User's Manual

RENESAS

CONFIDENTIAL

Linux Interface Specification Yocto recipe Start-Up Guide

User's Manual: Software

R-Car V3U/V3H/V3M Series

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

Renesas Electronics
www.renesas.com

Rev.4.19.2 Feb. 2022

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

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

How to Use This Manual

- **[Readers]**
  This manual is intended for engineers who develop products which use the R-Car V3U/V3H/V3M processor.

- **[Purpose]**
  This manual is intended to give users an understanding of the functions of the R-Car V3U/V3H/V3M processor device driver and to serve as a reference for developing hardware and software for systems that use this driver.

- **[How to Read This Manual]**
  It is assumed that the readers of this manual have general knowledge in the fields of electrical
  — Engineering, logic circuits, microcontrollers, and Linux.
  → Read this manual in the order of the CONTENTS.
  — To understand the functions of a multimedia processor for R-Car V3U/V3H/V3M
  → See the R-Car V3U/V3H/V3M User’s Manual.
  — To know the electrical specifications of the multimedia processor for R-Car V3U/V3H/V3M
  → See the R-Car V3U/V3H/V3M Data Sheet.

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

Table of Contents

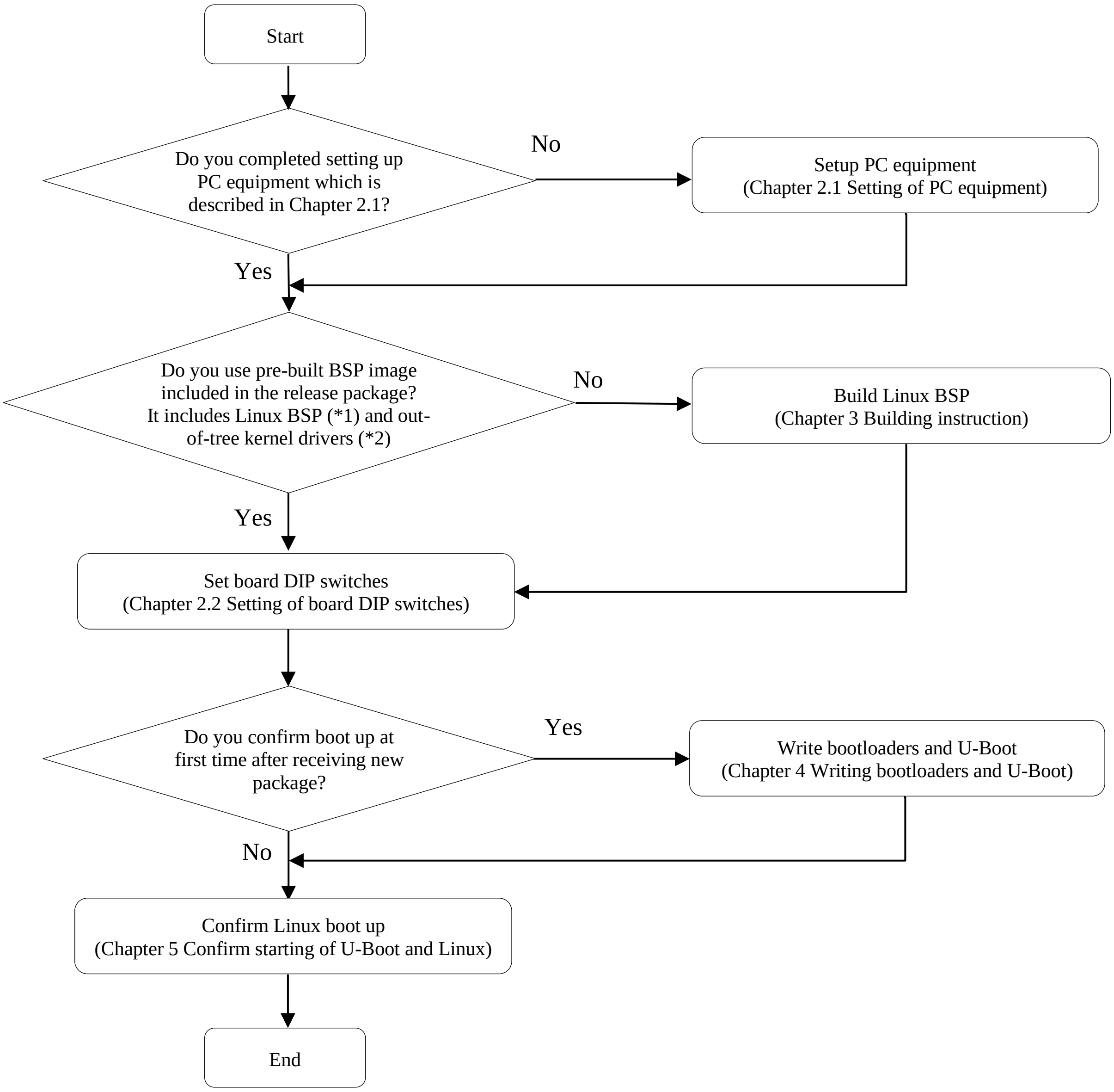
Introduction

This start-up guide explains R-Car V3U/V3H/V3M Yocto recipe package files, the system environments, the make method of kernel, the operating of U-Boot and so on.

This product R-Car V3U/V3H/V3M Yocto recipe is a basic package to operate built-in Linux and basic middleware on the R-Car V3U/V3H/V3M System Evaluation Board. Please contact Renesas Electronics person who provided this product to you in case of questions.

The meaning of red characters is update part from the previous document.

A flowchart for using this Start Up Guide.


Start
Do you completed setting up PC equipment which is described in Chapter 2.1?
No
Setup PC equipment
(Chapter 2.1 Setting of PC equipment)
Yes
Do you use pre-built BSP image included in the release package? It includes Linux BSP (*1) and out-of-tree kernel drivers (*2)
No
Build Linux BSP
(Chapter 3 Building instruction)
Yes
Set board DIP switches
(Chapter 2.2 Setting of board DIP switches)
Do you confirm boot up at first time after receiving new package?
Yes
Write bootloaders and U-Boot
(Chapter 4 Writing bootloaders and U-Boot)
No
Confirm Linux boot up
(Chapter 5 Confirm starting of U-Boot and Linux)
End

(*1) Linux BSP includes the following drivers.

Display
DMA Engine
GPIO
I2C
IPMMU
Kernel Core
MSIOF
Power Management
PWM
RWDT
SCIF/HSCIF
SD/MMC
U-Boot
Video Capture
PCIEC
CAN-FD
CMEM
CMT/TMU
Memory Manager
RPC
UIO
Ethernet-AVB

(*2) Out-of-tree Kernel drivers: Refer to Appendix: Out-of-tree driver information

1. R-Car V3U/V3H/V3M Linux BSP package files

This Yocto recipe will be taken.

The U-Boot source code from:

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

R-Car V3U/V3H/V3M Linux source code from:

https://github.com/renesas-rcar/linux-bsp.git

1.1 Reference (R-Car V3U/V3H/V3M)

Document	Document name	Version	Date
R-CarV3U Series User's Manual	[R19UH0144EJ0056] R-Car V3U series User's Manual with Image Recognition chapter.pdf	Rev.0.56	Feb. 05, 2021
R-CarV3U System Evaluation Board FALCON Hardware Manual RTP0RC779A0CPB0010S	e_Falcon_hwman_rev003_1204.pdf	Rev.0.03	Dec. 04, 2020
R-CarV3U System Evaluation Board FALCON Setup Manual	e_Falcon_suman_rev007_1204.pdf	Rev.0.07	Dec. 04, 2020
R-Car Series, 3rd Generation User's Manual: Hardware	r19uh0105ej0220-r-car-3rd-generation.pdf	Rev.2.20	Jun. 30, 2020
R-CarV3H System Evaluation Board Condor-I Setup Manual	e_Condor-I_suman_rev001_0711.pdf	Rev.0.01	Jul. 11, 2019
R-CarV3H System Evaluation Board Condor Setup Manual	ASOS-C18-009_Condor_suman_rev100_e.pdf	Rev.1.00	Feb. 28, 2018
R-CarV3M System Evaluation Board Eagle Setup Manual	e_Eagle_suman_rev003_0608.pdf	Rev.003	Jun. 08, 2017
R-Car V3H Starter Kit HW Manual	R20UT4302ED0100.pdf	Rev.3.00	Feb. 19, 2021
R-Car V3M Starter Kit HW Manual	REN_r12ut0003ed0230-rcarv3m_MAT_20190124.pdf	Rev.2.30	Jan. 24, 2019

2. Environmental Requirement

2.1 Setting of PC equipments

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

Table 1 R-Car V3U/V3H/V3M Linux BSP Environmental Requirement

Equipment	Software requirement and explanation	
Linux Host PC	Ubuntu 20.04 LTS (64bit) is recommended as OS. 32bit version is not supported. It is used as compiling and debugging environment. Please refer the following URL https://ubuntu.com/tutorials/install-ubuntu-desktop It can also be used as TFTP server and NFS server. **For "How to set IP address to Linux Host PC", please refer to Appendix**	
	TFTP server software	It is used when Hyper Flash is written by U-Boot or Image is downloaded. For installing TFTP server, please refer to Appendix
	NFS server software	It is used when File system is mounted by NFS. For installing NFS server, please refer to Appendix
Windows PC for terminal software	Windows 10 is recommended as OS. It is used as debugging environment. Terminal software and VCP driver are executed.	
	Terminal software	Please use following software. 1) Tera Term (Confirmed with Japanese version of Tera Term 4.88 Available at http://sourceforge.jp/projects/ttssh2)
	VCP driver	Please install in Windows Host PC. The host PC that connects to the Falcon board requires a USB driver for FT2232H. It can be obtained from the following URL. https://www.ftdichip.com/Drivers/VCP.htm The host PC that connects to the Condor/Condor-I board requires a USB driver for the CP2102N. It can be obtained from the following URL. https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers The host PC that connects to Eagle board requires a USB driver for the CP210.2 It can be obtained from the following URL. http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx The host PC that connects to V3H Starter Kit requires FTDI driver, it can be obtained from the following URL. http://www.ftdichip.com/Drivers/D2XX.htm USB become virtual COM port on terminal software. Please connect to: • CN10 on V3U System Evaluation Board. • CN11 on V3H/V3M System Evaluation Board. • CN6 on V3H Starter Kit.

Please connect R-Car V3U/V3H/V3M System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC with TFTP and NFS server as Figure 1 below.

In case of R-Car V3U, Ethernet connector is CN3 on breakout board, serial connector is CN10 on CPU board.

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

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

Recommended Environment

The following shows a Recommended Environment.

