RX610 Group

Using the RPDL for Serial EEPROM Control

R01AN0897EJ0100
Rev.1.00
Mar 28, 2012

Introduction

This application note describes serial EEPROM control using the serial communication interface API (SCI clock-synchronous single-master control) of the RX610 Group Renesas Peripheral Driver Library and the Renesas R1EX25xxx Series Serial EEPROM Control Software.

The sample application makes use of the following application note and library:

- Serial EEPROM control: Renesas R1EX25xxx Series Serial EEPROM Control Software Rev.1.03 (R01AN0565EJ0103)
- SCI clock-synchronous single-master control: Renesas Peripheral Driver Library (R20UT0083EE0100)

Target Device

RX610 Group

When applying this application note to MCUs other than the target device, make changes as necessary to match the MCU to be used and evaluate operation carefully.

Contents

1. Specifications

The sample code described in this application note operates with an RX610-RSK MCU connected to serial EEPROM (Renesas Electronics R1EX25xxx/HN58X25xxx Series) via a serial peripheral interface as shown in figure 1.1, Usage Example.

The sample application uses the SCI clock-synchronous single-master control functionality of the Renesas Peripheral Driver Library (RPDL) and the Renesas R1EX25xxx Series Serial EEPROM Control Software (serial EEPROM control middleware) to performs operations such as programming and reading the external serial EEPROM connected to the RX610-RSK.

The communication functions can be summarized as follows:

- The sample application uses the clock-synchronous (three-wire) serial communication function (with the channel fixed as SCI1) of the RX610.
- Serial EEPROM chip select is controlled by using a general port (P66).
- The clock-synchronous serial communication transfer bit rate is set to 3,125,000 bps.
- The clock-synchronous transmit/receive mode of the SCI is used for programming and reading the serial EEPROM, and the RXI interrupt of the SCI is enabled only during the read processing.

Note that the sample code described in this application note supports operation in little-endian mode only.

Table 1.1 Peripheral Functions and Their Uses

Peripheral Function	Use
SCI (serial communication interface)	Clock-synchronous (three-wire) serial communication

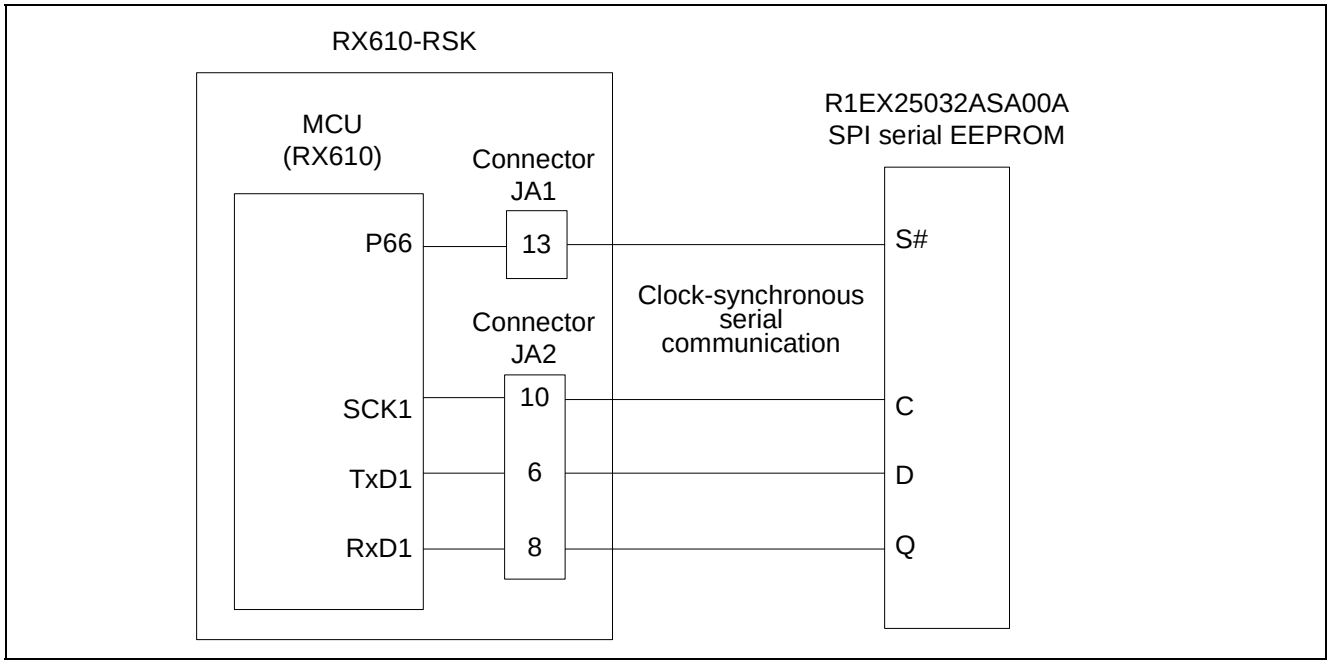


Figure 1.1 Usage Example

2. Operation Confirmation Conditions

The sample code described in this application note has been confirmed to run properly under the operating conditions listed below.

Table 2.1 Operation Confirmation Conditions

Item	Description
MCU	RX610 Group (R5F56108VNFP)
Device	Renesas Electronics R1EX25032ASA00A SPI serial EEPROM
Operating frequencies	• EXTAL: 12.5 MHz • ICLK: 100 MHz • PCLK: 50 MHz • BCLK: 25 MHz
SCI operating conditions*1	• Clock-synchronous (three-wire) serial communication • Channel used: Channel 1 • Bit rate: 3,125,000 bps • MSB-first transfer
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance embedded Workshop Version 4.09.00.007
C compiler	Renesas Electronics RX Standard Toolchain Version 1.1.0.0
	Compiler options: -cpu=rx600 -patch=rx610 -include="$(PROJDIR)\RPDL","$(HEWDIR)\." -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo
Endian mode	Little endian
Version of the sample code	Ver.1.00
Software used for evaluation	Renesas Electronics The R1EX25xxx Series' SPI serial EEPROM control software, Ver. 2.00
Board used	Renesas Starter Kit for RX610

Note: 1. The setting values used are the default values of the serial EEPROM control middleware.

3. Related Application Note and Library

The applications notes that are related to this application note are listed below. Reference should also be made to those application notes.

- Renesas R1EX25xxx Series Serial EEPROM Control Software Rev.1.03 (R01AN0565EJ)
- RX610 Group Peripheral Driver Library User's Manual (R20UT0083EE0100)

4. Hardware

4.1 List of Pins Used

Table 4.1 lists the pins used and their functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Description
SCK1	Output	Clock output
TxD1	Output	Master data output
RxD1	Input	Master data input
P66	Output	Slave device select output

4.2 Reference Circuit

Figure 4.1 shows an example wiring diagram of connections between the SCI1 of the RX610 and the R1EX25032ASA00A SPI serial EEPROM.

