R32C/100 Series

EEPROM Control Using UARTi Special Mode 1 (I^2C Mode)

R01AN0759EJ0100
Rev. 1.00
Oct. 28, 2011

Abstract

This document describes EEPROM control using UART2 special mode 1 (I^2C mode).
Special mode 1 in the R32C/118 Group has seven channels (UART0 to UART6). When using a channel other than UART2, refer to the UARTi-related register in the User's Manual: Hardware for setting details.

Products

R32C/116 Group
R32C/117 Group
R32C/118 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

This chapter describes EEPROM control using UART2 special mode 1 (I^2C mode). Writing data to the EEPROM (write mode) and reading data from the EEPROM (read mode) are performed.
The Renesas Electronics R1EX24xxx Series EEPROM is used in the explanation below.
The settings below comply with I^2C-bus communication protocols.

Settings

- Transfer rate: Approx. 350 kbps (1)
- Transfer data length: 1 to 256 bytes (not including the device address word (2) and memory address)
- Single-master communication (multi-master is not supported)
- Restart condition generation applicable when in read mode

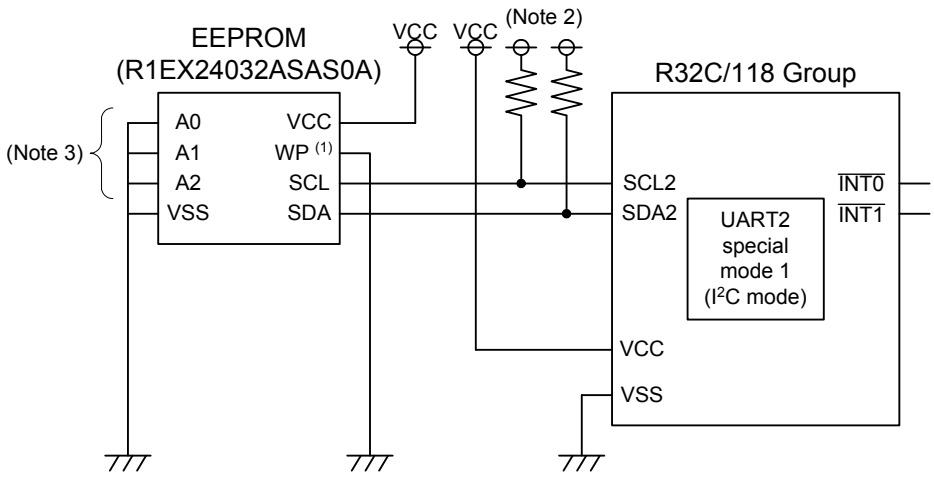
Notes:

1. The setting value is 378 kbps. (When clock synchronization is enabled, a sampling delay of 1 to 1.5 cycles of the U2BRG count source + the noise filter length occurs, high recognition of the SCL clock is delayed, and the high width of the SCL clock widens. Therefore, the actual SCL clock transfer rate will be slower than what is set. As clock synchronization is enabled in this application example, the actual transfer rate will be approx. 350 kbps (reference values: pull-up voltage is 5 V, pull-up resistor value is 1 kΩ).) This is the transfer rate in Standard-Mode and Fast-Mode.
2. The device address word is comprised of a 4-bit device code, a 3-bit device address code, and a 1-bit read/write code.

Table 1.1 lists the Peripheral Functions and Their Applications, Figure 1.1 shows a Connection Example, and Figure 1.2 shows the Communication Format.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
Serial interface (UART2)	Communication with the EEPROM
INT0 interrupt	Change mode to write mode
INT1 interrupt	Change mode to read mode



Notes:
1. In this application note, the WP pin of the EEPROM is not controlled by the MCU.
2. The SCL2 and SDA2 pins in the R32C/118 Group are set to N-channel open drain. Enable the pull-up resistors according to the user system.
3. Set the device address to pins A0 to A2. Connect VCC or VSS depending on the set device address.

Figure 1.1 Connection Example

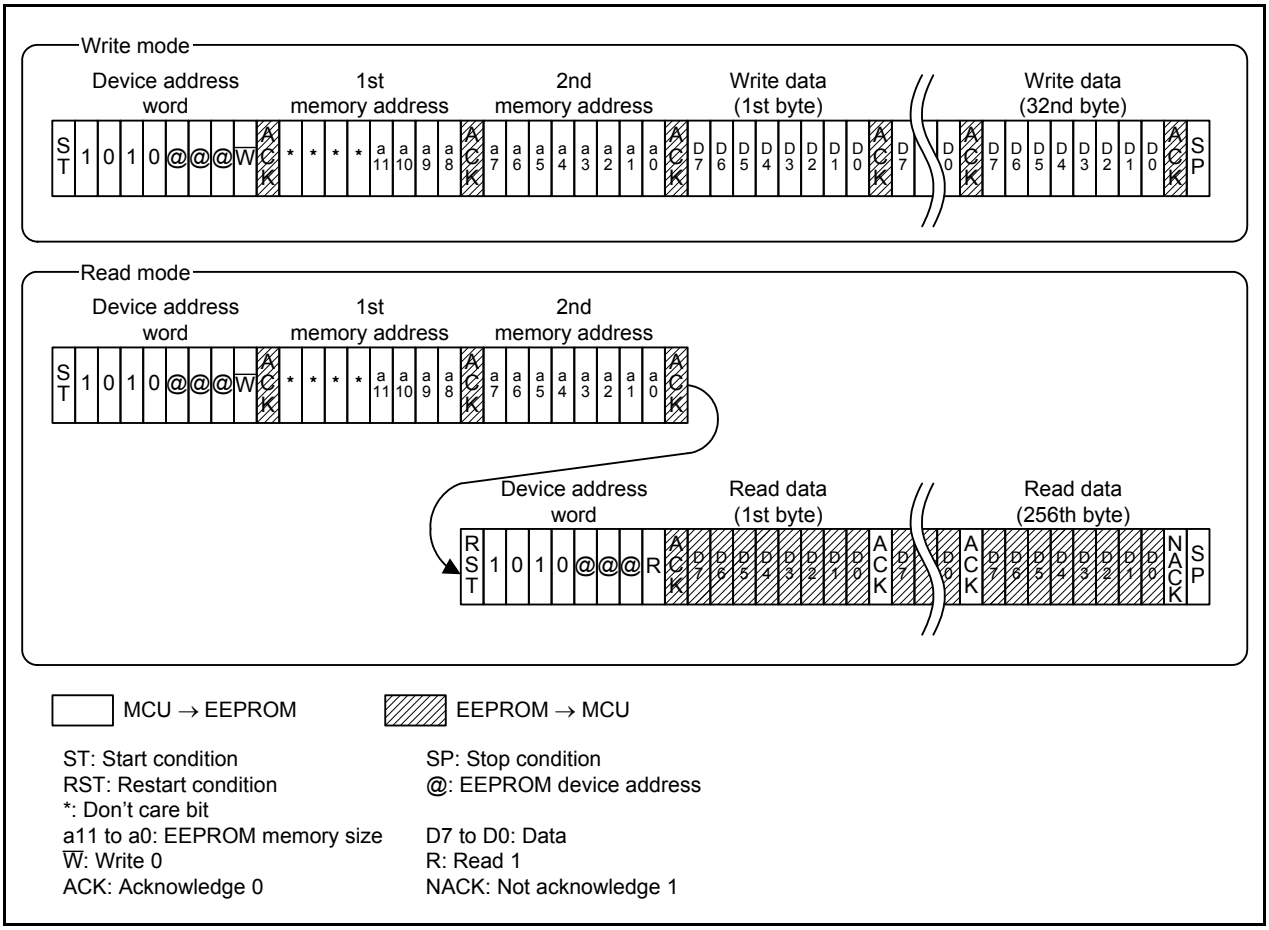


Figure 1.2 Communication Format

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F64189DFD (R32C/118 Group)
Device used	R1EX24032ASAS0A
Operating frequencies	• XIN clock: 16 MHz • PLL clock: 100 MHz • Base clock: 50 MHz • CPU clock: 50 MHz • Peripheral bus clock: 25 MHz • Peripheral clock: 25 MHz
Operating voltage	5 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.08
C compiler	Renesas Electronics Corporation R32C/100 Series C Compiler V.1.02 Release 01
	Compile options -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFBDC -c -finfo -dir “$(CONFIGDIR)” (Default setting is used in the integrated development environment.)
Operating mode	Single-chip mode
Sample code version	1.00

3. Reference Application Notes

Application notes associated with this application note are listed below. Refer to these application notes for additional information.

- R32C/100 Series Configuring PLL Mode (REJ05B1221)
- M16C Family, R8C Family I^2C-bus Interface Using UARTi Special Mode 1 (REJ05B1349)
- R32C/100 Series I^2C-bus Interface Using UARTi Special Mode 1 (Master Transmit/Receive) (REJ05B1395)

4. Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P7_0/SDA2	I/O	Data input and output in I^2C mode
P7_1/SCL2	Output	Clock output in I^2C mode
P8_2/$\overline{\text{INT0}}$	Input	Input for INT0 interrupt
P8_3/$\overline{\text{INT1}}$	Input	Input for INT1 interrupt

5. Software

This chapter describes how to write 256 bytes to the EEPROM (write mode) or read 256 bytes from the EEPROM (read mode) using the R32C/118 Group MCU. In write mode, 32 bytes of data (1 page) are written eight times (= 256 bytes). The mode changes to write mode when an INT0 interrupt is generated. The mode changes to read mode when an INT1 interrupt is generated.

The settings below are used with the serial interface (UART2) in special mode 1 (I^2C mode).
Settings

- Use I^2C mode.
- Use the internal clock for the transfer clock.
- Use f1 for the U2BRG count source.
- Set the SDA2 and SCL2 pins as N-channel open drain output.
- Use MSB first as the transfer format.
- Use transmission completed (TXEPT is 1) as the UART2 transmit interrupt source.
- Select clock delayed for the clock-phase setting.
- Use seven to eight cycles of the U2BRG count source for the SDA2 digital delay time.
- Use clock synchronization.
- Do not use the function that performs SCL2 wait output.
- Do not use the function that performs SDA2 output stop.
- Use the start condition/stop condition detection interrupt.
- Use the UART2 transmit interrupt.
- Do not use the UART2 receive interrupt.
- Set the transfer rate to approx. 378 kbps.

Transfer rate formula:
Transfer rate = U2BRG count source ÷ (2 × (U2BRG register setting value + 1))
= 25 MHz (f1) ÷ (2 × (32 + 1)
≈ 378.788 kbps

5.1 Operation Overview

5.1.1 Operation During Write Mode

(1) Initial setting
This operation initializes the system clock, UART2-related SFRs, and INT interrupt-related SFRs. When an INT0 interrupt is generated, write mode is set to the variable (mode) on the RAM, and the start condition is generated.

(2) Start condition/stop condition detection interrupt
When the start condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission.

(3) UART2 transmit interrupt
A UART2 transmit interrupt is generated at the falling edge of the 9th bit of the SCL clock. In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the first memory address to the U2TB register.

(4) UART2 transmit interrupt
In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the second memory address to the U2TB register.