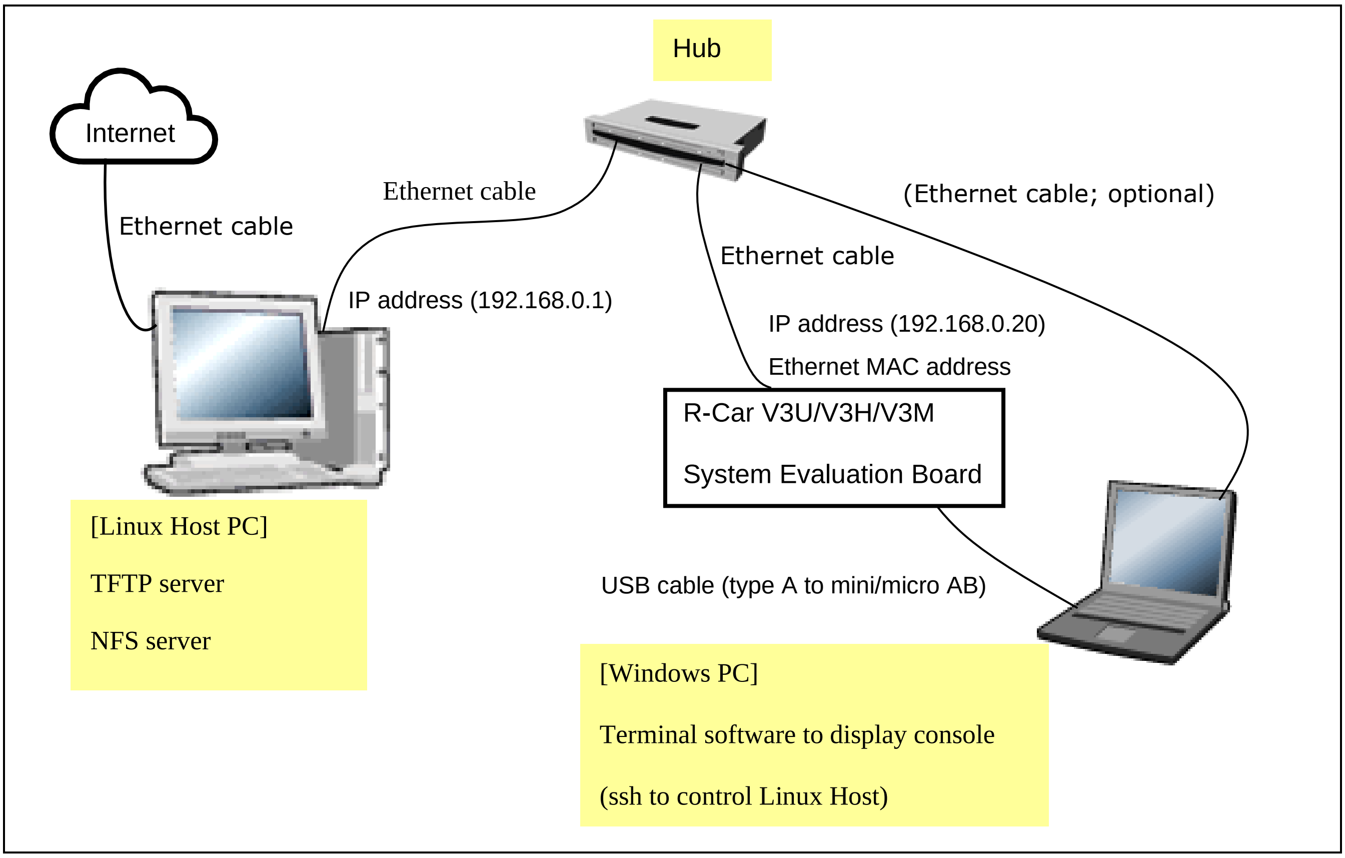


Figure 1. Recommended Environment for R-Car V3U/V3H/V3M Linux BSP

The LAN port on the Linux Host PC should be separated from that connecting to the Internet and that connecting to the system evaluation board. For example, prepare another LAN board or USB-LAN and add LAN port connecting to the system evaluation board.
The LAN port should be able to set a static IP address.

IP address (192.168.0.1):
This is used for IP address of TFTP server and NFS server on Linux Host PC.
U-boot environment variables name is "serverip".

IP address (192.168.0.20):
This is used for IP address of System Evaluation Board.
U-boot environment variables name is "ipaddr".

Ethernet MAC address:
This is used for Ethernet MAC address of the network device on System Evaluation Board.
U-boot environment variables name is "ethaddr".

USB cable:
Connect USB Host connector of Windows PC that is virtual COM port to CN10 of Falcon, CN11 of Condor, Condor-I or Eagle with USB cable for displaying console.

2.2 Setting of board dip switches

The setting of R-Car V3U/V3H/V3M System Evaluation Board's dip switches is shown in the following Table 2, Table 3, Table 4, Table 5.

Note) Please refer each evaluation board 's Setup manual in detail and set the dip switches according to your use case.

Note) We describe **bold character** about Dip switches which are different from the default settings of System Evaluation Board Setup Manual.

For Falcon board, please refer to “R-CarV3U System Evaluation Board Falcon Setup Manual (FALCON_suman_revx.pdf)” and find the picture below.

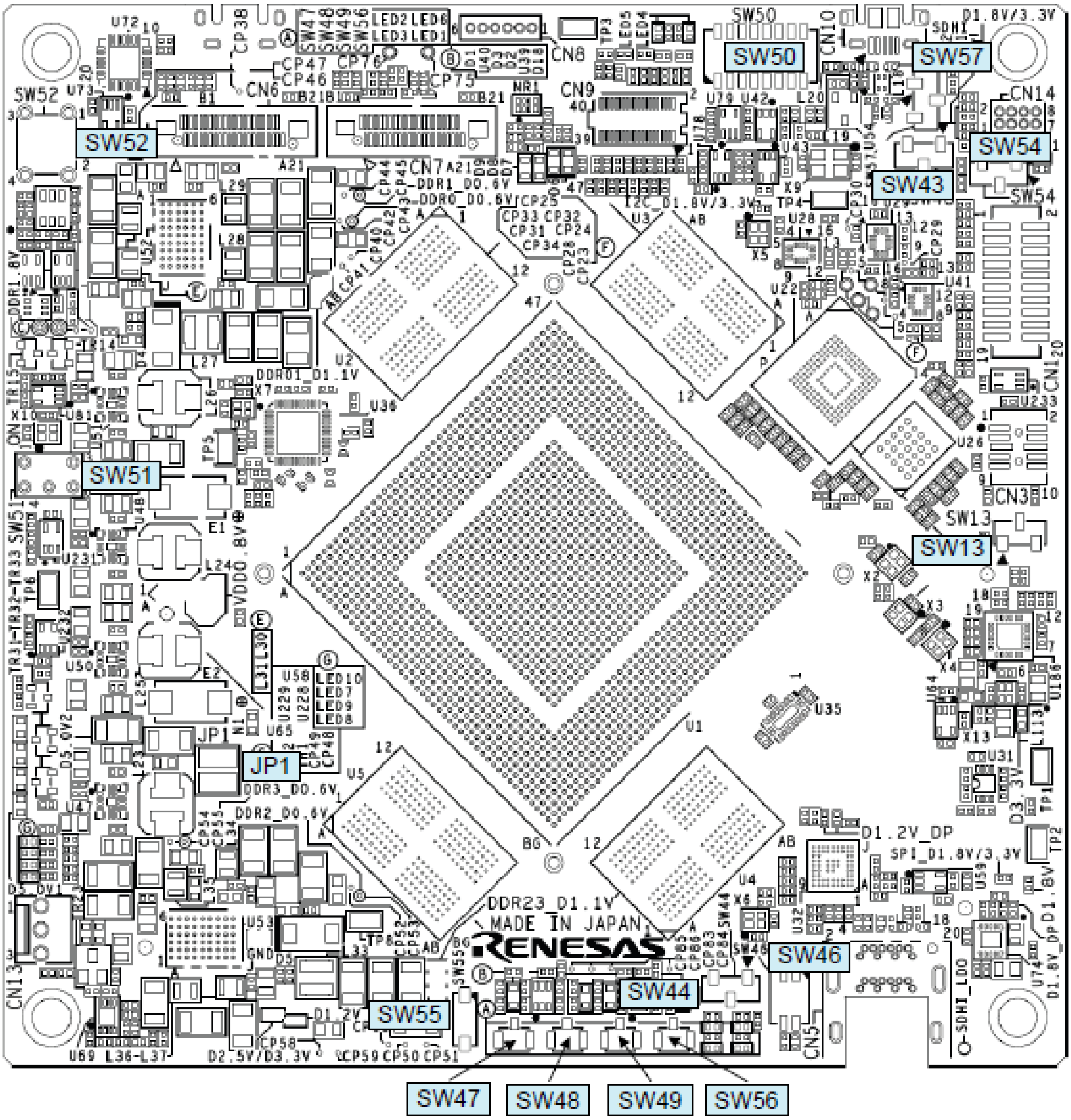


Figure 1.2.1 Location of Switches on the Falcon CPU Board (Component Side)

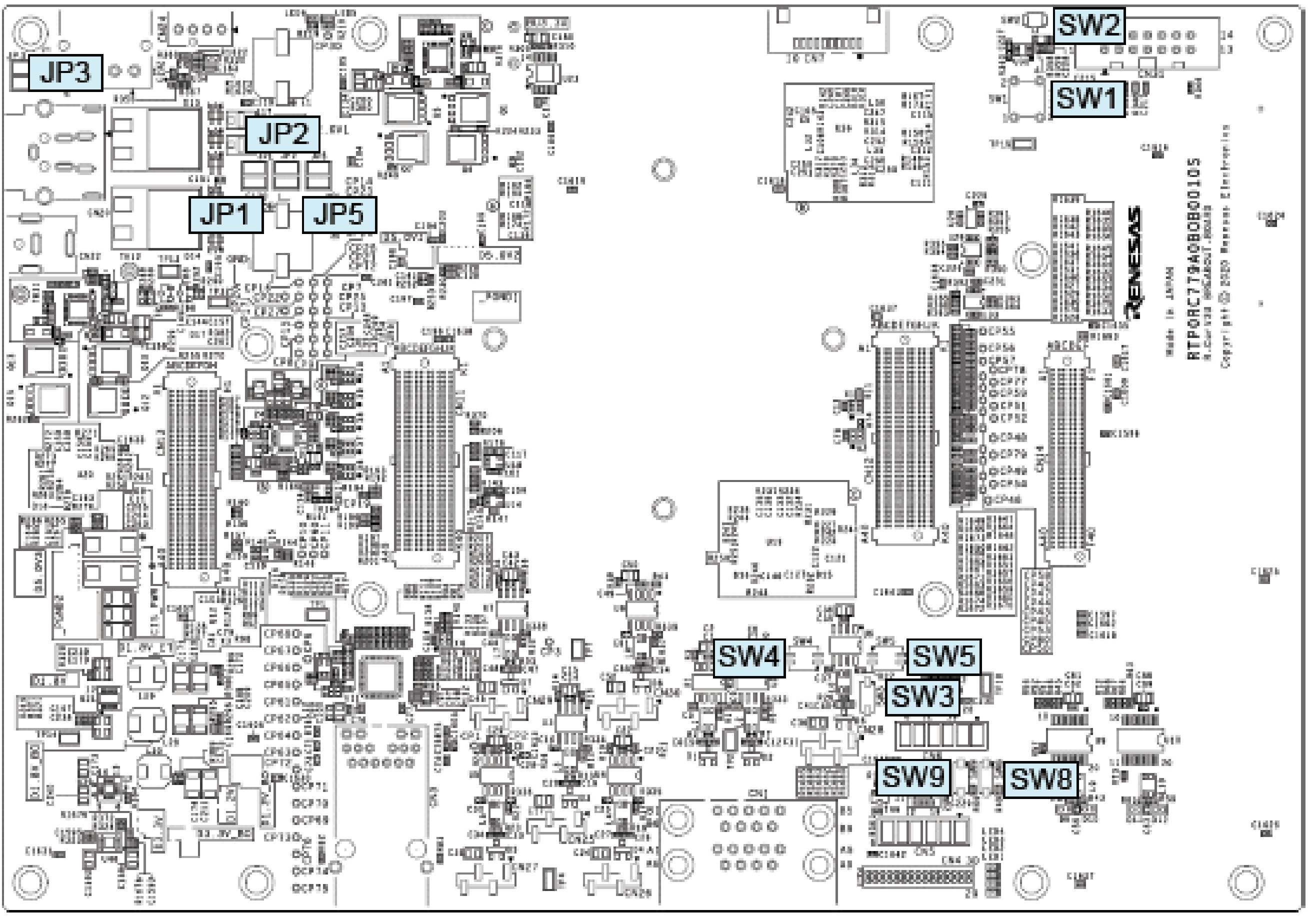


Figure 1.4.1 Location of Switches on the Breakout Board (Component Side)

