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

I²C Bus Interface 3 Transmission in Single-Master Mode (Write in EEPROM)

Summary

This application note describes an example of writing data in EEPROM by the SH7262/SH7264 Microcomputers (MCUs) I²C Bus Interface 3 (IIC3) transmission in single-master mode.

Target Device

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

Contents

1. Introduction

1.1 Specifications

- Specifies the SH7264 MCU as the master device, and EEPROM as the slave device to write data in EEPROM.
- Transfer rate is set to 391 kHz.

Note: Set the transfer rate to satisfy the specifications of EEPROM.

1.2 Modules Used

- I²C Bus Interface 3 (IIC3)

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)
EEPROM	HX58X24128FPIE (128 Kbit)

1.4 Related Application Note

Refer to the related application notes as follows:

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group I²C Bus Interface 3 Reception in Single-Master Mode (Read from EEPROM)

2. Applications

The SH7264 MCU (the master device) transfers data to an EEPROM (the slave device) using the IIC3 in the sample program.

2.1 IIC3 Operation

IIC3 is compliant to the I²C bus (Inter IC Bus) interface specifications invented by Philips and supports subsets. However, the configuration of registers to control the I²C bus partly differs from that of Philips.

The SH7264 IIC3 has the following features:

- Format options selectable, I²C bus format or clocked synchronous serial format
- Transmits or receives data continuously
  As the shift register, transmit data register and receive data register are separate registers, the IIC3 can transmit and receive data continuously.

The table below lists the features of two options of formats. Figure 1 shows the IIC3 block diagram. For details on the IIC3, refer to Section 17, I²C Bus Interface 3 in SH7262 Group, SH7264 Group Hardware Manual.

Table 1 Format Features

Format Name	Description
I²C Bus Format	• Automatically generates the START and STOP conditions in master mode • An output level of an ACK can be selected upon reception • Automatically loads an ACK bit upon transmission • Includes the bit synchronization/wait function The IIC3 monitors the SCL status per bit in master mode to synchronize automatically. When it is not ready for transfer, it specifies the SCL to low level to wait. • Six interrupt sources (1) Transmit data empty (including when slave address match) (2) Transmit end (3) Receive data full (including when slave address match) (4) Arbitration lost (5) NACK detection (6) Stop condition detection • Using the transmit data empty interrupt and the receive data full interrupt to activate the Direct Memory Access Controller (DMAC) and transfer data • Bus can be driven directly The SCL and SDA pins are driven by an NMOS open-drain output when selecting the bus drive function
Clocked synchronous serial format	• Four interrupt sources (1) Transmit data empty (2) Transmit end (3) Receive data full (4) Overrun error • Using the transmit data empty interrupt and the receive data full interrupt to activate the Direct Memory Access Controller (DMAC) and transfer data

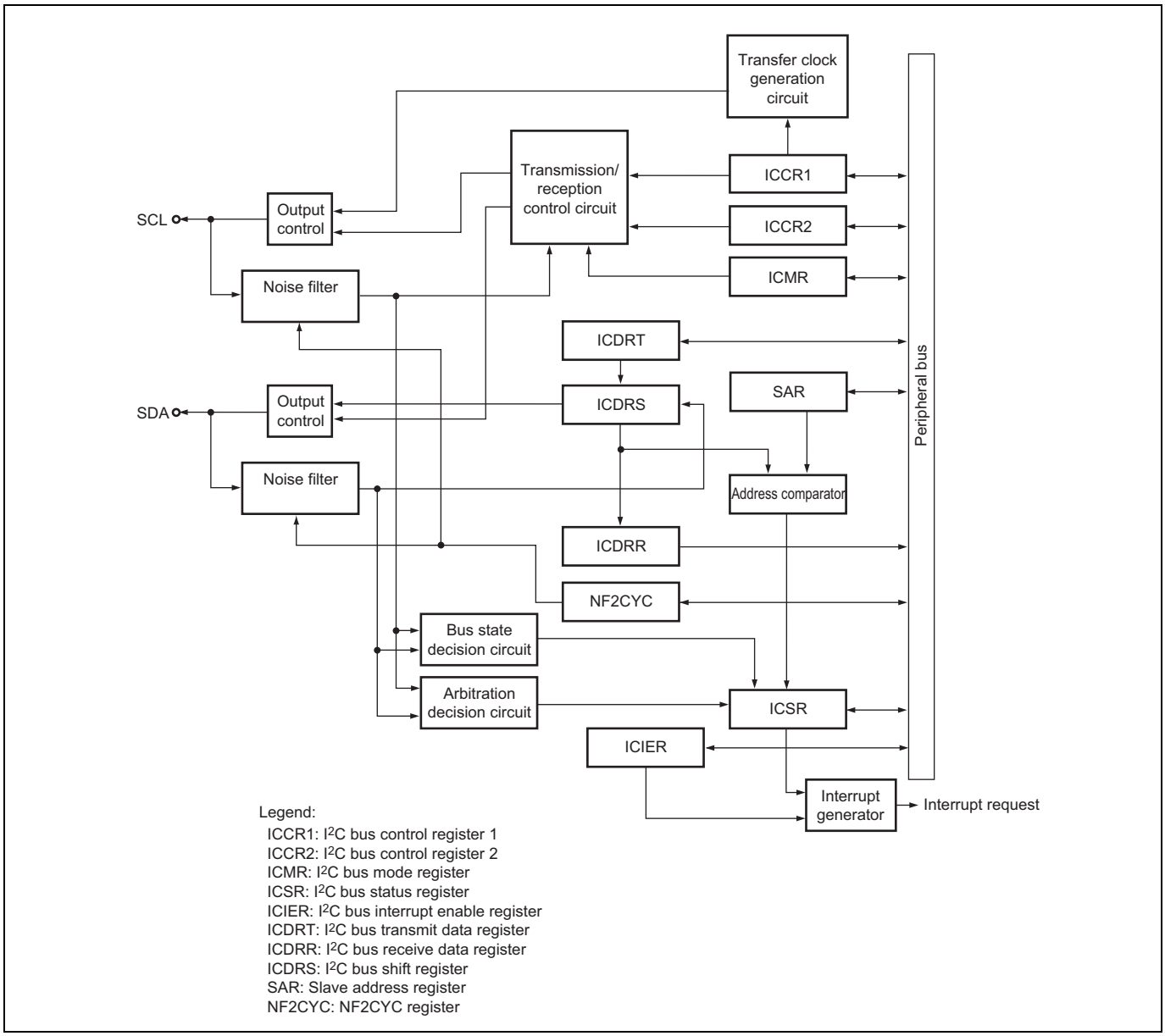

Figure 1 IIC3 Block Diagram

2.2 IIC3 Setting Procedure

This section describes how to set up the IIC3. Make sure to specify the transfer rate to satisfy the SH7264 MCU external specifications. Pφ/92 is specified in the sample program. The figure below shows the flow chart of the IIC3 setup example. For details on register settings, refer to the SH7262 Group, SH7264 Group Hardware Manual.

