User's Manual

Pod for IE850 In-circuit Emulator

RTE7701460EPA00000R

User's Manual

Target Devices:
RH850/D1x Series

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

Renesas Electronics
www.renesas.com

Rev.1.20 Jul 2017

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 disputes 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, drawing, chart, program, algorithm, application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and 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.
   Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)

Important

Before using this product, be sure to read this user’s manual carefully.
Keep this user’s manual, and refer to it when you have questions about this product.

Emulator:
"Emulator" in this document collectively refers to the following products manufactured by Renesas Electronics Corp.
(1) IE850 emulator main unit
(2) Pod

"Emulator" herein encompasses neither the customer's target system nor the host machine.

Purpose of use of the emulator:
This emulator is a device to support the development of systems that use products from the RH850/D1x series of Renesas microcontrollers. It provides support for system development in both software and hardware.
Be sure to use this emulator correctly according to said purpose of use. Please avoid using this emulator other than for its intended purpose of use.

For those who use this emulator:
This emulator can only be used by those who have carefully read the user’s manual and know how to use it.
Use of this emulator requires basic knowledge of electric circuits, logical circuits, and MCUs.

When using the emulator:
(1) This product is a development-support unit for use in your program development and evaluation stages. When a program you have finished developing is to be incorporated in a mass-produced product, the judgment as to whether it can be put to practical use is entirely your own responsibility, and should be based on evaluation of the device on which it is installed and other experiments.
(2) In no event shall Renesas Electronics Corp. be liable for any consequence arising from the use of this product.
(3) Renesas Electronics Corp. strives to provide workarounds for and correct trouble with products malfunctions, with some free and some incurring charges. However, this does not necessarily mean that Renesas Electronics Corp. guarantees the provision of a workaround or correction under any circumstances.
(4) The product covered by this document has been developed on the assumption that it will be used for program development and evaluation in laboratories. Therefore, it does not fall within the scope of applicability of the Electrical Appliance and Material Safety Law and protection against electromagnetic interference when used in Japan.
(5) Renesas Electronics Corp. cannot predict all possible situations and possible cases of misuse that carry a potential for danger. Therefore, the warnings in this user's manual and the warning labels attached to the emulator do not necessarily cover all such possible situations and cases. The customer is responsible for correctly and safely using this emulator.
(6) The product covered by this document has not been through the process of checking conformance with UL or other safety standards and IEC or other industry standards in countries other than Japan. This fact must be taken into account when the product is taken from Japan to another country.
(7) Renesas Electronics Corp. will not assume responsibility for direct or indirect damage caused by an accidental failure or malfunction of this emulator.

When disposing of the emulator:
Penalties may be applicable for incorrect disposal of this waste, in accordance with your national legislation.

Usage restrictions:
The emulator has been developed as a means of supporting system development by users. Therefore, do not use it as an embedded device in other equipment. Also, do not use it to develop systems or equipment for use in the following fields.

(1) Transportation and vehicular

(2) Medical (equipment that has an involvement in human life)

(3) Aerospace

(4) Nuclear power control

(5) Undersea repeaters

If you are considering the use of the emulator for one of the above purposes, please be sure to consult your local distributor.

About product changes:
We are constantly making efforts to improve the design and performance of this emulator. Therefore, the specification or design of this emulator, or this user's manual, may be changed without prior notice.

About rights:

(1) We assume no responsibility for any damage or infringement on patent rights or any other rights arising from the use of any information, products or circuits presented in this user’s manual.

(2) The information or data in this user’s manual does not implicitly or otherwise grant a license to patent rights or any other rights belonging to Renesas or to a third party.

(3) This user’s manual and this emulator are copyrighted, with all rights reserved by Renesas. This user’s manual may not be copied, duplicated or reproduced, in whole or part, without prior written consent from Renesas.

About diagrams:
Some diagrams in this user’s manual may differ from the objects they represent.

Precautions for Safety

This chapter, by showing the relevant diagrammatic symbols and their meanings, describes the precautions which should be taken in order to use this product safely and properly. Be sure to read and understand this chapter before using this product. Contact us if you have any questions about the precautions described here.

⚠ WARNING **WARNING** indicates a potentially dangerous situation that will cause death or heavy wound unless it is avoided.

⚠ CAUTION **CAUTION** indicates a potentially dangerous situation that will cause a slight injury or a medium-degree injury unless it is avoided.

In addition to the two above, the following are also used as appropriate.

△ means WARNING or CAUTION.

Example:  CAUTION AGAINST AN ELECTRIC SHOCK

⃠ means PROHIBITION.

Example:  DISASSEMBLY PROHIBITED

● means A FORCIBLE ACTION.

Example:  UNPLUG THE POWER CABLE FROM THE RECEPTACLE.

⚠ WARNING

Warnings for AC Power Supply:

If the separately sold AC power cable for the power adaptor does not fit the receptacle, do not alter the AC power cable and do not plug it in forcibly. Failure to comply may cause electric shock and/or fire.

Use an AC power cable which complies with the safety standard of the country.

Do not touch the plug of the AC power cable when your hands are wet. This may cause electric shock.

This product is connected signal ground with frame ground. If your developing product is transformless (not having isolation transformer of AC power), this may cause electric shock. Also, this may give an unrepairable damage to this product and your developing one.
While developing, connect AC power of the product to commercial power through isolation transformer in order to avoid these dangers.

Connect the plug of the AC power cable to the outlet when connecting this emulator and the target system in order to eliminate differences in potential between the grounds of the emulator and of the user's system.

If other equipment is connected to the same branch circuit, care should be taken not to overload the circuit.

When installing this equipment, insure that a reliable ground connection is maintained.

If you smell a strange odor, hear an unusual sound, or see smoke coming from this product, then disconnect power immediately by unplugging the AC power cable from the outlet.
Do not use this as it is because of the danger of electric shock and/or fire. In this case, contact your local distributor.

Before setting up this emulator and connecting it to other devices, turn off power or remove a power cable to prevent injury or product damage.

⚠ WARNING

Warnings to Be Taken for This Product:

Do not disassemble or modify this product. Personal injury due to electric shock may occur if this product is disassembled and modified. Disassembling and modifying the product will void your warranty.

Make sure nothing falls into the cooling fan on the top panel, especially liquids, metal objects, or anything combustible.

Note the following point on products which have a cooling fan.

When the fan does not operate due to, for example, a fault, the temperature of the emulator may be high enough to potentially cause injuries (such as burns) on contact. Accordingly, if the fan does not operate after the emulator is turned on, turn the emulator off immediately and send it to be repaired.

Warning for Installation:

Do not set this product in water or areas of high humidity. Make sure that the product does not get wet. Spilling water or some other liquid into the product may cause unrepairable damage.

Warning for Use Environment:

Care should be taken to ensure that the emulator is not used at temperatures exceeding the maximum ambient temperature.

⚠ CAUTION

Point for Caution Regarding the Power Adaptor:

Use only the dedicated power adaptor which is separately sold. Also, do not use the power adaptor for other equipment.

Cautions to Be Taken for Turning On the Power:

Take the steps below to turn the power to the emulator ON or OFF. Not following the order might cause damage to the target system or emulator.

When turning on the power: (1) Turn on the emulator; (2) turn on the target system; (3) connect the debugger (emulator software)

When turning off the power: (1) Disconnect the debugging session (emulator software); (2) turn off the target system; (3) turn off the emulator

Cautions to Be Taken for Handling This Product:

Use caution when handling the emulator. Be careful not to apply a mechanical shock.

Do not touch the connector pins of the emulator and the target system directly. Doing so may lead to the discharge of static electricity and so damage the internal circuits.

When attaching and removing the pod cable, hold a fixture (such as a connector) to avoid pulling the pod cable. Do not pull the emulator and board by the communications interface cable or the cable for connecting the target system since this might lead to the breaking of wires in the pod cable. Also, do not flex the pod cable excessively when installing the cable since this might lead to the breaking of wires in the cable.

Do not use inch-size screws for this equipment. The screws used in this equipment are all ISO (meter-size) type screws.

Caution to Be Taken for System Malfunctions:

If the emulator malfunctions because of interference like external noise, do the following to remedy the trouble.

(1) Exit the debugger (emulator software), and shut OFF the emulator and the target system.

(2) After a lapse of 10 seconds, turn ON the power of the emulator and the target system again, then launch the debugger (emulator software).

⚠ CAUTION

Note on Heat Generation While the Product is in Use:

Using this product for a long time might cause the product to have a high temperature. If this is the case, care must be taken to avoid injuries due to the heat such as low temperature burns.

Note on the Cover of the Emulation Pod:

Be sure to use the product with the cover of the emulation pod in place.

Note on Transporting the Product:

When sending your product for repair, use the packing box and cushioning material supplied with the product when it was delivered to you and specify caution in handling (handling as precision equipment). If packing of your product is not complete, it may be damaged during transportation. When you pack your product in a bag, make sure to use the conductive plastic bag supplied with the product. If you use a different bag, it may lead to further trouble with your product due to static electricity.

Caution to Be Taken for Disposal:

Penalties may be applicable for incorrect disposal of this waste, in accordance with your national legislation.

European Union Regulatory Notices:

The WEEE (Waste Electrical and Electronic Equipment) regulations put responsibilities on producers for the collection and recycling or disposal of electrical and electronic waste. Return of WEEE under these regulations is applicable in the European Union only. This equipment (including all accessories) is not intended for household use. After use the equipment cannot be disposed of as household waste, and the WEEE must be treated, recycled and disposed of in an environmentally sound manner.
Renesas Electronics Europe GmbH can take back end of life equipment, register for this service at “http://www.renesas.eu/weee”.

How to Use This Manual

Readers

This manual is intended for users who wish to perform debugging using the RTE7701460EPA00000R (generic name: pod). The readers of this manual are assumed to be familiar with the device functions and usage, and to have knowledge of debuggers.

Purpose

This manual is intended to give users an understanding of the basic specifications and correct usage of the pod.

Organization

This manual is divided into the following sections.

- Overview
- Names and Functions of Hardware
- Setup Procedure
- Notes
- Optional Product
- Maintenance and Warranty

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers. This manual describes the basic setup procedures and how to set switches.

To understand the overall functions and usages of the IE850

→ Read this manual in the order of Contents.

To know the manipulations, command functions, and other software-related settings of the IE850

→ See the user's manual of the debugger to be used.

Conventions

Note:	Footnote for item marked with Note in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH

Prefix indicating power of 2 (address space, memory capacity):

K (kilo): 2^{10} = 1,024

M (mega): 2^{20} = $1,024^2$

Terminology

The meanings of the terms used in this manual are described in the table below.

Term	Meaning
Target device	This is the device to be emulated.
Target system	This is the system to be debugged (system provided by the user). This includes the hardware and software provided by the user.
IE850	Name for IE850 emulator systems in general.
IE850 main	QB-V850E2
Pod	The pod is a peripheral of the IE850 main unit and serves as the interface with the target system.
Emulator	This is the product to emulate the target device. Refers to the IE850 main unit and pod in this document.

Related Documents Please refer to the related documents listed below in addition to this manual before using the product.

Name of documents		Document No.
Release note	E1/E20/E2 Emulator for the RH850 Family and the RH850 Pod for IE850 Release Note (Restrictions on the Emulator and the Pod when Used with CS+)	R20UT3695E
IE850 main unit	Main unit for IE850 In-circuit Emulator QB-V850E2 User's Manual	R20UT0824E
Exchange adaptor (EA)	RTE7701412CBF484T000R Exchange Adaptor for Connecting RH850/D1x Series in 484-pin 1.0-mm-pitch BGA	R20UT2991E
	RTE7701410CBF376T000R Exchange Adaptor for Connecting RH850/D1x Series in 376-pin 1.0-mm-pitch BGA	R20UT2992E
	RTE7701461CBF272T000R Exchange Adaptor for Connecting RH850/D1x Series in 272-pin 1.0-mm-pitch BGA	R20UT3899E
	RTE7701463CFK176T000R Exchange Adaptor for Connecting RH850/D1x Series in 176-pin 0.5-mm-pitch LFQFP	R20UT3900E
	RTE7701422CFK144T000R Exchange Adaptor for Connecting RH850/D1x Series in 144-pin 0.5-mm-pitch LFQFP (Former type number: RTE7701401CFK144T000R)	R20UT3901E
	RTE7701421CFK144T000R Exchange Adaptor for Connecting RH850/D1x Series in 144-pin 0.5-mm-pitch LFQFP	R20UT3902E

Caution: Related documents listed above are subject to change without prior notice. Thus, make sure that you have the latest documents for use in design.

Table of Contents

1. Overview

The RTE7701460EPA00000R (pod) is used together with the QB-V850E2 (IE850 main unit) in order to emulate RH850/D1x series microcontrollers from Renesas Electronics Corp.

The IE850 can be used to debug hardware and software efficiently when developing systems using the target device.

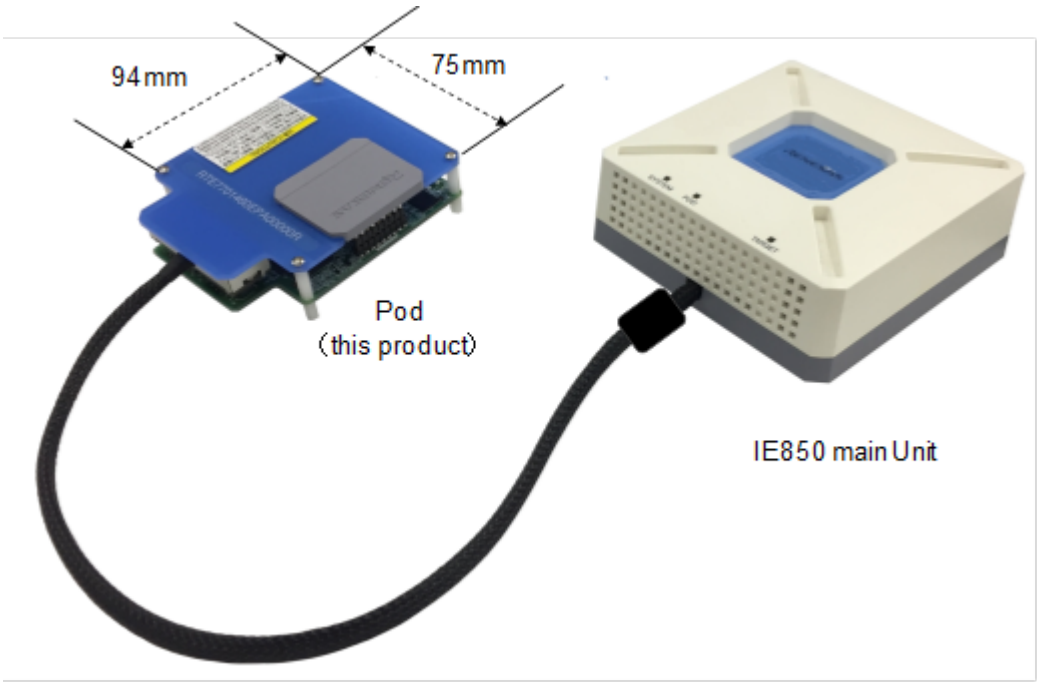


Figure 1-1 External Appearance

1.1 Hardware specification

The following table describes hardware specifications of the pod.

