

# RZ/G Verified Linux Package for 64bit kernel Version 1.0.6

R01TU0311EJ0106  
Rev. 1.06  
Apr 23, 2021

## Release Note for HTML5 update1

### Introduction

This release note describes the contents, building procedures for HTML5 (Gecko) and important points of the RZ/G Verified Linux Package for 64bit kernel (hereinafter referred to as “VLP64”). In this release, Linux packages for HTML5 is preliminary and provided AS IS with no warranty.

If you need information to build Linux BSPs without a GUI Framework of HTML5, please refer to “RZ/G Verified Linux Package for 64bit kernel Version 1.0.6 Release Note”.

This release note is the update version of “RZ/G Verified Linux Package for 64bit kernel Version 1.0.6 Release Note for HTML5 (Document Number: R01TU0311EJ0103)”. Please confirm “REVISION HISTORY” in the page 21 to check the update points.

### Contents

1. Release Items .....	2
2. Build environment.....	4
3. Building Instructions of a BSP .....	6
3.1 Setup the Linux Host PC to build images.....	6
3.2 Building images to run on the board .....	8
3.3 Building SDK .....	12
4. Components .....	13
5. Restrictions .....	14
6. Notes .....	15

## 1. Release Items

- **Name and version**

RZ/G Verified Linux Package for 64bit kernel

Version 1.0.6 (hereinafter referred to as “VLP v1.0.6”)

- **Distribution method**

Please visit the site below and create an account to download the packages. This site is for the entire RZ Family which includes the RZ/G series. Basic packages of VLP v1.0.6 which are listed in **Table 1** can be downloaded.

**RZ Family:**

<https://www.renesas.com/products/microcontrollers-microprocessors/rz-arm-based-high-end-32-64-bit-mpus>

- **Target board**

Hoperun Technology HiHope RZ/G2H platform (hihope-rzg2h) Rev 4.0

Hoperun Technology HiHope RZ/G2M platform (hihope-rzg2m) Rev 4.0 (\*)

Hoperun Technology HiHope RZ/G2N platform (hihope-rzg2n) Rev 4.0

Silicon Linux RZ/G2E evaluation kit (ek874) Rev C

(\*) There are 2 versions of RZ/G2M LSI devices. There are version 3.0 and 1.3 (hereinafter referred to as “RZ/G2M v3.0” and “RZ/G2M v1.3”).

- **Verified functions**

Linux BSP

- Linux Kernel
- Linux Drivers
- Graphics Libraries

GUI Framework

- Qt (LGPL version)
- Gecko

- **Evaluation vs Product Version**

Because the package that contains the VLP64 Yocto recipes also contains the multimedia packages, there are two release versions: Evaluation (eva) and Product (pro). Please note that both of these packages have the same exact functionality. The only difference is that when you execute an application that uses the evaluation version of the multimedia libraries, operation will automatically be stopped after a few hours. The product version does not have this time limitation. To acquire the product version, please contact Renesas to start the formal process of acquiring those releases.

- **File contents**

VLP64 is delivered by the files listed in **Table 1**.

**Table 1. RZ/G Verified Linux Package for 64bit kernel****Basic packages**

File	Description
rzg2_bsp_eva_v106.tar.gz (Evaluation version) (65MB) rzg2_bsp_pro_v106.tar.gz (Product version) (65MB)	Yocto recipe packages (including multimedia package)
oss_pkg_v106.7z (7.3GB)	Open source software packages  <b>See the Note below before download</b>
r01tu0277ej0107-rz-g.pdf	Build instruction without HTML5
r01tu0278ej0106-rz-g.pdf	Component list
r01tu0279ej0105-rz-g.pdf	Documents describing booting method and the required settings of bootloader

**Basic packages for HTML5 (Gecko) environment**

File	Description
rzg2_bsp_gecko_v106.tar.gz (614KB)	Yocto recipe packages for HTML5
oss_pkg_gecko_v106.7z (1.9GB)	Open source software packages for HTML5 (Gecko 68)
buildtool_gecko_v106.tar.xz (690MB)	Build tool for HTML5
r01tu0311ej0106-rz-g.pdf	This document

The above packages are downloaded with the different file names like as “REN\_rzg2\_bsp\_eva\_v10x.tar.gz\_ESW\_2021xxxx.gz”. When you use the packages, please rename them to the above file names.