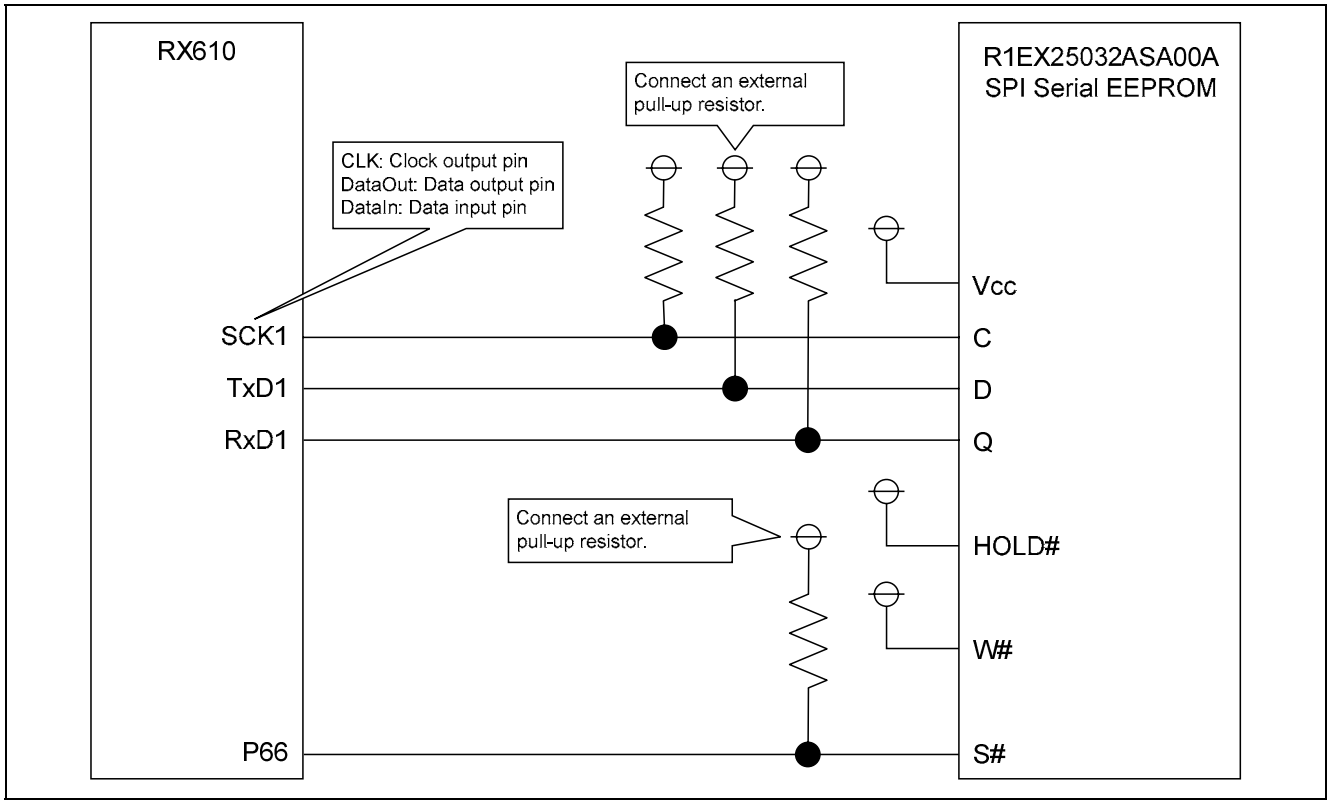


Figure 4.1 Wiring Diagram of RX610 SCI1 and R1EX25032ASA00A SPI Serial EEPROM

5. Software

This application note mainly describes the interaction between the RPDL and the serial EEPROM control middleware.

5.1 Operation Overview

The clock-synchronous (three-wire) serial communication function of the SCI is used to perform serial EEPROM control via the serial peripheral interface.

The sample application performs the following types of control:

- It controls data input and output in clock-synchronous mode (using the internal clock of the MCU).
- It controls clock-synchronous (three-wire) serial communication by the SCI by using the serial communication interface API of the RPDL.
- It controls the serial EEPROM by using the serial EEPROM control middleware.

The clock-synchronous single-master control software is the standard lower layer used with the serial EEPROM control middleware, but the sample application instead uses the serial communication interface API of the RPDL as the lower layer.

Interaction between the serial EEPROM control middleware and the RPDL is implemented by means of calls to RPDL functions by the serial EEPROM control middleware. For details, see table 5.10, Function Correspondence Table.

5.2 Required Memory Sizes

Table 5.1 lists the sizes of the different types of memory required by the sample application.

Table 5.1 Required Memory Sizes

Memory Used	Size	Remarks
ROM	3,184 bytes	Total ROM size
RAM	1,352 bytes	Total RAM size
Maximum user stack size	192 bytes	R_SPI_EEP_Wait_WBusy()
Maximum interrupt stack size	36 bytes	Interrupt_SCI1_RXI1 interrupt

Note: The required memory sizes differ depending on the version of the C compiler and the compiler options used.

5.3 File Structure

Table 5.2 lists the file structure of the sample application. Note that files that are generated automatically by the integrated development environment and files supplied with the RPDL are not listed.

For detailed instructions covering how to create and use files supplied with the RPDL, see “Using the library within your project” in the RX610 Group Peripheral Driver Library User’s Manual.

Table 5.2 File Structure

File Name	Overview	Remarks
testmain.c	Serial EEPROM control middleware module testing program	Makes use of the sample programs for verifying operation supplied with the serial EEPROM control middleware.
R_SIO_sci_rx.c	RPDL control module	Matches the names of the functions called by the serial EEPROM control middleware to perform clock-synchronous serial communication with the SCI with the names of functions used to control the RPDL.
R_SIO.h	Include header file for RPDL control module	Contains constant definitions and module prototype declarations for R_SIO_sci_rx.c.
mtl_com.c	Debugging modules using serial EEPROM control middleware	Acquires and clears error log data used for debugging the serial EEPROM control middleware.
mtl_com.h	Setting selection header file for serial EEPROM control middleware	Contains setting selection definitions and prototype declarations for the serial EEPROM control middleware.
mtl_tim.c	Loop timer related module of serial EEPROM control middleware	
mtl_tim.h	Include header file for loop timer related module of serial EEPROM control middleware	
R_SPI_EEP.h*[1]	Header file for serial EEPROM control middleware	For details, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.
R_SPI_EEP_io.c*[2]	I/O module of serial EEPROM control middleware	For details, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.
R_SPI_EEP_io.h	Definition header file for I/O module of serial EEPROM control middleware	For details, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.
R_SPI_EEP_sfr.h*[3]	Register definition header file for serial EEPROM control middleware	For details, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.
R_SPI_EEP_usr.c	User interface module of serial EEPROM control middleware	For details, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

Notes: 1. In R_SPI_EEP.h, the line "#include <stdint.h>" has been added.

2. In R_SPI_EEP_io.c, the line "#include "iodefine.h" has been added. In addition, the function "R_SPI_EEP_Write_Di" has been commented out.
3. In the sample application described in this application note, the header file R_SPI_EEP_sfr.h.rx610, which is supplied with the application note Renesas R1EX25xxx Series Serial EEPROM Control Software, is renamed to R_SPI_EEP_sfr.h, as specified in that application note.

   In addition, the value of EEP_BR, which is defined in R_SPI_EEP_sfr.h, is changed as follows:

   Original value: #define EEP_BR (uint8_t)0x03 /* BRR initial setting */
   New value: #define EEP_BR (uint32_t)3125000 /* bit rate setting */

   The specifications of the serial EEPROM control middleware stipulate that the setting value written to the bit rate register (BRR) is defined by EEP_BR, but the sample application conforms to the RPDL specifications, in which the bit rate value is defined by EEP_BR.

5.4 List of Constants

5.4.1 Return Value

Table 5.3 shows the return value used by the sample application.

Table 5.3 Return Value

Constant	Setting Value	Description
SIO_OK	(error_t)(0)	• Used by function: R_SIO_Disable R_SIO_Enable R_SIO_Tx_Data R_SIO_Rx_Data • Only SIO_OK is necessary for the combined operation of the serial EEPROM control middleware and the created functions.

5.4.2 Definitions

Table 5.4 lists the constant value for the test program supplied with the serial EEPROM control middleware. For information on the values defined in the serial EEPROM control middleware, refer to the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

Table 5.4 Constant Value for Test Program Supplied with Serial EEPROM Control Middleware

Constant	Setting Value	Description
EEP_BUF_SIZE	128	Size of buffers Cbuf1 and Cbuf2

5.5 List of Variables

Table 5.5 lists the global variables and table 5.6 the static variables.

For information on the variables used by the serial EEPROM control middleware, refer to the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

Table 5.5 Global Variables

Type	Variable Name	Description	Used by Function(s)
uchar FAR	Cbuf1	Storage buffer for data to be written to the serial EEPROM	Test200, Test300
uchar FAR	Cbuf2	Storage buffer for data to be read to the serial EEPROM	Test300
uchar	gDevNo	Device number of serial EEPROM*[1]	Test200, Test300
ulong	gAddr	Start address when writing to serial EEPROM	Test300
uchar	gStat	Serial EEPROM read status storage buffer	Test200

Note: 1. The specifications of the serial EEPROM control middleware allow selection of two devices. The sample application uses only one device, so that value of gDevNo is fixed at 0.

Table 5.6 Static Variables

Type	Variable Name	Description	Used by Function(s)
short	Ret	Storage variable for return value	Test200, Test300
unsigned char	Tr_Dummy	Storage buffer for dummy transmit data*[1]	R_SIO_Rx_Data

Note: 1. Tr_Dummy is a buffer required by the RPDL for receive processing, so it has been added in the sample application.

5.6 Diagram of Function Structure

Figures 5.1 to 5.6 show the structure of the program functions related to the sample application.

