

# E8a Emulator

Additional Document for User's Manual  
R0E00008AKCE00EP16

Renesas Microcomputer Development Environment System  
R8C Family / R8C/2x Series  
Notes on Connecting the R8C/24 and R8C/25

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corporation without notice. Please review the latest information published by Renesas Electronics Corporation through various means, including the Renesas Electronics Corporation website (<http://www.renesas.com>).

## 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.

---

# Contents

---

|           |  |    |
|-----------|--|----|
| Section 1 | Inside the E8a Emulator User's Manual.....           | 1  |
| Section 2 | E8a Emulator Specifications .....                    | 2  |
| Section 3 | Connecting the E8a Emulator to the User System ..... | 3  |
| Section 4 | E8a Connecting Connector Pin Assignments.....        | 4  |
| Section 5 | Examples of E8a Connections.....                     | 5  |
| Section 6 | Notes on Using the E8a Emulator.....                 | 8  |
| Section 7 | Debugger Setting .....                               | 15 |
| Section 8 | Applicable Tool Chain and Third-party Products ..... | 19 |

This user's manual is applicable to the E8a emulator software V.1.03 Release 00 or later.

## Section 1 Inside the E8a Emulator User's Manual

The E8a manual consists of two documents: the E8a User's Manual and the E8a Additional Document for User's Manual (this document). Be sure to read BOTH documents before using the E8a emulator.

(1) E8a Emulator User's Manual

The E8a Emulator User's Manual describes the hardware specifications and how to use the emulator debugger.

- E8a emulator hardware specifications
- Connecting the E8a emulator to the host computer or user system
- Operating the E8a emulator debugger
- Tutorial: From starting up the E8a emulator debugger to debugging

(2) E8a Additional Document for User's Manual

The E8a Additional Document for User's Manual describes content dependent on the MCUs and precautionary notes.

- MCU resources used by the E8a emulator
- Example of the E8a emulator connection or interface circuit necessary for designing the hardware
- Notes on using the E8a emulator
- Setting the E8a emulator debugger during startup

## Section 2 E8a Emulator Specifications

Table 2.1 shows the E8a emulator specifications for the R8C/24 and R8C/25 Groups.

Table 2.1 E8a Emulator Specifications for the R8C/24 and R8C/25 Groups

|                                   |  |
|-----------------------------------|--|
| Target MCUs                       | R8C Family R8C/2x Series<br>R8C/24 and R8C/25 Groups   |
| Available operating modes         | Single-chip mode   |
| Break functions                   | - Address match break, 4 points, or<br>Address match break, 2 points + Data condition break, 1 point<br>- PC break points (maximum 255 points)<br>- Forced break |
| Trace functions                   | Last 4 branch instructions   |
| Flash memory programming function | Available  |
| User interface                    | 1-line clock asynchronous serial interface (communication via MODE pin)<br>* UART1 function cannot be used in a user program                                     |
| MCU resources to be used          | - Flash memory: 2 KB [*1]<br>- Stack 8 bytes<br>- Address match interrupt<br>- UART1 function  |
| Emulator power supply             | Unnecessary (USB bus powered, power supplied from the PC)  |
| Interface with host machine       | USB (USB 1.1, full speed)<br>* Also connectable to host computers that support USB 2.0   |
| Power supply function             | Can supply 3.3 V or 5.0 V to the user system (maximum 300 mA)  |
| Power voltages                    | 3.0 - 5.5 V (f(XIN)=20MHz)<br>2.7 - 5.5 V (f(XIN)=10MHz)   |

Note:

1. Varies depending on the MCU. For details, see "Section 6 Notes on Using the E8a Emulator".

Table 2.2 shows the operating environment of the E8a emulator.

Table 2.2 Operating Environment

|               |                    |                                     |
|---------------|--------------------|-------------------------------------|
| Temperatures  | Active             | : 10°C to 35°C                      |
|               | Inactive           | : -10°C to 50°C                     |
| Humidity      | Active             | : 35% RH to 80% RH, no condensation |
|               | Inactive           | : 35% RH to 80% RH, no condensation |
| Vibrations    | Active             | : maximum 2.45 m/s <sup>2</sup>     |
|               | Inactive           | : maximum 4.9 m/s <sup>2</sup>      |
|               | Transportation     | : maximum 14.7 m/s <sup>2</sup>     |
| Ambient gases | No corrosive gases |                                     |

## Section 3 Connecting the E8a Emulator to the User System

Before connecting the E8a emulator to the user system, a connector must be installed in the user system so a user system interface cable can be connected. When designing the user system, refer to Figure 4.1 “E8a Connecting Connector Pin Assignments” and Figure 5.1 “Example of an E8a Connection”.

Before designing the user system, be sure to read the E8a Emulator User’s Manual and related device hardware manuals.

Table 3.1 shows the recommended connector for the E8a emulator.

Table 3.1 Recommended Connector

|                  | Type Number | Manufacturer | Specification        |
|------------------|-------------|--------------|----------------------|
| 14-pin connector | 2514-6002   | 3M Limited   | 14-pin straight type |

Connect E8a connecting connector pins 2, 4, 6, 10, 12 and 14 firmly to the GND on the user system board. These pins are used as an electric GND and monitor the connection of the user system connector. Note the pin assignments for the user system connector.

