RZ/G Verified Linux Package V2.1.21

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").

This document also describes the environment to build VLP without using "RZ/G Development Platform". If additional information about the platform is required, please refer to "RZ/G Linux Platform Tools User's Manual for the RZ/G Series".

Contents

1. Release Items

- **Name and version**
  RZ/G Verified Linux Package Version 2.1.21
  (hereinafter referred to as "VLP v2.1.21").

- **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 v2.1.21 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

  You can also download the basic packages of VLP v2.1.21 which are listed in **Table 1** from the site below.

  RZ/G Marketplace:
  https://www.renesas.com/products/microcontrollers-microprocessors/rz-cortex-a-mpus/rzg-linux-platform

- **Target boards**
  iWave RZ/G1H-PF Qseven Development Platform R2.1, R4.0
  iWave RZ/G1M-PF Qseven Development Platform R2.0, R5.0
  iWave RZ/G1N-PF Qseven Development Platform R3.4
  iWave RZ/G1E-PF SODIMM Development Platform R3.1, R4.0
  iWave RZ/G1C-PF Pi SBC Development Platform R2.0, R4.0

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

- **Verified functions**
  Linux BSP
  - Linux Kernel
  - Linux Drivers
  - Graphics Libraries
  GUI Framework
  - HTML5 (Gecko is used as a rendering engine)
  - Qt (LGPL version)

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

Table 1. RZ/G Verified Linux Package

Basic packages

File	Description
rzg_bsp_eva_v2121.tar.gz (Evaluation version) (100MB) rzg_bsp_pro_v2121.tar.gz (Product version) (100MB)	Yocto recipe packages
rzg_bsp_gecko_v2121.tar.gz (556KB)	Yocto recipe packages for HTML5
oss_pkg_v2121.7z (3.9GB)	Open source software packages
oss_pkg_gecko_v2121.7z (842MB)	Open source packages for HTML5
r01tu0269ej0122-rz-g.pdf	This document
r01tu0270ej0120-rz-g.pdf	Component list
setup_env_script_v2121.py (7.6KB)	Script for setting up a build environment
RTK0EF0045Z0004AZJ-v2.1.21.zip (101MB)	Verified Linux Package. This file includes the Yocto recipe packages and the necessary documents.

(Optional) A document and files for updating from VLP v2.1.19

File	Description
r01tu0271ej0120-rz-g.pdf	Patch application guide for VLP v2.1.19 users
v2119-to-v2121.patch.tar.gz (28KB)	Diff files of Yocto recipes compared with VLP v2.1.19

Note) Open source software 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**.

Additional packages

File	Description
RTK0EF0045Z9000AZJ-v2.1.8.zip (42MB)	RZ/G1 Group BSP Manual Set.

Note) Detailed information regarding the configuration (Device tree) and usage of the device drivers contained in this BSP can be downloaded from Renesas.com. Please download the "RZ/G1 Group BSP Manual Set".

- https://www.renesas.com/us/en/document/rzg1-group-bsp-manual-set-rtk0ef0045z9000azj-v2

