User's Manual

Linux Interface Specification Yocto recipe Start-Up Guide

User's Manual: Software

R-Car H3/M3 Series

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

Renesas Electronics
www.renesas.com

Rev.0.30 Oct. 2015

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.

(2012.4)

Trademark

- Linux® is a registered trademark or a trademark of Linus Torvalds in the United States and/or other countries.
- ARM® is a registered trademark or a trademark of ARM Company in the United States and/or other countries.
- 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 processor.

- **[Purpose]**
  This manual is intended to give users an understanding of the functions of the R-Car H3/M3 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
  → See the R-Car H3/M3 User's Manual.
  — To know the electrical specifications of the multimedia processor for R-Car H3/M3
  → See the R-Car H3/M3 Data Sheet.

- **[Conventions]**
  The following symbols are used in this manual.
  Data significance: Higher digits on the left and lower digits on the right
  Note: Footnote for item marked with Note in the text
  Caution: Information requiring particular attention
  Remark: Supplementary information
  Numeric representation: Binary ... ××××, 0b××××, or ××××B
  Decimal ... ××××
  Hexadecimal ... 0x×××× or ××××H
  Data type: Double word ... 64 bits
  Word … 32 bits
  Half word ... 16 bits
  Byte ... 8 bits

Table of Contents

Introduction 1

1. R-Car H3 Linux BSP package files 2
1.1 Reference (R-Car H3) 2

2. ENVIRONMENTAL REQUIREMENT 3
2.1 Setting of parts 3
2.2 Setting of dip switch 5

3. Building Instructions 6
3.1 In case of BSP + 3D Graphics + Multimedia package 8
3.2 In case of BSP + 3D Graphics (without Multimedia package) 11
3.3 In case of BSP Only 13

4. Writing of IPL/Secure 14
4.1 Writing data 14
4.2 Dip-Switch 14
4.3 How to write 14
4.4 IPL/Secure write 16

5. Confirm starting of U-Boot and Linux 17

6. Exporting Toolchains 19

7. Memory map 22

8. U-Boot command 27

Introduction

This start-up guide explains R-Car H3 Yocto recipe package files, the system environments, the make method of kernel, the operating of U-Boot and so on.

This product R-Car H3 Yocto recipe is a basic package to operate built-in Linux and basic middleware on the R-Car H3 System Evaluation Board. Please contact Renesas Electronics person who provided this product to you in case of questions.

1. R-Car H3 Linux BSP package files

This Yocto recipe will be taken

The U-Boot source code from:

git://github.com/renesas-rcar/u-boot.git

R-Car H3 Linux source code from:

git://git.kernel.org/pub/scm/linux/kernel/git/horms/renesas-bsp.git

1.1 Reference (R-Car H3)

document name	version	date
R-Car Series, 3rd Generation User's Manual: Hardware	Rev.0.50	2015/07/31
R-Car Series, 3rd Generation Electrical Characteristics	Rev.0.50	2015/07/31
R-CarH3-SiP System Evaluation Board Salvator-X Hardware Manual RTP0RC7795SIPB0010S	Rev.0.09	2015/09/29

2. ENVIRONMENTAL REQUIREMENT

2.1 Setting of parts

Host PC and terminal software are necessary for the operation of this product. Furthermore Ethernet cable is required to use NFS mount function. Please refer to Table 1.

Table 1 R-Car H3 Linux BSP Environmental Requirement

Equipment	Explanation
Linux Host PC	Ubuntu 14.04 LTS(64bit) is recommended as OS. It is used as building and debugging environment. It is used as TFTP server and NFS server.
Windows Host PC	Windows 7 is recommended as OS. It is used as debugging environment. Terminal software and VCP driver are executed.
Terminal software	Please use following software. 1) TeraTerm (Confirmed with Japanese version of Tera Term 4.87 Available at http://sourceforge.jp/projects/ttssh2)
VCP driver	Please install in Windows Host PC. Execute CP210xVCPInstaller_x86/x64.exe for install before connect. USB become virtual COM port on terminal software. Please connect to CN25 (Serial-USB Bridge CP2102) on H3 System Evaluation Board. (Available at http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx)
TFTP server software	It is used when Hyper Flash is written by U-Boot or Image is downloaded.
NFS server software	It is used when File system is mounted by NFS.

Recommended Environment

The following shows a Recommended Environment.

