User's Manual

32

OMX Integration Guide

Integration Guide: for Linux

— Preliminary —

Renesas Electronics
www.renesas.com

Rev.1.01 Dec. 15, 2014

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended 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; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

Table of Contents

Tables

1. Overview

1.1. About This Document

This document describes how to install the OMX Media Component into your system.

1.2. Summary

The integration of the OMX Media Component consists of the following steps.

1. Setup the build environment (see section 3).
2. Prepare the dependent modules(see section 4)
3. Prepare the UDF part(see section 5)
4. Install the OMX Media Component Libraries and the required files(see section 6)

When you use yocto recipe package files, refer to the yocto startup guide document to build and install the OMX Media Components and the related modules.

Since the settings such as the thread priority and the scheduling policy depend on the system requirement, the UDF part needs to be customized. For the details, see section 7.

1.3. Terminology

Table 1-1 lists the terms that are used in this document.

Table 1-1 Terminology

Term	Abbreviation	Description
OpenMAX IL	OMX	In this document, OMX means the Renesas implementation that is based on OpenMAX IL specifications.
OMX Media Component	-	An OpenMAX IL Component that has multi-media functions such as decoding and encoding.
Configuration File	-	A text file describes the information about the OMX Media Component configuration.
Base Configuration File	-	The base configuration file is the configuration file whose name is “omxr_config_base.txt”. The base configuration file has the description to include all the other configuration files.
Multi-Media Package	MMP	Multi-Media Package is the software package that provides multi-media functions for R-Car Series.
UVCS	-	Renesas proprietary video codec software module that provides multi-processing function for video decoding and encoding.
Memory Manager	-	Memory Manager is a module in the MMP. Memory Manager provides the functions to allocate the dynamic memory area for H/W IPs.
S3CTRL Driver	-	S3CTRL Driver is a module in the MMP. S3CTRL Driver converts provides the functions to convert from tiled-addressed picture data to linear-addressed picture data.
FDP Manager	-	FDP Manager is a module in the MMP. FDP Manager provides the video functions for deinterlacing, color space conversion and so on.
User Defined Functions	UDF	Hardware/OS Abstraction Layer of the OMX Media Component. UDF is provided as a sample of the OMX Media Component product in source code format.

1.4. Related Documents

Table 1-2 lists the related documents.

Table 1-2 List of Related Documents

No.	Document Name	Remarks
[1]	UVCS Adapted for Linux Common Engine Library Sample Codes Manual	
[2]	Memory Manager for Linux User’s Manual	
[3]	S3CTRL Driver for Linux User’s Manual	
[4]	FDP Manager for Linux User’s Manual	

2. Deliverables

2.1. OMX Product

Table 2-1 shows the OMX products and the dependencies between the products. Table 2-1 also defines the abbreviations for each product.
For example, *cmn*, *vdcmn*, *uvcscmn* and *h264d* are required to decode H.264 streams.

Table 2-1 OMX Product List

OMX Product Name	Abbrev.	Dependent Product
OMX Media Component Common Library for Linux	*cmn*	-
OMX Media Component Video Decoder Common Library for Linux	*vdcmn*	*cmn, uvcscmn*
OMX Media Component Video Encoder Common Library for Linux	*vecmn*	*cmn, uvcscmn*
OMX Media Component H.264 Decoder Library for Linux	*h264d*	*vdcmn*
OMX Media Component MPEG4 Decoder Library for Linux	*m4vd*	*vdcmn*
OMX Media Component VC-1 Decoder Library for Linux	*vc1d*	*vdcmn*
OMX Media Component MPEG2 Decoder Library for Linux	*m2vd*	*vdcmn*
OMX Media Component DivX Decoder Library for Linux	*dvxd*	*vdcmn*
OMX Media Component RealVideo Decoder Library for Linux	*rlvd*	*vdcmn*
OMX Media Component H.263 Decoder Library for Linux	*h263d*	*vdcmn*
OMX Media Component Sorenson H.263 Decoder Library for Linux	*soh263d*	*vdcmn*
OMX Media Component VP6 Decoder Library for Linux	*vp6d*	*vdcmn*
OMX Media Component VP8 Decoder Library for Linux	*vp8d*	*vdcmn*
OMX Media Component AVS Decoder Library for Linux	*avsd*	*vdcmn*
OMX Media Component H.264 Encoder Library for Linux	*h264e*	*vecmn*
UVCS Adapted for Linux Common Engine Library	*uvcscmn*	-

