RZ/G Verified Linux Package
Version 2.1.6-RT

Release Note

Introduction

This release note describes the contents, the building procedures and the important points of the RZ/G Verified Linux Package (hereinafter referred to as "VLP").

Contents

1.   Release Items

- **Name and version**
  RZ/G Verified Linux Package Version 2.1.6-RT

- **Distribution method**
  Provided via the RZ/G Marketplace. Please visit the sites below and create an account to download the packages.

  RZ/G Marketplace:
  English: https://www.renesas.com/us/en/products/rzg-linux-platform/rzg-marcketplace.html
  Japanese: https://www.renesas.com/jp/ja/products/rzg-linux-platform/rzg-marcketplace.html

- **Target boards**
  iWave RZ/G1H-PF Qseven Development Platform
  iWave RZ/G1M-PF Qseven Development Platform
  iWave RZ/G1N-PF Qseven Development Platform
  iWave RZ/G1E-PF SODIMM Development Platform
  iWave RZ/G1C-PF Pi SBC Development Platform

  These boards are provided by iWave Systems Technologies Pvt. Ltd.

- **Verified functions**
  Linux BSP
  - Linux Kernel
  - Linux Drivers
  - Graphics Libraries
  GUI Framework
  - Qt (LGPL version)

- **File contents**
  VLP is delivered by the files listed in **Table 1**.

Table 1. RZ/G Verified Linux Package

File		Explanation
rzg_bsp_eva_v216rt.tar.gz (Evaluation version)		Yocto recipe packages
rzg_bsp_pro_v216rt.tar.gz (Product version)		
oss_pkg_v216rt.7z.001 (500MB)	oss_pkg_v216rt.7z.006 (500MB)	Open source packages
oss_pkg_v216rt.7z.002 (500MB)	oss_pkg_v216rt.7z.007 (500MB)	
oss_pkg_v216rt.7z.003 (500MB)	oss_pkg_v216rt.7z.008 (500MB)	
oss_pkg_v216rt.7z.004 (500MB)	oss_pkg_v216rt.7z.009 (500MB)	
oss_pkg_v216rt.7z.005 (500MB)	oss_pkg_v216rt.7z.010 (491MB)	
r01tu0269ej0105-rz-g.pdf		This document
r01tu0270ej0105-rz-g.pdf		Component list
setup_env_script_v216rt.py		Script for setting up a build environment

Optional files for updating from VLP 2.1.4-RT

File	Explanation
r01tu0271ej0105-rz-g.pdf	Patch application guide for VLP 2.1.4-RT users
v214rt-to-v216rt.patch.tar.gz	Diff files of Yocto recipes compared with VLP 2.1.4-RT

Note)　Open source packages contain all source codes of OSS except for Linux kernel. These are the same versions of OSS used when VLP was verified.
These are also used in "offline" environment. Please refer to the section **2. Build environment**.

