CUSTOMER NOTIFICATION

ZUD-CD-10-0092 (1/19)
March 19, 2010
Yoshinari Ando, Team Manager Development Tool Solution Group Multipurpose Microcomputer Systems Division Microcomputer Operations Unit NEC Electronics Corporation

User’s Manual（Preliminary）

IECUBE2 main (QB-V850E2)

In-Circuit Emulator

Target Devices

V850E2M microcontroller

Contents

- **The information in this document is current as of March 2010. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "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.

  "Special": 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, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

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

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

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, USB interface cable, or connection to the target system was in an unsatisfactory state
- If the cable of the AC adapter, the USB interface cable, the POD probe, or the like was bent or pulled excessively
- When using an AC adapter not supported in the region of use
- If the product got wet
- If this product is connected to the target system when there is a potential difference between the GND of this product and GND of the target system.
- If the connectors or cables are plugged/unplugged while this product is in the power-on state.
- If excessive load is applied to the connectors or sockets.
- If a metal part of the power switch, cooling fan, or another such part comes in contact with an electrostatic charge
- If the product is used or stored in an environment where it may likely be exposed to electrostatic discharge or electrical noise

2. Safety precautions

- If used for a long time, the product may become hot (50°C to 60°C). Be careful of low 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 **1 Circumstances not covered by product guarantee**.

INTRODUCTION

Readers

This manual is intended for users who wish to perform debugging using the QB-V850E2 (generic name: IECUBE2). 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 IECUBE2.

Organization

This manual is divided into the following sections.

- General
- Names and functions of Hardware
- Optional function
- Notes

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 IECUBE2
→Read this manual in the order of the CONTENTS.

To know the manipulations, command functions, and other software-related settings of the IECUBE2
→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.

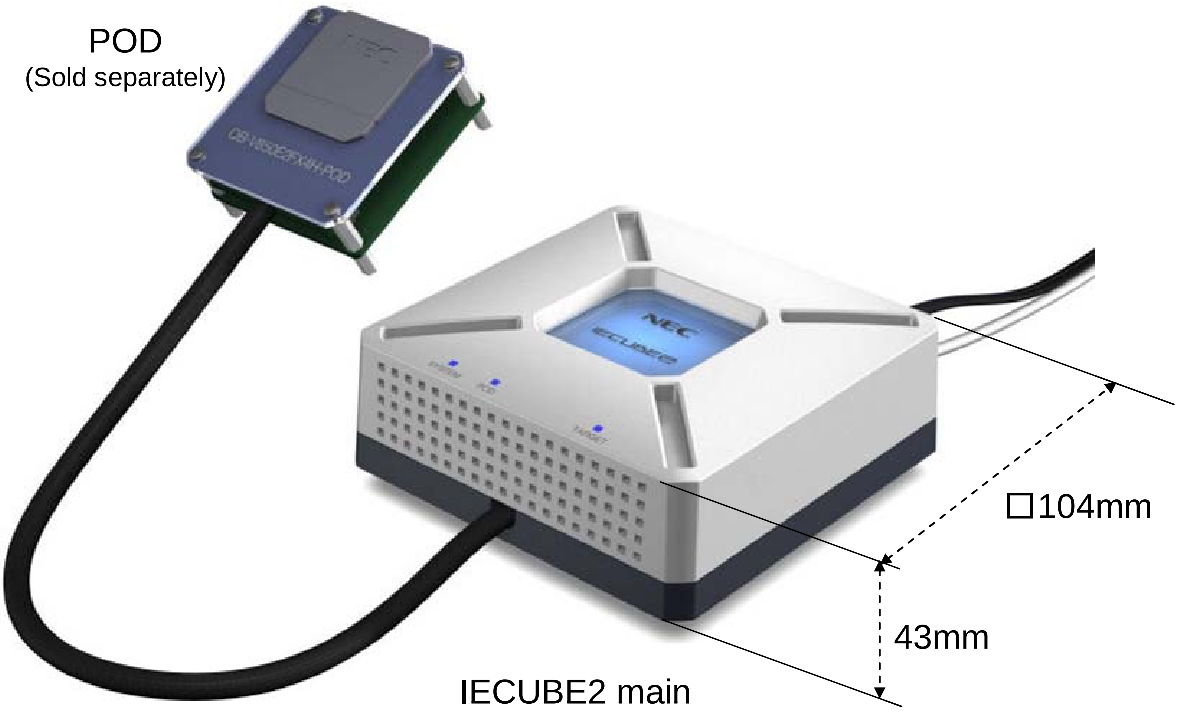
Term	Meaning
Target device	This is the device to be emulated.
Target system	This is the system to be debugged (system provided by the user). This includes the target program and the hardware provided by the user.
IECUBE®2	Generic name for NEC Electronics’ high-performance, compact in-circuit emulator.
POD	This is IECUBE2 peripheral to interface with the target system.

