RX62T Group

On-chip Flash Memory Reprogramming in Single-chip Mode via an UART Interface (Master)

R01AN0639EJ0101
Rev.1.01
Mar 13, 2012

Introduction

This application note describes the processing used to transfer, using asynchronous serial communication, the erase block number, write data size, and write data to the program described in the RX62T Group application note titled “On-chip Flash Memory Reprogramming in Single-chip Mode via an UART Interface (Slave)” (R01AN0640EJ).

See the “On-chip Flash Memory Reprogramming in Single-chip Mode via an UART Interface (Slave)” RX62T Group application note for details on erasing and programming the internal flash memory (the user MAT) in the slave.

Target Device

RX62T Group

This program can be used with other RX Family MCUs that have the same I/O registers (peripheral device control registers) as the RX62T Group. Check the latest version of the manual for any additions and modifications to functions. Careful evaluation is recommended before using this application note.

Contents

1. Specifications

- The master sends the erase block number, the write data size, and the write data to the slave using asynchronous serial communication and the slave programs its own user MAT.
- The SCI channel 0 (SCI0) module is used for asynchronous serial communication between the master and the slave.
- Asynchronous serial communication specifications
  - Bit rate: 31,250 bps
  - Data length: 8 bits
  - Parity bits: none
  - Stop bits: 1 bit
- When the switch connected to the master's external interrupt pin (IRQ0-B) is pressed, the master starts serial communications and controls programming of the slave's user MAT.
- Using communication commands, the master tells the slave which one of its user MAT erase blocks (EB00 to EB21) to erase. In this application note, the slave is told to erase the EB08 erase block.
- After the slave completes erasing EB08, the master transmits the write data size (4 bytes) and the write data (8 KB) to the slave.
- Handshaking is used to control communication between the master and slave. In particular, after sending a transmission to the slave, the master waits until an [ACCEPTABLE] (55h) command is returned. After receiving an [ACCEPTABLE] (55h) command, the master starts the next communication.
- When the slave has successfully reprogrammed the user MAT, the master reports the successful completion in the four LEDs connected to its I/O ports. Also, if an error occurs in the slave during communication. the slave will report that error with its LEDs.

Figure 1 shows the specifications of the system used in this application note.

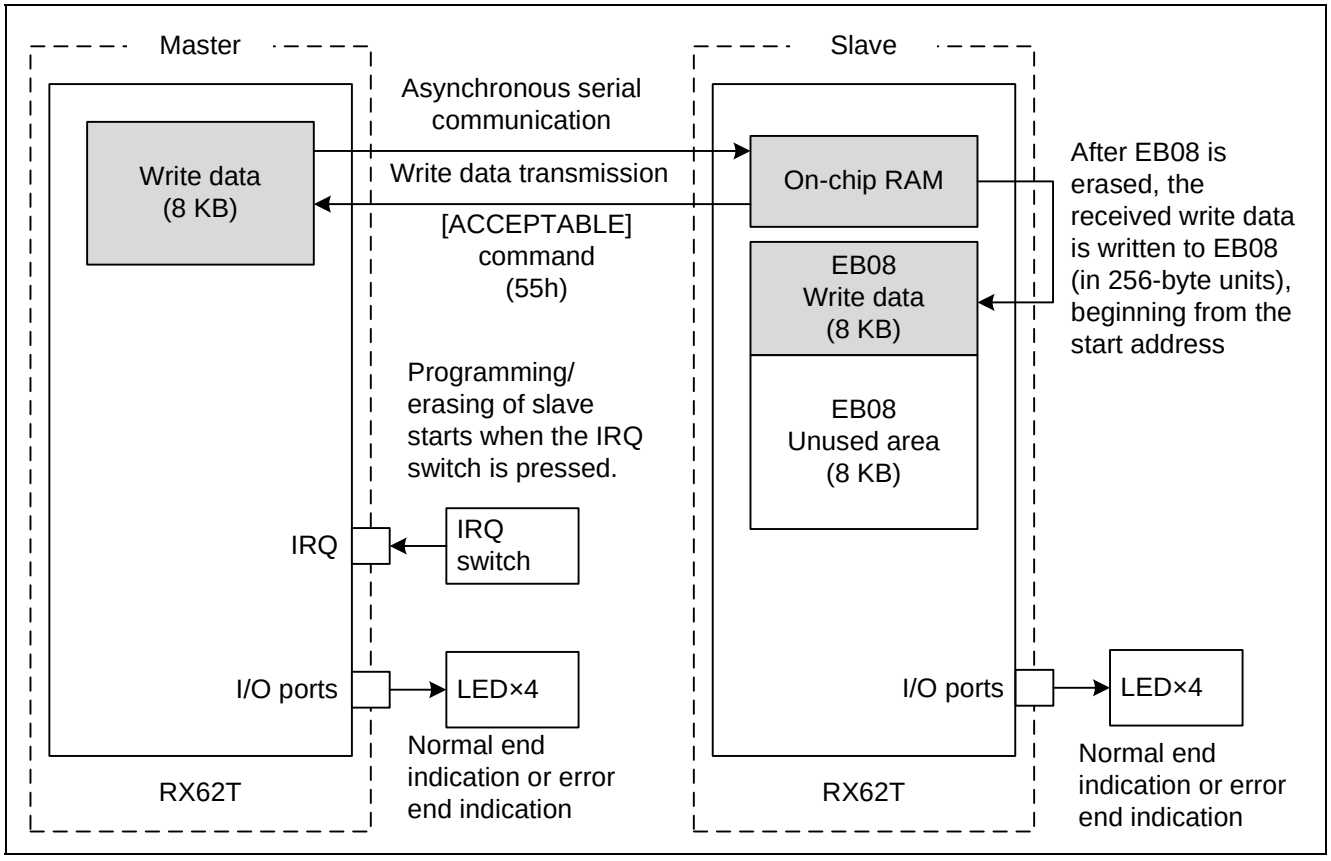
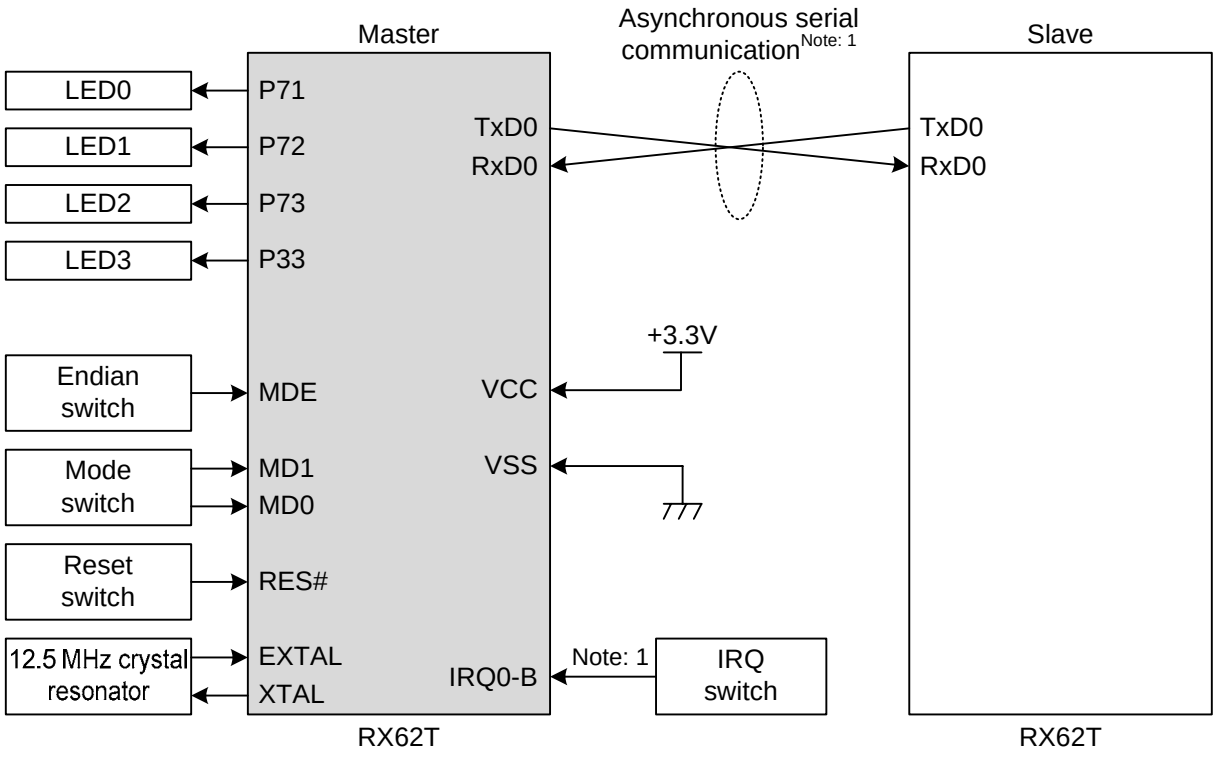


Figure 1 Specifications

Figure 2 shows a hardware configuration diagram of the master device as used in this application note.



Note: 1. The TxD0, RxD0, and IRQ0-B pins of the master are pulled up externally.

Figure 2 Hardware Configuration Diagram of Master Device

2. Operation Confirmation Environment

Table 1 lists the environment required for confirming master operation.

Table 1 Master Operation Confirmation Environment

Item	Description
Device	RX62T Group: R5F562TAADFP (ROM: 256 KB, RAM: 16 KB)
Board	Renesas Starter Kit (Under development as of February 21, 2012)
Power supply voltage	5.0 V
Input clock	12.5 MHz (ICLK = 100 MHz, PCLK = 50 MHz)
Operating temperature	Room temperature
High-performance Embedded Workshop	Version 4.09.00.007
Toolchain	RX Standard Toolchain (V.1.2.0.0)
Debugger/emulator	E1 Emulator
Debugger component	RX E1/E20 SYSTEM V.1.02.00

3. Functions Used

- Clock generation circuit
- Low Power Consumption
- Interrupt control unit
- I/O ports
- Serial Communications Interface
  For details, see the Users' Manual listed in 7, Reference Documents.

4. Operation

4.1 Operation Mode Settings

In the sample program, the master's mode pins are set to MD1 = 1, MD0 = 1 to select single-chip mode as the operating mode, the ROME bit in system control register 0 (SYSCR0) is set to 1 to enable the on-chip ROM.

The master is activated from the user MAT in single-chip mode.

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

Table 2 Operating Mode Settings of Master Device

Mode Pin		SYSCR0 Register		
MD1	**MD0**	**ROME**	**Operating Mode**	**On-chip ROM**
1	1	1	Single-chip mode	Enabled

Note: The initial setting of the ROME bit in the SYSCR0 register is SYSCR0.ROME = 1, so it is not necessary for the sample program to make settings to the SYSCR0 register.

4.2 Clock Settings

The evaluation board used for this application note includes a 12.5 MHz crystal oscillator.

Therefore this application note uses the following settings for the system clock (ICLK) and the peripheral module clock (PCLK): 8× (100 MHz) and 4× (50 MHz).

4.3 Endian Mode Setting

The sample program presented in this application note supports both big- and little-endian mode. Table 3 lists the hardware (MDE pin) endian mode settings of the master device. Note that the master and slave endian settings must match. Note that the master and slave endian settings must match.

