

AVNET RZBOARD V2L Engineering Services Evaluation and Development Kits Installation Guide

[Home](#) » [AVNET](#) » AVNET RZBOARD V2L Engineering Services Evaluation and Development Kits Installation Guide 



RZBOARD V2L Engineering Services Evaluation and Development Kits Installation Guide

Contents

- [1 RZBOARD V2L Engineering Services Evaluation and Development Kits](#)
- [2 Summary](#)
- [3 Chapter 1 Build Instructions](#)
- [4 Chapter 2 Standalone Build of Kernel and u-Boot](#)
- [5 Chapter 3 Flash-Programming and Board Operation](#)
- [6 Chapter Appendix](#)
- [7 Documents / Resources](#)
 - [7.1 References](#)
- [8 Related Posts](#)

RZBOARD V2L Engineering Services Evaluation and Development Kits

Copyright Statement:

- The RZBoard V2L and its related intellectual property are owned by Avnet.
- Avnet has the copyright of this document and reserves all rights. Any part of the document should not be modified, distributed or duplicated in any form without written permission issued by Avnet.

Disclaimer:

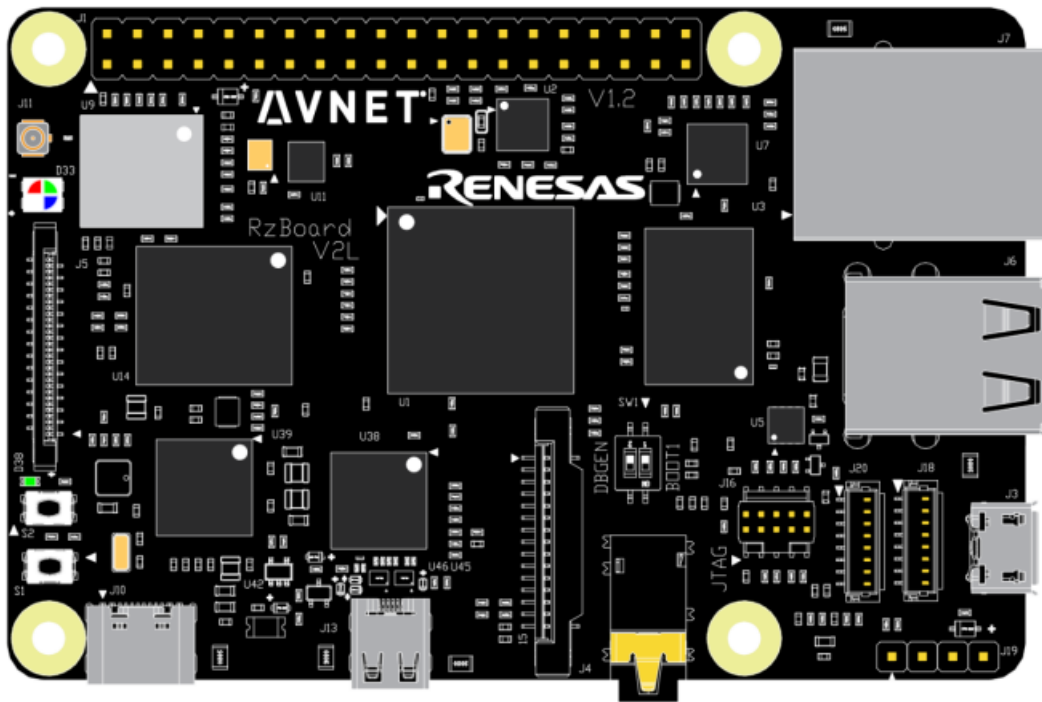
- Avnet does not make warranty of any kind, either expressed or implied, as to the program source code, software and documents provided along with the products, and including, but not limited to, warranties of fitness for a particular purpose; The entire risk as to the quality or performance of the program is with the user of these products.

