WebUI Package V1.0.0 for Verified Linux Package

VLP/F, VLP/G

Release Note

R01US0611EJ0100
Rev. 1.00
Jun 30, 2023

Introduction

This release note describes the contents, building procedures and important points of the WebUI Package V1.0.0 for RZ/Five Verified Linux Package (hereinafter referred to as “VLP/F”) and RZ/G Verified Linux Package (hereinafter referred to as “VLP/G”).

Webmin is the WebUI released in this package. It is a web-based system administration tool to provide the graphic user interface by network.

Contents

1. Release Items

- **Name and version**

WebUI Package V1.0.0 for below packages:

- RZ/Five Verified Linux Package Version 3.0.4 (hereinafter referred to as “VLP/F v3.0.4”) or later.
- RZ/G Verified Linux Package Version 3.0.3(hereinafter referred to as “VLP/G v3.0.3”) or later.

- **Distribution method**

Please visit the site below and create an account to download the packages. Basic packages of VLP/F v3.0.2-update1 which are listed in **Table 1-1** can be downloaded.

RZ/Five product page:
https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzfive-risc-v-general-purpose-microprocessors-risc-v-cpu-core-andes-ax45mp-single-10-ghz-2ch-gigabit-ethernet

RZ/Five Verified Linux Package [5.10-CIP]:
https://www.renesas.com/us/en/software-tool/rzfive-verified-linux-package-510-cip

Basic packages of VLP/G v3.0.3 which are listed in **Table 1-2** can be downloaded.

RZ Family:
https://www.renesas.com/products/microcontrollers-microprocessors/rz-arm-based-high-end-32-64-bit-mpus

RZ/G Verified Linux Package [5.10-CIP]:
https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzg-linux-platform/rzg-marketplace/verified-linux-package/rzg-verified-linux-package

- **Target board**

RZ/Five reference board

- RZ/Five Evaluation board Kit (smarc-rzfive) (*1)
  - RZ/Five SMARC Module Board (P/N: RTK9743F01C01000BE)
  - RZ SMARC Series Carrier Board (P/N: RTK97X4XXXB00000BE)

- RZ/G2L Evaluation Board Kit PMIC version (*2):
  - RZ/G2L SMARC Module Board v2.1
  - RZ SMARC Series Carrier Board v4.0

- RZ/G2LC Evaluation Board Kit (*3):
  - RZ/G2LC SMARC Module Board v1.0
  - RZ SMARC Series Carrier Board v4.0

- RZ/G2UL Evaluation Board Kit (*4):
  - RZ/G2UL SMARC Module Board v1.0
  - RZ SMARC Series Carrier Board v4.0

(*1) “RZ/Five Evaluation board Kit” includes the RZ/Five SMARC Module Board and the RZ SMARC Series Carrier Board.
The “Evaluation board Kit for RZ/Five MPU” will be called “RZ/Five Evaluation Kit” in the next section.

(*2) “RZ/G2L Evaluation Board Kit” and “RZG2L Evaluation Board Kit PMIC version” includes the RZ/G2L SMARC Module Board and the RZ SMARC Series Carrier Board.
The “Evaluation board Kit for RZ/G2L MPU” will be called “RZ/G2L Evaluation Kit” in the next section.

(*3) “RZG2LC Evaluation Board Kit” includes the RZ/G2LC SMARC Module Board and the RZ SMARC Series

Carrier Board.
The “Evaluation board Kit for RZ/G2LC MPU” will be called “RZ/G2LC Evaluation Kit” in the next section.

(*4) “RZ/G2UL Evaluation board Kit” includes the RZ/G2UL SMARC Module Board and the RZ SMARC Series Carrier Board.
The “Evaluation board Kit for RZ/G2UL MPU” will be called “RZ/G2UL Evaluation Kit” in the next section.

- **Functions**
  Linux BSP
  - Linux Kernel
  - Linux Drivers
  - Graphics Libraries (*)
  - Codec Libraries (*)

  GUI Framework
  - Qt (LGPL version) (*)

(*) Only supported by VLP/G v3.0.3.

- **File contents**
  BSP for VLP/F v3.0.4 is delivered by the files listed in **Table 1-1**. BSP for VLP/G v3.0.3 is delivered by the files listed in **Table 1-2.**

Table 1-1. RZ/Five Verified Linux Package

Basic packages

File	Description
RTK0EF0045Z0025AZJ-v3.0.4.zip(*1)	RZ/Five Verified Linux Package. This file includes the **Yocto recipe packages** and the necessary documents.
rzfive_vlp_v3.0.4.tar.gz(*1)	**Yocto recipe packages**
oss_pkg_rzfive_v3.0.4.7z(*1)	Open source software packages See the Note below before download
r01us0608ej0102-rz-five(Release Note).pdf	Release note for VLP/F v3.0.4
r01us0618ej0100-rz-five(Linux StartUp Guide smarcEVK RZFive).pdf	Documents describing building method, booting method and the required settings of bootloader for RZ/Five.

Table 1-2. VLP/G v3.0.3 Basic packages

File	Description
RTK0EF0045Z0021AZJ-v3.0.3.zip (*1)	Verified Linux Package. This file includes the **Yocto recipe packages** and the necessary documents.
rzg_bsp_v3.0.3.tar.gz (*1)	**Yocto recipe packages**
oss_pkg_rzg_v3.0.3.7z (*1)	Open source software packages See the Note below before download
r01us0553ej0106-rz-g(Release Note).pdf	Release note for VLP/G v3.0.3
r01us0554ej0105-rz-g(Component List).pdf	Component list
r01us0556ej0101-rz-g(Board_StartUp_Guide _smarcEVK).pdf	Documents describing booting method and the required settings of bootloader for **RZ/G2L, RZ/G2LC, and RZ/G2UL**.

Table 1-3. Additional files for WebUI Package v1.0.0

File	Description
RTK0EF0045Z88001ZJ_v1.0.0.zip (*1)	WebUI (Webmin) package. This file includes the **Yocto recipe packages** for WebUI (Webmin) and the necessary documents.
rzg_bsp_webui_v1.0.0.tar.gz	**Yocto recipe packages** to add the WebUI (Webmin) module.
r01us0611ej0100-rz-g(Release Note WebUI).pdf	This document

(*1) These packages are provided “AS IS” with no warranty and the license which is described in the source code. Please check the contents of the license, then consider the applicability to the product carefully.

Note) Open source software packages contain all source codes of OSSs. These are the same versions of OSSs used when VLP/F and VLP/G was verified.
If you are just evaluating VLP/F and RZ/Five series or VLP/G and RZ/G series, open source software packages are not mandatory to use. Usually, all the software can be built without using these files if your build machine is connected to the Internet.

