

SMARC EVK of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five

R01US0556EJ0103

Rev.1.03

Dec. 27, 2022

Start-up Guide

Introduction

This document provides a guide to prepare RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five reference boards to boot up with the Verified Linux Package of RZ/G2L Group and the Board Support Package of RZ/V2L Group and RZ/Five. Especially, procedures to write the bootloaders to each board are explained.

Bootloaders are written to the Flash ROM on the board using the Flash Writer tool provided by Renesas through the minimonitor utility. This document explains the way to write these files using the Flash Writer.

After obtaining your reference board, please be sure to follow this document and write the bootloaders to the Flash ROM before starting the evaluation.

Target Reference Board

RZ/G2L reference board

- RZ/G2L Evaluation Board Kit PMIC version (smarc-rzg2l-pmic) (*1)
 - RZ/G2L SMARC Module Board (P/N: RTK9744L23C01000BE)
 - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)

RZ/G2LC reference board

- RZ/G2LC Evaluation Board Kit (smarc-rzg2lc) (*2)
 - RZ/G2LC SMARC Module Board (P/N: RTK9744C22C01000BE)
 - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)

RZ/G2UL reference board

- RZ/G2UL Evaluation board Kit (smarc-rzg2ul) (*3)
 - RZ/G2UL SMARC Module Board (P/N: RTK9743U11C01000BE)
 - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)
 - Parallel to HDMI Conversion board (This board is included with RTK9743U11C01000BE and is not sold separately, so a part number does not exist.)

RZ/V2L reference board

- RZ/V2L Evaluation Board Kit PMIC version (smarc-rzv2l-pmic) (*4)
 - RZ/V2L SMARC Module Board (P/N: RTK9754L23C01000BE)
 - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)

RZ/Five reference board

- RZ/Five Evaluation board Kit (smarc-rzfive) (*5)
 - RZ/Five SMARC Module Board (P/N: RTK9743F01C01000BE)
 - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)



Please refer to Appendix 2.2 and 2.3 for how to distinguish and replace each board.



Please refer to Appendix 2.4 for how to connect the Parallel to HDMI Conversion board for RZ/G2UL Evaluation board Kit (smarc-rzg2ul).

- (*1) “RZ/G2L Evaluation Board Kit PMIC” includes the RZ/G2L SMARC Module Board and the RZ SMARC Series Carrier Board.
- (*2) “RZ/G2LC Evaluation Board Kit” includes the RZ/G2LC SMARC Module Board and the RZ SMARC Series Carrier Board.
- (*3) “RZ/G2UL Evaluation board Kit” includes the RZ/G2UL SMARC Module Board and the RZ SMARC Series Carrier Board.
The “Evaluation board Kit for RZ/G2UL MPU” will be called “RZ/G2UL Evaluation Kit” in the next section.
- (*4) “RZ/V2L Evaluation Board Kit PMIC” includes the RZ/V2L SMARC Module Board and the RZ SMARC Series Carrier Board.
- (*5) “RZ/Five Evaluation board Kit” includes the RZ/Five SMARC Module Board and the RZ SMARC Series Carrier Board.
The “Evaluation board Kit for RZ/Five MPU” will be called “RZ/Five Evaluation Kit” in the next section.

Target Software

- RZ/G Verified Linux Package version 3.0.1 or later. (hereinafter referred to as “VLP/G”)
- RZ/V Verified Linux Package version 3.0.2 or later. (hereinafter referred to as “VLP/V”)
- RZ/Five Verified Linux Package version 3.0.2 or later. (hereinafter referred to as “VLP/F”)

Contents

1. Renesas SMARC Evaluation Board kit of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five	4
1.1 Preparation before startup	4
1.1.1 Preparation of Hardware and Software	4
1.1.2 How to set boot mode and input voltage	5
1.1.3 How to set SW1	6
1.1.4 How to use debug serial (console output)	7
1.2 Startup Procedure	8
1.2.1 Power supply	8
1.2.2 Building files to write	9
1.2.3 Settings	10
1.3 Booting Flash Writer	12
1.4 Writing Bootloader	14
1.5 Create a microSD card to boot Linux	16
1.6 Setting U-boot	21
2. Appendix	24
2.1 Preparing Flash Writer	24
2.1.1 Preparing cross compiler	24
2.1.2 Building Flash Writer	25
2.2 How to distinguish each board	27
2.3 How to replace the SMARC Module Board	28
2.4 How to connect the Parallel to HDMI Conversion board for RZ/G2UL Evaluation kit (smarc-rzg2ul)	29
2.5 How to boot from eMMC	31
2.5.1 Rebuild rootfs	31
2.5.2 Writing Bootloader for eMMC Boot	31
2.5.3 Create a microSD card to boot linux for eMMC boot	33
2.5.4 Setting U-boot and Writing rootfs to eMMC	34
2.5.5 Setting U-boot for eMMC boot	36
3. Revision History	37

1. Renesas SMARC Evaluation Board kit of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five

1.1 Preparation before startup

1.1.1 Preparation of Hardware and Software

The following environment of Hardware and Software is used in the evaluation.

Hardware preparation:

- USB Type-C cable “AK-A8485011” (manufactured by Anker)
- USB PD Charger Anker “PowerPort III 65W Pod” (manufactured by Anker)
- USB Type-microAB cable (Any cables)
- micro HDMI cable (Any cables) (*1)
- PC Installed FTDI VCP driver and Terminal software (Tera Term) (*2)

(*1) RZ/Five Evaluation board Kit doesn't need micro HDMI cable.

(*2) Please install the FTDI driver that can be following website (<https://www.ftdichip.com/Drivers/VCP.htm>).

Software preparation:

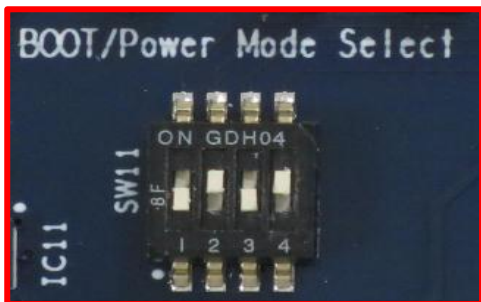
Table 1. Software Preparation

RZ/G2L PMIC ver	Linux kernel	Image-smarc-rzg2l.bin
	Device tree file	Image-r9a07g044l2-smarc.dtb
	Boot loader	<ul style="list-style-type: none"> • bl2_bp-smarc-rzg2l_pmic.srec • fip-smarc-rzg2l_pmic.srec
	Flash Writer	Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
RZ/G2LC	Linux kernel	Image-smarc-rzg2lc.bin
	Device tree file	Image-r9a07g044c2-smarc.dtb
	Boot loader	<ul style="list-style-type: none"> • bl2_bp-smarc-rzg2lc.srec • fip-smarc-rzg2lc.srec
	Flash Writer	Flash_Writer_SCIF_RZG2LC_SMARC_DDR4_1GB_1PCS.mot
RZ/G2UL	Linux kernel	Image-smarc-rzg2ul.bin
	Device tree file	Image-r9a07g043u11-smarc.dtb
	Boot loader	<ul style="list-style-type: none"> • bl2_bp-smarc-rzg2ul.srec • fip-smarc-rzg2ul.srec
	Flash Writer	Flash_Writer_SCIF_RZG2UL_SMARC_DDR4_1GB_1PCS.mot
RZ/V2L	Linux kernel	Image-smarc-rzv2l.bin
	Device tree file	Image-r9a07g054l2-smarc.dtb
	Boot loader	<ul style="list-style-type: none"> • bl2_bp-smarc-rzv2l_pmic.srec • fip-smarc-rzv2l_pmic.srec
	Flash Writer	Flash_Writer_SCIF_RZV2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
RZ/Five	Linux kernel	Image-smarc-rzfive.bin
	Device tree file	r9a07g043f01-smarc.dtb
	Boot loader	<ul style="list-style-type: none"> • fit-smarc-rzfive.srec • spl-smarc-rzfive.srec
	Flash Writer	Flash_Writer_SCIF_RZFIVE_SMARC.mot

1.1.2 How to set boot mode and input voltage

Please set the SW11 settings as follows.

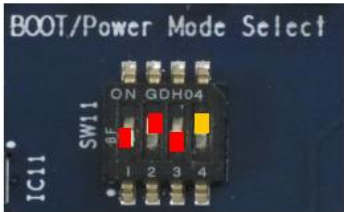



- Pin no1 to no3 of the SW11 is used to control boot mode of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five.
- Pin no4 of the SW11 is used to control the input voltage from power charger to 5V or 9V. Please use a 5V setting as initial setting.



SW11-1	OFF
SW11-2	ON
SW11-3	OFF
SW11-4	ON

Please select boot mode as below figures.

Currently we support 2 modes in 4 modes: SCIF Download mode and QSPI Boot mode. eSD Boot mode will be supported by the future update.