Functions supplied with the serial EEPROM control middleware are shown in black, functions created by the sample application are shown in red, and functions supplied with the RPDL are shown in blue.

For details of the functions created by the sample application and the functions supplied with the RPDL, see 5.9, Function Specifications.

For details of the functions supplied with the serial EEPROM control middleware, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

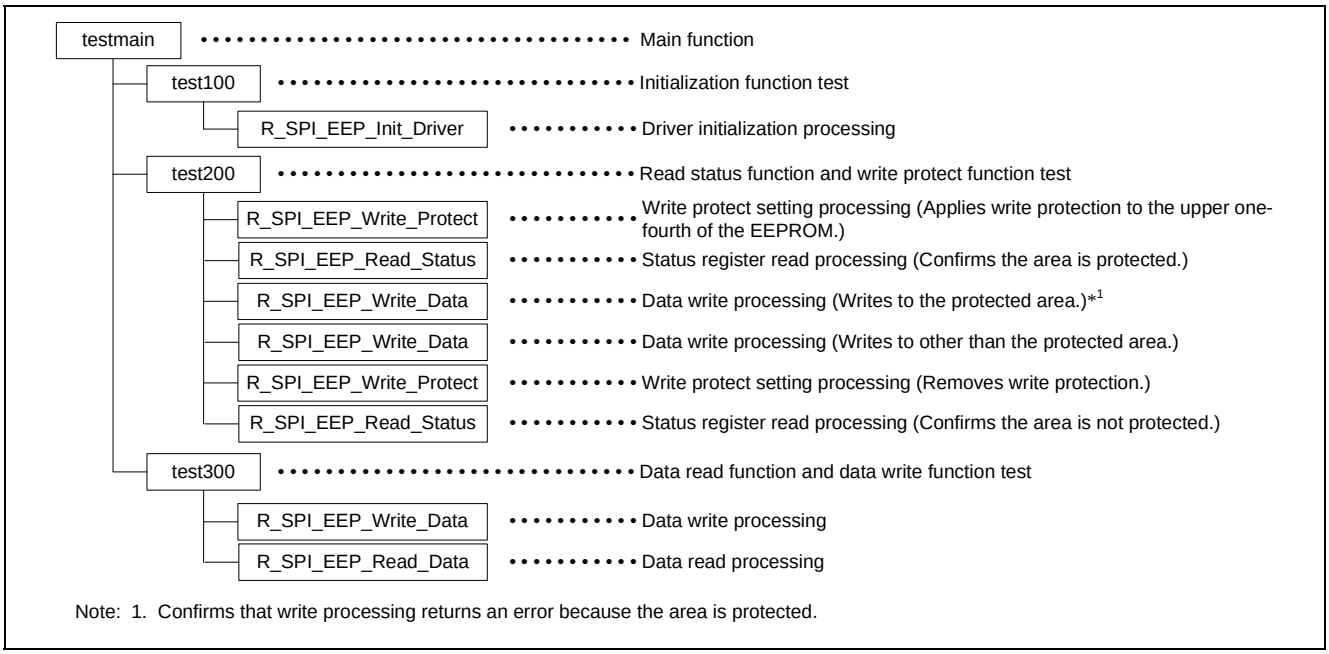


Figure 5.1 Function testmain

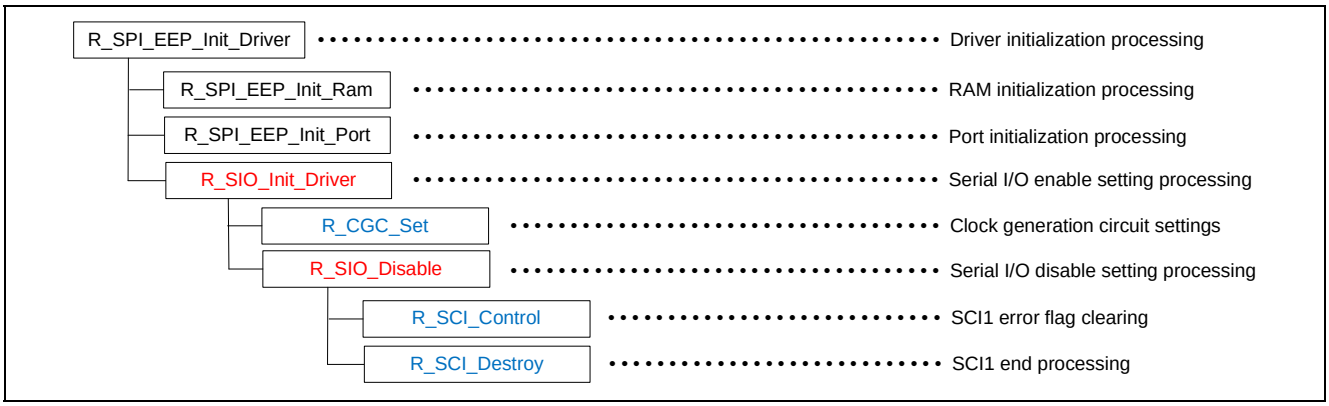
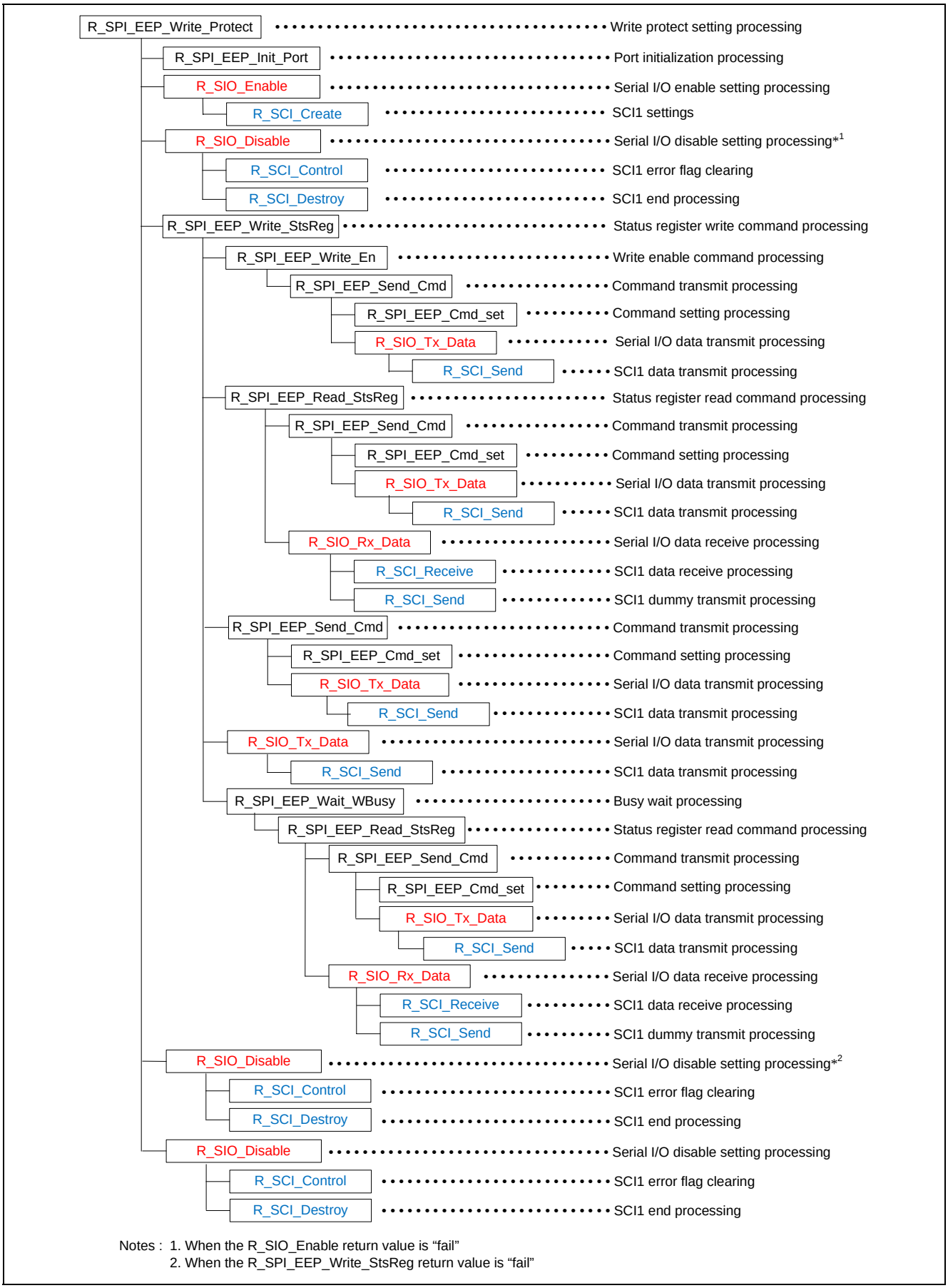
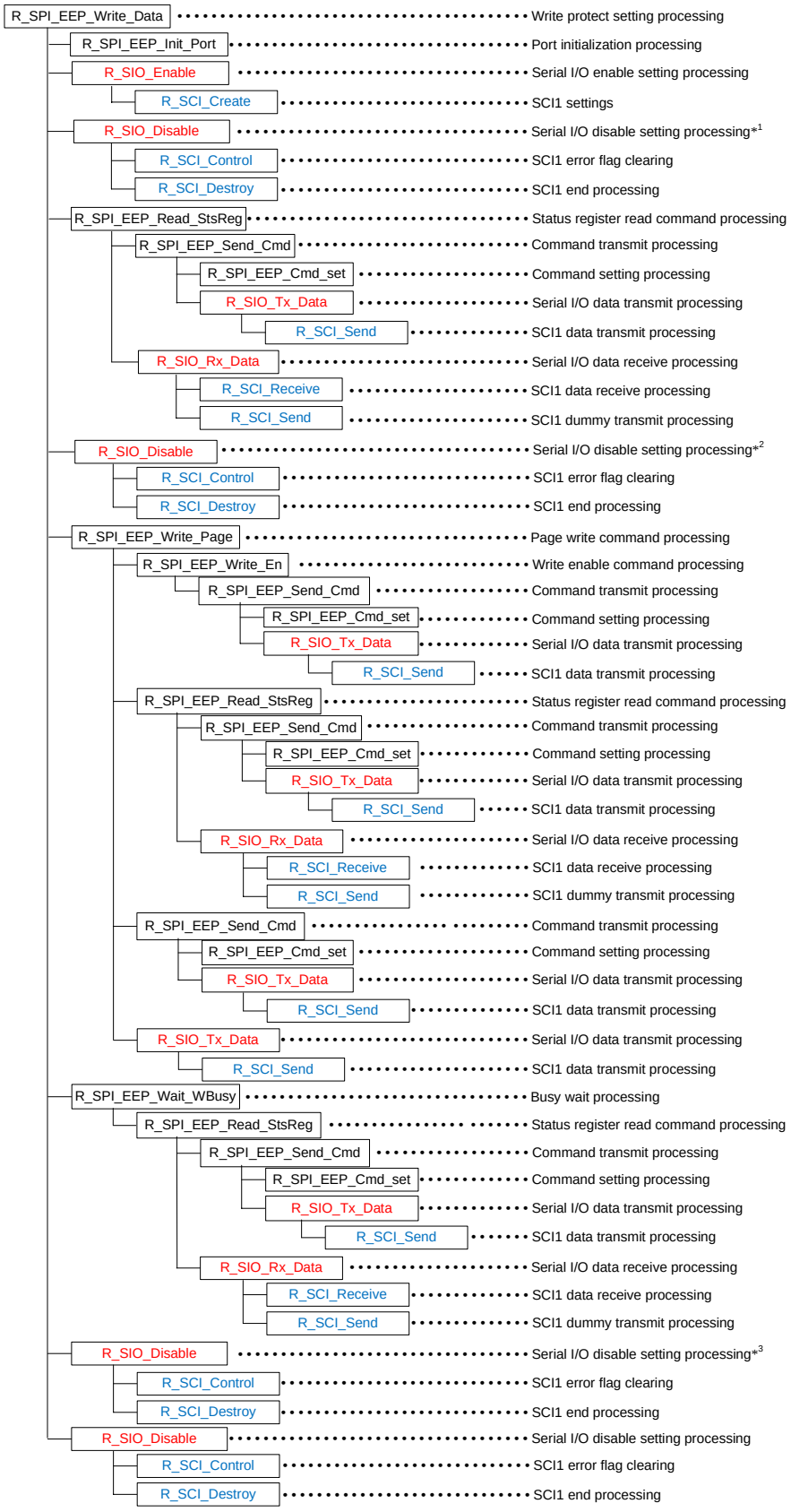