Note) Open source software packages contain all source codes of OSSs except for Linux kernel. These are the same versions of OSSs used when VLP64 was verified.

If you are just evaluating VLP64 and RZ/G series, open source software packages are not mandatory to use.

Usually, all the software can be built without using these files if your build machine is connected to the Internet.

Open source software packages are required for an “offline” environment. The word “offline” means an isolated environment which does not connect to any network. VLP64 can always build images in this “offline” environment by using these packages without affected from changes of original repositories of OSSs. Also, this “offline” environment always reproduces the same images as the images which were verified by Renesas. Note that if you build without using open source software packages, there are possibilities to use different source codes than Renesas used due to the implicit changes of the repositories of OSSs.

Most bootable images that VLP64 supports can be built on an “offline” environment. Please refer to **2. Build environment**.

## 2. Build environment

Figure 1 shows an overall constitution of the recommended environment of VLP64. This environment uses the equipment and the software listed in Table 2. Please refer to “RZ/G Verified Linux Package Start-Up Guide” for details about setting up the environment.

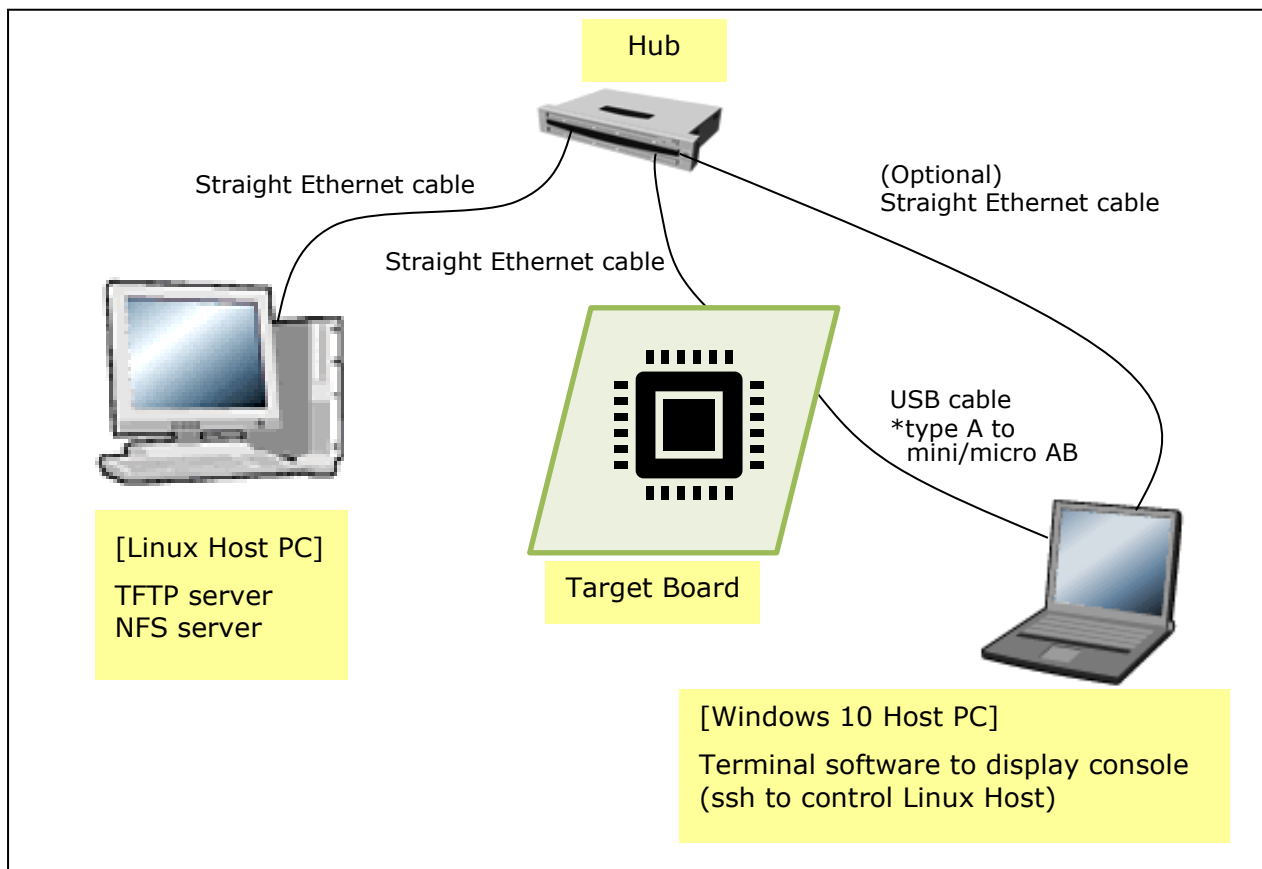


Figure 1. Recommended Environment

**Table 2. Equipment and Software Necessary for Developing Environments of RZ/G Linux Platform**

Equipment	Description
Linux Host PC	Used as build/debug environment 100GB free space on HDD is necessary
OS	<b>Ubuntu 16.04 LTS</b> 64 bit OS must be used.
TFTP server	Used for downloading the Linux kernel to the board
NFS server	Used for mounting rootfs via NFS
Windows Host PC	Used as debug environment, controlling with terminal software
OS	Windows 10
Terminal software	Used for controlling serial console of the target board Tera Term (latest version) is recommended Available at <a href="https://tssh2.osdn.jp/index.html.en">https://tssh2.osdn.jp/index.html.en</a>
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: <ul style="list-style-type: none"> <li><a href="http://www.ftdichip.com/Drivers/VCP.htm">http://www.ftdichip.com/Drivers/VCP.htm</a> (for ek874)</li> <li><a href="https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers">https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers</a> (for hihope-rzg2h, hihope-rzg2m, and hihope-rzg2n)</li> </ul> Please install VCP Driver corresponding to the target board.

Note) Build may fail if Ubuntu 14, 18 or 20 is used.

