f0000e80" >/sys/class/ravb_mse/mse0/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse0/avtp_tx_param/src_mac
# echo "3" >/sys/class/ravb_mse/mse0/avtp_tx_param/priority
# echo "1" >/sys/class/ravb_mse/mse0/avtp_tx_param/uniqueid
```

Set up sysfs on Listener board. Set src_mac talker MAC address. And set unique_id to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse0/avtp_rx_param/streamid
```

If change to sending format, both talker board and listener board needed following command.

```
# echo "iec61883-6" > /sys/class/ravb_mse/mse0/packetizer/name
```

Set up the audio output on Listener board

```
# amixer set "DVC Out" 1%
```

Run an ALSA application on Talker board.

```
# speaker-test -r 48000 -t sine -c 2 -F S16_LE -D `cat /sys/class/ravb_mse/mse0/info/device`
```

Run an ALSA application on Listener board.
After this operation, Listener board playbacks the sound that Talker board sends. (#12)

```
# arecord -D `cat /sys/class/ravb_mse/mse0/info/device` -c 2 -r 48000 -f S16_LE | aplay -D plughw:0,0
```

Notes: #12 plughw:0,0 is R-Car's sound device confirmed by 6.2.3.

7.1.2 How to send PCM audio data using MSE with MCH

This example shows how to transport PCM data and media clock via the Ethernet AVB network.
Listener is adjustment the audio clock using AVTP timestamps in receiving AVTP frames.

Set configuration parameters for talker, see Table 7.3.
Parameter does not present in Table 7.3 are set default.

Table 7.3 List of configuration for how to send audio using MSE with MCH in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
ptp_config/type	"capture"	gPTP type
ptp_config/deviceid	0	gPTP device number
ptp_config/capture_ch	2	gPTP capture channel
ptp_config/capture_freq	300	gPTP capture timestamps per Second
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name
delay_time/tx_delay_time_ns	40000000	delay time of capture data

Notes: #10 Check own MAC address using ifconfig command and set it.

Set configuration parameters for listener, see Table 7.4.
Parameter does not present in Table 7.4 are set default.

Table 7.4 List of configuration for how to send audio using MSE with MCH in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)
ptp_config/type	"capture"	gPTP type
ptp_config/deviceid	0	gPTP device number
ptp_config/capture_ch	2	gPTP capture channel
ptp_config/capture_freq	300	gPTP capture timestamps per Second
mch_config/enable	1	Enable MCH
media_audio_config/crf_type	"not use"	CRF type
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name
delay_time/rx_delay_time_ns	40000000	delay time of capture data

Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Step1:
Set up Ethernet MAC addresses, priority and Unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0       Link encap:Ethernet  HWaddr 76:90:50:00:00:00 ----------------------  MAC Address
            inet addr:192.168.0.6   Bcast:192.168.0.255   Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
            ...
# echo "91e0f0000e80" >/sys/class/ravb_mse/mse0/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse0/avtp_tx_param/src_mac
# echo "3" >/sys/class/ravb_mse/mse0/avtp_tx_param/priority
# echo "1" >/sys/class/ravb_mse/mse0/avtp_tx_param/uniqueid
```

Set up sysfs on Listener board. Set src_mac talker MAC address. And set unique_id to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse0/avtp_rx_param/streamid
```

Step2:
Enable ptp capture timestamps on both of Talker and Listener board.

```
# echo "capture" > /sys/class/ravb_mse/mse0/ptp_config/type
# echo "0" > /sys/class/ravb_mse/mse0/ptp_config/deviceid
# echo "2" > /sys/class/ravb_mse/mse0/ptp_config/capture_ch
# echo "300" > /sys/class/ravb_mse/mse0/ptp_config/capture_freq
```

Step3:
Enable media clock recovery on Listener board.

```
# echo "1" > /sys/class/ravb_mse/mse0/mch_config/enable
# echo "not use" > /sys/class/ravb_mse/mse0/media_audio_config/crf_type
```

Step4:
Play MSE with audio device on talker and listener.

If change to sending format, both talker board and listener board needed following command.

```
# echo "iec61883-6" > /sys/class/ravb_mse/mse0/packetizer/name
```

Set up the audio input on Talker board.

```
# amixer set "DVC In" 50%
```

Set up the audio output on Listener board.

```
# amixer set "DVC Out" 1%
```

Execute ptp4l both talker board and listener to synchronize the PTP time. (#13)

```
# ptp4l -i eth0 -f /etc/linuxptp/avb-demoapps.cfg &
```

Notes: #13 Overrun and underrun errors of ALSA will be occurred when starting MSE streaming before time-synchronization of ptp4l become stable state.
In the case of using Media clock recovery, MSE control processing interval of media stream by using gPTP timer interrupt. When gPTP timer has big jump by time-synchronization process, interval of gPTP timer interrupt different from interval of alsa's period significantly. Those errors of ALSA will be occurred by this difference.
It is recommended to start MSE streaming after time-synchronization of ptp4l become stable state.
It takes about 10 seconds after starting ptp4l and ptp4l status can be checked from system log by using journalctl command.

Calculate period time and buffer time in nanoseconds.
Period time should be divisible by capture interval to improve accuracy.
Following commands set period time and buffer time near default of aplay(2048).

```
# SAMPLE_RATE=48000
# MIN_PERIOD=$(expr ${SAMPLE_RATE} / 300)
# PERIOD=$(expr \( 2048 / ${MIN_PERIOD} \) \* ${MIN_PERIOD})
# PERIOD_TIME_NS=$(expr \( ${PERIOD} \* 1000000000 \) / ${SAMPLE_RATE})
# PERIOD_TIME=$(expr ${PERIOD_TIME_NS} / 1000)
# BUFFER_TIME=$(expr ${PERIOD_TIME_NS} \* 4 / 1000)

# echo ${PERIOD_TIME} ${BUFFER_TIME} ${PERIOD_TIME_NS}
40000 160000 40000000
```

On Talker board, connect a mic to MIC-IN.
Set delay time 1 period time inside MSE.
Run an ALSA application with period time and buffer time option.

```
# echo ${PERIOD_TIME_NS} > /sys/class/ravb_mse/mse0/delay_time/tx_delay_time_ns
# arecord -D plughw:0,0 \
        -c 2 -r ${SAMPLE_RATE} -f S16_LE -F ${PERIOD_TIME} -B ${BUFFER_TIME} | \
   aplay -D `cat /sys/class/ravb_mse/mse0/info/device` \                         #14
        -F ${PERIOD_TIME} -B ${BUFFER_TIME}
```

On Listener board, set delay time according to user application and run an ALSA application.
After this operation, Listener board playbacks the sound that Talker board sends.

```
# echo ${PERIOD_TIME_NS} > /sys/class/ravb_mse/mse0/delay_time/rx_delay_time_ns
# arecord -D `cat /sys/class/ravb_mse/mse0/info/device` \
        -c 2 -r ${SAMPLE_RATE} -f S16_LE -F ${PERIOD_TIME} -B ${BUFFER_TIME} | \   #14
   aplay -D plughw:0,0 \
        -F ${PERIOD_TIME} -B ${BUFFER_TIME}
```

Notes: #14 If the network is disconnected, time synchronization will be incomplete.
In that case execute the command again on both board.