Table 2 setting of Dip switches on Falcon board (R-Car V3U Ver.1.0)

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
Falcon CPU board										
SW50*1	**FPGA**	**C**	**ON**	**ON**	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**OFF**
SW57	**Hyper/SPI**	**C**			✓	-	-	-	-	-
SW13	**SPI/EX-SPI**	**C**	✓	-		-	-	-	-	-
SW43	MMC/SD	C		✓		-	-	-	-	-
SW44	DSI	C		✓		-	-	-	-	-
SW46	SOFTSW	C	OFF	OFF	-	-	-	-	-	-
SW47	PUSHSW0	C	Push SW	-	-	-	-	-	-	-
SW48	PUSHSW1	C	Push SW	-	-	-	-	-	-	-
SW49	PUSHSW2	C	Push SW	-	-	-	-	-	-	-
SW51	ACCSW	C		-	✓ (OFF)	-	-	-	-	-
SW52	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW54	VDDQ_SDHI	C		✓		-	-	-	-	-
(SW55)	MXBKUP	C	✓ (OFF)		-	-	-	-	-	-
SW56	RL78_CRST#	C	Push SW	-	-	-	-	-	-	-
(JP1)	PWSTG1	C	Open	-	-	-	-	-	-	-
Breakout board										
SW1	RL78_RST#	C	Push SW	-	-	-	-	-	-	-
SW2	RL78_PUSHSW	C	Push SW	-	-	-	-	-	-	-
SW3	CAN_SILENT	C		✓ (ON)	-	-	-	-	-	-
SW4	CAN5/FlexRay_A	C	ON	OFF	-	-	-	-	-	-
SW5	CAN5/FlexRay_B	C	ON	OFF	-	-	-	-	-	-
SW8	EN_FlexRay_A	C	✓ (OFF)		-	-	-	-	-	-
SW9	EN_FlexRay_B	C	✓ (OFF)		-	-	-	-	-	-
JP1	D12.0V2_A	C	Short	-	-	-	-	-	-	-
JP2	D12.0V2_B	C	Short	-	-	-	-	-	-	-
JP5	D12.0V2_C	C	Short	-	-	-	-	-	-	-
JP3	EX_PWRON	C	Open	-	-	-	-	-	-	-
JP4	D5.0V2	C	Open	-	-	-	-	-	-	-

C: Component side of the board, S: Solder side of the board.

*1: Booting is from the SPI flash memory (U6) on the Falcon CPU board or the SPI flash memory connected to the EX-SPI connector (CN3).

For Condor-I board, please refer to “R-CarV3H System Evaluation Board Condor-I Setup Manual (Condor-I_suman_revx.pdf)” and find the picture below.

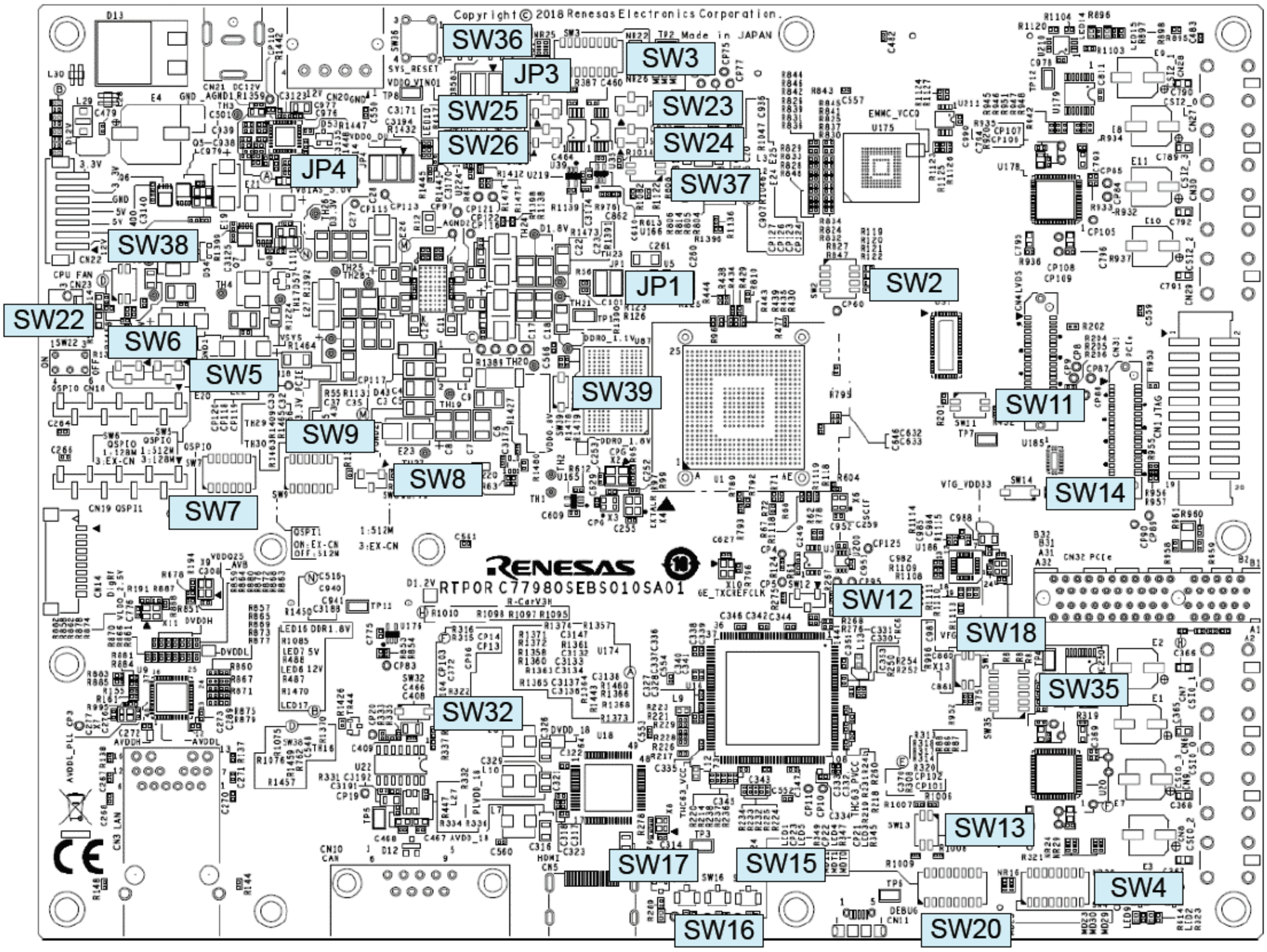


Figure 1.2.1 Location of Switches on the Condor-I Board (Component Side)

Table 3 setting of Dip switches on Condor-I board (R-Car V3H Ver.2.0)

Switch Number	Switch Name	Slide (C/S)	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW3	**MODESW-A**	**C**	**OFF**	**ON*1**	**ON**	**OFF**	**ON**	**OFF**	**ON**	**ON**
SW20	MODESW-B	C	OFF	ON	ON	ON	ON	ON	ON	ON
SW4	MODESW-C	C	OFF	ON	ON	ON	ON	ON	OFF	ON
SW13	MODESW-D	C	OFF	OFF	-	-	-	-	-	-
SW37	DBGMODE	C	OFF	OFF	-	-	-	-	-	-
SW2	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW5*2	**QSPI0-A**	**C**	✓	-		-	-	-	-	-
SW6*2	**QSPI0-B**	**C**	✓	-		-	-	-	-	-
SW7*2	**QSPI0-C**	**C**	**ALL OFF**						-	-	-
SW8	QSPI1-A	C		-	✓	-	-	-	-	-
SW9	QSPI1-B	C	ALL ON						-	-	-
SW1	DDRBKUP	C	OFF	OFF	-	-	-	-	-	-
SW11	LVDS-A	C	ON	ON	-	-	-	-	-	-
SW12	LVDS-B	C			✓	-	-	-	-	-
SW18	MIPI-SW	C	ON	ON	-	-	-	-	-	-
SW35	MAX9286	C	ON	ON	ON	OFF	ON	ON	-	-
SW32	GPIO/CAN	C		✓ (ON)	-	-	-	-	-	-
SW14	PCIe/HSSTP	C		✓ (ON)	-	-	-	-	-	-
SW23	MSIOF-A	C	✓	-		-	-	-	-	-
SW24	MSIOF-B	C	✓	-		-	-	-	-	-
SW25	MSIOF-C	C	✓	-		-	-	-	-	-
SW26	MSIOF-D	C	✓	-		-	-	-	-	-
SW36	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW15	PUSHSW0	C	Push SW	-	-	-	-	-	-	-
SW16	PUSHSW1	C	Push SW	-	-	-	-	-	-	-
SW17	PUSHSW2	C	Push SW	-	-	-	-	-	-	-
SW22	ACCSW	C		-	✓ (OFF)	-	-	-	-	-
SW38	TRG_PWR	C	ON	OFF	-	-	-	-	-	-
SW39	M0BKUP	C		✓ (ON)	-	-	-	-	-	-
JP1	V_TEST	C	Open	-	-	-	-	-	-	-
(JP3)	VDDQ_VIN01	C		-	-	-	-	-	-	-
(JP4)	VDDQ_DU	C	-	-	-	-	-	-	-	-

C: Component side of the board, S: Solder side of the board

*1: SW3 Pin2 ON to boot through ICUMXA, if boot through Cortex-R7 please set SW3 Pin2 OFF.

*2: 512-Mbit SPI flash memory (U6).

For Condor board, please refer to “R-CarV3H System Evaluation Board Condor Setup Manual (ASOS-C18-009_Condor_suman_revx.pdf)” and find the picture below.

