RZ/G3E-EVKIT

Linux Start-up Guide

R01US0805EJ0100
Rev.1.00
Jun. 30, 2025

Introduction

This document provides a guide to prepare RZ/G3E reference board to boot up with the Board Support Package.

This guide provides the following information:

- Building procedure
- Preparation for use
- Boot loader and U-Boot
- How to run this Linux package on the target board
- How to create a software development kit (SDK)

Target Reference Board

The target boards of this BSP are as below table.

Table 1. Target bord list

Device	**Evaluation Board**
RZ/G3E	RZ/G3E Evaluation Board Kit. (P/N: RTK9947E57S01000BE)

Target Software

- RZ/G3E Board Support Package version 1.0.0 or later. (hereinafter referred to as “BSP”)

Contents

1. Environmental Requirement

The environment for building the Board Support Package (hereinafter referred to as "BSP") is listed in Table1. Please refer to the documents below for details about setting up the environment:

Figure 1 shows the recommended environment for this package.

A Linux PC is required for building the software.

A Windows PC can be used as a serial terminal interface with software such as TeraTerm.

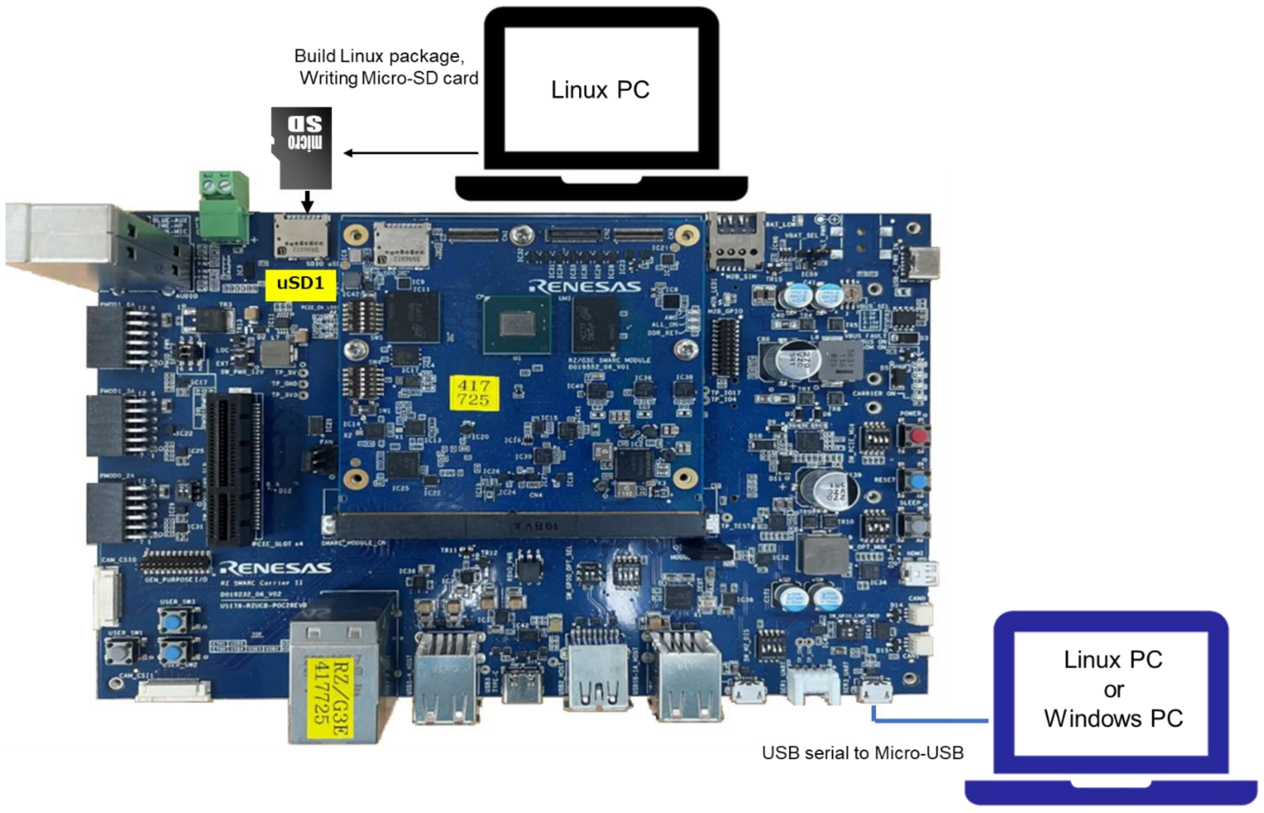


Figure 1. Recommend Environment

Table 2. Equipment and Software for Developing Environments of Linux Platform

Equipment		Description
Linux Host PC		Used as build/debug environment 100GB free space on HDD or SSD is necessary
	OS	Ubuntu 22.04 LTS (*1) 64 bit OS must be used.
Windows Host PC		Used as debug environment, controlling with terminal software
	OS	Windows 11
	Terminal software	Used for controling serial console of the target board Tera Term (latest version) is recommended Available at Releases TeraTermProject/teraterm(github.com)
	VCP Driver	Virtual COM Port driver which enables to communicate Windows Host PC and the target board via USB which is virtually used as serial port. Available at: http://www.ftdichip.com/Drivers/VCP.htm
USB serial to mini–USB Cable		Serial communication (UART) between the Evaluation Board Kit and Windows PC. The type of USB serial connector on the Evaluation Board Kit is USB type Mini-B.
micro–SD Card		Use to boot the system, and store applications.

Below images can be built “offline”.

- core-image-minimal
- core-image-weston (including the SDK build)

2. Build Instructions

⚠ The BSP is only built in **Ubuntu 22.04**

Ubuntu 22.04 is required to build the BSP. This is because it was the only host operating system tested and is a specific requirement for Yocto 5.0 (scarthgap).

2.1 Building images

This section describes the instructions to build the Board Support Package.
Before starting the build, run the command below on the Linux Host PC to install packages used for building the BSP.

```
$ sudo apt-get update
$ sudo apt install build-essential chrpath cpio debianutils diffstat file gawk \
gcc git iputils-ping libacl1 liblz4-tool locales python3 python3-git \
python3-jinja2 python3-pexpect python3-pip python3-subunit socat texinfo unzip \
wget xz-utils zstd
```

Please refer to the URL below for detailed information:

- https://docs.yoctoproject.org/5.0.9/brief-yoctoprojectqs/

Run the commands below and set the username and email address before starting the build procedure. Without this setting, an error occurs when building procedure runs git command to apply patches.

```
$ git config --global user.email "you@example.com"
$ git config --global user.name "Your Name"
```

Copy all files obtained from Renesas into your Linux Host PC prior to the steps below. The directory which you put the files in is described as <package download directory> in the build instructions.

(1) Set the environment variable of the package

Set the version number of the package you are using as an environment variable. The following is an example, so please rewrite it to match your package.

```
$ PACKAGE_VERSION=1.0.0
```

(2) Create a working directory at your home directory, and decompress Yocto recipe package

Run the commands below. The name and the place of the working directory can be changed as necessary.

```
$ mkdir ~/rzg3e_bsp_v${PACKAGE_VERSION}
$ cd ~/rzg3e_bsp_v${PACKAGE_VERSION}
$ cp ../<package download directory>/*.zip .
$ unzip ./RTK0EF0045Z0040AZJ-v${PACKAGE_VERSION}.zip
$ tar zxvf ./RTK0EF0045Z0040AZJ-v${PACKAGE_VERSION}/rzg3e_bsp_v${PACKAGE_VERSION}.\
tar.gz
```

Note) Please note that your build environment must have 200GB of free hard drive space to complete the minimum build. The Yocto BSP build environment is very large. Especially in case you are using a Virtual Machine, please check how much disk space you have allocated for your virtual environment.

(3) Enable Graphics and Video Codec (Optional)

If you want to enable the graphics package and the video codec package, please follow this step. You can use them simultaneously, or you can use either package individually. Please check the website page below for available combination with Yocto recipe package.

https://www.renesas.com/software-tool/rzg3e-board-support-package

The following instructions are for the EN packages. If you use the JP package, replace “EN” with “JP”.

For Graphics Package

Please copy the graphics package (RTK0EF0045Z14001ZJ-<version>_rzg_EN.zip or RTK0EF0045Z14001ZJ-<version>_rzg_JP.zip) to the working directory and then run the commands below.

```
$ unzip ./RTK0EF0045Z14001ZJ-<version>_rzg_EN.zip
$ tar zxvf ./RTK0EF0045Z14001ZJ-<version>_rzg_EN/meta-rz-features_graphics_\
<version>.tar.gz
```

For Video Codec Package

Please copy the video codec package (RTK0EF0207Z00001ZJ-<version>_rzg3e_EN.zip or RTK0EF0207Z00001ZJ-<version>_rzg3e_JP.zip) to the working directory and then run the commands below.

```
$ unzip ./RTK0EF0207Z00001ZJ-<version>_rzg3e_EN.zip
$ tar zxvf ./RTK0EF0207Z00001ZJ-<version>_rzg3e_EN/meta-rz-features_codec_\
<version>.tar.gz
```

(4) Build Initialize

Please initialize a build using the 'oe-init-build-env' script in Poky and point TEMPLATECONF to platform conf path.

```
$ TEMPLATECONF=$PWD/meta-renesas/meta-rz-distro/conf/templates/rz-conf/ source \
poky/oe-init-build-env build
```

(5) Add layers (Optional)

Please follow the steps below to add the layers you need. The steps add the settings to bblayers.conf.

- **Graphics**: Please run the command below if you need the Graphics library.

```
$ bitbake-layers add-layer ../meta-rz-features/meta-rz-graphics
```

- **Video Codec**: Please run the command below if you need the video codec library.

```
$ bitbake-layers add-layer ../meta-rz-features/meta-rz-codecs
```

- **Docker**: Please run the commands below if you want to include Docker. This means running Docker on the RZ board, not as using Docker as part of your build environment.

```
$ bitbake-layers add-layer ../meta-openembedded/meta-networking
$ bitbake-layers add-layer ../meta-openembedded/meta-filesystems
$ bitbake-layers add-layer ../meta-virtualization
```

(6) Start a build

Please run the commands below to start a build. Building the target file system image with BitBake can take several hours, depending on your host system's performance.

```
$ MACHINE=smarc-rzg3e bitbake <target>
```

<target> can be selected, so please refer to the following table for supported image.

