M16C/5M and M16C/57 Groups

Transmission/Reception Using the Serial Bus Interface in Standard Mode of 4-Wire Serial Bus Mode

R01AN0234EJ0100
Rev. 1.00
Mar. 30, 2012

1. Abstract

This document describes the method to transmit and receive data using 4-wire serial bus mode of the serial bus interface (SBI0) for the M16C/5M and M16C/57 Groups.

2. Introduction

The application example described in this document applies to the following microcomputers (MCUs):

- MCUs: M16C/5M and M16C/57 Groups

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

3. Application Example

The SBI0 has synchronous serial communication mode and 4-wire serial bus mode (standard mode and bidirectional mode), and the operation is selectable between master mode and slave mode.
This application note describes the method to transmit and receive data in 4-wire serial bus mode (standard mode) using two M16C/5M Group MCUs.

3.1 Outline

In this application example, the master device starts master transmission after the $\overline{\text{INT0}}$ interrupt request is received. The master device transmits one frame of data (16-bit) three times and the slave device receives them. Then the slave device transmits one frame of data (16-bit) three times and the master device receives them.
Figure 3.1 shows the Timing Chart.

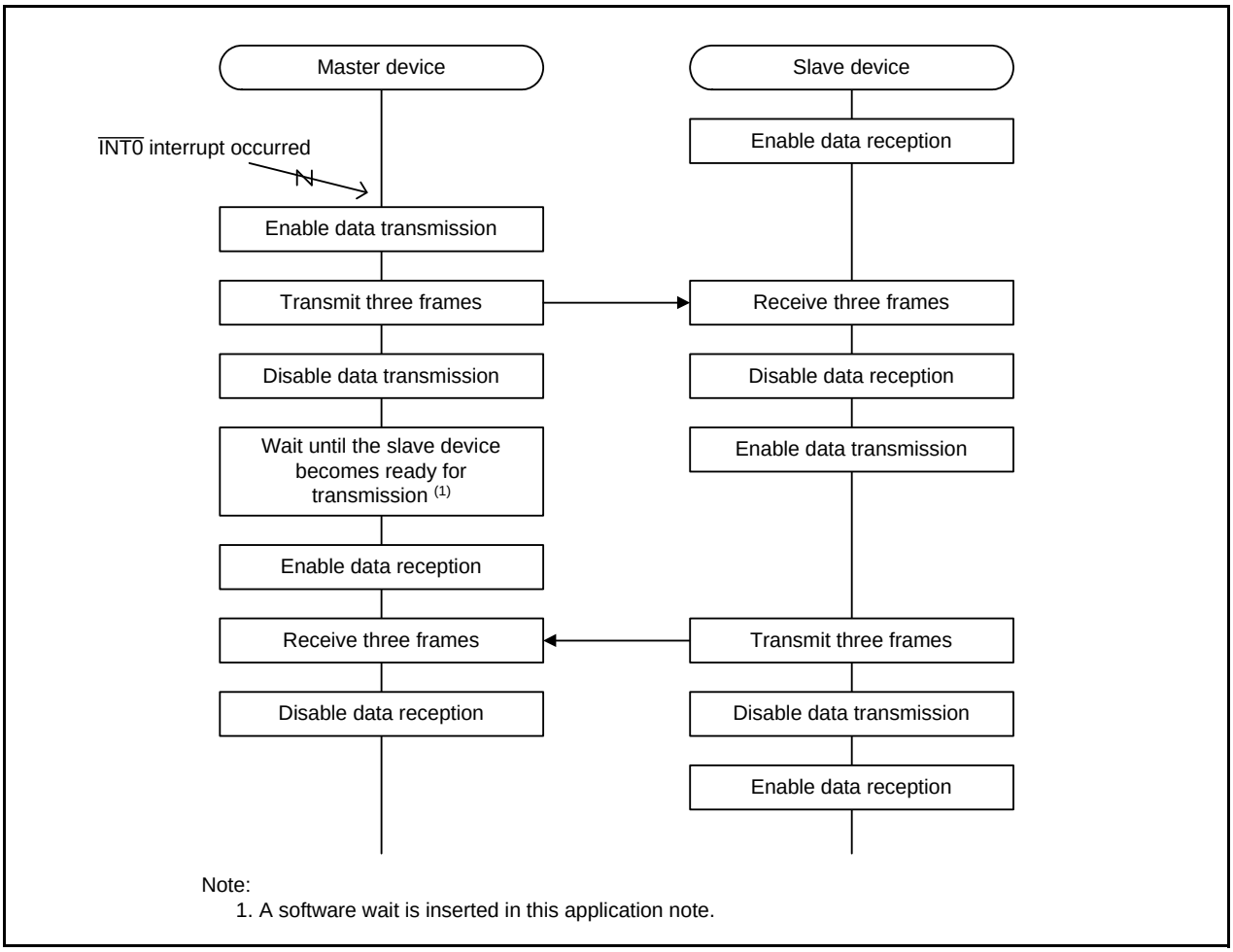


Figure 3.1 Timing Chart

3.2 Pins used

The statuses of pins used with 4-wire serial bus mode (standard mode) in this application note are listed in Table 3.1.

Table 3.1 Pin Statuses in 4-wire Serial Bus Mode (Standard Mode)

Device	Communication Status	Pin Status			
		$\overline{\text{SCS0}}$ (P3_3)	SSCK0 (P3_0)	SSI0 (P3_1)	SSO0 (P3_2)
Master device	Transmission	$\overline{\text{SCS0}}$ output	Clock output	(See Note 1)	Data output
	Reception	$\overline{\text{SCS0}}$ output	Clock output	Data input	(See Note 1)
Slave device	Reception	$\overline{\text{SCS0}}$ input	Clock input	(See Note 1)	Data input
	Transmission	$\overline{\text{SCS0}}$ input	Clock input	Data output	(See Note 1)

Note:
1. The pin can be used as a programmable I/O port. In this application note, the pin is used as an input port.

When transmitting, the master device outputs a low signal from the $\overline{\text{SCS0}}$ pin, then outputs the clock from the SSCK0 pin, and data from the SSO0 pin. When receiving, the master device outputs a low signal from the $\overline{\text{SCS0}}$ pin, then outputs the clock from the SSCK0 pin, and receives data input to the SSI0 pin.

When receiving, the slave device receives data input to the SSO0 pin when the clock is input to the SSCK0 pin while a low signal is input to the $\overline{\text{SCS0}}$ pin. When transmitting, the slave device outputs data from the SSI0 pin when the clock is input to the SSCK0 pin while a low signal is input to the $\overline{\text{SCS0}}$ pin.

Figure 3.2 shows a Connection Example of 4-Wire Serial Bus Mode (Standard Mode).

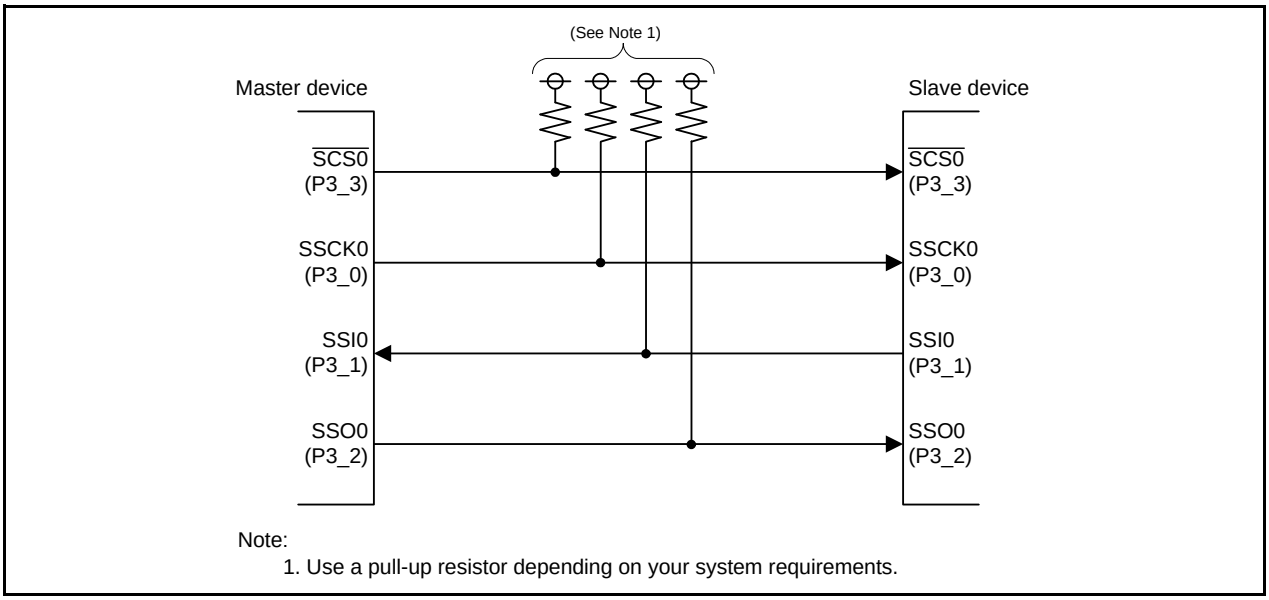


Figure 3.2 Connection Example of 4-Wire Serial Bus Mode (Standard Mode)

3.3 Specifications

This section describes the specifications of transmission and reception.