7.1.3 How to send PCM audio and media clock from other device

This example shows how to transport media clock on other device (like a listener) via the Ethernet AVB network.
Talker and listener are adjustment the audio clock using AVTP timestamps in receiving AVTP frames.
This example shows Talker adjustment.

Set configuration parameters for talker, see Table 7.5.
Parameter does not present in Table 7.5 are set default.

Table 7.5 List of configuration for how to send audio and media clock in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
ptp_config/type	"capture"	gPTP type
ptp_config/deviceid	0	gPTP device number
ptp_config/capture_ch	2	gPTP capture channel
ptp_config/capture_freq	300	gPTP capture timestamps per Second
mch_config/enable	1	Enable MCH
media_audio_config/crf_type	"rx"	CRF type
avtp_rx_param_crf/streamid	7690500000010002	CRF Stream ID (#15)
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name
delay_time/tx_delay_time_ns	40000000	delay time of capture data

Notes: #10 Check own MAC address using ifconfig command and set it.
Notes: #15 The streamid must be set to be the same as CRF talker's src_mac and uniqueid parameter.

Set configuration parameters for listener, see Table 7.6.
Parameter does not present in Table 7.6 are set default.

Table 7.6 List of configuration for how to send audio and media clock in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)
ptp_config/type	"capture"	gPTP type
ptp_config/deviceid	0	gPTP device number
ptp_config/capture_ch	2	gPTP capture channel
ptp_config/capture_freq	300	gPTP capture timestamps per Second
media_audio_config/crf_type	"tx"	CRF type
avtp_tx_param_crf/dst_mac	91e0f0000e81	Ethernet destination MAC address for CRF
avtp_tx_param_crf/src_mac	769050000001	Ethernet source MAC address for CRF (#10)
avtp_tx_param_crf/priority	3	CRF stream priority
avtp_tx_param_crf/uniqueid	2	CRF stream unique ID
packetizer/name	"aaf_pcm" or "iec61883-6"	Packet format name
delay_time/rx_delay_time_ns	40000000	delay time of capture data

Notes: #10 Check own MAC address using ifconfig command and set it.
Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Step1:
Set up Ethernet MAC addresses, priority and Unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 76:90:50:00:00:00 ---------- MAC Address
           inet addr:192.168.0.6   Bcast:192.168.0.255   Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
           ...
# echo "91e0f0000e80" > /sys/class/ravb_mse/mse0/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse0/avtp_tx_param/src_mac
# echo "3" > /sys/class/ravb_mse/mse0/avtp_tx_param/priority
# echo "1" > /sys/class/ravb_mse/mse0/avtp_tx_param/uniqueid
```

Set up sysfs on Listener board. Set src_mac talker MAC address. And set unique_id to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse0/avtp_rx_param/streamid
```

Step2:
Enable ptp capture timestamps on both of Talker and Listener board.

```
# echo "capture" > /sys/class/ravb_mse/mse0/ptp_config/type
# echo "0" > /sys/class/ravb_mse/mse0/ptp_config/deviceid
# echo "2" > /sys/class/ravb_mse/mse0/ptp_config/capture_ch
# echo "300" > /sys/class/ravb_mse/mse0/ptp_config/capture_freq
```

Step3:
Set up Ethernet MAC address and Unique ID among the information of CRF stream.

Set up sysfs on Listener board.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 76:90:50:00:00:01 ---------- MAC Address
           inet addr:192.168.0.5   Bcast:192.168.0.255   Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
           ...
# echo "tx" > /sys/class/ravb_mse/mse0/media_audio_config/crf_type
# echo "91e0f0000e81" > /sys/class/ravb_mse/mse0/avtp_tx_param_crf/dst_mac
# echo "769050000001" > /sys/class/ravb_mse/mse0/avtp_tx_param_crf/src_mac
# echo "3" > /sys/class/ravb_mse/mse0/avtp_tx_param_crf/priority
# echo "2" > /sys/class/ravb_mse/mse0/avtp_tx_param_crf/uniqueid
```

Set up sysfs on Talker board. Set send_clock_src_mac listener MAC address. Set send_clock_unique_id to same value of listener.

```
# echo "1" > /sys/class/ravb_mse/mse0/mch_config/enable
# echo "rx" > /sys/class/ravb_mse/mse0/media_audio_config/crf_type
# echo "7690500000010002" > /sys/class/ravb_mse/mse0/avtp_rx_param_crf/streamid
```

Step4:
Play MSE with audio device on talker and listener.

If change to sending format, both talker board and listener board needed following command.

```
# echo "iec61883-6" > /sys/class/ravb_mse/mse0/packetizer/name
```

Set up the audio input on Talker board.

```
# amixer set "DVC In" 50%
```

Set up the audio output on Listener board.

```
# amixer set "DVC Out" 1%
```

Execute ptp4l both talker board and listener to synchronize the PTP time. (#13)

```
# ptp4l -i eth0 -f /etc/linuxptp/avb-demoapps.cfg &
```

Notes: #13 Overrun and underrun errors of ALSA will be occurred when starting MSE streaming before time-synchronization of ptp4l become stable state.
In the case of using Media clock recovery, MSE control processing interval of media stream by using gPTP timer interrupt. When gPTP timer has big jump by time-synchronization process, interval of gPTP timer interrupt different from interval of alsa's period significantly. Those errors of ALSA will be occurred by this difference.
It is recommended to start MSE streaming after time-synchronization of ptp4l become stable state.
It takes about 10 seconds after starting ptp4l and ptp4l status can be checked from system log by using journalctl command.

Calculate period time and buffer time in nanoseconds.
Period time should be divisible by capture interval to improve accuracy.
Following commands set period time and buffer time near default of aplay(2048).

