REJ10J1877-0201

Everywhere you imagine. RENESAS

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/117, R32C/118, R32C/120, R32C/121, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161

User's Manual

Rev.2.01
Jun. 26, 2009

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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 R32C/100 Series.

Table 2.1 E8a Emulator Specifications for the R32C/100 Series

Target MCUs		M16C Family R32C/100 Series R32C/111 (100-pin), R32C/116, R32C/117, R32C/118, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161 Groups (R32C/120 and R32C/121 Groups [*1])
Available operating modes		Single-chip mode, Memory expansion mode * Microprocessor mode is not supported.
Power voltages		3.0 - 5.5V
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: 3 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/117, R32C/118, R32C/120, R32C/121, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157	- ROM size: 3 KB - RAM size: 364 bytes - Stack 32 bytes - Address match interrupt - Pins P50 and P55 [*2] - 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.1.00.01 or later

Notes

[*1] For these MCUs, the operations have been verified with the working sample MCUs only.

[*2] For 64-pin and 80-pin version 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

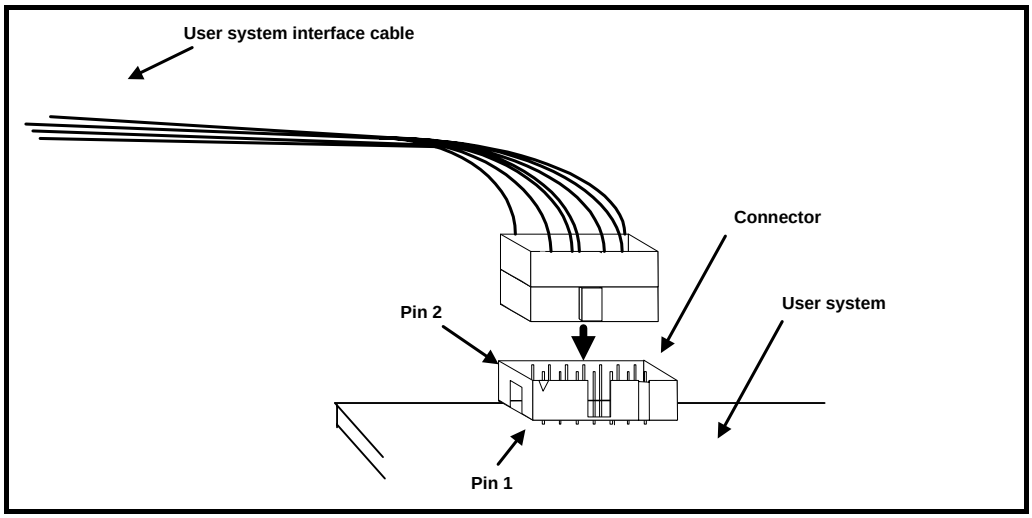
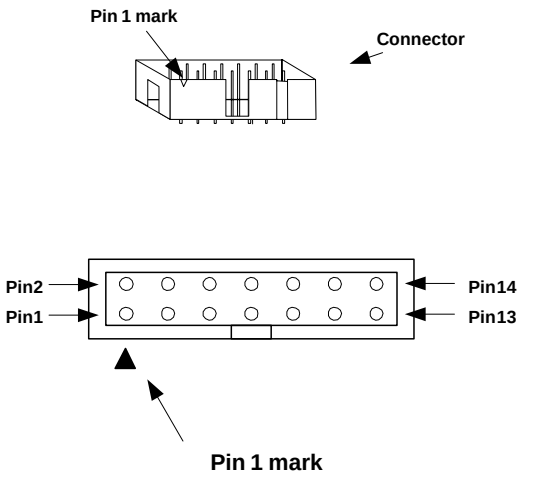


Figure 3.1 Connecting the User System Interface Cable with an E8a Connecting Connector

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.



Pin No.	R32C/111 (100-pin), R32C/116, R32C/117, R32C/118, R32C/120, R32C/121, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157 MCU Signals	R32C/111 (64-pin or 80-pin)	R32C/160, R32C/161 MCU Signals
1	P65(SCLK)	P65(SCLK)	P45(SCLK)
2	Vss	Vss	Vss
3	CNVss	CNVss	CNVss
4	P55(EPM)	P81(EPM)	P55(EPM)
5	P67(TxD)	P67(TxD)	P47(TxD)
6	Vss	Vss	Vss
7	P50(CE)	P80(CE)	P50(CE)
8	Vcc	Vcc	Vcc
9	P64(BUSY)	P64(BUSY)	P44(BUSY)
10	Vss	Vss	Vss
11	P66(RxD)	P66(RxD)	P46(RxD)
12	Vss	Vss	Vss
13	RESET#	RESET#	RESET#
14	Vss	Vss	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.

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/117, R32C/118, R32C/120, R32C/121, 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.

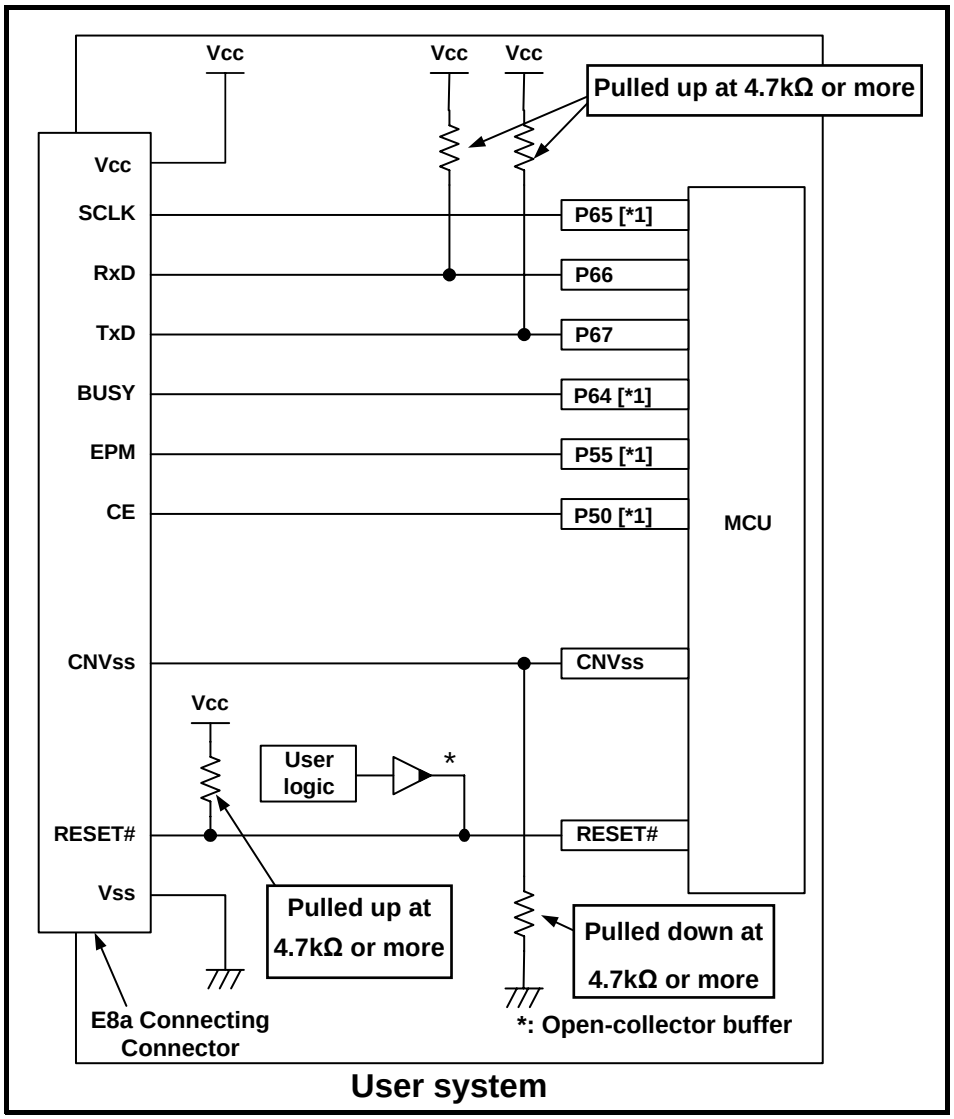


Figure 4.1 Example of an E8a Connection
(Single Power Supply and Single-chip Mode, MCUs Other Than R32C/160 and R32C/161)

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.

