SH7455 Group/SH7456 Group

Flash Rewrite (CAN Communication)

R01AN0166EJ0100
Rev.1.00
Aug 27, 2010

Introduction

This application note provides the user with an understanding of the rewrite program which receives data through the SH7455 Group/SH7456 Group CAN function and rewrites the received data to the on-chip flash memory.

Target Device

SH74552 (R5F74552KBG): Under development

Precautions

Although the contents of the application note have been verified, refer to the latest hardware manual and evaluate the user system thoroughly before embedding the rewrite program.

Contents

1. Overview

1.1 Specifications

The rewrite program (hereinafter referred to as P/E program) specifications are shown below.

An overview of the system configuration is shown in Figure 1.1, and an overview of the P/E program and Flash Rewrite Tool[*1] flowchart is shown in Figure 1.2.

- The P/E program is stored in the user boot MAT.
- The P/E program starts up in User Boot Mode[*2], and rewrites the user MAT area from EB02 to EB19 (Read Address: H'0000 4000 to H'000F FFFF, Write/Erase Address: H'FD80 4000 to H'FD8F FFFF).
- After transferring the P/E program stored in the user boot MAT to the internal IL memory and OL memory, the program is executed.
- Rewrite data is downloaded from the Flash Rewrite Tool by CAN communication.
- Data is downloaded in 256-byte units (ROM write units), and rewritten to the ROM.
- The data download and ROM rewrite processes are repeated until the target user MAT areas are rewritten completely.
- The start trigger for the P/E program and the request to send rewrite data are controlled by CAN communication with a specified command (refer to Table 2.2).
- The P/E program does not use the cache function.

1.2 On-chip Flash Memory

An overview of the SH7455 Group/SH7456 Group on-chip flash memory (hereinafter referred to as ROM) is shown below. Refer to "Section 12 ROM" in the "SH7455 Group, SH7456 Group User’s Manual: Hardware" for details.

- There are two kinds of memory area (user MAT and user boot MAT) in the same address space.
- Memory areas can be switched with the following setting:
  - The start-up mode setting after a hardware reset is executed
  - Bank-switching through ROM-related registers

*1 The Flash Rewrite Tool controls and sends rewrite data to the rewrite program by CAN communication.

*2 User boot mode can rewrite a user MAT area with desired interface. Rewritable user MAT area is H'0000 0000 to H'000F FFFF (1 Mbytes)

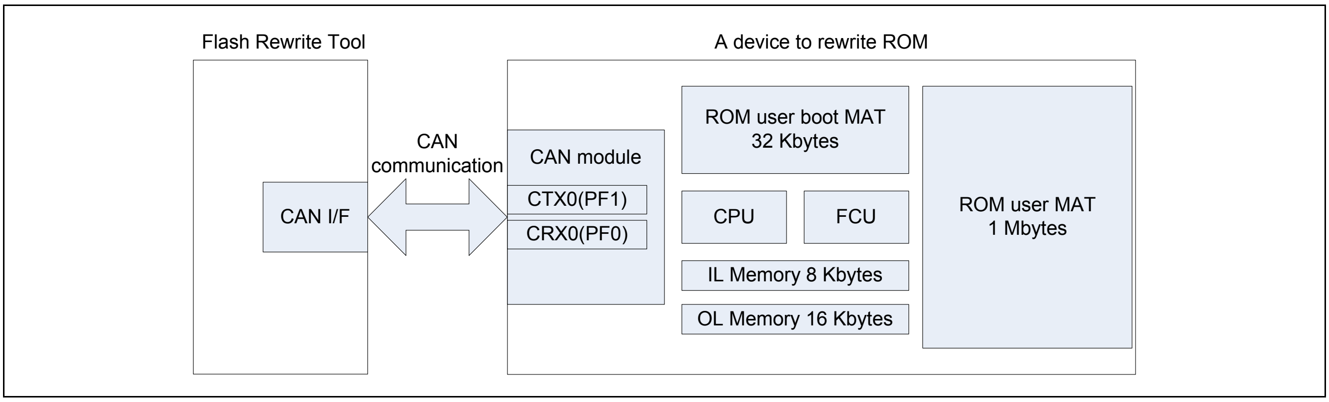


Figure 1.1 System Configuration Overview

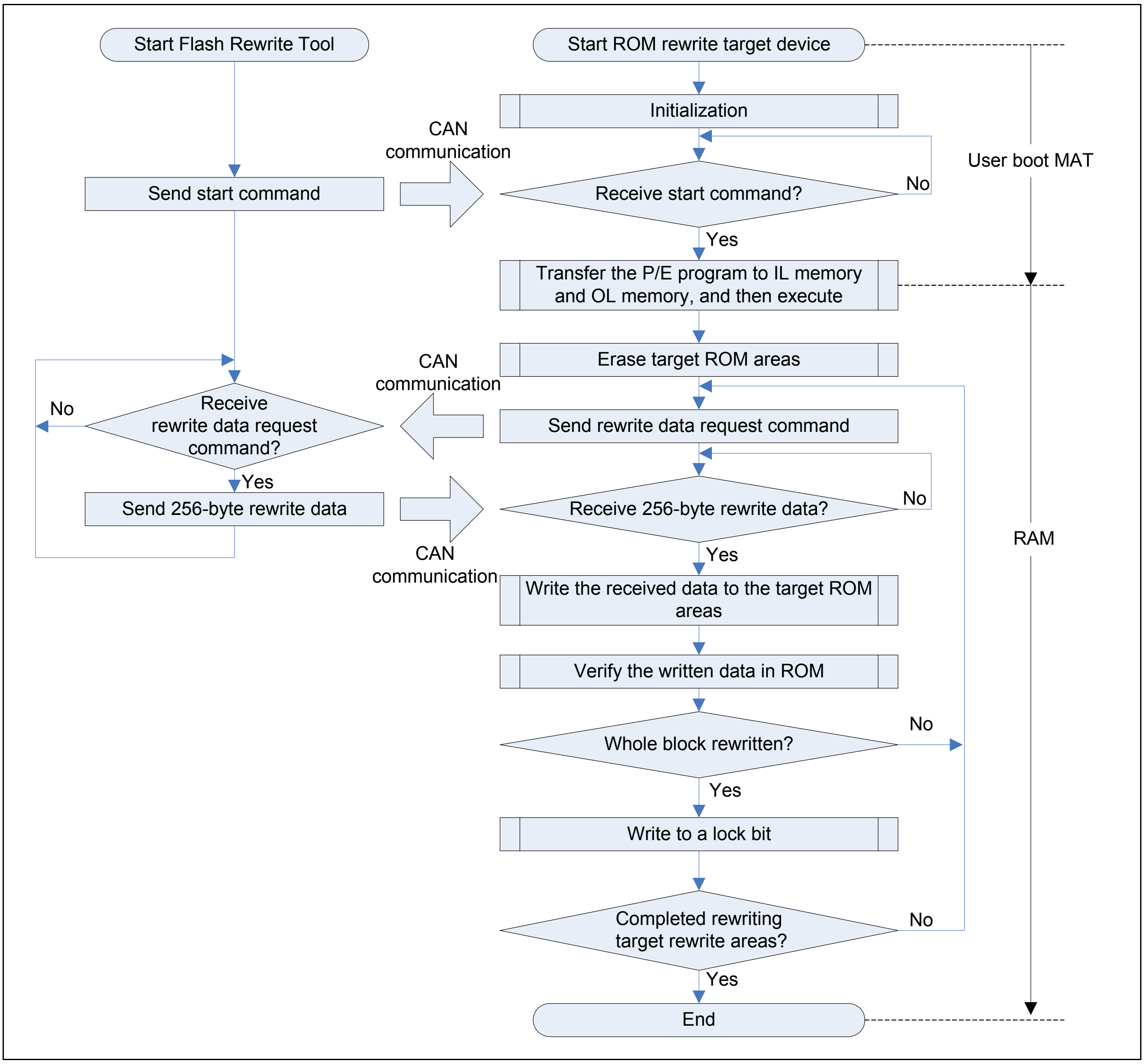

Start Flash Rewrite Tool
Start ROM rewrite target device
Initialization
CAN communication
Send start command
Receive start command?
No
Yes
User boot MAT
Transfer the P/E program to IL memory and OL memory, and then execute
Erase target ROM areas
CAN communication
No
Receive rewrite data request command?
Send rewrite data request command
Yes
Send 256-byte rewrite data
Receive 256-byte rewrite data?
No
CAN communication
Yes
Write the received data to the target ROM areas
RAM
Verify the written data in ROM
Whole block rewritten?
No
Yes
Write to a lock bit
Completed rewriting target rewrite areas?
No
Yes
End

