User's Manual

64

QNX 7.0 BSP
For R-Car M3 Platforms

User's Manual: Software

R-Car M3

— Preliminary —

Renesas Electronics
www.renesas.com

Rev.1.00 Mar 2017

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - “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.
   - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anticrime systems; safety equipment; and medical equipment not specifically designed for life support.
   - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system 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. 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.
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.

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.
- QNX and Neutrino are registered trademarks of QNX Software Systems Ltd. in certain jurisdictions.
- 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 M3 processor.

[Purpose]

This manual is intended to give users an understanding of building, booting up QNX 7.0 OS on R-Car M3 processor and verifying device drivers and to serve as a reference for developing software for systems that use this BSP.

[How to Read This Manual]

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, microcontrollers, Linux, and QNX.

→ Read this manual in the order of the **CONTENTS**.

To understand the functions of a device driver for R-Car M3

→ See the **R-Car M3 driver Manual**.

To know the electrical specifications of processor for R-Car M3

→ See the **R-Car 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

LIST OF FIGURES

LIST OF TABLES

1. Overview

1.1 Features

This document describes the procedures for reproducing QNX 7.0 image for R-Car M3 platforms and testing drivers on R-Car M3 platforms (Salvator-X and StarterKit boards).

In the case of V.0.0.1 BSP, below drivers/libraries/utilities are supported.

1) Startup
2) Serial driver
3) Audio driver
4) Network driver
5) I2C driver
6) SD Card driver
7) DMA libraries
8) MMC driver
9) SATA driver
10) Display driver
11) USB 2.0 function driver
12) USB 2.0, USB3.0 host driver
13) Hyper Flash driver
14) Libraries: DMA library, Video Capture library, imr
15) Utilities: camera_ctrl, i2c-control
16) Support: resource_seed, wdtkick-rcar

1.2 Scope

This document scope applies to the evaluation of reproducing QNX 7.0 image and testing QNX BSP drivers for R-Car M3 platforms.

1.3 Target System

1. Target platform: R-Car M3 platforms (Salvator-X and StarterKit boards) by Renesas Electronics Corporation.
2. Target software: QNX SDP 7.0.

1.4 List of Abbreviations and Acronym

Abbreviation	Full Form
BSP	Board Support Package
DMA	Direct Memory Access
SDP	Software Development Package
I2C	Inter-Integrated Circuit
SD	Secure Digital
eMMC	embedded Multi Media Card

1.5 Environmental Requirement

Table 1-1 Environment Requirement

Equipment	Explanation
Linux Host PC (Optional)	Ubuntu 12.04 LTS is recommended as OS It is used as building and debugging environment It is used as TFTP server
Windows Host PC	Windows 7 or XP are recommended as OS It is used as building and debugging environment. It is used as TFTP server Terminal software and VCP driver are executed.
QNX 7.0 SDP SP1	Install QNX 7.0 SDP SP1 on Windows host if Windows host is used to build. Install QNX 7.0 SDP SP1 on Linux host if Linux host is used to build. Please contact to QSSL
Terminal software	Please use one of below software. 1) Hyperterm (included in Windows XP) 2) TeraTerm (prefer latest version) 3) Terminal (included in QNX Momentics IDE)
Virtual Com Port driver	Please install in Windows Host PC. Execute CP210x_VCP_Win_XP_S2K3_Vista_7.exe to install USB to COM driver. USB becomes virtual COM port on terminal software. Please connect to CN25 (Serial-USB Bridge CP2102) on M3 Solution Board. (Available at http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx)
TFTP server software	It is used to download QNX image via U-boot. It is reference to use TFTP Server built in QNX Momentics IDE.

2. Quick Startup Guide

This section describes steps needed to run QNX image on R-Car M3 reference platforms.

2.1 Requisite

2.1.1 Hardware

1) SD card (or MicroSD Card)
2) Micro-USB cable for debug
3) HDMI monitor and HDMI cable, preferably Lilliput monitor
4) Speaker set (optional)
5) Microphone (optional)
6) USB memory stick (optional)
7) Ethernet cable
8) Hard disk (optional)

2.1.2 Software

1) QNX 7.0 binary image ifs-rcarm3-<board_name>.bin (ifs-rcarm3-starterkit.bin or ifs-rcarm3-salvatorx.bin). If not available, please follow section 3 for how to compile and build it.
2) Bootparam, Loader, ARM Trusted Firmware and U-Boot (refer to table 5.1) are already flashed in Hyper Flash. If not available, please follow section 5.2 & 5.3 to build and write those boot files to Hyper Flash.

2.2 Steps to run QNX 7.0 OS image

1) Prepare a SD Card with FAT32 format then copy ifs-rcarm3-<board_name>.bin image to it.
2) Insert the SD Card into CN13 or CN14 on Salvator-X board (or CN6 on StarterKit board). Then connect necessary hardware e.g., micro-USB cable, HDMI monitor, speakers, etc. For detail information, please refer section 5.1 and section 7.
3) Check the DIP switch settings are as per DIP switch settings in section 5.3.2 for Salvator-X board (or table 6-1 for StarterKit board).
4) Start serial terminal program. For detailed information, please refer section 5.1 for Salvator-X board (or section 6.1 for StarterKit board).
5) Power ON the board. From serial terminal program on PC, you will see loader, ARM Trusted Firmware and u-boot be running as below:

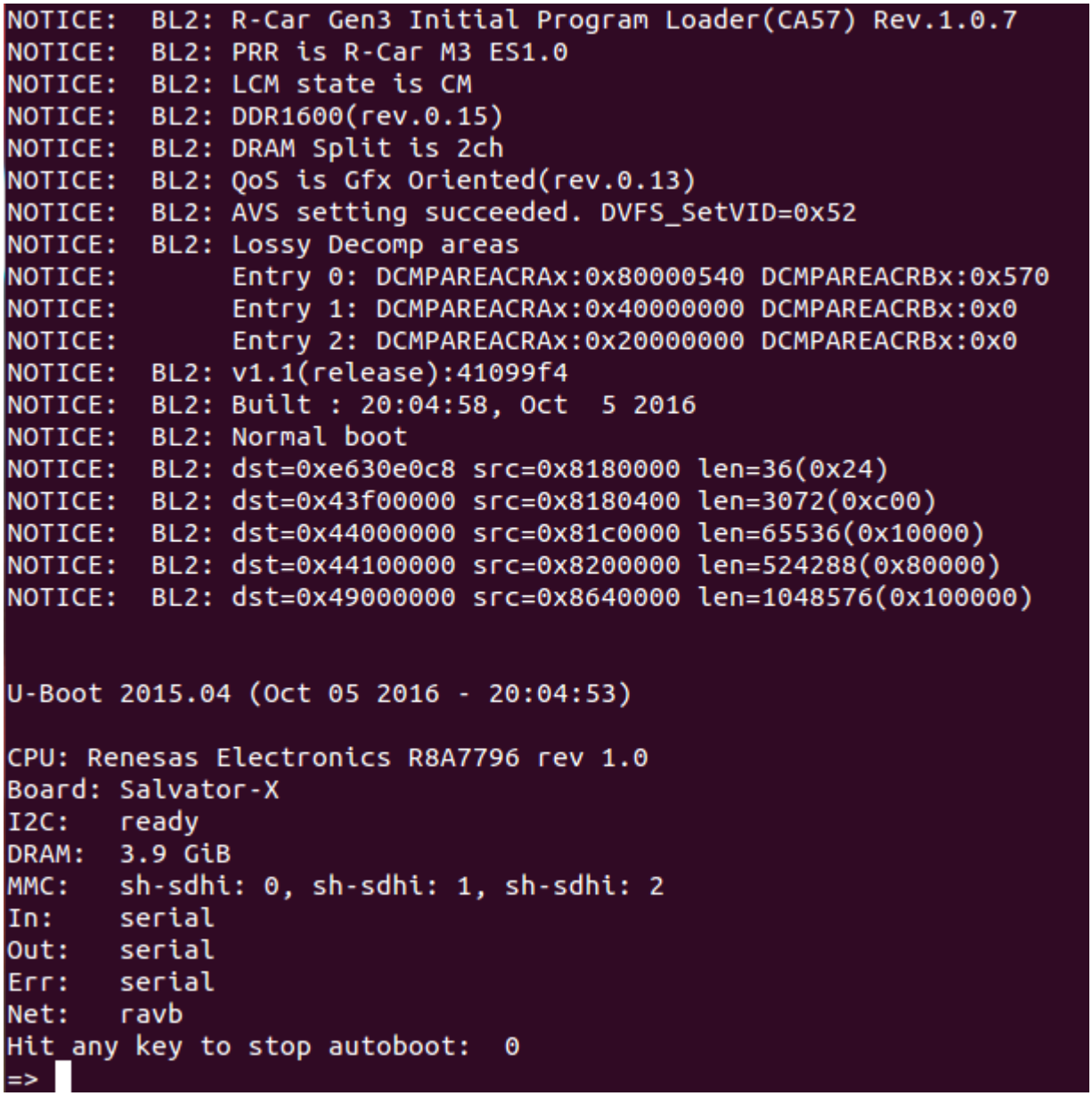

Figure 2-1: Serial log after power-on

6) Quickly hit any key on u-boot console then u-boot is ready for input commands. Quickly hitting time is normally within 3 seconds but depends on bootdelay value.

7) Run below commands to set and save u-boot environment variables. Those commands will download and run QNX image from SD card.

   => setenv load_run_qnx_sd 'fatload mmc [n] 0x40100000 ifs-rcarm3-<board_name>.bin; go 0x40100000'

   (CN13 n=0, CN14 n=2 on Salvator-X board, or CN6 n=0 on StarterKit board)

   => setenv bootcmd 'run load_run_qnx_sd'

   => saveenv

8) Reset the board and now u-boot copies QNX from SD card to RAM and runs it.