Table 3 Endian Mode Settings of Master Device (Hardware)

MDE pin	Endian
0	Little endian
1	Big endian

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

Table 4 Endian Mode Settings of Master Device (Compiler Options)

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

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

4.4 Asynchronous Serial Communication Specifications

In the program described in this application note, asynchronous serial communication is used to transmit communication commands, the erase block number, the write data size, and the write data. Note that the slave transmits the [ACCEPTABLE] command (55h) as a status command for handshaking. The used SCI0 TxD0 and RxD0 pins are each pulled up externally.

Table 5 shows the specifications of the asynchronous serial communication used here.

Table 5 Asynchronous Serial Communication Specifications

Item	Description
Channel	SCI channel 0 (SCI0)
Communication mode	Asynchronous mode
Bit rate	31,250 bps (PCLK = 50 MHz)
Data length	8 bits
Parity bit	None
Stop bit	1 bit
Error	Overrun error, framing error

4.4.1 Communication Command Specifications

Table 6 lists the specifications of the communication commands sent between the master and slave.

Table 6 Communication Command Specifications

Command	Value	Description	Communication Direction
FSTART	10h	Command to start programming/erasing of the user MAT of the slave	Master → slave
ERASE	11h	Command to start erasing of the user MAT of the slave	Master → slave
WRITE	12h	Command to start programming of the user MAT of the slave	Master → slave
ACCEPTABLE	55h	Status command used by the slave to inform the master that it is able to receive data from the master.	Slave → master

4.4.2 Communication Sequence

Figures 3 to 6 show the communication sequence between master and slave.

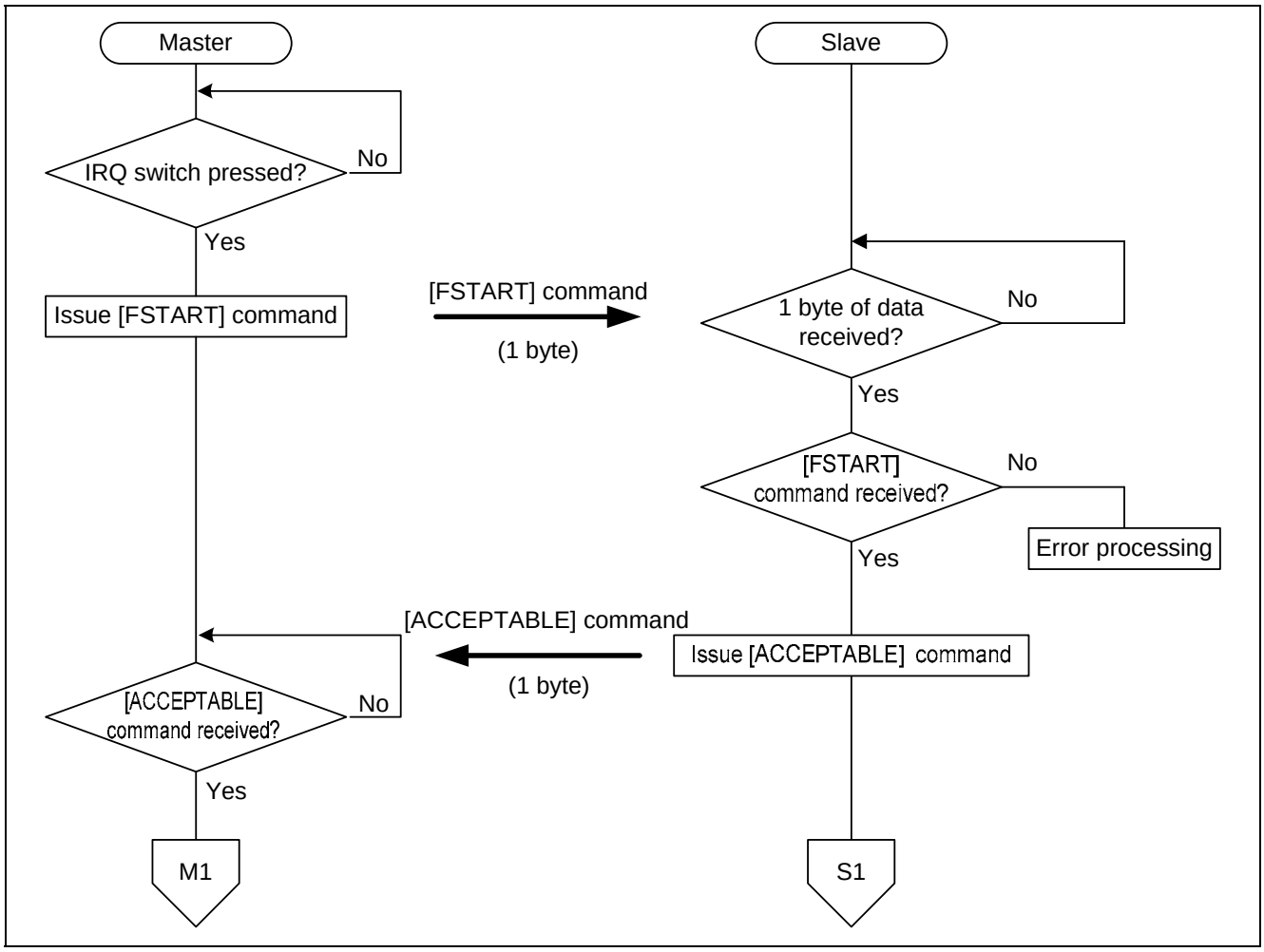


Figure 3 Communication Sequence (1)

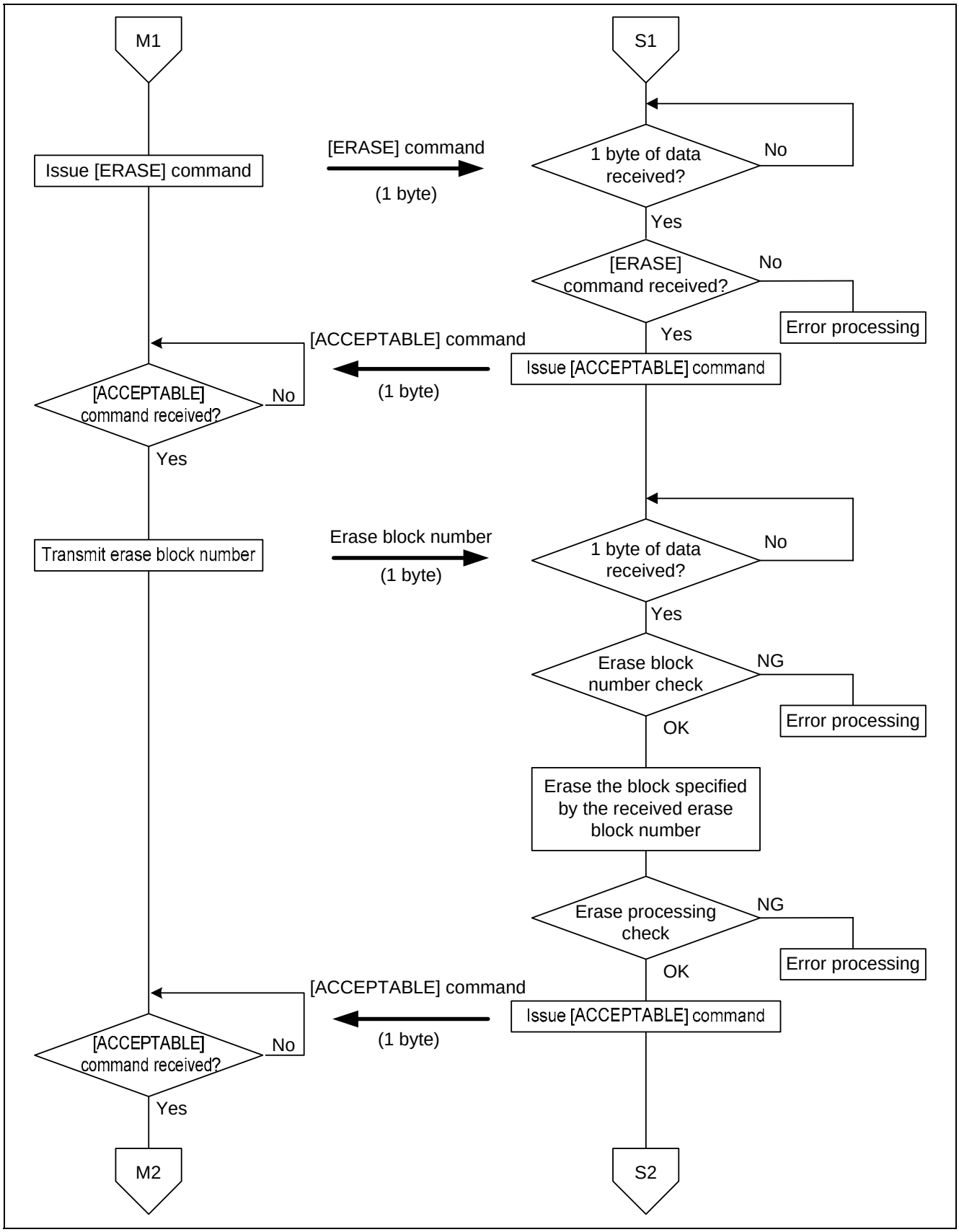


Figure 4 Communication Sequence (2)

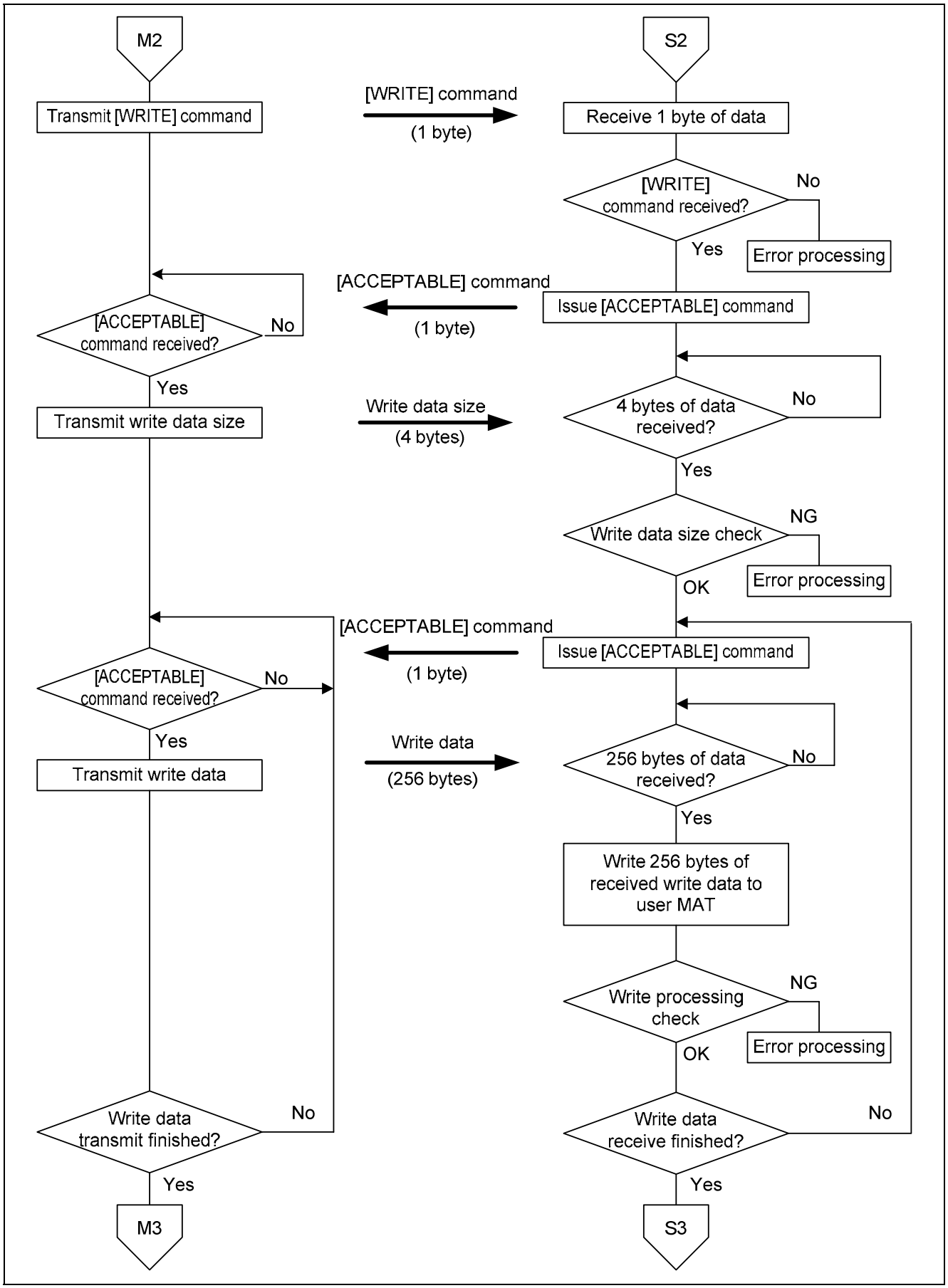


Figure 5 Communication Sequence (3)