2.2. Contents

Table 2-2 describes the file contents of the OMX products. ‘Software’ directory is provided as tar archive which can be extracted by the tar command.All the OMX products have the same directory structure, except *uvcscmn*. For the details of *uvcscmn,* see related document [1].

Table 2-2 File Contents of OMX Products

Directory Name			Contents
Software			
	OMXR		
		config	Configuration files
		include	Header files
		lib	This directory has sub directories for each compiler tool chain. Each sub directory contains OMX library files that are built with the tool chain. Choose the libraries in accordance with the system requirement. For the details of the compiler tool chain, see each OMX product manual.
	UDF_Linux		Source codes and makefiles of UDF for Linux environment. Only *cmn*, *vdcmn* and *vecmn* have this directory. The source codes in this directory must be built to integrate the OMX product. For the details, see section 5.
Documentation			
	en		User’s manual (English)

3. Build Environment Settings

3.1. BSP Environment

To set up the build environment, see the Linux BSP startup guide of the target board.

3.2. Prerequisites

GNU Autotools and autoreconf are required to build UDF part. Table 3-1 shows the confirmed tool versions.

Table 3-1 Confirmed Tool Versions

tool	Version
Host OS	Ubuntu 12.04.2 LTS
autoconf	Version 2.68
automake	Version 1.11.3
libtool	Version 2.4.2
autoreconf	Version 2.68

4. Preparation of Dependent Modules

4.1. MMP Modules

The UDF implementation depends on the modules of the MMP. Table 4-1 and Table 4-2 show the dependent modules for OMX Video Decoder (*vdcmn*) and OMX Video Encoder (*vecmn*), respectively. Please prepare the required modules previous to building the UDF part in accordance with related document [2], [3] and [4]. The required library files are supposed to be located in $path2lib directory. The $path2lib directory is an arbitrary directory on the Host PC.

Table 4-1 Required MMP modules for OMX Video Decoder (*vdcmn*)

Module Name	Note
Memory Manager	-
S3CTRL Driver	-
FDP Manager	-

Table 4-2 Required MMP modules for OMX Video Encoder (*vecmn*)

Module Name	Note
Memory Manager	-

To run OMX Video Decoder and OMX Video Encoder, the required libraries must be placed in the shared library search path of the target board. In addition, the kernel modules of these modules must be installed to Linux kernel. For the details of the procedure, see related document [2], [3] and [4].

4.2. UVCS Common Engine Library

The OMX Video Decoder (*vdcmn*) and OMX Video Encoder (*vecmn*) depends on the UVCS Common Engine Library (*uvcscmn*). The *uvcscmn* is a Linux kernel module and is provided as a sample source code. To run OMX Video Decoder and OMX Video Encoder, the *uvcscmn* must be installed to Linux kernel. For the details of the source code implementation and the install procedure, see related document [1].

5. Preparation of UDF Part

This section describes how to build the UDF part. $build_dir means a work directory path of the Host PC.

Step1: Setup Dependent Libraries and Headers

The libraries and header files of the dependent modules described in section 4.1 must be located in the Host PC. Table 5-1 shows the file names of the required libraries and the directory path. Table 5-2 shows the file names of the required header files and the directory path. $path2lib and $path2include in the tables are arbitrary directories on the Host PC.

