

CONFIDENTIAL

Ethernet AVB Software Launcher Application

Application Note

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>).

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/.

CONFIDENTIAL

Trademark

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

How to Use This Manual

- **[Readers]**

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

- **[Purpose]**

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

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

It is assumed that the readers of this manual have general knowledge in the fields of electrical

— engineering, logic circuits, microcontrollers, and Linux.

→ Read this manual in the order of the CONTENTS.

— To understand the functions of a multimedia processor for R-Car H3/M3/M3N/E3

→ See the R-Car H3/M3/M3N/E3 User's Manual.

— To know the electrical specifications of the multimedia processor for R-Car H3/M3/M3N/E3

→ See the R-Car H3/M3/M3N/E3 Data Sheet.

- **[Conventions]**

The following symbols are used in this manual.

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

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... 0bxxxx, or xxxxB

Decimal ... xxxx

Hexadecimal ... 0xxxxx or xxxxH

Data type: Double word ... 64 bits

Word ... 32 bits

Half word ... 16 bits

Byte ... 8 bits

CONFIDENTIAL

Table of Contents

1. Overview	1
1.1 Overview	1
1.2 Reference	1
1.2.1 Standard.....	1
1.2.2 Related Document	2
1.3 Restrictions	2
1.4 Terminology	3
2. Operating Environment.....	4
2.1 Hardware Environment	4
2.2 Software relationship	4
3. Software	5
3.1 Function of avblauncher.....	5
3.2 Function of mrpdummy	5
3.3 Module structure	6
3.4 Configuration parameter	7
3.5 The processing outline for avblauncher	8
3.6 The processing outline for mrpdummy	10
4. Integration	15
4.1 Build process with Yocto recipe	15
4.2 Build process without Yocto recipe	15
5. Execute the applications	16
5.1 Execute avblauncher	16
5.2 Execute mrpdummy	17
5.3 Command parameters	19
6. Configuration file format of Launcher Application.....	20
6.1 INI file format	20
6.2 The function to replace the configuration parameter	21
7. How to define the AVDECC Entity Model	22
8. Support status of AVDECC	30
9. Controller application for AVDECC	33

CONFIDENTIAL

LIST OF FIGURES

Figure 2.1	Software relationship	4
Figure 3.1	Module structure	6
Figure 3.2	Outline of process (1/2).....	8
Figure 3.3	Outline of process (2/2).....	9
Figure 3.4	Outline of process for mrpdummy	10
Figure 3.5	Join VLAN	11
Figure 3.6	Register SRP domain.....	11
Figure 3.7	Listener	12
Figure 3.8	Talker.....	13
Figure 3.9	Unregister SRP domain.....	14
Figure 3.10	Leave VLAN.....	14
Figure 6.1	Format of INI file for avblauncher	20
Figure 6.2	Example of INI file for avblauncher	20
Figure 7.1	Example of YAML file for AVDECC Entity Model.....	22
Figure 9.1	The processing sequence between Controller and Talker / Listener.....	33

LIST OF TABLES

Table 1.1	Standards	1
Table 1.2	Related documents	2
Table 1.3	Terminology.....	3
Table 3.1	Configuration parameter for avblauncher	7
Table 3.2	Configuration parameter for mrpdummy	7
Table 5.1	Command parameters for avblauncher	19
Table 5.2	Command parameters for mrpdummy	19
Table 6.1	Key name and value	20
Table 7.1	List of supported descriptors	23
Table 7.2	Parameter of ENTITY descriptor.....	23
Table 7.3	Parameter of CONFIGURATION descriptor	24
Table 7.4	Parameter of AUDIO_UNIT descriptor.....	24
Table 7.5	Parameter of VIDEO_UNIT descriptor.....	25
Table 7.6	Parameter of STREAM_INPUT and STREAM_OUTPUT descriptor.....	26
Table 7.7	Parameter of JACK_INPUT and JACK_OUTPUT descriptor	26
Table 7.8	Parameter of AVB_INTERFACE descriptor	26
Table 7.9	Parameter of CLOCK_SOURCE descriptor.....	27
Table 7.10	Parameter of LOCALE descriptor	27
Table 7.11	Parameter of STRINGS descriptor	27
Table 7.12	Parameter of STREM_PORT_INPUT and STREAM_OUTPUT descriptor	28
Table 7.13	Parameter of AUDIO_CLUSTER descriptor	28
Table 7.14	Parameter of VIDEO_CLUSTER descriptor	28
Table 7.15	Parameter of AUDIO_MAP descriptor.....	29
Table 7.16	Parameter of VIDEO_MAP descriptor.....	29
Table 7.17	Parameter of CLOCK_DOMAIN descriptor	29
Table 8.1	Supported message_type of ADP	30
Table 8.2	Supported message_type of AECP	30
Table 8.3	Supported command_type of AEM_RESPONSE	30
Table 8.4	Supported message_type of ACMP	32
Table 9.1	Example of application with Controller function.....	33

1. Overview

1.1 Overview

This manual explains the Launcher application (avblauncher) and mrpdummy on R-Car H3/M3/M3N/E3/D3. avblauncher provides functions to work together the application for protocol processing and streaming processing. mrpdummy is used to reserve bandwidth using IEEE Std 802.1Qat instead of the streaming application module in OpenAvnu project.

1.2 Reference

1.2.1 Standard

The following table shows the document related to module.

Table 1.1 Standards

Number	Issue	Title	Edition	Date
IEEE Std 802.1BA-2011	IEEE STANDARDS ASSOCIATION	IEEE Standard for Local and metropolitan area networks - Audio Video Bridging (AVB) Systems	-	30 September 2011
IEEE Std 802.1Q-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 Std 802.1AS-2011	IEEE STANDARDS ASSOCIATION	IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks	-	30 March 2011
IEEE Std 1722.1-2013	IEEE STANDARDS ASSOCIATION	IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722 Based Devices	-	23 August 2013

1.2.2 Related Document

The following table shows the document related to this module.

Table 1.2 Related documents

Number	Issue	Title	Edition
1	Renesas Electronics	R-Car Series, 3rd Generation User's Manual: Hardware	Rev.2.30
2	Renesas Electronics	R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual	Rev.1.09
3	Renesas Electronics	R-CarM3-SiP System Evaluation Board Salvator-X Hardware Manual	Rev.0.04
4	Renesas Electronics	R-CarH3-SiP/M3-SiP/M3N-SiP System Evaluation Board Salvator-XS Hardware Manual	Rev.2.04
5	Renesas Electronics	R-CarE3 System Evaluation Board Ebisu-4D Hardware Manual	Rev.1.01
6	Renesas Electronics	Linux Interface Specification Yocto recipe Release Note	Rev.5.9.0
7	Renesas Electronics	Linux Interface Specification Yocto recipe Start-Up Guide	Rev.5.9.0
8	Renesas Electronics	Ethernet AVB Software AVB Streaming Driver User's Manual: Software	Rev.2.30
9	Renesas Electronics	Ethernet AVB Software Start-Up Guide	Rev.2.30
10	Renesas Electronics	R-CarD3 System Evaluation Board Draak Hardware Manual	Rev.1.20

1.3 Restrictions

Nothing

1.4 Terminology

The following table shows the terminology related to this module.

Table 1.3 Terminology

Terms	Explanation
AVB	Audio Video Bridging
AVTP	Audio Video Transport Protocol (IEEE Std 1722)
MAAP	MAC Address Acquisition Protocol (IEEE Std 1722 Annex B)
SRP	Stream Reservation Protocol (IEEE Std 802.1Q)
MAC	Media Access Control
MMRP	Multiple MAC Registration Protocol (IEEE Std 802.1Q)
MSRP	Multiple Stream Registration Protocol (IEEE Std 802.1Q)
MVRP	Multiple VLAN Registration Protocol (IEEE Std 802.1Q)
VLAN	Virtual LAN (IEEE Std 802.1Q)
PTP	Precision Time Protocol (IEEE Std 1588)
gPTP	Generalized Precision Time Protocol (IEEE Std 802.1AS)
daemon_cl	gPTP daemon
mrpd	The MRP daemon
MSE	Media Streaming Engine
Streaming Application	Applications that send AVTP frame
POSIX	Portable Operating System Interface
AVDECC	Audio/video discovery, enumeration, connection management, and control (IEEE Std 1722.1)

2. Operating Environment

2.1 Hardware Environment

The hardware environment see "chapter 1.3 hardware environment" of the Ethernet AVB Software Start-Up Guide.

2.2 Software relationship

Launcher application support the operation of the following Ethernet AVB function.

- A) **daemon_cl**
This software that has been developed in the gtp project (<https://github.com/Avnu/gtp>). This software achieves the time synchronization between devices in the network by gPTP. It outputs the time synchronization results to POSIX shared memory. And provides a function to be shared with external applications.
- B) **mrpd**
This software that has been developed in the OpenAvnu project (<https://github.com/AVnu/OpenAvnu>). This software achieves bandwidth reservation between devices in the network by SRP. mrpd uses the Unix domain socket as an interface between the client software. mrpd handles bandwidth reservation information using original command messages between client software and processes SRP.
- C) **mrpdummy**
This software is client software of mrpd. mrpdummy does the processing of bandwidth reservation instead of Streaming Application that does not have a client function of mrpd.
- D) **Streaming Application**
Streaming Application assumes the application to achieve the streaming of Audio/Video.
- E) **maap_daemon**
This software that has been developed in the OpenAvnu project (<https://github.com/AVnu/OpenAvnu>). This software provides a function to allocate dynamically the multicast MAC addresses needed by AVTP in the network. maap_daemon uses Unix domain socket as an interface. maap_daemon returns the multicast MAC address as a command response from the client software.

The following figure shows the relationship between avblauncher and other applications.

