R-CarM2N System Evaluation Board GOSE
RTP0RC7793SEB00010S
Mini-Monitor Manual (SPI Boot)
Rev. 0.01

Automotive Information System Business Division,
1st Solution Business Unit

Renesas Electronics Corp.

Revision History

Date	Revision	Description
July 02, 2014	0.01	Newly created and released.

Contents

1. Overview

1.1. Overview of Mini-Monitor

The mini-monitor is a monitoring program with limited functions. It is a simplified monitoring program with the purposes of displaying the memory contents, editing the memory contents, loading the user program, etc. at initial evaluation of the board and microcomputer.

1.2. Configuration of SPI Boot Program

This is a mini-monitor program that is initiated in SPI boot mode and runs on the DDR3 memory (H'40000000 and the subsequent addresses) in the R-CarM2N.

Along with the loader program for booting from the SPI flash memory and the system information (address of the destination for transfer and size of the mini-monitor), the mini-monitor program has been written to the SPI flash memory (U17, S25FL032) on the GOSE board.

The address space in the SPI flash memory is divided into the following three areas.

(1) Area for the loader program (sector: SA0) that is transferred to the secure RAM by an on-chip ROM program when booting up is in SPI boot mode.

(2) Area for the user program (SA2 and subsequent areas) that is transferred to and executed in the DDR3 memory by the loader program.

(3) System area (sector: SA1) required to initiate the program.

Note: For configuration and operation of the mini-monitor for SPI boot, see section 5, Overview of SPI Boot Mode Operations.
For more details, see the section on booting in the R-CarM2N Hardware Manual.

<Caution>

When using the mini-monitor of the GOSE board, do not change the contents of the SPI flash memory (U17, S25FL032).

If the contents of this memory are changed or deleted by using the CS_M command, the mini-monitor for SPI boot cannot be used.

1.3. Execution Area and Work Area

The execution area and work area of a mini-monitor program are shown in the following.

Operation of the mini-monitor is not guaranteed if data in these areas is damaged during execution of the mini-monitor.

(1) Program execution area and work area

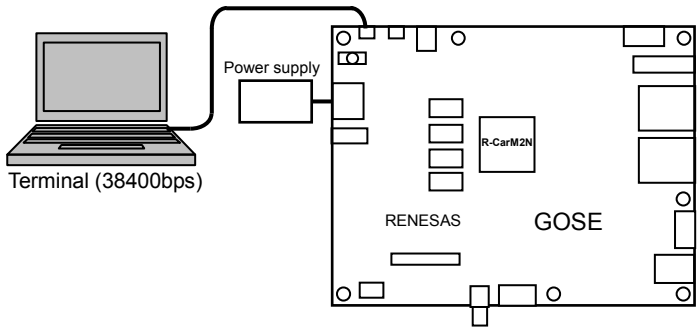
DDR3: H'4000_0000 to H'4027_FFFF (work area: H'4020_0000 to H'4027_FFFF)

2. Connection Configuration

Commands can be input and output by terminal software on the PC side. The USB cable (type A to micro B) is used to connect the PC and GOSE board. Before using the GOSE board, the USB driver (virtual COM port driver) needs to be installed into the PC. For details, refer to the “GOSE Setup Manual” and ”GOSE Hardware Manual”, and then install the USB driver and make the switch settings of the GOSE board.

< Connecting the PC and GOSE board >

For connection between the PC and GOSE board, use a USB cable (type A to micro B). Connect the type A end to the PC and the micro B end to CN19 (Debug Serial 0) on the GOSE board. * The SCIF0 is used as a serial interface in the R-CarM2N.



2.1. Serial Port Setting

Communication rate	38400 bps
Data length	8 bits
Parity	Not in use
Stop bit	1 bit
Flow control	Not provided

3. Startup

3.1. Switch Setting

Booting up of the board is initiated with the initial settings at shipment.

Only settings related to booting are given. The fill parts are the initial settings at shipment.

For more details, refer to the "GOSE Setup Manual" and "GOSE Hardware Manual".

Boot Mode and SPI Flash Memory Selection		Boot Device Selection			Connection of A[25]	SPI Selection (Connection of A[25])	Connection of A[24:20]	Remarks
		MD3 SW9 (4)	MD2 SW9 (3)	MD1 SW9 (2)	SW16	SW17	SW18	
QSPI boot (48.75 MHz)	U16, 512 Mbits	ON	OFF	ON	Pin 1 side	Pin 1 side	All OFF	
	U17, 32 Mbits					Pin 3 side		
QSPI boot (39 MHz)	U16, 512 Mbits	OFF	ON	ON		Pin 1 side		
	U17, 32 Mbits					Pin 3 side		

Note: The mini-monitor program in the SPI mode is only initiated with 16-Kbyte transfer. It is not initiated with 4-Kbyte transfer.

3.2. Startup Message

The following shows an example of the startup message. Note that the displayed contents may differ depending on the version.

When the ACC switch of the GOSE board is turned on, the startup message is displayed, indicating that the program is ready for input.

```
------------------------------------------------------------------------------

GOSE SPI_LOADER Vx.xx 20xx.xx.xx
 DEVICE S25FL032

GOSE MiniMonitor SPI_BOOT
 Work memory DRAM (H'40200000-)
 20xx.xx.xx Verx.xx ** Program on DRAM (H'40000000-)  ** >

------------------------------------------------------------------------------
```

Loader startup message

Mini-monitor startup message

Automotive Information System Business Division,
1st Solution Business Unit

Renesas Electronics Corp.

4. Mini-Monitor Commands

4.1. List of Commands

A list of commands is shown below. Note that the availability and specifications of commands may differ depending on the version.

