SH7262/SH7264 Group

REJ06B0978-0100
Rev.1.00
Apr. 28, 2010

Serial Communication Interface with FIFO, Configuring the Serial Communication in Clock Synchronous Mode (Full-duplex)

Summary

This application note describes the serial communication in clock synchronous mode provided by the SH7264 Serial Communication Interface with FIFO (SCIF).

Target Device

SH7262/7264 MCU (In this document, SH7262/SH7264 are described as SH7264.

Contents

1. Introduction

1.1 Specifications

- Initializes the SH7264 Serial Communication Interface with FIFO (SCIF) channel 1 as the transmitter and received in clock synchronous mode
- Executes full-duplex communication on SCIF channel 1 by transmit and receive interrupts. The transmit interrupt processing is activated by the transmit-FIFO-data-empty interrupt, and the receive interrupt processing is activated by the receive-FIFO-data-full interrupt.

1.2 Modules Used

- Serial Communication Interface with FIFO (SCIF)
- Interrupt controller

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequencies	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Technology Corp. High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Technology SuperH RISC engine Family C/C++ Compiler Package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug-gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 –nologo)

1.4 Related Application Notes

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group Serial Communication Interface with FIFO, Configuration to Transmit Strings in Asynchronous Mode
- SH7262/SH7264 Group Serial Communication Interface with FIFO, Configuration to Receive Strings in Asynchronous Mode

1.5 About Active-low Pins (Signals)

The symbol "#" suffixed to the pin (or signal) names indicates that the pins (or signals) are active-low.

2. Applications

This application uses the Serial Communication Interface with FIFO (SCIF).

2.1 SCIF Overview

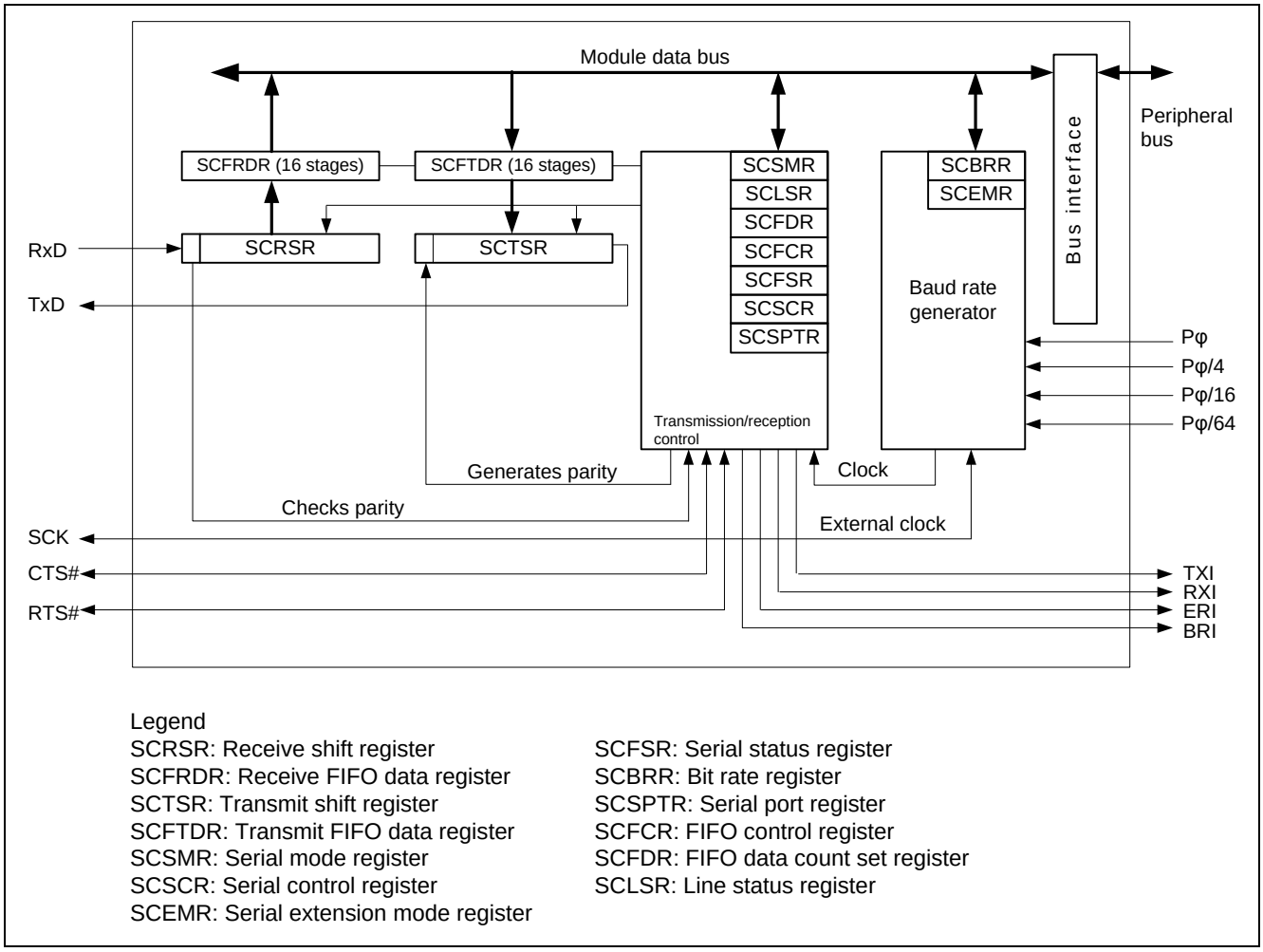
In clock synchronous mode, the SCIF transmits and receives data sync with the clock pulse. Either an internal clock or external clock from the SCK pin can be specified as the clock source. When specifying the internal clock, the SCIF outputs synchronous clock from the SCK pin. When specifying the external clock, the external device inputs the synchronous clock into the SCK pin. The communication format is specified as 8-bit data fixed.

Table 1 lists the overview of the clock synchronous mode. Figure 1 shows the SCIF block diagram.

Table 1 SCIF (Clock Synchronous Mode) Overview

Item	Description
Number of channels	8 (SCIF0 to SCIF7)
Clock source	Internal clock: Pφ, Pφ/4, Pφ16, Pφ64 Pφ: internal peripheral clock External clock: SCK0 to SCK3 pins input clock
Data format	Transfer data length: 8-bit fixed Order of transfer: LSB first fixed
Baud rate	When specifying the internal clock: 549 bps to 9 Mbps (Pφ is at 36 MHz) When specifying the external clock: Up to 3 Mbps (Pφ is at 36 MHz, external clock is at 9 MHz) Note: Make sure to satisfy the electric characteristics of the SH7264 and the connected external device.
Error detection	Overrun error
Interrupt request	Transmit-FIFO-data-empty interrupt (TXI) by the transmit FIFO data empty (TDFE) Break interrupt (BRI) by the break (BRK) or overrun error (ORER) Receive-FIFO-data-full (RXI) by the Receive FIFO data full (RDF) or data ready (DR) Receive-error interrupt (ERI) by the receive error (ER)
Others	• Outputs the synchronous clock from the SCK pin when specifying the internal clock • Supplying clock to unused channels can be stopped to reduce power consumption • The number of valid data stored in the Transmit and Receive FIFO data registers, and the number of receive errors stored in the Receive FIFO data register can be detected

Note: For more information about the SCIF, refer to the Serial Communication Interface with FIFO chapter in the SH7262 Group, SH7264 Group Hardware Manual.



