Renesas Synergy™ Platform

Access to External Flash Memory in Renesas Synergy™ Development Environments

Introduction

This document uses sample projects to describe the procedures for downloading data to the external flash memory and erasing and programming the external flash memory through the execution of a program at debugger start-up. It shows the procedure using the development kit, DK-S7G2 with the Renesas Synergy™ e² studio Integrated Solution Development Environment (hereafter referred to as the e² studio) or the IAR Embedded Workbench® for Renesas Synergy™ (hereafter referred to as the IAR EW for Synergy). These are the standard Renesas Synergy™ development environments.
For documents related to the Renesas Synergy™ development environment, see Renesas Electronics Synergy website (www.renesas.com/synergy/gallery), and select **Support** > **Documentation,** or go to the Renesas Electronics Synergy website documentation (www.renesas.com/synergy/docs).

Environment

Operation was confirmed in the following environments.

- e² studio ISDE v7.3.0 or later
- Synergy Software Package (SSP) v1.6.0 or later
- IAR EW for Synergy v8.23.3 or later
- Renesas Synergy™ Standalone Configurator (SSC) v7.3.0 or later
- SK-S7G2 or DK-S7G2 v3.1(only).
  Note: This application is not fully supported on DK-S7G2 v4.1 board.

Contents

1. Introduction

This document describes the procedures for downloading data to the QSPI flash memory mounted as external flash memory on the DK-S7G2 board and erasing and programming the external flash memory through the execution of a program.

Figure 1 is an overview of the memory map. The ‘Onboard flash area’ in the figure shows the area for the external flash memory on the DK-S7G2 board.

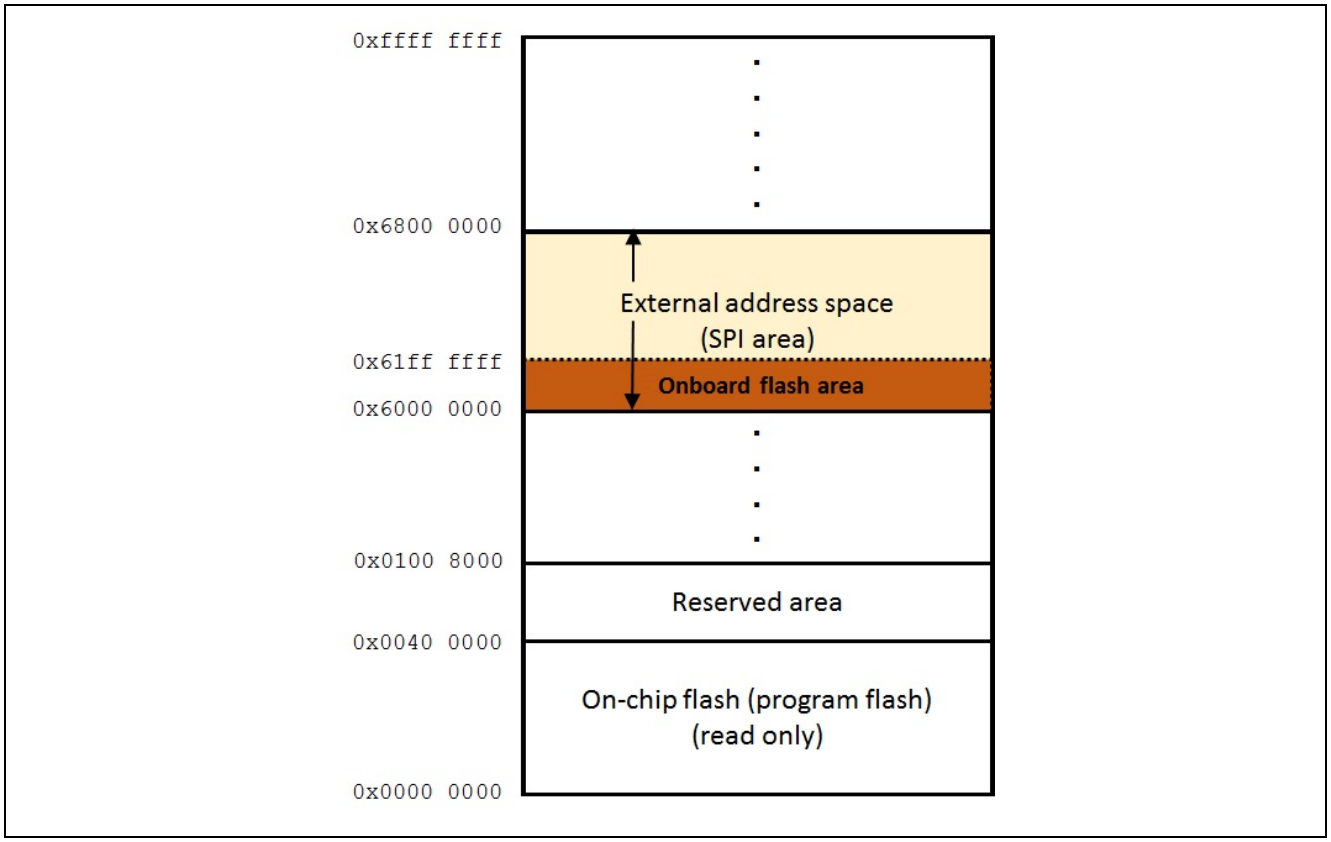

Figure 1. Memory Map (Overview)

2. Preparation

This chapter describes how to get the QSPI sample project, modify the linker script file, and set a DIP switch.

2.1 Getting the QSPI Sample Project

The QSPI sample project, which is required for downloading data to the external flash memory and for checking the erasure and programming of the external flash memory through the execution of a program, is included among the compressed files obtained by downloading this application note.

