

## E2 Emulator, E2 Emulator Lite

Additional Document for User's Manual  
(Notes on Connection of RE Devices)

Supported Devices:

RE Family

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 Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
  - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

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

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

(Rev.4.0-1 November 2017)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan

[www.renesas.com](http://www.renesas.com)

## Trademarks

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

## Contact information

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

[www.renesas.com/contact/](http://www.renesas.com/contact/).

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

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

### 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

### 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

### 3. Input of signal during power-off state

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

### 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

### 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

### 6. Voltage application waveform at input pin

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

### 7. Prohibition of access to reserved addresses

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

### 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

# Table of Contents

1. Overview .....	6
1.1 Overview of E2 Emulator and E2 Emulator Lite .....	6
1.2 Configuration of E2/E2 Lite Manuals .....	6
1.3 Supported Devices.....	7
1.4 Debugging Functions.....	7
2. Designing the User System .....	8
2.1 Connecting the E2/E2 Lite with the User System.....	8
2.2 Installing the Connector on the User System .....	8
2.2.1 Connecting the User System Interface Cable to the 20-Pin Connector .....	9
2.3 Pin Assignments of the Connector on the User System.....	10
2.3.1 20-Pin and 10-Pin Connector Specifications .....	10
2.4 Recommended Circuit between the Connector and the MCU.....	12
2.4.1 SWD Interface Connection between the 20-Pin and 10-Pin Connectors and the MCU.....	12
2.5 Notes on Connection .....	13
2.5.1 RES# Pin.....	13
2.5.2 MD Pin.....	14
2.5.3 EHMD Pin.....	14
2.5.4 GND .....	15
2.5.5 VCC.....	15
2.5.6 TxD9 and RxD9 Pins (Flash Programming via an SCI).....	15
2.6 Internal Circuits of the Emulator .....	16
2.6.1 Internal Circuits of the E2.....	16
2.6.2 Internal Circuits of the E2 Lite.....	17
3. Notes on Usage .....	18
3.1 Turning the Power On/Off.....	18
3.1.1 When a Separate Power Supply is Used for the User System.....	18
3.1.2 When Power is Supplied to the User System from the Emulator .....	19
3.2 Power Supply Function of the E2/E2 Lite .....	19
3.3 Notes on Using the Emulator Debugger.....	20
3.3.1 Notes on Connecting the Emulator Debugger .....	20
3.3.2 Notes on a Debugging Operation that Involves Reprogramming of Flash Memory .....	22
3.3.3 Note on Using Software Breaks in the On-Chip SRAM .....	24
3.3.4 Notes on Using Software Breaks (Common to the On-Chip SRAM and Flash Memory) .....	24
3.3.5 Note on Peripheral I/O Registers Occupied by the Debugger.....	25
3.3.6 Note on Using the Trace Function .....	25
3.3.7 Notes on Low-Power Modes.....	25
3.3.8 Notes on Power Supply Modes.....	26
3.3.9 Notes on Power Control Modes .....	26
3.3.10 Note on the Memory Protection Units (MPUs).....	26
3.4 MCUs that are Used in Debugging .....	27
3.5 Final Evaluation of the User Program.....	27

# Terminology

Some specific words used in this user's manual are defined below.

Host machine

This means a personal computer used to control the emulator.

User system

This means a user's application system in which the MCU to be debugged is used.

User program

This means the program to be debugged.

Programming Software

In this document, this indicates the Renesas Flash Programmer that can be used with the E2 or E2 Lite.

"#" at the end of a pin name (signal name)

"#" at the end of a pin name (signal name) indicates that the pin (signal) is active low (e.g., RES#).

# 1. Overview

## 1.1 Overview of E2 Emulator and E2 Emulator Lite

In this document, we describe 'E2 Emulator' as 'E2' and 'E2 Emulator Lite' as 'E2 Lite'.

The E2 and E2 Lite are on-chip debugging emulators for Renesas' mainstream MCUs.

The E2 Lite is highly affordable development tools providing basic debugging functions. The E2 has the debugging functions of the E2 Lite and also handles high-speed downloading at up to twice the rate of the E2 Lite. In addition, the E2 can supply power that is adjustable from 1.8 V to 5.0 V at 0.1-V intervals. As a development tool, the E2 allows more advanced debugging than the E2 Lite. The E2 can also serve as a Flash Programmer.

## 1.2 Configuration of E2/E2 Lite Manuals

The E2/E2 Lite manual consists of the following.

- E2 Emulator User's Manual
- E2 Emulator Lite User's Manual
- E2 Emulator, E2 Emulator Lite Additional Document for User's Manual

Be sure to read each user's manual before using the E2 or E2 Lite.

### (1) E2 emulator user's manual

The E2 emulator user's manual has the following contents:

- Components of the E2
- E2 hardware specification
- Connection to the E2 and the host machine and user system

### (2) E2 Emulator Lite user's manual

The E2 Emulator Lite user's manual has the following contents:

- Components of the E2 Lite
- E2 Lite hardware specification
- Connection to the E2 Lite and the host machine and user system

### (3) E2 Emulator, E2 Emulator Lite Additional Documents for User's Manual (Notes on Connection of RE Devices) (this document)

The E2 Emulator, E2 Emulator Lite Additional Documents for User's Manual (Notes on Connection of RE Devices) describes information necessary for hardware design such as connection examples and interface circuits.

### (4) Renesas Flash Programmer Flash memory programming software User's Manual

The Renesas Flash Programmer Flash memory programming software User's Manual describes the specifications of the software and the method of operation for the Renesas Flash Programmer.

- For the features and the method of operation of the E2 or E2 Lite emulator debugger, refer to the help system for the e<sup>2</sup> studio.

### 1.3 Supported Devices

**Table 1.1 Supported Device List**

Supported Device	E2			E2 Lite		
	SWD I/F	FINE I/F	SCI I/F	SWD I/F	JTAG I/F	SCI I/F
RE0 series	DBG	—	PRG	DBG	—	—

\* DBG: Can be used for debugging, PRG: Can be used for flash programming

### 1.4 Debugging Functions

**Table 1.2 Debugging Function List**

Item		Supported	Description
Memory reference or change during program execution		√	—
Break	Software break	√	2000 points
	Hardware break	√	Execution address: four points; data access: two points
	Forced break	√	—
Trace		√	Information acquired: <ul style="list-style-type: none"> <li>Branch-source PC values, branch-destination PC values, occurrence of interrupts, and return from interrupts</li> </ul> Start event: <ul style="list-style-type: none"> <li>Start of user-program execution</li> </ul> End events: <ul style="list-style-type: none"> <li>Stopping the user program and trace memory becoming full</li> </ul>
Performance measurement		√	Measurement item: Between start and stop of user program execution
Hot plug-in		×	—
Extended function of the E2	Solution for the tuning of current consumption	×	Current consumption by the user system is measured when the power is supplied from the E2.
	External trigger input/output	×	External triggers can be input to or output from the E2 expansion interface.

√: Supported; ×: Not supported

## 2. Designing the User System

### 2.1 Connecting the E2/E2 Lite with the User System

To connect the E2/E2 Lite, a connector for the user system interface cable must be mounted on the user system.

When designing the user system, read this section of this manual and the hardware manual for the MCUs.

### 2.2 Installing the Connector on the User System

Table 2.1 and Table 2.2 list the recommended connectors and user system interface cables for the E2/E2 Lite, respectively.

**Table 2.1 Recommended Connectors**

Connector	Type Number	Manufacturer	Specifications
20-pin (1.27-mm pin pitch) connector	FTSH-110-01-L-DV-K	Samtec	20-pin surface-mount technology (SMT) straight type
10-pin (1.27-mm pin pitch) connector	FTSH-105-01-L-DV-K	Samtec	10-pin SMT straight type
10-pin (1.27-mm pin pitch) connector	FTSH-105-01-L-DV* (without a marking for matching the position of the connector; keying shroud)	Samtec	10-pin SMT straight type

Note: When using a connector without a guide marking (keying-shroud type), take care with regard to the direction for insertion of the cable.

