RZ/V2L DRP-AI Support Package Version.7.41

Release Note

Introduction

This release note describes the contents of the RZ/V2L DRP-AI Support Package and how to construct its operating environments.

Contents

1. Release Items

The release items in the RZ/V2L DRP-AI Support Package are as follows.

- **Name and Version**

    RZ/V2L DRP-AI Support Package Version.7.41

- **Target Board**

    RZ/V2L Evaluation Board Kit

    Note: For the settings of RZ/V2L Evaluation Board Kit, please refer to the SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).

    The CMOS sensor (OV5645) in the Coral camera is no longer available and should not be used for mass production.

    Any software support provided is for evaluation purposes only.

- **Features**

    1. RZ/V2L DRP-AI Driver

        This is a Linux Package Recipe to use DRP-AI on RZ/V2L.

    2. RZ/V2L, RZ/V2M, RZ/V2MA AI Implementation Guide

        This is a guide on how to implement the AI model in RZ/V2L, RZ/V2M and RZ/V2MA.

    3. RZ/V2L AI Evaluation Software

        This is a software to evaluate AI models on RZ/V2L without coding.

    4. RZ/V2L DRP-AI Sample Application

        This is a set of sample applications for DRP-AI on RZ/V2L.

    5. RZ/V2L, RZ/V2M, RZ/V2MA DRP-AI Accuracy Report

        This is a report of evaluating the accuracy of AI models on DRP-AI.

    6. RZ/V2L, RZ/V2M, RZ/V2MA DRP-AI Performance Report

        This is a report of evaluating the performance of AI models on DRP-AI.

    Notes: Updated RZ/V Verified Linux Package V3.0.4 to RZ/V Verified Linux Package V3.0.5.

- **File Contents**

Table 1-1 shows the list of contents in this package.

Table 1-1 Contents list

Contents	Explanation
▯ r11an0549ej0741-rzv2l-drpai-sp.pdf	This document.
▭ rzv2l_drpai-driver	RZ/V2L DRP-AI Driver.
▯ meta-rz-drpai.tar.gz	Recipe supporting Verified Linux Package V3.0.5. Include the following: - DRP-AI Driver V2.10 - AI Evaluation Software V7.40 - Other OSS
▯ r01us0595ej0201-rzv-linux-drpai.pdf	DRP-AI Driver User's Manual (English).
▭ rzv_ai-evaluation-software	AI Evaluation Software.
▯ r11an0624jj0740-rzv-ai-eva-soft-guide.pdf	Document (Japanese).
▯ r11an0624ej0740-rzv-ai-eva-soft-guide.pdf	Document (English).
▯ rzv2l_ai-evaluation-software_ver7.40.tar.gz	Software environment.
▭ rzv2l_drpai-sample-application	DRP-AI Sample Application.
▯ r11an0573ej0740-rzv2l-drpai-application-note.pdf	Document (English).
▯ rzv2l_drpai-sample-application_ver7.40.tar.gz	Sample code.
▭ rzv_ai-implementation-guide	AI Implementation Guide.
▯ r11an0616jj0740-rzv-ai-imp-getstarted.pdf	Document (Japanese).
▯ r11an0616ej0740-rzv-ai-imp-getstarted.pdf	Document (English).
▯ rzv_ai-implementation-guide_ver7.40.tar.gz	Sample code.
▭ pytorch_resnet	Sample of PyTorch ResNet.
▯ r11an0617jj0740-rzv-ai-imp-resnet.pdf	Document (Japanese).
▯ r11an0617ej0740-rzv-ai-imp-resnet.pdf	Document (English).
▯ pytorch_resnet_ver7.40.tar.gz	Sample code.
▭ pytorch_mobilenet	Sample of PyTorch MobileNet.
▯ r11an0618jj0740-rzv-ai-imp-mobilenet.pdf	Document (Japanese).
▯ r11an0618ej0740-rzv-ai-imp-mobilenet.pdf	Document (English).
▯ pytorch_mobilenet_ver7.40.tar.gz	Sample code.
▭ pytorch_deeplabv3	Sample of PyTorch DeepLabV3.
▯ r11an0619jj0740-rzv-ai-imp-deeplabv3.pdf	Document (Japanese).
▯ r11an0619ej0740-rzv-ai-imp-deeplabv3.pdf	Document (English).
▯ pytorch_deeplabv3_ver7.40.tar.gz	Sample code.
▭ darknet_yolo	Sample of Darknet YOLOv3/v2, Tiny YOLOv3/v2.
▯ r11an0620jj0740-rzv-ai-imp-yolo.pdf	Document (Japanese).
▯ r11an0620ej0740-rzv-ai-imp-yolo.pdf	Document (English).
▯ darknet_yolo_ver7.40.tar.gz	Sample code.
▭ mmpose_hrnet	Sample of MMPose HRNet.
▯ r11an0621jj0740-rzv-ai-imp-hrnet.pdf	Document (Japanese).
▯ r11an0621ej0740-rzv-ai-imp-hrnet.pdf	Document (English).
▯ mmpose_hrnet_ver7.40.tar.gz	Sample code.
▭ Appendix	AI Implementation Guide appendix.
▯ r11an0622jj0740-rzv-ai-imp-appendix.pdf	Document for Tensorflow MobileNet (Japanese).
▯ r11an0622ej0740-rzv-ai-imp-appendix.pdf	Document for Tensorflow MobileNet (English).
▭ rzv_drpai-report	DRP-AI Reports.
▯ r11an0657ej0740-rzv-drpai-performance-report.pdf	DRP-AI Performance Report (English).
▯ r11an0628ej0720-rzv-drpai-accuracy-report.pdf	DRP-AI Accuracy Report (English).