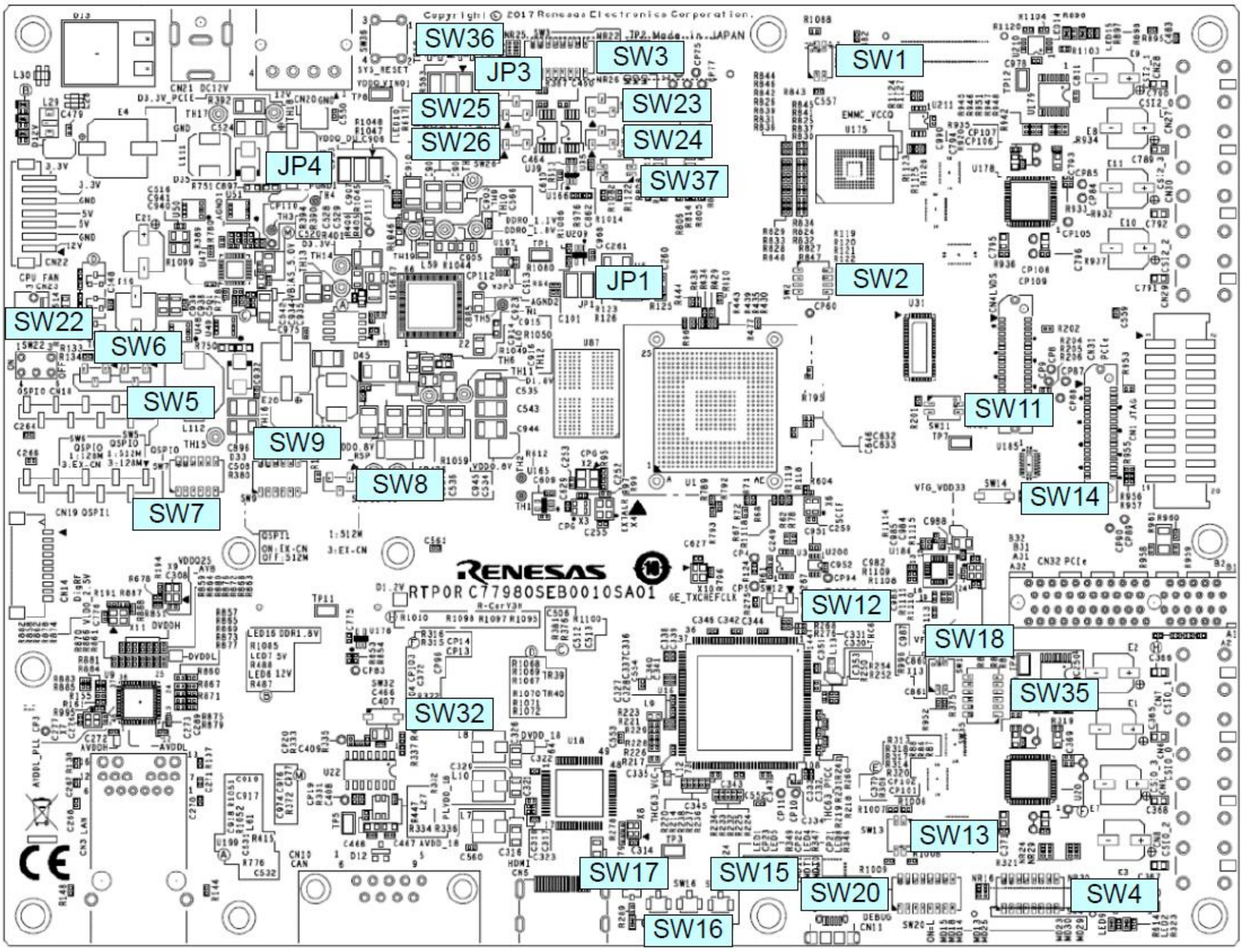


Figure 1.2.1 Location of Switches on the Condor Board (Component Side)

Table 4 setting of Dip switches on Condor board (R-Car V3H Ver.1.1)

Switch Number	Switch Name	Slide (C/S)	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW3	**MODESW-A**	**C**	**OFF**	**ON*1**	**ON**	**OFF**	**ON**	**OFF**	**ON**	**ON**
SW20	MODESW-B	C	OFF	ON	ON	ON	ON	ON	ON	ON
SW4	MODESW-C	C	OFF	ON	ON	ON	ON	ON	OFF	ON
SW13	MODESW-D	C	OFF	OFF	-	-	-	-	-	-
SW37	DBGMODE	C	OFF	OFF	-	-	-	-	-	-
SW2	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW5*2	**QSPI0-A**	**C**	✓	-		-	-	-	-	-
SW6*2	**QSPI0-B**	**C**	✓	-		-	-	-	-	-
SW7*2	**QSPI0-C**	**C**	**ALL OFF**					-	-	-
SW8	QSPI1-A	C		-	✓	-	-	-	-	-
SW9	QSPI1-B	C	ALL ON					-	-	-
SW1	DDRBKUP	C	OFF	OFF	-	-	-	-	-	-
SW11	LVDS-A	C	ON	ON	-	-	-	-	-	-
SW12	LVDS-B	C			✓	-	-	-	-	-
SW18	MIPI-SW	C	ON	ON	-	-	-	-	-	-
SW35	MAX9286	C	ON	ON	ON	OFF	ON	ON	-	-
SW32	GPIO/CAN	C		✓ (ON)	-	-	-	-	-	-
SW14	PCIe/HSSTP	C		✓ (ON)	-	-	-	-	-	-
SW23	MSIOF-A	C	✓	-		-	-	-	-	-
SW24	MSIOF-B	C	✓	-		-	-	-	-	-
SW25	MSIOF-C	C	✓	-		-	-	-	-	-
SW26	MSIOF-D	C	✓	-		-	-	-	-	-
SW36	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW15	PUSHSW0	C	Tactile SW	-	-	-	-	-	-	-
SW16	PUSHSW1	C	Tactile SW	-	-	-	-	-	-	-
SW17	PUSHSW2	C	Tactile SW	-	-	-	-	-	-	-
SW22	ACCSW	C		-	✓ (OFF)	-	-	-	-	-
JP1	V_TEST	C	Open	-	-	-	-	-	-	-
(JP3)	VDDQ_VIN01	C		-	-	-	-	-	-	-
(JP4)	VDDQ_DU	C	-	-	-	-	-	-	-	-

C: Component side of the board, S: Solder side of the board

*1: SW3 Pin2 ON to boot through ICUMXA, if boot through Cortex-R7 please set SW3 Pin2 OFF.

*2: 512-Mbit SPI flash memory (U6).

For Eagle board, please refer to “R-CarV3M System Evaluation Board Eagle Setup Manual (Eagle_suman_revx.pdf)” and find the picture below.

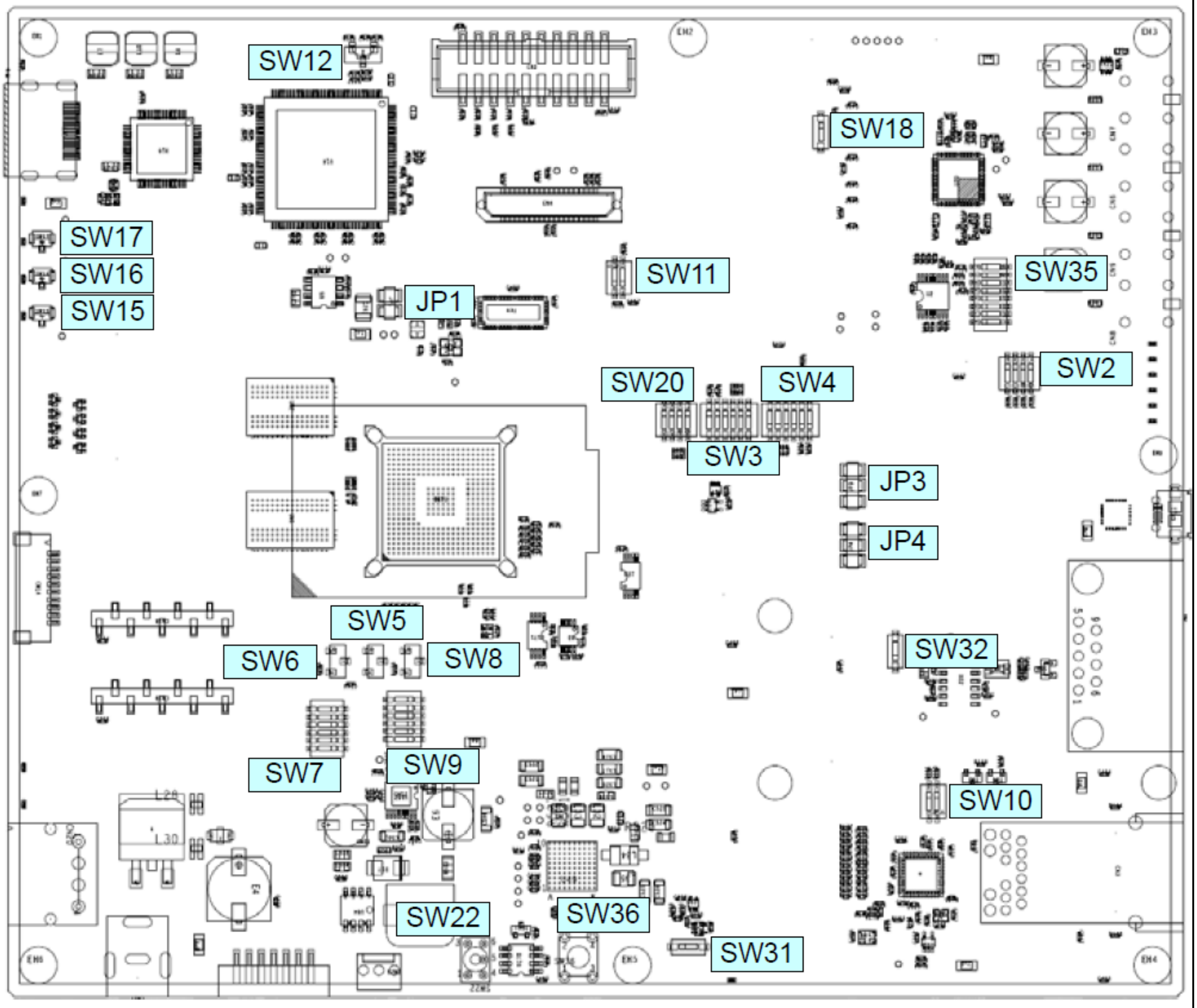


Figure 1.2.1 Location of Switches on the Eagle Board (Component Side)

Table 5 setting of Dip switches on Eagle board (R-Car V3M Ver.2.0)

Switch Number	Switch Name	Slide (C/S)	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW2	SOFTSW	C	OFF	OFF	OFF	OFF	-	-	-	-
SW3	**MODESW-A**	**C**	**OFF*1**	**ON**	**ON**	**OFF**	**ON**	**ON**	-	-
SW20	MODESW-B	C	OFF	ON	ON	ON	-	-	-	-
SW4	MODESW-C	C	OFF	ON	ON	ON	ON	ON	-	-
SW5*2	**QSPI0-A**	**C**	✓	-		-	-	-	-	-
SW6*2	**QSPI0-B**	**C**	✓	-		-	-	-	-	-
SW7*2	**QSPI0-C**	**C**	**ALL OFF**					-	-	-
SW8	QSPI1-A	C		-	✓	-	-	-	-	-
SW9	QSPI1-B	C	ALL ON					-	-	-
SW10	PHYAD	C	OFF	OFF	-	-	-	-	-	-
SW11	LVDS-A	C	ON	ON	-	-	-	-	-	-
SW12	LVDS-B	C			✓	-	-	-	-	-
SW15	TACTSW0	C	Tactile SW	-	-	-	-	-	-	-
SW16	TACTSW1	C	Tactile SW	-	-	-	-	-	-	-
SW17	TACTSW2	C	Tactile SW	-	-	-	-	-	-	-
SW18	MIPI-SW	C		✓ (ON)	-	-	-	-	-	-
SW22	ACCSW	C		-	✓ (OFF)	-	-	-	-	-
SW36	PRESET#	C	Push SW	-	-	-	-	-	-	-
SW31	DDRBKUP	C	✓ (OFF)		-	-	-	-	-	-
SW32	GPIO/CAN	C	✓ (OFF)		-	-	-	-	-	-
SW35	MAX9286	C	ON	ON	ON	OFF	ON	ON	OFF	ON
JP1	V_TEST	C	Open	-	-	-	-	-	-	-
(JP3)	VDDQ_VIN01	C		-		-	-	-	-	-
(JP40	VDDQ_DU	C		-		-	-	-	-	-

C: Component side of the board, S: Solder side of the board

*1: Booted through Cortex-R7.

