Reference Boards of RZ/G2H, RZ/G2M, RZ/G2N, and RZ/G2E

Start-up Guide

R01US0555EJ0101
Rev.1.01
Jun. 6, 2022

Introduction

This document provides a guide to prepare RZ/G2 reference boards to boot up with the Verified Linux Package v3.0.x. Especially, procedures to write bootloader to each board are explained.

Bootloaders are written to the Flash ROM on the board using the Flash Writer or the Mini Monitor provided by Renesas. This document explains the way to write these files using the Flash Writer.

Target Reference Board

RZ/G2 Group reference boards

- Hoperun Technology HiHope RZ/G2H platform (hihope-rzg2h)
- Hoperun Technology HiHope RZ/G2M platform (hihope-rzg2m)
- Hoperun Technology HiHope RZ/G2N platform (hihope-rzg2n)
- Silicon Linux RZ/G2E evaluation kit (EK874)

Target Software

- RZ/G Verified Linux Package version 3.0.0 or later. (hereinafter referred to as “VLP/G”)
- RZ/G Verified Linux Package for 64bit kernel version 1.0.12 or later. (hereinafter referred to as “VLP64”)

Note) RZ/G2L, RZ/G2LC, and RZ/G2UL are not the target boards in this document.

Contents

1. Hoperun Technology HiHope RZ/G2[HMN] platform (hihope-rzg2h, hihope-rzg2m, hihope-rzg2n)

1.1 Preparation

1.1.1 Building files to write

This board uses the files below as a bootloader. Please build them according to the Release Note and copy these files to the PC which runs a serial terminal software.

- bootparam_sa0.srec
- bl2-hihope-rzg2h.srec (RZ/G2H) or bl2-hihope-rzg2m.srec (RZ/G2M) or bl2-hihope-rzg2n.srec (RZ/G2N)
- cert_header_sa6.srec
- bl31-hihope-rzg2h.srec (RZ/G2H) or bl31-hihope-rzg2m.srec (RZ/G2M) or bl31-hihope-rzg2n.srec (RZ/G2N)
- u-boot-elf-hihope-rzg2h.srec (RZ/G2H) or u-boot-elf-hihope-rzg2m.srec (RZ/G2M) or u-boot-elf-hihope-rzg2n.srec (RZ/G2N)

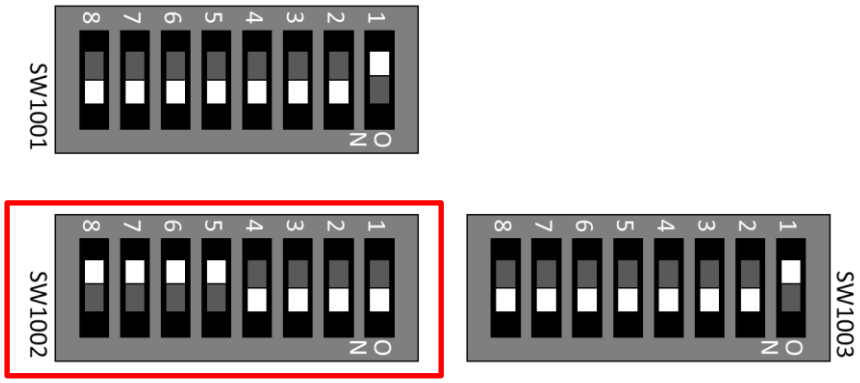
1.1.2 Settings

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

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

To set the board to SCIF Download mode, set the SW1002 as below:



SW1002

8	7	6	5	4	3	2	1
OFF	OFF	OFF	OFF	ON	ON	ON	ON

Note) Be careful not to change the SW1001 and SW1003

1.2 Booting Flash Writer

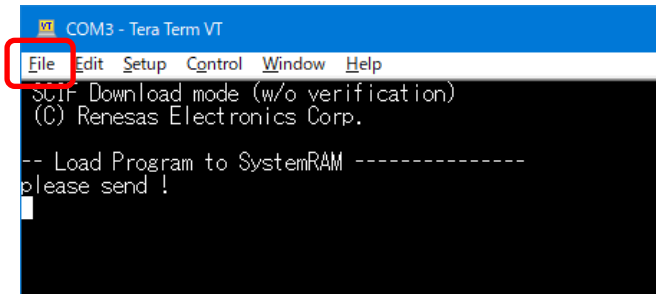
Turn on the power of the board by changing the SW2402. Messages below are shown on the terminal.

