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.

SH7262/SH7264 Group

Serial Sound Interface in Slave Receiver Mode

Summary

This application note describes an example of setting the SH7262/SH7264 Microcomputers (MCUs) Serial Sound Interface (SSI) in slave receiver mode.

Target Device

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

Contents

1. Introduction

1.1 Specifications

Sets the SH7264 Serial Sound Interface (SSI) in slave receiver mode to receive the PCM data. To transfer data from SSI, use the Direct Memory Access Controller (DMAC).

1.2 Modules Used

- Serial Sound Interface (SSI)
- Direct Memory Access Controller (DMAC)
- General-purpose I/O Ports
- Interrupt Controller

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequency	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Technology Corp. High-performance Embedded Workshop Ver.4.04.01
C compiler	Renesas Technology SuperH RISC engine Family C/C++ compiler package Ver.9.02 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 Note

Refer to the related application notes as follows:

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group Serial Sound Interface in Master Transmitter Mode
- SH7262/SH7264 Group Serial Sound Interface in Master Transceiver Mode

2. Applications

This application sets the sampling frequency of the SSI to 44.1 kHz to operate as the slave receiver.

2.1 SSI Operation

The SSI has the following features:

- Number of channels: 4
- Operating mode: Non-compressed mode

Non-compressed mode supports the serial audio streams divided by channels.

- Operates both as the transmitter and the receiver
- Channel 0 supports full-duplex transmission/reception
- Supports the serial bus format
- Asynchronous transfer between the data buffer and the shift register
- Clock divide ratio used in the serial bus interface selectable
- Controls the data transmission/reception by the DMAC or interrupts
- Oversampling clock options as follows:
  AUDIO_CLK pin
  AUDIO_X1, AUDIO_X2 pins
- Eight deep FIFO buffer included both in the transmitter and receiver

Figure 1 shows the SSI block diagram.

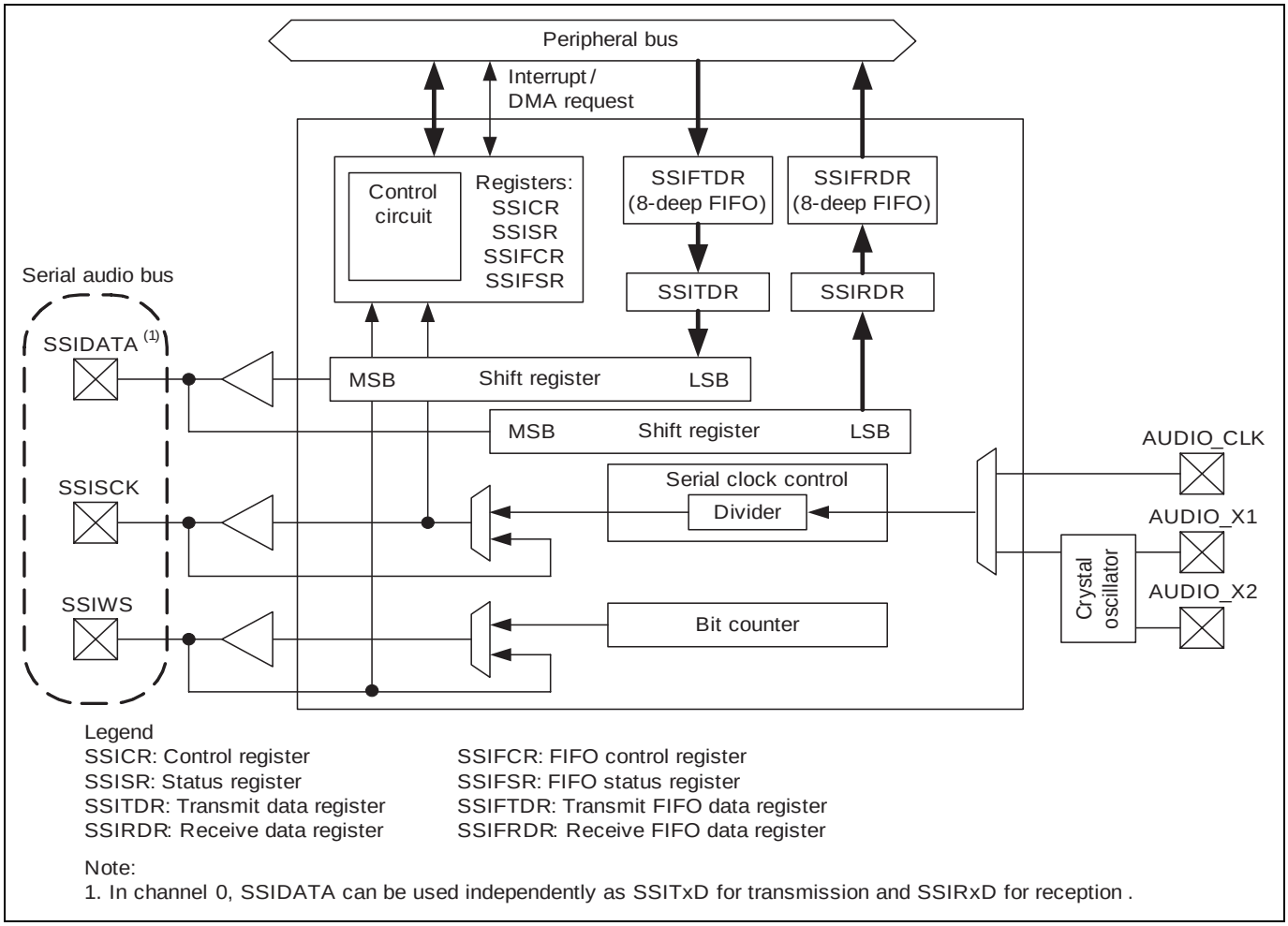


Figure 1 SSI Block Diagram

2.2 SSI Setting Procedure

Figure 2 shows the flow chart of setting the SSI. Figure 3 shows the flow chart of setting the DMAC.

