User's Manual

32

QB-70F3529-PD

User's Manual: Hardware

QB-V850E2 Emulation Pod
for V850E2/DK4

uPD70F3529

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 Technology Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

R20UT0991ED0100, Rev. 1.00
January 26, 2012

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

   The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

“Standard”:	Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
“High Quality”:	Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
“Specific”:	Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power, voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

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

2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.

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

3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.

   The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

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

5. Differences between Products
   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Regional Information

Some information contained in this document may vary from country to country. Before using any Renesas Electronics product in your application, please contact the Renesas Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

Visit

http://www.renesas.com

to get in contact with your regional representatives and distributors.

Preface

Readers This manual is intended for users who want to understand the functions of the concerned development tool.

Purpose This manual presents the hardware manual for the concerned development tool.

Module instances These microcontrollers may contain several instances of a dedicated module. In general the different instances of such modules are identified by the index “n”, where “n” counts from 0 to the number of instances minus one.

Legend Symbols and notation are used as follows:

- Weight in data notation: Left is high order column, right is low order column
- Active low notation: xxx (pin or signal name is over-scored) or /xxx (slash before signal name) or _xxx
- Memory map address: High order at high stage and low order at low stage

Note Additional remark or tip

Caution Item deserving extra attention

Numeric notation	Decimal:	XXXX
	Hexadecimal	0xXXXX
Numeric prefixes representing powers of 2 (address space, memory capacity):	K (kilo):	$2^{10} = 1024$
	M (mega):	$2^{20} = 1024^2 = 1,048,576$
	G (giga):	$2^{30} = 1024^3 = 1,073,741,824$

Register contents X, x = don’t care

Diagrams Block diagrams do not necessarily show the exact wiring in hardware but the functional structure. Timing diagrams are for functional explanation purposes only, without any relevance to the real hardware implementation.

How to Use This Manual

Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the pod. It is intended for users designing application systems incorporating the pod. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual. The manual comprises an overview of the product; descriptions of the pod, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Related Documents

The following documents are related to this pod or the devices subject of emulation.

The related documents listed below may include preliminary versions. However, preliminary versions are not marked as such.

Caution: The related documents listed are subject to change without notice.

Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Title	Document No.
User's Manual QB-70F3529-PD (V850E2/DK4)	R20UT0991ED0100 (this manual)
QB-V850E2 Preliminary User's Manual	ZUD-CD-10-0092_E
Operating Precautions QB-70F3529-PD	R01TU0029ED0100
User's Manual V850E2/DK4-H	R01UH0077ED0100

Terminology

Abbreviation	Full Form
(Target) device	The device to be emulated
Target (system)	The application system provided by the user. This includes the target program and the hardware provided by the user.
IECUBE®2	Generic name for Renesas Electronics high-performance, compact in-circuit emulator (QB-V850E2)
Pod	IECUBE2 peripheral to interface with the target system
Debug chip	The device used for emulation that is assembled on the pod. It is similar or identical to the target device.
Hi-Z	High Impedance
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Not connected
PLL	Phase Locked Loop
SFR	Special Function Register

Trademarks IECUBE is a registered trademark of Renesas Electronics.

Green Hills, the Green Hills logo, and MULTI are registered trademarks of Green Hills Software, Inc.

Windows is either registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

All other company, product, or service names mentioned in this documentation may be trademarks or service marks of their respective owners.

All trademarks and registered trademarks are the property of their respective owners.

Warning & Caution

This equipment complies with the EMC protection requirements

WARNING

This is a 'Class A' (EN 55022: 2006 + A1:2007) equipment. 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.

EEDT-ST-001-20

CAUTION

This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment. All test and measurement tool including the workbench must be grounded. The user/operator must be grounded using the wrist strap. The connectors and/or device pins should not be touched with bare hands.

EEDT-ST-004-10

For customers in the European Union only

Redemption of Waste Electrical and Electronic Equipment (WEEE) in accordance with legal regulations 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. Renesas Electronics Europe GmbH offers to take back the equipment. All you need to do is register at

http://www.renesas.eu/weee.

Notes for CMOS Devices

1. Voltage application waveform at input pin

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

2. Handling of unused pins

   Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

3. Precaution against ESD

   A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for printed wire boards with mounted semiconductor devices.

4. Status before initialisation

   Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned on, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

5. Power on/off sequence

   In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

6. Input of signals during power off state

   Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

Guarantee & Safety Precautions

1. Circumstances not covered by product guarantee

    - If the product was disassembled, altered, or repaired by the customer.
    - If it was dropped, broken, or given another strong shock.
    - Use at overvoltage, use outside guaranteed temperature range, storing outside guaranteed temperature range.
    - If power was turned on while the AC adapter, interface cable, or connection to the target system was in an unsatisfactory state.
    - If the cable of the AC adapter, the interface cable, the target cable, or the like was bent or pulled excessively.
    - If an AC adapter other than the supplied product was used.
    - If the product got wet.
    - If the product and target system were connected while a potential difference existed between the GND of the product and the GND of the target system.
    - If a connector or cable was connected or disconnected while power was being supplied to the product.
    - If an excessive load was applied to a connector or cable.
    - If the product is used or stored in an environment where an electrostatic or electrical noise is likely to occur.

2. Safety precautions

    - If used for a long time, the product may become hot (50 to 60℃). Be careful of high temperature burns and other dangers due to the product becoming hot.
    - Be careful of electrical shock. There is a danger of electrical shock if the product is used as described above in “Circumstances not covered by product guarantee”.
    - The AC adapter supplied with the product is exclusively for this product, so do not use it with other products.

Table of Contents

Chapter 1 Overview

Figure 1-1 A pod for QB-V850E2

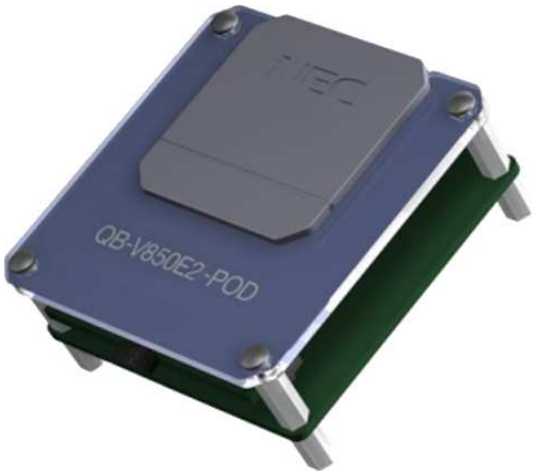
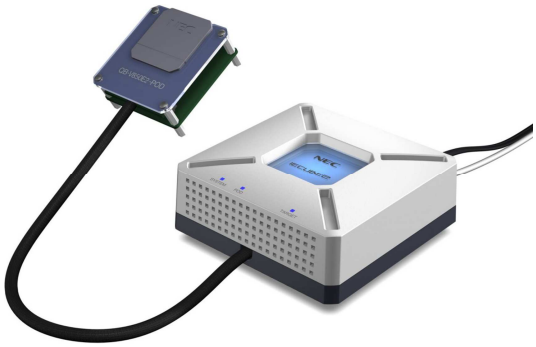


Figure 1-2 A pod assembled with QB-V850E2



1.1 Function

The QB-70F3529-PD is a pod that in combination with a QB-V850E2 becomes an in-circuit emulator for emulating the target device shown below.