Most bootable images VLP64 supports can be built on an “offline” environment.

The word “offline” means an isolated environment which does not connect to any network. Since VLP64 includes all necessary source codes of OSS except for the Linux kernel, VLP64 can always build images in this “offline” environment without affected from changes of repositories of OSS. Also, this “offline” environment reproduces the same images as the images which were verified by Renesas.

Below images can be built “offline”.

- core-image-weston
- core-image-weston-sdk

Below are not available in the “offline” environment. Please connect your Linux Host PC to the internet.

- Preparing a Linux Host PC

### 3. Building Instructions of a BSP

This chapter describes the instructions to build the Board Support Package (hereinafter referred to as “BSP”). Before starting the build, run the steps in the section 3.1 on the Linux Host PC to install packages used for building the BSP.

#### 3.1 Setup the Linux Host PC to build images

This section describes how to setup the Linux Host PC before starting the build.

##### (1) Install packages

Run the commands below on the Linux Host PC to install packages used for building the BSP.

```
$ sudo apt-get update
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \
build-essential chrpath socat cpio python python3 python3-pip python3-pexpect \
xz-utils debianutils iputils-ping libstdc++11-dev xterm p7zip-full autoconf2.13
$ sudo apt install clang llvm clang-3.9 llvm-3.9
$ sudo apt-get install build-essential libasound2-dev libcurl4-openssl-dev \
libdbus-1-dev libdbus-glib-1-dev libgconf2-dev libgtk-3-dev libgtk2.0-dev \
libpulse-dev libx11-xcb-dev libxt-dev nasm nodejs openjdk-8-jdk-headless \
python-dbus python-dev python-pip python-setuptools software-properties-common \
unzip uuid wget xvfb yasm zip
```

Please refer to the URL below for detailed information:

- <https://www.yoctoproject.org/docs/2.4.3/yocto-project-qs/yocto-project-qs.html>

Run the commands below and set the user name 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"
```

##### (2) Install GCC 7 and switch the default GCC

Run the commands below to install GCC7 used for building the BSP.

```
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc-7 g++-7
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-5 10 \
--slave /usr/bin/g++ g++ /usr/bin/g++-5
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 20 \
--slave /usr/bin/g++ g++ /usr/bin/g++-7
```

After the above step, GCC 7 is set as the default GCC compiler. In case to build VLP64 without Gecko or SDK, run this command to switch the default GCC compiler to GCC 5.