Table 1-1 Pod Hardware Specifications

Parameter	Specification
Emulation target device	484-pin BGA: RH850/D1M2H 376-pin BGA: RH850/D1M2 272-pin BGA: RH850/D1M1A 272-pin BGA: RH850/D1M1H 176-pin QFP: RH850/D1M1 176-pin QFP: RH850/D1L2H 144-pin QFP: RH850/D1L2 144-pin QFP: RH850/D1L1
Emulator system voltage	5.5V, 3.3V (supplied by IE850 main)
Target system interface voltage	Depend on target device
Maximum operating frequency	240 MHz
Input frequency of the oscillator	8 MHz (for the oscillator mounted on the pod as shipped)
Operating temperature range	0°C to 40°C (No condensation)
Storage temperature range	–15°C to 60 °C (No condensation)

1.2 System overview

The system configuration is described below. The pod cannot be used on its own. The IE850, an AC adaptor, and sockets are also required. These are sold separately.

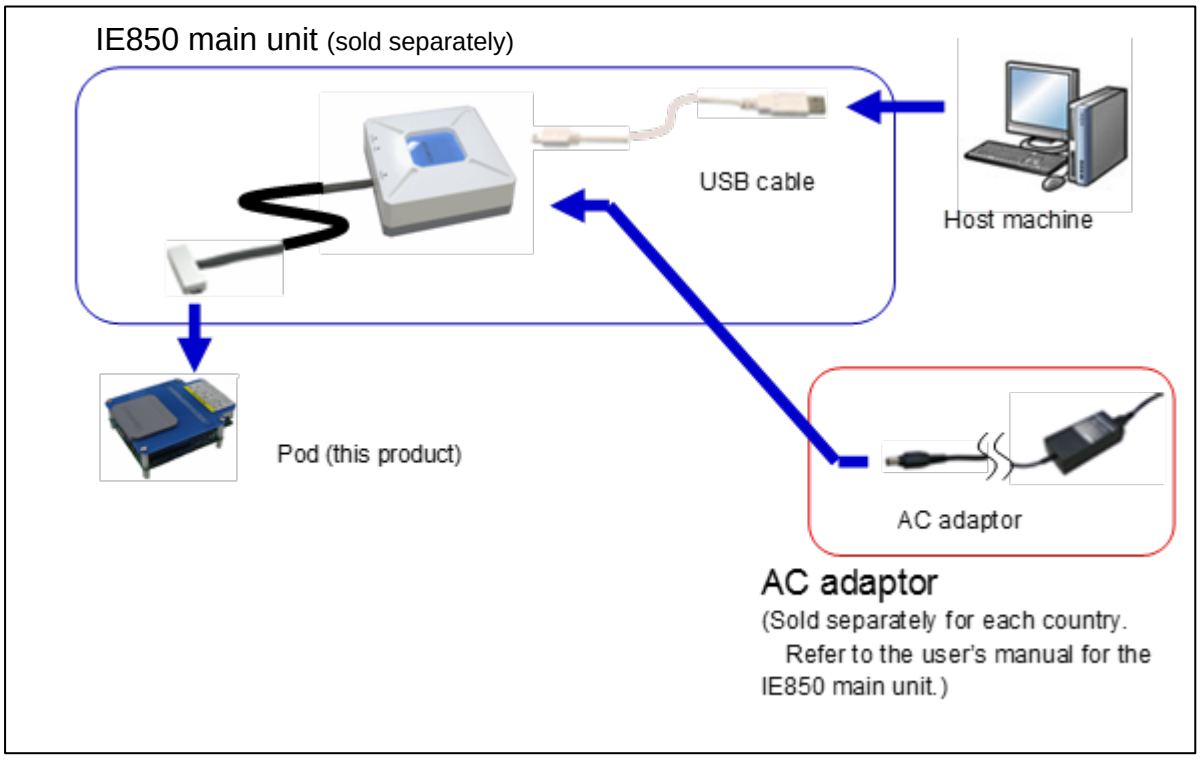


Figure 1-2 System Configuration (when the Target System is not Connected)

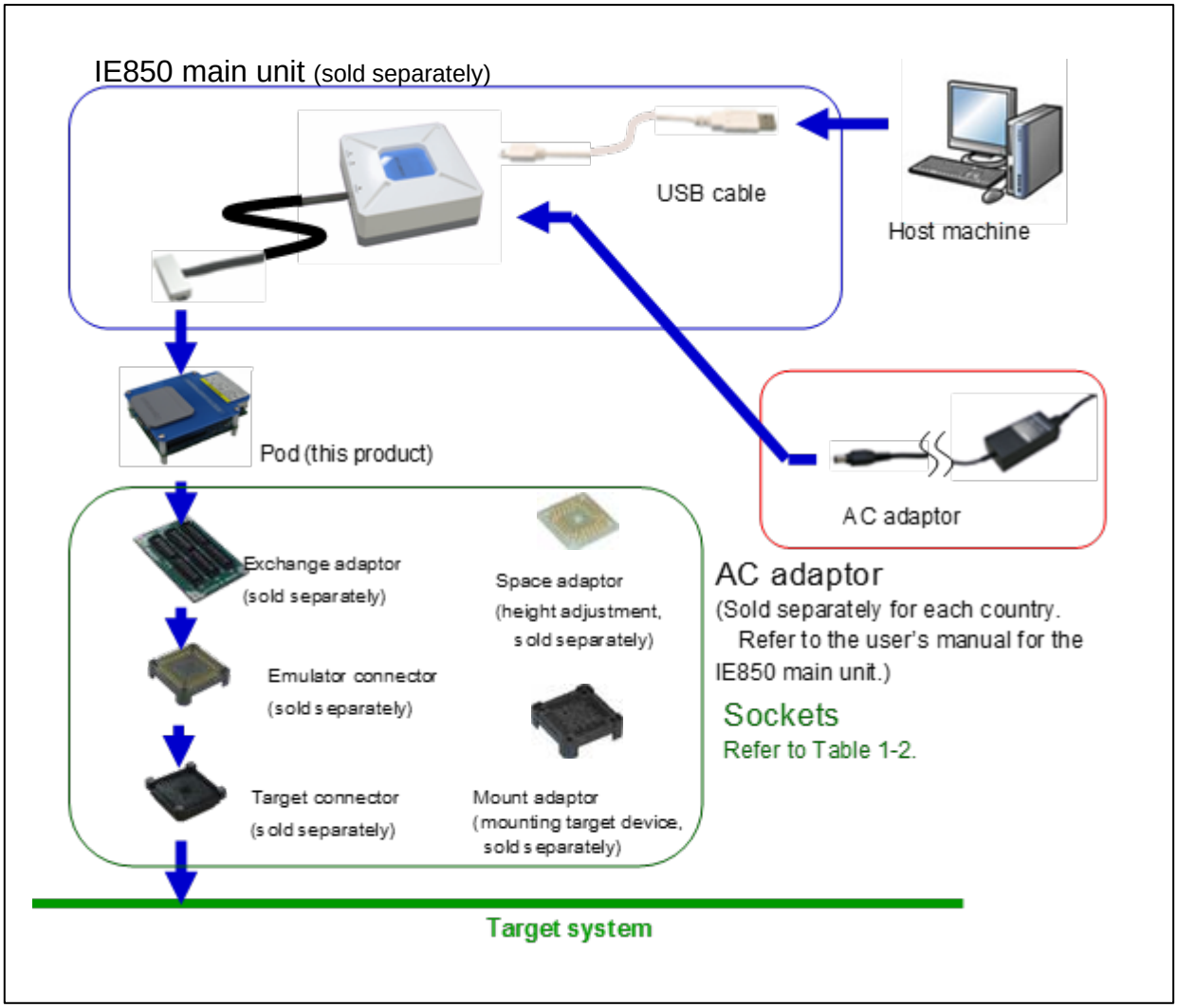


Figure 1-3 System Configuration (when the QFP Package is Selected)

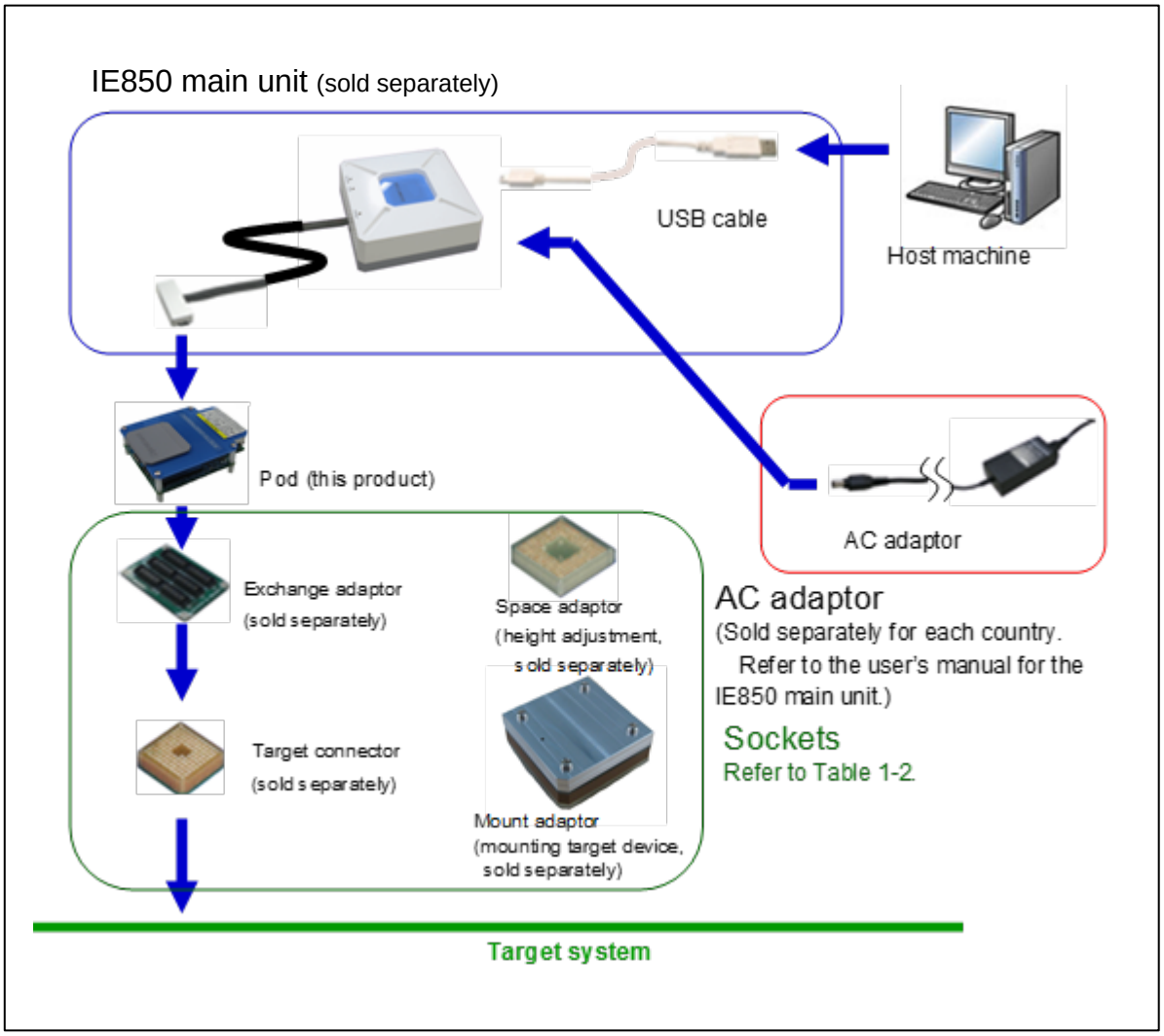


Figure 1-4 System Configuration (when the BGA Package is Selected)

The following table describes the corresponding sockets for the target device. Socket products are sold separately from this product (the pod). You can purchase the target connector (BSSOCKET), the space adaptor (CSSOCKET), and the mounting adaptor (LSPACK) for the BGA packages from Tokyo Eletech Corporation and the other components are available from Renesas Electronics Corporation.

Table 1-2 Sockets for the Target Device

Package Device	Exchange Adaptor	Emulator Connector	Target Connector	Space Adaptor	Mount Adaptor
BGA484 D1M2H	RTE7701412CBF484T000R	—	BSSOCKET484Z2627RE21N	CSSOCKET484Z2627RE03 CSSOCKET484Z2627RE04	LSPACK484Z2627RE01
BGA376 D1M2	RTE7701410CBF376T000R	—	BSSOCKET376Z2223RE21N	CSSOCKET376Z2223RE03 CSSOCKET376Z2223RE04	LSPACK376Z2223RE01
BGA272 D1M1A D1M1H	RTE7701461CBF272T000R	—	BSSOCKET272Z2021RE21N	CSSOCKET272Z2021RE03	LSPACK272Z2021RE11 LSPACK272Z2021RE12 *1
QFP176 D1M1 D1L2H	RTE7701463CFK176T000R	QB-176GM-YQ-01T	QB-176GM-NQ-01T	QB-176GM-YS-01T	QB-176GM-HQ-01T
QFP144 D1L2	RTE7701422CFK144T000R	QB-144GJ-YQ-01T	QB-144GJ-NQ-01T	QB-144GJ-YS-01T	QB-144GJ-HQ-01T
QFP144 D1L1	RTE7701421CFK144T000R	QB-144GJ-YQ-01T	QB-144GJ-NQ-01T	QB-144GJ-YS-01T	QB-144GJ-HQ-01T

Notes: 1. Contact Tokyo Eletech Corporation and select which adaptor to use according to the assignment of components on the target system.

1.3 Functional overview

IE850 is provided with a wealth of debugging functions to enable efficient program debugging, in addition to being used to emulate the operation of a target device. An overview of the functions is provided in this section.

Some functions are not supported, depending on the debugger to be used. See also the manual of the debugger to be used to confirm.

