User's Manual

CONFIDENTIA

R-Car Gen4 Flash writer sample software

User's Manual: Software

SoC for Car Information Terminal Applications
R-Car S4/V4H/V4M Series

Renesas Electronics
www.renesas.com

R11UZ0239EJ0130
Rev. 1.30.0 Sep. 2023

CONFIDENTIAL

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Trademark

How to Use This Manual

- **[Readers]**
  This manual is intended for engineers who develop products which use the R-Car S4 / V4H / V4M processor.

- **[Purpose]**
  This manual is intended to give users an understanding of the functions of the R-Car S4 / V4H / V4M 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.
  → Read this manual in the order of the CONTENTS.
  — To understand the functions of a multimedia processor for R-Car S4 / V4H / V4M
  → See the R-Car S4 / V4H User's Manual.
  — To know the electrical specifications of the multimedia processor for R-Car S4 / V4H / V4M
  → See the R-Car S4 / V4H 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

1. Overview

1.1 Overview

This document explains about R-Car Gen4 Flash writer sample software (hereafter referred to as “Flash writer”).
The Flash writer is downloaded from the Host PC via SCIF by Boot ROM.
The Flash writer transfers some of binary images from Host PC via SCIF and writes the binary images to QSPI Flash, HyperFlash and eMMC (hereafter referred to as “external flash memory”).

SoC-dependent parts are distinguished using the following labels.

R-Car S4 R-Car V4H R-Car V4M

1.2 Supported device

R-Car S4 R-Car V4H R-Car V4M

Supported devices are summarized in Table 1-1.

Table 1-1 Supported device

	R-Car S4 **Flash writer**		R-Car V4H R-Car V4M **Flash writer**	
Device	**Read**	**Write**	**Read**	**Write**
QSPI Flash	Support	Support (40MHz, 1bit)	Support	Support (40MHz, 1bit)
Octal SPI Flash	Not support	Not support	Not support	Not support
HyperFlash	Support	Support (40MHz, 8bit)	Support	Support (40MHz, 8bit)
eMMC	Support	Support (50MHz, 8bit)	Support	Support (50MHz, 8bit)

1.3 References

R-Car S4 R-Car V4H R-Car V4M

The following table shows the document related to this function.

Table 1-2 Related document

Number	Issue	Title	Edition
1	infineon (Cypress)	512 Mbit, 1.8 V Serial Peripheral Interface with Multi-I/O Flash	Rev. *M
2	JEDEC	Embedded Multi-Media Card (eMMC) Electrical Standard (5.01)	JESD84-B50.1
3	infineon (Cypress)	512 MBIT, 256 MBIT, 128 MBIT HYPERFLASH FAMILY Datasheet	Rev. *M
4	Renesas	R-Car S4 R-Car S4 System Evaluation Board Spider Hardware Manual	Refer to latest version
5	Renesas	R-Car S4 R-Car S4 System Evaluation Board Spider Setup Manual	Refer to latest version
6	Renesas	R-Car V4H R-Car V4H System Evaluation Board White Hawk Hardware Manual	Refer to latest version
7	Renesas	R-Car V4H R-Car V4H System Evaluation Board White Hawk Setup Manual	Refer to latest version
8	Renesas	R-Car V4M R-Car V4M System Evaluation Board Gray Hawk Hardware Manual	Refer to latest version
9	Renesas	R-Car V4M R-Car V4M System Evaluation Board Gray Hawk Setup Manual	Refer to latest version

1.4 Restrictions

R-Car S4 R-Car V4H R-Car V4M

There is no restriction in this revision.

2. Operating Environment

2.1 Hardware Environment

R-Car S4 R-Car V4H R-Car V4M

The following table lists the hardware needed to use this function.

Table 2-1 Hardware Environment

Name	Note
R-Car S4 R-Car S4 System Evaluation Board Spider	RTP8A779F0ASKB0SP0SA280
R-Car V4H R-Car V4H System Evaluation Board White Hawk	RTP8A779G0ASKB0FC0SA000
Host PC	Windows 10 is recommended for Windows host PC
USB cable (type A to micro B)	R-Car S4 Connect to CN21 when using UART connection. (HSCIF0) R-Car V4H R-Car V4M Connect to CN10 when using UART connection. (HSCIF0)

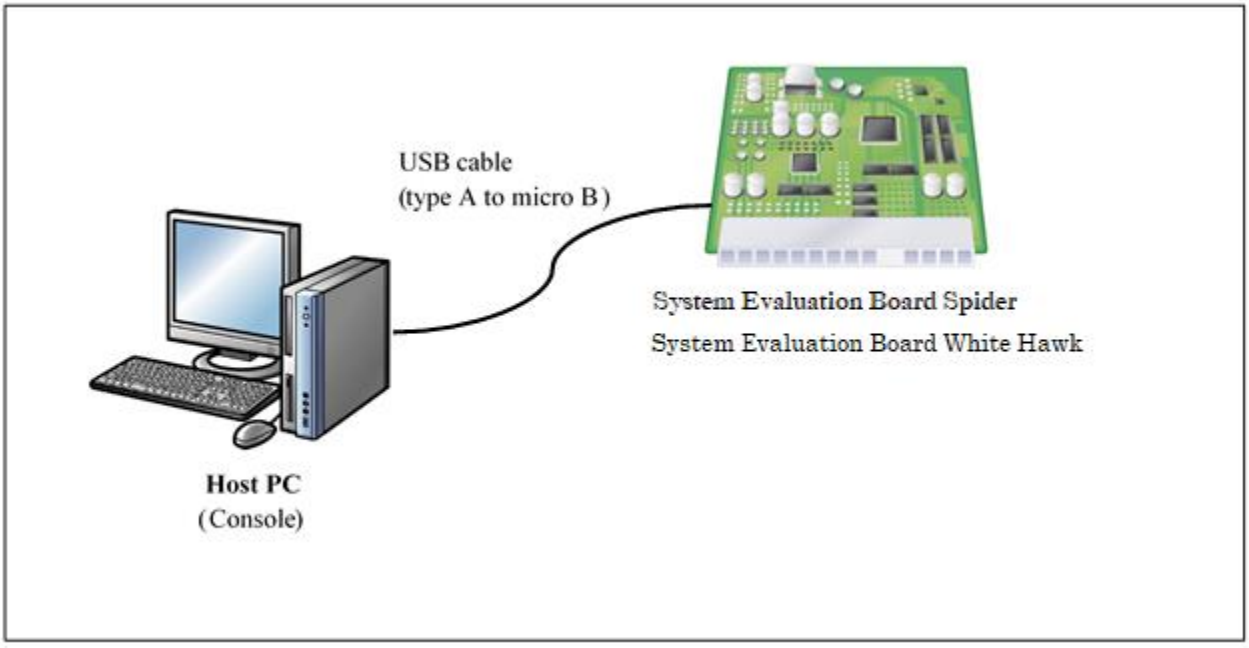

Figure 2-1 Recommended Environment

The following table shows external flash memory support for R-Car S4 and R-Car V4H and R-Car V4M.

Table 2-2 External flash memory support status

SoC	QSPI Flash	HyperFlash	eMMC
R-Car S4 R-Car S4 Ver.1.0 R-Car S4 Ver.1.1 R-Car S4 Ver.1.2	Support	Support	Support
R-Car V4H R-Car V4H Ver.1.0 R-Car V4H Ver.2.0 R-Car V4H Ver.2.1 R-Car V4H Ver.2.2	Support	Support	Support
R-Car V4M R-Car V4M Ver.1.0	Support	Support	Support

The following table shows which model number of external flash memory is supported.

Table 2-3 Model number of supported external flash memory

Board	QSPI Flash	HyperFlash	eMMC
Spider	S25FS512SAGMFV01	S26KS512SDPBHV02	MTFC32GAZAQHD-AAT
White Hawk	S25FS512SAGMFV01	S26KS512SDPBHV02	MTFC32GAZAQHD-AAT
Gray Hawk	S25FS512SAGMFV01	S26KS512SDPBHV02	MTFC32GAZAQHD-AAT

The following table shows hardware resource used by Flash writer.

Table 2-4 Hardware resource

Component	Channel	Purpose
RPC	-	Reading / Writing the QSPI Flash and HyperFlash.
PFC/GPIO	-	Setting the External pin.
CPG	-	Starts / Stops supplying the clock signal to modules.
DBSC4/PHY	-	Initialization for accessing to the SDRAM.
QoS	-	Setting the arbitration.
RT-DMAC	ch0	Accelerating software loading with the DMA.
HSCIF	ch0	Displaying starting message and log message.
QSPI Flash	-	Reading / Writing data from / to this device (External device).
HyperFlash	-	Reading / Writing data from / to this device (External device).
eMMC	-	Writing data to this device (External device).
SDHI	-	Reading / Writing the eMMC.
RST	-	Transferring of reset signal for each IP.
R-Car S4 LPDDR4X-SDRAM	-	Using for work buffer (External device).
R-Car S4 RT-SRAM	-	Executing the Flash Writer on RT-SRAM.
R-Car V4H R-Car V4M LPDDR5-SDRAM	-	Using for work buffer (External device).
R-Car V4H R-Car V4M RT-VRAM0	-	Executing the Flash Writer on RT-VRAM0.

2.2 Software Environment

R-Car S4 R-Car V4H R-Car V4M

The following table lists the software needed to use this function.

Table 2-5 Software environment list

Name	Note
GHS compiler	Comp Version V2020.1.5
Cygwin	Cygwin Version 3.1.5-1 make Version 4.3
Tera Term	Version 4.97 (SVN# 6995)

3. Software

3.1 Function

R-Car S4 R-Car V4H R-Car V4M

This package has the following functions.

- Read the QSPI Flash and HyperFlash memory data.
- Write images to the QSPI Flash and HyperFlash memory.
- Erase the QSPI Flash and HyperFlash memory.
- Display the CID/CSD/EXT_CSD registers of eMMC.
- Modify the EXT_CSD registers of eMMC.
- Read the eMMC memory data.
- Write images to the user data area of eMMC.
- Display the command help.

3.2 Module structure

R-Car S4 R-Car V4H R-Car V4M

This module structure is shown below.

```
flash_writer                                  : root directory of Flash writer
|-- ddr                                       : DRAM initialize code directory
|-- ICUMX_boot                                : boot code
|-- ICUMX_obj                                 : object file output directory
|-- ICUMX_output                              : output directory
|-- include                                   : header files directory
|-- qos_s4.c                                  : qos initialize code  S4
|-- qos_v4h_v4m.c                             : qos initialize code V4H / V4M
|-- cert_param.c                              : image header for SCIF download image
|-- common.c                                  : miscellaneous code
|-- devdrv.c                                  : log output code
|-- dg_emmc_access.c                          : eMMC writer code
|-- dg_emmc_config.c                          : eMMC device configuration code
|-- dg_flash_access.c                         : Flash writer code
|-- dgmodul.c                                 : Each device common writer code
|-- dgtable.c                                 : command table
|-- dma.c                                     : DMA driver code
|-- emmc_boot.c                               : eMMC driver code
|-- emmc_cmd.c                                : eMMC driver code
|-- emmc_erase.c                              : eMMC driver code
|-- emmc_init.c                               : eMMC driver code
|-- emmc_interrupt.c                          : eMMC driver code
|-- emmc_mount.c                              : eMMC driver code
|-- emmc_read.c                               : eMMC driver code
|-- emmc_utility.c                            : eMMC driver code
|-- emmc_write.c                              : eMMC driver code
|-- hscifdrv0.c                               : HSCIF driver code
|-- hyperflashdrv.c                           : HyperFlash driver code
|-- lsi_id.c                                  : identify the LSI type
|-- main.c                                    : main program
|-- message.c                                 : Help message
|-- micro_wait.c                              : miscellaneous code
|-- qspi_cmd.c                                : Command table for QSPI Flash device.
|-- ramckmdl.c                                : memory clear code
|-- remap.c                                   : SIC remap setting code
|-- rpchyperdrv.c                             : RPC driver code for HyperFlash
|-- rpcqspidrv.c                              : RPC driver code for QSPI Flash
|-- spiflash2drv.c                            : QSPI Flash driver code
|-- switch.c                                  : Dip-switch setting messages
|-- memory_icu_boot_for_S4.def                : Linker script for R-Car S4 (No dummy cert)
|-- memory_icu_boot_for_V4H.def               : Linker script for R-Car V4H (No dummy cert)
|-- memory_icu_boot_for_V4M.def               : Linker script for R-Car V4M (No dummy cert)
|-- memory_icu_boot_with_cert_for_S4.def      : Linker script for R-Car S4
|-- memory_icu_boot_with_cert_for_V4H.def     : Linker script for R-Car V4H
-- memory_icu_boot_with_cert_for_V4M.def      : Linker script for R-Car V4M
`-- makefile                                  : makefile
```

Figure 3-1 Module structure

3.3 Option setting

R-Car S4 R-Car V4H R-Car V4M

The Flash writer supports the following build option.