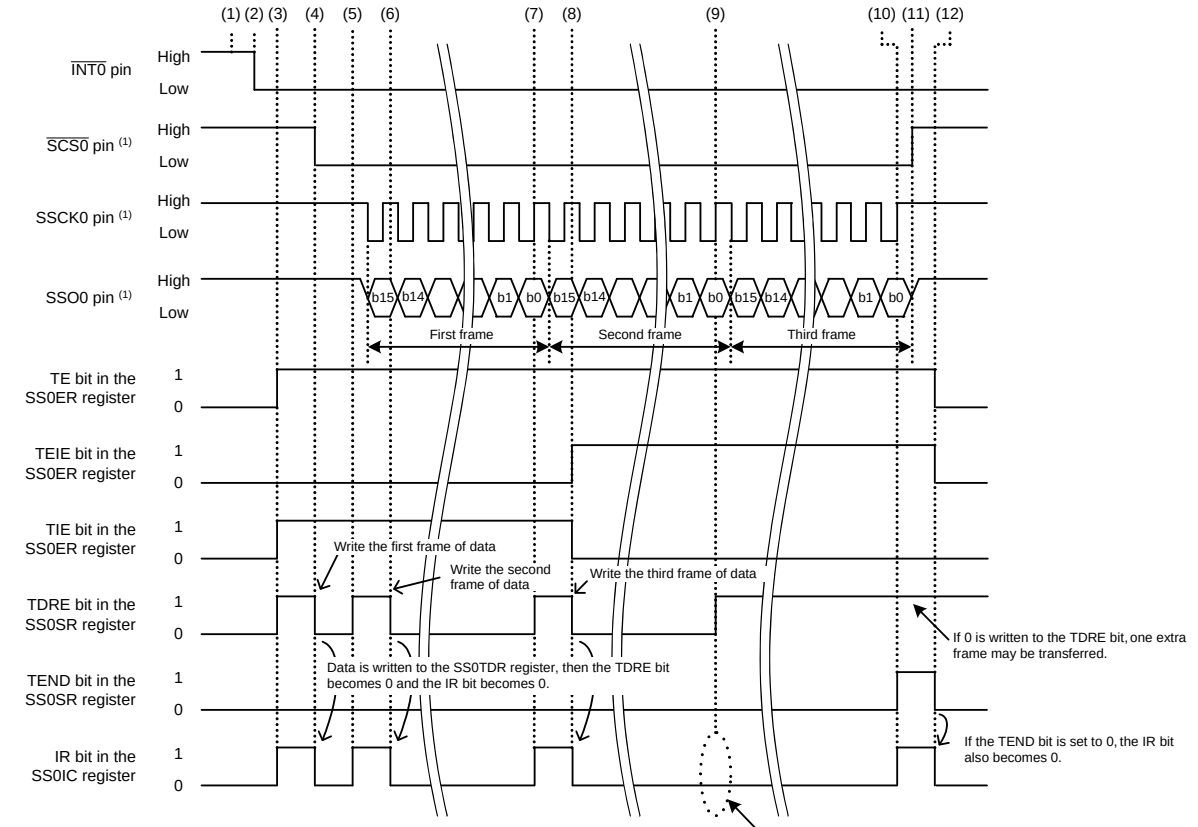
- Transmit/receive clock: f1/32
- Clock phase: Change data at odd edges and load data at even edges
- Clock polarity: High when clock is stopped
- Data transfer direction: Transmit and receive data with MSB first.
- Data transfer length: 16 bits
- Interrupts: Serial bus interface interrupt (interrupt priority level: 2)
  : $\overline{\text{INT0}}$ interrupt (interrupt priority level: 1)

3.4 Timing Diagrams

3.4.1 Master Transmission

Figure 3.3 shows the Master Transmission Timing Diagram.

(1) Initialize the serial bus interface.
(2) When a falling edge is input to the $\overline{\text{INT0}}$ pin, the $\overline{\text{INT0}}$ interrupt occurs.
(3) In the $\overline{\text{INT0}}$ interrupt handler, set the TE bit to 1 (enable data transmission) and the TIE bit to 1 (enable transmit data register empty interrupt request) in the SS0ER register. Then the TDRE bit in the SS0SR register becomes 1 (no data left in the SS0TDR register), and the transmit data register empty interrupt occurs.
(4) In the transmit data register empty interrupt handler, write the first frame data to the SS0TDR register. Then the TDRE bit becomes 0 (unsent data left in the SS0TDR register), and the IR bit in the SS0IC register becomes 0. The $\overline{\text{SCS0}}$ pin becomes low and the transmit/receive clock is output.
(5) When the first frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and the transmit data register empty interrupt occurs.
(6) In the transmit data register empty interrupt handler, write the second frame of data to the SS0TDR register.
(7) When the second frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and the transmit data register empty interrupt occurs.
(8) In the transmit data register empty interrupt handler, set the TIE bit to 0 (disable transmit data register empty interrupt request), the TEIE bit to 1 (enable transmit end interrupt request), then write the last frame of data to the SS0TDR register.
(9) When the last frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1, however the IR bit does not become 1.
(10) When the last bit is transferred, if the TDRE bit is 1, the TEND bit in the SS0SR register becomes 1 (transmission has completed), and the transmit end interrupt occurs.
(11) When the transmit/receive clock stops, the $\overline{\text{SCS0}}$ pin becomes high.
(12) In the transmit end interrupt handler, set the TEND bit to 0 (transmission continues), the TE bit to 0 (disable data transmission), and the TEIE bit to 0 (disable transmit end interrupt request). Then prepare for the receive operation.



The above assumes the following:

- The SSUMS bit in the SS0MR2 register is 1 (4-wire serial bus mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 1 (master mode).
- Bits BS3 to BS0 in the SS0BR register are 0000b (data transfer length is 16 bits).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges, load data at even edges).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (transfer/receive with MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 11b ($\overline{SCS0}$ pin as output).

Note:

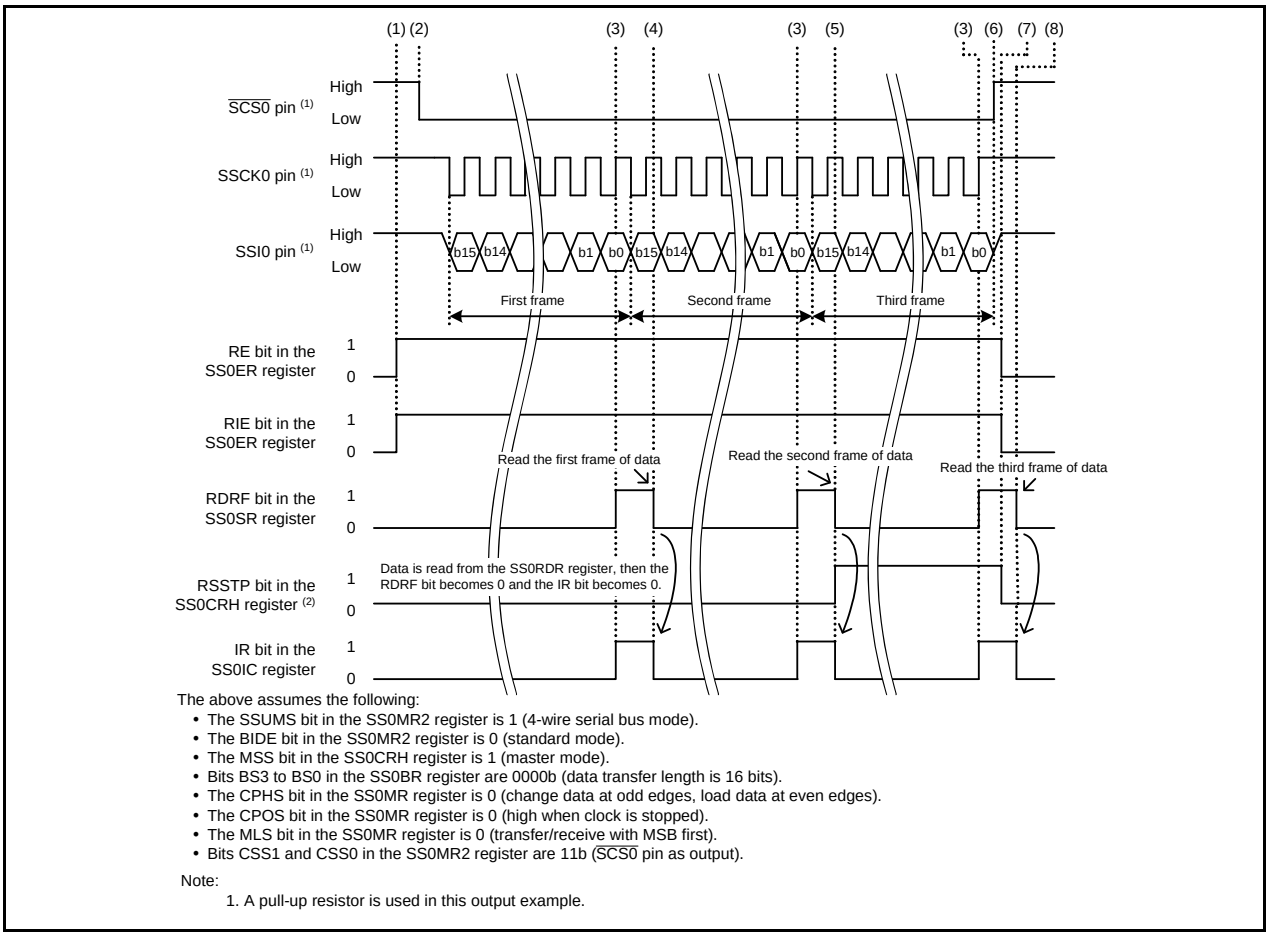
1. A pull-up resistor is used in this output example.

