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 Clocked Synchronous Serial Communication (Unidirectional Communication)

Introduction

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

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

- Channel 1 of the SCIF is initialized as a transmission module in clocked synchronous mode.
- Channel 2 of the SCIF is initialized as a reception module in clocked synchronous mode.
- Data transmitted from SCIF channel 1 is received by SCIF channel 2.

1.2 MCU Functions Used

- SCIF channel 1: Used as a transmission module
- SCIF channel 2: Used as a reception module

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

Data transmission/reception is synchronized with clock signal pulses in SCIF clocked synchronous mode. Either an internal clock or an external clock input from the SCK pin can be selected as the clock source. When an internal clock is selected, the serial clock is output via the SCK pin. When an external clock is selected, the serial clock is input from the SCK pin.

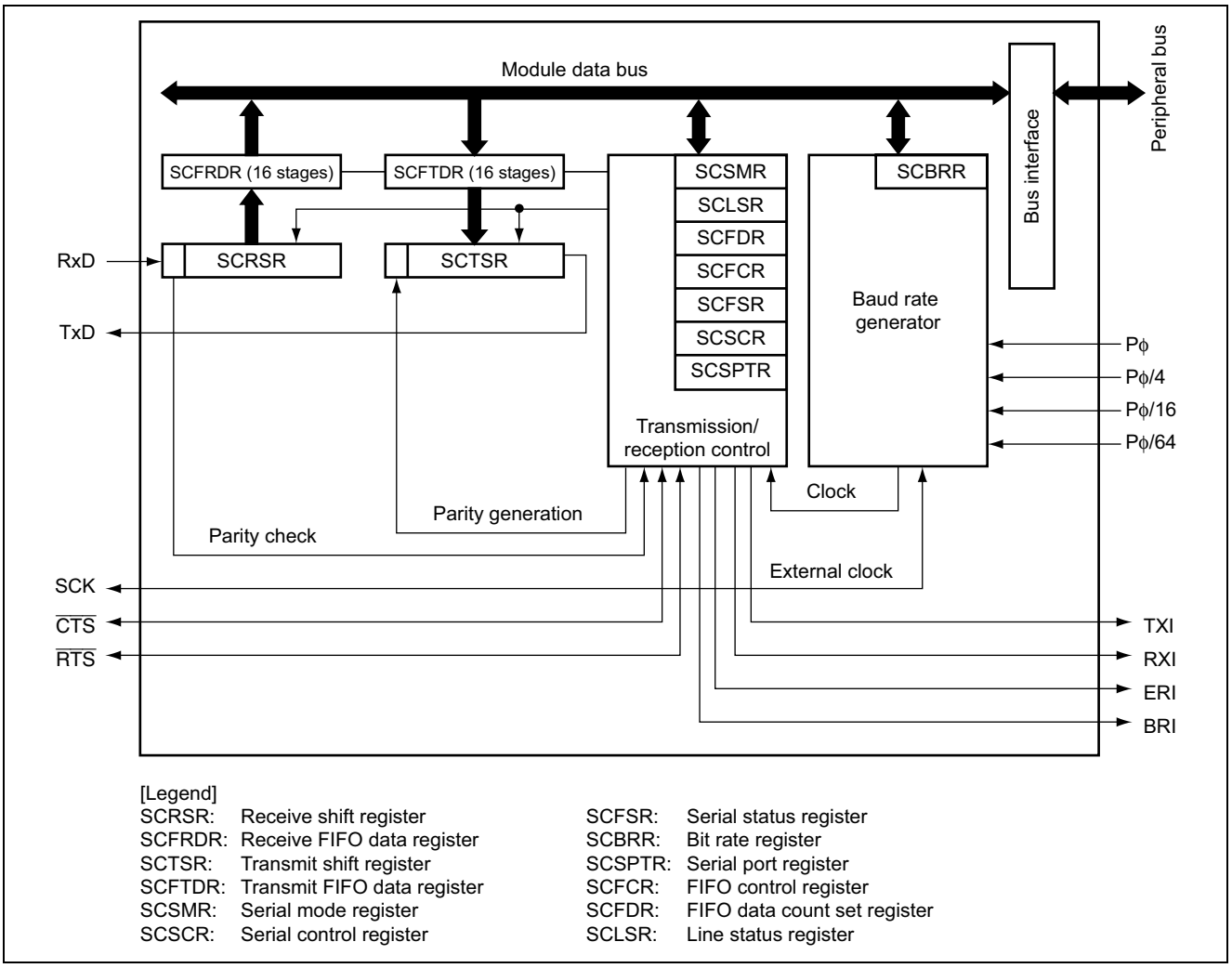
Communication data format is fixed at 8-bit length.

Table 1 summarizes the features of clocked synchronous mode. Figure 1 is a block diagram of the SCIF.

Table 1 Summary of SCIF (Clocked Synchronous 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
Data format	Transfer data length: 8 bits Transfer order: LSB first
Baud rate	Internal clock: 500 bps to 1000 kbps (Pϕ = 33 MHz) External clock: 5500 kbps at maximum (Pϕ = 33 MHz and externally input clock = 5.5 MHz)
Error detection	Overrun error
Interrupt requests	Transmit FIFO data empty interrupt (TXI), receive FIFO data full interrupt (RXI), and break interrupt (BRI), receive error (overrun error) interrupt (ERI)
Others	• When an internal clock is selected, the serial clock is output from the SCK pin. • Clock supply to unused channels can be stopped to save power. • 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), of the SH7206 Group Hardware Manual.



[Legend]

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

Figure 1 Block Diagram of SCIF

2.2 Procedure for Setting the MCU Modules

This section describes the basic setting procedures for SCIF clocked synchronous mode. Figures 2 and 3 show the example flow for initial settings of data transmission in clocked synchronous mode. Figures 4 and 5 show the example flow of initial settings for data reception in clocked synchronous mode. Figure 6 shows the example flow of data transmission processing. Figure 7 shows the example flow of data reception processing.

