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.

SH7080 Group

I²C2 Single-Master Reception (Reading from I²C-Bus EEPROM)

Introduction

This application note describes the receiving operation by the I²C bus interface 2 module (I²C2) in single-master mode. Please use this application note as a guide in designing user programs.

Although the programs given in this application note have been verified for correct operation, we strongly recommend that the user confirm correct operation before applying the programs in the actual application.

Target Device

SH7085

Contents

1. Specifications

- The SH7085's I²C bus interface 2 module (I²C2) is used to read 10 bytes of data from a two-wire serial EEPROM (HN58X2416; 16k bits; 2k words × 8 bits).
- The connection is made in single-master configuration, in which the SH7085 is the master device.
- The device code of the connected EEPROM is B'1010.
- Before reading data from the EEPROM, the EEPROM bus state is initialized.*
- Data is read from addresses H'0000 to H'0009 in the EEPROM.
- After 1 byte of data has been received, the acknowledge bit (ACK = 0) is output. (However, ACK = 1 is output after reception of the last byte.)
- The frequency of the I²C bus data transfer clock is 400 kHz.

Figure 1 shows an example of connection between the SH7085 and EEPROM, and table 1 shows the SH7085's I²C2 settings. Table 2 shows the device address word of the EEPROM used in the sample task.

Note: The EEPROM's SDA bus is forcibly placed in the input state, anticipating the case when the EEPROM is unable to receive from the master device because of the EEPROM's SDA bus remaining in the output state as a result of, for example, communication having been halted while the master device is receiving data from the EEPROM.

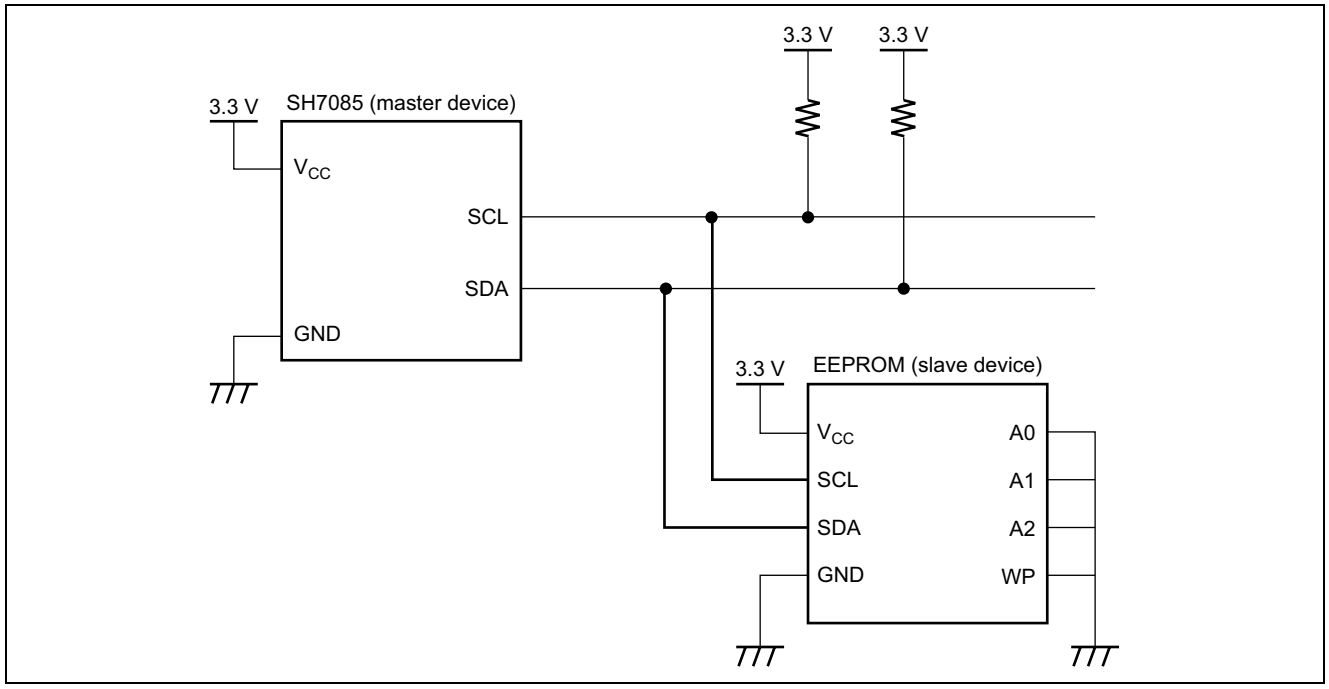


Figure 1 Example of Connection between SH7085 and EEPROM

Table 1 I²C2 Settings of SH7085

Item	Setting
Operating mode	Master reception mode
Transfer clock	400 kHz (Pϕ = 40 MHz)
Data bit length	9 bits (including the ACK bit)
Wait cycles inserted between data and ACK	None
Interrupt	None
ACK output	Outputs 1 only when the last data is received.
	Outputs 0 while data is received continuously.

Table 2 Device Address Word of EEPROM

Device Code				Device Address Code			R/W Code
1	0	1	0	a10	a9	a8	R = 1, W = 0

Note: a10 to a8 are the upper 3 bits of the EEPROM address.

2. Conditions for Application

Operating frequency:	Internal clock:	80 MHz
	Bus clock:	40 MHz
	Peripheral clock:	40 MHz
	MTU2 clock:	40 MHz
	MTU2S clock:	80 MHz
C compiler:	Version 7.1.04 from Renesas Technology Corp.	

3. Summary of MCU Functions Used

In this sample task, the I²C bus (Inter IC bus) is used to read data from the EEPROM.

3.1 I²C Bus Interface 2 (I²C2)

The I²C bus interface 2 (I²C2) conforms to the I²C bus interface specifications set up by Philips, and provides a subset of the I²C functions. Figure 2 shows a block diagram of the I²C2 module.

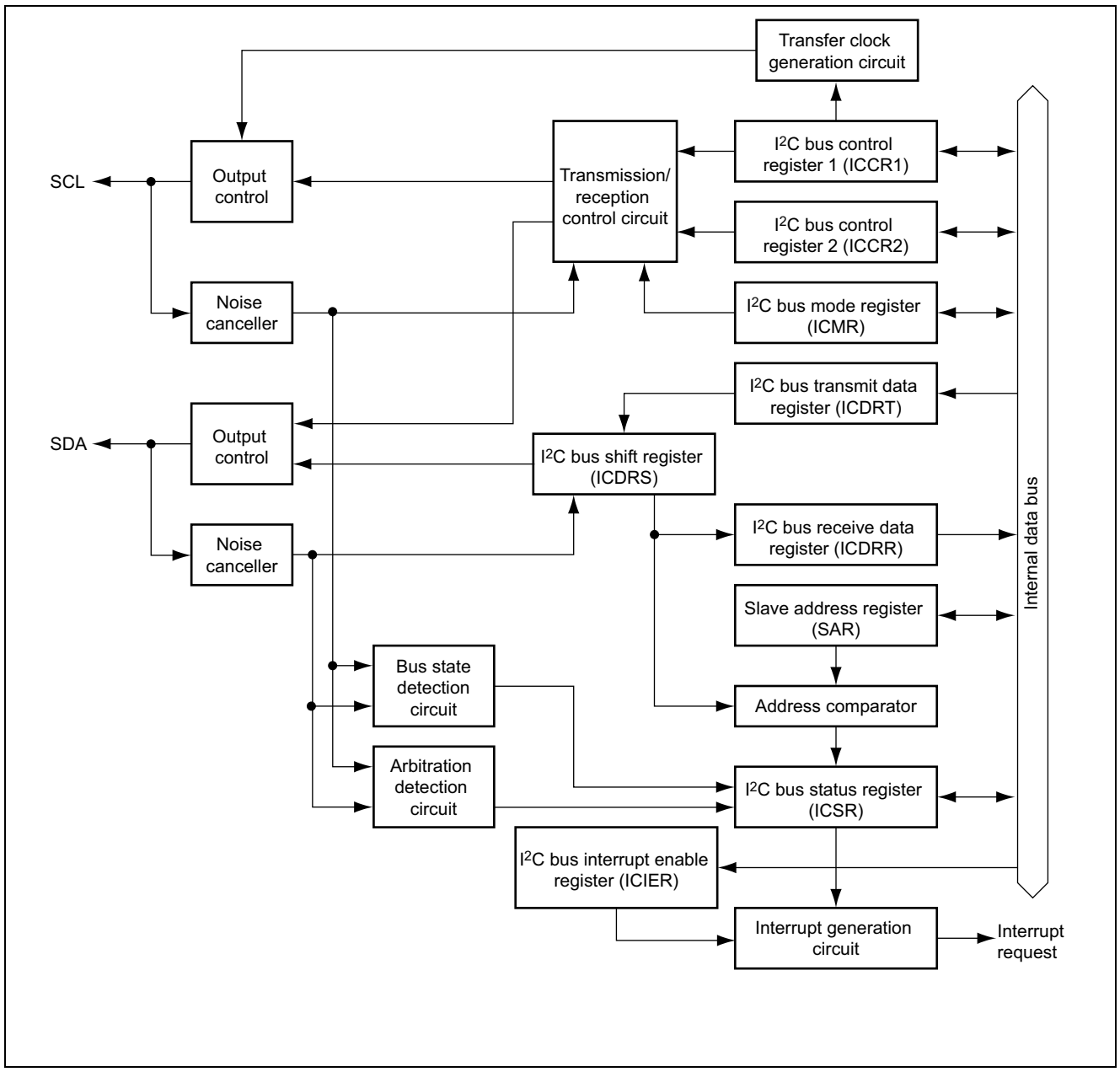


Figure 2 Block Diagram of I²C2 Module

- The I²C bus control register 1 (ICCR1) sets I²C2 operations.
- The I²C bus control register 2 (ICCR2) generates start and stop conditions, drives the SDA pin, monitors the SCL pin, and controls resetting of the I2C bus control circuitry.
- The I²C bus mode register (ICMR) selects MSB- or LSB-first transfer, controls the wait states in master mode, and selects the number of bits to be transferred.
- The I²C bus interrupt enable register (ICIER) enables interrupts, selects the use of the acknowledge bit, sets the acknowledge bit for transmission, and checks the received acknowledge bit.
- The I²C bus status register (ICSR) consists of various interrupt request and status flags.
- The I²C slave address register (SAR) selects a format and sets the slave address.
- The I²C bus transmit data register (ICDRT) holds data for transmission.
- The I²C bus receive data register (ICDRR) holds the received data.
- The I²C bus shift register (ICDRS) is used to transmit/receive data; this register cannot be accessed by the CPU.