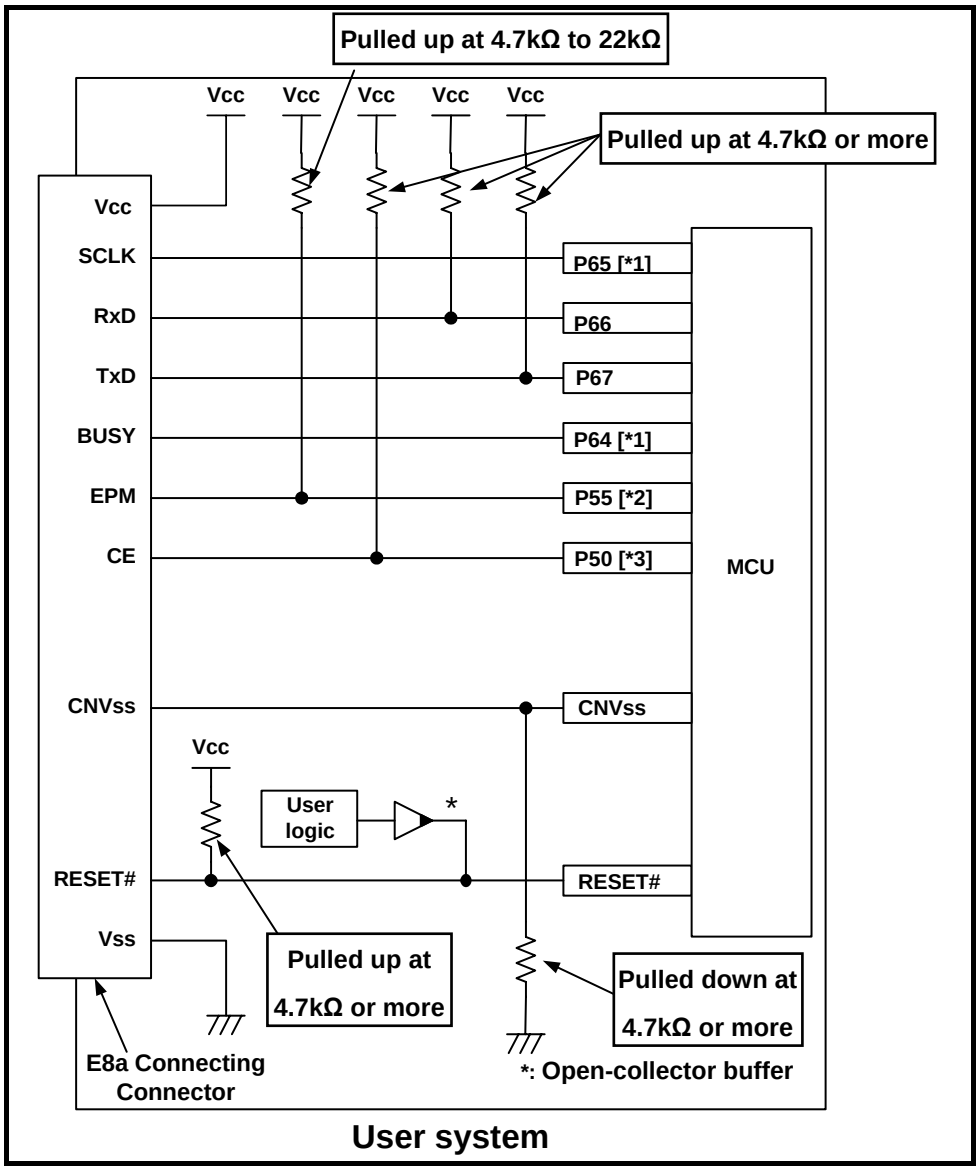


Figure 4.2 Example of an E8a Connection
(Single Power Supply and Memory Expansion Mode, MCUs Other Than R32C/160 and R32C/161)

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.

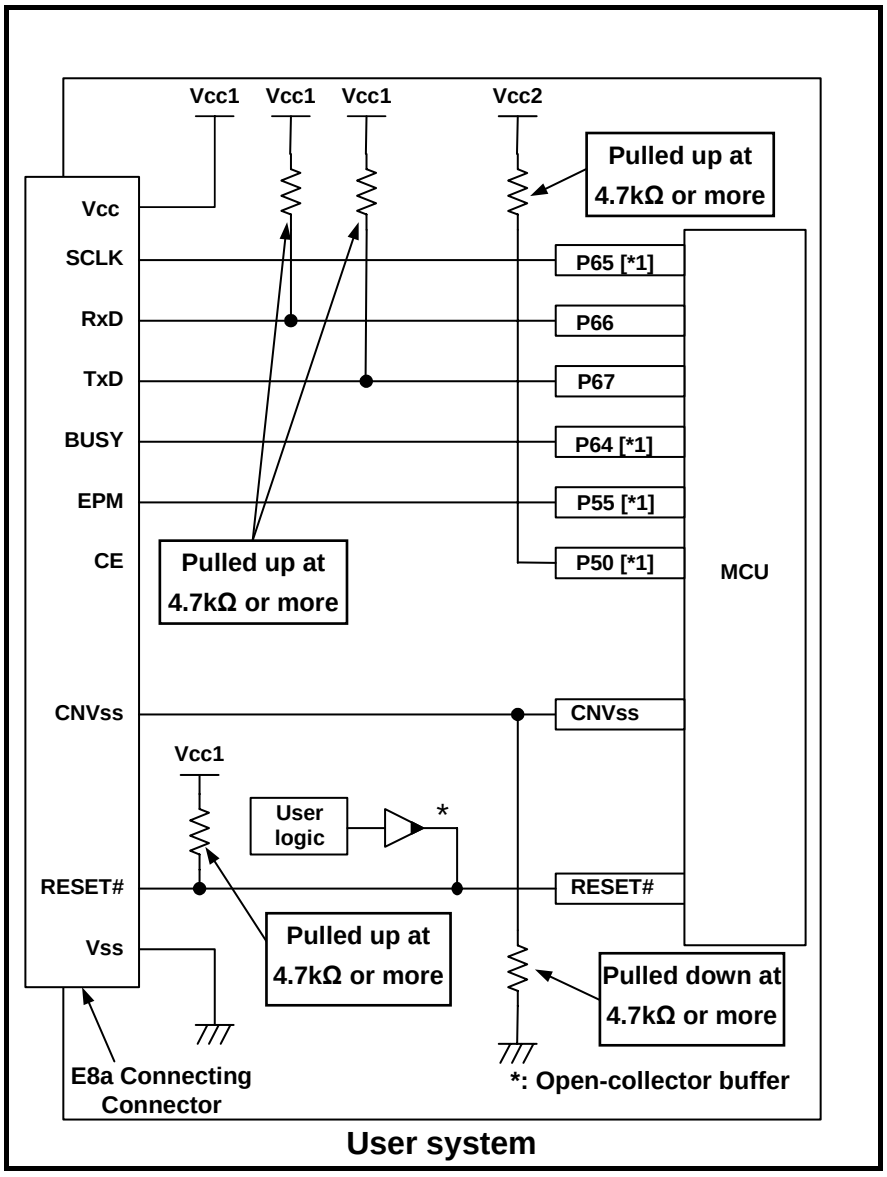


Figure 4.3 Example of an E8a Connection
(Dual Power Supply and Single-chip Mode, R32C/111 (100-pin) Only)

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.