Figure 3.3 Master Transmission Timing Diagram

3.4.2 Master Reception

Figure 3.4 shows the Master Reception Timing Diagram.

(1) Set the RE bit to 1 (enable data reception), and the RIE bit to 1 (enable receive data register full or overrun error interrupt request) in the SS0ER register (the timing is the same as (12) in 3.4.1 Master Transmission).
(2) When the SS0RDR register is dummy read, the $\overline{SCS0}$ pin becomes low and the transmit/receive clock is output.
(3) When one frame of data is received, the RDRF bit in the SS0SR register becomes 1 (received data left in the SS0RDR register) and the received data is stored in the SS0RDR register. Then the receive data register full interrupt occurs.
(4) In the receive data register full interrupt handler, read the received data from the SS0RDR register. Then the RDRF bit becomes 0 (no data left in the SS0RDR register). Thus the IR bit in the SS0IC register becomes 0.
(5) In the receive data register full interrupt handler, the RSSTP bit in the SS0CRH register is set to 1 (end receive operation after receiving the current frame) to receive the last frame. Read the received data from the SS0RDR register.
(6) When the transmit/receive clock stops, the $\overline{SCS0}$ pin becomes high.
(7) In the receive data register full interrupt handler, set the RSSTP bit to 0 (continue receive operation after receiving the current frame), the RE bit to 0 (disable data reception), and the RIE bit to 0 (disable receive data register full or overrun error interrupt request), then read the last frame.
(8) Read the received data from the SS0RDR register.



The above assumes the following:
- The SSUMS bit in the SS0MR2 register is 1 (4-wire serial bus mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 1 (master mode).
- Bits BS3 to BS0 in the SS0BR register are 0000b (data transfer length is 16 bits).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges, load data at even edges).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (transfer/receive with MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 11b ($\overline{SCS0}$ pin as output).

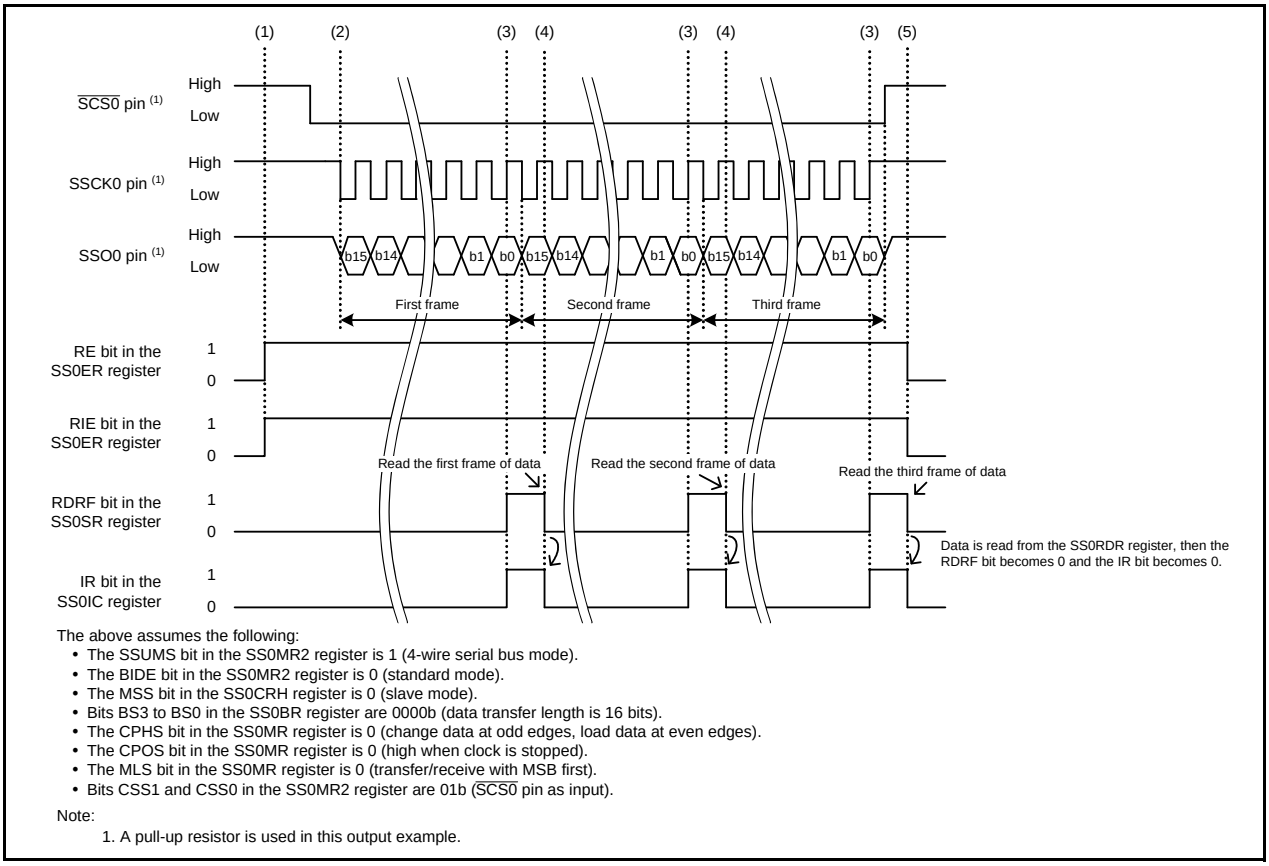
Note:
1. A pull-up resistor is used in this output example.

Figure 3.4 Master Reception Timing Diagram

3.4.3 Slave Reception

Figure 3.5 shows the Slave Reception Timing Diagram.

(1) Initialize the serial bus interface. Set the RE bit to 1 (enable data reception) and the RIE bit to 1 (enable receive data register full or overrun error interrupt request) in the SS0ER register. Dummy read the SS0RDR register.
(2) Receive data when the clock is input to the SSCK0 pin while the $\overline{\text{SCS0}}$ pin is low.
(3) When one frame of data is received, the RDRF bit in the SS0SR register becomes 1 (received data left in the SS0RDR register), and the received data is stored in the SS0RDR register. Then the receive data register full interrupt occurs.
(4) In the receive data register full interrupt handler, read the received data from the SS0RDR register. Then the RDRF bit becomes 0 (no data left in the SS0RDR register) and the IR bit in the SS0IC register becomes 0.
(5) When the last frame is received, set the RIE bit to 0 (disable receive data register full or overrun error interrupt request) and the RE bit to 0 (disable data reception). Then prepare for a transmit operation.



The above assumes the following:
- The SSUMS bit in the SS0MR2 register is 1 (4-wire serial bus mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 0 (slave mode).
- Bits BS3 to BS0 in the SS0BR register are 0000b (data transfer length is 16 bits).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges, load data at even edges).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (transfer/receive with MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 01b ($\overline{\text{SCS0}}$ pin as input).

