# Introduction

This start-up guide explains RZ/G2 Group Yocto recipe package files, the system environments, the make method of kernel, the operating of U-Boot and so on.

This product RZ/G2 Yocto recipe is a basic package to operate built-in Linux and basic middleware on the RZ/G2 System Evaluation Board. Please contact Renesas Electronics person who provided this product to you in case of questions.

Note: Currently, RZ/G2E, RZ/G2M v1.3, RZ/G2M v3.0, RZ/G2N and RZ/G2H, with reference boards EK874, HiHope-RZG2M, HiHope-RZG2N and HiHope-RZG2H are supported.

# RZ/G2 Linux BSP package files

This Yocto recipe will be taken

The U-Boot source code from:

https://github.com/renesas-rz/renesas-u-boot-cip.git

RZ/G2 Linux source code from:

<https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git>,branch=linux-4.19.y-cip   
<https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git>,branch=linux-4.19.y-cip-rt

## Reference (RZ/G2)

|  |  |
| --- | --- |
| Document name | Version |
| RZ/G2 Series User’s Manual: Hardware | --- |
| RZ/G2 System Evaluation Board Hardware Manual | --- |

## Environmental Requirement

Host PC and terminal software are necessary for the operation of this product. Furthermore, Ethernet cable is required to use NFS mount function. Please refer to Table 1.

Table 1 RZ/G2 Linux BSP Environmental Requirement

|  |  |
| --- | --- |
| Equipment | Explanation |
| Linux Host PC | Ubuntu 16.04 LTS (64bit) is recommended as OS. 32bit version is not supported.  It is used as building and debugging environment.  It is used as TFTP server and NFS server. |
| Windows Host PC | Windows 10 is recommended as OS.  It is used as debugging environment.  Terminal software and VCP driver are executed. |
| Terminal software | Please use following software.  1) Tera Term  (Confirmed with Japanese version of Tera Term 4.88  Available at <http://sourceforge.jp/projects/ttssh2> ) |
| VCP driver | Please install in Windows Host PC.  Execute CP210xVCPInstaller\_x86/x64.exe for install before connect. USB become virtual COM port on terminal software. Please connect to Serial-USB Bridge on RZG2 System Evaluation Board  (Available at <http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>) |
| TFTP server software | It is used when SPI Flash is written by U-Boot or Image is downloaded. |
| NFS server software | It is used when File system is mounted by NFS. |

**Recommended Environment**

The following shows a Recommended Environment.



Hub

[Linux Host PC]

TFTP server

NFS server

Straight Ethernet cable

USB cable (type A to mini/micro AB)

[Windows 10 Host PC]

Terminal software to display console

(ssh to control Linux Host)

(Straight Ethernet cable)

RZ/G2

System Evaluation Board

Figure 1. Recommended Environment for RZ/G2 Linux BSP

Note) Functions in covered with () are optional.

# Building Instructions

You can build BSP by using Yocto Project. Please execute following steps in ${WORK} directory on Linux Host PC. Filesystem by making following instruction is the one for testing current BSP package in Renesas. Please note that Renesas has not been verified with any other build configuration or modified recipes except “core-image-weston” configuration which is based on upstream Yocto Project deliverables and some additional packages correspond to gstreamer.

Note) Renesas executed following instructions with clean ${WORK}/build directory. You may use wipe-sysroot and/or bitbake -c cleansstate to reflect modifications of configuration files for Recipe as in open source Yocto Project’s standards, however Renesas strongly recommends to use recipe with clean ${WORK}/build directory for each configurations because there are some implicit dependency for header files exist to keep compatibility between application build scheme with/without proprietary software.

**Step 1 installation of required commands**

Ubuntu is used as Linux Host PC since Yocto Project Quick Start specifies Ubuntu as one of the distribution. In case of that you can install the required commands as follows.

Please refer to http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html for detail.

**$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib \  
build-essential chrpath socat libsdl1.2-dev xterm cpio python python3 \ python3-pip python3-pexpect xz-utils debianutils iputils-ping libssl-dev**

Note) There is a bitbake command in ${WORK}/poky/scripts/. Command path is available after step 6.

Note) When you use terminal interactions to build such as menuconfig under non-X terminal (ssh, etc.), please install “screen” command package to Host PC.

Note) Please set up user name and e-mail in Git. You can set up with ‘git config --global’. Please refer to online manual for git command.

Note) In Renesas environment, Ubuntu version is 16.04 LTS and git version is 2.7.4.

**Step 2 download of required files**

Required files (poky, meta-linaro) are downloaded by git clone.

**$ cd ${WORK}**

**$ git clone git://git.yoctoproject.org/poky**

**$ git clone git://git.linaro.org/openembedded/meta-linaro.git**

**$ git clone git://git.openembedded.org/meta-openembedded**

**$ git clone https://github.com/renesas-rz/meta-rzg2.git**

**$ git clone** [**http://git.yoctoproject.org/cgit.cgi/meta-gplv2**](http://git.yoctoproject.org/cgit.cgi/meta-gplv2)

**$ git clone https://github.com/meta-qt5/meta-qt5.git**

**Step 3 checkout**

Please checkout available version of each git clone.