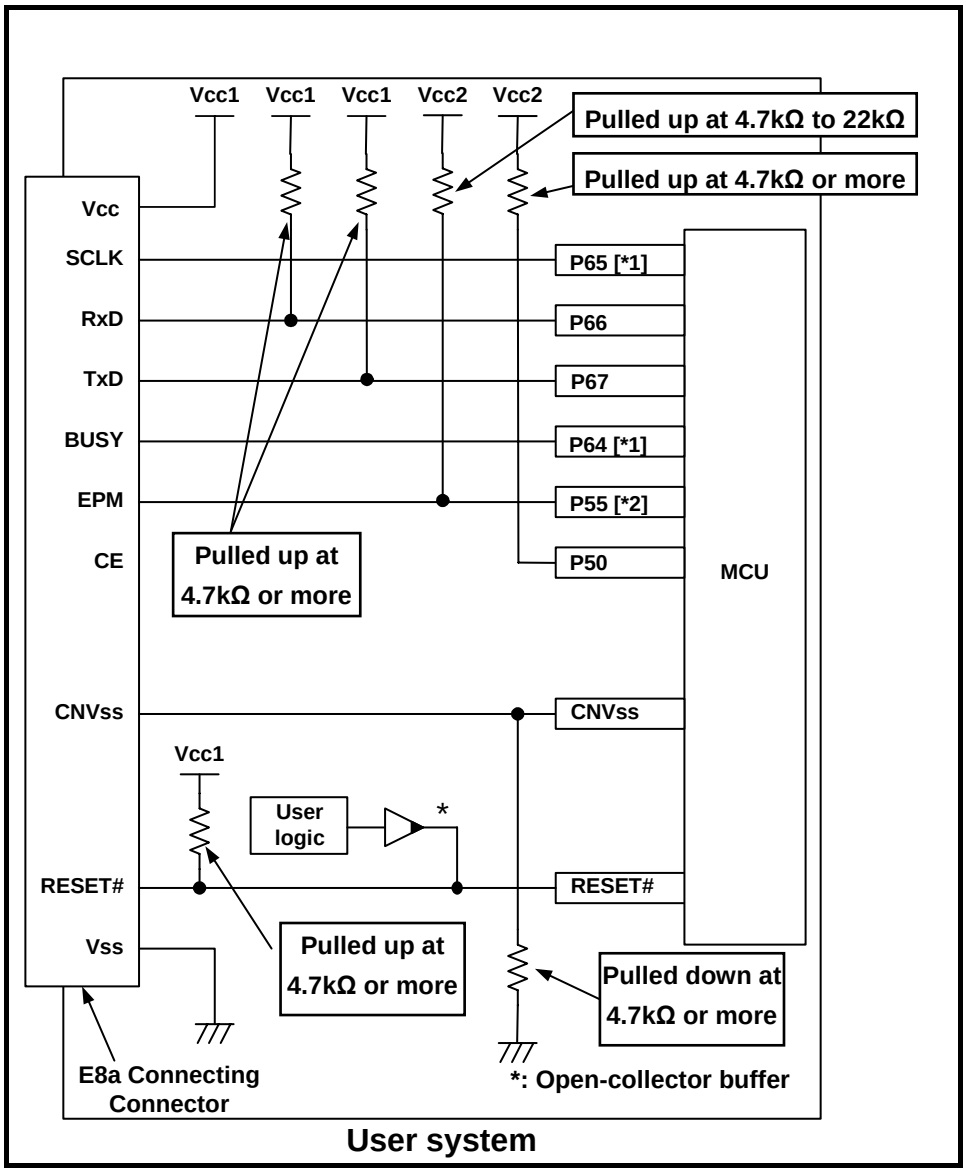


Figure 4.4 Example of an E8a Connection
(Dual Power Supply and Memory Expansion Mode, R32C/111 (100-pin) Only)

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 4.24 on page 26).

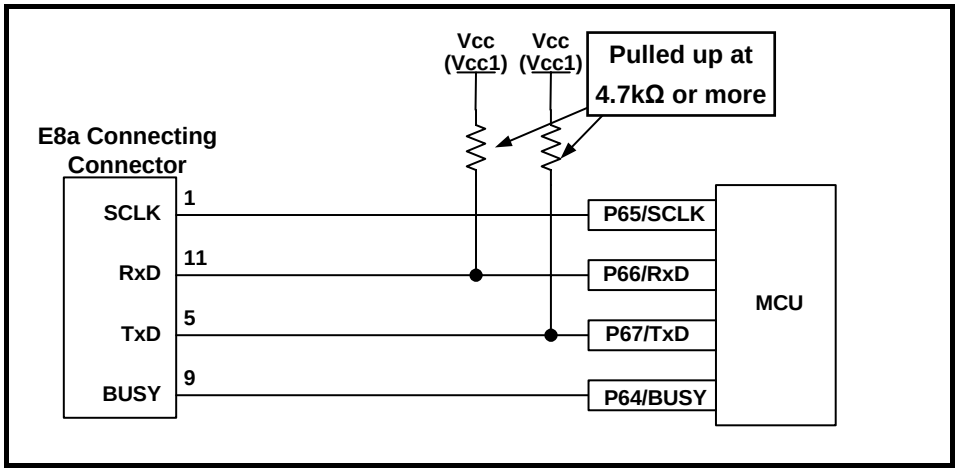


Figure 4.5 E8a Emulator and MCU Connection (MCUs Other Than R32C/160 and R32C/161)

(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 4.24 on page 26).

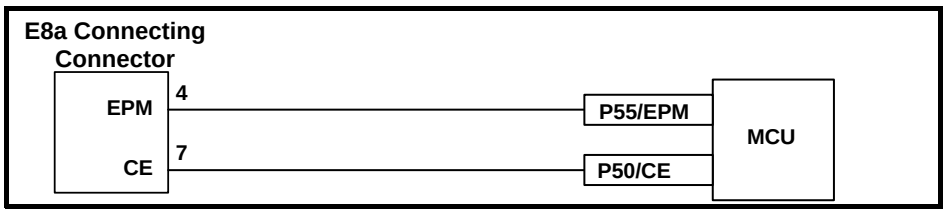


Figure 4.6 Connection of E8a Emulator and Pins P50 and P55
(Single Power Supply and Single-chip Mode, MCUs Other Than R32C/160 and R32C/161)

2. Single power supply and memory expansion mode

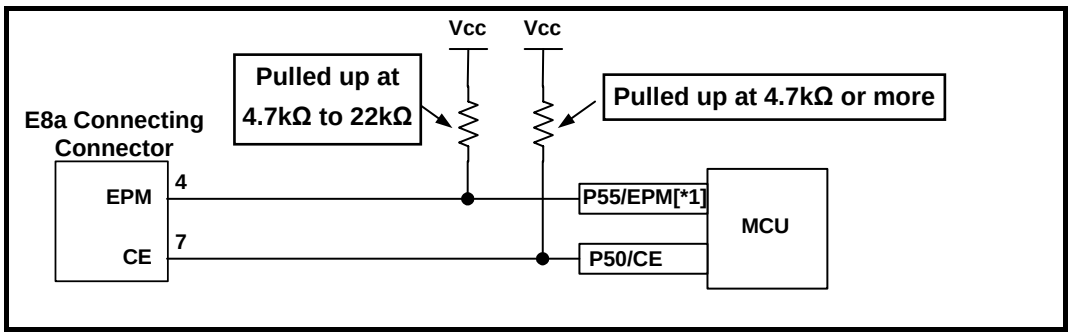


Figure 4.7 Connection of E8a Emulator and Pins P50 and P55
(Single Power Supply and Memory Expansion Mode, MCUs Other Than R32C/160 and R32C/161)

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 4.24 on page 26).

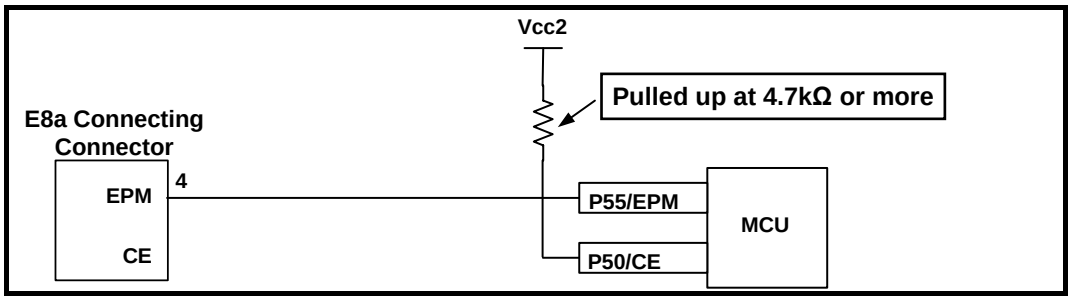


Figure 4.8 Connection of E8a Emulator and Pins P50 and P55
(Dual Power Supply and Single-chip Mode, R32C/111 (100-pin) Only)

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

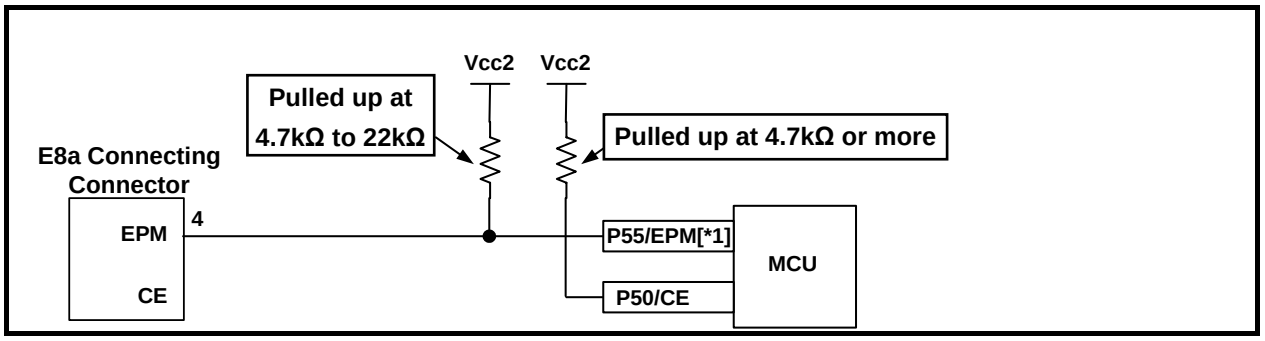


Figure 4.9 Connection of E8a Emulator and Pins P50 and P55
(Dual Power Supply and Memory Expansion Mode, R32C/111 (100-pin) Only)

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.

