RZ/N2L Group

RZ/N2L Industrial Network SOM Kit Application Note: Serial Host interface library

Introduction

This application note explains the Serial Host interface library for achieved shared memory (Referred to as DPRAM in this document) between master and slave device and shows an example of shared memory using the Serial Host interface library.

RZ/N2L Group

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Overview

By using serial host interface (SHOSTIF) function on RZ/N2L, it is possible to access to internal resources (SystemSRAM, MBXSEMRegister, SHOSTRegister) from out SPI master directly. And using mail box and semaphore (MBXSEM) functions, it is possible to achieve exclusive control between internal CPU (Cortex-R52) and out host CPU and interrupt output using mail box.

By using the above functions, it is possible to achieve shared memory between out SPI master and RZ/N2L. System configuration example by using RZ/T2L as out SPI master is shown in Fig.1.1.

This library excuses R/W to shared memory and notice treatment to CPU.

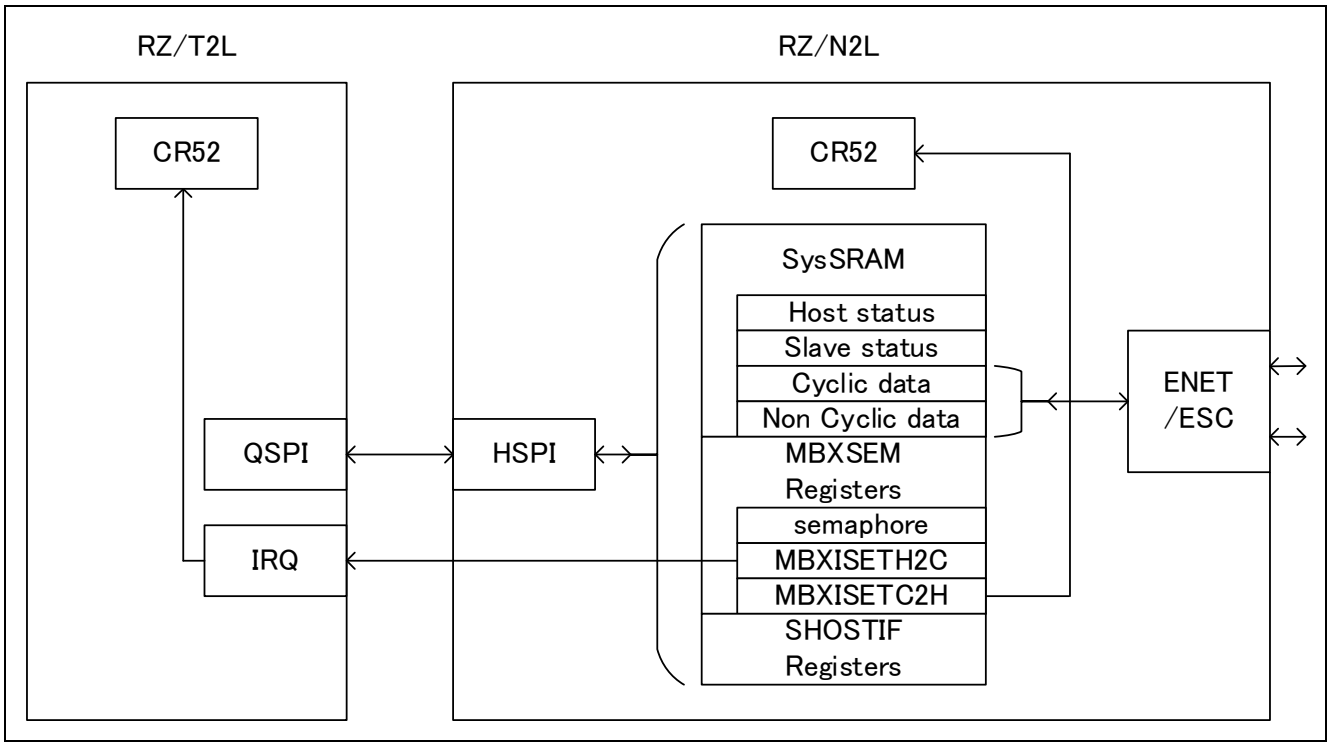

Figure 1-1 System configuration example

2. Operation confirmation condition

This library confirms the operation by the following conditions.

Table 2-1 Operation confirmation condition

Item		RZ/T2L	RZ/N2L
Board		Renesas Starter Kit+ for RZ/T2L	RZ/N2L Industrial Network SOM Kit (YCONNECT-IT-RZN2L)
Development environment	Renesas GCC	e2 studio_2023-01	e2 studio_2023-04
	IAR CC	IAR Embedded Workbench for Batch for ARM:9.32.1+RZ/T2L	IAR Embedded Workbench for ARM:9.32.2
FSP		RZ/T2 FSP v1.1.0	RZ/N2L FSP v1.2.0
Compiler	Renesas GCC	9.3.1.20200408	
	IAR CC	9.32.1.338	9.32.2
Emulator	Renesas GCC	SEGGER J-Link Base Ver.11.0	
	IAR CC	IAR systems I-jet	

2.1 Renesas Starter Kit+ for RZ/T2L

2.1.1 Configuration circuit setting

2.1.1.1 Switch

Table 2-2 Switch setting

SW	1	2	3	4	5	6	7	8	9	10
4	ON	ON	OFF	-	-	OFF	OFF	-	-	-
8	-	-	-	-	-	-	OFF	-	ON	-

2.1.1.2 Jumper

Table 2-3 Jumper setting

Connector	Setting	Function
CN17	Short 1-2	VCC1833_2 = 3.3V
CN32	Short 1-2	VCC1833_3 = 3.3V
CN33	2-3 Short	XSPI1_CS0# connects to CS# of expandable SPI connector (CN28)

2.1.1.3 Option link

Table 2-4 Option link setting

Number	Setting		Function
R201	R201	R202	P09_7 uses as IRQ12 of expandable SPI connector
R202	No mount (default mount)	Mount (default no mount)	

2.1.2 Peripheral functions

Table 2-5 Peripheral functions setting

Peripheral functions	Functions	Setting value
Expandable serial peripheral interface (xSPI)	Channel	xSPI1
	Mode	QSPI
	Protocol mode	4s-4s-4s
	Transaction submit	Manual command mode
	Transfer speed	12.5MHz
Interrupt controller (ICU)	Channel	IRQ9
	Detect mode selection	Negative edge
	Noise filter	Available
	Noise filter sampling clock selector	64 divider