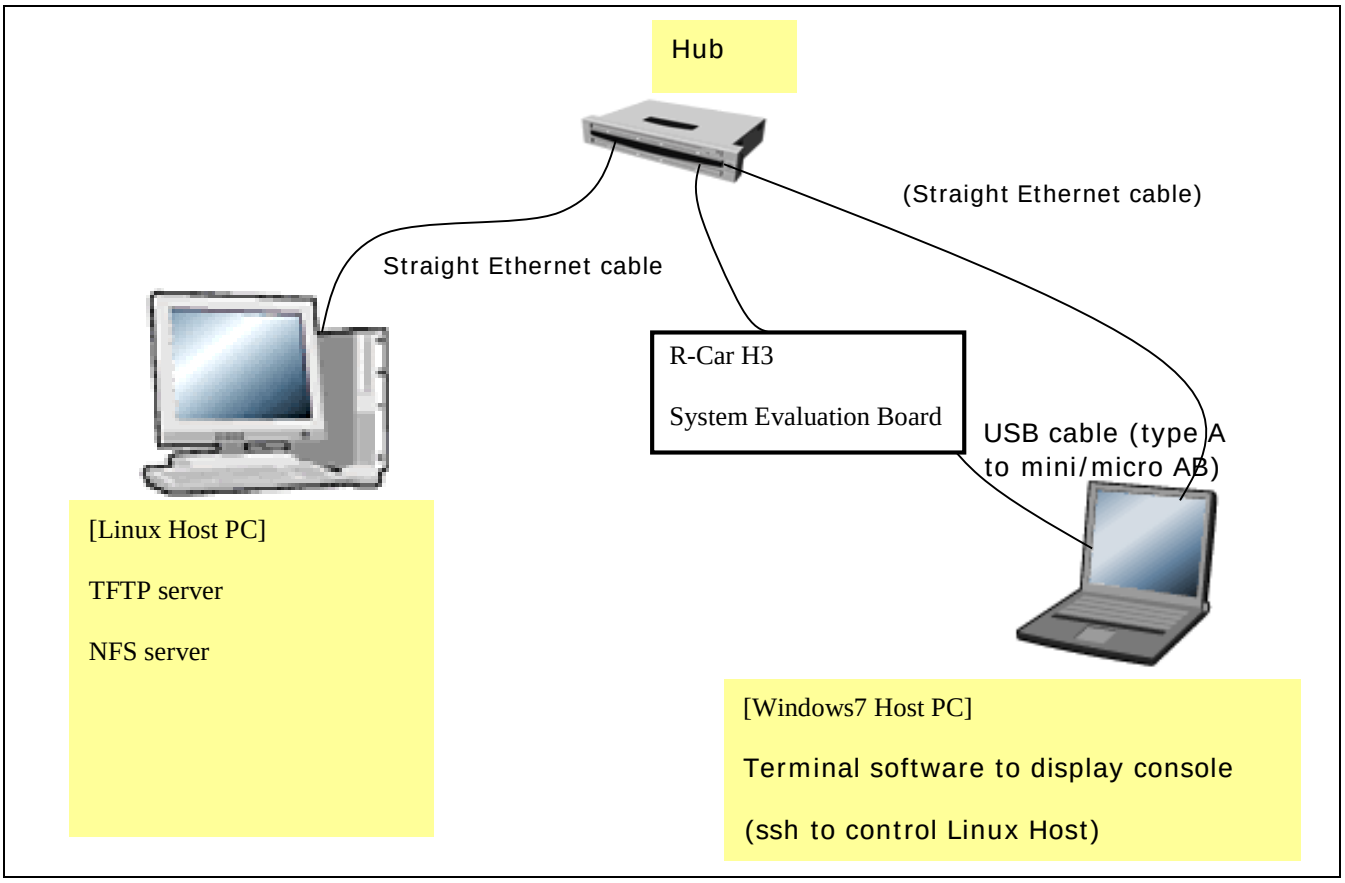

Figure 1 Recommended Environment for R-Car H3 Linux BSP

Note) Functions in covered with () are optional.

2.2 Setting of dip switch

The setting of R-Car H3 System Evaluation Board's dip switches is shown the following Table 2. Please refer to "R-CarH3-SiP System Evaluation Board RTP0RC7795SIPB0010S (Salvator-X) Setup Manual" for details.

Table 2 setting of Dip switches (R-Car H3)

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW2	QSPI-B	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW3	QSPI-C		Set(ON)	-	-	-	-	-	-
SW13	QSPI-D	set	-		-	-	-	-	-
SW4	SOFTSW	OFF	OFF	OFF	OFF	-	-	-	-
SW5	GPIO/PWM1	set			-	-	-	-	-
SW6	GPIO/PWM2	set			-	-	-	-	-
SW7	DDRBKUP-A		set		-	-	-	-	-
SW8	DDRBKUP-B	OFF	OFF	OFF	OFF	-	-	-	-
SW9	TRST#			set	-	-	-	-	-
SW10	MODESW-A	ON	ON	OFF	OFF	ON	ON	OFF	ON
SW11	MODESW-B	OFF	ON	ON	OFF	ON	ON	ON	ON
SW12	MODESW-C	OFF	ON	OFF	OFF	ON	ON	ON	ON
SW14	SSI78-M/S	set	-		-	-	-	-	-
SW15	USB-SW	set			-	-	-	-	-
SW16	SDHI0/DBG2-A	set	-		-	-	-	-	-
JP2	SDHI0/DBG2-B	set	-		-	-	-	-	-
SW28	VDDQVA_SD0	OFF	-	-	-	-	-	-	-
SW17	LVDS		set	-	-	-	-	-	-
SW20	TACTSW0		-	-	-	-	-	-	-
SW21	TACTSW1		-	-	-	-	-	-	-
SW22	TACTSW2		-	-	-	-	-	-	-
SW23	ACCSW	-	-	-	-	-	-	-	-
SW27	PRESET#		-	-	-	-	-	-	-
SW29	MIPI-SW	ON	ON	-	-	-	-	-	-
SW30	PHYAD	OFF	OFF	-	-	-	-	-	-

3. Building Instructions

You can build BSP by using Yocto Project. Please execute following steps in $WORK directory on Linux Host PC. Filesystem by making following instruction is the one for testing current BSP package in Renesas. Please note that Renesas has not been verified with any other build configuration or modified recipes except “core-image-wayland” configuration which is based on upstream Yocto Project deliverables and some additional packages correspond to gstreamer.

Note) Renesas executed following instructions with clean $WORK/build directory. You may use wipe-sysroot and/or bitbake -c cleansstate to reflect modifications of configuration files for Recipe as in open source Yocto Project’s standards, however Renesas strongly recommend to use recipe with clean $WORK/build directory for each configurations because there are some implicit dependency for header files exist to keep compatibility between application build scheme with/without proprietary software.

Step 1 installation of required commands

Ubuntu is used as Linux Host PC since Yocto Project Quick Start specifies Ubuntu as one of the distribution. In case of that you can install the required commands as follows.
Please refer to http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html for detail.