9) After QNX boots up, you will see the log in serial as below:

```
System page at phys:0000000040010000 user:ffffff8040202000 kern:ffffff8040201000
Starting next program at vffffff8060067190
MMFLAGS=1
Welcome to QNX Neutrino 7.0 on the R-CarM3 Salvator-X Board
Starting slogger and pipe servers...
Starting syslog...
Starting the session...
Starting watchdog ...
Starting Serial driver...
Starting I2C driver ...
Starting USB Host driver ...
Starting SPI Flash driver...
Starting Network driver...
Starting DHCP...
Starting Audio driver ...
Starting SDHI memory card driver ...
Starting USB2.0 Function driver as USB NCM device...
Starting MMC memory flash driver ...
Starting Display driver...
#
```

Figure 2-2 Boot up using u-boot commands

10) HDMI monitor displays as below:

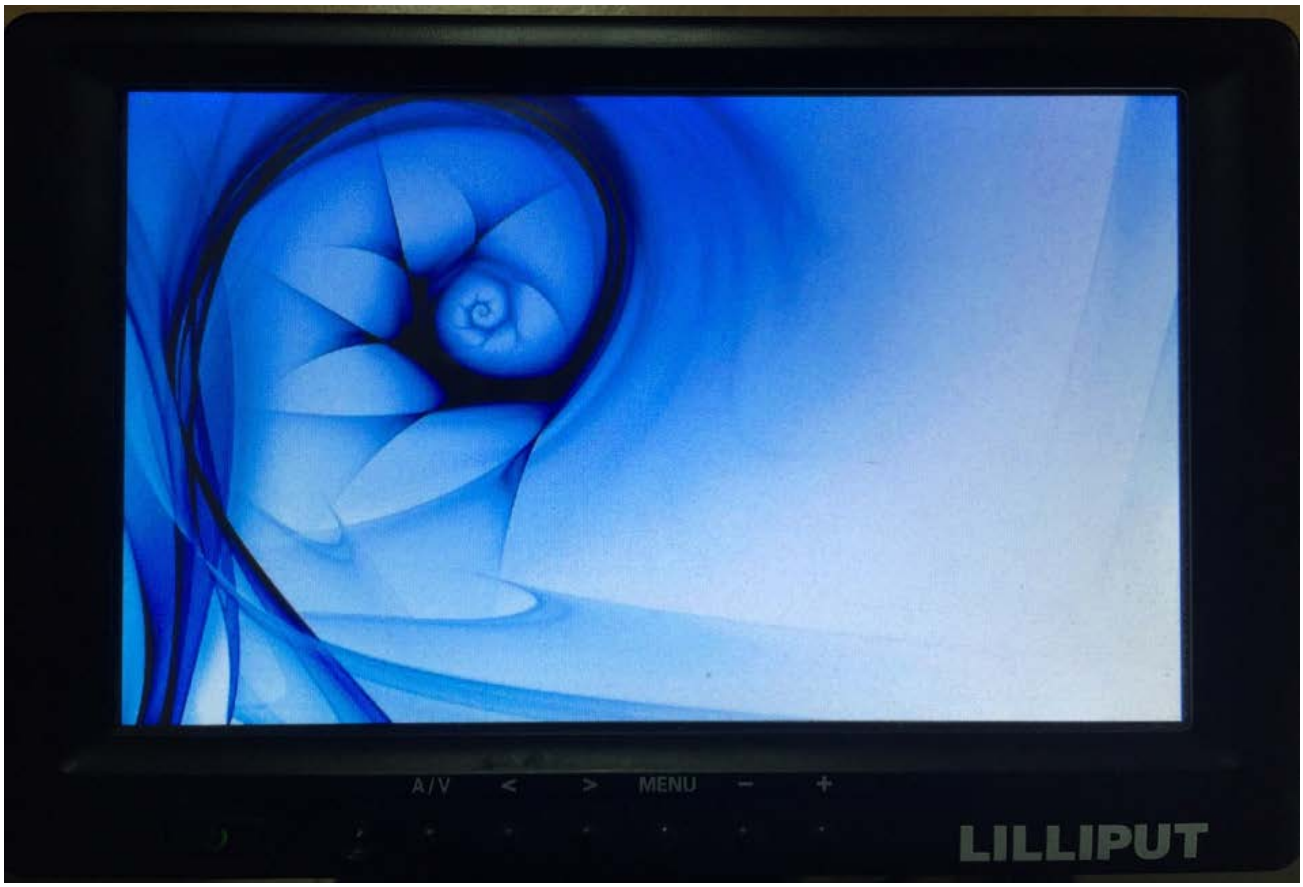

Figure 2-3 HDMI monitor displays

Please look into the section 5 and section 6 of this setup manual to run QNX image using various other methods

3. Building QNX 7.0 OS Image

3.1 Requisite

3.1.1 For Linux Host

1. Extract QNX 7.0 BSP for R-Car M3 board (filename: R_CarM3_qnx7_bsp_y_m_d.zip) to a folder on Linux Host PC (e.g., /tmp/R_CarM3_qnx7_bsp_y_m_d)
   Note: y, m, d is the release date of the current BSP you're using.
2. QNX SDP 7 package installed on Linux Host PC.

3.1.2 For Windows Host

1. Extract QNX 7.0 BSP for R-Car M3 board (filename: R_CarM3_qnx7_bsp_y_m_d.zip) to a folder on Windows Host PC (e.g., C:/R_CarM3_qnx7_bsp_y_m_d)
   Note: y, m, d is the release date of the current BSP you're using.
2. QNX SDP 700 package installed on Windows Host PC.

3.2 Build Mode

3.2.1 Single CPU Mode

This mode is used for running QNX image on single CPU mode.

Open "build" file at …/src/hardware/startup/boards/ rcarm3/<board_name>/build, set number of CPUs argument to 1 in startup command.

startup-rcarm3-<board_name> -P <N> -W #(Note: N = number of CPUs. It MUST be 1 for Single CPU mode)

3.2.2 Multi CPU mode

This mode is used for running QNX image on multi CPU mode (maximum 4 CPUs)

Open "build" file at …/src/hardware/startup/boards/rcarm3/<board_name>/build, set number of CPUs argument to <N> in startup command if you want to boot up QNX image on <N> CPUs. Currently, maximum of CPUs can be set is 4.

startup-rcarm3-<board_name> -P <N> -W #(Note: N = number of CPUs. It may be 1, 2, 3 or 4)

Currently, R-Car M3 SoC supports maximum 2 cores Cortex-A57 and 4 cores Cortex-A53.

3.2.3 Specifying unique MAC address for QNX network driver

This mode is used for running QNX image with specified unique MAC address.

Open "build" file at ../R_CarM3_qnx7_bsp_y_m_d/src/hardware/startup/boards/ rcarm3/<board_name>/build, and change the network driver launch command as below:

io-pkt-v6-hc -dravb mac=2e090a00830c, -ptcpip pkt_typed_mem=below4G &

Set MAC address for QNX image by changing "mac" argument with unique value.

3.3 Build BSP

For Linux environment

1. Make sure build configuration file was already configured as section 3.2.
2. To generate the QNX image for R-CarM3 boards, run the following commands from the terminal console on Linux host.

   $ cd <QNX_BSP_Root_Directory>

   If you are not root user, then run: $sudo bash

   $ source <QNX_700_Installed_Directory>/qnx700-env.sh

   (Default is /opt/qnx700/qnx700-env.sh)

   $ make clean; make all

For Windows environment

1. Make sure build configuration file was already configured as section 3.2.
2. To generate the QNX image for R-Car M3 boards, run the following commands from the terminal console on Windows host.
   - Move to QNX 7.0 Installed Directory and set up environment by running qnx700-env.bat file

     C:\qnx700\qnx700-env.bat
   - Change to BSP root directory and make QNX image

     C:\ R_CarM3_qnx7_bsp_y_m_d>bash

     # make clean; make

4. Memory Map

The following figures show memory map of R-Car M3 QNX 7.0 BSP package.

Note:

- The volume of SDRAM is total 4GB.
- The following regions are used as secure regions. It doesn't allow U-Boot and kernel to access those regions:
  - ✓ 63MB from 0x00_43F0_0000 to 0x00_47DF_FFFF is used as a secure region.
  - ✓ 8KB from 0x00_E630_0000 to 0x00_E630_1FFF in System RAM.