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.
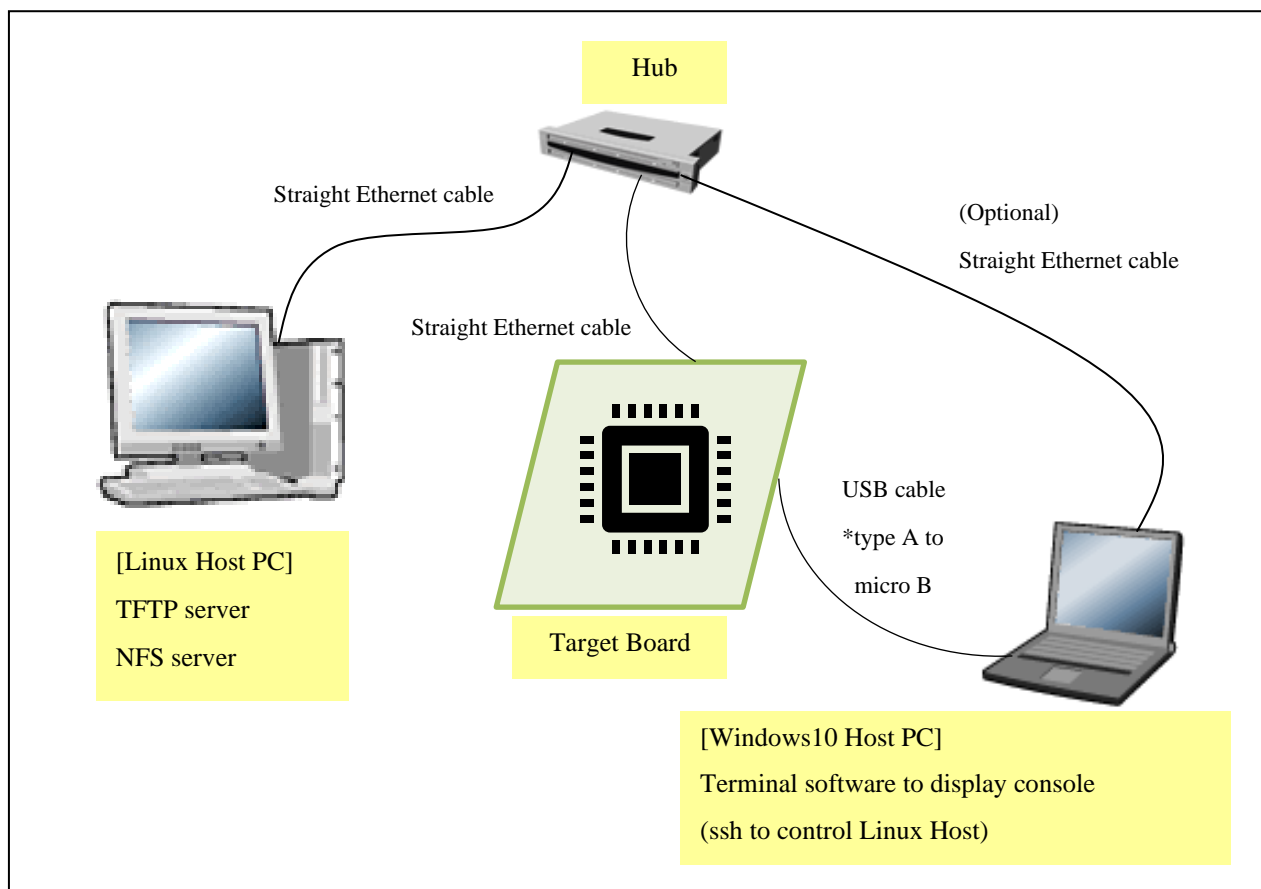

Figure 1. Recommended Environment

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 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.

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

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 (including the case using Gecko)
- 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