**Table 2.2 User System Interface Cables**

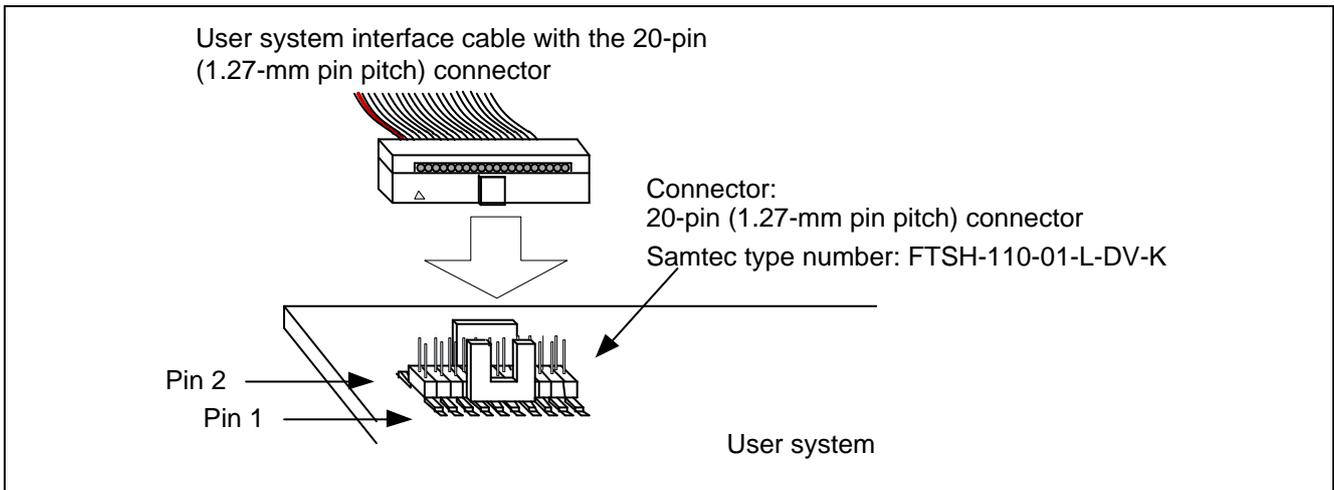
Cable Type	E2	E2 Lite
20-pin to 20-pin cable* (for the 20-pin (1.27-mm pin pitch) connector)	Comes with the product	Separately sold
20-pin to 10-pin cable (for the 20-pin (1.27-mm pin pitch) connector)	Separately sold	Separately sold

Note: The 20-pin to 20-pin cable can be connected to the guideless 10-pin (1.27-mm pin pitch) connector; when doing so, however, check the pin assignments and take care with regard to the direction for insertion of the cable.

Only connect the E2/E2 Lite after confirming that there are no mismatches of alignment on the user system port connector. Incorrect connection will result in the host machine, the E2/E2 Lite, and the user system emitting smoke or catching fire.

### 2.2.1 Connecting the User System Interface Cable to the 20-Pin Connector

Figure 2.1 shows how to connect the user system interface cable to the 20-pin connector of the E2/E2 Lite.



**Figure 2.1 Connecting the User System Interface Cable to the 20-Pin Connector of the E2/E2 Lite**


**CAUTION**

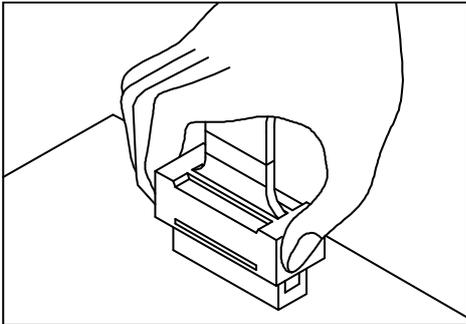
Notes on connector insertion and removal:



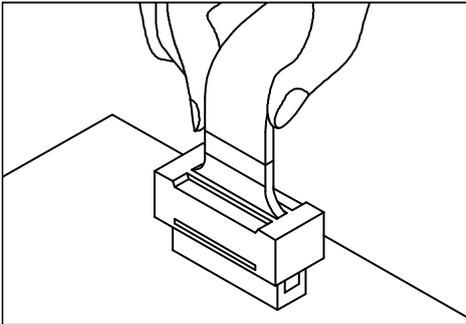
When connecting or disconnecting the user-system interface cable and the emulator or user system, grasp the connector cover at the end of the cable. Pulling the cable itself will damage the wiring.

Also, be aware that the user-system interface cable has the direction in which it must be inserted. If the cable is connected in the wrong direction, it may be damaged.

Correct example



Incorrect example



## 2.3 Pin Assignments of the Connector on the User System

### 2.3.1 20-Pin and 10-Pin Connector Specifications

Figure 2.2 shows the specifications of the 20-pin and 10-pin connectors.

Table 2.3 shows the pin assignments of the 20-pin and 10-pin connectors for SWD interface connection.

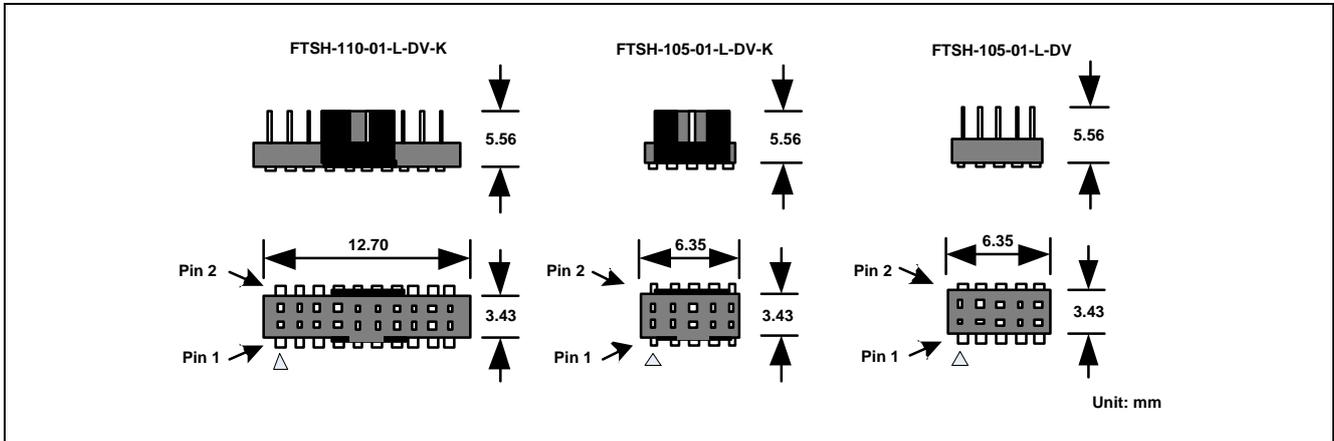


Figure 2.2 20-Pin and 10-Pin Connector Specifications

**Table 2.3 Pin Assignments of the 20-Pin and 10-Pin Connectors for SWD Interface Connection**

Pin No.	R7F0E01 Group		Note
	Signal *1	Direction *2	
1	VCC	–	Power supply
2	SWDIO/RxD9	I/O / Input	Clock for debugging communications/ Communications data for flash programming
3	GND *3	–	
4	SWCLK/TxD9	Input/ Output	Clock for debugging communications/ Communications data for flash programming
5	GND *3	–	
6	NC	–	
7	NC	–	
8	NC	–	
9	GND *3	–	
10	RES#	I/O	User system reset
11	NC *4	–	
12	NC *4	–	
13	NC *4	–	
14	NC *4	–	
15	GND *3, *4	–	
16	NC *4	–	
17	GND *3, *4	–	
18	NC *4	–	
19	GND *3, *4	–	
20	NC *4	–	

