RZ/V2L, RZ/V2L, RZ/V2MA AI EVALUATION SOFTWARE GUIDE

REV.7.20
SEPTEMBER 2022

This document describes AI Evaluation Software for RZ/V2L, RZ/V2M, RZ/V2MA.
In this manual, RZ/V2L, RZ/V2M, and RZ/V2MA are referred to as RZ/V2x.

R11AN0624EJ0720

BIG IDEAS FOR EVERY SPACE RENESAS

CONTENTS

- Overview P3
- Setup the Network Booting Environment P11
- Serial Communication on Ubuntu PC P22
- Booting Instructions for RZ/V2L P24
- Booting Instructions for RZ/V2M, RZ/V2MA P28
- Execute the Application P32
- Appendix: SD Card Booting P43
  - SD Card Setup P48
  - Install the serial port driver P52
  - SD card Booting for RZ/V2L P53
  - SD card Booting for RZ/V2M, RZ/V2MA P56

OVERVIEW

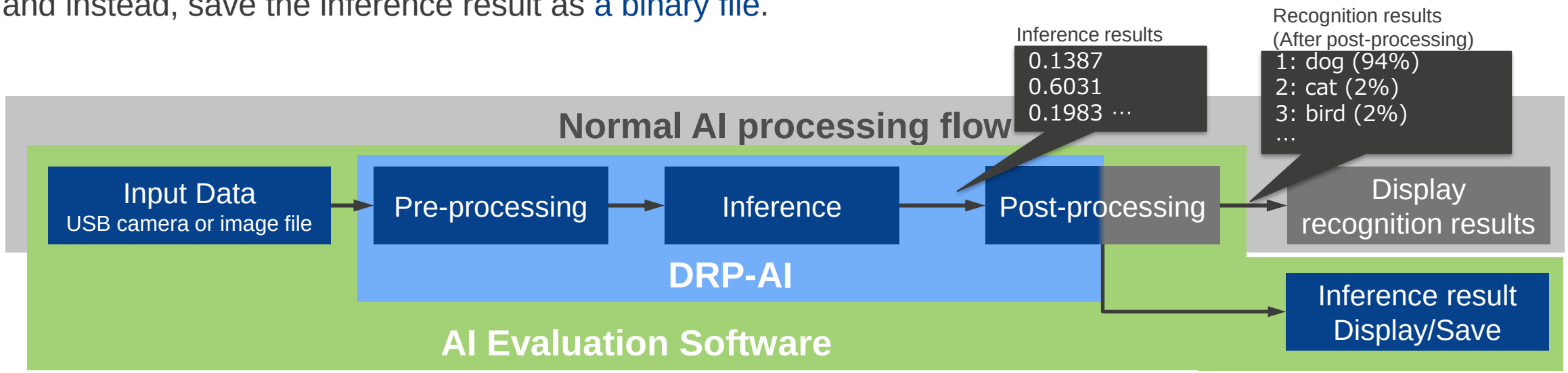
AI EVALUATION SOFTWARE

AI Evaluation Software is an application to evaluate image recognition AI models computed by DRP-AI.

It executes the AI model inference for the input data from USB camera or image files.

The normal AI inference execute the post-processing to obtain the recognition results.

Since this post-processing differs for each AI model, this application does not execute the post-processing and instead, save the inference result as a binary file.



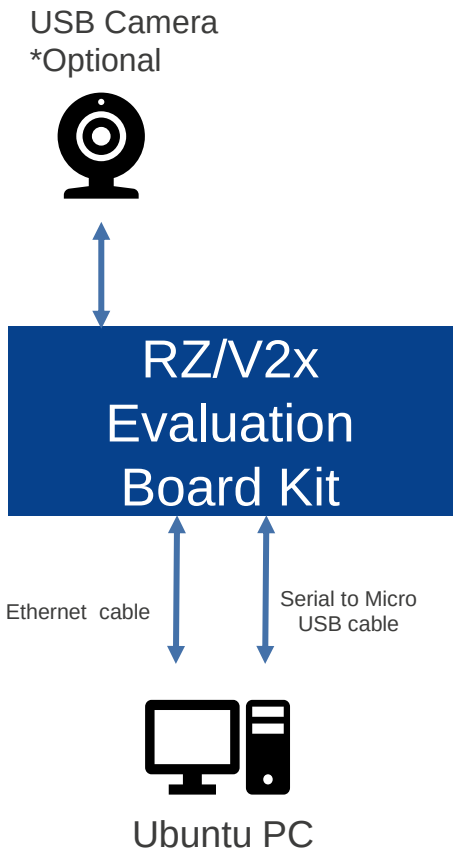
ABOUT THE BOOTING

This application can run on the following two ways of booting systems.
This document will explain the instructions for Network Booting. For the SD Card Booting, refer to the appendix.

Network Booting

This booting method will access the filesystem that is mounted on the Ubuntu PC server via ethernet.

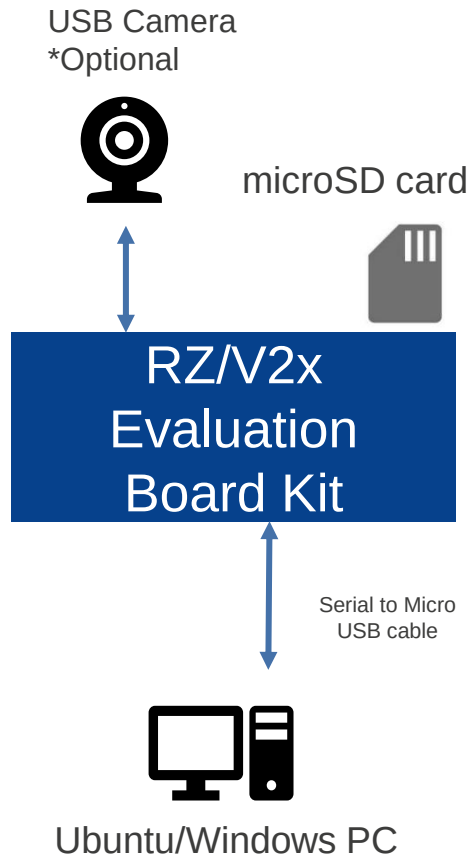
If you would like to use the large number of images, please use this booting method.



SD Card Booting

This booting method will use the filesystem on the microSD card.

If you would like to try the application as a demonstration, please use this booting method.



NECESSARY FILES

Please prepare the following files in order to carry out the instructions explained in this document.

Name	Filename		Details
AI Evaluation Software	RZ/V2L	rzv2l_ai-evaluation-software_ver7.20.tar.gz	AI Evaluation Software package itself
	RZ/V2M	rzv2m_ai-evaluation-software_ver7.20.tar.gz	
	RZ/V2MA	rzv2ma_ai-evaluation-software_ver7.20.tar.gz	
AI Implementation Guide	r11an0616ej0720-rzv-ai-imp-getstarted.pdf		A guide to implement AI models for DRP-AI
DRP-AI Translator	DRP-AI_Translator-v1.80-Linux-x86_64-Install		A tool to generate DRP-AI Object files. Refer to the AI Implementation Guide for its usage.

NECESSARY FILES

Details of AI Evaluation Software are as follow.
Files provided in this package is pre-built binary file of Linux Package and DRP-AI Support Package v7.20.

Name		Filename	Details
RZ/V2L Linux Package v3.0.0	Linux Kernel Image	Image-smarc-rzv2l.bin	Boot program
	Linux Device Tree File	Image-r9a07g054l2-smarc.dtb	Boot configuration file
	Root filesystem	core-image-weston-smarc-rzv2l.tar.bz2	Filesystem with AI Evaluation Software execution environment

Name		Filename	Details
RZ/V2M Linux Package v1.3.0 *1	Linux Kernel Image	Image-rzv2m.bin	Boot program
	Linux Device Tree File	r9a09g011gbg-evaluation-board.dtb	Boot configuration file
	Root filesystem	core-image-bsp-rzv2m.tar.bz2	Filesystem with AI Evaluation Software execution environment

Name		Filename	Details
RZ/V2MA Linux Package v1.0.0	Linux Kernel Image	Image-rzv2ma.bin	Boot program
	Linux Device Tree File	r9a09g055ma3gbg-evaluation-board.dtb	Boot configuration file
	Root filesystem	core-image-bsp-rzv2ma.tar.bz2	Filesystem with AI Evaluation Software execution environment

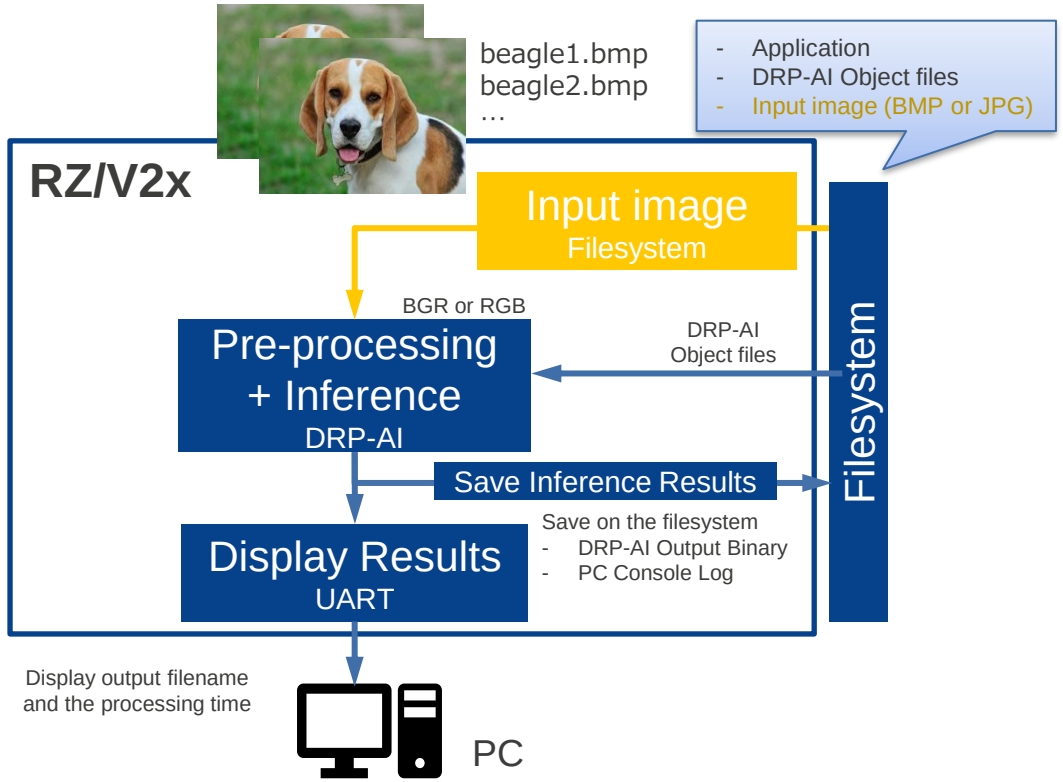
Note1 : Available in October, 2022.

ABOUT THE INPUT DATA TO DRP-AI

This application has two modes for input image data.
Both modes only support the VGA sized image data.