**$ cd ${WORK}/poky**

**$ git checkout -b tmp 7e7ee662f5dea4d090293045f7498093322802cc**

**$ cd ${WORK}/meta-linaro**

**$ git checkout -b tmp 75dfb67bbb14a70cd47afda9726e2e1c76731885**

**$ cd ${WORK}/meta-openembedded**

**$ git checkout -b tmp 352531015014d1957d6444d114f4451e241c4d23**

**$ cd ${WORK}/meta-gplv2**

**$ git checkout -b tmp f875c60ecd6f30793b80a431a2423c4b98e51548**

**$ cd ${WORK}/meta-qt5**

**$ git checkout -b tmp c1b0c9f546289b1592d7a895640de103723a0305**

**$ cd ${WORK}/meta-rzg2**

**$ git checkout -b tmp <tag>**

**<tag> : please check and choose the latest tag by ‘git tag’**

**$ git tag**

**….**

**BSP-1.0.4**

**BSP-1.0.5-RT**

**….**

**“-RT” is for Linux Realtime support**

Note) tmp is a temporary name of a local branch. We can use checkout command without branch. Please note that HEAD refers directly to commit (detached HEAD).

**Step 4 copy proprietary software into recipe directory structure**

To use licensed 3D graphics software and Multimedia package from Renesas, please copy deliverables of those software into recipe directory structure. Renesas provide shell script to copy those software.

**Copy All Proprietary Software Packages to ${PKGS\_DIR}:**

**$ mkdir ${PKGS\_DIR}**

**$ cp <zip of Proprietary Software Package> ${PKGS\_DIR}**

**Install them into recipe directory structure by shell script:**

**$ cd ${WORK}/meta-rzg2**

**$ sh docs/sample/copyscript/copy\_proprietary\_softwares.sh ${PKGS\_DIR}**

Note) Subdirectory is not supporting in ${PKGS\_DIR}. Please store all packages on the root of ${PKGS\_DIR}.

Note) Please use regular alphanumeric file name ([A-Za-z0-9\_] e.g.) for ${PKGS\_DIR} due to restrictions of current copy script.

**Step 5 execute source command**

Please execute source command with oe-init-build-env for setting environment.

**$ cd ${WORK}**

**$ source poky/oe-init-build-env**

**Step 6 copy bblayers.conf and local.conf**

Please copy configuration files from deliverables.

**$ cp ${WORK}/meta-rzg2/docs/sample/conf/<supported board name>/<toolchain>/\*.conf ./conf/.**

Note) <supported board name> is the one of the following: ek874, hihope-rzg2m, hihope-rzg2n, hihope-rzg2h.  
 <toolchain> is the one of the following: poky-gcc, linaro-gcc

**Step 7 enable Multimedia package**

Please modify configurations in ${WORK}/build/conf/local.conf by following instructions.

The following standard multimedia packages are enabled

|  |  |  |
| --- | --- | --- |
| **No.** | **Functions** | **Explanation** |
| 1 | MMNGR | Memory manager driver & shared libraries |
| 2 | VSPM | VSP driver & FDP driver & shared libraries |
| 3 | VSP2 | VSP2 driver |
| 4 | OMX | OMX common parts |

To enable optional multimedia functions, please add DISTRO\_FEATURES\_append to ${WORK}/build/conf/local.conf as DISTRO\_FEATURES\_append = “ <function name>”.

Note) These configurations exist near the end of local.conf.  
Note) DISTRO\_FEATURES\_append are commented out by the default. To enable functions, please uncomment it.

**For example**

**[Disable]**

**#DISTRO\_FEATURES\_append = " h264dec\_lib"**

**[Enable (default)]**

**DISTRO\_FEATURES\_append = " h264dec\_lib”**

The following list is package name to enable/disable as optional multimedia functions

| **No.** | **Function name** | **Default value** | **Explanation** |
| --- | --- | --- | --- |
| 1 | h264dec\_lib | Enable | H264 decoder library  RTM0AC0000XV264D30SL41C |
| 2 | h264enc\_lib | Enable | H264 encoder library  RTM0AC0000XV264E30SL41C |
| 3 | h265dec\_lib | Enable | H265 decoder library  RTM0AC0000XV265D30SL41C |

The following list is dependent package name

| **No.** | **Function name** | **Type Name** | **Dependent Packages** |
| --- | --- | --- | --- |
| 1 | h264dec\_lib | RTM0AC0000XV264D30SL41C | RTM0AC0000XVCMND30SL41C  RTM0AC0000XCMCTL30SL41C  RCG3VUDRL4101ZDO |
| 2 | h264enc\_lib | RTM0AC0000XV264E30SL41C | RTM0AC0000XVCMNE30SL41C  RTM0AC0000XCMCTL30SL41C  RCG3VUDRL4101ZDO |
| 3 | h265dec\_lib | RTM0AC0000XV265D30SL41C | RTM0AC0000XVCMND30SL41C  RTM0AC0000XCMCTL30SL41C  RCG3VUDRL4101ZDO |