Note that the product cannot be used stand-alone, but must be connected to a QB-V850E2.

Hardware and software can be debugged efficiently in the development of systems in which the target device is used.

This manual describes basic setup procedures, hardware specifications, system specifications and emulation features.

To understand the complete emulator system it is recommended to also consult the User's Manual of the QB-V850E2.

1.2 Device Subject of Emulation

The following device may be emulated with the pod.

Table 1-1 Target device subject of emulation

Device
uPD70F3529

1.3 Features

The features of the pod are almost identical to those of the device subject of emulation. For any features such as

- maximum operating frequency
- memory sizes
- peripheral components
- voltage range
- package

please consult the manual or data sheet of the device in question.

This manual only lists the differences to the device.

Table 1-2 Pod hardware specifications

Parameter	Specification
Sub-clock (XT1/XT2)	Fixed at 32.768 kHz

Table 1-3 Pod environmental specifications

Parameter	Specification
Operating temperature range	-0 to 40℃
Storage temperature range	-15 to 60℃
Environmental humidity range	10% to 80% relative humidity, no condensation
Dimensions	57mm x 57mm x 27mm
Power consumption	< 1.5W
Weight	approx. 40g

Table 1-4 Pod debugging features

Parameter		Specification
Emulation memory capacity	Internal code flash Internal data flash Internal RAM Internal BURAM Internal video memory	Same as device
	External memory	Same as device (none)
Program execution functions	Real-time execution function	Available
	Non-real-time execution function	Available (Step execution in source level depends on debugger)
Break functions	Hardware breakpoints	4 available
	Software breakpoints	8 available (possibly reduced by breakpoints required by the debugger internally; usually 1)
	Other	Forced break Trace full break Tier overflow break
Event functions	Detection of execution	Pre-execution: 4 points (only for break function) Post-execution: 8 points
	Detection of access	6 points
	Pass counter	12 bits
	Sequential	4 steps
	Modification while running	Available
Trace functions	Trace data types	Branch-source PC Branch-destination PC Access data Access address Access R/W status Time stamp DMA access data DMA access address DMA R/W status DMA transfer count DMA channel number
	Trace events	Delay trigger Section Qualify
	Other	Trace full stop
Other functions	Masks	Mask external reset Mask internal/peripheral resets Peripheral break function

1.4 Block Overview

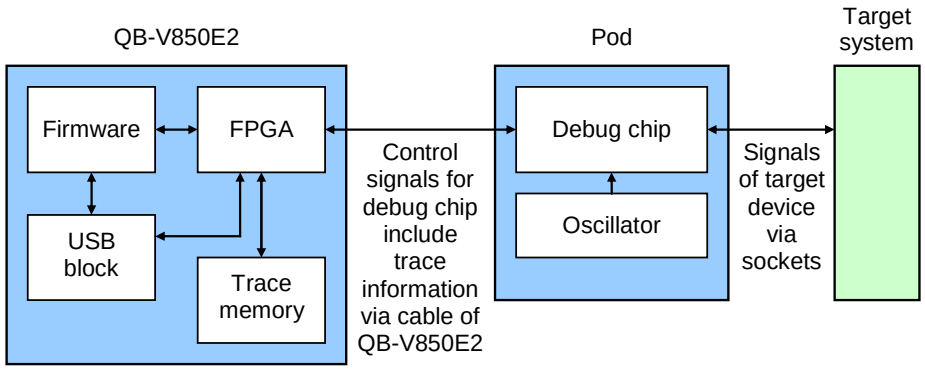


Figure 1-3 Block diagram of IECUBE2 system

1.5 Functional Overview

IECUBE2 is provided with a wealth of debug 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.

1.5.1 Program execution function (real-time 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.

1.5.2 Step execution function (non-real-time execution function)

The step execution function can be used to execute instructions one by one, in assemble instruction units. Only instructions to be executed purely in steps can be executed, because interrupts are not acknowledged during step execution.

Caution: Step execution to be performed at the C language level is performed by a debugger using the break function. In this case, interrupts are acknowledged in step execution. Consequently, if processing at the interrupt destination cannot be completed, step execution may not be completed. For handling such a case, see the manual of the debugger.

1.5.3 Break functions (program execution stop)

The break functions are used to stop program execution. With IECUBE2, program execution can be stopped under the following various conditions. See (1) to (5) for an overview of each break function.

- An address has been executed -> Hardware or software break function
- A variable has been accessed -> Hardware break function
- A specific time has elapsed -> Timer overflow break function

Variable values can be checked during a break and a program can be executed again by changing register values, because the CPU operates even during a break (while the program is stopped). Interrupts generated during the break are suspended, because basically peripheral functions also operate during the break. Use the peripheral break function to stop peripheral functions during the break.

(1) Hardware break function

The hardware break function is used to observe the CPU bus cycles and set a break for a specific fetch or access operation. For example, a break can be set by detecting a state where an address has been executed or a variable has been accessed. For states that can be set, see “Event function”.

Caution: The address for which a break has been set is at a position ahead of the address where an actual access has occurred, because the break set for the access (write, read) is detected at an MEM stage or a WB stage on the CPU pipeline.

(2) Software break function

The software break function is used to set a break when a specific address has been executed (fetched).

(3) Timer overflow break function

This function is used to set a break when a time set by using the time measurement function has elapsed. For example, if the execution time of a function must be 2ms, a break can be set when at least 2ms have elapsed between starting and ending the function. This function and the trace function can be used together to find the source that has taken time.

(4) Forced break function

This function is used to forcibly stop a program when it is desired to be stopped.

(5) Trace full break function

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

1.5.4 Trace function (program execution history)

The trace function can be used to check the CPU execution history (trace). Items (1) to (7) can be recorded in the execution history.

(1) Program counter (PC) of branch source and branch destination

The PCs of a branch source and a branch destination can be recorded in the history.

Consequently, practically all executed programs can be checked, because programs executed between branch points also will be clarified. The amount of trace memory used can be saved and more history items can be traced by that amount, by recording only branch information. (The amount of instructions that can be traced back depends on the number of branches.)

(2) Access data/access address

Access addresses for memories and peripheral I/O registers, and access data can be recorded in the history. Read and write operations can also be recorded in the history.

Caution: Accesses to CPU program registers (such as r1 and r2) and system registers (such as PSW and EIPC) cannot be recorded in the history.

(3) Time stamp

The time elapsed from the trace start point can be added to each trace information. The timer performance for time stamps is the same as that of the time measurement function.

(4) DMA access address, data, status, channel number, transfer count

When the DMA function of the target microcontroller is being used, the DMA access can be recorded in the history.

- Access address
- Access data
- Access status (R/W)
- DMA channel number
- Transfer count

(5) History of specific sections (section trace)

Only specific sections can be recorded in the history by using the event function in combination. For example, the execution history of from the start to the end of a function can be recorded.

(6) History of specific phenomenon occurred (qualify trace)

Only the occurrence of specific phenomena can be recorded in the history by using the event function in combination.