2. Build environment

Figure 1 shows an overall constitution of the recommended environment of VLP. 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.
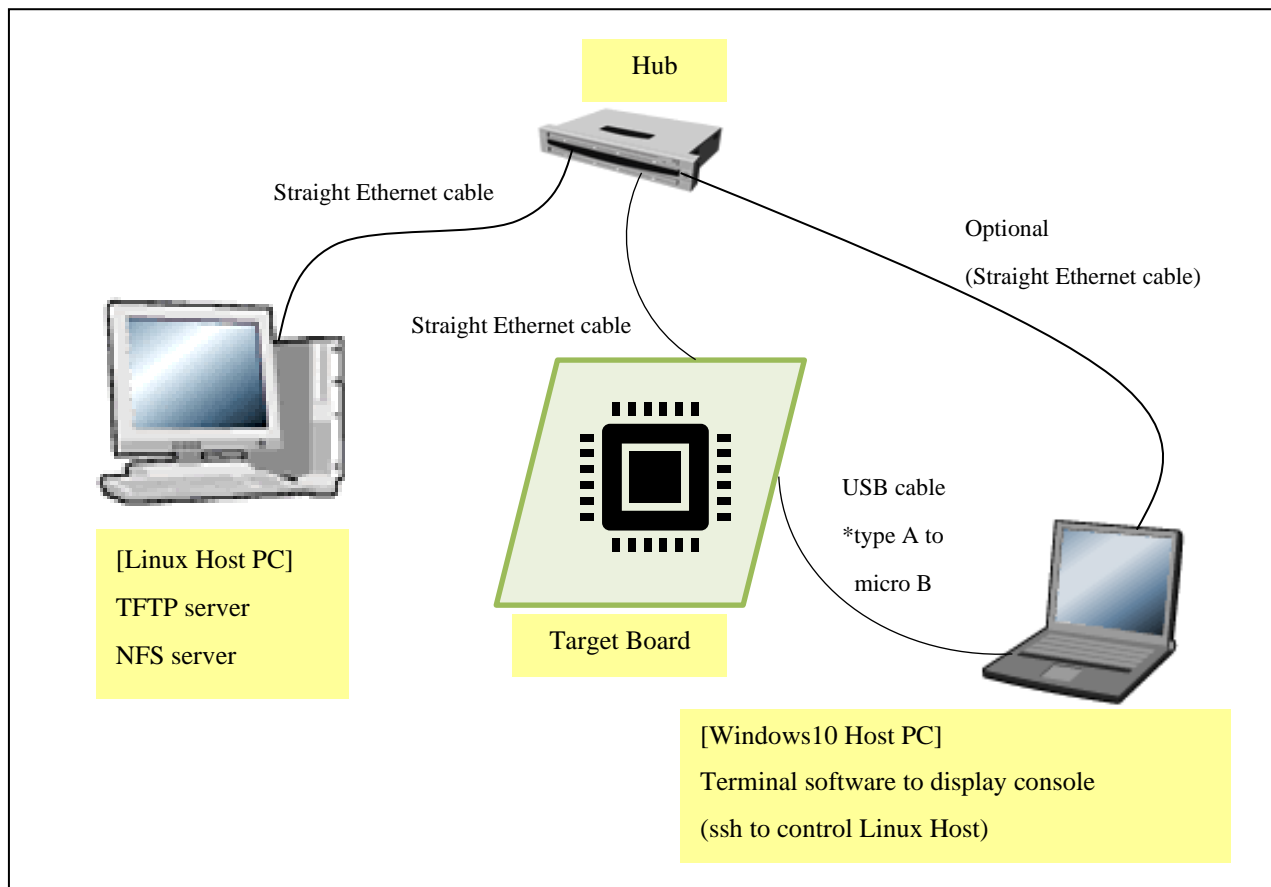
Figure 1. Recommended Environment

Most bootable images VLP supports can be built on an "offline" environment.
The word "offline" means an isolated environment which does not connect to any network. Since VLP includes all necessary source codes of OSS except for the Linux kernel, VLP 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-minimal
- 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
- Building images which use meta-rzg-demos layer

RENESAS

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

Equipment		Description
Linux Host PC		Used as build/debug environment Max 100GB free space on HDD is necessary
	OS	Ubuntu 16.04 LTS is recommended 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	Windows10 are recommended
	Terminal software	Used for controling serial console of the target board Tera Term (latest version) is recommended Available at https://ttssh2.osdn.jp/index.html.en
	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

3.   Building Instructions

This chapter describes the building instructions. There are two sections for building procedures (3.1 and 3.2). Please choose either one of them and run the commands from the chosen section. Section 3.1 describes the manual building method. Section 3.2 describes a method to use a script which automates the steps setting up the build environment and build the BSP. Section 3.2 runs automatically the same commands from section 3.1. Whichever the choice, the output will be the same.

