RX62N Group, RX621 Group

On-chip Flash Memory Reprogramming in the User Boot Mode (Master)

R01AN0185EJ0100
Rev.1.00
Dec 17, 2010

Introduction

This application note, relating to "On-chip Flash Memory Reprogramming in the User Boot Mode (Slave)" (R01AN0184EJ) for the RX62N and RX621 groups, describes transmission of the following pieces of data through clock synchronous serial communication: the block number of a block to be erased, programming data size, and the programming data.

For details on how to program/erase data to/from on-chip flash memory (the user MAT) in the user boot mode, refer to "On-chip Flash Memory Reprogramming in the User Boot Mode (Slave)" (R01AN0184EJ) for the RX62N and RX621 groups.

Target Device

RX62N group, RX621 group

This program is also available for the other RX families that have the similar I/O registers (peripheral device control registers) as the RX62N and RX621 groups. Note, however, that parts of functionalities have been modified or enhanced. Check these changes in the relevant manuals. Extensive evaluation tests should be conducted when using this application note.

Contents

1. Specification

- The master transmits an erase block number, programming data size, and programming data to the slave through clock synchronous serial communication, and the slave performs program/erase operation on its user MAT.
- The clock synchronous communication between the master and slave is accomplished using the SCI channel 2 (SCI2) module.
- The major clock synchronous communication specifications are: 2.4 Mbps bit rate, 8 data bits, and LSB first. The transfer clock is transmitted by the master device.
- Pressing the switch connected to the master's external interrupt pin (IRQ8-A) allows the master to start serial communication to control the slave's user-MAT program/erase operation.
- By using communication commands, the master instructs the slave to erase one of the erase blocks (EB00 to EB37) of the slave's user MAT. In the example given in this application note, EB30 is selected as the block to be erased.
- After the slave clears EB30, the master transmits programming data size (4 bytes) and programming data (8K bytes) to the slave.
- The master and slave use a handshake to control their communications. The slave uses an I/O port to send out an Assert (low) in the busy state and a Negate (high) when the busy state is reset. The master receives the output from the slave via an external interrupt pin (IRQ0-A) and starts the next transmission sequence when a rising edge is input.
- When the slave successfully completes the user MAT erasing/programming process, the four LEDs connected to the master's I/O ports indicate the successful termination.

Figure 1 shows the major specifications relevant to this application note.

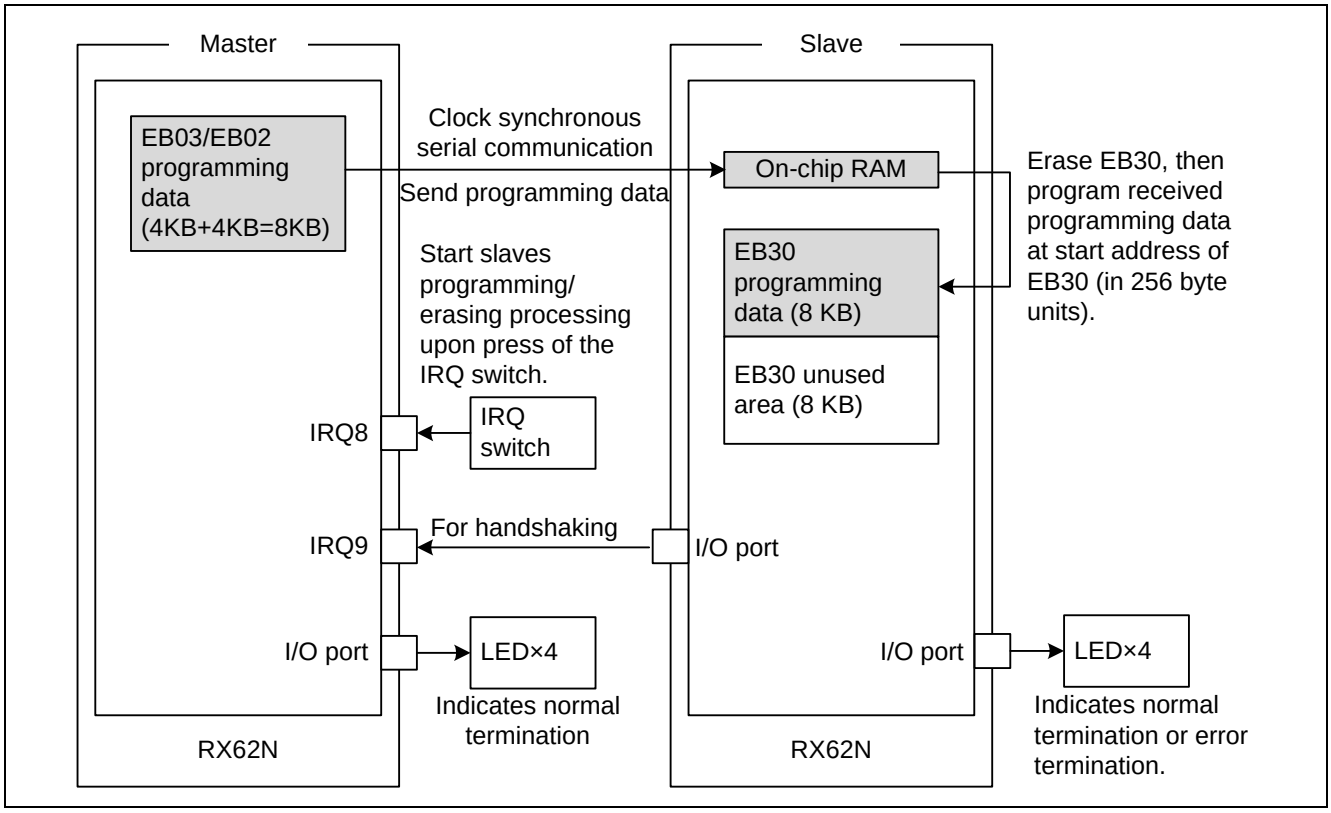

Figure 1 Specification Outline

Figure 2 shows the hardware configuration diagram for the slave device referred to in this application note.

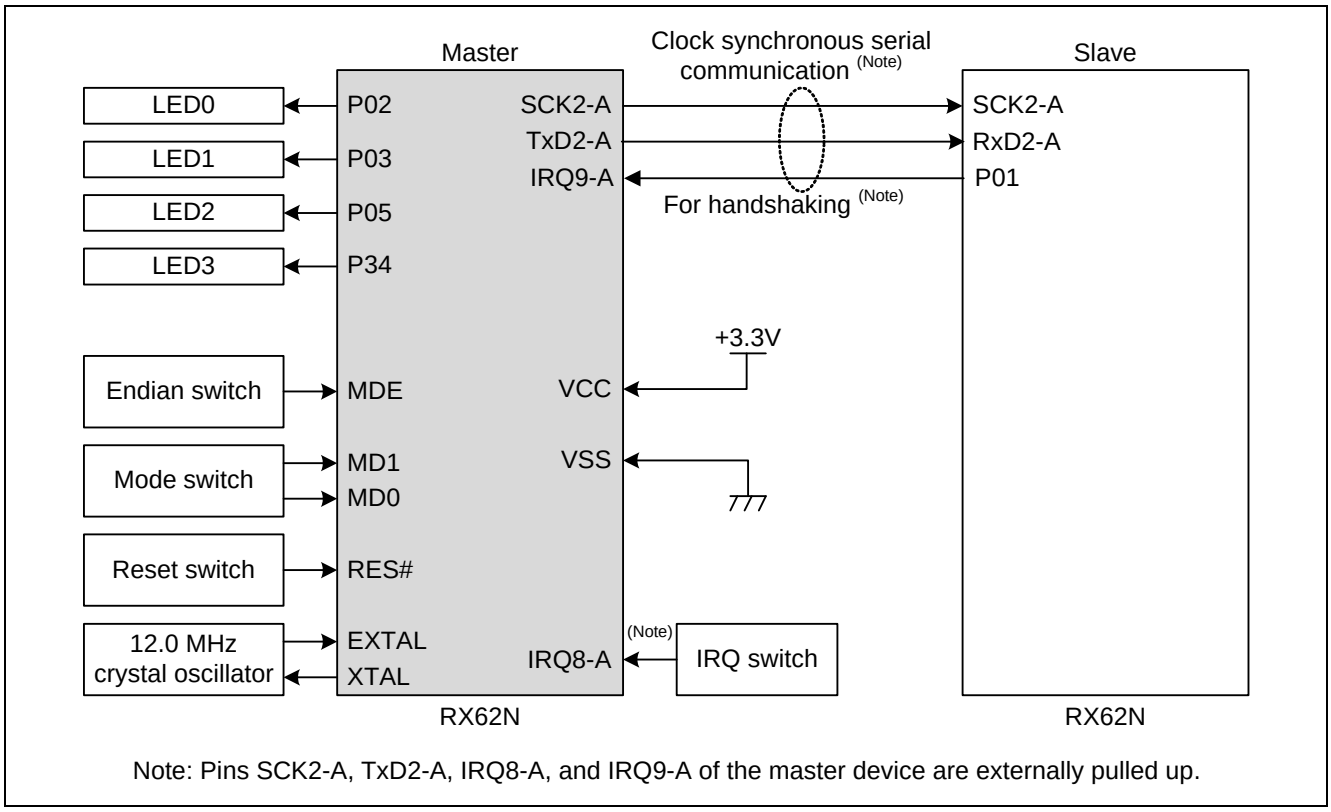

Figure 2 Master Hardware Configuration Diagram

2. Operating Environment

Table 1 summarizes the major characteristics of the environment in which the master is run.

Table 1 Master Operating Environment

Item	Description
Device	RX62N group: R5F562N8BDBG (ROM size: 512 K bytes, RAM size: 96 K bytes)
Board	Renesas starter kit (R0K5562N0S000BE)
Power voltage	5.0 V (CPU operating voltage is 3.3 V.)
Input clock	12.0 MHz (ICLK = 96 MHz, PCLK = 48 MHz, BCLK = 24 MHz)
Operating temperature	Room temperature
High-performance Embedded Workshop	Version 4.07.00.007
Toolchain	RX Standard Toolchain (V.1.0.0.0)
Debugger/Emulator	E20 emulator
Debugger component	RX E1/E20 SYSTEM V.1.00.84.000

3. Functions Used

- Clock Generation Circuit
- Low Power Consumption
- Interrupt Controller Unit (ICU)
- I/O ports
- Serial communication interface

See "User's Manual" listed in section 7. Reference Documents, for details.

4. Description of Operation

4.1 Setting the Operation Mode

In the example given in this application note, the master mode pin MD1 is set to 1 and mode pin MD0 to 1 to set the operating mode to single chip mode and the ROME bit of the system control register 0 (SYSCR0) is set to 1 to enable the on-chip ROM, and the EXBE bit of the SYSCR0 register is set to 0 to disable the external bus.

