

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.

SH7144/45 Group

Single-Master Transmission, Single-Master Reception

Contents

1.	Single-Master Transmission.....	2
1.1	Specifications	2
1.2	Operation.....	4
1.3	Software	5
1.4	Flowcharts	8
1.5	Program Listing	11
2.	Single-Master Reception	15
2.1	Specifications	15
2.2	Operation.....	16
2.3	Software	17
2.4	Flowcharts	20
2.5	Program Listing	24

1. Single-Master Transmission

1.1 Specifications

- Ten bytes of data are written to EEPROM (HN58X2464, 64k bits, 8 words × 8 bits) using channel 0 of the SH7145F's I²C Bus interface.
- The slave address of the connected EEPROM is [1010000], and data is written to EEPROM memory addresses H'0000 through H'0009.
- The write data is [H'01, H'02, H'03, H'04, H'05, H'06, H'07, H'08, H'09, H'0A].
- The devices connected to the I²C Bus of this system are a master device (SH7145F) and a slave device (EEPROM) in a single-master configuration.
- The I²C Bus data transfer clock frequency is 156 kHz.
- The SH7145F operating frequencies are 40 MHz for the CPU clock and 40 MHz for the on-chip peripheral clock.

Figure 1.1 shows an example of connection between an SH7145F and EEPROM.

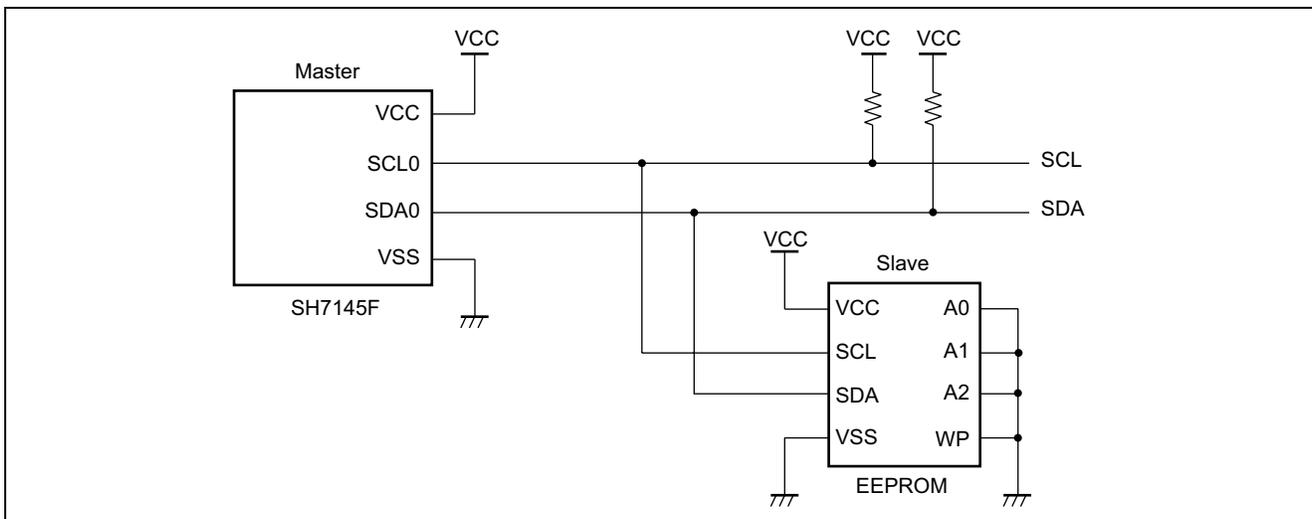


Figure 1.1 Example of Connection between SH7145F and EEPROM

The I²C Bus format used in this sample task is shown in figure 1.2.

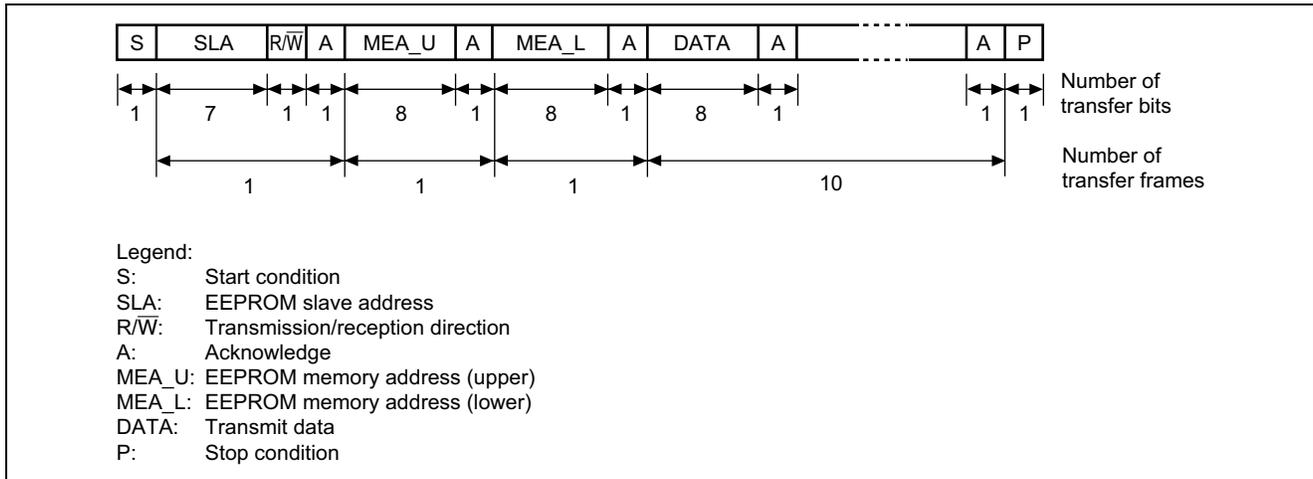


Figure 1.2 Transfer Format Used in this Task

1.2 Operation

Figure 1.3 shows the principles of operation of this task.

