SH7216 Group

Transmission by the I²C Bus Interface 3 Module in Master Operation

REJ06B0904-0101
Rev.1.01
Jun 25, 2010

Introduction

This application note describes transmission by the I²C bus interface 3 module (IIC3) of the SH7216 in the case of a single master on the I²C bus.

Target Device

SH7216

Contents

1. Preface 2
2. Overview 3
3. Listing of the Sample Program 12
4. Documents for Reference 18

1. Preface

1.1 Specifications

- Data are transferred from a master transmitter to a slave receiver, with SH7216 MCUs serving as both master and slave devices.
- The transfer rate is set at 403 kHz.

Note: Two SH7216 CPU boards (R0K572167) are used to run this sample program, and code included in the SH7216 Application Note *Reception by the I²C Bus Interface 3 Module in Slave Operation* (REJ06B0905) is used for the slave receiver software.

1.2 Module Used

- I²C bus interface 3 (IIC3)
- Clock pulse generator (CPG)
- Power-Down Modes
- Pin function controller (PFC)

1.3 Applicable Conditions

- MCU SH7216
- Operating frequency Internal clock: 200 MHz
  Bus clock: 100 MHz
  Peripheral clock: 50 MHz
- Integrated development environment
  High-performance Embedded Workshop Ver.4.05.01
  (from Renesas Electronics Corp.)
- C compiler SuperH RISC Engine Family C/C++ Compiler Package Ver.9.03.00 Release01
  from Renesas Electronics
- Compiler options Default settings of the High-performance Embedded Workshop
  (-cpu=sh2afpu -pic=1 -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)
- Slave receive module SH7216 CPU board (R0K572167)

1.4 Related Application Note

- SH7216 Group Application Note: Reception by the I²C Bus Interface 3 Module in Slave Operation (REJ06B0905)

2. Overview

In this sample program, the SH7216 (master device) transfers data to the SH7216 (slave device) by using I²C bus interface 3 (IIC3) module.

2.1 Operational Overview of Module Used

The I²C bus interface 3 (IIC3) module conforms to and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs in some respects from the register configuration implemented by Philips.

The features of the SH7216's I²C bus interface 3 (IIC3) are described below.

- I²C bus format and clock-synchronous serial format are selectable.
- Continuous transmission/reception
  Since the shift register, transmit data register, and receive data register are independent of each other, continuous data transfer is possible.

Table 1 is a list of the features of the available formats, and figure 1 shows a block diagram of the IIC3 module.

Table 1 Features of the Formats

Format	Features
I²C bus format	• Start and stop conditions are generated automatically in master mode. • Acknowledge output levels are selectable in data reception. • Acknowledge bit is automatically loaded in data transmission • On-chip bit synchronization/wait function In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed. • Six interrupt sources 1. Transmit data empty (including slave-address match) 2. Transmit end 3. Receive data full (including slave-address match) 4. Arbitration lost 5. NACK detection 6. Stop condition detection • Data transfer by the direct memory access controller (DMAC) can be activated by a transmit-data-empty or receive-data-full interrupt request. • Direct bus drive Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.
Clock-synchronous serial format	• Four interrupt sources 1. Transmit-data-empty 2. Transmit-end 3. Receive-data-full 4. Overrun error • Data transfer by the direct memory access controller (DMAC) can be activated by a transmit-data-empty or receive-data-full interrupt request.

Note: For details on IIC3, see the section on I²C Bus Interface 3 (IIC3) of the *SH7216 Group Hardware Manual (REJ09B0543)*.