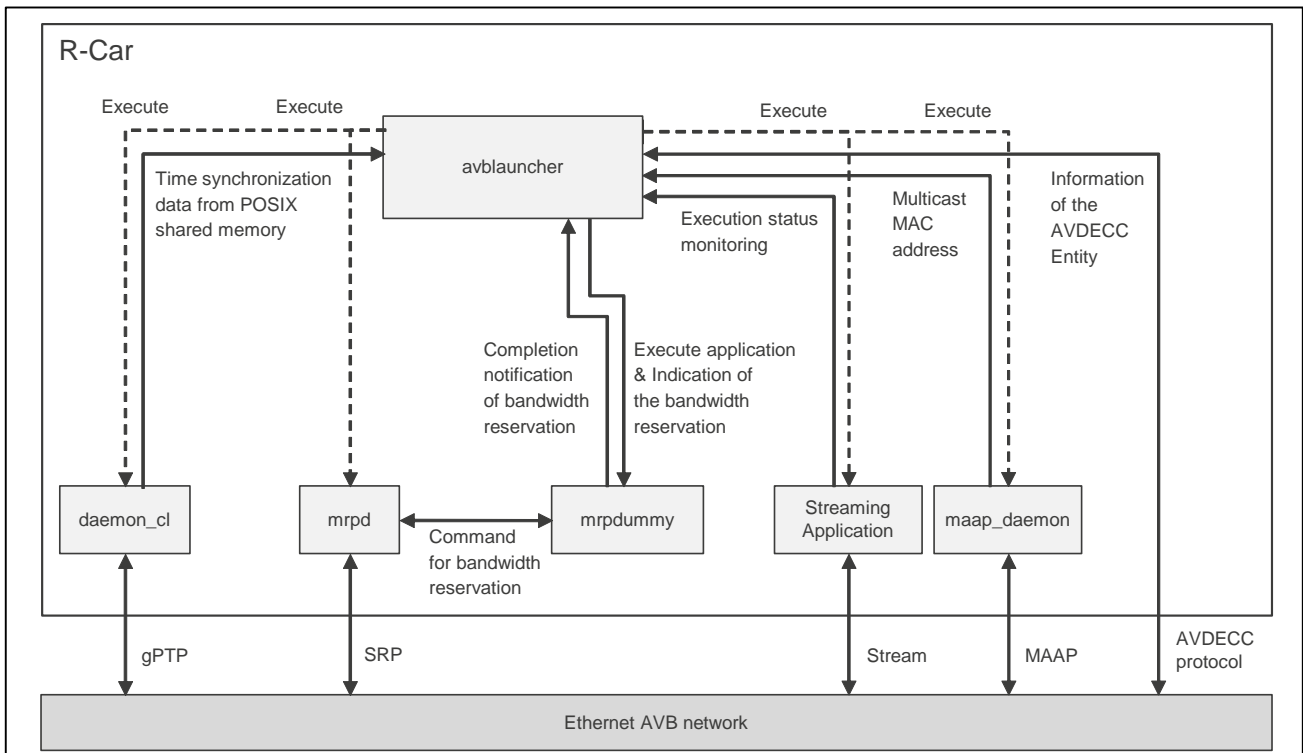


Figure 2.1 Software relationship

3. Software

3.1 Function of avblauncher

avblauncher has the following functions.

- 1 The Function of Command line option.
 - 1.1 Execute daemon_cl from avblauncher.
 - 1.2 Execute mrpd from avblauncher.
 - 1.3 Execute maap_daemon from avblauncher and initialize it.
 - 1.4 Specifying the network interface that is used for communication.
 - 1.5 Print a usage message.
- 2 The Function to load the configuration file
 - 2.1 Analyze INI files.
 - 2.2 Replace the reserved word that is in the command of application to the parameter.
- 3 The Function of AVDECC.
 - 3.1 Define itself as AVDECC Entity.
 - 3.2 Handling the AVDECC protocol data.
 - 3.3 Connection management by AVDECC.
- 4 Cooperation with other application
 - 4.1 daemon_cl
 - 4.1.1 Prevent multiple execution of daemon_cl.
 - 4.1.2 Execute daemon_cl.
 - 4.1.3 Get time synchronization information from a POSIX shared memory of daemon_cl.
 - 4.2 mrpd
 - 4.2.1 Prevent multiple execution of mrpd.
 - 4.2.2 Execute mrpd.
 - 4.3 mrpdummy
 - 4.3.1 Join/Leave VLAN.
 - 4.3.2 Join/Leave SRP domain.
 - 4.3.3 Declare/Withdraw Bandwidth reservation as Talker.
 - 4.3.4 Register/Unregister to Bandwidth reservation of Talker as Listener.
 - 4.3.5 Terminate mrpdummy. (Send a signal to mrpdummy.)
 - 4.3.6 Wait for the completion of the bandwidth reservation.
 - 4.4 Streaming Application
 - 4.4.1 Wait for ascapable before stream transmission
 - 4.4.2 Execute Streaming Application. The application command is executed via "/bin/sh".
 - 4.5 maap_daemon
 - 4.5.1 Prevent multiple execution of maap_daemon.
 - 4.5.2 Execute maap_daemon.
 - 4.5.3 Get multicast MAC address by the Unix socket interface from maap_daemon.

3.2 Function of mrpdummy

mrpdummy has the following functions.

- 1 Specify parameters for bandwidth reservation by command line.
- 2 Functions as mrpd client.
- 3 Join and leave the VLAN for AVB network.
- 4 Register and unregister SRP domain of AVB network.
- 5 Functions for talker
 - 5.1 Declare and withdraw talker advertise.
 - 5.2 Watch listeners for ready and notify ready to start streaming to talker application.
- 6 Functions for listener
 - 6.1 Watch a stream to be advertised by talker.
 - 6.2 Notify listener ready to talker.
- 7 Receive signals when the application stop streaming.

3.3 Module structure

The module structure is shown below.

```

avb-applications
|-- avb-demoapps
|   |-- avblauncher
|   |   |-- avblauncher.c           : Application top-level function for avblauncher
|   |   |-- avblauncher.h
|   |   |-- daemon_cl/             : IPC definition of daemon_cl
|   |   |-- inih/                  : Ben Hoyt's INI parser library
|   |   |-- maap_daemon/           : IPC definition of maap_daemon
|   |   |-- Makefile
|   |-- etc
|   |   |-- avblauncher             : Example ini, yaml and sh files for avblauncher
|   |   |-- avdecc_entity_audio.yaml
|   |   |-- mse_aaf_pcm_listener.ini
|   |   |-- mse_aaf_pcm.sh
|   |   |-- mse_aaf_pcm_talker.ini
|   |   |-- mse_cvf_h264_d13_listener.ini
|   |   |-- mse_cvf_h264_d13.sh
|   |   |-- mse_cvf_h264_d13_talker.ini
|   |   |-- mse_cvf_h264_listener.ini
|   |   |-- mse_cvf_h264.sh
|   |   |-- mse_cvf_h264_talker.ini
|   |   |-- mse_cvf_mjpeg_listener.ini
|   |   |-- mse_cvf_mjpeg.sh
|   |   |-- mse_cvf_mjpeg_talker.ini
|   |   |-- mse_iec61883_4_listener.ini
|   |   |-- mse_iec61883_4.sh
|   |   |-- mse_iec61883_4_talker.ini
|   |   |-- mse_iec61883_6_listener.ini
|   |   |-- mse_iec61883_6.sh
|   |   |-- mse_iec61883_6_talker.ini
|   |   |-- simple_application.sh
|   |   |-- simple_listener.ini
|   |   |-- simple_talker.ini
|   |-- daemon_cl                   : Example ini files for daemon_cl
|   |-- gptp_cfg.ini
|-- COPYING.MIT
|-- demo
|   |-- common
|   |   |-- netif_util.c           : Network interface control functions
|   |   |-- netif_util.h
|-- lib
|   |-- Makefile
|   |-- avdecc/                   : AVDECC protocol processing library
|   |   |-- jdksavdecc-c/         : IEEE 1722.1 library implemented in C
|   |   |-- msrp/                 : MSRP message handle library
|-- Makefile
|-- Makefile.include
|-- mrpdummy
|   |-- Makefile
|   |-- mrpdummy.c                 : Application top-level function for mrpdummy
|   |-- mrpdummy.h
|   |-- README

```

Figure 3.1 Module structure

3.4 Configuration parameter

The constant used in application is shown.

Table 3.1 Configuration parameter for avblauncher

Definition	Value	Contents
CMDNUM	30	The maximum number of an argument of Streaming Application.
GPTP_DAMEON_CMD	daemon_cl	The command of daemon_cl.
SRP_DAEMON_CMD	mrpd	The command of mrpd.
MAAP_DAEMON_CMD	maap_daemon	The command of maap_daemon.
MRPDUMMY_CMD	mrpdummy	The command of mrpdummy.
MAXSTR	INI_MAX_LINE	The maximum number of characters. The default value is defined in ini.h.
DAEMON_CL_SHM_NAME	/ptp	The name of the shared memory object.
MAAP_DAEMON_PORT	15364	The port number of Unix socket interface for maap_daemon.
MAAP_DAEMON_LOG	/dev/null	The name of log file for maap_daemon.
MAAP_DAEMON_START_ADDR	0x91e0f0000000	The start of address at dynamically allocate by maap_daemon.
MAAP_DAEMON_ADDR_RANGE	0xfe00	The range of address at dynamically allocate by maap_daemon.
DEBUG_LEVEL	1	The output level of the debug log.
SLEEP_TIME	100000	The time for usleep ().

Table 3.2 Configuration parameter for mrpdummy

Definition	Value	Contents
MRPDUMMY_FRAME_SIZE_DEFAULT	84	Default value of frame size
MRPDUMMY_INTERVAL_FRAME_DEFAULT	1	Default value of interval frame
STREAM_ID	0x0000000000000000	Default value of stream id
DEST_ADDR	0x91e0f0000e80	Default value of destination address
MSRP_RANK	0	Default value of msrp rank
LATENCY_TIME_MSRP	3900	Default value of latency
DEBUG_LEVEL	0	Debug log level

3.5 The processing outline for avblauncher

The following is an outline of operation in a program (main function).