Open source software packages are required for an “offline” environment. The word “offline” means an isolated environment which does not connect to any network. VLP can always build images in this “offline” environment by using these packages without affected from changes of original repositories of OSSs. 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.
Most bootable images that VLP supports can be built on an “offline” environment. Please refer to **2. Build environment**.

2. Build environment

The environment for building the BSP is listed in Table 2-1. Please refer to the below documents for details about setting up the environment:

SMARC EVK of RZ/Five Linux Start-up Guide (*)

SMARC EVK of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five Start-up Guide

(*) This document is newly prepared from VLP/Fv3.0.4.

A Linux PC is required for building the software.

A Windows PC can be used as the serial terminal interface with software such as TeraTerm.

Table 2-1. Equipment and Software Necessary for Developing Environments of RZ/Five and RZ/G Linux Platform

Equipment		Description
Linux Host PC		Used as build/debug environment 100GB free space on HDD is necessary
	OS	Ubuntu 20.04 LTS 64 bit OS must be used.
Windows Host PC		Used as debug environment, controlling with terminal software
	OS	Windows 10
	Terminal software	Used for controling serial console of the target board Tera Term (latest version) is recommended Available at https://ttssh2.osdn.jp/index.html.en
	VCP Driver	Virtual COM Port driver which enables to communicate Windows Host PC and the target board via USB which is virtually used as serial port. Available at: ● http://www.ftdichip.com/Drivers/VCP.htm

Most bootable images VLP supports can be built on an “offline” environment.
The word “offline” means an isolated environment which does not connect to any network. Since VLP includes all necessary source codes of OSS except for the Linux kernel, VLP can always build images in this “offline” environment without affected from changes of repositories of OSS. Also, this “offline” environment reproduces the same images as the images which were verified by Renesas.

Below images can be built “offline”.

- core-image-minimal
- core-image-bsp
- core-image-weston (*)
- core-image-qt (*)

(*) Only supported by VLP/G v3.0.3.

Below are not available in the “offline” environment. Please connect your Linux Host PC to the internet.

- Preparing a Linux Host PC

3. Building Instructions of a BSP

3.1 Building images to run on the board

(1) Build VLP image

Please refer to the Release Note for VLP or SMARC EVK of RZ/Five Linux Start-up Guide. And please follow the build instruction until the VLP image is built. After that please start from the next step to build including WebUI.

(2) Add Webmin to rootfs image as WebUI framework

Assume your working directory to build a VLP image is `$WORK`. Run the command below to edit "`~/$WORK/build/conf/bblayers.conf`".

```
$ bitbake-layers -F add-layer $(pwd)/../meta-openembedded/meta-webserver
```

Add below line to "`~/$WORK/build/conf/local.conf`".

```
IMAGE_INSTALL:append = " webmin "
RRECOMMENDS_webmin-module-fdisk_remove = "parted"
```

There are 2 Webmin modules are prepared as demo to help the evaluation of Webmin. If you want to add them to the built Webmin, please refer to the Section 3.2 Webmin – (3) Provided Webmin demo modules .

After the above procedure is finished, the "online" environment is required. Please change the line in the "`~/$WORK/build/conf/local.conf`" as below:

```
BB_NO_NETWORK = "0"
```

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

(3) Start a build

Run the commands below to start a build. Building an image which includes the Webmin.

```
$ bitbake core-image-<minimal/bsp/Weston/qt>
```

After the build is successfully completed, a similar output will be seen, and the command prompt will return.

```
NOTE: Tasks Summary: Attempted 3788 tasks of which 265 didn't need to be rerun and a
ll succeeded.
```

All necessary files listed in **Table 3-2 and Table 3-3** will be generated by the bitbake command and will be located in the **`build/tmp/deploy/images`** directory.

VLP can build a few types of images listed in **Table 3-1**. Please refer to the "Component list" for details about components of each image.

For the booting method and the required settings, please refer to:

"SMARC EVK of RZ/Five Linux Start-up Guide (R01US0618EJ)"

"SMARC EVK of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five Start-up Guide (R01US0556)".

The loader files written at the time of mass production may not be the latest. When using EVK for the first time, be sure to write Boot loader files generated by building latest BSP to your board.

Table 3-1. Supported images of VLP

Image name	Description
core-image-minimal	Image which includes minimal set of components
core-image-bsp	Image which includes minimal set of components, audio components, and some useful tools
core-image-weston	Standard image with graphics support.
core-image-qt	Enable Qt LGPL version.

Table 3-2. Image files for RZ/Five

RZ/Five	**Linux kernel**	Image-smarc-rzfive.bin
	Device tree file	Image-r9a07g043f01-smarc.dtb
	root filesystem	*<image name>*-smarc-rzfive.tar.bz2
	Boot loader	● fit-smarc-rzfive.srec ● spl-smarc-rzfive.srec
	Flash Writer	Flash_Writer_SCIF_RZFIVE_SMARC.mot

Table 3-3. Image files for RZ/G2L, RZ/G2LC and RZ/G2UL

RZ/G2L PMIC ver	**Linux kernel**	Image-smarc-rzg2l.bin
	Device tree file	Image-r9a07g044l2-smarc.dtb
	root filesystem	*<image name>*-smarc-rzg2l.tar.bz2
	Boot loader	● bl2_bp-smarc-rzg2l_pmic.srec ● fip-smarc-rzg2l_pmic.srec
	Flash Writer	Flash_Writer_SCIF_RZG2L_SMARC_PMIC_DDR4_2GB_1PCS.mot
RZ/G2LC	**Linux kernel**	Image-smarc-rzg2lc.bin
	Device tree file	Image-r9a07g044c2-smarc.dtb
	root filesystem	*<image name>*-smarc-rzg2lc.tar.bz2
	Boot loader	● bl2_bp-smarc-rzg2lc.srec ● fip-smarc-rzg2lc.srec
	Flash Writer	Flash_Writer_SCIF_RZG2LC_SMARC_DDR4_1GB_1PCS.mot
RZ/G2UL	**Linux kernel**	Image-smarc-rzg2ul.bin
	Device tree file	Image-r9a07g043u11-smarc.dtb
	root filesystem	*<image name>*-smarc-rzg2ul.tar.bz2
	Boot loader	● bl2_bp-smarc-rzg2ul.srec ● fip-smarc-rzg2ul.srec
	Flash Writer	Flash_Writer_SCIF_RZG2UL_SMARC_DDR4_1GB_1PCS.mot

3.2 Webmin

(1) How to use Webmin

After the build is completed, please refer to “SMARC EVK of RZ/Five Linux Start-up Guide” or “SMARC EVK of RZ/G2L, RZ/G2LC, RZ/G2UL, RZ/V2L, and RZ/Five Start-up Guide” to boot the RZ/Five and RZ/G evaluation board. Please run the command below on terminal to start Webmin service on board.

```
$ systemctl start webmin
```

On user PC’s browser, access the Webmin by URL http://board_ip_address:10000.

