RX62N, M16C/62P

SCI Migration Guide: M16C/62P to RX62N

REU05B0144-0120
Rev.1.20
Sep 30, 2010

Introduction

The following document describes the differences between the SCI modules found on the Renesas RX62N and M16C/62P devices.

Target Device

RX62N

M16C/62P

Contents

1. Features

Table 1.1 - SCI Features -lists the features of the SCI modules found on the RX62N and M16C/62P devices. Differences are highlighted.

Table 1.1 - SCI Features

	Specification	
Item	**RX62N**	**M16C/62P**
Number of channels	6	5
Communications modes:		
Synchronous serial I/O mode	Yes	
Asynchronous serial I/O mode	Yes	
Multi-processor mode	Yes	No
Smart Card mode	Yes	No
I2C mode	No	Yes
9-bit mode	No	Yes
SIM mode	No	Yes
IEBus mode	No	Yes
Collision detection	No	Yes
Hardware flow control (RTS/CTS)	No	Yes
Switchable MSB/LSB first	Yes	Yes
Inverted output	Yes	Yes

2. General Notes

- The peripheral clock (Pϕ) provides the SCI's time base. The improved high-speed core of the RX62N allows the peripheral clock to run at speeds of up to 50 MHz; maximum peripheral clock speed on the M16C/62P is 24 MHz.
- Applications taking advantage of the increased performance of the peripheral clock will have to adjust the serial port baud rate generator settings accordingly.
- I2C mode (Special Mode 1) of the M16C/62P is not available on the RX62N. A separate peripheral (IIC) provides two channels of I2C interface on the RX62N.
- The RX62N has a hardware-enabled multiprocessor mode simplifying applications that require multi-drop communications across a number of MCU's.
- Separate interrupts for transmit data register empty and transmission complete aid in development of applications that use external transceivers.
- 9-bit communications mode is not available on the RX62N; a ninth signalling bit (MP) is sent as part of multi-processor communications mode on the RX62N.
- The RX62N does not support hardware handshaking.

3. References

The hardware manual for the RX62N is:

REJ09B0552: RX62N Group, RX621 Group Hardware Manual

The software manual for the RX62N is:

REJ09B0435: RX Family Software Manual

3.1 Hardware Manual Relevant Chapters

Clock Generation Circuit – for details on how to setup the peripheral clock used by the SCI

I/O Registers – provides a complete listing of all registers

Low Power Consumption – for details on the Module Stop Control Registers

Interrupt Control Unit - for details on enabling interrupts from the SCI to the interrupt controller

I/O Ports – for details on the ICR and Port Function Control registers relevant to pins associated with the SCI

Serial Communications Interface (SCI) – for details on SCI-specific registers and operating modes

CRC Calculator – if the application requires CRC checking

4. Associated Registers

Table 4.1 - Registers Associated with SCI Operation

Name	Description	H/W Manual Chapter(s)
SYSTEM.SCKR	System Clock Control Register	Clock Generation Circuit
SYSTEM.MSTPCRB	Module Stop Control Register B	Low Power Consumption
ICU.IRx	Interrupt Request Register	Interrupt Control Unit
ICU.IERx	Interrupt Request Enable Register	Interrupt Control Unit
ICU.IPRx	Interrupt Priority Register	Interrupt Control Unit
PORTx.DDR	Port Data Direction Registers	I/O Ports
PORTx.ICR	Input Buffer Control Registers	I/O Ports
IOPORT.PFFSCI	Port Function Control Register F	I/O Ports
SCIx.SMR	Serial Mode Register	Serial Communications Interface (SCI)
SCIx.BRR	Baud Rate Register	Serial Communications Interface (SCI)
SCIx.SCR	Serial Control Register	Serial Communications Interface (SCI)
SCIx.TDR	Transmit Data Register	Serial Communications Interface (SCI)
SCIx.SSR	Serial Status Register	Serial Communications Interface (SCI)
SCIx.RDR	Receive Data Register	Serial Communications Interface (SCI)
SCIx.SCMR	Smart Card Mode Register	Serial Communications Interface (SCI)
SCIx.SEMR	Serial Extended Mode Register	Serial Communications Interface (SCI)

4.1 I/O Register Macros

New macros in the iodefine.h for RX family parts make it easier to refer to ICU control registers, module stop registers, DTC enable registers, and interrupt vector numbers by the logical names associated with the peripherals. These macros allow portability across RX family members by hiding specific register and vector numbers. See the documentation contained in iodefine.h for details.