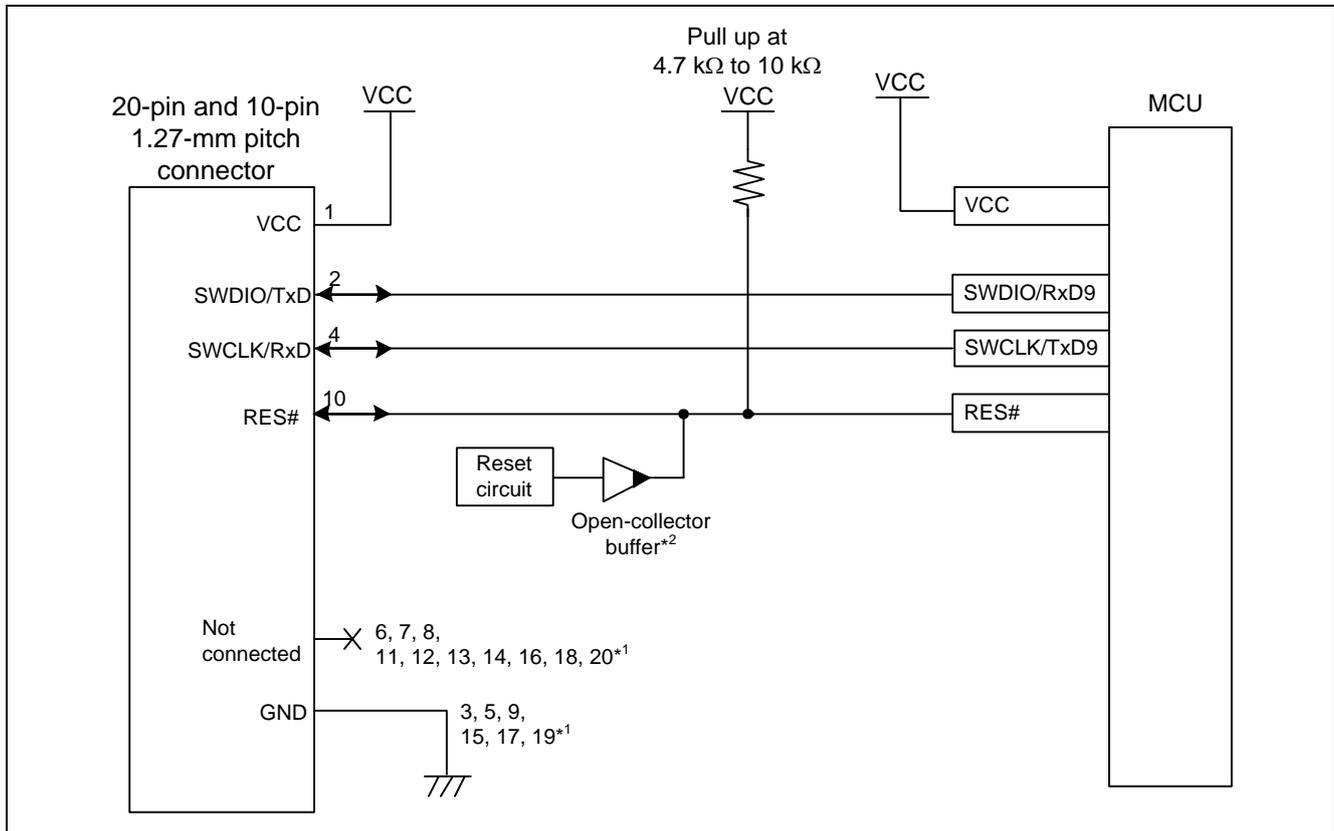
- Notes:
1. These are the names of the MCU pins at the time the E2/E2 Lite is connected (i.e. during debugging).
  2. Input to or output from the user system.
  3. Securely connect pins 3, 5, 9, 15, 17, and 19 of the connector to GND on the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E2/E2 Lite.
  4. If a 10-pin (1.27-mm pin pitch) connector is mounted on the user system, pins 11 to 20 are not used.

## 2.4 Recommended Circuit between the Connector and the MCU

This section shows recommended circuits for connection between the connector and the MCU when the E2/E2 Lite is used. For details on the handling of signals, refer to section 2.5, Notes on Connection.

### 2.4.1 SWD Interface Connection between the 20-Pin and 10-Pin Connectors and the MCU

Figure 2.3 shows a recommended circuit for connection between the 20-pin and 10-pin connectors and the MCU in SWD interface connection.



**Figure 2.3 Example of Connection between the 20-Pin and 10-Pin Connectors and the MCU in SWD Interface Connection**

- Notes:
1. If a 10-pin (1.27-mm pin pitch) connector is mounted on the user system, pins 11 to 20 are not used.
  2. The output of the reset circuit of the user system must be open collector.
  3. Since the E2 Lite does not support the SCI, it does not allow the programming of flash memory with the use of programming software.
  4. The MD pin can operate even if it is not connected to the E2/E2 Lite. When the pin is not connected to the E2/E2 Lite, a special circuit for the pin must be configured on the user system. For details on the handling of the MD pin, refer to section 2.5, Notes on Connection.

## 2.5 Notes on Connection

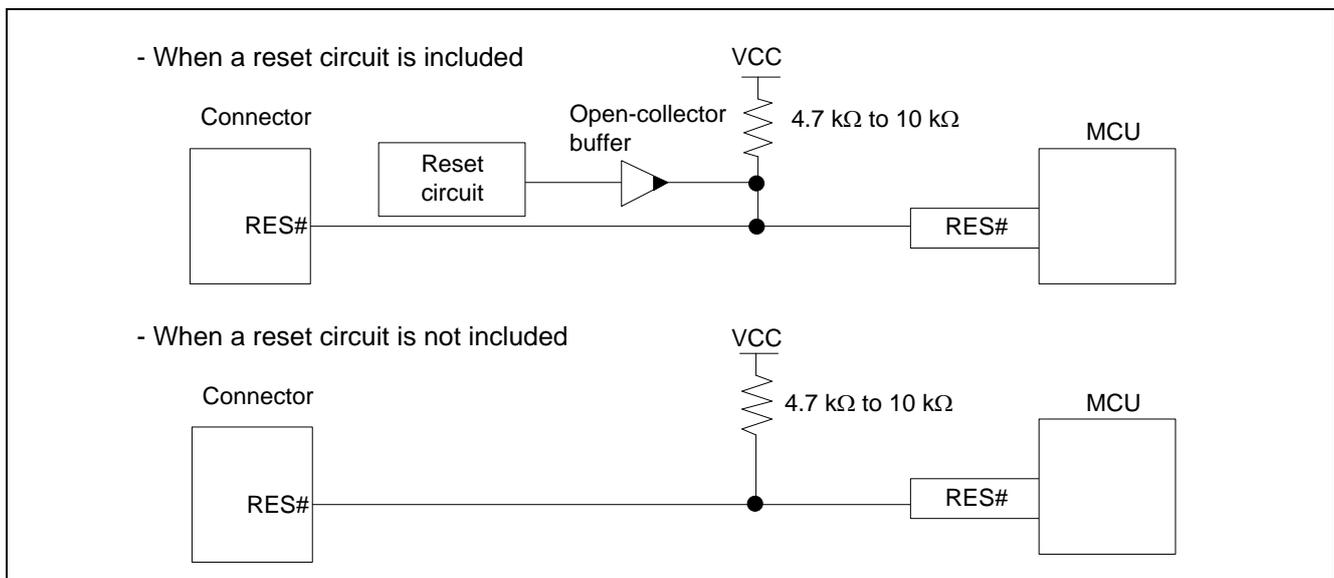
Wiring patterns between the connector and the MCU must be as short as possible (within 50 mm is recommended). Do not connect the signal lines between the connector and MCU to other signal lines on the board.

For the handling of pins while the E2/E2 Lite is not in use, refer to the hardware manual for the MCU.

### 2.5.1 RES# Pin

The E2/E2 Lite uses the RES# pin.

If the user system includes a user logic reset circuit, the output signal from the reset circuit must be connected to the RES# pin of the connector via an open-collector buffer as shown below. If there is no reset circuit, the RES# pin from the connector must be directly connected to the RES# pin of the MCU.



**Figure 2.4 Connection of the RES# Pin**

Do not install capacitors, series resistors, or filters on signal lines; if this is attempted, correct communication may not be established.

### 2.5.2 MD Pin

Connection of the MD pin to the connector is optional.

When debugging with the E2/E2 Lite, set the MD pin to the high level. When rewriting flash memory by the programming software, set the MD pin to the low level.

When the contents of flash memory are being rewritten by the programming software, the MD pin must be handled so that it is switched to the low level. The circuit for the MD pin of the MCU includes a pull-up resistor.

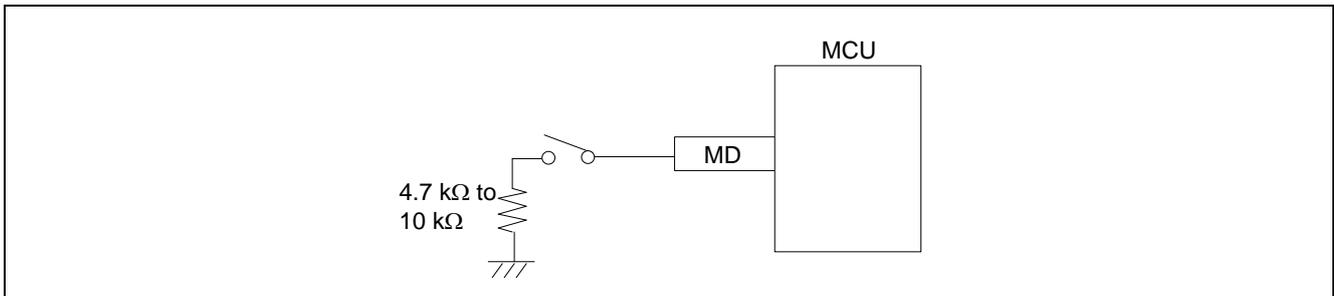


Figure 2.5 Connection of the MD Pin

### 2.5.3 EHMD Pin

In debugging with the E2/E2 Lite, the MCU cannot enter the energy harvesting mode. The EHMD pin must be handled so that it is switched to the low level.