No.	Input Command	Command Name	Description
1	D	memory Dump	Dumps memory contents.
2	DM	set&disp Dump Mode	Sets the bus width used when executing the memory Dump command.
3	F	Fill memory	Collectively changes the memory contents to the specified data.
4	FL	Fill memory(Long)	Collectively changes the memory contents to the specified data (longwords).
5	M	set Memory(byte)	Displays and changes the memory contents (bytes).
6	MW	set Memory(Word)	Displays and changes the memory contents (words).
7	ML	set Memory(Long)	Displays and changes the memory contents (longwords).
8	MV	Move memory	Moves the memory contents (copy).
9	L	Load program	Loads the user program.
10	G	Go program	Executes the user program.
11	R	display Registers	Displays a list of registers.
12	LF	Load Program to Flash memory (CN7:NOR Flash memory board)	Loads the user program to flash memory (command for writing the program to the flash memory board connected to CN7).
13	CF	Clear Flash memory (CN7: Flash memory board)	Erases all the data in flash memory (command for erasing all the data in the flash memory board connected to CN7).
14	LS	Load program to Spi flash (U16:S25FL512)	Loads the user program to the SPI flash memory (command for writing the program to U16 (S25FL512)).
15	CS	Clear Spi flash (U16:S25FL512)	Erases the entire data area in the SPI flash memory (command for erasing all the data in U16 (S25FL512)).
16	LS_M	Load program to Spi flash Minimonitor (U17:S25FL032)	Loads a program to the SPI flash memory to which the mini-monitor program has been written (command for writing a program to U17 (S25FL032)).
17	CS_M	Clear Spi flash Minimonitor (U17:S25FL032)	Erases all the data in the SPI flash memory to which the mini-monitor program has been written (command for erasing all the data in U17 (S25FL032)).
18	PCBV	read PCB Version spi flash	Reads the PCB version number in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
19	XPCBV	set PCB Version spi flash	Sets the PCB version number in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
20	INFO	read INFOrmation spi flash	Reads the information in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
21	PCBV_M	read PCB Version spi flash Minimonitor	Reads the PCB version number in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
22	XPCBV_M	set PCB Version spi flash Minimonitor	Sets the PCB version number in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
23	INFO_M	read INFOrmation spi flash Minimonitor	Reads the information in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
24	SUP	Scif speed UP	Improves the rate of transfer through the SCIF (to 921.6 Kbps).
25	H	help	Displays a list of commands.

4.2. Command Syntax

Described below are the definition of the command syntax and special characters used in the command specifications.

- Angle brackets (< >) indicate a mandatory parameter.
- A vertical bar separating one or more parameters (e.g.,{A|B|C}) indicate that only one of the parameters should be selected.
- Square brackets ([ ]) indicate an optional parameter.
- An underscore (_) indicates a visual spacing.
- (ret) indicates that the return (enter) key should be pressed.
- The first symbol (>) in the command line is a prompt, indicating that the program is ready for input.

4.3. Command Specifications

Described below are the command specifications. Note that the displayed contents may differ depending on the version.

4.3.1. D Command (memory Dump)

D_<sadr>[_<eadr>] (ret)
<sadr> : Dump start address
<eadr> : Dump end address (omittable)

Function	Dumps memory contents.
Description	(1) The memory contents from <sadr> to <eadr> are displayed. (2) When the <eadr> address is omitted, 256 bytes of memory contents from <sadr> are displayed. (3) After executing the memory Dump command, press enter key to display the subsequent 256 bytes of memory contents.
Usage example	>D_50000000 (ret) * The memory contents for 256 bytes from H'50000000 are dumped. 50000000 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000010 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000020 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000030 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000040 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000050 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000060 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000070 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000080 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 50000090 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000A0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000B0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000C0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000D0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000E0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" 500000F0 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 "UUUUUUUUUUUUUUUU" >

4.3.2. DM Command (set&disp Dump Mode)

DM_{ B|W|L } (ret)
<B> : Byte Dump Display
<W> : Word Dump Display
<L> : Long Word Dump Display
* With the parameter omitted, the dump width currently specified is displayed.

Function	Sets the bus width used when executing the memory Dump command and checks the setting value.
Description	(1) The bus width for data to be dumped when executing the memory Dump command should be selected and set in advance from among byte, word, and longword. Note that the initial bus width of the memory Dump command is set to byte. (2) <B> means byte. (3) <W> means word. (4) <L> means longword.
Usage example	(a) When setting the data bus width to longword >DM_L (ret) * The bus width for the memory Dump command is set to longword. > (b) When confirming the data bus width currently specified >DM DMmode = long * This means longword is set.

4.3.3. F Command (Fill memory)

F_<sadr>_<eadr>_<data> (ret) <sadr> : Fill start address <eadr> : Fill end address <data> : Fill data	
Function	Collectively changes the memory contents to the specified data (writing in byte units).
Description	The data specified in <data> is written in byte units to the area from <sadr> to <eadr>.
Usage example	>F_50000000_5000FFFF_55 (ret) * Data (H'55) is written to the area from H'50000000 to H'5000FFFF. >

4.3.4. FL Command (Fill memory (Long))

FL_<sadr>_<eadr>_<data> (ret) <sadr> : Fill start address <eadr> : Fill end address <data> : Fill data	
Function	Collectively changes the memory contents to the specified data (writing in longword units).
Description	The data specified in <data> is written in longword units to the area from <sadr> to <eadr>.
Usage example	>FL_50000000_5000FFFF_55AAFF00 (ret) * Data (H'55AAFF00) is written to the area from H'50000000 to H'5000FFFF. >

4.3.5. M Command (set Memory (byte))

M_<adr> (ret) <adr> : Edit Address	
Function	Displays and changes the memory contents (displaying and changing of data in byte units).
Description	(1) Display of the memory contents and the address to be changed <adr> are specified. (2) After executing the command, the memory contents of the specified address <adr> are displayed, and then the edit mode is entered. To change data, input the data to be changed and press the (ret) key. (3) In edit mode, the program proceeds to the subsequent address without changing data by pressing the (ret) key. The program returns to the previous address by inputting "^". (4) To end the edit mode, input "." (period).
Usage example	>M_50000000 (ret) *Memory contents at address H'50000000 are displayed and changed. 50000000 55 ? (ret) *Press the (ret) key to proceed to the subsequent address (no data is changed). 50000001 55 ? 12 (ret) *After inputting data (12), press the (ret) key to change the data to H'12. 50000002 55 ? 34 (ret) * After inputting data (34), press the (ret) key to change the data to H'34. 50000003 55 ? ^ *Input "^" to return to the previous address (to check the written data). 50000002 34 ? ^ *Input "^" to return to the previous address (to check the written data). 50000001 12 ? . *Input "." to end the procedure. >