4. Operation

In this sample task, the EEPROM bus state is initialized before data is read from the EEPROM.

4.1 EEPROM Bus Initialization

The EEPROM bus initialization here means to forcibly place the EEPROM's SDA bus into the input state. This initialization should be applied when the EEPROM is unable to receive from the master device because of the EEPROM's SDA bus remaining in the output state as a result of, for example, communication having been halted while the master device is receiving data from the EEPROM.

In the initialization processing, the SCL and SDA bus lines are driven by using the port I/O (general-purpose I/O) function. After generating a start condition, the MCU outputs dummy data (H'FF), receives acknowledge (ACK) from the EEPROM, and then generates a stop condition.

Figure 3 shows the initialization processing, and the software and hardware processing in the figure are described in table 3.

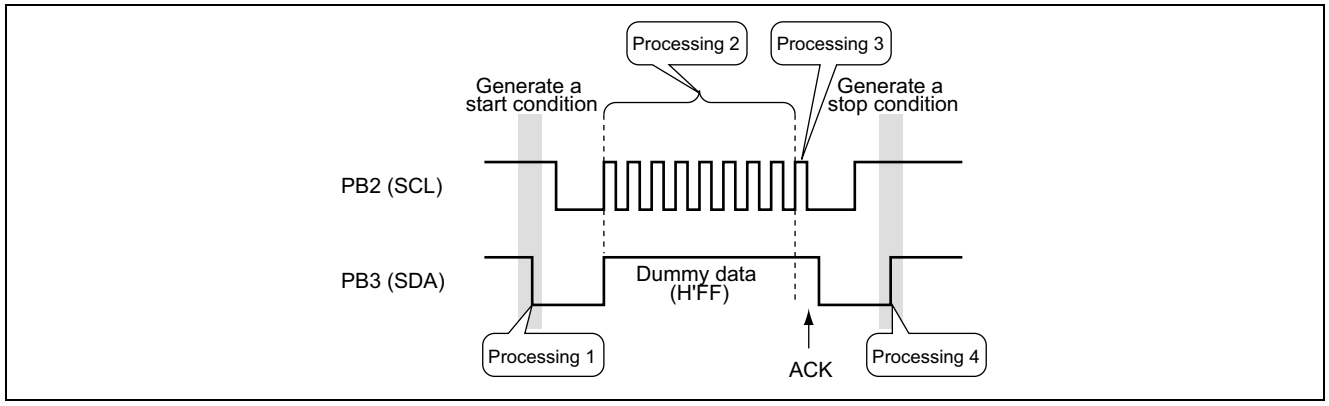


Figure 3 EEPROM Bus Initialization Processing

Table 3 Software and Hardware Processing in Bus Initialization

	Software Processing	Hardware Processing
Processing 1	• Set PB2 (SCL) and PB3 (SDA) as port (general input/output) pins. • Set the output level on PB2 and PB3 to high. • Set the output level on PB3 to low to generate a start condition. • Set the output level on PB2 to low.	—
Processing 2	• Output the transfer clock by changing the output level on PB2 with use of a software wait timer. • Fix the output level on PB3 to high to transmit dummy data (H'FF).	—
Processing 3	• Output the transfer clock (for receiving the acknowledge bit) from PB2. • Set PB3 as an input pin. • Read the state on the PB3 pin.	—
Processing 4	• Set the output level on PB2 and PB3 to low. • Set the output level on PB3 to high to generate a stop condition.	—

4.2 Reading Data from EEPROM

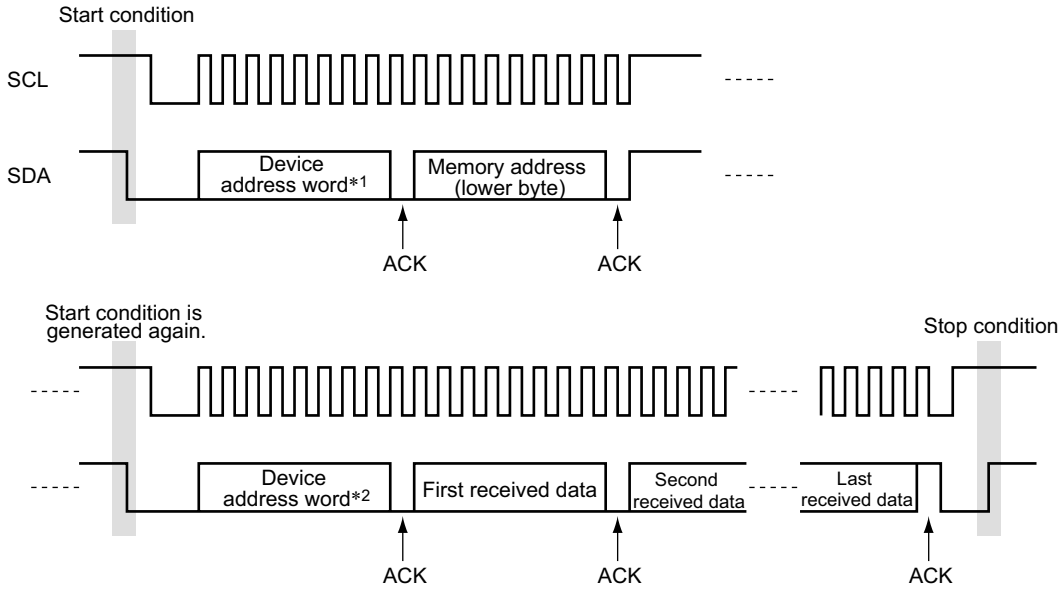
In this sample task, the address (H'0000) is specified by Random Reading, and then 10 bytes of data are continuously read by Sequential Read.

Figure 4 shows the contents of communication when data are read from the EEPROM.

After a start condition is generated, the device address word is transmitted with the R/W code set to 0 (specifying write operation). The lower byte of the EEPROM read start address is then transmitted.

A start condition is generated again to start reading. After the second generation of a start condition, the device address word is transmitted with the R/W code set to 1 (specifying read operation). This makes data to be output from the EEPROM sequentially in accordance with the I²C format.

The master outputs the acknowledge bit (ACK) on receiving one byte of data. The EEPROM outputs the next byte on receiving ACK = 0. The master outputs ACK = 1 on receiving the last byte and generates a stop condition.



Notes: 1. Slave address + upper 3 bits of memory address + R/W code (0): Specifies write operation.
2. Slave address + upper 3 bits of memory address + R/W code (1): Specifies read operation.

Figure 4 Contents of Communication When Data are Read from EEPROM

Figure 5 shows the operation when data reading from the EEPROM is started (start condition generated again), and figure 6 shows the ending operation. The software and hardware processing in figures 5 and 6 are described in tables 4 and 5, respectively.

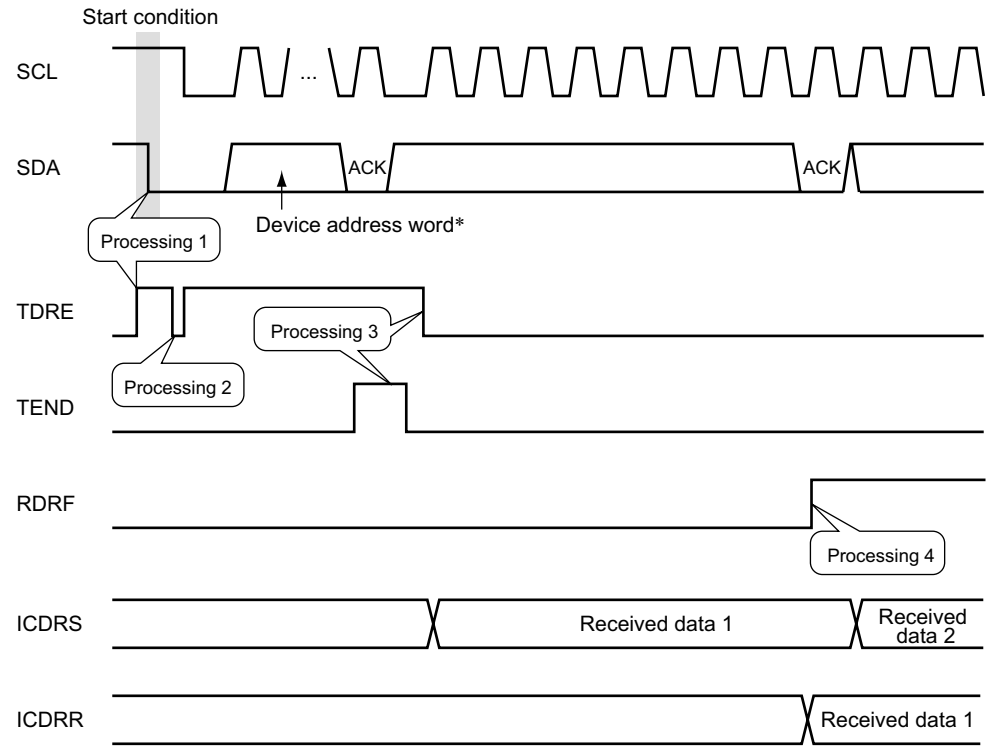


Figure 5 Operation when Reading from EEPROM Starts

Table 4 Software and Hardware Processing when Reading from EEPROM Starts

	Software Processing	Hardware Processing
Processing 1	Set BBSY to 1 and SCP to 0 in the ICCR2 register.	• Generate a start condition (second time). • Set the TDRE bit in the ICSR register to 1.
Processing 2	Write the device address word data to the ICDRT register (specifying read operation).	• Clear the TDRE bit in the ICSR register to 0. • Transfer the transmit data in the ICDRT register to the ICDRS register, and then output the data from the SDA pin. • Set the TDRE bit in the ICSR register to 1.
Processing 3	• Confirm that TEND in the ICSR register is 1. • Check the received acknowledge through the ACKBR bit in the ICIER register. • Clear the TEND bit in the ICSR register to 0. • Clear the TRS bit in the ICCR1 register to 0 (master reception mode). • Clear the TDRE bit in the ICSR register to 0. • Dummy-read the ICDRR register (starting reception).	• Set the TEND bit in the ICSR register to 1 after outputting the last bit of the transmit data. • Output the clock for reception by dummy-reading the ICDRR register.
Processing 4	• Confirm that RDRF in the ICSR register is 1. • Read the ICDRR register.	• Output the acknowledge bit after receiving the last bit. • Transfer the received data in the ICDRS register to the ICDRR register. • Set the RDRF bit in the ICSR register to 1. • Clear the RDRF bit in the ICSR register to 0 after the ICDRR register is read.

