

E8a Emulator

Additional Document for User's Manual

R0E00008AKCE00EP52

Renesas Microcomputer Development Environment System
M16C Family / R32C/100 Series

Notes on Connecting the R32C/111, R32C/116, R32C/116A,
R32C/117, R32C/117A, R32C/118, R32C/118A, R32C/120,
R32C/121, R32C/145, R32C/151, R32C/152,
R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161

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

	Page
1. Inside the E8a Emulator User's Manual	4
2. E8a Emulator Specifications	5
2.1 Emulator specifications	5
2.2 Applicable tool chain and third-party products	6
3. Connecting the E8a Emulator to the User System	7
3.1 Connector for connecting the E8a emulator and the user system	7
4. Examples of Pin Handling for Connecting the E8a	9
4.1 MCUs other than R32C/160 and R32C/161	9
4.1.1 Examples of pin handling for connecting the E8a (whole connection)	9
4.2 R32C/111 (64-pin and 80-pin)	18
4.2.1 Examples of pin handling for connecting the E8a (whole connection)	18
4.3 R32C/160 and R32C/161	21
4.3.1 Examples of pin handling for connecting the E8a (whole connection)	21
4.4 Interface circuit in the E8a emulator	24
5. Emulator Debugger Setting	25
5.1 [Emulator Setting] dialog box	25
5.2 [Emulator mode] tab	26
5.3 [Firmware Location] tab	28
5.4 [MCU Setting] tab	29
5.5 [Communication Baud Rate] tab	30
6. Notes on Using the E8a Emulator	31
6.1 MCU resources used by the E8a emulator	31
6.2 Flash memory	35
6.2.1 Note on debugging in CPU rewrite mode	35
6.2.2 Note on rewriting flash memory	35
6.2.3 Note on flash memory during user program execution	35
6.2.4 MCUs used for debugging	35
6.2.5 Flash memory ID code	36
6.3 Count source protect mode	37
6.4 Power supply	37
6.5 Operation during a user program halt	37
6.6 Functions of the E ² dataFlash	38
6.7 Debug functions	38

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 R32C/100 Series. Table 2.2 shows the operating environment of the E8a emulator.

Table 2.1 E8a Emulator Specifications for the R32C/100 Series

Target MCUs	M16C Family R32C/100 Series R32C/111, R32C/116, R32C/116A, R32C/117, R32C/117A, R32C/118, R32C/118A, R32C/120, R32C/121, R32C/145, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161 Groups	
Available operating modes	Single-chip mode, Memory expansion mode * Microprocessor mode is not supported.	
Power voltages	3.0 - 5.5V For details, refer to the hardware manual of the MCU.	
Debug functions		
Break functions	- Address match break, 8 points - PC break points (maximum 255 points) - Forced break	
Trace functions	None	
Flash memory programming function	Available	
User interface	R32C/160, R32C/161	Clock-synchronous serial (communication via P44/P45/P46/P47)
	Other Groups	Clock-synchronous serial (communication via P64/P65/P66/P67)
MCU resources to be used	R32C/160, R32C/161	- ROM size: 4 KB - RAM size: 364 bytes - Stack 32 bytes - Address match interrupt - Pins P50 and P55 - UART1 function and P44/P45/P46/P47
	R32C/111, R32C/116, R32C/116A, R32C/117, R32C/117A, R32C/118, R32C/118A, R32C/120, R32C/121, R32C/145, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157	- ROM size: 4 KB - RAM size: 364 bytes - Stack 32 bytes - Address match interrupt - Pins P50 and P55 [*1] - UART1 function and P64/P65/P66/P67
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	R32C E8a Emulator Debugger V.1.01.00 or later	

Notes

[*1] For 64-pin and 80-pin versions of the R32C/111 Group, the E8a emulator uses pins P80 and P81 instead of pins P50 and P55.

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-NC100 V.1.01 Release 00 or later
------------	--------------------------------------

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

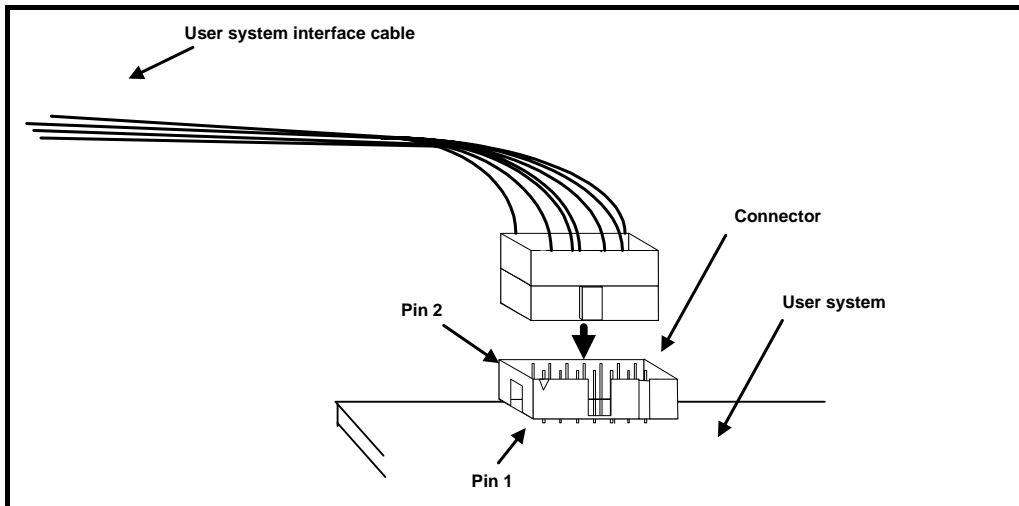


Figure 3.1 Connecting the User System Interface Cable with an E8a Connecting Connector

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.

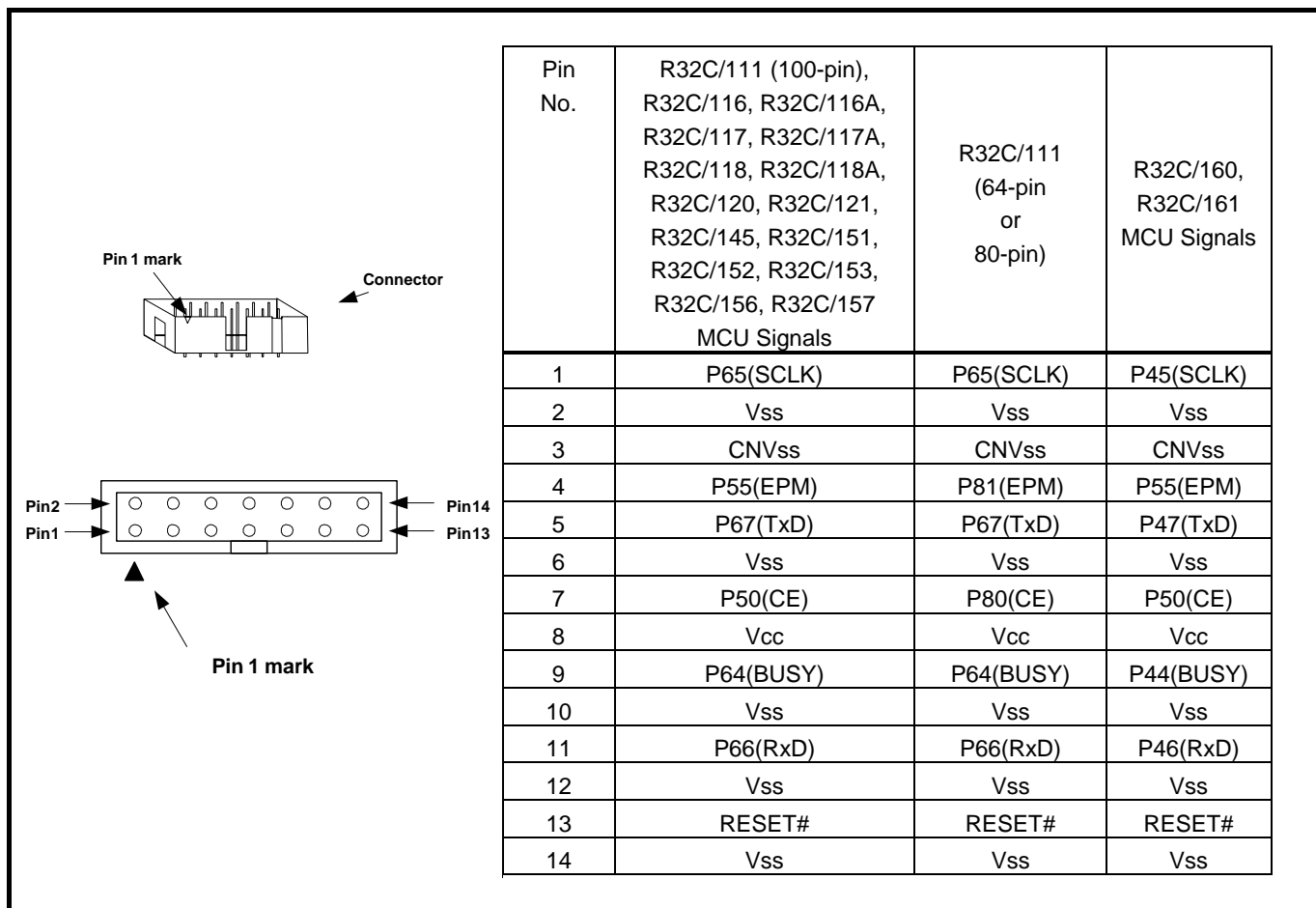


Figure 3.2 E8a Connecting Connector Pin Assignments

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.

4. Examples of Pin Handling for Connecting the E8a

4.1 MCUs other than R32C/160 and R32C/161

- Applicable MCUs: R32C/111 (100-pin), R32C/116, R32C/116A, R32C/117, R32C/117A, R32C/118, R32C/118A, R32C/120, R32C/121, R32C/145, R32C/151, R32C/152, R32C/153, R32C/156 and R32C/157

4.1.1 Examples of pin handling for connecting the E8a (whole connection)

The following show examples of pin handling for connecting the E8a. When using the E8a as a programmer, the connection specification between the E8a and the MCUs is the same as shown below.

- Single power supply and single-chip mode: See Figure 4.1.
- Single power supply and memory expansion mode: See Figure 4.2.
- Dual power supply and single-chip mode (R32C/111 (100-pin) only): See Figure 4.3.
- Dual power supply and memory expansion mode (R32C/111 (100-pin) only): See Figure 4.4.