*2: 512-Mbit SPI flash memory (U6). For V3H Starter Kit, please refer to “R-Car V3H Starter Kit HW Manual (R20UT4302ED0100.pdf)”.

For V3H Starter Kit, please refer to "R-Car V3H Starter Kit HW Manual (R20UT4302ED0100.pdf)" and find the below picture.

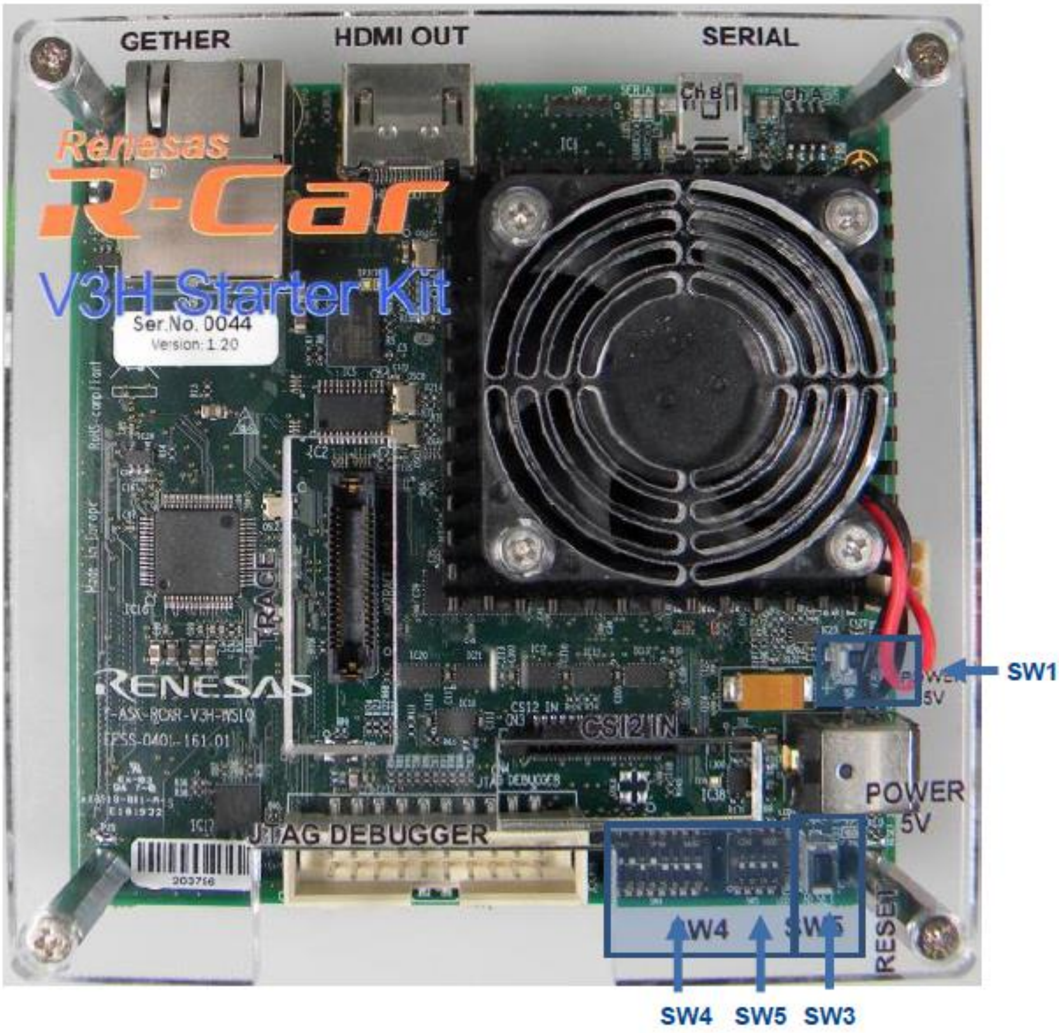


Table 6 V3H Ver.1.1, Ver.2.0 Starter Kit boot source selection

Switch Number	Switch Name	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW4	MODESW	ON	ON	OFF	ON	OFF	ON	OFF	OFF

For V3M Starter Kit, please refer to "R-Car V3M Starter Kit HW Manual (REN_r12ut0003ed0230-rcarv3m_MAT_20190124.pdf)" and find the below picture.

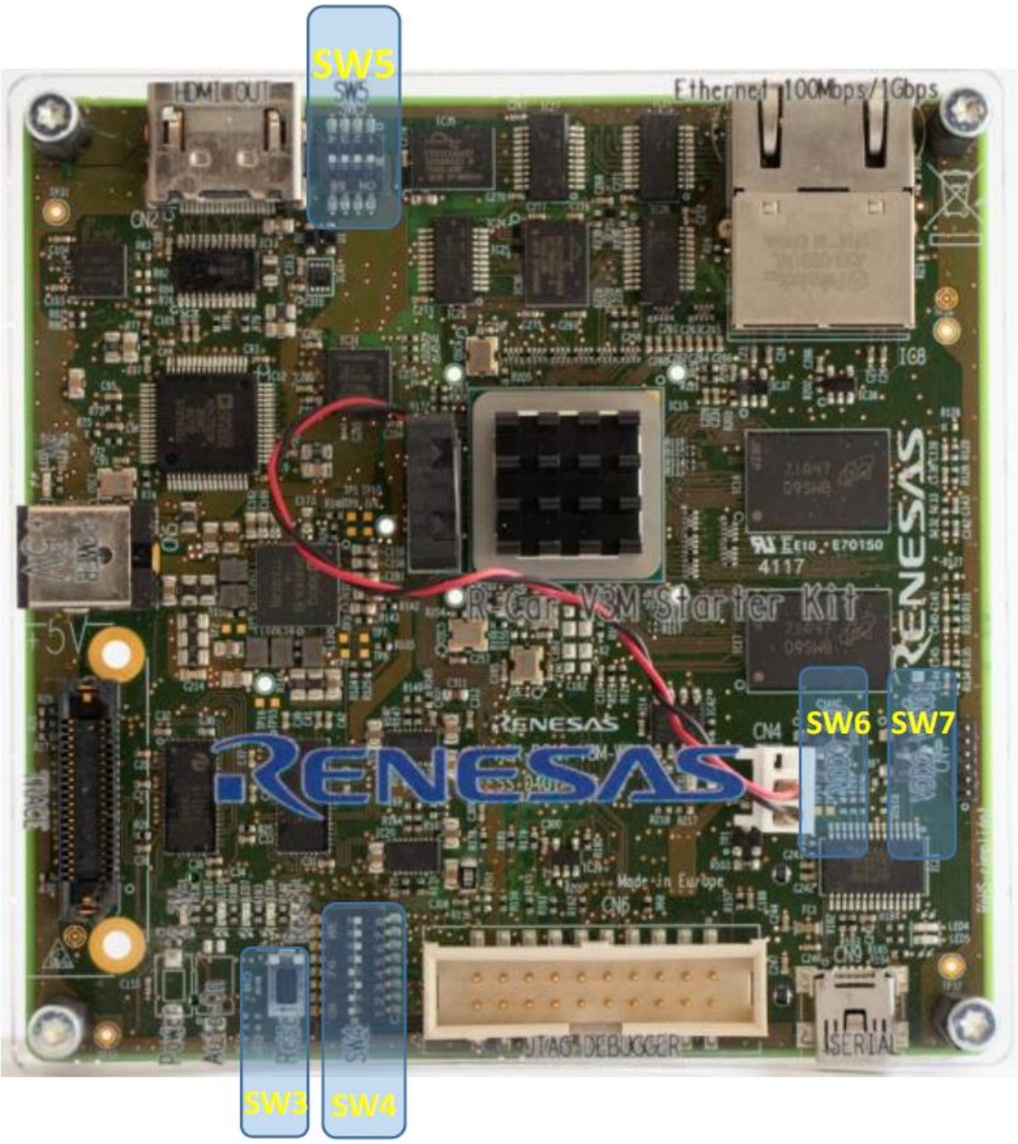


Table 7 V3M Ver.2.0 Starter Kit boot source selection

Switch Number	Switch Name	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW4	MODESW	ON	ON	OFF	ON	OFF	OFF	OFF	OFF

3. Building Instructions on Linux Host PC

3.1 Using build script

You can build BSP by using Yocto Project. Please execute following steps in ${WORK} directory on Linux Host PC. Root filesystem (which includes out of tree kernel modules and OSS tools) by making following instruction is the one for testing current BSP package in Renesas. Please note that Renesas has not been verified with any other build configuration or modified recipes except "rcar-image-adas".
Note) If you have executed Yocto build before, please remove your build directory (${WORK}/build) so that future builds happen from a clean environment.

Step 1 installation of required commands

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

Linux Host PC

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat cpio python3 python3-pip python3-pexpect \
xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa \
libsdl1.2-dev pylint3 xterm libarchive-zip-perl
```

Step 2 Execute the build script

Renesas provide the build script for users to simplify their build procedure. Please execute the script with following instruction:

Linux Host PC

```
$./build_yocto.sh <arg1> <arg2>
```

In which

- arg1 is board type, including "condor" or "falcon" or "eagle" or "all"
  - ○ "condor": all binaries of Condor, Condor-I and Condor Starter Kit will be compiled
  - ○ "eagle": all binaries of Eagle, Eagle Starter Kit will be compiled
  - ○ "falcon": all binaries of Falcon will be compiled
  - ○ "all" all binaries will be compiled
- arg2 is "adas", because ADAS software packages from Renesas is included.

After compiling successfully, the binaries will be created in ${WORK}/build/tmp/deploy/images/${supported_board}/ directory with the below description:

Root Filesystem: (rcar-image-adas-<v3u/v3x>.tar.bz2)

Image: file named "Image" (For V3U, V3H/V3M)

Device tree binaries:
- r8a77970-eagle.dtb (For Eagle V3M Ver.2.0)
- r8a77970-v3msk.dtb (For SK V3M Ver.2.0)
- r8a77970-v3msk-vbm.dtb (For SK V3M Ver.2.0 and Video Box Mini)
- r8a77980-condor.dtb (For V3H Ver.1.1)

- r8a77980-v3hsk.dtb (For SK V3H Ver.1.1)
- r8a77980-v3hsk-vbm.dtb (For SK V3H Ver.1.1 and Video Box Mini)
- r8a77980-es2-condor.dtb (For Condor-I V3H Ver.2.0)
- r8a77980-es2-condor.dtb (For Condor-I V3H Ver.2.1)
- r8a77980-es2-v3hsk.dtb (For SK V3H Ver.2.0)
- r8a77980-es2-v3hsk-vbm.dtb (For SK V3H Ver.2.0 and Video Box Mini)
- r8a779a0-falcon.dtb (For V3U Ver.1.0)

Uboot

- u-boot-elf-condor.srec (For Condor-I V3H Ver.2.0)
- u-boot-elf-v3hsk.srec (For SK V3H Ver.1.1, Ver.2.0)
- u-boot-elf-eagle.srec (For Eagle V3M)
- u-boot-elf-eagle-function.srec (For Eagle Function board V3M)
- u-boot-elf-v3msk.srec (For SK V3M Ver.2.0)
- u-boot-elf-falcon.srec (For V3U Ver.1.0)

3.2 Explanation about build script

This is the manual step in the build script:

Step 1 download of required files

Required files (poky) are downloaded by git clone.

Linux Host PC

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

Step 2 checkout

Please checkout available version of each git clone.

Linux Host PC