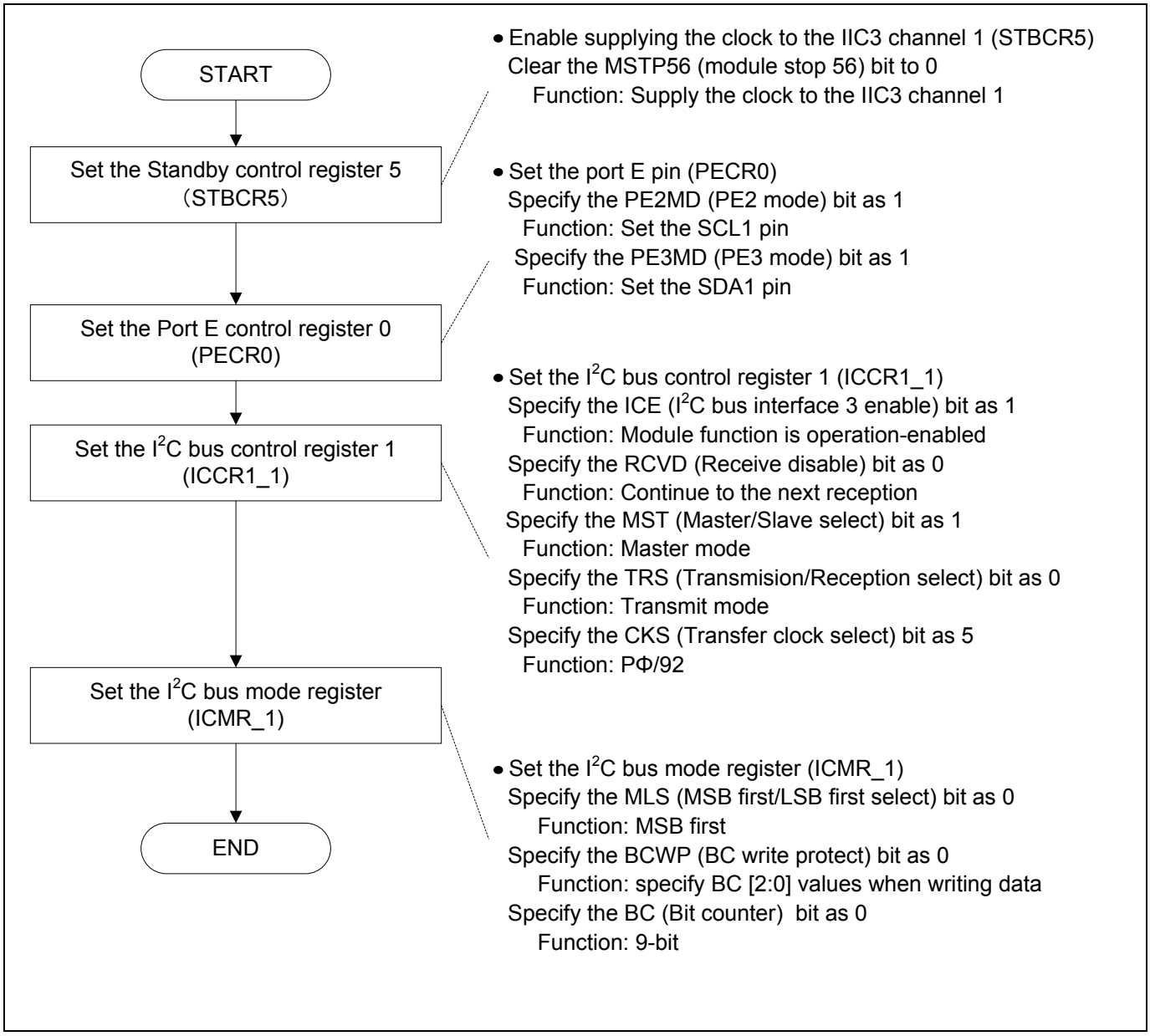

Figure 2 IIC3 Setup Flow Chart

2.3 Sample Program Operation

The sample program specifies the IIC3 in master transmit mode to write 10 bytes of data (sequential read). This sample program uses "B'1010" as the device code and "B'000" as the device address. For device codes and the device address, refer to the EEPROM data sheet provided by the manufacturer.

The memory address indicates the write start address on EEPROM, and the address is incremented when writing in EEPROM. The figure below shows the IIC3 page write operation. Figure 4 shows the operating environment of the sample program.