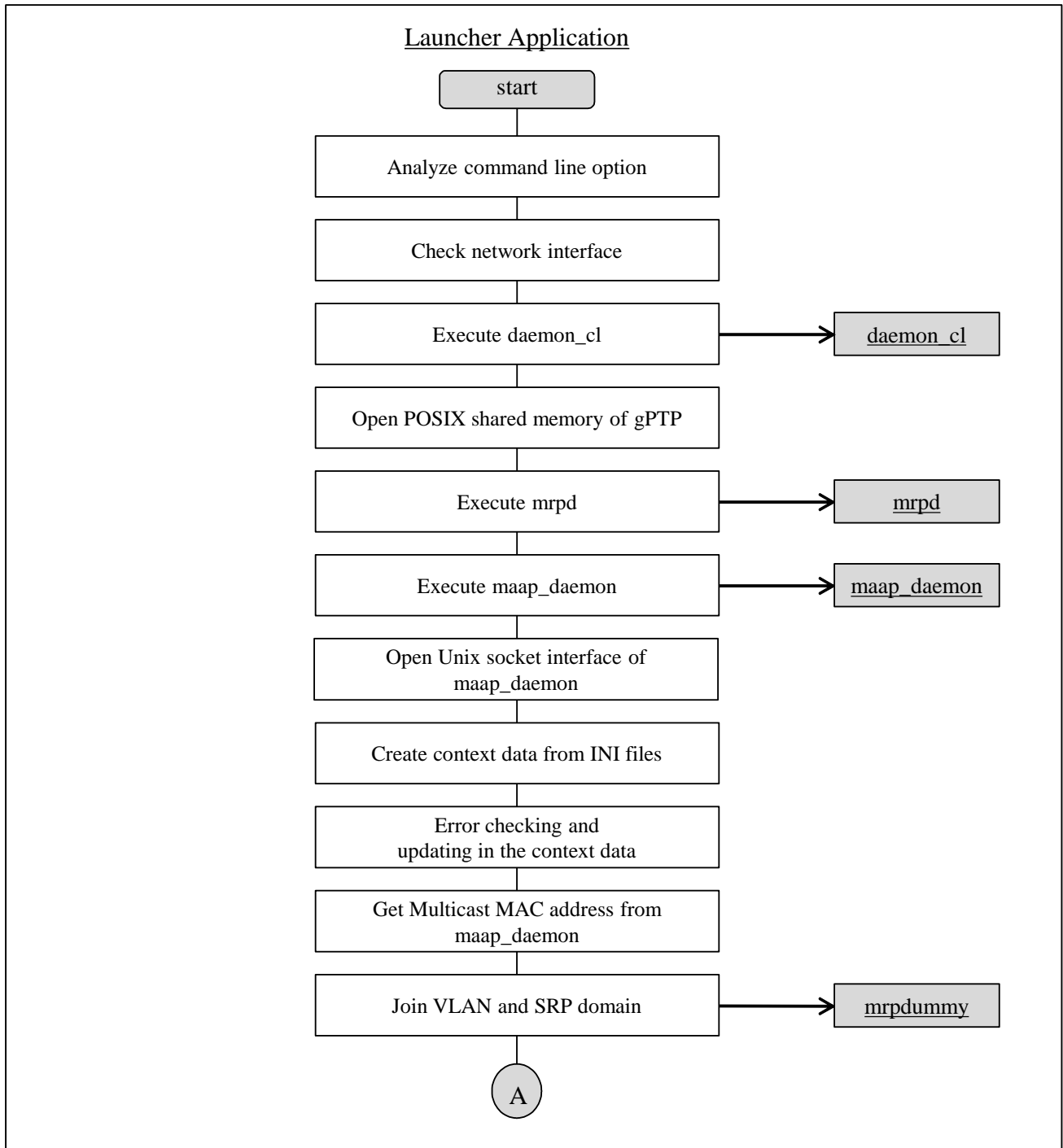


Figure 3.2 Outline of process (1/2)

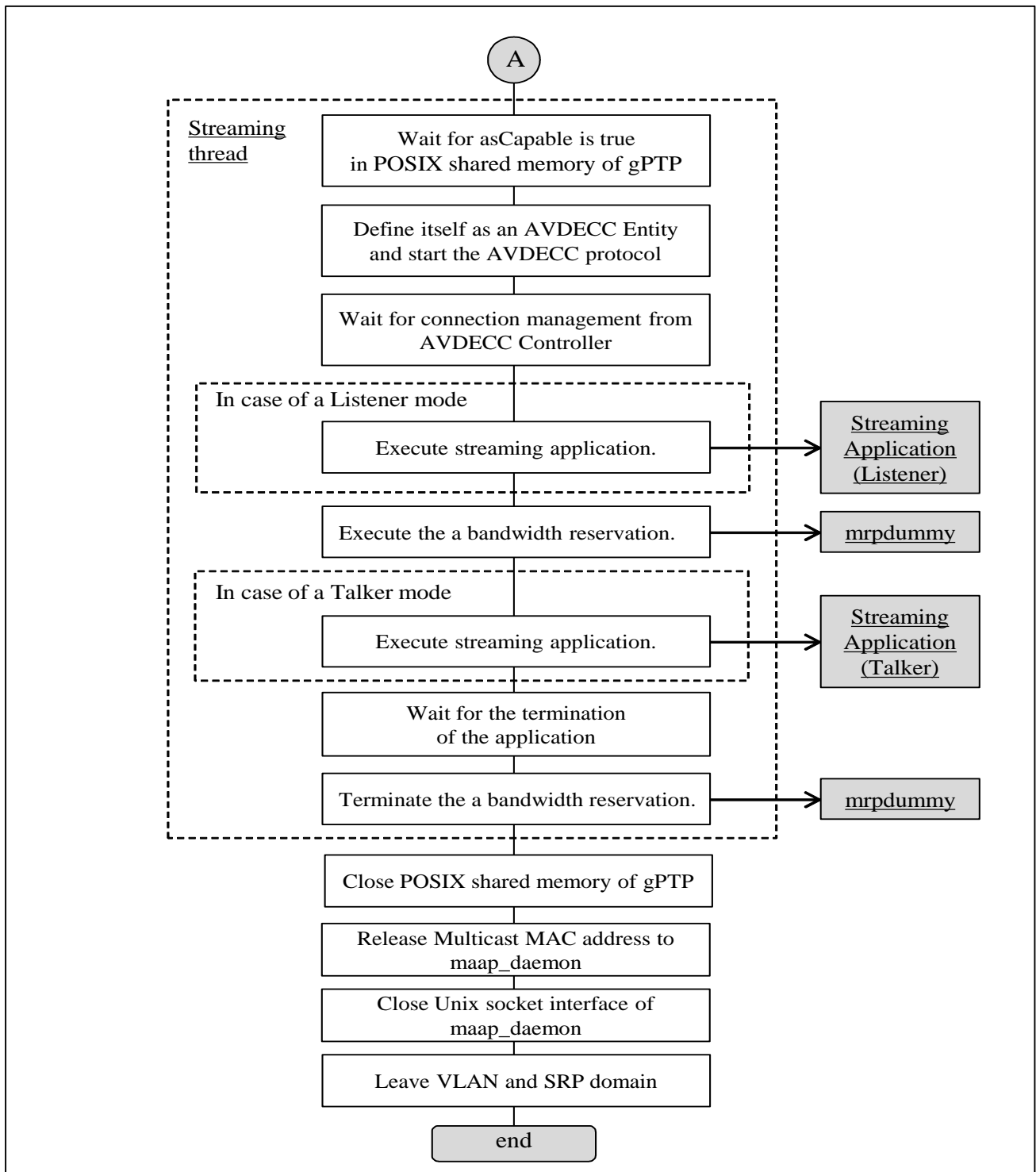


Figure 3.3 Outline of process (2/2)

3.6 The processing outline for mrpdummy

The following is an outline of operation in AVB application using mrpdummy.

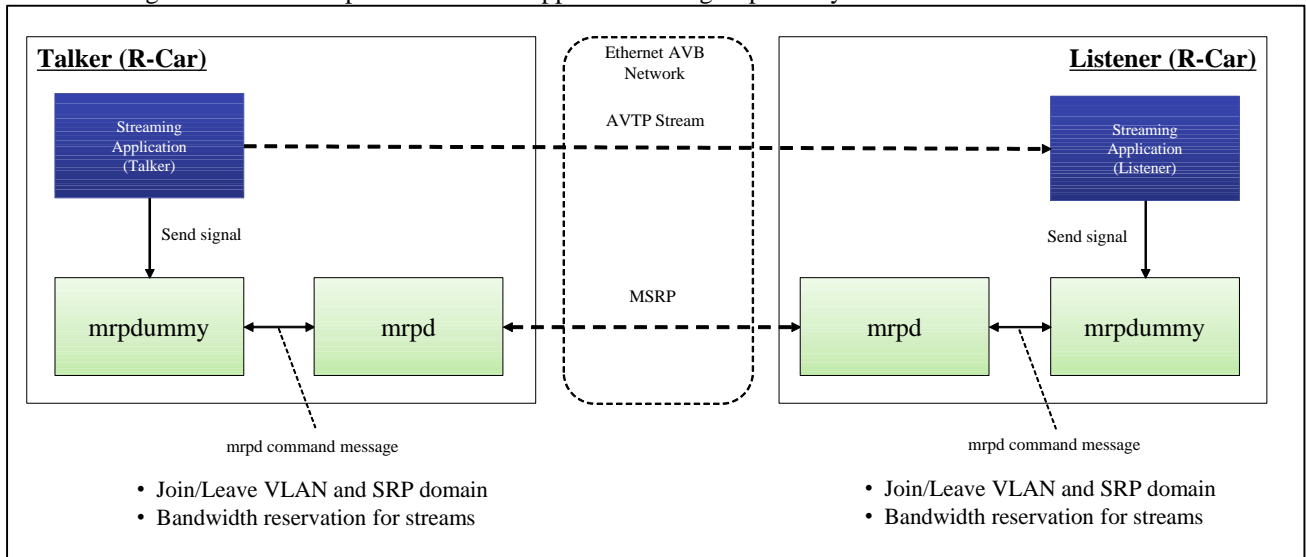


Figure 3.4 Outline of process for mrpdummy

mrpdummy change the operation mode according to the command parameter "m".

Step 1 Join VLAN

Talker and Listener applications join VLAN using the command parameter "-m 2". Specify VLAN ID using the command parameter "-V".

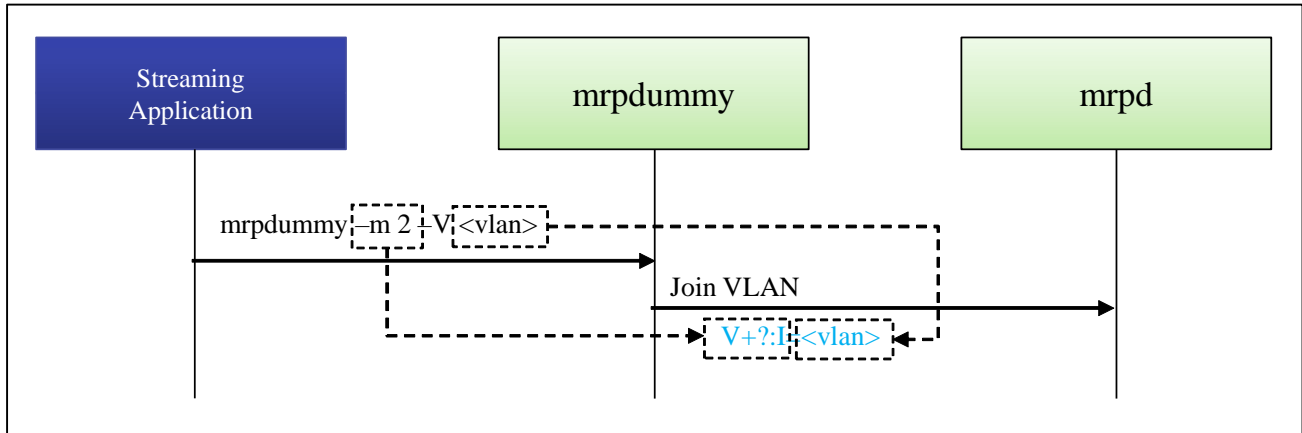


Figure 3.5 Join VLAN

Step 2 Register SRP domain

Talker and Listener applications register SRP domain using the command parameter "-m 4". Specify the SRP domain using the command parameter "-c" class id, "-p" priority and "-V" VLAN ID.