```
# SAMPLE_RATE=48000
# MIN_PERIOD=$(expr ${SAMPLE_RATE} / 300)
# PERIOD=$(expr \( 2048 / ${MIN_PERIOD} \) \* ${MIN_PERIOD})
# PERIOD_TIME_NS=$(expr \( ${PERIOD} \* 1000000000 \) / ${SAMPLE_RATE})
# PERIOD_TIME=$(expr ${PERIOD_TIME_NS} / 1000)
# BUFFER_TIME=$(expr ${PERIOD_TIME_NS} \* 4 / 1000)

# echo ${PERIOD_TIME} ${BUFFER_TIME} ${PERIOD_TIME_NS}
40000 160000 40000000
```

On Talker board, connect a mic to MIC-IN.
Set delay time 1 period time inside MSE.
Run an ALSA application with period time and buffer time option.

```
# echo ${PERIOD_TIME_NS} > /sys/class/ravb_mse/mse0/delay_time/tx_delay_time_ns
# arecord -D plughw:0,0 \
        -c 2 -r ${SAMPLE_RATE} -f S16_LE -F ${PERIOD_TIME} -B ${BUFFER_TIME} | \
   aplay -D `cat /sys/class/ravb_mse/mse0/info/device` \                          #14
        -F ${PERIOD_TIME} -B ${BUFFER_TIME}
```

On Listener board, set delay time according to user application and run an ALSA application.
After this operation, Listener board playbacks the sound that Talker board sends.

```
# echo ${PERIOD_TIME_NS} > /sys/class/ravb_mse/mse0/delay_time/rx_delay_time_ns
# arecord -D `cat /sys/class/ravb_mse/mse0/info/device` \
        -c 2 -r ${SAMPLE_RATE} -f S16_LE -F ${PERIOD_TIME} -B ${BUFFER_TIME} | \   #14
   aplay -D plughw:0,0 \
        -F ${PERIOD_TIME} -B ${BUFFER_TIME}
```

Notes: #14 If the network is disconnected, time synchronization will be incomplete.
In that case execute the command again on both board.

7.2 How to send H.264 video data

This chapter shows how to transport H.264 video data via the Ethernet AVB network.

7.2.1 How to send H.264 video file

This example shows how to transport a H.264 video file via the Ethernet AVB network.

Set configuration parameters for talker, see Table 7.7.
Parameter does not present in Table 7.7 are set default.

Table 7.7 List of configuration for how to send H.264 video file in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
packetizer/name	"cvf_h264" or "cvf_h264_d13"	Packet format name

Notes: #10 Check own MAC address using ifconfig command and set it.

Set configuration parameters for listener, see Table 7.8.
Parameter does not present in Table 7.8 are set default.

Table 7.8 List of configuration for how to send H.264 video file in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)
packetizer/name	"cvf_h264" or "cvf_h264_d13"	Packet format name

Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Prepare a MP4 file. If you have no file, you can create a file using following command on Talker board.

```
# gst-launch-1.0 \
   videotestsrc num-buffers=300 ! \
   video/x-raw, format=NV12, framerate=30/1, width=640, height=480 ! \
   omxh264enc \
      control-rate=2 \
      target-bitrate=10485760 \
      interval_intraframes=14 \
      periodicty-idr=2 ! \
   video/x-h264, profile=\(string\)high, level=\(string\)4.2 ! \
   h264parse ! \
   video/x-h264, stream-format=avc, alignment=au ! \
   qtmux ! \
   queue ! \
   filesink location=video.mp4
```

Set up Ethernet MAC addresses, priority and Unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0          Link encap:Ethernet   HWaddr 76:90:50:00:00:00 - - - - - - - - - - - - - - - - -   MAC Address
               inet addr:192.168.0.6   Bcast:192.168.0.255   Mask:255.255.255.0
               UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
               ...
# echo "91e0f0000e80" >/sys/class/ravb_mse/mse2/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse2/avtp_tx_param/src_mac
# echo "3" >/sys/class/ravb_mse/mse2/avtp_tx_param/priority
# echo "1" >/sys/class/ravb_mse/mse2/avtp_tx_param/uniqueid
```

Set up sysfs on Listener board. Set streamid from talker MAC address and uniqueid to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse2/avtp_rx_param/streamid
```

Change the sending format H.264, Both Talker and Listener boards execute the following command.

```
# echo "cvf_h264" > /sys/class/ravb_mse/mse2/packetizer/name
```

Play MSE with video device, both talker and listener boards execute the following command.

Talker board operations

```
# gst-launch-1.0 \
        filesrc location=./video.mp4 \
      ! qtdemux \
      ! h264parse \
      ! video/x-h264, width=640, height=480, stream-format=byte-stream, framerate=30/1, colorimetry=bt601 \
      ! v4l2sink sync=true device=`cat /sys/class/ravb_mse/mse2/info/device` show-preroll-frame=false
```

Listener board operations

```
# gst-launch-1.0 \
      v4l2src device=`cat /sys/class/ravb_mse/mse2/info/device` \
    ! video/x-h264, width=640, height=480, stream-format=byte-stream, framerate=30/1 \
    ! h264parse \
    ! omxh264dec \
    ! waylandsink sync=true
```

7.2.2 How to send a video test pattern

This example shows how to transport a test pattern created by GStreamer plugin 'videotestsrc' via the Ethernet AVB network.

Use the settings in chapter 7.2.1 except for the following Talker board operation.

