RL78/G14

Using the DTC Chain Transfer to Perform 3-Wire Serial I/O (Master Transmission/Reception) by Serial Array Unit in HALT Mode

Introduction

This application note explains to perform the master transmission of 3-wire serial I/O (CSI) using the DTC chain transfer in HALT mode.

Performs the master transmission/reception of CSI in the low power consumption HALT mode using DTC and serial array unit. The data size transmitted and received can be changed in 2 to 16 bytes range.

Target Device

RL78/G14

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

Contents

1. Introduction

In this application note, performs data transmission and reception in the low power consumption HALT mode using DTC and serial array unit (SAU). The data size transmitted and received can be changed in 2 to 16 bytes range.

Performs output of a transfer clock from SCK00 pin, output of transmitting data from SO00 pin, and input of receiving data from SI00 pin by using SAU as the CSI.

DTC transfers transmission and reception data from the transfer source address to the transfer destination address. The transfer end interrupt of CSI activates DTC.

Table 1.1 lists the peripherals functions and their applications. **Figure 1.1**shows the timing and communication format.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
SAU (unit 0, channel 0)	Performs clock synchronous serial communication
DTC	Transfers transmit data and receive data

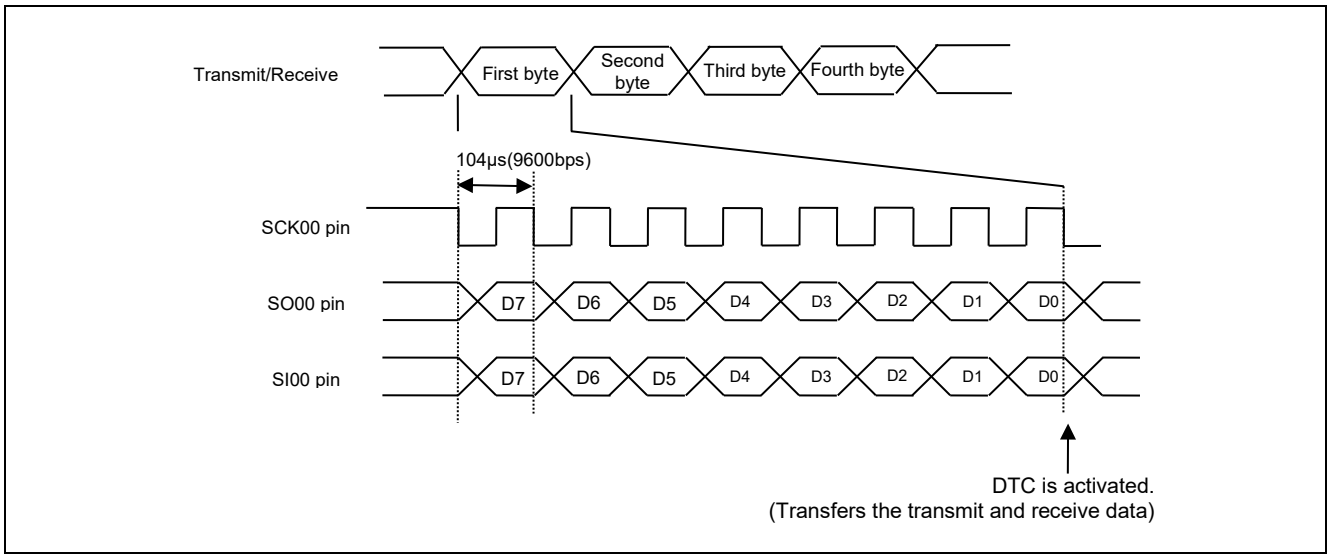


Figure 1.1 Timing and Communication Format (Transmitting/Receiving data is 4 bites)

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	RL78/G14 (R5F104PJAFB)
Operating frequencies	• High-speed on-chip oscillator clock (f_{HOCO}): 32 MHz (typical) • CPU/peripheral hardware clock (f_{CLK}): 32 MHz
Operating voltage	5.0 V (operation enabled from 2.9 to 5.5 V) LVD operation (V_{LVI}): 2.81 V at the rising edge or 2.75 V at the falling edge in reset mode
Integrated development environment (CS+)	Renesas Electronics Corporation CS+ V2.02.00
C compiler (CS+)	Renesas Electronics Corporation CA78K0R V1.70
Integrated development environment (e2studio)	Renesas Electronics Corporation e2studio V3.0.0.22
C compiler (e2studio)	Renesas Electronics Corporation KPIT GNURL78-ELF Toolchain V14.0.1
Board to be used	RL78/G14 CPU board (QB-R5F104PJ-TB)

3. Related Application Note

The application note that is related to this application note is listed below for reference.

- RL78/G13 Initialization (R01AN0451E) Application Note
- RL78/G14 How to Use the DTC for the RL78/G14 (R01AN0861E) Application Note
- RL78/G14 Using the DTC to Perform Continuous Clock Synchronous Serial Communication (R01AN1504E) Application Note
- RL78/G14, RL78/G1C, RL78/L12, RL78/L13, RL78/L1C Clock Synchronous Single Master Control Software Using CSI Mode of Serial Array Unit (R01AN1195E) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

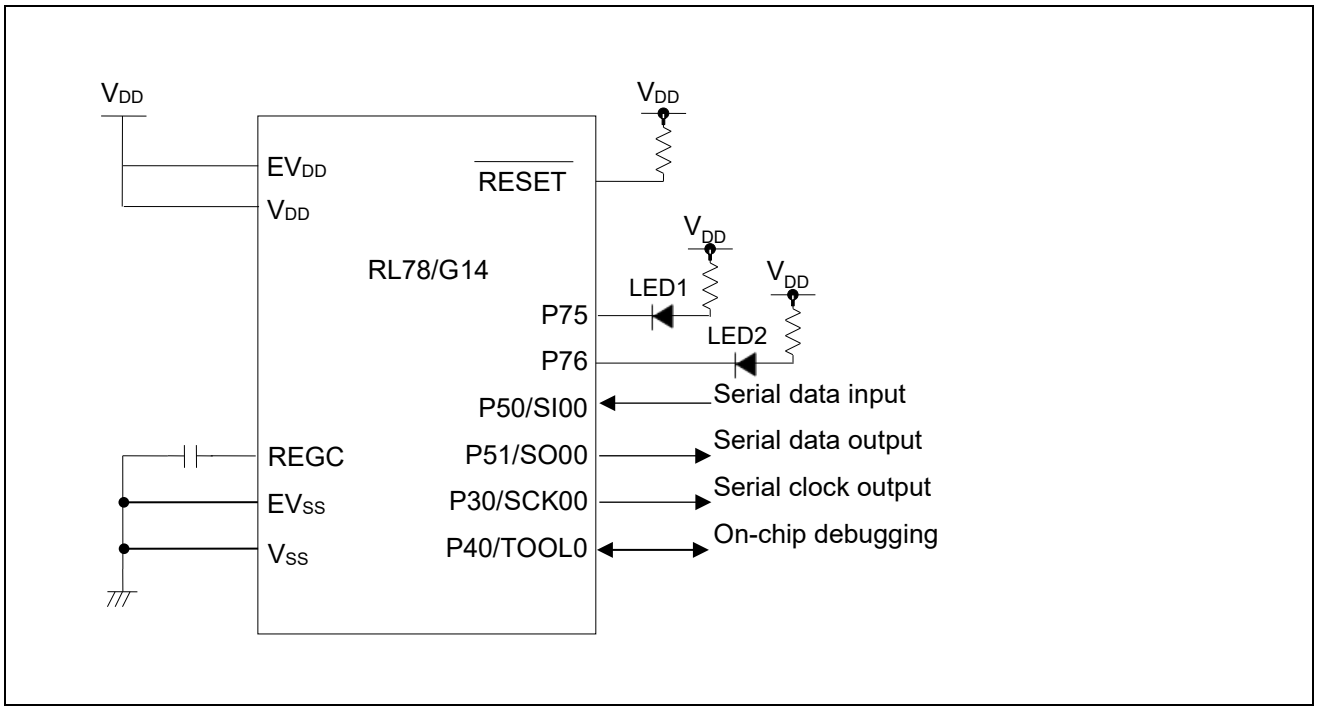


Figure 4.1 Hardware Configuration

Notes:
1. The above figure is simplified to show an overview of the hardware connection. When designing circuits, make sure to handle unused pins appropriately to satisfy the electrical characteristics. Connect input-only ports independently to either V_{DD} or V_{SS} via resistors.
2. Connect pins with names that begin with EV_{SS} to V_{SS}, and pins with names that begin with EV_{DD} to V_{DD}.
3. Make sure to set V_{DD} greater than the detection voltage (VLVI) specified by the LVD.

4.2 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
P75	Output	LED1 control
P76	Output	LED2 control
P50/SI00	Input	Serial data input
P51/SO00	Output	Serial data output
P30/SCK00	Output	Serial clock output

5. Software

This sample code uses the generating code function of compiler. And CS+ version or e2studio version uses the code generator property in order to edit functions generated. Since the mode of code generation is set up for "do nothing if a file exists" as follows, even if the code generation is performed, the file which already exists in a project is not updated. Please note that the file which exists in a project is updated but this sample code will operate abnormally, when the mode was set up for "merge a file" or "overwrites a file" and then code generation was performed.

Figure 5.1 and Figure 5.2 show the code generator property setting.

- CS+

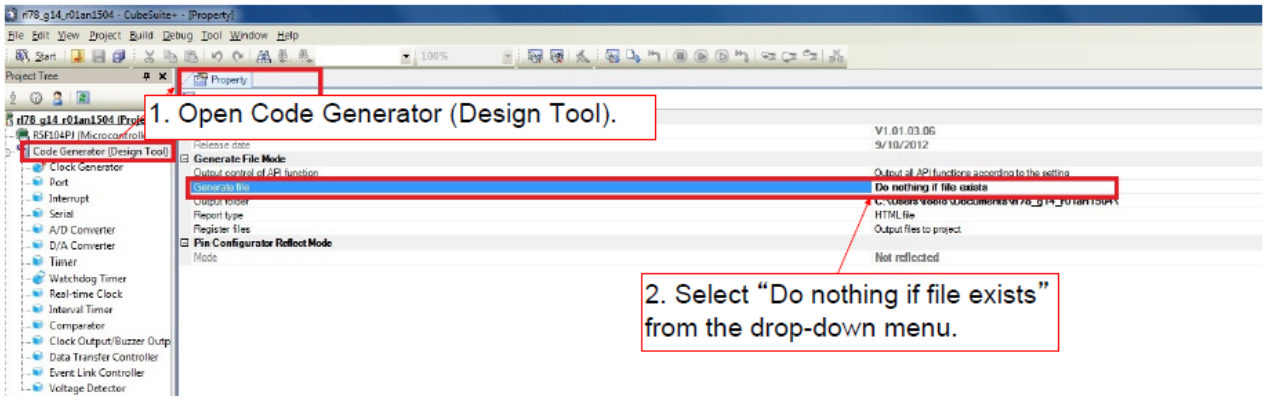


Figure 5.1 Code Generator Property Setting (CS+)

- e2studio

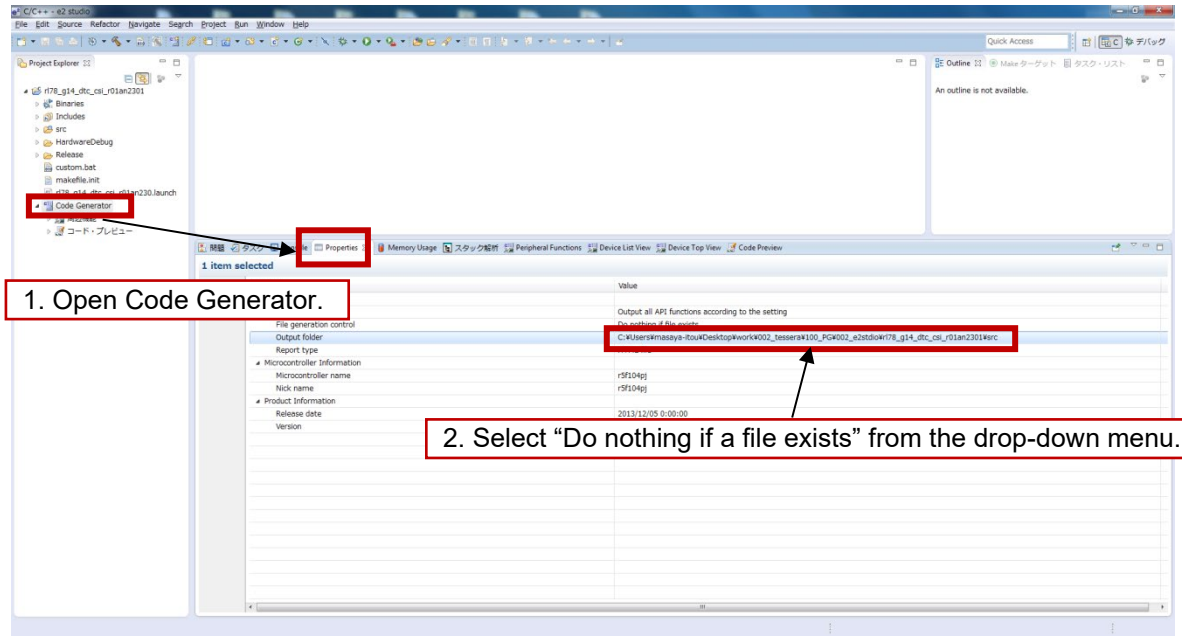


Figure 5.2 Code Generator Property Setting (e2studio)

5.1 Operation Overview

In this application note, transmits/receives data (2 to 16 bites) in the low power consumption HALT mode using DTC and SAU.