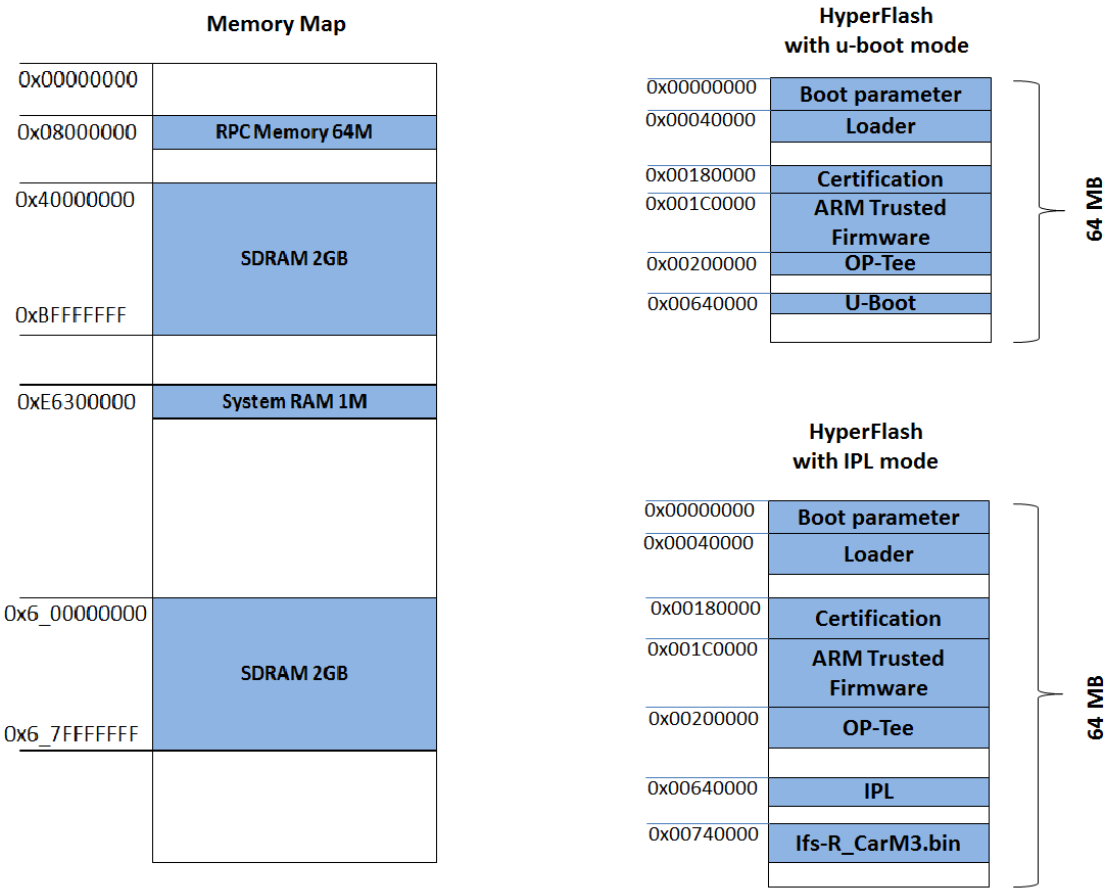

Figure 4-1 Memory map for useable storages

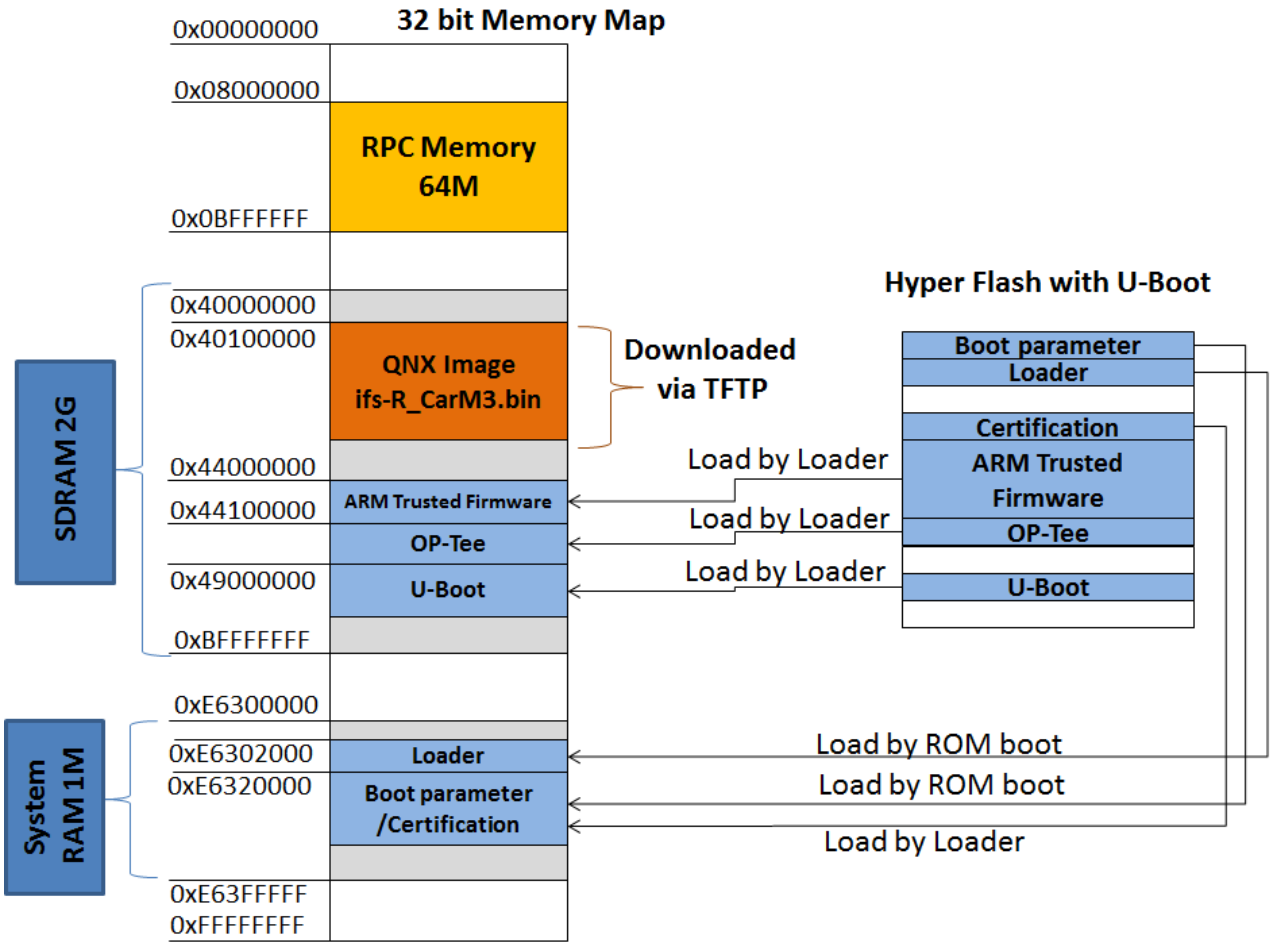

Figure 4-2 Boot Memory map (u-boot boot mode)

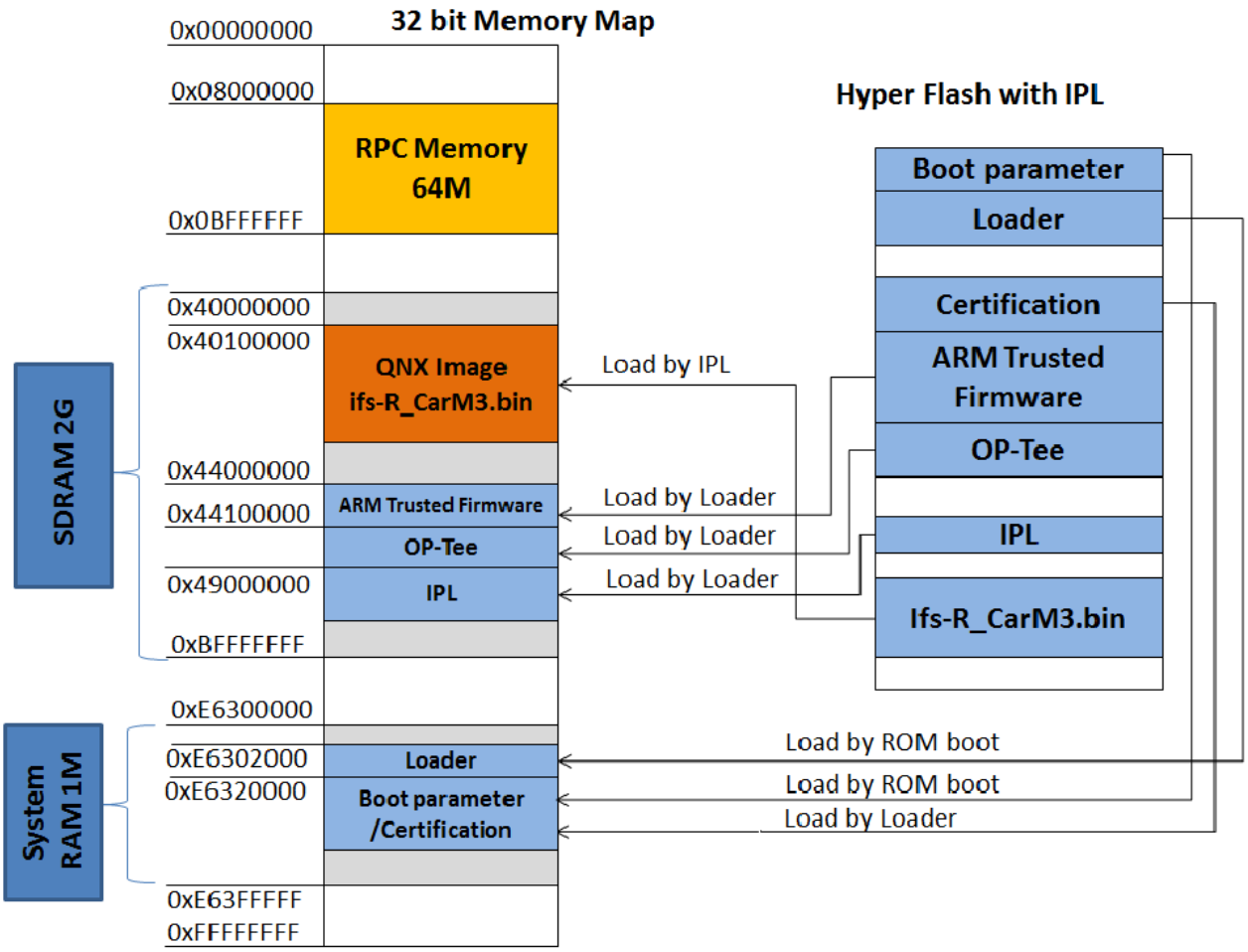

Figure 4-3 Boot memory map (IPL boot mode)

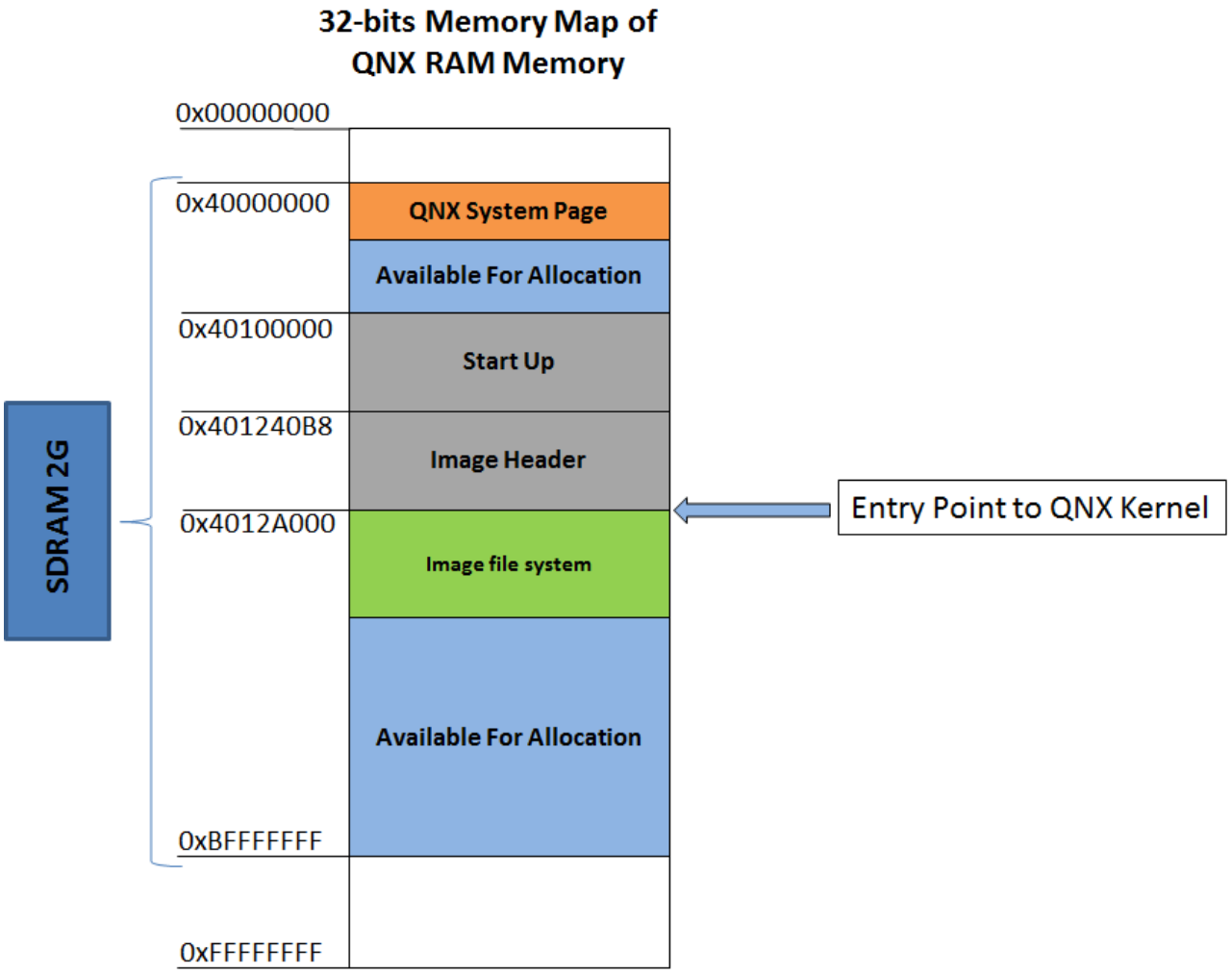

Figure 4-4 QNX RAM Memory map (when QNX is running)

5. Write QNX Image To R-Car M3 Salvator-X board

5.1 Hardware setup

— Start terminal console program on PC host (e.g., Tera Term). Assume that we're using COM1 port.