CHAPTER 1 GENERAL

The IECUBE2 is an in-circuit emulator for V850E2M microcontroller.

Hardware and software can be debugged efficiently in the development of systems in which the target device is used. This manual describes general information of IECUBE2 main unit. Please refer to user's manual of each POD for detail usage.

Figure 1-1. Description of external dimension



1.1 Hardware Specifications

The following table describes hardware specifications of IECUBE2 main.

Table 1-1. IECUBE2 Hardware Specifications

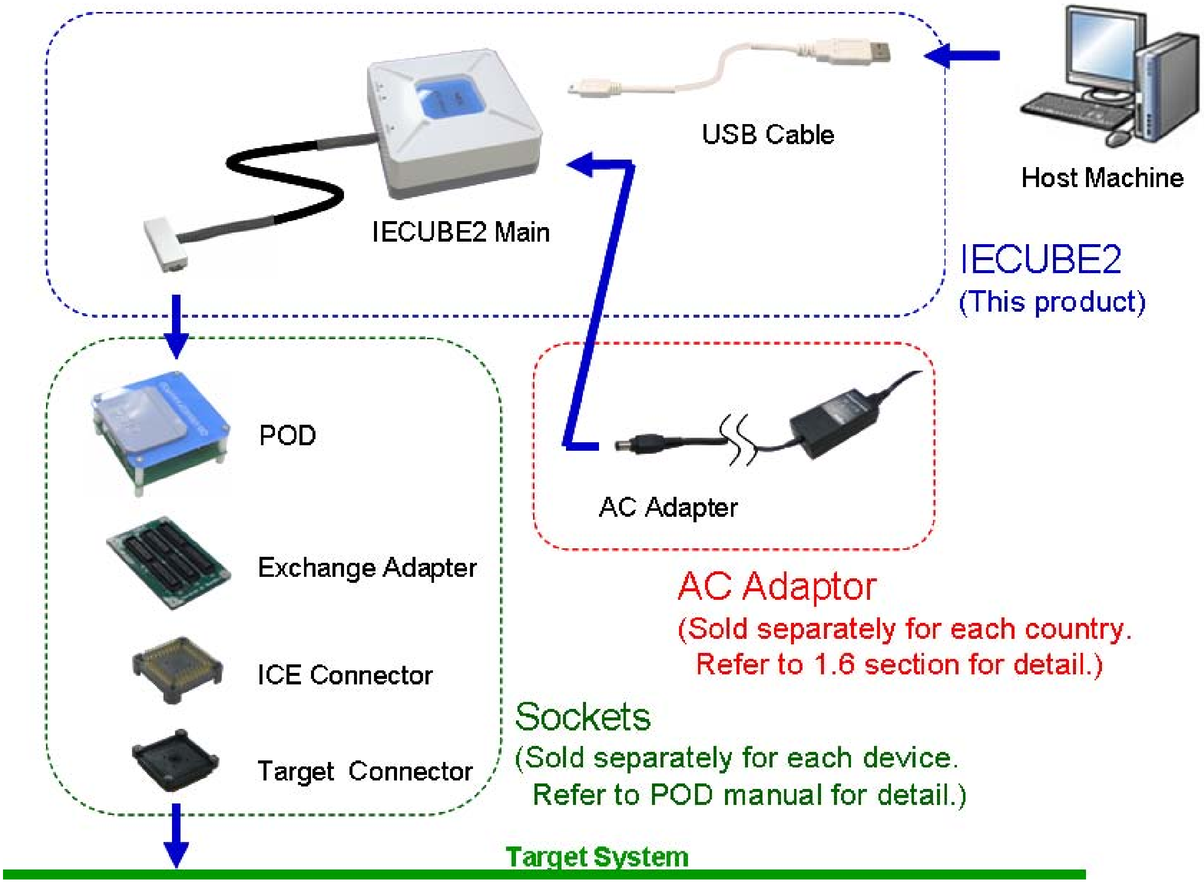
Parameter	Specification
Operating temperature range	0 to 40°C (No condensation)
Storage temperature range	−15 to 60°C (No condensation)
AC adapter power consumption	15 V, 4 A
Weight	Approx. 500 g
Host interface	USB interface (1.1, 2.0)
Trace memory	9M Byte (Approx. 512K frame)
	2.25G Byte (Approx. 128M frame) (When using long term trace option)

1.2 System Overview

The system configuration is described as below. IECUBE2 can not be used alone. POD, AC adapter and sockets are needed to use. These are sold separately.

