e² studio Integrated Development Environment

Debugging Linux Applications Running on the R-Car (with the Bus Monitor Trace Window as an Analysis Tool)

Introduction

This application note introduces the procedure for acquiring trace data of R-Car V4H through the command-line interface (CLI) of Linux which is running on an R-Car and displaying the data in the bus monitor trace window for use of the e² studio for R-Car.

Target Device

R-Car V4H

Contents

1. Environment

The following lists the elements of the environment used in the descriptions in this application note.

Table 1-1 List of Environment Elements

Item	Detail
R-Car V4H Reference Board	White Hawk [RTP8A779G2ASKB0F10SA001]
Host PC	• Windows 10 (for use of the terminal display console or running Linux in a VirtualBox virtual environment) • Ubuntu 20.04 LTS (e^2 studio, TFTP server, or NFS server)
e^2 studio for R-Car	Version 2023-03 rcar-xos_tool_e2studio_ubuntu_2023-03.R20230313-1010.tar.gz
R-Car V4H SDK	V3.12.0 rcar-xos_developer-adas-bootloader_v3.12.0_release.tar.gz rcar-xos_platform-sdkX_v3.12.0_release.tar.gz (sample source code)
R-Car V4H yocto Linux (Kernel version)	V5.10.147 rcar-xos_tool_yocto_linux_6.13.0.tar.gz

This application note describes the procedure for using an environment in which a Linux environment has been set up for the R-Car on the reference board. For the procedure for setting up the Linux environment, refer to the user's manual included in the SDK.

2. Connection Diagram

Figure 2-1 shows a connection between the reference board for the R-Car V4H and a host PC. The IP addresses for use between the host PC and the reference board are set as shown in the figure. Specify the IP addresses which are to be set to suit the environment you are using.
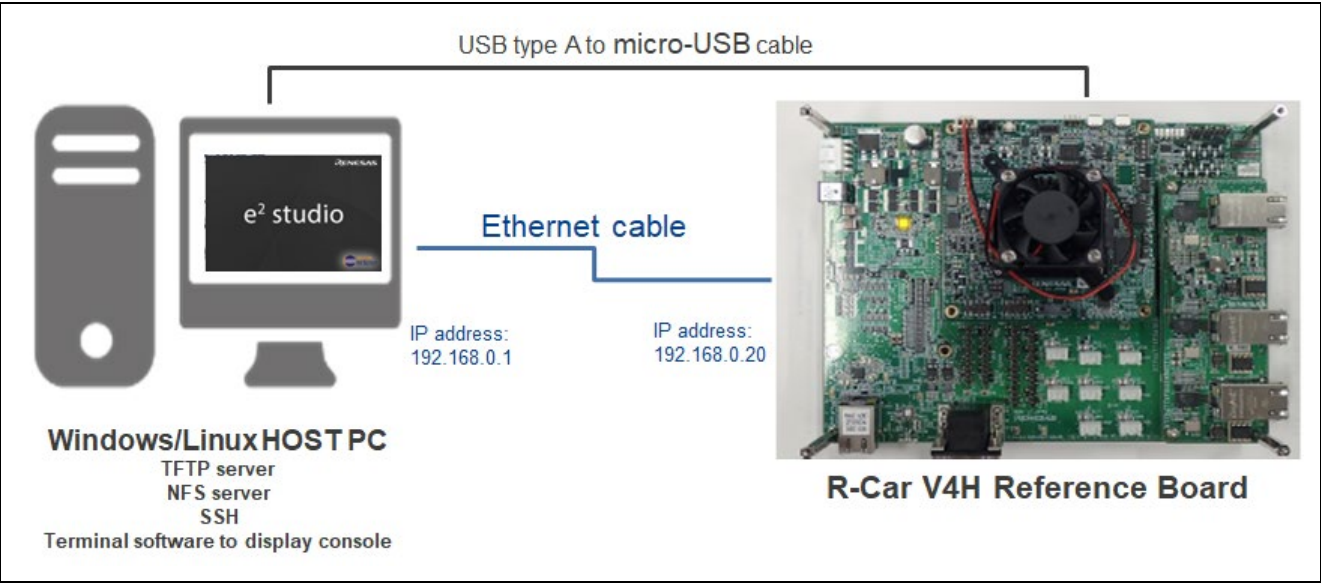

Figure 2-1   Connection between the Host PC and the Reference Board

3. Preparing Trace Data for Bus Monitoring

3.1 Starting Linux which Runs on an R-Car

Turn on the R-Car V4H and start Linux after reading an image for Linux running on an R-Car from the host PC.

The following shows the boot log messages in the serial console.

```
Using ethernet@e6800000 device

TFTP from server 192.168.0.1; our IP address is 192.168.0.20

Filename 'Image'.

Load address: 0x48080000

Loading: #################################################################

                :

       ##############################

       6 MiB/s

done

Bytes transferred = 33839616 (2045a00 hex)

Using ethernet@e6800000 device

TFTP from server 192.168.0.1; our IP address is 192.168.0.20

Filename 'r8a779g0-whitehawk.dtb'.

Load address: 0x48000000

Loading: #########

       5.5 MiB/s

Done
```

Figure 3-1   Preparing Data (1)

After the image for Linux running on the R-Car has been started, log in to Linux to confirm its version.

```
After images of Linux running on R-Car has been started, log in Linux to confirm its version.

[ OK ] Started Target Communication Framework agent.

[ OK ] Started Avahi mDNS/DNS-SD Stack.

[ OK ] Reached target Multi-User System.

       Starting Update UTMP about System Runlevel Changes...

[ OK ] Started Update UTMP about System Runlevel Changes.


Poky (Yocto Project Reference Distro) 3.1.11 v4x ttySC0

v4x login: root (Enter the password and press [Enter].)

root@v4x:~#

root@v4x:~# uname -a (Enter the command and press [Enter].)

Linux v4x 5.10.147-yocto-standard #1 SMP PREEMPT Fri Mar 3 11:25:12 UTC 2023 aarch64 aarch64 aarch64
GNU/Linux

root@v4x:~#
```

Figure 3-2   Preparing Data (2)

3.2 Preparing an Executable File for Acquiring Trace Data

When the e² studio is installed on the host PC, a pre-constructed stub file is stored in the DebugComp/RCar directory of the e² studio. The path name is given below.

"/home/<user name>/.eclipse/com.renesas.platform_XXXXXXXXX/DebugComp/RCar"

```
XXXX@XXXXXp:~$ ls -alF (Enter the command and press [Enter].)

total XXXXXX

drwxrwxr-x 9 XXXX XXXX    XXXX  MMM DD HH:MM ./

drwxrwxr-x 3 XXXX XXXX    XXXX  MMM DD HH:MM ../
                 :
-rw-rw-r-- 1 XXXX XXXX   572088  MMM DD HH:MM perfmodule2-5_10_0.ko

-rw-rw-r-- 1 XXXX XXXX   575584  MMM DD HH:MM perfmodule2-5_10_41.ko
                 :
-rw-rw-r-- 1 XXXX XXXX   494048  MMM DD HH:MM saperfmon.elf

-rw-rw-r-- 1 XXXX XXXX   630592  MMM DD HH:MM saperfmon_vh4.elf
                 :
XXXX@XXXXXp:~$
```

Figure 3-3   Preparing Data (3)

Select "perfmodule.ko" and "saperfmon.elf" corresponding to the version of Linux running on the R-Car and the target board. Store these files in the desired working directory of Linux.

Note that the following files are used in this environment.