DTC transfers transmission and reception data from the transfer source address to the transfer destination address. The transfer end interrupt of CSI activates DTC.

Performs output of a transfer clock from SCK00 pin, data transmission from SO00 pin, and data receiving from SI00 pin by using SAU as the CSI.

Operations in case the size of transmitted and received data is set as 4 bytes are indicated to following the (1) to (12). The data size transmitted and received can be changed in 2 to 16 bytes range by constant TX_RX_DATA_SIZE.

(1) **Initializes ports.**

<Setting conditions>

- Sets P75 and P76 to High output, and put the light of LED1 and LED2.
- Sets P50 (serial data input) to Low output.
- Sets P30 (serial clock output) and P51 (serial data output) to High output.

(2) **Initializes SAU.**

<Setting conditions>

- Uses single transfer mode.
- Sets the data length to 8 bits.
- Sets the data transfer sequence to MSB first.
- Sets the data transmit/receive timing to type 1.
- Sets the baud rate to 9600 bps.
- Sets the interrupt priority level to 3.

(3) **Initializes DTC.**

Sets the control data 0 (DTCD0) and control data 1 (DTCD1).

< DTCD0 setting conditions>

- Sets the activation source to “CSI00 transfer end”.
- Selects “enable the chain transfer”.
- Sets the transfer mode to “normal mode”.
- Sets the data length to “8 bits”.
- The transfer source is FFF10H (SIO00 register address), and sets the transfer source address control to “fixed”.
- The transfer destination is FE900H, and sets the transfer destination address control to “incremented”.
- Sets the number of transfers to a constant (TX_RX_DATA_SIZE － 1).
  (E.g. When TX_RX_DATA_SIZE is 4, set the number of transfer of DTC to 3.)
- Sets the transfer block size to 1 byte.

< DTCD1 setting conditions>

- Selects “disable the chain transfer”.
- Sets the transfer mode to “normal mode”.
- Sets the data length to “8 bits”.
- The transfer source is FE911H, and sets the transfer destination address control to “incremented”.
- The transfer destination is FFF10H (SIO00 register address), and sets the transfer source address control to “fixed”.
- Sets the transfer block size to 1 byte.

*Since “enable the chain transfer” is selected by DTCD0, DTC transmission of DTCD1 starts after the completion of DTC transmission of DTCD0.

(4) **Initializes the main processing.**

<Setting conditions>

- Sets “00H” into the variable (rcv_data[0 to 15]).
- Sets transmission data into the variable (snd_data[0 to 15]).
  (Transmission is described in Table 5.3 Constant Used in the Sample Code.)
- Activates CSI00.
  - Sets “1” (serial clock output value is “1”) into CKO00 bit of SO0 register.
  - Sets “0” (serial data output value is “0”) into SO00 bit of SO0 register.
  - Sets “1” (enables output by serial communication operation) into SOE00 bit of SOE0 register.
  - Sets “1” (sets the SE00 bit to 1 and enters the communication wait status) into SS00 bit of SS0 register.
  - Sets “0” (enable interrupt) into IF0H register and CSIIF00 register.
  - Sets “0” (not perform interrupt mask) into MK0H register and CSIMK00 register.

(5) **Activates DTC.**

<Setting conditions>

- Sets “1” (activation enable) into DTCEN13 bit of DTCEN1 register.

(6) **Starts transmission/reception of CSI00. 1 byte of the beginning is set up by software and the remaining 3 bytes are set up by DTC.**

- When transmitted/received data size is less than one, it becomes an error, and it becomes a normal end when other.

(7) **Shifts to HALT mode and waits for completion of DTC transmissionCSI00 when transmission and reception of CSI00 are started normally. Lights LED1 and LED2, shifts to HALT mode, and does not perform subsequent processing when transmission and reception of CSI00 are not started normally.**

(8) **The activation source of DTC (transfer end interrupt of SAU (INTCSI00)) is generated, and starts data transmission by activation of DTC. (DTC transmission is performed 3 times when TX_RX_DATA_SIZE is 4.)**

- Transfers received data to RAM. Transfers data (1 byte) of 0xFFF10 (SIO00 register) to 0xFE90n(n:0 to 2) (received data (rcv_data[n](n:0 to 2)).

  It is necessary to read the 4th byte of received data by software.

  - In the case of the 1st byte, transmits the data of 0xFFF10 (SIO00 register) to 0xFE900 (rcv_data [0]).
  - In the case of the 2nd byte, transmits the data of 0xFFF10 (SIO00 register) to 0xFE901 (rcv_data[1]).
  - In the case of the 3rd byte, transmits the data of 0xFFF10 (SIO00 register) to 0xFE902 (rcv_data[2]).

- Transfers sending data to SIO00 register. Transfers data of 0xFE91n(n:1 to 3) (snd_data[n](n:1 to 3)) to 0xFFF10 (SIO00 register).

  Since the 1st byte of sending data is set up by software, DCT sets the sending data from the 2nd byte.

  - In the case of the 2nd byte, transmits the data of 0xFE911 (snd _data[1]) to 0xFFF10 (SIO00 register).
  - In the case of the 3rd byte, transmits the data of 0xFE912 (snd _data[2]) to 0xFFF10 (SIO00 register).
  - In the case of the 4th byte, transmits the data of 0xFE913 (snd _data[3]) to 0xFFF10 (SIO00 register).

(9) **After the completion of transmission of DTC, transfer end interrupt (INTCSI00)) request of SAU which is the activation source is generated, and returns from HALT mode.**

- Sets “0” (the number of times of transmission/reception) into the variable (g_csi00_tx_count).

(10) **The transfer end interrupt (INTCSI00)) request of SAU is generated by the completion of 4th byte transmission/reception.**

- Reads out data of 0xFFF10 (SIO00 register) into 0xFE903 (received data (rcv_data[3])).
- Copies content of received data rcv_data[] to the variable for storing received data set_rcv_data[].

(11) **Initializes DTC. (The same contents as (3).)**

(12) **(5) to (11) is repeated afterwards.**

Figure 5.3 shows timing of transmission/reception and DTC activation, Figure 5.4 shows DTCD0 operation, and Figure 5.5 shows DTCD1 operation.

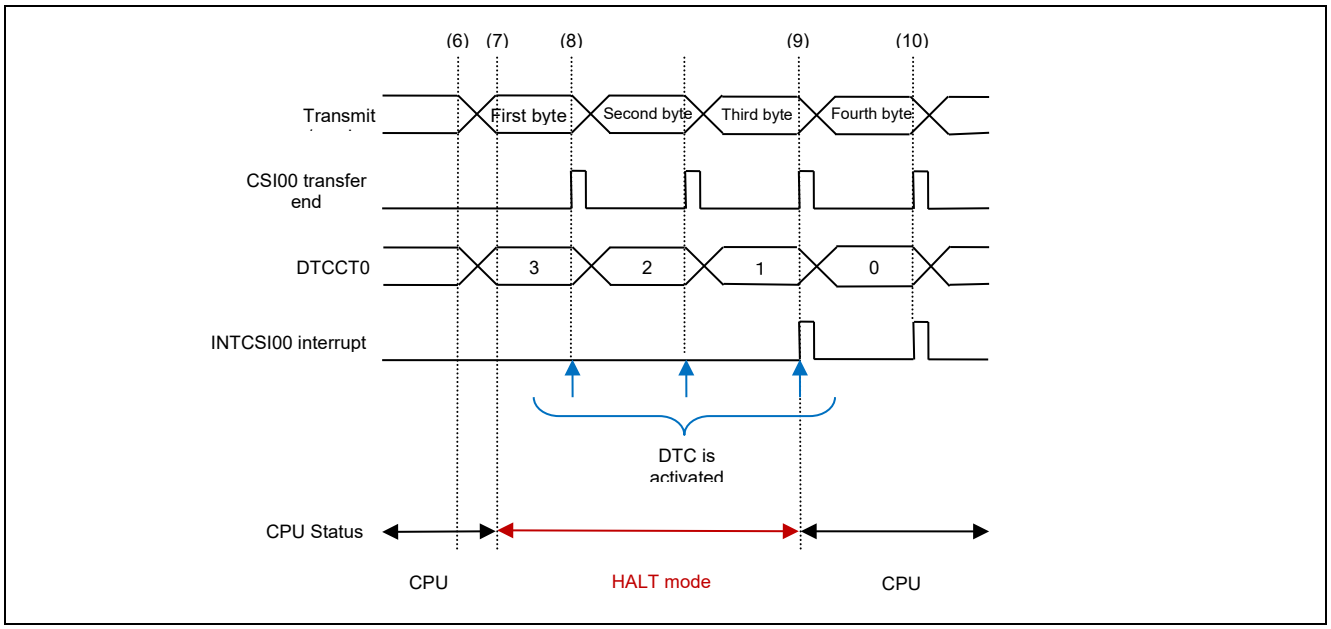


Figure 5.3 Timing of Transmission/Reception and DTC Activation

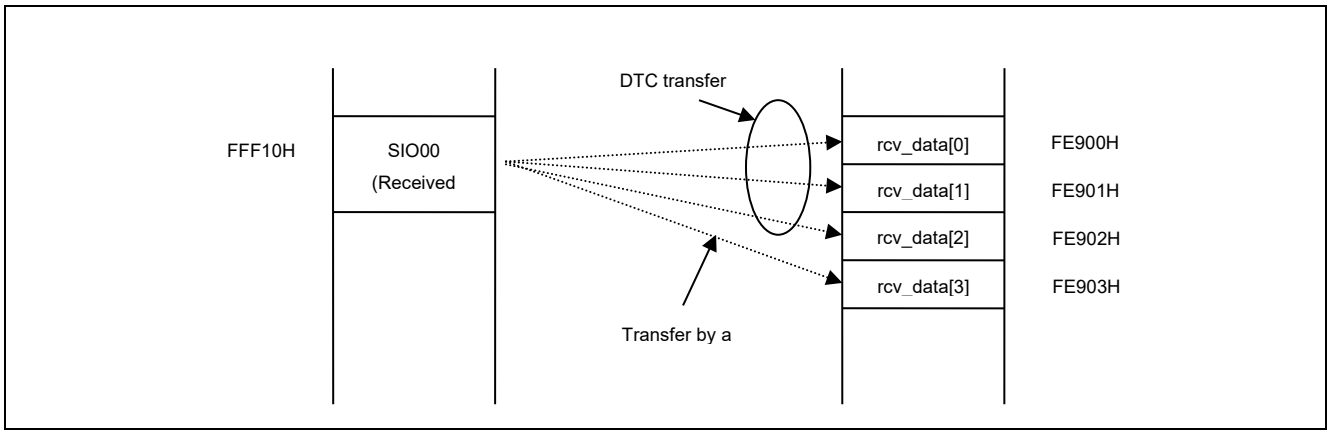


Figure 5.4 DTCD0 Operation

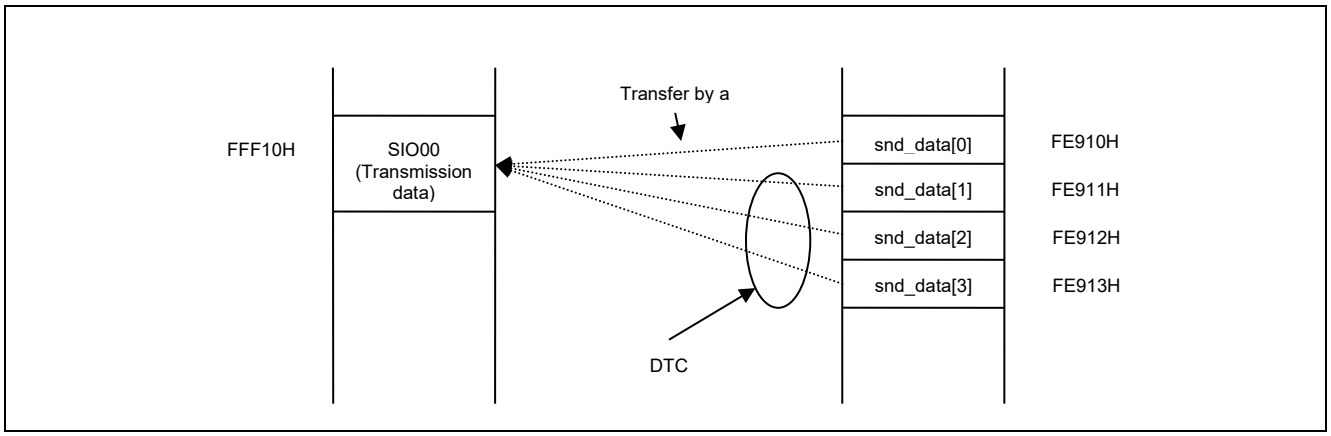


Figure 5.5 DTCD1 Operation

5.2 Section Composition

Table 5.1 lists the sections used in the sample code.

Table 5.1 Section Composition

Address	Start Address	Reference Variable	Description
DTC0DST	0FE900H	rcv_data[]	DTCD0 transfer destination address
DTC1SRC	0FE910H	snd_data[]	DTCD1 transfer source address

5.3 Option Byte Setting List

Table 5.2 lists the option byte settings.