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

Table 2 summarizes the operating mode settings for the master used in the example given in this application note.

Table 2 Master Operating Mode Settings

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

Note: The SYSCR0 register should never be set up during program execution since the ROME and EXBE bits of the SYSCR0 register are initialized as follows: SYSCR0.ROME = 1, SYSCR0.EXBE = 0

4.2 Setting up the Clocks

The evaluation board used in this application note is provided with a 12.0-MHz crystal oscillator.

Accordingly, the system clock (ICLK), peripheral module clock (PCLK), and external bus clock (BCLK) are set to ×8 (96 MHz), ×4 (48 MHz), and ×2 (24 MHz), respectively, in the example given in this application note.

4.3 Setting up Endian

This application note is compatible with both of big endian and little endian. The endian settings that can be set up by hardware (MDE pin) are listed in table 3. The endian settings of the master and slave must be identical.

Table 3 Endian Settings (Hardware)

MDE Pin	Endian
0	Little endian
1	Big endian

Table 4 lists the endian settings that can be set up using a compiler option.

Table 4 Endian Settings (Compiler Option)

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

Note: Set up the MDE pin according to the endian setting that is selected using the compiler option.

4.4 Clock Synchronous Serial Communication Specifications

In the example given in this application note, clock synchronous serial communication is carried out to transmit communication commands, erase block number, programming data size, and programming data from the master to the slave. The transfer clock is output by the master. The SCK2-A and TxD2-A pins of the SCI2 to be used are externally pulled up.

Table 5 lists the major clock synchronous communication specifications.

Table 5 Clock Synchronous Serial Communication Specifications

Item	Specification
Channel	SCI channel 2 (SCI2)
Communications mode	Clock synchronous mode
Bit rate	2.4 Mbps (at PCLK = 48 MHz)
Direction of data transfer	LSB first

4.4.1 Communication Command Specifications

Table 6 lists the major specifications for the communications commands exchanged between the master and slave.

Table 6 Communication Command Specifications

Command	Code	Description	Direction of Communication
FSTART	10h	Starts the user MAT programming/erasure processing on the slave.	Master → Slave
ERASE	11h	Starts the user MAT erasing processing on the slave.	Master → Slave
WRITE	12h	Starts the user MAT programming on the slave.	Master → Slave

4.4.2 Communication Flows

Figures 3 to 6 show the flows of communications between the master and slave devices.

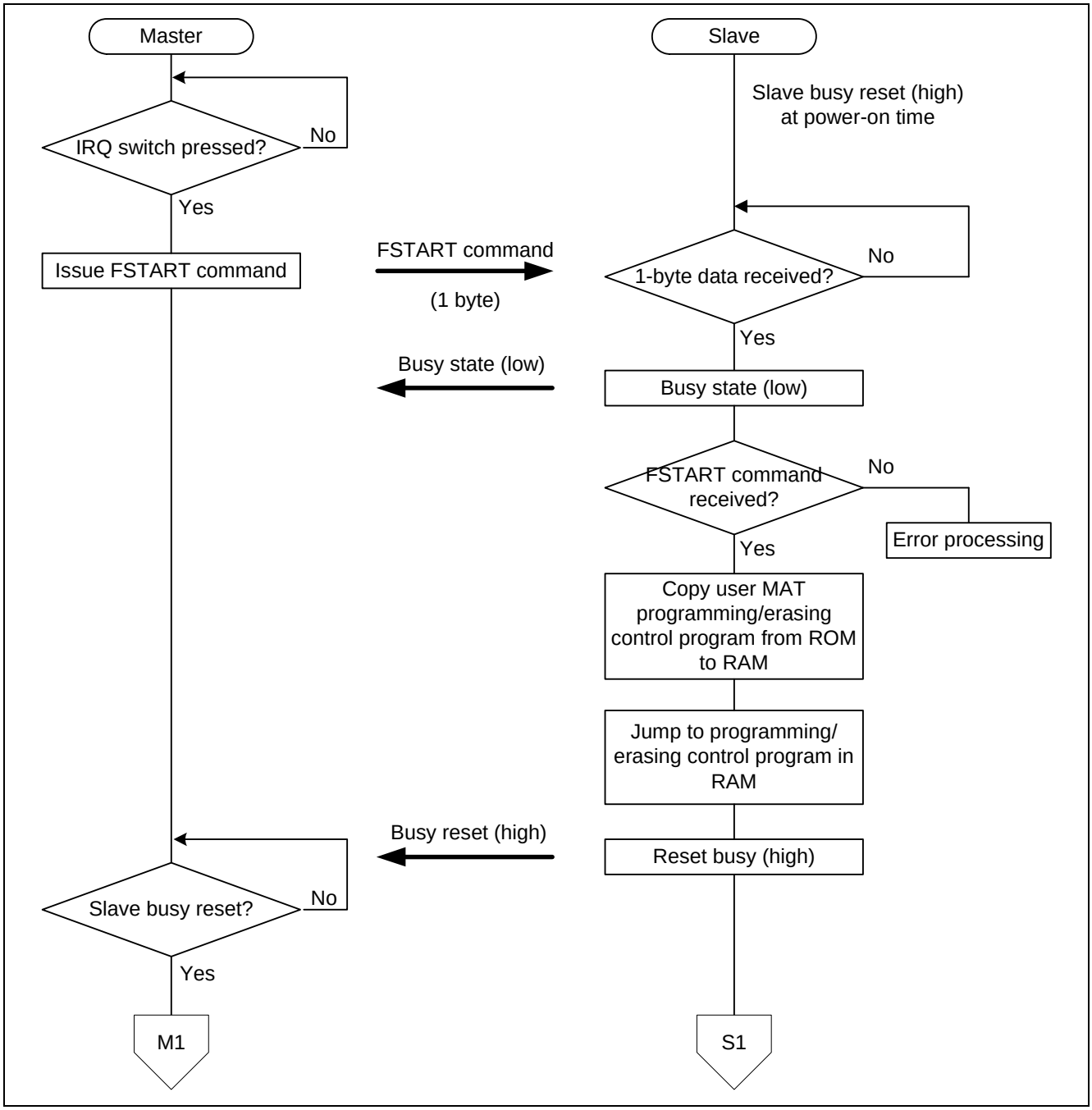

Figure 3 Communications Flow (1)

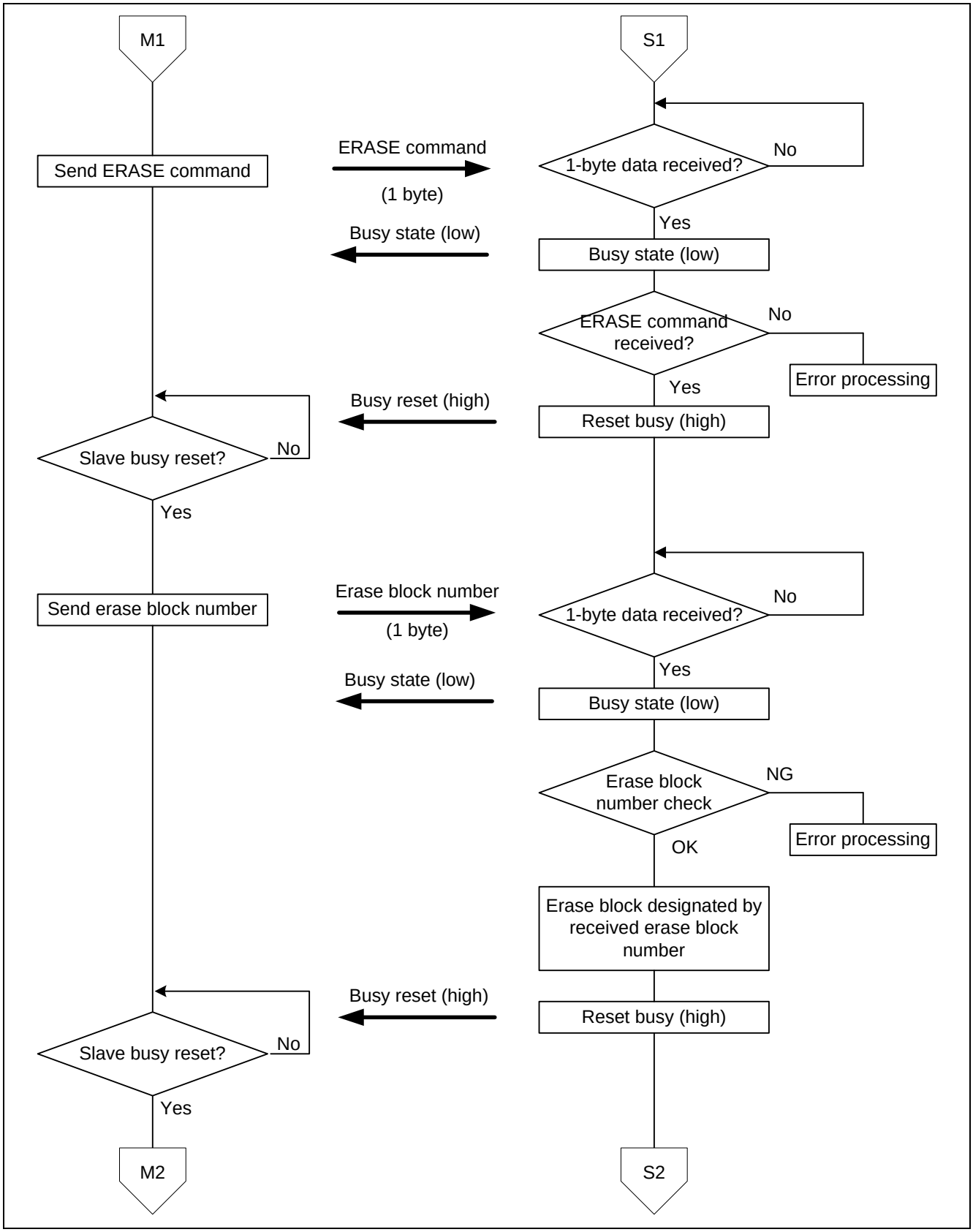

Figure 4 Communications Flow (2)