4.3.6. MW Command (set Memory (Word))

MW_<adr> (ret) <adr> : Edit Address	
Function	Displays and changes the memory contents (displaying and changing of data in word units).
Description	Refer to "M Command".
Usage example	>MW_50000000 (ret) *Memory contents at address H'50000000 are displayed and changed. 50000000 5555 ? (ret) *Press the (ret) key to proceed to the subsequent address (no data is changed). 50000002 5555 ? ABCD (ret) *After inputting data (ABCD), press the (ret) key to change the data to H'ABCD. 50000004 5555 ? ^ *Input "^" to return to the previous address (to check the written data). 50000002 ABCD ? . *Input "." to end the procedure. >

4.3.7. ML Command (set Memory (Long))

ML_<adr> (ret) < adr > : Edit Address	
Function	Displays and changes the memory contents (displaying and changing of data in longword units).
Description	Refer to "M Command".
Usage example	>ML_50000000 (ret) *Memory contents at address H'50000000 are displayed and changed. 50000000 55555555 ? (ret) *Press the (ret) key to proceed to the subsequent address (no data is changed). 50000004 55555555 ? 12345678 (ret) *After inputting data (12345678), input the (ret) key to change the data to H'12345678. 50000008 55555555 ? ^ *Input "^" to return to the previous address (confirming of the written data). 50000004 12345678 ? . *Input "." to end the procedure. >

4.3.8. MV Command (Move memory)

MV_<sadr>_<dadr>_<len> (ret) <sadr> : Move start address <dadr> : Move Dest address <len> : length	
Function	Moves (copies) the memory contents.
Description	The amount of memory contents equal to the size specified in <len> from <sadr> are copied to the area starting with <dadr>. * <dadr> cannot be specified in area 0.
Usage example	>MV_50000000_60000000_2000(ret) * Data in addresses from H'50000000 to H'50001FFF is copied to addresses from H'60000000 to H'60001FFF. >

4.3.9. L Command (Load program)

L (ret)	
Function	Loads the user program.
Description	Data in the S-type format is loaded to memory using the file transmitting function of terminal software. * Key input from the terminal is not possible while load is in progress.
Usage example	>L (ret) please send ! ('.' & CR stop load) * After the left message is displayed, the user program to be loaded from the terminal is selected, and the file is transmitted. > * After loading is completed, the command prompt ">" is displayed.

4.3.10. G Command (Go program)

G_<start_adr> (ret) <start_adr> : program start Address	
Function	Executes the user program.
Description	The program is executed from the address specified in <start_adr>.
Usage example	>G_50000000 (ret) * The program is executed from the address H'50000000. >

4.3.11. R Command (display Registers)

R (ret)	
Function	Displays a list of registers (command for debugging)
Description	The register values for users are displayed. *The values to be displayed are those of the registers for users which are stored in work memory for debugging of a program. They are not the values of the general-purpose registers used for the mini-monitor program.
Usage example	>R (ret) * The register values for each user are displayed. CPSR 0000011F R0-5 00000000 00000000 00000000 00000000 00000000 00000000 R6-12 00000000 00000000 00000000 00000000 00000000 00000000 00000000 R13(SP) 600FFFF0 R14(LR) 00000000 R15(PC) 60000000 >

4.3.12. LF Command (Load program to Flash memory)

LF (ret)	
Function	Loads the user program to the flash memory (writing the program to the flash memory board connected to CN7).
Description	The MOT file (program) in the S-type format is written to flash memory using the file transmitting function of terminal software.
Usage example	>LF (ret) RAM(H'50000000-H'53FFFFFF) Clear.... please send ! ('.' & CR stop load) * After the message shown on the left is displayed, the user program to be written from the terminal to flash memory is selected, and the file is transmitted. Spansion flash was detected in area 1 (Flash Board) Flash write buffer size is 32 words. Flash Memory Erase Start * Flash memory is erased. ..... Erase Completed FLASH Memory (area 1) Program * The program is written. ..... Write Completed > * After writing is completed, the command prompt ">" is displayed.
Notes	■ This command is enabled only when the flash memory board is connected to CN7. The maximum program size that can be written is 1 Mbyte. ■ The program is written to the flash memory board connected to CN7, which is mapped to the system area 1 space (CS1 space). ■ The program that runs on the system area 0 space (CS0 space) when booting is from area 0 can only be written by this command. ■ The DDR3 memory area (from H'50000000) is used as an area to load the data (program) in the S-type format.

4.3.13. CF Command (Clear Flash memory)

CF (ret)	
Function	Erases all the data in flash memory (erasing all the data in flash memory board connected to CN7).
Description	All the data of the flash memory board connected to CN7, which is mapped to the system area 1 space (CS1 space), is erased.
Usage example	>CF (ret) FLASH MEMORY 64MB ALL CLEAR... Spansion flash was detected in area 1 (On Board Flash). Flash write buffer size is 32 words. On Board FLASH MEMORY ALL CLEAR... Please wait * Erasing in progress. It takes approximately four minutes to erase data (chip erase). Completed * After erasing is completed, "Completed" is displayed
Notes	■ This command is only enabled when the flash memory board is connected to CN7.