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat libsdl1.2-dev xterm
```

Note) There is a bitbake command in $WORK/poky/scripts/. Command path is available after step 6.

Note) When you use terminal interactions to build such as menuconfig under non-X terminal (ssh, etc.), please install “screen” command package to Host PC

Step 2 download of required files

Required files (poky, meta-linaro) are downloaded by git clone.

```
$ cd $WORK
$ git clone git://git.yoctoproject.org/poky
$ git clone git://git.linaro.org/openembedded/meta-linaro.git
$ git clone git://github.com/renesas-rcar/meta-renesas.git
```

Step 3 checkout

Please checkout available version of each git clone.

```
$ cd $WORK/poky
$ git checkout -b tmp b74ea963cefffad9fbd91d4eb9b240f6a8c86cd0

$ cd $WORK/meta-linaro
$ git checkout -b tmp 3f1933242062d1cf473d20b86fa4582963ff9441

$ cd $WORK/meta-renesas
$ git checkout -b tmp d2a2b521ed1b807f82d861bd514c04468fd8bb42
```

Note) tmp is a temporary name of a local branch. We can use checkout command without branch. Please note that HEAD refers directly to commit (detached HEAD).

Step 4 apply patch file

Because Renesas proprietary software is using linaro-GCC, it needs to apply. Please perform patch command.

```
$ cd $WORK/meta-renesas
$ export PATCH_DIR=meta-rcar-gen3/docs/sample/patch/patch-for-linaro-gcc
$ patch -p1 < \
  ${PATCH_DIR}/0001-rcar-gen3-gcc-linaro-remove-patch-use-lib-for-aarch6.patch
$ patch -p1 < \
  ${PATCH_DIR}/0002-rcar-gen3-add-readme-for-building-with-Linaro-Gcc.patch
$ unset PATCH_DIR
```

3.1 In case of BSP + 3D Graphics + Multimedia package

When you use proprietary Multimedia and 3D Graphics software from Renesas, please execute as following steps. Otherwise please skip to section 3.2 or 3.3.

Step 5 copy proprietary software into recipe directory structure

To use licensed 3D graphics software from Renesas, please copy deliverables of those software into recipe directory structure. Renesas provide shell script to copy those software.

```
Copy from Proprietary Software Package Media such as CD-ROM into $PKGS_DIR:
$ mkdir $PKGS_DIR; cp (storage media)/(each packages) $PKGS_DIR; ……
Install them into recipe directory structure:
$ cd $WORK/meta-renesas
$ sh \
  meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh \
  $PKGS_DIR
```

Note) Subdirectory is not supporting in $PKGS_DIR. Please store all packages on the root of $PKGS_DIR.
Note) Please use regular alphanumeric file name ([A-Za-z0-9_] e.g.) for $PKGS_DIR due to restrictions of current copy script.

Step 6 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd $WORK
$ source poky/oe-init-build-env
```

Step 7 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp $WORK/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/linaro-gcc/mmp/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x

Step 8 enable 3D Graphics and Multimedia package to install GSX modules

To enable 3D Graphics packages which use GSX, please overwrite local.conf as following instructions or modify local.conf as descriptions in $WORK/meta-renesas/meta-rcar-gen3/README.proprietary

```
To install GSX kernel module:
Overwrite local.conf
$ cp conf/local-wayland.conf conf/local.conf (For Wayland)
```

Step 9 enable Multimedia package

Please modify configurations in $WORK/build/conf/local.conf by following instructions

Note) Please refer to $WORK/meta-renesas/meta-rcar-gen3/README.proprietary

The following standard multimedia packages are enabled

No.	Functions	Explanation
1	MMNGR	Memory manager driver & shared libraries
2	FDPM	FDP driver & shared libraries
3	VSPM	VSP driver & shared libraries
4	OMX	OMX common parts

To enable optional multimedia functions, please add DISTRO_FEATURES_append to $WORK/build/conf/local.conf as DISTRO_FEATURES_append = " <function name>".

Note) These configurations exist near the end of local.conf.

Note) DISTRO_FEATURES_append are commented out by the default. To enable functions, please uncomment it.

For example
[Disable]
```
#DISTRO_FEATURES_append = " h264dec_lib"
```

[Enable (default)]
```
DISTRO_FEATURES_append = " h264dec_lib"
```

The following list is package name to enable/disable as optional multimedia functions

No.	Function name	Default value	Explanation
1	h264dec_lib	Enable	H264 decoder library
2	aaclcdec_lib	Enable	AAC-LC decoder library
3	aaclcdec_mdw	Enable	AAC-LC 2ch decoder middleware library

Step 9 building with bitbake

Please build as follows. The file system (core-image-weston-<supported board name>.tar.bz2) is created in $WORK/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in $WORK/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in $WORK/poky/meta/conf/bitbake.conf.

```
$ cd $WORK/build
$ bitbake core-image-weston (for Wayland)
```

Note) Please add mem=2048M to bootargs. For details, please refer to Section5 step6.

3.2 In case of BSP + 3D Graphics (without Multimedia package)

When you use licensed 3D Graphics software but will not use proprietary Multimedia software from Renesas, please execute as following steps. Otherwise please skip to section 3.3.

Step 5 copy proprietary software into recipe directory structure

To use licensed 3D graphics software from Renesas, please copy deliverables of those software into recipe directory structure. Renesas provide shell script to copy those software.

```
Copy from Proprietary Software Package Media such as CD-ROM into $PKGS_DIR:
$ mkdir $PKGS_DIR; cp (storage media)/(each packages) $PKGS_DIR; ……
Install them into recipe directory structure:
$ cd $WORK/meta-renesas
$ sh \
  meta-rcar-gen3/docs/sample/copyscript/copy_proprietary_softwares.sh \
  $PKGS_DIR
```

Note) Subdirectory is not supporting in $PKGS_DIR. Please store all packages on the root of $PKGS_DIR.
Note) Please use regular alphanumeric file name ([A-Za-z0-9_] e.g.) for $PKGS_DIR due to restrictions of current copy script.

