

RH850/U2A-EVA Group

Start Up Guide(OPBT)

Introduction

This application note is intended to provide Option Bytes setting information to operate RH850/U2A-EVA Group (hereinafter referred to as U2A) in User mode and how to program Option Bytes.

Aim of this document and software is to provide supplemental information for the function on RH850/U2A. It is not intended to implement in the design for mass production.

There is no guarantee to update in this document and software to reflect the latest manual, errata, technical update and development environment. You are fully responsible for the incorporation or any other use of the information of this document in the design of your product or system, and refer to latest manual, errata, technical update and development environment.

Target Device

• RH850/U2A-EVA Group

Target development environment

Integrated development environment (CS+) from Renesas Electronics Corp.

Version : V8.03.00

Device file : DR7F702300.DVF

Integrated development environment (MULTI) from Green Hills Software Corp

Product	· IDE for V800	
FIUUUULI		

Version	: 2019.5.5 ((v 7.1.6)
101011	. 2010.0.0	v 1.1.0)

Target	: V800/RH850
--------	--------------

Device file : DR7F702300.DVF

EXEC file : ExecG3G4_V10201

Reference Document

RH850/U2A-EVA Group User's Manual: Hardware

The Hardware User's Manual provides information about functional and electrical behavior of the device.

At the release time of this application note the following manual version available:

• RH850/U2A-EVA User's Manual(Rev.1.00): R01UH0864EJ0100



Table of Contents

1.	Overview	3
1.1	Note	3
2.	Option Bytes setting	4
2.1	Setting of U2A Normal Operating Mode (OPBT3[1:0])	4
2.2	Setting the parameter related to MainOSC (OPBT10[28],[26:24],[22:20],[18:16],[15:12],[10:8],[6:4],[1],[0])	5
2.3	Setting CPU operating frequency (OPBT11[31:30])	5
2.4	Setting PLL operation (OPBT11[28])	6
2.5	Setting SVR operation (OPBT16[31])	6
3.	How to write Option Bytes to U2A (CS+)	7
3.1	Section setting	7
3.2	Data preparation	9
3.3	Build project and Download to debug tool	10
4.	How to write Option Bytes to U2A (MULTI)	14
4.1	Section setting	14
4.2	Data preparation	
4.3	Build project and Download to debug tool	17
Rev	vision History	22



1. Overview

The U2A differs from the previous generation products (RH850 / P1x and RH850 / F1x) in the following points when operating the U2A.

• Increased new Option Bytes setting items

Option Bytes setting items increased, and there are some new settings that require to pay attention before U2A is activated.

Changed how to write Option Bytes

Previous generation products:

Enter the setting value in hexadecimal for each option byte (32 bits) on the Flash programmer GUI, and the input value is written to the target device via the Flash programmer.

U2A:

The option byte settings are described the its setting value on the source file as well as an user program, and writes HEX data to the target device via the flash programmer.

In this document, the above difference information is explained based on the operation procedure of the actual device.

1.1 Note

In the RH850/U2A series, the option byte setting values are different for each product, so please set the corresponding value. For the details of option byte, please refer to the Hardware User's Manual.



2. Option Bytes setting

2.1 Setting of U2A Normal Operating Mode (OPBT3[1:0])

U2A is possible to set to Serial Programming Mode 0 by setting Option Bytes without setting external pins. The operating mode of U2A is set to Serial Programming Mode 0 at shipping from Renesas. The purpose for this initial setting is to simplify programming to U2A in the customer ECU production process. Therefore, when debugging and stand-alone operation of U2A, the user needs to set bit 1 and 0 of OPBT3 to "00" related to operating mode setting on user program and change operating mode of U2A from Serial Programming Mode.

Bit Position	Bit Name	Function
1, 0	STMSEL1, STMSEL0	These bits select operating mode and startup area. When FLMD0 pin is 0, the operating mode and startup area are selected depending on the combination of the STMSEL1 and STMSEL0. For details, see Section 5, Operating Modes .

Table 51.59OPBT3 Contents

Table 5.1 Model ist

	Table 5.1 Mode List										
	Pins					OPBT				Types of	Flash
	FLMD0	FLMD1	FLMD2	MODE0	TRST	STMSEL1	STMSEL0	Operating Mode	Startup Area	Interface*1	Interface
User Mode	0	х	х	х	0	0	0	Normal Operating Mode	User Area* ²	Nexus/LPD	—
Mode	0	x	x	х	0	0	1	User Boot Mode 0	User Boot Area* ³	Nexus/LPD	_
	0	x	×	x	0	1 Shpping	x Value	Serial Programming Mode 0	Boot firmware	-	CSI
	0	х	x	х	1* ⁵	x	0	Normal Operating Mode	User Area* ²	Nexus/LPD	—
	0	х	х	х	1* ⁵	х	1	User Boot Mode 0	User Boot Area* ³	Nexus/LPD	-
	1	0	×	×	×	×	×	Serial Programming Mode 1	Boot firmware	_	2-wire UART/CSI*4
	1	1	0	х	x* ⁵	x	x	User Boot Mode 1	User Boot Area* ³	Nexus/LPD	—
	1	1	1	0	х	х	х	Boundary Scan Mode	_	BSCAN	—

Caution

If the operating mode setting is not changed from initial setting of OPBT3[1:0] as above, U2A operates normally when the debugger is connected but U2A doesn't operate normally after the debugger is disconnected.



2.2 Setting the parameter related to MainOSC (OPBT10[28],[26:24],[22:20],[18:16],[15:12],[10:8],[6:4],[1],[0])

U2A is set parameters related to MainOSC such as frequency by the Option Bytes. It's necessary to be set bits 28, 26-24, 22-20, 18-16, 15-12, 10-8, 6-4, 1, 0 of OPBT10 related to MainOSC setting properly according to usage condition.