Table 1-3 System Specifications

Parameter		Specification
Program execution	Real-time execution	Available
	Step execution	Available
Reset	Forced reset	Available
Break	Software break	In ROM and RAM areas combined: 2000 points
	Hardware break	12 points including those used for both execution and CPU access conditions (8 points only for execution conditions, and 4 points for either execution or access conditions)
	Forced break	Available
	Trace-full break	Available
	Trace delay break	Available
Event	Number of events that can be set	8 points for execution, 8 points for CPU access, and 4 points for DMA access
	Available function	Break, trace, performance measurement
	Combination of events	OR, sequential
Trace	Priority in trace acquisition	Real-time trace mode Non-real-time trace mode
	Trace memory capacity	IE850 main unit: 9 Mbytes (512-K frames) QB-V850E2-SP (long-term trace option): 2.25 Gbytes (128-M frames) One frame = one set of branch information (branch-source and branch-destination PC values)
	Recorded data	Branch information, number of cycles for data access and for DMA access, software trace, time stamps
	Conditions to start and end recording to trace memory	Halting execution of the program or setting events
	Conditions for tracing data	Event condition
	Modes for recording to trace memory	Ring mode (overwrite mode), trace-full stop mode, trace-delay stop mode, trace-full break mode, trace-delay break mode

Parameter			Specification
Performance measurement functions	Time (1)	Measurement section	From run to break
		Items measured	Execution time
		Performance	32-bit counters
	Time (2)	Measurement section	From run to break, or between two event points
		Items measured	Execution time, total execution time, pass count, maximum execution time, minimum execution time
		Performance	32-bit counters (for three sections)
	Other than time	Items measured	Number of instructions executed (all or branches only), number of interrupts accepted (EI level or FE level), number of exceptions accepted (instruction asynchronous or instruction synchronous), clock cycles (all, while interrupts are inhibited, or other than for the processing of interrupts), number of instruction fetches requested, number of hits on the instruction cache
		Measurement section	From run to break, or between two event points
		Items measured	Maximum value, minimum value, latest value, total value, pass count
		Performance	32-bit counters (for four sections)
Masks	Masking resets		Possible
	Masking resets through the reset pin		Not possible
	Masking the PWRGD signal		Possible: The PWRCTL signal in the debug chip is looped back and input to the PWRGD signal on the pod.
Pseudo real-time RAM monitoring			Available (occupies a bus (steals cycles)) *1
Direct memory modification			Available (occupies a bus (steals cycles)) *1
Peripheral breaks			Available *2
Activating the settings of the Intelligent Cryptographic Unit (Slave type) (ICUS)			Possible (not possible if the settings of the ICUS is not activated)
Emulator detection by user programs			Available *3 Debugging startup register Initial value: 0000 0000_H Address: FA00 2078_H (CPU1)

Notes: 1. Only available for the general local RAM area.

2. The function to stop peripheral I/O operation in a break is called the peripheral break function.
Whether peripheral emulation functions are set or not is determined by the debugger.
Refer to the manual for the debugger you are using for how to set them.
Refer to the manual for the MCU you are using to check whether peripheral emulation functions are set.

3. For this function, any 32-bit value which is debugging information from the debugger is specified and held in the debugging startup register while the emulator is connected. This function can be used to determine the state of the emulator being connected or not from within user programs (refer to cautionary note 4.2.29).

1.3.1 Program execution function

The program execution function enables program execution equivalent to that of the target device. The executed program can be stopped under various conditions by using the break functions. The operation of only a function can be checked by executing a program, because a program can be executed from any address.

Single stepping (executing instructions one by one in units of assembly instructions) and source-level stepping (executing lines of C-language source code one by one) are available as types of stepped execution.

1.3.2 Reset

The CPU can be reset from the debugger with the forced reset function. The function is used to start program execution from the reset vector or to initialize the CPU during debugging.

1.3.3 Break (Program execution stop)

The break functions are used to stop program execution. With the IE850, program execution can be stopped under the following various conditions. See (1) to (4) for an overview of each break function.

Table 1-4 Conditions for Breaks and their Types

Condition for break	Type of break
Stopping after executing the instruction at an address	Hardware break function Software break function
Stopping after access to a variable	Hardware break function
Forcibly stopping	Forced break
Stopping if trace acquisition matches a certain condition	Trace-full break function Trace delay break function

(1) Hardware break

The hardware break function is used to set a break for a specific fetch or access operation. For example, a break is set by detecting a state where an address has been executed or a variable has been accessed.

(2) Software break

The software break function is used to set a break when a specific address has been executed (fetched). A break is set by temporarily replacing the instruction code of the specified address with the instruction for a break.

Caution Using the software break function for the on-board flash memory leads to rewriting of the on-board flash memory whenever a break is set, deleted, or executed. This reduces the number of cycles of rewriting relative to cases where the software break function is not used.

(3) Forced break

This function is used to forcibly stop a program when a user wants to stop a program.

(4) Other types of breaks

- Trace-full break function

This function is used to stop a program when the trace memory is full.

- Delay trigger break

This function is used to stop a program after acquiring a certain amount of trace data by hitting a certain event condition.

1.3.4 Event (Detection of specific operations)

The event function is used to detect specific fetching or other access by monitoring the bus cycle of the CPU and the external master such as DMAC. Operations where the instruction at an address is executed by the CPU or access to a variable is made by the CPU or the external master can be detected. Such specific operations are referred to as events. The event function is used by the break function, trace function, and performance measurement function.

1.3.5 Trace (Program execution history)

The trace function is used to check the execution history (trace information) of the CPU and the external master such as DMAC. Various types of functions shown from (1) to (4) can be used in the IE850.

(1) Priority of trace data acquisition

This feature enables selecting the priority for the acquisition of trace data.

Table 1-5 Priority of Trace Acquisition

Priority of Trace Acquisition	Descriptions
Real-time trace mode	Mode that gives priority to the program execution. If trace output is not fast enough, output of the trace information is temporarily halted. Thus, although the program runs in real time, trace information might be lost. This depends on the program being executed.
Non-real-time trace mode	Mode that gives priority to trace acquisition. If trace output is not fast enough, operation of the CPU is temporarily halted. Thus, the loss of trace information is suppressed* but the program loses the characteristic of real-time operations. **Caution *: Using this mode will always decrease the degree of information loss compared with the real-time trace mode. However, losing information might still not be avoidable. Whether this is so depends on the program being run.**

(2) Types of trace data

Types of trace data (trace information that can be acquired) are listed below.

Table 1-6 List of Types of Trace Data

Type of trace data	Descriptions
Branch source PC value Branch destination PC value (PC stands for program counter)	Branch source and branch destination PC values can be recorded in the history. The debugger complements the program with the instructions that would otherwise be executed between branch points and displays the complemented data. Thus, a program that was practically executed could be confirmed. **Caution: Execution history that can be displayed depends on the number of branches acquired.**
Access-execution PC Access address Accessed data	History of the PC values for executed access instructions for the memory or peripheral I/O register, access addresses, and accessed data from the CPU can be recorded. A history of reading and writing can also be recorded. **Caution: Access to the program registers of the CPU (such as r1 and r2) and system registers cannot be recorded in the history.**
DMA access cycle	History of access addresses and accessed data for the memory or peripheral I/O register from the DMA can be recorded. A history of reading and writing can also be recorded.
Time stamp (time tag)	Elapsed time between traced data can be added to each trace datum. The CPU clock is used in measurement for the time stamps.

(3) Conditions for recording in trace memory

This feature allows you to select how the trace memory is utilized.

Table 1-7 List of Conditions for Recording in Trace Memory

Conditions for recording in trace memory	Descriptions
Ring mode (overwrite mode)	New information is written over old information so that the latest information is consistently acquired. Trace acquisition continues until a break is reached.
Trace-full stop mode	When trace memory becomes full, further trace information is not acquired. The program continues to run.
Trace-full break mode	When trace memory becomes full, further trace information is not acquired and the program is stopped.
Delay trigger stop mode	The latest information is consistently acquired by writing new information over old information until a specific condition occurs. After the specific condition occurs, the selected amount of trace information is acquired, after which further trace information is not acquired. The program continues to run.
Delay trigger break mode	The latest information is consistently acquired by writing new information over old information until a specific condition occurs. After the specific condition occurs, the selected amount of trace information is acquired, after which further trace information is not acquired. The program is also stopped.

1.3.6 Performance measurement

Performance measurement is for measuring execution time from the time a program is started until it is stopped, the execution times of specific segments, or the items listed in Table 1-3, System Specifications.

1.3.7 Masks

Masks are used to help in debugging when the generation of an unintended reset might be thought possible or doubts exist about a signal level on the target system. The following items can be masked.

(1) System resets

System resets are completely masked (both the pin reset and internal resets). Masking is enabled by a setting in the debug chip (microcontroller).

Caution Masking only the pin reset is not possible with this pod, although this is selectable in some debuggers.

(2) PWRGD signal

Masking of the PWRGD signal on the target system leads to the PWRCTL signal in the debug chip (microcontroller) being looped back and input to the PWRGD signal on the pod.

1.3.8 Real-time RAM monitor

Display of on-board RAM areas during program execution is possible.

1.3.9 Change of direct memory

On-board RAM can be changed to a desired value during program execution.

1.3.10 Peripheral break

When the break function has been used to stop program execution, peripheral functions other than the watchdog timer continue to operate in general, but some peripheral functions can be stopped by using the peripheral break function. Refer to the user’s manual of the target device for information on the peripheral functions.

1.4 Regulatory notices

●European Union regulatory notices

This product complies with the following EU Directives. (These directives are only valid in the European Union.)

CE Certifications:

This product complies with the following European EMC standards.

・EMC Directive (2014/30/EU)

EN 55032 Class A

WARNING: This is a Class A product. This equipment can cause radio frequency noise when used in the residential area. In such cases, the user/operator of the equipment may be required to take appropriate countermeasures under his responsibility.

EN 55024

Information for traceability:

・Authorised representative & Manufacturer

Name: Renesas Electronics Corporation

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

・Person responsible for placing on the market

Name: Renesas Electronics Europe GmbH

Address: Arcadiastrasse 10, 40472 Dusseldorf, Germany

・Trademark and Type name

Trademark: Renesas

Product name: IE850 Emulator

Type name: RTE7701460EPA00000R

Environmental Compliance and Certifications:

・Waste Electrical and Electronic Equipment (WEEE) Directive 2012/19/EC

●United States Regulatory notices on Electromagnetic compatibility

This product complies with the following EMC regulation. (This is only valid in the United States.)

FCC Certifications:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

CAUTION: Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

1.5 Overview of emulation

An internal block diagram of this pod is shown below.

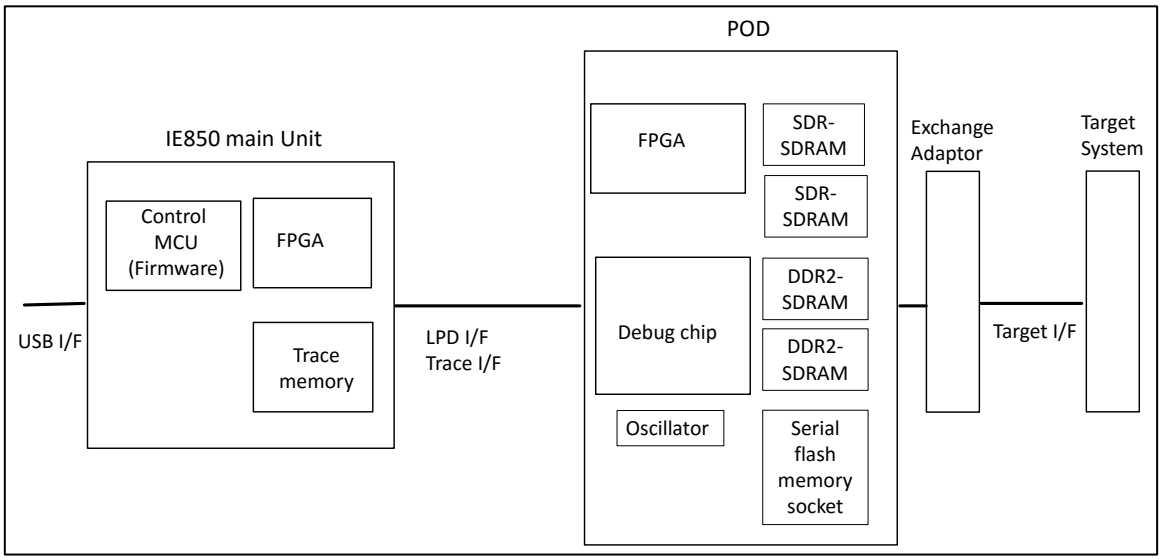


Figure 1-5 Overview of Internal Blocks

Debug chip:
The debug chip is used to emulate the operation of the individual target devices of the RH850/D1x series.

Oscillator:
In consideration of the oscillation characteristics, the oscillator on the pod provides the main and sub-oscillator signals instead of the oscillator on the target system. That is, the debug chip is driven by the oscillator on the pod. For details, refer to section 3.3.2, Replacing the clock.

Serial flash memory socket:
This socket can be used to mount serial flash memory. Due to the high-speed interface, the debug chip may have difficulty in access to the serial flash memory. In such cases, it is possible to place a memory circuit which is equivalent to the serial flash memory on the target system in this memory socket and emulate access to the serial flash memory.

SDR-SDRAM:
Since the SDR-SDRAM interface for the D1M1H is high speed, access to the SDR-SDRAM on the target system by the debug chip via a socket such as the EA is difficult. Mounting SDR-SDRAM (IS42S16320F-6BL from ISSI) on the pod and using it as if it were memory on the target system enables emulation of the SDR-SDRAM interface. In this case, there is no access to the SDR-SDRAM on the target system.

DDR2-SDRAM:
The DDR2-SDRAM interface for the D1M2(H) series is also high speed, access to the SDR-SDRAM on the target system by the debug chip via a socket such as the EA is difficult. Mounting DDR2 memory (MT47H128M16-25E-C from Micron) on the pod and using it as if it were memory on the target system enables emulation of the DDR2-SDRAM interface. In this case, there is no access to the DDR2 memory on the target system.

1.6 Package contents

RTE7701460EPA00000R package includes the items below. Confirm the items in the attached contents of the package.

Products supplied with RTE7701460EPA00000R:

- Pod
- Table of Toxic and Hazardous Substance and Elements
- Ferrite core

2. Names and Functions of Hardware

The following shows the names of pod and IE850 hardware units and their features.

