To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)
Send any inquiries to http://www.renesas.com/inquiry.

Note that the following URLs in this document are not available:
http://www.necel.com/
http://www2.renesas.com/

Please refer to the following instead:
Development Tools | http://www.renesas.com/tools
Download | http://www.renesas.com/tool_download

For any inquiries or feedback, please contact your region.
http://www.renesas.com/inquiry
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.
User’s Manual

QB-MINI2
Setup Manual

Partner Tool

Target Devices
V850 Microcontrollers
The information in this document is current as of March, 2008. 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).
INTRODUCTION

Readers
This manual is intended for engineers who use QB-MINI2 (alias: MINICUBE2) as well as MULTI, manufactured by Green Hills Software (hereinafter referred to as “MULTI”) for debugging.
Engineers who read this manual are presumed to have a good knowledge of the functions of the device and how to use it as well as debuggers.

Purpose
The purpose of this manual is to help users understand the basic ways of using MINICUBE2 and MULTI when introducing them.

Organization
This manual is divided into the following sections.

• General
• Software tool installation
• User program preparation
• System start
• Multi debugger start
• Interrupt source names and start options
• MINICUBE2 self-diagnosis and firmware update

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.

To understand the basic specifications and usages of MINICUBE2
→ Read this manual according to the CHAPTER 1 GENERAL.

To know the manipulations, command functions, and other software-related settings of MINICUBE2
→ 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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINICUBE2</td>
<td>Generic name of QB-MINI2</td>
</tr>
<tr>
<td>Target device</td>
<td>This is the device to be emulated.</td>
</tr>
<tr>
<td>Target system</td>
<td>This is the system to be debugged (user-created system). Refers to all hardware and software provided by the user.</td>
</tr>
<tr>
<td>MULTI</td>
<td>Integrated development environment MULTI, manufactured by Green Hills Software</td>
</tr>
</tbody>
</table>

Related Documents

Please use the following documents in conjunction with this manual.

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

Documents Related to Development Tools (User’s Manuals)

<table>
<thead>
<tr>
<th>Document Name</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB-MINI2 Operating Precautions</td>
<td>ZUD-CD-07-1212-E</td>
</tr>
<tr>
<td>QB-Programmer Programming GUI Operation User’s Manual</td>
<td>U18527E</td>
</tr>
<tr>
<td>MINICUBE2 Diagnostic Tool User’s Manual</td>
<td>U18588E</td>
</tr>
<tr>
<td>MINICUBE OCD Checker User’s Manual</td>
<td>U18591E</td>
</tr>
</tbody>
</table>

Caution  The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing, etc.
CONTENTS

CHAPTER 1  GENERAL ................................................................. 8

CHAPTER 2  SOFTWARE TOOL INSTALLATION ........................................... 9

CHAPTER 3  USER PROGRAM PREPARATION ............................................. 10

  3. 1 Securing Memory Space and Setting Security ID .................................. 10

  3. 2 Securing Serial Interface for Communication ...................................... 15

CHAPTER 4  SYSTEM START .......................................................... 16

CHAPTER 5  MULTI DEBUGGER START ..................................................... 18

APPENDIX A  INTERRUPT SOURCE NAMES AND START OPTIONS .................. 23

APPENDIX B  MINICUBE2 SELF-DIAGNOSIS AND FIRMWARE UPDATE ............... 24
CHAPTER 1  GENERAL

MULTI, manufactured by Green Hills Software, provides a high-performance and user-friendly integrated development environment, based on a unified GUI. Programs can be developed efficiently when MULTI is used together with on-chip debug emulator MINICUBE2 with a programming function.

A combined environment of MULTI and MINICUBE2 can be easily built by setting up a system according to this manual.

Figure 1-1. Environment By Combining MINICUBE2 And MULTI

Building a debugging environment for the target system by combining MINICUBE2 and MULTI

An overview of the setup flow is shown below. The setup is described in the following chapter.

Figure 1-2. setup flow

- Software tool installation: Required software tools are installed.
- User program preparation: Memory space required for debugging is secured and security ID settings are described to a program.
- System start: MINICUBE2, the host machine, and the target system are connected, and power is activated.
- MULTI debugger start: The 850eserv startup option is set and the MULTI debugger is started.
CHAPTER 2 SOFTWARE TOOL INSTALLATION

Install the following software tools.