Some examples:

Macro	Usage example
`IR(“module name”, “bit name”)`	`if ( `**`IR(SCI0,TXI0)`**` == 1)…`
`IEN(“module name”, “bit name”)`	**`IEN(SCI0,TXI0)`**` = 1 ;`
`IPR(“module name”, “bit name”)`	**`IPR(SCI0,TXI0)`**` = 0x02 ;`
`MSTP(“module name”)`	**`MSTP(SCI0)`**` = 0 ;`

`VECT(“module name”, “bit name”)`	`#pragma interrupt`
	`(MySciTxIsr(vect=`**`VECT(SCI0,TXI0)`**`))`

4.2 Register Comparison by Function

This section provides a graphic overview of the differences between the SCI registers on the MC16C/62P and the RX62N by grouping similar register bits by color. Bits that share the same color between the M16C and RX perform roughly the same type of function (i.e. green bits start and stop transmitters and receivers, light blue bits are for interrupt control).

M16C/62P SCI Registers

UiTB (i=0 to 2)

b15	b14	b13	b12	b11	b10	b9	b8
							D8

b7	b6	b5	b4	b3	b2	b1	b0
D[7-0]							

UiRB (i=0 to 2)

b15	b14	b13	b12	b11	b10	b9	b8
SUM	PER	FER	OER	ABT			D8

b7	b6	b5	b4	b3	b2	b1	b0
D[7-0]							

UiBRG (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
Baud rate divisor							

UiMR (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
IOPOL	PRYE	PRY	STPS	CKDIR	SMD2	SMD1	SMD0

UiC0 (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
UFORM	CKPOL	NCH	CRD	TXEPT	CRS	CLK1	CLK0

UiC1 (i=0 to 1)

b7	b6	b5	b4	b3	b2	b1	b0
UiERE	UiLCH	0	0	RI	RE	TI	TE

U2C1

b7	b6	b5	b4	b3	b2	b1	b0
U2ERE	U2LCH	U2RRM	U2IRS	RI	RE	TI	TE

RX62N SCI Registers

TDR

b7	b6	b5	b4	b3	b2	b1	b0
D[7-0]							

SSR

b7	b6	b5	b4	b3	b2	b1	b0
		ORER	FER	PER	TEND	MPB	MPBT

RDR

b7	b6	b5	b4	b3	b2	b1	b0
D[7-0]							

BRR

b7	b6	b5	b4	b3	b2	b1	b0
BRR[7:0]							

SMR

b7	b6	b5	b4	b3	b2	b1	b0
CM	CHR	PE	PM	STOP	MP	CKS[1:0]	

SCMR

b7	b6	b5	b4	b3	b2	b1	b0
BCP2				SDIR	SINV		SMIF

SCR

b7	b6	b5	b4	b3	b2	b1	b0
TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	

M16C/62P SCI Registers

UCON

b7	b6	b5	b4	b3	b2	b1	b0
0	RCSP	CLK MD1	CLK MD0	U1RRM	U0RRM	U1IRS	U0IRS

UiSMR (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
0	SSS	ACSE	ABSCS	LSYN	BBS	ABC	IICM

UiSMR2 (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
0	SDHI	SWC2	STAC	ALS	SWC	CSC	IICM2

UiSMR3 (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
DL2	DL1	DL0	0	NODC	0	CKPH	0

UiSMR4 (i=0 to 2)

b7	b6	b5	b4	b3	b2	b1	b0
SWC9	SCLHI	ACKC	ACKD	STSP SEL	STP REQ	RSTA REQ	STA REQ

RX62N SCI Registers

SEMR

b7	b6	b5	b4	b3	b2	b1	b0
			ABCS				ACS0

Legend

🟦	Multiprocessor & 9-bit mode	🟩	Transmitter & Receiver Control
🟨	Mode selection	🟥	Status flags
🟧	Transmit & receive data	🟪	Hardware pin selection & setup
⬜	Clock settings & baud rate control	🩵	Interrupt control

5. Comparison of Asynchronous Mode Setup

The code examples below show typical setup of SCI0 on the M16C/62P versus the RX62N.

