User's Manual

E8a Emulator

Additional Document for User's Manual

R0E00008AKCE00EP59

Renesas Microcomputer Development Environment System
R8C Family / R8C/3x Series
Notes on Connecting the R8C/32D, R8C/33D, R8C/35D and R8C/3GD

NOTICE:
There are corrections pages 19 and 24 in this document.
For details about the corrections, please refer to
Table 6.3 "SFRs Used by the E8a Emulator Program (1)" and
Table 6.4 "SFRs Used by the E8a Emulator Program (2)" on P.19.
Incorrect descriptions in 6.6 (4) are deleted on P.24.

Renesas Electronics
www.renesas.com

Rev.2.00 Apr 2010

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

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.
In this user's manual, the symbol # is used to show active LOW. (e.g. RESET#)

(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

2. E8a Emulator Specifications

2.1 Emulator specifications

Table 2.1 shows the E8a emulator specifications for the R8C/32D, R8C/33D, R8C/35D and R8C/3GD Groups. Table 2.2 shows the operating environment of the E8a emulator.

Table 2.1 E8a Emulator Specifications for the R8C/32D, R8C/33D, R8C/35D and R8C/3GD Groups

Target MCUs	R8C Family R8C/3x Series R8C/32D, R8C/33D, R8C/35D and R8C/3GD Groups
Available operating modes	Single-chip mode
Power voltages	1.8 - 5.5 V [*1] For details, refer to the hardware manual of the MCU.
Debug functions	
Break functions	- Address match break, 4 points, or - Address match break, 2 points + Data condition break, 1 point - PC break points (maximum 255 points) - Forced break
Trace functions	Last 4 branch instructions
Flash memory programming function	Available (when selecting the 'Program Flash' mode)
User interface	1-line clock asynchronous serial interface (communication via MODE pin)
MCU resources to be used	- ROM size: 2 KB - Stack 8 bytes - Address match interrupt
Emulator power supply	Unnecessary (USB bus powered, power supplied from the PC)
Interface with host machine	USB (USB 1.1, full speed)* * Also connectable to host computers that support USB 2.0 * Operation with all combinations of host machine, USB device and USB hub is not guaranteed for the USB interface.
Power supply function	Can supply 3.3 V or 5.0 V to the user system (maximum 300 mA)
Applicable emulator debugger	R8C E8a Emulator Debugger V.1.03.02 or later

Note

[*1] Set the power voltage to 2.7 V or above for rewriting the flash memory.
For details, refer to "6.6 (5) Note on debugging at less than 2.7V" on page 24.

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^2
	Inactive	: maximum 4.9 m/s^2
	Transportation	: maximum 14.7 m/s^2
Ambient gases	No corrosive gases	

2.2 Applicable tool chain and third-party products

You can debug a module created by the inhouse tool chain and third-party products listed in Table 2.3 below.

Table 2.3 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 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).

```
////////////////////////////////////////////////////
  *.h
        class BaseClass
        {
        public:
              int m_iBase;
        public:
              BaseClass() {
                    m_iBase = 0;
              }
              void BaseFunc(void);
        };

        class DerivedClass : public BaseClass
        {
        public:
              int m_iDerive;
        public:
              DerivedClass() {
                    m_iDerive     = 0;
              }
              void DerivedFunc(void);
        };


  *.cpp
        main()
        {
              class DerivedClass ClassObj;
              ClassObj.DerivedFunc();
              return;
        }

       void BaseClass::BaseFunc(void)
        {
              m_iBase = 0x1234;
        }

        void DerivedClass::DerivedFunc(void)
        {
              BaseFunc();
              m_iDerive     = 0x1234;
        }
////////////////////////////////////////////////////
```

Figure 2.1 Example code

```
////////////////////////////////////////////////////
     Case 1: If the PC value resides in the main() function
     (1)"ClassObj.m_iBase"                    : Cannot be referenced (*1)
     (2)"ClassObj.__b_BaseClass.m_iBase"      : Can be referenced (*2)
     (3)"ClassObj"
              -"__b_BaseClass"
                    -"m_iBase"                : Can be referenced (*3)
              -"m_iDerive"
                                  -: Expansion symbol
     Case 2: If the PC value resides in the DerivedClass::DerivedFunc() function
     (1)"m_iBase"                             : Cannot be referenced (*4)
     (2)"this->__b_BaseClass.m_iBase"         : Can be referenced (*5)
     (3)"__b_BaseClass.m_iBase"               : Can be referenced (*5)
     (4)"this"
              -"*"
                -"__b_BaseClass"
                      -"m_iBase"              : Can be referenced (*6)
                -"m_iDerive"
     (5)"__b_BaseClass"
              -"m_iBase"                      : Can be referenced (*6)
////////////////////////////////////////////////////
```

Figure 2.2 Watch window registration example

3. Connecting the E8a Emulator to the User System

3.1 Connector for connecting the E8a emulator and 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. Table 3.1 shows the recommended connector for the E8a emulator and Figure 3.2 shows E8a connecting connector pin assignments.

When designing the user system, refer to Figure 3.2 “E8a Connecting Connector Pin Assignments” and Section 3 “Connecting the E8a Emulator to the User System”.
Before designing the user system, be sure to read the E8a Emulator User’s Manual and related device hardware manuals.