Table 3. List of platforms and the board

Target Image Name	Purpose
core-image-minimal	Minimal set of components.
core-image-weston	Standard image with graphics support.

After the building is successfully completed, a similar output will be seen, and the command prompt will return. Please note that the number of tasks may change depending on your BSP version and other factors.

```
NOTE: Tasks Summary: Attempted 8281 tasks of which 16 didn't need to be rerun and all
succeeded.
```

All necessary files listed in the below table will be generated by the bitbake command and will be in the **"build/tmp/deploy/images"** directory.

Table 4. Image files

RZ/G3E	**Linux kernel**	Image-smarc-rzg3e.bin
	Device tree file	r9a09g047e57-smarc-smarc-rzg3e.dtb
	root filesystem	<target>-smarc-rzg3e.rootfs.tar.bz2
	Boot loader	bl2_bp_spi-smarc-rzg3e.srec bl2_bp_mmc-smarc-rzg3e.srec bl2_bp_esd-smarc-rzg3e.bin fip-smarc-rzg3e.srec fip-smarc-rzg3e.bin
	SD image	<target>-smarc-rzg3e.rootfs.wic.gz <target>-smarc-rzg3e.rootfs.wic.bmap
	Software Bill of Materials (SBOM)SBOM	*<image name>*-smarc-rzg3e.rootfs.spdx.tar.zst

If you want to know the components installed in the root file system, please check the manifest file. This file is created in the following path after building the images:

~/rzg3e_bsp_<package version>/build/tmp/deploy/images/smarc-rzg3e/ <target>-smarc-rzg3e.rootfs.manifest

2.2 Notes

This section summarizes options or notes that can be used during the build process. Please refer to them as needed.

(1) Exclude GPLv3 packages

Please note that OSS packages licensed under the GPLv3 are installed by default when you build this BSP following the instructions in section 2.1. If you wish to exclude these packages, please run the commands below to download meta-rz-nogplv3.

```
$ cd ~/rzg3e_bsp_v${PACKAGE_VERSION}/
$ git clone https://github.com/renesas-rz/meta-rz-nogplv3.git
$ cd meta-rz-nogplv3/
$ git checkout 7c9d35357d95a9603da412aa5a8ecc3c8f723658s
$ cd ..
```

Run the command below to add the layer. (*)

```
$ cd build
$ bitbake-layers add-layer ../meta-rz-nogplv3
```

(*) If you haven't run the command below yet, please run it first.

```
$ TEMPLATECONF=${PWD}/meta-renesas/meta-rz-boards/conf/templates/rz-conf/ \
source poky/oe-init-build-env
```

Note) Renesas provides no guarantees or support whatsoever. The customer is solely responsible for cloning the source from GitHub at their own discretion.

Note) Please note that meta-rz-nogplv3 and meta-virtualization cannot be enabled at the same time.

(2) Real time performance

If you want to use the kernel which improves the real-time performance, please add the line below to the file "~/rzg3e_bsp_v${PACKAGE_VERSION}/build/conf/local.conf".

```
PREFERRED_PROVIDER_virtual/kernel = "linux-renesas-rt"
```

(3) Locale Issue

If bitbake shows below error log after starting building.

```
Please make sure locale 'en_US.UTF-8' is available on your system
```

Please run below command on the build PC.

```
$ sudo apt-get install locales
:
:
Generation complete.

$ sudo dpkg-reconfigure locales
:
:
<Please choose case "en_US.UTF-8 UTF-8", then default the language to en_US.UTF-8>
:
This will select the default language for the entire system. If this system is a mul
ti-user system where not all users are able to speak the default language, they will
 experience difficulties.
```

```
  1. None  2. C.UTF-8  3. en_US.UTF-8
Default locale for the system environment: 3

Generating locales (this might take a while)...
  en_US.UTF-8... done
Generation complete.
$ sudo locale-gen en_US.UTF-8
Generating locales (this might take a while)...
  en_US.UTF-8... done
Generation complete.
$ sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
$ export LANG=en_US.UTF-8
```

(4) Docker

If you want to enable Docker, please refer to **8.1 Docker**. The section summarizes how to build and simply test Docker.

3. Preparing microSD Card

You can prepare the microSD card following instructions in this section.

(1) Set and unmount microSD card

Firstly, please insert the microSD card into your Linux PC and check its mount device name using the fdisk command. Then, please unmount the microSD card. Please note that sdb1 and sdb2 are used as example device names in the following step.

```
$ sudo fdisk -l
Disk /dev/sdb: 3.74 GiB, 3997171712 bytes, 7806976 sectors
Disk model: Storage Device
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: 0xxxxxxxxxx
$ umount /dev/sdb1
$ umount /dev/sdb2
```

(2) Expand disk image

Please run the command below to expand the disk image onto the microSD card. When you complete the following steps, the microSD card is prepared correctly. Please note that "sdb" is an example and it depends on your environment.

```
$ sudo bmaptool copy --bmap <wic image>.wic.bmap <wic image>.wic.gz /dev/sdb
```

The file nemes of *<wic image>* is listed in the Table 4.

Table 5. File and directory in the micro-SD card

Type/Number	Size	Filesystem	Contents
Primary #1	20MB	FAT32	bl2_bp_spi-smarc-rzg3e.srec bl2_bp_mmc-smarc-rzg3e.srec fip-smarc-rzg3e.srec
Primary #2	2.2GB	Ext4	./ ├── bin ├── boot │ ├── Image │ └── r9a09g047e57-smarc.dtb ├── dev ├── etc ├── home ├── lib ├── media ├── mnt ├── opt ├── proc ├── root ├── run ├── sbin ├── srv ├── sys ├── tmp ├── usr └── var

4. Reference Board Setting

4.1 Prerequisites

Chapter 4 introduces the switches, jumper pin settings, and other configuration options on the evaluation board. These changes are based on the factory default settings of the evaluation board and are intended for running Linux. Please note that changing settings other than those described here may cause Linux to malfunction.

Please refer to the manual below for your specific board for instructions on how to modify it. You can also find more detailed information about board settings in the manual for your specific board.

- RZ/G3E Evaluation Board User's Manual

After obtaining your reference board, please be sure to follow this document and write the bootloaders to the Flash ROM before starting the evaluation. Please refer to sections 4.5 and 4.6.

4.2 Overview of the evaluation environment

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

Hardware preparation (Users should purchase the following equipment.):

- USB Type-C cable compatible with USB PD. (e.g. AK-A8485011 (manufactured by Anker))
- USB PD Charger 45W (15V 3.0A) or more. (e.g. PowerPort III 65W Pod (manufactured by Anker))
- USB Type-A microB cable (Any cables)
- micro HDMI cable (Any cables)
- PC Installed FTDI VCP driver and Terminal software (Tera Term) (*1)

(*1) Please install the FTDI driver that can be following website (Releases TeraTermProject/teraterm(github.com)).

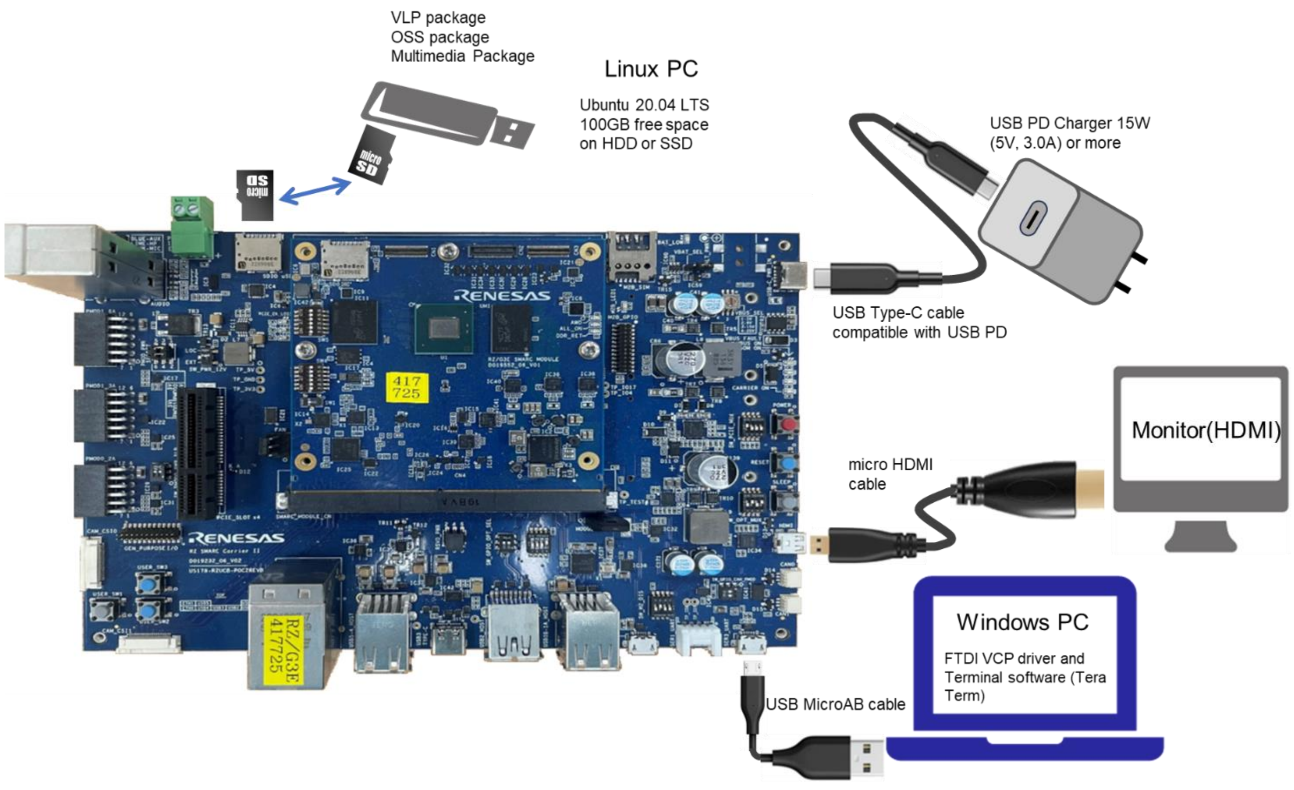

VLP package
OSS package
Multimedia Package
Linux PC
Ubuntu 20.04 LTS
100GB free space on HDD or SSD
USB PD Charger 15W (5V, 3.0A) or more
USB Type-C cable compatible with USB PD
Monitor(HDMI)
micro HDMI cable
Windows PC
FTDI VCP driver and Terminal software (Tera Term)
USB MicroAB cable

Figure 2. Recommend Environment

4.3 Board Settings of RZ/G3E Evaluation Board

4.3.1 How to set boot mode of RZ/G3E

This section explains how to set the boot mode of the evaluation board.