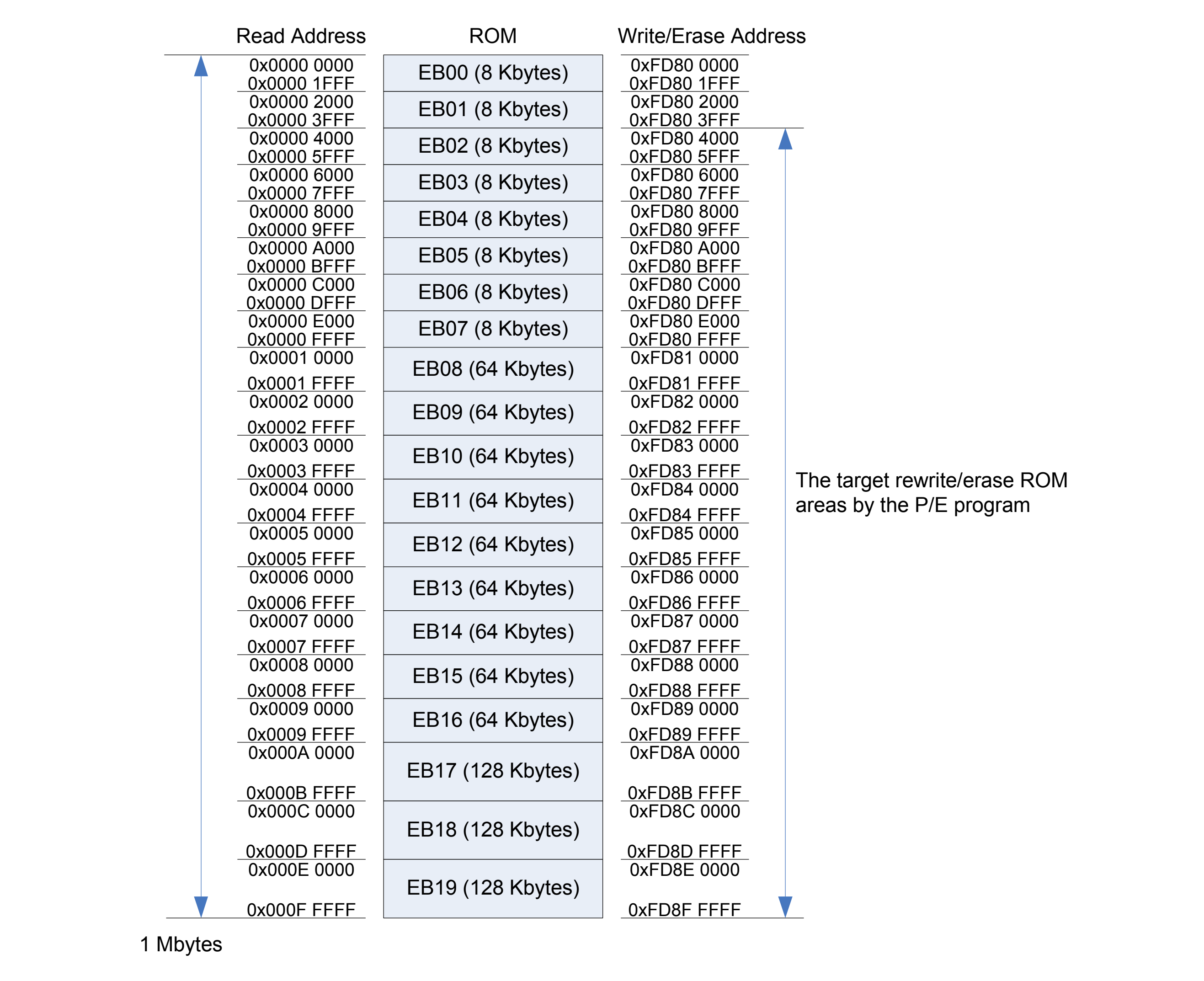
Figure 1.2 Overview of the P/E Program and the Flash Rewrite Tool

1.3 Functions

This section explains the SH7455 Group/SH7456 Group ROM features and registers used by the P/E program.

1.3.1 ROM

Figure 1.3 shows the block allocation and target rewrite areas in the P/E program.



Read Address	ROM	Write/Erase Address
0x0000 0000 0x0000 1FFF	EB00 (8 Kbytes)	0xFD80 0000 0xFD80 1FFF
0x0000 2000 0x0000 3FFF	EB01 (8 Kbytes)	0xFD80 2000 0xFD80 3FFF
0x0000 4000 0x0000 5FFF	EB02 (8 Kbytes)	0xFD80 4000 0xFD80 5FFF
0x0000 6000 0x0000 7FFF	EB03 (8 Kbytes)	0xFD80 6000 0xFD80 7FFF
0x0000 8000 0x0000 9FFF	EB04 (8 Kbytes)	0xFD80 8000 0xFD80 9FFF
0x0000 A000 0x0000 BFFF	EB05 (8 Kbytes)	0xFD80 A000 0xFD80 BFFF
0x0000 C000 0x0000 DFFF	EB06 (8 Kbytes)	0xFD80 C000 0xFD80 DFFF
0x0000 E000 0x0000 FFFF	EB07 (8 Kbytes)	0xFD80 E000 0xFD80 FFFF
0x0001 0000 0x0001 FFFF	EB08 (64 Kbytes)	0xFD81 0000 0xFD81 FFFF
0x0002 0000 0x0002 FFFF	EB09 (64 Kbytes)	0xFD82 0000 0xFD82 FFFF
0x0003 0000 0x0003 FFFF	EB10 (64 Kbytes)	0xFD83 0000 0xFD83 FFFF
0x0004 0000 0x0004 FFFF	EB11 (64 Kbytes)	0xFD84 0000 0xFD84 FFFF
0x0005 0000 0x0005 FFFF	EB12 (64 Kbytes)	0xFD85 0000 0xFD85 FFFF
0x0006 0000 0x0006 FFFF	EB13 (64 Kbytes)	0xFD86 0000 0xFD86 FFFF
0x0007 0000 0x0007 FFFF	EB14 (64 Kbytes)	0xFD87 0000 0xFD87 FFFF
0x0008 0000 0x0008 FFFF	EB15 (64 Kbytes)	0xFD88 0000 0xFD88 FFFF
0x0009 0000 0x0009 FFFF	EB16 (64 Kbytes)	0xFD89 0000 0xFD89 FFFF
0x000A 0000 0x000B FFFF	EB17 (128 Kbytes)	0xFD8A 0000 0xFD8B FFFF
0x000C 0000 0x000D FFFF	EB18 (128 Kbytes)	0xFD8C 0000 0xFD8D FFFF
0x000E 0000 0x000F FFFF	EB19 (128 Kbytes)	0xFD8E 0000 0xFD8F FFFF

Figure 1.3 Block Allocation and the P/E Program Target Rewrite/Erase Areas

1.3.2 ROM-Related Registers

ROM-related registers used by the P/E program, which starts in user boot mode, are shown below. ROM-related registers must be set by a program which performs in any areas except the ROM.

(1) ROM MAT Selection Register (ROMMAT)

- Used to switch the ROM MAT.
- This register is write enabled when H'3B is set to the KEY bits in 16-bit units.

Address	Bit		Function
H'FDFF A820	15 to 8	KEY	These bits enable or disable ROMSEL bit modification. The data written to these bits are not retained.
	7 to 1	-	Reserved Bits (should always be 0)
	0	ROMSEL	0: Selects the user MAT 1: Selects the user boot MAT

(2) Flash P/E Mode Entry Register (FENTRYR)

- Used to set ROM mode.
- This register is write enabled when H'AA is set to the FEKEY bits in 16-bit units.
- This register is write disabled when all ROM blocks are not in read mode (H'0000), and the blocks are set to P/E mode.
- This register is write disabled if a value other than H'01 is written to the lower 8 bits (b0 to b7).
- When an invalid data is set to this register, the flash control unit (FCU) initializes this register to H'0000.

Address	Bit		Function
H'FDFF A902	15 to 8	FEKEY	These bits enable or disable rewriting of the FENTRY0 bit. Write data to these bits are not retained.
	7 to 2	-	Reserved Bits (should always be 0)
	1	FENTRY1	These bits are always read as "0". The write value should always be "0".
	0	FENTRY0	0: The block of ROM from EB00 to EB19 (1MB) is in read mode*3 1: The block of ROM from EB00 to EB19 (1MB) is in P/E mode*4

(3) Flash Protect Register (FPROTR)

- Set the lock bits*5 to enable/disable write/erase protection.
- This register is write enabled when H'55 is set to the FPKEY bits in 16-bit units in P/E mode.
- When an invalid data is set to this register, the FCU initializes this register to H'0000.

Address	Bit		Address
H'FDFF A904	15 to 8	FPKEY	These bits enable or disable FPROTCN bit modification. The data written to these bits are not retained.
	7 to 1	-	Reserved Bits (should always be 0)
	0	FPROTCN	0: Enables protection through the lock bits 1: Disables protection through the lock bits

*3 The mode to read ROM.
*4 The mode to write/erase ROM.
*5 The bit for write/erase protection which is located at each ROM block.