Table 5.2 Option Byte Setting List

Address	Setting Value	Description
000C0H/010C0H	11101111B	Stops the watchdog timer. (counting is stopped when a reset is canceled)
000C1H/010C1H	01111111B	Sets the LVD in reset mode. Detection voltage: 2.81 V at the rising edge, 2.75 V at the falling edge.
000C2H/010C2H	11101000B	Sets the HOCO clock as 32 MHz in high-speed main (HS) mode.
000C3H/010C3H	10000100B	Enables on-chip debugging. The data of a flash memory is deleted at the time of on-chip debug security ID authentication failure.

5.4 Constant

Table 5.3 lists the constant used in the sample code.

Table 5.3 Constant Used in the Sample Code

Constant Name	Setting Value	Description
TX_RX_DATA_SIZE	**4**[Note]	Transmit/receive data size (byte)
BUFFER_SIZE	**16**	Transmit/receive buffer size
SND_DATA1	**55H**	Transmit data (1st byte)
SND_DATA2	**AAH**	Transmit data (2nd byte)
SND_DATA3	**C3H**	Transmit data (3rd byte)
SND_DATA4	**0FH**	Transmit data (4th byte)
SND_DATA5	**F0H**	Transmit data (5th byte)
SND_DATA6	**11H**	Transmit data (6th byte)
SND_DATA7	**80H**	Transmit data (7th byte)
SND_DATA8	**0AH**	Transmit data (8th byte)
SND_DATA9	**01H**	Transmit data (9th byte)
SND_DATA10	**FFH**	Transmit data (10th byte)
SND_DATA11	**B5H**	Transmit data (11th byte)
SND_DATA12	**62H**	Transmit data (12th byte)
SND_DATA13	**33H**	Transmit data (13th byte)
SND_DATA14	**A1H**	Transmit data (14th byte)
SND_DATA15	**79H**	Transmit data (15th byte)
SND_DATA16	**E1H**	Transmit data (16th byte)

Note: It can change in 2 to 16.

5.5 Variables

Table 5.6 lists the global variables, and Table 5.6 lists the static variable.

Table 5.4 Global Variables

Type	Variable Name	Contents	Function Used
uint8_t	rcv_data[]	Receive data	R_MAIN_UserInit r_csi00_interrupt r_csi00_callback_receiveend
uint8_t	snd_data[]	Transmit data	R_CSI00_Send_Receive
uint8_t	set_rcv_data[]	Store the receive data	r_csi00_callback_receiveend
uint8_t	gp_csi00_rx_address	Received data address	R_CSI00_Send_Receive r_csi00_interrupt r_csi00_callback_receiveend
uint8_t	gp_csi00_tx_address	Transmission data address	R_CSI00_Send_Receive
uint16_t	g_csi00_tx_count	The number of times of transmission/reception	R_CSI00_Send_Receive r_csi00_interrupt

Table 5.5 static Variable

Type	Variable Name	Contents	Function Used
MD_STATUS	md_status	Status flag	main

5.6 Functions

Table 5.6 shows functions list.

Table 5.6 Functions List

Function Name	Outline
hdwinit	Initialization
R_Systeminit	Peripheral function initialization
R_CGC_Create	CPU clock initialization
R_PORT_Create	Port initialization
R_SAU0_Create	SAU0 initialization
R_CSI00_Create	CSI00 initialization
R_DTC_Create	DTC initialization
main	Main processing
R_MAIN_UserInit	Main initialization
R_CSI00_Start	CSI00 operation start
R_DTCD0_Start	DTCD0 operation start
R_CSI00_Send_Receive	CSI00 transmission/reception start
r_csi00_interrupt	CSI00 transfer end interrupt
r_csi00_callback_error	CSI00 error callback function
r_csi00_callback_receiveend	CSI00 receive end callback function

5.7 Function Specifications

The following tables list the sample code function specifications.

[Function Name] hdwinit

Outline	Initialization
Header	None
Declaration	void hdwinit(void)
Description	Initializes the peripheral functions.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_Systeminit

Outline	Peripheral function initialization
Header	None
Declaration	void R_Systeminit(void)
Description	Initializes the peripheral functions used in this application note.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_CGC_Create

Outline	CPU clock initialization
Header	r_cg_cgc.h
Declaration	void R_CGC_Create(void)
Description	Initializes the CPU clock.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_PORT_Create

Outline	Port initialization
Header	r_cg_port.h
Declaration	void R_PORT_Create(void)
Description	Initializes the port.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_SAU0_Create

Outline	SAU0 initialization
Header	r_cg_serial.h
Declaration	void R_SAU0_Create(void)
Description	Initializes SAU0.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_CSI00_Create

Outline	CSI00 initialization
Header	r_cg_serial.h
Declaration	void R_CSI00_Create(void)
Description	Initializes CSI00.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_DTC_Create

Outline	DTC initialization
Header	r_cg_dtc.h
Declaration	void R_DTC_Create(void)
Description	Initializes the DTC.
Arguments	None
Return Value	None
Remarks	None

[Function Name] main

Outline	Main processing
Header	None
Declaration	void main(void)
Description	Performs the main processing.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_MAIN_UserInit

Outline	Main initialization
Header	None
Declaration	void R_MAIN_UserInit(void)
Description	Performs processing required to initialize the main processing.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_CSI00_Start

Outline	CSI00 operation start
Header	r_cg_serial.h
Declaration	void R_CSI00_Start(void)
Description	Starts CSI00 operation.
Arguments	None
Return Value	None
Remarks	None

[Function Name] R_DTCD0_Start

Outline	DTCD0 operation start	
Header	r_cg_dtc.h	
Declaration	void R_DTCD0_Start(void)	
Description	Starts the DTCD0 operation.	
Arguments	None	
Return Value	None	
Remarks	None	

[Function Name] R_CSI00_Send_Receive

Outline	CSI00 transmit/receive start	
Header	r_cg_serial.h	
Declaration	MD_STATUS R_CSI00_Send_Receive(uint8_t * const tx_buf, uint16_t tx_num, uint8_t * const rx_buf)	
Description	Prepares the data buffer for CSI00 communication (transmission/reception) and sets the first byte of the transmit data.	
Arguments	uint8_t * const tx_buf	Transmit data buffer pointer
	uint16_t tx_num	Transmit data size
	uint8_t * const rx_buf	Receive data buffer pointer
Return Value	MD_OK	Setting is completed, operation started
	MD_ARGERROR	Argument is incorrect
Remarks	None	

[Function Name] r_csi00_interrupt

Outline	CSI00 transfer end interrupt
Header	None
Declaration	__interrupt static void r_csi00_interrupt(void)
Description	Performs CSI00 transfer end interrupt handling.
Arguments	None
Return Value	None
Remarks	None

[Function Name] r_csi00_callback_error

Outline	CSI00 error callback function	
Header	r_cg_serial.h	
Declaration	static void r_csi00_callback_error(uint8_t err_type)	
Description	This function is called when the CSI00 error occurs.	
Arguments	uint8_t err_type	Error type
Return Value	None	
Remarks	The sample code does not include the error processing. Add processing to the user program as needed.	

[Function Name] r_csi00_callback_receiveend

Outline	CSI00 receive end callback function
Header	r_cg_serial.h
Declaration	static void r_csi00_callback_receiveend(void)
Description	This function is called when receiving the specified number of bytes of data is completed. Receiving data of specified size (TX_RX_DATA_SIZE) is copied to set_rcv_data[BUFFER_SIZE].
Arguments	None
Return Value	None
Remarks	None

5.8 Flowcharts

5.8.1 Overall Flowchart

Figure 5.6 shows the overall flow of the sample code.

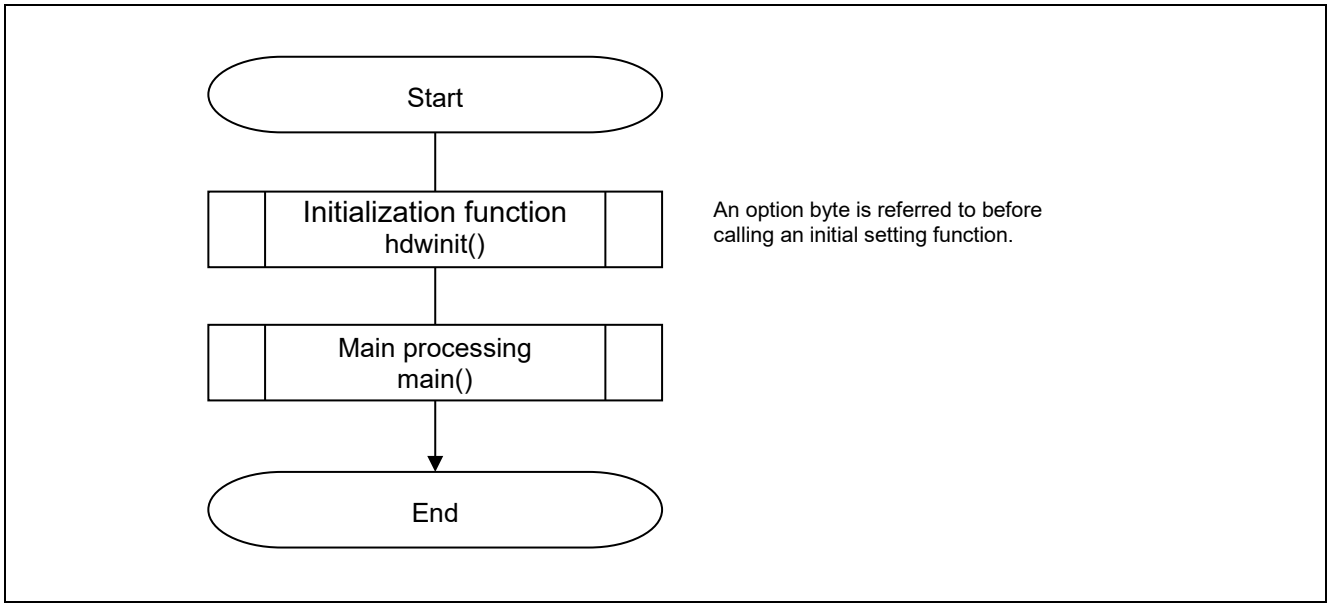


Figure 5.6 Overall Flow

5.8.2 Initialization

Figure 5.7 shows the initialization.

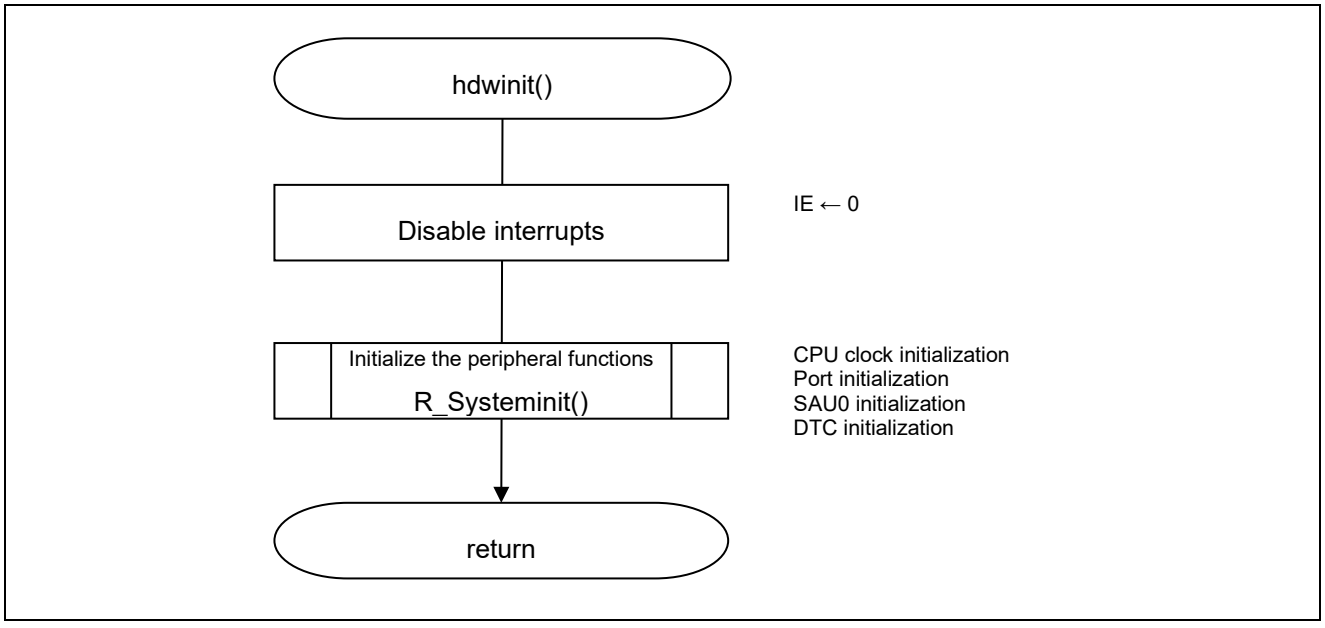


Figure 5.7 Initialization

5.8.3 Peripheral Function Initialization

Figure 5.8 shows the peripheral function initialization.