4.3.14. LS Command (Load program to Spi flash)

<table>
<tr><td colspan="2">LS (ret)</td></tr>
<tr><td>Function</td><td>Loads the user program to the SPI flash memory (writing the program to U16, S25FL512).</td></tr>
<tr><td>Description</td><td>This command is used to write the user loader program and user program to the SPI flash memory for users (U16, S25FL512).
The MOT file data (program) in the S-type format is written to the SPI flash memory using the file transmitting function of terminal software. The start address and program size of the user program are also written to sector 1 (SA1) in the SPI flash memory.</td></tr>
<tr><td>Usage example</td><td>
>LS (ret)
Load Program to Spiflash memory (U16:S25FL512S) * To enable U16 (S25FL512S), be sure to set switches according to these
SW18[5:1]=OFF! Setting OK? (Push Y key) messages and input Y. If the settings of switches are not made according to
SW16 1pin-Side! Setting OK? (Push Y key) the messages and Y is input in an attempt to proceed to the subsequent
SW17 1pin-Side! Setting OK? (Push Y key) procedure, there might be no response returned.

If the SPI device selected by switch settings and the connected SPI device do no match, the message shown below is displayed.
Check the switch settings and the command to be executed and re-execute the command.
SPI:READ DEVICE ID Error. Please check switch setting

Please select,SPI-FLASH Save Area.
1:Loader Program : Program to execute on SecureRAM
SPI_Address = H' 000_0000-H' 003_FFFF
2:User Program : Program to execute on DRAM or SecureRAM
SPI_Address = H' 008_0000-H' 3FF_FFFF

Select area(1-2)>2 * When the program to be written is the loader program and the user program (mini-monitor), input 1 and 2, respectively.
-- User Program ---------- * The flow of writing the user program is started.
Please Input User Program Start Address :<u>80000000</u> * Input the start address of the user program to be written.

Work RAM(H'50000000-H'53FFFFFF) Clear....
please send ! ('.' & CR stop load) * Select the user program to be written to the SPI flash memory from the terminal and transmit the program file.

SPI Data Clear(H'FF) Check :H'0080000-00BFFFF Clear OK?(y/n)
SPI Data Clear(H'FF) Check : OK * If any data has already been written to the target area, the message shown on the left is displayed. When the data can be erased, input “Y”.
After erasing is completed, ”Erase Completed” is displayed.
After that, check again. “OK” is displayed after checking is completed.
SAVE SPI-FLASH..... complete! * Writing is completed.

-- Save (Program Start Addres & Size) ----- * The flow of writing to SA1 is started.
SPI Data Clear(H'FF) :H'0040000-007FFFF Erasing..Erase Completed * Erasing of SA1
SAVE SPI-FLASH..... complete! * Writing to SA1 is completed.

=== SPI Flash Sector1 Save Information === * After writing to SA1 is completed, the written data is displayed.
Program Start Addres : H'80000000 * Start address of the user program (written to SPI address H’40000)
Program Size : H'0000D7C0 * Size of the user program (written to SPI address H’40004)
==========
</td></tr>
<tr><td>Notes</td><td>
■ Writing of loader program
• Select “1” from the selection menu to start writing the loader program.
• The loader program that runs on the Secure RAM (H'E630_0000 to H'E633_FFFF) can only be written by this command.
• The data write capacity is 16 Kbytes. <u>Do not load an MOT file data exceeding 16 Kbytes.</u>

■ Writing of user program
• Select “2” from the selection menu to start writing the user program.
• The program that runs on the DRAM (H'40000000 to H'7FFFFFFF) or Secure RAM (H'E6304000 to H'E633FFFF) area can only be written by this command. <u>Do not load an MOT file data exceeding the above range.</u>
• <u>After the message “Please Input User Program Start Address” has been displayed, input a start address of the MOT file to be loaded (smallest value) as the start address of the program. (This address is treated as the start address and branch address of the data transfer destination from SPI flash memory in the boot loader program of the mini-monitor.)
If the above-mentioned specifications are not satisfied, or the MOT file whose address differs from the input address is loaded, even if writing is completed, SPI boot is not initiated correctly from the loader program of the mini-monitor because the file has not been correctly written.</u>

■ The DDR3 (from H'50000000, from H'60000000) and Secure RAM are used as a work area.
</td></tr>
</table>

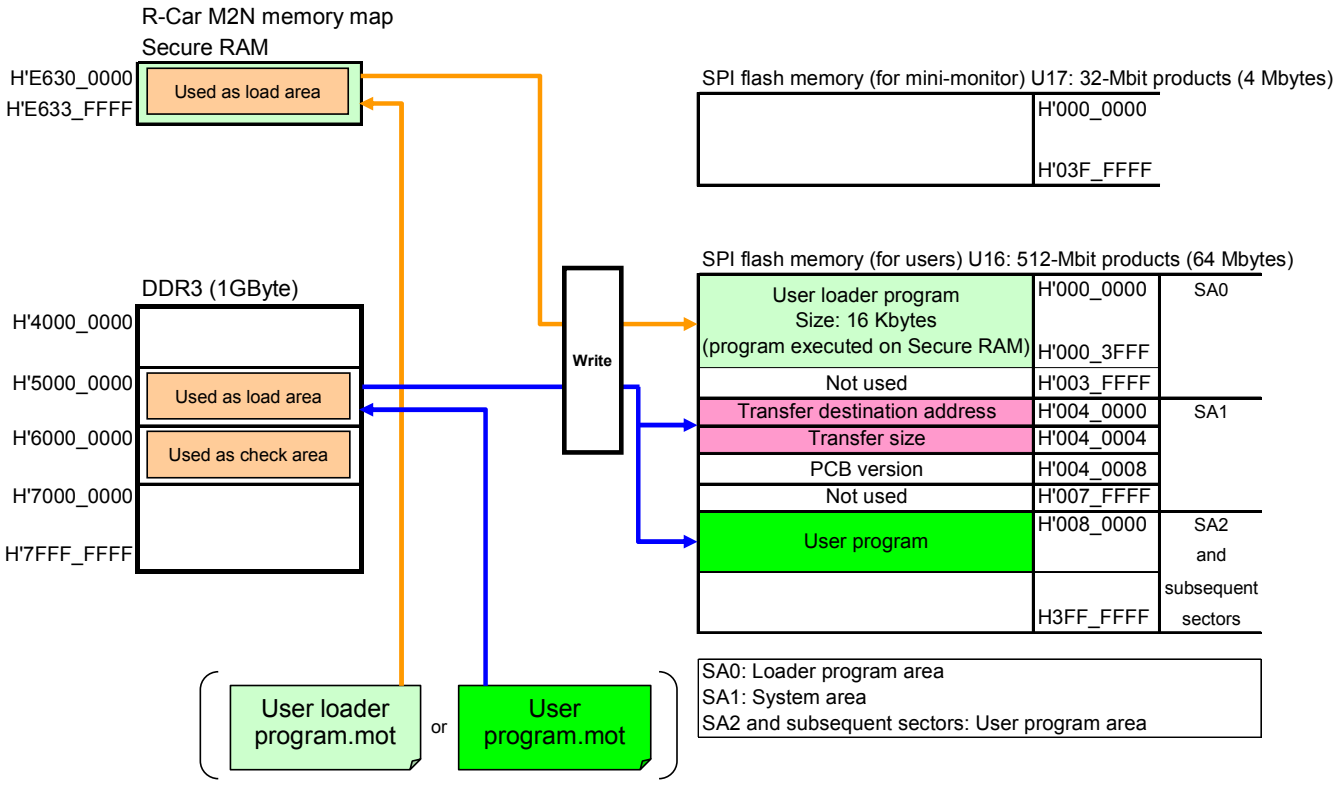
<Overview of write operations when executing a command>