- **Related Packages**

    1. RZ/V Verified Linux Package ver3.0.5 or later
    2. DRP-AI Translator v1.83 or later
    3. RZ/V2L ISP Support Package ver1.30 or later

- **Related Documents**

    1. RZ/V Verified Linux Package Release Note (R01US0565)
    2. SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617)
    3. DRP-AI Translator User's Manual (R20UT5010)
    4. RZ/V2L ISP Support Package Release Note (R11AN0561)

2. Operating Environment

This package uses two different environments.

Note: For the environment of AI Evaluation Software, please refer to the AI Evaluation Software Guide.

2.1   Translate Environment

This is the environment for translating AI model on Linux PC (Ubuntu 20.04 LTS).

Please refer to AI Implementation Guide for details of this environment.

2.2   Build Environment

This is the environment for building Linux Package of RZ/V2L on Linux PC (Ubuntu 20.04 LTS).

Please refer to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617) and 3 Build for the details of this environment.

Note: **In order to run DRP-AI Sample Application for MIPI/USB Camera version, please build with Mali Graphic Library**. Please refer to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617) for more information.

Note: **In order to run DRP-AI Sample Application for ISP version, both RZ/V2L DRP-AI Support Package and RZ/V2L ISP Support Package must be deployed to the Build Environment.** Please refer to RZ/V2L ISP Support Package Release Note (R11AN0561) to prepare the Build Environment for RZ/V2L ISP Support Package. **For application other than ISP version, please do not include ISP patch in the booting environment.**

2.3   Boot Environment

This is the environment for booting the Linux OS on RZ/V2L.

There are two booting methods, Network and SD card.

1.   Network Booting environment

     Please refer to 4.Network Booting for the details of this environment.

2.   SD Card Booting environment

     Please refer to 5.SD Card Booting for the details of this environment.

2.4   Compile Environment

This is the environment for cross-compiling the Linux application on Linux PC (Ubuntu 20.04 LTS) using RZ/V2L Linux Standard Development Kit (SDK).

Please refer to 6. SDK for the details of this environment.

Note: **In order to compile DRP-AI Sample Application for ISP version, both RZ/V2L DRP-AI Support Package and RZ/V2L ISP Support Package must be deployed to the Compile Environment.** Please refer to RZ/V2L ISP Support Package Release Note (R11AN0561) to prepare the Compile Environment for RZ/V2L ISP Support Package.

2.5   Execution Environment

This is the environment for executing the Linux application on RZ/V2L.

Please refer to RZ/V2L DRP-AI Sample Application Note for the details of this environment.

3. Build

This chapter will explain the procedure to deploy DRP-AI Driver to RZ/V2L Linux Package.

Before reading this chapter, please allocate RZ/V2L Linux Package and prepare the configuration files (bblayers.conf and local.conf) as explained in the SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).

Note: Before executing bitbake commands, apply all necessary recipes required for the Build Environment.

If you use the DRP-AI Sample Application to display on the HDMI monitor (MIPI/USB Camera version), please build with Mali Graphic Library. Please refer to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617) for more information.

For DRP-AI Sample Application for ISP version, please also add the recipes of RZ/V2L ISP Support Package to the Build Environment. For how to add the recipes, please refer to RZ/V2L ISP Support Package Release Note (R11AN0561). **For application other than ISP version, please do not include ISP patch in the booting environment.**

3.1 Preparation

3.1.1 Operating Environment

Please refer to the SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617) for the operating environment.