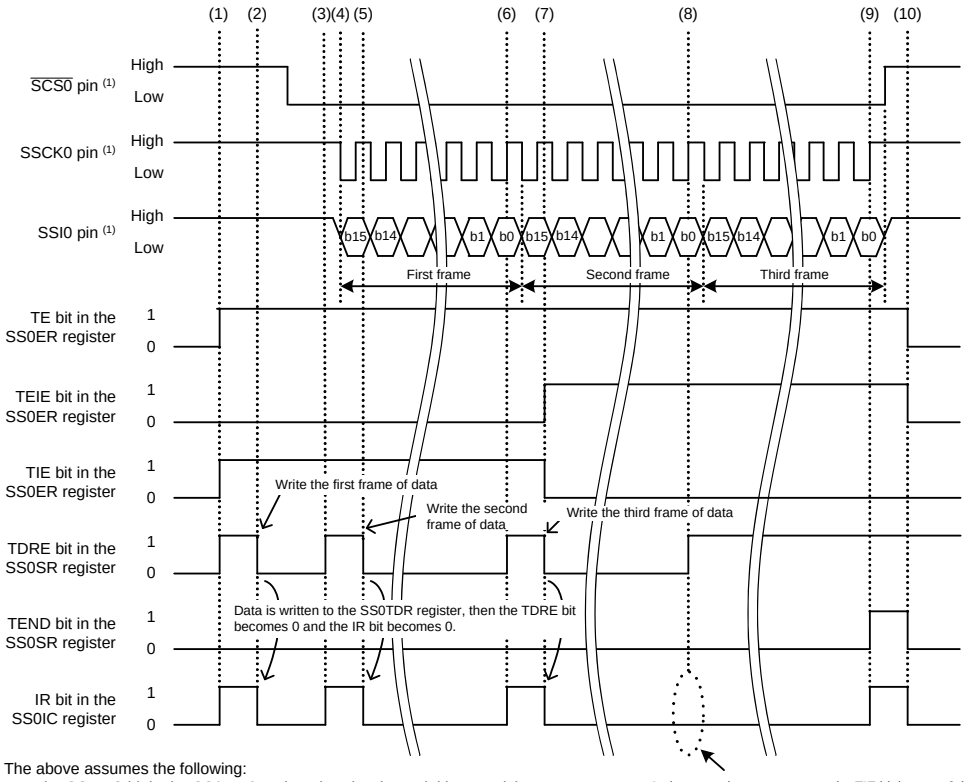
Note:
1. A pull-up resistor is used in this output example.

Figure 3.5 Slave Reception Timing Diagram

3.4.4 Slave Transmission

Figure 3.6 shows the Slave Transmission Timing Diagram.

(1) Set the TIE bit to 1 (enable transmit data register empty interrupt request) and the TE bit to 1 (enable data transmission) in the SS0ER register. Then the TDRE bit in the SS0SR register becomes 1 (no data left in the SS0TDR register) and the transmit data register empty interrupt occurs. (The timing is the same as (5) in 3.4.3 Slave Reception.)
(2) In the transmit data register empty interrupt handler, write the first frame of data to the SS0TDR register. Then the TDRE bit becomes 0 (unsent data left in the SS0TDR register) and the IR bit in the SS0IC register becomes 0.
(3) When the first frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and the transmit data register empty interrupt occurs.
(4) Transmit the data when the clock is input to the SSCK0 pin while the $\overline{\text{SCS0}}$ pin is low.
(5) In the transmit data register empty interrupt handler, write the second frame of data to the SS0TDR register.
(6) When the second frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and transmit data register empty interrupt occurs.
(7) In the transmit data register empty interrupt handler, set the TIE bit to 0 (disable transmit data register empty interrupt request) and the TEIE bit to 1 (enable transmit end interrupt request) in the SS0ER register. Write the last frame of data to the SS0TDR register.
(8) When the last frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1. However because the TIE bit is 0, the transmit data register empty interrupt does not occur.
(9) When the last bit is transferred and the TDRE bit is 1, the TEND bit in the SS0SR register becomes 1 (transmission has completed) and the transmit end interrupt occurs.
(10) In the transmit end interrupt handler, set the TEND bit to 0 (transmission continues), the TE bit to 0 (disable data transmission), and the TEIE bit to 0 (disable transmit end interrupt request).



The above assumes the following:

- The SSUMS bit in the SS0MR2 register is 1 (4-wire serial bus mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 0 (slave mode).
- Bits BS3 to BS0 in the SS0BR register are 0000b (data transfer length is 16 bits).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges, load data at even edges).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (transfer/receive with MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 01b ($\overline{SCS0}$ pin as input).

Note:

1. A pull-up resistor is used in this output example.

Figure 3.6 Slave Transmission Timing Diagram

3.5 Required Memory Size

Table 3.2 lists the Required Memory Size and Table 3.3 lists the RAM Used and Definition.

Table 3.2 Required Memory Size

Memory Used	Size		Remarks
	Master	Slave	
ROM (program only)	155 bytes	163 bytes	In modules r01an0234_src_master.c or r01an0234_src_slave.c only
RAM	15 bytes	15 bytes	
Maximum user stack usage	23 bytes	23 bytes	main function: 10 bytes (master) 10 bytes (slave) peripheral_init function: 7 bytes (master) 7 bytes (slave)
Maximum interrupt stack usage	21 bytes	22 bytes	

The required memory size varies depending on the C compiler version and compile options.

Table 3.3 RAM Used and Definition

Type	Variable Name	Size	Contents
unsigned short	snd_data[3]	6 bytes	Buffer to store the transmit data
unsigned char	snd_cnt	1 byte	Transmit counter
unsigned short	rcv_dada[3]	6 bytes	Buffer to store the received data
unsigned char	rcv_cnt	1 byte	Receive counter
unsigned char	f_snd_rcv	1 byte	Transmit/receive flag

4. Function Tables

4.1 Function Tables for Master Transmit/Receive Program

Declaration	void peripheral_init(void)		
Outline	Peripheral function initialization		
Argument	Name		Explanation
	None		—
Variable used (global)	Name		Usage
	None		—
Returned value	Type	Value	Explanation
	None	—	—
Function	Configure the serial bus interface for use as the master device.		

Declaration	void _int0(void)		
Outline	$\overline{\text{INT0}}$ interrupt handler		
Argument	Name		Explanation
	None		—
Variable used (global)	Name		Usage
	unsigned char snd_cnt		Transmit counter
	unsigned char f_snd_rcv		Transmit/receive flag
Returned value	Type	Value	Explanation
	None	—	—
Function	Set to enable transmit operation when a conflict error does not occur.		

Declaration	void _sbi(void)		
Outline	Serial bus interface interrupt handler		
Argument	Name		Explanation
	None		—
Variables used (global)	Name		Usage
	unsigned short snd_data[]		Buffer to store the transmit data
	unsigned char snd_cnt		Transmit counter
	unsigned short rcv_data[]		Buffer to store the received data
	unsigned char rcv_cnt		Receive counter
	unsigned char f_snd_rcv		Transmit/receive flag
Returned value	Type	Value	Explanation
	None	—	—
Function	Read the transmit/receive flag and determine the interrupt source. Perform the appropriate operation from the following depending on the determined interrupt source. • Transmit register empty interrupt Write transmit data to the SS0TDR register. When setting the last data, disable the transmit register empty interrupt and enable the transmit end interrupt. • Transmit end interrupt Disable the transmit end interrupt and transmit operation. Prepare for the receive operation. • Overrun error interrupt Disable receive operation. • Receive data register full interrupt Read the received data in the SS0RDR register. When the second to last frame of data is received, set the RSSTP bit in the SS0CRH register to 1 (end receive operation (stop the receive clock)), then read the received data. When the last frame is received, set the RE bit to 0 (disable data reception) in the SS0ER register and the RSSTP bit to 0 (continue receive operation (output the receive clock)), then read the received data.		

4.2 Function Tables for the Slave Transmit/Receive Program

Declaration	void peripheral_init(void)		
Outline	Peripheral function initialization		
Argument	Name		Explanation
	None		—
Variables used (global)	Name		Usage
	unsigned char rcv_cnt		Receive counter
	unsigned char f_snd_rcv		Transmit/receive flag
Returned value	Type	Value	Explanation
	None	—	—
Function	Configure the serial bus interface for use as the slave device.		

Declaration	void _sbi(void)		
Outline	Serial bus interface interrupt processing		
Argument	Name		Explanation
	None		—
Variables used (global)	Name		Usage
	unsigned short snd_data[]		Buffer to store the transmit data
	unsigned char snd_cnt		Transmit counter
	unsigned short rcv_data[]		Buffer to store the received data
	unsigned char rcv_cnt		Receive counter
	unsigned char f_snd_rcv		Transmit/receive flag
Returned value	Type	Value	Explanation
	None	—	—
Function	Read the transmit/receive flag and determine the interrupt source. Perform the appropriate operation from the following depending on the determined interrupt source. • Transmit register empty interrupt Write transmit data to the SS0TDR register. When setting the last data, disable the transmit register empty interrupt and enable the transmit end interrupt. • Transmit end interrupt Disable the transmit end interrupt and transmit operation. Prepare for the receive operation. • Overrun error interrupt Disable the receive operation. • Receive data register full interrupt Read the received data in the SS0RDR register. When the last data is received, disable the receive operation and receive interrupt, and prepare for the transmit operation.		