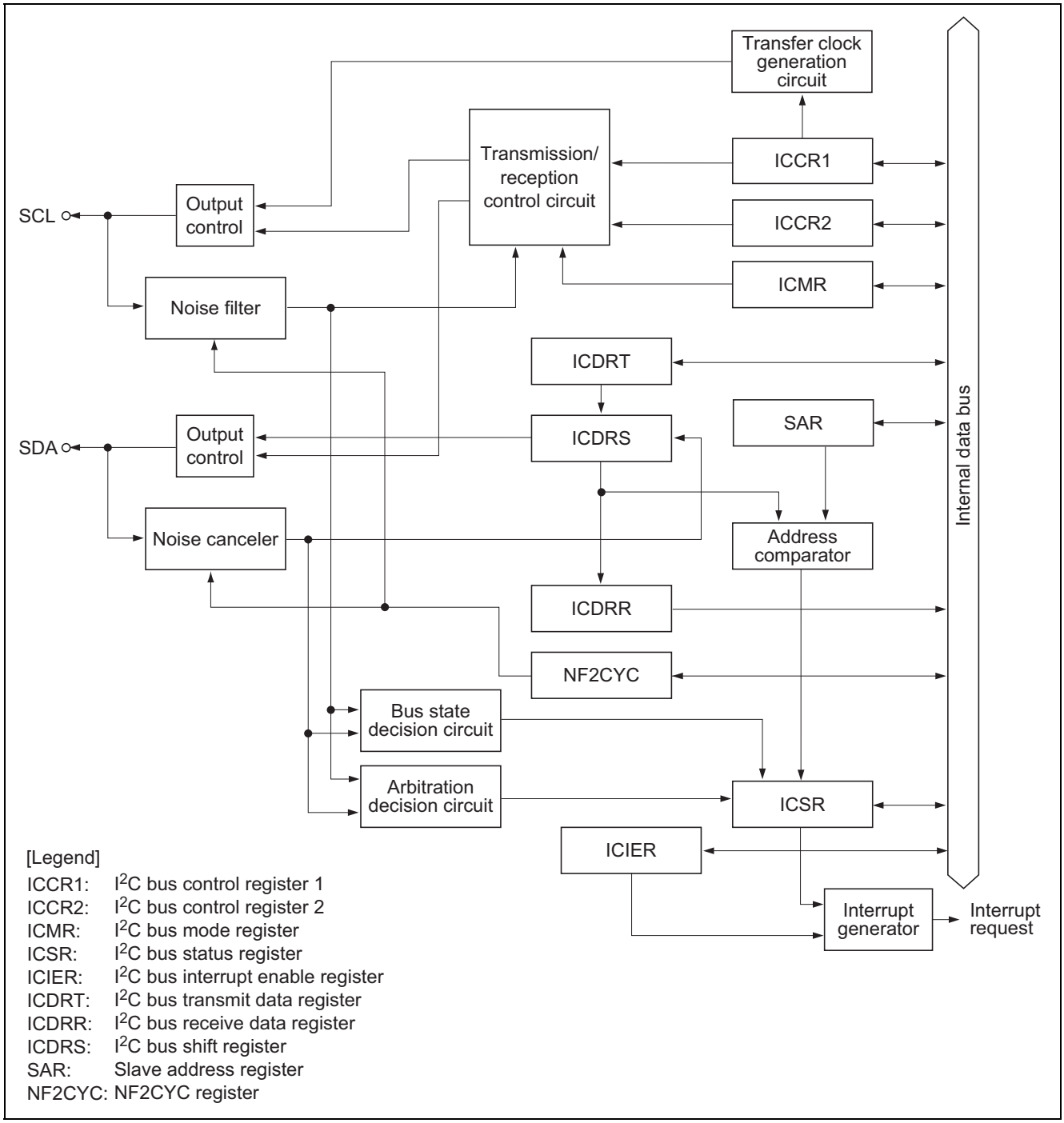

Figure 1 Overview of I²C Bus Interface 3

2.2 Procedure for Setting Module Used

This section describes the procedure for making initial settings for IIC3. The transfer rate must be set to meet the external specification. In this sample program, Pϕ/124 is specified as the transfer rate. Figure 2 shows an example of the initialization sequence for IIC3. For details on the settings of individual registers, see the *SH7216 Group Hardware Manual (REJ09B0543)*.