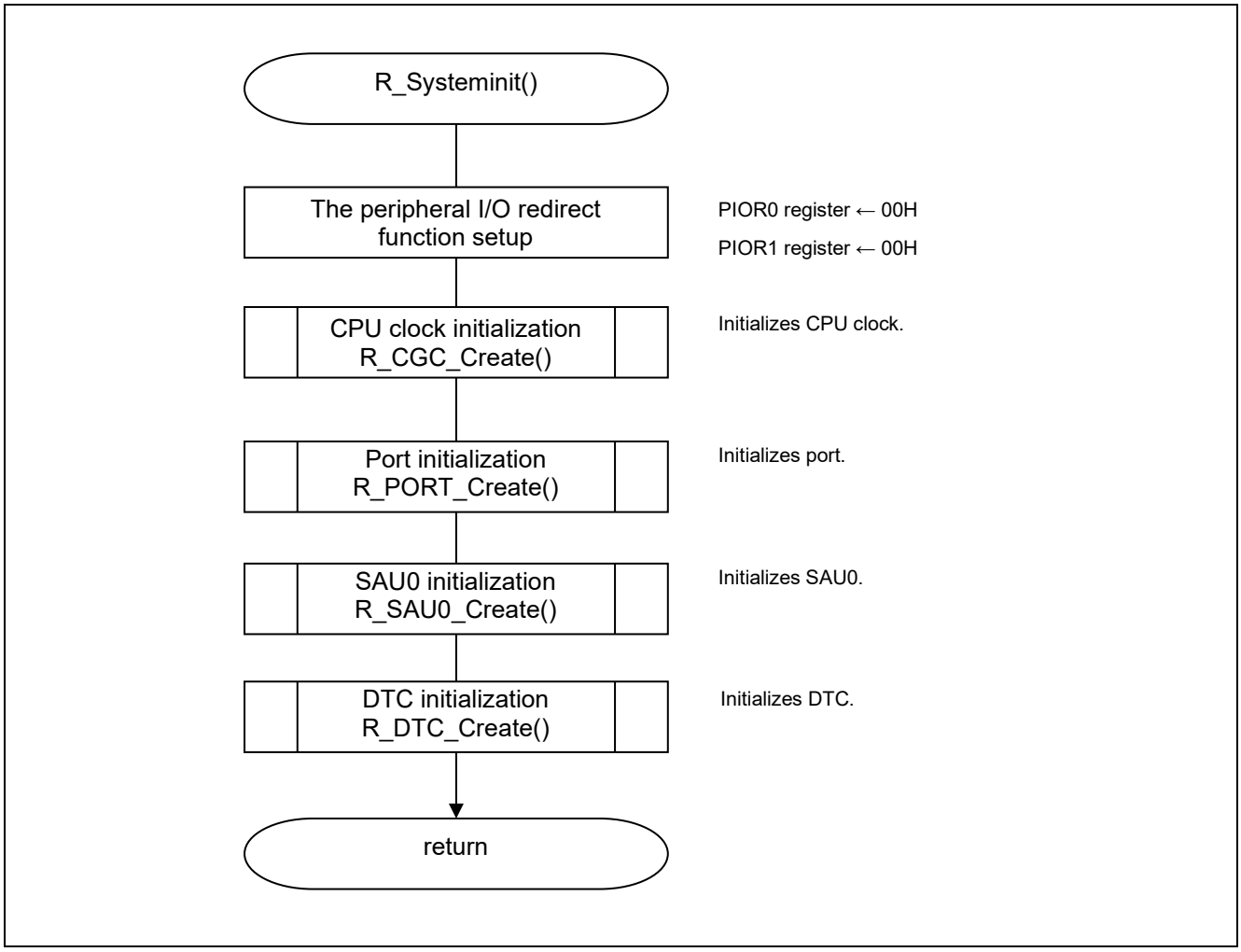


Figure 5.8 Peripheral Function Initialization

5.8.4 CPU Clock Initialization

Figure 5.9 shows the CPU clock initialization.

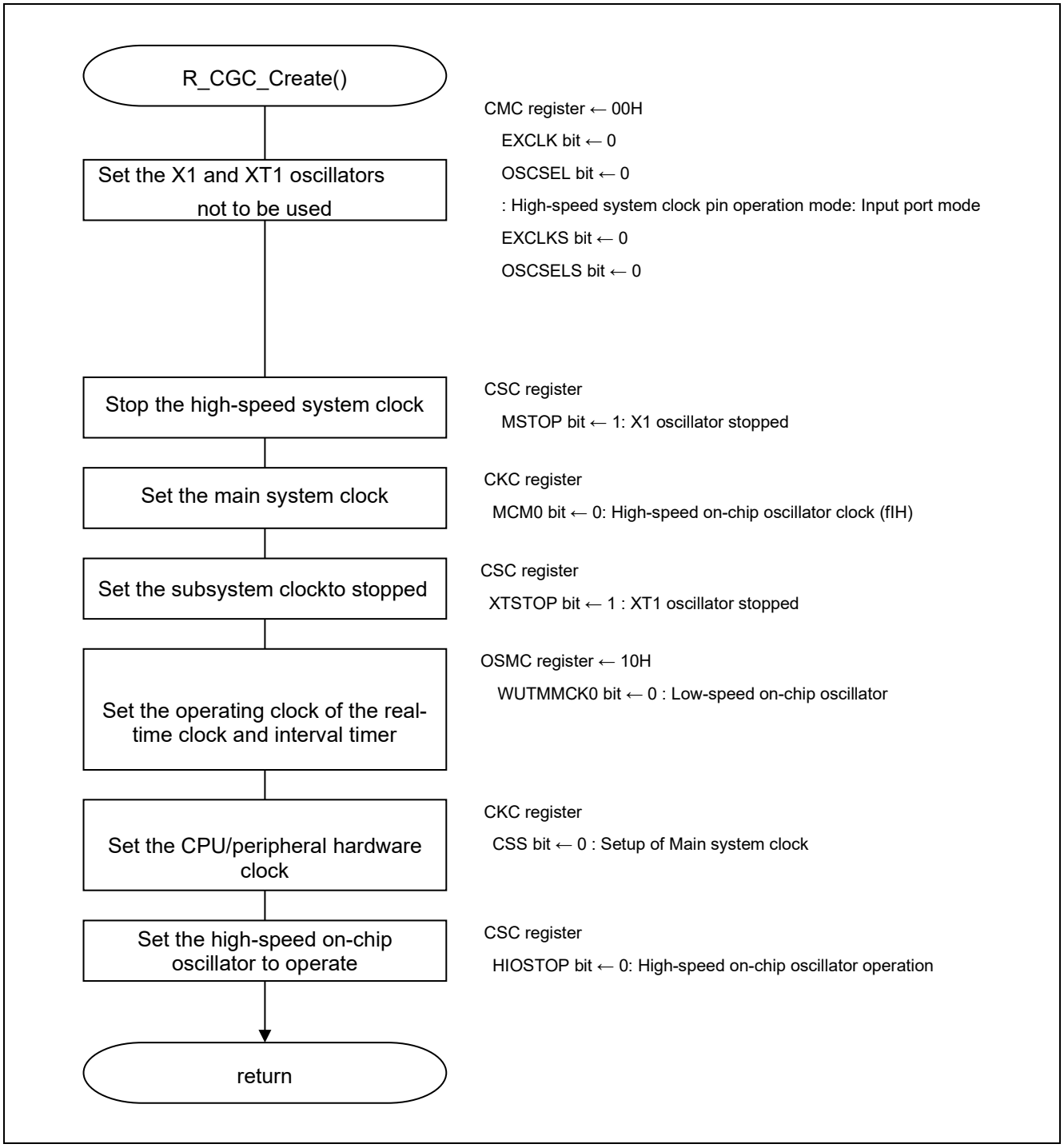


Figure 5.9 CPU Clock Initialization

5.8.5 Port Initialization

Figure 5.10 shows the port initialization.

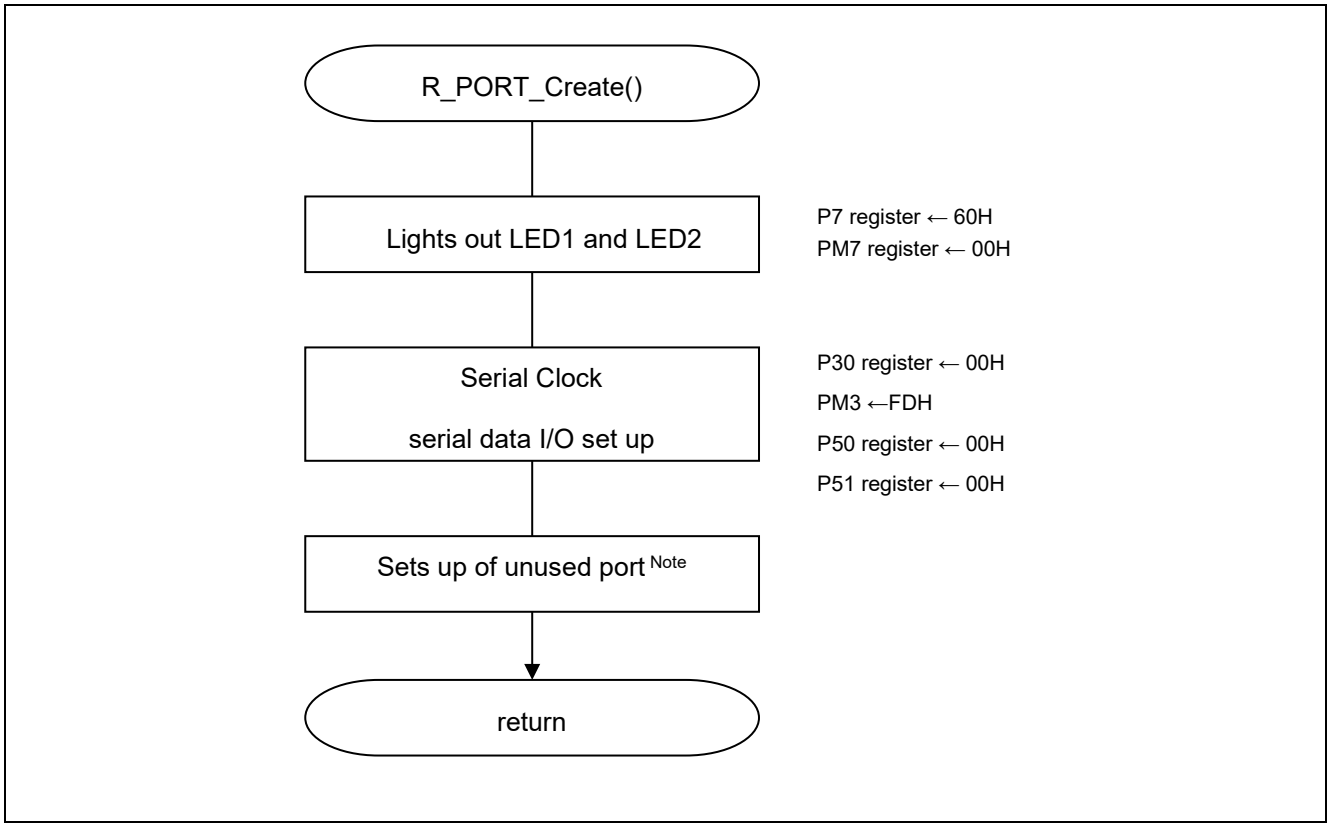


Figure 5.10 Port Initialization

Note: Refer to RL78/G13 Initialization (R01AN0451E) Application Note “Flowchart” for setup of unused ports.

Note: As for unused ports, make sure to handle unused pins appropriately to satisfy the electrical characteristics. Connect unused input-only ports independently to either VDD or VSS via resistors.

5.8.6 SAU0 Initialization

Figure 5.11 shows the SAU0 initialization.