Figure 2 shows a building outline of section 3.1 and 3.2. Once the building procedures are executed, some image files which names will depend on the target board name will be made. **Table 3** lists the names of the image files. Moreover, the GUI framework and the building procedures can be changed by a user's choice. Because of that, please select a target board and a GUI framework, then run the commands from section 3.1 or 3.2.
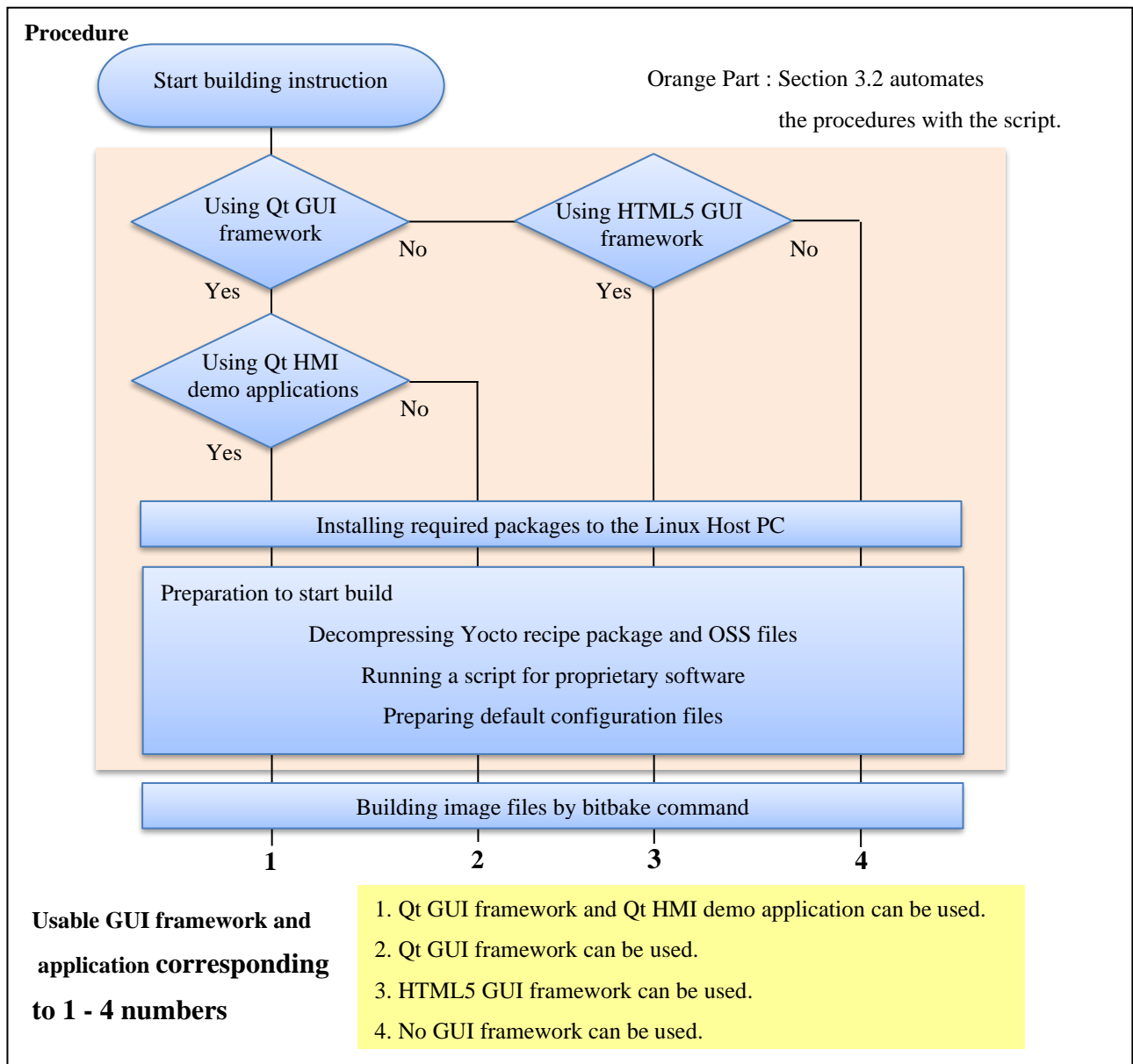
Figure 2. Building Outline

Note)   There are two GUI frameworks available, Qt and HTML5. One of them or none can be selected. In case Qt GUI framework is selected, please choose whether Qt HMI demo applications are required or not.