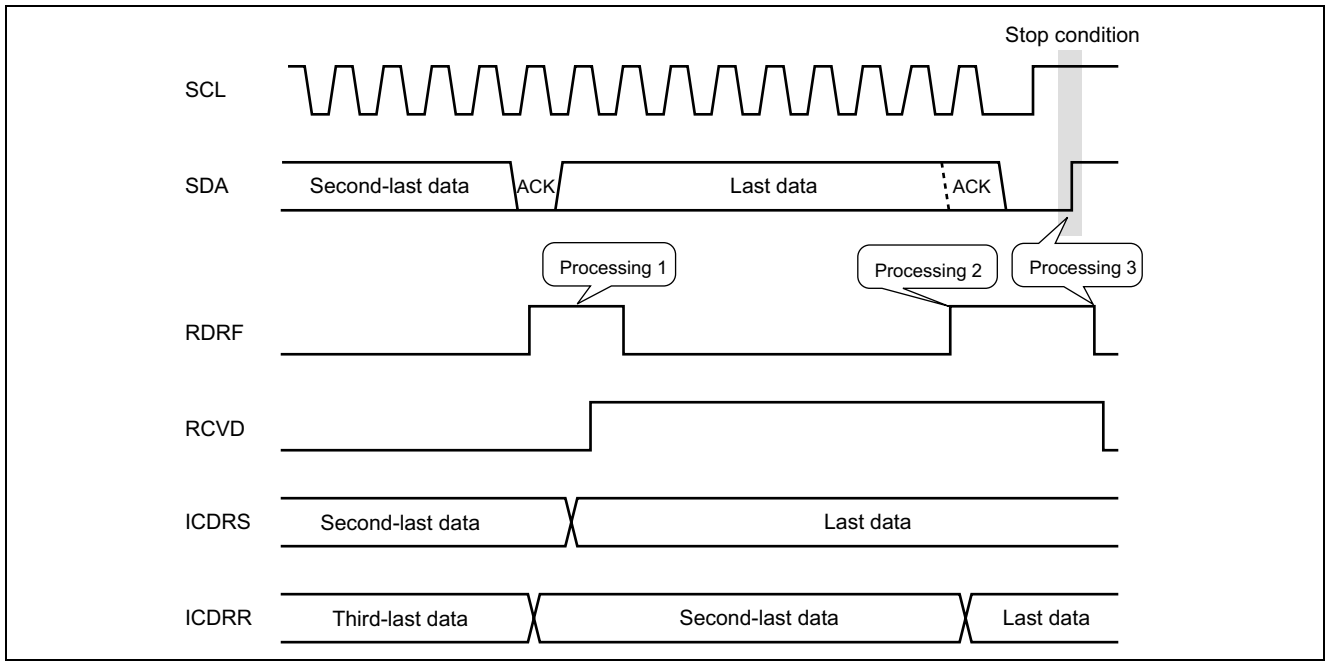


Figure 6 Operation when Reading from EEPROM Ends

Table 5 Software and Hardware Processing when Reading from EEPROM Ends

	Software Processing	Hardware Processing
Processing 1	• Confirm that RDRF in the ICSR register is 1. • Set the ACKBT bit in the ICIER register to 1. • Set the RCVD bit in the ICCR1 register to 1. • Read the ICDRR register.	• Transfer the received data in the ICDRS register to the ICDRR register. • Set the RDRF bit in the ICSR register to 1. • Clear the RDRF bit in the ICSR register to 0 after the ICDRR register is read.
Processing 2	• Confirm that RDRF in the ICSR register is 1. • Clear the STOP bit in the ICSR register to 0.	• Transfer the received data in the ICDRS register to the ICDRR register. • Set the RDRF bit in the ICSR register to 1.
Processing 3	• Set BBSY and SCP in the ICCR2 register to 0. • Confirm that STOP in the ICSR register is 1. • Read the ICDRR register (last data). • Clear the RCVD bit in the ICCR1 register to 0. • Set MST and TRS in the ICCR1 register to 0 (slave reception mode).	• Generate a stop condition. • Set the STOP bit in the ICSR register to 1.

5. Description of Software

5.1 Modules

Table 6 describes the modules used in the sample application.

Table 6 Description of Modules

Module Name	Label Name	Description
Main function	main()	Sets the EEPROM read start address, and calls the data read function.
I²C2 initialization function	init_iic()	Cancels module standby mode, and configures the PFC and I²C2.
Data read function	read_EEPROM()	Generates a start condition, reads data from the EEPROM, and generates a stop condition.
EEPROM address transmission function	set_addr_EEPROM()	Generates a start condition and a slave address, and sets the EEPROM address.
EEPROM bus initialization function	init_EEPROM()	Forcibly initializes the SDA bus that has hanged up due to a communication error or other reason.
I²C2 reset function	reset_iic()	Resets and halts the I²C2 module.
Start condition generation function	iic_start()	Generates a start condition through port processing.
Output level set function	iic_sda_out()	Sets the output level of the bus line.
Transmit data set function	iic_set()	Outputs the clock and transmit data through port processing.
1-byte transmission function	iic_bytesend()	Transmits 1 byte of data through port processing.
Acknowledge reception function	iic_ackck()	Receives the acknowledge from the EEPROM through port processing.
Stop condition generation function	iic_stop()	Generates a stop condition through port processing.
Software wait function	wait_timer()	Wait timer using a software counter

5.2 Variables

Table 7 lists the variables used in the sample application.

Table 7 Description of Variables

Label Name of Variable	Function	Used In
read_data[0:9]	Array for storing read data	Main function
address	Read start address of the EEPROM	Main function
addr	Copy of read start address of the EEPROM	Data read function EEPROM address transmission function
*r_data	Pointer variable to the array for storing read data	Data read function
num	Number of received data bytes	Data read function
ack	Acknowledge determination flag	Data read function
count	Counter for continuous data reading	Data read function
dummy	Variable for dummy reading	Data read function
data	Information of output levels on the PB2 (SCL) and PB3 (SDA) pins	Output level set function
scl	Information of output levels on the SCL line	Transmit data set function
sda	Information of output levels on the SDA line	Transmit data set function
tx_data	One byte of data for transmission through port processing	1-byte transmission function
ck_bit	Clock information for data transmission through port processing	1-byte transmission function
bit_data	One bit of data for transmission through port processing	1-byte transmission function
ack_flag	Acknowledge bit received through port processing	Acknowledge reception function
wait_cnt	Count value for software timer	Software wait function
cnt	Variable for software counter	Software wait function

5.3 Register Settings

The register settings used in the sample application are described below. Note that the set values are specifically used in the sample task and that they are different from the initial values.

5.3.1 Settings for the Clock Pulse Generator (CPG)

- Frequency Control Register (FRQCR)
  - Set value: H'0241
  - Function: Specifies the frequency division ratio.

Bit	Bit Name	Value	Description
15	—	0	Reserved
14 to 12	IFC[2] to IFC[0]	000	Frequency division ratio for internal clock (Iϕ) 000: ×1 (80 MHz when input clock is 10 MHz)
11 to 9	BFC[2] to BFC[0]	001	Frequency division ratio for bus clock (Bϕ) 001: ×1/2 (40 MHz when input clock is 10 MHz)
8 to 6	PFC[2] to PFC[0]	001	Frequency division ratio for peripheral clock (Pϕ) 001: ×1/2 (40 MHz when input clock is 10 MHz)
5 to 3	MIFC[2] to MIFC[0]	000	Frequency division ratio for MTU2S clock (MIϕ) 000: ×1 (80 MHz when input clock is 10 MHz)
2 to 0	MPFC[2] to MPFC[0]	001	Frequency division ratio for MTU2 clock (MPϕ) 001: ×1/2 (40 MHz when input clock is 10 MHz)

5.3.2 Settings for Power-Down Modes

- Standby Control Register 3 (STBCR3)
  - Set value: H'7F
  - Function: Controls the operation of individual modules in power-down modes.

Bit	Bit Name	Value	Description
7	MSTP15	0	0: The I²C2 runs.
6	MSTP14	1	1: Stops supply of the clock signal to the SCIF.
5	MSTP13	1	1: Stops supply of the clock signal to SCI_2.
4	MSTP12	1	1: Stops supply of the clock signal to SCI_1.
3	MSTP11	1	1: Stops supply of the clock signal to SCI_0.
2	MSTP10	1	1: Stops supply of the clock signal to the SSU.
1 and 0	—	11	Reserved

5.3.3 Settings for I²C Bus Interface 2 (I²C2)

- I²C Bus Control Register 1 (ICCR1)
  - Set value: H'B5
  - Function: Selects the operating mode and transfer clock of the I²C2.

Bit	Bit Name	Value	Description
7	ICE	1	1: Enables transfer operation (the SCL/SDA buses are driven).
6	RCVD	0	0: Performs the next reception when ICDRR is read.
5	MST	1	MST: Master/Slave Select
4	TRS	1*	TRS: Transmit/Receive Select 10: Master reception mode 11: Master transmission mode.
3 to 0	CKS[3] to CKS[0]	0101	Transfer Clock Select 0101: Sets the transfer rate to 400 kHz (Pϕ = 40 MHz).

Note: * Changed to 0 (reception) after an address is transmitted to the EEPROM.

- I²C Bus Control Register 2 (ICCR2)
  - Setting value: H'7D
  - Function: Generates start and stop conditions, drives the SDA pin, monitors the SCL pin, and controls resetting of the I²C bus control circuitry.

Bit	Bit Name	Value	Description
7	BBSY	0*	Indicates whether the I²C bus is occupied or released, and generates start and stop conditions. 0: I²C bus is released.
6	SCP	1*	Controls generation of start and stop conditions.
5	SDAO	1	1: The output on the SDA pin is high (when read). Changes the output on the SDA pin to Hi-Z (when written).
4	SDAOP	1	SDAO Write Protect Write 0 to this bit when writing to SDAO.
3	SCLO	1	1: The output on the SCL pin is high (read-only bit).
2	—	1	Reserved
1	IICRST	0	I²C Control Unit Reset Resets the control circuitry of the I²C when 1 is written.
0	—	1	Reserved

Note: * To generate a start condition, write b'10 to BBSY and SCP.
To generate a stop condition, write b'00 to BBSY and SCP.

- I²C Bus Mode Register (ICMR)
  - Setting value: H'38
  - Function: Selects MSB- or LSB-first transfer, controls the wait states in master mode, and selects the number of bits to be transferred.