(4) Flash Reset Register (FRESETR)

- Used to initialize the FCU.
- This register is write enabled when H'CC is set to the FRKEY bits in 16-bit units.
- When a reset is issued to the FCU during a ROM write/erase operation, the current process is aborted, and the FCU is initialized.
- After initializing the FCU (FRESET bit is 1) by software and maintaining the initialized state for 100 μs, by releasing FCU initialization (FRESET bit is 0), the FCU initialization process is completed.

Address	Bit		Function
H'FDFF A906	15 to 8	FRKEY	These bits enable or disable FRESET bit modification. The data written to these bits are not retained.
	7 to 1	-	Reserved Bits (should always be 0)
	0	FRESET	0: Issue no reset to the FCU. 1: Issues a reset to the FCU.

(5) Flash Access Status Register (FASTAT)

- Used to check the status of a ROM access error.
- When an unauthorized access occurs on ROM, the FCU sets the corresponding error bit.
- Used to check FCU command lock status.

Address	Bit		Function
H'FDFF A810	7	ROMAE	0: No ROM access error has occurred. 1: A ROM access error has occurred.
	6 to 5	-	Reserved Bits (should always be 0)
	4	CMDLK	0: The FCU is not in a command-locked state 1: The FCU is in a command-locked state
	3 to 0	-	Reserved Bits

(6) Flash Status Register 0 (FSTATR0)

- Used to check the FCU status and various error states.
- When an error occurs during a write/erase operation, the corresponding error bit is set.
- When the FCU detects an unauthorized command or unauthorized ROM access, the corresponding error bit is set.

Address	Bit		Function
H'FDFF A900	7	FRDY	0: Programming or erasure process, or lock bit read command processing. 1: None of the above is in progress.
	6	ILGLERR	0: The FCU has not detected any illegal command or illegal ROM access 1: The FCU has detected an illegal command or illegal ROM access
	5	ERSERR	0: Erasure process has been completed successfully 1: An error has occurred during erasure
	4	PRGERR	0: Programming has been completed successfully 1: An error has occurred during programming
	3	FCUSO	0: Sequence stopped 1: Sequence is in progress
	2 to 0	-	Reserved Bits (should always be 0)

(7) Flash Status Register 1 (FSTATR1)

- Used to check the FCU status and read lock bit data.
- When an error occurs during CPU processing, the FCU sets the corresponding error bit.
- When an error occurs while the FCU accesses internal data, the corresponding error bit is set.

Address	Bit		Function
H'FDFF A901	7	FCUERR	0: No error has occurred during the CPU process in the FCU. 1: An error has occurred during the CPU process in the FCU.
	6 to 5	-	Reserved Bits (should always be 0)
	4	FLOCKST	0: Protect status 1: Non-protected status
	3 to 2	-	Reserved Bits (should always be 0)
	1	FRDTCT	0: No FCU data access error has occurred. 1: An FCU data access error has occurred.
	0	-	Reserved Bit (should always mask this bit because the read value from this bit is undefined)

1.3.3 Controller Area Network (CAN) Channel 0

CAN channel 0 is used for transmitting and receiving control commands and receiving (downloading) rewrite data. The CAN communication specifications and control commands in the P/E program are shown in "2.1 Initial Settings".

1.3.4 Timer Unit (TMU) Channel 0

TMU channel 0 is used for the wait time control and timeout detection during rewrite processing.
The timer setting in the P/E program is shown in "2.3 Initial Setting of the P/E Program".

1.4 Operating Conditions

The operating conditions of the P/E program are shown in Table 1.1.

Table 1.1 Operating Conditions

Item	Contents
MCU	SH7455 Group/SH7456 Group
Input frequency	20 MHz
Clock frequencies	CPU clock (Ick): 160 MHz Periphery clock (Pck): 40 MHz
Operating mode	Single-chip mode: User boot mode
Development tool	Renesas Electronics High-performance Embedded Workshop Version 4.08.00.011
C/C++ compiler	Renesas Electronics C/C++ Compiler Package for SuperH RISC Engine Family V.9.03 Release 02

2. Program Specifications

This section explains the operation and specifications of the P/E program. The contents follow the order in the Figure 2.1 flowchart.

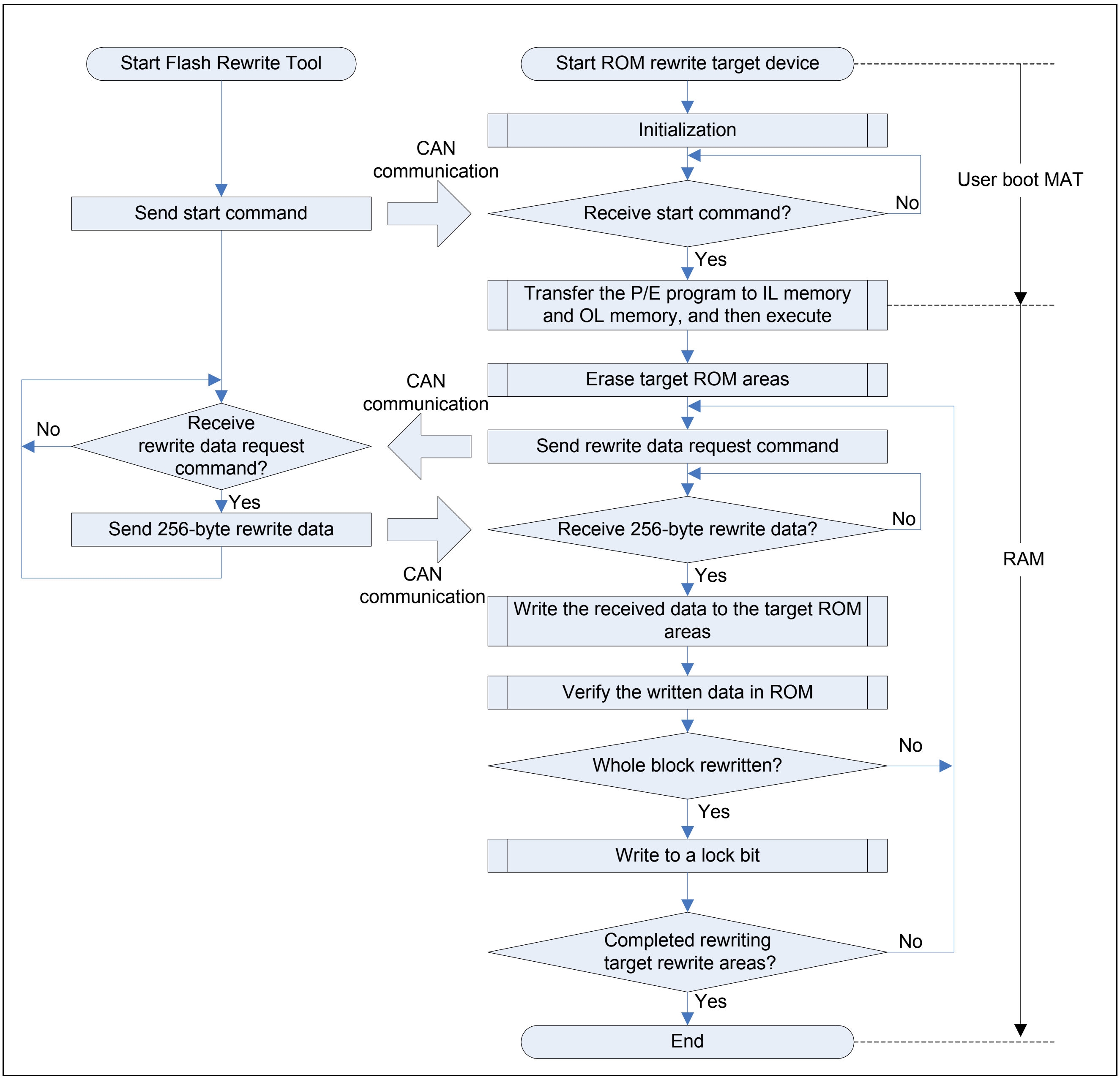

Start Flash Rewrite Tool
Send start command
Receive rewrite data request command?
No
Yes
Send 256-byte rewrite data
CAN communication
CAN communication
CAN communication
Start ROM rewrite target device
Initialization
Receive start command?
No
Yes
Transfer the P/E program to IL memory and OL memory, and then execute
Erase target ROM areas
Send rewrite data request command
Receive 256-byte rewrite data?
No
Yes
Write the received data to the target ROM areas
Verify the written data in ROM
Whole block rewritten?
No
Yes
Write to a lock bit
Completed rewriting target rewrite areas?
No
Yes
End
User boot MAT
RAM

Figure 2.1 Overview of the P/E Program and the Flash Rewrite Tool

2.1 Initial Settings

The initial setting executes programs in the user boot MAT. The initial setting flowchart is shown in Figure 2.2.

