To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

SH7206 Group

Example of Setting the SCIF for UART Reception

Introduction

This application note presents an example of configuring the serial communication interface with FIFO (SCIF) for UART reception.

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

- Channel 0 of the SCIF is initialized as a reception module in UART mode, and received data is stored in a buffer.

1.2 MCU Functions Used

- SCIF channel 0

1.3 Conditions for Application

- MCU: SH7206 (R5S72060)
- Operating frequency: Internal clock: 200 MHz
  Bus clock: 66.67 MHz
  Peripheral clock: 33.33 MHz
- C compiler: SuperH RISC engine Family C/C++ Compiler Package: Version 9.00
  (from Renesas Technology Corp.)
- Compiler options: Default setting of HEW (-cpu=sh2a -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Note

The operation of the sample program in this application note was confirmed with the configuration specified in the application note "Example of SH7206 Initial Configuration". Please refer to that note in combination with this one.

2. Description of Sample Application

This sample application uses the serial communication interface with FIFO (SCIF).

2.1 Summary of MCU Functions Used

In asynchronous mode (UART), the SCIF adds start and stop bits to character data and transmits/receives them, while establishing synchronization for each character. The start and stop bits indicate the start and end of communication, respectively. Either the internal clock signal or the external clock signal input from the SCK pin can be selected as a clock source. Communication mode can be specified in terms of data transfer format, transfer rate, and others.

Table 1 summarizes the features of UART communication by the SCIF, and figure 1 shows a block diagram of the SCIF.

Table 1 Summary of SCIF (Asynchronous Mode)

Item	Function
Number of channels	4 (SCIF0 to SCIF3)
Clock source	Internal clock: Pϕ, Pϕ/4, Pϕ/16, or Pϕ/64 (Pϕ: Peripheral clock) External clock: Clock input from the SCK0 to SCK3 pins (The SCIF is driven by the clock input from the pin divided by 16.)
Data format	Transfer data length: 7 or 8 bits Bit transfer order: LSB first Start bit: 1 bit Stop bit: 1 or 2 bits Parity bit: Even, odd, or none
Baud rate	Internal clock: 62.94 bps to 1031.25 kbps (Pϕ = 33 MHz) External clock: 515.625 kbps at maximum (when Pϕ = 33 MHz and externally input clock = 8.25 MHz)
Error detection	Parity, framing, and overrun errors
Interrupt requests	• Transmit FIFO data empty interrupt (TXI) • Break interrupt (BRI) • Receive FIFO data full interrupt (RXI) • Receive error interrupt (ERI)
Others	• Break can be detected. • Clock supply to unused channels can be stopped to save power. • Built-in modem control functions (RTS and CTS) are available on channel 3. • The number of valid data bytes in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.

Note: * For details on the SCIF, refer to section 15, Serial Communication Interface with FIFO (SCIF), in the SH7206 Group Hardware Manual.