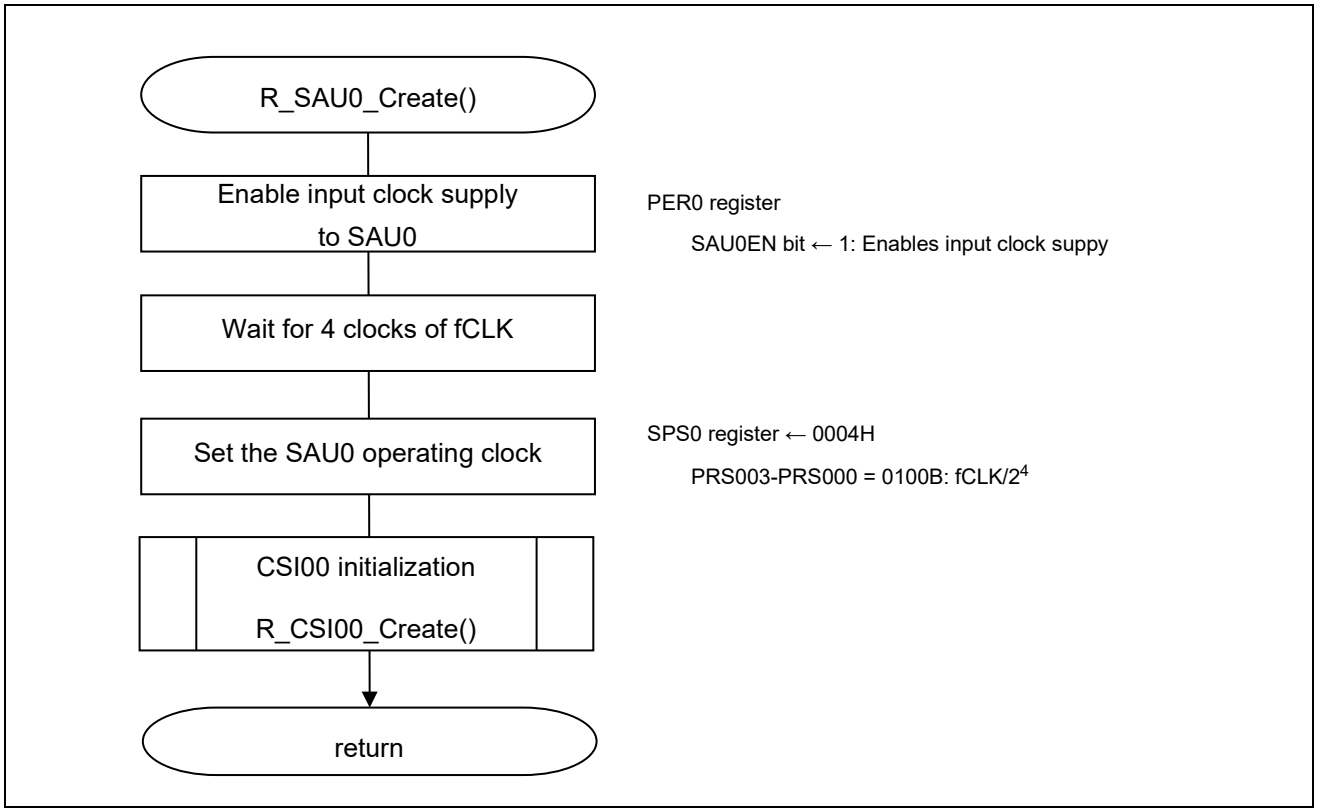


Figure 5.11 SAU0 Initialization

Enabling input clock supply to SAU0

・Peripheral enable register 0 (PER0)

Starts enabling input clock supply to SAU0.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
x	x	x	x	x	**1**	x	x

Bit 2

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. ・SFR used by the serial array unit 0 cannot be written. ・The serial array unit 0 is in the reset status.
1	**Enables input clock supply.** **・SFR used by the serial array unit 0 can be read and written.**

Setting the SAU0 operating clock

・Serial clock select register0 (SPS0)
Sets it to 2MHz.

Symbol: SPS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000
—	—	—	—	—	—	—	—	x	x	x	x	**0**	**1**	**0**	**0**

Bit 3-0

PRS 003	PRS 002	PRS 001	PRS 000		Section of operation clock (CK0)				
					f_{CLK}= 2MHz	f_{CLK}= 5MHz	f_{CLK}= 10MHz	f_{CLK}= 20MHz	f_{CLK}= 32MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	**1**	**0**	**0**	**$f_{CLK}/2^4$**	125 kHz	313 kHz	625 kHz	1.25 MHz	**2 MHz**
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.8 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.9 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

5.8.7 CSI00 Initialization

Figure 5.12 shows the CSI00 initialization.

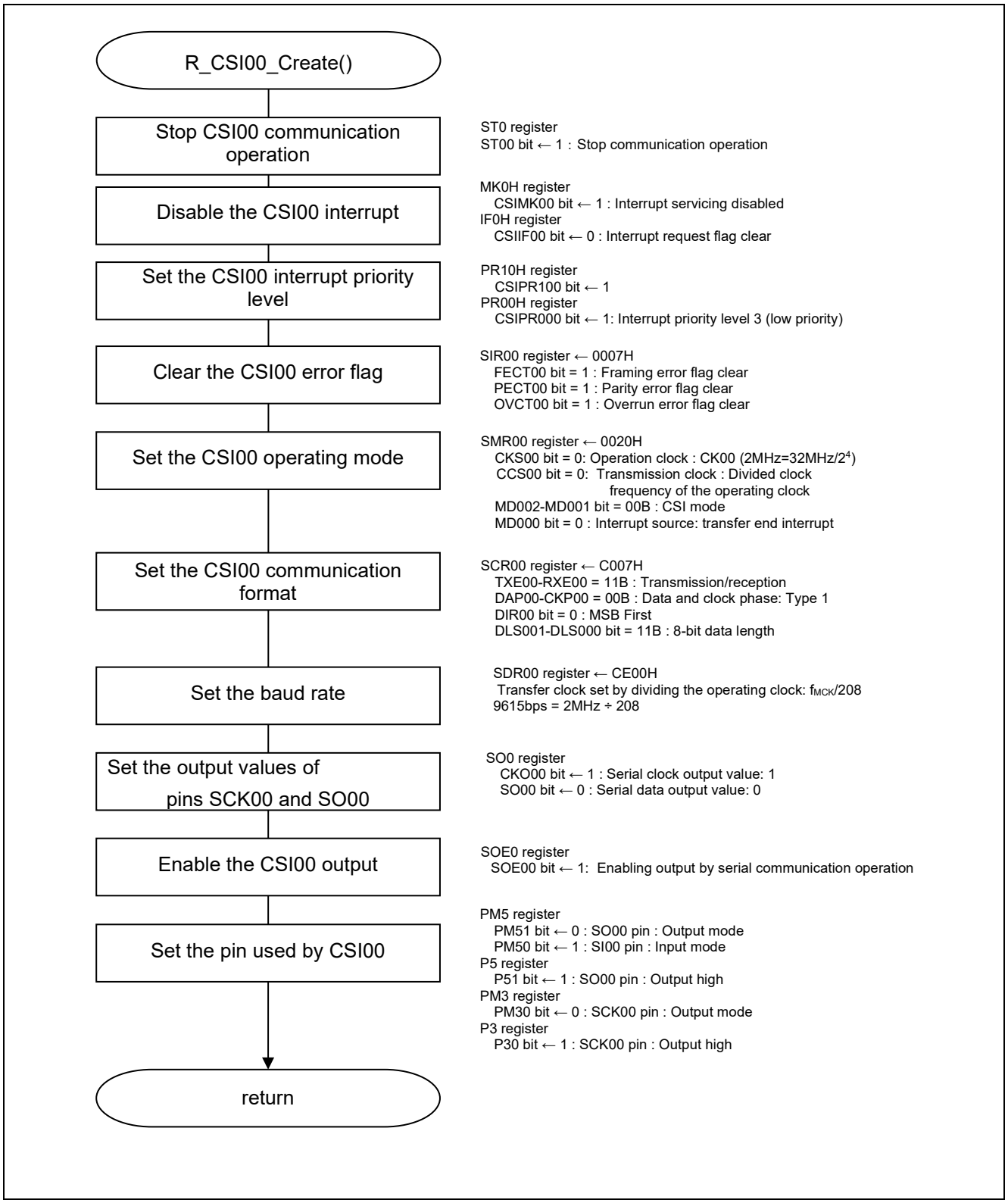


Figure 5.12 CSI00 Initialization

Stopping the CSI00 communication operation

・Serial channel stop register0(ST0)

Stops the CSI00communication operation.

Symbol: ST0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00
—	—	—	—	—	—	—	—	—	—	—	—	x	x	x	**1**

Bit 0

ST00	Serial channel stop register
0	No trigger operation
1	**Clears the SE00 bit to 0 and stops the communication operation**

Disabling the CSI00 interrupt

- Interrupt mask flag registers (MK0H)
  Interrupt servicing. disabled
- Interrupt request flag registers (IF0H)
  Interrupt request flag clear

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 **CSIMK00** IICMK00	1	1	SREMK2 TMMK11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**1**	—	—	x	x	x

Bit 5

CSIMK00	Interrupt servicing control
0	Interrupt servicing enabled
1	**Interrupt servicing disabled**

Symbol: IF0H

7	6	5	4	3	2	1	0
SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	0	0	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	—	—	x	x	x

Bit 5

CSIIF00	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Setting the CSI00 interrupt priority

- Priority Specification Flag Registers (PR10H,PR00H)
  Sets the priority level 3 (low priority).

Symbol: PR00H

7	6	5	4	3	2	1	0
SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 **CSIPR000** IICPR000	1	1	SREPR02 TMPR011H	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020
x	x	**1**	x	x	x	x	x

Symbol: PR10H

7	6	5	4	3	2	1	0
SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 **CSIPR100** IICPR100	1	1	SREPR12 TMPR111H	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120
x	x	**1**	x	x	x	x	x

Bit 5

CSIPR100	CSIPR000	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	**1**	**Specify level 3 (low priority level)**

Clearing the CSI00 error flag

- Serial flag clear trigger register 00 (SIR00)
  Sets up the clearing of an error flag.

Symbol: SIR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FEC 000	PEC 000	OVC 000
—	—	—	—	—	—	—	—	—	—	—	—	—	**1**	**1**	**1**

Bit 2

FECT00	Clear trigger of framing error of channel 0
0	Not cleared
1	**Clears the FEF00 bit of the SSR00 register to 0.**

Bit 1

PECT00	Clear trigger of parity error flag of channel 0
0	Not cleared
1	**Clears the PEF00 bit of the SSR00 register to 0.**

Bit 0

OVCT00	Clear trigger of overrun error flag of channel 0
0	Not cleared
1	**Clears the OVF00 bit of the SSR00 register to 0.**

Setting the CSI00 operating mode

- Serial mode register 00 (SMR00)
  Sets followings.
  Operating clock (f_{MCK}) : CK00
  Transfer clock (f_{TCLK}) : Divided fMCK
  Operating mode: CSI mode

Symbol: SMR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS 00	CCS 00	0	0	0	0	0	STS 00	0	SIS 000	1	0	0	MD 002	MD 001	MD 000
0	**0**	–	–	–	–	–	0	–	0	1	–	–	**0**	**0**	**0**

Bit 15

CKS00	Selection of operation clock (f_{MCK}) of channel 0
0	**Operation clock CK00 set by the SPS0 register**
1	Operation clock CK01 set by the SPS0 register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCS00 bit and the higher 7 bits of the SDR00 register, a transfer clock (f_{TCLK}) is generated.	

Bit 14

CCS00	Selection of transfer clock (f_{TCLK}) of channel 0
0	**Divided operation clock f_{MCK} specified by the CKS00 bit**
1	Clock input f_{SCK} from the SCK00 pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS00 = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDR00 register.	

Bit 2-1

MD002	MD001	Setting of operation mode of channel 0
0	**0**	**CSI mode**
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

Bit 0

MD000	Selection of interrupt source of channel 0
0	**Transfer end interrupt**
1	Buffer empty interrupt (Occurs when data is transferred from the SDR00 register to the shift register.)

Setting the CSI00 communication format

- Serial communication operation setting register 00 (SCR00)
  Sets followings.
  Operating mode: Enable transmission/reception
  Clock phase: Type 1
  Data transfer sequence: MSB first
  Data length: 8-bit data length

Symbol: SCR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE 00	RXE 00	DAP 00	CKP 00	0	EOC 00	PTC 001	PTC 000	DIR 00	0	SLC 001	SLC 000	0	1	DLS 001	DLS 000
1	**1**	**0**	**0**	—	x	x	x	**0**	—	x	x	—	—	**1**	**1**

Bit 15-14

TXE00	RXE00	Setting of operation mode of channel 0
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	**1**	**Transmission/reception**

Bit 13-12

DAP00	CKP00	Selection of data and clock phase in CSI mode
0	**0**	**Type 1**
0	1	Type 2
1	0	Type 3
1	1	Type 4

Bit 7

DIR00	Selection of data transfer sequence in CSI and UART modes
0	**Inputs/outputs data with MSB first.**
1	Inputs/outputs data with LSB first.

Bit 1-0

DLS001	DLS000	Setting of data length in CSI and UART modes
0	0	9-bit data length (stored in bits 0 to 8 of the SDR00 register) (settable in UART mode only)
0	1	7-bit data length (stored in bits 0 to 6 of the SDR00 register)
1	**1**	**8-bit data length (stored in bits 0 to 7 of the SDR00 register)**
Other than above		Setting prohibited

Setting the baud rate

- Serial data register 00 (SDR00)
  Sets the transfer clock to 9600 bps.

  (9600bps = f_{MCK} ÷ 208 = 2MHz ÷ 208)

Symbol: SDR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
1	**1**	**0**	**0**	**1**	**1**	**1**	0								

Bit 15-9

SDR00[15:9]							Transfer clock set by dividing the operating clock(f_{MCK})
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1	**1**	**0**	**0**	**1**	**1**	**1**	**f_{MCK}/208 (= f_{MCK}/{(103+1)×2})**

Setting the output values from pins SCK00 and SO00

- Serial output register 0 (SO0)
  Sets the Serial clock output value to “1”.
  Sets the Serial data output value to “0”.

Symbol: SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO03	SO02	SO01	SO00
—	—	—	—	x	x	x	**1**	—	—	—	—	x	x	x	**0**

Bit 8

CKO00	Serial clock output of channel 0
0	Serial clock output value is “0”.
1	**Serial clock output value is “1”.**

Bit 0

SO00	Serial data output of channel 0
0	**Serial data output value is “0”.**
1	Serial data output value is “1”.

Enabling the CSI00 output

- Serial output enable register 0(SOE0)
  Sets CSI00 to the enabling output.

Symbol:SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00
—	—	—	—	—	—	—	—	—	—	—	—	x	x	x	**1**

Bit 0

SOE00	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	**Enables output by serial communication operation.**

Setting the port I/O mode

- Port mode register n (PMn) (n:3/5)
  Sets PM50 to the input mode.
  Sets PM31/PM51 to the output mode.