Table 3   Image files

	Linux kernel	Device tree file	root filesystem	Kernel modules	u-boot
RZ/G1H	uImage-iwg21m.bin	uImage-r8a7742-iwg21m.dtb	core-image-weston-iwg21m.tar.bz2	modules-iwg21m.tgz	u-boot-iwg21m.bin
RZ/G1M	uImage-iwg20m-g1m.bin	uImage-r8a7743-iwg20d-q7-dbcm-ca.dtb	core-image-weston-iwg20m-g1m.tar.bz2	modules-iwg20m-g1m.tgz	u-boot-iwg20m-g1m.bin
RZ/G1N	uImage-iwg20m-g1n.bin	uImage-r8a7744-iwg20d-q7-dbcm-ca.dtb	core-image-weston-iwg20m-g1n.tar.bz2	modules-iwg20m-g1n.tgz	u-boot-iwg20m-g1m.bin
RZ/G1E	uImage-iwg22m.bin	uImage-r8a7745-iwg22d-sodimm.dtb	core-image-weston-iwg22m.tar.bz2	modules-iwg22m.tgz	u-boot-iwg22m.bin
RZ/G1C	uImage-iwg23s.bin	uImage-r8a77470-iwg23s-sbc.dtb	core-image-weston-iwg23s.tar.bz2	modules-iwg23s.tgz	u-boot-iwg23s.bin

Note)   If the error message below is displayed and the build fails, please set the user's name and email address according to the message and try again the build.

```
*** Please tell me who you are.

Run

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


to set your account's default identity.
Omit --global to set the identity only in this repository.


fatal: unable to auto-detect email address (got 'renesas@rzg.(none)')
```

3.1   Building instructions all manually

This section describes the manual building method. VLP supports Qt as GUI framework. Please run the commands depending on the GUI framework which will be used.

Before starting the manual build, run the command below on the Linux Host PC to install packages used for building BSP.

```
$ 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 libsdl1.2-dev xterm p7zip-full
```

Please refer to the URL below for detailed information:

- https://www.yoctoproject.org/docs/2.4.2/yocto-project-qs/yocto-project-qs.html

(1)   Set the shell variable

```
$ export WORK=[user's working directory]
```

Please enter an absolute path of the working directory.

(2)   Create the working directory, and decompress Yocto recipe package

```
$ mkdir -p $WORK
$ cd $WORK
$ tar xvzf ./rzg_bsp_type_v216rt.tar.gz
```

Please replace "*type*" by "*pro*" or "*eva*".   Copy the compressed Yocto recipe package files (rzg_bsp_*pro*_v216rt.tar.gz for product version, rzg_bsp_*eva*_v216rt.tar.gz for evaluation version) into the current directory prior to this step.

(3)   Execute the copy scripts for proprietary software

```
$ cd $WORK/meta-renesas/meta-rzg1
$ ./copy_mm_software_lcb.sh ../../MMP
$ ./copy_gfx_software_rzg1x.sh ../../MMP    (*1)
```

(*1) Run the script corresponding to the target board
      "copy_gfx_software_rzg1h.sh" RZ/G1H   "copy_gfx_software_rzg1e.sh" RZ/G1E
      "copy_gfx_software_rzg1m.sh" RZ/G1M   "copy_gfx_software_rzg1c.sh" RZ/G1C
      "copy_gfx_software_rzg1n.sh" RZ/G1N

(4)   Setup the build environment

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

Environment to build are set by the source command.

(5)   Prepare the default configuration files for the target board

Please note that this step has been changed from previous version.

Copy suitable template files into the work directory.

Note)   Fill the directory name corresponding to the target board to <board>
      iwg21m: iWave board for RZ/G1H   iwg22m: iWave board for RZ/G1E

iwg20m-g1m:	iWave board for RZ/G1M	iwg23s:	iWave board for RZ/G1C
iwg20m-g1n:	iWave board for RZ/G1N		

No GUI Framework required

```
$ cd $WORK/build
$ cp ../meta-renesas/meta-rzg1/templates/<board>/*.conf ./conf
```

Enable Qt

```
$ cd $WORK/build
$ cp ../meta-renesas/meta-rzg1/templates/<board>/qt/*.conf ./conf
```

Note) If Qt HMI demo applications are required, please copy the file as below instead of the command above.

```
$ cd $WORK/build
$ cp ../meta-rzg-demos/meta-rzg1/qt-hmi-demo/template/<board>/*.conf ./conf
```

(6) Decompress OSS files to "build" directory

```
$ cd $WORK/build
$ 7z x ./oss_pkg_v216rt.7z.001    (*2)
```

Copy the compressed Yocto recipe package files (oss_pkg_v216rt.7z.001 - oss_pkg_v216rt.7z.015) into the "build" directory prior to these steps. All OSS packages will be decompressed at the step marked "*2". 7ziped files are not able to decompress individually. All oss_pkg.7z files must be placed at the same directory before running 7z command.