```
$ sudo update-alternatives --config gcc
```

After running the above command, the following text is displayed. Please input the number corresponding to GCC5. In the following example, “1” corresponds to GCC5. In case to switch the default GCC compiler again, run the above command again.

```
There are 2 choices for the alternative gcc (providing /usr/bin/gcc).
```

Selection	Path	Priority	Status
* 0	/usr/bin/gcc-7	20	auto mode
1	/usr/bin/gcc-5	10	manual mode
2	/usr/bin/gcc-7	20	manual mode

```
Press <enter> to keep the current choice[*], or type selection number:
```

### (3) Decompress the build tools to build Gecko

Run the commands below to decompress the build tools to build Gecko. The tools will be decompressed to your home directory. Once this step is finished, it is not necessary to run this step every time you build.

```
$ cd
$ tar xf ~/buildtool_gecko_v106.tar.xz
$ ln -sf .rustup .multirust
$ echo 'export PATH="$HOME/.cargo/bin:$PATH"' >> .profile
```

## 3.2 Building images to run on the board

This section describes how to start the build.

**Copy all files obtained from Renesas into your home directory prior to the steps below.**

### (1) Create a working directory (user\_work), and decompress Yocto recipe packages

Run the commands below. Please replace “*eva*” with “*pro*” for the product version. **In case that ECC is not required when you use RZ/G2H, please refer to the Section 5 Restrictions - (5) u-boot.**

```
$ mkdir ~/user_work
$ cd ~/user_work
$ tar zxvf ~/rzg2_bsp_eva_v106.tar.gz
$ tar zxvf ~/rzg2_bsp_gecko_v106.tar.gz
```

Note) Please note that your build environment must have 100GB of free hard drive space in order to complete the 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.

Note) VLP64 is set video output to LVDS as default setting. In case using HDMI, please refer to the **Section 6 Notes - (2) Video output.**

### (2) Execute the copy script for proprietary software

Run the scripts below. Both product and evaluation version use the same script.

```
$ cd ~/user_work/meta-rzg2
$ sh docs/sample/copyscript/copy_proprietary_softwares.sh ../proprietary
```

### (3) Setup a build environment

Run the commands below. The environment to build is set by the source command.

```
$ cd ~/user_work
$ source poky/oe-init-build-env
```

### (4) Prepare the default configuration files for the target board

Run the commands below. 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

```
$ cd ~/user_work/build
$ cp ../meta-rzg2/docs/sample/conf/board/linaro-gcc/*.conf ./conf/
$ cp ./conf/local_gecko.conf ./conf/local.conf
$ cp ./conf/bblayers_gecko.conf ./conf/bblayers.conf
```



**(5) Decompress OSS files to “build” directory**

Run the commands below. This step is not mandatory and able to go to the step (7) in case the “offline” environment is not required. All OSS packages will be decompressed with this '7z' command.

```
$ cd ~/user_work/build
$ 7z x ~/oss_pkg_v106.7z
$ 7z x ~/oss_pkg_gecko_v106.7z
```

Note) If this step is omitted and BB\_NO\_NETWORK is set to “0” in next step, all source codes will be downloaded from the repositories of each OSS via the internet when running bitbake command. Please note that if you do not use an “offline” environment, a build may fail due to the implicit changes of the repositories of OSS.

**(6) Download Linux kernel source code**

Run the commands below.

```
$ cd ~/user_work/build
$ bitbake linux-renesas -c fetch
```

After the above procedure is finished, the “offline” environment is ready. If you want to prevent network access, please change the line in the “~/user\_work/build/conf/local.conf” as below:

```
BB_NO_NETWORK = "1"
```

To change BB\_NO\_NETWORK from “0” to “1”.

**(7) Start a build**

Run the commands below to start a build. Building an image can take up to a few hours depending on the user’s host system performance.

```
$ bitbake core-image-weston
```

After the build is successfully completed, a similar output will be seen, and the command prompt will return.

