

Smart Analog

R20AN0246EJ0100 Rev.1.00

System development procedure by using SA-Designer (RX family) Apr 3, 2013

Introduction

This application note is intended to explain the steps involved in developing a simple system in an environment that uses the Smart Analog, assuming the RX63N.

- Design of the Analog Front-end Circuit
- Creating Program
- Register and Build the Circuit Data
- Testing

Contents

1.	Introduction	2
1.1	Development Environment	3
1.1	1 Hardware	3
1.1	2 Software	3
2.	Development Procedure	4
2.1	Overview	4
2.1	1 Design of the Analog Front-end Circuit	5
(1)	Starting the SA-Designer	5
(2)	The Design of the New Circuit	5
(3)	Creating a Circuit Diagram	7
(4)	Generation Source File	8
2.1	2 Creating Program	9
(1)	Starting CubeSuite+	9
(2)	New Project	0
(3)	Creating Program	3
2.1	3 Register and Build the Circuit Data1	6
(1)	Register the Source Files to CubeSuite+1	6
(2)	Build1	8
2.1	4 Testing1	9
(1)	Download the Load Module	9
(2)	Registration Variable to the Watch Window	4
(3)	Run Program	6
•		~
3.	Sample Programs	8
(1)	Function main (in addition to the main function of RX63N.c)	8
(2)	Initialization function (Add to RX63N.C)	9
(3)	Interrupt function (Add to intprg.c)	0
(4)	Function SPI (Add to RX63N.c)	Ο



1. Introduction

This application note is intended to explain the steps involved in developing a simple system in an environment that uses the Smart Analog, assuming the RX63N. The system uses a temperature sensor built into the "smart analog IC". Depending on the temperature, change the blink rate of the LED. The system uses "Smart Analog IC" and "GR-SAKURA" as CPU board. The application note explains the procedures of the load module that uses the High-performance Embedded Workshop and the SA-Designer, and the procedures of testing program.



Figure 1 System summary



1.1 **Development Environment**

The application note uses the following development environments.

1.1.1 Hardware

- Host PC
- evaluation board: GR-SAKURA (RX63N), Smart Analog IC500
- E1 emulator



Figure 2 Hardware construction



Figure 3 Constructing RX63N and Analog chip

1.1.2 **Software**

- SA-Designer
- IDE CubeSuite+

(V1.00.00) (Version 1.02.01)



2. **Development Procedure**

2.1 **Overview**

The following instructions describe the construction procedures of the system.

It will be used the CubeSuite+ and the SA-Designer for the construction procedures. Followings are the steps of the system development.



Note: It will be needed to install the CubeSuite+ (At least Version V1.02.00) and the SA-Designer for the program operations.



2.1.1 **Design of the Analog Front-end Circuit**

(1) Starting the SA-Designer

Start the SA-Designer by selecting the [SA-Designer] from the Start menu.

(2) The Design of the New Circuit

For the design of the analog front-end circuit, choose the destination device and the folder.

😂 SA-D)esigr	ier				
Eile	⊻iew	<u>G</u> enerate	<u>T</u> ool	<u>H</u> elp		
💦 Start			C	c' d	1 🚳 1 🕜 1 😐 1	

GO	We recommend reading the tutorial to find out what you can d The tutorial contains the information on how to effectively use	lo with SA-Designer. SA-Designer.
Curben.	M Herosoft Amplifier, Ch3.	Syncronous detection
Design New	Circuit Diagram	Amphilier
GO	Create a new AFE circuit diagram.	
D-0 2 an		ANT THE R. LANS
Open Existin	ng Configuration File	2574 120 2 DAGA
GO	Load an AFE design file of SA-Designer. You can also use this button to load files output by Renesas V	/irtual AFE Designer (VA).
-	6 B loca and comme	

Click the GO in "Design New Circuit Diagram".



Choose the device and the folder for creating the codes in the "New dialog".

New Creation		X
Select device Device name:		
RAA730500Z (Smart Analog IC500)		
Source generation path		
Eolder name:]	
C:\Smart_Analog	<u>B</u> rowse	
C Select compiler		
Compiler for <u>u</u> se:		
CA78K0R (IDE:CubeSuite+)	*	
C Library		
Generate the <u>S</u> PI function.		
Generate the graph-analysis library.		

[Device]	Choose the device "RAA730500Z (Smart Analog IC500)"
[Folder name]	Choose the folder arbitrarily. Choose "Smart_Analog" as above image. Choose the folder as above image. The folder must be existed in the computer.



(3) Creating a Circuit Diagram

Design the analog front-end circuit to use a temperature sensor. Change the settings from the initial state of the circuit diagram as follows.



[Variable Output Voltage Regulator]	Set the switch to "ON".
[Temperature Sensor]	Set the switch to "ON".



(4) Generation Source File

Program the source codes to set the data of the designed circuits. Completion dialog will be displayed when you click the "[Generate] - [Generate Source File]".

😂 (untitled)	(upd	lated)	- SA-Designer							
<u>F</u> ile ⊻iew	Gen	nerate	<u>T</u> ool <u>H</u> elp							
🎲 Start 🛛 🛃	C	<u>G</u> ene	rate Source File	Ctrl+G		730500Z (Smart Analog IC500)				
	C	Gene	rate-> <u>R</u> egister Source File	Ctrl+R						
	💽 Gen		rate->Register-> <u>B</u> uild->Run	Ctrl+B						
		Gene	rate Options							
	SA	-Desig	ner Information(100001))						
	was completed.									
	ОК									

Three "C source files" will be made in the folder that you choose.

😂 Smart_Analog			(
<u>File E</u> dit <u>V</u> iew F <u>a</u> vorites <u>T</u>	ools <u>H</u> elp			7
🕒 Back 🔹 🕥 🕤 🏂 🍃	🔘 Search 🛛 🎼 F	olders 😭	35 🗙 🍤	•
Address 🛅 C:\Smart_Analog			~	r ラ Go
Name 🔺	Size	Туре		
📳 r_sadesigner.h	3 KB	H File		3
🗐 r_sadesigner_reg.c	4 KB	C File		3
📳 r_sadesigner_reg.h	3 KB	H File		3
<				>
3 objects		7.84 KB	😼 My Computer	



2.1.2 **Creating Program**

(1) Starting CubeSuite+

Launch the CubeSuite+, from menu of SA-Designer. It also can be launched from "Windows Start menu".