3.3.1 LSI

R-Car S4 R-Car V4H R-Car V4M

Select from the following table according to the using LSI.
If this option is not selected, the default value is S4.

Table 3-1 Build option LSI setting description

LSI	LSI setting
S4 [default]	Generate binary that works on R-Car S4 / S4N.
V4H	Generate binary that works on R-Car V4H.
V4M	Generate binary that works on R-Car V4M.

3.4 Detail of binary image

R-Car S4 R-Car V4H R-Car V4M

Describe the detail of the SCIF download image as follows:
In this case, boot from ICUMX and run Flash writer.

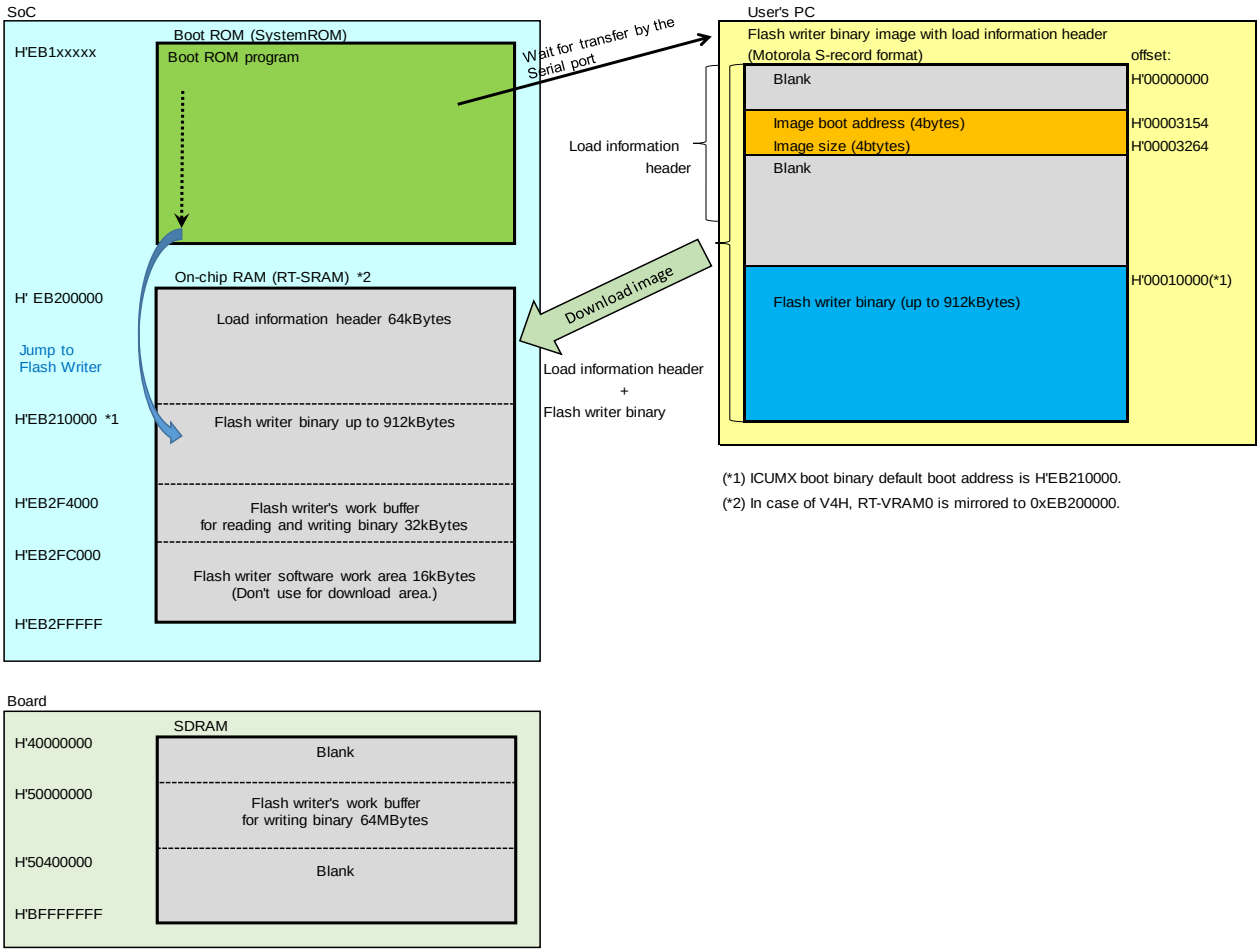

Figure 3-2 Detail of the SCIF download image

Following table shows Flash writer sample software's binary size.

Table 3-2 Binary size of Flash writer sample software

Name	Binary Size
R-Car S4 Flash writer sample software for S4	124 Kbytes
R-Car V4H Flash writer sample software for V4H	139 Kbytes
R-Car V4M Flash writer sample software for V4M	145 Kbyte

3.4.1 Release image

R-Car S4

Refer to Table 3-3 about information of release images when booted from QSPI Flash and HyperFlash.

Table 3-3 Information list of release images for S4 (QSPI Flash / HyperFlash)

Filename	Loaded by	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	BootROM	H'EB200000	H'000000	ICUMX IPL (Boot parameter)
icumx_loader.srec	BootROM	H'EB210000	H'040000	ICUMX IPL
cert_header_sa9.srec	ICUMX IPL	H'EB230000	H'240000	ICUMX IPL (Certificate)
dummy_fw.srec	ICUMX IPL	H'EB240000	H'280000	Dummy firmware (Instead of secure firmware)
dummy_rtos.srec	ICUMX IPL	H'E2100000	H'500000	Dummy RTOS (Instead of CR main OS)
AArch64_Dummy_CA55_Program.srec	ICUMX IPL	H'46400000	H'E00000	Dummy Secure monitor (Instead of Secure monitor or BL31)
AArch64_Dummy_CA55_Program2.srec	ICUMX IPL	H'50000000	H'F80000	Dummy U-Boot (Instead of U-Boot)
AArch64_Dummy_CA55_Program3.srec	ICUMX IPL	H'44100000	H'E80000	Dummy TEE OS (Instead of OP-TEE)
dummy_g4mh_case0.srec	ICUMX IPL	H'00000000	H'900000	Dummy G4MH program image
dummy_icumh_case1.srec	ICUMX IPL	H'00500000	H'380000	Dummy ICUMH program image

R-Car S4

Refer to Table 3-4 about information of release images when booted from eMMC.

Table 3-4 Information list of release images for S4 (eMMC)

Filename	Loaded by	Program Top Address	eMMC Save Partition	eMMC Save Sectors *1	Description
bootparam_sa0.srec	BootROM	H'EB200000	boot partition1	H'0000	ICUMX IPL (Boot parameter)
icumx_loader.srec	BootROM	H'EB210000	boot partition1	H'0080	ICUMX IPL
cert_header_sa9.srec	ICUMX IPL	H'EB230000	boot partition1	H'1200	ICUMX IPL (Certificate)
dummy_fw.srec	ICUMX IPL	H'EB240000	boot partition1	H'1400	Dummy firmware (Instead of secure firmware)
dummy_rtos.srec	ICUMX IPL	H'E2100000	boot partition1	H'2800	Dummy RTOS (Instead of CR main OS)
AArch64_Dummy_CA55_Program.srec	ICUMX IPL	H'46400000	boot partition1	H'7000	Dummy Secure monitor (Instead of Secure monitor or BL31)
AArch64_Dummy_CA55_Program2.srec	ICUMX IPL	H'50000000	boot partition1	H'7C00	Dummy U-Boot (Instead of U-Boot)
AArch64_Dummy_CA55_Program3.srec	ICUMX IPL	H'44100000	boot partition1	H'7400	Dummy TEE OS (Instead of OP-TEE)
dummy_g4mh_case0.srec	ICUMX IPL	H'00000000	boot partition1	H'4800	Dummy G4MH program image
dummy_icumh_case1.srec	ICUMX IPL	H'00500000	boot partition1	H'1C00	Dummy ICUMH program image

Note *1) 1 sector is 512 bytes.

Refer to following tables about information of release images when booted from QSPI Flash or HyperFlash and ICUMX IPL loads CX 2nd IPL.

R-Car S4

Table 3-5 Information list of release images for S4 when ICUMX IPL loads CX 2nd IPL (QSPI Flash / HyperFlash)

Filename	Loaded by	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	BootROM	H'EB200000	H'000000	ICUMX IPL (Boot parameter)
icumx_loader.srec	BootROM	H'EB210000	H'040000	ICUMX IPL
cert_header_sa9.srec	ICUMX IPL	H'EB230000	H'240000	ICUMX IPL (Certificate)
dummy_fw.srec	ICUMX IPL	H'EB240000	H'280000	Dummy firmware (Instead of secure firmware)
dummy_rtos.srec	ICUMX IPL	H'E2100000	H'500000	Dummy RTOS (Instead of CR main OS)
ca55_loader.srec	ICUMX IPL	H'E6300000	H'480000	CX 2nd IPL
dummy_g4mh_case0.srec	ICUMX IPL	H'00000000	H'900000	Dummy G4MH program image
dummy_icumh_case1.srec	ICUMX IPL	H'00500000	H'380000	Dummy ICUMH program image

R-Car S4

Table 3-6 Information list of release images for S4 when ICUMX IPL loads CX 2nd IPL (eMMC)

Filename	Loaded by	Program Top Address	eMMC Save Partition	eMMC Save Sectors *1	Description
AArch64_Dummy_CA55_Program.srec	CX 2nd IPL	H'46400000	boot partition1	H'7000	Dummy Secure monitor (Instead of Secure monitor or BL31)
AArch64_Dummy_CA55_Program3.srec	CX 2nd IPL	H'44100000	boot partition1	H'7400	Dummy TEE OS (Instead of OP-TEE)
AArch64_Dummy_CA55_Program2.srec	CX 2nd IPL	H'50000000	boot partition1	H'7C00	Dummy U-Boot (Instead of U-Boot)

Note *1) 1 sector is 512 bytes.

R-Car V4H R-Car V4M

Table 3-7 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (QSPI Flash)

Filename	Loaded by	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	BootROM	H'EB200000	H'000000	ICUMX IPL (Boot parameter)
icumx_loader.srec	BootROM	H'EB210000	H'040000	ICUMX IPL
cert_header_sa9.srec	ICUMX IPL	H'EB230000	H'240000	ICUMX IPL (Certificate)
dummy_fw.srec	ICUMX IPL	H'EB240000	H'280000	Dummy firmware (Instead of secure firmware)
cr52_loader.srec	ICUMX IPL	H'E6300000	H'480000	CX 2nd IPL

R-Car V4H R-Car V4M

Table 3-8 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (eMMC)

Filename	Loaded by	Program Top Address	eMMC Save Partition	eMMC Save Sectors *1	Description
dummy_rtos.srec	CX 2nd IPL	H'E2100000	boot partition1	H'0000	Dummy RTOS (Instead of CR main OS)
AArch64_Dummy_CA76_Program.srec	CX 2nd IPL	H'46400000	boot partition1	H'A000	Dummy Secure monitor (Instead of Secure monitor or BL31)
AArch64_Dummy_CA76_Program2.srec	CX 2nd IPL	H'50000000	boot partition1	H'AC00	Dummy U-Boot (Instead of U-Boot)

Note *1) 1 sector is 512 bytes.