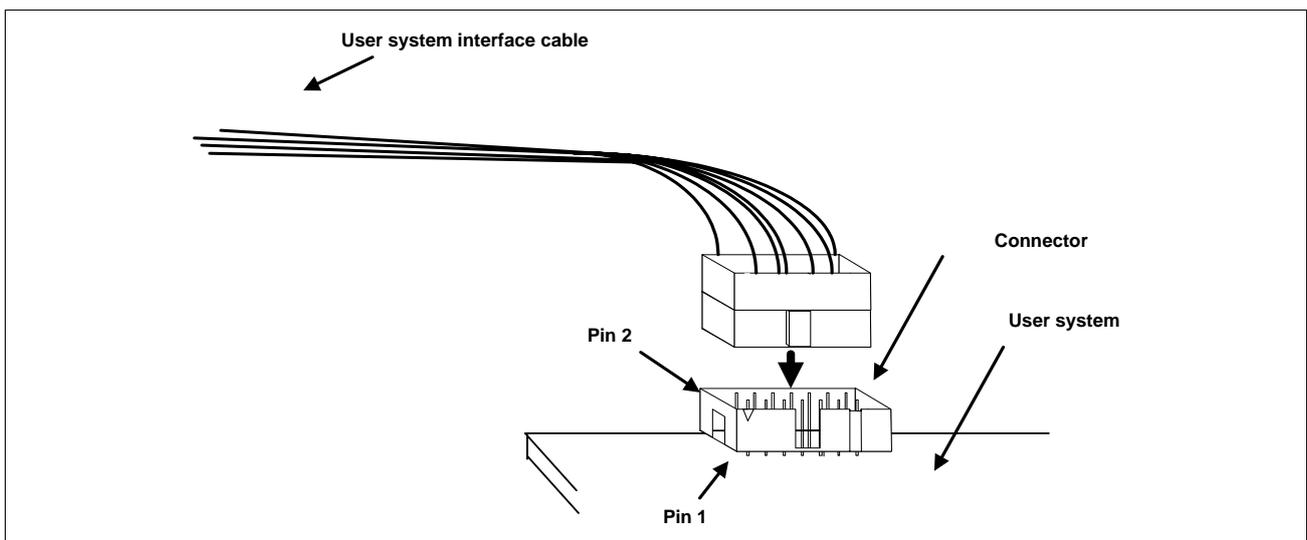


Figure 3.1 Connecting the User System Interface Cable with an E8a Connecting Connector

Notes:

1. Do not place any components within 3 mm area of the connector.
2. When using the E8a emulator as a programmer, connect it to the user system in the same way.

## Section 4 E8a Connecting Connector Pin Assignments

Figure 4.1 shows the pin assignments for the E8a connecting connector.

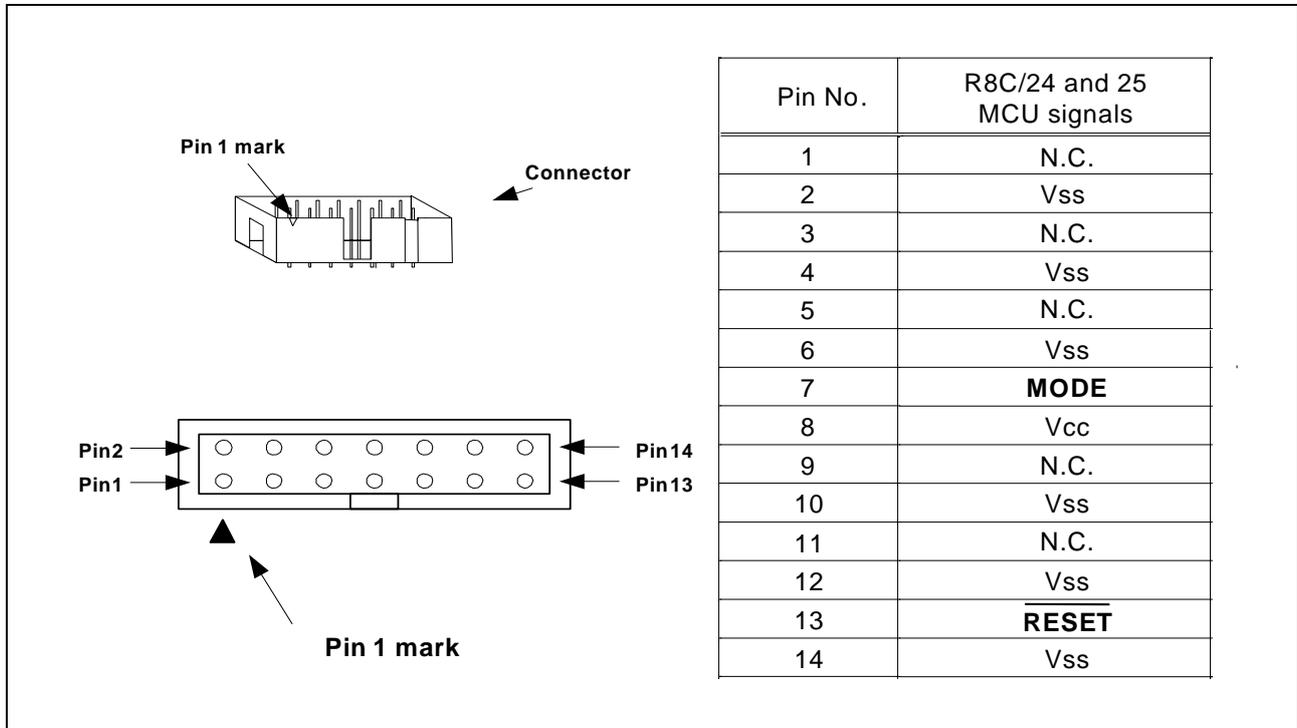


Figure 4.1 E8a Connecting Connector Pin Assignments

Note:

Pin 14 is used for checking the connection between the E8a and the user system, and pins 4, 6 and 10 are connected to the internal circuit. These pins are not directly connected to the Vss inside the E8a. Make sure pins 2, 4, 6, 10, 12 and 14 are all connected to the Vss.

## Section 5 Examples of E8a Connections

Figure 5.1 shows a connection example.