(1) Pin function settings
Set "I/O Port: PF0" to the CAN input pin function CRX0.
Set "I/O Port: PF1" to the CAN output pin function CTX0.

(2) CAN module initial settings
In the CAN module initial settings, command transmission/reception and rewrite data reception settings are performed. The CAN communication specifications in the P/E program are shown in Table 2.1, and control command specifications are shown in Table 2.2.

Table 2.1 CAN Communication Specifications

Item		Specification				
Communication transmission rate		500 kbps				
Format		Standard ID format				
Pins	Transmission side	CTX0 (PF1)				
	Reception side	CRX0 (PF0)				
Mailboxes		Mailbox	Tx/Rx	ID	Data Length	Usage
		C0MB0	Rx	H'100	1-byte	Receive start command
		C0MB1	Rx	H'111	8-byte	Receive ROM rewrite data
		C0MB32	Tx	H'101	1-byte	Send rewrite data request command
Sample point		62.50%				

Table 2.2 Control Command Specifications

Command	Data	Function
Start command	H'11	Request to start ROM rewrite process
Rewrite data request command	H'22	Request to send 256-byte rewrite data

(3) Preparation for reception
According to the above CAN communication specifications, initialize mailbox 0 to receive the start command.

(4) Start command reception
After receiving the start command from the Flash Rewrite Tool by CAN communication, the P/E program transfer is performed.

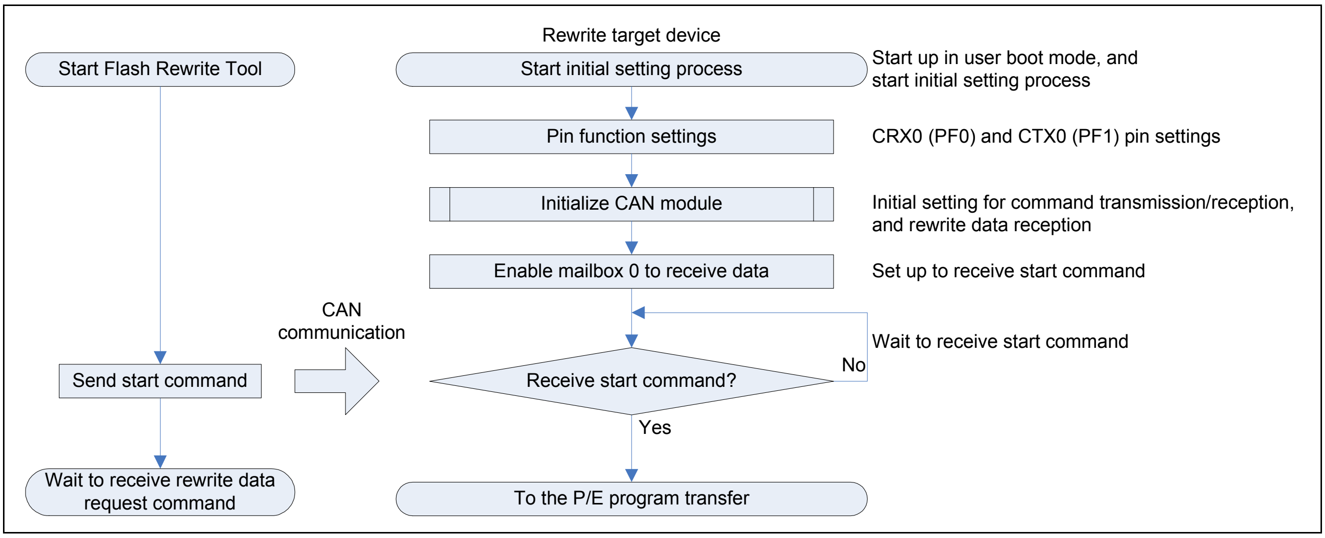

Rewrite target device
Start Flash Rewrite Tool
Start initial setting process
Start up in user boot mode, and start initial setting process
Pin function settings
CRX0 (PF0) and CTX0 (PF1) pin settings
Initialize CAN module
Initial setting for command transmission/reception, and rewrite data reception
Enable mailbox 0 to receive data
Set up to receive start command
CAN communication
Wait to receive start command
Send start command
Receive start command?
No
Yes
Wait to receive rewrite data request command
To the P/E program transfer

Figure 2.2 Initial Setting Process

2.2 Transfer and Execution of the P/E Program

The P/E program transfer process is executed in the user boot MAT area. After the initial settings are completed, if the start command is received from the Flash Rewrite Tool, the transfer process starts. After masking all the interrupt requests except NMI, the P/E program is transferred from the user boot MAT to IL memory and OL memory. After the P/E program transfer is completed, the P/E program in the IL memory and OL memory areas is executed.

The flowchart of transfer and execution process of the P/E program is shown in Figure 2.3.

(1) P/E program transfer

The P/E program in the user boot MAT area is transferred to IL memory and OL memory.

(2) P/E program execution

The P/E program transferred to the IL memory is executed.

P/E program processing is executed in the following order: Initial setting → ROM erase → ROM write.

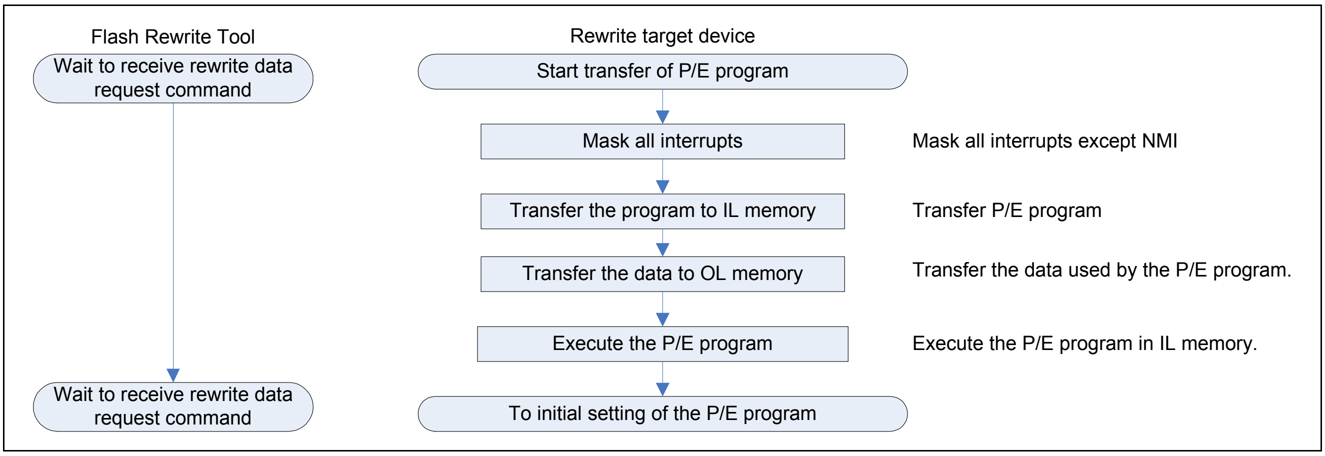

Flash Rewrite Tool
Wait to receive rewrite data request command
Wait to receive rewrite data request command
Rewrite target device
Start transfer of P/E program
Mask all interrupts
Transfer the program to IL memory
Transfer the data to OL memory
Execute the P/E program
To initial setting of the P/E program
Mask all interrupts except NMI
Transfer P/E program
Transfer the data used by the P/E program.
Execute the P/E program in IL memory.

Figure 2.3 Transfer and Execution Process of the P/E Program

Notes on the NMI Interrupt

When an NMI interrupt is generated, perform the procedure below in the NMI interrupt handler. During ROM erase/write, if an NMI interrupt is generated and the procedure below is not performed before turning off the power, ROM may fail.

In the attached sample code, the BL bit in the SR register is 1 (mask an interrupt request), and the NMIB bit in the ICR0 register is 0 (NMI interrupts will be deferred when BL is 1), so an NMI interrupt may not be generated.

(1) Wait until the FCU is in an idle state (the FRDY bit in the FSTATR0 register is 1).

(2) Check the current ROM mode. If it is in P/E mode, switch to read mode.

(3) After performing (1) and (2), perform an appropriate interrupt handler.

2.3 Initial Setting of the P/E Program

After executing the P/E program that was transferred to the IL memory and OL memory areas, perform the initial settings necessary to rewrite ROM. Figure 2.4 shows the flowchart of the initial settings for the P/E program.

(1) ROM MAT switch

The ROM memory areas are switched from the user boot MAT (the P/E program stored area) to the user MAT (target rewrite/erase area). If an error is detected, the process moves to "2.7 Error Processing (2)".