Table 3.1 Recommended Connector

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

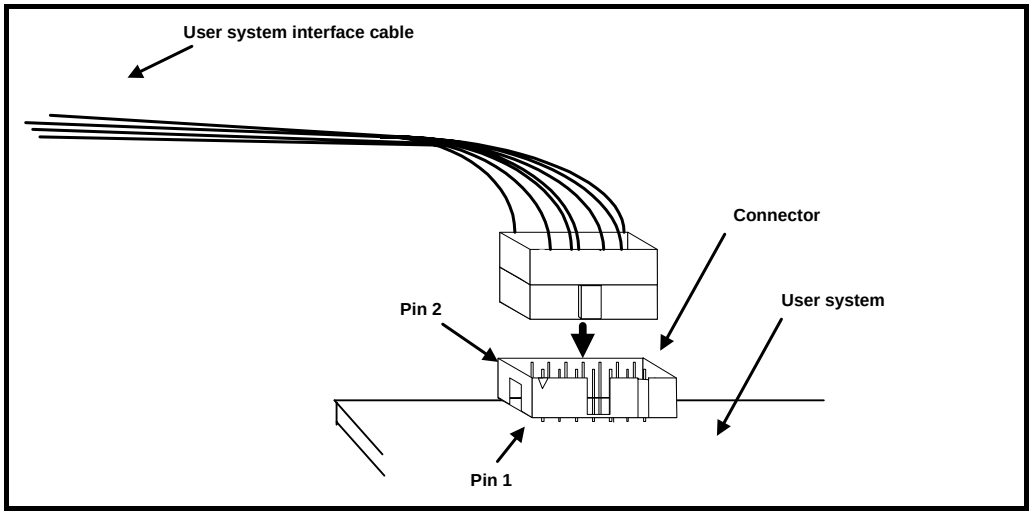
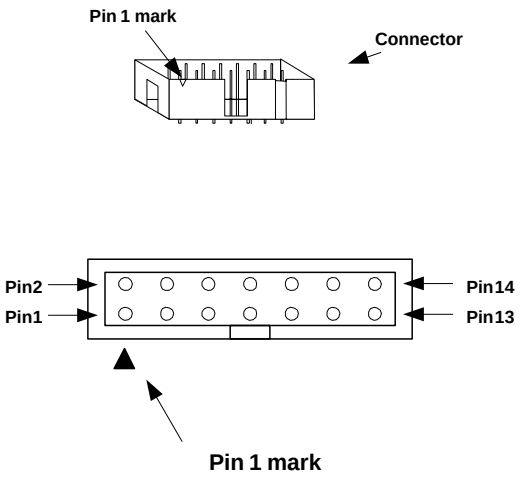


Figure 3.1 Connecting the User System Interface Cable with an E8a Connecting Connector

Notes

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



Pin NO	MCU Signals
1	N.C.
2	Vss
3	N.C.
4	Vss
5	N.C.
6	Vss
7	MODE
8	Vcc
9	N.C.
10	Vss
11	N.C.
12	Vss
13	RESET#
14	Vss

Figure 3.2 E8a Connecting Connector Pin Assignments

Notes

- Pin 14 is used for checking the connection between the E8a and the user system, and is 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.
- Note the pin assignments for the user system connector.
- Do not connect anything to the N.C. pin.

4. Examples of Pin Handling for Connecting the E8a

4.1 Examples of pin handling for connecting the E8a

Figure 4.1 shows an example of pin handling when connecting the E8a.
When using the E8a as a programmer, the connection specification between the E8a and the MCUs is the same as shown in Figure 4.1.

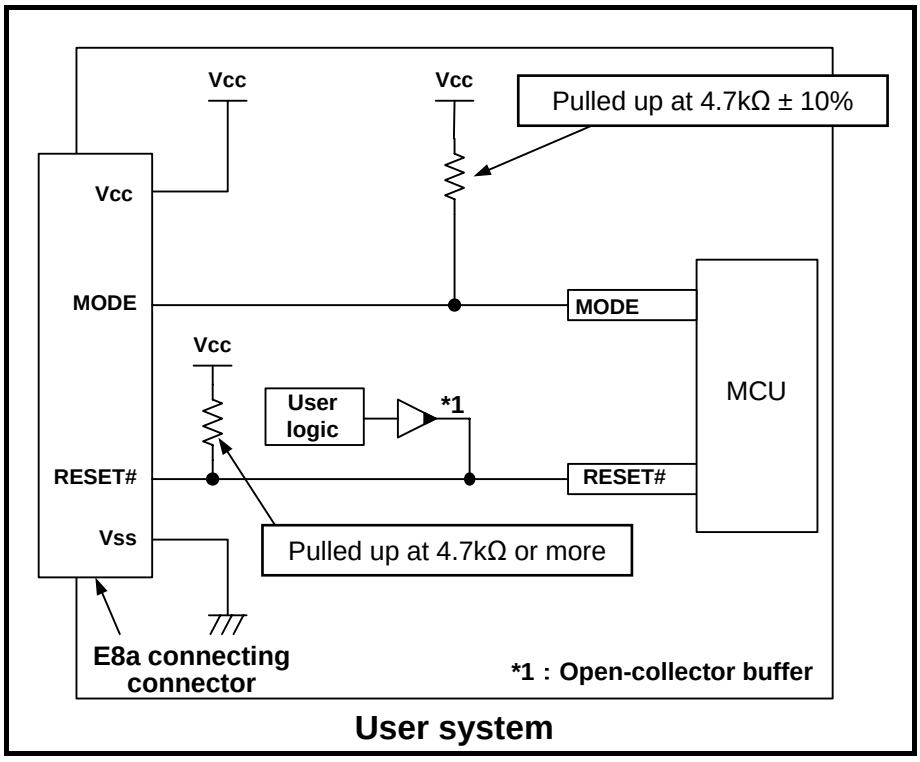


Figure 4.1 Example of an E8a Connection

(1) MODE pin

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.