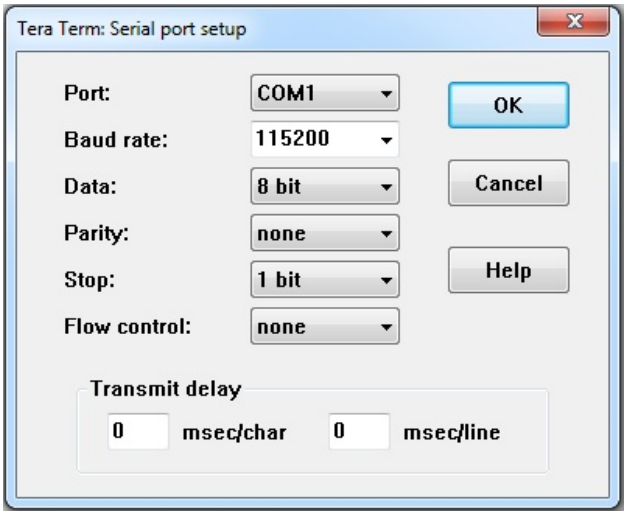

Figure 5-1 Serial port setup

— Connect micro-USB cable to micro USB port (CN25) and to PC.

— Connect Network connector (CN22) to your network.

— Recommended Environment

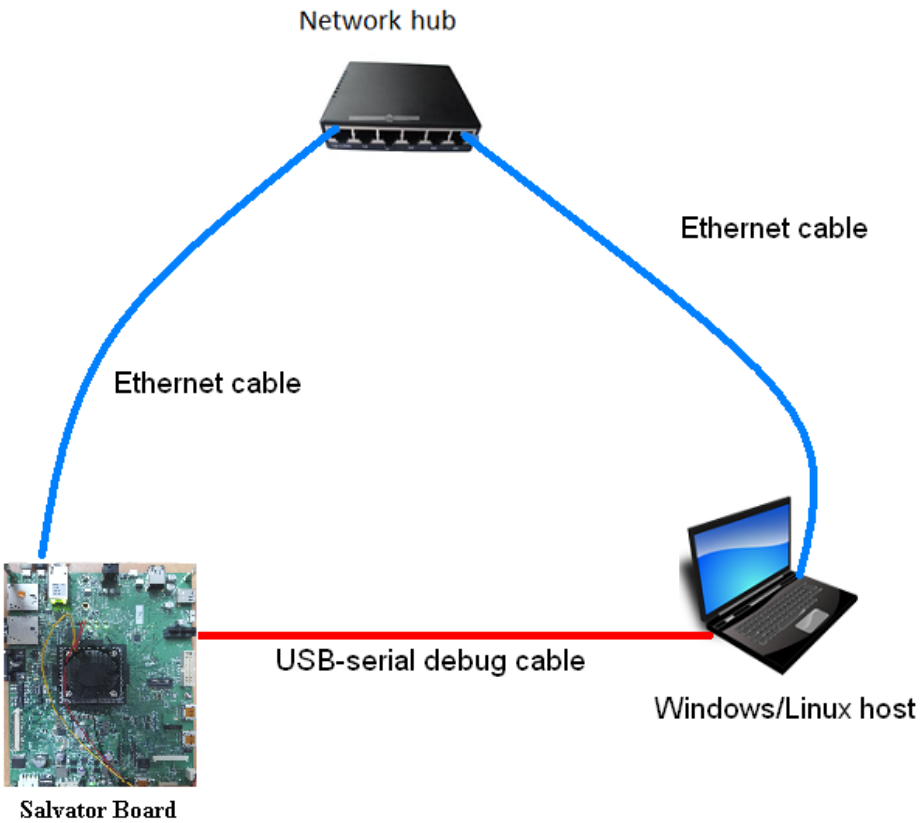

Figure 5-2 Recommended Environment setup

5.2 Write Bootparam, Loader and U-Boot to Hyper Flash

5.2.1 Boot files information

Table 5-1 Bootloader files

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 contact a Renesas Electronics sales office about *bootparam_sa0.srec* file.

5.2.2 Dip-Switch

— **QSPI Mode**

Table 5-2 QSPI mode setting

Switch Number	Switch Name	SoC version	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	-	OFF		-	-	-	-	-	-
SW10	MODESW-A	ES 1.0	ON	ON	OFF	ON	ON	OFF	ON	ON

— **Hyper Flash Mode**

Table 5-3 Hyper Flash mode setting

Switch Number	Switch name	SoC version	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	-	ON		-	-	-	-	-	-
SW10	MODESW-A	ES 1.0	ON	ON	ON	ON	ON	ON	OFF	OFF

— **AArch64 Mode**

Table 5.4 AArch64 mode setting

Switch Number	Switch name	SoC version	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
SW12	MODESW-C	ES 1.0	OFF	ON	ON	ON	ON	ON	ON	ON

5.2.3 Write Bootparam, Loader and U-Boot to Hyper Flash

The following procedure is used to write Bootparam, Loader and U-Boot to Hyper Flash on R-Car M3 Salvator-X board.

Set DIP switch “QSPI” & “AArch64” mode (refer to table 5.2 & 5.4).

Reset board then start Mini Monitor.

Set Dip switch (Select Hyper Flash)

SW1: ALL OFF

SW2: ALL OFF

SW3: ON

SW10: does not need to change

Execute "xls2" command (load program to flash).

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 of bootparam_sa0.srec in table 5.1 (H’E6320000) and "Enter".

After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address of bootparam_sa0.srec in table 5.1 (e.g., H’000000) and "Enter".

After "Please send ! ('.' & CR stop load)" is displayed, In case of Tera Term, transmit boot parameter file (*bootparam_sa0.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 to write other files to Hyper Flash.

Power OFF

Set dip switch to “Hyper Flash” and “AArch64” Mode (refer to table 5.3 & 5.4).

5.3 Download QNX image to RAM

5.3.1 Scope

This section scope applies for downloading QNX image to RAM on board using u-boot TFTP command.

5.3.2 Requisite

Host PC installed a TFTP server (such as tftpd32 from http://tftpd32.jounin.net). Or Linux host configured as TFTP server.

Refer to section 3.3 “Build the BSP” to configure and build the BSP.

Copy “ifs-rcarm3-salvatorx.bin” to TFTP server folder on Host PC.

Launch TFTP server on Windows Host PC, select path to “ifs-rcarm3-salvatorx.bin” image file.

Or Start TFTP server on Linux Host, if TFTP server on Linux host is used.

Set dip switch to “Hyper Flash” and AArch64” mode (refer to table 5.3 & 5.4).

5.3.3 Procedures for writing OS image to RAM using U-boot commands

```
NOTICE:  BL2: R-Car Gen3 Initial Program Loader(CA57) Rev.1.0.7
NOTICE:  BL2: PRR is R-Car M3 ES1.0
NOTICE:  BL2: LCM state is CM
NOTICE:  BL2: DDR1600(rev.0.15)
NOTICE:  BL2: DRAM Split is 2ch
NOTICE:  BL2: QoS is Gfx Oriented(rev.0.13)
NOTICE:  BL2: AVS setting succeeded. DVFS_SetVID=0x52
NOTICE:  BL2: Lossy Decomp areas
NOTICE:       Entry 0: DCMPAREACRAx:0x80000540 DCMPAREACRBx:0x570
NOTICE:       Entry 1: DCMPAREACRAx:0x40000000 DCMPAREACRBx:0x0
NOTICE:       Entry 2: DCMPAREACRAx:0x20000000 DCMPAREACRBx:0x0
NOTICE:  BL2: v1.1(release):41099f4
NOTICE:  BL2: Built : 20:04:58, Oct  5 2016
NOTICE:  BL2: Normal boot
NOTICE:  BL2: dst=0xe630e0c8 src=0x8180000 len=36(0x24)
NOTICE:  BL2: dst=0x43f00000 src=0x8180400 len=3072(0xc00)
NOTICE:  BL2: dst=0x44000000 src=0x81c0000 len=65536(0x10000)
NOTICE:  BL2: dst=0x44100000 src=0x8200000 len=524288(0x80000)
NOTICE:  BL2: dst=0x49000000 src=0x8640000 len=1048576(0x100000)


U-Boot 2015.04 (Oct 05 2016 - 20:04:53)

CPU: Renesas Electronics R8A7796 rev 1.0
Board: Salvator-X
I2C:   ready
DRAM:  3.9 GiB
MMC:   sh-sdhi: 0, sh-sdhi: 1, sh-sdhi: 2
In:    serial
Out:   serial
Err:   serial
Net:   ravb
Hit any key to stop autoboot:  0
=>
```

Figure 5-3 U-boot

Reset R-Car M3 Salvator-X board.

U-boot will come up. Hit any key to abort autoboot.

Set environment variables:

- Set MAC address for R-Car M3 Salvator-X board

  ```
  =>set ethaddr 2e:09:0a:00:83:0c (depend on the sticker on Ethernet port CN22)
  ```

- Set the server IP (IP address of Windows PC run TFTP server) and client IP (the IP address of R-Car M3 Salvator-X board):

  ```
  =>setenv serverip <serverip>
  =>setenv ipaddr <boardip>
  ```

- Set the image file name to be downloaded

```
=>setenv bootfile ifs-rcarm3-salvatorx.bin
```

- Set the start address of RAM to load image

```
=>setenv loadaddr 0x40100000
```

- Download the QNX image to RAM

```
=>tftp
```

- Start QNX image

```
=>go 0x40100000
```

QNX 7.0 will start up on the board.

```
System page at phys:0000000040010000 user:ffffff8040202000 kern:ffffff8040201000
Starting next program at vffffff8060067190
MMFLAGS=1
Welcome to QNX Neutrino 7.0 on the R-CarM3 Salvator-X Board
Starting slogger and pipe servers...
Starting syslog...
Starting the session...
Starting watchdog ...
Starting Serial driver...
Starting I2C driver ...
Starting USB Host driver ...
Starting SPI Flash driver...
Starting Network driver...
Starting DHCP...
Starting Audio driver ...
Starting SDHI memory card driver ...
Starting USB2.0 Function driver as USB NCM device...
Starting MMC memory flash driver ...
Starting Display driver...
#
```

Figure 5-4 QNX 7.0 boot up log

6. Write QNX Image To R-Car M3 StarterKit Board

6.1 Hardware setup

— Start terminal console program on PC host (e.g., Tera Term). Assume that we're using COM1 port.

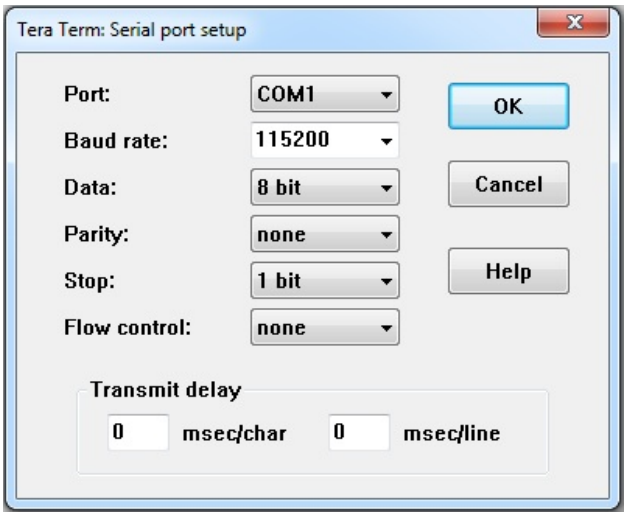

Figure 6-1 Serial port setup

— Connect micro-USB cable to micro USB port (CN12) and to PC.