3.4.2 Data format of Motorola S-record file

R-Car S4 R-Car V4H R-Car V4M

Refer to Figure 3-3 about data format of Motorola S-record file.

Example of transfer data format
(Motorola S-record file)

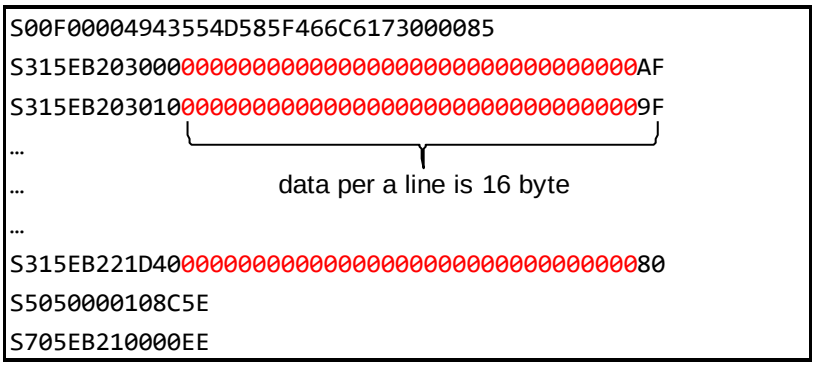

Figure 3-3 Example of Motorola S-record file

3.5 Command specification

R-Car S4 R-Car V4H R-Car V4M

The following table shows the command list.

Table 3-9 Command list

Command	Description
XRS	Read the QSPI Flash and HyperFlash memory data.
XLS2	Write S-record format images to the QSPI Flash and HyperFlash memory.
XLS3	Write raw binary images to the QSPI Flash and HyperFlash memory.
XCS	Erase the QSPI Flash and HyperFlash memory.
EM_DCID	Display the CID registers of eMMC.
EM_DCSD	Display the CSD registers of eMMC.
EM_DECSD	Display the EXT_CSD registers of eMMC.
EM_SECSD	Modify the EXT_CSD registers of eMMC.
EM_R	Read the eMMC memory data.
EM_W	Write S-record format images to the user data area of eMMC and the boot partition of eMMC.
EM_WB	Write raw binary images to the user data area of eMMC and the boot partition of eMMC.
EM_E	Erase the user data area of eMMC, and the boot partition of eMMC.
H	Display the command help.

3.6 How to use command

R-Car S4 R-Car V4H R-Car V4M

Describe how to use command listed in Table 3-9.

3.6.1 Common procedure

R-Car S4 R-Car V4H R-Car V4M

The following procedure is common to all commands.
Note) Refer to 5. How to run Flash writer for more detail of how to run Flash writer.

3.6.1.1 Step1 Connect cable

Connect USB Host connector of Windows Host PC that is virtual COM port to System Evaluation Board with USB cable for displaying console.
Note) About where connect USB cable on System Evaluation Board, refer to 2.1 Hardware Environment.

3.6.1.2 Step2 Setting the terminal software

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings.

R-Car S4
setting value : baud rate 1843200, 8bit data, parity none, stop 1 bit, flow control none.

R-Car V4H R-Car V4M
setting value : baud rate 921600, 8bit data, parity none, stop 1 bit, flow control none.

3.6.1.3 Step3 Start Flash writer

1. Set dip switch "SCIF download mode".
   About switch setting, refer to 5.1 Prepare for writing to the external flash memory.
2. Turn on the System Evaluation Board.
3. Send Flash writer binary to the terminal software.

```
SCIF Download mode (w/o verification)
 (C) Renesas Electronics Corp.

-- Load Program to RT-SRAM ---------------
please send !

Flash writer for R-Car SoC Series Rev.X.X.X MM.DD,YYYY
>
```

Figure 3-4 Log output of common procedure

3.6.2 External flash memory dip switch setting instruction list

R-Car S4 R-Car V4H R-Car V4M

The following table shows what to do when specific instruction is displayed.

R-Car S4

Table 3-10 Switch setting instruction list for external flash memory (S4)

SoC	Device	To do
S4	QSPI Flash 512Mbit	After “SW1 1pin-Side! Setting OK? (Push Y key)” is displayed, input “y”. After “SW2 Center! Setting OK? (Push Y key)” is displayed, input “y”.
	HyperFlash	After “SW2 1pin-Side! Setting OK? (Push Y key)” is displayed, input “y”.
	eMMC	After “SW3 Center! Setting OK? (Push Y key)” is displayed, input “y”.

R-Car V4H R-Car V4M

Table 3-11 Switch setting instruction list for external flash memory (V4H/V4M)

SoC	Device	To do
V4H V4M	QSPI Flash 512Mbit	After “SW13 1pin-Side! Setting OK? (Push Y key)” is displayed, input “y”. After “SW57 Center! Setting OK? (Push Y key)” is displayed, input “y”.
	eMMC	After “SW43 Center! Setting OK? (Push Y key)” is displayed, input “y”.

3.6.3 XRS : Read QSPI Flash and HyperFlash memory data

R-Car S4 R-Car V4H R-Car V4M

This command displays QSPI Flash and HyperFlash memory data.

The following shows the procedure of this command.
Note) The following input value is an example.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to reading device. Refer to 5.1 about dip switch setting.
Execute xrs command.

```
>xrs
===== Qspi/HyperFlash reading Command ========
Read Data from Spiflash
Please select,FlashMemory.
   1 : QspiFlash_512Mbit (U6 : S25FS512S)
   2 : QspiFlash Board   (CN3: S25FS512S)
   3 : HyperFlash        (U26: S26KS512S)
  Select (1-3)>
```

Figure 3-5 Log output when execute XRS command(1)

Input number of the external flash memory to read.
Refer to 3.6.2 about switch setting instruction.

- After "Please Input Reading Top Address" is displayed, input the start reading address in hexadecimal.
- After "Please Input Reading Size" is displayed, input the reading size in hexadecimal.

Note) The top address of QSPI Flash is 0x08000000, and end address is 0x0BFFFFFF on S25FS512S and S26KS512S.
Note) Input address must be 256bytes align.
Note) The size of enable to read at once is up to 16Kbytes.
Note) Input size must be 16bytes units.
[example]

```
===== Please Input Reading Top Address ============
  Please Input : H'08000000
===== Please Input Reading Size ============
  Please Input : H'200

  offset | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
---------+-------------------------------------------------
08000000 | 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00
08000010 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
08000020 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
...
080001D0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
080001E0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 3-6 Log output when execute XRS command(2)

3.6.4 XLS2 : Write S-record format images to the QSPI Flash and HyperFlash

R-Car S4 R-Car V4H R-Car V4M

This command writes the S-record format image to the QSPI Flash and HyperFlash.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute xls2 command.

```
>xls2
===== Qspi/HyperFlash writing Command =============
Load Program to Spiflash
Writes to any of SPI address.
Please select,FlashMemory.
   1 : QspiFlash_512Mbit (U6 : S25FS512S)
   2 : QspiFlash Board   (CN3: S25FS512S)
   3 : HyperFlash        (U26: S26KS512S)
  Select (1-3)>
```

Figure 3-7 Log output when execute XLS2 command

Input number of the external flash memory to write.
Refer to 3.6.2 about switch setting instruction.
Input data and transmit file

- After "Please Input Program Top Address" is displayed, input Program Top Address in
  R-Car S4 Table 3-3 or Table 3-5 and "Enter".
  R-Car V4H R-Car V4M Table 3-7 and "Enter".
- After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address in
  R-Car S4 Table 3-3 or Table 3-5 and "Enter".
  R-Car V4H R-Car V4M Table 3-7 and "Enter".
- After "please send !" is displayed, in case of Tera Term, transmit srec file in
  R-Car S4 Table 3-3 or Table 3-5 and "Enter".
  R-Car V4H R-Car V4M Table 3-7 and "Enter".

If there are any data in writing area, "SPI Data Clear(H'FF) Check :H'start address-end address Clear OK?(y/n)" is displayed. Then input "y".
After "SAVE SPI-FLASH..... complete!" is displayed, the prompt returns. It means finished writing data.
Please repeat xls2 command if other files are needed to write.

3.6.5 XLS3 : Write raw binary images to the QSPI Flash and HyperFlash

R-Car S4 R-Car V4H R-Car V4M

This command writes the raw binary image to QSPI Flash and HyperFlash.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute xls3 command.

```
>xls3
===== Qspi/HyperFlash writing Command =============
Load Program to Spiflash
Writes to any of SPI address.
Please select,FlashMemory.
   1 : QspiFlash_512Mbit (U6 : S25FS512S)
   2 : QspiFlash Board   (CN3: S25FS512S)
   3 : HyperFlash        (U26: S26KS512S)
  Select (1-3)>
```

Figure 3-8 Log output when execute XLS2 command

Input number of the QSPI Flash and HyperFlash to write.
Refer to 3.6.2 about switch setting instruction.
Input data and transmit file

- After "Please Input Program size" is displayed, input Program size in hexadecimal and "Enter".
- After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address in R-Car S4 Table 3-3 or Table 3-5 and "Enter".
  R-Car V4H R-Car V4M Table 3-7 and "Enter".
- After "please send !" is displayed, in case of Tera Term, transmit binary file by "File -> Send file...".
  Note) Please check the box "Binary" at Option in "Send file..." menu.

If there are any data in writing area, "SPI Data Clear(H'FF) Check :H'start address-end address Clear OK?(y/n)" is displayed. Then input "y".
After "SAVE SPI-FLASH..... complete!" is displayed, the prompt returns. It means finished writing data.
Please repeat xls3 command if other files are needed to write.

3.6.6 XCS : Erase the QSPI Flash and HyperFlash

R-Car S4 R-Car V4H R-Car V4M

This command erases all sectors of QSPI Flash and HyperFlash.
The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to erasing device. Refer to 5.1 about dip switch setting.
Execute xcs command.

```
>xcs
ALL ERASE SpiFlash or HyperFlash memory
Clear OK?(y/n)              Please select,FlashMemory.
   1 : QspiFlash_512Mbit (U6 : S25FS512S)
   2 : QspiFlash Board   (CN3: S25FS512S)
   3 : HyperFlash        (U26: S26KS512S)
  Select (1-3)>
```

Figure 3-9 Log output when execute XCS command

Input number of the external flash memory to erase.
Refer to 3.6.2 about switch setting instruction.
After "complete!" is displayed, the prompt returns. It means finished erasing selected Flash memory data.
Note) It takes time to erase QSPI Flash memory (512Mbit) data. Please wait without sending other commands until "complete!" is displayed.

3.6.7 EM_DCID : Display the CID registers of eMMC

R-Car S4 R-Car V4H R-Car V4M

This command displays the contents of the CID registers of the eMMC.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_dcid command.

[example]

```
>em_dcid

[CID Field Data]
[127:120]  MID  0x13
[113:112]  CBX  0x01
[111:104]  OID  0x4E
[103: 56]  PNM  0x47314A333745
[ 55: 48]  PRV  0x10
[ 47: 16]  PSN  0x0A0C4571
[ 15:  8]  MDT  0x97
[  7:  1]  CRC  0x00
```

Figure 3-10 Log output when execute EM_DCID command

3.6.8 EM_DCSD : Display the CSD registers of eMMC

R-Car S4 R-Car V4H R-Car V4M

This command displays the contents of the CSD registers of the eMMC.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_dcsd command.

[example]

```
>em_dcsd

[CSD Field Data]
[127:126]  CSD_STRUCTURE       0x03
[125:122]  SPEC_VERS           0x04
[119:112]  TAAC                0x7F
...
[ 11: 10]  FILE_FORMAT         0x00
[  9:  8]  ECC                 0x00
[  7:  1]  CRC                 0x00
```

Figure 3-11 Log output when execute EM_DCSD command

3.6.9 EM_DECSD : Display the EXT_CSD registers of eMMC

R-Car S4 R-Car V4H R-Car V4M

This command displays the contents of the EXT_CSD registers of the eMMC.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_decsd command.

[example]

```
>em_decsd

[EXT_CSD Field Data]
[505:505]  EXT_SECURITY_ERR                    0x00
[504:504]  S_CMD_SET                           0x01
[503:503]  HPI_FEATURES                        0x01
...
[142:140]  ENH_SIZE_MULT                       0x000000
[139:136]  ENH_START_ADDR                      0x00000000
[134:134]  SEC_BAD_BLK_MGMNT                   0x00
```

Figure 3-12 Log output when execute EM_DECSD command

3.6.10 EM_SECSD : Modify the EXT_CSD registers of eMMC

R-Car S4 R-Car V4H R-Car V4M

This command modifies the contents of the registers of EXT_CSD of the eMMC.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_secsd command.