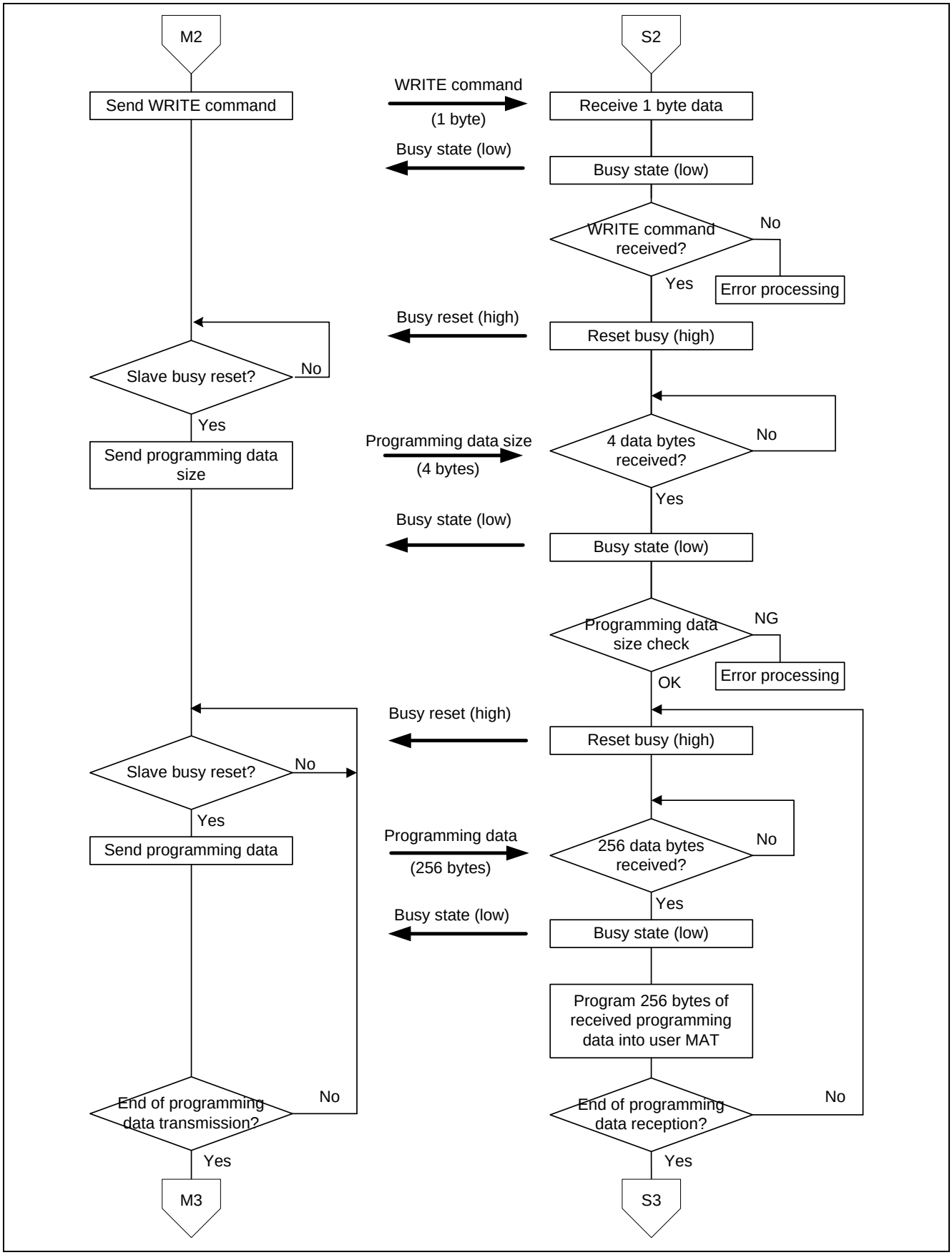

Figure 5 Communications Flow (3)

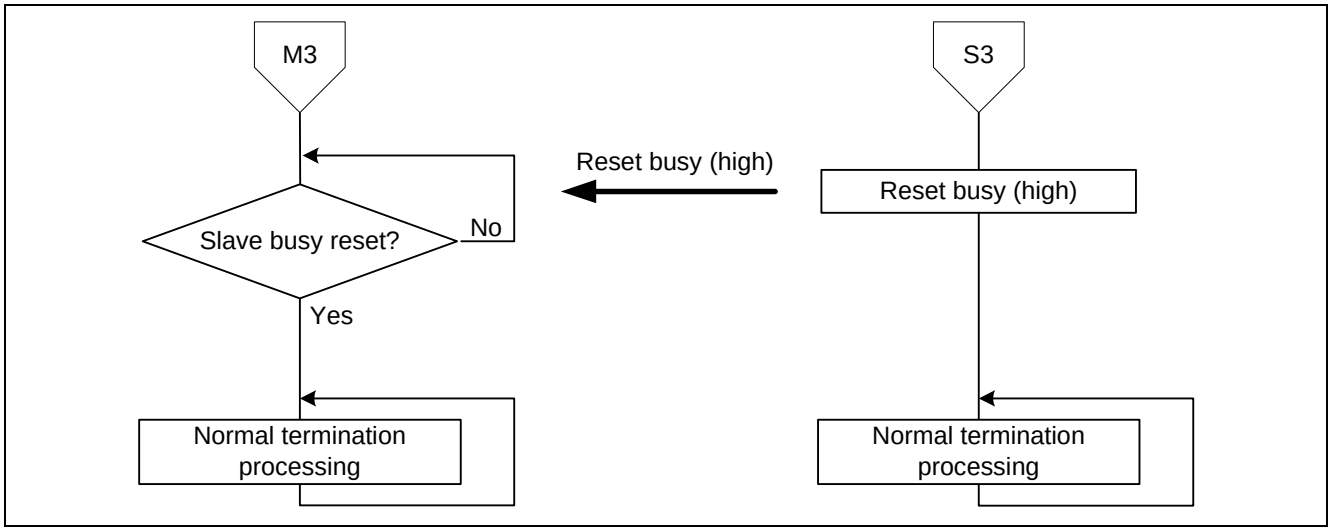

Figure 6 Communications Flow (4)

4.4.3 Erasure Block Number

After transmitting an ERASE command, the master transmits a 1-byte erase block number (1-byte data defined by a symbolic constant). Table 7 gives a list of erase block numbers, and figure 7 shows the major specifications for the erase block numbers.

Table 7 List of Erasure Block Numbers

Erasure Block Number		
Symbolic Constant Name	Value	**Description**
EB37_INDEX	00h	Specifies erase block 37 (size: 16 K bytes).
EB36_INDEX	01h	Specifies erase block 36 (size: 16 K bytes).
EB35_INDEX	02h	Specifies erase block 35 (size: 16 K bytes).
EB34_INDEX	03h	Specifies erase block 34 (size: 16 K bytes).
EB33_INDEX	04h	Specifies erase block 33 (size: 16 K bytes).
EB32_INDEX	05h	Specifies erase block 32 (size: 16 K bytes).
EB31_INDEX	06h	Specifies erase block 31 (size: 16 K bytes).
EB30_INDEX	07h	Specifies erase block 30 (size: 16 K bytes).
EB29_INDEX	08h	Specifies erase block 29 (size: 16 K bytes).
EB28_INDEX	09h	Specifies erase block 28 (size: 16 K bytes).
EB27_INDEX	0Ah	Specifies erase block 27 (size: 16 K bytes).
EB26_INDEX	0Bh	Specifies erase block 26 (size: 16 K bytes).
EB25_INDEX	0Ch	Specifies erase block 25 (size: 16 K bytes).
EB24_INDEX	0Dh	Specifies erase block 24 (size: 16 K bytes).
EB23_INDEX	0Eh	Specifies erase block 23 (size: 16 K bytes).
EB22_INDEX	0Fh	Specifies erase block 22 (size: 16 K bytes).
EB21_INDEX	10h	Specifies erase block 21 (size: 16 K bytes).
EB20_INDEX	11h	Specifies erase block 20 (size: 16 K bytes).
EB19_INDEX	12h	Specifies erase block 19 (size: 16 K bytes).
EB18_INDEX	13h	Specifies erase block 18 (size: 16 K bytes).
EB17_INDEX	14h	Specifies erase block 17 (size: 16 K bytes).
EB16_INDEX	15h	Specifies erase block 16 (size: 16 K bytes).
EB15_INDEX	16h	Specifies erase block 15 (size: 16 K bytes).
EB14_INDEX	17h	Specifies erase block 14 (size: 16 K bytes).
EB13_INDEX	18h	Specifies erase block 13 (size: 16 K bytes).
EB12_INDEX	19h	Specifies erase block 12 (size: 16 K bytes).
EB11_INDEX	1Ah	Specifies erase block 11 (size: 16 K bytes).
EB10_INDEX	1Bh	Specifies erase block 10 (size: 16 K bytes).
EB09_INDEX	1Ch	Specifies erase block 09 (size: 16 K bytes).
EB08_INDEX	1Dh	Specifies erase block 08 (size: 16 K bytes).
EB07_INDEX	1Eh	Specifies erase block 07 (size: 4 K bytes).
EB06_INDEX	1Fh	Specifies erase block 06 (size: 4 K bytes).
EB05_INDEX	20h	Specifies erase block 05 (size: 4 K bytes).
EB04_INDEX	21h	Specifies erase block 04 (size: 4 K bytes).
EB03_INDEX	22h	Specifies erase block 03 (size: 4 K bytes).
EB02_INDEX	23h	Specifies erase block 02 (size: 4 K bytes).
EB01_INDEX	24h	Specifies erase block 01 (size: 4 K bytes).
EB00_INDEX	25h	Specifies erase block 00 (size: 4 K bytes).

Erase Block Number (unsigned char type)

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

This application note assumes an erase block data of EB30_INDEX (07h) for the slave to program or erase the erase block EB30.

Note: Specify erase block numbers between EB37_INDEX (00h) and EB00_INDEX (25h) which are listed in table 7. If an erase block number 26h to FFh is specified, the slave will signal an error and perform error processing.

Figure 7 Erasure Block Number Specifications

4.4.4 Programming Data Size

The master transmits 4 bytes of programming data size data after transmitting the WRITE command. Figure 8 shows the major specifications for the programming data size.

Programming 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

This application note assumes a programming data size of 0000 2000h since the programming size of block data is set to 8 K bytes.

Notes: 1. The programming data size must be greater than 0 but not greater than the size of the erase block designated by the erase block number. If a 0 is specified or a size value greater than the size of the erase block designated by the erase block number is specified, the slave will signal an error and perform error processing.

2. The size of programming data that is to be transmitted is fixed at 256 bytes. If the size of the programming data is not a multiple of 256 bytes, the master sends to the slave device 256 bytes in every transmission operation with the last data block, which is less than 256 bytes long, padded with FFh bytes to make up a 256-byte programming data block.

Figure 8 Programming Data Size Specifications

4.5 Normal Termination Processing

The master indicates a normal termination condition using the four LEDs connected to the I/O port, when the slave successfully completes programming/erasure processing on the user MAT. On normal termination, LED0 to LED3 are turned on sequentially and repeatedly, one at a time.

4.6 LED Cabling

Figure 9 shows the cabling diagram for LED0 to LED3 that are connected to I/O ports of the master device.