Step 6 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd $WORK
$ source poky/oe-init-build-env
```

Step 7 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp $WORK/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/linaro-gcc/gfx-only/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x

Step 8 enable 3D Graphics package to install GSX modules

To enable 3D Graphics packages which use GSX, please overwrite local.conf as following instructions or modify local.conf as descriptions in $WORK/meta-renesas/meta-rcar-gen3/README.proprietary

```
To install GSX kernel module:
Overwrite local.conf
$ cp conf/local-wayland.conf conf/local.conf (For Wayland)
```

Step 9 building with bitbake

Please build as follows. The file system (core-image-weston-<supported board name>.tar.bz2) is created in $WORK/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in $WORK/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in $WORK/poky/meta/conf/bitbake.conf.

```
$ cd $WORK/build
$ bitbake core-image-weston (for Wayland)
```

Note) Please add mem=2048M to bootargs. For details, please refer to Section5 step6.

3.3 In case of BSP Only

When you will not use neither proprietary Multimedia software nor licensed 3D Graphics software from Renesas, please execute as following steps.

Step 5 execute source command

Please execute source command with oe-init-build-env for setting environment.

```
$ cd $WORK
$ source poky/oe-init-build-env
```

Step 6 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

```
$ cp $WORK/meta-renesas/meta-rcar-gen3/docs/sample/conf/<supported
board name>/linaro-gcc/bsp/*.conf ./conf/.
```

Note) <supported board name> is the one of the following: salvator-x

Note) core-image-weston is not supported even though local-wayland.conf is provided.

Step 7 building with bitbake

Please build as follows. The file system (core-image-minimal-<supported board name>.tar.bz2) is created in $WORK/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: salvator-x

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in $WORK/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in $WORK/poky/meta/conf/bitbake.conf.

```
$ cd $WORK/build
$ bitbake core-image-minimal
```

4. Writing of IPL/Secure

4.1 Writing data

filename	Program Top Address	Flash Save Address	description
bootparam_sa0.srec	0xE6320000	0x000000	Loader(Boot parameter)
bl2-<board_name>.srec	0xE6302000	0x040000	Loader
cert_header_sa6.srec	0xE6320000	0x180000	Loader(Certification)
bl31-<board_name>.srec	0x44000000	0x1C0000	ARM Trusted Firmware
tee-<board_name>.srec	0x44100000	0x200000	OP-Tee
u-boot-elf.srec	0x49000000	0x640000	U-boot

Note) Please refer to Chapter 5 for u-boot.

4.2 Dip-Switch

a) QSPI Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	ON	ON	ON	ON	ON	ON	-	-
SW2	QSPI-B	ON	ON	ON	ON	ON	ON	-	-
SW3	QSPI-C	set(OFF)		-	-	-	-	-	-
SW10	MODESW-A	ON	ON	OFF	OFF	ON	OFF	ON	ON

b) Hyper Flash Mode

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW1	QSPI-A	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW2	QSPI-B	OFF	OFF	OFF	OFF	OFF	OFF	-	-
SW3	QSPI-C		set(ON)	-	-	-	-	-	-
SW10	MODESW-A	ON	ON	OFF	OFF	ON	ON	OFF	ON

4.3 How to write

Please connect R-Car H3 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC.

In case of R-Car H3, Ethernet connector is CN22, serial connector is CN25.

Step 1 connect cable

Connect USB Host connector of Windows Host PC that is virtual COM port to CN25 of R-Car H3 System Evaluation Board with USB cable for displaying console.

Step 2 setting the terminal software

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none

Step 3 write data file to Hyper Flash

A file is written in Hyper Flash in the following procedures.

- Set dip switch “QSPI Mode”.
- Reset board then start mini monitor.
- Set dip switch (Select Hyper Flash)
  SW1: ALL OFF
  SW2: ALL OFF
  SW3: ON
  Note) SW10 does not need to change.
- Execute xls2 command (load program to flash).

```
SALVATOR MiniMonitor V0.05 2015.08.25
 Work Memory SystemRAM (H'E6328000-H'E632FFFF)
>xls2
===== Qspi/HyperFlash writing of Gen3 Board Command ============
Load Program to Spiflash
Writes to any of SPI address.
Please select,FlashMemory.
   1 : QspiFlash        (U5 : S25FS128S)
   2 : QspiFlash Board (CN3: S25FL512S)
   3 : HyperFlash       (SiP internal)
  Select (1-3)>
```

- Select HyperFlash. Input “3”.
- After "SW1 SW2 All OFF! Setting OK? (Push Y key)" is displayed, input "y".
- After "SW3 ON! Setting OK? (Push Y key)" is displayed, input "y".
- After "Please Input Program Top Address" is displayed, input Program Top Address in 4.1 and "Enter".
- After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address in 4.1 and "Enter".
- After "Please send ! ('.' & CR stop load)" is displayed, In case of Tera Term, transmit u-boot-elf.srec by "file -> file transmission(S)".
- If there are some data in writing area, "SPI Data Clear(H'FF) Check :H'00000000-0003FFFF Clear OK?(y/n)" is displayed. Then input "y".
- After "SAVE SPI-FLASH ..... complete!" is displayed, the prompt returns. It means finish.
- Please repeat the xls2 command, if other files are written.
- Power OFF
- Set dip switch to “Hyper Flash Mode”.
  Note) SW1, SW2, and SW3 are already Hyper Flash mode. Please set SW10.

4.4 IPL/Secure write

Please write the file described in Chapter 4.1 to Hyper Flash.

The data file is stored in the $WORK/build/tmp/deploy/images/<board_name> directory.

5.Confirm starting of U-Boot and Linux

Please connect R-Car H3 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC with TFTP and NFS server as Figure 1. Then please confirm normal starting of U-Boot and Linux with following step.

In case of R-Car H3, Ethernet connector is CN22, serial connector is CN25, D-Sub connector is CN15.

Note) This version supports only TFTP boot. Stand-alone boot (booting from flash memory) is supported by future release.

Step 1 setting Linux Host PC

Please install TFTP server and NFS server in Linux Host PC with apt-get command and so on. Please set /etc/xinetd.d/tftp of TFTP server and /etc/exports of NFS server according to your environment.

Step 2 connect cable

Connect USB Host connector of Windows Host PC that is virtual COM port to CN25 of R-Car H3 System Evaluation Board with USB cable for displaying console.

Step 3 setting the terminal software

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port.

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none

Step 4 write U-Boot to Hyper Flash

filename	Program Top Address	Flash Save Address	description
u-boot-elf.srec	0x49000000	0x640000	U-boot

The data file is stored in the $WORK/build/tmp/deploy/images/<board_name> directory.
Refer to Chapter 4.3 Step3 for write procedure.

Step 5 set U-Boot environment variables

Please start U-Boot by board reset. Please set and save environment variable as follows.