2.2 RZ/N2L Industrial Network SOM Kit

2.2.1 Configuration circuit setting

2.2.1.1 Jumper

Table 2-6 Jumper setting

	Setting	Function
J2	Short 1-2	HSPI_INT# is avalable

2.2.1.2 Option link

Table 2-7 Option link setting

Number	Setting	Function
R102	Not mount	Use PMOD (J14) connector as UART function.
R103	Not mount	
R104	Mount	
R105	Not mount	
R106	Mount	
R107	Mount	
R108	Not mount	
R109	Mount	
R110	Not mount	
R111	Mount	

2.2.2 Peripheral functions

Table 2-8 Peripheral function setting

Peripheral functions	Functions	Setting value
Serial host interface (SHOSTIF)	SPI flame format	SPI Quad mode
	Addressing mode	32 bit
	Byte swapping mode	byte swapping
	Serial clock phase	Toggling on middle of first bit of serial clock
	Serial clock polarity	Serial clock on non-active status is Low
	Transmit FIFO empty interrupt mask	Masking txe_intr interrupt
	Receive FIFO overflow interrupt mask	No masking rxf_intr interrupt
	Receive FIFO full interrupt mask	No masking txu_intr interrupt
	Transmit FIFO underflow mask	No masking txu_intr interrupt
	AHB error interrupt mask	No masking ahbe_intr interrupt
	SPI master error interrupt mask	No masking spime_intr interrupt
Mailbox and semaphore (MBXSEM)	Reading clear function of SEM0 register	Available

2.3 Connection between boards

Table 2-9 Connection between boards

Renesas Starter Kit+ for RZ/T2L				**RZ/N2L Industrial Network SOM Kit**			
CN	Pin	Port	Terminal functions	Connector	Pin	Port	Terminal functions
28	1	-	GROUND	J15	2	--	GROUND
	3	P17_7	XSPI1_CKP		10	P14_2	HSPI_CK
	4	P18_2	XSPI1_CS0#		4	P16_0	HSPI_CS#
	9	P17_4	XSPI1_IO3		7	P16_3	HSPI_IO3
	10	P17_3	XSPI1_IO2		5	P16_2	HSPI_IO2
	11	P17_0	XSPI1_IO1		6	P14_3	HSPI_IO1
	12	P18_0	XSPI1_IO0		3	P14_1	HSPI_IO0
	2	P09_7	IRQ12		9	P14_5	HSPI_INT#
	13	P08_6	IRQ9	J12	16	P13_7	MBX_HINT#

3. Library

3.1 Overview

This library is able to execute R/W exclusively internal resources (System SRAM, MBXSEM Register, SHOST Register) in RZ/N2L through API function and possible to acknowledge updating memory by Callback. Library overview is shown in Figure 3-1

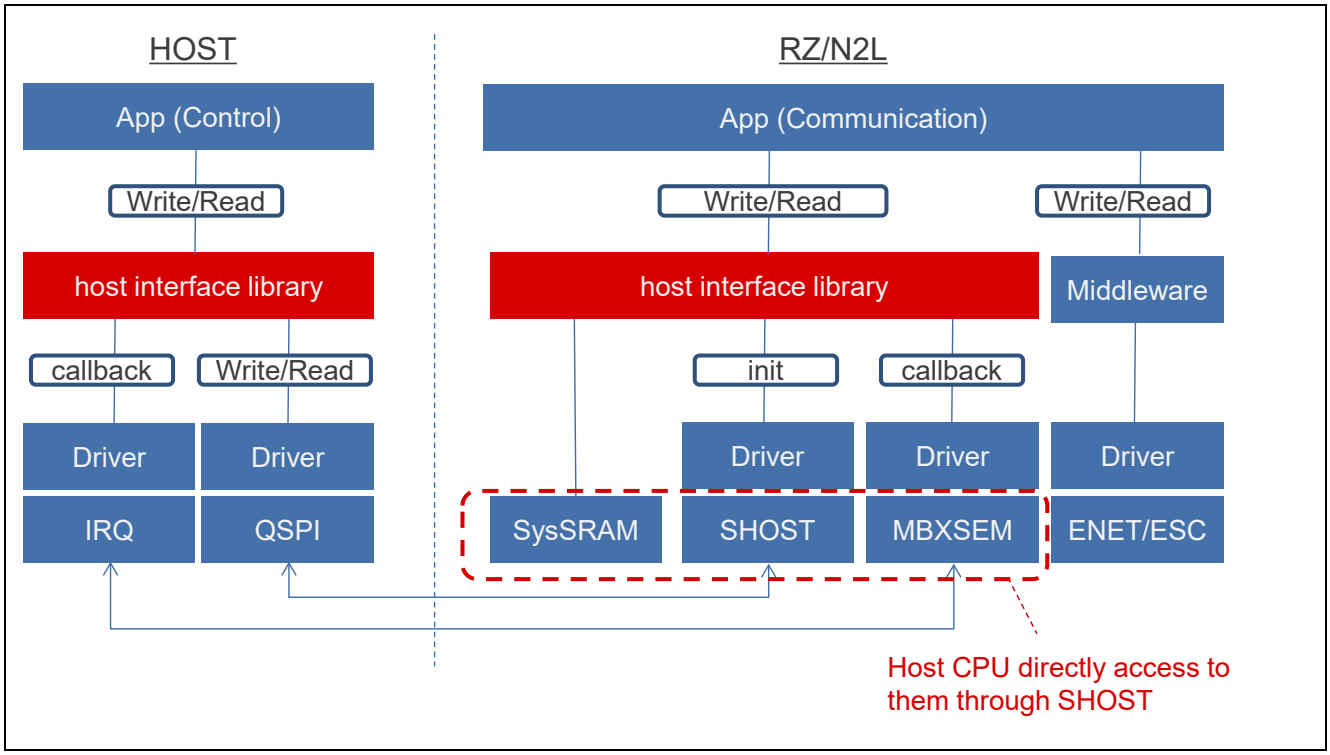

Figure 3-1 Library overview

3.2 Address view

Address of internal resources in RZ/N2L is shown on Table 3-1.