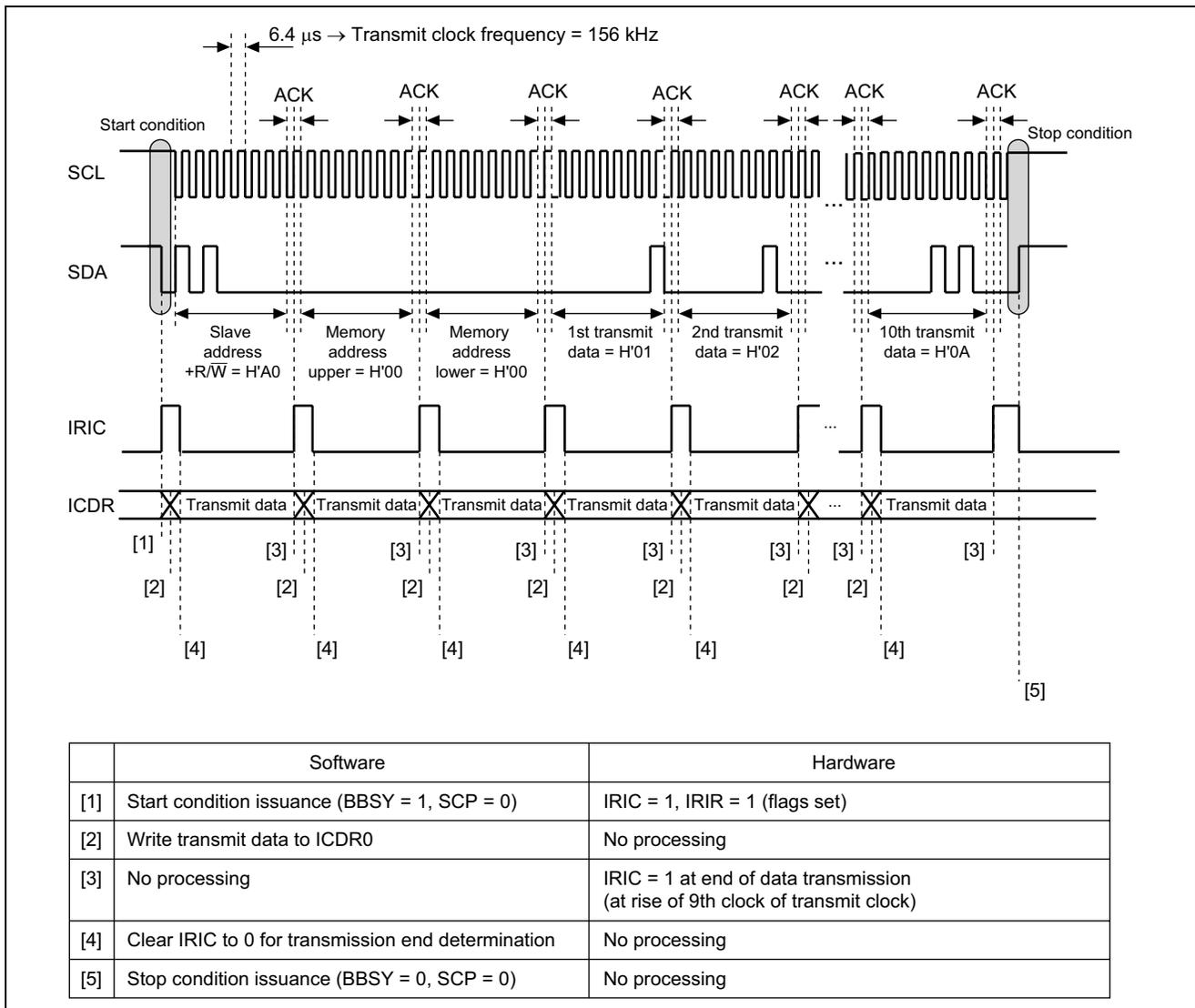


Figure 1.3 Principles of Operation of Single-Master Transmission

1.3 Software

(1) Modules

The modules used in this sample task are shown in the table below.

Table 1.1 Modules

Module Name	Label	Function
Main routine	main	I ² C initialization, pin setting
Dummy interrupt routine	dummy	Dummy interrupt handling
EEPROM write routine	Write_page_EEPROM	n-byte EEPROM write routine
Address setting routine	Set_adrs_EEPROM	Start condition generation, slave address issuance, EEPROM address setting

(2) Internal Registers Used

The internal registers used in this sample task are shown in the table below.

Table 1.2 Internal Registers Used

Register Name		Function	Address	Set Value
Bit(s)			Bit(s)	
MSTCR1		Module standby control register 1	H'FFFF861C	
	MSTP21	I ² C module standby control bit Module standby cleared when MSTP21 = 0	Bit 5	B'0
PBCR1		Port B control register 1 Used to set port B pin functions in combination with port B control register 2	H'FFFF8398	H'0C00
PBCR2		Port B control register 2 Used to set port B pin functions in combination with port B control register 1 Sets port B3 (PB3) pin function as I ² C SDA0 I/O pin Sets port B2 (PB2) pin function as I ² C SCL0 I/O pin	H'FFFF839A	H'0000
ICDR0		I ² C Bus data register 8-bit readable/writable register, used as transmission data register when transmitting, and as reception data register when receiving.	H'FFFF880E	—
SAR0		Slave address register	H'FFFF880F	H'00
	SVA6-0	Slave address Unique address different from that of other slaves connected to I ² C Bus is set in bits SVA6 to SVA0.	Bits 7 to 1	
	FS	Format select Selects transfer format, together with FSX bit in SARX. Transfer format is I ² C Bus format when FS = FSX = 0.	Bit 0	
SARX0		Second slave address register	H'FFFF880E	H'00
	SVAX6-0	Second slave address Unique address different from that of other slaves connected to I ² C Bus is set in bits SVAX6 to SVAX0.	Bits 7 to 1	
	FSX	Format select Selects transfer format, together with FS bit in SAR. Transfer format is I ² C Bus format when FS = FSX = 0.	Bit 0	

Register Name	Bit(s)	Function	Address	Set Value
			Bit(s)	
ICMR0		I ² C Bus mode register	H'FFFF880F	H'38
	MLS	MSB-first/LSB-first selection MSB-first when MLS = 0	Bit 7	
	WAIT	Wait insertion bit Data and acknowledgment transferred continuously when WAIT = 0	Bit 6	
	CKS2	Transfer clock selection 2-0	Bit 5	
	CKS1 CKS0	Used to set transfer clock frequency in combination with IICX0 bit in SCRX register. 156 kHz (P _φ = 40 MHz) when IICX = B'1, CKS[2:0] = B'111	Bit 4 Bit 3	
BC2 BC1 BC0	Bit counter Used to set number of data bits to be transferred next in I ² C Bus format as 9 bits (including ACK bit)/frame. BC[2:0] = B'000	Bit 2 Bit 1 Bit 0		
ICCR0		I ² C Bus control register	H'FFFF8808	H'89
	ICE	I ² C Bus interface enable (ICE) When ICE = B'1, I ² C module is enabled for transfer, and ICMR and ICDR registers are valid.	Bit 7	
	IEIC	I ² C Bus interrupt enable Interrupt requests disabled when IEIC = B'0	Bit 6	
	MST	Master/slave selection Slave mode when MST = B'0	Bit 5	
	TRS	Transmission/reception selection Transmit mode when TES = B'0	Bit 4	
	ACKE	Acknowledge bit determination selection When ACKE = B'1, continuous transfer is suspended when acknowledge bit is 1.	Bit 3	
	BBSY	Busy bit Bus released state when BBSY = B'0	Bit 2	
	IRIC	I ² C Bus interface interrupt request flag Interrupt generated when IRIC = B'1	Bit 1	
SCP	Start condition/stop condition issuance disable bit When SCP = B'0, issues start condition, stop condition in combination with BBSY flag.	Bit 0		
ICSR0		I ² C Bus status register	H'FFFF8809	—
	ESTP	Error stop condition detection flag	Bit 7	
	STOP	Normal stop condition detection flag	Bit 6	
	IRTR	I ² C Bus interface continuous transmission/reception interrupt request flag	Bit 5	
	AASX	Second slave address recognition flag	Bit 4	
	AL	Arbitration lost flag	Bit 3	
	AAS	Slave address recognition flag	Bit 2	
	ADZ	General call address recognition flag	Bit 1	
ACKB	Acknowledge bit Stores acknowledge data.	Bit 0		

Register Name	Function	Address	Set Value
		Bit(s)	
SCRX	Serial control register X	H'FFFF87F0	H'39
Reserved	Reserved bits Always read as 0. Write value should always be 0.	Bits 7, 6	
IICX0	I ² C transfer rate select 0 Selects master mode transfer rate in combination with CKD[2:0] in ICMR. IICX0 = B'1	Bit 5	
IICE	I ² C master enable When IICE = B'1, I ² C Bus interface register access is enabled.	Bit 4	
HNDS	Handshake reception bit When HNDS = B'1, continuous reception operation is disabled.	Bit 3	
Reserved	Reserved bit Always read as 0. Write value should always be 0.	Bit 2	
ICDRF0	Indicates whether there is valid receive data in ICDR.	Bit 1	
STOPIM	Stop condition detection interrupt mask When STOPIM = B'1, IRIC interrupt generation is suppressed in slave mode even if a stop condition is detected.	Bit 0	