5. Flowcharts

5.1 Master Transmit/Receive Program

5.1.1 Main Function

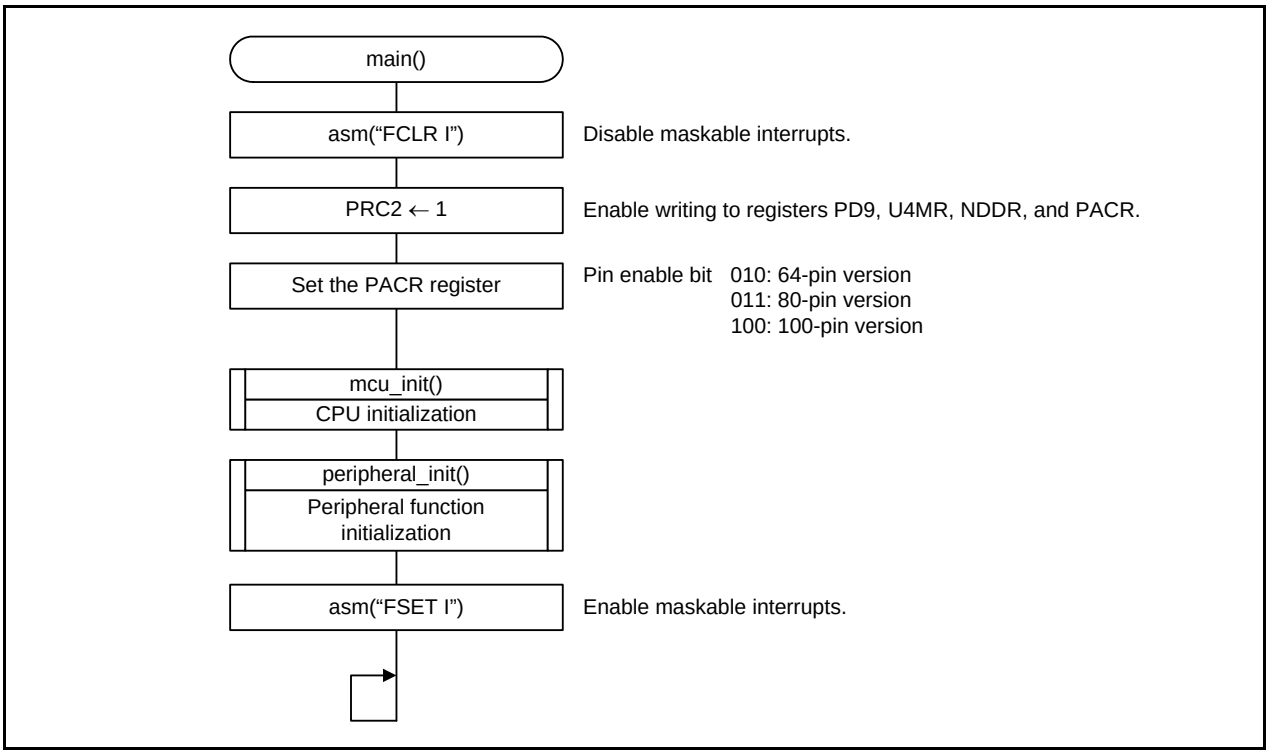


Figure 5.1 Main Function

5.1.2 Peripheral Function Initialization

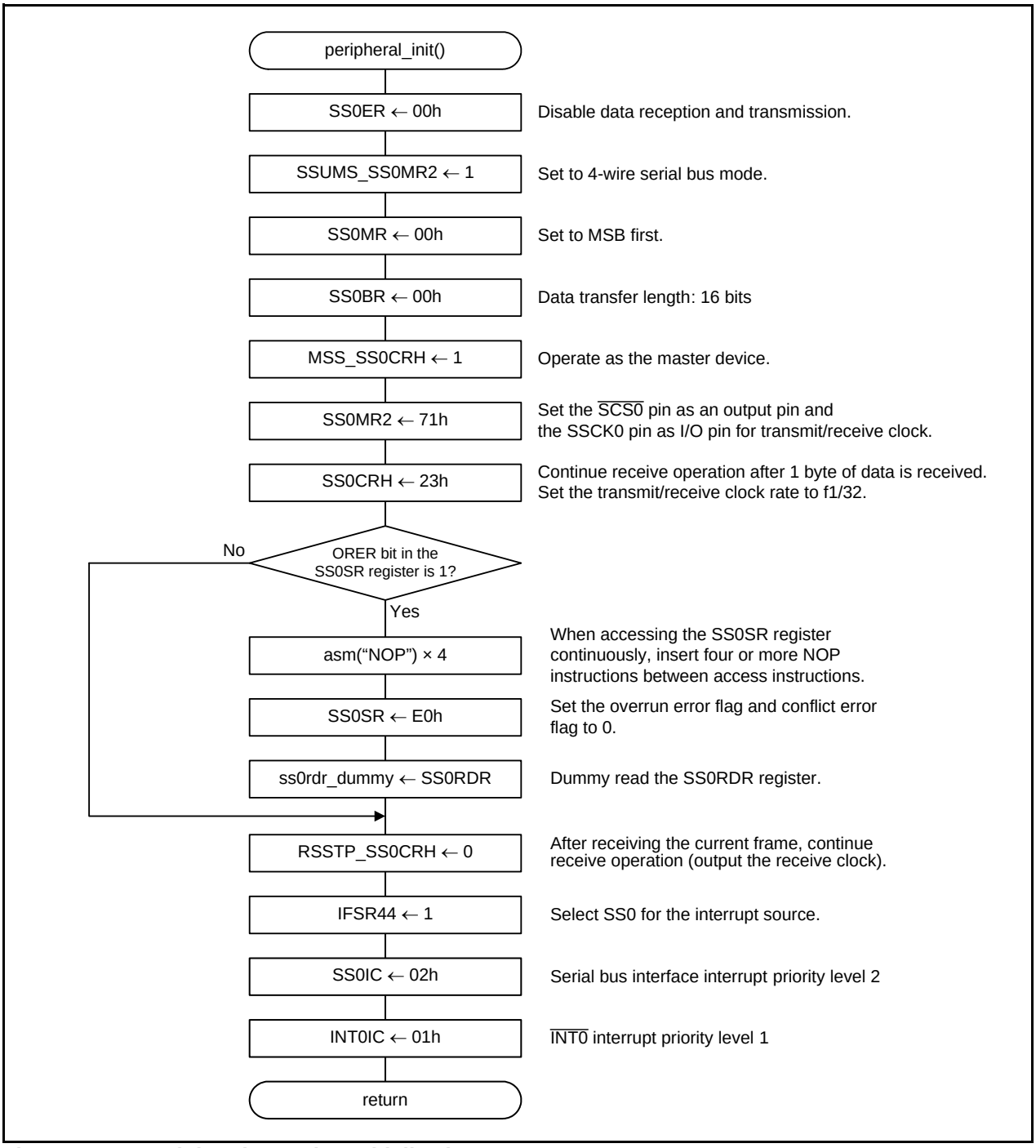
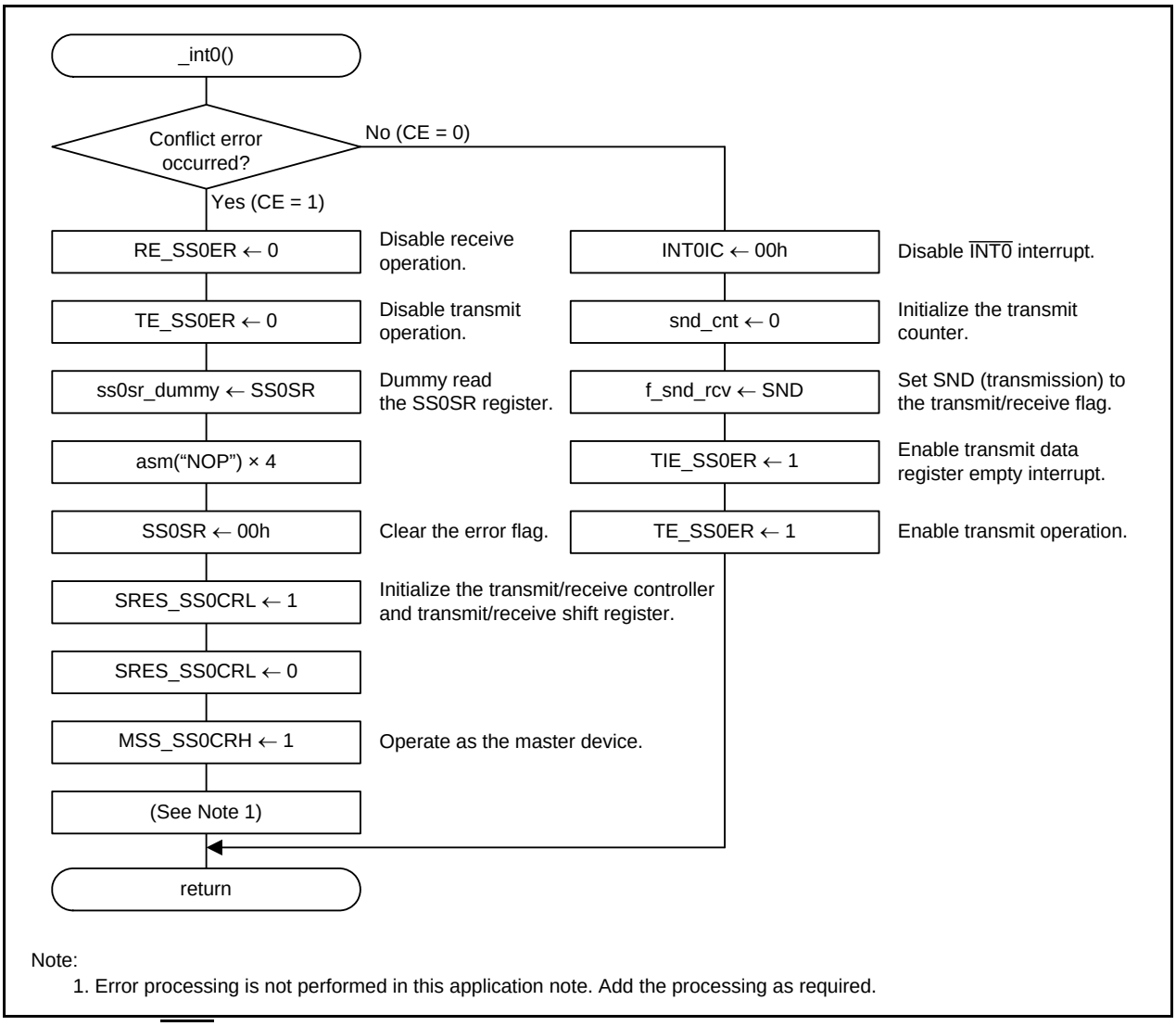