Bit Position	Bit Name	Function
31 to 29	Reserved	Set the value of valid area at the shipping.
28	MOSC_EXCLKINPUT	 Main OSC input clock select. 0: Direct clock input to X1 (EXCLK mode). Main OSC amplifier is disabled. 1: Normal crystal oscillation. Main OSC amplifier is enabled.
26 to 24	MOSC_FREQ[2:0]	Main OSC frequency selection bit 0 0 0 _B : 16 MHz 0 0 1 _B : 20 MHz 0 1 0 _B : 24 MHz 0 1 1 _B : 40 MHz 1 x x _B : setting prohibited (need to configure all bit)
22 to 20	MOSC_AMP_SEL_A [2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation destabilization.
18 to 16	MOSC_AMP_SEL_B [2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation stabilization.
15 to 12	MOSC_CAP_SEL [3:0]	Main OSC trimming configuration These bits control internal capacitance.
11	Reserved	Set the value of valid area at the shipping.
10 to 8	MOSC_RD_SEL_A [2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation destabilization.
6 to 4	MOSC_RD_SEL_B [2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation stabilization.
1	MOSC_SHTSTBY_A	Main OSC trimming configuration This bit controls OSC drivability during oscillation destabilization. MOSC_SHTSTBY_A must be set to 1.
0	MOSC_SHTSTBY_B	Main OSC trimming configuration This bit controls OSC drivability during oscillation stabilization. MOSC_SHTSTBY_B must be set to 0.

Table 51.65OPBT10 Contents

2.3 Setting CPU operating frequency (OPBT11[31:30])

CPU operating frequency for U2A is set by Option Bytes. It's necessary to be set bit 31 and 30 of OPBT11 related to CPU operating frequency according to usage condition.

Table 51.66	OPBT11	Contents
-------------	--------	----------

Bit Position	Bit Name	Function
31, 30	CKDIVMD[1:0]	Products of CPU Frequency & CPU System Clock Setting 0 x $_{\rm B}$: 240 MHz 1 0 $_{\rm B}$: 320 MHz 1 1 $_{\rm B}$: 400 MHz



2.4 Setting PLL operation (OPBT11[28])

Activation or deactivation of PLL at power-on for U2A is set by Option Bytes. It's necessary to be set bit 28 of OPBT11 related to PLL operation setting according to usage condition.

Table 51.66 OPBT11 Contents

Bit Position	Bit Name	Function
28	STARTUPPLL	Start Up of Main OSC and PLL 0: Main OSC and PLL are enabled 1: Main OSC and PLL are disabled

2.5 Setting SVR operation (OPBT16[31])

Activation or deactivation of SVR is set by Option Bytes. It's necessary to be set bit 31 of OPBT16 related to PLL operation setting according to usage condition.

Table 51.70 OPBT16 Contents

Bit Position	Bit Name	Function
31	SVRENABLE	SVR Enable setting. 0: Disabled (default) 1: Enabled CAUTION
		Make sure that the all SVR parameters to be set to OPBT16-23 are correct before enabling SVR. Otherwise, the output voltage of Power MOSFET may be unintentional value.

Caution

Make sure that all SVR parameters to be set to OPBT 16-23 are correct before enabling SVR. Otherwise, the output voltage of power MOSFET may be unintentional value.



3. How to write Option Bytes to U2A (CS+)

This section introduces how to write Option Bytes to U2A using Integrated Development Environment CS+ from Renesas Electronics Corp. (hereinafter referred to as CS+).

3.1 Section setting

The section name setting and the address to be set the section have to be specified the "Configuration Setting Area" on the flash memory in which the reset vector base address and Option Bytes value are allocated.

Figure 3.1 shows the sample source file for setting the Option Bytes displayed on CS +. The section name is set on line 5, and the .dw pseudo instruction is used for setting the reset vector and each option byte value. Refer to "set_csa.asm" of sample program for details.

00
0
0
0
FF
FF
FF
FE
FE [Modified]
OF
FF
C3
FF
FE
FF
8E
FF [Modified]
FD
7Z FF
FF
FF
FF
FF
00





Figure 3.2 shows how to set section address using Link options in CS+. As "Reset Vector (PE0)" in "Configuration Setting Area" is described after section name which is described in line 7 of set_csa.asm, the start address of section name(CSA_SECTION) corresponding to "Reset Vector (PE0)" is specified to the address FF32 1380H (FF32 1000H+ 0380H).

Table 51.10 Base address of Configuration Setting Area in case of Area 0 is valid (FSWASTAT_0.CFGVA=0)

Base Address Name <csak_base> (k = f, b)</csak_base>	Base Address	Bus Group
<csaf_base></csaf_base>	FF32 0800 _H (Configuration Setting Area 0)	P-Bus Group 1
<csab_base></csab_base>	FF32 1000 _H (Configuration Setting Area 1)	P-Bus Group 1

f ... front side (valid). b... back side (invalid).

Table 51.51 Configuration Setting Area (1/2)

Name	Address ^{*9}	State at the shipping ^{*1}	Write Protection ID ^{*2}	Read Protection ID ^{"3}	CSAVOF/ CSAVOFC Number
Reset Vector (PE0)	<csak_base>+ 0380_H</csak_base>	0000 0000 _H	Customer ID A	-	12