And you need to install "CubeSuite+" beforehand. Click [Startup IDE] of SA-Designer from "Tool", then "CubeSuite+" will be launched.

💱 (untitled)(updated) - SA-Designer									
File View Generate	Too	l Help							
💫 Start 🚚 🔚 🐔 📒	-	Startup IDE	2	RAA730500Z (Smart Analog IC500)					
		IDE Settings							
		Options							

Open "Menu Window" with pressing [Start] after starting CubeSuite+.

🙆 CubeSuite+ -	[Start]										X
<u>File Edit View Pr</u>	oject <u>B</u> uild <u>D</u> ebug	; <u>T</u> ool <u>W</u> indov	v <u>H</u> elp								
🛛 🚳 <u>S</u> tart 🛛 🛃	j 🗶 🖻 (6 9 9	a 🌲 🐴	_	•		n 📜 😡	ם ו ריי 🕼		Ĵ 23	++ ₹
<u>ା ବ ବ ଜ</u> କ୍ଷ	- B										
Project Tree 🛛 🗶	🔊 🖏 Start									•	• x
2 🕜 🤮 🗷		arn About (GO eate New F GO en Existing	Cube Suite We recomme The tutorial c Project A new project A new project Doads the pro	e + nd reading to ontains the t can be cre t can also to oject of Cub	the tutorial to fir information on ated. be created by re reSuite+. Can a	d out what car now to effective eusing the file c so be opened o	i be done in C aly use CubeSi onfiguration re directly from th	iubeSuite+. uite+. gistered to an	existing projec		
		1	Recent Pro	iente		Fa	vorite Proiecte				<u> </u>
	Output									Ą	×ι
	[EOF]										
	All Messages	/									•
F1 F2	F3	F4	FS	FG	F7	FB	F9	FID	FN	FI2	
									M DISC		



(2) New Project

Create a project workspace in the CubeSuite+.

🙆 CubeSuite+ - [[Start]										X
<u>Fi</u> le <u>E</u> dit <u>Vi</u> ew <u>P</u> ro	ject <u>B</u> uild <u>D</u> ebug	<u>T</u> ool <u>Wi</u> ndo	w <u>H</u> elp								
🔍 🕅 Start 🛛 🛃	📜 X 🖻 C	500	AA & AA	_	-	: Gr G	1	ה א 🗊) (þ) (þ) (þ) se	++ =
9929											
Project Tree 🛛 🗭 🗙	🔍 Start									•	×
2 3 2 2	Cre	GO ate New I GO	CubeSuit We recomme The tutorial c Project A new project A new project	e + nd reading thi ontains the in t can be creat t can also be	e tutorial to fini formation on h ed. created by re	d out what can ow to effective using the file co	be done in Cr y use CubeSu onfiguration rej	ubeSuite+. iite+. gistered to an	existing projec	- • - • - • • • • • • • • • • • • • • •	
Drop here to open the project file(*.mtpj).	Ope Output (BOF)	en Existin	g Project Loads the pro Recent Pro	iects	Suite+. Can als	o be opened d	lirectly from th	e following link	i.	- - -	×
E1 E2	Fi	F4	ES	FR	E7	ER	E9	EIII	FT	632	-
									DISC	ONNECT	

Press "GO" button in "Create New Project".



Set the project information in [Create Project] dialog.

Create Project		X							
Microcontroller:	RX	~							
Using microcontroller:									
(Search microcontroller)	Update								
R5F563NBC×LK(1 R5F563NBD×BG(R5F563NBD×FB(1 R5F563NBD×FC(1 R5F563NBD×FC(1 R5F563NBD×LC(1 R5F563NBD×LC(1 R5F563NBD×LC(1 R5F563NBD×LC(1) R5F563NBD×LC(1) R5F563NBD×LC(1) R5F563NDC×BG(1) R5F563NDC×BG(1)	45pin) 176pin) 44pin) 76pin) 00pin) 77pin) 45pin) 176pin) 44pin) 77pin) 45pin) 176pin) 44pin) 77pin) 44pin) 77pin) 44pin) 77pin) 77pin) 74pin) 74pin) 74pin) 75pin) 74p								
Kind of project:	Application(CC-RX)	~							
Project <u>n</u> ame:	R×63N								
P <u>l</u> ace:	C:\Smart_Analog Browse								
	✓ Make the project folder								
C:\Smart_Analog\RX63N\RX6	i3N.mtpj								
Pass the file composition o	an existing project to the new project								
Project to be passed:	(Input project file to be diverted.)								
Copy composition files in the diverted project folder to a new project folder.									
	<u>Lireate</u> Cancel <u>H</u> elp								

[Microcontroller]	Choose "RX".
[Using microcontroller]	Choose "R5F563NEDxFP(100pin)" in "RX63N".
[Kind of project]	Choose "Applications(CC-RX)".
[Project name]	Type "RX63N" as above image.
[Place]	Choose "Smart_Analog" as above image. Check "Make the project folder".

Press "Create" button.



The Project will be made and be displayed in the tree of the Project Tree panel.

🔞 RX63N - CubeSuite+ - [Proj	ect Tree]
<u>File E</u> dit <u>Vi</u> ew <u>Project Build D</u> ebu	ug <u>T</u> ool <u>Wi</u> ndow <u>H</u> elp
🕅 <u>S</u> tart 🛃 🛛 🤤 👗 🗈	®I♡<
S S & 2 S	
Project Tree 🛛 📮 🗙	Property 📝 RX63N.c 🗸 🗸
2 🕜 🙎 🔳	RX63N.c Property • +
🖃 🕂 RX63N (Project)	🗆 Build
R5F563NBDxFP (Microcontr	Yes
	Set individual compile option No
RX Simulator (Debug Tool)	File type U source
🛓 🗐 File	
dbsct.c	Set as build-target
intprg.c	Selects whether to build this life.
🖳 resetprg.c	
RX63N.c	Build Settings File Information
sbrk.c	Output
🖳 vecttbl.c	
iodefine.h	
sbrk.h	
stacksct.h	
typedefine.h	
vect.h	
_	
<	All Messages /
F1 F2 F3	F4 F5 F6 F7 F8 F9 F10 F1 F12



(3) Creating Program

Program the codes for using the clock settings and the function of A/D. As follows, the codes will be programmed in the RX63N.c and ntprg.c samples which are made by the CubeSuite+ as sample. Refer to "3. Sample Programs" for the programs that you need.

