RX Family

Dual Mode Usage Guide

Introduction

This application note provides an overview of dual mode and describes the preparation, debugging method, and how to write a program by using Renesas Flash Programmer (RFP hereafter). The descriptions are mainly based on examples of 2-MB products of the RX671 Group.

For 2-MB products of the RX671 Group in the default shipment state, the address range for bank 0 is from FFF0 0000h to FFFF FFFFh, and for bank 1 is from FFE0 0000h to FFEF FFFFh. Interpret the addresses in this document according to your MCU.

Target Devices

RX26T Group

RX651 Group

RX65N Group

RX66N Group

RX671 Group

RX72M Group

RX72N Group

When applying this application note to another MCU, make modifications according to the specifications of that MCU, and then perform careful evaluation.

Contents

1. Overview

1.1 Overview of Dual Mode

The bank mode switching function allows you to choose from two options: linear mode or dual mode. In dual mode, the user area in code flash memory is handled as two bank areas.

The startup bank selection function selects the bank area in which the program starts, thus providing a safe method of updating the program when a rewrite operation is suspended because of, for example, a reset. Figure 1.1 Example of Allocated Addresses When Switching the Startup Bank shows write addresses when the startup bank is switched in dual mode.

By resetting the MCU after setting the BANKSEL.BANKSWP[2:0] bits in the option-setting memory, the addresses are switched between bank 1 and bank 0, and then the program located at addresses FFF0 0000h to FFFF FFFFh is run. If the addresses are switched by selecting the startup bank, the addresses at which FACI commands can be issued are also replaced. For example, if the BANKSEL.BANKSWP[2:0] bits are set to 000b, the address range of the bank 0 area subject to processing by FACI commands is from FFE0 0000h to FFEF FFFFh. The startup bank selection function is disabled when linear mode is selected.

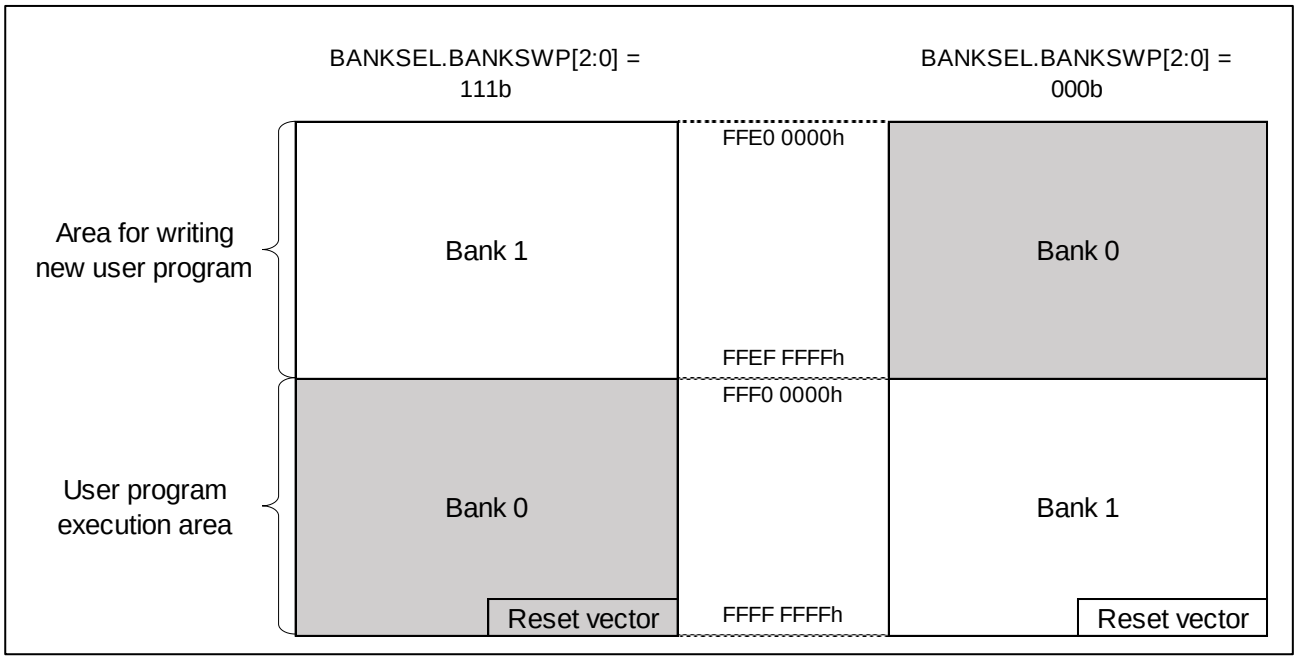


Figure 1.1 Example of Allocated Addresses When Switching the Startup Bank

1.2 Updating Firmware

This section describes the firmware update operation using the dual bank function.

The dual bank function ensures that firmware is updated safely when a rewrite operation is suspended because of, for example, a reset. Note, however, that the capacity of usable code flash memory is halved.

Figure 1.2 provides Overview of Firmware Update Operation.

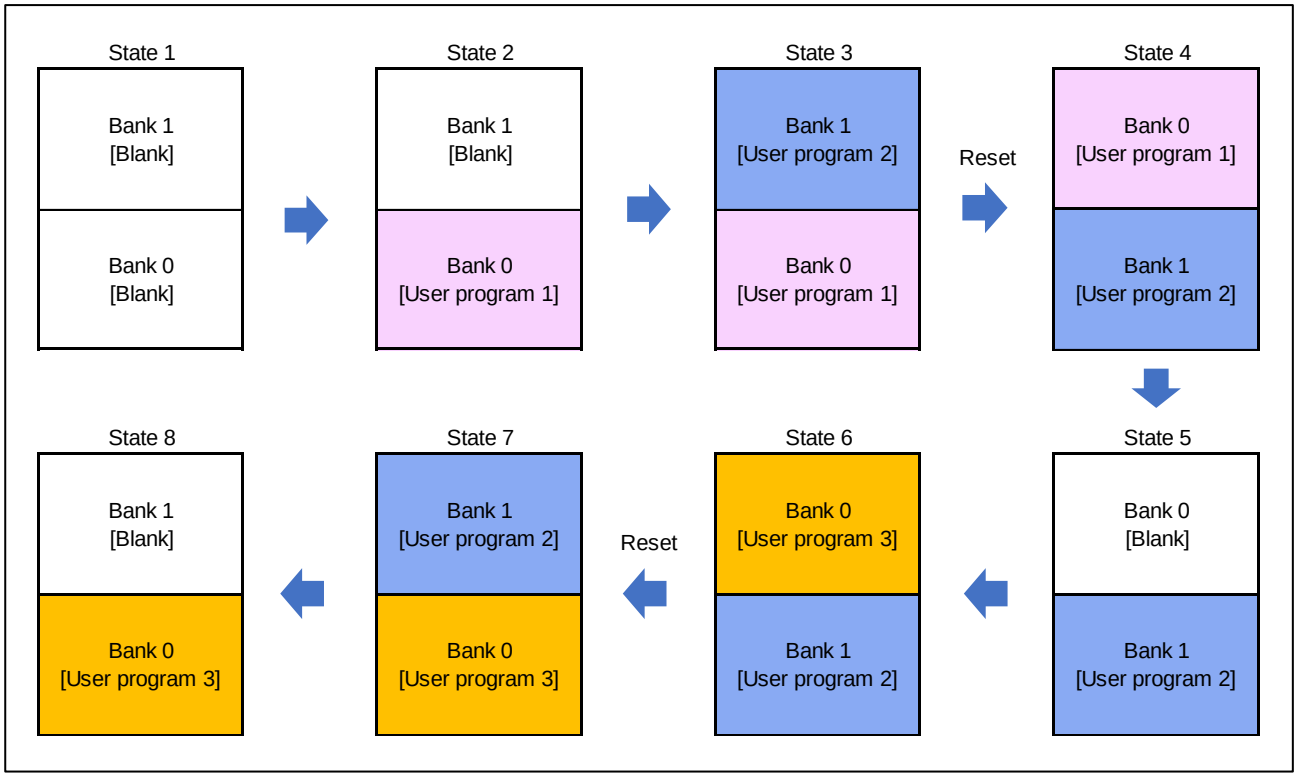


Figure 1.2 Overview of Firmware Update Operation

(1) State 1 is the state of the MCU at shipment. Writing user program 1 to bank 0 by using a flash writer or a similar tool causes a transition to state 2.
(2) After developing user program 2, writing user program 2 to bank 1 by using the flash rewrite program in bank 0 in state 2 causes a transition to state 3.*1
(3) Resetting the MCU after switching the BANKSWP[2:0] bits by using the flash rewrite program causes a transition to state 4. (Switching the BANKSWP[2:0] bits must be performed in the RAM.)
(4) In state 4, user program 2 in bank 1 is run. At this time, erasing user program 1 from bank 0 causes a transition to state 5.
(5) After developing a new user program, repeat the same steps. Interpret state 8 as state 2.

Note: 1. If rewrite fails because of an unexpected power outage or reset, BANKSWP[2:0] bits are not switched and user program 1 in bank 0 will operate after a reset. Erase the entire area of bank 1, and then perform step (2) again.

2. Preparation

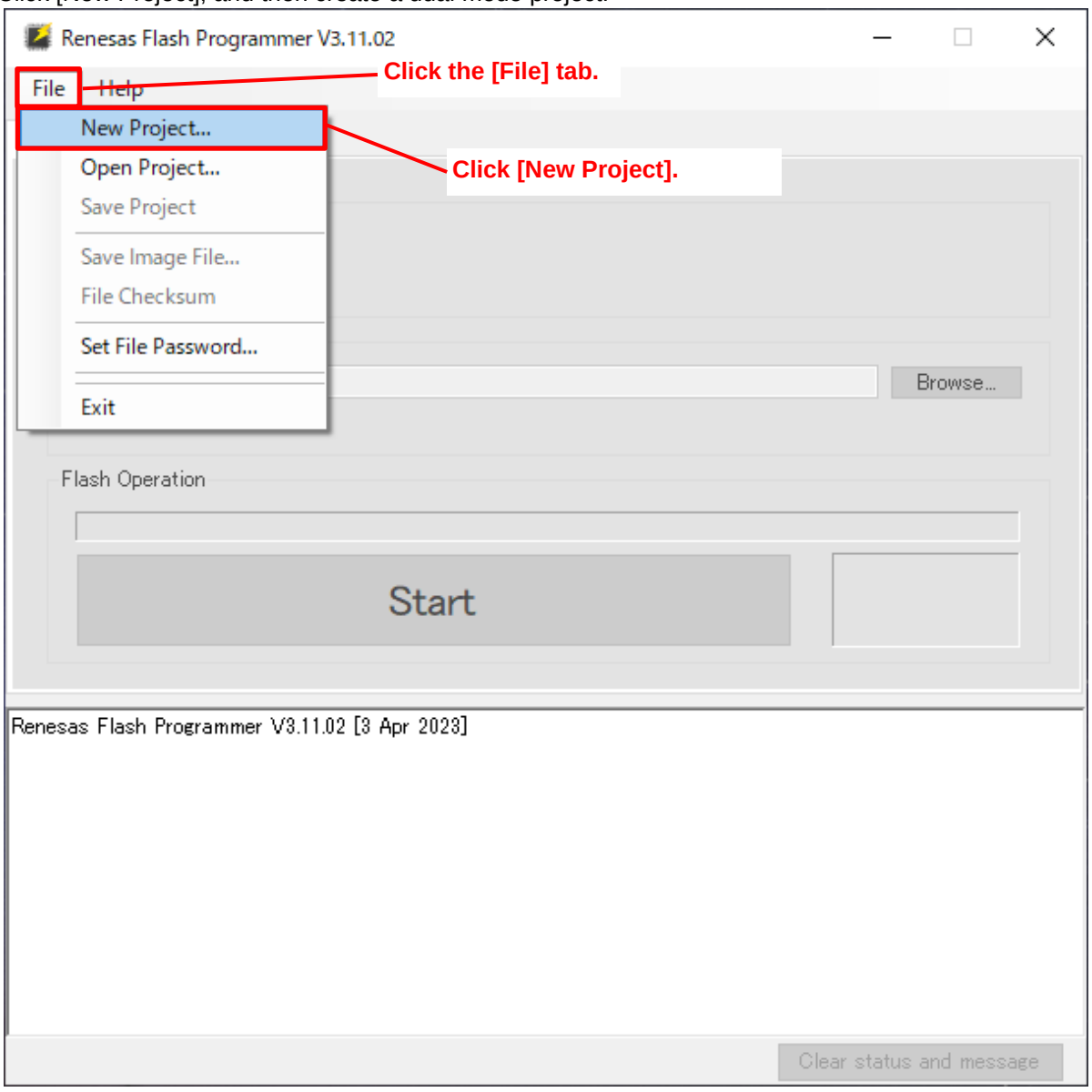
2.1 If Bank Positions Are Unclear

If the address positions of banks 0 and 1 are not clear, you can identify the addresses by reading the values in the option-setting memory by using RFP. Another solution is to erase the option-setting memory and entire flash memory to return them to the shipment state. For details about how to read values in the option-setting memory, see section 2.1.1, Reading the Option-Setting Memory. For details about how to erase the option-setting memory and flash memory, see section 2.1.2, Returning the MCU to the Shipment State.

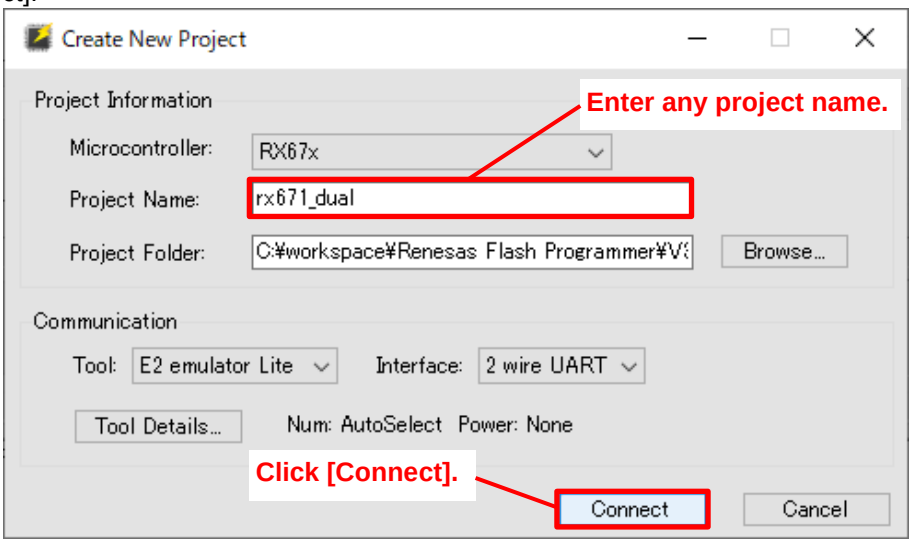
Note that RFP identifies an MCU differently according to whether the MCU is in dual mode or linear mode. Therefore, connecting a project whose bank mode differs from that of the MCU causes an error. When creating a new project, we recommend you add a string such as "_dual" or "_linear" to the project name according to the bank mode of the MCU to be connected. When connecting the project to the MCU for the first time, the bank mode information of the MCU is saved in the project.

2.1.1 Reading the Option-Setting Memory

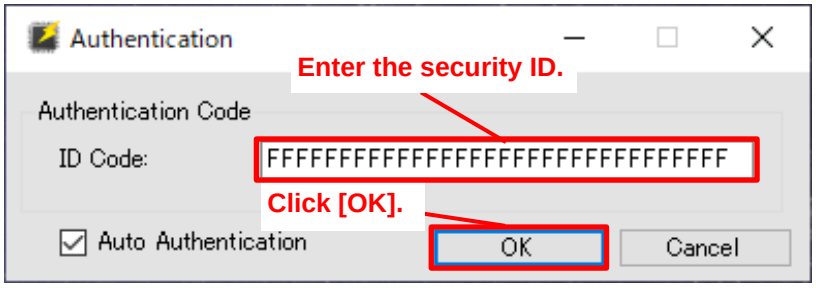
(1) Start RFP.
(2) Click the [File] tab.
(3) Click [New Project], and then create a dual mode project.



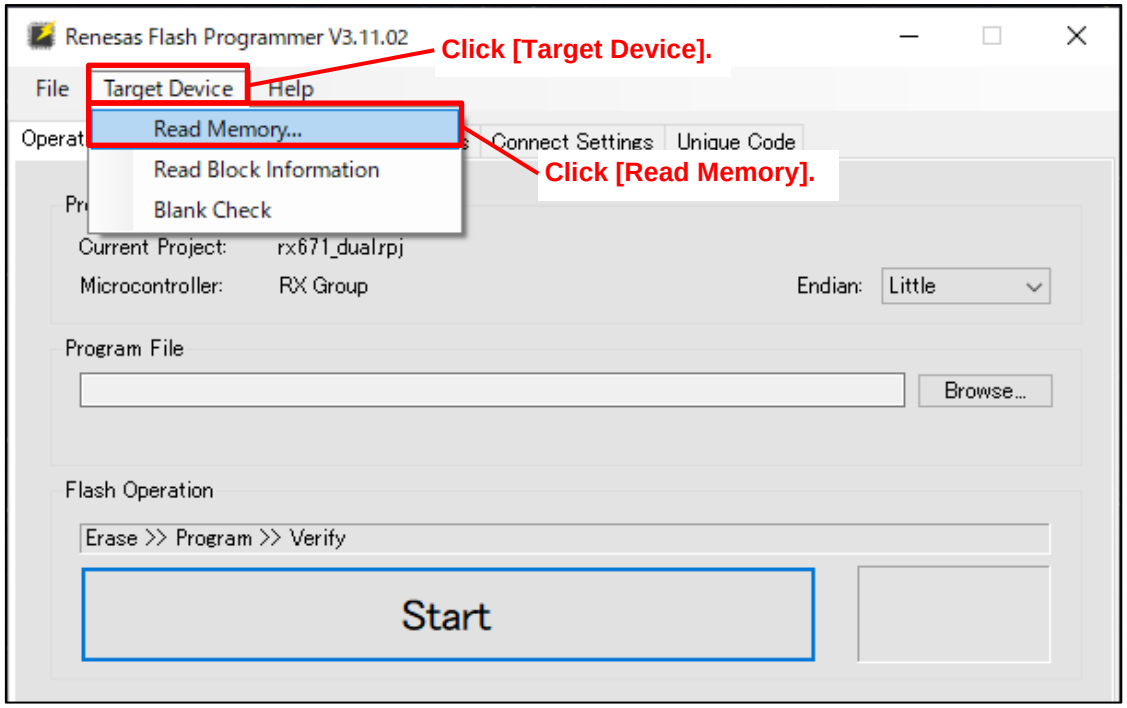
(4) Enter any project name in the [Project Name:] text box.
(5) Change other settings according to the system.
(6) Click [Connect].



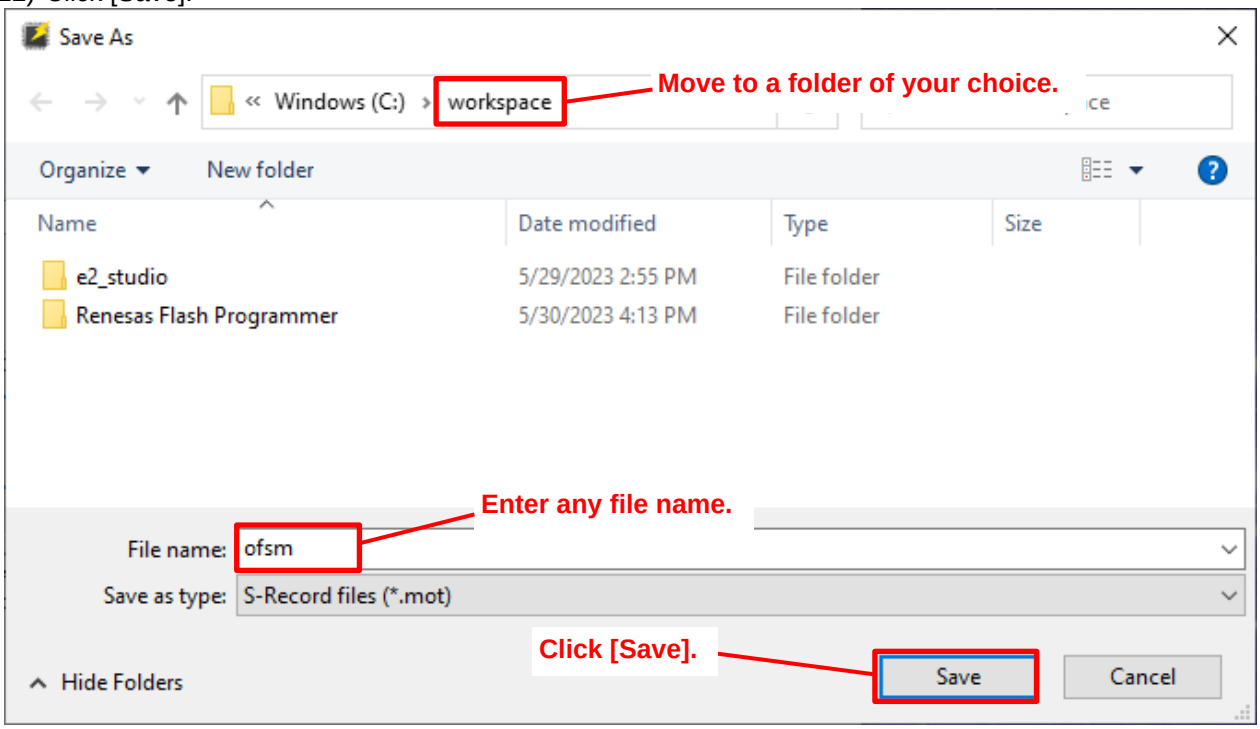
(7) In the [ID Code:] text box, enter the set security ID.
(8) Click [OK].



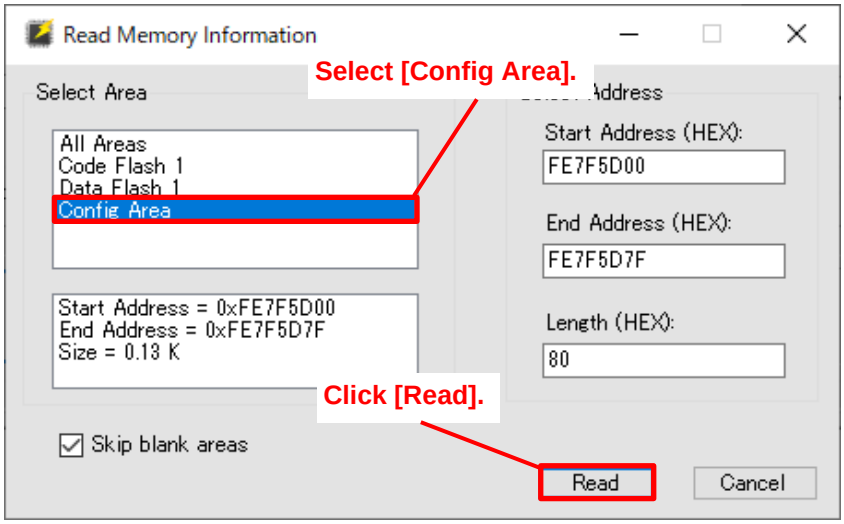
(9) Click the [Target Device] tab, and then click [Read Memory].