For example, a history of having accessed to only a variable can be recorded.

(7) Recording histories before and after specific phenomenon has occurred (delay trigger trace)

The history after a specific phenomenon has occurred can be recorded by using the event function in combination. This is similar to being able to observe a signal waveform by assuming an edge as a trigger, when using an oscilloscope to observe a signal.

For example, the program execution histories before and after a write access has been performed for a variable can be viewed.

1.5.5 Time measurement function

This function is used to measure the execution time of a specific section. The measurement start and end points can be set by using the event function.

In addition, the maximum, minimum, and average execution time and the number by which the measurement section has been passed can be measured.

1.5.6 Event function (specific CPU operation detection)

The event function is used to detect specific fetch and access operations by observing the CPU bus cycle. CPU operations, such as of an address being executed and a variable being accessed can be detected. Such specific CPU operations are called events. Use the event function together with the following functions.

- Hardware break function
- Trace function
- Time measurement function

The events that can be registered by using the event function are as follows.

(1) Pre-execution event

A pre-execution event is detected when execution of an address is attempted. It can be used only with the hardware break function. Four pre-execution event points can be specified.

Detection conditions that can be specified:

- Execution address

(2) Post-execution event

A post-execution event is detected when an address has been executed. The address of a post-execution event can be specified as a range. Up to eight post-execution event points can be specified, but if the execution address has been specified as a range, two points will be consumed. When the execution address has been specified as a range for all events, four event points can be specified.

Detection conditions that can be specified:

- Execution address (can be specified as a range)

(3) Access event

An access event is detected when an address has been accessed (read or written). The following detection conditions can be specified for an access event.

Up to six access event points can be specified, but if the access address has been specified as a range, two points will be consumed. When the access address has been specified as a range for all events, three event points can be specified.

Detection conditions that can be specified:

- Access address (can be specified as a range)
- Access data
- Access size
- Access status (read, write, both read and write)

1.5.7 Event link function (event combinations)

The event link function is used to combine into one event, events that have been registered by using the event function. It is used to detect a specific sequence, such as when an address has been executed after a variable was accessed.

Chapter 2 Hardware Setup

2.1 System configuration

The following adapters and connectors are necessary to connect the pod to the target system. These adapters and sockets are sold separately.

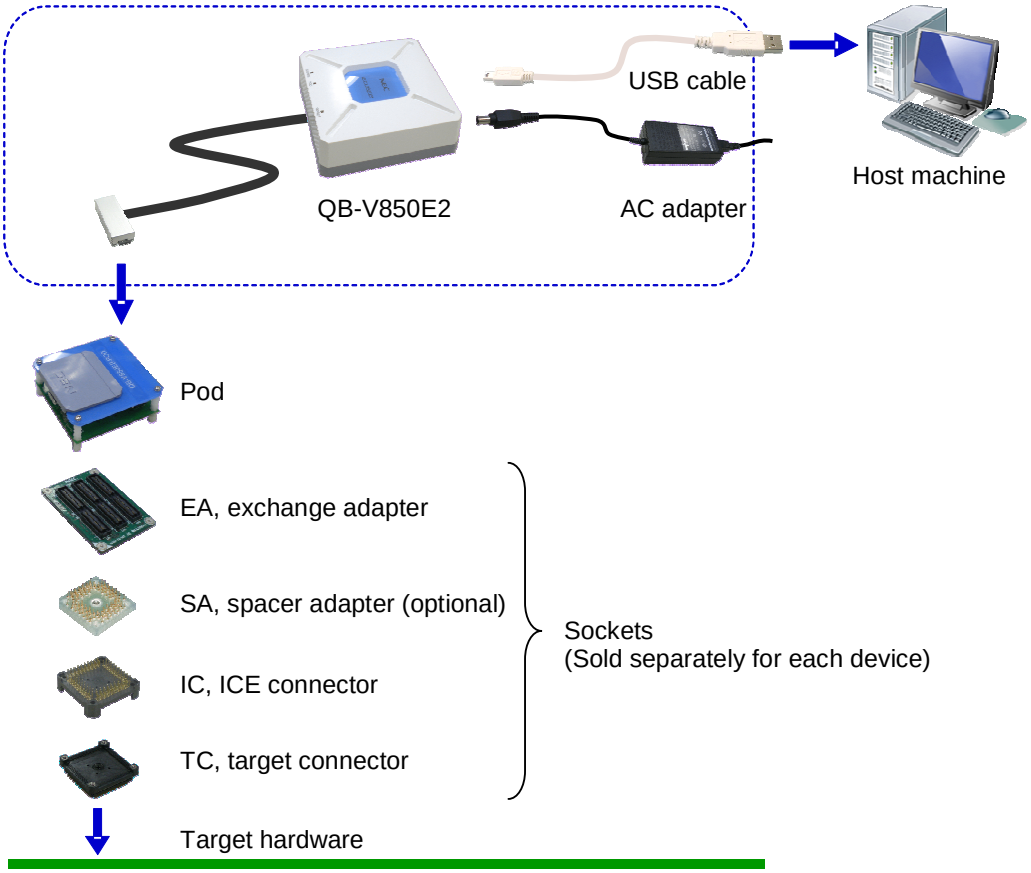


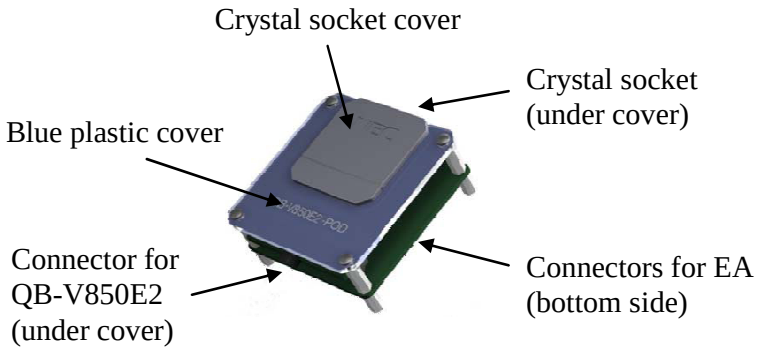
Table 2-1 Adapters and Connectors

Description	Product
IECUBE2 / main ICE	QB-V850E2
Pod	QB-70F3529-PD
EA, exchange adapter	QB-176GM-EA-64T
IC, ICE connector (also called EC, emulation connector)	QB-176GM-YQ-01T (YQPACK176SD*)
TC, target connector	QB-176GM-NQ-01T (NQPACK176SD*)

* Product of TET (Tokyo Eletech Corporation, http://www.tetc.co.jp)

Table 2-2 Optional Adapters

Description	Product
IC/EC+TC combined	(YQPACK176SD-T*)
SA, spacer adapter	QB-176GM-YS-01T (YQSOCKET176SD*)
MA, mount adapter	QB-176GM-HQ-01T (HQPACK176SD*)

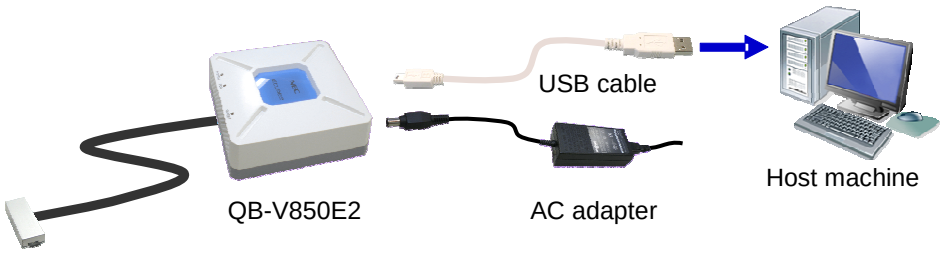


2.2 Connecting to the QB-V850E2

This section describes how to mount the pod to the IECUBE2. Refer to the picture found in 2.1.