(5) UART2 transmit interrupt
In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the write data to the U2TB register. Repeat step (5) until 32 bytes of data have been transmitted.

(6) UART2 transmit interrupt
A stop condition is generated in the UART2 transmit interrupt handling for every 32 bytes of data transmitted.

(7) Start condition/stop condition detection interrupt
When a stop condition is detected, a start condition/stop condition detection interrupt is generated, and stop condition detection processing is performed.

(8) Wait until data is written to EEPROM
In the stop condition detection processing, set a software loop to 5 ms in order for data to be written to the EEPROM. If write data remains after the stop condition detection processing, a start condition is generated.

Repeat steps (2) through (8) until 256 bytes have been transmitted.
After 256 bytes have been transmitted, the CPU waits for an INT0 interrupt or INT1 interrupt to be generated.

Figure 5.1 shows the Timing Diagram in Write Mode.

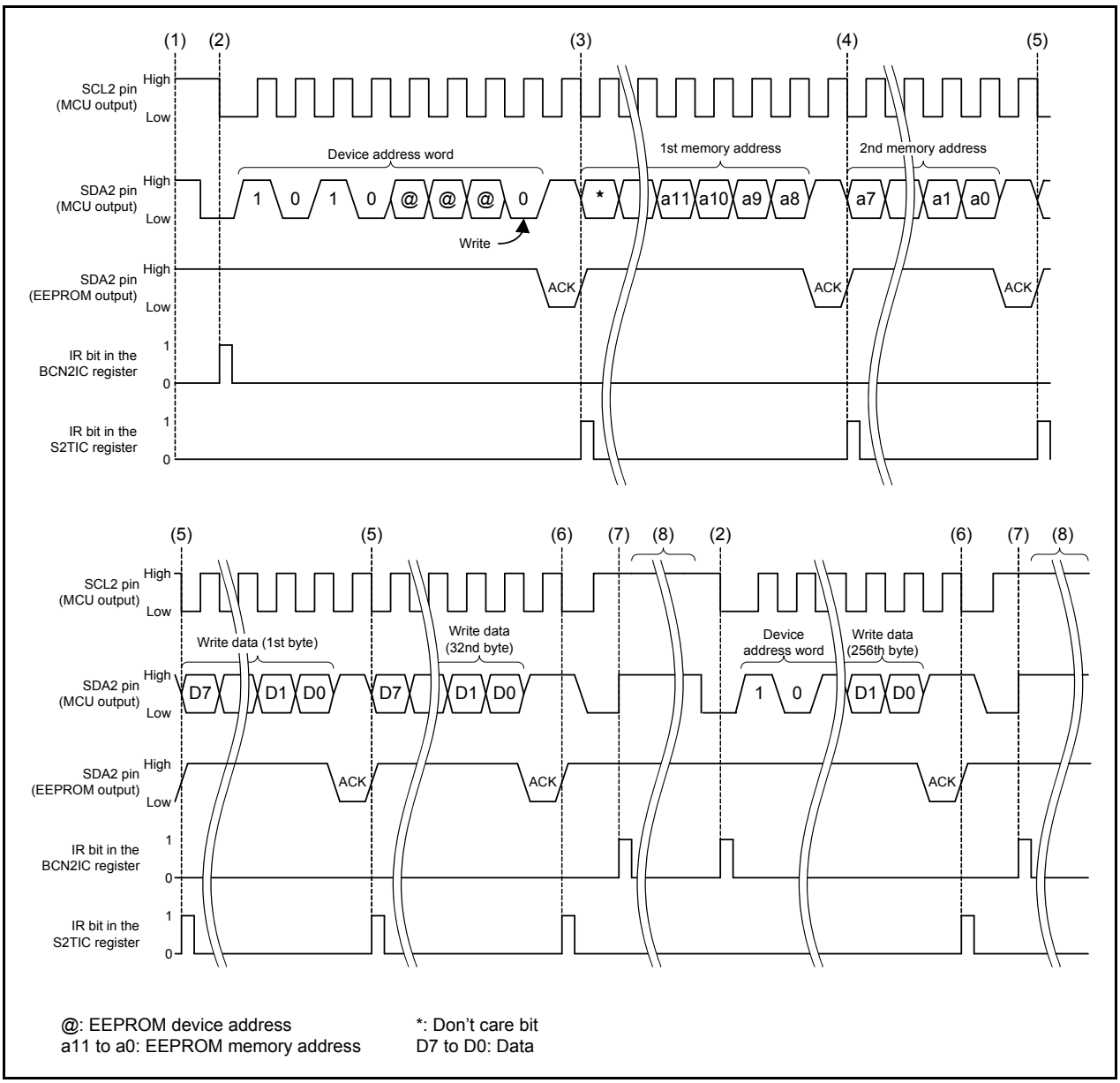


Figure 5.1 Timing Diagram in Write Mode

5.1.2 Operation During Read Mode

(1) Initial setting
This operation initializes the system clock, UART2-related SFRs, and INT interrupt-related SFRs. When an INT1 interrupt is generated, read mode is set to the variable (mode) on the RAM, and the start condition is generated.

(2) Start condition/stop condition detection interrupt
When the start condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission. (At this point, the 8th bit of the device address word is 0 (write code).)

(3) UART2 transmit interrupt
A UART2 transmit interrupt is generated at the falling edge of the 9th bit of the SCL clock. In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the first memory address to the U2TB register.

(4) UART2 transmit interrupt
In the UART2 transmit interrupt handling, confirm an ACK has been received and then set the second memory address to the U2TB register.

(5) UART2 transmit interrupt
Generate a restart condition in the UART2 transmit interrupt handling.

(6) Start condition/stop condition detection interrupt
After a restart condition generation is completed, a start condition/stop condition detection interrupt is generated. Set the device address word to the U2TB register, and start transmission. (At this point, the 8th bit of the device address word is 1 (read code).)

(7) UART2 transmit interrupt
In order to receive the next data, set dummy data and ACK data to the U2TB register.

(8) UART2 transmit interrupt
Read the read data from the U2RB register, and write it to the receive data buffer. In order to receive the next data, set dummy data and ACK data to the U2TB register. Repeat step (8) until 254 bytes of data have been received.

(9) UART2 transmit interrupt
Read the read data from the U2RB register, and write it to the receive data buffer. In order to receive the 256th byte of data (the last data), set dummy data and NACK data to the U2TB register.

(10) UART2 transmit interrupt
Read the read data from the U2RB register, and write it to the receive data buffer. Generate a stop condition in the UART2 transmit interrupt handling.

(11) Start condition/stop condition detection interrupt
When a stop condition is detected, a start/stop condition detection interrupt is generated, and stop condition detection processing is performed. Wait for an INT0 or INT1 interrupt to be generated.

Figure 5.2 shows the Timing Diagram in Read Mode.

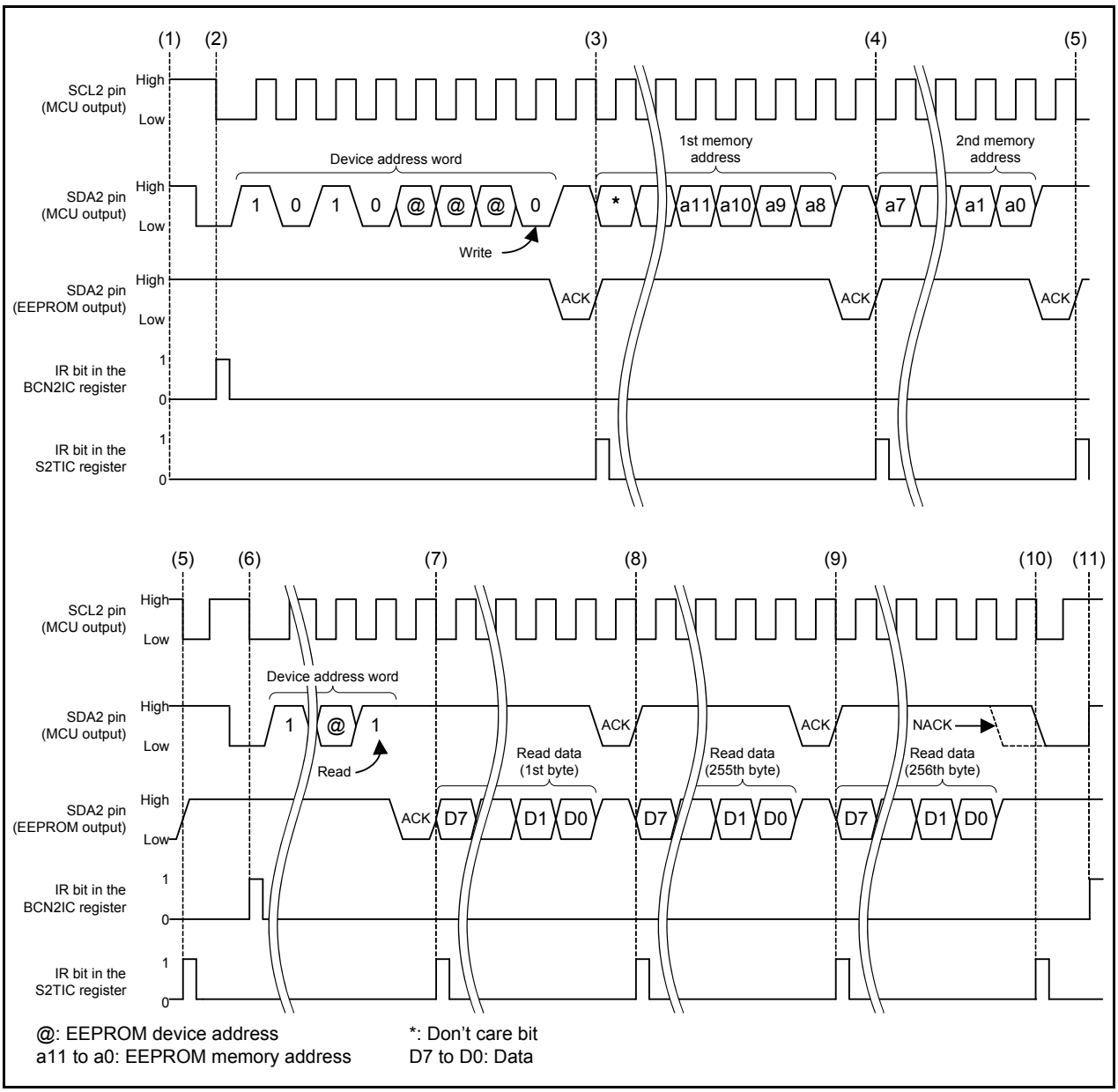


Figure 5.2 Timing Diagram in Read Mode

5.2 Constants

Table 5.1 lists the Constants Used in the Sample Code.