ę	🖓 🖓 🖉 🤻 i 🔲 💭 🗣 🔍 🎜 i 💋 Solutic	List	
	Project Tree 🛛 📮 🗙	🖌 set csa.asm 🚰 Property	- x
Sma	2 🕜 🙎 📓	CC.BH Property	+
ŝ	- R u2a16 startup (Project)	Output debug information Yee/Output to t	he or thrit fileV-DEBug)
	R7F702300 (Microcontroller)	Compress debug information No(-NOCOmpre	
۳		Delete local symbol name information	,
	CC-RH (Build Tool)	 Optimization Section Settings 	X
	RH850 E2 (Debug Tool)	Perform optimization at time of linkage	
	= S File	 Optimization(Details) Address 	Section Add
	hoot asm	Output external symbol allocation information file %ResetVectorPE0%	RESET PEO
	biodefine h	✓ Input File	EUNITEL PEO
	weatthil asm	> Object file	
	vectulo.asin	> Binary file %Reset VectorPE17	Kesel_PE1 New Overlay
	vectori.asm	> Symbol definition	EIINTTBL_PE1
	vecttblz.asm	V Output File %ResetVectorPE2?	4 RESET_PE2
	vecttbl3.asm	Output folder	FIINTTRI PE2
	set_csa.asm		
	PE0 (Subproject)	Veset VectorPE32	RESEI_PE3
	— IIII R7F702300 (Microcontroller)	System libraries	EIINTTBL_PE3
	- The CC-RH (Build Tool)	Use standard libraries 0x00020000	text
		Use "Standard/Mathematical Library" function	CSA SECTION
	- Program Analyzer (Analyze Tool)	Check memory smashing on releasing memory	Import
	🗄 🛅 File	Use "Non-local jump Library" function	Event
	- estart0.asm	✓ Output Code	Export
	main0.c	Specify execution start address	
	PEI (Subproject)	Fill with padding data at the end of a section	OK Cancel Help
	R7E702300 (Microcontroller)	Work around overrun fetch No	
	CC-RH (Build Tool)	Generate function list used for detecting illegal indirect function call No	
	S BH850 F2 (Debug Tool)	> List	
	Renoram Analyzer (Analyze Tool)	✓ Section	
		Section start address RESE I_PEU	EINTIBL_PEU/%Reset VectorPEU%RESET_PETEINTIBL_PE1/%
		Section that outputs external defined symbols to the file Section that outputs Section alignment	tputs external defined symbols to the file[U]
	Curriasin	> BOM to BAM manned section BOM to BAM manned section	anioi section[0]
		> Verify	apped section[0]
	PE2 (Subproject)	Message	V
	R/F/02300 (Microcontroller)	Section start address	
	CC-RH (Build Tool)	Specify the section start address	
	RH850 E2 (Debug Tool)	The sample value is set in this property by default. You need to set the appropriate value	
	- Program Analyzer (Analyze Tool)		
	e- 🔰 File	Common Options Compile Options Assemble Options Link Options Hex Out	put Options / I/O Header File Generation Options /
		luteut	
	main2.c	nformation(M0281003) : The device file was undeted. Undete the beader file by se	lecting [Generate]/[beader file] .
	🗄 🕂 🚮 PE3 (Subproject)	0.90b -> V1.10↓	
	R7F702300 (Microcontroller)	information(M0291003) : The device file was updated. Update the header file by se	لب.[Generate I/O header file]
	- 🔨 CC-RH (Build Tool)	nformation(M0291003) : The device file was updated. Update the header file by se	ار.[Generate I/O header file]
		0.906 -> V1.10,J	Lastian Personale I/O have des Cited I
		All Maccades	Tecting [Generate 1/U neader Tile].
		nii i i uuuguu /	· · · · · · · · · · · · · · · · · · ·

Figure 3.2 Setting section address



3.2 Data preparation

It sets Reset vector and each Option Bytes which allocated in Configuration Setting Area by the .dw pseudo instruction as Figure 3.3. The .dw pseudo instruction is the assembler instruction that initialize memory in units of 4 bytes. The comment "[Modified]" indicates where to change the value from the initial value. For details on the .dw instruction, check the CS + help.



Figure 3.3 Preparation of Reset vector, Option Bytes data



3.3 Build project and Download to debug tool

Follow the procedure below for writing Option Bytes to U2A. The set Option Bytes are valid from the next reset release.

① Build execution

Executes build and checks no error comment on CS+ output window.





2 Allowance setting for writing Option Bytes

Sets "Yes" to the item of "Allow downloading to the configuration setting area" on the Download File settings tab of the Debug Tool.

9	🖓 🖓 🖉 🤻 💷 💭 🗣 🥘 🗗 🖉 Solutio	on Li	t		
3	Project Tree 4 X		set_csa.asm 🚰 Property	• 1	×
Sma	2 🕜 🤮 📓	5	BH850 E2 Property	A	-
A N			Download		h
anu	R7F702300 (Microcontroller)	5	Download files	[5]	
<u>a</u>	- 🗛 Boot Loader (Configuration Tool for Multi-co	ŕ	CPU Reset after download	No	
	CC-RH (Build Tool)		Erase flash ROM before download	No	
	RH850 E2 (Debug Tool)		Automatic change method of event setting position	Suspend event	
	E- Ele		Allow downloading to the configuration setting area	Yes	1
	Build tool generated files		Allow downloading to the block protection area	No	
	boot asm		Allow downloading to the security setting area	No	
	indefine h		Allow downloading to the switch area	No	
		ľ	Debug Information		
	vectolo.asin		Execute to the specified symbol after CPU Reset	Yes	
	vecttbillasm		Specified symbol	_main	
	vecttbl2.asm		The upper limit size of the memory usage [MBytes]	500	
	wecttbl3.asm				
	set_csa.asm				
	E PEO (Subproject)				
	🔨 CC-RH (Build Tool)				
	🗄 🗍 File				
	The second se				
	am cstart0.asm				
	main0.c				
	PE1 (Subproject)				
	R7F702300 (Microcontroller)				
	CC-RH (Build Tool)				
	BH850 F2 (Debug Tool)				
	Program Analyzer (Analyze Tool)	Δ	ow downloading to the configuration setting area		
	File	S	ecifies whether to allow downloading to the configuration setting area. Ple	Please reconnect the debug tools when after downloded.	
	The Build tool generated files	Ι.		-	
	Build tool generated mes				
		$\boldsymbol{\lambda}$	Connect Settings / Debug Tool Settings Download File Setting	igs / Hook Transaction Settings /	۳
	DF2 (Culture is at)	Ou	put	ф ;	ĸ
	PZC (Subproject)		Build ended(Error:0, Warning:0)(PEO, DefaultBuild)	,j	7
	R/F/02300 (Wilcrocontroller)		Start build(PE1, DefaultBuild)J		
	CC-KH (Build Tool)		Start build(PE2, DefaultBuild)	μ	
	RH850 E2 (Debug Tool)		Build ended(Error:0, Warning:0)(PE2, DefaultBuild)	ــــــــــــــــــــــــــــــــــــــ	4
	Program Analyzer (Analyze Tool)		Start build(PE3, DefaultBuild) Build ended(Error:0, Warning:0)(PE3, DefaultBuild)		
	in the second seco		Start build(u2a16_startup, DefaultBuild)J		
	🕀 📶 Build tool generated files		Build ended(Error:0, Warning:0)(u2a16_startup, DefaultBu	Build), 	1
	estart2.asm	L,	Lineu(ouccess.o i rojects, Failed.0 Frojects)(Friday,	y, rebruary zo, zozo a.rr.oo rmy	
	main2.c 🗸	ΪĒ	F]		ų
	< >		II Messages / *Build Tool /		÷
		1.1			



Permission setting for writing Reset vector and Option Bytes



③ Download to debug tool

After executing the download to debug tool, the Option Bytes value is written to the "Configuration Setting Area" on the flash memory.

