

# APPLICATION NOTE

R01AN2537EJ0102

# RX113 Group

On-chip Flash Memory Programming Solution using USB Memory Mar 31, 2017

## **RX Driver Package Application**

## Introduction

This document is an application note of the on-chip flash memory programming solution using USB Memory.

This application note includes the main program that writes the program stored in the USB memory into the RX113 onchip flash memory and execute it.

The main program of the application note is used in combination with FAT file system, USB driver, Flash memory module included in the RX110, RX111, RX113, RX231 Group RX Driver Package.

## Target Device

RX113 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate.

## **Related Documents**

- Firmware Integration Technology User's Manual (R01AN1833EU)
- RX Family Board Support Package Module Using Firmware Integration Technology (R01AN1685EU)
- RX Family Adding Firmware Integration Technology Modules to Projects (R01AN1723EU)
- RX Family Adding Firmware Integration Technology Modules to CubeSuite+ Projects (R01AN1826EJ)

## Contents

1. Overview	3
1.1 This Application Note	
1.2 Operating Environment	
1.3 Module Structure	
1.4 File Structure	5
2 Acquiring a Development Environment	6
2.1 Acquire and install e <sup>2</sup> studio	6
2.1 Acquire a Compiler Package	6
	0
3. Building a Project	7
3.1 Create a Workspace	7
3.2 Create a Project	
3.3 Import a Project	10
3.4 Modify Configuration	15
3.4.1 Change Configuration	15
3.4.2 Change Project Setting	
4. Verify Operation	20
4.1 Build the Project	
4.2 Prepare for Debugging	
4.2.1 Configure Hardware	
4.2.2 Set up the RSK	
4.2.3 Prepare USB Memory	
4.3 Debug the Project	25
E Application over ious	20
5. Application overview	
5.1 Memory structure	
6. Main Program Specifications	31
6.1 Files	
6.2 Modules	
6.3 Flowcharts	

## 1. Overview

## 1.1 This Application Note

This application note describes the procedure for main program evaluation by combining the Board Support Package (referred to as "BSP"), Flash memory (referred to as "Flash API"), USB driver (Host Mass Storage Class Driver "USB HMSC" and "Basic Firmware"), M3S-TFAT-Tiny FAT file system (referred to as "TFAT") of Firmware Integration Technology (referred to as "FIT") modules included in the RX110, RX111, RX113, RX231 Group RX Driver Package.

This application note operates on the Renesas Starter Kit for RX113 (referred to as "RSK" in the remainder of this document)

The program (Sample program) executed after the programming is also available. The sample program is stored in the "demo" folder in each project.

## 1.2 Operating Environment

The table below lists the operating environment of the main program and the sample program.

#### Table 1-1 Operating Environment

Items	Contents
Microcontroller	RX113 Group
Evaluation board	Renesas Starter Kit for RX113 (Part No.: R0K505113S000BE)
Integrated development environment (IDE)	e2 studio, V4.1.0 or later
C Compiler	RX Family C/C++ Compiler Package V2.03.00 or later
Emulator	E1
Endian	Little endian
RX Driver Package	RX110, RX111, RX113, RX231 Group RX Driver Package Ver.1.01 (R01AN2670EJ)*

Note:\* Operation of this application note has been verified when the modules in the RX Driver Package mentioned above are incorporated. If any of the modules used in this application note are replaced with a different module, the user must verify the operation.



## 1.3 Module Structure

Figure 1-1 shows Module structure of the main program, and Table 1-2 Modules lists the FIT modules to be included in the main program. Some modules are included in the sample program.



#### Figure 1-1 Module Structure

#### Table 1-2 Modules

Туре	Module	FIT Module Name	Rev.
BSP	Board Support Package (BSP)	r_bsp	3.01
Middleware	M3S-TFAT-Tiny FAT file system (TFAT)	r_tfat_rx	3.02
	M3S-TFAT-Tiny Memory Driver Interface	r_tfat_driver_rx	1.02
Device Driver	USB Basic Firmware	r_usb_basic	1.01
	USB Host Mass Storage Class (USB HMSC)	r_usb_hmsc	1.01
	LCD controller/driver (LCDC)	r_lcdc_rx	1.00
	Flash memory (Flash API)	r_flash_rx	1.30
Application	Main program FIT module	r_flash_writer_rx113	1.00

## 1.4 File Structure

Figure 1-2 shows the file structure used in this application note.



Figure 1-2 File Structure

When the ZIP file provided with this application note is decompressed, a folder with the same name is created, and the various folders and files are created within that folder

The "workspace" folder is the project to build "On-chip Flash Memory Programming application using the USB memory". To use the  $e^2$  studio, import the project into the workspace.

Documents that describe using the FIT modules in various development environments are included in the **reference\_documents** folder. The document "Adding Firmware Integration Technology Modules to Projects" (R01AN1723EU) describes the method for including the FIT modules, as a FIT plugin, in an e<sup>2</sup> studio project. The document "Adding Firmware Integration Technology Modules to CS+ Projects" (R01AN1826EJ) describes the method for including the FIT modules in a CubeSuite+ project.