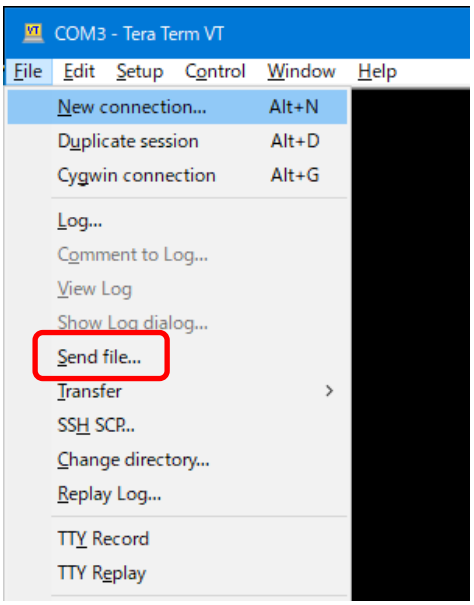
```
 SCIF Download mode (w/o verification)
 (C) Renesas Electronics Corp.

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

Send an image of Flash Writer (AArch64_Flash_writer_SCIF_DUMMY_CERT_E6300400_hihope.mot) using terminal software after the message “please send !” is shown.

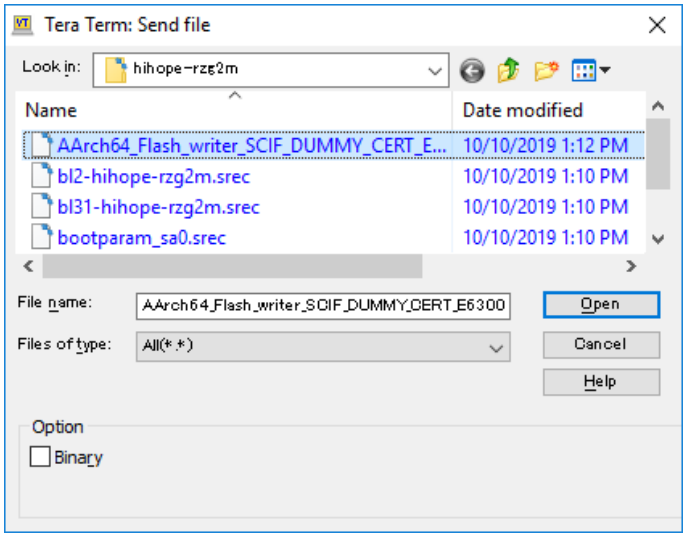
Below is a sample procedure with Tera Term.

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

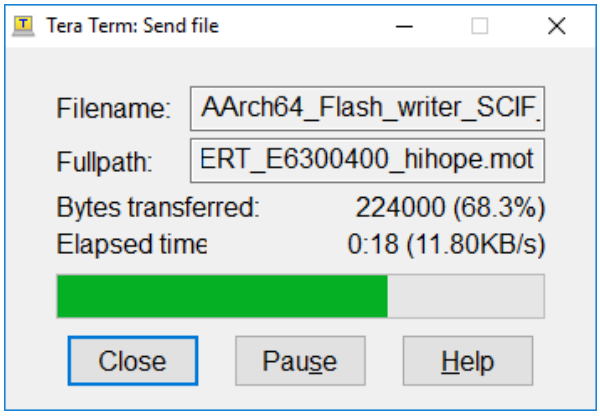




Then, select the image to be send and click “Open” button.



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



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

```
Flash writer for RZ/G2M V1.00 Sep.24,2019
>
```

1.3 Writing Bootloader

"xls2" command of Flash Writer is used to write binary files. This command receives binary data from the serial port and write the data to specified address of the Flash ROM with information where the data should be loaded on the address of the main memory.

This is an example of writing "bootparam_sa0.srec" which should be placed to E6320000h of the main memory to 000000h of the Flash ROM.

```
>xls2
===== Qspi writing of RZ/G2 Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
 Winbond : W25M512JW
Program Top Address & Qspi Save Address
===== Please Input Program Top Address ============
  Please Input : H'E6320000