Legend

SCRSR: Receive shift register
SCFRDR: Receive FIFO data register
SCTSR: Transmit shift register
SCFTDR: Transmit FIFO data register
SCSMR: Serial mode register
SCSCR: Serial control register
SCEMR: Serial extension mode register
SCFSR: Serial status register
SCBRR: Bit rate register
SCSPTR: Serial port register
SCFCR: FIFO control register
SCFDR: FIFO data count set register
SCLSR: Line status register

Figure 1 SCIF Block Diagram

2.2 Configuration Procedure

This section describes how to configure the SH7264 SCIF in clock synchronous mode. Figure 2 and Figure 3 show flow charts of configuring the transmission/reception in clock synchronous mode. Figure 4 shows the flow chart of the transmit interrupt processing in clock synchronous mode. Figure 5 shows the flow chart of the receive interrupt processing in clock synchronous mode. Figure 6 shows the flow chart of the overrun error interrupt processing.

For details on register settings, refer to the Serial Communication Interface with FIFO chapter in the SH7262 Group, SH7264 Group Hardware Manual.

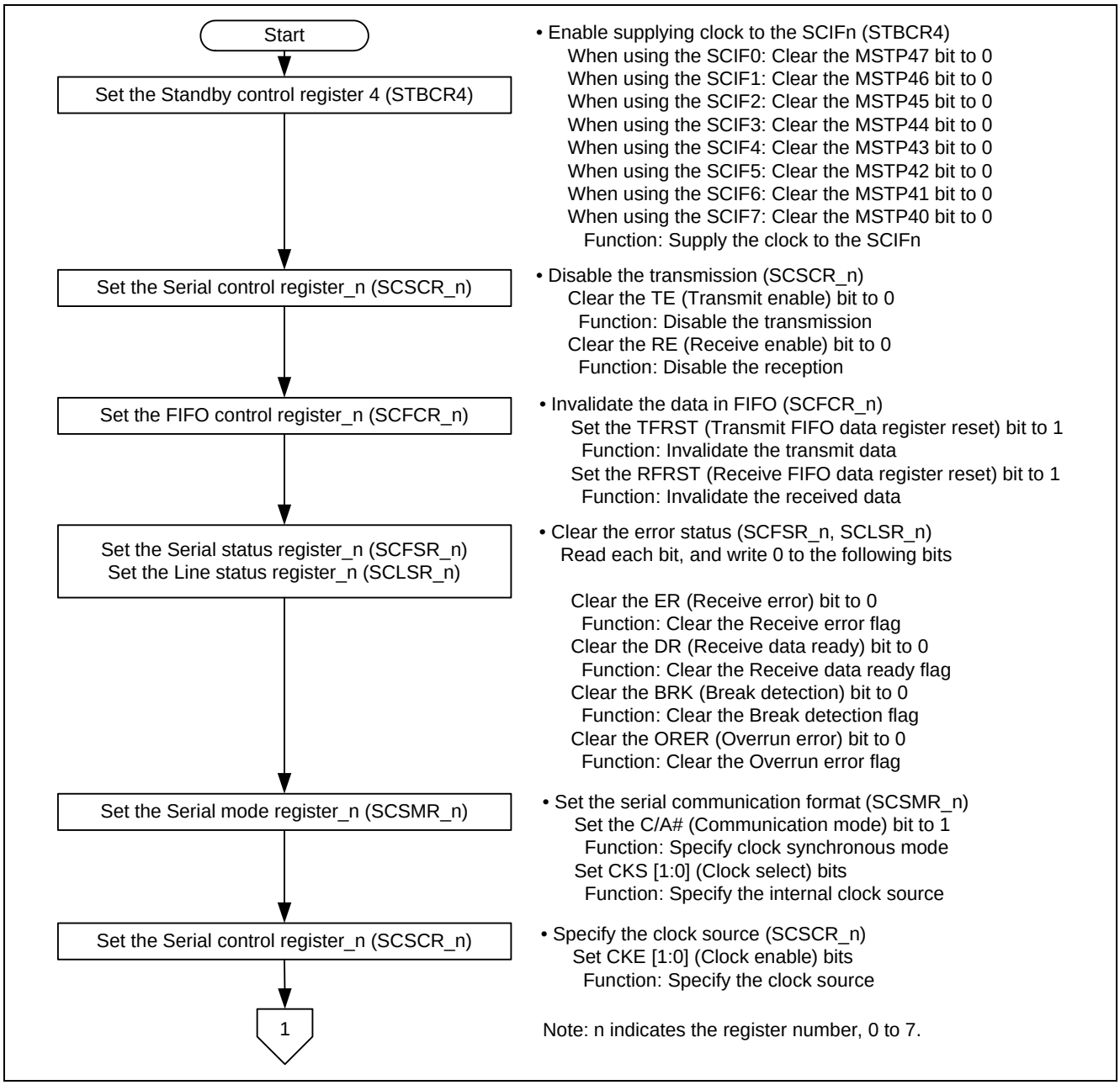


Figure 2 Flow Chart for Configuring the Transmission/Reception in Clock Synchronous Mode (1/2)