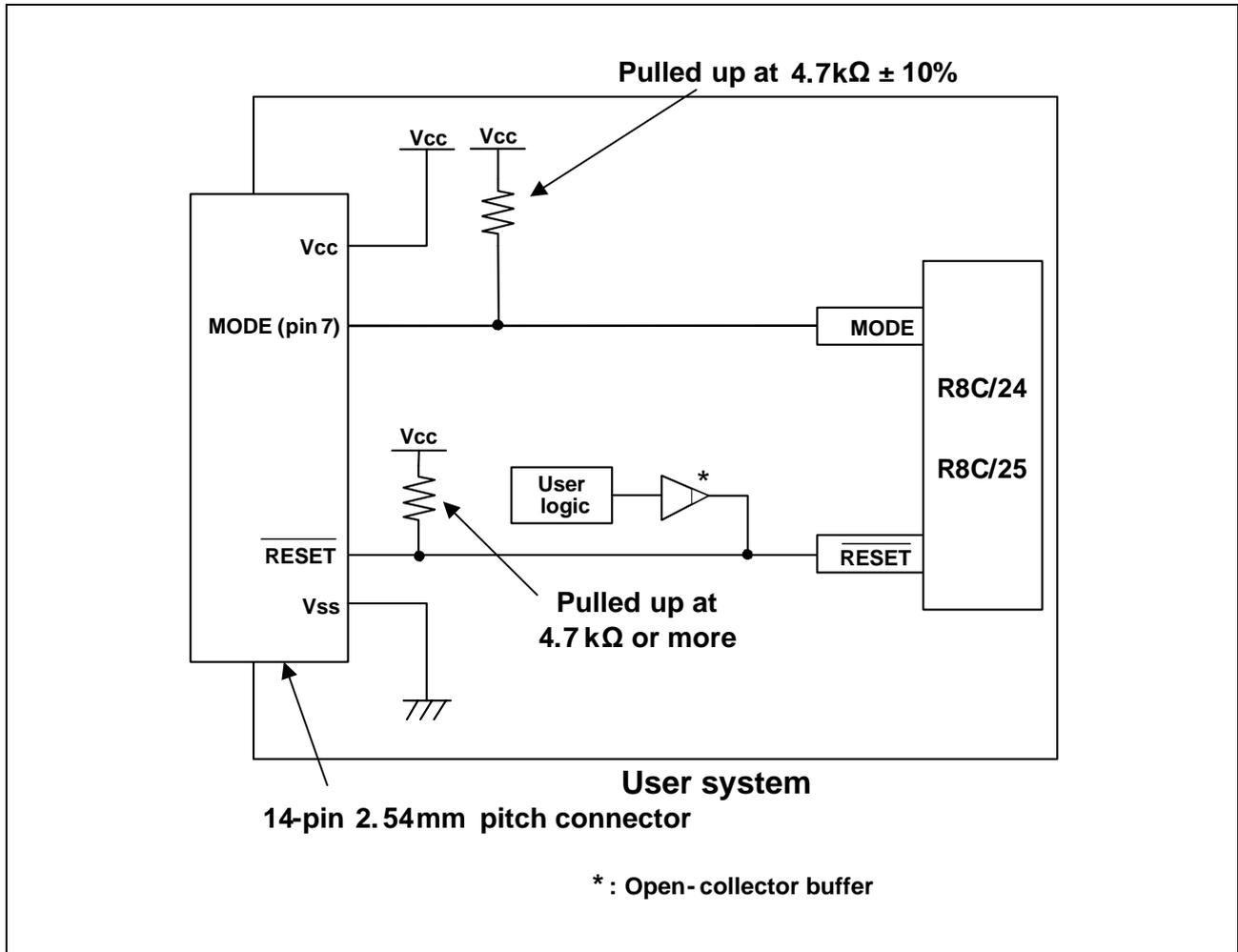


Figure 5.1 Example of an E8a Connection

When using the emulator as a programmer, the connection specification between the E8a and the MCUs is the same as shown in Figure 5.1.

## Points to Remember

1. The E8a emulator uses the MODE pin for MCU control and forced break control. Pull up the E8a emulator and MCU pins and connect the E8a emulator.

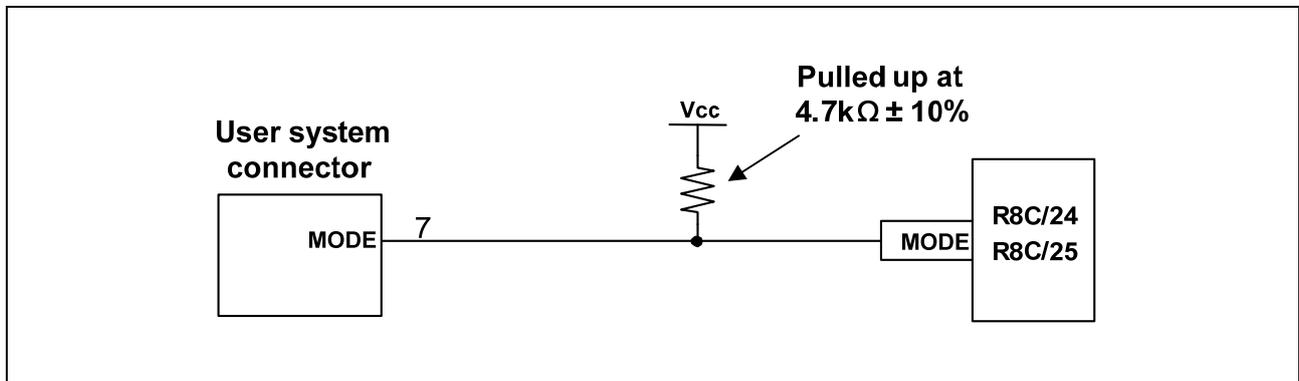


Figure 5.2 E8a Emulator and MODE Pin Connection

2. The  $\overline{\text{RESET}}$  pin is used by the E8a emulator. Therefore, use an open-collector output buffer or a CR reset circuit as the reset circuit for the user system. The recommended pull-up value is 4.7 k $\Omega$  or more. The MCU can be reset by outputting “L” from the E8a emulator. However, if the reset IC output is “H”, the user system reset circuit cannot be set to “L”. As such, the E8a emulator will not operate normally.

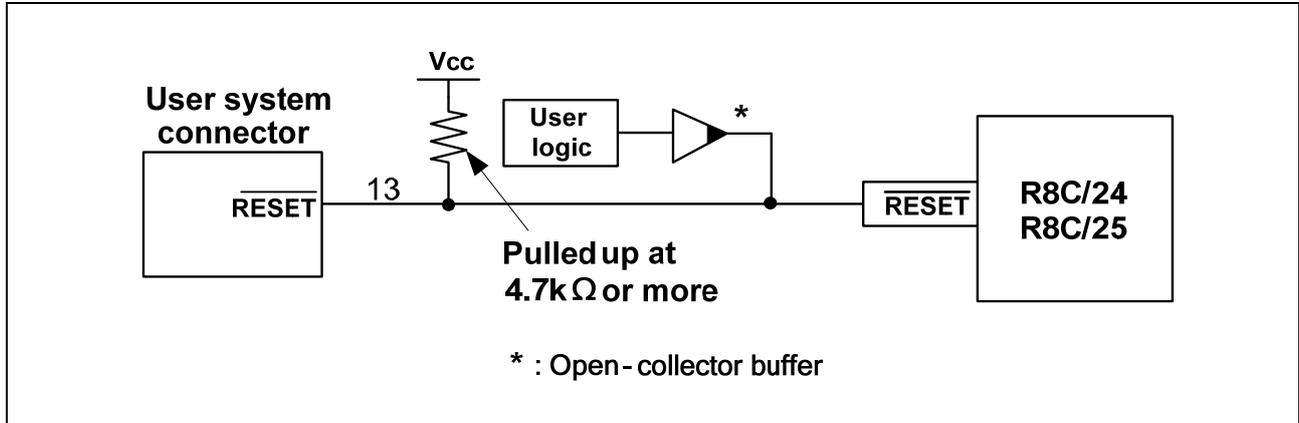


Figure 5.3 Example of a Reset Circuit

3. Connect Vss and Vcc to the Vss and Vcc of the MCU, respectively.
4. Do not connect anything to the N.C. pin.
5. The amount of voltage input to Vcc must be within the specified range of the MCU.
6. Pin 14 is used for checking the connection between the E8a and the user system, and pins 4, 6 and 10 are connected to the internal circuit. These pins are not directly connected to the Vss inside the E8a. Make sure pins 2, 4, 6, 10, 12 and 14 are all connected to the Vss.