```
>em_secsd
  Please Input EXT_CSD Index(H'00 - H'1FF) :
```

Figure 3-13 Log output when execute EM_SECSD command (1)

Enter the address of the EXT_CSD register in hexadecimal.

```
>em_secsd
  Please Input EXT_CSD Index(H'00 - H'1FF) :b1
  EXT_CSD[B1] = 0x0A
  Please Input Value(H'00 - H'FF) :
```

Figure 3-14 Log output when execute EM_SECSD command (2)

Enter the settings of EXT_CSD register in hexadecimal.

```
>em_secsd
  Please Input EXT_CSD Index(H'00 - H'1FF) :b1
  EXT_CSD[B1] = 0x0A
  Please Input Value(H'00 - H'FF) :8
  EXT_CSD[B1] = 0x08
```

Figure 3-15 Log output when execute EM_SECSD command (3)

The EXT_CSD register has been modified.

3.6.11 EM_R : eMMC memory data

R-Car S4 R-Car V4H R-Car V4M

This command displays eMMC memory data.

The following shows the procedure of this command.
Note) The following input value is an example.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to reading device. Refer to 5.1 about dip switch setting.
Execute em_r command.
Refer to 3.6.2 about switch setting instruction.

```
>em_r
EM_R Start --------------
------------------------------------------------------------
Please select,eMMC Partition Area.
 0:User Partition Area    : 31080448 KBytes
  eMMC Sector Cnt : H'0 - H'03B47FFF
 1:Boot Partition 1       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
 2:Boot Partition 2       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
------------------------------------------------------------
  Select area(0-2)>
```

Figure 3-16 Log output when execute EM_R command(1)

Input number of the partition to read.
- After "Please Input Start Address in sector" is displayed, input the start reading sector in hexadecimal.
- After "Please Input Size in sector" is displayed, input the reading sector size in hexadecimal.

Note) The top sector of eMMC is 0x0, and end sector is referred to first log output.
Note) The size of enable to read at once is up to 16Kbytes (20 sectors).
[example]

```
-- Boot Partition 1 Program -----------------------------
Please Input Start Address in sector :0
Please Input Size in sector :1

partition:00000001
 offset   | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
---------+--------------------------------------------------
00000000 | 01 F6 02 E0 00 00 00 00 00 00 00 00 00 00 00 00
00000010 | 77 FE 02 E0 00 00 00 00 00 00 00 00 00 00 00 00
...
000001E0 | 76 2E 02 E0 00 00 00 00 00 00 00 00 00 00 00 00
000001F0 | 76 1E 02 E0 08 00 00 00 18 00 10 00 28 00 20 00
```

Figure 3-17 Log output when execute EM_R command(2)

3.6.12 EM_W : Write S-record format images to eMMC

R-Car S4 R-Car V4H R-Car V4M

This command writes the S-record format image to any partition of eMMC.

The following shows the procedure of this command.
Note) In case of V4H, the sum of eMMC Save Sectors and binary size is up to 0xFC00 sector.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_w command.
Refer to 3.6.2 about switch setting instruction.

```
>em_w
EM_W Start --------------
-----------------------------------------------------------
Please select,eMMC Partition Area.
 0:User Partition Area    : 31080448 KBytes
  eMMC Sector Cnt : H'0 - H'03B47FFF
 1:Boot Partition 1       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
 2:Boot Partition 2       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
-----------------------------------------------------------
  Select area(0-2)>
```

Figure 3-18 Log output when execute EM_W command

Input data and transmit file.

- Input number of the eMMC Save Partition in Table 3-8.
- After "Please Input Start Address in sector :" is displayed, input eMMC Save Sectors in
  R-Car S4 Table 3-4 or Table 3-6 and "Enter".
  R-Car V4H R-Car V4M Table 3-8 and "Enter".
- After "Please Input Program Start Address : " is displayed, input Program Top Address in
  R-Car S4 Table 3-4 or Table 3-6 and "Enter".
  R-Car V4H R-Car V4M Table 3-8 and "Enter".
- After "please send !" is displayed, in case of Tera Term, transmit srec file in
  R-Car S4 Table 3-4 or Table 3-6 and "Enter".
  R-Car V4H R-Car V4M Table 3-8 and "Enter".

After "EM_W Complete!" is displayed, the prompt returns. It means finished writing data.
Please repeat em_w command if other files are needed to write.

3.6.13 EM_WB : Write raw binary images to eMMC

R-Car S4 R-Car V4H R-Car V4M

This command writes the raw binary image to any partition of the eMMC.

The following shows the procedure of this command.
Note) In case of V4H, the sum of eMMC Save Sectors and binary size is up to 0xFC00 sector.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to writing device. Refer to 5.1 about dip switch setting.
Execute em_wb command.
Refer to 3.6.2 about switch setting instruction.

```
>em_wb
EM_W Start --------------
-----------------------------------------------------
Please select,eMMC Partition Area.
 0:User Partition Area    : 31080448 KBytes
  eMMC Sector Cnt : H'0 - H'03B47FFF
 1:Boot Partition 1       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
 2:Boot Partition 2       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
-----------------------------------------------------
  Select area(0-2)>
```

Figure 3-19 Log output when execute EM_WB command

Input data and transmit file.

- Input number of the eMMC Save Partition in Table 3-8.
- After "Please Input Start Address in sector :" is displayed, input eMMC Save Sectors in
  R-Car S4 Table 3-4 or Table 3-6 and "Enter".
  R-Car V4H R-Car V4M Table 3-8 and "Enter".
- After "Please Input File size(byte) :" is displayed, input Program size in hexadecimal and "Enter".
- After "please send binary file!" is displayed, in case of Tera Term, transmit binary file by "File -> Send file...".
  Note) Please check the box "Binary" at Option in "Send file..." menu.

After "EM_WB Complete!" is displayed, the prompt returns. It means finished writing data.
Please repeat em_wb command if other files are needed to write.

3.6.14 EM_E : Erase eMMC

R-Car S4 R-Car V4H R-Car V4M

This command erases any partition of the eMMC.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to erasing device. Refer to 5.1 about dip switch setting.
Execute em_e command.
Refer to 3.6.2 about switch setting instruction.

```
>em_e
EM_E Start --------------
------------------------------------------------------------
Please select,eMMC Partition Area.
 0:User Partition Area    : 31080448 KBytes
  eMMC Sector Cnt : H'0 - H'03B47FFF
 1:Boot Partition 1       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
 2:Boot Partition 2       : 32256 KBytes
  eMMC Sector Cnt : H'0 - H'0000FBFF
------------------------------------------------------------
  Select area(0-2)>
```

Figure 3-20 Log output when execute EM_E command

Input number of the eMMC partition to erase.
After "EM_E Complete!" is displayed, the prompt returns. It means finished erasing selected eMMC partition.

3.6.15 H : Display the command help

R-Car S4 R-Car V4H R-Car V4M

Displays a description of the commands.

The following shows the procedure of this command.

Do common procedure written in 3.6.1.1, 3.6.1.2 and 3.6.1.3.
Set dip switch according to erasing device. Refer to 5.1 about dip switch setting.
Execute h command.

```
>h
        HyperFlash/SPI Flash write command
 XRS            read HyperFlash/SPI Flash data
 XCS            erase program to HyperFlash/SPI Flash
 XLS2           write program to HyperFlash/SPI Flash
 XLS3           write program to HyperFlash/SPI Flash(Binary)

        eMMC write command
 EM_DCID        display register CID
 EM_DCSD        display register CSD
 EM_DECSD       display register EXT_CSD
 EM_SECSD       change register EXT_CSD byte
 EM_R           read eMMC data
 EM_W           write program to eMMC
 EM_WB          write program to eMMC (Binary)
 EM_E           erase program to eMMC
 H              help
>
```

Figure 3-21 Log output when execute H command

4. How to build the Flash writer

R-Car S4 R-Car V4H R-Car V4M

This chapter is described how to build the Flash writer.

4.1 Prepare the Compiler

R-Car S4 R-Car V4H R-Car V4M

GHS compiler is required to build. Please prepare on your own.
Install the GHS compiler according to Green Hills Software Products Installation Guide (GHS License Package).

4.2 Prepare the Cygwin

R-Car S4 R-Car V4H R-Car V4M

Download the Cygwin installer from the site shown below.
Cygwin (https://www.cygwin.com/)
Launch the downloaded installer and install according to the instructions of the installer.
Install at least the package "make: The GNU version of the 'make' utility".

4.3 Prepare the source code

R-Car S4 R-Car V4H R-Car V4M

Decompress release package.
Move to directory of Flash writer source code.

```
$ cd RTM8RC4000ZLPL0S00JPZ0E/src/flash_writer
```

Figure 4-1 example of move to Flash writer

4.4 Build the Flash writer binary

R-Car S4 R-Car V4H R-Car V4M

S-record file is built by the following command on Cygwin.
Note) Please set the path and license to the compiler beforehand properly.

R-Car S4

```
$ make clean
$ make LSI=S4
```

Figure 4-2 example of building Flash writer for S4

Output the following image.

- ./flash_writer/ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_S4.mot

R-Car V4H

```
$ make clean
$ make LSI=V4H
```

Figure 4-3 example of building Flash writer for V4H

Output the following image.

- ./flash_writer/ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_V4H.mot

R-Car V4M

```
$ make clean
$ make LSI=V4M
```

Figure 4-4 example of building Flash writer for V4M

Output the following image.

- ./flash_writer/ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_V4M.mot

5. How to run Flash writer

5.1 Prepare for writing to the external flash memory

R-Car S4 R-Car V4H R-Car V4M

Start the target in the SCIF download mode and run Flash writer sample software.

5.1.1 In case of setting by CPU board (SW BOARD M)

R-Car S4 R-Car V4H R-Car V4M

The following table shows the MD Setting for SCIF download mode.

R-Car S4

Table 5-1 MD setting for SCIF download mode for S4

Item	MD1	MD2	MD3	MD4	MD6	MD7	MD31	MD32
MD Setting	OFF (1)	OFF (1)	OFF (1)	OFF (1)	ON (0)	OFF (1)	ON (0)	OFF (1)

Note) About external flash memory select setting, refer to Number 4 and 5 in Table 1-2.

Connect the Host PC to CN21 connector using the USB cable.
The following table shows the setting of terminal software.

Table 5-2 Terminal software configuration for S4 when SCIF download mode

Item	Value
Download mode	HSCIF
Baud rate	1843200bps
Data bit length	8bits
Parity check	none
Stop bits	1bit
Flow control	none
Line feed code (Transmit)	CR

R-Car V4H R-Car V4M

Table 5-3 MD setting for SCIF download mode for V4H

Item	MD1	MD2	MD3	MD4	MD6	MD7	MD31	MD32
MD Setting	OFF (1)	OFF (1)	OFF (1)	OFF (1)	ON (0)	OFF (1)	OFF (1)	ON (0)

Note) About external flash memory select setting, refer to Number 6 and 7 in Table 1-2.

Connect the Host PC to CN10 connector using the USB cable.
The following table shows the setting of terminal software.

Table 5-4 Terminal software configuration for V4H when SCIF download mode

Item	Value
Download mode	HSCIF
Baud rate	921600bps
Data bit length	8bits
Parity check	none
Stop bits	1bit
Flow control	none
Line feed code (Transmit)	CR

R-Car S4 R-Car V4H R-Car V4M
Terminal software outputs the following log at power ON the target.