(1) MULTI debugger
This is the integrated development environment MULTI debugger manufactured by Green Hills Software.
Execute setup.exe to install the debugger. For details, see the document related to MULTI.

(2) 850eserv
This is the debug server used to connect MINICUBE2 and the MULTI debugger. It is included with the integrated development environment MULTI and should be installed together with MULTI. Use the latest version of 850eserv, due to the relation with the Exec library, described in (3), below.

(3) Exec library
This is the dynamic link library called by 850eserv to control MINICUBE2.
Due to the relation with 850eserv, described in (2), above, download the latest version of the Exec library from the following Web sites and copy the complete set of extracted files to the same folder as MULTI.exe.


(4) USB driver for MINICUBE2
This USB driver is required to connect the host machine and MINICUBE2.
Download it from the same Web site as the Exec library, described in (3), above, and extract it.
A plug-and-play dialog will be displayed when MINICUBE2 and the host machine are connected after the USB driver file has been extracted. Specify as the destination for storing the USB driver the folder that was extracted before (QB2ALL).

(5) Device file
This file holds device-specific information and is used by 850eserv.
Download it from the following Web sites and extract it. The device file name after extracting (.800 file extension) and the path information of the extracted folder are used for specifying options when starting the MULTI debugger. For details, see CHAPTER 5 MULTI DEBUGGER START.


CHAPTER 3 USER PROGRAM PREPARATION

The following preparations must be performed for user programs so that MINICUBE2 can communicate with the target device and perform each debug function. These preparations require the user programs and linker directive to be edited. Refer to the following 3.1 Securing Memory Space and Setting Security ID to perform editing and building.

3.1 Securing Memory Space and Setting Security ID

The shaded areas in Figure 3-1 are areas to which user programs and data cannot be placed because a monitoring program for debugging will be built in. The 10-byte areas shown in yellow are areas to which ID codes are embedded so that the contents of a memory cannot be read by a third party. When using MULTI, MINICUBE2 can be started when the security ID and embedded ID match as the argument of 850eserv start option “-id”. Be sure, therefore, to surely manage the embedded security ID code. If the ID code is forgotten, the flash memory must be erased and the debugger must be restarted with the ID code set as “0xffffffffffffffff” (10 bytes).

To secure memory space and set the security ID, describe the assemble source and linker directive source, shown on the next page.

Figure 3-1. Memory Spaces Is Placed A Monitoring Program For Debugging

- Monitor program area for debugging

Note: When there is a receive error interrupt or a receive status interrupt, the corresponding vector must be secured.
(1) Example 1 of securing memory space and setting the security ID

- Program description (Add the following as the assemble source.)

```
--### DBG0 Vector ###
.section ".DBG0", "ax" -- 0x60
.word 0xffffffff

--### Security ID ###
--### Set ID code arbitrarily. ###
.section ".S_CODE", "a" -- 0x70
.byte 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

--### Serial Receive Interrupt Vector ###
.section ".INTCB0R", "ax" Note
.word 0xffffffff

--### ROM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_ROM", "ax"
.rept 0x200
.word 0xffffffff
.endr

--### RAM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_RAM", "abw"
.global monitorramsym
monitorramsym:
.space 16
```

**Note** This description is applicable when CSIB0 is used as the interface of MINICUBE2 and the target device. Change the “INTCB0R” section according to the receive interrupt source of serial communication. See Appendix A for interrupt source names. They are also listed in the user’s manual of the target device. Refer to the following page when there is a receive error interrupt or a receive status interrupt.
• Linker directive description (Insert the following into the linker directive.)

The following example is applicable when the internal ROM size is 256 KB and the internal RAM end address is 0x3ffefff.

```
MEMORY
{
  iROM : ORIGIN = 0x00000000 , LENGTH = 256k-2k
  MC2ROM : ORIGIN = 254k , LENGTH = 2k
  iRAM : ORIGIN = 0x03FFC000, LENGTH = 12k-16
  MC2RAM : ORIGIN = 0x03FFE000, LENGTH = 16
}

SECTIONS
{
  .RESET  0x00000000 :> iROM
  .DBG0  0x00000060 :> iROM
  .S_CODE  0x00000070 :> iROM
  .INTCB0R  0x00000230 :> iROM  //Note
  .MC2MON_ROM  :> MC2ROM
  .MC2MON_RAM  :> MC2RAM
}
```

**Note** Use the source name described in the program as the interrupt source name (INTCB0R section). Also, describe as the address a vector address corresponding to the interrupt source. See the user’s manual of the target device for vector addresses.
(2) Example 2 of securing memory space and setting the security ID (when there is a receive status interrupt during serial communication)

- Program description (Add the following as the assemble source.)