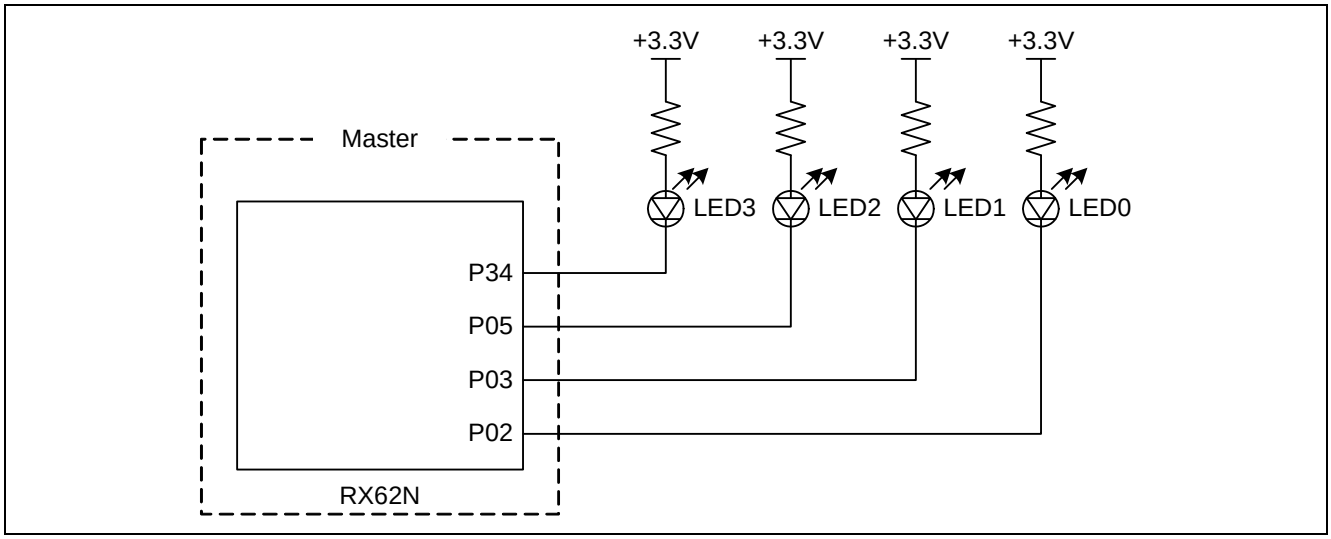

Figure 9 Master LED Cabling Diagram

As seen from figure 9, LED0 to LED3 turn off when the I/O ports (P02, P03, P05, and P34) are set high and on when the I/O ports are set low. Table 8 shows the relationship between the I/O port outputs and LED states.

Table 8 Master I/O Port Outputs and LED States

I/O Port	Register Setting	I/O Port State	LED State	
P02	PORT0.DR.B2 = 1, PORT0.DDR.B2 = 1	High output	LED0	Off
	PORT0.DR.B2 = 0, PORT0.DDR.B2 = 1	Low output		On
P03	PORT0.DR.B3 = 1, PORT0.DDR.B3 = 1	High output	LED1	Off
	PORT0.DR.B3 = 0, PORT0.DDR.B3 = 1	Low output		On
P05	PORT0.DR.B5 = 1, PORT0.DDR.B5 = 1	High output	LED2	Off
	PORT0.DR.B5 = 0, PORT0.DDR.B5 = 1	Low output		On
P34	PORT3.DR.B4 = 1, PORT3.DDR.B4 = 1	High output	LED3	Off
	PORT3.DR.B4 = 0, PORT3.DDR.B4 = 1	Low output		On

4.7 IRQ Switch

A connection diagram of the IRQ switch connected to the master's external interrupt pin (IRQ8-A) is shown in figure 10.

Pressing the IRQ switch connected to the master starts the slave performing programming/erasing processing on the user MAT.

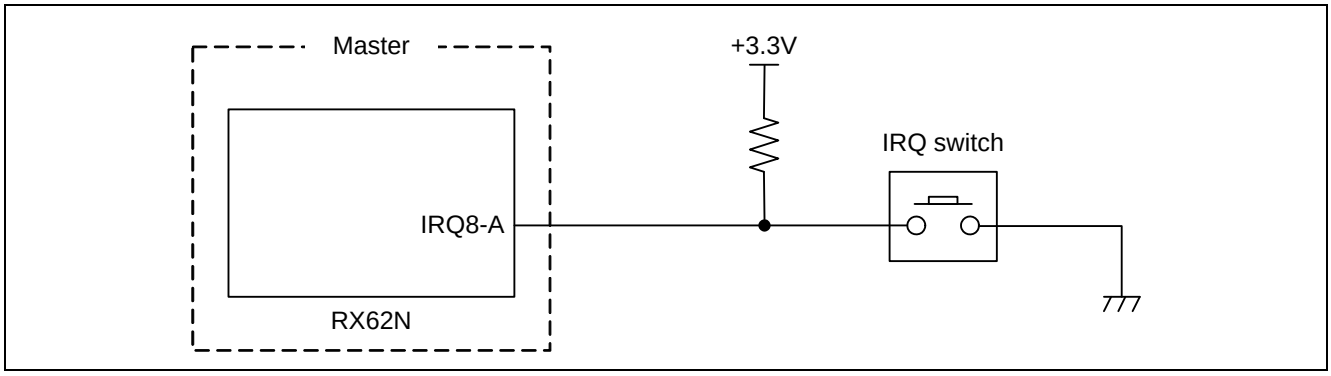

Figure 10 Master IRQ Switch Connection Diagram

By detecting a falling edge on the IRQ8-A pin, the master becomes aware of the IRQ switch being pressed. Interrupt handling is not performed, and whether the IRQ switch is pressed is determined by detecting that the interrupt status flag (IR072.IR) for IRQ8 is set to 1.

4.8 Handshake Control

The master performs a handshake with the slave to control communication. Output signals from the slave's Busy port are input to the external interrupt pin (IRQ9-A).

To control the handshake, after serial transmission, the master waits until the slave's Busy port is negated (high). The master detects a rising edge caused by negating the slave's Busy port (high), and starts the next serial transmission.

4.9 Section Settings

The section settings for the master device are listed in table 9.

Table 9 Master Section Settings

Section Name	Start Address	Description
B	0000 1000h	Uninitialized data area (ALIGN = 4)
R		RAM area in which the [D] section is mapped by the ROMization support option.
SU		User stack area
SI		Interrupt stack area
CP_DATA_1	FFFF C000h	Constant area (ALIGN = 1) (Programming data (8 Kbytes))
PResetPRG	FFFF E000h	Program area (PowerON_Reset_PC program)
C	FFFF E100h	Constant area (ALIGN = 4)
C$DSEC		Table for initializing the sections in the initialized data area
C$BSEC		Table for initializing the sections in the uninitialized data area
C$VECT		Variable vector area
D		Initialized data area (ALIGN = 4)
P		Program area
PIntPRG		Program area (interrupt program)
FIXEDVECT	FFFF FFD0h	Fixed vector area

5. Software Description

5.1 File Organization

The file organization of the master device is summarized in table 10. For the files that are not listed in table 10, files that are automatically generated by HEW are used.

Table 10 Master File Organization

File Name	Description
resetprg.c (*[1])	Performs initialization.
main.c	Main processing, communication command transmission through clock synchronous serial communication with the slave, transmission of erase block number, programming data size, and programming data, LED indication at normal termination.

Note: *1 A file automatically generated by the High-performance Embedded Workshop, whose commented out code for calling the HardwareSetup function in the PowerON_Reset_PC function is re-enabled so that the HardwareSetup function in the main.c file can be called from the PowerON_Reset_PC function.

5.2 Functions

A list of functions available for the master device is given in table 11 and the function hierarchy of the master functions in figure 11

Table 11 List of Functions for the Master

Function Name	File Name	Outline
PowerON_Reset_PC	resetprg.c	Initialization function
HardwareSetup	main.c	MCU initialization function
main	main.c	Main function
Indicate_Ending_LED	main.c	Normal termination processing function
SCI_Trs1byte	main.c	1-byte data transmit function
SCI_Trsnbyte	main.c	n-byte data transmit function

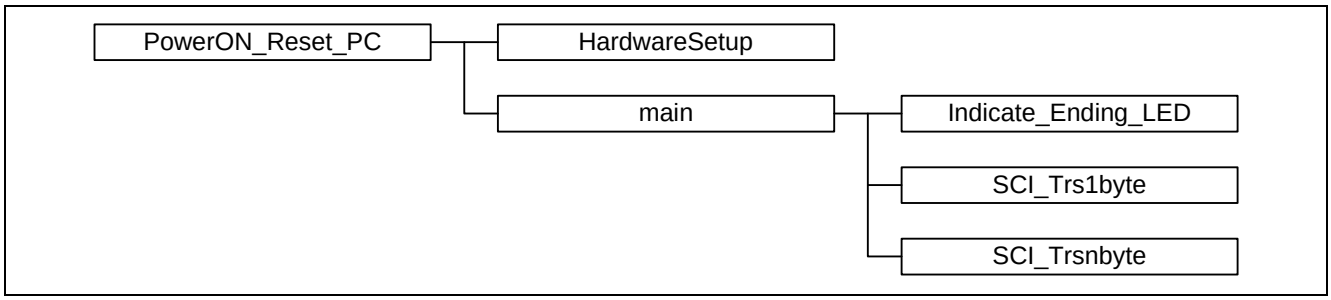

Figure 11 Master Function Hierarchical Diagram

5.3 Symbolic Constant Description

Table 12 lists the symbolic constants that are to be used by the master device.