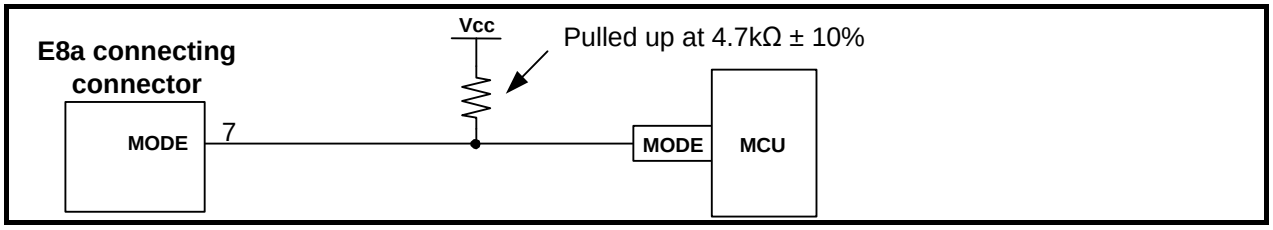


Figure 4.2 E8a Emulator and MODE Pin Connection

(2) RESET# pin

The 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 kΩ 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.

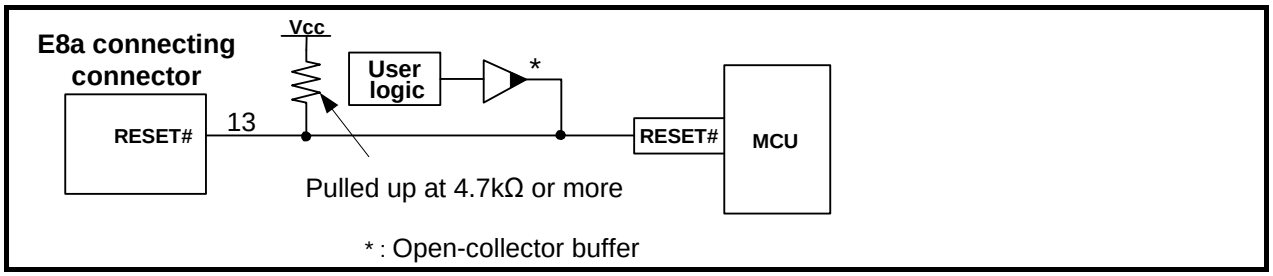


Figure 4.3 Example of a Reset Circuit

(3) Other pins

- Connect Vss and Vcc to the Vss and Vcc of the MCU, respectively.
- The amount of voltage input to Vcc must be within the specified range of the MCU.
- 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.
- Do not connect anything to the N.C. pin.

4.2 Interface circuit in the E8a emulator

Figure 4.4 shows the interface circuit in the E8a emulator. Use this figure as a reference when determining the pull-up resistance value.

