RX62N Group, RX621 Group

RIIC Multi-Master Communication

R01AN0630EJ0100
Rev.1.00
Sep 27, 2011

Introduction

This application note presents an example of communication in multi-master mode using the RIIC (I^2C bus interface) module provided by Renesas microcontrollers.

Target Devices

RX62N Group and RX621 Group devices

The sample program provided in this application note can also be used with other RX Family microcontrollers that have the same I/O registers (peripheral control registers) as the RX62N Group and RX621 Group devices. Note, however, that there are changes such as added functionality in certain sections, so users must check the corresponding manuals carefully. Operation must be fully evaluated in advance if the code in this application note is used in an end product.

Contents

1. Specifications

This sample program first performs a master mode transmission operation (10 bytes) and then performs a master mode reception operation (10 bytes). It uses a communication bit rate of 100 kbps.

Table 1 lists peripheral functions used and their application and figure 1 shows the connection diagram.

Table 1 Peripheral Functions and Usage

Peripheral function	Usage
RIIC	For RIIC communication
Interrupt controller (ICU)	For interrupts

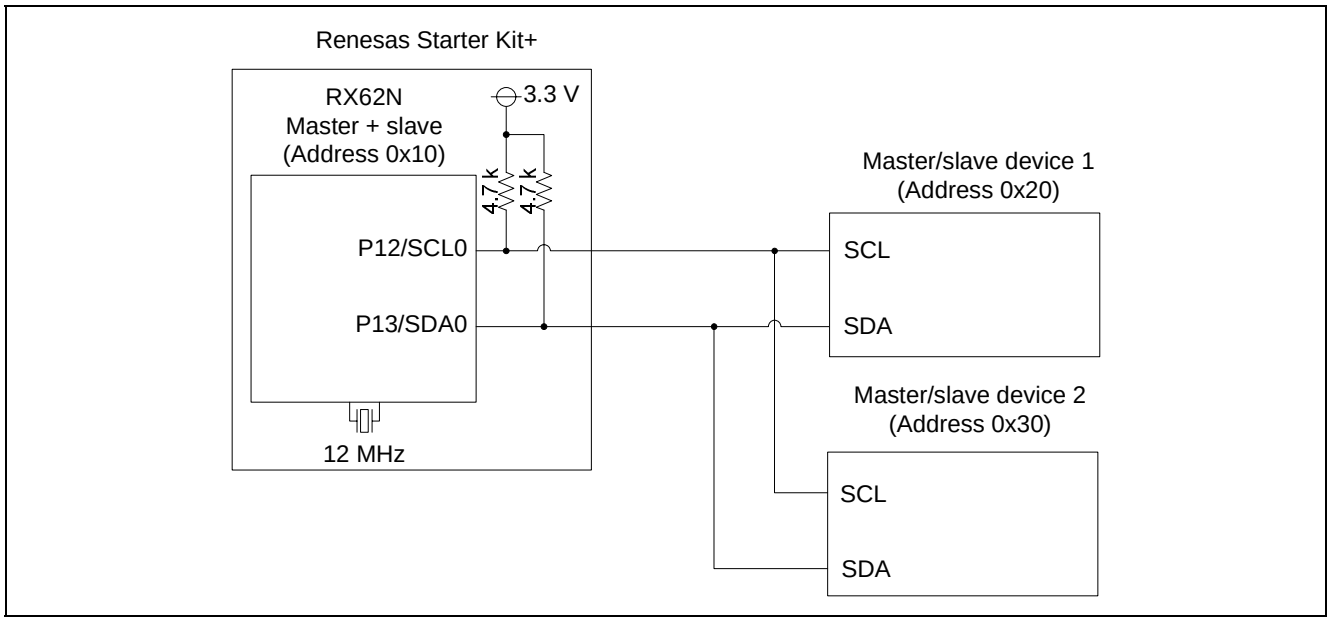

Figure 1 Connection Diagram

2. Operation Verification Environment

Operation of the sample code provided in this application note has been verified in the following environment.

Table 2 Operation Verification Environment

Item	Description
Microcontroller used	RX62N (R5F562NBDBG)
Operating frequency	12 MHz (ICLK = 96 MHz, PCLK = 48 MHz)
Operating voltage	5.0 V (CPU operating voltage of 3.3 V)
Integrated development environment	Version 4.08.00.011
Compiler	RX Standard Toolchain (V.1.0.1.0)
Operating mode	Single-chip mode
Sample code version	1.00
Board used	The RSK + RX62N (R0K5562N0C00BE) provided with the Renesas Development Tools (Catalog number: R0K5562N0S000BE)

3. Software Description

3.1 Operation Overview

- The program performs master mode transmission. If the arbitration lost state is detected during master mode transmission, the communication of the other master device is given priority and the program switches to slave mode operation.
- The program performs master mode reception. If the arbitration lost state is detected during master mode reception, the communication of the other master device is given priority and the program switches to slave mode operation.
- If the slave address matches during slave mode operation, a slave reception or transmission operation is started according to the state of the R/W# bit.
- A callback function is called when slave mode communication completes or when master mode communication completes.

Table 3 lists the settings for the RIIC module used in this application note.

Table 3 RIIC Settings

Item	Description
Channel	Channel 0
Master/slave operation	Master transmission, master reception, slave reception, slave transmission
Address format	7-bit address format
Slave address	0x10
Transmission speed	100 kbps
Arbitration lost detection	Master arbitration lost detection
Timeout detection	• Count when the SCL line is low or high • Long mode (16-bit counter)

Note: Refer to the RX62N Group and RX621 Group Hardware Manual and the I²C bus specifications for details on the I^2C bus communication format.

3.2 File Structure

Table 4 lists the files used for the sample code. The files automatically generated by the integrated development environment are not listed.

Table 4 File Structure

File Name	Function	Notes
main.c	Main processing	
riic.c	RIIC control related processing	
riic_int.c	RIIC interrupt processing	
riic.h	RIIC related header files	
intprg.c	File automatically generated by HEW (Only the RIIC interrupt functions used in this program are deleted.)	

3.3 Constants

Table 5 lists the constants used in the sample code.

Table 5 Constants

Constant	Setting Value	Description
RIIC_OK	0	Used as the return value of the RiicStart() function.
RIIC_NG	1	Used as the return value of the RiicStart() function.
RIIC_BUS_BUSY	2	Used as the return value of the RiicStart() function.
RIIC_ST_MST_IDLE	0	Used as the return value of the RiicGetMstState() function.
RIIC_ST_MST_BUSY	1	Used as the return value of the RiicGetMstState() function.
RIIC_ST_MST_NACK	2	Used as the return value of the RiicGetMstState() function.
RIIC_ST_MST_AL	3	Used as the return value of the RiicGetMstState() function.
RIIC_ST_MST_TMO	4	Used as the return value of the RiicGetMstState() function.
RIIC_ST_MST_COMPLETE	5	Used as the return value of the RiicGetMstState() function.
RIIC_SET	1	Used as a flag setting value.
RIIC_CLEAR	0	Used as a flag setting value.
SELF_ADDRESS	0x10	Self address
MST_DATA_NUM	(10+1)	Used as the master send/receive count.
SLV_DATA_NUM	10	Used as the slave send/receive count.
MST_WRITE	0x00	Used as the argument to RiicMstStart().
MST_READ	0x01	Used as the argument to RiicMstStart().

3.4 Variables

Table 7 lists the global variables.

Table 7 Global Variables

Type	Variable Name	Description	Functions Used By
uint8_t	MstTrmBuff[256]	Master mode transmit buffer	main
uint8_t	MstRcvBuff[256]	Master mode receive buffer	main
uint8_t	SlvTrmBuff[256]	Slave mode transmit buffer	main
uint8_t	SlvRcvBuff[256]	Slave mode receive buffer	main

3.5 Functions

Table 8 lists the functions.