— Connect Network connector (CN7) to your network.

— Recommended Environment

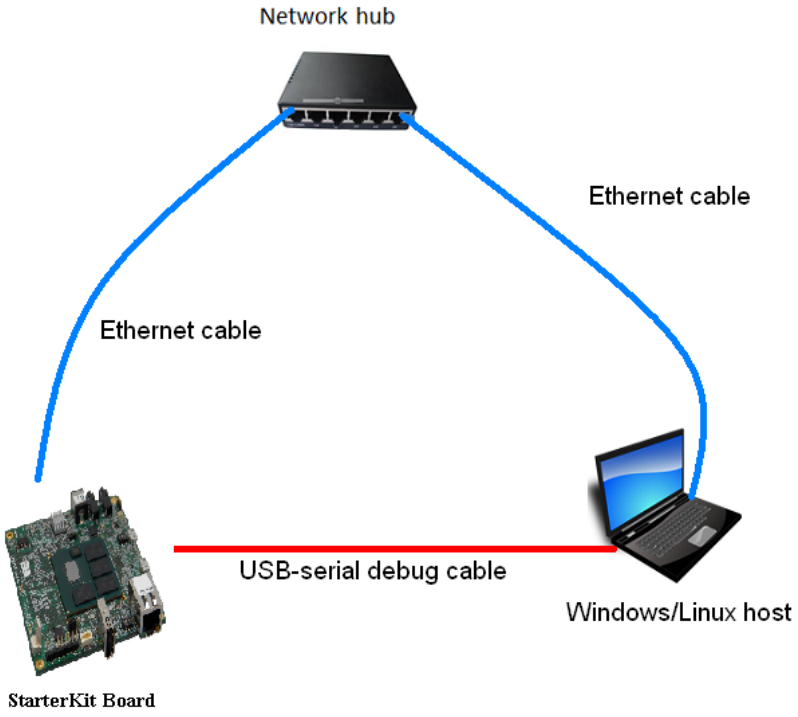

Figure 6-2 Recommended Environment setup

— Dip-Switch

Table 6-1 QSPI mode setting

Switch Number	Switch Name	Pin1	Pin2	Pin3	Pin4
SW1	QSPI/Hyper Flash SW	ON	-	-	-
SW2	SOFTSW	OFF	OFF	OFF	OFF
SW6	MODESW	OFF	OFF	ON	ON
JP1	QSPI SEL	ON	ON	OFF	ON

6.2 Download QNX image to RAM

6.2.1 Scope

This section scope applies for downloading QNX image to RAM on board using u-boot TFTP command.

6.2.2 Requisite

Host PC installed a TFTP server (such as tftpd32 from http://tftpd32.jounin.net). Or Linux host configured as TFTP server.

Refer to section 3.3 “Build the BSP” to configure and build the BSP.

Copy “ifs-rcarm3-starterkit.bin” to TFTP server folder on Host PC.

Launch TFTP server on Windows Host PC, select path to “ifs-rcarm3-starterkit.bin” image file.

Or Start TFTP server on Linux Host, if TFTP server on Linux host is used.

Set dip-switch following the table 6-1.

6.2.3 Procedures for writing OS image to RAM using U-boot commands

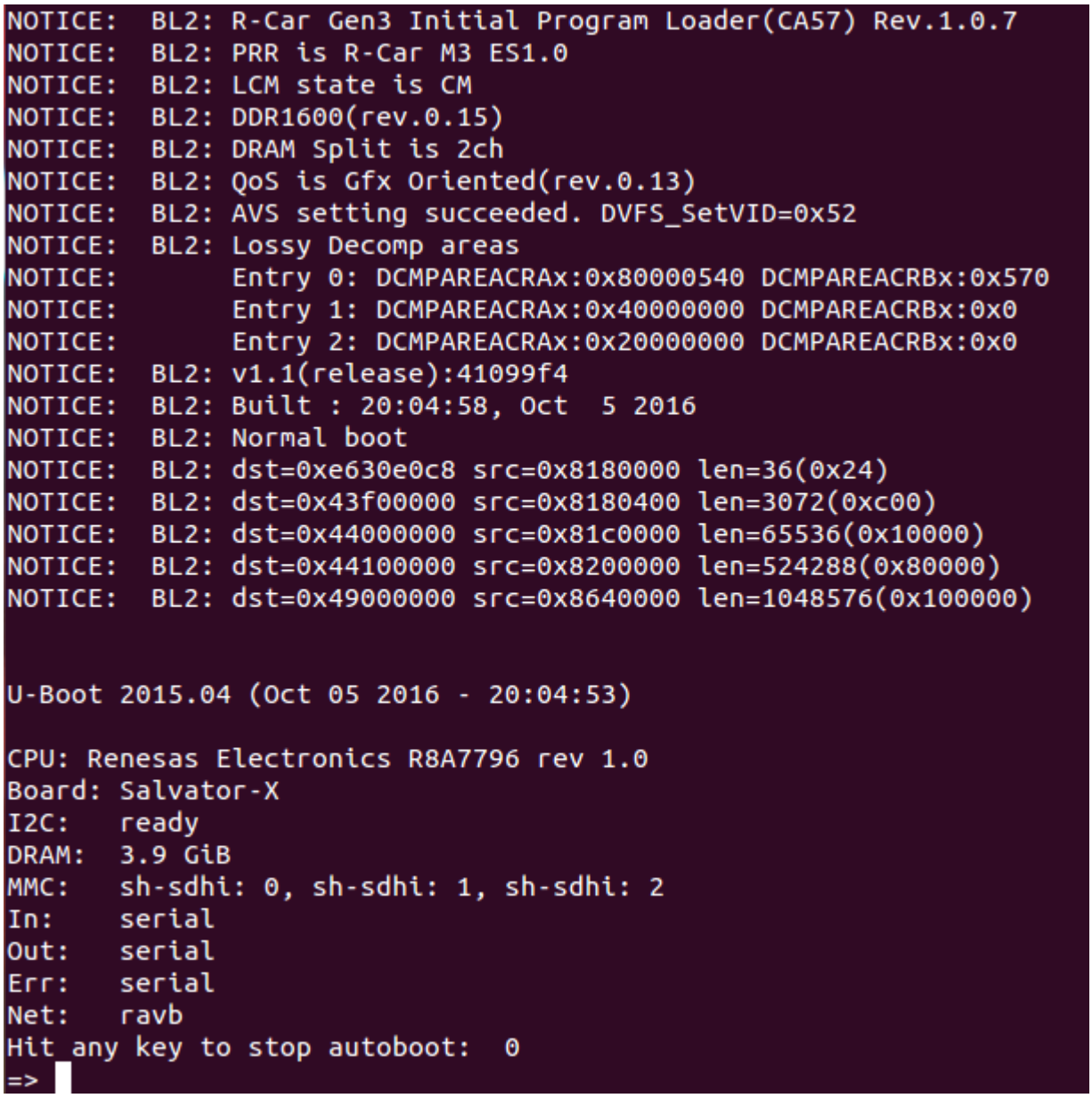

Figure 6-3 U-boot

Reset R-Car M3 StarterKit board.

U-boot will come up. Hit any key to abort autoboot.

Set environment variables:

- Set MAC address for R-Car M3 StarterKit board

```
=>set ethaddr 2e:09:0a:00:83:0c (depend on the sticker on Ethernet port CN7)
```

- Set the server IP (IP address of Windows PC run TFTP server) and client IP (the IP address of R-Car M3 StarterKit board):

```
=>setenv serverip <serverip>
=>setenv ipaddr <boardip>
```

- Set the image file name to be downloaded

```
=>setenv bootfile ifs-rcarm3-starterkit.bin
```

- Set the start address of RAM to load image

```
=>setenv loadaddr 0x40100000
```

- Download the QNX image to RAM

```
=>tftp
```

- Start QNX image

```
=>go 0x40100000
```

QNX 7.0 will start up on the board.

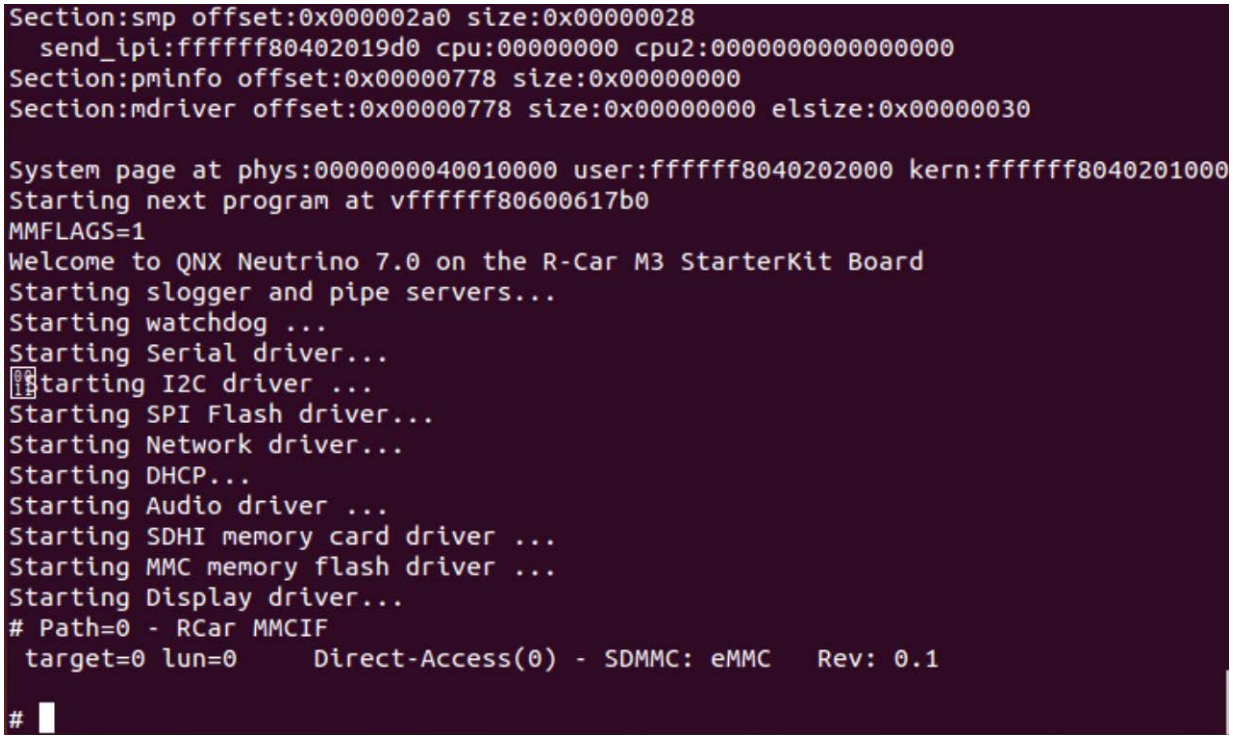

Figure 6-4 QNX 7.0 boot up log

7. Test drivers

7.1 Network

— List all running processes to check whether network is running or not:

```
# pidin a
    pid Arguments
     …
    12298 io-pkt-v6-hc -dravb mac=020060005200, -ptcpip pkt_typed_mem=below4G
    40971 dhclient -nw -lf /tmp/dhclient.leases ravb0
     …
#
```