■ Writing of loader program (selection menu "1")

The MOT file data is loaded without change to the Secure RAM, which is a loader program execution area, and then the amount of program equal to the size of the MOT file starting with the top of the Secure RAM is written to the SPI flash memory.

■ Writing of user program (selection menu "2")

The MOT file data is once stored in the DDR3 (H'50000000), and then the amount of data equal to the size of the MOT file is written to the SPI flash memory.



* Before writing of the program (before erasing of the sector), the MOT file is loaded and then the data in the SPI area expanded in sector units, whose size is as large as that of the MOT file, is loaded to the check area for SPI data to check the data.
If any data has already been written (other than H'FF), the message determining whether the data can be erased is displayed. When the data can be erased, the target sector is erased and then write operations are started.

4.3.15. CS Command (Clear Spi flash)

CS (ret)	
Function	Erases the entire data area in the SPI flash memory for users (**erasing of all the data in U16, S25FL512**).
Description	All the data in the SPI flash memory for users (U16:S25FL512) is erased.
Usage example	>CS (ret) ALL ERASE Spiflash memory (U16:S25FL512S) Clear OK?(y/n) ! * When the data can be erased, input "Y". SW18[5:1]=OFF! Setting OK? (Push Y key) * Similar to an LS command, be sure to set switches according to the SW16 1pin-Side! Setting OK? (Push Y key) messages and input Y. SW17 1pin-Side! Setting OK? (Push Y key) ERASE QSPI-FLASH (103sec[typ])..... complete! * After erasing is completed, "Complete" is displayed.

4.3.16. LS_M Command (Load program to Spi flash Minimonitor)

LS_M (ret)	
Function	Loads the mini-monitor to the SPI flash memory (writing the program to U17, S25FL032).
Description	This command is used to write the mini-monitor loader program and mini-monitor program to the SPI flash memory for the mini-monitor (U17, S25FL032). The MOT file data (program) in the S-type format is written to the SPI flash memory using the file transmitting function of terminal software. The top address and program size of the user program are also written to sector 1 (SA1) in the SPI flash memory.
Usage example	■ Usage example of writing of user program >LS_M (ret) Load Program to Spiflash memory (U17:S25FL032P) * To enable U17 (S25FL032P), be sure to set switches according to these SW18[5:1]=OFF! Setting OK? (Push Y key) messages and input Y. If the settings of switches are not made according to SW16 1pin-Side! Setting OK? (Push Y key) the messages and Y is input in an attempt to proceed to the subsequent SW17 3pin-Side! Setting OK? (Push Y key) procedure, there might be no response returned. If the SPI device selected by switch settings and the connected SPI device do no match, the message shown below is displayed. Confirm the switch settings and the command to be executed and re-execute the command. SPI:READ DEVICE ID Error. Please check switch setting --- Please select,SPI-FLASH Save Area. 1:Loader Program : Program to execute on Secure RAM SPI_Address = H' 000_0000-H' 000_FFFF 2:User Program : Program to execute on DRAM or SecureRAM SPI_Address = H' 002_0000-H' 03F_FFFF --- Select area(1-2)>2 * When the program to be written is the loader program and the user program (mini-monitor), input 1 and 2, respectively. -- User Program ---------- * The flow of writing the user program is started. Please Input User Program Start Address :**40000000** * Input the top address of the user program to be written. (Input 40000000 for the SPI boot mini-monitor program) Work RAM(H'50000000-H'503FFFFF) Clear.... please send ! ('.' & CR stop load) * Select the user program to be written to the SPI flash memory from the terminal and transmit the program file. SPI Data Clear(H'FF) Check :H'020000-02FFFF Clear OK?(y/n) * If any data has already been written to the target area, the SPI Data Clear(H'FF) Check : OK message shown on the left is displayed. When the data can be erased, input "Y". After erasing is completed, "Erase Completed" is displayed. After that, check again. "OK" is displayed after checking is completed. SAVE SPI-FLASH..... complete! * Writing is completed. -- Save (Program Start Addres & Size) ----- * The flow of writing to SA1 is started. SPI Data Clear(H'FF) :H'010000-01FFFF Erasing..Erase Completed * Erasing of SA1 SAVE SPI-FLASH..... complete! * Writing to SA1 is completed. === SPI Flash Sector1 Save Information === * After writing to SA1 is completed, the written data is displayed. Program Start Addres : H'40000000 * Start address of the user program (written to SPI address H'10000) Program Size : H'0000D7C0 * Size of the user program (written to SPI address H'10004) ==========
Notes	■ Writing of loader program • Select "1" from the selection menu to start writing the loader program. • The loader program that runs on the Secure RAM (H'E630_0000 to H'E633_FFFF) can only be written by this command. • The data write capacity is 16 Kbytes. **Do not load an MOT file data exceeding 16 Kbytes.** ■ Writing of user program • Select "2" from the selection menu to start writing the user program. • The program that runs on the DRAM (H'40000000 to H'7FFFFFFF) or Secure RAM (H'E6304000 to H'E633FFFF) area can only be written by this command. **Do not load an MOT file data exceeding the above range.** • **After the message "Please Input User Program Start Address" has been displayed, input a start address of the MOT file to be loaded (smallest value) as the start address of the program. (This address is treated as the start address and branch address of the data transfer destination from SPI flash memory in the boot loader program of the mini-monitor.)** **If the above-mentioned specifications are not satisfied, or the MOT file whose address differs from the input address is loaded, even if writing is completed, SPI boot is not initiated correctly from the loader program of the mini-monitor because the file has not been correctly written.** ■ The DDR3 (from H'50000000, from H'60000000) and Secure RAM are used as a work area.