7. Figure 5.4 shows the interface circuit in the E8a emulator. Use this figure as a reference when determining the pull-up resistance value.

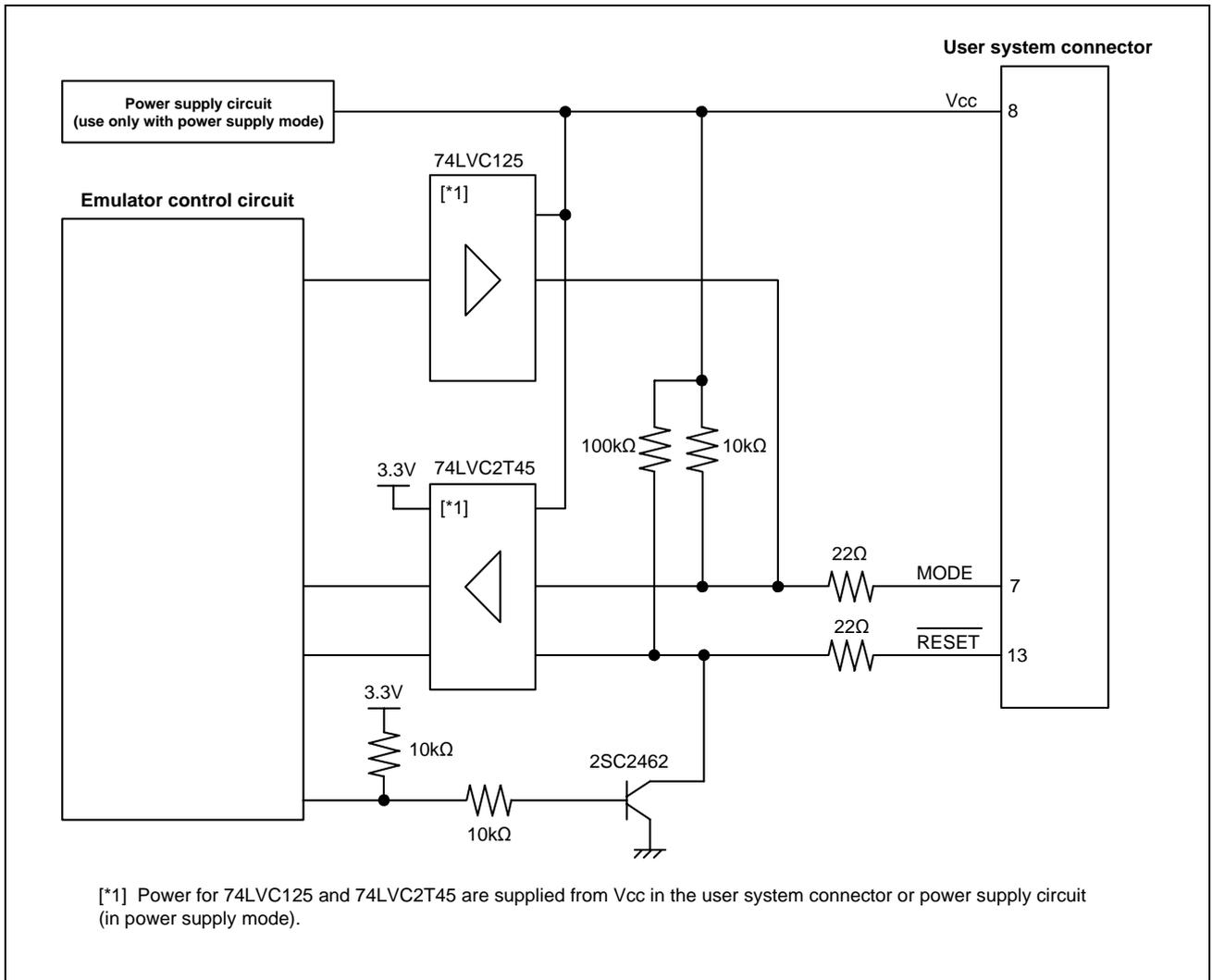


Figure 5.4 Interface Circuit Inside the E8a Emulator (For Reference)

## Section 6 Notes on Using the E8a Emulator

### 1. Program area for the E8a emulator

Table 6.1 lists the program areas allotted for the E8a emulator. Do not change this area allocation, otherwise the E8a emulator will not control the MCU. If settings were changed, disconnect the debugger and then reconnect it.

Table 6.1 Program Area for the E8a Emulator

| Group  | Part No. | ROM Size         |           | Program Area for E8a Emulator   |  |   |
|--------|----------|------------------|-----------|---|--|---|
|        |          | Programming Area | Data Area | Vector Area   | ROM Area (Default Area)  |   |
| R8C/24 | R5F21244 | 16 KB            | -         | FFE4h - FFE7h,<br>FFE8h - FFEbH,<br>FFECh - FFEFh,<br>FFF4h - FFF7h,<br>FFF8h - FFFBh,<br>FFFCh - FFFEh | -  |   |
|        | R5F21245 | 24 KB            | -         |   | -  |   |
|        | R5F21246 | 32 KB            | -         |   | 2 KB of the ROM area [*1]<br>(User Flash Area 8000h - 87FFh)   |   |
|        | R5F21247 | 48 KB            | -         |   | -  |   |
|        | R5F21248 | 64 KB            | -         |   | 2 KB of the ROM area [*1]<br>(User Flash Area 13800h - 13FFFh) |   |
| R8C/25 | R5F21254 | 16 KB            | 2 KB      |   |  | -   |
|        | R5F21255 | 24 KB            | 2 KB      |   |  | -   |
|        | R5F21256 | 32 KB            | 2 KB      |   |  | 2 KB of the ROM area [*1]<br>(Data Flash Area 2400h - 2BFFh)<br>or<br>(User Flash Area 8000h - 87FFh)   |
|        | R5F21257 | 48 KB            | 2 KB      |   |  | -   |
|        | R5F21258 | 64 KB            | 2 KB      |   |  | 2 KB of the ROM area [*1]<br>(Data Flash Area 2400h - 2BFFh)<br>or<br>(User Flash Area 13800h - 13FFFh) |

#### Note:

- When using the MCU types R5F21246, R5F21248, R5F21256 or R5F21258, the [Emulator Setting] dialog box shown in Figure 6.1 is displayed. Specify the area which will not be used by the user system. If the [Enable advanced setting] check box is checked, the area used by the emulator can be specified with the address (Figure 6.2).