- impsample_core_link (sample application for adding bus monitor [QOS] messages to the trace data)

- perfmodule.ko (file for operating an elf file)

- saperfmon.elf (executable file for acquiring trace data)

Figure 3-4   Preparing Data (4)

Store the files in the desired directory of Linux running on the R-Car and assign the executable right to the files to be executed.

```
root@v4x:~/XXXXXXXX# chmod 755 ./saperfmon.elf ./impsample_core_link (Enter the command and press [Enter].)

root@v4x:~/XXXXXXXX#

root@v4x:~/XXXXXXXX# ls -alF (Enter the command and press [Enter].)

total XXXX
                :
-rwxr-xr-x  1 root root   XXXXXX HHH DD  YYYY impsample_core_link*

-rw-r--r--  1 root root   XXXXXX HHH DD  YYYY perfmodule.ko

-rwxr-xr-x  1 root root   XXXXXX HHH DD  YYYY saperfmon.elf*
                :
root@v4x:~/XXXXXXXX#
```

Figure 3-5   Preparing Data (5)

Notes:

- The sample application (impsample_core_link) uses a sample file (rcar-xos/v3.12.0/samples/impsample_core_link/) included in the SDK.
- The perfmodule.ko and saperfmon.elf files are rebuilt after some source code has been changed with reference to "DTA_guide.pdf" in the support directory for performance-monitor (/home/<user name>/.eclipse/com.renesas.platform_XXXXXXXXXX/DebugComp/RCar/performance-monitor).

The performance-monitor directory is generated by starting the e^2 studio on the host PC, creating the R-Car project, and proceeding with debugging.

Select [File] -> [New] -> [Renesas C/C++ Project] -> [Renesas R-Car] to open the [New C/C++ Project] dialog box.
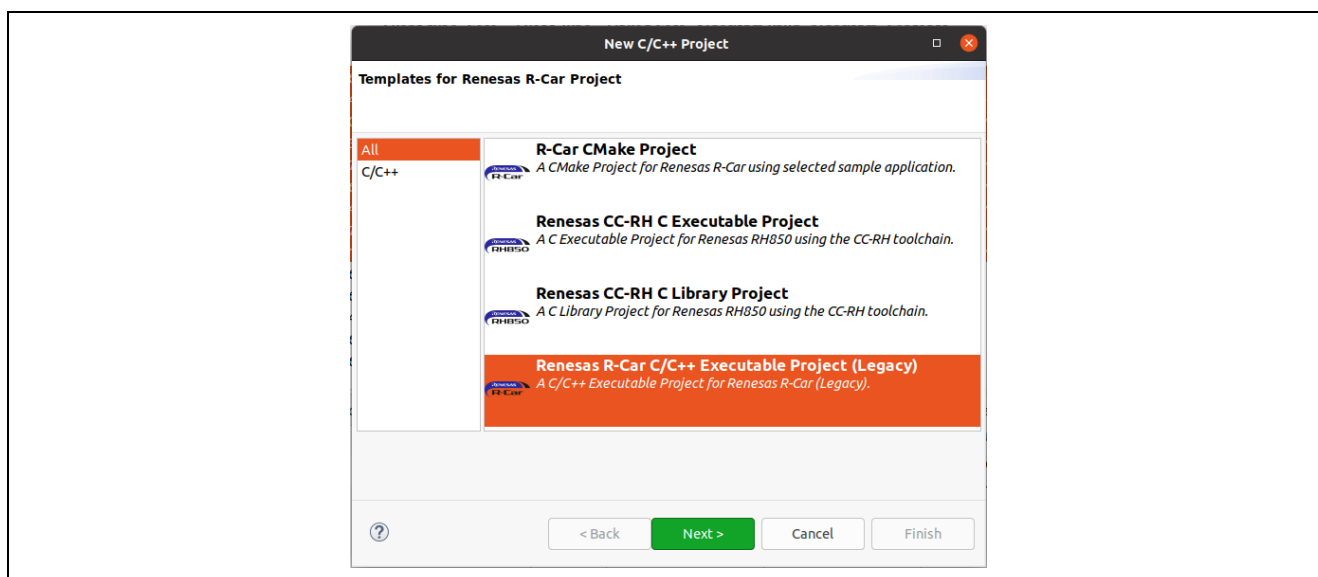

Figure 3-6   Preparing Data (6)

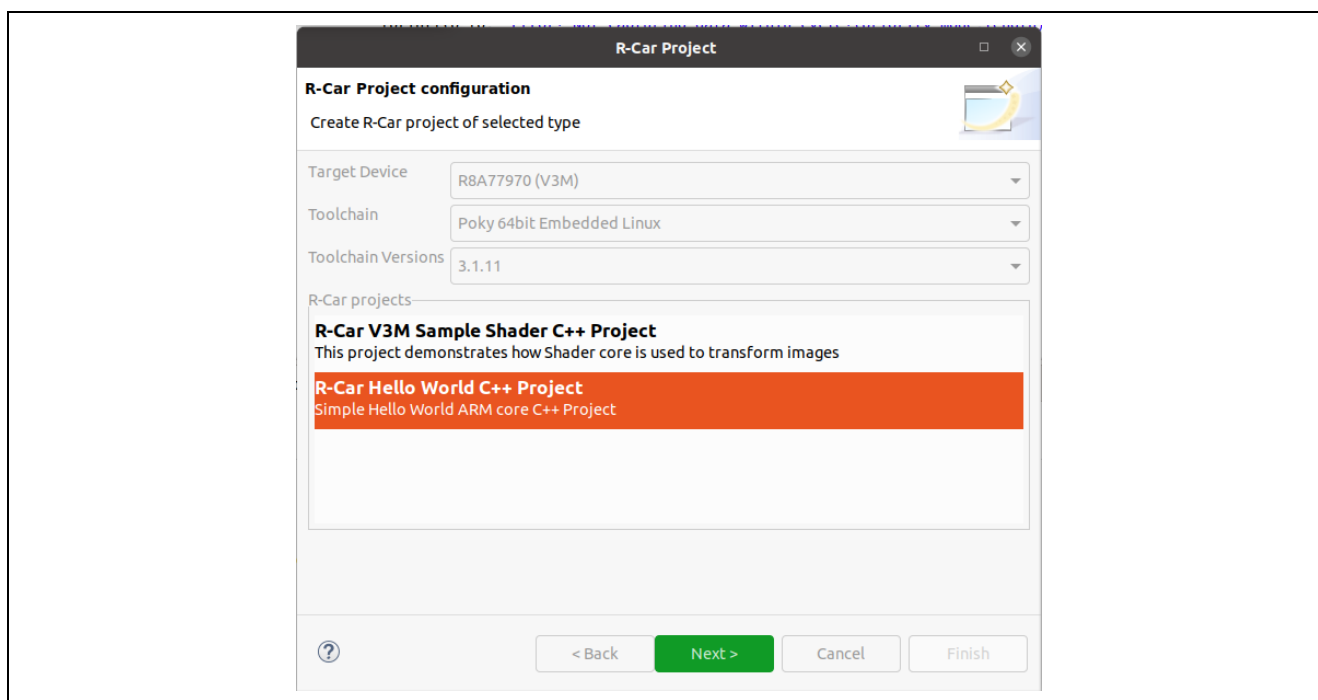Select "Hello World" as a simple project and click on the [Next] button.