Bit	Bit Name	Value	Description
7	MLS	0*[1]	0: MSB-first
6	WAIT	0*[2]	0: Data and acknowledge bit are transferred continuously.
5 and 4	—	11	Reserved
3	BCWP	1	BC Write Protect Write 0 to this bit when writing to the BC[2] to BC[0] bits.
2 to 0	BC[2] to BC[0]	000	000: 9 bits (data and acknowledge bit) are transferred.

Notes: 1. This bit must be set to 0 (MSB-first) when the I²C bus format is used.
2. This bit must be set to 0.

- I²C Bus Interrupt Enable Register (ICIER)
  - Setting value: H'04
  - Function: Enables or disables interrupt sources and controls the acknowledge bit.

Bit	Bit Name	Value	Description
7	TIE	0	0: Disables the transmit data empty interrupt request (IITXI).
6	TEIE	0	0: Disables the transmit end interrupt request (IITEI).
5	RIE	0	0: Disables the receive data full interrupt request (IIRXI).
4	NAKIE	0	0: Disables the NACK receive interrupt request (IINAKI).
3	STIE	0	0: Disables the stop condition detection interrupt request (IISTPI).
2	ACKE	1	1: Halts data transfer when the received acknowledge bit is 1.
1	ACKBR	0	Holds the acknowledge data in transmission mode (read-only bit).
0	ACKBT	0	0: Transmits ACK = 0 in reception mode.

- I²C Bus Status Register (ICSR)
  - Setting value: H'00
  - Function: Provides various interrupt request and status flags.

Bit	Bit Name	Value	Description
7	TDRE	0	Transmit data register empty flag
6	TEND	0	Transmit end flag
5	RDRF	0	Receive data register full flag
4	NACKF	0	No acknowledge detection flag
3	STOP	0	Stop condition detection flag
2	AL/OVE	0	Arbitration lost flag/overrun error flag
1	AAS	0	Slave address recognition flag
0	ADZ	0	General call address recognition flag

- I²C Bus Transmit Data Register (ICDRT)
  - Setting value: H'FF (initial value)
  - Function: Holds data for transmission.
- I²C Bus Receive Data Register (ICDRR)
  - Setting value: H'FF
  - Function: Holds the received data. (Read only)

5.3.4 Settings for the Pin Function Controller (PFC)

- Port B Control Register L1 (PBCRL1)
  - Setting value: H'4400
  - Function: Selects the functions of the multiplexed pins on port B (PB3 to PB0).

Bit	Bit Name	Value	Description
15	—	0	Reserved
14 to 12	PB3MD[2] to PB3MD[0]	100*	PB3 mode 100: SDA I/O (I²C2)
11	—	0	Reserved
10 to 8	PB2MD[2] to PB2MD[0]	100*	PB2 mode 100: SCL I/O (I²C2)
7	—	0	Reserved
6 to 4	PB1MD[2] to PB1MD[0]	000	PB1 mode 000: PB1 I/O (port)
3	—	0	Reserved
2 to 0	PB0MD[2] to PB0MD[0]	000	PB0 mode 000: PB1 I/O (port)

Note: * In EEPROM bus initialization, these bits are set to 000 to use the multiplexed pins as general I/O (port) pins.

- Port B IO Register (PBIORL)
  - Set value: H'000C
  - Function: Selects the signal directions on the port B pins.

Bit	Bit Name	Value	Description
15 to 10	—	All 0	Reserved
9	PB9IOR	0	0: PB9 is input.
8	PB8IOR	0	0: PB8 is input.
7	PB7IOR	0	0: PB7 is input.
6	PB6IOR	0	0: PB6 is input.
5	PB5IOR	0	0: PB5 is input.
4	PB4IOR	0	0: PB4 is input.
3	PB3IOR	1	1: PB3 is output (ignored when SDA is selected by PBCRL1).
2	PB2IOR	1	1: PB2 is output (ignored when SCL is selected by PBCRL1).
1	PB1IOR	0	0: PB1 is input.
0	PB0IOR	0	0: PB0 is input.

Note: This setting is used in EEPROM bus initialization.

5.3.5 Settings for I/O Ports

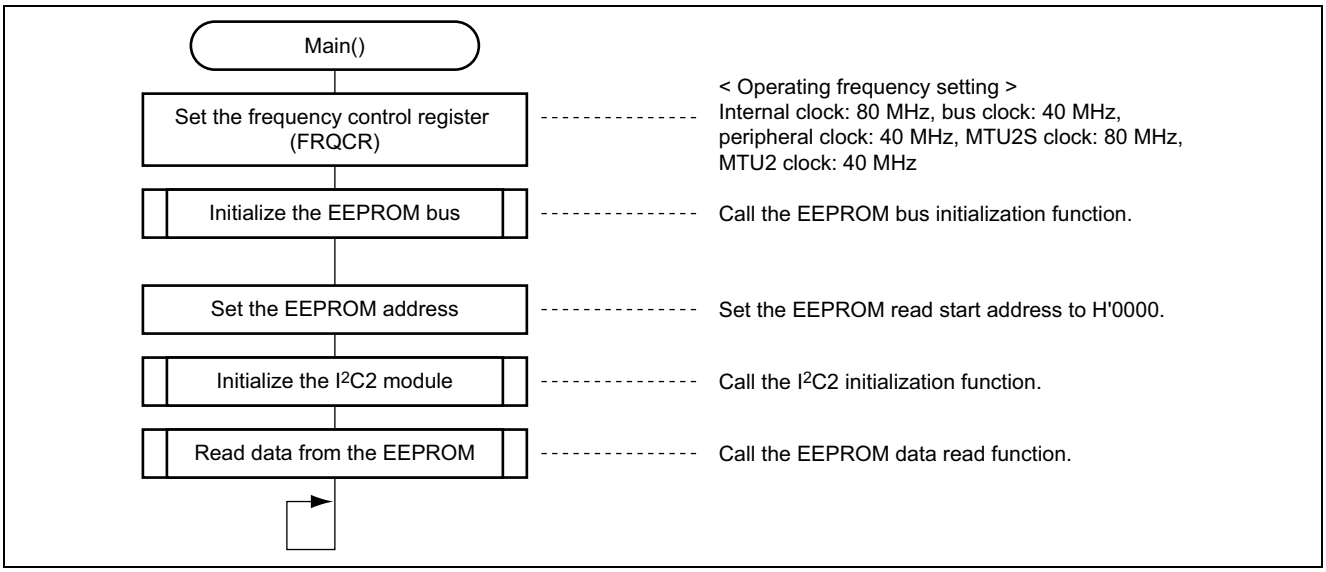
- Port B Data Register L (PBDRL)
  - Set value: H'000C
  - Function: Holds the data for port B.

Bit	Bit Name	Value	Description
15 to 10	—	All 0	Reserved
9	PB9DR	0	0: Low level on PB9
8	PB8DR	0	0: Low level on PB8
7	PB7DR	0	0: Low level on PB7
6	PB6DR	0	0: Low level on PB6
5	PB5DR	0	0: Low level on PB5
4	PB4DR	0	0: Low level on PB4
3	PB3DR	1	1: High level on PB3 (Produces a transmit data signal in EEPROM bus initialization.)
2	PB2DR	1	1: High level on PB2 (Produces a clock signal in EEPROM bus initialization.)
1	PB1DR	0	0: Low level on PB1
0	PB0DR	0	0: Low level on PB0

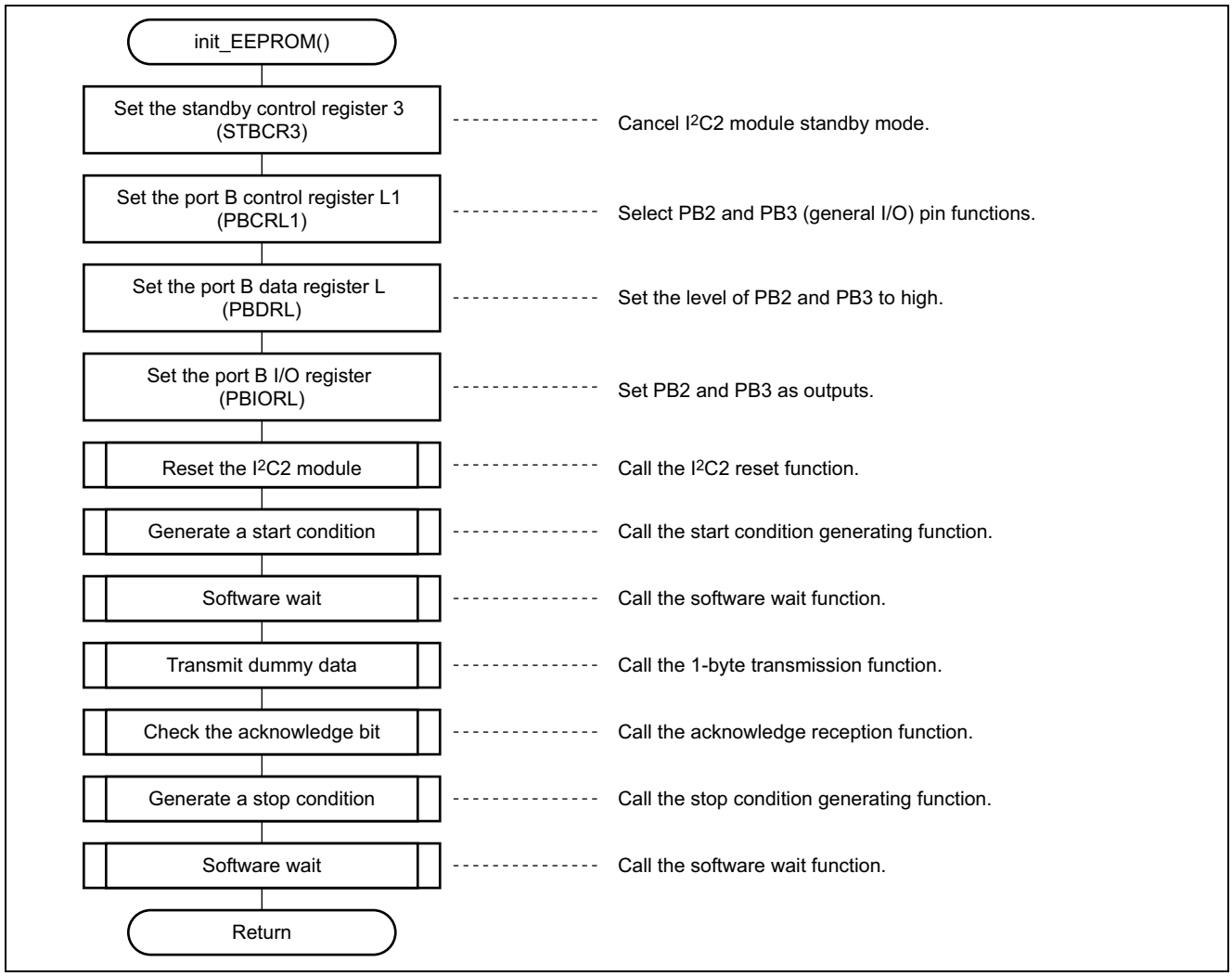
Note: This setting is used in EEPROM bus initialization.