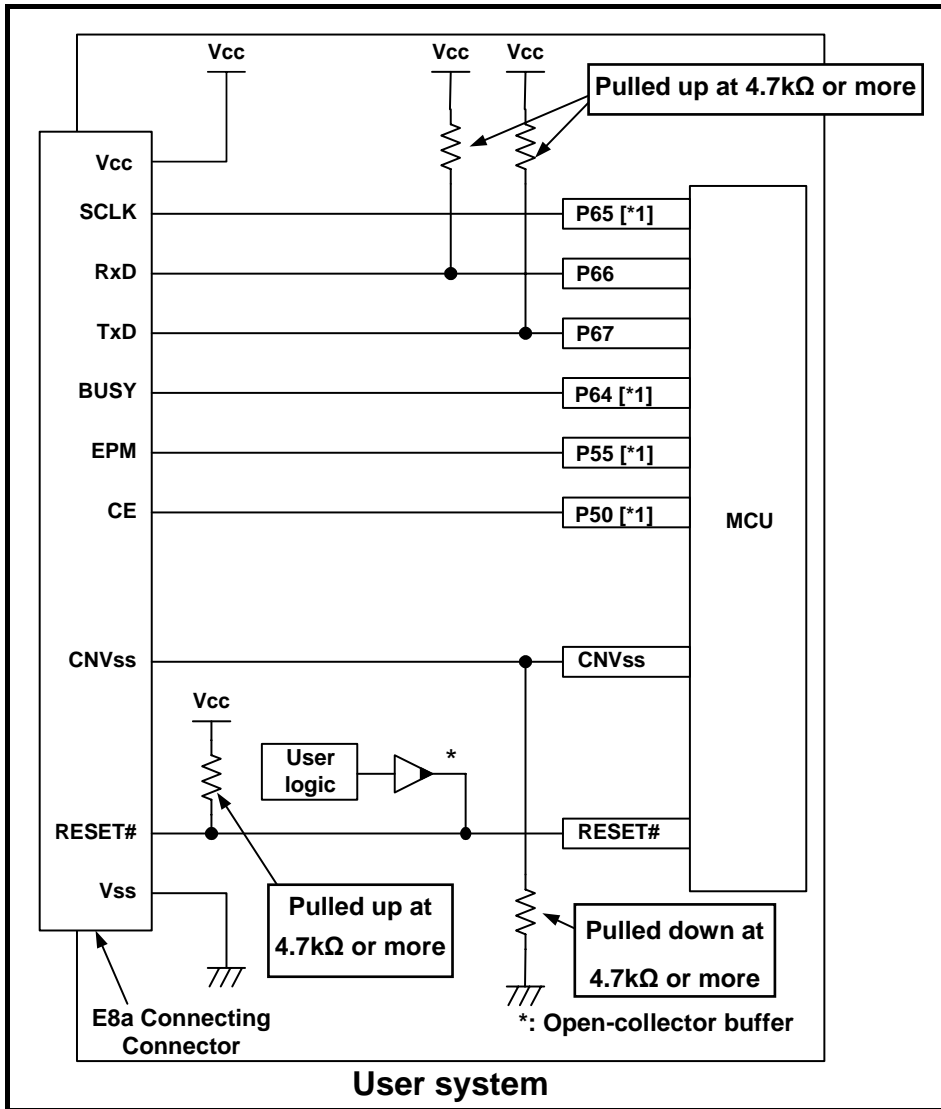


Figure 4.1 Example of an E8a Connection
 (Single Power Supply and Single-chip Mode, MCUs Other Than R32C/160 and R32C/161)

Note

[*1] For details on setting pins P64 and P65, refer to “(1) SCLK, RxD, TxD and BUSY pins” on page 14.
 For details on setting pins P50 and P55, refer to “(2) EPM# and CE# pins” on page 15.

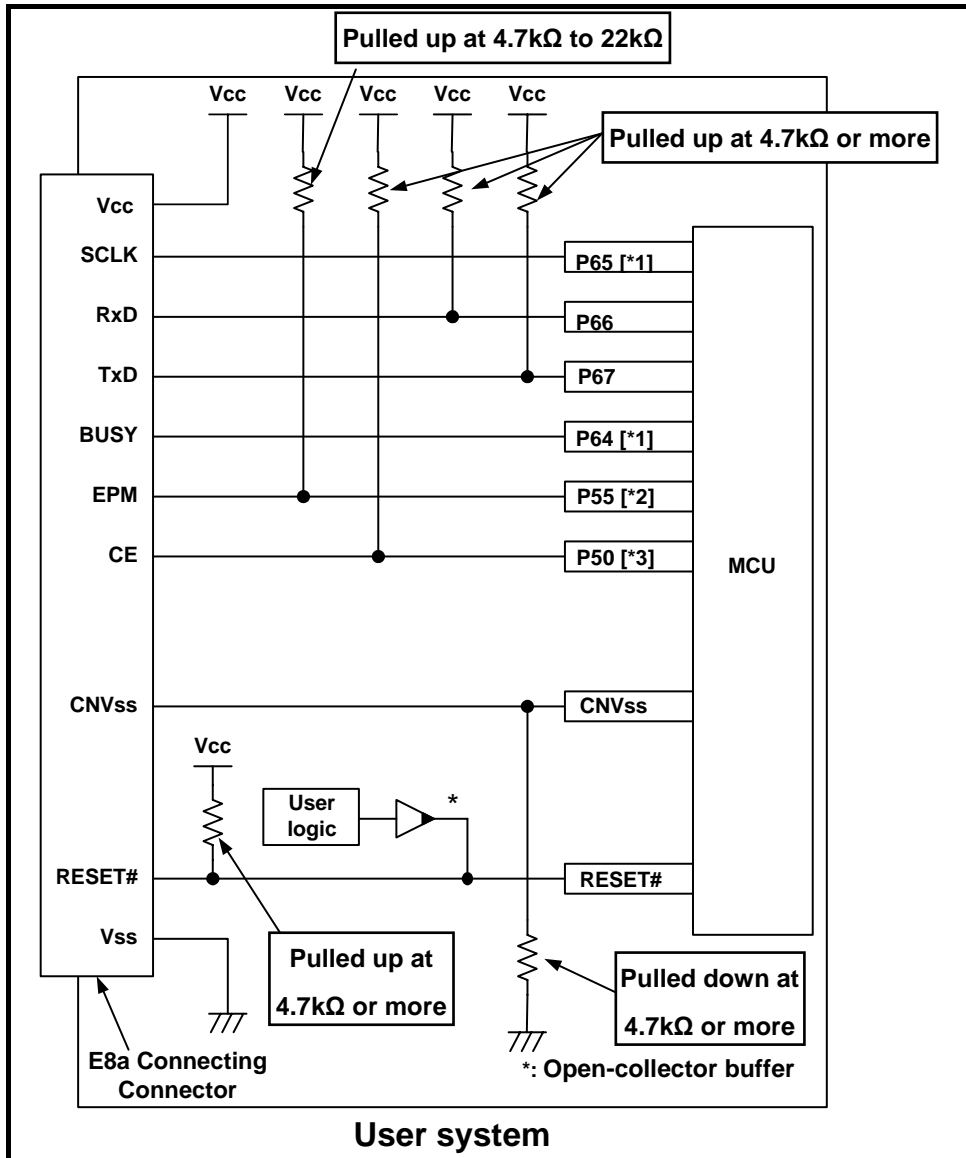


Figure 4.2 Example of an E8a Connection
 (Single Power Supply and Memory Expansion Mode, MCUs Other Than R32C/160 and R32C/161)

Notes

- [*1] For details on setting pins P64 and P65, refer to “(1) SCLK, RxD, TxD and BUSY pins” on page 14.
- [*2] The HOLD# signal cannot be used. Pull up P55 on the user system.
- [*3] P50 is used as the WRL#/WR# pin. The E8a emulator outputs “H” to the CE pin when going to boot mode (resetting the MCU). In other cases, the CE pin is in a Hiz state. This prevents signal collision between the E8a emulator and the MCU. The WRL#/WR# pin does not affect the memory because the pin has a low active signal.

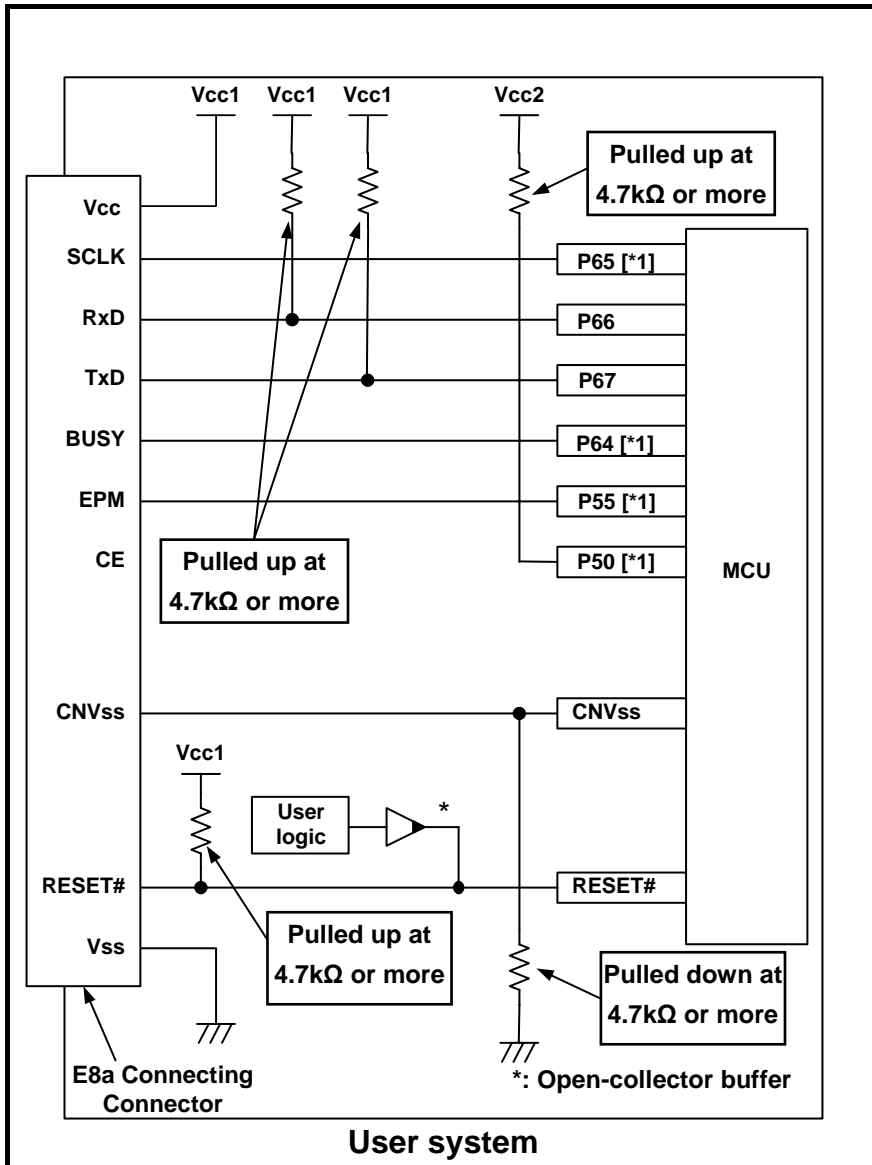


Figure 4.3 Example of an E8a Connection
 (Dual Power Supply and Single-chip Mode, R32C/111 (100-pin) Only)

Note

[*1] For details on setting pins P64 and P65, refer to “(1) SCLK, RxD, TxD and BUSY pins” on page 14.
 For details on setting pins P50 and P55, refer to “(1) EPM# and CE# pins” on page 15.