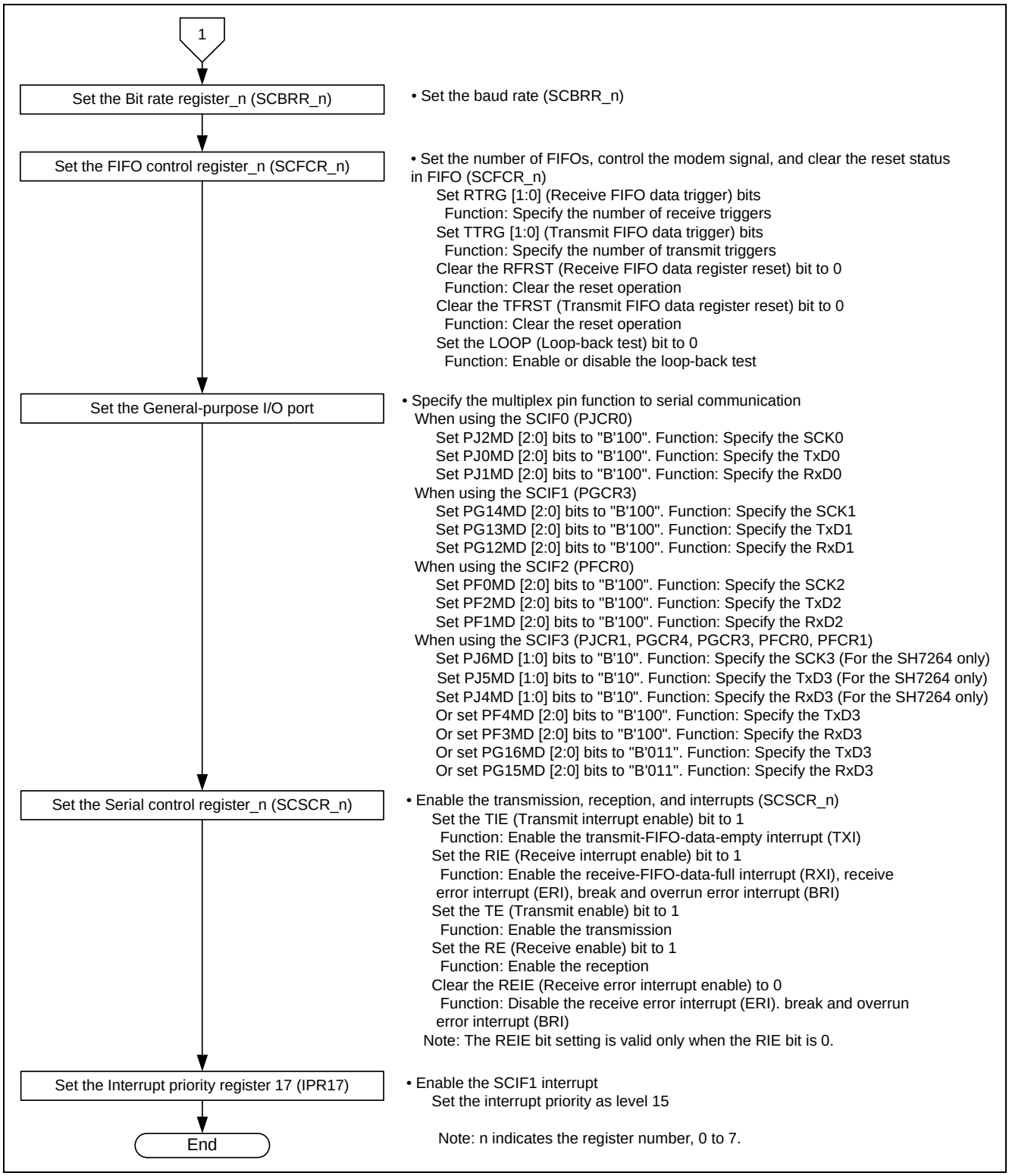


Figure 3 Flow Chart for Configuring the Transmission/Reception in Clock Synchronous Mode (2/2)

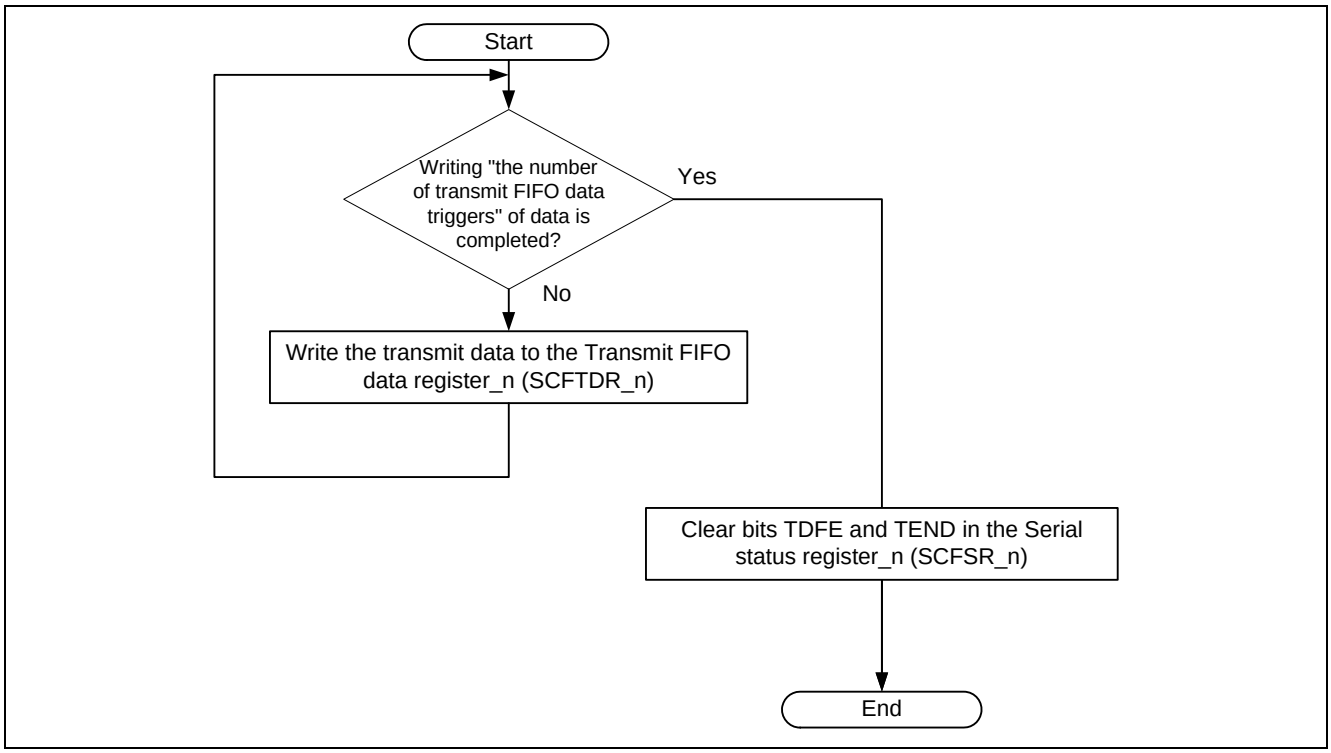


Figure 4 Flow Chart of the Transmit Interrupt Processing (TXI) in Clock Synchronous Mode

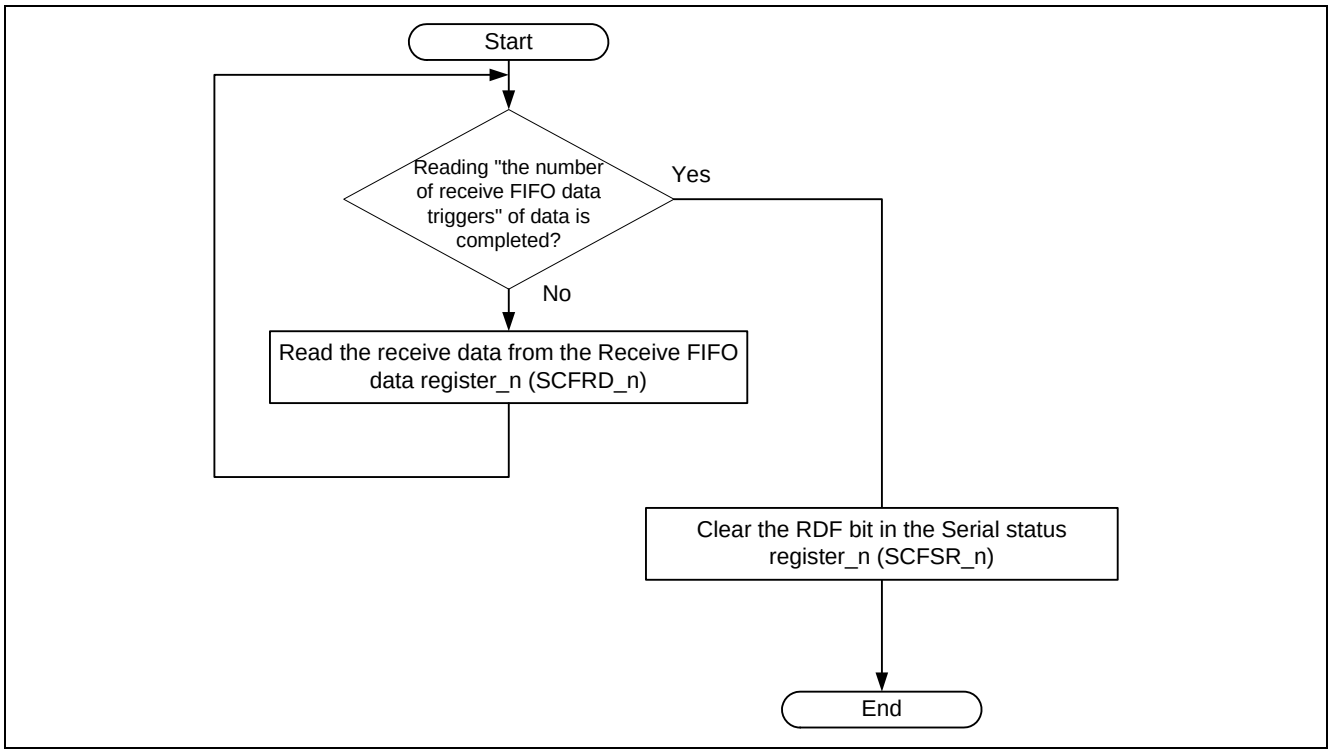


Figure 5 Flow Chart of the Receive Interrupt Processing (RXI) in Clock Synchronous Mode