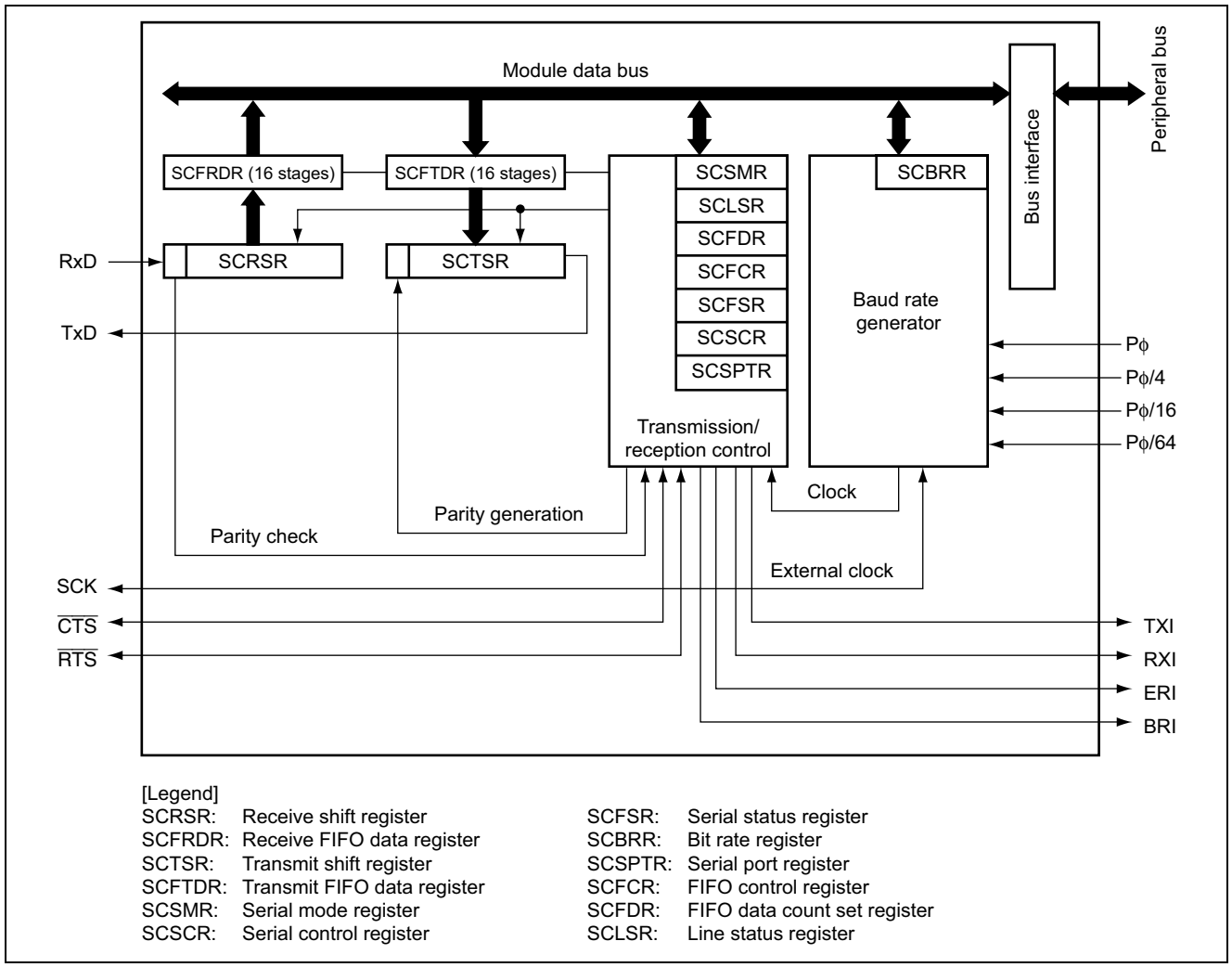


Figure 1 Block Diagram of SCIF

2.2 Procedure for Setting the MCU Modules

This section describes the basic setting procedures of the SCIF for UART mode (asynchronous) communication. Figures 2 and 3 show an example flow of initial settings for UART mode reception, and figure 4 shows an example flow of UART mode reception processing.

For details on the settings of individual registers, refer to the SH7206 Group Hardware Manual.