Figure 3-1 Input the board’s ip address to load the Webmin login page

Login with following account.

- Username: admin
- Password: password

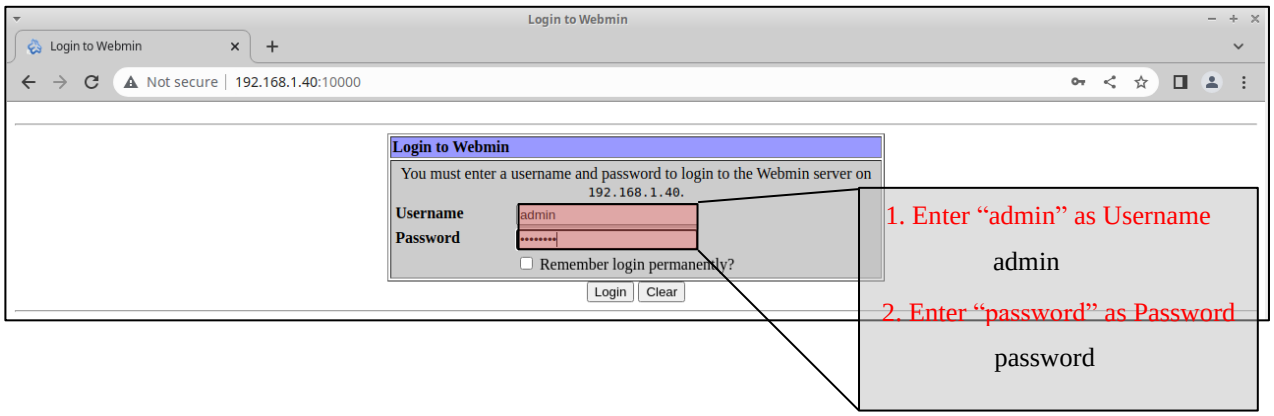

Figure 3-2 Input Username and Password to login Webmin

When login success, the index page as below should be shown.

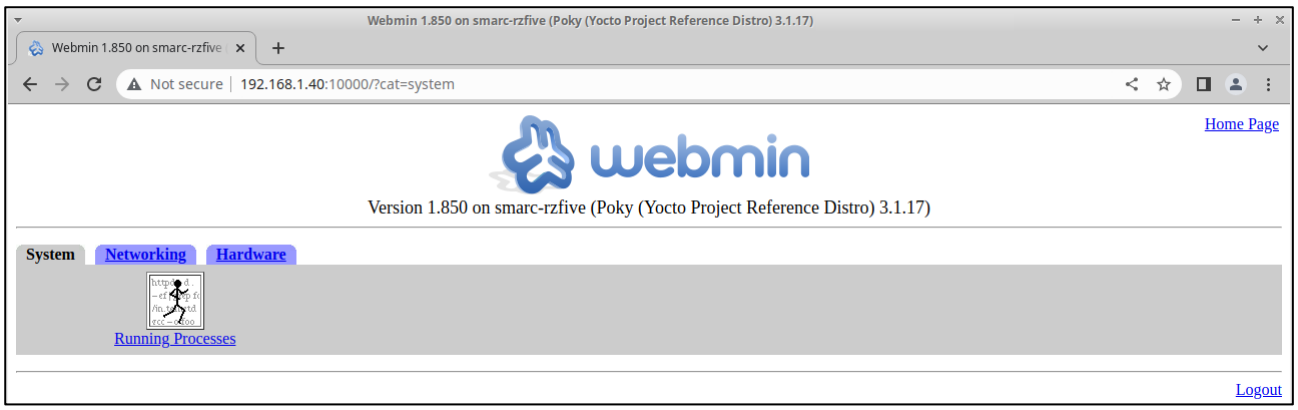

Figure 3-3 Webmin index page

(2) Function provided by Webmin by default

The Webmin package provides below functions in default:

- Check memory usage.
- Check CPU usage.
- Run Linux command and check execution result.

1. Check memory usage

To check the memory usage, please select the **Running Processes** from Webmin index page and click the **Memory** label. User can check the total memory, the free memory size and cached memory size.

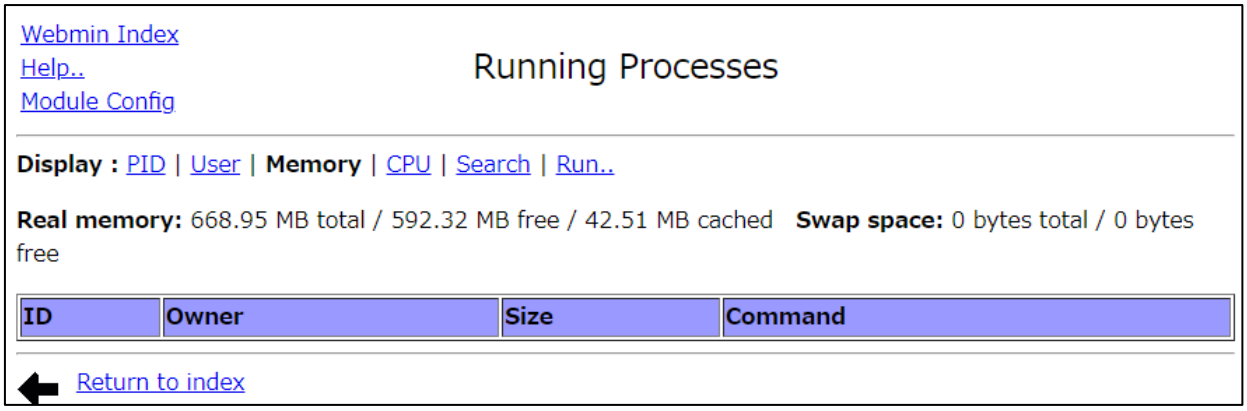

Figure 3-4 Check memory usage

2. Check CPU usage

To check the CPU usage, select the **Running Processes** from Webmin index page and click **CPU** label. User can check average CPU load in 1 minutes, 5 minutes and 15 minutes.