3.1.2 Recipe Configuration

This chapter uses meta-rz-drpai.tar.gz, which content is shown below.

```
📁 meta-rz-features
    📁 meta-rz-drpai
        📁 conf
                📄 layer.conf
        📁 include
                📄 core-image-sdk.inc
                📄 extend_packages.inc
        📁 recipes-core
            📁 images
                📄 core-image-%.bbappend
        📁 recipes-kernel
            📁 linux
                📄 linux-renesas_5.10.bbappend
                📁 linux-renesas
                📁 Kernel-module-udmabuf
                    📄 kernel-module-udmabuf.bb
                    📁 kernel-module-udmabuf
        📁 recipes-drpai
            📁 drpai
                📄 drpai_2.1.0.bb
                📁 files
                    📄 COPYING
                    📄 drpai.h
        📁 recipes-app
            📁 ai-eva-sw
                📄 ai-eva-sw_7.4.0.bb
                📁 ai-eva-sw
                    📄 ai-eva-sw_7.4.0.tar.bz2
        📁 recipes-support
            📁 opencv
                📄 opencv_%.bbappend
```
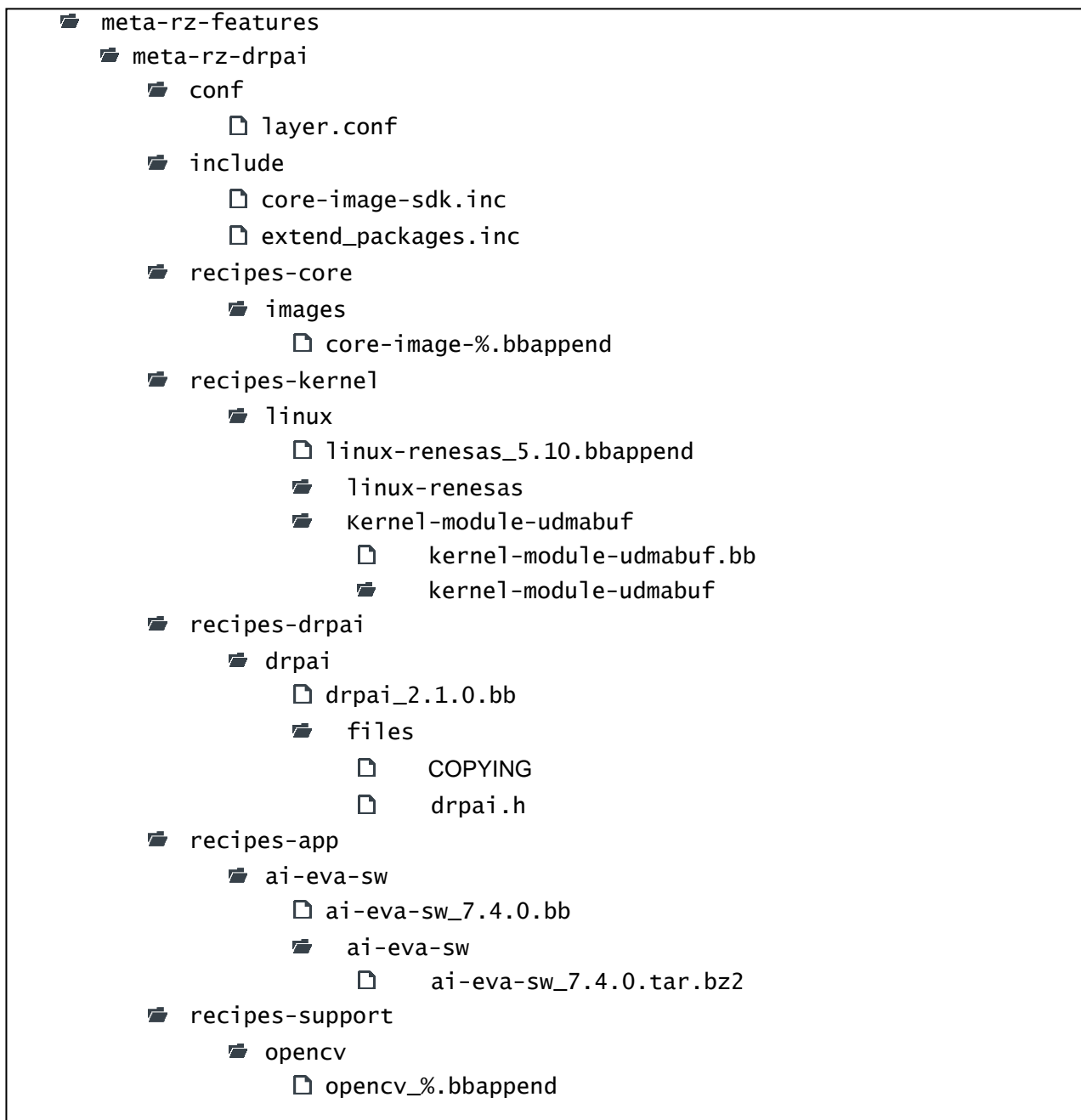
Figure 3-1 Directory Structure of Recipe

3.2 Build Instructions

3.2.1 Install Software Package

Please install the following package for environment setup on Ubuntu PC. These are common necessary software package for this chapter.

```
$ 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 \
libsdl1.2-dev xterm p7zip-full libyaml-dev libssl-dev
```

3.2.2 Add the Environmental Variable

Set the working directory as the environmental variable.

```
$ export WORK=/home/user/user_work
```

Note: Specify the working directory in red above according to your machine. Example above uses "user/user_work".

3.2.3 Unzip the DRP-AI Support Package Recipe

Place the rzv2l_drpai-driver directory to the working directory and run the following command.

```
$ cd $WORK
$ tar -zxvf ./rzv2l_drpai-driver/meta-rz-drpai.tar.gz
```

After executed the command, meta-rz-drpai directory file will be shown in the meta-rz-features directory.

```
📁 WORK
    📁 build
    📁 meta-gplv2
    📁 meta-openembedded
    📁 meta-qt5
    📁 meta-renesas
    📁 meta-rz-features
        📁 meta-rz-bootloaders
        📁 meta-guis
        📁 meta-rz-drpai
        📁 meta-rz-graphics
    📁 meta-virtualization
    📁 poky
    📁 rzv2l_drpai-driver
        📄 meta-rz-drpai.tar.gz
```

3.2.4　Set the Build Environment Variable

Run the following command to set the environment variable for the build.

Note:　The environmental variable will be reset if the terminal is closed. Please run the command for each time you open the terminal.

```
$ cd $WORK
$ TEMPLATECONF=$PWD/meta-renesas/meta-rzv2l/docs/template/conf/ source \
poky/oe-init-build-env build
$ bitbake-layers add-layer ../meta-rz-features/meta-rz-drpai
```

3.2.5　Decompress OSS files to "build" directory (Optional)

Run the commands below. This step is not mandatory and able to go to 3.2.6 Build in case the "offline" environment is not required. All OSS packages will be decompressed with this '7z' command.

```
$ cp ../../<package download directory>/*.7z .
$ 7z x oss_pkg_rzv_drpai_v7.41.7z
```

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.

Open source software packages are required for an "offline" environment. The word "offline" means an isolated environment which does not connect to any network. 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.

After the above procedure is finished, the "offline" environment is ready. If you want to prevent network access, please change the line in the "~/rzv_vlp_<package version>/build/conf/local.conf" as below:

```
BB_NO_NETWORK = "1"
```

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

3.2.6　Build

Run the bitbake command to build the Linux Package.

```
$ cd $WORK/build
$ MACHINE=smarc-rzv2l bitbake core-image-weston
```

After the Build, following files will be generated under `$WORK/build/tmp/deploy/images/smarc-rzv2l`.

Table 3-1 Generated Files after Build

Filename	Name
Image-smarc-rzv2l.bin	Linux Kernel Image
Image-r9a07g054l2-smarc.dtb	Linux Device Tree File
core-image-weston-smarc-rzv2l.tar.bz2	A set of root filesystem

3.2.7 HDMI monitor resolution

In RZ/V2L Linux Package, the HDMI monitor resolution is set to FHD. Since the RZ/V2L DRP-AI Sample Application uses HD resolution as default, we recommend changing the HDMI monitor resolution from FHD to HD. To change it, change /etc/xdg/weston/weston.ini of the root filesystem as follows.

```
[core]
idle-time=0
require-input=false
repaint-window=17
[output]
name=HDMI-A-1            ⎫
mode=1280x720           ⎬ Add
                        ⎭
```

4. Network Booting

This chapter will explain the procedure for the Network Booting.