Table 8 Functions

Function Name	Description
main	Main processing
CbSlaveTrm	Callback function (when slave mode transmission completes)
CbSlaveRcv	Callback function (when slave mode reception completes)
CbMaster	Callback function (when master mode transmission or reception completes)
RiicIni	User interface function. RIIC initialization. Communication enable/disable setting.
RiicMstStart	User interface function. Master mode communication start (master mode transmit or receive)
RiicGetMstState	User interface function. Master mode communication result acquisition
RiicSlvStart	User interface function. Slave mode operation start (slave mode transmit or receive)
RiicTDRE	Transmit data empty interrupt handler
RiicTEND	Transmit complete interrupt handler
RiicRDRF	Receive data full interrupt handler
RiicSTOP	Stop condition detection interrupt handler
RiicNACK	NACK detection interrupt handler
RiicAL	Arbitration lost detection interrupt handler
RiicTMO	Timeout detection interrupt handler

3.6 Function Specifications

The specifications of the functions used in the sample code are listed below.

main	
Overview	Main processing
Header	None
Declaration	void main(void)
Description	• RIIC initialization • Slave operation start • Master transmission start • Master reception start
Arguments	None
Return values	None
Notes	

CbSlaveTrm	
Overview	Callback function (when slave transmission completes)
Header	None
Declaration	void CbSlaveTrm(void)
Description	Called after slave transmission completes.
Arguments	None
Return values	None
Notes	

CbSlaveRcv	
Overview	Callback function (when slave reception completes)
Header	None
Declaration	void CbSlaveRcv(void)
Description	Called after slave reception completes.
Arguments	None
Return values	None
Notes	

CbMaster	
Overview	Callback function (when master transmission or reception completes)
Header	None
Declaration	void CbMaster(void)
Description	Called after master transmission or reception completes.
Arguments	None
Return values	None
Notes	

RiicIni	
Overview	RIIC initialization
Header	riic.h
Declaration	void RiicIni(uint8_t, uint8_t)
Description	Initializes the RIIC module.
Arguments	First argument: uint8_t in_SelfAddr The address of this device itself (The low-order bit must be set to 0.) Second argument: uint8_t in_Enable 0: RIIC communication disabled Any other value: RIIC communication enabled
Return values	None
Notes	

RiicSlvStart	
Overview	Starts slave mode operation
Header	riic.h
Declaration	void RiicSlvStart(uint8_t * , uint8_t * , uint32_t , CallBackFunc, CallBackFunc)
Description	Starts slave mode operation. If the slave address matches during slave mode operation, slave transmission or slave reception is started according to the R/W# bit. • Slave Transmission — After the start of slave mode transmission, this function transmits data from the start of the transmit buffer specified with the second argument until a NACK is received. — Transmits 0xFF if the transmit count exceeds the transmit data count specified in the third argument. — After transmission completes, it calls the callback function specified in the fourth argument. However, if a NACK is received while transmitting data that is less than the transmit data count specified in the third argument, it does not call the call back function but rather transmits from the start of the transmit buffer specified in the second argument when slave mode transmission is started again. • Slave Reception — After the start of slave mode reception, this function receives data from the start of the receive buffer specified with the second argument until a stop condition is detected. — Receive data in excess of the receive data count specified in the third argument is not stored in the receive buffer specified in the first argument. — After reception completes, it calls the callback function specified in the fifth argument. However, if a stop condition is detected while receiving data that is less than the receive data count specified in the third argument it does not call the call back function but rather receives data from the start of the transmit buffer specified in the second argument when slave mode reception is started again (that is, it overwrites the buffer).
Arguments	First argument: uint8_t* in_RcvAddr Slave mode reception data storage pointer Data is stored in the buffer pointed to by this argument. Second argument: uint8_t* in_TrmAddr Slave mode transmission data storage pointer Data is transmitted from the buffer pointed to by this argument. Third argument: uint32_t in_num Transmit/receive data count Specifies the number of data items for reception or transmission. Fourth argument: CallBackFunc cbTrm Slave mode transmission complete callback function This function is called when slave mode transmission completes. Fifth argument: CallBackFunc cbRcv Slave mode reception complete callback function This function is called when slave mode reception completes.
Return values	None
Notes	When slave mode transmission or reception completes, slave mode operation stops. Applications should call this function again to continue slave mode operation.

RiicStart	
Overview	Starts master mode transmission
Header	riic.h
Declaration	uint8_t RiicMstStart(uint8_t, uint8_t *, uint32_t, CallBackFunc)
Description	Starts master mode transmission. After master mode transmission is started, this function calls the callback function under the following conditions. • NACK received • Arbitration lost detected • Timeout detected • Master mode transmission or reception completes The results of communication listed above can be acquired with the RiicGetMstState() function.
Arguments	First argument: uint8_t* in_addr Slave address (the low-order bit is the R/W bit). When the low-order bit is 0, master mode transmission is performed, and when 1, master mode reception is performed. Second argument: uint8_t* in_buff Pointer to the data storage area used for communication. For master mode transmission, data is transmitted from the buffer pointed to by this argument. For master mode reception, data is stored in the buffer pointed to by this argument. Third argument: uint32_t in_num Transmit/receive data count Specifies the number of data items for reception or transmission. The transmission address is also included. Fourth argument: CallBackFunc cb The callback function.
Return values	RIIC_OK Normal completion RIIC_NG Argument error (when the transmit/receive data count is less than 2) RIIC_BUS_BUSY Bus busy
Notes	

RiicGetMstState		
Overview	Acquire master mode communication result	
Header	riic.h	
Declaration	uint8_t RiicGetMstState(void)	
Description	Returns the result of master mode communication.	
Arguments	None	
Return values	RIIC_ST_MST_IDLE	Before the start of communication
	RIIC_ST_MST_BUSY	Communication in progress
	RIIC_ST_MST_NACK	NACK received
	RIIC_ST_MST_AL	Arbitration lost detected
	RIIC_ST_MST_TMO	Timeout detected
	RIIC_ST_MST_COMPLETE	Master mode transmission or reception completed
Notes		

3.7 Flowcharts

3.7.1 Main Processing

Figure 2 shows the flowchart for the main function.