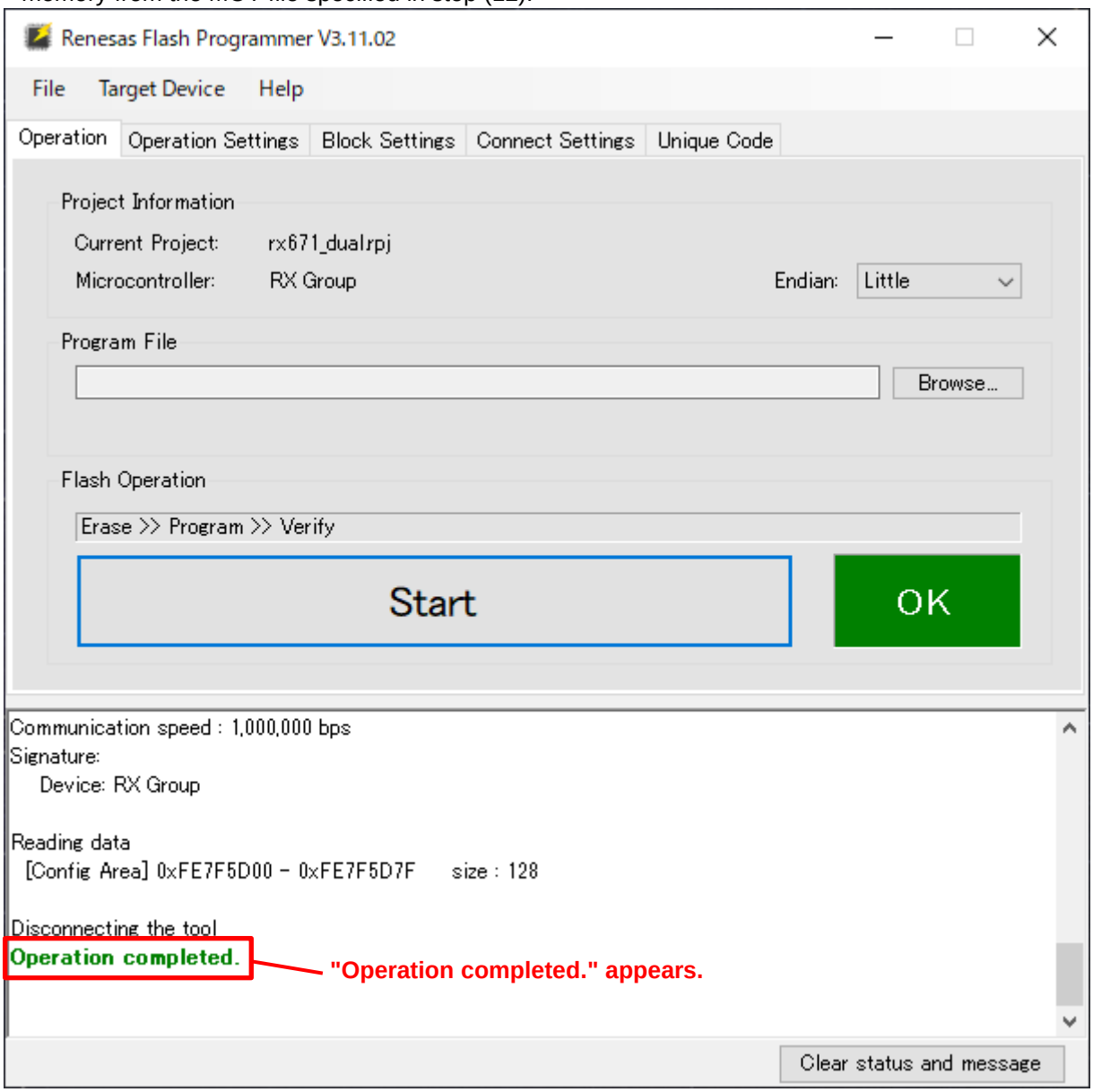
(10) Move to a folder of your choice
(11) Enter any file name in the [File name] text box.
(12) Click [Save].



(13) In the [Select Area] field, select [Config Area].
(14) Click [Read].



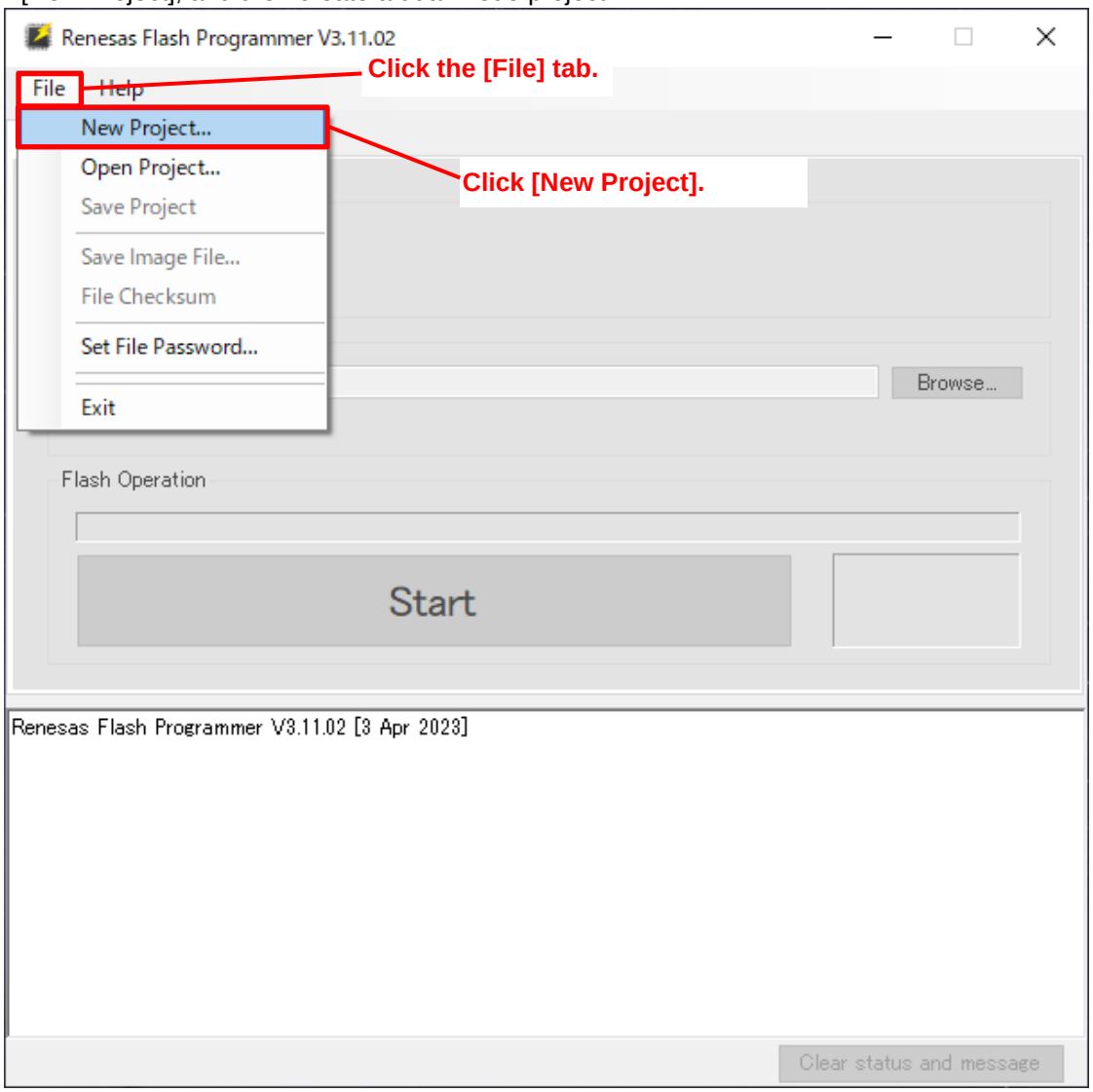
(15) When "Operation completed." appears, processing is complete. You can read the option-setting memory from the MOT file specified in step (12).



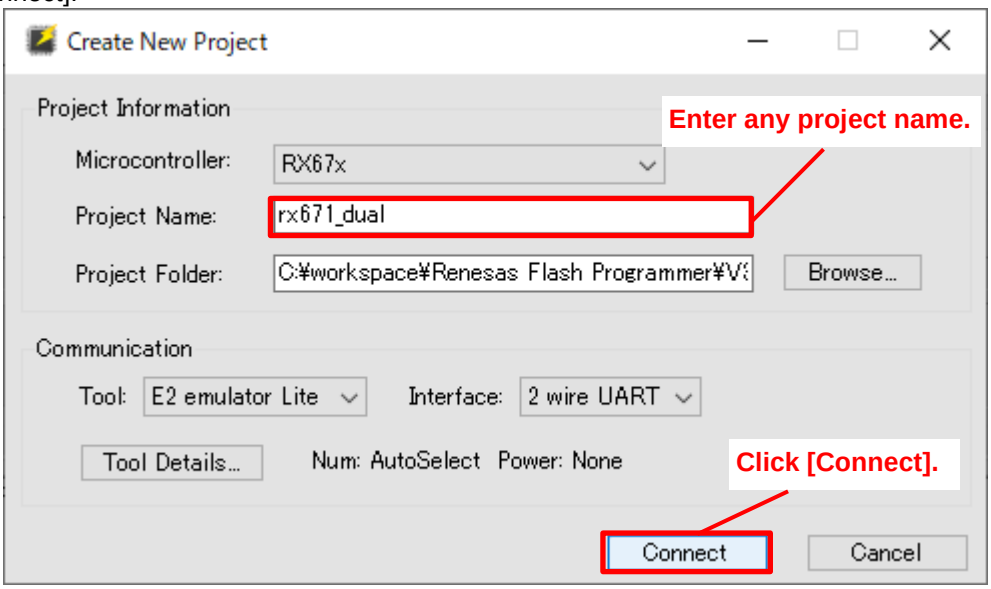
2.1.2 Returning the MCU to the Shipment State

If an access window has been set, you need to release the setting in advance. For details, refer to user's manual of your MCU.

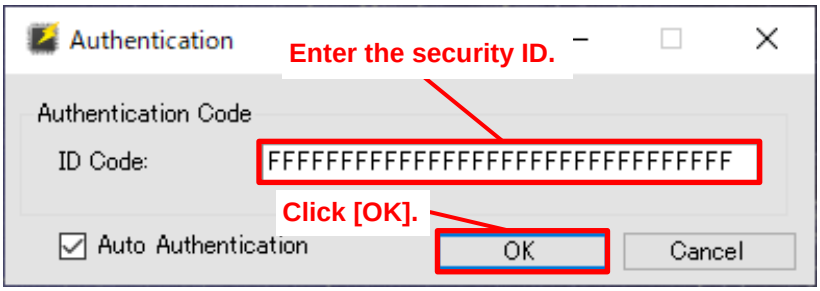
(1) Start RFP.
(2) Click the [File] tab.
(3) Click [New Project], and then create a dual mode project.



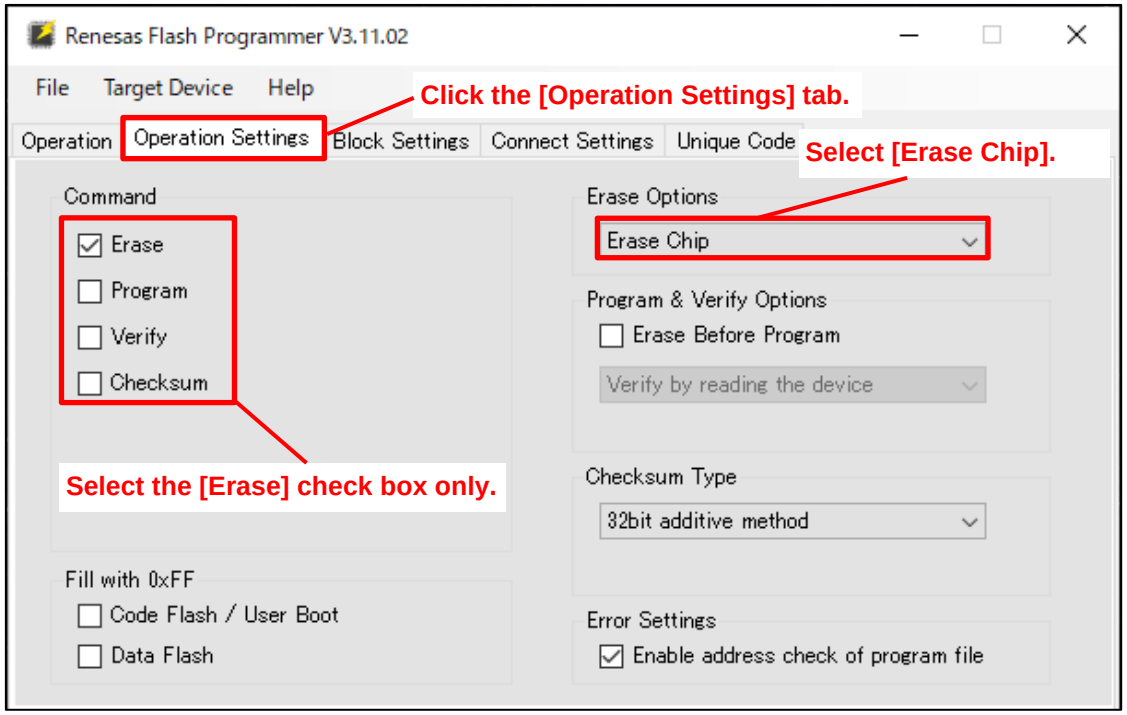
(4) Enter any project name in the [Project Name:] text box.
(5) Change other settings according to the system.
(6) Click [Connect].



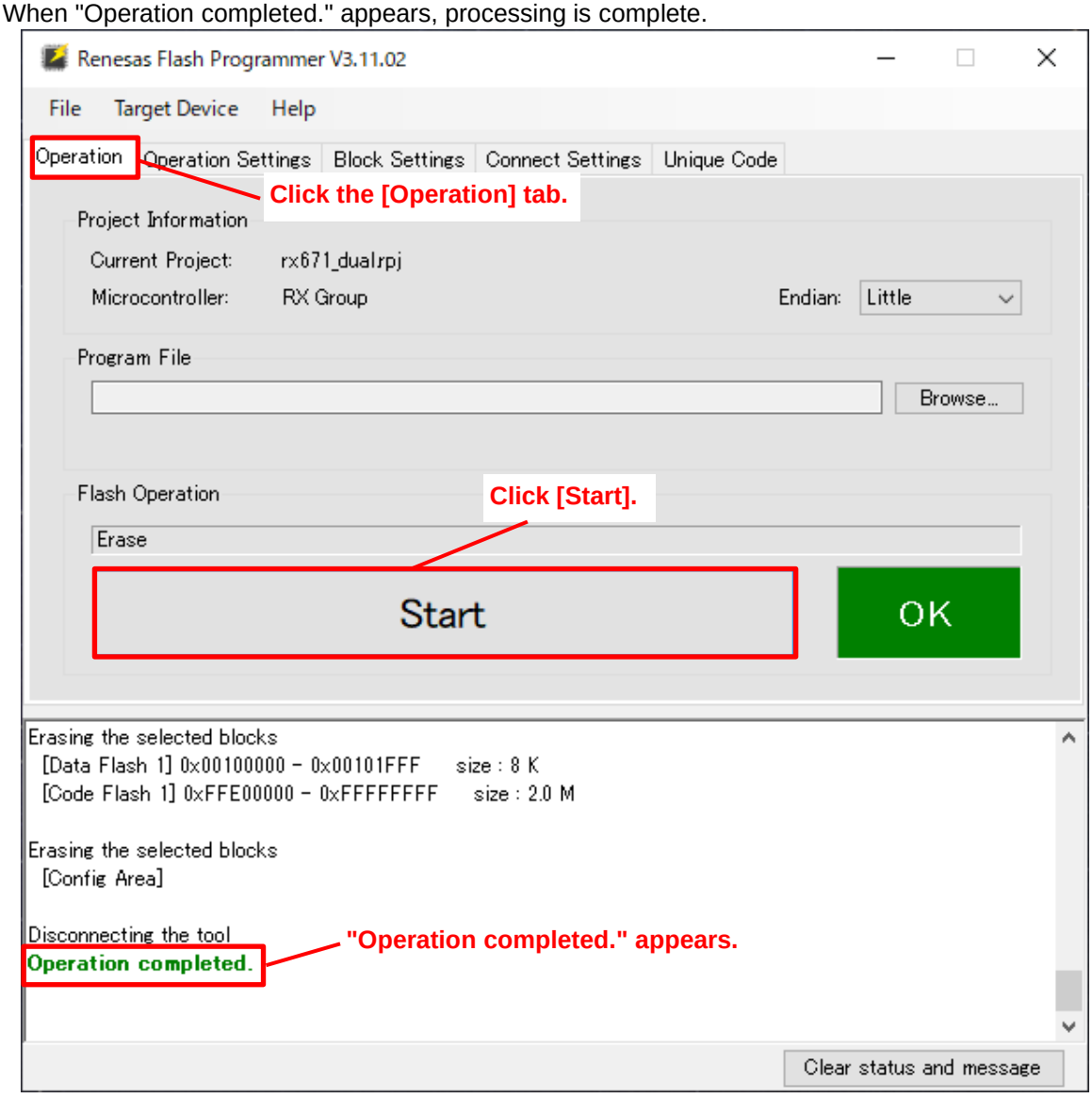
(7) In the [ID Code:] text box, enter the set security ID.
(8) Click [OK].



(9) Click the [Operation Settings] tab.
(10) For [Command], select the [Erase] check box only.
(11) For [Erase Options], select [Erase Chip].



(12) Open the [Operation] tab, and then click [Start].
(13) When "Operation completed." appears, processing is complete.



3. Debugging Method

This section describes how to perform debugging for dual mode by using e² studio.

To develop and debug a user program, place it in the execution area from FFF0 0000h to FFFF FFFFh irrespective of the addresses at which the user program is written.

To develop a new user program, see section 3.1, When Developing User Program Version 1.00. To update a user program, see section 3.2, When Developing User Program Version 1.01 or Later.

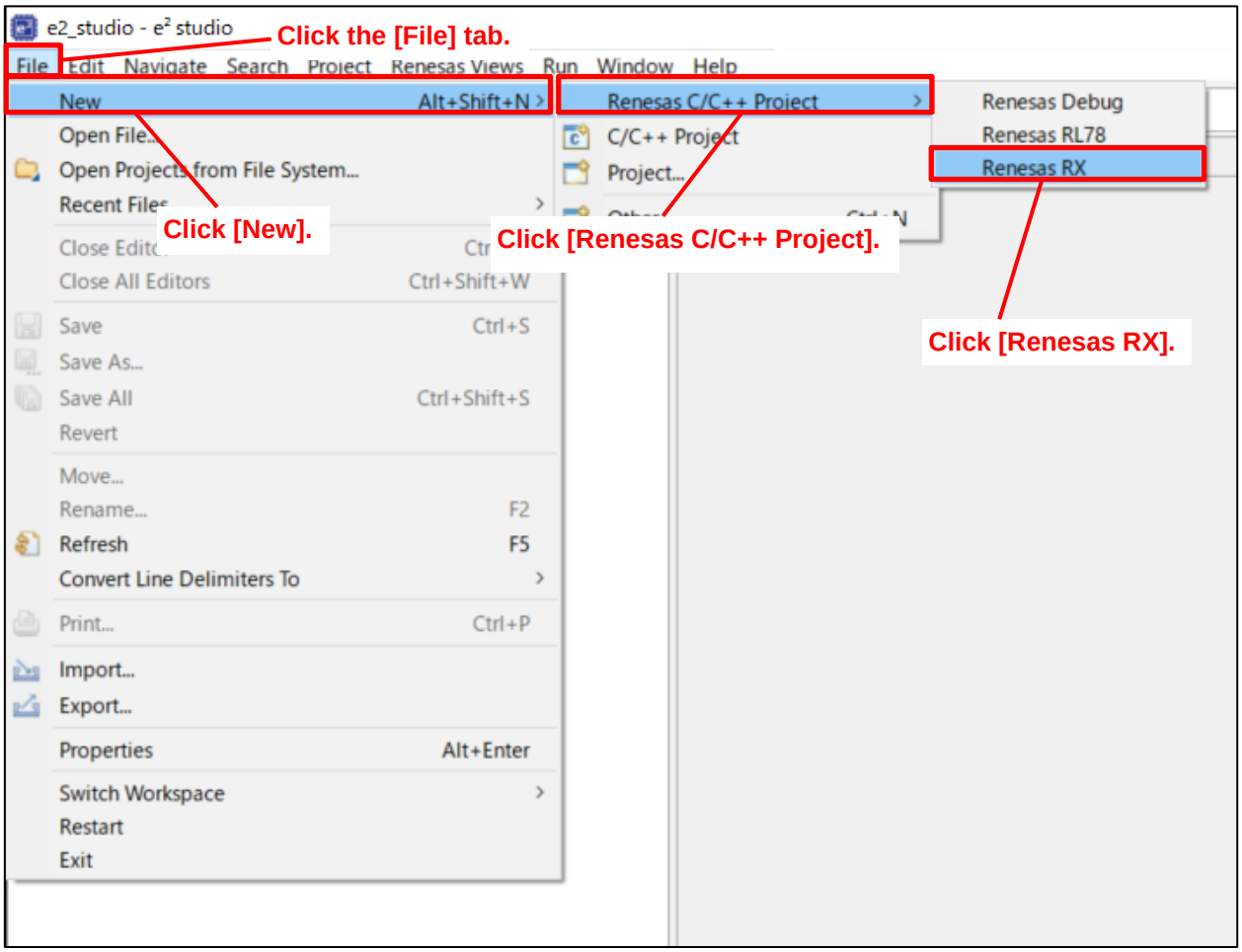
3.1 When Developing User Program Version 1.00

This section describes the procedures for creating a project of a user program, setting up Board Support Packages (BSP hereafter), and setting the debug configuration.