2.1 Pod

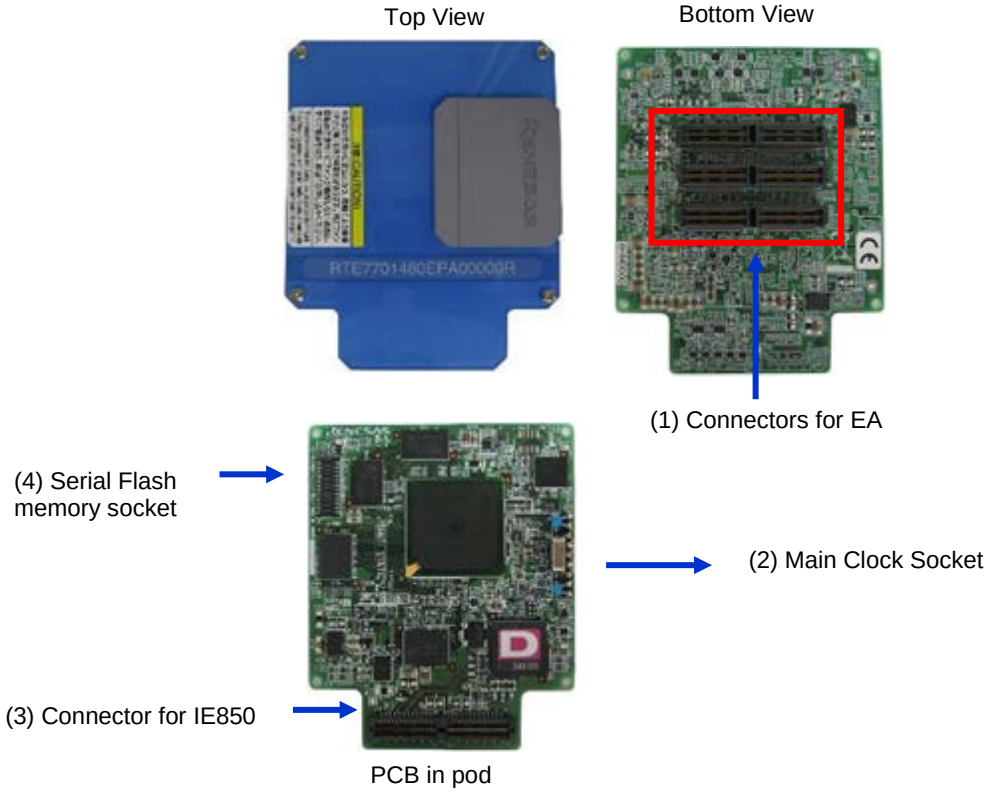


Figure 2-1 Names of Parts of Pod

(1) Connector for EA

This is the connector for connecting to the exchange adaptor (EA).

(2) Clock Socket (SOC1)

This is the socket for the main oscillator. An 8-MHz oscillator is mounted upon shipment.

(3) Connector for the IE850

This is the connector for connecting the pod cable of the IE850 main unit.

(4) Serial flash memory socket (SOC2)

This socket is for mounting serial flash memory on the pod. All pins of the socket are short-circuited to each other by shunts in the product as shipped. When flash memory is not mounted, be sure to leave the pins of this socket short-circuited. When flash memory is mounted, refer to the equivalent circuits in appendix A.

2.2 IE850 main unit

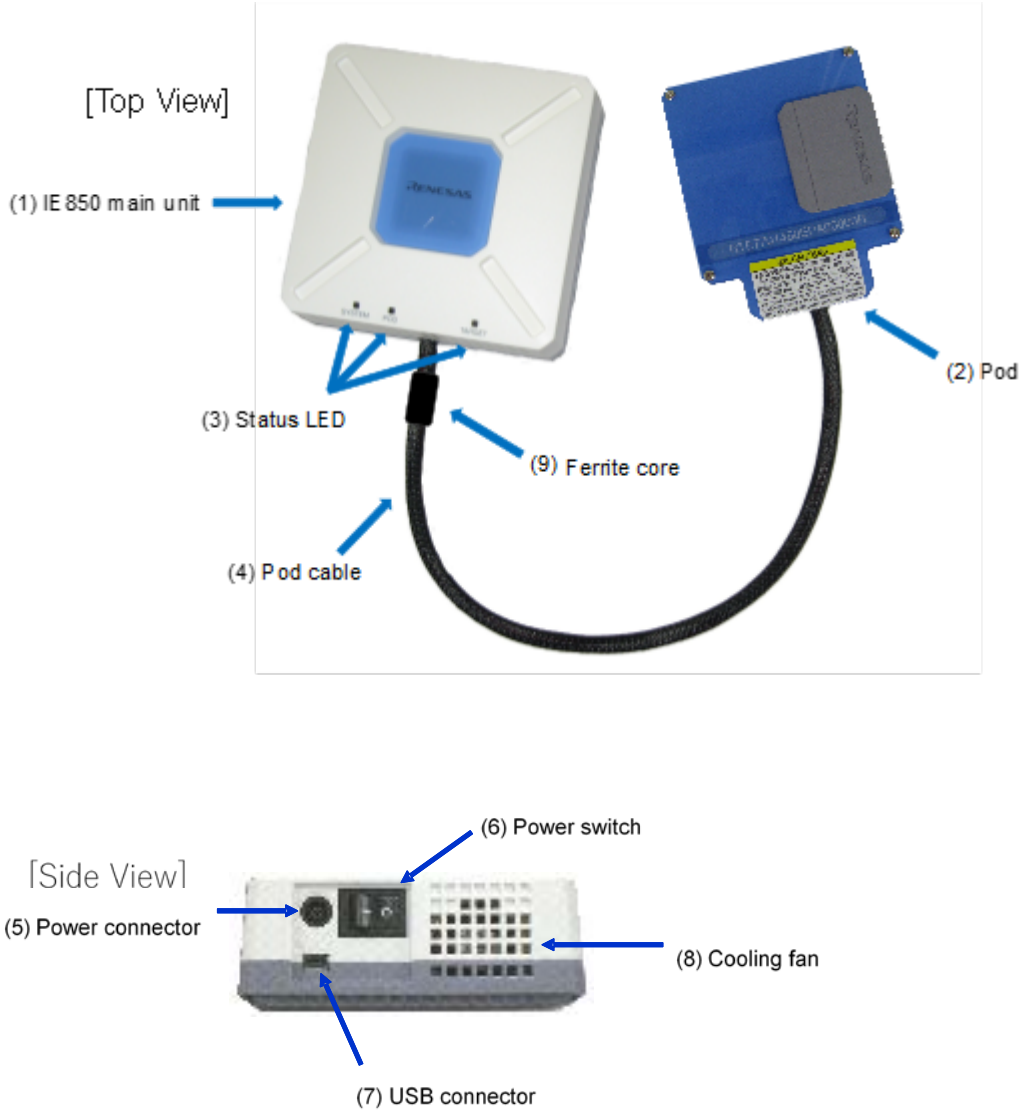


Figure 2-2 Names of Pars of IE850

(1) IE850 main unit

The IE850 main unit is the main controller of debugging. The IE850 main unit is sold separately. The control program (firmware) and FPGA data will need to be rewritten in accord with the pod to be connected. For more information on rewriting, refer to http://www.renesas.com/ie850.

(2) Pod

This product. Please refer to section 2.1.

(3) Status LED

The status LEDs turn on or blink according to specific causes as described in the table below. If any LED does not turn on or not blink, IE850 main unit might be broken. In this case, contact a Renesas Electronics sales representative or distributor.

LED name	Description
SYSTEM	This LED turns on when the power switch is turned on. This LED blinks if the FPGA in the IE850 main unit is not running correctly. In this case, the IE850 main unit might be broken.
POD	This LED turns on when communication with the emulation pod is established.
TARGET	This LED turns on when the target system is turned on.

(4) Pod cable

This coaxial cable is used to connect the IE850 main unit and emulation pod. The cable length is 37 cm. Be careful not to excessively bend this cable because doing so might break wires in the cable.

(5) Power connector

This connector is for the AC adaptor.

(6) Power switch

This switch turns the power on and off. Press the “|” side to turn on the power or the “O” side to turn off the power.

(7) USB connector

This connector is for a USB cable.

(8) Cooling fun

This fan cools down the IE850 main unit internal units. Be careful not to obstruct the vents.

(9) Ferrite core

This component serves as a countermeasure against EMI.

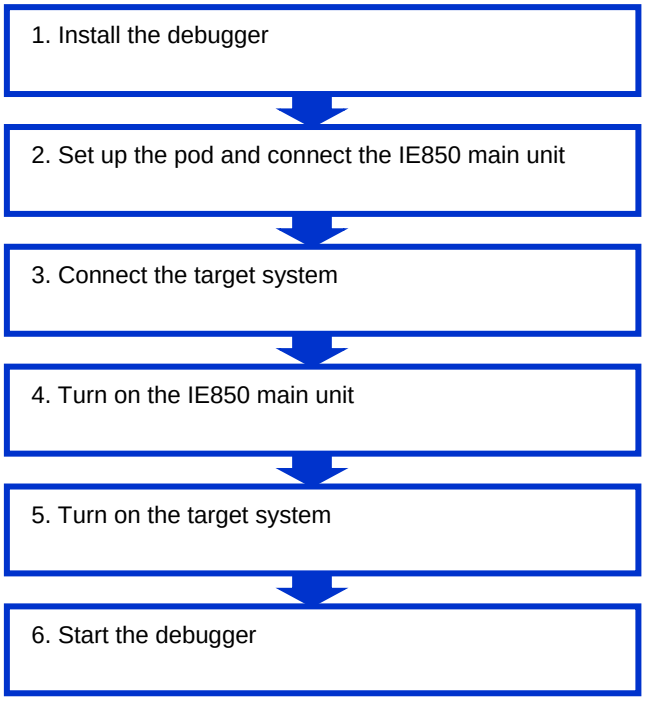
3. Setup Procedure

This chapter explains the IE850 main unit and the procedures for setting up the pod and for connecting the main unit via the pod to the target system. Since connecting only the IE850 main unit and pod allows you to start a debugger, you are then able to start developing a user program. Proceed with setting up to suit your field of application.

Setup can be completed by performing installation/setup in the order in which it appears in this chapter.

Perform setup along the lines of the following procedure.

To shut down the system, refer to section **3.8** Shut down procedure **.**



3.1 Installing the debugger

Install the necessary debugger before setting up the hardware.

See the user's manual of the debugger for installation instructions.

"Debugger" here refers to an integrated development environment such as CS+ or MULTI integrated environment from Green Hills Software.

3.2 Setting up the IE850 main unit

Rewrite the control program (firmware) and FPGA data in the IE850 main unit according to the pod in use. When you are using this pod, the firmware must be overwritten with the latest data for the RH850.

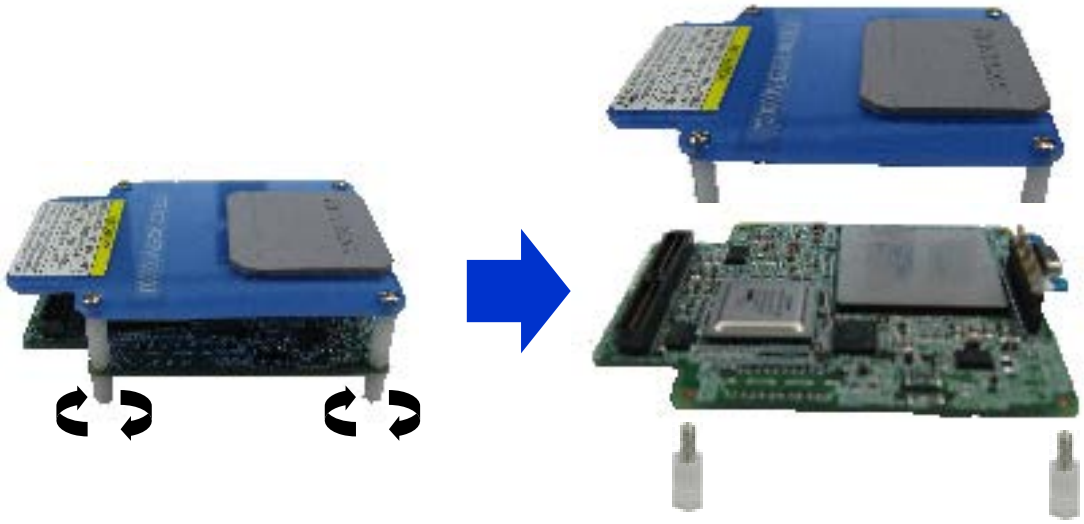
Check the information on the IE850 on the Web site: https://www.renesas.com/en-us/products/software-tools/tools/emulator/ie850-formerly-iecube2.html

3.3 Setting up the pod and connecting the IE850 main unit

Set up the clock on the pod and connect the IE850 main unit.

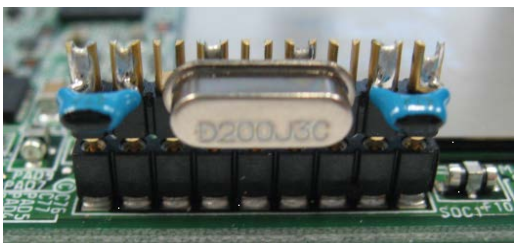
3.3.1 Removing pod cover

Remove the pod cover as shown below.



3.3.2 Replacing the clock

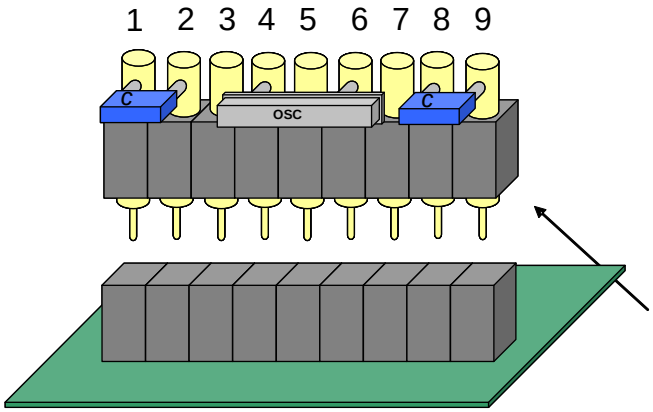
The main-oscillator clock is generated by the oscillator on the pod, as shown in the figure below.



This product has an 8-MHz resonator. If the oscillator on the target system has a different frequency, the oscillator components can be changed so that it becomes the same by setting up the oscillator with reference to the positions of the resonator and capacitors as shown below.

Caution: This product does not support clock input from an oscillator on the target system.

The 32.768-kHz clock in the pod is used as the subclock.



1-2: Capacitor
3: Power for the resonator
(Unavailable when the resonator is not in use)
4-6: Resonator
8-9: Capacitor
Pins 5 and 7 are open circuit

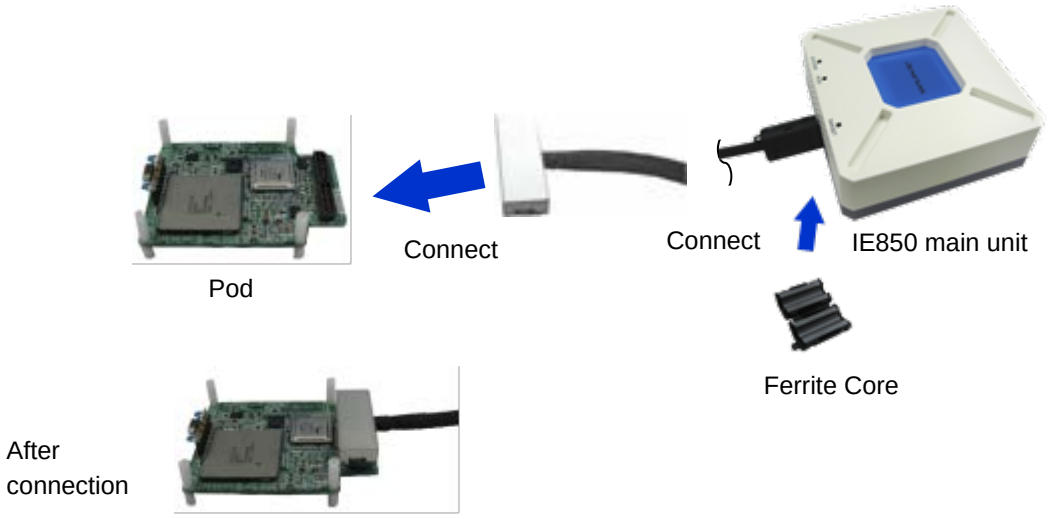
To remove the module, use a tool such as a precision screwdriver.