```
NOTE: Tasks Summary: Attempted 7427 tasks of which 16 didn't need to be rerun and all succeeded.
```

All necessary files listed in **Table 3** will be generated by the bitbake command and will be located in the **build/tmp/dep/lay/images** directory.

Note) When you launch the browser application on the board, please refer to **6. Notes - (12)**.

Table 3. Image files

	Linux kernel	root filesystem	Boot loader
<b>RZ/G2H</b>	Image-hihope-rzg2h.bin	<image name>- hihope- rzg2h.tar.bz2	u-boot-elf-hihope-rzg2h.srec bootparam_sa0.srec bl2-hihope-rzg2h.srec bl31-hihope-rzg2h.srec tee-hihope-rzg2h.srec cert_header_sa6.srec
<b>RZ/G2M v3.0 (*) v1.3 (*)</b>	Image-hihope-rzg2m.bin	<image name>- hihope- rzg2m.tar.bz2	u-boot-elf-hihope-rzg2m.srec bootparam_sa0.srec bl2-hihope-rzg2m.srec bl31-hihope-rzg2m.srec tee-hihope-rzg2m.srec cert_header_sa6.srec
<b>RZ/G2N</b>	Image-hihope-rzg2n.bin	<image name>- hihope- rzg2n.tar.bz2	u-boot-elf-hihope-rzg2n.srec bootparam_sa0.srec bl2-hihope-rzg2n.srec bl31-hihope-rzg2n.srec tee-hihope-rzg2n.srec cert_header_sa6.srec
<b>RZ/G2E</b>	Image-ek874.bin	<image name>- ek874.tar.bz2	u-boot-elf-ek874.srec bootparam_sa0.srec bl2-ek874.srec bl31-ek874.srec tee-ek874.srec cert_header_sa6.srec

<image name> will be the name used in the step (7).

#### Device tree files

	Type 1	Type 2	Type 3	Type 4
<b>RZ/G2H</b>	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
<b>RZ/G2M v1.3 (*)</b>	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
<b>RZ/G2M v3.0 (*)</b>	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
<b>RZ/G2N</b>	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
<b>RZ/G2E</b>	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.** For the booting method and the required settings, please refer to the “RZ/G2 Reference Boards Linux Start-up Guide”.

(\*) 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
-----------------------------------

Note) The dtb files listed in the Table 3 **cannot be used for the early version of boards.** See the section **6. Notes - (1)**. In case you use the hihope-rzg2m board which has RZ/G2M v3.0 LSI, the dtb files listed in the Table 3 can be used because the board is the latest version.

### 3.3 Building SDK

To build Software Development Kit (SDK), run the commands below after the steps (1) – (7) of section 3.2 are finished. Before starting to build SDK, please switch the default GCC compiler to GCC 5. Please refer to the step (2) of section 3.1.

The SDK allows you to build custom applications outside of the Yocto environment, even on a completely different PC. The results of the commands below are ‘installer’ that you will use to install the SDK on the same PC, or a completely different PC.

For building general applications:

```
$ cd ~/user_work/build
$ cp ../meta-rzg2/docs/sample/conf/board/linaro-gcc/*.conf ./conf/
$ bitbake core-image-weston-sdk -c populate_sdk
```

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

The resulting SDK installer will be located in **build/tmp/deploy/sdk/**

The SDK installer will have the extension .sh

To run the installer, you would execute the following command:

```
$ sudo poky-glibc-x86_64-core-image-weston-sdk-aarch64-toolchain-2.4.3.sh
```

Please refer to “RZ/G2 Group Application Note” and “Verified Linux Package Start-up Guide” for the usage of SDK.

## 4. Components

Compared to the RZ/G Verified Linux Package (hereinafter referred to as “VLP”) for 32bit kernel and previous version of VLP64, this version uses different version of Linux kernel and supports newer HTML5 rendering engine. Please also refer to the “Component list” for details.

**Table 4. Versions of commonly used components**

Components	VLP for 32bit kernel v2.1.7	VLP for 64bit kernel v1.0.4	VLP for 64bit kernel v1.0.6
Linux kernel	4.4.222-cip45	4.19.106-cip21	4.19.140-cip33
GCC	7.2.1 (Linaro GCC 7.2-2017.11)	7.3.1 (Linaro GCC 7.3-2018.05)	7.3.1 (Linaro GCC 7.3-2018.05)
glibc	2.19 (CIP Core Jessie)	2.19 (CIP Core Jessie)	2.28 (CIP Core Buster)
busybox	1.22.0 (CIP Core Jessie)	1.22.0 (CIP Core Jessie)	1.30.1 (CIP Core Buster)
openssl	1.0.2o	1.0.2o	1.1.1d (CIP Core Buster)
gststreamer1.0	1.12.2	1.12.2	1.12.2
wayland	1.13.0	1.13.0	1.13.0
weston	2.0.0	2.0.0	2.0.0
python3	3.5.3	3.5.3	3.5.3
Qt (LGPL version)	5.6.3	5.6.3	5.6.3
Gecko	60	68	68
Docker	-	17.06	17.06

Note) VLP includes Debian 10 (Buster) based CIP Core Packages indicated as “(CIP Core Buster)” in the **Table 4**. Note that the Buster version of packages are enabled in the default settings instead of the Jessie version of them from this release. These packages can be replaced with other versions of packages. Please refer to **6. Notes - (13)** to enable them. VLP is verified with the default CIP Core Packages. For more technical information, please contact Renesas.

## 5. Restrictions

### (1) Drivers

In this release, the below drivers can be used but are not fully tested on all MPUs.

- Bluetooth
- Wifi
- IPMMU

### (2) USB camera 3.0

RZ/G2E cannot stream with higher resolution than Full HD when you use a camera with USB 3.0.

### (3) CSI40

Disable CSI40 in RZ/G2H, N and G2M v3.0 by default due to SW limitation.

### (4) Wifi and Bluetooth

Low performance while using 2.4GHz Wifi/Bluetooth and USB 3.0 device in Hihope RZ/G2H, M, and N at the same time due to noise. If using USB3.0 device, should connect to 5GHz Wifi network.

### (5) u-boot

In case that ECC is disabled, RZ/G2H cannot bootup. If ECC is not required when you use RZ/G2H, please run the commands below to apply the patch after **the step (1) in the section 3.2**.

```
$ cd ~/user_work/meta-rzg2
$ patch -p1 < ../extra/0001-u-boot-add-a-revert-patch-to-fix-hangup-issue-\
when-n.patch
```

## 6. Notes

### (1) Bootloader/Linux kernel

The dtb files listed in the **Table 3** 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-hihopec-rzg2m-**rev2**.dtb (main board only)
- Image-r8a774a1-hihopec-rzg2m-**rev2**-ex.dtb (main + sub board)
- Image- r8a774a1-hihopec-rzg2m-**rev2**-ex-idk-1110wr.dtb (main + sub board + LVDS-IF)
- Image-r8a774a1-hihopec-rzg2m-**rev2**-ex-mipi-2.1.dtb (main + sub board + MIPI/CSI2 cameras)

#### HiHope RZ/G2N board:

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

#### Silicon Linux RZ/G2E board (EK874):

- Please refer to **(8) GPU** in this section.

Note) Board revision is printed below the Hoperun logo on boards.

### (2) Video output

VLP64 is set video output to LVDS as default setting. In case to use HDMI as a video output, please apply the patch with these commands.

```
$ cd ~/user_work/meta-rzg2
$ patch -p1 < ../extra/0001-BSPv1.0.6-Add-HDMI-support-for-RZ-G2.patch
```

After applying above patch, please note to below points.

#### Audio

In case 48kHz audio, please set SW2404 to P1 side in HiHope Rev4 Boards of RZ/G2H, M, and N.

#### Bluetooth

Firmware of Bluetooth is integrated into the kernel. Therefore, this step is not necessary.

```
hciattach /dev/ttySC1 texas 3000000
```

Instead, this step is required.