Figure 5.2 Peripheral Function Initialization

5.1.3 $\overline{\text{INT0}}$ Interrupt Handler



Note:
1. Error processing is not performed in this application note. Add the processing as required.

Figure 5.3 $\overline{\text{INT0}}$ Interrupt Handler

5.1.4 Serial Bus Interface Interrupt Handler

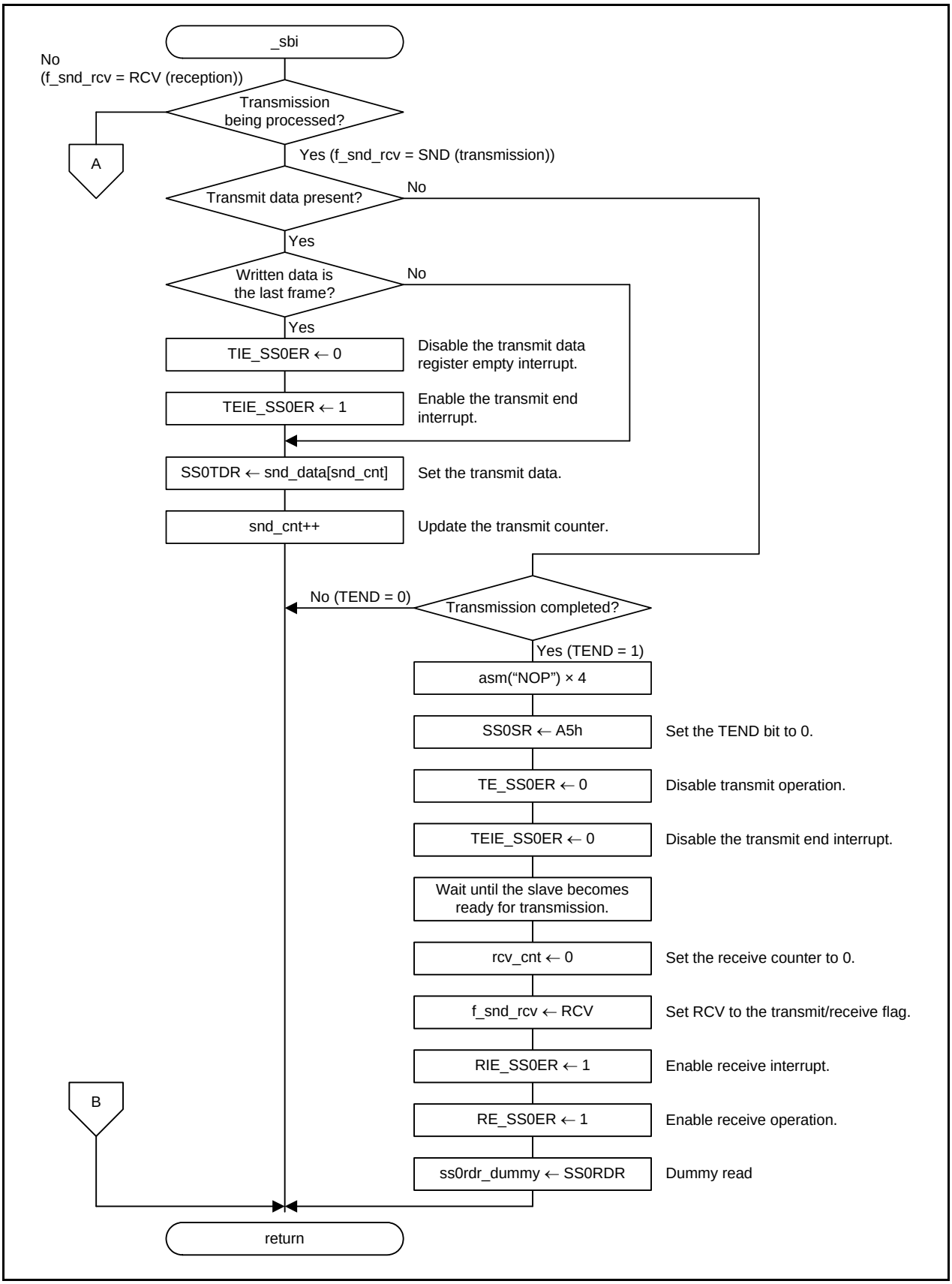
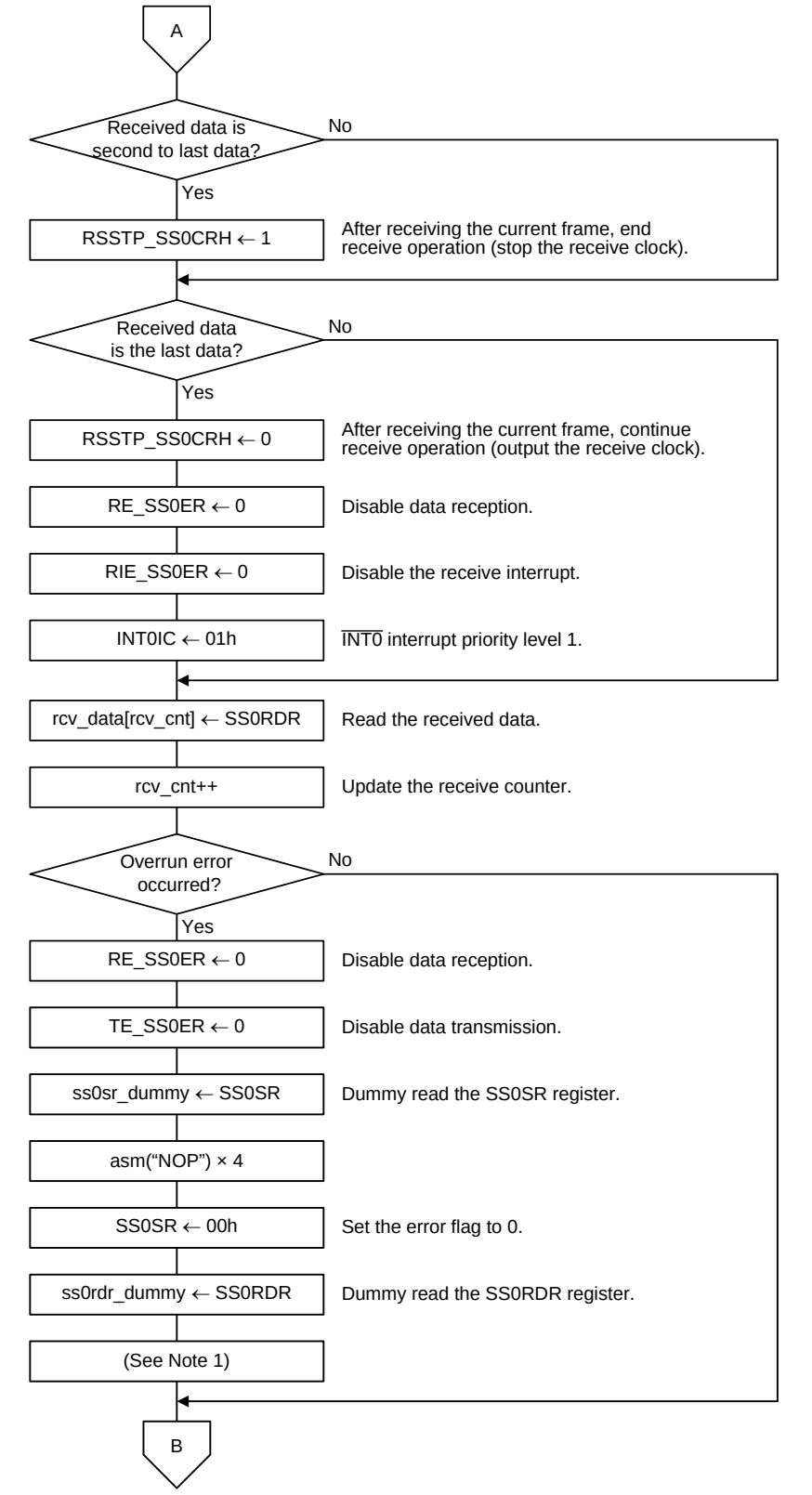