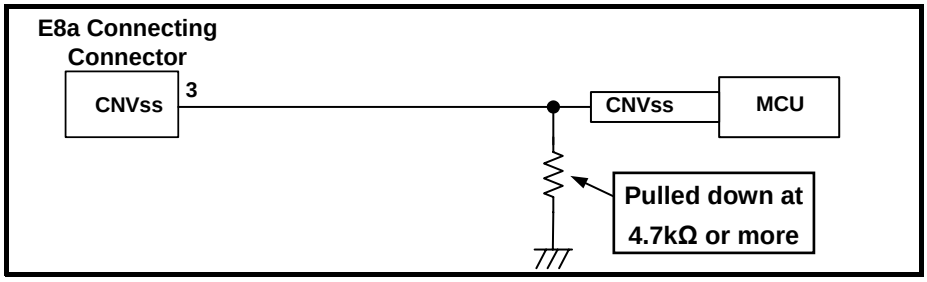


Figure 4.10 E8a Emulator and CNVss Pin Connection

(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.

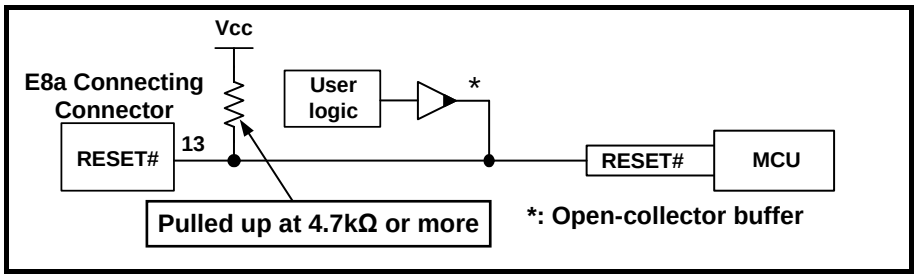


Figure 4.11 Example of a Reset Circuit

(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.

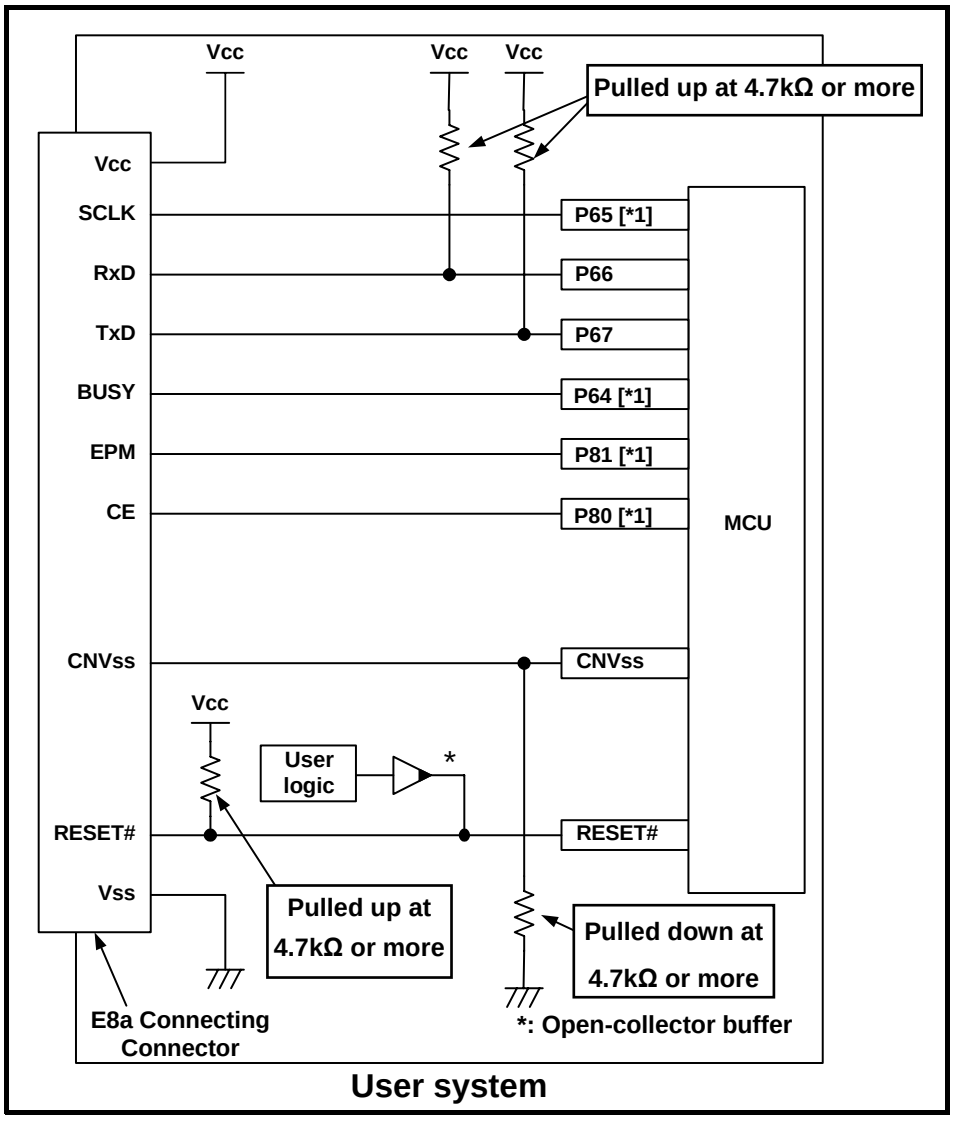


Figure 4.12 Example of an E8a Connection
(Single Power Supply and Single-chip Mode, R32C/111 (64-pin and 80-pin) Only)

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 4.24 on page 26).

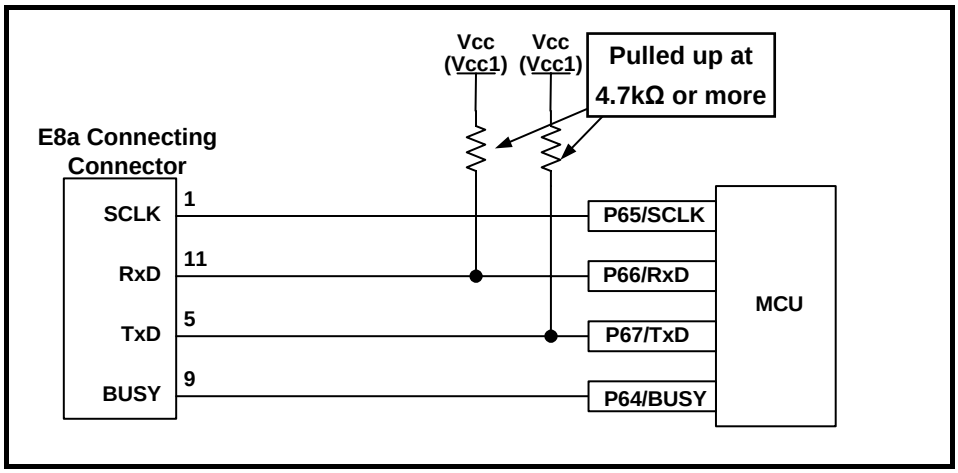


Figure 4.13 E8a Emulator and MCU Connection (R32C/111 (64-pin and 80-pin) Only)

(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 4.24 on page 26).

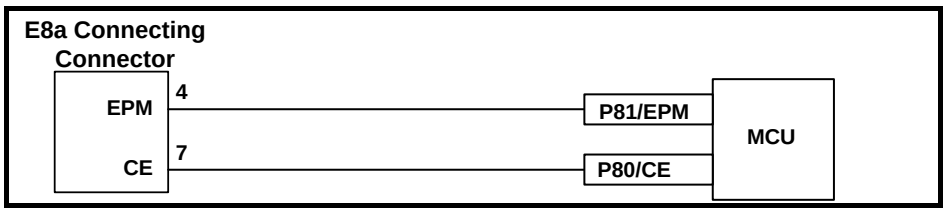


Figure 4.14 Connection of E8a Emulator and Pins P80 and P81 (R32C/111 (64-pin and 80-pin) Only)

(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.