🏽 🚳 Z9-KS) 🔒 🔒 🗿 🕹 X 🐚 🛍 🔊 🤆 🖁	🔺 🦀 💌 🔽 100% 💌 🕅 🕅 Default	uild 🔹 🗐 💐 🥎 🕒 🚱 👘 🌚 🕞 💷 🖓	
: 💎 🖓 🖉 🤻 💷 💭 🗣 🔍 🎜 🛛 🌌 Solut	ion List		
😱 Project Tree 🛛 📮 🗙	set_csa.asm 🚰 Property		- >
S 2 3 2 3			•
7 Diale at a start of the start	RH850 E2 Property		y - +
	✓ Download		
K/F/02300 (Microcontroller)	> Download files	[5]	
- Cy Boot Loader (Configuration Tool for Multi-ci	CPU Reset after download	No	
	Automatic change method of event petting position	NO Support event	
KH850 E2 (Debug Tool)	Allow downloading to the configuration setting area	Yes	
E File	Allow downloading to the block protection area	No	
	Allow downloading to the security setting area	No	
boot.asm	Allow downloading to the switch area	No	
	✓ Debug Information		
	Execute to the specified symbol after CPU Reset	Yes	
	Specified symbol	_main	
wecttbl2.asm	The upper limit size of the memory usage [MBytes]	500	
wecttbl3.asm			
set_csa.asm			
- R PE0 (Subproject)			
R7F702300 (Microcontroller)			
CC-RH (Build Tool)			
RH850 F2 (Debug Tool)			
Program Analyzer (Analyze Tool)			
File			
T Build tool generated files			
Ball sctart0 arm			
DE1 (Culturationt)			
R/F/02300 (Microcontroller)			
CC-RH (Build Tool)			
RH850 E2 (Debug Tool)			
	Allow downloading to the configuration setting area	Disasa sasaa saka dabuu kada udaa afaa damaladad	
😑 🛄 File	Specifies whether to allow downloading to the configuration setting	area. Please reconnect the debug tools when after downloded.	
Build tool generated files			
cstart1.asm	Connect Settings / Debug Tool Settings / Download File	Settings Hook Transaction Settings	•
main1.c	Outrut	/	
🖨 🕂 PE2 (Subproject)	Duiput		Ψ .
	Start build(PE1, DefaultBuild)	μ	1
	Build ended(Error:0, Warning:0)(PE1, DefaultBuild	ل, (
	Start build(PE2, DefaultBuild))	
- Program Analyzer (Analyze Tool)	Start build(PE3, DefaultBuild),J	· · · ·	
E-T File	Build ended(Error:0, Warning:0)(PE3, DefaultBuild	ل (
Build tool generated files	Build ended(Error:0, Warning:0)(u2a16 startup, De	faultBuild)ل	
and cstart2 asm	========= Ended(Success:5 Projects, Failed:0 Projects)(له===============================	
main? c	TEDE1		
×			~
< >>	All Messages Build Tool		•

Figure 3.6

Download execution



After download successfully, the pop screen is shown on CS+ as Fig. 3.7, and the allowance setting of "Allow downloading to the configuration setting area" on the Download File Settings tab of the Debug Tool is changed to "No" automatically.

[ion List	
Project Tree 4 X	🖌 set csa.asm 🚰 Property	▼ X
S 2 3 2 3	RH850 E2 Property	9 - +
z 🖂 🕂 u2a16 startup (Project)*	v Download	
2 R7F702300 (Microcontroller)	> Download files [5]	
Boot Loader (Configuration Tool for Multi-co	CPU Reset after download	
CC-RH (Build Tool)	Erase flash ROM before download No	
RH850 F2 (Debug Tool)	Automatic change method of event setting position Suspend event	
File	Allow downloading to the configuration setting area No	
TR Puild tool generated files	Allow downloading to the block protection area No	
heat asm	Allow downloading to the security setting area No	
	Allow downloading to the switch area No	
iodefine.h	V Debug Information	
vecttbl0.asm	Execute to the specified symbol after CPU Reset Yes	
wecttbl1.asm	Specified symbolmain	
vecttbl2.asm	The upper limit size of the memory usage [MBytes] 500	
🔤 vecttbl3.asm	Warning(W0210002)	
set_csa.asm	Wanning(Woz10002)	
E		
R7F702300 (Microcontroller)	Download was completed, but the following problem may exist.	
CC-RH (Build Tool)	[Direct Warring Cause]	
RH850 E2 (Debug Tool)	Memory write to the area requiring reconnection of the debug tool was executed.	
Program Analyzer (Analyze Tool)	Please reconnect debug tool.(W0617020)	
The second second		
The Puild tool generated files		
and a stratt of generated mes		
cstartu.asm		
mainu.c	+ Create contact info OK Help	
PE1 (Subproject)		
R7F702300 (Microcontroller)		
🔨 CC-RH (Build Tool)		
	Allow downloading to the configuration setting area	
🖮 🎒 File	Specifies whether to allow downloading to the configuration setting area. Please reconnect the debug tools when after download.	
Build tool generated files		
cstart1.asm	Connect Settings / Debug Tool Settings Download File Settings / Hook Transaction Settings /	
main1.c	Connect Sectings A Score for Sectings A source sectings A non- numbered on Sectings	
PE2 (Subproject)	Output	д х
R7E702300 (Microcontroller)	Build ended(Error:0, Warning:0)(PEO, DefaultBuild)	^
CC-RH (Build Tool)	Start build(HeI, DefaultBuild)	
N PH950 E2 (Debug Teel)	Start build(PE2, DefaultBuild),J	
	Build ended(Error:0, Warning:0)(PE2, DefaultBuild),	
Program Analyzer (Analyze Tool)	Start build ende(Error:0, Warning:0)(PE3, DefaultBuild)	
E- File	Start build(u2a16_startup, DefaultBuild)	
	Internet Build ended(Error:U, Warning:U)(UZa)6 startup, VetaultBuild)	
cstart2.asm		
main2.c 🗸	LEOFJ	~
<	All Messages / *Build Tool /	•
		v