## 2. Acquiring a Development Environment

## 2.1 Acquire and install e<sup>2</sup> studio

Access the following URL and download the  $e^2$  studio.

http://japan.renesas.com/e2studio\_download

This document requires you to use  $e^2$  studio V4.1.0 or later. If the version older than V4.1.0 is used, some functions of the  $e^2$  studio may not be available. For download, obtain the latest version of the  $e^2$  studio on the website

## 2.2 Acquire a Compiler Package

Access the following URL and download the RX Family C/C++ Compiler Package.

http://japan.renesas.com/e2studio download



## 3. Building a Project

This application note includes environment-built project. The procedure to import a Project using e2 studio Smart browser is described below.

### 3.1 Create a Workspace

1. Start e<sup>2</sup> studio.

#### 2. Enter an arbitrary workspace folder in the displayed dialog box and click [OK].

e <sup>2</sup> Workspace Launcher	<b>—</b>	
Select a workspace		
e2 studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.		
Workspace: C:\workspace	✓ <u>B</u> rowse	Enter a workspace
		loidel.
Use this as the default and do not ask again		
	OK Cancel	—— Click [ <b>OK]</b> .

#### 3. When the following window is displayed, click [Workbench].

e <sup>2</sup> C/C++ - e2 studio	Droject Pup Win	dow Holp					
	Eroject <u>R</u> un <u>H</u> in	uow <u>n</u> up				<b>∆</b> ⇔ ⇔ ≠	
Mo	100						
AAC	ICU						
		-	201	udi	_		
		e	250	uai	•		
	0	1	RA	$\wedge$			Click [Workbonch]
	Z	Y	YOY	$\sim$	Ø		
					Go to	Workbench the workbench	
RENESAS							



## RX113 Group On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

## 3.2 Create a Project

When using the Smart browser function, the target project or file needs to be selected. To use this function, create the project that specified the target device (Note 1).

Note 1: The project to be created here is a dummy to use the Smart browser.

1. Click [File (F)], [New (N)], then [C Project] to create new C project. Start [Create New project wizard].



2. Input any project name and select [Renesas RXC Toolchain]. Click [Next (N)].

e <sup>2</sup> C Project		9
C Project Selected toolchain is not integrated. Please in	nstall from "www.kpitgnutools.com/downloadCenter.php" and integrate.	
Project name: sample		
Use default location		
Location: C:\WorkSpace\RDP\20160125\samp	le Browse	
Create Directory for Project		
Project type:	Toolchains:	
<ul> <li></li></ul>	KPIT GNUARM-NONE-EABI Toolchain KPIT GNURL78-ELF Toolchain KPIT GNURX-ELF Toolchain KPIT GNUSH-ELF Toolchain Renesas CCRL Toolchain Renesas SHC Toolchain	——— Select "Renesas RXC Toolchain".
Show project types and toolchains only if the	ey are supported on the platform	_
0	< Back Next > Finish Cancel	Click here.

# On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

3. Set Select Target to R5F51138AxFP for RX113 100 Pin device. For other items, any setting is OK. When the setting is completed, click [Finish(F)].

a <sup>2</sup> C Project			
e2 studio - Project Gener Select Target Specific Settin	ation gs		
Toolchain Version : Debug Hardware: Data endian : Select Target: Select Configurations Ø Hardware Debug Ø Debug using Simu Release (no debug	v2.03.00 E1 (RX) Little-endian data R5F56107VxFP  : Debug using hardware silator : Debug using simulator j) : Project without any debug information	* 10-10-10-10-10-10-10-10-10-10-10-10-10-1	
Build configurations w will be built for the act (RX700) the debug has	vill be created in the project only for the selected debug mode options, however twe configuration i.e., first configuration selected from group. Based on the dev dware (E1 (RX)) and debug target (R5F56107VxFP), debug configuration will be a	by default the project ce selection you made utomatically created for	
?	< gack Next >	Einish Cancel	— Click her

#### 4. Click [OK].

ummary		×
Project Summary:		
PROJECT GENERATOR PROJECT NAME : samp PROJECT DIRECTORY : CPU SERIES : CPU TYPE : TOOLCHAIN NAME : TOOLCHAIN VERSION : GENERATION FILES :		
And and the second seco	ngʻalonci saraqakor. Ngʻalonci shaqator Ngʻalonci Sagashdima A Ngʻalonci nashtik s	
Click OK to generate the project or	Cancel to abort.	cel Click her

## 3.3 Import a Project

Import the project of Main program in the workspace created.

This application note includes the projects that select a file by;

• Project that selects a file by switch

1. Select the project created in "3.2 Create a Project" from Project explorer.



2. Click on [Renesas Views]  $\rightarrow$  [e2 Solution Tool kit]  $\rightarrow$  [Smart browser] to start the Smart browser.



3. Click on the [Application Note] on the [Smart browser] tabbed page.

Properties	🔋 Memory Usage	🔓 Stack Analysis	🌸 Smart Brows	er 🛿
date Appli	cation Notes	Cli	ick here.	
Docum	ient No.	Rev.	Issue Date	Sample C

4. Click [Update]	]. 🚱				
lysis 🏽 🌒 Smart Brov	vser 🛛		\$ \$	▽ □ □	Click here.
Issue Date	Sample Code	Remarks			
ISSUE Date	Sample Code	Nemarks			

5. Select the application note and right-click. Then, click on [Sample code(Project import)] in the context menu. (Note 1).