Figure 5.2 Function R_SPI_EEP_Init_Driver



Notes : 1. When the R_SIO_Enable return value is “fail”
2. When the R_SPI_EEP_Write_StsReg return value is “fail”

Figure 5.3 Function R_SPI_EEP_Write_Protect



Notes : 1. When the R_SIO_Enable return value is “fail”
2. When the R_SPI_EEP_Read_StsReg return value is “fail”
3. When the R_SPI_EEP_Wait_WBusy return value is “fail”

Figure 5.4 Function R_SPI_EEP_Write_Data

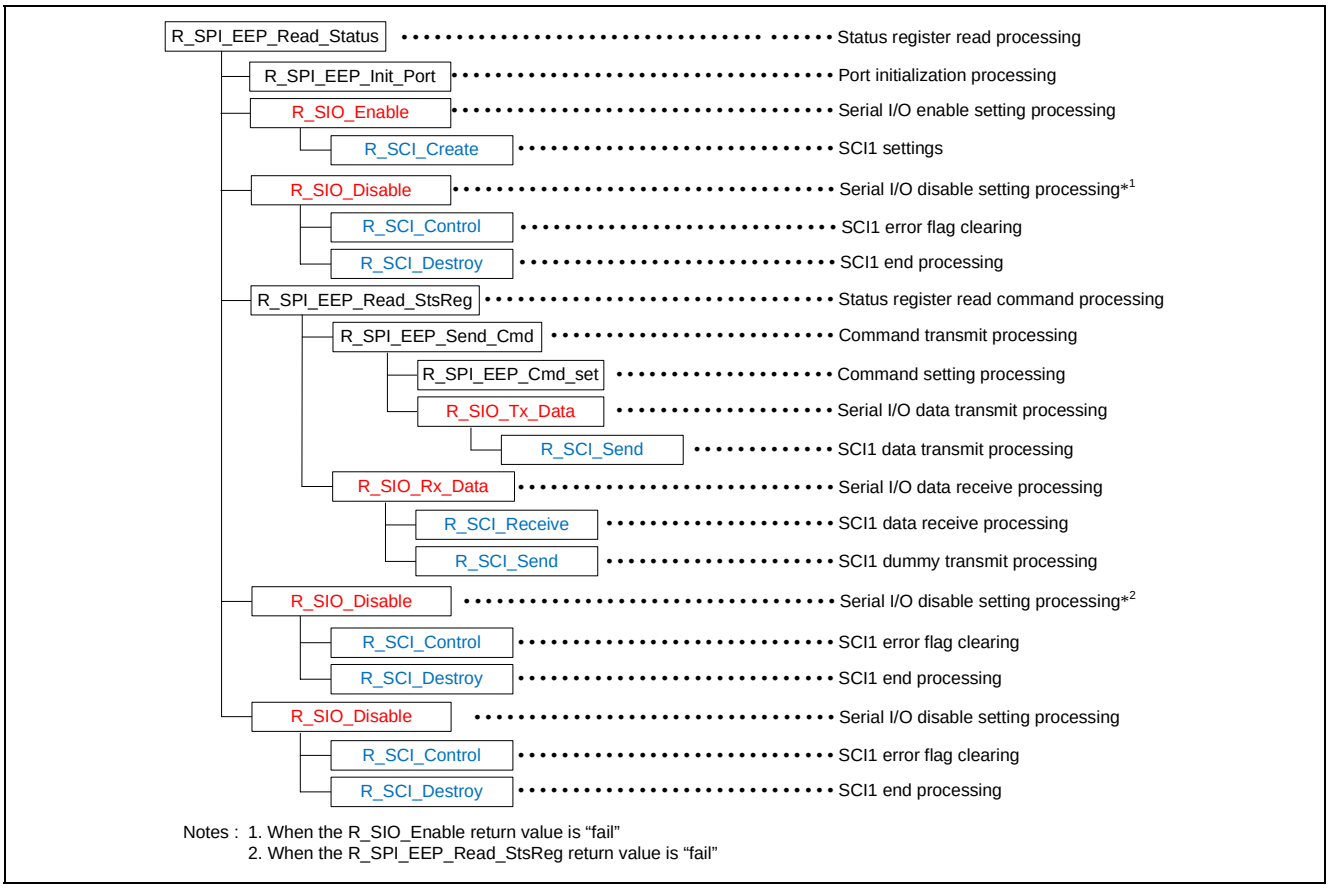


Figure 5.5 Function R_SPI_EEP_Read_Status

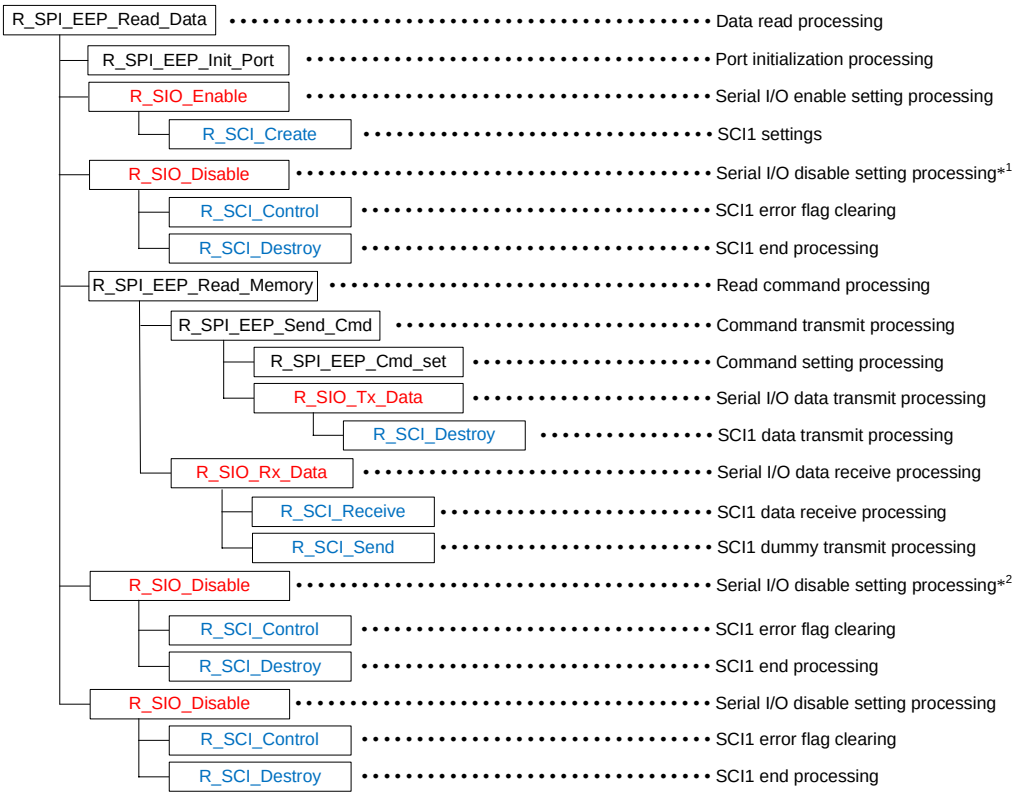


Figure 5.6 Function R_SPI_EEP_Read_Data

5.7 List of Functions

5.7.1 Functions Created by the Sample Application

Table 5.7 lists the functions created by the sample application. For details of these functions, see 5.9.1, Specifications of Functions Created by the Sample Application.

Table 5.7 Functions Created by the Sample Application

Function	Description
R_SIO_Init_Driver	Driver initialization processing
R_SIO_Disable	Serial I/O disable setting processing
R_SIO_Enable	Serial I/O enable setting processing
R_SIO_Tx_Data	Serial I/O data transmit processing
R_SIO_Rx_Data	Serial I/O data receive processing
SCI1RxFunc	Call-back function for receive processing

5.7.2 Functions Supplied with the Serial EEPROM Control Middleware

Table 5.8 lists the functions supplied with the serial EEPROM control middleware. For details of these functions, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

Table 5.8 Functions Supplied with the Serial EEPROM Control Middleware

Function	Description
testmain	Test program
test100	Initialization function test
test200	Read status function and write protect function test
test300	Data read function and data write function test
Trap	Infinite loop processing
R_SPI_EEP_Init_Driver	Driver initialization processing
R_SPI_EEP_Write_Protect	Write protect setting processing
R_SPI_EEP_Read_Status	Status read processing
R_SPI_EEP_Read_Data	Data read processing
R_SPI_EEP_Write_Data	Data write processing
R_SPI_EEP_Init_Ram	Port initialization processing
R_SPI_EEP_Init_Port	Port initialization processing
R_SPI_EEP_Write_StsReg	Status register write command processing
R_SPI_EEP_Read_StsReg	Status register read command processing
R_SPI_EEP_Write_Page	Page write command processing
R_SPI_EEP_Wait_WBusy	Busy wait processing
R_SPI_EEP_Read_Memory	Read command processing
R_SPI_EEP_Send_Cmd	Command transmit processing
R_SPI_EEP_Cmd_set	Command setting processing
R_SPI_EEP_Write_En	Write enable command processing
R_SPI_EEP_Write_Di	Write disable command processing

5.7.3 RPDL Functions Used by the Sample Application

Table 5.9 lists the RPDL functions used by the sample application. For details of these functions, see 5.9.2, Specifications of RPDL Functions Used by the Sample Application.

Table 5.9 RPDL Functions Used by the Sample Application

Function	Description
R_SCI_Create	SCI channel settings
R_SCI_Destroy	SCI channel end
R_SCI_Send	Data transmit on SCI channel
R_SCI_Receive	Data receive on SCI channel
R_SCI_Control	SCI channel control

5.8 Function Correspondence Table

The function correspondence table below lists the RPDL functions called by the serial EEPROM control middleware and the processing performed by those functions.

Table 5.10 Function Correspondence Table

Function Called by Serial EEPROM Control Middleware	RPDL Function Contained in Function	Processing Performed by RPDL Function
R_SIO_Init_Driver	R_CGC_Set	Clock settings
R_SIO_Disable	R_SCI_Control	Error flag clearing
	R_SCI_Destroy	SCI end processing
R_SIO_Enable	R_SCI_Cleate	SCI settings
R_SIO_Tx_Data	R_SCI_Send	SCI data transmit processing
R_SIO_Rx_Data	R_SCI_Receive	SCI receive settings
	R_SCI_Send	SCI dummy transmit processing

5.9 Function Specifications

5.9.1 Specifications of Functions Created by the Sample Application

The specifications of the functions created by the sample application are given below.

R_SIO_Init_Driver

Overview	Driver initialization processing
Header	R_SIO.h
Declaration	error_t R_SIO_Init_Driver(void)
Description	• Uses function R_CGC_Set (supplied with RPDL) to set the clock. • Uses function R_SIO_Disable to initialize the driver.
Arguments	None
Return values	SIO_OK ; Successful operation
Notes	Uses function R_SIO_Disable to perform the following processing, including the SCI communication state before driver initialization: • Stops transmit/receive. • Clears SSR PER, FER, and OERE flags.

R_SIO_Disable

Overview	Serial I/O disable setting processing
Header	R_SIO.h
Declaration	error_t R_SIO_Disable(void)
Description	• Uses function R_SCI_Control (supplied with RPDL) to clear the SSR PER, FER, and OERE flags. • Uses function R_SCI_Destroy (supplied with RPDL) to stop SCI transmit/receive.
Arguments	None
Return values	SIO_OK ; Successful operation
Notes	For details of functions R_SCI_Control and R_SCI_Destroy, see 5.9.2, Specifications of RPDL Functions Used by the Sample Application.

R_SIO_Enable

Overview	Serial I/O enable setting processing
Header	R_SIO.h
Declaration	error_t R_SIO_Enable(uint8_t BrgData)
Description	Uses function R_SCI_Create (supplied with RPDL) to enable the serial I/O function, release serial I/O from the module stopped state, and set the bit rate.
Arguments	uint32_t BrgData ; Bit rate setting value
Return values	SIO_OK ; Successful operation
Notes	For details of function R_SCI_Create, see 5.9.2, Specifications of RPDL Functions Used by the Sample Application.