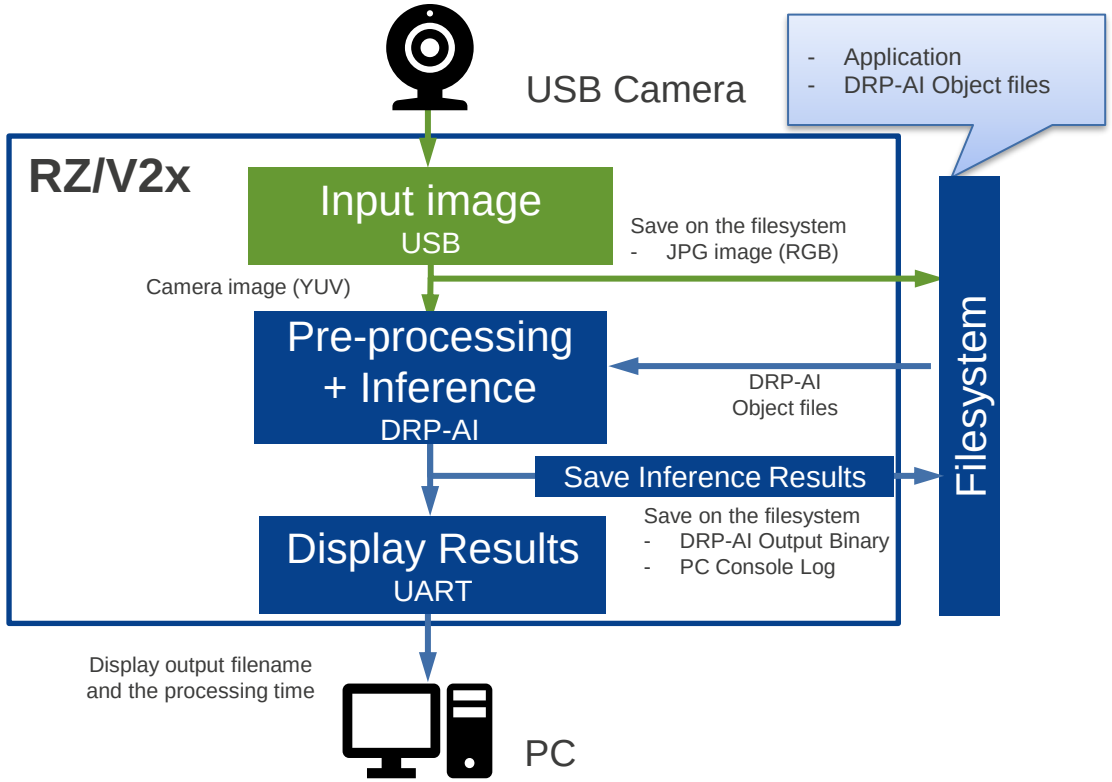
Image Mode

Uses the image file on the filesystem
Automatically executes the inference for images stored in the specified directory



Camera Mode

Uses the captured image from USB camera.
Users can determine when to execute the inference.



IN/OUT DATA

Following table shows the input/output images and display contents for each mode.

No.	Function	Image Mode	Camera Mode
1	Input Data	**BMP image** : Supports Windows Bitmap v3, v4, v5 (BGR) **JPG image** : Only supports ".jpg" files (RGB) Maximum number of input image : 20000	**USB camera captured image** : Captured by the application. (YUY2)
		Both modes only support VGA size (640x480)	
2	DRP-AI	Uses the DRP-AI Object files on the filesystem (Following 6 files, which are the output of DRP-AI Translator) 1. *_addr_intm.txt 2. *_weight.dat 3. *_drpcfg.mem 4. drp_param.bin 5. aimac_desc.bin 6. drp_desc.bin	
3	Displaying results	Display the followings on PC console by UART - Directory name of DRP-AI Object files - **Name of input image files** - DRP-AI processing time - Name of DRP-AI output binary files - Name of PC console log file	Display the followings on PC console by UART - Directory name of DRP-AI Object files - DRP-AI processing time - Name of DRP-AI output binary files - **Name of captured JPG image file from the USB camera** - Name of PC console log file
4	Output Data	<TIME>.log : Log file that contains the same contents as the No.3 <IMAGE_NAME>.bin : DRP-AI output binary file	<TIME>.log : Log file that contains the same contents as the No.3 output<N>_<TIME>.bin : DRP-AI output binary file **output<N>_<TIME>.jpg** : JPG image file captured from USB camera

COMMAND EXECUTION ENVIRONMENT

In this document, the instructions will be shown as Linux commands.

Commands are written in the following notation.

```
$ Command
```

Furthermore, this document will distinguish the command execution environment by the following notation.

1. Ubuntu PC

   ```
   $ printenv
   ```

2. Board Linux console

   Regardless of the serial communication device, following notation will be used.

   ```
   # printenv
   ```

3. Board U-boot console for setting the booting environment variable.

   Regardless of the serial communication device, following notation will be used.

   ```
   => printenv
   ```

SETUP THE NETWORK BOOTING ENVIRONMENT

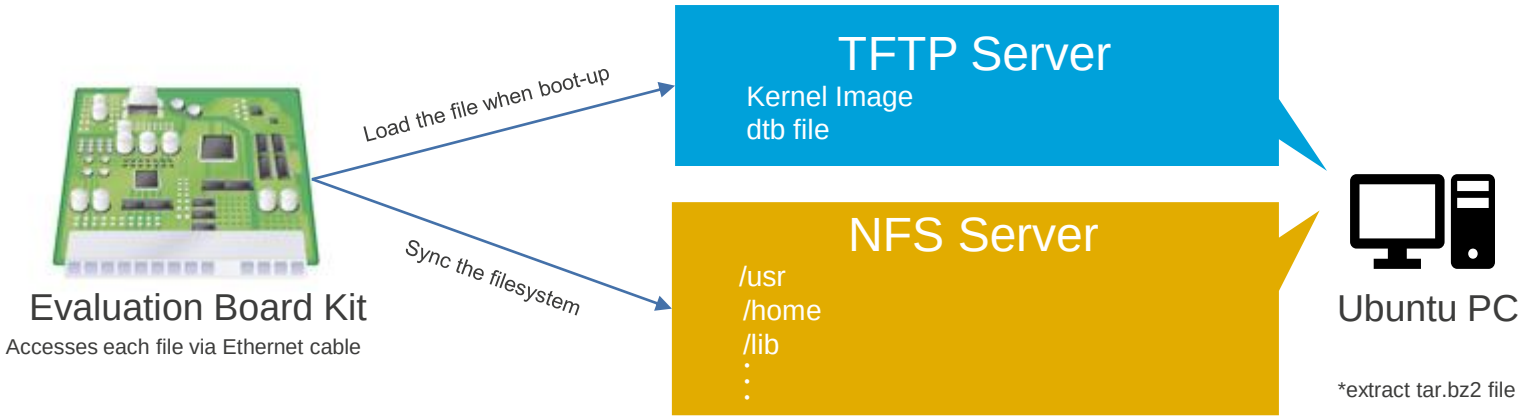
This chapter is for the Network Booting.

Regarding the SD Card Booting, please refer to the Appendix.

WHAT IS NETWORK BOOTING?

- Network Booting mount the files used on the board onto the Ubuntu PC server.
- Following three files are necessary for the booting. These files are placed on each mounted server.

File	Description	Mounted Server
Kernel Image	Linux kernel image (The boot program)	TFTP server on Ubuntu PC
dtb file	Linux device tree file (The configuration file for booting)	TFTP server on Ubuntu PC
tar.bz2 file	Root filesystem	NFS server on Ubuntu PC



- This chapter will explain how to setup each server and how to allocate each file.

NECESSARY EQUIPMENT FOR RZ/V2L

Necessary equipment for the Network Booting and its operating environment is as follows.

Equipment	Details	Where to obtain
RZ/V2L Evaluation Board Kit	Linux bootloader is pre-written to eMMC. Note: For writing bootloader, refer to RZ/V2L Linux Package Start-up Guide.	Provided by Renesas
Serial to Micro USB Cable	Used for serial communication between Linux PC and the board.	
AC adapter	Power supply	This equipment should be prepared by users.
USB Cable Type-C	Connect AC adapter and the board.	
Ethernet cable	Used for ethernet communication between Linux PC and the board.	
Linux PC	Ubuntu 20.04 is pre-installed. Operating Environment: Intel Core i5 11th generation Memory 16GB	
USB camera	Operating Environment: Logitech C930E WEBCAM	

NECESSARY EQUIPMENT FOR RZ/V2M, RZ/V2MA

Necessary equipment for the Network Booting and its operating environment is as follows.

Equipment	Details	Where to obtain
RZ/V2x Evaluation Board Kit	Linux bootloader is pre-written to eMMC Note: For writing bootloader, refer to RZ/V2x Linux Package Start-up Guide.	Provided by Renesas
Serial to Micro USB cable	Used for serial communication between Linux PC and the board.	This equipment should be prepared by users.
AC adapter	Power supply	
Ethernet cable	Used for ethernet communication between Linux PC and the board.	
Linux PC	Ubuntu 20.04 is pre-installed. Operating Environment: Intel Core i5 11th generation Memory 16GB	
USB camera	Operating Environment: Logicool C930e Web Cam *Optional	
Type-C to USB-A Adaptor	Necessary since USB port on RZ/V2x Evaluation Board Kit is Type-C. *Optional Operating Environment: SANWA SUPPLY AD-USB28CAF	

NECESSARY SOFTWARE

- The instructions explained in this chapter use the following files. To obtain the files, refer to the Necessary Files
  - Kernel Image
  - dtb file
  - tar.bz2 file

- This document uses vim as a text editor.

```
$ vi hello.txt
```

- Please pre-install the following necessary software packages on Ubuntu PC.

```
$ 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
$ sudo apt-get install python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 xterm
$ sudo apt-get install python3-subunit mesa-common-dev p7zip-full
```

- In the Network Booting, following additional package is required. Please install them by executing following command.

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

- Please install the following package for the serial communication on Ubuntu PC.

```
$ sudo apt-get install cu
```

BUILD TFTP SERVER

Please execute the following instructions on Ubuntu PC.

This chapter uses following software. Please install them according to the Necessary Software.

- tftp
- tftpd-hpa

1. Create the directory for the TFTP server.

```
$ sudo mkdir /tftpboot
```

Note: TFTP server directory name will be used later. Please use the above name.

2. Change the TFTP server configuration.

Open /etc/default/tftpd-hpa file and write the configuration shown on the right.

```
$ sudo vi /etc/default/tftpd-hpa
```

3. Start the TFTP server.

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

2. TFTP Server Configuration BEFORE

```
# /etc/default/tftpd-hpa

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

2. TFTP Server Configuration AFTER

```
# /etc/default/tftpd-hpa

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

BUILD NFS SERVER

Please execute the following instructions on Ubuntu PC.

This chapter uses following software. Please install them according to the Necessary Software

- nfs-common
- nfs-kernel-server

1. Create the directory for the NFS server. Note: The server directory name will be used later. Please use the above name.

   For RZ/V2L:

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

   For RZ/V2M, RZ/V2MA:

   ```
   $ sudo mkdir -p /nfs/rzv2m
   ```

2. Start the NFS server.

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

3. Change the NFS server configuration.
   Open /etc/exports file and add the following line at the end of the file.

   ```
   $ sudo vi /etc/exports
   ```

   For RZ/V2L:

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

   For RZ/V2M, RZ/V2MA:

   ```
   …
   /nfs/rzv2m *(rw,no_subtree_check,sync,no_root_squash)
   ```

4. Refresh the NFS server.

   ```
   $ sudo exportfs -a
   ```

CONFIRM THE SERVER

Please execute the following instructions on Ubuntu PC.

1. Confirm the TFTP server is successfully built by executing following commands.

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

-> If the command returns without any errors, the TFTP server is built successfully.

2. Confirm the NFS server is successfully built by executing following commands.

For RZ/V2L:

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

For RZ/V2M, RZ/V2MA:

```
$ showmount -e localhost
Export list for localhost:
/nfs/rzv2m *
```

-> If it shows the same contents as above, the NFS server is built successfully.

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

SETUP THE STATIC IP ADDRESS

Please execute the following instructions on Ubuntu PC.

In order to enable ethernet communication between Ubuntu PC and the board, static IP address must be set on Ubuntu PC.

1. Disable the default network setting. (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 99-netcfg.yaml file.

```
$ sudo vi /etc/netplan/99-netcfg.yaml
```

3. Write the contents shown on the right.

   (enp0s3 may be different depending on the environment.)

4. Restart the network.

```
$ sudo netplan apply
```

3. 99-netfcfg.yaml

```
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
```

※ Since this IP address will be used later, please use the address shown here.
Also, please write the whitespace and indent exactly as shown above.

ALLOCATE NECESSARY FILES

Please execute the following instructions on Ubuntu PC.