(3) Variables

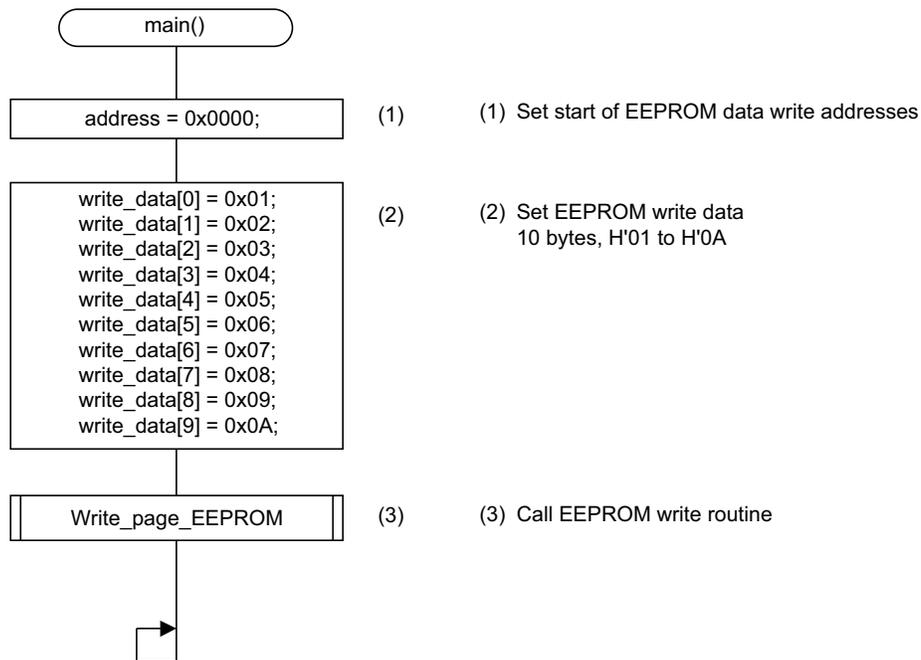
Variable	Function	Data Length	Initial Value	Module
write_data[0]	1st byte of transmit data	1 byte	H'01	Main routine
write_data[1]	2nd byte of transmit data	1 byte	H'02	Main routine
write_data[2]	3rd byte of transmit data	1 byte	H'03	Main routine
write_data[3]	4th byte of transmit data	1 byte	H'04	Main routine
write_data[4]	5th byte of transmit data	1 byte	H'05	Main routine
write_data[5]	6th byte of transmit data	1 byte	H'06	Main routine
write_data[6]	7th byte of transmit data	1 byte	H'07	Main routine
write_data[7]	8th byte of transmit data	1 byte	H'08	Main routine
write_data[8]	9th byte of transmit data	1 byte	H'09	Main routine
write_data[9]	10th byte of transmit data	1 byte	H'0A	Main routine
address	EEPROM write start address	2 bytes	H'0000	Main routine
adrs	EEPROM write start address copy	2 bytes	—	EEPROM write routine
num	Number of transmit data	1 byte	H'0A	EEPROM write routine
w_data	Pointer variable to transmit data array variable write_data	4 bytes	—	EEPROM write routine
ack	Acknowledge reception determination flag	1 byte	H'01	EEPROM write routine

(4) RAM Used

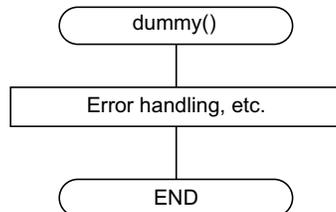
This sample task does not use any RAM apart from the variables.