R_SIO_Tx_Data

Overview	Serial I/O data transmit processing
Header	R_SIO.h
Declaration	error_t R_SIO_Tx_Data(uint16_t TxCnt, uint8_t FAR* pData)
Description	Uses function R_SCI_Send (supplied with RPDL) to transmit the number of bytes of data specified by an argument.
Arguments	• uint16_t TxCnt ; Transmit byte count • uint8_t FAR* pData ; Pointer to transmit data storage buffer
Return values	SIO_OK ; Successful operation
Notes	For details of function R_SCI_Send, see 5.9.2, Specifications of RPDL Functions Used by the Sample Application.

R_SIO_Rx_Data

Overview	Serial I/O data receive processing
Header	R_SIO.h
Declaration	error_t R_SIO_Rx_Data(uint16_t RxCnt, uint8_t FAR* pData)
Description	• Uses function R_SCI_Receive (supplied with RPDL) to set the receive byte count and receive data storage buffer pointer specified by arguments. • Uses function R_SCI_Send (supplied with RPDL) to transmit dummy data.
Arguments	• uint16_t TxCnt ; Receive byte count • uint8_t FAR* pData ; Pointer to receive data storage buffer
Return values	SIO_OK ; Successful operation
Notes	Before receive processing, a number of 0xFF data units equal to the receive count, the maximum receive byte count, are stored in the buffer used by the function for dummy transmission.

SCI1RxFunc

Overview	Call-back function for receive processing
Header	
Declaration	void SCI1RxFunc(void)
Description	This is a call-back function used during receive processing.
Arguments	None
Return values	None
Notes	• The call-back is executed after all the data is received. • The function performs no processing internally.

5.9.2 Specifications of RPDL Functions Used by the Sample Application

The specifications of the RPDL functions used by the sample application are given below.

R_CGC_Set

Overview	Clock generation circuit settings
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_CGC_Set(uint32_t, uint32_t, uint32_t, uint32_t, uint8_t);
Description	• This function must be used to make clock settings before making SCI settings by using the RPDL. • The function makes the following settings: Input frequency; ICLK, PCLK, and BCLK settings; and a setting option related to the clock generation circuit.
Arguments	• 1st argument: 12.5E6 ; Input frequency: 12.5 MHz • 2nd argument: 100E6 ; System clock frequency: 100 MHz • 3rd argument: 50E6 ; Peripheral module clock frequency: 50 MHz • 4th argument: 25E6 ; Bus clock frequency: 25 MHz • 5th argument: PDL_CGC_BCLK_DISABLE ; Setting option: BCLK functions as input pin.
Return values	• All parameters correctly and exclusively specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes this function for making clock settings. For information on this function, see “Clock Generation Circuit” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_CGC_Set” in the “Clock Generation Circuit” description in the RX610 Group Peripheral Driver Library User’s Manual.

R_SCI_Control

Overview	SCI control (SCI error flag clearing)
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Control(uint8_t data1, uint8_t data2);
Description	Clears SCI error flags.
Arguments	• 1st argument: 1 ; SCI channel to be controlled • 2nd argument: PDL_SCI_CLEAR_RECEIVE_ERROR_FLAGS ; Control option: Clear error flags.
Return values	• All parameters correctly specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_Control” in the “Serial Communication Interface” description in the above manual.

R_SCI_Destroy

Overview	SCI end processing
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Destroy(uint8_t data);
Description	Disables SCI transfer operation and puts the SCI into the module stopped state. Note: The SMR register is not initialized, so the transfer format when the SCI restarts is the same as that used before it was stopped.
Arguments	1st argument: 1 ; SCI channel on which transfer operation is disabled
Return values	• All parameters correctly and exclusively specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_Destroy” in the “Serial Communication Interface” description in the above manual.

R_SCI_Create

Overview	SCI settings
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Create(uint8_t data1, uint32_t data2, uint32_t data3, uint8_t data4);
Description	• Enables the SCI. • Makes the following settings related to the SCI: Transfer format, TxDn pin, RxDn pin, data transfer format, transfer bit rate, and SCI interrupt priority level. • This function is used to make settings before initiating SCI transfer operation. • Settings can be changed by using this function only when SCI transfer operation is disabled.
Arguments	• 1st argument: 1 ; Selected SCI channel • 2nd argument: ; Channel settings PDL_SCI_SYNC	; Clock-synchronous, PDL_SCI_TX_CONNECTED	; TxDn pin output required, PDL_SCI_RX_CONNECTED	; RxDn pin input required, PDL_SCI_MSB_FIRST	; MSB-first transfer PDL_SCI_CLK_INT_OUT ; bit clock output on SCKn pin • 3rd argument: BrgData ; Transfer bit rate: 3.125 MHz • 4th argument: 1 ; SCI interrupt priority level
Return values	• All parameters correctly and exclusively specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_Create” in the “Serial Communication Interface” description in the above manual.

R_SCI_Send

Overview	SCI data transmit processing
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Send(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func);
Description	• Makes SCI transmit settings. • Transmits the transmit count specified by the 4th argument and the data in the buffer specified by the 3rd argument. • When no call-back function is specified by the 5th argument, transfer processing is performed by using polling to determine transmit end. Control returns from the function when all transmit processing completes.
Arguments	• 1st argument: 1 ; SCI channel selected for transmission • 2nd argument: PDL_NO_DATA ; Control option: Default setting • 3rd argument: pData ; Transmit data storage buffer • 4th argument: TxCnt ; Total transmit count • 5th argument: PDL_NO_FUNC ; Call-back function: None
Return values	• All parameters correctly and exclusively specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_Send” in the “Serial Communication Interface” description in the above manual.

R_SCI_Receive

Overview	SCI receive settings
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Receive(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func1, void * func2);
Description	• Makes SCI receive settings. • Stores receive data in the buffer specified by the 3rd argument, using function R_SCI_Send to perform the number of dummy transfers specified as the receive count by the 4th argument. • The receive timing is the same as the dummy transmit timing. This function does not perform receive processing. • When all receive (dummy transmit) processing completes, the call-back function specified by the 5th argument is executed after a RxI interrupt occurs.
Arguments	• 1st argument: 1 ; SCI channel selected for reception • 2nd argument: PDL_SCI_DMAC_DTC_TRIGGER_DISABLE, ; DMAC and DTC transfers disabled during control option data byte transmission • 3rd argument: pData ; Receive data storage buffer • 4th argument: RxCnt ; Total receive count • 5th argument: SCI1RxFunc ; Call-back function (after all receive operations complete: SCI1RxFunc() • 6th argument: PDL_NO_FUNC ; Call-back function (for error handling): None
Return values	• All parameters correctly specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_Receive” in the “Serial Communication Interface” description in the above manual.

R_SCI_Send

Overview	Dummy transmit processing for receive processing
Header	r_pdl_sci.h, r_pdl_definitions.h
Declaration	bool R_SCI_Send(uint8_t data1, uint16_t data2, uint8_t * data3, uint16_t data4, void * func);
Description	• This function is used to perform dummy transmission during receive processing. • The transmit/receive count is set by the 4th argument, and receive processing proceeds as the number of dummy transmissions specified as the receive count are performed. • The receive timing is the same as the dummy transmit timing.
Arguments	• 1st argument: 1 ; SCI channel for transfers • 2nd argument: PDL_NO_DATA ; Control option: Default setting • 3rd argument: Tr_Dummy ; Dummy transmit data storage buffer • 4th argument: RxCnt ; Total receive count • 5th argument: PDL_NO_FUNC ; Call-back function: None
Return values	• All parameters correctly and exclusively specified: True (1) • Parameter not correctly specified: False (0)
Notes	• The contents of each dummy transmission is 0xFF. The value 0xFF does not match any EEPROM control command. Do not use a value other than 0xFF for dummy transmissions. • A number of 0xFF data units equal to the receive count are stored in the dummy transmit buffer (Tr_Dummy), so it is necessary to assign the buffer name as the 3rd argument. • The RPDL for the RX610 includes six functions for making SCI settings, including this one. For information on these functions, see “Serial Communication Interface” in the API List by Peripheral Function section of the RX610 Group Peripheral Driver Library User’s Manual. • For details of the arguments of this function, see “R_SCI_send” in the “Serial Communication Interface” description in the above manual.