3.   Building Instructions

This chapter describes the building instructions of Board Support Package (hereinafter referred to as "BSP"). 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.
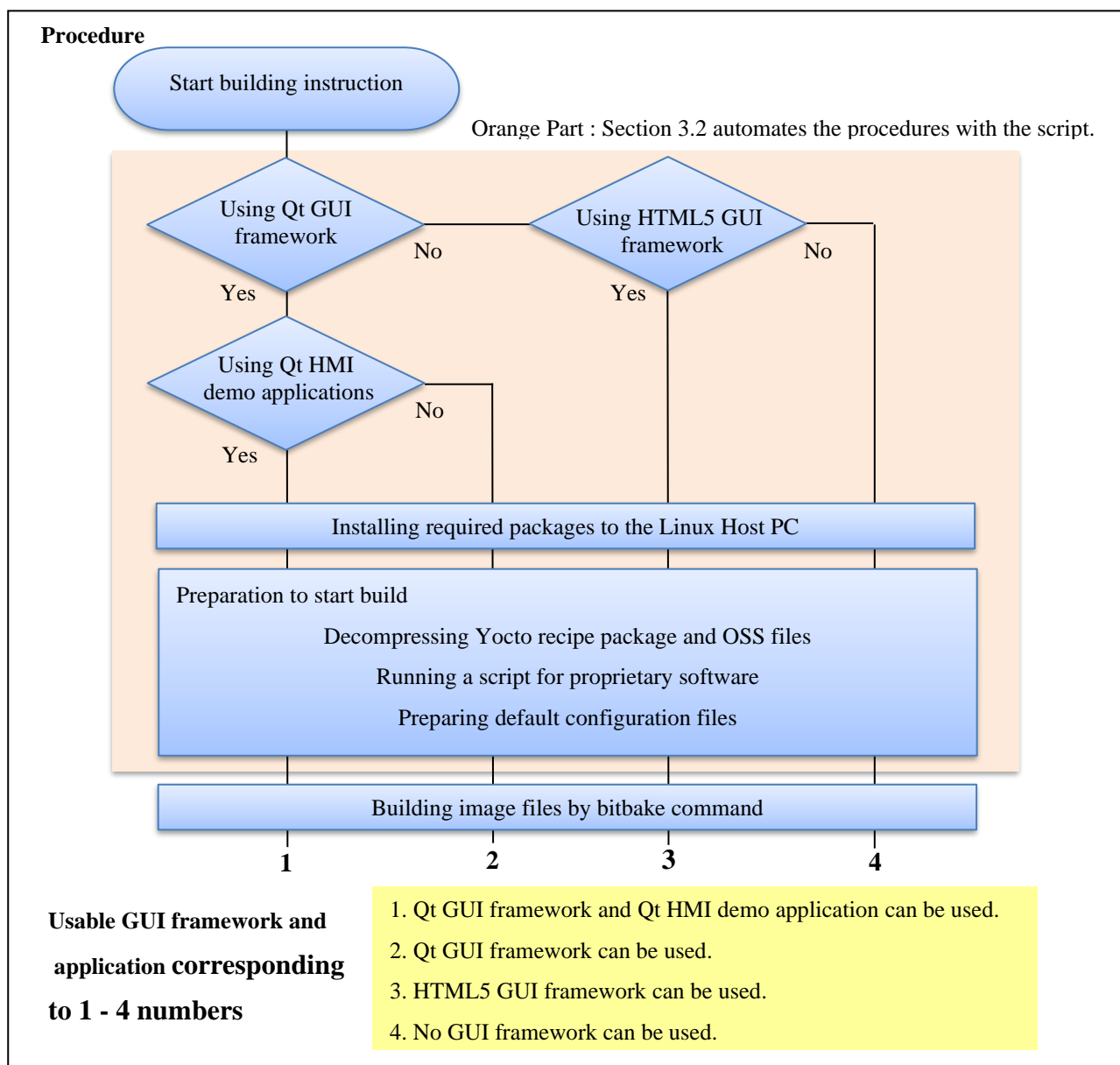