— If network driver has not run yet, run it by command (using IPv6 networking manager)

```
# io-pkt-v6-hc -dravb mac=020060005200, -ptcpip pkt_typed_mem=below4G &
```

If dhcp client has not run yet, run it by

```
# dhclient -nw -lf /tmp/dhclient.leases ravb0
```

— View IP address configured by DHCP server

```
# ifconfig
   lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33136
            inet 127.0.0.1 netmask 0xff000000
            inet6 ::1 prefixlen 128
            inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
   ravb0: flags=8a43<UP,BROADCAST,RUNNING,ALLMULTI,SIMPLEX,MULTICAST> mtu 1500
            address: 2e:09:0a:00:83:0c
            media: Ethernet autoselect (1000baseT full-duplex,flowcontrol)
            status: active
            inet 10.20.14.68 netmask 0xffffff00 broadcast 10.20.14.255
            inet6 fe80::2c09:aff:fe00:830c%ravb0 prefixlen 64 scopeid 0x11
```

— If IP address is available, test your driver by send echo packets to your server machine.

```
# ping 10.20.14.10
PING 10.20.14.1 (10.20.14.1): 56 data bytes
64 bytes from 10.20.14.1: icmp_seq=0 ttl=255 time=6 ms
64 bytes from 10.20.14.1: icmp_seq=1 ttl=255 time=0 ms
64 bytes from 10.20.14.1: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 10.20.14.1: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 10.20.14.1: icmp_seq=4 ttl=255 time=0 ms
----10.20.14.1 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0/0/6 ms   variance = 3 ms^2
```

7.2 I2C

7.2.1 Start Driver

- Start driver if it has not run:

```
# i2c-rcar-A –p<port_address> -i<interrupt_number> --u<unit_number>
e.g., i2c-rcar-A -p0xE6518000 -i320 --u1
```

7.2.2 Test Driver

- Run command as below:

```
# i2c-control device=<device name> slave=<slave address> count=<number>
data=<data> subaddr=<sub-address> speed=<bus speed> mode=<write/read> <-v>
```

Option:

- ***Device name***: the path of i2c driver that you want to open. Ex: "/dev/i2c4".
- ***Slave address***: the address of device.
- ***Count***: number of bytes to be read.
- ***Data***: bytes to be written. In read mode, data byte is accepted 1 byte for sub-address.
- ***Mode***: write or read: mode for write or read data.
- ***Speed***: set bus speed (accept from 50000 to 400000).
- ***-v*** : show log and print read data.

Example: Use i2c-control to read from and write to the Versarclock5 5P49V5923A (slave address = 0x6A):

- Write data to the Versarclock5 with format data: data=data0:data1: … :dataN and subaddr=<sub-address>. Write memory address (sub-address = 0x10) and 4 bytes data to the Versarclock5:

```
# i2c-control device="/dev/i2c4" slave=0x6A data=02:03:0f:01 subaddr=0x10
speed=400000 mode=write -v
```

```
#
# i2c-control device="/dev/i2c4" slave=0x6a data=02:03:0f:01 subaddr=0x10 speed=400000 mode=write -v
device = /dev/i2c4, slave_add = 0x6a, speed = 400000, mode = write
Set bus speed to 400000Hz
Send 4 bytes of data
   Data sent: 10h 2h 3h fh 1h
# i2c-control device="/dev/i2c4" slave=0x6a count=4 subaddr=0x10 speed=400000 mode=read -v
device = /dev/i2c4, slave_add = 0x6a, speed = 400000, mode = read
Set bus speed to 400000Hz
   Data received: 2h 3h fh 1h
#
```

Figure 7-1 I2C test

7.3 Hyper Flash

7.3.1 Start Driver

- Start Hyper Flash driver if it was not started automatically when boot up QNX image.

```
# devf-rcar_qspi
```

7.3.2 Create Flash file system partition

- Use “flashctl” utility to create/erase/format SPI flash partition. Commands/options are as follows:

```
# flashctl –p <partition_path> -o <offset> -e –f -v
-e : erase partition
-f : format partition
-v : verbose
```

- **Requisite**

Use command “ls /dev/fs0*” to know how many partitions present. Then erase all of those partitions (except the primary partition (**fs0p0**)) by using below command:

```
# flashctl –p /dev/fs0p1 -ev
# flashctl -p /dev/fs0p2 -ev
```

After that, restart SPI Flash driver. We’ll have only one partition (fs0p0).

```
# slay devf-rcar_qspi
# devf-rcar_qspi &
```

- **Divide Partition**

We divide flash disk (fs0) to 2 partitions, the primary partition (dev/fs0p0 with size is 32MB) stores bootparam,SPI loader and u-boot/QNX image. The 2nd partition (dev/fs0p1 with size is 32MB) is for other purposes.

```
# flashctl -p /dev/fs0 -o 32M -ev
# flashctl -p /dev/fs0p0 -o 32M -fv
```

7.3.3 Restart driver

- Restart the driver we will have /dev/fs0p1 partition.

```
# slay devf-rcar_qspi
# devf-rcar_qspi
```

7.4 USB Driver

7.4.1 USB 2.0 Host

When testing USB 2.0 host on StarterKit board, it doesn’t need to set dip-switch setting. In the following procedure, dip-switch setting only applies for Salvator-X board.

— Change SW15 to 1-side to test USB 2.0 host on channel 0.

- **Test USB 2.0 host (EHCI)**

— If io-usb or devb-umass is running, kill it

```
# slay io-usb
# slay devb-umass
```

— Then start EHCI driver:

```
# io-usb -t memory=/memory/below4G -dehci ioport=0xEE080100,irq=0x8C,\
ioport=0xEE0A0100,irq=0x90,memory=/memory/below4G
```

— Start mass storage service:

```
# devb-umass blk noatime,commit=none,maxio=256,cache=10m cam pnp\
 mem name=/memory/below4G &
```

— Plug a USB stick in CN9/CN10A on Salvator-X board (or CN5 on StarterKit board). Mount and test read/write speed:

```
# mount –tdos /dev/hdXtY /usb
# cp –V /usb/filename /tmp        (Read data from USB)
# cp –V /tmp/filename /usb        (Write data to USB)
```

Note:

Replace hdXtY with the actual USB device node created in /dev/.

Slay io-usb function before using USB host with CN9 connector (Salvator-X board) if it is running.

- **Test USB 2.0 host (OHCI)**

— If io-usb or devb-umass is running, kill it

```
# slay io-usb
# slay devb-umass
```

— Start OHCI driver:

```
# io-usb -t memory=/memory/below4G -dohci ioport=0xEE080000,irq=0x8C,\
ioport=0xEE0A0000,irq=0x90,ioport=0xEE0C0000,irq=0x91,memory=/memory/below4G
```

— Start mass storage service:

```
# devb-umass blk noatime,commit=none,maxio=256,cache=10m cam pnp\
 mem name=/memory/below4G &
```

— Plug a USB stick in CN9/CN10A (or CN5 on StarterKit board). Mount and test read/write speed:

```
# mount –tdos /dev/hdXtY /usb
# cp –V /usb/filename /tmp        (Read data from USB)
# cp –V /tmp/filename /usb        (Write data to USB)
```

Note: Replace hdXtY with the actual USB device node created in /dev/.

7.4.2 USB 3.0 Host (Salvator-X Only)

- **Test USB 3.0 host (XHCI)**

— If io-usb or devb-umass is running, kill it

```
# slay io-usb
# slay devb-umass
```

— Start XHCI driver:

```
# io-usb –dxhci ioport=0xEE000000,irq=0x86,memory=/memory/below4G
```

— Start mass storage service:

```
# devb-umass blk noatime,commit=none,maxio=256,cache=10m cam pnp mem
name=/memory/below4G &
```

— Plug a USB 3.0 stick in CN11. Verify USB port speed by running the command below and checking "Upstream Port Speed" property is "Super Speed":

```
# usb -vvv
```

```
# usb -vvvvvv
USB 0 (XHCI) v1.10, v1.01 DDK, v1.01 HCD
    Control, Interrupt, Bulk(SG), Isoch(Stream), High speed

Device Address              : 1
Upstream Host Controller    : 0
Upstream Device Address     : 0
Upstream Port               : 1
Upstream Port Speed         : Super
Vendor                      : 0x0781 (SanDisk)
Product                     : 0x5581 (Ultra)
Device Release              : r1.00
USB Spec Release            : v3.00
Serial Number               : 4C530001040728115010
Class                       : 0x00 (Independent per interface)
Max PacketSize0             : 512
Languages                   : 0x0409 (English)
Current Frame               : 12060 (1024 bytes)
```

Figure 7-2 USB device details

— Mount and test read/write speed:

```
# mount –tdos /dev/hdXtY /usb
# cp –V /usb/filename /tmp        (Read data from USB)
# cp –V /tmp/filename /usb        (Write data to USB)
```

Note: Replace hdXtY with the actual USB device node created in /dev/.

7.4.3 USB 2.0 Function (Salvator-X Only)

— Please make sure SW15 is on middle side (2-side)

— Connect USB port CN9 on the board with a Ubuntu PC

— If io-usb-dcd is running on the board, kill it

```
# slay io-usb-dcd
```

— Then run commands as follows.

```
# io-usb-dcd -n /dev/otg/io-usb-dcd -d usbncm-hsusb-rcar ioport=0xe6590000,\
irq=139,verbose=10
# mount -Tio-pkt -o  verbose=0,path=/dev/otg/io-usb-
dcd,protocol=ncm,usbdnet_mac=000022446688,mac=020022446688 devnp-usbdnet.so
# ifconfig ncm0 192.168.0.1
# ulink_ctrl -l 1 -s /dev/otg/io-usb-dcd
```

— On Ubuntu PC, open Terminal. From Terminal, type “ifconfig -a” to see all network interfaces. Interface “usb0” which represents for USB NCM interface with the board, will appear.

```
usb0      Link encap:Ethernet  HWaddr 00:00:22:44:66:88
          inet6 addr: fe80::200:22ff:fe44:6688/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX
```

— Set IP address for “usb0” interface, then try “ping” command to verify data transfer between board and PC.

```
# ifconfig usb0 192.168.0.2
# ping 192.168.0.1
PING 10.20.14.12 (10.20.14.12) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=1.70 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=1.75 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=1.73 ms
```

7.5 Audio Driver

Audio driver implements all SSI channels. With board, SSI0 and SSI1 are connecting to AK4613VQ CODEC, SSI0 for playback and SSI1 for recording. We can specify HDMI channel used to input audio data.