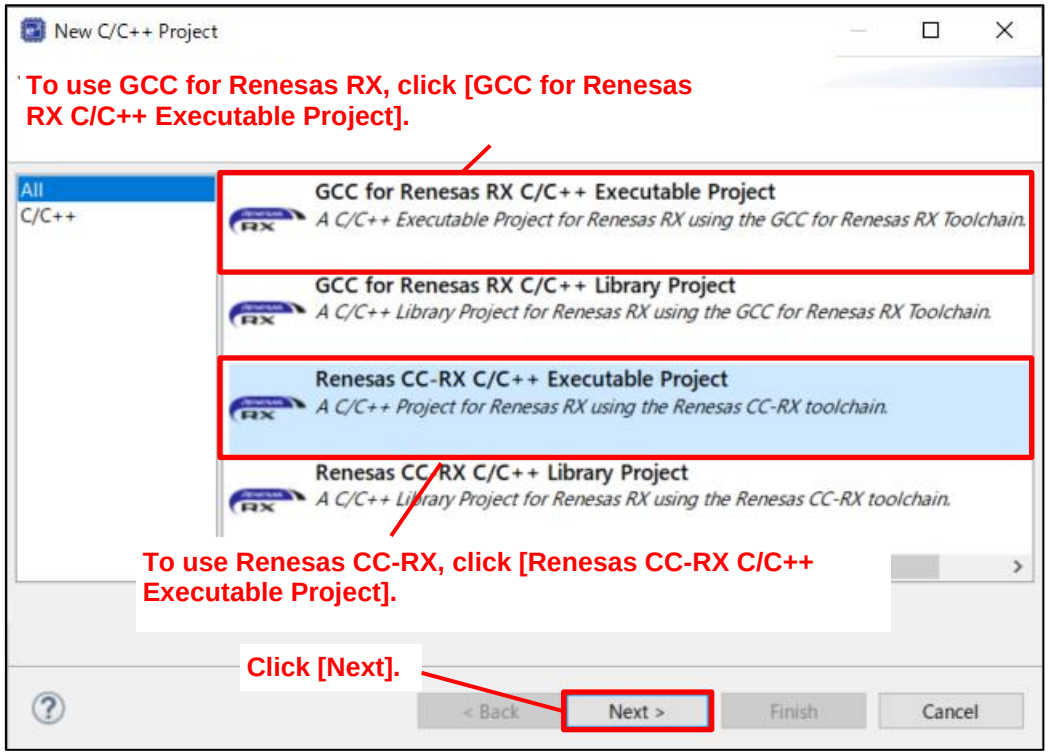
Section 3.1.1 describes Procedure for Creating a Project, section 3.1.2 describes Procedure for Setting Up BSP, and section 3.1.3 describes Procedure for Setting the Debug Configuration.

3.1.1 Procedure for Creating a Project

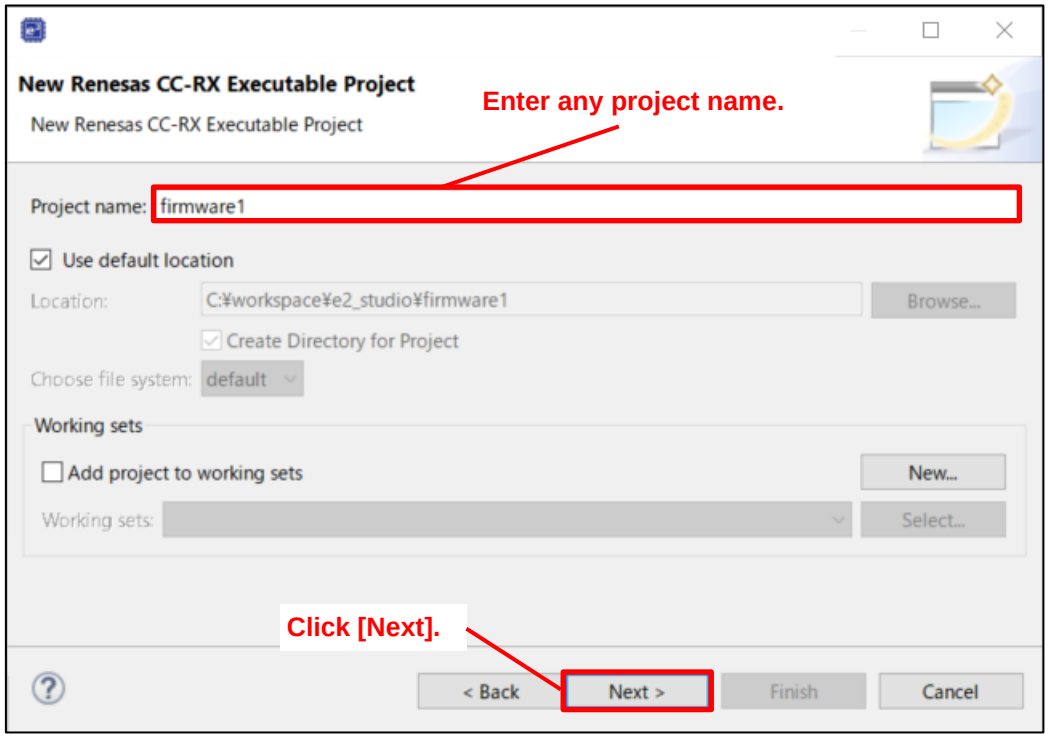
(1) Start e² studio.
(2) Click the [File] tab.
(3) Click [New], [Renesas C/C++ Project], and then [Renesas RX].



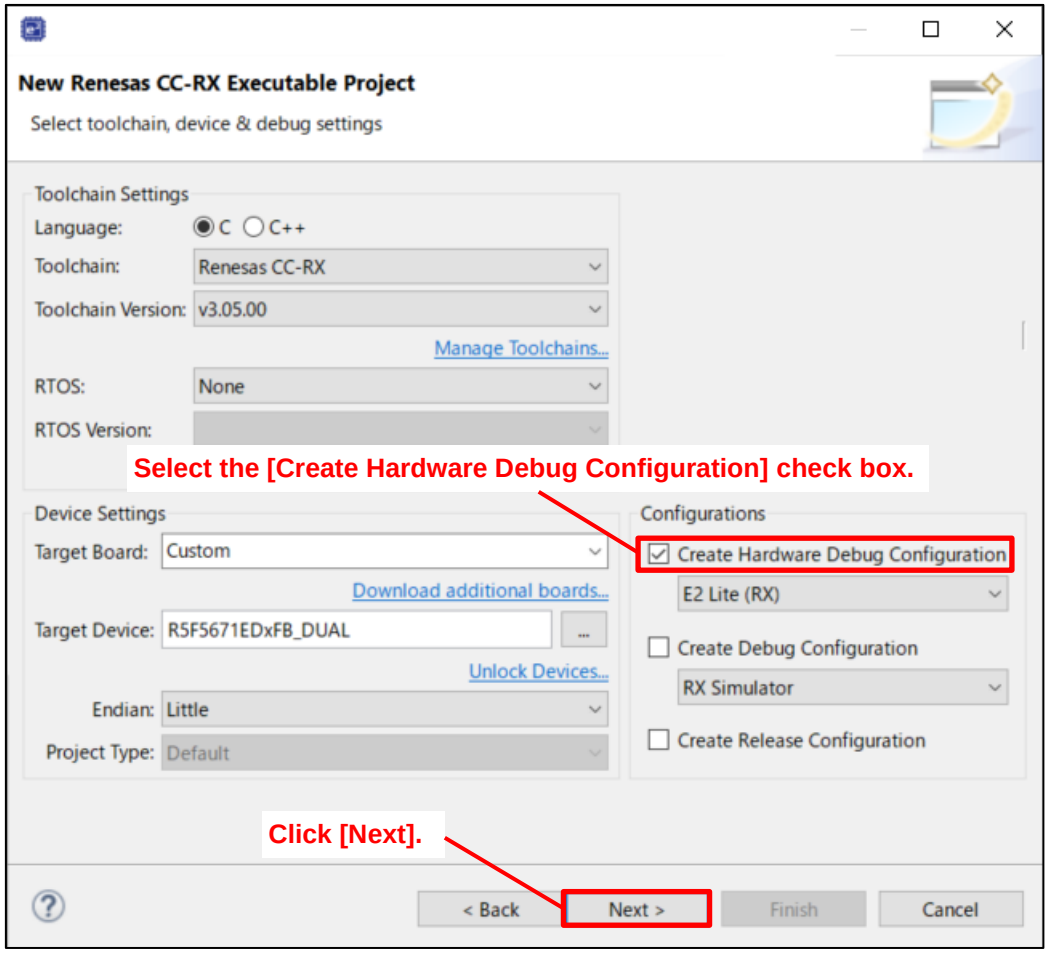
(4) Click [GCC for Renesas RX C/C++ Executable Project] or [Renesas CC-RX C/C++ Executable Project] according to the complier you are using (this example uses Renesas CC-RX).
(5) Click [Next].



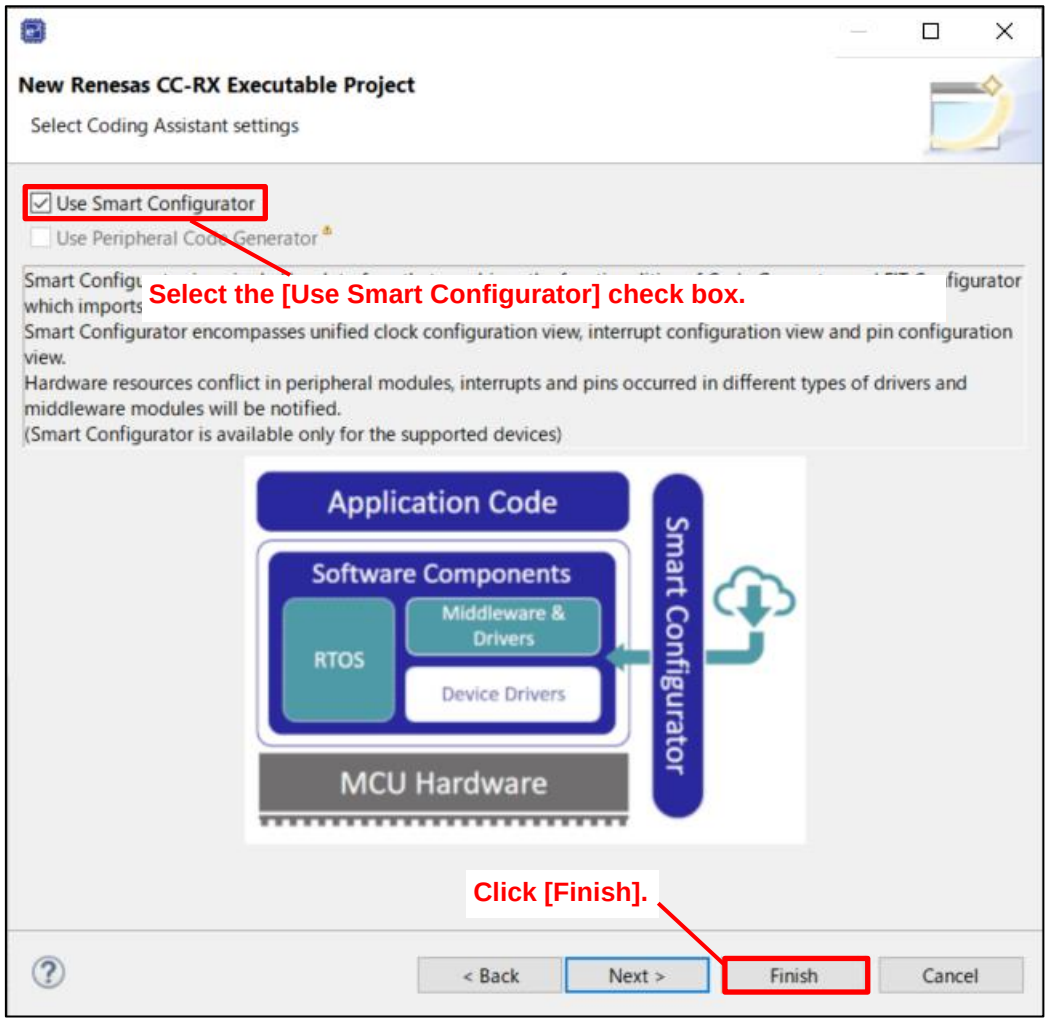
(6) Enter any project name in the [Project name:] text box.
(7) Change other settings according to the system.
(8) Click [Next].



(9) Select the [Create Hardware Debug Configuration] check box.
(10) Change other settings according to the system.
(11) Click [Next].

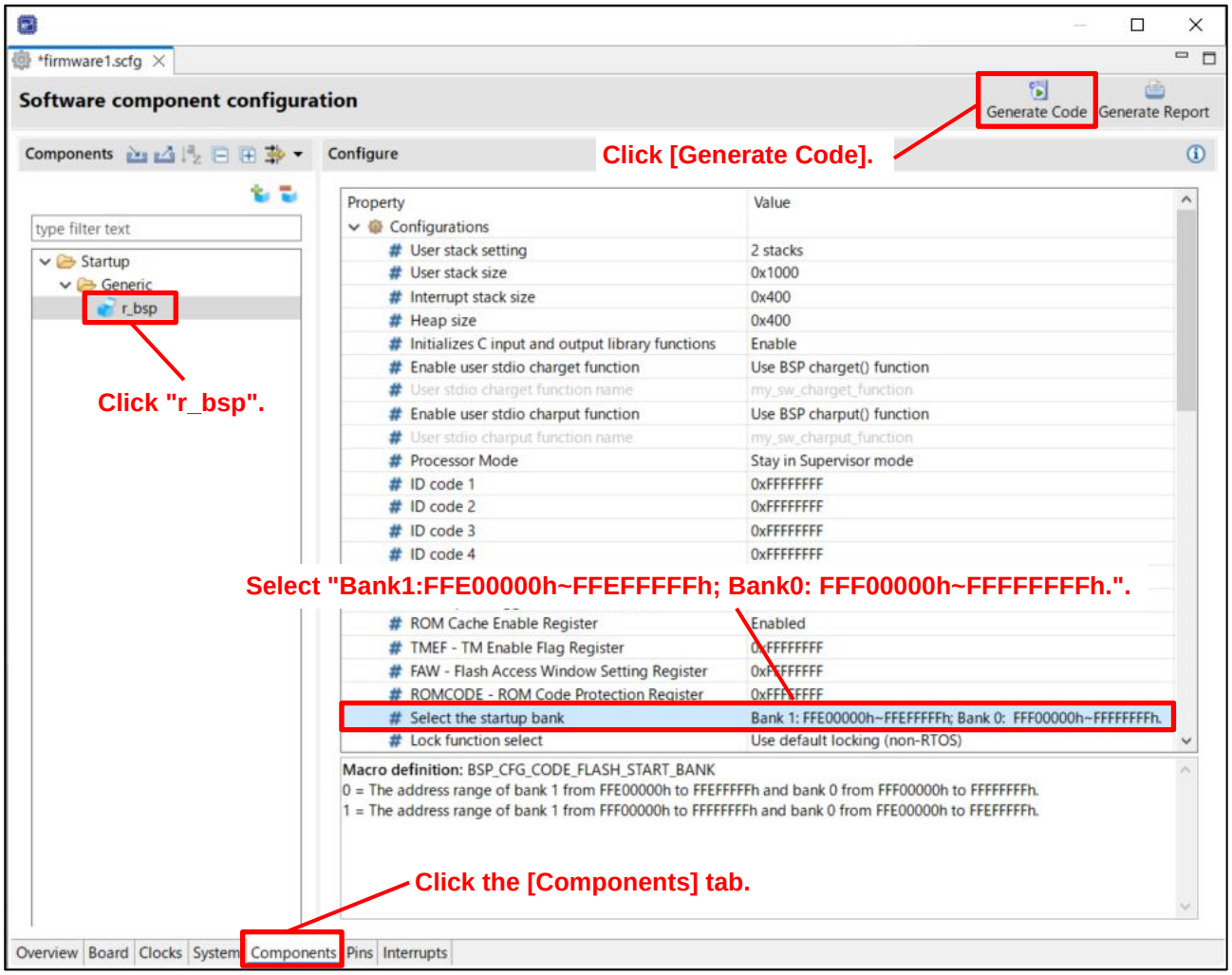


(12) Select the [Use Smart Configurator] check box.
(13) Click [Finish] to complete creation of the project.



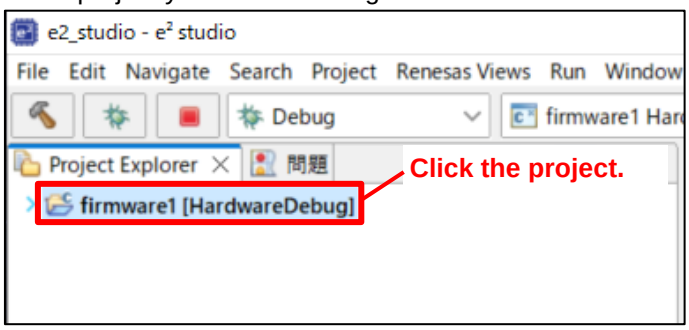
3.1.2 Procedure for Setting Up BSP

(1) Open Smart Configurator.
(2) Click the [Components] tab.
(3) Click "r_bsp".
(4) For "Select the startup bank", select "Bank1:FFE00000h~FFEFFFFFh; Bank0: FFF00000h~FFFFFFFFh.".
(5) Change other settings according to the system.
(6) Click [Generate Code] to apply the BSP contents to the source code.
(7) Perform a build. Setting up BSP is now complete.

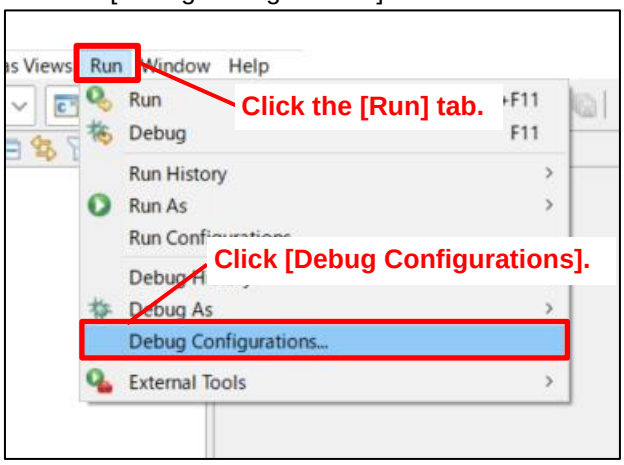


3.1.3 Procedure for Setting the Debug Configuration

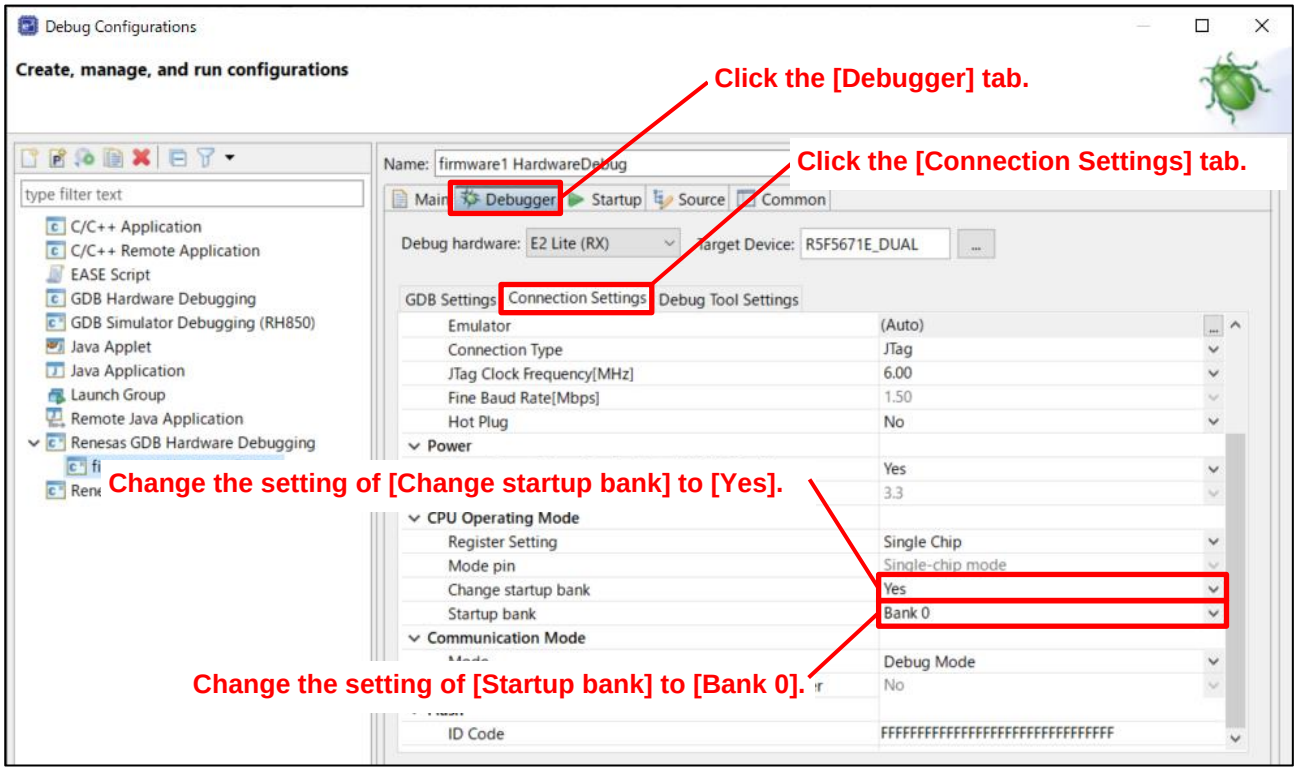
(1) In Project Explorer, click the project you want to configure.



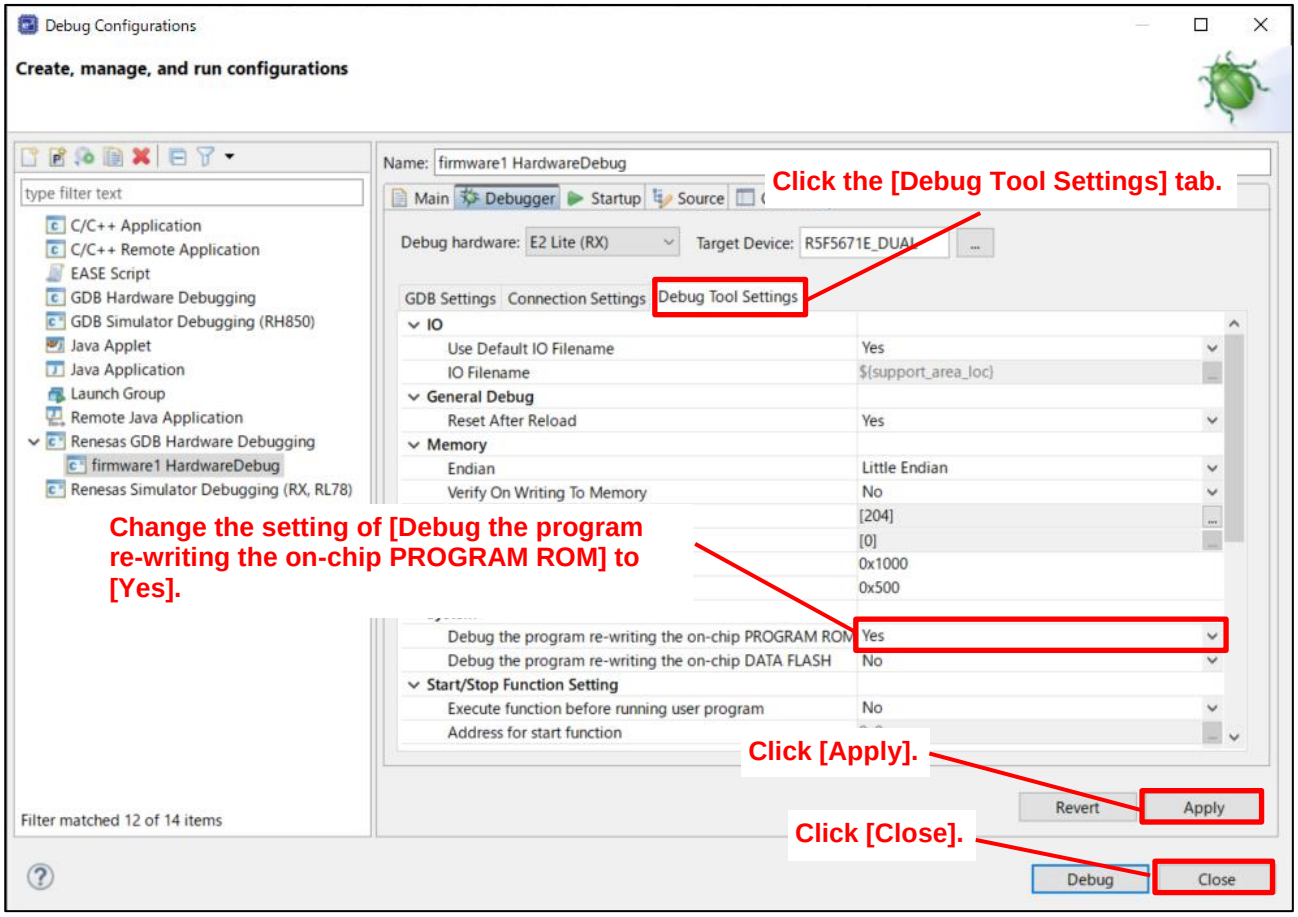
(2) Click the [Run] tab, and then click [Debug Configurations].