Figure 3.7 Download successfully

Writing the Option Bytes could be completed through the above procedure.



4. How to write Option Bytes to U2A (MULTI)

This section introduces how to write Option Bytes to U2A using MULTI as the Integrated Development Environment from Green Hills Software. (hereinafter referred to as MULTI).

4.1 Section setting

The section name setting and the address to be set the section have to be specified the "Configuration Setting Area" on the flash memory in which the reset vector base address and Option Bytes value are allocated.

Figure 4.1 shows the sample source file for setting the Option Bytes displayed on MULTI. The section name is set on line 5, and the .dw pseudo instruction is used for setting the reset vector and each option byte value. Refer to "set_csa.850" of sample program for details.

File	Edit View Bloc	k Tools Version Config	Windows Help	
ß		¢ ⇔ א מי ג		
C:¥	Users¥a5105348¥Docu	ments¥GHS Projects¥U2A16¥Re	ev.1.01¥OPBT¥u2a16_startup_ghs¥set_csa.850	~ 1
1				^
2				
3	set Con	figuration Setting Area	a (Option Bytes, Reset Vector)	
4				
5	.sectio	n "CSA_SECTION", const		
6	.align	4		
7	.dw	0×0000000	Reset Vector (PE0) Shipping : 0x0000000	
8	.dw	0×00000000	Reset Vector (PE1) Shipping : 0x00000000 [Modifi	,ed J
9	.dw	0×00800000	Reset Vector (PE2) Shipping : 0x00800000	
10	.dw	0×00800000	Reset Vector (PE3) Shipping : 0x00800000	
11	.dw	ØXFFFFFFF	Reserved Shipping : 0xFFFFFFF	
12	.dw	0xFFFFFFF	Reserved Shipping : ØXFFFFFFF	
13	.dw	0xFFFFFFF	Reserved Shipping : ØXFFFFFFF	
14	.dw	0XFFFFFFF	Reserved Shipping : 0xFFFFFFF	
15	.dw	0x3FF30010	Option byte 0 (OPBI0) Shipping : 0x3FF30010	
17	.dw	0x70555555	Option byte 1 (OPD11) Shipping : 0x76555555	
10	.uw		Option byte 2 (OPB12) Shipping : 0x7FFFFFF	odl
10	.uw		Option byte 5 (OPBT4) Shipping : 0x816FFFFFF [Houlin	euj
20	.uw	0x6C6C6C61	Deserved Shipping : 0x8666666	
20	.dw	0xFFFF0FC3	Option byte 6 (OPBT6) Shipping : 0xFFFF0FC3	
22	dw.	0xFFFFFFFF	Option byte 7 (OPBT7) Shipping : 0xFFFFFFF	
23	.dw	ØxFEFEFEF	Ontion byte 8 (OPBT8) Shipping : 0xFFFFFFF	
24	. dw	0xFFF1FFFF	Option byte 9 (OPBT9) Shipping : 0xFFF1FFF	
25	.dw	ØxFBFD288F	Option byte 10 (OPBT10) Shipping : 0xFBFD288F	
26	. dw	ØxEFFFFFF	Option byte 11 (OPBT11) Shipping : 0xFFFFFFFF [Modifi	ed1
27	.dw	ØxFFFFFFD	Option byte 12 (OPBT12) Shipping : 0xFFFFFFFD	
28	.dw	ØxFFFFFFF	Option byte 13 (OPBT13) Shipping : 0xFFFFFFFF	
29	. dw	0x00000192	Option byte 14 (OPBT14) Shipping : 0x00000192	
30	.dw	0xFFFFFFF	Reserved Shipping : 0xFFFFFFF	
31	.dw	0x6FFFFFF	Option byte 16 (OPBT16) Shipping : 0x6FFFFFFF	
32	.dw	Øxfffffff	Option byte 17 (OPBT17) Shipping : 0xFFFFFFFF	
33	.dw	0xFFFFFFFF	Option byte 18 (OPBT18) Shipping : 0xFFFFFFF	
34	.dw	0xFFFFFFF	Option byte 19 (OPBT19) Shipping : 0xFFFFFFFF	
35	.dw	0xF088FF00	Option byte 20 (OPBT20) Shipping : 0xF088FF00	
36	.dw	0x000000CC	Option byte 21 (OPBT21) Shipping : 0x000000CC	
37	.dw	0×C000C000	Option byte 22 (OPBT22) Shipping : 0xC000C000	
38	.dw	0xFFFFC000	Option byte 23 (OPBT23) Shipping : 0xFFFFC000	
39				
				~
			Ln 1/39.Col 1	
				1 1 1

Figure 4.1 Setting Section name



Figure 4.2 shows how to set section address and size using link directive file (.ld) in MULTI.

As "Reset Vector (PE0)" in "Configuration Setting Area" is described after section name which is described in line 7 of set_csa.850, the start address of section name(CSA_SECTION) corresponding to "Reset Vector (PE0)" is specified to the address FF32 1380H (FF32 1000H+ 0380H).

Also, as the area is set between "Reset Vector(PE0)"(FF32 1380) and "OPBT23"(FF32 13FC) ,the section size is specified to 0x80 in this example of setting.