2.2 Linker Script File

The linker script file defines the external flash memory area and section name.

In the linker script file, a 64-MB area is defined as the external flash memory. However, the external flash memory on the DK-S7G2 board has 32 MB. Modify the setting for the area in the way that suits the given development environment.

2.2.1 Modifying the Linker Script File (s7g2.ld) for the e² studio

The linker script file is stored in the following directory:

```
project\script\
```

Modify the underlined sections as shown below.

```
QSPI_FLASH (rx) : ORIGIN = 0x60000000, LENGTH = 0x4000000 /* 64M, Change in
QSPI section below also */
```

Modify as follows:

```
QSPI_FLASH (rx) : ORIGIN = 0x60000000, LENGTH = 0x2000000 /* 32M, Change in
QSPI section below also */

__qspi_region_max_size__ = 0x4000000; /* Must be the same as defined in MEMORY
above */
```

Modify as follows:

```
__qspi_region_max_size__ = 0x2000000; /* Must be the same as defined in MEMORY
above */
```

Store the modified linker script file in the original directory.

Note: Save the original linker script file with a different name before overwriting it.
In this description, the linker script file is modified to be used with the DK-S7G2 board. If you are using another board, modify the linker script file to match the amount of external flash memory on that board.

2.2.2 Modifying the Linker Script File (s7g2.icf) for IAR EW for Synergy

The linker script file is stored in the following directory:

```
QSPI sample project\script
```

Modify the underlined section as shown below:

```
define symbol region_QSPI_end    = 0x63FFFFFF;
```

Modify as follows:

```
define symbol region_QSPI_end    = 0x61FFFFFF;
```

Store the modified linker script file in the original directory.

Note: Save the original linker script file with a different name before overwriting it.
In this description, the linker script file is modified to be used with the DK-S7G2 board. If you are using another board, modify the linker script file to match the amount of external flash memory on that board.

Note: For DK-S7G2 v4.1 board, make the following changes in case you come across a download error. You can also use the backup file present in the project root directory.

Modify the underlined section as shown below:

```
place in QSPI_region    { block QSPI_NON_RETENTIVE_BLOCK,
section .qspi_flash  };
```

Modify as follows:

```
place in QSPI_region    { block QSPI_NON_RETENTIVE_BLOCK,
rw section .qspi_flash  };
```

2.3 DIP Switch Setting

The external flash memory on the DK-S7G2 board is enabled or disabled through a DIP switch setting. To use the downloading function described in this document, turn on **switch 2** (QSPI) on DIP switch block S5.

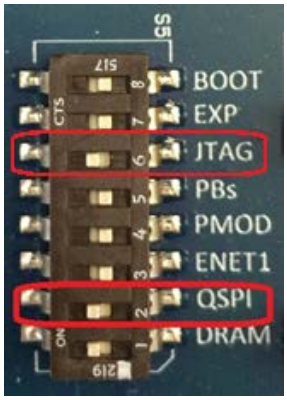

Figure 2. DIP Switch Setting

3. QSPI Sample Project

The QSPI sample project in the compressed files can be used with both the e^2 studio and IAR EW for Synergy.

3.1 Overview of the QSPI Sample Project

This section describes the memory map of this sample project.

The `blinky()` function is allocated to the range starting at 0x60000000 in the external flash memory area (this function causes LED1 and LED2 on the main board to blink on and off at one-second intervals).

The area from 0x60010000 to 0x60017fff is used for erasure and programming of the external flash memory through execution of the program.

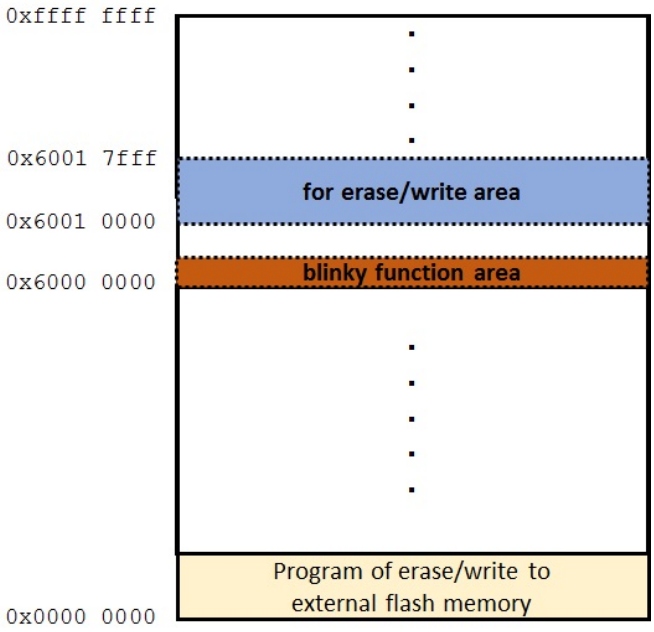

Figure 3. Memory Map of the Project

3.2 Operation of the QSPI Sample Project

Operations of this sample project are in the following order. The results are displayed on the debugger by using the semi-hosting[1] function.

1. Data in the areas from 0x60000000 to 0x60000010 and from 0x60010000 to 0x60010010 are displayed in the debugger to confirm the downloading of data to the external flash memory when the debugger starts up.
2. Data in the areas from 0x60000000 to 0x60000010 and from 0x60010000 to 0x60010010 are again displayed in the debugger to confirm erasing of the sector in the area from 0x6001 0000 to 0x6001 7fff through execution of the program.
3. Data in the areas from 0x60000000 to 0x60000010 and from 0x60010000 to 0x60010010 are again displayed in the debugger to confirm the writing of 0x9999, 0x8888, 0x7777, 0x6666, and 0x5555 to the area from 0x60010000 to 0x60010010 through execution of the program.
4. The `Blinky()` function that has been downloaded to the external flash memory is executed.