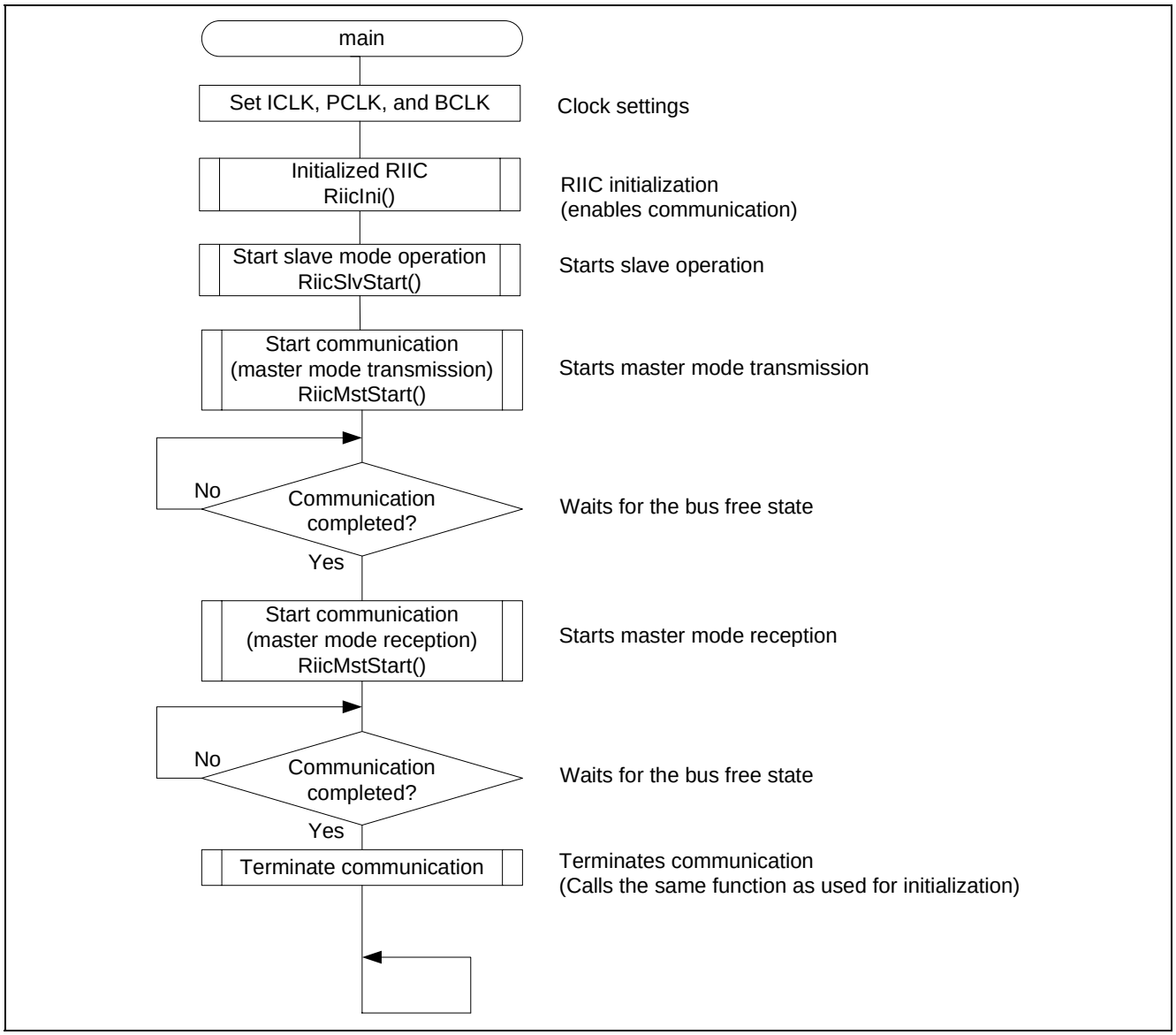

Figure 2 Main Processing

3.7.2 Callback Function (Slave Mode Transmission Complete)

Figure 3 shows the flowchart for the callback function (slave mode transmission complete).

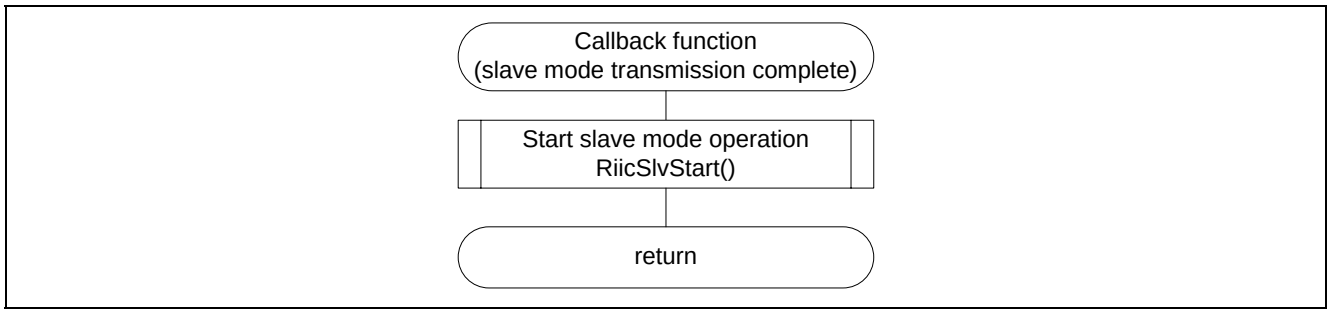

Figure 3 Callback Function (Slave Mode Transmission Complete)

3.7.3 Callback Function (Slave Mode Reception Complete)

Figure 4 shows the flowchart for the callback function (slave mode reception complete).

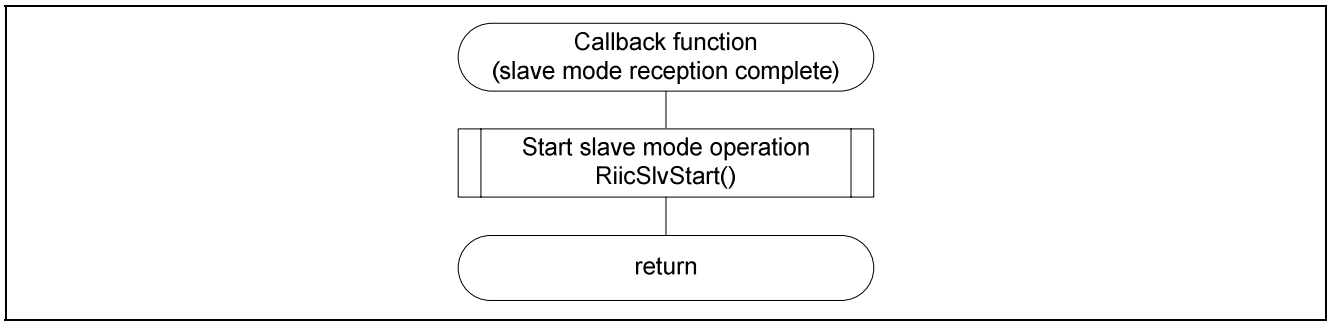

Figure 4 Callback Function (Slave Mode Reception Complete)

3.7.4 Callback Function (Master Mode Transmission or Reception Completes)

Figure 5 shows the flowchart for the callback function (master mode transmission or reception completes).

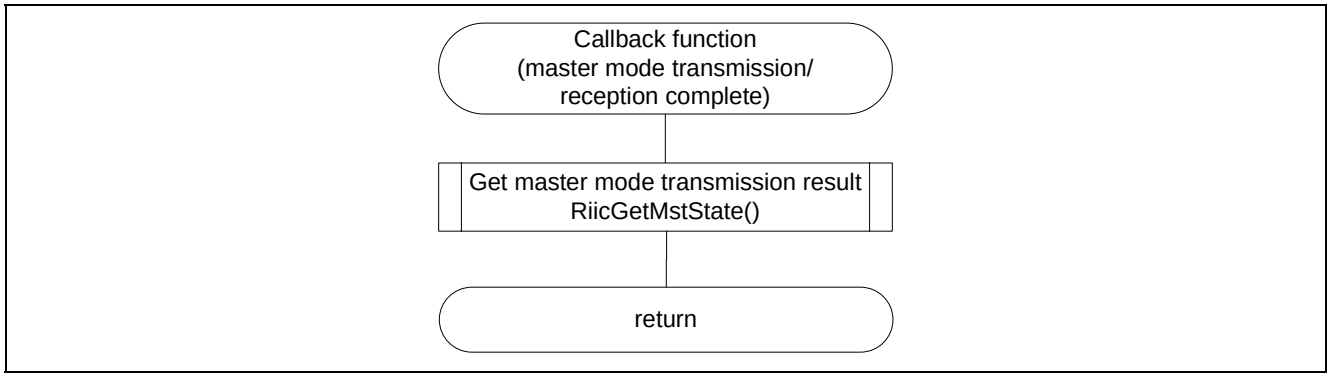

Figure 5 Callback Function (Master Mode Transmission or Reception Completes)

3.7.5 RIIC Initialization

Figure 6 shows the flowchart for RIIC initialization.