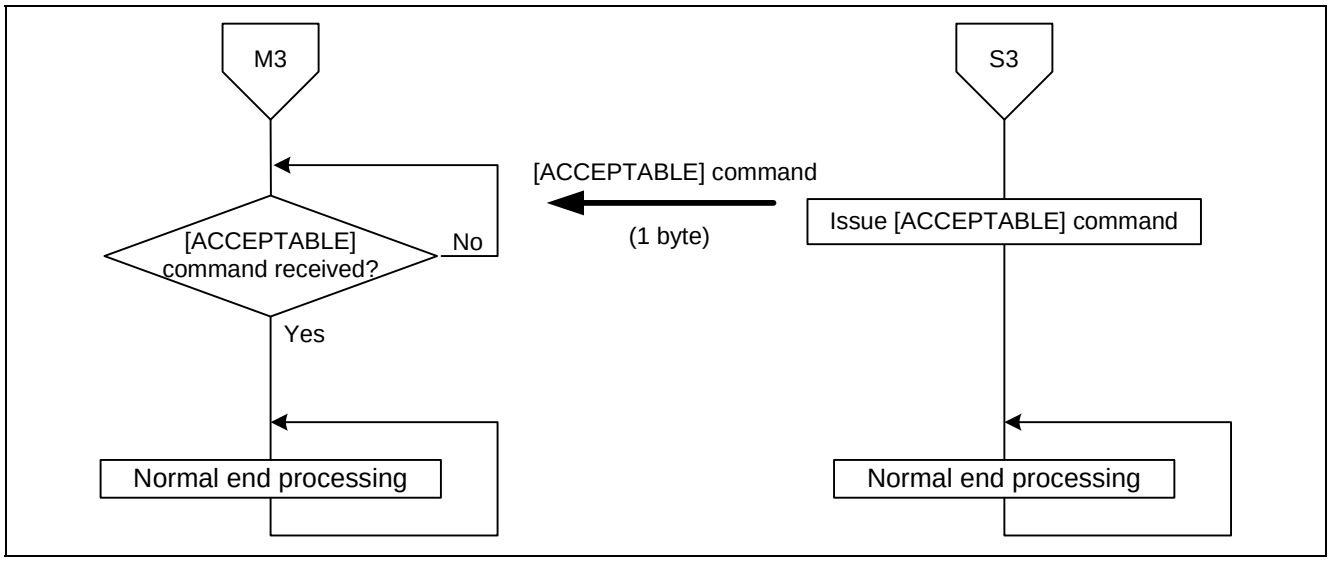


Figure 6 Communication Sequence (4)

4.4.3 Erasure Block Number

After transmitting an ERASE command, the master transmits the erase block number (one byte of data defined by a symbolic constant). Note that since the slave uses the code in the "RX600 Simple Flash API" application note Rev.2.20 (R01AN0544EU0220), the constants defined in the r_flash_api_rx600.h header file are used for this erase block number.

Table 7 lists the erase block numbers defined in r_flash_api_rx600.h. Figure 7 shows the erase block number specifications.

See the “RX600 Simple Flash API” application note listed in section 7, Reference Documents, for details on the erase block numbers.

Table 7 Erase Block Numbers from r_flash_api_rx600.h

Erasure Block Number		
Symbolic Constant	**Value**	**Description**
BLOCK_0	0	Specifies erasure block EB0 (size: 4 KB)
BLOCK_1	1	Specifies erasure block EB1 (size: 4 KB)
BLOCK_2	2	Specifies erasure block EB2 (size: 4 KB)
BLOCK_3	3	Specifies erasure block EB3 (size: 4 KB)
BLOCK_4	4	Specifies erasure block EB4 (size: 4 KB)
BLOCK_5	5	Specifies erasure block EB5 (size: 4 KB)
BLOCK_6	6	Specifies erasure block EB6 (size: 4 KB)
BLOCK_7	7	Specifies erasure block EB7 (size: 4 KB)
BLOCK_8	8	Specifies erasure block EB8 (size: 16 KB)
BLOCK_9	9	Specifies erasure block EB9 (size: 16 KB)
BLOCK_10	10	Specifies erasure block EB10 (size: 16 KB)
BLOCK_11	11	Specifies erasure block EB11 (size: 16 KB)
BLOCK_12	12	Specifies erasure block EB12 (size: 16 KB)
BLOCK_13	13	Specifies erasure block EB13 (size: 16 KB)
BLOCK_14	14	Specifies erasure block EB14 (size: 16 KB)
BLOCK_15	15	Specifies erasure block EB15 (size: 16 KB)
BLOCK_16	16	Specifies erasure block EB16 (size: 16 KB)
BLOCK_17	17	Specifies erasure block EB17 (size: 16 KB)
BLOCK_18	18	Specifies erasure block EB18 (size: 16 KB)
BLOCK_19	19	Specifies erasure block EB19 (size: 16 KB)
BLOCK_20	20	Specifies erasure block EB20 (size: 16 KB)
BLOCK_21	21	Specifies erasure block EB21 (size: 16 KB)

Erasure block number (unsigned char type)

b7	b6	b5	b4	b3	b2	b1	b0
BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0

The sample program presented in this application note programs and erases erasure block EB08 of the slave, so the erasure block number value is [BLOCK_8(8)].

Note: A value shown in table 7, [BLOCK_2(2)] to [BLOCK_21(21)], should be specified as the erasure block number. If a value of [0] to [1] or [22] to [255] is specified as the erasure block number, the slave determines an error to have occurred and error handling takes place.

Figure 7 Erasure Block Number Specifications

4.4.4 Write Data Size

After transmitting a [WRITE] command, the master transmits 4 bytes of write data size. Figure 8 shows the specifications of the write data size.

Write data size (unsigned long type)

b31	b30	b29	b28	b27	b26	b25	b24
SZ31	SZ30	SZ29	SZ28	SZ27	SZ26	SZ25	SZ24
b23	b22	b21	b20	b19	b18	b17	b16
SZ23	SZ22	SZ21	SZ20	SZ19	SZ18	SZ17	SZ16
b15	b14	b13	b12	b11	b10	b9	b8
SZ15	SZ14	SZ13	SZ12	SZ11	SZ10	SZ09	SZ08
b7	b6	b5	b4	b3	b2	b1	b0
SZ07	SZ06	SZ05	SZ04	SZ03	SZ02	SZ01	SZ00

The sample program uses a write size of 8 KB, so the write data size value is [0000 2000h].

Notes: 1. The write data size must be greater than zero and less than or equal to the erase block size for the specified erase block. If 0 or a value greater than the erase block size is specified, the slave will recognize an error and perform error handling.

2. The size of write data transmissions is fixed at 256 bytes. Consequently, if the write data size specifies a value that is not a multiple of 256 bytes, the master transmits write data in units of 256 bytes and then fills in the final unit of write data, which is less than 256 bytes, with bytes of value FFh as padding to reach a total of 256 bytes, which it transmits to the slave.

Figure 8 Write Data Size Specifications

4.4.5 Overrun Error

In this application note, if an overrun error occurs during master asynchronous serial communication reception (the SCI0.SSR.ORER bit is set to 1), the master will perform error handling.

4.4.6 Framing Error

In this application note, if a framing error occurs during master asynchronous serial communication reception (the SCI0.SSR.FER bit is set to 1), the master will perform error handling.

4.5 Normal End Processing

When reprogramming the slave's user MAT has completed normally, the master indicates that normal completion in the four connected LEDs. The normal end indication consists of LED0 to LED3 illuminating one after another in a sequence that is repeated multiple times.

4.6 Error Handling

Table 8 lists the LED display shown when an error occurs in the master in this application note. The master's error handling routine displays the error state in the four LEDs.

Table 8 LED Display when an Error Occurs in the Master

		LED Indication			
Error No.	**Description**	**LED3**	**LED2**	**LED1**	**LED0**
Error No. 01	An overrun or framing error occurred.	Off	Off	Off	On

4.7 LED Connections

Figure 9 shows the connections of the master I/O ports and LED0 to LED3.

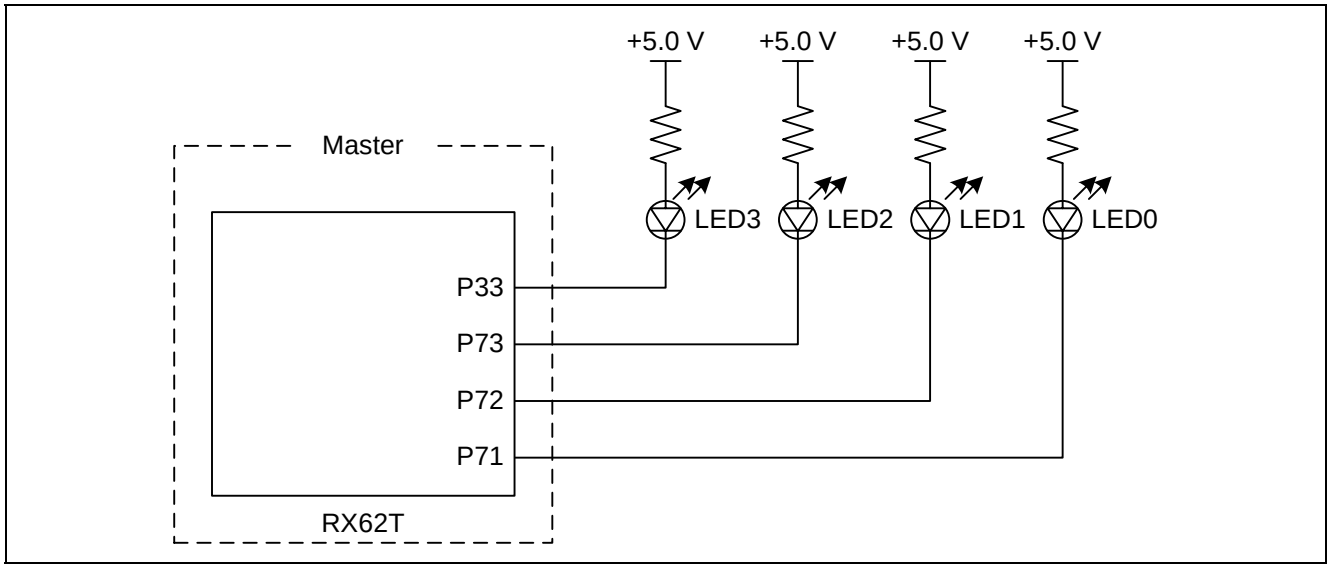


Figure 9 Master Device LED Connection Diagram

As shown in figure 9, high-level output from an I/O port (P71, P72, P73, or P33) causes the corresponding LED among LED0 to LED3 to turn off, and low-level output causes the corresponding LED to illuminate. Table 9 shows the correspondence between I/O port output and LED states.