(3) In the [Debugger] tabbed page, click the [Connection Settings] tab.
(4) Change the setting of [Change startup bank] to [Yes].
(5) Change the setting of [Startup bank] to [Bank 0].



(6) Click the [Debug Tool Settings] tab.
(7) Change the setting of [Debug the program re-writing the on-chip PROGRAM ROM] to [Yes].
(8) Click [Apply], and then click [Close]. Setting the debug configuration is now complete.



3.2 When Developing User Program Version 1.01 or Later

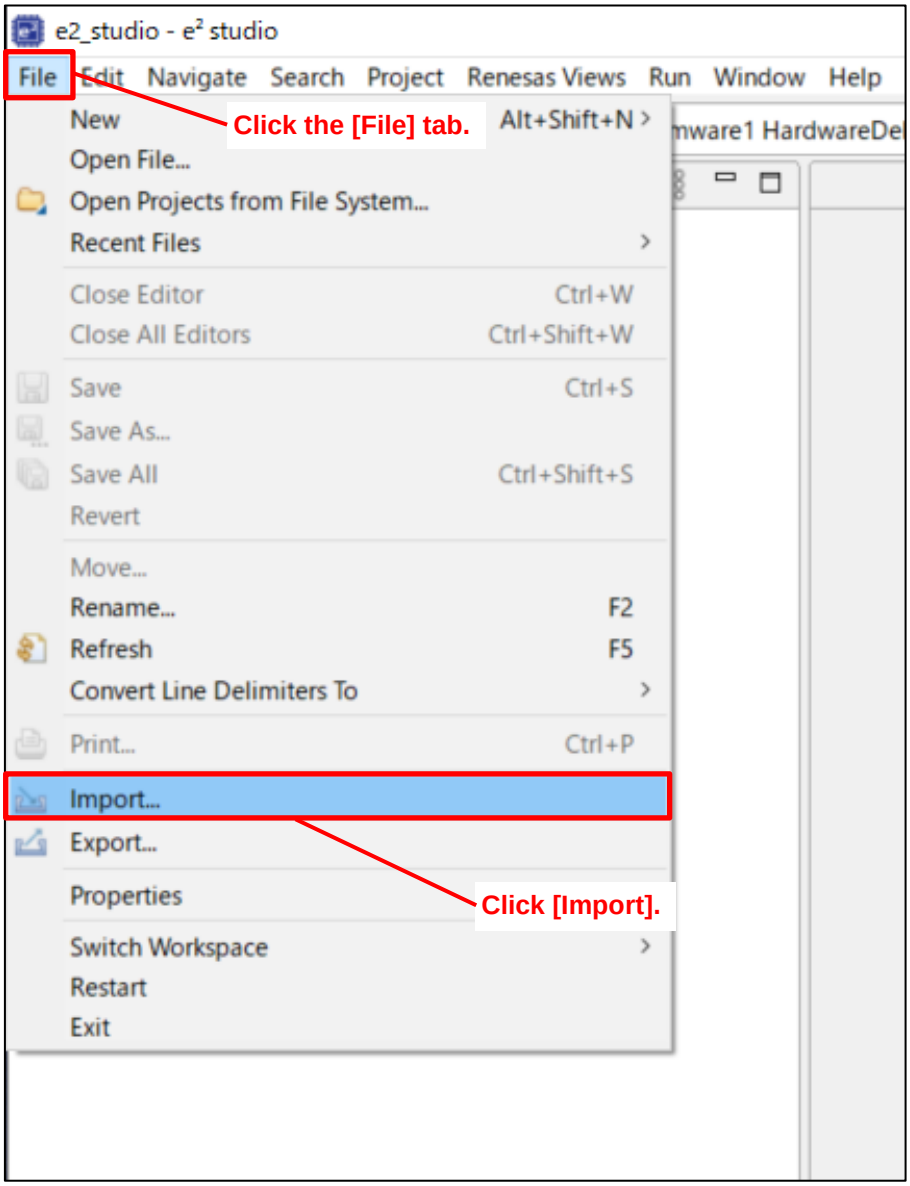
This section describes the procedures for creating a project for updating the user program version 1.00, setting up BSP, and setting the debug configuration.

Section 3.2.1 describes Procedure for Creating a Project, section 3.2.2 describes Procedure for Setting Up BSP, and section 3.2.3 describes Procedure for Setting the Debug Configuration.

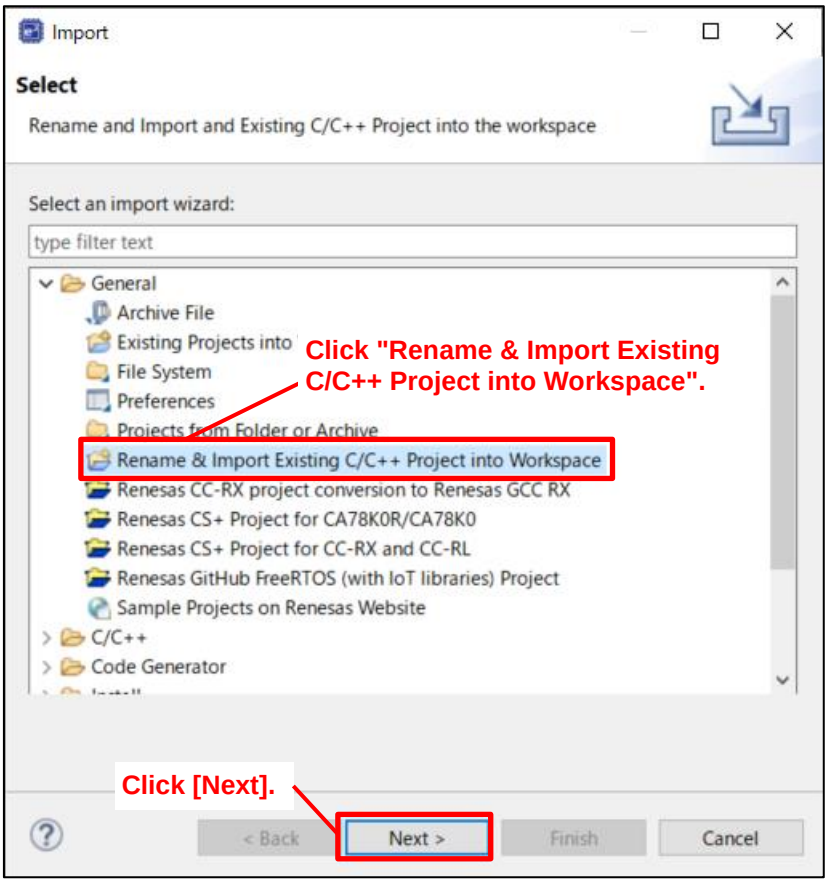
3.2.1 Procedure for Creating a Project

This section describes the procedure for creating a project based on the user program version 1.00.

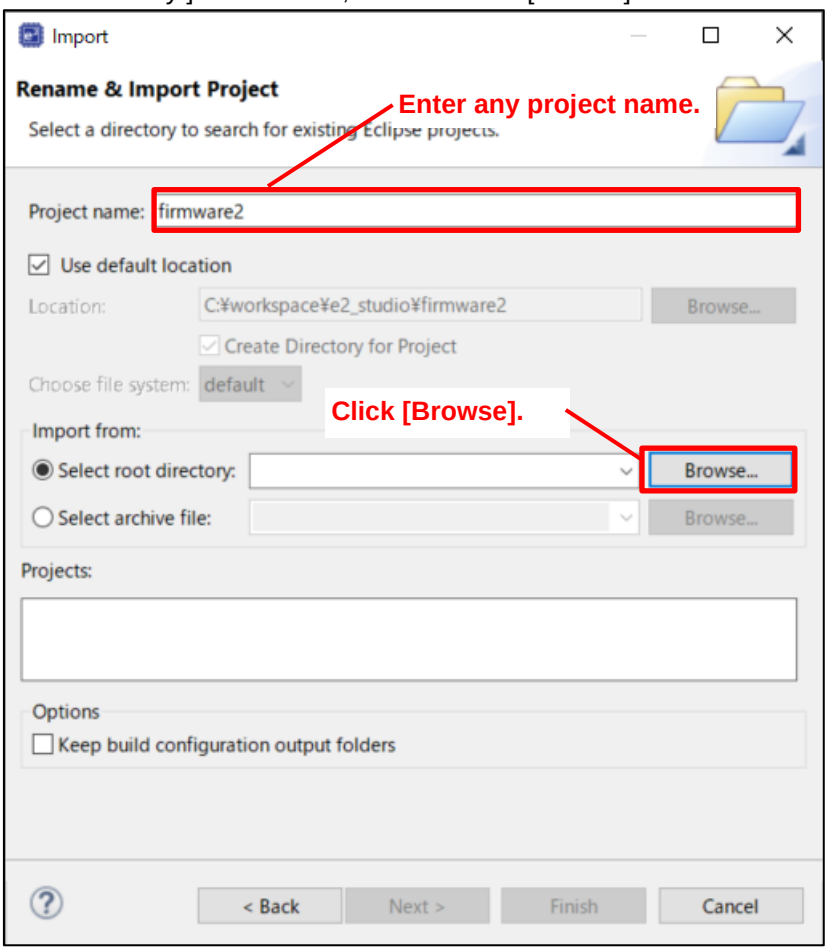
(1) Start e^2 studio.
(2) Click the [File] tab, and then click [Import].



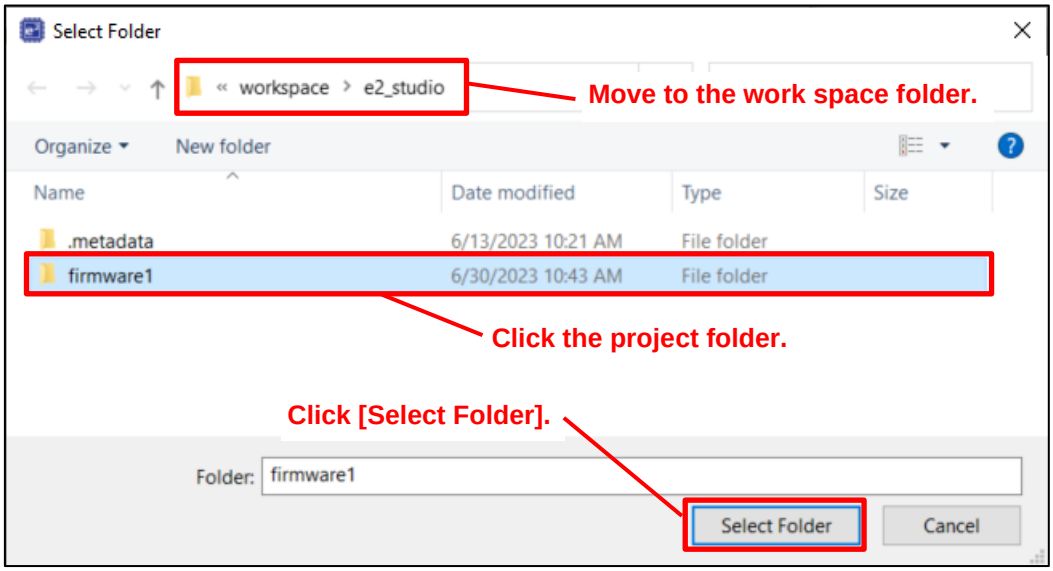
(3) In the [General] folder, click "Rename & Import Existing C/C++ Project into Workspace".
(4) Click [Next].



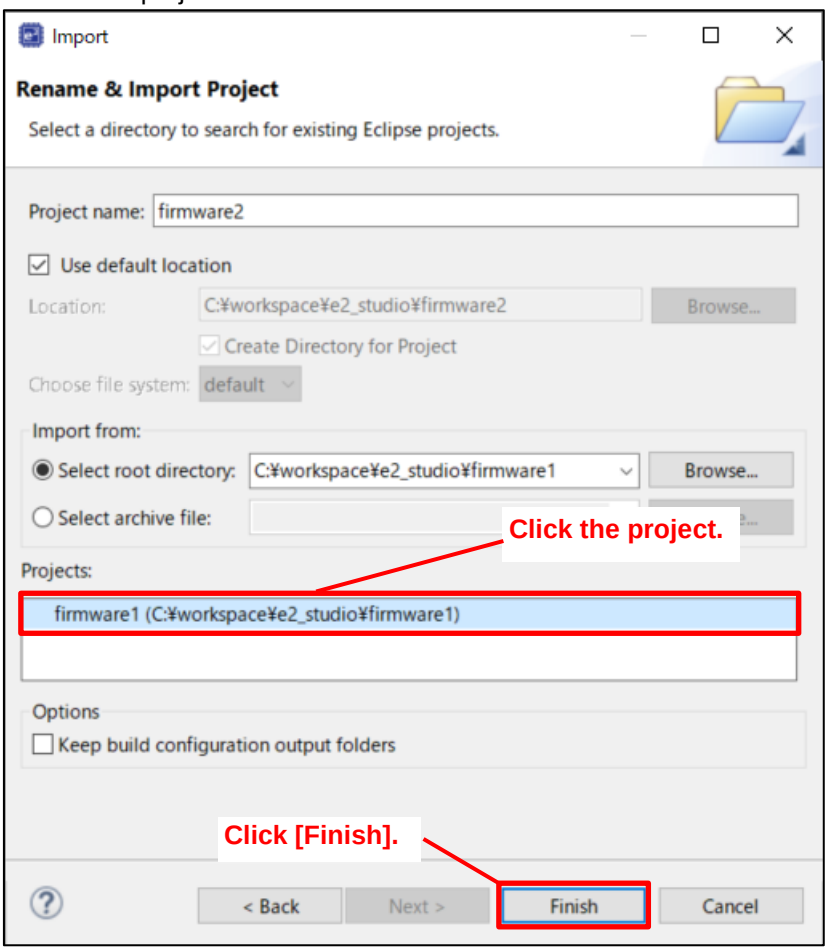
(5) Enter any project name in the [Project name:] text box.
(6) Select the [Select root directory:] radio button, and then click [Browse].



(7) Move to the e^2 studio work space folder.
(8) Click the project folder of the user program to be used as the base program.
(9) Click [Select Folder].



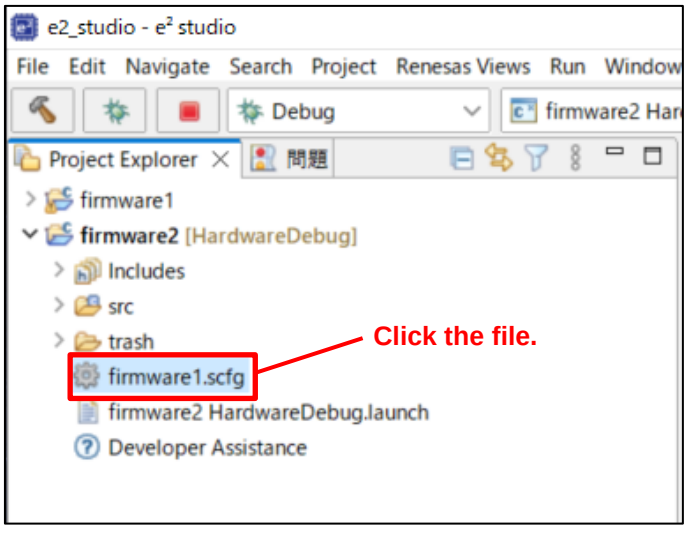
(10) In the [Projects:] field, click the project you selected in step (8).
(11) Click [Finish] to create the project.



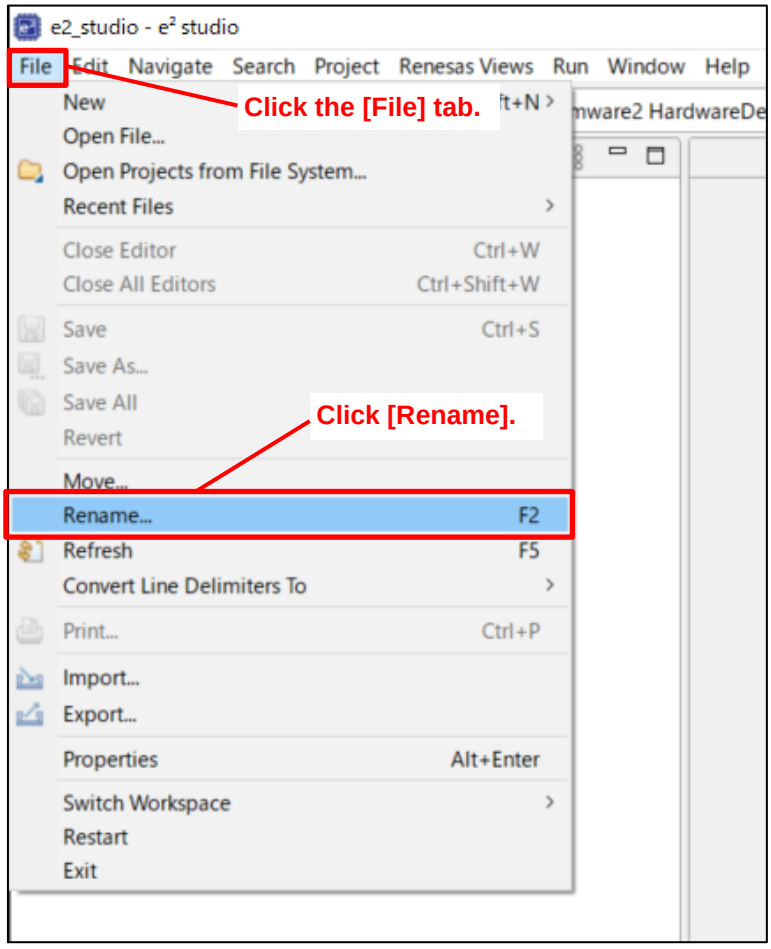
The Smart Configurator file name in the created project is not automatically changed, and the file name of the original project is retained. The following describes how to rename the file.

Note that using the file name of the original project does not affect operation. The following procedure is optional.

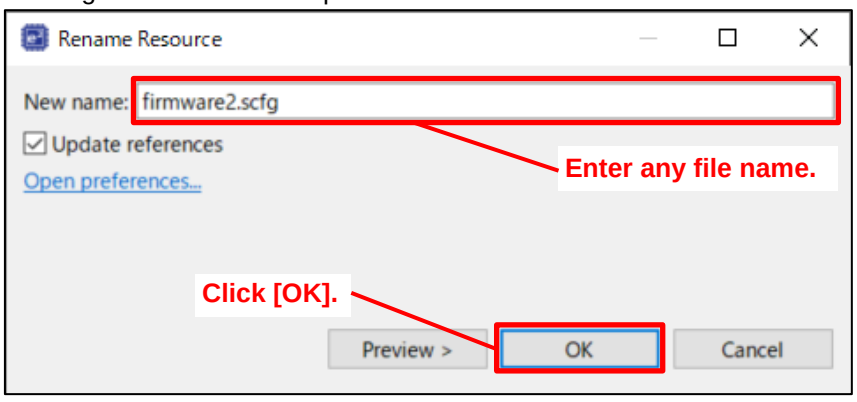
(12) Click the file you want to rename.



(13) Click the [File] tab, and then click [Rename].



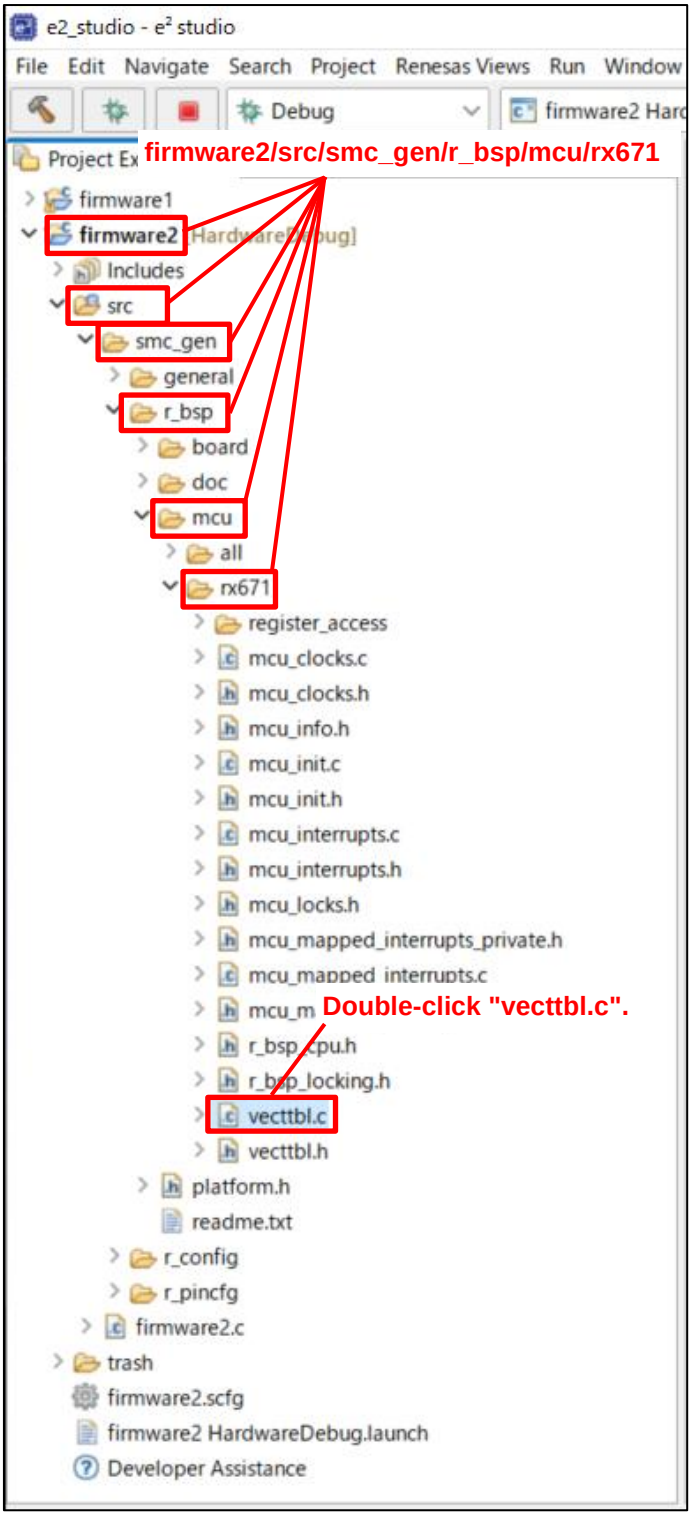
(14) Enter any file name in the [New name:] text box.
(15) Click [OK]. Renaming the file is now complete.