Refer to the SH7262 Group, SH7264 Group Hardware Manual for details on registers.

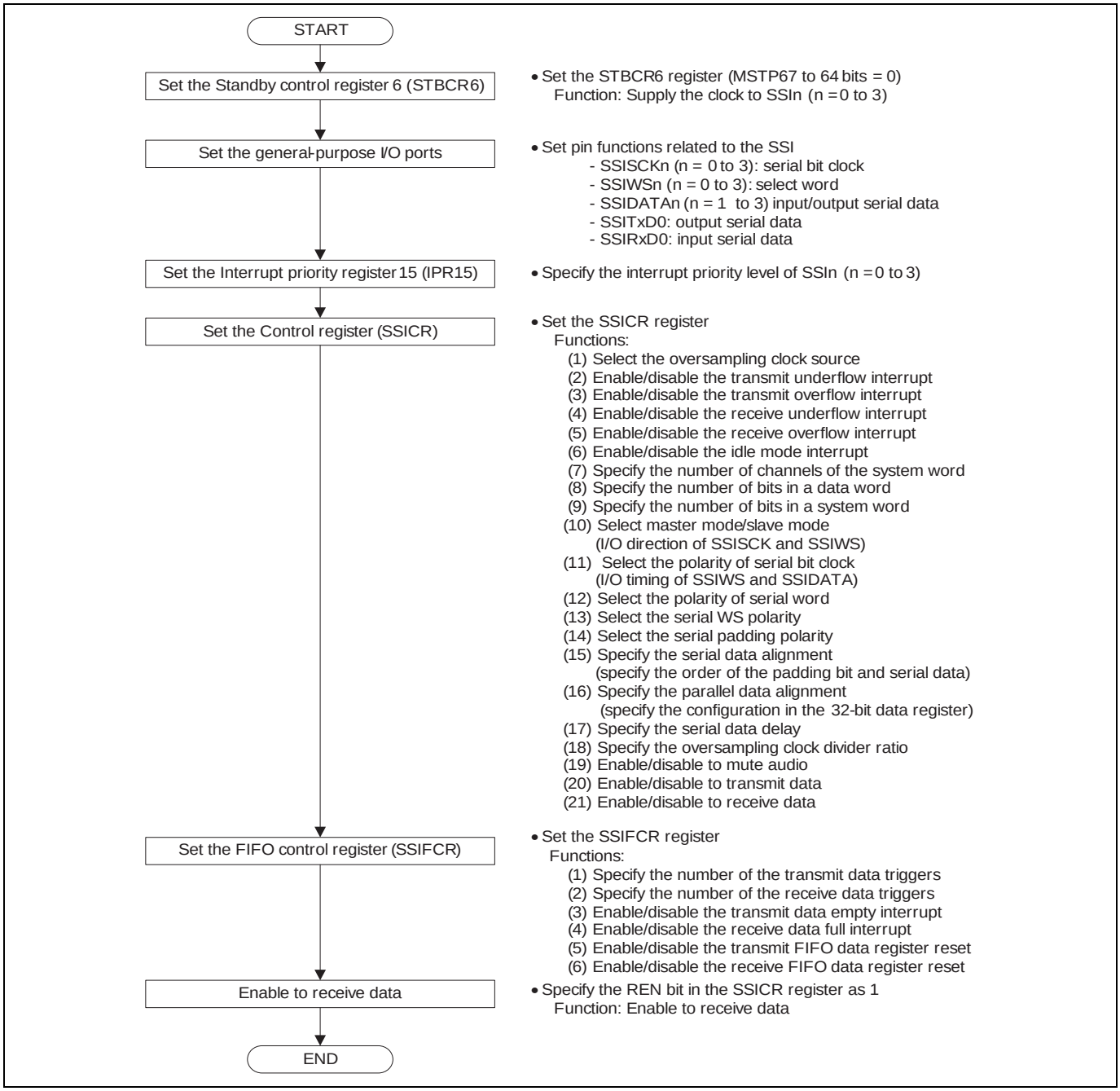


Figure 2 SSI Setup Flow Chart

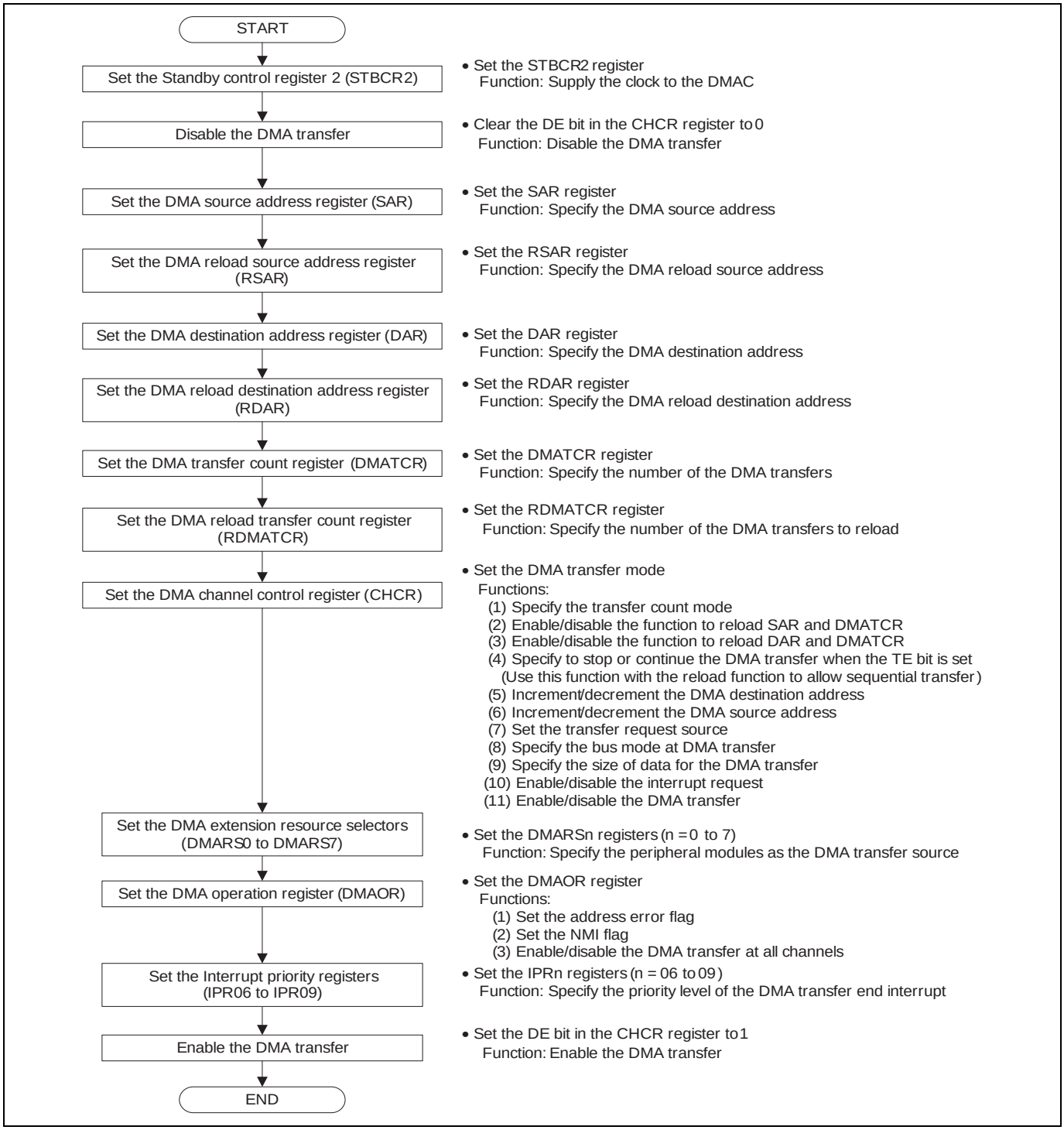


Figure 3 DMAC Setup Flow Chart

2.3 Sample Program Operation

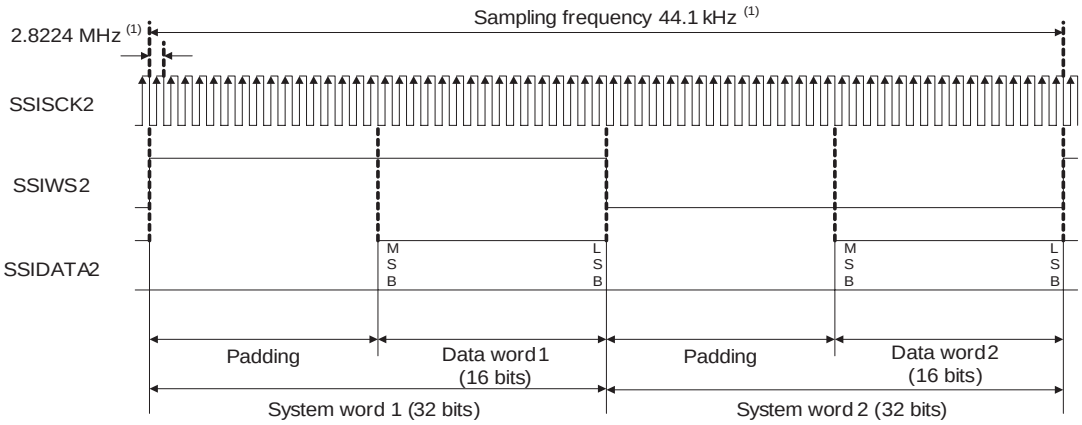
In the sample program, the SSI receives the data word from the SSIDATA2 pin, and transfers the data to the Receive FIFO data register (SSIFRDR) via the Receive data register (SSIRDR) in the SSI channel 2. When the data is transferred into the SSIFRDR register, the SSI activates the channel 2 of the DMAC by the DMA transfer request (transmit data full interrupt).

Set the receive buffer to store ten samples (40 bytes) of data as the transfer destination of the DMAC. The receive buffer consists of two planes and receive data sequentially by switching two planes back and forth. Read the data in the receive buffer by the DMA transfer end interrupt.