The boot mode can be changed using the DIP switch "SW_MODE". Please adjust it according to the figures below.

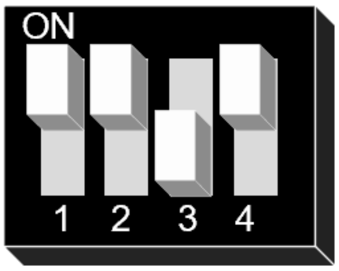


Figure 3. eSD Boot Mode

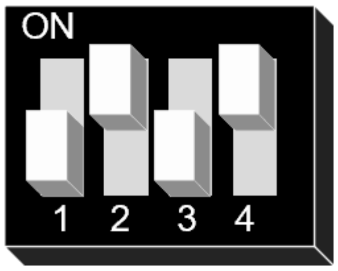


Figure 4. SCIF Download Mode

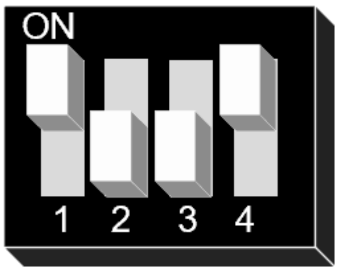


Figure 5. eMMC Boot Mode

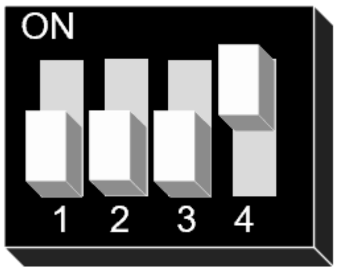


Figure 6. SPI Boot Mode

4.3.2 How to use debug serial (console ouput)

Connect the USB Type Micro-AB cable to the connector "SER3_UART".

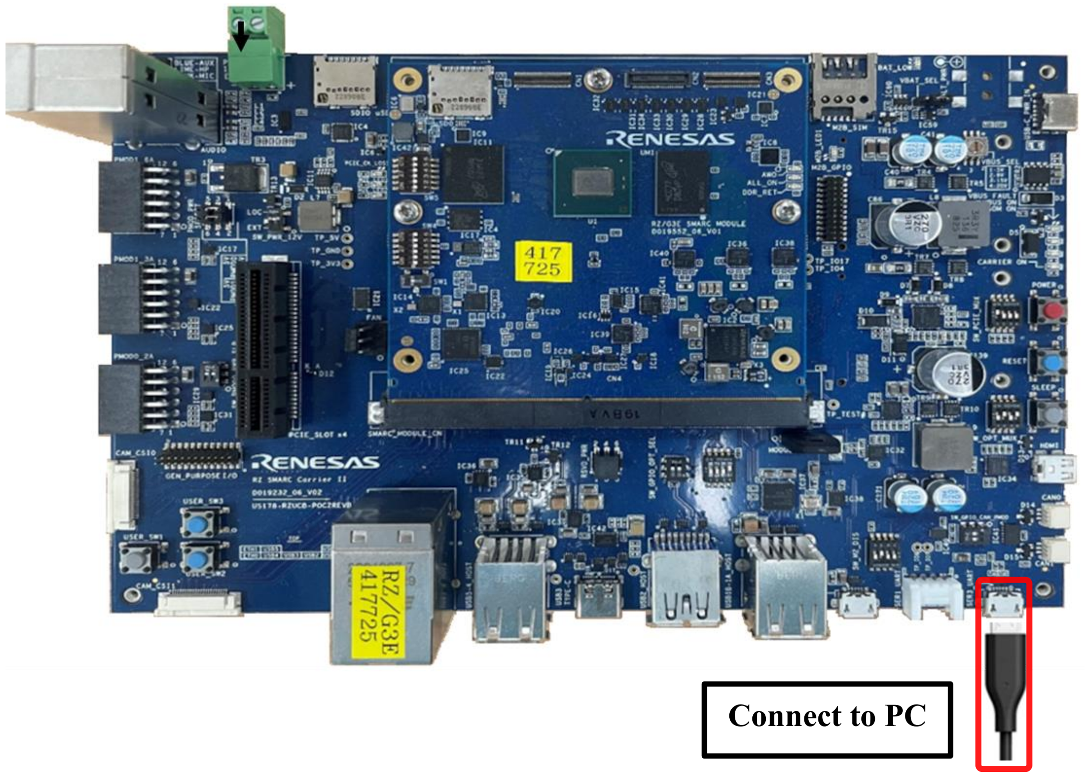


Figure 7. Connecting console for debug

4.3.3 How to set CN15 for the HDMI output (Optional)

This step is optional. Skip this step when you do not use the HDMI output.

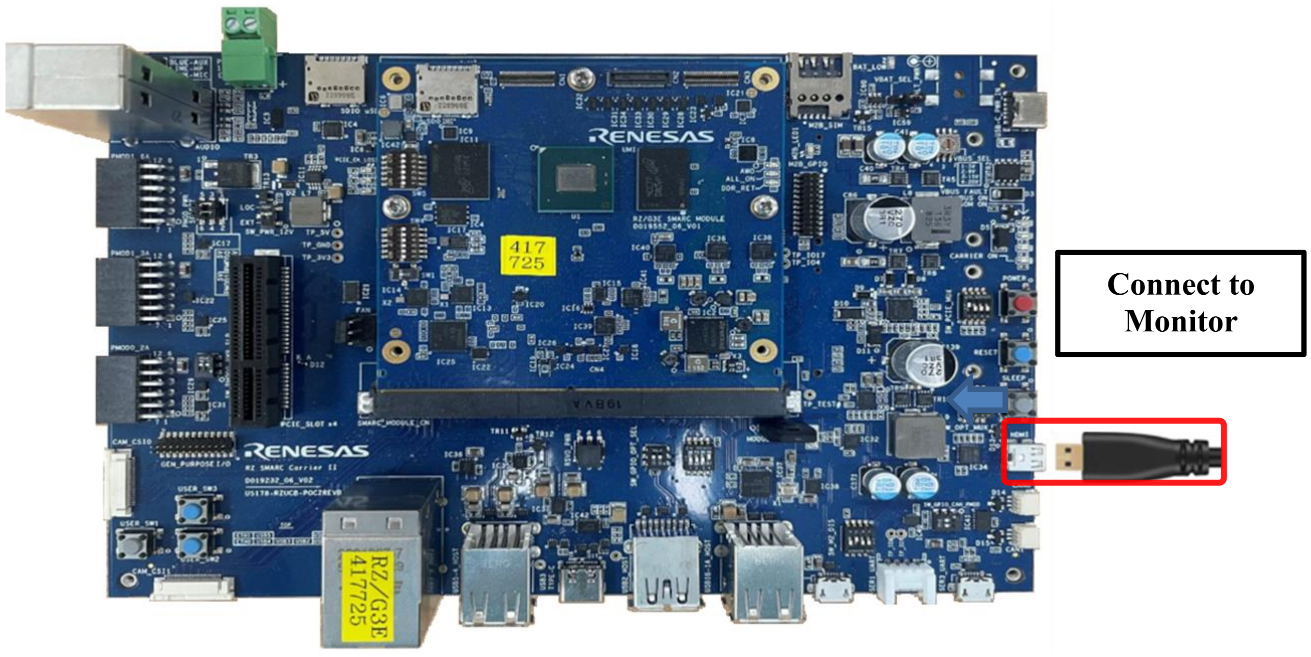


Figure 8. How to connect the module

4.4 Startup Procedure for RZ/G3E

This section explains how to power on the evaluation board and set up Tera Term for controlling the target board's serial console.

4.4.1 Power Supply

1. Set the RZ/G3E SMARC EVK boot switch selector ("SW_MODE") to the boot mode which you want.
   Please check Section 4.2 and 4.3 for details such as other boot mode, DIP switch, and jumper pin settings.
2. Attach the PD capable USB Type-C cable to the USB Type-C port ("USB-C_PWR_IN").
3. Connect the other end of the Type-C cable to the source that is the USB-C power adapter.
4. Check that two LEDs ("VBUS ON", "SOM ON", and "CARRIER ON") are illuminated.
5. Push the "Power" red button for 1 msec and visually check that LED ("CARRIER_PWR_ON") is illuminated.