===== Please Input Qspi Save Address ===
```

```
  Please Input : H'000000
Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ('.' & CR stop load)
```

Send the data of “bootparam_sa0.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-H'00007FFF Erasing.
........Erase Completed
SAVE SPI-FLASH.......
======= Qspi  Save Information  =================
 SpiFlashMemory Stat Address : H'00000000
 SpiFlashMemory End Address  : H'00000E67
============================================================
```

```
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”.

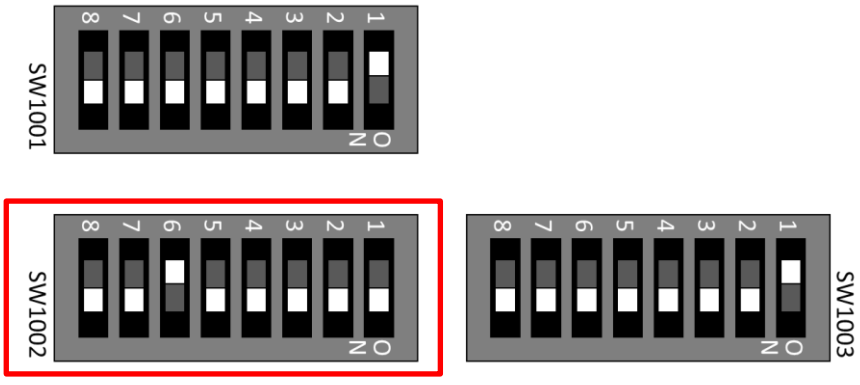
Write all necessary files using the addresses listed at Table 1 and turn off the power of the board by changing the SW2402.

Table 1. Addresses for each file

File name	Address to load to RAM	Address to save to ROM
bootparam_sa0.srec	E6320000	000000
bl2-hihope-rzg2h.srec or bl2-hihope-rzg2m.srec or bl2-hihope-rzg2n.srec	E6304000	040000
cert_header_sa6.srec	E6320000	180000
bl31-hihope-rzg2h.srec or bl31-hihope-rzg2m.srec or bl31-hihope-rzg2n.srec	44000000	1C0000
tee-hihope-rzg2h.srec or tee-hihope-rzg2m.srec or tee-hihope-rzg2n.srec	44100000	200000
u-boot-elf-hihope-rzg2m.srec or u-boot-elf-hihope-rzg2n.srec	50000000	300000

1.4 Setting U-boot

To set the board to SPI Boot mode, set the SW1002 as below:



SW1002

8	7	6	5	4	3	2	1
ON	ON	OFF	ON	ON	ON	ON	ON

Note) Be careful not to change the SW1001 and SW1003

Turn on the power of the board by changing the SW2402.

```
U-Boot 2020.10 (May 07 2021 - 13:48:35 +0000)

CPU: Renesas Electronics R8A774A1 rev 1.1/rev 1.2
Model: Hoperun Technology HiHope RZ/G2M platform (hihope-rzg2m)
DRAM:  3.9 GiB
Bank #0: 0x048000000 - 0x0bfffffff, 1.9 GiB
Bank #1: 0x600000000 - 0x67fffffff, 2 GiB

MMC:   sd@ee100000: 0, sd@ee160000: 1
Loading Environment from SPI Flash... SF: Detected w25m512jw with page size 25
6 Bytes, erase size 4 KiB, total 32 MiB
*** Warning - bad CRC, using default environment
```

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 SPI Flash... SF: Detected w25m512jv with page size 256 B
ytes, erase size 4 KiB, total 32 MiB
Erasing SPI flash...Writing to SPI flash...done
OK
```

In case booting from micro SD card, set environment variables using the commands below:

```
=> setenv bootargs 'root=/dev/mmcblk1p2 rootwait'
=> setenv bootcmd 'fatload mmc 0:1 0x48080000 Image-hihope-rzg2m.bin; fatload
mmc 0:1 0x48000000 Image-r8a774a1-hihope-rzg2m.dtb; booti 0x48080000 - 0x48000
000'
=> saveenv
Saving Environment to SPI Flash... SF: Detected w25m512jv with page size 256 B
ytes, erase size 4 KiB, total 32 MiB
Erasing SPI flash...Writing to SPI flash...done
OK
```

Note) The setting above assumes the SD card has two partitions and stores data as below:
First partition: formatted as FAT, includes Image-hihope-rzg2m.bin and Image-r8a774a1-hihope-rzg2m.dtb
Second partition: formatted as ext4, rootfs image is expanded
Please refer to chapter 3. Create a microSD card to boot Linux.

Note) Please replace the file names in “bootcmd” according to the Release Note.

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

2. Silicon Linux RZ/G2E evaluation kit (EK874)

2.1 Preparation

2.1.1 Building files to write

This board uses the files below as a bootloader. Please build them according to the Release Note and copy these files to the PC which runs a serial terminal software.