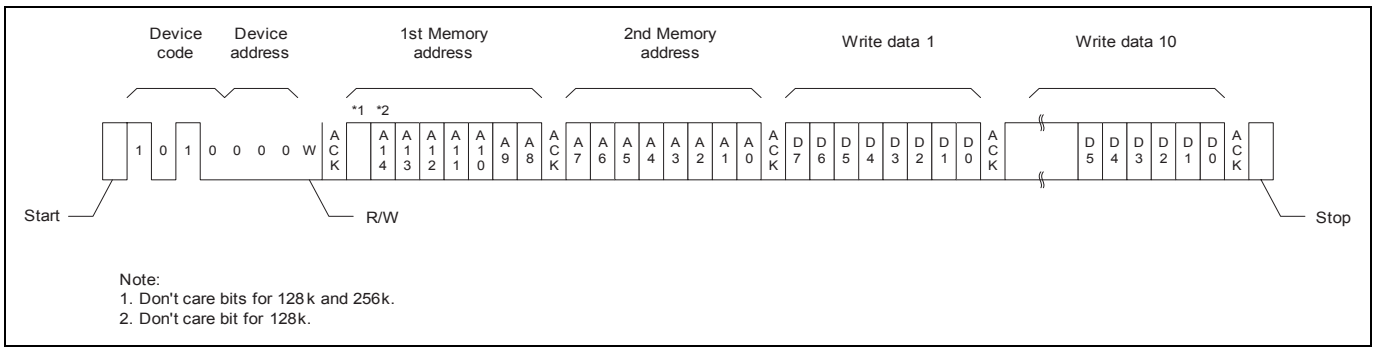

Figure 3 Page Write Operation

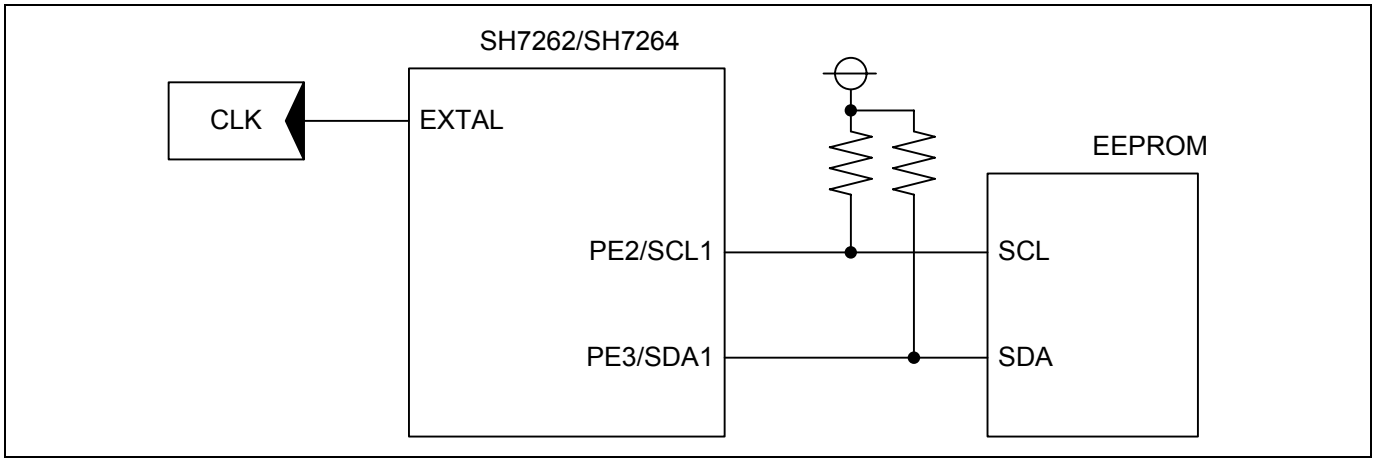

Figure 4 Sample Program Operating Environment

2.4 Sample Program Flow

Table 2 lists the register settings in the sample program. Table 3 lists the macro definitions used in the sample program. Figure 5 to Figure 9 show flow charts of the sample program.

Table 2 Register Settings (Default Setting)

Register Name	Address	Setting	Function
Standby control register 5 (STBCR5)	H'FFFE 0410	H'00	MSTP56 = "0": IIC3 channel 1 is operating
I²C bus control register 1 (ICCR1_1)	H'FFFE E400	H'B5	ICE = "1": SCL/SDA pins are driven by bus RCVD = "0": Following reception enabled MST = "1", TRS = "0": Master transmit mode CKS = "B'0101": transfer rate is PΦ/92
I²C bus mode register (ICMR_1)	H'FFFE E402	H'30	MLS = "0": MSB first BCWP = "0": Set the BC value when writing BC = "B'000": 9 bits

Table 3 Macro Definitions

Macro Definitions	Setting	Function
EEPROM_MEM_ADDR	H'0000	EEPROM start address
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