SCIF Download Mode(*1)	QSPI Boot (1.8V) Mode																
 <table border="1" style="margin-left: 20px;"> <tr><td>SW11-1</td><td>OFF</td></tr> <tr><td>SW11-2</td><td>ON</td></tr> <tr><td>SW11-3</td><td>OFF</td></tr> <tr><td>SW11-4</td><td>ON</td></tr> </table>	SW11-1	OFF	SW11-2	ON	SW11-3	OFF	SW11-4	ON	 <table border="1" style="margin-left: 20px;"> <tr><td>SW11-1</td><td>OFF</td></tr> <tr><td>SW11-2</td><td>OFF</td></tr> <tr><td>SW11-3</td><td>OFF</td></tr> <tr><td>SW11-4</td><td>ON</td></tr> </table>	SW11-1	OFF	SW11-2	OFF	SW11-3	OFF	SW11-4	ON
SW11-1	OFF																
SW11-2	ON																
SW11-3	OFF																
SW11-4	ON																
SW11-1	OFF																
SW11-2	OFF																
SW11-3	OFF																
SW11-4	ON																
eMMC Boot (1.8V) Mode	eSD Boot Mode																
 <table border="1" style="margin-left: 20px;"> <tr><td>SW11-1</td><td>ON</td></tr> <tr><td>SW11-2</td><td>OFF</td></tr> <tr><td>SW11-3</td><td>OFF</td></tr> <tr><td>SW11-4</td><td>ON</td></tr> </table>	SW11-1	ON	SW11-2	OFF	SW11-3	OFF	SW11-4	ON	 <table border="1" style="margin-left: 20px;"> <tr><td>SW11-1</td><td>ON</td></tr> <tr><td>SW11-2</td><td>ON</td></tr> <tr><td>SW11-3</td><td>OFF</td></tr> <tr><td>SW11-4</td><td>ON</td></tr> </table>	SW11-1	ON	SW11-2	ON	SW11-3	OFF	SW11-4	ON
SW11-1	ON																
SW11-2	OFF																
SW11-3	OFF																
SW11-4	ON																
SW11-1	ON																
SW11-2	ON																
SW11-3	OFF																
SW11-4	ON																

Please select input voltage setting as below.

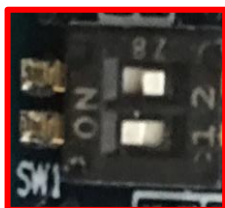
SW11-4	Input voltage selection
OFF	Input 9V
ON	Input 5V

1.1.3 How to set SW1

Renesas SMARC RZ/G2L Evaluation kit PMIC (smarc-rzg2l-pmic), RZ/G2LC Evaluation kit (smarc-rzg2lc) and RZ/V2L Evaluation kit PMIC (smarc-rzv2l-pmic)

Please set the SW1 settings to follows.

- Pin no1 of the SW1 is used to select the JTAG debug mode or not. JTAG is not used, so set SW1-1 to normal operation mode.
- Pin no2 of the SW1 is used to select the eMMC or microSD mode. Please set SW1-2 to eMMC mode.



SW1-1	ON
SW1-2	OFF

SW1-1	DEBUGEN
OFF	JTAG debug mode
ON	Normal operation

SW1-2	microSD/eMMC selection
OFF	Select eMMC on RTK9744L23C01000BE
ON	Select microSD slot on RTK9744L23C01000BE

The selection of microSD slot and eMMC on the SMARC module is exclusive

Renesas SMARC RZ/G2UL Evaluation kit (smarc-rzg2ul) and RZ/Five Evaluation kit (smarc-rzfive)

Please set the SW1 settings to follows.

- The SW1-1 is used to select the JTAG debug mode or not. JTAG is not used to in the procedure for booting up RZ/G2UL and RZ/Five Group Board Support Package, so please set SW1-1 to normal operation mode.
- The SW1-2 is used to select the eMMC or microSD mode. Please set SW1-2 to eMMC mode.
- The SW1-3 is used to select the Ethernet or multiple functions mode. The multiple functions are enabled CAN0, CAN1, PMOD0 and PMOD1. Please set SW1-3 to other function mode.



SW1-3	OFF
SW1-2	OFF
SW1-1	ON

SW1-1	DEBUGEN
OFF	JTAG debug mode
ON	Normal operation

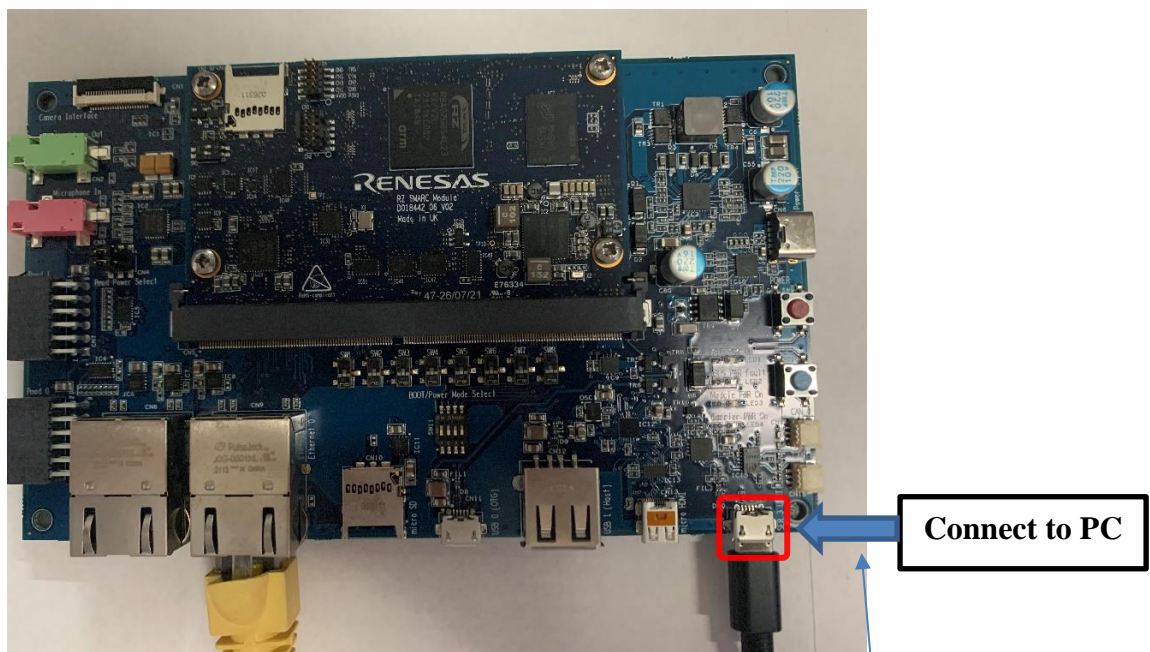
SW1-2	SD/MMC selection
OFF	Select eMMC memory
ON	Select microSD card

The selection of microSD slot and eMMC on the SMARC module is exclusive

SW1-3	Ether0/CAN0, CAN1, SSI1, RSPI1 selection
OFF	CAN0, CAN1, PMOD0 Type-2A, PMOD Type-6
ON	Ether0

1.1.4 How to use debug serial (console output)

Please connect USB Type-microAB cable to CN14.