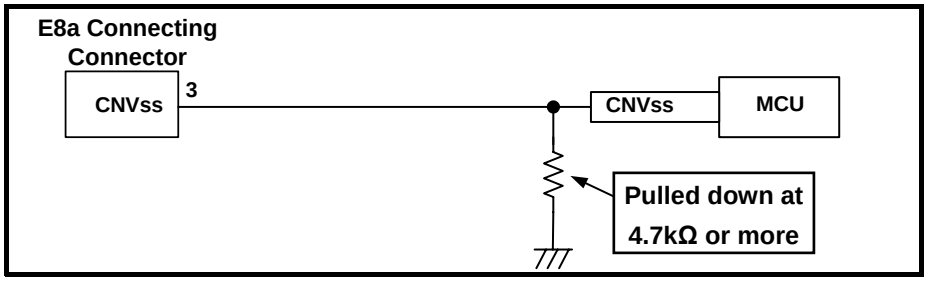


Figure 4.15 E8a Emulator and CNVss Pin Connection

(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.

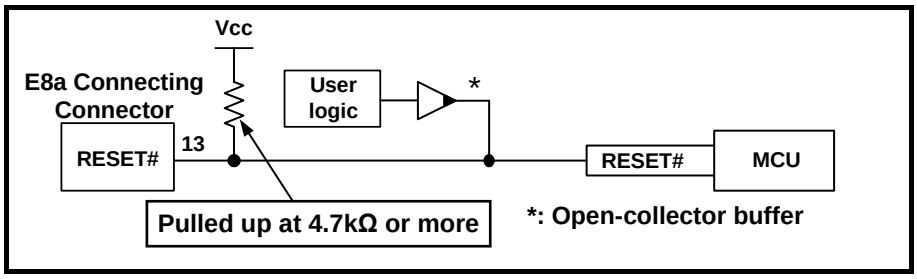


Figure 4.16 Example of a Reset Circuit

(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.
- Memory expansion mode: See Figure 4.18.

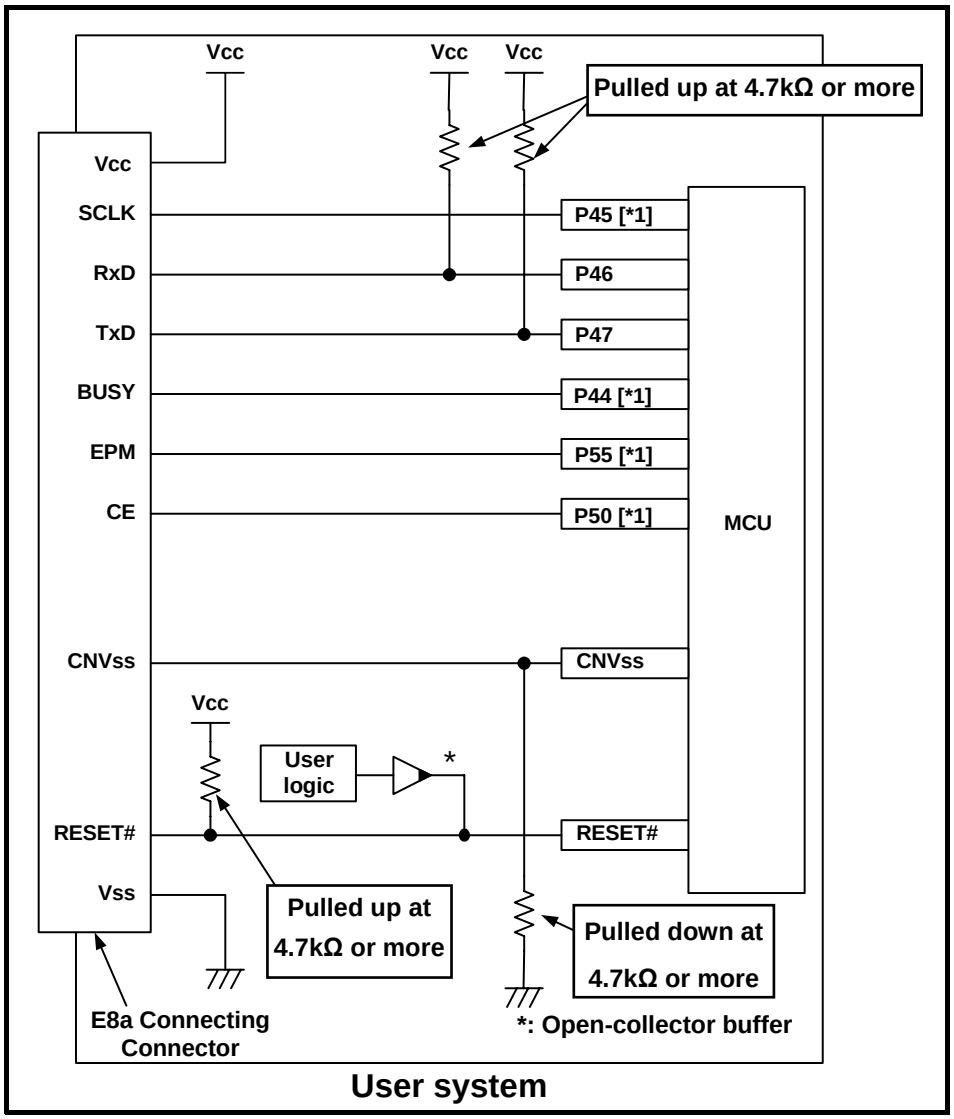


Figure 4.17 Example of an E8a Connection
(Single-chip Mode, R32C/160 and R32C/161 Only)

Note

[*1] For details on setting pins P44 and P45, refer to “4.3.1 (1) SCLK, RxD, TxD and BUSY pins” on page 23.
For details on setting pins P50 and P55, refer to “4.3.1 (2) EPM# and CE# pins” on page 24.

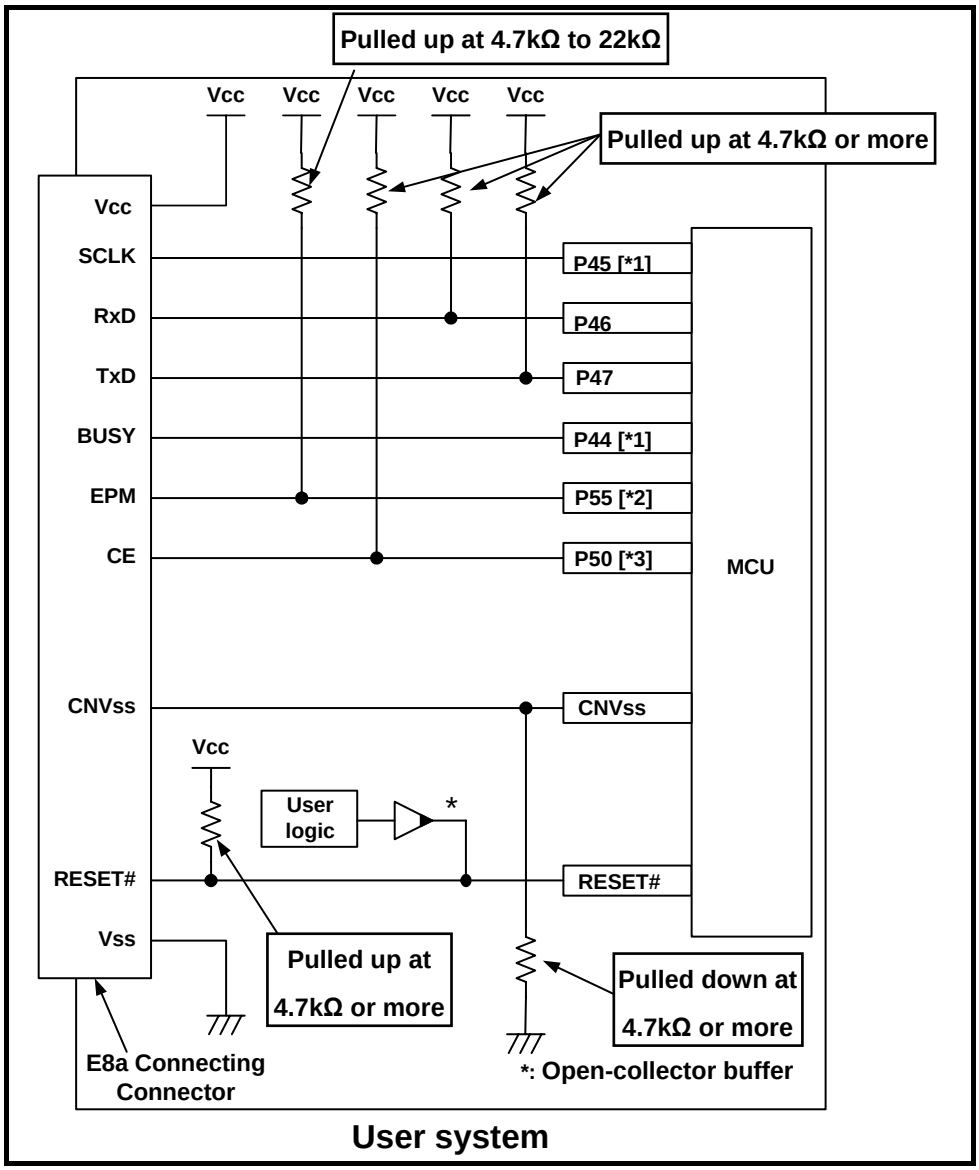


Figure 4.18 Example of an E8a Connection
(Memory Expansion Mode, R32C/160 and R32C/161 Only)

Notes

[*1] For details on setting pins P44 and P45, refer to “4.3.1 (1) SCLK, RxD, TxD and BUSY pins” on page 23.