Table 12 List of Master's Symbolic Constants

Symbolic Constant Name	Setting	Description	Used In
FSTART	0x10	Programming/erasure start command	main
ERASE	0x11	Erasure start command	main
WRITE	0x12	Programming start command	main
LED_ON	0	LED on time value	Indicate_Ending_LED
LED_OFF	1	LED off time value	HardwareSetup Indicate_Ending_LED
RSK_LED0	PORT0.DR.BIT.B2	Evaluation board mounted LED0 on/off control	HardwareSetup Indicate_Ending_LED
RSK_LED1	PORT0.DR.BIT.B3	Evaluation board mounted LED1 on/off control	HardwareSetup Indicate_Ending_LED
RSK_LED2	PORT0.DR.BIT.B5	Evaluation board mounted LED2 on/off control	HardwareSetup Indicate_Ending_LED
RSK_LED3	PORT3.DR.BIT.B4	Evaluation board mounted LED3 on/off control	HardwareSetup Indicate_Ending_LED
RSK_LED0_DDR	PORT0.DDR.BIT.B2	Evaluation board mounted LED0 input/output control	HardwareSetup
RSK_LED1_DDR	PORT0.DDR.BIT.B3	Evaluation board mounted LED1 input/output control	HardwareSetup
RSK_LED2_DDR	PORT0.DDR.BIT.B5	Evaluation board mounted LED2 input/output control	HardwareSetup
RSK_LED3_DDR	PORT3.DDR.BIT.B4	Evaluation board mounted LED3 input/output control	HardwareSetup
FALL_EDGE	1	Falling edge setting	HardwareSetup
RISE_EDGE	2	Rising edge setting	HardwareSetup
SW_ON	1	The START_SW_IR value when the IRQ switch is turned on	—
SW_OFF	0	The START_SW_IR value when the IRQ switch is turned off	HardwareSetup
START_SW_IR	ICU.IR[IR_ICU_IRQ8].BIT.IR	IRQ switch state	main
START_SW_PFC	IOPORT.PF8IRQ.BIT.ITS8	Pin selection for IRQ switch	HardwareSetup
START_SW_ICR	PORT0.ICR.BIT.B0	Input buffer setting for IRQ switch	HardwareSetup
START_SW_IRQMD	ICU.IRQCR[8].BIT.IRQMD	Detection setting for IRQ switch	HardwareSetup
ASSERT	0	Setting when the Busy port is asserted	main
NEGATE	1	Setting when the Busy port is negated	main
SLAVE_BUSY_IR	ICU.IR[IR_ICU_IRQ9].BIT.IR	Busy port signal state	HardwareSetup main
SLAVE_BUSY_PFC	IOPORT.PF8IRQ.BIT.ITS9	Pin selection for Busy port signal	HardwareSetup
SLAVE_BUSY_ICR	PORT0.ICR.BIT.B1	Input buffer setting for Busy port signal	HardwareSetup
SLAVE_BUSY_IRQMD	ICU.IRQCR[9].BIT.IRQMD	Detection setting for Busy port signal	HardwareSetup

Table 12 List of Master's Symbolic Constants (Continued)

Symbolic Constant Name	Setting	Description	Used In
EB37_INDEX	0x00	Erase block number to be sent to designate the erase block to be programmed or erased by the slave.	main
EB36_INDEX	0x01		
EB35_INDEX	0x02		
EB34_INDEX	0x03		
EB33_INDEX	0x04		
EB32_INDEX	0x05		
EB31_INDEX	0x06		
EB30_INDEX	0x07		
EB29_INDEX	0x08		
EB28_INDEX	0x09		
EB27_INDEX	0x0A		
EB26_INDEX	0x0B		
EB25_INDEX	0x0C		
EB24_INDEX	0x0D		
EB23_INDEX	0x0E		
EB22_INDEX	0x0F		
EB21_INDEX	0x10		
EB20_INDEX	0x11		
EB19_INDEX	0x12		
EB18_INDEX	0x13		
EB17_INDEX	0x14		
EB16_INDEX	0x15		
EB15_INDEX	0x16		
EB14_INDEX	0x17		
EB13_INDEX	0x18		
EB12_INDEX	0x19		
EB11_INDEX	0x1A		
EB10_INDEX	0x1B		
EB09_INDEX	0x1C		
EB08_INDEX	0x1D		
EB07_INDEX	0x1E		
EB06_INDEX	0x1F		
EB05_INDEX	0x20		
EB04_INDEX	0x21		
EB03_INDEX	0x22		
EB02_INDEX	0x23		
EB01_INDEX	0x24		
EB00_INDEX	0x25		
WAIT_SCI1BIT	23	Wait time after the BRR register in SCI2 is set	HardwareSetup
WAIT_LED	2000000	Time of an interval between LED turning on and off when the slave successfully completes program/erase operation on the user MAT.	Indicate_Ending_LED
TRS_SIZE	256	Transmission size for programming data	main
BUF_SIZE	8192	Size of programming buffer	main
WRITE_SIZE	BUF_SIZE	Size of the area to store programming data	main

5.4 const Variable Description

Table 13 lists the const variable that is to be used by the master device.

Table 13 List of Master const Variables

Variable Name	Type	Description
SAMPLE_DATA[BUF_SIZE]	const unsigned char	Data to be transmitted to the slave and written to the user MAT (8192 bytes) In the example given in this application note, SAMPLE_DATA[BUF_SIZE] is allocated to the CP_DATA_1 section, corresponding to the erase blocks EB03 to EB02 (FFFF C000h to FFFF DFFFh).

5.5 RAM Variable Description

In the example given in this application note, there are no RAM variables to be used by the master's user program.

5.6 Description of the I/O Registers Used

This section describes the I/O registers that are used by the program on the master device. The settings that are described in this document are those values which are used in the example program given in this application note; they differ from their initialized values.

(1) Clock Generation Circuit

- System clock control register (SCKCR) Number of bits: 32 bits Address: 0008 0020h

Bit	Symbol	Setting	Bit Name	Description	R/W
b11-b8	PCK[3:0]	0001	Peripheral module clock(PCLK) select	0001: ×4 PCLK = 48 MHz (when EXTAL clock frequency = 12.0 MHz)	R/W
b19-b16	BCK[3:0]	0010	External bus clock (BCLK) select	0010: ×2 BCLK = 24 MHz (when EXTAL clock frequency = 12.0 MHz)	R/W
b23	PSTOP1	0	BCLK output stop	0: BCLK output	R/W
b27-b24	ICK[3:0]	0000	System clock (ICLK) select	0000: ×8 ICLK = 96 MHz (when EXTAL clock frequency = 12.0 MHz)	R/W

(2) I/O ports

- Port 0 data register (P0.DR) Number of bits: 8 bits Address: 0008 C020h

Bit	Symbol	Setting	Bit Name	Description	R/W
b2	B2	0	P02 output data	0: Output data = 0	R/W
		1		1: Output data = 1	
b3	B3	0	P03 output data	0: Output data = 0	R/W
		1		1: Output data = 1	
b5	B5	0	P05 output data	0: Output data = 0	R/W
		1		1: Output data = 1	

- Port 3 data register (P3.DR) Number of bits: 8 bits Address: 0008 C023h

Bit	Symbol	Setting	Bit Name	Description	R/W
b4	B4	0	P34 output data	0: Output data = 0	R/W
		1		1: Output data = 1	

- Port 0 data direction register (P0.DDR) Number of bits: 8 bits Address: 0008 C000h

Bit	Symbol	Setting	Bit Name	Description	R/W
b2	B2	1	P02 input/output select	1: Output port	R/W
b3	B3	1	P03 input/output select	1: Output port	R/W
b5	B5	1	P05 input/output select	1: Output port	R/W

- Port 3 data direction register (P3.DDR) Number of bits: 8 bits Address: 0008 C003h

Bit	Symbol	Setting	Bit Name	Description	R/W
b4	B4	1	P34 input/output select	1: Output port	R/W

- Port function control register 8 (PF8IRQ) Number of bits: 8 bits Address: 0008 C108h

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	ITS8	0	IRQ8 pin select	0: P00 is designated as the IRQ8-A input pin	R/W
b1	ITS9	0	IRQ9 pin select	0: P01 is designated as the IRQ9-A input pin	R/W

- Port function control register F (PFFSCI) Number of bits: 8 bits Address: 0008 C10Fh

Bit	Symbol	Setting	Bit Name	Description	R/W
b2	SCI2S	0	SCI2 pin select	0: P12 is designated as the RxD2-A pin P11 is designated as the SCK2-A pin P13 is designated as the TxD2-A pin	R/W

- Port 0 input buffer control register (P0.ICR) Number of bits: 8 bits Address: 0008 C060h9

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	B0	1	P00 input buffer control	1: Enables the input buffer for P00	R/W
b1	B1	1	P01 input buffer control	1: Enables the input buffer for P01	R/W

(3) Low Power Consumption

- Module stop control register B (MSTPCRB) Number of bits: 32 bits Address: 0008 0014h

Bit	Symbol	Setting	Bit Name	Description	R/W
b29	MSTPB29	0	Serial communications interface 2 module stop	0: The SCI2 module stop state is canceled.	R/W

(4) Serial communication interface 2 (SCI2)

- SCI2 serial control register (SCI2.SCR) Number of bits: 8 bits Address: 0008 8252h
  (In serial communications interface mode (SCI2.SCMR.SMIF bit = 0))

Bit	Symbol	Setting	Bit Name	Description	R/W
b1-b0	CKE[1:0]	00	Clock enable	(Clock synchronous mode) 00: On-chip clock The SCK2 pin is configured for clock output.	R/W (*1)
b2	TEIE	0	Transmit end interrupt enable	0: TEI2 interrupt requests are disabled.	R/W
b5	TE	0	Transmit enable	0: Serial transmission is disabled	R/W (*2)
		1		1: Serial transmission is enabled	
b7	TIE	0	Transmit interrupt enable	0: TXI2 interrupt requests are disabled	R/W
		1		1: TXI2 interrupt requests are enabled	

Notes: *1 Writable only when TE = 0 and RE = 0.
*2 A 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

- SCI2 serial mode register (SCI2.SMR) Number of bits: 8 bits Address: 0008 8250h
  (In serial communication interface mode (SCI2.SCMR.SMIF bit = 0))

Bit	Symbol	Setting	Bit Name	Description	R/W
b1-b0	CKS[1:0]	00	Clock select	00: PCLK clock (n = 0) (*1)	R/W (*2)
b7	CM	1	Communications mode	1: Run in clock synchronous mode.	R/W (*2)

Notes: *1 See "User's Manual" listed in section 7, Reference Documents, for the value of n.
*2 Writable only when SCI2.SCR.TE = 0 and SCI2.SCR.RE = 0 (serial transmission is disabled and serial reception is disabled).

- SCI2 smart card mode register (SCI2.SCMR) Number of bits: 8 bits Address: 0008 8256h

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	SMIF	0	Smart card interface mode select	0: Serial communications interface mode	R/W (*1)
b3	SDIR	0	Smart card data transfer direction	0: Transmitted/received in LSB first mode.	R/W (*1)

Note: *1 Writable only when SCI2.SCR.TE = 0 and SCI2.SCR.RE = 0 (serial transmission is disabled and serial reception is disabled).

- SCI2 bit rate register (SCI2.BRR) (*1) Number of bits: 8 bits Address: 0008 8251h

Bit	Symbol	Setting	Bit Name	Description	R/W
b7-b0	—	00000100	—	04h: Bit rate = 2.4 M bps (when PCLK = 48 MHz)	R/W (*2)

Notes: *1 See "User's Manual" listed in section 7, Reference Documents, for the settings of BRR.
*2 Always readable. Writable only when SCI2.SCR.TE = 0 and SCI2.SCR.RE = 0 (both serial transmission and reception are disabled).

- SCI2 serial status register (SCI2.SSR) Number of bits: 8 bits Address: 0008 8254h
(In serial communication interface mode (SCI2.SCMR.SMIF bit = 0))

Bit	Symbol	Setting	Bit Name	Description	R/W
b2	TEND	—	Transmit end	0: Character transmission in progress. 1: Character transmission completed.	R

- SCI2 transmit data register (SCI2.TDR) Number of bits: 8 bits Address: 0008 8253h

Bit	Symbol	Setting	Bit Name	Description	R/W
b7-b0	—	— (*1)	—	The data to be transmitted is stored	R/W

Note: *1 The data to be transmitted is set.