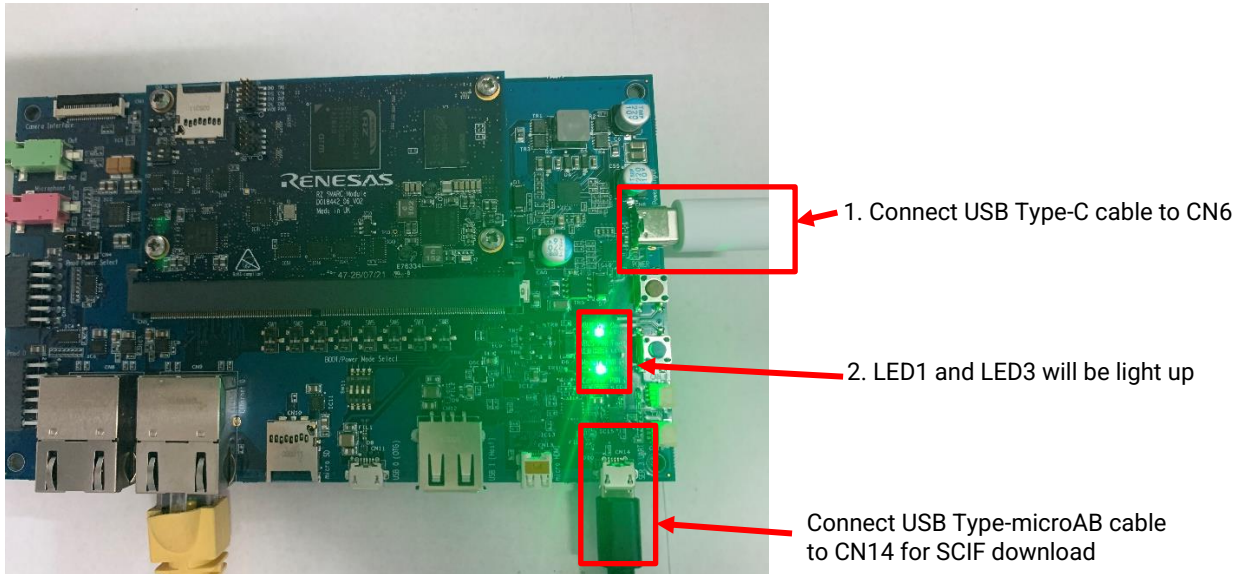
CN14:USB Type-microAB Connector

1.2 Startup Procedure

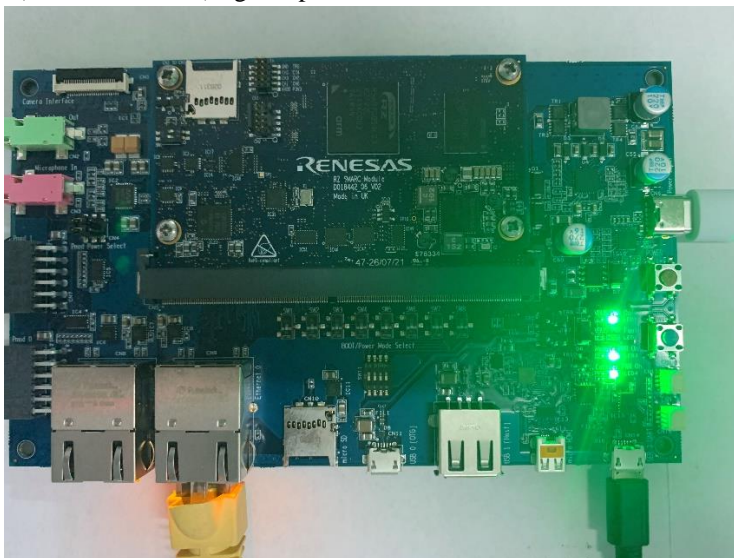
Hereafter, RZ/V2L Evaluation Board Kit PMIC version picture is used as a representative. If you use RZ/G2L, RZ/G2LC, RZ/G2UL, and RZ/Five Evaluation Board Kit PMIC version, you can use Connectors on same location as RZ/V2L Evaluation Board Kit PMIC version.

1.2.1 Power supply

1. Connect USB-PD Power Charger to USB Type-C Connector (CN6).
2. LED1(VBUS Power ON) and LED3 (Module PWR On) lights up.



3. Press the power button(SW9) to turn on the power.
Note: When turn on the power, press and hold the power button for 1 second.
When turn off the power, press and hold the power button for 2 seconds
4. LED4(Carrier PWR On) lights up.



1.2.2 Building files to write

The evaluation boards use the files in the Table 2 as the boot loaders. Please build them according to the Release Note and copy these files to a PC which runs serial terminal software.

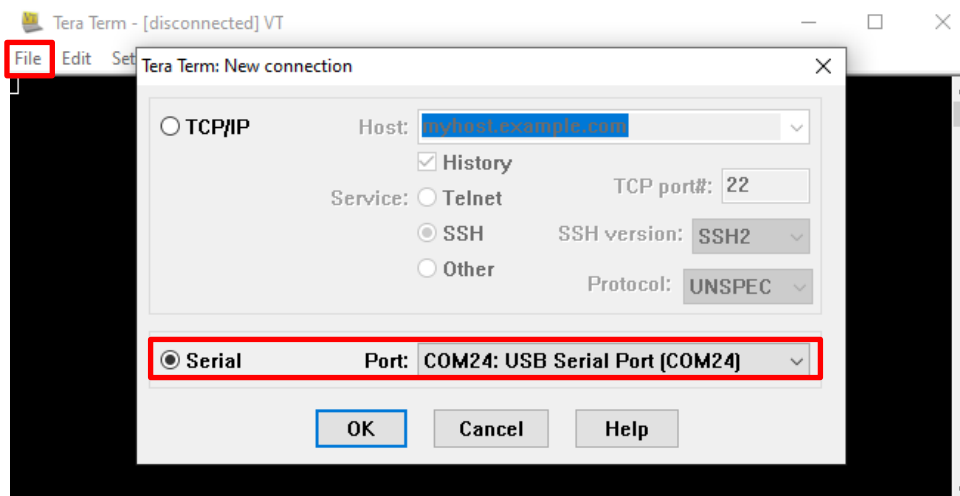
Table 2. File names of Boot loader

Board	File name of Boot loader
RZ/G2L Evaluation Board Kit PMIC version	<ul style="list-style-type: none">• bl2_bp-smarc-rzg2l_pmic.srec• fip-smarc-rzg2l_pmic.srec
RZ/G2LC Evaluation Board Kit	<ul style="list-style-type: none">• bl2_bp-smarc-rzg2lc.srec• fip-smarc-rzg2lc.srec
RZ/G2UL Evaluation Board Kit	<ul style="list-style-type: none">• bl2_bp-smarc-rzg2ul.srec• fip-smarc-rzg2ul.srec
RZ/V2L Evaluation Board Kit PMIC version	<ul style="list-style-type: none">• bl2_bp-smarc-rzv2l_pmic.srec• fip-smarc-rzv2l_pmic.srec
RZ/Five Evaluation Board Kit	<ul style="list-style-type: none">• fit-smarc-rzfive.srec• spl-smarc-rzfive.srec

1.2.3 Settings

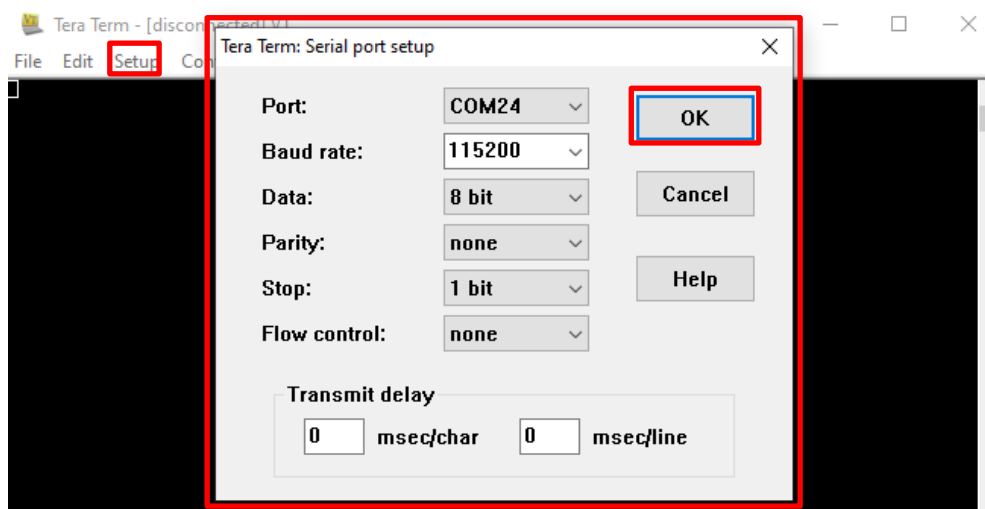
Connect between the board and a control PC by USB serial cable according to the Release Note.

1. Bring up the terminal software and select the “File” > “New Connection” to set the connection on the software.



2. Select the “Setup” > “Serial port” to set the settings about serial communication protocol on the software. Set the settings about serial communication protocol on a terminal software as below:

- Speed: 115200 bps
- Data: 8bit
- Parity: None
- Stop bit: 1bit
- Flow control: None



3. To set the board to SCIF Download mode, set the SW11 as below (please refer 2.1.2):



SW11

1	2	3	4
OFF	ON	OFF	ON

4. After finished all settings, when pressed the reset button SW10, the messages below are displayed on the terminal.



3. Press the Blue button

1.3 Booting Flash Writer

Turn on the power of the board by pressing SW9. The messages below are shown on the terminal.

```

SCIF Download mode
(C) Renesas Electronics Corp.

-- Load Program to SystemRAM -----
please send !
    
```

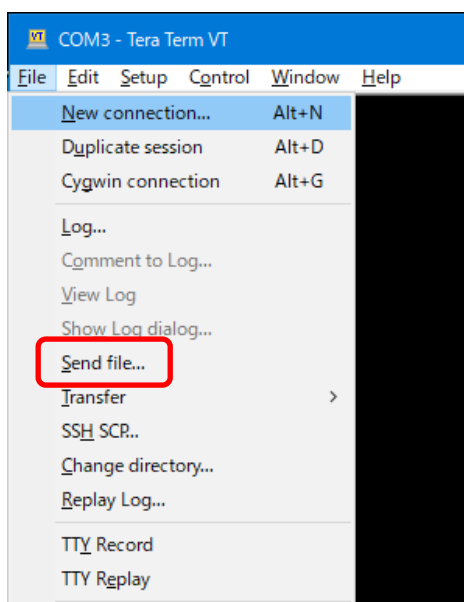
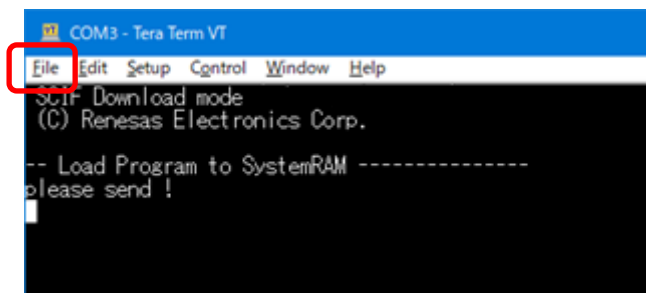
Send an image of Flash Writer using terminal software after the message “please send !” is shown. Please refer to the Table 3 below to know which file name of Flash Writer should be sent.