**Step 8 enable/disable other functions**

Please modify configurations in ${WORK}/build/conf/local.conf by following instructions.

| **No.** | **Function** | **Default support** | **How to** |
| --- | --- | --- | --- |
| 1 | Support GPLv3, GPLv3+ softwares | No | Default in local.conf:  INCOMPATIBLE\_LICENSE = "GPLv3 GPLv3+"  To enable:  #INCOMPATIBLE\_LICENSE = "GPLv3 GPLv3+" |
| 2 | Support 32 bits application | Yes | Default in local.conf:  require conf/multilib.conf  MULTILIBS = "multilib:lib32"  DEFAULTTUNE\_virtclass-multilib-lib32 = "armv7vethf-neon"  USE\_32BIT\_PKGS = "1"  To disable:  #require conf/multilib.conf  #MULTILIBS = "multilib:lib32"  #DEFAULTTUNE\_virtclass-multilib-lib32 = "armv7vethf-neon"  #USE\_32BIT\_PKGS = "1" |

**Step 9 building with bitbake**

Please build as follows. The file system (<core-image-target>-<supported board name>.tar.bz2) is created in ${WORK}/build/tmp/deploy/images/<supported board name>/ directory.

Note) <supported board name> is the one of the following: ek874, hihope-rzg2m, hihope-rzg2n, hihope-rzg2h.  
 <core-image-target> is the one of the following: core-image-bsp, core-image-weston, core-image-qt, core-image-hmi

Note) Build by bitbake might need several hours under the influence of Linux Host PC performance and network environment.

Note) The bitbake downloads some package while building. Then the bitbake might stop for network timeout or link error. In this case, please get applicable package in ${WORK}/build/downloads directory whenever build stops by wget command, or please review timeout definitions of package download (wget, etc.) described in ${WORK}/poky/meta/conf/bitbake.conf.

**$ cd ${WORK}/build**

**$ bitbake <core-image-target>**

**<core-image-target> can be:**

**core-image-bsp : basic BSP suport**

**core-image-weston : BSP with MMP and Graphic support**

**core-image-qt : BSP with MMP, Graphic and Qt support**

**core-image-hmi : BSP with MMP, Graphic and hmi demos**

# Writing of IPL/Secure

## Writing data

|  |  |  |  |
| --- | --- | --- | --- |
| Filename | Program Top Address | Flash Save Address | Description |
| bootparam\_sa0.srec | 0xE6320000 | 0x000000 | Loader(Boot parameter) |
| bl2-<board>.srec | 0xE6304000 | 0x040000 | Loader |
| cert\_header\_sa6.srec | 0xE6320000 | 0x180000 | Loader(Certification) |
| bl31-<board>.srec | 0x44000000 | 0x1C0000 | ARM Trusted Firmware |
| tee-<board>.srec | 0x44100000 | 0x200000 | OP-Tee |
| u-boot-elf-<board>.srec | 0x50000000 | 0x300000 | U-Boot |

Note) <board>: ek874, hihope-rzg2m, hihope-rzg2n, hihope-rzg2h.

## Dip-Switch

## Switch setting for EK874 (RZG2E)

1. SCIF Download Mode

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Switch  Number | Pin1 | Pin2 | Pin3 | Pin4 | Pin5 | Pin6 |
| SW12 | OFF | OFF | OFF | - | - | - |

1. Boot Mode

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Switch  Number | Pin1 | Pin2 | Pin3 | Pin4 | Pin5 | Pin6 |
| SW12 | ON | ON | ON | - | - | - |

## Switch setting for HiHope-RZG2M, HiHope-RZG2N and HiHope-RZG2H

1. SCIF Download Mode

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Switch  Number | Pin1 | Pin2 | Pin3 | Pin4 | Pin5 | Pin6 | Pin7 | Pin8 |
| SW1002 | ON | ON | ON | ON | OFF | OFF | OFF | OFF |

1. Boot Mode

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Switch  Number | Pin1 | Pin2 | Pin3 | Pin4 | Pin5 | Pin6 | Pin7 | Pin8 |
| SW1002 | ON | ON | ON | ON | ON | OFF | ON | ON |

## How to write

Please connect RZ/G2 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC.

**Step 1 connect cable**

Connect USB Host connector of Windows Host PC that is virtual COM port to RZ/G2 System Evaluation Board with USB cable for displaying console.

**Step 2 setting the terminal software**

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port.

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none.

**Step 3 write data file to SPI Flash**

A file is written in SPI Flash in the following procedures.