Symbol: PM5

7	6	5	4	3	2	1	0
PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50
—	—	—	—	x	x	**0**	**1**

Bit 1

PM51	P51 pin I/O mode selection
0	**Output mode (the pin functions as an output port (output buffer on))**
1	Input mode (the pin functions as an input port (output buffer off))

Bit 0

PM50	P50 pin I/O mode selection
0	Output mode (the pin functions as an output port (output buffer on))
1	**Input mode (the pin functions as an input port (output buffer off))**

Symbol: PM3

7	6	5	4	3	2	1	0
1	1	1	1	1	1	PM31	PM30
—	—	—	—	x	x	x	**0**

Bit 0

PM30	P30 pin I/O mode selection
0	**Output mode (the pin functions as an output port (output buffer on))**
1	Input mode (the pin functions as an input port (output buffer off))

Setting the port output value

- Port register n(Pn) (n:3/5)
  Sets P51/P30 to the high output.

Symbol: P5

7	6	5	4	3	2	1	0
P57	P56	P55	P54	P53	P52	P51	P50
—	—	—	—	x	x	**1**	x

Bit 1

P51	Output data control (in output mode)
0	Output 0
1	**Output 1**

Symbol: P3

7	6	5	4	3	2	1	0
0	0	0	0	0	0	P31	P30
—	—	—	—	x	x	x	**1**

Bit 0

P30	Output data control (in output mode)
0	Output 0
1	**Output 1**

5.8.8 DTC Initialization

Figure 5.13 shows the DTC initialization.

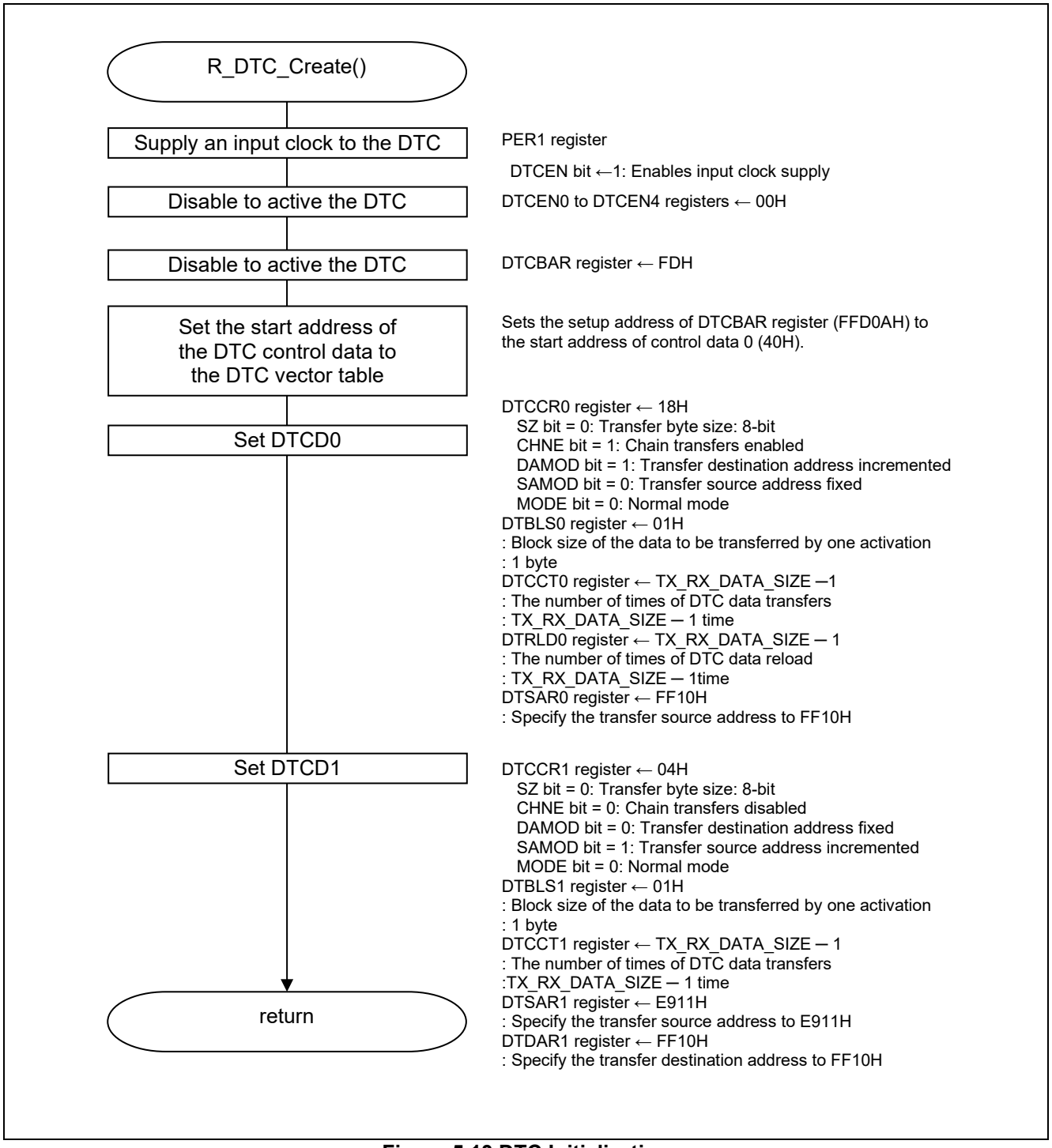


Figure 5.13 DTC Initialization

Supplying an input clock to the DTC

・Peripheral enable register 1（PER1）
Starts supplying an input clock to the DTC

Symbol: PER1

7	6	5	4	3	2	1	0
DACEN	TRGEN	CMPEN	TRD0EN	DTCEN	0	0	TAU0EN
x	x	x	x	**1**	x	x	x

Bit 3

DTCEN	Control of DTC input clock supply
0	Stops input clock supply.
1	**Enables input clock supply.**

Disabling to activate DTC0

・DTC activation enable register i (DTCENi）(i=0 to 4)
Disables DTC activation.

Symbol: DTCENi

7	6	5	4	3	2	1	0
DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Bit 7 to 0 (The example of the bit 7 indicates the following. (From the bit 7 to the bits 0 are the same contents.))

DTCENi7	DTC activation enable i7
0	**Activation disabled**
1	Activation enabled

DTC base address

・DTC base address register（DTCBAR）
Sets “FDH” to the DTC base address.

Symbol: DTCBAR

7	6	5	4	3	2	1	0
DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0
1	**1**	**1**	**1**	**1**	**1**	**0**	**1**

Setting the DTC control register

・DTC control register 0（DTCCR0）
Sets it as 8 bits, chain transfer enabled, and normal mode.

Symbol: DTCCR0

7	6	5	4	3	2	1	0
0	SZ	RPTINT	CHNE	DAMOD	SAMOD	PRTSEL	MODE
-	**0**	**0**	**1**	**1**	**0**	**0**	**0**

Bit 6

SZ	Transfer Data size selection
0	**8 bits**
1	16 bits

Bit 5

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

Bit 4

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	**Chain transfers enabled**
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

Bit 3

DAMOD	Transfer destination address control
0	Fixed
1	**Incremented**
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

Bit 2

SAMOD	Transfer source address control
0	**Fixed**
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

Bit 1

PRTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

Bit 0

MODE	Transfer mode selection
0	**Normal mode**
1	Repeat mode

Setting the DTC control register 1

- DTC control register 1 (DTCCR1)
  Sets it as 8 bits, chain transfer disabled, and normal mode.

Symbol: DTCCR1

7	6	5	4	3	2	1	0
0	SZ	RPTINT	CHNE	DAMOD	SAMOD	PRTSEL	MODE
-	0	0	0	0	1	0	0

Bit 6

SZ	Transfer Data size selection
0	**8 bits**
1	16 bits

Bit 5

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

Bit 4

CHNE	Enabling/disabling chain transfers
0	**Chain transfers disabled**
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

Bit 3

DAMOD	Transfer destination address control
0	**Fixed**
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

Bit 2

SAMOD	Transfer source address control
0	Fixed
1	**Incremented**
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

Bit 1

PRTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

Setting the DTC block size register

- DTC block size register（DTBLSi）(i=0 to 1)
  Sets the DTC0 block size to 1 byte.
  Disables DTC activation.

Symbol: DTBLSi

7	6	5	4	3	2	1	0
DTBLSi7	DTBLSi6	DTBLSi5	DTBLSi4	DTBLSi3	DTBLSi2	DTBLSi1	DTBLSi0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**

DTBLSi	Transfer Block Size	
	8-Bit Transfer	16-Bit Transfer
00H	256 bytes	512 bytes
01H	**1 byte**	2 bytes
02H	2 bytes	4 bytes
. . .	. . .	. . .
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Setting the DTC transfer count register

- DTC transfer count register I (DTCCTi) (i=0-1)
  Sets the number of times of transfers to n (1 to 15).
  Disables DTC activation.

Symbol: DTCCTi

7	6	5	4	3	2	1	0
DTCCTi7	DTCCTi6	DTCCTi5	DTCCTi4	DTCCTi3	DTCCTi2	DTCCTi1	DTCCTi0
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**

DTCCTi	Number of Transfers
00H	256 times
01H	Once
02H	2 times
03H	**3 times**
. . .	. . .
FEH	254 times
FFH	255 times

In this sample code, the setting value of a DTCCTi register changes according to the values of constant TX_RX_DATA_SIZE. Here, the setting value in the case of TX_RX_DATA_SIZE = 4 is indicated.

Setting the DTC transfer count reload register 0

- DTC transfer count reload register 0 (DTRLD0)
  Sets the number of times of reload to n (1 to 15).

Symbol: DTRLD0

7	6	5	4	3	2	1	0
DTRLD07	DTRLD06	DTRLD05	DTRLD04	DTRLD03	DTRLD02	DTRLD01	DTRLD00
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**

In this sample code, the setting value of a DTCCTi register changes according to the values of constant TX_RX_DATA_SIZE.

Here, the setting value in the case of TX_RX_DATA_SIZE = 4 is indicated.

Setting the DTC source address register 0

- DTC source address register 0 (DTSAR0)
  Specifies the transfer source address for data transfer to “FF10H”.

Symbol: DTSAR0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSAR015	DTSAR014	DTSAR013	DTSAR012	DTSAR011	DTSAR010	DTSAR09	DTSAR08	DTSAR07	DTSAR06	DTSAR05	DTSAR04	DTSAR03	DTSAR02	DTSAR01	DTSAR00
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

Setting the DTC source address register 1

- DTC source address register 1 (DTSAR1)
  Specifies the transfer source address for data transfer to “E911H”.

Symbol: DTSAR1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSAR115	DTSAR114	DTSAR113	DTSAR112	DTSAR111	DTSAR110	DTSAR19	DTSAR18	DTSAR17	DTSAR16	DTSAR15	DTSAR14	DTSAR13	DTSAR12	DTSAR11	DTSAR10
1	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**

Setting the DTC destination address register 0

- DTC destination address register 0 (DTDAR0) (j=0 to 23)
  Specify the transfer destination address for data transfer to “E900H”.

Symbol: DTDAR0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDAR015	DTDAR014	DTDAR013	DTDAR012	DTDAR011	DTDAR010	DTDAR09	DTDAR08	DTDAR07	DTDAR06	DTDAR05	DTDAR04	DTDAR03	DTDAR02	DTDAR01	DTDAR00
1	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Setting the DTC destination address register 1

- DTC destination address register 1 (DTDAR1) (j=0 to 23)
  Specify the transfer destination address for data transfer to “FF10H”.

Symbol: DTDAR1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDAR115	DTDAR114	DTDAR113	DTDAR112	DTDAR111	DTDAR110	DTDAR19	DTDAR18	DTDAR17	DTDAR16	DTDAR15	DTDAR14	DTDAR13	DTDAR12	DTDAR11	DTDAR10
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**

5.8.9 Main Processing

Figure 5.14 shows the main processing.