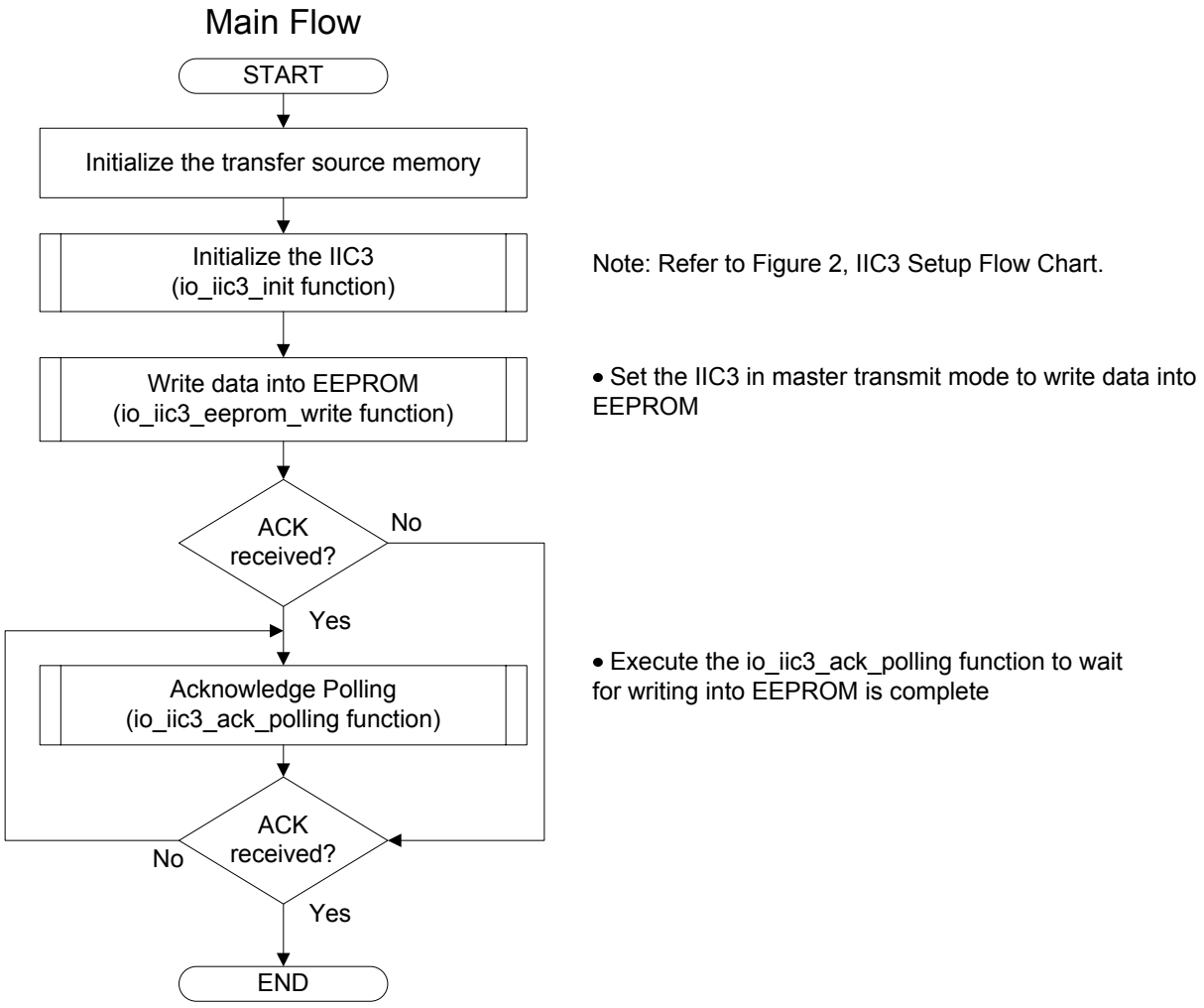

Figure 5 Sample Program Flow Chart (1/5)

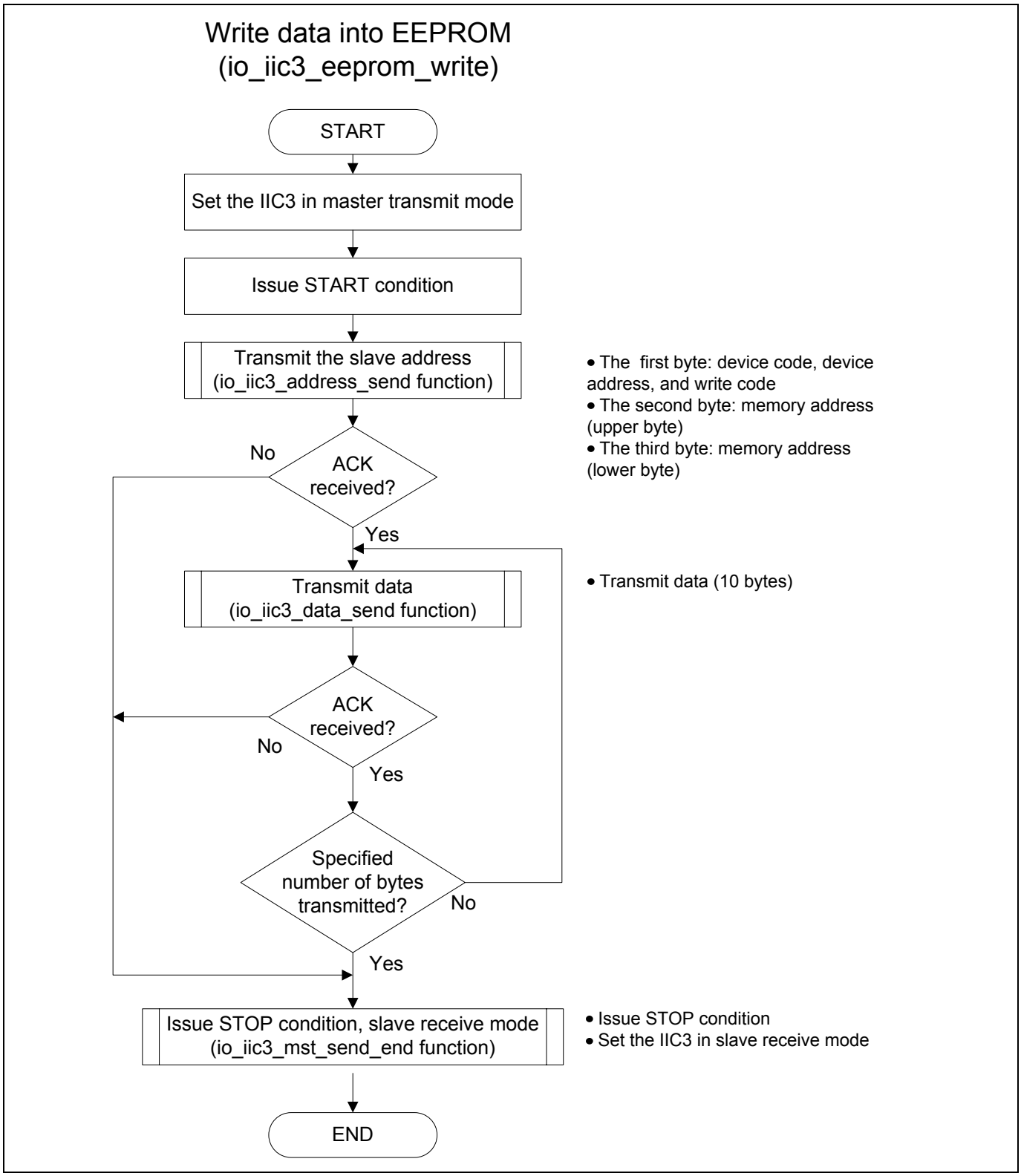

Figure 6 Sample Program Flow Chart (2/5)

