AVNET®

/ RZBOARD V2L

Linux Yocto Development Guide

V2.7

Copyright Statement:

Disclaimer:

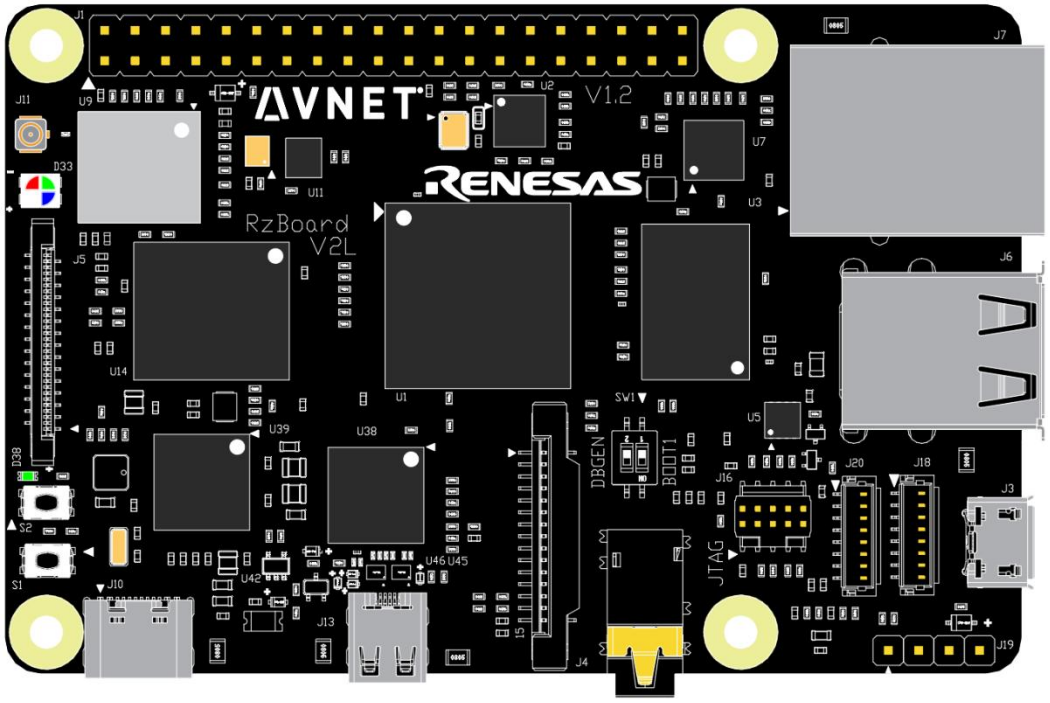
Revision History

Version	Note	Author	Release Date
v1.0	Initial version	Lily	2022/07/08
v2.0	Update Yocto Project to 3.1.14	Lily	2022/09/28
v2.1	Procedure and readability improvements	Peter	2022/11/08
v2.2	1.Description of reorganizing structure of Yocto source code 2. Script ***make_rz_uboot.sh*** is now provided on github	Lily	2022/11/28
v2.3	Miscellaneous document readability edits	Peter	2022/12/05
V2.4	1. Provided direct links to Renesas support packages 2. Git config is unneeded	Evan	2023/04/05
V2.5	1. Updated link to Renesas codec support package	Evan	2023/04/26
V2.6	1. Make updates based on the updates of Renesas.	Lily	2023/08/09
V2.7	1. Make updates based on the updates of Renesas:Updated Yocto Project to 3.1.21, Kernel version to 5.10.175	Lily	2023/10/31

Contents

Summary

RZBoard is a development SBC board designed and manufactured by Avnet, based on RZ/V2L 64bit Arm A55 MPUs with DRP-AI acceleration from Renesas Electronics.



RZBoard is fitted with a Renesas RZ/V2L dual-core MPU (p/n R9A77G054L2)

This document describes key aspects of development on RZBoard using Yocto Linux

---

Note: Two methods are detailed in this document for building the Linux image and boot files:

1) Yocto build (simpler than the standalone method)
2) Standalone build

For simplicity, Renesas recommends using the **Yocto build method**

https://renesas.info/wiki/RZ-G/RZG_yocto#Online_vs_Offline_Yocto_build

Chapter 1 Build Instructions

1.1 Setup Build Environment

To setup the build environment the following resources are required:

- Hardware: At least 300GB of disk space and 8GB of RAM
- Software: Ubuntu 64-bit OS, 20.04 LTS version (Ubuntu Desktop or Ubuntu Server version).
- You could also run the Ubuntu 64-bit OS on virtual machine or in docker container.

The following packages are required for the development environment.