5.9.3 Specifications of Test Programs Supplied with the Serial EEPROM Control Middleware

The specifications of the test programs supplied with the serial EEPROM control middleware are given below.

testmain

Overview	Main program
Header	
Declaration	void testmain(void)
Description	• This is the main program of the sample application. • It runs functions test100, test200, and test300.
Arguments	None
Return values	None
Notes	To use this function as the main program, resetprg.c has been modified. For details, see 7.1, Modifications to resetprg.c.

test100	
Overview	Initialization function test
Header	mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h
Declaration	void Test100(void)
Description	Performs a test of function R_SPI_EEP_Init_Driver.
Arguments	None
Return values	None
Notes	For details of function R_SPI_EEP_Init_Driver, see the application note Renesas R1EX25xxx Series Serial EEPROM Control Software.

test200	
Overview	Read status function and write protect function test
Header	mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h
Declaration	void Test200(void)
Description	• Performs a test of function R_SPI_EEP_Write_Protect. • Performs a test of function R_SPI_EEP_Read_Status.
Arguments	None
Return values	None
Notes	For details of functions R_SPI_EEP_Write_Protect and R_SPI_EEP_Read_Status, see Renesas R1EX25xxx Series Serial EEPROM Control Software.

test300	
Overview	Data read function and data write function test
Header	mtl_com.h, R_SPI_EEP.h, R_SPI_EEP_io.h
Declaration	void Test300(void)
Description	• Performs a test of function R_SPI_EEP_Write_Data. • Performs a test of function R_SPI_EEP_Read_Data.
Arguments	None
Return values	None
Notes	For details of functions R_SPI_EEP_Write_Protect and R_SPI_EEP_Read_Status, see Renesas R1EX25xxx Series Serial EEPROM Control Software.

trap	
Overview	Infinite loop processing
Header	
Declaration	void trap(void)
Description	Runs an infinite loop.
Arguments	None
Return values	None
Notes	When a function of the serial EEPROM control middleware returns an error during execution of function Test100, Test200, or Test300, control branches to this function.

5.10 Flowcharts

5.10.1 Driver Initialization Processing

Figure 5.7 is a flowchart of the driver initialization processing.

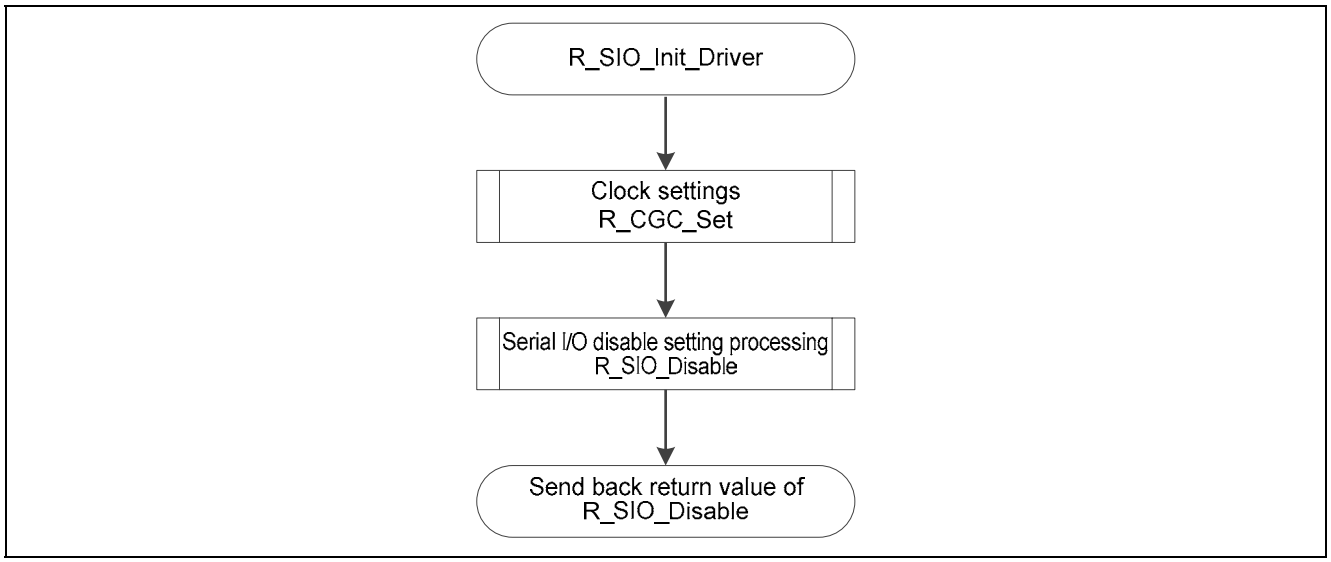


Figure 5.7 Driver Initialization Processing

5.10.2 Serial I/O Disable Setting Processing

Figure 5.8 is a flowchart of the serial I/O disable setting processing.

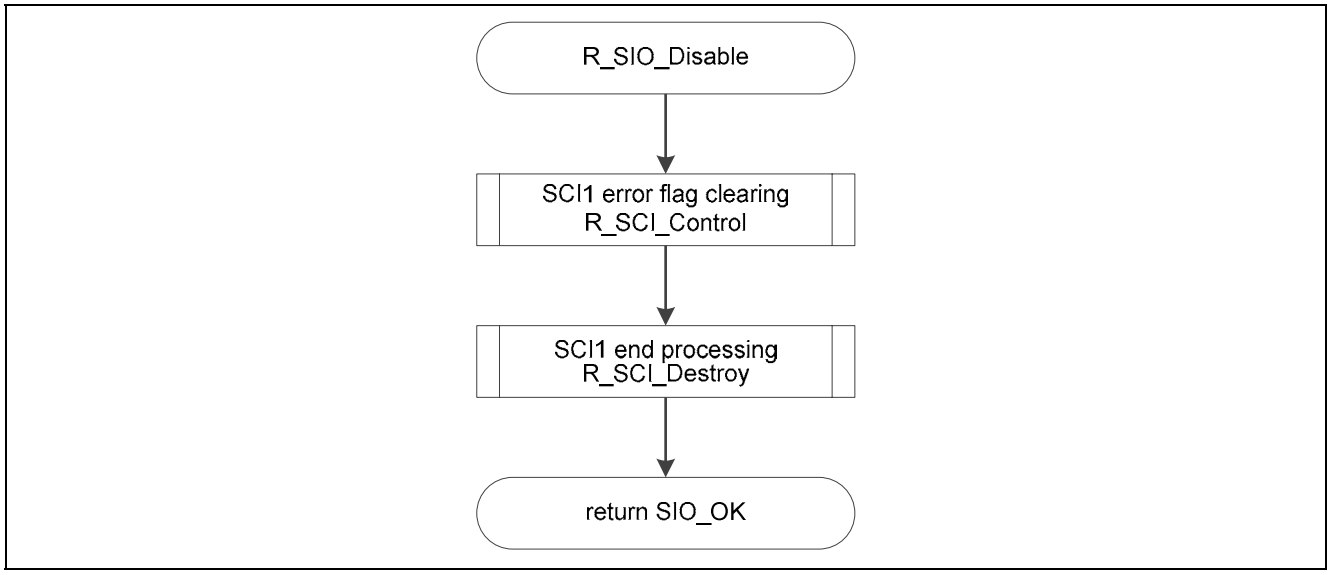


Figure 5.8 Serial I/O Disable Setting Processing

5.10.3 Serial I/O Enable Setting Processing

Figure 5.9 is a flowchart of the serial I/O enable setting processing.

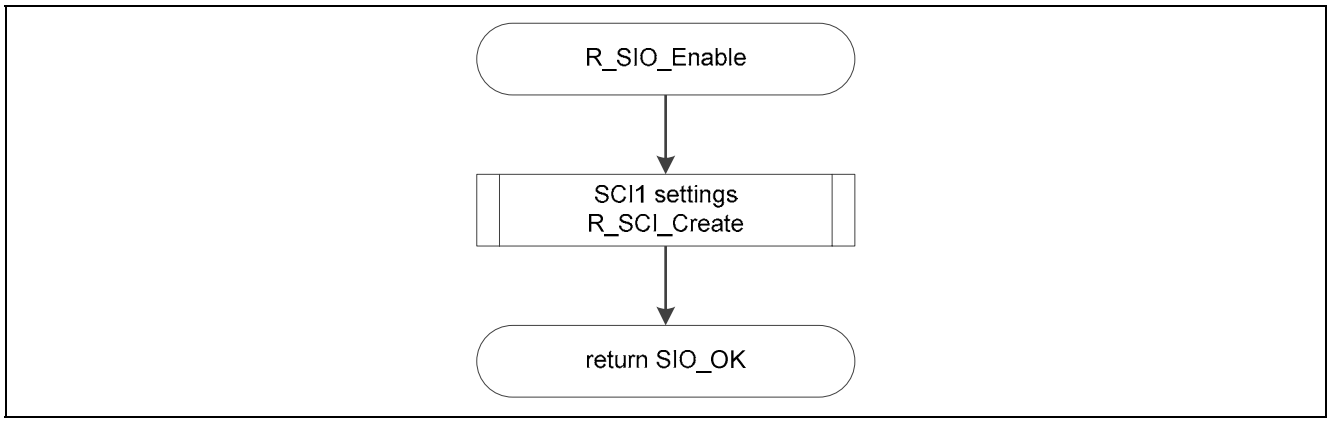


Figure 5.9 Serial I/O Enable Setting Processing

5.10.4 Serial I/O Data Transmit Processing

Figure 5.10 is a flowchart of the serial I/O data transmit processing.