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

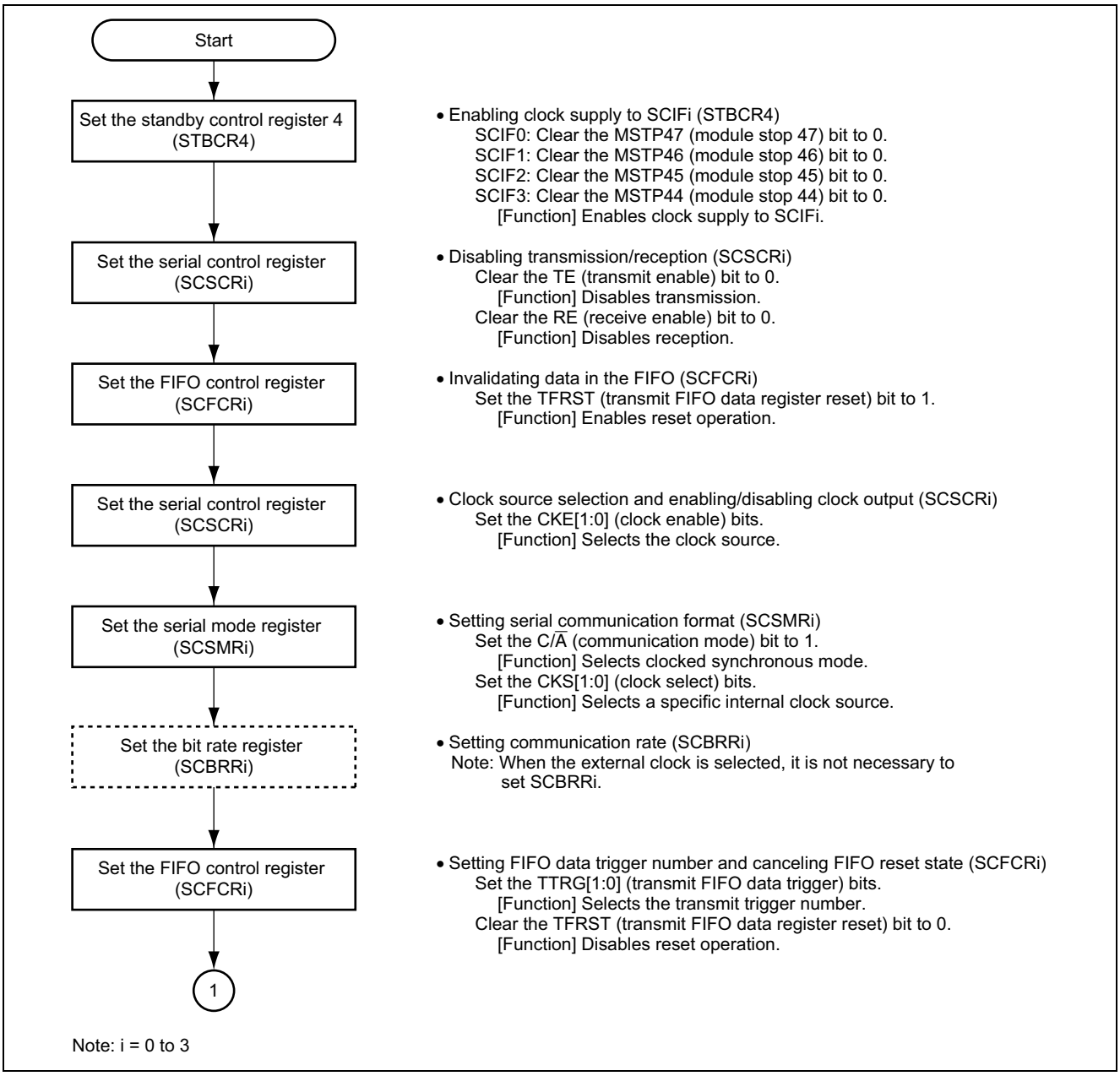


Figure 2 Example Flow of Initial Settings for Transmission in Clocked Synchronous Mode (1)