3.2.2 Procedure for Setting Up BSP

You need to disable the code of the option-setting memory in vecttbl.c to prevent the setting of the BANKSEL.BANKSWP[2:0] bits from being output to MOT files.

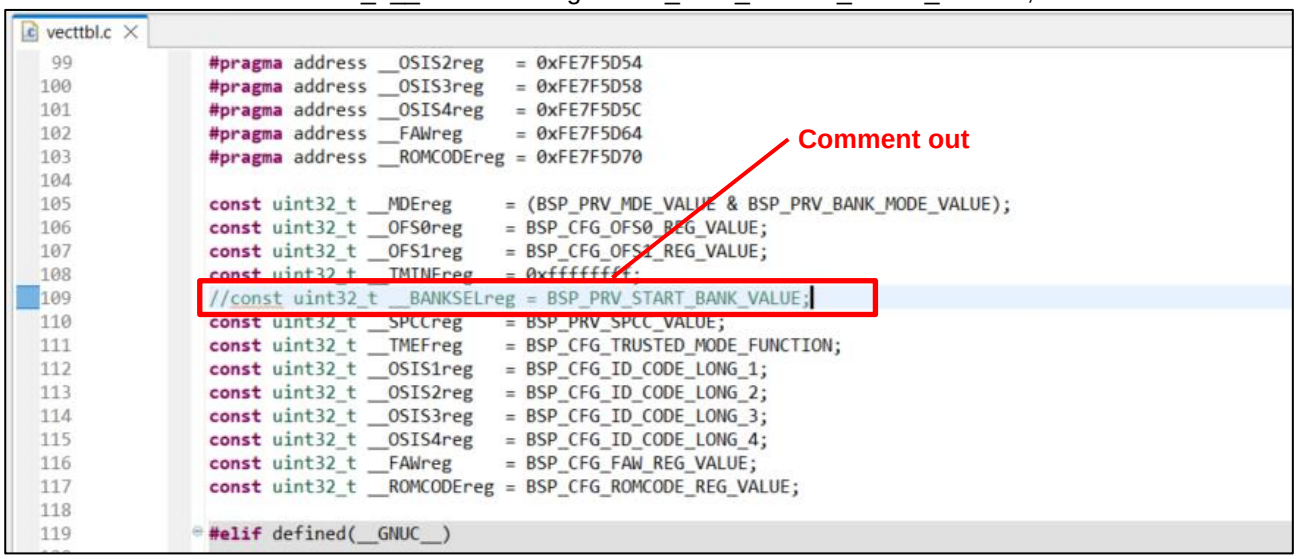
(1) In Project Explorer, in the path "firmware2/src/smc_gen/r_bsp/mcu/rx671", double-click "vecttbl.c". (In the pathname, "firmware2" indicates the project name and "rx671" indicates the MCU being used.)



(2) Change the code in vecttbl.c. The position to be changed in the code varies depending on the compiler being used.

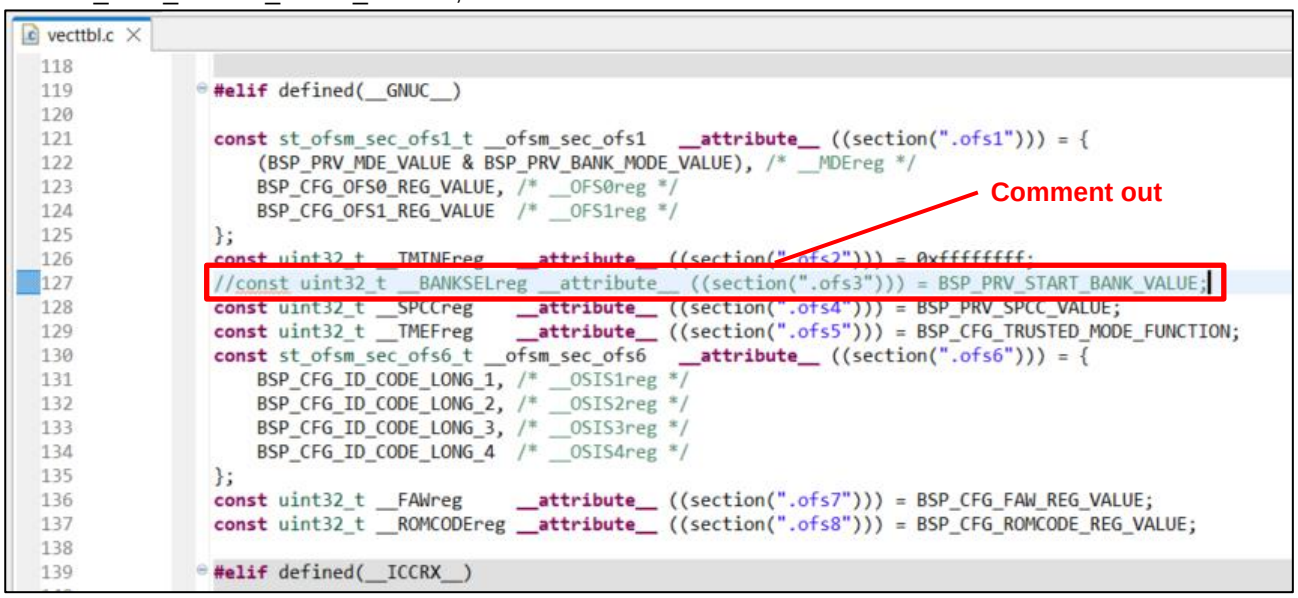
• For Renesas CC-RX

Comment out "const uint32_t __BANKSELreg = BSP_PRV_START_BANK_VALUE;" on line 109.



• For GCC for Renesas RX

Comment out "const uint32_t __BANKSELreg __attribute__ ((section(".ofs3"))) = BSP_PRV_START_BANK_VALUE;" on line 127.



(3) Perform a build. Setting up BSP is now complete.

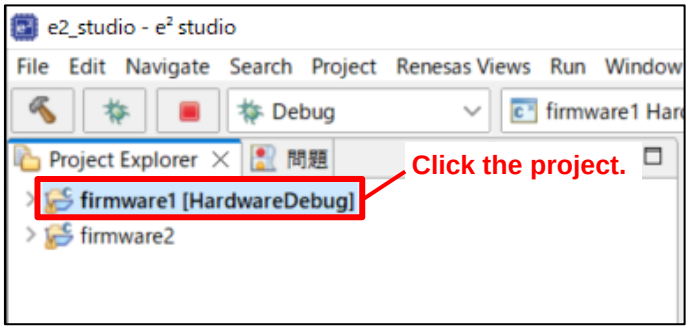
3.2.3 Procedure for Setting the Debug Configuration

The procedure is the same as section 3.1.3, Procedure for Setting the Debug Configuration.

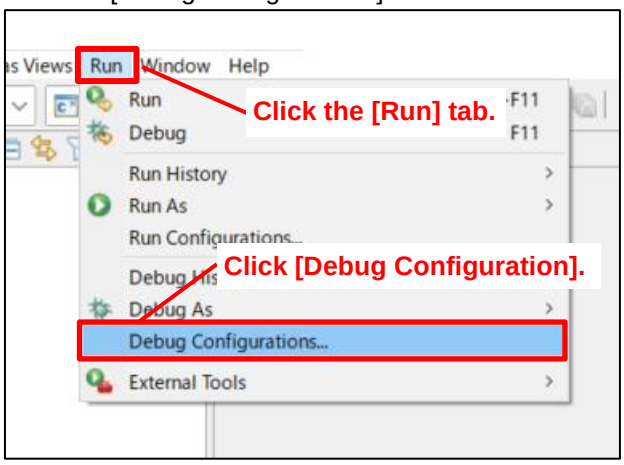
3.3 Debugging the Startup Bank Switching Function

The following describes an example of the procedure for downloading user program 1 to bank 0 and user program 2 to bank 1, and then debugging the program when switching from bank 0 to bank 1.

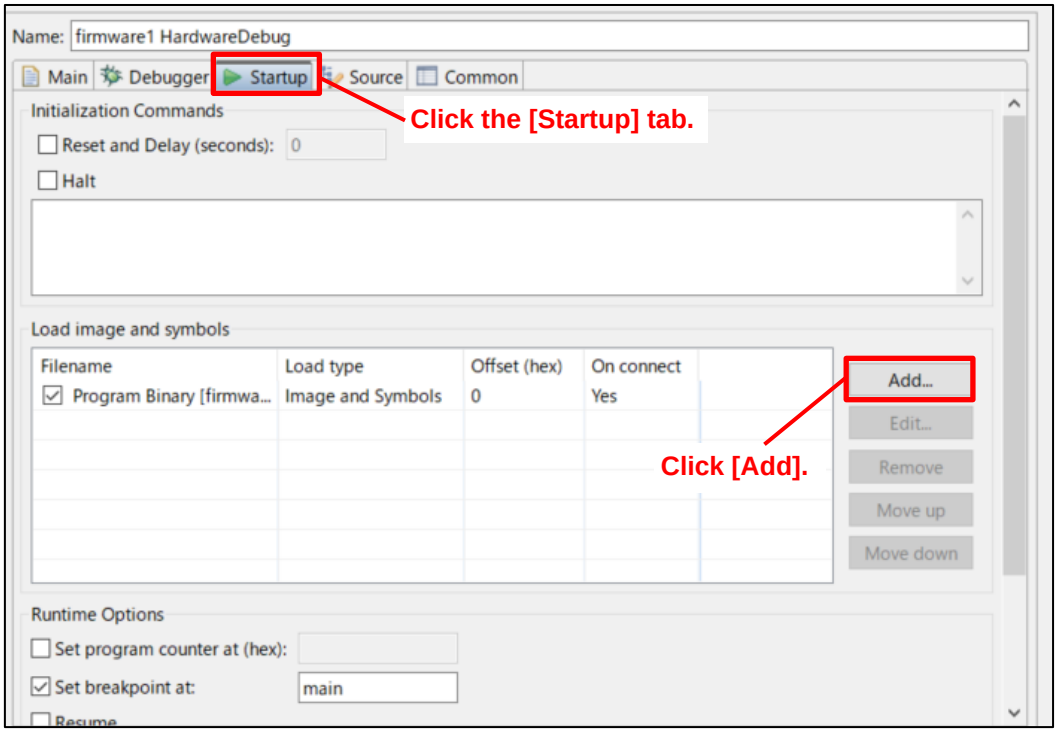
(1) In Project Explorer, click the project of user program 1 to be downloaded to bank 0.



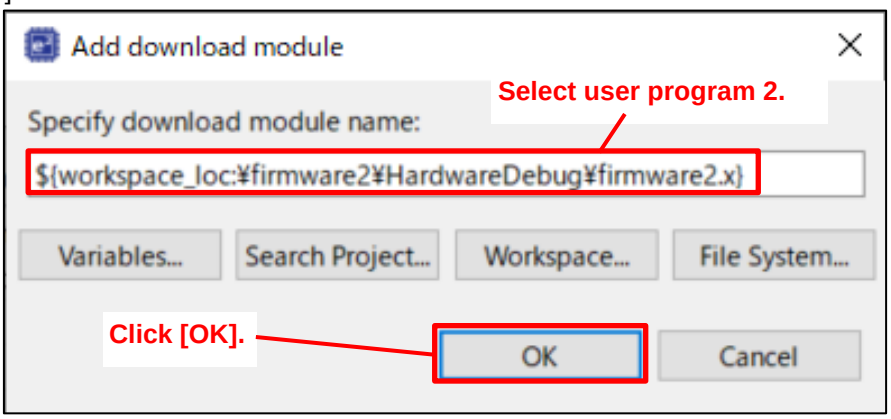
(2) Click the [Run] tab, and then click [Debug Configurations].



(3) Open the [Startup] tab, and then click [Add].

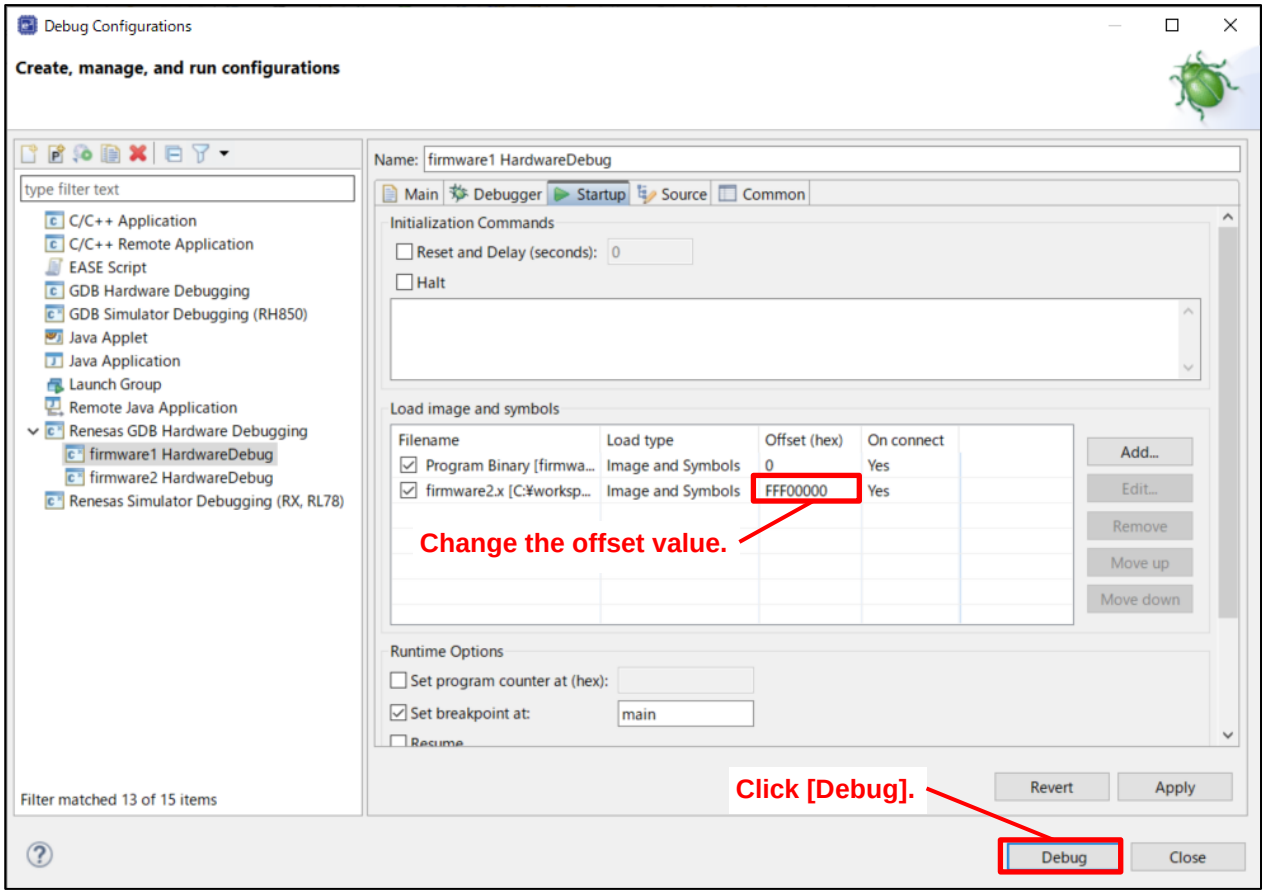


(4) In the [Specify download module name:] field, select user program 2 to be downloaded to bank 1, and then click [OK].

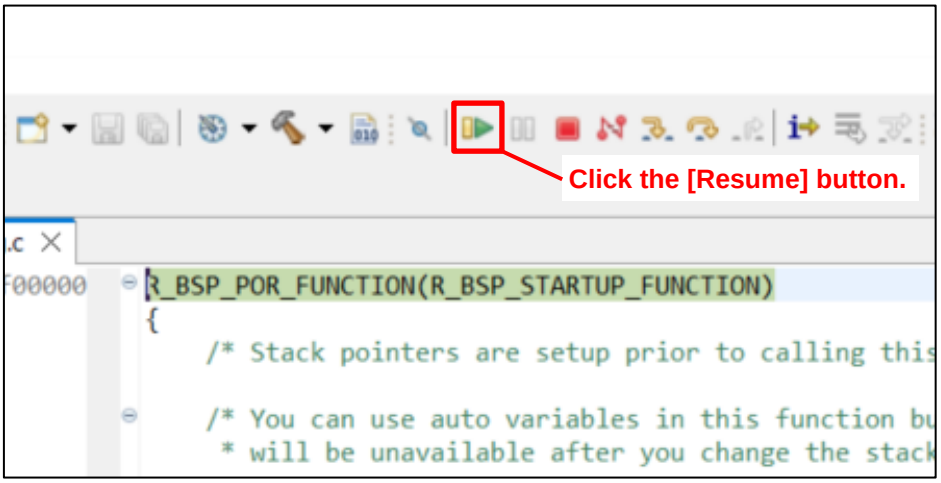


(5) Added user program 2 is developed in bank 0 (FFF0 0000h to FFFF FFFFh) and downloading it to bank 1 (FFE0 0000h to FFEF FFFFh) requires the [Offset (hex)] value to be changed. The value specified here is the offset value, not the address value. For 2-MB products of the RX671 Group, specify -100000H as the offset from the address. Because negative values cannot be entered, enter FFF00000H, which is two's complement.

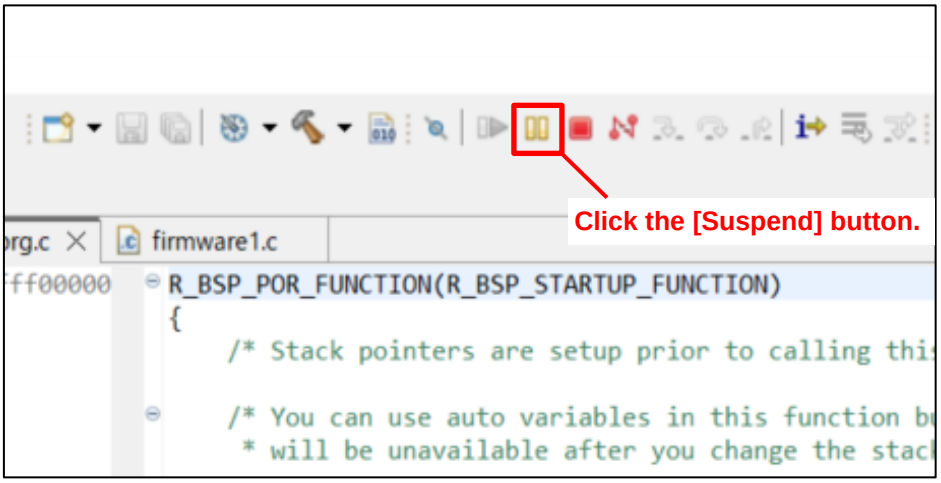
(6) Connect the MCU, and then click [Debug].



(7) After writing is complete, click the [Resume] button.

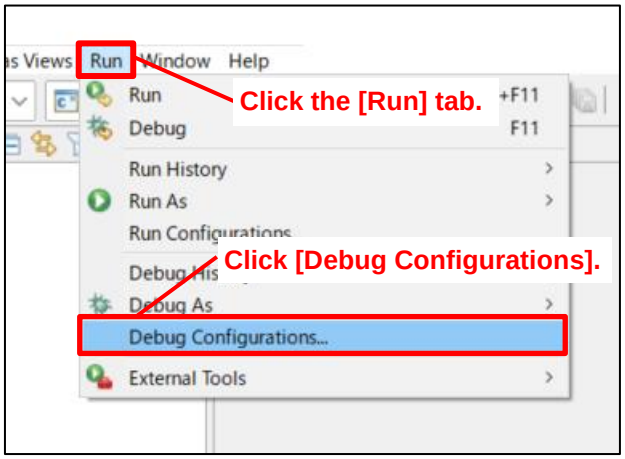


(8) After operation confirmation of user program 1 in bank 0 completes, click the [Suspend] button.

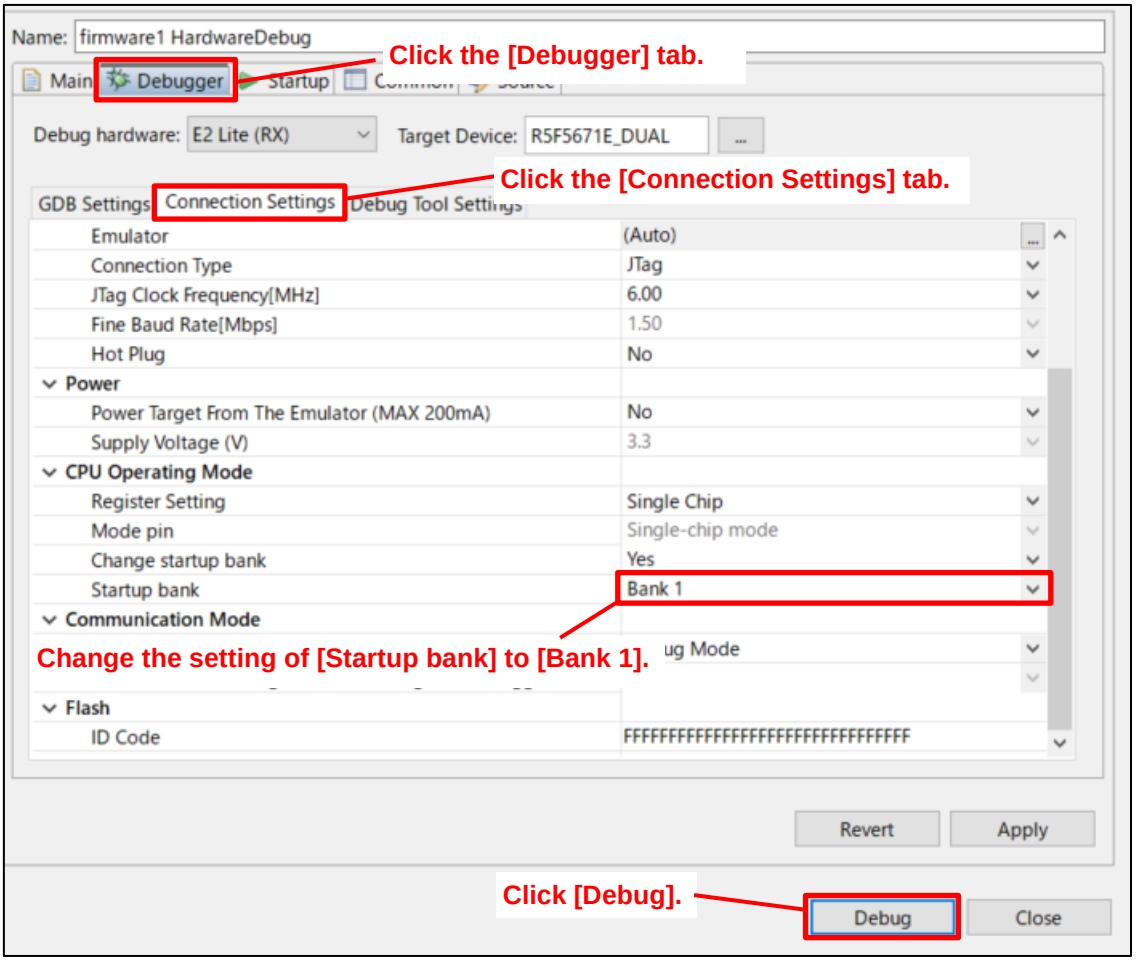


The following describes the procedure for switching the startup bank to bank 1 to confirm the operation of user program 2.