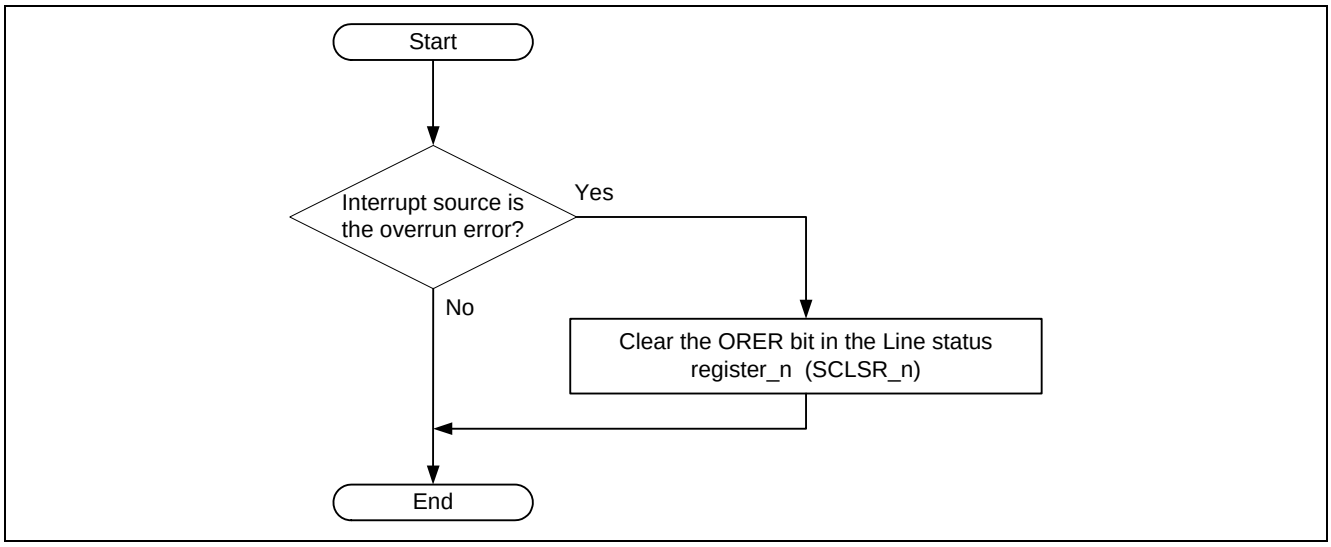


Figure 6 Flow Chart of the Overrun Error Interrupt Processing (BRI)

2.3 Sample Program Operation

This sample program uses the SCIF channel 1 as the transmitter and receiver in clock synchronous mode. It uses the SCIF1 loop-back test function, connects the TxD1 (serial transmit pin) and RxD1 (serial receive pin) internally, and executes the full-duplex serial communication. Set the LOOP (loop-back test) bit in the FIFO control register (SCFCR) to 1 to enable the loop-back test function.

Table 2 lists the transmission settings for the sample program. Figure 7 shows the operation timing of the sample program.

Table 2 Sample Program Transmission (Reception) Settings

Communication Format	Setting
Communication mode	Clock synchronous mode
Number of channel to use	Channel 1
Interrupt	Transmit-FIFO-data-empty Receive-FIFO-data-full
Baud rate	100 kbps
Data length	8-bit
Bit order	LSB first
Synchronous clock	SCIF channel 1 external clock output
Number of FIFO data triggers	Receive: 8, transmit: 8
Loop-back test function	Enabled (TxD1 pin and RxD1 pin are connected internally)