Table 51.10 Base address of Configuration Setting Area in case of Area 0 is valid (FSWASTAT_0.CFGVA=0)

Base Address Name <csak_base> (k = f, b)</csak_base>	Base Address	Bus Group
<csaf_base></csaf_base>	FF32 0800 _H (Configuration Setting Area 0)	P-Bus Group 1
<csab_base></csab_base>	FF32 1000 _H (Configuration Setting Area 1)	P-Bus Group 1

f ... front side (valid). b... back side (invalid).

Table 51.51 Configuration Setting Area (1/2)

Name	Address ^{*9}	State at the shipping ¹	Write Protection ID ^{*2}	Read Protection ID ⁻³	CSAVOF/ CSAVOFC Number	
Reset Vector (PE0)	<csak_base>+ 0380_H</csak_base>	0000 0000 _H	Customer ID A	-	12	







4.2 Data preparation

It sets Reset vector and each Option Bytes which allocated in Configuration Setting Area by the .dw pseudo instruction as Figure 4.3. The .dw pseudo instruction is the assembler instruction that initialize memory in units of 4 bytes. The comment "[Modified]" indicates where to change the value from the initial value.

:¥Users¥a	5105348¥Do	cuments¥GHS Pro	ojects¥U	J2A16¥Re	v.1.01¥OPB	T¥u2a16_star	tup_ghs¥se	t_csa.850			~	1
	set Co	nfiguration	Setti	ing Are	a (Option	n Bytes, R	eset Vec	tor)				
	.secti	on "CSA_SEC	TION",	const								
	.align	4										
	.dw	0×0000000	0			Rese	t Vector	(PE0)	Shipping :	0x00000000		
	.dw	0x0000000	0			Rese	t Vector	(PE1)	Shipping :	0x00000000	[Modified]	
	.dw	0x0080000	0			Kese	t Vector	(PE2)	Shipping :	0x00800000		
,	.aw	0x0080000				Kese	t vector	(PE3)	Shipping :	0x00800000		
	. dw	OXFEFFEEEE				Rese	wed		Shipping :	OXFEFFEFE		
	.dw	0xFFFFFFFFF				Kese	rved		Shipping :	0xFFFFFFFFFF		
	dw.	Øverererer	-			Rece	eved		Shinning :	ØxEFFFFFFF		
	. dw	0x3FE30010	a			Onti	on byte	Ø (OPBTØ)	Shinning :	0x3FE30010		
	.dw	0×F0FB000	a			Opti	on byte	1 (OPBT1)	Shipping :	0xF0FB0000		
,	.dw	0x7FFFFFFF	F			Opti	on byte	2 (OPBT2)	Shipping :	0x7FFFFFFF		
8	.dw	0xF1FFFFF	c l			Opti	on byte	3 (OPBT3)	Shipping :	0xF1FFFFFF	[Modified]	
)	.dw	0x0C0C0C0	F			Opti	on byte	4 (OPBT4)	Shipping :	0x0C0C0C0F		
)	.dw	0xFFFFFFF	F			Rese	rved		Shipping :	ØxFFFFFFF		
L	.dw	0xFFFF0FC	3			Opti	on byte	6 (OPBT6)	Shipping :	0xFFFF0FC3		
2	.dw	0xFFFFFFF	F			Opti	on byte	7 (OPBT7)	Shipping :	0xFFFFFFFF		
\$.dw	0xFFFFFFF	E			Opti	on byte	8 (OPBT8)	Shipping :	0xFFFFFFFE		
Ļ.	.dw	0xFFF1FFF	F			Opti	on byte	9 (OPBT9)	Shipping :	0xFFF1FFFF		
5	.dw	0xFBFD288	E			Opti	on byte	10 (OPBT10)	Shipping :	0xFBFD288E		
5	.dw	0xEFFFFFF	F			Opti	on byte	11 (OPBT11)	Shipping :	0xFFFFFFFF	[Modified]	
'	.dw	0xFFFFFFF	D			Opti	on byte	12 (OPBT12)	Shipping :	0xFFFFFFFD		
\$.dw	0xFFFFFFF	F			Opti	on byte	13 (OPBT13)	Shipping :	0xFFFFFFFF		
	.dw	0x0000019	2			Opti	on byte	14 (OPBT14)	Shipping :	0x00000192		
)	.dw	0xFFFFFFF	F			Rese	rved		Shipping :	0xFFFFFFFF		
	.dw	0x6FFFFFF	F			Opti	on byte	16 (OPBT16)	Shipping :	0x6FFFFFFF		
2	.dw	0xFFFFFFF	F			Opti	on byte	17 (OPBT17)	Shipping :	0xFFFFFFFF		
	.dw	ØxFFFFFFF	F			Opti	on byte	18 (OPBT18)	Shipping :	ØxFFFFFFF		
	.dw	ØxFFFFFFF				Opti	on byte	19 (OPBT19)	Shipping :	ØxFFFFFFFF		
	.dw	0xF088FF0	0			Opti	on byte	20 (OPBT20)	Shipping :	0xF088FF00		
	.dw	0×000000C				Opti	on byte	21 (OPBT21)	Shipping :	0x000000CC		
	.dw	0xC000C000	0			Opti	on byte	22 (OPBT22)	Shipping :	0xC000C000		
	.dw	ØXFFFFC00	0			Opti	on byte	23 (OPB123)	Shipping :	0XFFFFC000		
,												
										n 1/39.Col 1		

Figure 4.3 Preparation of Reset vector, Option Bytes data



4.3 Build project and Download to debug tool

Follow the procedure below for writing Option Bytes to U2A. The set Option Bytes are valid from the next reset release.

① Build execution

Executes build and checks no error comment on MULTI output window.



Figure 4.4

Build execution



2 Debug execution

Execute debug on MULTI window.