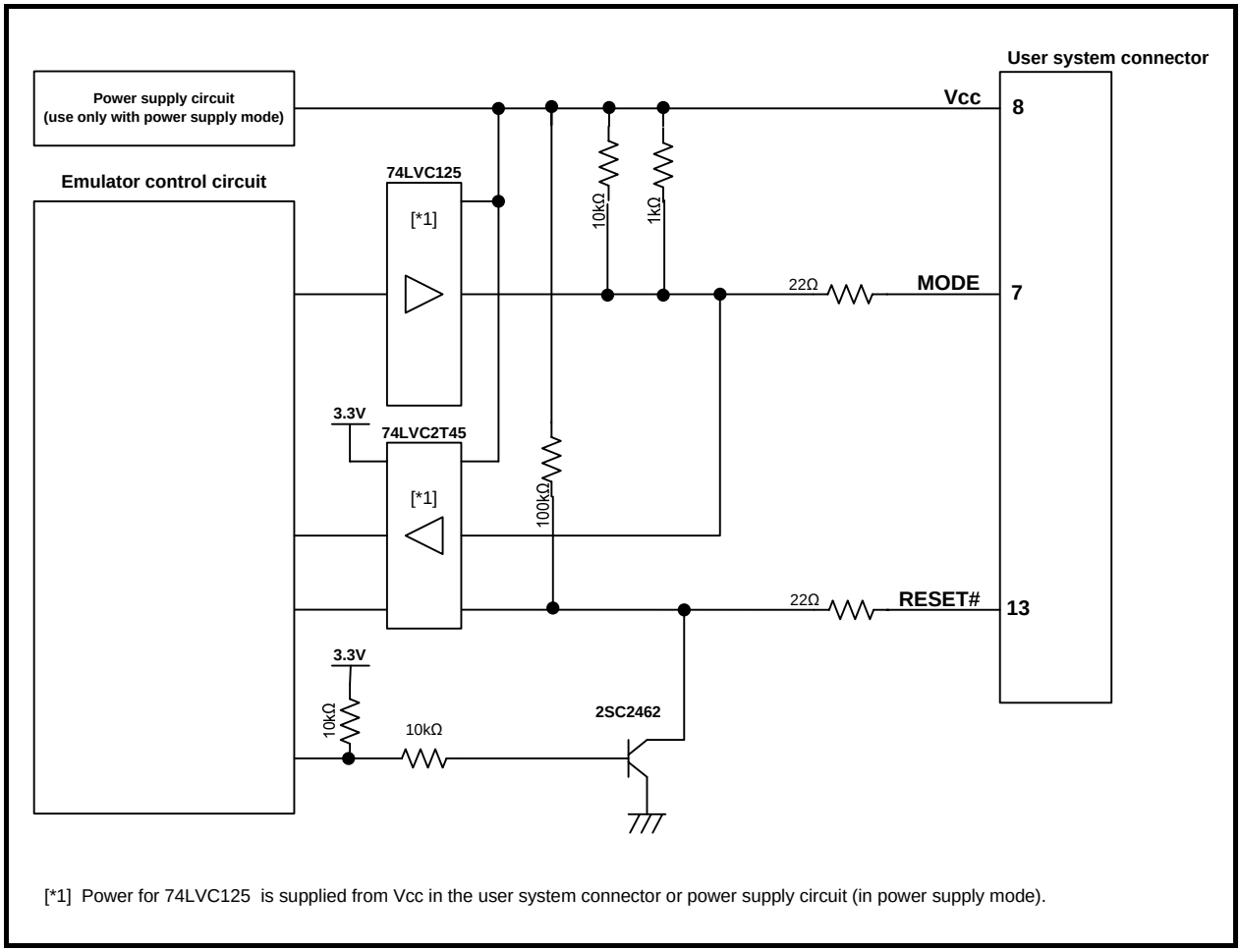


Figure 4.4 Interface Circuit inside the E8a Emulator (For Reference)

5. Emulator Debugger Setting

5.1 [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.

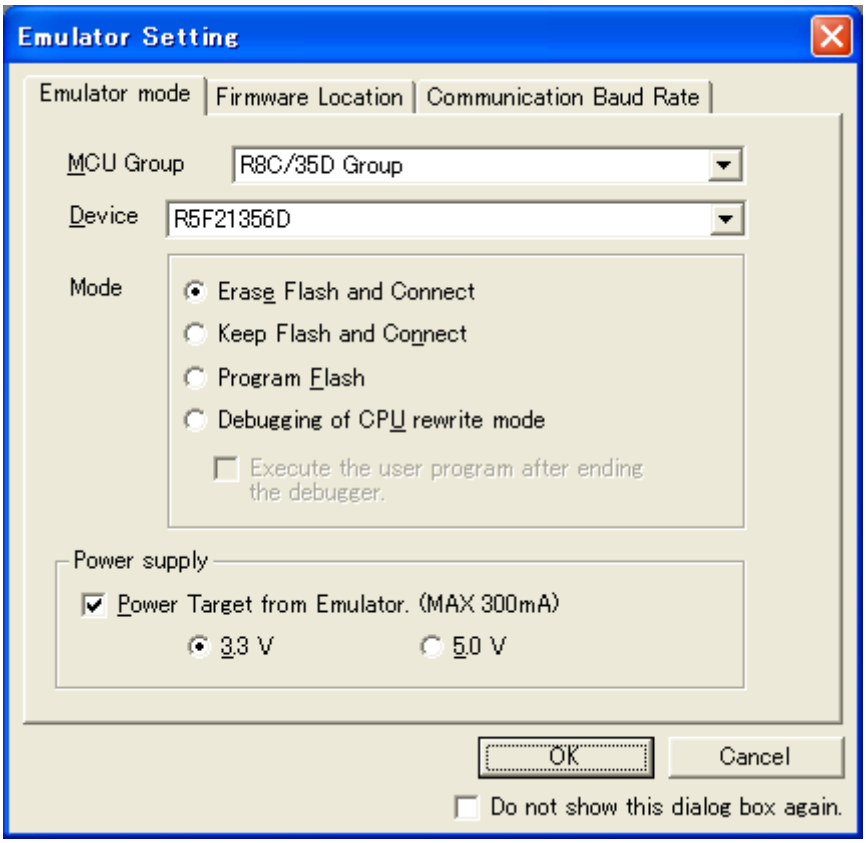


Figure 5.1 [Emulator Setting] Dialog Box

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.