🖹 Problems 🧔 Tasks 📮 Console	🔲 Properties 🔋 Memo	ry Usage 🛭 😭 Stack An	alysis 🛯 🛶 Smart Browser	8
Device:		Last u	pdated:	1000
User's Manual: Hardware Technical	Jpdate Application Notes	5		
matches				
Title		Document No.	Rev.	Issue Da
THE REPORT AND ADDRESS OF	the second se	the second state of	Apr (- A)	10,000
WP Millionly (PC Multimos	Open		Sec. 6.0	100.00
All and a Table Medical	Sample Code (downloa	d)	100 A 100	100.00
Mark Roberty ST Multi-Huse	Sample Code (import p	rojects)	Click here.	100.000
Internet, states states	Property		-1.8	1000
			bish.	10.00
•		III		

Note 1: If authentication by My Renesas has never been performed, "My Renesas" dialog opens when downloading the file. Enter your mail address and password registered in the Renesas website.

e <sup>2</sup> My Renesas		×
My Renesas		
Enter the e-mail addres They allow you to dow	; and password that you registered for My Renesas. Iload documents and software by using Smart Browser.	
Email Address:		
Password:		
Create a My Pen	esse account to use our tool download convices, receive Neucletter / Undate Natice, and take adventage of our other convices	
Click [About My	Renesas] to register it.	
	Alexed Md. Pressen	OK Canad
	About My Kenesas	Cancel



## 6. Click [Agree].

e <sup>2</sup> End User License Agreement (Sample Code)	8
BY DOWNLOADING ALL OR ANY PORTION OF "THE SOFTWARE", YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT AND	=
AGREE THAT THIS AGREEMENT IS ENFORCEABLE BETWEEN YOU AND RENESAS ELECTRONICS CORP., LIKE ANY WRITTEN NEGOTIATED	
AGREEMENT SIGNED BY YOU. IF YOU DO NOT AGREE THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE.	
AGREEMENT	
Renesas Electronics Corp., a Japanese corporation, with a principal place of business at 6-2 Otemachi 2-chome, Chiyoda-Ku, Tokyo, Japan.	
("Renesas") owns all intellectual property in the Software and permits you to use the Software (defined as below) only in accordance with the terms of	f
this Agreement.	
1. Definitions. 	Ŧ
Agree Disagree	2

#### 7. Save the application note.

e <sup>2</sup> Save As			23
G v 🖡 🕨 Com	puter 🕨	+ 4+ Search Co	mputer 🔎
Organize 🕶			s • 0
Desktop Downloads Ecent Places Libraries Documents Music Fictures Videos	<ul> <li>Hard Disk Drives (3)</li> <li>Local Disk (C:)</li> <li>Local Disk (C:)</li> <li>Local Disk (Q:)</li> <li>Local Disk (Q:)</li> </ul>		
File name:			
Hide Folders		Save	Cancel

## On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

8. Select the project to be imported, and click [Finish(F)].

Select me project to be imported, and enew [1 mish(1)].	
e² Import	
Import Projects Select a directory to search for existing Eclipse projects.	
Select root directory:     Select archive file:	B <u>r</u> owse B <u>r</u> owse
Projects:	Select All Deselect All Refresh
Options ✓ Searc <u>h</u> for nested projects ✓ <u>C</u> opy projects into workspace ■ H <u>i</u> de projects that already exist in the workspace	
Working sets         Morking sets:	S <u>e</u> lect Click here.
( <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

This application note includes the following projects.

Project name	Contents
r_flash_writer_rx113_switch	Project that selects a file by switch

## RX113 Group

# On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

9. Delete the project (shown as "sample" here) created to use the Smart browser as this is not required.

	C/C++ - e2		New	•		
1	File Edit So		Go Into		as Views Ru	
1111	📑 🗕 🖪 🕼		Open in New Window		ã ▼ 诸 ▼	
ſ	🍋 Project Exp		Сору	Ctrl+C		
	Þ 🈂	ß	Paste	Ctrl+V		
	E sample	×	Delete	Delete	Cli	ck here
			Source	•	1	
	Right clic	k he	retove			
			Rename	F2		
		$\geq$	Import			
		N.	Export			

## 3.4 Modify Configuration

In this project, the configuration file setting and project setting for each FIT module are changed to configure the application. The detail is shown as follows.

Refer to this information when building new project. To use the project imported, go to "4. Verify Operation".

#### 3.4.1 Change Configuration

The configuration files for each FIT module configuring this application require modification.

Refer to the manuals and other files in the **doc** folder for each FIT module for details on the items and the settings in the configuration files.

The places to be changed in the configuration files are shown below.

#### (1) Change the number of drives for USB Mini

Change the number of drives for USB Mini defined by r\_tfat\_driver\_rx configuration file as follows.

[r\_config/r\_tfat\_driver\_rx\_config.h]

```
/* Number of logical drives to be used.
   Setting to 0 : unused memory
        other : number of logical drives
   (USB and SDHI can be used together.)
*/
#define TFAT_USB_DRIVE_NUM (0)
#define TFAT_SDHI_DRIVE_NUM (0)
#define TFAT_USB_MINI_DRIVE_NUM (1)
```

(2) Change the device allocation

Allocate the device to the drive number. In this sample, drive 0 is allocated to USB Mini.