1. Copy the following two files to the TFTP server by executing the commands below.

Board	file	file name
RZ/V2L	Kernel Image	Image-smarc-rzv2l.bin
	dtb file	Image-r9a07g054l2-smarc.dtb
RZ/V2M	Kernel Image	Image-rzv2m.bin
	dtb file	r9a09g011gbg-evaluation-board.dtb
RZ/V2MA	Kernel Image	Image-rzv2ma.bin
	dtb file	r9a09g055ma3gbg-evaluation-board.dtb

```
$ sudo cp <PATH_to_FILE>/<Kernel Image> /tftpboot
$ sudo cp <PATH_to_FILE>/<dtb file> /tftpboot
```

Note: 1. <PATH_to_FILE> is a path to each file.
2. Change <Kernel Image> and <dtb file> by referring to the above table.

ALLOCATE NECESSARY FILES

2. Extract the following files to the NFS server by executing the commands below.

For RZ/V2L:

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

For RZ/V2M:

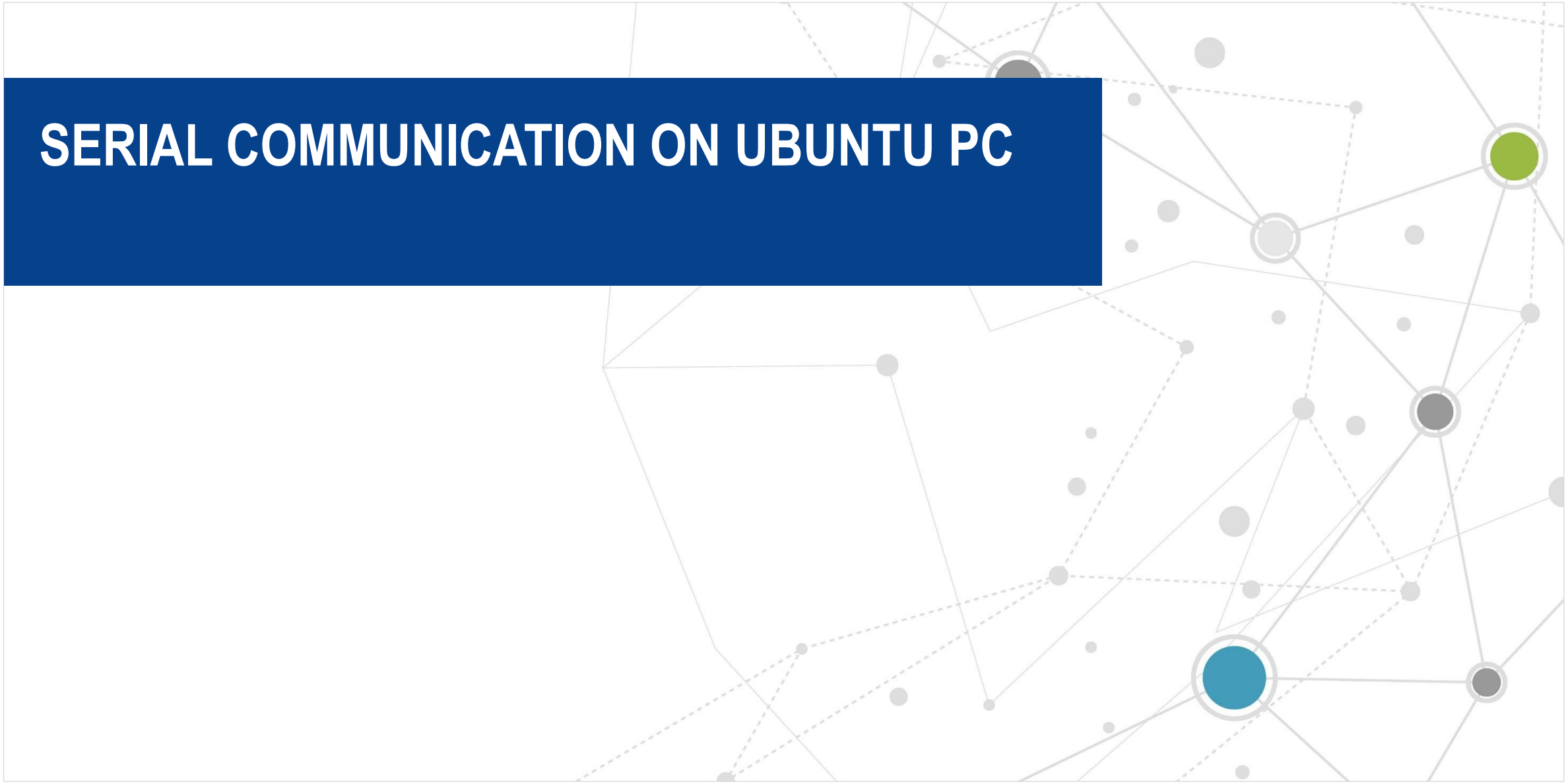
```
$ sudo tar xfj <PATH_to_FILE>/core-image-bsp-rzv2m.tar.bz2 -C /nfs/rzv2m
```

For RZ/V2MA:

```
$ sudo tar xfj <PATH_to_FILE>/core-image-bsp-rzv2ma.tar.bz2 -C /nfs/rzv2m
```

Note: <PATH_to_FILE> is a path to each file.

SERIAL COMMUNICATION ON UBUNTU PC



SERIAL COMMUNICATION UBUNTU + BOARD

Please execute the following instructions on Ubuntu PC.

Network Booting assumes that serial communication will be done between Ubuntu PC and the board.

- This chapter uses following software. Please install them according to the Necessary Software
  - cu
- Followings are the instruction to carry out the serial communication.

1. Connect the Ubuntu PC and the board by the Serial to Micro USB cable.
2. Obtain the name of serial port. RZ/V2M has 2 ports, please use ttyUSB1.

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

3. Change the permission of the serial port. Please use ttyUSB1 for RZ/V2M.

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

4. Execute the following command to start the serial communication. Please use ttyUSB1 for RZ/V2M.

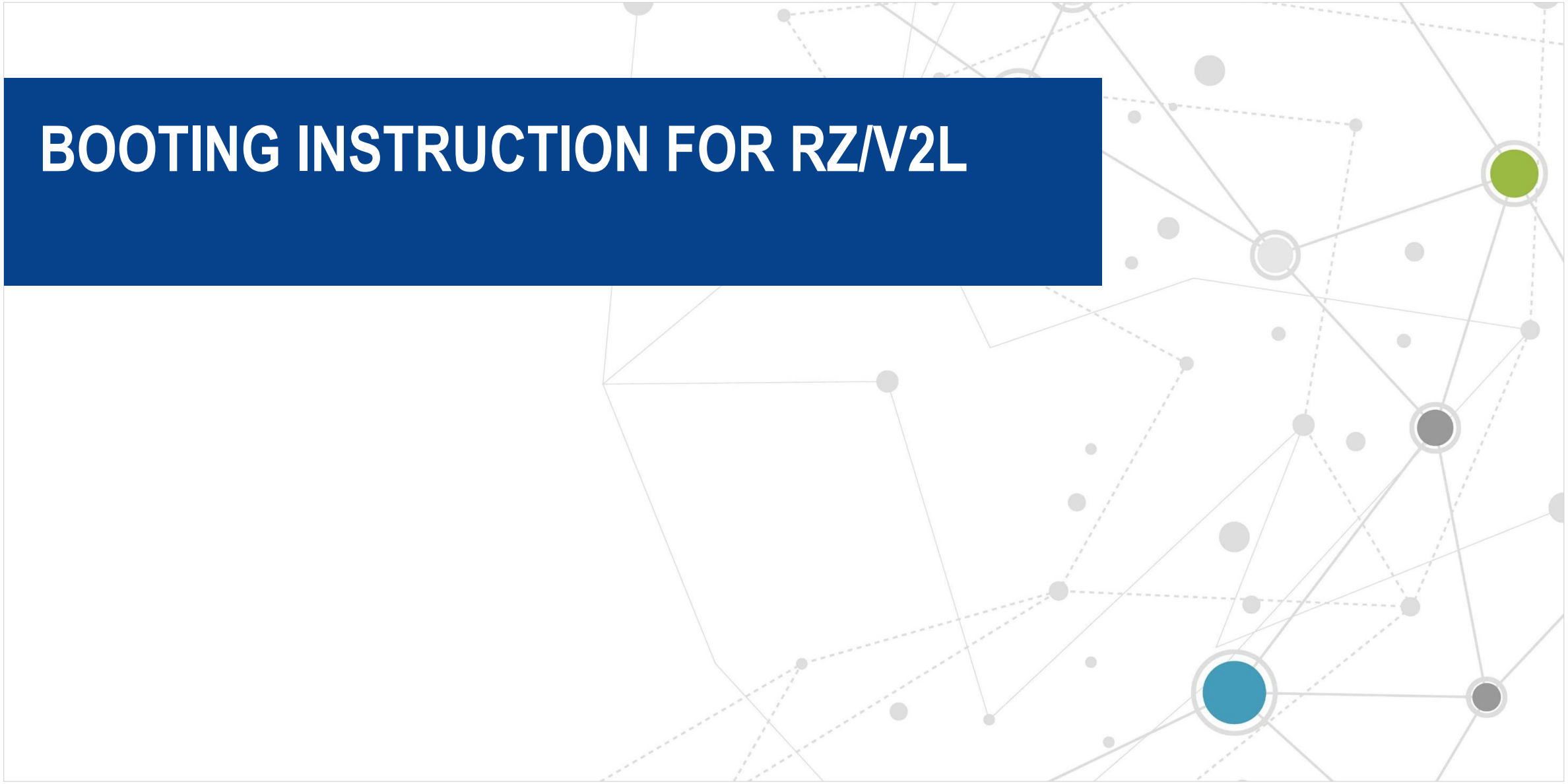
```
$ cu -s 115200 -l /dev/ttyUSB0 --parity none --nostop
```

* To end the communication, enter “~.”

5. Open another console on the Ubuntu PC and change "crtscts" option. Please use ttyUSB1 for RZ/V2M.

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

BOOTING INSTRUCTION FOR RZ/V2L



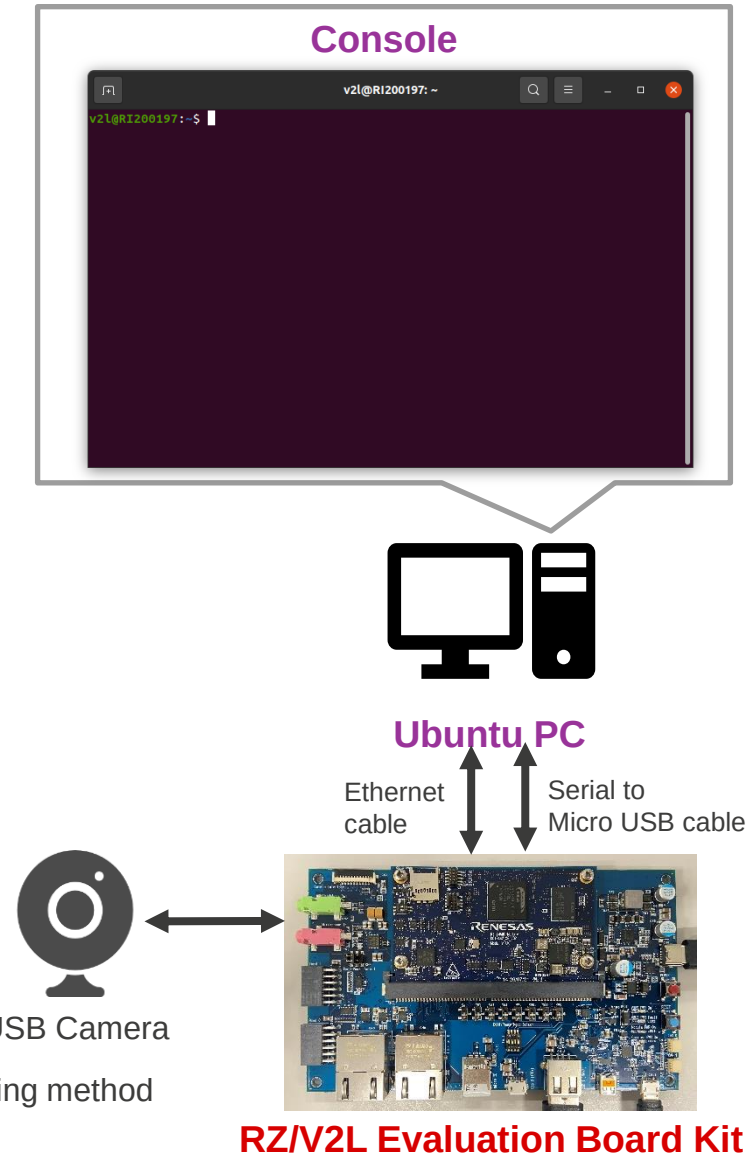
BOOTING OVERVIEW (Network Booting)

- Instruction overview (Details will be explained in the next page)

This instruction assumes that following setup are already done.