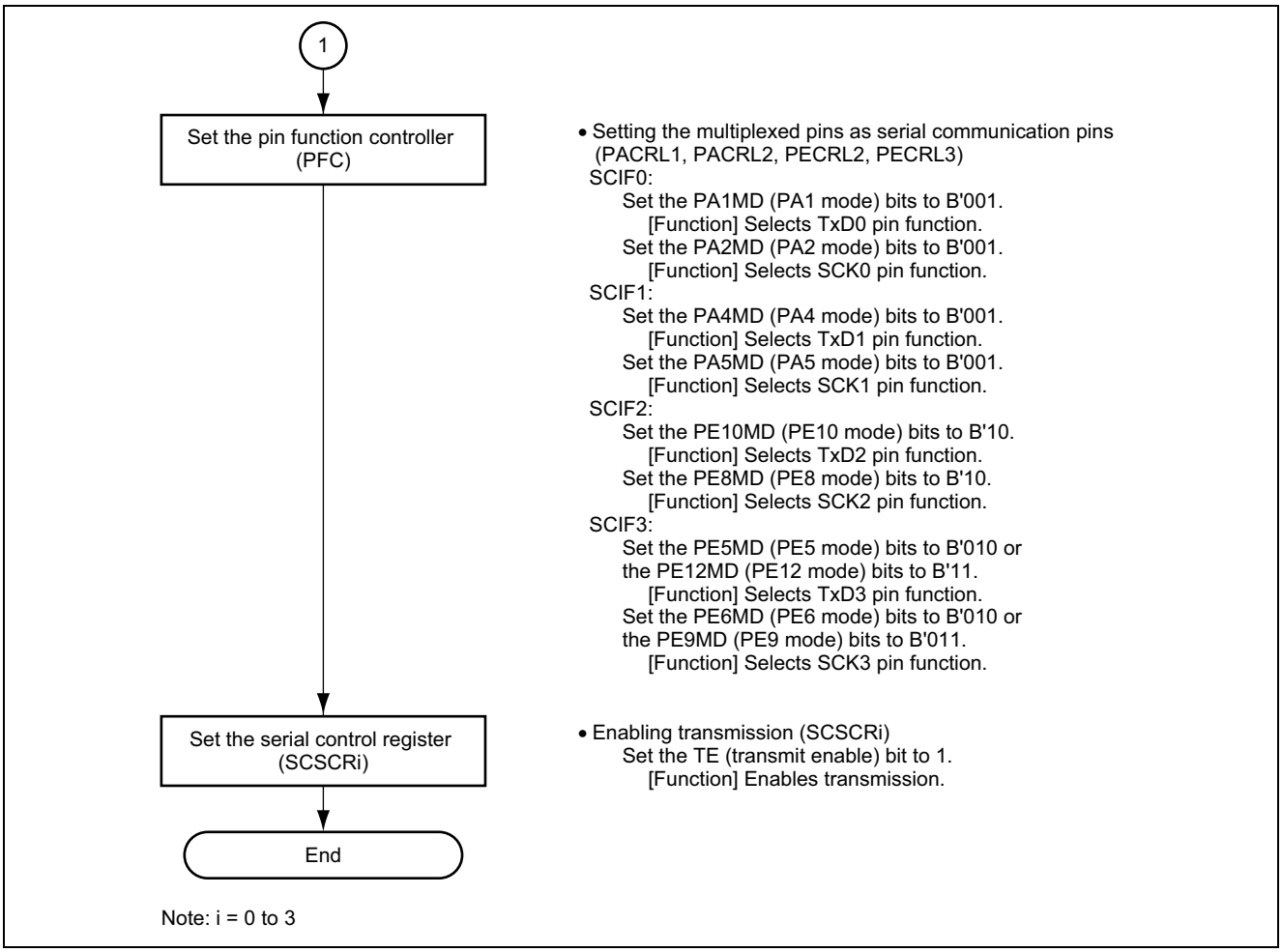
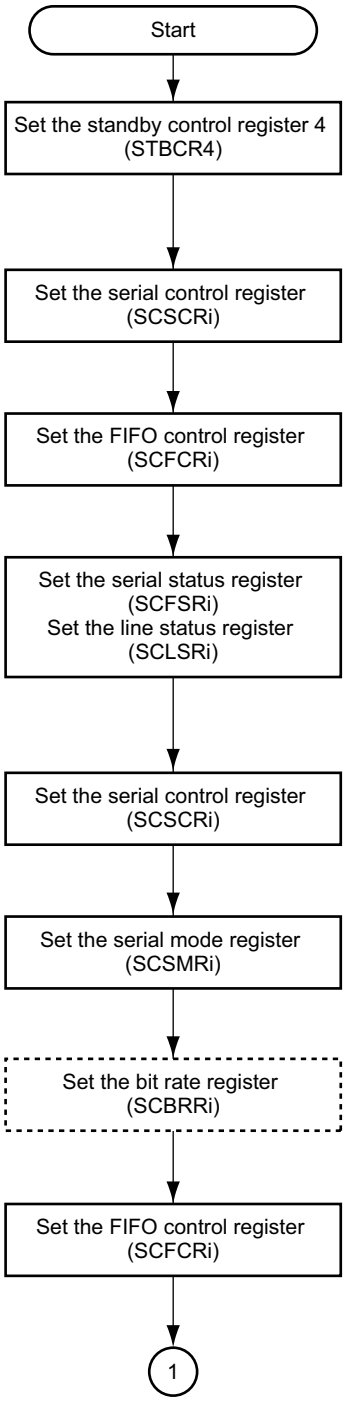


Figure 3 Example Flow of Initial Settings for Transmission in Clocked Synchronous Mode (2)



- Enabling clock supply to SCIFi (STBCR4)
  - SCIF0: Clear the MSTP47 (module stop 47) bit to 0.
  - SCIF1: Clear the MSTP46 (module stop 46) bit to 0.
  - SCIF2: Clear the MSTP45 (module stop 45) bit to 0.
  - SCIF3: Clear the MSTP44 (module stop 44) bit to 0.
  - [Function] Enables clock supply to SCIFi.

- Disabling transmission/reception (SCSCRi)
  - Clear the TE (transmit enable) bit to 0.
    - [Function] Disables transmission.
  - Clear the RE (receive enable) bit to 0.
    - [Function] Disables reception.

- Invalidating data in the FIFO (SCFCRi)
  - Set the RFRST (receive FIFO data register reset) bit to 1.
    - [Function] Enables reset operation.

- Clearing error status (SCFSRi, SCLSRi)
  Read each bit, and then write 0.
  - Clear the ER (receive error) bit to 0.
    - [Function] Clears the receive error flag.
  - Clear the BRK (break detection) bit to 0.
    - [Function] Clears the break detection flag.
  - Clear the ORER (overrun error) bit to 0.
    - [Function] Clears the overrun error flag.

- Clock source selection and enabling/disabling clock output (SCSCRi)
  - Set the CKE[1:0] (clock enable) bits.
    - [Function] Selects the clock source.

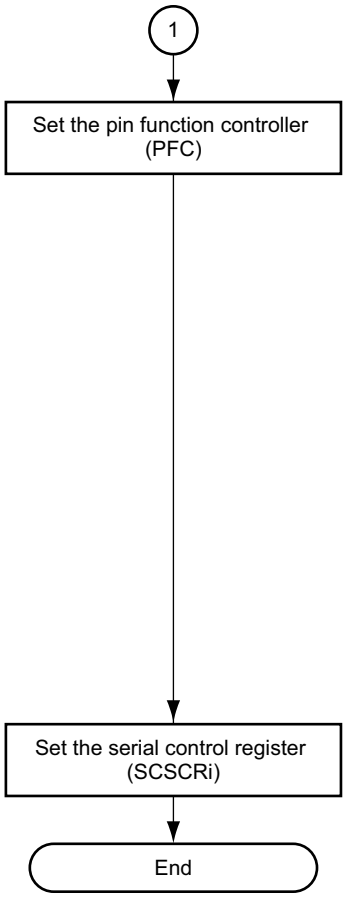
- Setting serial communication format (SCSMRi)
  - Set the C/$\overline{A}$ (communication mode) bit to 1.
    - [Function] Selects clocked synchronous mode.
  - Set the CKS[1:0] (clock select) bits.
    - [Function] Selects an internal clock source.