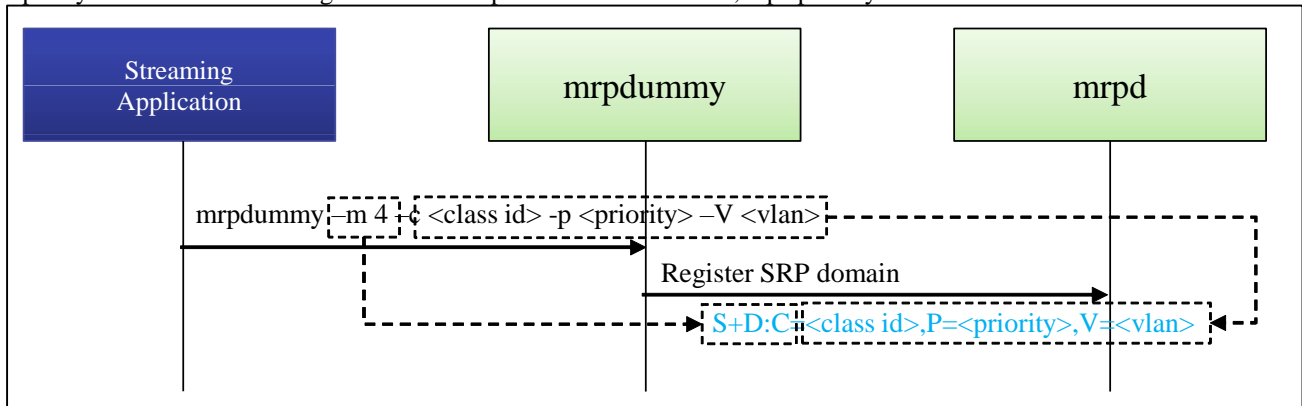


Figure 3.6 Register SRP domain

Step 3.1 Listener

Listener applications reply ready using the command parameter "-m 1".

Specify stream id using the command parameter "-S".

Listener application send signal when stop streaming.

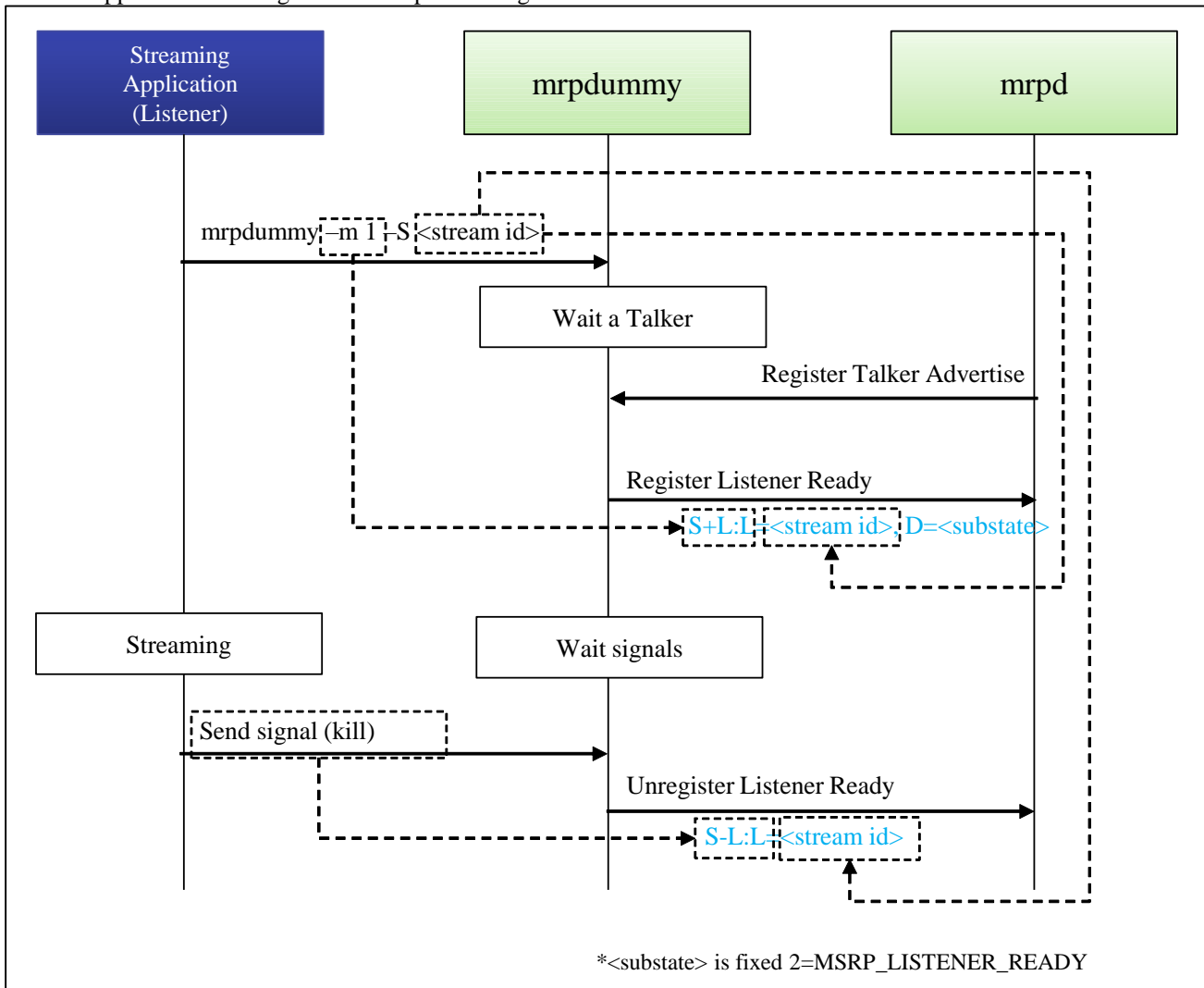


Figure 3.7 Listener

Step 3.2 Talker

Talker application advertise a stream using the command parameter "-m 0".
Specify stream id using the command parameter "-S", and see 5.3 about other options.
Talker application wait streaming until mrpdummy send "reservation completion".
Talker application send signal when stop streaming.

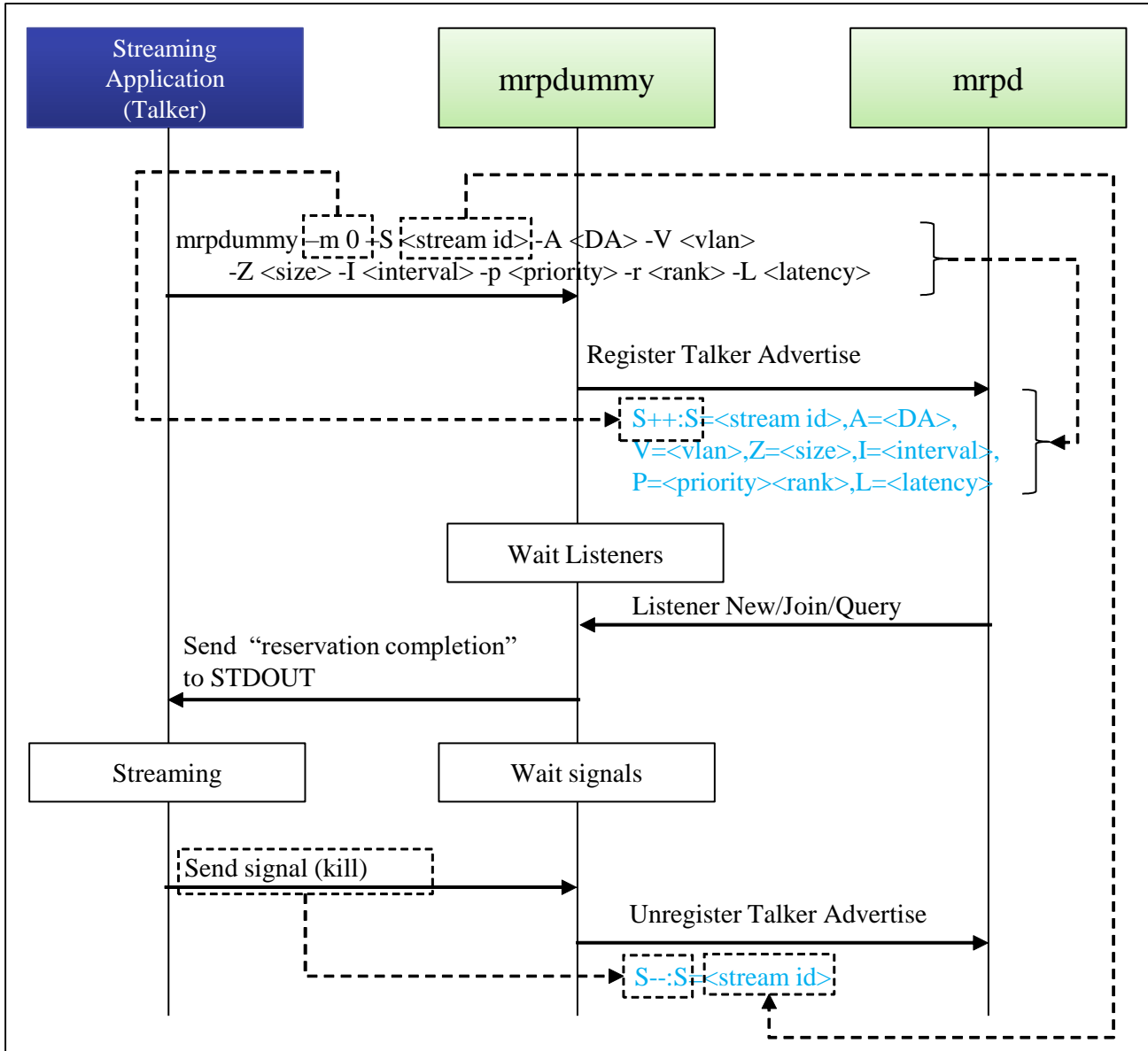


Figure 3.8 Talker

Step 4 Unregister SRP domain

Talker and Listener applications unregister SRP domain using the command parameter "-m 5".
Specify the SRP domain using the command parameter "-c" class id, "-p" priority and "-V" VLAN ID.

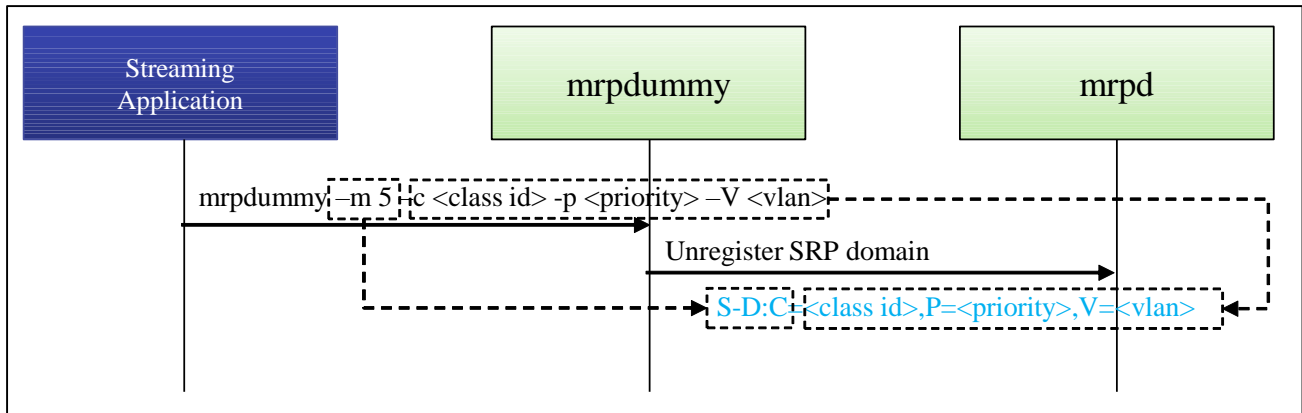


Figure 3.9 Unregister SRP domain

Step 5 Leave VLAN

Talker and Listener applications leave VLAN using the command parameter "-m 3".
Specify VLAN ID using the command parameter "-V".