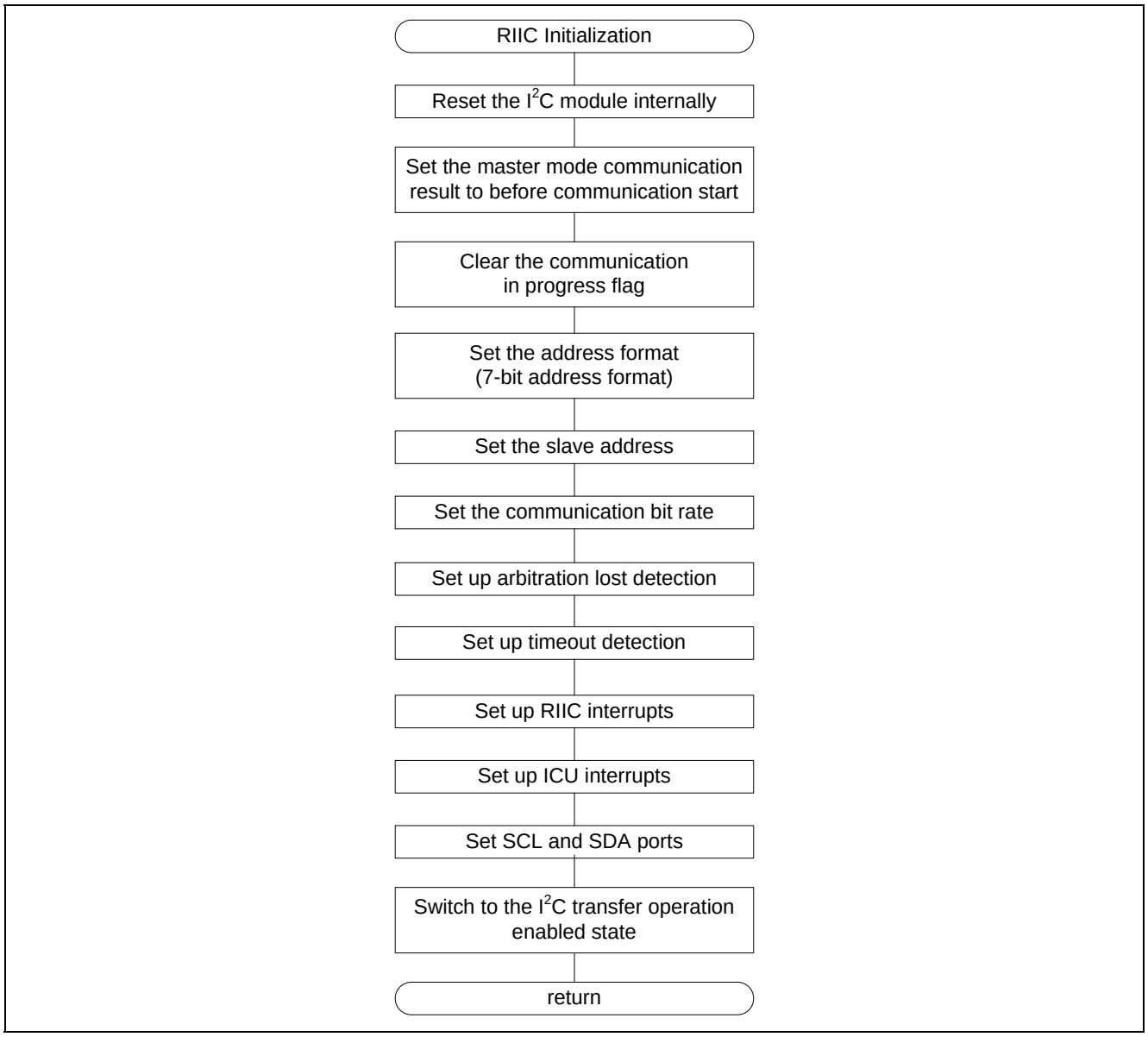

Figure 6 RIIC Initialization

3.7.6 Start Slave Mode Operation

Figure 7 shows the flowchart for starting slave mode operation.

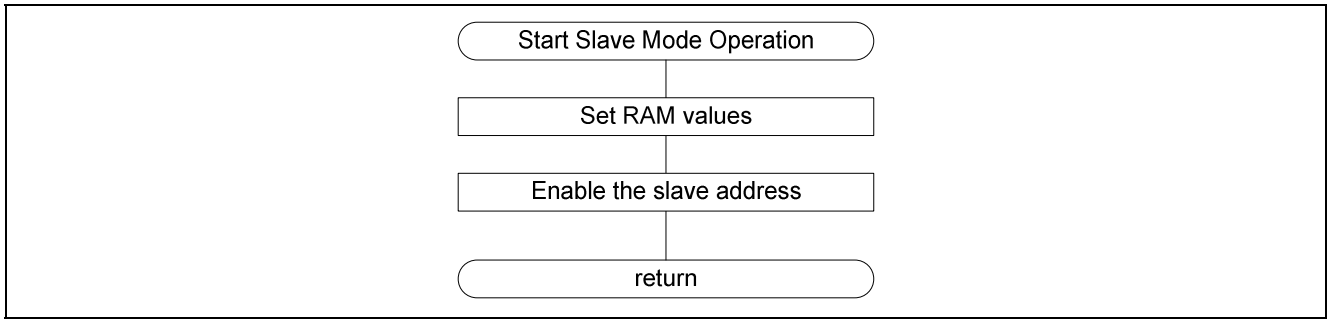

Figure 7 Starting Slave Mode Operation

3.7.7 Start Master Mode Communication

Figure 8 shows the flowchart for starting master mode communication.