  - Setup the Network Booting Environment
  - Serial Communication on Ubuntu PC

1. Connect the **Board** and **Ubuntu PC** by the Serial to Micro USB cable.
2. Connect the **Board** and **Ubuntu PC** by the Ethernet cable.
3. Connect the power cable to the **Board**
4. Connect the **USB camera** to the **Board** *Camera Mode Only
5. Turn on the **Board**
6. Open the **console** on **Ubuntu PC** and execute the serial communication.
7. Change the U-boot environment variable on the **console**. *Only when changing the booting method
8. If booted successfully, the login screen will be displayed on the **console**



BOOTING INSTRUCTIONS DETAILS

■ Assumption: Executed Preparation

1. Connect the **Board** and **Ubuntu PC** by the Serial to Micro USB cable.

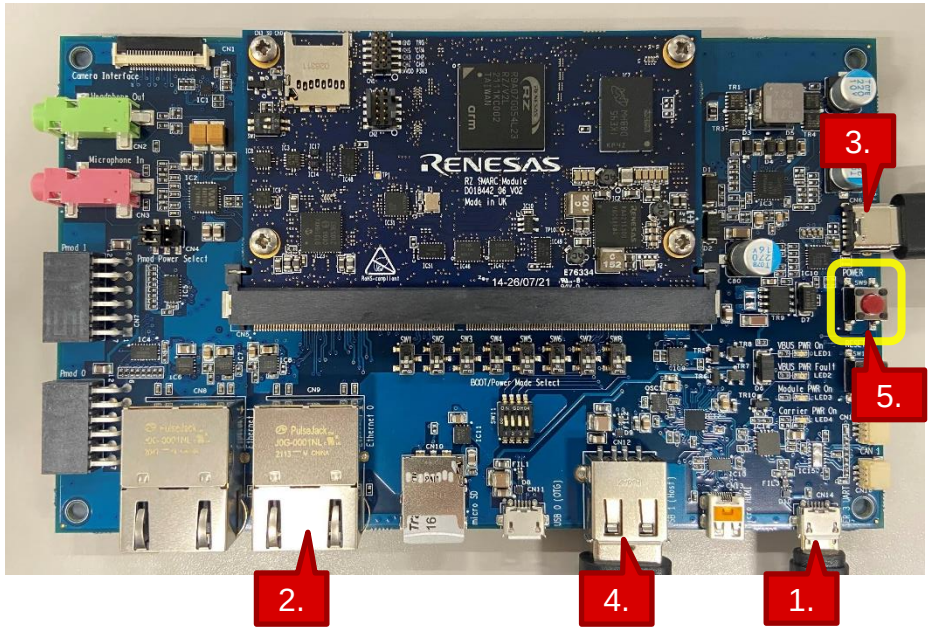
2. Connect the **Board** and **Ubuntu PC** by the Ethernet cable.

3. Connect the power cable to the **Board**

4. Connect the **USB camera** to the **Board** *Camera Mode only

5. Turn on the **Board**

6. Open the **console** on **Ubuntu PC** and execute the serial communication.
   Refer to the Serial Communication



BOOTING INSTRUCTIONS DETAILS

7. Change the U-boot environment variable on the **console**. * Only when changing the booting method

   1. After turned on the board, keep pressing ENTER key on the **console**
   2. U-boot console will be activated.
   3. Enter the following commands

```
=> 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
```

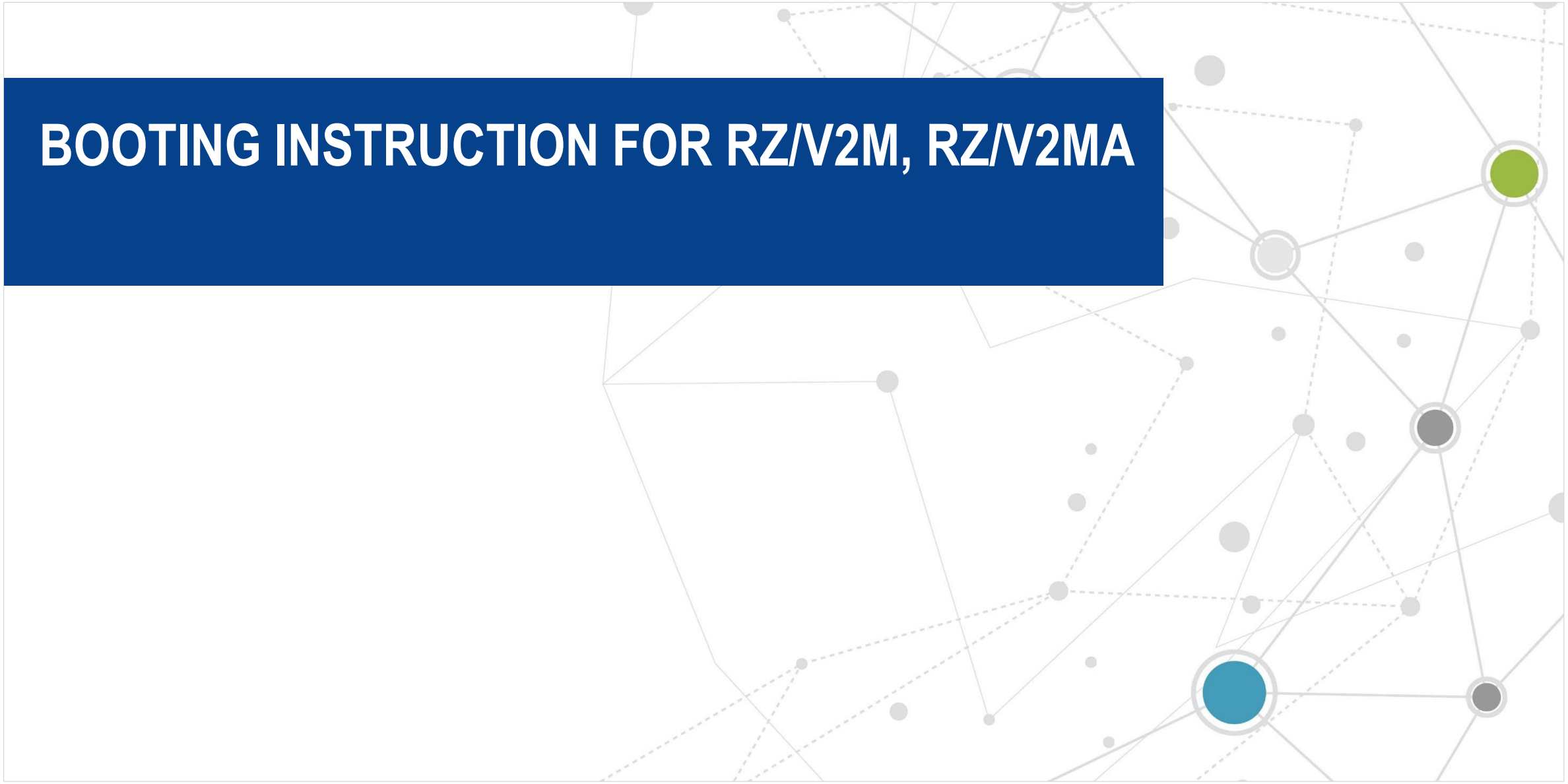
※serverip = IP address of Ubuntu PC
ipaddr = IP address of the board
The Ubuntu PC address is the address stated in Setup the Static IP Address

8. After the boot-up, the login screen will be shown on the **console**.
   Log-in to the system using the following information.

   – user: "root"

   – password: none

BOOTING INSTRUCTION FOR RZ/V2M, RZ/V2MA



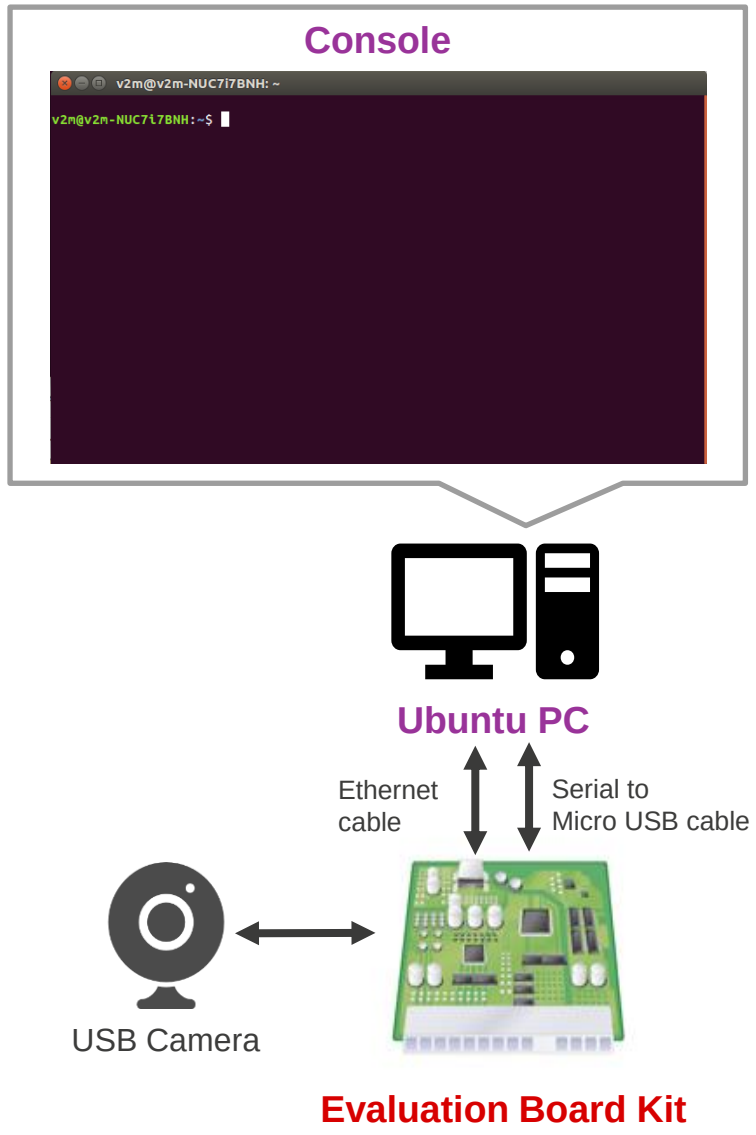
BOOTING OVERVIEW (Network Booting)

- Instruction overview (Details will be explained in the next page)

This instruction assumes that following setup are already done.