Table 3-1 Address

Address from HOST (32-bit addressing mode)	Address from RZ/N2L	Assignment
0x000000-0x01FFFF	0x30000000-0x3007FFFC	System SRAM (512KB)
0x020000-0x03FFFF	0x30080000-0x300FFFFC	System SRAM (512KB)
0x040000-0x05FFFF	0x30100000-0x3017FFFC	System SRAM (512KB)
0x060000-0x0603FF	0x80240000-0x80240FFC	MBXSEM Registers (4KB)
0x060400-0x0607FF	0x80241000-0x80241FFC	SHOST Registers (4KB)

3.3 File configuration

File configuration of this library is shown on Table 3-2

Table 3-2 file organization

File name	Feature
libSHOSTIF_LIBRARY.a	Main library
r_host_interface.h	API definition, configuration, prototype declaration
r_host_interface_cfg.h	User setting file

3.4 Using peripheral modules

Table 3-3 Using peripheral module

HOST / RZ/N2L	Peripheral module
HOST	xSPI
	IRQ
RZ/N2L	SHOSTIF
	MBXSEM

4. API explanation

4.1 Constants

4.1.1 Definition table

Table 4-1 Configuration table

Configuration	Type	Definition	Value	Explanation
API returned value	dpram_error_t	DPRAM_SUCCESS	0x00	API normal end
		DPRAM_WAIT_INIT	0x01	API no initialization
		DPRAM_ADDR_ERR	0x02	Out of address range
		DPRAM_LOCKED	0x03	Using resources
DPRAM status	dpram_status_t	DPRAM_STATUS_RESET	0x00	DPRAM initialization
		DPRAM_STATUS_INIT	0x01	Initialization completion of peripheral function
		DPRAM_STATUS_READY	0x02	DPRAM operable
		DPRAM_STATUS_RUN	0x03	Application execution

4.1.2 Driver functions

This library is necessary to register on function pointer of global variable number regarding peripheral driver functions calling from library. Register functions corresponding function pointer on driver side.

Table 4-2 Global variable number table

Configuration	Variable	Type	Member	Explanation
Xspi driver function pointer	g_xspi_api	st_xspi_api_t	void (*open)(void)	xspi driver Open function
			void (*write)(uint32_t * p_src, uint32_t * p_dst, uint32_t cnt);	xspi driver Write function p_src：Pointer of writing source address p_dst：Pointer of writing forward address cnt：Continuous writing count
			void (*read)(uint32_t * p_src, uint32_t * p_dst, uint32_t cnt);	xspi driver Read function p_src：Pointer of reading source address p_dst：Pointer of reading forward address cnt：Continuous reading count
			void (*close)(void);	xspi driver Close function
IRQ driver function pointer	g_irq_api	st_irq_api_t	void (*open)(void);	IRQ driver Open function
			void (*close)(void);	IRQ driver Close function
SHOSTIF driver function pointer	g_shostif_api	st_shostif_api_t	void (*open)(void);	SHOSTIF driver Open function
			void (*close)(void);	SHOSTIF driver Close function
MBXSEM driver function pointer	g_mbxsem_api	st_mbxsem_api_t	void (*open)(void);	MBXSEM driver Open function
			void (*close)(void);	MBXSEM driver Close function
Calling back function pointer	g_call_back_api	st_ call_back_func_t	void (*dpram_open)(void)	Calling back of DPRAM Open function
			void (*dpram_read)(void);	Calling back of DPRAM Read function
			void (*dpram_write)(void);	Calling back of DPRAM Write function
			void (*dpram_trigger)(void);	Calling back of DPRAM interrupt function

4.2 API functions

Table 4-3 Functions table

Function name	Features
R_DPRAM_Open	Execute initialization of using peripheral functions
R_DPRAM_Write	Write to System SRAM, SHOSTIF register and MBXSEM register
R_DPRAM_Read	Read from System SRAM, SHOSTIF register and MBXSEM register
R_DPRAM_SetTrigger	Acknowledge updating the other
R_DPRAM_ClearTrigger	Clear updating acknowledgement
R_DPRAM_Close	Module stop for using peripheral functions
R_DPRAM_GetStatus	Get DPRAM status
R_DPRAM_SetStatus	Set DPRAM status

4.2.1 R_DPRAM_Open

R_DPRAM_Open		
Overview	DPRAM initial function	
Header file	r_host_interface.h	
Declaration	void R_DPRAM_Open(void)	
Description	• Initialization of peripheral function for DPRAM control • Updating DPRAM initialization status	
Arguments	void	
Returned value	dpram_error_t	API normal end API no initialization Using resources

4.2.2 R_DPRAM_Write

R_DPRAM_Open		
Overview	DPRAM writing function	
Header file	r_host_interface.h	
Declaration	void R_DPRAM_Write(uint8_t const * const p_src, uint8_t * const p_dst, uint32_t byte_cnt)	
Description	• Write byte_cnt words from p_src address to p_dst address on DPRAM • After judging DPRAM status, acknowledge returned value in initialization • After checking address range, acknowledge returned value in case of out of address range • Controlling semaphore, acknowledge returned value in case of locking	
Arguments	uint8_t const * const p_src uint8_t * const p_dst uint32_t byte_cnt	Pointer of write original address Pointer of write forwarding address Writing byte counts
Returned value	dpram_error_t	API normal end API no initialization Out of address range Using resources

4.2.3 R_DPRAM_Read

R_DPRAM_Read		
Overview	DPRAM reading function	
Header file	r_host_interface.h	
Declaration	void R_DPRAM_Read(uint8_t const * const p_src, uint8_t * const p_dst, uint32_t byte_cnt)	
Description	• Read byte_cnt words from p_src address to p_dst address on DPRAM • After judging DPRAM status, acknowledge returned value in initialization • After checking address range, acknowledge returned value in case of out of address range • Controlling semaphore, acknowledge returned value in case of locking	
Arguments	uint8_t const * const p_src	Pointer of read original address
	uint8_t * const p_dst	Pointer of read forwarding address
	uint32_t byte_cnt	Reading byte counts
Returned value	dpram_error_t	API normal end API no initialization Out of address range Using resources