* Set dip switch “SCIF download mode”.
* Reset board then start SCIF download mode.
* After “Please send !” displayed, In case of Tera Term, transmit file AArch64\_Flash\_writer\_SCIF\_DUMMY\_CERT\_E6300400\_<board\_name>.mot which is stored in ${WORK}/build/tmp/deploy/images/<board\_name>, by "File -> Send file (S)".
* Execute xls2 command (load program to flash).

**SCIF Download mode (w/o verification)**

**(C) Renesas Electronics Corp.**

**-- Load Program to SystemRAM ---------------**

**please send !**

**RZ/G2 Scif Download MiniMonitor V1.00 2019.04.12**

**Work Memory : SystemRAM**

**Board Judge : Used Board-ID**

**Board Name : HiHope RZ/G2M**

**Product Code : RZ/G2M ES1.1**

**>xls2**

**===== Qspi/HyperFlash writing of Gen3 Board Command =============**

**Load Program to Spiflash**

**Writes to any of SPI address.**

**Winbond : W25M512JV**

**Program Top Address & Qspi/HyperFlash Save Address**

**===== Please Input Program Top Address ============**

**Please Input : H'**

* After "Please Input Program Top Address" is displayed, input Program Top Address in 3.1 and "Enter".
* After "Please Input Qspi/HyperFlash Save Address" is displayed, input Flash Save Address in 3.1 and "Enter".
* After "Please send ! ('.' & CR stop load)" is displayed, In case of Tera Term, transmit files in 3.1 by "File -> Send file (S)".
* If there are some data in writing area, "SPI Data Clear(H'FF) Check :H'00000000-0003FFFF Clear OK?(y/n)" is displayed. Then input "y".
* After "SAVE SPI-FLASH ....... complete!" is displayed, the prompt returns. It means finish.
* Please repeat the xls2 command, if other files are written.
* Power OFF.
* Set dip switch to “Boot Mode”.

## IPL/Secure write

Please write the file described in Chapter 4.1 to SPI Flash.

The data file is stored in the ${WORK}/build/tmp/deploy/images/<board\_name> directory.

# Confirm starting of U-Boot and Linux

Please connect RZ/G2 System Evaluation Board, Windows Host PC with terminal software for console and Linux Host PC with TFTP and NFS server as Figure 1. Then please confirm normal starting of U-Boot and Linux with following step. Please refer to 2.2 for dip switch setting.

**Step 1 setting Linux Host PC**

Please install TFTP server and NFS server in Linux Host PC with apt-get command and so on. Please set /etc/xinetd.d/tftp of TFTP server and /etc/exports of NFS server according to your environment.

**Step 2 connect cable**

Connect USB Host connector of Windows Host PC that is virtual COM port to RZ/G2 System Evaluation Board with USB cable for displaying console.

**Step 3 setting the terminal software**

Activate the Terminal Software on Windows Host PC. Configure the Terminal Software on Windows Host PC as followings. Please refer to Table 1 about the VCP driver for making a USB host connector into a virtual COM port.

[setting value] baud rate 115200, 8bit data, parity none, stop 1 bit, flow control none.

**Step 4 write U-Boot to SPI Flash**

|  |  |  |  |
| --- | --- | --- | --- |
| Filename | Program Top Address | Flash Save Address | Description |
| u-boot-elf-\*.srec | 0x50000000 | 0x300000 | U-Boot |

Note) \*: ek874, hihope-rzg2m, hihope-rzg2n, hihope-rzg2h.

The data file is stored in the ${WORK}/build/tmp/deploy/images/<board\_name> directory.

Refer to Chapter 4.3 Step3 for write procedure.

**Step 5 set U-Boot environment variables**

Please refer to 2.2 for dip switch setting.

Please start U-Boot by board reset. Please set and save environment variable as follows.

**=> setenv ethaddr xx:xx:xx:xx:xx:xx**

**=> setenv ipaddr 192.168.0.20**

**=> setenv serverip 192.168.0.1**

**=> setenv bootcmd 'tftp 0x48080000 Image;tftp 0x48000000 Image-<SOC\_FAMILY>-<Device\_Tree>;booti 0x48080000 - 0x48000000'**

Note) <SOC\_FAMILY> is the following: r8a774c0, r8a774a1, r8a774a3, r8a774b1, r8a774e1.

Note) For RZ/G2E (SOC\_FAMILY r8a774c0), the device trees are as follow:

* Image-r8a774c0-ek874\*.dtb. (for latest version of LSI)
* Image-r8a774c0-esXX-ek874\*.dtb. (for an old version of LSI)

Note) For RZ/G2M v1.3 (SOC\_FAMILY r8a774a1), the device trees are as follow:

* Image-r8a774a1-hihope-rzg2m-ex\*.dtb (for latest version of Board)
* Image-r8a774a1-hihope-rzg2m-ex-rev2\*.dtb (for an old version of Board)

Note) For RZ/G2M v3.0 (SOC\_FAMILY r8a774a3), the device trees are as follow:

* Image-r8a774a3-hihope-rzg2m-ex\*.dtb (for latest version of Board)

Note) For RZ/G2N (SOC\_FAMILY r8a774b1), the device trees are as follow:

* Image-r8a774b1-hihope-rzg2n-ex\*.dtb (for latest version of Board)
* Image-r8a774b1-hihope-rzg2n-ex-rev2\*.dtb (for an old version of Board)

Note) For RZ/G2H (SOC\_FAMILY r8a774e1), the device trees are as follow:

Image-r8a774e1-hihope-rzg2h-ex\*.dtb (for latest version of Board)

**Step 6 change the bootargs by U-Boot**

To change bootargs which passed to the kernel in boot sequence, please modify it by “setenv bootargs” command of U-Boot.