- bootparam_sa0.srec
- bl2-ek874.srec
- cert_header_sa6.srec
- bl31-ek874.srec
- u-boot-elf-ek874.srec

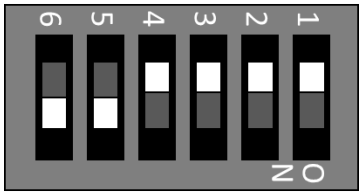
2.1.2 Settings

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

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

To set the board to SCIF Download mode, set the SW12 which is placed near the micro SD card slot as below:



6	5	4	3	2	1
ON	ON	OFF	OFF	OFF	OFF

2.2 Booting Flash Writer

Turn on the power of the board by changing the SW23. Messages below are shown on the terminal.

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

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

Send an image of Flash Writer (AArch64_Flash_writer_SCIF_DUMMY_CERT_E6300400_ek874.mot) from terminal software after the message “please send !” is shown.

For detail of the procedure, please refer to the section 1.2 Booting Flash Writer.

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

```
Flash writer for RZ/G2E V1.00 Sep.24,2019
>
```

2.3 Writing Bootloader

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

For detail of the procedure, please refer to the section **1.3 Writing Bootloader**.

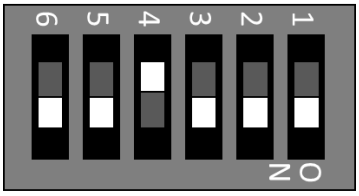
Write all necessary files using the addresses listed at Table 2 and turn off the power of the board by changing the SW23.

Table 2. Addresses for each file

File name	Address to load to RAM	Address to save to ROM
bootparam_sa0.srec	E6320000	000000
bl2-ek874.srec	E6304000	040000
cert_header_sa6.srec	E6320000	180000
bl31-ek874.srec	44000000	1C0000
tee-ek874.srec	44100000	200000
u-boot-elf-ek874.srec	50000000	300000

2.4 Setting U-boot

To set the board to SPI Boot mode, set the SW12 which is placed near the micro SD card slot as below:



6	5	4	3	2	1
ON	ON	OFF	ON	ON	ON

Turn on the power of the board by changing the SW23.

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

CPU:   Renesas Electronics R8A774C0 rev 1.1
Model: Silicon Linux RZ/G2E evaluation kit EK874 (CAT874 + CAT875)
DRAM:  1.9 GiB
WDT:   watchdog@00000000e6020000
WDT:   Started with servicing (60s timeout)
MMC:   mmc@ee100000: 0
Loading Environment from SPIFlash... SF: Detected w25q512jv with page size 256
 Bytes, erase size 4 KiB, total 64 MiB
OK
In:    serial@e6e88000
Out:   serial@e6e88000
Err:   serial@e6e88000
```

```
U-boot WDT started!
Net:
Error: ethernet@e6800000 address not set.
No ethernet found.

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 SPI Flash... SF: Detected w25m512jv with page size 256 B
ytes, erase size 4 KiB, total 32 MiB
Erasing SPI flash...Writing to SPI flash...done
OK
```

In case booting from micro SD card, set environment variables using the commands below:

```
=> setenv bootargs 'root=/dev/mmcblk1p2 rootwait'
=> setenv bootcmd 'fatload mmc 0:1 0x48080000 Image-ek874.bin; fatload mmc 0:1
 0x48000000 Image-r8a774c0-ek874.dtb; booti 0x48080000 - 0x48000000'
=> saveenv
Saving Environment to SPI Flash... SF: Detected w25m512jv with page size 256 B
ytes, erase size 4 KiB, total 32 MiB
Erasing SPI flash...Writing to SPI flash...done
OK
```

When you use the early version of EK874 (Revision A, B, C), please set “bootargs” as below.