4.2.4 R_DPRAM_SetTrigger

R_DPRAM_SetTrigger		
Overview	Acknowledgement to host/slave function	
Header file	r_host_interface.h	
Declaration	void R_DPRAM_SetTrigger(void)	
Description	• Acknowledgement from host to slave or from slave to host using MBXSEM function on RZ/N2L • After judging DPRAM status, acknowledge returned value in case of initialization • Controlling semaphore, acknowledge returned value in case of locking	
Arguments	void	
Returned value	dpram_error_t	API normal end API no initialization Using resources

4.2.5 R_DPRAM_ClearTrigger

R_DPRAM_ClearTrigger		
Overview	Clearing acknowledgement to host/slave function	
Header file	r_host_interface.h	
Declaration	void R_DPRAM_ClearTrigger(void)	
Description	• Clearing acknowledgement from host to slave or from slave to host using MBXSEM function on RZ/N2L • Controlling semaphore, acknowledge returned value in case of locking	
Arguments	void	
Returned value	dpram_error_t	API normal end Using resources

4.2.6 R_DPRAM_Close

R_DPRAM_Close		
Overview	Stopping host interface library function	
Header file	r_host_interface.h	
Declaration	Dpram_error_t R_DPRAM_Close(void)	
Description	• Module stop of peripheral function for controlling DPRAM	
Arguments	void	
Returned value	dpram_error_t	API normal end

4.2.7 R_DPRAM_GetStatus

R_DPRAM_GetStatus		
Overview	DPRAM get status function	
Declaration	dpram_error_t R_DPRAM_GetStatus (dpram_status_t * p_status)	
Description	• Returning arguments on DPRAM status	
Arguments	dpram_status_t * p_status	Stored pointer of status
Returned value	dpram_error_t	API normal end

4.2.8 R_DPRAM_SetStatus

R_DPRAM_SetStatus		
Overview	DPRAM status setting function	
Declaration	dpram_error_t R_DPRAM_SetStatus (dpram_status_t status)	
Description	• Setting DPRAM status by arguments	
Arguments	dpram_status_t status	Status
Returned value	dpram_error_t	API normal end

4.3 User setting files(r_shost_interface_cfg.h)

Creates a definition that can be changed by the user. In addition to host/slave switching, this library defines host /slave status, Cyclic IO, and NonCyclic IO addresses in shared memory, and these addresses can be changed. Please refer to 5.3 Data configuration for details.

Table 4-4 User definition table

Function	Definition	Value	Comments
Host/slave exchange	DPRAM_HOST	-	Definition on host side
address definition	DPRAM_CFG_ADDR_STATUS_HOST	0x500000	Address of Event Status for host
	DPRAM_CFG_ADDR_STATUS_SLAVE	0x500004	Address of Event Status for RZ/N2L
	DPRAM_CFG_ADDR_DATA_CYCLIC_IO	0x540000	Address of cyclic I/O data
	DPRAM_CFG_ADDR_DATA_NON_CYCLIC_IO	0x580000	Address of non-cyclic I/O data

5. Sample application

5.1 Overview

When Renesas Starter Kit+ for RZ/T2L as host connects with each PC by using Host interface library, this is a sample application which displays each status, acknowledges updating memory, and Read/Writes to shared memory indicating address, data exclusively on terminal software(This example uses TeraTerm).

5.2 Application configuration

Application configuration is shown in Figure 5-1

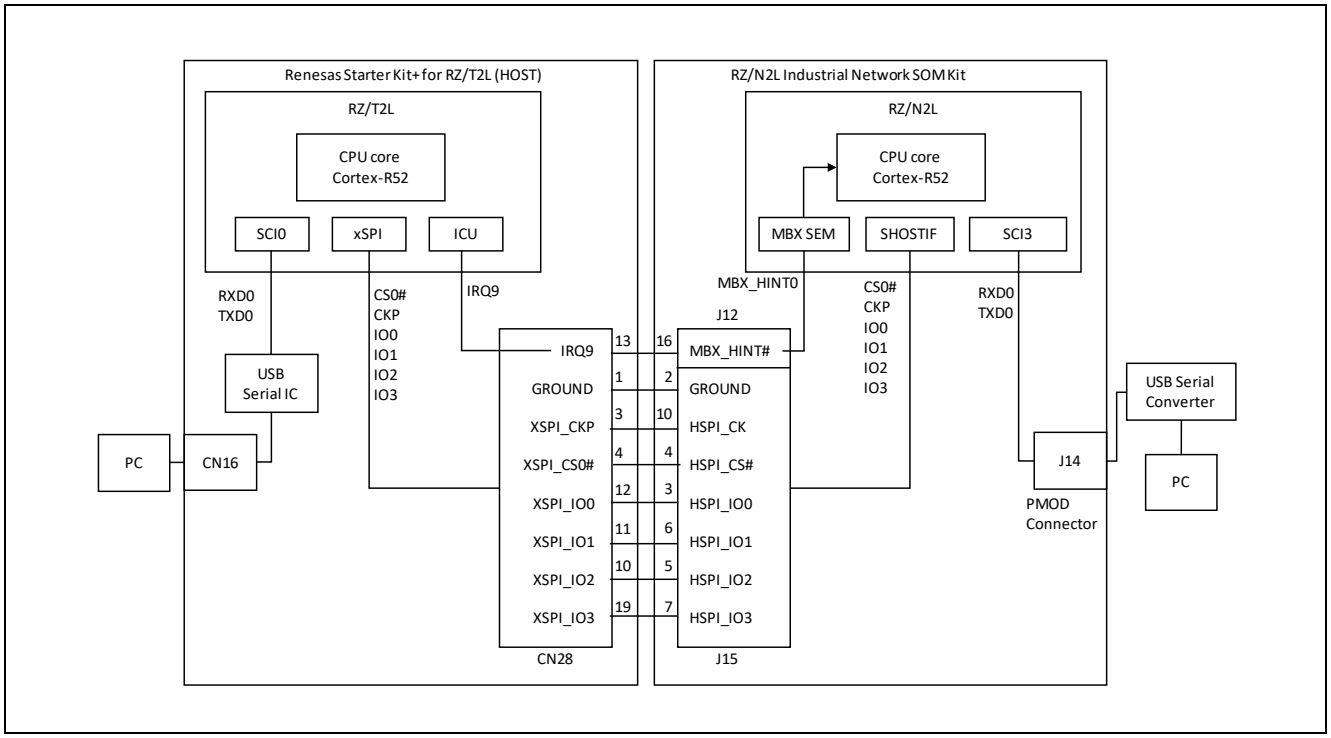

Figure 5-1 Sample application configuration

Serial communication setting on terminal software is shown on Table 5-1

Table 5-1 Serial setting

Item	Setting value
Baud rate	115200bps
Data	8bit
Parity	None
Stop bits	1bit
Flow control	None

5.3 Data configuration

Sample application defines data configuration like Table 5-2 and uses an application.