Table 3. File names of Flash Writer

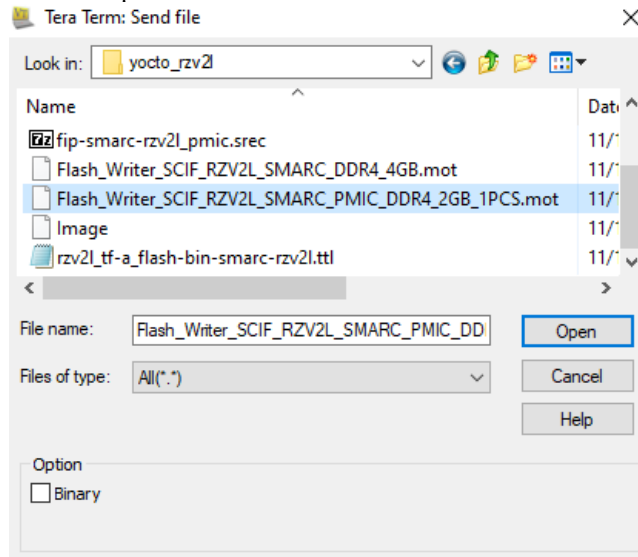
Board name	File name of Flash Writer
RZ/G2L Evaluation Board Kit PMIC version	Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
RZ/G2LC Evaluation Board Kit	Flash_Writer_SCIF_RZG2LC_SMARC_DDR4_1GB_1PCS.mot
RZ/G2UL Evaluation Board Kit	Flash_Writer_SCIF_RZG2UL_SMARC_DDR4_1GB_1PCS.mot
RZ/V2L Evaluation Board Kit PMIC version	Flash_Writer_SCIF_RZV2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
RZ/Five Evaluation Board Kit	Flash_Writer_SCIF_RZFIVE_SMARC.mot

Below is a sample procedure with Tera Term.

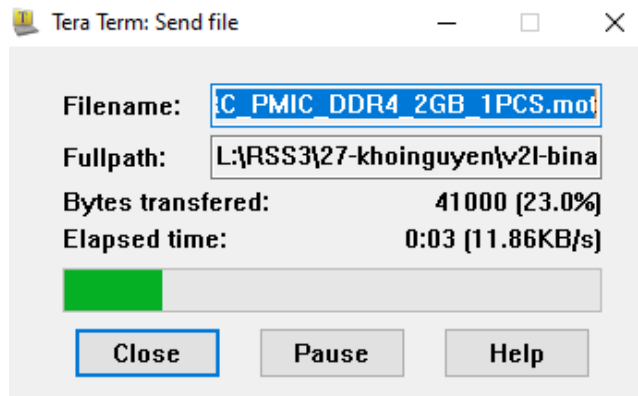
Open a “Send file” dialog by selecting “File” → “Sendfile” menu.



Then, select the image to be send and click “Open” button.



The image will be sent to the board via serial connection.



After successfully downloading the binary, Flash Writer starts automatically and shows a message like below on the terminal.

```
Flash writer for RZ/V2 Series V1.02 Nov.15,2021
Product Code : RZ/V2L
>
```

1.4 Writing Bootloader

For the boot operation, two boot loader files need to be written to the target board. Corresponding bootloader files and specified address information are depending on each target board as described in **Table 4**.

“XLS2” command of Flash Writer is used to write boot loader binary files. This command receives binary data from the serial port and writes the data to a specified address of the Flash ROM with information where the data should be loaded on the address of the main memory.

For example, this part describes how to write boot loader files in the case of RZ/V2L Evaluation Board Kit PMIC version.:

```
>XLS2
===== Qspi writing of RZ/G2 Board Command =====
Load Program to Spiflash
Writes to any of SPI address.
  Micron : MT25QU512
Program Top Address & Qspi Save Address
===== Please Input Program Top Address =====
  Please Input : H'11E00

===== Please Input Qspi Save Address ===
  Please Input : H'00000
Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ( '.' & CR stop load)
```

Send the data of “bl2_bp-smarc-rzv2l_pmic.srec” from terminal software after the message “please send !” is shown.

After successfully download the binary, messages like below are shown on the terminal.

```
SPI Data Clear(H'FF) Check :H'00000000-0000FFFF Erasing..Erase Completed
SAVE SPI-FLASH.....
===== Qspi Save Information =====
  SpiFlashMemory Stat Address : H'00000000
  SpiFlashMemory End Address : H'00009A80
=====
```

```
SPI Data Clear(H'FF) Check : H'00000000-0000FFFF,Clear OK?(y/n)
```

In case a message to prompt to clear data like above, please enter “y”.

Next, write another loader file by using XLS2 command again.

```
>XLS2
===== Qspi writing of RZ/G2 Board Command =====
Load Program to Spiflash
Writes to any of SPI address.
  Micron : MT25QU512
Program Top Address & Qspi Save Address
===== Please Input Program Top Address =====
  Please Input : H'00000

===== Please Input Qspi Save Address ===
  Please Input : H'1D200
Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ( '.' & CR stop load)
```

Send the data of “fip-smarc-rzv2l_pmic.srec” from terminal software after the message “please send !” is shown.

After successfully download the binary, messages like below are shown on the terminal.

```
SPI Data Clear(H'FF) Check :H'00000000-0000FFFF Erasing..Erase Completed
SAVE SPI-FLASH.....
===== Qspi Save Information =====
SpiFlashMemory Stat Address : H'0001D200
SpiFlashMemory End Address : H'000CC73F
=====
```

```
SPI Data Clear(H'FF) Check : H'00000000-0000FFFF,Clear OK?(y/n)
```

In case a message to prompt to clear data like above, please enter “y”.

After writing two loader files normally, turn off the power of the board by changing the SW11.

Table 4. Address for sending each loader binary file

RZ/G2L Evaluation Board Kit PMIC version

File name	Address to load to RAM	Address to save to ROM
bl2_bp-smarc-rzg2l_pmic.srec	11E00	00000
fip-smarc-rzg2l_pmic.srec	00000	1D200

RZ/G2LC Evaluation Board Kit

File name	Address to load to RAM	Address to save to ROM
bl2_bp-smarc-rzg2lc.srec	11E00	00000
fip-smarc-rzg2lc.srec	00000	1D200

RZ/G2UL Evaluation Board Kit

File name	Address to load to RAM	Address to save to ROM
bl2_bp-smarc-rzg2ul.srec	11E00	00000
fip-smarc-rzg2ul.srec	00000	1D200

RZ/V2L Evaluation Board Kit PMIC version

File name	Address to load to RAM	Address to save to ROM
bl2_bp-smarc-rzv2l_pmic.srec	11E00	00000
fip-smarc-rzv2l_pmic.srec	00000	1D200

RZ/Five Evaluation Board Kit

File name	Address to load to RAM	Address to save to ROM
spl-smarc-rzfive.srec	11E00	00000
fit-smarc-rzfive.srec	00000	20000

1.5 Create a microSD card to boot Linux

To boot from SD card, over 4GB capacity of blank SD card is needed. You can use Linux Host PC to expand the kernel and the rootfs using USB card reader or other equipment.

Please format the card according to the following steps before using the card:

(1) Non-connect microSD card to Linux Host PC

```
$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0  30.9G  0 disk
├─sda1 8:1    0   512M  0 part /boot/efi
├─sda2 8:2    0     1K  0 part
└─sda5 8:5    0  30.3G  0 part /
sr0   11:0   1  1024M  0 rom
```

(2) Connect microSD card to Linux Host PC with USB adapter

(3) Check the device name which is associated to the microSD card.

```
$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0  30.9G  0 disk
├─sda1 8:1    0   512M  0 part /boot/efi
├─sda2 8:2    0     1K  0 part
└─sda5 8:5    0  30.3G  0 part /
sdb   8:16   1  29.7G  0 disk
└─sdb1 8:17   1  29.7G  0 part
sr0   11:0   1  1024M  0 rom
```

The message above shows the card associated with the /dev/sdb. **Be careful not to use the other device names in the following steps.**

(4) Unmount automatically mounted microSD card partitions

If necessary, unmount all mounted microSD card partitions.

```
$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            745652          0    745652   0% /dev
:
snip
:
/dev/sdb1       511720        4904    506816   1% /media/user/A8D3-393B
$ sudo umount /media/user/A8D3-393B
```

If more than one partition has already been created on microSD card, unmount all partitions.

(5) Change the partition table

microSD card needs two partitions as listed in Table 5.

Table 5. Partitions of microSD card

Type/Number	Size	Filesystem	Contents
Primary #1	500MB (minimum 128MB)	FAT32	Linux kernel Device tree
Primary #2	All remaining	Ext4	root filesystem

Set the partition table using the fdisk command like this.

```
$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): o

Created a new DOS disklabel with disk identifier 0x6b6aac6e.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-62333951, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62333951, default 62333951): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): (Push the enter key)
First sector (1026048-62333951, default 1026048): (Push the enter key)
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1026048-62333951, default 62333951): (Push the enter key)

Created a new partition 2 of type 'Linux' and of size 29.2 GiB.

Command (m for help): p
Disk /dev/sdb: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Disk model: Transcend
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```

Disklabel type: dos
Disk identifier: 0x6b6aac6e

Device      Boot   Start       End  Sectors  Size Id Type
/dev/sdb1           2048   1026047   1024000   500M 83 Linux
/dev/sdb2       1026048 62333951 61307904 29.2G 83 Linux

Filesystem/RAID signature on partition 1 will be wiped.

Command (m for help): t
Partition number (1,2, default 2): 1
Hex code (type L to list all codes): b

Changed type of partition 'Linux' to 'W95 FAT32'.

Command (m for help): w
The partition table has been altered.
Syncing disks.