[r\_config/r\_tfat\_driver\_rx\_config.h]

#define TFAT_DRIVE_ALLOC_NUM_0	TFAT_CTRL_USB_MINI
#define TFAT_DRIVE_ALLOC_NUM_1	NULL
#define TFAT_DRIVE_ALLOC_NUM_2	NULL
#define TFAT_DRIVE_ALLOC_NUM_3	NULL
#define TFAT_DRIVE_ALLOC_NUM_4	NULL
#define TFAT_DRIVE_ALLOC_NUM_5	NULL
#define TFAT_DRIVE_ALLOC_NUM_6	NULL
#define TFAT DRIVE ALLOC NUM 7	NULL

#### (3) Change DTC transfer setting

The following DTC definition is described in r\_usb\_basic\_mini\_config.h.

Enable the "USB\_NOUSE" definition, as DTC transfer is not performed in the sample.

[r config/r usb basic mini config.h]

```
/* DTC DEFINE */
#define DTC_USE_PIPE_NUM USB_NOUSE
//#define DTC_USE_PIPE_NUM USB_PIPE1
//#define DTC_USE_PIPE_NUM USB_PIPE3
//#define DTC_USE_PIPE_NUM USB_PIPE4
//#define DTC_USE_PIPE_NUM USB_PIPE5
```

#### (4) Change TFAT setting

TFAT definition is described in r\_usb\_hmsc\_mini\_config.h. To use TFAT, enable the following macro.

```
[r config/r usb hmsc mini config.h]
#define USB_TFAT_USE_PP
```

#### (5) Change Flash API setting

To program code flash, enable the following macro.

```
[r config/r flash rx config.h]
#define FLASH_CFG_CODE_FLASH_ENABLE (1)
```

## RX113 Group On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

## 3.4.2 Change Project Setting

The contents changed from default setting of the project setting is shown. To check the project setting, use the following procedure.

1. 5	Select the project for	the e <sup>2</sup> studio and	l right-click. T	Then, click	[Property (R)].
------	------------------------	-------------------------------	------------------	-------------	-----------------

Image: Second Control       Open in New Window         Image: Project Explorer 12       Copy       Ctrl+V         Image: Project Explorer 12       Image: Project Ctrl+V         Image: Project Explorer 12       Delete       Delete         Source       Move       Rename       F2         Import       Export       Build Project       Clean Project         Close Project       Close Project       Close Unrelated Projects       Image: Project Section	New >	C/C++ - Edit Source Refactor Navigate Sear
Project Explorer S2 Copy Ctrl+C   Paste Ctrl+V   Paste Paste   Paste Ctrl+V   Paste Paste   Paste Past	Open in New Window	• 🗉 🖻 🛎   🐵 • 🗞 • 👼 🗙 🛙
Paste       Ctrl+V         Delete       Delete         Source       Move         Rename       F2         Import       Export         Build Project       Clean Project         Close Project       Close Projects         Build Configurations       Make Targets         Index       Profile As         Debug As       Profile As         Compare With       Profile Team         Compare With       Prestore from Local History         Renerasa Quick Settings       Alt+Q	Copy Ctrl+C	Project Explorer 🛿 📄 😫 🤝
Selete       Delete         Source       Move         Rename       F2         Import       Export         Export       Build Project         Clean Project       Close Project         Close Project       Close Unrelated Projects         Build Configurations       Make Targets         Index       Profile As         Profile As       Profile As         Debug As       Fam         Run As       Team         Compare With       Fam         Restore from Local History       Alt+Q	🕐 Paste Ctrl+V	
Source Move Rename F2 Import Export Build Project Clean Project Close Unrelated Projects Build Configurations Make Targets Index Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	💥 Delete Delete	
Move F2 Rename F2 Import Export Build Project Clean Project Clean Project Close Project Close Unrelated Projects Build Configurations Make Targets Index Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History Restore from L	Source +	
Rename       F2         Import       Export         Build Project       Clean Project         Close Project       Close Project         Close Unrelated Projects       Build Configurations         Make Targets       Index         Index       Index         Exclude from build       Profile As         Debug As       Run As         Team       Compare With         Restore from Local History       Restore from Local History         Renesas Quick Settings       Alt+Q	Move	
Import       Export         Export       Build Project         Clean Project       Clean Project         Close Project       Close Unrelated Projects         Build Configurations       Make Targets         Index       Index         Exclude from build       Profile As         Debug As       Run As         Team       Compare With         Restore from Local History       Alt+Q	Rename F2	
Export         Build Project         Clean Project         Close Project         Close Unrelated Projects         Build Configurations         Make Targets         Index         Exclude from build         Profile As         Debug As         Run As         Team         Compare With         Restore from Local History         Image: Reseas Quick Settings	import	
Build Project         Clean Project         Close Project         Close Unrelated Projects         Build Configurations         Make Targets         Index         Exclude from build         Profile As         Debug As         Run As         Team         Compare With         Restore from Local History         Restore from Local History	Export	
Clean Project Clean Project Close Project Close Project Close Unrelated Projects Build Configurations Make Targets Index Index Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Build Project	
Image: Close Project       F5         Close Project       Close Unrelated Projects         Build Configurations       Make Targets         Index       Make Targets         Exclude from build       Profile As         Debug As       Pobug As         Run As       Team         Compare With       Restore from Local History         Image: Renesas Quick Settings       Alt+Q	Clean Project	
Close Project Close Unrelated Projects Build Configurations Make Targets Index Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History	Refresh     F5	
Close Unrelated Projects Build Configurations Make Targets Index Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Close Project	
Build Configurations Make Targets Index  Exclude from build  Profile As Debug As Run As Run As Team Compare With Restore from Local History  Renesas Quick Settings Alt+Q	Close Unrelated Projects	
Make Targets Index Index Exclude from build Exclude from build Profile As Debug As Run As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Build Configurations	
Index       Exclude from build       Profile As       Debug As       Run As       Team       Compare With       Restore from Local History       Renesas Quick Settings	Make Targets	
Exclude from build Profile As Debug As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Index +	
Profile As Debug As Run As Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Exclude from build	
Debug As     >       Run As     >       Team     >       Compare With     >       Restore from Local History        Image: Setting state of the s	Profile As	
Run As  Team Compare With Restore from Local History Renesas Quick Settings Alt+Q	Debug As 🔸	
Team     >       Compare With     >       Restore from Local History     >       Renesas Quick Settings     Alt+Q	Run As 🕨	
Compare With Restore from Local History Renesas Quick Settings Alt+Q	Team +	
Restore from Local History  Renesas Quick Settings Alt+Q	Compare With	
e <sup>2</sup> Renesas Quick Settings Alt+Q	Restore from Local History	
	e Renesas Quick Settings Alt+Q	
Renesas Tool Settings Alt+T	Renesas Tool Settings Alt+T	
Kun C/C++ Code Analysis	Kun C/C++ Code Analysis	
Command Prompt	Command Prompt	
Bronertier Alta Enter	Properties Alta-Enter Click he	and the second