Table 5-2 Data configuration

Address from HOST (32-bit addressing mode)	Address from RZ/N2L	Description
05 0000	0x30140000	Event Status for host
05 0001-05 0003	0x30140004-0x3014001C	Reserved
05 0004	0x30140020	Event Status for RZ/N2L
05 0005-05 3FFF	0x30140024-0x3014FFFC	Reserved
05 4000-05 7FFF	0x30150000-0x3015FFFC	cyclic I/O data (64KB)
05 8000-05. FFFF	0x30160000-0x3017FFFC	non-cyclic I/O data (128KB)

5.4 Application overview

Event Status is displayed on PC terminal software after starting up. Main routine is executed when STATUS on both host and RZ/N2L is READY.
Main routine indicates Write command or Read command from host or slave. When Write command is indicated, shared memory is written, result is displayed on PC terminal software, and notifies memory updates. When Read command is indicated, shared memory is read, and PC terminal software is displayed. When a shared memory update notification is received, Displays on the terminal software that a shared memory update notification has been received.

5.4.1 Operation sequence

5.4.1.1 Startup sequence

Figure 5.2 shows the initialization sequence.

After the host completes initialization of the peripheral module to be used, it polls until DPRAM_STATUS_READY can be read from Event Status for RZ/N2L.If DPRAM_STATUS_READY can be read from Event Status for RZ/N2L, write DPRAM_STATUS_READY to Event Status for host.

After that, write DPRAM_STATUS_RUN to Event Status for host to complete initialization.RZ/N2L writes DPRAM_STATUS_READY to Event Status for RZ/N2L after completing the initialization of the peripheral module used.Then, poll until DPRAM_STATUS_READY can be read from Event Status for host. If DPRAM_STATUS_READY can be read from Event Status for host, write DPRAM_STATUS_RUN to Event Status for RZ/N2L to complete initialization.

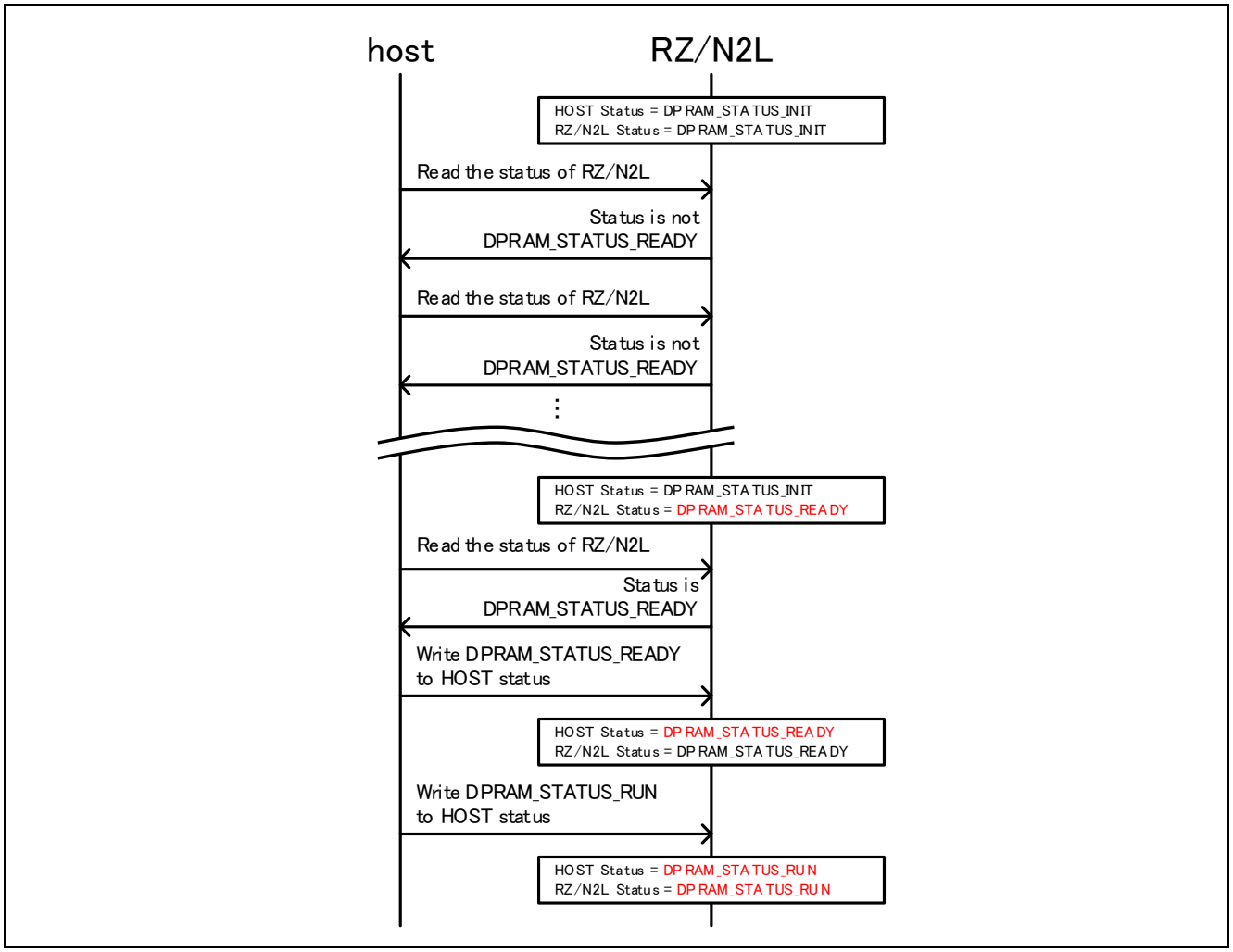

Figure 5-2 initialization sequence

5.4.1.2 DPRAM access sequence

Figure 5-3 shows the DPRAM access sequence.

When both the host and RZ/N2L access DPRAM, exclusive control is performed using the MBXSEM semaphore register of RZ/N2L.