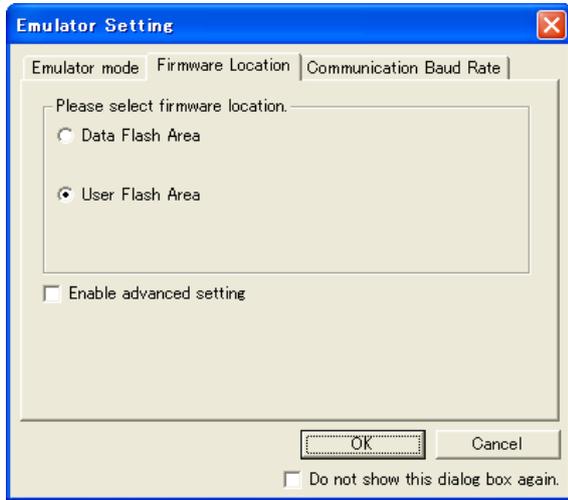


Figure 6.1 [Firmware Location] tab of [Emulator Setting] Dialog Box (1)

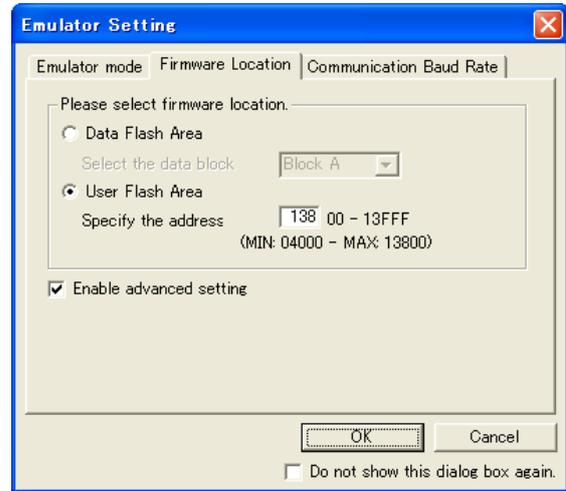


Figure 6.2 [Firmware Location] tab of [Emulator Setting] Dialog Box (2)

- When the system is launched, the E8a emulator initializes the general registers and some of the flag registers as shown in Table 6.2.

Table 6.2 E8a Emulator Register Initial Values

| Status                  | Register             | Initial Value                                  |
|-------------------------|----------------------|--|
| E8a Emulator Activation | PC                   | Reset vector value in the vector address table |
|                         | R0 to R3 (bank 0, 1) | 0000h  |
|                         | A0, A1 (bank 0, 1)   | 0000h  |
|                         | FB (bank 0, 1)       | 0000h  |
|                         | INTB                 | 0000h  |
|                         | USP                  | 0000h  |
|                         | ISP                  | 05FFh  |
|                         | SB                   | 0000h  |
|                         | FLG                  | 0000h  |

- The E8a emulator communicates with the MCU by using the MODE and  $\overline{\text{RESET}}$  pins.
- The E8a emulator uses up to 8 bytes of the stack pointer during a user program break. Therefore, set aside 8 bytes for the stack area.
- Peripheral function used by the E8a emulator program  
 UART1 is used by the E8a emulator. Do not use UART1 in the user program.

## 6. SFRs used by the E8a emulator program

The SFRs listed in Table 6.3 are used by the E8a emulator program as well as the user program. Do not change the value in the memory window, etc., by other than the user program. Note that although the SFRs can be changed during user program execution, the changed value cannot be read at the break.

The SFRs listed in Table 6.4 are used by the E8a emulator program, not the user program. Do not change the registers, otherwise the E8a cannot control the MCU. Note that UART1 transmit interrupt control register S1TIC and UART1 receive interrupt control register S1RIC always read out values used by the emulator.

The SFRs listed in Tables 6.3 and 6.4 are not initialized by selecting [Debug] -> [Reset CPU] or by using the RESET command. If register contents are referred to, a value that has been set in the E8a emulator program will be read out.

Table 6.3 SFRs Used by the E8a Emulator Program (1)

| Address | Register   | Symbol | Bit      |
|---------|--|--------|----------|
| 000Ah   | Protect register                                 | PRCR   | Bit 0    |
| 0023h   | High-speed on-chip oscillator control register 0 | FRA0   | Bit 0    |
| 0024h   | High-speed on-chip oscillator control register 1 | FRA1   | All bits |
| 01B3h   | Flash memory control register 4                  | FMR4   | Bit 7    |

Table 6.4 SFRs Used by the E8a Emulator Program (2)

| Address       | Register                                  | Symbol | Bit      | Notes on Using the E8a Emulator |
|---------------|---|--------|----------|---------------------------------|
| 0010h - 0012h | Address match interrupt register 0        | RMAD0  | All bits | [*1]                            |
| 0013h         | Address match interrupt enable register   | AIER   | All bits | [*1]                            |
| 0014h - 0016h | Address match interrupt register 1        | RMAD1  | All bits | [*1]                            |
| 00A8h         | UART1 transmit/receive mode register      | U1MR   | All bits | [*1]                            |
| 00A9h         | UART1 bit rate register                   | U1BRG  | All bits | [*1]                            |
| 00AAh, 00ABh  | UART1 transmit buffer register            | U1TB   | All bits | [*1]                            |
| 00ACh         | UART1 transmit/receive control register 0 | U1C0   | All bits | [*1]                            |
| 00ADh         | UART1 transmit/receive control register 1 | U1C1   | All bits | [*1]                            |
| 00AEh, 00AFh  | UART1 receive buffer register             | U1RB   | All bits | [*1]                            |
| 00F8h         | Port mode register                        | PMR    | Bit 4    | [*2]                            |

Notes:

1. Do not change this register value.
2. Do not change the value of the bits listed in the column to the left. When operating this register, make changes using the bit operation instructions to avoid changing the bit values.