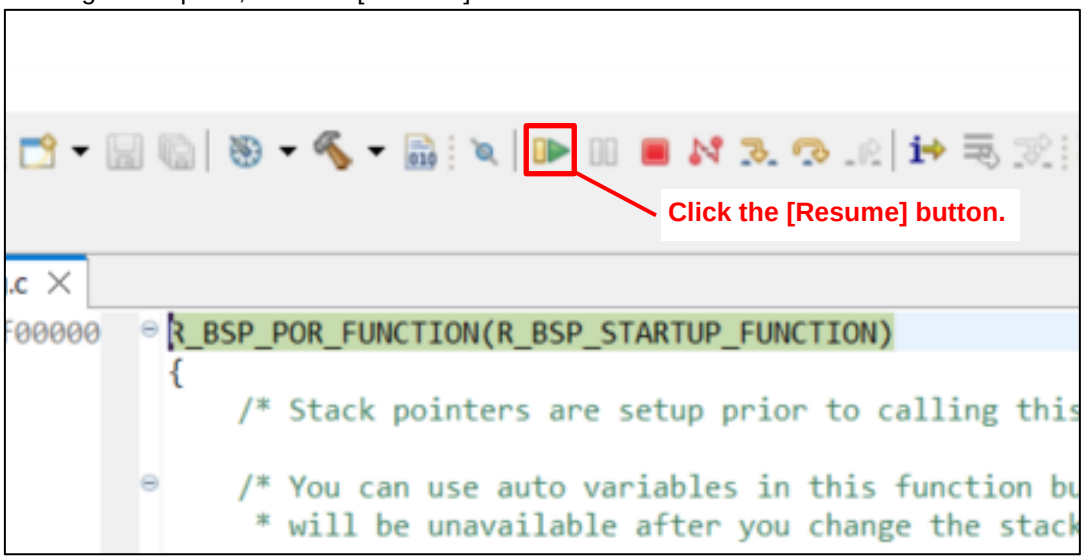
(9) Again, click the [Run] tab, and then click [Debug Configurations].



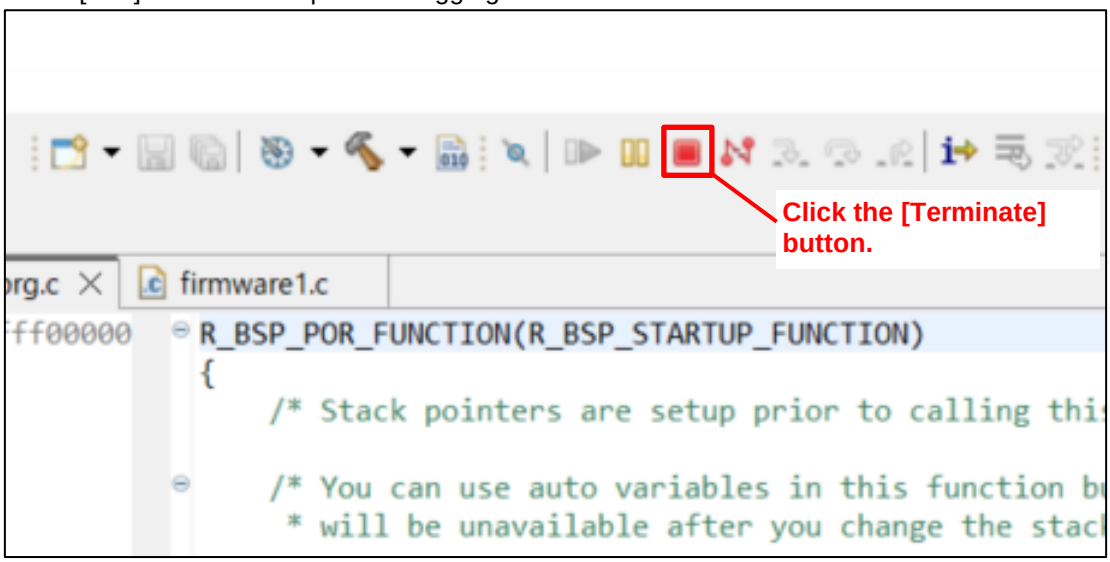
(10) In the [Debugger] tabbed page, click the [Connection Settings] tab.
(11) Change the setting of [Startup bank] to [Bank 1].
(12) Click [Debug].



(13) After writing is complete, click the [Resume] button.



(14) Confirm that user program 2 is operating in bank 1.
(15) Click the [End] button to complete debugging.



3.4 Outputting User Program MOT Files

This section describes how to output MOT files in e² studio.

The method varies depending on whether an offset exists. To output normal MOT files, see section 3.4.1, Procedures for Outputting MOT Files. To output offset MOT files, see section 3.4.2, Procedures for Outputting Offset MOT Files.

3.4.1 Procedures for Outputting MOT Files

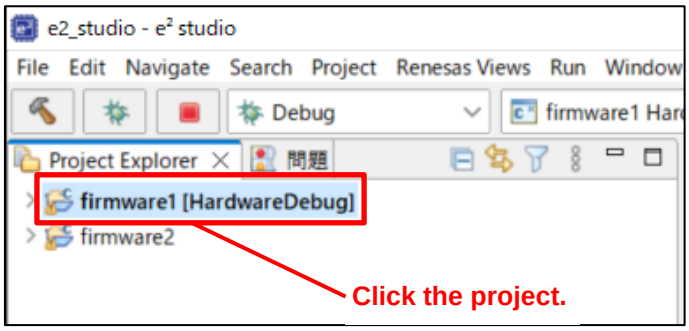
The following describes how to output a normal MOT file.

The method varies depending on the compiler being used. If you are using Renesas CC-RX, see section 3.4.1.1, When Using Renesas Electronics C/C++ Compiler Package for RX Family. If you are using GCC for Renesas RX, see section 3.4.1.2, When Using GCC for Renesas RX.

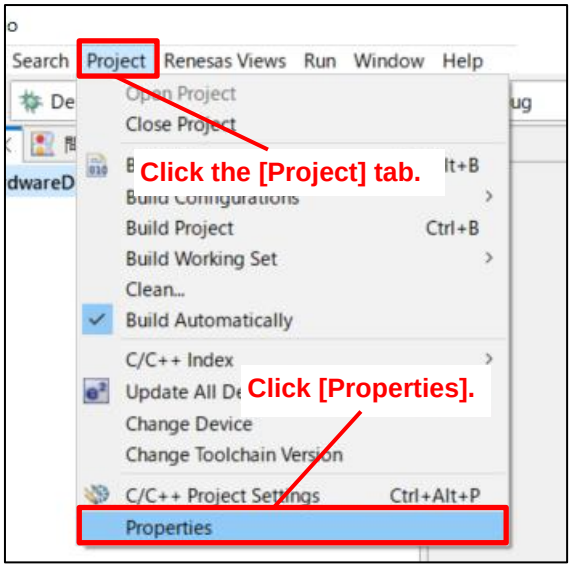
3.4.1.1 When Using Renesas Electronics C/C++ Compiler Package for RX Family

(1) Start e² studio.

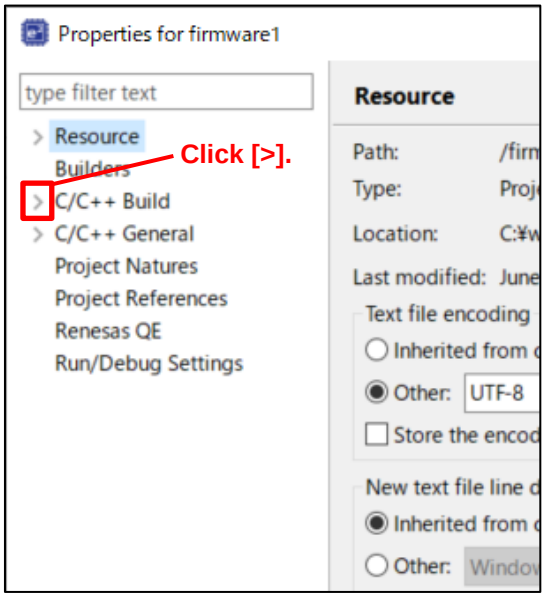
(2) In Project Explorer, click the project for which you want to output the file.



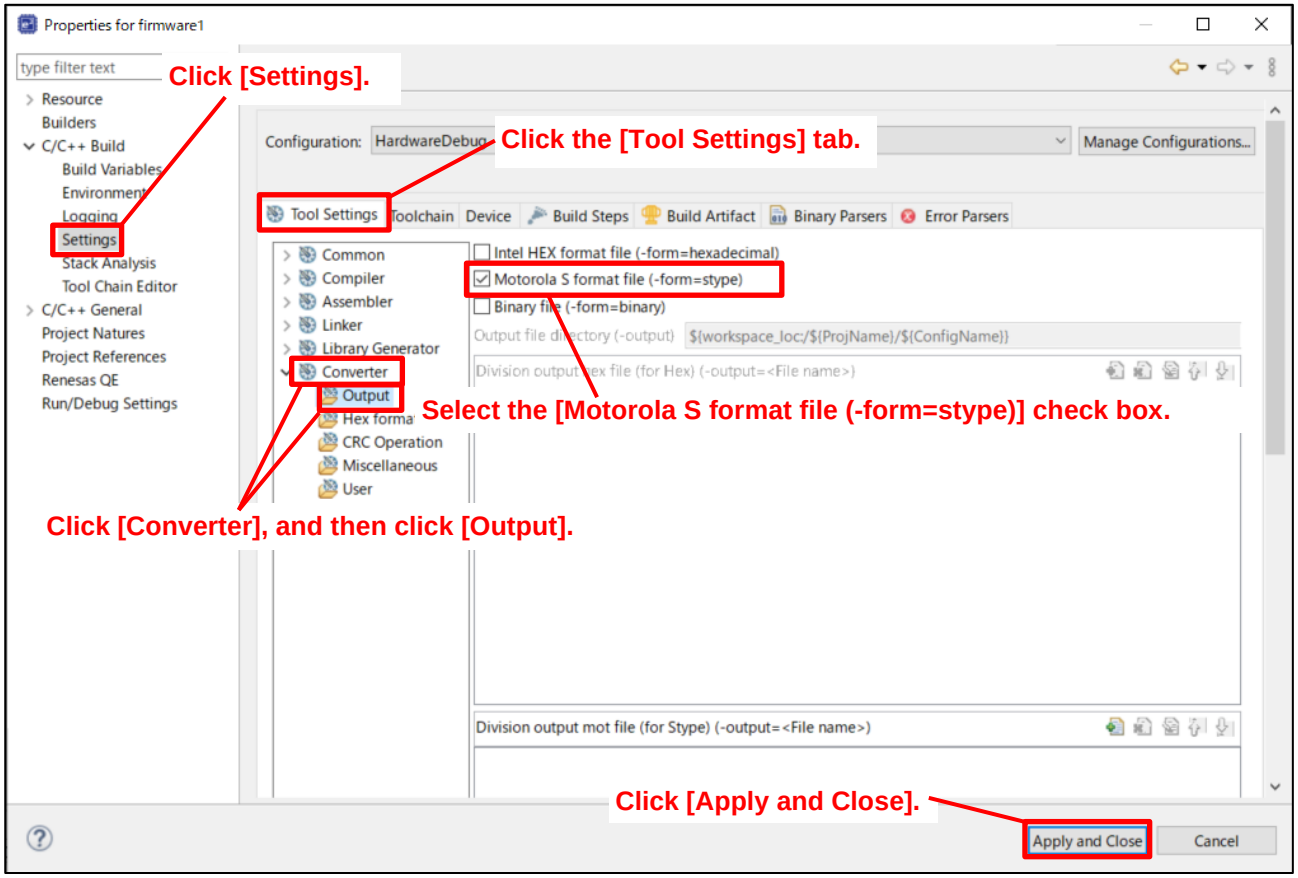
(3) Click [Project], and then click [Properties].



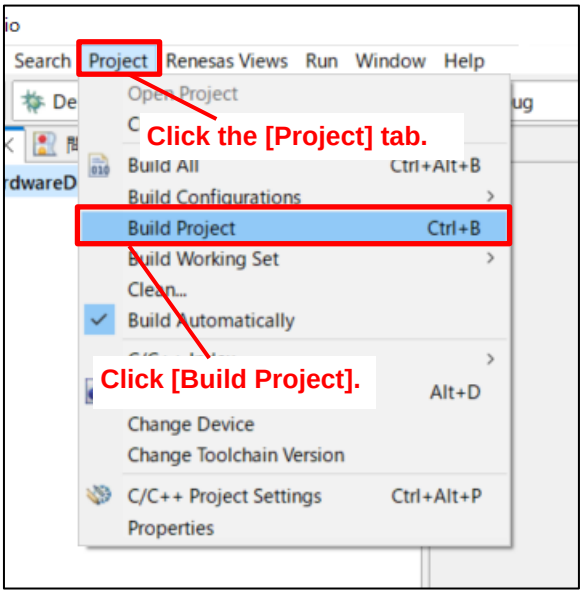
(4) Click [>] for [C/C++ Build].



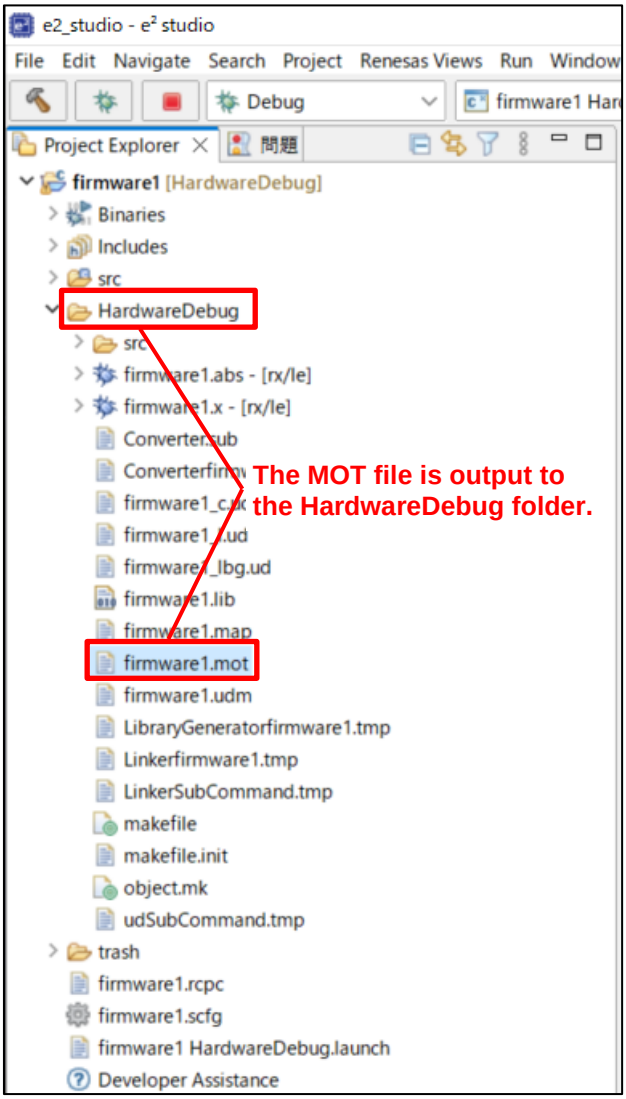
(5) Click [Settings].
(6) Click the [Tool Settings] tab.
(7) Click [Converter], and then click [Output].
(8) Select the [Motorola S format file (-form=stype)] check box.
(9) Click [Apply and Close].



(10) Click [Project], and then click [Build Project].

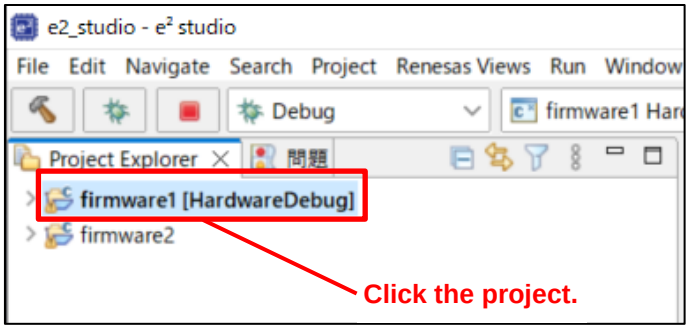


(11) After the build is complete, the MOT file is output to the HardwareDebug folder in the project.

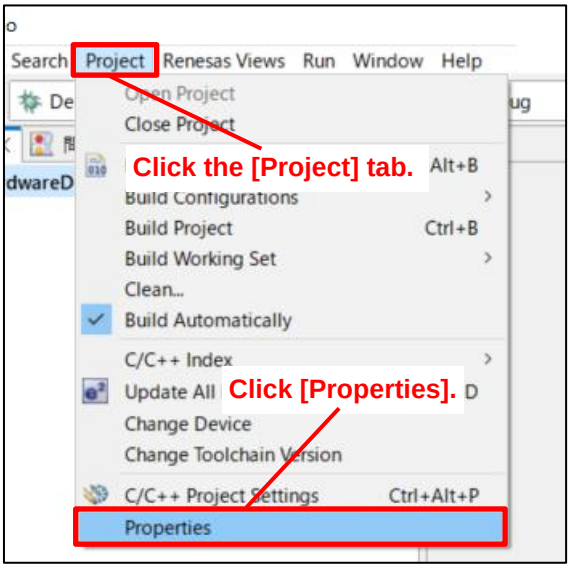


3.4.1.2 When Using GCC for Renesas RX

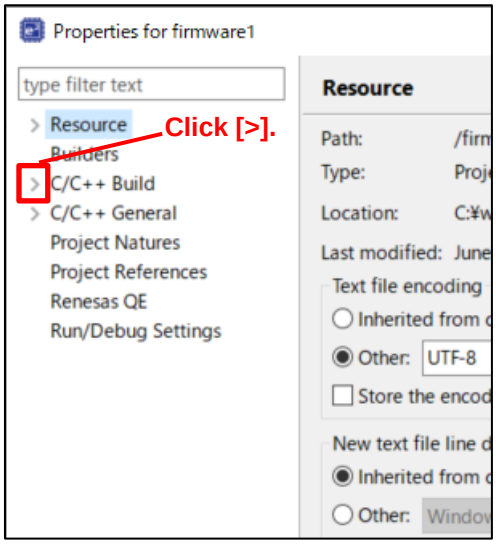
(1) Start e² studio.
(2) In Project Explorer, click the project of which you want to change the setting.



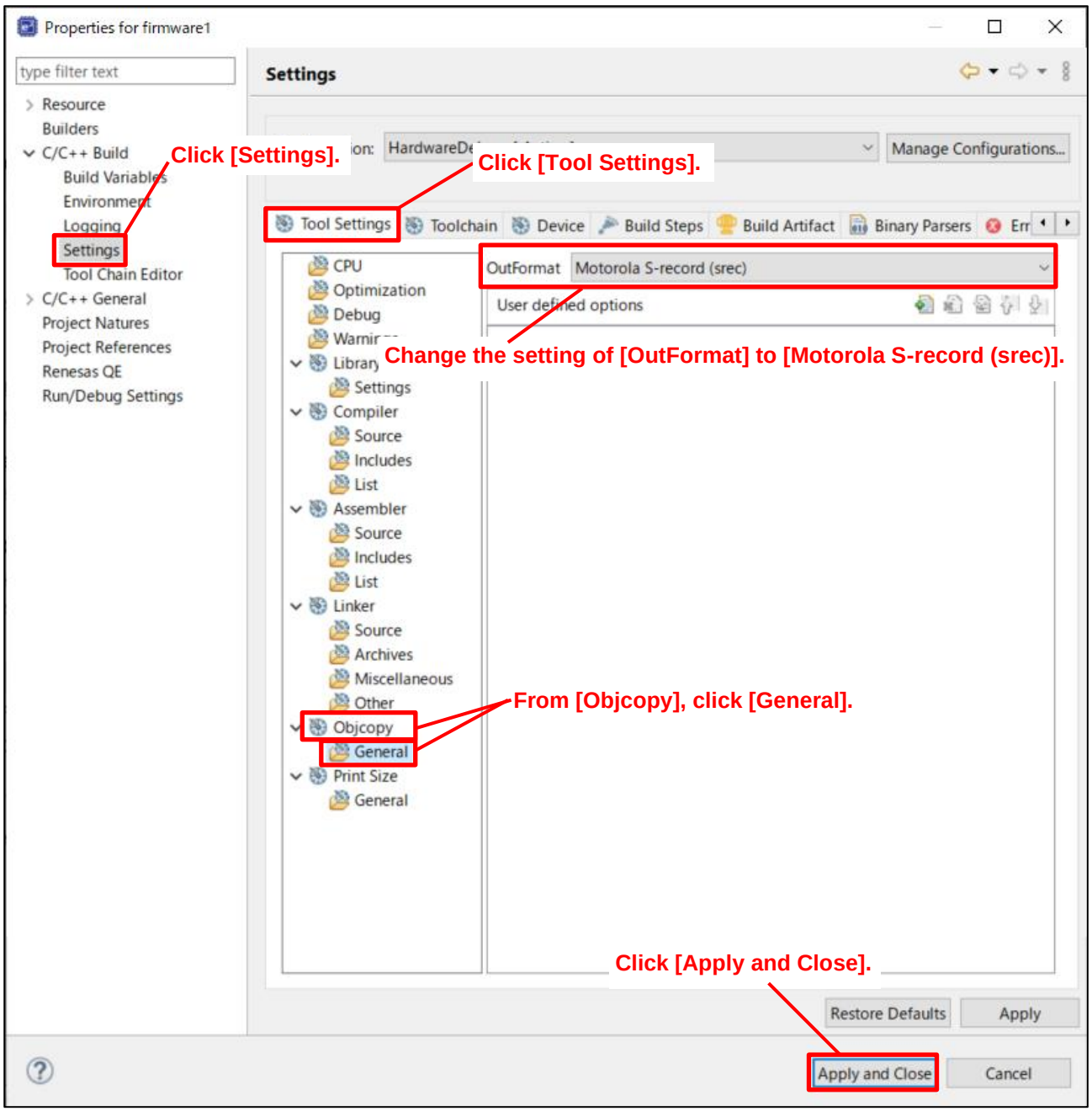
(3) Click [Project], and then click [Properties].