```
# gst-launch-1.0 \
        videotestsrc \
      ! video/x-raw, width=640, height=480, framerate=30/1 \
      ! omxh264enc target-bitrate=15000000 \
      ! h264parse \
      ! video/x-h264, width=640, height=480, stream-format=byte-stream, framerate=30/1 \
      ! v4l2sink sync=true device=`cat /sys/class/ravb_mse/mse2/info/device` show-preroll-frame=false
```

When you execute the command mentioned above, sometimes the frame rate is lower than you set. The reason may be the delay of performance that building the test pattern in software procedure (videotestsrc).

7.3 How to send MJPEG video data

This example shows how to transport a MJPEG Video file via the Ethernet AVB network.

Set configuration parameters for talker, see Table 7.9.
Parameter does not present in Table 7.9 are set default.

Table 7.9 List of configuration for how to send MJPEG video file in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
packetizer/name	"cvf_mjpeg"	Packet format name
media_video_config/fps_numerator	30	Framerate parameter (frames)
media_video_config/fps_denominator	1	Framerate parameter (seconds)

Notes: #10 Check own MAC address using ifconfig command and set it.

Set configuration parameters for listener, see Table 7.10.
Parameter does not present in Table 7.10 are set default.

Table 7.10 List of configuration for how to send MJPEG video file in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)
packetizer/name	"cvf_mjpeg"	Packet format name

Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Prepare a MJPEG file. If you have no file, you can create a file using following command on Talker board

```
# gst-launch-1.0 \
        videotestsrc pattern=0 num-buffers=300 ! \
        video/x-raw, format=I420, framerate=30/1, width=640, height=480 ! \
        jpegenc ! \
        qtmux ! \
        filesink location=./movie.mov
```

Set up Ethernet MAC addresses, priority and Unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0        Link encap:Ethernet  HWaddr 76:90:50:00:00:00 - - - - - - - - -   MAC Address
             inet addr:192.168.0.6   Bcast:192.168.0.255   Mask:255.255.255.0
             UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
             ...
# echo "91e0f0000e80" >/sys/class/ravb_mse/mse2/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse2/avtp_tx_param/src_mac
# echo "3" >/sys/class/ravb_mse/mse2/avtp_tx_param/priority
# echo "1" >/sys/class/ravb_mse/mse2/avtp_tx_param/uniqueid
```

Set up sysfs on Listener board. Set streamid from talker MAC address and uniqueid to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse2/avtp_rx_param/streamid
```

Change the sending format MJPEG, both talker and listener boards execute the following command.

Talker board operations

```
# echo "cvf_mjpeg" > /sys/class/ravb_mse/mse2/packetizer/name
# echo "30" >/sys/class/ravb_mse/mse2/media_video_config/fps_numerator
# echo "1" >/sys/class/ravb_mse/mse2/media_video_config/fps_denominator
```

Listener board operations

```
# echo "cvf_mjpeg" > /sys/class/ravb_mse/mse2/packetizer/name
```

Play MSE with video device, both talker and listener boards execute the following command.

Talker board operations

```
# gst-launch-1.0 \
        filesrc location=./movie.mov ! \
        qtdemux ! \
        jpegparse ! \
        videorate ! \
        image/jpeg,width=640,height=480,framerate=30/1,colorimetry=bt601 ! \
        v4l2sink sync=true device=`cat /sys/class/ravb_mse/mse2/info/device` show-preroll-frame=false
```

Listener board operations

```
# gst-launch-1.0 \
        v4l2src device=`cat /sys/class/ravb_mse/mse2/info/device` ! \
        image/jpeg, width=640, height=480, framerate=30/1 ! \
        jpegparse ! \
        jpegdec ! \
        vspfilter ! \
        video/x-raw, format= NV12, width=640, height=480 ! \
        waylandsink sync=true
```

When you execute the command mentioned above, sometimes the frame rate is lower than you set. The reason may be the delay of performance using software decoder (jpegdec).

7.4 How to send MPEG2-TS data

This example shows how to transport a MPEG-2-TS file via the Ethernet AVB network.

Set configuration parameters for talker, see Table 7.11.
Parameter does not present in Table 7.11 are set default.

Table 7.11 List of configuration for how to send MPEG2-TS file in talker

Configuration parameter	Value	Note
avtp_tx_param/dst_mac	91e0f0000e80	Ethernet destination MAC address
avtp_tx_param/src_mac	769050000000	Ethernet source MAC address (#10)
avtp_tx_param/priority	3	AVB stream priority
avtp_tx_param/uniqueid	1	AVB stream unique ID
media_mpeg2ts_config/transmit_mode	pcr	MPEG2-TS transmit mode.

Notes: #10 Check own MAC address using ifconfig command and set it.

Set configuration parameters for listener, see Table 7.12.
Parameter does not present in Table 7.12 are set default.

Table 7.12 List of configuration for how to send MPEG2-TS file in listener

Configuration parameter	Value	Note
avtp_rx_param/streamid	7690500000000001	AVB Stream ID (#11)

Notes: #11 The streamid must be set to be the same as talker's src_mac and uniqueid parameter.

Prepare MPEG2-TS file contains H.264 video and AAC-LC Audio.

Set up Ethernet MAC addresses, priority and Unique ID among the information of sending stream.
Set up sysfs on Talker board.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 76:90:50:00:00:00 ------------------ MAC Address
          inet addr:192.168.0.6   Bcast:192.168.0.255   Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
          ...
# echo "91e0f0000e80" >/sys/class/ravb_mse/mse4/avtp_tx_param/dst_mac
# echo "769050000000" > /sys/class/ravb_mse/mse4/avtp_tx_param/src_mac
# echo "3" >/sys/class/ravb_mse/mse4/avtp_tx_param/priority
# echo "1" >/sys/class/ravb_mse/mse4/avtp_tx_param/uniqueid
# echo "pcr" >/sys/class/ravb_mse/mse4/media_mpeg2ts_config/transmit_mode
```

Set up sysfs on Listener board. Set streamid from talker MAC address and uniqueid to same value of talker.

```
# echo "7690500000000001" > /sys/class/ravb_mse/mse4/avtp_rx_param/streamid
```

Play MSE with video device, both talker and listener boards execute the following command.
Note that the following command is a procedure for playing the MPEG2-TS file contains H.264 video and AAC-LC Audio. If need to use other formats, see GStreamer's documents.

Talker board operations

```
# gst-launch-1.0 \
        filesrc location=./movie.ts blocksize=36096 \
      ! video/mpegts, systemstream=true \
      ! v4l2sink sync=true device=`cat /sys/class/ravb_mse/mse4/info/device` show-preroll-frame=false
```

In this example, in order to make the input size of MPEG2-TS data to a multiple of the MPEG2-TS packet size, set the common multiple of TS and M2TS format to the blocksize property of filesrc.