## 7. Interrupts used by the E8a emulator program

The BRK instruction interrupt, address match interrupt, single-step interrupt and address break interrupt are used by the E8a emulator program. Therefore, make sure the user program does not use any of these interrupts. The E8a emulator changes these interrupt vector values to the values to be used by the emulator. No problems occur if the interrupt vector values are written in the user program.

## 8. Debugging during a watchdog timer operation

When running the E8a emulator program, the program refreshes the watchdog timer. If memory access is executed through memory reference or modification, the watchdog timer will be refreshed by the E8a emulator program. Note that this timing will differ from the actual operational timing.

## 9. Flash memory ID code

This MCU function prevents the Flash memory from being read out by anyone other than the user. The ID code in Table 6.5 written to the flash memory of the MCU must match the ID code displayed in the Figure 6.3 [ID Code verification] dialog box at debugger startup, otherwise the debugger cannot be launched. Note that when the ID code is FFh, FFh, FFh, FFh, FFh, FFh, FFh, the ID code is regarded as undefined. In this case, the ID code is automatically authenticated and the [ID Code verification] dialog box is not displayed.

In 'Program Flash' mode, the contents of the user program are input into the ID code area. When debugging in other modes, FFh, FFh, FFh, FFh, FFh, FFh, FFh is written into the ID code area regardless of the contents of the downloaded user program.

Table 6.5 ID Code Storage Area of R8C/24 and 25

| Address | Description             |
|---------|-------------------------|
| FFDFh   | First byte of ID code   |
| FFE3h   | Second byte of ID code  |
| FFEBh   | Third byte of ID code   |
| FFEFh   | Fourth byte of ID code  |
| FFF3h   | Fifth byte of ID code   |
| FFF7h   | Sixth byte of ID code   |
| FFFBh   | Seventh byte of ID code |

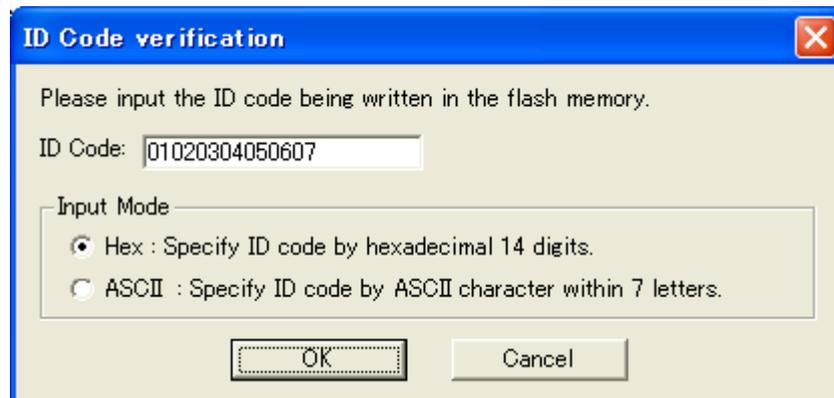


Figure 6.3 [ID Code verification] Dialog Box

Note on 'Program Flash' mode:

When the ID code is specified by the -ID option of the lmc30, download the MOT file or HEX file. When the X30 file is downloaded, the ID code is not valid. When downloading the X30 file, specify the ID code using an assembler directive command such as ".BYTE". The file to which the ID code specified by the assembler directive command ".ID" is output varies depending on the version of the assembler. For details, refer to the Assembler User's Manual.

10. Operation clock during a user program halt

When the user program halts, the emulator changes the CPU clock to the internal high-speed on-chip oscillator (approx. 8MHz) to operate. However, the peripheral features operate with the clock specified by the user program.

11. Reset

The reset vector is used by the E8a emulator program. If the MCU is reset while executing the user program, control is transferred to the E8a emulator program and the user program is forced to stop. Do not use the voltage monitor 0 reset, voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset and software reset, otherwise the E8a emulator will not operate normally.

12. Memory access during emulation execution

When referring to or modifying the memory contents, the user program is temporarily halted. For this reason, a real-time emulation cannot be performed. When a real-time emulation is necessary during a program execution, disable the automatic update in the watch window or fix the display in the memory window before running the program so that memory accesses do not occur during an execution.

13. When the E8a emulator does not supply power to the user system, it consumes the power voltage of the user system from several mA to more than 10 mA. This is because the user power supply drives 74LVC125, 74LVC1T45 and 74LVC2T45 to make the communication signal level match the user system power supply voltage.

14. When debugging, the Flash memory is frequently rewritten by the E8a emulator. Therefore, do not use an MCU that has been used for debugging in products. Also, as the E8a emulator program is written to the MCU while debugging, do not save the contents of the MCU Flash memory which were used for debugging nor use them as the ROM data for products.

15. Reserved area

The addresses not specified in the Hardware Manual for R8C/24 and R8C/25 Groups are reserved area. Do not change the contents. Otherwise, the E8a emulator cannot control the MCU.

16. Debugging in stop mode or wait mode

When debugging in stop mode or wait mode, do not operate windows until the program stops at the breakpoint by setting the breakpoint at the line of the program which will be executed after the stop mode or wait mode is cancelled. In addition, disable the automatic update in the watch window or fix the display in the memory window before running the program so memory accesses do not occur during an execution. When the program is forcibly stopped, or the memory is referred to or modified in stop or wait mode, the stop or wait mode will be cancelled.

17. Peripheral I/Os during a halt

During a user program halt, interrupts are not accepted although peripheral I/Os continue to run. For example, a timer interrupt is not accepted although the timer continues to count when a user program is stopped by a break after the timer started.

## 18. Exceptional step execution

## a) Software interrupt instruction

Step execution cannot be performed in the internal processing of instructions (undefined, overflow, BRK and INT) which generate a software interrupt continuously in the program.

Example: INT instruction

```

NOP
NOP
INT#3
NOP
JMP MAIN
INT_3:
NOP
NOP
NOP
REIT

```

↳ Passes through if the STEP execution is carried out.

← Program should be stopped at this address.

## b) INT instruction

To debug the user program with the INT instruction, set a PC break for the internal processing of the INT instruction and execute the program with the GO command.

Example:

```

NOP
INT #3
NOP
JMP MAIN
INT_3:
NOP Break
NOP
REIT

```

Execute using GO command.