Table 5-1 Required MMP libraries

Module Name	Library File Name (so name)	Library Path (Example)
Memory Manager	libmmgr.so	$path2lib
S3CTRL Driver	libs3ctl.so	$path2lib
FDP Manager	libfdpm.so	$path2lib

Table 5-2 Required MMP Header Files

Module Name	Required Header Files	Include Path (Example)
Memory Manager	mmngr_user_public.h	$path2include/mmngr/user/include
S3CTRL Driver	s3ctl_user_public.h	$path2include/s3ctl/user/include
FDP Manager	fdpm_api.h fdpm_public.h fdpm_def.h	$path2include/fdpm/user/include $path2include/fdpm/kernel/include

Step2: Setup the UDF build tree

In this step, copy the OMX source files from each deliverable directory to $build_dir directory. $cmn_dir, $vdcmn_dir and $vecmn_dir mean ‘Software’ directory of the *cmn*, *vdcmn* and *vecmn* deliverables, respectively.

At first, copy the directory that contains the OMX header files from the *cmn* deliverable to $build_dir.

```
$ cd $build_dir
$ cp –r $cmn_dir/OMXR/include ./
```

Then, copy the UDF source code from the *cmn*, *vdcmn* and *vecmn* deliverables to $build_dir.

```
$ cd $build_dir
$ cp –r $cmn_dir/UDF_Linux/* ./
$ cp –r $vdcmn_dir/UDF_Linux/* ./    # When you install vdcmn
$ cp –r $vecmn_dir/UDF_Linux/* ./    # When you install vecmn
```

Step 3: Generate configure script

Generate the configure script as follows:

```
$ cd $build_dir
$ chmod u+x autogen.sh
$ ./autogen.sh
```

Step 4: Generate Makefile

Execute the script that is generated at the Step 3 with the arguments and the environmental variables that are described in Table 5-3 and Table 5-4.

Table 5-3 Command Argument for the Configure Script

Argument	Instruction
--prefix	Specify the path of the directory on the Host PC where the UDF libraries are installed.
--host	Specify ‘arm-linux’.

Table 5-4 Environmental Variable for Configure Script

Environmental Variable Name	Instruction
CC	Specify the path of the C cross compiler of the build environment.
CFLAGS	Specify the path of the directories where the header files of the dependent modules are stored.
LDFLAGS	Specify the path of the directories where the library files of the dependent modules are stored.
OMXR_DEFAULT_CONFIG_FILE_NAME	Specify the path of the base configuration file in the rootfs of the target board. If not specified, the default is set as ‘/usr/lib/omxr_config_base.txt’.

The following is an example of the configure script execution. In this example, the target directory of the install is $temp_installdir and let $path2compiler be a directory in which the compiler tool chain is installed.

```
$ cd $build_dir
$ ./configure ¥
   --prefix=$temp_installdir ¥
   --host=arm-linux ¥
   CC=$path2complier/gcc-linaro-arm-linux-gnueabihf-4.7-2013.02-01-20130221_linux/bin/arm-linux-gnueabihf-gcc ¥
   CFLAGS=" ¥
      -I/$path2include/mmngr/user/include ¥
      -I/$path2include/fdpm/user/include ¥
      -I/$path2include/fdpm/kernel/include ¥
      -I/$path2include/s3ctl/user/include" ¥
   LDFLAGS= ¥
      -L/$path2lib/lib ¥
   OMXR_DEFAULT_CONFIG_FILE_NAME=/usr/lib/omxr_config_base.txt
```

Step 5: Build

Run make command by using the Makefile that is generated at the step 4.

```
$ cd $build_dir
$ make
```

Step 6: Install

Install the UDF libraries to the target directory as follows. The library files will be generated in the directory that is specified by '--prefix' option at Step 4.

```
$ cd $build_dir
$ make install
```

If the target directory is on the Host PC, copy the libraries to the library directory on the target board.

Table 5-5 lists the library files that are installed by this installation procedure.

Table 5-5 Installed Libraries

Deliverable	Library File Name
cmn	libomxr_utility.so*
vdcmn	libomxr_uvcs_udf.so* libomxr_videoconverter.so* libomxr_cnvosdep.so* libomxr_cnvpfdp.so*
vecmn	ibomxr_uvcs_udf.so*

6. Installation of OMX Media Component

This section describes the installation procedure of the OMX Media Component.