- Setting communication rate (SCBRRi)
  Note: When the external clock is selected, it is not necessary to set SCBRRi.

- Setting FIFO data trigger number and canceling FIFO reset state (SCFCRi)
  - Set the RTRG[1:0] (receive FIFO data trigger) bits.
    - [Function] Selects the receive trigger number.
  - Clear the RFRST (receive FIFO data register reset) bit to 0.
    - [Function] Disables reset operation.

Note: i = 0 to 3

Figure 4 Example Flow of Initial Settings for Reception in Clocked Synchronous Mode (1)



• Setting the multiplexed pins as serial communication pins (PACRL1, PACRL2, 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'010.
[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 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.

Figure 5 Example Flow of Initial Settings for Reception in Clocked Synchronous Mode (2)

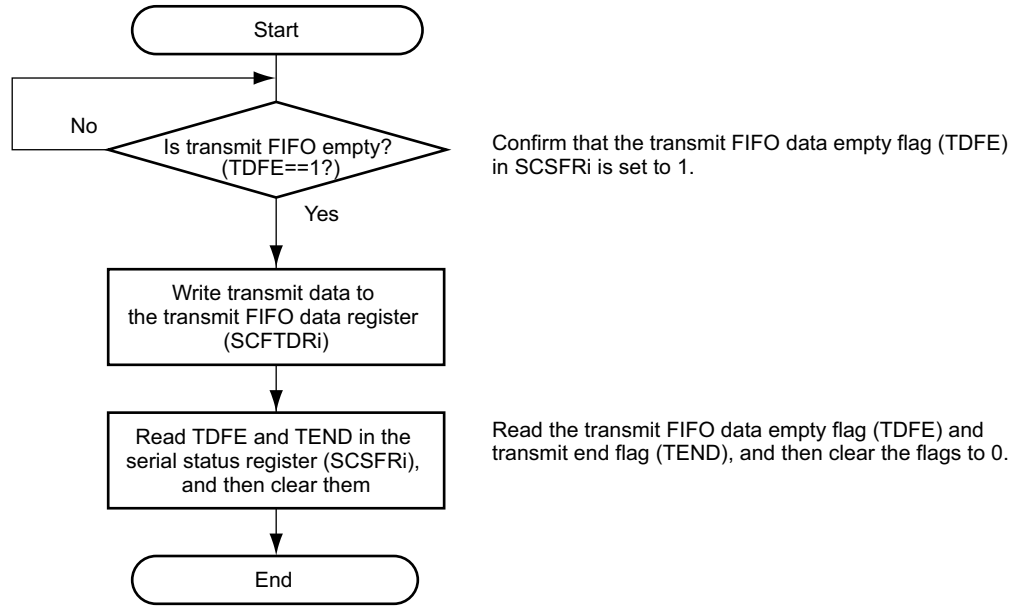


Figure 6 Example Flow of Transmission Processing in Clocked Synchronous Mode

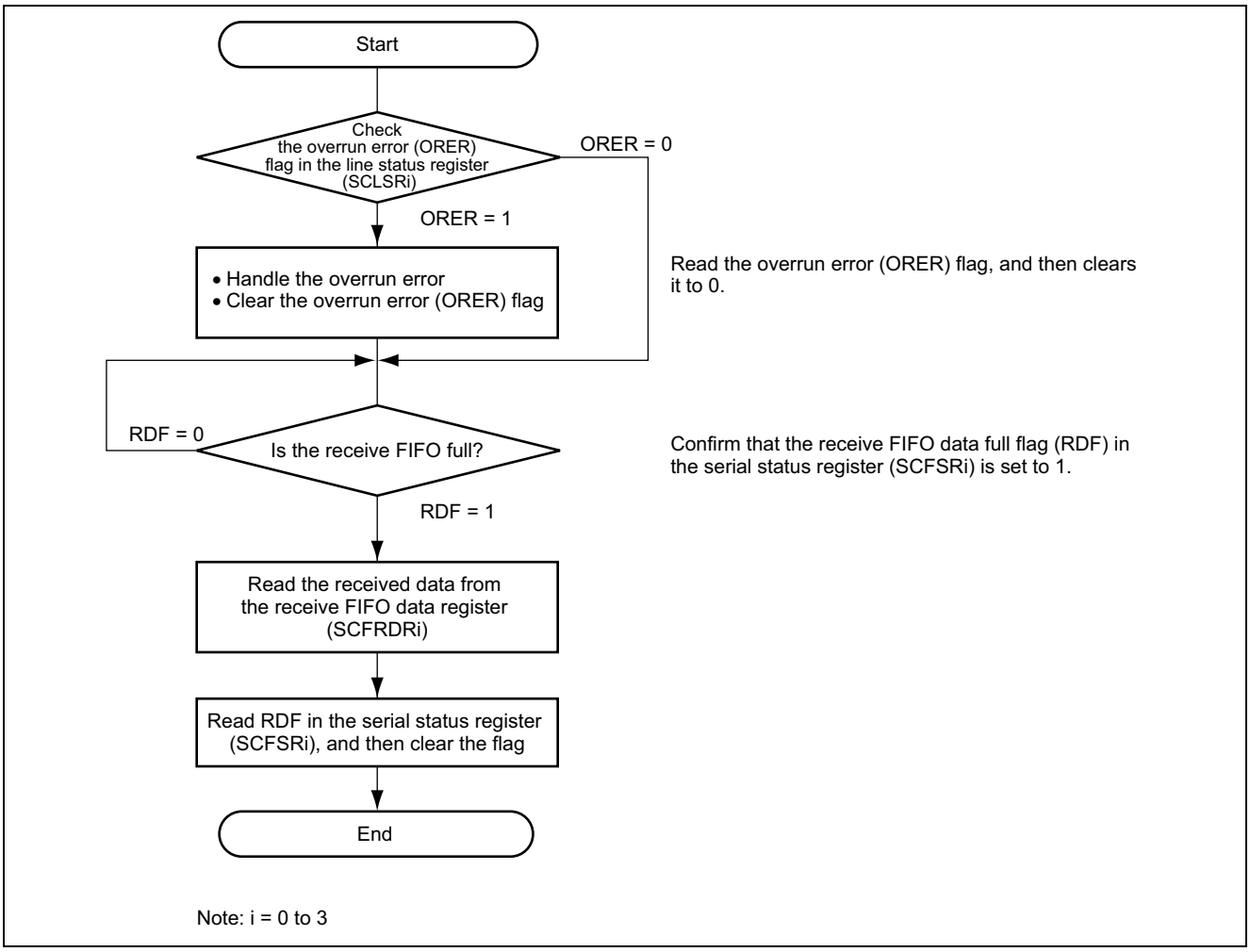


Figure 7 Example Flow of Reception Processing in Clocked Synchronous Mode

2.3 Operation of Sample Program

The sample program uses SCIF channel 1 as a data transmission module, and SCIF channel 2 as a data reception module. Communication is carried out using the loop-back test function, in which the serial transmit pin TxD1 of SCIF channel 1 and the serial receive pin RxD2 of SCIF channel 2 are internally connected.