```

Then, check the partition table with the commands below:

```

$ partprobe
$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 29.74 GiB, 31914983424 bytes, 62333952 sectors
Disk model: Maker name etc.
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6b6aac6e

Device      Boot   Start       End  Sectors  Size Id Type
/dev/sdb1           2048   1026047   1024000   500M  b W95 FAT32
/dev/sdb2       1026048 62333951 61307904 29.2G  83 Linux

```

(6) Format and mount the partitions

If the partitions were automatically mounted after the step 4, please unmount them according to the step 3.

Then format the partitions using the command below:

```

$ sudo mkfs.vfat -v -c -F 32 /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
/dev/sdb1 has 64 heads and 32 sectors per track,
hidden sectors 0x0800;
logical sector size is 512,
using 0xf8 media descriptor, with 1024000 sectors;
drive number 0x80;
filesystem has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 1000 sectors, and provides 127746 clusters.
There are 32 reserved sectors.
Volume ID is a299e6a6, no volume label.
Searching for bad blocks 16848... 34256... 51152... 68304... 85072... 10209
6... 119376... 136528... 153552... 170576... 187472... 204624... 221648... 238
928... 256208... 273744... 290768... 308048... 325328... 342480... 359504... 3
76656... 393680... 410576... 427216... 444624... 462032... 479184... 495952...

$ sudo mkfs.ext4 -L rootfs /dev/sdb2

```

```
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 7663488 4k blocks and 1916928 inodes
Filesystem UUID: 63dddb3f-e268-4554-af51-1c6e1928d76c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

(7) Remount microSD card

After format, **remove the card reader and connect it again** to mount the partitions.

(8) Write files to the microSD card

Check the mount point name with df command.

```
$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            745652          0    745652   0% /dev
:
:
:
snip
:
/dev/sdb1        510984          16    510968   1% /media/user/A299-E6A6
/dev/sdb2       30041556       45080  28447396  1% /media/user/rootfs
```

Copy kernel and device tree file to the first partition.

```
$ cp $WORK/build/tmp/deploy/images/<board>/<Linux kernel> /media/user/A299-E6A6
6
$ cp $WORK/build/tmp/deploy/images/<board>/<Devise tree> /media/user/A299-E6A6
```

Expand rootfs to the second partition.

```
$ cd /media/user/rootfs
$ sudo tar jxvf $WORK/build/tmp/deploy/images/<board>/<root filesystem>
```

Above samples are suitable only for **RZ/G2L group** and **RZ/V2L**.

Please replace **<board>** by the name below:

RZ/G2L Evaluation Board Kit:	smarc-rzg2l
RZ/G2LC Evaluation Board Kit:	smarc-rzg2lc
RZ/G2UL Evaluation Board Kit:	smarc-rzg2ul
RZ/V2L Evaluation Board Kit:	smarc-rzv2l

When you use RZ/Five Evaluation board Kit, copy kernel and device tree file to the first partition like as below.

```
$ cp $WORK/build/tmp-glibc/deploy/images/smarc-rzfive/<Linux kernel> /media/user/A299-E6A6
er/A299-E6A6
$ cp $WORK/build/tmp-glibc/deploy/images/smarc-rzfive/<Devise tree> /media/user/A299-E6A6
```

When you use RZ/Five Evaluation board Kit, expand rootfs to the second partition like as below.

```
$ cd /media/user/rootfs
$ sudo tar jxvf $WORK/build/tmp-glibc/deploy/images/smarc-rzfive/<root filesystem>
tem>
```

The file names of *<Linux kernel>*, *<Device tree>*, and *<root filesystem>* are listed in the Table 6.

Table 6. Image files for RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five

RZ/G2L	Linux kernel	Image-smarc-rzg2l.bin
	Device tree file	Image-r9a07g044l2-smarc.dtb
	root filesystem	<i><image name></i> -smarc-rzg2l.tar.bz2
RZ/G2LC	Linux kernel	Image-smarc-rzg2lc.bin
	Device tree file	Image-r9a07g044c2-smarc.dtb
	root filesystem	<i><image name></i> -smarc-rzg2lc.tar.bz2
RZ/G2UL	Linux kernel	Image-smarc-rzg2ul.bin
	Device tree file	Image-r9a07g043u11-smarc.dtb
	root filesystem	<i><image name></i> -smarc-rzg2ul.tar.bz2 (*)
RZ/V2L	Linux kernel	Image-smarc-rzv2l.bin
	Device tree file	Image-r9a07g054l2-smarc.dtb
	root filesystem	<i><image name></i> -smarc-rzv2l.tar.bz2
RZ/Five	Linux kernel	Image-smarc-rzfive.bin
	Device tree file	Image-r9a07g043f01-smarc.dtb
	root filesystem	<i><image name></i> -smarc-rzfive.tar.bz2 (*)

Please replace *<image name>* by the name below table.

Table 7. Supported images

Image name	Description
core-image-minimal	Image which includes minimal set of components
core-image-bsp	Image which includes minimal set of components, audio components, and some useful tools
core-image-weston	Standard image which includes the graphics support
core-image-qt	Image which includes the graphics library and Qt LGPL version

(*) RZ/G2UL and RZ/Five support only core-image-minimal and core-image-bsp.

1.6 Setting U-boot

To set the board to SPI Boot mode, set the SW11 as below:

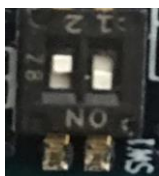


SW11

1	2	3	4
OFF	OFF	OFF	ON

Note:-

Set the SW1 on SoM module to eMMC mode. Please refer to the section 1.1.3.



Turn on the power of the board by pressing the reset button SW10.

```
U-Boot 2021.10 (Mar 31 2022 - 03:57:20 +0000)

CPU:   Renesas Electronics K rev 14.15
Model: smarc-rzg2l
DRAM:  1.9 GiB
MMC:   sd@11c00000: 0, sd@11c10000: 1
Loading Environment from MMC... OK
In:    serial@1004b800
Out:   serial@1004b800
Err:   serial@1004b800
Net:   eth0: ethernet@11c20000
Hit any key to stop autoboot:  0
=>
```

Following the messages above, many warning messages will be shown. These warnings are eliminated by setting correct environment variables. Please set default value and save them to the Flash ROM.

```
=> env default -a
## Resetting to default environment
=> saveenv
Saving Environment to MMC... Writing to MMC(0)...OK
=>
```

In case booting from micro SD card on SMARC carrier board, set environment variables using the commands below. The commands below are for the RZ/V2L board. Please replace the file names in “bootcmd” according to the Release Note when you use other boards.

```
=> setenv bootargs 'root=/dev/mmcblk1p2 rootwait'
=> setenv bootcmd 'mmc dev 1;fatload mmc 1:1 0x48080000 Image-smarc-rzv2l.bin;
fatload mmc 1:1 0x48000000 Image-r9a07g05412-smarc.dtb; booti 0x48080000 - 0x
48000000'
=> saveenv
Saving Environment to MMC... Writing to MMC(0)...OK
```

bootargs: 'root=/dev/mmcblk1p2 rootwait'

root filesystem is partition 2 of block 1 on microSD card.

bootcmd: 'mmc dev 1;fatload mmc 1:1 0x48080000 <Linux kernel>;

fatload mmc 1:1 0x48000000 <Device tree>;

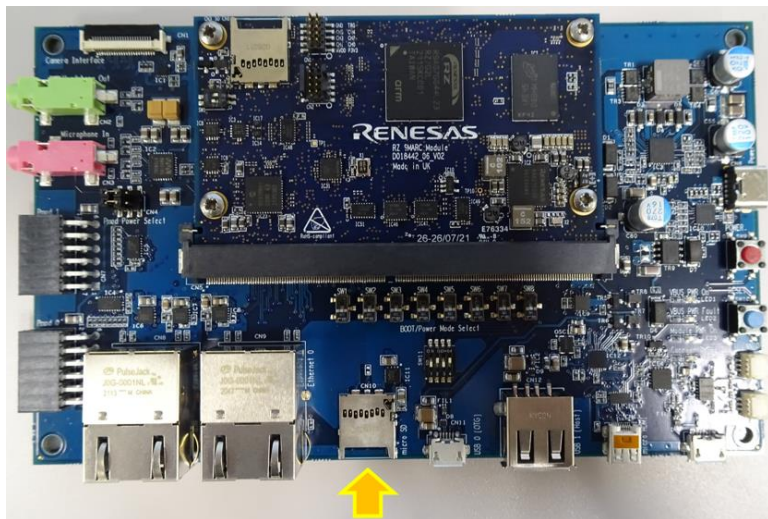
booti 0x48080000 - 0x48000000'

Note) The setting above assumes the SD card has two partitions and stores data as below:

First partition: formatted as FAT, includes Image-smarc-rzv2l.bin and Image-r9a07g05412-smarc.dtb

Second partition: formatted as ext4, rootfs image is expanded

Set microSD card to slot on carry board.



Now the board can bootup normally. Please turn off and on the power again to boot up the board.

After obtaining your reference board, please be sure to follow the document and write the bootloaders to the Flash ROM before starting the evaluation.

Before booting the board, please be sure to confirm the bootloaders which are built with your BSP/VLP are written to your board.