[*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.

(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 4.24 on page 26).

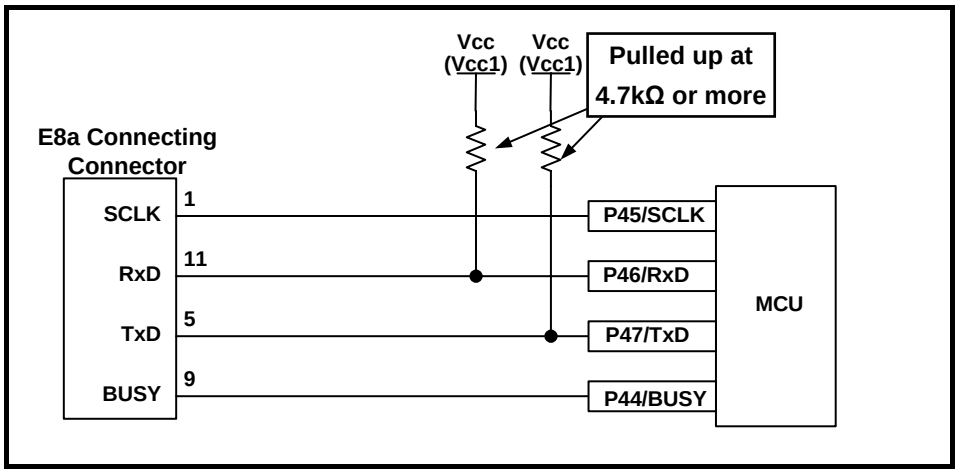


Figure 4.19 E8a Emulator and MCU Connection (R32C/160 and R32C/161 Only)

(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 4.24 on page 26).

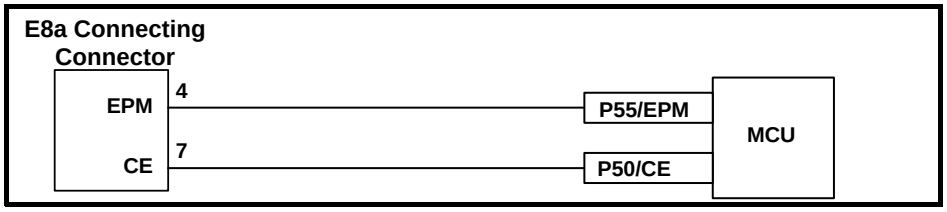


Figure 4.20 Connection of E8a Emulator and Pins P50 and P55
(Single-chip Mode, R32C/160 and R32C/161 Only)

2. Memory expansion mode

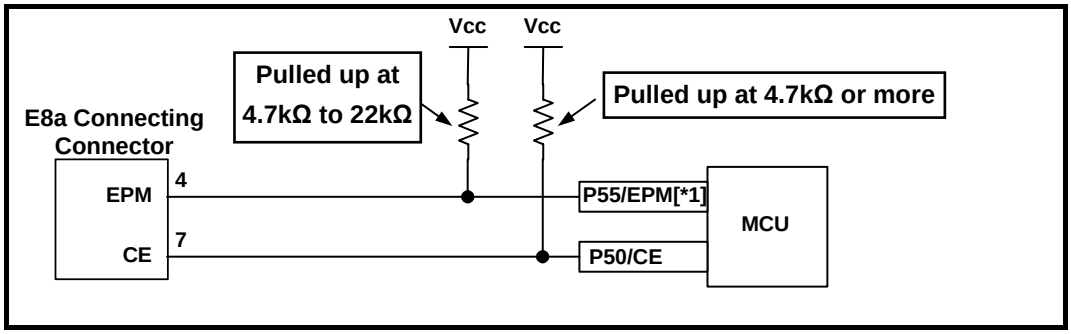


Figure 4.21 Connection of E8a Emulator and Pins P50 and P55
(Memory Expansion Mode, R32C/160 and R32C/161 Only)

Note

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

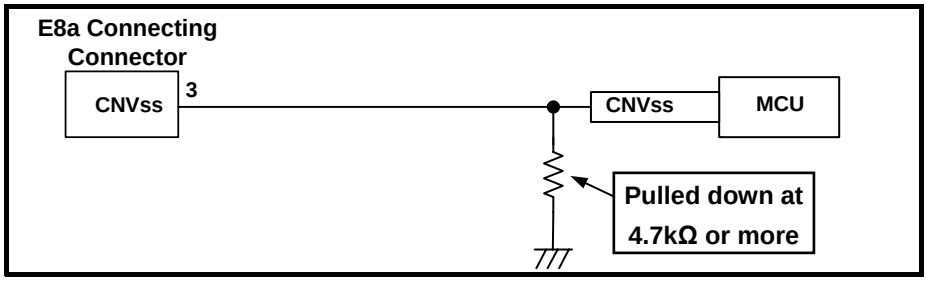


Figure 4.22 E8a Emulator and CNVss Pin Connection

(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.