```
rftkill unblock bluetooth
```

#### SATA

SATA interface on HiHope Rev4 Boards of RZ/G2H, N is enabled by setting switches as below. (SATA cannot be supported in the RZ/G2N Rev2 board due to HW limitation).

- SW1001-7 on main board: OFF
- SW43 on sub board: ON

### (3) Weston

Due to the specification of open source software (Weston 2.0.0), it is not recommended to resize application windows. Please consider designing the application to use fixed sized windows.

### (4) Video playback

Due to the specification of open source software (GStreamer and others) and drivers, multiple GStreamer pipelines with hardware scale cannot run.

Also, below formats of video are not supported.

- NV61
- YUV420
- YUV422
- YUV444
- H.264, 80Mbps

### (5) ECC

The ECC function for DRAM has two modes: 8bit data/5bit ECC mode and 64bit data/8bit ECC mode. 8bit data/5bit ECC mode can be evaluated by the following method. When applying the ECC function to products or need other details including the method to enable 64bit data/8bit ECC mode, please contact Renesas.

Enable the function by changing the lines below in the `local.conf`.

Disable:

```
# MACHINE_FEATURES_append = " ecc"  
# ECC_MODE = "Partial"
```

Enable:

```
MACHINE_FEATURES_append = " ecc"  
ECC_MODE = "Full"
```

This sets 8bit data/5bit ECC mode for all DRAM region. After building, please replace all images including boot loaders.

### (6) SDHI

Early revision of EK874 boards cannot detect insertion of SD card. Please plugged in a card before turning on the power.

### (7) VIN

One camera input is enabled in default settings. Two camera inputs can be enabled in RZ/G2H, N and M v3.0 . Please refer to the Video Capture Driver User's Manual that is included in the RZ/G2 Group BSP Manual Set for more details.

VIN on early revision of HiHope RZ/G2M and N boards fails to work. Please use newer boards in case VIN is necessary.



**(8) GPU**

The dtb files listed in the **Table 3** 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 mmc 0:1 0x480
00000 Image-r8a774c0-ek874.dtb ; fi ; booti 0x48080000 - 0x48000000'
```

**(9) Docker**

Docker is disabled in the default settings of VLP64. To enable Docker, please uncomment the below line inside the file `~/user_work/build/conf/local.conf` before building images.

```
#MACHINE_FEATURES_append = " docker"
```

**(10) Wifi**

Wifi is disabled in default settings but modules necessary for Wifi functions are installed into rootfs. In case Wifi is necessary, please enable it from a console as below.

```
$ rfkill list
```

If this command shows “Soft blocked: yes”, run “unblock” command like this.

```
$ rfkill unblock wlan
```

Then, continue below.

```
$ connmanctl
connmanctl> scan wifi
connmanctl> services
connmanctl> agent on
connmanctl> connect <network_name>
    <input password>
connmanctl> quit
```

You may need to retry the “connect” command few times.

Note that some settings relating about radio waves should be adjusted according to the laws of each region. Please refer to general information in books and websites about Linux networking.

### (11) USB Video Class

USB Video Class (UVC) driver is not installed with the default settings of VLP64 due to its big size.

In case UVC devices such as USB cameras are necessary, please install the driver by adding the line below to `local.conf`.

```
IMAGE_INSTALL_append = " kernel-module-uvcdvideo "
```

### (12) HTML5 (Gecko)

When you launch the browser application on the board, please run the below command on a terminal software or a console on the board.

```
$ GDK_BACKEND=wayland firefox <content>
```

Please replace `<content>` with an URL or a path of a local content like the below examples:

```
$ GDK_BACKEND=wayland firefox https://www.renesas.com/us/en/
$ GDK_BACKEND=wayland firefox /home/root/index.html
```

### (13) CIP Core Packages

VLP includes Debian 10 (Buster) based CIP Core Packages indicated as “(CIP Core Buster)” in the **Table 4** and is enabled by the default settings. In this release, these packages can be replaced with other versions of packages. “Buster-full” is the default CIP Core Packages and VLP is verified with it. There are two more types, Buster-limited and Jessie. Please follow the below 4 procedures to enable them. Please refer to the **Table 5** for the versions of the packages.

Note that network access is required to start the build process when you enable these packages except for Buster-full which is set as the default setting.

CIP Core Packages are going to be maintained by the Civil Infrastructure Platform project. For more technical information, please contact Renesas.

