

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	3
1.1 Overview of Dual Mode	3
1.2 Updating Firmware	4
2. Preparation	5
2.1 If Bank Positions Are Unclear	5
2.1.1 Reading the Option-Setting Memory	5
2.1.2 Returning the MCU to the Shipment State	9
3. Debugging Method	13
3.1 When Developing User Program Version 1.00	13
3.1.1 Procedure for Creating a Project	13
3.1.2 Procedure for Setting Up BSP	17
3.1.3 Procedure for Setting the Debug Configuration	18
3.2 When Developing User Program Version 1.01 or Later	20
3.2.1 Procedure for Creating a Project	20
3.2.2 Procedure for Setting Up BSP	25
3.2.3 Procedure for Setting the Debug Configuration	26
3.3 Debugging the Startup Bank Switching Function	27
3.4 Outputting User Program MOT Files	
3.4.1 Procedures for Outputting MOT Files	
3.4.1.1 When Using Renesas Electronics C/C++ Compiler Package for RX Family	
3.4.1.2 When Using GCC for Renesas RX	35
3.4.2 Procedures for Outputting Offset MOT Files	
4. Writing a Dual-Mode Program by Using RFP	41
4.1 Writing a Program in Bank 0 of the MCU in Shipment State	41
4.2 Writing in Bank 1	45
5. Notes	49
5.1 Option-Setting Memory Settings	49
6. Reference Documents	50
Revision History	51



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.

Renesas Flash Programmer \ File Help	/3.11.02 Click the [File] tab.	- 🗆 X
New Project	K	
Open Project	Click [New Project].	
Save Project		
Save Image File		
File Checksum		
Set File Password		
Exit		Browse
	Start	
Renesas Flash Programmer V3.11.	.02 [3 Apr 2023]	
		Clear status and message



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

🕻 Create New Project	-	_		×
Project Information	Enter an	у рі	r <mark>ojec</mark> t n	ame.
Microcontroller:	RX67x V			
Project Name:	rx671_dual			
Project Folder:	C:¥workspace¥Renesas Flash Programmer¥V{		Browse	
Communication				
Tool: E2 emulato	r Lite 🗸 Interface: 2 wire UART 🗸			
Tool Details	Num: AutoSelect Power: None			
	Click [Connect].]	Cano	el

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

🕻 Authentication	Enter the s	security ID.		Х
Authentication Code				
ID Code:	FFFFFFFFFF	FFFFFFFFFFFF	FFFFFFFF	
	Click [OK].			
🗹 Auto Authentic	ation	OK	Cance	

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

📕 Rer	nesas Flash Prog	rammer V3.11.02	Click [Target Device].	_		\times
File	Target Device	Help				
Operat	Read Mem	nory	Connect Settings Unique Code			
	Read Block	k Information	Click [Read Memory].			
Pri	Blank Che	ck				
Ċ	urrent Project:	rx671_dua1rpj				_
M	licrocontroller:	RX Group	Endia	an: Little	~	
	ogram File Ish Operation			E	Browse	
E	Erase >> Program	>> Verify]
[St	tart			



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

Save As				×
\leftrightarrow \rightarrow \checkmark \uparrow Windows (C:) \rightarrow w	vorkspace Move to	o a folder of yo	ur choice.	
Organize 🔻 New folder				?
Name	Date modified	Туре	Size	
e2_studio Renesas Flash Programmer	5/29/2023 2:55 PM 5/30/2023 4:13 PM	File folder File folder		
File name: ofsm	Enter any file name.			~
Save as type: S-Record files (*.mot)				\sim
∧ Hide Folders	Click [Save].		Save Cancel	

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

Read Memory Information	- 🗆 X
Select Area Select [Config /	Area].
All Areas Code Flash 1 Data Flash 1	Start Address (HEX): FE7F5D00
Config Area	End Address (HEX): FE7F5D7F
Start Address = 0xFE7F5D00 End Address = 0xFE7F5D7F Size = 0.13 K	Length (HEX): 80
Click [Read].	
☑ Skip blank areas	Read Cancel



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

Kenesas Flash Programmer V3.11.02		_		×
File Target Device Help				
Operation Operation Settings Block Settings Connect Settings Unique Code				
	indian:	Little	~	
Program File		Bri	owse	
Flash Operation				
Erase >> Program >> Verify				
Start		Ok	(
Communication speed : 1,000,000 bps Signature: Device: RX Group				^
Reading data [Config Area] 0xFE7F5D00 - 0xFE7F5D7F size : 128				
Disconnecting the tool Operation completed. appears.				
р [Clear	status an	d messa	v ge



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.