3.3 Starting the QSPI Sample Project in the e² studio

The QSPI sample project for the e² studio must be imported to be used. This section describes the procedure for importing the QSPI sample project.

To skip the development walkthrough in this document and open a completed project in e² studio, refer to *Importing a Renesas Synergy Project* (`r11an0023eu0121-synergy-ssp-import-guide.pdf`) for instructions on importing the project into e² studio and building the project. The included `Simple_QSPI_Example.zip` file contains the completed project.

1. Click on the **Debugger** tab. Select **J-Link ARM** for **Debug hardware** and **R7FS7G2** for **Target Device** and click on the **Debug** button.

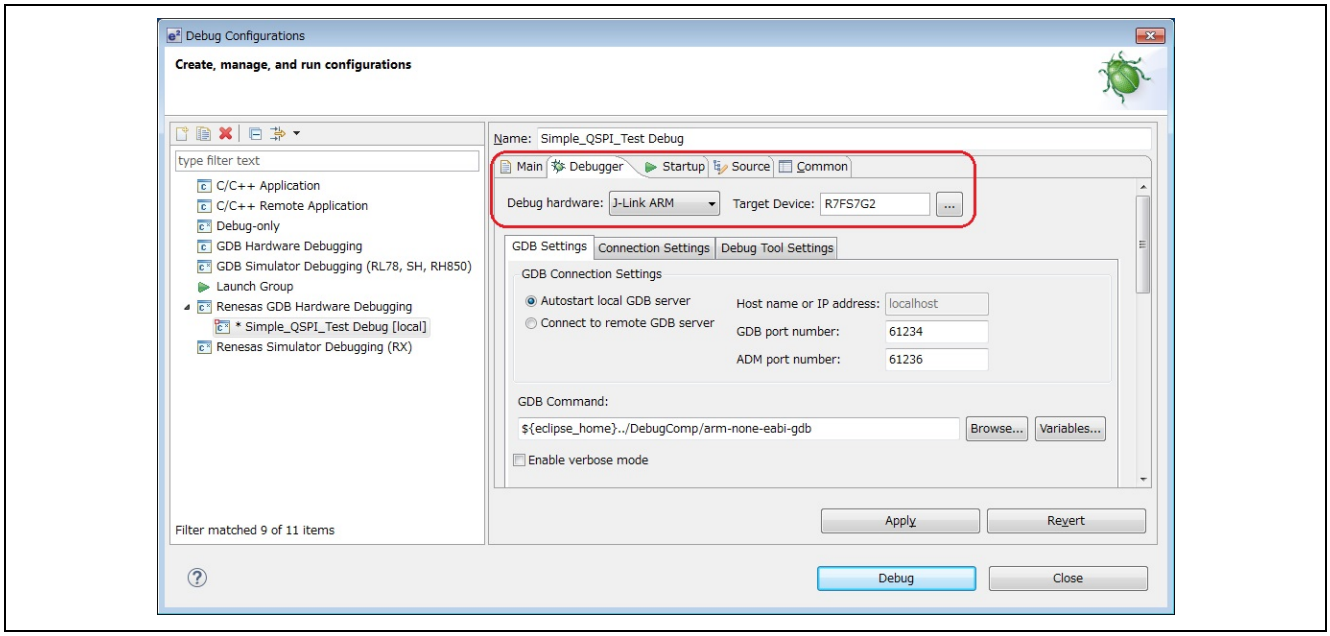

Figure 4. Specifying the Debugger

[1] The semi-hosting function used by this sample project displays the result of the `printf()` function on a debugger.

2. After the debugger starts up, open `blinky.c` and confirm that the `blinky()` function has been allocated to the external flash memory area.

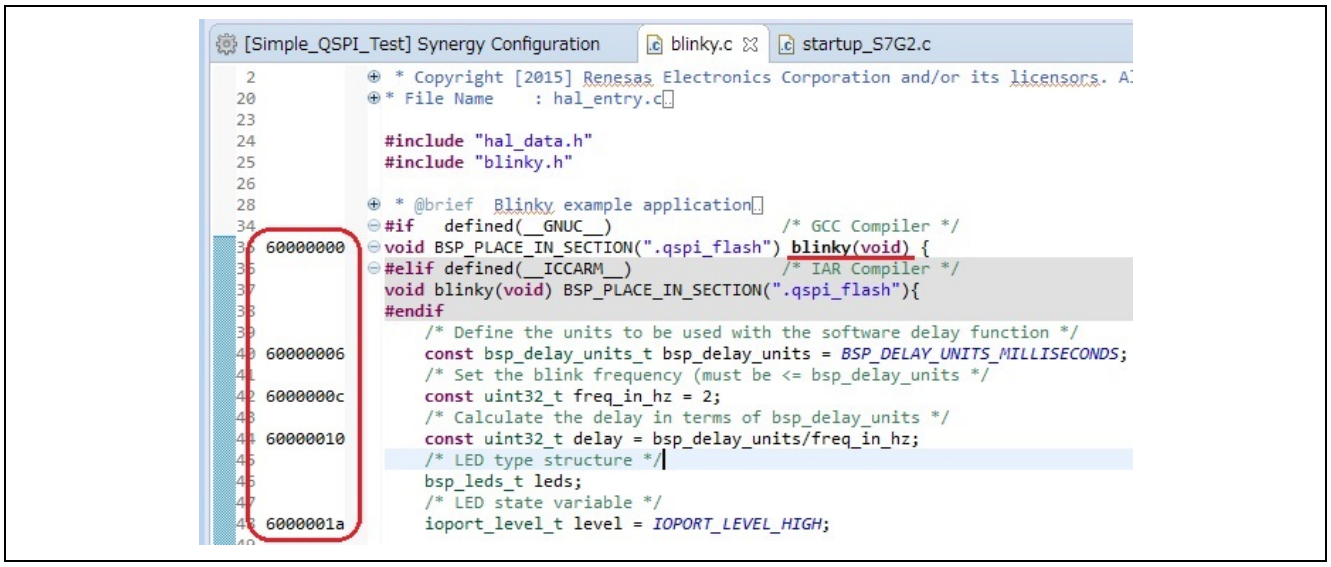

Figure 5. Allocation of the Blinky() Function (in the e² studio)

3. Check the data in address 0x60010000 in the **Memory** view. For details on using the **Memory** view, refer to section 3.3.2.