Table 5.1 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
DEVICE_ADDR_WORD	1010 0000b	Device address word b7 to b4: Device code b3 to b1: Slave address b0: R/W code
EEPROM_MEM_ADDR	0000h	EEPROM memory address
LENGTH	32	Transfer size per page
W_SIZE	256	Transmit data size (W_SIZE ≤ BUFSIZE)
R_SIZE	256	Receive data size (R_SIZE ≤ BUFSIZE)
BUFSIZE	256	Transmit/receive data buffer size
PAGE	(Note 1)	Number of pages transmitted
IIC_BRG	(33 - 1)	25 MHz ÷ (2 × (32 + 1)) ≈ 378 kbps
WAIT_LOOP	50000	Wait time to write to EEPROM
DISABLE	0	Disable
ENABLE	1	Enable
BUSY	0	Midcommunication
RDY	1	Communication completed
PAR_ERR	FFh	Parameter error
IDLE	FFh	Idle
WRITE	0	R/W (write mode)
READ	1	R/W (read mode)
R_CODE	1	EEPROM read code
W_CODE	0	EEPROM write code
IIC_SP_ON	1	Generate stop condition
IIP_SP_OFF	0	Do not generate stop condition
PD_IIC	pd7	PD7_0: SDA2, PD7_1: SCL2
PD_IIC_INIT	0000 0011b	PD7_0: Output, PD7_1: Output
PD_IIC_INIT_INPUT	1111 1100b	PD7_0: Input, PD7_1: Input

Note:
1. ((W_SIZE + LENGTH - 1) ÷ LENGTH)

5.3 Structure/Union List

Figure 5.3 shows the Structure/Union Used in the Sample Code.

```
typedef union{
   struct{
      unsigned char b0:1;
      unsigned char b1:1;
      unsigned char b2:1;
      unsigned char b3:1;
      unsigned char b4:1;
      unsigned char b5:1;
      unsigned char b6:1;
      unsigned char b7:1;
   }bit;
   unsigned char all;
}byte_dt;

typedef union{
   struct{
      unsigned char b0:1;
      unsigned char b1:1;
      unsigned char b2:1;
      unsigned char b3:1;
      unsigned char b4:1;
      unsigned char b5:1;
      unsigned char b6:1;
      unsigned char b7:1;
      unsigned char b8:1;
      unsigned char b9:1;
      unsigned char b10:1;
      unsigned char b11:1;
      unsigned char b12:1;
      unsigned char b13:1;
      unsigned char b14:1;
      unsigned char b15:1;
   }bit;
   struct{
      unsigned char byte0;
      unsigned char byte1;
   }byte;
   unsigned short all;
}word_dt;

byte_dt iic_str1;                          /* Device address word */
#define device_addr_word  iic_str1.all     /* b7 to b4: Device code (fixed) b3 to b1: Device address code b0: R/W code */
#define iic_rw            iic_str1.bit.b0  /* 0: Write (master transmit) 1: Read (master receive) */

byte_dt iic_str2;                          /* Status */
#define iic_status        iic_str2.all     /* All statuses */
#define iic_start         iic_str2.bit.b0  /* 1: Midcommunication      0: Communication completed */
#define iic_err_par       iic_str2.bit.b1  /* 1: Parameter error       0: No error detected */
#define iic_err_nack      iic_str2.bit.b2  /* 1: NACK detection error  0: No error detected */
```

Figure 5.3 Structure/Union Used in the Sample Code

5.4 Variables

Table 5.2 lists the Global Variables.

Table 5.2 Global Variables

Type	Variable Name	Contents	Function Used
unsigned short	iic_length	Transfer data length	iic_master_start, master_trn_int, master_rcv_int
unsigned char	trn_data[BUFSIZE]	Transmit data buffer	main, stp_int
unsigned char	rcv_data[BUFSIZE]	Receive data buffer	main
unsigned short	mem_addr	Memory address buffer	main, stp_int, mem_addr_trn_int
unsigned char	mode	Save mode	main, _int0, _int1
unsigned char far	*ptr_buf	Buffer pointer	iic_master_start, master_trn_int, master_rcv_int
unsigned short	cnt_tr_index	Number of transmitted/received bytes	sta_int, stp_int, _uart2_trans, master_trn_int, master_rcv_int
unsigned short	cnt_remain_t_index	Remaining transmit data counter	iic_master_start, stp_int
unsigned char	cnt_mem_addr	Memory address transmit counter	_start_stop_condition_detection, sta_int, stp_int, _uart2_trans
unsigned char	cnt_page	Page counter	iic_master_start, stp_int

5.5 Functions

Table 5.3 lists the Functions.

Table 5.3 Functions

Function Name	Outline
main	Main processing
uart2_init	UART2 initial setting
iic_master_start	Master control start processing
_start_stop_condition_detection	Start condition/stop condition detection interrupt handling
sta_int	Start condition detection processing
stp_int	Stop condition detection processing
re_sta_int	Restart condition detection processing
_uart2_trans	UART2 transmit interrupt handling
mem_addr_trn_int	Memory address transmission
master_trn_int	Master transmission
master_rcv_int	Master reception
int_init	INT0 and INT1 initial setting
_int0	INT0 interrupt handling
_int1	INT1 interrupt handling

5.6 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	The initial setting of the system clock, UART2, INT0, and INT1 are performed. After the initial setting of INT0 and INT1, wait for the INT0 and INT1 interrupts to be generated. In write mode, the value of the variable (mode) on the RAM is the write data for the EEPROM; in read mode the value is the read data for the EEPROM.
Argument	None
Returned value	None
Remark	

uart2_init	
Outline	UART2 initial setting
Header	None
Declaration	void uart2_init(unsigned char init)
Description	Set UART2 special mode 1 (I^2C mode).
Argument	First argument: init: I^2C mode enabled/disabled
Returned value	None
Remark	

iic_master_start	
Outline	Master control start processing
Header	None
Declaration	unsigned char iic_master_start(unsigned char addr, unsigned char rw, unsigned char far *buf, unsigned short len)
Description	Processing to start transmission/reception. All statuses are initialized at the start of this function, and a parameter check of the argument is performed. If the parameters are outside the range of the argument, the parameter error flag becomes 1, and the returned value PAR_ERR is returned. When there is a parameter error, master control start processing is not performed. Confirm the following bus states: • When the bus is busy, the returned value BUSY is returned, and master control start processing is not performed. • When the bus is free, the returned value RDY is returned, and master control start processing is performed. The midcommunication flag becomes 1, and a start condition is generated.
Arguments	First argument: addr : Device address word Second argument: rw : Write mode/read mode Third argument: *buf : Pointer for transmission buffer or reception buffer Fourth argument: len : Transfer data length
Returned value	• When bus is busy: BUSY • When bus is free: RDY • When there is a parameter error: PAR_ERR
Remark	

_start_stop_condition_detection	
Outline	Start condition/stop condition detection interrupt handling
Header	None
Declaration	void _start_stop_condition_detection(void)
Description	An interrupt is generated when a start condition/stop condition is detected. The sta_int function is called when a start condition generation is completed, the re_sta_int function is called when a restart condition generation is completed, and a stp_int function is called when a stop condition is detected.
Argument	None
Returned value	None
Remark	

sta_int	
Outline	Start condition detection processing
Header	None
Declaration	void sta_int(void)
Description	This function is called from the start condition/stop condition detection interrupt handling. After transmission/reception is enabled, the device address word is transmitted.
Argument	None
Returned value	None
Remark	

stp_int	
Outline	Stop condition detection processing
Header	None
Declaration	void stp_int(void)
Description	This function is called from the start condition/stop condition detection interrupt handling. If there is no data remaining to transmit/receive, UART2-related SFRs that were rewritten during communication return to their initial setting, and the midcommunication flag becomes 0. Enable the INT0 and INT1 interrupts.
Argument	None
Returned value	None
Remark	

re_sta_int	
Outline	Restart condition detection processing
Header	None
Declaration	void re_sta_int(void)
Description	This function is called from the start condition/stop condition detection interrupt handling. When in read mode, the device address word is transmitted.
Argument	None
Returned value	None
Remark	

_uart2_trans	
Outline	UART2 transmit interrupt handling
Header	None
Declaration	void _uart2_trans(void)
Description	An interrupt is generated at the falling edge of the 9th bit of the SCL clock. The U2RB register is read at the start of this function. • When a device address word is transmitted, a stop condition is generated when a NACK is detected. • Before the memory address is transmitted Read the mem_addr_trn_int function. • After the memory address is transmitted a) While in write mode, the master_trn_int function is called. b) While in read mode before a restart condition is generated, a restart condition is generated. c) While in read mode after a restart condition is generated, the master_rcv_int function is called. A stop condition is generated when communication is completed.
Argument	None
Returned value	None
Remark	

mem_addr_trn_int	
Outline	Memory address transmission
Header	None
Declaration	unsigned char mem_addr_trn_int(unsigned short rb_data, unsigned char cnt, unsigned char len)
Description	This function is called in the UART2 transmit handling. The memory address is transmitted. • When the first byte of the memory address is transmitted, only the first byte of the memory address is transmitted. • When the second byte of the memory address is transmitted, after the second byte of the memory address is transmitted, the memory address transmitted next is updated.
Argument	First argument: rb_data : Data read from the U2RB data Second argument: cnt : Number of data counted Third argument: len : Transfer data length
Returned value	• When a stop condition is generated: IIC_SP_ON • When a stop condition is not generated: IIC_SP_OFF
Remark	

master_trn_int	
Outline	Master transmission
Header	None
Declaration	unsigned char master_trn_int(unsigned short rb_data)
Description	This function is read in the UART2 transmit interrupt handling. When an ACK is detected, and the data is not the last byte (the next transmission starts), the returned value is IIC_SP_OFF. When a NACK is detected (the NACK detection error flag becomes 1), and the last byte has been transmitted, the returned value is IIC_SP_ON.
Argument	First argument: rb_data: Data read from the U2RB register
Returned value	• When a stop condition is generated: IIC_SP_ON • When a stop condition is not generated: IIC_SP_OFF
Remark	

master_rcv_int	
Outline	Master reception
Header	None
Declaration	unsigned char master_rcv_int(unsigned short rb_data)
Description	This function is read in the UART2 transmit interrupt handling. Store the argument value in the data buffer (except when the value is the device address word). If the next data is the last byte, set a NACK; if the next data is not the last byte, set an ACK. Then start the next reception. When the data is not the last byte, the returned value IIC_SP_OFF is returned. When the last byte reception is completed, the returned value IIC_SP_ON is returned.
Argument	First argument: rb_data: Data read from the U2RB register
Returned value	• When a stop condition is generated: IIC_SP_ON • When a stop condition is not generated: IIC_SP_OFF
Remark	

int_init	
Outline	INT0 and INT1 initial setting
Header	None
Declaration	void int_init(void)
Description	Set the INT0 and INT1 interrupt priority levels to 1.
Argument	None
Returned value	None
Remark	

_int0	
Outline	INT0 interrupt handling
Header	None
Declaration	void _int0(void)
Description	Disable the INT0 and INT1 interrupts, and enable write mode.
Argument	None
Returned value	None
Remark	

_int1	
Outline	INT1 interrupt handling
Header	None
Declaration	void _int1(void)
Description	Disable the INT0 and INT1 interrupts, and enable read mode.
Argument	None
Returned value	None
Remark	

5.7 Flowcharts

5.7.1 Main Processing

Figure 5.4 and Figure 5.5 show the Main Processing.