Network Booting is a booting method that mount the server on Ubuntu PC to access the files which are extracted to memory on the RZ/V2L Evaluation Board Kit.

4.1 Hardware Configuration

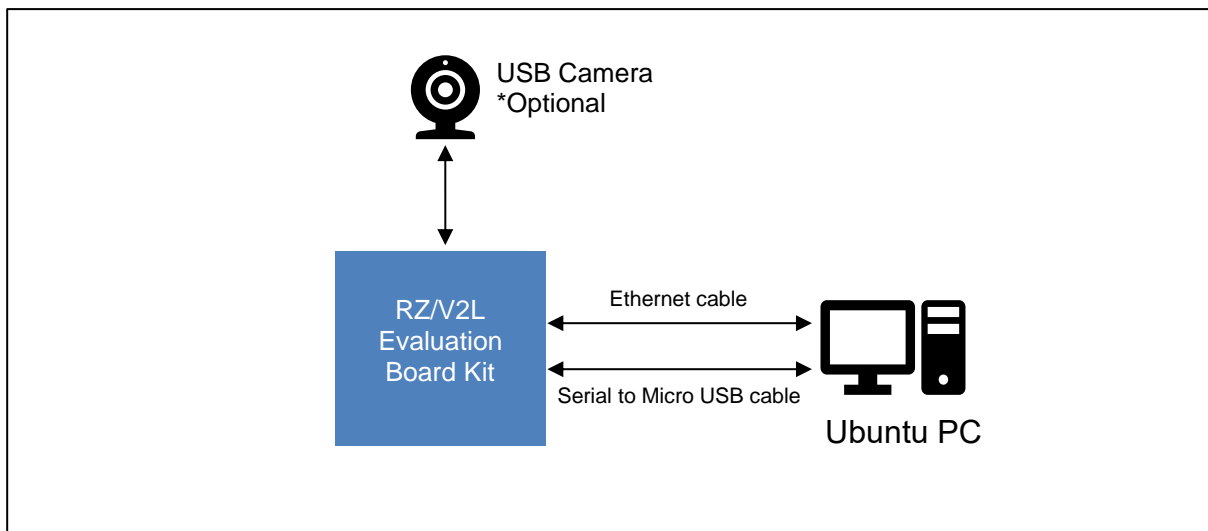Figure 4-1 shows the hardware configuration for the Network Booting.

Figure 4-1 Hardware Configuration

4.2 Preparation

4.2.1 Equipment

Necessary equipment for network booting is as follows.

Table 4-1 Necessary Equipment for Network Booting

Equipment	Details
RZ/V2L Evaluation Board Kit	**Please rewrite RZ/V2L Linux bootloader by referring to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).**
Linux PC	Used as build/debug environment for RZ/V2L Linux software.
- OS	Ubuntu 20.04 LTS 64-bit OS must be used.
- TFTP server	Used for downloading the Linux kernel to board.
- NFS server	Used for mounting rootfs via NFS.
Serial to Micro USB Cable	Used for serial communication between PC and board.
Ethernet cable	Used for ethernet communication between Linux PC and board.
USB camera	(Optional) Used for image capture. Operation Environment: Logitech C930E WEBCAM

4.2.2 Files for Booting

Table 4-2 shows the necessary files for booting and their mounted server.

Table 4-2 Necessary Files for Network Booting

Filename	Description	Mounted Server
Image-smarc-rzv2l.bin	Linux Kernel Image (The boot program)	TFTP server on Ubuntu PC
Image-r9a07g054l2-smarc.dtb	Linux Device Tree File (The configuration file for booting)	TFTP server on Ubuntu PC
core-image-weston-smarc-rzv2l.tar.bz2	A set of root filesystem	NFS server on Ubuntu PC

Note: All files listed above are created in 3 Build.

Necessary files listed above will be mounted to RZ/V2L Evaluation Board Kit as described below.
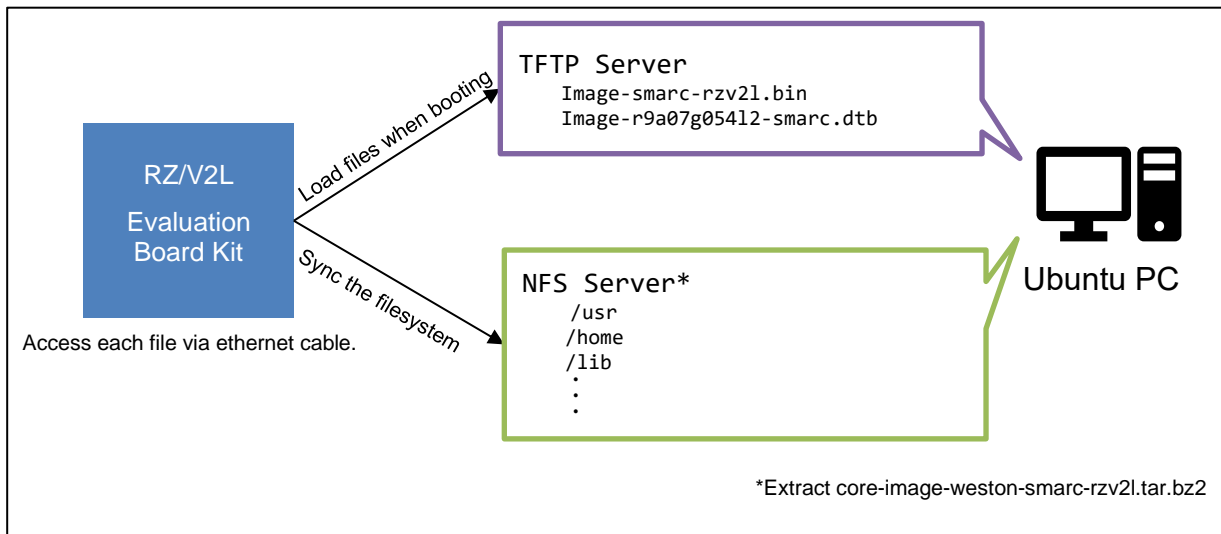

Figure 4-2 Network Booting Overview

4.2.3 Software Package

Please install the following packages and packages used in the 3.2.1 Install Software Package.