🙆 RX63N - CubeSuite+ - [Proje	ct Tree]
<u>Fi</u> le <u>E</u> dit <u>Vi</u> ew <u>P</u> roject <u>B</u> uild <u>D</u> ebu	j <u>T</u> ool <u>W</u> indow <u>H</u> elp
i 🚳 Start 🛃 🛛 🥛 💥 🐚	≗ା୬୯ାଲ≇≜≜
- 💎 🖓 🖉 🧐 🚱	
Project Tree 🛛 📮 🗙	Property 🕑 RX63N.c 🗸 🗸
2 🕜 🙎 🔳	E RX63N.c Property
	Build Set as build-target Yes Set individual compile option No File type C source
e resetpra.c	Build Settings File Information
sbrk.c	Output q
vecttbl.c	
	All Messages
F1 F2 F3	F4 F5 F6 F7 F8 F9 F10 F11 F12

Open the source file "RX63N.c."



C RX63N - Cubesuite+ - [RX63N.C]
<u>Eile E</u> dit <u>Vi</u> ew <u>P</u> roject <u>B</u> uild <u>D</u> ebug <u>T</u> ool <u>Wi</u> ndow <u>H</u> elp
🔍 Start 🛃 📲 🖄 🗠 🝽 🖉 🖓 🖓 🦀 🖌 💽 🥤 🕼 🖓 🖓 🖓 🖓 👘 🔘 🕑 🗠 🖙 🧯
Project Tree 4 X Property RX63N.c* intprg.c - X
2 ③ 2 I Z 51 1 51 51 ⇒ ~ ∽ Columns -
<pre>RX Simulator (Debug Tool) Program Analyzer (Analyze Ti File File File File File File File Fil</pre>
r sadesigner reg.h
All Messages (*Rapid Build /
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
Line 31/40 Column 39 Insert Western European (Windows) 👬 DISCONNECT

Add procedures to the main function of the source file RX63N.c.

Also add the function hwinit and SPI initialization function to the RX63N.c.

Refer to "3. Sample Programs" for the programs that you need.



🙆 RX63N - CubeSuite+ - [intprg.c]	
<u>File E</u> dit <u>Vi</u> ew <u>Project Build D</u> ebug <u>T</u> ool	Window Help
🕅 <u>S</u> tart 🚚 🛛 🥛 🔏 🗈 🖄 🛩	? (~ ﷺ ♣ ♣ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
😔 😔 🔛 🧐 🍕	
Project Tree 7 🗙	🚰 Property 📲 RX63N.c* 📲 intprg.c* 📲 resetprg.c 🔷 👻 🗙
2 🕜 🙎 🔳	§ § § §]
KX63N (Project)* KSF563NBDxFP (Microcontrol	Line 212 // AD ADIO
CC-RX (Build Tool) RX Simulator (Debug Tool) C Program Analyzer (Analyze To File File File	<pre>213 Evoid Excep_AD_ADIO(void){ } 214 215 // S12AD S12ADIO 216 #include "iodefine.h" 211 extern volatile unsigned short g_temp; 212 extern volatile unsigned into groupit</pre>
<pre>dbact.c dbsct.c intprg.c* RX63N.c* sbrk.c iodefine.b</pre>	<pre>213 extern volatile unsigned int g_count; 213 = void Excep_S12AD_S12ADI0(void) { 220 g_temp = S12AD.ADDR4; 221 g_count++; 222 } 223 // ICU GROUP0 224 = void Excep_ICU_GROUP0(void) { } 225 // ICU_CDOUP1</pre>
sbrk.h	Z20 // ICU GROUPI Z27 Ewold Even ICH (BOURDI (wold) / 1
typedefine.h	Jutput q 🗙
vect.h	[BOF]
<pre>c r cadesigner reg c</pre>	All Messages ARapid Build
F1 F2 F3 F4	F5 F6 F7 F8 F9 F10 F11 F12
	Line 215/658 Column 1 Insert Western European (Windows) 👬 DISCONNECT 🚽

Add processing of Excep_S12AD_S12ADI0 interrupt function of the source file intprg.c.

Refer to "3. Sample Programs" for the programs that you need.



2.1.3 **Register and Build the Circuit Data**

(1) Register the Source Files to CubeSuite+

Register the source file, which is made by the SA-Designer to the project that you created in the CubeSuite+.

🔰 (untitled)(updated) - SA-Designer						
<u>File V</u> iew	Gen	erate <u>T</u> ool <u>H</u> elp				
🎲 Start 🛛 🛃	C	Generate Source File	Ctrl+G	730500Z (Smart Analog IC500)		
	C)	Generate-> <u>R</u> egister Source File	Ctrl+R			
	D	Generate->Register-> <u>B</u> uild->Run	Ctrl+B			
		Generate Options				
SA-Designer Information(100002)						
Generation of an AFE register data source file and registration of the file to a project were completed.						
OK						



The Circuit data source file will be registered in the project tree.

© RX63N - CubeSuite+ - [Property]									
<u>File E</u> dit <u>Vi</u> ew <u>P</u> roject <u>B</u> uild <u>D</u> ebug <u>T</u> c	ol <u>Wi</u> ndow	<u>H</u> elp							
						M) ⊊≣ "			
				_		1 5 44		000	
- Con Con Way 20 43									
Project Tree 7 🗙	Pro 😭	operty [RX63N.c	intprg.c					- ×
2 🕜 🙎 🗷	🔚 intprg	j.c Property							• +
RX Simulator (Debug Tool)	🗆 Build								
	Set as I	build-target	1 . ·		Yes				
🖃 🕥 File	Set indi	ividual comp	ile option		No				
👘 📶 Build tool generated files 🔤	File type	e			L source	2			
🖳 🕒 dbsct.c									
intprg.c									
resetprg.c									
RX63N.c									
sbrk.c									
vecttbl.c									
iodefine.h 📃									
	Set as bu	uild-target							
stacksct.h	Selects wi	hether to bui	ld this file.						
typedefine.h									
vect.h	Build Ce	ttinge	File Informal	tion /					
r_sadesigner_reg.c									
	Output								д X
r_sadesigner_reg.h	[EOF]								
	All Mess	sages 🔶 *	Rapid Build	/					+
F1 F2 F3 F4		-5	FG	F7	FB	F9	FID	FN	FI2
								<mark>a‰</mark> D:	ISCONNECT



(2) Build

Choose "Rebuild Project", then make the load module file.