5.2 [Emulator mode] tab

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

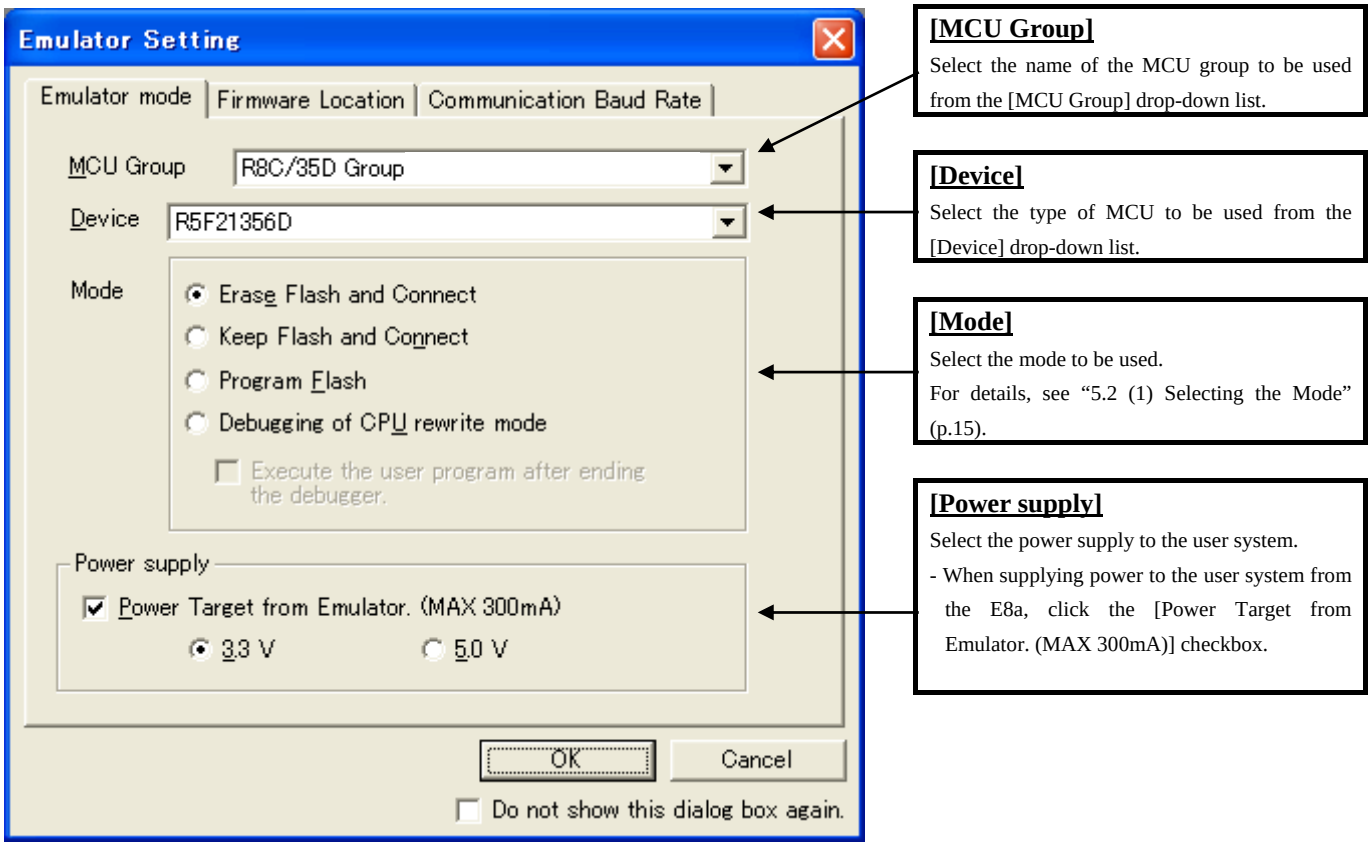


Figure 5.2 [Emulator mode] Tab of [Emulator Setting] Dialog Box

(1) Selecting the Mode

Table 5.1 Selecting the Mode

Mode	Usage	Description
Erase Flash and Connect [*2]	Debugging only [*1]	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 [*2]		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 [*2]	Simple programmer	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. 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.
Debugging of CPU rewrite mode [*3]	Debugging only [*1]	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.

Notes

[*1] These modes are available only for debugging. Programs written in these modes cannot be executed from the CPU. If you want to execute a program from the CPU, use Program Flash mode.

[*2] When starting up in these modes, lock bits in all the blocks of the flash memory will be unlocked. Note that the lock bits of the downloaded blocks will be unlocked after downloading the user program.

[*3] When debugging a program in CPU rewrite mode, memory reference or modification functions can be used. However, do not use these functions in the following condition.
· While write instruction is being executed to the register which requires continuous writing (ex. FMR13 bit)
The MCU does not recognize the writing is continuously executed if the write instruction is interrupted by the memory reference or modification process.

5.3 [Firmware Location] tab

You can specify the address of the firmware location in the [Firmware Location] tab.