Step	M16C/62P (SCI0)	RX62N (SCI0)
1. Cancel SCI stop state		`MSTP(SCI0) = 0 ;`
2. Disable SCI during configuration	`u0c1 = 0x00 ;`	`SCI0.SCR.BYTE = 0x00 ;`
3. Configure I/O pins		
Set Rx port pin as input		`P2.DDR.BIT.B1 = 0 ;`
Set Tx port pin as output	`pd6 |= 0x08;`	`P2.DDR.BIT.B0 = 1 ;`
Enable Rx pin as peripheral input		`P2.ICR.BIT.B1 = 1 ;`
4. Set up clock source & SCI mode	`u0c0 = 0x10 ;`	`SCI0.SMR.BYTE = 0x00 ;`
	`u0mr = 0x05 ;`	
5. Set bit rate	`u0brg = 0x4d ;`	`SCI0.BRR = 0x50 ;`
	`/* Wait 1 bit time */`	`/* Wait 1 bit time */`
6. Clear Rx & Tx buffers	`u0tb = u0rb ;`	
	`u0tb = 0 ;`	
7. Configure Tx interrupt source	`ucon = 0x00 ;`	
8. Set interrupt priority & enable interrupts	`DISABLE_IRQ`	`SCI0.SCR.BIT.RIE = 0x01 ;`
	`s0ric = 0x04 ;`	`SCI0.SCR.BIT.TIE = 0x01 ;`
	`s0tic = 0x04 ;`	`IPR(SCI0, ) = 0x01 ;`
	`ENABLE_IRQ`	`IEN(SCI0, RXI0) = 0x01 ;`
		`IEN(SCI0, TXI0) = 0x01 ;`
9. Enable transmitter & receiver	`u0c1 = 0x05 ;`	`SCI0.SCR.BYTE |= 0x30 ;`

A detailed description of each step follows:

5.1 Step 1 – Cancel SCI stop state (RX62N)

The RX has sophisticated power management circuitry that allows it to minimize power consumption by powering down individual peripherals. By default, the majority of peripherals are powered off after reset and must be explicitly turned on before use. The registers for a peripheral will have indeterminate values until the peripheral is powered, so it is important to enable the peripheral before accessing its registers. Clearing the appropriate bits in the Module Stop Control Registers enables peripherals.

The MSTP macro provided in iodefine.h takes the peripheral's logical name as an argument and accesses the appropriate Module Stop Control Register and bit for SCI0. This must be completed before accessing any registers in SCI0.

5.2 Step 2 – Disable SCI during configuration (M16C/62P and RX62N)

To prevent inadvertent interrupts, the SCI's transmitter and receiver should be disabled while carrying out the remaining configuration steps.

5.3 Step 3 – Configure I/O Pins (M16C/62P and RX62N)

All port pins are defined as inputs after reset. In this step the transmit pins used by the SCI are configured as outputs by setting bits in the Data Direction Register (DDR) of the port. The RX62 also requires the input buffer to be enabled in the Input Buffer Control Register (ICR) to gate the external signal to the peripheral.

5.4 Step 4 – Set up clock source & SCI mode (M16C/62P and RX62N)

The registers in this step are used to set the clock source that drives the SCI peripheral and other basic serial parameters such as number of data bits, parity, and number of stop bits. The M16C/62P has more configuration registers than the RX62N, but for basic asynchronous operation the settings shown here are sufficient.

5.5 Step 5 – Set bit rate (M16C/62P and RX62N)

The SCI in each part can be clocked from various sources. The selected clock in combination with the Bit Rate Register (BRG or BRR) defines the bit rate used for transmission and reception. See the data sheet for each part for the formula used to calculate bit rates.

5.6 Step 6 – Clear Rx & Tx buffers (M16C/62P)

The receive and transmit registers contain indeterminate data after reset on the M16C/62P. Data in the receive data register (u0rb) is discarded by reading the register. A zero is written to the transmit data register (u0tb) to initialize it.

5.7 Step 7 – Configure Tx interrupt source (M16C/62P)

The SCI transmitter is double-buffered: data written to the transmit buffer is copied to a shift register for transmission. Once the data from the transmit buffer is copied to the shift register the transmit buffer is available to receive more data; this event generates the transmit buffer empty (TI) condition that can be used to trigger an interrupt.

Data in the shift register is shifted out of the chip at the defined bit rate. Once the last bit of data, plus any parity bit, has been shifted out of the chip a transmit complete (TXEPT) can be used to trigger a transmitter interrupt.

The M16C/62P has a single interrupt request for the transmitter. One of the two transmitter events can be selected to generate the request: transmit buffer empty (TI) or transmit completed (TXEPT).

On the RX62N, each condition (transmit buffer empty and transmit complete) can generate a separate interrupt.

5.8 Step 8 – Set interrupt priority & enable interrupts (M16C/62P and RX62N)

The interrupt controller on the M16C/62P is slightly less sophisticated and requires fewer steps to enable serial interrupts. Setting an interrupt priority level greater than zero enables the SCI interrupts; an interrupt level is set for both the receiver interrupt (s0ric) and transmitter interrupt (s0tic).

The RX62N requires additional steps to fully engage the interrupt mechanism. Enable bits (SCI.SCR.RIE, SCI.SCR.TIE) in the SCI peripheral must be set to gate the SCI signals to the ICU. Next, assign a priority level in the ICU’s interrupt priority register (ICU.IPR). Finally, individual enable bits in the ICU’s Interrupt Enable Registers (ICU.IER) must be set allow interrupts to be triggered. (Note that in the example above the IPR and IEN macros from iodefine.h are used to set the control bits in the ICU).