🎘 C:¥U2A16 Startup¥U2A16_Startup.gp	j - MULTI Project	Manager —	
File Edit View Build Connect D	ebug Tools W	Vindows Help	
💃 💕 🔲 🔩 🖌 🏥 🛍 🗠 ୯	🛠 🖫 🕱 🛛		
Find: V	Deb	2ug (E5)	~
Name	Туре	Options	4
□ U2A16 Startup#U2A16_Startup.gpj □ StartupBuild.gpj □ set_csa.850 □ set_csa.850 □ startup.850 □ startup.2.850 □ startup.PE0.850 □ startup.PE1.850 □ startup.PE3.850 □ main_pe0.c □ main_pe1.c □ main_pe3.c □ startupBuild.map	Top Project Program Assembly Subproject Assembly Assembly Assembly Assembly Assembly C File C File	<pre>-cpu=rh850g4mh -I.¥device -ffunctionspreprocess_linke -bsp generic -I.¥inc -Ig -dual_debug -ignore_debug_refit -g -section '.text=.mytext' -section '.text=.mytext1' -section '.text=.mytext2' -section '.text=.mytext3'</pre>	er_directive_tull erences -object_
<			>
Compiling main_peo.c because ma Compiling main_pel.c because ma Compiling main_pe2.c because ma Compiling main_pe3.c because ma Linking StartupBuild because in Done Build successful (Mon Feb 17 2	ain_pel.o doe ain_pel.o doe ain_pe2.o doe ain_pe3.o doe t does not ex 1:11:13 2020)	s not exist s not exist s not exist s not exist ist	^
Initializing Debugger done.			<u> </u>
Status Info Command Debug contai	ining programs	V850/	RH850

Figure 4.5 Debug execution



③ Allowance setting for writing Option Bytes

Execute the follow command on the cmd tab of MULTI Window for allowance downloading to the configuration setting area.

target FLASHEXTRA config1 on

Execute the follow command on the cmd tab of MULTI Window for checking allowance downloading to the configuration setting area.

target FLASHEXTRA

🗩 C:¥U2A16 Startup¥StartupBuild:0x1 - MULTI Debugger	_		×
File Debug View Browse Target TimeMachine Tools Config Windows Help			
		۵. 🔊	
Target Status ☆ ⇒mmmp0x16c0 _start: □ 850eserv2 Debug Connection ☆ > 0x16c2 _start+0x2:	e200 ff8000	04	Ŷ
			>
Source V File: V Func: _start		~	≑ 🦈
<pre>MULTI> target FLASHEXTRA config1 on MULTI> target FLASHEXTRA Extend Data area(0xff320000-0xff3207ff) access is disabled Configuration Setting area 0(0xff320840-0xff320843) access is disabled Configuration Setting area 1(0xff320844-0xff320fff) access is disabled Configuration Setting area 1(0xff321040-0xff321043) access is enabled Configuration Setting area 1(0xff321044-0xff3217ff) access is enabled Security Setting area 0(0xff321840-0xff32184b) access is disabled Security Setting area 0(0xff32184c-0xff3217ff) access is disabled Security Setting area 0(0xff32184c-0xff3217ff) access is disabled Security Setting area 1(0xff322040-0xff3218fb) access is disabled Security Setting area 1(0xff322040-0xff3227ff) access is disabled Security Setting area 1(0xff32204c-0xff3227ff) access is disabled Block Protection area 0(FPSYS0)(0xff322840-0xff32287f) access is disabled Block Protection area 0(FPSYS0)(0xff32080-0xff32307f) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff3237ff) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff3237ff) access is disabled Block Protection area 1(FPSYS0)(0xff340800-0xff3407ff) access is disabled Block Protection area 0(FPSYS1)(0xff340800-0xff3407ff) access is disabled Block Protection area 0(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 0(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340880-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340880-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340840-0xff3407ff) access</pre>) DTP) DTP) DTP) DTP)		^
Data Flash (Blank check) area(0xff400000-0xff48fff) access is disabled (OTF Extend Data (Blank check) area(0xff520000-0xff5207ff) access is disabled (OT Extend Data (Blank check) area(0xff520800-0xff5217ff) access is disabled (OT Extend Data (Blank check) area(0xff521800-0xff5227ff) access is disabled (OT Extend Data (Blank check) area(0xff522800-0xff5237ff) access is disabled (OT Extend Data (Blank check) area(0xff525000-0xff5237ff) access is disabled (OT Extend Data (Blank check) area(0xff526800-0xff5267ff) access is disabled (OT Extend Data (Blank check) area(0xff526800-0xff5267ff) access is disabled (OT Extend Data (Blank check) area(0xff540000-0xff540ff) access is disabled (OT Extend Data (Blank check) area(0xff540000-0xff540ff) access is disabled (OT Extend Data (Blank check) area(0xff540000-0xff547ff) access is disabled (OT Extend Data (Blank check) area(0xff540000-0xff547ff) access is disabled (OT Extend Data (Blank check) area(0xff540000-0xff547ff) access is disabled (OT Extend Data (Blank check) area(0xff57800-0xff547ff) access is disabled (OT Extend Data (Blank check) area(0xff57800-0xff547ff) access is disabled (OT Extend Data (Blank check) area(0xff57800-0xff5747ff) access is disabled (OT Extend Data (Blank check) area(0xff574800-0xff5747ff) access is disabled (OT Extend Data (Blank check) area(0xff574800-0xff5747ff) access is disabled (OT Extend Data (Blank check) area(0xff574800-0xff5747ff) access is disabled (OT Extend Data (Blank check) area(0xff574800-0xff574fff) acce	2) (P) (P) (P) (P) (P) (P) (P) (P) (P) (P		Ň
Cmd Trg* I/O Py Tfc* In section: .text	STOPPE	ED	



Permission setting for writing Reset vector and Option Bytes



(4) Download to debug tool

After executing the download to debug tool, the Option Bytes value is written to the "Configuration Setting Area" on the flash memory.