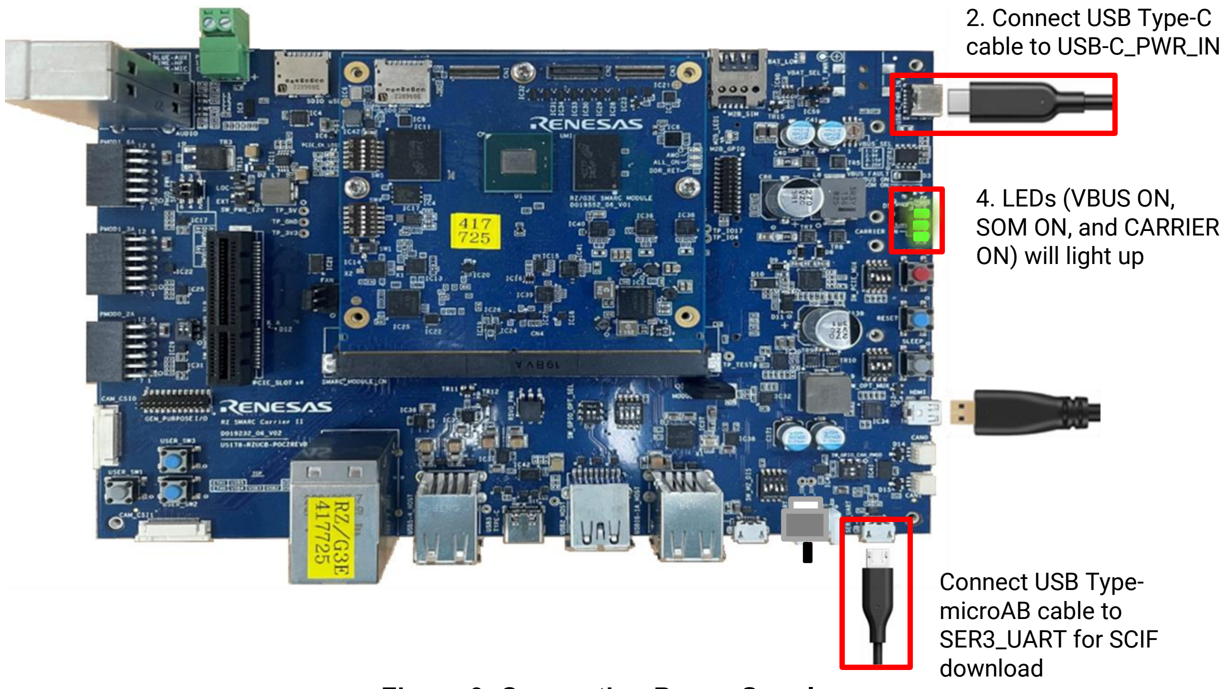

2. Connect USB Type-C cable to USB-C_PWR_IN
4. LEDs (VBUS ON, SOM ON, and CARRIER ON) will light up
Connect USB Type-microAB cable to SER3_UART for SCIF download

Figure 9. Connecting Power Supply

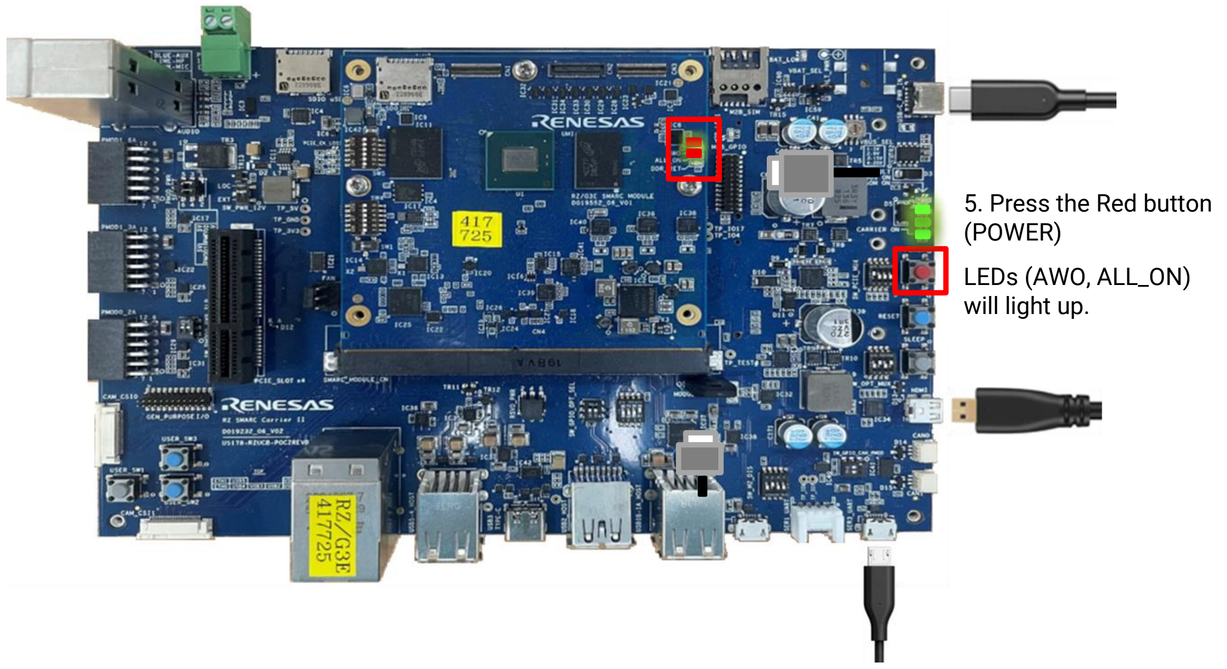

5. Press the Red button (POWER)
LEDs (AWO, ALL_ON) will light up.

Figure 10. Power ON

4.4.2 Building files to write

The evaluation board uses the files in the following table as the boot loaders. The boot loader files are generated by the build instructions in section 2.1. Once the building is finished, please copy these files to a PC which runs serial terminal software.

Table 6. File names of Boot loader

Board	File name of Boot loader
RZ/G3E Evaluation Board	• bl2_bp_spi-smarc-rzg3e.srec • fip-smarc-rzg3e.srec

4.4.3 Settings

Connect the board to a host PC using the USB serial cable according to the Release Note.

1. To set the board to SCIF download mode, set "SW_MODE" according to the switch setting in Section 4.3.1.
2. Power the RZ/G3E SMARC EVK according to Section 4.4.
3. Launch the terminal software and select "File" > "New Connection" to set the connection on the software.
4. Select "Setup" > "Serial port" to configure settings related to serial communication protocols on the software. Set the serial communication protocol settings on a terminal software as follows.

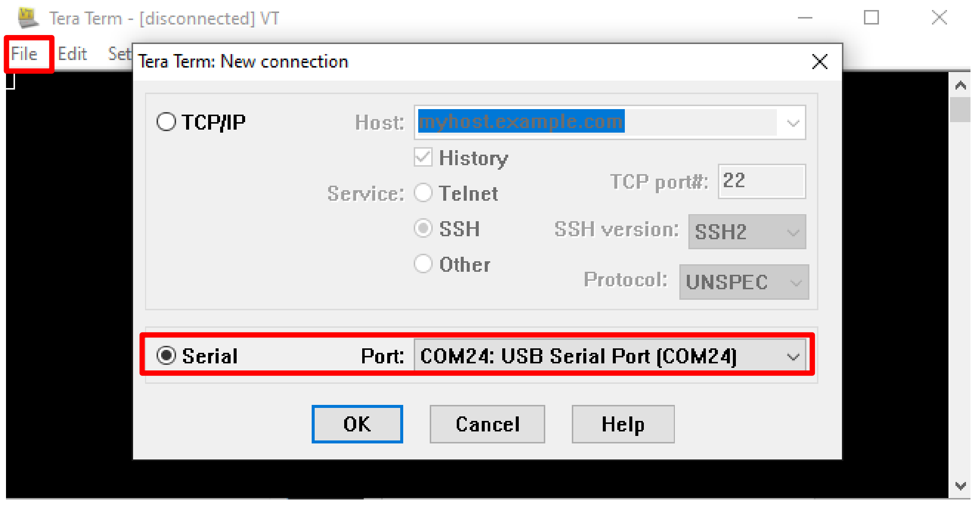
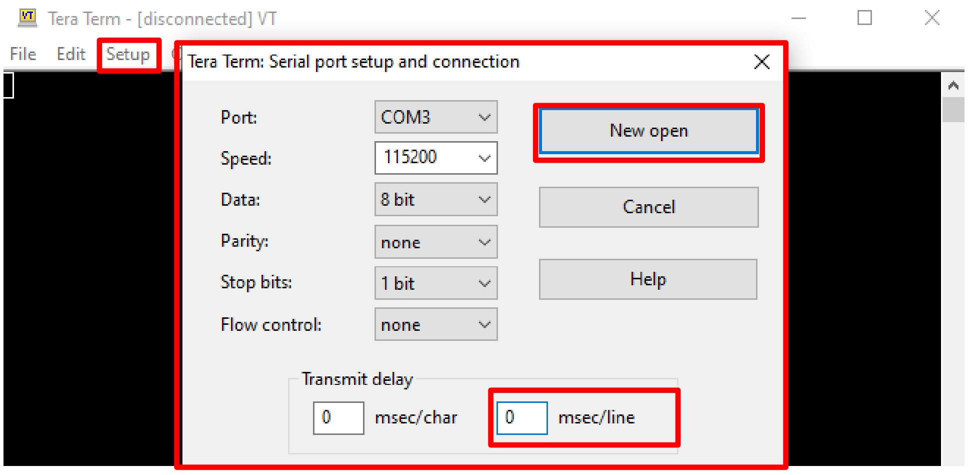


Figure 11. Set "Serial"

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



Tera Term: Serial port setup and connection

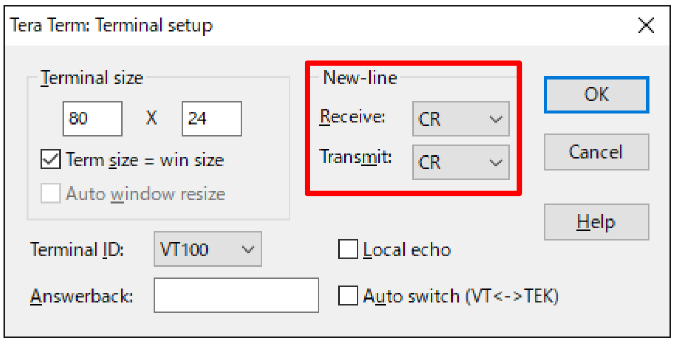
Port:	COM3
Speed:	115200
Data:	8 bit
Parity:	none
Stop bits:	1 bit
Flow control:	none

Transmit delay
0 msec/char
0 msec/line

Figure 12. Terminal Setting

Select the "Setup" > "Terminal" to set the new-line code.

- New-line:"CR" or "AUTO"



Tera Term: Terminal setup

Terminal size
80 X 24
☑ Term size = win size
☐ Auto window resize

New-line
Receive: CR
Transmit: CR

Terminal ID: VT100
Answerback:
☐ Local echo
☐ Auto switch (VT<->TEK)

After completing all settings, press the RESET button and display the messages below on the terminal.