These are common necessary software package for this chapter.

```
$ sudo apt-get install tftp tftpd-hpa nfs-common nfs-kernel-server cu
```

4.3 Setup

4.3.1 Build TFTP Server

This section will explain how to build TFTP server which will mount the boot program.

Please complete the instruction in 4.2.3. Software Package before reading this section.

This section uses following packages.

- tftp

- tftpd-hpa

1. Create a directory for TFTP server.

```
$ sudo mkdir /tftpboot
```

   Note: Directory name will be used at booting step. Please use the above directory name.

2. Setup the TFTP server configuration file.
    Create /etc/default/tftpd-hpa file and write the following configuration.

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS=":69"
TFTP_OPTIONS="--secure"
```


3. Start TFTP server.

```
$ sudo systemctl enable tftpd-hpa
$ sudo systemctl restart tftpd-hpa
```


4. Confirm the TFTP server is successfully started.
    Execute the following commands.

```
$ sudo chmod 777 /tftpboot
$ sudo echo "Hello" > /tftpboot/hello.txt
$ sudo tftp localhost
> get hello.txt
```


If no error message is displayed, TFTP server is successfully started.

Note: If above command did not show the expected result, please restart the Ubuntu PC, and try it again.

4.3.2   Setup the Files for TFTP Server

Extract following two files on the TFTP server.

- Image-smarc-rzv2l.bin
- Image-r9a07g054l2-smarc.dtb

Please execute the following command on Ubuntu PC, which the TFTP server is already started.

Note: <PATH_to_FILE> is a path to the above files.

```
$ sudo cp <PATH_to_FILE>/Image-smarc-rzv2l.bin /tftpboot
$ sudo cp <PATH_to_FILE>/Image-r9a07g054l2-smarc.dtb /tftpboot
```

4.3.3   Build NFS Server

This section will explain how to build the NFS server that will mount the root filesystem.

Please complete the instruction in 4.2.3. Software Package before reading this section.

This section uses following packages.

- nfs-common
- nfs-kernel-server

1.   Start the NFS server.

```
$ sudo /etc/init.d/nfs-kernel-server start
```

2.   Create a directory for NFS server.

```
$ sudo mkdir /nfs/rzv2l -p
```

   Note: Directory name will be used at booting step. Please use the above directory name.

3.   Modify the NFS server configuration.
     Add the following line at the end of `/etc/exports` file.

```
…
/nfs/rzv2l *(rw,no_subtree_check,sync,no_root_squash)
```

4.   Refresh the NFS server.

```
$ sudo exportfs -a
```

5.   Confirm that the NFS server is successfully started.
     Execute the following command. If the same result is shown, the NFS server is successfully started.

```
$ showmount -e localhost
Export list for localhost:
/nfs/rzv2l *
```

Note: If above command did not show the expected result, please restart the Ubuntu PC, and try it again.

4.3.4　Setup the Files for NFS Server

Extract the following file on the NFS server.

- core-image-weston-smarc-rzv2l.tar.bz2

Please execute the following command on Ubuntu PC, which the NFS server is already started.
Note:　<PATH_to_FILE> is a path to the above files

```
$ sudo tar xfj <PATH_to_FILE>/core-image-weston-smarc-rzv2l.tar.bz2 -C /nfs/rzv2l
```

4.3.5　Setup the Static IP Address

In order to have ethernet communication between Ubuntu PC and RZ/V2L Evaluation board Kit, the IP address of Ubuntu PC must be static.

1. Disables the default network settings. The yaml filename may be different depending on the environment.

```
$ sudo mv /etc/netplan/01-network-manager-all.yaml /etc/netplan/01-network-manager-all.yaml.disabled
```

2. Create /etc/netplan/99-netcfg.yaml and add the following line.

```
network:
    version: 2
    ethernets:
        enp0s3:
            addresses: [192.168.1.10/24]
            gateway4: 192.168.1.1
            nameservers:
                addresses: [192.168.1.1]
                search: []
            optional: true
```

Note1: The name "enp0s3" may be different depending on the environment.
Note2: Since IP address stated here will be used when booting, please use the above address.

3. Restart the network.

```
$ sudo netplan apply
```

4.3.6 Setup the Serial Communication

Please execute the instructions explained in this section on Ubuntu PC.

Network booting assumes the serial communication will be done between Ubuntu PC and RZ/V2L Evaluation Board Kit.
Please complete the instruction in 4.2.3. Software Package before reading this section.

This section uses following packages.

- cu

1.   Connect Ubuntu PC and RZ/V2L Evaluation Board Kit with the Serial to Micro USB Cable.

2.   On Ubuntu PC, retrieve the name of serial port by executing the following command.

```
$ ls -l /dev/serial/by-id/
total 0
…-> ../../ttyUSB0
```

3.   Change the permission of serial port.

```
$ sudo chmod 666 /dev/ttyUSB0
```

4.   Execute the following command to start the serial communication.