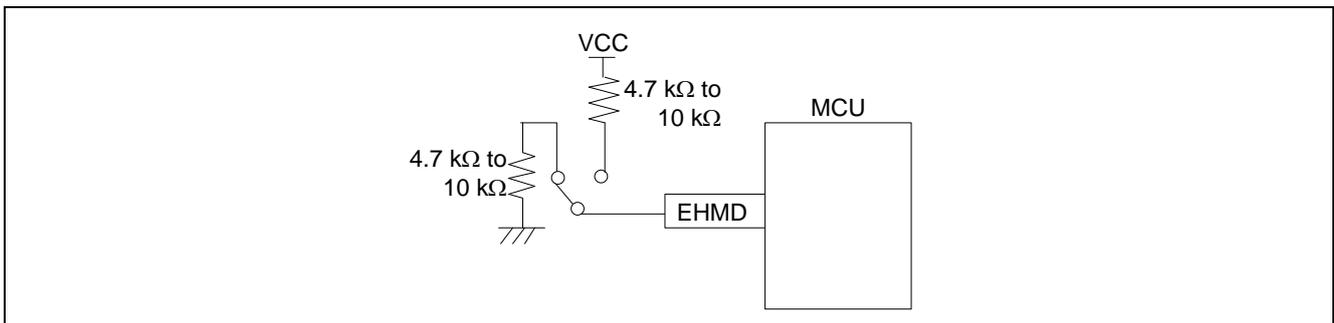


Figure 2.6 Connection of the EHMD Pin

### 2.5.4 GND

The pins of the connector marked "GND" must be at the same ground level as the VSS pin of the MCU.

### 2.5.5 VCC

Connect the VCC of the connector to the VCC (power supply) of the user system.

Use the emulator within the power supply voltage of 1.8 V to 5.5 V and within the operating voltage range of the MCU.

When power is supplied to the user system from other than the emulator, the E2/E2 Lite consumes the power supply for the last output and first input buffers of the emulator.

- E2: 3.3 V: approximately 20 mA, 5.0 V: approximately 40 mA
- E2 Lite: 3.3 V: approximately 20 mA, 5.0 V: approximately 40 mA

The E2/E2 Lite can supply power to a simple evaluation system.

- E2: Can supply power of 1.8 V to 5.0 V, up to 200 mA.
- E2 Lite: Can supply power of 3.3 V, up to 200 mA.

When using the power supply function of the E2 or E2 Lite, check the voltage that is actually being supplied to the user system since this depends on the environment.

---

Power supply from the E2/E2 Lite depends on the quality of the USB power supply of the host machine, and as such, precision is not guaranteed. When writing a program that requires reliability, do not use the power supply function of the E2/E2 Lite. Use a stable, separate power supply for the user system. As the software when writing a program in a mass-production process, use the Renesas Flash Programmer.

For details on the programming software, refer to the following:

Renesas Flash Programmer: <https://www.renesas.com/RFP>

---

When the MCU is changed to low power mode, the internal debugging circuit continues to run. This leads to the MCU drawing more electric current than is listed in the DC characteristics of the target MCU.

---



## WARNING

Warning for Turning the Power On/Off:

When supplying power, ensure that there are no shorts between Vcc and GND. Only connect the E2/E2 Lite after confirming that there are no mismatches of alignment on the user system port connector. Incorrect connection will result in the host machine, the E2/E2 Lite, and the user system emitting smoke or catching fire.

### 2.5.6 TxD9 and RxD9 Pins (Flash Programming via an SCI)

When flash memory is programmed via an SCI, the RxD9 and TxD9 pins must be connected to the emulator. If the MCU has multiple RxD9 or TxD9 pins, check the hardware manual of the MCU to confirm which of the pins is used in boot mode.

## 2.6 Internal Circuits of the Emulator

### 2.6.1 Internal Circuits of the E2

Figure 2.7 shows the internal circuits of the E2 with the MCU connected via the SWD interface.

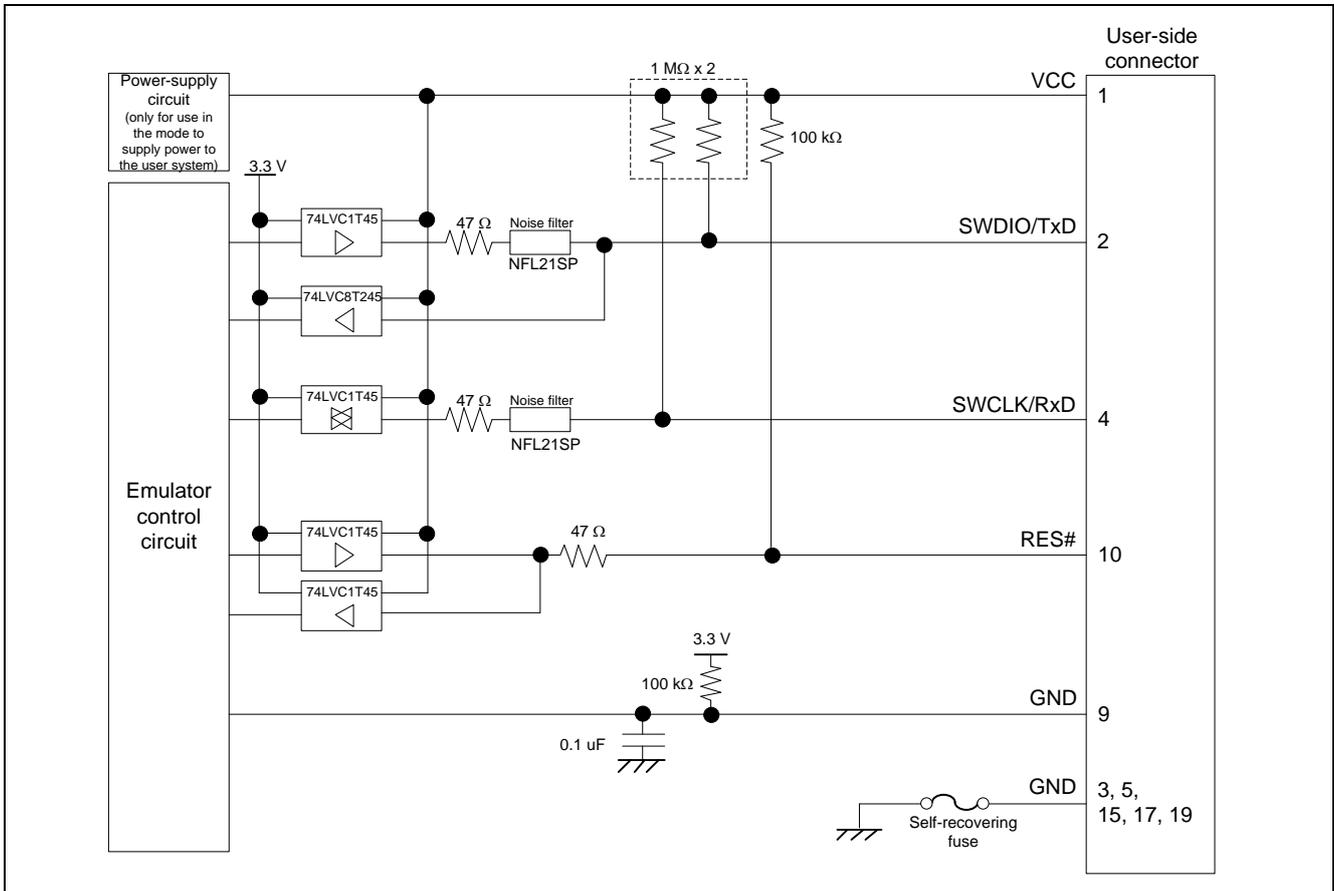


Figure 2.7 Internal Circuits of the E2 with the MCU Connected via the SWD Interface

2.6.2 Internal Circuits of the E2 Lite

Figure 2.8 shows the internal circuits of the E2 Lite with the MCU connected via the SWD interface.