## 19. "Go to cursor" function

The "Go to cursor" function is actualized using an address match break. Therefore, when you execute the "Go to cursor" command, all the address match breaks and hardware breaks you set become invalid, while all the PC breaks remain valid.

## 20. Note on PC break point

When downloading a user program after modifying it, the set address of PC break may not be corrected normally depending on the modification. Therefore, break points other than the set PC breaks may shift. After downloading a user program, check the setting of PC breaks in the event point window and reset it.

## 21. Note on setting the break condition of the hardware breaks

When selecting the normal address bus (selecting "Address" radio button in the [Break condition] dialog box) as a condition for the address bus, do not set the following addresses. Otherwise, an unauthorized break may occur.

- Address in the interrupt vector table
- Address set in the interrupt vector table (interrupt routine start address)
- Branch address of the branch instruction

Both fixed vector table and variable vector table are included with the interrupt vector table above.

#### 22. Note on debugging in CPU rewrite mode

CPU rewrite can be executed only for the data area. If it is executed for the program area, the E8a emulator cannot control the MCU. When rewriting the data area, do not halt the user program, while setting up the CPU rewrite mode and releasing it. If halted, the E8a emulator may not control the MCU. In addition, disable the automatic update in the watch window or fix the display in the memory window before running the program so memory accesses do not occur during an execution.

To check the data after executing the CPU rewrite mode, halt the program after releasing the CPU rewrite mode and refer to the memory window, etc.

If CPU rewrite can be executed for the data area, and erase/program process can be suspended, do not use software breaks.

#### 23. Note on rewriting Flash memory

Do not reset the MCU when rewriting the Flash memory.

The Flash memory is rewritten when the “Flash memory write end” is displayed in the output window of the High-performance Embedded Workshop. If the MCU is reset when rewriting the Flash memory, the user program or the E8a emulator program may be disrupted.

Flash memory rewrite occurs:

- When downloading the user program
- After setting PC breaks in the Flash memory and executing the user program
- After canceling PC breaks in the Flash memory and executing the user program
- After rewriting the value of the Flash memory in the memory window and executing the user program

#### 24. Note on the E8a emulator power supply

When writing a program with the E8a emulator for mass production processes, the program requires reliability, so do not use the E8a emulator power supply function. Supply power separately to the user system according to the allowable voltage for MCU writing. Voltage supplied from the E8a emulator depends on the quality of the USB power supply of the PC, and as such, precision is not guaranteed.

#### 25. Note on internal power low consumption

Make sure that bit 0 of voltage detect register 2 (VCA2) for the E8a emulator is set to “0: Low consumption disabled”. If “1” is selected, the E8a emulator will not control the MCU.

## Section 7 Debugger Setting

### 1. [Emulator Setting] dialog box

The [Emulator Setting] dialog box is provided for setting items that need to be set when the debugger is launched. The contents set from this dialog box (excluding [Power Supply] group box items) also become valid the next time the debugger is launched. When launching the debugger for the first time after creating a new project work space, the [Emulator Setting] dialog box is displayed with the Wizard.

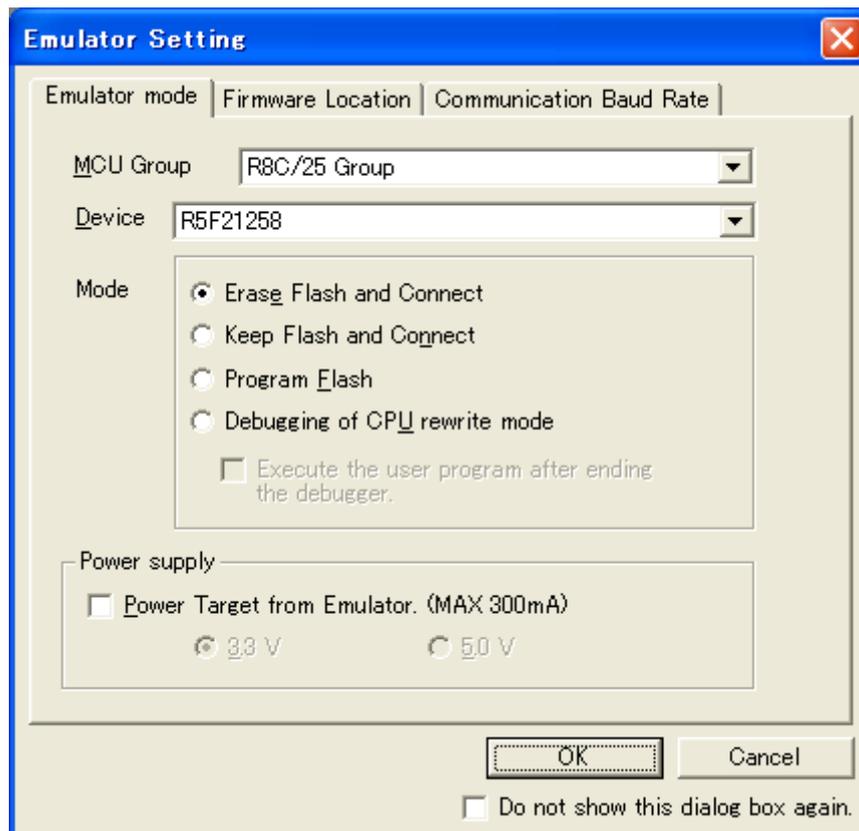


Figure 7.1 [Emulator Setting] Dialog Box

If you check “Do not show this dialog box again.” at the bottom of the [Emulator Setting] dialog box, the [Emulator Setting] dialog box will not be displayed the next time the debugger is launched. You can open the [Emulator Setting] dialog box using one of the following methods:

- After the debugger is launched, select Menu -> [Setup] -> [Emulator] -> [Emulator Setting...].
- Hold down the Ctrl key while launching the debugger.

When “Do not show this dialog box again.” is checked, the E8a does not supply power to the user system.

## 2. [Emulator mode] tab

Device selection, mode specification and power supply setting are made from the [Emulator mode] tab of the [Emulator Setting] dialog box.