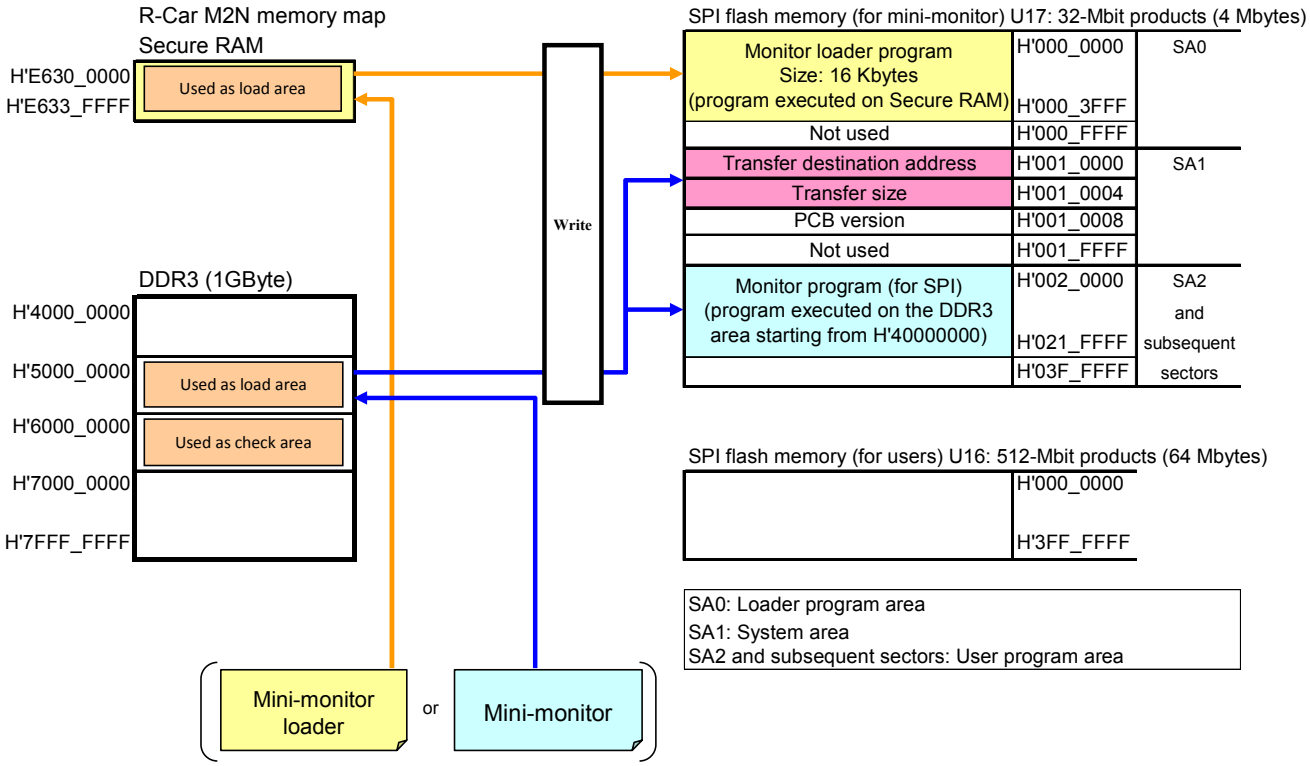
<Overview of write operations when executing a command>

■ Writing of loader program (selection menu “1”)

The MOT file data is loaded without change to Secure RAM, which is a loader program execution area, and then the amount of program equal to the size of the MOT file starting with the top of Secure RAM is written to the SPI flash memory.

■ Writing of user program (selection menu “2”)

Since the mini-monitor program runs on the DDR3 memory (H’4000_0000 and the subsequent addresses), the MOT file data is once stored in the DDR3 (H'50000000), and then the amount of data equal to the size of the MOT file is written to the SPI flash memory.



* Before writing of the program (before erasing of the sector), the MOT file is loaded and then the data in the SPI area expanded in sector units, whose size is as large as that of the MOT file, is loaded to the check area for SPI data to check the data.
If any data has already been written (other than H’FF), the message determining whether the data can be erased is displayed. When the data can be erased, the target sector is erased and then write operations are started.

4.3.17. CS_M Command (Clear Spi flash Minimonitor)

CS_M (ret)	
Function	Erases the entire data area in Spi_Flash for the mini-monitor (**erasing of all the data in U17, S25FL032**).
Description	All the data in Spi_Flash where the mini-monitor has been written is erased.
Usage example	>CS_M (ret) ALL ERASE Spiflash memory (U17:S25FL032P) Clear OK?(y/n) ! * When the data can be erased, input “Y”. SW18[5:1]=OFF! Setting OK? (Push Y key) * Similar to an LS command, be sure to set switches according to the SW16 1pin-Side! Setting OK? (Push Y key) messages and input Y. SW17 3pin-Side! Setting OK? (Push Y key) ERASE QSPI-FLASH (32sec[typ])....complete! * After erasing is completed, ”complete” is displayed.