**=> setenv bootargs 'rw root=/dev/nfs nfsroot=192.168.0.1:/export/rfs ip=192.168.0.20'**

**Step 7 save environment variables**

**=> saveenv**

**Step 8 set file system**

Please extract file system (core-image-weston(bsp|qt|hmi)-<supported board name>.tar.bz2). Please export /export directory of NFS server.

**$ mkdir /export/rfs**

**$ cd /export/rfs**

**$ sudo tar xvf core-image-weston(bsp|qt|hmi)-<supported board name>.tar.bz2**

Note) <supported board name> is the following: ek874, hihope-rzg2m, hihope-rzg2n, hihope-rzg2h.

**Step 9 start Linux**

After board reset, U-Boot is started. After countdown, Linux boot messages are displayed. Please confirm login prompt after Linux boot messages.

Note) When MAC Address is rewritten, it is necessary to reset.

# Exporting Toolchains

Please refer Documents from Yocto Project to export Toolchains such as

<http://www.yoctoproject.org/docs/current/adt-manual/adt-manual.html>.

And please use build target of bitbake as “core-image-weston(qt)-sdk -c populate\_sdk” to generate package.

Note) When you use “ld” directly but not via gcc (in case of building Kernel, Driver or U-Boot), please disable LDFLAGS with ‘unset LDFLAGS’. Furthermore, in kernel build, ‘make menuconfig’ occurs error by ncurses. In this case, please set PKG\_CONFIG\_PATH and disable PKG\_CONFIG\_SYSROOT\_DIR.

**$ export PKG\_CONFIG\_PATH=$OECORE\_NATIVE\_SYSROOT/usr/lib/pkgconfig  
$ unset PKG\_CONFIG\_SYSROOT\_DIR**

Note) Please do not use same shell environment to other compilation/debugging purpose (also make menuconfig of linux kernel, e.g.) but cross compilation for RZ/G2E|G2M|G2N|G2H which shell environment with “source” command to setup environment variables for the SDK. Because some environment variables for cross compilation interferes execution of other tools on the same shell environment.

**Example of instruction:**

In following examples, it’s assumed that it’s already extracted and prepared recipe environment such as in the instructions of Section 3 (must done just before execution of bitbake, at least). You may reuse ${WORK}/build while you reuse same configuration after executing bitbake as in Section 3 for this purpose.

**Step 1 configure architectures of Host PC which are installed this toolchain**

Please modify SDKMACHINE description on ${WORK}/build/conf/local.conf.

**On ${WORK}/build/conf/local.conf**

# This variable specified the architecture to build SDK/ADT items for and means

# you can build the SDK packages for architectures other than the machine you are

# running the build on (i.e. building i686 packages on an x86\_64 host.

# Supported values are i686 and x86\_64

#SDKMACHINE ?= "i686"

**SDKMACHINE ?= "x86\_64"**

Note) 32bit Ubuntu 14.04 is not supported.

**Step 2 building toolchain package with bitbake**

Note) Please perform “bitbake core-image-minimal -c populate\_sdk” in BSP Only.

**$ cd ${WORK}/build**

**$ bitbake core-image-weston(qt)-sdk -c populate\_sdk**

**$ cp tmp/deploy/sdk/poky-glibc-x86\_64-core-image-weston(qt)-sdk-** **aarch64-toolchain-2.4.3.sh (shared dir. where able to access from each Host PCs)**

**Step 3 Install toolchain on each Host PCs**

**$ sudo (shared dir. where able to access from each Host PCs)/poky-glibc-x86\_64-core-image-weston(qt)-sdk-aarch64-toolchain-2.4.3.sh**

[sudo] password for (INSTALL person): **(password of your account)**

Enter target directory for SDK (default: /opt/poky/2.4.3): **(just a return)**

Extracting SDK...done

Setting it up...done

When it request to re configure please just enter to keep default value

Compile also drivers which will not load (COMPILE\_TEST) [N/y/?] <Enter>

Local version - append to kernel release (LOCALVERSION) [-yocto-standard] -yocto-standard

Automatically append version information to the version string (LOCALVERSION\_AUTO) [Y/n/?] <Enter>

…

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 environment setup script e.g.