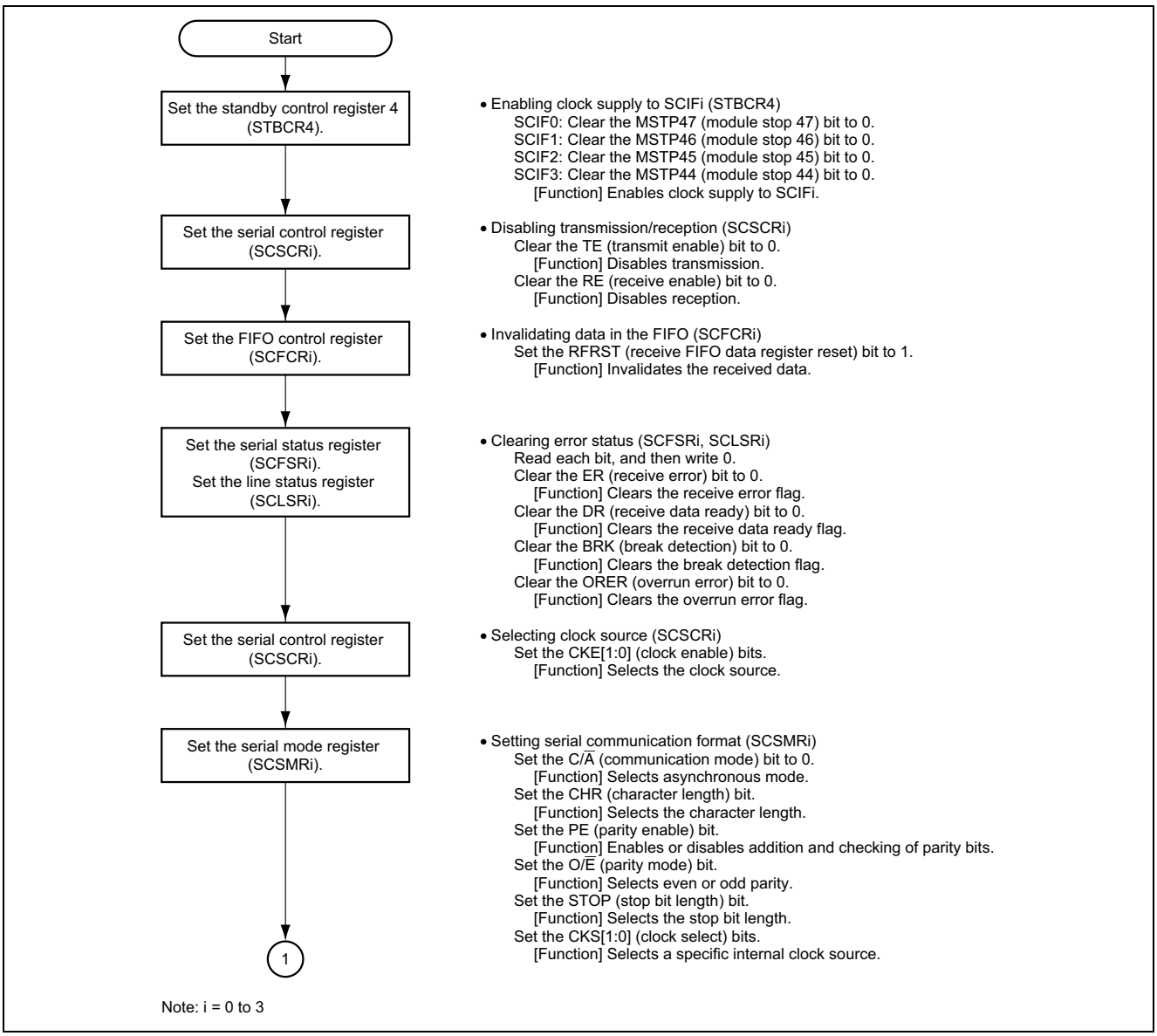
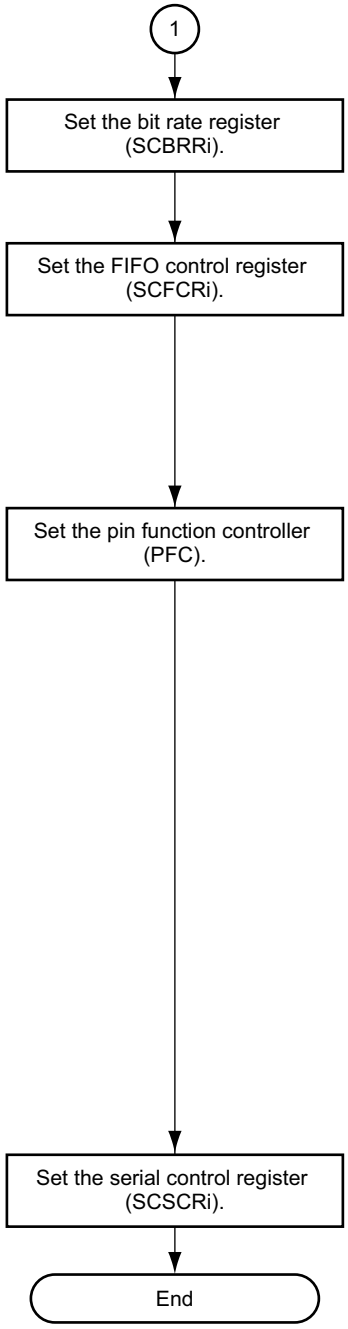


Figure 2 Example Flow of Initial Settings for UART Mode Reception (1)



• Setting transfer rate (SCBRRi)

• Setting the number of stages in the FIFO to be used, enabling/disabling modem control signals, and canceling FIFO reset state (SCFCRi)
Set the RTRG[1:0] (receive FIFO data trigger) bits.
[Function] Selects the receive trigger number.
Set the MCE (modem control enable) bit.
[Function] Enables or disables modem control signals ($\overline{CTS}$ and $\overline{RTS}$).
Clear the RFRST (receive FIFO data register reset) bit to 0.
[Function] Disables resetting of the receive FIFO
Clear the LOOP (loop-back test) bit to 0.
[Function] Disables loop-back test.

• Setting the multiplexed pins as serial communication pins (PACRL1, PECRL2, PECRL3)
SCIF0: Set the PA0MD (PA0 mode) bits to B'001.
[Function] Selects RxD0 pin function.
Set the PA2MD (PA2 mode) bits to B'001.
[Function] Selects SCK0 pin function.
SCIF1: Set the PA3MD (PA3 mode) bits to B'001.
[Function] Selects RxD1 pin function.
Set the PA5MD (PA5 mode) bits to B'001.
[Function] Selects SCK1 pin function.
SCIF2: Set the PE7MD (PE7 mode) bits to B'010.
[Function] Selects RxD2 pin function.
Set the PE8MD (PE8 mode) bits to B'10.
[Function] Selects SCK2 pin function.
SCIF3: Set the PE4MD (PE4 mode) bits to B'010 or the PE11MD (PE11 mode) bits to B'011.
[Function] Selects RxD3 pin function.
Set the PE9MD (PE9 mode) bits to B'100.
[Function] Selects RTS3 pin function.
Set the PE6MD (PE6 mode) bits to B'010 or the PE9MD (PE9 mode) bits to B'011.
[Function] Selects SCK3 pin function.

• Enabling reception (SCSCRi)
Set the RE (receive enable) bit to 1.
[Function] Enables reception.