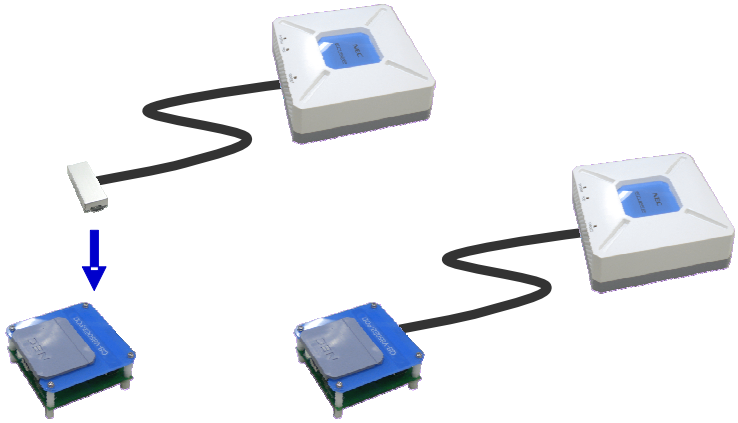
All operations must be performed while power is switches off.

1) Connect the IECUBE2 to a host PC by the USB cable.
Supply it by the AC adapter.
For further details and setup of the IECUBE2, refer to the IECUBE2 User's Manual.



2) To connect the pod to the IECUBE2:

- Remove the blue top cover of the pod by removing the bolts.
- Plug the cable connector of the IECUBE2 into the pod's connector CN1. Observe the orientation (pin 1).
- Replace the top blue cover of the pod.

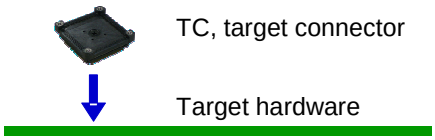


2.3 Connecting to the Target

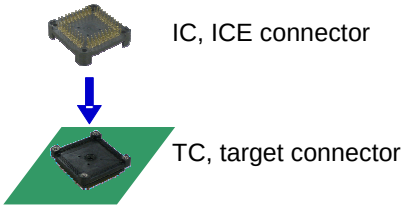
This section describes how to mount and connect components including connectors. Refer to the picture found in 2.1.

All operations must be performed on powered-down systems.

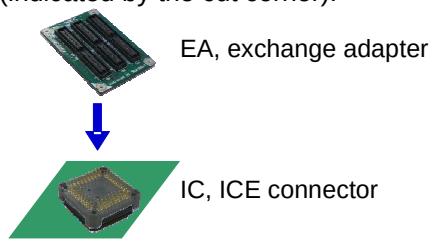
1) Solder the target connector to the target board. For details about how to do so, refer to the document supplied with the connector.



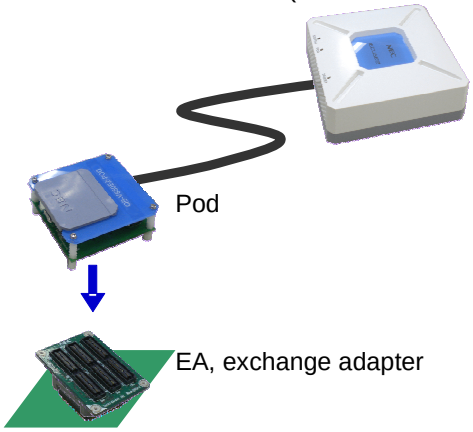
2) Connect the ICE connector to the target connector by placing it on the target connector and fastening the four screws. For details about how to do so, refer to the document supplied with the connector.



3) Connect the exchange adapter to the ICE connector by inserting the pins of the IC into the socket of the EA. Be sure to align the position of pin 1 (indicated by the cut corner).

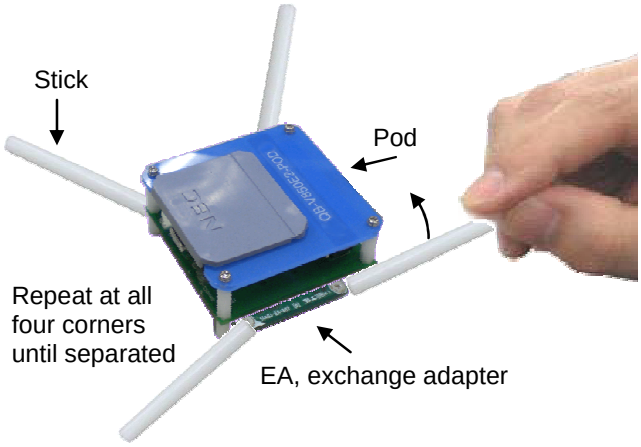


4) Connect the pod to the exchange adapter by mating the three connectors. Observe the orientation (view the middle connector)



To remove the exchange adapter from the pod at a later time, use the plastic lever included in the IECUBE2 package to carefully separate the two units (refer to IECUBE2 User's Manual).

Caution: Do not put any pressure on the small components of the pod. They can be damaged easily.



2.4 Clock Module

The main clock (X1) is generated by the crystal on the emulation pod.

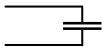
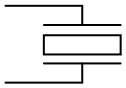
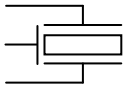
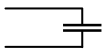
It is mounted on a carrier connector under the grey plastic lid. By default a 4MHz crystal is mounted.

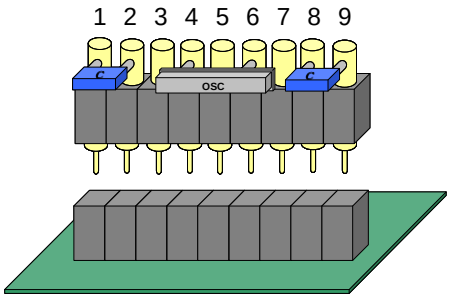
In case a different frequency or crystal is required, it can be replaced by following these procedures:

1) Open the emulation pod cover as shown below:

2) Mount the resonator or crystal and capacitors onto the carrier connector

Table 2-3 Clock module pin-out

Pin	Intended usage	Signal	Intended usage (crystal)	Intended usage (resonator)
1	Capacitor	GND		
2	Capacitor	Xtal (X1)		
3	NC	NC		
4	Crystal/resonator	Xtal (X1)		
5	Resonator	GND		
6	Crystal/resonator	Xtal (X2)		
7	NC	NC		
8	Capacitor	Xtal (X2)		
9	Capacitor	GND		



3) Replace the carrier currently inserted to CN2 of the pod by the custom made one.