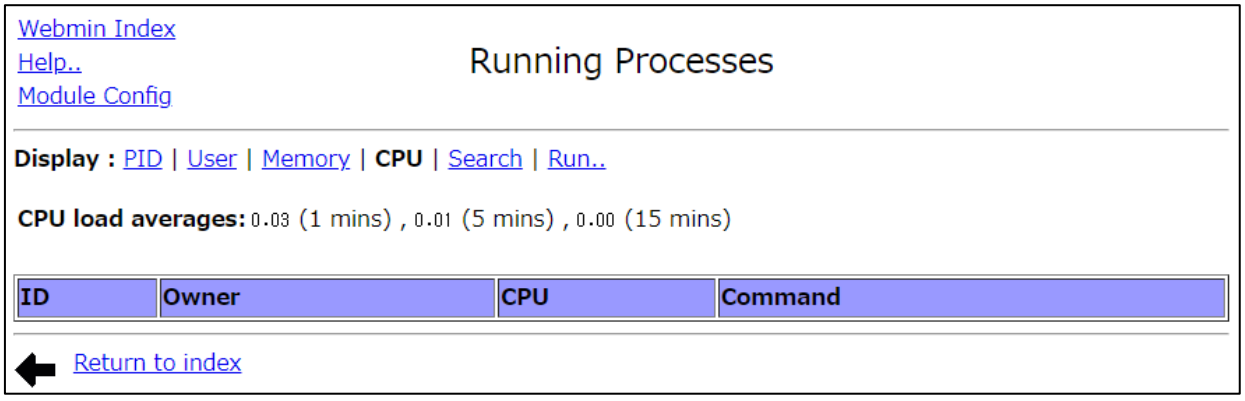

Figure 3-5 Check CPU usage

3. Run Linux command and check execution result

Select the **Running Processes** from Webmin index page and click **Run..** label.

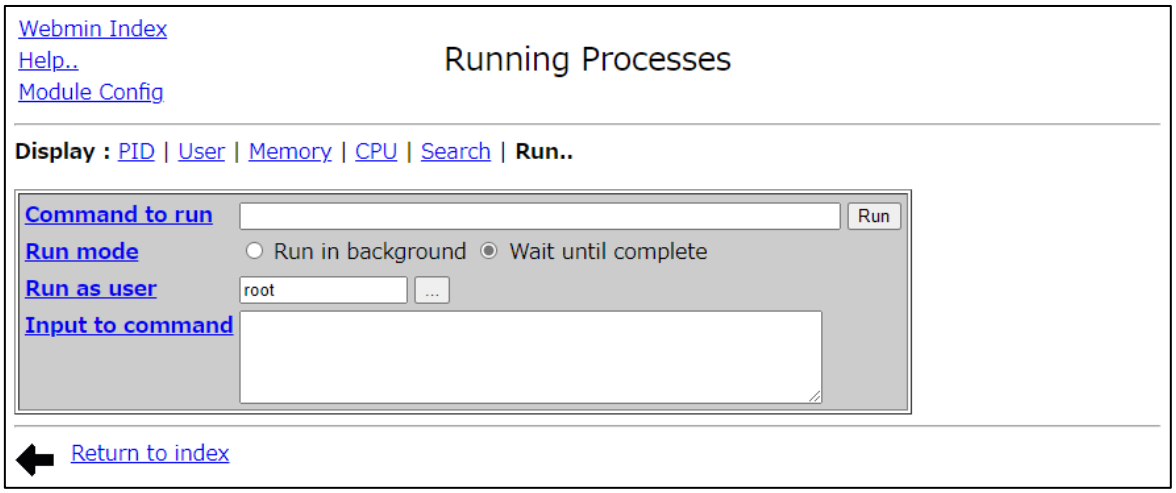

Figure 3-6 Page to run Linux command

Box **Command to run** is for typing the Linux command to run on board.

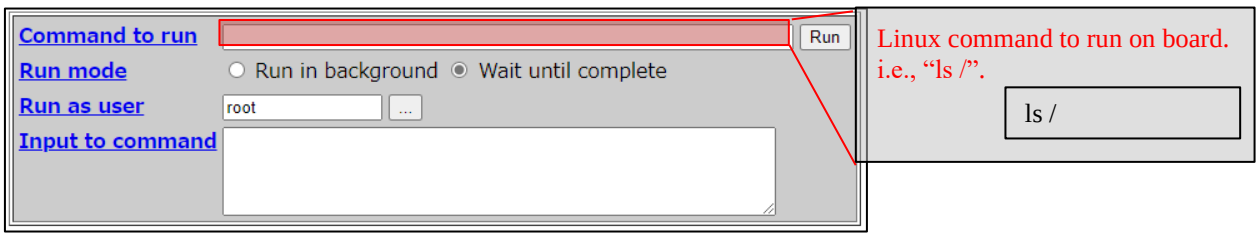

Figure 3-7 Input the Linux command to “Command to run”

For each command, **Run mode** is for choosing to run it in background or wait until the command is complete.

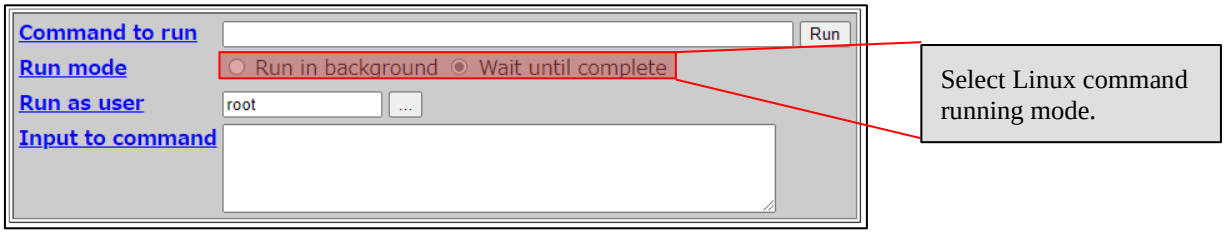

Figure 3-8 Select the run mode of Linux command

Box **Run as user** is to choose the Linux command run as which user.

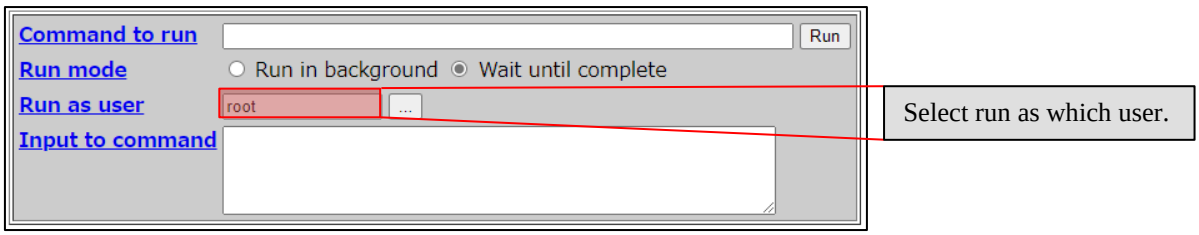

Figure 3-9 Select user to run this Linux command

For box **Input to command**, any text input to this field will be passed to the command as standard input.