$ . /opt/poky/2.4.3/environment-setup-aarch64-poky-linux

$ . /opt/poky/2.4.3/environment-setup-armv7vehf-neon-pokymllib32-linux-gnueabi

**Step 4 setup environment variables for each compilation on each Host PCs**

Please setup environment variables as follows or integrate set-up sequence into your build script or Makefile.

**$ cd (Your working directory)**

**$ source /opt/poky/2.4.3/environment-setup-aarch64-poky-linux**

**$ export LDFLAGS=””**

**$ ${CC} (Your source code).c …..**

# Memory map

Following from Figure 2 to Figure 6 show memory map of this RZ/G2E|G2M|G2N|G2H Linux BSP package.

Note)

* The volume of SDRAM is total 2GB (RZ/G2E System Evaluation Board EK874), 4GB (RZ/G2M System Evaluation Board HiHope-RZG2M), 4GB (RZ/G2N System Evaluation Board HiHope-RZG2N), 4GB (RZ/G2H System Evaluation Board HiHope-RZG2H).
* 2GB from 0x00\_4000\_0000 to 0x00\_BFFF\_FFFF is a shadow area from 0x04\_0000\_0000 to 0x04\_7FFF\_FFFF.
* The following regions are used as a secure region. It doesn’t allow U-Boot and kernel to access those regions.
  + 63MB from 0x00\_43F0\_0000 to 0x00\_47DF\_FFFF in SDRAM
  + 16KB from 0x00\_E630\_0000 to 0x00\_E630\_3FFF in System RAM
* **In case the configuration of BSP + 3D Graphics + Multimedia package, it doesn't allow to store any data in "CMA for Lossy comp" (default: 0x00\_5400\_0000 - 0x00\_56FF\_FFFF) region which is for media playback before kernel boots up. Any data stored in this region are read through the decompression module in AXI-Bus, so a normal data (not a decoded frame) will be corrupted.**

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

SPI Flash

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_5000\_0000

ARM Trusted Firmware

U-Boot

ARM Trusted Firmware

Shadow area

Certification

IPL

Boot parameter

System RAM

Boot parameter

IPL

Load by Boot

ROM program

N/A

0x08\_0000\_0000

0x00\_E630\_0000

Secure

Region

Legacy

0x04\_8000\_0000

0x180000

0x040000

0x0

0x1C0000

0x300000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

N/A

N/A

Option

Secure

Region

0x00\_8000\_0000

0x04\_4000\_0000

U-Boot

Figure 2. RZ/G2E System Evaluation Board EK874 memory map (Boot)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

0x00\_4000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_4808\_0000

ARM Trusted Firmware

Shadow area

N/A

0x08\_0000\_0000

Secure

Region

Legacy

0x04\_8000\_0000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

N/A

N/A

CMA 256 MB

0x00\_C000\_0000

0x00\_5800\_0000

Option

0x00\_8000\_0000

0x04\_4000\_0000

0x00\_5700\_0000

0x00\_6800\_0000

0x00\_5400\_0000

dtb

Kernel Image

CMA for MMP 128 MB

0x00\_5020\_0000

CMA (256MB)

Figure 3. RZ/G2E System Evaluation Board EK874 memory map (Linux)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

Hyper Flash

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_5000\_0000

ARM Trusted Firmware

U-Boot

ARM Trusted Firmware

OP-Tee

Shadow area

Load by IPL

Certification

IPL

Boot parameter

System RAM

Boot parameter

IPL

Load by Boot

ROM program

N/A

U-Boot

0x08\_0000\_0000

0x00\_E630\_0000

Secure

Region

Legacy

0x04\_8000\_0000

0x180000

0x040000

0x0

0x1C0000

0x300000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

SDRAM 2GB

N/A

Option

Secure

Region

Figure 4. RZ/G2M System Evaluation Board (HIHOPE-RZG2M) memory map (Boot)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_4808\_0000

ARM Trusted Firmware

OP-Tee

Shadow area

N/A

dtb

0x08\_0000\_0000

Secure

Region

Legacy

0x04\_8000\_0000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

SDRAM 2GB

N/A

CMA for MMP

(256MB)

CMA

(512MB)

0x00\_7800\_0000

0x00\_8800\_0000

0x00\_5800\_0000

0x00\_5700\_0000

0x00\_5400\_0000

Option

Kernel Image

CMA for Lossy comp

(48MB)

Figure 5. RZ/G2M System Evaluation Board (HiHope-RZG2M) memory map (Linux)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

SPI Flash

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_5000\_0000

ARM Trusted Firmware

U-Boot

ARM Trusted Firmware

Shadow area

Certification

IPL

Boot parameter

System RAM

Boot parameter

IPL

Load by Boot

ROM program

N/A

0x08\_0000\_0000

0x00\_E630\_0000

Secure

Region

Legacy

0x04\_8000\_0000

0x180000

0x040000

0x0

0x1C0000

0x300000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

N/A

Option

Secure

Region

0x00\_8000\_0000

0x04\_4000\_0000

U-Boot

SDRAM 2GB

Figure 6. RZ/G2N System Evaluation Board (HiHope-RZG2N) memory map (Boot)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x06\_0000\_0000