Note: i = 0 to 3

Figure 3 Example Flow of Initial Settings for UART Mode Reception (2)

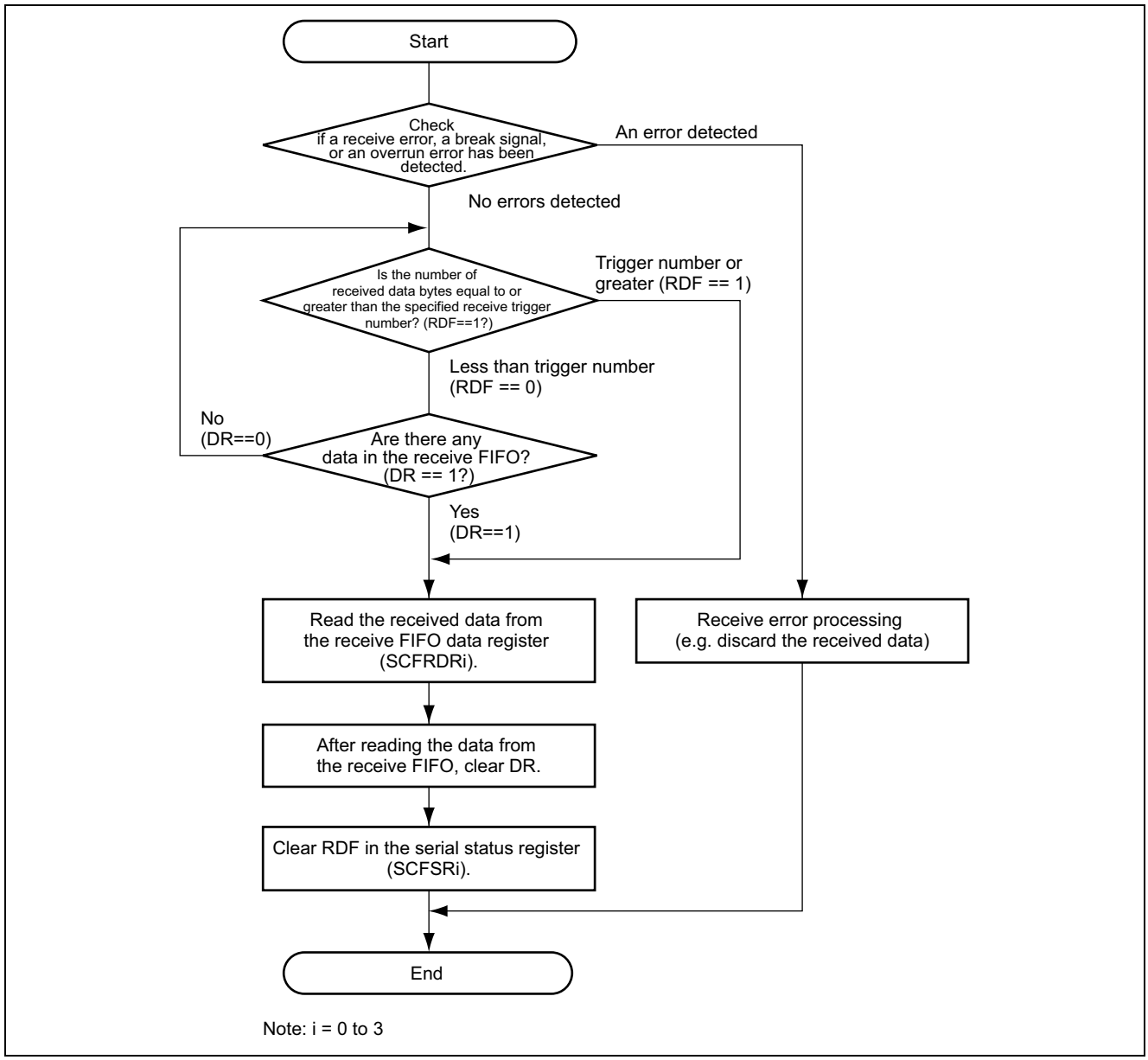


Figure 4 Example Flow of UART Mode Reception Processing

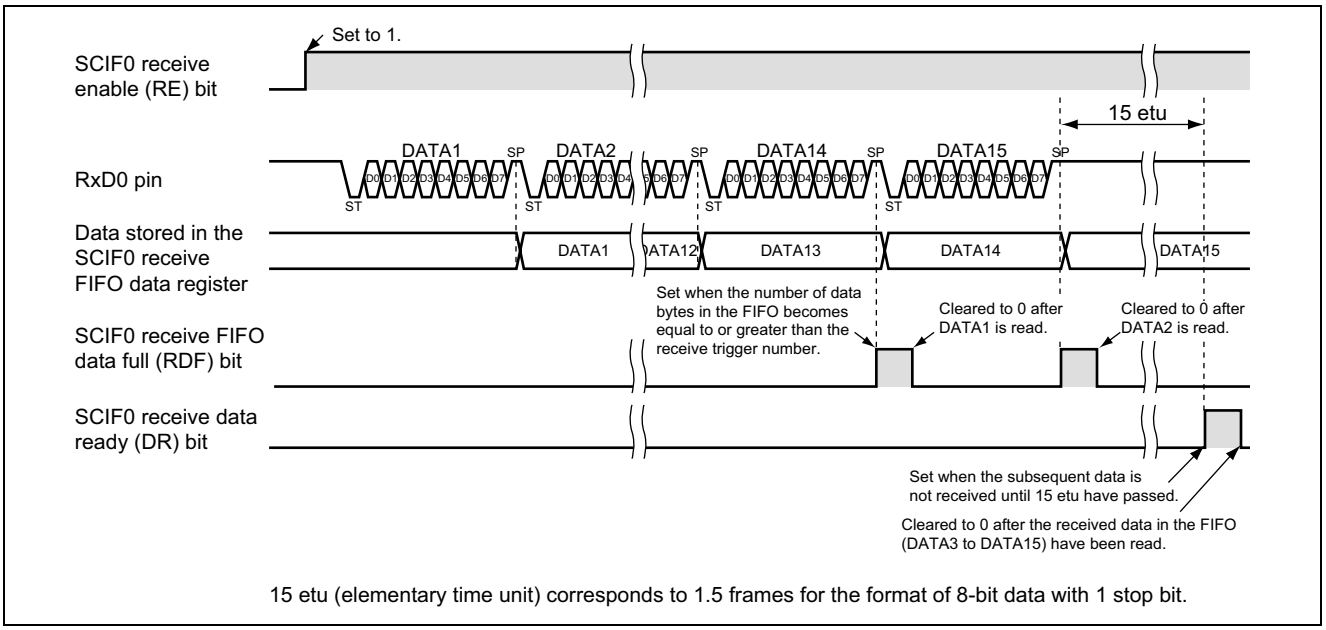
2.3 Operation of Sample Program

The sample program uses the SCIF channel 0 in UART mode and performs reception processing. When the receive FIFO data full flag (RDF bit) in the serial status register (SCFSR_0) is set, the received data is read from the receive FIFO data register. After data is read from the receive FIFO data register, the RDF flag is cleared. If the receive FIFO data full flag is not set, the receive data ready bit (DR bit) is checked; and if data is in the receive FIFO, the received data is read and the DR bit is cleared.

Table 2 shows the communication function settings of the sample program, and figure 5 shows the timing of the sample program operation.

Table 2 Communication Function Settings of Sample Program

Communication Format	Function Setting
Communication mode	UART (asynchronous)
Channel used	Channel 0
Interrupts	Not used
Transfer rate	115.2 kbps
Data size	8 bits
Parity	None
Stop bit	1 bit
Modem control	RTS/CTS disabled
Bit transfer order	LSB first
Receive FIFO data trigger	14



15 etu (elementary time unit) corresponds to 1.5 frames for the format of 8-bit data with 1 stop bit.

Figure 5 Timing of Sample Program Operation

2.4 Register Settings and Processing Sequence of Sample Program

The sample program initializes the SCIF channel 0 in UART mode, and then checks the receive FIFO data full flag (RDF bit) in the serial status register (SCFSR_0). If the bit indicates that the receive FIFO is full (RDF = 1), received data reading processing is performed.