## RX113 Group On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application



— Project setting of the main program

The main program setting is changed from default setting to the contents listed in Table 3-1 Changed build setting Modules for building, and in Table 3-2 Changed debug setting for debugging

Table 3-1	Changed b	uild setting Modules
-----------	-----------	----------------------

Items	Changed contents	Description
Compiler	Check "Generate debug information"	Outputs the debug information required when
- Object		debugging.
Assembler	Check "Generate debug information"	Outputs the debug information to a
- Object		relocatable file.
Linker	Add "\${workspace_loc:/\${ProjName}/	Requires the setting when using TFAT.
- Input	r_tfat_rx/lib/tfat_rx200_little.lib}" (Note)	(required when using TFAT)
Linker	Remove PResetPRG and PIntPRG from	Requires the setting when using BSP
- Section	section definition (Note)	(required when using FIT)
	Change P Section to P* Section (Note)	Requires the setting when using BSP
		(required when using FIT)
	Add RPFRAM Section after R Section	Requires the setting of the area Flash API
	(Note)	uses (required when using Flash API)
Linker	Map from ROM to RAM	Requires the setting of the area Flash API
- Output	Add PFRAM=RPFRAM to the section	uses (required when using Flash API)
	(Note)	

Note The setting change is required when creating the project that includes each FIT module for BSP, TFAT, and Flash API. For the setting, refer to the manuals, etc. in the **doc** folder of each FIT module.

Table 3-2	Changed	debug	setting
-----------	---------	-------	---------

Items	Changed contents	Description
Debugger	Change "Re-write the on-chip	Required when debugging the program re-
<ul> <li>Debug tool setting</li> </ul>	program ROM" to "Yes"	writing on-chip flash memory.

— Project setting of the sample program

The changed contents from default setting when building is listed in .

 Table 3-3
 Changed build setting (sample)

Items	Changed contents	Description
Linker	Remove PResetPRG and PIntPRG	Requires the setting when using BSP
- Section	from section definition (Note)	(required when using FIT)
	Change P Section to P* Section (Note)	Requires the setting when using BSP
		(required when using FIT)
	Change the address of C_1 section to "0xFFFF 0000"	Define start address to program
	Change the address of FIXEDVECT section to "0xFFFF BF80"	Define start address of fixed vector table
Linker	Change output file/type to "Binary via	Set the file type to write in the USB memory
- Output	absolute"	

Note The setting change is required when creating the project that includes each BSP FIT module. For the setting, refer to the manuals, etc. in the **doc** folder of each BSP FIT module.

## 4. Verify Operation

## 4.1 Build the Project

Use the following procedure to build the project and generate a load module.

#### 1. Click the project to build from the **Project Explorer**.



#### ictor Navigate Search Project Run Window Help Open Project - 🔨 - 🔐 💽 🐔 \* **Close Project** 8 Build All Ctrl+B **Build Configurations** ٠ **Build Project** Click here. Build Working Set ۶ Clean... **Build Automatically** Renesas Tool Settings Alt+T Renesas Quick Settings Alt+Q Update All Dependencies Alt+D Make Target . C/C++ Index ٠ Properties

#### 2. Click **Build project** from the **Project** menu.



## On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

3. When "Build complete" is displayed on the **Console panel**, the build will have completed.

Problems Tesks Console (r\_metwork\_solution\_n64m) Copyright (C) 2011, 2013 Renesas Electronics Corporation Renesas Optimizing Linker Completed 'Finished building target:' make --no-print-directory post-build

 \*Build complete.'