🙆 RX 63N - CubeSuite+ - [l	RX63N.c]	
File Edit View Project Build	Debug Tool Window Help	
🚳 Start 🛃 🍟 🙀 E	Build Project F7	- ";: 🖬 🖓 ";: 🖏 🗅, "\ 💿 💿 "\ 🖙 ";
i 💎 💎 🖉 🧐 🍕 🕅 F	Rebuild Project Shift+F7	
Project Tree 🔯 🐼	Clean Project	X63N.c* 🗹 intora.c* 🗹 resetpra.c
9 (7) 🙎 😰 F	Rapid Build	
🖃 📑 RX63N (Project) 🌱 🗸	Jpdate Dependencies	
ERSF563NBD×F	Build RX63N	<machine.h></machine.h>
- 🔨 CC-RX (Build T 🗾 🛛	Rebuild RX63N	"rodefine.n" /* RX63N 1/0 define table "r sadesigner reg h" /* Smart Analog include
		"r sadesigner.h" /* Smart Analog include
	LIEAN RADUN	
En III File III E III	Jpdate Dependencies of RX63N	IC_Create(void);
und cool (5top Build Ctrl+F7	IC_Write(smartanalog_t * const p_saic_data); IC Read(smartanalog t * const p_saic_data, smartanalog t * (
🔄 intprg.c* 🏹 E	Build Mode Settings	<pre>it(void);</pre>
resetprg.d 🔁 🕫	- Batch Build	artanalog_t gp_smartanalog_data[];
RX63N.c* 🎬	Sacch Ballatti	/* set Port data for LED
📲 sbrk.c 🏼 🏹 E	Build Option List	ed 1 PORTA.PODR.BIT.B1 /* LED1
vecttbl.c	14 = #define	led 2 PORTA.PODR.BIT.B2 /* LED2
iodefine.h	15 🛱#define	led 3 PORTA.PODR.BIT.B6 /* LED3
sbrk.h	16	
🔚 stacksct.h		
🔤 typedefine.h	Output	ч х
wect.h	[EOF]	
		*Ranid Build
F7 Open F2 Rename F3 Fir	nd N F4 Replac F5 Go	F6 Build & F7 Build P F8 Ignore F9 Set/Del F18 Step 0 F77 Step In F182 Jump t
Menu about build.		Line 157/268 Column 18 Insert Western European (Windows)

[Caution]

The following warning message will be shown after the link is started.

** L1100 (W) Cannot find "L" specified in option "start"

This message can be removed by deleting the section "L" which is specified by default during project creation. To delete this section, go to [CC-RX(Build Tools)] - [Properties] - [Link Options] - [Section] in the project tree.



2.1.4 **Testing**

(1) Download the Load Module

Set the debug tool to use from the "Using Debug Tool" in the project tree of the CubeSuite+.

🙆 RX63N - CubeSuite+ - [Project Tr	ree]	
File Edit View Project Build Debug Tor	ol Window Help	
🏽 🔍 Start 🛃 🛛 🦉 🛣 🐁 👘	୬ ୯ ଲ 🐥 🔺 📩 🔽 🖬 🗒 🖏	ם אין אין וווי וווי וווי ווויי ווויי
S S 💀 🖉 S		
Project Tree 7 🗙	Property AKS3N.c intprg.c resetprg.c	→ ×
2 🕜 🤮 🛢	1 1 -> -> -> -> Columns -	
	Line 1 #include <machine.h> 2 =#include "iodefine.h" 3 =#include "r_sadesigner_reg.h" two Tralest and there is the transferred to the transfer</machine.h>	/* RX63N I/O define table /* Smart Analog include /* Smart Analog include
Property	RX E1(JTAG) rdid);	/
Till Build tool gene rated mes disct.c intera c	<pre>/ vol RX E20(Serial)</pre>	<pre>sic_data); ic_data, smartanalog_t * ;</pre>
resetprg.c	10 ext RX Simulator gp_smartanalog_data[]; 11 E	/* set Port data for LED
sbrk.c	12 Etafine led_0 PORTA.PODR.BIT.B0	/* LED0
iodefine.h	Output	
sbrk.h	[EOF]	<u>^</u>
vect.h	All Messages / *Rapid Build / *Build Tool	<u>×</u>
F7 Open F2 Rename F3 Find N F4 F	Replac	
Uses JTAG connected E1 as the debug tool.		

[RX Simulator(Debug Tool)] - [Using Debug Tool]	Choose the "RX E1(JTAG)".
---	---------------------------



Set the clock and choose "Power target from the emulator" in Property window of Debug Tool.

🔕 RX63N - CubeSuite+ - [Property				
<u>File Edit Vi</u> ew <u>P</u> roject <u>B</u> uild <u>D</u> ebug <u>T</u>	ool <u>Wi</u> ndow <u>H</u> elp			
🗑 🕅 Start 🚚 📰 🎽 🙀 🐴	ାର ଦାୟ କିଲ୍ 🔹 "ଭୋଲେ "ଭୋଲେ ଲାଭାର ର ଲାହେ 💷 "			
1 4 4 W 6 4				
Project Tree 4 🗙	🚰 Property 📓 RX63N.c. 📓 intprg.c. 🔮 resetprg.c. 💌 💌 🗙			
2 🕜 🙎 🛛	🔊 🔊 RX E1(JTAG) Property 📀 +			
🖃 🕂 RX63N (Project)*	🗆 Internal ROM/RAM 🔼			
R5F563NBDxFP (Microcontroller)	Size of internal ROM[KBytes] 1024			
CC-RX (Build Tool)	Size of internal RAM[KBytes] 128			
RX E1(JTAG) (Debug Tool)	Size of Data-lash memory[KBytes] 32			
	Main clock source EXTAI			
🖕 🛐 File	Main clock frequency[MHz] 12.0000			
🚡 📶 Build tool generated files	Allow changing of the clock source on writing internal fla No			
	Connection with Emulator			
intprg.c	Emulator serial No.			
resetprg.c	E Lonnection with Larget Board			
RX63N.c	Supply voltage			
	Communications method JTAG			
vecttbl.c	JTAG clock[MHz] 16.5			
	🖻 Flash 👱			
sbrk.h	Power target from the emulator. (MAX 200mA)			
stacksct.h	Select whether or not to supply power from the emulator.			
typedefine.h				
vect.h	Connect Settings / Debug Tool Settings / Download File Settings / Hook Transaction Settings /			
r_sadesigner_reg.c				
r_sadesigner.n				
<	All Messages *Rapid Build *Build Tool			
F1 F2 F3 F	4 F5 F6 F7 F8 F9 F10 F1 F12			

[Clock] - [Main clock frequency]	Specify "12.0000".
[Connection with Target Board] - [Power target from the emulator]	Choose "YES".