Table 3 shows the register settings related to the SCIF channel 0 made by the sample program. Figure 6 shows the processing flow of the sample program.

Table 3 Register Settings of Sample Program

Register Name	Address	Setting	Description
Standby control register 4 (STBCR4)	H'FFFE040C	H'74	MSTP47 = 0: SCIF0 runs (clock is supplied).
Port A control register L1 (PACRL1)	H'FFFE3816	H'0001	PA0MD[2:0] = B'001: RxD0 input mode
Serial mode register_0 (SCSMR_0)	H'FFFE8000	H'0000	• C/$\overline{A}$ = 0: UART mode • CHR = 0: 8-bit data • PE = 0: Parity bit addition and checking disabled • STOP = 0: 1 stop bit • CKS[1:0] = B'00: Pϕ clock
Serial control register_0 (SCSCR_0)	H'FFFE8008	H'0000	• TE = 0: Transmission disabled • RE = 0: Reception disabled • CKE[1:0] = B'00: Internal clock (SCK is an input pin.)
		H'0010	RE = 1: Reception enabled
FIFO control register_0 (SCFCR_0)	H'FFFE8018	H'0002	RFRST = 1: Reset operation for the receive FIFO data register is enabled.
		H'00C0	• RFRST = 0: Reset operation for the receive FIFO data register is disabled. • RTRG[1:0] = B'11: Receive FIFO data trigger number is 14. *[1]
Bit rate register_0 (SCBRR_0)	H'FFFE8004	H'08	115.2 kbps
Serial status register_0 (SCFSR_0)	H'FFFE8010	H'FF6E*[2]	• ER = 0: Receive error flag cleared • BRK = 0: Break detection flag cleared • DR = 0: Receive data ready flag cleared To clear these bits, read the bits while they are set and write 0 to them.
Line status register_0 (SCLSR_0)	H'FFFE8024	H'0000	ORER = 0: Overrun error flag cleared To clear this bit, read the bit while it is set and write 0 to it.