```
U-Boot 2021.10 (Mar 31 2022 - 03:57:20 +0000)

CPU:   Renesas Electronics K rev 14.15
Model: smarc-rzg2l
DRAM:  1.9 GiB
MMC:   sd@11c00000: 0, sd@11c10000: 1
```

```
Loading Environment from MMC... OK
In:    serial@1004b800
Out:   serial@1004b800
Err:   serial@1004b800
Net:   eth0: ethernet@11c20000
Hit any key to stop autoboot:  0
switch to partitions #0, OK
mmc1 is current device
16947712 bytes read in 1054 ms (15.3 MiB/s)
37530 bytes read in 5 ms (7.2 MiB/s)
Moving Image from 0x48080000 to 0x48200000, end=49290000
## Flattened Device Tree blob at 48000000
   Booting using the fdt blob at 0x48000000
   Loading Device Tree to 0000000057fff3000, end 0000000057fff299 ... OK

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x412fd050]
[    0.000000] Linux version 5.10.83-cip1-yocto-standard (oe-user@oe-host) (aa
rc
:
:
Poky (Yocto Project Reference Distro) 3.1.14 smarc-rzg2l ttySC0

BSP: RZG2L/RZG2L-SMARC-EVK/3.0.0
LSI: RZG2L
Version: 3.0.0
smarc-rzg2l login: root
Last login: Sun Sep 20 10:44:25 UTC 2020
root@smarc-rzg2l:~#
```

2. Appendix

2.1 Preparing Flash Writer

Flash Writer is built automatically when building BSP by bitbake command. Please refer to the Release Note of the RZ/G2L BSP Group, the RZ/V2L BSP, and the RZ/Five BSP to obtain a binary file of Flash Writer.

If you need latest one, please get source code from the GitHub repository and build it according to the following instructions. In general, new revision of reference boards requires latest Flash Writer.

2.1.1 Preparing cross compiler

For SMARC Evaluation Board kit of RZ/G2L, RZ/G2LC, RZ/G2UL, and RZ/V2L

FlashWriter runs on target boards. Please get cross compiler built by Linaro or setup a Yocto SDK.

- **ARM toolchain:**

```
$ cd ~/
$ wget https://developer.arm.com/-/media/Files/downloads/gnu-a/10.2-2020.11/binrel/gcc-arm-10.2-2020.11-x86_64-aarch64-none-elf.tar.xz
$ tar xvf gcc-arm-10.2-2020.11-x86_64-aarch64-none-elf.tar.xz
```

- **Yocto SDK:**

Build an SDK according to Release Notes and install it to a Linux Host PC. Then, enable the SDK as below.

```
$ source /opt/poky/3.1.14/environment-setup-aarch64-poky-linux
```

For SMARC Evaluation Board kit of RZ/Five

FlashWriter runs on target boards. Please get cross compiler built by GNU toolchain or setup a Yocto SDK.

- **GNU toolchain:**

```
$ git clone https://github.com/riscv/riscv-gnu-toolchain
$ cd riscv-gnu-toolchain
$ ./configure --prefix=/opt/riscv
$ sudo make linux -j
```

- **Yocto SDK:**

Build an SDK according to Release Notes and install it to a Linux Host PC. Then, enable the SDK as below.

```
$ source /usr/local/oe-core-x86_64/environment-setup-riscv64-oe-linux
```


2.1.2 Building Flash Writer

For SMARC Evaluation Board kit of RZ/G2L, RZ/G2LC, RZ/G2UL, and RZ/V2L

Get source codes of Flash Writer from the GitHub repository and checkout the branch rz_g2l.

```
$ cd ~/
$ git clone https://github.com/renesas-rz/rzg2_flash_writer.git
$ cd rzg2_flash_writer
$ git checkout -b tmp 4168466783f06fa7f2aa5782c597803a6882ed2f
```

Build Flash Writer as an s-record file by the following commands. Please specify a target board by “BOARD” option.

- **ARM toolchain:**

```
$ export PATH=$PATH:~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-elf/bin
$ export CROSS_COMPILE=aarch64-none-elf-
$ export CC=${CROSS_COMPILE}gcc
$ export AS=${CROSS_COMPILE}as
$ export LD=${CROSS_COMPILE}ld
$ export AR=${CROSS_COMPILE}ar
$ export OBJDUMP=${CROSS_COMPILE}objdump
$ export OBJCOPY=${CROSS_COMPILE}objcopy

$ make clean
$ make BOARD=<board>
```

- **Yocto SDK:**

```
$ make clean
$ make BOARD=<board>
```

Please replace <board> to a proper option according to this table.

Table 8. BOARD option

Target board	BOARD option <board>	Image to be generated
smarc-rzg2l-pmic	RZG2L_SMARC_PMIC	Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
smarc-rzg2lc	RZG2LC_SMARC	Flash_Writer_SCIF_RZG2LC_SMARC_DDR4_1GB_1PCS.mot
smarc-rzg2ul	RZG2UL_SMARC	Flash_Writer_SCIF_RZG2UL_SMARC_DDR4_1GB_1PCS.mot
smarc-rzv2l-pmic	RZV2L_SMARC_PMIC	Flash_Writer_SCIF_RZV2L_SMARC_PMIC_DDR4_2GB_1PCS.mot

For SMARC Evaluation Board kit of RZ/Five

Get source codes of Flash Writer from the GitHub repository and checkout the branch rz_g2l.

```
$ cd ~/
$ git clone https://github.com/renesas-rz/rzg2_flash_writer.git
$ cd rzg2_flash_writer
$ git checkout rz_five
```

Build Flash Writer as an s-record file by the following commands. Please specify a target board by “BOARD” option.