4.3.18. PCBV Command (read PCB Version spi flash)

PCBV (ret)	
Function	Reads the PCB version number in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
Description	Reads the PCB version number stored in SA1 of the SPI flash memory for users (U16, S25FL512).
Usage example	>PCBV (ret) === SPI(S25FL512) PCB Ver Save Information ====== SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 1pin-Side! Setting OK? (Push Y key) PCB Version : H'00000000 * The PCB version is displayed. (example) ========== H'00000000: PCB/0 (RTP0RCxxxxSEB00010S) H'00000001: PCB/1 (RTP0RCxxxxSEB00011S) H'00000002: PCB/2 (RTP0RCxxxxSEB00012S)

4.3.19. XPCBV Command (set PCB Version spi flash)

XPCBV (ret)	
Function	Sets the PCB version number in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
Description	Changes the PCB version number stored in SA1 of the SPI flash memory for users (U16, S25FL512).
Usage example	>XPCBV (ret) === SPI(S25FL512) PCB Ver Save Information ====== SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 1pin-Side! Setting OK? (Push Y key) Now PCB Version : H'00000000 * The PCB version is displayed. Change PCB Version ?(Y/N)y * Enter Y if you wish to change the version. Please Input New PCB Version * If Y is entered, Set Data :2 * Input the desired value (up to eight hexadecimal digits). SPI Data Clear(H'FF):H'040000-07FFFF Erasing..Erase Completed SAVE SPI-FLASH..... complete! === SPI Flash PCB Ver Save Information ========= PCB Version : H'00000002 * The PCB version after the change is displayed. ========== (Data written to H'40008 (address of the SPI flash memory))
Note	■ In general, do not change the value from that which was written at shipment. If a value needs to be changed for some reason, be sure to set a value which is suitable for the board version indicated on the board. Example: For RTP0RCxxxxSEB00012S, set the underlined part “2”.

4.3.20. INFO Command (read INFOrmation spi flash)

INFO (ret)	
Function	Reads the information in the system information area (SA1) of the SPI flash memory for users (U16, S25FL512).
Description	Reads the information (address of the destination for transfer, size, PCB version number) stored in SA1 of the SPI flash memory for users (U16, S25FL512).
Usage example	>INFO (ret) === SPI(S25FL512) SA1 System Area Information === SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 1pin-Side! Setting OK? (Push Y key) Program Start Address : H'40000000 * System information is displayed. Program Size : H'0000D6B0 PCB Version : H'00000002 ==========

4.3.21. PCBV_M Command (read PCB Version spi flash Minimonitor)

PCBV_M (ret)	
Function	Reads the PCB version number in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
Description	Reads the PCB version number stored in SA1 of the SPI flash memory for the mini monitor (U17, S25FL032).
Usage example	>PCBV_M (ret) === SPI(S25FL032P) PCB Ver Save Information ====== SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 3pin-Side! Setting OK? (Push Y key) PCB Version : H'00000000 * The PCB version is displayed. (example) ========== H'00000000: PCB/0 (RTP0RCxxxxSEB00010S) H'00000001: PCB/1 (RTP0RCxxxxSEB00011S) H'00000002: PCB/2 (RTP0RCxxxxSEB00012S)

4.3.22. XPCBV_M Command (set PCB Version spi flash Minimonitor)

XPCBV_M (ret)	
Function	Sets the PCB version number in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
Description	Changes the PCB version number stored in SA1 of the SPI flash memory for the mini monitor (U17, S25FL032).
Usage example	>XPCBV_M (ret) === SPI(S25FL032P) PCB Ver Save Information ====== SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 3pin-Side! Setting OK? (Push Y key) Now PCB Version : H'00000000 * The PCB version is displayed. Change PCB Version ?(Y/N)y * Enter Y if you wish to change the version. Please Input New PCB Version * If Y is entered, Set Data :2 * Input the desired value (up to eight hexadecimal digits). SPI Data Clear(H'FF):H'010000-01FFFF Erasing..Erase Completed SAVE SPI-FLASH..... complete! === SPI Flash PCB Ver Save Information ========= PCB Version : H'00000002 * The PCB version after the change is displayed. ========== (Data written to H'10008 (address of the SPI flash memory))
Notes	■ In general, do not change the value from that which was written at shipment. If a value needs to be changed for some reason, be sure to set a value which is suitable for the board version indicated on the board. Example: For RTP0RCxxxxSEB00012S, set the underlined part “2”.

4.3.23. INFO_M Command (read INFOrmation spi flash Minimonitor)

INFO_M (ret)	
Function	Reads the information in the system information area (SA1) of the SPI flash memory for the mini monitor (U17, S25FL032).
Description	Reads the information (address of the destination for transfer, size, PCB version number) stored in SA1 of the SPI flash memory for the mini monitor (U17, S25FL032).
Usage example	>INFO_M (ret) === SPI(S25FL032P) SA1 System Area Information === SW18[5:1]=OFF! Setting OK? (Push Y key) * Set switches according to the messages and input Y. SW16 1pin-Side! Setting OK? (Push Y key) SW17 3pin-Side! Setting OK? (Push Y key) Program Start Address : H'40000000 * System information is displayed. Program Size : H'0000D6B0 PCB Version : H'00000002 ==========

4.3.24. SUP Command (Scif speed UP)

SUP (ret)	
Function	Improves the rate of SCIF transfer.
Description	Changes the baud rate of the SCIF to 921.6 Kbps.
Usage example	>sup Scif speed UP Change to 921.6Kbps baud rate setting of the SCIF. OK? (y/n) * Enter Y to raise the transfer rate. Please change to 921.6Kbps baud rate setting of the terminal. * Change the baud rate to 921.6 Kbps. After changing the setting for the baud rate of the terminal, key input is again possible. * The command prompt ">" is displayed in response to pressing the return key, and the command reception state can be checked.