```
=> setenv ethaddr xx:xx:xx:xx:xx:xx
=> setenv ipaddr 192.168.0.20
=> setenv serverip 192.168.0.1
=> setenv bootcmd 'tftp 0x48080000 Image;tftp 0x48000000 Image-r8a7795-
salvator-x.dtb;booti 0x48080000 - 0x48000000'
```

Note) This version does not support saveenv command.

Step 6 change the bootargs by U-Boot

To change bootargs which passed to the kernel in boot sequence, please modify it by “setenv bootargs” command of u-boot.

```
=> setenv bootargs 'console=ttySC0,115200 mem=2048M rw root=/dev/nfs
nfsroot= 192.168.0.1:/export/rfs ip=192.168.0.20'
```

Step 7 set file system

Please extract file system (core-image- weston-<supported board name>.tar.bz2). Please export /export directory of NFS server.

```
$ mkdir /export/rfs
$ cd /export/rfs
$ sudo tar xvf core-image-weston(minimal)-<supported board name>.tar.bz2
```

Note) <supported board name> is the following: salvator-x

Note) lib32 file system(lib32-core-image-weston(minimal)-<supported board name>.tar.bz2) must not use it.

Step 8 start Linux

Please execute the following commands manually, without rest.

Note) This version does not support saveenv command. Please do not reset.

```
$ run bootcmd
```

Note) In core-image-weston, please connect input device (keyboard, mouse) to USB port.

6. Exporting Toolchains

Please refer Documents from Yocto Project to export Toolchains such as
http://www.yoctoproject.org/docs/current/adt-manual/adt-manual.html.
And please use build target of bitbake as “core-image-weston-sdk -c populate_sdk” to generate package.

Note) When you use “ld” directly but not via gcc (in case of building Kernel, Driver or u-boot), please disable LDFLAGS with ‘export LDFLAGS=”” ’.

Note) Please do not use same shell environment to other compilation/debugging purpose (also make menuconfig of linux kernel, e.g.) but cross compilation for R-Car H3 which shell environment with “source” command to setup environment variables for the SDK. Because some environment variables for cross compilation interferes execution of other tools on the same shell environment.

Example of instruction:

In following examples, it’s assumed that it’s already extracted and prepared recipe environment such as in the instructions of Section 3 (must done just before execution of bitbake, at least). You may reuse $WORK/build while you reuse same configuration after executing bitbake as in Section 3 for this purpose.

Step 1 configure architectures of Host PC which are installed this toolchain

When users of toolchain execute different architecture of Host PC other than build environment (x86_64 and i686), please modify SDK_MACHINE description on $WORK/build/conf/local.conf

On $WORK/build/conf/local.conf

```
# This variable specified the architecture to build SDK/ADT items for and means
# you can build the SDK packages for architectures other than the machine you are
# running the build on (i.e. building i686 packages on an x86_64 host._
# Supported values are i686 and x86_64
#SDKMACHINE ?= "i686"
SDKMACHINE ?= "x86_64"
# ← When toolchain used in 32bit Host PC but building toolchain with x86_64
```

Note) Ubuntu 14.04 64bit is recommended.

Step 2 building toolchain package with bitbake

```
$ cd $WORK/build
(For 64bit SDK)
$ bitbake core-image-weston-sdk -c populate_sdk
$ cp tmp/deploy/sdk/poky-glibc-x86_64(or i686)-core-image-weston-sdk-aarch64-toolchain-1.8.1.sh (shared dir. where able to access from each Host PCs)

(For 32bit SDK)
$ bitbake lib32-core-image-weston-sdk -c populate_sdk
$ cp tmp/deploy/sdk/poky-glibc-x86_64(or i686)-lib32-core-image-weston-sdk- aarch64-toolchain-1.8.1.sh (shared dir. where able to access from each Host)
```

Note) Poky doesn't well support toolchain SDK for multilib. The 64bit SDK only supports for 64bit application. Similarly, the 32bit SDK only supports for 32bit application.

Note) Please perform "bitbake core-image-minimal -c populate_sdk" in BSP-only.

Step 3 Install toolchain on each Host PCs

```
(For 64bit SDK)
$ sudo  (shared dir. where able to access from each Host PCs)/poky-glibc-x86_64(or i686)-core-image-weston- aarch64-toolchain-1.8.1.sh

(For 32bit SDK)
$ sudo  (shared dir. where able to access from each Host PCs)/poky-glibc-x86_64(or i686)-lib32-core-image-weston- aarch64-toolchain-1.8.1.sh

 [sudo] password for (INSTALL person): (password of your account)
Enter target directory for SDK (default: /opt/poky/poky1.8.1): (just a return)
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

Step 4 setup environment variables for each compilation on each Host PCs

Please setup environment variables as follows or integrate set-up sequence into your build script or Makefile.

```
$ cd (Your working directory)
(For 64bit SDK)
$ source /opt/poky/1.8.1/environment-setup-aarch64-poky-linux
(For 32bit SDK)
$ source /opt/poky/1.8.1/environment-setup-armv7ahf-vfp-pokymllib32-
linux-gnueabi
$ export LDFLAGS=""
$ $CC  (Your source code).c .....
```

Note) In 64bit SDK, 32bit environment script (environment-setup-armv7ahf-vfp-pokymllib32-linux-gnueabi) does not work.