6. Flowcharts

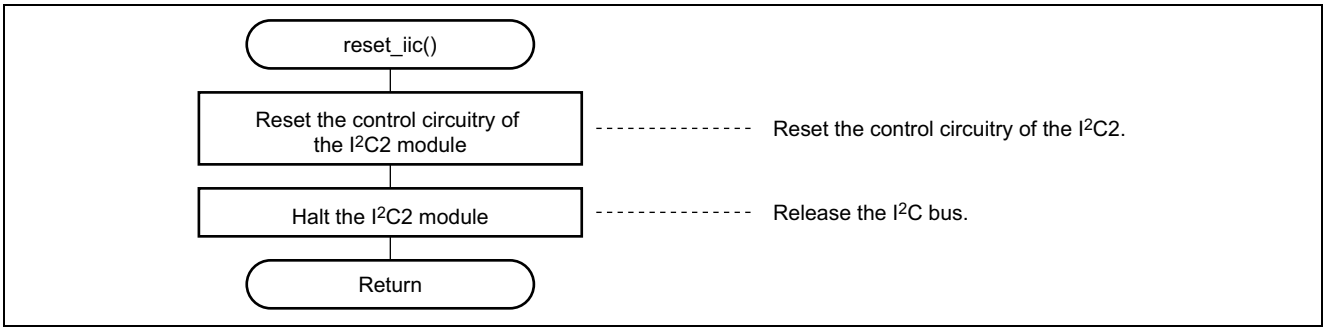
6.1 Main Function



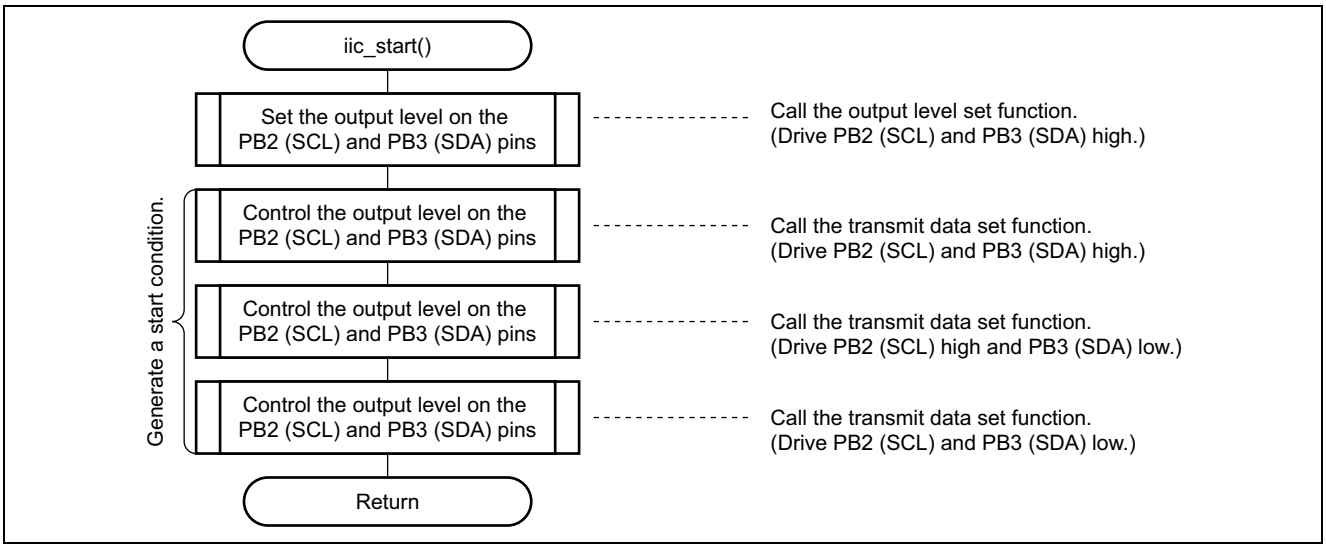
6.2 EEPROM Bus State Initialization Function



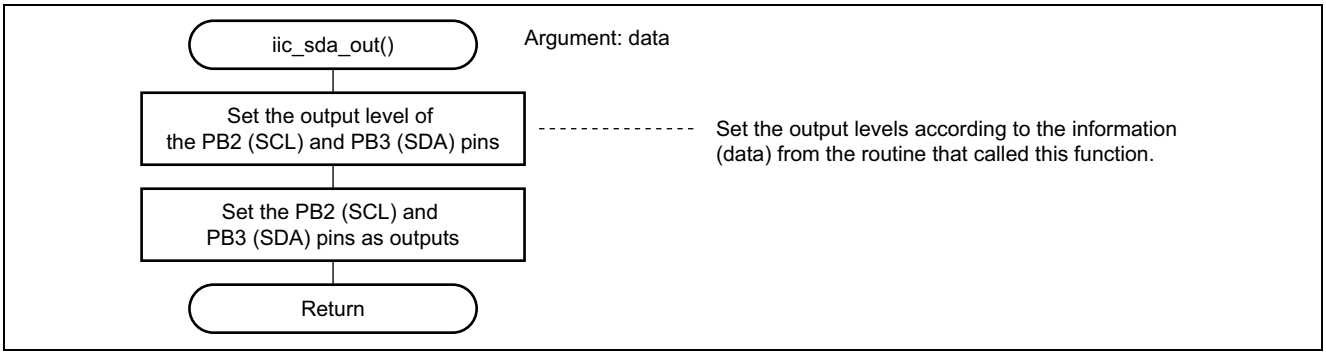
6.2.1 I²C2 Reset Function



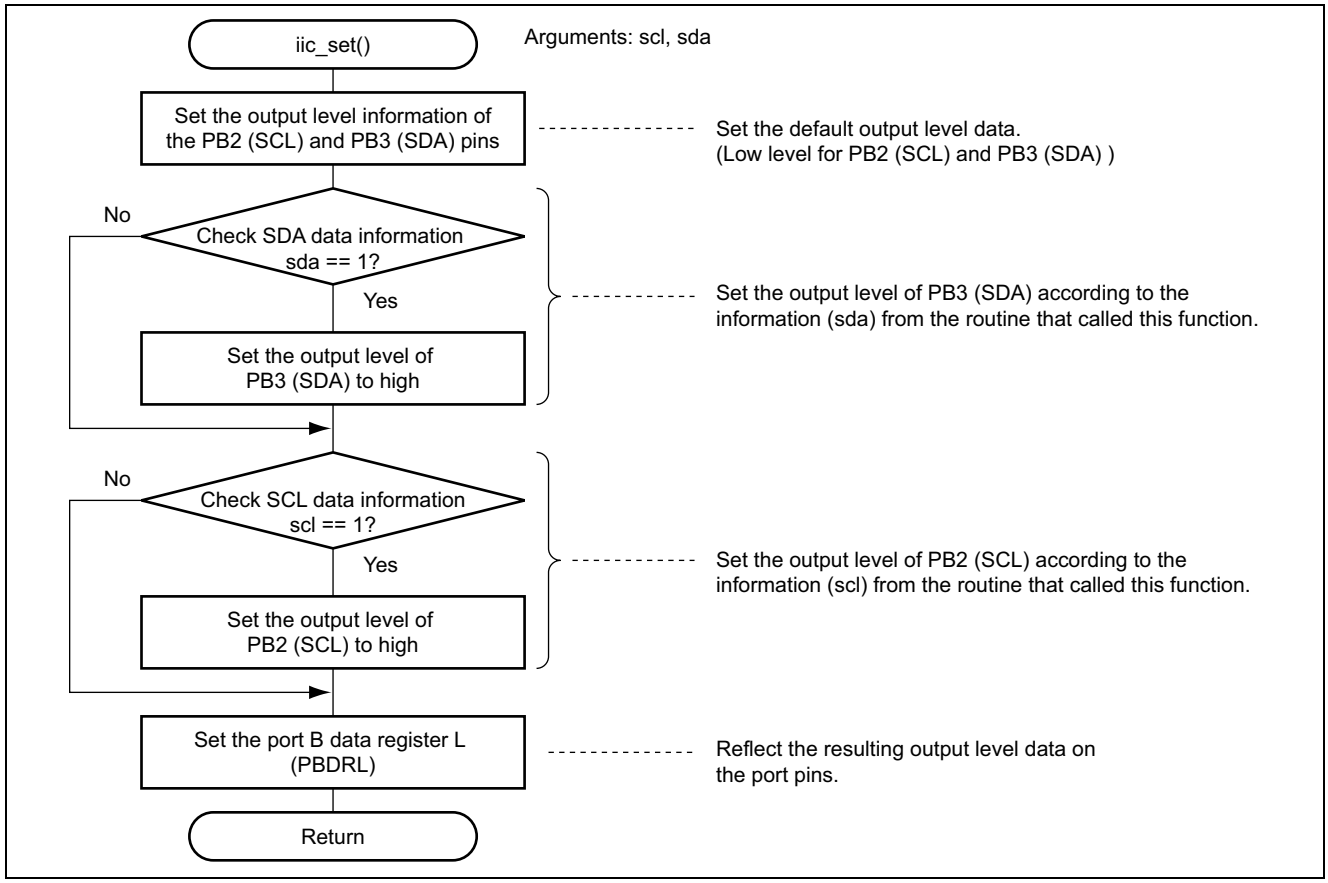
6.2.2 Start Condition Generation Function