```
=> setenv bootargs 'root=/dev/mmcblk0p2 rootwait'
```

Note) The setting above assumes the SD card has two partitions and stores data as below:
First partition: formatted as FAT, includes Image-ek874.bin and Image-r8a774c0-ek874.dtb
Second partition: formatted as ext4, rootfs image is expanded.
Please refer to chapter 3. Create a microSD card to boot Linux.

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

3. 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 micro-SD card, unmount all partitions.

(5) Change the partition table

MicroSD card needs two partitions as listed in Table 3.

Table 3. 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 6233395
1): +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 62333
951): (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-E6A
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.

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

HiHope RZ/G2H board: hihope-rzg2h

HiHope RZ/G2M board: hihope-rzg2m

HiHope RZ/G2N board: hihope-rzg2n

EK874 RZ/G2E board: ek874

Please replace <*Linux kernel*> and <*root filesystem*> by the name below:

Table 4. File names of kernel and root filesystem

	Linux kernel	root filesystem
RZ/G2H	Image-hihope-rzg2h.bin	<*image name*>-hihope-rzg2h.tar.bz2
RZ/G2M v1.3 **RZ/G2M v3.0**	Image-hihope-rzg2m.bin	<*image name*>-hihope-rzg2m.tar.bz2
RZ/G2N	Image-hihope-rzg2n.bin	<*image name*>-hihope-rzg2n.tar.bz2
RZ/G2E	Image-ek874.bin	<*image name*>-ek874.tar.bz2

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

minimal: core-image-minimal

bsp: core-image-bsp

weston: core-image-weston

qt: core-image-qt

Please replace <*Device tree*> by the name below:

Table 5. File names of Device tree

Device tree	Type 1	Type 2	Type 3	Type 4
RZ/G2H	Image-r8a774e1-hihope-rzg2h.dtb	Image-r8a774e1-hihope-rzg2h-ex.dtb	Image-r8a774e1-hihope-rzg2h-ex-idk-1110wr.dtb	Image-r8a774e1-hihope-rzg2h-ex-mipi-2.1.dtb
RZ/G2M v1.3 (*1)	Image-r8a774a1-hihope-rzg2m.dtb	Image-r8a774a1-hihope-rzg2m-ex.dtb	Image-r8a774a1-hihope-rzg2m-ex-idk-1110wr.dtb	Image-r8a774a1-hihope-rzg2m-ex-mipi-2.1.dtb
RZ/G2M v3.0 (*1)	Image-r8a774a3-hihope-rzg2m.dtb	Image-r8a774a3-hihope-rzg2m-ex.dtb	Image-r8a774a3-hihope-rzg2m-ex-idk-1110wr.dtb	Image-r8a774a3-hihope-rzg2m-ex-mipi-2.1.dtb
RZ/G2N	Image-r8a774b1-hihope-rzg2n.dtb	Image-r8a774b1-hihope-rzg2n-ex.dtb	Image-r8a774b1-hihope-rzg2n-ex-idk-1110wr.dtb	Image-r8a774b1-hihope-rzg2n-ex-mipi-2.1.dtb
RZ/G2E EK874 Rev C (*2)	Image-r8a774c0-cat874-revc.dtb	Image-r8a774c0-ek874-revc.dtb	Image-r8a774c0-ek874-revc-idk-2121wr.dtb	Image-r8a774c0-ek874-revc-mipi-2.1.dtb
RZ/G2E EK874 Rev E (*2)	Image-r8a774c0-cat874.dtb	Image-r8a774c0-ek874.dtb	Image-r8a774c0-ek874-idk-2121wr.dtb	Image-r8a774c0-ek874-mipi-2.1.dtb

There are 4 types of the device tree files. Available devices are different depending on them. Please refer to the following description:

- **Type1**: Main board only
- **Type2**: Main board + Sub board
- **Type3**: Main board + Sub board + LVDS panel
- **Type4**: Main board + Sub board + MIPI/CSI2 cameras

Please note that users who use the combination of main and sub boards need to use type2-4 as a device tree file. **If the dtb files of type1 are used, interfaces on the sub board such as Ethernet are not able to be used.**

(*1) There are 2 types of RZ/G2M LSI (“RZ/G2M v3.0” and “RZ/G2M v1.3”). In case you use the hihope-rzg2m board which has one of them, the same image files can be used, but the same device tree files cannot be used. Please refer to the above table. If the board prints the messages below when turn on the power, RZ/G2M v3.0 is used on your board. In case of RZ/G2M v1.3, “R8A774A1” will be displayed.