```
$ cu -s 115200 -l /dev/ttyUSB0 --parity none --nostop
```

The configuration for serial communication is as follows.

Configuration Item	Value
Baud rate	115200
Data bit	8bit
Parity	none
Stop bit	1bit

5.   Open another console on the Ubuntu PC and change "crtscts" option.

```
$ stty -F /dev/ttyUSB0 -crtscts
```

6.   To end the communication, enter "~.".

4.4 Boot

This chapter will explain how to boot from the network.

Board information such as switch configuration will not be explained.

If you use the previous version of the RZ/V2L Linux Package, rewrite the loader files to eMMC.

For more details about switch configuration and how to write the loader files, please refer to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).

1. Setup the hardware according to 4.1 Hardware Configuration.

2. Turn on the Ubuntu PC and start the server.

    Normally, the PC automatically starts the server.

3. Carry out the serial communication according to 4.3.6 Setup the Serial Communication.

4. Turn on the RZ/V2L Evaluation Board Kit and change the U-boot environment variable.

    Note: Only when switching the booting method from SD Card Booting

    (1)   After turned on the RZ/V2L Evaluation Board Kit, keep pressing the ENTER key.

    (2)   The U-boot console will be activated.

```
=> env default -a
=> setenv ipaddr 192.168.1.11
=> setenv serverip 192.168.1.10
=> setenv netmask 255.255.255.0
=> setenv ethaddr 02:11:22:33:44:55
=> setenv boot_tftp 'tftpboot 0x48080000 Image-smarc-rzv2l.bin; tftpboot
0x48000000 Image-r9a07g054l2-smarc.dtb; booti 0x48080000 - 0x48000000'
=> setenv bootargs root=/dev/nfs rw nfsroot=${serverip}:/nfs/rzv2l,nfsvers=3
ip=${ipaddr}:${serverip}::${netmask}:rzv2l:eth0
=> setenv bootcmd run boot_tftp
=> saveenv
=> boot
```

    (3)   Enter the following commands.

           Details of above setting are as follow.

           `ipaddr`   : IP address of RZ/V2L Evaluation Board Kit

           `serverip`   : IP address of Ubuntu PC (IP address defined in 4.3.5 Setup the Static IP Address)

           `bootcmd`   : boot command

5. After the boot-up, login screen will be shown on serial communication console.

    Login information is as follows.

        `user: "root"`
        `password: none`

5. SD Card Booting

This chapter will explain about the SD Card Booting.

SD Card Booting is a booting method that mount the microSD card to access the files which are extracted to memory on the RZ/V2L Evaluation Board Kit.

5.1 Hardware Configuration

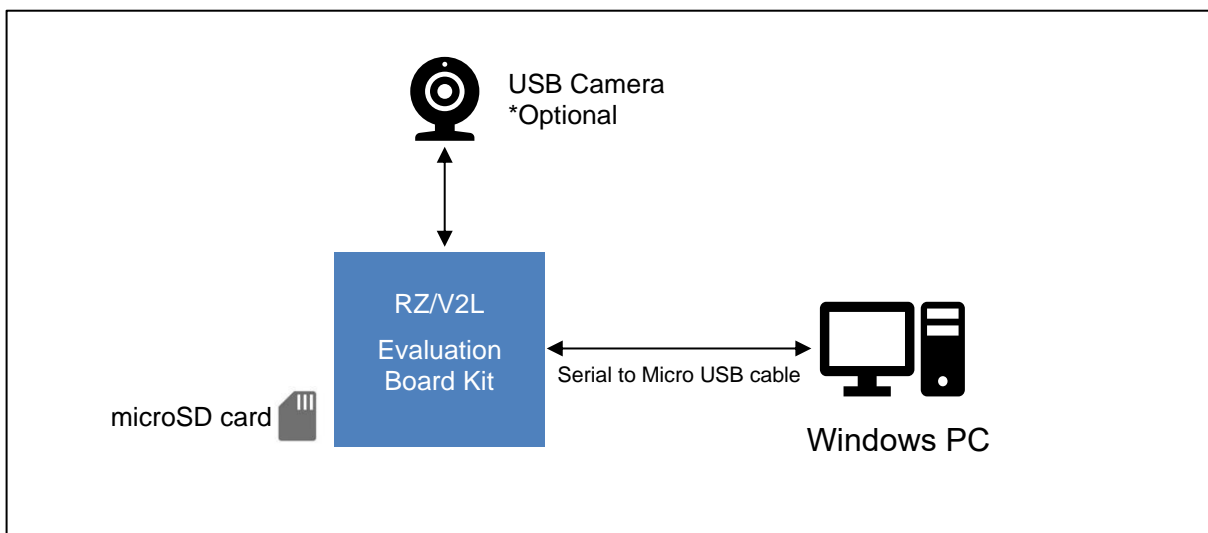Figure 5-1 shows the hardware configuration for SD Card Booting.

Figure 5-1 Hardware Configuration

5.2   Preparation

5.2.1   Equipment

Necessary equipment for SD Card booting is as follows.

Table 5-1 Necessary Equipment for SD Card Booting

Equipment	Details
RZ/V2L Evaluation Board Kit	**Please rewrite RZ/V2L Linux bootloader by referring to SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).**
Linux PC	Used for creating microSD card.
- OS	Ubuntu 20.04 LTS 64-bit OS must be used.
- SD card reader	Used for creating microSD card.
Windows PC	Used for Serial communication display.
- OS	Windows 10
- Terminal software	Used for controlling serial console of the target board. Operation Environment: Tera Term
- Serial port driver	Virtual COM Port driver which enables to communicate Windows PC and the target board via USB which is virtually used as serial port.
Serial to Micro USB Cable	Used for serial communication between PC and board.
microSD card	Used for SD Card Booting. Operation Environment: Transcend USH-I microSD 300S 16GB
USB camera	(Optional) Used for image capture. Operation Environment: Logitech C930E WEBCAM

5.2.2   Files for Booting

Table 5-2 shows the necessary files for SD Card Booting and their mounted partition on microSD card.

Table 5-2 Necessary Files for SD Card Booting

Filename	Description	Mounted Server
Image-smarc-rzv2l.bin	Linux Kernel Image (The boot program)	Partition 1
Image-r9a07g054l2-smarc.dtb	Linux Device Tree File (The configuration file for booting)	Partition 1
core-image-weston-smarc-rzv2l.tar.bz2	A set of root filesystem	Partition 2

Note: All files listed above are created in 3 Build.

5.2.3   Software Package

Please install the packages used in the 3.2.1 Install Software Package before executing this chapter.

5.3   Setup

5.3.1   Setup the SD Card

Please execute the instructions explained in this section on Ubuntu PC.

1.   Format the microSD card as shown in the following table.

Table 5-3 Format of MicroSD Card

Partition No.	Size	Format
1	128MB or more	fat32
2	The rest	ext4

2.   Setup the partition 1
   Confirm the microSD card is inserted to Ubuntu PC and execute the following command.
   Notes: /dev/sdb is the device number for microSD card. Change it according to your system.
         <PATH_to_FILE> is a path to each file.