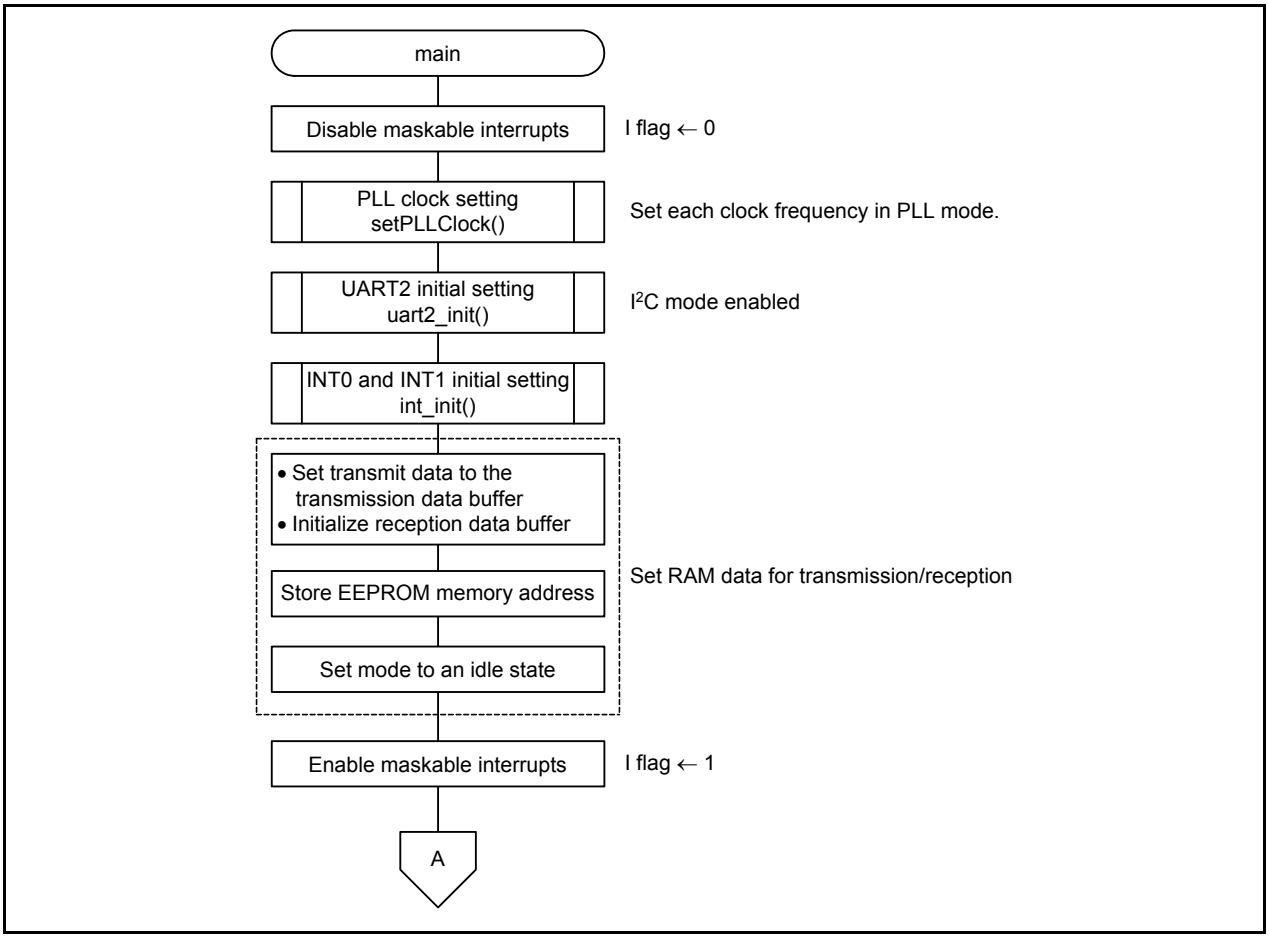


Figure 5.4 Main Processing (1/2)

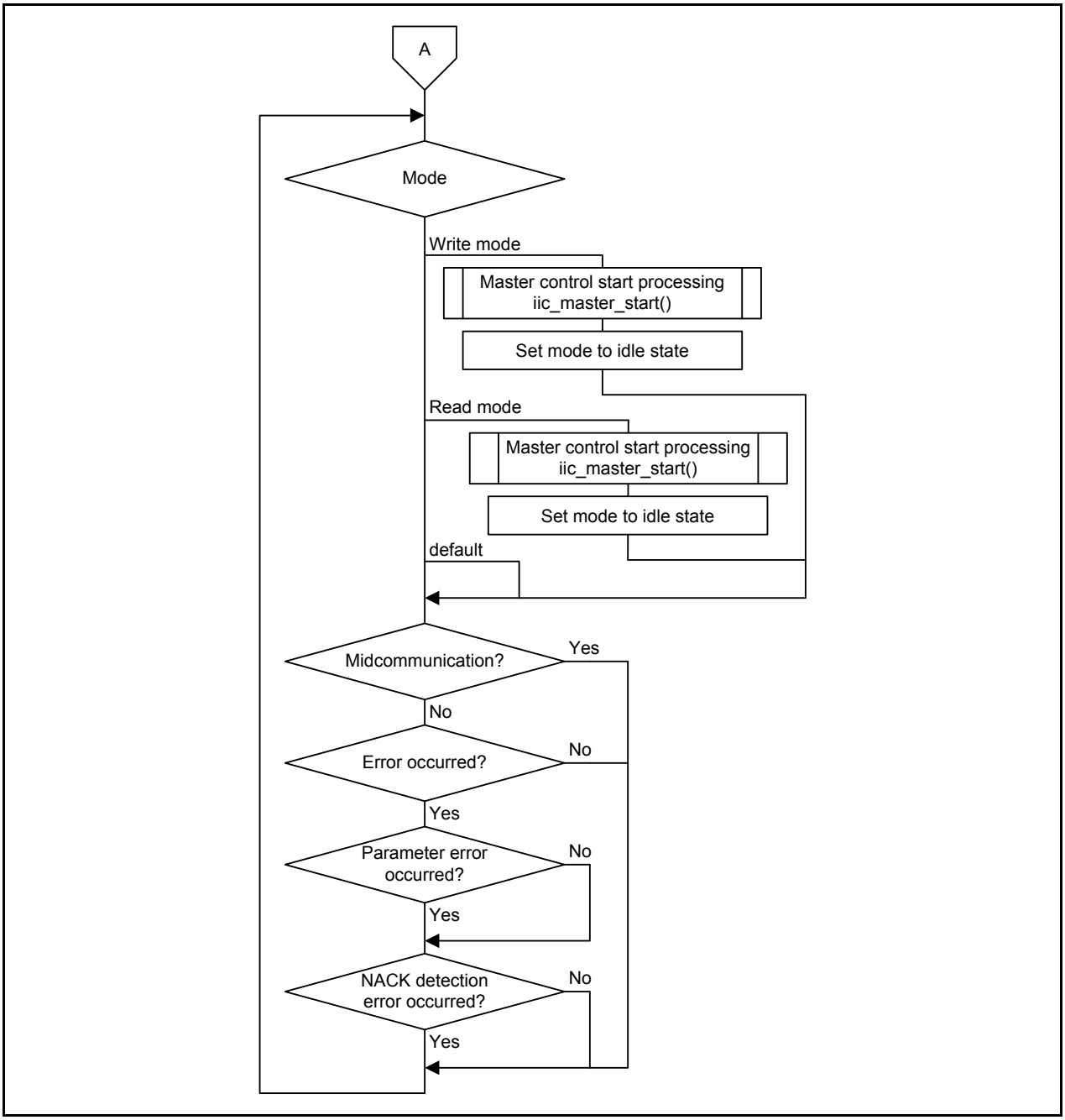


Figure 5.5 Main Processing (2/2)

5.7.2 UART2 Initial Setting

Figure 5.6 shows the UART2 Initial Setting.

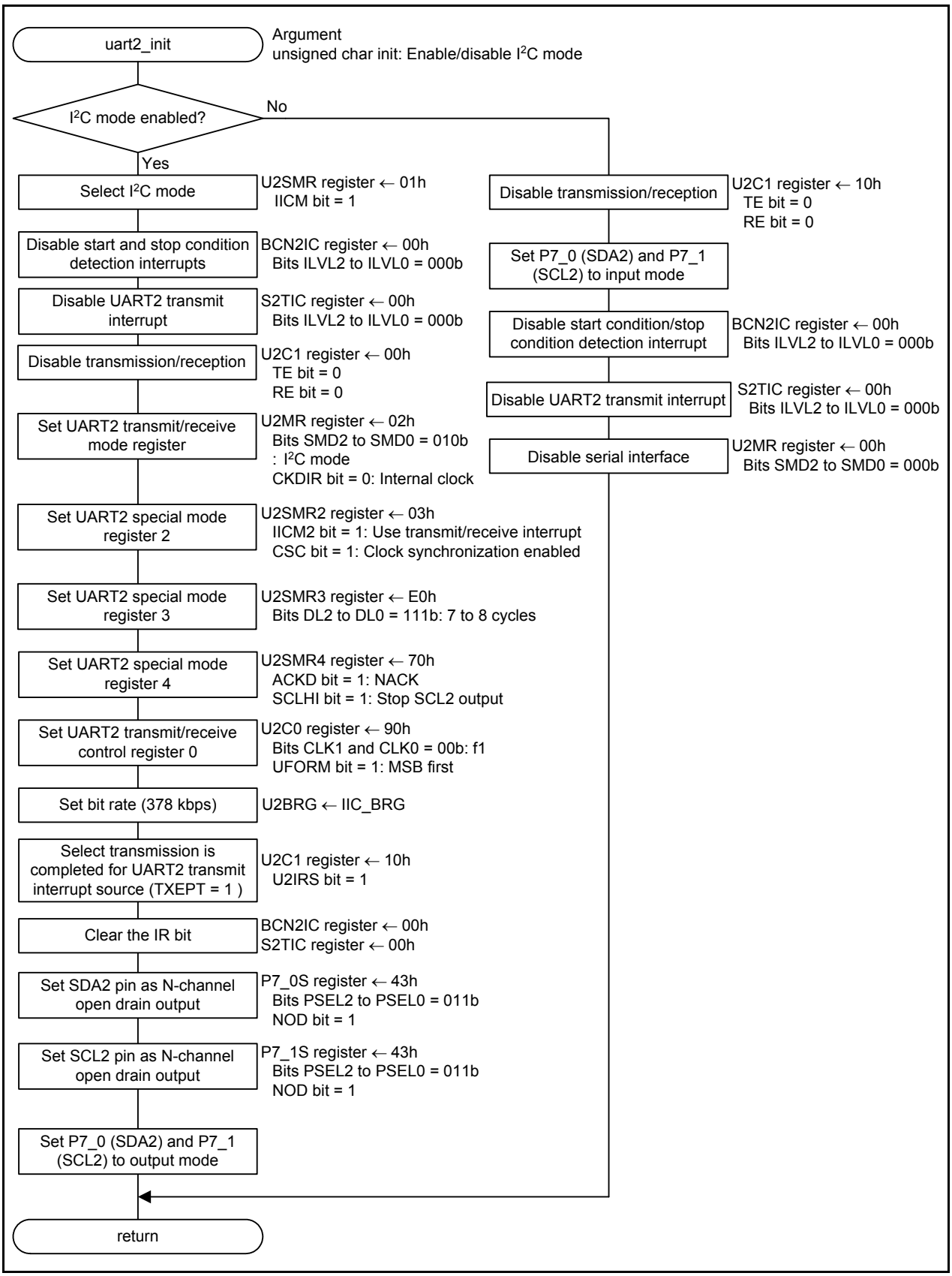


Figure 5.6 UART2 Initial Setting

5.7.3 Master Control Start Processing

Figure 5.7 shows Master Control Start Processing.