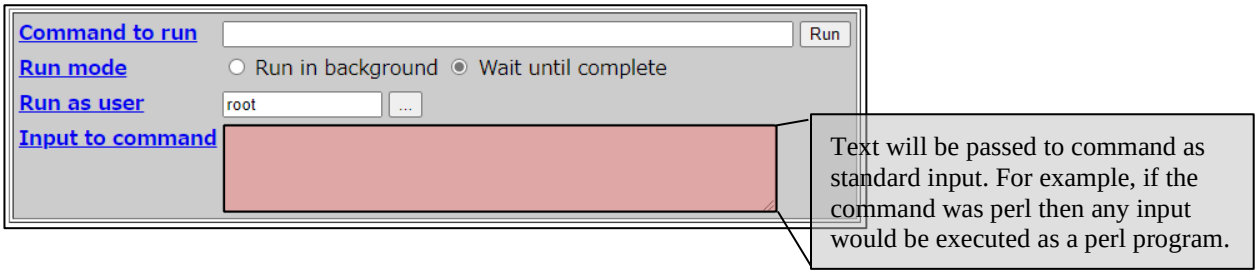

Figure 3-10 Standard input pass to the executed command

User can type the Linux command in box Command to run and click **Run** button to run Linux command. When the command execution is finished, it will update this page to show the output. For example, when user ran the command "ls / -lh" to check the directories in root path, this page will be updated with the result of running "ls / -lh" on board.

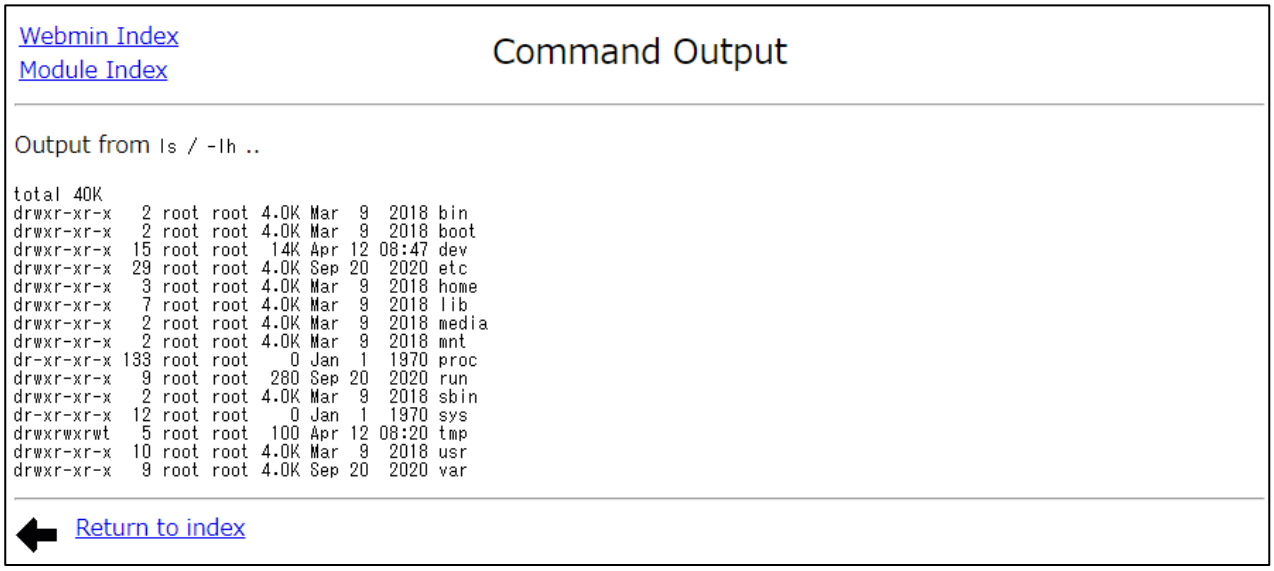

Figure 3-11 Execution result of command "ls / -lh"

(3) Provided Webmin demo modules

1. How to build Webmin demo modules

2 Webmin modules are prepared and provided as demo. These 2 modules are

- Disk-usage: Counts the amount of disk space used by directories specified by the user, and displays a tree showing the usage of all directories and sub-directories.
- Firewall: This module can be used to set up a firewall on a Linux system with IPtables enabled, or edit any part of an existing firewall.

To build these 2 modules with webmin, please follow the steps below.

Decompress the Webmin module with below command.

```
$ cd ~/$WORK
$ cp ../<package download directory>/rzg_bsp_webui_v1.0.0.tar.gz .
$ tar zxvf rzg_bsp_webui_v1.0.0.tar.gz
$ cd ~/$WORK/build
```

Run the command below to edit "`~/$WORK/build/conf/bblayers.conf`".

```
$ bitbake-layers -F add-layer $(pwd)/../meta-rz-features/meta-rz-demos
```

Add below line to "~/$WORK/build/conf/local.conf".

```
IMAGE_INSTALL:append = " iptables cronie procps "
IMAGE_INSTALL:append = " packagegroup-webmin-demos "
```

Start the build with command below.

```
$ bitbake core-image-<minimal/bsp/Weston/qt>
```

2. Webmin Firewall demo module

When the build process was finished, user can see the **Linux Firewall** icon under category **Networking**.

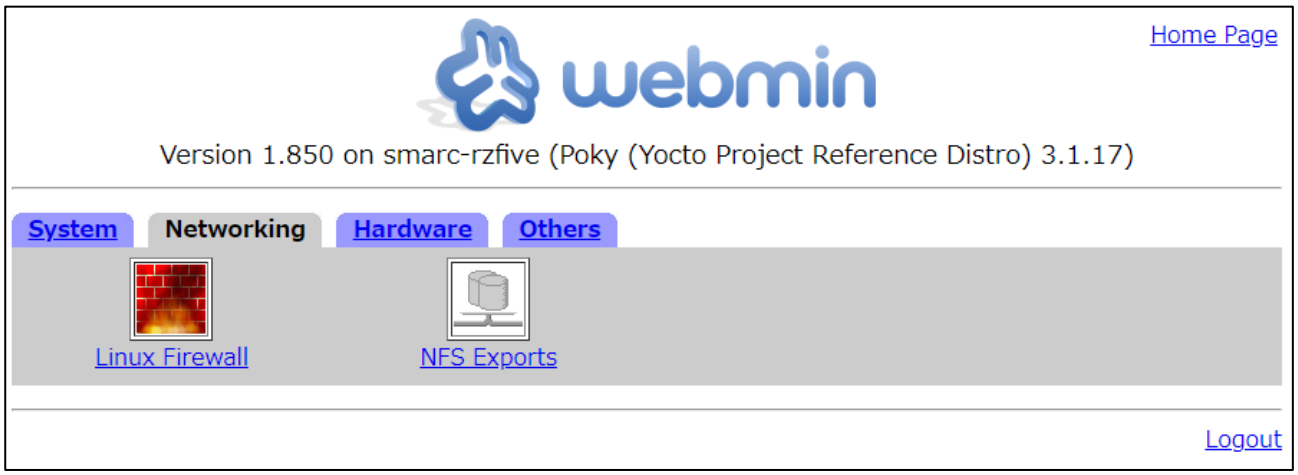

Figure 3-12 Firewall icon under category "Networking"

In the start page of Firewall module, users can set initial setting for this module. User can allow all traffic or specify the ethernet port for a specific rule. Select the desired configuration and click the **Setup Firewall** button to enter the setting page.