```
SCIF Download mode (w/o verification)
(C) Renesas Electronics Corp.

-- Load Program to RT-SRAM ---------------
please send !
```

Figure 5-1 log output of SCIF Download mode

Transfer S-record file after the log output.
S-record file for ICUMX on

- R-Car S4 : ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_S4.mot
- R-Car V4H : ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_V4H.mot
- R-Car V4M : ICUMX_output/ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_V4M.mot

When the transfer is successful, the following log is output.

```
Flash writer for R-Car SoC Series Rev.X.X.X. MM.DD,YYYY
>
```

Figure 5-2 log output of Flash writer

Please enter the any key from the console after starting Flash writer.
R-Car S4 To use SCIF, enter the key from the console connected to CN21.
R-Car V4H R-Car V4M To use SCIF, enter the key from the console connected to CN10.
Note) After entering the key, the other console becomes unusable.

For details about how to write to the external flash memory, please refer to Section 3.6.4 and 3.6.12.

5.1.2 In case of setting by CPLD tool (S4)

R-Car S4

The following shows the switch setting by CPLD for SCIF download mode.

1. Set terminal software configuration referring to Table 5-2.
2. Connect the USB cable between PC and CN21 on S4 board.
3. Turn on the S4 board.
4. Launch the CPLD tool, "S4_Spider_Configurator.exe".
5. According to "How_to_use_S4_CPLD_rev002.pdf", set up the CPLD tool.
6. Referring the Figure 5-3, put a check on the "SCIF/HSCIF0" on "Boot from [MD4/3/2/1]" and "HSCIF0(1'843'200 bd)" on "SCIF speed [MD32/31]" setting.
7. Click the "Write CPLD (non-volatile)".
8. Press the white reset button (SW15).
9. Transfer S-record file "ICUMX_Flash_writer_SCIF_DUMMY_CERT_EB203000_S4.mot"
10. Please enter the any key from the console after starting Flash writer.

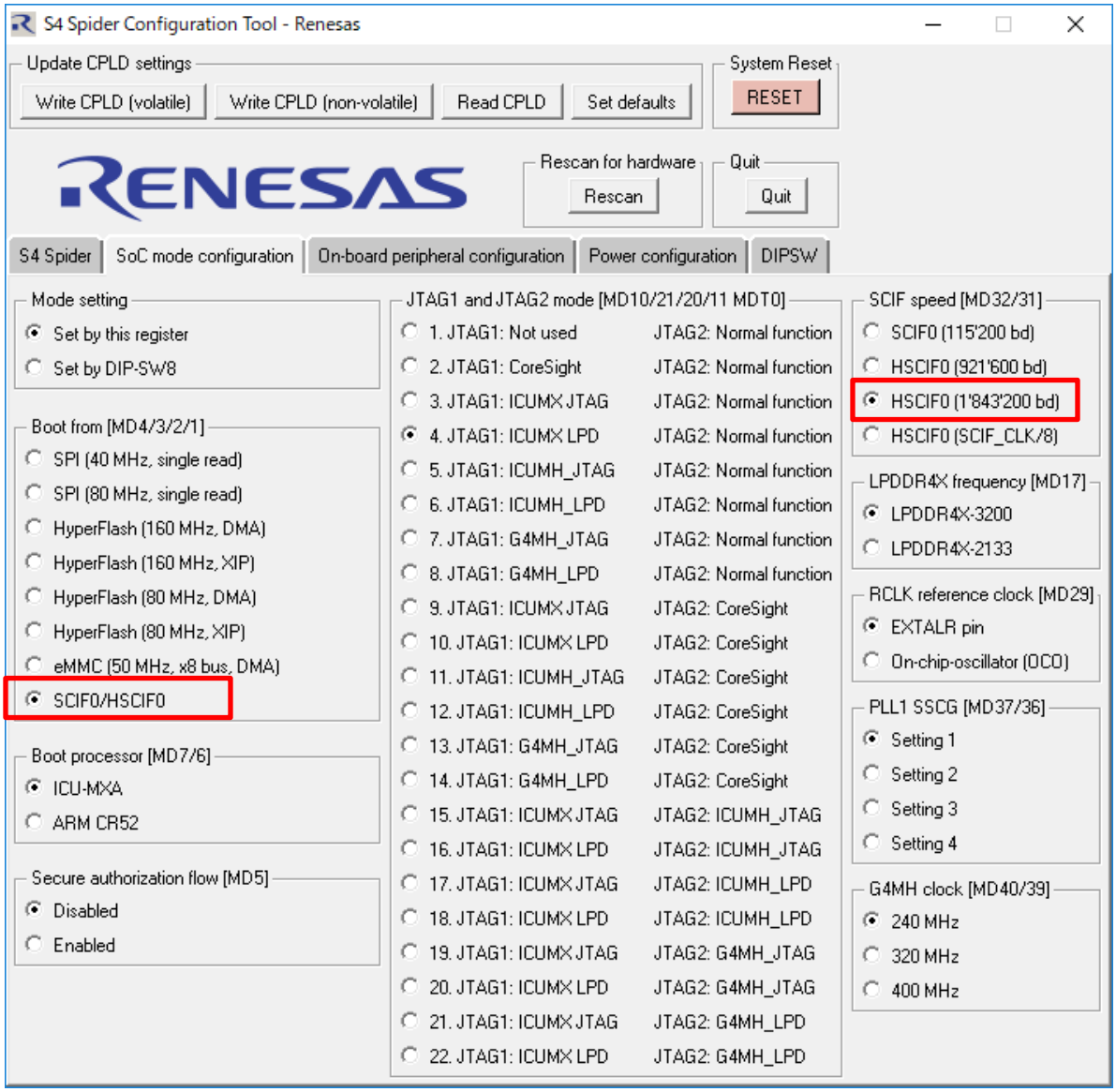

Figure 5-3 CPLD tool setting

5.2 Prepare for booting from the external flash memory

5.2.1 In case of setting by CPU board (SW BOARD M)

R-Car S4

To boot from the external flash memory, need to change the MD Setting from Table 5-1.
The following table shows the MD Setting for each external flash memory boot mode.

Table 5-5 MD setting for booting from external flash memory (S4)

Device	MD1	MD2	MD3	MD4	MD6	MD7	MD31	MD32
QSPI Flash (40 MHz)	ON (0)	ON (0)	OFF (1)	ON (0)	ON (0)	OFF (1)	Refer to Table 5-6	
HyperFlash (160 MHz)	ON (0)	OFF (1)	ON (0)	ON (0)	ON (0)	OFF (1)		
eMMC (50 MHz)	OFF (1)	ON (0)	OFF (1)	OFF (1)	ON (0)	OFF (1)		

Note) About external flash memory select setting, refer to Number 4 and 5 in Table 1-2.

Connect the Host PC to CN21 connector using the USB cable.
The following table shows the setting of terminal software.

Table 5-6 Terminal software configuration for S4 when boot from the external flash memory

	MD setting	
Item	**SCIF (MD[32:31]=b'00)**	**HSCIF (MD[32:31]=b'10)**
Baud rate	115200bps	1843200bps
Data bit length	8bits	8bits
Parity check	none	none
Stop bits	1bit	1bit
Flow control	none	none

R-Car V4H R-Car V4M

To boot from the external flash memory, need to change the MD Setting from Table 5-3.
The following table shows the MD Setting for each external flash memory boot mode.

Table 5-7 MD setting for booting from external flash memory (V4H)

Device	MD1	MD2	MD3	MD4	MD6	MD7	MD31	MD32
QSPI Flash (40 MHz)	ON (0)	ON (0)	OFF (1)	ON (0)	ON (0)	OFF (1)	Refer to Table 5-8	
eMMC (50 MHz)	OFF (1)	ON (0)	OFF (1)	OFF (1)	ON (0)	OFF (1)		

Note) About external flash memory select setting, refer to Number 6 and 7 in Table 1-2.

Connect the Host PC to CN10 connector using the USB cable.
The following table shows the setting of terminal software.