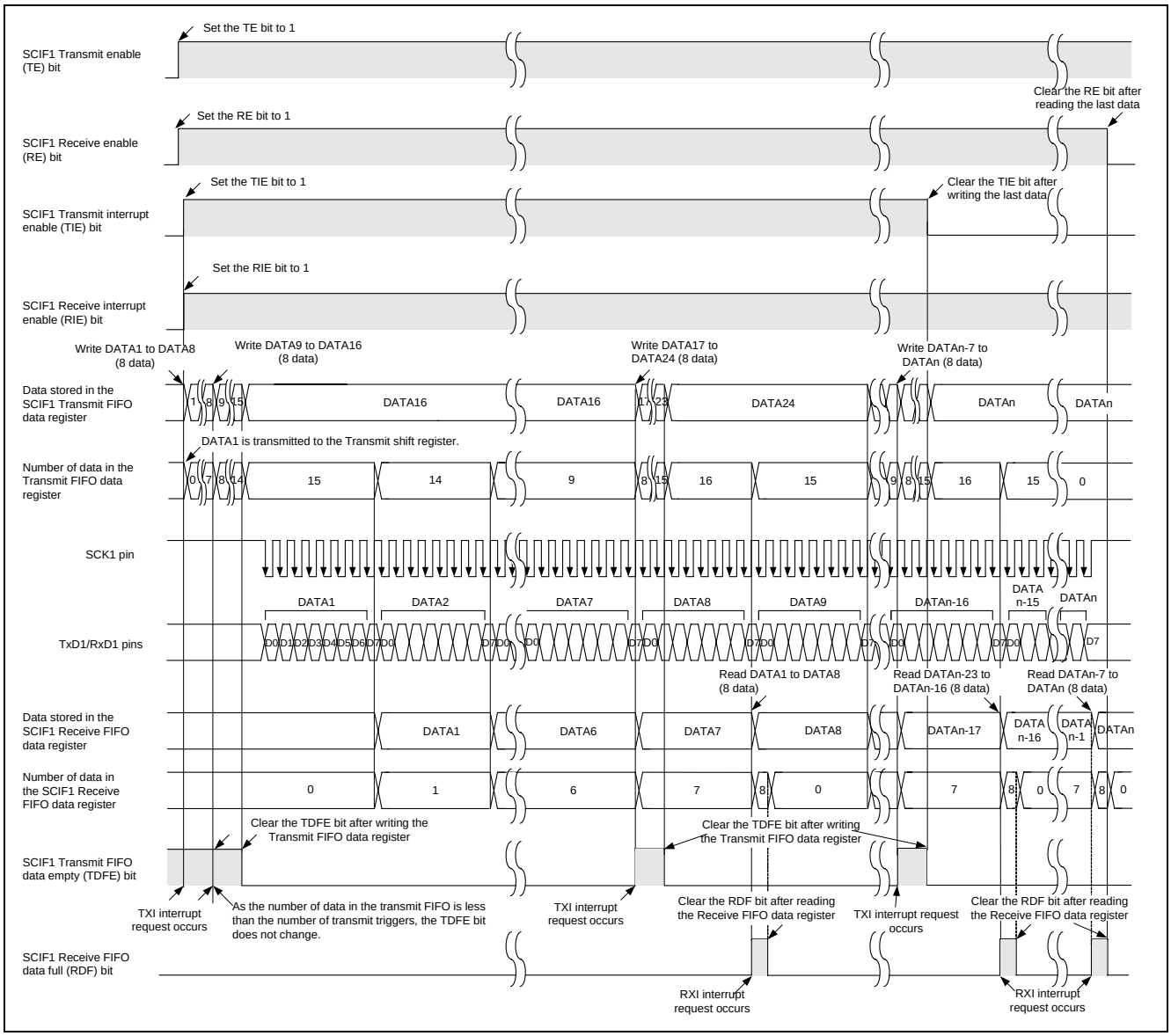


Figure 7 Sample Program Operation Timing

2.4 Sample Program Procedure

The sample program uses the SCIF channel 1 as the transmitter and receiver in clock synchronous mode to execute the full-duplex communication by the transmit and receive interrupts. The transmit interrupt processing specifies the number of transmit FIFO data triggers as 8 to write 8-byte transmit data in the Transmit FIFO data register (SCFTDR_1) by an interrupt processing. The receive interrupt processing specifies the number of receive FIFO data triggers as 8 to read 8-byte received data from the Receive FIFO data register (SCFRDR_1) by an interrupt processing.

Table 3 lists the register settings in the sample program. Figure 8 to Figure 10 show flow charts of the sample program.

Table 3 Register Settings

Register Name	Address	Setting	Description
Standby control register 4 (STBCR4)	H'FFFE 040C	H'BF	MSTP46 = "0": SCIF1 is operating (Supplies the clock)
Port G control register 3 (PGCR3)	H'FFFE 38C8	H'0444	• PG14MD [2:0] = "B'100": SCK1 output (SCIF1) • PG13MD [2:0] = "B'100": TxD1 output (SCIF1) • PG12MD [2:0] = "B'100": RxD1 output (SCIF1)
Serial control register_1 (SCSCR_1)	H'FFFE 8808	H'0000	• TE = "0": Disable the transmission • RE = "0": Disable the reception • CKE [1:0] "B'00": Internal clock/SCK pin outputs the synchronous clock
		H'00F0	• TIE = "1": Enable the transmit interrupt • RIE = "1": Enable the receive interrupt, receive error interrupt, break and overrun error interrupt • TE = "1": Enable the transmission • RE = "1": Enable the reception • REIE = "0": Disable the interrupt error interrupt, break and overrun error interrupt
FIFO control register_1 (SCFCR_1)	H'FFFE 8818	H'0006	• TFRST = "1": Enable to reset the transmit FIFO • RFRST = "1": Enable to reset the receive FIFO
		H'0081	• RTRG [1:0] = B'10: Number of data in the receive FIFO is 8 • TTRG [1:0] = B'00: Number of data in the transmit FIFO is 8 • TFRST = "0": Disable to reset the transmit FIFO • RFRST = "0": Disable to reset the receive FIFO • LOOP = "1": Enable the loop-back test
Serial status register_1 (SCFSR_1)	H'FFFE 8810	H'FF6E	• ER = "0": Receive error • BRK = "0": Break detected • DR = "0": Receive data ready Read the bit before clearing to 0.
Line status register_1 (SCLSR_1)	H'FFFE 8824	H'0000	• ORER = "0": Overrun error Read the bit before clearing to 0.
Serial mode register_1 (SCSMR_1)	H'FFFE 8800	H'0080	• C/A# = "1": Clock synchronous mode • CKS [1:0] = "0": Peripheral clock
Bit rate register_1 (SCBRR_1)	H'FFFE 8804	H'59	• Set the baud rate as 100 kbps