```
$ sudo mkdir -p /mnt/sd
$ sudo mount /dev/sdb1 /mnt/sd
$ sudo cp <PATH_to_FILE>/Image-smarc-rzv2l.bin /mnt/sd
$ sudo cp <PATH_to_FILE>/Image-r9a07g054l2-smarc.dtb /mnt/sd
$ sync
$ sudo umount /mnt/sd
```

3.   Setup the partition 2
   Confirm the microSD card is inserted to Ubuntu PC and execute the following command.
   Notes: /dev/sdb is the device number for microSD card. Change it according to your system.
         <PATH_to_FILE> is a path to each file.

```
$ sudo mount /dev/sdb2 /mnt/sd
$ sudo tar xfj <PATH_to_FILE>/core-image-weston-smarc-rzv2l.tar.bz2 -C /mnt/sd
$ sync
$ sudo umount /mnt/sd
```

4.   Eject the microSD card from Ubuntu PC.

5.3.2   Install the Serial Port Driver

In order to carry out the serial communication between Windows PC and RZ/V2L Evaluation Board Kit, following serial port driver must be installed.

https://ftdichip.com/drivers/vcp-drivers/

Download the software "Virtual COM port (VCP) driver" from the windows version "setup executable" on the download page and extract it.

Run the exe file extracted to install the serial port driver.

5.4　Boot

This section will explain how to boot-up from the microSD card.

Board information such as switch configuration will not be explained.

If you use the previous version of the RZ/V2L Linux Package, rewrite the loader files to eMMC.

For more details about switch configuration and how to write the loader files, please refer to the SMARC EVK of RZ/V2L Linux Start-Up Guide (R01US0617).

1.　Setup the hardware according to 5.1 Hardware Configuration.

2.　Carry out the serial communication.
　　For Ubuntu PC, refer to the 4.3.6 Setup the Serial Communication.

　　For Windows PC, use the terminal emulator software.

　　Configuration for serial communication is as follows.

Configuration items	Value
Baud rate	115200
Data bit	8bit
Parity	none
Stop bit	1bit

3.　Turn on the RZ/V2L Evaluation Board Kit and change the U-boot environment variable.

　　Note: Only when switching the booting method from Network Booting.

　　(1)　　　After turned on the RZ/V2L Evaluation Board Kit, keep pressing the ENTER key.

　　(2)　　　The U-boot console will be activated.

　　(3)　　　Enter the following commands.

```
=> env default -a
=> setenv bootargs 'root=/dev/mmcblk1p2 rootwait'
=> setenv bootcmd 'mmc dev 1;fatload mmc 1:1 0x48080000 Image-smarc-rzv2l.bin;
fatload mmc 1:1 0x48000000 Image-r9a07g054l2-smarc.dtb; booti 0x48080000 –
0x48000000'
=> saveenv
=> boot
```

　　Details of above setting are as follow.

　　bootcmd: boot command

4.　After the boot-up, login screen will be shown on serial communication console.

　　Login information is as follows.

　　　user: "root"

　　　password: none

6. SDK

This chapter will explain how to construct the Compile Environment, SDK, which is necessary to compile a RZ/V2L Linux application.

The compiling procedures are explained in the RZ/V2L DRP-AI Sample Application Note.

This chapter assumes that procedures explained in the 3 Build are already executed and Build Environment is already prepared.

Note: In order to compile DRP-AI Sample Application for ISP version, both RZ/V2L DRP-AI Support Package and RZ/V2L ISP Support Package must be deployed to the SDK. Before reading this chapter, please make sure to deploy the RZ/V2L ISP Support Package to Build Environment as explained in RZ/V2L ISP Support Package Release Note (R11AN0561).

6.1   About the SDK

SDK (Software Development Kit) is a development environment that will allow users to cross-compile the C/C++ source code for RZ/V2L Linux (ARM64).

6.2   Build SDK

In the working directory, which is set in 3.2.2 Add the Environmental Variable, run the following command.

Note: Before executing following commands, apply all necessary recipes required for the Compile Environment.

I.e., To prepare the SDK for the RZ/V2L DRP-AI Sample Application for ISP version, apply the RZ/V2L DRP-AI Support Package recipe and RZ/V2L ISP Support Package recipe first and then execute the following commands.