  - Setup the Network Booting Environment
  - Serial Communication on Ubuntu PC

1. Connect the **Board** and **Ubuntu PC** by the Serial to Micro USB cable.
2. Connect the **Board** and **Ubuntu PC** by the Ethernet cable.
3. Connect the power cable to the **Board**.
4. Connect the **USB Camera** to the **Board**. *Only when using the Camera mode
5. Open the **console** on **Ubuntu PC** and execute the serial communication.
6. Turn on the **Board**
7. Change the U-boot environment variable on the **console**.
   *Only when changing the booting method
8. If booted successfully, the login screen will be displayed on the **console**



BOOTING INSTRUCTIONS DETAILS

■ Assumption: Executed Preparation

1. Connect the **Board** and **Ubuntu PC** by the Serial to Micro USB cable.

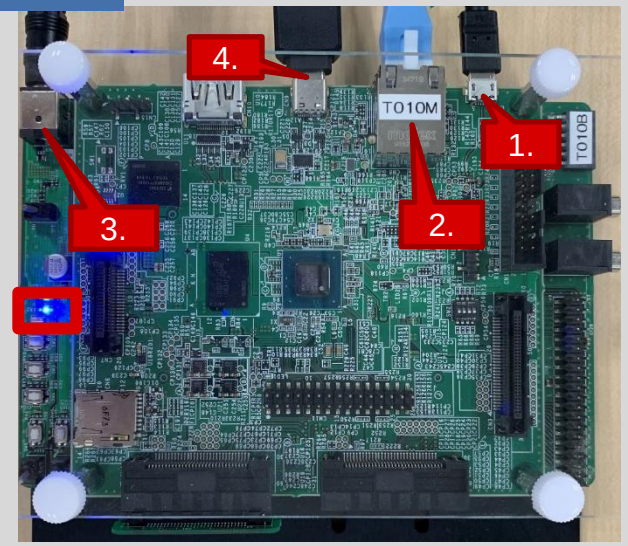
2. Connect the **Board** and **Ubuntu PC** by the Ethernet cable.

3. Connect the power cable to the **Board**. *The LED lights up

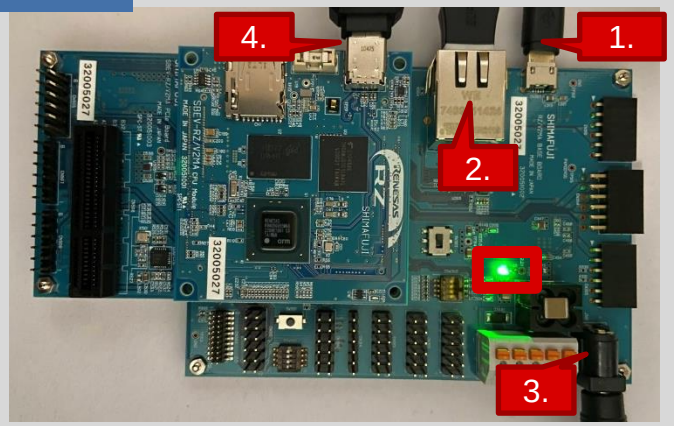
4. Connect the **USB Camera** to the **Board**. *Only when using the Camera mode

5. Open the **console** on **Ubuntu PC** and execute the serial communication.
   Refer to the Serial Communication

RZ/V2M



RZ/V2MA



BOOTING INSTRUCTIONS DETAILS

6. Turn on the **Board**. *Another two LEDs light up

7. Change the U-boot environment variable on the **console**.
   * Only when changing the booting method

   1. After turned on the board, keep pressing ENTER key on the **console**
   2. U-boot console will be activated.
   3. Enter the following commands

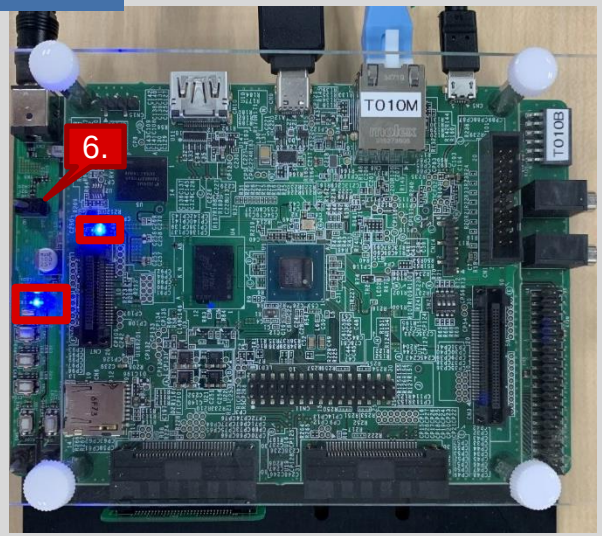
```
=> env default -a
=> setenv ipaddr 192.168.1.11
=> setenv serverip 192.168.1.10
=> setenv bootcmd run bootnfs
=> saveenv
=> boot
```

※serverip = IP address of Ubuntu PC
ipaddr = IP address of the board
The Ubuntu PC address is the address stated in Setup the Static IP Address

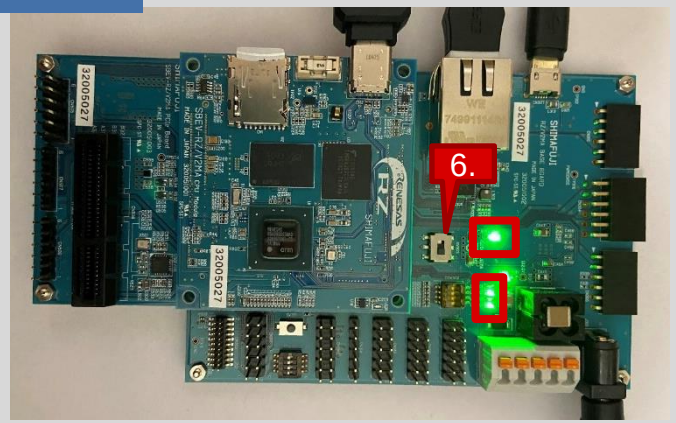
8. After the boot-up, the login screen will be shown on the **console**.
   Log-in to the system using the following information.

   – user: "root"
   – password: none

RZ/V2M



RZ/V2MA



EXECUTE THE APPLICATION

EXECUTION ENVIRONMENT

Following is the directory structure of execution environment.

*Necessary files for the application.
Files generated by the application.

home	
root	
RZV_AI_Eva_SW	
resnet50_bmp	Sample DRP-AI Object files
resnet50_bmp_weight.dat	DRP-AI Translator output of PyTorch Resnet-50 model for the Image Mode (BMP image)
resnet50_bmp_drpcfg.mem	Note: Among the DRP-AI Translator output, only necessary files are listed here.
aimac_desc.bin	
drp_desc.bin	
drp_param.bin	
resnet50_bmp_addrmap_intm.txt	
resnet50_jpg	Sample DRP-AI Object files
resnet50_jpg_weight.dat	DRP-AI Translator output of PyTorch Resnet-50 model for the Image Mode (JPG image)
・・・	
resnet50_cam	Sample DRP-AI Object files
resnet50_cam_weight.dat	DRP-AI Translator output of PyTorch Resnet-50 model for the Camera Mode
・・・	
bmp_img	Sample input image
sample.bmp	BMP image
jpg_img	Sample input image
sample.jpg	JPG image
<prefix>_output	Output of AI Evaluation Software. <prefix> is the name of DRP-AI Object files directory used. i.e., resnet50_bmp
<img_dir>	Output directory for the Image Mode. <img_dir> is the directory name specified as input data. i.e., bmp_img
<img>.bin	DRP-AI output binary. <img> is the name of input file. i.e., sample.bmp
<TIME>.log	AI Evaluation Software console log.
capture	Output directory for the Camera Mode.
output<N>_<TIME>.bin	DRP-AI output binary
output<N>_<TIME>.jpg	Camera capture image file (JPG)
<TIME>.log	AI Evaluation Software console log.
rzv_ai_eva_sw	AI Evaluation Software
start_app.sh	A script to execute AI Evaluation Software
config.ini	Configuration file for AI Evaluation Software

START THE APPLICATION

- Execute the following commands to start the application.

  Move to the application directory.

```
# cd ~/RZV_AI_Eva_SW
```

  Execute the application

```
# ./start_app.sh <M> -d <DRP-AI_EXE> [-i <IMG_DIR>]
```

  – Details of the command line arguments are as follow.

Mode	<M>	<DRP-AI_EXE>	<IMG_DIR>
Image Mode	I	Directory name for the Image Mode DRP-AI Object files Default: resnet50_bmp	Directory name that contains the input images. Default: bmp_img
Camera Mode	C	Directory name for the Camera Mode DRP-AI Object files Default: resnet50_cam	N/A (Uses camera image)

Loading screen

```
# ./start_app.sh I
IMAGE_MODE
…
[INFO] DRP-AI Execution Binary: resnet50_bmp
[START] Loading DRP-AI Data…
  [START] Loading resnet50_bmp/resnet50_bmp_weight.dat : size …
  [END]   Loading resnet50_bmp/resnet50_bmp_weight.dat
  [START] Loading resnet50_bmp/resnet50_bmp_drpcfg.mem : size …
  [END]   Loading resnet50_bmp/resnet50_bmp_drpcfg.mem
  [START] Loading resnet50_bmp/drp_param.bin : size …
  [END]   Loading resnet50_bmp/drp_param.bin
  [START] Loading resnet50_bmp/aimac_desc.bin: size …
  [END]   Loading resnet50_bmp/aimac_desc.bin
  [START] Loading resnet50_bmp/drp_desc.bin : size …
  [END]   Loading resnet50_bmp/drp_desc.bin
[END] Loading DRP-AI Data : Total loading time  1.13 s
…
```

- After the application started, it loads the DRP-AI Object files.
  (Loading time depends on the model. E.g., PyTorch ResNet-50 takes approximately 1 seconds.)
- After loaded the DRP-AI Object files, the inference will be available.

INFERENCE IN IMAGE MODE

- Image Mode starts the inference automatically after it loaded the DRP-AI Object files.
- The sample console log is shown on the right.

```
# ./start_app.sh I
IMAGE MODE
[INFO] Image Directory : bmp_img
[INFO] DRP-AI Execution Binary: resnet50_bmp
[START] Loading DRP-AI Data…
…
[END] Loading DRP-AI Data : Total loading time  1.13 s
[bmp_img/sample.bmp]
1 images are loaded from img

Inference 1 ----------------------------------------------------
Input: bmp_img/sample.bmp
  DRP-AI processing time : 35.61 msec
  Output Binary          : resnet50_bmp_output/bmp_img/sample.bmp.bin
[INFO] 1 out of 1 images are processed.
[INFO] Output Log: resnet50_bmp_output/bmp_img/0722011508.log
```

- About the input image
  - The application only loads BMP images and JPG images stored in the specified input directory.
  - If the loading fails, the image will be skipped and the next image will be loaded.
  - All loaded images will be used for the inference.
  - The maximum number of input images allows is 20000.
- The end of the application
  - After the inference of all loaded image, the application automatically terminates.
  - After the termination, files listed on the right can be checked on the filesystem.

Generated files

These files will be saved in <DRP-AI_EXE>_output/<IMG_DIR>

Note that if there was a file with the same name, it would be overwritten.

- DRP-AI output binary file : <IMG_NAME>.bin
  Generated per inference
- Console log file: <TIME>.log
  Generated per application execution

INFERENCE IN CAMERA MODE

- Camera Mode waits the user key input after it loaded the DRP-AI Object files.
- The sample console log is shown on the right.

```
# ./start_app.sh C
CAMERA MODE
[INFO] DRP-AI Execution Binary: resnet50_cam
[START] Loading DRP-AI Data…
…
[END] Loading DRP-AI Data : Total loading time  0.93 s

Press ENTER to capture an image or q to quit:

Inference 1 ------------------------------------------------------
DRP-AI processing time : 36.19 msec
  Output Binary          : resnet50_cam_output/capture/output1_0722011507.bin
  Output Image           : resnet50_cam_output/capture/output1_0722011507.jpg
Press ENTER to capture an image or q to quit:
[INFO] Output Log: resnet50_bmp_output/capture/0722011508.log
```

Pressed ENTER

Pressed q

- About the user key input
  - If ENTER is pressed, inference will be executed.
  - If q is pressed, the application will terminate.
  - If other key is pressed, it will cause an error and the application will wait the user key input again.
- The end of the application
  - After the termination, files listed on the right can be checked on the filesystem.

Generated files

These files will be saved in <DRP-AI_EXE>_output/capture.