Listener board operations (H.264 video and AAC-LC audio)

```
# gst-launch-1.0 \
        v4l2src device=`cat /sys/class/ravb_mse/mse4/info/device` \
      ! queue \
   ! tsdemux name=demux \
      ! queue \
      ! aacparse \
      ! omxaaclcdec \
      ! audioconvert ! audioresample \
      ! audio/x-raw, format=S16LE, rate=44100, channels=2 \
      ! alsasink device='plughw:0,0' \
   demux. \
      ! queue \
      ! h264parse \
      ! omxh264dec \
      ! waylandsink sync=true
```

8. Appendix

8.1 Sysfs Comparison List

Directory configuration of sysfs interface has been significantly updated from v0.30 to v0.40. For this reason, tables for comparison with the v0.30 specifications are shown below.

Table 8.1 Device node comparison list

v0.30	v0.40(latest)	Note
/sys/module/mse_core/mseX	/sys/class/ravb_mse/mseX	X: Device number

Table 8.2 Sysfs parameter comparison list

v0.30		v0.40(latest)	Note
type		-	deletion
kind		info/type	
device		info/device	
network_adapter_name		network_device/module_name	
network_adapter_name_tx		network_device/device_name_tx	
		network_device/device_name_tx_crf	Separate for CRF
network_adapter_name_rx		network_device/device_name_rx	
		network_device/device_name_rx_crf	Separate for CRF
packetizer_name		packetizer/name	
AVTP parameters for talker	dst_mac	avtp_tx_param/dst_mac	
	src_mac	avtp_tx_param/src_mac	
	unique_id	avtp_tx_param/uniqueid	
	send_clock_dst_mac	avtp_tx_param_crf/dst_mac	
	send_clock_src_mac	avtp_tx_param_crf/src_mac	
	send_clock_unique_id	avtp_tx_param_crf/uniqueid	
	vlan	avtp_tx_param/vlan	
		avtp_tx_param_crf/vlan	Separate for CRF
	priority	avtp_tx_param/priority	
		avtp_tx_param_crf/priority	Separate for CRF
AVTP parameters for listener	src_mac	avtp_rx_param/streamid[7-2]	
	unique_id	avtp_rx_param/streamid[1-0]	
	send_clock_src_mac	avtp_rx_param_crf/streamid[7-2]	Separate for listener
	send_clock_unique_id	avtp_rx_param_crf/streamid[1-0]	Separate for listener
media_clock_type send_clock		media_audio_config/crf_type	See the Table 8.3
bytes_per_frame		media_audio_config/samples_per_frame	Separate for audio
		media_video_config/bytes_per_frame	
fps_frames		media_video_config/fps_denominator	
fps_seconds		media_video_config/fps_numerator	
bitrate		media_video_config/bitrate	
-		media_mpeg2ts_config/tspackets_per_frame	Support from v0.40
-		media_mpeg2ts_config/bitrate	Support from v0.40
-		media_mpeg2ts_config/pcr_pid	Support from v0.40
ptp_clock		ptp_config/type	
ptp_clock_device		ptp_config/deviceid	
ptp_capture_ch		ptp_config/capture_ch	
ptp_capture_freq		ptp_config/capture_freq	
recovery_capture_freq		ptp_config/recovery_capture_freq	
media_clock_recovery		mch_config/enable	
max_transit_time		delay_time/max_transit_time_ns	
talker_delay_time		delay_time/tx_delay_time_ns	
listener_delay_time		delay_time/rx_delay_time_ns	

Table 8.3 crf_type comparison list

v0.30		**v0.40(latest)**	**Note**
media_clock_type	**send_clock**	**media_audio_config/crf_type**	
avtp	Don't care	not use	
crf	0	rx	
crf	1	tx	

REVISION HISTORY	Ethernet AVB Software Media Streaming Engine User’s Manual: Software

Rev.	Date	Description	
		Page	Summary
0.10	Apr. 22, 2016	-	First edition issued
0.20	May. 20, 2016	1	1.1 Overview Add M3 support
		1	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
0.30	Aug. 24, 2016	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		3	1.4. Terminology Update of Table 1.3 - Add terms of CRF, MJPEG, GStreamer and VSP. - Modify ordering of each term.
		4	2.2. Module Configuration Add description of the MCH.
		6-7	3. Function Add figures of the MCH and CRF following. - Figure 3.2. Module relationship (ALSA) with MCH - Figure 3.3. Module relationship (ALSA) with CRF stream
		9	3.1. Packetize/De-packetize Update of Table 3.1 - Add CVF H.264 format (Remove T.B.D). - Add CVF MJPEG format. - Add IEC 61883-4 format for future version.
		10	3.3. V4L2 adapter Update of Table 3.3 - Add MJPEG format.
		10	3.3. V4L2 adapter Update of Table 3.4 - Add MPEG2-TS format for future version.
		11	3.4. EAVB Adapter Update of Table 3.7 - Add MJPEG format
		13, 22-26	3.4.1. Ethernet AVB Packet Format Add description of the new support formats following. - Chapter 3.4.1.2. AVTP IEC61883-4 MPEG2-TS format for future version. - Chapter 3.4.1.6. H.264 Compressed Video Format (CVF H264). - Chapter 3.4.1.7. MJPEG Compressed Video Format (CVF MJPEG). - Chapter 3.4.1.8. Clock Reference Format (CRF).
		27-28	4. External Interface Update of Table 4.1 - Modify this table for describe to parameters regarding audio configuration. - Add new parameters about MCH and CRF
		29	4. External Interface Add Table 4.2 for describe to parameters regarding video configuration.
		31	5. Integration Update of Figure 5.1 - Update new files
		32	5.2.1. Build kernel modules only Remove step of modify the kernel configuration.