Note that the address of a valid interrupt service routine (ISR) must be installed at the appropriate location in the interrupt vector table. The ISR must follow special entry and exit procedures to save and restore processor context. This is done automatically by the Renesas compiler using the `#pragma interrupt` construct.

5.9 Step 9 – Enable transmitter & receiver (M16C/62P and RX62N)

At this point the hardware was been properly configured and the transmitter and receiver may be enabled. Note that any application-level initialization necessary (such as setting up receive and transmit buffers) should be completed before enabling the transmitter and receiver.

6. Comparison of Polled Asynchronous Mode Communications

Event	M16C/62P (SCI0)	RX62N (SCI1)
Receiving a character		
	`unsigned short u0rbCopy;`	`unsigned char rdrCopy ;`

Check flag	`if (ri_u0c1 == 1)`
	`{`
		`if (IR(SCI1,RXI1) == 0x01)`
		`{`

Read data	`u0rbCopy = u0rb ;`	`rdrCopy = SCI1.RDR ;`

Clear flag	`// RI is cleared by read`
	`// of Receive Buffer`
	`// Register (U0RB)`
		`IR(SCI1,RXI1) = 0 ;`

M16C/62P: Data in LSB, errors in MSB RX62N: Check SSR register for errors	`}`	`}`
Transmitting a character		
Wait for space in Tx data register	`while (ti_u0c1 == 0) ;`	`while (IR(SCI1,TXI1) == 0) ;`

Clear flag	`// TI is cleared by write`
	`// to Transmit Buffer`
	`// Register (U0TB)`
		`IR(SCI1,TXI1) = 0 ;`

Write a character to the data buffer	`u0tb = txChar ;`	`SCI1.TDR = txChar ;`
Transmission is complete		
Check flag	`if (txept_u0c0 == 1)`	`if (IR(SCI1,TEI1) == 1)`

Clear flag	`{`
	`u0c1 = 0x00 ;`
		`{`
		`IR(SCI1,TEI1) = 0 ;`

Perform other actions such as turning off a transceiver	`.`
	`.`
	`.`
	`// Other actions`
	`}`
		`.`
		`.`
		`.`
		`// Other actions`
		`}`

Communications Error		

	`unsigned short u0rbCopy;`
	`unsigned char rxChar ;`
		`unsigned char ssrCopy ;`
		`unsigned char rxChar ;`

Read status flags (and data on M16C/62P)	`u0rbCopy = u0rb ;`	`ssrCopy = SCI1.SSR.BYTE ;`

Check error flags	`if (u0rbCopy & 0x8000)`
	`{`
	`// Process errors here`
	`}`
		`if (ssrCopy & 0x38)`
		`{`
		`// Process errors here`
		`}`

M16C/62P: No error, process received data in LSB	`else`
	`{`
	`rxChar = (char) _`
	`(u0rbCopy & 0xFF);`
	`}`	

Clearing Overrun Errors		

M16C/62P: Disable and re-enable receiver to clear the error
RX62N: Clear overrun error bit, purge receive data buffer
	`// Overrun error`
	`if (u0rbCopy & 0x1000)`
	`{`
	`re_u0c1 = 0 ;`
	`re_u0c1 = 1 ;`
	`}`
		`// Overrun error`
		`if (ssrCopy & 0x20)`
		`{`
		`SCI1.SSR.BIT.ORER = 0 ;`
		`IR(SCI1,RXI1) = 0 ;`
		`rxData = SCI1.RDR ;`
		`}`

6.1 Notes on sample polling code

On the M16C/62P, the receive error flags are contained in the same 16-bit register as the receive data. Perform a single 16-bit read of the register and then look at the MSB to see if any errors are associated with the received character. The bits for parity error and framing error are automatically cleared when the register is read; disabling and re-enabling the receiver clears the overrun error flag.

On the RX62N, error flags are in a separate register from the received data. This can cause race conditions when using the SCI in polled mode; it is strongly recommended that interrupt driven communications be used for the SCI.

IMPORTANT: The RIE and TIE bits in the SCI Control Register (SCR) must be set to a 1 to allow polling of the Interrupt Request (IR) bits in the ICU.

Website and Support

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

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Feb.16.2010	—	First edition issued
1.10	Apr.13.2010	—	Updated to new Renesas Electronics format.
1.20	Sep.30.2010	4	Added section 3.3
		5-7	Added section 4.2

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in 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.

5. Differences between Products

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

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
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.

RENESAS

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.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +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: +44-1628-585-100, Fax: +44-1628-585-900

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

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +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
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -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
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0