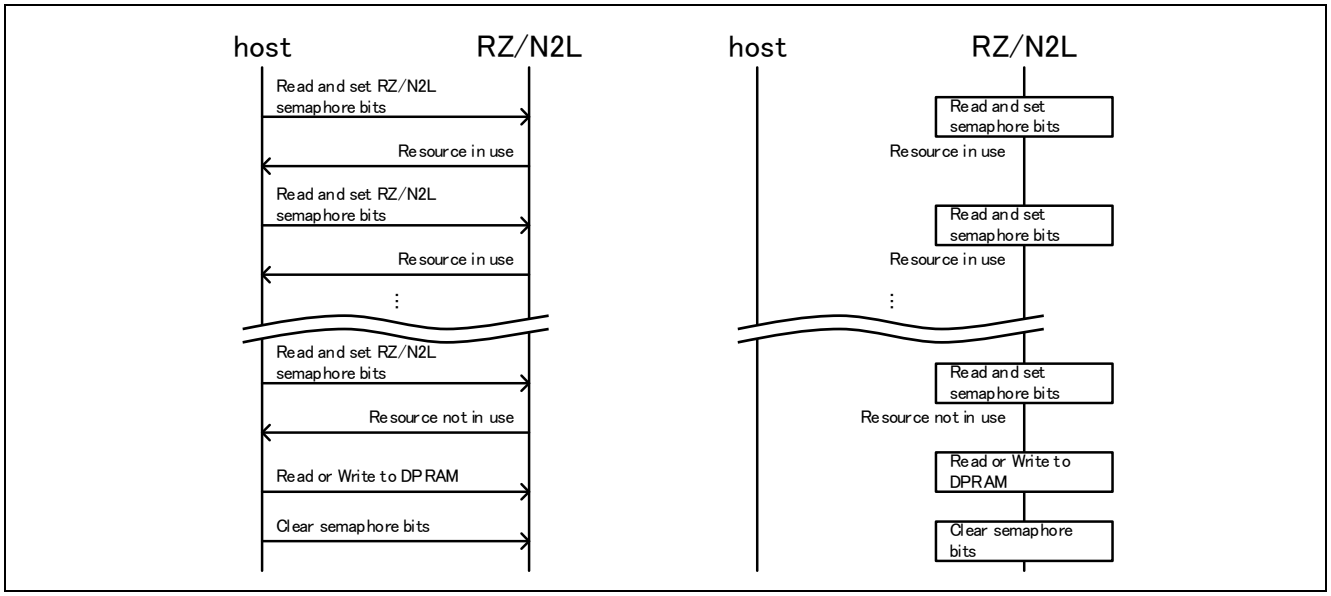

Figure 5-3 DPRAM access sequence

5.4.1.3 DPRAM Update notification sequence

Figure 5-4 shows the DPRAM update notification sequence.

When updating DPRAM from the host or RZ/N2L, the update is notified to the other party using the RZ/N2L's MBXSEM interrupt function.

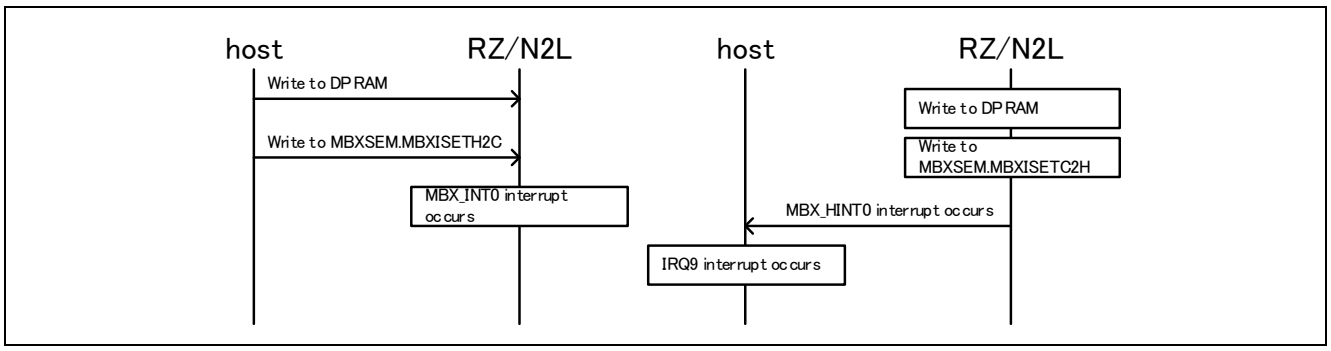

Figure 5-4 DPRAM Update notification sequence

5.4.2 Flowchart

5.4.2.1 Starting up flowchart

Starting up flowchart is shown in Figure 5-5. First, initialize the UART for communication with the terminal software and display the initial information on the terminal software. Next, execute the DPRAM_Open function, display the status if there is an update, and repeat the execution until the return value is DPRAM_SUCCESS. The DPRAM_Open function returns DPRAM_SUCCESS when both the host and RZ/N2L become operational.