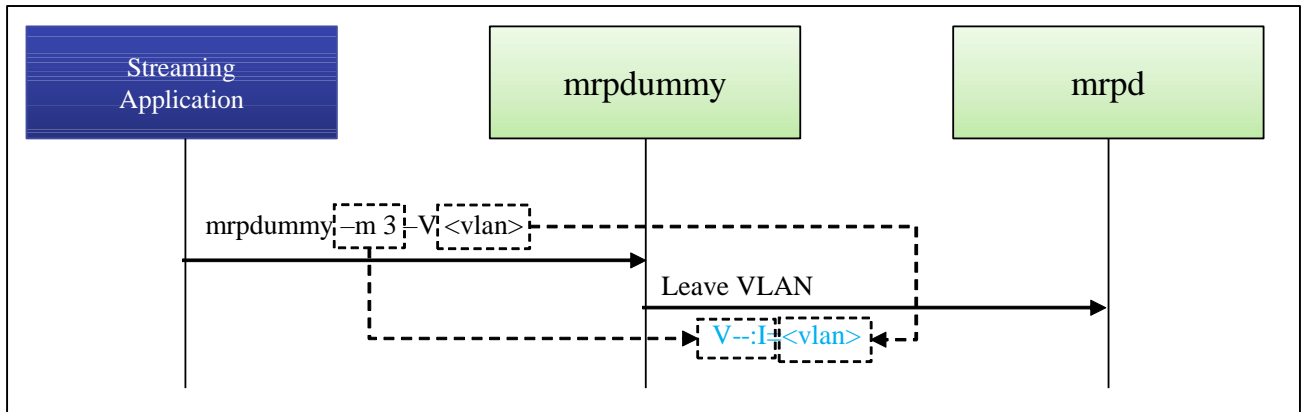


Figure 3.10 Leave VLAN

4. Integration

4.1 Build process with Yocto recipe

Please refer to "Chapter 3 Build the kernel and rootfs" in Ethernet AVB Software Start-Up Guide.

4.2 Build process without Yocto recipe

If you want to build application without bitbake, please run these steps in this chapter.

Step 1 Set environment variables

```
# export TOP_DIR=${WORK}
# source /opt/poky/${YOCTOVERSION}/environment-setup-aarch64-poky-linux
```

Step 2 Clone source code

```
# git clone git://github.com/renesas-rcar/avb-applications.git -b rcar-gen3
# cd avb-applications/avb-demoapps
```

Step 3 Execute make

Please execute "make".

```
# make CC="$CC" -C lib/msrp
# make CC="$CC" -C lib/avdecc
# make CC="$CC" TOP_DIR=$(pwd) -C mrpdummy
# make CC="$CC" TOP_DIR=$(pwd) -C avblauncher
```

Step 4 Install to the target file system

Please install executables to your Root File system.

```
# cp -r etc/daemon_cl /export/rfs/etc/
# cp -r etc/avblauncher /export/rfs/etc/

# make install INSTALL_DIR=/export/rfs/usr/bin
```

Step 5 Prepare controller application for AVDECC

This application does not have the controller function of AVDECC. When using AVDECC, it is necessary to prepare a controller application. For details of applications with controller functions, please refer to chapter.9.

5. Execute the applications

This chapter shows an example about how to execute Launcher application and mrpdummy. Please start Linux on the target board and execute the following commands.

5.1 Execute avblauncher

Step 1 Describe the AVDECC Entity model data to the YAML file. (When AVDECC processing is enabled)

When executing AVDECC processing with Launcher application, it is necessary to define AVDECC Entity Model data. For details, see Chapter 7.

Step 2 Describe the setting to the INI file.

Launcher application will execute the streaming application and protocol processing application using the information in the configuration file. Refer to Chapter.6 and prepare the INI file.

Step 3 Execute the application

Please execute "avblauncher". Refer to Chapter.5.3 for details of option parameter. Non-option argument is handled as an INI file. It is also possible to specify multiple INI files.

```
# avblauncher -g [configuration file] -a -m -i [network I/F] [INI file] ...
```

Execution example:

```
# avblauncher -g /etc/daemon_cl/gptp_cfg.ini -a -m -i eth0 sample.ini
```

Step 4 Execute the connection management of AVDECC (When AVDECC processing is enabled)

Send the AECp message of AVDECC from the AVDECC Controller and perform connection management.

An example of execution when "avdecccmdline" is used as a controller application of AVDECC is shown below.

In the following example, avdecccmdline expects execution from Host PC.

(For details of "avdecccmdline", refer to Chapter.9)

In the following command example, the EntityID of the Talker side is "xx:xx:xx:xx:xx:xx:xx:xx" and the EntityID of the Listener side is "yy:yy:yy:yy:yy:yy:yy:yy ". It is a case to connect to the Talker's StreamOutput descriptor(descriptor_index=0) and Listener's Stream Input descriptor(descriptor_index=0).

Information such as Entity ID and descriptor_index depends on the AVDECC Entity Model of each device.

Execution example:

```
$ avdecccmdline
AVDECC Controller version: vX.Y.Z
1 (lo, address: <127.0.0.1>)
2 (eth0, address: <192.168.59.127>)
Enter the interface number (1-2): 2

$ list
End Station | Name | Entity ID | Firmware Version | MAC
-----
C 0 | R-Car Talker | 0xXXXXXXXXXXXXXXXXX | vX.X.0 | (Talker's MAC Address)
C 1 | R-Car Listener | 0xYYYYYYYYYYYYYYYYY | vX.X.0 | (Listener's MAC Address)

$ connect 0xXXXXXXXXXXXXXXXXX 0 0xYYYYYYYYYYYYYYYYY 0
```

If Talker and Listener are successfully connected, avblauncher start the streaming processing.

5.2 Execute mrpdummy

This chapter is the procedure of bandwidth reservation using mrpd and mrpdummy. Depending on the configuration of the INI files that specified in avblauncher, mrpd and mrpdummy is executed automatically. If you want to run the bandwidth reservation processing separately from the processing of avblauncher, please do the bandwidth reservation in the following procedure.

Step 1 Execute "mrpd"

Please execute "mrpd".

```
# mrpd -dlmvs -i [interface]
```

Execution example:

```
# mrpd -dlmvs -i eth0
```

Step 2 Join VLAN

Please execute "mrpdummy" on talker and listener boards.

```
# mrpdummy -m 2 -V [vlan]
```

Execution example:

```
# mrpdummy -m 2 -V 2
```

Step 3 Register SRP domain

Please execute "mrpdummy" on talker and listener boards.

```
# mrpdummy -m 4 -c [class id] -p [priority] -V [vlan]
```

Execution example:

```
# mrpdummy -m 4 -c 6 -p 3 -V 2
```

Step 4.1 Execute Listener application

Please execute "mrpdummy" on listener board.

```
# mrpdummy -m 1 -S [stream id]
```

Execution example:

```
# mrpdummy -m 1 -S 7690500000000001
```

<wait the talker executing Step4.2 below>

await talker.
reservation completion

Step 4.2 Execute Talker application

Please execute "mrpdummy" on talker board.

```
# mripdummy -m 0 -S [stream id] -A [DA] -V [vlan] -Z [size] -I [interval] -p [priority] -r [rank] -L [latency]
```

Execution example:

```
# mripdummy -m 0 -S 7690500000000001 -A 91e0f0000e80 -V 2 -Z 84 -I 1 -p 3 -r 0 -L 3900  
  
<wait the listener executing Step4.1>  
  
await listener.  
reservation completion
```

Step 5 Stop application

Send signal by type ctrl-c.

Unregister SRP domain and leave VLAN by using following commands.

```
# [type CTRL-C]  
# mripdummy -m 5 -c [class id] -p [priority] -V [vlan]  
# mripdummy -m 3 -V [VLAN_ID]
```

Execution example:

```
# [type CTRL-C]  
# mripdummy -m 5 -c 6 -p 3 -V 2  
# mripdummy -m 3 -V 2
```


5.3 Command parameters

The command to be used is shown below when executing the application.

Table 5.1 Command parameters for avblauncher

short option	Default value	Contents
-g	gptp_cfg.ini	Execute daemon_cl from Launcher application. Specify configuration file for daemon_cl.
-m	-	Execute mrpd from Launcher application.
-a	-	Execute maap_daemon from Launcher application, and connection by Unix socket interface.
-i	eth0	Specify Network interface name.
-h	-	Print a usage message.

Table 5.2 Command parameters for mrpdummy

short option	Default value	Contents
-m	0	Specify operation mode. 0: run talker mode 1: run listener mode 2: join vlan 3: leave vlan 4: register SRP domain 5: unregister SRP domain
-S	0000000000000000	Specify Stream ID.
-A	91e0f0000e80	Specify Destination Address.
-V	2	Specify VLAN ID.
-Z	84	Specify max frame size.
-I	1	Specified number of frame in interval of SR class.
-c	6	Specify SR class ID.
-p	3	Specify SR class priority.
-r	0	Specify rank.
-L	3900	Specify latency.
-v	-	Set verbose mode.
-h	-	Show help message.

6. Configuration file format of Launcher Application

avblauncher will execute the streaming application and protocol processing application from the information in the configuration file. The format of the configuration file use the INI file format. This chapter describes the format and Key of the INI file. And this chapter also describes the function to replace the command of streaming applications that are described in the INI file to the Key value

6.1 INI file format

The section name appears on a line by itself, in square brackets ([and]). All keys after the section declaration are associated with that section. There is no explicit "end of section" delimiter; sections end at the next section declaration, or the end of the file. Sections may not be nested. The section name must be specified in this launcher application. The Section name is declared to INI_SECTION_NAME (See Table 3.1). Every key has a name and a value, delimited by an equals sign (=). The name appears to the left of the equals sign. The name and the value of the key that can be handled in avblauncher are listed on the Table 6.1. Semicolons (;) at the beginning of the line indicate a comment. Comment lines are ignored.