Additional information: You can open the lid on the top of the pod, as shown in the figure below to set the clock.

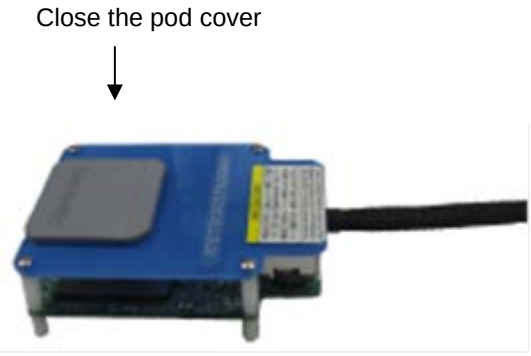
Clock settings can be made after opening the pod cover.

3.3.3 Connecting the IE850 main unit to the pod

Connect the ferrite core to the IE850 main unit and the IE850 main unit to the pod as shown in the figure below.



Finally, close the cover of the pod as the last step in connecting it to the IE850 main unit.

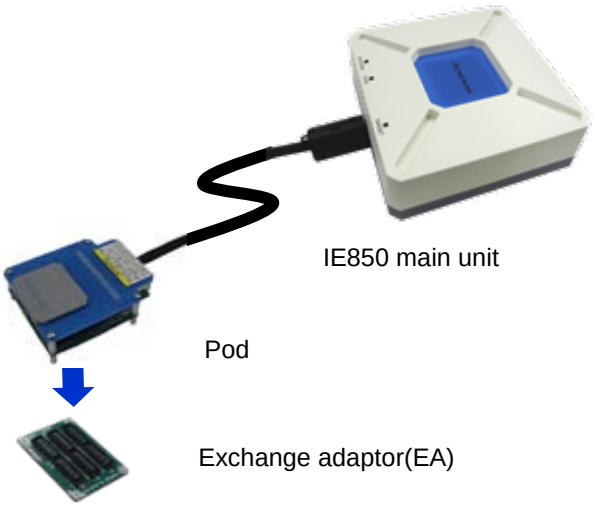


3.4 Connecting the target system

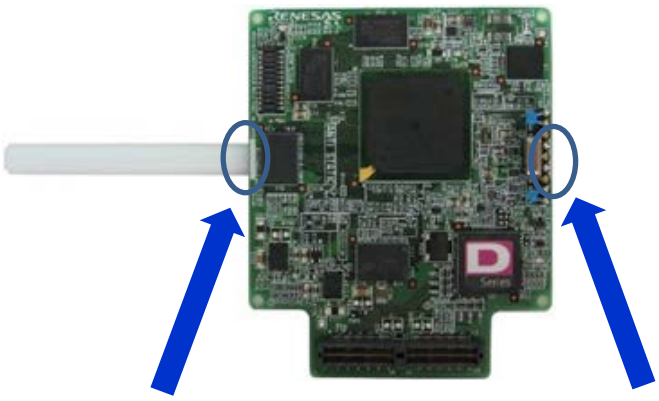
This section describes the overall process of connection to the point where the target system is connected.

3.4.1 Connecting the pod and target system

Connect sockets such as the exchange adaptor (EA), emulator connector (EC), and target connector (TC) to the pod for connection to the target system.
Refer to the user's manual for the exchange adaptor (EA) supporting the target device for more information on the connection of the emulator connector (EC) and target connector (TC).
Once the pod and target system are connected, make sure that the pod cable is not excessively bent. An image of connecting the IE850 main unit and pod to the exchange adaptor (EA) is shown below for reference.



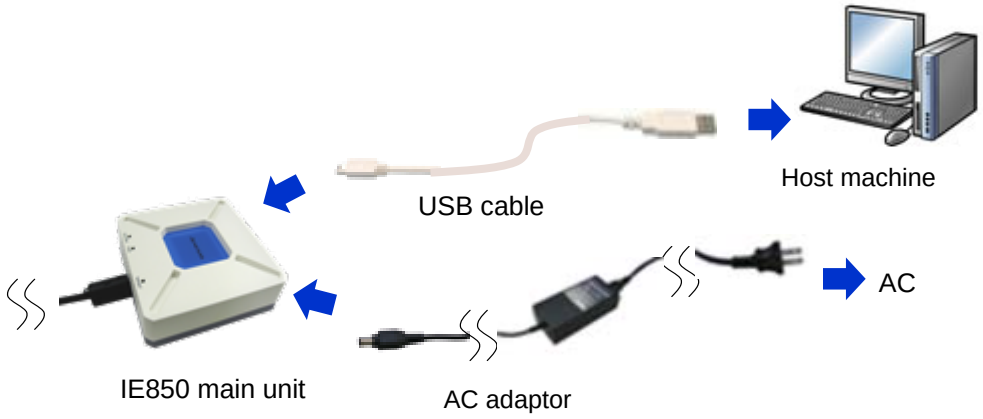
When the pod is removed from the EA, insert the stick for this purpose between the pod and the EA according to the silk-screen marking indicating the position for inserting the stick to remove the EA, and lift up the pod little by little with gentle leverage. Take care to insert the stick in the correct position so that it does not dislodge components on the back side of the pod. The stick for removing the EA comes with the IE850 main unit (QB-V850E2).



The silk-screen marking for inserting the stick to remove the EA

3.4.2 Connecting the USB cable and AC adaptor

Connect the USB cable and power supply adaptor as shown below. At this time, make sure that the IE850 main unit is not on.



[Side View]

3.5 Turning on the IE850 main unit

Turn on the IE850 main unit. At this time, make sure that the target system is not on.

After the power is initially turned on, the SYSTEM and POD LEDs light up. If these LEDs blink or remain off, the IE850 main unit might be broken. In this case, contact a Renesas Electronics sales representative or distributor.

[Side View]

[Top View]

3.6 Turning on the target system

After the power is on, the TARGET LED lights up. If the LED remains off, connectors might be connected poorly, or might be broken. Confirm if the connectors are poorly connected. If this is not the case, contact a Renesas Electronics sales representative or distributor.

[Top View]

After the power is on, the TARGET LED lights up.

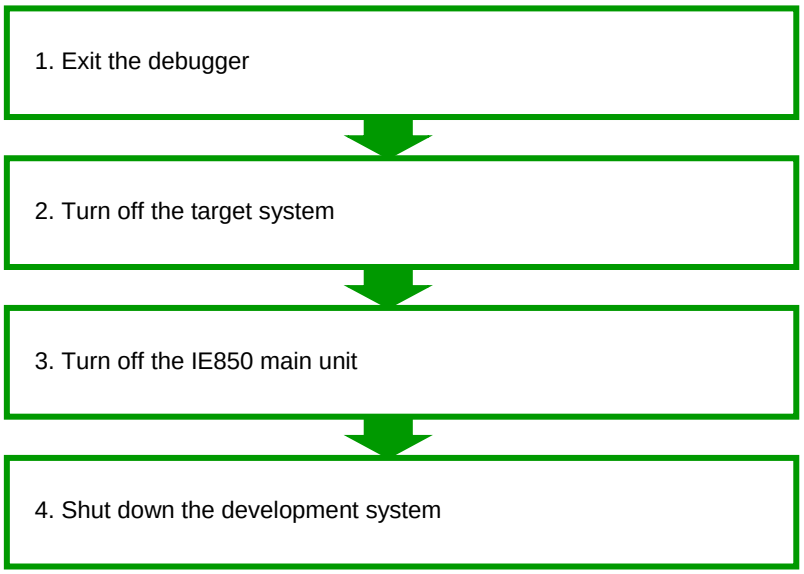
3.7 Starting the debugger

After the above procedure, start the debugger.

For details about debugging procedures, see the user's manual of the debugger.

3.8 Shut down procedure

Shut down the development system according to the procedure below.



4. Notes

This chapter explains the common notes of the IE850.

4.1 Notes on differences between actual device and emulator

When connecting the emulator and target system for debugging, the emulation duplicates the behavior of an actual device on the target system to the greatest extent possible, but the following differences do exist between the behavior of the emulation and actual device. We therefore urge you to perform an evaluation on the actual device as the final evaluation step before mass production. It is your responsibility to ensure that the target system is suitable.

4.1.1 Behavior after target system is powered on

After turning on power to the device mounted on the target system, the program runs following release from the reset state. The emulator, however, will not start the program until the debugger has downloaded the program and performed a start-execution operation.

Also, although the emulator is able to download and run object code before the initial variable values and other information have been ROMized, the actual device will not function normally if the object code is not ROMized.

4.1.2 Downloaded programs

Programs are downloaded to the flash memory of a debugging chip mounted on a pod. In order to run the program successfully, however, you should always download the program before starting debugging.

4.1.3 Function of on-chip debugging emulator

Debugging cannot proceed while an on-chip debugging emulator is connected to the target system.

4.1.4 DBTRAP instruction

The DBTRAP instruction is used for software break and thus cannot be used in programs with the emulator.

4.1.5 Serial programming function

The serial programming function cannot be used with the emulator during debugging.

4.1.6 Halt mode (skipped number)

The information that was previously under this number has been integrated into 4.2.11.

4.1.7 Current drawn

The emulator might in fact draw less current than the actual device since a part of power is provided by an emulator. Thus, the customer should use the actual device in a final evaluation before mass production and judge the suitability of adopting the product.

4.1.8 ECC errors

When an emulator is started, the local RAM area is initialized to H'0000 0000. This leads to the following difference from the actual device.

- The initial values in RAM immediately after starting an emulator are different from the initial values (undefined values).
- ECC errors due to non-initialization of RAM are not detected with the emulator.

To emulate ECC errors, set the following option.

- The RAM area is not initialized when the emulator is started.

Note that ECC errors in the working RAM area (FEBF DA00H to FEBF FFFFH) may not be detected because this area is initialized by the emulator as part of starting emulation. This is the case even if the option for not initializing the RAM area when the emulator is started has been selected.

4.1.9 OTP flag

Do not set the one-time programming (OTP) flag in self-programming with the emulator. Note that setting of the flag makes downloading from the debugger to flash memory impossible.

4.1.10 Operations in response to resets and interrupts when an emulator is in use (skipped number)

The information that was previously under this number has been integrated into 4.2.9 and 4.2.10.

4.1.11 Option byte register

The debugger cannot write new values to the bits of the option byte register indicated below since they are used by the emulator. Also, do not attempt self-programming to write new values to these bits.

- OPJTAG1 and OPJTAG0 bits (bits 17 and 16 of the OPBT0 register)

  The value of OPJTAG1 and OPJTAG0 bits is always 11B.

4.1.12 Oscillator

The emulator does not support clock input from an oscillator on the target system. The main clock oscillator on the pod is used as the main clock (refer to section 3.3.2, Replacing the clock). The 32.768-kHz clock in the pod is used as the subclock. Proceed with final evaluation to confirm operation of the target system after mounting the actual device and while using the oscillator on the target system.

4.1.13 SDR-SDRAM interface

The emulator does not support access to SDR-SDRAM on the target system. The SDR-SDRAM (IS42S16320x*-6BL (*: version D or later)) from Integrated Silicon Solution Inc. is installed in the pod.

4.1.14 DDR2-SDRAM interface

The emulator does not support access to DDR2-SDRAM on the target system. Two DDR2-SDRAMs (MT47H128M16RT-25E x 2) from Micron Technology, Inc. are installed in the pod.

4.1.15 REG0VCC,OSCVCC,ISOVDD,REG1VCC,PLLVCC,SDRBVCC,SDRBCKVCC, and SDRAVCC pins

REG0VCC,OSCVCC,ISOVDD,REG1VCC,PLLVCC,SDRBVCC,SDRBCKVCC, and SDRAVCC pins are provided from the emulator on the pod. These pins cannot be supplied from the target system.

4.1.16 REG0C, REG1C1, REG1C2, REG1C3, and REG1C4 pins

Stabilizing capacitors are connected to the stabilizing capacitor connection pins (REG0C, REG1C1, REG1C2, REG1C3, and REG1C4) on the pod. These pins are not connected on the target system.

4.1.17 VSS pins for each power subsystem

The following VSS pins are connected to the common GND in the emulator.

PLLVSS, A0VSS, B0VSS, B1VSS, B2VSS, B3VSS, B4VSS, B5VSS, MVSS, SFVSS, ZPDVSS, EVSS,

REG0VSS, REG1VSS, RVSS, ISMVSS, OSCVSS, SDRAVSS, ISOVSS, SDRBVSS, and SDRBCKVSS.

4.1.18 A/D converter

Results from the A/D converter differ from those on the actual device because of the exchange adaptor etc. between the debug chip and the target system.

4.1.19 Power-On-Clear Reset 0 (POC0RES)

Since POC0RES in response to REG0VCC dropping below a certain level is only pseudo-emulated, the timing of reset operations differs from that on the actual device.

Also, when power to the target system is restored, the value in the reset source register will differ from that of the actual device.

4.1.20 Power-On-Clear Reset 1 (POC1RES)

The pod is not capable of emulating the generation of POC1RES in response to REG1VCC dropping. Even if REG1VCC drops below the required level, POC1RES will not be generated.

4.1.21 ISOPWRES

The pod is not capable of emulating the ISOPWRES reset. Do not set the PWRGD pin to the low level other than in DEEPSTOP mode.

4.1.22 PWRCTL pin

The state of the PWRCTL pin in the POC may differ from that in the actual device. For details, refer to appendix A.

4.1.23 FCU RAM

The FCU RAM cannot be emulated.

4.1.24 OCTA

The OCTA cannot be emulated.

4.1.25 Serial RGB

The serial RGB cannot be emulated.

4.1.26 Hyper Bus

The maximum operating frequency available for emulation is 60 MHz.

4.1.27 SFMA0

The pod is capable of emulation at the maximum operating frequency of SFMA0.

However, since SFMA0 is a high-speed interface, access to serial flash memory on the target system may be difficult. This depends on the state of the target system.

In cases where difficulties arise, configure the pod with a memory circuit equivalent to the flash memory on the target system in the serial flash memory socket (SOC2).

The pod is not capable of emulating setting of the driving ability of SFMA0, for which the setting is in the DSCTRL register.

4.1.28 SFMA1

The operating frequency that is available for emulation is max. 40 MHz (SDR interface).

4.1.29 SFMA2

The pod is not capable of emulating SFMA2.

4.1.30 NANDC

The pod is not capable of emulating burst transfer to code flash memory, data flash memory, and local RAM of the NANDC.

4.1.31 Error Control Module(ECM)

The pod is not capable of emulating the INTECM error input of the ECM.

4.1.32 Performance Monitor

Assignment of the ID for the performance monitoring function of the debug chip differs from that of the actual device.

PMRID[3:0] PMWID[3:0]	Assignments of the debug chip
0	0: CPU Subsystem(DMA, OCD)
1	1: CPU Subsystem(CPU)
2	2: JCUA
3	3: ETNB
4	4: MLB
5	5: RLE/Sprite Unit0 and 1
6	6: RLE/Sprite Unit2 and 3
7	7: GPU2D (D1M2 emulation)
8	8: GPU2D (other emulation)
9	9: NANDC

4.1.33 Open LDI

Assignments of the open LDI function to port pins of the debug chip differ from those on the RH850/D1M1A.

Port name	Assignments of the open LDI of the debug chip	Assignments of the open LDI on the RH850/D1M1A
P44_10, P44_11	OLDI0_CH1_P, OLDI0_CH1_N	-
P45_0, P45_1	OLDI0_CH0_CLKP, OLDI0_CH0_CLKN	OLDI0_CH0_CLKP, OLDI0_CH0_CLKN
P45_2, P45_3	OLDI0_CH2_P, OLDI0_CH2_N	OLDI0_CH1_P, OLDI0_CH1_N
P45_4, P45_5	OLDI0_CH3_P, OLDI0_CH3_N	OLDI0_CH2_P, OLDI0_CH2_N
P45_6, P45_7	OLDI0_CH4_P, OLDI0_CH4_N	OLDI0_CH3_P, OLDI0_CH3_N
P45_8, P45_9	-	OLDI0_CH4_P, OLDI0_CH4_N

When the open LDI is to be emulated in the pod, be sure to follow the procedure below.

- Select "● Open LDI Emulation" with the switch (SW for [P45_8, P45_9]) on the exchange adaptor RTE7701461CBF272T000R. This assigns P44_10 and P44_11 of the debug chip to the positions of the P45_8 and P45_9 pins of RH850/D1M1A on the exchange adaptor.
  When Open-LDI emulation is selected, the assignment of Open LDI pin functions to P44_10 and P44_11 of the RH850/D1M1A cannot be emulated. Do not use P44_10 or P44_11 on the target system.

- Set the selection of channels of the open LDI CH control register (OPLDI0CHCR) in the debug chip as follows. Change CH1, CH2, CH3, and CH4 to CH4, CH1, CH2, and CH3, respectively. This makes emulation of the open LDI available for the RH850/D1M1A.

  When the OPLDI0CHCR register is set as D1M1A_VALUE for the RH850/D1M1A, the settings for the debug chip can be calculated as follows.

  ((D1M1A_VALUE + 0x0003) & 0x0003) |
  ((D1M1A_VALUE + 0x0030) & 0x0030) |
  ((D1M1A_VALUE + 0x0300) & 0x0300) |
  ((D1M1A_VALUE + 0x3000) & 0x3000)

4.2 Cautionary notes on debugging

4.2.1 Hardware break (access) function (the timing of a break occurring)

When the hardware break (access) function is in use, a break in response to the reading or writing of specified data by a read–modify–write instruction will occur after the instruction. Other hardware breaks (access) occur before the instruction.

4.2.2 Hardware break (access) function (forms of read or write access that are not detectable)

In general, no break occurs even if a read or write access condition is satisfied by the instructions below.

(1) CAXI, SET1, CLR1, NOT1, and TST1

(2) PREPARE, DISPOSE, PUSHSP, POPSP, SWITCH, CALLT, and SYSCALL

However, the detection of reading by the instructions in (1) is possible, but only if no data condition is specified.

4.2.3 Hardware break (access) function (EIINT table)

Do not set the address of the EIINT table as a hardware break condition. If a break occurs, it will not be possible, in some cases, to return from the interrupt processing even if EIRET is executed.

4.2.4 Initialization of RAM areas

All RAM areas for use by a program must be initialized when an emulator is in use. Before the emulator is used, if any setting is made to initialize the RAM area when the emulator is started, ECC errors are not generated since the debugger initializes the RAM area. However, when the actual device is operated with a program which does not initialize the RAM area, ECC errors will be generated, preventing normal program operation.

ROMization is also required because any data downloaded from the emulator to the RAM area before program execution will also be initialized. For details, refer to the user's manual for the compiler you are using.

4.2.5 Reset of pins(skipped number)

The information that was previously under this number has been integrated into 4.2.9.

4.2.6 Trace function

The following restrictions apply to the trace function.

- Writing of data in the form of pushing by executing the PUSHSP or PREPARE instruction might not be traced.
- In the case of section trace, for example, the instruction immediately before the fetched instruction that actually caused tracing to start might be included in trace data.
- In some cases, acquired trace information will be lost. This depends on the program being executed. The lost

information cannot be restored, but the fact of the loss is indicated (displayed).

- When data-qualified tracing (point tracing), i.e. tracing only of data in access to a specific address, is specified, tracing proceeds with any data conditions ignored, even if read access conditions are set. Tracing is still governed by conditions other than data conditions.

4.2.7 Power-saving modes

When a power-saving mode is in use, the restrictions below apply.

- For debugging of a program, ensure that the program sets WUFMSK0[0] to 0.
- Release from the DEEPSTOP mode follows any of the following operations or conditions while the user program is being executed.
  - Break
  - Memory access
  - Setting an event
- The power supply to the Iso area (CPU, RAM, peripheral module, etc.) is not stopped in the DEEPSTOP mode during debugging. For this reason, RAM or registers which have undefined initial values retain these values. Be sure to initialize them after returning to the RUN mode.

4.2.8 PBG

When you are using an emulator, leave the PROTDEB bit of the FSGDxxDPROTn register in PBG with the setting 1 (which allows access by a debug master). Changing the bit to a value other than its initial value may lead to normal access to memory becoming impossible.

4.2.9 Resets when an emulator is in use

Table 4-1 shows the states of the device when an emulator is in use and the operation in response to a reset (i.e. a user-system reset) issued by the user system or the user program. During single-stepped execution, the emulator masks the user-system reset so that it can continue to emulate the source code of the program line-by-line rather than in realtime. For C-source-level stepped execution, the reset is masked in different ways depending on the debugger; single-stepped execution is used or the user program is executed by setting temporary breakpoints. Accordingly, this user's manual cannot define whether a reset is masked by the emulator or not, so refer to the user's manual for the debugger you are using.

Table 4-1 State of the Device and Masking of User-system Resets by the Emulator

		State of the device			
		During a break	**Single stepping**	**Executing the user program**	**Stepping at C-source level**
Reset mask specification of the debugger	**Not masked**	Masked*		Not masked	Depends on the debugger
	Masked	Masked*			

- When a reset is issued by the debugger (by using a reset button of the debugger, etc.), the CPU is always reset regardless of whether the masking of resets is currently enabled or disabled. After a reset from the debugger, breaks are generated in all CPUs.
- Resets generated in the states marked (*) in table 4-1 will be discarded. For example, when a setting for software-reset processing is made during single-stepped execution or a software reset by setting a register is applied by the debugger during a break, the reset is not applied.
- Do not allow the generation of a reset in the form of a pin reset from the target system other than while a program is in execution regardless of the presence of masking as described above. A reset generated while the program is running may cause the debugger to hang.

4.2.10 Interrupts when an emulator is in use

Table 4-2 shows the states of the device when an emulator is in use and the operation in response to an interrupt. During single-stepped execution, the emulator masks interrupts so that it can continue to emulate the source code of the program line-by-line rather than in realtime. For stepped execution of interrupt processing, set a breakpoint at the start of the interrupt service routine, then generate an interrupt during the execution of a user program so that the break at the start of the interrupt service routine is applied. For C-source-level stepped execution, interrupts are masked in different ways depending on the debugger; single-stepped execution is used or the user program is executed by setting temporary breakpoints. Accordingly, this user's manual cannot define whether interrupts are masked by the emulator or not, so refer to the user's manual for the debugger you are using.

Table 4-2 State of the Device and Masking of Interrupts by the Emulator

State of the device			
During a break	**Single stepping**	**Executing the user program**	**Stepping at C-source level**
Interrupts masked*		Interrupts not masked (operation is according to the settings of the user system)	Depends on the debugger

- Interrupts (EIINT, FEINT, and FPI) which have been generated in the state marked (*) in table 4-2 are held pending and interrupt processing is performed after the interrupt mask is removed.

4.2.11 HALT mode and stepped execution of the HALT instruction

If a break occurs, the device is released from HALT mode.
When a HALT instruction is encountered during single step execution (execution in units of assembly instruction), a break is set at the next instruction following the HALT instruction, and the mode does not change to the HALT mode.
When a HALT instruction is encountered during C-source-level stepped execution, whether or not the transition to the HALT mode proceeds depends on the facilities of the debugger.

4.2.12 Stepped execution of an instruction which would lead to a transition to the DEEPSTOP mode

Stepped execution has two variants: Single step execution (execution in units of assembly instructions) and C-source-level stepped execution (execution in units of statements or functions in C language source code). When an instruction which would lead to a transition to the DEEPSTOP mode on the actual device is executed during single step execution, the program breaks at the address at the time of the reset and does not switch to the DEEPSTOP mode. When an instruction which would lead to a transition to the DEEPSTOP mode on the actual device is executed during C-source-level stepped execution, whether or not the transition to the DEEPSTOP mode proceeds depends on the facilities of the debugger.

4.2.13 Cautionary note on connecting an emulator (pin reset)

The pin reset signal continuing to be asserted while communications between the emulator and MCU are being prepared when the emulator is connected causes incorrect communications. Thus, ensure that the reset signal does not remain asserted when the emulator is connected.

4.2.14 Access to I/O resources in the MCU

Access to I/O resources (registers and RAM) in the MCU by the debugger (i.e. access through the memory or I/O register window) proceeds in the same way as access from a user program.

Examples (for the actual operation of I/O resources, refer to the manual of the MCU you are using)

- Access to FCU-RAM resources

  Normal access will not proceed unless the FCU-RAM enable bit is set.
- Access to the PBG area

  Attempted access to the PBG area will not proceed while the guard is enabled. Also, this is within the scope of error detection.

4.2.15 Registers in peripheral modules

Ensure that a clock signal is correctly supplied to the serial flash memory interface A (SFMA), the HyperBus controller (HYPB), the OctaBus controller (OCTA), and the CANFD before any register in the SFMA, HYPB, OCTA, and CANFD is accessed from a debugger (via the I/O register window). Otherwise a timeout will occur in some cases.

4.2.16 Performance measurement

In the case of measuring a specific section, if the intervals between the start and the end of one measurement, and between the end of that measurement and the start of the next is short, the measurement might not be possible. To obtain correct measurements, the interval* should be long enough.

*: The required detection interval depends on the operating frequency and the LPD communications frequency of the MCU. The LPD frequency is 20 MHz.

4.2.17 Reset

If a CPU reset is generated while a program is running, debugging functions (breaks, events, tracing, and timer) and the operation of the MCU may become unstable.

4.2.18 Rewriting of on-chip flash memory (RAM)

The debugger uses the internal RAM area (working RAM: from FEBF DA00H to FEBF FFFFH) when rewriting the flash memory*. When the DMAC or an external master requires access to that area, change the working RAM area and set an area that is not used by the DMAC or the external master.

Note: Rewriting of flash memory proceeds in response to any of the operations below.

- Downloading to on-chip flash memory
- Changes to on-chip flash memory by using the [Memory] panel or the [Disassemble] panel
- Setting or cancellation of software breaks
- Re-execution after a software break is encountered (including stepped execution)

4.2.19 Rewriting of on-chip flash memory (clock monitor)

The debugger changes the PLL multipliers as shown below when the flash memory is rewritten*. Thus, rewriting the flash memory raises a possibility of the frequency becoming higher than that currently in use. If the frequency surpasses the upper limit which was set by the clock monitor (CLMA), this prevents rewriting of the flash memory. If the change in the clock frequency due to the debugger is a problem, set [Change the clock to flash writing] in the [Property] panel to [No].

[Changes in the PLL multiplier]

- D1M2(H) and D1M1A:

When MainOSC=8 MHz, frequencies are multiplied by 30 (CPU clock at 240 MHz).
When MainOSC=10 MHz, frequencies are multiplied by 24 (CPU clock at 240 MHz).
When MainOSC=12 MHz, frequencies are multiplied by 20 (CPU clock at 240 MHz).
When MainOSC=16 MHz, frequencies are multiplied by 15 (CPU clock at 240 MHz).

- D1M1H:

When MainOSC=8 MHz, frequencies are multiplied by 25 (CPU clock at 200 MHz).
When MainOSC=10 MHz, frequencies are multiplied by 20 (CPU clock at 200 MHz).
When MainOSC=16 MHz, frequencies are multiplied by 12.5 (CPU clock at 200 MHz).

- D1M1:

When MainOSC=8 MHz, frequencies are multiplied by 20 (CPU clock at 160 MHz).
When MainOSC=10 MHz, frequencies are multiplied by 16 (CPU clock at 160 MHz).
When MainOSC=16 MHz, frequencies are multiplied by 10 (CPU clock at 160 MHz).

- D1L1, D1L2(H):

When MainOSC=8 MHz, frequencies are multiplied by 15 (CPU clock at 120 MHz).
When MainOSC=10 MHz, frequencies are multiplied by 12 (CPU clock at 120 MHz).
When MainOSC=12 MHz, frequencies are multiplied by 10 (CPU clock at 120 MHz).
When MainOSC=16 MHz, frequencies are multiplied by 7.5 (CPU clock at 120 MHz).

Note: Rewriting of flash memory proceeds in response to any of the operations below.

- Downloading to on-chip flash memory
- Changes in on-chip flash memory due to operations in the memory panel
- Setting or cancellation of software breaks
- Re-execution after a software break is encountered (including stepped execution)

4.2.20 Rewriting of on-chip flash memory (register value)

The debugger causes PLL0 to oscillate and to be supplied to the CPU during writing to the flash memory*. Thus, the value of CKSC_IPLL0S_ACT.PLL0SACT will be changed from 0 (initial value) to 1. If the change of the bit's value creates a problem, set [Change the clock to flash writing] in the [Property] panel to [No].

Note: Rewriting of flash memory proceeds in response to any of the operations below.

- Downloading to on-chip flash memory
- Changes in on-chip flash memory due to operations in the memory panel
- Setting or cancellation of software breaks
- Re-execution after a software break is encountered (including stepped execution)

4.2.21 Breaks during execution of code for making clock settings

The flash memory cannot be programmed if a break occurs while the MCU is running code written to memory for making clock settings (setting of the main oscillator or PLL frequency divider and so on).

If you wish either of the following types of operation to proceed when a break has occurred during clock settings, set [Change the clock to flash writing] in the [Property] panel to [No].

a. Any operation that involves programming of the flash memory (e.g. re-downloading)
b. Setting or deleting software breakpoints

Also, do not set software breakpoints within code for making clock settings.

4.2.22 Event functions (64-bit access)

Do not set any access events with the condition in 64-bit units. The emulator may detect access in a unit other than 64 bits as satisfying such conditions or other events may not operate normally.

4.2.23 Event functions (in the order of event detection)

In the following cases, since the orders of instructions and event detection may not operate as set, to measure the time or performance in sequential events, section tracing, and desired sections may not be possible.

- An event is set for consecutive instructions but the two instructions are executed at the same time.
- An access event detects adjacent read and write instructions, since the timing of event detection differs in write and read access and the timing may be detected in the order of reading then writing, even though the instructions are executed in the order of writing then reading.

4.2.24 Event functions (bit-manipulation instructions)

When a read or write access condition is set for an event, the writing cycle of read-modify-write generated by a bit-manipulation instruction is not detected as an event. This condition cannot be used as a trigger for a break, trace acquisition, or performance measurement in the case of such instructions.

4.2.25 Satisfaction of two break conditions before a single break

If another read-access event is detected immediately before a transition to the break state due to a forced break or an event break, a further break occurs immediately after execution of the program is resumed because the break request was accepted as a read-access event at the time of resumption.

4.2.26 Software break function (RAM areas)

The software break function is implemented by replacing instructions. Thus, note that no break will occur if the value at an address where a software break has been set is rewritten by a user program which is running.

4.2.27 Pin-reset mask function

The function to only mask pin resets is not available.

4.2.28 Trace Ring Mode

To improve the rate and smoothness of the output of messages, the debug chip of this pod only outputs information on the differences between a previous trace message that is used as a reference and subsequent trace messages of the same type.

The debugger restores the trace messages from the reference messages and differential information, and then displays the full trace information.

The emulator circuits periodically issue requests for the output of reference messages by the debug chip.

Thus, in trace ring mode (overwriting mode), since trace messages to serve as references located around the top of trace memory may be overwritten, restoration from differential information subsequent to that may not be possible until a further reference message is in the buffer.

Note that some debuggers may not display trace messages which could not be restored.

4.2.29 Emulator detection by user programs

Even if debugging information is specified, note that the value will be initialized to 0000 0000H if a reset is generated. Debugging information is specified again when a break occurs in all CPUs and when the user program is re-executed after a reset.

5. Optional Product

5.1 Long term trace option

This chapter explains an optional product QB-V850E2-SP for extending the trace memory.

5.1.1 General

The QB-V850E2-SP is an option product that can expand trace memory for the IE850. Please confirm the supported versions of the debugger.

Figure 5-1 QB-V850E2-SP

5.1.2 Setup procedure

This section describes how to connect the QB-V850E2-SP to the IE850 main unit.

1. Remove the cover from the connector on the top side of the QB-V850E2-SP module. A screwdriver will be necessary to remove the cover.	
2. Remove the USB cable and power supply adaptor from the IE850 main unit, then remove the cover on the bottom side of the IE850 main unit.	
3. Connect the IE850 main unit on the QB-V850E2-SP as shown in the picture. Now connect the USB cable and power supply adaptor to the IE850 main unit.	

The IE850 automatically detects the expanded trace memory when the QB-V850E2-SP is connected. Please set the capacity of the trace memory in the debugger.

5.1.3 Cautionary note when using the QB-V850E2-SP

(1) Support of QB-V850E2-SP and debugger

The QB-V850E2-SP can only be used in the MULTI integrated environment provided by Green Hills Software.
It cannot be used in the integrated environments produced by Renesas for developing microcontroller software.

(2) Break when QBV850E2-SP is in use

When the QB-V850E2-SP is in use, the trace-full break function is not available.

6. Maintenance and Warranty

This chapter covers basic maintenance, warranty information, provisions for repair and the procedures for requesting a repair.

6.1 Maintenance

(1) If dust or dirt collects on this product, wipe it off with a dry soft cloth. Do not use thinner or other solvents because these chemicals can cause the surface coating to separate.

(2) When you do not use this product for a long period, disconnect it from the power supply, host machine, and target system.

6.2 Warranty

(1) This product comes with a one-year warranty after purchase.

Should the product break down or be damaged while you're using it under normal conditions in accord with its user's manual, it will be repaired or replaced free of cost.

(2) However, if the following types of failure or damage to the product occur during the term of the warranty, repairing or replacing the product will incur a cost.

a) Failure or damage attributable to the misuse or abuse of the product or its use under other abnormal conditions.

b) Failure or damage attributable to improper handling of the product after purchase, such as dropping the product while it is being transported or otherwise moved.

c) Failure or damage to the product caused by other pieces of equipment connected to it.

d) Failure or damage attributable to fire, earthquakes, thunderbolts, floods, or other natural disasters, or to abnormal voltages, etc.

e) Failure or damage attributable to modifications, repairs, adjustments, or other acts in relation to the product by parties other than Renesas Electronics Corp.

(3) Consumables (e.g., sockets and adaptors) are beyond the scope of repair and replacement.

In the above cases, contact your local distributor. If you are renting the product, consult the company you are renting it from or the owner.

6.3 Repair provisions

(1) Repairs not covered by warranty

Problems arising in products for which more than one year has elapsed since purchase are not covered by warranty.

(2) Replacement not covered by warranty

If your product's fault falls into any of the following categories, the fault will be corrected by replacing the entire product instead of repairing it, or you will be advised to purchase a new product, depending on the severity of the fault.

- Faulty or broken mechanical portions
- Flaws, separation, or rust in coated or plated portions
- Flaws or cracks in plastic portions
- Faults or breakage caused by improper use or unauthorized repair or modification
- Heavily damaged electric circuits due to overvoltage, overcurrent or shorting of power supply
- Cracks in the printed circuit board or burnt-down patterns
- A wide range of faults that make replacement less expensive than repair
- Faults that are not locatable or identifiable

(3) Expiration of the repair period

- We accept requests for repairs up to three years after production of a given model has ceased.
- Repairing a given model may become impossible even during the above period if the parts required for the repairs are no longer available.

(4) Carriage fees for sending your product to be repaired

Carriage fees for sending your product to us for repair are at your own expense.

6.4 How to request repairs

If your product is found faulty, fill in a Repair Request Sheet downloadable from the following URL and email the sheet and send the product to your local distributor.

http://www.renesas.com/repair

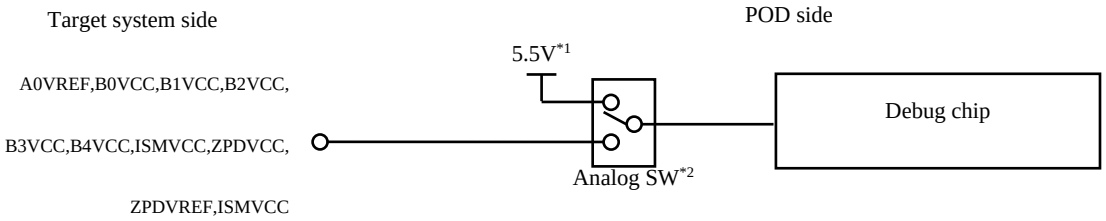
⚠ CAUTION

Note on Transporting the Product:

When sending your product for repair, use the packing box and cushioning material supplied with the MCU unit when it was delivered to you and specify caution in handling (handling as precision equipment). If packing of your product is not complete, it may be damaged during transportation. When you pack your product in a bag, make sure to use the conductive plastic bag supplied with the MCU unit (usually a blue bag). If you use a different bag, it may lead to further trouble with your product due to static electricity.

Appendix A Characteristics of Target Interface

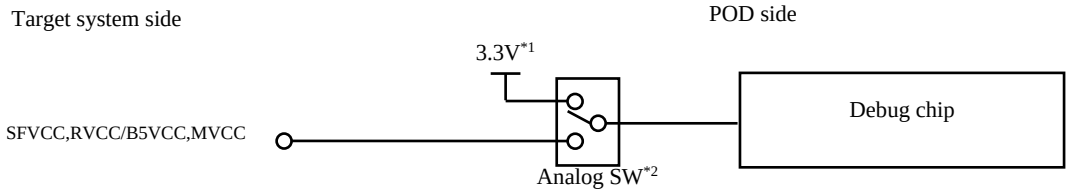
The target interface (interface carrying the signals which connect the in-circuit emulator and target system) behaves as if the actual device is connected in terms of functionality. However, in terms of characteristics, the behavior of the target interface sometimes differs from the behavior when the actual device is connected. The various equivalent circuits below apply to the target interface of this product.



*1:Normal internal power for emulation

*2:Internal power for emulation that is supplied when the target system is not connected.

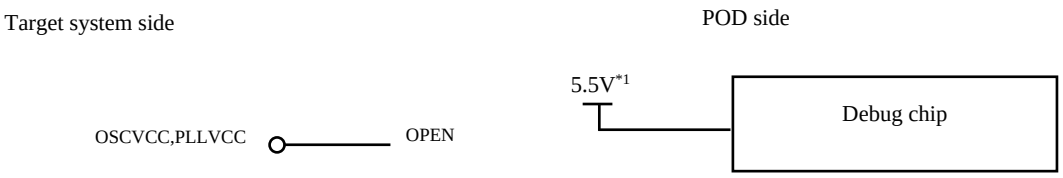
Figure A- 1 Equivalent Circuit



*1:Normal internal power for emulation

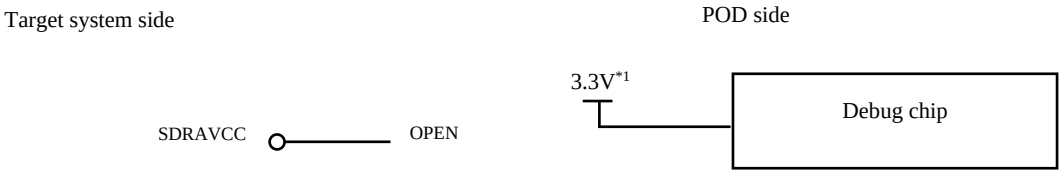
*2:Internal power for emulation that is supplied when the target system is not connected.

Figure A- 2 Equivalent Circuit



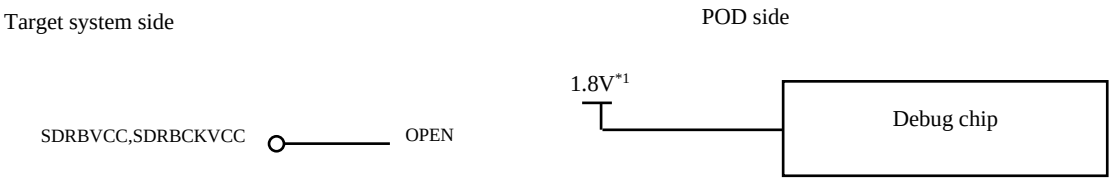
*1:Normal internal power for emulation

Figure A- 3 Equivalent Circuit



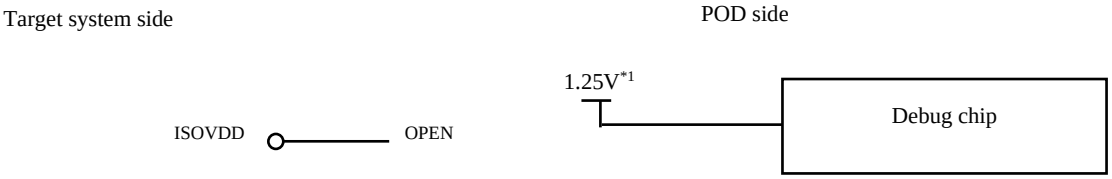
*1:Normal internal power for emulation

Figure A- 4 Equivalent Circuit



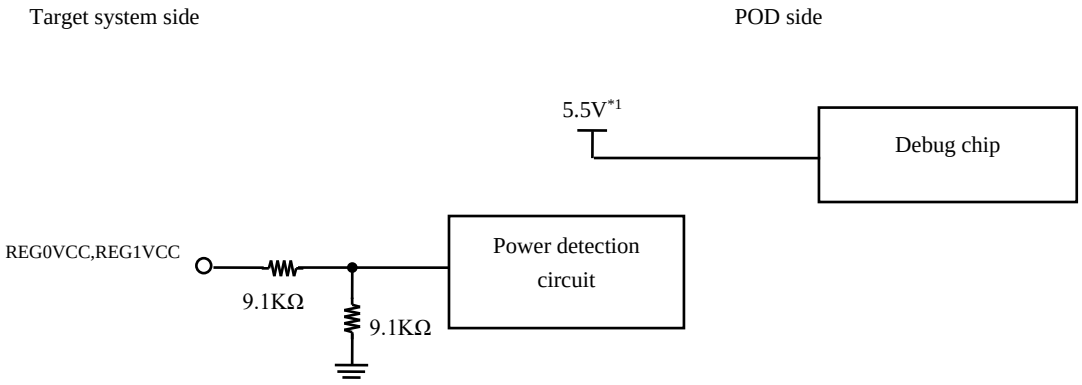
*1:Normal internal power for emulation

Figure A- 5 Equivalent Circuit



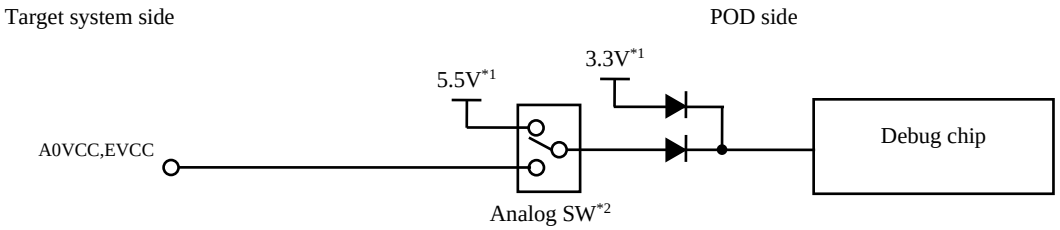
*1:Normal internal power for emulation

Figure A- 6 Equivalent Circuit



*1:Normal internal power for emulation

Figure A- 7 Equivalent Circuit



*1:Normal internal power for emulation

*2:Internal power for emulation that is supplied when the target system is not connected.

Figure A- 8 Equivalent Circuit

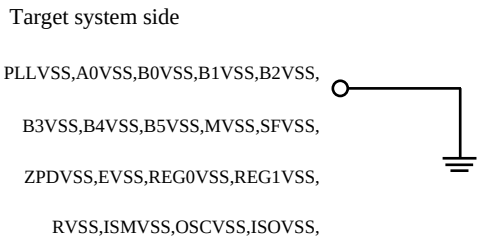


Figure A- 9 Equivalent Circuit

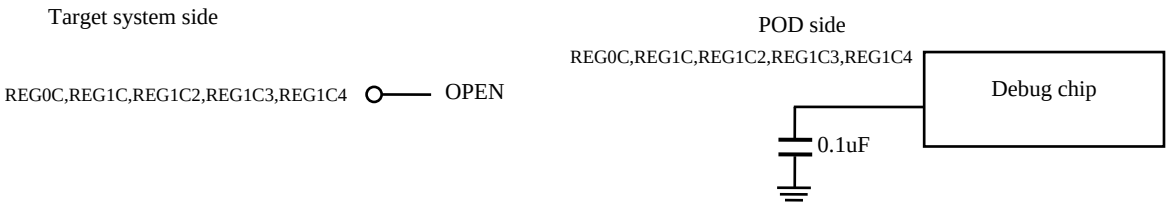


Figure A- 10 Equivalent Circuit

Figure A- 11 Equivalent Circuit

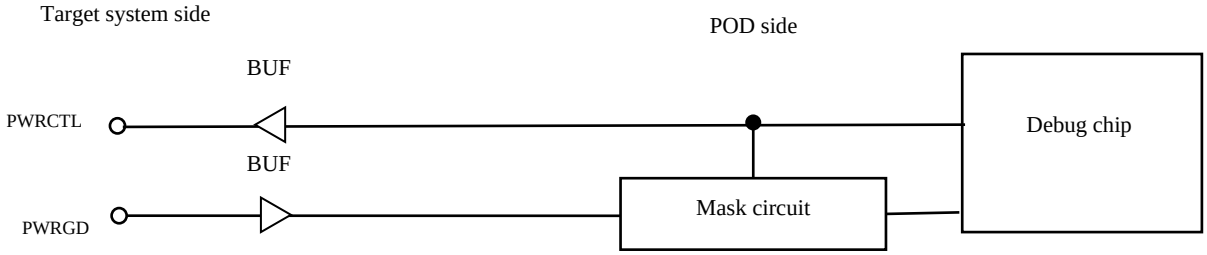
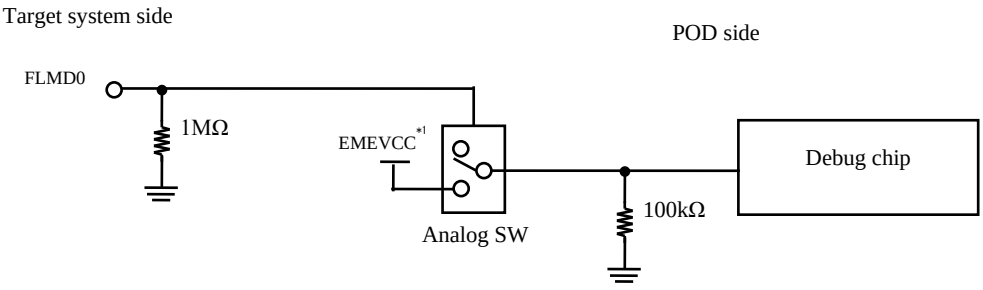
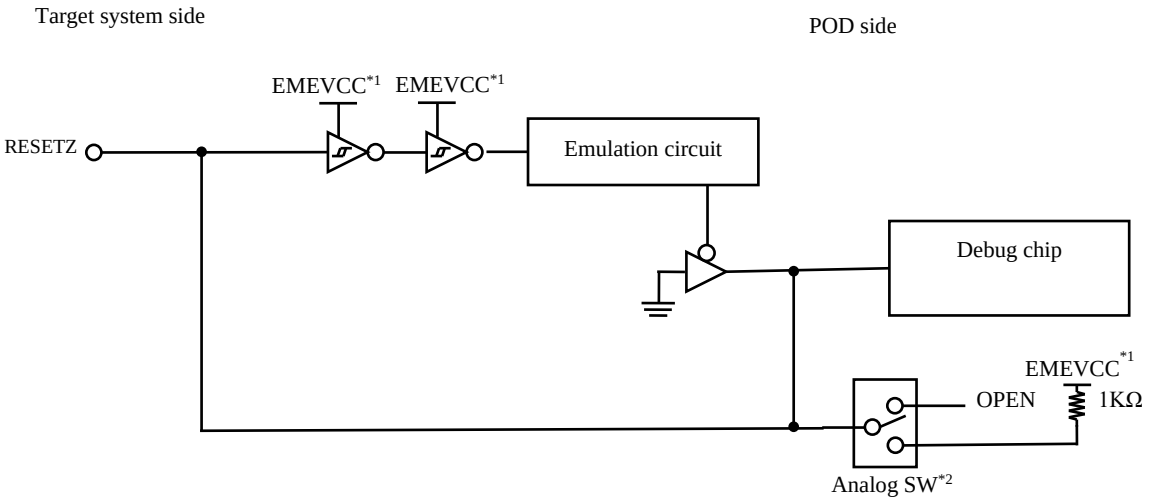


Figure A- 12 Equivalent Circuit



1:Normal internal power for emulation

Figure A- 13 Equivalent Circuit



*1:Normal internal power for emulation

*2:Open-circuit when a target system is connected and pulled up when a target system is not connected.

Figure A- 14 Equivalent Circuit

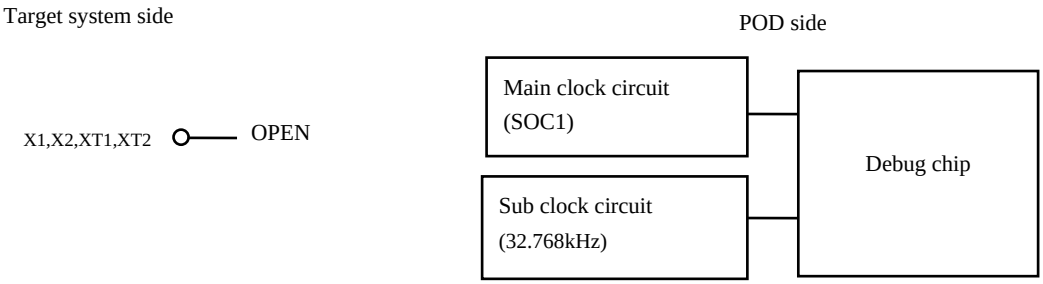
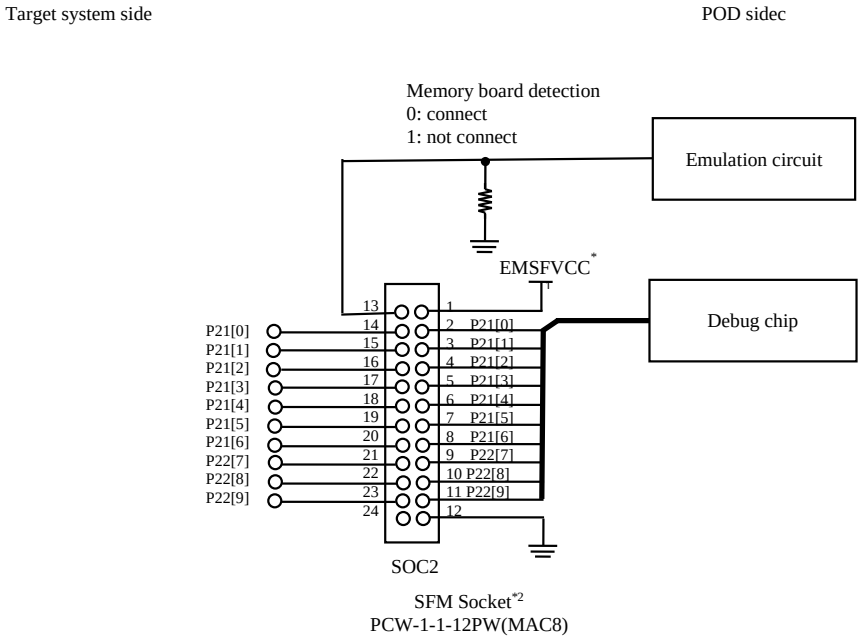


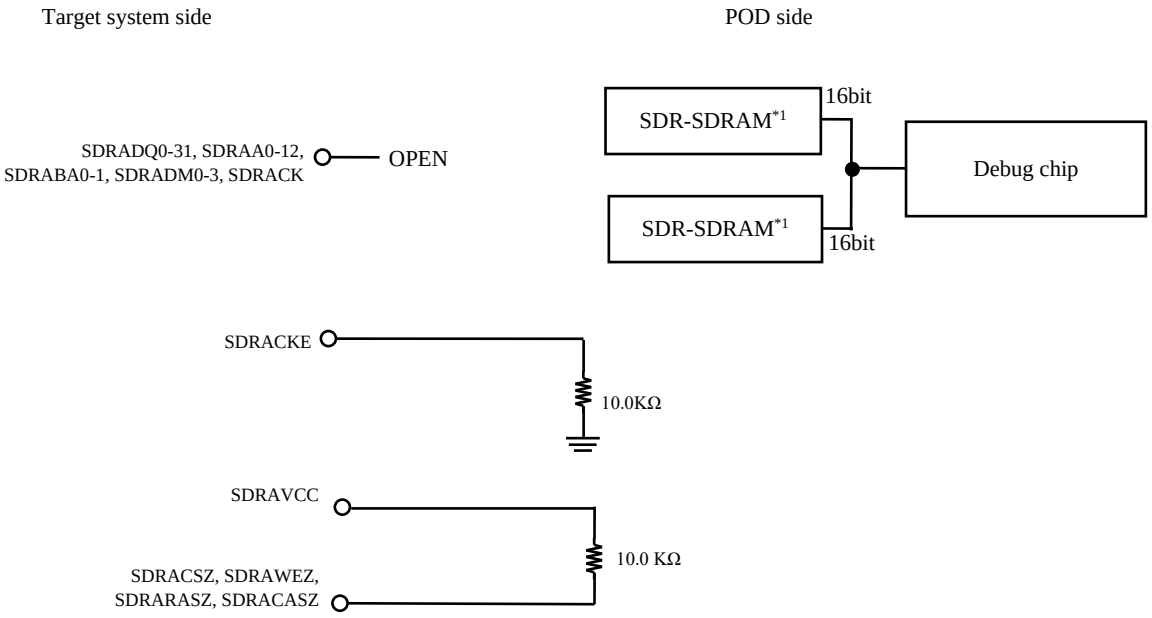
Figure A- 15 Equivalent Circuit



*1:Normal internal power for emulation

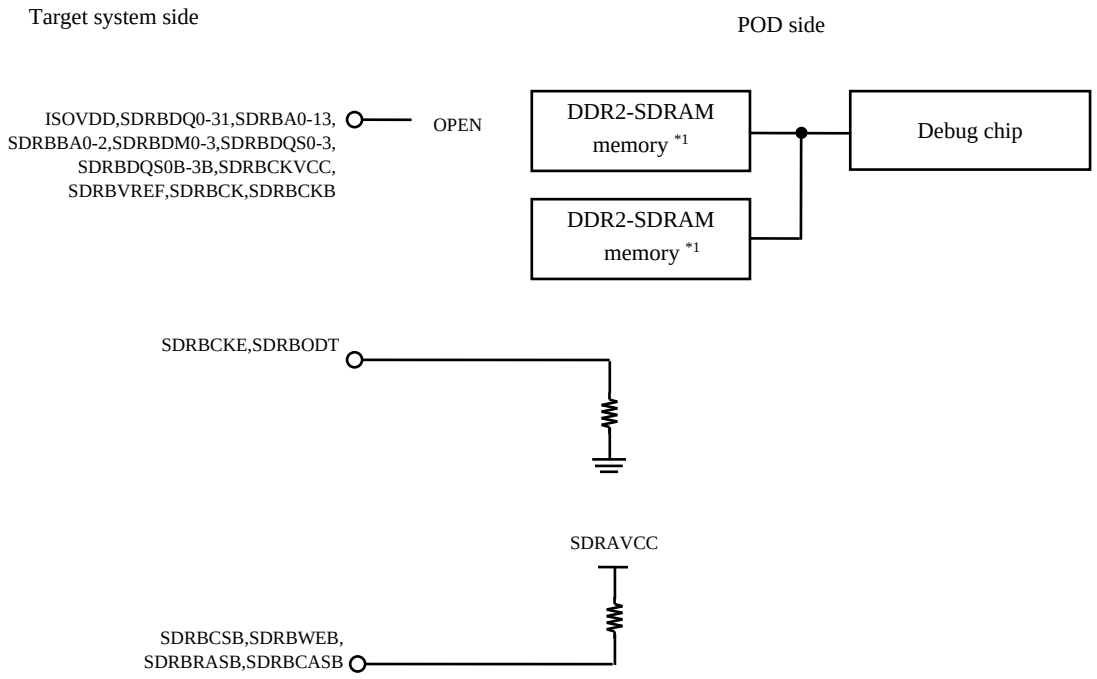
*2:Short-circuited by shunts in the product as shipped.

Figure A- 16 Equivalent Circuit



1:IS42S16320x-6BL(*:version D or later) (Integrated Silicon Solution Inc.)

Figure A- 17 Equivalent Circuit



*1: MT47H128M16RT-25E (Micron Technology)

Figure A- 18 Equivalent Circuit

Revision History	Pod for IE850 In-circuit Emulator RTE7701460EPA00000R User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Dec 01, 2016	—	Preliminary
1.10	Feb 01, 2017	23	Emulator detection by user programs was added.
		44	Descriptions were added to 4.1.27.
		50	4.2.18 was newly added.
		52	4.2.29 was newly added.
		63	Figure A-17、A-18 were added.
1.20	Jul 01, 2017	21	Table 1-2 was modified.
		28	1.4 was modified.
		41-50	4.1.6, 4.1.10, 4.2.5 were skipped.
			Descriptions were added to 4.1.8
			4.1.9 was modified.
			The title of 4.2.1 was changed.
			The title of 4.2.9 ,4.2.10,4.2.11 were changed and the descriptions were added.

Pod for IE850 In-circuit Emulator
RTE7701460EPA00000R
User's Manual

Publication Date: Dec 01, 2016 Rev. 1.00
Feb 01, 2017 Rev. 1.10
Jul 01, 2017 Rev.1.20

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2017 Renesas Electronics Corporation. All rights reserved.

Colophon 4.1

Pod for IE850 In-circuit Emulator

RTE7701460EPA00000R

User's Manual

Renesas Electronics Corporation

R20UT3839EJ0120