Figure 3-7   Preparing Data (7)

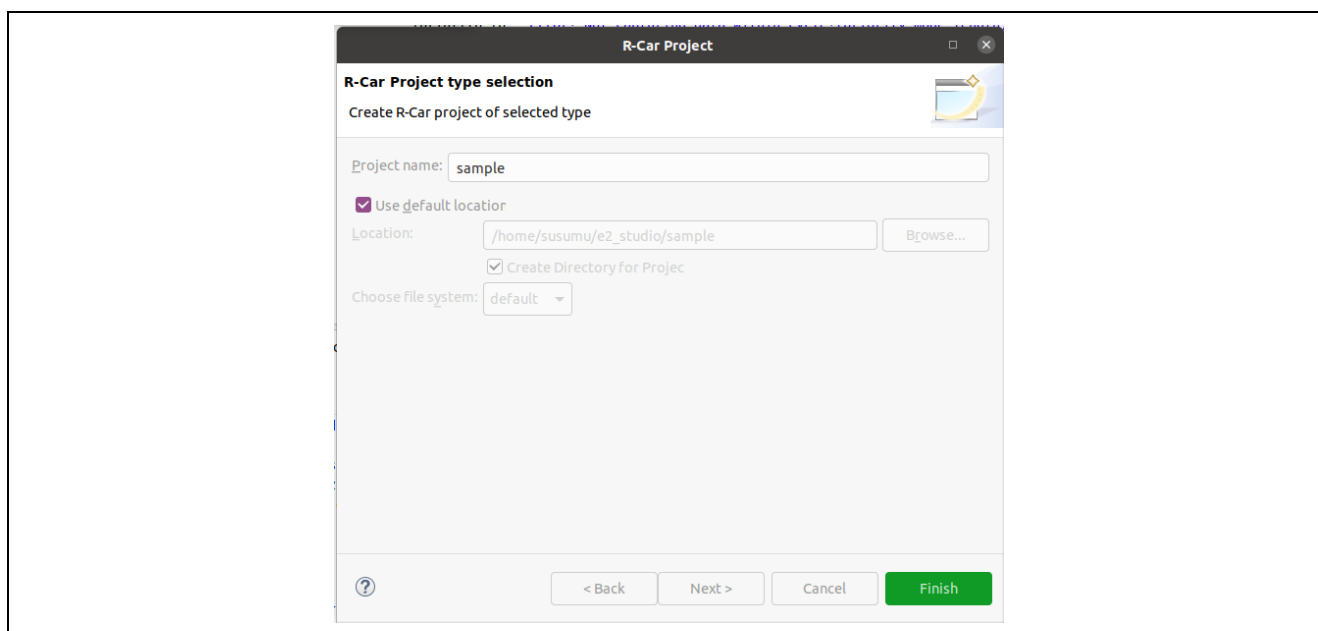Enter the desired project name and click on the [Finish] button.

Figure 3-8   Preparing Data (8)

When the created project is displayed in the list of projects, start debugging. Even if debugging fails to start, a support directory for performance-monitor will be generated under "~/DebugComp/RCar". Refer to "DTA_guide.pdf".

For the path to the support directory, open the [About e^2 studio for R-Car] dialog box and click on the [Installation Details] button by selecting the [About e^2 studio for R-Car] item from the [Help] menu.
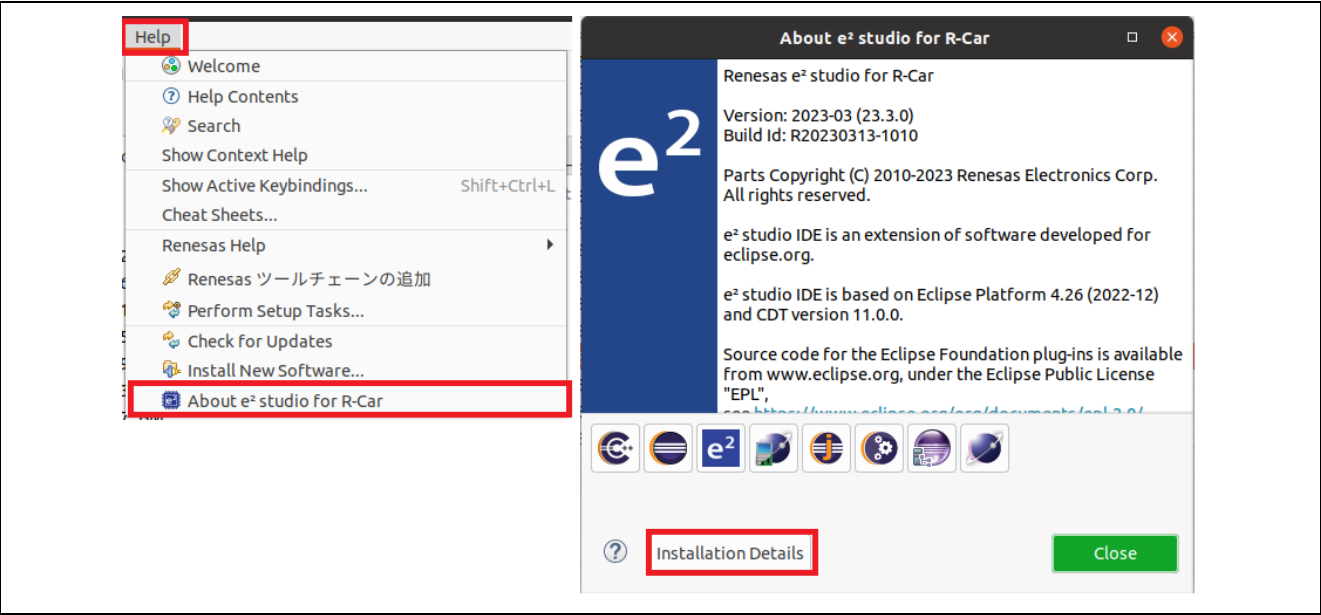

Figure 3-9   Preparing Data (9)

The [e^2 studio for R-Car Installation Details] window will open. Selecting the [Support Folders] tab enables confirmation of the path to the support directories.
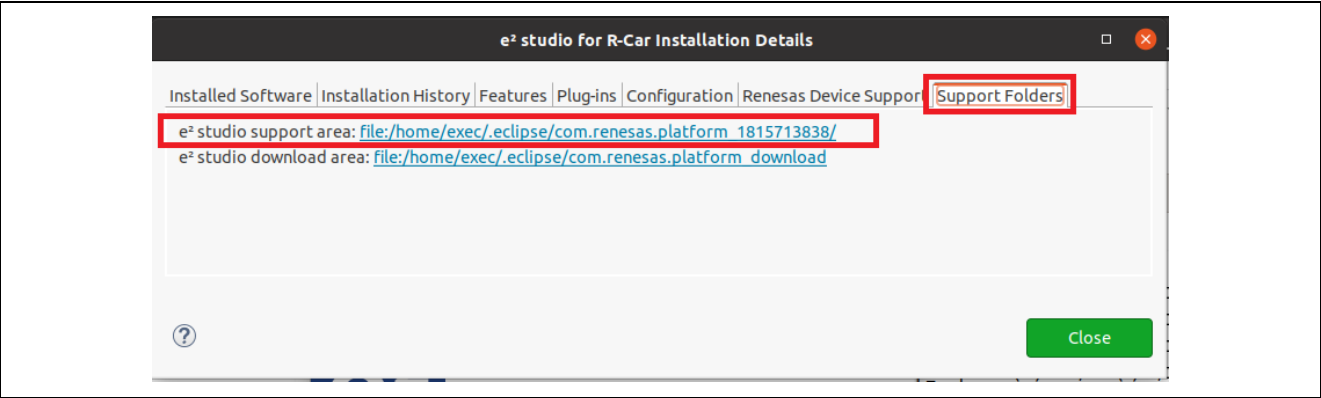

Figure 3-10   Preparing Data (10)

3.3   Starting Bus Monitoring

Start bus monitoring in the desired working directory of Linux running on the R-Car.

3.3.1   Loading a Module for Operating an elf File

Before starting bus monitoring, load a module. If an attempt is made to load a module that has already been loaded, an error message will be displayed.