Set the Timer and Memory access in Property window of Debug Tool.

	-										
RX63N - CubeSuite+ - [Property	1										
<u>File E</u> dit <u>Vi</u> ew <u>Project</u> <u>Build D</u> ebug]	<u>ool Window H</u> e	∋lp									
🔍 Start 🚽 릚 🍟 🐰 🖿	こうる	1 40 M	*	•	-	G 📜 🕅	D D H (14) ei Ci 🕺		
- 🖓 🖗 🧐 🧐 -											
Project Tree 🛛 📮 🗙	Proper	ty F	1X63N.c	tprg.c 🗹 resetprg	.c				• x		
2 🕜 🙎 🔳	RX E1(JT	FAG) Prop	perty						•+		
🖃 📝 RX63N (Project)*	Access M	emory W	/hile Running						^		
R5E563NBDxEP (Microcoptroller	Access by	Access by stopping execution									
CC-BX (Build Tool)	Update the	display d	uring execution		Yes			_			
	Update inte	erval[ms]			500						
RX EI(JIAG) (Debug 100)	🛛 🛛 Register										
Program Analyzer (Analyze Tool) PC display	during the	execution		No						
E J File	🗆 Break										
😨 📶 Build tool generated files	Type of bre	eakpoints	to be preferentia	ally used	Hardwar	e break					
dbsct.c	🗆 System										
intprg.c	Debug the	program	re-writing the on	-chip PROGRAM R0	DM No						
resetprg.c	Debug the	Debug the program re-writing the on-chip DATA FLASH					No				
RX63N.c	Execute th	ie specifie	ed routine immed	liately before execut	ion i No						
shrk.c	Execute th	ie specifie	ed routine immed	liately after the user	proj No						
	Usage of tr	ace funct	ion		Trace			0.000 States			
	Operation a	after trace	memory is full		Overwrit	e trace memor	y and continue e	execution			
sbrk.h	Trace data	type			Branch						
stacksct.h	Output time	estamp			No						
typedefine.h											
vect.h	Use 64bit c	counter			NO FO OOO	d					
r_sadesigner_reg.c	Uperating t	requency	[MHZ]		50.000	Ч			~		
- 🚽 r_sadesigner.h - 🔚 r_sadesigner_reg.h	Operating fro	equency ne count l	(MHz) to be displayed a	as time, specify an o	perating frequ	iency from 0.0	1001 to 999.999	l.			
	Quere Connect De	-conigs	Cobbag Tool.		10001110 5000		ook mansaccion	Secongs			
	U RORI								# A		
	[KOB]										
<	All Messag	es (*F	Rapid Build	*Build Tool	-		100	1.5	•		
F1 F2 F3	F4	FS	FG	F7	F8	F9	FHD	FTI	H2		
	C		0		University						

[Access Memory While Running] - [Access by stopping execution]	Choose "YES".
[Timer] - [Operating frequency]	Specify "50.0000".



Set the "debug information" in Property window of Debug Tool.

c											
🙆 RX63N - CubeSuite+ - [Property											
<u>File Edit Vi</u> ew <u>P</u> roject <u>B</u> uild <u>D</u> ebug <u>T</u>	ool <u>Wi</u> ndow <u>t</u>	telp									
🕅 <u>S</u> tart 🛃 🔚 🍟 🔏 🖻 🛍	5 S S S S S S S S S S S S S S S S S S S	a 🚑 🚑		•	÷ 🖓	T	- 🔂 D	<u>۱</u>		≥ H+J 0	93 GJ -
- 💎 🖓 🖉 🧐											
Project Tree 🛛 🕈 🗙	Prope	erty 📝 RX63	N.c 🗹 intp	rg.c/ 🥤 resetp	rg.c						→ X
2 🕜 🙎 🔳	🔊 RX E1(ITAG) Property									•+
PY63N (Project)*	🗆 Downloa	d									
	⊞ Download	files			[1]						
CC DW (D 11 T D	CPU Res	et after downloa	d		Yes						
CC-RX (Build Tool)	Erase flas	h ROM before o	download		No						
RX E1(JTAG) (Debug Tool)	Erase dat	a flash ROM be	fore download	ł	No						
Program Analyzer (Analyze Tool)	Automatic	change metho	d of event se	tting position	Suspend	d event					
🖮 🎒 File	🗆 Debugli	nformation							_		
🛓 📶 Build tool generated files	Execute t	o the specified	symbol after C	PU Reset	No						~
	Specify th	e debugged ov	erlay section		No						
intpra.c											
e resetora.c											
SDFK.C											
vecttbl.c											
iodefine.h											
🔚 stacksct.h											
Lynedefine.h											
vect b											
r_sadesigner_reg.c											
r_sadesigner.h	Execute to	the specified	symbol afte	r CPU Reset							
🔤 r_sadesigner_reg.h	Specifies wł	nether to execu	ite the progra	m until the spec	ified symbol po	sition follow	ving CPU	reset.			
		,									
	Connect S	iettings 🖌 D	ebug Tool Se	ttings 🗎 Dow	nload File Set	ttings 人	Hook Tra	ansaction S	Settings		-
	Output										д X
	[EOF]										-
<	All Messa	nes (*Rapid	l Build 🖌 *E	Build Tool							<u> </u>
EI EE EI	F4	ES	FG	F1	FR	E9	F	m	FTT		np

[Debug Information] - [Execute to the specified symbol after CPU Reset]

Choose "No".



Choose [Download] from the [Debug]. Connect to the Debug Tool for downloading the load module.