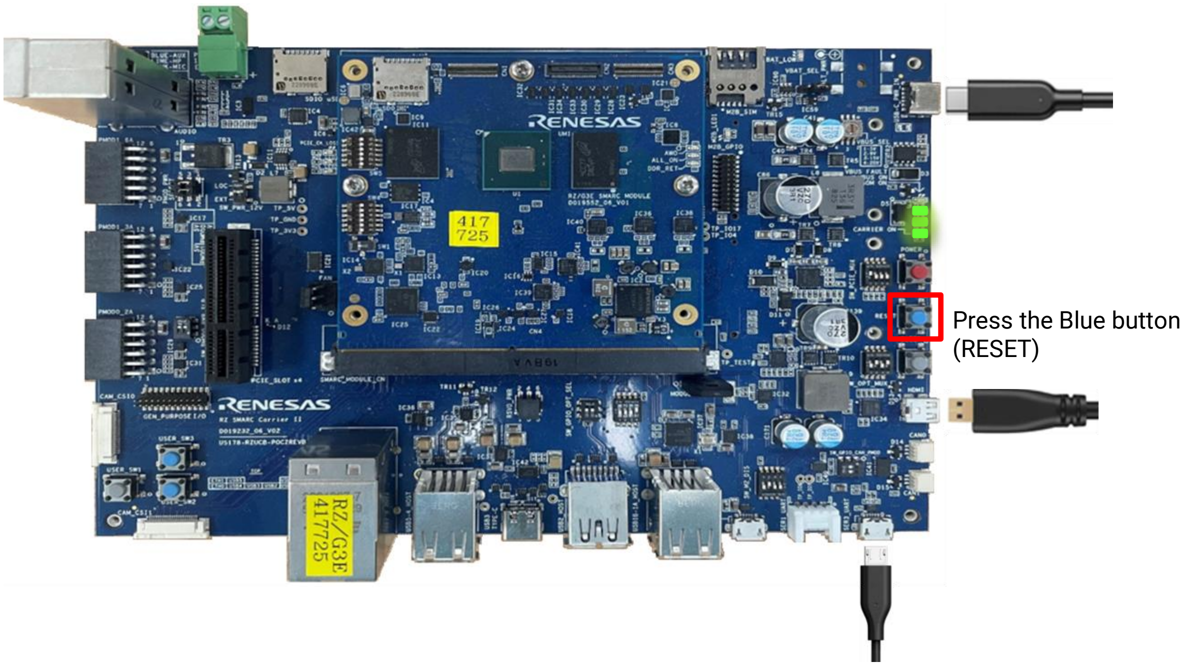
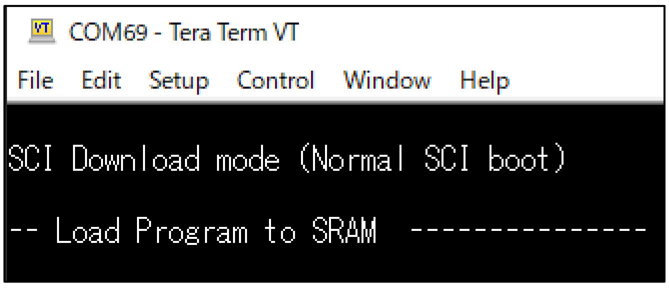


Figure 13. Press the RESET button



```
SCI Download mode (Normal SCI boot)
-- Load Program to SRAM ---------------
```

Figure 14. Terminal log of SCIF Download mode

4.5 Download Flash Writer to RAM

This section explains how to download the Flash Writer from a Windows PC.

(1) Turn on the evaluation board

Turn on the board by pressing the RESET button. The messages below are shown on the terminal.

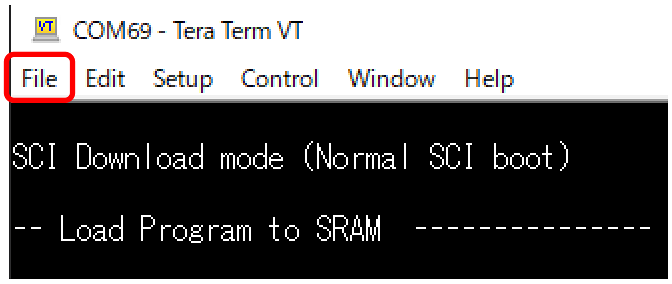
```
SCI Download mode (Normal SCI boot)

-- Load Program to SRAM  ---------------
```

(2) Send Flash Writer using terminal software

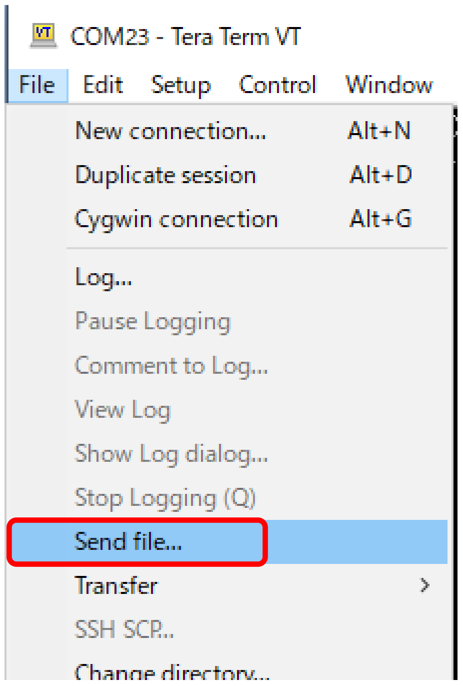
Send an image of Flash Writer (Flash_Writer_SCIF_RZG3E_EVK_LPDDR4X.mot) using terminal software. Below is a sample procedure with Tera Term.

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

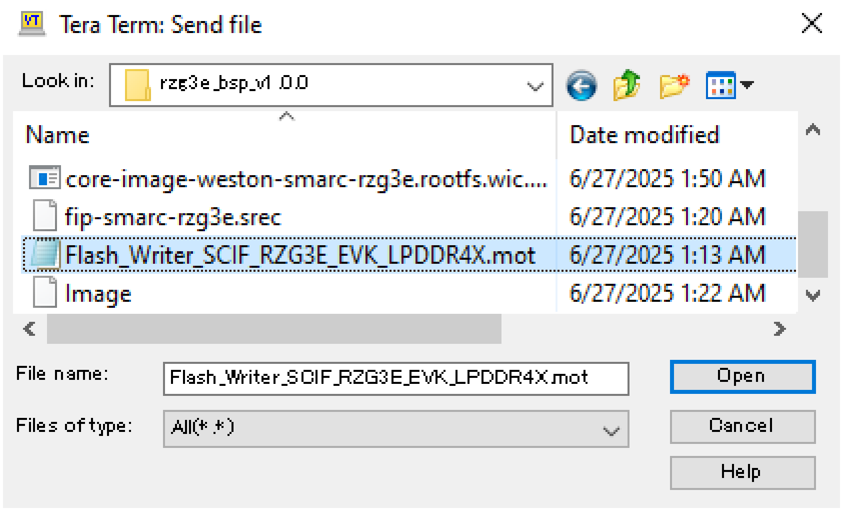


```
SCI Download mode (Normal SCI boot)

-- Load Program to SRAM  --------------
```



- Then, select an image of Flash Writer (Flash_Writer_SCIF_RZG3E_EVK_LPDDR4X.mot) and click “Open” button.


Tera Term: Send file

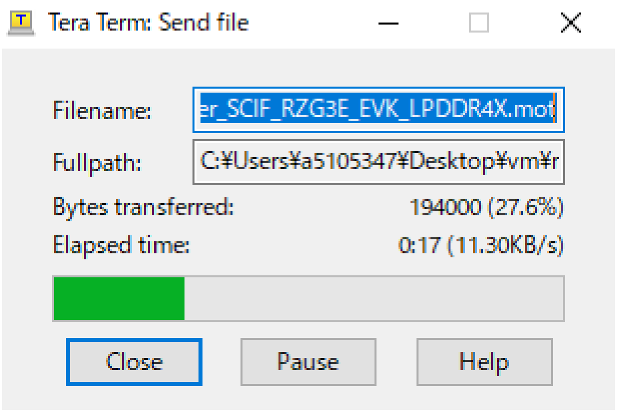
Look in: rzg3e_bsp_v1.0.0

Name	Date modified
core-image-weston-smarc-rzg3e.rootfs.wic....	6/27/2025 1:50 AM
fip-smarc-rzg3e.srec	6/27/2025 1:20 AM
Flash_Writer_SCIF_RZG3E_EVK_LPDDR4X.mot	6/27/2025 1:13 AM
Image	6/27/2025 1:22 AM

File name: Flash_Writer_SCIF_RZG3E_EVK_LPDDR4X.mot

Files of type: All(*.*)

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


Tera Term: Send file

Filename: er_SCIF_RZG3E_EVK_LPDDR4X.mot

Fullpath: C:¥Users¥a5105347¥Desktop¥vm¥r

Bytes transferred: 194000 (27.6%)

Elapsed time: 0:17 (11.30KB/s)

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

```
Flash writer for RZ/G3E Series
 Product Code : RZ/G3E
>
```

4.6 Write the Bootloader

For the boot operation, two boot loader files are required to be written to the target board using Flash Writer commands. The specific boot loader files and their respective addresses are detailed in Table 7. Please follow the instructions in this section.

Table 7. Address for sending each loader binary file

File name	Address to load to RAM	Address to save to ROM
bl2_bp_spi-smarc-rzg3e.srec	H'8003600	H'00000
fip-smarc-rzg3e.srec	H'00000	H'60000

(1) Change serial port speed

Before writing these boot loader files, you need to adjust the Flash Writer's transfer rate. Change it from the default 115200 bps to a high-speed setting of 921600 bps using the "SUP" command within the Flash Writer.

```
>SUP
Scif speed UP
Please change to 921.6Kbps baud rate setting of the terminal.
```

After "SUP" command, change the serial communication protocol speed from 115200bps to 921600bps as well by following the steps described in Section 4.4.3, and push the enter key.

(2) XLS2 command to write boot loader

Next, "XLS2" command of Flash Writer 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.

Run the following command and addresses.

```
>XLS2
===== Qspi writing of RZ/G3E Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
 Dialog : AT25QL128A
Program Top Address & Qspi Save Address
===== Please Input Program Top Address ============
  Please Input : H'8003600

===== Please Input Qspi Save Address ===
  Please Input : H'0
please send ! ('.' & CR stop load)
```

Send the data of "bl2_bp_spi-smarc-rzg3e.srec" from terminal software after the message "please send !" is shown.

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

```
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
======= Qspi  Save Information  =================
 SpiFlashMemory Stat Address : H'00000000
 SpiFlashMemory End Address  : H'00021170
===========================================================
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”.

(3) XLS2 command to write boot loader again

Write another loader file by using XLS2 command again.