Note) This step is not mandatory in case the "offline" environment is not used.
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 there is a possibility to fail a build because of implicit changes of the repository of OSS.

(7) Download Linux kernel source code

```
$ cd $WORK/build
$ bitbake linux-renesas -c fetch
```

Once this step is finished, the Linux Host PC can be disconnected from the network. If you want to prevent network access, please change the line in the ${WORK}/build/conf/local.conf as below:

```
BB_NO_NETWORK = "1"
```

To change BB_NO_NETWORK from "0" to "1".

Note) The source code of the linux kernel is not included in oss_packages.7z from this release. This step obtains the source code from CIP's Git repository.

(8) Start the build

```
$ cd $WORK/build
$ bitbake core-image-weston
```

Note) Please connect the Linux Host PC to the network if Qt HMI demo applications are required.

RENESAS

Building an image can take up to a few hours depending on the user's host system performance.

After the build is successfully completed, a similar output will be seen:

```
NOTE: Tasks Summary: Attempted 4945 tasks of which 16 didn't need to be rerun and al
l succeeded.
```

and the command prompt will return.

All necessary files listed in **Table 3** will be generated by the *bitbake* command at build/tmp/deploy/images directory.

3.2   Building instructions with script

This section describes the script building method. Note that the script automates the steps to setup the build environment. There are additional commands to run manually once the script finished running. Please make a directory (hereinafter called "[*user's working directory*]") somewhere in the Linux Host PC and copy the following files into the directory in advance. These files can be downloaded from RZ/G Marketplace.

- The Yocto recipe package   (Either "rzg_bsp_eva_v216rt.tar.gz" or "rzg_bsp_pro_v216rt.tar.gz")
- The Open Source packages   ("oss_pkg_v216rt.7z.001" - "oss_pkg_v216rt.7z.010")
- The script ("setup_env_script_v216rt.py")

After that, please run the following commands.

Note)   Python3 is required to run the script. This script is experimental and provided AS IS.

(1)   Set a shell variable

```
$ export WORK=[user's working directory]
```

  Please enter an absolute path.

(2)   Run the script

```
$ cd $WORK
$ ./setup_env_script_v216rt.py
```

The following message will be displayed. Please enter the number corresponding to the target board and the GUI framework that will be used. A sudo command to update a package list will then run, hence enter the password.

```
What is your target board?
Please enter one of the following numbers corresponding to the target board.
[1] RZ/G1H, [2] RZ/G1M, [3] RZ/G1N, [4] RZ/G1E, [5] RZ/G1C


Please Enter Number:


Which GUI framework do you want to use?
Please enter one of the following numbers corresponding to the GUI framework.
[1] Qt, [2] HTML5, [3] None* (*Does not use Qt and HTML5.)


Please Enter Number:
```

The following message will be displayed once the script's sequence is completed. Then run manually the next commands from this section.

Once the script finishes, a directory named "user_work" and a file named "v216rt_script.log" will be made in the [*user's working directory*]. In case the script fails to setup or is interrupted by the user, please delete the "user_work" directory in the [*user's working directory*], and then try it again from step (2).

```
**   The sequence of this script was successfully completed.   **

**   Please refer to the manual and execute the next procedures.   **
```

Please enter "y" in case the Qt HMI demo applications are required.

```
Do you want to include the Qt HMI demo applications? (y/n):
```

The script will search packages in the user's Linux Host PC. The following confirmation message will be displayed if the required packages are not installed. If "y" is chosen, a command to install the packages will run. If "n" is chosen, the script will stop.

```
It is necessary to install the following packages to your Linux Host PC.

Packages = [the required packages]


Is it OK to install the packages? (y/n) :
```

[*the required packages*] will change depending on the chosen GUI framework.