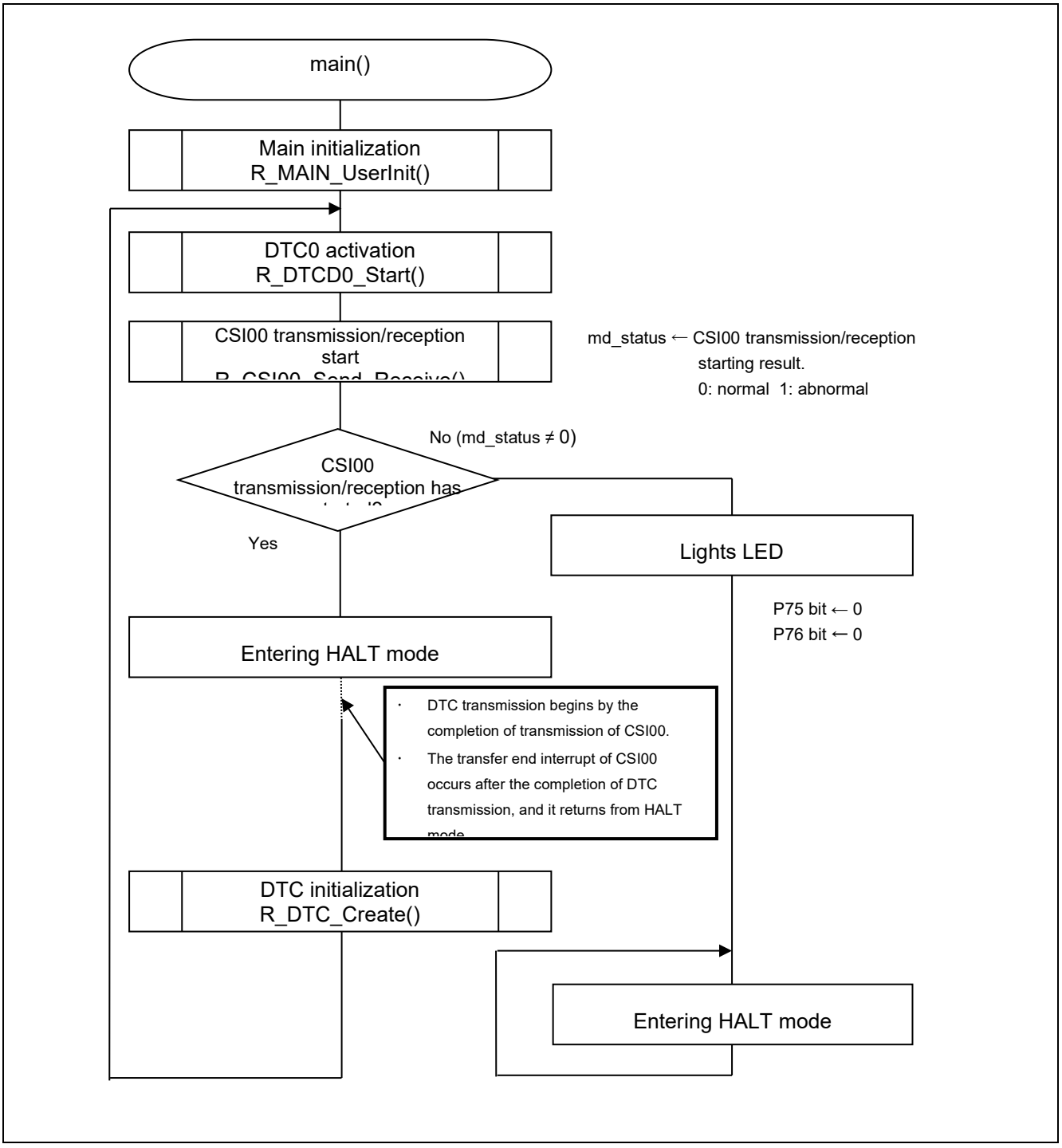


Figure 5.14 Main Processing

5.8.10 Main Initialization

Figure 5.15 shows the main initialization.

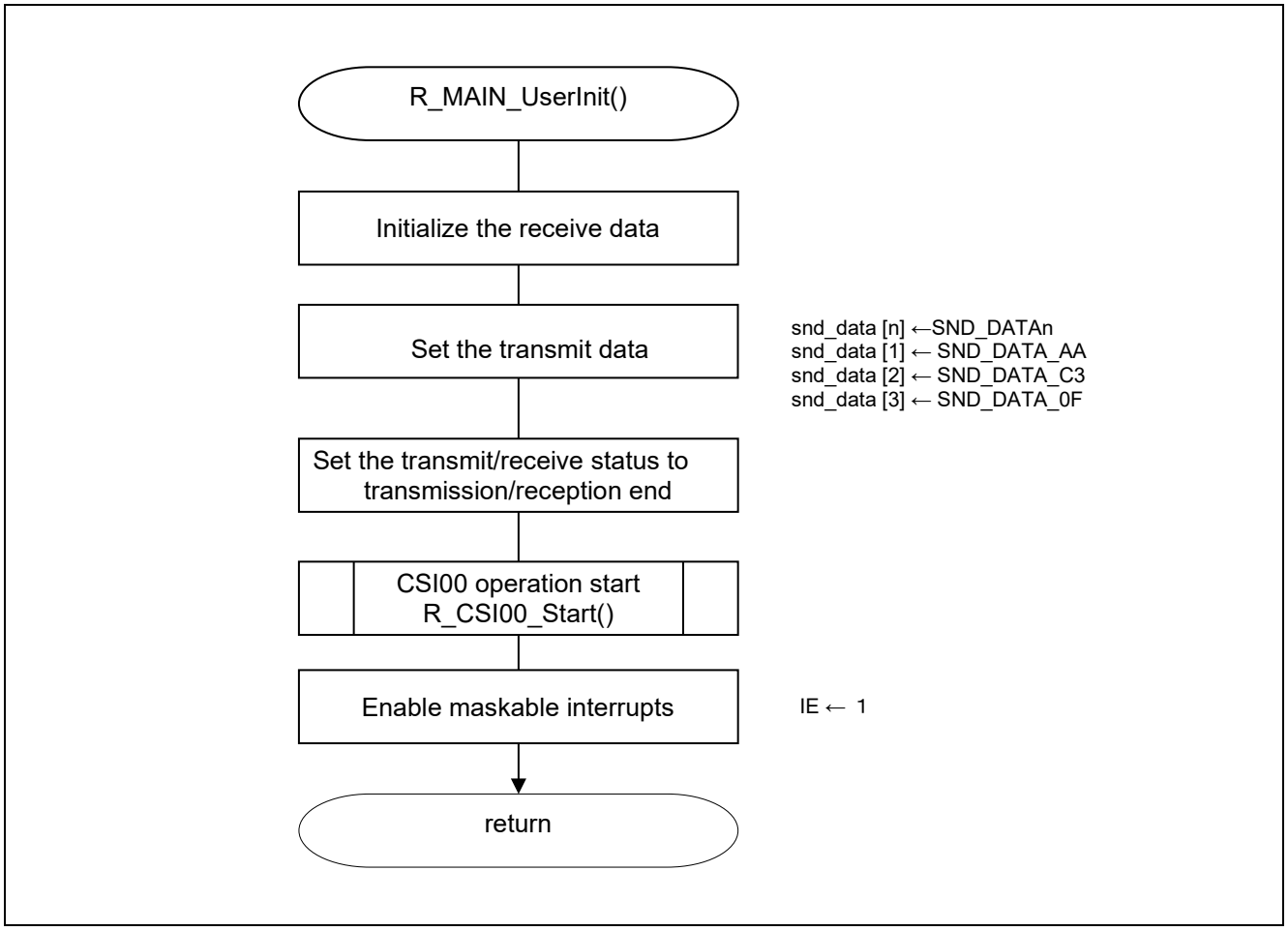


Figure 5.15 Main Initialization

5.8.11 CSI00 Operation Start

Figure 5.16 shows the CSI00 operation start.

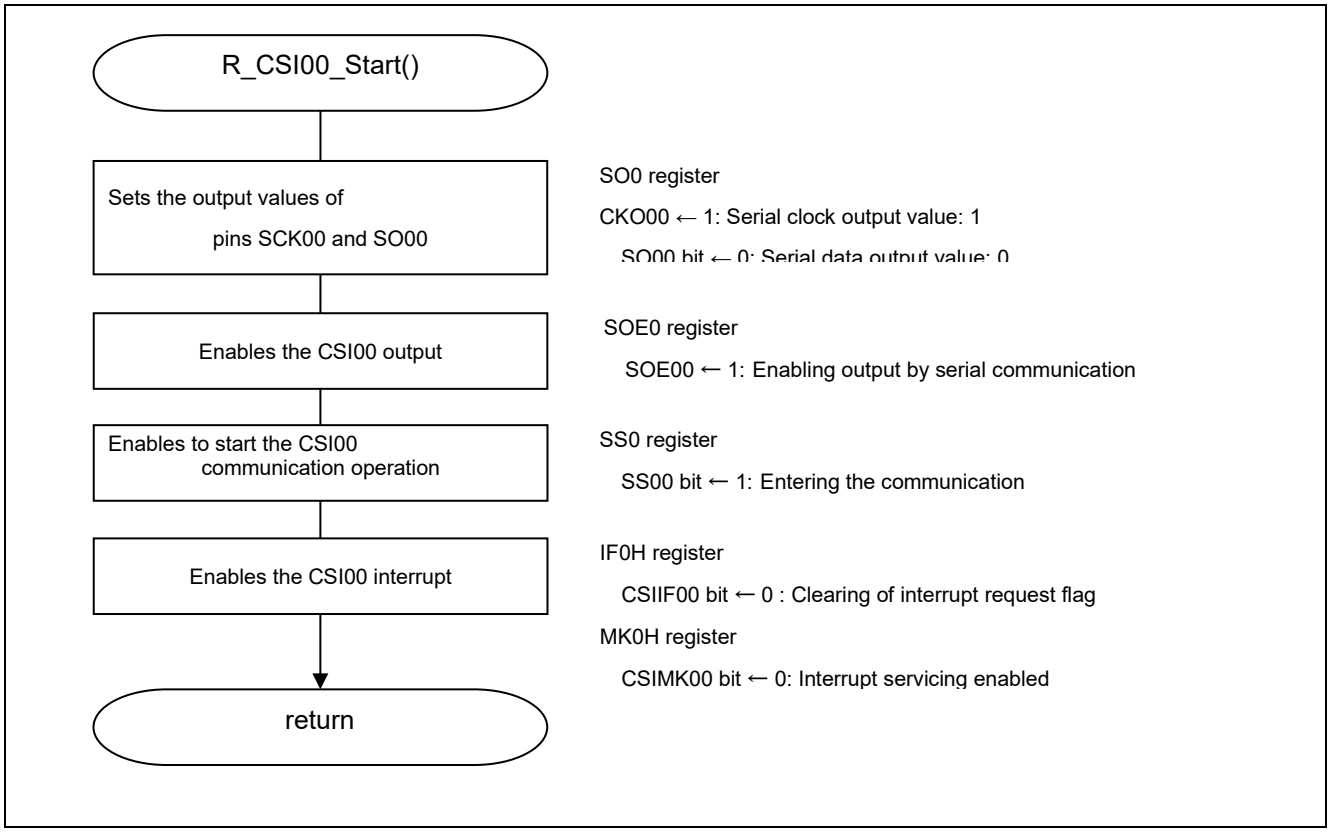


Figure 5.16 CSI00 Operation Start

Setting the output values from pins SCK00 and SO00

・Serial output register 0 (SO0)
Sets the serial clock output value to 1.
Sets the serial data output value to 0.

Symbol: SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00
—	—	—	—	x	x	x	**1**	—	—	—	—	x	x	x	**0**

Bit 8

CKO00	Serial clock output of channel 0
0	Serial clock output value is “0”.
1	**Serial clock output value is “1”.**

Bit 0

SO00	Serial data output of channel 0
0	**Serial data output value is “0”.**
1	Serial data output value is “1”.

Enabling the CSI00 output

・Serial output enable register 0 (SOE0)
Sets CSI00 as output enable.

Symbol: SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00
—	—	—	—	—	—	—	—	—	—	—	—	x	x	x	**1**

Bit 0

SOE00	Serial output enable/stop of channel 0
0	Stops output by serial communication operation.
1	**Enables output by serial communication operation.**

Enabling to start the CSI00 communication operation

・Serial channel start register 0 (SS0)
Sets it the communication wait status.

Symbol: SS0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00
—	—	—	—	—	—	—	—	—	—	—	—	x	x	x	**1**

Bit 0

SS00	Operation start trigger of channel 0
0	No trigger operation
1	**Sets the SE00 bit to 1 and enters the communication wait status.**

Enabling the CSI00 interrupt

- Interrupt Request Flag Registers (IF0H)
  Clears the interrupt request flag.
- Interrupt Mask Flag Registers (MK0H)
  Disables the interrupt servicing.

Symbol: IF0H

7	6	5	4	3	2	1	0
SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 **CSIIF00** IICIF00	0	0	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	—	—	x	x	x

Bit 5

CSIIF00	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request is generated, interrupt request status

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 **CSIMK00** IICMK00	1	1	SREMK2 TMMK11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	—	—	x	x	x

Bit 5

CSIMK00	Interrupt servicing control
0	**Interrupt servicing enabled**
1	Interrupt servicing disabled

5.8.12 DTC0 Activation

Figure 5.17 shows the DTC0 activation.

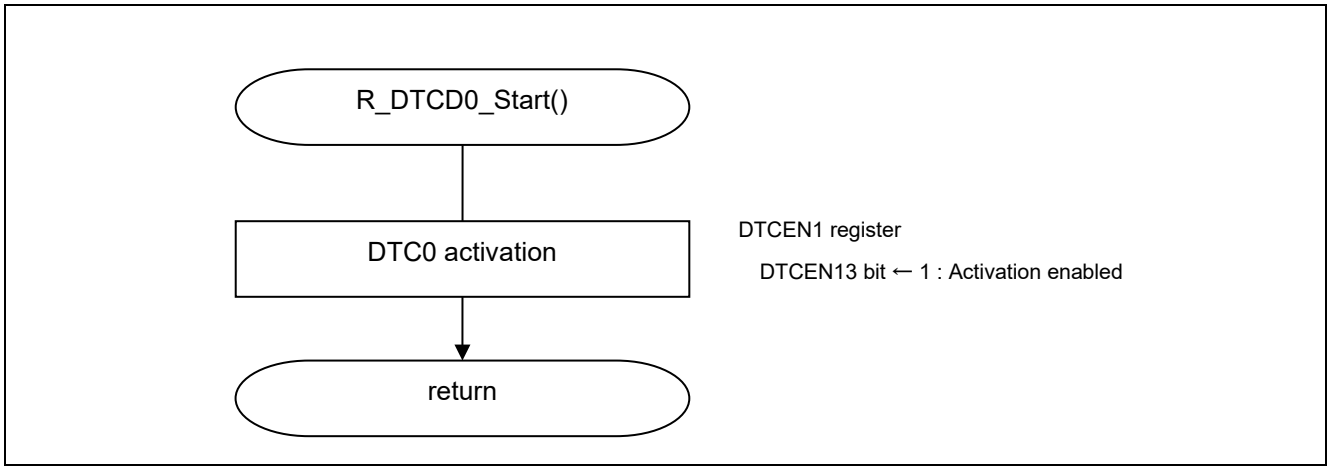


Figure 5.17 DTC0Activation

DTC0 activation

・DTC activation enable register 1 (DTCEN1）
Enables DTC activation.