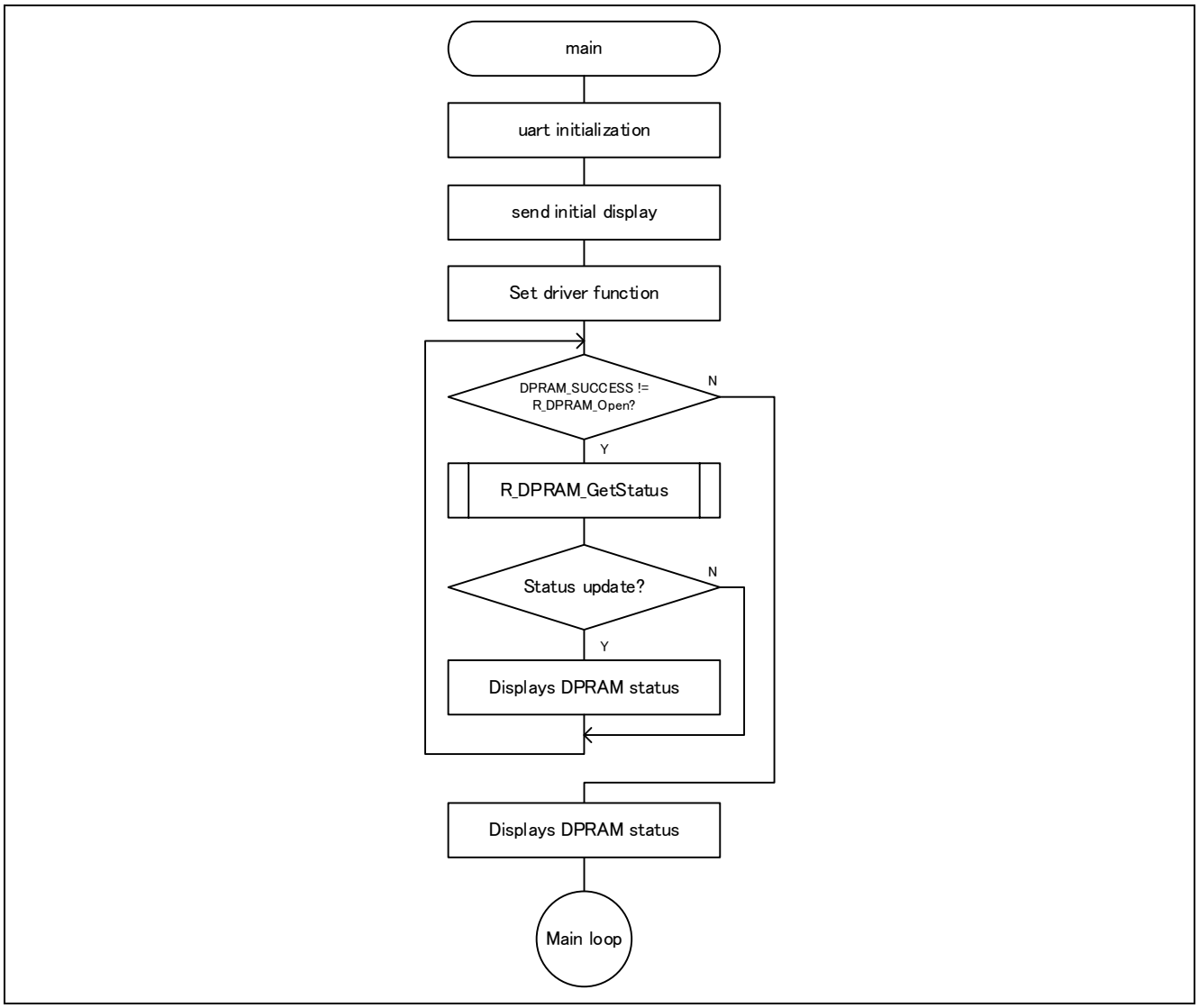

Figure 5-5 Startup flow

5.4.2.2 Main flowchart

Main flowchart is shown in Figure 5-6. Explanation is displayed on terminal software, and waiting until command is entered from keyboard. When the command input is completed, after judging command, executing the corresponding treatment, and result is shown on terminal software. Finally, check the notification from the other party and display the DPRAM update on terminal software.