Note) In 32bit SDK, 64bit environment script(environment-setup-aarch64-poky-linux) does not work.

7. Memory map

Following Figure 2 and Figure 3 Figure 4 show memory map of this R-Car H3 Linux BSP package.

Note)

- The volume of SDRAM is total 4GB.
- 2GB from 0x00_4000_0000 to 0x00_BFFF_FFFF is a shadow area from 0x04_0000_0000 to 0x04_7FFF_FFFF.
- 63MB from 0x00_43F0_0000 to 0x00_47DF_FFFF is used as a secure region. It doesn't allow U-Boot and kernel to access that region.

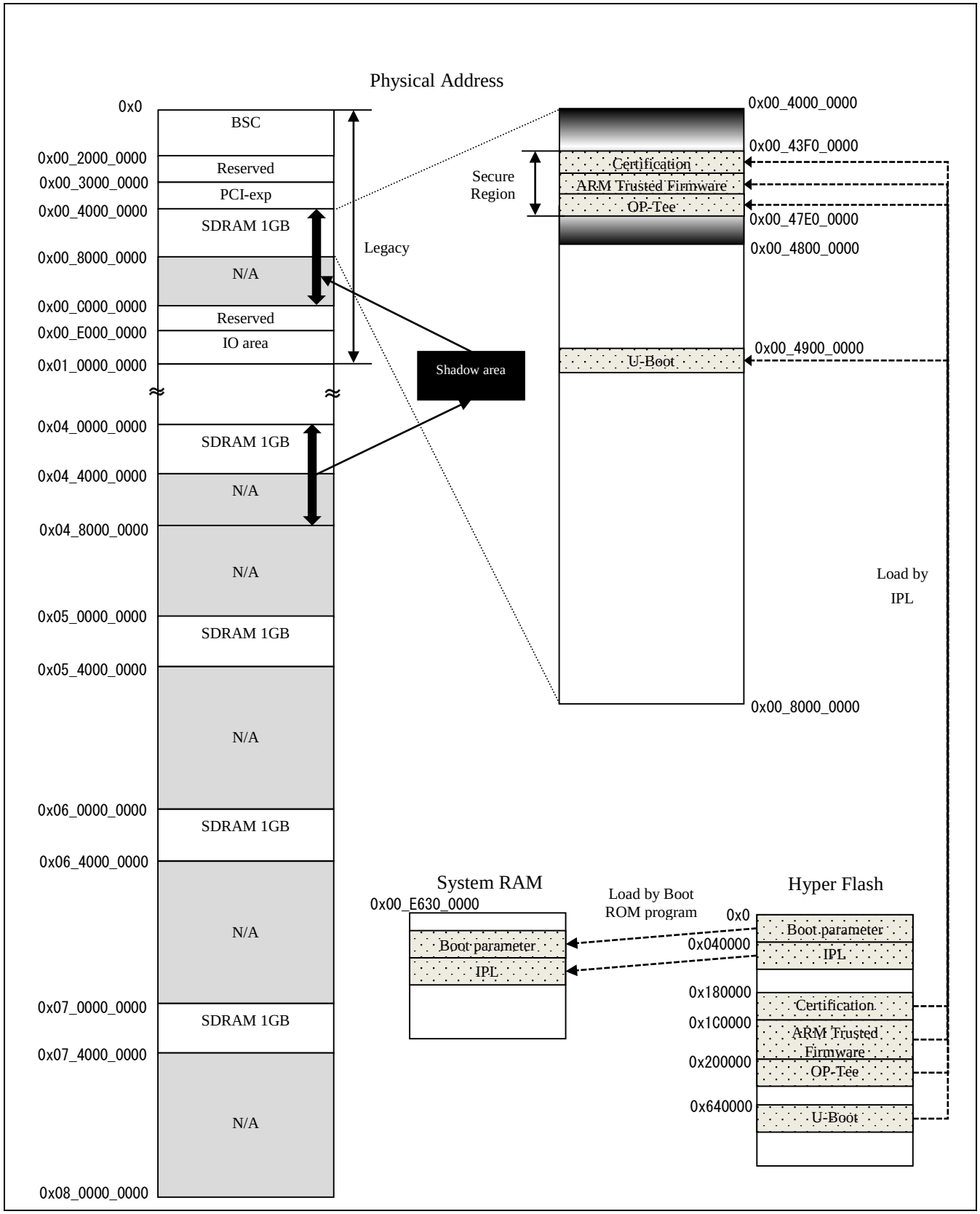

Figure 2 H3 memory map (Boot)