SCIF channel 1 is set to output the serial clock, and SCIF channel 2 is set to allow the external clock to be input. To share the clock, the SCK1 and SCK2 pins are connected. Table 2 shows the communication function settings of the sample program. Figure 8 illustrates the timing of the sample program operation.

Table 2 Communication Function Settings of Sample Program

Communication Format	Function Setting
Communication mode	Clocked synchronous mode
Channel used	SCIF1: Data transmission, SCIF2: Data reception
Interrupts	Not used
Communication rate	100 kbps
Data size	8 bits
Bit order	LSB first
Serial clock	SCIF1: Serial clock output; SCIF2: External clock input
FIFO data trigger number	SCIF1: 0; SCIF2: 1

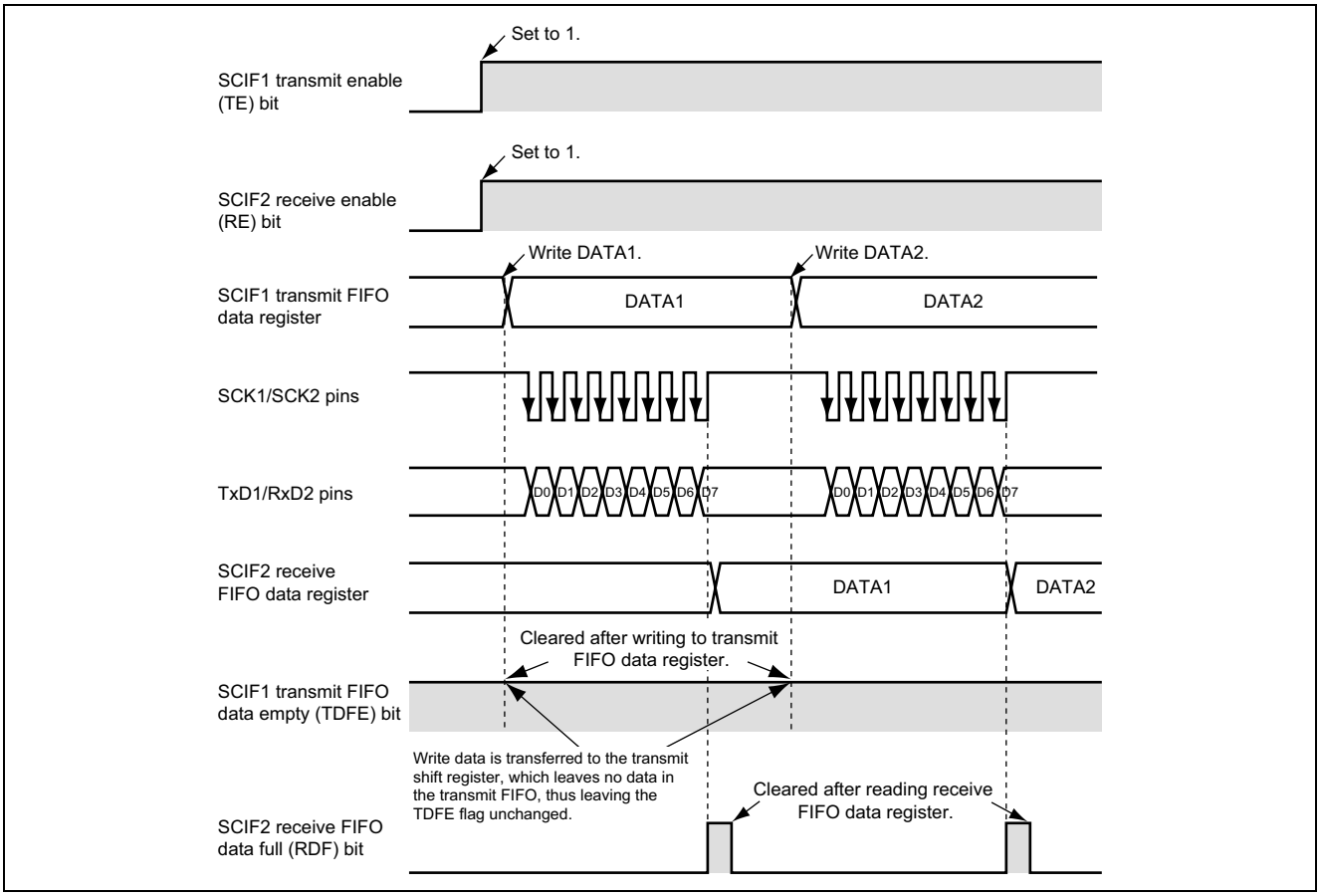


Figure 8 Timing of Sample Program Operation

2.4 Register Settings and Processing Sequence of Sample Program

The sample program initializes SCIF channel 1 in clocked synchronous transmission mode, and SCIF channel 2 in clocked synchronous reception mode. Afterwards, the program alternately repeats the processing of transmitting character string data from SCIF channel 1 and the processing of receiving the character string by SCIF channel 2.

The register settings for SCIF channel 1 and SCIF channel 2 of the sample program are shown in tables 3 and 4, and the processing flow of the sample program are shown in figure 9.