The required packages can be installed using the bash script below:

```
$ sudo apt-get update
$ sudo apt install -y 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 \
rsync curl locales bash-completion
```

1.2 Fetch Source Code

1.2.1 Download Renesas software packages

Due to licensing restrictions on the Renesas website, users are required to download in person. The links have been provided below:

#	Package Name	Version	File to be downloaded
1	RZ/V Verified Linux Package	V3.0.4	RTK0EF0045Z0024AZJ-v3.0.4.zip
2	RZ MPU Graphics Library	Evaluation Version V1.1.0	RTK0EF0045Z13001ZJ-v1.1.0_EN.zip
3	RZ MPU Codec Library	Evaluation Version V1.1.0	RTK0EF0045Z15001ZJ-v1.1.0_EN.zip
4	RZ/V2L DRP-AI Support Package	V7.40	rzv2l-drp-ai-support-package-version-740
5	RZ/V2L Multi-OS Package	V1.11	r01an6238ej0111-rzv2l-cm33-multi-os-pkg.zip

After downloading these packages, copy them to the **home directory (~/)** to prepare for the next step.

Note: The “Evaluation” packages contain a time limitation that stops the software after a few hours.

For more information please refer to RZ/V2L Wiki.

1.2.2 Extract Renesas software packages

Download a script to extract the file.

```
$ cd ~/
```

```
$ wget https://raw.githubusercontent.com/Avnet/meta-rzboard/rzboard_dunfell_5.10_v3/tools/create_yocto_rz_src.sh
$ chmod a+x create_yocto_rz_src.sh
$ ls ~/
RTK0EF0045Z0024AZJ-v3.0.4.zip     RTK0EF0045Z15001ZJ-v1.1.0_EN.zip
r01an6238ej0111-rzv2l-cm33-multi-os-pkg.zip  RTK0EF0045Z13001ZJ-v1.1.0_EN.zip
create_yocto_rz_src.sh           r11an0549ej0740-rzv2l-drpai-sp.zip
```

Run the script to generate **yocto_rzboard/** directory.

```
$ ./create_yocto_rz_src.sh
$ ls yocto_rzboard/
extra  meta-gplv2  meta-openembedded  meta-qt5  meta-renesas  meta-rz-features  meta-
virtualization  poky
```

1.2.3 Download meta-rzboard

```
$ cd ~/yocto_rzboard
$ git clone https://github.com/Avnet/meta-rzboard.git -b rzboard_dunfell_5.10_v3
```

So far, all the yocto related sources are in place.

```
$ ls ~/yocto_rzboard
extra  meta-gplv2  meta-openembedded  meta-qt5  meta-renesas  meta-rz-features  meta-rzboard
meta-virtualization  poky
```

1.3 Yocto Build of BSP

1.3.1 Edit build configuration

```
cd ~/yocto_rzboard
$ mkdir -p ./build/conf
$ cp meta-rzboard/conf/rzboard/* build/conf/
$ ls build/conf/
bblayers.conf  local.conf  site.conf
```

The directory to which all Yocto packages are downloaded, can be set by User in **conf/local.conf**:

```
DL_DIR ?= "${HOME}/downloads"
```

1.3.2 Setting build environment and Build

```
$ cd ~/yocto_rzboard/
$ source poky/oe-init-build-env build/
$ bitbake avnet-core-image
```

After the build has successfully completed, the output files are deployed in:

/yocto_rzboard/build/tmp/deploy/images/rzboard/

Flash_Writer_SCIF_rzboard.mot	FlashWriter image tool
bl2_bp-rzboard.srec	BL2 bootloader file in S-record format
fip-rzboard.srec	BL31 bootloader plus u-Boot packages in S-record format
avnet-core-image-rzboard-xxxx.rootfs.wic	System image, this includes: Linux kernel, DTB and root file system.
Image	Kernel image
rzboard.dtb	RZBoard device tree binary
overlays/rzboard-*.dtbo	RZBoard device tree overlay binary
avnet-core-image-rzboard-xxxx.rootfs.tar.bz2	System image compressed archive file

Chapter 2 Standalone Build of Kernel and u-Boot

This chapter describes how to build u-Boot and the kernel using either Yocto SDK or ARM GCC in a standalone (offline) environment.

2.1 Cross-compile tool chain

The cross-compile tool chain that is used, can be ARM GCC or Yocto SDK.

2.1.1 ARM GCC

Download the tool chain for the A-profile architecture on arm Developer GNU-A Downloads page. It is recommended to use the 10.3 version for this release. You can download the "gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu.tar.xz ", and decompress the file into a local directory.

```
$ mkdir ~/toolchain
$ tar -xJf gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu.tar.xz -C ./toolchain
```

Execute the following command to check that the toolchain can be directly run.

```
$ cd toolchain/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu/bin/
$ ./aarch64-none-linux-gnu-gcc -v
```

To compile a project with ARM GCC, first set the environment with the following commands before building:

```
$ TOOLCHAIN_PATH=$HOME/toolchain/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu/bin
$ export PATH=$TOOLCHAIN_PATH:$PATH
$ export ARCH=arm64
$ export CROSS_COMPILE=aarch64-linux-gnu-
```

2.1.2 Yocto SDK

Generate an SDK from the Yocto Project build environment with the following command after generating the image in the previous chapter.

```
$ cd ~/yocto_rzboard/
$ source poky/oe-init-build-env build
$ bitbake avnet-core-image -c populate_sdk
```