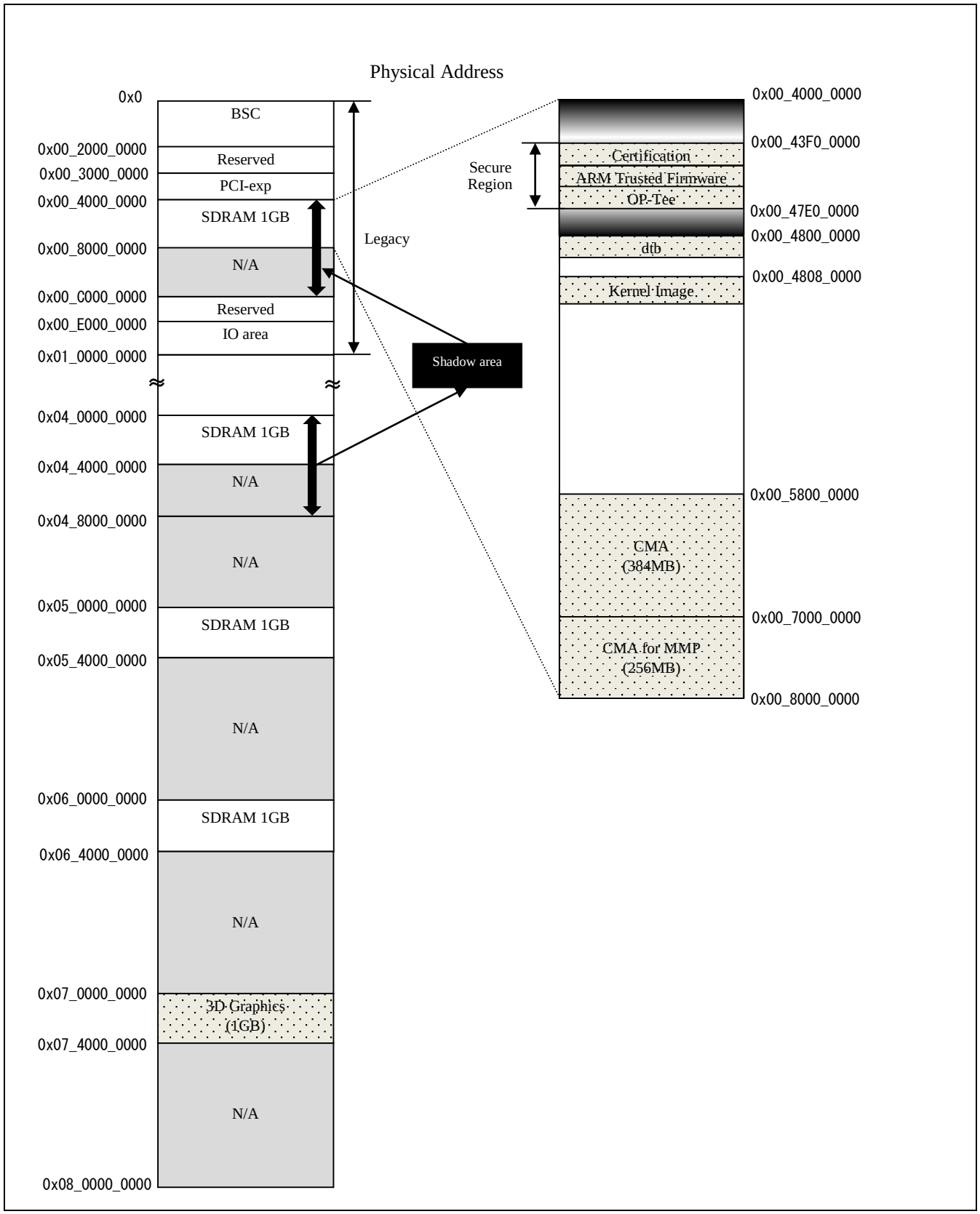

Figure 3 H3 memory map (Linux)

Note)

- Kernel region is assigned by Kernel device tree arch/arm64/boot/dts/renesas/xxx.dts and specified with “mem” option in bootargs.
- Because of specifying mem=2048M, Kernel region consists of three parts:
  - 896MB from 0x00_4800_0000 to 0x00_7FFF_FFFF
  - 1GB from 0x05_0000_0000 to 0x05_3FFF_FFFF
  - 128MB from 0x06_0000_0000 to 0x06_07FF_FFFF (the remaining 896 MB from 0x06_0800_0000 is un-used.)
- 1GB from 0x07_0000_0000 to 0x07_3FFF_FFFF is used as 3D Graphics region.
- Kernel uses 4KB page size (VA_BITS=39) and 3 levels of translation tables. Both regions of User and Kernel are 512GB. Refer to Documentation/arm64/memory.txt.
- Detail information about kernel memory map in virtual address space, refer to User’s manual of Kernel.
- There are two types of CMA regions:
  - Default CMA region: It is for kernel and general drivers.

    This region is defined in device tree (arch/arm64/boot/dts/renesas/xxxx.dts).

```
linux,cma {
   compatible = "shared-dma-pool";
   reusable;
   reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
   linux,cma-default;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```

    Note)

    - The CMA region can be adjusted by changing the start address and the size.
    - Should take care of the lack of memory allocated by kernel and general drivers when reducing the region size.
  - CMA region for MMP: It is for multimedia package (specific H/Ws).

    This region is defined in device tree (arch/arm64/boot/dts/renesas/xxxx.dts).