To use IECUBE2, refer to the user’s manual of each POD.

Figure 1-2. System Configuration



1.3 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.

1.3.1 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.

1.3.2 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.

1.3.3 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 break function, 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 2 ms, a break can be set when at least 2 ms 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.

1.3.4 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 traces 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.

1.3.5 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.

1.3.6 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)

1.3.7 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.

1.3.8 Peripheral break function

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 POD user’s manual which peripheral functions can be stopped.

1.3.9 Mask function

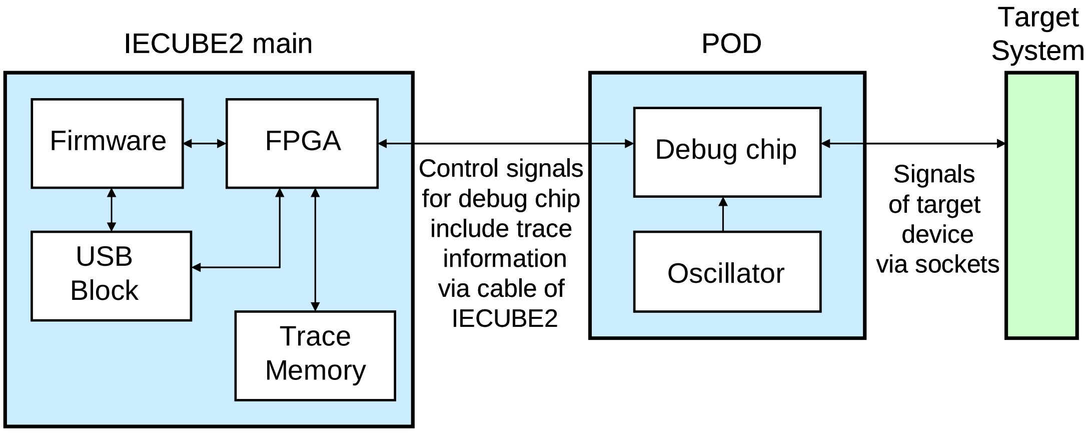
The mask function can be used to mask the following sources.

- _RESET terminal
- internal reset (For example, watch dog timer)

1.4 Block Overview

An internal block overview of the functions is described as below.

Figure 1-3. System Configuration


IECUBE2 main
POD
Target System
Firmware
FPGA
USB Block
Trace Memory
Control signals for debug chip include trace information via cable of IECUBE2
Debug chip
Oscillator
Signals of target device via sockets

1.5 Package Contents

QB-V850E2 package includes the items below. The list contains only items which are delivered commonly to all regions and that depending on region more items may be available. Therefore, confirm that the items in the attached packing list.

Products supplied with QB-V850E2

1: IECUBE2 main
2: USB interface cable
3: Stick for unplugging EA from POD
4: Table of Toxic and Hazardous Substance and Elements

1.6 AC adaptor

The specifications of the AC adapter for IECUBE2 differ depending on the region of use.
Be sure to use an AC adapter corresponding to the region of use.

The IECUBE2 is not provided with an AC adapter. It must be purchased separately.
Note: In European region, AC adapter is included and need not to be purchased separately.

Table 1-2. Part Numbers of AC Adapter for IECUBE2 Classified by Region

Product	Destination (Region)Notes 1, 2	Part NumberNote 3
AC adapter (sold separately)	Japan	QB-V850E2-PW-JP
	USA	QB-V850E2-PW-EA
-	Europe	(AC adapter is included in IECUBE2)

Notes 1. Products are shipped only on order from each region.