Figure 2. Building Outline

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

Board	Linux kernel	Device tree file	root filesystem	Kernel modules	u-boot
RZ/G1H Rev4.0	uImage-iwg21m.bin	uImage-r8a7742-iwg21d-q7.dtb uImage-r8a7742-iwg21d-q7-dbcm-ca.dtb	core-image-weston-iwg21m.tar.bz2	modules-iwg21m.tgz	u-boot-iwg21m.bin
RZ/G1H Rev2.1	uImage-iwg21m.bin	uImage-r8a7742-iwg21d-q7-rev2.dtb uImage-r8a7742-iwg21d-q7-dbcm-ca-rev2.dtb	core-image-weston-iwg21m.tar.bz2	modules-iwg21m.tgz	u-boot-iwg21m.bin
RZ/G1M Rev5.0	uImage-iwg20m-g1m.bin	uImage-r8a7743-iwg20d-q7.dtb uImage-r8a7743-iwg20d-q7-dbcm-ca.dtb	core-image-weston-iwg20m-g1m.tar.bz2	modules-iwg20m-g1m.tgz	u-boot-iwg20m-g1m.bin
RZ/G1M Rev2.0	uImage-iwg20m-g1m.bin	uImage-r8a7743-iwg20d-q7-rev2.dtb uImage-r8a7743-iwg20d-q7-dbcm-ca-rev2.dtb	core-image-weston-iwg20m-g1m.tar.bz2	modules-iwg20m-g1m.tgz	u-boot-iwg20m-g1m.bin
RZ/G1N Rev5.0	uImage-iwg20m-g1n.bin	uImage-r8a7744-iwg20d-q7.dtb uImage-r8a7744-iwg20d-q7-dbcm-ca.dtb	core-image-weston-iwg20m-g1n.tar.bz2	modules-iwg20m-g1n.tgz	u-boot-iwg20m-g1n.bin
RZ/G1N Rev3.4	uImage-iwg20m-g1n.bin	uImage-r8a7744-iwg20d-q7-rev3.dtb uImage-r8a7744-iwg20d-q7-dbcm-ca-rev3.dtb	core-image-weston-iwg20m-g1n.tar.bz2	modules-iwg20m-g1n.tgz	u-boot-iwg20m-g1n.bin
RZ/G1E Rev4.0	uImage-iwg22m.bin	uImage-r8a7745-iwg22d-sodimm.dtb	core-image-weston-iwg22m.tar.bz2	modules-iwg22m.tgz	u-boot-iwg22m.bin
RZ/G1E Rev3.1	uImage-iwg22m.bin	r8a7745-iwg22d-sodimm-rev3.dtb	core-image-weston-iwg22m.tar.bz2	modules-iwg22m.tgz	u-boot-iwg22m.bin
RZ/G1C Rev4.0	uImage-iwg23s.bin	uImage-r8a77470-iwg23s-sbc.dtb	core-image-weston-iwg23s.tar.bz2	modules-iwg23s.tgz	u-boot-iwg23s.bin
RZ/G1C Rev2.0	uImage-iwg23s.bin	uImage-r8a77470-iwg23s-sbc-rev2.dtb	core-image-weston-iwg23s.tar.bz2	modules-iwg23s.tgz	u-boot-iwg23s.bin

3.1   Building instructions of a BSP all manually

This section describes the manual building method. VLP supports two GUI frameworks: Qt and HTML5 (Gecko). But these cannot be used at the same time. 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

In addition, run the command below to install packages used for building Gecko.

```
$ sudo apt-get install autoconf2.13
$ sudo apt install clang llvm clang-3.9 llvm-3.9
```

Run the command**s** 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"
```

(1)   Set the shell variable

Run the command below and enter an absolute path of the working directory.

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

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

Run the commands below. Please replace "*type*" by "*pro*" or "*eva*". Copy the compressed Yocto recipe package files (rzg_bsp_*pro*_v2121.tar.gz for product version, rzg_bsp_*eva*_v2121.tar.gz for evaluation version) into the current directory prior to this step.

```
$ mkdir -p $WORK
$ cd $WORK
$ tar xvzf ./rzg_bsp_type_v2121.tar.gz
```

IMPORTANT NOTE   This VLP set video output to LVDS as default setting. In case using HDMI, please refer to the section **6. Notes (2) Video output**.

Additionally, in case to use HTML5, run the commands below to decompress Yocto recipe package for building Gecko.