```
CPU: Renesas Electronics R8A774A3
```

(*2) There are 2 revisions of the ek874 board (Rev C, E). In case you use the ek874 board, the same image files can be used, but the same device tree files cannot be used. Please refer to the above table. You can check the board revision with the silk printed on the board.

Note) The dtb files listed in the Table 5 cannot be used for the early version of boards. See the **4. Notes** section. In case you use the hihope-rzg2m board which has RZ/G2M v3.0 LSI, the dtb files listed in the Table 5 can be used because the board is the latest version.

4. Notes

(1) Device tree for RZ/G2M and RZ/G2N

The dtb files listed in the **Table 5** cannot be used for the early revision of Hoperun boards. If you are using revision 2 boards, please use below files. These are automatically generated at the same place as the other image files when building a BSP.

HiHope RZ/G2M board:

- Image-r8a774a1-hihope-rzg2m-**rev2**.dtb (main board only)
- Image-r8a774a1-hihope-rzg2m-**rev2**-ex.dtb (main + sub board)
- Image- r8a774a1-hihope-rzg2m-**rev2**-ex-idk-1110wr.dtb (main + sub board + LVDS-IF)
- Image-r8a774a1-hihope-rzg2m-**rev2**-ex-mipi-2.1.dtb (main + sub board + MIPI/CSI2 cameras)

HiHope RZ/G2N board:

- Image-r8a774b1-hihope-rzg2n-**rev2**.dtb (main board only)
- Image-r8a774b1-hihope-rzg2n-**rev2**-ex.dtb (main + sub board)
- Image- r8a774b1-hihope-rzg2n-**rev2**-ex-idk-1110wr.dtb (main + sub board + LVDS-IF)
- Image-r8a774b1-hihope-rzg2n-**rev2**-ex-mipi-2.1.dtb (main + sub board + MIPI/CSI2 cameras)

(2) Device tree for RZ/G2E

The dtb files listed in the Table 5 cannot be used for the early version (ES1.0) of RZ/G2E.

If the board prints the messages below when turn on the power, ES1.0 of RZ/G2E is implemented on the board.

```
[    0.000096] NOTICE:  BL2: RZ G2E Initial Program Loader(CA53)
[    0.004373] NOTICE:  BL2: Initial Program Loader(Rev.1.0.23)
[    0.009991] NOTICE:  BL2: PRR is RZG G2E Ver.1.0
```

```
CPU: Renesas Electronics R8A774C0 rev 1.0
```

In this case, please use below dtb files instead. These are built simultaneously when building normal dtb files.

- Image-r8a774c0-**es10**-cat874.dtb (main board only)
- Image-r8a774c0-**es10**-ek874.dtb (main + sub board)
- Image-r8a774c0-**es10**-ek874-idk-2121wr.dtb (main + sub board + LVDS panel)
- Image-r8a774c0-**es10**-ek874-mipi-2.1.dtb (main + sub board + MIPI/CSI2 cameras)

In case both old and new RZ/G2E LSIs are used in your laboratory at the same time, dtb files can automatically be selected using the environment variable `cut_ver` which is set by u-boot program according to the LSI's information. Please store multiple dtb files in an SD card and set `bootcmd` of u-boot like this.

```
setenv bootcmd 'fatload mmc 0:1 0x48080000 Image; if test "${cut_ver}" = "10";
 then fatload mmc 0:1 0x48000000 Image-r8a774c0-es10-ek874.dtb; else fatload m
mc 0:1 0x48000000 Image-r8a774c0-ek874.dtb ; fi ; booti 0x48080000 - 0x4800000
0'
```

5. Appendix

Preparing Flash Writer

Flash Writer is built automatically when building BSP by bitbake command. Please refer to the Release Note of the Verified Linux Package 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.

5.1 Preparing cross compiler

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

Linaro toolchain:

```
$ cd ~/
$ wget https://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/a
arch64-elf/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-elf.tar.xz
$ tar xvf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-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
```

5.2 Building Flash Writer

Get source codes of Flash Writer from the GitHub repository and checkout according to the version of VLP you use.

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

For VLP/G v3.0.xx

```
$ cd rzg2_flash_writer
$ git checkout -b tmp 6606c4f4351f56fb3bd84d7836c01c8932a4f884
```

For VLP64 v1.0.xx

```
$ cd rzg2_flash_writer
$ git checkout -b v1.05 v1.05
```

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

Linaro toolchain:

```
$ make -f makefile.linaro clean
$ CROSS_COMPILE=~/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-elf/bin/aarch64-elf-
 make -f makefile.linaro BOARD=<board>
```

Yocto SDK:

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

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

Target board	**BOARD option** *<board>*	**Image to be generated**
HiHope RZ/G2H	HIHOPE	AArch64_Flash_writer_SCIF_DUMMY_CERT_E6300400_hihope.mot
HiHope RZ/G2M		
HiHope RZ/G2N		
EK874	EK874	AArch64_Flash_writer_SCIF_DUMMY_CERT_E6300400_ek874.mot

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar. 31, 2022	−	First edition issued.
1.01	Jun. 6, 2022	11	Add the instructions to format SD cards.
		17	Add the Notes section.
		18	Move the section “Preparing Flash Writer” to the chapter 4 “Appendix”.