```
$ export PATH=$PATH:/opt/riscv/bin
$ make clean
$ make BOARD=RZFIVE_SMARC
```

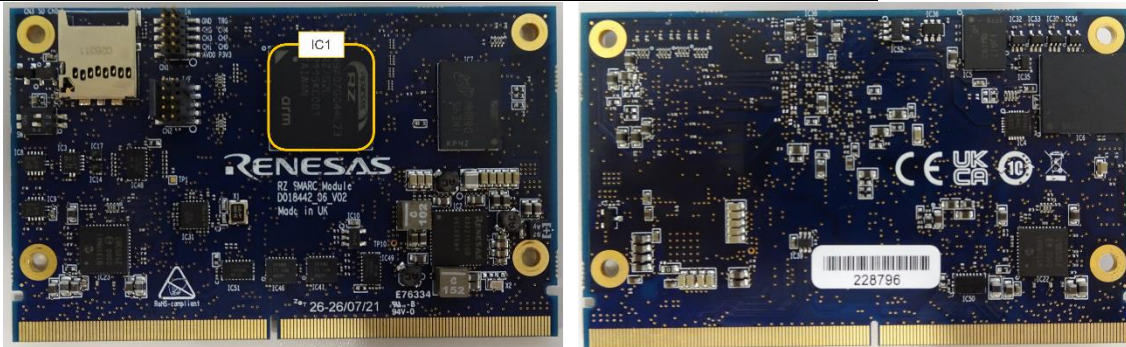
After above steps, “Flash_Writer_SCIF_RZFIVE_SMARC.mot” is generated.

2.2 How to distinguish each board

The differences between each board are as follows.

Comparison of RZ/G2L Evaluation Board Kit and RZ/V2L Evaluation Board Kit:

	RZ/G2L Evaluation Kit	RZ/V2L Evaluation Kit
IC1	RZ/G2L (R9A07G044L23GBG)	RZ/V2L (R9A07G054L23GBG)



Comparison of RZ/G2L Evaluation Board Kit and RZ/G2LC Evaluation Board Kit:

	RZ/G2L Evaluation Kit	RZ/G2LC Evaluation Kit
IC1	RZ/G2L (R9A07G044L23GBG)	RZ/G2LC (R9A07G054GBG)
IC7	MT40A1G16KD-062E:E (2GB)	MT40A512M16LY-062EIT:E (1GB)
IC23	Ethernet PHY (KSZ9131RNXC)	Not mounted
CN1	10-pin connector for ADC	Not mounted
SW1	2bit bus switch	6bit bus switch



Comparison of RZ/G2UL Evaluation Board Kit and RZ/Five Evaluation Board Kit:

	RZ/G2UL Evaluation Kit	RZ/Five Evaluation Kit
IC1	RZ/G2UL (R9A07G043U11GBG)	RZ/Five (R9A07G043F01GBG)



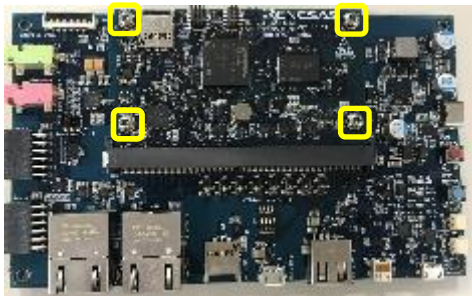
2.3 How to replace the SMARC Module Board

Please be careful when replacing the board as follows.

1. Remove the four screws.

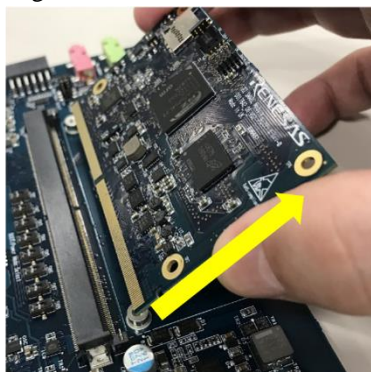
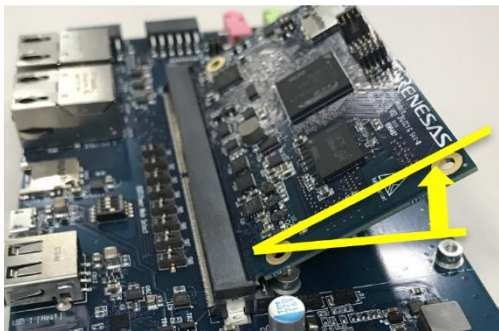
Note: The screw thread is a special shape, so be careful not to crush the screw thread.

Please recommend to prepare a torx screwdriver which is a "T6" head size.

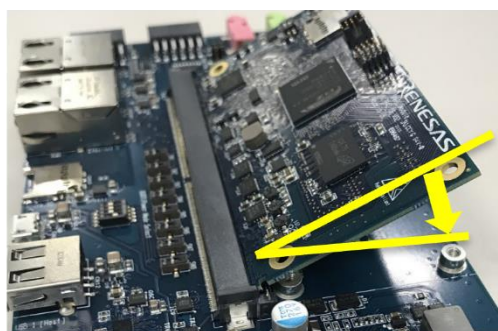
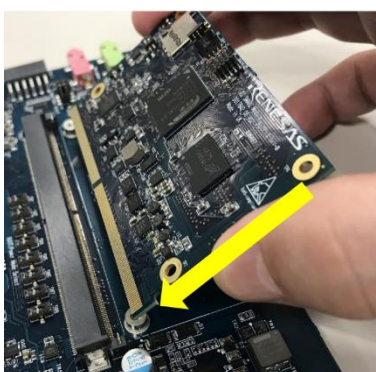


Specially shaped screw threads

2. Remove the screw and the board will stand up at an angle. Slide it out.

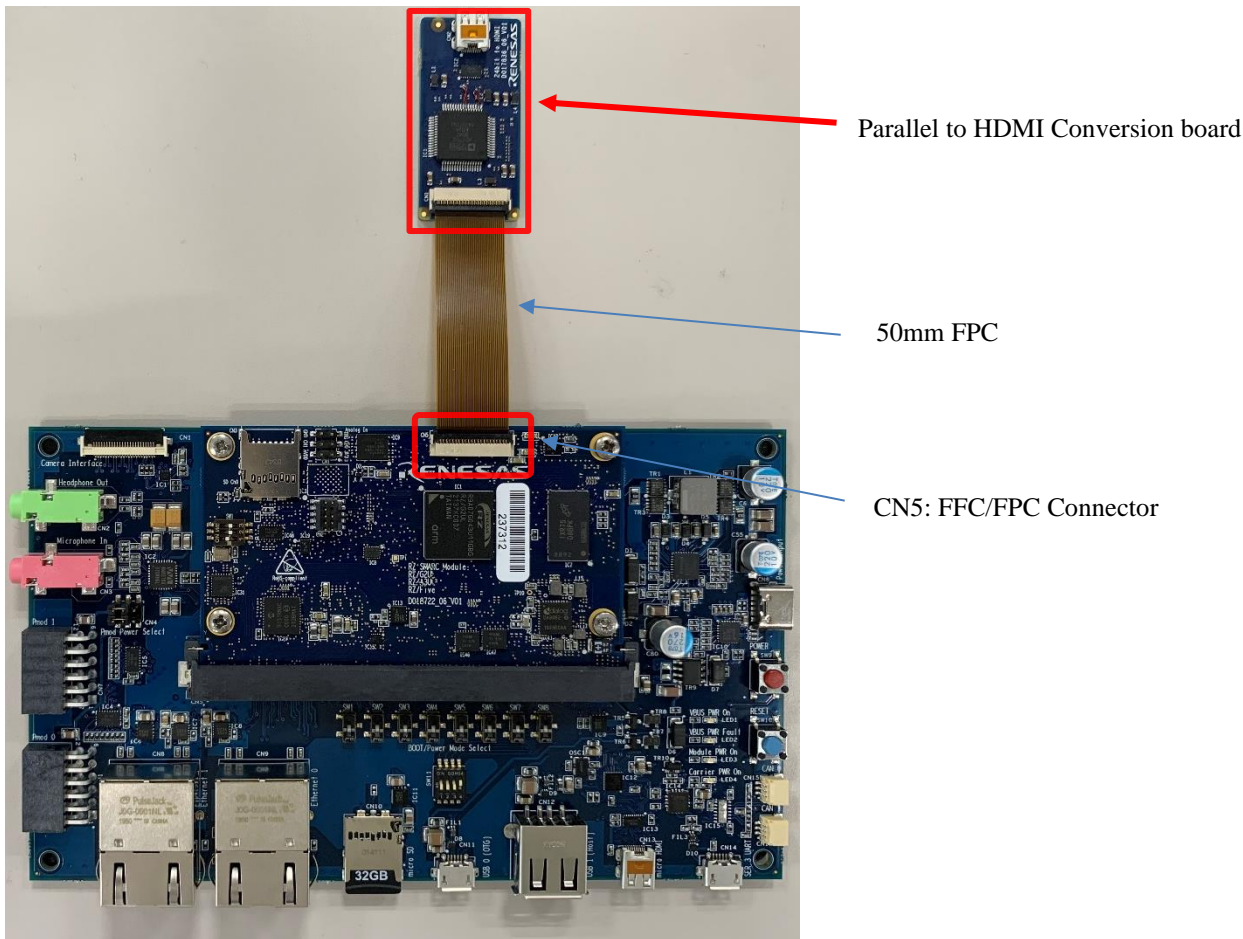


3. Insert the replacement the board diagonally, then roll the SMARC board parallel to the board and fix it with screws.



2.4 How to connect the Parallel to HDMI Conversion board for RZ/G2UL Evaluation kit (smarc-rzg2ul)

Please connect CN1 of the Parallel to HDMI Conversion board to CN5 via the 50mm FPC (228-000071-01).



A termination procedure of the 50mm FPC is follows.

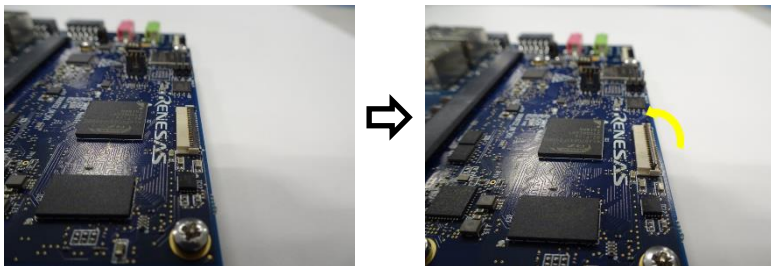
The top and bottom surfaces of the FPC are shown below.



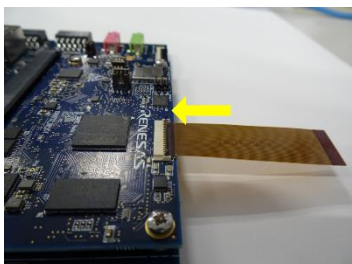
Top surface

Bottom surface

1. Lift up the actuator. Use thumb or index finger.



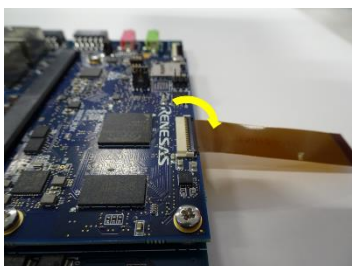
2. Fully insert the FPC in the connector parallel to mounting surface, with the exposed conductive traces facing down



3. Rotate down the actuator until firmly closed.

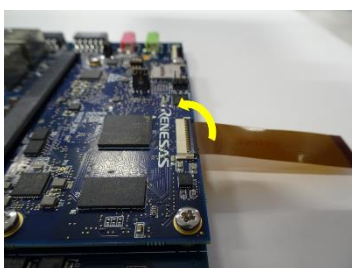
Note: The PFC must be fully inserted in the connector. If not fully inserted, the actuator will not close properly.

Should this be the case, lift up the actuator and repeat the process (starting with Step 1 above)



4. FPC Removal

- 1) Lift up the actuator.
- 2) Carefully remove the FPC



2.5 How to boot from eMMC

In this section, the steps to boot from eMMC on RZ/G2L, RZ/G2LC, RZ/G2UL, and RZ/V2L is described.

Please note that RZ/Five doesn't support the eMMC boot now. When you require it, please contact Renesas.

2.5.1 Rebuild rootfs

Build a rootfs according to Release Note. After that, please rebuild the rootfs with the below commands.