0x07\_0000\_0000

N/A

0x06\_8000\_0000

Shadow area

N/A

0x08\_0000\_0000

Secure

Region

Legacy

0x04\_8000\_0000

N/A

0x00\_8000\_0000

0x04\_4000\_0000

0x00\_4000\_0000

0x00\_C000\_0000

0x00\_4800\_0000

0x00\_4808\_0000

ARM Trusted Firmware

OP-Tee

dtb

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

CMA for MMP

(256MB)

CMA

(512MB)

0x00\_7800\_0000

0x00\_8800\_0000

0x00\_5800\_0000

0x00\_5700\_0000

0x00\_5400\_0000

Option

Kernel Image

CMA for Lossy comp

(48MB)

SDRAM 2GB

Figure 7. RZ/G2N System Evaluation Board (HiHope-RZG2N) memory map (Linux)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x05\_8000\_0000

0x07\_0000\_0000

N/A

Hyper Flash

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_5000\_0000

ARM Trusted Firmware

U-Boot

ARM Trusted Firmware

OP-Tee

Shadow area

Load by IPL

Certification

IPL

Boot parameter

System RAM

Boot parameter

IPL

Load by Boot

ROM program

N/A

U-Boot

0x08\_0000\_0000

0x00\_E630\_0000

Secure

Region

Legacy

0x04\_8000\_0000

0x180000

0x040000

0x0

0x1C0000

0x300000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

SDRAM 2GB

N/A

Option

Secure

Region

N/A

0x06\_0000\_0000

Figure 8. RZ/G2H System Evaluation Board (HIHOPE-RZG2H) memory map (Boot)

BSC

0x0

Physical Address

Reserved

PCI-exp

0x00\_2000\_0000

0x00\_3000\_0000

SDRAM 2GB

Reserved

0x00\_4000\_0000

0x00\_C000\_0000

IO area

0x00\_E000\_0000

0x01\_0000\_0000

～

～

～

～

0x04\_0000\_0000

SDRAM 2GB

0x05\_0000\_0000

N/A

0x05\_8000\_0000

0x07\_0000\_0000

N/A

0x00\_4000\_0000

0x00\_C000\_0000

0x06\_8000\_0000

0x00\_4800\_0000

0x00\_4808\_0000

ARM Trusted Firmware

OP-Tee

Shadow area

N/A

dtb

0x08\_0000\_0000

Secure

Region

Legacy

0x04\_8000\_0000

Certification

0x00\_43F0\_0000

0x00\_47E0\_0000

SDRAM 2GB

N/A

CMA for MMP

(256MB)

CMA

(512MB)

0x00\_7800\_0000

0x00\_8800\_0000

0x00\_5800\_0000

0x00\_5700\_0000

0x00\_5400\_0000

Option

Kernel Image

CMA for Lossy comp

(48MB)

N/A

0x06\_0000\_0000

Figure 9. RZ/G2H System Evaluation Board (HiHope-RZG2H) memory map (Linux)

Note)

* Kernel region is assigned by Kernel device tree arch/arm64/boot/dts/renesas/xxx.dts and totally mapped to 1920MB (RZ/G2E System Evaluation Board EK874), 3968MB (RZ/G2M System Evaluation Board HiHope-RZG2M), 3968MB (RZ/G2N System Evaluation Board HiHope-RZG2N), 3968MB (RZ/G2H System Evaluation Board HiHope-RZG2H)

Kernel region consists of 1 part: (RZ/G2E System Evaluation Board EK874)

* + 1920MB from 0x00\_4800\_0000 to 0x00\_BFFF\_FFFF

Kernel region consists of 2 part: (RZ/G2M System Evaluation Board HiHope-RZG2M)

* + 1920MB from 0x00\_4800\_0000 to 0x00\_BFFF\_FFFF
  + 2GB from 0x06\_0000\_0000 to 0x06\_7FFF\_FFFF

Kernel region consists of 2 part: (RZ/G2N System Evaluation Board HiHope-RZG2N)

* + 1920MB from 0x00\_4800\_0000 to 0x00\_BFFF\_FFFF
  + 2GB from 0x04\_8000\_0000 to 0x04\_EFFF\_FFFF

Kernel region consists of 2 part: (RZ/G2H System Evaluation Board HiHope-RZG2H)

* + 1920MB from 0x00\_4800\_0000 to 0x00\_BFFF\_FFFF
  + 2GB from 0x05\_0000\_0000 to 0x05\_7FFF\_FFFF

There are three types of CMA regions.

They are defined in device tree (arch/arm64/boot/dts/renesas/xxxx.dts).