```
$ cd ${WORK}/poky
$ git checkout -b tmp <commit ID>

$ cd ${WORK}/meta-openembedded
$ git checkout -b tmp <commit ID>

$ cd ${WORK}/meta-renesas
$ git checkout -b tmp <commit ID>
```

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

In case of BSP + ADAS packages

When you want to use ADAS software packages from Renesas, please execute as following steps.

Step 3 execute source command

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

Linux Host PC

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

Step 4 copy bblayers.conf and local.conf

Please copy configuration files from deliverables.

Linux Host PC

```
$ cp ${WORK}/meta-renesas/meta-rcar-bsp/docs/sample/conf/${supported_board}/poky-gcc/adas/bblayers.conf conf/bblayers.conf
$ cp ${WORK}/meta-renesas/meta-rcar-bsp/docs/sample/conf/${supported_board}/poky-gcc/adas/local.conf conf/local.conf
```

Note) ${supported_board} is the one of the following: falcon, condor, eagle.

Step 5 building with bitbake

Please build as follows. The file system (rcar-image-adas-<v3u/v3x>.tar.bz2) is created in ${WORK}/build/tmp/deploy/images/${supported_board}/ directory.

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

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

Linux Host PC

```
$ cd ${WORK}/build
$ bitbake rcar-image-adas
```

4. Writing boot loaders and u-boot

In detail, please refer Flash writer manual (RENESAS_RCV3HV3U_ICUMXA_Flash_Writer_Application_Note_vx.pdf, RENESAS_RCV3M_CR7_Flash_Writer_Application_Note_vx.pdf).

Step 1 Power off target board, and turn to SCIF download mode and writing mode to 512-Mbit SPI flash memory by setting the following switches.

FALCON case:

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
Falcon CPU board										
SW50	MODESW	C	ALL OFF (*1)				OFF	ON (*2)	OFF	OFF
SW13	SPI/EX-SPI	C	✔	-		-	-	-	-	-

(*1) SCIF download mode

(*2) Booted through ICUMXA

Condor or Condor-I board case:

Switch Number	Switch Name	lide (C/S	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW3	MODESW-A	C	OFF (*1)	ON	ON	OFF	ALL OFF (*2)			
SW5	QSPI0-A	C	✔ (*3)	-		-	-	-	-	-
SW6	QSPI0-B	C	✔ (*3)	-		-	-	-	-	-
SW7	QSPI0-C	C	ALL OFF (*3)					-	-	-

(*1) Booted through ICUMXA

(*2) SCIF download mode

Eagle board case:

Switch Number	Switch Name	Slide (C/S)	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6
SW3	MODESW-A	C	OFF (*1)	ON	ALL OFF (*2)			
SW5	QSPI0-A	C	✔ (*3)	-		-	-	-
SW6	QSPI0-B	C	✔ (*3)	-		-	-	-
SW7	QSPI0-C	C	ALL OFF (*3)					-

(*1) Booted through Cortex-R7

(*2) SCIF download mode

(*3) Switch setting is when writing to 512-Mbit SPI flash memory.

Starter Kit V3H case**:**

Switch Number	Switch Name	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8
SW4	MODESW	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF

Starter Kit V3M case**:**

As it is described in “RENESAS_RCV3M_CR7_Flash_Writer_Application_Note_vx.pdf”, execute 1-5 to set SCIF download mode

with Configuration Tool (V3M_StarterKit_Configurator.exe). Then go to Step4 (4-2).

1. Connect USB Host connector of Windows PC that is virtual COM port to CN9 of Starter Kit V3M with USB cable.

2. Power ON Starter Kit V3M board.

3. Execute the Configuration Tool.

   - Select "On board peripheral configuration Tab > QSPI device > on-board QSPI0".

     Then press "Write CPLD (volatile)".

   - Select "SOC mode Configuration Tab > MODE setting > Set by this register".

     and select "SOC mode Configuration Tab > Boot from [MD4/3/2/1] > SCIF0".

     Then press "Write CPLD (volatile)".

4. Close the Configuration Tool.

5. Activate the Terminal Software on Windows PC.

6. Press the 'RESET'(SW3) on Starter Kit V3M board.

Step 2 Connect cable

Connect USB Host connector of Windows PC that is virtual COM port to CN10 of FALCON board with USB cable for displaying console.

Connect USB Host connector of Windows PC that is virtual COM port to CN11 of Eagle, Condor or Condor-I board with USB cable for displaying console.

Connect USB Host connector of Windows PC that is virtual COM port to CN6 of Starter Kit V3H with USB cable for displaying console.

Step 3 Set the terminal software

Activate the Terminal Software on Windows PC.

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

Step 4 Run Flash writer tool

(4-1) Power ON target board.

Terminal software outputs the following log.

Windows PC for terminal software

```
SCIF Download mode (w/o verification)

(C) Renesas Electronics Corp.

-- Load Program to RT-SRAM ---------------

please send !
```

(4-2) Please drag and drop the following Flash writer tool mot file.

(FALCON board case)

/os/bootloader/(v3u)/(atf or smoni)/ ICUMXA_Flash_writer_SCIF_DUMMY_CERT_EB200000_falcon.mot

(Condor, Condor-I board case) and (Starter Kit V3H case)

/os/bootloader/(v3h1 or v3h2)/(atf or smoni)/ICUMXA_Flash_writer_SCIF_DUMMY_CERT_EB200400_condor.mot

(Eagle board case) and (Starter Kit V3M case)

/os/bootloader/v3m2/(atf or smoni)/AArch32_Flash_writer_SCIF_DUMMY_CERT_E6300400_EAGLE.mot

When the transfer is successful, the following log is output. (The SOC and version will be changed.)

Windows PC for terminal software

```
Flash writer for R-Car V3U Series V1.15 Sep.15,2020
```

About FALCON case, set the following switches to use QSPI Flash with keeping the board power on.

Switch Number	Switch Name	Side (C/S)	Pin1	Pin2	Pin3	Pin4	Pin5	Pin6	Pin7	Pin8
Falcon CPU board										
SW50	MODESW	C	ON	ON	OFF	ON	OFF	ON	OFF	OFF

Step 5 Write data file to the Serial Flash

Write each S-record format images of loaders to the Serial Flash by flashing tool which is transferred on Step4.

(5-1) Writing data

FALCON board case:

Folder path: /os/bootloader/(v3u)/(atf or smoni)

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'EB200000	H'000000	Loader (Boot parameter)
icumxa_loader.srec	H'EB300000	H'040000	Loader
cert_header_sa17.srec	H'EB200000	H'440000	Loader (Certification)
dummy_fw.srec	H'EB380000	H'480000	Dummy FW (instead of secure firmware)
dummy_rtos.srec	H'EB200000	H'580000	Dummy RTOS (instead of CR52 main OS)
bl31-falcon.srec (or smoni.srec)	H'46400000	H'680000	BL31 ATF (or Secure Monitor)
u-boot-elf-falcon.srec	H'50000000	H'780000	U-boot

Condor, Condor-I board case:

Folder path: /os/bootloader/(v3h1 or v3h2)/(atf or smoni)/

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'EB200000	H'000000	Loader (Boot parameter)
icumxa_loader.srec	H'EB2D8000	H'040000	Loader
dummy_fw.srec	H'EB2B4000	H'0C0000	Dummy FW (instead of secure firmware)
cert_header_sa6.srec	H'EB200000	H'180000	Loader (Certification)
dummy_rtos.srec	H'EB200000	H'1C0000	Dummy RTOS (instead of CR7 main OS)
bl31-condor.srec (or smoni.srec)	H'46400000	H'2C0000	BL31 ATF (or Secure Monitor)
u-boot-elf-condor.srec	H'50000000	H'840000	U-boot

Starter Kit V3H case:

Folder path: /os/bootloader/(v3h1 or v3h2)/(atf or smoni)/

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'EB200000	H'000000	Loader (Boot parameter)
icumxa_loader.srec	H'EB2D8000	H'040000	Loader
dummy_fw.srec	H'EB2B4000	H'0C0000	Dummy FW (instead of secure firmware)
cert_header_sa6.srec	H'EB200000	H'180000	Loader (Certification)
dummy_rtos.srec	H'EB200000	H'1C0000	Dummy RTOS (instead of CR7 main OS)
bl31-condor.srec (*1)	H'46400000	H'2C0000	BL31 ATF (*1)
u-boot-elf-v3hsk.srec	H'50000000	H'840000	U-boot

Note: Only “rcar-xos_s240918-axa-1_v1.x.0_release” , use Secure Monitor (smoni.srec))

Eagle board case:

Folder path: /os/bootloader/v3m2/(atf or smoni)/

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'E6320000	H'000000	Loader (Boot parameter)
cr7_loader.srec	H'E6304000	H'040000	Loader
cert_header_sa3.srec	H'E6320000	H'0C0000	Loader (Certification)
bl31-eagle.srec (or smoni.srec)	H'46400000	H'2C0000	BL31 ATF (or Secure Monitor)
dummy_rtos.srec	H'42220800	H'1C0000	Dummy RTOS (instead of CR7 main OS)
u-boot-elf-eagle.srec	H'50000000	H'840000	U-boot

Starter Kit V3M case:

Folder path: /os/bootloader/v3m2/atf/

Filename	Program Top Address	Flash Save Address	Description
bootparam_sa0.srec	H'E6320000	H'000000	Loader (Boot parameter)
cr7_loader_v3msk.srec	H'E6304000	H'040000	Loader
cert_header_sa3.srec	H'E6320000	H'0C0000	Loader (Certification)
bl31-eagle.srec	H'46400000	H'2C0000	BL31 ATF
dummy_rtos.srec	H'42220800	H'1C0000	Dummy RTOS (instead of CR7 main OS)
u-boot-elf-v3msk.srec	H'50000000	H'840000	U-boot

(5-2) Using xls2 command to write

The following is writing “FALCON bootparam_sa0.srec case” on (5-1) Writing data table.

(a) Input “xls2” command.

```
>xls2                                        Windows PC for terminal software
```

The following log is output.

```
                                             Windows PC for terminal software
===== Qspi/HyperFlash writing of Gen3 Board 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)>
```

(b) Select, 512-Mbit SPI flash. The following is FALCON case.

```
Select (1-3)>1                               Windows PC for terminal software
```

The following log is output.

```
                                             Windows PC for terminal software
READ ID OK.

          Program Top Address & Qspi/HyperFlash Save Address

          ===== Please Input Program Top Address ============

            Please Input : H'
```

(c) Input Program Top Address of the tareget file on (5-1) Writing data table

Windows PC for terminal software

```
> EB200000
```

The following log is output.

Windows PC for terminal software

```
===== Please Input Qspi/HyperFlash Save Address ===

              Please Input : H'
```

(d) Input Qspi/HyperFlash Save Address of the tareget file on (5-1) Writing data table

Windows PC for terminal software

```
> 0
```

The following log is output.

Windows PC for terminal software

```
Work RAM(H'50000000-H'53FFFFFF) Clear....

           please send ! ('.' & CR stop load)
```

(e) Drag and drop the target file on the table of (5-1) Writing data

The following log is output. **("bootparam_sa0.srec" case)**

Windows PC for terminal software

```
SPI Data Clear(H'FF) Check :H'00000000-0003FFFF,Clear OK?(y/n)
```

Push “y”key

Windows PC for terminal software

```
>y
```

Windows PC for terminal software

```
H'00000000-H'00007FFF Erasing.