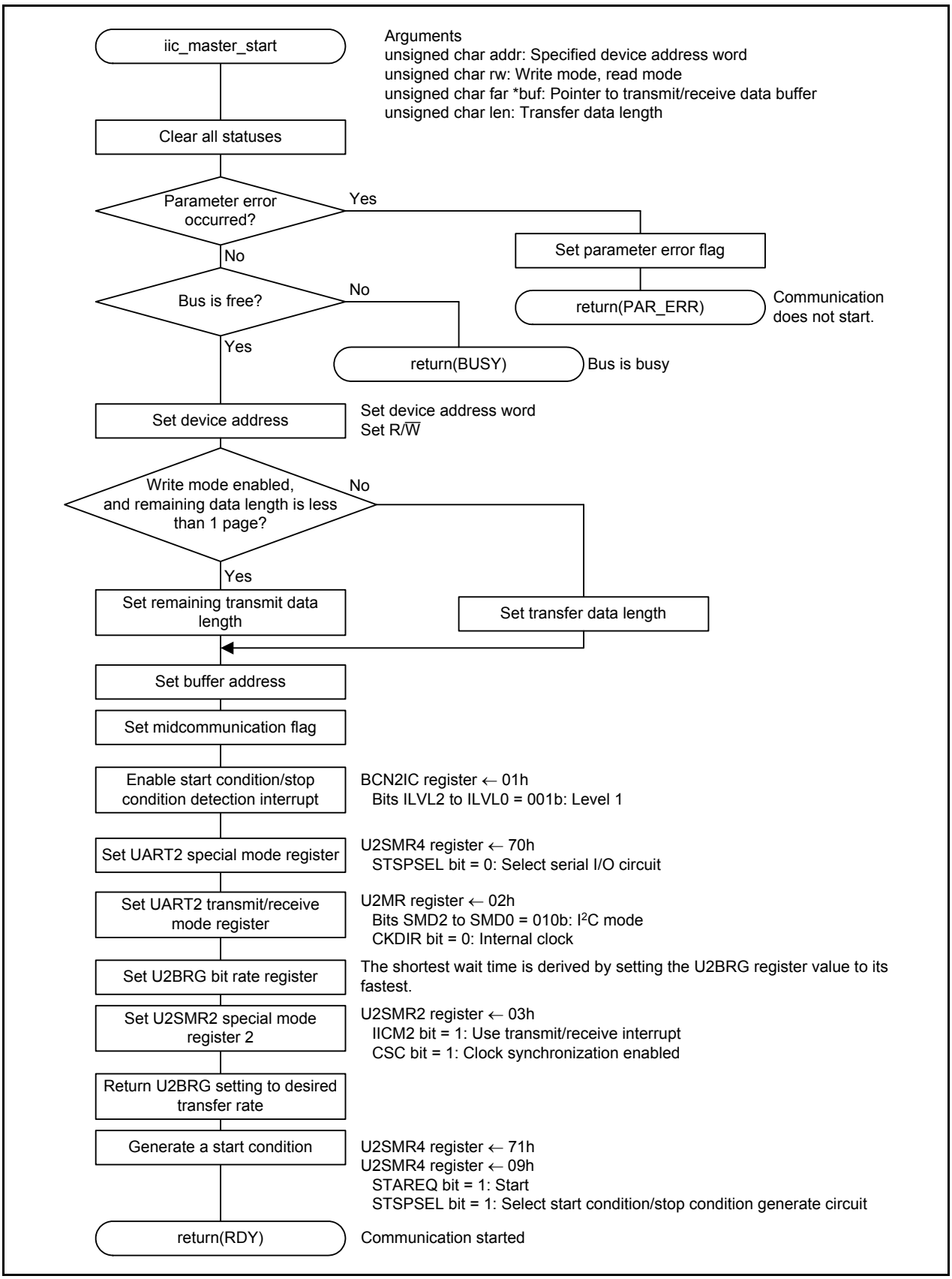


Figure 5.7 Master Control Start Processing

5.7.4 Start Condition/Stop Condition Detection Interrupt Handling

Figure 5.8 shows Start Condition/Stop Condition Detection Interrupt Handling.

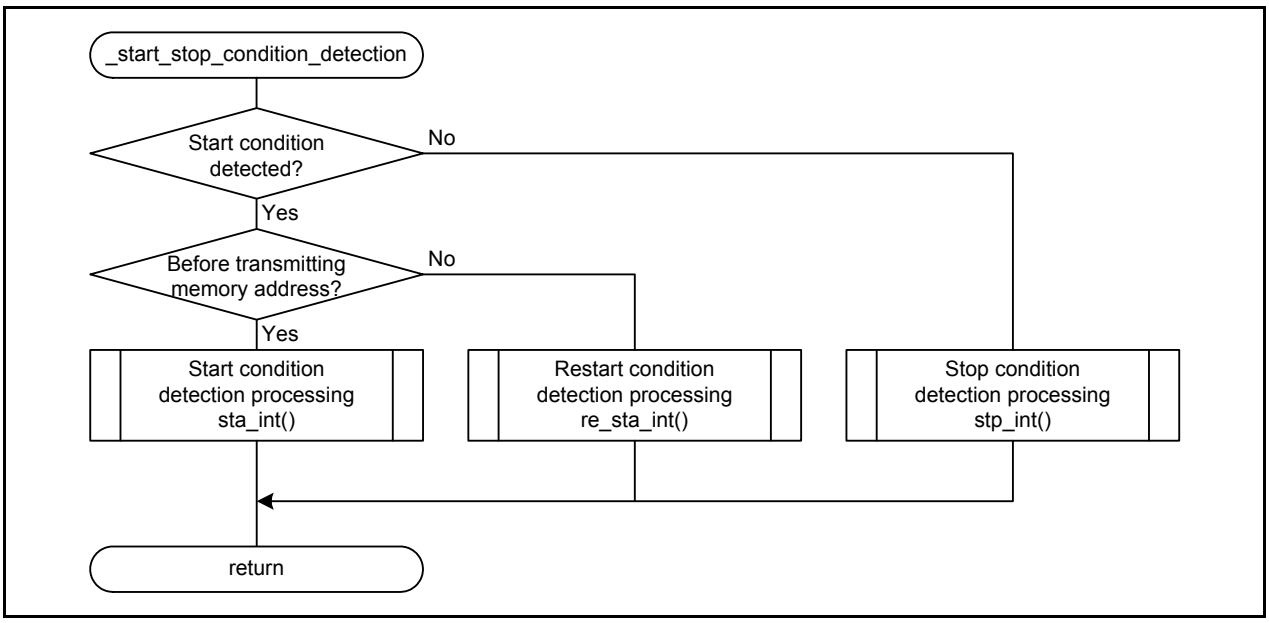


Figure 5.8 Start Condition/Stop Condition Detection Interrupt Handling

5.7.5 Start Condition Detection Processing

Figure 5.9 shows Start Condition Detection Processing.

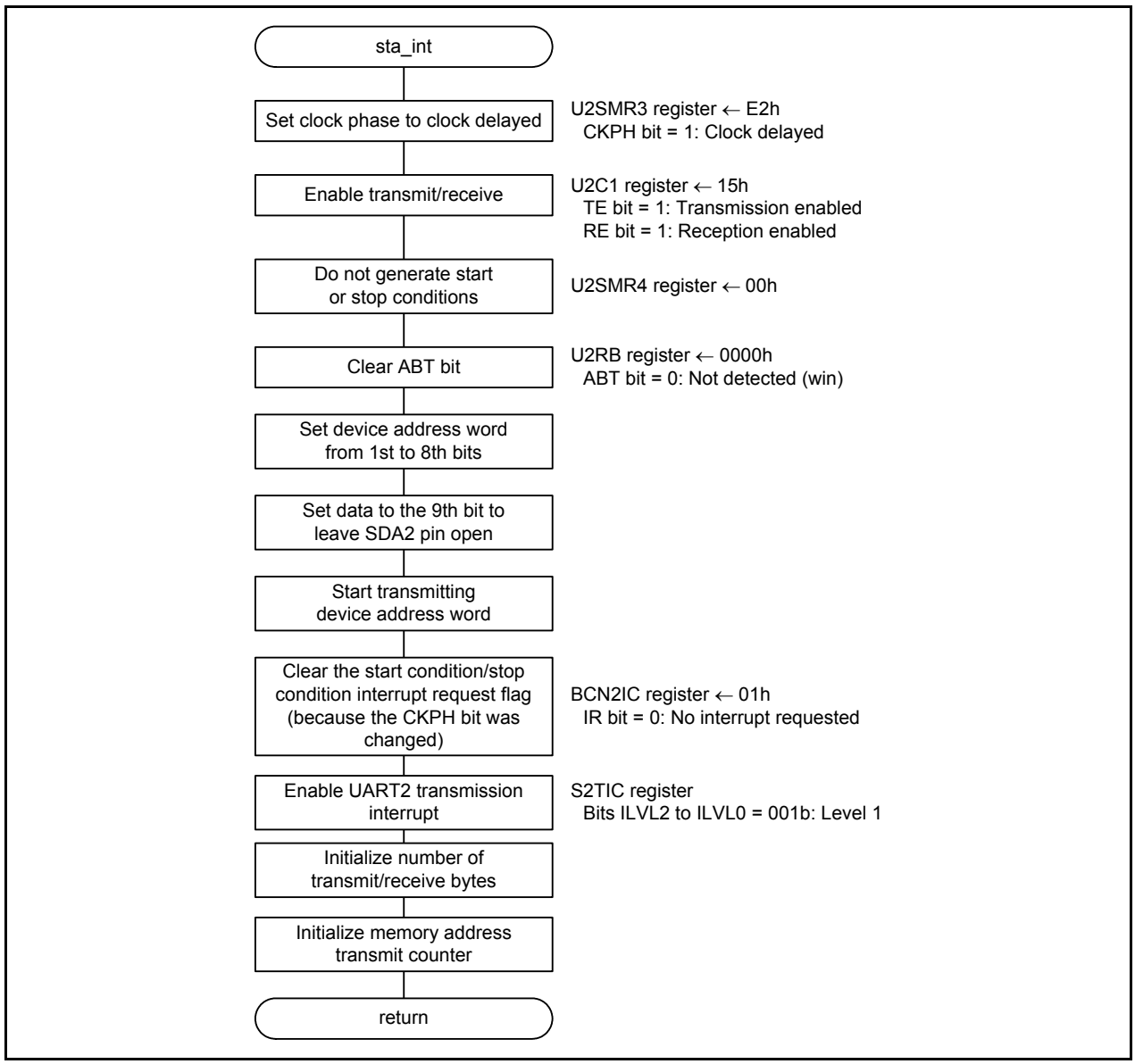


Figure 5.9 Start Condition Detection Processing

5.7.6 Stop Condition Detection Processing

Figure 5.10 and Figure 5.11 show Stop Condition Detection Processing.