*To unload a module, run the rmmod command.

```
root@v4x:~/XXXXXXXX# insmod ./perfmodule.ko (Enter the command and press [Enter].)

[ 711.531951] Perfmodule - Board Type V4H detected (0x5c12)

root@v4x:~/XXXXXXXX#

root@v4x:~/XXXXXXXX# insmod  ./perfmodule.ko (Enter the command and press [Enter].)

insmod: ERROR: could not insert module ./perfmodule.ko: File exists

root@v4x:~/XXXXXXXX#
```

Figure 3-11   Preparing Data (11)

3.3.2 Operating an elf File with Event Matching and QoS Monitor

When <-qevt> and <40, 41, 42, 43 (master of CCIO-4)> options are specified for the executable file for acquiring trace data (saperfmon.elf), the trace_YYYYMMDD_HHMMSS directory is created to start tracing. Then, a sample application (impsample_core_link) is executed.

```
root@v4x:~/XXXXXXXX# ./saperfmon.elf -qevt 40,41,42,43 & (Enter the command and press [Enter].)

[1] 499

root@v4x:~/e2studio_BusMon_3# Current working dir: /home/root/XXXXXXXX (desired directory name)

uio PMU driver 1 auto-detection: /dev/uio369

uio DBSC driver auto-detection: /dev/uio368

uio QOS driver auto-detection: /dev/uio367

Trace directory: /home/root/XXXXXXXX/trace_YYYYMMDD_HHMMSS


QOS Flat Event Summary
======================
0 - Merge_int(W)

1 - Secure_engine(W)
              :
V4H config end, reading thread started.

Time translation sent back.

Time translation sent back.

Initial sent back.


root@v4x:~/XXXXXXXX# ./impsample_core_link (Enter the command and press [Enter].)

<IMP Framework Demo start>


RCAR_SOC      V4H

ATMLIB Version 2.20.0

IMPFW Version 3.12.0

Execute IMP with IMP core and PSC core

-- OutPutMemory Image data of SPM --

00 00 00 00 04 00 00 00 08 00 00 00 0C 00 00 00
        :
F0 10 F4 10 F8 10 FC 10

[RESULT] PASS

<IMP Framework Demo end>


impsample_core_link return = 0 (OK:0, NG:else)

root@v4x:~/XXXXXXXX#
```

Figure 3-12   Preparing Data (12)

Notes:
- You can confirm that the bus monitor message has been added in trace.txt within the trace_YYYYMMDD_HHMMSS directory.

```
                    :
QOS: CCI0(W):0x0000001E CCI0(R):0x0000002E CCI1(W):0x00000000 CCI1(R):0x00000000, EpochTS: 1677857377620530097

QOS: CCI0(W):0x0000000F CCI0(R):0x0000002E CCI1(W):0x00000000 CCI1(R):0x00000000, EpochTS: 1677857377621569721

QOS: CCI0(W):0x0000000F CCI0(R):0x0000008A CCI1(W):0x00000000 CCI1(R):0x00000000, EpochTS: 1677857377622613065

QOS: CCI0(W):0x0000000F CCI0(R):0x0000000F CCI1(W):0x00000000 CCI1(R):0x00000000, EpochTS: 1677857377623650409

QOS: CCI0(W):0x00000740 CCI0(R):0x000000A6 CCI1(W):0x00000314 CCI1(R):0x000029EF, EpochTS: 1677857377624711213

QOS: CCI0(W):0x0000002E CCI0(R):0x000000A9 CCI1(W):0x0000002E CCI1(R):0x00000000, EpochTS: 1677857377625748678

QOS: CCI0(W):0x0000003D CCI0(R):0x0000002E CCI1(W):0x0000001E CCI1(R):0x00000000, EpochTS: 1677857377626786082

QOS: CCI0(W):0x0000005C CCI0(R):0x000005B1 CCI1(W):0x0000000F CCI1(R):0x00000000, EpochTS: 1677857377627829966

QOS: CCI0(W):0x000000A8 CCI0(R):0x000000D6 CCI1(W):0x00000000 CCI1(R):0x00000000, EpochTS: 1677857377628874870

QOS: CCI0(W):0x0000007A CCI0(R):0x0000002D CCI1(W):0x0000000F CCI1(R):0x00000000, EpochTS: 1677857377629919054
                    :
```

Figure 3-13   Preparing Data (13)

- You can confirm that the stream_XXX file has been added within the trace_YYYYMMDD_HHMMSS directory.

```
root@v4x:~/XXXXXXXX/trace_YYYYMMDD_HHMMSS# ls -alF (Enter the command and press [Enter].)
                    :
-rw-r--r-- 1 root root   XXX MMM DD  YYYY stream_250
                    :
root@v4x:~/XXXXXXXX/trace_YYYYMMDD_HHMMSS#
```

Figure 3-14   Preparing Data (14)

- If you execute a sample application (impsample_core_link) after you have executed "saperfmon.elf", an error may occur. In such cases, repeat the execution of the application about ten times.

```
root@v4x:~/XXXXXXXX# ./impsample_core_link (Enter the command and press [Enter].)
        :
[RESULT] FAIL
<IMP Framework Demo end>
<< [ERROR] R_OSAL_Deinitialize : code 1024>>
impsample_core_link return = -1 (OK:0, NG:else)
root@v4x:~/XXXXXXXX# ./impsample_core_link (Enter the command and press [Enter].)
        :
<< [ERROR] R_OSAL_Deinitialize : code 1024>>
root@v4x:~/XXXXXXXX# ./impsample_core_link (Enter the command and press [Enter].)
        :
<< [ERROR] R_OSAL_Deinitialize : code 1024>>
root@v4x:~/XXXXXXXX# ./impsample_core_link (Enter the command and press [Enter].)
        :
[RESULT] PASS
<IMP Framework Demo end>
impsample_core_link return = 0 (OK:0, NG:else)
root@v4x:~/XXXXXXXX#
```

Figure 3-15   Preparing Data (15)

4. Operations for Debugging

4.1 Creating a "Tracing" Project

Start the e² studio on the host PC and create a "Tracing" project.

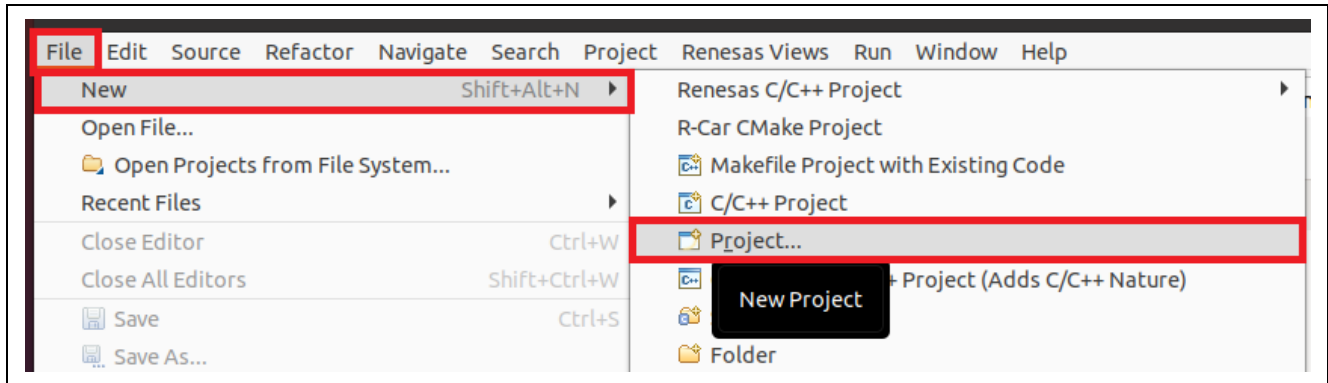Select [File] -> [New] -> [Project…] to open the [New Project] dialog box.

Figure 4-1   Operation for Debugging (1)

Select [Tracing Project] in the [New Project] dialog box and click on the [Next] button.