4) Close the emulation pod cover.

Note: A clock source on the target hardware is not supported.

The sub-clock (XT1) is generated by a 32768Hz crystal on the emulation pod and cannot be modified.

2.5 Break LED

A blue LED on the pod, D2 (the only LED), will light up when the tool is in break mode.

During normal run operation the LED will be dark.

Chapter 3 Emulation Features

3.1 Emulation Registers

The device contains some registers that may be used when running on the emulator.

3.1.1 Clock Monitor (CLMA0/1)

The clock monitor has the ability to emulate a clock failure for testing purposes by writing to the emulation register CLMAnEMU0.

Please refer to the device User's Manual, chapter 8.7.5 "Clock Monitor registers", section (7) "CLMAnEMU0 – CLMAn emulation register 0" for more details.

Please note that this register can only be written to during break mode.

3.2 Peripheral Break

Some peripherals may optionally be configured to stop operation during break mode.

Please refer to the device User's Manual, chapter 34.1.1 "Modules behaviour during emulation break" for more details.

This feature must be globally enabled by writing to the register EPC (Emulation peripheral break control), described in the device User's Manual, chapter 34.3 "Emulation Break Control".

This feature can be enabled for the individual peripherals by registers inside the peripherals.

For details, please refer to the device User's Manual and the chapters for the individual peripherals. A global overview may be found in chapter 34.1.1 "Modules behaviour during emulation break", in table 34-2 "Modules with optional emulation break"

Table 3-1 Modules with optional emulation break

Module	Module
TAUBn	Timer Array Unit B
TAUJ0	Timer Array Unit J
RTCA0	Real-Time Clock
CSIGn	Clocked Serial Interface G
URTEn	Asynchronous Serial Interface E
IICBn	I^2C Interface
IISA0	I^2S Interface
PCMP0	PCM-PWM
ADCA0	A/D Converter A
ISM0	Intelligent Stepper Motor Driver
SG0	Sound Generator
OSTM0	OS Timer

3.3 Signal Masking

It is possible to mask some vital signals for testing purposes. When the masks are set, the signals have no effect.

The signals in question are

- RESET (the external RESETZ pin)
- TRESET (any reset, including resets from peripherals)

For more details, please refer to the eserv documentation and the device User's Manual, chapter 34.1.2 "Signal masking".

Chapter 4 Operation

4.1 Power-up Sequence

To start operation, please power-up the system in the given order.

Do not turn on the target hardware before turning on the IECUBE2.

1) Turn on the IECUBE2. (The target system should not be powered yet.)
   The SYSTEM and POD LEDs of the IECUBE2 must light up steadily.
2) Turn on the target system hardware.
   The TARGET LED of the IECUBE2 must light up steadily.
3) Start the debugger software.

4.2 Power-down Sequence

Do not turn off the IECUBE2 before turning off the target hardware.

1) Exit the debugger software.
2) Turn off the target system hardware.
3) Turn off the IECUBE2.

4.3 Connect String

When using the Green Hills MULTI Debugger, you need to enter a connection string to connect to the emulator.

```
connect 850eserv2 -iecube -e2 -df=DF3529.800
                  –id FFFFFFFFFFFFFFFFFFFFFFFF –tc
                  -nouse_extbrk
```

`connect`	instructs the debugger to connect to a target hardware
`850eserv2`	is the target server to be used
`-iecube`	is the emulator to be used
`-e2`	is the type of CPU core (V850E2)
`-df=DF3529.800`	instructs the debugger to use the devicefile DF3529.800 You may want to add the `–ip` option to provide a path. See the eserv documentation for details
`-id [ID]`	provides the security ID of the device that is to be connected to. Default is FFFFFFFFFFFFFFFFFFFFFFFF (96 bits)
`-tc` (optional)	indicates that a target hardware is connected. For stand-alone operation omit this option.
`-nouse_extbrk` (optional)	allows the usage of all available breakpoints as software breakpoints.

For more details, please refer to the eserv documentation.

Chapter 5 Differences between Emulator and Device

5.1 Additional Information on Emulator

This section describes items that the user should be aware of, specific to the emulator.

5.1.1 Real-chip based emulator

The emulator uses a similar device as the target device to emulate most functions.

Any limitations that apply to the target device also apply to the emulator, if not stated otherwise.

5.1.2 Reset by debugger

A reset issued by the debugger (e.g. by the command string “target rst”) will behave the same way as power-cycling the device. I.e. the status SFR RESF will be cleared.

5.1.3 Reset during break

(1) External reset

If a reset is applied to the target pin RESETZ during break mode, it will be handled as soon as run mode is resumed.

(2) Reset by LVI

If a reset is caused by LVI (when REG0VDD drops below the set level) during break mode, no reset occurs, even when run mode is resumed, unless.REG0VDD remains below the set level.

(3) Reset by CLMAn

If CLMAn is enabled and a clock failure occurs during break mode, the reset will be handled as soon as run mode is resumed.

(4) Reset by SWRES

The protected SFR SWRES cannot be written during break mode. Therefore a reset by SWRES cannot occur during break mode.

(5) Reset by WDTAn

The watchdog timers (WDT) are stopped and locked during break mode. Therefore a reset by WDTAn cannot occur during break mode.

5.1.4 Target detection voltage

The target is considered to be present and powered when the REG0VDD supply is above 2.4V. Below that voltage the TARGET LED of QB-V850E2 will be dark..

5.1.5 Self-Programming of Data Flash

Self-programming of data flash is fully supported.

Additional convenient features are available from the eserv shell. For more details, please refer to the eserv documentation.

5.1.6 Flash content during repair and update

For security reasons and to verify flash functionality, the content of both code flash and data flash will be erased when the tool is returned for a repair or for an update. Be sure to save any required data prior to returning the tool.

5.1.7 All target pins

All target pins are connected via sockets, connectors and PCBs.

Due to the increased wire length, connector impedance and capacitance, the electrical characteristics of the pins may differ slightly between the emulator and the device.

5.1.8 ADC pins

As all target pins, the pins relevant for the ADC are connected via sockets, connectors and PCBs.

Conversion results of the ADC may differ slightly due to impedance or noise encountered on this connection.

5.1.9 Decoupling capacitors

The decoupling capacitors for the debug chip are located on the pod. This is necessary, as the distance between debug chip and capacitors must be short.

This introduces additional capacitance on the power lines.

Also, the capacitors on the target circuit are of no function when using the emulator.

5.1.10 Watchdog timer (WDTA) during break

During break mode the WDTA is always stopped, even if peripheral break is disabled.

5.1.11 Hardware breakpoint address mask for iRAM

When setting a hardware breakpoint (BRA/BRS) in the iRAM area, the address bits 31 to 29 are ignored.

E.g. setting a breakpoint at 0xFExxxxxx will also cause a break at the following addresses: 0x1Exxxxxx, 0x3Exxxxxx, 0x5Exxxxxx, 0x7Exxxxxx, 0x9Exxxxxx, 0xBExxxxxx and 0xDExxxxxx.