1.4 Flowcharts

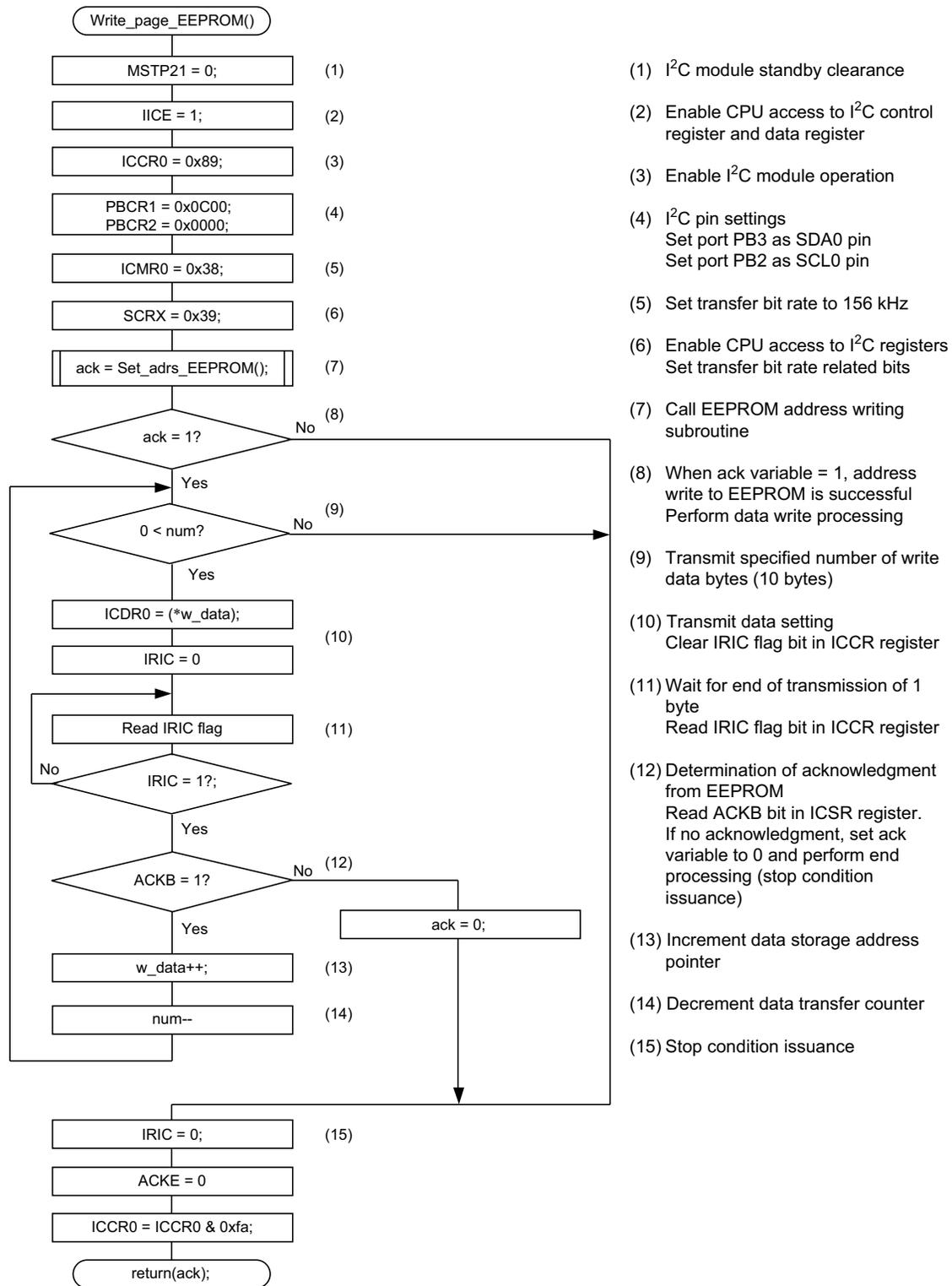
(1) Main routine



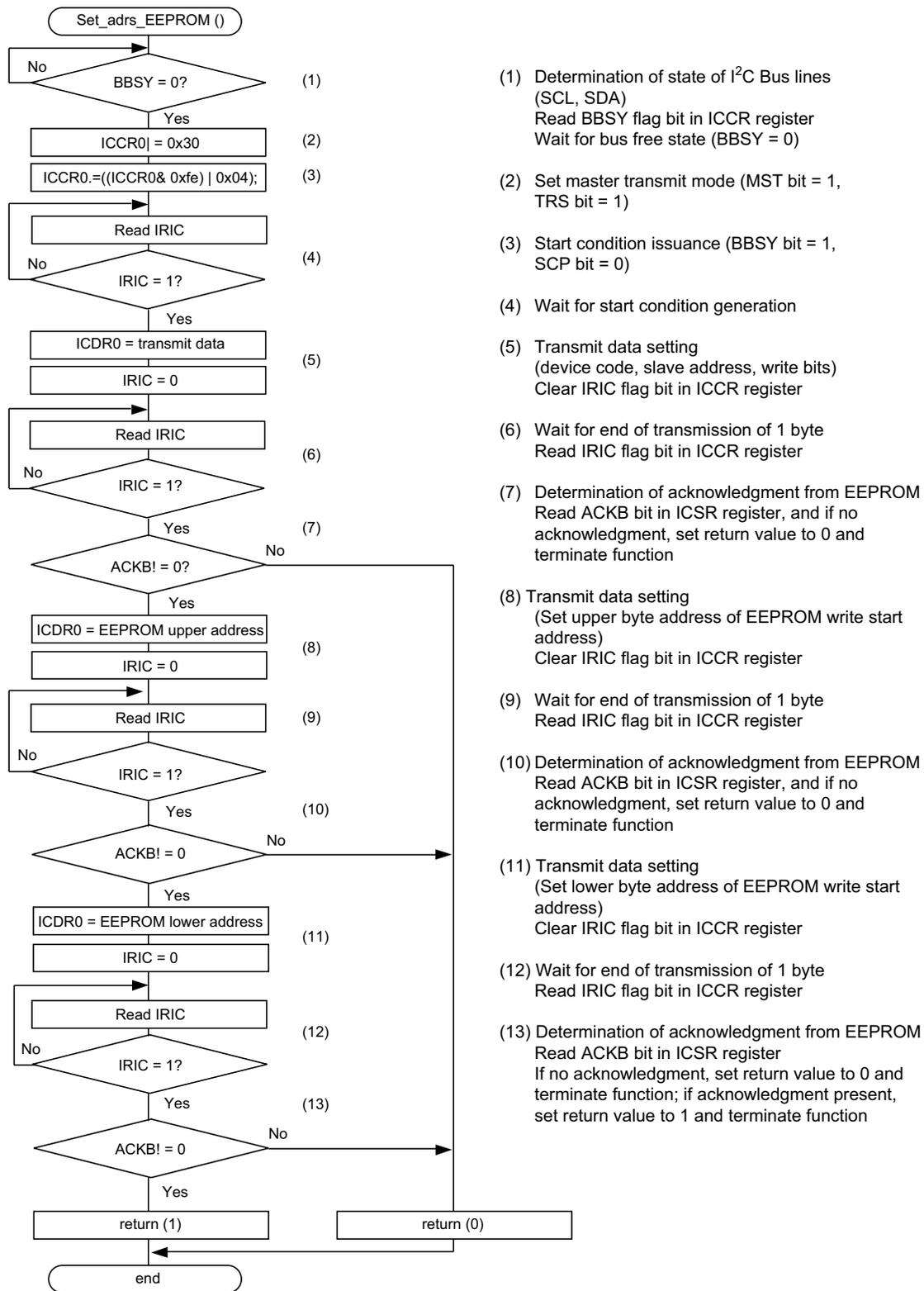
(2) Dummy interrupt routine



(3) EEPROM write subroutine



(4) Start condition issuance, slave address and EEPROM memory address transmission subroutine



1.5 Program Listing

```

//*****
//  SH7144F Group -SH7145- I2C-bus Application Note
//      Single master transmit
//      n Byte data write/read 64kbit EEPROM
//      Clock :CPU=40MHz (External input=10MHz)
//      :Peripheral=40MHz
//      I2c bit rate :156kHz
//      Written      :2003/2/1 Rev.2.0
//*****
#include <machine.h>
#include "iodefine.h"    // SH7145  I/O Register Definition

//----- Symbol Definition -----
#define DEVICE_CODE  0xa0          // EEPROM DEVICE CODE:b'1010
#define SLAVE_ADRS   0x00          // SLAVE ADRS:b'00
#define IIC_DATA_W   0x00          // WRITE DATA:b'0
#define IIC_DATA_R   0x01          // READ DATA:b'1
#define DATA_NUM    10           // data size

//----- Function Definition -----
void main(void);
void dummy(void);

unsigned char Write_page_EEPROM(unsigned short,unsigned char*,unsigned char);
unsigned char Set_adrs_EEPROM(unsigned short);

//*****
//  main
//*****
void main(void)
{
    unsigned short  address;        // EEPROM memory address

    address= 0x0000;                // set EEPROM address

    // set write data
    write_data[0]=0x01;
    write_data[1]=0x02;
    write_data[2]=0x03;
    write_data[3]=0x04;
    write_data[4]=0x05;
    write_data[5]=0x06;
    write_data[6]=0x07;
    write_data[7]=0x08;
    write_data[8]=0x09;
    write_data[9]=0x0a;

```

```

// EEPROM data write
Write_page_EEPROM(address,write_data,DATA_NUM);

while(1);

}

//*****
// dummy interrupt function
//*****
#pragma interrupt(dummy)
void dummy(void)
{
    // Interrupt error
}

//*****
// Write_page_EEPROM
// argument1 ;write address(unsigned short)
// argument2 ;write data(unsigned char)
// argument3 ;write data number(unsigned char)
// return ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
//*****
unsigned char Write_page_EEPROM(unsigned short adrs,unsigned char* w_data,unsigned char
num)
{
    unsigned char ack; // ACK check flag

    // Set standby mode
    P_STBY.MSTCR1.BIT.MSTP21 = 0; // disable I2C standby mode

    ack = 1;
    P_IIC.SCRX.BIT.IICE = 1; // Enables CPU access to the register

    P_IIC.ICCR0.BYTE = 0x89;
        // ICE(7)=b'1 Enable I2C bus interface
        // IEIC(6)=b'0 Disables the interrupt
        // MST(5)=b'0 Slave mode
        // TRS(4)=b'0 Receive mode
        // ACKE(3)=b'1 Continuous data transfer is halted
        // BBSY(2)=b'0
        // IRIC(1)=b'0
        // SCP(0)=b'1 Start/stop condition issuance disabling

    // set I2C pin function
    P_PORTB.PBCR1.WORD = 0x0c00; // SDA0(PB3-32pin@SH7145F),SCL0(PB2-31pin@SH7145F)
    P_PORTB.PBCR2.WORD = 0x0000;

```

```

P_IIC.ICMR0.BYTE = 0x38;
    // MILS(7)=b'0 MSB first
    // WAIT(6)=b'0 A wait state is inserted between DATA and ACK
    // CKS2[2:0](5:3)=b'111 Transfer clock select
    //          156kHz@(P-fai=40MHz,IICX=1)
    //          39.1kHz@(P-fai=10MHz,IICX=1)
P_IIC.SCRX.BYTE = 0x39;
    // IICX(5)=b'1 transfer-rate select, reference CKS bit
    // IICE(4)=b'1 Enables CPU access to the register
    // HNDS(3)=b'1
    // STOPIM(0)=b'1 disables interrupt requests

// Set device code,EEPROM address
ack = Set_adrs_EEPROM(adrs);

// EEPROM write data Transmission (n byte)
if( ack==1 ){
    for( ; 0<num; num-- ){
        P_IIC.ICDR0.BYTE = (*w_data); // write data set
        P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
        while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
        if( P_IIC.ICSR0.BIT.ACKB != 0 ){ // Test the acknowledge bit
            ack = 0; // no ACK
            break;
        }
        w_data++; // write data pointer increment
    }
}

// Stop condition issuance
P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
P_IIC.ICCR0.BIT.ACKE = 0; // set AKCE=0
P_IIC.ICCR0.BYTE = P_IIC.ICCR0.BYTE & 0xfa; // Stop condition
issuance(BBSY=0,SCP=0)

return(ack);
}

```

```

//*****
//  Set_adrs_EEPROM
//      argument1  ;write address(unsigned short)
//      return      ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
//*****
unsigned char Set_adrs_EEPROM(unsigned short adrs)
{

    while( P_IIC.ICCR0.BIT.BBSY!=0 ); // BUS FREE?(BBSY==0→Bus Free)

    // Master-Transmission,Generate the start condition.
    P_IIC.ICCR0.BYTE |= 0x30; // Select master transmit mode(MST=1,TRS=1)

    P_IIC.ICCR0.BYTE=((P_IIC.ICCR0.BYTE & 0xfe) | 0x04);
    // Generate start condition(BBSY=1,SCP=0)
    while( P_IIC.ICCR0.BIT.IRIC==0 );
    // Wait for a start condition generation

    // Slave address+W Transmission
    P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_W);
    // data set
    P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
    }

    // EEPROM upper address Transmission(1byte)
    P_IIC.ICDR0.BYTE = (unsigned char)(adrs>>8); // data set
    P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
    }

    // EEPROM lower address Transmission(1byte)
    P_IIC.ICDR0.BYTE = (unsigned char)(adrs & 0x00ff); // data set
    P_IIC.ICCR0.BIT.IRIC = 0; // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){ // Test the acknowledge bit
    return (0); // no ACK
    }

    return (1); // ACK OK
}

```

2. Single-Master Reception

2.1 Specifications

- Ten bytes of data are read from EEPROM (HN58X2464, 64k bits, 8 words × 8 bits) using channel 0 of the SH7145F's I²C Bus interface.
- The slave address of the connected EEPROM is [1010000], and data in EEPROM memory addresses H'0000 through H'0009 is read.
- The read data is captured in a variable array.
- The devices connected to the I²C Bus of this system are a master device (SH7145F) and a slave device (EEPROM) in a single-master configuration.
- The I²C Bus data transfer clock frequency is 156 kHz.
- The SH7145F operating frequencies are 40 MHz for the CPU clock and 40 MHz for the on-chip peripheral clock.

Figure 2.1 shows an example of connection between an SH7145F and EEPROM.