Table 9 Master I/O Port Output and LED States

I/O Port	**Register Setting**	**I/O Port State**	**LED State**	
P71	PORT7.DR.B1 = 1, PORT7.DDR.B1 = 1	High-level output	LED0	Off
	PORT7.DR.B1 = 0, PORT7.DDR.B1 = 1	Low-level output		On
P72	PORT7.DR.B2 = 1, PORT7.DDR.B2 = 1	High-level output	LED1	Off
	PORT7.DR.B2 = 0, PORT7.DDR.B2 = 1	Low-level output		On
P73	PORT7.DR.B3 = 1, PORT7.DDR.B3 = 1	High-level output	LED2	Off
	PORT7.DR.B3 = 0, PORT7.DDR.B3 = 1	Low-level output		On
P33	PORT3.DR.B3 = 1, PORT3.DDR.B3 = 1	High-level output	LED3	Off
	PORT3.DR.B3 = 0, PORT3.DDR.B3 = 1	Low-level output		On

4.8 IRQ Switch

Figure 10 shows a diagram of the connection between the external interrupt pin (IRQ0-B) of the master and the IRQ switch. Programming/erasing of the user MAT of the slave starts when the IRQ switch connected to the master is pressed.

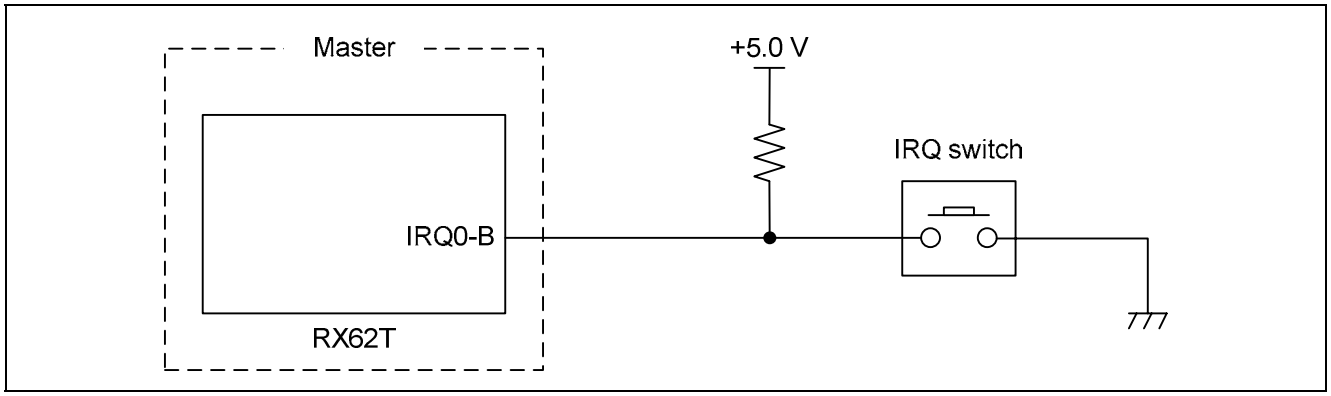


Figure 10 Master IRQ Switch Connection Diagram

The master determines that the IRQ switch is in the depressed state by detecting the falling edge of the IRQ0-B pin. No interrupt handling is performed, and the determination of the IRQ switch state is made by detecting that the IRQ0-B interrupt status flag (bit IR in IR064) has been set to 1.

4.9 Handshaking Control

The master uses handshaking with the slave for communications control.

Handshaking control in the master consists of waiting after serial communication until an [ACCEPTABLE] command (55h) is received from the slave. The master only starts the next serial communication after it has received an [ACCEPTABLE] command from the slave.

4.10 Section Settings

Table 10 shows the section settings for the master device.

Table 10 Section Settings of Master Device

Section	Start Address	Description
B_1	0000 1000h	Uninitialized data area (ALIGN = 4)
R_1		Area mapped in RAM as the [D-1] section by the ROM support option.
B_2		Uninitialized data area (ALIGN = 2)
R_2		Area mapped in RAM as the [D-2] section by the ROM support option.
B		Uninitialized data area (ALIGN = 4)
R		Area mapped in RAM as the [D] section by the ROM support option.
SI		Interrupt stack area
CP_DATA_1	FFFF C000h	Constant area (ALIGN = 1) (write data (8 KB))
PResetPRG	FFFF E000h	Program area (PowerON_Reset_PC program)
C_1	FFFF E100h	Constant area (ALIGN = 1)
C_2		Constant area (ALIGN = 2)
C		Constant area (ALIGN = 4)
C$*		Section initialization table of uninitialized data area, relocatable vector area
D_1		Initialized data area (ALIGN = 1)
D_2		Initialized data area (ALIGN = 2)
D		Initialized data area (ALIGN = 4)
P		Program area
PIntPRG		Program area (interrupt program)
W*		Switch statement branch table area (ALIGN = 4)
FIXEDVECT	FFFF FFD0h	Fixed vector area

5. Software Description

5.1 File Structure

Table 11 shows the file structure of the master device. In addition to the files listed in table 11, some files generated automatically by High-performance Embedded Workshop are used as well.

Table 11 File Structure of Master Device

File Name	Description
resetprg.c[Note: 1]	Initial settings
main.c	In addition to main processing, this program handles send/receive control for communication commands transmitted by asynchronous serial communication with the slave, transmission control for sending the erase block number, write data size, and write data, and LED display control for both normal completion and when an error occurs.
r_flash_api_rx600.h	Simple Flash API include header file (see the erase block number)[Note: 2]
r_flash_api_rx600_config.h	Simple Flash API parameter settings include header file[Note: 2]
mcu_info.h	Simple Flash API parameter settings include header file[Note: 2]

Notes: 1. This file is generated automatically by High-performance Embedded Workshop. In the sample program it has been edited to restore a line in the PowerON_Reset_PC function calling the HardwareSetup function, which was originally commented out. In the edited version the HardwareSetup function in the main.c file is called from the PowerON_Reset_PC function.
2. See the Simple Flash API application note for details.

5.2 Function Structure

Table 12 lists the functions for the master device and figure 11 shows the hierarchy of these functions.

Table 12 Master Device Functions

Function	File Name	Description
PowerON_Reset_PC	resetprg.c	Initial settings function
HardwareSetup	main.c	MCU initial settings function
main	main.c	Main function
Indicate_Ending_LED	main.c	Normal end processing function
SCI_Trs1byte	main.c	1 byte data transmission function
SCI_Trsnbyte	main.c	n byte data transmission function
SCI_Rcv1byte	main.c	1 byte data reception function

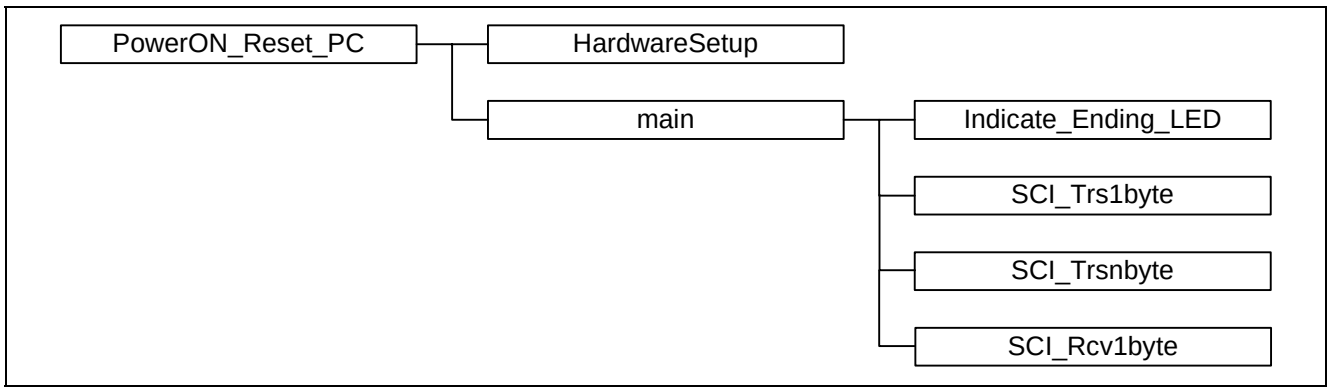


Figure 11 Hierarchy of Master Device Functions

5.3 Symbolic Constants

Table 13 lists the symbolic constants used by the master device.

Table 13 Symbolic Constants of Master Device

Symbolic Constant	Setting Value	Description	Functions Used By
FSTART	0x10	Programming/erase start command	main
ERASE	0x11	Erase start command	main
WRITE	0x12	Programming start command	main
ACCEPTABLE	0x55	Status command sent from the slave	main
LED_ON	0	Set value used when the LED is on	Indicate_Ending_LED SCI_Rcv1byte
LED_OFF	1	Set value used when the LED is off	HardwareSetup Indicate_Ending_LED SCI_Rcv1byte
RSK_LED0	PORT7.DR.BIT.B1	On/off control of LED 0 on the evaluation board	HardwareSetup Indicate_Ending_LED SCI_Rcv1byte
RSK_LED1	PORT7.DR.BIT.B2	On/off control of LED 1 on the evaluation board	HardwareSetup Indicate_Ending_LED
RSK_LED2	PORT7.DR.BIT.B3	On/off control of LED 2 on the evaluation board	HardwareSetup Indicate_Ending_LED
RSK_LED3	PORT3.DR.BIT.B3	On/off control of LED 3 on the evaluation board	HardwareSetup Indicate_Ending_LED
RSK_LED0_DDR	PORT7.DDR.BIT.B1	I/O control for LED 0 on the evaluation board	HardwareSetup
RSK_LED1_DDR	PORT7.DDR.BIT.B2	I/O control for LED 1 on the evaluation board	HardwareSetup
RSK_LED2_DDR	PORT7.DDR.BIT.B3	I/O control for LED 2 on the evaluation board	HardwareSetup
RSK_LED3_DDR	PORT3.DDR.BIT.B3	I/O control for LED 3 on the evaluation board	HardwareSetup
FALL_EDGE	1	Falling edge setting	HardwareSetup
RISE_EDGE	2	Rising edge setting	HardwareSetup
SW_ON	1	START_SW_IR value when the IRQ switch is on	—
SW_OFF	0	START_SW_IR value when the IRQ switch is off	HardwareSetup
START_SW_IR	ICU.IR[IR_ICU_IRQ0].BIT.IR	IRQ switch state	main
START_SW_PFC	IOPORT.PF8IRQ.BIT.ITS0	IRQ switch pin selection	HardwareSetup
START_SW_ICR	PORTE.ICR.BIT.B5	IRQ switch input buffer setting	HardwareSetup
START_SW_IRQMD	ICU.IRQCR[0].BIT.IRQMD	IRQ switch detection setting	HardwareSetup
RxD0_ICR	PORTB.ICR.BIT.B1	RxD0 input buffer setting	HardwareSetup
WAIT_SCI1BIT	1920	Standby time data used after setting the SCI0 BRR register	HardwareSetup

Symbolic Constant	Setting Value	Description	Functions Used By
WAIT_LED	2000000	LED illumination interval data for indication of successful completion of programming/erasing of slave user MAT	Indicate_Ending_LED
TRS_SIZE	256	Write data transmit size	main
BUF_SIZE	8192	Write buffer size	main
WRITE_SIZE	BUF_SIZE	Write data storage area size	main

5.4 Constant Variables

Table 14 lists the constant variables used by the master device.