 18:42:40 Build Finished (took 33s.884es)

R01AN2537EJ0102 Rev.1.02 Mar 31, 2017

## 4.2 Prepare for Debugging

## 4.2.1 Configure Hardware

The evaluation board must be configured before starting debugging.

A table of the required equipment and its configuration are shown below.

## Table 4-1 Hardware Configuration

Device	Supplementary Information
Development PC	
RSK	Evaluation board
E1 Emulator	Included in Renesas Starter Kit for RX113
USB memory	Memory that is formatted as either FAT or FAT32.



Figure 4-1 Operating environment example

## RX113 Group On-chip Flash Memory Programming Solution using USB Memory RX Driver Package Application

#### 4.2.2 Set up the RSK

The RSK settings required to operate the main program are shown below.

Set the USB mode (Host/Peripheral). Set jumper J12 to match the setting of USB\_FUNCSEL\_PP in r\_usb\_basic\_mini\_config.h.

#### Table 4-2Jumper Settings

Devices	Jumper	Setting contents
When use USB in host mode.	J12	Short 1 to 2.
(USB_FUNCSEL_PP = USB_HOST_PP)		(*selected this time)
When use USB in peripheral mode.	J12	Short 2 to 3.
(USB_FUNCSEL_PP = USB_PERI_PP)		



Note: The image and the actual configuration will be different.

#### Figure 4-2 RSK Jumper Locations



#### 4.2.3 Prepare USB Memory

Store the binary file of the sample program on the USB memory.

Open the **demo** folder in the project of the main program, and decompress the **sample1.zip** file and save it into the desired location (folder). Copy the **sample.bin** file in the decompressed **sample/release** folder to the USB memory.

Organize  Include in libra	ry      Share with      New folder			10 -		
🚖 Favorites	A Name	Date modified	Туре	Size		
📃 Desktop	🔒 r_bsp	2/5/2016 12:10 PM	File folder			
Downloads	👪 src	2/5/2016 12:10 PM	File folder			
📃 Recent Places	LinkerSubCommand.tmp	10/27/2015 6:11 PM	TMP File		1 KB	
	🗋 makefile	10/27/2015 6:11 PM	File		5 KB	
词 Libraries	C sample abs	10/27/2015 6:11 PM	ABS File		16 KB	
Documents	sample.bin	10/27/2015 6:11 PM	BIN File		48 KB	
J Music	sample.lib	10/27/2015 6:11 PM	LIB File	3	61 KB	
E Pictures	sample.map	10/27/2015 6:11 PM	MAP File		24 KB	
😸 Videos	sample.x	10/27/2015 6:11 PM	X File		16 KB	
	sources.mk	10/27/2015 6:11 PM	MK File		2 KB	
		Copy the	2			
Computer + I	Removable Disk (E:) 🕨	sample1	.bin file	4y Search	Remov.	- 5
Organize 👻 🚺 Open 🛛	New folder			j≣ <b>•</b>		
🔆 Favorites	Name	Date modified	Туре	Size		
Desktop	sample.bin	10/23/2015 5:09 PM	BIN File	-	48 KB	



## 4.3 Debug the Project

Use the following procedure to start debugging the project.

- 1. Connect the development PC to the E1 emulator with a USB cable, and connect E1 emulator to the RSK with user system interface cable.
- 2. Connect the RSK to the adapter and turn on the power.
- 3. Click on [Debug Configurations] in the e2 studio Run menu.



4. Click on [r\_flash\_writer\_rx113.x] under [Renesas GDB Hardware Debugging].

e <sup>2</sup> Debug Configurations			<b>—</b>	
Create, manage, and run configu	ations		T.	
type filter text      Cobug-only      GDB Hardware Debugging      Obb Simulator Debugging      Renesas GDB Hardware Debu      Renesas Simulator Debugain      Renesas Simulator Debugain      Renesas Simulator Debugain	Name: Main ☆ Debugger ► Startup Project: C/ ++ Application: Build (if required) before launching Build configuration: Use Active © Enable auto build @ Use workspace settings	Sommon Source Variables Search Project Disable auto build Configure Workspace Settings Apply	Browse   Revert	Select [r_flash_writer_RX113.x]
?		Debug	Close	

5. Click on [Debug Tool setting] → [System] → [Re-write the on-chip program ROM] and select [Yes], then, click on [Debug].

e <sup>2</sup> Debug Configurations			<b>—</b>	
Create, manage, and run configur	ations		Ť.	
Image: Specific and Specifi	Name Maa Sterre EL/RXV Startue Commo Debu hardware EL/RXV Target I GDB Settings Connection Setting Debug Tool Use Default 10 Filename I O Filename Break Force Hardware Breakpoints Memory Endian Internal Flash Memory Overwrite External Memory Overwrite External Memory Overwrite External Memory Areas Work RAM Start Address Work	Device:	* * * * * * * * * * *	<ul> <li>Select [Debugger]</li> <li>Select [Debug Tool setting]→ [System]→ [Re- write the on-chip program ROM] and select [Yes]</li> </ul>
?		Debug	Ciose	Click on [Debug]

6. When the following message is displayed, click [Yes].



- 7. When the load module download completes, a Debug perspective opens.
- 8. Click [Resume] on the toolbar. The program will be executed and a break will occur at the start of the main function.