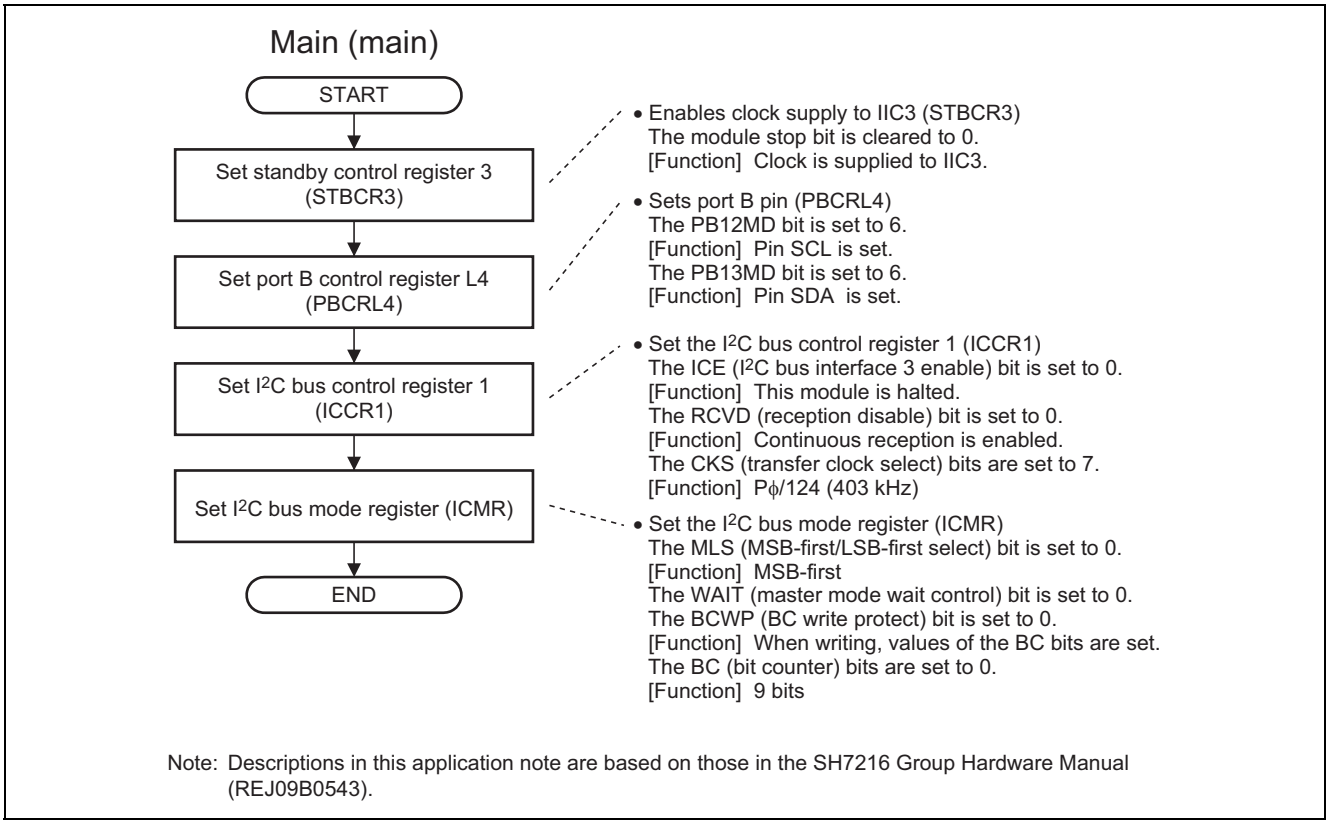

Figure 2 Example of the Initialization Sequence for I²C Bus Interface 3

2.3 Operation of the Sample Program

In this sample program, IIC3 is placed in master transmit mode, and transmits 10 bytes for writing to a page within the SH7216.

The device code b'1010 and the device address b'000 are used.

Figure 3 shows the operations for writing to a single page, and figure 4 shows the operating environment of this sample program.

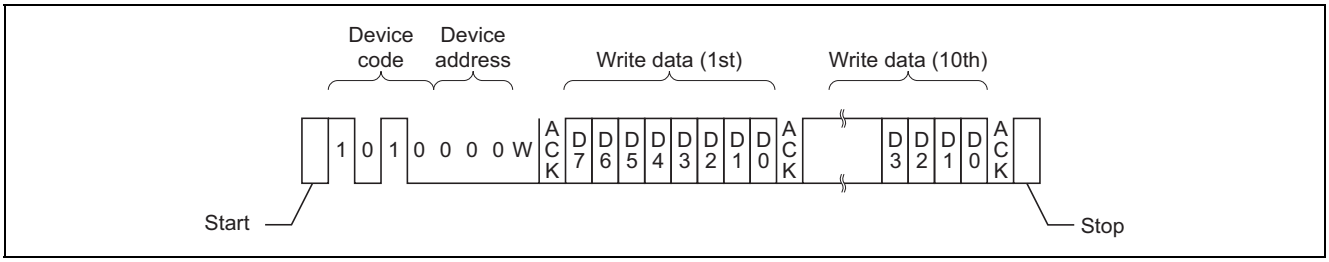

Figure 3 Page Write Operation

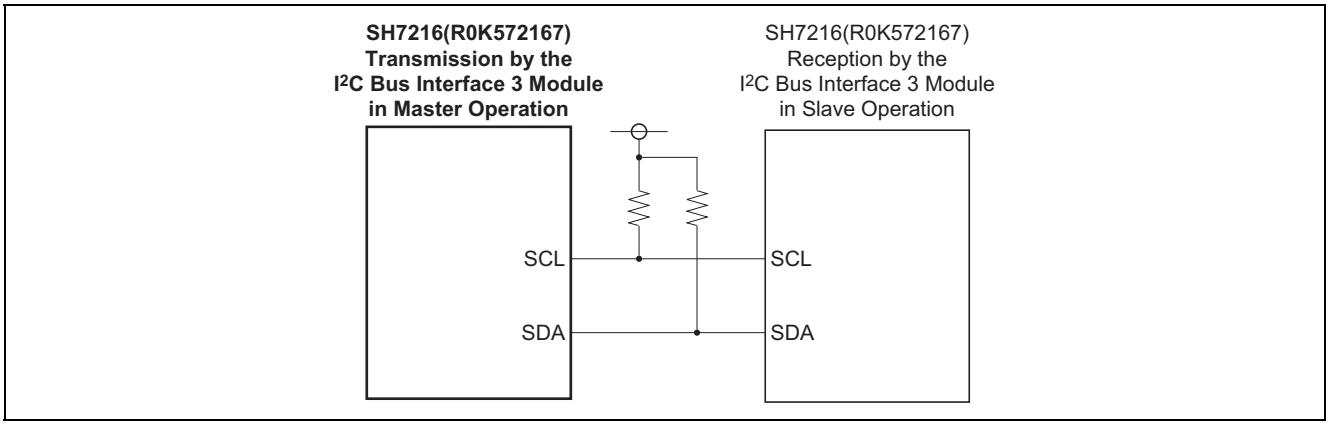

Figure 4 Operating Environment of the Sample Program

2.4 Sequence of Processing by the Sample Program

Table 2 gives the register settings in the sample program. Table 3 shows macro definitions in the sample program. Figures 5 to 8 show the flow of processing by the sample program.