........Erase Completed

H'00000000-0003FFFF Erasing..Erase Completed

SAVE SPI-FLASH....... complete!
```

(f) Repeat (a)-(e) until you complete writing each S-record format images of loaders

(5-3) After write to images, power off target board.

(5-4) Return the switches which were changed on Step1 to the switches on Chapter.2.2.

5. Confirm starting of U-Boot and Linux

Please refer to 2.2 for dip switch setting then follow the steps below

Step 1 set file system and store kernel image + dtb

Linux Host PC

```
$ mkdir /export/rfs
$ cd /export/rfs
```

Please store file system (rcar-image-adas-{SoC}.tar.bz2) to /export/rfs directory of NFS server, and extract it as follows.

Linux Host PC

```
$ sudo tar xvf rcar-image-adas-${SoC}.tar.bz2
```

Note) ${SoC} is v3u in case of V3U, v3x in case of V3H/V3M.

Please store kernel Image and device tree binaries (.dtb) to **/srv/tftp** of TFTP server

Step 2 Set U-Boot environment variables

Please refer to 2.2 for dip switch setting.

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

Windows PC for terminal software

```
=> setenv ethaddr xx:xx:xx:xx:xx:xx
=> setenv ipaddr 192.168.0.20
=> setenv serverip 192.168.0.1
=> setenv bootcmd 'tftp 0x48080000 Image;tftp 0x48000000
<Device_Tree>;booti 0x48080000 - 0x48000000'
```

Note) For R-Car V3U (r8a779a0), the <Device_Tree> are as follows:

- r8a779a0-falcon.dtb: for R-Car V3U Ver.1.0 on Falcon board.

For R-Car V3H (r8a77980), the <Device_Tree> are as follows:

- r8a77980-es2-condor.dtb: for R-Car V3H Ver.2.0 on Condor-I board.
- r8a77980-condor.dtb: for R-Car V3H Ver.1.1 on Condor board.
- r8a77980-es2-v3hsk.dtb: for R-Car V3H Ver.2.0 on Starter Kit.
- r8a77980-v3hsk.dtb: for R-Car V3H Ver.1.1 on Starter Kit.

For R-Car V3M (r8a77970), the <Device_Tree> are as follows:

- r8a77970-eagle.dtb: for R-Car V3M Ver.2.0 on Eagle board.
- r8a77970-v3msk.dtb: for R-Car V3M Ver.2.0 on Starter Kit board.

Step 3 change the bootargs by U-Boot

To change bootargs which is passed to the kernel in boot sequence, please modify it by “setenv bootargs” command of U-Boot. Below command means mount the root file system as read/write option via NFS server whose server IP is 192.168.0.1, root directory is /export/rfs and the client IP is 192.168.0.20

Windows PC for terminal software

```
=> setenv bootargs 'rw root=/dev/nfs nfsroot=192.168.0.1:/export/rfs
ip=192.168.0.20 cma=560M'
```

Step 4 save environment variables to save persistent storage

By the "saveenv" command, we do not need to execute Step5-Step7 for next time.

Windows PC for terminal software

```
=> saveenv
```

Step 5 start Linux

After board reset, U-Boot is started. After countdown, Linux boot messages are displayed. Please confirm login prompt after Linux boot messages.

Appendix:

How to install TFTP server and NFS server

How to install TFTP server

Note: make sure you are **connected to the internet**. If there is an error, try *`apt update -y`* to update your application repository database.

1.Install tftpd-hpa with sudo privilege

Linux Host PC

```
$ sudo apt install tftpd-hpa -y
```

Note: -y flag means to assume yes and silently install, without asking you questions in most cases.
Once the installation is complete, you will be able to download files from the TFTP server.
The default configuration file for tftpd-hpa is /etc/default/tftpd-hpa**.**
The default root directory where files will be stored is /srv/tftp
If you want to upload to the TFTP server, please continue next step.

2. Edit tftpd-hpa configuration file

Linux Host PC

```
$ sudo vi /etc/default/tftpd-hpa
```

And change the following line to give R/W permission to TFTP_DIRECTORY folder:

[Old] `TFTP_OPTIONS="--secure"`
[New] **`TFTP_OPTIONS="--secure --create"`**

and save the file and exit the vi editor.

3. Restart the tftpd-hpa Service

To make the changes take effect, the tftpd-hpa service must be restarted. This can be accomplished by performing the following command.

Linux Host PC

```
$ sudo service tftpd-hpa restart
```

At this point you should now have a TFTP server that allows you to both download and upload files.

How to install NFS server

To boot an embedded system over NFS, an NFS server must be available on the local network. First, we need to install the NFS server. Once the server has been installed, export a directory to use as the root filesystem. This is often done using the `/etc/exports` file. This document assumes that the root filesystem for the board will be located at `/export/rfs` on the NFS server.

Note: make sure you are **connected to the internet**. If there is an error, try *`apt update -y`* to update your application repository database.

1.Install the NFS kernel server package and rpcbind. with sudo privilege

Linux Host PC

```
$ sudo apt install nfs-kernel-server
$ sudo apt install rpcbind
```

This installs additional packages such as keyutils, nfs-common, rpcbind, and other dependencies required for the NFS server to function as expected.

2. Configuring the NFS Server

The NFS server's shares are configured in the /etc/exports file. Each share takes a list of option names to configure its behavior. Before configuring the NFS server, the following decisions need to be made (and a list of options created) based on the intended use(s) of the NFS share being created:

- Do NFS clients need the ability to write to the NFS share? If so, it needs to be exported `rw`. Otherwise, export it as `ro`
- If the filesystem will be writeable, is performance or data integrity more important? If performance, async will provide more performance at the risk of lost or corrupt data should the NFS server shut down unexpectedly during a write operation. Otherwise, `sync` should be used.
- Is reliability or security more important? If reliability is, use `no_subtree_check`. Otherwise, use `subtree_check`.
- Is user level access control required? If not, getting the NFS server working is much simpler with the `no_root_squash` option.

Here is an example of the /etc/exports file that to export the `/export/rfs` directory to all machines, writeable, with reliability and performance options turned on

Linux Host PC

```
$ sudo vi /etc/exports
```

And modify as follow

```
/export/rfs *(rw,async,no_root_squash)
```

3. Notify the NFS server after making any changes to the export file

Linux Host PC

```
$ exportfs -a
```

4. Activate NFSv2

```
$ sudo vi /etc/default/nfs-kernel-server
```

Linux Host PC

And add as follow

RPCNFSDOPTS="$RPCNFSDOPTS -V 2"

5. Restart the nfs Service

```
$ sudo service nfs-kernel-server restart
```

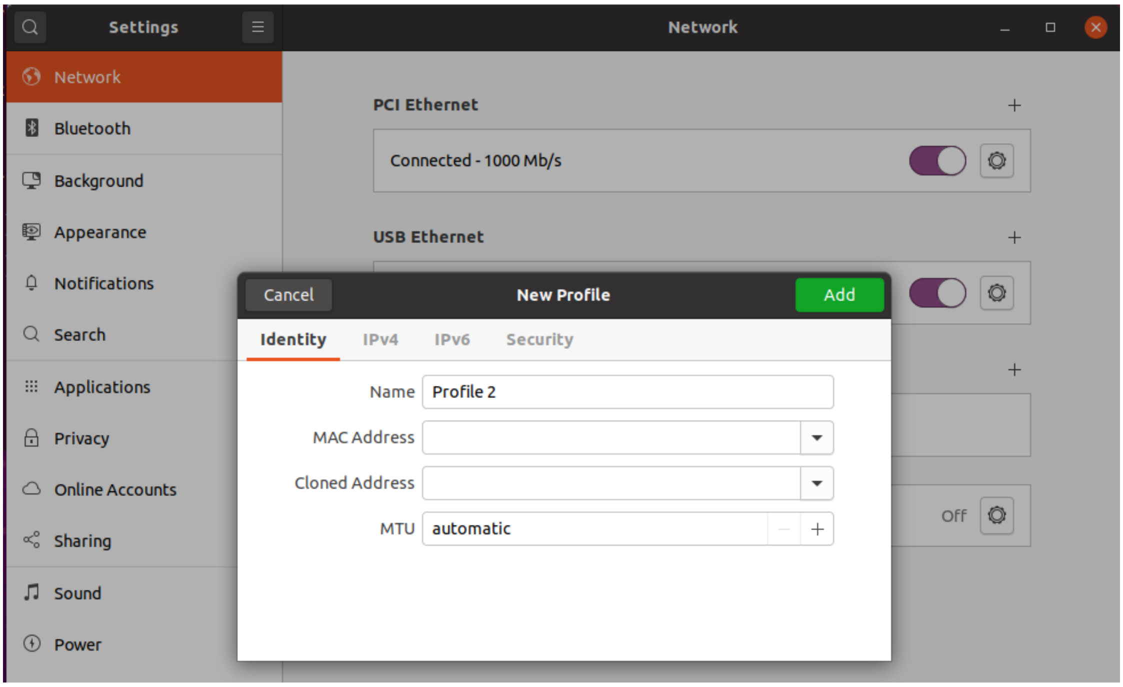
Linux Host PC

Now users can configure the Client to boot from NFS server.

How to set IP address to Linux Host PC

In this case, we confirm with USB3.0 Gigabit LAN adapter.

1. Open Ubuntu > Settings > Network > USB Ethernet > "Identity" Tab



Name: Profile 2

MAC Address:

Cloned Address:

MTU: automatic

2. Select target MAC address which can be check with "lsusb" and "ip add" command.

3. Select "IPv4" tab > "IPv4 Method > Manual".

Set as follows.

Address	Netmask	Getway
192.168.0.1	255.255.255.0	0.0.0.0

3.Click "Add" and reboot Linux Host PC.

Exporting Toolchains

Please refer Documents from Yocto Project to export Toolchains such as
https://www.yoctoproject.org/docs/3.1.3/sdk-manual/sdk-manual.html.
And please use build target of bitbake as “rcar-image-adas -c populate_sdk” to generate package.

Note) When you use “ld” directly but not via gcc (in case of building Kernel, Driver or U-Boot), please disable LDFLAGS with ‘unset LDFLAGS’. Furthermore, in kernel build, ‘make menuconfig’ occurs error by ncurses. In this case, please set PKG_CONFIG_PATH and disable PKG_CONFIG_SYSROOT_DIR.

Linux Host PC

```
$ export PKG_CONFIG_PATH=$OECORE_NATIVE_SYSROOT/usr/lib/pkgconfig
$ unset PKG_CONFIG_SYSROOT_DIR
```

Note) Please do not use same shell environment to other compilation/debugging purpose (also make menuconfig of linux kernel, e.g.) but cross compilation for R-Car V3U/V3H/V3M which shell environment with “source” command to setup environment variables for the SDK. Because some environment variables for cross compilation interferes execution of other tools on the same shell environment.

Example of instruction:

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

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

Please modify SDKMACHINE description on ${WORK}/build/conf/local.conf.