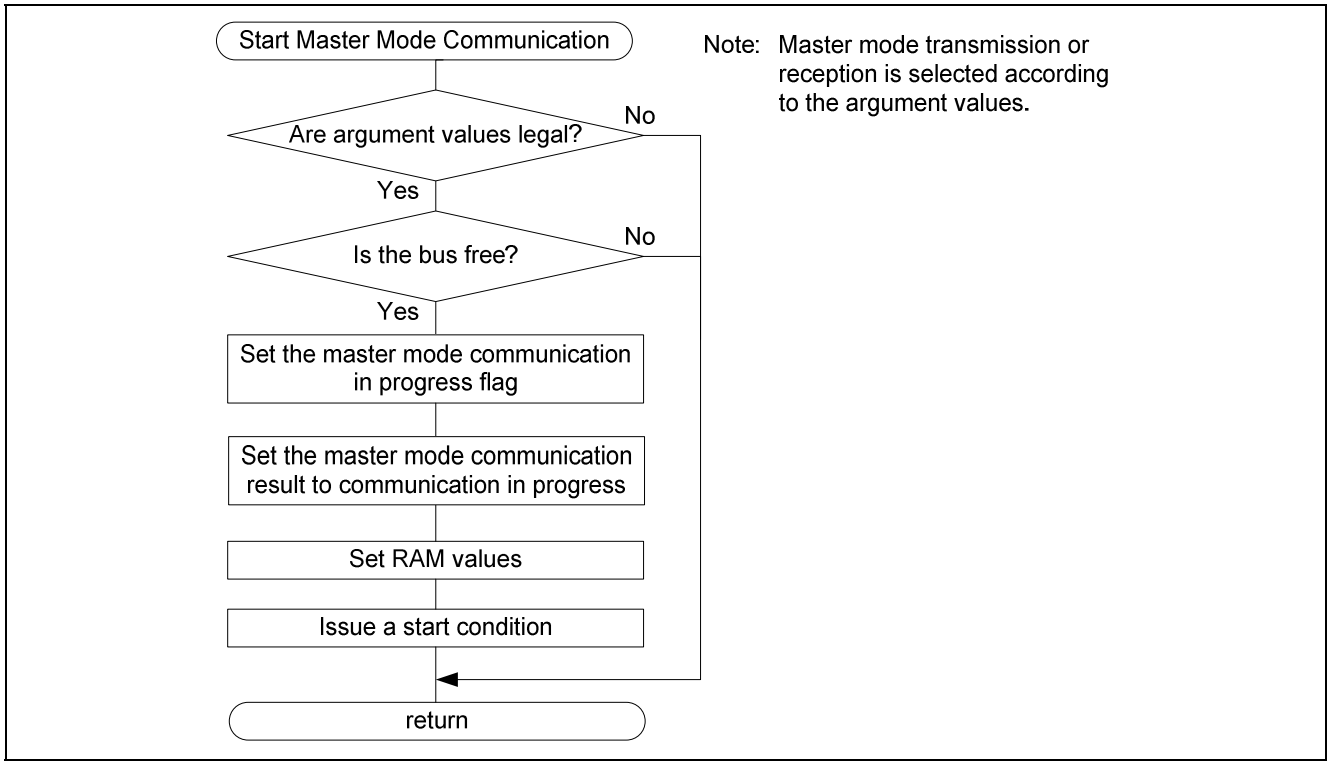

Figure 8 Starting Master Mode Communication

3.7.8 Acquire Master Mode Communication Result

Figure 9 shows the flowchart for acquiring the result of master mode communication.

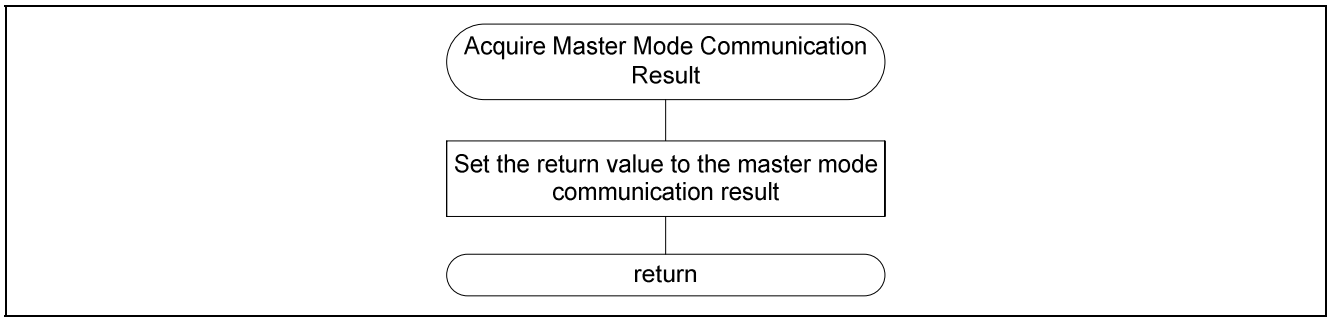

Figure 9 Acquiring the Master Mode Communication Result

3.7.9 RDRF Interrupt Handling

Figure 10 shows the flowchart for RDRF interrupt handling.

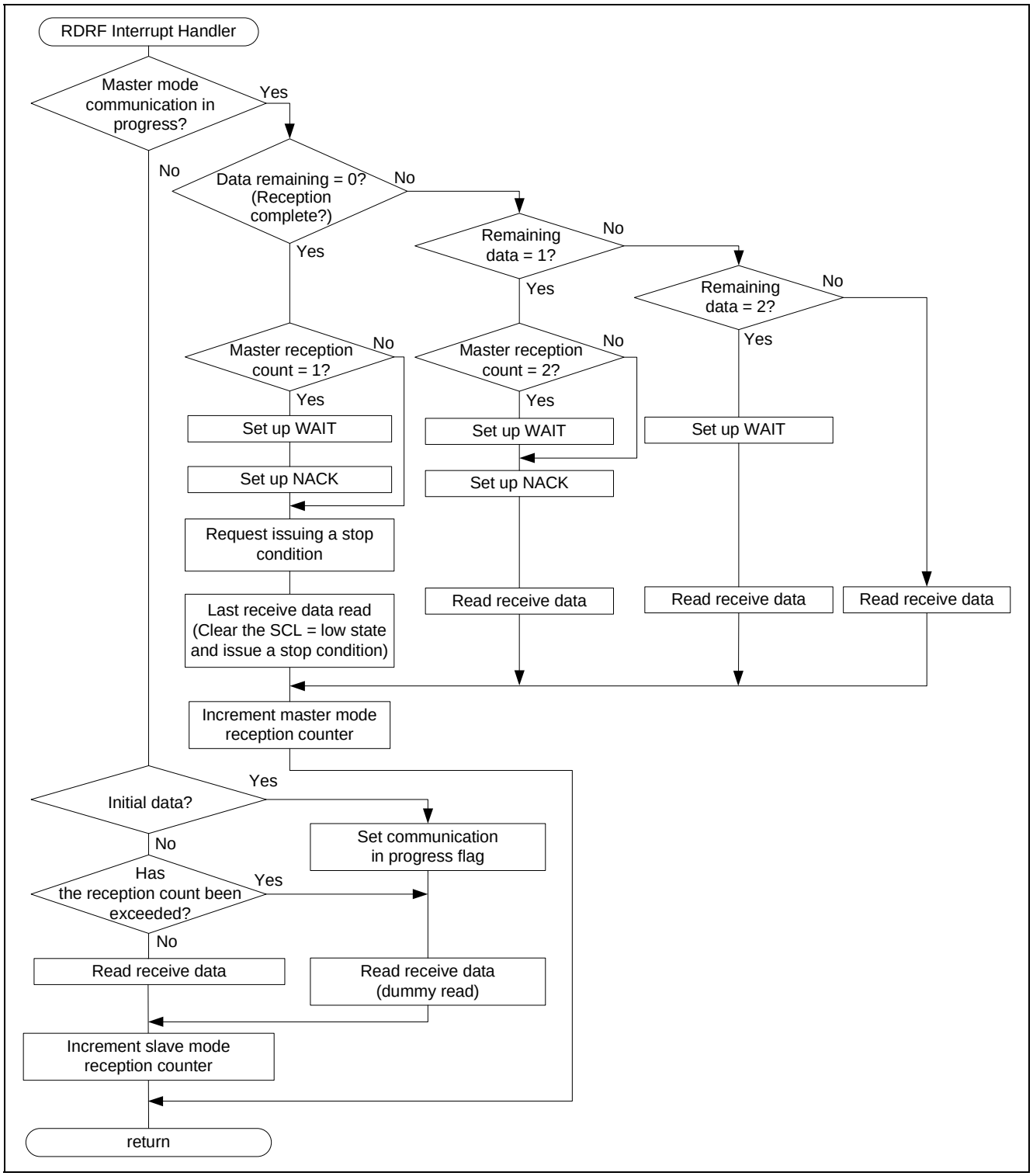

Figure 10 RDRF Interrupt Handling

3.7.10 TDRE Interrupt Handling

Figure 11 shows the flowchart for TDRE interrupt handling.