Format:

[The section name] name=value ; comment text
--

Figure 6.1 Format of INI file for avblauncher

Example:

<pre>[avblauncher] MODE=talker GPTP=enable SRP=enable AVDECC=/etc/avblauncher/avdecc_entity_audio.yaml SR_CLASS=6 SR_PRIORITY=3 VLAN_ID=2 UNIQUE_ID=1 DEST_ADDR=91:e0:f0:00:0e:01 MAX_FRAME_SIZE=124 FRAME_INTERVALS=1 LATENCY=3900 STREAM_ID=76:90:50:00:00:00:00:01 APPLICATION=simple_talker -f /tmp/data.bin -m 0 -u \$UNIQUE_ID\$ -i eth0 -D \$DEST_ADDR\$</pre>

Figure 6.2 Example of INI file for avblauncher

Table 6.1 Key name and value

No	Key Name	Key Value and Range	Description	mandatory	
				Talker mode	Listener mode
1	MODE	talker / listener	To specify the mode of operation		
2	GPTP	enable / disable	To enable waiting for asCapable before streaming.	v	v
3	SRP	enable / disable	To enable bandwidth reservation processing.	v	v

4	AVDECC	[yaml file path] / disable	To enable the AVDECC protocol handling process.	v	v
5	SR_CLASS	0-7	SRclassID	v	v
6	SR_PRIORITY	0-7	SRclassPriority	v	v
7	VLAN_ID	1-4094	VLAN ID	v	v
8	UNIQUE_ID	1-65535	Unique ID in StreamID	v	
9	DEST_ADDR	00:00:00:00:00:00-FF:FF:FF:FF:FF:FF / MAAP	Stream destination MAC address When MAAP is selected, destination MAC address is generated by maap_daemon.	v	
10	MAX_FRAME_SIZE	46-1500	Maximum frame size sent by Talker	v	
11	FRAME_INTERVALS	1-65535	Maximum number of frames sent per class measurement interval	v	
12	LATENCY	0-4294967295	Accumulated Latency	v	
13	STREAM_ID	00:00:00:00:00:00:00:00-FF:FF:FF:FF:FF:FF:FF:FF	Stream ID of the receiving target for Listener		v
14	APPLICATION	e.g.) /app.sh	Application command	v	v

6.2 The function to replace the configuration parameter

This function replaces the reserved word that was found by analyzing the string that is specified in the "APPLICATION" in the INI file to the value. The replacement process is based on the following operations.

- "\$ (dollar sign)" is delimiter.
- Handling the character string enclosed in "\$ (dollar sign)" as the configuration parameter.
- If the string ("\$(dollar sign)" is not included) matches the Key name in Table 6.1, to replace the configuration parameter ("\$(dollar sign)" is also included) to the Key value.
Key value of the string is converted to lowercase.
- If the "\ (backslash)" exists, the next one character handled as character. (Escape processing for describing the delimiter)

An example is shown below. The configuration parameter is "UNIQUE_ID". This word is replaced to the Key value, and command will be processed in Launcher application.

e.g.)

[Before the replacement]

```
APPLICATION=simple_talker -i eth0 -f /file -u $UNIQUE_ID$
```

[After the replacement]

```
APPLICATION=simple_talker -i eth0 -f /file -u 1
```

The command is executed via "/bin/sh". Please note that parameter beginning with "\$" that is not converted will be handled as variable of the shell.

7. How to define the AVDECC Entity Model

The AVDECC Entity Model (AEM) describes the internal structure of the AVDECC Entity as a hierarchy of objects, with each object providing information about itself, its children and where it is in the hierarchy. To define the AVDECC Entity Model requires information of AEM descriptor. The avblauncher reads the information of AEM descriptor that was described in the YAML file, to define itself as the AVDECC Entity Model.

In this chapter, describes how to define "Entity" and "Configuration" of the AEM descriptor as an example.

For more information about the AVDECC Entity Model and the descriptor, please refer to Chapter 7 of the IEEE1722.1 standard. A list of descriptors supported by this program is shown in Table 7.1.

How to write AEM data

1. Define AEM data starting from the hash of the entity descriptor.
2. Describe the entity descriptor parameters in one array.
3. Describe the one configuration descriptor parameters in one array in the "configurations" hash.
4. Describe the other descriptor parameters in one array in the "descriptor" hash.
5. From the "descriptors" hash as a starting point, a hash of another descriptor is defined, and the parameters of one descriptor are defined in one array.
6. If the descriptor parameter is variable-length data, describe the data string directly as a hexadecimal number.
7. Define another descriptor with a hash and describe it the same way.

```

entity:                                     # 1
- entity_id: 2E:09:0A:FF:FF:00:83:02       # 2
  entity_model_id: 00:22:97:00:00:00:12:34
  entity_capabilities: 0x450b
  talker_capabilities: 0x4801
  listener_capabilities: 0x4001
  controller_capabilities: 0x00000000
  available_index: 0
  association_id: 00:00:00:00:00:00:00:00
  entity_name: R-Car 1in/1out
  vendor_name_string: 0
  model_name_string: 1
  firmware_version: v2.5.0+dirty
  group_name: R-Car AVB
  serial_number: 12345678
  current_configuration: 0
  configurations:                           # 3
    - object_name: Configuration 0
      localized_description: 0xFFFF
      descriptors:                           # 4
        audio_unit:                          # 5
          - object_name: Audio Unit 0
            localized_description: 0xFFFF
            :
            sampling_rates_count: 1           # 6
            sampling_rates: 0000bb80
          stream_input:                       # 7
            - object_name: Input 0
              localized_description: 0xFFFF
              :

```

Figure 7.1 Example of YAML file for AVDECC Entity Model

Table 7.1 List of supported descriptors

No	Descriptor name	Supported	Description
1.	ENTITY	v	
2.	CONFIGURATION	v	
3.	AUDIO_UNIT	v	
4.	VIDEO_UNIT	v	
5.	SENSOR_UNIT		
6.	STREAM_INPUT	v	
7.	STREAM_OUTPUT	v	
8.	JACK_INPUT	v	
9.	JACK_OUTPUT	v	
10.	AVB_INTERFACE	v	
11.	CLOCK_SOURCE	v	
12.	MEMORY_OBJECT		
13.	LOCALE	v	
14.	STRINGS	v	
15.	STREAM_PORT_INPUT	v	
16.	STREAM_PORT_OUTPUT	v	
17.	EXTERNAL_PORT_INPUT		
18.	EXTERNAL_PORT_OUTPUT		
19.	INTERNAL_PORT_INPUT		
20.	INTERNAL_PORT_OUTPUT		
21.	AUDIO_CLUSTER	v	
22.	VIDEO_CLUSTER	v	
23.	SENSOR_CLUSTER		
24.	AUDIO_MAP	v	
25.	VIDEO_MAP	v	
26.	SENSOR_MAP		
27.	CONTROL		
28.	SIGNAL_SELECTOR		
29.	MIXER		
30.	MATRIX		
31.	MATRIX_SIGNAL		
32.	SIGNAL_SPLITTER		
33.	SIGNAL_COMBINER		
34.	SIGNAL_DEMULTIPLEXER		
35.	SIGNAL_MULTIPLEXER		
36.	SIGNAL_TRANSCODER		
37.	CLOCK_DOMAIN	v	

Details of the parameters of descriptor supported in Table 7.1 are described following tables.

Parameters that are set to Auto are determined by specific values or other parameters.

In the case of Manual parameters, it is necessary to describe it in the YAML file when defining the descriptor.

Table 7.2 Parameter of ENTITY descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x0
2.	descriptor_index	Auto	Fixed value of 0x0
3.	entity_id	Manual	
4.	entity_model_id	Manual	
5.	entity_capabilities	Manual	
6.	talker_stream_sources	Auto	Number of STREAM_OUTPUT of current_configuration
7.	talker_capabilities	Manual	
8.	listener_stream_sinks	Auto	Number of STREAM_INPUT of current_configuration

9.	listener_capabilities	Manual	
10.	controller_capabilities	Manual	
11.	available_index	Manual	
12.	association_id	Manual	
13.	entity_name	Manual	
14.	vendor_name_string	Manual	
15.	model_name_string	Manual	
16.	firmware_version	Manual	
17.	group_name	Manual	
18.	serial_number	Manual	
19.	configurations_count	Auto	Number of definition of CONFIGURATION descriptor
20.	current_configuration	Manual	

Table 7.3 Parameter of CONFIGURATION descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x1
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	descriptor_counts_count	Auto	Number of descriptor definitions related to configuration
6.	descriptor_counts_offset	Auto	Fixed value of 108
7.	descriptor_counts	Auto	It is determined by the definition of the descriptor under the Configuration descriptor.

Table 7.4 Parameter of AUDIO_UNIT descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x2
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	clock_domain_index	Manual	
6.	number_of_stream_input_ports	Manual	
7.	base_stream_input_port	Manual	
8.	number_of_stream_output_ports	Manual	
9.	base_stream_output_port	Manual	
10.	number_of_external_input_ports	Manual	
11.	base_external_input_port	Manual	
12.	number_of_external_output_ports	Manual	
13.	base_external_output_port	Manual	
14.	number_of_internal_input_ports	Manual	
15.	base_internal_input_port	Manual	
16.	number_of_internal_output_ports	Manual	
17.	base_internal_output_port	Manual	
18.	number_of_controls	Manual	
19.	base_control	Manual	
20.	number_of_signal_selectors	Manual	
21.	base_signal_selector	Manual	
22.	number_of_mixers	Manual	
23.	base_mixer	Manual	
24.	number_of_matrices	Manual	
25.	base_matrix	Manual	
26.	number_of_splitters	Manual	
27.	base_splitter	Manual	
28.	number_of_combiners	Manual	

29.	base_combiner	Manual	
30.	number_of_demultiplexers	Manual	
31.	base_demultiplexer	Manual	
32.	number_of_multiplexers	Manual	
33.	base_multiplexer	Manual	
34.	number_of_transcoders	Manual	
35.	base_transcoder	Manual	
36.	number_of_control_blocks	Manual	
37.	base_control_block	Manual	
38.	current_sampling_rate	Manual	
39.	sampling_rates_offset	Auto	Fixed value of 144
40.	sampling_rates_count	Manual	
41.	sampling_rates	Manual	

Table 7.5 Parameter of VIDEO_UNIT descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x3
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	clock_domain_index	Manual	
6.	number_of_stream_input_ports	Manual	
7.	base_stream_input_port	Manual	
8.	number_of_stream_output_ports	Manual	
9.	base_stream_output_port	Manual	
10.	number_of_external_input_ports	Manual	
11.	base_external_input_port	Manual	
12.	number_of_external_output_ports	Manual	
13.	base_external_output_port	Manual	
14.	number_of_internal_input_ports	Manual	
15.	base_internal_input_port	Manual	
16.	number_of_internal_output_ports	Manual	
17.	base_internal_output_port	Manual	
18.	number_of_controls	Manual	
19.	base_control	Manual	
20.	number_of_signal_selectors	Manual	
21.	base_signal_selector	Manual	
22.	number_of_mixers	Manual	
23.	base_mixer	Manual	
24.	number_of_matrices	Manual	
25.	base_matrix	Manual	
26.	number_of_splitters	Manual	
27.	base_splitter	Manual	
28.	number_of_combiners	Manual	
29.	base_combiner	Manual	
30.	number_of_demultiplexers	Manual	
31.	base_demultiplexer	Manual	
32.	number_of_multiplexers	Manual	
33.	base_multiplexer	Manual	
34.	number_of_transcoders	Manual	
35.	base_transcoder	Manual	
36.	number_of_control_blocks	Manual	
37.	base_control_block	Manual	