Table 6-1 lists the library file names and the configuration file names for each OMX Media Component product that is to be installed.
For the dependencies between the OMX Media Component products, see section 2.1.

Table 6-1 OMX Media Component Libraries and Configuration Files

Product Name	Library File Name (Real Name)	Configuration File Name
cmn	libomxr_core.so.2.x.x libomxr_mc_cmn.so.2.x.x	omxr_config_base.txt (base configuration file)
vdcmn	libomxr_mc_vcmn.so.2.x.x libomxr_mc_vdcmn.so.2.x.x libuvcs_dec.so.1.x.x libvcp3_mcvd.so.1.x.x	omxr_config_vdcmn.txt
vecmn	libomxr_mc_vcmn.so.2.x.x libomxr_mc_vdemn.so.2.x.x libuvcs_enc.so.1.x.x libvcp3_mcve.so.1.x.x	omxr_config_vecmn.txt
h264d,	libomxr_mc_h264d.so.2.x.x libvcp3_avcd.so.1.x.x	omxr_config_h264d.txt
m4vd	libomxr_mc_m4vd.so.2.x.x libvcp3_m4vd.so.1.x.x	omxr_config_m4vd.txt
vc1d	libomxr_mc_vc1d.so.2.x.x libvcp3_vc1d.so.1.x.x	omxr_config_vc1d.txt
m2vd	libomxr_mc_m2vd.so.2.x.x libvcp3_m2vd.so.1.x.x	omxr_config_m2vd.txt
dvxd	libomxr_mc_divxd.so.2.x.x libvcp3_dvxd.so.1.x.x	omxr_config_divxd.txt
rlvd	libomxr_mc_rlvd.so.2.x.x libvcp3_rlvd.so.1.x.x	omxr_config_rlvd.txt
h263d	libomxr_mc_h263d.so.2.x.x libvcp3_hv3d.so.1.x.x	omxr_config_h263d.txt
soh263d	libomxr_mc_soh263d.so.2.x.x libvcp3_srsd.so.1.x.x	omxr_config_soh263d.txt
vp6d	libomxr_mc_vp6d.so.2.x.x libvcp3_vp6d.so.1.x.x	omxr_config_vp6d.txt
vp8d	libomxr_mc_vp8d.so.2.x.x libvcp3_vp8d.so.1.x.x	omxr_config_vp8d.txt
avsd	libomxr_mc_avsd.so.2.x.x libvcp3_avsd.so.1.x.x	omxr_config_avsd.txt
h264e	libomxr_mc_h264e.so.2.x.x libvcp3_avce.so.1.x.x	omxr_config_h264e.txt

To install the OMX Media Component deliverables, follow the following steps. In the following procedure, let $installdir be a directory that is shared library path in the rootfs of the target board.

Step1: Copy the real name files to the target board

Copy the real name files of each OMX Media Component deliverable to $installdir.

```
$ cd (Software/OMXR/lib directory of each OMX MediaComponent Deliverable)
$ cp libomxr_mc_vdcmn.so.2.x.x   $installdir/
```

Step2: Create symbolic links to the real name files

Create symbolic links (so name and linker name) to each real name file. The following is an example to make symbolic links to libomxr_mc_vdcmn.so.2.x.x.

```
$ cd $installdir/
$ ln –s libomxr_mc_vdcmn_.so.2.x.x   libomxr_mc_vdcmn.so.2     # so name
$ ln –s libomxr_mc_vdcmn..so.2       libomxr_mc_vdcmn.so       # linker name
```

Step3: Copy the configuration files to the target board

$config_path is a directory in which the configuration files are placed. The default path of this directory is specified in the UDF build procedure (see section 5). Copy the configuration file of each OMX Media Component deliverable to $config_path.

```
$ cd (Software/OMXR/config directory of each OMX Media Component Deliverable)
$ cp omxr_config_*.txt $config_path/
```

All of the configuration files must be located in the same directory.