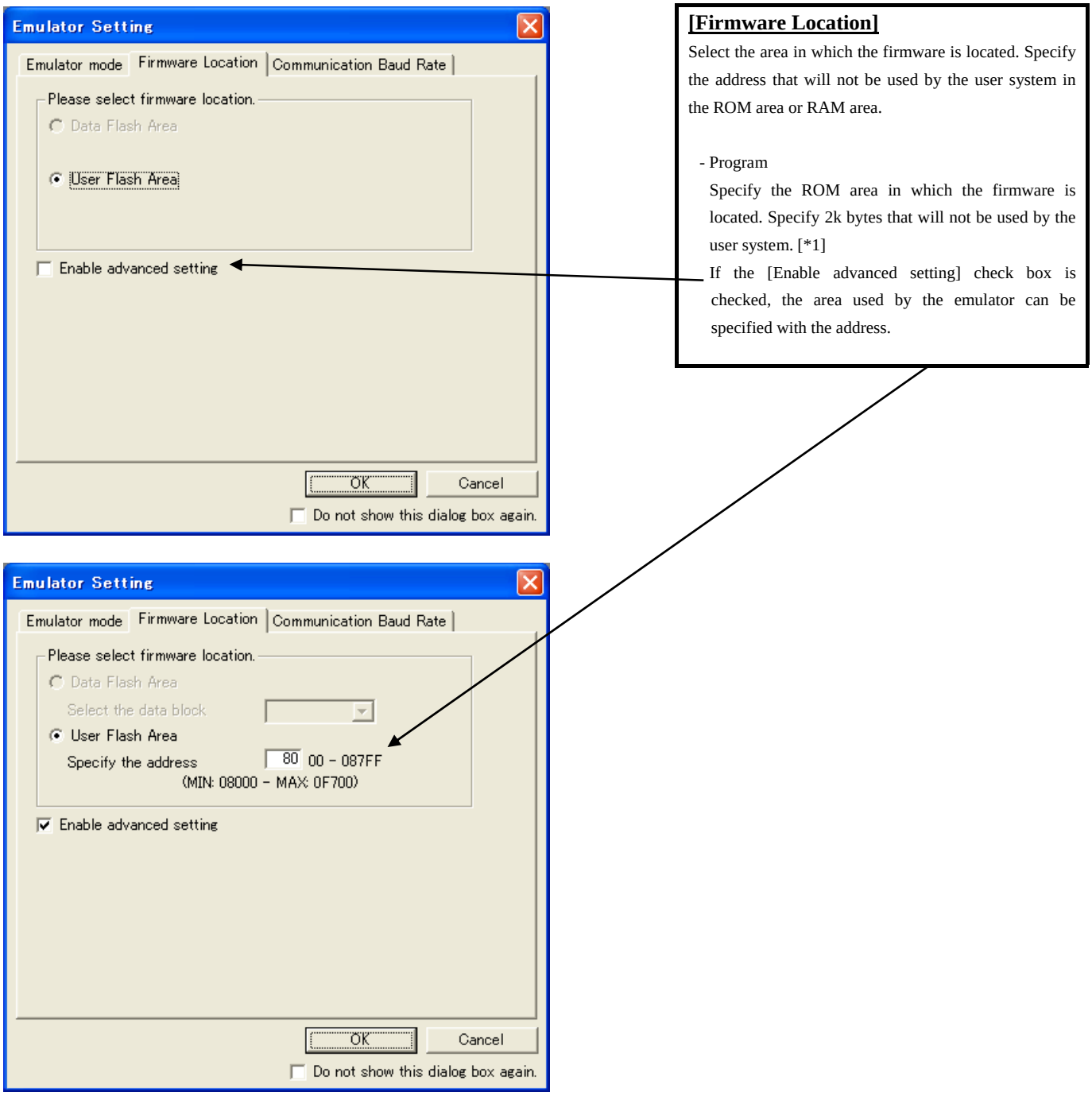


Figure 5.3 [Firmware Location] tab of [Emulator Setting] Dialog Box

Note

[*1] When using the MCU whose ROM size is other than 32 KB, the options in this [Firmware Location] tab are displayed in gray because this setting is unnecessary.

5.4 [Communication Baud Rate] tab

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

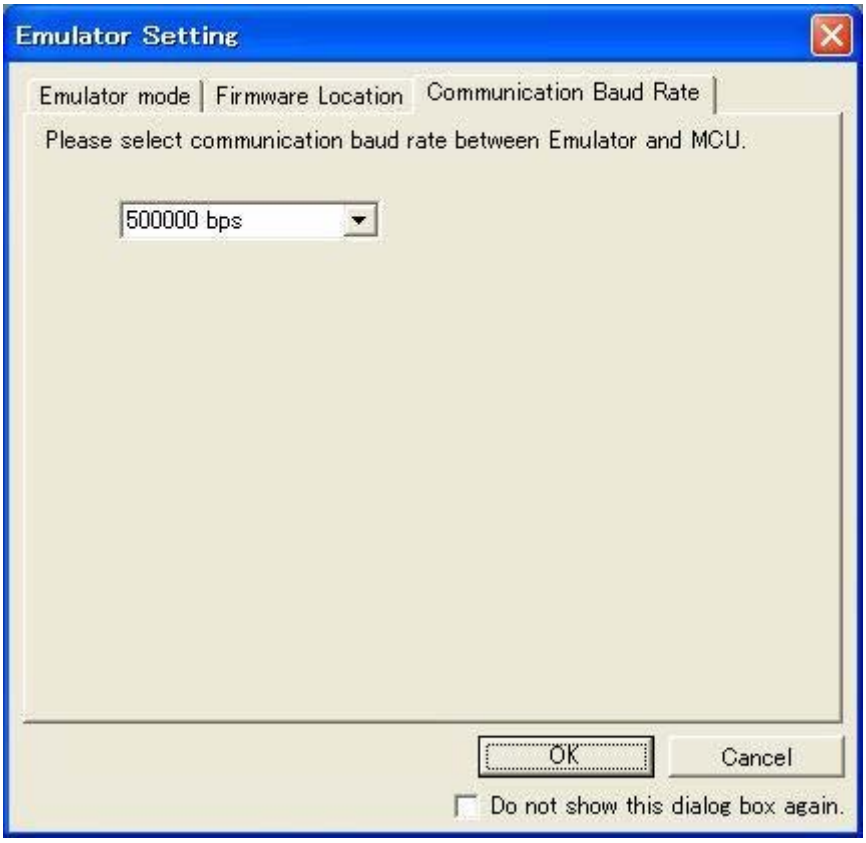


Figure 5.4 [Communication Baud Rate] Tab

6.Notes on Using the E8a Emulator

6.1 MCU resources used by 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	
		Program ROM	Data Flash	Vector Area	ROM Area
R8C/32D	R5F21321D	4 KB	-	FFE4h - FFE7h, FFE8h - FFEBh, FFECh - FFEFh, FFF4h - FFF7h, FFFCh - FFFFh	-
	R5F21322D	8 KB			-
	R5F21324D	16 KB			-
R8C/33D	R5F21331D	4 KB			-
	R5F21332D	8 KB			-
	R5F21334D	16 KB			-
	R5F21335D	24 KB			-
	R5F21336D	32 KB			2 KB of the ROM area [*1]
R8C/35D	R5F21354D	16 KB			-
	R5F21355D	24 KB			-
	R5F21356D	32 KB			2 KB of the ROM area [*1]
R8C/3GD	R5F213G1D	4 KB			-
	R5F213G2D	8 KB			-
	R5F213G4D	16 KB			-
	R5F213G5D	24 KB			-
	R5F213G6D	32 KB			2 KB of the ROM area [*1]

Note

[*1] When starting the debugger, the [Emulator Setting] dialog box is displayed. Specify the area which will not be used by the user system. For details, see 5.3 [Firmware Location] tab.

(2) Pins used by the E8a emulator

The E8a emulator controls the MCUs by using the following pins depending on the usage.