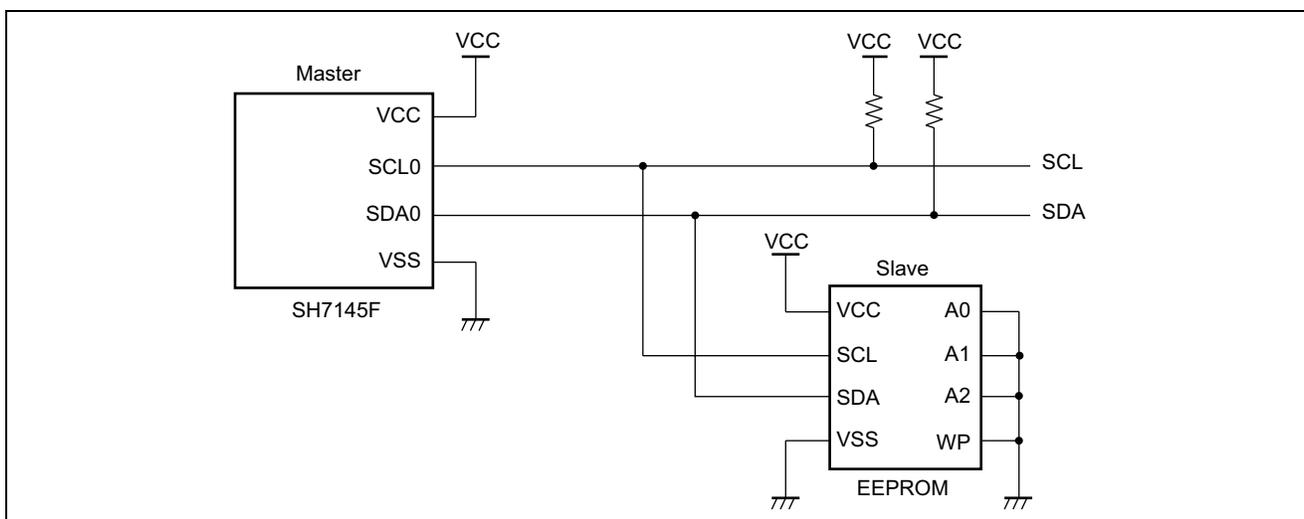


Figure 2.1 Example of Connection between SH7145F and EEPROM

The I²C Bus format used in this sample task is shown in figure 2.2.

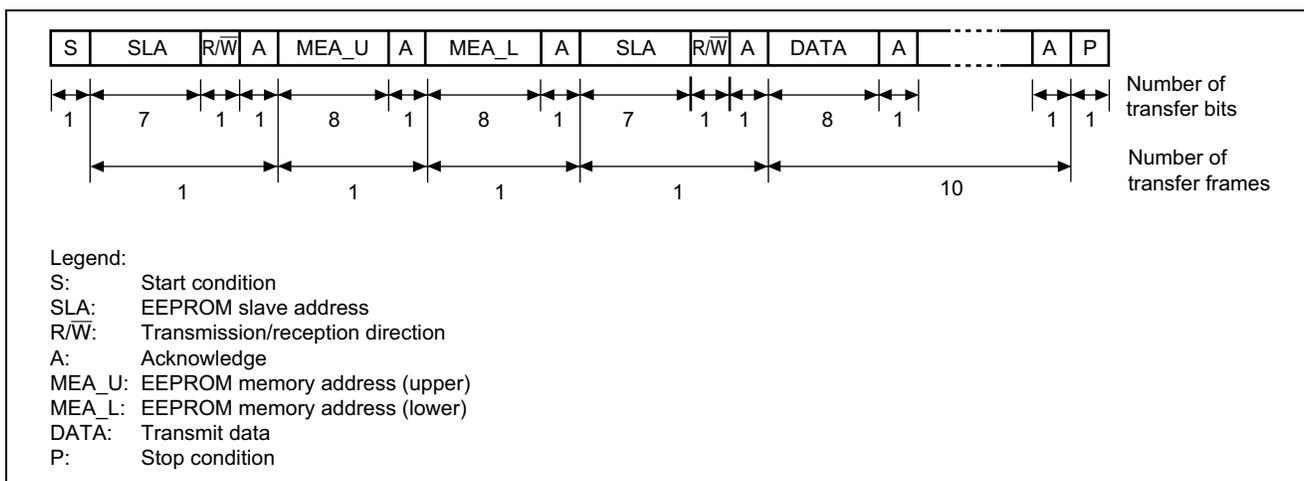


Figure 2.2 Transfer Format Used in this Task

2.2 Operation

Figure 2.3 shows the principles of operation of this task.

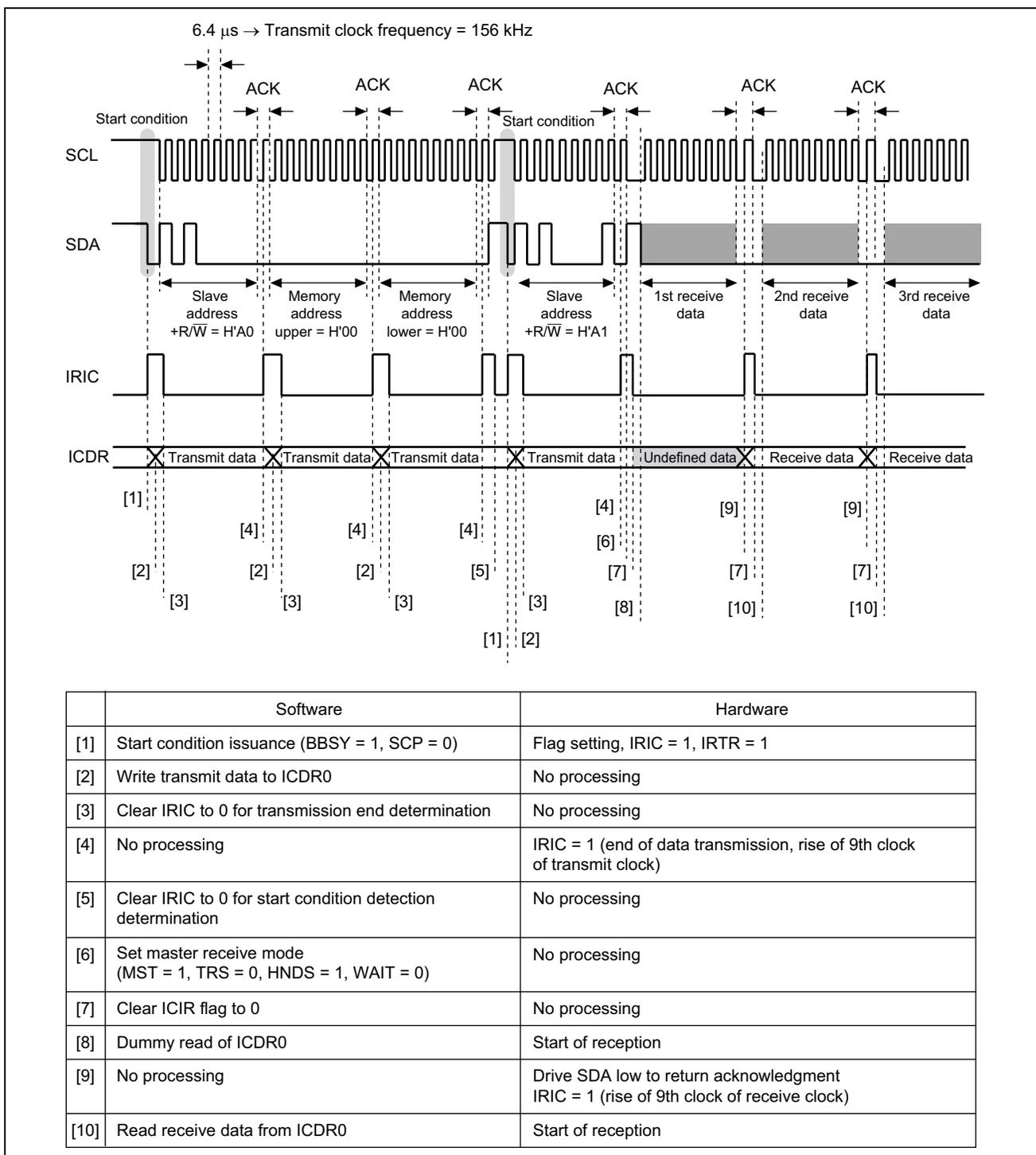


Figure 2.3 Principles of Operation of Single-Master Reception

2.3 Software

(1) Modules

The modules used in this sample task are shown in the table below.