```
>XLS2
===== Qspi writing of RZ/G3E Board Command =============
Load Program to Spiflash
Writes to any of SPI address.
 Dialog : AT25QL128A
Program Top Address & Qspi Save Address
===== Please Input Program Top Address ============
  Please Input : H'0

===== Please Input Qspi Save Address ===
  Please Input : H'60000
Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ('.' & CR stop load)
```

Send the data of “fip-smarc-rzg3e.srec” from terminal software after the message “please send !” is shown.

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

```
Erase SPI Flash memory...
Erase Completed
Write to SPI Flash memory.
======= Qspi  Save Information  =================
 SpiFlashMemory Stat Address : H'00060000
 SpiFlashMemory End Address  : H'0014738E
===========================================================
```

```
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”.

(4) Change back the serial port speed

After successfully writing the two loader files, change the serial communication protocol speed from 921600 bps to 115200 bps by following the steps outlined in Section 4.4.3.

(5) Power off the evaluation board

Finally, power off the evaluation board by pressing the POWER button.

4.7 Change Boot Mode and set U-Boot variables

To set the board to SPI Boot mode, set the DIP switch “SW_MODE” as below:

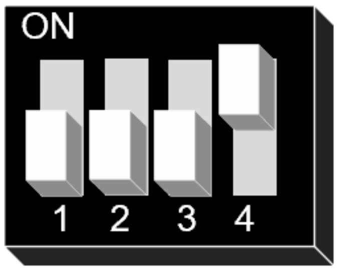


Figure 15. SPI Boot mode

Turn on the power switch of the board. The following log will be displayed, and a countdown will begin. Please press the Enter key at this point.

```
U-Boot 2024.07 (Jun 13 2025 - 15:56:19 +0000)

CPU:   Renesas Electronics CPU rev 1.0
Model: Renesas SMARC EVK based on r9a09g047e57
DRAM:  3.9 GiB
Core:  45 devices, 18 uclasses, devicetree: separate
MMC:   mmc@15c00000: 0, mmc@15c10000: 1, mmc@15c20000: 2
Loading Environment from MMC... Reading from MMC(0)... OK
In:    serial@11c01400
Out:   serial@11c01400
Err:   serial@11c01400
Hit any key to stop autoboot:  0
=>
```

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
=>
```

Now, the evaluation board is ready to boot Linux.

5. Booting and Running Linux

This chapter explains how to boot Linux on the evaluation board.

5.1 Set micro-SD card

Set micro-SD1 card to slot of the evaluation board.

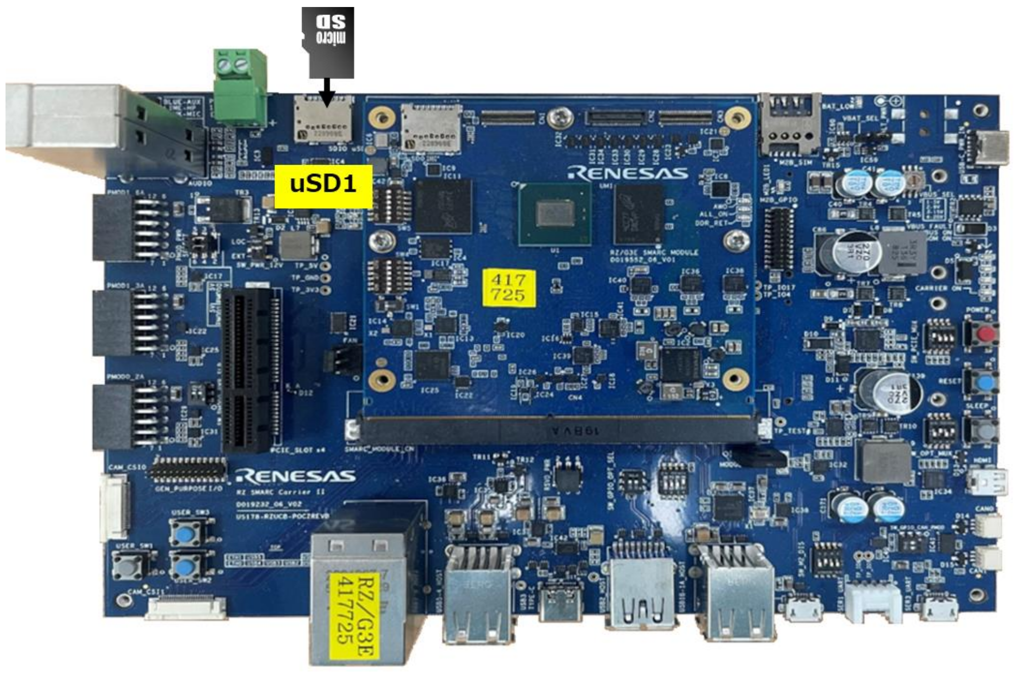


Figure 16. Set micro-SD card to micro-SD1 slot

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

5.2 Power on the board and Startup Linux

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

```
U-Boot 2024.07 (Jun 13 2025 - 15:56:19 +0000)

CPU:   Renesas Electronics CPU rev 1.0
Model: Renesas SMARC EVK based on r9a09g047e57
DRAM:  3.9 GiB
Core:  45 devices, 18 uclasses, devicetree: separate
MMC:   mmc@15c00000: 0, mmc@15c10000: 1, mmc@15c20000: 2
Loading Environment from MMC... Reading from MMC(0)... OK
In:    serial@11c01400
Out:   serial@11c01400
Err:   serial@11c01400

Hit any key to stop autoboot:  0
## Resetting to default environment
switch to partitions #0, OK
mmc1 is current device
24373760 bytes read in 974 ms (23.9 MiB/s)
69535 bytes read in 5 ms (13.3 MiB/s)
Moving Image from 0x48080000 to 0x48200000, end=499d0000
```

```
## Flattened Device Tree blob at 48000000
   Booting using the fdt blob at 0x48000000
Working FDT set to 48000000
   Loading Device Tree to 0000000057fec000, end 0000000057ffff9e ... OK
Working FDT set to 57fec000

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x412fd050]
[    0.000000] Linux version 6.1.107-cip28-yocto-standard (oe-user@oe-host)
 :
 :
Poky (Yocto Project Reference Distro) 5.0.8 smarc-rzg3e ttySC0

smarc-rzg3e login: root
root@smarc-rzg3e:~#
```

Note: As a reference, this log is written with the BSP v1.0.0.

5.3 Shutdown the Board

To power down the system, follow the instructions below.

(1) Run shutdown command

Run shutdown command on the console as below. After that, the shutdown sequence will start.

```
root@smarc-rzg3e:~# shutdown -h now
```

Note: Run this command during the power-off sequence on rootfs.

(2) Confirm that the power is off and turn off the power switch

After executing the shutdown command, you will see the "reboot: Power down" message. Then, turn off the "POWER" switch.

6. Building the SDK

This chapter explains how to build the Software Development Kit (SDK).

To build the SDK, follow the instructions below after completing steps (1) – (6) of Section 2.1.

The SDK builds custom applications outside of the Yocto environment, even on a completely different PC. The commands below will generate an "installer" that you can use to install the SDK on the same PC or a different one.

For building general applications:

```
$ cd ~/rzg3e_bsp_v${PACKAGE_VERSION}/build
$ MACHINE=smarc-rzg3e bitbake <target> -c populate_sdk
```

The resulting SDK installer will be in "**`build/tmp/deploy/sdk/`**".

To run the installer, execute the following command:

```
$ sudo sh rz-vlp-glibc-x86_64-<target>-cortexa55-smarc-rzg3e-toolchain-5.0.8.sh
```

7. Application Building and Running

This chapter explains how to build an application with SDK and run it on the evaluation board.

7.1 Make an application

Here is an example of how to make an application running on BSP. The following steps will generate the "Hello World" sample application.

Note that you must build (bitbake) a core image for the target and prepare SDK before making an application.

7.1.1 How to extract SDK

(1) Install toolchain on a Host PC:

```
$ sudo sh rz-vlp-glibc-x86_64-<target>-cortexa55-smarc-rzg3e-toolchain-5.0.8.sh
```

Note:

sudo is optional in case user wants to extract SDK into a restricted directory (such as: /opt/).

If the installation is successful, the following messages will appear:

```
/mnt/host$ sudo sh ./build/tmp/deploy/sdk/ rz-vlp-glibc-x86_64-<target>-cortexa55-.\
smarc-rzg3e-toolchain-5.0.8.sh
Poky (Yocto Project Reference Distro) SDK installer version x.x.xx
==================================================================
Enter target directory for SDK (default: /opt/poky/x.x.xx):
You are about to install the SDK to "/opt/poky/x.x.xx". Proceed [Y/n]? Y
Extracting SDK.................................................................
..........................................................done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the envir
onment setup script e.g.
$ . /mnt/work/sdk/environment-setup-cortexa55-poky-linux
```

(2) Set up cross-compile environment:

```
$ source /<Location in which SDK is extracted>/environment-setup-cortexa55-poky-linux
```

Note:

User needs to run the above command once for each login session.

```
$ source /opt/poky/x.x.xx/environment-setup-cortexa55-poky-linux
```

7.1.2 How to build Linux application

(1) Make a directory for the application on the Linux host PC.

```
$ mkdir ~/hello_apl
$ cd ~/hello_apl
```

(2) Make the following two files (an application file and Makefile) in the directory for the application. Here, the application is made by automake and autoconf.

- main.c

```
#include <stdio.h>
/* Display "Hello World" text on terminal software */
/* RZ/G3E Linux Start-up Guide */
/* R01US0682EJ0094 Rev.0.94 Page 29 of 42 */
/* Jul 24, 2024 */
int main(int argc, char** argv)
{
printf("\nHello World\n");
return 0;
}
```

- Makefile

```
APP = linux-helloworld
SRC = main.c
all: $(APP)
CC ?= gcc
# Options for development
CFLAGS = -g -O0 -Wall -DDEBUG_LOG
$(APP):
  $(CC) -o $(APP) $(SRC) $(CFLAGS)
install:
  install -D -m755 $(APP) $(DESTDIR)/home/root/$(APP)
clean:
  rm -rf $(APP)
```

(3) Make the application by the generated makefile.