SSI setting in the sample program is as follows:

- Channel used: channel 2
- Operating mode: slave receiver
- Data transmission controlled by: DMAC
- Data word length: 16 bits
- System word length: 32 bits
- Padding bit: Low level
- No delay between the SSIWS and SSIDATA signals
- Outputs the SSIWS and SSIDATA signals at the falling edge of the SSISCK signal

Figure 4 shows the signal waveform in the sample program. Figure 5 shows the sample program block diagram.



Note 1: Sampling frequency depends on the clock frequency of the master.

Figure 4 Signal Output Waveform in the Sample Program

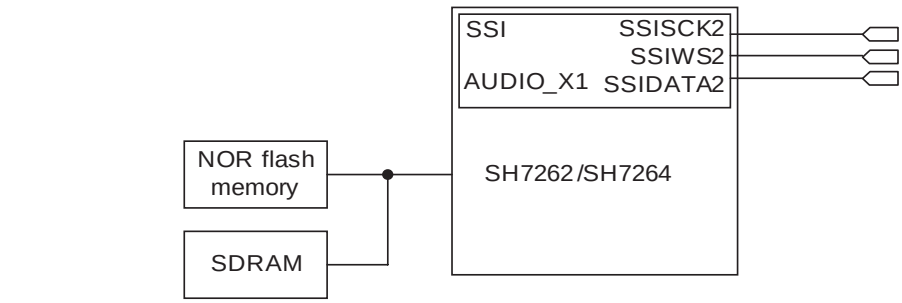


Figure 5 Sample Program Block Diagram

2.4 Sample Program Procedure

The table below lists the SSI registers setting in the sample program.

Table 2 lists the DMAC registers setting in the sample program. Figure 6 shows the flow charts of the sample program.

Table 1 SSI Registers Setting

Register Name	Address	Value	Description
Control register 2 (SSICR_2)	H'FFFF 1000	H'0C0B 1520	• CKS bit = B'0 (As this is slave mode, setting is not required) • TUIEN bit = B'0 (Disables the Transmit underflow interrupt) • TOIEN bit = B'0 (Disables the Transmit overflow interrupt) • RUIEN bit = B'1 (Enables the Receive underflow interrupt) • ROIEN bit = B'1 (Enables the Receive overflow interrupt) • IIEN bit = B'0 (Disables Idle mode interrupt) • CHNL [1:0] bits = B'00 (System words have one channel) • DWL [2:0] bits = B'001 (Data word length: 16 bits) • SWL [2:0] bits = B'011 (System word length: 32 bits) • SCKD bit = B'0 (Serial bit clock input, slave mode) • SWSD bit = B'0 (Serial word select input, slave mode) • SCKP bit = B'0 (Latches SSIWS and SSIDATA signals at the rising edge of the SSISCK signal) • SWSP bit = B'1 (SSIWS is High for the 1st channel, and is Low for the 2nd channel) • SPDP bit = B'0 (Padding bits are low) • SDTA bit = B'1 (Transmits and receives in the order of padding bits, and serial data) • PDTA bit = B'0 (Transmits and receives lower bits of parallel data) • DEL bit = B'1 (No delay between the SSIWS and SSIDATA) • CKDV bits [3:0]= B'0010 (Specifies the oversampling clock as audio Φ4) • MUEN bit = B'0 (Not muted) • TEN bit = B'0 (Disables to transmit data) • REN bit = B'0 (Disables to receive data)
		H'0C0B 1521	• REN bit = B'1 (Enables to receive data)
FIFO control register 2 (SSIFCR_2)	H'FFFF 1010	H'0000 0004	• TTRG[1:0] bits = B'00 (Number of transmit data triggers: 7) • RTRG [1:0] bits = B'00 (Number of receive data triggers: 1) • TIE bit = B'0 (Disables the transmit data empty interrupt) • RIE bit = B'1 (Enables the receive data full interrupt) • TFRST bit = B'0 (Disables to reset the transmit FIFO) • RFRST bit = B'0 (Disables to reset the receive FIFO)

Table 2 DMAC Registers Setting

Register Name	Address	Setting	Description
DMA channel control register 2 (CHCR_2)	H'FFFE 102C	H'0000 0000	• DE bit = B'0 (Disables the DMA transfer)
		H'1010 4814	• TC bit = B'0 (Transmits data once by one transfer request) • RLDSAR bit = B'0 (Disables the SAR reload function) • RLDDAR bit = B'1 (Enables the DAR reload function) • DAF bit, SAF bit = B'00 (Not used) • DO bit = B'0 (Not used) • TL bit = B'0 (Not used) • TEMASK bit = B'1 (Continues the DMA transfer when the TE bit is set) • HE bit, HIE bit = B'00 (Not used) • AM bit, AL bit = B'00 (Not used) • DM [1:0] bits = B'01 (Increments the destination address) • SM [1:0] bits = B'01 (Source address fixed) • RS [3:0] bits = B'1000 (Specifies the DMA extension resource) • DL bit, DS bit = B'00 (Not used) • TB bit = B'0 (Specifies the cycle steal mode) • TS [1:0] bits = B'10 (Specifies the longword transfer) • IE bit = B'1 (Enables an interrupt request) • DE bit = B'0 (Disables the DMA transfer)
		H'1010 4815	• DE bit = B'1 (Enables the DMA transfer)
DMA source address register 2 (SAR_2)	H'FFFE 1020	H'FFFF 101C	• Specifies the SSIFRDR register 2 as the DMA transfer source start address
DMA destination address register 2 (DAR_2)	H'FFFE 1024	Internal RAM	• Specifies the internal RAM as the DMA transfer destination start address
DMA reload destination address register 2 (RDAR_2)	H'FFFE 1124	Internal RAM	• Specifies the internal RAM as the DMA reload transfer destination start address
DMA transfer count register 2 (DMATCR_2)	H'FFFE 1028	H'0000 000A	• Number of transfers: 10
DMA reload transfer count register 2 (RDMATCR_2)	H'FFFE 1128	H'0000 000A	• Number of transfers: 10
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	• CMS [1:0] bits = B'00 (Normal mode) • PR [1:0] bits = B'00 (Channel priority level: Fixed mode 1) • AE bit = B'0 (Clears the address error flag) • NMIF bit = B'0 (Clear the NMI interrupt) • DME bit = B'1 (Enables the DMA transfer on all channels)
DMA extension resource selector 1 (DMARS1)	H'FFFE 1304	H'002B	• Specifies the SSI channel 2 as the transfer request source of the DMA channel 2