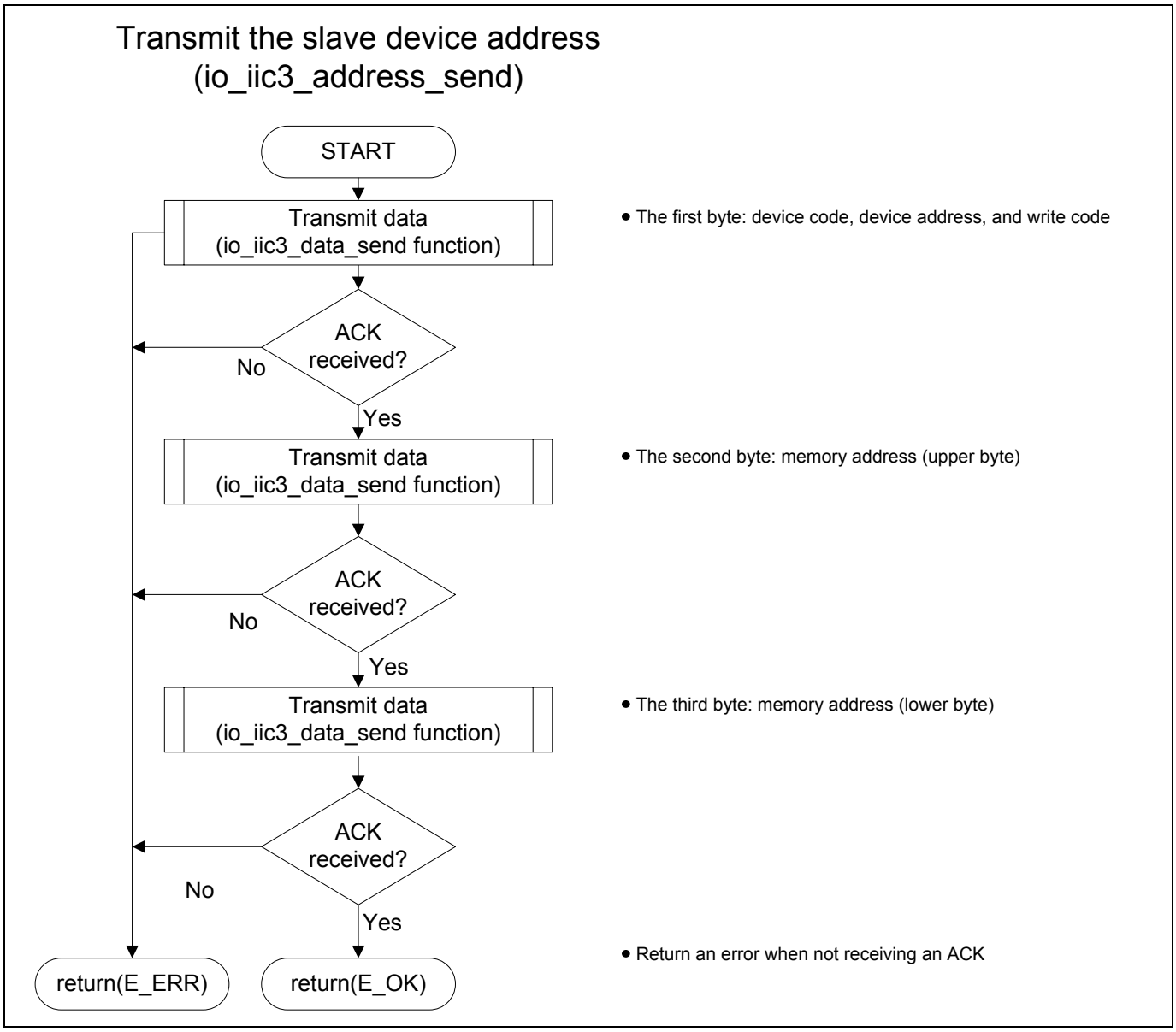

Figure 7 Sample Program Flow Chart (3/5)