```
$ make
```

```
renesas@49d1d5882435:/mnt/host/linux-helloworld$ make
aarch64-poky-linux-gcc -mtune=cortex-a55 -fstack-protector-strong -D_FORTIFY_SOURCE=
2 -Wformat -Wformat-security -Werror=format-security --sysroot=/opt/poky/3.1.31/sysroo
ts/aarch64-poky-linux -o linux-helloworld main.c -g -O0 -Wall -DDEBUG_LOG
renesas@49d1d5882435:/mnt/host/linux-helloworld$ ls -l
total 24
-rwxr-xr-x 1 renesas renesas 15320 May 10 23:32 linux-helloworld
-rw-r—r—1 renesas renesas 148 Sep 24 2020 main.c
-rw-r—r—1 renesas renesas 250 Sep 24 2020 Makefile
```

After that, confirm that the executable application (the sample file name is “linux-helloworld”) is generated in the hello_apl directory. The application must be cross compiled for Aarch64.

The sample application could be written by the following procedure. The application should be stored in the ext3 partition.

```
$ sudo mount /dev/sdb2 /media/
$ cd /media/usr/bin
$ sudo cp ~/hello_apl/linux-helloworld .
$ sudo chmod +x linux-helloworld
```

Notes: 1. “sdb2” (above in red) may depend on using system.
 2. is an optional directory name to store the application.

7.2 Run a sample application

Power on the evaluation board kit and start the system. After booting, run the sample application with the following command.

```
smarc-rzg3e login: root
root@smarc-rzg3e:~# /usr/bin/linux-helloworld

Hello World
root@smarc-rzg3e:~#
```

8. Appendix

8.1 Docker

This section explains how to build the BSP with Docker enabled and how to obtain and run the "hello-world" image.

(1) Add layers to enable Docker

After completing step (4) in section 2.1, run the following commands. Once executed, proceed until step (6) in section 2.1 to build the BSP with Docker enabled.

```
$ cd ~/rzg3e_bsp_v${PACKAGE_VERSION}/build
$ bitbake-layers add-layer ../meta-openembedded/meta-networking
$ bitbake-layers add-layer ../meta-openembedded/meta-filesystems
$ bitbake-layers add-layer ../meta-virtualization
```

(2) Enable Docker in local.conf

To enable Docker, add the following line to "~/rzg3e_bsp_v${PACKAGE_VERSION}/build/conf/local.conf".

```
DISTRO_FEATURES:append = " virtualization docker"
```

(3) Boot the evaluation board

Follow chapters 4 and 5, boot the evaluation board, and proceed to the next step.

(4) Start Docker Service

Firstly, launch the docker service with the following command.

```
root@smarc-rzg3e:~# systemctl start docker
```

(5) Check Docker on the evaluation board (Optional)

Optionally run the following commands to display the status of the Docker Daemon and its version.

```
root@smarc-rzg3e:~# systemctl status docker.service
root@smarc-rzg3e:~# docker version
```

(6) Hello world

Run the commands below to get the docker image of hello-world and run the image.

```
root@smarc-rzg3e:~# docker pull hello-world
root@smarc-rzg3e:~# docker run hello-world
Hello from Docker!
```

8.2 How to boot from eMMC

This section explains how to boot Linux from eMMC mode.

8.2.1 Writing Bootloader for eMMC Boot

(1) Set SW_MODE as SCIF Download mode and Use Flash Writer (please refer 4.3)

(2) Write BL2 Loader to eMMC.

Enter "EM_W" command and specify eMMC partition, address and program top address.

```
>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 : 8003600
Work RAM (H'00020000-H'000FFFFF) Clear....
please send ! ('.' & CR stop load)
```

Then, send BL2 Loader "bl2_bp_mmc-smarc-rzg3e.srec" in the same way as Flash Writer.

(3) Write BL2 Loader to eMMC.

Enter "EM_W" command and specify eMMC partition, address and program top address.

```
>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 :300
Please Input Program Start Address : 0
Work RAM (H'00020000-H'000FFFFF) Clear....
please send ! ('.' & CR stop load)
```

Then, send FIP Loader "fip-smarc-rzg3e.srec" in the same way as Flash Writer.

(4) Modify EXT_CSD registers.

Enter “EM_SECSD“ command and set the following values.

```
>EM_SECSD
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_SECSD
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
```

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/G3E Evaluation Board Kit

File name	Partition to save to eMMC	Address to save to eMMC	Address to load to RAM
bl2_bp_sip-smarc-rzg3e.srec	1	00000001	8003600
fip-smarc-rzg3e.srec	1	00000300	00000

8.2.2 Prepare a microSD card for writing image to eMMC

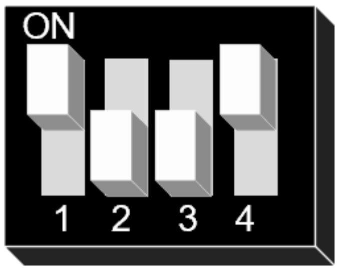
Create a microSD card according to step 2 and write rootfs to the 2nd section.

```
cd $WORK/build/tmp/deploy/images/smarc-rzg3e
sudo mount /media/user/rootfs <device name>
sudo cp core-image-minimal-smarc-rzg3e.rootfs.tar.gz /media/user/rootfs/root/
sudo umount /media/user/rootfs
```

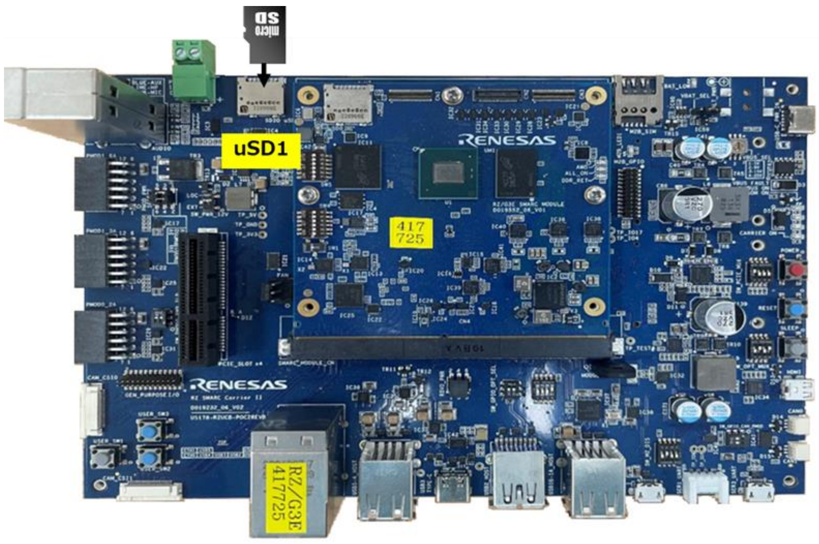
<device name> may be "/dev/sd*2" (* is a letter representing the physical drive). E.g., "/dev/sda2"

8.2.3 Write rootfs to eMMC

(1) Set the DIP switch “SW_MODE” as follows:



(2) Insert the SD card created in 4.3 into the uSD1 Card Slot.



(3) Power ON

Turn on the Board Power SW first until 3rdLED from Carrier Power LEDs and 2 LEDs from SOM Power LEDs are turned on.

(4) Set up U-Boot environmental variables.

```
=> env default -a
## Resetting to default environment
=> setenv bootargs 'console=ttySC0,115200 rw rootwait earlycon root=/dev/mmcblk1p2'
=> setenv bootcmd 'mmc dev 1;ext4load mmc 1:2 0x48080000 /boot/Image;ext4load mmc 1:2
0x48000000 /boot/r9a09g047e57-smarc.dtb;booti 0x48080000 - 0x48000000'
=> saveenv
Saving Environment to MMC... Writing to MMC(0)... OK
=>
```

(5) Make partition tables on eMMC with fdisk command:

```
root@smarc-rzg3e:~# fdisk /dev/mmcblk0

The number of cylinders for this disk is set to 1942528.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): o
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that the previous content
won't be recoverable.


The number of cylinders for this disk is set to 1942528.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

```

Command (m for help): n
Partition type
   p   primary partition (1-4)
   e   extended
p
Partition number (1-4): 1
First sector (16-124321791, default 16): (Push the enter key)
Using default value 16
Last sector or +size{,K,M,G,T} (16-124321791, default 124321791): +500M

Command (m for help): n
Partition type
   p   primary partition (1-4)
   e   extended
p
Partition number (1-4): 2
First sector (1024016-124321791, default 1024016): (Push the enter key)
Using default value 1024016
Last sector or +size{,K,M,G,T} (1024016-124321791, default 124321791): (Push the enter
 key)
Using default value 124321791

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

(6) Format eMMC as ext4.

```
root@smarc-rzg3e:~# mkfs.ext4 /dev/mmcblk0p1
root@smarc-rzg3e:~# mkfs.ext4 /dev/mmcblk0p2
```

(7) Write rootfs to eMMC.

```
root@smarc-rzg3e:~# mount /dev/mmcblk0p2 /mnt/
root@smarc-rzg3e:~# tar xf /root/core-image-minimal-smarc-rzg3e.tar.gz -C /mnt/
root@smarc-rzg3e:~# umount /dev/mmcblk0p2
```

8.2.4 Setting U-boot and boot from eMMC

(1) Set SW_MODE

Set SW_MODE as eMMC boot mode (please refer 8.2.3(1)):

(2) Connect devices

Refer to the figure at 4.2.

(3) Turn on the Board Power SW

Turn on the Board Power SW first until 3rdLED from Carrier Power LEDs and 2 LEDs from SOM Power LEDs are turned on.

(4) Press any key to stop autoboot within 3 seconds after u-boot starts.

```
NOTICE:  BL2: v2.10.5(release):2.10.5/rzg3e_1.2.0-dirty
NOTICE:  BL2: Built : 15:26:33, Jun 13 2025
… …
… …
Err: serial@11c01400
Net:
Error: ethernet@15C40000 No valid MAC address found.
```

```
Error: ethernet@15C30000 No valid MAC address found.
Error: ethernet@15C30000 No valid MAC address found.
Error: ethernet@15C40000 No valid MAC address found.
No ethernet found.
Hit any key to stop autoboot: 0
=>
=>
```

(5) Set up U-Boot environmental variables for eMMC boot.