Kenesas Flash Programmer	/3.11.02 Click the [File] tab.		-		×
New Project	k				
Open Project Save Project	Click [New Projec	:t].			
Save Image File File Checksum					
Set File Password					
Exit				Browse	
Flash Operation	Start				
Renesas Flash Programmer V3.11	.02 [3 Apr 2023]				
			Clear statu	s and mess	age



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

🕻 Create New Project		_		×
Project Information		Enter any	project	name.
Microcontroller: RX67	× ~		/	
Project Name: r×671	_dual			
Project Folder: C:¥wo	rkspace¥Renesas Flash Program	nmer¥V{	Browse	
Communication Tool: E2 emulator Lite	✓ Interface: 2 wire UART	~		
Tool Details No	m: AutoSelect Power: None	Clic	k [Conne	ect].
		onnect	Can	cel

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

📕 Authentication	Enter the security ID.	_		\times
Authentication Code				_
ID Code:	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	FFFFI	FFFFF	F
🖂 Auto Authentica			Cano	el



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

Renesas Flash Programmer V3.11.02	– 🗆 X
File Target Device Help Click the [O	peration Settings] tab.
Operation Operation Settings Block Settings Connect	Settings Unique Code Select [Erase Chip].
Command Erase Program Verify Checksum	Erase Options Erase Chip Program & Verify Options Erase Before Program Verify by reading the device Verify by reading the device
Select the [Erase] check box only.	Checksum Type 32bit additive method ~
Fill with 0xFF Code Flash / User Boot Data Flash	Error Settings Enable address check of program file



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

Renesas Flash Programmer V3.11.02			_		Х
File Target Device Help					
Operation Operation Settings Block Settings	Connect Settings	Unique Code			
Click the [Operati	ion] tab.				
Project Information					
Current Project: rx671_dua1rpj					
Microcontroller: RX Group		Endian	Little	~	
Program File					
			Br	owse]
Flash Operation	Click [Start].				
Erase	/				
Erdeo					
Star	t		Oł	<	
	t		Oł	<	
Erasing the selected blocks			OF	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF siz	ze : 8 K		Oł	<	^
Erasing the selected blocks	ze : 8 K		Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF size [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF Erasing the selected blocks	ze : 8 K		Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF size [Code Flash 1] 0xFFE00000 - 0xFFFFFFFFF	ze : 8 K		Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF siz [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF Erasing the selected blocks [Config Area]	ze : 8 K size : 2.0 M	ppears.	Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF siz [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF Erasing the selected blocks [Config Area]	ze : 8 K	ppears.	Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF siz [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF Erasing the selected blocks [Config Area] Disconnecting the tool "Operation	ze : 8 K size : 2.0 M	ppears.	Oł	<	^
Erasing the selected blocks [Data Flash 1] 0x00100000 - 0x00101FFF siz [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF Erasing the selected blocks [Config Area] Disconnecting the tool "Operation	ze : 8 K size : 2.0 M		O k		*



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].

	e2_studio - e ² studio Click the	File] tab.	
File	Edit Navigate Search Project	Renesas Views R	Run Window Help
	New	Alt+Shift+N >	Renesas C/C++ Project > Renesas Debug
	Open File.		C/C++ Project Renesas RL78
	Open Projects from File System		Project Renesas RX
	Recent Files	>	Cide N
	Close Edito	Ctr Click	k [Renesas C/C++ Project].
	Close All Editors	Ctrl+Shift+W	
	Save	Ctrl+S	Click [Renesas RX].
	Save As		
0	Save All	Ctrl+Shift+S	
	Revert		
	Move		
	Rename	F2	
8	Refresh	F5	
	Convert Line Delimiters To	>	
۵	Print	Ctrl+P	
ès	Import		
4	Export		
	Properties	Alt+Enter	
	Switch Workspace	>	
	Restart		
	Exit		



- (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].

C/C++ A C/C++ Executable Project for Renesas RX using the GCC for Renesas RX Toolchain. GCC for Renesas RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain. Renesas CC-RX C/C++ Executable Project A C/C++ Project for Renesas RX using the Renesas CC-RX toolchain. Renesas CC/RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the Renesas CC-RX toolchain. To use Renesas CC-RX, click [Renesas CC-RX C/C++	All				
A C/C++ Library Project for Renesas RX using the GCC for Renesas RX Toolchain. Renesas CC-RX C/C++ Executable Project A C/C++ Project for Renesas RX using the Renesas CC-RX toolchain. Renesas CC/RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the Renesas CC-RX toolchain. To use Renesas CC-RX, click [Renesas CC-RX C/C++			C for Renes	as RX Too	lchair
A C/C++ Project for Renesas RX using the Renesas CC-RX toolchain. Renesas CC/RX C/C++ Library Project A C/C++ Library Project for Renesas RX using the Renesas CC-RX toolchain. To use Renesas CC-RX, click [Renesas CC-RX C/C++			or Renesas R	X Toolcha	in.
To use Renesas CC-RX, click [Renesas CC-RX C/C++			X toolchain.		
			as CC-RX too	olchain.	
		To use Renesas CC-RX, click [Renesas CC-RX C/C+ Executable Project].	+		
	?	< Back Next > Fin	ish	Cance	1

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

0				- D X
	RX Executable Project	Enter any project r	name.	Ď
Project name: firm	nware1			
Use default loc	ation			
Location:	C:¥workspace¥e2_studio¥firm	nware1		Browse
	Create Directory for Project	t		
Choose file system	default 🗠			
Working sets				
Add project to	working sets			New
Working sets:			~	Select
	Click [Next].	~		
?		< Back Next >	Finish	Cancel



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

2		_		×
	RX Executable Project evice & debug settings			Ŷ
Toolchain: Toolchain Version: RTOS: RTOS Version:	● C ○ C++ Renesas CC-RX ~	onfiguration] check I	oox.	[
Device Settings	\sim	Configurations		
Target Board: Cu Target Device: R5	stom <u>Download additional boards</u> F5671EDxFB_DUAL Unlock Devices	Create Hardware Debu E2 Lite (RX) Create Debug Configu		⇒ ∼
Endian: Litt Project Type: De	tle ~	RX Simulator	uration	~
	Click [Next].			
?	< Back	Next > Finish	Canc	el



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

8				6 <u>—</u> 7		×
New Renesas CC-RX Executable Pro Select Coding Assistant settings	oject					3
Use Smart Configurator					-	-
Smart Configu which imports Select the [Use Si Smart Configurator encompasses unified view. Hardware resources conflict in peripheral middleware modules will be notified. (Smart Configurator is available only for	clock configuration vie I modules, interrupts ar	ew, interrupt config nd pins occurred ir	guration view		configu	
Softv	plication Code vare Component Middleware 8 Drivers Device Drivers CU Hardware	Smart Config	¢			
	Click [f	Finish].				
?	< Back	Next >	Finish		Canc	el



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~FFEFFFFh; Bank0: FFF00000h~FFFFFFFh.".
- (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.

💽 e	2_stud	lio - e² stud	io						
File	Edit	Navigate	Search	Project	Renesas Vie	ws	Run	Win	dow
3	1	*	🎋 De	bug	~	C×	firmv	/are1	Har
P	roject	Explorer >	< 🔝 🖪	題	Click the	e p	roje	ct.	
>	∮ firn	nware1 [Hai	rdwareD	ebug]					

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

C		Click the [Run] tab. 1
2.5	The Debug	F11
~	Run History	>
	Run As	>
	Run Configura	
	Debug H	ck [Debug Configurations
	to Debug As	>
	Debug Config	urations
	-	

- (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].

Debug Configurations			
Create, manage, and run configurations	Click	the [Debugger] tab.	Ť.
C 2 % 🗎 🗶 🖻 7 •	Name: firmware1 HardwareDebug	Click the [Connection Setting	gs] tab.
type filter text	📄 Main 🍄 Debugger 🍉 Startup 🦆 Source 🔽 Co	mmon	
C/C++ Application C/C++ Remote Application EASE Script		ee: R5F5671E_DUAL	
C GDB Hardware Debugging	GDB Settings Connection Settings Debug Tool Settin	ngs	
GDB Simulator Debugging (RH850)	Emulator	(Auto)	
🗾 Java Applet	Connection Type	JTag	~
Java Application	JTag Clock Frequency[MHz]	6.00	~
Aunch Group	Fine Baud Rate[Mbps]	1.50	~
Remote Java Application	Hot Plug	No	~
Renesas GDB Hardware Debugging	✓ Power		
- Change the setting of	of [Change startup bank] to [Yes].	Yes	~
Renk Change the Setting C	or following startup parking to fresh	3.3	~
	✓ CPU Operating Mode		
	Register Setting	Single Chip	~
	Mode pin	Single-chip mode	~
	Change startup bank	Yes	~
	Startup bank	Bank 0	~
	 Communication Mode 		
	Made	Debug Mode	~
Change the se	etting of [Startup bank] to [Bank (D]. r No	~
	ID Code	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	~



(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.

🖻 🏚 🗎 🗶 🖻 🍸 🕶	Name: firmware1 HardwareDebug	the Debug Teel Setti	ngol tob
be filter text	Main 🏂 Debugger 🕨 Startup 🦉 Source 🔲 (the [Debug Tool Setti	ngsj tab.
C/C++ Application C/C++ Remote Application EASE Script GDB Hardware Debugging	Debug hardware: E2 Lite (RX) ~ Target Device: R5F5 GDB Settings Connection Settings Debug Tool Settings	i671E_DUAL	
GDB Fardware Debugging GDB Simulator Debugging (RH850)	V IO		
Java Applet	Use Default IO Filename	Yes	~
Java Application	IO Filename	\${support_area_loc}	
Launch Group	✓ General Debug	#feebbox/meatest	
Remote Java Application	Reset After Reload	Yes	~
Renesas GDB Hardware Debugging	✓ Memory		
📑 firmware1 HardwareDebug	Endian	Little Endian	~
Renesas Simulator Debugging (RX, RL78)	Verify On Writing To Memory	No	~
Change the petting of		[204]	
	of [Debug the program	[0]	
re-writing the on-chi	p PROGRAM ROM] to	0x1000	
[Yes].		0x500	
	Debug the program re-writing the on-chip PROGRAM RO	OM Yes	~
	Debug the program re-writing the on-chip DATA FLASH	No	~
	✓ Start/Stop Function Setting		
	Execute function before running user program	No	~
	Address for start function		- ~
	Click [/	Apply].	
		Revert	Apply



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 e² studio.

(2) Click the [File] tab, and then click [Import].

.	e2_studio - e² stu	dio						
	Edit Navigate		Project	Renesas Views	Run	Wir	ndow	Help
C	New Open File Open Projects fr Recent Files	click the		o. Alt+Shift+N	> n	ware1		wareDe
	Close Editor Close All Editors	;		Ctrl+W Ctrl+Shift+W				
	Save Save As			Ctrl+S				
	Save All Revert			Ctrl+Shift+S				
A	Move Rename Refresh Convert Line De	lincitere Tr		F2 F5				
۵	Print	infinters io	,	Ctrl+P	_			
è	Import							
4	Export							
	Properties			Click [Impor	t].			
	Switch Workspa	ce			>			
	Restart						- 11	
	Exit							



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

(4) Click [Next].

elect		1	-
Rename and Import and Existing C/C++ Project into the workspace	:e	Ľ	1
Select an import wizard:			
type filter text			
 General Archive File Existing Projects into Click "Rename & Import C/C++ Project into Workspander Preferences Projects from Folder or Archive Renewas CC-RX project conversion to Renesas GCC RX Renesas CS+ Project for CA78K0R/CA78K0 Renesas GS+ Project for CC-RX and CC-RL Renesas GitHub FreeRTOS (with IoT libraries) Project Sample Projects on Renesas Website C/C++ Code Generator 	orkspac		<
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].

Import			_	
Rename & Impor	-	Enter al	ny project na	ame. 📂
Project name: firm	iware2			
Use default loca	ation			
Location:	C:¥workspace¥	e2_studio¥firmware	2	Browse
	Create Direc	tory for Project		
Choose file system:				
Import from:		Click [Brows	ej.	
Select root direct	ctory:		~	Browse
O Select archive fi	ile:		~	Browse
Projects:				
Options	figuration output	folders		
?	< Back	Next >	Finish	Cancel

- (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].

Select Folder					×						
← → ∽ ↑ 🖡	« workspace > e2_studio	Move	to the work sp	ace folder.							
Organize - Ne	ew folder				?						
Name	^	Date modified	Туре	Size							
.metadata		6/13/2023 10:21 AM	File folder		_						
📕 firmware1		6/30/2023 10:43 AM	File folder								
	Click the project folder. Click [Select Folder].										
	Folder: firmware1		Select Folder	Cancel							



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

📴 Import								×		
Rename & Import Project Select a directory to search for existing Eclipse projects.										
Project name:	firmware2									
✓ Use default	Use default location									
Location:	C:¥w	orkspace¥e2	_studio¥fin	mware2			Browse.			
	Cre	ate Directo	y for Proje	ct						
Choose file syst	em: defai	ult \sim								
Import from:										
Select root of	directory:	C:¥workspa	ce¥e2_stud	io¥firmw	/are1	~	Browse.			
O Select archiv	ve file:				Click the	e pro	ject.			
Projects:										
firmware1 (C	:¥workspa	ce¥e2_studio	o¥firmware	1)						
Options										
Keep build o	configurati	on output fo	Iders							
	Click [Finish].									
?		Back	Next >	Ì	Finish		Cance	1		

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.

e2_studio - e ² studio										
File	Edit	Navi	gate	Search	Project	Renesa	s Views	Run	Wir	ndow
«	*	*		🂠 De	bug		~	firmv	vare2	2 Har
🔁 Р	🍋 Project Explorer 🗙 🔝 問題 🛛 📄 🐄 🍞 🖇 🖵 🗖									
> 🍃	§ firm	ware	1							
~ 🕼	ら firm	ware	2 [Ha	rdwareD	ebug]					
)	- 🔊 I	nclude	es							
)) 😕 s	rc								
)	Bt	rash			\sim	Click th	e file	•		
	懲 f	irmwa	are1.s	cfg						
	firmware2 HardwareDebug.launch									
	() ()	Develo	oper A	Assistance	e					
1										



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

	e2_studio - e² studi Edit Navigate		Project	Panacas	TIONS	Rup	14/5	ndow	Help
File	New			ile] tab.					iwareDe
	Open File Open Projects fro	m File Sv	stem			8			
_	Recent Files	in the by	sterna			>			
	Close Editor			C	trl+W				
	Close All Editors			Ctrl+Sh	ift+W				
	Save			(Ctrl+S				
	Save As								
6	Save All		Click	[Renam	el				
	Revert		Ollon	Internation	-].				
	Move								
	Rename				F2				
\$]	Refresh				F5				
	Convert Line Deli	miters To				>			
۵	Print			(Ctrl+P				
2	Import								
4	Export								
	Properties			Alt+	Enter				
	Switch Workspace	e				>			
	Switch Workspace Restart	e				<i>`</i>			

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

Rename Resource			\times
New name: firmware2.scfg			
✓ Update references	Enter an	y filo na	mo
Open preferences		ly me na	me.
Click [OK].			
Preview >	ОК	Cance	el



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.
```

99	<pre>#pragma addressOSIS2reg = 0xFE7F5D54</pre>	
100	<pre>#pragma addressOSIS3reg = 0xFE7F5D58</pre>	
101	<pre>#pragma addressOSIS4reg = 0xFE7F5D5C</pre>	
102	#pragma addressFAWreg = 0xFE7F5D64 / Comment out	
103	<pre>#pragma addressROMCODEreg = 0xFE7F5D70</pre>	
104		
105	<pre>const uint32_tMDEreg = (BSP_PRV_MDE_VALUE & BSP_PRV_BANK_MODE_VALUE);</pre>	
106	<pre>const uint32_tOFS0reg = BSP_CFG_OFS0_BFG_VALUE;</pre>	
107	<pre>const uint32_tOFS1reg = BSP_CFG_OFS1_REG_VALUE;</pre>	
108	const uint32 t IMINEreg = 0xffffff	
109	<pre>//const uint32_tBANKSELreg = BSP_PRV_START_BANK_VALUE;</pre>	
110	<pre>const uint32_tSPCCreg = BSP_PRV_SPCC_VALUE;</pre>	
111	<pre>const uint32_tTMEFreg = BSP_CFG_TRUSTED_MODE_FUNCTION;</pre>	
112	<pre>const uint32_tOSIS1reg = BSP_CFG_ID_CODE_LONG_1;</pre>	
113	<pre>const uint32_tOSIS2reg = BSP_CFG_ID_CODE_LONG_2;</pre>	
114	<pre>const uint32_tOSIS3reg = BSP_CFG_ID_CODE_LONG_3;</pre>	
115	<pre>const uint32_tOSIS4reg = BSP_CFG_ID_CODE_LONG_4;</pre>	
116	<pre>const uint32_tFAWreg = BSP_CFG_FAW_REG_VALUE;</pre>	
117	<pre>const uint32_tROMCODEreg = BSP_CFG_ROMCODE_REG_VALUE;</pre>	
118		
119	<pre>@ #elif defined(GNUC)</pre>	

For GCC for Renesas RX

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

vecttbl.c ×	
118	
119	<pre>#elif defined(_GNUC)</pre>
120	
121	<pre>const st_ofsm sec_ofs1_tofsm_sec_ofs1attribute ((section(".ofs1"))) = {</pre>
122	(BSP_PRV_MDE_VALUE & BSP_PRV_BANK_MODE_VALUE), /*MDEreg */
123	BSP_CFG_OFS0_REG_VALUE, /*OFS0reg */ Comment out
124	BSP_CFG_OFS1_REG_VALUE /*OFS1reg */
125	};
126	const uint32 t TMINEreg attribute ((section("ots2"))) = 0xffffffff;
127	<pre>//const uint32_tBANKSELregattribute ((section(".ofs3"))) = BSP_PRV_START_BANK_VALUE;</pre>
128	<pre>const uint32_tSPCCregattribute_ ((section(".ofs4"))) = BSP_PRV_SPCC_VALUE;</pre>
129	<pre>const uint32_tTMEFregattribute ((section(".ofs5"))) = BSP_CFG_TRUSTED_MODE_FUNCTION;</pre>
130	<pre>const st_ofsm_sec_ofs6_tofsm_sec_ofs6attribute ((section(".ofs6"))) = {</pre>
131	<pre>BSP_CFG_ID_CODE_LONG_1, /*OSIS1reg */</pre>
132	<pre>BSP_CFG_ID_CODE_LONG_2, /*OSIS2reg */</pre>
133	<pre>BSP_CFG_ID_CODE_LONG_3, /*OSIS3reg */</pre>
134	BSP_CFG_ID_CODE_LONG_4 /*OSIS4reg */
135	};
136	<pre>const uint32_tFAWregattribute ((section(".ofs7"))) = BSP_CFG_FAW_REG_VALUE;</pre>
137	<pre>const uint32_tROMCODEregattribute ((section(".ofs8"))) = BSP_CFG_ROMCODE_REG_VALUE;</pre>
138	
139	<pre>#elif defined(_ICCRX_)</pre>

(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.

💽 ež	2_stud	lio - e² stud	io						
File	Edit	Navigate	Search	Project	Renesas Vie	ews	Run	Wir	dow
«	1	*	🂠 De	bug	~	C×	firmv	vare1	Han
🔁 Р	Project Explorer × 🔝 問題 Click the project. □								
> 🔓	∮ firn	nware1 [Ha	rdwareD	ebug]					
> 🔓	∮ firm	ware2							

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

	Run Click the [Run] tab.	F11 F11
ep I	Run History	>
	O Run As	>
	Run Configurations	
	Debug Vis Click [Debug Configu	ratior
	🎄 Debug As	>
	Debug Configurations	
	Q External Tools	>

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

lame: firmware1 HardwareDebu		ommon			
Initialization Commands Reset and Delay (seconds): Halt	Clic	k the [Sta	rtup] tab.		^ ~
Load image and symbols Filename Program Binary [firmwa	Load type Image and Symbols	Offset (hex) 0	On connect Yes		Add
			Cli	ck [Add].	Remove Move up
Runtime Options					Move down
Set program counter at (hex Set breakpoint at:	i): main]			



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

Add download module									
Specify download module name: Select user program 2.									
<pre>\${workspace_loc:¥firmware2¥HardwareDebug¥firmware2.x}</pre>									
Variables	Search Project	Workspace	File System						
Click [Of	(]	ОК	Cancel						

- (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].

Debug Configurations						— C	x i	
Create, manage, and run configurations						ć	Ś	
 Image: Second Second	Name: firmware1 HardwareDebi	Load type Image and Symbols Image and Symbols Image and Symbols	Offset (hex)	On connect Yes Yes		Add Edit Remove Move up Move dowr		
Filter matched 13 of 15 items	Resume	C	lick [Del	bug]. 🔨	Revert	Ap	~ ply	
۲					Debug	0	lose]



(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.

📬 🕶 🔚 🐚 🐌 🕶 🔦	- 🖬 🔍 🕨 🔲 🔳 🗱 🔊 💿 🕼 🖬 🗮 📆
org.c × 💽 firmware1.c	Click the [Suspend] button.
ff00000 • R_BSP_POR_F	UNCTION(R_BSP_STARTUP_FUNCTION)
{ /* Stac	k pointers are setup prior to calling thi
	can use auto variables in this function b be unavailable after you change the stac

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].

2	の格	Run Click the [Run] tab.	+F11 F11
0	0	Run History Run As	>
40		Run Configurations Debug His Debug As	uration
		Debug Configurations	
	Q.	External Tools	>



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

Click the [Connection Settings] tab. Emulator (Auto) Connection Type JTag JTag Clock Frequency[MHz] 6.00 Fine Baud Rate[Mbps] 1.50 Hot Plug No	~
Emulator (Auto) Connection Type JTag JTag Clock Frequency[MHz] 6.00 Fine Baud Rate[Mbps] 1.50 Hot Plug No	^
JTag Clock Frequency[MHz] 6.00 Fine Baud Rate[Mbps] 1.50 Hot Plug No	~
Fine Baud Rate[Mbps] 1.50 Hot Plug No	~
Hot Plug No	
	\sim
Device.	~
Power	
Power Target From The Emulator (MAX 200mA) No	~
Supply Voltage (V) 3.3	\sim
CPU Operating Mode	
Register Setting Single Chip	~
Mode pin Single-chip mode	~
Change startup bank Yes	~
Startup bank Bank 1	~
Communication Mode	
hange the setting of [Startup bank] to [Bank 1]. ^{ug Mode}	~
nange me cound e. fermich senul te fermicit.	~
Flash ID Code FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
Change the setting of [Startup bank] to [Bank 1].	

(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.

1 📑 🕶 🛙	a (a) e	9 - 🐔 -	D	D 🛛	N 3. 9	3 .e i	• 🗟 🔊 i
					Click	the [Terion.	minate]
org.c 🗙 🛛	👌 firmwar	re1.c					
ff00000	{			BSP_START	_	-	ling this
	•			to variab lable afte			



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.

e2_studio - e ² studio	
File Edit Navigate Search Project	Renesas Views Run Window
🐔 🐐 🔳 🎋 Debug	✓ Firmware1 Hat
🍋 Project Explorer 🗙 🔝 問題	E 🕏 🎖 🕴 🗖 🖬
> 😂 firmware1 [HardwareDebug]	
> 🚰 firmware2	
Cli	ck the project.

(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 e² studio.
- (2) Click the [Run] tab, and then click [Debug Configurations].

~	00	Run Click the [Run] tab.	+F11
120 5	10	Debug	F11
		Run History	>
	0	Run As	>
		Run Configurations	
		Debug Ks Click [Debug Configu	ration
	蓉	Debug As	>
		Debug Configurations	
	0	External Tools	

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

Debug Configurations Create, manage, and run configurations	lick the [Debugger] tab.	□ ×	
C 🖻 🏚 🗎 🗮 🖻 🍸 🕶	Name: firmware1 HardwareDebug	Click the [Connection Setti	ngel tab
type filter text	Mair 🕸 Debugger 🕨 Startup 🦉 Source		ngsj tab.
C C/C++ Application C C/C++ Remote Application EASE Script G GDB Hardware Debugging	GDB Settings Connection Settings Debug Too		
GDB Simulator Debugging (RH850)	Emulator	(Auto)	^
Java Applet	Connection Type	JTag	~
Java Application Launch Group	JTag Clock Frequency[MHz]	6.00	~
Remote Java Application	Fine Baud Rate[Mbps]	1.50	~
Renesas GDB Hardware Debugging	Hot Plug	No	~
firmware1 HardwareDebug	✓ Power	o ti	
Renesas Simulator Debugging (RX, RL78)	Power Target From The Emulator (MAX 20	0mA) Yes 3.3	~
Kenesas sinidiator Debugging (KX, KE76)	Supply Voltage (V)	3.3	· · · ·
	CPU Operating Mode Register Setting	Single Chip	~
	Mode pin	Single-chip mode	*
	Change startup bank	Yes	<u> </u>
	Startup bank	Bank 0	~
	Communication Mode	Durin O	
	Mode	Debug Mode	~
	Change the setting of [St		~
	ID Code	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	~

(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].

Click the [Window] tab.						
Window	Help			ທ່າງ ເລຍ.		
New	Window	1				
Edito	or >	-				
App	earance >	BS	P POR FUNCT	ION(R BSP	STARTUP	FUNCTION
Shov	v View →	۲	Build Targets	`-		
Pers	pective >	6	C/C++ Project	ts		
		_	now View].		Alt+Shi	ft+Q, C
	gation	1≣	Include Brow	ser		
	erences		Optimization	Assistant		
21		臣	Outline		Alt+Shif	ft+Q. 0
214		 @	Problem Deta	ils		
21	-		Problems		Alt+Shi	#+0 X
210	-				Alt+Shi	11+Q, X
21		6	Project Explo	rer		
219	-		Properties			
220	-	A	Search		Alt+Shi	ft+Q, S
22:		-	Smart Browse	er		
22	s fffaaaao		Smart Manua	1		
	Click [Other].	æ	Tasks			
22		_				
	o 7 fff00017	\$	Terminal			
22			Other		Alt+Shif	ft+Q, Q
	0					

(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

R Debugger Console ×
firmware1 HardwareDebug [Renesas GDB Hardware Debugging] rx-elf-gdb -rx-force-isa=v3 -rx-force-double-fpu (7.8.2)
<pre>monitor set_io_access_width,RW,1,91829-9182a,9182a-9182b,9182b-9182c,9182c-9182d,9182d-9182 monitor set_io_access_width,RW,1,91837-91838,91838-91839,91839-9183a,9183a-9183b,9183b-9183</pre>
<pre>monitor set_io_access_width,RW,1,a0b00-a0b03,a0b05,a0b0c-a0b0f,a0b17-a0b1f,c1200-c120a,c120 monitor set_io_access_width,RW,1,c1298-c1299,c1300-c1304,c1324,c1326,c1328,c1380-c1382,c138</pre>
monitor set io access width Pull of 1220 of 1240 of 1240 of 1260 of 1600 monitor set io access with Enter the command, and then press the Enter key. 1005 monitor set io access is is in access a fact, which a command, and then press the enter key. 1005
monitor set_lo_acces_width.BW 1.7Fe010.7Fe014.7Fe018,7fe0d0,7fe0d4 dump srcc memory firmware2.mot 0xffe00000 0xffeffff

(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.

📕 Renesas Flash Programmer	V3.11.02 —— Click the [File] tab.		-		×
File Help					
New Project					
Open Project	Click [New Project].				
Save Project					
Save Image File					
File Checksum					
Set File Password	-				
Exit			Br	owse	
Flash Operation	Start				
Renesas Flash Programmer V3.11	.02 [3 Apr 2023]				
		Clear :	status ar	nd messa	age



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

📕 Create New Project		_		×
Project Information		Enter en	un to ot u	
Microcontroller:	RX67x	Enter any	project n	ame.
Project Name:	rx671_linear			
Project Folder:	C:¥workspace¥Renesas Flash Program	nmer¥V{	Browse	
Communication Tool: E2 emulator	Lite 🗸 Interface: 2 wire UART	~		
Tool Details	Num: AutoSelect Power: None			
	Click [Connect].	onnect	Cano	el

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

(8) Click [OK].

🕻 Authentication	Enter the security ID.	-	\times
Authentication Code			
	FFFFFFFFFFFFFFFFFFFFFF	FFFFFFFF	F
🗹 Auto Authentica	Click [OK].	Can	cel



(9) Click the [Operation Settings] tab.

(10) For [Command], select the [Program] and [Verify] check boxes.

🜠 Renesas Flash Programmer V3.11.02	– 🗆 X						
File Target Device Help Click the	Operation Settings] tab.						
Operation Operation Settings Block Settings Conn	nect Settings Unique Code						
Command	Erase Options						
Erase	Erase Selected Blocks \sim						
Program	Program & Verify Options						
Verify Select the [Progra	m] and [Verify] check boxes.						
Checksum	Verify by reading the device $~\sim~$						
	Checksum Type						
	32bit additive method \sim						
Fill with 0xFF							
Code Flash / User Boot Error Settings							
🗌 Data Flash	Enable address check of program file						



- (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.

File Target Device Click the [Operation] tab.	– 🗆 X					
Operation Operation Settings Block Settings Connect Settings Unique Code						
Project Information Current Project: rx671_linear.rpj Microcontre For [Program File], select the user program.	for the user program.					
Program File						
C:¥workspace¥e2_studio¥firmware1¥HardwareDebug¥firmware1.mot	Browse					
	594DFCCA					
Flash Operation Click [Start].						
Program >> Verify						
Start	ок					
[Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64	^					
Verifying data [Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64 Disconnecting the tool "Operation completed." appears. Operation completed.						
Uperation completed.						
	Clear status and message					



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 optionsetting 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.

📕 Renesas Flash Programmer		_	-	×
File Help	Click the [File] tab.			
New Project	K			
Open Project	Click [New Project].			
Save Project				
Save Image File				
File Checksum				
Set File Password				_
Exit			Browse	
	Start			
Renesas Flash Programmer V3.11	.02 [3 Apr 2023]			
		Clear stat	tus and mess	age



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

📕 Create New Project		_		×
Project Information	Enter			
Microcontroller:	RX67x	any p	project na	ame.
Project Name:	rx671_linear]		
Project Folder:	C:¥workspace¥Renesas Flash Programmer¥V3	:	Browse	
Communication Tool: E2 emulator				
Tool Details	Num: AutoSelect Power: None			
	Click [Connect].		Cance	el

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

(8) Click [OK].

🕻 Authentication	Enter the security ID.	_		×
Authentication Code ID Code:	FFFFFFFFFFFFFFFFFF	FFFFF	FFFFF	
Auto Authentic	Click [OK].		Cance	el



) Click the [Operation Settings] tab. (9 (1

🌠 Renesas Flash Programmer V3.11.02	- 🗆 ×
File Target Device Help	k the [Operation Settings] tab.
Operation Operation Settings Block Settings	Connect Settings Unique Code
Command	Erase Options
Erase	Erase Selected Blocks \sim
🗹 Program	Program & Verify Options
Verify Select the [Program] and [Verify] check boxes.
Checksum	Verify by reading the device \sim
	Checksum Type
	32bit additive method \sim
Fill with 0xFF	
Code Flash / User Boot	Error Settings
🗖 Data Flash	Enable address check of program file



- (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.

📓 Renesas Flash Programmer V3.11.02 —					
File Target Device Click the [Operation] tab.					
Operation Operation Settings Block Settings Connect Settings Unique Code					
Project Information Select the endian appropriate fo	r the user program.				
Current Project: rx671_linear.rpj					
	Endian: Little 🗸 🗸				
For [Program File], select the user program. Program File					
C:¥workspace¥e2_studio¥firmware1¥HardwareDebug¥firmware1.mot	Browse				
CRC-32 :	594DFCCA				
Flash Operation Click [Start].					
Program >> Verify					
Start OK					
Start	ок				
[Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48	ОК	^			
	OK	^			
[Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64 Verifying data	OK	^			
[Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64 Verifying data [Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48	OK	^			
[Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64 Verifying data [Config Area] 0xFE7F5D00 - 0xFE7F5D2F size : 48 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F size : 64 Disconnecting the tool "Operation completed." appears.	OK	^			



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

		Description	
Rev.	Date of issue	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 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

- 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.
- 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.
- This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 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
- (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: <u>www.renesas.com/contact/</u>.