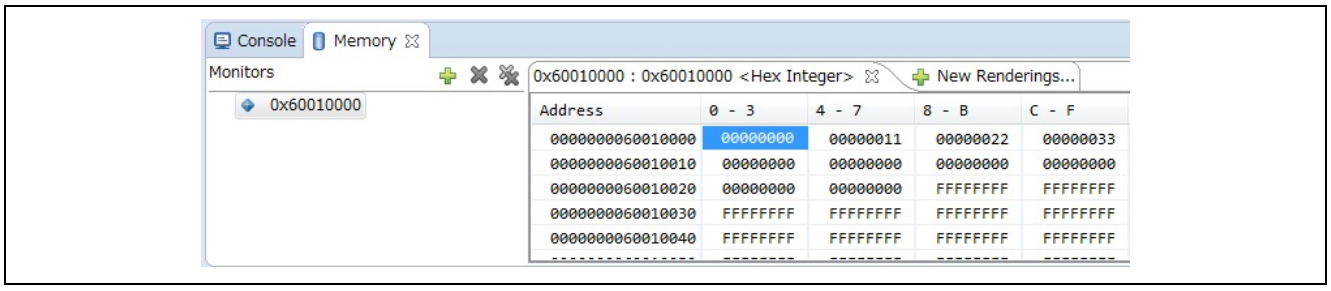

Figure 6. Displaying Data from the External Flash Memory in the Memory View

4. Set a breakpoint at the point where the `blinky()` function is called from `hal_entry.c`.

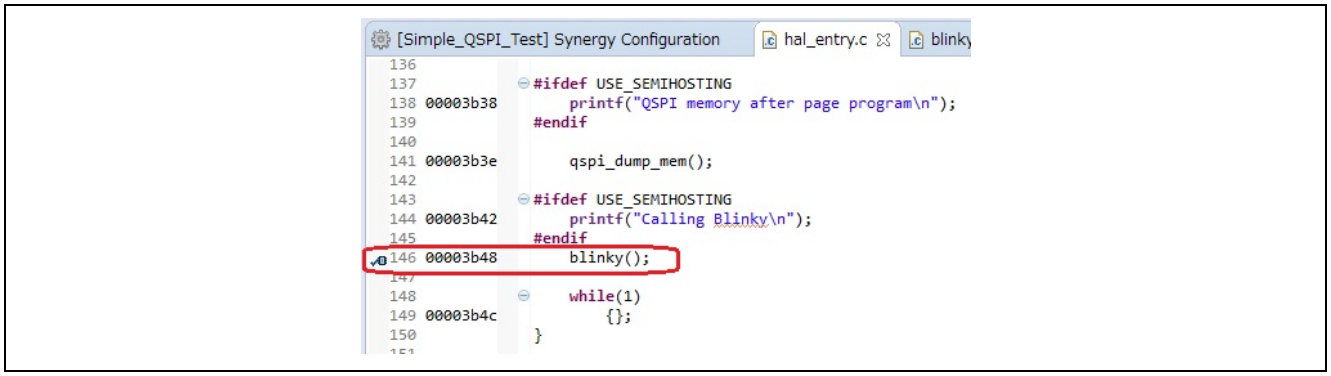

Figure 7. Setting a Breakpoint

5. When the program is executed, it stops at the breakpoint that has been set in step 6. Check erasure and programming of the external flash memory. For the results to expect from erasure and programming of the external flash memory, see section 3.3.2.
6. When the program is made to start running again, the `blinky()` function downloaded to the external flash memory area is executed to cause LED1 and LED2 on the board to blink on and off at one-second intervals.

3.3.1 Checking Data in the External Flash Memory

Use the **Memory** view to check the data in the external flash memory. This section describes the procedure for checking the data in the external flash memory by using the **Memory** view.

1. Select the **Window** > **Show View** > **Memory** or **Window** > **Show View** > **Other…** > **Debug** > **Memory** menu item.

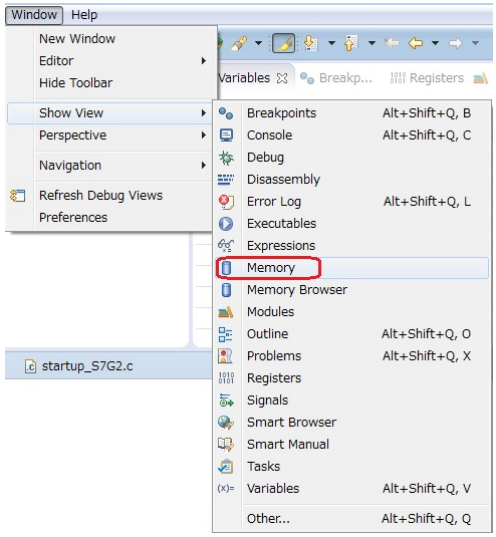

Figure 8. Setting the Display of the Memory View

2. Clicking on the **+** button shows the **Monitor Memory** dialog box. Enter **0x60010000** and check the displayed data from memory.