Figure 5.4 Serial Bus Interface Interrupt Handler (1/2)



Note:
1. Error processing is not performed in this application note. Add the processing as required.

Figure 5.5 Serial Bus Interface Interrupt Handler (2/2)

5.2 Slave Transmit/Receive Program

5.2.1 Main Function

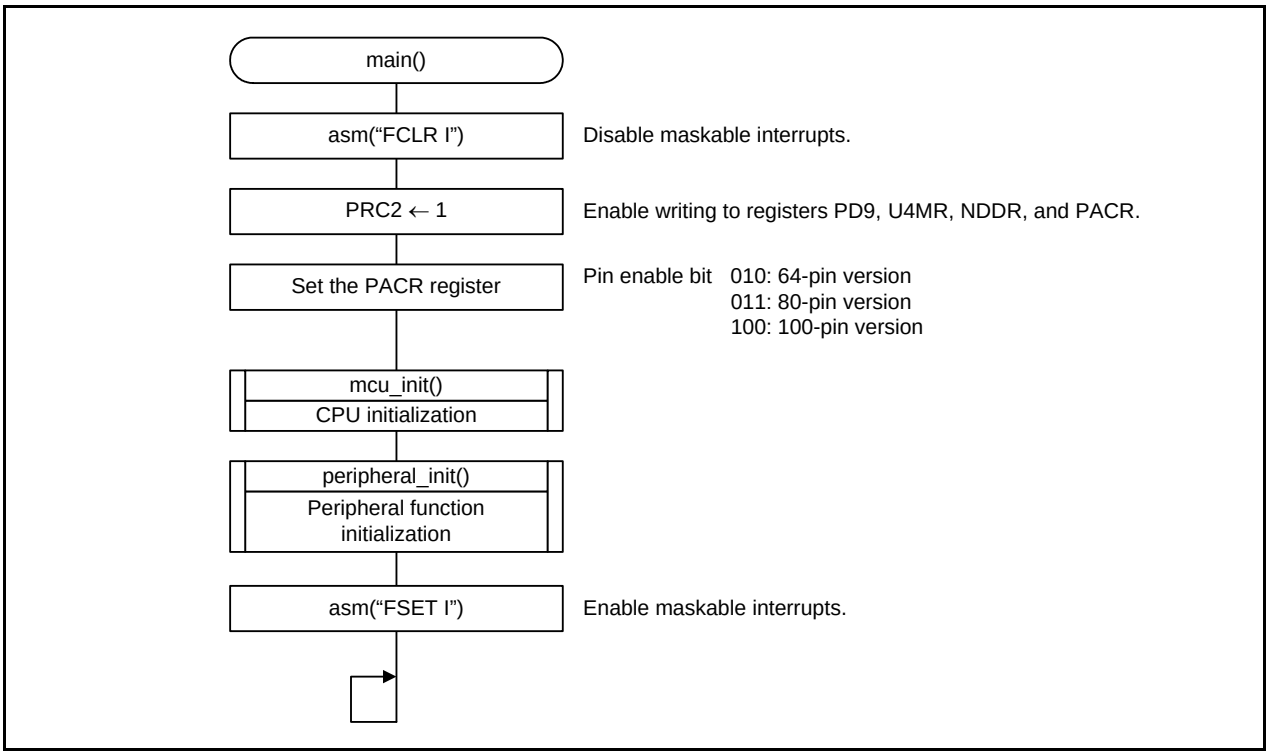


Figure 5.6 Main Function

5.2.2 Peripheral Function Initialization

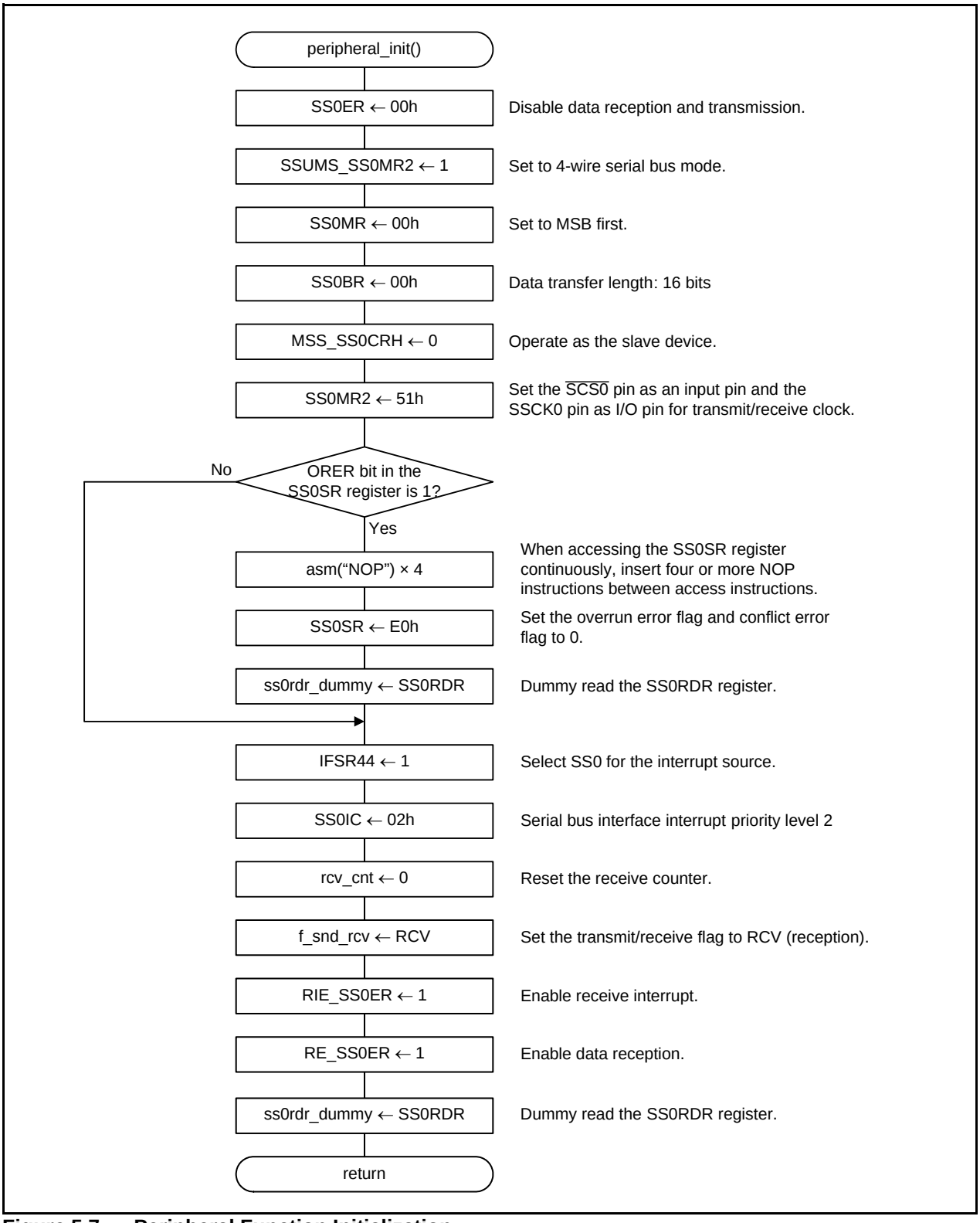


Figure 5.7 Peripheral Function Initialization

5.2.3 Serial Bus Interface Interrupt Handler

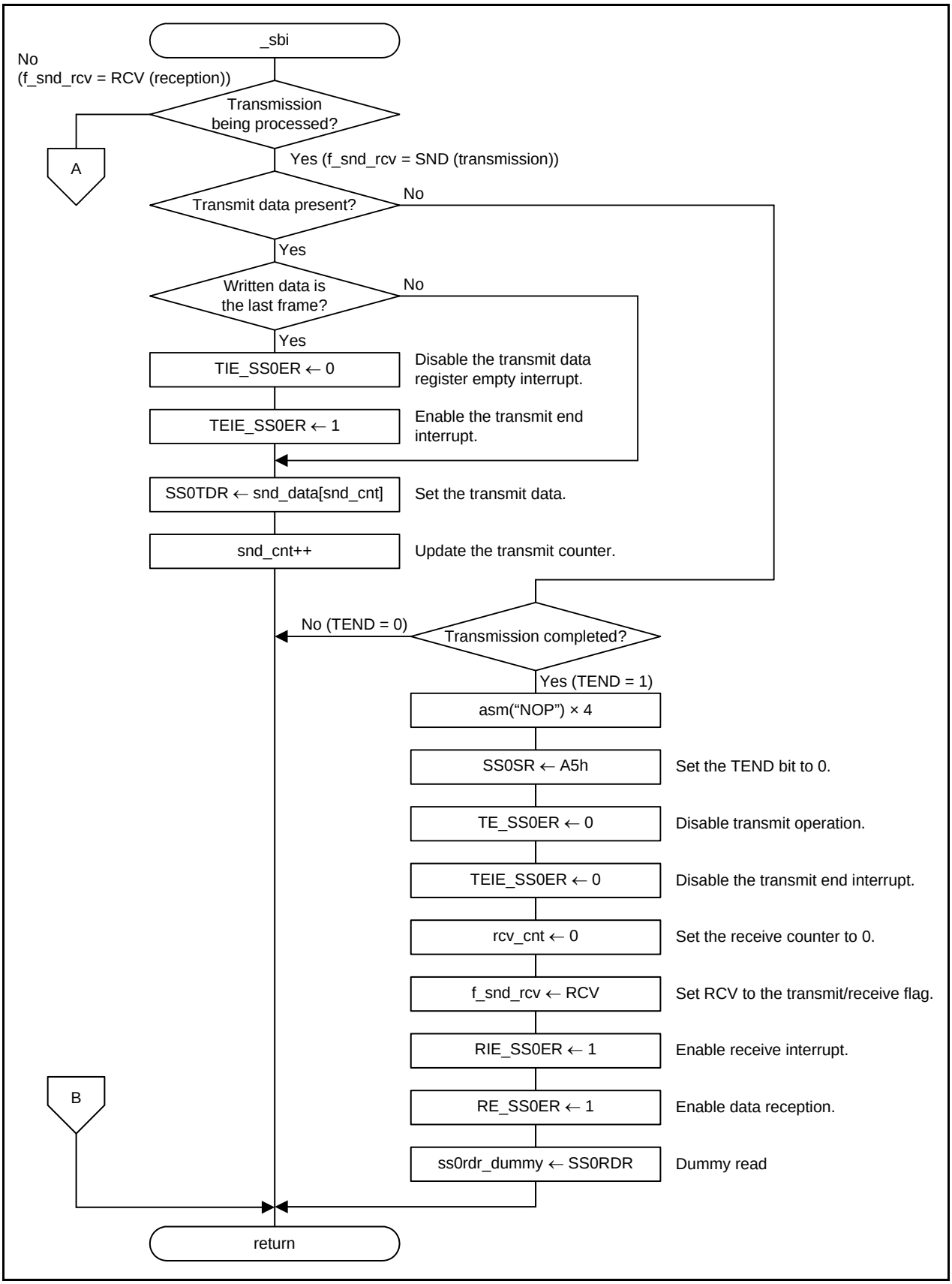
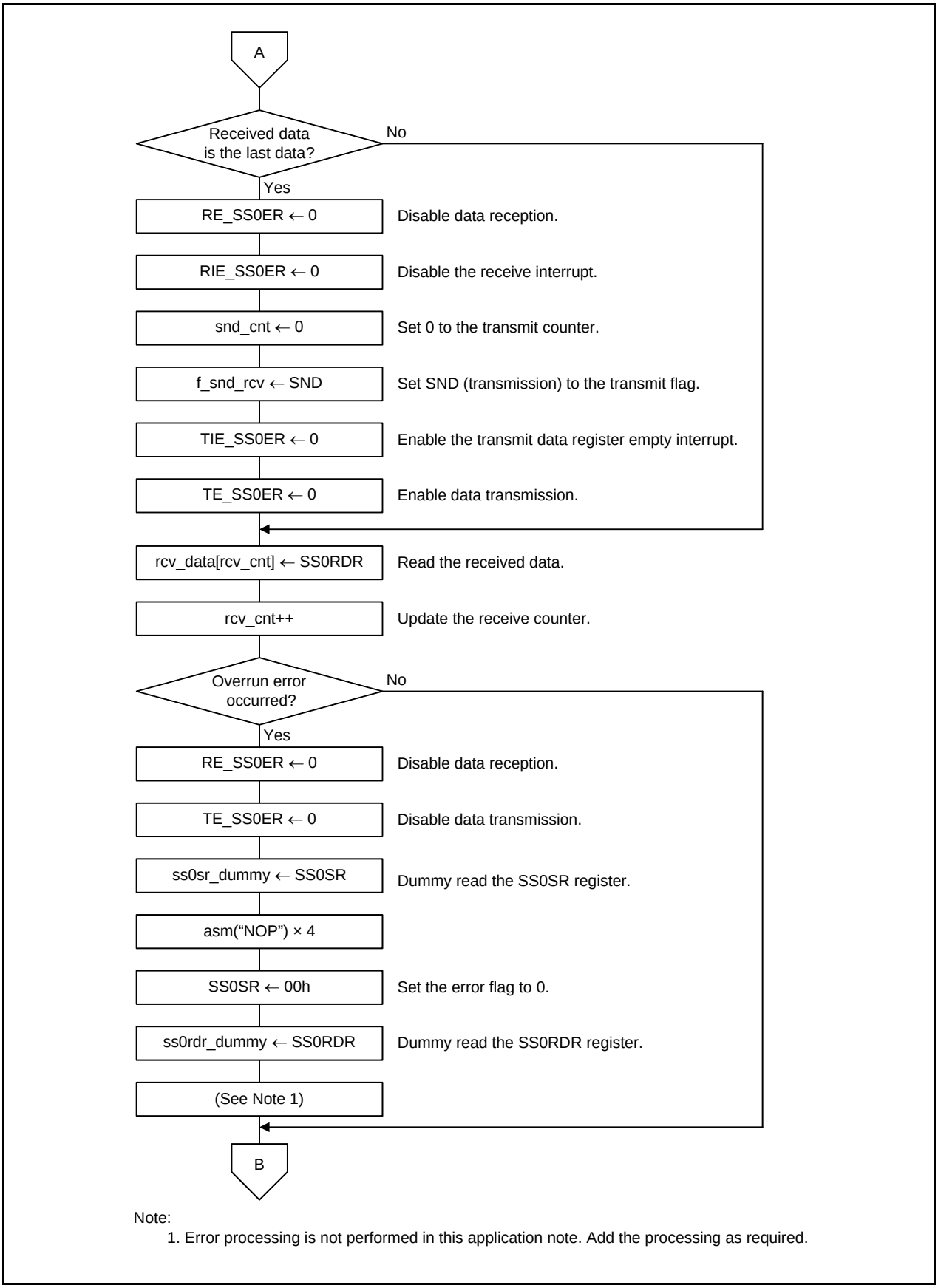


Figure 5.8 Serial Bus Interface Interrupt Handler (1/2)



Note:
1. Error processing is not performed in this application note. Add the processing as required.

Figure 5.9 Serial Bus Interface Interrupt Handler (2/2)

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Document

M16C/5M Group, M16C/57 Group User's Manual: Hardware Rev.1.10
The latest version 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 User's Manual
M16C Series, R8C Family C Compiler Package V.5.45
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/contact/

Revision History	M16C/5M and M16C/57 Groups Transmission/Reception Using the Serial Bus Interface in Standard Mode of 4-Wire Serial Bus Mode

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 30, 2012	—	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