		34	5.2.3. Integration devices - Add procedure of load MMP kernel modules. - Add description of relationship of the MSE module and /dev/video files.
		36-40	6.1. How to send PCM audio data using MSE - Update the description of each procedure. - Add Table 6.1 List of sysfs configuration value for PCM audio. - Add descriptions of regarding MCH and CRF.
		42-44	6.2 How to send H.264 video data using MSE - Update the description of each procedure. - Add Table 6.2 List of sysfs video configuration value for H.264 video.
		45-46	6. Example of how to use the MSE Add descriptions new support use case following. - Chapter 6.3. How to send MJPEG video file - Chapter 6.4. How to send MPEG2-TS for future version.
0.40	Jan. 13, 2017	-	Revise the chapter composition according to support for new functions. - Move Chapter 3.1 Packetize/De-packetize to Chapter 3.5. - Move Chapter 3.2 ALSA adapter to Chapter 3.1. - Move Chapter 3.3 V4L2 adapter to Chapter 3.2. - Move Chapter 3.4 EAVB adapter to Chapter 3.3. - Move descriptions for packet format in Chapter 3.4.1 Ethernet AVB Packet Format to Chapter 5.1 Ethernet Frame Format in the new Chapter 5. Packet Format. - Add the new Chapter 3.4 MCH adapter. - Add the new Chapter 3.6 Network Traffic Shaping. - Move Chapter 5. Integration to Chapter 6. - Move Chapter 6. Example of how to use the MSE to Chapter 7. - Add new Chapter 8. Appendix.
		3	1.4 Terminology Update of Table 1.3 - Add "MPEG-2 TS", "PCR" and "PID" for terms.
		4	2.2 Module Configuration Update of Figure 2-1 - Add MSE Configuration, MCH Adapter and PTP dummy.
		6	3 Function Update of Figure 3-2 - Add MCH Adapter at Talker and Listener.
		7	3 Function Update of Figure 3-3 - Add MCH Adapter at Talker and Listener. - Add Media Clock Recovery Handler at Listener. - Remove PTP Driver at Talker and Listener.
		9	3.1 ALSA Adapter Update of Table 3.1 - Add Sample format S32_LE/S32_BE, S18_3LE/S18_3BE, S20_3LE/S20_3BE and S24_3LE/S24_3BE - Change Sampling rate max value from 96000Hz to 192000Hz
		9	3.2 V4L2 Adapter Update Table 3.2 for V4L2 H.264 - Add video format H.264 AVC, remove video format MJPEG. - Update picture size and framerate. - Remove the capability of bitrate.
		9	3.2 V4L2 Adapter Update of Table 3.3 for V4L2 MJPEG - Split Table from old V4L2 adapter capability. - Update picture size and framerate.
		9	3.2 V4L2 Adapter Update of Table 3.4 V4L2 adapter capability for MPEG-2 TS

		10	3 Function Add descriptions new support functions following. - Chapter 3.4. MCH Adapter
		10	3.5 Packetize/De-packetize - Remove T.B.D at MPEG-2 TS.
		11	3.5.1 Ethernet AVB Packet Format Update of Table 3.6 for IEC 61883-6 - Remove packet format AAF PCM - Add sampling rate 88200Hz, 96000Hz, 176400Hz and 192000Hz. - Add bit depth 20bit integer. - Update channel max from 32 to 24.
		11	3.5.1 Ethernet AVB Packet Format Add Table 3.7 for AAC PCM - Split Table from old audio packet format. - Add sampling rate 8000Hz, 16000Hz, 88200Hz, 96000Hz, 176400Hz and 192000Hz. - Add bit depth 32bit integer. - Update channel max from 32 to 24.
		11	3.5.1 Ethernet AVB Packet Format Update of Table 3.8 - Remove line of picture size, framerate and bitrate.
		11	3.5.1 Ethernet AVB Packet Format Update of Table 3.9 - Update picture size format. - Remove line of framerate and bitrate.
		11	3.5.1 Ethernet AVB Packet Format Add following tables. - Add Table 3.10 Video packet format for MPEG-2 TS. - Add Table 3.11 Control packet format for CRF.
		12	3.6 Network Traffic Shaping - Add descriptions for network traffic shaping.
		13	4. External Interface Add Table 4.1 for device node.
		14-16	4.1 Sysfs Interface Revise configuration parameters - About details of changing, see the Chapter 8.1 Sysfs Comparison List. - Update of Table 4.2 for sysfs device node. - Add Table 4.3 for common configuration parameters. - Remove common configuration parameters from Table 4.4 and Table 4.5.
		17-50	4. External Interface Add descriptions new support interface following. - Chapter 4.2 Ioctl interface - Chapter 4.3 Adapter interface
		52-53	5. Packet format Add descriptions new support packet format following. - Chapter 5.2 AVTP IEC61883-4 MPEG2-TS format
		67	6.1 Directory Configuration. Update of Figure 6-1. - Add new files.
		69	6.2.2 Build MSE modules with the Linux kernel Add function MSE MCH Adapter at step4.
		70-71	6.2.3 Integration devices Update descriptions check method following. - Update To check MSE devices - Update To check ALSA devices - Update To check V4L2 devices

		72	6.3 Option Setting Add descriptions new support setting following. - Chapter 6.3.1 Module parameters
		73	7 Example of how to use the MSE Remove description T.B.D.
		73	7.1 How to send audio data - Remove Table 6.1 List of sysfs audio configuration value for PCM audio.
		73-74	7.1.1 How to send speaker test tone using MSE - Update the description of each procedure. - Add configuration list at Table 7.1 and Table 7.2.
		75-76	7.1.2 How to send PCM audio data using MSE with MCH - Update the description of each procedure. - Add configuration list at Table 7.3 and Table 7.4.
		77-79	7.1.3 How to send PCM audio and Media clock from other device - Update the description of each procedure. - Add configuration list at Table 7.5 and Table 7.6.
		80	7.2 How to send H.264 video data - Remove Table 6.2 List of sysfs video configuration value for H.264 video.
		80-81	7.2.1 How to send H.264 video file - Update the description of each procedure. - Add configuration list at Table 7.7 and Table 7.8.
		83-84	7.3 How to send MJPEG video data - Remove Table 6.3 List of sysfs video configuration value for MJPEG video. - Update the description of each procedure. - Add configuration list at Table 7.9 and Table 7.10.
		85-86	7 Example of how to use the MSE Add descriptions new use case following. - Chapter 7.4 How to send MPEG2-TS data
		87-88	8 Appendix Add Chapter 8.1 Sysfs Comparison List
0.50	Mar. 15, 2017	1	1.2.1 Standard Update of Table 1.1 Standards - Update the version of IEEE Std 1722. - Add RFC 2435, IEC 61883-4 and IEC 61883-6.
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		2	1.3 Restrictions - Remove a restriction. see 3.Functions.
		3	1.4 Terminology Update of Table 1.3 Terminology - Update the version of IEEE Std 1722. - Add PTP and gPTP.
		5	3. Function - Add notes about streaming the audio
		9	3.1 ALSA Adapter Update of Table 3.1 ALSA Adapter capabilities - Add notes about period size
		9	3.2 V4L2 Adapter Update of Table 3.4 V4L2 adapter capabilities (MPEG2-TS) - Add notes about BDAV
		11	3.5.1 Ethernet AVB Packet Format Update of Table 3.9 Video Packet format (MJPEG) - Add the description of Q-table and restart marker
		15	4.1 Sysfs Interface Update of Table 4.4 List of sysfs audio configuration parameter for MSE - Change range of media_audio_config/samples_per_frame from "0,1-740" to "0,1-