Table 14 Constant Variables of Master Device

Constant	Type	Description
SAMPLE_DATA[BUF_SIZE]	const unsigned char	Write data (8,192) to be sent to the slave for programming the user MAT In this sample program, the SAMPLE_DATA[BUF_SIZE] buffer is allocated in the CP_DATA_1 section and allocated to erase blocks EB03 to EB02 (FFFF C000h to FFFF DFFFh).

5.5 RAM Variables

The master program portion of the sample program does not use RAM variables.

5.6 I/O Registers

The I/O registers of the master device used by the sample program are listed below. Note that the setting values shown are those used in the sample program and differ from the initial setting values.

(1) Clock Generation Circuit

System Clock Control Register (SCKCR) **Number of Bits: 32** **Address: 0008 0020h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b11 to b8	PCK[3:0]	0001	Peripheral module clock (PCLK) select bits	0001: × 4 PCLK = 50 MHz	R/W
b27 to b24	ICK[3:0]	0000	System clock (ICLK) select bits	0000: × 8 ICLK = 100 MHz	R/W

(2) I/O Ports

Port 7 Data Register (P7.DR) **Number of Bits: 8** **Address: 0008 C027h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b1	B1	0	P71 output data storage bit	0: Output data = 0	R/W
		1		1: Output data = 1	
b2	B2	0	P72 output data storage bit	0: Output data = 0	R/W
		1		1: Output data = 1	
b3	B3	0	P73 output data storage bit	0: Output data = 0	R/W
		1		1: Output data = 1	

Port 3 Data Register (P3.DR) **Number of Bits: 8** **Address: 0008 C023h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b3	B3	0	P33 output data storage bit	0: Output data = 0	R/W
		1		1: Output data = 1	

Port 7 Data Direction Register (P7.DDR) **Number of Bits: 8** **Address: 0008 C007h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b1	B1	1	P71 I/O select bit	1: Output port	R/W
b2	B2	1	P72 I/O select bit	1: Output port	R/W
b3	B3	1	P73 I/O select bit	1: Output port	R/W

Port 3 Data Direction Register (P3.DDR) **Number of Bits: 8** **Address: 0008 C003h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b3	B3	1	P33 I/O select bit	1: Output port	R/W

Port Function Register 8 (PF8IRQ) **Number of Bits: 8** **Address: 0008 C108h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0, b1	ITS[1:0]	01	IRQ0 pin select bit	01: PE5 set as IRQ0-B input pin	R/W

Port E Input Buffer Control Register (PE.ICR) **Number of Bits: 8** **Address: 0008 C06Eh**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b5	B5	1	PE5 input buffer control bit	1: PE5 input buffer enabled	R/W

Port B Input Buffer Control Register (PB.ICR) **Number of Bits: 8** **Address: 0008 C06Bh**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b1	B1	1	PB1 input buffer control bit	1: PB1 input buffer enabled	R/W

(3) Low Power Consumption

Module Stop Control Register B (MSTPCRB) **Number of Bits: 32** **Address: 0008 0014h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b31	MSTPB31	0	Serial communication interface 0 module stop setting bit	0: SCI0 module stop state canceled	R/W

(4) Serial Communications Interface 0 (SCI0)

SCI0 Serial Control Register (SCI0.SCR) **Number of Bits: 8** **Address: 0008 8242h**

(Serial communication interface mode (SMIF bit in SCI0.SCMR = 0))

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b1, b0	CKE[1:0]	00	Clock enable bits	(For asynchronous communication) 00: Internal baud rate generator The SCK0 pin is set to be an I/O port.	R/W Note: 1
b2	TEIE	0	Transmit end interrupt enable bit	0: TEI0 interrupt disabled	R/W
b4	RE	0	Receive enable bit	0: Serial reception disabled	R/W Note: 2
		1		1: Serial reception enabled	
b5	TE	0	Transmit enable bit	0: Serial transmission disabled	R/W Note: 2
		1		1: Serial transmission enabled	
b6	RIE	0	Receive interrupt enable bit	0: RXI0 and ERI0 interrupts disabled	R/W
		1		1: RXI0 and ERI0 interrupts enabled	
b7	TIE	0	Transmit interrupt enable bit	0: TXI0 interrupt disabled	R/W
		1		1: TXI0 interrupt enabled	

Notes: 1. Writing to these bits is possible only when the TE and RE bits are both cleared to 0.
2. A value of 1 may be written to either these bits only when the TE and RE bits are both cleared to 0. Also, 0 may be written to both the TE and RE bits after one of them has been set to 1.

SCI0 Serial Mode Register (SCI0.SMR) **Number of Bits: 8** **Address: 0008 8240h**

(Serial communication interface mode (SMIF bit in SCI0.SCMR = 0))

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b1, b0	CKS[1:0]	00	Clock select bit	00: PCLK clock (n = 0) Note: 1	R/W Note: 2
b2	MP	0	Multiprocessor mode bit	(Only in asynchronous communication mode) 0: Multiprocessor communication function disabled	R/W Note: 2
b3	STOP	0	Stop bits length select bit	(Only in asynchronous communication mode) 0: One stop bit	R/W Note: 2
b5	PE	0	Parity enable bit	(Only in asynchronous communication mode) • Transmission 0: No parity bits • Reception 0: Reception with no parity	R/W Note: 2
b6	CHR	0	Character length bit	(Only in asynchronous communication mode) 0: Transmission and reception with an 8-bit data length	R/W Note: 2
b7	CM	0	Communication mode bit	0: Asynchronous mode	R/W Note: 2

Notes: 1. For information on *n* setting values, see the User's Manual listed in 7, Reference Documents.
2. Writing to these bits is possible only when the TE and RE bits in SCI0.SCR are both cleared to 0 (serial transmission and serial reception both disabled).

SCI0 Smart Card Mode Register (SCI0.SCMR) **Number of Bits: 8** **Address: 0008 8246h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0	SMIF	0	Smart card interface mode select bit	0: Serial communication interface mode	R/W Note: 1
b3	SDIR	0	Bit order selection bit	0: LSB-first transmission/reception	R/W Note: 1

Note: 1. Writing to this bit is possible only when the TE and RE bits in SCI0.SCR are both cleared to 0 (serial transmission and serial reception both disabled).

SCI0 Bit Rate Register (SCI0.BRR) **Number of Bits: 8** **Address: 0008 8241h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b7 to b0	—	00110001 Note: 1	—	31h: Bit rate = 31,250 bps (When PCLK is 50 MHz)	R/W Note: 2

Notes: 1. For information on *BRR* setting values, see the User's Manual listed in 7, Reference Documents.
2. While this register can be read at any time, it can only be written when both the SCI0.SCR.TE bit and the SCI0.SCR.RE bits are 0 (serial transmission disabled and serial reception disabled).

SCI0 Serial Status Register (SCI0.SSR) **Number of Bits: 8** **Address: 0008 8244h**

(Serial communication interface mode (SMIF bit in SCI0.SCMR = 0))

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b2	TEND	—	Transmit end flag	0: Character transmission in progress 1: Character transmission finished	R
b4	FER	— Note: 1	Framing error flag	0: No framing error occurred 1: A framing error occurred	R/(W) Note: 2
b5	ORER	— Note: 1	Overrun error flag	0: No overrun error occurred 1: An overrun error occurred	R/(W) Note: 2

Notes: 1. The FER and ORER bits are handled as read-only in this application note. Writing to these bits to clear the flags to 0 is not performed.
2. Only writing 0 to clear the flag is allowed.

SCI0 Transmit Data Register (SCI0.TDR) **Number of Bits: 8** **Address: 0008 8243h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b7 to b0	—	— Note: 1	—	Stores transmit data.	R/W

Note: 1. The transmitted data is stored in this field.

SCI0 Receive Data Register (SCI0.RDR) **Number of Bits: 8** **Address: 0008 8245h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b7 to b0	—	—	—	Stores receive data.	R

(5) Interrupt Control Unit (ICU)

Interrupt Priority Register 80 (IPR80) **Number of Bits: 8** **Address: 0008 7380h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b3 to b0	IPR[3:0]	0000	SCI0 Interrupt priority level setting bits	0000: Level 0 (interrupt disabled)	R/W

IRQ Control Register 0 (IRQCR0) **Number of Bits: 8** **Address: 0008 7500h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b3, b2	IRQMD[1:0]	01	IRQ0 detection select bits	01: Falling edge	R/W

Interrupt Request Enable Register 1A (IER1A) **Number of Bits: 8** **Address: 0008 721Ah**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b7	IEN7	0	RXI0 Interrupt request enable bit 7	0: RXI0 interrupt disabled	R/W

Interrupt Request Enable Register 1B (IER1B) **Number of Bits: 8** **Address: 0008 721Bh**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0	IEN0	0	TXI0 Interrupt request enable bit 0	0: TXI0 interrupt disabled	R/W

Interrupt Request Register 064 (IR064) **Number of Bits: 8** **Address: 0008 7040h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0	IR	0	IRQ0 Interrupt status flag	0: No IRQ0 interrupt request 1: IRQ0 interrupt request	R/(W) *1

Note: 1. Only 0 may be written to this bit to clear the flag. Writing 1 is prohibited.

Interrupt Request Register 215 (IR215) **Number of Bits: 8** **Address: 0008 70D7h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0	IR	0	RXI0 Interrupt status flag	0: No RXI0 interrupt request 1: RXI0 interrupt request	R/(W) *1

Note: 1. Only 0 may be written to this bit to clear the flag. Writing 1 is prohibited.

Interrupt Request Register 216 (IR216) **Number of Bits: 8** **Address: 0008 70D8h**

Bit	Symbol	Setting Value	Bit Name	Function	R/W
b0	IR	0	TXI0 Interrupt status flag	0: No TXI0 interrupt request 1: TXI0 interrupt request	R/(W) *1

Note: 1. Only 0 may be written to this bit to clear the flag. Writing 1 is prohibited.

5.7 Function Specifications

The specifications of the master device functions are as follows.