Table 2.1 Modules

Module Name	Label	Function
Main routine	main	I ² C initialization, pin setting
Dummy interrupt routine	dummy	Dummy interrupt handling
EEPROM read routine	Read_page_EEPROM	n-byte EEPROM read routine (n > 1)
Address setting routine	Set_adrs_EEPROM	Start condition generation, slave address issuance, EEPROM address setting

(2) Internal Registers Used

The internal registers used in this sample task are shown in the table below.

Table 2.2 Internal Registers Used

Register Name		Function	Address	Set Value
Bit(s)			Bit(s)	
MSTCR1		Module standby control register 1	H'FFFF861C	
	MSTP21	I ² C module standby control bit Module standby cleared when MSTP21 = 0	Bit 5	B'0
PBCR1		Port B control register 1 Sets port B pin functions in combination with port B control register 2	H'FFFF8398	H'0C00
PBCR2		Port B control register 2 Used to set port B pin functions in combination with port B control register 1 Sets port B3 (PB3) pin function as I ² C SDA0 I/O pin Sets port B2 (PB2) pin function as I ² C SCL0 I/O pin	H'FFFF839A	H'0000
ICDR0		I ² C Bus data register 8-bit readable/writable register, used as transmission data register when transmitting, and as reception data register when receiving.	H'FFFF880E	—
SAR0		Slave address register	H'FFFF880F	H'00
	SVA6-0	Slave address Unique address different from that of other slaves connected to I ² C Bus is set in bits SVA6 to SVA0.	Bits 7 to 1	
	FS	Format select Selects transfer format, together with FSX bit in SARX. Transfer format is I ² C Bus format when FS = FSX = 0.	Bit 0	
SARX0		Second slave address register	H'FFFF880E	H'00
	SVAX6-0	Second slave address Unique address different from that of other slaves connected to I ² C Bus is set in bits SVAX6 to SVAX0.	Bits 7 to 1	
	FSX	Format select Selects transfer format, together with FS bit in SAR. Transfer format is I ² C Bus format when FS = FSX = 0.	Bit 0	

Register Name	Bit(s)	Function	Address	Set Value
			Bit(s)	
ICMR0		I ² C Bus mode register	H'FFFF880F	H'38
	MLS	MSB-first/LSB-first selection MSB-first when MLS = 0	Bit 7	
	WAIT	Wait insertion bit Data and acknowledgment transferred continuously when WAIT = 0	Bit 6	
	CKS2 CKS1 CKS0	Transfer clock selection 2-0 Used to set transfer clock frequency in combination with IICX0 bit in SCRX register. 156 kHz (P _φ = 40 MHz) when IICX = B'1, CKS[2:0] = B'111	Bit 5 Bit 4 Bit 3	
	BC2 BC1 BC0	Bit counter Used to set number of data bits to be transferred next in I ² C Bus format as 9 bits (including ACK bit)/frame. BC[2:0] = B'000	Bit 2 Bit 1 Bit 0	
ICCR0		I ² C Bus control register	H'FFFF8808	H'89
	ICE	I ² C Bus interface enable (ICE) When ICE = B'1, I ² C module is enabled for transfer, and ICMR and ICDR registers are valid.	Bit 7	
	IEIC	I ² C Bus interrupt enable Interrupt requests disabled when IEIC = B'0	Bit 6	
	MST	Master/slave selection Slave mode when MST = B'0	Bit 5	
	TRS	Transmission/reception selection Receive mode when TES = B'0	Bit 4	
	ACKE	Acknowledge bit determination selection When ACKE = B'1, continuous transfer is suspended when acknowledge bit is 1.	Bit 3	
	BBSY	Busy bit Bus released state when BBSY = B'0	Bit 2	
	IRIC	I ² C Bus interface interrupt request flag Interrupt generated when IRIC = B'1	Bit 1	
SCP	Start condition/stop condition issuance disable bit When SCP = B'0, issues start condition, stop condition in combination with BBSY flag.	Bit 0		
ICSR0		I ² C Bus status register	H'FFFF8809	—
	ESTP	Error stop condition detection flag	Bit 7	
	STOP	Normal stop condition detection flag	Bit 6	
	IRTR	I ² C Bus interface continuous transmission/reception interrupt request flag	Bit 5	
	AASX	Second slave address recognition flag	Bit 4	
	AL	Arbitration lost flag	Bit 3	
	AAS	Slave address recognition flag	Bit 2	
	ADZ	General call address recognition flag	Bit 1	
ACKB	Acknowledge bit Stores acknowledge data.	Bit 0		

Register Name	Bit(s)	Function	Address	Set Value
			Bit(s)	
SCRX		Serial control register X	H'FFFF87F0	H'39
	Reserved	Reserved bits Always read as 0. Write value should always be 0.	Bits 7, 6	
	IICX0	I ² C transfer rate select 0 Selects master mode transfer rate in combination with CKD[2:0] in ICMR. IICX0 = B'1	Bit 5	
	IICE	I ² C master enable When IICE = B'1, I ² C Bus interface register access is enabled.	Bit 4	
	HNDS	Handshake reception bit When HNDS = B'1, continuous reception operation is disabled.	Bit 3	
	Reserved	Reserved bit Always read as 0. Write value should always be 0.	Bit 2	
	ICDRF0	Indicates whether there is valid receive data in ICDR.	Bit 1	
	STOPIM	Stop condition detection interrupt mask When STOPIM = B'1, IRIC interrupt generation is suppressed in slave mode even if a stop condition is detected.	Bit 0	

(3) Variables

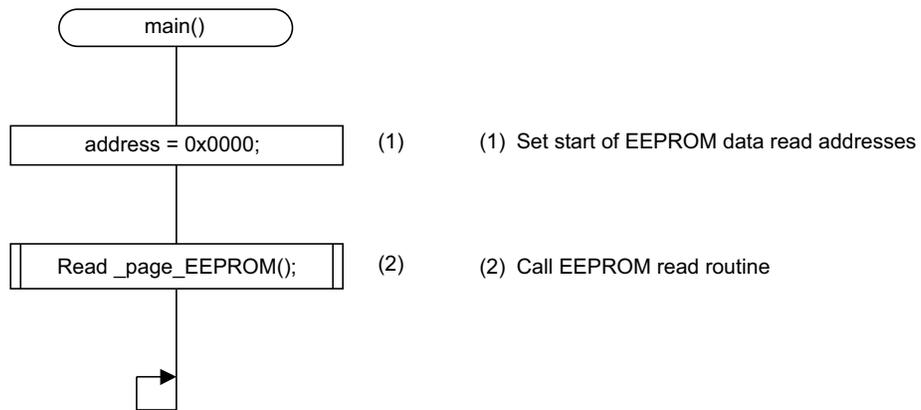
Variable	Function	Data Length	Initial Value	Module
read_data[0]	1st byte of receive data	1 byte	—	Main routine
read_data[1]	2nd byte of receive data	1 byte	—	Main routine
read_data[2]	3rd byte of receive data	1 byte	—	Main routine
read_data[3]	4th byte of receive data	1 byte	—	Main routine
read_data[4]	5th byte of receive data	1 byte	—	Main routine
read_data[5]	6th byte of receive data	1 byte	—	Main routine
read_data[6]	7th byte of receive data	1 byte	—	Main routine
read_data[7]	8th byte of receive data	1 byte	—	Main routine
read_data[8]	9th byte of receive data	1 byte	—	Main routine
read_data[9]	10th byte of receive data	1 byte	—	Main routine
address	EEPROM read start address	2 bytes	H'0000	Main routine
adrs	EEPROM read start address copy	2 bytes	—	EEPROM read routine
num	Number of receive data	1 byte	H'0A	EEPROM read routine
r_data	Pointer variable to receive data array variable read_data	4 bytes	—	EEPROM read routine
dummy	Dummy read data	1 byte	—	EEPROM read routine
ack	Acknowledge reception determination flag	1 byte	H'01	EEPROM read routine

(4) RAM Used

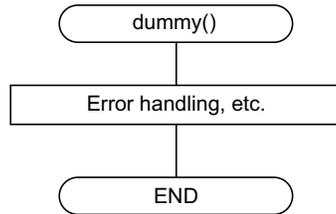
This sample task does not use any RAM apart from the variables.

