RX62N Group, RX621 Group

Asynchronous SCIa Communication Using the DTCa Module

R01AN0267EJ0101
Rev.1.01
Nov 01, 2018

Introduction

This application note presents a sample program that performs SCI (serial communications interface) asynchronous serial communication using the Renesas MCU's DTC (data transfer controller) module.

Target Devices

The RX62N Group and RX621 Group products

Other members of the RX Family that have the same I/O registers (peripheral unit control registers) as the RX62N Group and RX621 Group products can also use the code from this application note. Note, however, that since certain aspects of the functions used may be changed in other devices due to function additions or other differences, the documentation for the device used must be checked carefully before using this code. When using this code in an end product or other application, its operation must be tested and evaluated thoroughly.

Contents

1. Specifications

The code presented in this application note performs serial communication (transmission and reception) by performing data transfers between RAM and the SCI (serial communications interface) module using the data transfer controller (DTC). Figure 1 presents an overview of this asynchronous serial data transmission and reception that uses the DTC.

1. This sample program uses the SCI channel 2 and the DTC module.
2. An 8-bit length, one stop bit, no parity bit communication format is used.
3. For the transmit operation, the DTC is activated by the transmit data empty interrupt request, and it transfers the transmit data, which is prepared in advance from an arbitrary transfer source to the SCI transmit data register (TDR).
4. For the transmit operation, the DTC is activated by the receive data full interrupt request, and it transfers the receive data from the SCI receive data register (RDR) to an arbitrary transfer destination that is prepared in advance.
5. When the specified number of transfers have been performed, these operations are reinitialized.

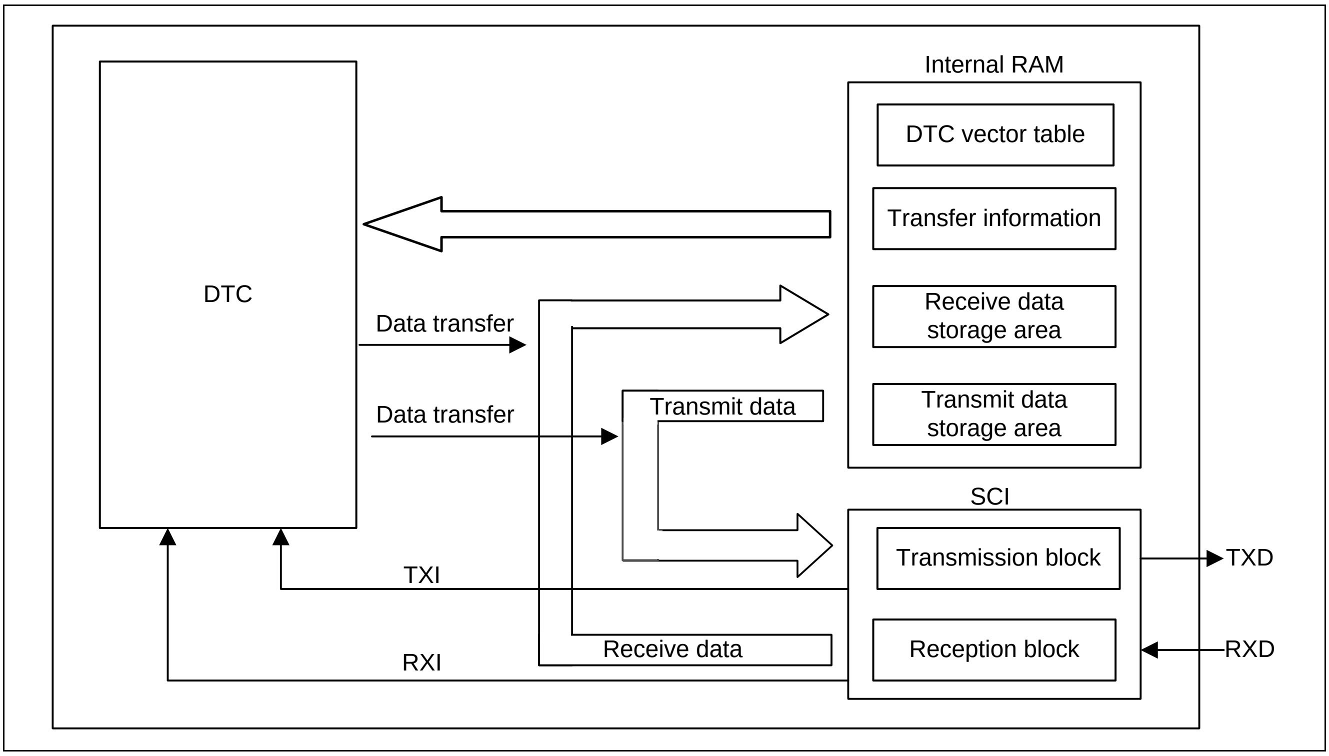

Figure 1 Overview of Asynchronous Serial Data Transmission/Reception Using the DTC

2. Operation Confirmation Environment

Table 1 lists the environment required for confirming master operation.

Table 1 Operation Confirmation Environment

Item	Description
Microcontroller used	R5F562N8BDBG (RX62N Group)
Operating frequency	Main clock: 12 MHz System clock(ICLK):96 MHz Peripheral module clock(PCLK):48 MHz External bus clock(BCLK):24 MHz
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01.007
C compiler	Renesas Electronics Corporation RX Standard Toolchain (V1.2.1.0)
	Compiler options -cpu=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -section=L=C -nologo
iodefine.h version	Version 1.4
emulator	E1
Endian order	Little Endian, Big Endian
Operating mode	single-chip mode
Processor mode	User mode
Sample code version	Version 1.01
Board used	Renesas Starter Kit+ for RX62N (R0K5562N0S100BE)

3. Functions Used

- Clock generation circuit
- Low power consumption functions
- Interrupt control unit (ICU)
- Serial communication interface (SCI)
- Data transfer controller (DTC)

See the RX62N Group, RX621 Group User's Manual: Hardware for detailed information.

4. Operation

4.1 Operation Mode Settings

In the sample program, mode pins are set to MD1 = 1, MD0 = 1 to select single-chip mode as the operating mode, the ROME bit in system control register 0 (SYSCR0) is set to 1 to enable the on-chip ROM, and the EXBE bit in the SYSCR0 register is cleared to 0 to disable the external bus.

Table 2 lists the operating mode settings used in the sample program.

Table 2 Operating Mode Settings

Mode Pin		SYSCR0 Register				
MD1	**MD0**	**ROME**	**EXBE**	**Operating Mode**	**On-Chip ROM**	**External Bus**
1	1	1	0	Single-chip mode	Enabled	Disabled

Note: The initial settings of the ROME and EXBE bits in the SYSCR0 register are SYSCR0.ROME = 1 and SYSCR0.EXBE = 0, so it is not necessary for the sample program to make settings to the SYSCR0 register.

4.2 Clock Settings

The evaluation board used for this application note includes a 12.0 MHz crystal oscillator.

Therefore this application note uses the following settings for the system clock (ICLK), the peripheral module clock (PCLK), and the external bus clock (BCLK): 8× (96 MHz), 4× (48 MHz), and 2× (24 MHz).

4.3 Endian Mode Setting

The sample program presented in this application note supports both big- and little-endian mode. Table 3 lists the hardware endian mode settings of the master device.

Table **3 Endian Mode Settings (Hardware)**

MDE pin	Endian
0	Little endian
1	Big endian

Table 4 lists the endian settings used in the compiler options.

Table 4 Endian Mode Settings (Compiler Options)

MCU Option	Endian
endian = little	Little endian
endian = big	Big endian

Note: Set the MDE pin to match the endian mode selected as a compiler option.

4.4 Bit Order Settings

The program in this application note supports both right and left as the bit order. Table 5 lists the bit order settings in the microcontroller option in the compiler options.

Table 5 Bit Order Settings (Compiler Options)

MCU Option	Bit Order
bit_order = right	Bit field members are allocated in order starting with the low-order bit. (Default)
bit_order = left	Bit field members are allocated in order starting with the high-order bit.

Notes: 1. In this application note, bit fields are used in the I/O register definitions file (iodefine.h). In the I/O register definitions file, "left" is specified with the #pragma bit_order extension, and the bit field members are allocated in order starting with the high-order bit.

2. If both the bit_order compiler option and the #pragma bit_order extension are specified, the #pragma bit_order extension specification takes precedence. Thus the bit fields defined in the I/O register definitions file will be allocated in order starting with the high-order bit, regardless of the compiler options bit_order specification.

4.5 SCI Settings

Table 6 lists the SCI communication function settings used in this sample program.

Table 6 SCI Settings and Conditions

Channel Used	SCI 2
Communication mode	Asynchronous serial communication mode
Interrupts	• Receive error interrupt (ERI) • Receive data full interrupt (RXI) • Transmit data empty interrupt (TXI) • Transmit complete interrupt (TEI)
Communication speed	38,400 bps (PCLK = 48 MHz)
Data length	8-bit data
Stop bits	1 stop bit
Parity	None

4.6 DTC Settings

Table 7 lists the DTC transfer conditions used in this sample program.

Table 7 DTC Transfer Conditions

Condition	SCI Transmit DTC Transfer Conditions (TXI2)	SCI Receive DTC Transfer Conditions (RXI2)
Transfer information	Full address mode	Full address mode
Transfer mode	Normal transfer mode	Normal transfer mode
Transfer count	256	256
Transfer data	Size: byte Data content: 256 bytes with values from H'00 to H'FF	Size: byte Data content: An arbitrary 256 bytes of data
Transfer source	Internal RAM	Receive data register (SCI2.RDR)
Transfer destination	Transmit data register (SCI2.TDR)	Internal RAM
Transfer source address	The transfer source address is incremented after the transfer	The transfer source is fixed
Transfer destination address	The transfer destination is fixed	The transfer destination address is incremented after the transfer
Start event	Started on the SCI transmit data empty interrupt	Started on the SCI receive data full interrupt
Interrupts	An interrupt is enabled to the CPU after the specified data transfer completes.	An interrupt is enabled to the CPU after the specified data transfer completes.

4.7 Operation Timing

Figure 2 shows the timing of the operations performed by this sample program.

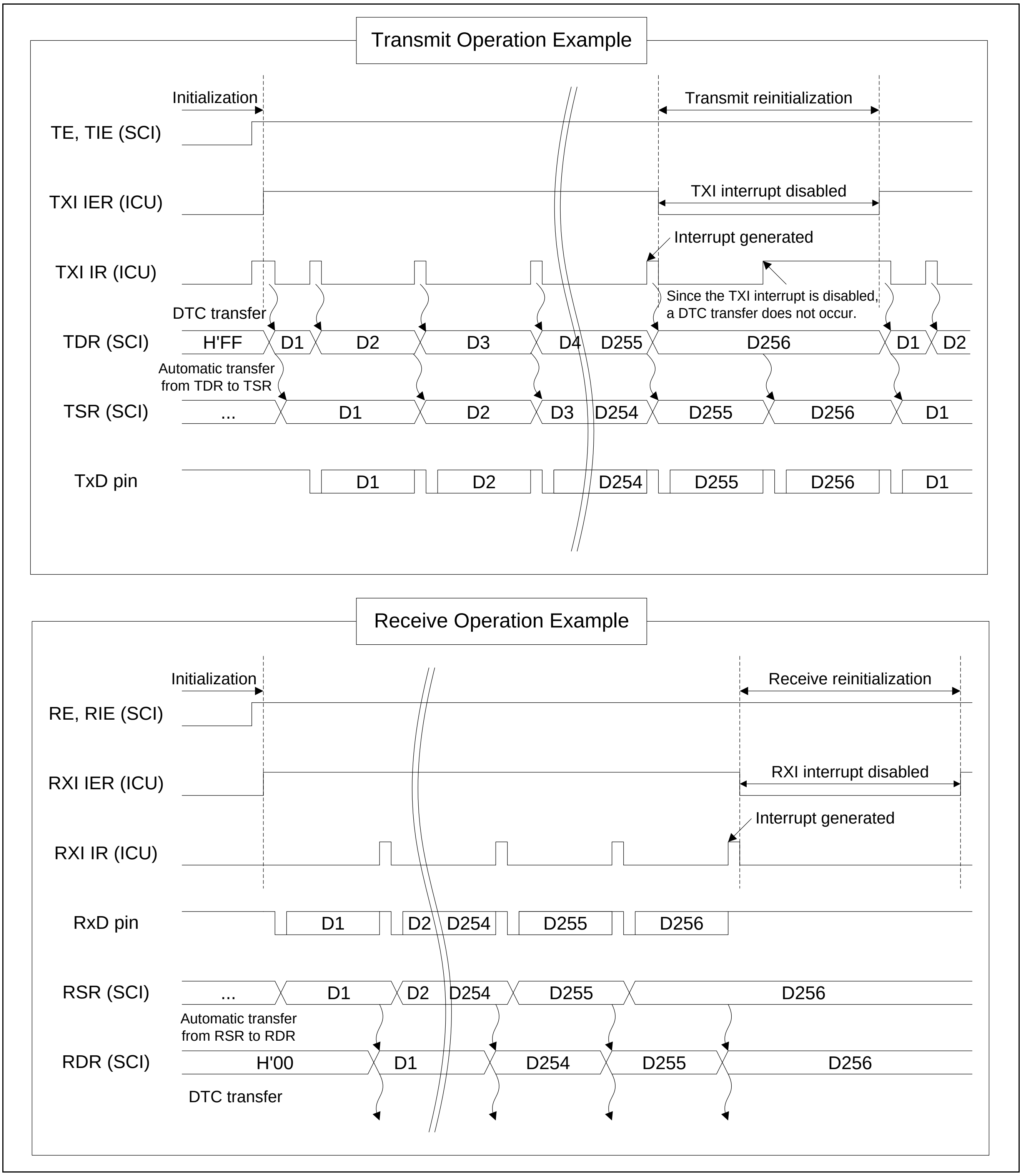

Figure 2 Operation Timing

Note: Note that when the DTC module is used in combination with communication functions in the RX62N Group and RX621 Group microcontrollers, if the next transfer request occurs before the IR flag is cleared automatically, the transfer request will be lost.
See section 11.7, Usage Notes, in the RX62N Group and RX621 Group User's Manual - Hardware for details.

5. Software

5.1 Constants

Table 8 lists the constants used in the sample code.

Table 8 Constants

Constant Name	Set Value	Usage
TXD_MAX	256	DTC transfer count

5.2 Structures and Unions

Figure 3 shows the structures and unions used in the sample program.

```
struct st_dtc_full{
 union{
     unsigned long LONG;
     struct{
        unsigned long MRA_MD     :2;    /* The MRA.MD bit */
        unsigned long MRA_SZ     :2;    /* The MRA.SZ bit */
        unsigned long MRA_SM     :2;    /* The MRA.SM bit */
        unsigned long            :2;
        unsigned long MRB_CHNE   :1;    /* The MRB.CHNE bit */
        unsigned long MRB_CHNS   :1;    /* The MRB.CHNS bit */
        unsigned long MRB_DISEL  :1;    /* The MRB.DISEL bit */
        unsigned long MRB_DTS    :1;    /* The MRB.DTS bit */
        unsigned long MRB_DM     :2;    /* The MRB.DM bit */
        unsigned long            :2;
        unsigned long            :16;
     }BIT;
 }MR;
 void*SAR; /* The SAR register */
 void*DAR; /* The DAR register */
 struct {
   unsigned long CRA:16   /* The CRA register */
   unsigned long CRB:16   /* The CRB register */
 }CR;
};
```

Figure 3 Structures and Unions Used in the Sample Code (DTC Transfer Information)

5.3 Variables

Table 9 lists the variables used in the sample program.

Table 9 Variables

Type	Variable	Usage	Functions
unsigned char	recvBuff[BUF_SIZE]	Array variable that holds the serial receive data	main, dtc_init, int_sci_rxi2
unsigned char	trnsBuff[BUF_SIZE]	Array variable that holds the serial transmit data	dtc_init, int_sci_tei2
st_dtc_full	dtc_rx	Structure variable that holds the DTC transfer information for SCI reception	dtc_init, int_sci_rxi2
st_dtc_full	dtc_tx	Structure variable that holds the DTC transfer information for SCI transmission	dtc_init, int_sci_tei2
void*	dtc_table[256]	DTC vector table that allocates addresses for the dtc_rx/dtc_tx DTC transfer information	dtc_init

5.4 Functions

Table 10 lists the functions used in this application note's sample program.

Table 10 Functions

Function Name	Operation
HardwareSetup	Initialization, clock settings, and clearing the module stop state
main	Main processing
icu_init	ICU initialization and setting the interrupt levels
sci2_init	SCI initialization, transfer clock DTC settings
dtc_init	DTC initialization, transfer information setup, DTC vector base register setting, and enabling DTC activation
int_sci_txi2	Transfer interrupt
int_sci_tei2	Transmit complete interrupt
int_sci_rxi2	Receive interrupt
int_sci_eri2	Receive error interrupt

5.5 Processing Flow

Figures 4 to 12 show the processing flow of the sample program.

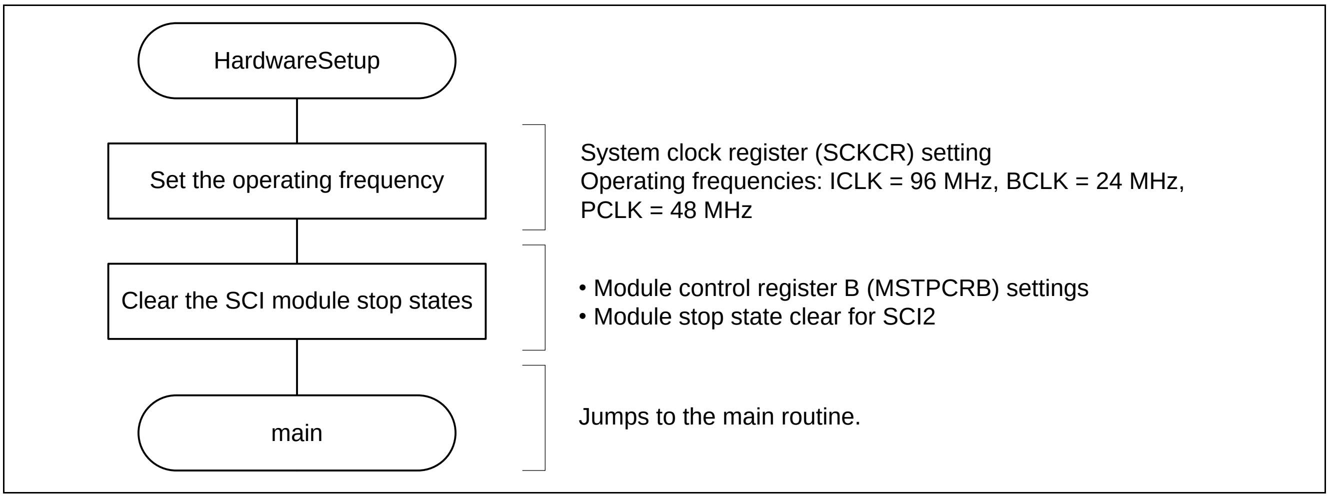

Figure 4 Initialization Processing

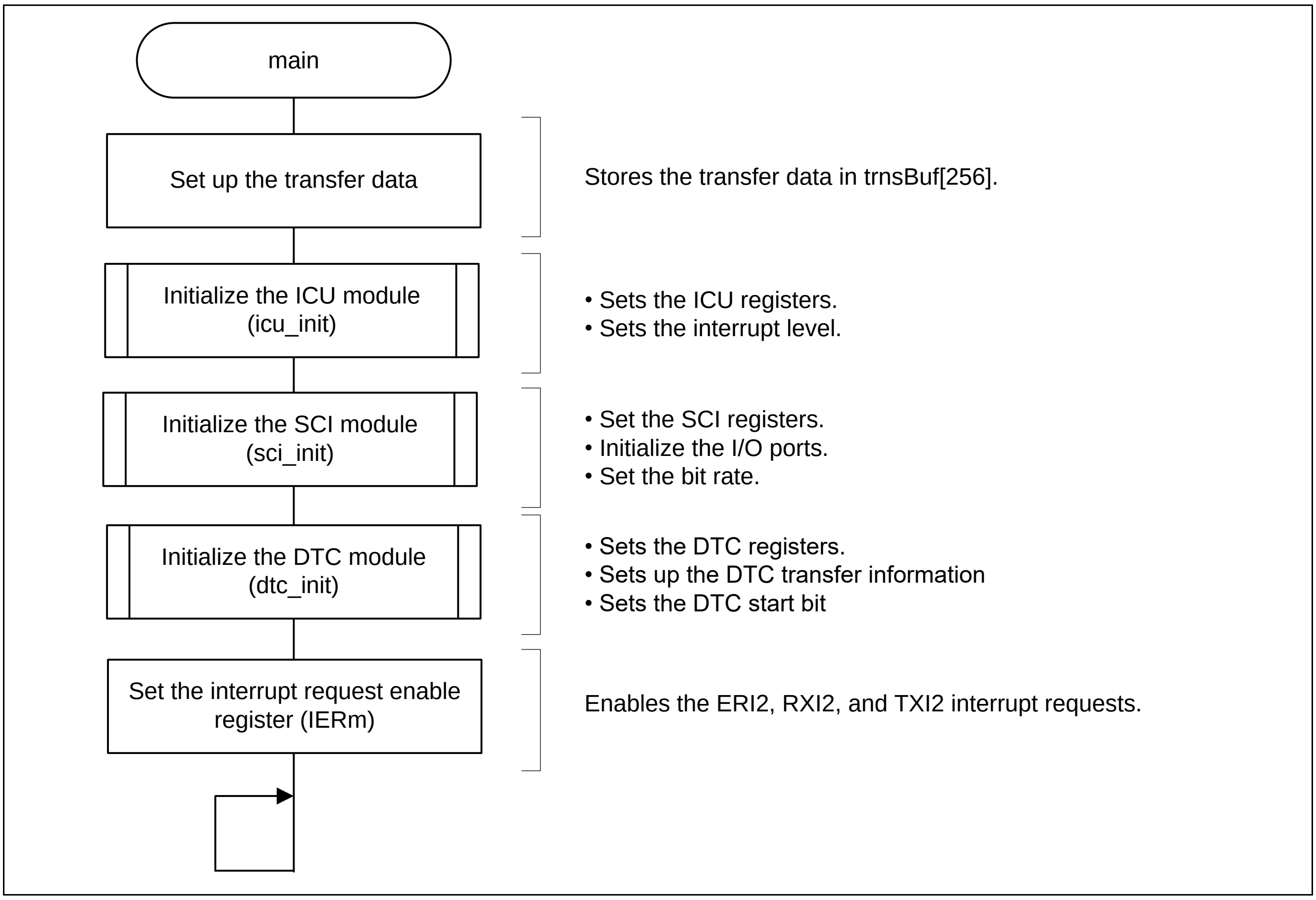

Figure 5 Main Processing

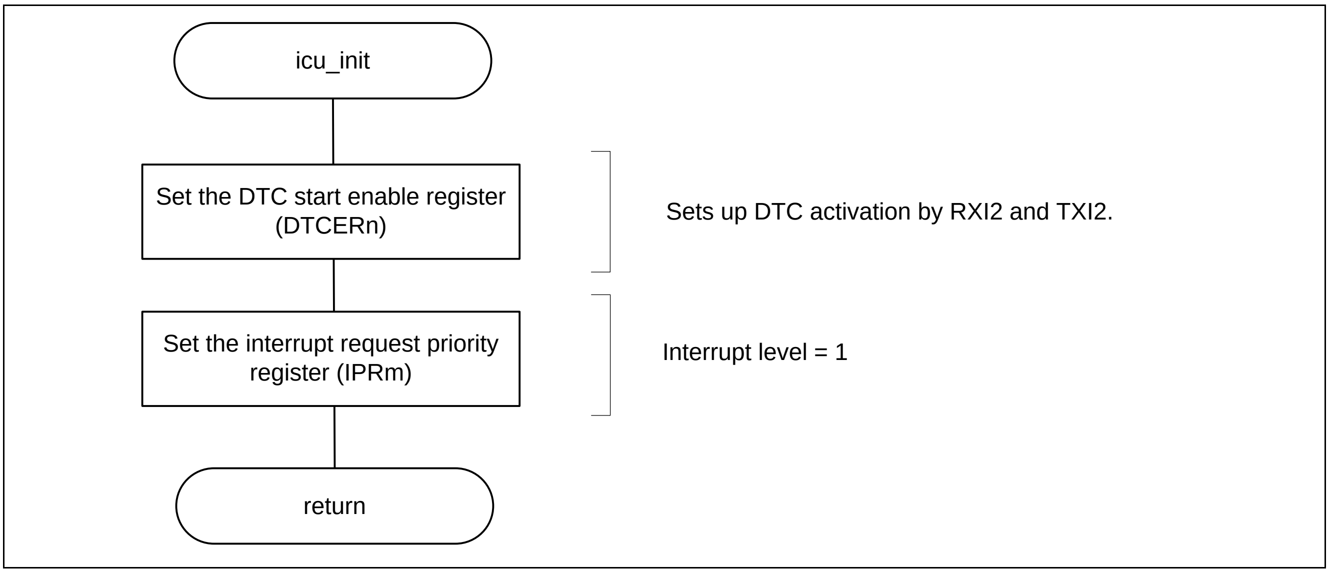

Figure 6 ICU Initialization

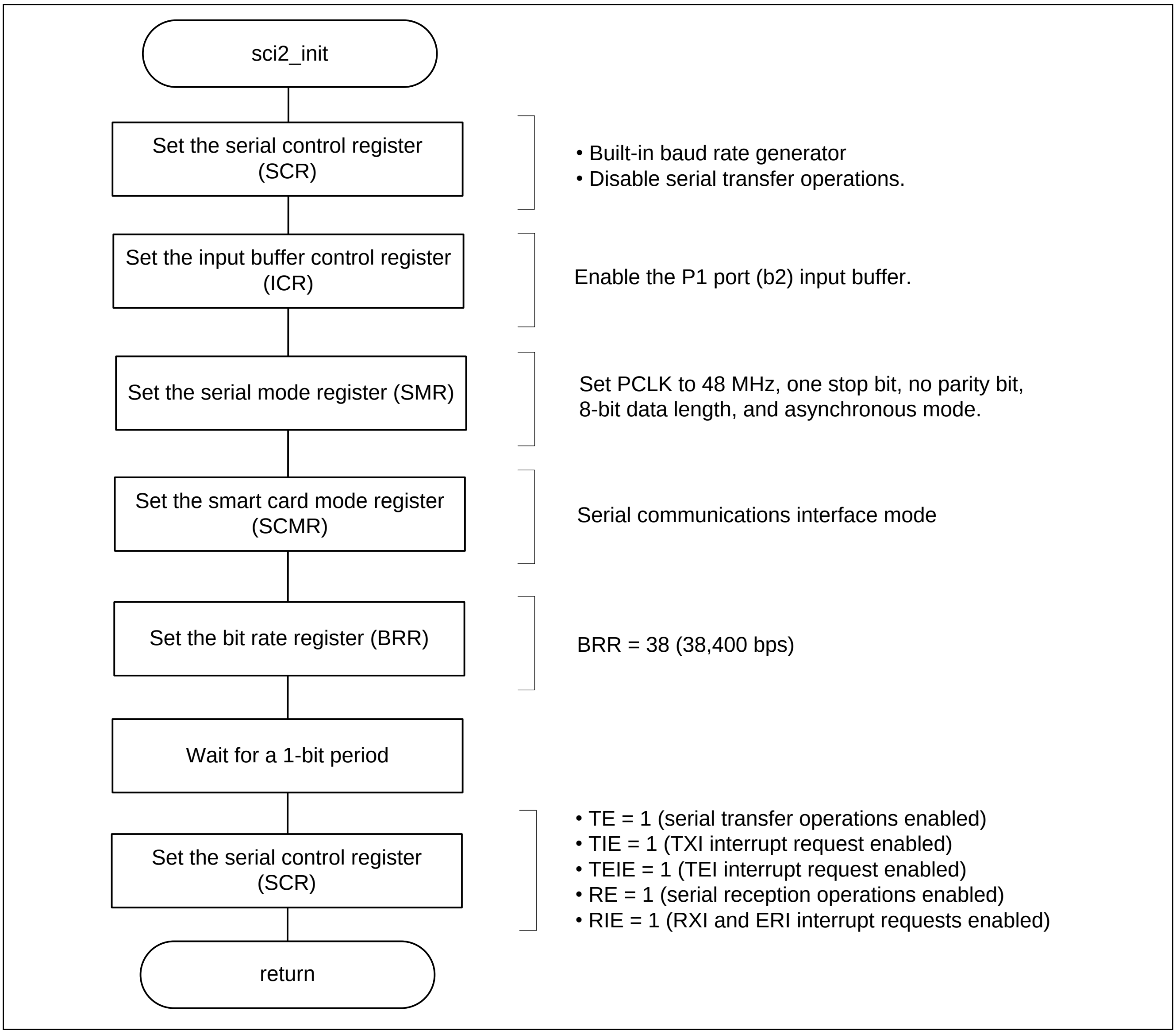

Figure 7 SCI Initialization

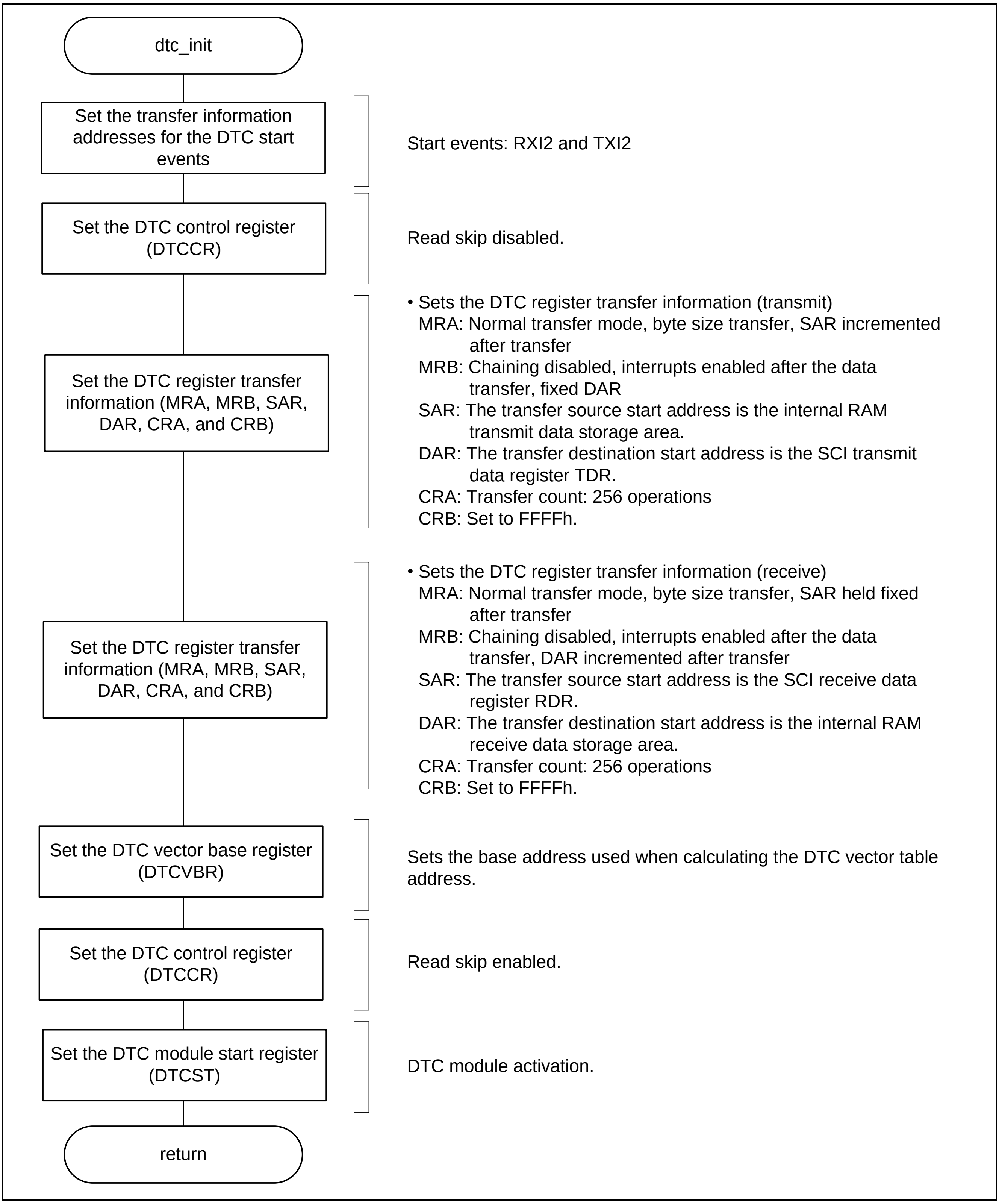

Figure 8 DTC Initialization

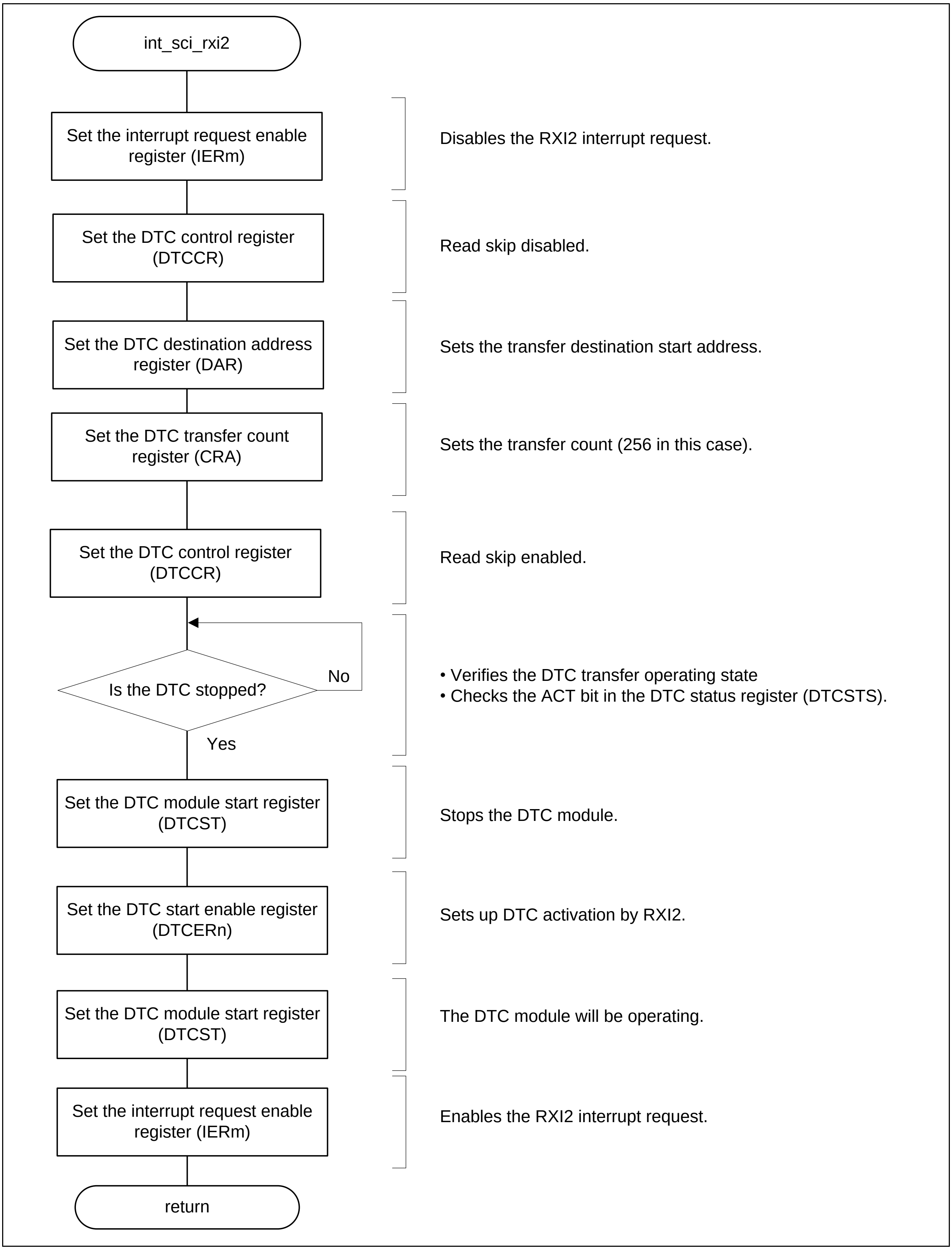

Figure 9 Receive Interrupt

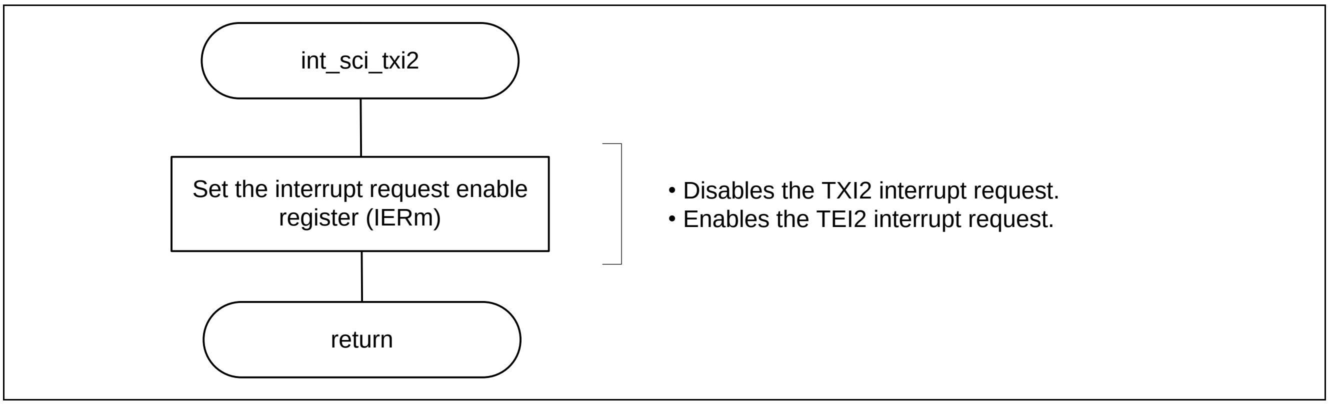

Figure 10 Transmit Interrupt

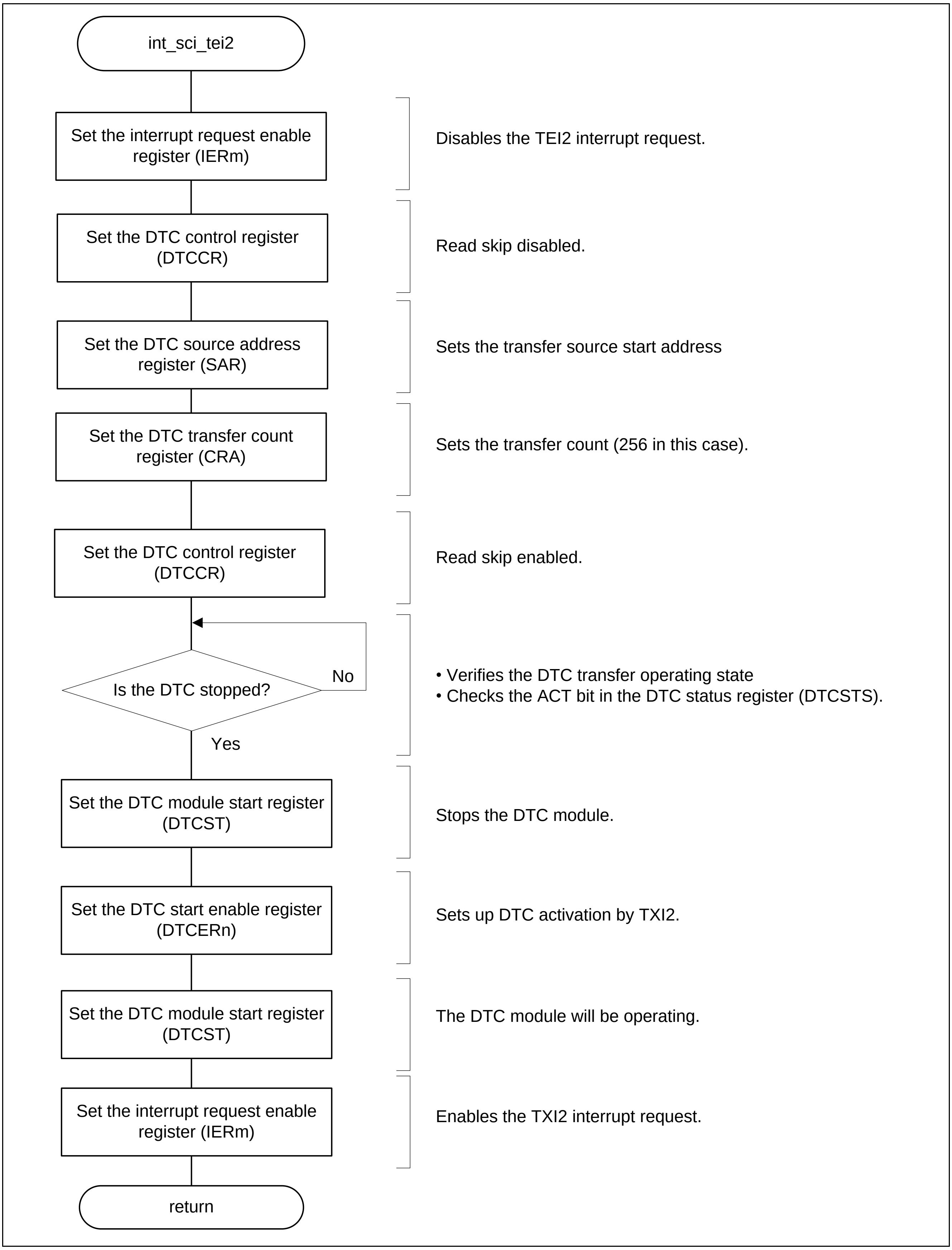

Figure 11 Transmit Complete Interrupt

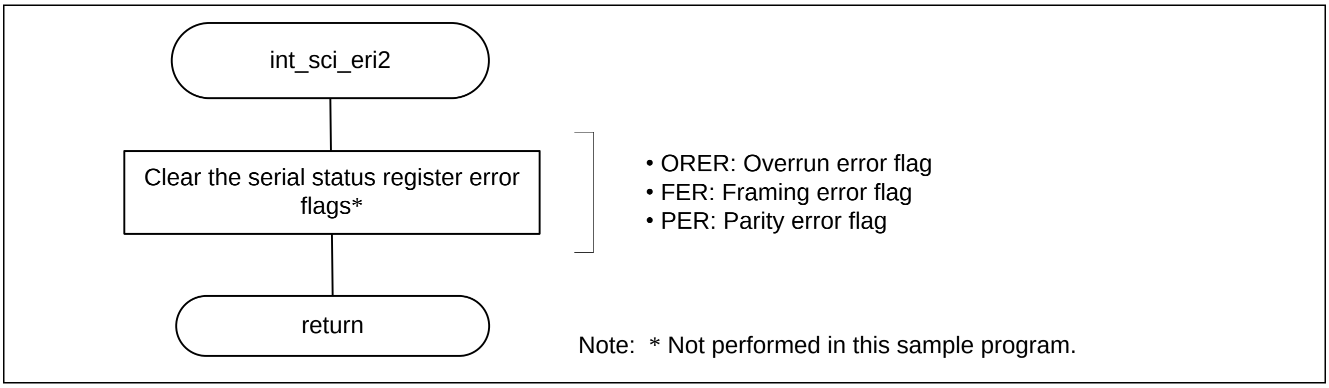

Figure 12 Receive Error Interrupt

6. Reference Documents

- Hardware Manual
  RX62N Group, RX621 Group User's Manual: Hardware
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Software Manual
  RX Family User's Manual: Software
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Development Environment Manual
  RX Family C/C++ Compiler Package User's Manual
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Technical Updates
  (The latest information can be downloaded from the Renesas Electronics Web site.)

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.14.11	—	First edition issued
1.01	Nov.01.18	3	Table 1 Operation Confirmation Environment changed

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

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

   ¾ 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.

   ¾ 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.

   ¾ 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.

   ¾ 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 a product with a different part number, confirm that the change will not lead to problems.

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

RENESAS

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

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

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.2