Note that if there was a file with the same name, it would be overwritten.

- DRP-AI output binary file : output<N>_<TIME>.bin
  Generated per inference
- JPG camera capture image: output<N>_<TIME>.jpg
  Generated per inference
- Console log file: <TIME>.log
  Generated per application execution

INFERENCE RESULT

- This application generates the binary file (*.bin) to output the inference result of DRP-AI.
- Details of the output binary file from AI Evaluation Software are as follow.
  - Number of data: Depends on the model output size.
    e.g., [1 x 1000] = 1000 for PyTorch ResNet
  - Data width: 4 bytes if FP32
    2 bytes if FP16
  - Byte order: Little endian

- In order to get the recognition result from the output data, CPU post-processing must be applied.
- The AI Implementation Guide explains the example of post-processing for PyTorch ResNet, PyTorch MobileNet, Darknet YOLO, MMPose HRNet and PyTorch DeepLabV3 by Python3.

ABOUT THE DRP-AI OBJECT FILES

- In order to execute inference with DRP-AI, DRP-AI Object files converted from the ONNX AI model are necessary.
- To create DRP-AI Object files, following three types of input files are required.
  Image Mode and Camera Mode requires different configuration in file b.

  a. *.onnx : ONNX model to be converted

  b. prepost_*.yaml : Configuration file of the pre/post-processing. This file will be different depending on the mode.

  c. addrmap_in_*.yaml : Configuration file of the address map.

- AI Implementation Guide explains how to modify the file b using ResNet-50 provided by PyTorch as an example.
  - All above files required to generate the DRP-AI Object files for PyTorch ResNet-50 are provided by the AI Implementation Guide.
  - Please see the *drpai_samples* directory in the pytorch_resnet_ver7.20.tar.gz.
- After prepared the input files, execute the DRP-AI Translator according to AI Implementation Guide to create DRP-AI Object files.

APPLICATION ERROR

- When an error occurred, this application displays its error code as shown on the right.

```
[ERROR] <ERROR_CAUSE>
[ERROR] Err Code <ERROR_CODE> : <ERROR_STRING>
```

- The details of error codes are as follow. Note: Refer to the RZ/V2MA AI Implementation Guide for more details of DRP-AI Driver API

Error Code	Error Name	Type	Details
0x01	ERR_INVALID_ARG	APP	Command line arguments are invalid. If this error occurred, check the Command line arguments.
0x02	ERR_IMG_LIST_OVERFLOW	APP	For Image Mode only. Overflow error for the number of input image data. If this error occurred, check the number of input images. The maximum number of input images is 20000.
0x10	ERR_OPEN	FILE I/O	File/Directory opening error. If this error occurred, check the file to be opened.
0x11	ERR_FORMAT	FILE I/O	File format error. If this error occurred, check the file format is supported by the application.
0x12	ERR_READ	FILE I/O	File reading error. if this error occurred, check the file is readable or not.
0x13	ERR_MMAP	FILE I/O	MMAP allocation error.
0x14	ERR_MALLOC	FILE I/O	MALLOC error
0x15	ERR_CLOSE	FILE I/O	File/Directory closing error. If this error occurred, check the file to be closed.
0x20	ERR_DRPAI_TIMEOUT	DRP-AI	Time out error for DRP-AI end interruption. See DRP-AI Error.
0x21	ERR_DRPAI_START	DRP-AI	System call error of DRP-AI Driver API, ioctl(DRPAI_START).
0x22	ERR_DRPAI_ASSIGN	DRP-AI	System call error of DRP-AI Driver API, ioctl(DRPAI_ASSIGN). See DRP-AI Object files Address Error.
0x23	ERR_DRPAI_WRITE	DRP-AI	Error of DRP-AI Driver API, write().
0x24	ERR_DRPAI_READ	DRP-AI	Error of DRP-AI Driver API, read().
0x25	ERR_DRPAI_SELECT	DRP-AI	System call error of DRP-AI Driver API, select().
0x26	ERR_DRPAI_STATUS	DRP-AI	System call error of DRP-AI Driver API, ioctl(DRPAI_GET_STATUS)
0x30	ERR_V4L2_OPEN	V4L2	For Camera Mode only. Failed to open V4L2 device.
0x31	ERR_V4L2_QUERYCAP	V4L2	For Camera Mode only. Failed to query V4L2 device capability.
0x32	ERR_V4L2_S_FMT	V4L2	For Camera Mode only. Failed to set V4L2 configuration.
0x33	ERR_V4L2_REQBUFS	V4L2	For Camera Mode only. Failed to request an image buffer of V4L2
0x34	ERR_V4L2_QUERYBUF	V4L2	For Camera Mode only. Failed to receive information of the image buffer requested by V4L2_REQBUFS.
0x35	ERR_V4L2_STREAMON	V4L2	For Camera Mode only. Failed to start streaming the V4L2 device.
0x36	ERR_V4L2_QBUF	V4L2	For Camera Mode only. Failed to set the V4L2 buffer to the streaming queue.
0x37	ERR_V4L2_DQBUF	V4L2	For Camera Mode only. Failed to unset the V4L2 buffer from the streaming queue.
0x38	ERR_V4L2_STREAMOFF	V4L2	For Camera Mode only. Failed to terminate streaming the V4L2 device.

DRP-AI OBJECT FILES ADDRESS ERROR

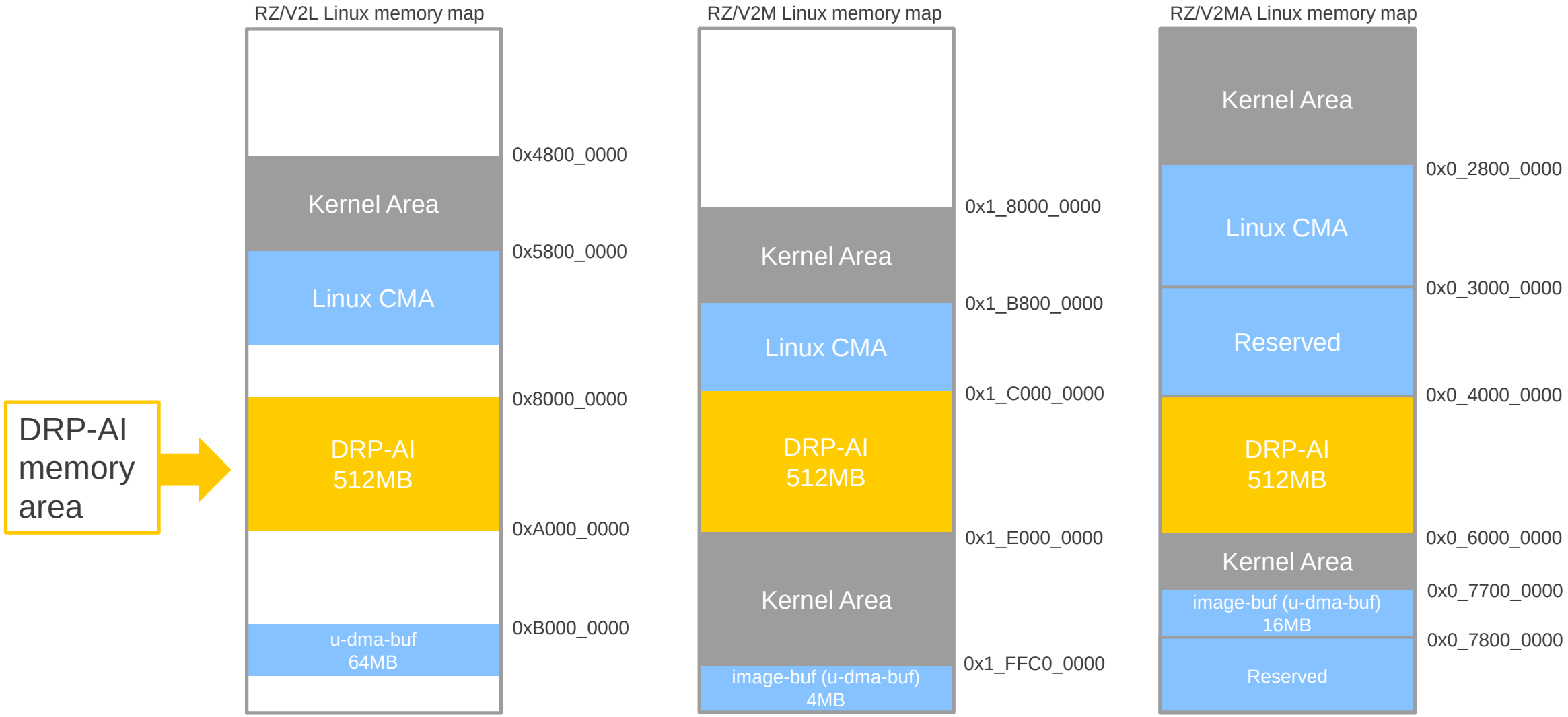
- Because of DRP-AI specification, each DRP-AI Object files must be allocated to the address.
  which satisfy the following restriction.

  1. Start address must be aligned to 64-byte boundary.
  2. Address and size listed in *_addr_intm.txt must be in the DRP-AI memory area.

     See the next page for the DRP-AI memory area.

- If the above restriction is not satisfied, DRP-AI Driver API error ERR_DRPAI_ASSIGN will occur.
  Note: DRP-AI Translator will generate the address, which satisfies the restriction 1, to *_addr_intm.txt.

DRP-AI OBJECT FILES ADDRESS ERROR



DRP-AI ERROR

- DRP-AI will notify the application when the processing is finished.
- However, there may be a case that DRP-AI end notification does not occur.
- If the above case occurred and the DRP-AI did not return the end notification for a certain period, the application will abort

  ⇒ ERR_DRPAI_TIMEOUT
- The default threshold for the ERR_DRPAI_TIMEOUT is 5 seconds. Users can change this value in the configuration file (config.ini)

```
# cat config.ini
; Comment …
[DRPAI_TIMEOUT]
5
```

Note that line started with ";" is a comment.

- If the ERR_DRPAI_TIMEOUT occurred, DRP-AI cannot be restarted.
- If you would like to execute the inference again, please reboot the board.

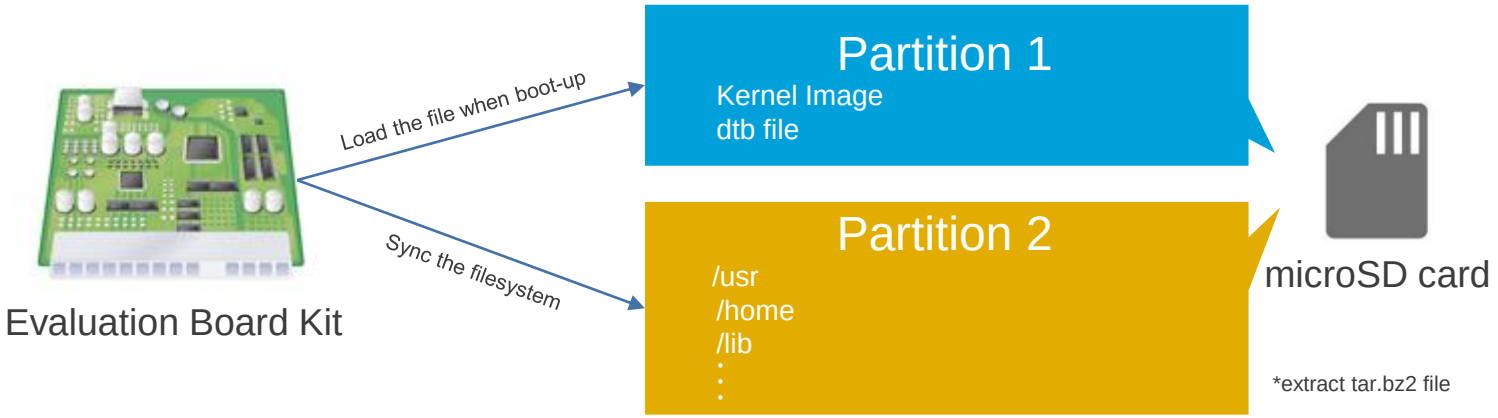
APPENDIX
SD CARD BOOTING

This appendix will explain the SD Card Booting.