Revision History

Version	Note	Author	Release Date
v1.0	Initial version	Lily	7/8/2022
v2.0	Update Yocto Project to 3.1.14	Lily	9/28/2022
v2.1	Procedure and readability improvements	Peter	11/8/2022
v2.2	1.Description of reorganizing structure of Yocto source code 2. Script make_rz_uboot.sh is now provided on github	Lily	11/28/2022
v2.3	Miscellaneous document readability edits	Peter	12/5/2022

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 libssl1.2-dev xterm p7zip-full libyaml-dev \rsync curl locales bash-completion
```

Set Git configuration:

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "you@example.com"
```

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 following six software packages from the Official RZ/V2L Website

#	Package Name	Version	File to be downloaded
1	RZ/V Verified Linux Package	v3.0.0-update2	RTK0EF0045Z0024AZJ-v3.0.0-updat e2.zip
2	RZ MPU Graphics Library	Evaluation Version V1 .2	RTK0EF0045Z13001ZJ-v1.21_EN.zip
3	RZ MPU Codec Library	Evaluation Version V0 .58	RTK0EF0045Z15001ZJ-v0.58_EN.zip
4	RZ/V2L DRP-AI Support Package	V7.20	r11an0549ej0720-rzv2l-drpai-sp.zip
5	RZ/V2L ISP Support Package	V1.20	r11an0561ej0120-rzv2l-isp-sp.zip
6	RZ/V2L Multi-OS Package	V1.02	r01an6238ej0102-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/metarzbord/rzboard\_dunfell\_5.10/tools/create\_yocto\_rz\_src.sh
$ chmod a+x create_yocto_rz_src.sh
$ ls ~/
```

```
create_yocto_rz_src.sh RTK0EF0045Z0024AZJ-v3.0.0-update2.zip
r01an6238ej0102-rzv2l-cm33-multi-os-pkg.zip RTK0EF0045Z13001ZJ-v1.21_EN.zip
r11an0549ej0720-rzv2l-drpai-sp.zip RTK0EF0045Z15001ZJ-v0.58_EN.zip
r11an0561ej0120-rzv2l-isp-sp.zip
```

Run the script to generate yocto_rzboard/ directory.

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

1.2.3 Download meta-rzboard

```
$ cd ~/yocto_rzboard
$ git clone https://github.com/Avnet/meta-rzboard.git-brzboard\_dunfell\_5.10
```

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

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

1.3 Yocto Build of BSP

1.3.1 Edit build configuration

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

flashwriter 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 8.3 version for this release. You can download the “gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu.tar.xz”, and decompress the file into a local directory.

```
$ mkdir ~/toolchain
```

```
$ tar -xJf gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu.tar.xz -C ~/toolchain
```

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

```
$ cd toolchain/gcc-arm-8.3-2019.03-x86_64-aarch64-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-8.3-2019.03-x86_64-aarch64-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-coreimage-aarch64-rzboard-toolchain-3.1.14.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-3.1.14.sh
```

Poky (Yocto Project Reference Distro) SDK installer version 3.1.14

Enter target directory for SDK (default: /opt/poky/3.1.14):

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

```
$ source /opt/poky/3.1.14/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-uBoot.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: Chapter 1

```
$ 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-brzboard\_v2l\_v5.10
```

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 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>



<https://www.avnet.me/rzboard>

Documents / Resources

 	AVNET RZBOARD V2L Engineering Services Evaluation and Development Kits [pdf] Installation Guide RZBOARD V2L Engineering Services Evaluation and Development Kits, RZBOARD V2L, Engineering Services Evaluation and Development Kits, Services Evaluation and Development Kits, Evaluation and Development Kits, Development Kits
--	--

References

- [Renesas.info](#)
- [Downloads | GNU-A Downloads – Arm Developer](#)
- [GitHub - Avnet/meta-rzboard: Yocto meta-layer for Renesas V2L board](#)
- [GitHub - Avnet/renesas-linux-cip: Based on https://github.com/renesas-rz/rz_linux-cip.git branch rzv2l-cip41](#)
- [raw.githubusercontent.com/Avnet/rzboard-build-tools/main/make_rz_uboot.sh](#)
- [RZ/G Series 32/64-bit MPU - Renesas.info](#)
- [RZ/G2 and RZ/V2 BSP Information Page - Renesas.info](#)
- [RZ-G/RZ-G2 BSP Porting - Renesas.info](#)
- [Device Tree - Renesas.info](#)
- [RZ/G kernel Information - Renesas.info](#)
- [RZ/G Yocto Information - Renesas.info](#)
- [RZ/V Series 64-bit MPU - Renesas.info](#)
- [RZ/V2L SMARC Board by Renesas - Renesas.info](#)
- [RZBoard V2L | Avnet Boards](#)
- [RZ/V2L - General-Purpose Microprocessor Equipped With Renesas' Original AI Accelerator "DRP-AI", 1.2GHz Dual-Core Arm® Cortex®-A55 CPU, 3D Graphics, and Video Codec Engine | Renesas](#)
- [renesas.com/us/en/document/mas/rzg2-group-linux-bsp-porting-guide](#)