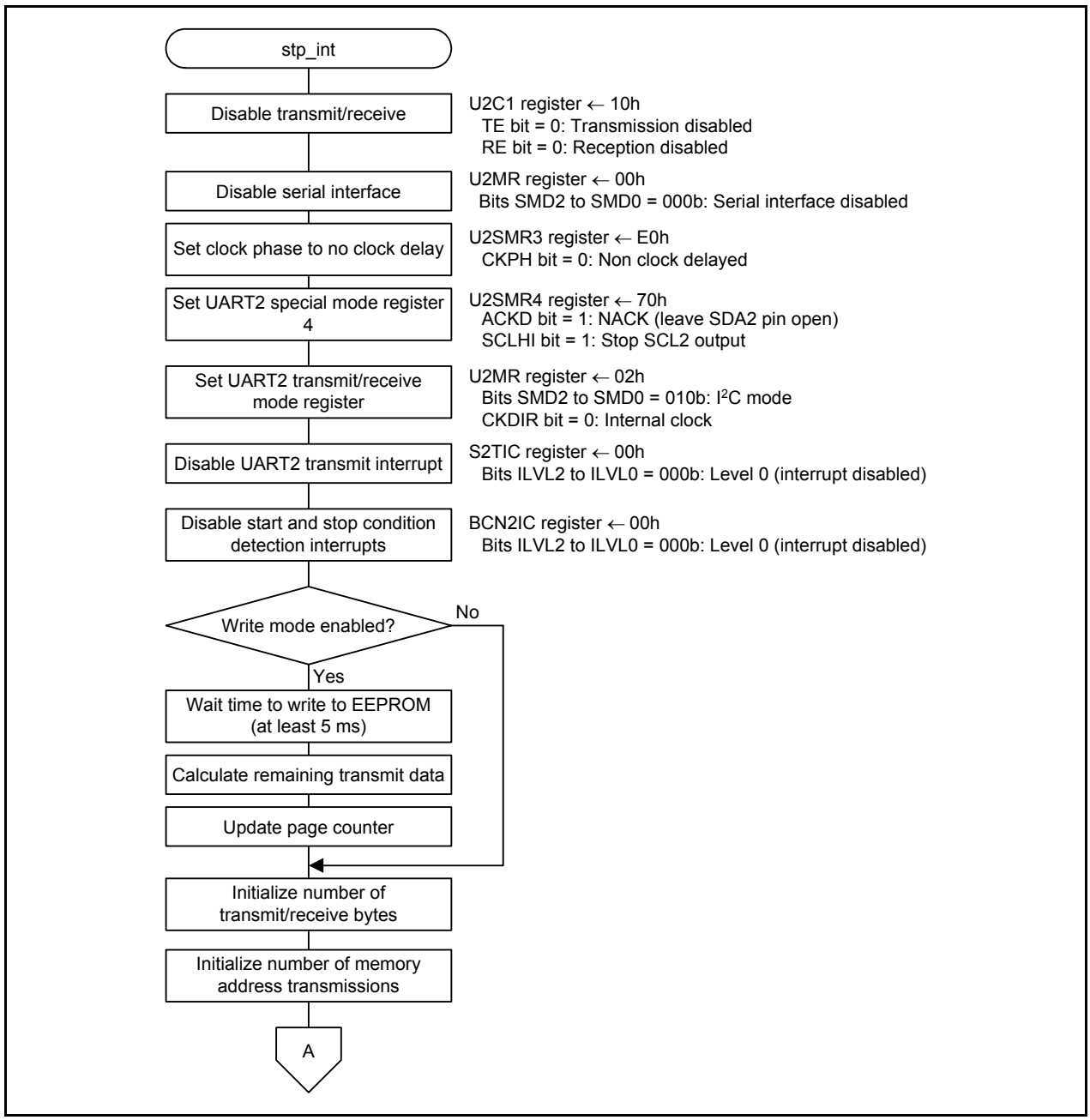


Figure 5.10 Stop Condition Detection Processing (1/2)

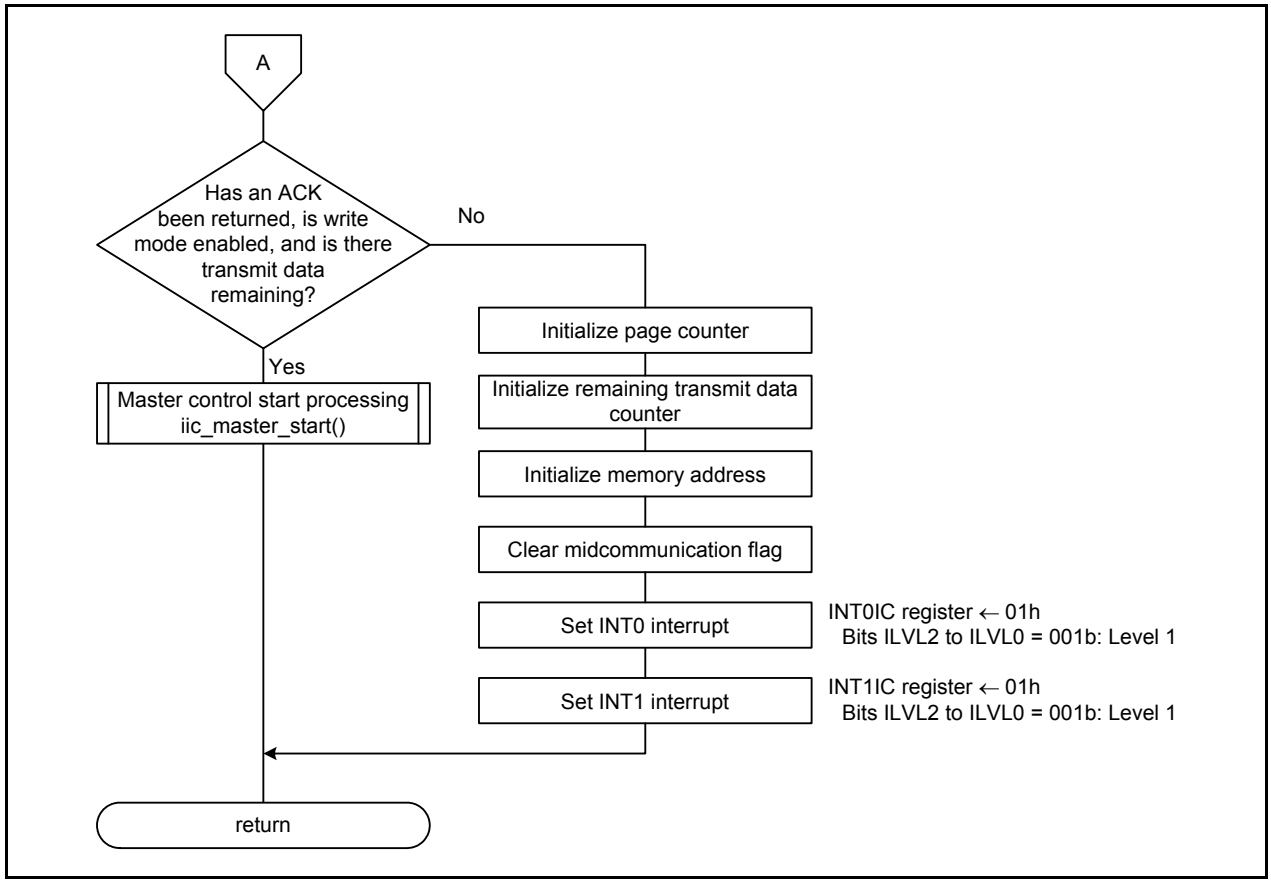


Figure 5.11 Stop Condition Detection Processing (2/2)

5.7.7 Restart Condition Detection Processing

Figure 5.12 shows Restart Condition Detection Processing.

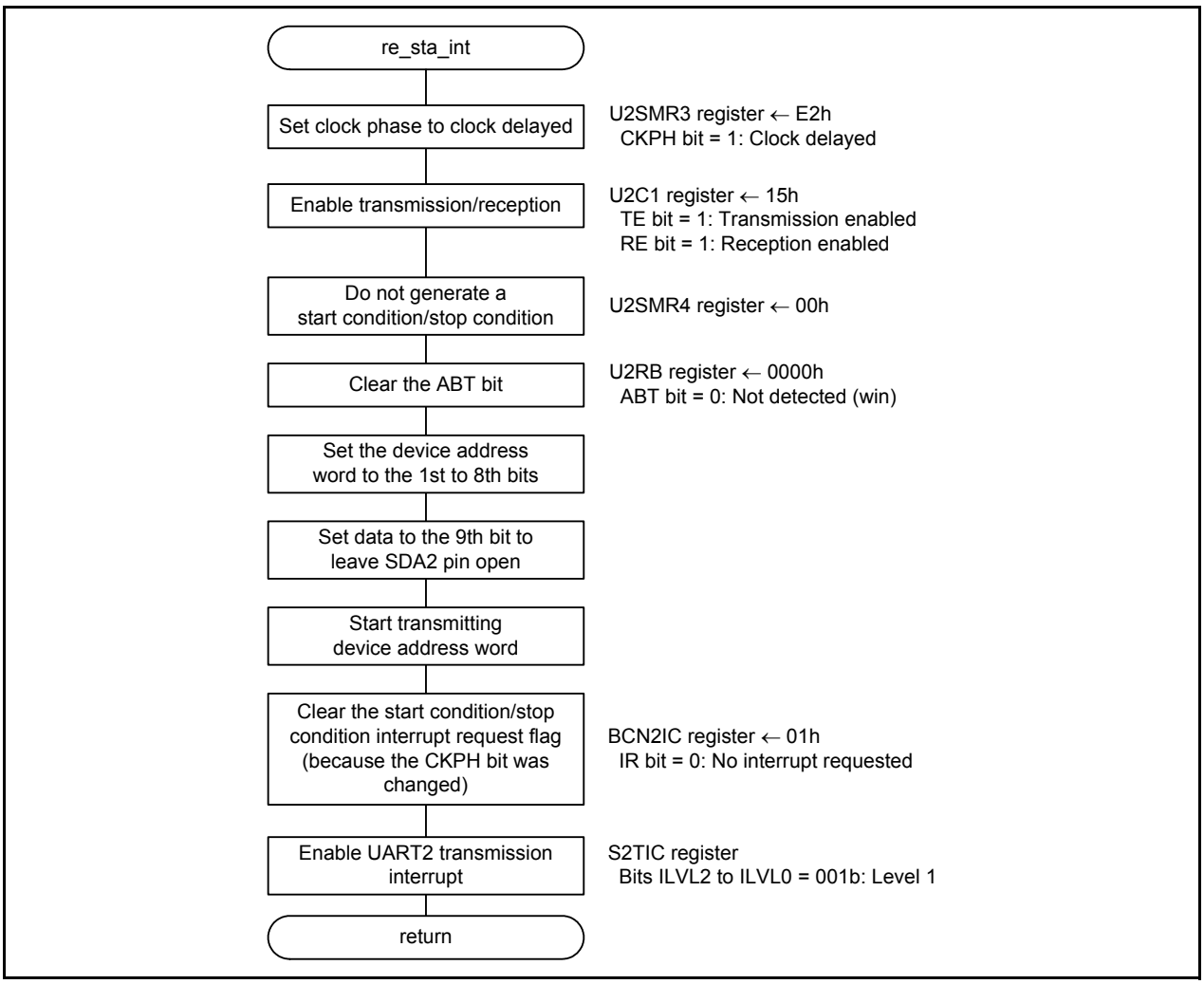


Figure 5.12 Restart Condition Detection Processing

5.7.8 UART2 Transmit Interrupt Handling

Figure 5.13 shows UART2 Transmit Interrupt Handling.