2.4 Flowcharts

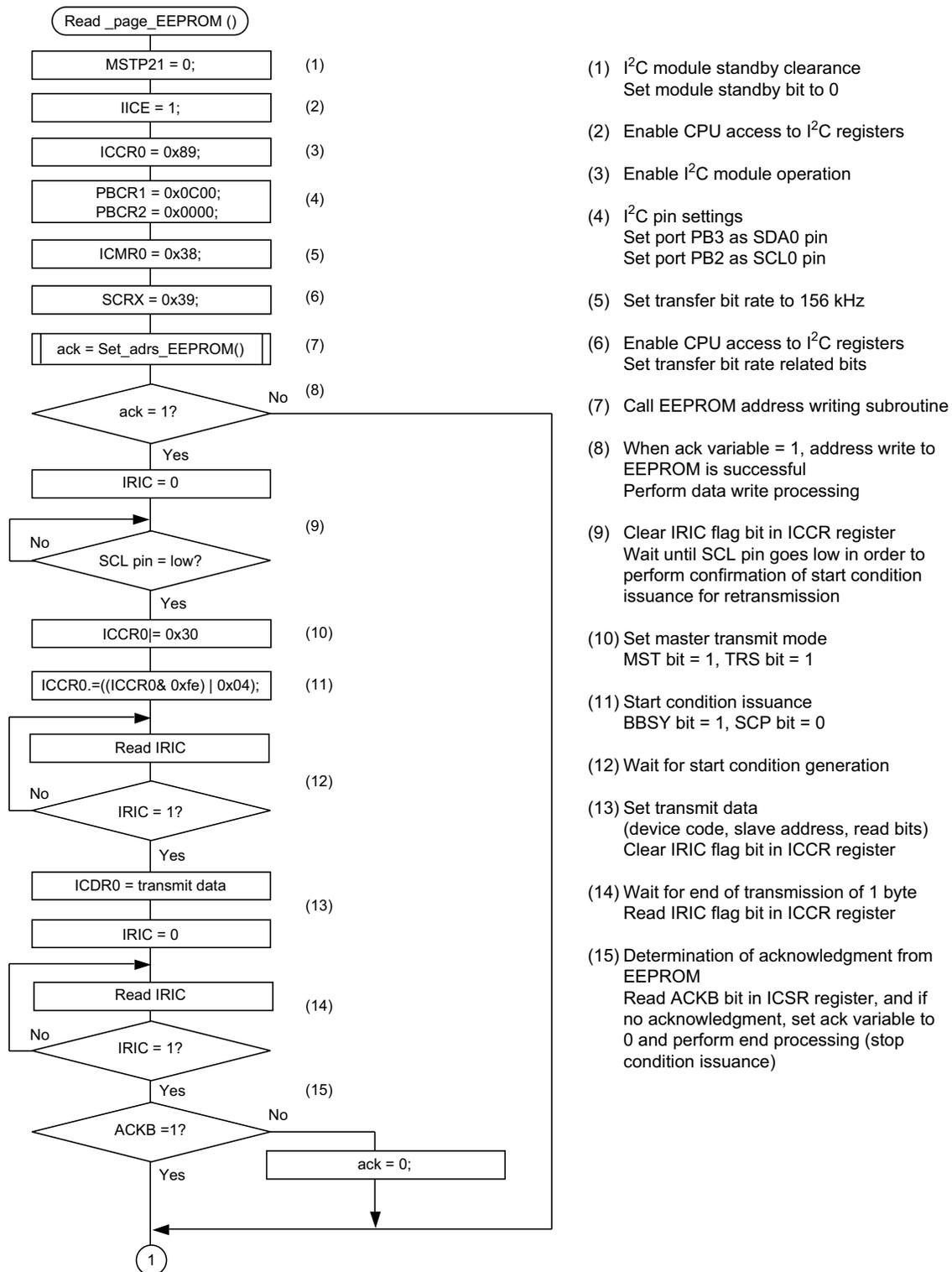
(1) Main routine



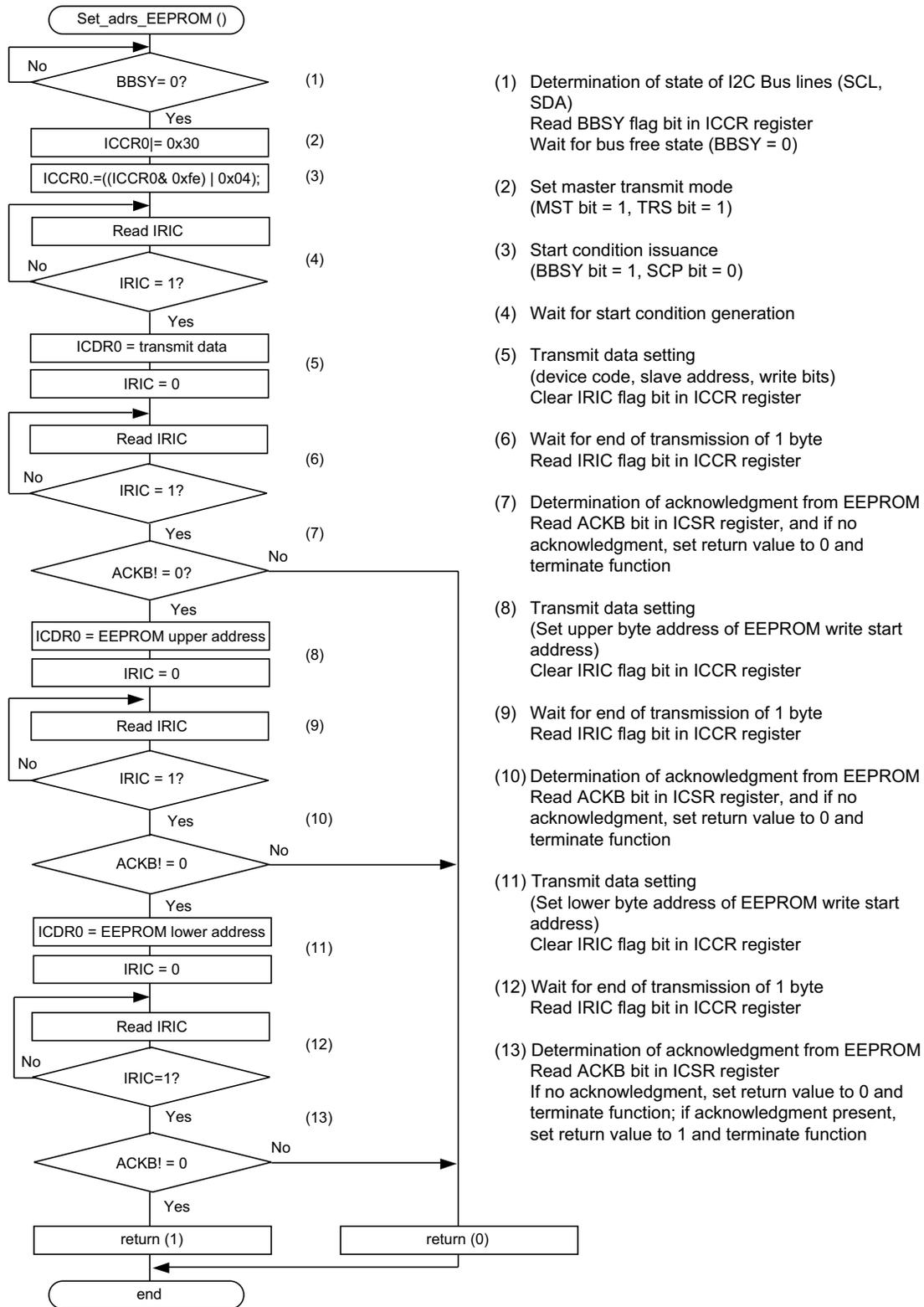
(2) Dummy interrupt routine



(3) EEPROM read subroutine



(4) Start condition issuance, slave address and EEPROM memory address transmission subroutine