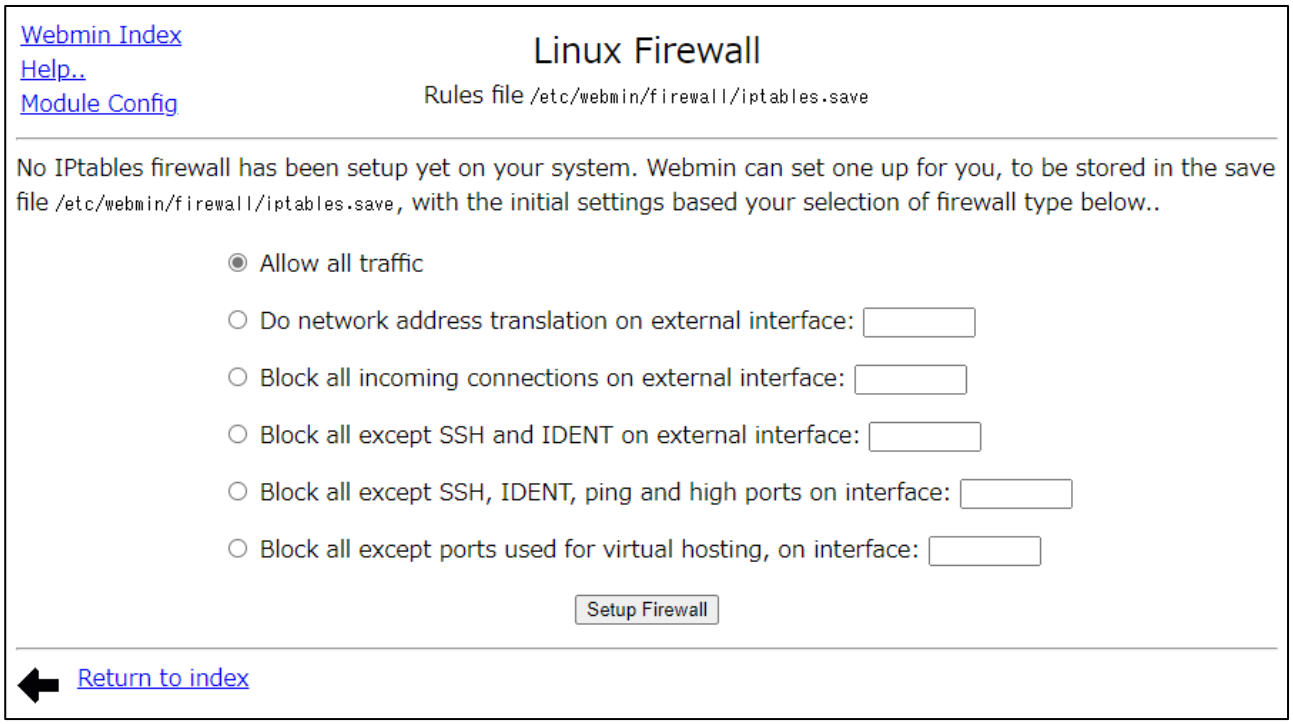

Figure 3-13 Start page of Firewall module

In the setting page for Firewall module, user is allowed to add settings for firewall rules in detail to control the network traffic.

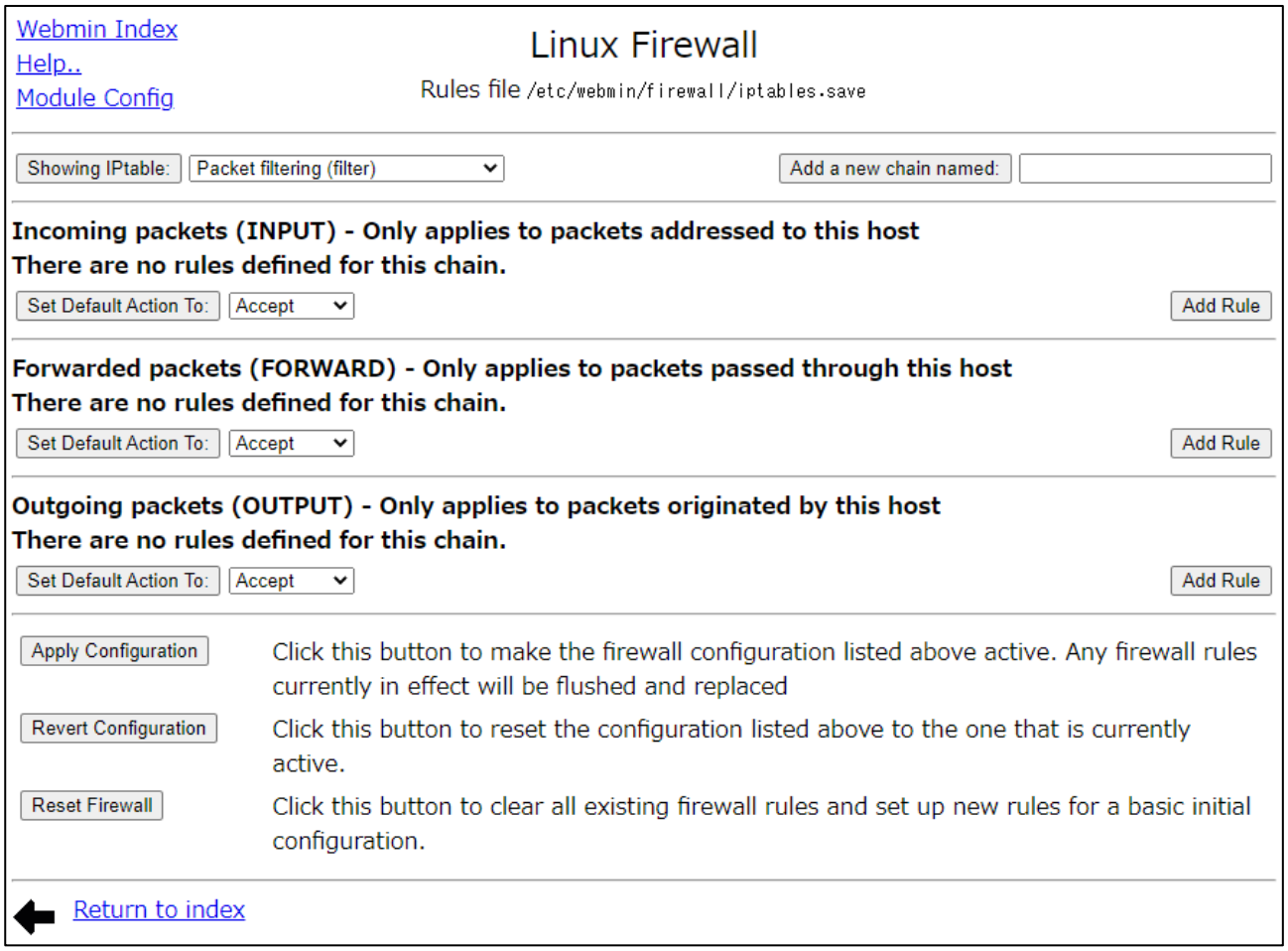

Figure 3-14 Setting page of Firewall module

3. Webmin Disk Usage demo module

The disk-usage module is under category **Others**.