Table 3 Register Settings for Transmission Module

Register Name	Address	Setting	Description
Standby control register 4 (STBCR4)	H'FFFE040C	H'B4	MSTP46 = 0: The SCIF1 runs (clock is supplied).
Port A control register L2 (PACRL2)	H'FFFE3814	H'0011	PA4MD[2:0] = B'001: TxD1 output mode (SCIF1) PA5MD[2:0] = B'001: SCK1 input/output mode (SCIF1)
Serial mode register_1 (SCSMR_1)	H'FFFE8800	H'0080	C/$\overline{A}$ = 1: clocked synchronous mode CKS[1:0] = B'00: Pϕ clock
Serial control register_1 (SCSCR_1)	H'FFFE8808	H'0000	TE = 0: Transmission disabled. RE = 0: Reception disabled. CKE[1:0] = B'00: The SCK pin outputs an internal clock as the serial clock.
		H'0020	TE = 1: Transmission enabled.
FIFO control register_1 (SCFCR_1)	H'FFFE8818	H'0004	TFRST = 1: Reset operation for the transmit FIFO data register is enabled.
		H'0030	TFRST = 0: Reset operation for the transmit FIFO data register is disabled. TTRG[1:0] = B'11: The transmit FIFO data trigger number is 0. *
Bit rate register_1 (SCBRR_1)	H'FFFE8804	H'07	100 kbps

Note: * Transmit FIFO data trigger number is the number of remaining data bytes in the transmit FIFO which sets the transmit FIFO data empty (TDFE) flag in the serial status register (SCFSR).

Table 4 Register Settings for Reception Module

Register Name	Address	Setting	Description
Standby control register 4 (STBCR4)	H'FFFE040C	H'D4	MSTP45 = 0: The SCIF2 runs (clock is supplied).
Port E control register L2 (PECRL2)	H'FFFE3A14	H'2000	PE7MD[2:0] = B'010: RxD2 input mode (SCIF2)
Port E control register L3 (PECRL3)	H'FFFE3A12	H'0002	PE8MD[1:0] = B'10: SCK2 input/output mode (SCIF2)
Serial mode register_2 (SCSMR_2)	H'FFFE9000	H'0080	C/$\overline{A}$ = 1: Clocked synchronous mode
Serial control register_2 (SCSCR_2)	H'FFFE9008	H'0002	TE = 0: Transmission disabled RE = 0: Reception disabled CKE[1:0] = B'10: External clock is input from the SCK pin as the serial clock.
		H'0012	RE = 1: Reception enabled
FIFO control register_2 (SCFCR_2)	H'FFFE9018	H'0002	RFRST = 1: Reset operation for the receive FIFO data register is enabled.
		H'0000	RFRST = 0: Reset operation for the receive FIFO data register is disabled. RTRG[1:0] = B'00: The receive FIFO data trigger number is 1. *1
Bit rate register_2 (SCBRR_2)	H'FFFE9004	H'07	100 kbps
Serial status register_2 (SCFSR_2)	H'FFFE9010	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_2 (SCLSR_2)	H'FFFE9024	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 register value is ANDed with H'FF6E to clear the ER, BRK, and DR bits.