Notes: 1. Receive FIFO data trigger number is the number of data bytes in the receive FIFO which sets the RDF flag in the serial status register (SCFSR).
2. The value of SCFSR_0 is ANDed with H'FF6E to clear the ER, BRK, and DR bits.

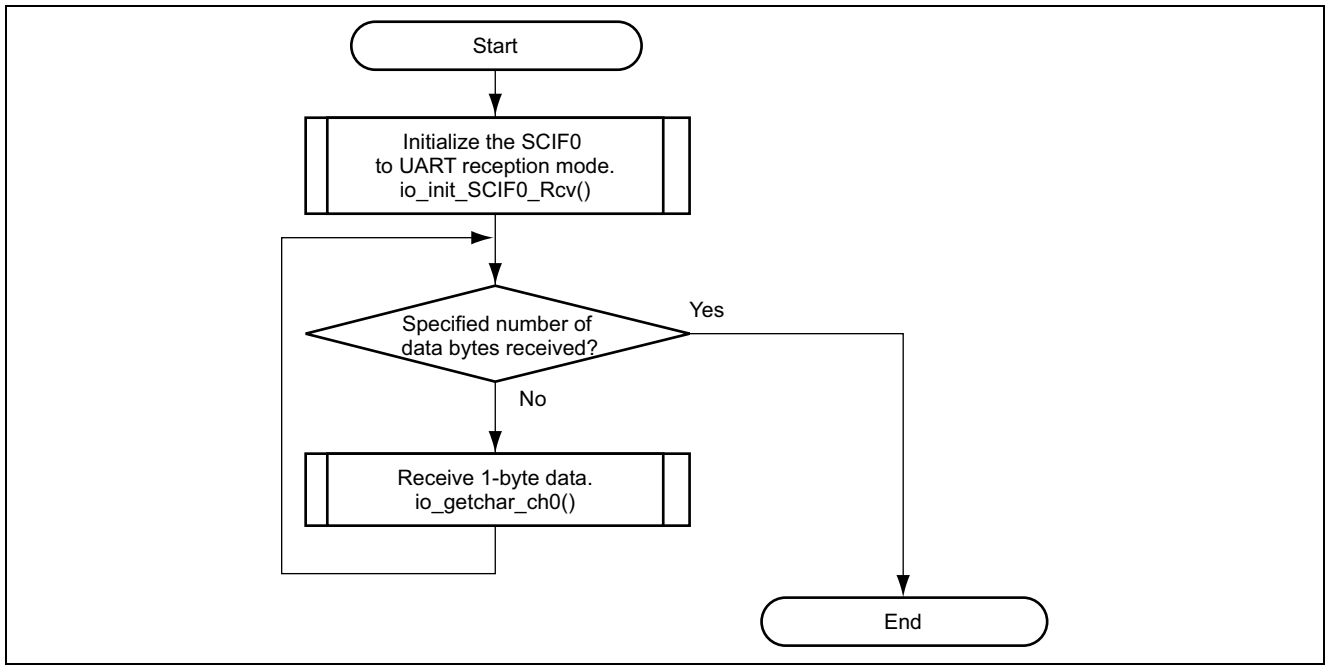


Figure 6 Processing Flow of Sample Program

3. Sample Program Listing

- Sample Program Listing: main.c (1)

```
/*""FILE COMMENT""*******************************************************************************
*
*   System Name: SH7206 Sample Program
*   File Name  : main.c
*   Contents   : Sample program for asynchronous (UART) serial reception by the serial
*                communication interface with FIFO (SCIF)
*   Version    : 1.00.00
*   Model      : M3A-HS60
*   CPU        : SH7206
*   Compiler   : SHC9.0.00
*
*   note       : Sample program for asynchronous (UART) reception by the SCIF0
*
*                <Caution>
*                This sample program is for reference
*                and its operation is not guaranteed.
*                Customers should use this sample program for technical reference
*                in software development.
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   and Renesas Solutions Corp. All Rights Reserved
*
*   history    : 2004.11.04 ver.1.00.00
*""FILE COMMENT END""*************************************************************************/
#include "iodefine.h"                   /* iodefine.h is automatically created by HEW      */

/* ==== Prototype declaration ==== */
void main(void);
void io_init_SCIF0_Rcv(int);
unsigned char io_getchar_ch0(void);

/* ==== Type declaration ==== */
/* SCIF baud rate setting */
typedef struct  {
    unsigned char scbrr;
    unsigned short scsmr;
} SH7206_BAUD_SET;

/* ---- Values for baud rate specification ---- */
enum{
    CBR_1200,
    CBR_2400,
    CBR_4800,
    CBR_9600,
    CBR_19200,
    CBR_31250,
    CBR_38400,
    CBR_57600,
    CBR_115200
};

/* ==== Register value table ==== */
SH7206_BAUD_SET scif_baurs[] = {
    {214, 1},        /*  1200bps (-0.07%) */
    {106, 1},        /*  2400bps ( 0.39%) */
    {214, 0},        /*  4800bps (-0.07%) */
    {106, 0},        /*  9600bps ( 0.39%) */
    { 53, 0},        /* 19200bps (-0.54%) */
    { 32, 0},        /* 31250bps ( 0.00%) */
    { 26, 0},        /* 38400bps (-0.54%) */
    { 17, 0},        /* 57600bps (-0.54%) */
    {  8, 0}         /*115200bps (-0.54%) */
};
```