--### DBG0 Vector ###
.section ".DBG0", "ax" -- 0x60
.word 0xffffffff

--### Security ID ###
--### Set ID code arbitrarily. ###
.section ".S_CODE", "a" -- 0x70
.byte 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff

--### Serial Receive & Status Interrupt Vector ###
.section ".INTUD0S", "ax" -- Note
.word 0xffffffff
.section ".INTUD0R", "ax" -- Note
.word 0xffffffff

--### ROM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_ROM", "ax"
.rept 0x200
.word 0xffffffff
.endr

--### RAM area secured for MINICUBE2 MONITOR ###
.section ".MC2MON_RAM", "abw"
.global monitorramsym
monitorramsym:
.space 16

Note This description is applicable when UARTD0 is used as the interface of MINICUBE2 and the target device. Change the interrupt name according to the receive interrupt and receive status interrupt of serial communication. See Appendix A for interrupt source names. They are also listed in the user's manual of the target device.
• Linker directive description (Insert the following into the linker directive.)

The following example is applicable when the internal ROM size is 256 KB and the internal RAM end address is 0x3ffefff.

```plaintext
MEMORY
{
  iROM : ORIGIN = 0x00000000, LENGTH = 256k-2k
  MC2ROM : ORIGIN = 254k, LENGTH = 2k
  iRAM : ORIGIN = 0x03FFC000, LENGTH = 12k-16
  MC2RAM : ORIGIN = 0x03FFE000, LENGTH = 16
}

SECTIONS
{
  .RESET  0x00000000 :> iROM
  .DBG0  0x00000060 :> iROM
  .S_CODE  0x00000070 :> iROM
  .INTUD0S  0x000002B0 :> iROM  // Note
  .INTUD0R  0x000002C0 :> iROM  // Note
  .MC2MON_ROM :> MC2ROM
  .MC2MON_RAM :> MC2RAM
}

Note  Use the source names described in the program as the interrupt source names. Also, describe as the address a vector address corresponding to the interrupt source. See the user’s manual of the target device for vector addresses.
3. 2 Securing Serial Interface for Communication

Create the user program, paying attention to the following items.

- **Serial interface registers**
  Do not use the user program for setting the registers related to UART and CSI-H/S, which are used for communication.

- **Serial communication interrupt mask registers**
  Generally, do not use the user program for changing the interrupt mask registers of UART and CSI-H/S, which are used for communication. To use the IMR register to change the interrupt mask settings in a batch, do not change the setting values of the target interrupt masks by a read-modify-write operation.

- **Port-related registers**
  To use UART for communication, do not perform register settings for ports such that the TxD and RxD pins become invalid. For CSI-H/S, do not perform register settings for ports such that the SI, SO, SCK, and H/S pins become invalid. The H/S pin is used as the port output for debugging.

Two examples are described below.

*Example 1* Settings other than the following are prohibited when the V850ES/KJ2 is the target device and UART0 is used.

<table>
<thead>
<tr>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFC3</td>
</tr>
<tr>
<td>x x x x x x x 0</td>
</tr>
<tr>
<td>x: Arbitrary</td>
</tr>
<tr>
<td>PMC3L</td>
</tr>
<tr>
<td>x x x x x x 1 1</td>
</tr>
<tr>
<td>x: Arbitrary</td>
</tr>
</tbody>
</table>

*Example 2* Settings other than the following are prohibited when the V850ES/HG2 is the target device and CSIB0 is used.

<table>
<thead>
<tr>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMC4</td>
</tr>
<tr>
<td>x x x x x 1 1 1</td>
</tr>
<tr>