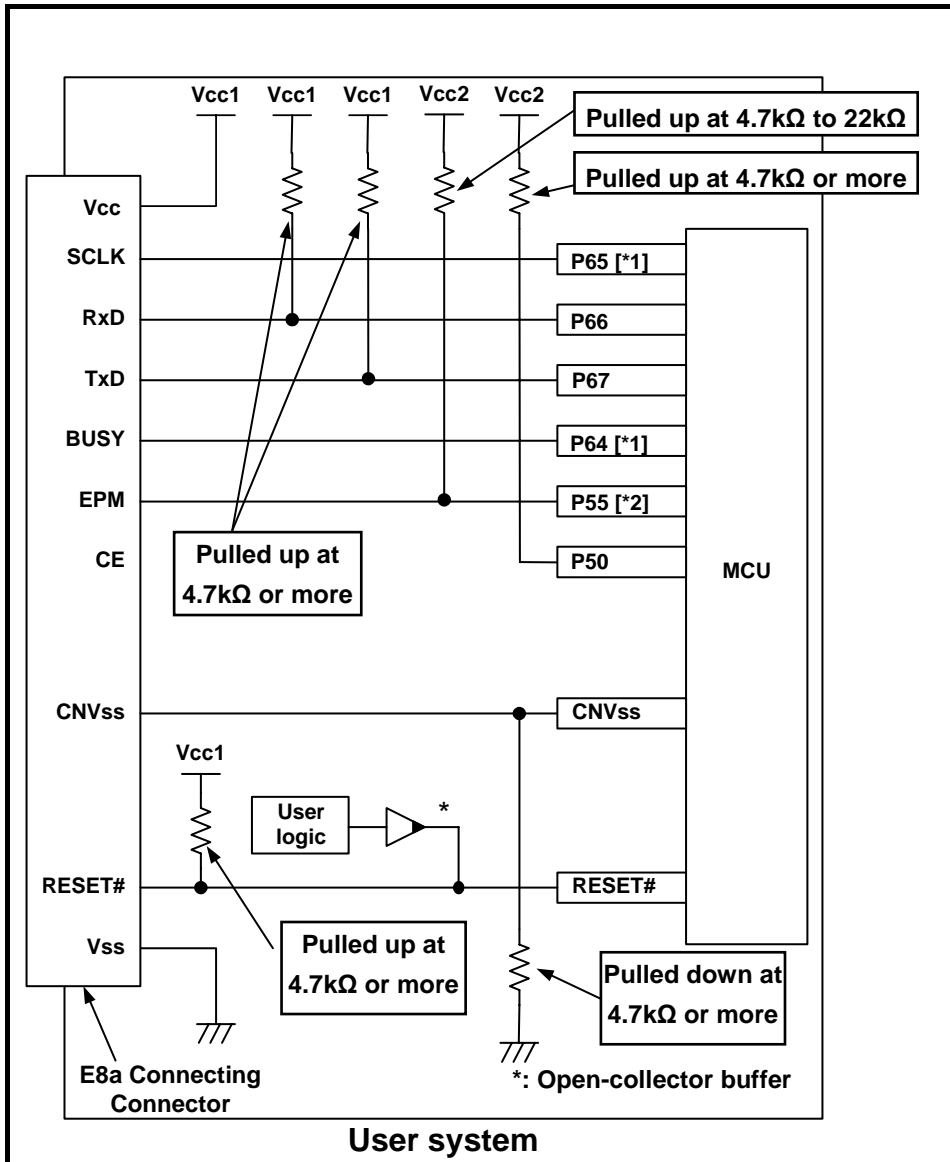


Figure 4.4 Example of an E8a Connection
 (Dual Power Supply and Memory Expansion Mode, R32C/111 (100-pin) Only)

Notes

- [*1] For details on setting pins P64 and P65, refer to “(1) SCLK, RxD, TxD and BUSY pins” on page 14.
- [*2] The HOLD# signal cannot be used. Pull up P55 on the user system.

(1) SCLK, RxD, TxD and BUSY pins

Pins P64(BUSY), P65(SCLK), P66(RxD) and P67(TxD) are used exclusively by the E8a emulator.

Connect pins P66 and P67 to the E8a emulator after pulling up the MCU pins at the Vcc (Vcc1) level.

For P64 and P65, pull up the pins at the Vcc (Vcc1) level or pull down them according to the MCU pin state after disconnecting the E8a emulator.

P64 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

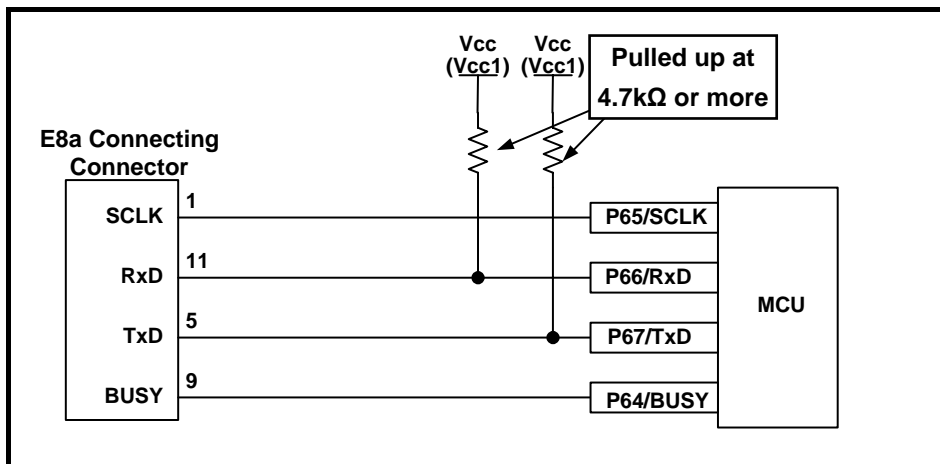


Figure 4.5 E8a Emulator and MCU Connection (MCUs Other Than R32C/160 and R32C/161)

(2) EPM# and CE# pins

The E8a emulator uses pins P50(CE#) and P55(EPM#) for MCU control. Connect the E8a emulator to the MCU pins.

1. Single power supply and single-chip mode

For P50 and P55, pull up the pins at the Vcc level or pull down them according to the MCU pin state after disconnecting the E8a emulator. P50 and P55 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

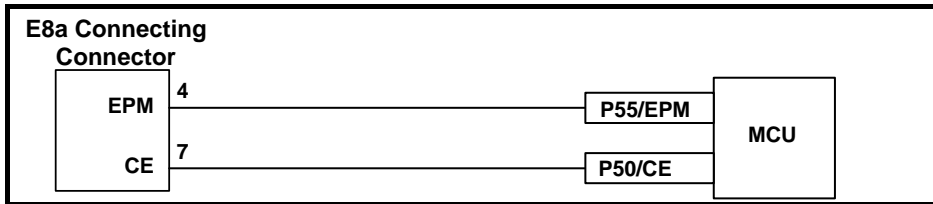


Figure 4.6 Connection of E8a Emulator and Pins P50 and P55 (Single Power Supply and Single-chip Mode, MCUs Other Than R32C/160 and R32C/161)

2. Single power supply and memory expansion mode

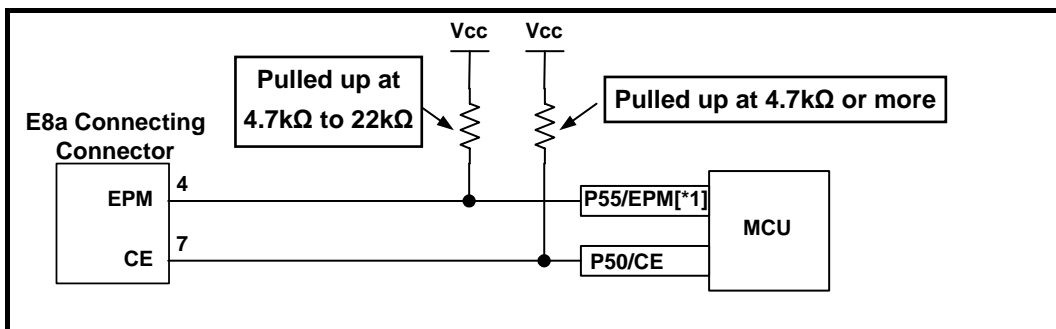


Figure 4.7 Connection of E8a Emulator and Pins P50 and P55 (Single Power Supply and Memory Expansion Mode, MCUs Other Than R32C/160 and R32C/161)

Note

[*1] The HOLD# signal cannot be used. Pull up P55 at the Vcc level on the user system.

3. Dual power supply and single-chip mode (R32C/111 (100-pin) Only)
 Pull up P55 at the Vcc2 level or pull down it according to the MCU pin state after disconnecting the E8a emulator. P55 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

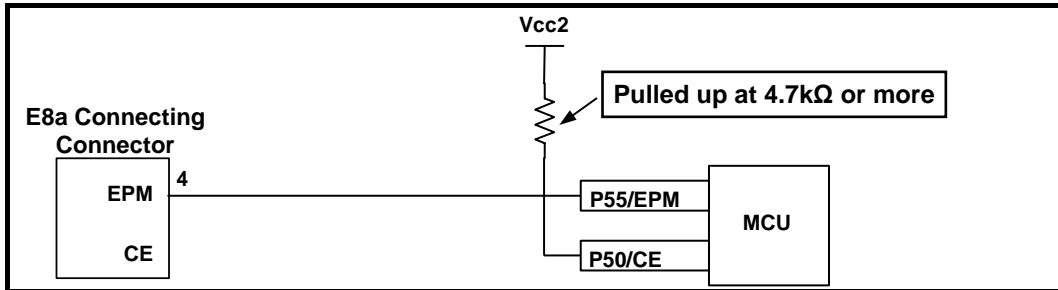


Figure 4.8 Connection of E8a Emulator and Pins P50 and P55 (Dual Power Supply and Single-chip Mode, R32C/111 (100-pin) Only)

4. Dual power supply and memory expansion mode (R32C/111 (100-pin) Only)

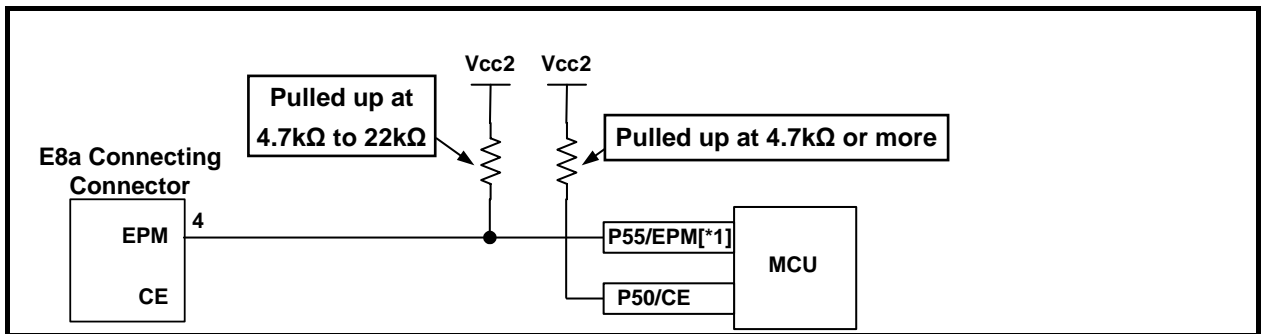


Figure 4.9 Connection of E8a Emulator and Pins P50 and P55 (Dual Power Supply and Memory Expansion Mode, R32C/111 (100-pin) Only)

Note

[*1] The HOLD# signal cannot be used. Pull up P55 at the Vcc2 level on the user system.