The [Tracing Project] dialog box appears. Enter the desired project name and click on the [Finish] button.
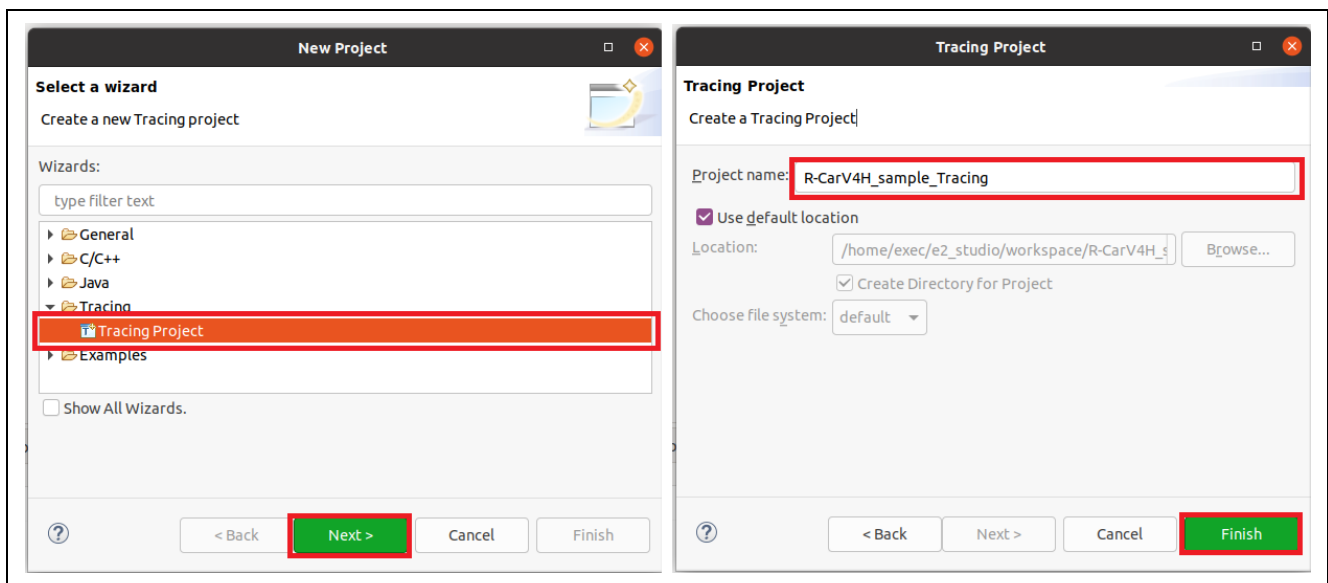

Figure 4-2   Operation for Debugging (2)

4.2   Setting Trace Profiling

Select "Fetch Remote Traces…" from the trace project with the desired name to open the [Fetch Remote Traces] dialog box.
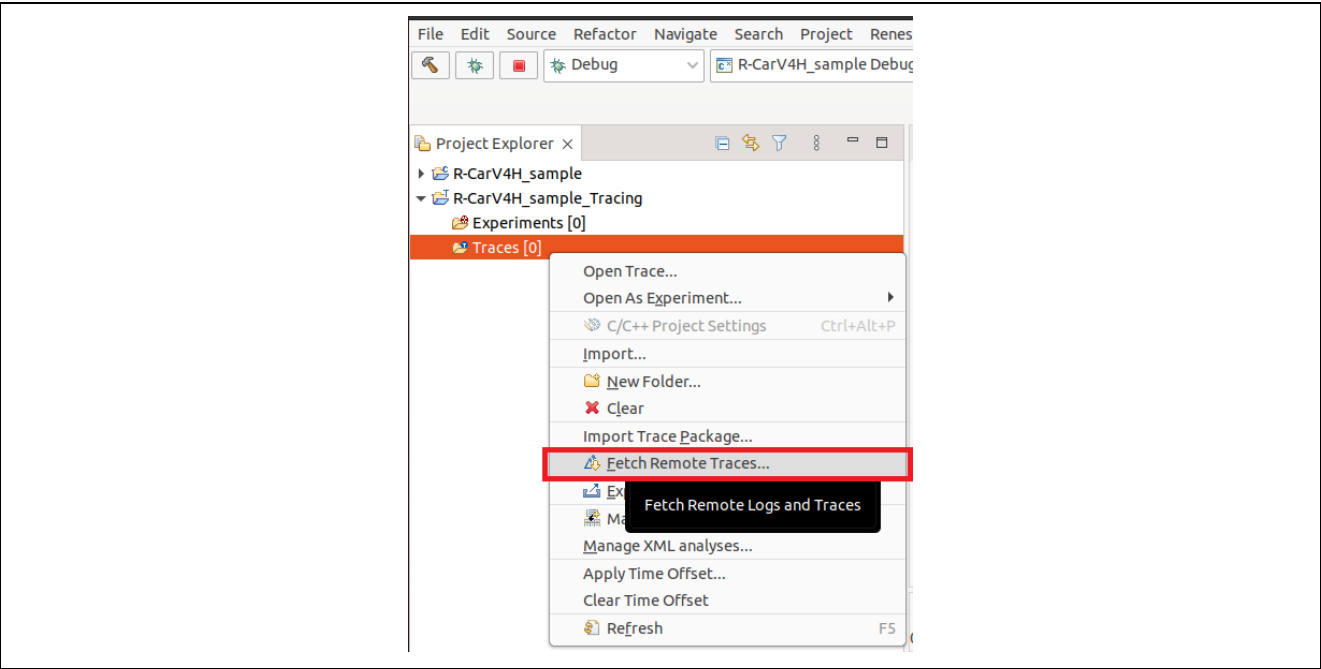

Figure 4-3   Operation for Debugging (3)

Click on the [Manage Profiles] button to open the [Preferences (Filtered)] dialog box.
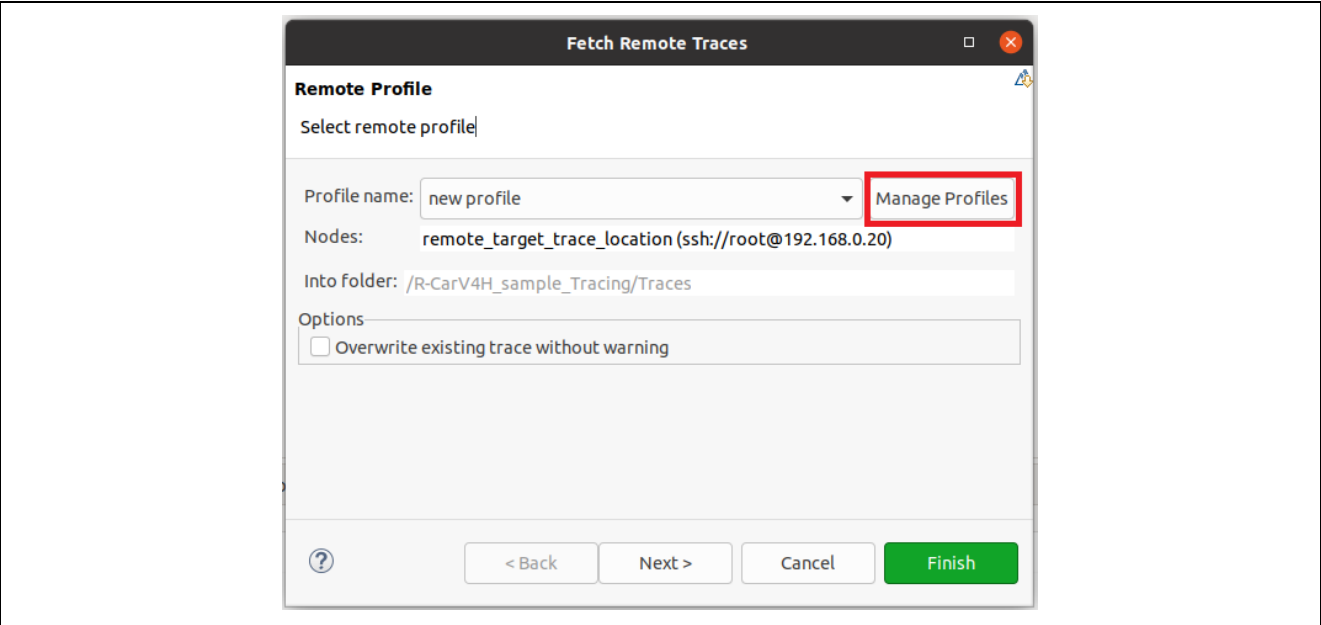

Figure 4-4   Operation for Debugging (4)

RENESAS

In the [Preferences (Filtered)] dialog box, set the trace profiling for the SSH connection as shown in the figure below.

- URI: Specify the root user name and IP address of the target (R-Car V4H reference board).
- Root path: Specify the desired directory of Linux running on the R-Car for use in remote tracing.
- Recursive: Select the checkbox.
- Trace type: Select the "R-Car Performance Trace" format.

Click on the [Apply and Close] button in the [Preferences (Filtered)] dialog box to return to the [Fetch Remote Traces] dialog box.
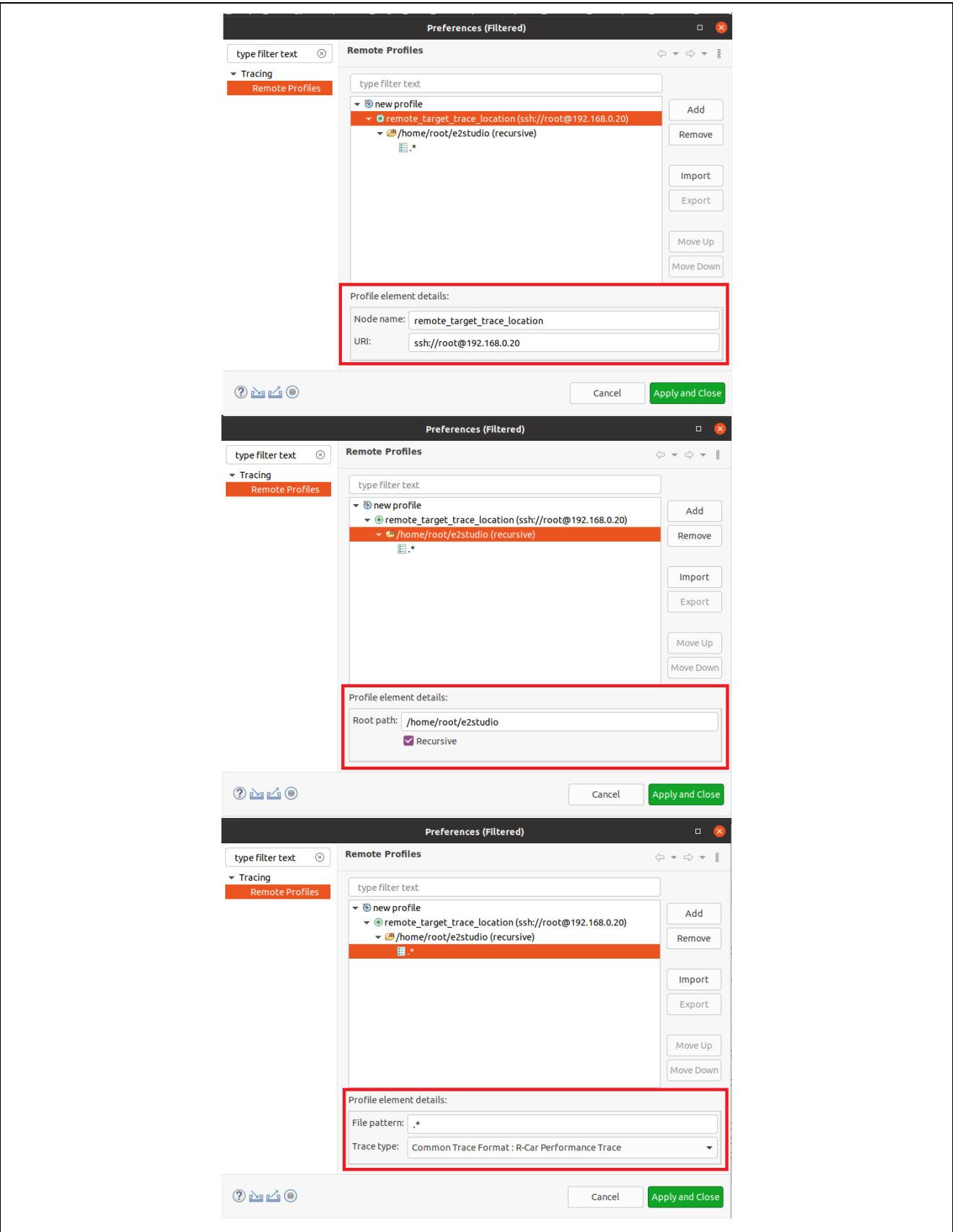
Figure 4-5   Operation for Debugging (5)

4.3   Importing Trace Data

Click on the [Next] button in the [Fetch Remote Traces] dialog box.
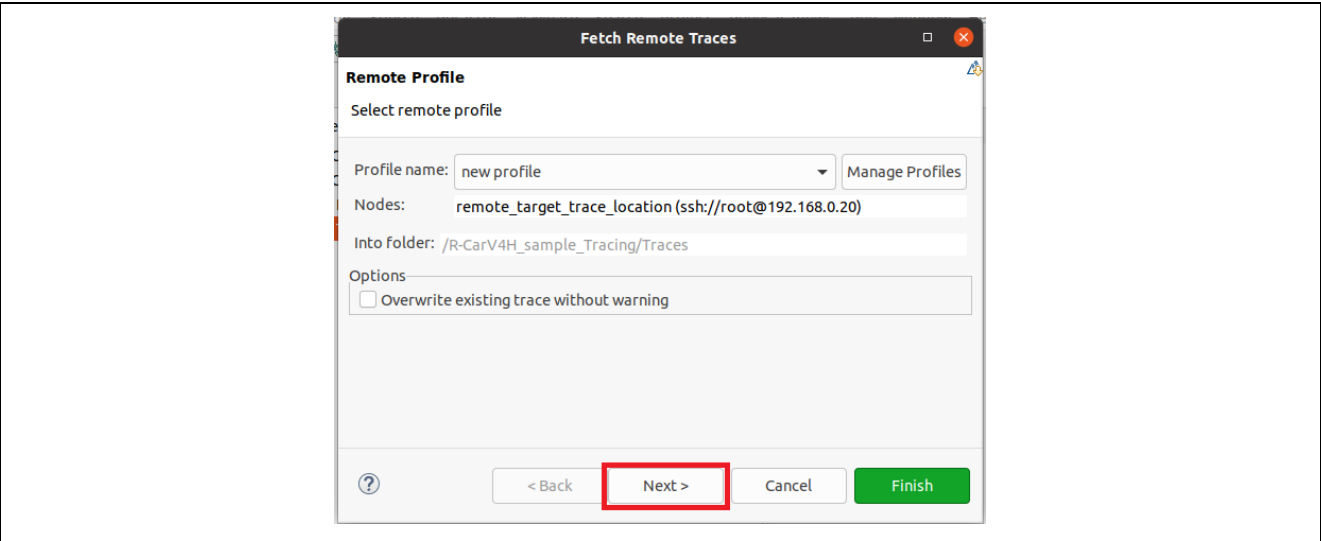

Figure 4-6   Operation for Debugging (6)