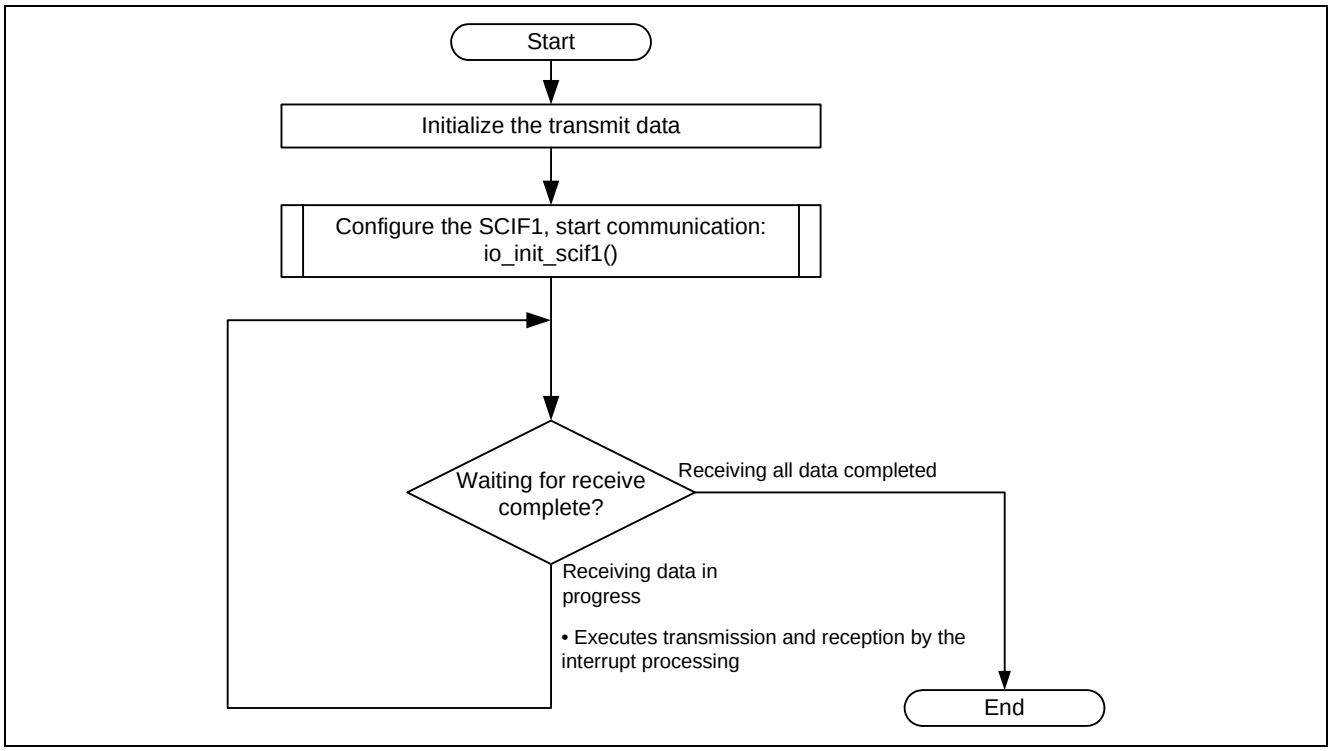


Figure 8 Sample Program Main Flow Chart

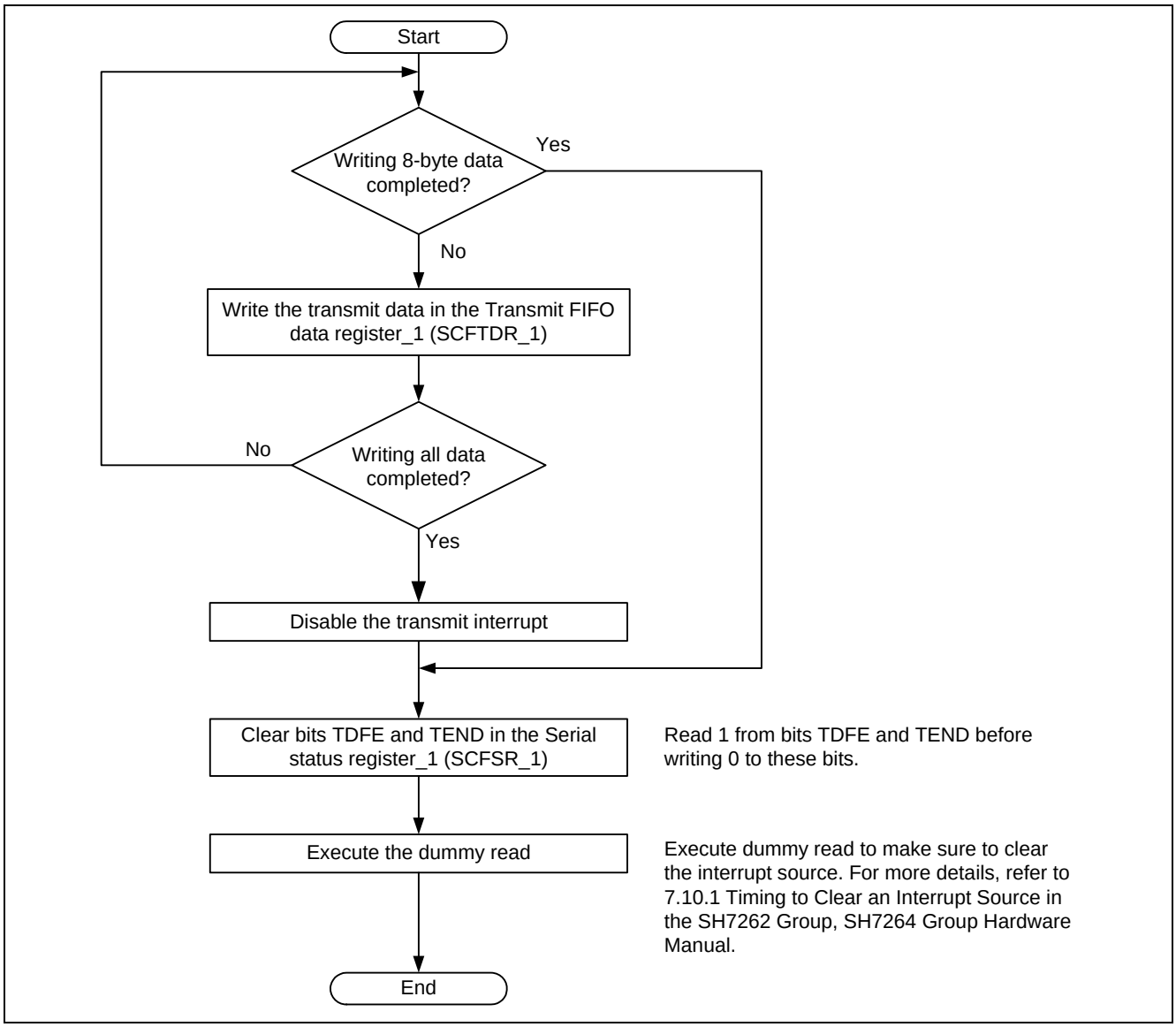


Figure 9 Sample Program Transmit Processing Flow Chart

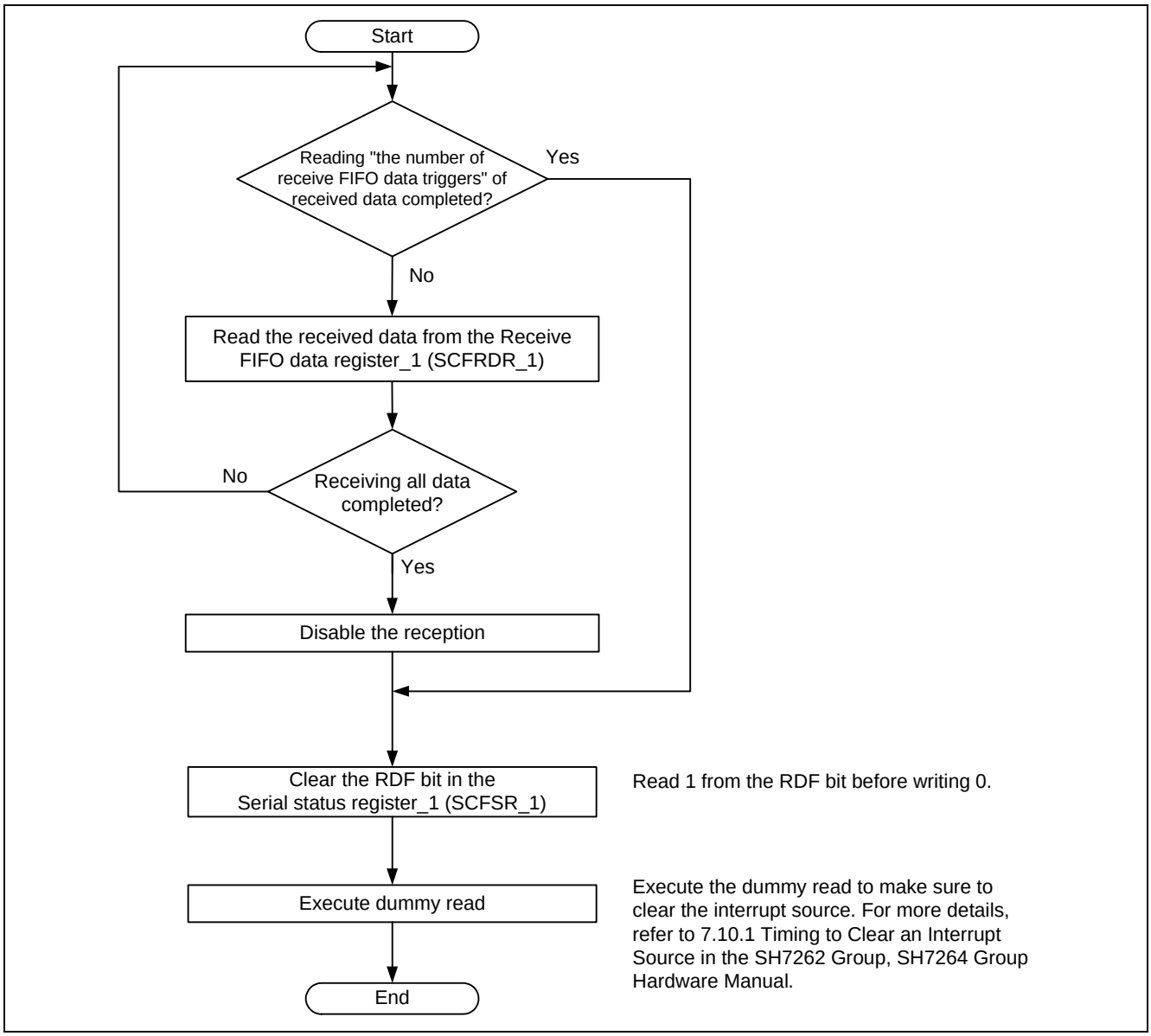


Figure 10 Sample Program Receive Processing Flow Chart

3. Sample Program Listing

3.1 Supplement to the Sample Program

As the capacity of the SH7264 large-capacity internal RAM varies as 1 MB or 640 KB, depending on the MCU type, the section alignment and register setting must be partly altered. To support both MCU types, this application note provides two types of sample programs (workspaces) for 1-MB RAM and 640-KB RAM.