```
linux,multimedia {
   compatible = "shared-dma-pool";
   reusable;
   reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;
};

0xXXXXXXXX is start address of CMA region.
0xYYYYYYYY is size of CMA region.
```

    Note)

    - Refer to User’s manual of Memory Manager in order to change CMA for MMP region.

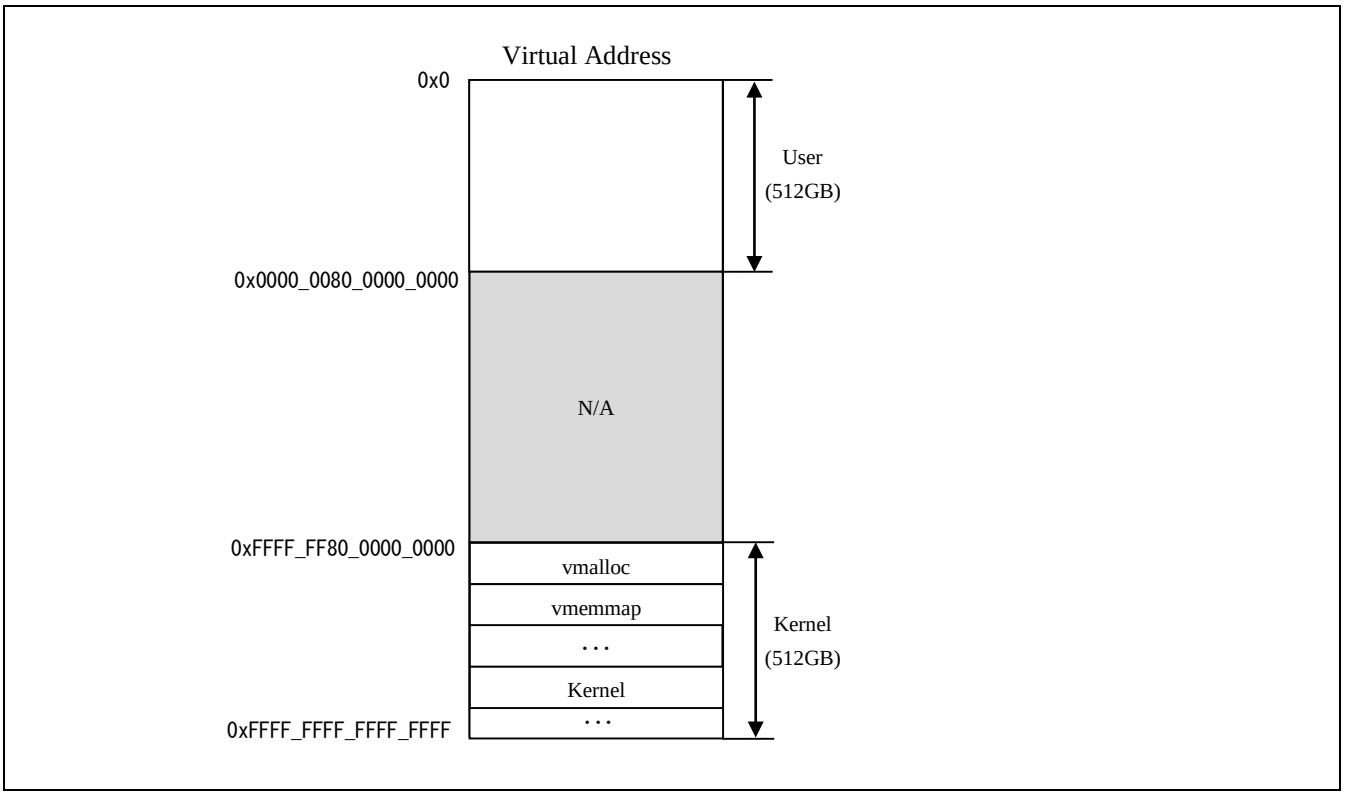

Figure 4 H3 memory map (Virtual)

Note)

- Kernel uses 4KB page size (VA_BITS=39) and 3 levels of translation tables. Both regions of User and Kernel are 512GB. Refer to Documentation/arm64/memory.txt.

- Detail information about kernel memory map in virtual address space, refer to User's manual of Kernel.

8. U-Boot command

Please refer to U-Boot user's manual about available U-boot command for R-Car H3 Linux BSP. The help or "?" command shows U-Boot command list, but be careful that it includes some unsupported command.

REVISION HISTORY	Linux Interface Specification Yocto recipe Start-Up Guide User's Manual: Software

Rev.	Date	Description	
		Page	Summary
0.10	Sep 07. 2015	—	First Edition for Yocto Recipe Package with YP1.8, based on Start-Up Guide Rev 1.30 for Kernel 3.4 LTSI
0.20	Sep 30. 2015	2	1 Modified URL of U-boot and Linux source code.
		6	3 Deleted description about core-image-x11
		6	3 step2 Changed git repository URL
		9	3.2 Added note about not support 3D graphics package 3.2 step5,7 Changed command process
		12	4.1 Changed u-boot filename. u-boot.srec -> u-boot-elf.srec
		21, 22	7 Modified memory map (Fig2, 3)
0.30	Oct 30 2015	2	1.1 Modified document version
		3	2.1 Updated table1
		4	2.1 Updated Figure1
		7	3 Modified checkout commit id 3 step4 Updated apply patch command
		8-12	3.1 Added description build procedure for MMP and 3D Graphics
		14	4.4 Deleted note
		18	5 step6 Added mem=2048 to bootargs 5 step8 Added note
		19-21	6 Modified SDK revision 1.8 -> 1.8.1
		22	7 Modified the description of the address range. 0x00_C000_0000 -> 0x00_BFFF_FFFF 7 Modified the address range and size of a secure region.
		23	7 Modified the address of System RAM. 0x00_E630_0000 -> 0x00_E632_0000 7 Modified the load destination of Certification in HyperFlash. System RAM -> SDRAM
		24, 25	7 Modified memory map (Fig3) and update note about CMA 7 Added note about specifying mem=2048M 7 Added note about 3D graphics region
		26	7 Separated virtual memory map (Fig4) and note about virtual memory

Linux Interface Specification Yocto recipe Start-Up Guide
User's Manual: Software

Publication Date: Rev.0.30 Oct. 30, 2015

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 America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.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, P. R. China 200333
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, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2015 Renesas Electronics Corporation. All rights reserved.

Colophon 4.0

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

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

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

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

© 2015 Renesas Electronics Corporation. All rights reserved.
Colophon 3.0

Linux Interface Specification
Yocto recipe Start-Up Guide

Renesas Electronics Corporation