7.5.1 Test Specifications

7.5.1.1 Test overview

— To make sure that playback and record processes of audio driver meet its specifications, we need the following applications/utilities:

- A PCM player can play wave & mp3 files
- A PCM recorder can record sound/our voice.

— Launch audio driver

```
# io-audio -d rcar-ssi [driver options]
```

Table 7-1 driver options

No	Option	Description
1	scu=<value>	Specify SCU module used or not, SCU is used by default. - value=1: SCU is used. - value=0: SCU is not used.
2	tx_ssi_num=<value>	Specify SSI channel number used for playback(Single or multi-channel SSI will be specified): - value=1 or 3 or 4 (by default tx_ssi_num=1) 3 or 4 SSI channels for transmission and reception cannot coexists. But only be able to hear all channels of multi-channel SSI via HDMI in multi-channel SSI transmission.
3	rx_ssi_num=<value>	Specify SSI channel number used for record (Single or multi-channel SSI will be specified): - value=1 or 3 or 4 (by default rx_ssi_num=1) 3 or 4 SSI channels for transmission and reception cannot coexists. But Board doesn't support multi-channel SSI reception.
4	tx_ssi_idx=<value>	Specify SSI channel index (from 0 to 9) used in playback if Single SSI (tx_ssi_num=1) is specified in #2. Only SSI0 is routed to input of AK4613, but SSI0 to SSI9 can be specified to route to HDMI controller. SSI0 is used by default.
5	rx_ssi_idx=<value>	Specify SSI channel index (from 0 to 9) in record if Single SSI (rx_ssi_num=1) is specified in #3. Only SSI1 is routed to output of AK4613. SSI1 is used by default.
6	tx_ssi_mode=<value>	Specify transmission SSI mode: - value=master : Transmission SSI will output clock signals (by default). - value=slave : Transmission SSI will input clock signals.
7	rx_ssi_mode=<value>	Specify reception SSI mode: - value=master : Reception SSI will output clock signals (by default). - value=slave : Reception SSI will input clock signals.
8	tx_voices=<value>	Specify maximum number of playback voices. 2 or 6 or 8 voices will be specified. 2 (stereo) is specified by default.
9	rx_voices=<value>	Specify maximum number of record voices. 2 or 6 or 8 voices will be specified. 2 (stereo) is specified by default.
10	tx_bus_format=<value>	Specify bus format for playback: - value=i2s : Stereo/ multi-channel format (by default) - value=tdm : TDM format - value=mono : monaural format
11	rx_bus_format=<value>	Specify bus format for record: - value=i2s : Stereo/ multi-channel format (by default)

		- value=tdm : TDM format - value=mono : monaural format
12	tx_pin_mode=<value>	Specify clock and WS pin mode for playback: - value=0 : Use pins independently (by default) - value=1 : Use common pins with another SSI - value=2 : Use common pins with another SSI and another SSI in master
13	rx_pin_mode=<value>	Specify clock and WS pins mode for record: - value=0 : Use pins independently - value=1 : Use common pins with another SSI - value=2 : Use common pins with another SSI and another SSI in master (by default)
14	tx_sample_size=<value>	Specify sample size of bytes for playback. 2 is specified by default.
15	rx_sample_size=<value>	Specify sample size of bytes for record. 2 is specified by default.
16	tx_data_align=<value>	Specify data align for playback: - value=left_align : Left align - value=right_align : Right align (by default)
17	rx_data_align=<value>	Specify data align for playback: - value=left_align : Left align (by default) - value=right_align : Right align
18	tx_tdm_mode=<value>	Specify TDM mode for playback if TDM format (tx_bus_format=tdm) is specified in #8: - value=0 : TDM Extend Mode (by default) - value=1 : TDM 16ch Mode - value=2 : TDM Split Mode - value=3 : TDM Ex-Split Mode
19	rx_tdm_mode=<value>	Specify TDM mode for record if TDM format (rx_bus_format=tdm) is specified in #9: - value=0 : TDM Extend Mode (by default) - value=1 : TDM 16ch Mode - value=2 : TDM Split Mode - value=3 : TDM Ex-Split Mode
20	tx_src_idx=<value>	Specify SCU-SRC channel index (from 0 to 9) used in playback if scu=1 is specified in #1. SRC0 is used by default.
21	rx_src_idx=<value>	Specify SCU-SRC channel index (from 0 to 9) used in capture if scu=1 is specified in #1. SRC1 is used by default.
22	tx_cmd_idx=<value>	Specify SCU-CMD channel index (from 0 to 1) used in playback if scu=1 is specified in #1. CMD0 is used by default.

23	hdmi_idx=<value>	Specify HDMI channel index (from 0 to 1) used. HDMI0 is used by default.
24	clock_scr=<value>	Specify clock source is used to create clock rates: - value=internal : Use internal clock S0D4 200MHz - value=external : Use external clock 22.5792MHz input to AUDIO_CLK_A pin (by default)

— Launch Example:

SSI0 for playback and SSI1 for record via AK4613 CODE:

 # io-audio -d rcar-ak4613

7.5.1.2 Equipment and Software use

— Playback audio file

Table 7-2 Audio file format

Type	Format	File Name
AUDIO	WAVE, 8 kHz	
	WAVE, 11.025 kHz	
	WAVE, 16 kHz	
	WAVE, 22.050 kHz	
	WAVE, 32kHz	
	WAVE, 44.1 kHz	
	WAVE, 48 kHz	
	mp3	

— Test programs (wave, waverec, mmrplay)

We can copy the audio files to a location under /tmp of the target.

7.5.1.3 Test Items

Table 7-3 Audio test items

No.	Test Item	Test Outline
1	Playback test	Specify an audio file and make sure that the specified file is replayed.
2	Record test	Perform recording and make sure that input audio is recorded.

7.5.2 Test Procedure

7.5.2.1 Playback

— Start audio driver to play/record as described in 6.3.1 (you need to kill io-audio process if it has been started in a different mode already)

— Connect a speaker/headphone to port CN24-Lower on Salvator-X board (or CN8 on StarterKit board).

— Start wave from the command line on the console. Specify the full path for the playback file as a wave argument.

— Check the sound from the speaker by hearing it.

Example: *# wave /tmp/44k.wav*

— To change volume during playing, focus the terminal window screen on the PC and then:

- Press key ‘q’ to increase left volume
- Press key ‘a’ to descrease left volume
- Press key ‘e’ to increase right volume
- Press key ‘d’ to descrease right volume
- Press key ‘w’to increase left and right volumes concurrently
- Press key ‘s’to descrease left and right volumes concurrently

7.5.2.2 Recording

— Start audio driver to playback/record as described in 6.4.1 (you need to kill io-audio process if it has been started in a different mode already).

— Connect a speaker/headphone to port CN24-Upper on Salvator-X board (or CN9 on StarterKit board).

— Start waverec from the command line on the console.

Example: # waverec /tmp/1

— To change volume during recording, focus the terminal window screen on the PC and then:

- Press key ‘q’ to increase left volume
- Press key ‘a’ to descrease left volume
- Press key ‘e’ to increase right volume
- Press key ‘d’ to descrease right volume
- Press key ‘w’to increase left and right volumes concurrently
- Press key ‘s’to descrease left and right volumes concurrently

Table 7-4 Audio record command's options

Argument	Description
-r <sampling rate>	Specifies the sampling rate (48000/44100/22050/11025). By default 44100 is used .
-8	Specifies the 8-bit mode. By default, the 16-bit mode is used.
-a [card#:]<dev#>	Specifies the card number and device number for recording. By default, card 0 and device 0 is specified.
-m	Specifies the monaural recording. By default, the stereo recording is performed.
-t <time>	Specifies the recording time. By default, it is 5 seconds.

— Make sure that record data is created. Then, replay to check the recorded audio.

7.6 SD Card Driver

7.6.1 Start/stop the driver

1. Start the driver

   Start SD card driver if it is not started automatically when boot up QNX. From the console, run the driver as

follows:

— For Salvator-X board:

```
# devb-sdmmc-rcar_sdhi-salvatorx blk cache=2m cam pnp,bounce=128k mem
name=below4G sdio idx=0
And
# devb-sdmmc-rcar_sdhi-salvatorx blk cache=2m cam pnp,bounce=128k mem
name=below4G sdio idx=3
```

— For StarterKit board:

```
# devb-sdmmc-rcar_sdhi-starterkit blk cache=2m cam pnp,bounce=128k mem
name=below4G sdio idx=0
And
# devb-sdmmc-rcar_sdhi-starterkit blk cache=2m cam pnp,bounce=128k mem
name=below4G sdio idx=3
```

After the driver started successfully, device nodes hdX and hdXtY - with X, Y varying depending on the SD card partition type and whether there is any other block devices are created in /dev/ or not. To check this “run ls /dev” as follows.

```
# ls dev
bpf             mem             ptyp7           stdout          ttyp4
bpf0            null            ramdisk0        tap             ttyp5
console         otg             ramdisk0t77     tap0            ttyp6
crypto          pipe            screen          tap1            ttyp7
dbgmem          profiler        sem             tap2            tun0
hd0             ptyp0           ser1            tap3            tun1
hd0t12          ptyp1           shmem           text            tun2
i2c2            ptyp2           slog            tty             tun3
i2c4            ptyp3           snd             ttyp0           tymem
io-usb          ptyp4           socket          ttyp1           zero
i2c4            ptyp5           stderr          ttyp2
log             ptyp6           stdin           ttyp3
```

In above case, device node *hd0* and *hd0t12* is created in /dev/

Meaning of X and Y in device node hdXtY is as follows:

— X is device node index.

— Y is partition specifier on SD card.

The following is some partition specifier which QNX supports:

- DOS partition: 6, 11, 12
- QNX4 partition: 77, 78, 79
- QNX6 partition: 177, 178, 179

2. Mount the SD card

From the console, “mount” a filesystem on the SD card as follows:

— DOS partition:

```
# mount –w –tdos /dev/hdXtY /sd
```

— QNX4 partition:

```
# mount –w –tqnx4 /dev/hdXtY /sd
```

— QNX6 partition:

```
# mount –w –tqnx6 /dev/hdXtY /sd
```

Replace hdXtY with the actual SD device node created in /dev/

Make sure that a folder with the mount name /sd is created using “ls”.

```
# ls
bin     etc     proc    sd      usr
dev     lib     sbin    tmp     var
```

3. Un-mount the SD card

From the console, un-mount the SD card as follows:

```
# umount /sd
```

Make sure that the folder /sd disappeared.

4. Stop the driver

From the console run command below to stop driver

```
# slay devb-sdmmc-rcar_sdhi
```

Make sure that the /dev/hdX and /dev/hdXtY disappeared using “ls /dev” and the devb-sdmmc-rcar_sdhi process is not listed using “pidin”.