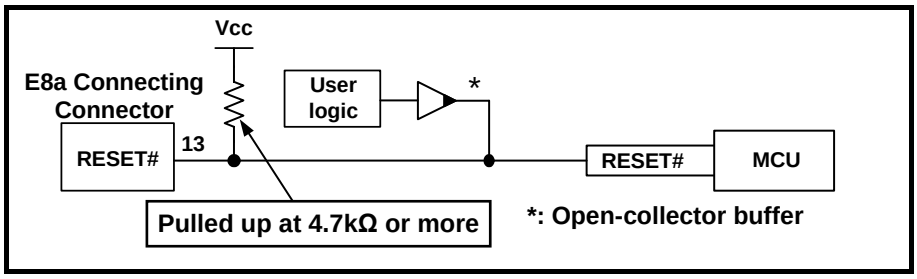


Figure 4.23 Example of a Reset Circuit

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

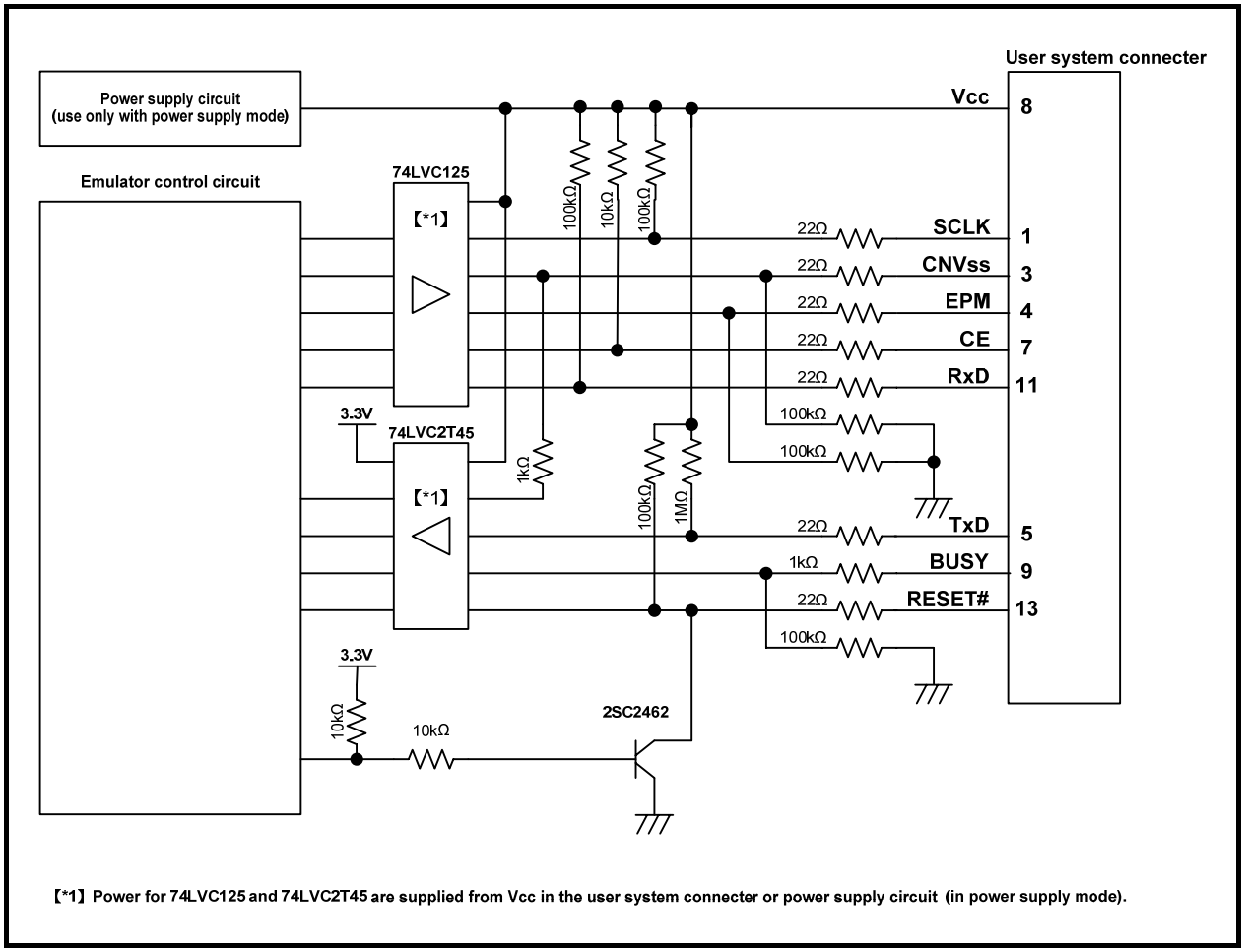


Figure 4.24 Interface Circuit inside the E8a Emulator (For Reference)

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.

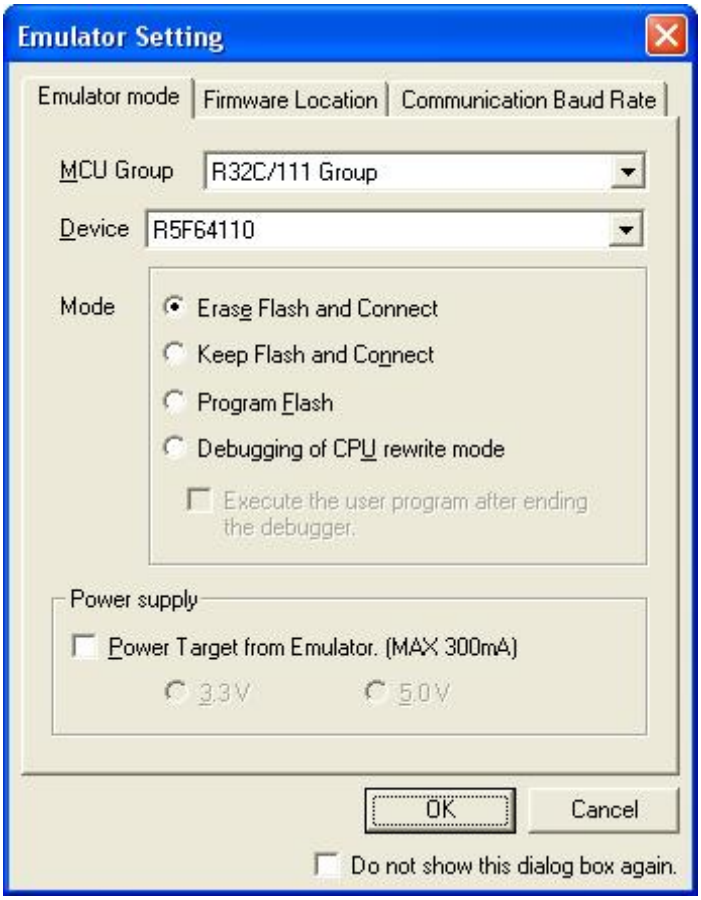


Figure 5.1 [Emulator Setting] Dialog Box

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.

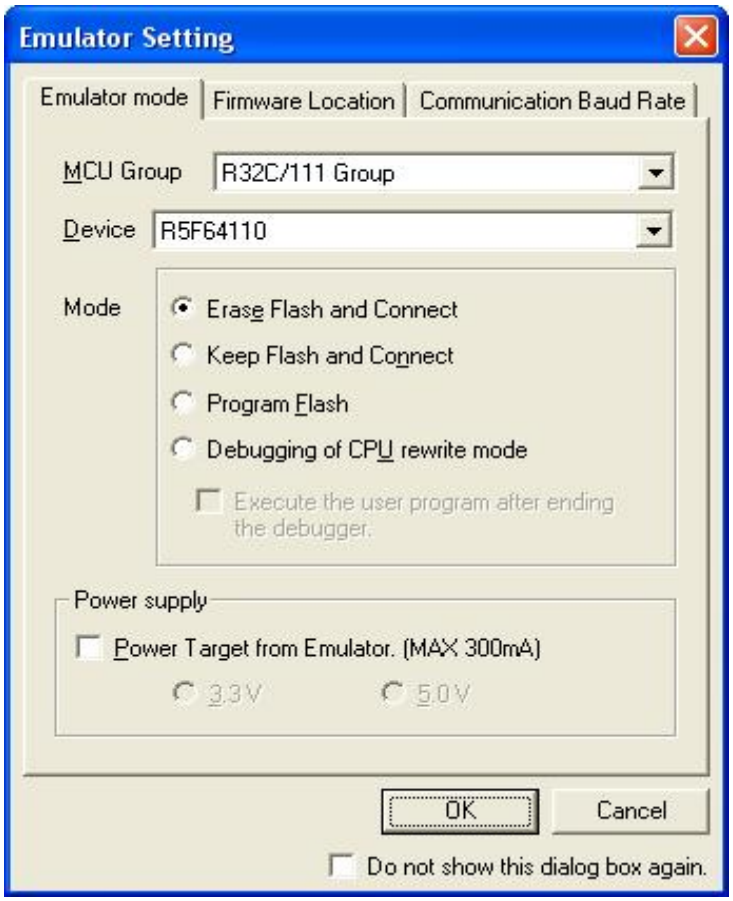
You can open the [Emulator Setting] dialog box using one of the following methods:

- 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.



[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.29).

[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.

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	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.

5.3 [Firmware Location] tab

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

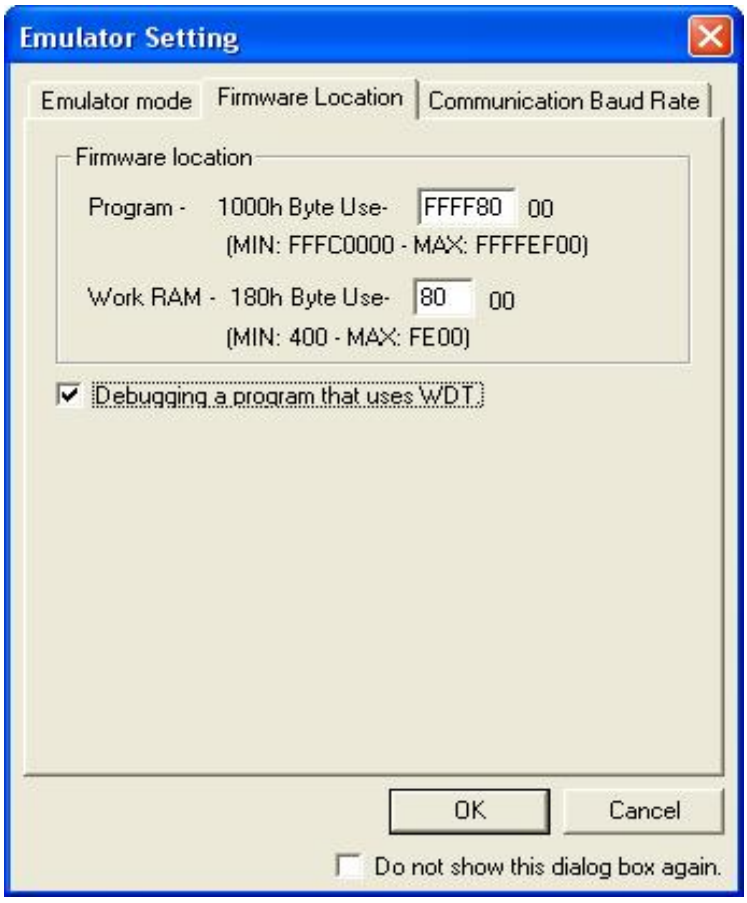


Figure 5.3 [Firmware Location] tab of [Emulator Setting] Dialog Box

[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 3K bytes that will not be used by the user system.
- 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 [Communication Baud Rate] tab