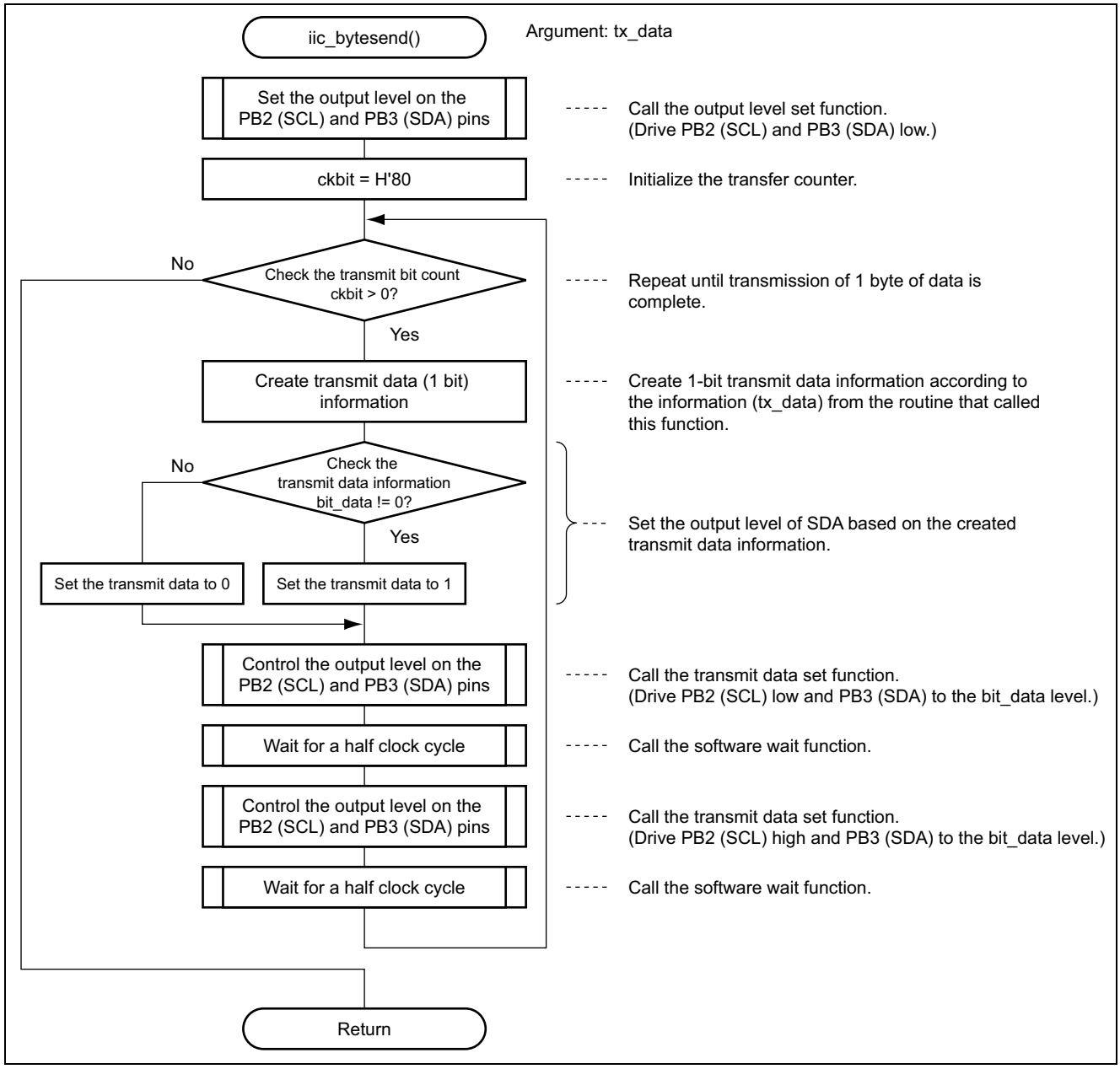
6.2.3 Output Level Set Function



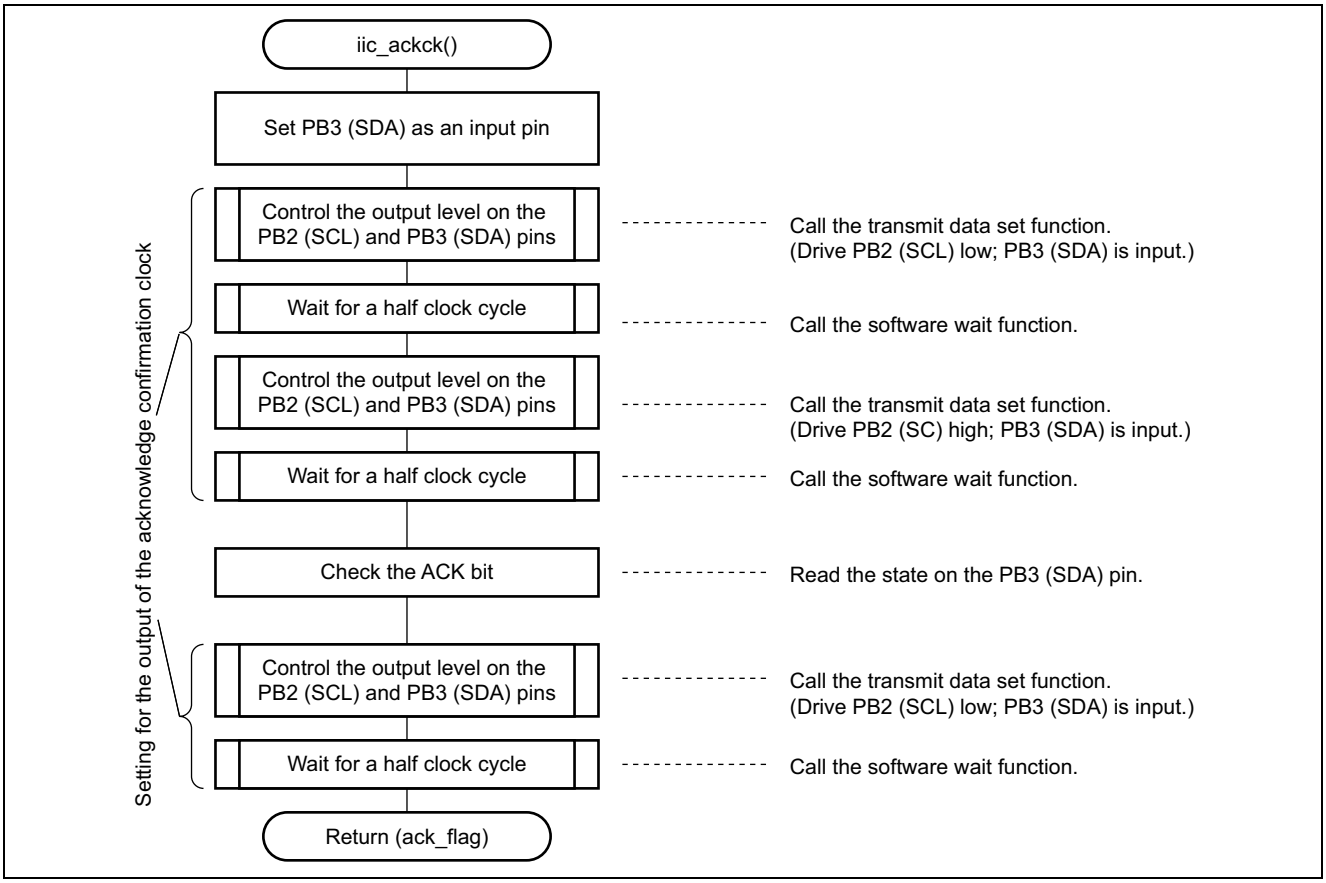
6.2.4 Transmit Data Set Function



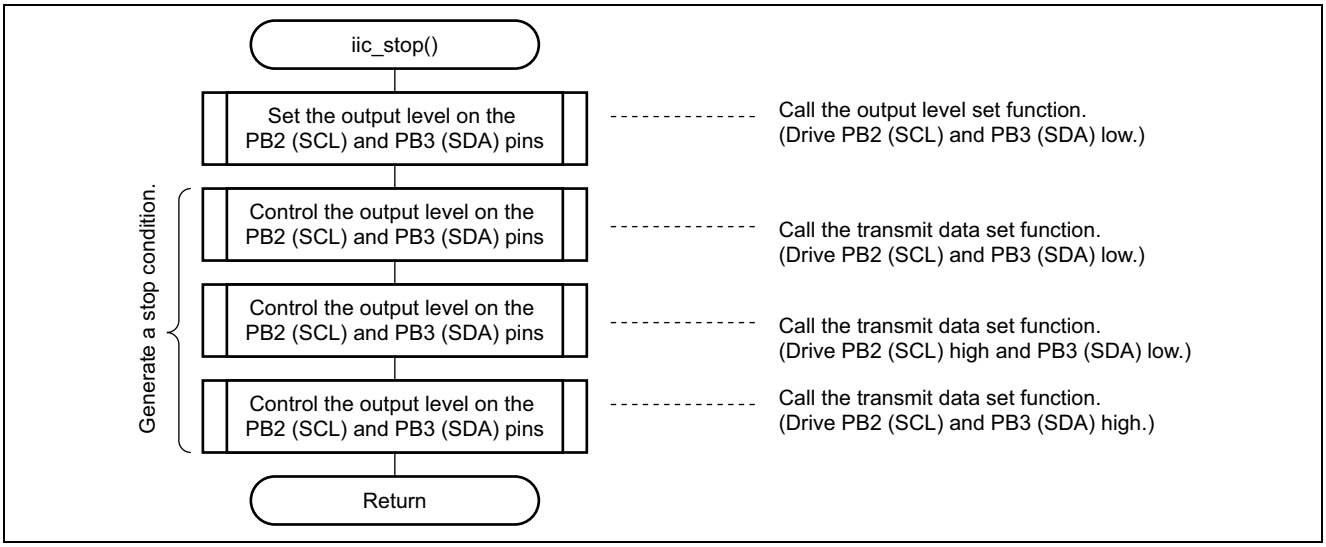
6.2.5 1-Byte Transmission Function



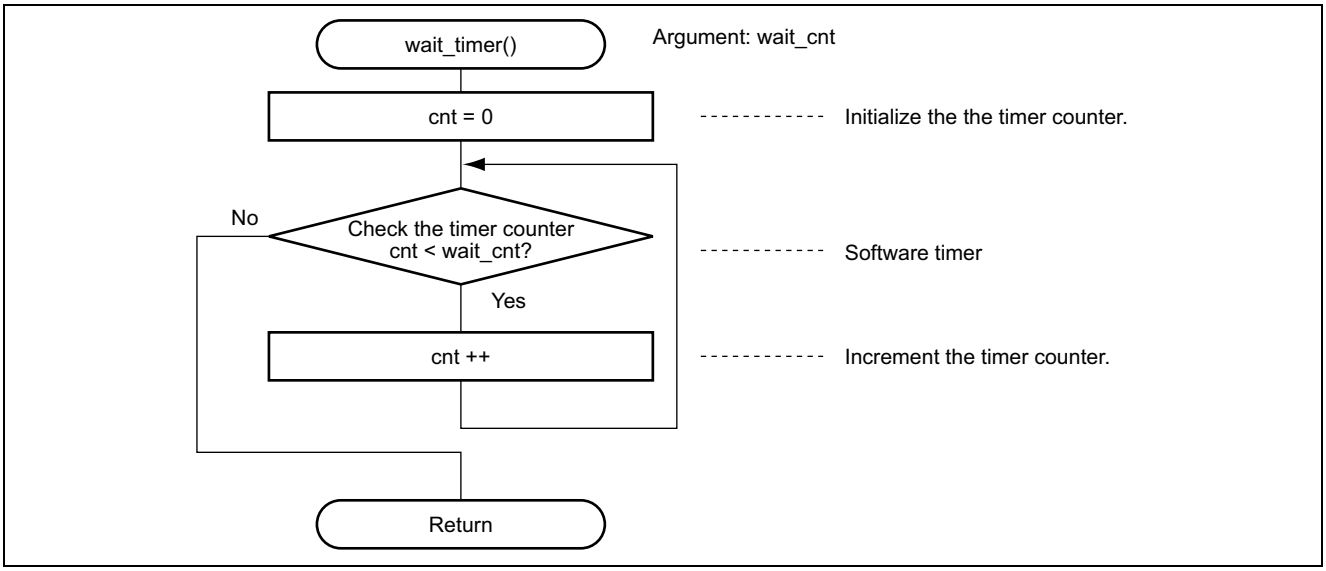
6.2.6 Acknowledge Reception Function



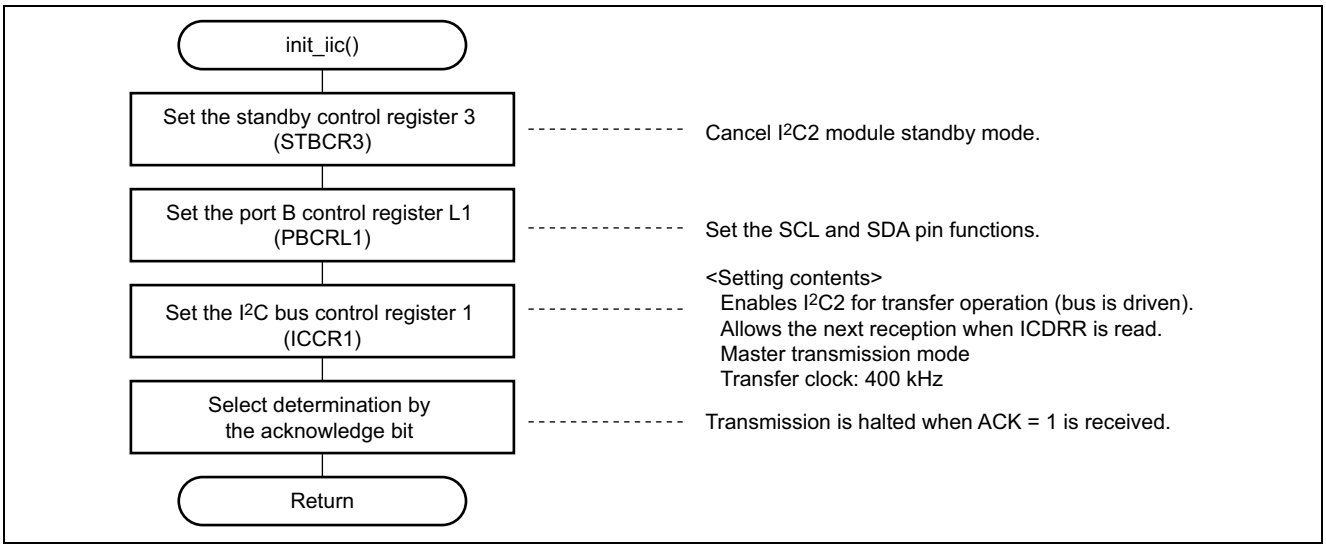
6.2.7 Stop Condition Generation Function