(1) PowerON_Reset_PC Function

(a) Functional overview

The PowerON_Reset_PC function initializes the stack pointer (a #pragma entry declaration causes the compiler automatically to generate ISP/USP initialization code at the start of the PowerON_Reset_PC function), sets INTB (set_intb function: embedded function), initializes FPSW (set_fpsw function: embedded function), initializes the RAM area section (_INITSCT function: standard library function), calls the HardwareSetup function, initializes PSW (set_psw function: embedded function), and sets user mode as the processor mode. Then it calls the main function.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

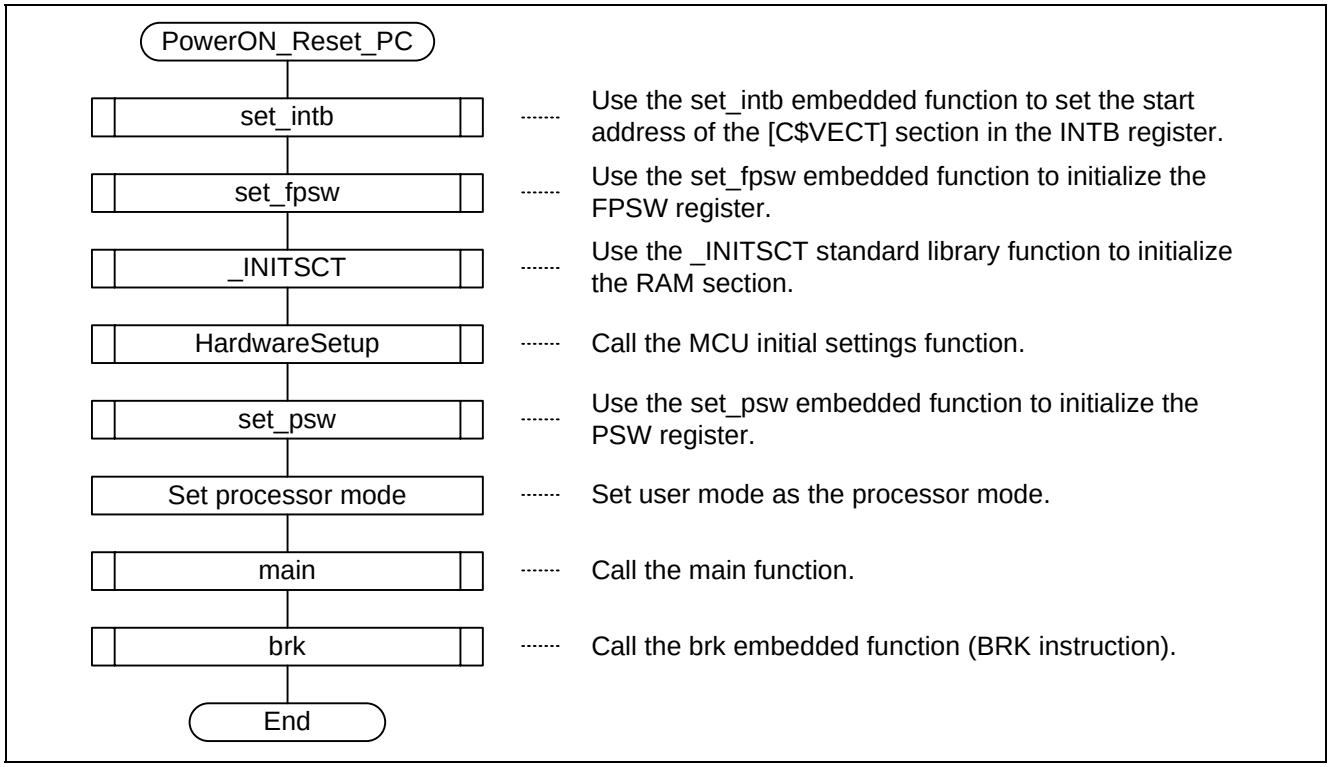


Figure 12 Flowchart (PowerON_Reset_PC) (Master)

(2) HardwareSetup Function

(a) Functional overview

The HardwareSetup function makes initial settings to the MCU. It makes initial clock settings (system clock (ICLK) and peripheral module clock (PCLK)), initial I/O settings for the I/O ports (P71, P72, P73, and P33) connected to LED0 to LED3, the initial I/O port function setting for the pin (PE5/IRQ0-B) connected to the IRQ switch, and initial settings to SCI0.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

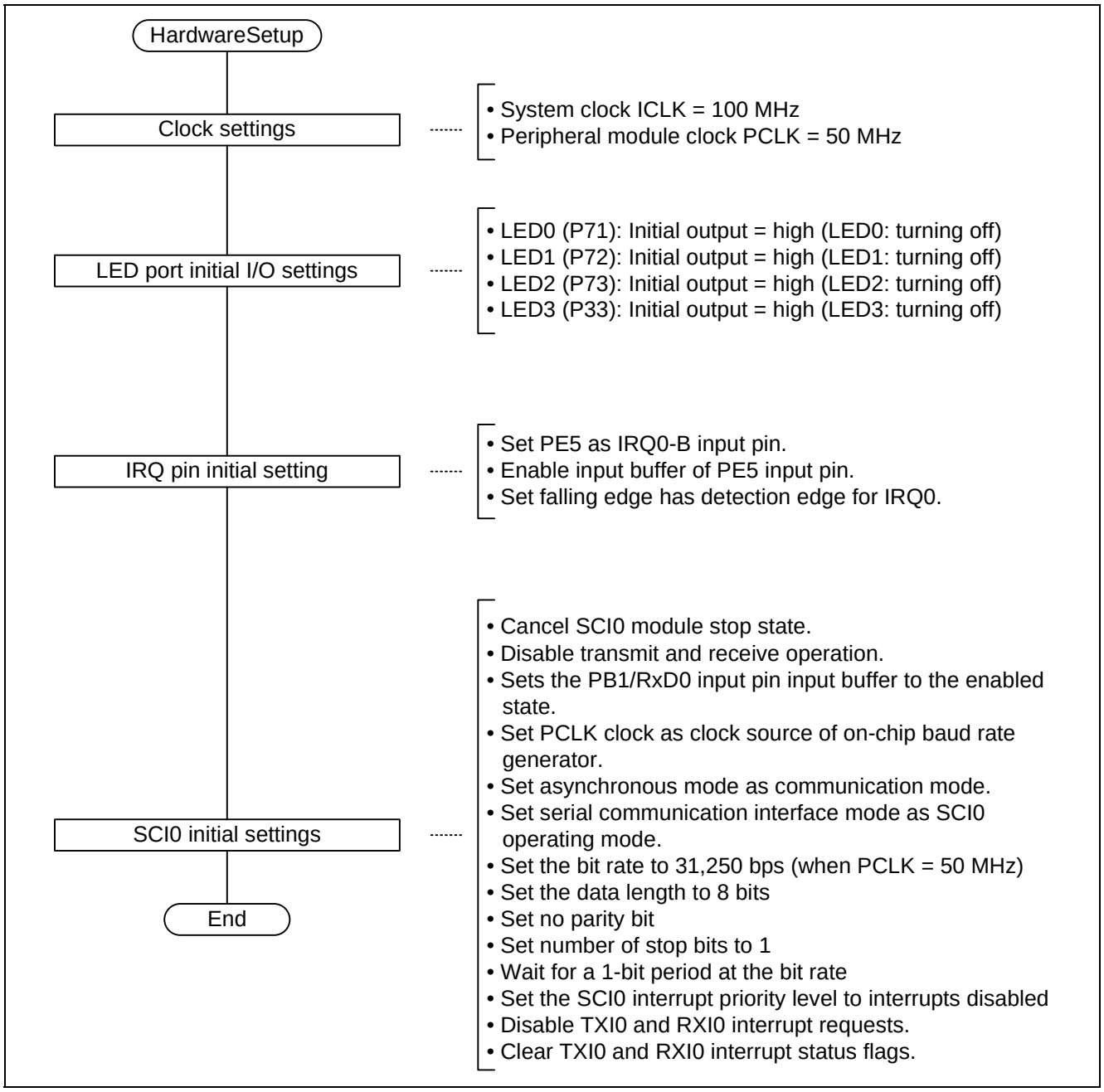


Figure 13 Flowchart (HardwareSetup) (Master)

(3) main Function

(a) Functional overview

The main function determines when the IRQ switch has been pressed, controls transmission and reception of communications commands to and from the slave, controls transmission of erasure block number, controls transmission of write data size, controls transmission of write data, controls reception of [ACCEPTABLE] commands sent from the slave, and calls the Indicate_Ending_LED at normal end.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

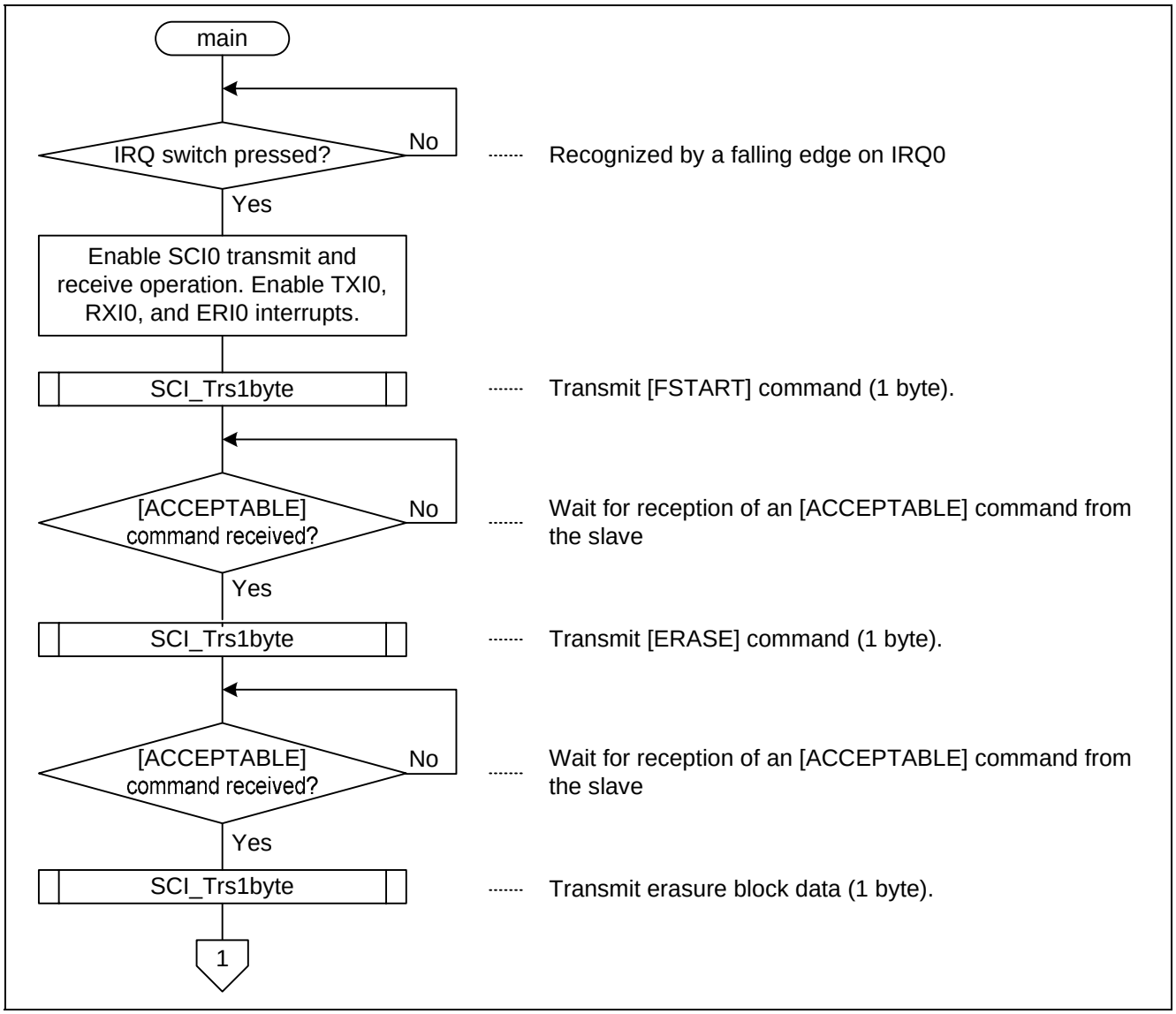


Figure 14 Flowchart (main) (1) (Master)