Table 5-8 Terminal software configuration for V4H when boot from the external flash memory

	MD setting	
Item	**SCIF (MD[32:31]=b'00)**	**HSCIF (MD[32:31]=b'01)**
Baud rate	115200bps	921600bps
Data bit length	8bits	8bits
Parity check	none	none
Stop bits	1bit	1bit
Flow control	none	none

5.2.2 In case of setting by CPLD tool (S4)

R-Car S4

The following shows the switch setting by CPLD for booting from QSPI Flash.

1. Set terminal software configuration referring to Table 5-6.
2. Referring the Figure 5-4, put a check on the "SPI (40MHz, single read)" on "Boot from [MD4/3/2/1]" and “SCIF0(115’200 bd)” in “SCIF speed [MD32/31]” setting. *1
3. Click the "Write CPLD (non-volatile)".
4. Press the white reset button (SW15).

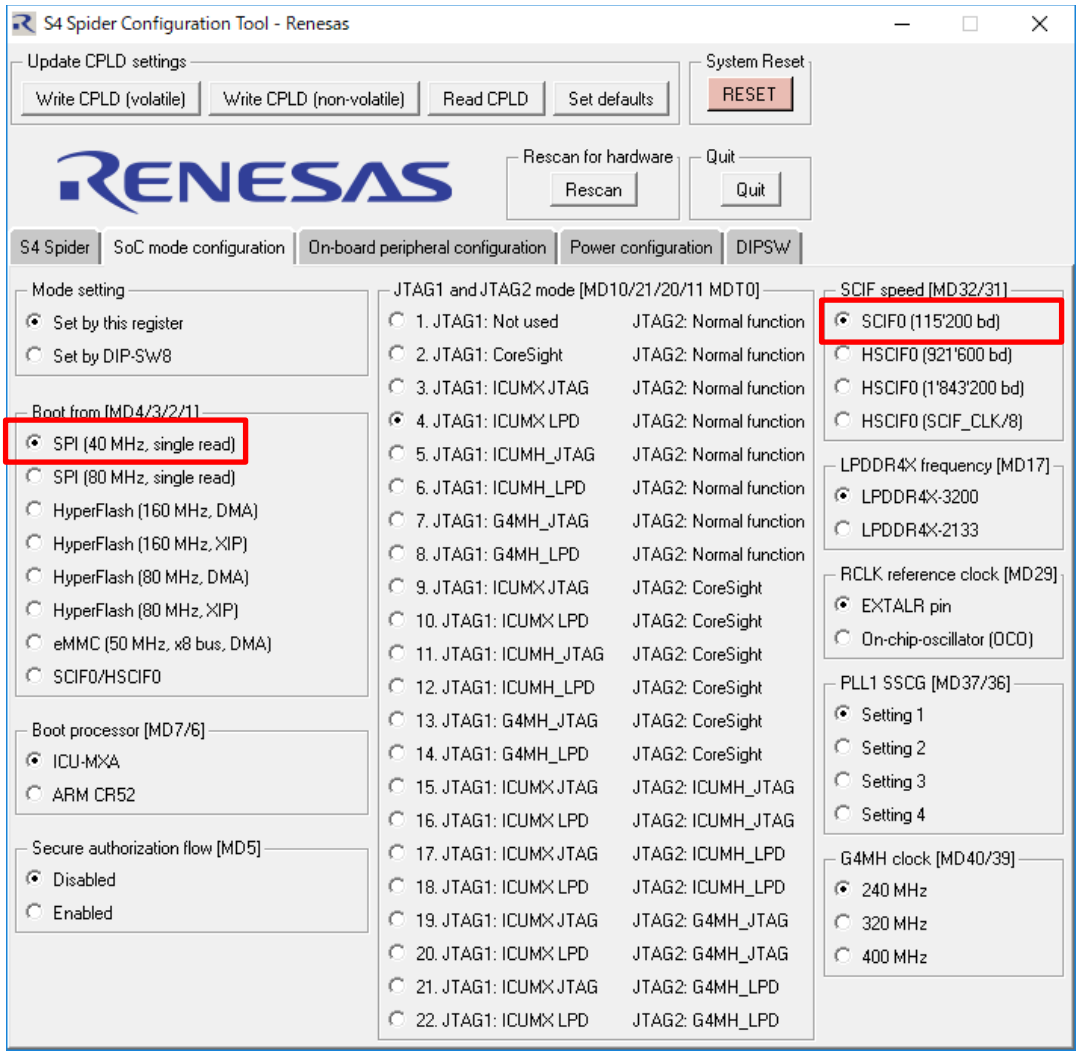

Figure 5-4 CPLD setting for booting from QSPI Flash

Note *1) If boot on HSCIF 1843200, put a check to ‘HSCIF0(1’843’200 bd)’ in ‘SCIF speed [MD32/31]’ setting.

The following shows the switch setting by CPLD for booting from HyperFlash.

1. Set terminal software configuration referring to Table 5-6.
2. Referring the Figure 5-5, put a check on the "HyperFlash (160MHz, DMA)" on "Boot from [MD4/3/2/1]" and “SCIF0(115’200 bd)” in “SCIF speed [MD32/31]” setting. *1
3. Click the "Write CPLD (non-volatile)".
4. Press the white reset button (SW15).

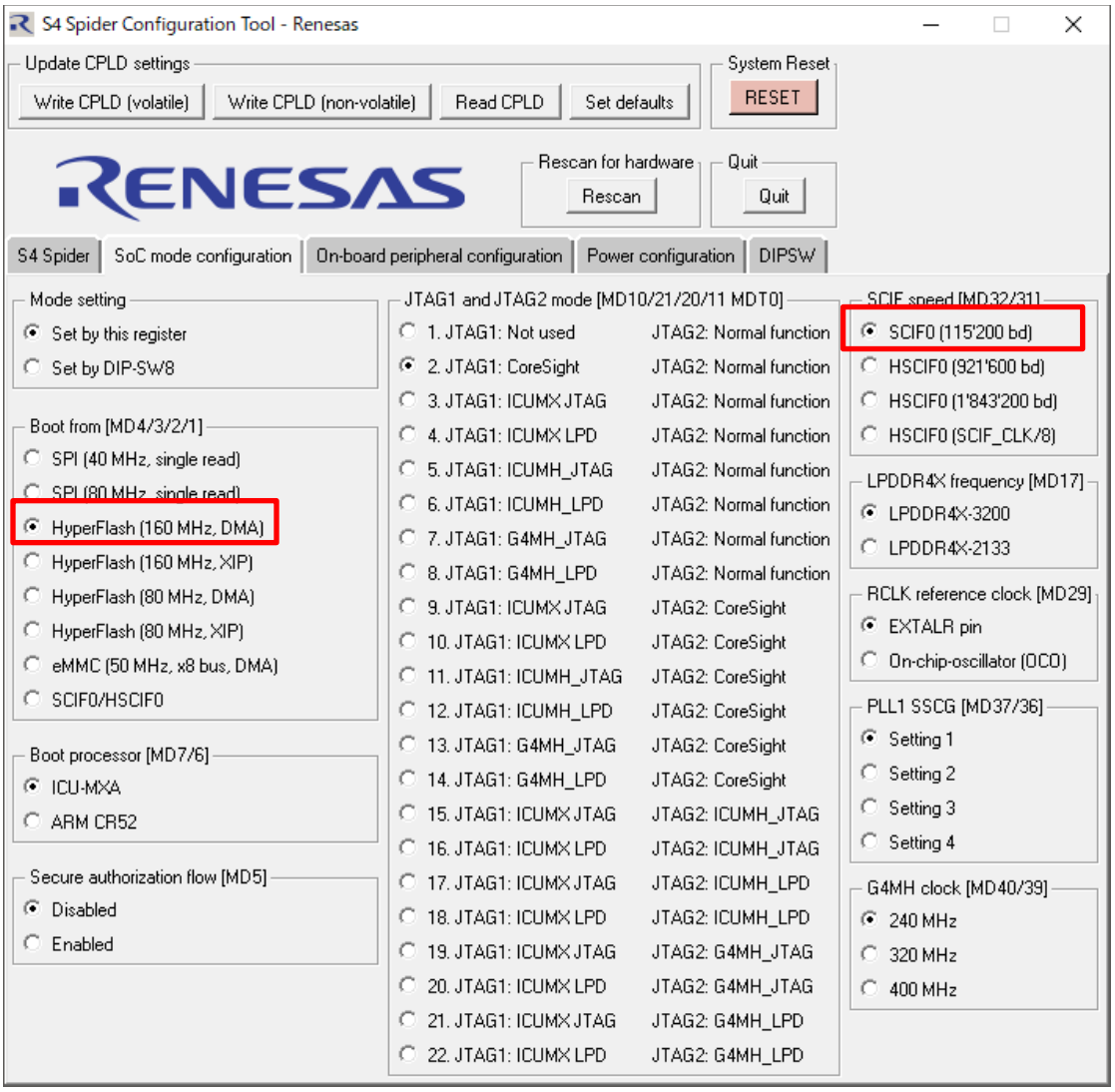

Figure 5-5 CPLD setting for booting from HyperFlash

Note *1) If boot on HSCIF 1843200, put a check to ‘HSCIF0(1’843’200 bd)’ in ‘SCIF speed [MD32/31]’ setting.

The following shows the switch setting by CPLD for booting from eMMC.

1. Set terminal software configuration referring to Table 5-6.
2. Referring the Figure 5-6, put a check on the "eMMC (50MHz, x8 bus, DMA)" on "Boot from [MD4/3/2/1]" and “SCIF0(115’200 bd)” on “SCIF speed [MD32/31]” setting. *1
3. Referring the Figure 5-7, put a check on the “on-board eMMC” on “eMMC” setting.
4. Click the “Write CPLD (non-volatile)”.
5. Press the white reset button (SW15).

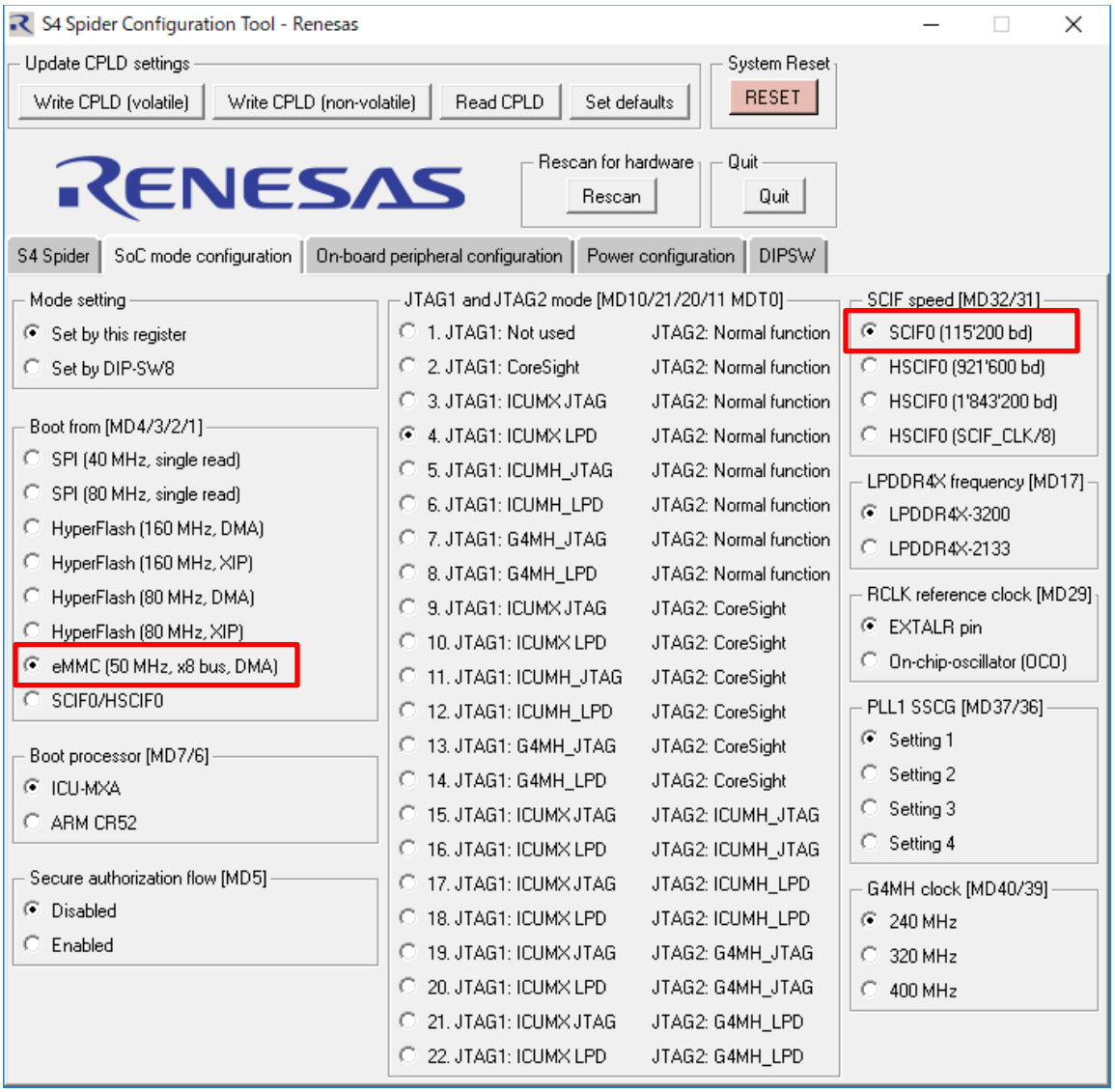

Figure 5-6 CPLD setting for booting from eMMC (1)

Note *1) If boot on HSCIF 1843200, put a check to ‘HSCIF0(1’843’200 bd)’ in ‘SCIF speed [MD32/31]’ setting.

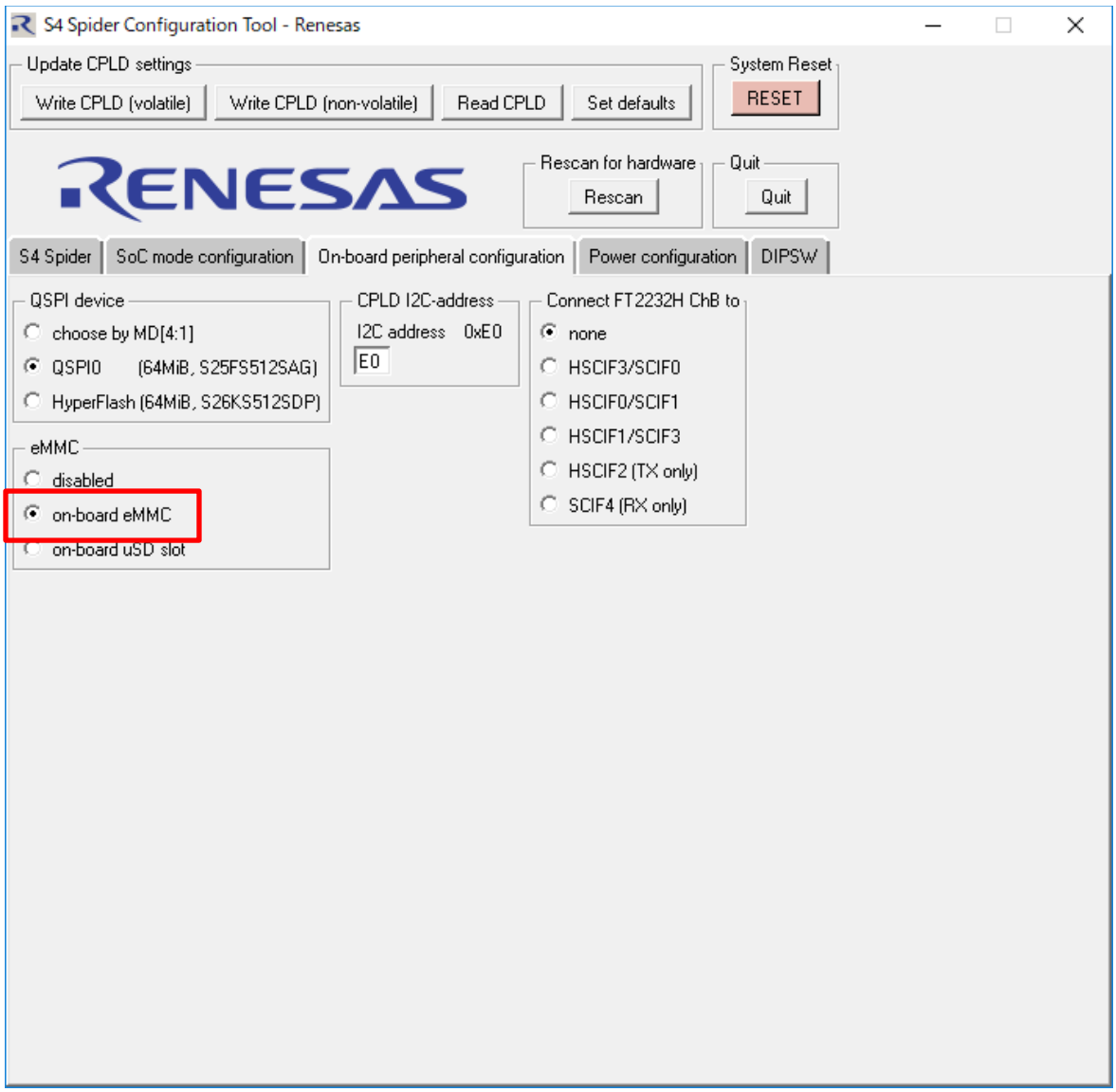

Figure 5-7 CPLD setting for booting from eMMC (2)

6. Error case to handle

R-Car S4 R-Car V4H R-Car V4M

6.1 EXT_CSD incorrect setting case

If error is occurred when execute eMMC command, please check the following description and restart.

- Please Check the correct setting of EXT_CSD. If the wrong setting is present, to set the correct setting using EM_SECSD command.
- Program start address error of S-record file.

The following shows the setting of High speed SDR(50MHz) x8 bus width mode, Boot partition 1 enable.

Table 6-1 EXT_CSD setting list

Address	Register Name	Field name	Bit field	Settings
EXT_CSD[179]	PARTITION_CONFIG	BOOT_ACK	[6]	0x0
		BOOT_PARTITION_ENABLE	[5:3]	0x1
EXT_CSD[177]	BOOT_BUS_CONDITIONS	BOOT_MODE	[4:3]	0x1
		BOOT_BUS_WIDTH	[1:0]	0x2

For details of EXT_CSD, please refer to 1.3 References No.2.

6.2 Program start address error of S-record format file

After the message "Please Input User Program Start Address" has been displayed, input a start address of the S-record format file to be loaded (smallest value) as the start address of the program. (This address is treated as the start address and branch address of the data transfer destination from the external flash memory in the program.)
Please check the program start address and write again program using XLS2 and EM_W command.

6.3 Command for QSPI Flash and HyperFlash

If following log is output, please check board switch setting. About switch setting, refer to 5.1.