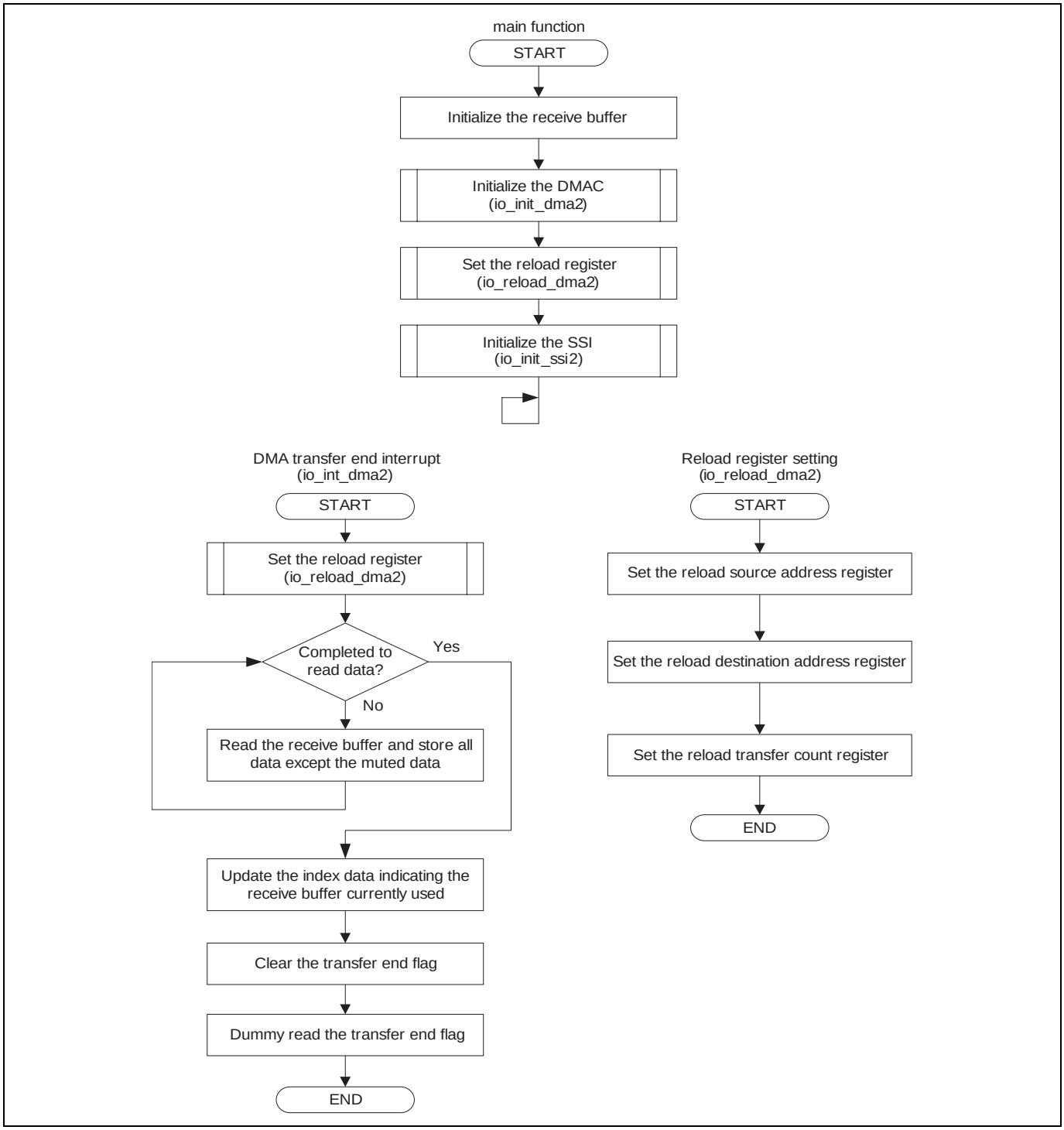


Figure 6 Sample Program Flow Chart

3. Sample Program Listing

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

```
1     /*""FILE COMMENT""************ Technical reference data ************************
2     *
3     *       System Name : SH7264 Sample Program
4     *       File Name   : main.c
5     *       Abstract    : SSI in Slave Receiver Mode
6     *       Version     : 1.00.00
7     *       Device      : SH7262/SH7264
8     *       Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
9     *                   : C/C++ compiler package for the SuperH RISC engine family
10    *                   :                              (Ver.9.02 Release00).
11    *       OS          : None
12    *       H/W Platform: M3A-HS64G50 (CPU board)
13    *       Disclaimer  :
14    *
15    *       The information described here may contain technical inaccuracies or
16    *       typographical errors. Renesas Technology Corporation and Renesas Solutions
17    *       assume no responsibility for any damage, liability, or other loss rising
18    *       from these inaccuracies or errors.
19    *
20    *       Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
21    *       AND Renesas Solutions Corp. All Rights Reserved
22    *
23    *       History     : Feb.24,2009 Ver.1.00.00
24    *""FILE COMMENT END""*************************************************************/
25    #include <string.h>
26    #include "iodefine.h"     /* This file is automatically generated by the
27                                   High-performance Embedded Workshop. */
28
29    /* ==== Macro declaration ==== */
30    #define SSI_DATASIZE 40u
31    #define SSI_MUTEDATA 0x00000000ul
32
33    /* ==== Prototype declaration ==== */
34    void main(void);
35    void io_init_ssi2(void);
36    void io_init_dma2(void *src, void *dst, size_t size);
37    void io_reload_dma2(void *src, void *dst, size_t size);
38
39    /* ==== Variable declaration ==== */
40    unsigned long Buff[2][SSI_DATASIZE/sizeof(unsigned long)];  /* Receive buffer */
41    unsigned int BuffIdx;             /* Index data indicating the receive buffer currently used */
42    unsigned long RcvData[100];           /* Receive buffer to store all data except the muted data */
43    unsigned int RcvCnt;              /* Number of the receive data stored */
44
```

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

```
45    /*""FUNC COMMENT""*****************************************************************
46     * ID          :
47     * Outline     : main
48     *-------------------------------------------------------------------------------
49     * Include     : #include "iodefine.h"
50     *-------------------------------------------------------------------------------
51     * Declaration : void main(void);
52     *-------------------------------------------------------------------------------
53     * Description : Initializes the SSI, and receives data.
54     *-------------------------------------------------------------------------------
55     * Argument    : void
56     *-------------------------------------------------------------------------------
57     * Return Value: void
58     *""FUNC COMMENT END""*************************************************************/
59    void main(void)
60    {
61      RcvCnt = 0u;                                  /* Number of the receive data */
62      BuffIdx = 0u;                                    /* Receive buffer index */