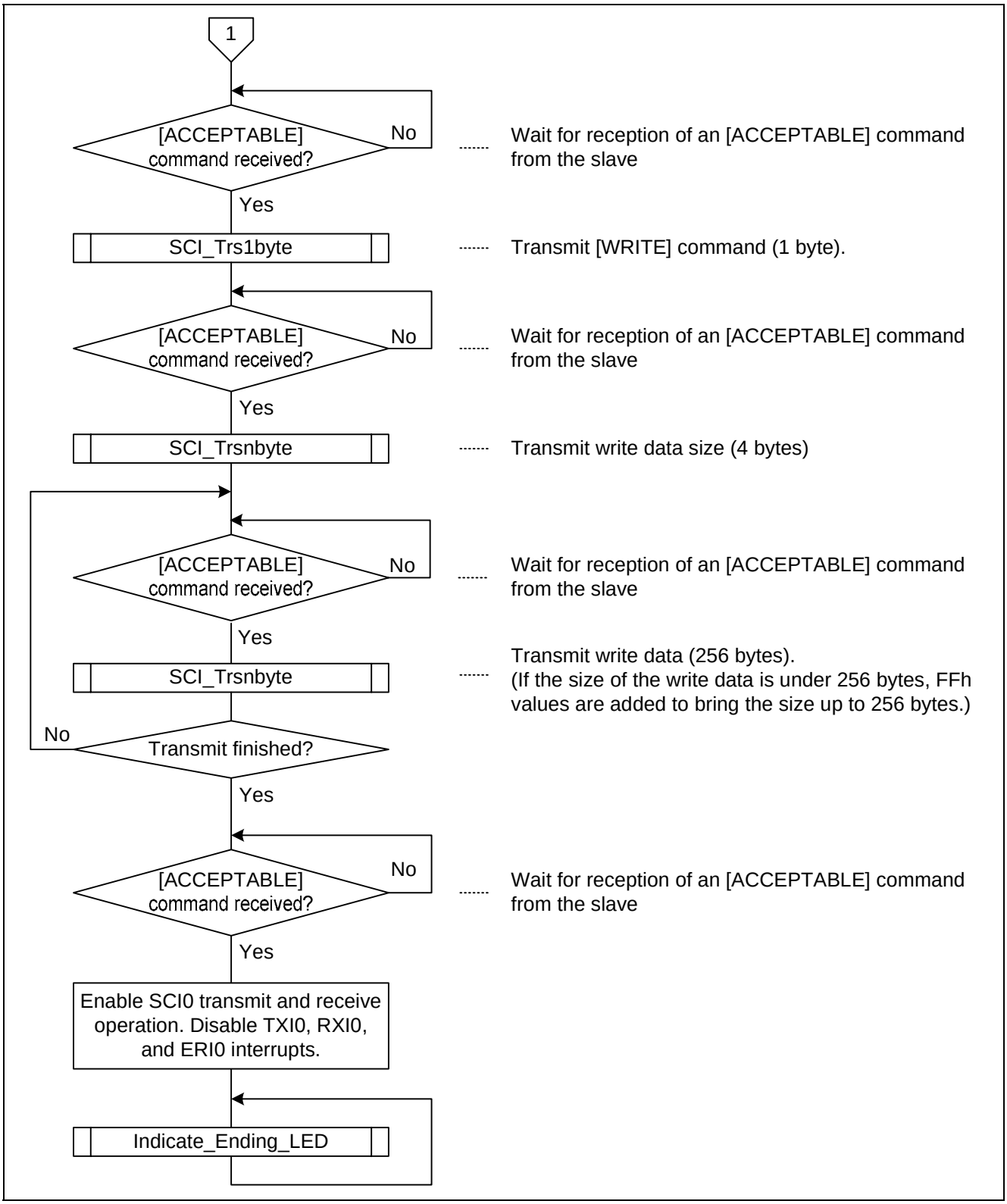


Figure 15 Flowchart (main) (2) (Master)

(4) Indicate_Ending_LED Function

(a) Functional overview

When programing/erasing of the slave's user MAT completes successfully, the Indicate_Ending_LED function indicates a normal end using LED0 to LED3. The function illuminates LED0 to LED3 one at a time in sequence.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

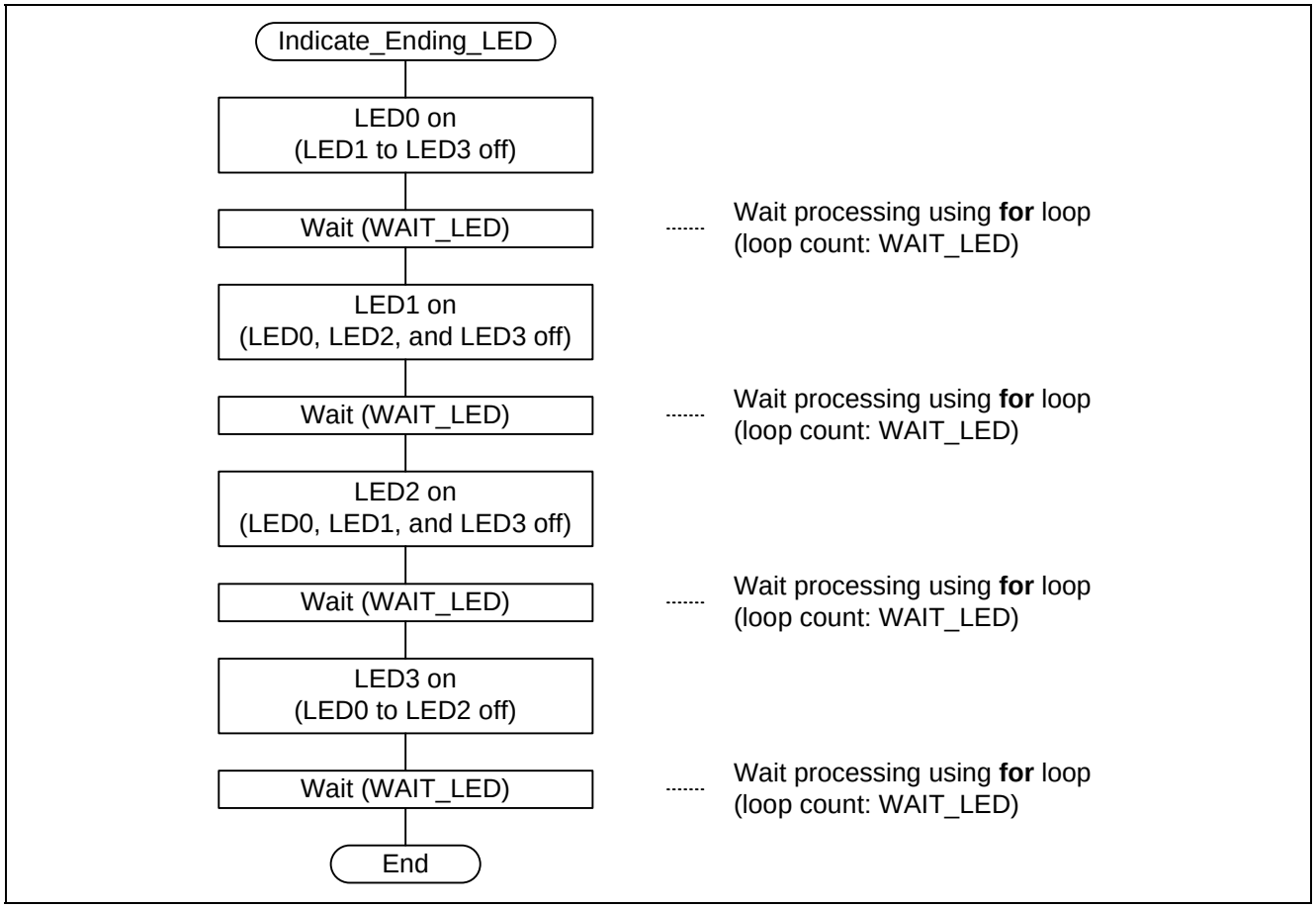


Figure 16 Flowchart (Indicate_Ending_LED) (Master)

(5) SCI_Trs1byte Function

(a) Functional overview

The SCI_Trs1byte function controls transmission of one byte of data using asynchronous serial communication by SCI0.

(b) Arguments

Table 15 lists the arguments used by this function.

Table 15 Arguments of SCI_Trs1byte Function

Arguments	Type	Description
1st argument	unsigned char	Transmit data byte count obtained using asynchronous serial communication by SCI0

(c) Return values

None

(d) Flowchart

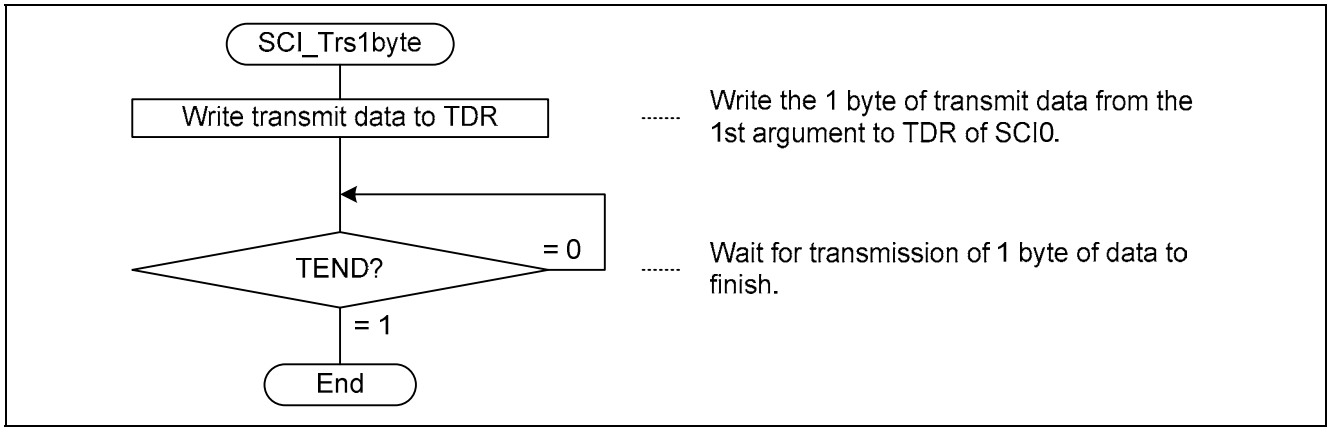


Figure 17 Flowchart (SCI_Trs1byte) (Master)

(6) SCI_Trsnbyte Function

(a) Functional overview

The SCI_Trsnbyte uses the asynchronous serial communication function of the SCI0 to control transmission of n bytes (n is the first argument and unsigned short type).