Table 2 Register Settings Used in Sample Program

Register Name	Address	Setting Value	Description
Standby control register 3 (STBCR3)	H'FFFE0408	H'00	MSTP33 = "0": IIC3 operates.
I²C bus control register 1 (ICCR1)	H'FFFEE000	H'B7	ICE = "1": SCL and SDA pins are placed in bus-drive state. RCVD = "0": Continuous reception is enabled. MST = "1", TRS = "1": Master transmit mode CKS = "B'0111": transfer rate Pϕ/124
I²C bus mode register (ICMR)	H'FFFEE002	H'30	MLS = "0": MSB-first BCWP = "0": Allows the writing of values to the BC bits. BC = "B'000": 9 bits

Table 3 Macro Definitions Used in Sample Program

Macro Definition	Setting Value	Description
DEVICE_CODE	H'A0	Device code
DEVICE_ADDR	H'00	Device address
IIC_DATA_WR	H'00	Write code
IIC_DATA_RD	H'01	Read code
IIC3_DATA	10	Data transfer size

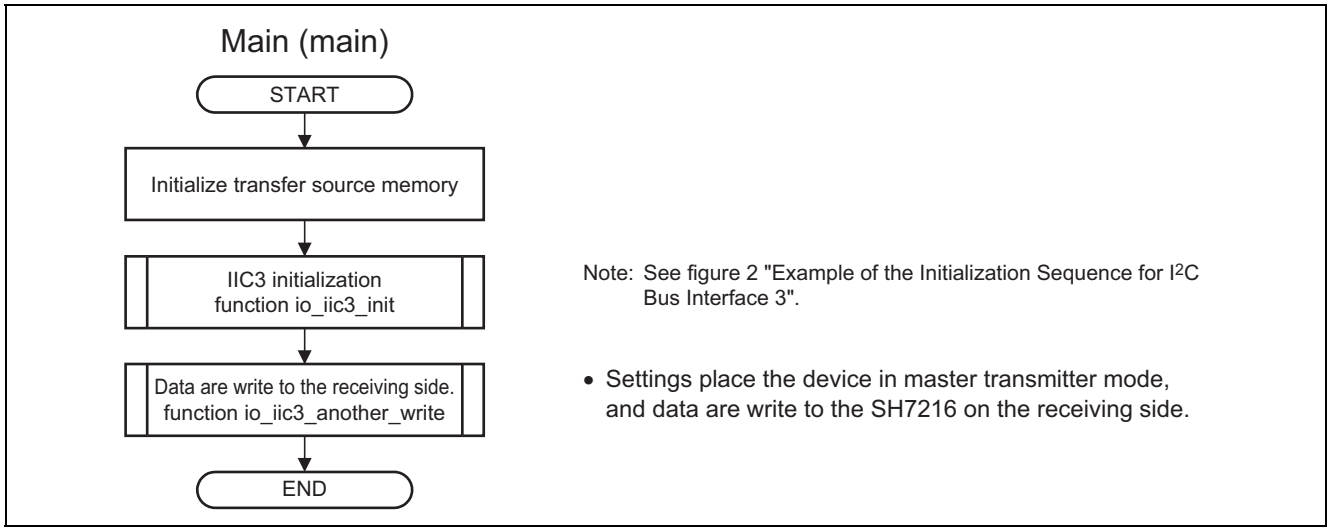

Figure 5 Flow of Processing by the Sample Program (1)