- For debugging/programming: RESET# pin and MODE pin

(3) Registers initialized by the E8a emulator

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

(4) 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.
- The SFRs listed in Table 6.3 and Table 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
01B6h	Flash memory control register 2	FMR2	Bit 7
0024h	High-speed on-chip oscillator control register 1	FRA1	All bits

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

Address	Register	Symbol	Bit	Notes on Using the E8a Emulator
0024h	High-speed on-chip oscillator control register 1	FRA1	All bits	[*1]
01C0h - 01C2h	Address match interrupt register 0	RMAD0	All bits	[*1]
01C3h	Address match interrupt enable register 0	AIER0	All bits	[*1]
01C4h - 01C6h	Address match interrupt register 1	RMAD1	All bits	[*1]
01C7h	Address match interrupt enable register 1	AIER1	All bits	[*1]

Note [*1]: Do not change this register value.

(5) Stack area used by the E8a emulator

The E8a emulator uses up to 8 bytes of the stack pointer (ISP) during a user program break. Therefore, set aside 8 bytes for the stack area.

(6) Reset

The reset vector is used by the E8a emulator program. If the MCU is reset (hardware reset) while executing the user program, control is transferred to the E8a emulator program and the user program is forced to stop. Do not perform any reset other than the hardware reset, otherwise the E8a emulator will run out of control.

If the automatic memory update is enabled in the memory or watch window, do not perform a hardware reset to the MCU. Otherwise the E8a emulator will run out of control.

(7) Interrupts used by the E8a emulator program (unusable)

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) Reserved area

The addresses not specified in the Hardware Manual of MCUs are reserved area. Do not change the contents. Otherwise, the E8a emulator cannot control the MCU.

(9) Count source protection mode

Count source protection mode cannot be debugged with the E8a emulator.

(10) High-speed on-chip oscillator

When debugging with the E8a emulator, the high-speed on-chip oscillator does not stop although the options for high-speed on-chip oscillator enable bit are available and FRA00 can be set to “high-speed on-chip oscillator off”. To check the functions of low power consumption etc. with high-speed on-chip oscillator off, make the evaluation with the final products or system manufactured by you in which only the user program is written to the MCU and the E8a emulator is disconnected.

The functions can also be checked by writing only the user program to the MCU in the ‘Program Flash’ mode, ending the debugger, then executing the user program. In the [Emulator Setting] dialog box displayed when starting the debugger, select [Program Flash], then check [Execute the user program after ending the debugger].

6.2 Flash memory

6.2.1 Notes on debugging in CPU rewrite mode

(1) Unrewritable area in CPU rewrite mode

When debugging in CPU rewrite mode, do not rewrite CPU for the flash memory block containing the following areas. If these areas are rewritten, the E8a emulator will not control the MCU.

- Fixed interrupt vector area
- Areas containing the E8a emulator programs

(2) Operation in CPU rewrite mode

- 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.
- When rewriting the Flash memory in the program area, select Menu -> [Setup] -> [Emulator] -> [System...] to open the [Configuration] dialog box in the High-performance Embedded Workshop. In this dialog box, change the [Flash memory synchronization] setting to [Flash memory to PC] and set the debugger cache to OFF. In this setting, the Flash memory is read whenever a break occurs, which takes some time. Use it with the [Disable] setting except when debugging in CPU rewrite mode.

6.2.2 Note on rewriting flash memory

(1) Do not reset nor execute debugging operations to the MCU when rewriting the flash memory.

Flash memory rewrite ends when the “Flash memory write end” is displayed in the output window of the High-performance Embedded Workshop. If the MCU is reset or debugged 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

6.2.3 Note on flash memory during user program execution

Do not rewrite the flash area from the memory window, etc., except from the user program during user program execution.

6.2.4 MCUs used for debugging

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.

6.2.5 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 Figure 6.1 [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.

The values written into the ID code area differs depending on the mode.

- 'Program Flash' mode: Contents of the user program
- Modes other than 'Program Flash' mode: FFh, FFh, FFh, FFh, FFh, FFh, FFh
  (regardless of the contents of the downloaded user program)

Table 6.5 ID Code Storage Area

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

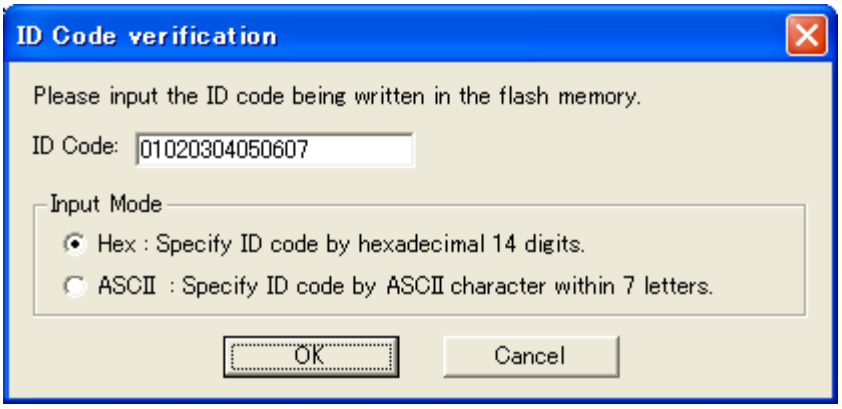


Figure 6.1 [ID Code verification] Dialog Box

Notes

Notes 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.

6.3 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.

Also, the E8a emulator sets the lower 4 bits of the option function select register 2 (OFS2: 0FFDBh) to 1111b.