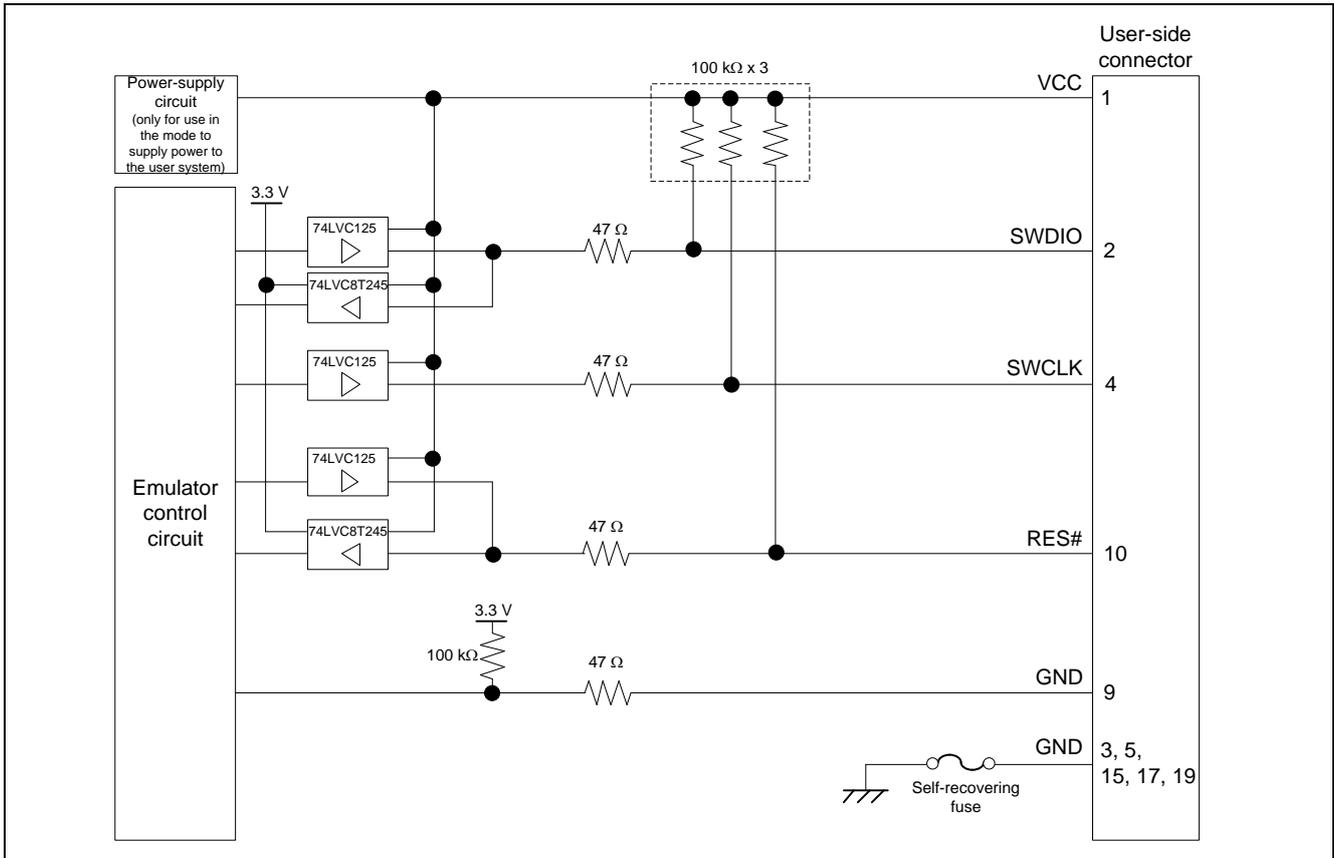


Figure 2.8 Internal Circuits of the E2 Lite with the MCU Connected via the SWD Interface

## 3. Notes on Usage

### 3.1 Turning the Power On/Off

Turn the power of the E2/E2 Lite and the user system following the procedure below.

#### 3.1.1 When a Separate Power Supply is Used for the User System

<When using the emulator>

- (1) Check that the power is off.
- (2) Connect the user system.  
Connect the emulator and the user system with a user-system interface cable.
- (3) Connect the host machine and turn on the emulator.  
Connect the emulator and the host machine with a USB interface cable. The E2/E2 Lite is turned on by connecting the USB interface cable.
- (4) Launch the emulator debugger or programming software.  
Launch the emulator debugger or programming software.
- (5) Turn on the user system.  
Turn on the user system.
- (6) Connect the emulator debugger or programming software to the emulator.  
Connections may vary depending on software.

<When finished using the emulator>

- (1) Disconnect the emulator from the emulator debugger or programming software.  
Disconnections may vary depending on software.
- (2) Turn off the user system.  
Turn off the user system.
- (3) Close the emulator debugger or programming software.  
Close the emulator debugger or programming software.
- (4) Turn off the emulator and disconnect the emulator.  
Disconnect the USB interface cable from the emulator. The E2/E2 Lite is turned off by disconnecting from the USB interface cable.
- (5) Disconnect the user system.  
Disconnect the user-system interface cable from the user system.

### CAUTION

Notes on the User System Power Supply:



While the power of the user system is on, do not turn off the host machine or unplug the USB interface cable.

The user system may be damaged due to leakages current.

### 3.1.2 When Power is Supplied to the User System from the Emulator

<When using the emulator>

(1) Connect the user system.

Connect the emulator and the user system with a user-system interface cable.

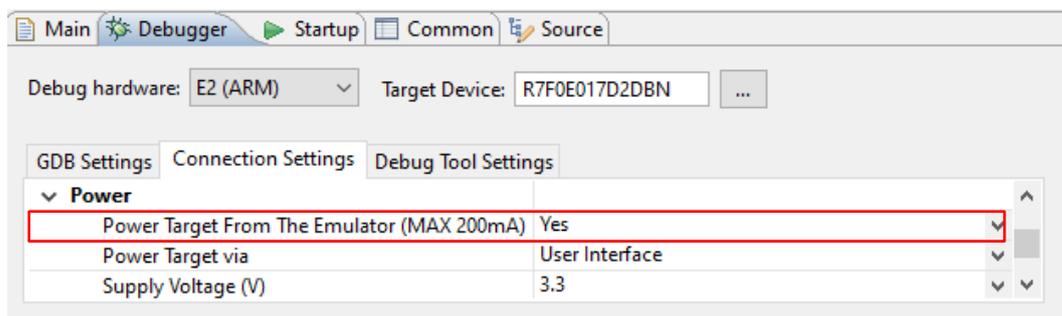
(2) Connect the host machine and turn on the emulator.

Connect the emulator and the host machine with a USB interface cable, then turn on the emulator.

(3) Launch the emulator debugger.

Launch the emulator debugger and select the setting of power supply to the user system.

Under [Power] on the [Connection Settings] tabbed page, select [Yes] for [Power Target From The Emulator (MAX 200mA)] (the operation is explained for the case when the e<sup>2</sup> studio is in use). Refer to section 3.3, Notes on Using the Emulator Debugger, for how to open the [Debug Configurations] window.



**Figure 3.1 Setting for Supplying Power**

(4) Connect the emulator debugger or programming software to the emulator.

Connections may vary depending on software.

<When finished using the emulator>

(1) Disconnect the emulator from the emulator debugger or programming software.

Disconnections may vary depending on software.

(2) Close the emulator debugger or programming software.

Close the emulator debugger or programming software.

(3) Turn off the emulator and disconnect the emulator.

Disconnect the USB interface cable from the emulator, then turn off the emulator.

(4) Disconnect the user system.

Disconnect the user-system interface cable from the user system.

## 3.2 Power Supply Function of the E2/E2 Lite

The E2/E2 Lite can supply power to a simple evaluation system.

- E2: Can supply power of 1.8 V to 5.0 V, up to 200 mA.
- E2 Lite: Can supply power of 3.3 V, up to 200 mA.

When using the power supply function of the E2 or E2 Lite, check the voltage that is actually being supplied to the user system since this depends on the environment.

### 3.3 Notes on Using the Emulator Debugger

This section describes how to set the [Debug Configurations] window of the e<sup>2</sup> studio. To open the [Debug Configurations] window, click on [Run] → [Debug Configurations...] or the downward-pointing arrow next to the  icon → [Debug Configurations...].

