

CUSTOMER NOTIFICATION

SUD-DT-03-0277-E
July 3, 2003
Koji Nishibayashi, Senior System Integrator Microcomputer Group 2nd Solutions Division Solutions Operations Unit NEC Electronics Corporation

CP(K),O

V850 Series Integrated Debugger

ID850NW V2.52

Operating Precautions

For PC-9800 Series (Windows™ based)
and IBM PC/AT Compatibles (Windows™ based)

Be sure to read this document before using the product.

- 1. ID850NW V2.52 OPERATING ENVIRONMENT2
- 2. FILES NECESSARY FOR EXECUTING ID850NW V2.523
- 3. INSTALLATION4
 - 3.1 Device Driver4
 - 3.2 ID850NW4
 - 3.3 Device File6
- 4. UNINSTALLATION7
 - 4.1 ID850NW7
- 5. MAJOR REVISIONS FROM ID850NW V1.11 TO V2.52.....9
- 6. RESTRICTIONS AND CAUTIONS11
 - 6.1 Restrictions11
 - 6.2 Cautions14

Thank you for purchasing the integrated debugger ID850NW.

This document describes the points to be noted when using this integrated debugger. Be sure to read this document before using the integrated debugger.

For details of how to use and the functions of the integrated debugger, refer to the online help supplied with the product or User's Manual "ID850NW Integrated Debugger V2.51 Operation (Windows™-based)" (document number: U16525E).

Also refer to the document "ID850NW Ver. 2.51 Changes in Operation User's Manual (document number: SUD-DT-03-0276-E)".

1. ID850NW V2.52 OPERATING ENVIRONMENT

(1) Host machine

PC-9821 series, PC98-NX series, and IBM PC/AT compatible machines

CPU: Pentium 166 MHz or above

Main memory: 64 MB or more (128 MB or more recommended)

OS: Windows 98, Windows NT 4.0, Windows 2000, Windows Me, Windows XP

* It is recommended to install the latest service pack for each OS.

(2) Supported devices

<1> When the NEC Electronics emulator is connected

Target Device	Emulator
Device with debug control unit, microcontroller with Nx85ET core	IE-70000-MC-NW-A

<2> When the Midas Lab emulator is connected

Target Kit
<ul style="list-style-type: none"> • KIT-NB85E-TP • KIT-V850E/ME2-TP • KIT-V850E/MA3-TP • KIT-V850E/MA3-IE

When the Midas Lab emulator is used, a DLL for connection (N-EXEC for RTE) is necessary.

This item can be downloaded from the following URL.

http://www.midas.co.jp/download/pro_sam.htm#N-EXEC

(3) Supported tools

- C compiler package: CA850 V2.50 or later
 - System performance analyzer: AZ850 V3.10 or later
 - Performance analysis tuning tool: TW V1.10 or later^{Note}
- Note** TW V1.00 included in SP850 V3.00 is not supported.
- Interface card

<1> When the NEC Electronics interface card is used

- Device driver: IE-PC Driver V2.10 or later

Connected IE	Supported Interface Card
IE-70000-MC-NW-A	IE-70000-98-IF-B, -C, IE-70000-PC-IF-B, -C, IE-70000-PCI-IF or IE-70000-PCI-IF-A, IE-70000-CD-IF-A

<2> When the Midas Lab interface card is used

Consult Midas Lab.

2. FILES NECESSARY FOR EXECUTING ID850NW V2.52

The following files are necessary for using ID850NW V2.52.

(1) ID850NW system file (this product)

Included in the folder "ID850NW" in the CD-ROM.

(2) Device driver (supplied with this product)

Included in the folder "ID850NWDIRECTOR" in the CD-ROM.

(3) Device file (supplied with this product)

Programs unique to each device are supplied in the device file.

Be sure to use the device file corresponding to the device to be debugged.

Note that the version of the corresponding device file may differ depending on the version of the EVA chip in the in-circuit emulator. Refer to the document included with the device file or in-circuit emulator to be used.