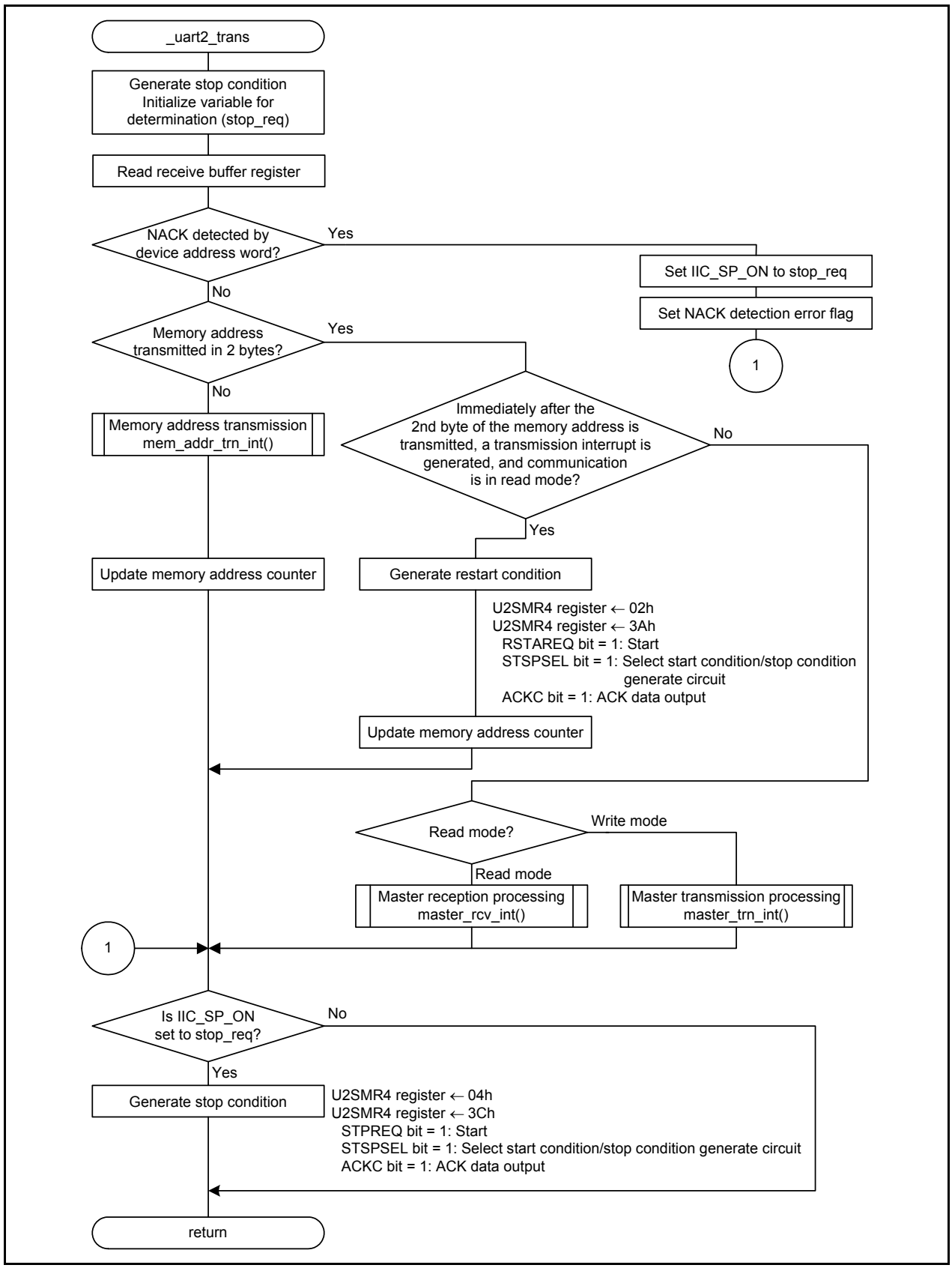


Figure 5.13 UART2 Transmit Interrupt Handling

5.7.9 Memory Address Transmission

Figure 5.14 shows Memory Address Transmission.

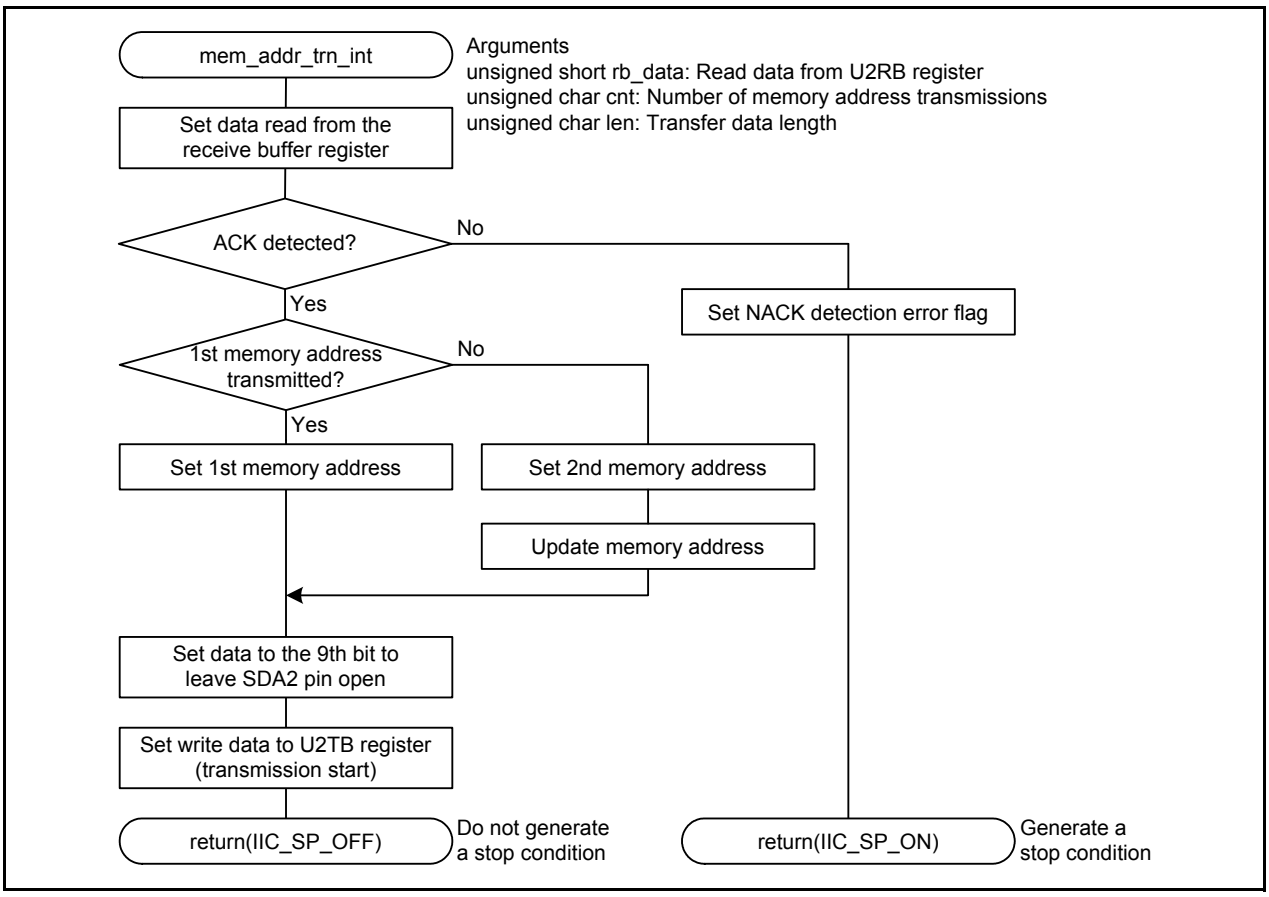


Figure 5.14 Memory Address Transmission

5.7.10 Master Transmission

Figure 5.15 shows Master Transmission.