(5) Interrupt Controller Unit (ICU)

- Interrupt source priority register 82 (IPR82) Number of bits: 8 bits Address: 0008 7382h

Bit	Symbol	Setting	Bit Name	Description	R/W
b3-b0	IPR[3:0]	0000	SCI2 interrupt priority level	0000: Level 0 (interrupts disabled)	R/W

- IRQ control register 8 (IRQCR8) Number of bits: 8 bits Address: 0008 7508h

Bit	Symbol	Setting	Bit Name	Description	R/W
b3-b2	IRQMD[1:0]	01	IRQ8 detection sense select	01: Falling edge	R/W

- IRQ control register 9 (IRQCR9) Number of bits: 8 bits Address: 0008 7509h

Bit	Symbol	Setting	Bit Name	Description	R/W
b3-b2	IRQMD[1:0]	10	IRQ9 detection sense select	10: Rising edge	R/W

- Interrupt request enable register 1C (IER1C) Number of bits: 8 bits Address: 0008 721Ch

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	IEN0	0	TXI2 interrupt request enable 0	0: TXI2 interrupt requests are disabled	R/W

- Interrupt request register 072 (IR072) Number of bits: 8 bits Address: 0008 7048h

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	IR	0	IRQ8 interrupt status	0: No IRQ8 interrupt present 1: IRQ8 interrupt present	R/W (*1)

Note: *1 Only 0 can be written to clear the flag. Writing a 1 is prohibited.

- Interrupt request register 073 (IR073) Number of bits: 8 bits Address: 0008 7049h

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	IR	0	IRQ9 interrupt status	0: No IRQ9 interrupt present 1: IRQ9 interrupt present	R/W (*1)

Note: *1 Only 0 can be written to clear the flag. Writing a 1 is prohibited.

- Interrupt request register 224 (IR224) Number of bits: 8 bits Address: 0008 70E0h

Bit	Symbol	Setting	Bit Name	Description	R/W
b0	IR	0	TXI2 interrupt status	0: Not TXI2 interrupt present 1: TXI2 interrupt present	R/W (*1)

Note: *1 Only 0 can be written to clear the flag. Writing a 1 is prohibited.

5.7 Functional Specifications

This section contains the specifications for the functions that to be used by the program on the master device.

(1) PowerON_Reset_PC Function

(a) Functional overview

The PowerON_Reset_PC function initializes the stack pointer (the ISP/USP initialization code is automatically generated by the compiler at the beginning of the function when the #pragma entry is declared for the PowerON_Reset_PC function), sets up the INTB (set_intb function: an intrinsic function), initializes the FPSW (set_fpsw function: an intrinsic function), initializes the RAM area sections (_INITSCT function: standard library function), calls the HardwareSetup function, initializes the PSW (set_psw function: an intrinsic function), and sets the processor mode to user mode. Subsequently, the function calls the main function.

(b) Arguments

None

(c) Return value

None

(d) Flowchart

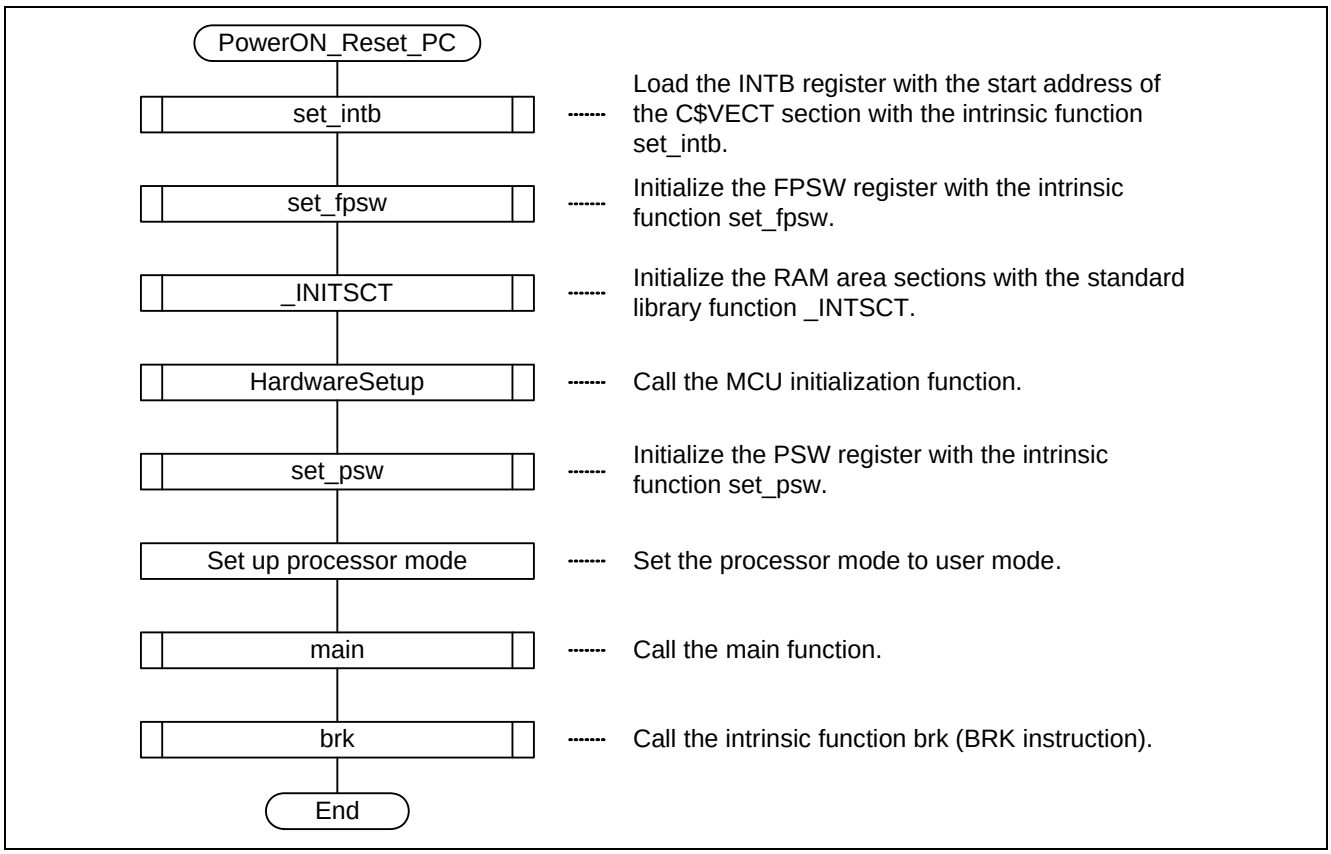

Figure 12 Flowchart (PowerON_Reset_PC) (Master)

(2) HardwareSetup Function

(a) Functional overview

The HardwareSetup function initializes the MCU. Specifically, the following are initialized: clocks (system clock (ICLK), peripheral module clock (PCLK), external bus clock (BCLK)), the pin functions of the I/O port (P00/IRQ8-A) connecting the switch, IRQ9 connected to the slave's Busy port, and the SCI2. Also, HardwareSetup specifies initial output setting for the I/O port (P02, P03, P05, and P34) connecting LED0 to LED3.

(b) Arguments

None

(c) Return value

None

(d) Flowchart

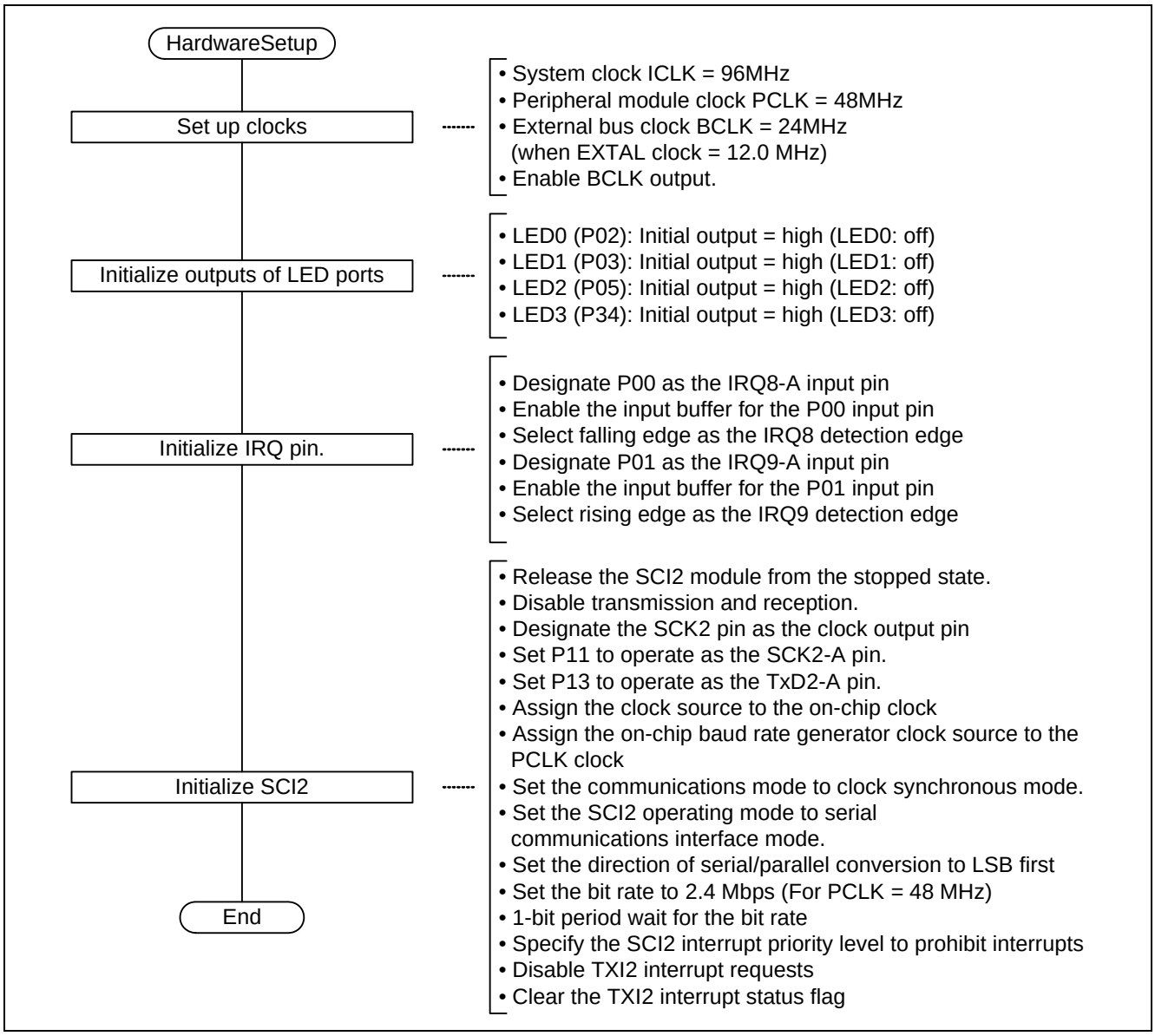

Figure 13 Flowchart (HardwareSetup) (Master)