(3) CNVss pin

The E8a emulator uses the CNVss pin for MCU control.

Pull down the E8a emulator and MCU pins and connect the E8a emulator.

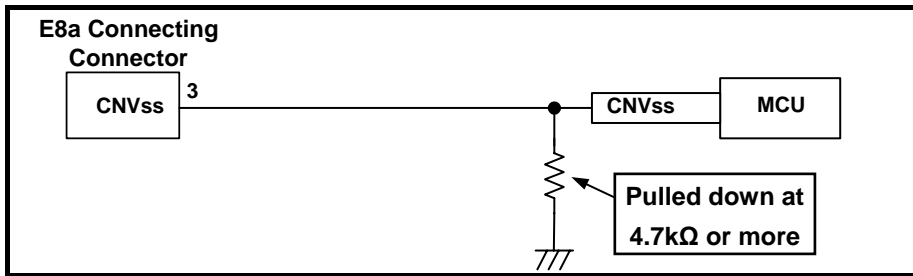


Figure 4.10 E8a Emulator and CNVss Pin Connection

(4) 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.

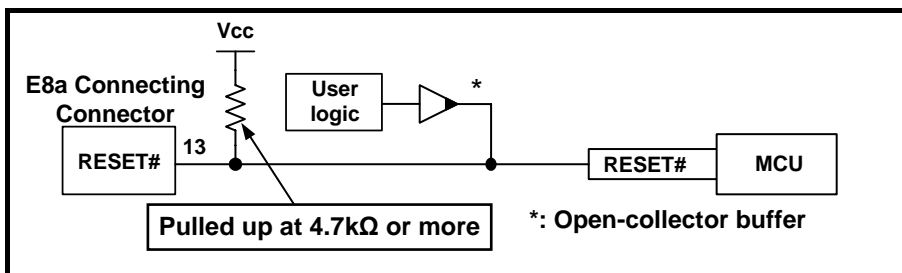


Figure 4.11 Example of a Reset Circuit

(5) Other pins

- Connect Vss and Vcc to the Vss and Vcc (Vcc1) of the MCU, respectively.
- The amount of voltage input to Vcc (Vcc1, Vcc2) must be within the specified range of the MCU.
- If NMI# interrupts are not used, make sure the NMI# pin is pulled up to the Vcc (Vcc1) pin through a resistor.
- 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 that pins 2, 6, 10, 12 and 14 are all connected to the Vss.

4.2 R32C/111 (64-pin and 80-pin)

4.2.1 Examples of pin handling for connecting the E8a (whole connection)

The following show examples of pin handling for connecting the E8a. When using the E8a as a programmer, the connection specification between the E8a and the MCUs is the same as shown below.

- Single power supply and single-chip mode: See Figure 4.12.

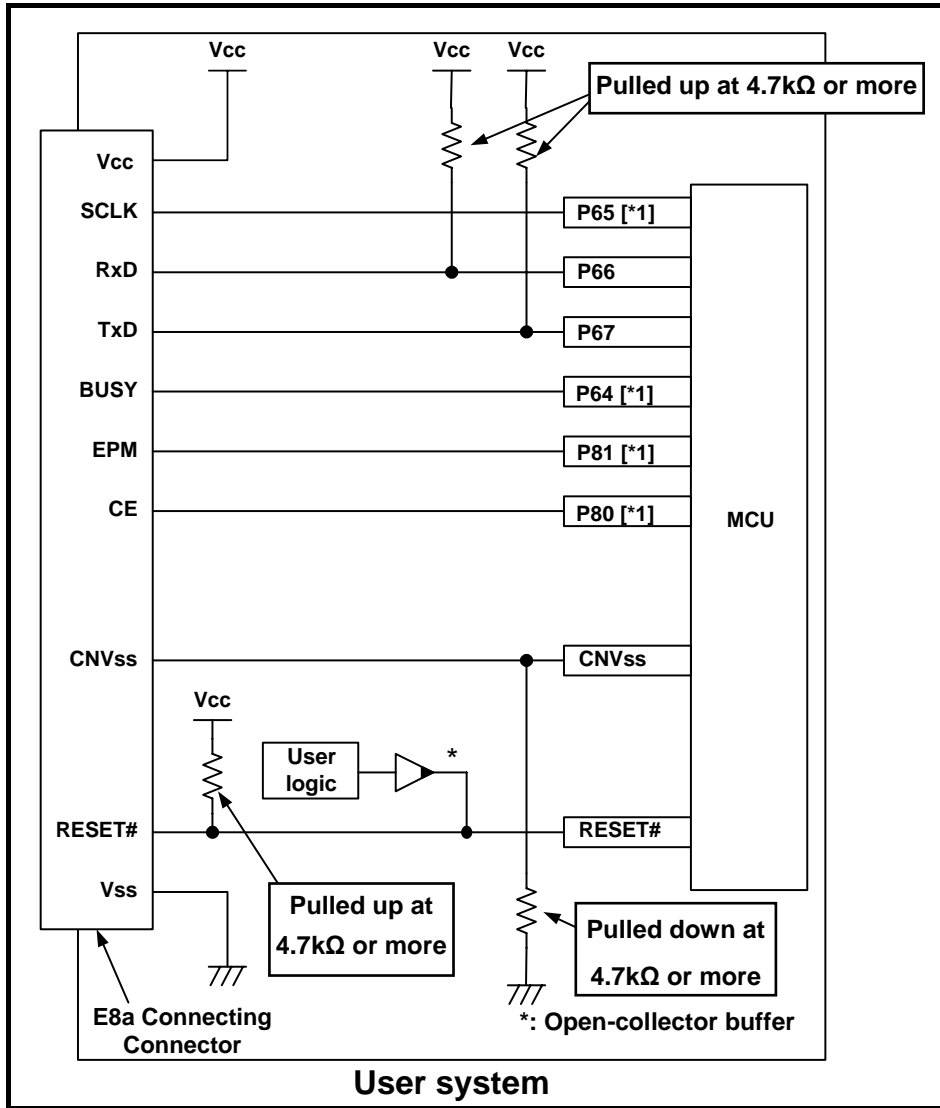


Figure 4.12 Example of an E8a Connection
(Single Power Supply and Single-chip Mode, R32C/111 (64-pin and 80-pin) Only)

Note

[*1] For details on setting pins P64 and P65, refer to “4.2.1 (1) SCLK, RxD, TxD and BUSY pins” on page 19.
For details on setting pins P80 and P81, refer to “4.2.1 (2) EPM# and CE# pins” on page 19.

(1) SCLK, RxD, TxD and BUSY pins

Pins P64(BUSY), P65(SCLK), P66(RxD) and P67(TxD) are used exclusively by the E8a emulator.

Connect pins P66 and P67 to the E8a emulator after pulling up the MCU pins at the Vcc (Vcc1) level.

For P64 and P65, pull up the pins at the Vcc (Vcc1) level or pull down them according to the MCU pin state after disconnecting the E8a emulator.

P64 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

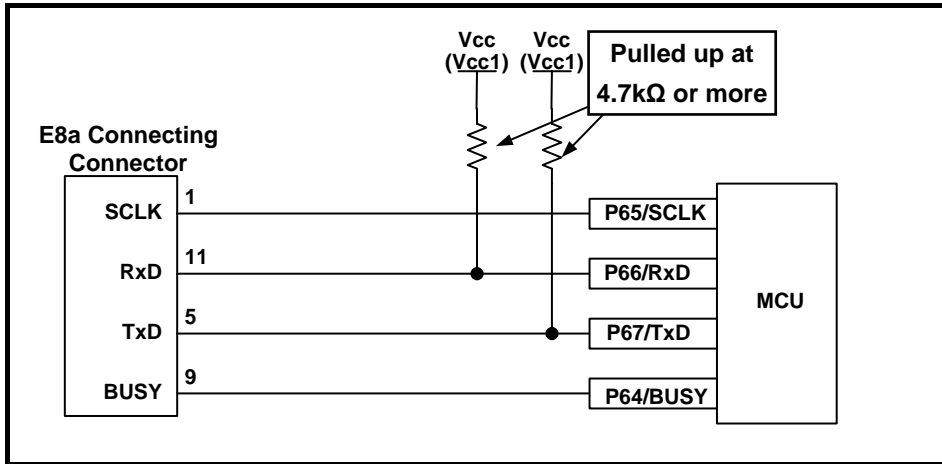


Figure 4.13 E8a Emulator and MCU Connection (R32C/111 (64-pin and 80-pin) Only)

(2) EPM# and CE# pins

The E8a emulator uses pins P80(CE#) and P81(EPM#) for MCU control.

Connect the E8a emulator to the MCU pins.

For P80 and P81, pull up the pins at the Vcc level or pull down them according to the MCU pin state after disconnecting the E8a emulator. P80 and P81 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

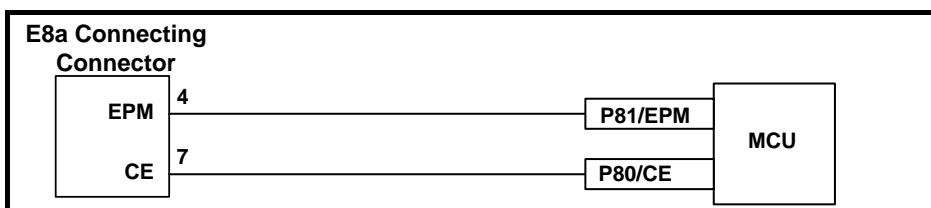


Figure 4.14 Connection of E8a Emulator and Pins P80 and P81 (R32C/111 (64-pin and 80-pin) Only)

(3) CNVss pin

The E8a emulator uses the CNVss pin for MCU control.

Pull down the E8a emulator and MCU pins and connect the E8a emulator.