63      /* ==== Initializes the DMAC/enable the DMA transfer ==== */
64      io_init_dma2(    (void *)&SSIF2.SSIFRDR,              /* Source address */
65                   Buff[BuffIdx],                  /* Destination address */
66                   SSI_DATASIZE);                  /* Number of bytes */
67      io_reload_dma2(  (void *)&SSIF2.SSIFRDR,
68                   Buff[BuffIdx^1u],
69                   SSI_DATASIZE);
70
71      /* ==== Initializes the SSI2 ==== */
72      io_init_ssi2();
73
74      while(1){
75        /* Program end */
76      }
77    }
78
```

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

```
79    /*""FUNC COMMENT""*****************************************************************
80     * ID          :
81     * Outline     : Initializes the SSI
82     *-------------------------------------------------------------------------------
83     * Include     : #include "iodefine.h"
84     *-------------------------------------------------------------------------------
85     * Declaration : void io_init_ssi2(void);
86     *-------------------------------------------------------------------------------
87     * Description : Transfers data in slave receiver mode.
88     *             : The sampling frequency is at 44.1 kHz.
89     *-------------------------------------------------------------------------------
90     * Argument    : void
91     *-------------------------------------------------------------------------------
92     * Return Value: void
93     *""FUNC COMMENT END""*************************************************************/
94    void io_init_ssi2(void)
95    {
96      /* ---- Supplies the clock to the SSI ---- */
97      CPG.STBCR6.BIT.MSTP65 = 0u;         /* SSIF2 */
98
99      /* ----Selects the SSI pin functions ---- */
100     PORT.PFCR0.BIT.PF3MD = 2u;          /* SSISCK2 */
101     PORT.PFCR1.BIT.PF4MD = 2u;          /* SSIWS2 */
102     PORT.PFCR1.BIT.PF5MD = 2u;          /* SSIDATA2 */
103
104     /* ---- Specifies the SSI interrupt level ---- */
105     INTC.IPR15.BIT._SSI2 = 1u;          /* ssi2 */
106
107     /* ---- Sets the Control register (SSICR) ---- */
108     SSIF2.SSICR.LONG = 0x0C0B1520ul;
109       /*
110           bit31       : reserve 0
111           bit30       : CKS : 0--------------------   AUDIO_X1 input (Not used)
112           bit29       : TUIEN : 0------------------   Disables the transmit underflow interrupt
113           bit28       : TOIEN : 0------------------   Disables the transmit overflow interrupt
114           bit27       : RUIEN : 1------------------   Enables the receive underflow interrupt
115           bit26       : ROIEN : 1------------------   Enables the receive overflow interrupt
116           bit25       : IIEN : 0-------------------   Disables the idle mode interrupt
117           bit24       : reserve 0
118           bit23 to 22 : CHNL : B'00----------------   System words have one channel
119           bit21 to 19 : DWL : B'001----------------   Data word length: 16 bits
120           bit18 to 16 : SWL : B'011----------------   System word length: 32 bits
121           bit15       : SCKD : 0-------------------   Serial bit clock input, slave mode
122           bit14       : SWSD : 0-------------------   Serial word WS input, slave mode
123           bit13       : SCKP : 0-------------------   Latches at the rising edge of the SSISCK
124           bit12       : SWSP : 1-------------------   High level at 1st channel,
125                                                      low level at 2nd channel
```