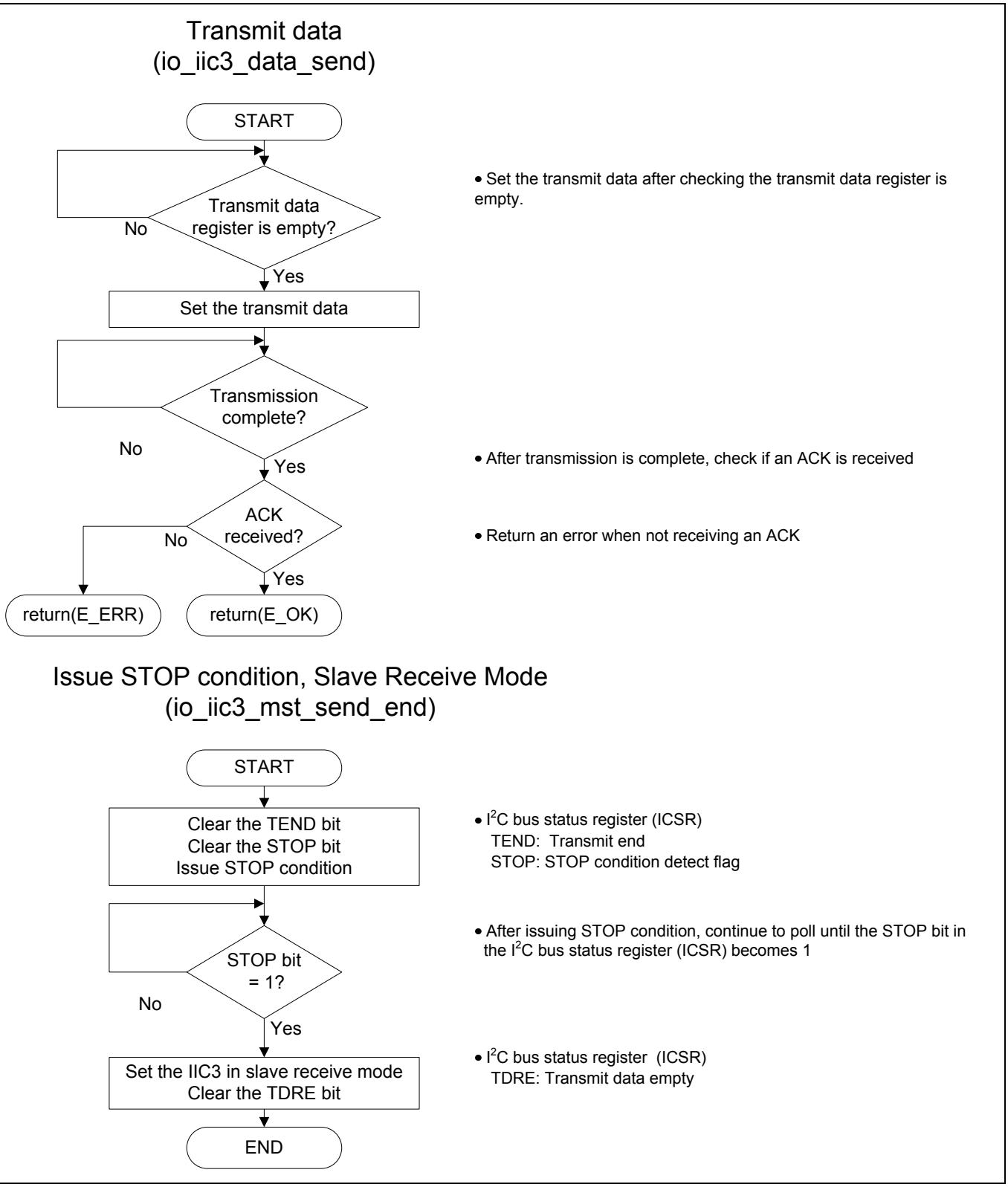

Figure 8 Sample Program Flow Chart (4/5)

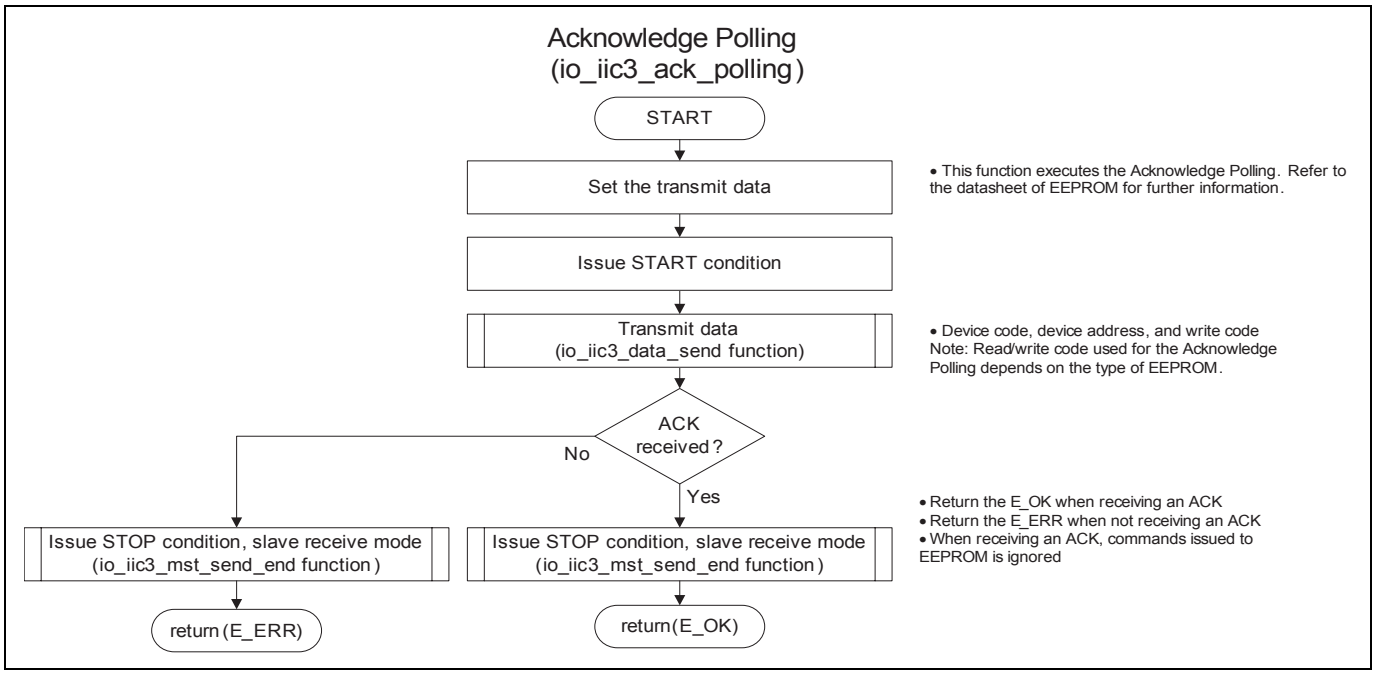

Figure 9 Sample Program Flow Chart (5/5)

3. Sample Program Listing

3.1 Sample Program Listing "main.c" (1/9)

```
1       /*""FILE COMMENT""*********** Technical reference data *************************
2       *
3       *       System Name : SH7264 Sample Program
4       *       File Name   : main.c
5       *       Abstract    : IIC3 Master transmit mode sample program
6       *       Version     : 1.00.00
7       *       Device      : SH7264/SH7262
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     : Jan.23,2009 Ver.1.00.00
24      *""FILE COMMENT END""*************************************************************/
25      #include <machine.h>
26      #include "iodefine.h"       /* SH7264 iodefine  */
27
28      /* ==== symbol definition ==== */
29      #define EEPROM_MEM_ADDR 0x0000
30      #define DEVICE_CODE 0xA0   /* EEPROM device code   :b'1010       */
31      #define DEVICE_ADDR 0x00   /* EEPROM device address:b'000    */
32      #define IIC_DATA_WR 0x00   /* Data write code      :b'0      */
33      #define IIC_DATA_RD 0x01   /* Data read code       :b'1      */
34      #define IIC3_DATA 10
35
36      #define E_OK 0
37      #define E_ERR -1
38
39      /* ==== RAM allocation variable declaration ==== */
40      unsigned char WriteData[IIC3_DATA];
41
```