The SDK will be generated under: ~/yocto_rzboard/build/tmp/deploy/sdk/./poky-glibc-x86_64-avnet-core-image-aarch64-rzboard-toolchain-3.1.21.sh

Execute this script to install the SDK. The default location is **/opt** but it can be placed anywhere on the host machine.

```
$ ./poky-glibc-x86_64-avnet-core-image-aarch64-rzboard-toolchain-3.1.21.sh

Poky (Yocto Project Reference Distro) SDK installer version 3.1.21
==========================================================================
Enter target directory for SDK (default: /opt/poky/3.1.21):
You are about to install the SDK to "/opt/poky/3.1.21". Proceed [Y/n]? Y
Extracting SDK.....................done
```

When using Yocto SDK to compile a project, first execute the following command to configure the environment variables:

```
$ source /opt/poky/3.1.21/environment-setup-aarch64-poky-linux
```

2.2 Build u-Boot in a standalone environment

To build a bootloader for RZBoard, the user should get cloned different repositories, such as renesas-u-Boot.git, trusted-firmware-a.git and bootparameter code. After compiling the code of each repository, the user has to perform some steps to generate the final boot image.

A shell script named ***make_rz_uboot.sh*** is provided to simplify these processes. So the user can download the all code and build with this script. The boot image build steps are as follows:

Download the bash script into the **tmp** directory and change the file mode:

```
$ mkdir ~/tmp
$ cd ~/tmp
$ wget https://raw.githubusercontent.com/Avnet/rzboard-build-tools/main/make_rz_uboot.sh
$ chmod a+x make_rz_uboot.sh
```

Execute the script with parameter **-g** to download the code first:

```
$ ./make_rz_uboot.sh -g
```

Then execute the script with parameter **-rz** to start the build process:

```
$ ./make_rz_uboot.sh -rz
```

The following outputs are generated by default in the tmp directory:

- bl2_bp_rzboard.srec
- fip_rzboard.srec

Please refer to RZBoard-Linux-Yocto-Usermanual.pdf to update these boot images to the board.

2.3 Build Kernel in a standalone environment

Get the Linux source code

```
$ cd ~/
$ git clone https://github.com/Avnet/renesas-linux-cip.git -b rzboard_v2l_v5.10.175
```

Check that the environment variables are correctly set:

```
$ echo $CROSS_COMPILE  $ARCH
```

Build the kernel sources

```
$ cd ~/renesas-linux-cip
$ make distclean
$ make rzboard_defconfig
$ make -j4
```

Execute the ‘ls’ command to view the Image, dtb file and dtbo files after compilation.

```
$ ls arch/arm64/boot/Image
$ ls arch/arm64/boot/dts/renesas/rzboard.dtb
$ ls arch/arm64/boot/dts/renesas/overlays/rzboard-*.dtbo
```

Execute the following command to compile the kernel modules, and install the modules to *rootfs* in the current directory.

```
$ make modules
$ make modules_install INSTALL_MOD_PATH=./rootfs
```

Chapter 3 Flash-Programming and Board Operation

To program the generated new Bootloader and System image files into RZBoard's eMMC memory, use the procedure described in the ***RZBoard-Linux-Yocto-UserManual***

(This also detailed in section 12 of the ***RZBoard Hardware User Guide***)

For guidance on power-up RZBoard, the boot-up process, and how to exercise the supported BSP features of RZBoard, please refer to ***RZBoard-Linux-Yocto-UserManual***

All Avnet documents are accessible via the RZBoard product page at https://www.avnet.me/rzboard

Chapter 4 Appendix

4.1 Hardware Documents

For hardware details please refer to:

- ***RZBoard Hardware User Guide***
-
- ***RZBoard Block Diagram***
-

4.2 Software Documents and Links

RZBoard supports Yocto Linux, for additional information, please refer to the following documents accessible from the RZBoard product page at https://www.avnet.me/rzboard

- ***RZBoard Linux Yocto User Manual***
- - Describes how to reflash RZBoard and aspects of the BSP functionality
-
- ***RZBoard Linux Yocto Development Guide***
- - Detailed guidance on how to rebuild the Linux system image using Yocto (this document)
-
- ***RZ/G2 Group Linux BSP Porting Guide*** (this is applicable to RZ/V2L)
- https://www.renesas.com/us/en/document/mas/rzg2-group-linux-bsp-porting-guide
-
- ***RZ/V and RZ/G key Wiki pages on Renesas.info***
- https://renesas.info/wiki/RZ-V
- https://renesas.info/wiki/RZ-G
- https://renesas.info/wiki/RZ-G/RZG_kernel
- https://renesas.info/wiki/RZ-G/RZG_DeviceTree
- https://renesas.info/wiki/RZ-G/RZ-G2_BSP
- https://renesas.info/wiki/RZ-G/RZ-G2_BSP_Porting
-

4.3 Contact Information

Product Webpage: https://www.avnet.me/rzboard