🙆 RX63N - CubeSuite+ - [Рго	perty]	
<u>File E</u> dit <u>Vi</u> ew <u>P</u> roject <u>B</u> uild	Deb	ug <u>T</u> ool <u>W</u> indow <u>H</u> elp	
🏼 🔍 Start 🛃 🔚 🍟 λ	D,	Download	- ji 🖬 🖬 ji 🖓 🖣 🐂 🔘 🕑 🗠 i 🖙 📮 🍹
i 💎 🖓 🔬 🧐 🍕	6	Build & Download F6	
Project Tree	00	<u>C</u> onnect to Debug Tool	tra c 📝 resetura c
2 🕜 🙎 🗷	00	<u>H</u> ot Plug-in	
🗉 🕂 RX63N (Project)*	D)	<u>U</u> pload	
🔚 🛤 R5F563NBDxFP (Micr		Disconnect from Debug Tool Shift+F6	<u>m</u>
🔨 CC-RX (Build Tool)			No.
RX E1(JTAG) (Debug		<u>S</u> top Shirt+rs	d No
		<u>G</u> o F5	tting position Suspend event
🖃 🔟 File	D	Ignor <u>e</u> Break and Go F8	2011 Receit No.
dhsct.c	9±	Step In F11	
intprg.c	(I)	Step Over F10	
resetprg.c	è-	Return Out Shift+E11	
	Hes		
sbrk.c		CPO Reser	
vecttbl.c	14	Restart	
iodefine.h			
stacksct.b			
typedefine.h			
vect.h			
	g.c		
r_sadesigner.h		Execute to the specified symbol a	iter CPU Reset
🦾 🔚 r_sadesigner_req	g.h	Specifies whether to execute the pro-	gram until the specified symbol position following CPU reset.
		Connect Settings 🖌 Debug Tool	Settings Download File Settings Hook Transaction Settings /
		Output	д х
		[EOF]	
<		All Messages	*Build Tool /
FI Open H F2 Rename F3	Find	Next FY Replace F5 Go F6 Build	&
Menu about debug.			



(2) Registration Variable to the Watch Window

Register the variables to the watch window for checking the operation of the program.

RX63N - RX E1(JTAG) - Cube	Suite+	- [RX63N.c]						
File Edit View Project Build Debug	g Tool	Window Help						
🏽 🙉 Start 🚚 📄 🍟 🔏 🗈	B	9 (° #1 🏶 🕷	•	- 📜 🖬 🖬		🖗 🗗 🙌 I (■ > > + ↓	s≡ Ç≡ 岸
i 💎 🖓 🚂 🧐 🤻								
Project Tree 🛛 📮 🗙		RX63N.c Mintprg	c/ 🗹	resetprg.c 🔄 Property 🐔 Disassemble1		•	× Local Variables	д х
2 🕜 🙎 🔳	30	1 in the last the las	Colum	ns *			Notation	- Heg 🚆
🖃 🕂 RX63N (Project)*	Line	Address	16			-	Current	-
🛲 R5F563NBD×FP (Microconti	13		[#define led_1 PORTA.POI	R.B	IT.B1	Name	
🔨 CC-RX (Build Tool)	14		[∃#define led_2 PORTA.POI	R.B.	IT.B2		
RX E1(JTAG) (Debug Tool)	16			Effectine red_5 PORTA.POL	к.в.	11.50	≡	
	17			I/* Change the value according	to	the system		
i Di File	18			#define DEF_TMP 20		-		
ia 7∥ Build tool generated fil∈	19		[_/* * * * * * * * * * * * * * *	* *	* * * * *		
dbsct.c	20				1			
intprg.c	21			volatile unsigned short g_tem		E Degister to Wate	-61	
resetprg.c	23			volatile unsigned int g com	4 📷	Register to wat		
RX63N.C	24			volatile unsigned int g tim	1	Register to Anal	ysis Chart	
SDFK.C	25				1	Register Action B	Event	
	26	ffff858c		void main(void)	\sim	Cut	CERTA	
	27		[∃{	đ	Cut	Cun+x	
sorkin	28			weletile showt def.	6	Сору	Ctrl+C	
	30			volatile short del;	œ.	Paste	Ctrl+V	
vect.h	31	ffff858e		hwinit();	-88		Ctrl+F	
	32					C - T-	Chill C	📕 CPU Re
🔤 r_sadesigner.h	33	ffff8591	(<pre>R_SAIC_Create(); </pre>	7.9	GO 10	Ctri+G	д ж
🔚 r_sadesigner_reg.h	34	ffff859d		while (ig temp) {	2	Forward to Next	Cursor Position	^
	36	ffff859c		nop();	5	Back to Last Cur	sor Position	
	37			}	_	Co ha Ulara		
	38	ffff85a3	(_ g_temp_ref = g_temp;	<u>₹</u>	Go to Here		
	39				-	Set PC to Here		
	40			while(1){	2	Jump to Eupction	n F12	× (-
						Tag Jump	Shift+E12	
Fr Upen H FC Hename FS Find N	ext fr	r Heplace		re Build & Fi Build Pr Fe Ignore B		ray sump	5111112	ne Jump to
Line 21/268	Column	30 Insert We	stern E	uropean (Windows) 🔳 BREAK 🛛 🔂 Oxfffi	ft 🖏	Jump to Disasser	mble	

Choose "g_temp" in the "RX63N.c" and click the right button and then choose "Register to Watch1". And register "g_temp_ref" in the same operation.



The variables will be displayed in the "Watch1 window".





(3) Run Program

Check the system operations. "CPU Reset" must be chosen before execute the program.



Choose "CPU Reset" from "Debug" and then choose "Go" to execute the program.



Touch your finger to the microcomputer of "Smart Analog IC500". When you touch the "Smart Analog IC", then the temperature of the microcomputer will increase and get lower value of the variable "g_temp" of A/D conversion. Also, the blink rate of LED will be increased.





3. Sample Programs

The followings are the sample programs which are used in the application note.