2. Contact a distributor or an NEC Electronics sales representative for information on regions other than the above.

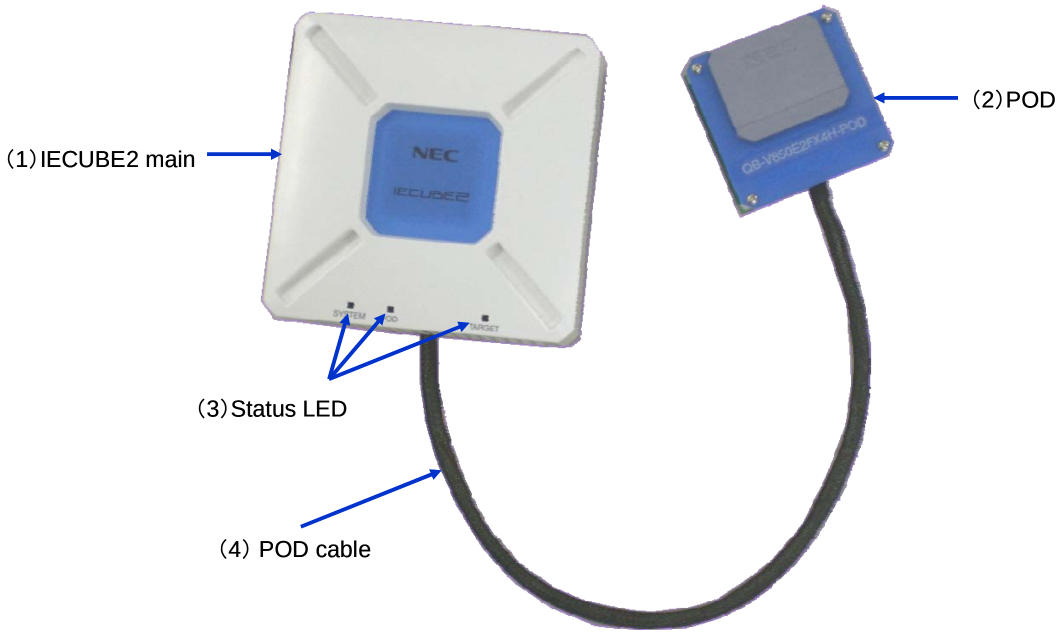
3. Only the AC adapter usable in each region can be ordered.

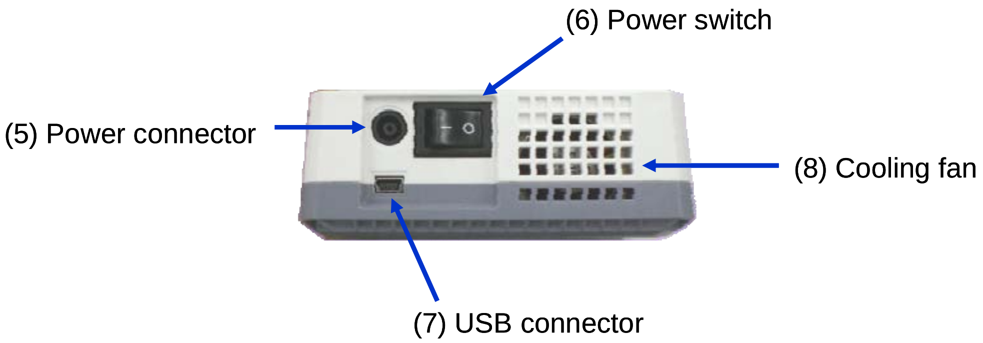
CHAPTER 2 Names and Functions of Hardware

2.1 IECUBE2 main

The following shows the names of IECUBE2 hardware units and their features.

Figure 2-1. Names of parts of IECUBE2





(1) IECUBE2 main

IECUBE2 main is unit that controls debugging

(2) POD

POD is sold separately. POD is IECUBE2 peripheral to interface with the target system. This unit is equipped with the major features for emulating the real device. Please refer to user's manual of POD for details.

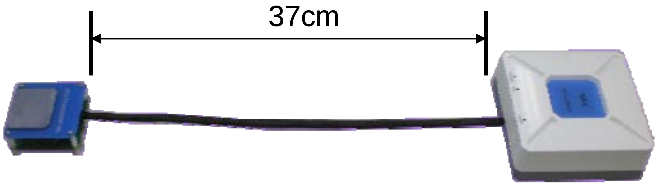
(3) Status LED

The status LEDs turn on or blink according to specific causes as described in the table below. If any LED does not turn on, IECUBE2 might be broken. In this case, contact an NEC 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 IECUBE2 is not running correctly. In this case, IECUBE2 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 IECUBE2 main unit and emulation POD. The cable length is shown below. Be careful not to excessively bend this cable because doing so might break the cable.