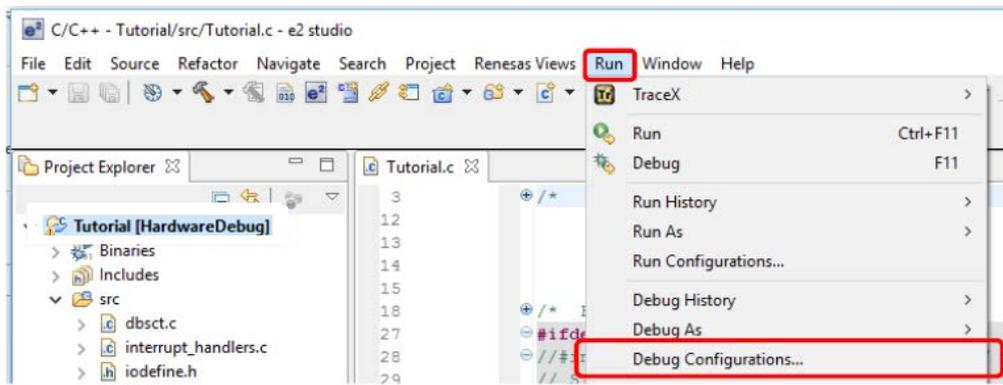


Figure 3.2 Opening the [Debug Configurations] Window

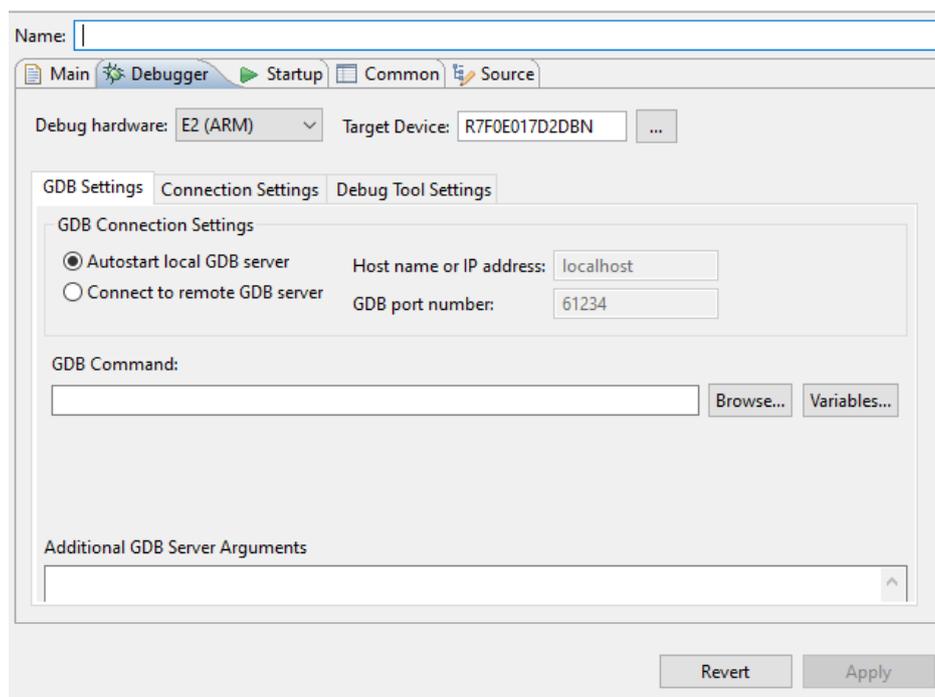


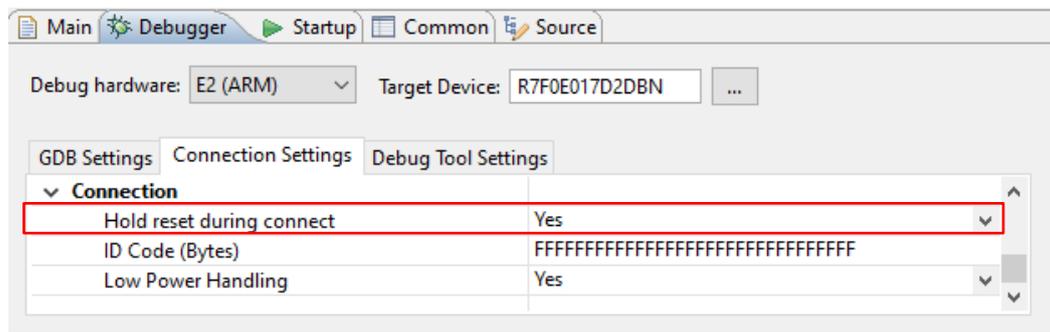
Figure 3.3 [Debug Configurations] Window

#### 3.3.1 Notes on Connecting the Emulator Debugger

##### (1) Reset state

Under [Connection] on the [Connection Settings] tabbed page, be sure to select [Yes] for [Hold reset during connect] (the operation is explained for the case when the e<sup>2</sup> studio is in use).

When [Yes] is selected, during connection to the emulator debugger, the emulator maintains the low-level output on the RES# pin of the MCU and places the MCU in the OCD mode. However, to start the operation of the built-in debugging circuits of the MCU, the emulator releases the reset for about 50 msec while it is connected, and the user program automatically runs during that period.



**Figure 3.4 Setting of [Hold reset during connect]**

#### (2) Startup mode

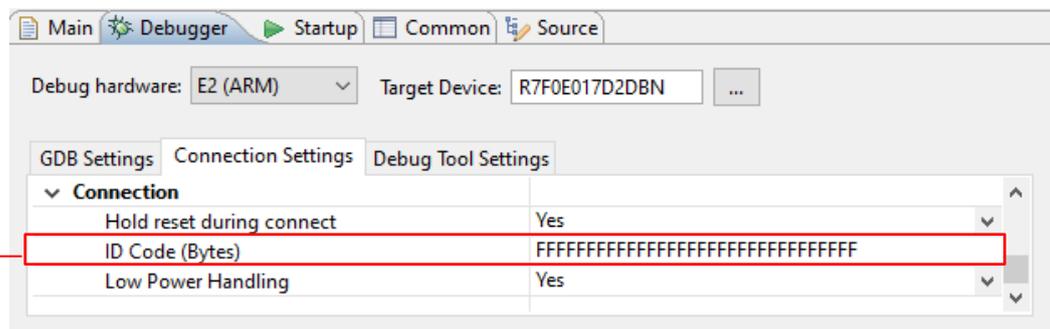
When the emulator debugger is connected, handle the pins on the user system so that the startup mode of the MCU is normal mode.

In normal mode, the MD pin is at the high level and the EHMD pin is at the low level. Correct connection of the emulator debugger is not possible in energy harvesting startup mode or SCI/USB boot mode.

#### (3) Debugging after rewriting ID code

If the ID code (OSIS register) has been rewritten, enter the new ID code.

Here, the ID code to be entered means the value to be entered in [ID Code (Bytes)] under [Connection] on the [Connection Settings] tabbed page (the operation is explained for the case when the e<sup>2</sup> studio is in use).



For example, when the following values have been secured, enter 00112233445566778899aabbccddeeff.

```
const unsigned long __OSISreg[4] = {
    0x00112233,
    0x44556677,
    0x8899aabb,
    0xccddeeff
};
```

**Figure 3.5 Setting of [ID Code (Bytes)]**

#### (4) Entering the ALeRASE command for ID code

If an ALeRASE command is entered in [ID Code (Bytes)] under [Connection] on the [Connection Settings] tabbed page, the code flash memory and option-setting memory will be erased when the emulator debugger is connected.

For the conditions under which the ALeRASE command is usable, refer to the hardware manual for the MCU you are using. If the ALeRASE command is entered then the emulator debugger is connected while the command is not usable on the MCU, an error message "Failed to erase all flash memory by the ID for erasing all flash memory." is displayed and the connection process is suspended.

### 3.3.2 Notes on a Debugging Operation that Involves Reprogramming of Flash Memory

A “debugging operation that involves reprogramming of flash memory” refers to the following operations of the emulator debugger.

- Downloading data to flash memory
- Using software break functions in flash memory
  - (a) Setting and canceling breakpoints
  - (b) Executing or step-executing programs from a breakpoint
  - (c) Using the “Run to Line” function from the state where a break was set

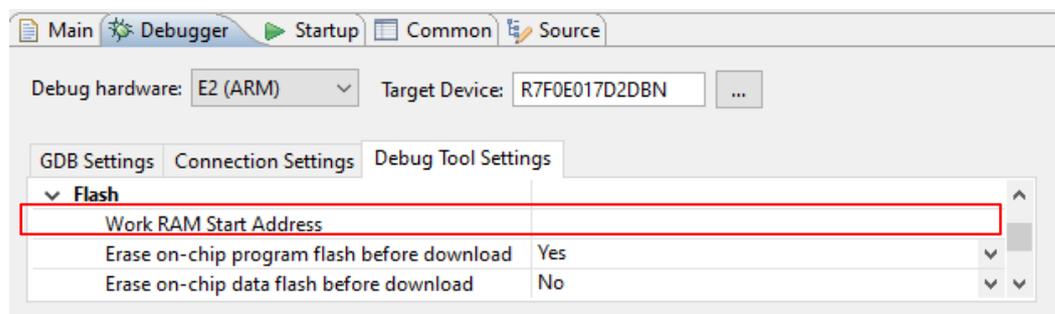
#### (1) Program for reprogramming flash memory

Since the emulator debugger enables a debugging operation that involves reprogramming of flash memory, the emulator writes the program for reprogramming flash memory to the on-chip SRAM and executes the program to reprogram the flash memory. After the flash memory has been reprogrammed, the emulator debugger restores the on-chip SRAM to its initial state.

#### (2) Destination for allocation of the program for reprogramming flash memory

By default, the program for reprogramming flash memory is allocated to the 5-Kbyte space from the address 2000 8000h in the on-chip SRAM. If the default allocation destination is not available due to the security settings or DMAC/DTC transfer\*, enter the start address of an available space in the on-chip RAM in units of 1000h bytes against [Work RAM Start Address] under [Flash] on the [Debug Tool Settings] tabbed page for the emulator debugger (the operation is explained for the case when the e<sup>2</sup> studio is in use).