Equally, setting a breakpoint at any one of the listed addresses will cause a break at all other of the listed addresses.

5.2 Cautions

This section describes items that the user should be aware of.

They are however functionally identical to the device and are only listed for convenience. They do not describe limitations of the emulator.

5.2.1 X1 clock mandatory during download

Download of application code requires a valid clock source present at X1.

If you do not have a clock module inserted in CN2 of the pod, download will fail with the following message:

```
Download failed, error during write 0x(address) -
0x(address)
debug server: download of "(filename)" failed
load: Download failed.
```

```
Memory write error 0xc2d: user system err (PLL lock
failed)
internal flash write error
```

5.2.2 Data flash access only 32-bit

The data flash can only be read in 32-bit units. Reading in 8- or 16-bit units will return 0x00 (0x0000).

When opening a memory view window in the debugger, the memory access size commonly is set to 8-bit and only 0x00 will be seen as data, even of the data flash contains other data.

Be sure to set the access size to 32-bit to be able to view the correct contents of the data flash.

5.2.3 iRAM ECC errors for uninitialized memory

After power-up of the device the contents of the iRAM is undefined. The RAM has ECC (error correction code) bits that are also undefined.

When a memory cell is read before it is initialised by writing to it, very likely an ECC exception will be raised.

Be sure to initialise any memory before reading from it.

Be especially careful when executing code from iRAM, as the prefetch feature may pre-read memory from many bytes above the actual program counter.

If the emulator is not turned off between application runs, the content of the iRAM will be conserved. It is possible that no ECC errors will occur, even if RAM cells have not been initialised in the current application run, but in a previous run.

To ensure to detect an ECC error, power-cycle the emulator to invalidate the iRAM content.

5.2.4 FLMD0 during RESET

Be sure to set the mode pin FLMD0 = L during reset.

If the FLMD0 pin is at a high level during reset release, the device will enter flash writer mode. This mode is not supported on the emulator.

5.2.5 Standby entry sequence debugging

The standby entry sequence may not be interrupted by breakpoints and it may not be executed by single-stepping.

If the sequence is interrupted in any way it may not deliver the expected result. A pre-mature wake-up may occur if the debugger's actions trigger a wake-up condition that does not have its source in the application or target hardware.

Be sure not to interrupt the standby entry sequence in any way.

Also be aware that some debugger functions like memory windows, RAM monitors or terminal I/O may cause breaks without explicit user requests.

5.2.6 Forced standby release

When the emulator is in a standby mode, do not stop execution from the debugger when you have not enabled the debugger as wake-up source.

In that case the debug chip will not wake up from standby mode and the debugger will hang up due to missing response from the debug chip.

If you want to be able to enter break mode from standby mode, be sure to set WUFMSKH0.bit15 (see device documentation for details).

5.2.7 Possible trace data loss at reset

When the debug chip is reset, e.g. by a POC event, the FIFO of the trace data is also cleared. If the FIFO still contained data, that trace data may be lost.

5.2.8 Flash lifetime

As the code flash and data flash are physically represented by flash, the same limitations apply as for the target device.

The number of write cycles and the data retention time are limited.

Within reasonable usage, the lifetime should not be exceeded on the emulator.

In case an error during download occurs due to a defective flash cell, an error message will appear.

```
0x0c2a:user system err (erase failed)
The erasing of flash memory failed.

0x0c2b:user system err (write failed)
The writing of flash memory failed.

0x0c2c:user system err (verify failed) The internal
verification of flash memory failed.
```

5.2.9 Consumption of data during break

Some SFRs react on a read operation, considering the content of the register to be consumed.

Some examples of these SFRs are:

- URTEnRX
- CSIGnRX0
- IICBnDAT
- FCNnCMRGRX
- FCNnCMTGTX

Be sure not to read these registers (e.g. by a watch window) during break mode. If you read the register, the application behaviour may be influenced (e.g. buffer underrun, no indication of buffer overflows, incorrect data).

5.3 Differences

This section describes differences between the emulator and the device.

These differences are due to design.

However, the functionality of the device still can be achieved.

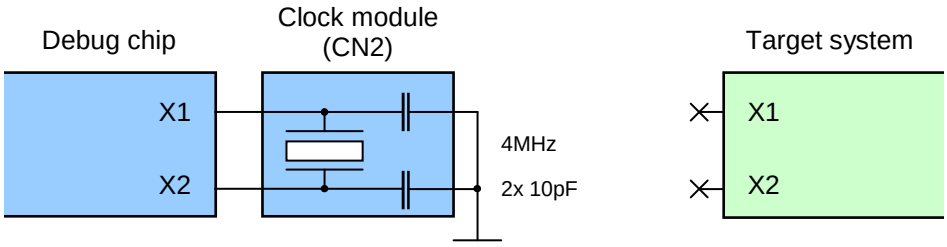
5.3.1 X1/X2 pins

The X1/X2 pins cannot be used from the target hardware.

These pins of the debug chip are connected to the clock module connector (CN2) on the pod.

If you need to change the crystal, please use the clock module.

The target pins are not connected (open).



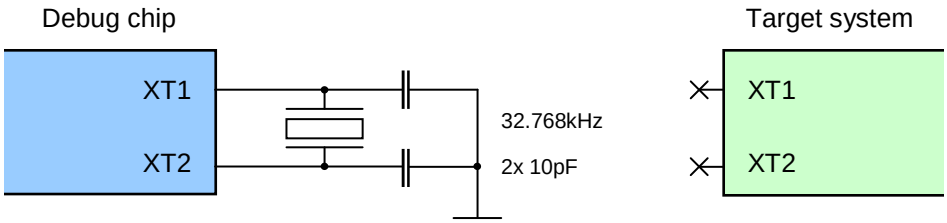
5.3.2 XT1/XT2 pins

The XT1/XT2 pins cannot be used from the target hardware.

These pins of the debug chip are connected to a 32.768 kHz crystal on the pod.

This crystal cannot be changed.

The target pins are not connected (open).



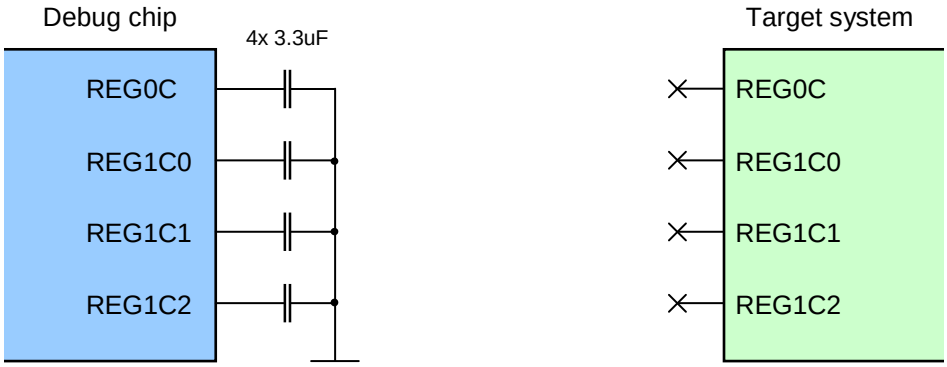
5.3.3 REGCn pins

The REGC0 and REGC1 pins are not used from the target hardware.