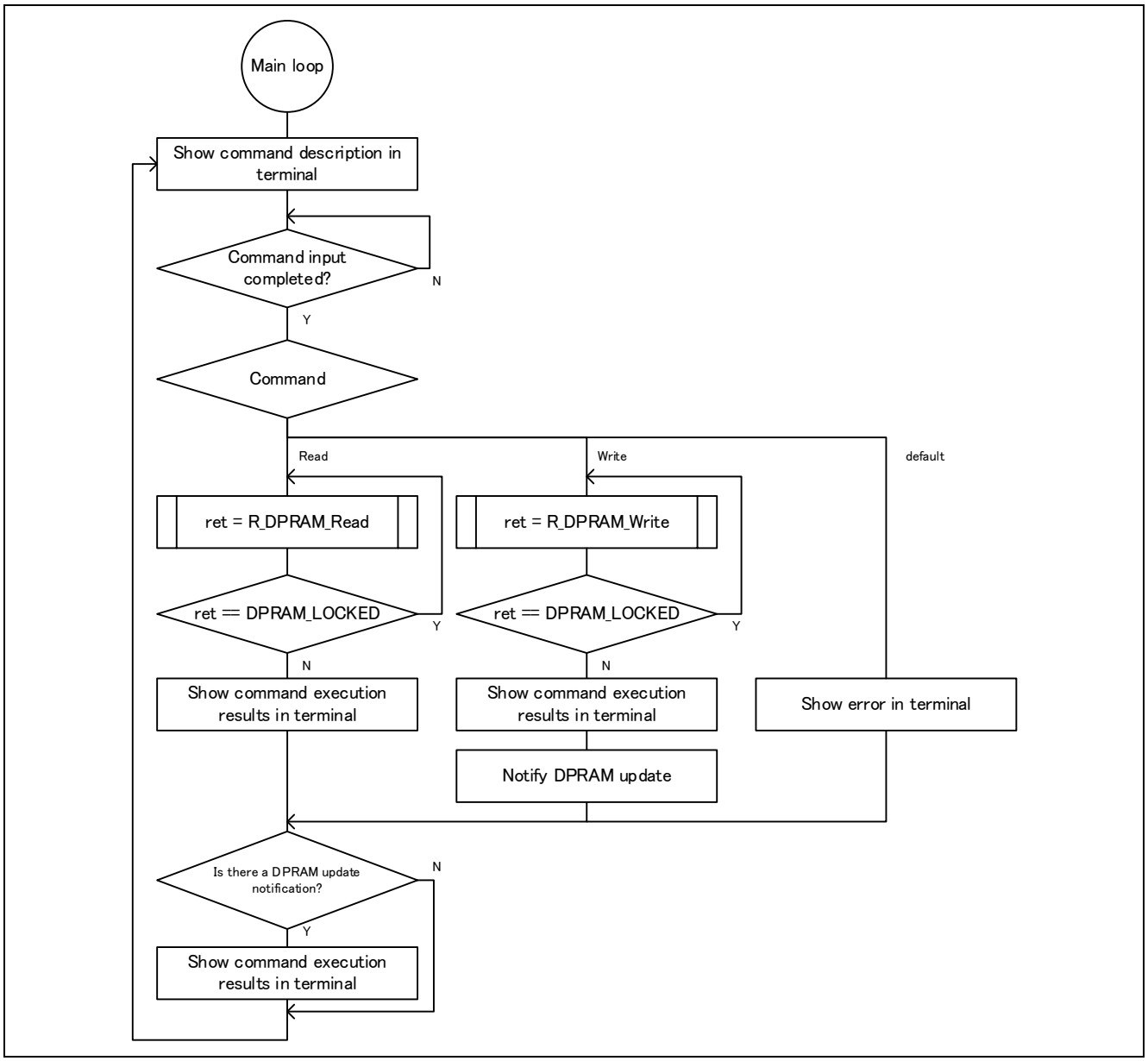

Figure 5-6 main flow

5.5 Terminal software display and operation method

After starting up, each program information is displayed and status information on DPRAM library is shown on terminal software. After finishing initialization on both host and slave DPRAM, [application start] is displayed, main routine is executed. Display of starting up of HOST is shown in Figure 5-7. Display of starting up of RZN2L is shown in Figure 5-8.

```
[INFO]----------------------------------------------------
[INFO]host interface library application program
[INFO]terminal window on HOST
[INFO]----------------------------------------------------
```

Figure 5-7 HOST Startup screen

```
[INFO]----------------------------------------------------
[INFO]host interface library application program
[INFO]terminal window on RZN2L
[INFO]----------------------------------------------------
```

Figure 5-8 RZN2L Startup screen

After the startup screen, the initialization status of DPRAM is displayed. Figure 5-9. shows DPRAM initialization.

```
[INFO]DPRAM initialization
[INFO]DPRAM ready
[INFO]DPRAM application start
```

Figure 5-9 Command input waiting screen

Figure 5-10 shows the command input waiting screen. Wait until [r] or [w]+Enter is entered from the keyboard for e2studio, or [r] or [w] for EWARM. This input section is case sensitive.

```
INFO]----------------------------------------------------
[INFO]Please enter [r] or [w] and ENTER
[INFO][r] is read command
[INFO][w] is write command
```

Figure 5-10 Command input waiting screen

When the command is entered, the screen will move to the address input waiting screen. Figure 5-11 shows the address input waiting screen. For e2sutudio, enter the 32-bit hexadecimal data + Enter as an address from the keyboard. For EWARM, wait until the 32-bit hexadecimal data is input. This input location is not case sensitive. Please do not input more than 32bit.

```
[INFO]Enter the 32-bit address in hexadecimal and press ENTER.
```

Figure 5-11 Address input waiting screen

In the case of the [w] command, after entering the address, the screen will transition to the write data input waiting screen. Figure 5-12 shows the data input waiting screen. For e2sutudio, 32-bit hexadecimal data + Enter is input as the keyboard or data, and for EWARM, wait until 32-bit hexadecimal data is input. This input location is not case sensitive. Please do not input more than 32bit.

Figure 5-12 data input waiting screen

After the command input is completed, the corresponding processing will be executed. When accessing DPRAM, the library performs semaphore control, and if it is locked, it will display [DPRAM_LOCKED] and retry. If it is not locked, the execution result will be displayed on the terminal software. Figure 5-13 shows the execution result display screen.

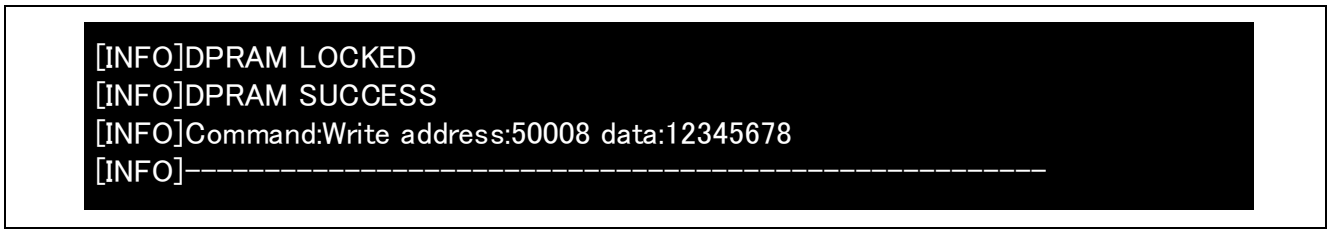

Figure 5-13 Results display screen

After executing the Write command, notify the other party of the DPRAM update. If the host updates the DPRAM, it will be notified on the RZ/N2L terminal software screen as shown in Figure 5-14. When RZ/N2L updates DPRAM, a notification will be displayed on the host terminal software screen as shown in Figure 5-15.

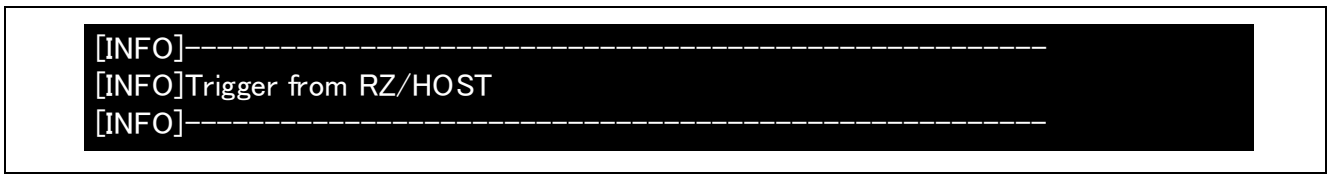

Figure 5-14 RZ/N2L Command notification screen

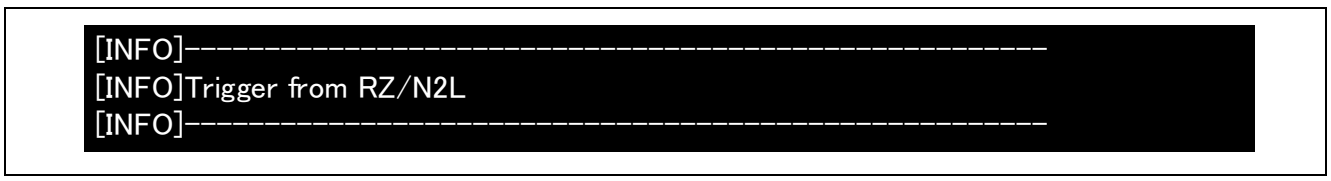

Figure 5-15 HOST Command notification screen

6. Revision History

Rev.	Date	Description	
		Page	Summary
1.0	2023.12.22	-	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance 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 the 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.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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. 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.