Note: The DMAC or DTC will continue to operate even during a break. Take care that a transfer source or destination for the DMAC or DTC is not within the address range of the working RAM where the program is to be allocated.



**Figure 3.6 Setting of [Work RAM Start Address]**

#### (3) Interrupts and resets during execution of the program for reprogramming flash memory

Interrupts other than non-maskable interrupts are to be masked while the program for reprogramming flash memory is being executed. Also, so that the program for reprogramming flash memory is correctly executed, all interrupt source flags which have been set before executing the program are cleared.

If a non-maskable interrupt occurs, the emulator continues running the program for reprogramming flash memory. If a reset occurs while the program for reprogramming flash memory is being executed, the emulator shows an error message and stops processing. Since doing so may damage the contents of flash memory, do not apply a reset while the program is running.

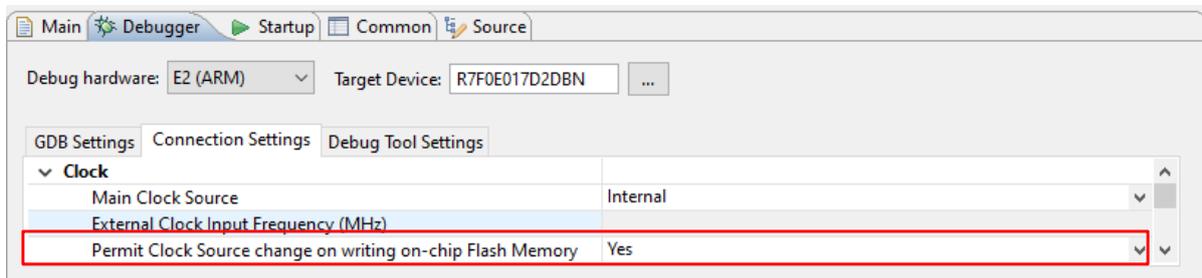
#### (4) Conditions for downloading data to flash memory being available

When the MCU satisfies all the following conditions, the downloading of data to flash memory can proceed.

- (a) The power supply mode of the MCU is the all power supplied mode (ALLPWON).
- (b) The power control mode of the MCU is the normal mode (NORMAL).
- (c) The code flash memory of the MCU is in read mode.

- (d) The frequency of the system clock (ICLK) of the MCU is 1 MHz or higher.\*

Note: For [Clock] in the [Connection Settings] tabbed page of the emulator debugger, when [Yes] is selected for [Permit Clock Source change on writing on-chip Flash Memory], condition (d) can be excluded (the operation is explained for the case when the e<sup>2</sup> studio is in use).



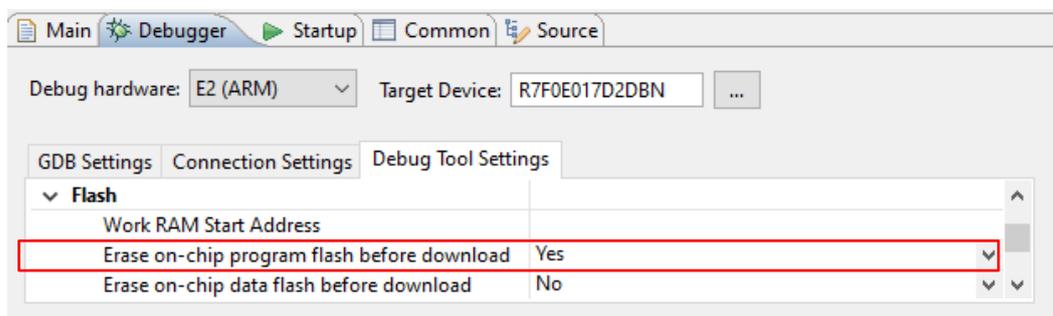
**Figure 3.7 Setting of [Permit Clock Source change on writing on-chip Flash Memory]**

- (e) The flash memory function stop bit (FLFSTP) in the function stop control register (FSTPCR) is 0.

If data are downloaded to flash memory while any condition is not satisfied, the emulator shows an error message and stops processing. In such a case, restart downloading of data to flash memory after the CPU has been reset or reconnect the emulator debugger after reviewing its settings.

#### (5) Downloading data to a secure area

When downloading data to a secure area which has been specified for the security MPU, for [Flash] in the [Debug Tool Settings] tabbed page of the emulator debugger, select [Yes] for [Erase on-chip program flash before download] (the operation is explained for the case when the e<sup>2</sup> studio is in use).



**Figure 3.8 Setting of [Erase on-chip program flash before download]**

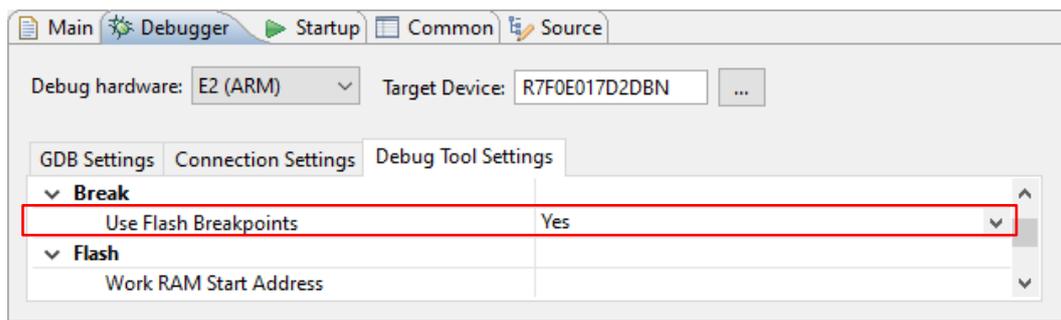
#### (6) Access window function

When the access window function of the MCU is to be used, only reprogram the flash memory in the area specified for access.

#### (7) Conditions for using software breaks in flash memory

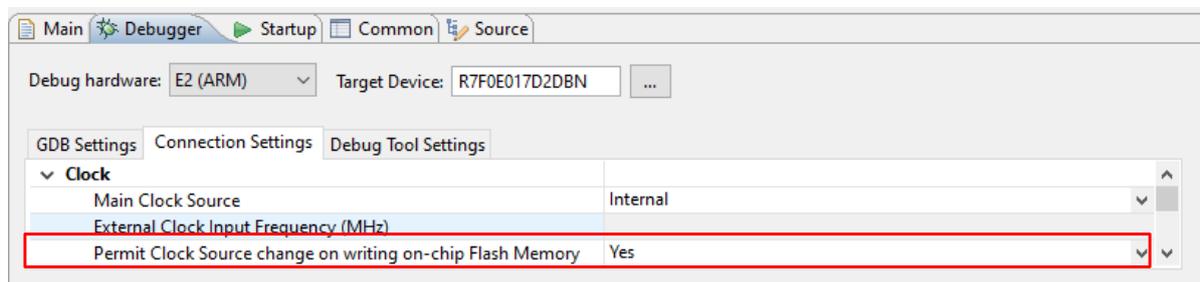
When the MCU satisfies all the following conditions, the software break function for flash memory is enabled.

- The power supply mode of the MCU is the all power supplied mode (ALLPWON).
- The power control mode of the MCU is the normal mode (NORMAL).
- The code flash memory of the MCU is in read mode.
- The frequency of the system clock (ICLK) of the MCU is 1 MHz or higher.\*
- The flash memory function stop bit (FLFSTP) in the function stop control register (FSTPCR) is 0.
- For [Break] in the [Debug Tool Settings] tabbed page, [Yes] is selected for [Use Flash Breakpoints] (the operation is explained for the case when the e<sup>2</sup> studio is in use).



**Figure 3.9 Setting of [Use Flash Breakpoints]**

Note: For [Clock] in the [Connection Settings] tabbed page of the emulator debugger, when [Yes] is selected for [Permit Clock Source change on writing on-chip Flash Memory], condition (d) can be excluded (the operation is explained for the case when the e<sup>2</sup> studio is in use).



**Figure 3.10 Setting of [Permit Clock Source change on writing on-chip Flash Memory]**

If the software break function is used with any condition not satisfied, the emulator shows an error message. In such a case, use the hardware break or confirm that conditions (a) to (f) above are satisfied.

#### (8) Flash memory I/O register

After a debugging operation that involves reprogramming of flash memory, the value of the flash memory I/O register is rewritten by the emulator debugger.

### 3.3.3 Note on Using Software Breaks in the On-Chip SRAM

#### (1) Overwriting of software breakpoints by user programs

If a software breakpoint is overwritten by a user program, the program will not stop even if it runs through the address. In such a case, set the software breakpoint after the target on-chip SRAM has been rewritten by the program.