- b1, b0: Watchdog timer underflow period set bit 11: 3FFFh
- b3, b2: Watchdog timer refresh acknowledgement period set bit 11: 100%

6.4 Power supply

(1) Consumption current

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.

(2) 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.

6.5 Operation during a user program halt

(1) 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 clock to operate. However, the peripheral features operate with the clock specified by the user program.

(2) Peripheral I/Os during a user program 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.

6.6 Debug functions

(1) 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.

(2)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.

(3) "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.

(4) 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 when the memory is referred to or modified in stop mode or wait mode, these mode will be cancelled. ~~Also, do not change to the wait mode with CM30 bit set to "1". Otherwise, the E8a emulator cannot control the MCU.~~

(5) Note on debugging at less than 2.7V

As flash rewrite occurs when the operations below are executed, if the operating voltage of the MCU is less than 2.7V, do not perform these operations:

- Downloading the user program
- Setting and canceling PC breaks (Setting/canceling event breaks are available)
- Rewriting the value of the Flash memory in the memory window

(6) Note on the CPU clock

Do not use the CPU clock at less than 32.768 kHz (XCIN clock).

(7) Low-current-consumption read mode

When debugging in low-current-consumption read mode or the state that the flash memory is stopped, 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 each mode or state is cancelled.

(8) 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 (see Figure 6.2).

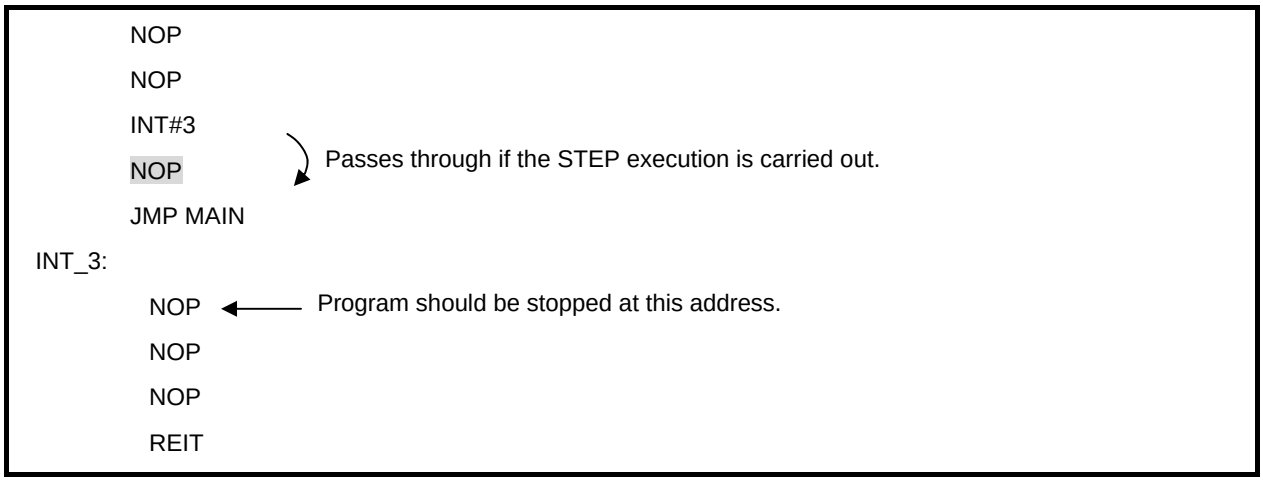


Figure 6.2 Example of Software Interrupt Instruction

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 (see Figure 6.3).

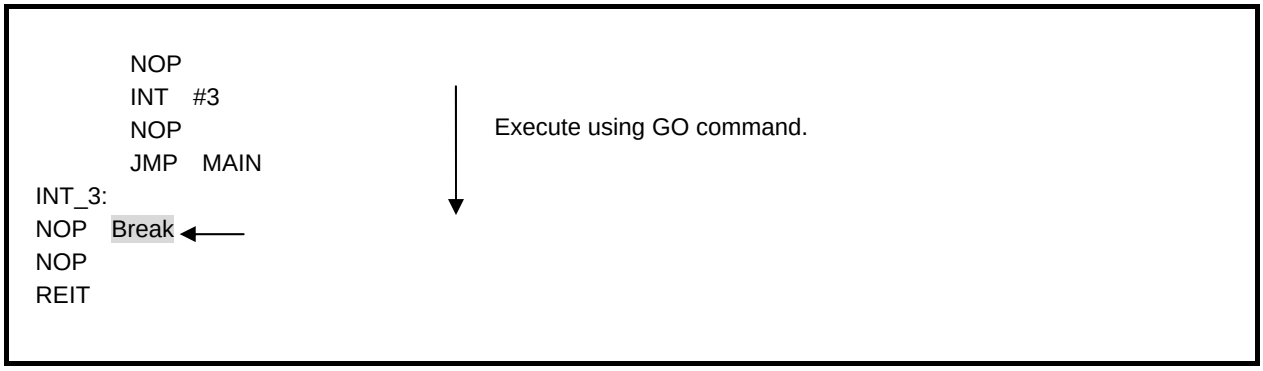


Figure 6.3 Example of INT Instruction

(9) Note on using automatic memory update

When the automatic memory update is enabled in the memory or watch window, do not execute Step Out or Multiple-step. Otherwise, it will take longer to update memory data and the operation will be delayed.

(10) 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.

E8a Emulator
Additional Document for User's Manual
Notes on Connecting the R8C/32D, R8C/33D, R8C/35D and R8C/3GD

Publication Date: Apr 30, 2010 Rev.2.00

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

Colophon 1.0

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

Renesas Electronics Corporation

REJ10J1944-0200