3.2 Sample Program Listing "main.c" (2/9)

```
42    /* ==== prototype declaration ==== */
43    void main(void);
44    int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,
45                             unsigned short w_adr,unsigned int w_size,unsigned char* w_buf);
46    int io_iic3_data_send(unsigned char data);
47    int io_iic3_address_send(unsigned char* data);
48    void io_iic3_mst_send_end(void);
49    int io_iic3_init(void);
50    int io_iic3_ack_polling(void);
51
52    /*""FUNC COMMENT""*****************************************************************
53     * ID          :
54     * Outline     : Sample program main
55     *-------------------------------------------------------------------------------
56     * Include     :
57     *-------------------------------------------------------------------------------
58     * Declaration : void main(void);
59     *-------------------------------------------------------------------------------
60     * Description : Transmits data to EEPROM using the IIC3 master transmit mode.
61     *-------------------------------------------------------------------------------
62     * Argument    : void
63     *-------------------------------------------------------------------------------
64     * Return Value: void
65     *""FUNC COMMENT END""*************************************************************/
66    void main(void)
67    {
68      int i,ack;
69
70      /* ==== Creates the write data ==== */
71      for(i=0;i<IIC3_DATA;i++){
72        WriteData[i] = IIC3_DATA+i;
73      }
74
75      /* ==== Initializes the IIC3 ==== */
76      io_iic3_init();
77
78      /* ==== Transmits data in the IIC3 master transmit mode ==== */
79      ack = io_iic3_eeprom_write(DEVICE_CODE,     /* Device code */
80                         DEVICE_ADDR,          /* Device address */
81                         0x0000,               /* Write start address */
82                         sizeof(WriteData),        /* Write data size */
83                         WriteData);           /* Buffer storing data */
84
85      if( ack == E_OK ){
```

3.3 Sample Program Listing "main.c" (3/9)

```
86        /* ==== Acknowledge Polling ==== */
87        while(io_iic3_ack_polling() != E_OK){
88            /* Waits until reprogramming EEPROM internally is complete */
89        }
90      }
91
92      while(1){
93        /* Loop */
94      }
95    }
96
97    /*""FUNC COMMENT""*****************************************************************
98     * ID          :
99     * Outline     : Initializes the IIC3 module
100    *-------------------------------------------------------------------------------
101    * Include     : #include "iodefine.h"
102    *-------------------------------------------------------------------------------
103    * Declaration : int io_iic3_init(void);
104    *-------------------------------------------------------------------------------
105    * Description : Initializes the IIC3 channel 1.
106    *-------------------------------------------------------------------------------
107    * Argument    : void
108    *-------------------------------------------------------------------------------
109    * Return Value: E_OK
110    *""FUNC COMMENT END""*************************************************************/
111   int io_iic3_init(void)
112   {
113
114     /* ---- STBCR5 ---- */
115     CPG.STBCR5.BIT.MSTP56   = 0;    /* IIC3 channel 1 is operating */
116
117     /* ---- PORT ---- */
118     PORT.PECR0.BIT.PE2MD = 0x01;/* SCL1 select */
119     PORT.PECR0.BIT.PE3MD = 0x01;/* SDA1 select */
120
121
122     /* ---- Enables the IIC3 module operation ---- */
123     IIC3_1.ICCR1.BIT.ICE = 1u;      /* IIC3 module is operation-enabled */
124     IIC3_1.ICCR1.BIT.RCVD = 0u;     /* Continues to the next reception */
125     IIC3_1.ICCR1.BIT.MST = 1u;      /* Selects the master */
126     IIC3_1.ICCR1.BIT.TRS = 1u;      /* Selects the transmission */
127
128     IIC3_1.ICCR1.BIT.CKS = 5u;      /* Transfer clock rate at Pφ/92 (391 kHz) */
```

3.4 Sample Program Listing "main.c" (4/9)

```
129     /*  --- Sets the IIC bus mode register (ICMR) --- */
130     IIC3_1.ICMR.BYTE = 0x30u;
131               /*
132                     bit 7      : MLS:0 ---------------------    MSB first
133                     bits 6 to 4: Reserve:1 -----------------    Reserve bit
134                     bit 3      : BCWP:0---------------------    Not set
135                     bits 2 to 0: BC0:0, BC1:0,BC0:0---------    IIC format 9-bit
136               */
137
138     return(E_OK);
139   }
140
141   /*""FUNC COMMENT""*****************************************************************
142    * ID          :
143    * Outline     : Write data into EEPROM
144    *-------------------------------------------------------------------------------
145    * Include     : #include "iodefine.h"
146    *-------------------------------------------------------------------------------
147    * Declaration : int io_iic3_eeprom_write(unsigned char d_code,
148    *             :                          unsigned char d_adr,
149    *             :                          unsigned short w_adr,
150    *             :                          unsigned int w_size,
151    *             :                          unsigned char* w_buf);
152    *-------------------------------------------------------------------------------
153    * Description : Writes the w_size bytes of data stored in the buffer specified
154    *             : by the "w_buf" into EEPROM specified by the device code "d_code",
155    *             : device address "d_adr". Specify the memory address of EEPROM by
156    *             : the "w_adr".
157    *-------------------------------------------------------------------------------
158    * Argument    : unsigned char d_code : Device code
159    *             : unsigned char d_adr  : Device address
160    *             : unsigned short w_adr : Write start address
161    *             : unsigned int w_size  : Write data size
162    *             : unsigned char* w_buf : BUffer storing the write data
163    *-------------------------------------------------------------------------------
164    * Return Value: ACK received: E_OK
165    *             : NO ACK received: E_ERR
166    *""FUNC COMMENT END""*************************************************************/
```

3.5 Sample Program Listing "main.c" (5/9)

```
167   int io_iic3_eeprom_write(unsigned char d_code,unsigned char d_adr,unsigned short w_adr,
168                            unsigned int w_size,unsigned char* w_buf)
169   {
170     int ack = E_OK;
171     int i;
172     unsigned char send[3];
173
174     send[0] = (unsigned char)(d_code|((d_adr & 0x7)<<1)|IIC_DATA_WR);
175     send[1] = (unsigned char)((w_adr>>8) & 0x00ff);
176     send[2] = (unsigned char)(w_adr & 0x00ff);
177
178     while(IIC3_1.ICCR2.BIT.BBSY == 1u){
179       /* Waits for the bus release */
180     }
181     IIC3_1.ICCR1.BYTE |= 0x30u;                              /* Sets the IIC3 in the
182                                                              master transmit mode */
183     IIC3_1.ICCR2.BYTE = ((IIC3_1.ICCR2.BYTE & 0xbfu)|0x80u);   /* Issues START condition */
184
185     ack = io_iic3_address_send(send);                        /* Transmits the 1st, 2nd,
186                                                              and 3rd bytes */
187
188     if(ack == E_OK){
189       /* Received an ACK from the specified device */
190       for(i=0;i<w_size;i++){
191           ack = io_iic3_data_send(*w_buf++);            /* Transmits data */
192           if(ack == E_ERR){
193               break;
194           }
195       }
196       io_iic3_mst_send_end();
197     }
198     else{
199       /* No ACK received from the specified device */
200       io_iic3_mst_send_end();
201     }
202     return(ack);
203   }
204
```