WHAT IS SD CARD BOOTING?

- SD Card Booting mount the files used on the board from the microSD card.
- Following three files are necessary for the booting. These files are placed on each partition on microSD card.

File	Description	MicroSD card partition
Kernel Image	Linux kernel image (The boot program)	Partition 1
dtb file	Linux device tree file (The configuration file for booting)	Partition 1
tar.bz2 file	Root filesystem	Partition 2



- This chapter will explain how to setup the partitions, how to allocate each file and the booting instructions.

NECESSARY EQUIPMENT FOR RZ/V2L

Necessary equipment for and its operating environment the SD Card Booting is as follows.

Equipment	Details	Where to obtain
RZ/V2L Evaluation Board Kit	Linux bootloader is pre-written to eMMC. Note: For writing bootloader, refer to RZ/V2L Linux Package Start-up Guide.	Provided by Renesas
Serial to Micro USB Cable	Used for serial communication between PC and the RZ/V2L Evaluation Board.	
AC adapter	Power supply	This equipment should be prepared by users.
USB Cable Type-C	Connect AC adapter and the board.	
microSD card	Operating Environment: Transcend USH-I microSD 300S 16GB	
PC	Used for the serial communication console. Operating Environment : Windows 10	
USB camera	Operating Environment: Logitech C930E WEBCAM	
Linux PC	Used for Setup microSD card. Operating Environment: Ubuntu 20.04	
SD card reader	Used for Setup microSD card.	

NECESSARY EQUIPMENT FOR RZ/V2M, RZ/V2MA

Necessary equipment for and its operating environment the SD Card Booting is as follows.

Equipment	Details	Where to obtain
RZ/V2x Evaluation Board Kit	Linux bootloader is pre-written to eMMC Note: For writing bootloader, refer to RZ/V2x Linux Package Start-up Guide.	Provided by Renesas
Serial to Micro USB cable	Used for serial communication between Linux PC and the board.	This equipment should be prepared by users.
AC adapter	Power supply	
microSD card	Operating Environment: Transcend USH-I microSD 300S 16GB	
PC	Used for the serial communication console. Operating Environment : Windows 10	
USB camera	Operating Environment: Logicool C930e Web Cam *Optional	
Type-C to USB-A Adaptor	Necessary since USB port on the RZ/V2x Evaluation Board Kit is Type-C. Operating Environment: SANWA SUPPLY AD-USB28CAF *Optional	
Linux PC	Used for Setup microSD card Operating Environment: Ubuntu 20.04	
SD card reader	Used for Setup microSD card	

NECESSARY SOFTWARE

- The instructions explained in this chapter use the following files. To obtain the files, refer to the Necessary Files
  - Kernel Image
  - dtb file
  - tar.bz2 file

- This document uses vim as a text editor.

```
$ vi hello.txt
```

- Please pre-install the following necessary software packages on Ubuntu PC.

```
$ 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
$ sudo apt-get install python3-pexpect xz-utils debianutils iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 xterm
$ sudo apt-get install python3-subunit mesa-common-dev p7zip-full
```

- Additional package is required for the SD Card Booting. Please install it by executing the following command.

```
$ sudo apt-get install gparted
```

SD CARD SETUP 1

This chapter will explain the example of setting up the microSD card.

1. Obtain the device number of microSD card.
   1. Before inserting the microSD card to Linux PC, execute the following command on Linux PC.

```
$ df -h
Filesystem              Size                    Used        Avail       Use %       Mounted on
udev                    …                       …           …           …           /dev
…                       …                       …           …           …           …
/dev/sda1               …                       …           …           …           /
tmpfs                   …                       …           …           …           …
```

   2. Insert the microSD card to Linux PC.
   3. Execute the command again.

```
$ df -h
Filesystem              Size                    Used        Avail       Use %       Mounted on
udev                    …                       …           …           …           /dev
…                       …                       …           …           …           …
/dev/sda1               …                       …           …           …           /
tmpfs                   …                       …           …           …           …
/dev/sdb1               …                       …           …           …           …
```

Difference

Device Number
/dev/sdb : Entire SD card.
/dev/sdbN : Partition N of the SD card
Here, the log shows the partition 1.

   4. In the example above, “/dev/sdb” is the device number of microSD card.

SD CARD SETUP 2

Note: This document uses Gparted to create the partition on the microSD card.
Please use the suitable method to create the partition in 2.2 based on your environment.

2. Create the partition on microSD card.

   This chapter uses following software. Please install them according to the Necessary Software.

   - gparted

1. Start the gparted. (GUI will be started)
   Please replace “/dev/sdb” with the device number confirmed in 1.4

```
$ sudo gparted /dev/sdb
```

2. On GUI, create the microSD card partition as follows.

Partition Number	Size	Format
1	128MB (123MiB) or more	fat32
2	The rest	ext4

Do not forget to apply the changes after created the partition.

3. If the partitions are created successfully, “df” command will show the two “/dev/sdb”.

```
$ df -h
Filesystem  Size                 Used        Avail       Use %       Mounted on
…                    …                       …           …           …           …
/dev/sdb1            …                       …           …           …           …
/dev/sdb2            …                       …           …           …           …
```

Created partitions

SD CARD SETUP 3

3. Setup the partition 1.

   1. Following kernel files need to be written to the partition 1.

Board	file	file name
RZ/V2L	Kernel Image	Image-smarc-rzv2l.bin
	dtb file	Image-r9a07g054l2-smarc.dtb
RZ/V2M	Kernel Image	Image-rzv2m.bin
	dtb file	r9a09g011gbg-evaluation-board.dtb
RZ/V2MA	Kernel Image	Image-rzv2ma.bin
	dtb file	r9a09g055ma3gbg-evaluation-board.dtb

   2. Insert the microSD card to Ubuntu PC and execute the following commands.

```
$ sudo mkdir -p /mnt/sd
$ sudo mount /dev/sdb1 /mnt/sd
$ sudo cp <PATH_to_FILE>/<Kernel Image> /mnt/sd
$ sudo cp <PATH_to_FILE>/<dtb file> /mnt/sd
$ sync
$ sudo umount /dev/sdb1
```

Note: 1. Change “/dev/sdb” to your microSD card device number.
2. <PATH_to_FILE> is a path to each file.
3. For <Kernel Image> and <dtb file>, refer to the above table.

SD CARD SETUP 4

4. Setup the partition 2

   1. Following filesystem files need to be written to the partition 2.

Board	tar.bz2 file
RZ/V2L	core-image-weston-smarc-rzv2l.tar.bz2
RZ/V2M	core-image-bsp-rzv2m.tar.bz2
RZ/V2MA	core-image-bsp-rzv2ma.tar.bz2

   2. Insert the microSD card to Ubuntu PC and execute the following commands.

```
$ sudo mount /dev/sdb2 /mnt/sd
$ sudo tar xfj <PATH_to_FILE>/<tar.bz2 file> -C /mnt/sd
$ sync
$ sudo umount /dev/sdb2
```

Note: 1. Change “/dev/sdb” to your microSD card device number.
2. <PATH_to_FILE> is a path to each file.
3. For <tar.bz2 file>, refer to the above table.

5. Eject the microSD card.

   1. Execute the following command and remove the microSD card from the Ubuntu PC.

```
$ sudo eject /dev/sdb
```

INSTALL THE SERIAL PORT DRIVER

The serial communication between Windows PC and RZ/V2L Evaluation Board requires the following driver.

- For RZ/V2L
  - https://ftdichip.com/drivers/vcp-drivers/

1. Download the software "Virtual COM port (VCP) driver" from the windows version “setup executable” on the download page and extract it.
2. Run the **exe file** extracted to install the serial port driver.

- For RZ/V2M, RZ/V2MA
  - https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers

1. Download the software "CP210x Windows Driver" from the download page and extract it.
2. Run the **exe file** extracted to install the serial port driver.

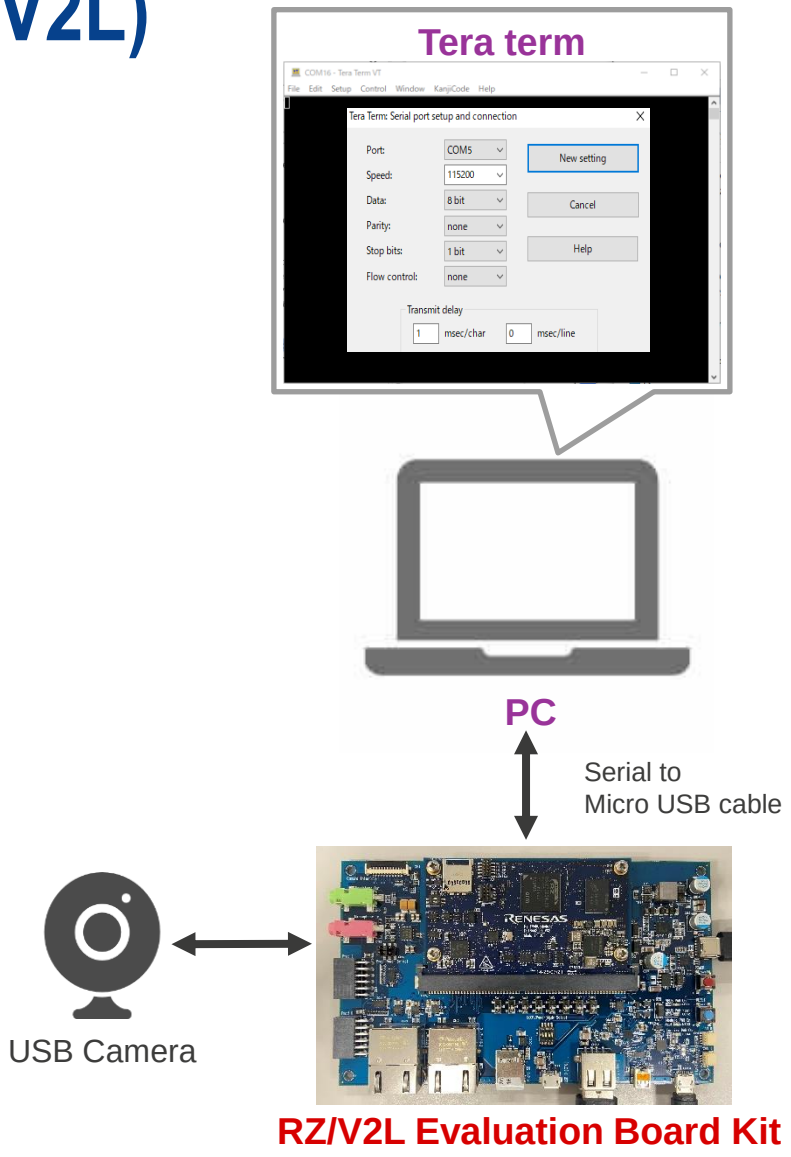
BOOTING OVERVIEW (SD Card Booting for RZ/V2L)

■ Instruction overview (Details will be explained in the next page)

This instruction assumes that following setup are already done.