7.6.2 Write data

1. Start SD card driver
2. Mount the SD card to a folder, for example /sd
3. Copy a file to the /sd folder

```
# cp testfile /sd
```

Make sure that the file exists in SD card and the content of file is correct.

7.6.3 Read data

1. Start SD card driver
2. Mount the SD card to a folder, for example /sd
3. Copy a file from the sd folder

```
# cp /sd/testfile /tmp
```

Make sure that the file exists in /tmp folder and file content is the same with original file from SD card.

7.7 MMC Driver

7.7.1 Start/stop the driver

1. Start the driver

   From the console, start the driver as follows if it is not started automatically when boot up QNX:

```
# devb-sdmmc-rcar_mmcif blk cache=2m cam bounce=128k mem name=below4G sdio
   idx=0
```

   After the driver is started successfully, a device node hdX is created in /dev/ (X=0,1,2,…). To check this, from the console run “ls /dev” as follows:

```
# ls dev
bpf             null            ramdisk0        tap             ttyp5
bpf0            otg             ramdisk0t77     tap0            ttyp6
console         pipe            screen          tap1            ttyp7
crypto          profiler        sem             tap2            tun0
dbgmem          ptyp0           ser1            tap3            tun1
hd0             ptyp1           shmem           text            tun2
i2c2            ptyp2           slog            tty             tun3
i2c4            ptyp3           snd             ttyp0           tymem
io-usb          ptyp4           socket          ttyp1           zero
i2c4            ptyp5           stderr          ttyp2
log             ptyp6           stdin           ttyp3
mem             ptyp7           stdout
```

   In above case, device node *hd0* is created in /dev/

   Note that: if partition(s) is (are) created partitions already, beside hdX, device node hdXtY is(are) created in /dev/

2. Create partitions on eMMC flash

   If partition(s) is (are) created already, you can skip this step, but if you want re-create partitions you should use “fdisk” to delete the existing partitions as below:

```
#fdisk /dev/hdX delete –a
```

   Please note that, after using “fdisk” to delete the existing partitions, you need to stop and then start again the driver as told in step 1.

   To create partitions on eMMC flash, using “fdisk” utility. The following is an example of creating a QNX4, QNX6 or DOS partition on eMMC flash with assuming device node *hd0* is created in /dev/ after started the driver.

   - QNX4 partition

```
#fdisk /dev/hd0 add -t 77
```

- QNX6 partition

```
#fdisk /dev/hd0 add -t 178
```

- DOS partition

```
#fdisk /dev/hd0 add -t 12
```

If you want to create a partition with size less than size of the flash, you use "-p" option. The following is example of creating a DOS partition and a QNX6 partition using "-p" option, each partition is 50% of the flash:

```
#fdisk /dev/hd0 add -t 11 –p 50
#fdisk /dev/hd0 add –t 178 –p 50
```

After run "fdisk", you will see device node *hd0t77* or *hd0t178* or *hd0t12* beside device node *hd0* created in /dev/ and to check this you need to stop then start again the driver as told in step 1 and then "run ls /dev" as follows:

```
# ls dev
bpf             mem             ptyp7           stdout          ttyp4
bpf0            null            ramdisk0        tap             ttyp5
console         otg             ramdisk0t77     tap0            ttyp6
crypto          pipe            screen          tap1            ttyp7
dbgmem          profiler        sem             tap2            tun0
hd0             ptyp0           ser1            tap3            tun1
hd0t178         ptyp1           shmem           text            tun2
i2c2            ptyp2           slog            tty             tun3
i2c4            ptyp3           snd             ttyp0           tymem
io-usb          ptyp4           socket          ttyp1           zero
i2c4            ptyp5           stderr          ttyp2
log             ptyp6           stdin           ttyp3
```

In above case, device node *hd0t178* created in /dev/ after using "fdisk" to create a QNX6 partition.

3. Format filesystem on the eMMC flash

After created partition, you need to format a filesystem on it.

- QNX4 partition: Use "dinit" to format a QNX4 filesyetem on QNX4 partitions. If "dinit" utility is not available, you need to copy it to /tmp folder. The following is example of formatting QNX4 partition which is created in step 2.

```
# dinit -h /dev/hd0t77
```

- QNX6 partition: Use "mkqnx6fs" to format a QNX6 filesystem on QNX6 partitions. If "mkqnx6fs" utility is not available, you need to copy it to /tmp folder. The following is example of formatting QNX6 partition which is created in step 2.

```
# mkqnx6fs /dev/hd0t178
```

- DOS partition: Use “mkdosfs” to format a DOS filesystem on DOS partitions. If “mkdosfs” utility is not available, you need to copy it to /tmp folder. The following is example of formatting DOS partition which is created in step 2.

```
# mkdosfs /dev/hd0t12
```

4. Mount filessystem on the eMMC flash

   To access a filessystem on the eMMC flash, you need to “mount” it. The following is example of mounting a filesystem on partition which created and formatted in step 2 and 3.

   - QNX4 partition

```
# mount -w –tqnx4 /dev/hd0t77 /mmc
```

   - QNX6 partition

```
# mount -w –tqnx6 /dev/hd0t178 /mmc
```

   - DOS partition

```
# mount -w –tdos /dev/hd0t12 /mmc
```

   Make sure that a folder with the mount name /mmc is created using “ls” command.

```
# ls
bin     etc     mmc     sbin    usr
dev     lib     proc    tmp     var
```

5. Un-mount the filesystem on the eMMC flash

   From the console, un-mount the filesystem on the eMMC flash as follows:

```
# umount /mmc
```

   Make sure that the folder /mmc disappears.

6. Stop the driver

   From the console run command below to stop the driver

```
# slay devb-sdmmc-rcar_mmcif
```

   Make sure that the /dev/hdX and /dev/hdXtY disappeared using “ls /dev” and the devb-sdmmc-rcar_mmcif process is not listed using “pidin”.

7.7.2 Write data

1. Start MMC driver
2. Create partition and format filesystem on eMMC flash
3. Mount eMMC flash to /mmc folder
4. Copy a file to the /mmc folder

```
# cp –V testfile /mmc
```

   Use option –V to see speed of transfer to eMMC flash. Make sure that the file exists in eMMC flash and the

content of file is correct.

7.7.3 Read data

1. Start MMC driver
2. Create partition and format filesystem on eMMC flash
3. Mount eMMC flash to a folder, for example /mmc
4. Copy a file from /mmc folder to /tmp folder

```
# cp –V /mmc/testfile /tmp
```

Use option –V to see speed of transfer from eMMC flash. Make sure that the file exists in /tmp folder and file content is the same with original file from the eMMC flash.

7.8 Watchdog Timer

— If Watchdog Counter becomes overflow then an internal reset will generated after 1 minute from start.

— If the Watchdog Counter is reset periodically, Watchdog Counter will not be overflow and a reset will not be generated.

— The Watchdog Timer is started at the time of starting QNX image. In order to test reset generated by the Watchdog Counter overflow, from console, run below command:

```
# kill <wdkick_process_id>
```

Waiting for about 1 minute a reset will occur and u-boot program or QNX OS is restarted.

7.9 VIN capture library (Salvator-X only)

— VIN1 support:
  - o Connector: RCA
  - o Video Interface: ITU-R BT.656
  - o Video Standard: PAL/NTSC/SECAM

— VIN0 support:
  - o Connector: HDMI
  - o Video Interface: ITU-R BT.601/BT.709
  - o Video Standard: 720x480p60, 1280x720p60, 1920x1080i60, 720(1440)x480i60, 720x576p50, 1280x720p50, 1920x1080i50, 720(1440)x576i50, 800x600(SVGA)@60, 640x480(VGA)@60, 800x480(WVGA)@60, 1024x768(XGA)@60

— When capture HDMI video from consumer electronic devices which require EDID (Extend Display Identification Data) such as laptop, preferred resolution is set to 1920x1080i60 and camera-ctrl application must be started before connecting HDMI cable. In case of using a HDMI Signal Generator, it is not necessary.

— To test the driver, please run command as below:

- For testing VIN0 on HDMI0 monitor

```
#camera_ctrl -display=1 -pipeline=5 -device=0 -bsize=[width]x[height] -
size=[width]x[height]
```

- For testing VIN1 on HDMI0 monitor

```
#camera_ctrl -display=1 -pipeline=5 -device=1 -bsize=[width]x[height] -
size=[width]x[height]
```

- For testing VIN0 on LVDS monitor

```
#camera_ctrl -display=2 -pipeline=1 -device=0 -bsize=[width]x[height] -
size=[width]x[height]
```

- For testing VIN1 on LVDS monitor

```
#camera_ctrl -display=2 -pipeline=1 -device=1 -bsize=[width]x[height] -
size=[width]x[height]
```

- For testing VIN0 on VGA monitor

```
#camera_ctrl -display=3 -pipeline=9 -device=0 -bsize=[width]x[height] -
size=[width]x[height]
```

- For testing VIN1 on VGA monitor

```
#camera_ctrl -display=3 -pipeline=9 -device=1 -bsize=[width]x[height] -
size=[width]x[height]
```

Note: Please press "Ctrl + C" key to exit "camera_ctrl" application test.

7.10 Display driver

Table 7-5 Describes all display which Display driver supports

Number	Type	DU	Pipeline	BSP configuration	Default resolution	Connector	
						Salvator-X	StaterKit
1	HDMI0	DU1	5*1, 6, 7, 8	Main display	1920x1080	CN16	CN4
2	LVDS	DU0	1*1, 2, 3, 4	Secondary display	1280x800	CN18	X
3	VGA	DU2	9*1, 10, 11, 12	Secondary display	1280x720	CN15	X

Note:

*1: scaling pipeline

7.10.1 Start/stop the driver

If display driver has not run yet, run it by

```
#GRAPHICS_ROOT=/usr/lib/graphics/R_CarM3
#LD_LIBRARY_PATH=/usr/lib:/lib:/lib/dll:$LD_LIBRARY_PATH
#screen &
```

To stop the driver, use "kill <screen process ID>" or "slay screen" then make sure screen process has been terminated.

7.10.2 Test driver

- Connect LVDS monitor to port CN18.
- Connect VGA monitor to port CN15.
- Connect one HDMI monitor to CN16 (HDMI0)
- Make sure that screen has been started.

To each display, run sw-vsync with display=1 or 2 or 3

```
# sw-vsync –display=[1,2,3] –size=[width]x[height] &
```

To test scaling on LVDS monitor, run following command:

```
# yuv-test -format=nv12 -pipeline=1 -display=3 –size=[width]x[height] &
```

To test scaling on HDMI0 monitor, run following command:

```
# yuv-test -format=nv12 -pipeline=5 -display=1 –size=[width]x[height] &
```

REVISION HISTORY	QNX 7.0 BSP for R-Car M3 Platforms

Rev.	Date	Description	
		Page/Section	Summary
1.00	Mar 3, 2017	—	First Edition issued.

QNX 7.0 BSP for R-Car M3 Platforms
User's Manual: Software

Publication Date: Rev.1.00 Mar 3, 2017

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com
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 95050-2554, 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

QNX 7.0 BSP for R-Car M3 Platforms

Renesas Electronics Corporation