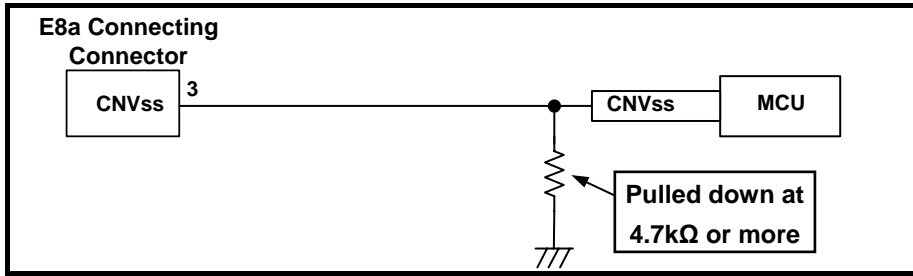


Figure 4.15 E8a Emulator and CNVss Pin Connection

(4) 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.

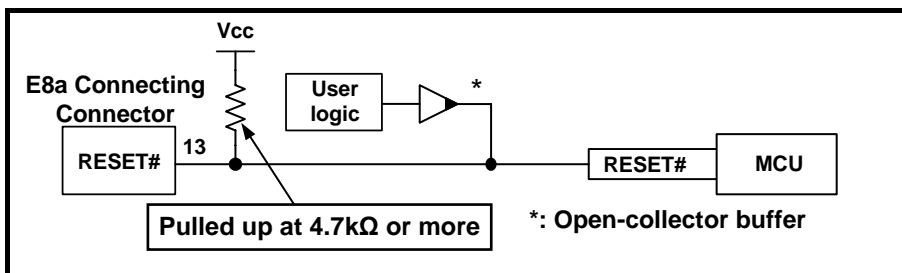


Figure 4.16 Example of a Reset Circuit

(5) 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.
- If NMI# interrupts are not used, make sure the NMI# pin is pulled up to the Vcc pin through a resistor.
- 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 that pins 2, 6, 10, 12 and 14 are all connected to the Vss.

4.3 R32C/160 and R32C/161

4.3.1 Examples of pin handling for connecting the E8a (whole connection)

The following show examples of pin handling for connecting the E8a. When using the E8a as a programmer, the connection specification between the E8a and the MCUs is the same as shown below.

- Single-chip mode: See Figure 4.17.

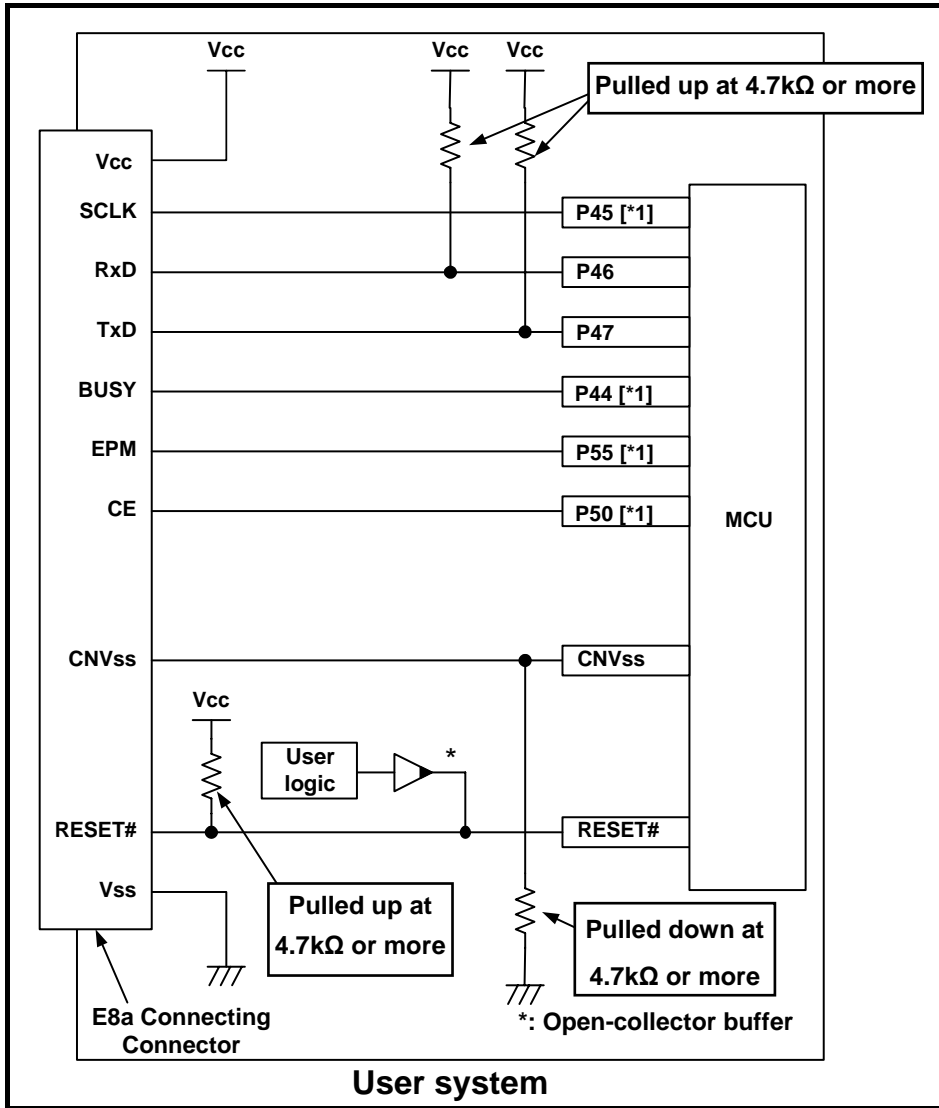


Figure 4.17 Example of an E8a Connection
(Single-chip Mode, R32C/160 and R32C/161 Only)

Note

[*1] For details on setting pins P44 and P45, refer to “4.3.1 (1) SCLK, RxD, TxD and BUSY pins” on page 22.
For details on setting pins P50 and P55, refer to “4.3.1 (2) EPM# and CE# pins” on page 22.

(1) SCLK, RxD, TxD and BUSY pins

Pins P44(BUSY), P45(SCLK), P46(RxD) and P47(TxD) are used exclusively by the E8a emulator.

Connect pins P46 and P47 to the E8a emulator after pulling up the MCU pins at the Vcc (Vcc1) level.

For P44 and P45, pull up the pins at the Vcc (Vcc1) level or pull down them according to the MCU pin state after disconnecting the E8a emulator.

P44 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

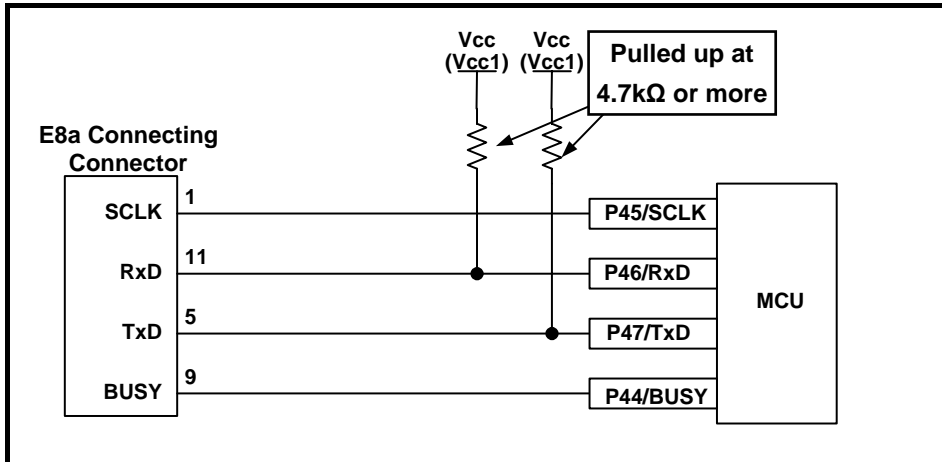


Figure 4.18 E8a Emulator and MCU Connection (R32C/160 and R32C/161 Only)

(2) EPM# and CE# pins

The E8a emulator uses pins P50(CE#) and P55(EPM#) for MCU control.

Connect the E8a emulator to the MCU pins.

1. Single-chip mode

For P50 and P55, pull up the pins at the Vcc level or pull down them according to the MCU pin state after disconnecting the E8a emulator. P50 and P55 may be in a Hiz state while the E8a emulator is active. Therefore, set the pin resistance value so the voltage cannot be at the midpoint potential, depending on the voltage dividing of the resistance inside the E8a emulator (Figure 4.22 on page 24).

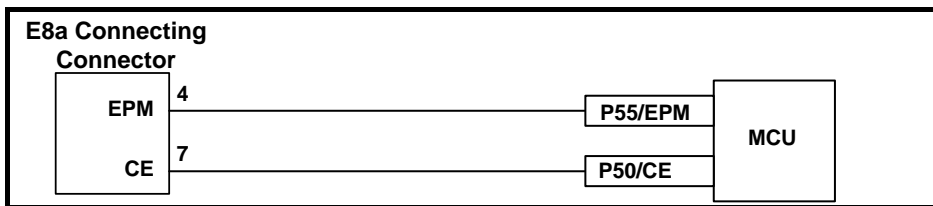


Figure 4.19 Connection of E8a Emulator and Pins P50 and P55 (Single-chip Mode, R32C/160 and R32C/161 Only)

(3) CNVss pin

The E8a emulator uses the CNVss pin for MCU control.

Pull down the E8a emulator and MCU pins and connect the E8a emulator.

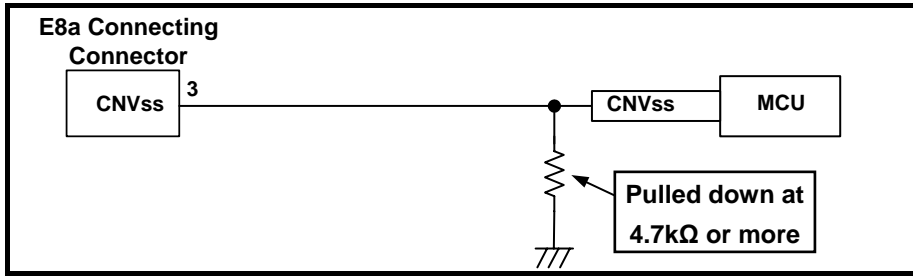


Figure 4.20 E8a Emulator and CNVss Pin Connection

(4) 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.