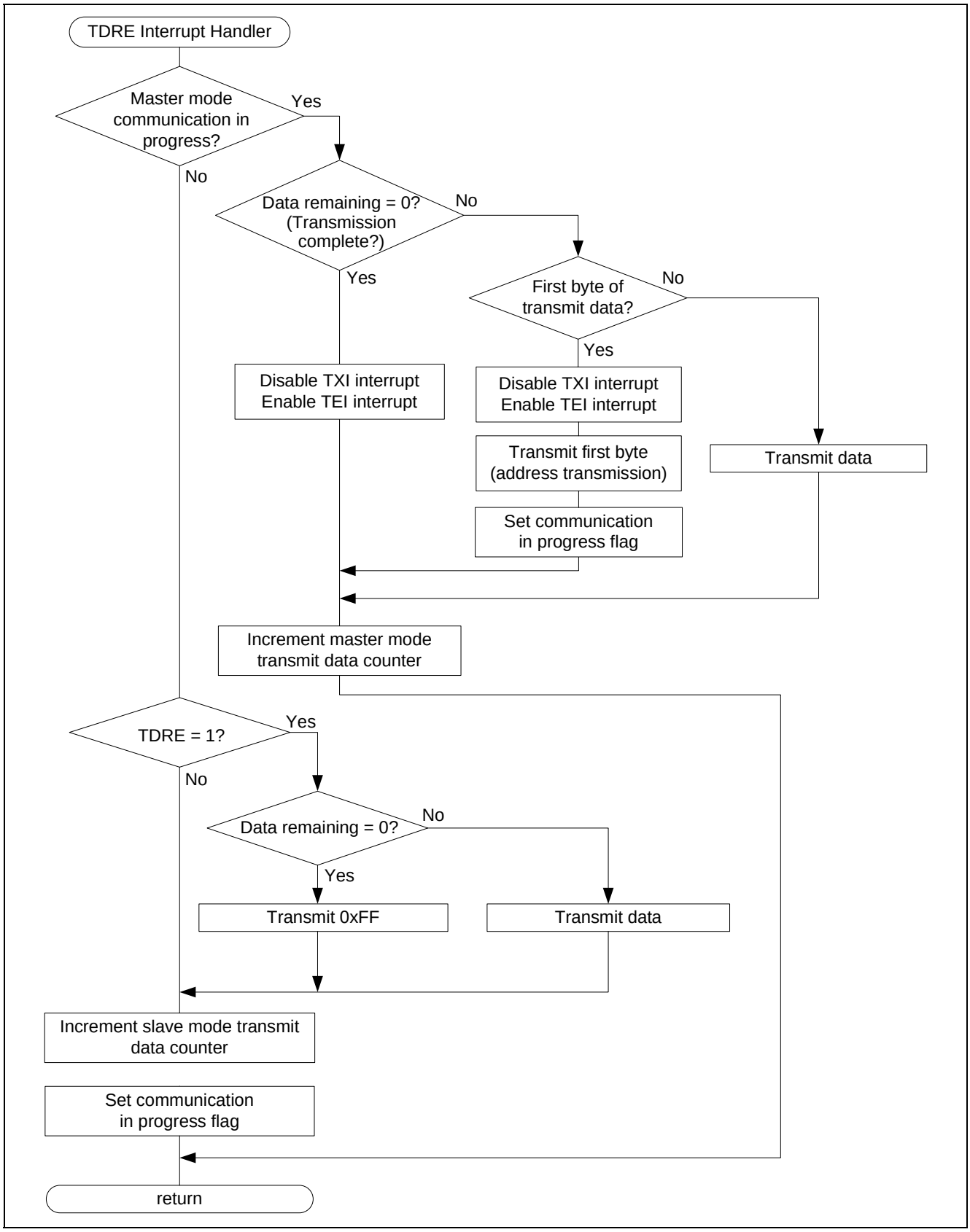

Figure 11 TDRE Interrupt Handling

3.7.11 TEND Interrupt Handling

Figure 12 shows the flowchart for TEND interrupt handling.

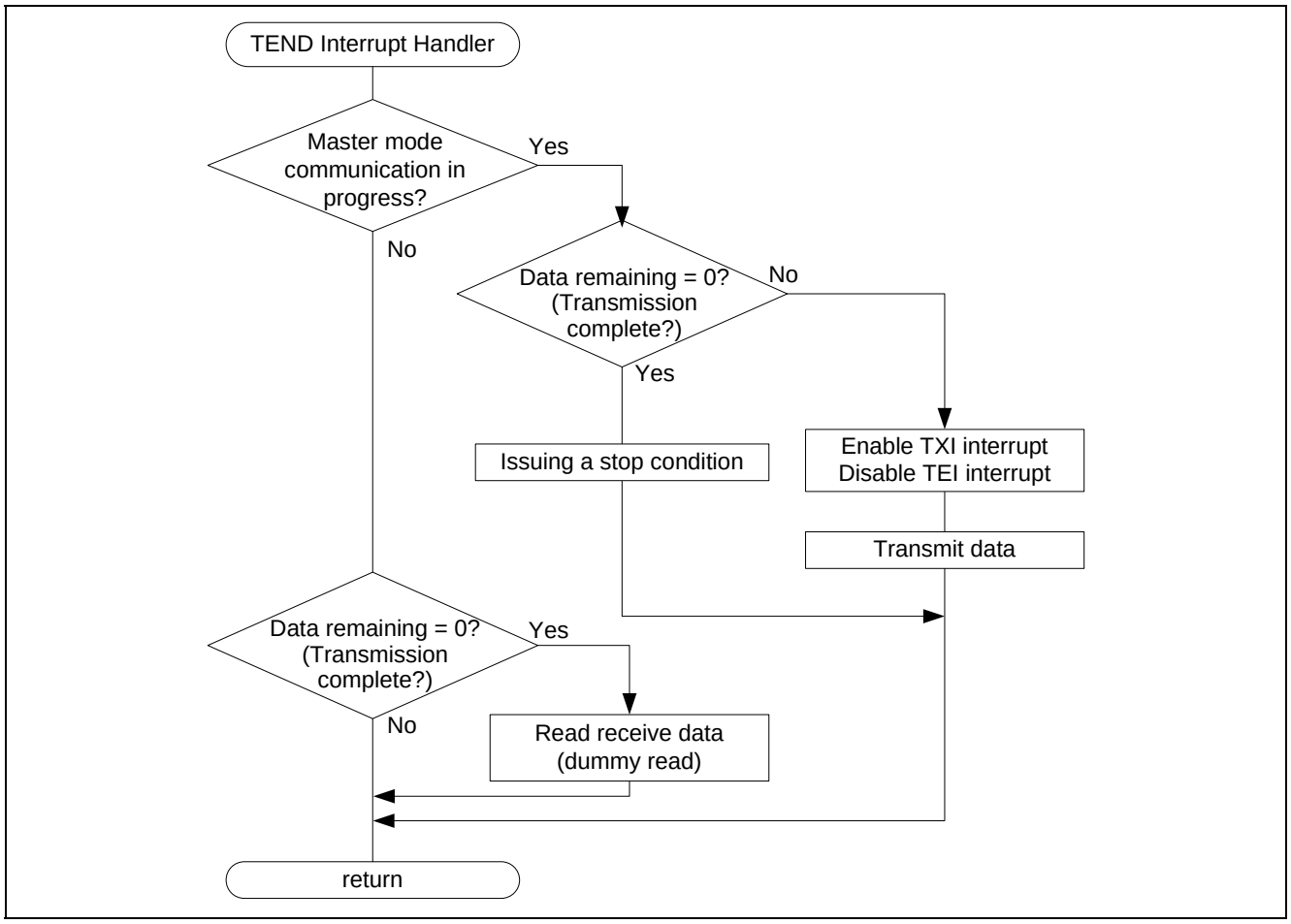

Figure 12 TEND Interrupt Handling

3.7.12 Stop Condition Detection Interrupt Handling

Figure 13 shows the flowchart for stop condition detection interrupt handling.

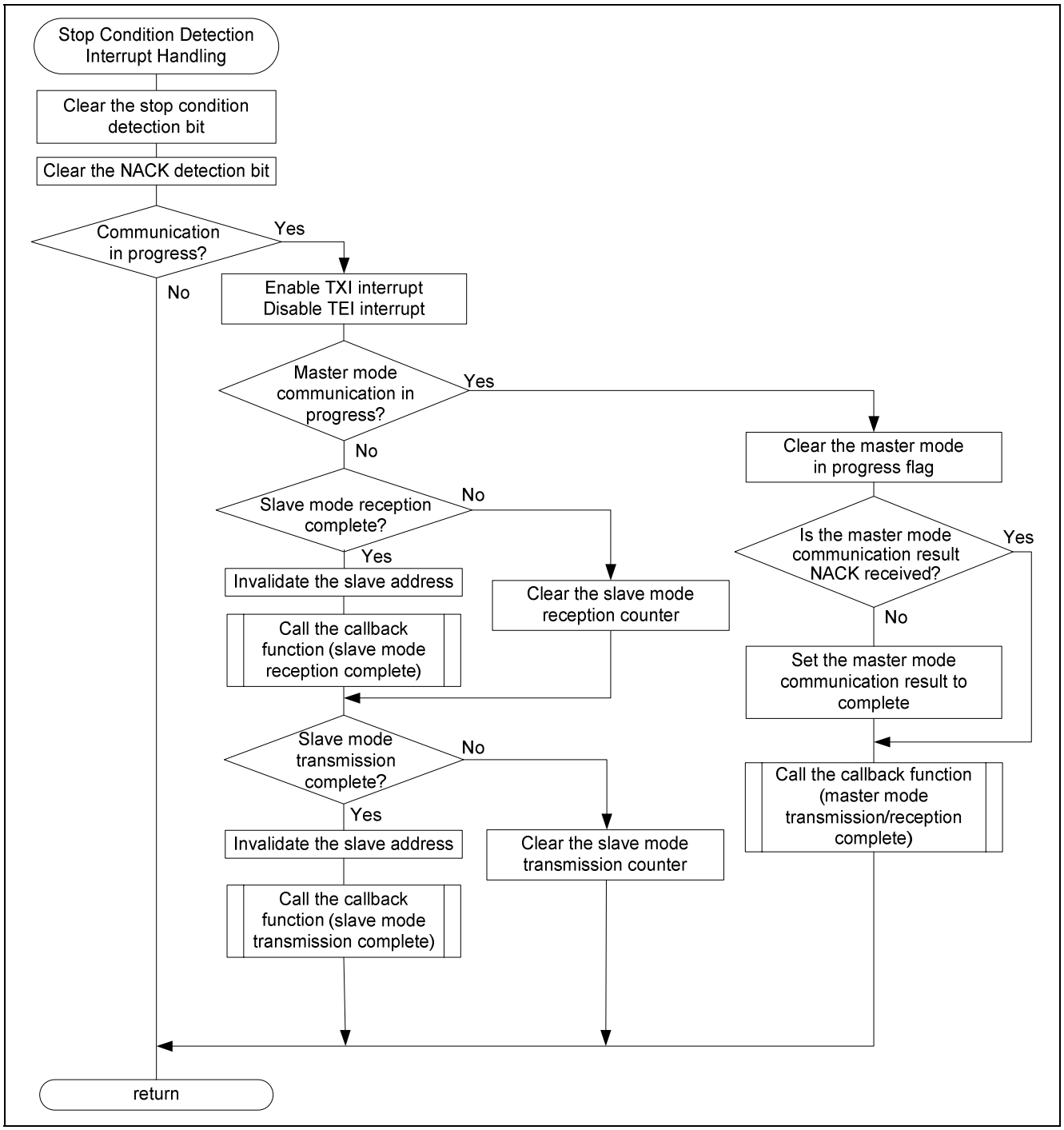

Figure 13 Stop Condition Detection Interrupt Handling

3.7.13 NACK Interrupt Handling

Figure 14 shows the flowchart for NACK interrupt handling.

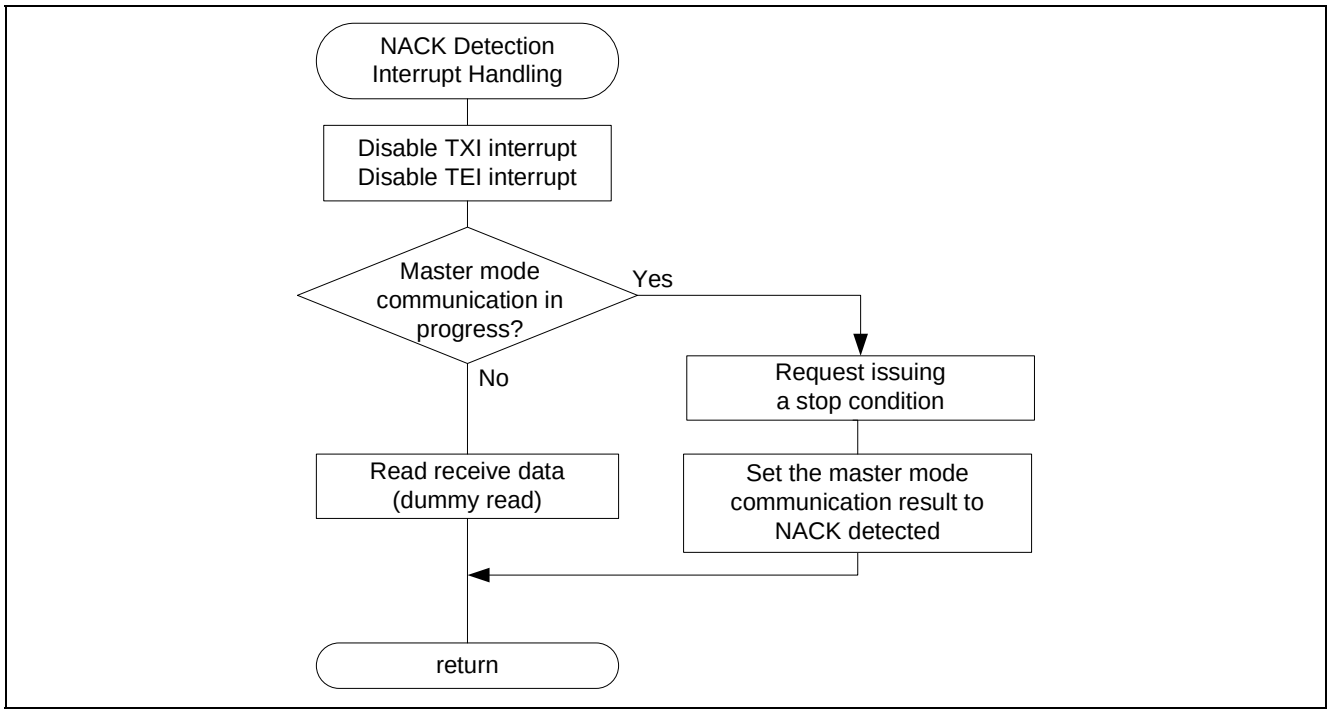

Figure 14 NACK Interrupt Handling

3.7.14 Arbitration Lost Detection Interrupt Handling

Figure 15 shows the flowchart for arbitration lost detection interrupt handling.

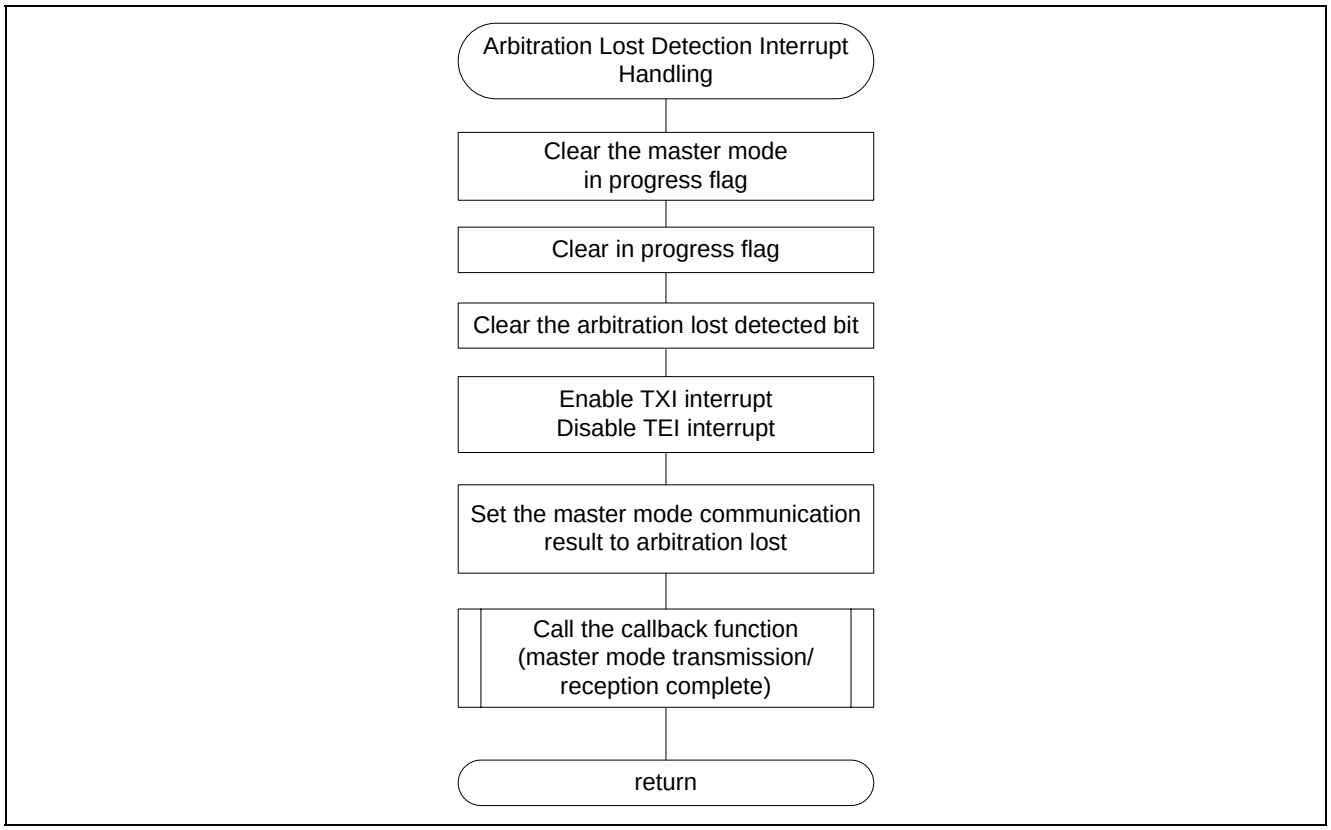

Figure 15 Arbitration Lost Detection Interrupt Handling

3.7.15 Timeout Detection Interrupt Handling

Figure 16 shows the flowchart for timeout detection interrupt handling.

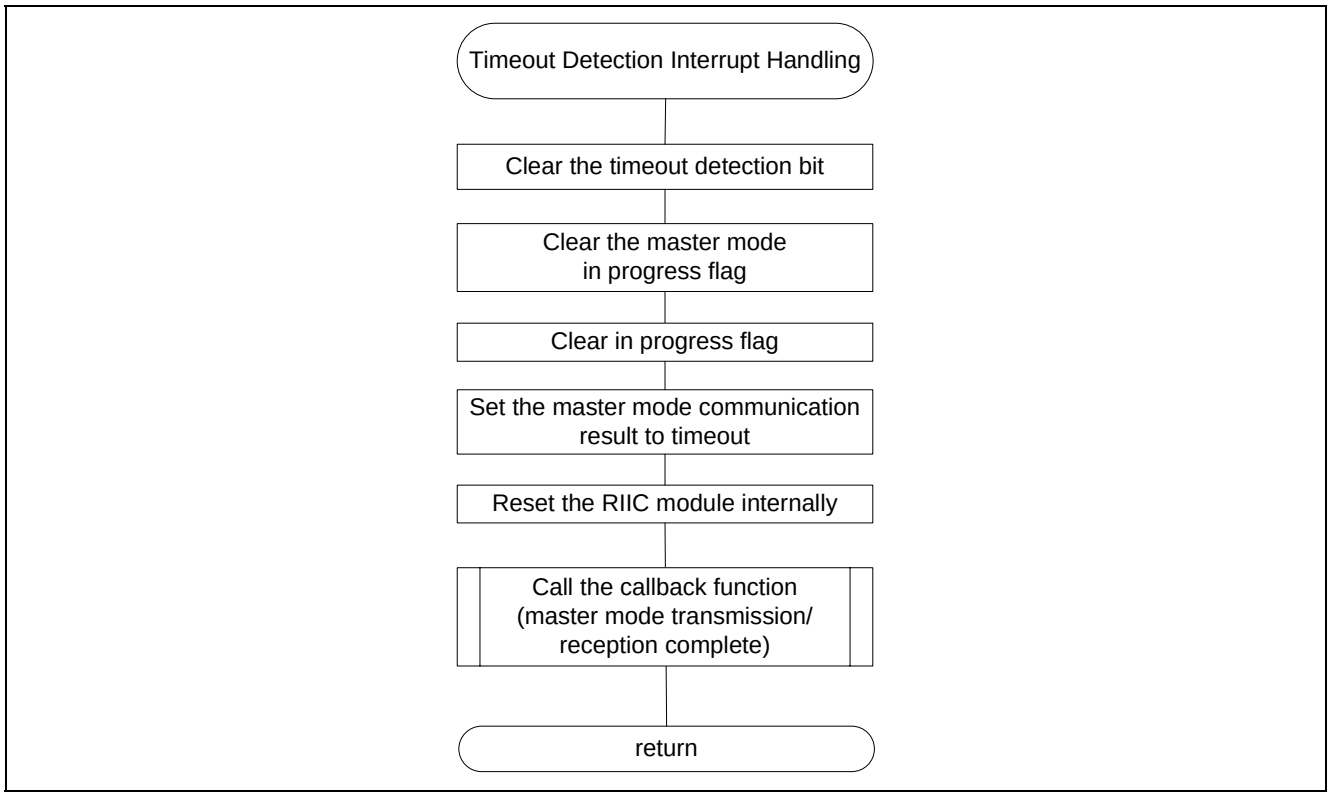

Figure 16 Timeout Detection Interrupt Handling

4. Sample Code

The sample code can be downloaded from the Renesas Electronics Corporation web site.

5. Reference Documents

- RX62N Group, RX621 Group User's Manual: Hardware, Revision 1.11
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Technical Updates and Technical Manuals
  (The latest information can be accessed at the Renesas Electronics Web site.)
- RX Family C Compiler Package, Version.1.0.1.0
  (The latest version can be downloaded from the Renesas Electronics Web site.)

Website and Support

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

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

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.27.11	—	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 type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type 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

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.1