(3) main Function

(a) Functional overview

The main function determines that the IRQ switch is pressed, and controls transmission of the following: communication command to the slave, erase block number, programming data size, and programming data. Also, the main function controls serial communication handshake, and calls the ndicate_Ending_LED function on successful completion.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

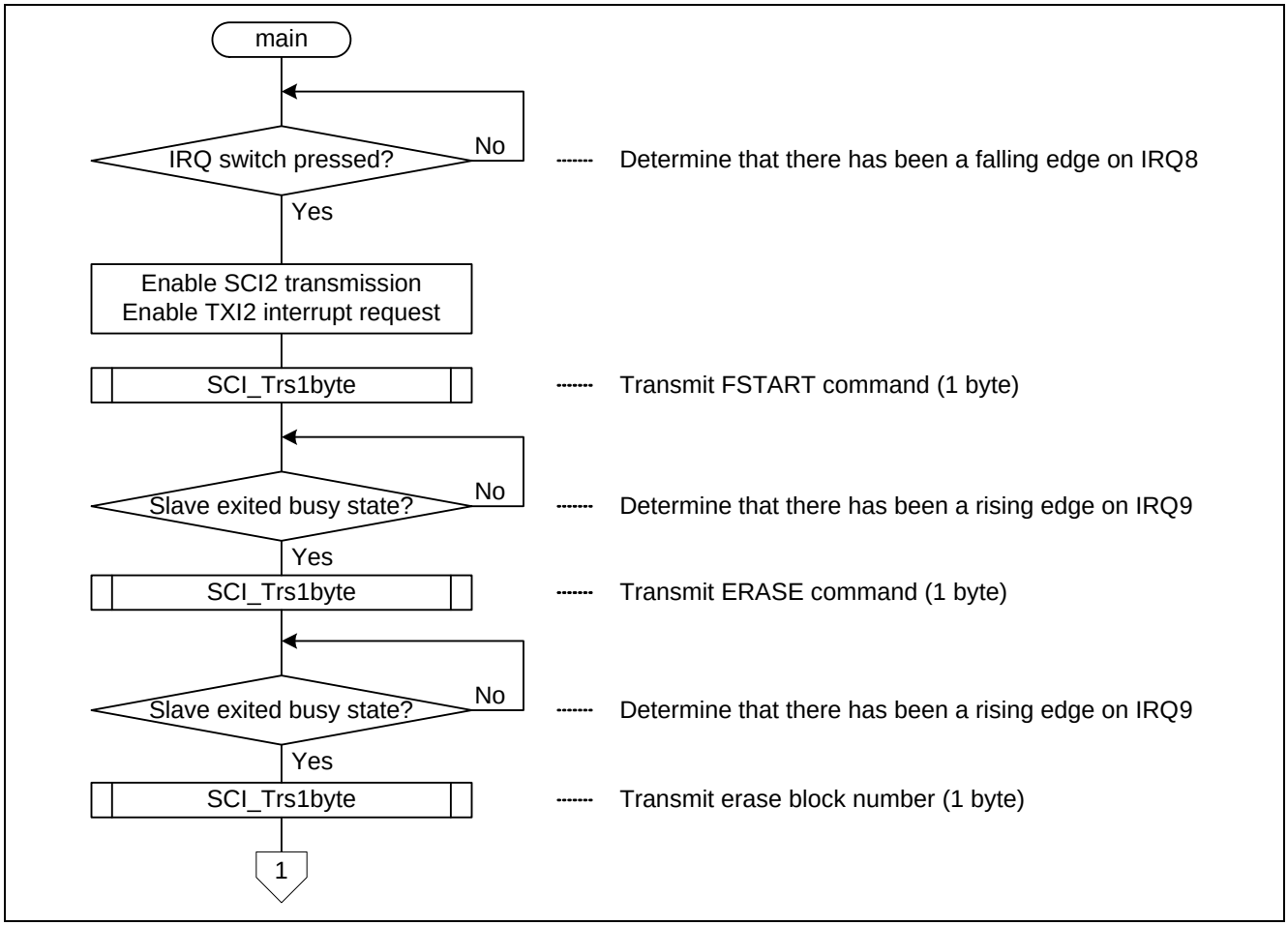

Figure 14 Flowchart (main) (1) (Master)

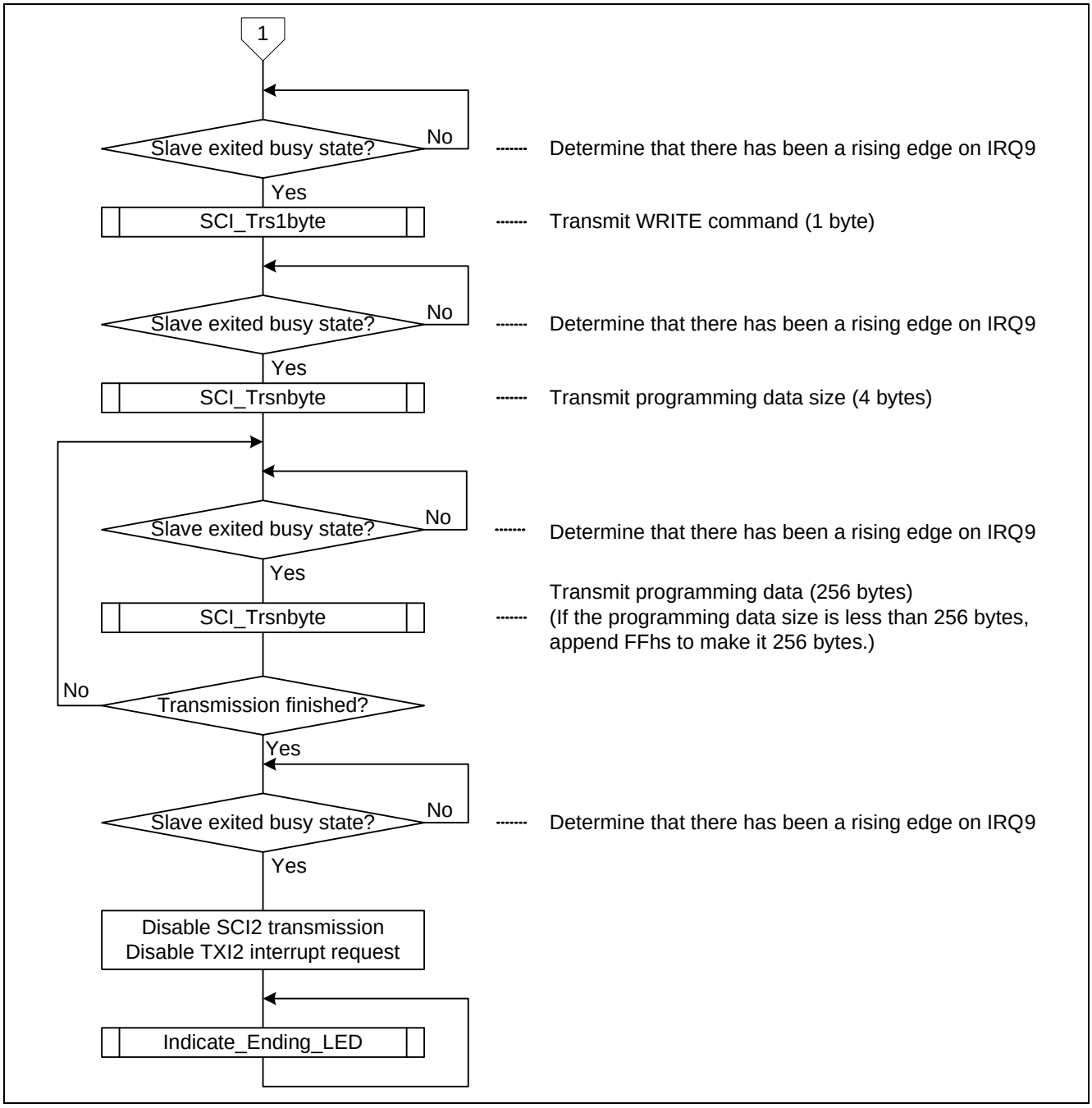

Figure 15 Flowchart (main) (2) (Master)

(4) Indicate_Ending_LED Function

(a) Functional overview

The Indicate_Ending_LED function indicates a normal termination using LED0 to LED3 when the slave successfully completes program/erase operation on the user MAT. It turns on LED0 to LED3 one by one in order.