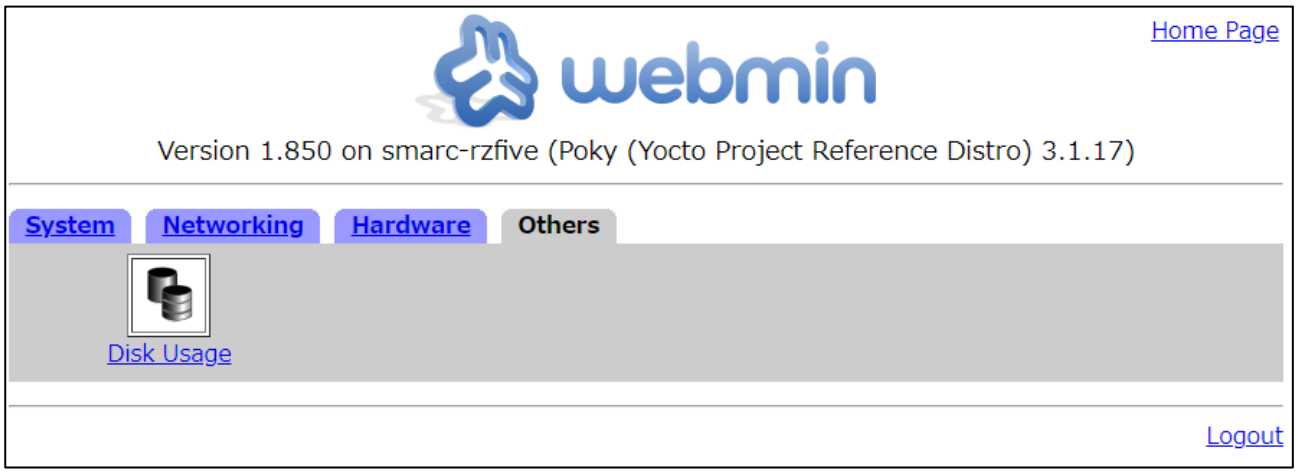

Figure 3-15 Disk Usage under category "Others"

Click the icon **Disk Usage** can enter to the start page of this module. The start page is as **Figure 3-16**. In the start page, it is showing the size of each directory under the path specified by user. Button **Setup Scheduled Collection** allows user to set a schedule when to collect the usage of disk. And button **Collect Usage Now** will update the info of the disk

usage shown in start page of this module. User can specify the directory to check its disk usage, please click the **Module Config** to set the directory.

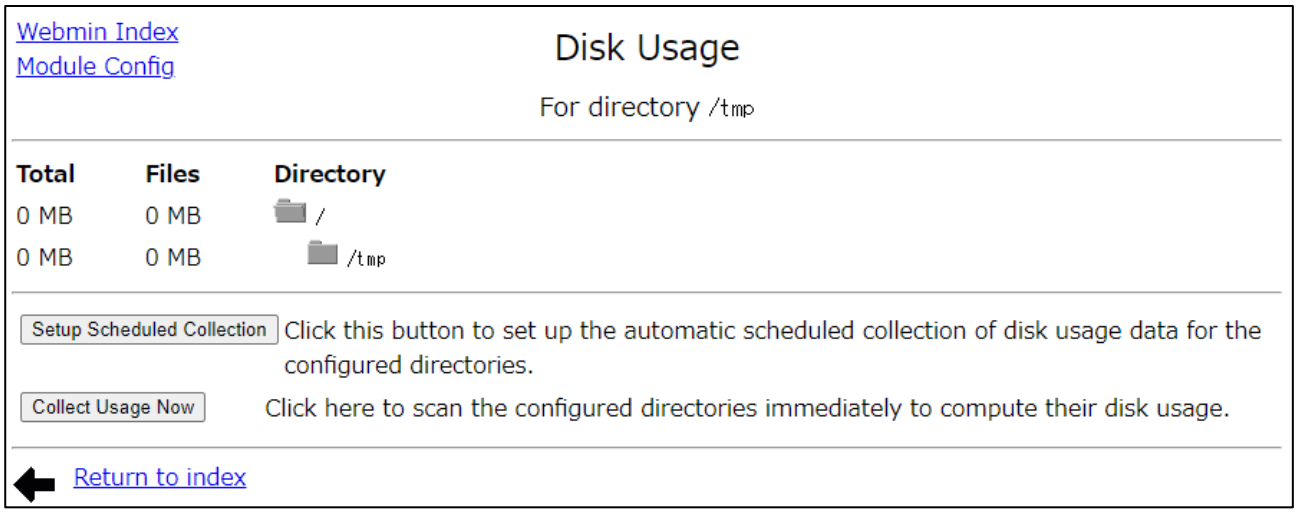

Figure 3-16 Start page of Disk Usage

(4) How to add Webmin module

Webmin has provided many modules to sufficient its usage. This part is describing how to integrate the add-on modules to Webmin.

1. To install module which does not require to download

There are 2 kinds of modules for Webmin. Some modules already integrate into webmin.tar.gz. they are listed in **Table 3-4**. It does not require Yocto to download these modules when building the Webmin.