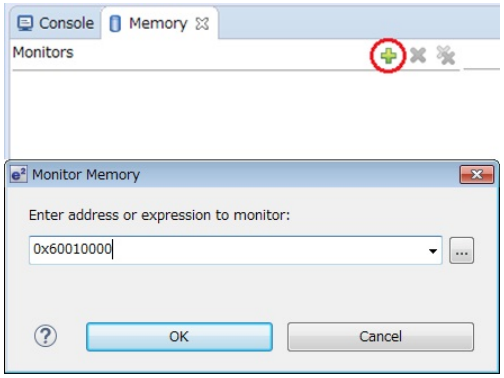

Figure 9. Setting the Memory View

Data has been written to address 0x60010000 as shown in Figure 10.

Figure 10. Checking the Data in the External Flash Memory

Note: The SEGGER JTAG debugger normally caches the memory area that corresponds to the external QSPI flash device. This caching improves performance by also preventing real-time changes to the QSPI flash device from being visible in this memory window. To view changes to this memory in real-time, we added the following two lines to the `Simple_QSPI_Example Debug.jlink` file in the source project:

```
[FLASH]
CacheExcludeSize = 0x800000
CacheExcludeAddr = 0x60010000
```

3.3.2 Checking Erasure and Programming of the External Flash Memory

The code for the erasure and programming of the external flash memory is in the `hal_entry()` function, which is in the `hal_entry.c` source file. The semi-hosting function is used to display the results in the **Renesas Debug Virtual Console** view of the debugger. To show the **Renesas Debug Virtual Console** view, select **Renesas Debug Virtual Console** by clicking on the button to open a console while the **Console** view is being displayed.

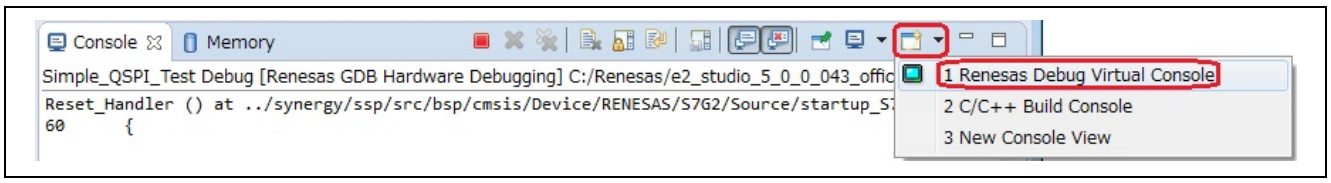

Figure 11. Showing the Renesas Debug Virtual Console View

The results of execution are displayed as shown in Figure 12.

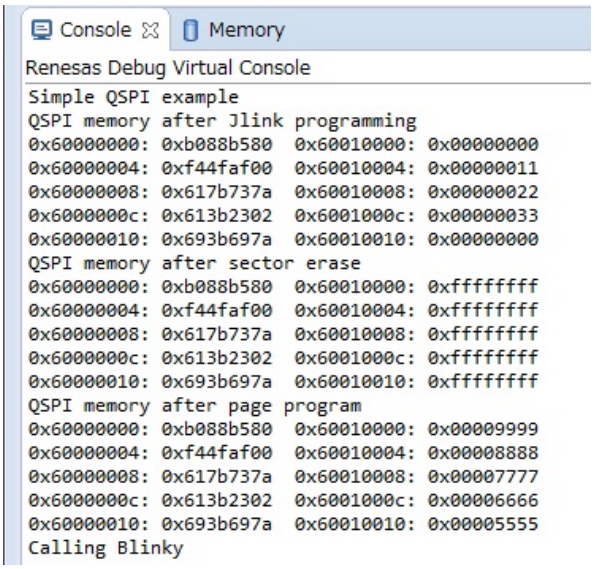

Figure 12. Results Shown by Executing the Debug Virtual Console

3.4 Starting the QSPI Sample Project in the IAR EW for Synergy

The QSPI sample project for IAR EW for Synergy can be used by opening a workspace. This section describes how to open a workspace.

1. Select the **File > Open Workspace** menu and open a workspace. Specify `QSPI_Example.eww` as the workspace from among the QSPI sample project files that have been decompressed.