```
$ cd $WORK
$ tar xvzf ./rzg_bsp_gecko_v2121.tar.gz
```

Note)   In case to use HTML5 on iWave boards of RZ/G1E and RZ/G1C, please refer to **6. Notes (3) HTML5** and apply a patch to the meta-renesas directory.

(3) Execute the copy scripts for proprietary software

Run the scripts below. It's necessary to run a script corresponding to the target board at the command (*1). Please replace "*x*" according 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		

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

(4) Setup the build environment

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

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

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

There are 4 types of the default configuration files. Select one of them and copy suitable files into the work directory by the commands below. 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 HTML5 (Gecko)**

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

- **Enable Qt**

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

- **Enable Qt HMI demo applications**

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

(6) Decompress OSS files to "build" directory

Copy the compressed OSS package files (oss_pkg_v2121.7z, oss_pkg_gecko_v2121.7z) into the "build" directory prior to this step.

Run the commands below.

```
$ cd $WORK/build
$ 7z x ./oss_pkg_v2121.7z
```

Additionally, in case to use HTML5, run the commands below to decompress OSS files for building Gecko.

```
$ cd $WORK/build
$ 7z x ./oss_pkg_gecko_v2121.7z
```

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 the 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

Run the commands below. The source code of the linux kernel is not included in oss_pkg_v2121.7z. This step obtains the source code from CIP's Git repository. Once this step is finished, the Linux Host PC can be disconnected from the network.

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

If you want to prevent network access, please change the line in the ${WORK}/build/conf/local.conf as below:

Change BB_NO_NETWORK from "0" to "1".

```
BB_NO_NETWORK = "1"
```

Note) If Qt HMI demo applications are required, please set "BB_NO_NETWORK" to "0" and connect the Linux Host PC to the network.

(8)   Start the build

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

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

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.

Note)   If the user's name and email address are not set before starting the build, an error will occur, and the messages below will be displayed. In case to cause this error, another error will occur when you try to build again. Please set the user's name and email address like as below messages and delete [*user's working directory*]. Then try to build again from the step (1).

```
*** 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)')
```

3.2 Building instructions of a BSP 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 same directory in advance**. These files can be downloaded from Renesas Electronics Website. Please refer to "Distribution method**"** in the chapter 1.

Table 4. Basic files for the script building method

File	Description
Either rzg_bsp_eva_v2121.tar.gz (Evaluation version) or rzg_bsp_pro_v2121.tar.gz (Product version)	Yocto recipe packages
rzg_bsp_gecko_v2121.tar.gz	Yocto recipe packages for HTML5 (*4)
oss_pkg_v2121.7z	Open source software packages
oss_pkg_gecko_v2121.7z	Open source software packages for HTML5 (*5)
setup_env_script_v2121.py	Script for setting up a build environment

Note) In case not to use HTML5, it is not necessary to copy the files which are marked (*4) and (*5) in the [*user's working directory*].

If the Git has not been configured yet, set the user name and email address before starting 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"
```

After that, please run the following steps.

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

(1) Set a shell variable

Run the command below and enter an absolute path of the working directory.

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

(2) Run the script

Run the commands below to start the script.

```
$ cd $WORK
$ python3 ./setup_env_script_v2121.py
```

The following message will be displayed. Enter the number corresponding to the target board.

```
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:
```

Then enter the number corresponding to the GUI framework that will be used. A sudo command to update a package list will then run, hence enter the password.

```
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 steps (3) - (5) of section 3.2. Once the script finishes, a directory named **"user_work"** and a file named "v2121_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 the beginning of this step.

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

```
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) :
```

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 access. 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 software package 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_***_v2121.tar.gz) or Open Source
Package (oss_pkg_v2121.7z) in your working directory.
Please copy the files to the directory and try again to run this script.
```

RENESAS

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 the beginning of this step.

```
ERROR: Command = [the executed command] failed.
This script will stop.
```

Note)   If the script finishes, "v2121_script.log" is created in the [*user's working directory*]. You can check the executed commands and the results of them with this log file.