(1) Function main (In addition to the main function of RX63N.c)

```
#include <machine.h>
#include "iodefine.h"
#include "r_sadesigner_reg.h"
#include "r_sadesigner.h"
                                             /* RX63N I/O define table
                                                                               */
                                             /* Smart Analog include
                                                                               */
                                             /* Smart Analog include
void R_SAIC_Create(void);
void R_SAIC_Write(smartanalog_t * const p_saic_data);
void R_SAIC_Read(smartanalog_t * const p_saic_data, smartanalog_t * const p_saic_read_buf);
void hwinit(void);
extern smartanalog_t gp_smartanalog_data[];
                                             /* set Port data for LED
                                                                               */
#define led_0
                    PORTA. PODR. BIT. BO
                                             /* LED0
                                                                               */
                    PORTA. PODR. BIT. B1
                                             /* LED1
#define led_1
                                                                               */
#define led 2
                    PORTA. PODR. BIT. B2
                                             /* LED2
                    PORTA. PODR. BIT. B6
                                             /* LED3
#define led_3
#define DEF_TMP 20
volatile unsigned short g_temp
                                        = 0;
volatile unsigned short g_temp_ref;
volatile unsigned int g_count
                                       = 0:
volatile unsigned int g_timeofswitch = 10000;
void main(void)
ł
    volatile short def;
    hwinit();
    R_SAIC_Create();
                                             /* for Smart Analog
                                                                               */
    while(!g_temp) {
        nop();
    g_temp_ref = g_temp;
                                             /* read start condition
                                                                               */
    while(1) {
        def = g_temp_ref - g_temp;
        if ( g_count > g_timeofswitch ) {
            led_0 = ~led_0;
led_1 = ~led_1;
led_2 = ~led_2;
            |ed 3 = |ed 3;
            g_count = 0;
            if ( def > DEF_TMP ) {
                g_timeofswitch = 5000;
            } else {
                g_timeofswitch = 25000;
            }
        }
    }
}
```



(2) Initialization function (Add to RX63N.c)

#define F #define F void hwir	PSW_I_FLG 0x000 PSW_I_CLR 0x000 nit(void)	10000 00000			
set_p	osw(PSW_I_CLR);				
SYSTE	M. PRCR. WORD	= 0xA503;	/* disable Reg	gister protection	*/
SYSTE Syste Syste	EM. MSTPCRA. LONG EM. MSTPCRB. LONG EM. MSTPCRC. LONG	= 0xFFFDFFFF; = 0xFFFDFFFF; = 0xFFFF0000;	/* enable MSTF /* enable MSTF /*	P S12AD P RSPI0	*/ */ */
SYSTE Syste	EM. SCKCR3. WORD EM. MOSCCR. BIT. M	= 0x0200; 0STP = 0;	/* select Mair /* enable Mair	n Clock n Clock	*/ */
MPC. F MPC. F	PWPR.BIT.BOWI PWPR.BIT.PFSWE	= 0; = 1;	/* disable MPC	protection	*/
MPC. F MPC. F MPC. F	PC5PFS. BYTE PC6PFS. BYTE PC7PES. BYTE	= 0x0D; = 0x0D; = 0x0D;	/* set SPI RSF	PCKA/MOSIA/MISOA	*/
MPC. F	P44PFS. BIT. ASEL	= 1;	/* set Smart A	Analog TEMP_OUT	*/
MPC. F MPC. F	PWPR. BIT. BOWI PWPR. BIT. PFSWE	= 1; = 0;	/* enable MPC	protection	*/
PORT1 PORT1 PORT1	. PDR. BIT. B2 . PMR. BIT. B2 . PODR. BIT. B2	= 1; = 0; = 1;	/* init Port f	For SAIC RESET	*/
SYSTE	M. PRCR. WORD	= 0xA500;	/* enable Regi	ster protection	*/
PORTA PORTA	A. PODR. BYTE A. PDR. BYTE	= 0; = 0x47;	/* set Port fo	or LED	*/
PORT4 PORT4	I. PDR. BIT. B4 I. PMR. BIT. B4	= 0; = 0;	/* set Port fo	or TEMP_OUT	*/
Porto Porto	C. PDR. BYTE C. PMR. BYTE	= 0x70; = 0xE0;	/* set Port Ou /* set Port Ge	itput PC4, PC5, PC6 eneral PC5, PC6, PC7	*/ */
PORTO	. PODR. BIT. B4	= 1;	/* set Port PC	C4 for CS	*/
S12AD S12AD S12AD	D. ADCSR. BYTE D. ADANSO. WORD D. ADCSR. BYTE	= 0; = 0x0010; = 0xD0;	/* set STZAD /* clear ADST, /* set ANOO4 /* set ADST,AD	CKS DCS, ADIE	*/ */ */
led_0 led_2)	= 1; = 1;			

set_psw(PSW_I_FLG);

}



(3) Interrupt function (Add to intprg.c)



(4) Function SPI (Add to RX63N.c)