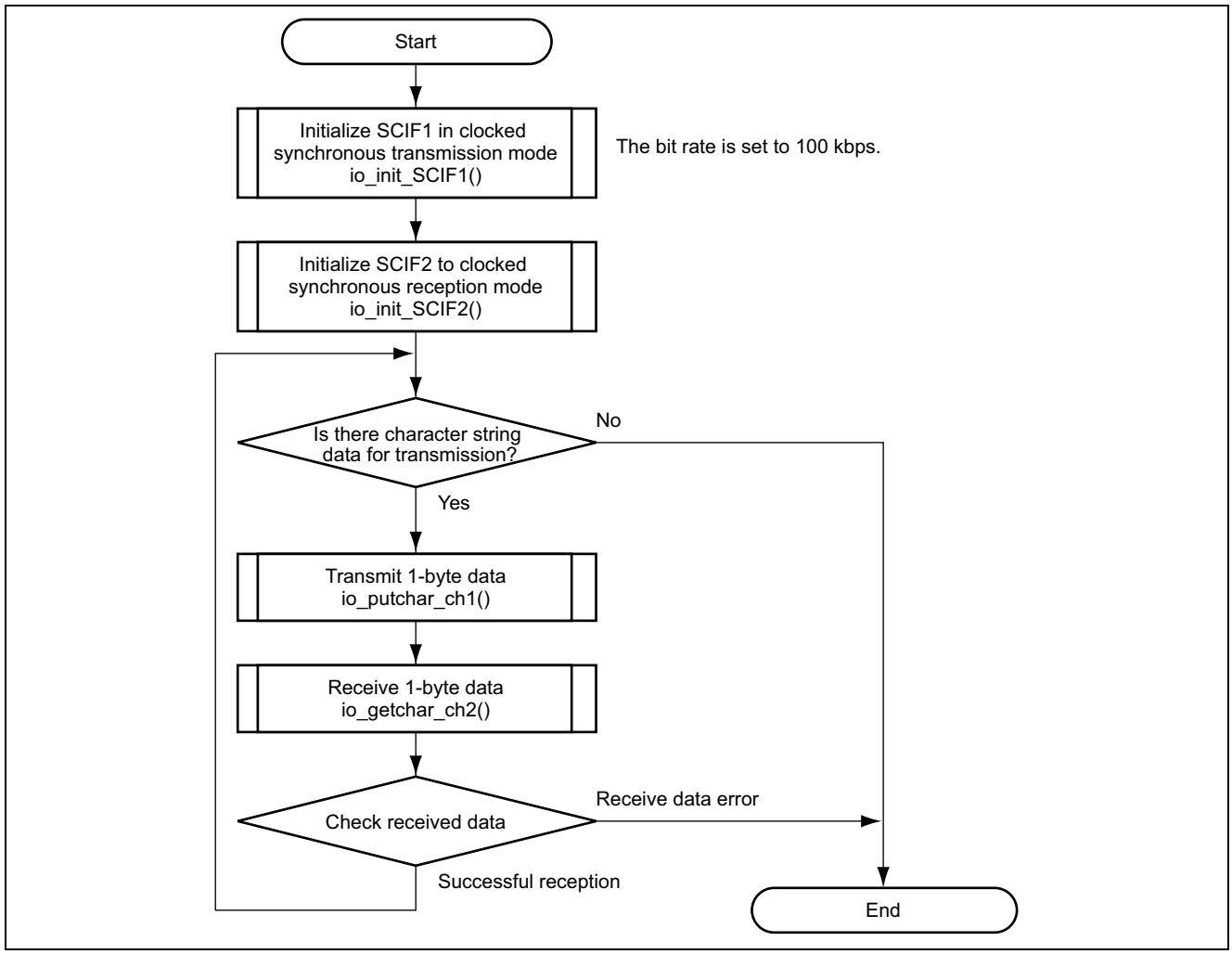


Figure 9 Main Processing Flow of Sample Program

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 clocked synchronous serial data 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 clocked synchronous data reception using SCIF1 and SCIF2
*
*                <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_SCIF1(int);
void io_init_SCIF2(void);
void io_putchar_ch1(unsigned char);
unsigned char io_getchar_ch2(void);

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

/* ---- Values for bit rate specification ---- */
enum{
    CBR_500,
    CBR_1K,
    CBR_2_5K,
    CBR_5K,
    CBR_10K,
    CBR_25K,
    CBR_50K,
    CBR_100K,
    CBR_250K,
    CBR_500K,
    CBR_1M
};

```

- Sample Program Listing: main.c (2)

```
/* ==== Bit rate setting value table ==== */
SH7206_BAUD_SET scif_baud[] = {
    {255, 3},     /*       500Hz  */
    {125, 3},     /*     1,000Hz  */
    {200, 2},     /*     2,500Hz  */
    {100, 2},     /*     5,000Hz  */
    {200, 1},     /*    10,000Hz  */
    { 80, 1},     /*    25,000Hz  */
    {160, 0},     /*    50,000Hz  */
    { 80, 0},     /*   100,000Hz  */
    { 31, 0},     /*   250,000Hz  */
    { 15, 0},     /*   500,000Hz  */
    {  7, 0}      /* 1,000,000Hz  */
};

/*""FUNC COMMENT""*****************************************************************************
* ID              :
* Module summary: Main function of the sample program
*                 (clocked synchronous serial I/O reception)
*-----------------------------------------------------------------------------------------------
* Include       : None
*-----------------------------------------------------------------------------------------------
* Declaration   : void main(void)
*-----------------------------------------------------------------------------------------------
* Functional description:
*               : Configures SCIF1 in clocked synchronous serial transmission mode,
*               : and SCIF2 in clocked synchronous serial reception mode. SCIF1 transmits
*               : character string data, and SCIF2 receives it.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Note          : The TxD1 pin must be connected to the RxD2 pin.
*               : The SCK1 pin must be connected to the SCK2 pin.
*""FUNC COMMENT END""*************************************************************************/
void main(void)
{
    const unsigned char send_data[] = "12345 ABCDEFG\n"; /* Character string for transmission*/
    const unsigned char *psnd;
    unsigned char rcv_data[16];
    unsigned char *prcv;

    /* ==== Initialize SCIF1 in clocked synchronous transmission mode ==== */
    io_init_SCIF1(CBR_100K);            /* Bit rate: 100 kbps                          */

    /* ==== Initialize SCIF2 in clocked synchronous reception mode ==== */
    io_init_SCIF2();

    psnd = send_data;                   /* Initialize the transmit data pointer        */
    prcv = rcv_data;                    /* Initialize the receive data pointer         */

    /* ==== Is there character data for transmission? ==== */
    while(*psnd != 0){
        /* ==== Transmit 1-byte data ==== */
        io_putchar_ch1(*psnd);
        /* ==== Receive 1-byte data ==== */
        *prcv = io_getchar_ch2();
        /* ==== Check the received data ==== */
        if(*psnd != *prcv){
            break;                      /* Receive data error                          */
        }
        psnd++;                         /* Update the pointers                         */
        prcv++;
    }

    while(1){
        /* End of Program */
    }
}
```

- Sample Program Listing: main.c (3)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module Summary: SCIF1 initialization
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : void io_init_SCIF1(int bps)
*-----------------------------------------------------------------------------------------------
* Functional description:
*               : Initializes SCIF1 in clocked synchronous serial mode.
*               : Transmission enabled / serial clock output /
*               : transmit FIFO data trigger number = 0
*               : The bit rate is specified in bps.
*-----------------------------------------------------------------------------------------------
* Argument      : int bps : Index of the bit rate setting table
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Note          : 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_SCIF1(int bps)
{
    /* ==== Canceling power-down mode ==== */
    /* ---- Set standby control register 4 (STBCR4) ---- */
    CPG.STBCR4.BIT.MSTP46 = 0;          /* Start clock supply to the SCIF1                */

    /* ==== SCIF1 initialization ==== */
    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF1.SCSCR.WORD = 0x0000;          /* Disable transmission/reception by SCIF1        */

    /* ---- Set FIFO control register (SCFCRi) ---- */
    SCIF1.SCFCR.BIT.TFRST = 1;          /* Reset the transmit FIFO data register          */

    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF1.SCSCR.BIT.CKE = 0x0;          /* B'00: Internal clock; serial clock output      */

    /* ---- Set serial mode register (SCSMRi) ---- */
    SCIF1.SCSMR.WORD = scif_baud[bps].scsmr | 0x0080u;
                                        /* Communication mode 1: Clocked synchronous mode */
                                        /* Clock select        : Table value              */

    /* ---- Set bit rate register (SCBRRi) ---- */
    SCIF1.SCBRR.BYTE = scif_baud[bps].scbrr;

    /* ---- Set FIFO control register (SCFCRi) ---- */
    SCIF1.SCFCR.WORD = 0x0030;          /* Transmit FIFO data trigger number: 0           */
                                        /* Transmit FIFO data register reset: Disabled    */

    /* ==== Setting pin function controller (PFC) ==== */
    PORT.PACRL2.BIT.PA4MD = 1;          /* Set the PA4 pin for TxD1 output (PACRL2)       */
    PORT.PACRL2.BIT.PA5MD = 1;          /* Set the PA5 pin for SCK1 input/output (PACRL2) */

    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF1.SCSCR.BIT.TE = 1;             /* Enable transmission by SCIF1 */
}
```