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

```
126           bit11       : SPDP : 0------------------   Padding bits are low level
127           bit10       : SDTA : 1------------------   Transmits and receives in the order of
128                                                      padding bits, and serial data
129           bit9        : PDTA : 0------------------   Transmits and receives lower bits of
130                                                      parallel data
131           bit8        : DEL : 1-------------------   No delay between the SSIWS and SSIDATA
132           bit7 to 4   : CKDV : B'0010-------------   Specifies the oversampling clock as
133                                                      the AUDIO clock/4 (Not used)
134           bit3        : MUEN : 0------------------   Not muted (Not used)
135           bit2        : reserve 0
136           bit1        : TEN : 0-------------------   Disables to transmit data
137           bit0        : REN : 0-------------------   Disables to receive data
138       */
139     /* ---- Sets the FIFO control register (SSIFCR) ---- */
140     SSIF2.SSIFCR.LONG = 0x00000004ul;
141       /*
142           bit31 to 8 : reserve 0
143           bit7 to 6  : TTRG : B'00---------------- Number of transmit data triggers: 7
144           bit5 to 4  : RTRG : B'00---------------- Number of receive data triggers: 1
145           bit3       : TIE : 0-------------------- Disables the transmit data empty
146                                                      interrupt request
147           bit2       : RIE : 1-------------------- Enables the receive data full
148                                                      interrupt request
149           bit1       : TFRST : 0------------------ Disables to reset the transmit FIFO
150                                                      data register
151           bit0       : RFRST : 0------------------ Disables to reset the receive FIFO
152                                                      data register
153       */
154     /* ---- Starts to receive data ---- */
155     SSIF2.SSICR.BIT.REN = 1u;
156
157   }
158   /*""FUNC COMMENT""*****************************************************************
159    * ID          :
160    * Outline     : SSI interrupt
161    *-------------------------------------------------------------------------------
162    * Include     : #include "iodefine.h"
163    *-------------------------------------------------------------------------------
164    * Declaration : void io_int_ssi2(void);
165    *-------------------------------------------------------------------------------
166    * Description : Handles the SSI interrupts.
167    *-------------------------------------------------------------------------------
168    * Argument    : void
169    *-------------------------------------------------------------------------------
170    * Return Value: void
171    *""FUNC COMMENT END""*************************************************************/
172   void io_int_ssi2(void)
173   {
```

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

```
174     /* Receive overflow error */
175     if(SSIF2.SSISR.BIT.RUIRQ == 1u){
176       SSIF2.SSISR.BIT.RUIRQ = 0u;
177       while(1){
178           /* dead loop */
179       }
180     }
181     /* Receive overflow error */
182     if(SSIF2.SSISR.BIT.ROIRQ == 1u){
183       SSIF2.SSISR.BIT.ROIRQ = 0u;
184       while(1){
185           /* dead loop */
186       }
187     }
188   }
189
190   /*""FUNC COMMENT""*****************************************************************
191    * ID          :
192    * Outline     : DMA transfer initialization
193    *-------------------------------------------------------------------------------
194    * Include     : #include "iodefine.h"
195    *-------------------------------------------------------------------------------
196    * Declaration : void io_init_dma2(void *src, void *dst, size_t size);
197    *-------------------------------------------------------------------------------
198    * Description : Transfers the "size" bytes of data from the source address "src" to
199    *             : the destination address "dst" by the DMAC.
200    *             : As it continues to transfer data after the DMA transfer is complete,
201    *             : specify the reload register separately.
202    *             : Enables the DMA transfer end interrupt.
203    *             : Specifies the transfer size in units of longword, and the SSI2 as
204    *             : the transfer source.
205    *             : When the transfer size, and source or destination address alignment does
206    *             : not match, the operation will not be guaranteed.
207    *-------------------------------------------------------------------------------
208    * Argument    : void *src   : source address
209    *             : void *dst   : destination address
210    *             : size_t size : transfer size (in bytes).
211    *-------------------------------------------------------------------------------
212    * Return Value: void
213    *""FUNC COMMENT END""*************************************************************/
214   void io_init_dma2(void *src, void *dst, size_t size)
215   {
216     /* ---- Sets the Standby control register 2 ---- */
217     CPG.STBCR2.BIT.MSTP8 = 0u;             /* DMAC operates */
218
219     /* ---- Disables the DMA transfer ---- */
220     DMAC.CHCR2.BIT.DE = 0u;
221
```