```
DEVICE ID Error. Please check switch setting
READ ID = 0x00000000
```

Figure 6-1 Device ID error

If following log is output, please check input value and image size.
Note) In case of XRS command, flash address must be 256 bytes boundary.
Note) In case of XLS2 and XLS3 command, flash address must be 4 bytes boundary.

```
===== Please Input Reading Top Address ============
  Please Input : H'07ffffff
Memory Boundary Error
```

Figure 6-2 Memory boundary error

If following log is output, please check input value.
User can't specify the value other than flash address range.

```
===== Please Input Reading Top Address ============
  Please Input : H'07fffffC
Address Error
Flash Address Range is 0x08000000--0x0BFFFFFF
```

Figure 6-3 Flash Address Range error

If following log is output, hardware defect is suspected.

```
ERASE QSPI-FLASH (220sec[typ])....Fail!
```

Figure 6-4 Erase error on XCS command

6.4 Command for eMMC

If following log is output, hardware defect is suspected.

```
eMMC Init ERROR!
```

Figure 6-5 eMMC initialization error

If following log is output, hardware defect is suspected.

```
EM_XXXXX(command name) CMD8 ERR!
```

Figure 6-6 CMD8 error

If following log is output, hardware defect is suspected.

```
EM_SECSD ERASE_GROUP_DEF CHG ERR!
```

Figure 6-7 ERASE_GROUP_DEF change error on EM_SECSD

If following log is output, hardware defect is suspected.

```
EM_SECSD ERR!
```

Figure 6-8 EM_SECSD error

If following log is output, hardware defect is suspected.

```
EM_W Partition select FAIL
```

Figure 6-9 Partition select fail

If following log is output, please check input value and image size.
User can't specify the value other than eMMC address range.

```
Please Input Start Address in sector :fbff
Please Input Program Start Address : eb210000
Work RAM(H'50000000-H'57FFFFFF) Clear....
please send ! (Motorola S-record)
SAVE -eMMC.......
Error occurred in emmc_exec_cmd func.:00000004
EM_W ERR
```

Figure 6-10 Address range error on eMMC

7. How to customize

R-Car S4 R-Car V4H R-Car V4M

This chapter explains the Flash writer customization procedure.

7.1 How to customize for another QSPI Flash

Flash writer supports QSPI Flash other than implemented on Spider board and White Hawk board.
About implemented QSPI Flash on Spider and White Hawk board, refer to Table 2-3.

If QSPI Flash on user board is different from on Spider board and White Hawk board, modify source code referring to following.
Target file is shown in Figure 7-1.

```
flash_writer
 |-- include
 |      \-- qspi_cmd.h
 |
 \-- qspi_cmd.c
```

Figure 7-1 File containing another vendor's QSPI Flash

1. Flash writer provides both structure of command table st_qspi_cmd_tbl_t and definition, in qspi_cmd.h. Modify USER_ADDED_QSPI from 0 to 1.
2. DEVID_XXXXXXXXX is defined in qspi_cmd.h. User needs to modify definition name and value according to user's QSPI Flash.
3. Command value for user's QSPI Flash is defined in qspi_cmd.c. e.g.) XXXXXXXXX_READ_FAST
   User needs to modify definition name and value according to user's QSPI Flash.
4. Modified defines are used in qspi_cmd.c, so user need to modify according to modified name.

The parts that the user need to modify are shown in blue.

```
#define USER_ADDED_QSPI          (0U)     /* 0:Disable 1:Enable */
...
#define DEVID_S25FS512S          (0x00200201U)       /* ID S25FS512S (03h:4Dh) */
#if USER_ADDED_QSPI == 1
#define DEVID_XXXXXXXXX          (0xFFFFFFFFU)
#endif /* USER_ADDED_QSPI == 1 */
```

Figure 7-2 Where the user changes in qspi_cmd.h

```
#if USER_ADDED_QSPI == 1
/* User can customize for another vendor's QSPI Flash. */
#define VENDOR_NUM                       (2U)
/* Command for XXXXXXXXX */
#define XXXXXXXXX_READ_FAST              (0x00U) /* read_fast */
```

```
#define XXXXXXXXX_SEC_ER_4BYTE_ADDR     (0x00U) /* sector_erase_4byte_addr */
#define XXXXXXXXX_PARA_4KBYTE_ER        (0x00U) /* parameter_4kbyte_erase */
#define XXXXXXXXX_PP_4BYTE_ADDR         (0x00U) /* pp_4byte_addr */
#define XXXXXXXXX_READ_ANY_REG          (0x00U) /* read_any_register */
#define XXXXXXXXX_WRITE_ANY_REG         (0x00U) /* write_any_register */
#define XXXXXXXXX_READ_STATUS           (0x00U) /* read_stts_register */
#define XXXXXXXXX_WRITE_ENABLE          (0x00U) /* write_enable */
#define XXXXXXXXX_BULK_ERASE            (0x00U) /* bulk_erase */
#endif /* USER_ADDED_QSPI == 1 */

...

#if USER_ADDED_QSPI == 1
    /* Command table for XXXXXXXXX */
    /* User can customize for another vendor's QSPI Flash. */
    {
        XXXXXXXXX_READ_FAST,            /* read_fast */
        XXXXXXXXX_SEC_ER_4BYTE_ADDR,    /* sector_erase_4byte_addr */
        XXXXXXXXX_PARA_4KBYTE_ER,       /* parameter_4kbyte_erase */
        XXXXXXXXX_PP_4BYTE_ADDR,        /* pp_4byte_addr */
        XXXXXXXXX_READ_ANY_REG,         /* read_any_register */
        XXXXXXXXX_WRITE_ANY_REG,        /* write_any_register */
        XXXXXXXXX_READ_STATUS,          /* read_status */
        XXXXXXXXX_WRITE_ENABLE,         /* write_enable */
        XXXXXXXXX_BULK_ERASE            /* bulk_erase */
    }
#endif /* USER_ADDED_QSPI == 1 */

...