Please refer to the URL below for detailed information:

- https://www.yoctoproject.org/docs/2.4.2/yocto-project-qs/yocto-project-qs.html

This script requires network connection. If the Linux Host PC is not connected to the network, the following message will be displayed. Please connect the Linux Host PC to the network and press enter. Please press "Ctrl + C" to stop the script in case the network is not available.

```
WARNING: Linux Host PC is not connected to the network.

Please connect to the network.

Process wait until it is online.

Will you try again? (Yes : [enter], No : [Ctrl + C]):
```

Note)   If there are no Yocto recipe package or Open Source packages inside the [*user's working directory*], the following messages will be displayed and the script will stop.

```
ERROR: There are no Yocto recipe package(rzg_bsp_***_v216rt.tar.gz) or Open Source

Packages (oss_pkg_v216rt.7z.***) in your working directory.

Please copy the files to the directory and try again to run this script.
```

Note)   If the script fails to run the setup command, the following messages will be displayed and the script will stop. Please then delete the "user_work" directory in the [*user's working directory*] and try it again from step (2).

```
ERROR: Command = [the executed command] failed.

This script will stop.
```

(3)　Setup the environment to use bitbake command

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

(4)　Download Linux kernel source code

```
$ cd $WORK/build
$ bitbake linux-renesas -c fetch
```

Once this step is finished, the Linux Host PC can be disconnected from the network. If you want to prevent network access, please change the line in the ${WORK}/build/conf/local.conf as below:

```
BB_NO_NETWORK = "1"
```

To change BB_NO_NETWORK from "0" to "1".

Note)　The source code of the linux kernel is not included in oss_packages.7z from this release. This step obtains the source code from CIP's Git repository.

(5)　Start the build

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

Building an image can take up to a few hours depending on the user's host system performance.

After the build is successfully completed, a similar output will be seen:

```
NOTE: Tasks Summary: Attempted 4945 tasks of which 16 didn't need to be rerun and all succeeded.
```

and the command prompt will return.

All necessary files listed in **Table 3** will be generated by the bitbake command at `user_work/build/tmp/deploy/images` directory.

3.3 Building SDK

To build Software Development Kit (SDK), run the commands below after building a BSP.

```
$ cd $WORK/build
$ bitbake core-image-weston-sdk -c populate_sdk
```

Note) Please setup a building environment to enable Qt before building BSP. This SDK is also valid for images which enabled Gekco or disabled GUI frameworks.

4. Changes from previous version

The Linux kernel has been replaced to newer one. Version information of all components are available at "Component list of VLP". Almost all components are same as VLP 2.1.4-RT.

Table 4. Versions of commonly used components

Components	VLP 2.1.4-RT	VLP 2.1.6-RT
Linux kernel	4.4.190-cip36-rt25	4.4.208-cip41-rt27
GCC	7.2.1 (Linaro GCC 7.2-2017.11)	7.2.1 (Linaro GCC 7.2-2017.11)
Glibc	2.19 (CIP)	2.19 (CIP)
Binutils	2.28-5 (CIP)	2.28-5 (CIP)
Busybox	1.22.0 (CIP)	1.22.0 (CIP)
Openssl	1.0.1t (CIP)	1.0.1t (CIP)
gstreamer1.0	1.12.2	1.12.2
Wayland	1.13.0	1.13.0
Weston	2.0.0	2.0.0
python2	2.7.17	2.7.17
python3	3.5.3	3.5.3
Qt (LGPL version)	5.6.3	5.6.3

Note)   CIP version of components is going to be maintained by CIP project for over ten years.

We recommend switching to Ubuntu 16.04 from 14.04, which was used for previous version of VLP. Support for Ubuntu 14.04 LTS expired in Apr. 2019.

The booting method and the required settings are not changed from the previous version. Please refer to "RZ/G Verified Linux Package Start-Up Guide".

5. Restrictions

In this version, HTML5 (Gecko) is not verified.

6. Note

(1)   Weston

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

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.