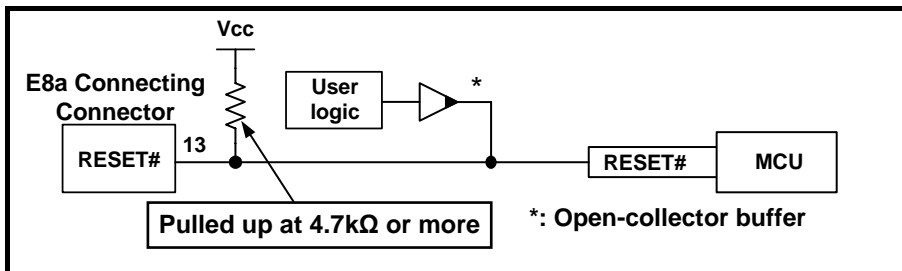


Figure 4.21 Example of a Reset Circuit

(5) 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.
- If NMI# interrupts are not used, make sure the NMI# pin is pulled up to the Vcc pin through a resistor.
- 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 that pins 2, 6, 10, 12 and 14 are all connected to the Vss.

4.4 Interface circuit in the E8a emulator

Figure 4.22 shows the interface circuit in the E8a emulator. Use this figure as a reference when determining the pull-up resistance value.

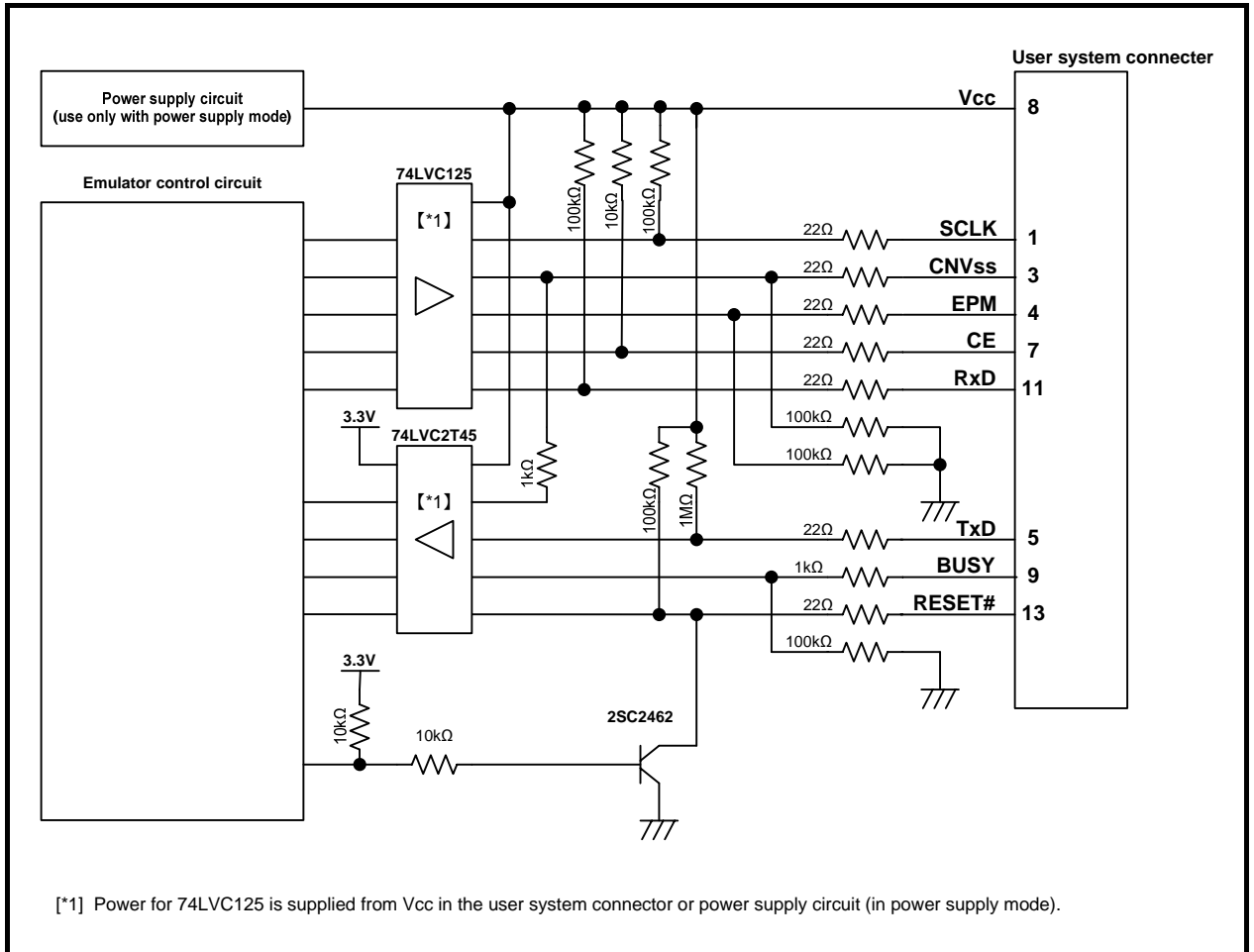


Figure 4.22 Interface Circuit inside the E8a Emulator (For Reference)

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.

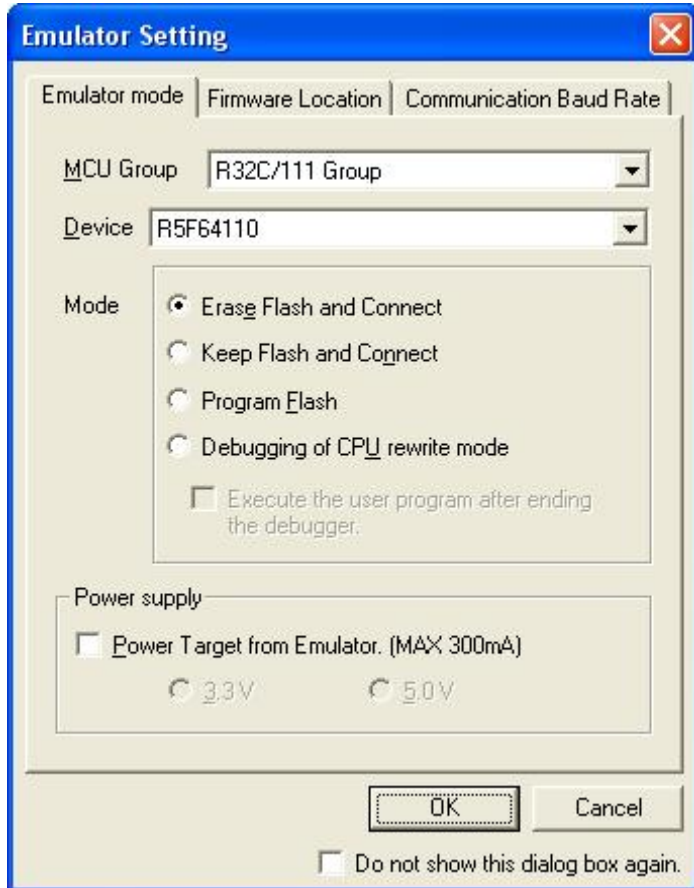


Figure 5.1 [Emulator Setting] Dialog Box

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.

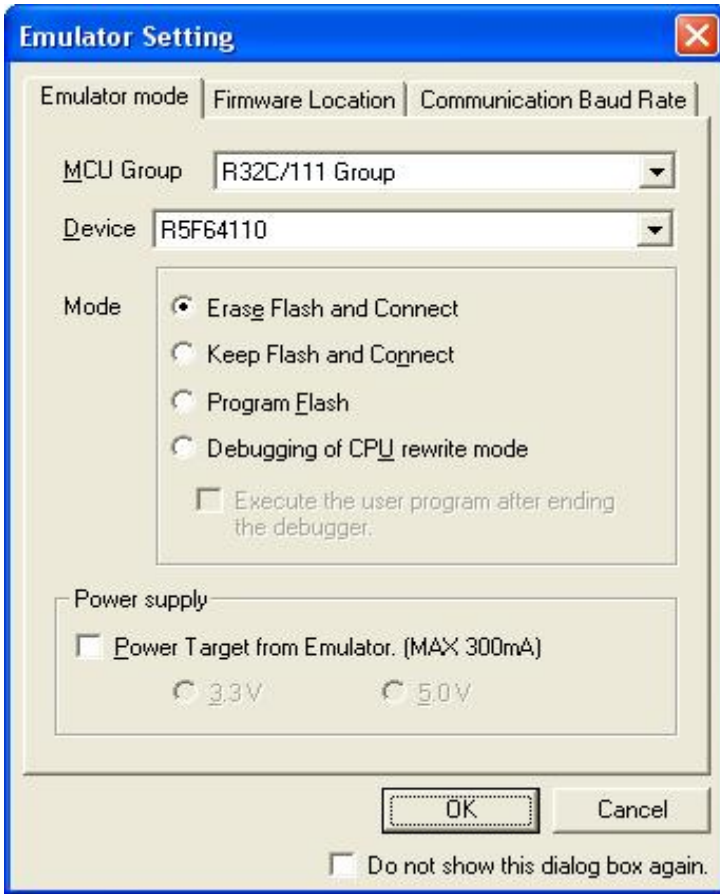


Figure 5.2 [Emulator mode] Tab of [Emulator Setting] Dialog Box

[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]

Select the mode to be used. For details, see “5.2 (1) Selecting the Mode” (p.27).

[Power supply]

Select the power supply to the user system.
- When supplying power to the user system from the E8a, click the [Power Target from Emulator. (MAX 300mA)] checkbox.

Note that when debugging the system which operates the MCU with a dual power supply, power cannot be supplied from the E8a.

(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 [*3]	<p>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.</p> <p>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.</p>
Debugging of CPU rewrite mode	Debugging only [*1]	<p>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.</p> <ul style="list-style-type: none"> - Setting the PC break points - Changing the memory contents in the Flash memory area <p>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.</p>

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 downloading the user program in this mode, a checksum is displayed. The checksum covers the E²dataFlash area. However, when ECC is used, the ECC area is not covered by the checksum.
- In addition, note that the E²dataFlash area is erased when starting up in this mode.

5.3 [Firmware Location] tab

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

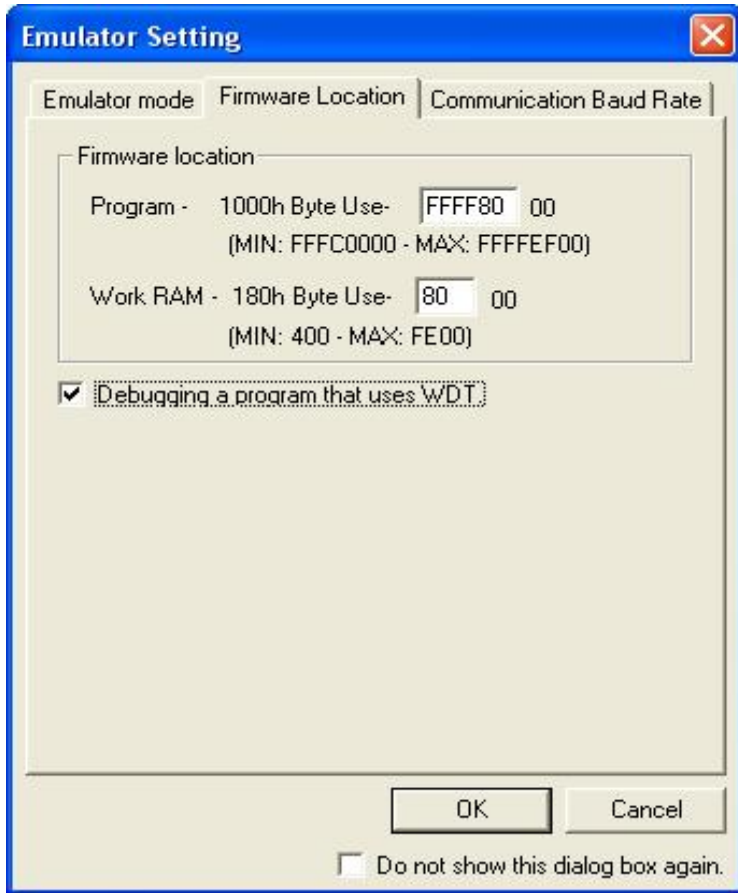


Figure 5.3 [Firmware Location] tab of [Emulator Setting] Dialog Box

[Firmware Location]
 Select the area in which the firmware is located. Specify the address that will not be used by the user system in the ROM area or RAM area.