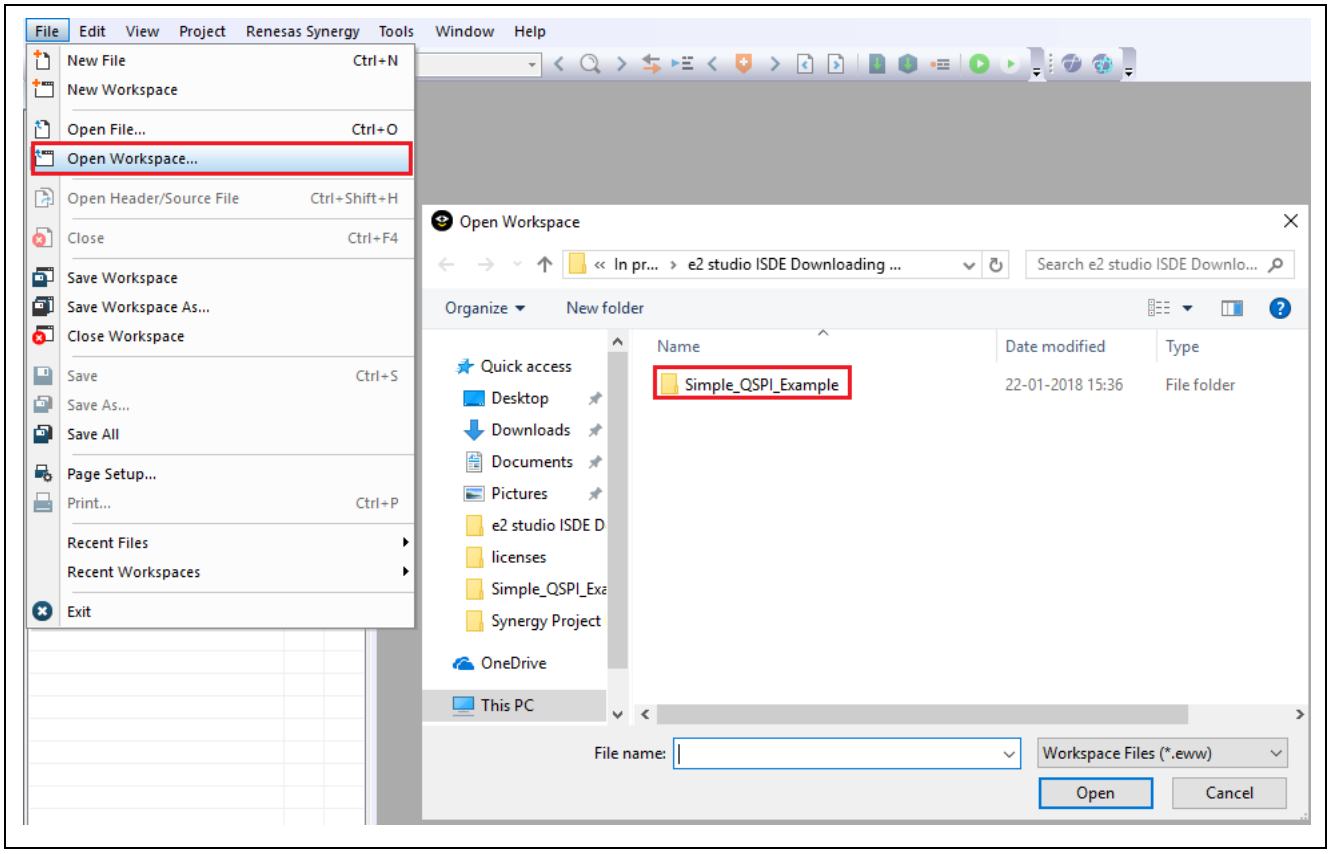

Figure 13. Opening a Workspace

2. The QSPI sample project has now been registered in the workspace. Open the Renesas Synergy™ Standalone Configurator (SSC) and set up a configuration. To set up a configuration, right-click on the Synergy icon under the project tree and select **Open Renesas Synergy Configurator…** or click on the **Synergy Configuration** icon. If you are starting the SSC for the first time, you will be required to set the directories where the SSC and SSP have been installed, and the full path to the license file.

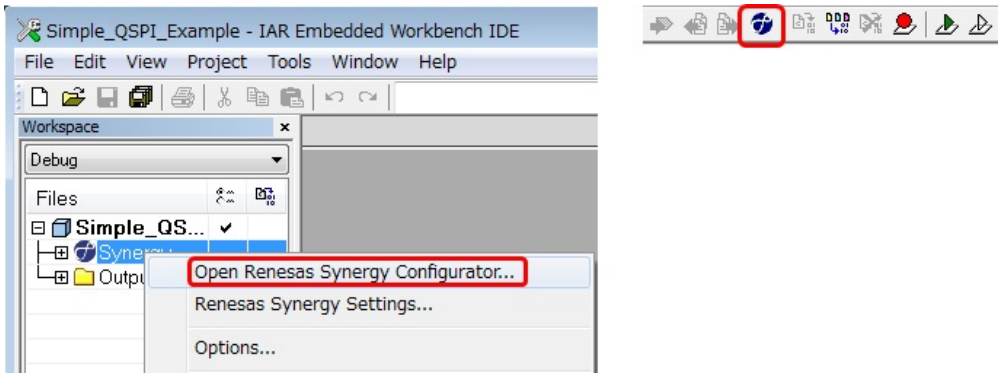

Figure 14. Starting the SSC

Specify the directory where the SSC is installed, the full path to the license file, and click on the **OK** button to complete the setting.

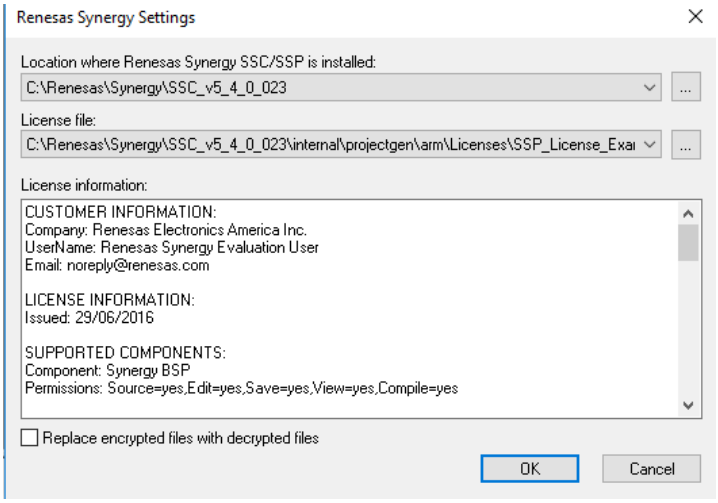

Figure 15. Setting the Directory where the SSC is Installed and the Path to the License File

3. Open the SSC as described in step 2 and generate a project. After the project has been generated, click on the **x** button in the upper right corner of the window to close the window. If the window remains open, control is not returned to IAR EW for Synergy.

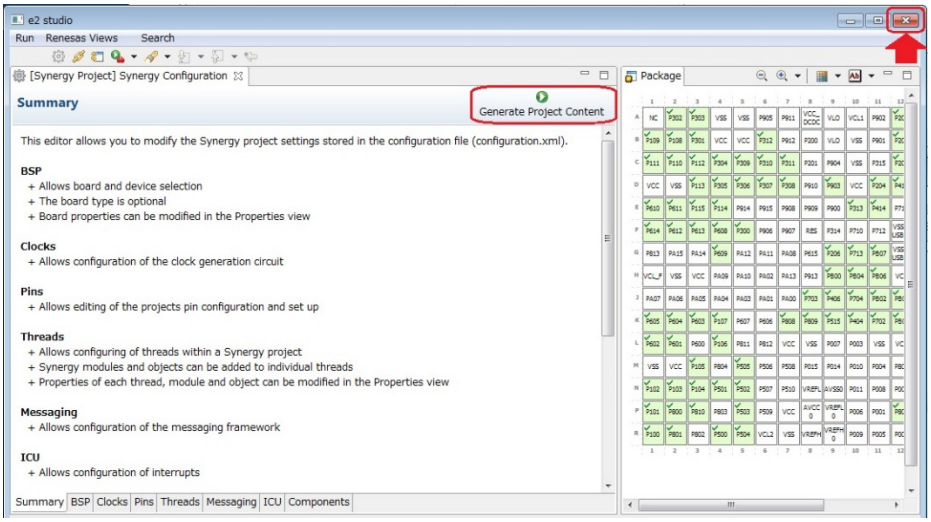

Figure 16. Generating a Project and Ending SSC Operation

4. When control is returned to IAR EW for Synergy, build the project. To build the project, press the **F7** key or select **Project > Make**.

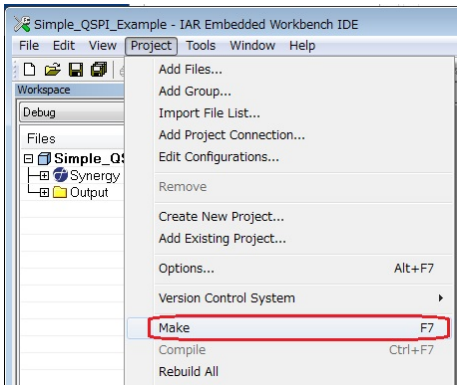

Figure 17. Building the Project

3.4.1 Starting and Checking the Debugger

To start the debugger, select **Project** > **Download and Debug** or click on the **Download and Debug** icon.