These pins of the debug chip are connected to capacitors on the pod.

The target pins are not connected (open).

This difference should have no impact.

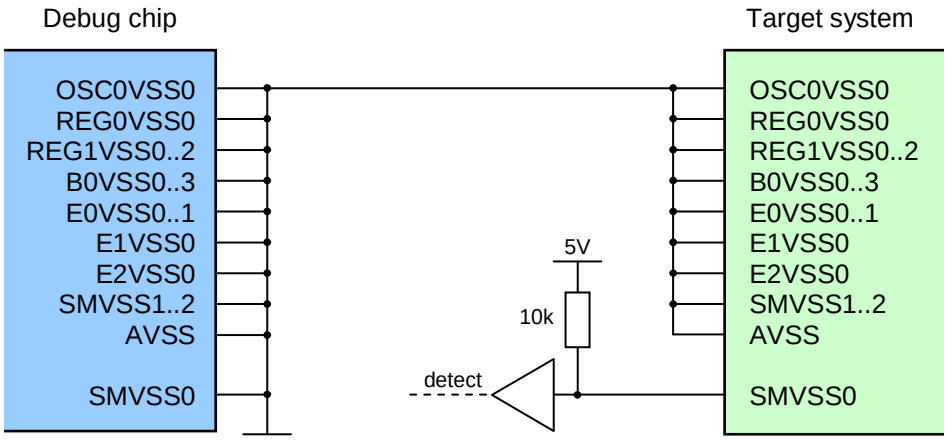


5.3.4 GND pins

All power pins considered to be GND, i.e. any pin names ending in xxxVSS, are connected internally, by low impedance.

This is also true for AVSS, but not for AVREFM.

The only exception is SMVSS0 (pin 107), which is used as a target detection. This pin must be connected to common ground on the target hardware.



5.3.5 Power pins

All power pins considered to be a supply i.e. any pin names ending in xxxVDD, are connected to the debug chip via a power switch.

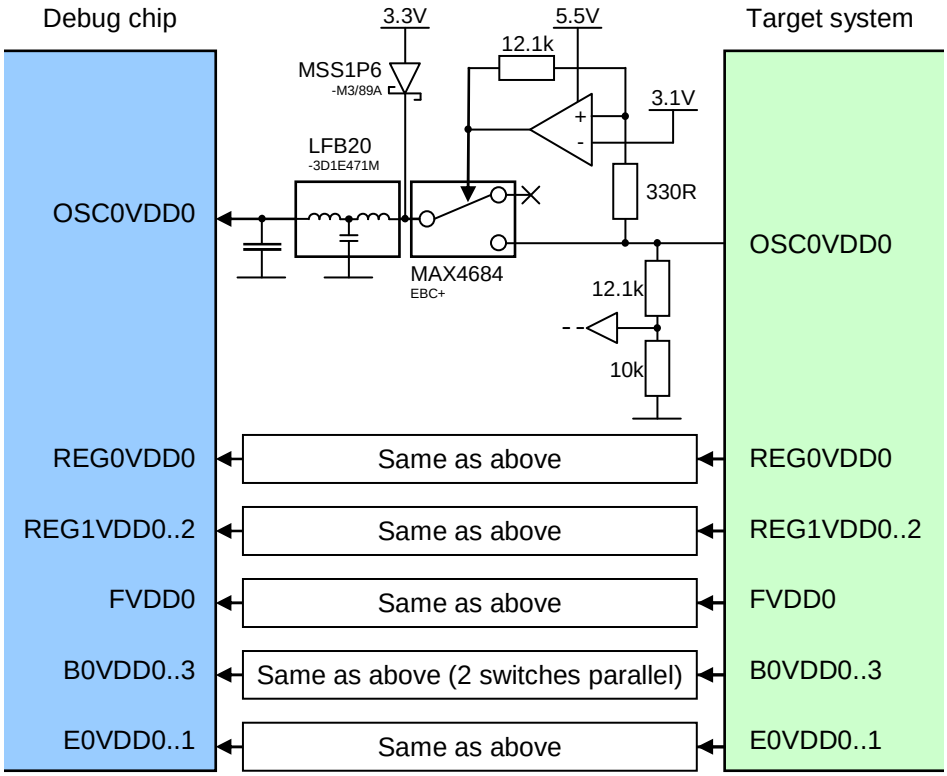
The power domains are switched individually (i.e. B0VDD and E0VDD are separate switches). However the domains are grouped (i.e. B0VDD0 and B1VDD1 are connected internally on the pod).

In stand-alone operation mode the power is switched to internal 3.3V supplies.

In target mode, the target voltages are switched to the debug chip.

This is also true for AVREFP and AVREFM.

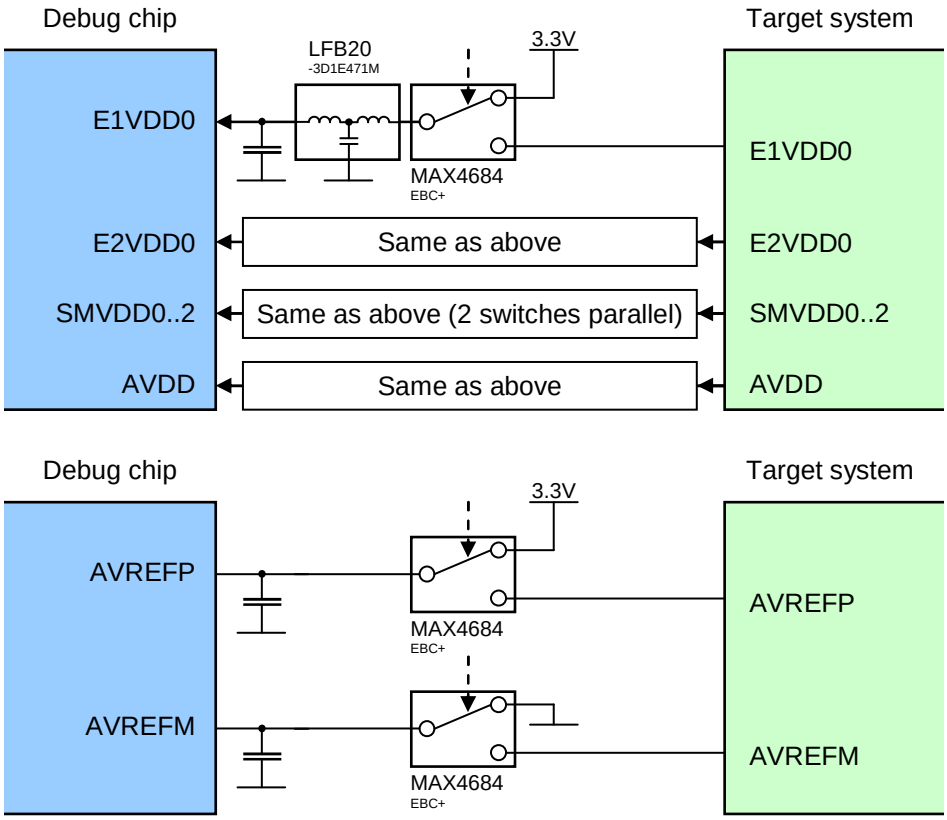
The switches and the EMI filters that are inserted before the debug chip have a small impedance (max. 0.8 Ohms) that has influence on the true supply of the debug chip. However, normally this impedance can be ignored.