#### 1. Buster-full (default):

The following lines are added as default in the `local.conf`.

```
CIP_MODE = "Buster-full"
BBMASK .= "|perl_debian"
```

#### 2. Buster-limited:

This setting is to enable the limited packages from CIP Core Buster. Comment out the lines of Buster-full and add a line to enable Buster-limited in the `local.conf`.

```
# CIP_MODE = "Buster-full"
# BBMASK .= "|perl_debian"
CIP_MODE = "Buster-limited"
```

### 3. Jessie:

This setting enables Debian 8 (jessie) based CIP Core Packages. After the step **3.2 (1)**, run the commands below to build HTML5 (Gecko) with Debian 8 (jessie) based CIP Core Packages.

```
$ cd ~/user_work
$ patch -p1 < ./extra/vBuster-to-vJessie_gecko_v106.patch
```

After the step **3.2 (4)**, comment out the lines of Buster-full and add the lines to enable Jessie in the `local.conf`.

```
# CIP_MODE = "Buster-full"
# BBMASK .= "|perl_debian"
CIP_MODE = "Jessie"
BBMASK .= "|openssl_debian|nodejs_10.15.3|libgit2_0.27.9"
```

Note) When you want to build the BSP for HTML5 (Gecko) with the CIP Core Packages except for Jessie after the above steps, please create a new working directory and run the steps from **3.2 (1)**.

### 4. No CIP Core Packages:

If the CIP Core Packages are unnecessary, comment out and add the following lines to disable CIP Core Packages in the `local.conf`.

```
# CIP_MODE = "Buster-full"
CIP_MODE = "none"
```

Note) The above 4 settings disable GPLv3 packages as default. In case the GPLv3 packages are required, please comment out the following line in the `local.conf`.

```
# INCOMPATIBLE_LICENSE = "GPLv3 GPLv3+"
```

By building the BSP, the packages will be replaced as below in the table.

**Table 5. Versions of Packages**

<b>Package</b>	<b>Buster-limited (non-GPLv3)</b>	<b>Buster-full (non-GPLv3)</b>	<b>Buster-full (allow GPLv3)</b>	<b>Jessie (non-GPLv3)</b>	<b>No CIP Core (allow GPLv3)</b>
busybox	1.30.1 (buster)	1.30.1 (buster)	1.30.1 (buster)	1.22 (jessie)	1.24.1
openssl	1.1.1d (buster)	1.1.1d (buster)	1.1.1d (buster)	1.0.1t (jessie)	1.0.2o
glibc	2.28 (buster)	2.28 (buster)	2.28 (buster)	2.19 (jessie)	2.26
binutils	-	-	2.31.1 (buster)	-	-
coreutils	6.9	6.9	8.30 (buster)	6.9	8.27
gnupg	1.4.7	1.4.7	2.2.12 (buster)	1.4.7	2.2.0
libassuan	2.4.3	2.5.2 (buster)	2.5.2 (buster)	2.4.3	2.4.3
libpam	1.3.0	1.3.1 (buster)	1.3.1 (buster)	1.3.0	1.3.0
libgcrypt	1.7.3	1.8.4 (buster)	1.8.4 (buster)	1.7.3	1.7.3
libunistring	0.9.7	0.9.10 (buster)	0.9.10 (buster)	0.9.7	0.9.7
perl	5.24.1	5.24.1	5.28.1 (buster)	5.24.1	5.24.1
bash	3.2.57	3.2.57	4.4	3.2.57	4.4
diffutils	2.8.1	-	3.6	-	-
dosfstools	2.11	2.11	4.1	2.11	4.1
gawk	3.1.5	-	4.1.4	3.1.5	4.1.4
m4	1.4.9	-	1.4.18	1.4.9	1.4.18
make	3.81	3.81	4.2.1	3.81	4.2.1
sed	4.1.2	4.1.2	4.2.2	4.1.2	4.2.2

REVISION HISTORY	RZ/G Verified Linux Package for 64bit kernel Version 1.0.6 Release Note for HTML5 update1
------------------	--

Rev.	Date	Description	
		Page	Summary
1.03	Nov. 30, 2020	—	The first release of VLP v1.0.6
1.06	Apr. 23, 2021	3	Add a new file (buildtool_gecko_v106.tar.xz) to the basic packages for HTML5 (Gecko) environment.
		7	Change the step (3) in the section 3.1 to use the new file.
		8	Change the step (1) in the section 3.2 to create the working directory.

## Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/contact/>

All trademarks and registered trademarks are the property of their respective owners.