(b) Arguments

Table 16 lists the arguments used by this function.

Table 16 Arguments of SCI_Trsnbyte Function

Arguments	Type	Description
1st argument	unsigned short	Number of bytes of data transmitted using asynchronous serial communication function of SCI0
2nd argument	unsigned char *	Start address of storage location for transmit data

(c) Return values

None

(d) Flowchart

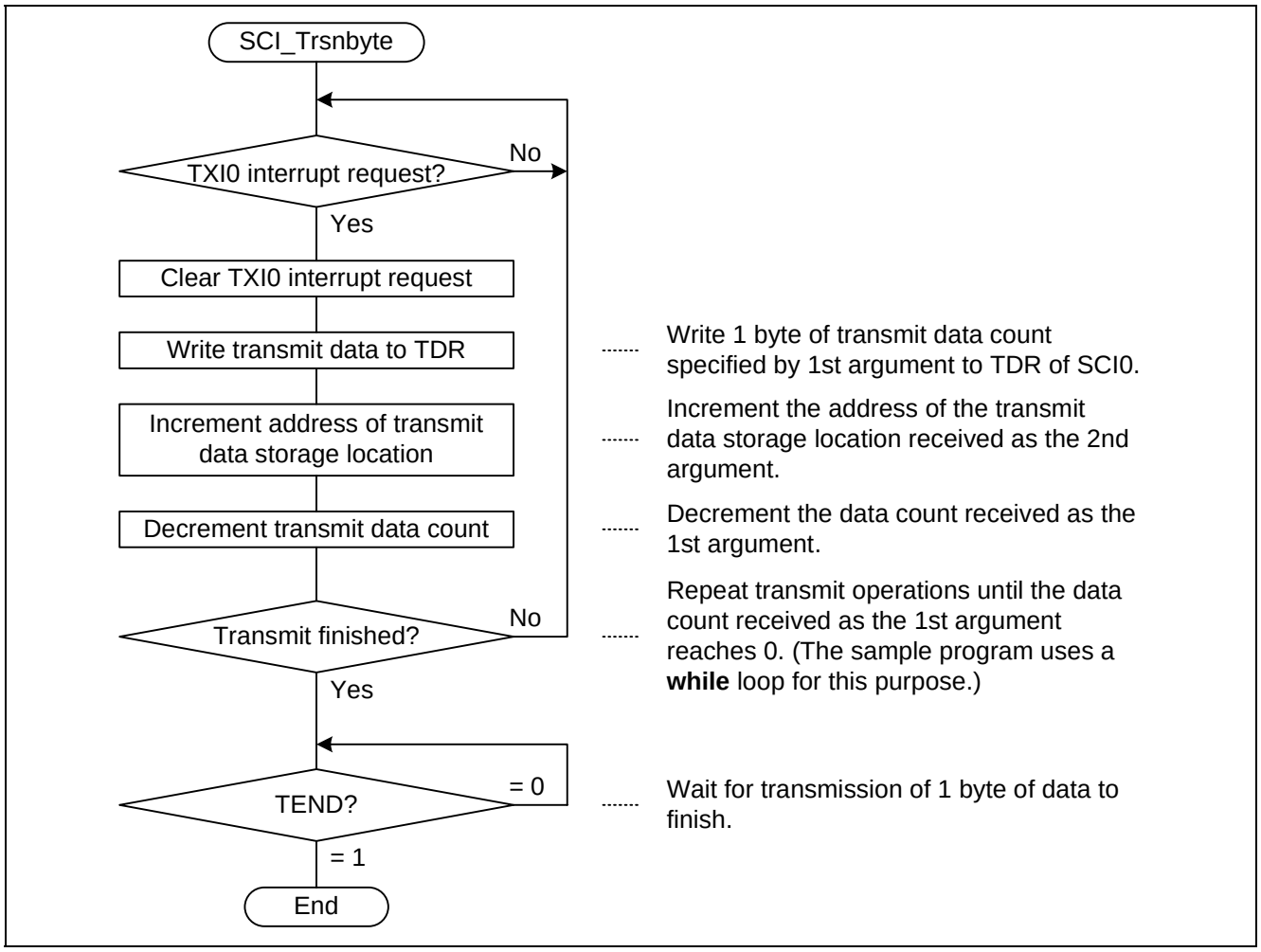


Figure 18 Flowchart (SCI_Trsnbyte) (Master)

(7) SCI_Rcv1byte Function

(a) Functional overview

The SCI_Rcv1byte function performs the reception control for receiving 1 byte of data over SCI0 asynchronous serial communication.

(b) Arguments

None

(c) Return values

Table 17 lists the return values used by this function.

Table 17 Return Values SCI_Rcv1byte Function

Type	Description
unsigned char	The one byte of receive data from the SCI0 asynchronous serial communication.

(d) Flowchart

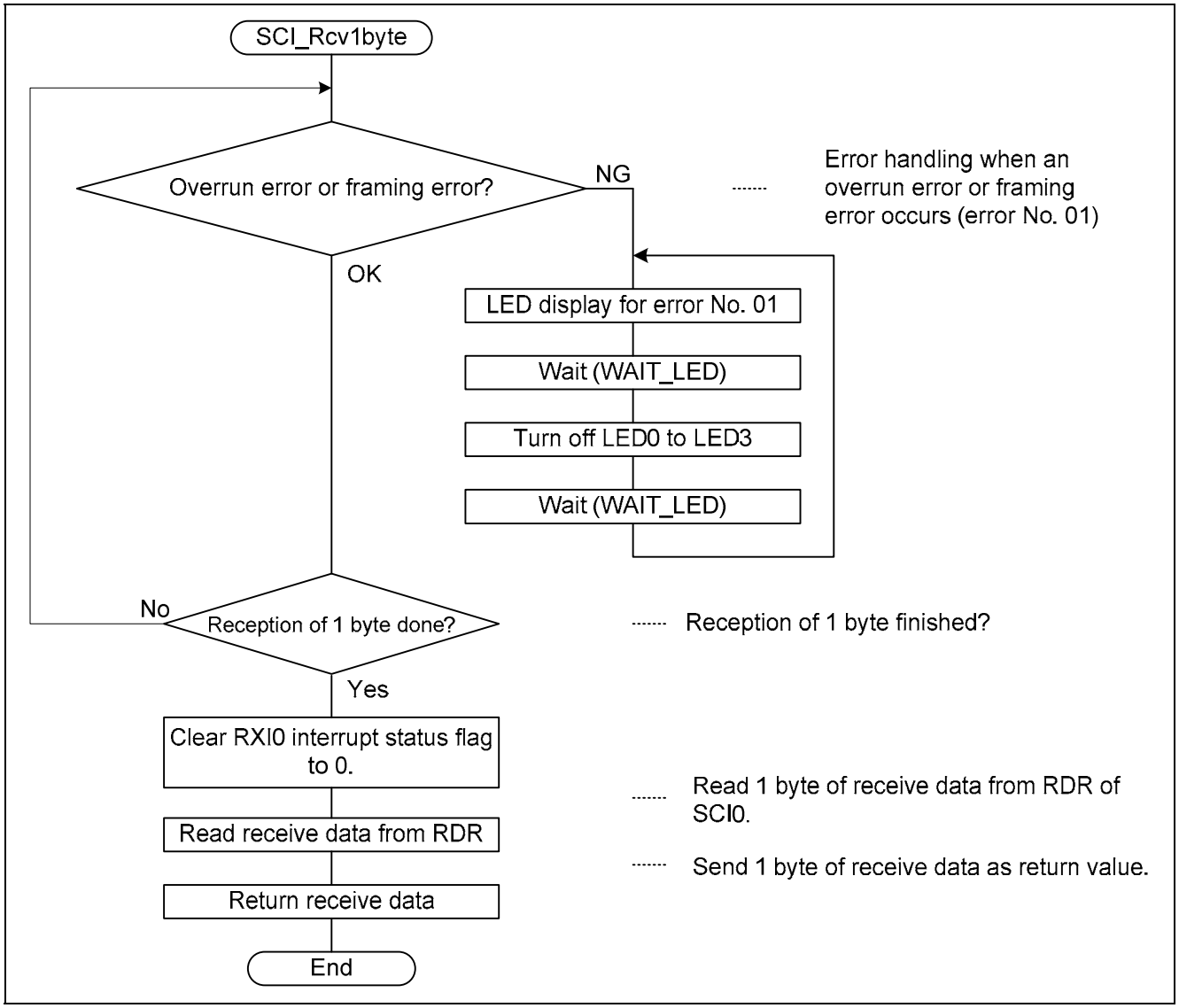


Figure 19 Flowchart (SCI_Rcv1byte) (Master)

6. Usage Notes

6.1 Notes on the wait time for a 1-bit period for the bit rate at SCI0 initialization

In this application note, the 1-bit period wait time for the bit rate after setting the bit rate register (SCI0.BRR) at SCI initialization is measured using a software timer. Since the bit rate for SCI0 asynchronous serial communication is 31,250 bps, the bit period is calculated as follows.

The 1-bit period for the 31,250 bps bit rate is: 32 µs.

In this application note, the 1-bit period wait time for the bit rate is implemented by iterating a while loop with the loop count defined by the WAIT_SCI1BIT symbolic constant. If we take the number of cycles to execute one iteration of the while loop to be 5 cycles (which can be verified from the assembly language output by the compiler), the number of iterations can be calculated as follows.

while loop run count = wait duration / (cycle count per **while** loop iteration * ICLK cycle duration)

Note that the CPU's instruction processing time can differ due to pipelining, so the above-mentioned number of cycles per **while** loop iteration (5 cycles) is a rough estimate of the instruction processing time.

In the sample program, the wait duration is calculated as 96 [µs] to provide a sufficient margin, as follows:

while loop run count = = WAIT_SCI1BIT = 96 [µs] / (5 * 10 [ns]) = 1,920 (ICLK = 100 MHz)

Therefore, the symbolic constant WAIT_ SCI1BIT is defined as 1,920.

To use this application note, users should either carefully evaluate the CPU instruction execution time or use a timer to measure this time.

7. Reference Documents

- User's Manual
  RX62T Group User's Manual: Hardware Rev.1.10
  (The latest version can be downloaded from the Renesas Electronics Web site.)

  RX Family User's Manual: Software Rev.1.00
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Development Environment Manual
  RX Family C/C++ Compiler Package User's Manual Rev.1.01
  (The latest version can be downloaded from the Renesas Electronics Web site.)

- Application Note
  RX62T Group On-chip Flash Memory Reprogramming in Single Chip Mode via an UART Interface (Slave) (R01AN0640EJ)
  (The latest information can be downloaded from the Renesas Electronics Web site.)

  RX600 Series Simple Flash API for RX600 Rev.2.20 (R01AN0544EU)
  (The latest information can be downloaded from the Renesas Electronics Web site.)

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

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep.27.11	—	First edition issued
1.01	Mar.13.12	4	Table 1 Master Verified Operating Environment: evaluation environment updated.
		11	Change to the file name for the RX600 Simple Flash API in section 4.4.3, Erase Block Number, addition of a document number, and change to the title of section 7.
		15	Table 10 Master Section Settings: Setting values modified.
		16	Table 11 Master File Structure: RX600 Simple Flash API description modified.
		33	Change to the version number of the RX600 Simple Flash API in section 7, Reference Documents, and removal of the R01AN0640JJ version number.
		—	HEW workspace generated with the environment from table 1.

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