(2) Timer unit initial setting for the wait and timeout detection

The timer count is stopped, and then the counter clock is initialized. The wait and timeout detection for each process in the P/E program are shown in Table 2.3.

Table 2.3 Wait/Timeout Detection Time

Item		Definition Label	Setting Value	Unit
Wait time	FRESET "1" setting time	tRESW2	100	μs
Timeout detection time	256-byte write time	tP256	12	ms
	8 Kbytes block erase time	tE8K	150	ms
	64 Kbytes block erase time	tE64K	1120	ms
	128 Kbytes block erase time	tE128K	2240	ms

Besides the above table, mode transition standby time (tMODEW = 1μs) and timeout of lock bit read (tLCKR = 10μs) are used to follow usage examples of ROM in the hardware manual.

(3) FCU initialization process

After setting the FRESET bit in the FRESETR register to 1 (issues reset to the FCU) by software and maintaining the initialized state for 100 μs, reset the FRESET bit to 0 (issues no reset to the FCU).

(4) ROM P/E mode setting

Depending on a target block, FENTRY0 bit in the FENTRYR register is set to 1 (a target ROM block is in P/E mode), and wait 1 μs by software for mode transition.

The ROM P/E mode entry bit correspondence table is shown in Table 2.4. EB02 to EB19 (H'FD80 4000 to H'FD8F FFFF) is the write/erase target area in the P/E program, so the P/E program sets the FENTRY0 bit to 1.

Table 2.4 ROM P/E Mode Entry Bit Correspondence Table

Model Name	Bit Name	Target Block Number	Writing/Erase ROM Address
SH7455/SH7456 Group	FENTRY0	EB00-EB19	H'FD80 0000-H'FD8F FFFF

After the initial setting, the target block erase process is executed.

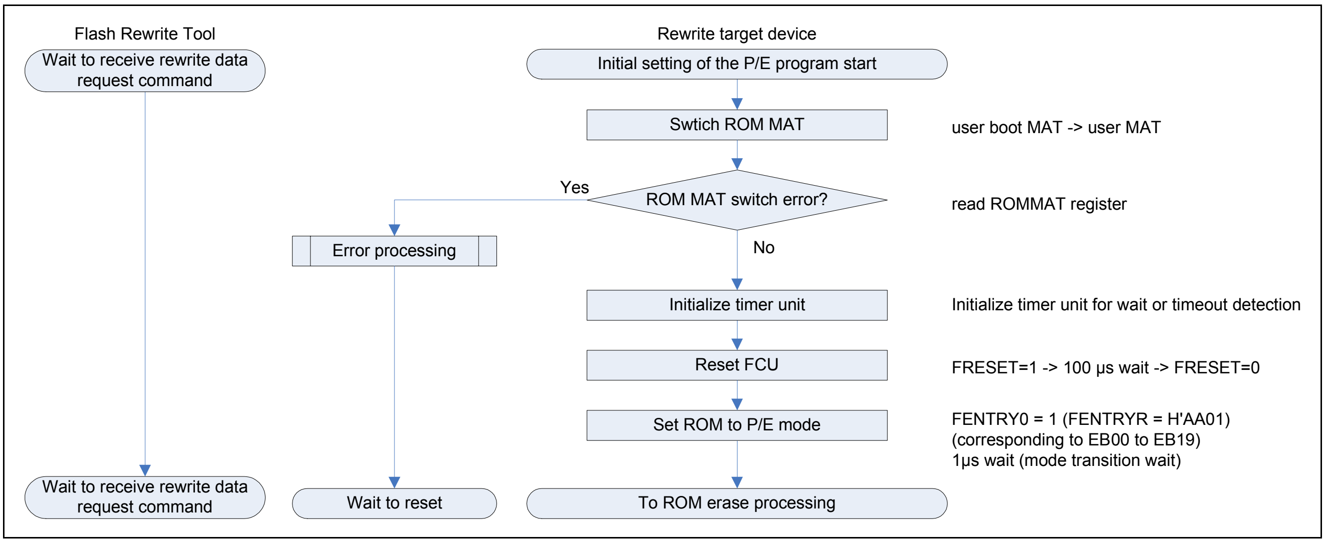

Flash Rewrite Tool
Rewrite target device
Wait to receive rewrite data request command
Initial setting of the P/E program start
Swtich ROM MAT
user boot MAT -> user MAT
ROM MAT switch error?
Yes
No
read ROMMAT register
Error processing
Initialize timer unit
Initialize timer unit for wait or timeout detection
Reset FCU
FRESET=1 -> 100 μs wait -> FRESET=0
Set ROM to P/E mode
FENTRY0 = 1 (FENTRYR = H'AA01)
(corresponding to EB00 to EB19)
1μs wait (mode transition wait)
Wait to receive rewrite data request command
Wait to reset
To ROM erase processing

Figure 2.4 P/E Program Initial Setting Process

2.4 ROM Erase

Erase target ROM areas. The ROM erase flowchart is shown in Figure 2.5.

(1) Disable lock bit protection

If erase block is protected, disable the lock bit protect. Refer to "2.6 Disabling Lock Bit Protection" for details.

(2) Erase block

Use the block erase command to erase the target block.

In case an error is detected during this process, the process moves to "2.7 Error Processing (1)".

Processes (1) and (2) above are repeated until all target blocks are erased.

After ROM is erased, ROM write is executed.

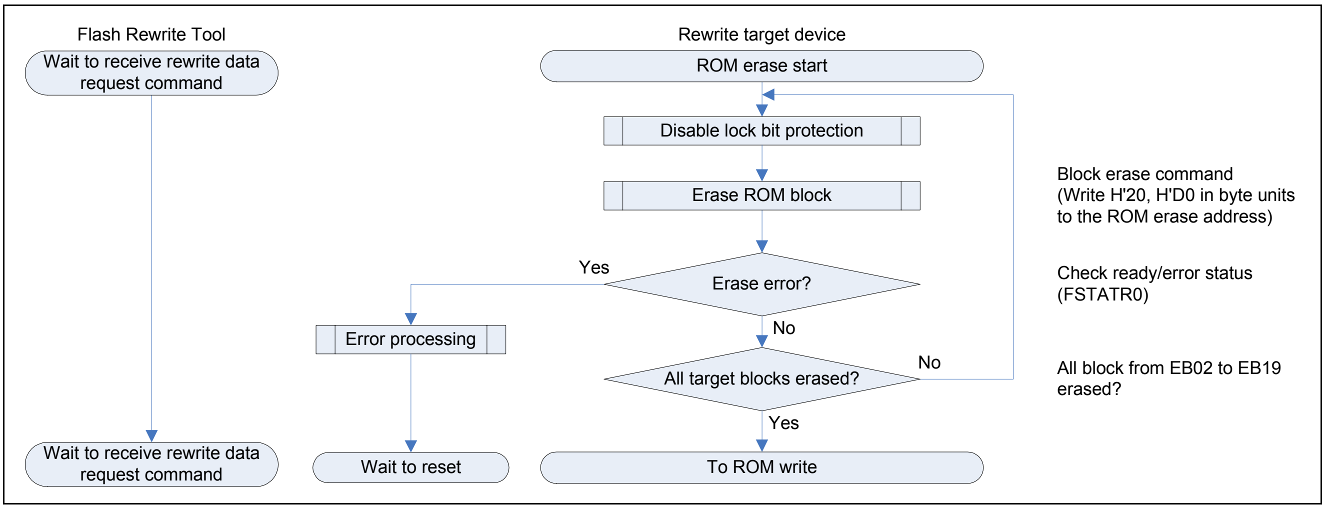

Flash Rewrite Tool
Wait to receive rewrite data request command
Rewrite target device
ROM erase start
Disable lock bit protection
Erase ROM block
Block erase command (Write H'20, H'D0 in byte units to the ROM erase address)
Yes
Erase error?
Check ready/error status (FSTATR0)
Error processing
No
All target blocks erased?
No
All block from EB02 to EB19 erased?
Yes
Wait to receive rewrite data request command
Wait to reset
To ROM write

Figure 2.5 ROM Erase

2.5 ROM Write

The data downloaded from the Flash Rewrite Tool is written in the target ROM areas. In this process, (1) is executed before writing to each ROM block. Steps (2) to (4) are repeated successively in 256-byte units until all target blocks are written. When a block is successfully written, (5) is executed to protect the data.

Details of steps (2) to (5) are show in 2.5.1 to 2.5.4.

(1) Disable lock bit protection (refer to "2.6 Disabling Lock Bit Protection")
(2) Download rewrite data from the Flash Rewrite Tool.
(3) Write the download data to ROM.
(4) Verify the written data in ROM.
(5) Write lock bit.

If an error is detected in step (3) or (5), the process moves to "2.7 Error Processing (1)". If an error is detected in step (4), the process moves to "2.7 Error Processing (2)".