(a) **R_SAIC_Create()**

```
/* R_SAIC_Create();
                                                      */
void R_SAIC_Create(void)
{
  volatile uint16_t w_count;
  PORT1. PODR. BIT. B2 = 0;
                               /* Analog IC Reset
                                                      */
  /* Change the waiting time according to the system */
  for (w_count = OU; w_count < 0x82; w_count++)</pre>
   {
     nop ();
  }
  PORT1. PODR. BIT. B2 = 1;
                              /* Analog IC Reset release
                                                      */
  R_SAIC_Write(gp_smartanalog_data);
}
```



```
(b) R_SAIC_Write()
```

```
/* R_SAIC_Write(gp_smartanalog_data);
                                                                         */
void R_SAIC_Write(smartanalog_t * const p_saic_data)
Ł
   volatile uint8_t adrs;
   volatile uint8_t dat;
   volatile uint8_t wait;
   smartanalog_t *p_saic_write;
   p_saic_write = p_saic_data;
   RSPIO, SPCR, BYTE
                      = 0;
                                        /* clear SPE, SPTIE
                                                                         */
                                         /* set ICU
                                                                         */
   ICU. IPR[102]. BYTE = 1;
                                         /* set interrupt priority S12AD
                                                                         */
   ICU. IR[102]. BIT. IR = 0;
ICU. IR[40]. BIT. IR = 0;
                                        /* clear interrupt S12AD
                                                                         */
                                        /* clear interrupt RSPI SPTI
                                                                         */
                                        /* enable interrupt RSPI SPTI
   ICU. IER[5]. BIT. IENO = 1;
                                                                         */
   ICU. IER[12]. BIT. IEN6 = 1;
                                        /* enable interrupt S12AD
                                                                         */
   RSPIO. SPCR. BYTE
                                        /* set SPMS, MSTR, TXMD
                      = 0 \times 0 B:
                                                                         */
   RSPIO. SPCMDO. WORD
                      = 0x0703;
                                        /* set SPB, LSBF, CPOL, CPHA
                                                                         */
   RSPIO. SPBR
                      = 0x05;
   RSPIO. SPCR. BIT. SPE = 1;
                                        /* set SPE
                                                                         */
   RSPIO. SPCR. BIT. SPTIE = 1;
                                        /* set SPTIE
                                                                         */
   while (p_saic_write->address != 0xff)
       PORTC. PODR. BIT. B4= 0;
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tSKA) */
       ł
           nop():
       }
       adrs = (p_saic_write->address & 0x7f) | 0x80; /* 0x80 data write mode*/
       RSPIO. SPDR. WORD. H = adrs;
                                       /* send SAIC Address data
                                                                        */
       while (ICU. IR[40].BIT. IR == OU); /* wait for CSI send
                                                                         */
       ICU. IR[40]. BIT. IR = OU;
       dat = p_saic_write->data;
       RSPIO. SPDR. WORD. H = dat;
       while (ICU.IR[40].BIT.IR == OU); /* wait for CSI send
                                                                        */
       ICU. IR[40]. BIT. IR = OU;
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tKSA) */
       ł
           nop();
                                         /* SAIC CS=H
       PORTC. PODR. BIT. B4= 1;
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tSHA) */
       ł
           nop();
       p_saic_write++;
   RSPIO. SPCR. BIT. SPTIE = 0;
   RSPIO. SPCR. BIT. SPE = 0;
}
```

```
(c) R\_SAIC\_Read()
```

```
/* R_SAIC_Read(gp_smartanalog_data, saic_read_buf);
                                                                            */
void R_SAIC_Read(smartanalog_t * const p_saic_data, smartanalog_t * const p_saic_read_buf)
Ł
   volatile uint8_t adrs;
   volatile uint8_t wait;
   smartanalog_t *p_saic_write;
   smartanalog_t *p_saic_read;
   p_saic_write = p_saic_data;
   p_saic_read = p_saic_read_buf;
   RSPIO. SPCR. BYTE
                        = 0;
                                           /* clear SPE, SPTIE
                                          /* set ICU
                                                                            */
                                          /* set interrupt priority S12AD
   ICU. IPR[102]. BYTE = 1;
                                                                            */
   ICU. IR[102]. BIT. IR = 0;
ICU. IR[40]. BIT. IR = 0;
                                          /* clear interrupt S12AD
                                                                            */
                                          /* clear interrupt RSPI SPTI
                                                                            */
   ICU. IER[5]. BIT. IENO = 1;
                                          /* enable interrupt RSPI SPTI
                                                                            */
   RSPIO. SPCR. BYTE
                       = 0x09;
                                         /* set SPMS, MSTR
   RSPIO. SPCMDO. WORD = 0 \times 0703;
                                         /* set SPB, LSBF, CPOL, CPHA
                                                                            */
   RSPIO. SPBR
                        = 0 \times 05;
   RSPIO. SPCR. BIT. SPE = 1;
                                          /* set SPE
                                                                            */
   RSPIO. SPCR. BIT. SPTIE = 1;
                                          /* set SPTIE
                                                                            */
   while (p_saic_write->address != 0xff)
       PORTC. PODR. BIT. B4= 0;
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tSKA) */
        {
           nop();
       }
       adrs = (p_saic_write->address) & 0x7f;
       p_saic_read->address = adrs; /* send SAIC Address data
RSPI0.SPDR.WORD.H = adrs; /* send SAIC Address data
                                                                           */
       RSPIO. SPDR. WORD. H = adrs;
                                                                            */
       while (ICU. IR[40].BIT. IR == OU); /* wait for CSI send
                                                                           */
       ICU. IR[40]. BIT. IR = OU;
       RSPIO. SPDR. WORD. H = 0xff;
                                         /* send CSI dummy data
                                                                           */
       while (ICU. IR[40]. BIT. IR == OU);
                                         /* wait for CSI send
                                                                           */
       ICU. IR[40]. BIT. IR = OU;
       p_saic_read->data = (unsigned char)RSPI0.SPDR.WORD.H;
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tKSA) */
           nop ();
       PORTC. PODR. BIT. B4= 1;
                                           /* SAIC CS=H
       for (wait = OU; wait < 10U; wait++) /* SA Stable waiting time (tSHA) */
           nop();
       p_saic_write++;
       p_saic_read++;
   RSPIO. SPCR. BIT. SPTIE = 0;
   RSPIO. SPCR. BIT. SPE = 0;
}
```

Website and Support

Renesas Electronics Website http://www.renesas.com/

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

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



Revision Record

		Description			
Rev.	Date	Page	Summary		
1.00	Apr 3, 2013	_	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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

- 1. Handling of Unused Pins
 - Handle unused pins in accord 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.
- Differences between Products
 Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
 - The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

Notice

- Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
- Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
- 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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 systems; anti-disaster systems; anti-crime systems; and safety 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 (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable 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. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages 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 its 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 be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, 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 use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
- 9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
- 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
- 11. This document may not be 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, or if you have any other inquiries. (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.
- (Note 2) Refessas Electronics product(s) means any product developed of manufactured by or for Reflessas Electronics.



SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc. 2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A. Tei: +1-408-588-6000, Fax: +1-408-588-6130
Renesas Electronics Canada Limited 1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada Tei: +1-905-898-5441, Fax: +1-905-898-3220
Renesas Electronics Europe Limited Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K Tel: +44-1628-651-700, Fax: +44-1628-651-804
Renesas Electronics Europe GmbH Arcadiastrasse 10, 40472 Dusseldorf, Germany Tei: +49-211-65030, Fax: +49-211-6503-1327
Renesas Electronics (China) Co., Ltd. 7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China Tei + 86-10-8235-1155, Fax: +86-10-8235-7679
Renesas Electronics (Shanghai) Co., Ltd. Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China Tei: +86-21-5877-1818, Fax: +86-21-6887-7898
Renesas Electronics Hong Kong Limited Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong Tei: +852-2886-9318, Fax: +852 2886-9022/9044
Renesas Electronics Taiwan Co., Ltd. 13F, No. 363, Fu Shing North Road, Taipei, Taiwan Tei: +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 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia Tei: +60-3-7955-9300, Fax: +60-3-7955-9510
Renesas Electronics Korea Co., Ltd. 11F., Samik Lavied' or Bidg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea Tei: +82-2-558-3737, Fax: +82-2-558-5141