- Sample Program Listing: main.c (4)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module Summary: SCIF2 initialization
*-----------------------------------------------------------------------------------------------
* Include       : None
*-----------------------------------------------------------------------------------------------
* Declaration   : void io_init_SCIF2(void)
*-----------------------------------------------------------------------------------------------
* Functional description:
*               : Initializes SCIF2 in clocked synchronous serial mode.
*               : Reception enabled / serial clock input /
*               : receive FIFO data trigger number = 1
*               : The bit rate is specified in bps.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Note          : None
*""FUNC COMMENT END""*************************************************************************/
void io_init_SCIF2(void)
{
    /* ==== Canceling power-down mode ==== */
    /* ---- Set standby control register 4 (STBCR4) ---- */
    CPG.STBCR4.BIT.MSTP45 = 0;          /* Start clock supply to SCIF2                    */

    /* ==== SCIF2 initialization ==== */
    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF2.SCSCR.WORD = 0x0000;          /* Disable transmission/reception by SCIF2        */

    /* ---- Setting FIFO control register (SCFCRi) ---- */
    SCIF2.SCFCR.BIT.RFRST = 1;          /* Reset the receive FIFO data register           */

    /* ---- Set serial status register (SCFSRi) ---- */
    SCIF2.SCFSR.WORD &= 0xff6e;         /* Clear the ER, BRK, and DR bits                 */

    /* ---- Set line status register (SCLSRi) ---- */
    SCIF2.SCLSR.BIT.ORER = 0;           /* Clear the ORER bit                             */

    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF2.SCSCR.BIT.CKE = 0x2;          /* B'10: External clock; serial clock input       */

    /* ---- Set serial mode register (SCSMRi) ---- */
    SCIF2.SCSMR.WORD = 0x0080;          /* Communication mode 1: Clocked synchronous mode */

    /* ---- Set FIFO control register (SCFCRi) ---- */
    SCIF2.SCFCR.WORD = 0x0000;          /* Receive FIFO data trigger number: 1 */
                                        /* Modem control enable            : Disabled     */
                                        /* Receive FIFO data register reset: Disabled     */

    /* ==== Setting pin function controller (PFC) ==== */
    PORT.PECRL2.BIT.PE7MD = 2;          /* Set the PE7 pin for RxD2 input (PECRL2)        */
    PORT.PECRL3.BIT.PE8MD = 2;          /* Set the PE8 pin for SCK2 input/output (PECRL3) */

    /* ---- Set serial control register (SCSCRi) ---- */
    SCIF2.SCSCR.BIT.RE = 1;             /* Enable reception by SCIF2                      */
}
```

- Sample Program Listing: main.c (5)

```
241 /*""FUNC COMMENT""*****************************************************************************
242 * ID            :
243 * Module summary: SCIF1 1-byte (one character) transmission processing
244 *-----------------------------------------------------------------------------------------------
245 * Include       : #include "iodefine.h"
246 *-----------------------------------------------------------------------------------------------
247 * Declaration   : void io_putchar_ch1(unsigned char c)
248 *-----------------------------------------------------------------------------------------------
249 * Functional description:
250 *               : Checks the transmit FIFO data empty flag in the SCIF1 serial status register
251 *               : (SCFSR1) to see if SCIF1 is ready for the next transmission (FIFO empty).
252 *               : If it is, transmits the 1-byte data passed as an argument.
253 *-----------------------------------------------------------------------------------------------
254 * Argument      : unsigned char c : Transmit data
255 *-----------------------------------------------------------------------------------------------
256 * Return value  : None
257 *-----------------------------------------------------------------------------------------------
258 * Note          : None
259 *""FUNC COMMENT END""*************************************************************************/
260 void io_putchar_ch1(unsigned char c)
261 {
262     /* ==== Transmit FIFO empty? (TDFE==1?) ==== */
263     while(SCIF1.SCFSR.BIT.TDFE == 0){
264         /* Wait until the TDFE flag is set */
265     }
266
267     /* ==== Write transmit data to transmit FIFO data register (SCFTDR1) ==== */
268     SCIF1.SCFTDR.BYTE = c;
269
270     /* ==== Read TDFE and TEND in serial status register (SCSFR1),
271                                                             and then clear the flags ==== */
272     SCIF1.SCFSR.WORD &= ~0x0060u;
273 }
274
```

- Sample Program Listing: main.c (6)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module summary: SCIF2 1-byte (one character) reception processing
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : unsigned char io_getchar_ch2(void)
*-----------------------------------------------------------------------------------------------
* Functional description:
*               : Checks the state of the receive FIFO data full flag in the SCIF2 serial
*               : status register (SCFSR2), and if it is set, reads the received data from
*               : the receive FIFO data register.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : Received data
*-----------------------------------------------------------------------------------------------
* Note          : None
*""FUNC COMMENT END""*************************************************************************/
unsigned char io_getchar_ch2(void)
{
    unsigned char data;

    /* ==== Check the overrun error (ORER) flag in line status register (SCLSR2)==== */
    if ( SCIF2.SCLSR.BIT.ORER == 1) {
        /* Perform overrun error processing */

        /* Clear the ORER flag */
        SCIF2.SCLSR.BIT.ORER = 0;

    }

    /* ==== Receive FIFO full? ==== */
    while(SCIF2.SCFSR.BIT.RDF == 0){
        /* Wait until the RDF flag is set */
    }

    /* ==== Read the received data in receive FIFO data register (SCFRDR2) ==== */
    data = SCIF2.SCFRDR.BYTE;

    /* == Read the RDF flag in serial status register (SCFSR2), and then clear the flag == */
    SCIF2.SCFSR.BIT.RDF = 0;

    return data;
}
/* 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.