- Sample Program Listing: main.c (2)

```
 64 /* ==== Receive data buffer ==== */
 65 unsigned char rcv_data[16];
 66
 67  /*""FUNC COMMENT""*****************************************************************
 68  * ID            :
 69  * Module summary: Main function of the sample program
 70  *                 (Asynchronous serial I/O reception processing)
 71  *-----------------------------------------------------------------------------------
 72  * Include       : #include "iodefine.h"
 73  *-----------------------------------------------------------------------------------
 74  * Declaration   : void main(void)
 75  *-----------------------------------------------------------------------------------
 76  * Functional description:
 77  *               : Sets the SCIF0 to UART reception mode, and stores
 78  *               : 16 bytes of received data in the buffer.
 79  *-----------------------------------------------------------------------------------
 80  * Argument      : None
 81  *-----------------------------------------------------------------------------------
 82  * Return value  : None
 83 *-----------------------------------------------------------------------------------
 84  * Notes         :
 85  *""FUNC COMMENT END""*************************************************************/
 86 void main(void)
 87 {
 88     int i;
 89
 90     /* ==== Initialize SCIF0 in UART reception mode ==== */
 91     io_init_SCIF0_Rcv(CBR_115200);   /* Bit rate: 115.2 kbps                    */
 92
 93     /* ==== Receive data ==== */
 94     for(i=0; i < sizeof(rcv_data); i++){
 95         /* ---- Receive 1-byte data ---- */
 96          rcv_data[i] = io_getchar_ch0();
 97     }
 98
 99     while(1){
100         /* Program end */
101     }
102 }
```

- Sample Program Listing: main.c (3)

```
/*""FUNC COMMENT""*************************************************************************
* ID             :
* Module Summary: SCIF0 initialization
*-------------------------------------------------------------------------------------------
* Include        : #include "iodefine.h"
*-------------------------------------------------------------------------------------------
* Declaration    : void io_init_SCIF0_Rcv(int bps)
*-------------------------------------------------------------------------------------------
* Functional description:
*                : Initializes SCIF0 as a reception module in asynchronous mode (UART).
*                : Asynchronous (UART)/ 8 bits / No parity/ 1 stop bit/ RTS/CTS disabled
*                : Baud rate is specified as argument bps.
*-------------------------------------------------------------------------------------------
* Argument       : int bps : Value for baud rate specification
*-------------------------------------------------------------------------------------------
* Return value   : None
*-------------------------------------------------------------------------------------------
* Notes          : The baud rate setting values given in this program are those when the
*                : peripheral clock (Pf) frequency is 33 MHz.
*                : If a different clock is used, the baud rate setting values must be changed.
*""FUNC COMMENT END""*********************************************************************/
void io_init_SCIF0_Rcv(int bps)
{
    /* ==== Canceling power-down mode ==== */
    /* ---- Set standby control register 4 (STBCR4) ---- */
    CPG.STBCR4.BIT.MSTP47 = 0;      /* Start clock supply to SCIF0                          */

    /* ==== SCIF0 initialization ==== */
    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF0.SCSCR.WORD = 0x0000;      /* Disable transmission/reception by SCIF0              */

    /* ---- Set FIFO control register (SCFCRi) ---- */
    SCIF0.SCFCR.BIT.RFRST = 1;      /* Reset receive FIFO data register                     */

    /* ---- Set serial status register (SCFSRi) ---- */
    SCIF0.SCFSR.WORD &= 0xff6eu;    /* Clear the ER, BRK, and DR bits                       */

    /* ---- Set line status register (SCLSRi) ---- */
    SCIF0.SCLSR.BIT.ORER = 0;       /* Clear the ORER bit                                   */

    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF0.SCSCR.BIT.CKE = 0x0;      /* B'00 : Internal clock                                */

    /* ---- Set serial mode register (SCSMRi) ---- */
    SCIF0.SCSMR.WORD = scif_baurs[bps].scsmr;
                                    /* Communication mode 0: Asynchronous mode              */
                                    /* Character length   0: 8-bit data                     */
                                    /* Parity enable      0: Disable addition and checking */
                                    /* Parity mode        0: Even parity                    */
                                    /* Stop bit length    0: 1 bit                          */
                                    /* Clock select       0: Table value                    */

    /* ---- Set bit rate register (SCBRRi) ---- */
    SCIF0.SCBRR.BYTE = scif_baurs[bps].scbrr;

    /* ---- Set FIFO control register (SCFCRi) ---- */
    SCIF0.SCFCR.WORD = 0x00c0;     /* RTS output active trigger       : Initial value      */
                                   /* Receive FIFO data trigger       : 14 bytes           */
                                   /* Modem control enable            : Disabled           */
                                   /* Receive FIFO data register reset: Disabled           */
                                   /* Loop-back test: Disabled */

    /* ---- Pin function controller (PFC) setting ---- */
    PORT.PACRL1.BIT.PA0MD = 1;     /* Set the PA0 pin as RxD0 (PACRL1)                     */
```