			738". - Add refer table 4.5 for describe CRF and MCH setting
		15	4.1 Sysfs Interface Add new Table 4.5 for List of available combinations of CRF and MCH
		16	4.1 Sysfs Interface Update of Table 4.6 List of sysfs video configuration parameter for MSE - Change range of media_video_config/bytes_per_frame from "0,1-1480" to "0,128-1476". - Add notes on this value.
		34	4.2.33 struct mse_media_audio_config Update of Table 4.15 struct mse_media_audio_config - Change range of samples_per_frame from "0,1-740" to "0,1-738". - Add a reference of Table 4.5.
		34	4.2.34 struct mse_media_video_config Update of Table 4.16 struct mse_media_video_config - Change range of bytes_per_frame from "0,1-1480" to "0,128-1476".
		36	4.2.37 struct mse_mch_config Update of Table 4.19 struct mse_mch_config - Add a reference of Table 4.5.
		48	4.3.20 struct mse_audio_config Update of Table 4.25 struct mse_audio_config - Change range of samples_per_frame from "0,1-740" to "0,1-738".
		49	4.3.21 struct mse_video_config Update of Table 4.26 struct mse_video_config - Change member name "n" to "numerator". - Change member name "m" to "denominator". - Change range of bytes_per_frame from "0,1-1480" to "0,128-1476".
		51	4.3.24 struct mse_adapter_network_ops - Delete member "priv"
		53-68	5. Packet Format Update the version of IEEE Std 1722 in the below. - 5.1 Ethernet Frame Format. - 5.2 AVTP IEC61883-4 MPEG2-TS Format. - 5.3 AVTP IEC61883-6 AM824 Format. - 5.4 PCM AVTP Audio Format (AAF PCM). - 5.6 H.264 Compressed Video Format (CVF H264). - 5.7 MJPEG Compressed Video Format (CVF MJPEG) - 5.8 Clock Reference Format (CRF).
		66	5.7 MJPEG Compressed Video Format (CVF MJPEG) Update of Figure 5-9 MJPEG Compressed Video Format - Fix field of "type"
		69	6.1 Directory Configuration - Delete file "mse_core.h"
		70	6.2.1 Build kernel modules only - Change build commands to adapt to MCH.
		71	6.2.2 Build MSE modules with the Linux kernel - Update examples of the Kconfig and Makefile.
		73-74	6.2.3 Integration devices - Add description regarding related modules of avb-streaming and avb-mch. - Change description of ALSA device name. - Update examples of shell commands.
		75	6.3.1 Module Parameters Update of Table 6.1 Module parameters - Change alsa_devices max number from 2 to 8.
		78-80	7.1.2 How to send PCM audio data using MSE with MCH - Change operations to omit unnecessary options for aplay. - Add shell command to execute ptp4l. - Fix value of ptp_config/capture_ch in Table 7.3, 7.4 and example commands.

			- Add settings of delay_time/tx_delay_time_ns in Table 7.3 and example commands. - Add settings of delay_time/rx_delay_time_ns in Table 7.4 and example command.
		81-83	7.1.3 How to send PCM audio and media clock from other device - Change operations to omit unnecessary options for aplay. - Fix value of ptp_config/capture_ch in Table 7.5, 7.6 and example command. - Add settings of delay_time/tx_delay_time_ns in Table 7.5 and example command. - Add settings of delay_time/rx_delay_time_ns in Table 7.6 and example command. - Fix value of avtp_tx_param_crf/dst_mac in Table 7.6 and example command. - Add shell command to execute ptp4l. - Add to set delay time in Table 7.5, 7.6 and examples of shell command.
		84-86	7.2 How to send H.264 video data Fix copy and paste problem of example of the command below. "# gst-launch-1.0 \"
		87-88	7.3 How to send MJPEG video data Fix copy and paste problem of example of the command below. "# gst-launch-1.0 \"
		89	7.4 How to send MPEG2-TS data Fix copy and paste problem of example of the command below. "# gst-launch-1.0 \"
1.00	Jul. 12, 2017	5	3. Function Add description of MCH adapter
		8	Table 3.1 ALSA Adapter capabilities Add note regarding the maximum AVTP payload size.
		8	Table 3.2 V4L2 adapter capabilities(H.264) Change the maximum Height from 3840 to 2160.
		14	Figure 4-1 overall outline procedure of time-sensitive application Add descriptions about talker delay time, max transit time and listener delay time
		39	4.3.4 mse_set_audio_config Add EBUSY and ENOSPC in Error number.
		40	4.3.6 mse_set_video_config Add EBUSY and ENOSPC in Error number.
		41	4.3.8 mse_set_mpeg2ts_config Add EBUSY and ENOSPC in Error number.
		42	4.3.9 mse_open Add EAGAIN in Error number.
		42	4.3.10 mse_close Add EBUSY and EPERM in Error number.
		43	4.3.11 mse_start_streaming Add EBUSY and EPERM in Error number.
		43	4.3.12 mse_stop_streaming Add EPERM in Error number.
		44	4.3.13 mse_start_transmission Add EBUSY, EPERM and EAGAIN in Error number.
		49	4.3.24 struct mse_adapter_network_ops Remove set_option, check_receive
		50	4.3.25 struct mch_ops Change interface of MCH
		51	4.3.26 struct mse_ptp_ops Change interface of MCH
		52	4.3 Adapter Interface Add new chapter - 4.3.27 mse_err, mse_warn, mse_info, mse_debug - 4.3.28 mse_compare_param_key, mse_name_strlcpy