The above displayed power domains (OSC0VDD, REG0VDD, REG1VDD, FVDD, B0VDD, E0VDD) will draw less current from the target than the device when the target is powered below approx. 3.0V, as the internal supply will be used.

The above displayed power domain will draw more power than the device when the target is powered above 3.3V and the device enters standby mode, as the debug chip does not physically switch off the power.

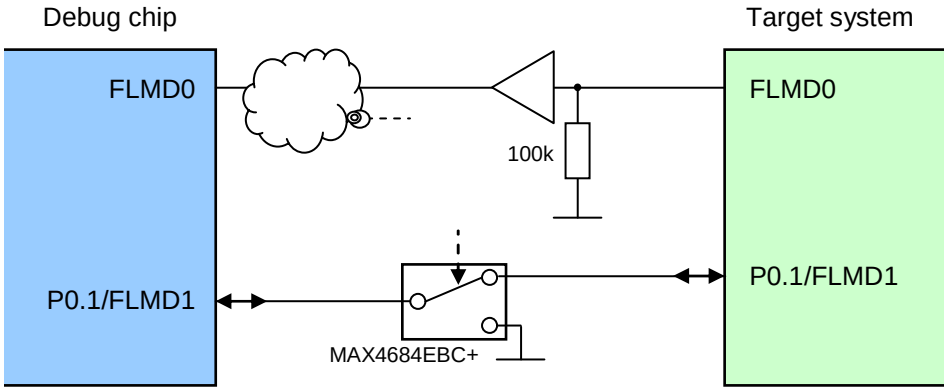
The current of B0VDD and of E0VDD may be slightly greater than that of the device.



5.3.6 FLMD pins

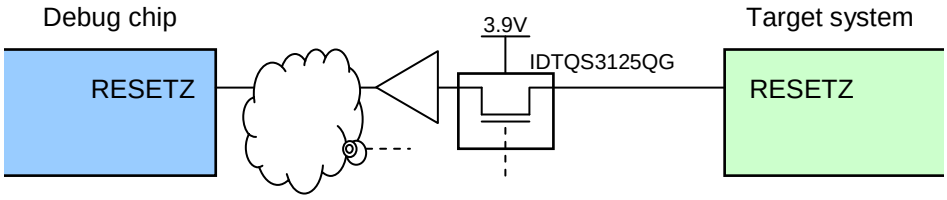
The mode pin FLMD0 is not connected to the debug chip directly. It is used to control some glue logic. Therefore the input characteristics will differ.

The port pin P0.1 with the alternate function FLMD1 is connected to the debug chip through a switch.



5.3.7 RESETZ pin

The RESETZ pin is not connected to the debug chip directly. It is used to control some glue logic. Therefore the input characteristics will differ.



5.3.8 WAKE pin during RESET

The state of the WAKE pin during an external reset (RESETZ pin is pulled low) is not hi-impedance (hi-Z) as for the target device, but at a driven logic high level.

5.3.9 POC (power-on-clear)

The POC level detection is performed discretely via an ADC. The response time is much higher than for the device, where the REG0VDD is monitored by a comparator.

5.3.10 Power during Reset and Deep Stop

During Deep Stop and RESET, the debug chip doesn't physically power off. The functional behaviour however is the same.

However, as the power is not physically removed, the content of the iRAM is maintained. This is a difference to the target device, where the content truly becomes undefined.

5.3.11 Buffer supply voltages below 3.3V

The supply voltages of the buffers of JP0, P0, P27, P28, P40, P41 and WAKE at the debug chip will follow the target voltage linearly in the range of 3.3V to 5.5V.

In the target voltage range of 3.0V to 3.3V, the buffer voltage will be linearly tracked when the target voltage has not dropped below 3.0V before, and remain at 3.0V until the target voltage reaches 3.3V or more again. This is a hysteresis function.

For target voltages below 3.0V, the debug chip will be powered internally with 3.0V.

This will result in a different VOH and/or VIH of the buffers for target voltages below 3.0V.

5.3.12 Buffer supply voltages in standby/power-down mode

The supply voltages of the buffers of JP0, P0, P27, P28, P40, P41 and WAKE and of the isolated area ISO at the debug chip is always powered with a minimum voltage of approx. 3.0V, even if the target voltage drops below that voltage.

As a consequence, turning off the voltages at the target does not power-down the device's region and/or buffers.

Especially the ports are affected. To make the emulator behave equally to the device, be sure to disable the buffers' drivers before powering off a voltage region or entering standby mode.

5.3.13 Flash features

Some features of the code flash are not available directly, as in the device. They can be emulated via eserv commands.

For details, please refer to the eserv documentation.

(1) Self-programming

Flash self-programming can be supported by eserv command FLASHSELF.

(2) IDCODE

The IDCODE emulation will not physically modify the device's IDCODE. Please use the eserv command IDCODE.

(3) Mask options (option bytes)

The emulation of flash mask options will not physically modify the device's flash. Please use the eserv command OPBYTE.

(4) Security bits

The emulation of security features will not physically modify the device's flash. Please use the eserv command FSECFLAG. All device protections are available (chip erase, block erase, program, read-out, boot block cluster, flash shielding)

Note that the security settings apply to the emulated system, not the pod itself. It is safe to use these restrictions, the pod cannot be made unusable by these settings.

(5) Programmable reset vector

The emulation of the programmable reset vector feature will not physically modify the device's flash. Please use the eserv command FLASHRESET.

(6) Flash shield

The area that is protected by the flash shield feature can be defined by the eserv command FSHIELDWINDOW.

5.4 Limitations

This section describes limitations between the emulator and the device.

These limitations are due to design.

The functions of the device cannot be emulated fully, or even not at all.

5.4.1 VLVF function (very-low-voltage-indicator)

The VLVF flag is not supported.

Even when REG0VDD at the target drops below the threshold voltage, the flag does not indicate this event.

This is related to the fact that REG0VDD at the debug chip physically never drops below approx. 3.0V.

Effectively the VLVF flag reflects the true situation, i.e. that the voltage at the debug chip never dropped below the threshold and that RAM retention is guaranteed.

5.4.2 Flashwriter mode

The flashwriter mode is not supported.

Programming via a flash programmer such as the PG-FP5 or similar is not possible.

5.4.3 OCD mode

The OCD mode is not supported.

Debugging via the OCD interface e.g. by the E1 emulator is not possible.

It is only possible to debug using the QB-V850E2.

5.4.4 Boundary scan mode is not supported

Boundary scan via the JTAG port is not possible.

Please use a target device for this feature.

Revision History

Rev.	Date	Chapter	Page	Description
1.00	Jan 26, 2012	All	All	First edition

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

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

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130
Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220
Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900
Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327
Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679
Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898
Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044
Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670
Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001
Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510
Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2012 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

QB-70F3529-PD

Renesas Electronics Corporation

R20UT0991ED0100