Table 3-4. Modules included in Webmin.tar.gz

webmin-module-acl	webmin-module-idmapd	webmin-module-passwd
webmin-module-adsl-client	webmin-module-inetd	webmin-module-phpini
webmin-module-ajaxterm	webmin-module-init	webmin-module-postgresql
webmin-module-apache	webmin-module-inittab	webmin-module-ppp-client
webmin-module-at	webmin-module-ipsec	webmin-module-proc
webmin-module-backup-config	webmin-module-iscsi-client	webmin-module-proftpd
webmin-module-bacula-backup	webmin-module-iscsi-server	webmin-module-quota
webmin-module-bandwidth	webmin-module-iscsi-target	webmin-module-raid
webmin-module-bind8	webmin-module-iscsi-tgtd	webmin-module-samba
webmin-module-bsdfdisk	webmin-module-krb5	webmin-module-servers
webmin-module-change-user	webmin-module-ldap-client	webmin-module-shell
webmin-module-cron	webmin-module-ldap-server	webmin-module-shorewall6
webmin-module-custom	webmin-module-ldap-useradmin	webmin-module-smart-status
webmin-module-dhcpd	webmin-module-logrotate	webmin-module-software
webmin-module-disk-usage	webmin-module-lpadmin	webmin-module-sshd
webmin-module-exports	webmin-module-lvm	webmin-module-status

webmin-module-fail2ban	webmin-module-man	webmin-module-syslog
webmin-module-fdisk	webmin-module-mediatomb	webmin-module-syslog-ng
webmin-module-file	webmin-module-mon	webmin-module-system-status
webmin-module-filemin	webmin-module-mount	webmin-module-tcpwrappers
webmin-module-firewall	webmin-module-mysql	webmin-module-time
webmin-module-firewall6	webmin-module-net	webmin-module-tunnel
webmin-module-firewalld	webmin-module-nis	webmin-module-updown
webmin-module-grub	webmin-module-package-updates	webmin-module-useradmin
webmin-module-webmin	webmin-module-webmincron	webmin-module-webminlog
webmin-module-xinetd	webmin-module-pam	webmin-module-usermin
webmin-module-htaccess-htpasswd		

When adding the modules which does not require to download. Assume the targeted module is with name "newmodule".
To add the "newmodule" to rootfs image, please edit file "`~/$WORK/ meta-rz-features/meta-rz-demos/recipes-webadmin/packagegroups/packagegroup-webmin-demos.bb`" and add code below.

```
RDEPENDS_packagegroup-webmin-demos += " webmin-module-newmodule "
```

2. For module which need to download

Other modules are in Webmin github repository, but not in table **Table 3-4**. They are the modules which require to download when building the Webmin.

To add these modules, we need to do the following things:

- Prepare file module.info of target module in Yocto recipe.
- Generate the sha256sum of the file module.info.
- Add the github URL and sha256sum to Yocto recipe.

Prepare file module.info of target module in yocto layer. Assume the target module with name "newmodule" and it is in the Webmin's Github repository. Users need to download the file module.info of "newmodule" from Webmin repository. Make directory "newmodule" under path "`~/$WORK/meta-rz-features/meta-rz-demos /recipes-webadmin/webmin/files/`".

```
$ cd ~/$WORK/meta-rz-features/meta-rz-demos/recipes-webadmin/webmin/files/
$ mkdir newmodule
```

Copy the downloaded module.info to the created directory.

```
$ cp ${download_path}/module.info ./newmodule
```

Generate the sha256sum of the file module.info. The sha256sum is required by Yocto recipe, please use the below command to generate the sha256sum for module.info file.

```
$ sha256sum ./newmodule/module.info
```

Add the github URL and sha256sum to Yocto recipe. Edit file "`~/$WORK/meta-rz-features/meta-rz-demos/recipes-webadmin/webmin/webmin_1.850.bbappend`" to add the repository information of "newmodule" and path of module.info to SRC_URI.

When editing SRC_URI, beside of the github repository, *subpath*, *destsuffix* and *name* need to be specified. For the *subpath* and *name*, they should be same with the module name, so they are "subpath=newmodule" and "name=newmodule". *destsuffix* should be set as "${S}/newmodule".

```
SRC_URI_append += " git://${WEBMIN_GIT};protocol=https;subpath=newmodule;destsuffix=
${S}/newmodule;name=newmodule \
file://newmodule/module.info;subdir=${S};name=newmodule.module.info \
"
```

Note) The "${WEBMIN_GIT}" in variable SRC_URI_append should be the URL for target respository. When installing the module from Webmin official Github repository, it does not need to change.

Add source revision hash code of the module to Yocto recipe. Edit file "~/$WORK/meta-rz-features/meta-rz-demos/recipes-webadmin/webmin/webmin_1.850.bbappend" to add below code.

```
SRCREV_newmodule = "9dc7367dd6ba1e6090edfbb81915ac515da0ef49"
```

Add sha256sum of the module.info to Yocto recipe.
Assume the generated sha256sum is "xxx". Please edit file "~/$WORK/meta-rz-features/meta-rz-demos/recipes-webadmin/webmin/webmin_1.850.bbappend" to add below code.

```
SRC_URI[newmodule.module.info.sha256sum] = "xxx"
```

3. When add-on module requires special kernel configuration

Please use the bitbake command as below to enter the menuconfig and enable the configuration that requested to enable.

```
$ bitbake linux-renesas -c menuconfig
```

When the options were selected, please save the configuration and restart the build process with bitbake command.

4. Start the build progress

When the modification for target module was done, issue the bitbake command to start the build progress and integrate the Webmin module into rootfs image.

```
$ bitbake core-image-minimal
```

(5) How to make Webmin module

Beside of the standard module supported by official Webmin. User can also make Webmin module to customize the Webmin application. Please refer to the below URL link for learning how to make Webmin module.

- https://doxfer.webmin.com/Webmin/Development
- https://doxfer.webmin.com/Webmin/Module_Development

1. How to add the self-made Webmin module

To add self-made Webmin module, users need to fork the source code of Webmin in version 1.850 on Github. Add the self-made module to Webmin and push to the forked Webmin repository. When building the self-made module with Webmin, please refer the content in **3.2-(4)-3 For module which need to download**.

4. Components

The components which are commonly used in this release are listed in **Table 4-1**. Please also refer to the manifest file for details.
Please refer to the below example:

```
$WORK/build/tmp/deploy/images/<board>/core-image-<image-name>-<board>.manifest
```

Note:

- <board> is smarc-rzfive, smarc-rzg2ul or smarc-rzg2lc.
- <image-name> is minimal, bsp, Weston or qt.

Table 4-1 Versions of commonly used components

Components	VLP/G v3.0.3	VLP/F v3.0.4
Linux kernel	5.10.158-cip22	5.10.175-cip29
GCC	8.3.0 (Arm GCC 8.3-2019.03)	8.3.0 (RISC-V GCC 8.3-2019.03)
glibc	2.28	2.28
busybox	1.31.1	1.30.1
openssl	1.1.1n	1.1.1n
Webmin	1.850	1.850

5. Restrictions

(1) Docker

Docker container is not supported.

(2) EMMC boot

EMMC boot will be supported in the next release.

6. Notes

(1) GPLv3 packages

In this release, the GPLv3 packages are disabled as default in *build/conf/local.conf*:

```
INCOMPATIBLE_LICENSE = "GPLv3 GPLv3+"
```

If you want to use GPLv3, just hide this line:

```
#INCOMPATIBLE_LICENSE = "GPLv3 GPLv3+"
```

7. Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jun. 39, 2023	-	First edition issued.

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/