```
$ cd ${your build directory} (ex.$ cd ~/rzv_v1p_v3.0.0)
$ source poky/oe-init-build-env
$ echo 'IMAGE_INSTALL_append = " e2fsprogs-mke2fs"'>> conf/local.conf
$ bitbake core-image-xxxxxx (ex. core-image-weston)
```

2.5.2 Writing Bootloader for eMMC Boot

For the boot operation, EXT_CSD register of eMMC need to be modified and two boot loader files need to be written to the target board. Modifying register address and value information are described in **Table 8**, and corresponding bootloader files and specified address information are depending on each target board as described in **Table 9**.

After booting the Flash Writer (Refer to [1.3 Booting Flash Writer](#)), “EM_SECS” command of Flash Writer is used to modify EXT_CSD register of eMMC to enable eMMC boot.

Then, “EM_W” command of Flash Writer is used to write boot loader binary files. This command receives binary data from the serial port and writes the data to a specified address of the eMMC with information where the data should be loaded on the address of the main memory.

For example, this part describes how to modify EXT_CSD register and write boot loader files in the case of RZ/V2L Evaluation Board Kit PMIC version.:

```
>EM_SECS
Please Input EXT_CSD Index(H'00 - H'1FF) :b1
EXT_CSD[B1] = 0x00
Please Input Value(H'00 - H'FF) :2
EXT_CSD[B1] = 0x02
>EM_SECS
Please Input EXT_CSD Index(H'00 - H'1FF) :b3
EXT_CSD[B3] = 0x00
Please Input Value(H'00 - H'FF) :8
EXT_CSD[B3] = 0x08
```

```
>EM_W
EM_W Start -----
-----
Please select,eMMC Partition Area.
0:User Partition Area   : 62160896 KBytes
  eMMC Sector Cnt : H'0 - H'0768FFFF
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)>1
-- Boot Partition 1 Program -----
Please Input Start Address in sector :1
Please Input Program Start Address : 11e00
Work RAM(H'50000000-H'50FFFFFF) Clear....
please send ! ( '.' & CR stop load)
```

Send the data of “bl2_bp-smarc-rzv2l_pmic.srec” from terminal software after the message “please send !” is shown.

After successfully download the binary, messages like below are shown on the terminal.

```
SAVE -FLASH.....
EM_W Complete!
```

Next, write another loader file by using EM_W command again.

```
> EM_W
EM_W Start -----
-----
Please select,eMMC Partition Area.
0:User Partition Area   : 62160896 KBytes
  eMMC Sector Cnt : H'0 - H'0768FFFF
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)>1
-- Boot Partition 1 Program -----
Please Input Start Address in sector :100
Please Input Program Start Address : 0
Work RAM(H'50000000-H'50FFFFFF) Clear....
please send ! ( '.' & CR stop load)
```

Send the data of “fip-smarc-rzv2l_pmic.srec” from terminal software after the message “please send !” is shown.

After successfully download the binary, messages like below are shown on the terminal.

```
SAVE -FLASH.....
EM_W Complete!
```

After writing two loader files normally, turn off the power of the board by changing the SW11.

Table 8. Address of EXT_CSD register of eMMC for eMMC boot

Address	Value to write
0xB1	0x02
0xB3	0x08

Table 9. Address for sending each loader binary file for eMMC boot

RZ/G2L Evaluation Board Kit PMIC version

File name	Partition to save to eMMC	Address to save to eMMC	Address to load to RAM
bl2_bp-smarc-rzg2l_pmic.srec	1	00000001	11E00
fip-smarc-rzg2l_pmic.srec	1	00000100	00000

RZ/G2LC Evaluation Board Kit

File name	Partition to save to eMMC	Address to save to eMMC	Address to load to RAM
bl2_bp-smarc-rzg2lc.srec	1	00000001	11E00
fip-smarc-rzg2lc.srec	1	00000100	00000

RZ/G2UL Evaluation Board Kit

File name	Partition to save to eMMC	Address to save to eMMC	Address to load to RAM
bl2_bp-smarc-rzg2ul.srec	1	00000001	11E00
fip-smarc-rzg2ul.srec	1	00000100	00000

RZ/V2L Evaluation Board Kit PMIC version

File name	Partition to save to eMMC	Address to save to eMMC	Address to load to RAM
bl2_bp-smarc-rzv2l_pmic.srec	1	00000001	11E00
fip-smarc-rzv2l_pmic.srec	1	00000100	00000

2.5.3 Create a microSD card to boot linux for eMMC boot

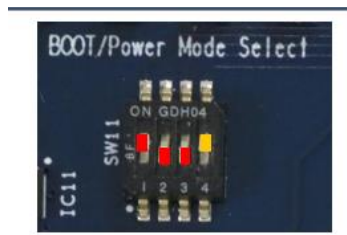
Create a microSD card by [1.5 Create a microSD card to boot linux](#). After finishing step(1) to (8), please back to following instructions before un-mounting a microSD card.

Copy a kernel image, a device tree file and rootfs to the second partition of the microSD card.

```
$ cd /media/user/rootfs/home/root/
$ sudo cp $WORK/build/tmp/deploy/images/<board>/<Linux kernel> ./
$ sudo cp $WORK/build/tmp/deploy/images/<board>/<Devise tree> ./
$ sudo cp $WORK/build/tmp/deploy/images/<board>/<root filesystem> ./
$ cd $WORK
$ sudo umount /media/user/rootfs
```

2.5.4 Setting U-boot and Writing rootfs to eMMC

To set the board to eMMC Boot mode, set the SW11 as below:

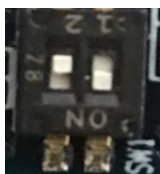


SW11

1	2	3	4
ON	OFF	OFF	ON

Note:-

Set the SW1 on SoM module to eMMC mode. Please refer to the section 1.1.3.



Refer to [1.6 Setting U-boot](#) and set the u-boot environment variables (please ignore switch settings). Then, turn on the power of the board by pressing the reset button SW10.

After booting Linux, please login as root and create partitions on eMMC

```

root@smarc-rzv21:~# fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.35.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): o
Created a new DOS disklabel with disk identifier 0xf3d53104.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): (Push the enter key)
Partition number (1-4, default 1): (Push the enter key)
First sector (2048-124321791, default 2048): (Push the enter key)
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-124321791, default 124321791): +500M

Created a new partition 1 of type 'Linux' and of size 500 MiB.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): (Push the enter key)

```

```

Using default response p.
Partition number (2-4, default 2): (Push the enter key)
First sector (1026048-124321791, default 1026048): (Push the enter key)
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1026048-124321791, default 1243
21791): (Push the enter key)

Created a new partition 2 of type 'Linux' and of size 58.8 GiB.

Command (m for help): p
Disk /dev/mmcblk0: 59.29 GiB, 63652757504 bytes, 124321792 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf3d53104

Device          Boot   Start       End   Sectors  Size Id Type
/dev/mmcblk0p1          2048   1026047   1024000  500M 83 Linux
/dev/mmcblk0p2    1026048 124321791 123295744  58.8G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@smarc-rzv2l:~#

```

Format eMMC.

```

root@smarc-rzv2l:~# mkfs.ext4 /dev/mmcblk0p1
root@smarc-rzv2l:~# mkfs.ext4 /dev/mmcblk0p2

```

Format eMMC and write the kernel, the device tree, and the rootfs.

```

root@smarc-rzv2l:~# mount /dev/mmcblk0p1 /mnt/
root@smarc-rzv2l:~# cp Image-smarc-rzv2l.bin /mnt/
root@smarc-rzv2l:~# cp Image-r9a07g05412-smarc.dtb /mnt/
root@smarc-rzv2l:~# umount /dev/mmcblk0p1
root@smarc-rzv2l:~# mount /dev/mmcblk0p2 /mnt/
root@smarc-rzv2l:~# tar xf /home/root/core-image-weston-smarc-rzv2l.tar.bz2 \
-C /mnt/
root@smarc-rzv2l:~# umount /dev/mmcblk0p2

```

2.5.5 Setting U-boot for eMMC boot

Reset the board by pressing the reset button SW10.

```
U-Boot 2021.10 (Apr 22 2022 - 03:04:59 +0000)

CPU:   Renesas Electronics K rev 16.15
Model: smarc-rzv2l
DRAM:  1.9 GiB
MMC:   sd@11c00000: 0, sd@11c10000: 1
Loading Environment from MMC... OK
In:    serial@1004b800
Out:   serial@1004b800
Err:   serial@1004b800
Net:   eth0: ethernet@11c20000
Hit any key to stop autoboot:  0
=>
```

Set environment variables to boot from eMMC. The following variables are for the RZ/V2L board. Please replace the file names in “bootcmd” according to the Release Note when you use other boards, and set IP address of your Linux Host PC..

```
=> setenv bootargs 'root=/dev/mmcblk0p2 rootwait'
=> setenv bootcmd 'mmc dev 1; ext4load mmc 0:1 0x48080000 Image-smarc-rzv2l.bi
n; ext4load mmc 0:1 0x48000000 Image-r9a07g05412-smarc.dtb; booti 0x48080000 -
0x48000000'
=> saveenv
Saving Environment to MMC... Writing to MMC(0)...OK
```

Please reset the board again for eMMC boot.

3. Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 31, 2022	-	First edition issued.
1.01	Jun. 6, 2022	15	Modify the Table4 of RZ/Five Evaluation Board Kit.
		16	Add instruction to format SD cards.
		24	Move the section “Preparing Flash Writer”, “How to distinguish each board”, and “How to replace the SMARC Module Board” to the chapter 2 “Appendix”.
1.02	Jun. 17, 2022	14	Update the description in the section 1.4.
		31	Add the descriptions for booting from eMMC.
1.03	Dec. 27, 2022	20	Add the table 7 to explain the image names.
		31	Add the sentence to explain the section.