2.5 Program Listing

```

//*****
//  SH7144F Group -SH7145-      I2C-bus Application Note
//      Single master receive
//      n Byte data write/read 64kbit EEPROM
//      Clock :CPU=40MHz  (External input=10MHz)
//      :Peripheral=40MHz
//      I2c bit rate:156kHz
//      Written   :2003/2/1  Rev.2.0
//*****
#include <machine.h>
#include "iodefine.h"

//----- Symbol Definition -----
#define DEVICE_CODE    0xa0      // EEPROM DEVICE CODE:b'1010
#define SLAVE_ADRS    0x00      // SLAVE ADRS:b'000
#define IIC_DATA_W    0x00      // WRITE DATA:b'0
#define IIC_DATA_R    0x01      // READ DATA:b'1
#define DATA_NUM     10        // data size

//----- Function Definition -----
void main(void);
void dummy(void);

unsigned char Read_page_EEPROM(unsigned short,unsigned char*,unsigned char);
unsigned char Set_adrs_EEPROM(unsigned short);

//*****
//      main
//*****
void main(void)
{
    unsigned short  address;      // EEPROM memory address
    unsigned char   read_data[DATA_NUM]; // read data

    address= 0x0000;              // set EEPROM address

    // EEPROM data read
    Read_page_EEPROM(address,read_data,DATA_NUM);

    while(1);
}

```

```

//*****
//  dummy interrput function
//*****
#pragma interrupt(dummy)
void dummy(void)
{
    //  Interrput error
}

//*****
//
//      Read_page_EEPROM
//      argument1  ;read address(unsigned short)
//      argument2  ;read data(unsigned char)
//      argument2  ;read data number (unsigned char)
//      return     ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
//*****
unsigned char Read_page_EEPROM(unsigned short adrs,unsigned char* r_data, unsigned char
num)
{
    unsigned char ack;           // ACK flag
    unsigned char count;        // read data number
    unsigned char dummy;        // dummy data

    // Set standby mode
    P_STBY.MSTCR1.BIT.MSTP21 = 0; // disable I2C standby mode

    ack =1;
    P_IIC.SCRX.BIT.IICE = 1;      // Enables CPU access to the register
    P_IIC.ICCR0.BYTE = 0x89;
        // ICE(7)=b'1      Enable I2C bus interface
        // IEIC(6)=b'0     Disables the interrupt
        // MST(5)=b'0      Slave mode
        // TRS(4)=b'0      Receive mode
        // ACKE(3)=b'1     Continuous data transfer is halted
        // BBSY(2)=b'0
        // IRIC(1)=b'0
        // SCP(0)=b'1      Start/stop condition issuance disabling

    // set I2C pin function
    P_PORTB.PBCR1.WORD = 0x0c00; // SDA0(PB3-32pin@SH7145F),SCL0(PB2-31pin@SH7145F)
    P_PORTB.PBCR2.WORD = 0x0000;

    P_IIC.ICMR0.BYTE = 0x38;
        // MILS(7)=b'0     MSB first
        // WAIT(6)=b'0     A wait state is inserted between DATA and ACK
        // CKS2[2:0](5:3)=b'111 Transfer clock select
        //                  156kHz@(@P-fai40MHz,IICX=1)
        //                  39.1kHz@(@P-fai10MHz,IICX=1)

    P_IIC.SCRX.BYTE = 0x39;
        // IICX(5)=b'1     transfer-rate select,reference CKS bit
        // IICE(4)=b'1     Enables CPU access to the register
        // HNDS(3)=b'1     Set this bit to 1

```

```

// STOPIM(0)=b'1           disables interrupt requests

// Set device code,EEPROM address
ack = Set_adrs_EEPROM(adrs);           // set device code,EEPROM address

if( ack==1){

    P_IIC.ICCR0.BIT.IRIC = 0;           // clear IRIC
    while(P_PORTB.PBDR.BIT.PB2DR!=0);   // check SCL0 pin state == low?

    // Master-Transmission,Generate the start condition.
    P_IIC.ICCR0.BYTE |= 0x30;           // Select master transmit mode(MST=1,TRS=1)
    P_IIC.ICCR0.BYTE=((P_IIC.ICCR0.BYTE & 0xfe)|0x04);
                                           // Generate start condition(BBSY=1,SCP=0)
    while( P_IIC.ICCR0.BIT.IRIC==0 );   // Wait for a start condition generation

    // Slave address+R Transmission
    P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_R);
                                           // data set
    P_IIC.ICCR0.BIT.IRIC = 0;           // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 );   // Wait 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){       // Test the acknowledge bit
        ack = 0;                         // no ACK
    }
}

if( ack==1 ){
    // Master receive operation (HNDS=1,WAIT=0)
    P_IIC.ICCR0.BIT.TRS = 0;           // Select receive mode(TRS=0)
    P_IIC.ICMR0.BIT.WAIT = 0;          // set wait=0
    P_IIC.ICSR0.BIT.ACKB = 0;          // set ACK data =0
    P_IIC.SCRX.BIT.HNDS = 1;           // set HNDS bit =1

    P_IIC.ICCR0.BIT.IRIC = 0;           // clear IRIC

    // Start data receiving
    if(num>1){                           // case nByte data read (n>1)
        dummy = P_IIC.ICDR0.BYTE;       // dummy read
        while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for 1 byte to be received
        P_IIC.ICCR0.BIT.IRIC = 0;       // clear IRIC
        for( count=2; count<num; count++ ){ // (num-2)byte read
            *r_data = P_IIC.ICDR0.BYTE; // read receive data
            while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for 1 byte to be received
            P_IIC.ICCR0.BIT.IRIC = 0;    // clear IRIC
            r_data++;
        }
    }
}

```

```

P_IIC.ICSR0.BIT.ACKB = 1;           // set ACK data =1

if(num==1){                         // case 1Byte read
    dummy = P_IIC.ICDR0.BYTE;       // dummy read
}else{                                // case nByte data read (n>1)
    *r_data = P_IIC.ICDR0.BYTE;     // read receive data(n-1)
    r_data++;
}
while( P_IIC.ICCR0.BIT.IRIC==0 );   // Wait for 1 byte to be received
P_IIC.ICCR0.BIT.IRIC = 0;           // clear IRIC

// End data receiving
P_IIC.ICCR0.BIT.TRS = 1;            // Select transmit mode
*r_data = P_IIC.ICDR0.BYTE;         // read END receive data
}

// Stop condition issuance
P_IIC.ICCR0.BYTE = P_IIC.ICCR0.BYTE & 0xfa;
// Stop condition issuance(BBSY=0,SCP=0)

return(ack);
}

/*****
//      Set_adrs_EEPROM
//      argument1      ;write address(unsigned short)
//      return         ;1=OK/0=NG EEPROM NO_ACK(unsigned char)
/*****
unsigned char Set_adrs_EEPROM(unsigned short adrs)
{
    while( P_IIC.ICCR0.BIT.BBSY!=0 ); // BUS FREE?(BBSY==0→Bus Free)

    // Master-Transmission,Generate the start condition.
    P_IIC.ICCR0.BYTE |= 0x30;         // Select master transmit
mode(MST=1,TRS=1)

    P_IIC.ICCR0.BYTE=((P_IIC.ICCR0.BYTE & 0xfe) | 0x04);
// Generate start condition(BBSY=1,SCP=0)
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait for a start condition generation

    // Slave address+W Transmission
    P_IIC.ICDR0.BYTE = (unsigned char)(DEVICE_CODE|SLAVE_ADRS|IIC_DATA_W);
// data set
    P_IIC.ICCR0.BIT.IRIC = 0;         // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
    if( P_IIC.ICSR0.BIT.ACKB!=0 ){    // Test the acknowledge bit
        return (0);                  // no ACK
    }
    // EEPROM upper address Transmission(1byte)
    P_IIC.ICDR0.BYTE = (unsigned char)(adrs>>8); // data set
    P_IIC.ICCR0.BIT.IRIC = 0;         // clear IRIC
    while( P_IIC.ICCR0.BIT.IRIC==0 ); // Wait 1byte transmitted
}

```

```
if( P_IIC.ICSR0.BIT.ACKB!=0 ){           // Test the acknowledge bit
    return (0);                          // no ACK
}
// EEPROM lower address Transmission(1byte)
P_IIC.ICDR0.BYTE = (unsigned char)(adrs & 0x00ff); // data set
P_IIC.ICCR0.BIT.IRIC = 0;                // clear IRIC
while( P_IIC.ICCR0.BIT.IRIC==0 );        // Wait 1byte transmitted
if( P_IIC.ICSR0.BIT.ACKB!=0 ){           // Test the acknowledge bit
    return (0);                          // no ACK
}
return (1);                              // ACK OK
}
```

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors.
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.