As the MCU with 640-KB RAM must be write-enabled before writing data in the data-retention RAM, the System control register 5 (SYSCR5) is set to write-enable the RAM in the sample program for 640-KB RAM.

Review your product and use the appropriate workspace.

3.2 Sample Program Listing "main.c" (1/8)

```
1       /***********************************************************************
2       *   DISCLAIMER
3       *
4       *   This software is supplied by Renesas Electronics Corp. and is only
5       *   intended for use with Renesas products. No other uses are authorized.
6       *
7       *   This software is owned by Renesas Electronics Corp. and is protected under
8       *   all applicable laws, including copyright laws.
9       *
10      *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11      *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12      *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13      *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14      *   DISCLAIMED.
15      *
16      *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17      *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18      *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19      *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20      *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21      *
22      *   Renesas reserves the right, without notice, to make changes to this
23      *   software and to discontinue the availability of this software.
24      *   By using this software, you agree to the additional terms and
25      *   conditions found by accessing the following link:
26      *   http://www.renesas.com/disclaimer
27      ***********************************************************************
28      * (C) 2010 Renesas Electronics Corporation. All rights reserved.
29      *""FILE COMMENT""************ Technical reference data *************************
30      *   System Name : SH7264 Sample Program
31      *   File Name   : main.c
32      *   Abstract    : Serial Communication Interface with FIFO (SCIF), transmission and
33      *               : reception in clock synchronous mode sample program
34      *   Version     : 1.00.00
35      *   Device      : SH7262/SH7264
36      *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37      *               : C/C++ compiler package for the SuperH RISC engine family
38      *               :                             (Ver.9.03 Release00).
39      *   OS          : None
40      *   H/W Platform: M3A-HS64G50(CPU board)
41      *   Description :
42      ***********************************************************************
43      *   History     : Jan.29,2010 ver.1.00.00
44      *""FILE COMMENT END""*****************************************************/
45      #include <machine.h>
46      #include "iodefine.h"       /* SH7264 iodefine  */
47
```

3.3 Sample Program Listing "main.c" (2/8)

```
48    /* ==== Prototype declaration ==== */
49    void main(void);
50    void io_init_scif1(int);
51    void io_interrupt_scif1_bri(void);
52    void io_interrupt_scif1_rxi(void);
53    void io_interrupt_scif1_txi(void);
54
55    /* ==== Type definition ==== */
56    /* SCIF baud rate setting */
57    typedef struct  {
58      unsigned char scbrr;
59      unsigned short scsmr;
60    } SH7264_BAUD_SET;
61
62    /* ---- Baud rate specified value ---- */
63    enum{
64      CBR_500,
65      CBR_1K,
66      CBR_2_5K,
67      CBR_5K,
68      CBR_10K,
69      CBR_25K,
70      CBR_50K,
71      CBR_100K,
72      CBR_250K,
73      CBR_500K,
74      CBR_1M
75    };
76    /* ---- Number of data to transfer ---- */
77    #define DATA_NUM       0x100u     /* Set the value in multiples of the receive triggers */
78
79    /* ==== Variable definition ==== */
80    /* ---- Register setting table (P clock = 36 MHz) ---- */
81    static SH7264_BAUD_SET scif_baud[] = {
82      {255, 3},    /*       500 Hz  */
83      {140, 3},    /*     1,000 Hz  */
84      {224, 2},    /*     2,500 Hz  */
85      {112, 2},    /*     5,000 Hz  */
86      {224, 1},    /*    10,000 Hz  */
87      { 89, 1},    /*    25,000 Hz  */
88      {179, 0},    /*    50,000 Hz  */
89      { 89, 0},    /*   100,000 Hz  */
90      { 35, 0},    /*   250,000 Hz  */
91      { 17, 0},    /*   500,000 Hz  */
92      {  8, 0}     /* 1,000,000 Hz  */
```

3.4 Sample Program Listing "main.c" (3/8)

```
93    };
94    unsigned int DataNum = DATA_NUM;  /* Number of data to transmit */
95    unsigned int SndCnt;              /* Transmit counter */
96    unsigned int RcvCnt;              /* Receive counter */
97    unsigned char SndData[DATA_NUM];  /* Transmit data storage buffer */
98    unsigned char RcvData[DATA_NUM];  /* Receive data storage buffer */
99
100   /*""FUNC COMMENT""*****************************************************************
101    * ID            :
102    * Outline       : Sample program main (Clock synchronous serial I/O communication).
103    *-------------------------------------------------------------------------------
104    * Include       : "iodefine.h"
105    *-------------------------------------------------------------------------------
106    * Declaration   : void main(void);
107    *-------------------------------------------------------------------------------
108    * Description   : Uses the SCIF1 transmit-FIFO-data-empty interrupt, and receive
109    *               : -FIFO-data-full interrupt to transmit or receive the test data.
110    *-------------------------------------------------------------------------------
111    * Argument      : void
112    *-------------------------------------------------------------------------------
113    * Return Value  : void
114    *-------------------------------------------------------------------------------
115    * Note          :
116    *""FUNC COMMENT END""*************************************************************/
117   void main(void)
118   {
119     int i;
120
121     /* ==== Initializes the transmit data ==== */
122     for( i = 0; i < DATA_NUM; i++ ){
123       SndData[i] = (unsigned char)i;
124     }
125
126     /* ==== Initializes the SCIF1, and starts communication ==== */
127     io_init_scif1(CBR_100K);    /* Specifies the bit rate as 100 kbps */
128
129     /* ==== Waits until the reception is completed ==== */
130     while(RcvCnt < DATA_NUM){
131       /* Transmits or receives data by the interrupt processing */
132     }
133
134     while (1) {
135       /* Program end */
136     }
137   }
138
```

3.5 Sample Program Listing "main.c" (4/8)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : Configure the SCIF1
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_init_scif1(int bps);
 *-----------------------------------------------------------------------------
 * Description   : Configures the SCIF1 in clock synchronous mode.
 *               : Sets it in clock synchronous mode, and external clock output.
 *               : Also, it enables the transmit/receive interrupts.
 *-----------------------------------------------------------------------------
 * Argument      : int bps ; I : Bit rate specified value
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          : The above baud rate specified value is applicable when using
 *               : the peripheral clock (operating frequency for the peripheral
 *               : module using the internal clock) is 36 MHz. Alter the baud rate
 *               : setting when using other clocks.
 *""FUNC COMMENT END""*************************************************************/
void io_init_scif1(int bps)
{
  /* ==== Wakes up the MCU from power-down mode ==== */
  /* ---- Sets the Standby control register 4 (STBCR4) ---- */
  CPG.STBCR4.BIT.MSTP46 = 0;      /* Starts to supplying clock to the SCIF1 */

  /* ==== Configures the SCIF1 ==== */
  /* ---- Sets the Serial control register (SCSCRi) ---- */
  SCIF1.SCSCR.WORD = 0x0000;      /* SCIF1 stops transmission/reception */

  /* ---- Sets the FIFO control register (SCFCRi) ---- */
  SCIF1.SCFCR.WORD = 0x0006;      /* Resets the receive/transmit FIFO */

  /* ---- Sets the Serial status register (SCFSRi) ---- */
  SCIF1.SCFSR.WORD &= 0xff6eu;/* Clears bits ER, BRK, and DR */

  /* ---- Sets the Line status register (SCLSRi) ---- */
  SCIF1.SCLSR.BIT.ORER = 0;       /* Clears the ORER bit */