```
$ cd $WORK
$ TEMPLATECONF=$PWD/meta-renesas/meta-rzv2l/docs/template/conf/ source \
poky/oe-init-build-env build
$ MACHINE=smarc-rzv2l bitbake core-image-weston -c populate_sdk
```

After the Build, following files will be generated under `$WORK/build/tmp/deploy/sdk`.

Table 6-1 Generated Files after Build

Filename	Name
poky-glibc-x86_64-core-image-weston-aarch64-smarc-rzv2l-toolchain-x.x.x.sh	SDK Installer Shell Script

Note: "x.x.x" is replaced by SDK version.

6.3 Install SDK

This section uses the file generated in 6.2 Build SDK.

1. On Ubuntu PC, run the following command.

```
$ cd $WORK/build/tmp/deploy/sdk/
$ sudo sh poky-glibc-x86_64-core-image-weston-aarch64-smarc-rzv2l-toolchain-x.x.x.sh
```

Note: "x.x.x" is replaced by SDK version.

In the Installer, specify the SDK installation directory and enter "Y".

Example below installs the SDK to default directory, "/opt/poky/x.x.x".

```
Poky (Yocto Project Reference Distro) SDK installer version x.x.x
================================================================
Enter target directory for SDK (default: /opt/poky/x.x.x):
You are about to install the SDK to "/opt/poky/x.x.x". Proceed[Y/n]? Y
Extracting SDK........................................................done
Setting it up...done
Setup /etc/ld.so.conf for the SDK with multilib environment
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/x.x.x/environment-setup-aarch64-poky-linux
$ . /opt/poky/x.x.x/environment-setup-armv7vet2hf-neon-vfpv4-pokymllib32-
linux-gnueabi
```

2. Confirm that following directories and files are generated under the specified directory.

Note: "/opt/poky/x.x.x" will be replaced with your SDK installed path.

```
📁 opt
    📁 poky
        📁 x.x.x
            📁 sysroots
            🗋 environment-setup-aarch64-poky-linux
            🗋 environment-setup-armv7vet2hf-neon-vfpv4-pokymllib32-linux-gnueabi
            🗋 site-config-aarch64-poky-linux
            🗋 site-config-armv7vet2hf-neon-vfpv4-pokymllib32-linux-gnueabi
            🗋 version-aarch64-poky-linux
            🗋 version-armv7vet2hf-neon-vfpv4-pokymllib32-linux-gnueabi
```

Figure 6-1 Directory Structure of SDK

The file "`environment-setup-aarch64-poky-linux`" will be used to change the environment variable to compile the RZ/V2L Linux application.

To compile the RZ/V2L Linux application, please refer to the RZ/V2L DRP-AI Sample Application Note.

Version History

Ver.	Date	Description		
		Page	Summary	
0.10	7 Jul, 2021	-	Issued.	
0.90	31 Aug, 2021	-	Supported the RZ/V2L Linux Package update to ver0.8.0.	
			Supported the DRP-AI Translator update to v1.52.	
			Added the RZ/V2L DRP-AI Driver.	
			Added the RZ/V2L DRP-AI Sample Application.	
			Added the RZ/V2L DRP-AI Accuracy Report.	
			Added the RZ/V2L DRP-AI Performance Report.	
		6	Added Chapter 2.2 "Build Environment"	
		6	Added Chapter 2.4 "Compile Environment"	
		6	Added Chapter 2.5 "Execution Environment"	
		7	Added Chapter 3 "Build"	
		10	Added Chapter 4 "Network Booting"	
		17	Added Chapter 5 "SD Card Booting"	
		21	Added Chapter 6 "SDK"	
0.91	30 Sep, 2021	5	Updated the RZ/V2L AI Implementation Guide to ver0.91.	
			Updated the Sample source codes for RZ/V2L AI Implementation Guide to ver0.91.	
			Updated the RZ/V2L AI Evaluation Software Guide to ver0.91.	
		5	Updated the following DRP-AI object files for rzv2l_ai-implementation-guide_ver0.91.tar.gz.	
			▪ resnet50_cam	
			▪ resnet50_jpg	
			▪ tinyyolov2_cam	
			▪ tinyyolov2_jpg	
			▪ yolov3_cam	
			▪ yolov3_jpg	
5.00	24 Dec, 2021	-	Supported the RZ/V2L Linux Package update to ver1.0.0.	
			Supported the DRP-AI Translator update to v1.60.	
			Supported the DRP-AI Driver update to v1.00.	
7.00	28 Apr, 2022	-	Supported the RZ/V2L Linux Package update to ver1.0.1.	
			Supported the DRP-AI Translator update to v1.70.	
			Supported the RZ/V2L ISP Support Package v1.10.	
		10	Added Chapter 3.2.7 "HDMI monitor resolution"	
7.10	29 Jul, 2022	-	Supported the RZ/V2L Linux Package update to ver3.0.0.	
			Supported the RZ/V2L ISP Support Package v1.20.	
		11	Deleted Chapter 3.2.5 "Apply the patch"	
		9, 10, 11	Updated DRP-AI Support Package Recipe	
7.20	31 Oct, 2022	-	Supported RZ/V2L Linux Package update to ver3.0.0-update2.	
			Supported the DRP-AI Translator update to v1.80.	
			Supported the DRP-AI Driver update to v1.20.	
7.30	31 Jan, 2023	-	Supported RZ/V2L Linux Package update to ver3.0.2.	
			Supported the DRP-AI Translator update to v1.81.	
			Supported the DRP-AI Driver update to v2.00.	
			Supported the RZ/V2L ISP Support Package v1.21.	
7.40	31 Jul, 2023	-	Supported the RZ/V Verified Linux Package update to ver3.0.4.	
			Supported the DRP-AI Translator update to v1.82.	
			Supported the DRP-AI Driver update to v2.10.	
			Supported the RZ/V2L ISP Support Package v1.30.	
7.41	31 Jan, 2024	-	Supported the RZ/V Verified Linux Package update to ver3.0.5.	

			Supported the DRP-AI Translator update to v1.83. Supported offline build.

RENESAS

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

    "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

    "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

    Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.