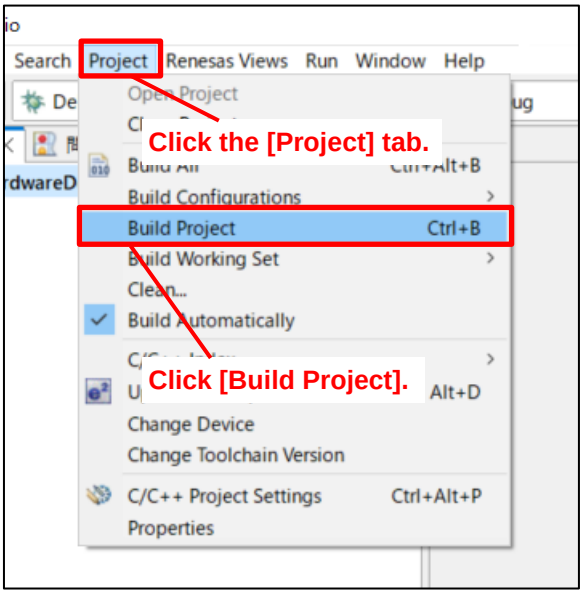
(4) Click [>] for [C/C++ Build] to expand the pull-down menu.



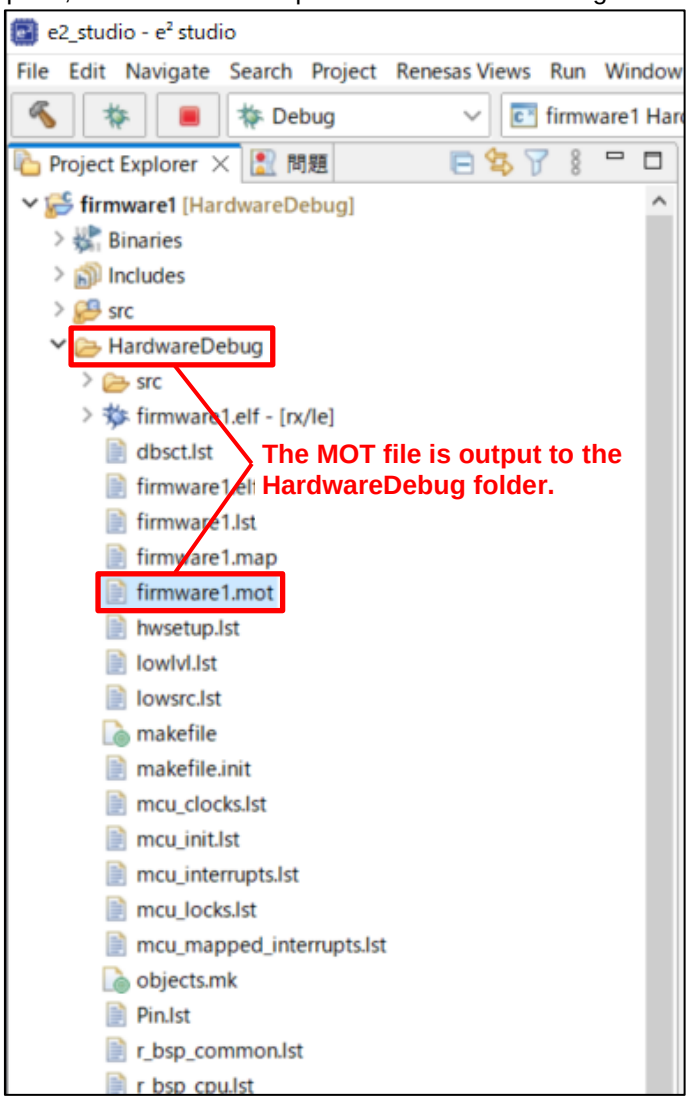
(5) Click [Settings].
(6) Click the [Tool Settings] tab.
(7) From [Objcopy], click [General].
(8) Change the setting of [OutFormat] to [Motorola S-record (srec)].
(9) Click [Apply and Close].



(10) Click [Project], and then click [Build Project].



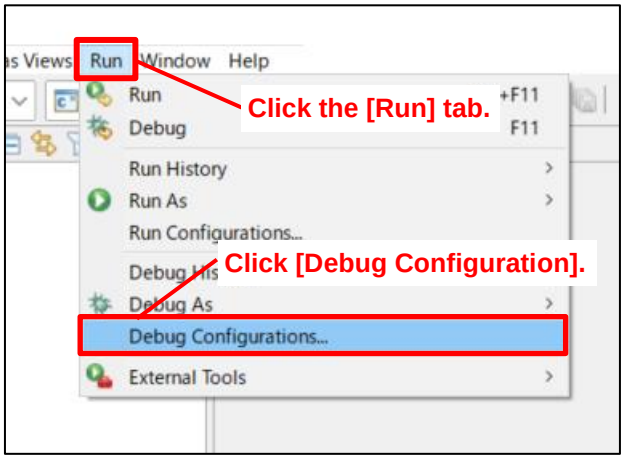
(11) After the build is complete, the MOT file is output to the HardwareDebug folder in the project.



3.4.2 Procedures for Outputting Offset MOT Files

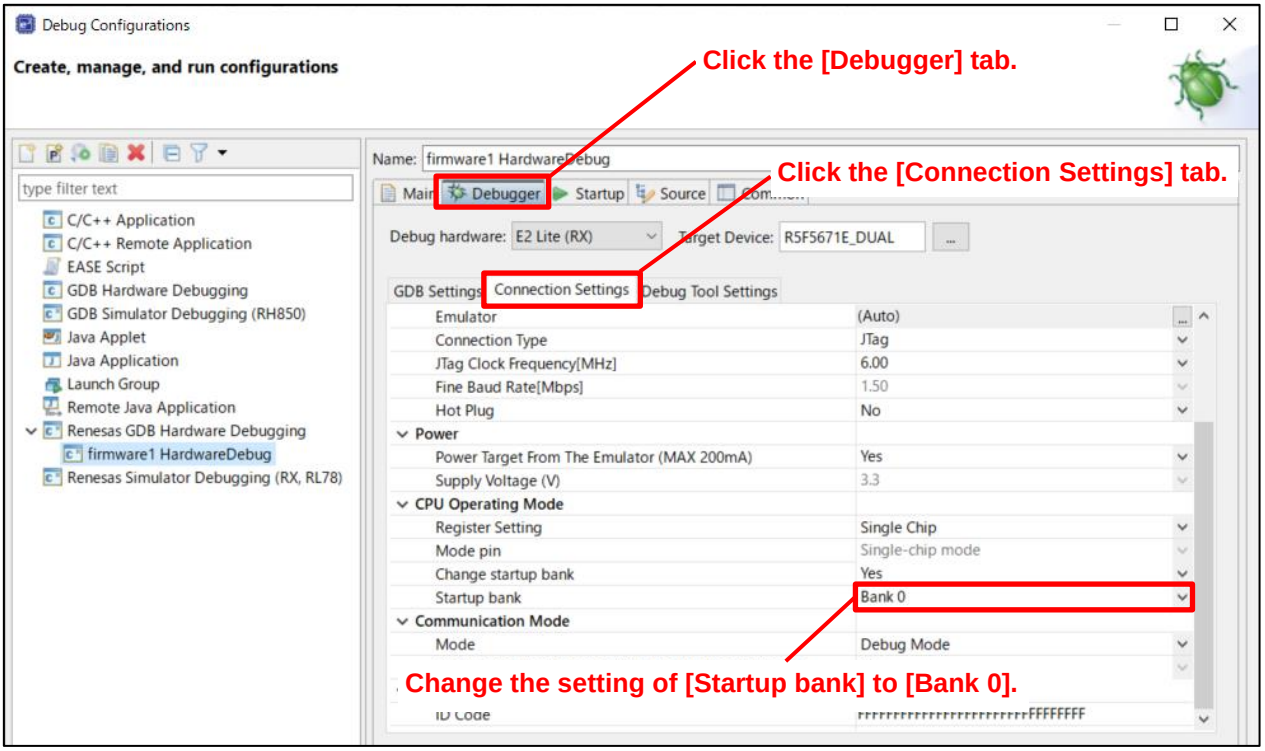
(1) Start e^2 studio.

(2) Click the [Run] tab, and then click [Debug Configurations].



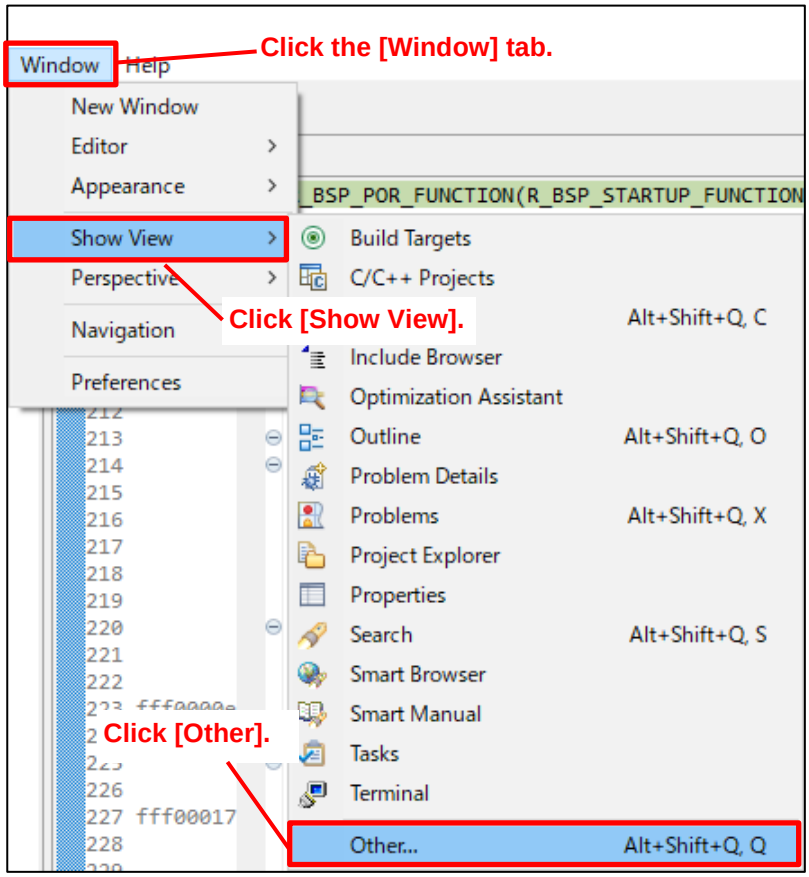
(3) In the [Debugger] tabbed page, click the [Connection Settings] tab.

(4) Change the setting of [Startup bank] to [Bank 0].

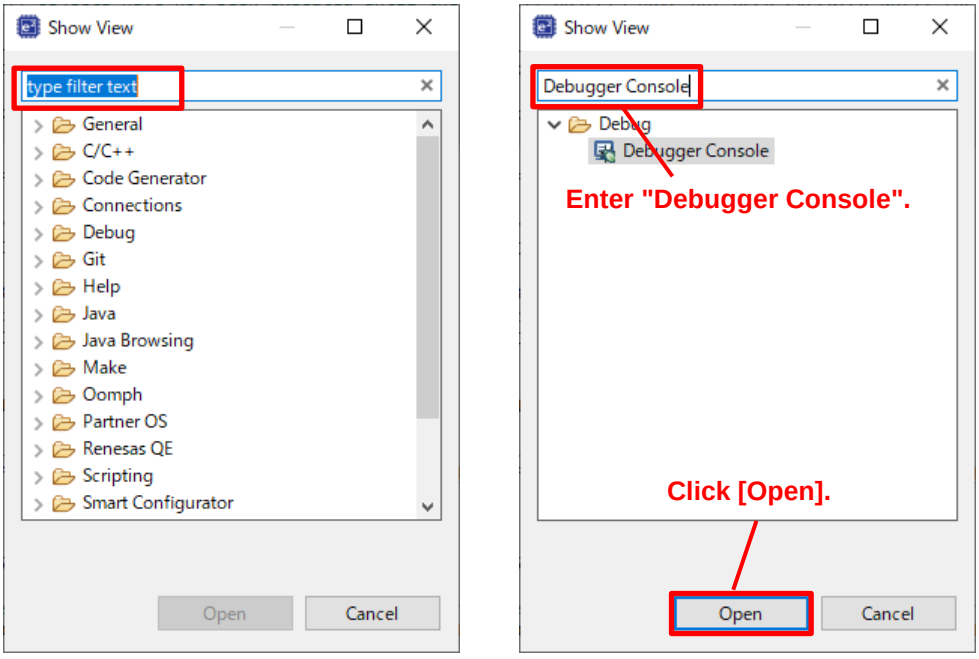


(5) Add an offset to the program located at addresses FFF0 0000h to FFFF FFFFh, and then download the program to addresses FFE0 0000h to FFEF FFFFh. For details, see section 3.3, Debugging the Startup Bank Switching Function, steps (3) to (6).

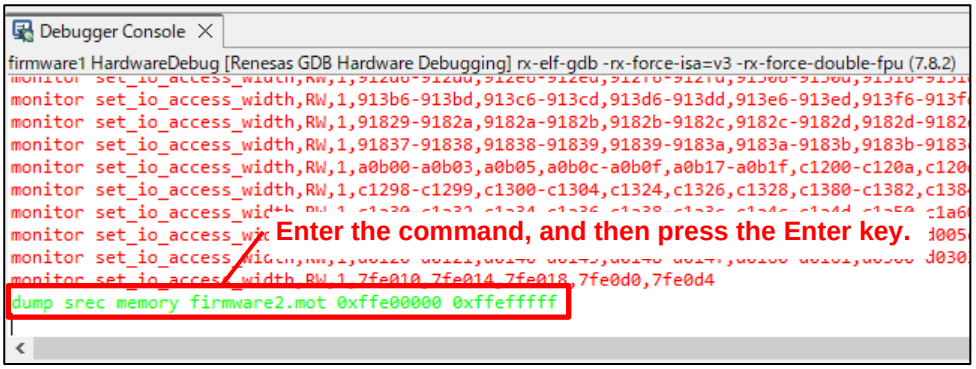
(6) Click [Window], [Show View], and then [Other].



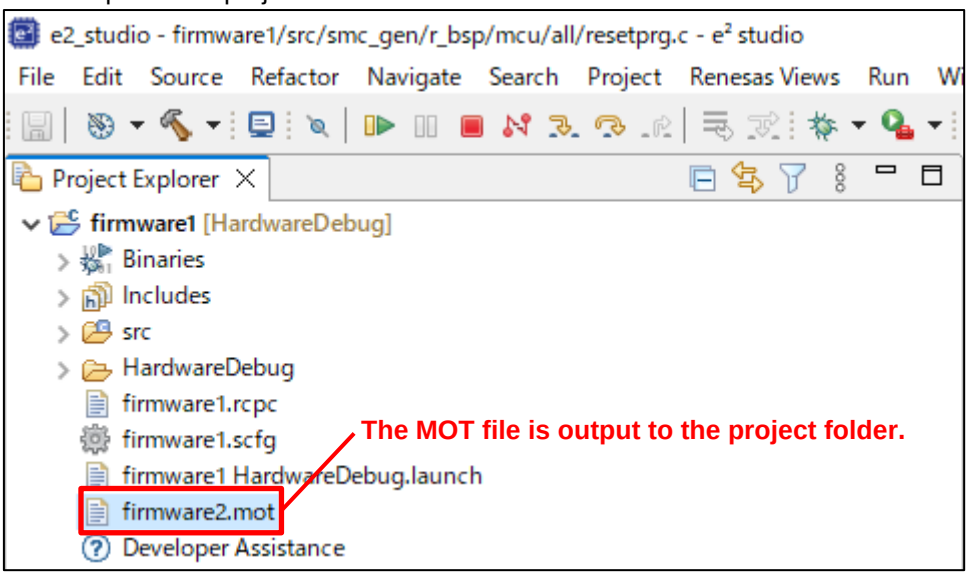
(7) In the [Show View] dialog box, enter "Debugger Console", and then click [Open].



(8) Run the following GDB command to output the MOT file of the program that was downloaded to addresses FFE0 0000h to FFEF FFFFh with offset added. Enter any file name for the MOT file.
Command: dump srec memory firmware2.mot 0xffe00000 0xffefffff



(9) The MOT file is output to the project folder.

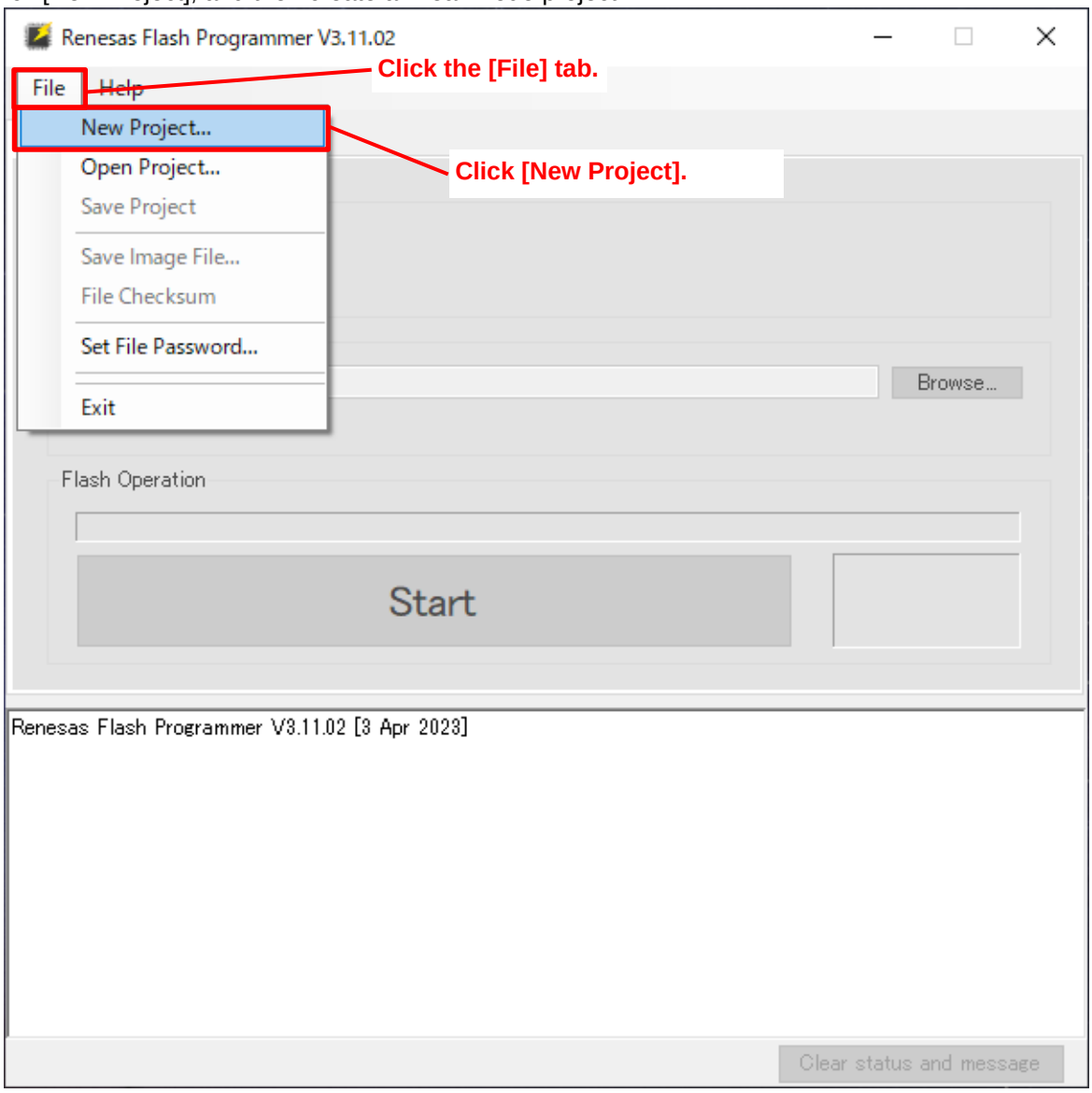


4. Writing a Dual-Mode Program by Using RFP

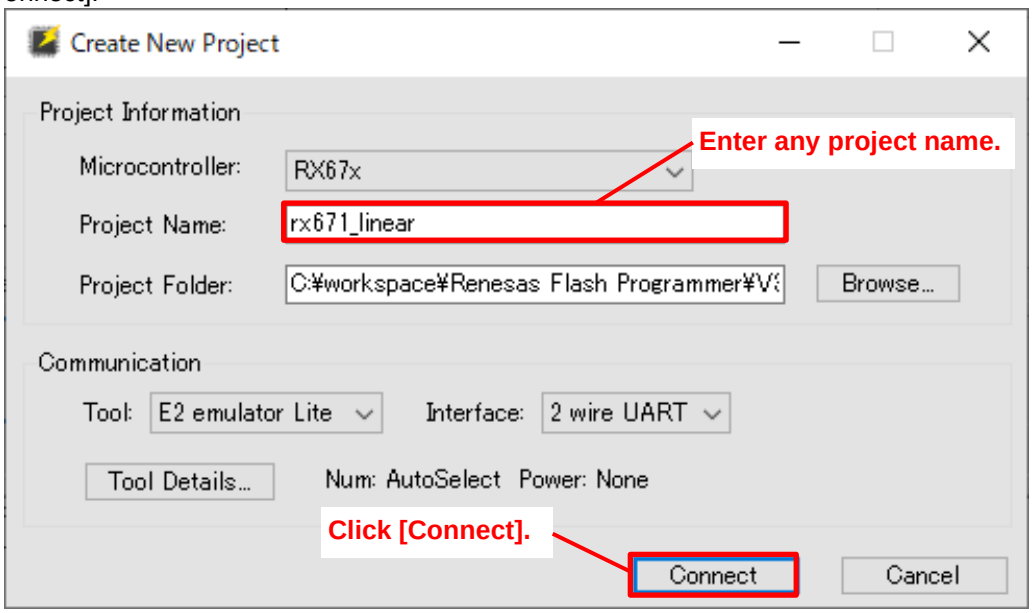
4.1 Writing a Program in Bank 0 of the MCU in Shipment State

Prepare a MOT file (created in section 3.1, When Developing User Program Version 1.00) that contains a program to be written to bank 0 and the option-setting memory settings. When the program is written by using the following procedure with RFP, dual mode is set after a reset, and then the program written in bank 0 operates.