Table 7.6 Parameter of STREAM_INPUT and STREAM_OUTPUT descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	STREAM_INPUT: Fixed value of 0x5 STREAM_OUTPUT: Fixed value of 0x6
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	clock_domain_index	Manual	
6.	stream_flags	Manual	
7.	current_format	Manual	
8.	formats_offset	Auto	Fixed value of 132
9.	number_of_formats	Manual	
10.	backup_talker_entity_id_0	Manual	
11.	backup_talker_unique_id_0	Manual	
12.	backup_talker_entity_id_1	Manual	
13.	backup_talker_unique_id_1	Manual	
14.	backup_talker_entity_id_2	Manual	
15.	backup_talker_unique_id_2	Manual	
16.	backedup_talker_entity_id	Manual	
17.	backedup_talker_unique_id	Manual	
18.	avb_interface_index	Manual	
19.	buffer_length	Manual	
20.	formats	Manual	

Table 7.7 Parameter of JACK_INPUT and JACK_OUTPUT descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	JACK_INPUT: Fixed value of 0x7 JACK_OUTPUT: Fixed value of 0x8
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	jack_flags	Manual	
6.	jack_type	Manual	
7.	number_of_controls	Manual	
8.	base_control	Manual	

Table 7.8 Parameter of AVB_INTERFACE descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x9
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	mac_address	Manual	
6.	interface_flags	Manual	
7.	clock_identity	Manual	
8.	priority1	Manual	
9.	clock_class	Manual	
10.	offset_scaled_log_variance	Manual	
11.	clock_accuracy	Manual	
12.	priority2	Manual	
13.	domain_number	Manual	
14.	log_sync_interval	Manual	

15.	log_announce_interval	Manual	
16.	log_pdelay_interval	Manual	
17.	port_number	Manual	

Table 7.9 Parameter of CLOCK_SOURCE descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0xA
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	clock_source_flags	Manual	
6.	clock_source_type	Manual	
7.	clock_source_identifier	Manual	
8.	clock_source_location_type	Manual	
9.	clock_source_location_index	Manual	
10.	descriptor_type	Manual	
11.	descriptor_index	Manual	
12.	object_name	Manual	
13.	localized_description	Manual	
14.	clock_source_flags	Manual	
15.	clock_source_type	Manual	
16.	clock_source_identifier	Manual	
17.	clock_source_location_type	Manual	

Table 7.10 Parameter of LOCALE descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0xC
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	locale_identifier	Manual	
4.	number_of_strings	Auto	Number of STRING descriptors
5.	base_strings	Manual	

Table 7.11 Parameter of STRINGS descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0xD
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	string_0	Manual	
4.	string_1	Manual	
5.	string_2	Manual	
6.	string_3	Manual	
7.	string_4	Manual	
8.	string_5	Manual	
9.	string_6	Manual	

Table 7.12 Parameter of STREAM_PORT_INPUT and STREAM_OUTPUT descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	STREAM_PORT_INPUT: Fixed value of 0xE STREAM_PORT_OUTPUT: Fixed value of 0xF
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	clock_domain_index	Manual	
4.	port_flags	Manual	
5.	number_of_controls	Manual	
6.	base_control	Manual	
7.	number_of_clusters	Manual	
8.	base_cluster	Manual	
9.	number_of_maps	Manual	
10.	base_map	Manual	

Table 7.13 Parameter of AUDIO_CLUSTER descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x14
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	signal_type	Manual	
6.	signal_index	Manual	
7.	signal_output	Manual	
8.	path_latency	Manual	
9.	block_latency	Manual	
10.	channel_count	Manual	
11.	format	Manual	

Table 7.14 Parameter of VIDEO_CLUSTER descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x15
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	signal_type	Manual	
6.	signal_index	Manual	
7.	signal_output	Manual	
8.	path_latency	Manual	
9.	block_latency	Manual	
10.	format	Manual	
11.	current_format_specific	Manual	
12.	supported_format_specifics_offset	Manual	
13.	supported_format_specifics_count	Manual	
14.	current_sampling_rate	Manual	
15.	supported_sampling_rates_offset	Manual	
16.	supported_sampling_rates_count	Manual	
17.	current_aspect_ratio	Manual	
18.	supported_aspect_ratios_offset	Manual	
19.	supported_aspect_ratios_count	Manual	
20.	current_size	Manual	
21.	supported_sizes_offset	Manual	
22.	supported_sizes_count	Manual	

23.	current_color_space	Manual	
24.	supported_color_spaces_offset	Manual	
25.	supported_color_spaces_count	Manual	
26.	supported_format_specifics	Manual	
27.	supported_sampling_rates	Manual	
28.	supported_aspect_ratios	Manual	
29.	supported_sizes	Manual	
30.	supported_color_spaces	Manual	

Table 7.15 Parameter of AUDIO_MAP descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x17
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	mappings_offset	Manual	
4.	number_of_mappings	Manual	
5.	mappings	Manual	
6.	descriptor_type	Manual	
7.	descriptor_index	Manual	
8.	mappings_offset	Manual	
9.	number_of_mappings	Manual	
10.	mappings	Manual	

Table 7.16 Parameter of VIDEO_MAP descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x18
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	mappings_offset	Manual	
4.	number_of_mappings	Manual	
5.	mappings	Manual	
6.	descriptor_type	Manual	
7.	descriptor_index	Manual	
8.	mappings_offset	Manual	
9.	number_of_mappings	Manual	
10.	mappings	Manual	

Table 7.17 Parameter of CLOCK_DOMAIN descriptor

No	Parameter name	Type	Description
1.	descriptor_type	Auto	Fixed value of 0x24
2.	descriptor_index	Auto	It is set from 0 in order of definition of the same descriptor.
3.	object_name	Manual	
4.	localized_description	Manual	
5.	clock_source_index	Manual	
6.	clock_sources_offset	Manual	
7.	clock_sources_count	Manual	
8.	clock_sources	Manual	

8. Support status of AVDECC

Describe the support status for the ADP, AECP, and ACMP handled by AVDECC. Each protocol acts as Talker / Listener role of AVDECC Entity. Controller role is not included. For details of each protocol, refer to the IEEE 1722.1 standard.

Table 8.1 Supported message_type of ADP

No	message_type	Supported	Description
1.	ENTITY_AVAILABLE	v	
2.	ENTITY_DEPARTING	v	
3.	ENTITY_DISCOVER	v	ENTITY_AVAILABLE is returned for the command from the controller.

Table 8.2 Supported message_type of AECP

No	message_type	Supported	Description
1.	AEM_COMMAND		For controller
2.	AEM_RESPONSE	v	
3.	ADDRESS_ACCESS_COMMAND		For controller
4.	ADDRESS_ACCESS_RESPONSE		
5.	AVC_COMMAND		For controller
6.	AVC_RESPONSE		
7.	VENDOR_UNIQUE_COMMAND		For controller
8.	VENDOR_UNIQUE_RESPONSE		
9.	HDCP_APM_COMMAND		For controller
10.	HDCP_APM_RESPONSE		

There are multiple functions (command_type) in AEM_COMMAND and AEM_RESPONSE. It shows a list of supported functions. Currently, only AEM_RESPONSE is supported.

Table 8.3 Supported command_type of AEM_RESPONSE

No	command	Supported	Description
1.	ACQUIRE_ENTITY		
2.	LOCK_ENTITY		
3.	ENTITY_AVAILABLE	v	
4.	CONTROLLER_AVAILABLE		
5.	READ_DESCRIPTOR	v	
6.	WRITE_DESCRIPTOR		
7.	SET_CONFIGURATION		
8.	GET_CONFIGURATION		
9.	SET_STREAM_FORMAT		
10.	GET_STREAM_FORMAT		
11.	SET_VIDEO_FORMAT		
12.	GET_VIDEO_FORMAT		
13.	SET_SENSOR_FORMAT		
14.	GET_SENSOR_FORMAT		
15.	SET_STREAM_INFO		
16.	GET_STREAM_INFO		
17.	SET_NAME		
18.	GET_NAME		
19.	SET_ASSOCIATION_ID		
20.	GET_ASSOCIATION_ID		
21.	SET_SAMPLING_RATE		
22.	GET_SAMPLING_RATE		

CONFIDENTIAL

23.	SET_CLOCK_SOURCE		
24.	GET_CLOCK_SOURCE		
25.	SET_CONTROL		
26.	GET_CONTROL		
27.	INCREMENT_CONTROL		
28.	DECREMENT_CONTROL		
29.	SET_SIGNAL_SELECTOR		
30.	GET_SIGNAL_SELECTOR		
31.	SET_MIXER		
32.	GET_MIXER		
33.	SET_MATRIX		
34.	GET_MATRIX		
35.	START_STREAMING		
36.	STOP_STREAMING		
37.	REGISTER_UN SOLICITED_NOTIFICATION		
38.	DEREGISTER_UN SOLICITED_NOTIFICATION		
39.	IDENTIFY_NOTIFICATION		
40.	GET_AVB_INFO		
41.	GET_AS_PATH		
42.	GET_COUNTERS		
43.	REBOOT		
44.	GET_AUDIO_MAP		
45.	ADD_AUDIO_MAPPINGS		
46.	REMOVE_AUDIO_MAPPINGS		
47.	GET_VIDEO_MAP		
48.	ADD_VIDEO_MAPPINGS		
49.	REMOVE_VIDEO_MAPPINGS		
50.	GET_SENSOR_MAP		
51.	ADD_SENSOR_MAPPINGS		
52.	REMOVE_SENSOR_MAPPINGS		
53.	START_OPERATION		
54.	ABORT_OPERATION		
55.	OPERATION_STATUS		
56.	AUTH_ADD_KEY		
57.	AUTH_DELETE_KEY		
58.	AUTH_GET_KEY_LIST		
59.	AUTH_GET_KEY		
60.	AUTH_ADD_KEY_TO_CHAIN		
61.	AUTH_DELETE_KEY_FROM_CHAIN		
62.	AUTH_GET_KEYCHAIN_LIST		
63.	AUTH_GET_IDENTITY		
64.	AUTH_ADD_TOKEN		
65.	AUTH_DELETE_TOKEN		
66.	AUTHENTICATE		
67.	DEAUTHENTICATE		
68.	ENABLE_TRANSPORT_SECURITY		
69.	DISABLE_TRANSPORT_SECURITY		
70.	ENABLE_STREAM_ENCRYPTION		
71.	DISABLE_STREAM_ENCRYPTION		
72.	SET_MEMORY_OBJECT_LENGTH		
73.	GET_MEMORY_OBJECT_LENGTH		
74.	SET_STREAM_BACKUP		
75.	GET_STREAM_BACKUP		
76.	AUTH_DELETE_KEY		

Table 8.4 Supported message_type of ACMP

No	Descriptor name	Supported	Description
1.	CONNECT_TX_COMMAND	v	Sent to Talker when sending CONNECT_RX_COMMAND to Listener
2.	CONNECT_TX_RESPONSE	v	
3.	DISCONNECT_TX_COMMAND	v	Sent to Talker when sending CONNECT_RX_COMMAND to Listener
4.	DISCONNECT_TX_RESPONSE	v	
5.	GET_TX_STATE_COMMAND		For controller
6.	GET_TX_STATE_RESPONSE	v	
7.	CONNECT_RX_COMMAND		For controller
8.	CONNECT_RX_RESPONSE	v	
9.	DISCONNECT_RX_COMMAND		For controller
10.	DISCONNECT_RX_RESPONSE	v	
11.	GET_RX_STATE_COMMAND		For controller
12.	GET_RX_STATE_RESPONSE	v	
13.	GET_TX_CONNECTION_COMMAND		For controller
14.	GET_TX_CONNECTION_RESPONSE		

9. Controller application for AVDECC

This application does not have the controller function of AVDECC. When using AVDECC, a Controller application is required. Table 9.1 is an example as an application with the function of Controller. Figure 9.1 shows the processing sequence for connecting Talker and Listener by the Controller. The method of sending protocol data from Controller depends on the specification of each Controller Application.