9. After the break at the start of the main function, click [Resume] on the tool bar again.

Connect the USB memory.

Data will be read from the USB memory automatically and written to the flash.

If the MCU is restarted and the value displayed on the LCD is counted up as shown in the image below, then it means that the program has completed successfully.





#### 5. Application overview

The sample program counts up the value displayed on the 7 segment display of the RSK RX113 LCD every 1 second. It only requires the on-board memory of the MCU for its operation and no other memory is required.

The memory settings are configured to suit the ROM and RAM capacities of RSK RX113 in the sample program.

- 1. Initializes the LCD driver.
- 2. Displays "SAMPL" on the display part of the LCD.
- 3. Performs 1-second wait processing.
- 4. Displays a counted-up value (0000 to 9999) on the 7 segment display of the LCD.
- 5. Repeats step 3 and 4.

Note: This sample program has been created for evaluating the main program operation. Therefore the program operates on memory areas that are not used by the main program.

## 5.1 Memory structure

0000 0000h **On-chip RAM** 0001 0000h Reserved address 0008 0000h Peripheral I/O register 0010 0000h E2 DataFlash 0010 2000h **Reserved address** 2 6 FFF8 0000h Main program ROM FFFF 0000h Sample program ROM FFFF 8000h Sample program Start information ROM FFFF C000h Main program Start information ROM FFFF FFFFh Note 1 : The memory structure is for RX113 MCU in the RSKRX113. Section address is different depending on the CPU model and main program size used. Also, modification is required depending on the user system.

It shows RX113 MCU memory map on the RSKRX113 used in this application.

Figure 5-1 Memory map

## 6. Main Program Specifications

## 6.1 Files

The main program is included in the program. The source files of the main program is included in the **src** folder.

The main program FIT module name is "r\_flash\_writer\_rx113". The source files are included in **src** folder.

The file structure of the FIT modules and the main program files are listed below.

The main program files are listed in Table 6-1 , and the FIT modules used are listed in Table 6-2 .

#### Table 6-1Main Program Files

Folder	File	Description
	main.c	Source file of main processing
	r_rsk_extention_lcd.c	Source file of LCD driver application processing
	r_rsk_flashdriver.c	Source file of flash processing
	r_rsk_keydriver.c	Source file of key input processing
src	r_rsk_keydriver.h	Header file of key input processing
	r_rsk_lcddriver.c	Source file of LCD driver call processing
	r_rsk_lcddriver.h	Header file of LCD driver call processing
	r_rsk_leddriver.c	Source file of the LED initialization
-	r_rsk_leddriver.h	Header file of the LED initialization
	r_usb_hmsc_apl.c	Source file of total control processing
	r_usb_hmsc_apl.h	Header file of USB driver call processing

## Table 6-2FIT modules used

Folder name	Contents
r_bsp	Board Support Package (BSP) file group
r_config	FIT module config file
r_flash_rx	Flash memory (Flash API) file group
r_lcdc_rx	LCD Controller/Driver (LCDC) file group
r_tfat_driver_rx	M3S-TFAT-Tiny Memory driver interface file group
r_tfat_rx	M3S-TFAT-Tiny FAT file system (TFAT) file group
r_usb_basic_mini	USB Basic Firmware file group
r_usb_hmsc_mini	USB Host Mass Storage Class (USB HMSC) file group

## 6.2 Modules

The following table lists the modules.

## Table 6-3 Modules

File	Module	Description
		Main processing in the main program
main.c	main	Processing to open the flash driver and call for the
		usb_main function.
	usb_main	Initializations and total control processing
	usb_hmsc_driver	HMSC driver task processing
	msc_detach_device	Processing for disconnecting the USB
	msc_connect_wait	Wait processing for connecting the USB device
	mee drive	Processing for connecting the USB device and
		mounting the TFAT file system
	msc_data_ready	Processing for dummy state before read state
	msc data read	Processing for reading from the USB memory and
		writing to the flash
	usb_hsmpl_device_state	USB driver callback processing
r ush hmsc ani c	msc_configured	Processing for notifying of USB connection
1_000_111100_upi.0	msc_drive_complete	Processing for notifying of completion of USB connection
	msc_detach	Processing for notifying of USB disconnection
	usb_mcu_init	USB port initialization
	ush board init	LED and LCD initializations and processing for
		enabling USB interrupts.
	usb_driver_init	USB driver initialization
	apl_init	Control table initialization
	msc_registration	Processing for registering the USB callback function
	msc_event_set	Event setting processing
	msc_event_get	Processing for detecting a switch input and obtaining an event
	msc_led_control	LED control processing
	usb_cpu_LedInitial	LED initialization
r_rsk_leddriver.c	usb_cpu_led_set_bit	LED individual control processing
	usb_cpu_led_set_data	LED whole control processing
	Init_Extension_LCD	LCD port initialization
	usb_lcd_print_14seg_string	Processing for turning on 14 segments
r_rsk_extention_lcd.c	usb_lcd_print_u7seg_digit	Processing for displaying on 7 segments (upper part)
	usb_lcd_print_c7seg_digit	Processing for displaying on 7 segments (middle part)
	dfw_lcd_ascii_calc	Processing for ASCII character conversion
	usb_cpu_LcdInitial	LCD module initialization
r_rsk_lcddriver.c	usb_cpu_LcdDisp	LCD display processing
	string_length_count	Processing for the number of strings information
r_rsk_flashdriver.c	SAMPLE_FLASH_Write	Flash erase/program processing
—	usb_cpu_key_read	Chattering elimination processing
	usb_cpu_sw_data	Processing for detecting a switch input
r_rsk_keydriver.c	usb_cpu_sw1_data	Switch 1 input
	usb_cpu_sw2_data	Switch 2 input
	usb_cpu_sw3_data	Switch 3 input

## 6.3 Flowcharts

#### (1) Main processing

This is the main function which is firstly called from the startup routine of the board support package (BSP).

The function performs flash open processing and calls the usb\_main function.

Although an infinite loop is performed in the usb\_main function, an infinite loop is also performed in the end of this function.





#### (2) Main processing of USB

This is main processing which performs various initializations and controls states of processing order.



Figure 6-2 Main processing of USB (1)



Figure 6-3 Main processing of USB (2)

(3) Port initialization of USB driver

This performs initial port setting for the USB driver.



Figure 6-4 Port initialization of USB driver

(4) Initialization of LCD driver, LED driver and USB driver

This initializes the LCD, LED, and USB drivers.



Figure 6-5 Initialization of LCD driver, LED driver and USB driver

(5) USB driver initialization

This is processing for starting up the USB driver.



Figure 6-6 USB driver initialization

#### (6) Control table Initialization

This initializes the control table.



#### Figure 6-7 Control table initialization

#### (7) Registration of class driver

This registers the class driver.

msc_registration		Sample code processing
Preparation for class driver information		FIT module API
R_usb_hstd_DriverRegistration	Registers the host class driver.	
return		

#### Figure 6-8 Registration of class driver

#### (8) USB HMSC driver task

This is processing for the USB HMSC driver task.



Figure 6-9 USB HMSC driver task

#### (9) Wait for USB device detection

This is the function to wait for USB device connection.



Figure 6-10 Wait for USB device detection

(10) USB device connection, and TFAT file system mount

This reads the USB drive information and mounts the TFAT file system.



Figure 6-11 USB device connection, and TFAT file system mount

#### (11) Setting before read state

This is the dummy control function.





#### (12) Data read from USB memory, and data write to flash memory

This reads the data from the USB memory and writes the data to the flash.



Figure 6-13 Data read from USB memory, and data write to Flash memory

#### (13) USB disconnection

This disconnects the USB device and starts up the sample program.





Figure 6-14 USB disconnection

## Website and Support

Renesas Electronics Website <u>http://www.renesas.com/</u>

Inquiries

http://www.renesas.com/contact/

All trademarks and registered trademarks are the property of their respective owners.



## **Revision History**

		Description		
Rev.	Date	Page	Summary	
1.02	Mar 31, 2017	- First edition issued		

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.
- 2. Processing at Power-on

The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

#### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not
  access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
- 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.
- 5. Differences between Products

Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

— The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.

#### 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 disputes 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, drawing, chart, program, algorithm, application xamples 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others. 4. You shall not alter, modify, copy, or otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products. 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below "Standard" Computers: office equipment: communications equipment: test and measurement equipment: audio and visual equipment: home electronic appliances; machine tools; personal electronic equipment; and 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 Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics. 6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges. 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you. 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations 9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions 10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party. 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products. (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics. (Rev.3.0-1 November 2016) RENESAS **Renesas Electronics Corporation** SALES OFFICES http://www.renesas.com Refer to "http://www.renesas.com/" for the latest and detailed information Renesas Electronics America Inc. 2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A. Tel: +1-408-588-6000, Fax: +1-408-588-6130 Renesas Electronics Canada Limited 9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3 Tel: +1-905-237-2004

Notice

Renesas Electronics Europe Limited Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH Arcadiastrasse 10, 40472 Düsseldorf, Germany Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd. Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd. Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333 Tel: +86-21-2226-0888, Fax: +86-21-2226-0999 Renesas Electronics Hong Kong Limited Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tel: +852-2265-6688, Fax: +852 2886-9022 Renesas Electronics Taiwan Co., Ltd. 13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd. 80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949 Tei: +65-6213-0200, Fax: +65-6213-0300 Renesas Electronics Malaysia Sdn.Bhd. Unit 1207, Block B, Menara Amcorp, Amcorp Tel: +60-3-7955-9390, Fax: +60-3-7955-9510 p Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Renesas Electronics India Pvt. Ltd. No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India Tel: +91-80-67208700, Fax: +91-80-67208777 Renesas Electronics Korea Co., Ltd. 12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea Tel: +82-2-558-3737, Fax: +82-2-558-5141