×	C:¥U2A	16 Sta	artup ³	¥Start	tupBu	uild:0x1	- N	IULTI	Debu	iggei	r										_			>	<
Fil	e Debu	g V	liew	Brov	vse	Target	Т	ïmeMa	achir	ne	Tool	s (Confi	g V	Vindo	ows	Н	elp							
	🔁 🍋	£	3	⇒,	T			ᡱ 👱	٩	đ	Au M		l₽	₽	0	S	M	(@	0	×	R	۱	熱		
	Target □ 850e □ 9 1 □ 0 □ 0 □ 0 □ 0 □ 0 □ 0 □ 0 □ 0	eser tart Cor Cor	v2 Du upBu e O, e 1, e 2, e 3,	ebug ild R R R	Con for .rtu .rtu .rtu	necti My Ta <mark>µpBui</mark> µpBui µpBui∣	d d d	Status	ppe ed ed	d	☆ テ	₽		Dx16 Dx16 Dx16 Dx16 Dx16 Dx16 Dx16 Dx16	c0 c2 c6 cc cc cc cc cc cc cc cc cc cc cc cc		tari tari tari tari tari tari tari tari	t: t+0; t+0; t+0; t+0; t+0; t+0; t+0; t+0;	x(2: x(6: x(a: x(10: x(1	e f e 4 4 3 0 5 3 f 2 0 0 3 4 4 4 0 0 2 0 0 0 0 0	200 f800i 81f 808 207 806 200 6220i 6220i 6220i 6220i 6220i 800 a00 a1c 8241 7800i 9e0 db2 62a1i	004 014 03e 001 000 024	fde0 fdc0		~
_		-										<		_									_	>	
Sou	irce ~	File	: L					1000		0.00		<u> </u>	Func:	_st	art								<u>~</u>	(= 1	₽
Target: Flash Extra area 0xff321380 to 0x80 byte writing Target: Flash Extra area is updated. Please restart 850eserv2. Target: Flash Memory 0x0 to 0x3240 byte writing Target: Flash Memory 0x3240 to 0x1 byte writing Target: Flash Memory 0x400000 to 0x1000 byte writing Target: Flash Memory 0x800000 to 0x26c0 byte writing Target: Flash Memory 0xc00000 to 0x1000 byte writing Downloading program text and data. Please Wait Download complete. Flash Extra area is updated. Please restart 850eserv2. Flash Extra area is updated. Please restart 850eserv2. Flash Memory 0x0 to 0x3240 byte writing Flash Memory 0x0 to 0x3240 byte writing Flash Memory 0x400000 to 0x1000 byte writing Flash Memory 0x800000 to 0x1000 byte writing Flash Memory 0x240 to 0x1 byte writing Flash Memory 0x240 to 0x1000 byte writing Flash Memory 0x240 to 0x1000 byte writing Flash Memory 0x20000 to 0x26c0 byte writing Flash Memory 0x20000 to 0x1000 byte writing																									
<																								>	•
Cm	d Trg*	<u></u> [/(Py J	_Tfc [*]	In se	ectio	n: .tex	t												STOP	PED			

Figure 4.7 Download execution



After download successfully, the allowance setting is changed to "disabled".

🞢 C:¥U2A16 Startup¥StartupBuild:0x1 - MULTI Debugger	_	
File Debug View Browse Target TimeMachine Tools Config Windows Help		
◀ ሎ Ӻ ラ → ♪ ▶ 🔳 ৫ ় 🎂 👌 🔝 🗄 🕼 💭 🍳 🍳 🍳 🔍	晃 🔳 🕷	. 🕅
Target Status ☆ → more 0x16c0 _start: □ 850eserv2 Debug Connection ✓ ● 0x16c2 _start+0x2: □ StartupBuild for My Tar; ✓ ● 0x16c6 _start+0x6:	e200 ff800004 e81f	^
		>
Source V File: V Func: start		⊻ 🗢 🖈
MULTI> target FLASHEXTRA Extend Data area(Dxff320000-0xff3207ff) access is disabled Configuration Setting area 0(0xff320844-0xff320ff) access is disabled (OTF Configuration Setting area 1(0xff321040-0xff321043) access is disabled (OTP) Security Setting area 1(0xff321040-0xff3217ff) access is disabled (OTP) Security Setting area 0(0xff321840-0xff3217ff) access is disabled (OTP) Security Setting area 0(0xff321840-0xff3217ff) access is disabled (OTP) Security Setting area 1(0xff322040-0xff3227ff) access is disabled (OTP) Security Setting area 1(0xff322040-0xff3227ff) access is disabled Block Protection area 0(PFSYS0)(0xff322840-0xff3227ff) access is disabled Block Protection area 0(PFSYS0)(0xff328080-0xff3227ff) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff32307f) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff3237ff) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff3207ff) access is disabled Block Protection area 1(FPSYS0)(0xff32080-0xff3207ff) access is disabled Block Protection area 1(FPSYS1)(0xff340080-0xff3407ff) access is disabled Block Protection area 0(FPSYS1)(0xff34080-0xff3407ff) access is disabled Block Protection area 0(FPSYS1)(0xff34080-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff34080-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340880-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340880-0xff3407ff) access is disabled Block Protection area 1(FPSYS1)(0xff340880-0xff3407ff) access is disabled CMFf374800-0xff374ff) access is disabled (OTP) Switch area 10xff374800-0xff327fff) access is disabled (CExtend Data (Blank check) area(0xff528000-0xff527ff) access is disabled (CExtend Data (Blank check) area(0xff528000-0xff527ff) access is disabled (CExtend Data (Blank check) area(0xff528000-0xff527ff) access is dis	P) (OTP) (
<		>
Cmd Trg* I/O Py Tfc* In section: .text	STOPPED	

Figure 4.8

Download successfully

Writing the Option Bytes could be completed through the above procedure.



Revision History

		Revision contents						
Rev.	Date	Page	Summary					
0.50	2019.04.08	-	1 st edition					
0.70	2020.03.31	-	Supporting U2A16					
			Adding Chapter 4. How to write Option Bytes to U2A (MULTI)					
1.00	2020.09.30	-	Revision update					
1.01	2021.03.18	-	Modified OPBT2 value.					
			Adding a Section 1.1 Note.					



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

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

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

- 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
- 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
- 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renease Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
- 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 subsidiaries.
- (Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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