7. System Dependent Settings

7.1. Thread Setting

7.1.1. Thread Priority (Video)

This section describes how to set the appropriate priorities to threads of OMX Media Components in the user systems. An OMX Media Component calls the OmxrCreateThread function to create threads. Since OmxrCreateThread is defined in the UDF part, the user can set the thread priority and the scheduling policy for each thread. The following is the prototype of the OmxrCreateThread function:

```
OMX_ERRORTYPE OmxrCreateThread(OMX_U32 *pu32ThreadId, void *pFunction, OMX_PTR pvInfo,
OMX_STRING strPriority, OMX_U32 u32Attribute)
```

To determine the thread priority between OMX Media Component threads, OMX Media Component passes the strings that indicate the thread priority as a parameter to OmxrCreateThread function. Table 7-1 shows the relation between threads and the thread priority and describes the basic role of each thread.

Table 7-1 Thread Priority

strPriority	Thread Priority	Role
"PRIORITY.HIGH.MC.CMN.ST"	High	The main thread of the OMX Media Component.
"PRIORITY.HIGH.MC.CMN.CB"	High	The callback thread of the OMX Media Component.
"PRIORITY.HIGH.ME.UDF.UVCS"	High	The listener thread for UVCS Common Engine Library. This thread is implemented in the UDF.
"PRIORITY.HIGH.CNV"	High	The main thread of the Converter module. This thread is implemented in the UDF.
"PRIORITY.LOW.ME.COPY"	Low	The thread for Video Decoder Component that copies an input stream to the internal stream buffer by memcpy.

7.2. Memory Requirement

7.2.1. Memory Areas Allocated by Memory Manager

This section describes memory areas allocated by Memory Manager in the UDF part. In order to allocate memory area which is accessible by hardware IPs, the UDF part uses Memory Manager. For the details of Memory Manager, see related document [2]. Table 7-2 and Table 7-3 show memory areas allocated by Memory Manager.

Table 7-2 Memory Areas Allocated by Memory Manager (video decoder)

Memory Type	Area
input buffer	CMA
output buffer	CMA
work buffer	CMA
stream_work_0	CMA
stream_work_1	CMA
stream_work_2	MVBUF
stream_work_4	CMA
stream_work_5	CMA
frame_mem	OMXBUF

Table 7-3 Memory Areas Allocated by Memory Manager (video encoder)

Memory Type	Area
input buffer	CMA
output buffer	CMA
stream_work_0	CMA
stream_work_1	CMA
stream_work_2	CMA
stream_work_4	OMXBUF
stream_work_5	CMA

7.2.2. Memory Placement Restrictions

This section describes memory placement restrictions and the way to deal with them in the UDF part.
Table 7-4 and Table 7-5 show memory-placement restrictions in UDF part of video decoder.

Table 7-4 Memory Placement Restrictions (video decoder)

Memory Type	Restrictions	Implementations
stream_work_2	This area should be allocated in the MVBUF. For the details of MVBUF, see related document [2].	This area is allocated as MVBUF.
frame_mem	This area should be allocated in S3CTRL managed area, which is unit of power-of-2 sized bytes. For the deatails of S3CTRL, see related document [3]. Also, for restriction of image placement on frame memory, extra 65536bytes is required.	This area is allocated as OMXBUF, and its size is calculated by following formula. *-roundup_po2(size + 65536)* where <roundup_po2> is rounding up to the power of 2 (i.e. 256, 512, 1024 …), and <size> is actual frame memory size.

Table 7-5 Memory Placement Restrictions (video encoder)

Memory Type	Restrictions	Implementations
stream_work_4	This area should be allocated in the OMXBUF. For the details of OMXBUF, see related document [2].	This area is allocated as OMXBUF.

7.2.3. Hardware Address

This section describes about ‘hardware address’ that is described in User’s Manual.
Use ‘hardware address’ as address that is accessible from hardware in case using the following hardware IPs: VCP, 2DDMAC, and FDP. On the other hand, in case using the other hardware IPs, must use physical address that is translated from ‘hardware adress’ by MemoryManager. For the details of address translation of Memory Manager, see related document [2].