<td>x: Arbitrary</td>
</tr>
<tr>
<td>PMCCM</td>
</tr>
<tr>
<td>x x x x x x 0</td>
</tr>
<tr>
<td>x: Arbitrary</td>
</tr>
<tr>
<td>PMCM</td>
</tr>
<tr>
<td>x x x x x x 0</td>
</tr>
<tr>
<td>x: Arbitrary</td>
</tr>
<tr>
<td>PCM</td>
</tr>
<tr>
<td>x x x x x x</td>
</tr>
<tr>
<td>Writing</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Note</td>
</tr>
</tbody>
</table>

*Note* The monitoring program changes the port value corresponding to the H/S pin, according to the debugger status. When port register settings are to be manipulated in 8-bit units, there will be no problem if read-modify-writing is performed in the user program, but an unintended operation may result if an interrupt for debugging occurs before writing.
CHAPTER 4 SYSTEM START

Start the system by using the following procedure.

(1) MINICUBE2 switch setting

Set the mode selection switch to M2 (V850 microcontroller connection setting). Set the power supply selection switch, referring to the following table.

**Caution** Do not toggle the MINICUBE2 switch when a USB cable is connected. Remove the USB cable when toggling the switch.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Supplies 3 V from MINICUBE2 to the target system. <em>(Note)</em></td>
</tr>
<tr>
<td>5</td>
<td>Supplies 5 V from MINICUBE2 to the target system. <em>(Note)</em></td>
</tr>
<tr>
<td>T</td>
<td>Uses the target system power supply. MINICUBE2 detects the target system power supply. Also, MINICUBE2 uses that power supply for the communication interface.</td>
</tr>
</tbody>
</table>

**Note** The maximum rated current is 100 mA. Do not use a target system exceeding this value. Also, power is always supplied when MINICUBE2 and the host machine are connected.

(2) Target system connection

Connect MINICUBE2 and the target system as shown in the Figure 4-1. At this time, the power of the target system must be turned off.

![Figure 4-1. Connection of MINICUBE2 and Target System](image)
(3) USB connection

Connect MINICUBE2 and the target system as shown in the Figure 4-2. At this time, the power of the target system must be turned off.

When the power supply selection switch is set to “T”, the mode LED blinks in white after connection.

When the power supply selection switch is set to “3” or “5”, the mode LED glows steadily in white after connection.

Figure 4-2. Connection of MINICUBE2 and Host Machine

MINICUBE2  USB cable  Host machine

(4) Target system power activation

Activate the power of the target system. When the power supply selection switch is set to “3” or “5”, this step is not required. The mode LED glows steadily in white after power activation.
CHAPTER 5  MULTI DEBUGGER START

This chapter describes the procedure up to starting the MULTI debugger.

(1) Connection organizer start
Click the target connection button in the MULTI launcher to open the connection organizer, as shown below.

(2) Creating a new method
Select [New] from the [Method] menu in the connection organizer menu bar, as shown below.
(3) Setting a new method

Set the items in the new method dialog box and click the [Create] button, as shown below.

Click the [Create] button.

Name: Enter an arbitrary name.
Type: Select as shown at left.

(4) Connection editor editing

Enter options in the [Other Options] field in the connection editor dialog box. See Table 5-1 for option contents. The entry in the figure below is as follows (the tail of the entry is hidden in the figure).