- Sample Program Listing: main.c (4)

```
182     /* ---- Set serial control register (SCSCRi) ---- */
183     SCIF0.SCSCR.BIT.RE = 1;             /* Enable SCIF0 reception                          */
184 }
185
186 /*""FUNC COMMENT""*****************************************************************************
187 * ID            :
188 * Module summary: SCIF0 1-byte (one character) reception processing
189 *-----------------------------------------------------------------------------------------------
190 * Include       : #include "iodefine.h"
191 *-----------------------------------------------------------------------------------------------
192 * Declaration   : unsigned char io_getchar_ch0(void)
193 *-----------------------------------------------------------------------------------------------
194 * Functional description:
195 *               : Confirms that the receive FIFO data full flag (RDF) in the SCIF0 serial
196 *               : status register (SCFSR0) is set, and then reads the received data
197 *               : from the receive FIFO data register.
198 *               : Returns 0 when a framing error, parity error, overrun error, or the
199 *               : break signal is detected.
200 *-----------------------------------------------------------------------------------------------
201 * Argument      : None
202 *-----------------------------------------------------------------------------------------------
203 * Return value  : Received data
204 *-----------------------------------------------------------------------------------------------
205 * Note          : None
206 *""FUNC COMMENT END""*************************************************************************/
207 unsigned char io_getchar_ch0(void)
208 {
209     unsigned char data;
210
211     /* ==== Checking for receive error, break signal, and overrun error ==== */
212     if((SCIF0.SCFSR.WORD & 0x0090u ) || (SCIF0.SCLSR.BIT.ORER == 1)) {
213         /* Receive error processing (discard received data) */
214         SCIF0.SCSCR.BIT.RE = 0;           /* Disable reception                             */
215         SCIF0.SCFCR.BIT.RFRST = 1;        /* Enable receive FIFO reset                     */
216         SCIF0.SCFCR.BIT.RFRST = 0;        /* Disable receive FIFO reset                    */
217         SCIF0.SCFSR.WORD &= ~0x0091u;     /* Clear the ER, BRK, and DR bits                */
218
219         SCIF0.SCLSR.BIT.ORER = 0;         /* Clear the ORER bit                            */
220         SCIF0.SCFSR.BIT.RDF = 0;          /* Clear the RDF bit                             */
221
222         SCIF0.SCSCR.BIT.RE = 1;           /* Enable reception                              */
223         return 0;
224     }
225
226     /* ==== Is the number of received data bytes equal to or
227                           greater than the specified       trigger number? (RDF==1?) ==== */
228     while(SCIF0.SCFSR.BIT.RDF == 0){
229         /* Is data in the receive FIFO? (DR==1?) */
230         if(SCIF0.SCFSR.BIT.DR == 1){
231             /* Stop waiting for the RDF flag to be set because data is in the FIFO */
232             break;
233         }
234     }
235
236     /* ==== Read received data from receive FIFO data register (SCFRDR0) ==== */
237     data = SCIF0.SCFRDR.BYTE;
238
239     /* ==== In succession to reading from receive FIFO, clear DR ==== */
240     SCIF0.SCFSR.BIT.DR = 0;    /* DR bit is cleared if the FIFO is empty */
241
242     /* ==== Clear RDF in serial status register (SCFSR0) ==== */
243     SCIF0.SCFSR.BIT.RDF = 0;    /* Clear the RDF bit */
244
245     return data;
246 }
247 /* End of File */
```

4. Reference Documents

- SH-2A SH2A-FPU Software Manual (Rev.3.00)
  (Download the latest edition from the website of Renesas Technology Corp.)
- SH7206 Group Hardware Manual (Rev. 1.00)
  (Download the latest edition from the website of Renesas Technology Corp.)

5. Website

- Website of Renesas Technology Corp.
  http://www.renesas.com/

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep.14.05	—	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.