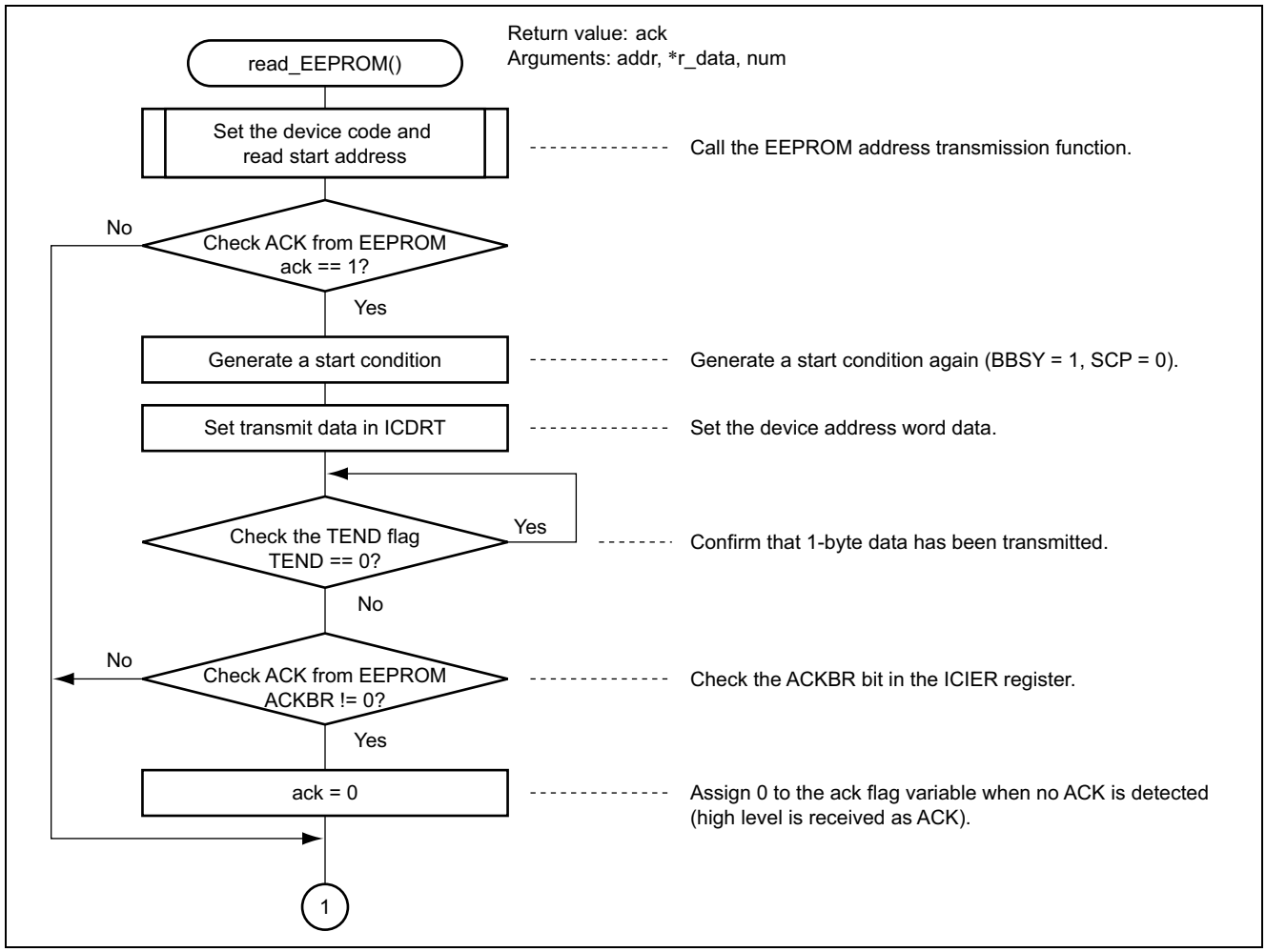
6.2.8 Software Wait Function

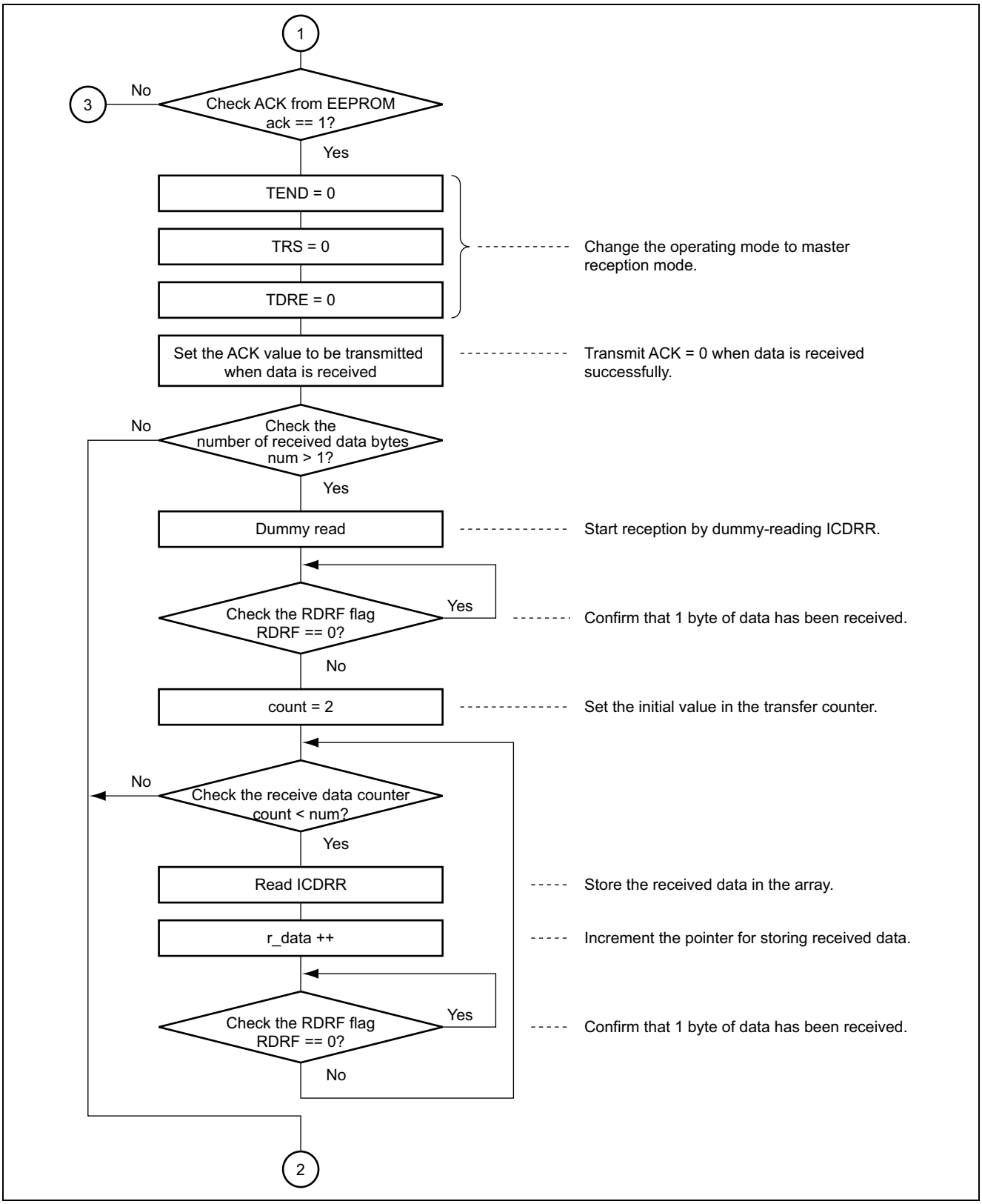


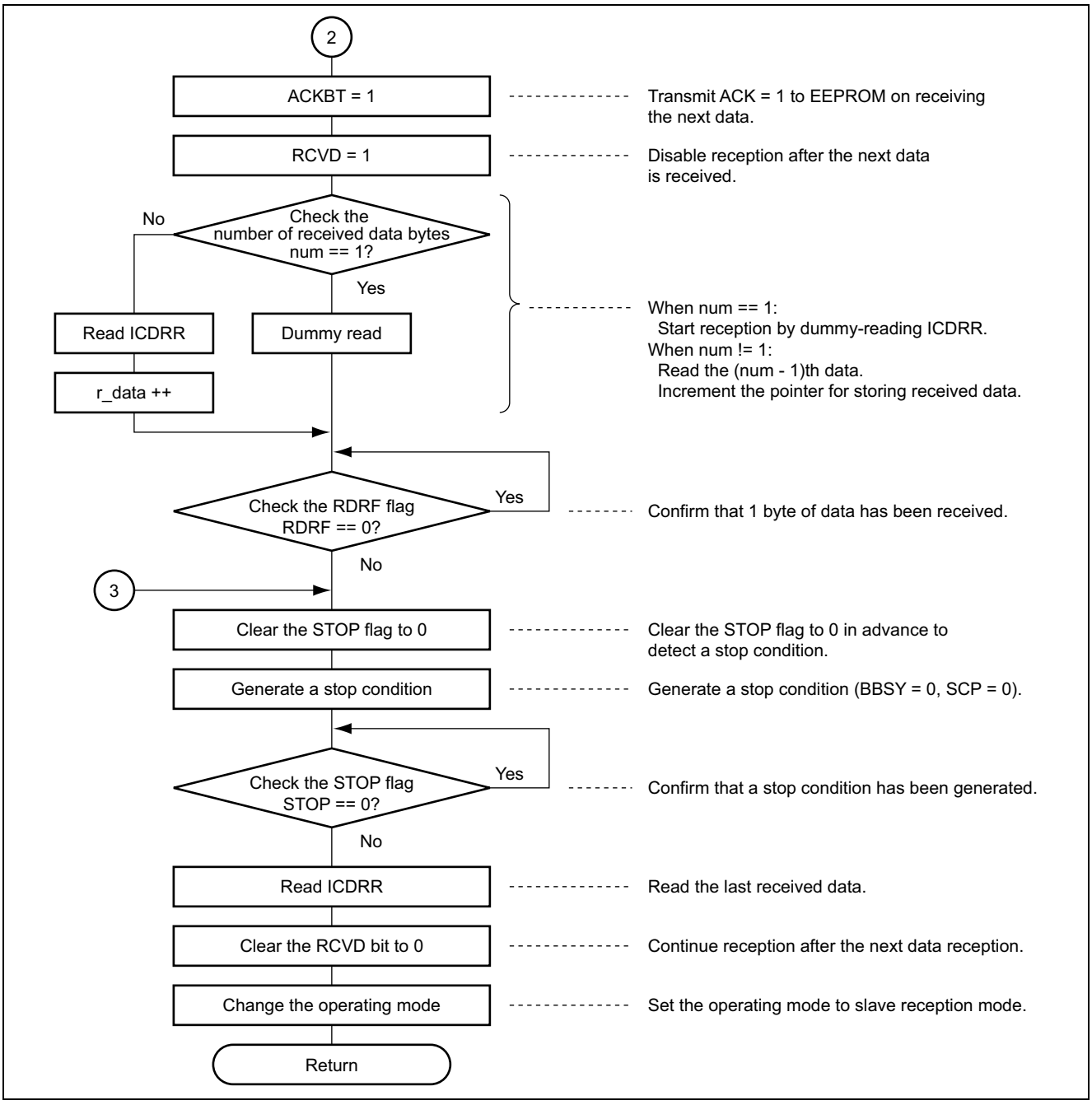
6.3 I²C2 Initialization Function



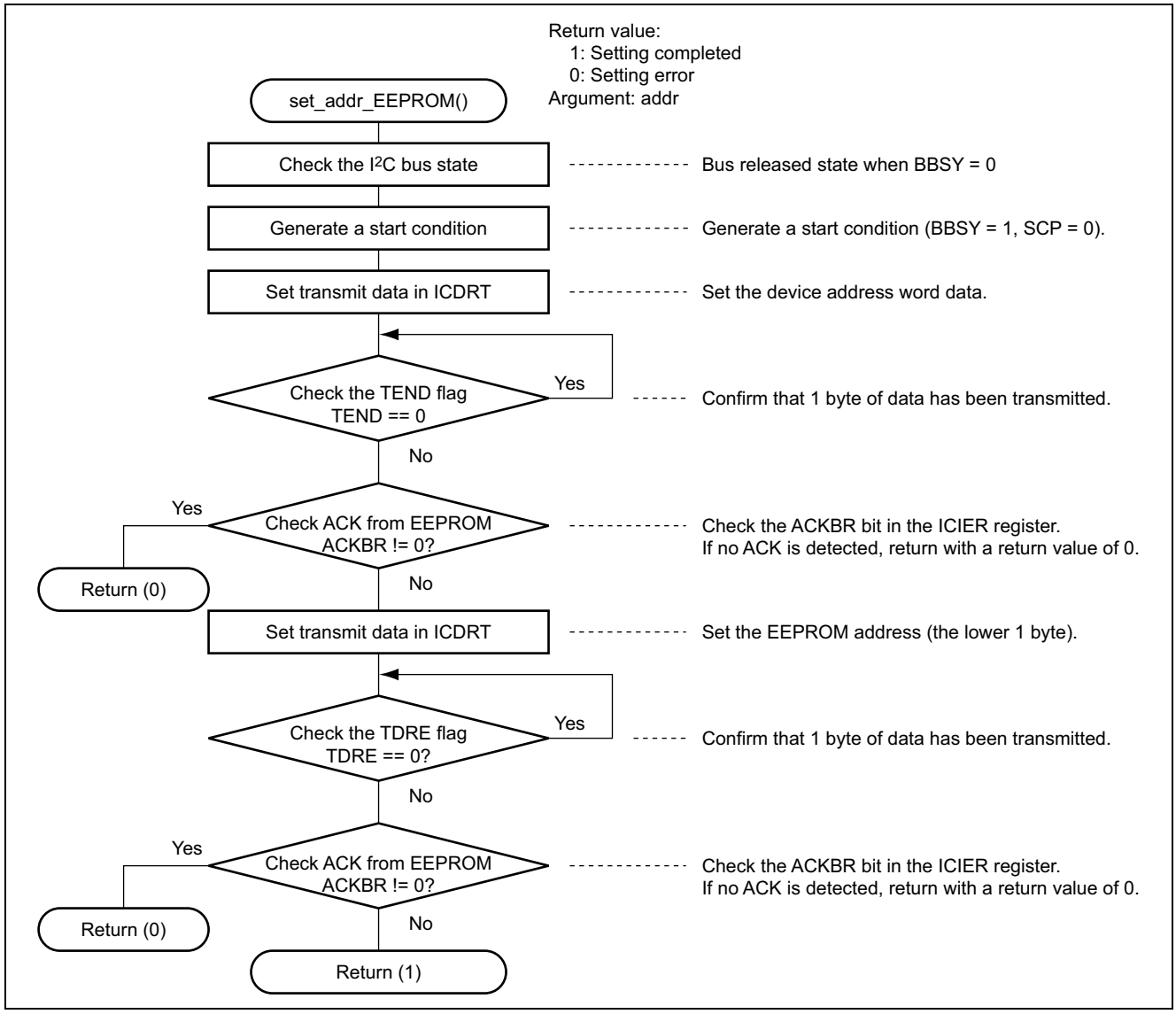
6.4 Data Read Function




1
3
No
Check ACK from EEPROM
ack == 1?
Yes
TEND = 0
TRS = 0
TDRE = 0
Change the operating mode to master reception mode.
Set the ACK value to be transmitted when data is received
Transmit ACK = 0 when data is received successfully.
No
Check the number of received data bytes
num > 1?
Yes
Dummy read
Start reception by dummy-reading ICDRR.
Check the RDRF flag
RDRF == 0?
Yes
Confirm that 1 byte of data has been received.
No
count = 2
Set the initial value in the transfer counter.
No
Check the receive data counter
count < num?
Yes
Read ICDRR
Store the received data in the array.
r_data ++
Increment the pointer for storing received data.
Check the RDRF flag
RDRF == 0?
Yes
Confirm that 1 byte of data has been received.
No
2


2
ACKBT = 1
Transmit ACK = 1 to EEPROM on receiving the next data.
RCVD = 1
Disable reception after the next data is received.
No
Check the number of received data bytes num == 1?
Yes
Read ICDRR
Dummy read
r_data ++
When num == 1:
Start reception by dummy-reading ICDRR.
When num != 1:
Read the (num - 1)th data.
Increment the pointer for storing received data.
Check the RDRF flag RDRF == 0?
Yes
Confirm that 1 byte of data has been received.
No
3
Clear the STOP flag to 0
Clear the STOP flag to 0 in advance to detect a stop condition.
Generate a stop condition
Generate a stop condition (BBSY = 0, SCP = 0).
Check the STOP flag STOP == 0?
Yes
Confirm that a stop condition has been generated.
No
Read ICDRR
Read the last received data.
Clear the RCVD bit to 0
Continue reception after the next data reception.
Change the operating mode
Set the operating mode to slave reception mode.
Return

6.4.1 EEPROM Address Transmission Function



7. Website

- Website of Renesas Technology Corp.
  http://www.renesas.com/

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep.14.05	—	First edition issued

Keep safety first in your circuit designs!

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

Notes regarding these materials

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