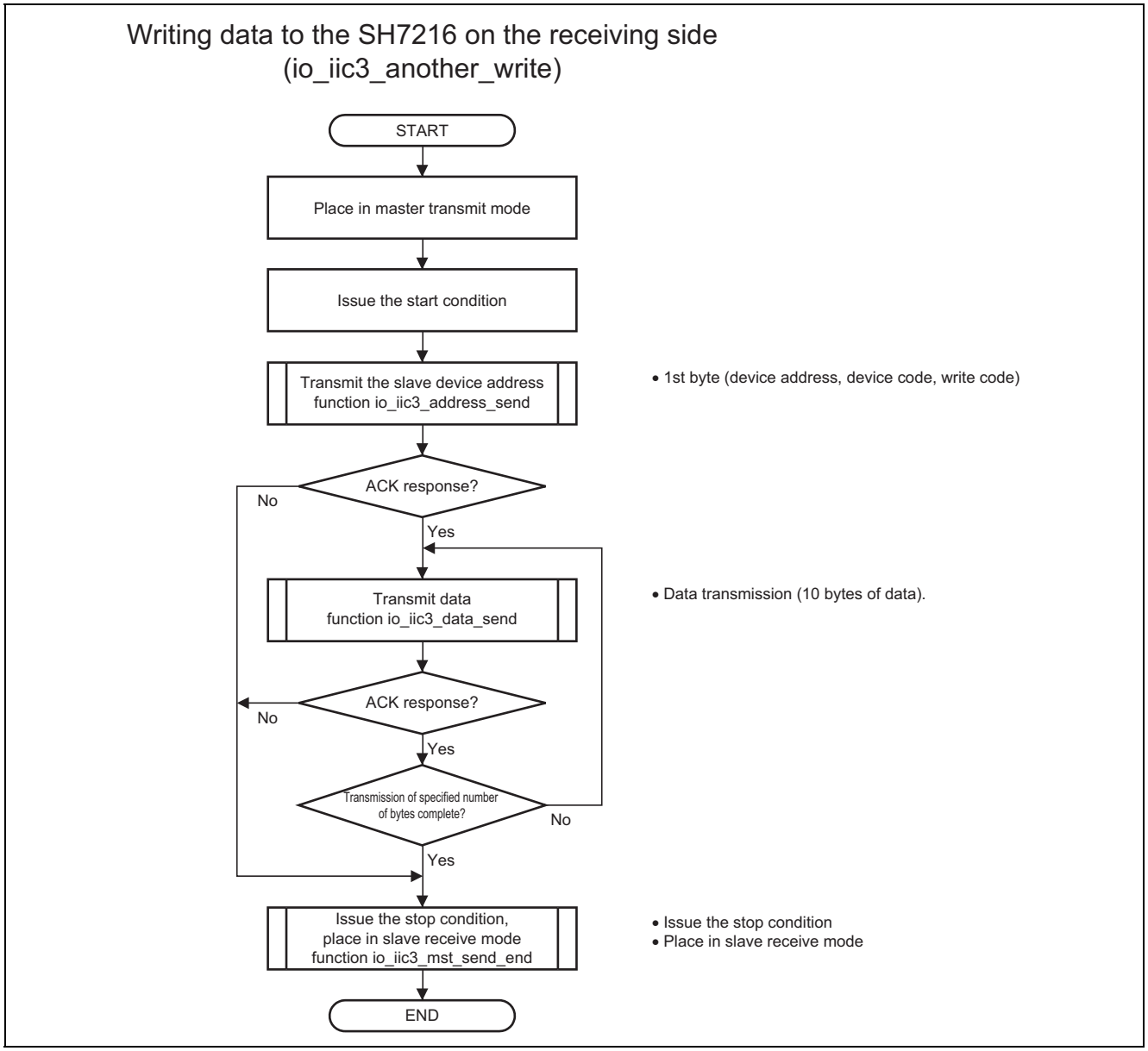

Figure 6 Flow of Processing by the Sample Program (2)