(b) Arguments

None

(c) Return values

None

(d) Flowchart

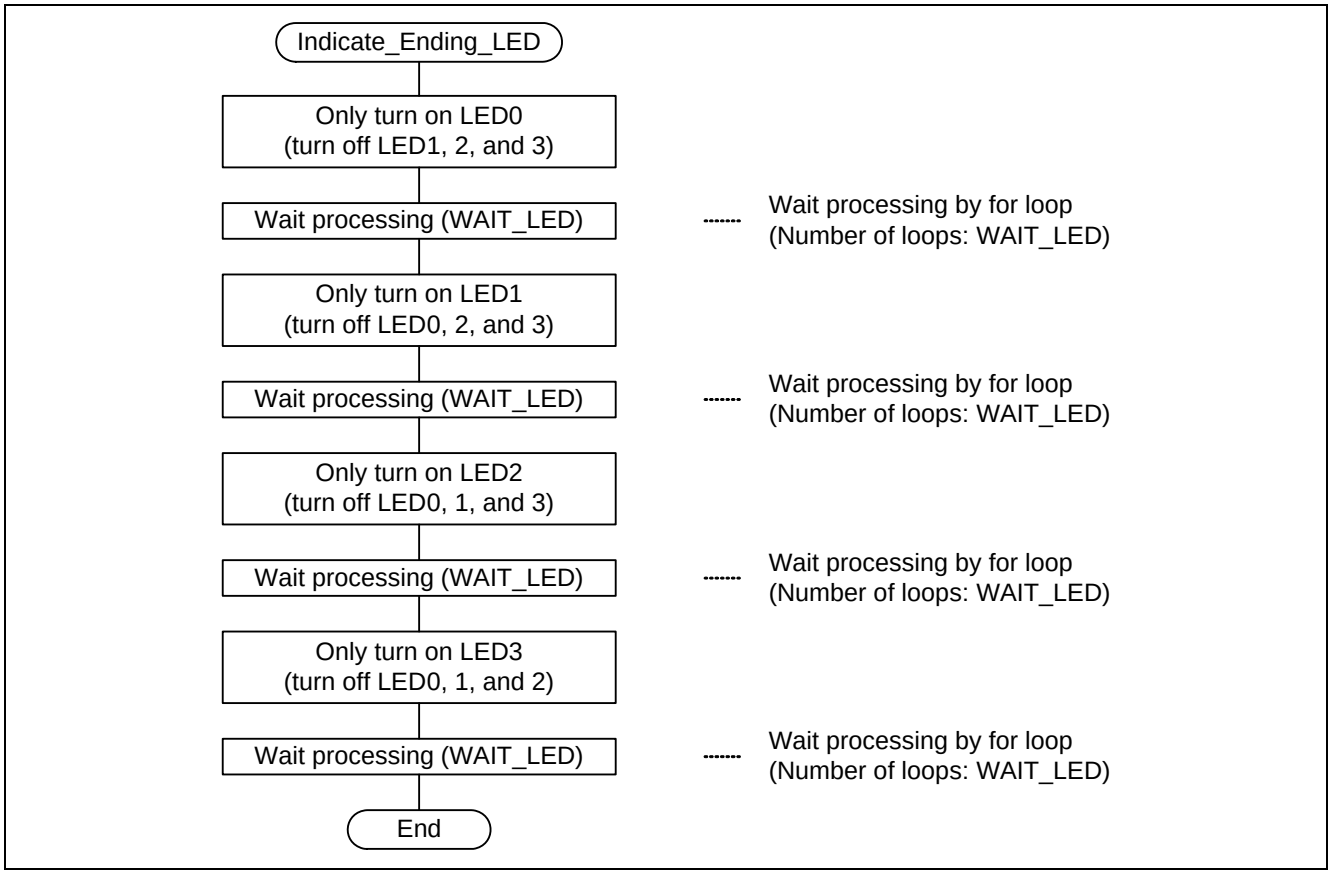

Figure 16 Flowchart (Indicate_Ending_LED) (Master)

(5) SCI_Trs1byte Function

(a) Functional overview

The SCI_Trs1byte function controls transmission of 1-byte data through SCI2 clock synchronous serial communication.

(b) Arguments

Table 14 shows the arguments used by this function.

Table 14 List of SCI_Trs1byte Function Arguments

Argument	Type	Description
First argument	unsigned char	1-byte data to be transmitted through SCI2 clock synchronous serial communication

(c) Return values

None

(d) Flowchart

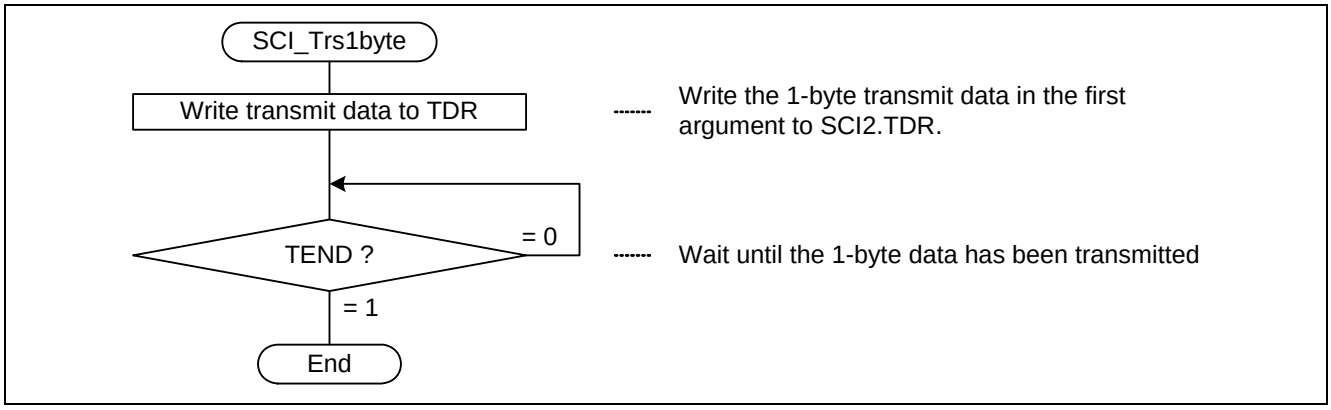

Figure 17 Flowchart (SCI_Trs1byte) (Master)

(6) SCI_Trsnbyte Function

(a) Functional overview

The SCI_Trsnbyte function controls transmission of n-byte data (“n” is the unsigned-short-type first argument) through SCI2 clock synchronous serial communication.

(b) Arguments

Table 15 shows the arguments used by this function.

Table 15 List of SCI_Trsnbyte Function Arguments

Argument	Type	Description
First argument	unsigned short	The number of bytes of data to be transmitted through SCI2 clock synchronous serial communication.
Second argument	unsigned char *	The start address of the area in which transmit data is stored.

(c) Return values

None

(d) Flowchart

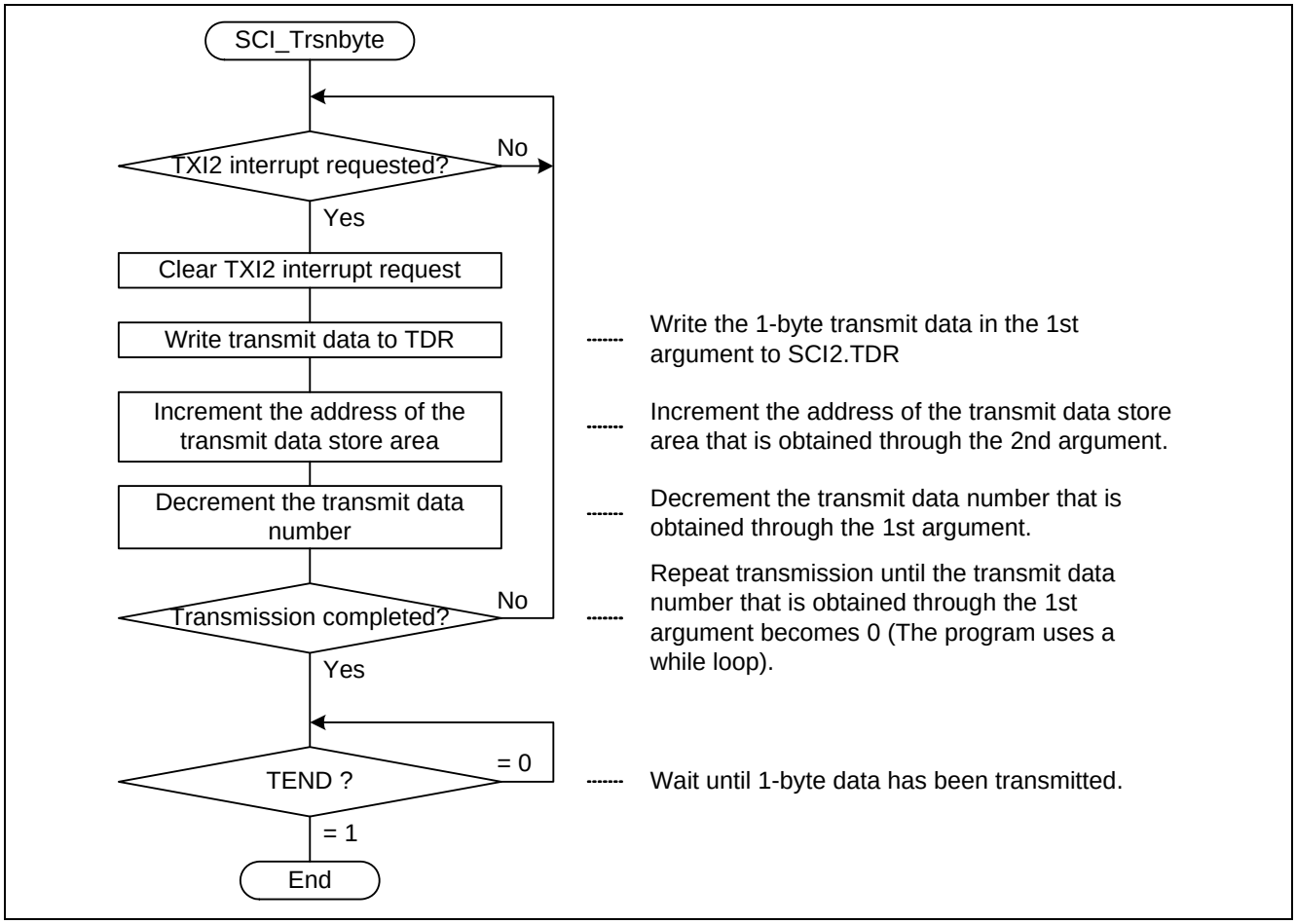

Figure 18 Flowchart (SCI_Trsnbyte) (Master)

6. Usage Notes

6.1 1-bit Period Wait Time for Bit Rate after SCI2 Initialization

In the example given in this application note, a 1-bit period wait time for the bit rate that is obtained after the bit rate register (SCI2.BRR) is set during SCI initialization is measured using a software timer. Since the bit rate for SCI2 clock synchronous serial communication is set to 2.4 Mbps, the following value can be calculated.

1-bit period for the bit rate 2.4 Mbps = 416.666 [ns]

In the example given in this application note, during a 1-bit period wait time for a bit rate, a while loop is executed the number of times specified by the symbolic constant WAIT_SCI1BIT. If the number of cycles for executing a while loop once is 5 (can be checked in assembly language outputted by the compiler), the following value can be calculated.

The number of times a while loop is executed = wait time/(the number of cycles for executing a while loop once × ICLK cycle time)

Note that since the CPU instruction processing time varies depending on pipeline processing, the above number of cycles for executing a while loop once (5 cycles) is an approximate value of instruction processing time.

In the example given in this application note, the wait time is set to 1250 [ns] with a margin, so the following value is obtained.

The number of times a while loop is executed = WAIT_SCI1BIT = 1250 [ns] / (5 × 10.666 [ns]) = 23 (for ICLK = 96 MHz)

Therefore, WAIT_SCI1BIT is defined as 23.

When using the example given in this application note, make an extensive evaluation of the CPU's instruction execution time or measure the time in question using a timer.

7. Reference Documents

- User's Manuals
  RX62N Group, RX621 Group User's Manual: Hardware (R01UH0033EJ)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

  RX Family User's Manual; Software (REJ09B0435)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

- Development Environment Manual
  RX Family C/C++ Compiler Package User's Manual (REJ10J2062)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

- Application Notes
  RX62N Group, RX621 Group
  On-chip Flash Memory Reprogramming in the User Boot Mode (Slave) (R01AN0184EJ)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

- Technical Updates
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Dec 17, 2010	—	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: +49-211-6503-0, 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.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
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 1.0