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

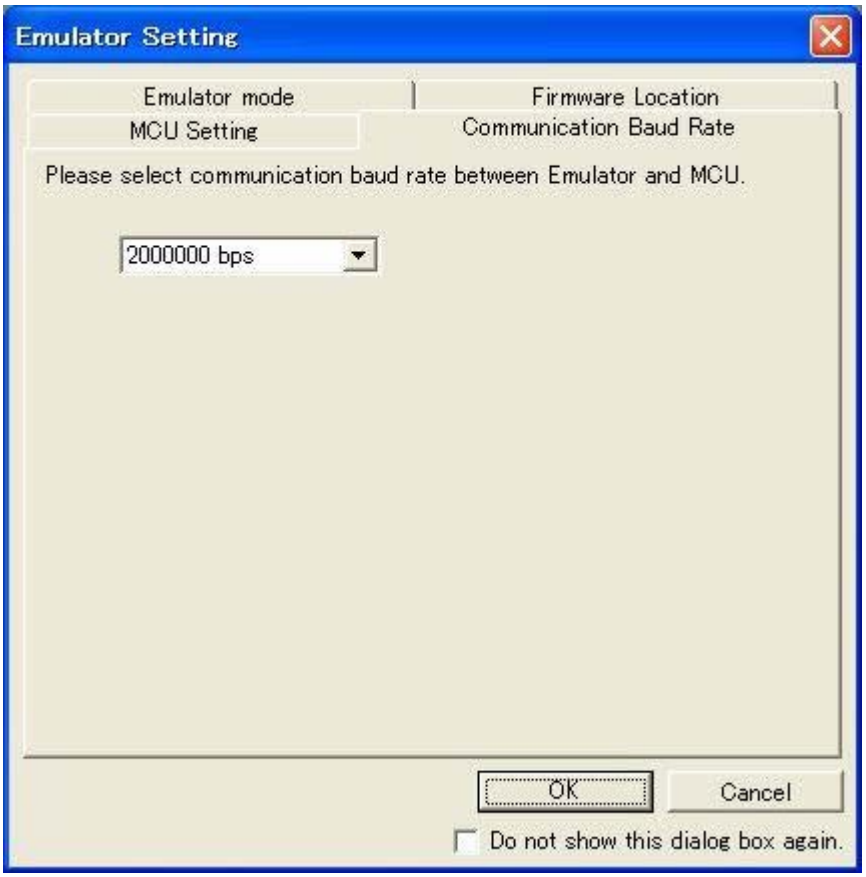


Figure 5.4 [Communication Baud Rate] Tab

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 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

Program Area for E8a Emulator		
Vector Area	ROM Area	RAM Area
FFFFFFFCh - FFFFFFFFh	3 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.

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]

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] 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.
- 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 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 6.5 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
FFFFFFFCh	Fifth byte of ID code
FFFFFFFDh	Sixth byte of ID code
FFFFFFFEh	Seventh byte of ID code

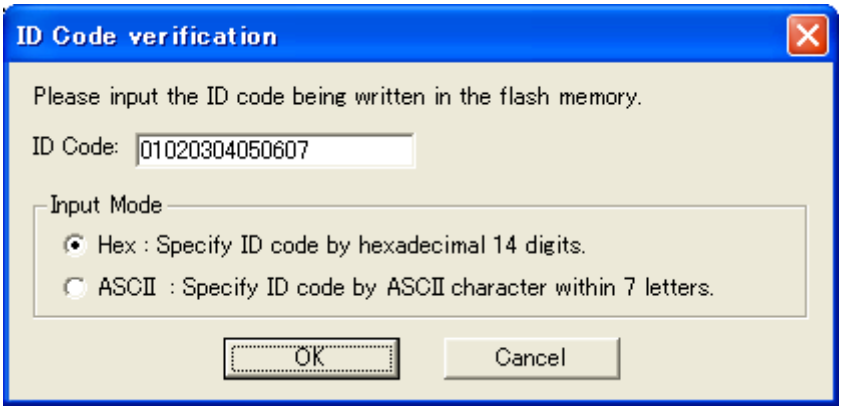


Figure 6.1 [ID Code verification] Dialog Box

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: FFFFFFEFh) 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 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) Debugging the E^2PROM emulation data flash area

This software does not support the following function.

- Reference and setting of E^2PROM emulation data flash area by emulator commands

(8) 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.

(9) 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).

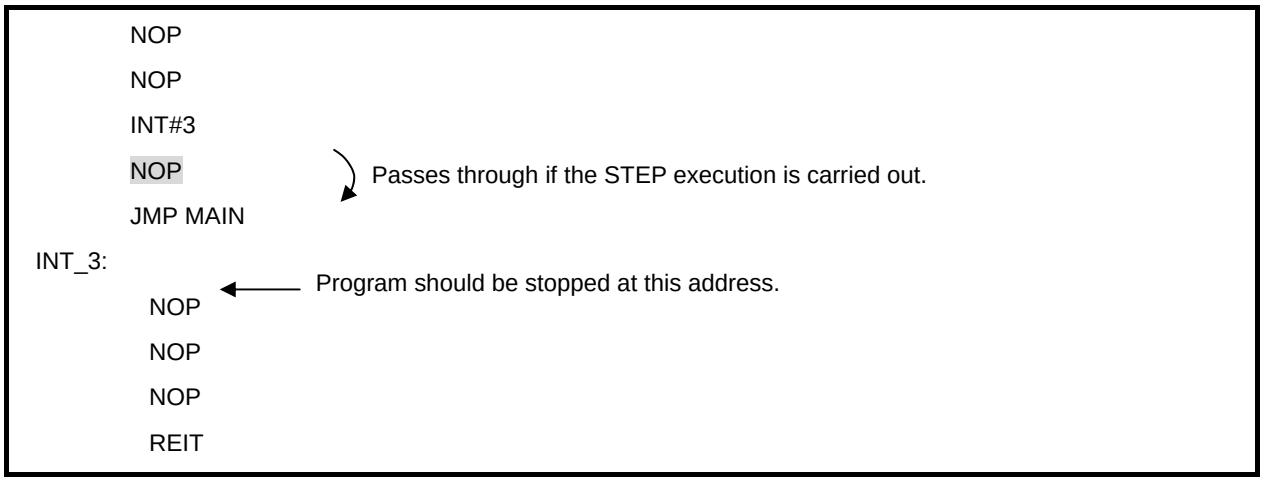


Figure 6.2 Example of Software Interrupt Instruction

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).

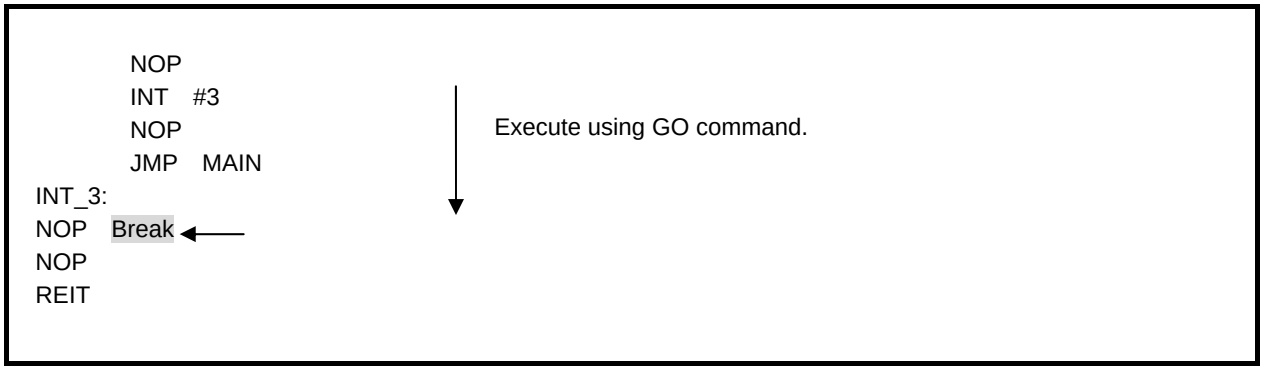


Figure 6.3 Example of INT Instruction

(10) 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.

(11) I/O file

I/O files for the working sample MCUs are not prepared. Refer to “I/O File Format” in the online help of the High-performance Embedded Workshop to make an I/O file for those MCUs.

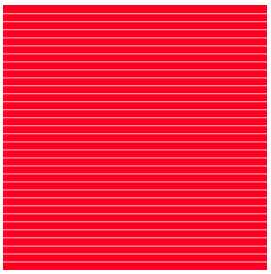
E8a Emulator
Additional Document for User's Manual
Notes on Connecting the R32C/111, R32C/116, R32C/117, R32C/118, R32C/120, R32C/121, R32C/151, R32C/152, R32C/153, R32C/156, R32C/157, R32C/160 and R32C/161

Publication Date: Jun. 26, 2009 Rev.2.01

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department
Renesas Solutions Corp.

E8a Emulator
Additional Document for User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan