E1/E20/E2 Emulator, E2 Emulator Lite

Additional Document for Manual
(Notes on Connection of RL78)

Supported Devices:
RL78 Family
RL78/D1A, RL78/F12, RL78/F13, RL78/F14, RL78/F15, RL78/F1A, RL78/F22, RL78/F23, RL78/F24, RL78/F25, RL78/G11, RL78/G12, RL78/G13, RL78/G14, L78/G1A, L78/G1C, RL78/G1D, RL78/G1E, RL78/G1F, RL78/G1G, L78/G1H, RL78/G1P, RL78/G13A, RL78/G21, RL78/G22, RL78/G23, RL78/G24, RL78/H1D, RL78/I1A, RL78/I1B, RL78/I1C, RL78/I1D, RL78/I1E, RL78/L12, RL78/L13, RL78/L1A, RL78/L1C, RL78/L23, RL78/FGIC

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

Renesas Electronics
www.renesas.com

Rev.9.50 Oct 2026

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. 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.
6. 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.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. 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.5.0-1 October 2020)

Corporate Headquarters

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

© 2026 Renesas Electronics Corporation. All rights reserved.

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.

Contents

1. Overview 7
1.1 Overview of E1/E20/E2 Emulator and E2 emulator Lite 7
1.2 Note on Using E20 7
1.3 Configuration of E1/E20/E2/E2 Lite Manuals 8
1.4 Supported Devices 9
1.5 Overview of the Specifications of the External Trigger Input and Output 11
1.5.1 Assignments of the External Trigger Input and Output Pins 11
1.5.2 Specifications of the External Trigger Inputs and Outputs 12

2. Designing the User System 13
2.1 Connecting the E1/E20/E2/E2 Lite to the User System 13
2.2 Installing the Connector on the User System 13
2.2.1 Connecting the User System Interface Cable to the 14-Pin Connector 13
2.3 Pin Assignments of the Connector on the User System 17
2.3.1 14-Pin Connector Specifications 17
2.4 Recommended Circuits between the Connector and the MCU 19
2.4.1 Connection between the 14-Pin Connector and the RL78 Family MCUs in General 19
2.4.2 Connection between the 14-Pin Connector and the RL78 Family MCUs (Only the RL78/G11, the RL78/G21, and the 20- and 24-Pin Versions of the RL78/G12) 21
2.4.3 Connection between the 14-Pin Connector and the RL78 Family MCU (Only the RL78/I1C) 22
2.5 Notes on Connection 25
2.5.1 RESET# Pin 25
2.5.2 TOOL0 Pin 28
2.5.3 GND 29
2.5.4 VDD 29
2.6 Internal Circuits of the Emulator 31
2.6.1 Internal Circuits of the E1 (when the RL78 Family is Connected) 31
2.6.2 Internal Circuits of the E20 (when the RL78 Family is Connected) 32
2.6.3 Internal Circuits of the E2 (when the RL78 Family is Connected) 33
2.6.4 Internal Circuits of the E2 Lite (when the RL78 Family is Connected) 35
2.7 Notes on Designing the User System 36
2.7.1 Isolator for the E1 36
2.7.2 Small Connector Conversion Adapter for the E1 36

3. Notes on Usage 38
3.1 Turning the Power On/Off 38
3.1.1 When a Separate Power Supply is Used for the User System 38
3.1.2 When Power is Supplied to the User System from the Emulator (E1/E2/E2 Lite Only) 39
3.2 Power Supply Function of the E1/E2/E2 Lite 39
3.3 MCU Resources to be Occupied 40
3.3.1 Securing an Area for the Debugging Monitor Program 41
3.3.2 Securing a Stack Area for Debugging 42
3.3.3 Setting an On-Chip Debugging Option Byte 43
3.3.4 Setting a Security ID 44
3.3.5 Setting a Serial Programming Security ID 45
3.4 Reset 46
3.4.1 Operation after a Reset 46
3.4.2 SP Value after a Reset 46
3.4.3 Internal Reset caused by illegal instruction 46
3.5 Flash Memory 46
3.5.1 Flash Memory Programming by Self-Programming 46
3.5.2 Operation for Voltages and Flash Operation Modes Not Permitting Flash Memory Rewriting 47

3.6 GDIDIS 47
3.7 RESET# Multiplexed Pin 47
3.8 MCUs that are Used in Debugging 48
3.8.1 Usage in Mass-Production 48
3.8.2 Standalone Operation 48
3.9 Final Evaluation of the User Program 48
3.10 Debug Functions 49
3.10.1 Stepped Execution 49
3.10.2 [Go to Here] 49
3.10.3 Debugging in Standby Mode 49
3.10.4 Pseudo-Real-Time RAM Monitor Function or Pseudo-Dynamic Memory Modification Function 50
3.10.5 Start/Stop Functions 50
3.10.6 Emulation of Flash Memory CRC Accumulator Function 51
3.10.7 Break Function 51
3.10.8 Setting and Deleting Events during User Program Execution 51
3.10.9 Trace Function 51
3.10.10 Battery Backup Function 51
3.10.11 RAM-ECC Function 51
3.10.12 Extended Functions of the E2 52
3.10.13 Points for Caution on Using the RL78/G2x and RL78/L2x 52
3.10.14 Points for Caution on Using the RL78/F2x 54
3.10.15 Note on self-programming for flash-memory 56
3.10.16 State of high-speed on chip oscillator during on-chip debugging 56
3.10.17 Setting of download mode (RL78/FGIC[RAJ240100]) 56
3.10.18 Stopped address by hardware break 56
3.10.19 Setting for Using Bank Swap with BTBLS 57
3.10.20 Precautions during Bank Programming Mode 57
3.10.21 Open-source software used 57

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 Renesas Flash Programmer that can be used with the E1, E20, 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., RESET#).

Extended function of the E2

This means an extended function which is available with the E2 emulator.

E2 expansion interface

This means the interface required for extended functions of the E2 emulator.

1. Overview

This document explains notes on usage when the E1, E20, and E2 emulators and the E2 emulator Lite are connected to RL78 devices.

1.1 Overview of E1/E20/E2 Emulator and E2 emulator Lite

In this document, we describe ‘E2 Emulator’ as ‘E2’, ‘E1 Emulator’ as ‘E1’, ‘E20 Emulator’ as ‘E20’ and ‘E2 emulator Lite’ as ‘E2 Lite’.

The E1, E20, E2, and E2 Lite are on-chip debugging emulators for Renesas' mainstream MCUs.

The E1 and E2 Lite are highly affordable development tools providing basic debugging functions. The E20 is a development tool allowing sophisticated debugging through enhanced functions such as tracing and RAM monitoring as well as the basic debugging functions of the E1 and E2 Lite. The E2 is a development tool providing a solution for the tuning of current consumption and an external trigger function as extended functions beyond the basic debugging functions of the E1 and E2 Lite. The E1/E20/E2/E2 Lite can also serve as a Flash Programmer.

1.2 Note on Using E20

The large trace function and the real-time RAM monitoring function, which are the primary features of the E20, cannot be used with RL78 devices.

The available functions are equivalent to those of the E1/E2 Lite (only the internal trace function in the MCU and memory reference and modification during execution).

The power supply function from the E20 is not supported.

1.3 Configuration of E1/E20/E2/E2 Lite Manuals

The E1, E20, E2, and E2 Lite manuals consist of the following.

- E2 Emulator Manual
- E1/E20 Emulator User's Manual
- E2 Emulator Lite Manual
- E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for Manual

Be sure to read each user's manual before using the E1, E20, E2 or E2 Lite.

When you use extended functions of the E2, be sure to read the application notes for those functions.

(1) E2 Emulator Manual

The E2 emulator manual has the following contents:

— Components of the E2
— Hardware specifications of the E2
— Connection to the E2 and the host machine and user system

(2) E1/E20 Emulator User's Manual

The E1/E20 emulator user's manual has the following contents:

— Components of the E1/E20
— Hardware specifications of the E1/E20
— Connection to the E1/E20 and the host machine and user system

(3) E2 Emulator Lite Manual

The E2 Emulator Lite manual has the following contents:

— Components of the E2 Lite
— Hardware specifications of the E2 Lite
— Connection to the E2 Lite and the host machine and user system

(4) E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (this document)

The E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) describes information necessary for hardware design such as connection examples, interface circuits, and notes on using the emulator.

(5) Manual and help for the emulator debugger

The manual and help for the emulator debugger describe the functions of the E1/E20/E2/E2 Lite emulator debugger and the operating instructions.

Refer to the following.

— CS+ Integrated Development Environment User's Manual: RL78 Debug
— Help for e^2 studio
— RL78 Family CS+ Debugging Using Hot Plug-in Function

When using C-SPY made by IAR Systems, also refer to "IAR C-SPY Hardware Debugger System User Guide issued by IAR Systems" published by IAR Systems.

(6) Application note for extended functions of the E2

The application note for extended functions of the E2 includes an explanation and descriptions of operation methods for an extended function of the E2 emulator. Refer to the Application Note for the Solution for the Tuning of Current Consumption.

1.4 Supported Devices

Table 1-1 Supported Device List

Item	Description		
Target MCUs *2	RL78 Family (RL78-S2 Core) *1		
		RL78/D1x:	RL78/D1A
		RL78/F1x:	RL78/F12
		RL78/G1x:	RL78/G12, RL78/G13, RL78/G1A, RL78/G1C, RL78/G1D, RL78/G1E, RL78/G1P, RL78/G13A
		RL78/I1x:	RL78/I1A
		RL78/L1x:	RL78/L12, RL78/L13
	RL78 Family (RL78-S3 Core) *1		
		RL78/F1x:	RL78/F13, RL78/F14, RL78/F15, RL78/F1A
		RL78/G1x:	RL78/G11, RL78/G14, RL78/G1H, RL78/G1F, RL78/G1G
		RL78/I1x:	RL78/I1B, RL78/I1C, RL78/I1D, RL78/I1E
		RL78/H1x:	RL78/H1D
		RL78/L1x:	RL78/L1A, RL78/L1C
		RL78/FGIC	
	*3	RL78/G2x:	RL78/G21, RL78/G22, RL78/G23, RL78/G24
	*3	RL78/F2x:	RL78/F22, RL78/F23, RL78/F24, RL78/F25
	*3	RL78/L2x:	RL78/L23

Notes: 1. For details on the RL78-S2 core or RL78-S3 core, refer to "RL78 Family User's Manual: Software".
2. When using the RL78/G10, RL78/G1M, or RL78/G1N, refer to "E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (R20UT2937)".
3. The E1 and E20 emulators do not support the RL78/G2x MCU. Use the E2 emulator or E2 emulator Lite.

Table 1-2 Debugging Function List

Item		Type of the RL78 Core		Description
		RL78-S2	RL78-S3	
Memory reference or change during program execution				
	Pseudo-real-time RAM monitor (RRM)	Supported	Supported	CPU is used when monitoring
	Dynamic Memory Modification (DMM)	Supported	Supported	CPU is used when changing
Event		1 point	2 points (max)	Can be used for hardware break (or trace*1)
Break	Software break	Supported	Supported	2000 points
	Hardware break	Supported	Supported	Execution address or data access
	Forced break	Supported	Supported	
Trace	Acquired information	Not supported	Supported *2	Branch source PC information
	Start event	Not supported	Supported *2	User program execution start and event start
	End event	Not supported	Supported *2	User program stop, event completion, and full trace memory
Performance measurement function	Measurement item	Supported	Supported	Between start and stop of user program execution
	Performance	Supported	Supported	Resolution: 100 μs Maximum measurement time: 119 hours 18 min
Hot plug-in		Not supported	Supported *3	
Coverage measurement function		Not supported	Not supported	
Extended function of the E2	Solution for the tuning of current consumption	Supported	Supported	Current consumption by the user system is measured when the power is supplied from the E2.
	External trigger input/output	Supported	Supported	External triggers can be input to or output from the E2 expansion interface.

Notes: 1. Only devices in which the trace function was implemented.
2. The trace function is not supported in an RL78/G14 MCU whose ROM size is 64 Kbytes or less. The trace function is also not supported for RL78/G11, RL78/G1G, and RL78/FGIC MCUs.
3. Only RL78/F1x series(except F12), RL78/F2x

1.5 Overview of the Specifications of the External Trigger Input and Output

This section describes the specifications of the external trigger input and output, which is one of extended functions of the E2.

1.5.1 Assignments of the External Trigger Input and Output Pins

The following figure and table show the assignments of the external trigger input and output pins for the E2 expansion interface.

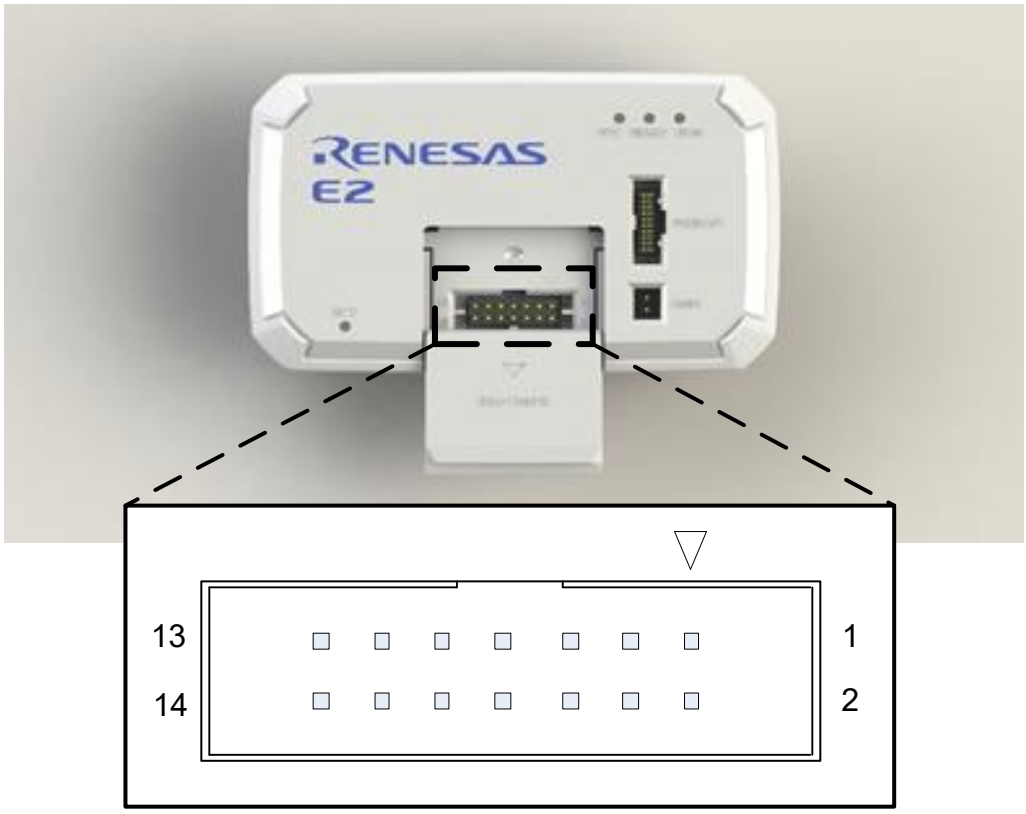


Figure 1-1 E2 Expansion Interface

Table 1-3 Assignments of the External Trigger Input and Output Pins for the E2 Expansion Interface

Pin No.	Input/Output	Description
1	—	Pin Nos. 1 to 8 are not used. These pins must be left open-circuit.
2	—	
3	—	
4	—	
5	—	
6	—	
7	—	
8	—	
9	Output	External trigger output (ch. 0)
10	Output	External trigger output (ch. 1)
11	Input	External trigger input (ch. 0)
12	Input	External trigger input (ch. 1)
13	—	GND
14	Output	A pin for output of the power-supply voltage for the E2 expansion interface (1.8 V to 5.0 V)

1.5.2 Specifications of the External Trigger Inputs and Outputs

Table 1-4 Specifications of the External Trigger Inputs and Outputs

Input signal channels	E2 expansion interface: 2 (ch. 0: pin 11, ch. 1: pin 12)
Output signal channels	E2 expansion interface: 2 (ch. 0: pin 9, ch. 1: pin 10)
Voltage of the E2 expansion interface	• When the power is supplied from the emulator: voltage to be supplied (any voltage from 1.8 V to 5.0 V, as specified in the debugger) • When the power is not supplied from the emulator: VDD voltage
Conditions for detecting an external trigger input	Detecting edges (rising, falling, or both) Detecting a level (low or high)
Operation during the input of an external trigger	Break* Recording the data from measuring the current consumption while the low or high level is being input
Condition for detecting an external trigger output	Detecting a break or a trigger condition for measuring the current consumption
Operation during the output of an external trigger	When a break is detected, a low- or high-level pulse is output (the pulse width can be set to times in the range from 1 μs to 65535 μs). When a trigger condition for measuring the current consumption is detected, a high-level pulse is output (the pulse width can be set to times in the range from 1 μs to 65535 μs). Otherwise, a high level is output while a condition is being satisfied.

Note: About 12 μs will be required from the detection of a condition for an external trigger input being satisfied until the user program is stopped.

2. Designing the User System

2.1 Connecting the E1/E20/E2/E2 Lite to the User System

To connect the E1/E20/E2/E2 Lite to the user system, 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 MCU in use.

2.2 Installing the Connector on the User System

Table 2-1 shows the recommended connectors for the E1/E20/E2/E2 Lite.

Table 2-1 Recommended Connectors

Connector	Type Number	Manufacturer	Specifications
14-pin connector	7614-6002	3M Japan Limited	14-pin straight type (Japan)
	2514-6002	3M	14-pin straight type (other countries)

Note: Connection to the 38-pin connector of the E20 is not supported. To use the E20, use the 38-pin to 14-pin conversion adapter that comes with the E20 for connection to the 14-pin connector on the user system.

2.2.1 Connecting the User System Interface Cable to the 14-Pin Connector

The following figures show how to connect the user system interface cable to the 14-pin connector of each emulator.

E1/E2 Lite: Figure 2-1

(Connecting the User System Interface Cable to the 14-Pin Connector of the E1/E2 Lite Emulator)

E20: Figure 2-2

(Connecting the User System Interface Cable to the 14-Pin Connector of the E20 Emulator)

E2: Figure 2-3

(Connecting the User System Interface Cable to the 14-Pin Connector of the E2 Emulator)

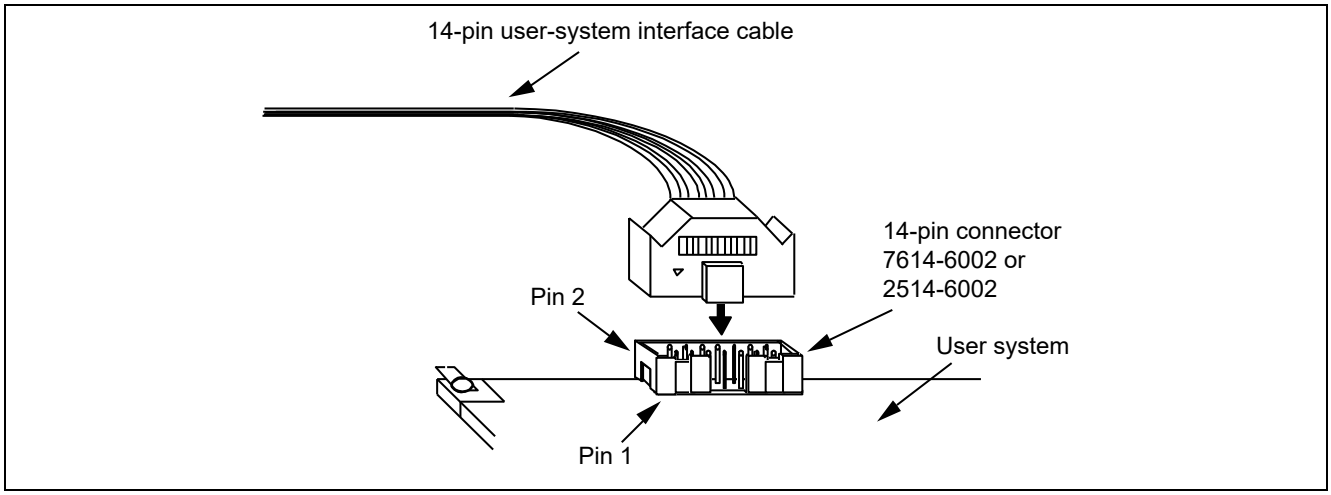


Figure 2-1 Connecting the User System Interface Cable to the 14-Pin Connector of the E1/E2 Lite Emulator

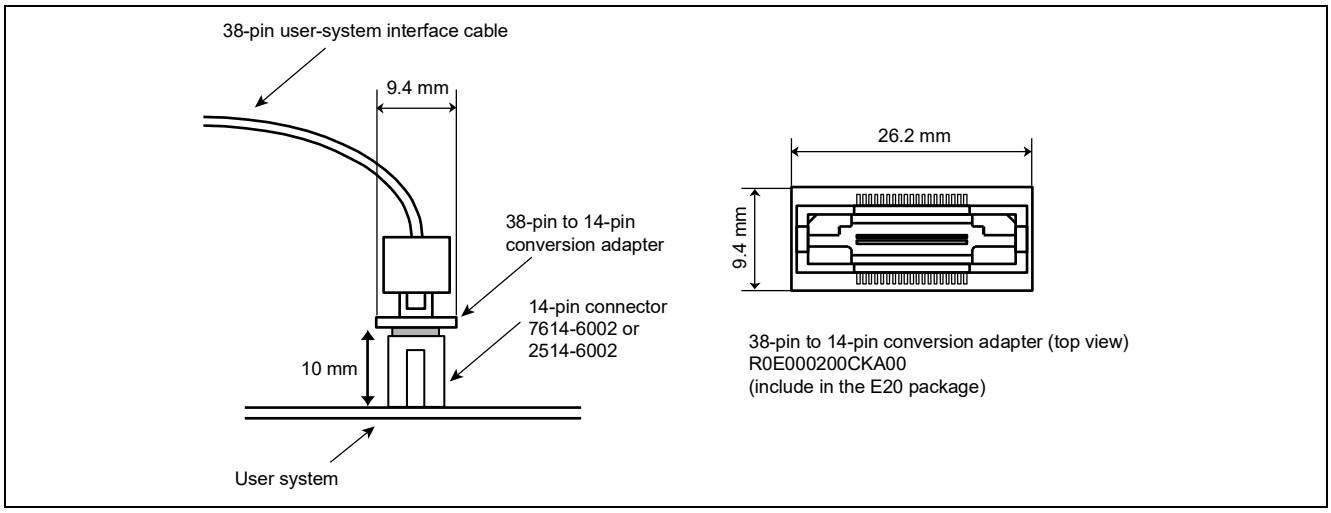


Figure 2-2 Connecting the User System Interface Cable to the 14-Pin Connector of the E20 Emulator

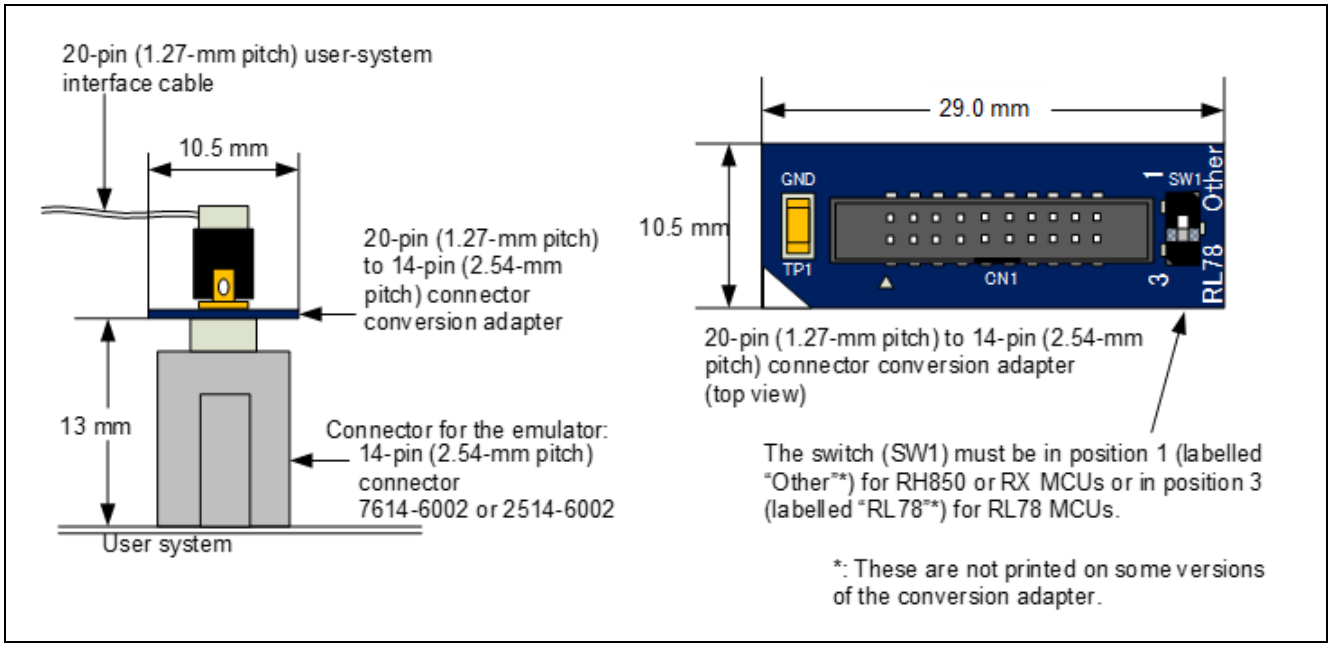
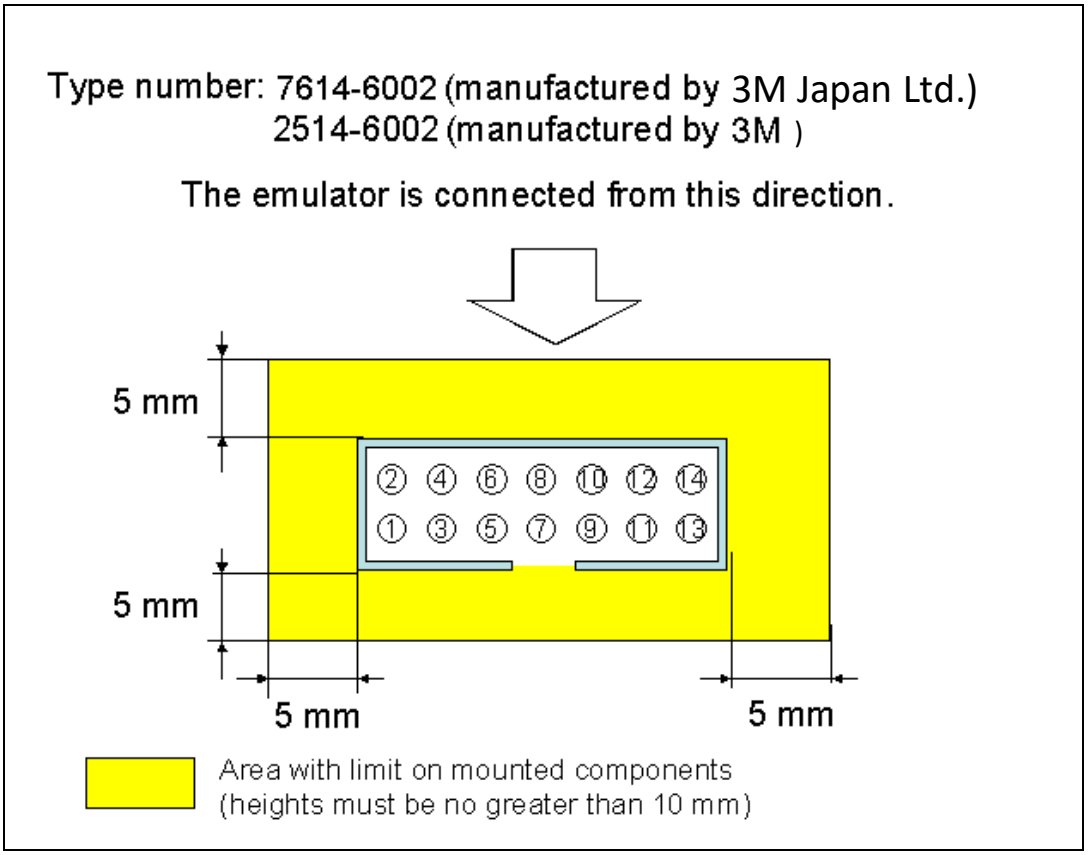


Figure 2-3 Connecting the User System Interface Cable to the 14-Pin Connector of the E2 Emulator

CAUTION

Limit to the height on connector periphery:

For a case where the R0E000200CKA00 is used for connecting the E20 to a 14-pin connector, do not mount other components with a height of 10 mm or more within 5 mm of the connector on the user system.

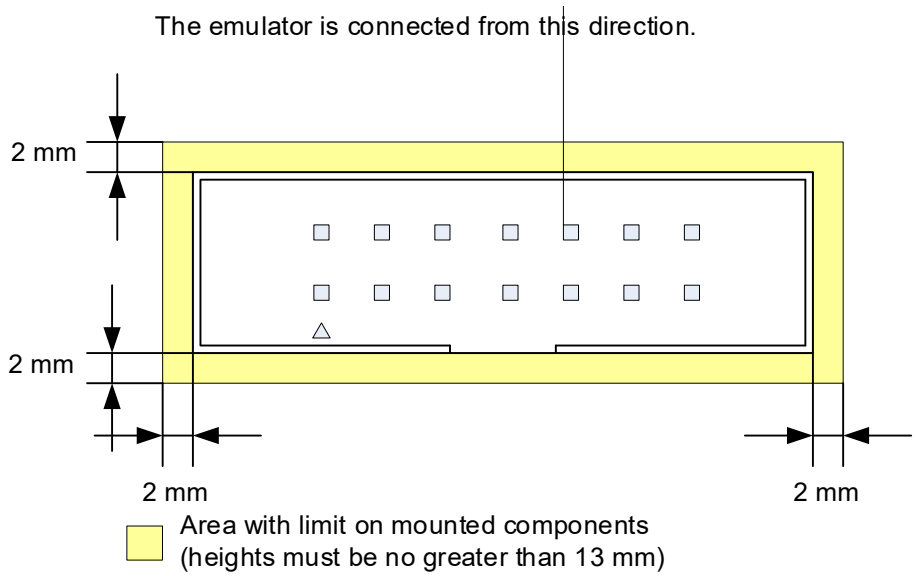


CAUTION

Limit to the height on connector periphery:

If the 20-pin (1.27-mm pin spacing) to 14-pin (2.54-mm pin spacing) connector conversion adapter is in use and the E2 is to be connected to the 14-pin connector, do not mount components with a height of 13 mm or more on the user system within a rectangle 2 mm from the edges of the connector.

Type number: 7614-6002 (manufactured by 3M Ltd.) (Japan)
: 2514-6002 (manufactured by 3M) (other countries)



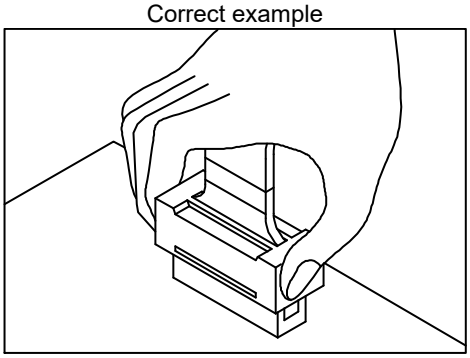
CAUTION

Notes on connector insertion and removal:

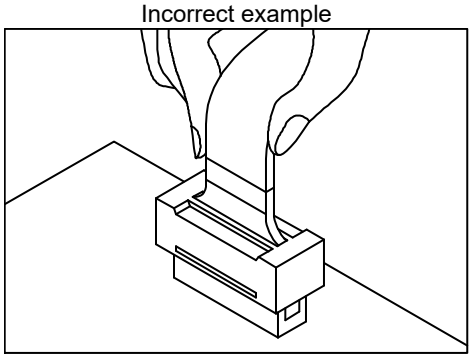
When connecting or disconnecting the user-system interface cable and the emulator or user system, grasp the connector cover at the end of the cable. Pulling the cable itself will damage the wiring.

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 14-Pin Connector Specifications

Figure 2-4 shows the specifications of the 14-pin connector.

Table 2-2 on the following pages shows the pin assignments of the 14-pin connector.

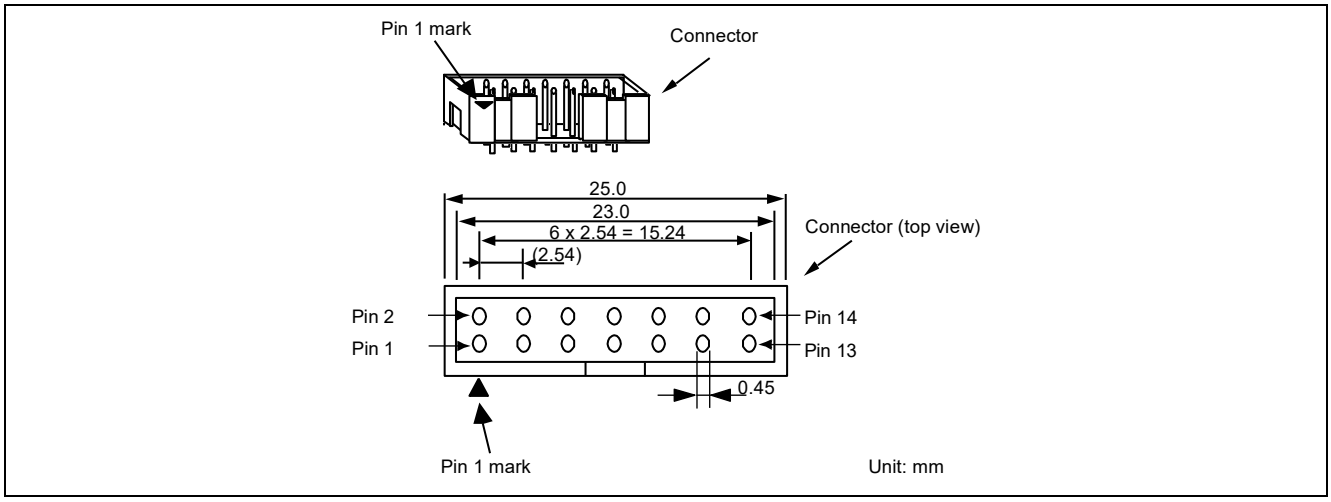


Figure 2-4 Emulator Connector Specifications (14 Pins)

Table 2-2 14-Pin Connector Pin Assignments

Pin No.	RL78 family MCUs in general *7		Only for the RL78/G11, the RL78/I1C, the RL78/G21, and 20- and 24-pin versions of the RL78/G12		Note
	Signal *1	Direction *2	Signal *1	Direction *2	
1	R.F.U *5	—	R.F.U *5	—	
2	GND *3	—	GND *3	—	
3	R.F.U *5	—	R.F.U *5	—	
4	R.F.U *5	—	RSTPU	Input	This pin is used to pull up the reset line (only when the RL78/G11, the RL78/I1C, the RL78/G21, and 20- and 24-pin versions of the RL78/G12 are in use).
5	TOOL0	I/O	TOOL0	I/O	This pin is used to transmit command/data to the target device.
6	RESET_IN	Output	RESET_IN	Output	This pin is used to input a reset signal from the user system.
7	R.F.U *5	—	R.F.U *5	—	
8	VDD	—	VDD	—	
9	EMVDD *6	—	EMVDD *6	—	
10	RESET_OUT *4	Input	RESET_OUT *4	Input	This pin is used to output a reset signal to the target device.
11	R.F.U *5	—	R.F.U *5	—	
12	GND *3	—	GND *3	—	
13	RESET_OUT *4	Input	RESET_OUT *4	Input	This pin is used to output a reset signal to the target device.
14	GND *3	—	GND *3	—	

*1 For details on the programming software, refer to the following.
https://www.renesas.com/RFP

*2 "Input" refers to input from the emulator to the user system and "output" refers to output from the user system to the emulator.

*3 Securely connect pins 2, 12, and 14 of the connector to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite.

*4 Securely connect both pin 10 and pin 13.

*5 This pin is reserved. Perform the open processing.

*6 Connect the driving power supply for the TOOL0 pin to EMVDD.
Please connect VDD when the MCU does not have power supplies other than VDD such as EVDD.
The E2 Lite only supports a single power supply. If you are using an MCU that requires two or more power supplies with the E2 Lite, use a power supply other than VDD, such as EVDD, which has the same voltage as VDD.

*7 This indicates RL78 family MCUs other than the RL78/G11, the RL78/I1C, the RL78/G21, and 20- and 24-pin versions of the RL78/G12.

2.4 Recommended Circuits between the Connector and the MCU

This section shows recommended circuits for connection between the connector and the MCU when the E1/E20/E2/E2 Lite is in use. For processing of signals, refer to section 2.5, Notes on Connection.

2.4.1 Connection between the 14-Pin Connector and the RL78 Family MCUs in General

Figure 2-5 shows a recommended circuit for connection between the 14-pin connector and the RL78 family MCUs in general.

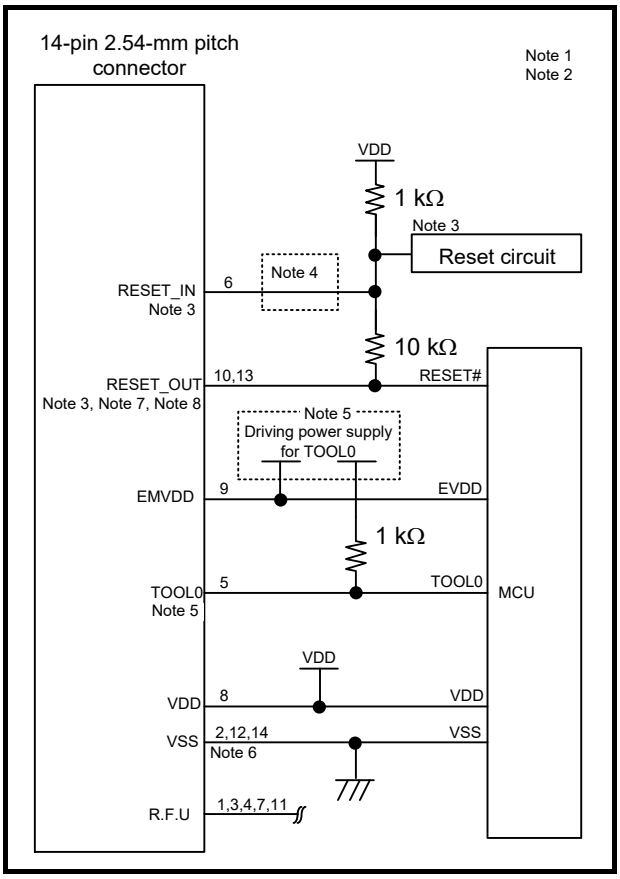


Figure 2-5 Example of Connection between the 14-Pin Connector and the RL78 Family MCUs in General

Notes:

1. The circuits and resistance values listed are recommended but not guaranteed. Determine the circuit design and resistance values by taking into account the specifications of the target device and noise. For flash programming for mass production, perform sufficient evaluation about whether the specifications of the target device are satisfied.
2. For processing of pins not used by the E1/E20/E2/E2 Lite, refer to the hardware manual for the device.
3. For details on how to handle the RESET# pin, refer to section 2.5, Notes on Connection.
4. The RESET_IN pin is used only in debugging. The emulator operates for flash programming by the programming software whether this pin is connected or not.
5. The driving power supply for the TOOL0 pin is different depending on devices. Refer to the user's manual for the device.

   Please connect VDD when the MCU does not have power supplies other than VDD such as EVDD.
6. Securely connect pins 2, 12, and 14 of the connection to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite.
7. Securely connect both pin 10 and pin 13.
8. When you use hot plug-in, install a ceramic capacitor (approx. 0.1 μF) between the RESET# pin and GND in order to suppress a noise to the RESET# pin that would occur when the emulator is connected.

2.4.2 Connection between the 14-Pin Connector and the RL78 Family MCUs (Only the RL78/G11, the RL78/G21, and the 20- and 24-Pin Versions of the RL78/G12)

Figure 2-6 shows a recommended circuit for connection between the 14-pin connector and the RL78 family MCUs (only the RL78/G11, the RL78/G21, and the 20- and 24-pin versions of the RL78/G12).

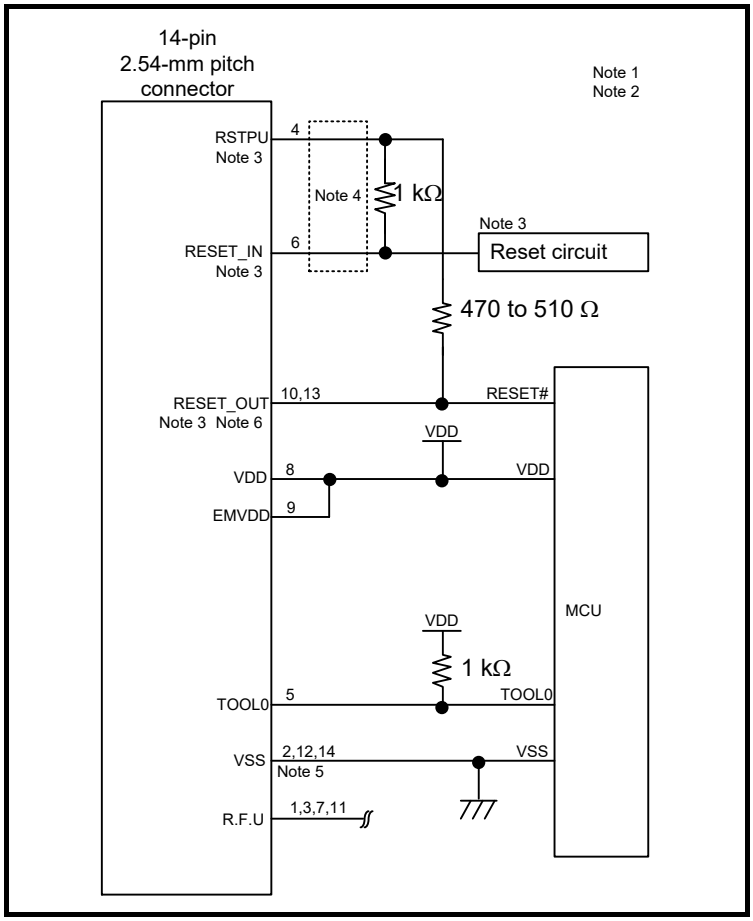


Figure 2-6 Example of Connection between the 14-Pin Connector and the RL78 Family MCUs (Only the RL78/G11, the RL78/G21, and the 20- and 24-Pin Versions of the RL78/G12)

Notes:

1. The circuits and resistance values listed are recommended but not guaranteed. Determine the circuit design and resistance values by taking into account the specifications of the target device and noise. For flash programming for mass production, perform sufficient evaluation about whether the specifications of the target device are satisfied.
2. For processing of pins not used by the E1/E20/E2/E2 Lite, refer to the hardware manual for the device.
3. The recommended circuit for the RESET# pin differs depending on whether the multiplexed functions are used. For details on how to handle the RESET# pin, also refer to section 2.5, Notes on Connection.
4. The RESET_IN and RSTPU pins are used only in debugging. The emulator operates for flash programming by the programming software whether this pin is connected or not.
5. Securely connect pins 2, 12, and 14 of the connection to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite.
6. Securely connect both pin 10 and pin 13.

2.4.3 Connection between the 14-Pin Connector and the RL78 Family MCU (Only the RL78/I1C)

The following shows a recommended circuit for connection between the 14-pin connector and the RL78 family MCU (only the RL78/I1C).

Figure 2-7 shows the case when the battery backup function is in use.

Figure 2-8 shows the case when the battery backup function is not in use.

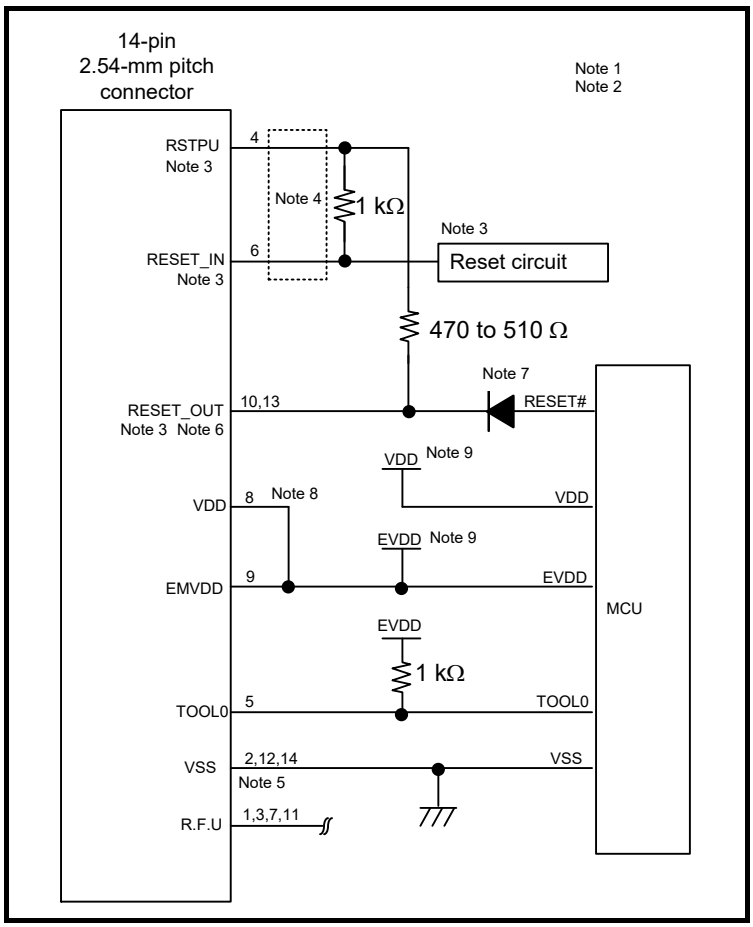


Figure 2-7 Example of Connection when the Battery Backup Function of the RL78/I1C is in Use

Notes:

1. The circuits and resistance values listed are recommended but not guaranteed. Determine the circuit design and resistance values by taking into account the specifications of the target device and noise. For flash programming for mass production, perform sufficient evaluation about whether the specifications of the target device are satisfied.
2. For processing of pins not used by the E1/E20/E2/E2 Lite, refer to the hardware manual for the device.
3. The recommended circuit for the RESET# pin differs depending on whether the multiplexed functions are used. For details on how to handle the RESET# pin, also refer to section 2.5, Notes on Connection.
4. The RESET_IN and RSTPU pins are used only in debugging. The emulator operates for flash programming by the programming software whether this pin is connected or not.
5. Securely connect pins 2, 12, and 14 of the connection to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite.
6. Securely connect both pin 10 and pin 13.
7. We recommended that you use a diode of the BAT54 series from NXP Semiconductors.
8. When the battery backup function is in use, connect VDD of the 14-pin connector to EVDD of the target MCU. Do not connect VDD of the target MCU to the 14-pin connector.

   Do not disconnect the EVDD power supply while debugging involves the battery backup function.
9. Depending on the number of pins of the target MCU, VDD and EVDD may be the same pin. In such cases, connect both VDD and EMVDD of the 14-pin connector to the power pin of the target MCU.

   Do not disconnect this power supply while debugging involves the battery backup function.

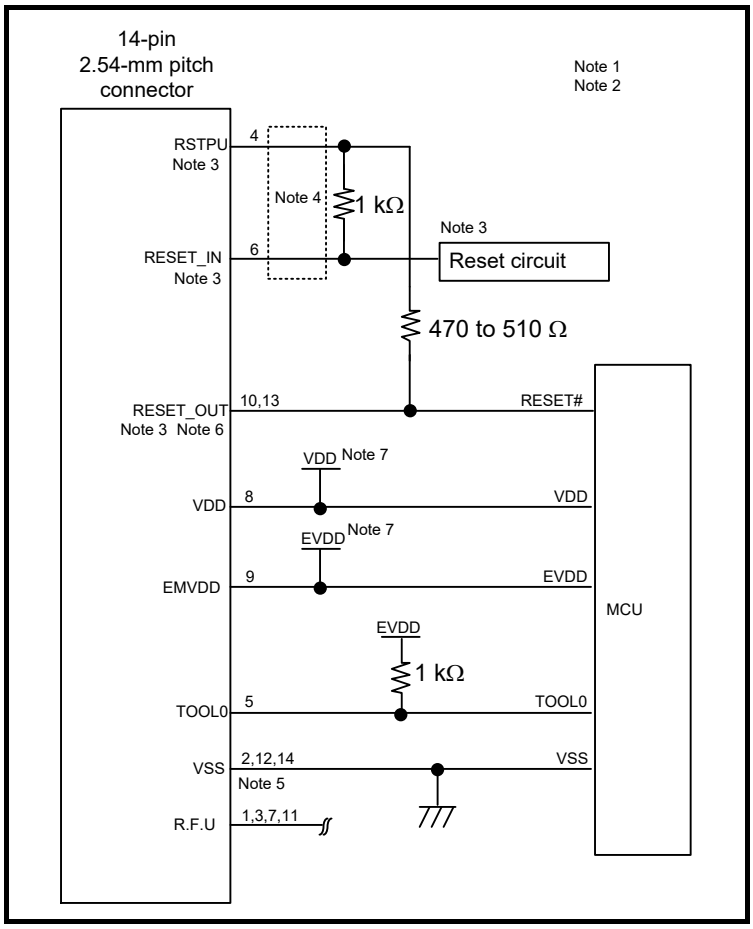


Figure 2-8 Example of Connection when the Battery Backup Function of the RL78/I1C is not in Use

Notes:

1. The circuits and resistance values listed are recommended but not guaranteed. Determine the circuit design and resistance values by taking into account the specifications of the target device and noise. For flash programming for mass production, perform sufficient evaluation about whether the specifications of the target device are satisfied.
2. For processing of pins not used by the E1/E20/E2/E2 Lite, refer to the hardware manual for the device.
3. The recommended circuit for the RESET# pin differs depending on whether the multiplexed functions are used. For details on how to handle the RESET# pin, also refer to section 2.5, Notes on Connection.
4. The RESET_IN and RSTPU pins are used only in debugging. The emulator operates for flash programming by the programming software whether this pin is connected or not.
5. Securely connect pins 2, 12, and 14 of the connection to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite.
6. Securely connect both pin 10 and pin 13.
7. Depending on the number of pins of the target MCU, VDD and EVDD may be the same pin. In such cases, connect both VDD and EMVDD of the 14-pin connector to the power pin of the target MCU.

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 E1/E20/E2/E2 Lite is not in use, refer to the hardware manual for the MCU.

2.5.1 RESET# Pin

The RESET# pin is used by the E1/E20/E2/E2 Lite to monitor the pin state and issue a reset to the device.

Therefore, a reset signal on the user system is once input to the E1/E20/E2/E2 Lite where it is controlled by masking and then output to the target device. Connection examples of the RESET# pin section are shown in Table 2-3 and Figure 2-9 to Figure 2-14.

When flash programming by the programming software is to be performed, the RESET# pin should be designed so that the reset signal on the user system does not conflict with the reset signal from the E1/E20/E2/E2 Lite.

When you use hot plug-in, install a ceramic capacitor (approx. 0.1 μF) between the RESET# pin and GND in order to suppress a noise to the RESET# pin that would occur when the emulator is connected.

Hot plug-ins are not available if there is no reset circuit.

Table 2-3 Connection Examples According to the Reset Circuit on the User System

No.	Target Device	Case	Reference
1	RL78 family MCUs in general	There is a reset circuit on the user system. (Resistors are used to switch between usage and non-usage of the emulator.)	Figure 2-9
2		There is a reset circuit on the user system. (A jumper is used to switch between usage and non-usage of the emulator.)	Figure 2-10
3		The power-on reset circuit is the only reset circuit on the user system.	Figure 2-11
4	RL78/G11, 20- and 24-pin versions of the RL78/G12 RL78/G21	The pin on which the RESET# function is multiplexed is in use for the reset function. There is no reset circuit if the section enclosed by the dotted lines described in note 2 for figure 2-12 is removed.	Figure 2-12
		The pin on which the RESET# function is multiplexed is in use for another function. (when using the P125/KR1/SI01 function which is multiplexed with RESET#)	Figure 2-13
5	RL78/I1C	The battery backup function is not in use. There is no reset circuit if the section enclosed by the dotted lines described in note 2 for figure 2-12 is removed.	Figure 2-12
6		The battery backup function is in use. There is no reset circuit if the section enclosed by the dotted lines described in note 2 for figure 2-14 is removed.	Figure 2-14

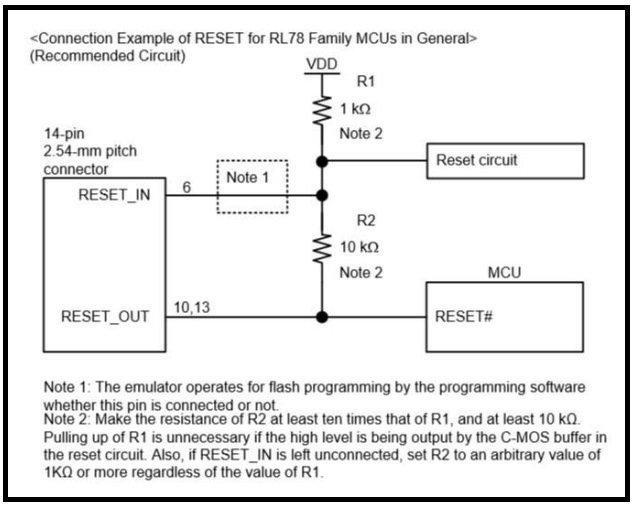


Figure 2-9 Connection Example 1 of RESET#
(RL78 Family MCUs in General)

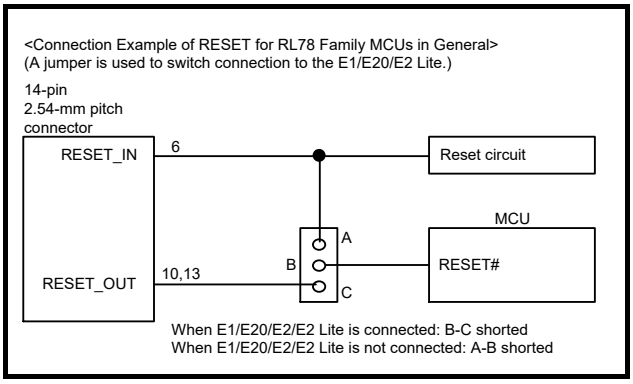


Figure 2-10 Connection Example 2 of RESET#
(RL78 Family MCUs in General)

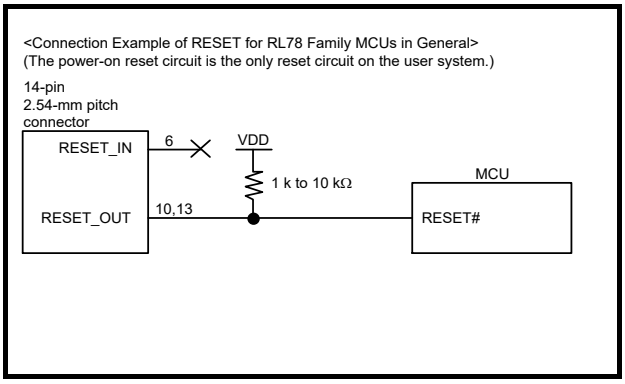


Figure 2-11 Connection Example 3 of RESET#
(RL78 Family MCUs in General)

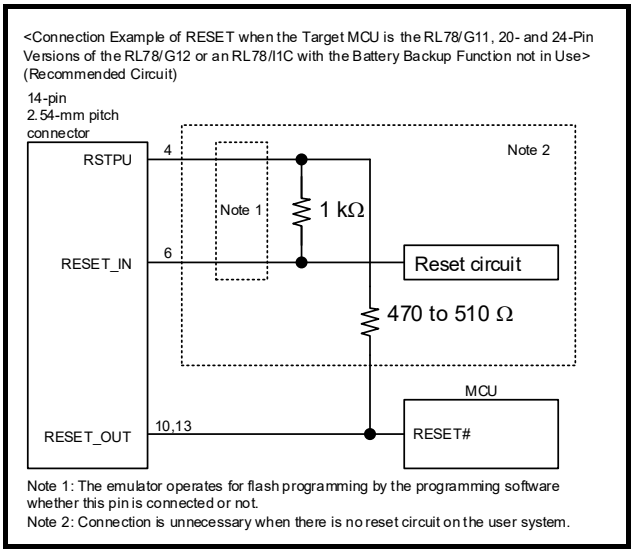


Figure 2-12 Connection Example 1 of RESET#
(when the Target MCU is the RL78/G11, 20- and 24-Pin Versions of the RL78/G12, the RL78/G21 or an RL78/I1C with the Battery Backup Function not in Use)

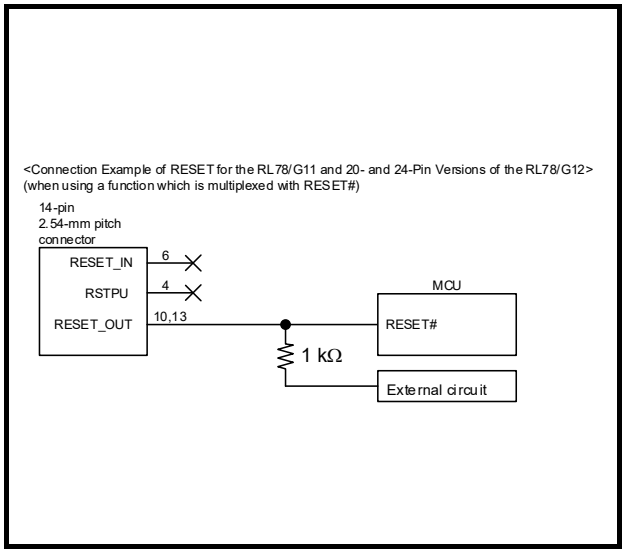


Figure 2-13 Connection Example 2 of RESET#
(the RL78/G11, the RL78/G21, and 20- and 24-Pin Versions of the RL78/G12)

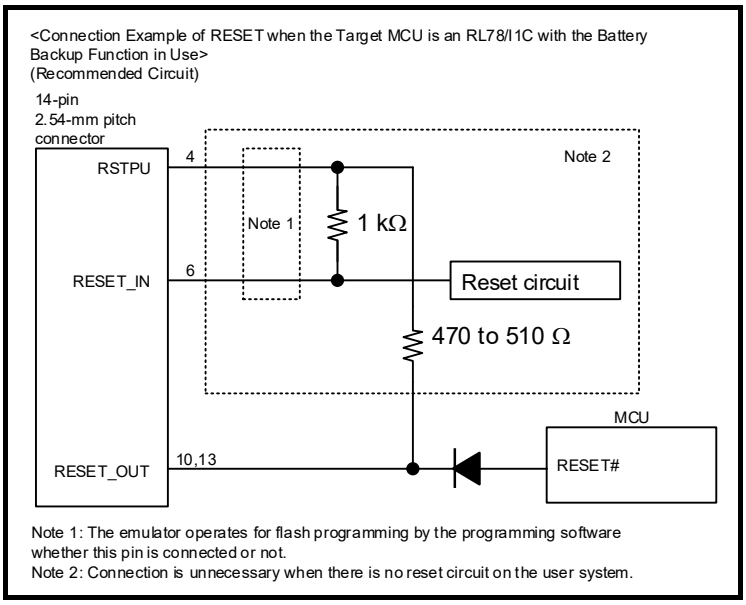


Figure 2-14 Connection Example of RESET#
(when the Target MCU is an RL78/I1C with the Battery Backup Function in Use)

The following describes points for caution regarding Figure 2-9 to Figure 2-14.

- Do not install capacitors, series resistors, or filters on signal lines; if attempted, correct communication may not be established. There is an exception, however: capacitors can be inserted between VSS and RESET_IN.
- The circuits and resistance values listed are recommended but not guaranteed. Determine the circuit design and resistance values by taking into account the specifications of the target device and noise.
- Securely connect pins 2, 12, and 14 to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2/E2 Lite. Securely connect both pin 10 and pin 13.

2.5.2 TOOL0 Pin

The E1/E20/E2/E2 Lite uses the TOOL0 pin. Any functions that are multiplexed on this pin are not available.

Pull up the signals of the TOOL0 pin at 1 kΩ and do not arrange these signal lines in parallel with or across other high-speed signal lines.

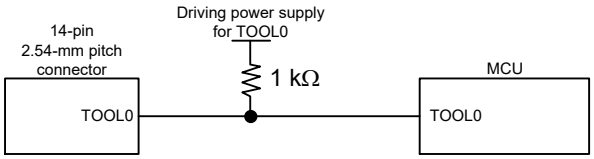


Figure 2-15 Connection Example of the TOOL0 Pin

Do not use adjacent resistors for pull-up of the TOOL0 pin because they may affect or may be affected from other pins.
Do not install capacitors, series resistors, or filters on signal lines; if attempted, correct communication may not be established.

2.5.3 GND

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

2.5.4 VDD

Connect the VDD of the connector to the VDD (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.

Note: With the E2 emulator, when the user system has a separate power supply, the emulator can be used within the range of power supply voltage from 1.6 V to 5.5 V and within the range of operating voltage for the MCU.

When power is supplied to the user system from other than the emulator, the E1/E20/E2/E2 Lite consumes the power supply for the last output and first input buffers of the emulator.

E1:	3.3 V: approximately 20 mA,	5.0 V: approximately 40 mA
E20:	3.3 V: approximately 40 mA,	5.0 V: approximately 100 mA
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 E1/E2/E2 Lite can supply power to a simple evaluation system.

E1: Can supply power of 3.3 V or 5.0 V, up to 200 mA.

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 E1/E2/E2 Lite, check the voltage supplied to the user system. Particularly when the 5.0-V supply option is selected, the voltage depends on the USB VBUS power-supply voltage.

E1: The voltage may drop by 0.5 V or more below 5.0 V.

E2: The voltage may drop by 0.3 V or more below 5.0 V.

The on-chip debugging circuit in the device operates during on-chip debugging. Therefore current consumption of the device increases. When correctly evaluating current consumption of the device, do not connect the E1/E20/E2/E2 Lite.

Power supply from the E1/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 E1/E2/E2 Lite. Use a stable, separate power supply for the user system. When writing a program for mass production processes, use the Renesas Flash Programmer.

For details on the programming software, refer to:

https://www.renesas.com/RFP

⚠ WARNING

Warning for Turning the Power On/Off:

When supplying power, ensure that there are no shorts between VDD and GND. Only connect the E1/E20/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 E1/E20/E2/E2 Lite, and the user system emitting smoke or catching fire.

2.6 Internal Circuits of the Emulator

2.6.1 Internal Circuits of the E1 (when the RL78 Family is Connected)

Figure 2-16 shows the internal circuits of the E1 with the RL78 family connected.

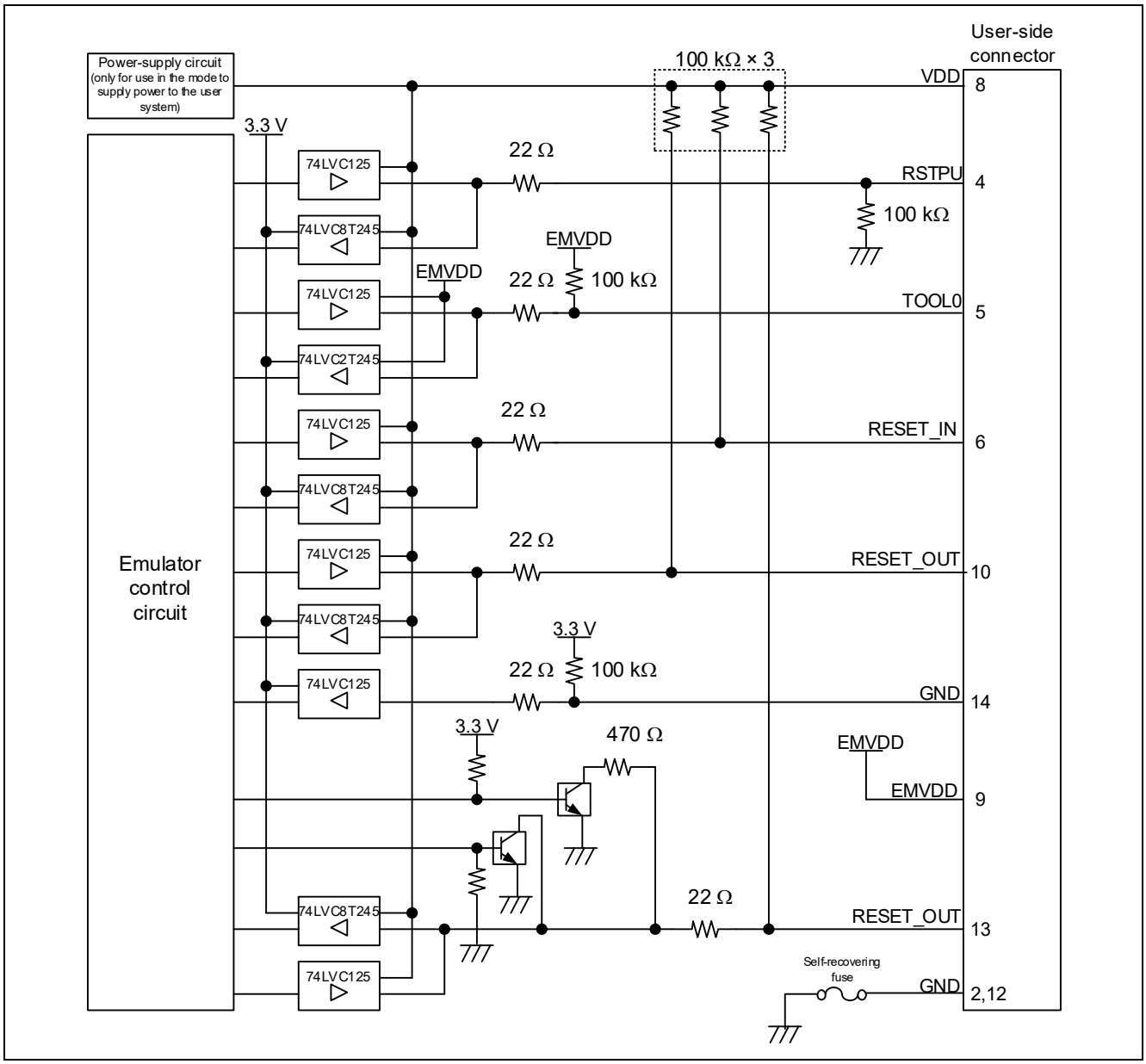


Figure 2-16 Internal Circuit of the E1 (when the RL78 Family is Connected)

Notes1: E1/E20 emulator doesn't support RL78/G2x, RL78/F2x, RL78/L2x.

2.6.2 Internal Circuits of the E20 (when the RL78 Family is Connected)

Figure 2-17 shows the internal circuits of the E20 with the RL78 family connected.

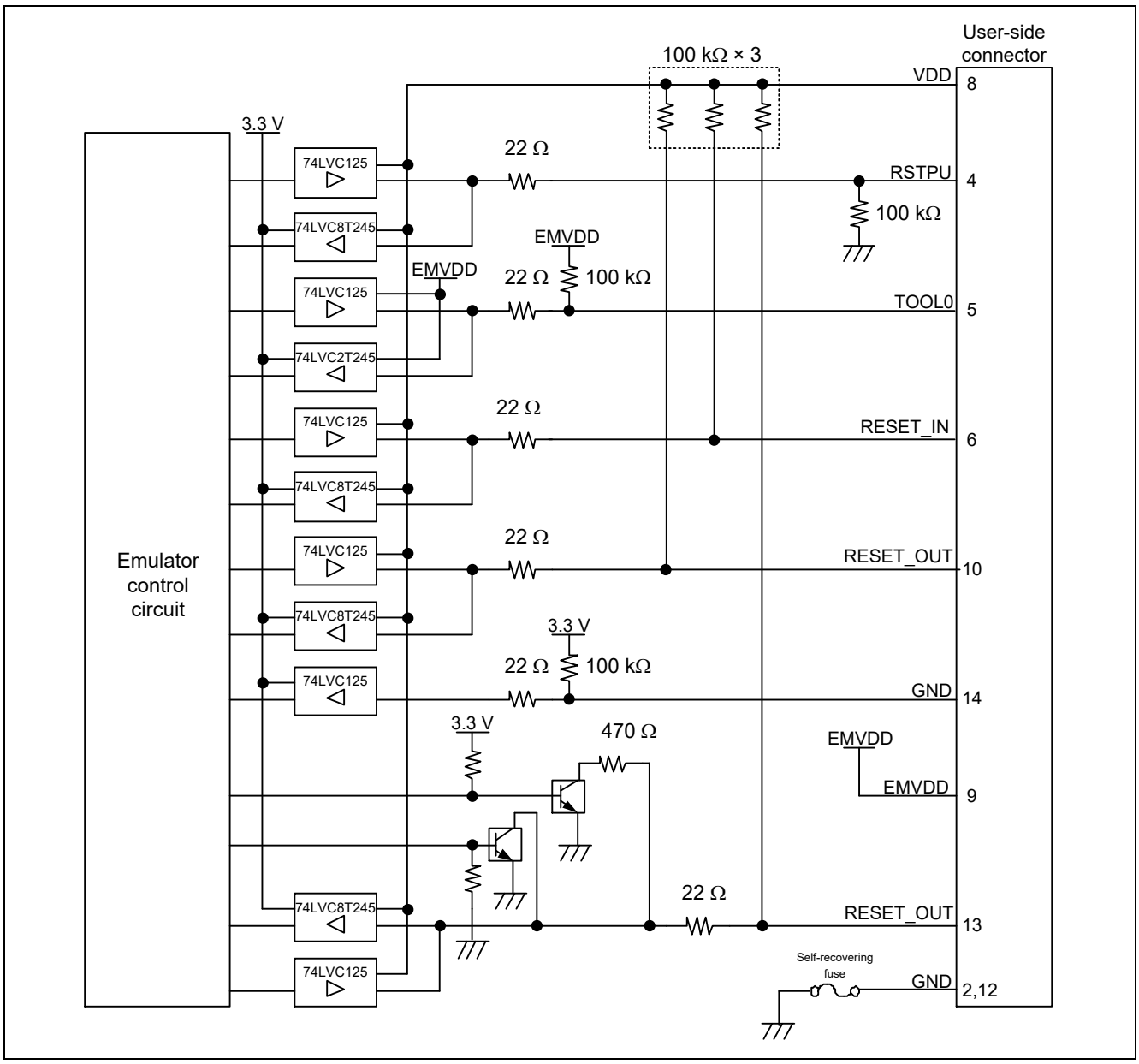


Figure 2-17 Internal Circuits of the E20 (RL78 Family)

Notes1: E1/E20 emulator doesn't support RL78/G2x, RL78/F2x, RL78/L2x.

2.6.3 Internal Circuits of the E2 (when the RL78 Family is Connected)

Figure 2-18 and Figure 2-19 respectively show the internal circuits of product revisions C and D of the E2 with the RL78 family connected.

The alphabet at the end of the serial No. written on the E2 main unit indicates the product revision.

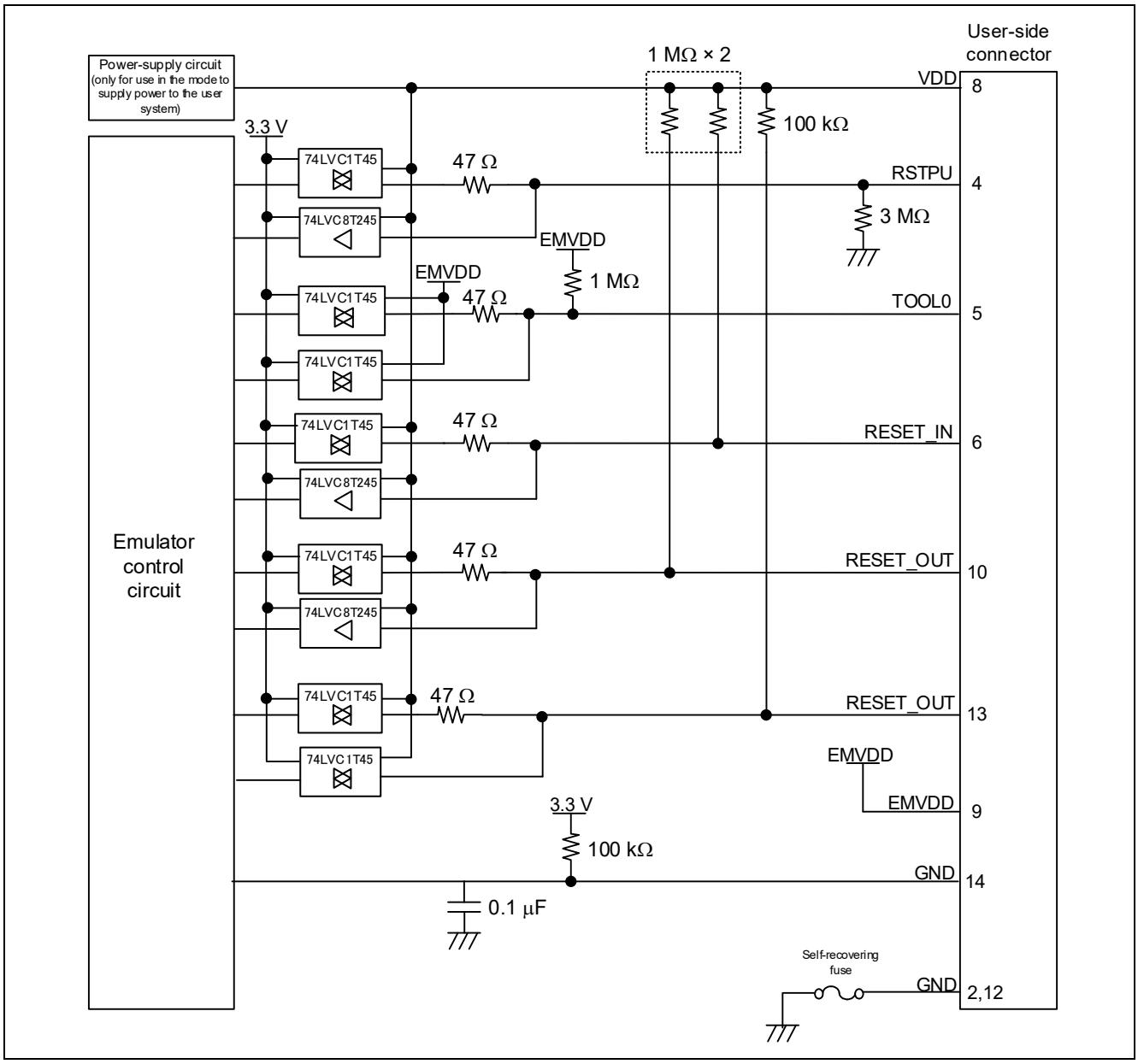


Figure 2-18 Internal Circuits of the E2 (Rev. C)

Set the switch on the 20-pin (1.27-mm pin spacing) to 14-pin (2.54-mm pin spacing) connector conversion adapter to position “3”.

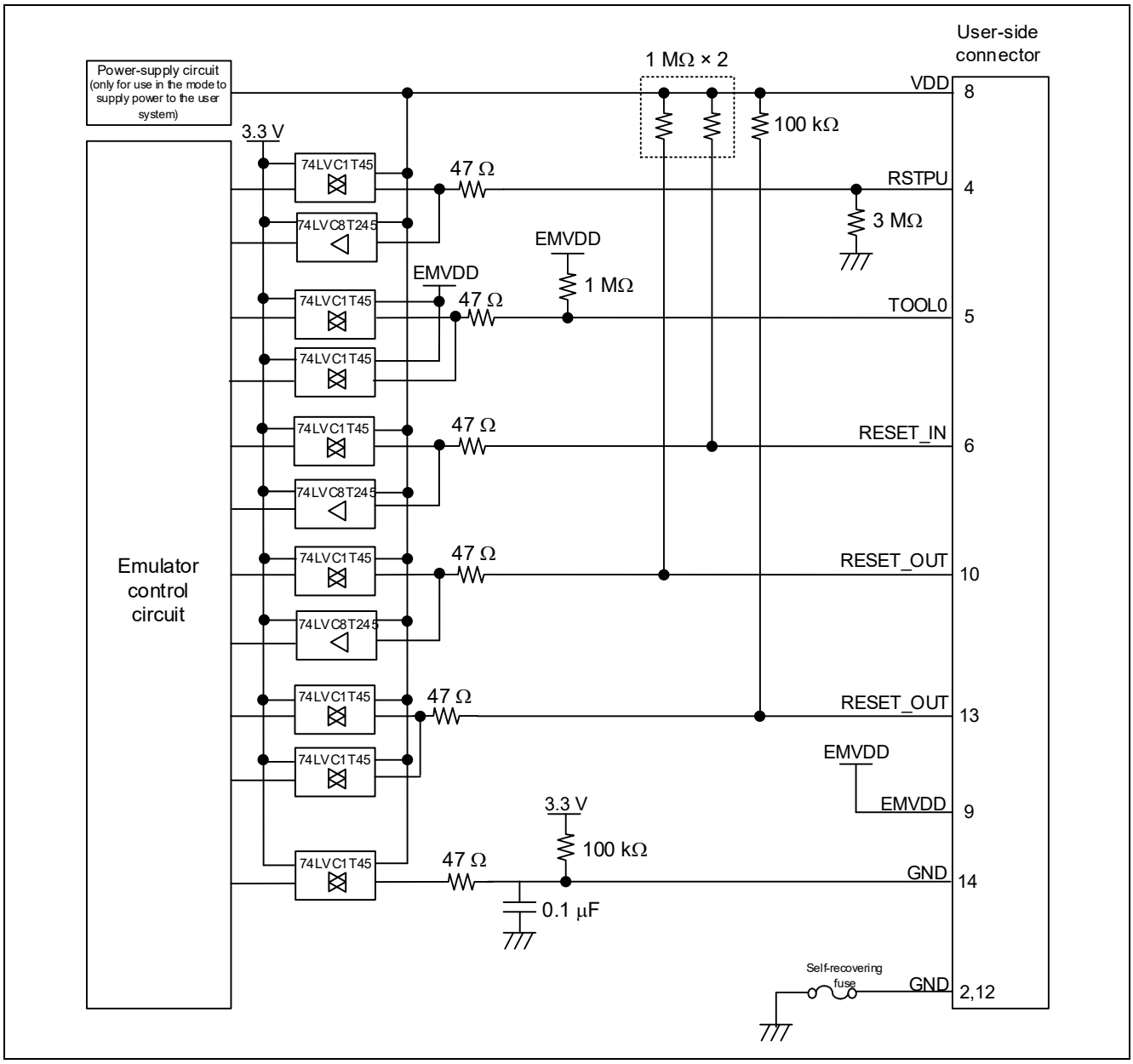


Figure 2-19 Internal Circuits of the E2 (Rev. D)

Set the switch on the 20-pin (1.27-mm pin spacing) to 14-pin (2.54-mm pin spacing) connector conversion adapter to position “3”.

2.6.4 Internal Circuits of the E2 Lite (when the RL78 Family is Connected)

Figure 2-20 shows the internal circuits of the E2 Lite with the RL78 family connected.

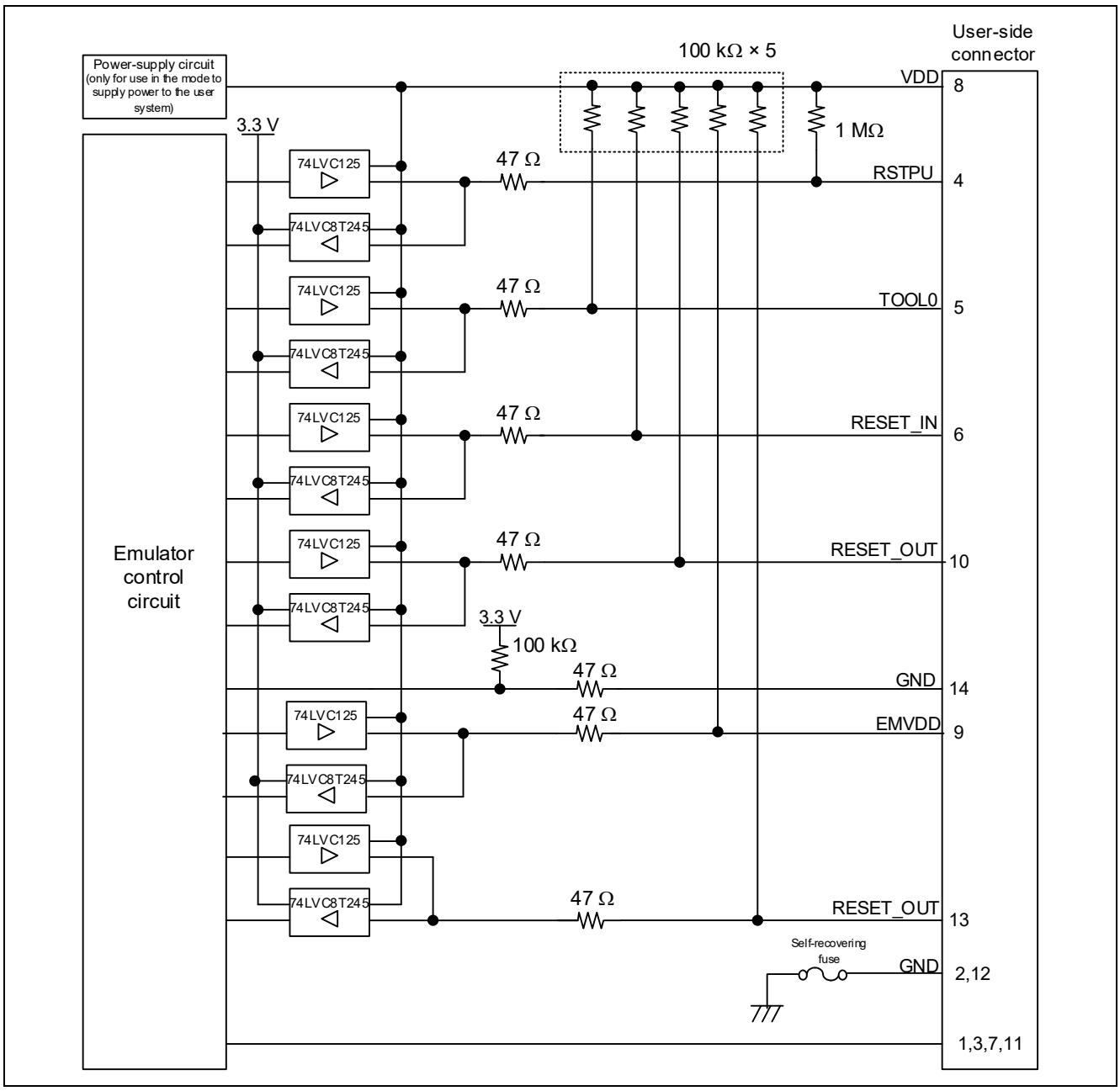


Figure 2-20 Internal Circuits of the E2 Lite (when the RL78 Family is Connected)

2.7 Notes on Designing the User System

2.7.1 Isolator for the E1

For a debugging environment where there is a difference in potential between the GND of the user system and that of the host PC, use the isolator for the E1 emulator (R0E000010ACB20) which is separately available from Renesas.

That is, use the isolator for the E1 (R0E000010ACB20) with the E2/E2 Lite in such situations.

2.7.2 Small Connector Conversion Adapter for the E1

A small connector conversion adapter for the E1 (R0E000010CKZ11) is separately available from Renesas for user system boards which are too small to mount the 14-pin connector that is the standard connector for the E1, E2, and E2 Lite. By using the adapter, you can reduce the area taken up by the connector mounted on your system.

However, when you use the small connector conversion adapter for the E1, be aware that the pin assignments of the connector differ from those of the standard interface connector for the E1, E2, and E2 Lite. The pin assignments on the 14-pin connector when the small connector conversion adapter for the E1 is used are shown in Table 2-4.

The small connector conversion adapter for the E1 can also be used with the E2 and E2 Lite.

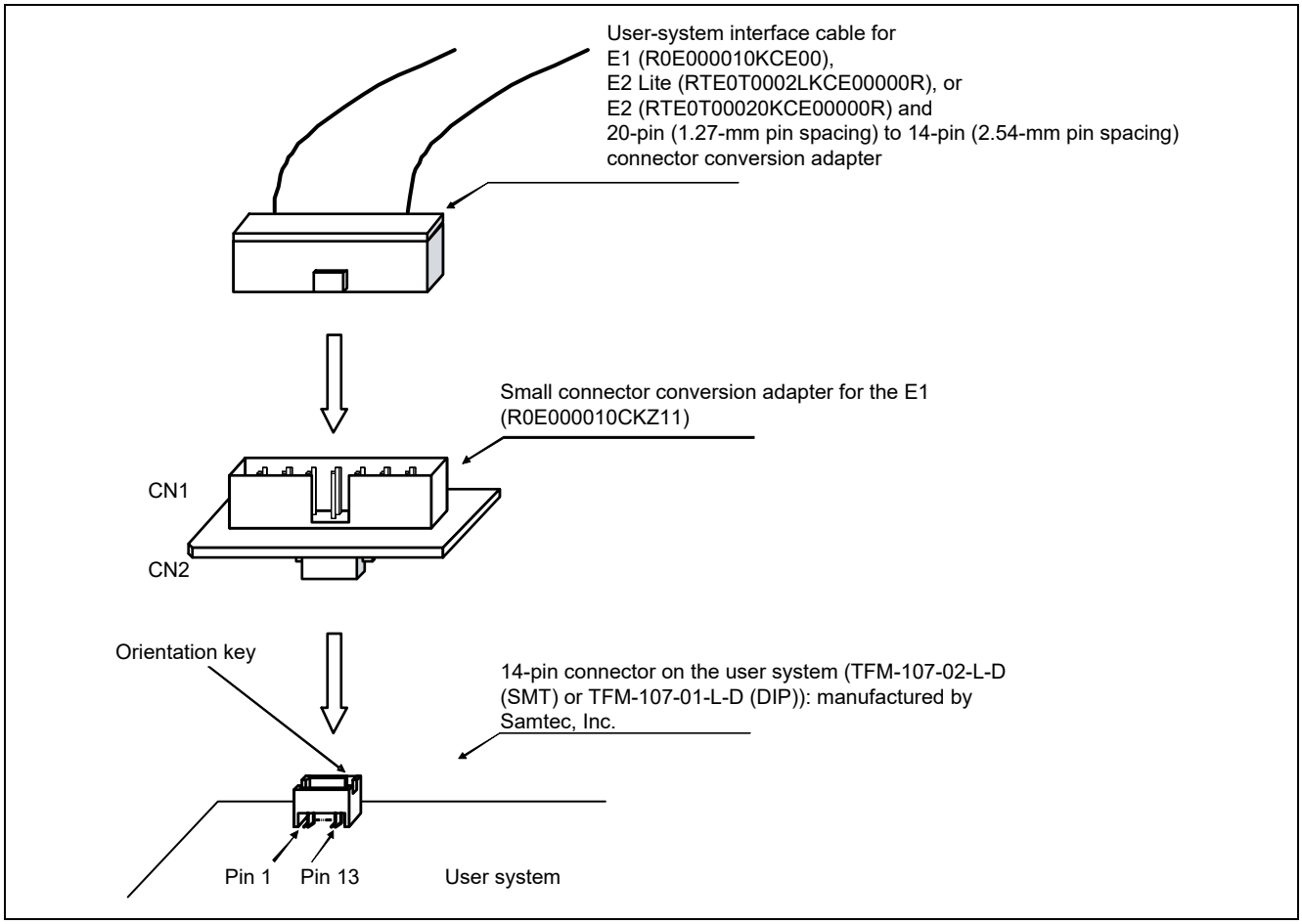


Figure 2-21 Usage of the Small Connector Conversion Adapter for the E1

Table 2-4 Connector Pin Assignments when the Small Connector Conversion Adapter for the E1 is Used

Pin No.	RL78 family MCUs in general		Only for RL78/G11, RL78/I1C, and 20- and 24-pin versions of RL78/G12		Note
	Signal *1	Direction *2	Signal *1	Direction *2	
1	GND *3	—	GND *3	—	
2	R.F.U *5	—	R.F.U *5	—	
3	VDD	—	VDD	—	
4	R.F.U *5	—	R.F.U *5	—	
5	RESET_OUT *4	Input	RESET_OUT *4	Input	This pin is used to output a reset signal to the target device.
6	GND *3	—	GND *3	—	
7	RESET_OUT *4	Input	RESET_OUT *4	Input	This pin is used to output a reset signal to the target device.
8	R.F.U *5	—	R.F.U *5	—	
9	R.F.U *5	—	R.F.U *5	—	
10	TOOL0	I/O	TOOL0	I/O	This pin is used to transmit command/data to the target device.
11	R.F.U *5	—	RSTPU	Input	This pin is used to pull up the reset line (only when the RL78/G11, the RL78/I1C, and 20- and 24-pin versions of the RL78/G12 are in use).
12	RESET_IN	Output	RESET_IN	Output	This pin is used to input a reset signal from the user system.
13	EMVDD *6	—	EMVDD *6	—	
14	GND *3	—	GND *3	—	

*1 For details on the programming software, refer to the following.
https://www.renesas.com/RFP

*2 “Input” refers to input from the emulator to the user system and “output” refers to output from the user system to the emulator.

*3 Securely connect pins 1, 6, and 14 of the connector to GND of the user system. These pins are used for electrical grounding as well as for monitoring of connection with the user system by the E1/E20/E2 Lite.

*4 Securely connect both pin 5 and pin 7.

*5 This pin is reserved. Perform the open processing.

*6 Connect the driving power supply for the TOOL0 pin to EMVDD.
Please connect VDD when the MCU does not have power supplies other than VDD such as EVDD.
The E2 Lite only supports a single power supply. If you are using an MCU that requires two or more power supplies with the E2 Lite, use a power supply other than VDD, such as EVDD, which has the same voltage as VDD.

3. Notes on Usage

3.1 Turning the Power On/Off

Turn the power of the E1/E20/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.
Check that the user system is turned off. When using the E20, check its power switch is off.
(2) Connect the user system.
Connect the emulator and the user system with a user-system interface cable.
When the external trigger input/output function is to be used with the E2, connect the pins to be used.
(3) Connect the host machine and turn on the emulator.
Connect the emulator and the host machine with a USB interface cable. The E1/E2/E2 Lite is turned on by connecting the USB interface cable. When using the E20, turn on its power switch.
(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) Launch the emulator debugger or connect the programming software to the emulator.
Connections may vary depending on software.

<When finished using the emulator>

(1) Close the emulator debugger or disconnect the emulator from the 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 the programming software.
Close the emulator debugger or the programming software.
(4) Turn off the emulator and disconnect the emulator.
When using the E20, turn off its power switch. Disconnect the USB interface cable from the E1/E2/E2 Lite. The E1/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.
When the external trigger input/output function is to be used with the E2, disconnect the pins that were in use.

CAUTION

Notes on the User System Power Supply:

While the power of the user system is on, do not turn off the host machine, unplug the USB interface cable, or turn off the power switch of the E20.

The user system may be damaged due to leakages current.

3.1.2 When Power is Supplied to the User System from the Emulator (E1/E2/E2 Lite Only)

<When using the emulator>

(1) Connect the user system.
Connect the emulator and the user system with a user-system interface cable.
When the external trigger input/output function is to be used with the E2, connect the pins to be used.
(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 or programming software.
Launch the emulator debugger or programming software and select the setting of power supply to the user system.
(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 debugger or programming software from the emulator.
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.
When the external trigger input/output function is to be used with the E2, disconnect the pins that were in use.

3.2 Power Supply Function of the E1/E2/E2 Lite

When using the power supply function of the E1/E2/E2 Lite, check the voltage supplied to the user system. Particularly when the 5.0-V supply option is selected, the voltage depends on the USB VBUS power-supply voltage.

E1: The voltage may drop by 0.5 V or more below 5.0 V.

E2: The voltage may drop by 0.3 V or more below 5.0 V.

Note that the E2 Lite supports power supply of 3.3 V only.

When debugging a system with two power supplies (VDD, EVDD, etc.) to the MCU, power cannot be supplied from the E1/E2/E2 Lite.

For the E2, 1.8 V to 5.0 V can be specified as the supply voltage and 0 V can also be specified during execution of the user program. However, when an extended function is specified, do not set the voltage to 0 V during execution of the user program.

When the supply voltage is set to 0 V by the E2 during execution of the user program, continued debugging may not be possible if the voltage on the TOOL0 pin is unstable during the voltage drop. Also refer to the notes in section 2.5.2, TOOL0 Pin.

3.3 MCU Resources to be Occupied

Figure 3-1 shows the areas which are occupied by the E1/E20/E2/E2 Lite for debugging.

These areas (shaded sections) are used for debugging. These areas should not be changed to save the user program or data. If a change is made, control by the E1/E20/E2/E2 Lite is no longer possible.

However, when "No" is selected in the [Permit flash programming] property of the debugger, the internal ROM spaces shown in Figure 3-1 and Figure 3-2 are not used (only the internal RAM spaces are used).

When selecting not to permit flash memory rewriting with the debugger's property, also refer to section 3.3.3, Setting an On-Chip Debugging Option Byte, and section 3.5.2, Operation for Voltages and Flash Operation Modes Not Permitting Flash Memory Rewriting.

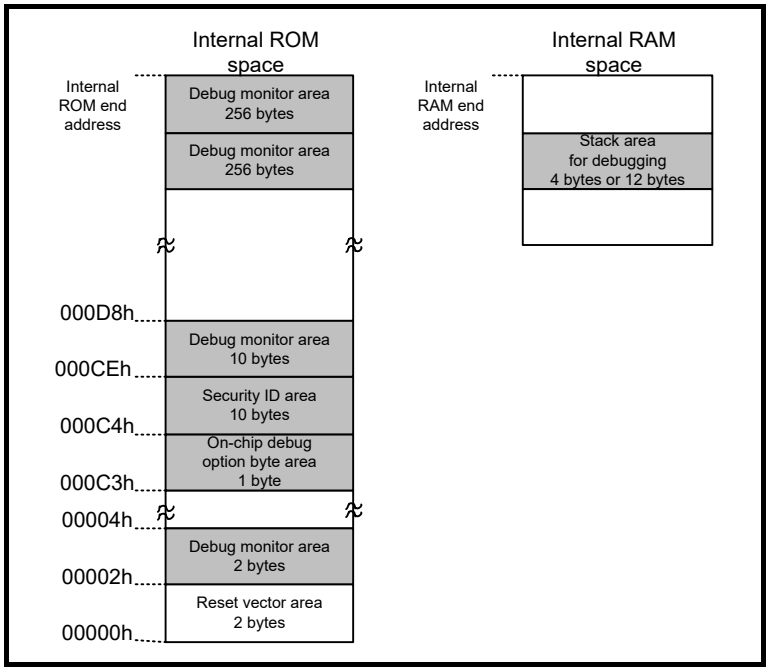


Figure 3-1 MCU Resources to be Occupied (E1/E20)

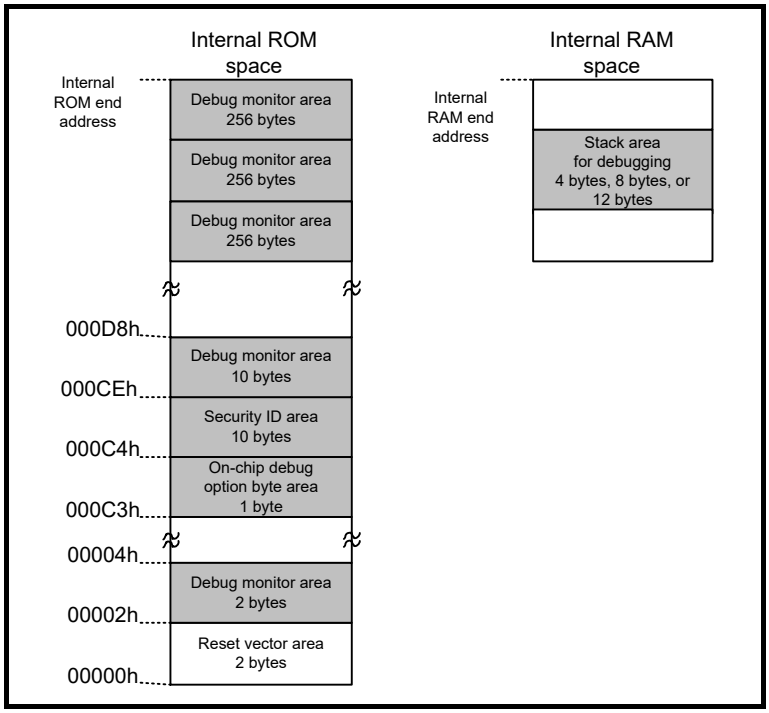


Figure 3-2 MCU Resources to be Occupied (E2/E2 Lite)

*1	The reset vector area is used by the program for the E1/E20/E2/E2 Lite when performing debugging with the E1/E20/E2/E2 Lite. If the contents of the reset vector area are changed, control by the E1/E20/E2/E2 Lite is no longer possible.

3.3.1 Securing an Area for the Debugging Monitor Program

You may need to secure the area to which the debugging monitor program is to be allocated. The monitor program initializes the debugger communications interface and handles processing to make the CPU run or break execution.

The user program or data must not be placed within 23 bytes of the on-chip debugging option byte, and leave an area of at least 768 bytes*1 before the address where the internal ROM area ends.

In addition, the reset vector must be changed if it points to an address to which the monitor program is allocated.

[Securing the area]

Specifically securing the area for the monitor program is not necessarily required if the user program does not use the area.
However, to avoid problems that may occur while the debugger is starting up, we recommend securing this area in advance by using a build tool or some other means.

Notes:	1	The required area differs according to the state of usage of the pseudo-RRM/pseudo-DMM function, the start/stop function (only supported by the E2/E2 Lite), and the monitoring point of the extended function of the E2 (only supported by the E2). Example (device with 256 Kbytes of on-chip ROM) Pseudo-RRM/pseudo-DMM function, start/stop function, and monitoring point are not in use: Monitor program is located at addresses 0x3FF00 to 0x3FFFF (256 bytes). Pseudo-RRM/pseudo-DMM function, start/stop function, or monitoring point is in use: Monitor program is located at addresses 0x3FE00 to 0x3FFFF (512 bytes). Pseudo-RRM/pseudo-DMM function and start/stop function are in use*2: Monitor program is located at addresses 0x3FD00 to 0x3FFFF (768 bytes).
	2	A monitoring point cannot be used with the pseudo-RRM/pseudo-DMM function or the start/stop function.

3.3.2 Securing a Stack Area for Debugging

The debugger requires 4 bytes as a stack area for debugging except if start/stop functions (only supported by the E2/E2 Lite) are in use, in which case the size of the stack area for debugging is up to 8 bytes. Since this area is allocated immediately below the main stack area, the address of this area varies with increases and decreases in the stack size. That is, if a program is using none of the stack (the stack is empty), the 4- or 8-byte stack area for the debugger remains.

Make sure the stack area for debugging does not go beyond the range of the internal RAM space.*1

When a monitoring point (extended function of the E2) is in use, the size of the stack area for debugging is 6 bytes.

Figure 3-3 to Figure 3-5 show examples in which the stack area has increased with the start address of the internal RAM space being 0xFCF00.

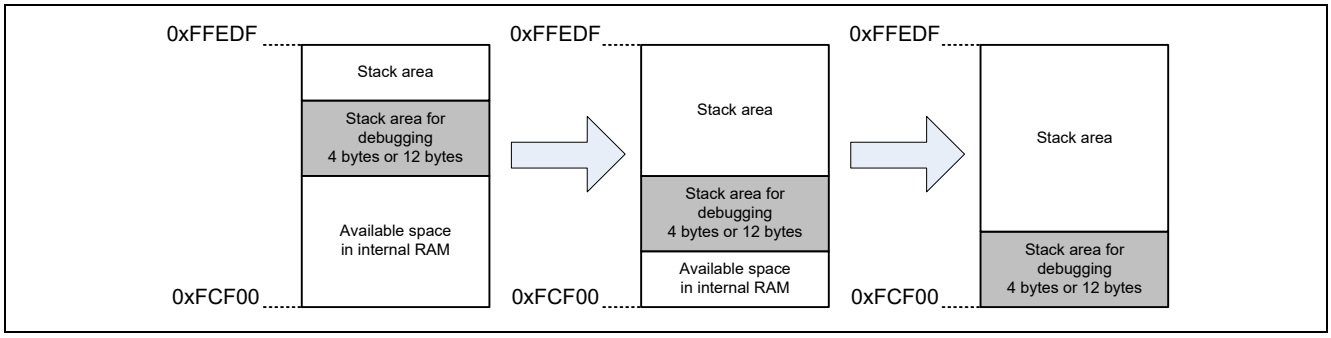


Figure 3-3 Variation of Address of Stack Area for Debugging (E1/E20)

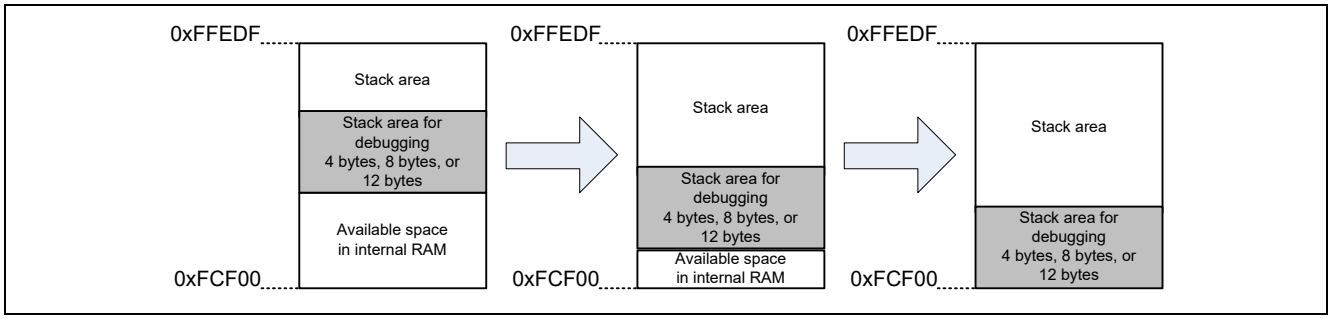


Figure 3-4 Variation of Address of Stack Area for Debugging (E2 Lite)

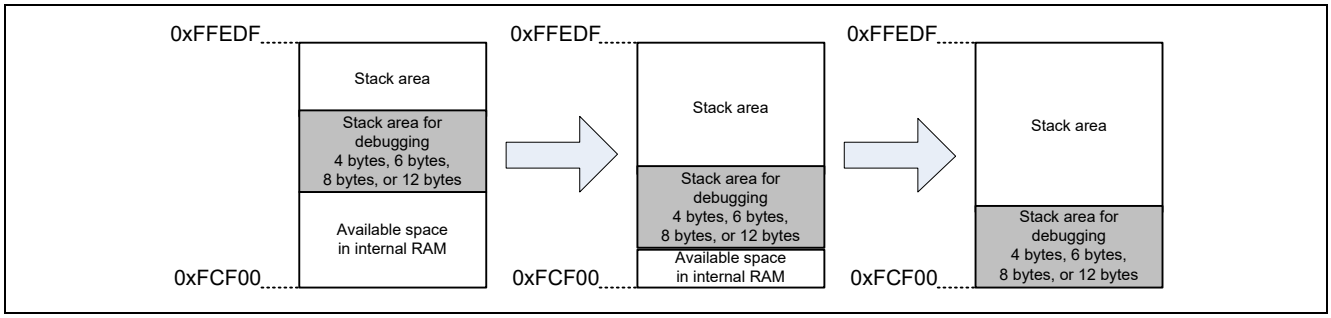


Figure 3-5 Variation of Address of Stack Area for Debugging (E2)

*1 During self-programming, the size of the stack area for debugging is up to 12 bytes.
Refer to the self-programming manual for how to secure the stack area for self programming.
The start/stop function is disabled for self-programming.

3.3.3 Setting an On-Chip Debugging Option Byte

This is the area for the security setting to prevent the flash memory from being read by an unauthorized person.

For settable values, refer to the user's manual for each MCU.

When an on-chip debugging emulator is to be connected, be sure to set a security ID.

[Setting an on-chip debugging option byte]*1

Set the on-chip debugging option byte in either of the following ways.

(a) Embed the on-chip debugging option byte at address 0xC3 in the user program.
(b) Set the on-chip debugging option byte by the build tool.

For details on the setting method, refer to the user's manual for the build tool.

*1	If the value of the on-chip debugging option byte set in the device disables on-chip debugging (OCDENSET = 0), the debugger cannot be started when "No" is selected in the [Permit flash programming] property of the debugger. Also, the user program may run out of control while starting the debugger.

3.3.4 Setting a Security ID

This setting is required to prevent the memory from being read by an unauthorized person. Embed a security ID at addresses 0xC4 to 0xCD in the internal flash memory. The debugger starts only when the security ID that is set during debugger startup and the security ID set at addresses 0xC4 to 0xCD match. If the ID codes do not match, the debugger manipulates the target device in accordance with the value set to the on-chip debug option byte area (refer to the hardware manual for each MCU). *1

If the user has forgotten the security ID to enable debugging, erase the flash memory and set the security ID again.

Set the security ID in either of the following ways. When both methods (a) and (b) are carried out at the same time, method (b) has priority.

(a) Embed the security ID at addresses 0xC4 to 0xCD in the user program.
For example if the security ID shown in Table 3-1 is embedded as follows, the security ID set by the debugger is ″0123456789ABCDEF1234″ (not case-sensitive).*2 *3

(b) Set the security ID by common options of the build tool.
For details on the setting method, refer to the user's manual for the build tool.

Table 3-1 Example of Security ID Setting

Address	Value
0x000C4	0x01
0x000C5	0x23
0x000C6	0x45
0x000C7	0x67
0x000C8	0x89
0x000C9	0xAB
0x000CA	0xCD
0x000CB	0xEF
0x000CC	0x12
0x000CD	0x34

Notes:
1. It's possible that address and length of security ID is different on every device, so refer to the user's manual for the device.
2. When connecting a debugger to a device for which the security ID has been set, the security ID needs to be entered in the debugger. For details on the authentication method, refer to the user's manual for the debugger in use.
3. The setting "0xFFFFFFFFFFFFFFFFFFFF", in which the value at all addresses is "0xFF", is prohibited.

3.3.5 Setting a Serial Programming Security ID

This setting is to prevent to write malicious program or tamper the program by an unauthorized person. Serial programming security ID is deployed at 0xD6 to 0xE5 of the internal flash memory.

On starting debugger, id certification is done, if it doesn't match, you can't debug. And If you write this ID on using memory panel, you can't debug the chip anymore.

The setting "0xFFFFFFFFFFFFFFFFFFFF", in which the value at all addresses is "0xFF", can be set.

The way to set serial programming security ID is below.

- Embed serial programming security ID by user program.
  You can embed serial programming security ID to address 0xD6 to 0xE5 on user program.
  For example if the serial programming security ID shown in is embedded as follows, the serial programming security ID set by the debugger is ″0123456789ABCDEF1234556789ABCEF0″ (not case-sensitive). *1

Table 3-2 Example of Serial Programming Security ID Setting

Address	Value
0x000D6	0x01
0x000D7	0x23
0x000D8	0x45
0x000D9	0x67
0x000DA	0x89
0x000DB	0xAB
0x000DC	0xCD
0x000DD	0xEF
0x000DE	0x12
0x000DF	0x34
0x000E0	0x56
0x000E1	0x78
0x000E2	0x9A
0x000E3	0xBC
0x000E4	0xDE
0x000E5	0xF0

Notes: 1 When connecting a debugger to a device for which the serial programming security ID has been set, the serial programming security ID needs to be entered in the debugger. For details on the authentication method, refer to the user's manual for the debugger in use.

3.4 Reset

3.4.1 Operation after a Reset

After an external pin reset or internal reset, the monitor program performs debug initialization processing.

Consequently, the time from reset occurrence until user program execution differs from that in the actual device operation. If “No” is selected in Permit flash programming in property of the debug tool, the time until the user program is executed compared with the time when “Yes” is selected is delayed (max. 100 ms).

3.4.2 SP Value after a Reset

While debugging with the E1/E20/E2/E2 Lite, the SP value after a reset becomes as follows:

- When the internal RAM space of the device is 768 bytes or more: FC00h
- When the internal RAM space of the device is less than 768 bytes:
  Start address of internal RAM + 0x20
  [Example] When the start address of internal RAM is 0xFEF00: 0xEF20

3.4.3 Internal Reset caused by illegal instruction

Internal reset of device doesn’t occur on using the emulators.

3.5 Flash Memory

3.5.1 Flash Memory Programming by Self-Programming

(1) Areas which cannot be rewritten by self-programming
If a space where the debug monitor program is allocated is rewritten by flash self-programming, the E1/E20/E2/E2 Lite can no longer operate normally. This caution also applies to boot swapping for such an area.

(2) Break occurring during self-programming
Though breaks other than a forced break can be generated while self-programming is in progress, stepped execution cannot be performed from where the break occurred. If a forced break occurs during self-programming, control by the E1/E20/E2/E2 Lite may not be possible.

(3) Use of the pseudo-RRM and pseudo-DMM functions during self-programming
The pseudo-RRM and pseudo-DMM functions are not usable while the user program is executing functions of the flash self-programming library or the data flash access library.

(4) Display in the watch panel during self-programming
Since access to memory by the RAM monitoring function is disabled while the user program is executing functions of the flash self-programming library or the data flash library, the values of variables registered in the watch panel will be displayed as “?”.

3.5.2 Operation for Voltages and Flash Operation Modes Not Permitting Flash Memory Rewriting

When a voltage with which rewriting or erasure of flash memory cannot be performed has been selected or flash programming has been disabled with the debugger's property, the following debugger operations that accompany flash memory rewriting will cause an error in the debugger and the operations will be invalidated.

- Writing to internal flash memory
- Setting or canceling of software breakpoint
- Starting execution at the set software breakpoint position
- Stepped execution at the set software breakpoint position
- Step-over execution, Return Out execution
- Execution to the caret position
- Setting, changing, or canceling of hardware breaks
- Masking/unmasking of internal reset
- Switching of peripheral breaks
- Setting or canceling a monitoring point (extended function of the E2)

The operating frequency range and operating voltage range are set with the flash operation mode. Correct operation may not be carried out when the operating frequency range or operating voltage range is exceeded.

3.6 GDIDIS

The global digital input disable register (GDIDIS) is used to prevent through-current flowing from the input buffers when the EVDD power supply is turned off (EVDD = 0 V). Since this also disables communications through the TOOL0 pin when the E1/E20/E2/E2 Lite is in use, do not set GDIDIS to 1 (prohibiting input to the input buffers).

3.7 RESET# Multiplexed Pin

In the RL78/G11, the RL78/G21 and the 20- and 24-pin versions of the RL78/G12, the RESET# pin has multiplexed pin functions. Therefore, if any of the following conditions are all satisfied, control by the E1/E20/E2/E2 Lite is no longer possible.

- The reset/port multiplexed pin is used as a port (PORTSELB = 0) due to the option byte setting (C1h).
- The option byte setting (C1h) has been rewritten during a break.
- "Yes" is selected in the [Mask INTERNAL RESET signal] property of the debugger.

3.8 MCUs that are Used in Debugging

3.8.1 Usage in Mass-Production

MCUs that are connected to the E1/E20/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.

Also, as the E1/E20/E2/E2 Lite 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.

3.8.2 Standalone Operation

After downloading a load module file to the device to for on-chip debugging, do not check the operation of this device without E1/E20/E2/E2 Lite.

A device after debugging contains the specific program for on-chip debugging, so it is different from actual operation.

3.9 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 Renesas Flash Programmer programming software or a flash programmer, without the emulator connected.

3.10 Debug Functions

3.10.1 Stepped Execution

(1) Cautions on using step-in (stepped execution)

The value of some SFRs (special function registers) might remain unchanged while stepping into code. If the values of the SFRs do not change while stepping into code, operate the microcontroller by continuously executing the instructions instead of executing them in steps.

Stepping into code: Instructions in the user-created program are executed one by one.

Continuous execution: The user-created program is executed from the current PC value.

(2) Step-in (stepped execution) of division operation (target: device with multiplier and divider/multiply-accumulator).

When the instruction which sets 1 to bit 0 (DIVST) of multiplication/division control register (MDUC) is stepped, the division operation is not finished. The stepped execution of the division operation of C source file is not relevant.

(3) Illegal memory access detection

When the function to detect illegal memory accesses is enabled (IAWCTL.7 = 1) and the following cases are performed, an internal reset will not occur.

— The stepped execution is performed for an instruction that will generate an illegal memory access.

— The software breakpoint is set at an instruction that will generate an illegal memory access, and then you execute a program from that point.

(4) Function of RAM guard

When the function of RAM guard is enabled (except IAWCTL.5, 4 = 0, 0), the guarded RAM will be overwritten when the stepped execution to overwrite the guarded RAM is performed.

The guarded RAM will be overwritten when you overwrite the guarded RAM by using memory panel during a break.

Note: When a program is continuously executed from an instruction where a software breakpoint has been set, that instruction itself is handled by stepped execution.

3.10.2 [Go to Here]

If [Go to Here] is selected, the software breakpoints and event breakpoints that have been set so far will be temporarily invalidated.

3.10.3 Debugging in Standby Mode

The break is an interrupt function of the CPU. Standby mode is released by the break generated by using the following debugging functions.

<1> Forced break

<2> Stepped execution of the standby instruction (Stops the user program after execution instruction)

<3> Short break generated by the pseudo-RRM function (Break When Readout)

<4> Short break generated by the pseudo-DMM function (Break When Write)

<5> Short break generated by setting a breakpoint while executing the user program

3.10.4 Pseudo-Real-Time RAM Monitor Function or Pseudo-Dynamic Memory Modification Function

Note the following points when using the pseudo-real-time RAM monitor (RRM) function or the pseudo-Dynamic Memory Modification (DMM) function.

- Standby mode (HALT or STOP) may be cancelled during monitoring.
- The pseudo-RRM function or the pseudo-DMM function does not operate while the CPU operating clock is stopped.
- When the number of monitoring points is numerous, the response of the debugger becomes slow.
- Influence of the debugger response becomes small by using a watch panel instead of a memory panel.
- When operating on the sub clock, neither the pseudo-RRM function nor the pseudo-DMM function will operate.
- Even when the RAM guard function is enabled, memory contents can be rewritten by the pseudo-DMM function.

3.10.5 Start/Stop Functions

The E1 and E20 do not have a “start/stop” function but the E2 and E2 Lite do. Only e^2 studio do. Note the following points if you intend to use this facility.

- Even if the start/stop functions write new values to the CPU registers, the states of the registers are restored at the ends of the functions.
- Stepped execution of the start/stop functions is not possible. However, every time stepping over of CALL instructions and so on are internally run, the start/stop function is enabled.
- Breaks cannot be used in start/stop functions.
- When execution of a user program starts from an address where a software breakpoint has been set, the instruction at the breakpoint is executed, then the start function is run. The order of execution is (a), (b), and (c) below.

  (a) Stepped execution (due to the break) of the instruction where the breakpoint is set

  (b) Running the start function

  (c) Executing instructions of the user program following the first address, where the breakpoint is set (continuous execution except in cases of breakpoints or stepped execution)

- When the start function is enabled and the user program is executed from an address at which an event break condition is set, the condition can only be satisfied after the start function has been executed. Stepped-execute the instruction to start the user program from the next address from that at which an event break condition is set, or disable the start function if you do not require it.
- If you intend to use a stop function, specify a subroutine which returns normally. If the specified subroutine does not return normally, the emulator debugger cannot control execution. To restore control, issue a reset of processing by the debugger.
- Hot plug-in is disabled while the start/stop function is in use.

3.10.6 Emulation of Flash Memory CRC Accumulator Function

(1) High-speed CRC (code flash: all area)

The operation result of CRC is different from actual result. Because monitor program is arranged and reset vector is rewritten by on-chip debugging. Please check the operation of high-speed CRC by using IECUBE or using device without E1/E20/E2/E2 Lite(Only IECUBE supports device can be confirmed by IECUBE).

(2) General-purpose CRC (code flash: specification area)

The operation result of CRC at following areas is different from actual result. Because monitor program is arranged and reset vector is rewritten by on-chip debugging.

— Reset vector area
— Debugging monitor area
— On-chip debugging option byte area
— Software break setting area

3.10.7 Break Function

When "System" is selected in the [Monitor clock] property of the debugger, if a break occurs while operating on the sub clock, the E1/E20/E2/E2 Lite switches the system clock to the main clock. Note the following in such a case.

- If the external clock (EXCLK) is set as the main clock, when a break occurs with the external clock stopped, an error will occur because the clock cannot be switched.
- Even if an SFR related to the clock is changed during a break, this change will be set in the device just before the user program is executed. Accordingly, the status flag does not change the moment the SFR is changed.
- When an SFR is rewritten to change the system clock to the sub clock during a break, though the E1/E20/E2/E2 Lite switches to the sub clock just before the user program is executed, this sometimes results in an error after user program execution due to the oscillation stabilization wait time. In this case, set "User" for [Monitor clock] before debugging.

3.10.8 Setting and Deleting Events during User Program Execution

Events can be set and deleted during user program execution. However, if a pin reset occurs after an event has been set or deleted, the contents of the event that was set or deleted during user program execution will be cleared.

3.10.9 Trace Function

When selecting not to permit flash memory rewriting with the debugger's property, the debug monitor area will not be located within 256 bytes below the end address of internal ROM. However, trace data cannot be acquired for this area.

3.10.10 Battery Backup Function

Debugging the battery backup function of the RL78/I1B, RL78G2x and RL78/L2x are not supported.

3.10.11 RAM-ECC Function

During on-chip debugging, do not enable ECC test mode for the RAM-ECC function. Doing so will disable control by the emulator.

Target devices: the RL78/F1x Series(except F12), the RL78/F2x Series

3.10.12 Extended Functions of the E2

3.10.12.1 Solution for the Tuning of Current Consumption

For notes on the solution for the tuning of current consumption, refer to the application note.

3.10.12.2 External Trigger Input and Output

- Use the value of the high-level voltage which can be input to the external trigger input in accordance with the value that has been set for the power-supply voltage of the E2 expansion interface.
- The value of the high-level voltage which is output on the external trigger outputs depends on the value that has been set for the power-supply voltage of the E2 expansion interface.
- When a break is detected and a low- or high-level pulse is output by the external trigger output function, output of the pulse is delayed after the program has stopped by a time from about one hundred microseconds to hundreds of milliseconds.
  This delay is up to 350 ms when the operating frequency is 15 kHz.

3.10.13 Points for Caution on Using the RL78/G2x and RL78/L2x

3.10.13.1 Flash Read Protection Area for the RL78/G2x and RL78/L2x

The RL78/G2x and RL78/L2x are equipped with a flash read protection area, a feature that allows you to prohibit reading from a specified memory region. Instruction fetching from the area specified as the flash read protection area is possible but reading is not. All locations in the area are read as 0xFF.

The displays for disassembly in the debugger in use with the emulator are as follows.

- CS+ ([Disassemble] panel): “?”
- e^2 studio ([Disassembly] view): “brk1”

Only set the flash read protection area after having completed debugging in the area to be set for flash read protection.

Points for caution on debugging when flash read protection has been set:

- Debugging is not possible in the area set for flash read protection.
- Do not attempt writing to or initializing the flash read protection area from the [Memory] panel; doing so will erase all blocks in the area.
- To prevent erroneous erasure of the area, we recommend setting “Prohibit inside the range” in the flash shield window function and disabling changes to the settings for the flash shield window.

3.10.13.2 Operation with 1.8-V and Lower Voltages

When you are using the RL78/G2x and RL78/L2x with 1.8-V or a lower voltage, specify [Yes] for [Use wide voltage mode] (for CS+) and [Use Wide Voltage Mode] (for the e^2 studio) in the [Connect Settings] tabbed page of the debugger. If you specify [No], the emulator will not be activated.

The items [Use wide voltage mode] (for CS+) and [Use Wide Voltage Mode] (for the e^2 studio) are not related to the facilities of the MCU.

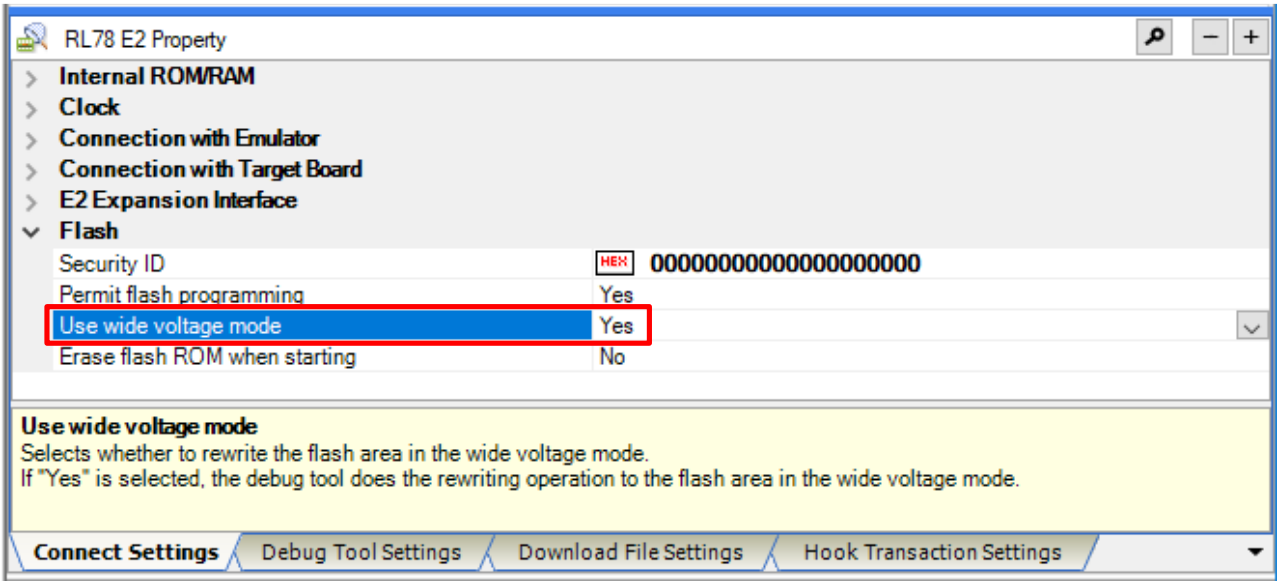


Figure 3-6 [Connect Settings] Tabbed Page for CS+

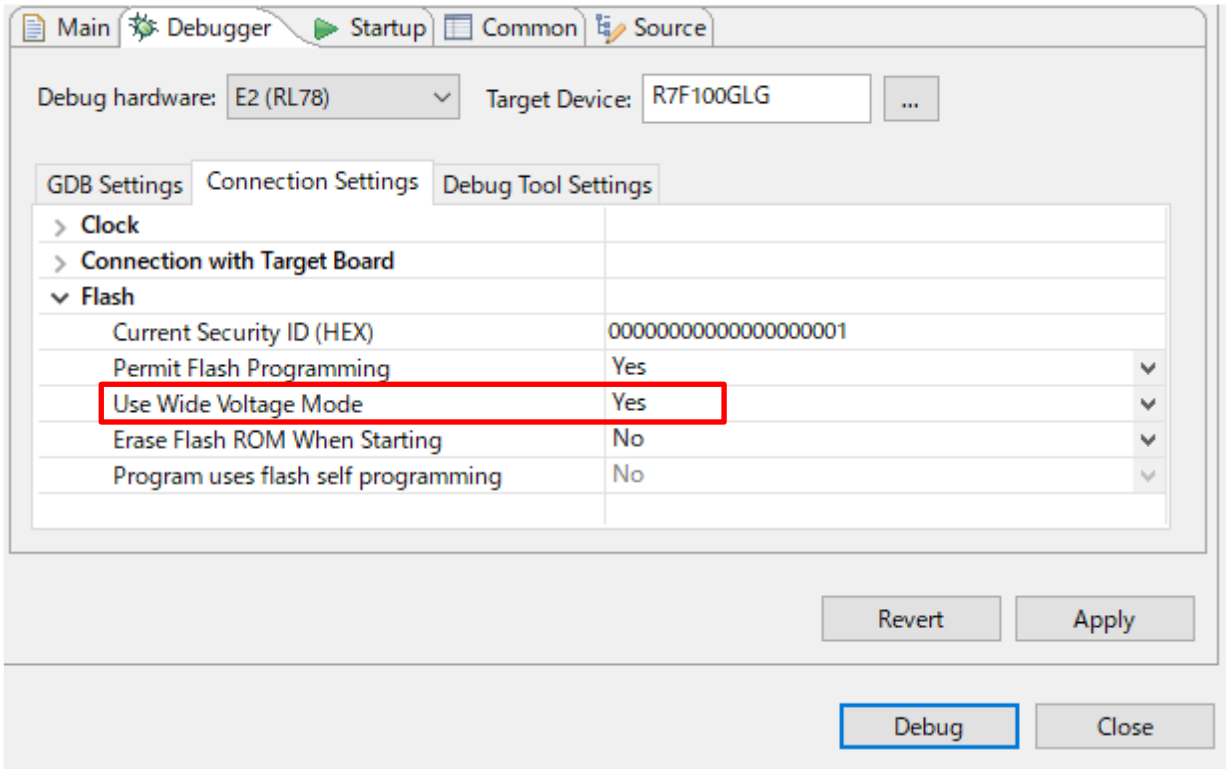


Figure 3-7 [Connect Settings] Tabbed Page for the e^2 studio

3.10.14 Points for Caution on Using the RL78/F2x

3.10.14.1 Specify RAM area size

In RL78/F2x, RAMSAR register is added to specify the size of the RAM area that can be used.

With the initial value of this register, only 4KB(0xFEF00 or later) can be used for the RAM area. If you want to use over 4KB, set the value of the RAMSAR register in the program.

The value of this register can be rewritten only once after reset.

3.10.14.2 Caution about writing of security ID

In RL78/F2x, if you rewrite OCD security ID(0xC6-0xD5) on memory panel, ID authentication of debugger fails and the program stops even if it is executed.

If you rewrite OCD security ID in memory panel, perform CPU reset in order to perform ID authentication again with the debugger.

3.10.14.3 Caution about debugging program that causes internal reset

In RL78/F2x in the case of a user program that causes internal reset immediately after execution, if a forced break is performed after the execution starts, it will break at address 0xF08C0.

Take one of the following measures.

(a) To avoid the internal reset, select "Yes" in the [Mask INTERNAL RESET signal] property of the debugger.

(b) Modify the cause the reset in the program.

3.10.14.4 Caution of Secure boot function

In RL78/F2x, if set security option byte(0xC4) to enable secure boot(set bit0 to "0"), you can't do on-chip debugging.

And on this condition, if set on-chip debug option byte(0xC3) to on-chip debug prohibited(OCDENSET=0), this chip can't debug anymore.

If Secure Boot is enabled and on-chip debugging is enabled, you can enable on-chip debugging by setting "Erase Flash ROM When Starting" to "Yes" and erasing the entire flash area.

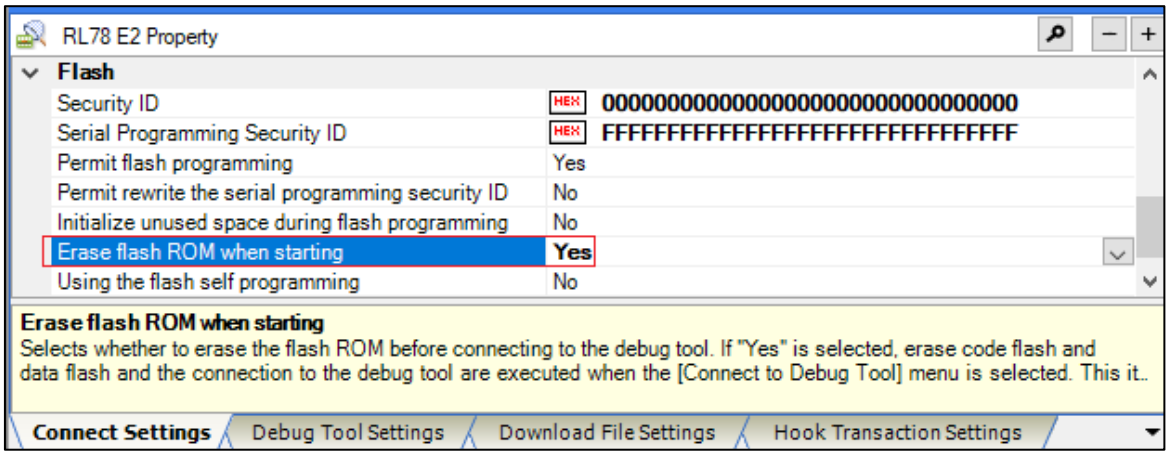


Figure 3-8 Flash ROM erase setting at startup

3.10.14.5 Caution of Secure boot function

Figure 3-9 shows the area occupied by E2/E2 Lite for debugging on RL78/F2x

These areas(shaded areas) are the space used for debugging. Do not change this area such as user programs and data placement. If you change it, you will not be able to control it with E2/E2 Lite.

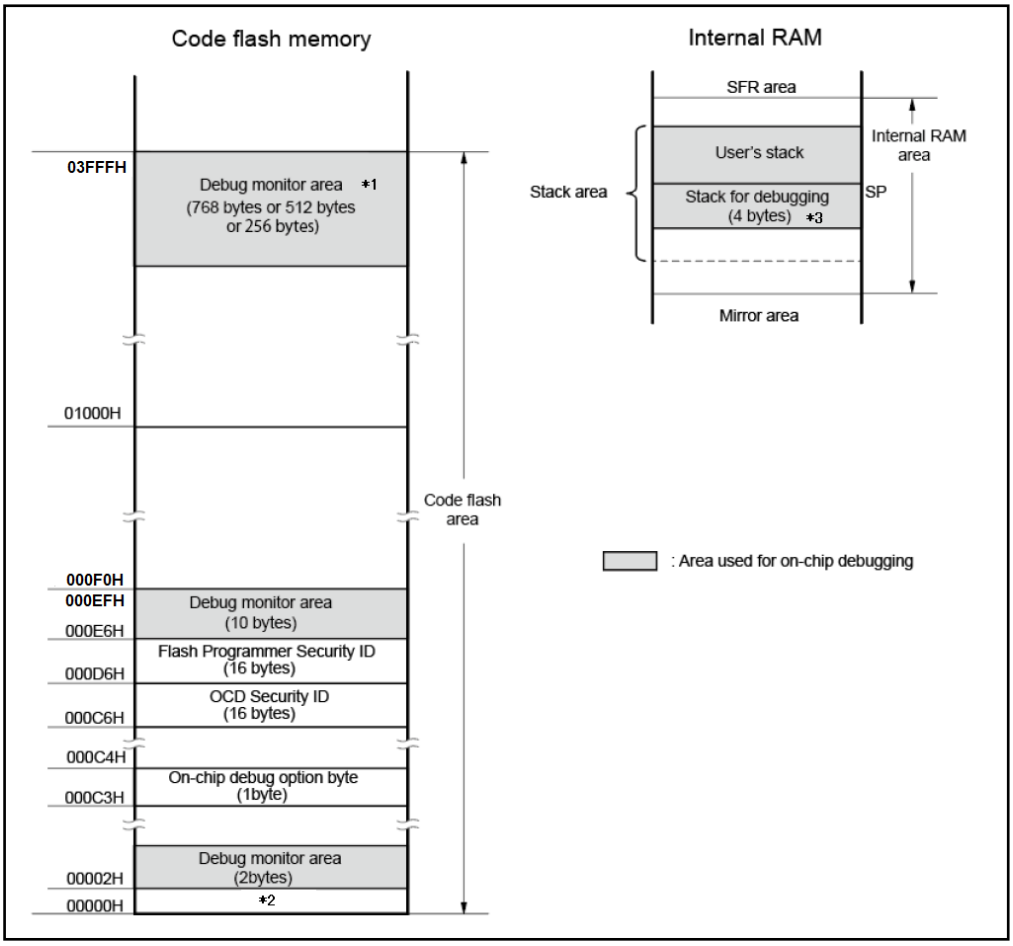


Figure 3-9 Microcomputer resources occupied by RL78/F2x

Note:	1	The required area differs according to the state of usage of the pseudo-RRM/pseudo-DMM function, the start/stop function (only supported by the E2/E2 Lite), and the monitoring point of the extended function of the E2 (only supported by the E2).
		Example (device with 256 Kbytes of on-chip ROM) Pseudo-RRM/pseudo-DMM function, start/stop function, and monitoring point are not in use: Monitor program is located at addresses 0x3FF00 to 0x3FFFF (256 bytes). Pseudo-RRM/pseudo-DMM function, start/stop function, or monitoring point is in use: Monitor program is located at addresses 0x3FE00 to 0x3FFFF (512 bytes). Pseudo-RRM/pseudo-DMM function and start/stop function are in use Monitor program is located at addresses 0x3FD00 to 0x3FFFF (768 bytes).
	2	During debugging, the reset vector is rewritten to the address assigned to the monitor program.
	3	This area is allocated just below the stack area, so the address of this area depends on whether the stack is incremented or decremented. In other words, it consumes 4 bytes extra for the stack area consumed by the user program. For self-programming, use 12 bytes as a stack area for debugging.

3.10.14.6 Reset for reprogramming Internal RAM

For RL78/F2x devices, the following area in internal RAM will be reprogrammed if reset is applied during program execution.

0xFF5F8 – 0xFF820

Please write program data on the area with the user program after reset if program data is allocated in this area.

3.10.15 Note on self-programming for flash-memory

Please DO NOT use the first 128 bytes of internal RAM as debugger will use when satisfying the following both criteria

- using RL78/G2x, RL78/L2x and RL78/F2x series devices
- selecting [Yes] in [Program uses flash self programming] under [Flash] on [Debug Configurations].

3.10.16 State of high-speed on chip oscillator during on-chip debugging

During on-chip debugging, high-speed on-chip oscillator will be oscillation state regardless of any CPU operation mode or any value is set on the bit0[HIOSTOP] of CSC register.

Please be careful peripherals which are supplied clock source apart from CPU will remain in operation similarly. The peripherals such as 32-bit interval timer (TML32) or Logic and event link controller (ELCL) for RL78/G2x and RL78/L2x series are the example.

3.10.17 Setting of download mode (RL78/FGIC[RAJ240100])

On RAJ240100 which is one of RL78/FGIC, please select download mode to "Data priority". If "Speed priority" is selected, a part of data will be deleted and the chip cannot be used.

3.10.18 Stopped address by hardware break

Since a hardware break is a post-PC break, if the set address is executed or the data access condition is satisfied, the program may stop at an address several cycles ahead.
To stop the program at the expected address, use a software break.

3.10.19 Setting for Using Bank Swap with BTBLS

For RL78/L2x series models equipped with the bank swap function, the debugger can modify the BTBLS bit of the target MCU at startup to enable the bank swap function.

When changing the setting, the debugger sets BTBLS to 07H and BAPR to 0H on the target MCU.

When configuring the debugger to use the bank swap function at startup, if the target MCU meets either of the following conditions, the debugger will start without changing the settings:

- The BTBLS bit is not at its default value.
- The BAPR bit is set to 0.

If either condition applies and you want to set the debugger to use the bank swap function at startup, please initialize the debug target MCU in advance using Renesas Flash Programmer or similar tools.

3.10.20 Precautions during Bank Programming Mode

For models in the RL78/L2x series that are equipped with the bank swap function, data updates to the rewrite bank using the bank programming function can be viewed in the debugger.

However, during the period when the rewrite bank is in programming mode, debug functions that involve stopping the user program cannot be used.

If the user program stops during this period, the debugger will lose control. Therefore, please configure or use the debugger in a way that prevents the following debug functions—which involve stopping the user program—from being activated:

- Forced break function
- Software break
- Event break
- “Go to Here” function
- Pseudo-RRM function
- Pseudo-DMM function
- Short break generated by setting a breakpoint while executing the user program
- Break by trace full
- Break by E2 extended functions

3.10.21 Open-source software used

The E2/E2 Lite uses the following open-source software.

- libusb (https://libusb.info/)
  libusb is licensed under the GNU Lesser General Public License version 2.1.
  Full license text at (https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html.en).

Revision History	E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for Manual (Notes on Connection of RL78)

Rev.	Date	Description	
		Page	Summary
4.01	Mar.16.16	All	Information about RL78/F15, RL78/G1H, and RL78/I1E groups was added.
		30	Note (3) was added to section 3.5.1.
5.00	Mar.16.16	All	Information about RL78/I1C and RL78/L1A groups was added.
		15, 16	Section 2.4.3, Connection between the 14-Pin Connector and the RL78 Family MCU (Only the RL78/I1C), was added.
6.00	Sep.16.16	All	Information about RL78/G11 group was added.
		36	Note (3) in section 3.10.1 was changed.
6.01	Mar.01.17	36	Note (4) was added to section 3.10.1.
7.00	Jul.01.17	All	Information about the E2 emulator was added.
7.01	Apr.16.19	All	Information about RL78/FGIC was added.
		48	Section 3.10.11, RAM-ECC Function, was newly added to Notes on Usage.
7.02	Jul.01.20	All	Information about RL78/G1P and RL78/G13A was added.
8.00	Mar.16.21	All	Information about RL78/G23 was added.
		34	A diagram of the internal circuits of product revision D of the E2 was added.
		51, 52	Section 3.10.13, Points for Caution on Using RL78/G23, was added.
9.00	Jan.20.22	All	Information about RL78/F24 was added.
		44	Note (1) was added to section 3.3.4.
		45	Setting of Serial programming security ID was added to section 3.2.5.
		54, 55	Section 3.10.14, Points for Caution on Using the RL78/F24, was added.
9.01	Jan.20.23	All	Information about RL78/G15, RL78/H1D, RL78/G22, Rl78/F23 was added.
		26	Figure2-9 Caution about RESET_IN when unconnected was added.
		43	Content of footnote was modified.
		51	Section 3.10.6, Device information that IECUBE can confirm was added.
		56	Section 3.10.14.6, Caution of Reset on RL78/F2x was added.
		56	Section 3.10.15, Caution on self-programming was added.
		56	Section 3.10.16, Caution of high-speed on chip oscillator was added.
		56	Section 3.10.17, Caution about setting of hardware break point on RAM area was added.
		56	Section 3.10.18, Caution about setting of download mode was added.
9.10	Oct.06.23	All	Delete describe of RL78/G15(moved to “R20UT2937 E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User’s Manual(Notes on Connection of RL78)”
		All	Information about RL78/G24 was added.
9.20	Sep.20.24	46	Section 3.4.3, Caution about Internal caused by illegal instruction was added
		50	Section 3.10.5, describe only e2studio can do “start/stop” function
		51	Section 3.10.10, add the RL78/G2x to not supported devices
9.30	Jan.20.25	56	Section 3.10.18, Stopped address by hardware break was added.
		56	Section 3.10.19, Open-source software used was added.
9.40	Oct.6.25	ALL	Information about RL78/L23, RL78/F22, RL78/F25 was added.
		57	Section 3.10.19, Setting for Using Bank Swap with BTBLS was added.
		57	Section 3.10.20, Precautions During Bank Programming Mode was added.
9.50	Oct.28.26	ALL	Information about RL78/G21 was added.

E1/E20/E2 Emulator, E2 Emulator Lite
Additional Document for Manual
(Notes on Connection of RL78)

Publication Date: Rev.9.50 Oct.28.26

Published by: Renesas Electronics Corporation

E1/E20/E2 Emulator, E2 Emulator Lite
Additional Document for Manual
(Notes on Connection of RL78)

Renesas Electronics Corporation

R20UT1994EJ0950