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

```
222     /* ---- Sets the DMA source address register ---- */
223     DMAC.SAR2.LONG = (unsigned long)src;
224
225     /* ---- Sets the DMA destination address register ---- */
226     DMAC.DAR2.LONG = (unsigned long)dst;
227
228     /* ---- Sets the DMA transfer count register ---- */
229     DMAC.DMATCR2.LONG = size >> 2u;
230
231     /* ---- Sets the DMA channel control register ---- */
232     DMAC.CHCR2.LONG = 0x10104814ul;
233       /*
234            bit31      : TC : 0----------------- Transmits data once by one request
235            bit30      : reserve 0
236            bit29      : RLDSAR : 0------------- Disables the SAR reload function
237            bit28      : RLDDAR : 1------------- Enables the DAR reload function
238            bit27      : reserve 0
239            bit26      : DAF : 0---------------- Not used
240            bit25      : SAF : 0---------------- Not used
241            bit24      : reserve 0
242            bit23      : DO : 0----------------- Not used
243            bit22      : TL : 0----------------- Not used
244            bit21      : reserve 0
245            bit20      : TEMASK : 1------------- Continues the DMA transfer when
246                                                  TE bit is set
247            bit19      : HE : 0----------------- Not used
248            bit18      : HIE : 0---------------- Not used
249            bit17      : AM : 0----------------- Not used
250            bit16      : AL : 0----------------- Not used
251            bit15 to 14: DM[1:0] : B'01--------- Increments the destination address
252            bit13 to 12: SM[1:0] : B'00--------- Source address fixed
253            bit11 to 8 : RS[3:0] : B'1000------- Specifies the DMA extension resource selector
254            bit7       : DL : 0----------------- Not used
255            bit6       : DS : 0----------------- Not used
256            bit5       : TB : 0----------------- Specifies the cycle steal mode
257            bit4 to 3  : TS : B'10-------------- Specifies the longword transfer
258            bit2       : IE : 1----------------- Enables an interrupt request
259            bit1       : TE : 0----------------- Transfer end flag
260            bit0       : DE : 0----------------- Disables the DMA transfer
261       */
262     /* ----Sets the DMA extension resource selector 0---- */
263     DMAC.DMARS1.BIT.CH2MID = 0x0Au;         /* MID = SSI2 */
264     DMAC.DMARS1.BIT.CH2RID = 0x03u;         /* RID  */
265
266     /* ----Sets the DMA operation register ---- */
267     DMAC.DMAOR.WORD &= 0xFFF9u;             /* Clears the AE, NMIF bits */
268     DMAC.DMAOR.BIT.DME = 1u;           /* Enables the DMA transfer on all channels */
269
```

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

```
270     /* ---- Sets the interrupt priority level register ---- */
271     INTC.IPR06.BIT._DMAC2 = 1u;
272     /* ---- Enables the DMA transfer ---- */
273
274     DMAC.CHCR2.BIT.DE = 1ul;            /* Enables the DMA transfer */
275   }
276
277   /*""FUNC COMMENT""*****************************************************************
278    * ID          :
279    * Outline     : DMA transfer reload setting
280    *-------------------------------------------------------------------------------
281    * Include     : #include "iodefine.h"
282    *-------------------------------------------------------------------------------
283    * Declaration : void io_reload_dma2(void *src, void *dst, size_t size);
284    *-------------------------------------------------------------------------------
285    * Description : Specifies values in the reload source address register, reload destination
286    *             : address register, and reload transfer count register.
287    *             : Specify the transfer size in units of longword.
288    *             : When the transfer size, and source or destination address alignment
289    *             : does not match, the operation will not be guaranteed.
290    *-------------------------------------------------------------------------------
291    * Argument    : void *src   : source address
292    *             : void *dst   : destination address
293    *             : size_t size : transfer size (in bytes).
294    *-------------------------------------------------------------------------------
295    * Return Value: void
296    *""FUNC COMMENT END""*************************************************************/
297   void io_reload_dma2(void *src, void *dst, size_t size)
298   {
299     /* ---- Sets the DMA reload source address register ---- */
300     DMAC.RSAR2.LONG= (unsigned long)src;
301
302     /* ---- Sets the DMA reload destination address register ---- */
303     DMAC.RDAR2.LONG= (unsigned long)dst;
304
305     /* ---- Sets the DMA reload transfer count register ---- */
306     DMAC.RDMATCR2.LONG= size >> 2u;
307
308   }
309
```

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

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : DMA transfer end interrupt
 *-----------------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration : void io_int_dma2(void);
 *-----------------------------------------------------------------------------
 * Description : Reads all the receive data except the muted data from the receive buffer.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void io_int_dma2(void)
{
  volatile unsigned long dummy;
  unsigned long rdata;
  int i;

  /* ---- Updates the reload register ---- */
  io_reload_dma2((void *)&SSIF2.SSIFRDR, Buff[BuffIdx], SSI_DATASIZE);

  /* ---- Reads the receive data ---- */
  for(i=0; i<SSI_DATASIZE/sizeof(unsigned long); i++){
    rdata = Buff[BuffIdx][i];

    if(SSI_MUTEDATA != rdata){
        RcvData[RcvCnt++] = rdata;
        if( RcvCnt >= (sizeof(RcvData)/sizeof(unsigned long)) ){
            RcvCnt = 0u;
        }
    }
  }
  /* ---- Updates the index of the receive buffer ---- */
  BuffIdx ^= 1u;

  /* ---- Clears the transfer end flag ---- */
  DMAC.CHCR2.BIT.TE = 0u;
  dummy = DMAC.CHCR2.BIT.TE;      /* Dummy read */
}
/* End of File */
```

4. References

- Software Manual
  SH-2A/SH-2A-FPU Software Manual Rev. 3.00
  (Download the latest version from the Renesas website.)

- Hardware Manual
  SH7262 Group, SH7264 Group Hardware Manual Rev. 1.00
  (Download the latest version from the Renesas website.)

Website and Support

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

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

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Apr 14, 2009	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.