Select the required trace type and import all items except for the text log. After that, click on the [Finish]
button.
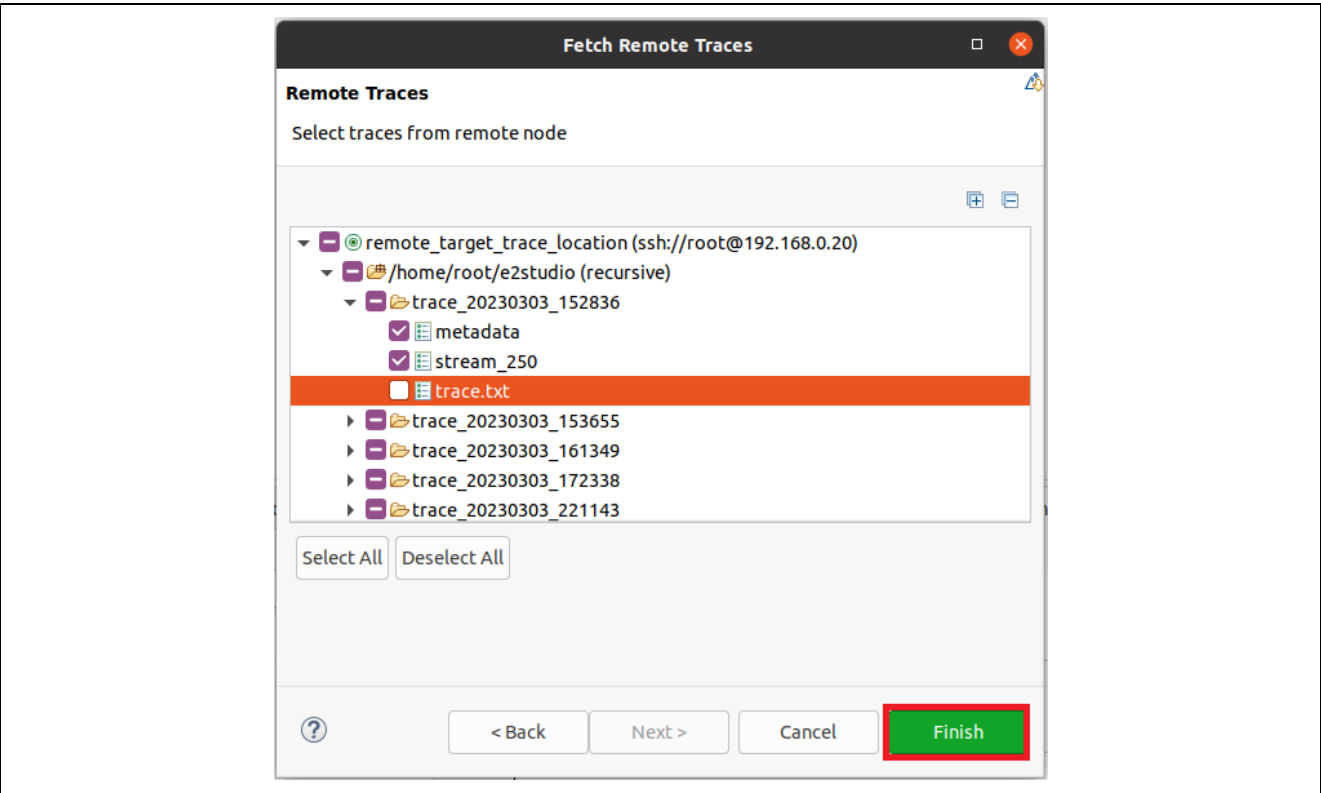
Figure 4-7   Operation for Debugging (7)

4.4 Displaying Trace Data

A trace directory is displayed in the trace project. Right-click on the name of the trace directory and specify "R-Car Performance Trace" as the trace type.
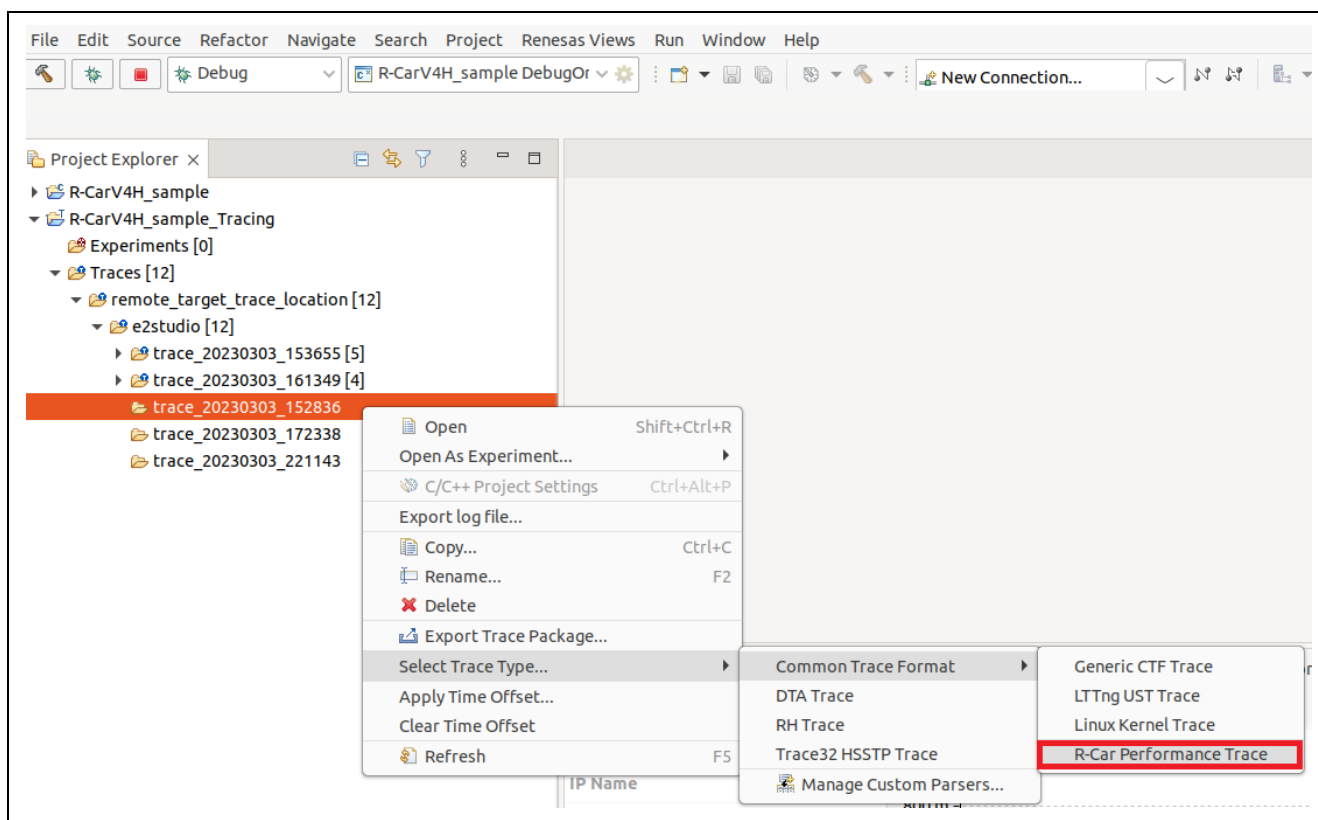

Figure 4-8   Operation for Debugging (8)

Double-clicking on the "trace_YYYYMMDD_HHMMSS" directory in the trace project displays a list of tracing operations.
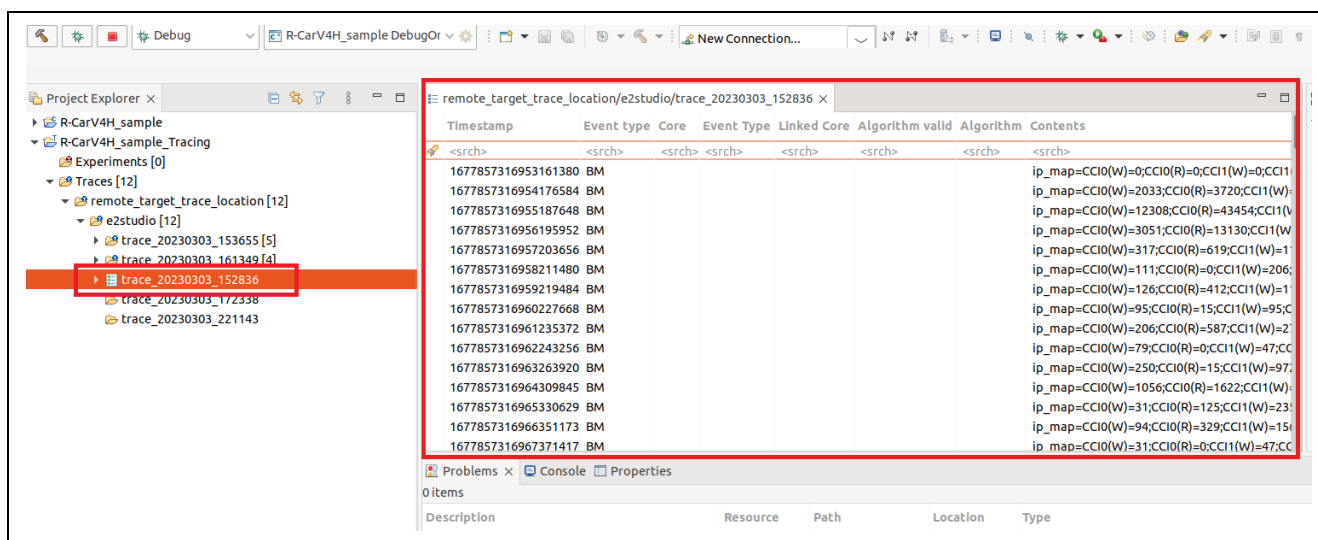
Figure 4-9   Operation for Debugging (9)