#if USER_ADDED_QSPI == 1
    /* User can customize for another vendor's QSPI Flash. */
    {
        DEVID_XXXXXXXX
    }
#endif /* USER_ADDED_QSPI == 1 */
};
```

Figure 7-3 Where the user changes in qspi_cmd.c

Note) Source code of QSPI Flash driver in spiflash2drv.c must be modified according to user's QSPI Flash. Modification point is different as vendor of user's QSPI Flash.

REVISION HISTORY	R-Car Gen4 Flash writer sample software User's Manual: Software

Rev.	Date	Description	
		Page	Summary
0.2.0	Sep. 3, 2021	—	New creation.
0.2.1	Sep. 27, 2021	15-16	5. How to run Flash writer Changed Table5-1 Added Table5-4
0.3.0	Oct. 1, 2021	1	Table 1 1 Related document Added related document for eMMC.
		3	Table 2 2 External flash memory support status of R-Car S4 Table 2 3 Model number of supported external flash memory Modified to support eMMC.
		6	3.5 Command specification Modified note about which command is supported.
		8	3.6.4 eMMC dip switch setting list Added description.
		12-13	3.6.8 EM_DCID : Display the CID registers of eMMC Added chapter.
		13	3.6.9 EM_DCSD : Display the CSD registers of eMMC Added chapter.
		14	3.6.10 EM_DECSD : Display the EXT_CSD registers of eMMC Added chapter.
		14-15	3.6.11 EM_SECSD : Modify the EXT_CSD registers of eMMC Added chapter.
		15-16	3.6.12 EM_W : Write S-record format images to eMMC Added chapter.
		16-17	3.6.13 EM_WB : Write raw binary images to eMMC Added chapter.
		17-18	3.6.14 EM_E : Erase eMMC Added chapter.
		20	Table 5 3 Additional dip switch configuration for write to the eMMC on Spider Added table.
		22	5.1.2 In case of setting by CPLD tool Added chapter.
		23	Table 5 5 SW8 configuration for QSPI boot and eMMC boot on Spider for CPU board Added in case of eMMC boot.
		24-26	5.2.2 In case of setting by CPLD tool Added chapter.
		27	6.Error case to handle Added chapter.
0.4.0	Oct. 15, 2021	1	Table 1 1 Related document Added related document Number 3, 4, and 5.
		3	Table 2 2 External flash memory support status of R-Car S4 Table 2 3 Model number of supported external flash memory Modified to support HyperFlash.
		4	3.1 Function Added 'Display the command help'.
		4	Figure 3.1 Module structure Modified figure.
		6	Table 3-2 Binary size of Flash writer sample software Added table.
		7	3.5.1 Executable RAM area Added chapter.
		9	Table 3 5 QSPI Flash and HyperFlash dip switch setting list Table 3 6 Switch setting instruction list for QSPI Flash Removed tables.
		9	3.6.2 External flash memory dip switch setting list Removed this chapter.

Rev.	Date	Description	
		Page	Summary
		11	Table 3 6 Information of writing data for the QSPI Flash and HyperFlash boot for Spider Added Dummy TEE OS image. Modified Program Top Address of cert_header_sa9.srec.
		12	3.6.6 XLS3 : Write raw binary images to the QSPI Flash and HyperFlash memory Added chapter.
		13	3.6.7 XCS : Erase the QSPI Flash and HyperFlash Added description about HyperFlash.
		16	Table 3 7 Information of writing data for the eMMC boot for Spider Modified eMMC Save Sectors. Modified Program Top Address of cert_header_sa9.srec. Added Dummy TEE OS image.
		17	3.6.12 EM_W : Write S-record format images to eMMC 3.6.13 EM_WB : Write raw binary images to eMMC Added procedure about switch setting instruction.
		20	3.6.15 H : Display the command help Added chapter.
		22	5.1.1 In case of setting by CPU board Removed Table 5 2 and Table 5 3 and modified description.
		22	Table 5 1 MD setting for SCIF download mode Added table.
		25	5.2.1 In case of setting by CPU board Removed Table 5 5 and modified description.
		25	Table 5 3 MD setting for booting from external flash memory Added table.
0.5.0	Dec. 3, 2021	7	3.5.1 Executable RAM area Added SystemRAM to Table.
		11	3.6.4 XLS2 : Write S-record format images to the QSPI Flash and HyperFlash Modifid Program Top Address of dummy_rtos.srec from H'40100000 to H'E2100000. Added 'Table 3 7 Information of writing data for Flash and eMMC boot (if write ca55_loader.srec) for Spider'.
		18	3.6.11 EM_W : Write S-record format images to eMMC Modifid Program Top Address of dummy_rtos.srec from H'40100000 to H'E2100000. Added 'Table 3 9 Information of writing data for eMMC boot (if write ca55_loader.srec to Flash) for Spider'.
		19	3.6.11 EM_W : Write S-record format images to eMMC 3.6.12 EM_WB : Write raw binary images to eMMC Added Note about writable sector size to description.
0.6.0	Jan. 6, 2021	7	3.4.1 Release image Added chapter.
		11	3.6.4 XLS2 : Write S-record format images to the QSPI Flash and HyperFlash Removed following tables. Table 3-6 Information of writing data for QSPI Flash and HyperFlash only boot for Spider Table 3-7 Information of writing data for Flash and eMMC boot (if write ca55_loader.srec) for Spider
		18	3.6.11 EM_W : Write S-record format images to eMMC Removed following tables. Table 3-8 Information of writing data for the eMMC only boot for Spider Table 3-9 Information of writing data for eMMC boot (if write ca55_loader.srec to Flash) for Spider
0.7.0	Feb. 2, 2022	4	Figure 3.1 Module structure Modified ICUMXA to ICUMX.
		6	Figure 3.2 Detail of the SCIF download image Modified ICUMXA to ICUMX.
		7	Table 3-3 Information list of release image (QSPI Flash and HyperFlash) Modified ICUMXA to ICUMX.
		7	Table 3-4 Information list of release image (eMMC) Modified ICUMXA to ICUMX.
		8	Table 3-5 Information list of release image when ICUMX IPL loads CX 2nd IPL

Rev.	Date	Description	
		Page	Summary
			Modified ICUMXA to ICUMX.
		10	Figure 3.3 Log output of common procedure Modified Flash writer version to X.X.X.
		23	4.4 Build the Flash writer binary Modified ICUMXA to ICUMX and ‘Output the following image’.
		25	Figure 5.2 log output of Flash writer Modified Flash writer version to X.X.X.
		26	5.1.2 In case of setting by CPLD tool Modified ICUMXA to ICUMX and ‘Transfer S-record file’.
0.8.0	Mar. 3, 2022	—	Added labels to all chapters.
		1	1.1 Overview Added description about label R-Car S4 R-Car V4H. Added ‘1.2 Supported device’.
		2	Table 1 2 Related document Added Number 6 and 7.
		3	Table 2 1 Hardware Environment Added ‘R-Car V4H System Evaluation Board White Hawk’.
		4	Table 2 2 External flash memory support status Table 2 3 Model number of supported external flash memory Added R-Car V4H Ver.1.0 to table.
		5	Table 2 4 Hardware resource Added ‘LPDDR5-SDRAM’ and ‘RT-VRAM0’.
		7	Figure 3 1 Module structure Update figure according to source code.
		8	3.3.1 LSI Modified build option name from ‘BOARD’ to ‘LSI’.
		9	Figure 3 2 Detail of the SCIF download image Update figure. Table 3 2 Binary size of Flash writer sample software Added the size of Flash writer for V4H.
		10	Table 3 3 Information list of release images for S4 (QSPI Flash / HyperFlash) Updated Flash Save Address. Table 3 4 Information list of release image for V4H (QSPI Flash) Added table.
		11	Table 3 5 Information list of release images for S4 (eMMC) Updated eMMC Save Sectors.
		13	Table 3 8 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (QSPI Flash) Table 3 9 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (eMMC) Added table.
		15	Table 3 12 Executable RAM area for V4H Added table.
		16	3.6.1.2 Step2 Setting the terminal software Added setting value for V4H.
		17	Table 3 14 Switch setting instruction list for external flash memory (V4H) Added table.
		18	3.6.3 XRS : Read QSPI Flash and HyperFlash memory data Added following note. Note) The size of enable to read at once is up to 16Kbytes.
		19	3.6.4 XLS2 : Write S-record format images to the QSPI Flash and HyperFlash Removed following note. Note) Writable S-record format image size to QSPI Flash and HyperFlash is up to 4MBytes. Added description about reference table for V4H.
		20	3.6.5 XLS3 : Write raw binary images to the QSPI Flash and HyperFlash memory Removed following note. Note) Writable raw binary image size to QSPI Flash and HyperFlash is up to 4Mbytes. Added description about reference table for V4H.

Rev.	Date	Description	
		Page	Summary
		25	3.6.11 EM_W : Write S-record format images to eMMC Removed following note. Note) Writable S-record format image size to eMMC is up to 4Mbytes. Added description about reference table for V4H.
		26	3.6.12 EM_WB : Write raw binary images to eMMC Removed following note. Note) Writable raw binary image size to eMMC is up to 4Mbytes. Added description about reference table for V4H.
		29	Figure 4 2 example of building Flash writer for S4 Added 'LSI=S4'. 4.4 Build the Flash writer binary Updated output file name.
		30	Figure 4 3 example of building Flash writer for V4H Added figure.
		31	Table 5 3 MD setting for SCIF download mode for V4H Added table.
		31-32	5.1.1 In case of setting by CPU board (SW BOARD M) Added description about V4H.
		32	Table 5 4 Terminal software configuration for V4H when SCIF download mode Added table.
		33	5.1.2 In case of setting by CPLD tool (S4) Modified Flash writer's S-record file name.
		34	Table 5 6 Terminal software configuration for S4 when boot from the external flash memory Added HSCIF setting.
		35	Table 5 7 MD setting for booting from external flash memory (V4H) Table 5 8 Terminal software configuration for V4H when boot from the external flash memory Added table. Added description for V4H.
		36-38	5.2.2 In case of setting by CPLD tool (S4) Added Note *1.
0.8.1	Mar. 17, 2022	34	Table 5 5 MD setting for booting from external flash memory (S4) Fixed MD setting.
		35	Table 5 7 MD setting for booting from external flash memory (V4H) Table 5 8 Terminal software configuration for V4H when boot from the external flash memory Fixed MD setting.
0.9.0	May 23, 2022	1	Table 1-1 Supported device Modified from T.B.D. to Support in HyperFlash of R-Car V4H. Removed Note.
		4	Table 2 2 External flash memory support status Added R-Car S4 Ver.1.1 to SoC. Modified from T.B.D. to Support in R-Car V4H Ver.1.0 of HyperFlash.
		4	Table 2 3 Model number of supported external flash memory Added R-Car S4 Ver.1.1 to SoC. Modified from T.B.D. to S26KS512SDPBHV02 in R-Car V4H Ver.1.0 of HyperFlash.
		6	Table 2 5 Software environment list Added Tera Term.
		7	3.1 Function Added 'Read the eMMC memory data'.
		8	Figure 3 1 Module structure Updated according to latest source code.
		10	Figure 3 2 Detail of the SCIF download image Added SDRAM.
		10	Table 3 2 Binary size of Flash writer sample software Updated binary size.
		11-14	Table 3 3 Information list of release images for S4 (QSPI Flash / HyperFlash) Table 3 4 Information list of release images for V4H (QSPI Flash)

Rev.	Date	Description	
		Page	Summary
			Table 3 5 Information list of release images for S4 (eMMC) Table 3 6 Information list of release images for S4 when ICUMX IPL loads CX 2nd IPL(QSPI Flash / HyperFlash) Table 3 8 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (QSPI Flash) Modified Program Top Address of cert_header_sa9.srec from H'EB206000 to H'EB230000.
		14	Table 3 8 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (QSPI Flash) Modified Filename from cx_loader.srec to cr52_loader.srec.
		14	Table 3 9 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (eMMC) Removed AArch64_Dummy_CA76_Program3.srec (Dummy object).
		15	3.4.2 Data format of Motorola S-record file Added chapter.
		16	Table 3 10 Command list Added EM_R.
		17	3.5.1 Executable RAM area Removed chapter.
		26	3.6.11 EM_R : Read QSPI Flash and HyperFlash memory data Added chapter.
		33-34	Table 5 2 Terminal software configuration for S4 when SCIF download mode Table 5 4 Terminal software configuration for V4H when SCIF download mode Added Line feed code (Transmit).
		43-44	6.3 Command for QSPI Flash and HyperFlash 6.4 Command for eMMC Added chapter.
		45-46	7. How to customize Added chapter.
0.9.1	Jul. 15, 2022	14	Table 3 9 Information list of release images for V4H when ICUMX IPL loads CX 2nd IPL (eMMC) Changed eMMC Save Sctors.
0.10.0	Oct. 7, 2022	-	Changed User's Manual and ReleaseNote document number.
		9	3.3.1 LSI Table 3 1 Build option LSI setting description Added S4N.
		11	3.4.1 Release image Table 3 4 Information list of release images for V4H (QSPI Flash) Removed table.
0.13.0	Jan. 12, 2023	9	3.3.1 LSI Table 3-1 Build option LSI setting description Removed 'S4N' row. Added 'S4N' to 'S4' row.
0.13.1	Aug. 2, 2023	19-20	3.6.4 XLS2 : Write S-record format images to the QSPI Flash and HyperFlash 3.6.5 XLS3 : Write raw binary images to the QSPI Flash and HyperFlash Fixed a link to table 3-7.
1.25.0	Aug.24.2023	1-30,32,33,36,41,44	Added description about label R-Car V4M
		4	2.1 Hardware Environment Add table
		8	3.2 Module structure Add QOS initialize file. Add memory def.
		9	3.3.1 LSI Add build option
		33	5.1.1 In case of setting by CPU board (SW BOARD M) Add output file
1.30.0	Sep.20.2023	4,10	Table 2 2 External flash memory support status

Rev.	Date	Description	
		Page	Summary
			Table 2 3 Model number of supported external flash memory Table 3 2 Binary size of Flash writer sample software Add line for V4M
		18	Table 3 11 Switch setting instruction list for external flash memory Add for V4M

CONFIDENTIAL

R-Car Gen4 Flash writer sample software
User's Manual: Software

Publication Date: Rev.0.2.0 Sep. 3, 2021
Rev.1.30.0 Sep. 20, 2023

Published by: Renesas Electronics Corporation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

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

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

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

R-Car Gen4 Flash writer sample software User's Manual

Renesas Electronics Corporation