```
=> env default -a
## Resetting to default environment
=> setenv bootargs 'console=ttySC0,115200 rw rootwait earlycon root=/dev/mmcblk0p2'
=> setenv bootcmd 'mmc dev 0;ext4load mmc 0:2 0x48080000 /boot/Image;ext4load mmc 0:2
0x48000000' /boot/r9a09g047e57-smarc.dtb;booti 0x48080000 - 0x48000000'
=> saveenv
Saving Environment to MMC... Writing to MMC(0)... OK
=>
```

(6) Do not push any key within 3 seconds.

```
NOTICE:  BL2: v2.10.5(release):2.10.5/rzg3e_1.2.0-dirty
NOTICE:  BL2: Built : 15:26:33, Jun 13 2025
… …
… …
Err: serial@11c01400
Net:
Error: ethernet@15C40000 No valid MAC address found.
Error: ethernet@15C30000 No valid MAC address found.
Error: ethernet@15C30000 No valid MAC address found.
Error: ethernet@15C40000 No valid MAC address found.
No ethernet foundNo ethernet found.
Hit any key to stop autoboot: 0
switch to partitions #0, OK
mmc1 is current device
39703040 bytes read in 1890 ms (20 MiB/s)
46420 bytes read in 5 ms (8.9 MiB/s)
Moving Image from 0x48080000 to 0x48200000, end=499d0000
## Flattened Device Tree blob at 48000000
   Booting using the fdt blob at 0x48000000
Working FDT set to 48000000
   Loading Device Tree to 0000000057fec000, end 0000000057ffff9e ... OK
Working FDT set to 57fec000
```

(7) Login by "root".

```
Starting kernel ...
[ 0.000000] Booting Linux on physical CPU 0x0000000000 [0x412fd050]
[ 0.000000] Linux version 6.1.107-cip28-yocto-standard (oe-user@oe-host) (aarch64-poky
-linux-gcc
(GCC) 13.3.0, GNU ld (GNU Binutils) 2.42.0.20240723) #1 SMP PREEMPT Fri Mar 14 15:21:2
2 UTC 2025
[ 0.000000] Machine model: Renesas SMARC EVK based on r9a09g047e57
[ 0.000000] earlycon: scif0 at MMIO 0x0000000011c01400 (options '115200n8')
[ 0.000000] printk: bootconsole [scif0] enabled
[ 0.000000] efi: UEFI not found.
[ 0.000000] Reserved memory: created CMA memory pool at 0x0000000058000000, size 256 M
iBcd ..
… …
… …
Welcome to Poky (Yocto Project Reference Distro) 5.0.8 (scarthgap)!
… …
… …
Poky (Yocto Project Reference Distro) 5.0.8 smarc-rzg3e ttySC0
smarc-rzg3e login: root
WARNING: Poky is a reference Yocto Project distribution that should be used for
testing and development purposes only. It is recommended that you create your
own distribution for production use.
root@smarc-rzg3e:~#
```

8.3 How to boot from eSD

This section explains how to boot Linux from eSD mode.

Regarding the eSD (Embedded SD) booting, please note the following:

- The eSD boot procedure using microSD card described in this guide is for evaluation purposes only.
- If you use the eSD boot, please implement the eSD on your board according to the standard "SD Specification Part 1 eSD Addendum (version 2.10)".
- The reboot command cannot be used when using the eSD boot procedure using microSD card described in this guide.

8.3.1 Writing WIC File to SD Card

(1) Expand disk image with bmaptool.

For Linux PC:

```
$ sudo bmaptool copy --bmap core-image-minimal-smarc-rzg3e.wic.bmap core-image-minimal
-smarc-rzg3e.wic.gz <device name>
$ sync
```

<device name> may be "/dev/sd*" (* is a letter representing the physical drive). E.g., "/dev/sda"

For Windows PC:

Copy "core-image-minimal-smarc-rzg3e.wic.gz" to Windows PC.

Write the image (core-image-minimal-smarc-rzg3e.wic.gz) to an SD card using your favorite tool.

E.g.,

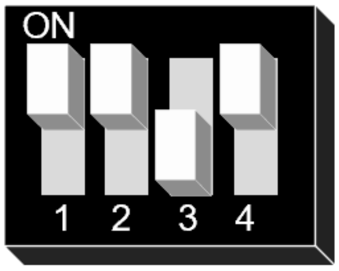
balenaEtcher (https://etcher.balena.io)

Win32 Disk Imager (https://sourceforge.net/projects/win32diskimager/)

Win32 Disk Imager requires the decompressed file (=core-image-minimal-smarc-rzg3e.wic)

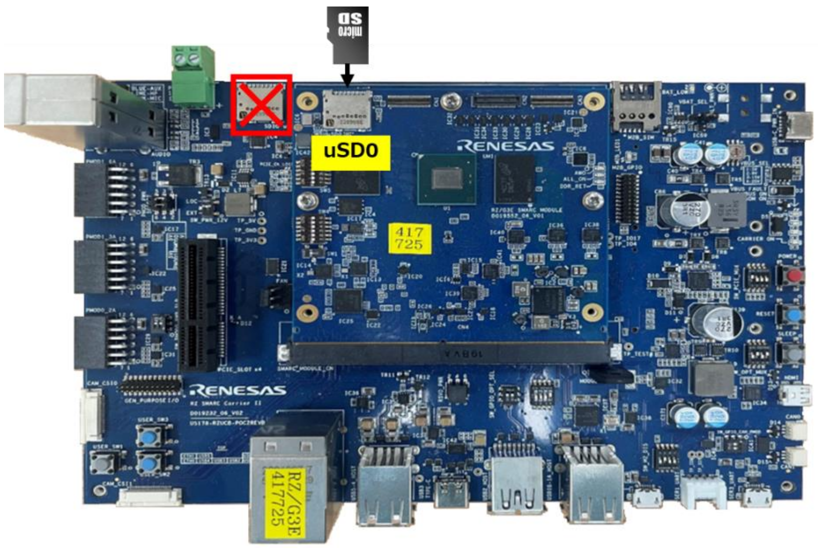
8.3.2 Change eSD Boot Mode

(1) Set the DIP switch "SW_MODE" as follows:



8.3.3 Setting U-boot and boot from eSD

Insert the SD card created in 'APPENDIX 2-2' into the uSD0 Card Slot.



Connect devices if necessary (see also the figure at 'Step 3-1').

Turn on the Board Power SW first until 3rd LED from Carrier Power LEDs and 2 LEDs from SOM Power LEDs are turned on.

(1) Set up U-Boot environmental variables.

After power on, press any key to stop autoboot within 3 seconds after u-boot starts.

```
NOTICE:  BL31: v2.10.5(release):2.10.5/g3e_1.1.0-dirty
NOTICE:  BL31: Built : 14:47:04, Feb  7 2025
… …
… …
Err:   serial@11c01400
Net:
Error: ethernet@15C40000 address not set.
Error: ethernet@15C30000 address not set.
Error: ethernet@15C30000 address not set.
Error: ethernet@15C40000 address not set.
No ethernet found.
Hit any key to stop autoboot:  0
=>
=>
```

Set up U-Boot environmental variables.

```
=> env default -a
## Resetting to default environment
=> setenv bootargs 'console=ttySC0,115200 rw rootwait earlycon root=/dev/mmcblk0p2'
=> setenv bootcmd 'mmc dev 0;ext4load mmc 0:2 0x48080000 /boot/Image;ext4load mmc 0:2
0x48000000 /boot/r9a09g047e57-smarc.dtb;booti 0x48080000 - 0x48000000'
=> saveenv
Saving Environment to MMC... Writing to MMC(0)... OK
=>
```

(2) Reset or Re-power ON and login by "root".

Do not push any key within 3 seconds.

```
NOTICE:  BL2: v2.10.5(release):2.10.5/rzg3e_1.2.0-dirty
NOTICE:  BL2: Built : 15:26:33, Jun 13 2025
… …
… …
U-Boot 2024.07 (Jun 13 2025 - 15:56:19 +0000)

CPU:   Renesas Electronics CPU rev 1.0
Model: Renesas SMARC EVK based on r9a09g047e57
DRAM:  3.9 GiB
Core:  45 devices, 18 uclasses, devicetree: separate
MMC:   mmc@15c00000: 0, mmc@15c10000: 1, mmc@15c20000: 2
Loading Environment from MMC... Reading from MMC(0)... OK
In:    serial@11c01400
Out:   serial@11c01400
Err:   serial@11c01400
Net:
Error: ethernet@15C40000 No valid MAC address found.

Error: ethernet@15C30000 No valid MAC address found.

Error: ethernet@15C30000 No valid MAC address found.

Error: ethernet@15C40000 No valid MAC address found.
No ethernet found.

Hit any key to stop autoboot:  0
switch to partitions #0, OK
mmc1 is current device
24373760 bytes read in 976 ms (23.8 MiB/s)
69535 bytes read in 6 ms (11.1 MiB/s)
Moving Image from 0x48080000 to 0x48200000, end=499d0000
## Flattened Device Tree blob at 48000000
   Booting using the fdt blob at 0x48000000
Working FDT set to 48000000
   Loading Device Tree to 0000000057fec000, end 0000000057ffff9e ... OK
Working FDT set to 57fec000

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x412fd050]
[    0.000000] Linux version 6.1.107-cip28-yocto-standard (oe-user@oe-host) (aarch64-p
oky-linux-gcc (GCC) 13.3.0, GNU ld (GNU Binutils) 2.42.0.20240723) #1 SMP PREEMPT Thu
Jun 19 12:42:30 UTC 2025
[    0.000000] Machine model: Renesas SMARC EVK based on r9a09g047e57
[    0.000000] earlycon: scif0 at MMIO 0x0000000011c01400 (options '115200n8')
[    0.000000] printk: bootconsole [scif0] enabled
[    0.000000] efi: UEFI not found.
[    0.000000] Reserved memory: created CMA memory pool at 0x0000000058000000, size 25
6 MiB
… …
… …
Welcome to Poky (Yocto Project Reference Distro) 5.0.8 (scarthgap)!
… …
… …
Poky (Yocto Project Reference Distro) 5.0.8 smarc-rzg3e ttySC0
```

```
smarc-rzg3e login: root

WARNING: Poky is a reference Yocto Project distribution that should be used for
testing and development purposes only. It is recommended that you create your
own distribution for production use.
root@smarc-rzg3e:~#
```

9. Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun. 30, 2025	-	First edition issued.

Website and Support

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

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