7.2.4. OMXBUF fragmentation that is caused by multiple instances

In case of using OMXBUF, which is memory area is physically successive memory area, used by multiple instances, there is possibility of memory allocation failure which is caused by fragmentation. Figure 7-1 illustrates an example of the OMXBUF fragmentation situation.

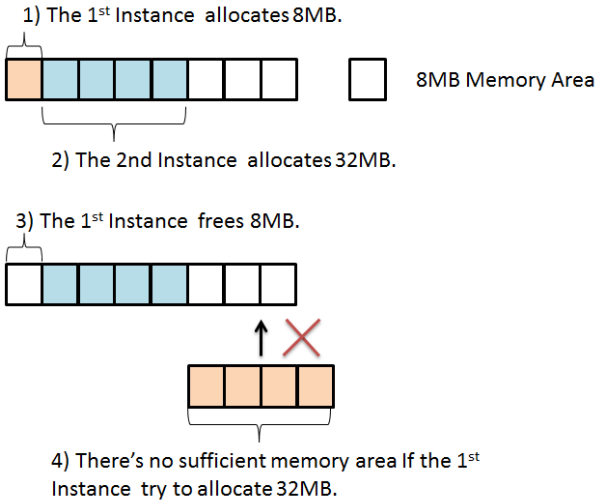

Figure 7-1 Problem: OMXBUF fragmentation that is caused by multiple instances

To avoid the fragmentation above, modify the UDF part. Figure 7-2 illustrates the example of the solution to avoid OMXBUF fragmentation. For example, it works well that extending the first 8MB allocation to 32MB, if the instance requires 8MB and subsequently requires 32MB.

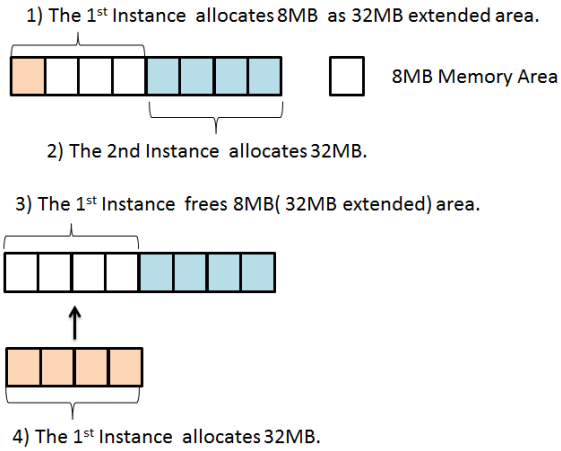

Figure 7-2 Solution: OMXBUF fragmentation that is caused by multiple instances

Regarding modification of the UDF part, see the "OmxrAllocateWorkBuffer_allocfunc" function in "omxr_workbuffer_func.c" file.

REVISION HISTORY	OMX Integration Guide Integration Guide : for Linux

Rev.	Date	Description	
		Page	Summary
0.06	Jan. 24, 2014	―	Draft revision based on Japanese User's Manual Rev.0.06.
0.07	Mar. 13, 2014	6	Modify description about file contents.
		9	Add permission setting to Step3.
0.0.8	Jul. 23, 2014	15	Add section 7.2 Memory Requirement
1.0.0	Jul. 29, 2014	15	Add section 7.2.1 Memory Areas Allocated by Memory Manager
	Aug. 8,2014	15	Fix Table 0-1, and add Table 7-5
	Aug. 21,2014	17	Add section 7.2.3.Hardware Address17
	Oct. 27,2014	15	Add "work buffer" in Table7-2.
1.0.1	Dec. 15, 2014	18	Add the section 7.2.4

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

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

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

OMX Integration Guide
Integration Guide: for Linux

Publication Date: Rev. 1.01 Dec. 15, 2014

Published by: Renesas Electronics Corporation

OMX Integration Guide

Integration Guide: for Linux

Renesas Electronics Corporation