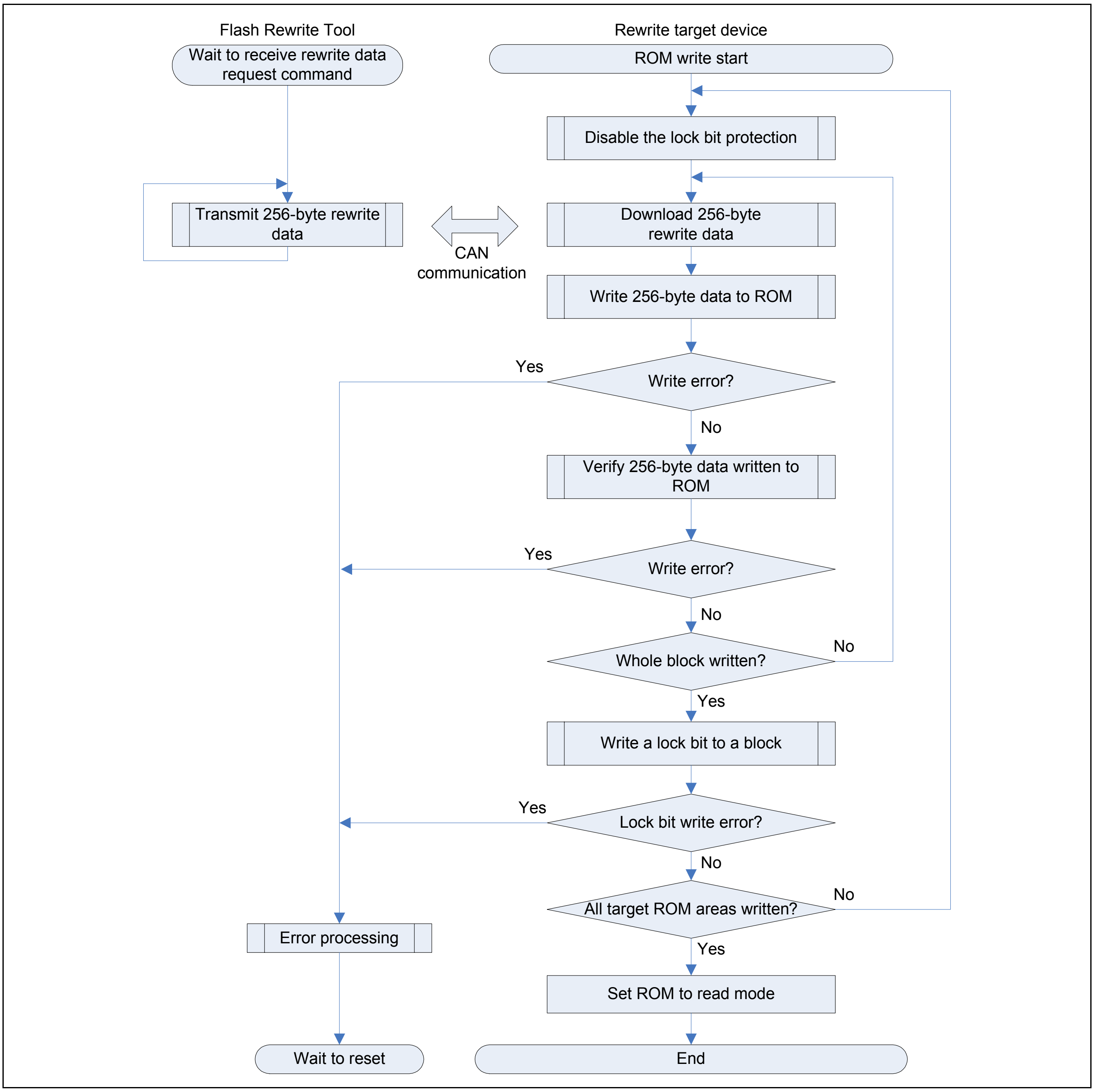

Flash Rewrite Tool
Rewrite target device
Wait to receive rewrite data request command
ROM write start
Disable the lock bit protection
Transmit 256-byte rewrite data
CAN communication
Download 256-byte rewrite data
Write 256-byte data to ROM
Yes
Write error?
No
Verify 256-byte data written to ROM
Yes
Write error?
No
Whole block written?
No
Yes
Write a lock bit to a block
Yes
Lock bit write error?
No
All target ROM areas written?
No
Yes
Error processing
Set ROM to read mode
Wait to reset
End

Figure 2.6 ROM Write Overview

2.5.1 Rewrite Data Download from the Flash Rewrite Tool

Transmit the rewrite data request command to the Flash Rewrite Tool, and receive (download) 256-byte rewrite data from the Flash Rewrite Tool by CAN communication.

The flowchart of downloading the 256-byte rewrite data is shown in Figure 2.7.

(1) Transmission of the rewrite data request command

According to the CAN communication specification (Table 2.1) and control command specification (Table 2.2), the rewrite data request command is transmitted to the Flash Rewrite Tool.

(2) Download 256-byte rewrite data

According to the CAN communication specification (Table 2.1), 256-byte rewrite data is downloaded from the Flash Rewrite Tool. The downloaded data is stored in OL memory.

When the 256-byte data download is completed, that data is written to ROM.
Even if there is no rewrite data in a part of the ROM area, the P/E program must receive consecutive 256-byte data, so the Flash Rewrite Tool must send H'FF (initial ROM data) data instead of no rewrite data.

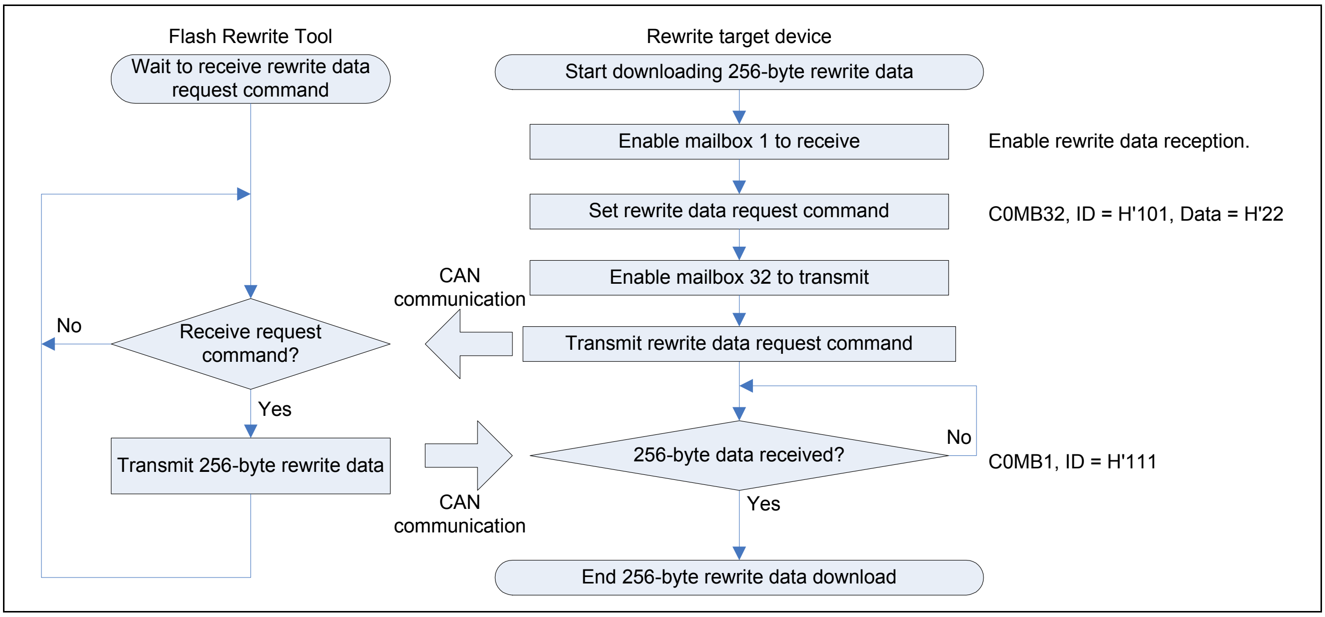

Flash Rewrite Tool
Wait to receive rewrite data request command
No
Receive request command?
Yes
Transmit 256-byte rewrite data
CAN communication
CAN communication
Rewrite target device
Start downloading 256-byte rewrite data
Enable mailbox 1 to receive
Enable rewrite data reception.
Set rewrite data request command
C0MB32, ID = H'101, Data = H'22
Enable mailbox 32 to transmit
Transmit rewrite data request command
256-byte data received?
No
C0MB1, ID = H'111
Yes
End 256-byte rewrite data download

Figure 2.7 256-byte Rewrite Data Download

2.5.2 256-byte Data Write to ROM

Downloaded 256-byte data in 2.5.1 is written to ROM. The flowchart of the 256-byte data write to ROM is shown in Figure 2.8.