* + Default CMA region: It is for kernel, general drivers and multimedia package.

|  |
| --- |
| linux,cma {  compatible = "shared-dma-pool";  reusable;  reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;  linux,cma-default;  };  0xXXXXXXXX is start address of CMA region.  0xYYYYYYYY is size of CMA region. |

Note)

* + - * 128 MB in this CMA (RZ/G2M (v1.3, v3.0) |G2N|G2H 512MB, RZ/G2E 256MB) is reserved for kernel and general drivers, and the remaining RZ/G2M (v1.3, v3.0) |G2N|G2H 384 MB, RZ/G2E 128MB is reserved for multimedia package.
      * The CMA region can be adjusted by changing the start address and the size.
      * Should take care of the lack of memory allocated by kernel and general drivers when reducing the region size.
  + CMA region for MMP: It is for multimedia package (specific H/Ws).

|  |
| --- |
| mmp\_reserved: linux,multimedia {  compatible = "shared-dma-pool";  reusable;  reg = <0x00000000 0xXXXXXXXX 0x0 0xYYYYYYYY>;  };  0xXXXXXXXX is start address of CMA region.  0xYYYYYYYY is size of CMA region. |

Note)

* + - * Refer to User’s manual of Memory Manager in order to change CMA region for MMP.

Virtual Address

0x0

0x0001\_0000\_0000\_0000

0xFFFF\_FFFF\_FFFF\_FFFF

0xFFFF\_0000\_0000\_0000

N/A

User

(256TB)

Kernel

(256TB)

vmalloc

vmemmap

･･･

Kernel

･･･

Figure 10 . RZ/G2 memory map (Virtual)

Note)

* Kernel uses 4KB page size (VA\_BITS=48) and 4 levels of translation tables. Both regions of User and Kernel are 256TB. Refer to Documentation/arm64/memory.txt.
* Detail information about kernel memory map in virtual address space, refer to User’s manual of Kernel.

# U-Boot command

Please refer to U-Boot user's manual about available U-Boot command for RZ/G2 Linux BSP.   
The help or “?” command shows U-Boot command list, but be careful that it includes some unsupported command.

# System Service

In RZ/G2 VLP64 environment, some services are added to root filesystem to assist related drivers.

## Watchdog Service

Watchdog Service is a systemd service, which is used to generate a reset when system is freeze.

It is automatically loaded in root filesystem and runs background during operation with binary file:

/usr/bin/watchdog-test -d -t 60 -e

**/usr/bin/watchdog-test -d -t 60 -e**

This binary is compiled from “tools/testing/selftests/watchdog/watchdog-test.c” in kernel source code.

The meaning of each parameter:

* -d: Turn off the watchdog timer
* -t: set time out to 60s
* -e: Turn on the watchdog timer
* -p: set ping rate (default value is 1s if not set)

To control Watchdog Service interface, we can refer similar commands of systemd service on linux. The following table show supported commands:

Table 2 Supported commands

|  |  |
| --- | --- |
| Command | Description |
| systemctl stop watchdog | Stop Watchdog Service |
| systemctl start watchdog | Start Watchdog Service |
| systemctl restart watchdog | Restart Watchdog Service |
| systemctl disable watchdog | Disable Watchdog Service in root filesystem |
| systemctl enable watchdog | Enable Watchdog Service in root filesystem |

Watchdog Service is automatically activated by default. To turn off Watchdog Service, please choose one of the following ways:

* Stop Watchdog Service in runtime (Turn off only once. If you reset or powering up board again, watchdog service still starts again):
  + systemctl stop watchdog.
* Disable Watchdog Service in runtime (Turn off completely. If you reset or powering up board again, watchdog service does not start):
  + systemctl stop watchdog.
  + systemctl disable watchdog.
* Remove Watchdog Service from BSP packages: remove “watchdog” package in “recipes-images/core-image-renesas-base.inc”

**IMAGE\_INSTALL\_append = " \**

**bash \**

**v4l-utils \**

**i2c-tools \**

**coreutils \**

**- watchdog \**

**"**

## Video Input Initializing Service

In RZ/G2 environment, there is a service named “vin” which automatically sets connecting among VIN, CSI2 and OV5645 camera sensor.

This script automatically sets default links for:

|  |  |  |
| --- | --- | --- |
| **Processor** | **CSI20** | **CSI40** |
| RZ/G2H | VIN4/VC0 | VIN0/VC0 |
| RZ/G2M v1.3 | VIN4/VC0 | - |
| RZ/G2M v3.0 | VIN4/VC0 | VIN0/VC0 |
| RZ/G2N | VIN4/VC0 | VIN0/VC0 |
| RZ/G2E | - | VIN4/VC0 |

To change the connection in runtime, user can use a script at **“/home/root/vin-init.sh”** created by vin service then modify some parameters which are described in this script based on user’s purpose and connection tables (Table 4.3, 4.4, 4.5, 4.6 in Video Capture User’s Manual).

To remove “vin” service from BSP packages, please remove “vin-init” package in “recipes-images/core-image-bsp.inc”:

**IMAGE\_INSTALL\_append = " \**

**….**

**rt-tests \**

**ltp \**

**openssl \**

**- vin-init \**

**"**