		75	6.3.1 Module Parameters Change the range of major 0-4095 to 0-4094
		76-90	7. Example of how to use the MSE Change example value of MAC address to 76:90:50:00:00:00/76:90:50:00:00:01. - avtp_tx_param/src_mac - avtp_rx_param/streamid - avtp_tx_param_crf/src_mac - avtp_rx_param_crf/streamid
		78-79	7.1.2 How to send PCM audio data using MSE with MCH Remove description of ptp_config/recovery_capture_freq in Table 7.4 and step3.
		78-80	7.1.2 How to send PCM audio data using MSE with MCH Change example value of delay_time to 40000000 (40ms). - delay_time/tx_delay_time_ns - delay_time/rx_delay_time_ns
		78-80	7.1.2 How to send PCM audio data using MSE with MCH Change operations to set period time and buffer time for aplay and arecord.
		81-83	7.1.3 How to send PCM audio and media clock from other device Remove description of ptp_config/recovery_capture_freq in Table 7.6 and step3.
		81-84	7.1.3 How to send PCM audio and media clock from other device Change example value of delay_time to 40000000 (40ms). - delay_time/tx_delay_time_ns - delay_time/rx_delay_time_ns
		81-84	7.1.3 How to send PCM audio and media clock from other device Change operations to set period time and buffer time for aplay and arecord.
1.10	Nov. 14, 2017	All	Add R-Car M3N support.
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		80	7.1.2 How to send PCM audio data using MSE with MCH Add notes for using ptp4l.
		84	7.1.3 How to send PCM audio and media clock from other device Add notes for using ptp4l.
1.20	Jan. 29, 2018	-	Update Arm trademark
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		2	1.3 Restrictions Add the restriction in case of generating MP4 file with using qtmux.
		9	3.2 V4L2 Adapter Add description about using MPEG2-TS format.
		46	4.3.13 mse_start_transmission Add description about buffer_size parameter.
		52	4.3.24 struct mse_adapter_network_ops Add struct module *owner to structure.
		53	4.3.25 struct mch_ops Add struct module *owner to structure.
		54	4.3.26 struct mse_ptp_ops Add struct module *owner to structure.
		79-87	7.1 How to send audio data Change command examples for stability. - Change the way of specifying R-Car's sound device from hw:0,0 to plughw:0,0.
		88-90	7.2 How to send H.264 video data Change command examples for stability. - Add colorimetry=bt601 to caps. - Add sync=true property to sink element of Gstreamer. - Add show-preroll-frame=false property to Talker's sink element of Gstreamer.
		91-92	7.3 How to send MJPEG video data

			Change command examples for stability. - Change the file for streaming from .mjpeg file to .mov file. - Add colorimetry=bt601 to caps. - Add sync=true property to sink element of Gstreamer. - Add show-preroll-frame=false property to Talker's sink element of Gstreamer.
		93-94	7.4 How to send MPEG2-TS data Change command examples for stability. - Change the way of specifying R-Car's sound device from hw:0,0 to plughw:0,0. - Add sync=true property to sink element of Gstreamer. - Add show-preroll-frame=false property to Talker's sink element of Gstreamer.
1.30	Feb. 22, 2018	All	Add R-Car E3 support.
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		2	1.3 Restrictions Remove the restriction in case of generating MP4 file with using qtmux.
		13	Table 4.1 Device Node Change the range of Major number 240-254 to 234-254,384-511.
		78	Table 6.1 Module parameters Change the range of major 0-4094 to 0-511.
1.40	Oct. 5, 2018	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		14-17	4.1 Sysfs Interface Update of Table 4.3, 4.4, 4.6, 4.7 - Remove delay_time parameters from common parameters. - Add delay_time parameters to Audio/Video/MPEG2-TS parameters.
		14-17	4.1 Sysfs Interface Update of Table 4.4, 4.6, 4.7 Change default value of "0" for Video and MPEG2-TS. - delay_time/tx_delay_time_ns - delay_time/rx_delay_time_ns Change range from "0-4294967295" to "0-1073741824". - delay_time/tx_delay_time_ns - delay_time/rx_delay_time_ns
		36	4.2.35 struct mse_media_mpeg2ts_config Update of Table 4.17 struct mse_media_mpeg2ts_config - Add transmit_mode.
		38	4.2.38 struct mse_delay_time Update of Table 4.20 struct mse_delay_time Change default value of "0" for Video and MPEG2-TS. - tx_delay_time_ns - rx_delay_time_ns Change range from "0-4294967295" to "0-1073741824". - tx_delay_time_ns - rx_delay_time_ns
		51	4.3.22 struct mse_mpeg2ts_config Update of Table 4.27 struct mse_mpeg2ts_config - Add transmit_mode.
		76	6.2.3 Integration devices Change to always load MCH and MCH Adapter modules.
		93	7.4 How to send MPEG2-TS data Add setting "pcr" to "media_mpeg2ts_config/transmit_mode" in command example.
2.00	Dec. 25, 2018	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		94	7.4 How to send MPEG2-TS data

			Add "blocksize=36096" property to filesrc element of GStreamer.
2.10	Apr. 06, 2021	-	Update AddressList
2.20	Aug. 16, 2021	1	Add R-Car D3 support
		2	Table 1.2 Related documents - Add R-Car D3 Hardware Manual - Update revision for GStreamer User Manual
2.30	Dec 01, 2021	-	Update Notice page
		2	Table 1.2 Related documents: Change version of “R-Car Series, 3rd Generation User's Manual: Hardware” from v1.00 to v2.30
		17	Table 4.6 List of sysfs video configuration parameter for MSE - Add media_video_config/class_interval_frames - Add media_video_config/max_interval_frames Table 4.7 List of sysfs mpeg2ts configuration parameter for MSE - Add media_mpeg2ts_config/class_interval_frame - Add media_mpeg2ts_config/max_interval_frames Add note #7 and #8 for using class_interval_frames and max_interval_frames
		35	Table 4.16 struct mse_media_video_config - Add class_interval_frames and max_interval_frames
		36	Table 4.17 struct mse_media_mpeg2ts_config - Add class_interval_frames and max_interval_frames
2.31	Dec 25, 2023	All	Update version of the document.

CONFIDENTIAL

Ethernet AVB Software Media Streaming Engine
User's Manual: Software

Publication Date:	Rev.0.10	Apr. 22, 2016
	Rev.2.31	Dec. 25, 2023

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

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

Renesas Electronics America Inc. Milpitas Campus
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus
6024 Silver Creek Valley Road, San Jose, CA 95138, USA
Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, #06-02 Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：https://www.renesas.com/contact/

Colophon 3.1

Ethernet AVB Software
Media Streaming Engine

Renesas Electronics Corporation