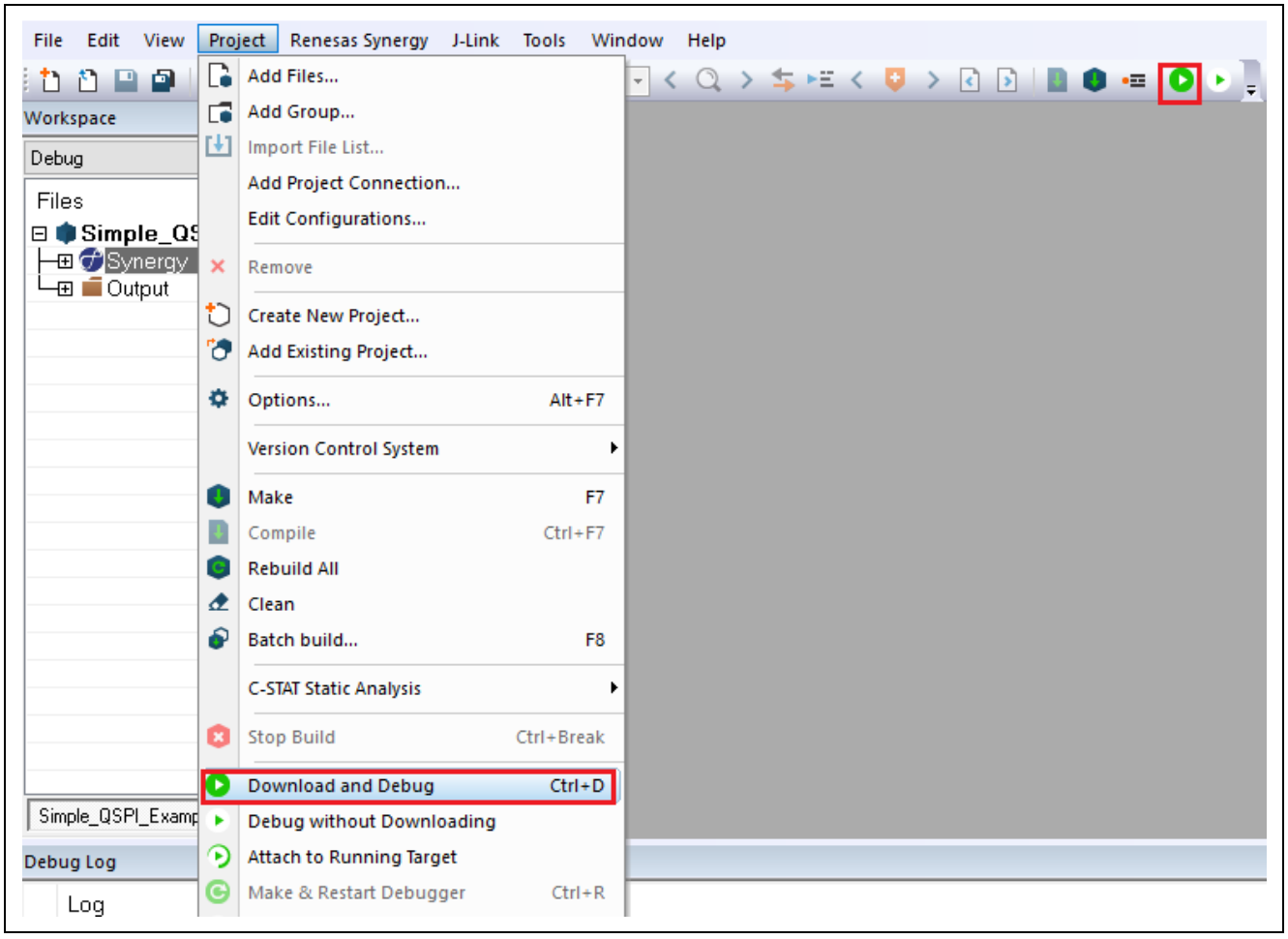

Figure 18. Starting the Debugger

1. When the debugger is started, select **View > Disassembly** to open the **Disassembly** window. By entering 'blinky' in the **Go to** text box in the **Disassembly** window and pressing the **Enter** key, you can confirm that the `blinky()` function has been allocated to the external flash memory area.

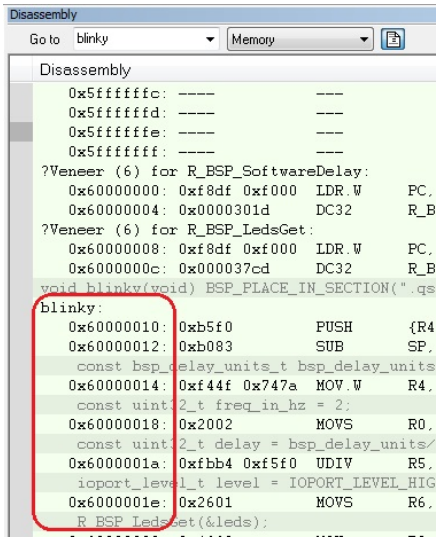

Figure 19. Allocation of the blinky() Function in the IAR EW for Synergy

2. Check the data in address 0x60010000 in the **Memory** window. Display the **Memory** window by selecting **View** > **Memory**. By entering **0x60010000** in the **Go to** text box in the **Memory** window, you can check the data at those locations in the external flash memory.

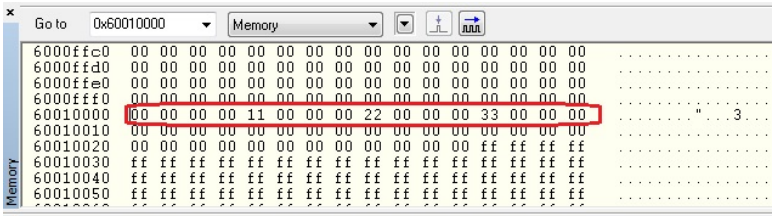

Figure 20. Displaying the External Flash Memory Area in the Memory Window

3. To use the semi-hosting function to check the results of executing the program, use the **Terminal I/O** window. Select **View** > **Terminal I/O** to open the **Terminal I/O** window.
4. Set a breakpoint at the point where the `blinky()` function is called from `hal_entry.c`.

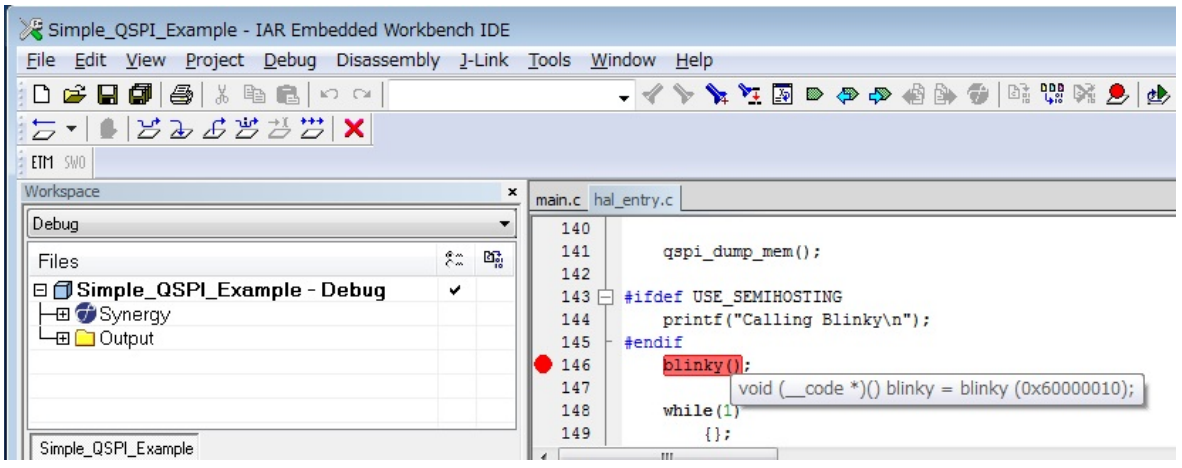

Figure 21. Setting a Breakpoint

5. While the program executes up to the breakpoint, the data in the external flash memory are displayed in the **Terminal I/O** window. A break then occurs at the breakpoint that was set in step 4. The data is displayed in the **Terminal I/O** window as shown in Figure 22 by using the semi-hosting function.

Figure 22. Display in the Terminal I/O Window

6. When continuous execution of the program resumes, the `blinky()` function downloaded to the external flash memory area is executed to cause LED1 and LED2 on the board to blink on and off at one-second intervals.

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar.28.17	-	Initial version
1.01	Aug.24.17	-	Updated to SSP v1.3.0
1.02	Sep.27.17	1	Environment of SSP version changed
1.03	Jan.19.18	-	Updated for SSP v1.4.0
1.04	May.07.19	-	Added note for DK-S7G2 v4.1. Updated for SSP v1.6.0.

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. 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.4.0-1 November 2017)

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/.