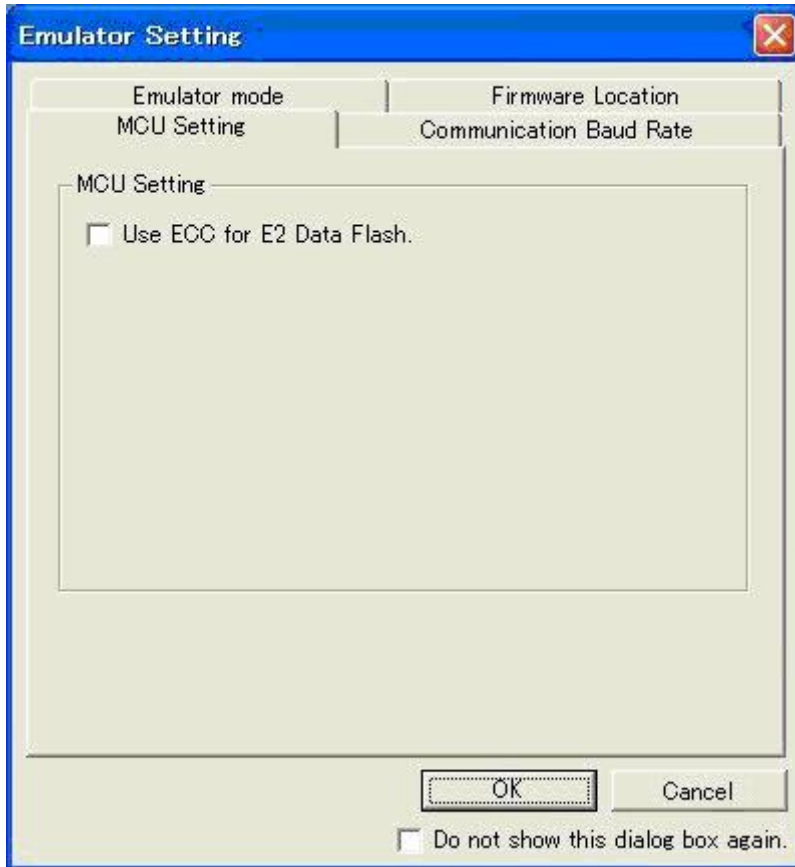
- Program
 Specify the ROM area in which the firmware is located. Specify 4 K bytes that will not be used by the user system. The data area cannot be specified.
- Work RAM
 Specify the RAM area in which the firmware is located. Specify 364 bytes that will not be used by the user system.

[Debugging of program that uses WDT]
 When debugging the user program using the watchdog timer, click this check box.

- Unchecked: WDT is not used.
 If the watchdog timer is enabled with this box unchecked during debugging, the E8a emulator will not operate normally.
- Checked: WDT is used.
 The E8a emulator program refreshes the watchdog timer during program operation. 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.

5.4 [MCU Setting] tab

For MCUs with E²dataFlash, the [MCU Setting] tab is displayed in the [Emulator Setting] dialog box. Be sure to set up the tab according to the setting of the E²dataFlash of the program to debug. For MCUs without E²dataFlash, the checkbox is gray and cannot be selected.



Use ECC for E2 Data Flash

Select this checkbox if you want to use ECC for an MCU with E²dataFlash. You cannot check this box if you selected an MCU without E²dataFlash. Be sure to make the setting according to the setting of the E²dataFlash of the program to debug. [*1]

Note

[*1] The block configuration of the E²dataFlash is determined by the setting of this checkbox. If the setting is made erroneously, the E²dataFlash area will be displayed incorrectly.

5.5 [Communication Baud Rate] tab

Select communication baud rate between the E8a and MCU in the [Communication Baud Rate] tab.

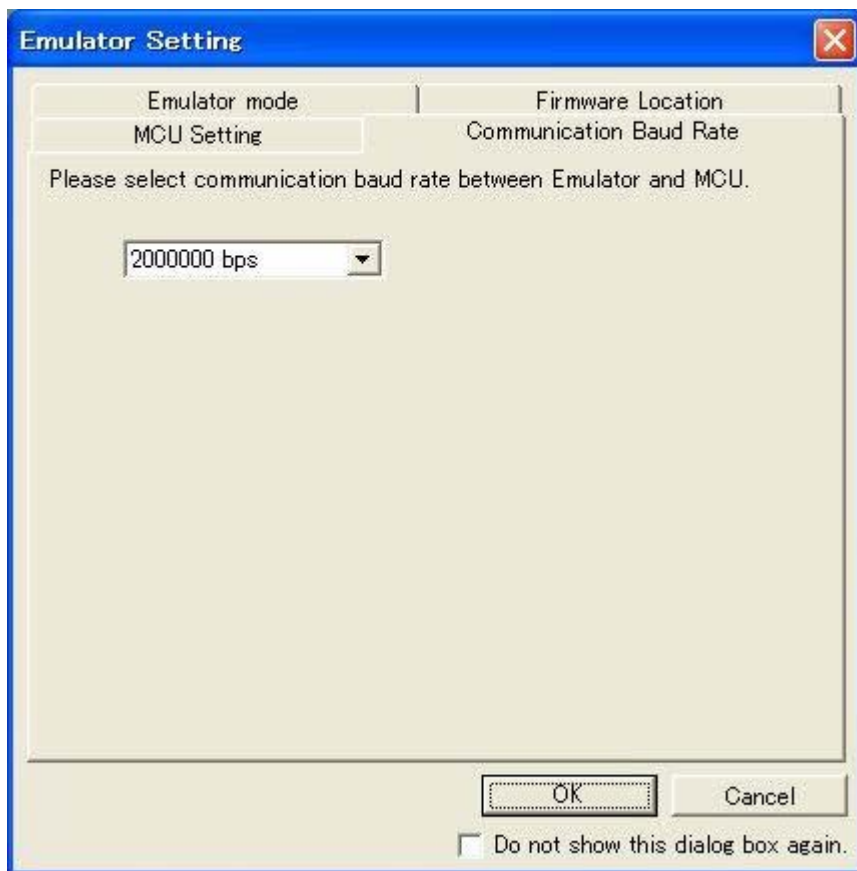


Figure 5.4 [Communication Baud Rate] Tab

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 area for the E8a emulator. Do not change this area, otherwise the E8a emulator will not control the MCU. In this case, disconnect the debugger and then reconnect it.

Table 6.1 Program Area for the E8a Emulator

Program Area for E8a Emulator		
Vector Area	ROM Area	RAM Area
FFFFFFFCh - FFFFFFFFh	4 KB of the Program Area [*1]	364 B [*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 (R32C/160 and R32C/161): RESET#, CNVss, P50, P55, P44, P45, P46, and P47 pins
- For debugging/programming (R32C/111 (64-pin and 80-pin)): RESET#, CNVss, P80, P81, P64, P65, P66, and P67 pins
- For debugging/programming (other than the above): RESET#, CNVss, P50, P55, P64, P65, P66, and P67 pins

(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	R0 to R7 (bank 0, 1)	0000h
	A0 to A3 (bank 0, 1)	00000000h
	FB (bank 0, 1)	00000000h
	SB (bank 0, 1)	00000000h
	PC	Reset vector value in the vector address table
	INTB (bank 0, 1)	00000000h
	USP	00000000h
	ISP	Work RAM Address for the E8a emulator + 180h [*1]
	SVF	00000000h
	SVP	00000000h
	VCT	00000000h
	DMD0 to DMD3	XXXX XXXX XXXX XXXX XXXX XXXX XX00 0000b
	DCT0 to DCT3	000000h
	DCR0 to DCR3	000000h
	DSA0 to DSA3	00000000h
	DSR0 to DSR3	00000000h
	DDA0 to DDA3	00000000h
	DDR0 to DDR3	00000000h

Note

[*1] The Work RAM address for the E8a emulator is specified in the [Firmware Location] tab of the [Emulator Setting] dialog box.

(4) SFRs used by the E8a emulator program

The SFRs listed in Tables 6.3 and 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 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.

The SFR listed in Table 6.5 cannot be used by the user program when the E8a is running.

Table 6.3 SFRs Used by the E8a Emulator Program (R32C/160 and R32C/161)

Address	Register	Symbol	Bit	Notes on Using the E8a Emulator
02E8h	UART1 transmit/receive mode register	U1MR	All bits	[*1]
02EAh, 02EBh	UART1 transmit buffer register	U1TB	All bits	[*1]
02ECh	UART1 transmit/receive control register 0	U1C0	All bits	[*1]
02EDh	UART1 transmit/receive control register 1	U1C1	All bits	[*1]
02EEh, 02EFh	UART1 receive buffer register	U1RB	All bits	[*1]
03C8h	Port P4 register	P4	Bits 4, 5, 6 and 7	[*2]
03CAh	Port P4 direction register	PD4	Bits 4, 5, 6 and 7	[*2]
400CEh	Port P4_7 port function select register	P4_7S	All bits	[*1]

Table 6.4 SFRs Used by the E8a Emulator Program (MCUs other than R32C/160 and R32C/161)

Address	Register	Symbol	Bit	Notes on Using the E8a Emulator
02E8h	UART1 transmit/receive mode register	U1MR	All bits	[*1]
02EAh, 02EBh	UART1 transmit buffer register	U1TB	All bits	[*1]
02ECh	UART1 transmit/receive control register 0	U1C0	All bits	[*1]
02EDh	UART1 transmit/receive control register 1	U1C1	All bits	[*1]
02EEh, 02EFh	UART1 receive buffer register	U1RB	All bits	[*1]
03CCh	Port P6 register	P6	Bits 4, 5, 6 and 7	[*2]
03CEh	Port P6 direction register	PD6	Bits 4, 5, 6 and 7	[*2]
400DEh	Port P6_7 port function select register	P6_7S	All bits	[*1]

Table 6.5 SFR for which the Value of the Bit cannot be Changed when the E8a is Running

Address	Register	Symbol	Bit	Notes on Using the E8a Emulator
40047h	System clock control register 1	CM1	Bit 0	[*3]

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.
- [*3] When the E8a is running, the value of the bit is fixed to "0". (PLL Clock Oscillator Stop is fixed to 0: PLL clock oscillator running.)
- [*4] UART1 transmit interrupt control register S1TIC and UART1 receive interrupt control register S1RIC always read out values used by the emulator.