4.3.25. H Command (Help)

H (ret)	
Function	Displays a list of commands.
Description	Displays a list of commands of the mini monitor.
Usage example	>H D {sadr {eadr}} memory dump (DM sets dump size) DM {B|W|L} set&disp dump mode F [sadr] [eadr] [data] fill memory FL [sadr] [eadr] [data] fill memory(LONG) M [adr] set memory(BYTE) MW [adr] set memory(WORD) ML [adr] set memory(LONG) MV [sadr] [dadr] [len] move memory L load program G {start_adr} go program R display register LF load Program to Flash memory CF Clear Flash memory LS Load program to Spi flash (U16:S25FL512) CS Clear Spi flash (U16:S25FL512) LS_M Load program to Spi flash Minimonitor (U17:S25FL032) CS_M Clear Spi flash Minimonitor (U17:S25FL032) PCBV read PCB Version (U16:S25FL512) XPCBV set PCB Version (U16:S25FL512) INFO read Information (U16:S25FL512) PCBV_M read PCB Version (U17:S25FL032) XPCBV_M set PCB Version (U17:S25FL032) INFO_M read Information (U17:S25FL032) SUP Scif speed UP (Change to 921.6Kbps baud rate setting) H help >

5. Overview of SPI Boot Mode Operations

A program to control the SPI has been written to the boot ROM in the R-CarM2N. When the R-CarM2N is booted up in SPI boot mode, the program written to the external SPI flash memory (SPI addresses: H'00000 to H'03FFF (for 16-Kbyte transfer)) is transferred to Secure RAM (H'E6300000 to H'E6303FFF) of the R-CarM2N. After that, the CPU fetches instructions from the Secure RAM address (from H'E6300000) and the program is executed.

5.1. Configuration and Operation Details of Mini-Monitor for SPI Boot

The mini-monitor for SPI boot consists of the following two programs.

(1) Loader program

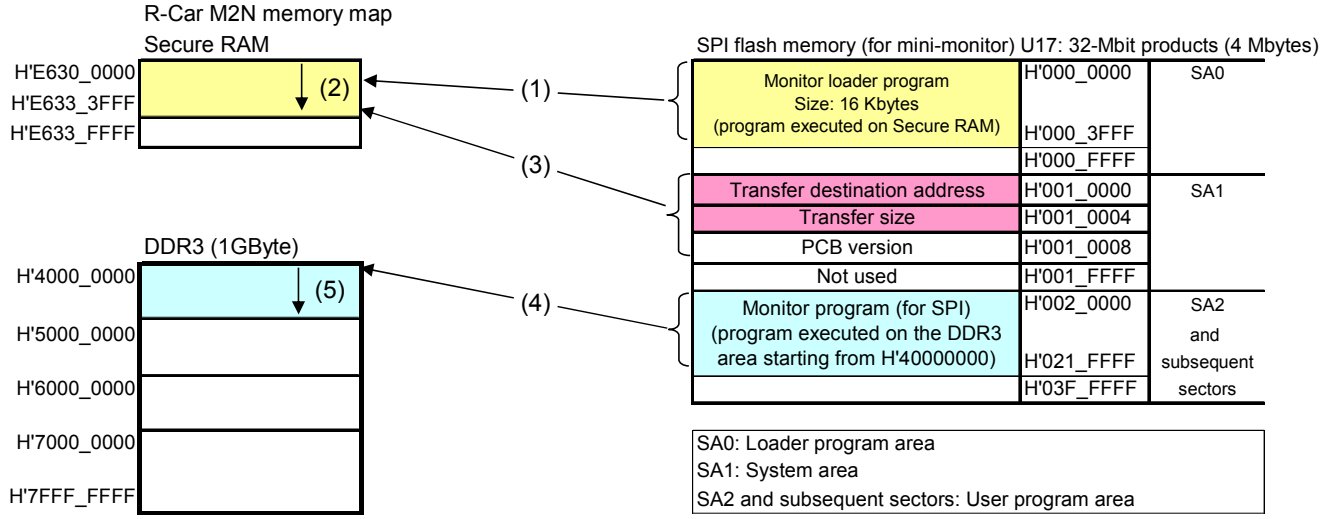
This is a program which is executed at SPI boot and runs on Secure RAM.

- Initializes on-chip modules (PFC, GPIO, LBSC, DDR3, SCI, etc.).
- Reads sector 1 (SA1) in SPI flash memory to obtain the information on the transfer destination address and transfer size.
- Copies the amount of data equal to the transfer size from sector 2 (SA2) and the following sectors in SPI flash memory to the transfer destination address. These sectors contain the mini-monitor program, and the destination address for the program is H'40000000.
- Starts execution of the mini-monitor program by changing the PC setting to the transfer destination address (H'40000000 for the mini-monitor program).

(2) Mini-monitor (program)

This is the mini-monitor program for SPI boot without the initialization part, and it runs on the DDR3 memory.

5.2. Schematic Diagram of SPI Boot Operations



<Description of operations>

(1)	(a) The power is turned on in SPI boot mode. When booting is from the SPI flash memory, 16 Kbytes of data starting from the address H'00000 of the SPI flash memory is transferred to Secure RAM. (This operation is performed automatically by the boot ROM program in the R-CarM2N when booting is from the SPI flash memory.)
(2)	(b) After data transfer is completed, the program is started from the top of the transferred loader program. (c) The initial settings for the PFC, GPIO, LBSC, DDR3, SCIF, etc. are made, and the startup message of the loader program is displayed on the terminal.
(3)	(d) The loader program reads the information on the transfer destination address and transfer size from the SA1 area in the SPI flash memory.
(4)	(e) The amount of data equal to the transfer size is copied from SA2 and the subsequent sectors in the SPI flash memory to the transfer destination address by using the information obtained in step (d). These sectors contain the mini-monitor program, and the destination address for the program is H'40000000.
(5)	(f) Starts execution of the mini-monitor program by changing the PC setting to the transfer destination address (H'40000000). (g) The mini-monitor is started up and the startup message is displayed on the terminal.

Note) Operations described above are for the loader and main program of the mini-monitor written by an LS_M command. The operations described in steps (c) and the subsequent steps differ depending on the loader program created by a user and the structure of the user program.