3.6 Sample Program Listing "main.c" (6/9)

```
205  /*""FUNC COMMENT""*****************************************************************
206   * ID          :
207   * Outline     : Transmit the slave device address
208   *-------------------------------------------------------------------------------
209   * Include     :
210   *-------------------------------------------------------------------------------
211   * Declaration : int io_iic3_address_send(unsigned char* data);
212   *-------------------------------------------------------------------------------
213   * Description : Transmits the address of the slave device (1 byte) and the memory
214   *             : address (2 bytes) specified by the argument "data".
215   *-------------------------------------------------------------------------------
216   * Argument    : unsigned char* data : Transmit data
217   *-------------------------------------------------------------------------------
218   * Return Value: ACK received: E_OK
219   *             : No ACK received: E_ERR
220   *""FUNC COMMENT END""*************************************************************/
221  int io_iic3_address_send(unsigned char* data)
222  {
223    int ack;
224
225    ack = io_iic3_data_send(*data++);       /* Slave device address */
226    if(ack == E_ERR){
227      return(ack);
228    }
229    ack = io_iic3_data_send(*data++);       /* 1st memory address */
230    if(ack == E_ERR){
231      return(ack);
232    }
233    ack = io_iic3_data_send(*data);         /* 2nd memory address */
234    if(ack == E_ERR){
235      return(ack);
236    }
237    return(ack);
238  }
239
```

3.7 Sample Program Listing "main.c" (7/9)

```
240   /*""FUNC COMMENT""*****************************************************************
241    * ID          :
242    * Outline     : Transmit one byte of data
243    *-------------------------------------------------------------------------------
244    * Include     : #include "iodefine.h"
245    *-------------------------------------------------------------------------------
246    * Declaration : int io_iic3_data_send(unsigned char data);
247    *-------------------------------------------------------------------------------
248    * Description : Transmits the"data" as the following steps.
249    *             : 1. Waits for the ICDRT empty
250    *             : 2. Sets the transmit data
251    *             : 3. Confirms the transmission is complete
252    *             : 4. Confirms an ACK is received
253    *-------------------------------------------------------------------------------
254    * Argument    : unsigned char data : Transmit data
255    *-------------------------------------------------------------------------------
256    * Return Value: ACK received: E_OK
257    *             : NO ACK received: E_ERR
258    *""FUNC COMMENT END""*************************************************************/
259   int io_iic3_data_send(unsigned char data)
260   {
261     int ack;
262
263     while(IIC3_1.ICSR.BIT.TDRE == 0u){
264       /* Waits for the ICDRT empty */
265     }
266     IIC3_1.ICDRT = data;
267     while(IIC3_1.ICSR.BIT.TEND == 0u){
268       /* Waits until the transmission is complete */
269     }
270     if(IIC3_1.ICIER.BIT.ACKBR == 0u){
271       ack = E_OK;
272     }
273     else{
274       ack = E_ERR;
275     }
276     return(ack);
277   }
278
```

3.8 Sample Program Listing "main.c" (8/9)

```
279   /*""FUNC COMMENT""*****************************************************************
280    * ID          :
281    * Outline     : Issue STOP condition
282    *-------------------------------------------------------------------------------
283    * Include     : #include "iodefine.h"
284    *-------------------------------------------------------------------------------
285    * Declaration : void io_iic3_mst_send_end(void);
286    *-------------------------------------------------------------------------------
287    * Description : Issues STOP condition, and switches the mode to slave receive mode.
288    *-------------------------------------------------------------------------------
289    * Argument    : void
290    *-------------------------------------------------------------------------------
291    * Return Value: void
292    *""FUNC COMMENT END""*************************************************************/
293   void io_iic3_mst_send_end(void)
294   {
295     IIC3_1.ICSR.BIT.TEND = 0u;      /* Clears the TEND flag */
296     IIC3_1.ICSR.BIT.STOP = 0u;      /* Clears the STOP flag */
297     IIC3_1.ICCR2.BYTE &= 0x3fu;     /* Issues STOP condition */
298
299     while(IIC3_1.ICSR.BIT.STOP == 0u){
300       /* Waits for the bus release */
301     }
302
303     IIC3_1.ICCR1.BYTE &= 0xcfu;     /* Slave receive mode */
304     IIC3_1.ICSR.BIT.TDRE = 0u;      /* Clears the TDRE */
305   }
306
```

3.9 Sample Program Listing "main.c" (9/9)

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : Acknowledge Polling
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration : io_iic3_ack_polling
 *-----------------------------------------------------------------------------
 * Description : Checks if the write cycle of the EEPROM is finished.
 *              : EEPROM ignores the input command and does not return an ACK when
 *              : the write cycle is not finished. Access EEPROM after checking that
 *              : the write cycle of EEPROM is finished. Read/Write codes to transmit
 *              : upon the Acknowledge Polling depends on the type of EEPROM.
 *              : Refer to the datasheet of EEPROM for further information.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value: E_OK   : NOT_BUSY
 *              : E_ERR : BUSY (EEPROM is in the write cycle).
 *""FUNC COMMENT END""*************************************************************/
int io_iic3_ack_polling(void)
{
  int ack = E_OK;
  unsigned char send = (unsigned char)(DEVICE_CODE|((DEVICE_ADDR & 0x7)<<1)|IIC_DATA_WR);

  while(IIC3_1.ICCR2.BIT.BBSY == 1u){
    /* Waits for the bus release */
  }
  IIC3_1.ICCR1.BYTE |= 0x30u;                              /* Sets the IIC3 in
                                                           master transmit mode */
  IIC3_1.ICCR2.BYTE = ((IIC3_1.ICCR2.BYTE & 0xbfu)|0x80u);  /* Issues START condition */

  ack = io_iic3_data_send(send);

  io_iic3_mst_send_end();     /* Issues STOP condition */

  return(ack);
}
/* 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.