(5) Stack area used by the E8a emulator

The E8a emulator uses up to 32 bytes of the stack pointer (ISP) during a user program break. Therefore, set aside 32 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 use the software reset and watchdog timer 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 BRK2 instruction interrupt and single-step 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) Interrupts used by the E8a emulator program (NMI)

If NMI interrupts are used, be sure to take the necessary precautions before executing the user program like disabling 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. If an NMI interrupt occurs while the user program halts or when memory contents are referenced or modified during user program execution, the E8a emulator cannot control the MCU.

(9) DMACII transfer complete interrupt

If DMACII transfer complete interrupts are used, be sure to take the necessary precautions before executing the user program like disabling 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. If a DMACII transfer complete interrupt occurs while the user program halts or when memory contents are referenced or modified during user program execution, the E8a emulator cannot control the MCU.

(10) 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.

6.2 Flash memory

6.2.1 Note 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 following area. If these areas are rewritten, the E8a emulator will not control the MCU.

- Block 0 area (addresses FFFF8000h - FFFFFFFFh) and block containing the E8a emulator program

(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.
- 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.

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 6.6 written to the flash memory of the MCU must match the ID code displayed in the 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 6.6 ID Code Storage Area

Address	Description
FFFFFFE8h	First byte of ID code
FFFFFFE9h	Second byte of ID code
FFFFFFEAh	Third byte of ID code
FFFFFFEBh	Fourth byte of ID code
FFFFFFECh	Fifth byte of ID code
FFFFFFEDh	Sixth byte of ID code
FFFFFFEEh	Seventh byte of ID code

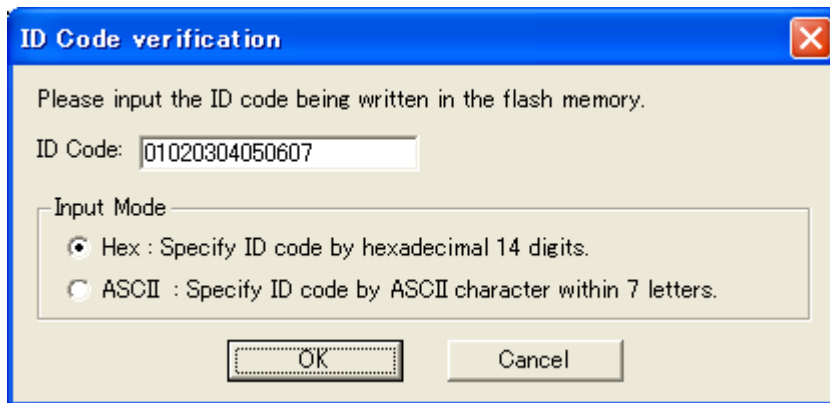


Figure 6.1 [ID Code verification] Dialog Box

Notes

Notes on 'Program Flash' mode:

- When the ID code is specified by the -ID option of the lmc100, 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 Count source protect mode

When downloading the program that enables the count source protect mode, the E8a emulator sets both of the bit 2 and bit 3 of the optional function select area (OFS: FFFFFFFEh) to 0b.

b3, b2: Watchdog Timer Prescaler Select Bit 0 0 : Divide-by-8 (WDK3 to WDK2 = 00b)

Applicable MCUs:

R32C/120, R32C/121, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161

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.

Note that when debugging the system which operates the MCU with a dual power supply, power cannot be supplied from the E8a.

6.5 Operation during a user program halt

(1) 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.

6.6 Functions of the E²dataFlash

- (1) Program download

Program download is possible as well as user program download.
- (2) Memory access to the E²dataFlash area (during user program halt)

Memory access to the E²dataFlash area is possible as well as the ROM and RAM.

When the memory is accessed from the memory window, read/write is performed according to the mode selected with the [Use ECC for E2 Data Flash] checkbox in the [MCU Setting] tab (see 5.4 [MCU Setting] tab) in the init dialog box which appears at debugger startup. If the setting of this checkbox differs from that of the ECC Control Bit (ECC enabled /ECC disabled) in the user program, memory reference/modification cannot be performed correctly.
- (3) Memory access to the E²dataFlash area (during user program execution)

Memory access to the E²dataFlash is not possible. If the E²dataFlash area is displayed in the memory or other windows, “1” will be displayed for all the bits of the displayed area. Do not stop the user program when it is processing the access to the E²dataFlash area. If the user program stops, the operation to the E²dataFlash is suspended, and the access to the E²dataFlash may not be processed properly even after the user program is restarted. Disable the automatic update in the windows before running the user program so access to the E²dataFlash does not occur during an execution.
- (4) Other

PC break points cannot be set in the E²dataFlash area.

6.7 Debug functions

- (1) Memory access during user program 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) Setting of address match break during user program execution

When adding or cancelling the address match break, the user program is temporarily halted. For this reason, a real-time emulation cannot be performed.
- (3) 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.

If a low-speed clock such as the sub clock is used as the operation clock of the MCU, setting or canceling PC breaks may take time. Use event breaks as the first choice.
- (4) “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.
- (5) 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.
- (6) Low power consumption mode

When debugging in low power consumption 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 low power consumption mod is cancelled.

(7) DMAC and DMACII during a user program halt

When the user program is halted or when the memory is referred to or modified during user program execution, DMA transfer is disabled. In such cases, the E8a emulator sets the registers below as following. Therefore, if you refer to the registers below in the memory window, etc., it shows that DMA is disabled.

- DMA0 Mode Register (DMD0)		
Transfer mode select bit (bit 1, 0)	00:	DMA transfer disabled
- DMA1 Mode Register (DMD1)		
Transfer mode select bit (bit 1, 0)	00:	DMA transfer disabled
- DMA2 Mode Register (DMD2)		
Transfer mode select bit (bit 1, 0)	00:	DMA transfer disabled
- DMA3 Mode Register (DMD3)		
Transfer mode select bit (bit 1, 0)	00:	DMA transfer disabled
- Interrupt Control Register		
Interrupt request level select bit (bit 2, 1, 0)	000:	Level 0 (interrupt disabled)
- Interrupt Control Register		
Interrupt request bit (bit 3)	0:	Interrupt not requested [*1]

Do not enable DMA transfer from the memory window, etc., but enable it in the user program.

Note

[*1] When restarting the user program, though the E8a emulator sets back the value of a DMA mode register to the previous value that was set before the program stops, the interrupt request bit remains 0.

(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).

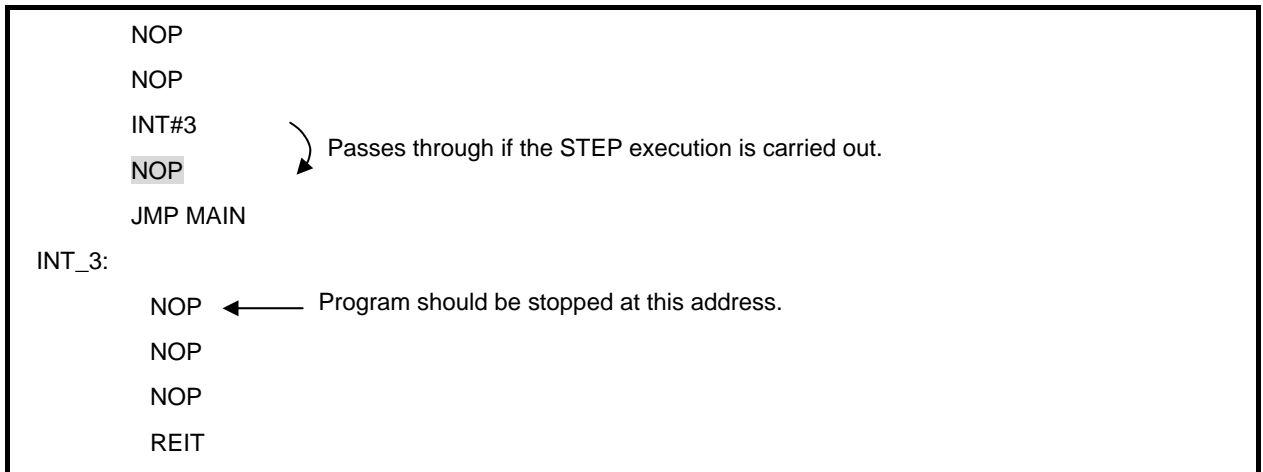


Figure 6.2 Example of Software Interrupt Instruction

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).

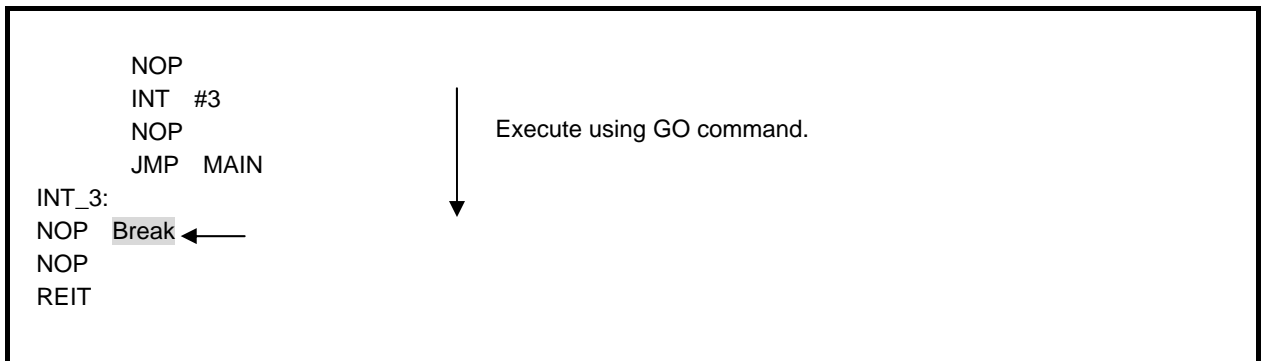


Figure 6.3 Example of INT Instruction

(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.

E8a Emulator (R0E00008AKCE00)

Additional Document for User's Manual

Notes on Connecting the R32C/111, R32C/116, R32C/116A,
R32C/117, R32C/117A, R32C/118, R32C/118A, R32C/120,
R32C/121, R32C/145, R32C/151, R32C/152,
R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161

Publication Date: Apr 08, 2011 Rev.7.00

Published by: Renesas Electronics Corporation

Edited by: Microcomputer Tool Development Department 2
Renesas Solutions Corp.



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