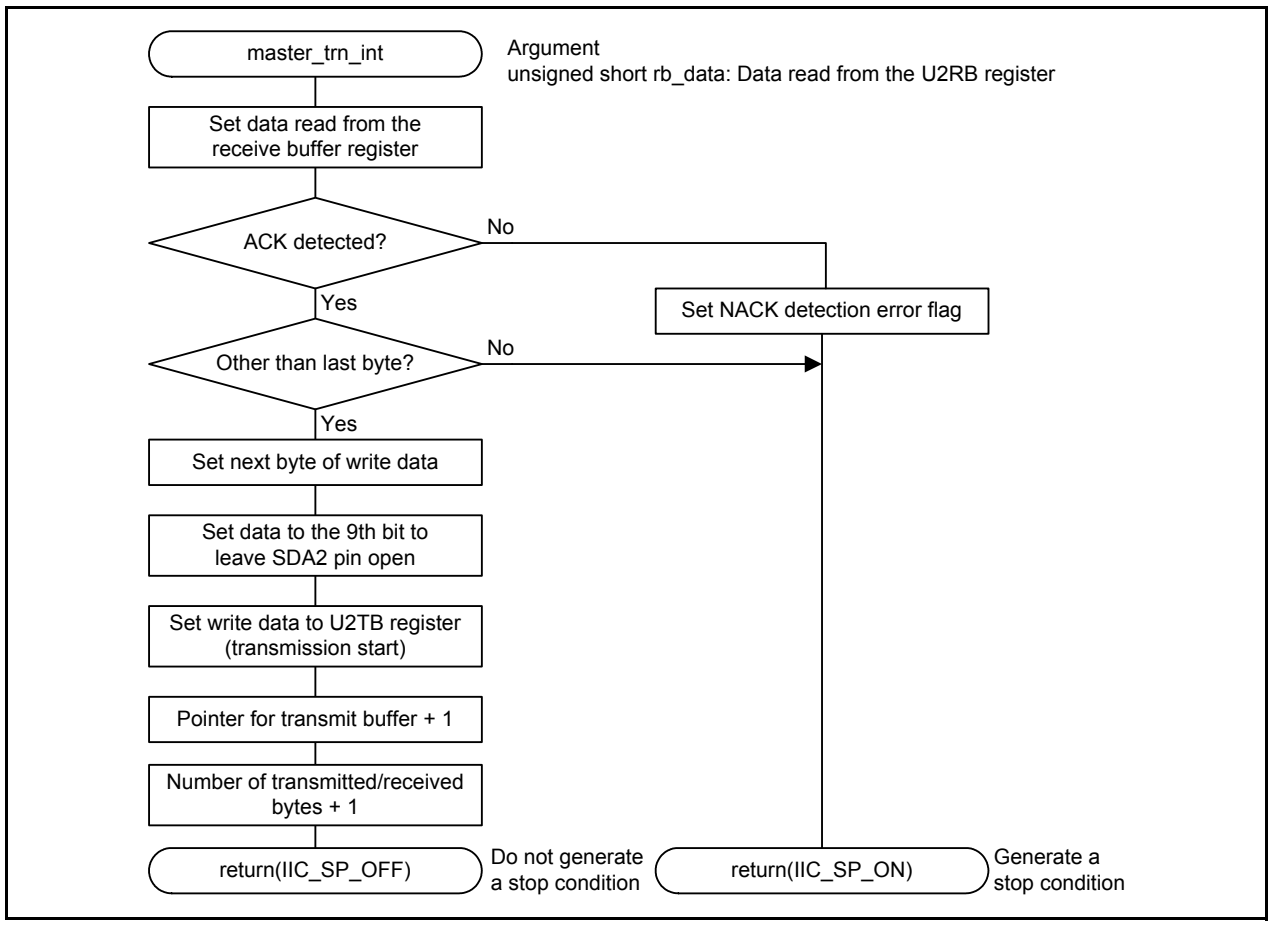


Figure 5.15 Master Transmission

5.7.11 Master Reception

Figure 5.16 shows Master Reception.

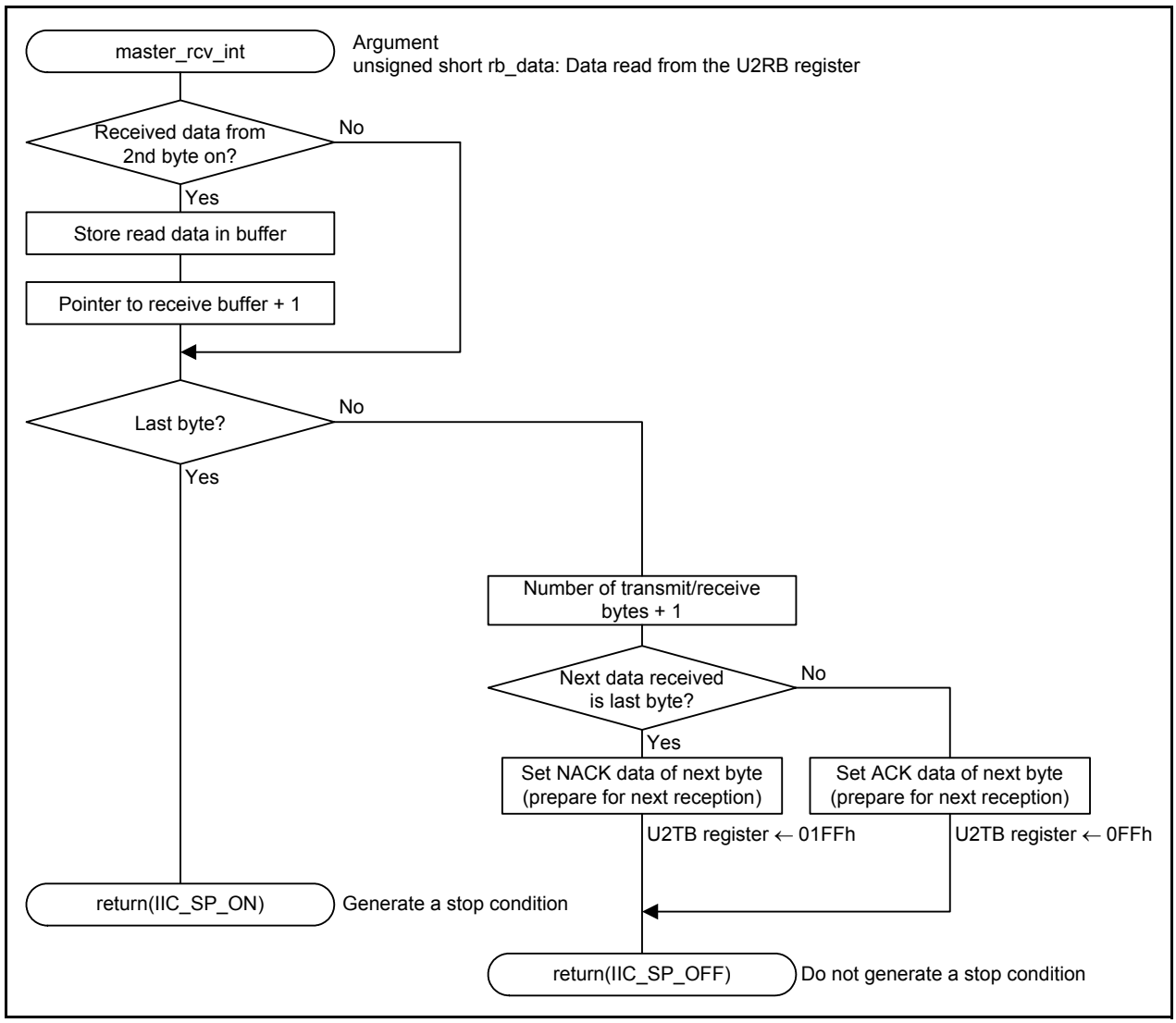


Figure 5.16 Master Reception

5.7.12 INT0 and INT1 Initial Setting

Figure 5.17 shows the INT0 and INT1 Initial Setting.

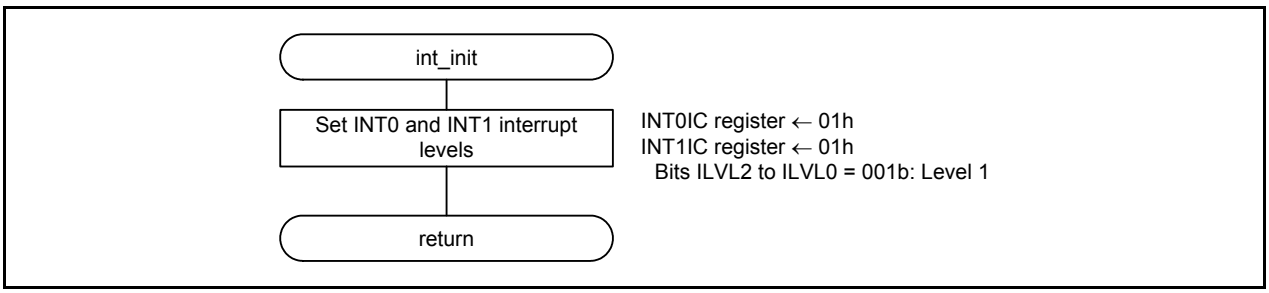


Figure 5.17 INT0 and INT1 Initial Setting

5.7.13 INT0 Interrupt Handling

Figure 5.18 shows INT0 Interrupt Handling.

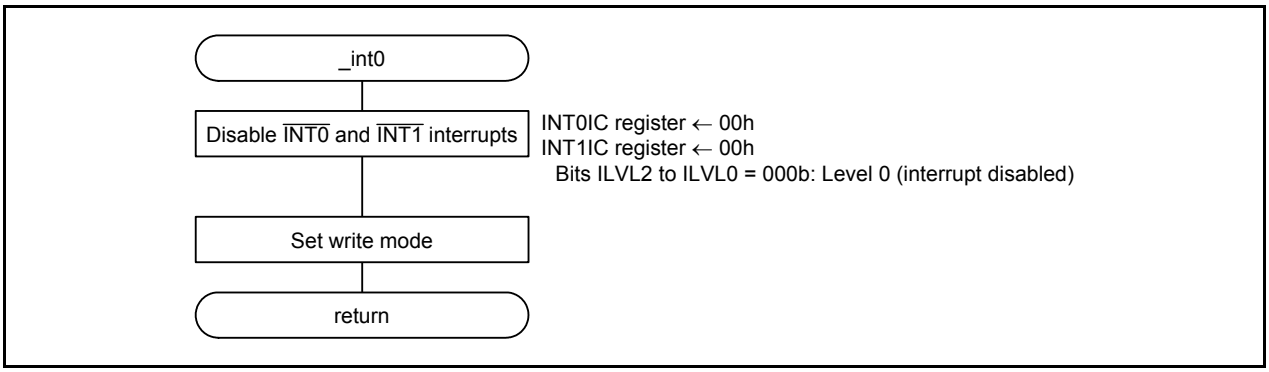


Figure 5.18 INT0 Interrupt Handling

5.7.14 INT1 Interrupt Handling

Figure 5.19 shows INT1 Interrupt Handling.

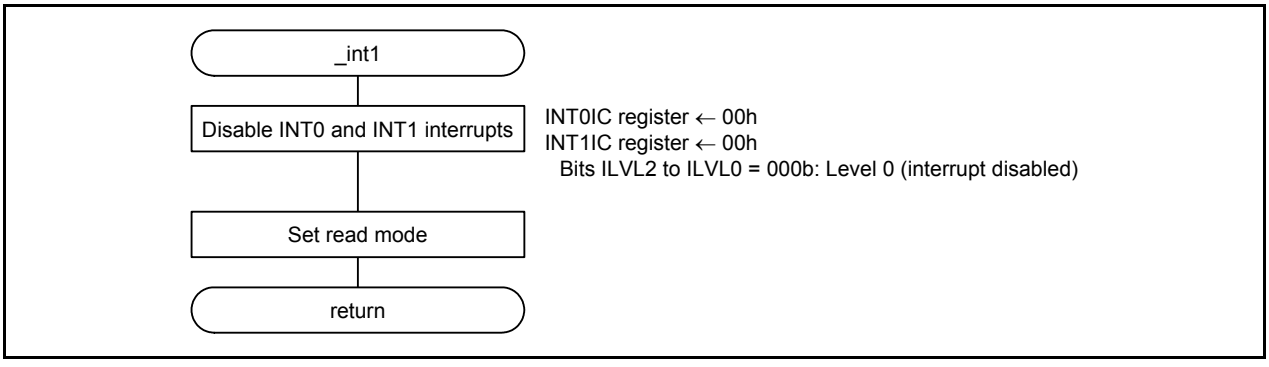


Figure 5.19 INT1 Interrupt Handling

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

R32C/116 Group User's Manual: Hardware Rev.1.10
R32C/117 Group User's Manual: Hardware Rev.1.10
R32C/118 Group User's Manual: Hardware Rev.1.10

The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C/100 Series C Compiler Package V.1.02
C Compiler User's Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com/

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

Revision History	R32C/100 Series EEPROM Control Using UARTi Special Mode 1 (I^2C Mode)

Rev.	Date	Description	
		Page	Summary
1.00	Oct. 28, 2011	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1