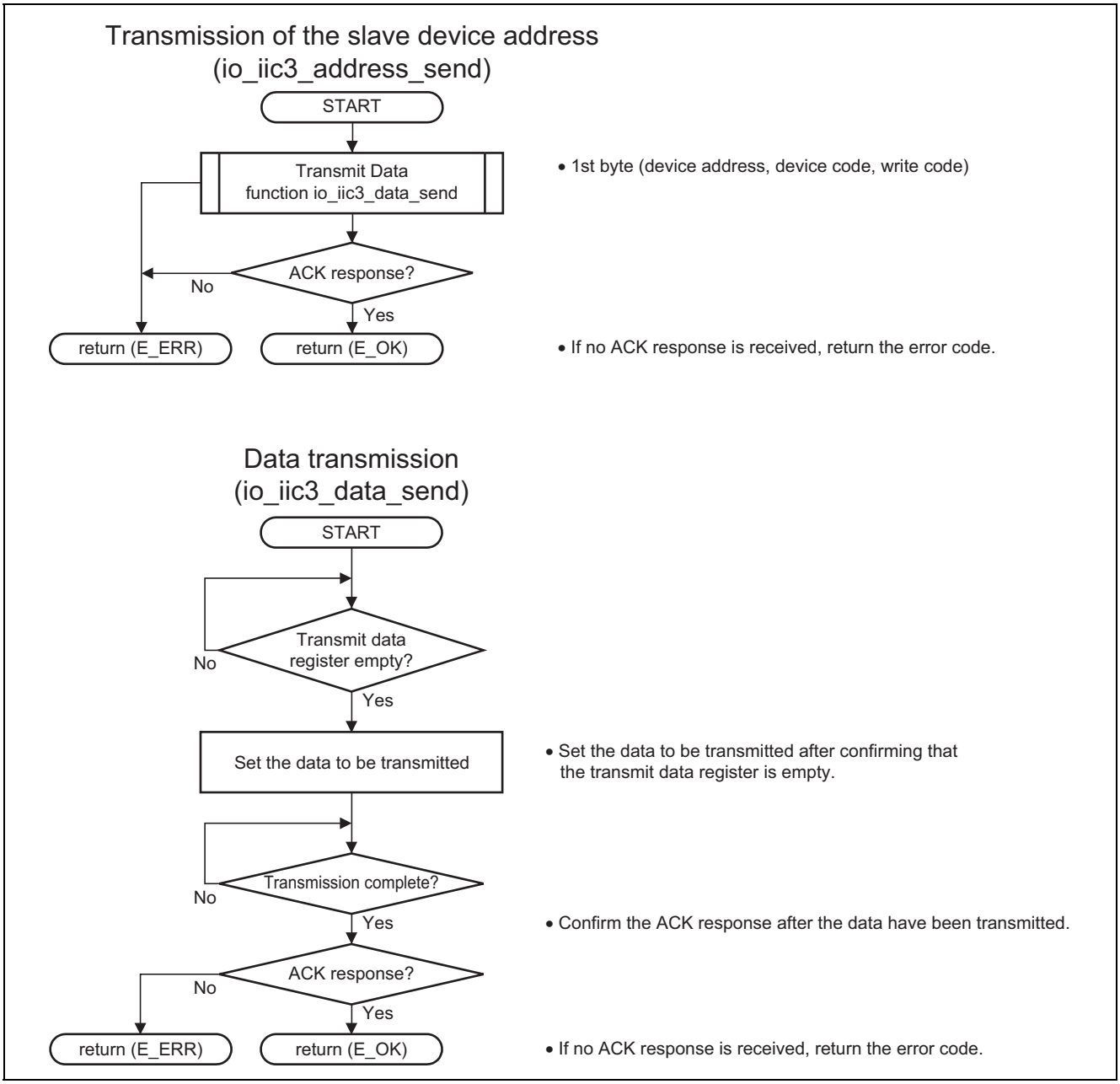

Figure 7 Flow of Processing by the Sample Program (3)

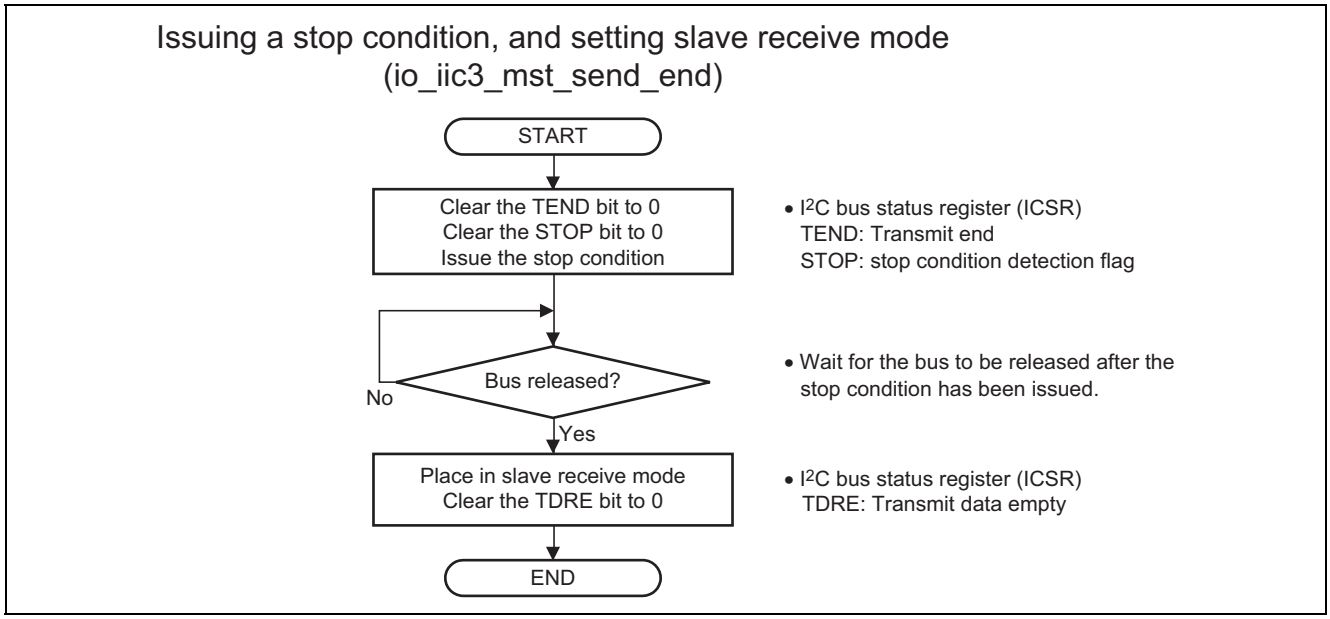

Figure 8 Flow of Processing by the Sample Program (4)

3. Listing of the Sample Program

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Technology Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Technology Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
*   Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
*""FILE COMMENT""************ Technical reference data *************************
*   System Name : SH7216 Sample Program
*   File Name   : main.c
*   Abstract    : Sample Program of Compare Match Timer
*   Version     : 1.00.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.05.01).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.03 Release01).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description :
*   note        : This program sends data to the SH7211 base on the
*   : slave-receive side by using the master-send mode of IIC3.
***********************************************************************
*   History     : Jun.8,2009 Ver.1.00.00
*""FILE COMMENT END""*****************************************************/
#include <machine.h>
#include "iodefine.h"       /* SH7216 iodefine  */