(1) 256-byte data write to ROM
256-byte data is written to ROM by a program command.
If an error is detected, the process moves to "2.7Error Processing (1)".

After writing 256-byte data to ROM, the ROM data is read and verified.

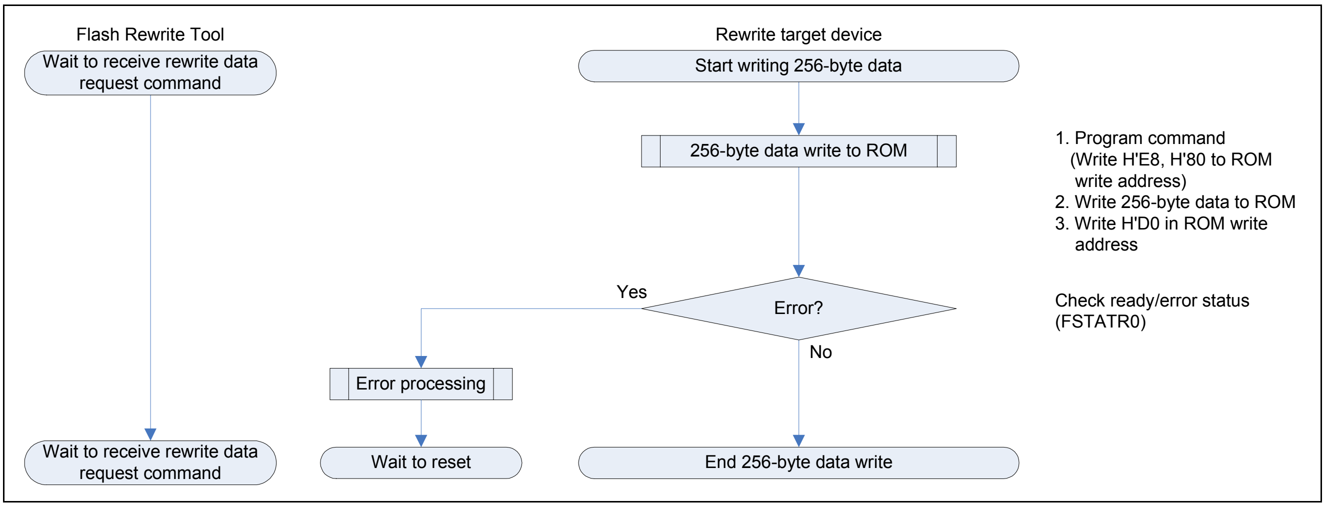

Flash Rewrite Tool
Wait to receive rewrite data request command
Rewrite target device
Start writing 256-byte data
256-byte data write to ROM
1. Program command (Write H'E8, H'80 to ROM write address)
2. Write 256-byte data to ROM
3. Write H'D0 in ROM write address
Yes
Error?
No
Check ready/error status (FSTATR0)
Error processing
Wait to receive rewrite data request command
Wait to reset
End 256-byte data write

Figure 2.8 256-byte Data ROM Write

2.5.3 Verifying 256-byte Data Written to ROM

Compare 256-byte data in ROM (2.5.2) with the 256-byte data in OL memory (2.5.1) to check data integrity of the written data in ROM. The flowchart of ROM data verification is shown in Figure 2.9.

(1) Setting ROM read mode
The FENTRY 0 bit of a FENTRYR register is set to 0 (ROM read mode).
Wait 1 μs for ROM mode transition by software.

(2) Comparing 256-byte data
Compare the 256-byte data in ROM with the 256-byte data in the OL memory. When a difference in the data is detected, the P/E program judges this as an error. If the error is detected, the process moves to "2.7 Error Processing (2)".

(3) Setting ROM P/E mode
Set the target blocks to P/E mode. Refer to "2.3 Initial Setting of the P/E Program (4)" for details.

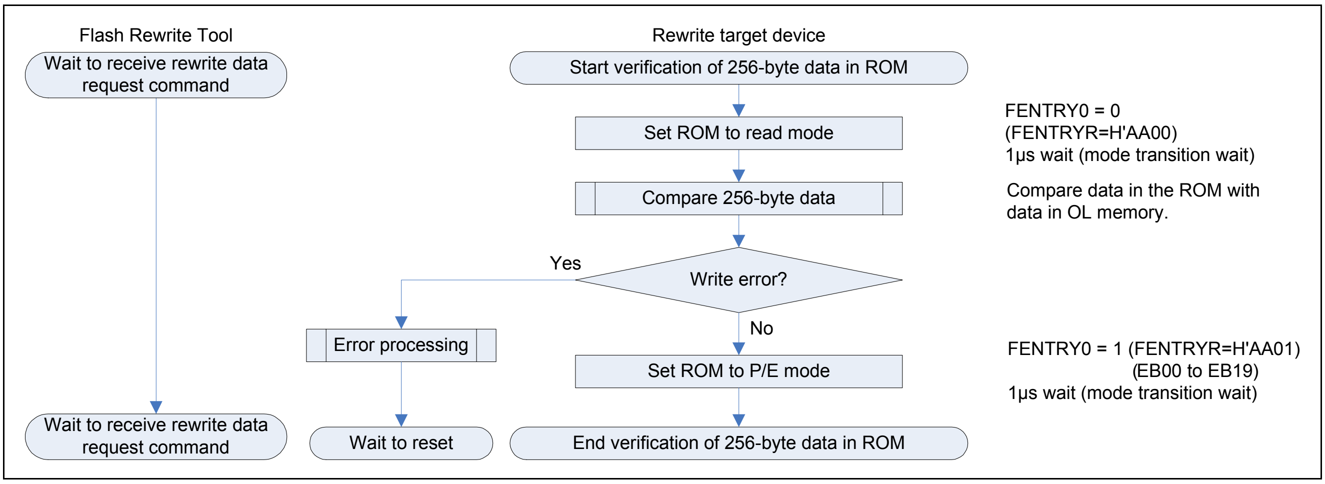

Flash Rewrite Tool
Rewrite target device
Wait to receive rewrite data request command
Start verification of 256-byte data in ROM
Set ROM to read mode
FENTRY0 = 0
(FENTRYR=H'AA00)
1μs wait (mode transition wait)
Compare 256-byte data
Compare data in the ROM with data in OL memory.
Yes
Write error?
No
Error processing
Set ROM to P/E mode
FENTRY0 = 1 (FENTRYR=H'AA01)
(EB00 to EB19)
1μs wait (mode transition wait)
Wait to receive rewrite data request command
Wait to reset
End verification of 256-byte data in ROM

Figure 2.9 ROM Data Verification

2.5.4 Lock Bit Write

After writing to a block, write the lock bit to protect the data. The flowchart of the lock bit write is shown in Figure 2.10. Refer to Figure 1.3 for details of each block size.

(1) Writing the lock bit

Use the lock bit program command to write the lock bit of the target block. If an error is detected, the process moves to "2.7 Error Processing (2)".

Processing is complete when all target areas (H'FD80 4000 to H'FD8F FFFFF (EB02 to EB19)) have been written.

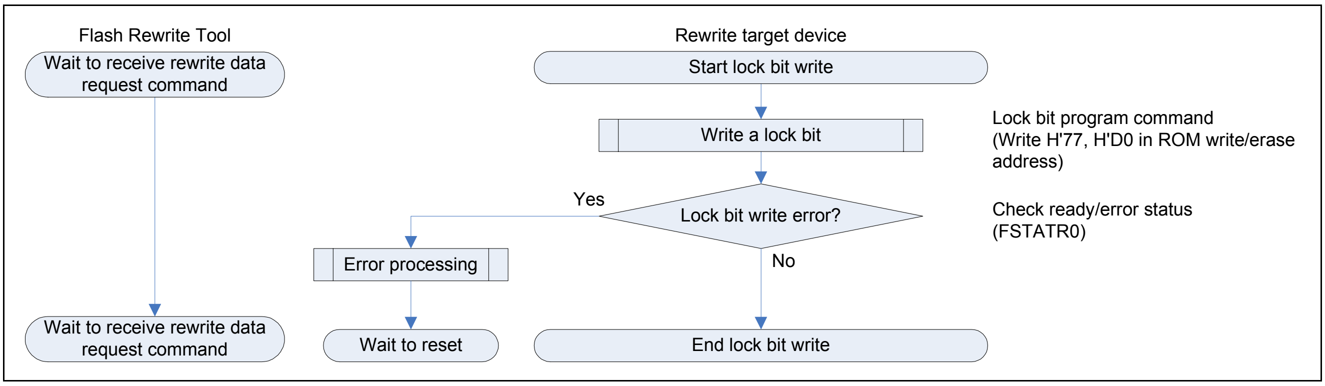

Flash Rewrite Tool
Wait to receive rewrite data request command
Wait to receive rewrite data request command
Rewrite target device
Start lock bit write
Write a lock bit
Lock bit program command (Write H'77, H'D0 in ROM write/erase address)
Lock bit write error?
Check ready/error status (FSTATR0)
Yes
No
Error processing
Wait to reset
End lock bit write

Figure 2.10 Lock Bit Write

2.6 Disabling Lock Bit Protection

Read the status of the lock bit for each block. If the block is protected, disable the lock bit protection. The flowchart for Disabling Lock Bit Protection is shown in Figure 2.11.