### 3.3.4 Notes on Using Software Breaks (Common to the On-Chip SRAM and Flash Memory)

#### (1) Setting software breakpoints in a secure area

Do not set software breakpoints in a secure area which has been specified by the security MPU. If this is attempted, an instruction code may be restored as incorrect data on release of the break.

#### (2) Reading an address where a software breakpoint has been set

Do not read an address where a user program has set a software breakpoint. Doing so may cause the program to operate in a different way from in the normal state.

#### (3) Viewing memory in the [Memory] view

During execution of the user program, if a range of memory in which a software breakpoint has been set is displayed in the [Memory] view of the emulator debugger, the value (BKPT instruction code) that is shown is different from that in the actual program data.

#### (4) Removal of software breakpoints when the emulator debugger is to be disconnected

When the emulator debugger is to be disconnected, remove all software breakpoints that have been set. At this time, reset the CPU since the emulator debugger is certain to reprogram the flash memory.

### 3.3.5 Note on Peripheral I/O Registers Occupied by the Debugger

#### (1) Peripheral I/O registers occupied by the debugger

The emulator debugger occupies the following peripheral I/O registers during debugging. Do not change the values of these registers, since continued debugging might not be possible after having done so.

- Debug stop control register (DBGSTOPCR)
- Micro trace buffer (MTB) (SFR area)
- System control OCD control register (SYOCDRCR)

### 3.3.6 Note on Using the Trace Function

#### (1) MTB tracing

When the MTB trace function is in use, the trace recording area is the size of recorded tracing that has been selected for the emulator debugger from the address where the on-chip SRAM starts. When the on-chip SRAM is in use by a user program, do not use the trace recording area. For details on the address where the on-chip SRAM starts, refer to the hardware manual for the given MCU.

If the size of the trace recording area for the emulator debugger is changed, the startup routine must also be changed.

### 3.3.7 Notes on Low-Power Modes

#### (1) Debugging in SSTBY or SNOOZE mode

In SSTBY or SNOOZE mode, the emulator debugger does not have access to the system bus of the MCU. While the user program is being executed or during mode transitions of the MCU, setting and viewing of the memory or peripheral I/O registers and setting and changing breakpoints are not possible.

#### (2) Forcibly stopping a program in SSTBY or SNOOZE mode

When a program is forcibly stopped in SSTBY or SNOOZE mode, proceed with one of the following operations. Each operation leads to release from SSTBY or SNOOZE mode.

- Using [Reset ] of the emulator debugger makes the MCU stop the user program and go to the position indicated by the reset vector.
- Using [Suspend ] of the emulator debugger stops the MCU at the next instruction after the WFE instruction which led to the mode transition. When [Suspend ] is to be used, for [Connection] on the [Connection Settings] tabbed page of the emulator debugger, select [Yes] for [Low Power Handling] (the operation is explained for the case when the e<sup>2</sup> studio is in use).

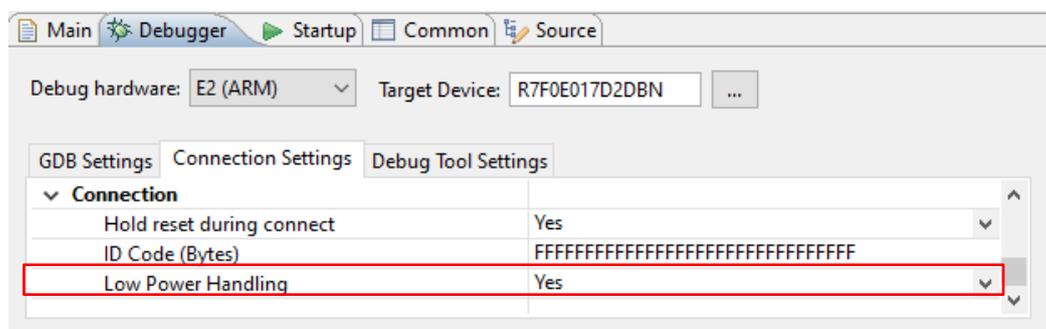


Figure 3.11 Setting of [Low Power Handling]

## (3) Debugging in DSTBY mode

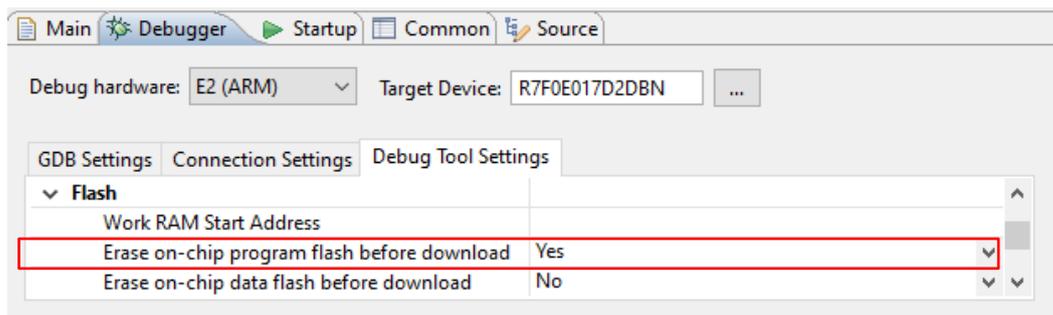
Debugging a program that has entered DSTBY is not supported.

### 3.3.8 Notes on Power Supply Modes

## (1) Downloading a program that includes a transition to MINPWON or EXFPWON mode

To download a program that includes a transition to MINPWON or EXFPWON mode, for [Flash] on the [Debug Tool Settings] tabbed page of the emulator debugger, select [Yes] for [Erase on-chip program flash before download] (the operation is explained for the case when the e<sup>2</sup> studio is in use).

If [No] is selected, invalid data which can be read in the state where the power to the flash memory is shut down may be written to code flash memory.



**Figure 3.12 Setting of [Erase on-chip program flash before download]**

## (2) Displaying code flash memory in MINPWON or EXFPWON mode

While the emulator is operating with the MCU in MINPWON or EXFPWON mode, the state of flash memory to which power has been shut down cannot be confirmed even if contents of code flash memory are being displayed on the emulator debugger.

Program data that have been downloaded continue to be displayed by the emulator debugger. To confirm the transition to MINPWON or EXFPWON mode, refer to the status register.

## (3) Current drawn during debugging

Since the debugging circuits within the MCU are always active during connection of the emulator debugger, the MCU draws more current than in the actual user system. Take care on this point when attempting to measure the current drawn in the user system.

### 3.3.9 Notes on Power Control Modes

## (1) Transitions to the boost mode

Transitions of the MCU to the boost mode are not possible during debugging.

## (2) Operation with the PLL

The PLL cannot be used as the MCU's operating clock during debugging.

### 3.3.10 Note on the Memory Protection Units (MPUs)

## (1) Access to protected areas

Table 3.1 shows the operation of the emulator debugger in response to attempted access to areas protected by each of the MPUs.

**Table 3.1 Access to MPU-Protected Areas**

Protected Area	Operation of the Emulator Debugger
Arm® MPU	Accessible
Bus master MPU	The protected area is not accessible.
Slave MPU	Accessible
Security MPU	On-chip SRAM/peripheral I/O registers <ul style="list-style-type: none"> <li>• Reading: A dummy value (0x00) is read.</li> <li>• Writing: Writing is ignored.</li> </ul> Flash memory <ul style="list-style-type: none"> <li>• Reading: A dummy value (0x00) is read.</li> <li>• Writing: Writing is possible if data are downloaded and written.</li> </ul>

### 3.4 MCUs that are Used in Debugging

After debugging with the E2/E2 Lite, if the MCU is disconnected from the emulator and run on its own, correct operation cannot be guaranteed. To operate the MCU on its own, use the programming software to re-program the MCU.

MCUs that are connected to the E2/E2 Lite and used in debugging are placed under stress by repeated programming of flash memory during emulation. Do not use MCUs that were used in debugging in mass-production for end users.

### 3.5 Final Evaluation of the User Program

Before entering the mass-production phase, be sure to perform a final evaluation of the program which has been written to the flash ROM by the programming software, without the E2/E2 Lite connected.

Revision History	<b>E2 Emulator, E2 Emulator Lite</b> <b>Additional Document for User's Manual</b> <b>(Notes on Connection of RE Devices)</b>
------------------	--

Rev.	Date	Description	
		Page	Summary
1.00	Jul.16.19	—	—

---

E2 Emulator, E2 Emulator Lite  
Additional Document for User's Manual  
(Notes on Connection of RE Devices)

Publication Date: Rev.1.00      Jul.16.19

Published by:      Renesas Electronics Corporation

---

E2 Emulator, E2 Emulator Lite  
Additional Document for User's Manual  
(Notes on Connection of RE Devices)