/* ==== symbol definition ==== */
#define DEVICE_CODE 0xA0        /* another device code    : b'1010     */
#define DEVICE_ADDR 0x00        /* another device address : b'000      */
#define IIC_DATA_WR 0x00        /* Data write code       : b'0        */
#define IIC_DATA_RD 0x01        /* Data read code        : b'1        */
#define IIC3_DATA 10

#define E_OK 0
#define E_ERR -1

/* ==== RAM allocation variable declaration ==== */
unsigned char WriteData[IIC3_DATA];
```

Figure 9 Sample Program Listing: main.c (1)

```
/* ==== prototype declaration ==== */
void main(void);
int io_iic3_another_write(unsigned char d_code, unsigned char d_adr,
                          unsigned int w_size, unsigned char *w_buf);
int io_iic3_data_send(unsigned char data);
int io_iic3_address_send(unsigned char *data);
void io_iic3_mst_send_end(void);
int io_iic3_init(void);

void io_led_ini(void);
void io_init_cmt0(void);

/*""FUNC COMMENT""*****************************************************************
 * Outline      : sumple program main
 *---------------------------------------------------------------------------------
 * Include      :
 *---------------------------------------------------------------------------------
 * Declaration  : void main(void);
 *---------------------------------------------------------------------------------
 * Function     : sumple program main
 *---------------------------------------------------------------------------------
 * Argument     : void
 *---------------------------------------------------------------------------------
 * Return Value : none
 *---------------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""************************************************************/
void main(void)
{
    int i;

    /* ==== Writing data creation ==== */
    for(i = 0; i < IIC3_DATA; i++){
        WriteData[i] = IIC3_DATA+i;
    }

    /* ==== IIC3 initialization ==== */
    io_iic3_init();
    /* ==== IIC3 master-send mode ==== */
    io_iic3_another_write(DEVICE_CODE,           /* Device code */
                          DEVICE_ADDR,          /* Device address */
                          sizeof(WriteData),    /* Writing data's size */
                          WriteData);           /* Store the writing data */
    while(1){
        /* Loop */
    }
}
```

Figure 10 Sample Program Listing: main.c (2)

```
/*""FUNC COMMENT""*************************************************************
 * Outline      : IIC3-module initial setting
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration  : int io_iic3_init(void);
 *-----------------------------------------------------------------------------
 * Function     : IIC3-module initial setting
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value : E_OK
 *-----------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""*********************************************************/
int io_iic3_init(void)
{
    STB.CR3.BIT._IIC3 = 0x00;       /* Clear the IIC3 module standby mode */

    /* ---- PFC setting ---- */
    PFC.PBCRL4.BIT.PB12MD = 0x06;    /* SCL selection  */
    PFC.PBCRL4.BIT.PB13MD = 0x06;    /* SDA selection */
PFC.PBPCRL.BIT.PB12PCR = 1;   /* SCL pull-up  */
PFC.PBPCRL.BIT.PB13PCR = 1;   /* SDL pull-up  */
    /* ----IIC31 module operation disabled ---- */
    IIC3.ICCR1.BIT.ICE = 0x00;      /* IIC3 transfer disabled state */
    IIC3.ICCR1.BIT.ICE = 0x01;      /* IIC3 transfer enabled */
    IIC3.ICCR1.BIT.RCVD = 0x00;     /* Continue the next receive data */
    IIC3.ICCR1.BIT.CKS = 0x07;      /* Transfer clock rate : Pϕ/124(403kHz) */
    /*  ---IIC bus mode register (ICMR) setting --- */
    IIC3.ICMR.BYTE = 0x30;
                /*
                   bit7   : MLS:0 ---------------------  MSB first
                   bit6   : WAIT:0 --------------------  No WAIT insertion
                   bit5-4 : Reserve:1 -----------------  Reserve bits
                   bit3   : BCWP:0 --------------------  Unsetting
                   bit2-0 : BC2:0, BC1:0, BC0:0 -------  IIC bus format: 9 bits
                */

    return(E_OK);
}
```

Figure 11 Sample Program Listing: main.c (3)

```
/*""FUNC COMMENT""*************************************************************
 * Outline      : Write data for slave-side
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration  : int io_iic3_mst_send(unsigned char d_code,
 *              :                      unsigned char d_adr,
 *              :                      unsigned int w_size,
 *              :                      unsigned char *w_buf);
 *-----------------------------------------------------------------------------
 * Function     : This master-send mode choose slave-side is specified
 *  : by d_code & d_adr.
 *              : Sending data to be written in slave-side is specified by w_buf,
 *  : and the data is "w_size"BYTE.
 *              : (DEVICE_CODE = d_code, DEVICE_ADDR = d_adr,
 *  : sizeof(WriteData) = w_size, WriteData = *w_buf)
 *-----------------------------------------------------------------------------
 * Argument     : unsigned char d_code ; Device code
 *              : unsigned char d_adr  ; Device address
 *              : unsigned int w_size  ; Writing data size
 *              : unsigned char* w_buf ; Store the writing data
 *-----------------------------------------------------------------------------
 * Return Value : E_OK  ; ACK response
 *              : E_ERR ; No ACK response
 *-----------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""*************************************************************/
int io_iic3_another_write(unsigned char d_code, unsigned char d_adr,
                          unsigned int w_size, unsigned char *w_buf)
{
    int ack = E_OK;
    int i;
    unsigned char send[1];