Symbol: DTCEN1

7	6	5	4	3	2	1	0
DTCEN17	DTCEN16	DTCEN15	DTCEN14	DTCEN13	DTCEN12	DTCEN11	DTCEN10
0	**0**	**0**	**0**	**1**	**0**	**0**	**0**

Bit 3

DTCEN13	DTC activation enable 13
0	Activation disabled
1	**Activation enabled**

5.8.13 CSI00 Transmission/Reception Start

Figure 5.18 shows the CSI00 transmission/reception start.

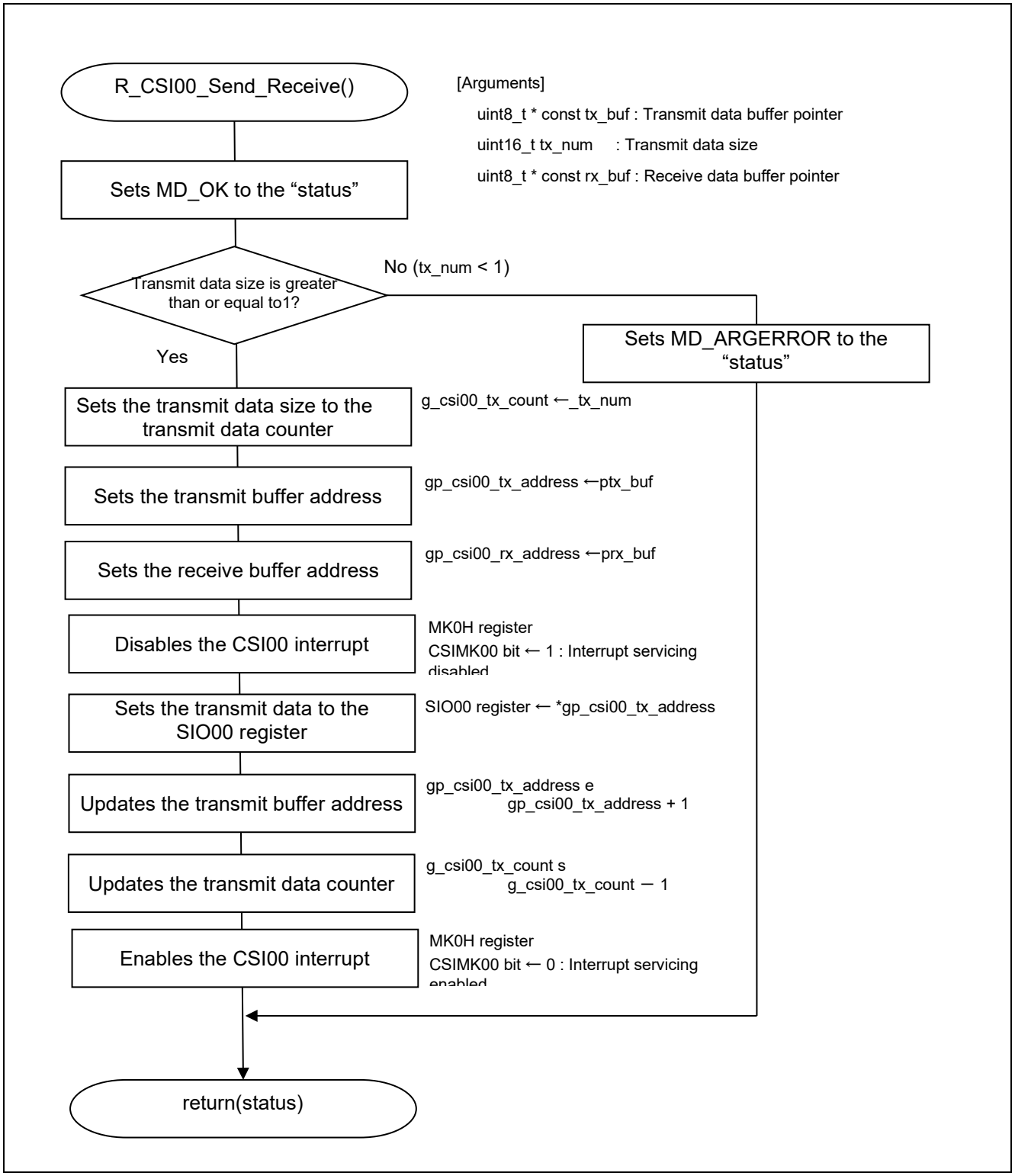


Figure 5.18 CSI00 Transmission/Reception Start

Enabling the CSI00 interrupt

- Interrupt mask flag registers (MK0H)
  Interrupt servicing enabled

Symbol: MK0H

7	6	5	4	3	2	1	0
SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 **CSIMK00** IICMK00	1	1	SREMK2 TMMK11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20
x	x	**0**	—	—	x	x	x

Bit 5

CSIMK00	Interrupt servicing control
0	**Interrupt servicing enabled**
1	Interrupt servicing disabled

Setting the transmit data

- CSI00 data register (SIO00)
  Writes the transmit data

Symbol: SIO00

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—
00H～FFH							

5.8.14 CSI00 Transfer End Interrupt

Figure 5.19 shows the CSI00 transfer end interrupt.

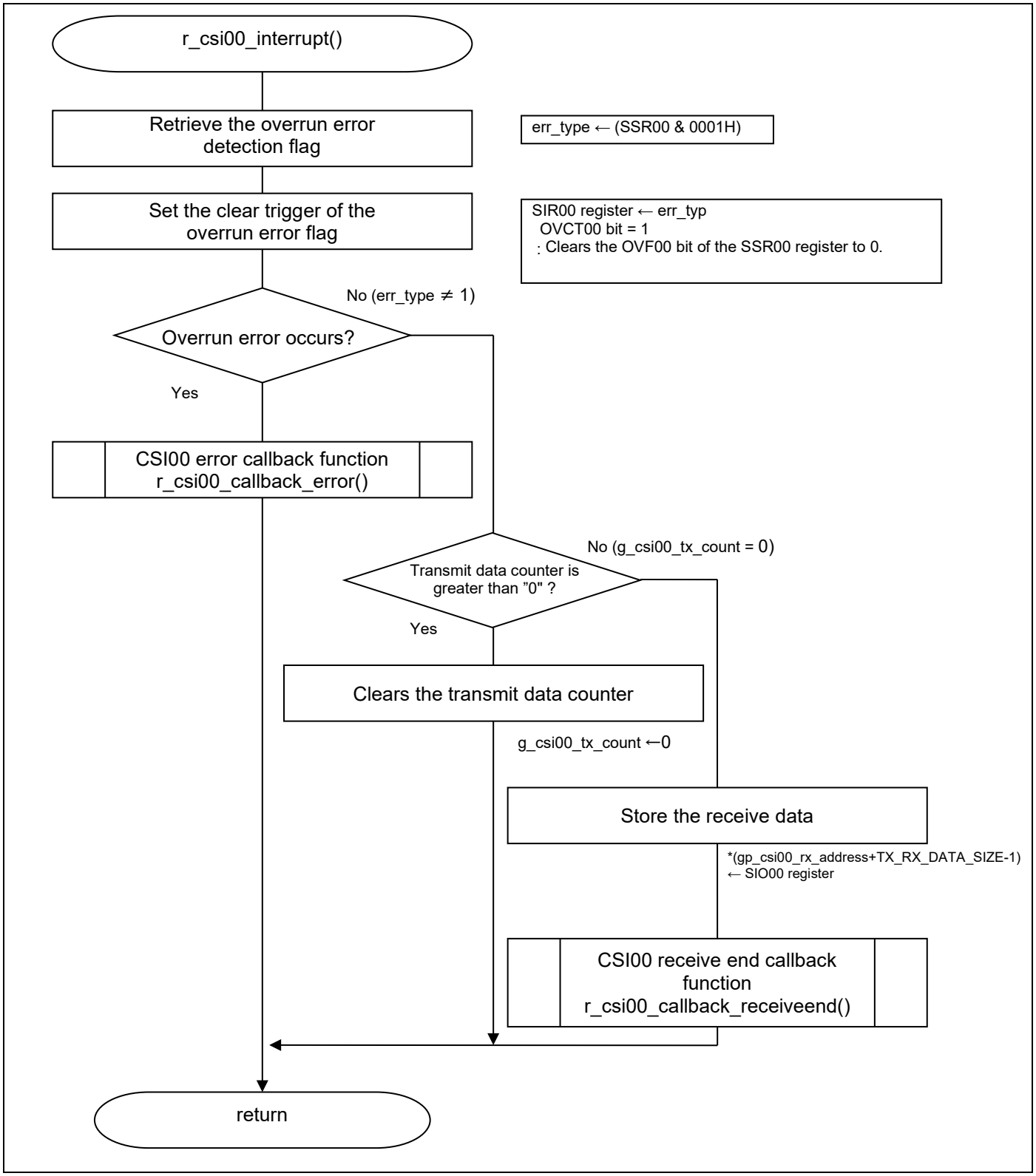


Figure 5.19 CSI00 Transfer End Interrupt

Retrieving the overrun error detection flag status

- Serial status register 00 (SSR00)

Symbol: SSR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TSF 00	BFF 00	0	0	FEF 00	PEF 00	OVF 00

Bit 0

OVF00	Overrun error detection flag of channel 00
0	No error occurs.
1	An error occurs.

Setting the clear trigger of the overrun error flag

- Serial flag clear trigger register 00 (SIR00)
  Clears an overrun error flag when an overrun error occurs.

Symbol: SIR00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FEC T00	PEC T00	OVC T00
—	—	—	—	—	—	—	—	—	—	—	—	—	x	x	**1**

Bit 0

OVCT00	Clear trigger of overrun error flag of channel 00
0	Not cleared
1	**Clears the OVF00 bit of the SSR00 register to 0.**

Storing the receive data

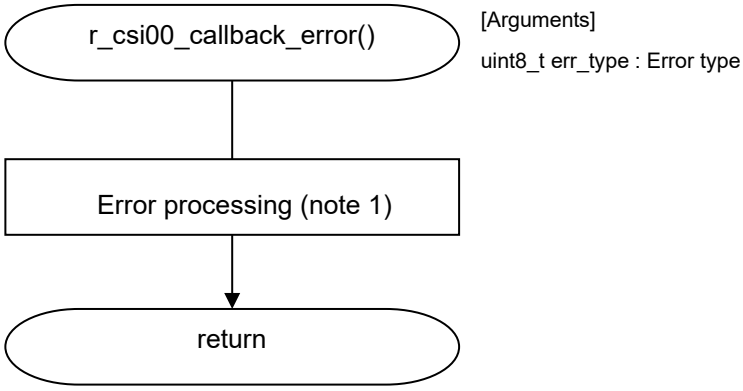
- CSI00 data register (SIO00)
  Reads the receive data.

Symbol: SIO00

7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—

5.8.15 CSI00 Error Callback Function

Figure 5.20 shows the CSI00 error callback function.



Note1. The sample code does not include the error processing. Add processing to the user program as needed.

Figure 5.20 CSI00 Error Callback Function

5.8.16 CSI00 Receive End Callback Function

Figure 5.21 shows the CSI00 receive end callback function.

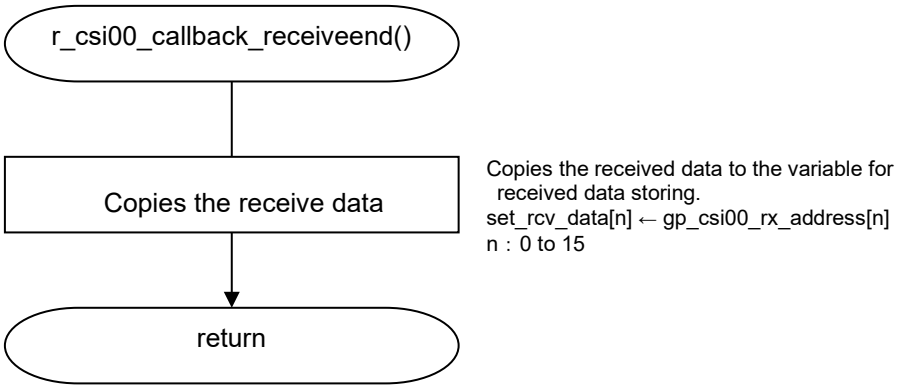


Figure 5.21 CSI00 Receive End Callback Function

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G14 User's Manual: Hardware Rev.3.20 (R01UH0146EJ)

RL78 Family User's Manual: Software Rev.2.20 (R01US0015EJ)

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.

Revision History <RL78/G14 Using the DTC Chain Transfer to Perform 3-Wire Serial I/O (Master Transmission/Reception) by Serial Array Unit in HALT Mode**>**

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.22, 2014	**—**	First edition issued
1.01	Aug.24, 2015	8	Error correction of DTCD1 transfer source address and destination address
1.10	Sep.12. 2022	4	Updated operation check conditions

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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

Notice

1. 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.