Table 9.1 Example of application with Controller function

No	Application	Project name	URL	License
1.	avdecccmdline	avdecc-lib	https://github.com/audioscience/avdecc-lib	MIT
2.	avdecc	avdecc-cmd	https://github.com/jdkoftinoff/avdecc-cmd	BSD-2-Clause

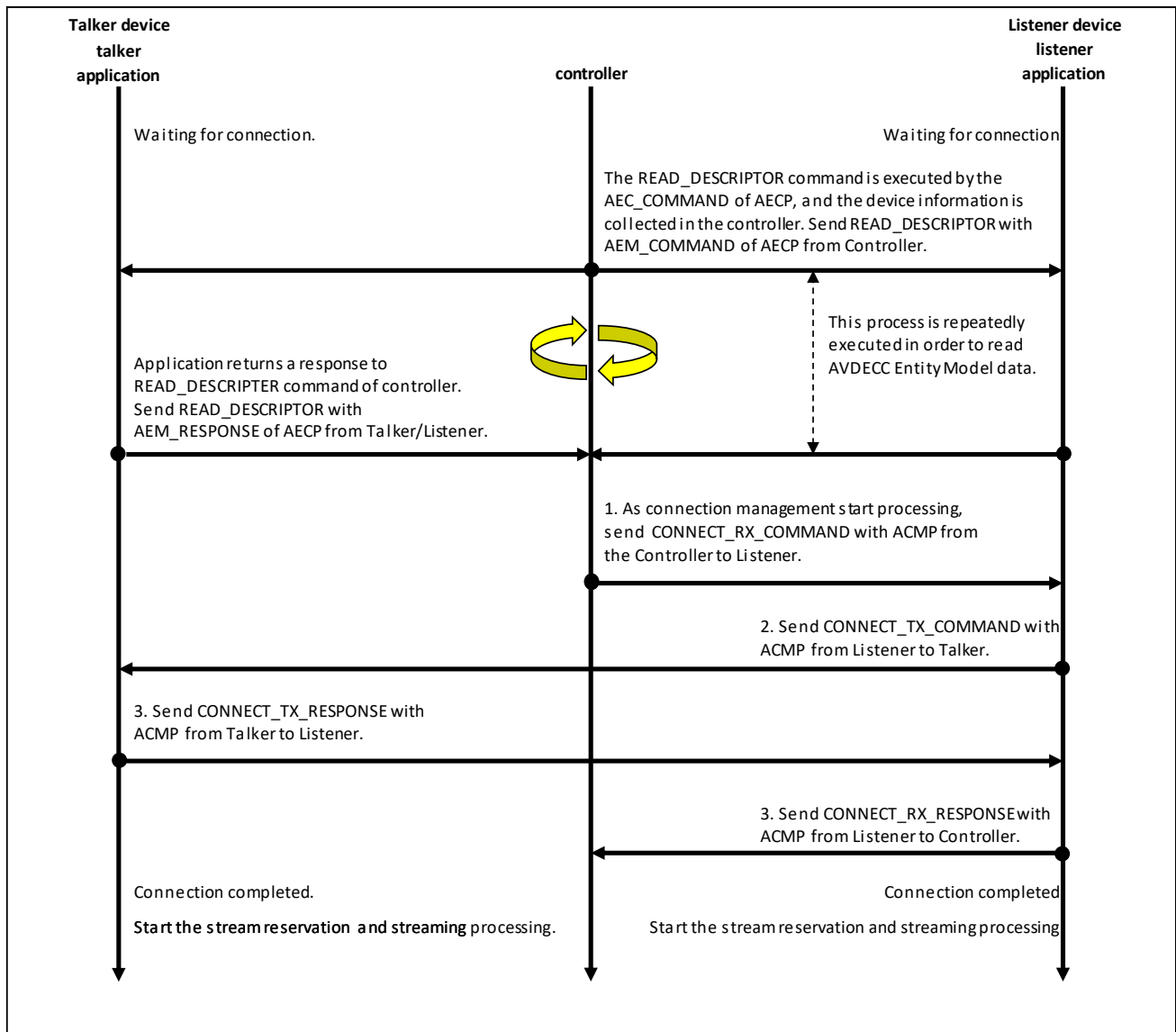


Figure 9.1 The processing sequence between Controller and Talker / Listener

CONFIDENTIAL

REVISION HISTORY	Ethernet AVB Software Launcher Application Application Note
------------------	--

Rev.	Date	Description	
		Page	Summary
0.10	Apr. 22, 2016	-	First edition issued
0.20	May. 20, 2016	2	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.
		2	1.3 Restrictions Add restrictions of maap_daemon
		3	1.4 Terminology Update of Table 1.3 - Add MAAP
		4	2.2 Software relationship, Figure 2.1 Add E) maap_daemon
		5	3.1 Function of avblauncher Add section 1.3 and 3.5 for function of maap_daemon
		6	3.3 Module structure Update of Table3.1 - Add new sample ini files and script
		7	3.4 Configuration parameter Update of Table 3.1 - Add MAAP_DAEMON_CMD - Add MAAP_DAEMON_SHM_KEY
		8-9	3.5 The processing outline for avblauncher Update of Figure 3.2 and add of Figure 3.3 - Add function of maap_daemon
		13	4.2 Build process without Yocto recipe Step1 - Delete the steps to set the environment variable for 32bit
		14-15	5. Execute the applications Modify the example of the execution command
		16	5.3 Command parameters Update of Table 5.1 - Add option -a for maap_daemon
		17-18	6.1 INI file format Add the setting example related to the MAAP Update of Table 6.1 - Changed the specification of Key Name "DEST_ADDR"
		18	6.2 The function to replace the configuration parameter Update the description of the conversion operation
0.40	Jan. 13, 2017	1	1.2.1 Standard Update of Table 1.1 - Add "IEEE Std 1722.1-2013"
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		4	2.2 Software relationship, Figure 2.1 - Add AVDECC protocol
		5	3.1 Function of avblauncher - Add function of AVDECC - Added comment about application command being executed via "/bin/sh".
		6	3.3 Module structure Update of Table3.1 - Add new sample yaml file for AVDECC - Add avdecc library and jdksavdecc-c library

CONFIDENTIAL

		9	3.5 The processing outline for avblauncher Update Figure 3.3 - Add function related to AVDECC.
		13	4.2 Build process without Yocto recipe Step 3 - Add build process of AVDECC library Step 5 - Add new
		14	5. Execute the applications Modify the example of the execution command Step 1 and Step 4 - Add processing method when executing AVDECC
		18-19	6.1 INI file format Describe the INI file format and example separately. - Added Figure 6.1 and Figure 6.2 Add the setting example related to the AVDECC Update of Table 6.1 - Added the specification of Key Name "AVDECC" 6.2 The function to replace the configuration parameter - Added comment about application command being executed via "/bin/sh".
		20-27	7. How to define the AVDECC Entity Model Add a new chapter related to AVDECC
		28-30	8. Support status of AVDECC Add a new chapter related to AVDECC
		31	9. Controller application for AVDECC Add a new chapter related to AVDECC
0.50	Mar. 15, 2017	1	1.2.1 Standard Update of Table 1.1 - Update IEEE Std 1722 to 1722-2016
		1, 4	1.1. Overview, 2.2 Software relationship Modify project name of OpenAVB to OpenAvnu
		2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		3	1.5 Terminology Update of Table 1.3 - Add comments of related Standard at PTP, gPTP and AVDECC - Remove duplicated word of MAAP.
1.00	Jul. 12, 2017	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
		11	3.6 The processing outline of mrpdummy Step 2 Register SRP domain - Fixes wrong value of command parameter from "-m 2" to "-m 4".
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.
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 Delete restriction of maap_daemon
		4	2.2 Software relationship e) maap_daemon IPC of maap_daemon changed from SystemV shared memory to Unix socket interface.
		5	3.1 Function of avblauncher 1.3 Add maap_daemon initialize when execute maap_daemon. 4.5.3 Change IPC of maap_daemon to Unix socket interface.
		6	3.3 Module structure Update of Figure 3.1 - Add definition of daemon_cl and maap_daemon.

CONFIDENTIAL

			- Remove example ini file of daemon_cl for 100Mbps link speed.
		7	3.4 Configuration parameter Update of Table 3.1 - Delete definition of shared memory for maap_daemon. - Add definition of Unix socket interface for maap_daemon.
		8-9	3.5 The processing outline for avblauncher Update of Figure 3.2 and Figure 3.3 - Change IPC of maap_daemon to Unix socket interface.
		15	4.2 Build process without Yocto recipe Update the version of Yocto SDK to 2.4.1.
		19	5.3 Command parameters Update of Table 5.1 - Add connection with Unix socket interface to explanation of "-a" option.
		27	7 How to define the AVDECC Entity Model Fix typo of Table 7.9 title.
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.
1.40	Sep. 26, 2018	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document.
2.00	Dec. 25, 2018	2	1.2.2 Related Document Update of Table 1.2 - Update the version of related document. - Add Salvator-XS Board Hardware Manual. - Add Ebisu-4D Board Hardware Manual.
2.10	Apr. 06, 2021	-	- Update AddressList
2.20	Aug. 16, 2021	1	Add R-Car D3 support
		2	Table 1.2 Related documents - Update revision - Add R-CarD3 Hardware Manual
		4	2.2 Software relationship - Update project information for daemon_cl
2.30	Dec. 01, 2021	-	- Update Notice page
		2	Table 1.2 Related documents: Update version of "R-Car Series, 3rd Generation User's Manual: Hardware": From v1.00 to v2.30
2.31	Dec. 25, 2023	All	Update version of the document.

CONFIDENTIAL

Ethernet AVB Software Launcher Application
Application Note

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



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/>

Ethernet AVB Software Launcher Application



Renesas Electronics Corporation