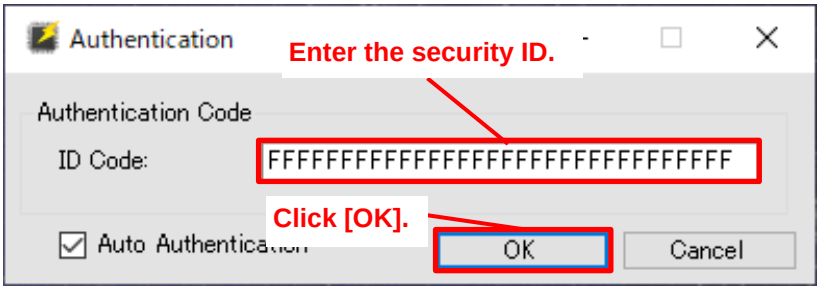
(1) Start RFP.
(2) Click the [File] tab.
(3) Click [New Project], and then create a linear mode project.



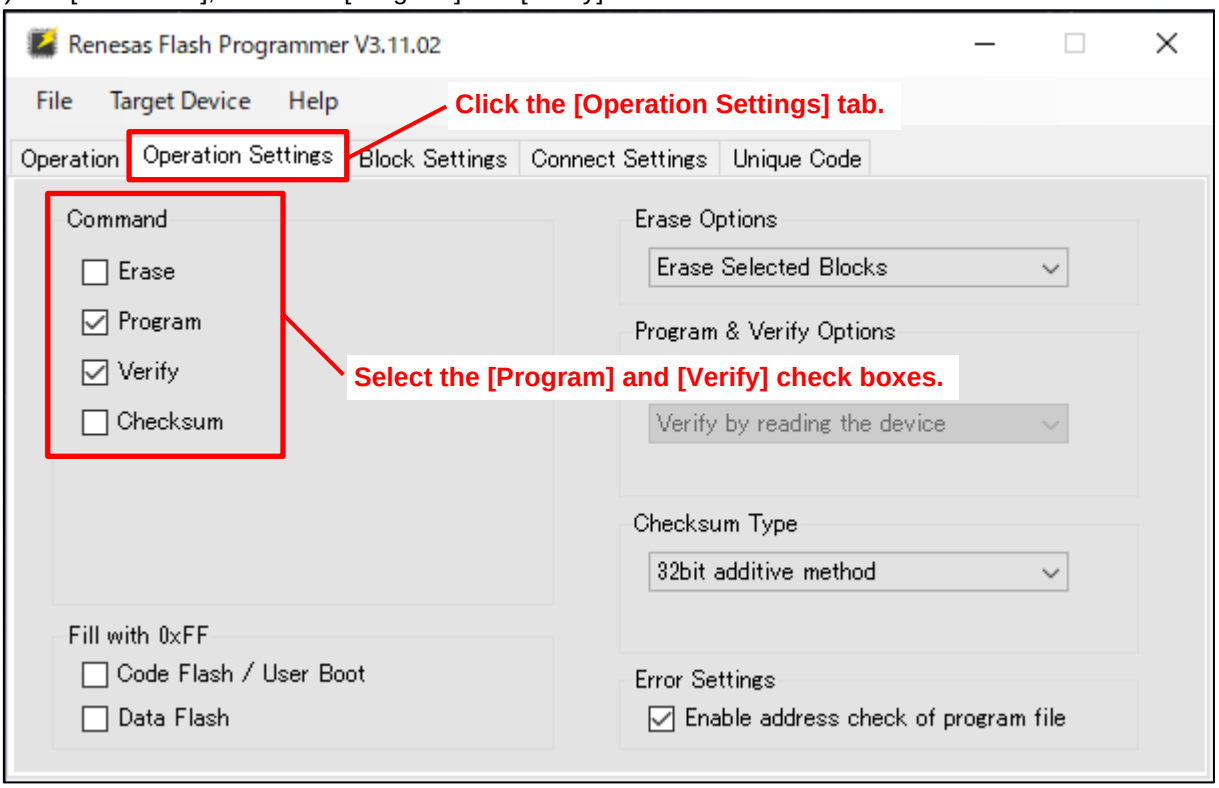
(4) Enter any project name in the [Project Name:] text box.
(5) Change other settings according to the system.
(6) Click [Connect].



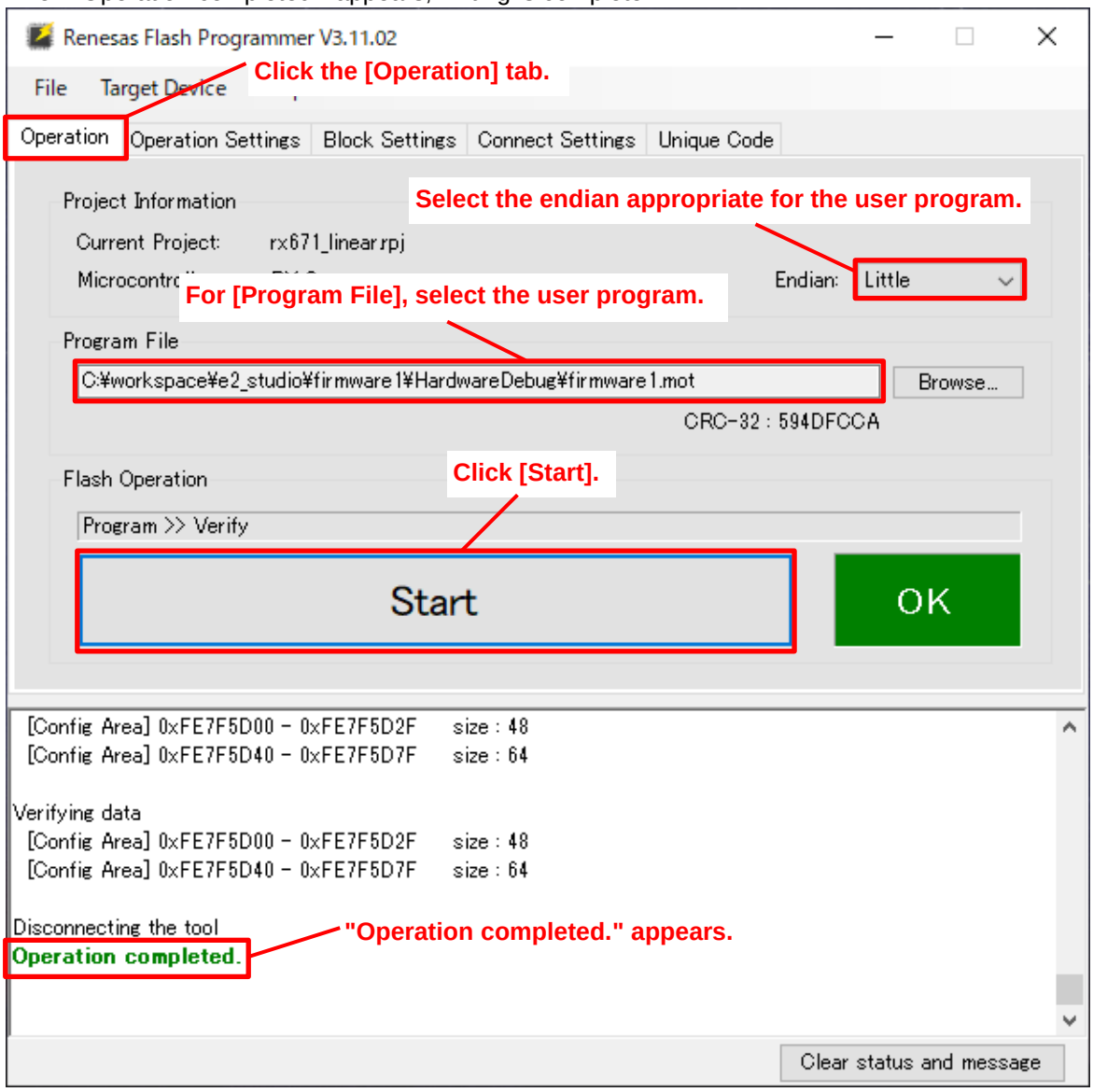
(7) In the [ID Code:] text box, enter the set security ID.
(8) Click [OK].



(9) Click the [Operation Settings] tab.
(10) For [Command], select the [Program] and [Verify] check boxes.



(11) Click the [Operation] tab.
(12) For [Endian:], select the endian appropriate for the user program.
(13) For [Program File], select the user program to be written.
(14) Click [Start].
(15) When "Operation completed." appears, writing is complete.

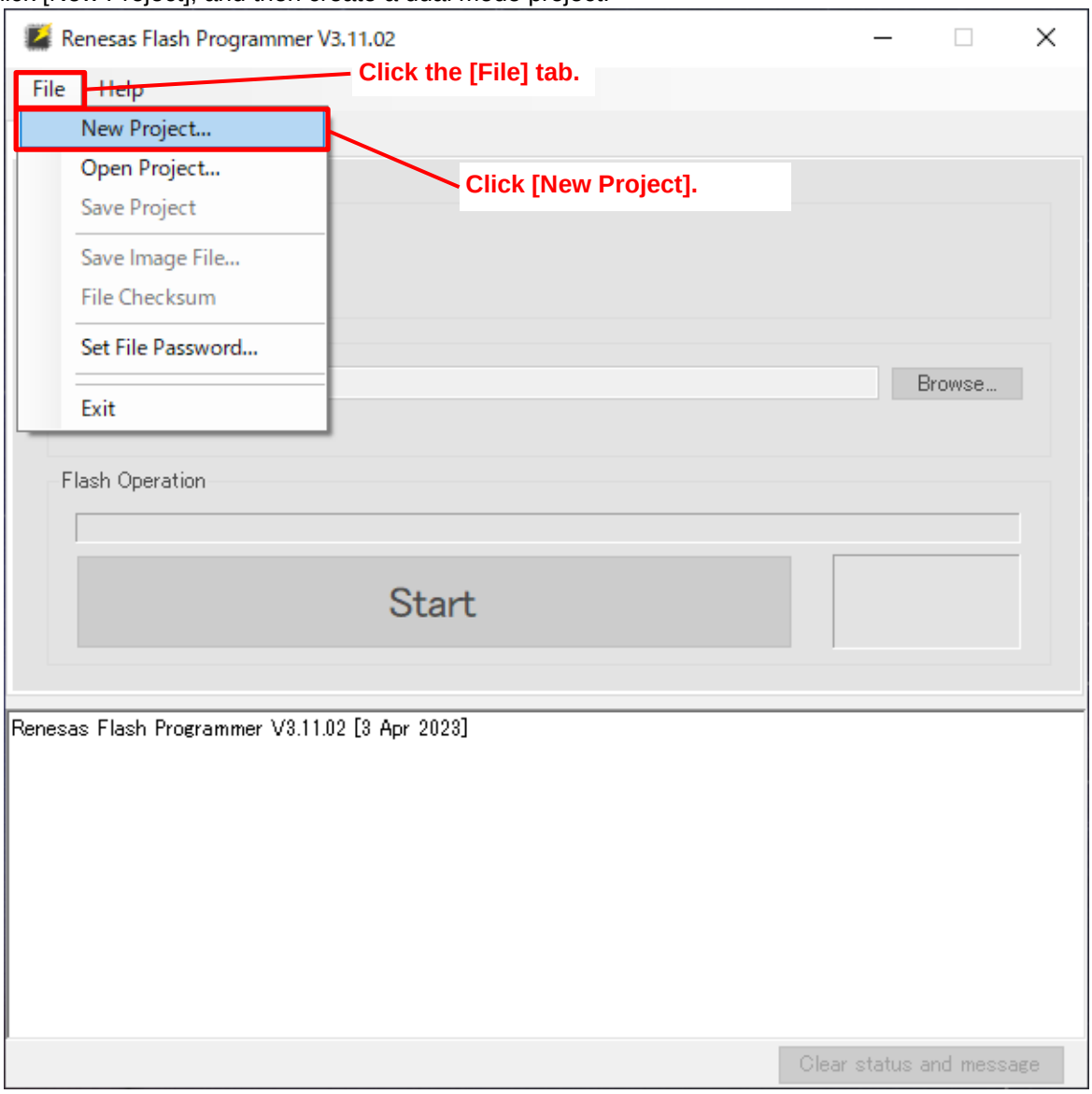


4.2 Writing in Bank 1

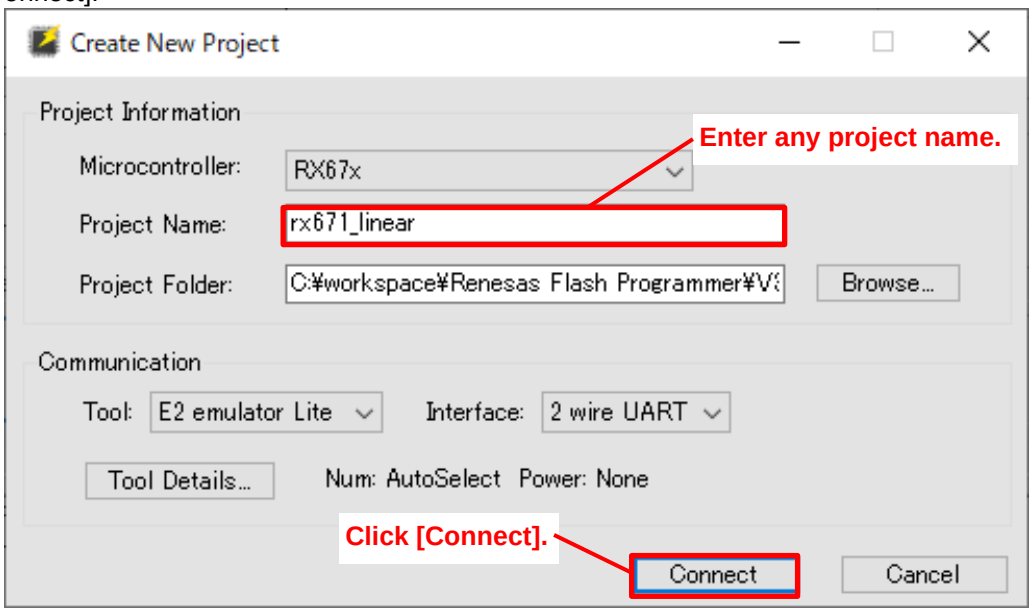
This section describes how to write a user program in bank 1 after writing the user program in bank 0 in section 4.1, Writing a Program in Bank 0 of the MCU in Shipment State.

Prepare the MOD file (created in section 3.4.2, Procedures for Outputting Offset MOT Files without option-setting memory settings) to be written to bank 1.

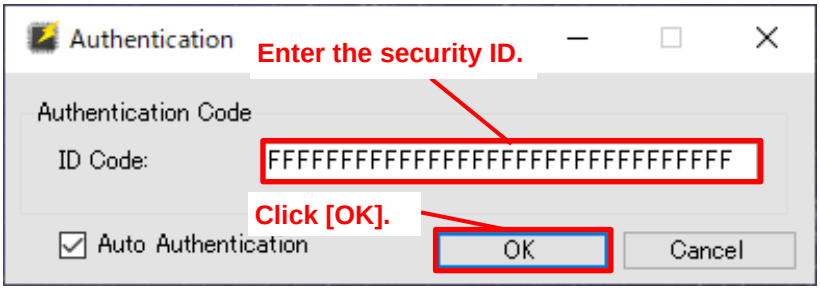
(1) Start RFP.
(2) Click the [File] tab.
(3) Click [New Project], and then create a dual mode project.



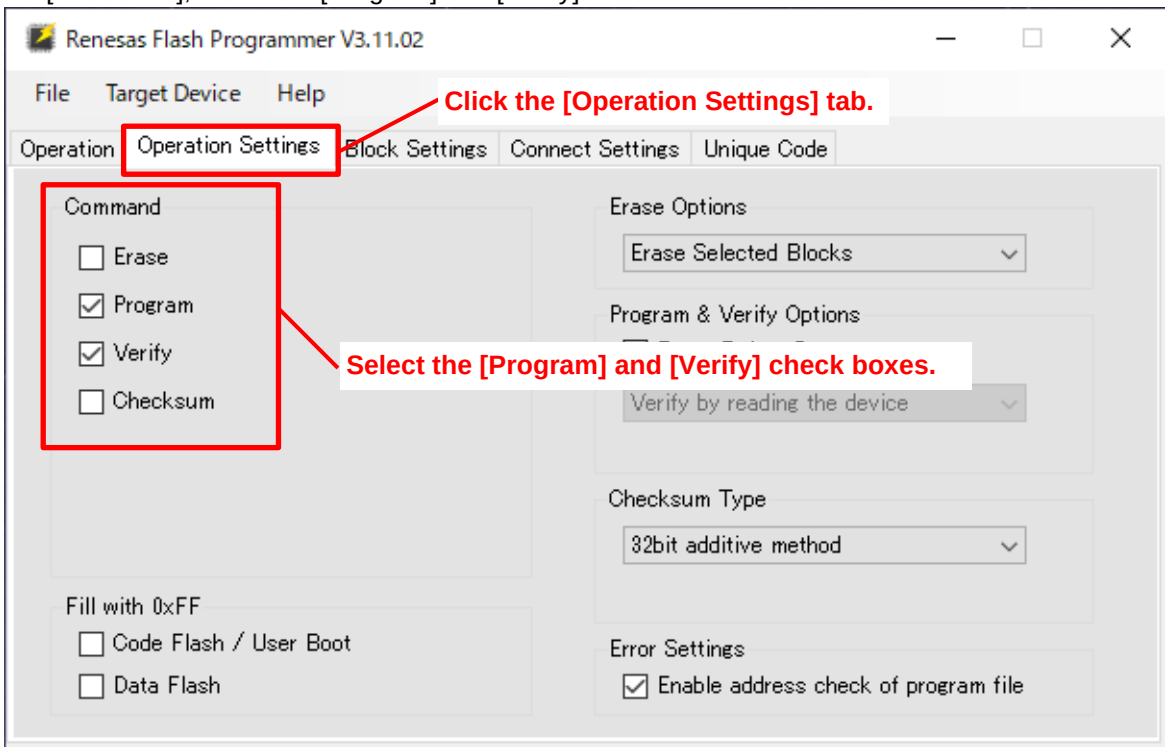
(4) Enter any project name in the [Project Name:] text box.
(5) Change other settings according to the system.
(6) Click [Connect].



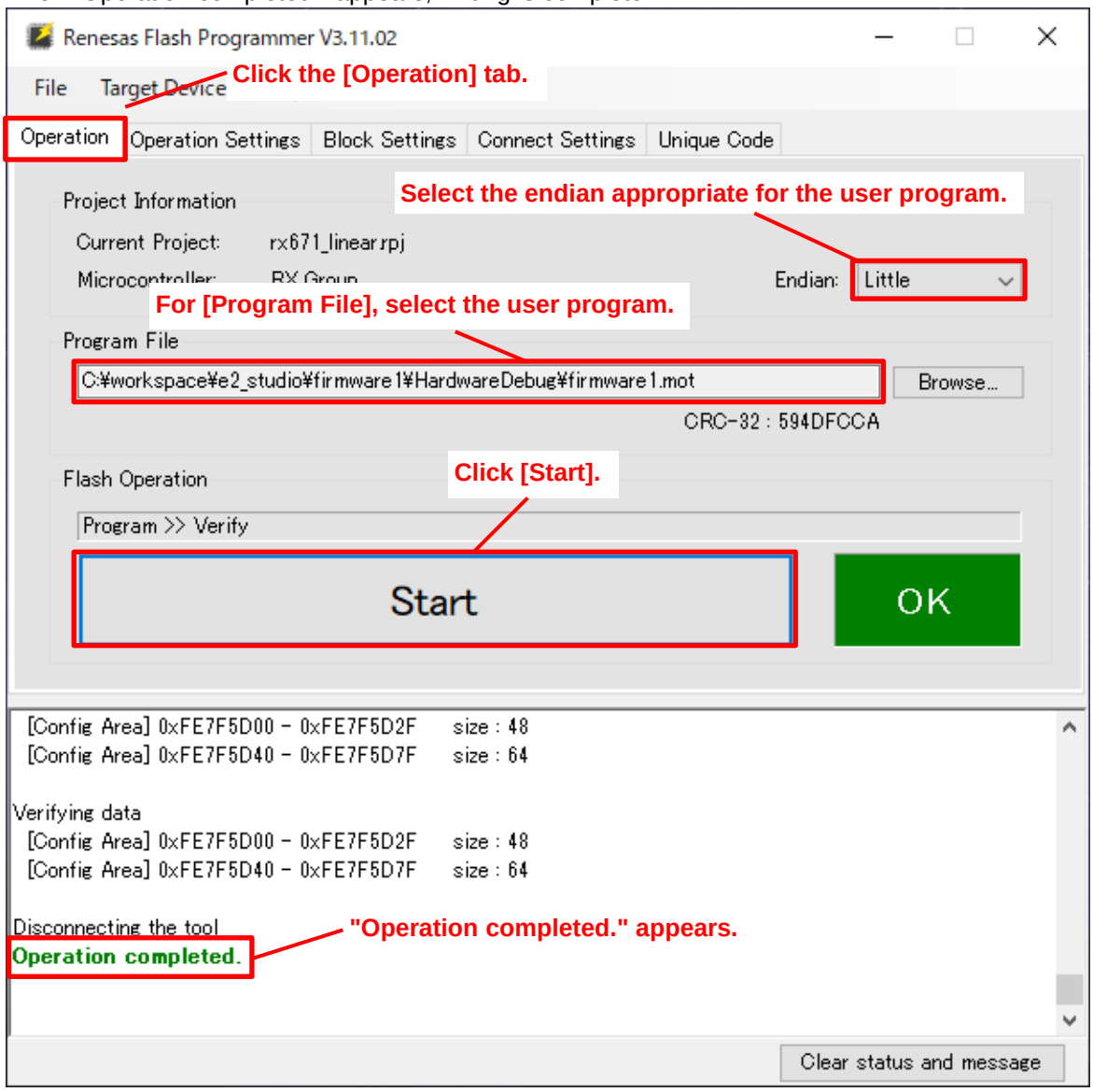
(7) In the [ID Code:] text box, enter the set security ID.
(8) Click [OK].



(9) Click the [Operation Settings] tab.
(10) For [Command], select the [Program] and [Verify] check boxes.



(11) Click the [Operation] tab.
(12) For [Endian:], select the endian appropriate for the user program.
(13) For [Program File], select the user program to be written.
(14) Click [Start].
(15) When "Operation completed." appears, writing is complete.



5. Notes

5.1 Option-Setting Memory Settings

The option-setting memory settings can be specified in the source code. However, do not set the BANKSEL.BANKSWP[2:0] bits for user programs of version 1.01 or later. Switching is allowed for self-programming only.

6. Reference Documents

User's Manual: Hardware

RX671 Group User's Manual: Hardware (R01UH0899EJ)
(The latest version can be downloaded from the Renesas Electronics website.)

Application Note

RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685EJ)
(The latest information can be downloaded from the Renesas Electronics website.)

RX Family Flash Module Using Firmware Integration Technology (R01AN2184EJ)
(The latest information can be downloaded from the Renesas Electronics website.)

RX Family Firmware Update Module Using Firmware Integration Technology (R01AN5824EJ)
(The latest information can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest information can be downloaded from the Renesas Electronics website.)

Revision History

Rev.	Date of issue	Description	
		Page	Summary
1.00	June 30, 2023	—	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/.

© 2023 Renesas Electronics Corporation. All rights reserved.