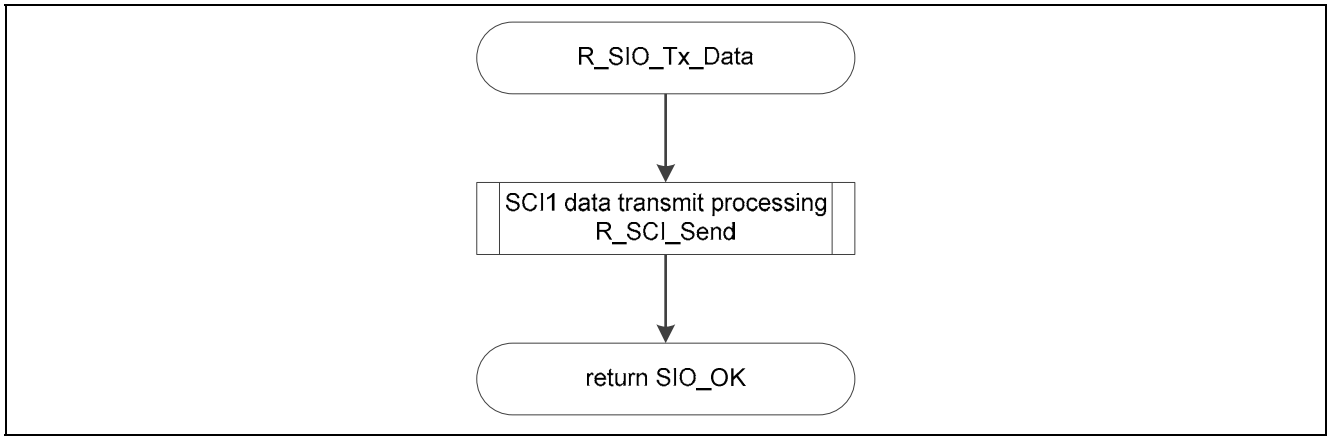


Figure 5.10 Serial I/O Data Transmit Processing

5.10.5 Serial I/O Data Receive Processing

Figure 5.11 is a flowchart of the is a flowchart of the serial I/O data receive processing.

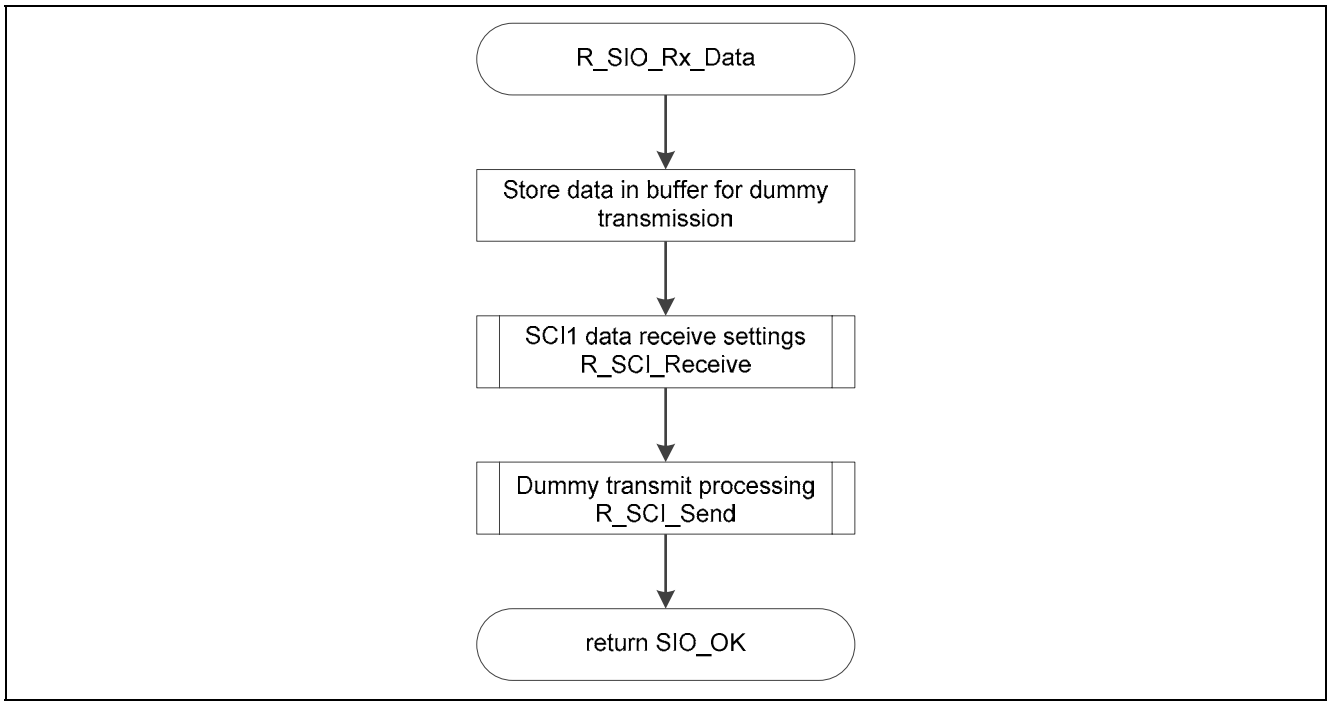


Figure 5.11 Serial I/O Data Receive Processing

5.10.6 Call-Back Function for Receive Processing

Figure 5.12 is a flowchart of the call-back function for receive processing.

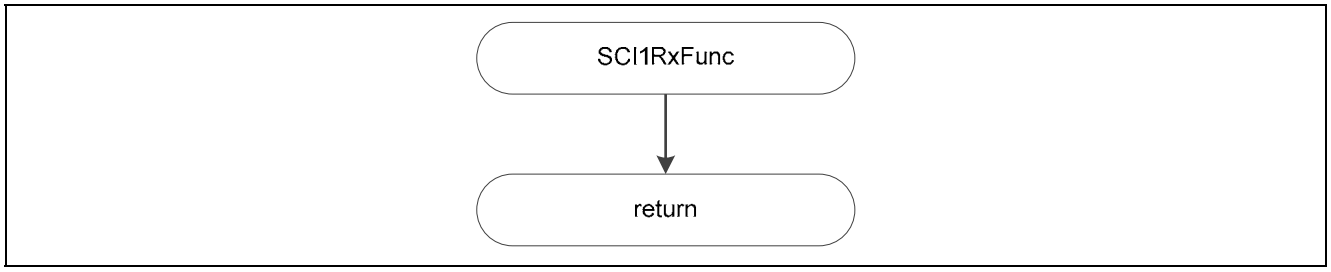


Figure 5.12 Call-Back Function for Receive Processing

6. Sample Code

The sample code is available for download from the Renesas Electronics Web site.

7. Reference Documents

- RX610 Group User's Manual: Hardware, Rev.1.11
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Technical Updates/Technical News
  (The latest information can be downloaded from the Renesas Electronics Web site.)
- C Compiler Manual
  RX Family C/C++ Compiler Package V.1.01 Release 00
  RX Family C/C++ Compiler Package User's Manual V.1.0.1. Rev.1.00
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Application Note
  Serial EEPROM control: Renesas R1EX25xxx Series Serial EEPROM Control Software Rev.1.03 (R01AN0565EJ)
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Library
  Renesas Peripheral Driver Library (R20UT0083EE0100)
  (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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar.28.12	—	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 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 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 a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU 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.

Notice

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

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

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

RENESAS

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

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

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

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

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

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

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

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

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

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

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

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

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

Colophon 1.1