Double-clicking on "System Performance – Bus Monitor (QOS trace)" in the trace project opens the view for bus monitoring.

Enable checkboxes for [CCI0(R)], [CCI0(W)], [CCI1(R)], and [CCI1(W)] in the [IP Name] column. The trace data in time selected from the list of tracing operations are displayed in the view.
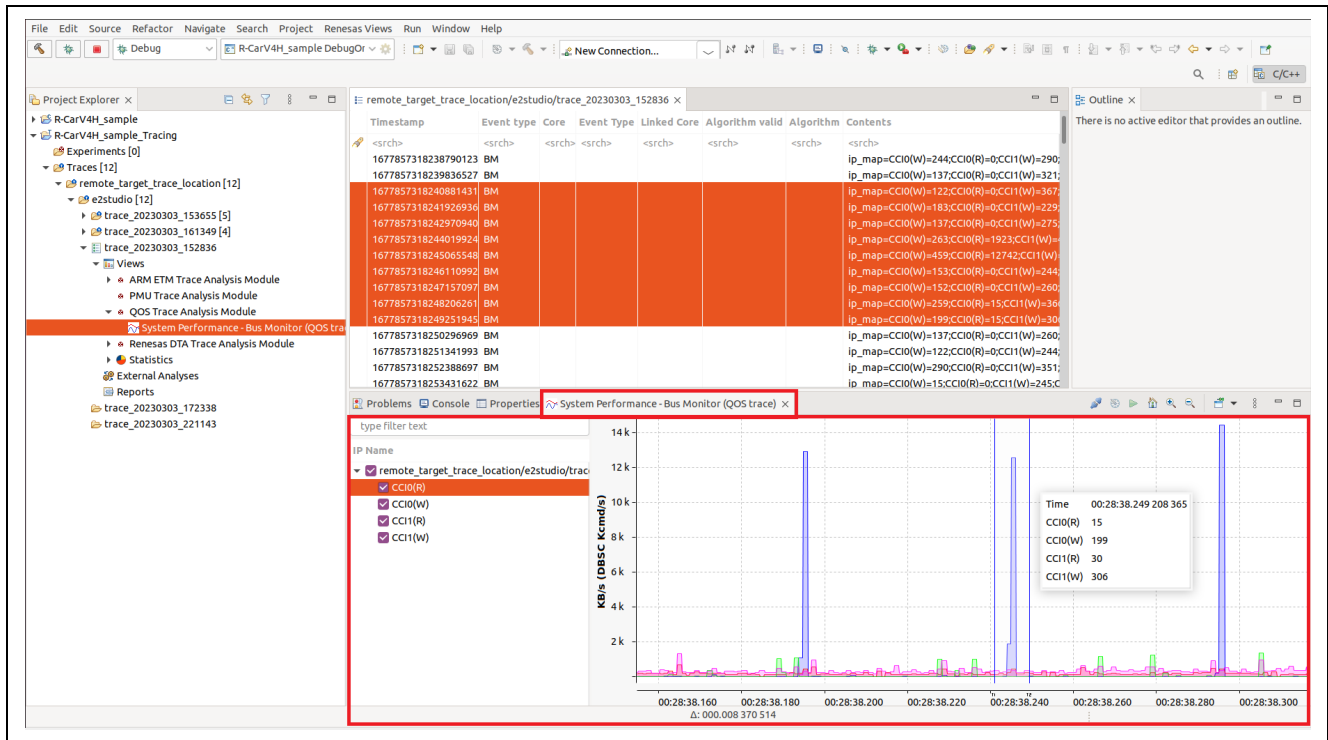
Figure 4-10　Operation for Debugging (10)

The [Show View] dialog box is opened by selecting [Window] -> [Show View] -> [Other…]. Selecting [Renesas R-Car Trace] -> [System Performance – Bus Monitor (QOS trace)] and clicking on the [Open] button also opens the view for bus monitoring.
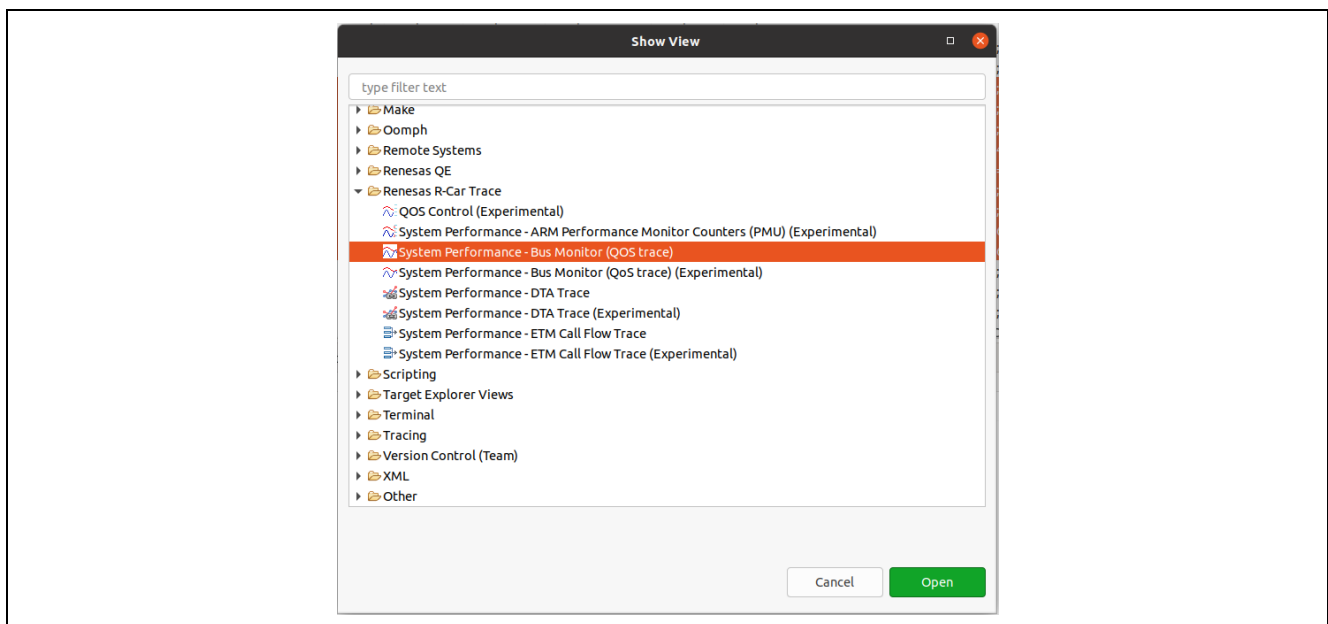
Figure 4-11　Operation for Debugging (11)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.31.23	-	First Edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(Rev.5.0-1   October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

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