- SD Card Setup
- Install the Serial Port Driver
- Install the terminal emulator (e.g., Tera term) on Windows PC

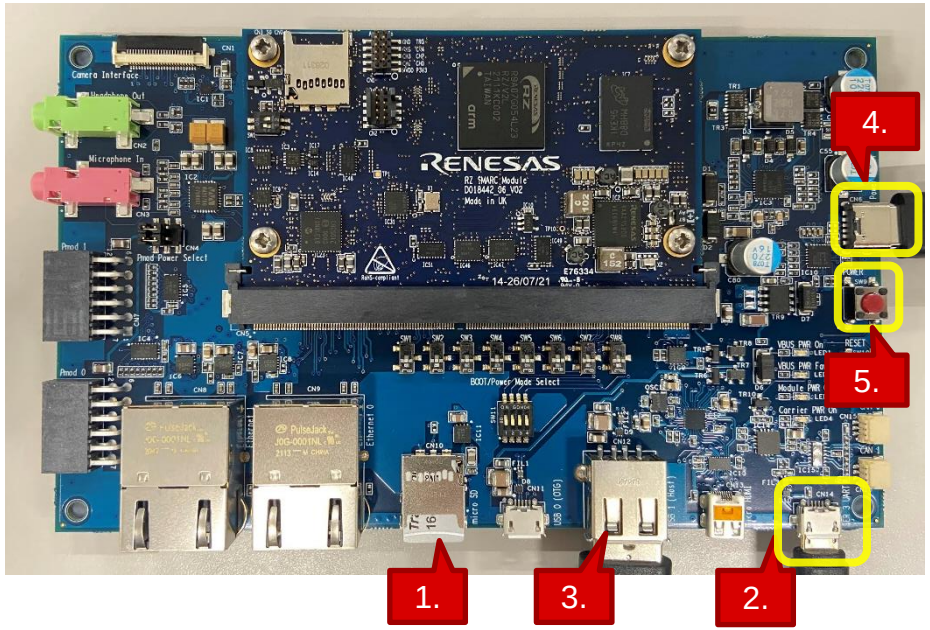
1. Insert the microSD card to the **Board**.
2. Connect the **Board** and **PC** by the Serial to Micro USB cable.
3. Connect the **USB camera** to the **Board**. * Camera Mode Only
4. Connect the power cable to the **Board**.
5. Turn on the **Board**.
6. Open the **Tera term** and connect with COMS.
7. Change the U-boot environment variable on the **Tera term.**
   *Only when changing the booting method.
8. If booted successfully, the login screen will be displayed on the **Tera term.**



BOOTING INSTRUCTIONS DETAILS

Assumption: Executed Preparation.

1. Insert the microSD card to the **Board**.
2. Connect the **Board** and **PC** by the USB Serial to Micro USB cable.
3. Connect the **USB camera** to the **Board**. *Camera Mode only
4. Connect the power cable to the **Board**.
5. Turn on the **Board**.
6. Open the **Tera term** and connect with COMS.
    1. Click the Windows Start button and search “Tera term”
    2. Open the Tera term
    3. Select the “Serial ”
    4. Open the configuration windows from the “Setup” and change the setting as shown in the right.



Tera term: Configuration

Terminal

New-line	Receive: Auto
	Transmit: CR

Serial port

Baud rate	: 115200
Data	: 8bit
Parity	: none
Stop	: 1bit

BOOTING INSTRUCTIONS DETAILS

7. Change the U-boot environment variable on the **Tera term**.

   * Only when changing the booting method.

   1. Keep pressing ENTER key on the **Tera term.**
   2. U-boot console will be activated.
   3. Execute 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
```

8. After the boot-up, the login screen will be shown on the **console**. Log-in to the system using the following information.

   – user: "root"

   – password: none

Other instructions are the same as the Network Booting.

- Execute The Application

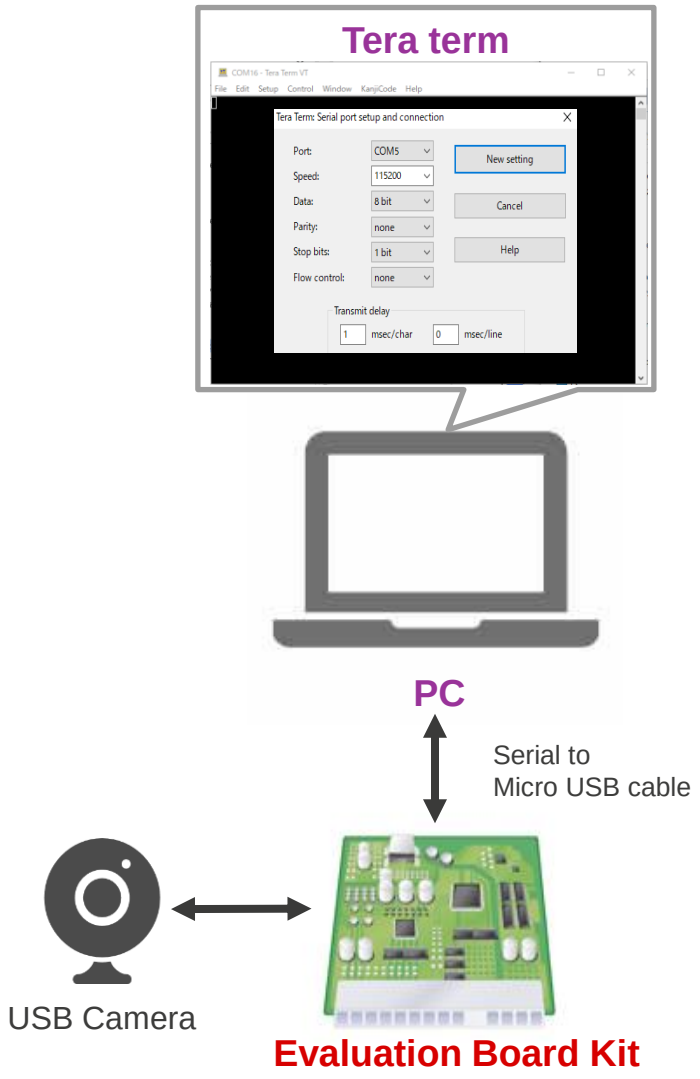
BOOTING OVERVIEW (SD Card Booting for RZ/V2M,RZ/V2MA)

■ Instruction overview (Details will be explained in the next page)

This instruction assumes that following setup are already done.

- SD Card Setup
- Install the Serial Port Driver
- Install the terminal emulator (e.g., Tera term) on Windows PC

1. Insert the microSD card to the **Board**.
2. Connect the **Board** and **PC** by the Serial to Micro USB cable.
3. Connect the power cable to the **Board**.
4. Connect the **USB Camera** to the **Board**. *Only when using the Camera mode
5. Open the **Tera term** and connect with COMS.
6. Turn on the **Board**.
7. Change the U-boot environment variable on the **Tera term.**
   *Only when changing the booting method.
8. If booted successfully, the login screen will be displayed on the **Tera term.**

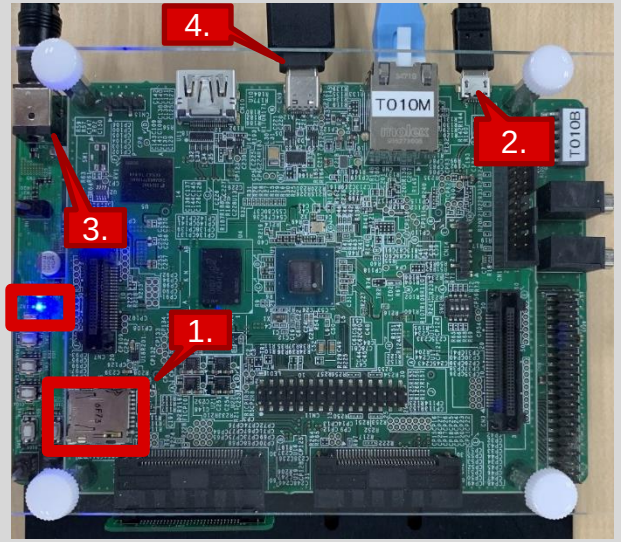


BOOTING INSTRUCTIONS DETAILS

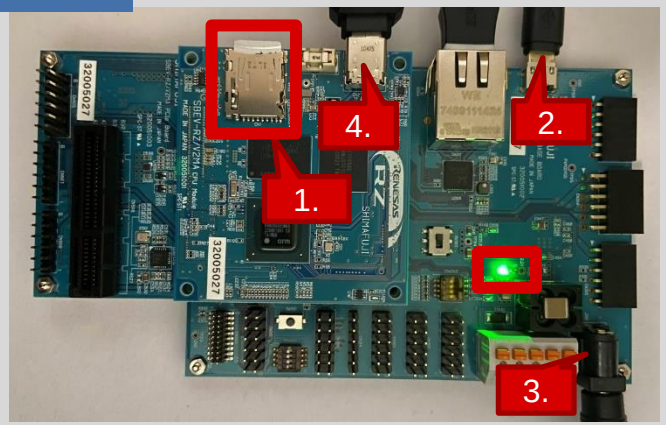
Assumption: Executed Preparation.

1. Insert the microSD card to the **Board**.
2. Connect the **Board** and **PC** by the USB Serial to Micro USB cable.
3. Connect the power cable to the **Board**. *LED lights up
4. Connect the **USB Camera** to the **Board**. *Only when using the Camera mode

RZ/V2M



RZ/V2MA



BOOTING INSTRUCTIONS DETAILS

5. Open the **Tera term** and connect with COMS.
    1. Click the Windows Start button and search “Tera term”
    2. Open the Tera term
    3. Select the “Serial ”
    4. Select the port “COMS *** : **Standard** COM Port (COMS)”
    5. Open the configuration windows from the “Setup” and change the setting as shown in the right.

Tera term: Configuration

Terminal

New-line	Receive: Auto
	Transmit: CR

Serial port

Baud rate	: 115200
Data	: 8bit
Parity	: none
Stop	: 1bit

BOOTING INSTRUCTIONS DETAILS

6. Turn on the **Board**. *Another three LEDs light up

7. Change the U-boot environment variable on the **Tera term**.

   * Only when changing the booting method.

   1. After **6**, keep pressing ENTER key on the **Tera term.**
   2. U-boot console will be activated.
   3. Execute the following commands.

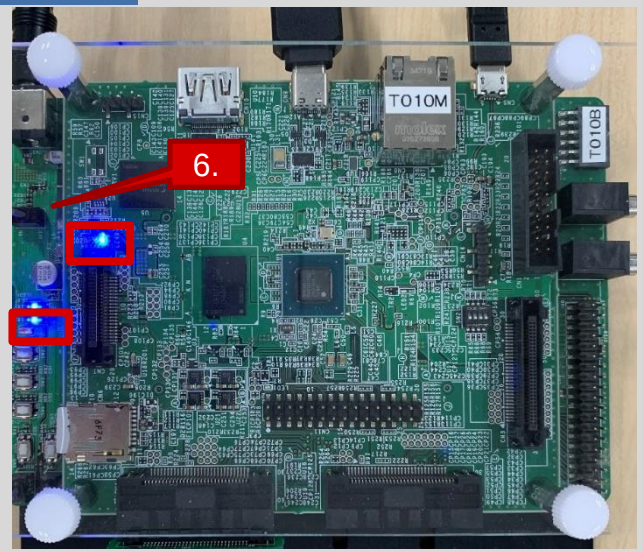
```
=> env default -a
=> setenv bootcmd run bootsd
=> saveenv
=> boot
```

8. After the boot-up, the login screen will be shown on the **Tera term**. Log-in to the system using the following information.
   - user: "root"
   - password: none

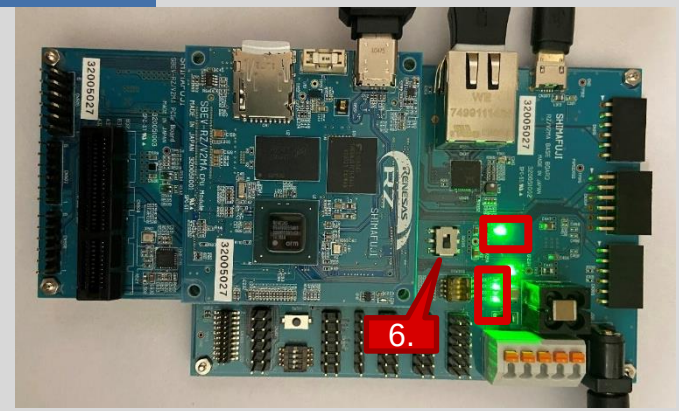
Other instructions are the same as the Network Booting.

- Execute The Application

RZ/V2M



RZ/V2MA



Renesas.com

VERSION HISTORY

Date	Version	Contents
Sep. 29, 2022	7.20	Issued. (Unified AI Evaluation Software Guide for RZ/V2L, RZ/V2M, RZ/V2MA)