```
On ${WORK}/build/conf/local.conf
# SDK target architecture
#
# This variable specifies the architecture to build SDK items for and means
# you can build the SDK packages for architectures other than the machine you are
# running the build on (i.e. building i686 packages on an x86_64 host).
# Supported values are i686 and x86_64
SDKMACHINE ?= "x86_64"
```

Note) 32bit Ubuntu 14.04 is not supported.

Step 2 building toolchain package with bitbake

Linux Host PC

```
$ cd ${WORK}/build
$ bitbake rcar-image-adas -c populate_sdk
$ cp tmp/deploy/sdk/poky-glibc-x86_64-rcar-image-adas-aarch64-${SoC}-
toolchain-3.1.3.sh (shared dir. where able to access from each Host PCs)
```

Note) ${SoC} is v3u in case of V3U, v3x in case of V3H/V3M.

Note) Please perform “bitbake rcar-image-minimal -c populate_sdk” in BSP Only.

Step 3 Install toolchain on each Host PCs

```
Linux Host PC
$ sudo (shared dir. where able to access from each Host PCs)/poky-glibc-x86_64-rcar-image-adas-aarch64-<board name>-toolchain-3.1.3.sh

 [sudo] password for (INSTALL person): (password of your account)
Enter target directory for SDK (default: /opt/poky/3.1.3): (just a return)
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

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

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

```
Linux Host PC
$ cd (Your working directory)
$ source /opt/poky/3.1.3/environment-setup-aarch64-poky-linux
$ export LDFLAGS=“”
$ ${CC} (Your source code).c …
```

Note for generating & using Windows SDK

Please do the following steps to create Windows SDK:

Step 1 prepare meta-mingw

```
Linux Host PC
$ cd ${WORK}
$ git clone https://git.yoctoproject.org/git/meta-mingw
$ cd meta-mingw
$ git checkout -b tmp 524de686205b5d6736661d4532f5f98fee8589b7
$ cd ${WORK}/build
$ bitbake-layers add-layer ../meta-mingw/
```

Step 2 edit SDKMACHINE in local.conf

```
On ${WORK}/build/conf/local.conf

SDKMACHINE ?= "x86_64-mingw32"
```

Step 3 building toolchain (Windows SDK) package with bitbake

```
$ bitbake rcar-image-adas -c populate_sdk                  Linux Host PC
```

After the build finished, the Windows SDK locates at **${WORK}/build/tmp/deploy/sdk** with the following files:

poky-glibc-x86_64-mingw32-rcar-image-adas-aarch64-${SoC}-toolchain-3.1.3.host.manifest
poky-glibc-x86_64-mingw32-rcar-image-adas-aarch64-${SoC}-toolchain-3.1.3.target.manifest
poky-glibc-x86_64-mingw32-rcar-image-adas-aarch64-${SoC}-toolchain-3.1.3.tar.xz (Windows SDK tarball)
poky-glibc-x86_64-mingw32-rcar-image-adas-aarch64-${SoC}-toolchain-3.1.3.testdata.json

Note) ${SoC} is v3u in case of V3U, v3x in case of V3H/V3M.

Step 4 re-zip Windows SDK tarball to fix symlinks broken issue

```
                                                            Linux Host PC
$ windows_sdk=”poky-glibc-x86_64-mingw32-rcar-image-adas-aarch64-
${SoC}-toolchain-3.1.3”
$ cd ${WORK}/build/tmp/deploy/sdk
$ tar -xf ${windows_sdk}.tar.xz -C ${windows_sdk}
$ zip -r ${windows_sdk}.zip ${windows_sdk}
```

Note) Please do the following steps to generate Windows SDK 7z format:

1) unzip Windows SDK zip on Windows PC.
2) Re-zip by 7z application.

In Step 1), warnings might occur, because Windows cannot distinguish between file names with upper case and lower case. So, please do not overwrite existing files, if asked.

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

On Windows cmd:

```
                                                            Windows PC
> cd %SDK_PATH%
> environment-setup-aarch64-poky-linux.bat
> %CC% --version
aarch64-poky-linux-gcc (GCC) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
> %CC% <your source code>.c
```

Out-of-tree driver information

Out of tree drivers in **rcar-image-adas-v3x.tar.bz2** are as follows:

- CMEM driver:
    /lib/modules/5.4.72-yocto-standard/extra/cmemdrv.ko
    /usr/include/linux/cmemdrv.h

- MMNGR driver:
    /lib/modules/5.4.72-yocto-standard/extra/mmngr.ko
    /lib/modules/5.4.72-yocto-standard/extra/mmngrbuf.ko
    /usr/lib/libmmngr.so
    /usr/lib/libmmngrbuf.so
    /usr/local/include/mmngr_buf_private_cmn.h
    /usr/local/include/mmngr_buf_user_private.h
    /usr/local/include/mmngr_buf_user_public.h
    /usr/local/include/mmngr_private_cmn.h
    /usr/local/include/mmngr_public_cmn.h
    /usr/local/include/mmngr_user_private.h
    /usr/local/include/mmngr_user_public.h

- QoS driver:
    /lib/modules/5.4.72-yocto-standard/extra/qos.ko
    /usr/lib/ibqos.so
    /bin/qos_tp
    /include/qos_public.h
    /include/qos_public_common.h

- CPU RTT driver for xOS1.1:
    /lib/modules/5.4.72-yocto-standard/extra/cpurttmod.ko
    /usr/local/include/cpurtt_common.h
    /usr/local/include/cpurtt_common_userdef.h

- CPU RTT driver for xOS2:
    /lib/modules/5.4.72-yocto-standard/extra/cpurttmod2_v3h2.ko
    /lib/modules/5.4.72-yocto-standard/extra/cpurttmod2_v3h1.ko
    /lib/modules/5.4.72-yocto-standard/extra/cpurttmod2_v3m2.ko
    /usr/local/include/cpurttmod2_v3m2/cpurtt_common.h
    /usr/local/include/cpurttmod2_v3h1/cpurtt_common.h
    /usr/local/include/cpurttmod2_v3h2/cpurtt_common.h

Out of tree drivers in **rcar-image-adas-v3u.tar.bz2** are as follows:

- CMEM driver:
    /lib/modules/5.4.72-yocto-standard/extra/cmemdrv.ko
    /usr/include/linux/cmemdrv.h

- MMNGR driver:
    /lib/modules/5.4.72-yocto-standard/extra/mmngr.ko
    /lib/modules/5.4.72-yocto-standard/extra/mmngrbuf.ko

/usr/lib/libmmngr.so
/usr/lib/libmmngrbuf.so
/usr/local/include/mmngr_buf_private_cmn.h
/usr/local/include/mmngr_buf_user_private.h
/usr/local/include/mmngr_buf_user_public.h
/usr/local/include/mmngr_private_cmn.h
/usr/local/include/mmngr_public_cmn.h
/usr/local/include/mmngr_user_private.h
/usr/local/include/mmngr_user_public.h

- QoS driver:
/lib/modules/5.4.72-yocto-standard/extra/qos.ko
/usr/lib/ibqos.so
/bin/qos_tp
/include/qos_public.h
/include/qos_public_common.h

CONFIDENTIAL

REVISION HISTORY	Linux Interface Specification Yocto recipe Start-Up Guide User's Manual: Software		
Rev.	**Date**	**Description**	
		Page	**Summary**
4.4.0	Sep 30, 2020	—	First Edition for Yocto Recipe Package with YP3.1, based on Start-Up Guide Rev 4.1.0
4.7.0	Dec 24, 2020	—	Added V3H information
		2	Updated reference versions Added "R-Car Series, 3rd Generation User's Manual: Hardware Rev.1.50"
		3	Table 1 Updated VCP driver information
		5	Table 2 Updated setting of Dip switches for V3U Falcon according to "R-CarV3U System Evaluation Board FALCON Setup Manual Rev.0.07"
		6	Table 3 Added setting of Dip switches (R-Car V3H Condor-I)
		7	Step 1 Updated required packages according to YP3.1.3
		8	Step 3 Updated Commit ID, and the Notice for V3U
		9-10	Step 5 Updated path to sample configuration files
		12-13	Step 5 Added R-Car V3H device tree information
		14-15	Updated SDK information for YP3.1.3
4.9.0	Feb 26, 2021	5 - 6	Updated Dip switch setting information
		8	Step 3 Updated Commit ID, and the Notice for V3U
		12	Step 4 Updated U-Boot Flash Save Address for V3U and V3H
		13, 15	Added note for build output of Condor
4.10.0	Mar 31, 2021	—	Added V3M information Added Ubuntu version 20.04 LTS
		2	Updated reference information
		3	Added Eagle board driver information
		4-8	2.2 Added Eagle/Condor dip switch setting.
		10	Step 3 Updated Commit ID NOTICE for V3U Added patch "0004-media-rcar-vin-Remove-reset-function.patch"
		11,12	Added Eagle board, improved guideline
		13	Added "3.3 Creating CRC checksum of root filesystem (*.wic.gz.crc)"
		17-19	"6. Exporting Toolchains" Added Windows SDK information
4.11.1	Apr 27, 2021	—	Added V3H Starter Kit information
		10	Step 3 Updated Commit ID
		15	Step 4 Update Flash Save Address of V3U (Falcon board)
4.11.2	Apr 27, 2021	10	Step 3 Updated Commit ID
4.12.0	May 28, 2021	—	Removed "3.3 Creating CRC checksum of root filesystem (*.wic.gz.crc)" Removed Ubuntu 16.04 information
		9	Step 1 Added libarchive-zip-perl which provides crc32 command
		10	Step 3 Updated Commit ID Remove "0004-media-rcar-vin-Remove-reset-function.patch"
4.13.1	June 30, 2021	2	Add a flowchart for using this Start Up Guide
		3	Article 1.1: Add document name
		4	Chapter 2: Add explanation of TFTP server and NFS server, IP address, MAC address
		7-16	Add board picture from Set Up guide.

		17-19	Chapter 3: Add flowchart for Yocto build instruction and build script, remove additional VIN driver patch and MAX96712.cfg file
		22-29	Chapter 4: Add writing boot loaders and U-boot
		32-35	Guideline how to set up TFTP server, NFS server, IP address
4.14.0	July 30, 2021	3	Add R-Car V3M Starter Kit HW Manual
		18	Add setting dip switches for V3M Starter Kit
		20, 21	Add V3M Starter Kit device tree and U-boot
		24	Update Flash write manual reference
		26	Update V3U flash writer name
		25, 29	Add writing boot loaders and u-boot for V3M Starter Kit
		34	Add device tree for V3M Ver.2.0 on Starter Kit board
4.15.0	Aug 27, 2021	6	Add internet connection for Linux host PC
		19	Add note to use clean build directory
		20	Add u-boot-elf-eagle-function.srec (For Eagle Function board V3M)
4.16.0	Sep 24, 2021	1,2	Update flowchart, add out-of-tree drivers note.
		19,20	Add device tree of Starter Kit and Video Box Mini
		39	Add information of out-of-tree kernel modules
4.19.0	Dec 24, 2021	31	Add `cma=560M option to bootargs`
4.19.1	Dec 24, 2021	20	Update github urls to https protocol

CONFIDENTIAL

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

Publication Date: February 25, 2022 Rev.4.19.2

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

Colophon 7.0

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

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

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

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

Colophon 3.1

Linux Interface Specification
Yocto recipe Start-Up Guide

Renesas Electronics Corporation