(3)   Setup the environment to use bitbake command

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

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

(4)   Download Linux kernel source code

Run the commands below. The source code of the linux kernel is not included in oss_packages.7z. This step obtains the source code from CIP's Git repository. Once this step is finished, the Linux Host PC can be disconnected from the network.

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

If you want to prevent network access, please change the line in the ${WORK}/build/conf/local.conf as below:

Change BB_NO_NETWORK from "0" to "1".

```
BB_NO_NETWORK = "1"
```

Note)   If Qt HMI demo applications are required, please set "BB_NO_NETWORK" to "0" and connect the Linux Host PC to the network.

(5)   Start the build

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

```
$ cd $WORK/user_work/build
$ 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 4945 tasks of which 16 didn't need to be rerun and al
l succeeded.
```

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

Note) If the user's name and email address are not set before starting the build, an error will occur, and the messages below will be displayed. In case to cause this error, another error will occur when you try to build again. Please set the user's name and email address like as below messages and delete [*user's working directory*]. Then try to build again from the step (1).

```
*** 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)')
```

3.3   Building instructions of 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 is available at "Component list of VLP". Almost all components are same as VLP v2.1.19.

Table 5. Versions of commonly used components

Components	VLP v2.1.19	VLP v2.1.21
Linux kernel	4.4.302-cip76	4.4.302-cip83
GCC	7.2.1 (Linaro GCC 7.2-2017.11)	7.2.1 (Linaro GCC 7.2-2017.11)
Glibc	2.28-10+deb10u1 (CIP)	2.28-10+deb10u2 (CIP)
Busybox	1.30.1 (CIP)	1.30.1 (CIP)
Openssl	1.1.1n-0+deb10u6 (CIP)	1.1.1n-0+deb10u6 (CIP)
gstreamer1.0	1.12.2	1.12.2
Wayland	1.13.0	1.13.0
Weston	2.0.0	2.0.0
python3	3.5.3	3.5.3
Qt (LGPL version)	5.6.3	5.6.3

Note)   By default settings, VLP v2.1.9 and later integrate Debian 10 (Buster) based CIP Core Packages which indicated as "(CIP)" in the **Table 5**. Older Debian 8 (Jessie) based CIP packages are still supported just in case. Please refer to **6. Notes (4)** for how to select between Debian versions to be built in.
For more technical information, please contact Renesas.

Note)   Python2 has been removed since VLP v2.1.9 because it is EOL.

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

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

(1) In this version, Wifi/Bluetooth on the iWave RZ/G1E rev4.0 board has not been supported.

6. Notes

(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.

(2) Video output

This VLP 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-renesas
$ patch -p1 < ../extra/0001-VLP2.1.21-Add-HDMI-support-for-RZ-G1.patch
```

Note) Due to hardware changes on the iWave RZ/G1H R4.0 board, HDMI output is not supported (HDMI output is supported on RZ/G1H R2.1 board).

(3) HTML5

- **Patch for iWave Platform of RZ/G1E, RZ/G1C**

  ✓ iWave RZ/G1E-PF SODIMM Development Platform
  ✓ iWave RZ/G1C-PF Pi SBC Development Platform

  The above target boards don't have enough memory to operate HTML5(Gecko) properly. In case to use HTML5 on them, please run the commands below to adjust memory for HTML5 after the step **(2)** of the section **3.1**.

```
$ cd $WORK/meta-renesas
$ patch -p1 < ../extra/0001-mmngr-iwg22m-iwg23s-reducing-CMA-reserved-for-MMP-\
to.patch
```

  After that, run the steps (3) - (8) of the section 3.1.

Note) Gecko for RZ/G1E and RZ/G1C is tested on the environment of CMA 128MB, the other functions are tested on that of CMA 256MB which is default.

- **WebGL (Web Graphics Library)**

RZ/G1E, RZ/G1C cannot use WebGL of JavaScript API for 2D/3D graphics.

(4) CIP Core Packages

VLP includes Debian 10 (Buster) based CIP Core Packages which indicated as "(CIP)" in **Table 5** and enabled in the default settings. Replacing by Debian 8 (Jessie) based CIP Core Packages is not recommended but can do following instructions as below. CIP Core Packages are going to be maintained by the Civil Infrastructure Platform project. For more technical information, please contact Renesas.

Note) Debian 8 (Jessie) based CIP Core Packages can be built, but they are not fully tested. They are preliminary and provided AS IS with no warranty.

Buster-full (default):

The following line is added as default in the `local.conf`:

```
CIP_MODE = "Buster-full"
```

Jessie:

This setting selects Debian 8 (Jessie) based CIP Core Packages. Add the following line in the `local.conf`:

```
CIP_MODE = "Jessie"
```

Additional OSS packages will be downloaded via the internet to build Debian 8 (Jessie) based CIP Core Packages when running bitbake command, so please connect the Linux Host PC to the network before starting the build, and change the line in the ${WORK}/build/conf/local.conf as below:

Change BB_NO_NETWORK from "1" to "0".

```
BB_NO_NETWORK = "0"
```

Table 6 shows a comparison between Jessie and Buster based CIP core packages.

Table 6. Versions of Packages

Package	Jessie	Buster
busybox	1.22	1.30.1
openssl	1.0.1t	1.1.1n
glibc	2.19	2.28-10+deb10u1
binutils	2.25	2.31.1
openssh	7.5p1	7.9p1
coreutils	8.27	8.30
gnupg	2.2.0	2.2.12
libassuan	2.4.3	2.5.2
libpam	1.3.0	1.3.1
libgcrypt	1.8.0	1.8.4
libunistring	0.9.7	0.9.10
libunwind	1.2	1.2
perl	5.24.1	5.28.1
bash	4.4	4.4
diffutils	3.6	3.6
dosfstools	4.1	4.1
gawk	4.1.4	4.1.4
m4	1.4.18	1.4.18
make	4.2.1	4.2.1
sed	4.2.2	4.2.2
asciidoc	8.6.9	8.6.9
cpio	2.12	2.12
cronie	1.5.1	1.5.1
expect	5.45	5.45
gpgme	1.9.0	1.9.0
groff	1.22.3	1.22.3
gzip	1.8	1.9
logrotate	3.12.3	3.12.3
man	1.6g	1.6g
patchelf	0.9	0.9
pixz	1.0.6	1.0.6
slang	2.3.1a	2.3.1a
swig	3.0.12	3.0.12
unfs3	0.9.22-r497	0.9.22-r497
xmlto	0.0.28	0.0.28

(5)   GPIO of RZ/G1H

Since VLP v2.1.8-RT release, the driver module that controls the GPIO on RZ/G1H was updated. Please refer to "RZ/G1 Group BSP Manual Set" for details. For example, to export and create a device node for the GPIO pin GPIO5_0 in older VLPs, user should run command below:

```
$ echo 832 > /sys/class/gpio/export
```

In recent VLPs, it was changed like following:

```
$ echo 836 > /sys/class/gpio/export
```

To check the GPIO ranges, please execute the below command in the Linux command line (RZ/G1H as an example):

```
$ cat /sys/kernel/debug/pinctrl/e6060000.pin-controller/gpio-ranges
GPIO ranges handled:
0: e6050000.gpio GPIOS [992 - 1023] PINS [0 - 31]
0: e6051000.gpio GPIOS [962 - 991] PINS [32 - 61]
0: e6052000.gpio GPIOS [932 - 961] PINS [64 - 93]
0: e6053000.gpio GPIOS [900 - 931] PINS [96 - 127]
0: e6054000.gpio GPIOS [868 - 899] PINS [128 - 159]
0: e6055000.gpio GPIOS [836 - 867] PINS [160 - 191]
```

From the command line output of above example, we can see GPIO5 bank (with physical address e6055000) has the range of [836 – 867]. It means number 836 is assigned to GPIO5_0 by Linux.

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.