    send[0] = (unsigned char)(d_code | ((d_adr & 0x7) << 1) | IIC_DATA_WR);

    while(IIC3.ICCR2.BIT.BBSY == 0x01){
        /* It weits untill the bas is opened */
    }
  /* It sets it to the master send mode */
    IIC3.ICCR1.BYTE |= 0x30;
  /* Start condition is issued */
    IIC3.ICCR2.BYTE = ((IIC3.ICCR2.BYTE & 0xbf) | 0x80);

    ack = io_iic3_address_send(send);                       /* 1st BYTE send */

    if(ack == E_OK){
        /* When there is ack reaction from the specified device */
        for(i = 0; i < w_size; i++){
            ack = io_iic3_data_send(*w_buf++);              /* Data send */
            if(ack == E_ERR){
                break;
            }
        }
     io_iic3_mst_send_end();
    }
    else{
        /* When there is ack reaction from the specified device */
        io_iic3_mst_send_end();
    }
    return(ack);
}
```

Figure 12 Sample Program Listing: main.c (4)

```
/*""FUNC COMMENT""*************************************************************
 * Outline      : Slave device address send
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration  : int io_iic3_address_send(unsigned char *data);
 *-----------------------------------------------------------------------------
 * Function     : The address of the slave device specified with
 *  : data (1byte) is sended.
 *-----------------------------------------------------------------------------
 * Argument     : unsigned char *data ; Sending data
 *-----------------------------------------------------------------------------
 * Return Value : E_OK  ; ACK response
 *              : E_ERR ; No ACK response
 *-----------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""*********************************************************/
int io_iic3_address_send(unsigned char *data)
{
    int ack;

    ack = io_iic3_data_send(*data++);   /* Address of slave device */
    if(ack == E_ERR){
        return(ack);
    }
    return(ack);
}
/*""FUNC COMMENT""*************************************************************
 * Outline      : 1byte data sending
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration  : int io_iic3_data_send(unsigned char data);
 *-----------------------------------------------------------------------------
 * Function     : Data sending. The processing procedure is as follows.
 *              :  1. ICDRT waits for even empty
 *              :  2. Sending data set
 *              :  3. Data sending completion confirmation
 *              :  4. ACK response confirmation
 *-----------------------------------------------------------------------------
 * Argument     : unsigned char data ; Sending data
 *-----------------------------------------------------------------------------
 * Return Value : E_OK  ; ACK response
 *              : E_ERR ; No ACK response
 *-----------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""*********************************************************/
int io_iic3_data_send(unsigned char data)
{
    int ack;
    while(IIC3.ICSR.BIT.TDRE == 0x0){
        /* 1. ICDRT waits for even empty */
    }
    IIC3.ICDRT = data;        /* 2. Sending data set */
    while(IIC3.ICSR.BIT.TEND == 0x00){
        /* 3. Data sending completion confirmation */
    }
  /* 4. ACK response confirmation */
    if(IIC3.ICIER.BIT.ACKBR == 0){
        ack = E_OK;
    }
    else{
        ack = E_ERR;
    }
    return(ack);
}
```

Figure 13 Sample Program Listing: main.c (5)

```
/*""FUNC COMMENT""*************************************************************
 * Outline      : Stop condition issue
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration  : void io_iic3_mst_send_end(void);
 *-----------------------------------------------------------------------------
 * Function     : The conditions stop is issued, and it switches to
 *  : the slave-receive mode.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value : none
 *-----------------------------------------------------------------------------
 * Notice       :
 *""FUNC COMMENT END""*********************************************************/
void io_iic3_mst_send_end(void)
{
    IIC3.ICSR.BIT.TEND = 0x00;      /* TEND Flag clear */
    IIC3.ICSR.BIT.STOP = 0x00;      /* STOP Flag clear */
    IIC3.ICCR2.BYTE &= 0x3f;        /* The conditions stop is issued */

    while(IIC3.ICSR.BIT.STOP == 0x00){
        /* It waits untill the bas is opened */
    }

    IIC3.ICCR1.BYTE &= 0xcf;        /* It switches to the slave-receive mode */
    IIC3.ICSR.BIT.TDRE = 0x00;      /* TDRE clear */
}
/* End of File */
```

Figure 14 Sample Program Listing: main.c (6)

4. Documents for Reference

- Software Manual
  SH-2A, SH2A-FPU Software Manual (REJ09B0051)
  The most up-to-date version of this document is available on the Renesas Electronics Website.

- Hardware Manual
  SH7216 Group Hardware Manual (REJ09B0543)
  The most up-to-date version of this document is available on the Renesas Electronics 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	Jul.28.09	—	First edition issued
1.01	Jun.25.10	—	Modifications to source project due to change in FRQCR setting method

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: +49-211-65030, 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
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