(1) Reading the current lock bit status

Use the lock bit read command to read the block protection status. If the block is protected (FLOCKST bit in the FSTATR1 register is 0), (2) is executed. If the block is not protected (FLOCKST bit in the FSTATR1 register is 1), the process ends.

If an error is detected, the process moves to "2.7 Error Processing (1)".

(2) Disabling lock bit protection

Set the FLOCKST bit in the FSTATR1 register to 1 and disable the lock bit protect of a block.

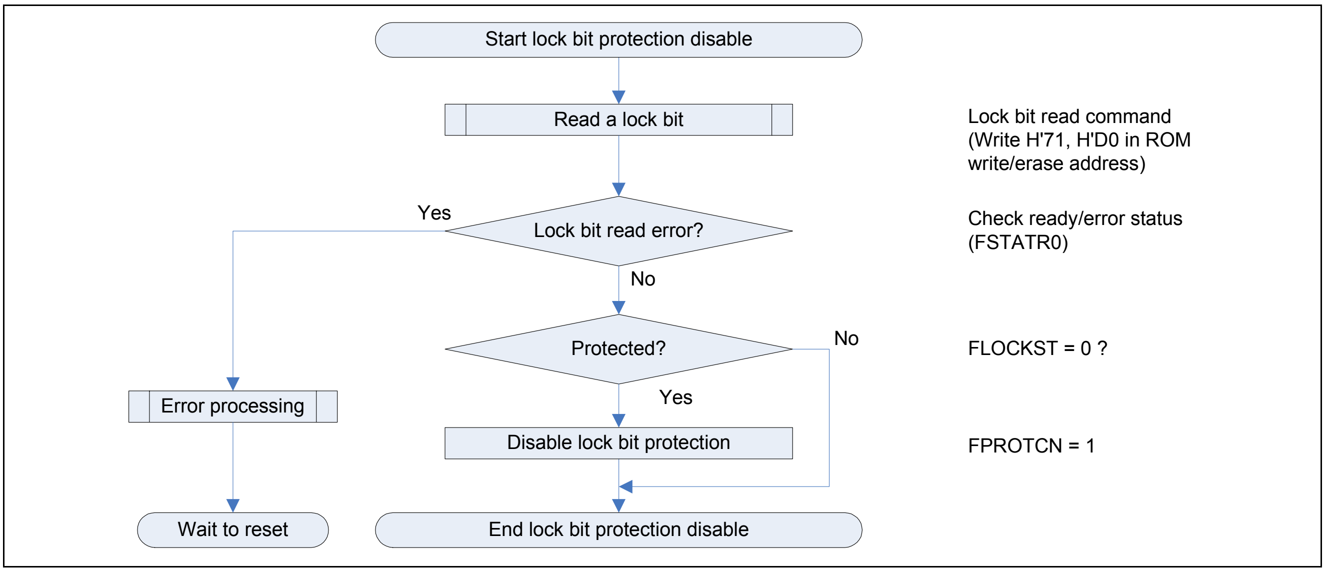


Figure 2.11 Disabling Lock Bit Protection

2.7 Error Processing

Explain about error processing during the ROM rewrite process. An error processing flowchart is shown in Figure 2.12.

(1) Error processing while ROM is in P/E mode
Abort the current process, reset the FCU, and wait until a hardware reset is executed.

(2) Error processing while ROM is in read mode
Abort the current process and wait until a hardware reset is executed.

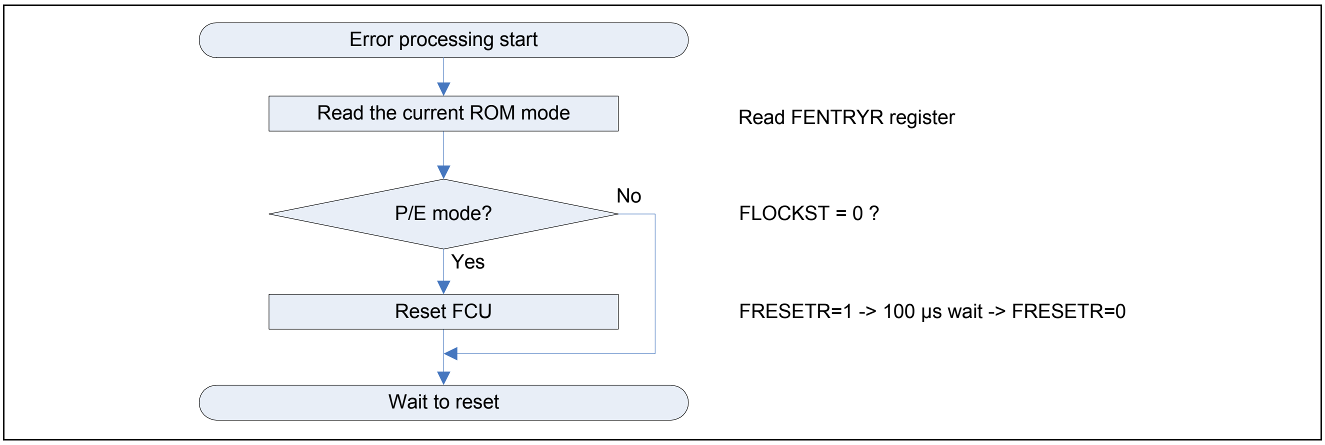


Figure 2.12 Error Processing

2.8 File Structure

The file structure of the P/E program is shown in Table 2.5.

Table 2.5 File Structure

File Name	Contents
dbsct.c	B section and D section setting file
env.inc	Address definition of exception event register and interrupt event register
main.c	Main function program
platform.h	Type declaration file
resetprg.c	Reset program
rom_rewrite_ubm.h	Macro definition file for the P/E program
sh7455_iodefine_20100805.h	SH7455 Group/SH7456 Group SFR definition file
stacksct.h	Stack size definition file
typedefine.h	Type declaration file (project generator created)
vect.inc	Vector definition file
vecttbl.src	Interrupt vector table definition file
vhandler.src	Interrupt handler program

2.9 Section setting

The section setting in the P/E program is shown in Table 2.6

Table 2.6 User Boot MAT Section Setting

Address	Map	Section Name	Explanation
H'0000 0000	Internal ROM	RSTHandler	Reset handler
H'0000 0800		INTHandler	Exception/Interrupt handler
		VECTTBL	Vector table
		INTTBL	Interrupt mask table
H'0000 1000		PResetPRG	Reset program
H'0000 2000		P_UB_PRG_TOP	P/E program storage area
		D_UB_data	Rewrite address data in ROM area
H'0000 3000		P	Program area
		C	Constant area
		C$BSEC	B section initialization table
		C$DSEC	D section initialization table
		D	Initialization data area
H'E500 E000	OL memory	R_UB_data_RAM	Rewrite address data in RAM area
		B	Non-initialized data area
		R	Initialization data area
		S	Stack address area
H'E520 0000	IL memory	P_UB_PRG_RAM	P/E program transfer and execution area

Sections mapped from ROM to RAM

ROM	RAM
D	R
P_UB_PRG_TOP	P_UB_PRG_RAM
D_UB_data	R_UB_data_RAM

2.10 Function List

The functions used in the P/E program are listed below.

Function name	can_init
Argument	None
Return value	None
Function	The CAN module and the I/O port used by the CAN module are initialized.
Note	None

Function name	ubrom_ram
Argument	None
Return value	None
Function	Transfer the P/E program stored in the user boot MAT area to IL memory and OL memory.
Note	Execute the can_init function before calling this function.

Function name	rom_rewrite_start
Argument	None
Return value	None
Function	Download rewrite data by CAN communication, and write 1008 Kbytes of data in the target user MAT area (ROM address: H '0000 4000 to H' 000F FFFF).
Note	Execute the can_init function before calling this function.

3. Reference Documents

- Hardware manual
  SH7455 Group, SH7456 Group User’s Manual: Hardware Rev.0.50 (R01UH0030EJ0050)
  Contact Renesas Electronics Corporation to inquire about the latest version.

Website and Support

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

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

REVISION HISTORY	SH7455 Group/SH7456 Group Flash rewrite (CAN communication)

Rev.	Date	Description	
		Page	Summary
1.00	Aug 27, 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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

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 one with a different type number, confirm that the change will not lead to problems.

   - The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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