(5) Power connector

This connector is for the power supply cable.

(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 fan

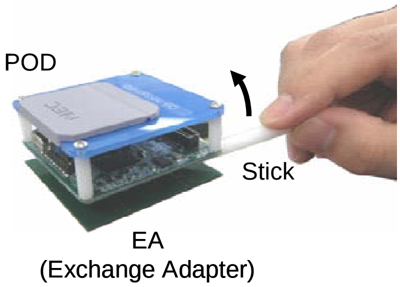
This fan cools down the IECUBE2 internal units. Be careful not to obstruct the vents.

2.2 Stick for unplugging EA from POD

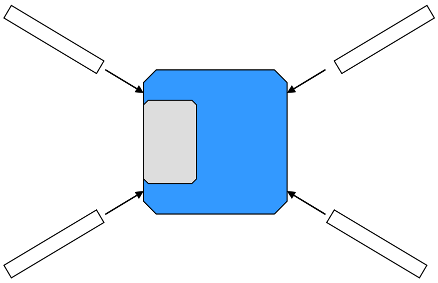
The stick as following figure is used to unplug Exchange Adapter (EA) from POD.

Figure 2-2. Stick for unplugging EA from POD

To unplug EA from POD, insert the stick in a corner of POD and slightly push up POD.



Do the same thing in other corners as below.



.

CHAPTER 3 NOTES

This chapter explains the common notes of IECUBE2.

3.1 Note for time stamp of trace function

A time stamp of trace function does not indicate the instruction execution time/cycle count. A time stamp indicates a time IECUBE2 main received the trace data from a debug chip in POD. The output timing of the trace data is a little different from the instruction execution time/cycle. And in case of including more than one trace data per execution, such as access address and access data, some errors may be observed by FIFO queuing.

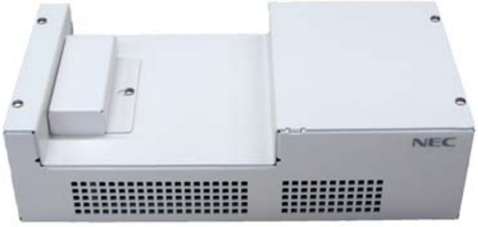
CHAPTER 4 LONG TERM TRACE OPTION

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

4.1 General

The QB-V850E2-SP is a trace memory extension for IECUBE2. Please make sure the supported version of debugger software.

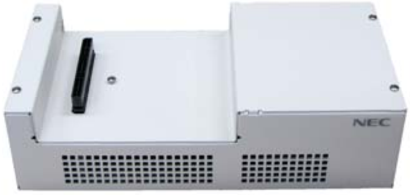
Figure 4-1. QB-V850E2-SP



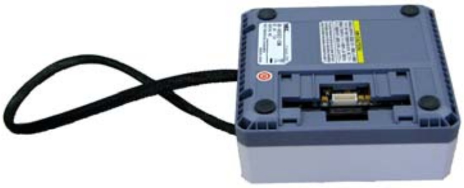
4.2 SETUP PROCEDURE

This section describes how to connect the QB-V850E2-SP to the IECUBE2 main.

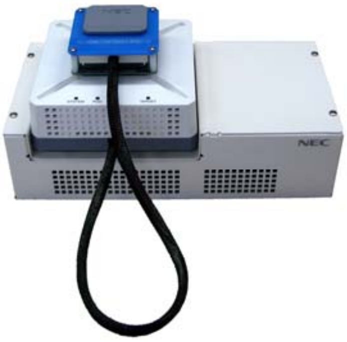
1. Remove the cover from the connector on the top side of the QB-V850E2-SP module. It might be necessary to loose the screw a little bit.



2. Make sure that the IECUBE2 is switched off and the USB cable and power supply adapter is not connected to the IECUBE2 main module. Then Remove the cover on the bottom side of the IECUBE2 main module.



3. Mount the IECUBE2 main module on the QB-V850E2-SP as shown in the picture. Now connect the USB cable and power supply adapter to the IECUBE2 and turn the IECUBE2 on.



IECUBE2 detects trace memory extension automatically when QB-V850E2-SP is connected. Configuration in hardware or debugger software is not necessary.