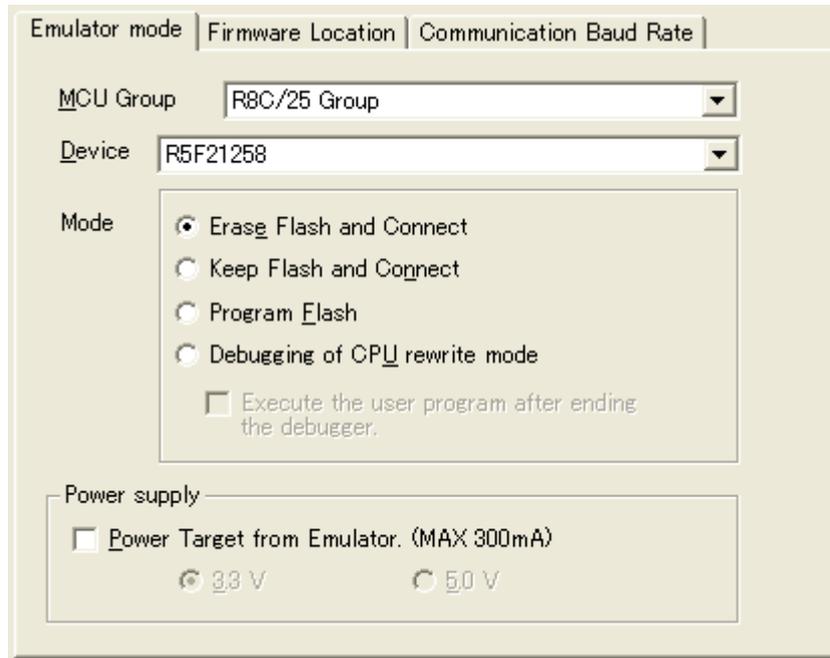


Figure 7.2 [Emulator mode] Tab

### [MCU Group]

Select the name of the MCU group to be used from the [MCU Group] drop-down list.

### [Device]

Select the type of MCU to be used from the [Device] drop-down list.

**[Mode]**

- Erase Flash and Connect

When starting the debugger, the E8a emulator erases the Flash memory data for the MCUs and simultaneously writes the E8a emulator program.

- Keep Flash and Connect

When launching the debugger, the E8a emulator retains the Flash memory data for the MCUs. Note that the area for the E8a emulator program and the vector area used by the E8a emulator will change.

- Program Flash

The E8a emulator starts as a simple programmer. When downloaded, the E8a writes only the user program (E8a emulator program is not written). Therefore, the program cannot be debugged in this mode.

- Debugging of CPU rewrite mode

Select this setting when debugging the program which rewrites the CPU. In this mode, the following debug operation which rewrites the Flash memory cannot be executed.

- Setting the PC break points

- Changing the memory contents in the Flash memory area

In this mode, when starting the debugger, the E8a emulator erases the Flash memory data for the MCUs and simultaneously writes the E8a emulator program.

When [Execute the user program after ending the debugger.] is selected, with the E8a emulator connected to the user system, the user program is executed at the same time the debugger is terminated. This check box setting is available only when the [Program Flash] mode is selected.

**[Power supply]**

When supplying power to the user system from the E8a, click the [Power Target from Emulator. (MAX 300mA)] check box.

**3. [Firmware Location] tab**

For details, see “1. Program area for the E8a emulator” in “Section 6. Notes on Using the E8a Emulator”.

## 4. [Communication Baud Rate] tab

Select communication baud rate between the E8a and MCU in the [Communication Baud Rate] tab. 500000bps (default setting) should be selected during normal use.

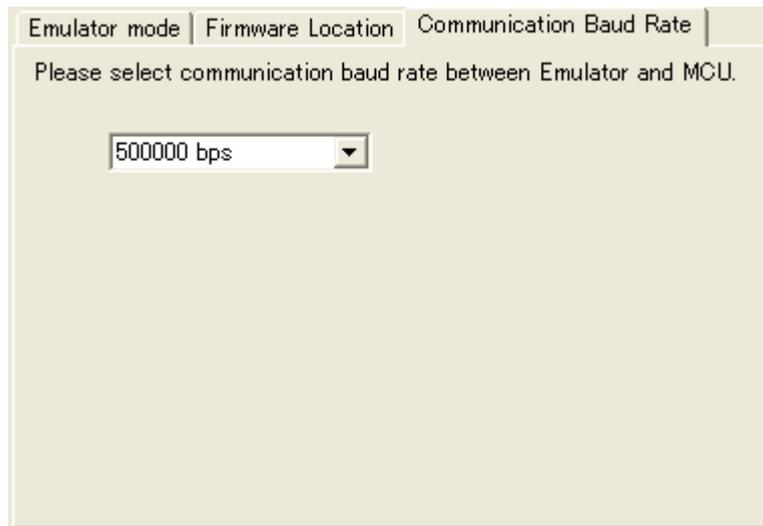


Figure 7.3 [Communication Baud Rate] Tab

## Section 8 Applicable Tool Chain and Third-party Products

With the R8C/2x Series E8a emulator, you can debug a module created by the inhouse tool chain and third-party products listed in Table 8.1 below.

Table 8.1 Applicable Tool Chain and Third-party Products

|                      |   |
|----------------------|---|
| Tool chain           | M3T-NC30WA V.5.20 Release 01 or later   |
| Third-party products | TASKING M16C C/C++/EC++ Compiler V.2.3r1 or later<br>IAR EWM16C V.2.12 or later |

### Notes on debugging the load modules created in ELF/DWARF2 format

If the load module was created in ELF/DWARF2 format using TASKING M16C C/C++/EC++ compiler V3.0r1, the precautionary note described below must be observed when displaying member variables of the base class in the watch window.

**Precautionary Note:**

If any class object with a base class is defined, the following problems may occur:

Case 1: Member variables of the base class cannot be referenced directly from the class object (\*1).

=> Use indirect references from the class object to refer to member variables of the base class (\*2) (\*3).

Case 2: If the PC value resides in any member function of a derived class, member variables of the base class cannot be referenced directly (\*4).

=> Use indirect references from “this” pointer to refer to member variables of the base class (\*5) (\*6).

---

E8a Emulator (R0E00008AKCE00)  
Additional Document for User's Manual  
Notes on Connecting the R8C/24 and R8C/25

Publication Date: Apr 30, 2010 Rev.3.00

Published by: Renesas Electronics Corporation

Edited by: Microcomputer Tool Development Department 2  
Renesas Solutions Corporation

---



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

---

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.  
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited  
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited  
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH  
Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.  
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.  
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited  
Unit 1601-1613, 16/F, Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.  
7F, No.363 Fu Shing North Road Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.  
1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.  
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.  
11F, Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

E8a Emulator (R0E00008AKCE00)  
Additional Document for User's Manual