```
-minicube2 -noint -p=csib0 -df=df3707.800 --id ffffffff -ip= D:\device_file\v850e
```

Click the [Connect] button after the options have been entered.
Table 5-1. Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Content</th>
<th>Necessity of Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-minicube2</td>
<td>Option to be set when MINICUBE2 is the target.</td>
<td>Required option</td>
</tr>
<tr>
<td>-noint</td>
<td>Sets to non-interactive mode when a pio, a register, or an sfr command is issued. When non-interactive mode is set, a wait for input is not performed when the option is referenced.</td>
<td>Recommended option</td>
</tr>
<tr>
<td>-p</td>
<td>Specifies the name of the serial communication that connects MINICUBE2. Specify it by referring to APPENDIX A, INTERRUPT SOURCE NAMES AND START OPTIONS.</td>
<td>Required option</td>
</tr>
<tr>
<td>-df</td>
<td>Specify the device file name (.800 file extension) according to the target device. See the device file user’s manual (pdf) expanded in the folder generated when the device file was extracted (see CHAPTER 2, (5) Device file).</td>
<td></td>
</tr>
<tr>
<td>-ip</td>
<td>Specify the folder in which the device file is stored.</td>
<td></td>
</tr>
<tr>
<td>-id</td>
<td>Specify the security ID.</td>
<td></td>
</tr>
<tr>
<td>-noiop</td>
<td>Option enabling referencing of programmable I/O area memories and changing memories. It is set when using the memory window to reference programmable I/O areas.</td>
<td>As required</td>
</tr>
<tr>
<td>-X0</td>
<td>Does not clear the BBS area to zero when downloading (default).</td>
<td></td>
</tr>
<tr>
<td>-X1</td>
<td>Specify this option when clearing the BBS area to zero when downloading.</td>
<td></td>
</tr>
</tbody>
</table>

(5) MINICUBE2 connection check

When the [Connect] button in the connection editor dialog box is clicked, 850eserve and MINICUBE2 communicate, and if a normal connection between them is confirmed, “Connected” will be displayed in the status column of the connection organizer, as shown below.

See the “V850E ICE Server Reference Manual” (document included in the MULTI environment set) when an error has occurred.
The following error message will be displayed when the -p parameter is not set normally. In such a case, confirm whether the option descriptions of the connection editor are correct.

![Error message]

"-minicube2" Error. Please add "-p=<portname>".
Selectable port: UARTA0/CSI00
Connection: No remote connection established.
(6) MULTI debugger start and clock settings

Start the MULTI debugger from the MULTI launcher.

Execute the dclock command from the command pane, as follows. In the example, the main clock is set to 5 MHz and the sub-clock is set to 32.768 kHz. Change the settings as required. Specify an oscillation clock (a clock before PLL multiplication) as the main clock.

MULTI > dclock 5000 32768 swoff

The base window of the debugger immediately after steps (1) to (6) have been executed and the program has been downloaded is shown below. See the following documents for the debug operations hereinafter.

- MULTI debugger command
- MULTI User’s Guide Debugger
- 850eserv target command
- V850E ICE Server Reference Manual

Execute the dclock command from the command pane.
## APPENDIX A  INTERRUPT SOURCE NAMES AND START OPTIONS

<table>
<thead>
<tr>
<th>Target Device</th>
<th>MINICUBE2 Connection Serial Communication</th>
<th>Interrupt Vector</th>
<th>-p Option Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>V850ES/Fx3</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>UARTD0</td>
<td>INTUD0S (UARTD0 status interrupt)</td>
<td>uard0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td>V850ES/FG3L</td>
<td>CSIB0</td>
<td>INTUD0S (UARTD0 status interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>UARTD0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td>V850ES/Hx2</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td>V850ES/Hx3</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>UARTD0</td>
<td>INTUD0R (UARTD0 receive complete interrupt)</td>
<td>uard0</td>
</tr>
<tr>
<td>V850E/IA4</td>
<td>CSIB0</td>
<td>INTCB0RE (CSIB0 receive error interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0RE (UARTA0 receive error interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td>V850ES/IE2</td>
<td>CSIB0</td>
<td>INTCB0RE (CSIB0 receive error interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0RE (UARTA0 receive error interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td>V850ES/Ix3</td>
<td>CSIB0</td>
<td>INTCB0RE (CSIB0 receive error interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0RE (UARTA0 receive error interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td></td>
</tr>
<tr>
<td>V850ES/Jx2</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>CSIB3</td>
<td>INTCB3R (CSIB3 receive complete interrupt)</td>
<td>csib3</td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0R (CSIB0 receive complete interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td>V850ES/Jx3</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>CSIB3</td>
<td>INTCB3R (CSIB3 receive complete interrupt)</td>
<td>csib3</td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td>V850ES/Kx1+</td>
<td>CSIB0</td>
<td>INTCB0R (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>CSIB3</td>
<td>INTCB3R (CSIB3 receive complete interrupt)</td>
<td>csib3</td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTUA0R (UARTA0 receive complete interrupt)</td>
<td>uarta0</td>
</tr>
<tr>
<td>V850E/MA3</td>
<td>CSIB0</td>
<td>INTCSI00 (CSIB0 receive complete interrupt)</td>
<td>csib0</td>
</tr>
<tr>
<td></td>
<td>UARTA0</td>
<td>INTSR0 (UARTA0 receive complete interrupt)</td>
<td>uarta0</td>
</tr>
</tbody>
</table>

User's Manual U19158EJ1V0UM 23
APPENDIX B  MINICUBE2 SELF-DIAGNOSIS AND FIRMWARE UPDATE

When MINICUBE2 is not operating correctly, the MINICUBE utilities can be used to perform a self-diagnosis. The MINICUBE utilities are also used when updating the internal firmware of MINICUBE2. The MINICUBE utilities can be downloaded from the following Web sites. See the included documents for how to use them.