  /* ---- Sets the Serial mode register (SCSMRi) ---- */
  SCIF1.SCSMR.WORD = scif_baud[bps].scsmr | 0x0080u ;
                                /* Communication mode, 1: Clock synchronous mode */
                                /* Clock select: Setting in table */

```

3.6 Sample Program Listing "main.c" (5/8)

```
184     /* ---- Sets the Serial control register (SCSCRi) ---- */
185     SCIF1.SCSCR.BIT.CKE = 0x0;      /* B'00: Outputs internal clock/synchronous clock */
186
187     /* ---- Sets the Bit rate register (SCBRRi) ---- */
188     SCIF1.SCBRR.BYTE = scif_baud[bps].scbrr;
189
190     /* ---- Sets the FIFO control register (SCFCRi) ---- */
191     SCIF1.SCFCR.WORD = 0x0081;      /* RTS output active trigger: Default value */
192                                 /* Number of receive FIFO data triggers, B'10: 8 */
193                                 /* Number of transmit FIFO data triggers, B'00: 8 */
194                                 /* Modem control enable: Disabled */
195                                 /* Receive FIFO data register reset: Disabled */
196                                 /* Transmit FIFO data register reset: Disabled */
197                                 /* Loop-back test, 1: Enabled */
198
199     /* ==== Sets the General-purpose I/O port ==== */
200     PORT.PGCR3.BIT.PG14MD = 4;      /* Specifies the SCK1 pin */
201     PORT.PGCR3.BIT.PG13MD = 4;      /* Specifies the TxD1 pin */
202     PORT.PGCR3.BIT.PG12MD = 4;      /* Specifies the RxD1 pin */
203
204     /* ---- Enable transmission, reception, and interrupt ---- */
205     SCIF1.SCSCR.WORD |= 0x00f0u;
206
207     /* ---- Enable TXI1, RXI1 interrupts ---- */
208     INTC.IPR17.BIT._SCIF1 = 15; /* Specifies as level 15 */
209   }
210
```

3.7 Sample Program Listing "main.c" (6/8)

```
211  /*""FUNC COMMENT""*************************************************************
212   * ID            :
213   * Outline       : Interrupt processing by break signal or overrun error (BRI).
214   *-----------------------------------------------------------------------------
215   * Include       : "iodefine.h"
216   *-----------------------------------------------------------------------------
217   * Declaration   : void io_interrupt_scif1_bri(void);
218   *-----------------------------------------------------------------------------
219   * Description   : Processes the overrun error which occurs in clock synchronous mode.
220   *               : This sample program only clears the ORER bit.
221   *-----------------------------------------------------------------------------
222   * Argument      : void
223   *-----------------------------------------------------------------------------
224   * Return Value  : void
225   *-----------------------------------------------------------------------------
226   * Note          :
227   *""FUNC COMMENT END""*********************************************************/
228  void io_interrupt_scif1_bri(void)
229  {
230    volatile unsigned short dummy;
231
232    /* ==== Checks the overrun error ==== */
233    if(SCIF1.SCLSR.BIT.ORER == 1) {
234
235      /* ---- Clears the interrupt source (clears the ORER bit) ---- */
236      SCIF1.SCLSR.BIT.ORER = 0;
237      dummy = SCIF1.SCLSR.WORD;  /* Dummy read to make sure to clear the interrupt source */
238    }
239  }
```

3.8 Sample Program Listing "main.c" (7/8)

```
240  /*""FUNC COMMENT""*****************************************************************
241   * ID            :
242   * Outline       : Receive-FIFO-data-full or data-ready interrupt (RXI)
243   *-------------------------------------------------------------------------------
244   * Include       : "iodefine.h"
245   *-------------------------------------------------------------------------------
246   * Declaration   : void io_interrupt_scif1_rxi(void);
247   *-------------------------------------------------------------------------------
248   * Description   : Executes the SCIF1 RXI interrupt processing. The receive-FIFO
249   *               : -data-full interrupt activates the processing in clock
250   *               : synchronous mode. This sample program configures the number of
251   *               : receive FIFO data triggers as 8, and reads data in units of 8.
252   *               : Note that the data ready interrupt does not occur when
253   *               : receiving data less than the number of receive FIFO data
254   *               : triggers in clock synchronous mode.
255   *-------------------------------------------------------------------------------
256   * Argument      : void
257   *-------------------------------------------------------------------------------
258   * Return Value  : void
259   *-------------------------------------------------------------------------------
260   * Note          :
261   *""FUNC COMMENT END""*************************************************************/
262  void io_interrupt_scif1_rxi(void)
263  {
264    int fifo_cnt;
265    volatile unsigned short dummy;
266
267    /* ==== Calculates the readable size in the receive FIFO ==== */
268    fifo_cnt = 8;
269
270    /* ==== Reads data from the receive FIFO ==== */
271    while(fifo_cnt-- != 0 ){
272
273      /* ---- Reads the received data ---- */
274      RcvData[RcvCnt++] = SCIF1.SCFRDR.BYTE;
275
276      /* ---- Reading all data is completed? ---- */
277      if(RcvCnt == DataNum){
278          /* ---- Disables the reception ---- */
279          SCIF1.SCSCR.BIT.RE = 0;
280          break;
281      }
282    }
283    /* ==== Clears the interrupt source (clears the RDF bit) ==== */
284    SCIF1.SCFSR.BIT.RDF = 0;
285    dummy = SCIF1.SCFSR.WORD;   /* Dummy read to make sure to clear the interrupt source */
286  }
```

3.9 Sample Program Listing "main.c" (8/8)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : SCIF1 transmit interrupt (TXI).
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_interrupt_scif1_txi(void);
 *-----------------------------------------------------------------------------
 * Description   : Executes the SCIF1 TXI interrupt processing. The transmit-FIFO-
 *               : data-empty interrupt activates this processing. This sample program
 *               : writes the transmit data in eight bytes, according to the number
 *               : of received data in the receive processing. When there is no data
 *               : to transmit, it disables the transmit-FIFO-data-empty interrupt.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          : After completing to write all transmit data, check to see the
 *               : transmit end (SCFSR.TEND) bit that all transmit data in FIFO
 *               : is emitted.
 *""FUNC COMMENT END""*************************************************************/
void io_interrupt_scif1_txi(void)
{
  int i, fifo_cnt;
  volatile unsigned short dummy;

  /* ==== Calculates the readable size in the transmit FIFO ==== */
  fifo_cnt = 8;

  /* ==== Writes data in the transmit FIFO ==== */
  while(fifo_cnt-- != 0 ){

    /* ---- Writes the transmit data ---- */
    SCIF1.SCFTDR.BYTE = SndData[SndCnt++];

    /* ---- Writing all data is completed? ---- */
    if(SndCnt == DataNum){
        /* ---- Disables the transmit interrupt ---- */
        SCIF1.SCSCR.BIT.TIE = 0;
        break;
    }
  }
  /* ==== Clears the interrupt source, and starts transmission (Clears bits TDFE,and TEND) ==== */
  SCIF1.SCFSR.WORD &= ~0x0060u;
  dummy = SCIF1.SCFSR.WORD;       /* Dummy read to make sure to clear the interrupt source */
}

/* End of File */
```

4. References

- Software Manual
  SH-2A/SH2A-FPU Software Manual Rev. 3.00
  The latest version of the software manual can be downloaded from the Renesas website.

- Hardware Manual
  SH7262 Group, SH7264 Group Hardware Manual Rev. 2.00
  The latest version of the hardware manual can be downloaded from the Renesas website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Apr.28.10	—	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.
5. 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.

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