3. INSTALLATION

Before installing ID850NW V2.52, it is necessary to install the following.

- Device driver
- ID850NW
- Device file

3.1 Device Driver

Install the device driver by following the instructions in "README_E.TXT" in the folder "ID850NW \DRIVER\" on the CD-ROM.

3.2 ID850NW

3.2.1 Notes on installation

- <1> Because it may be necessary to restart the computer after installation, terminate all other applications.
- <2> To re-install ID850NW V2.52, uninstall the copy of ID850NW V2.52 already installed. If this product is installed in a different directory without uninstalling the first copy of ID850NW V2.52, the first copy of ID850NW V2.52 already installed cannot be uninstalled.
- <3> Do not install ID850NW V2.52 in a directory with a name containing a space or a multi-byte character; otherwise the related tools, such as the project manager, that are installed in the same directory as this product may not be correctly executed depending on their version.
- <4> The following file will be created after ID850NW V2.52 has been installed. This file is necessary for uninstalling ID850NW V2.52 and must not be deleted (the installation destination is assumed to be C:\NECTools32).
C:\NECTools32\SETUP*.*
- <5> The ID850NW help is supplied in the HTML help format. If the help is not displayed correctly, proceed as follows.
 - Install Microsoft Internet Explorer 5.0 or later (IE5.0 SP 2 or later recommended)
- <6> The font size of the help menu is affected by the font settings in Internet Explorer. The display is too large in the default size ("Medium") of Internet Explorer. Therefore, change the font size to "Smallest" in [Fonts] from the [View] menu and restart the help.
In case of IE 4.0, however, the font size for the help menu cannot be changed.

- <7> The project manager is not supported in ID850NW V2.52 or later. Use PM plus.
PM plus is included in CA850 V2.50 or later, so upgrade to CA850 to V2.50 or later. If the project manager is used with ID850NW V2.50 or earlier, the following warning is displayed.
Wf109: project manager cannot be used with the debugger of this version. Please use PM plus.

- <8> ID850NW V2.52 can be installed in the English Windows environment only.
If an attempt is made to install ID850NW V2.52 in an environment other than English Windows, a message indicating a file transfer error is displayed and the installation is aborted.
This situation also occurs if a language other than English is specified as the system language in the "Regional Settings Properties" tab.

3.2.2 Installation procedure

This section explains the installation procedure, assuming that ID850NW V2.52 is installed under the directory "C:\nectools32".

- <1> Turn on power to the host machine and start Windows.

- <2> Insert the CD-ROM in the CD-ROM drive. The installer is started automatically.
If it does not start automatically, start the program "setup.exe" from the directory NEC_ID850NW\ID850NW\DISK1. After starting the program, follow the directions on the screen.

- <3> The following file is necessary for uninstalling ID850NW V2.52 and must not be deleted (the installation destination is assumed to be C:\NECTools32).
C:\NECTools32\SETUP*.*

3.3 Device File

To install the device file, use the dedicated installer "DFINST.EXE" supplied with ID850NW V2.52 or other 32-bit version products. This installer allows the device file to be installed in the Win32 environment.

In the Win32 environment of ID850NW V2.52, be sure to use "DFINST" because it is necessary to register the device file information in the registry.

The procedure for installing the device file is explained below. For details of the dedicated installer DFINST, refer to the online help provided in DFINST.

- <1> Start "DFINST.EXE". To install a new device file, click on the opening screen that is displayed after the installer has been started, and select the install information file "NECSETUP.INI" from the device file product disk.

- <2> To move a device file already registered in "NECDEV.INI" to the Win32 environment, select "Select Source" in DFINST and specify "NECDEV.INI". Select a model from those displayed in the "Source" field, and click .

- <3> To uninstall the device file, select a model from those displayed in the "Registry" field in DFINST, check "Delete File", and then click .

4. UNINSTALLATION

4.1 ID850NW

This section describes how to uninstall ID850NW V2.52.

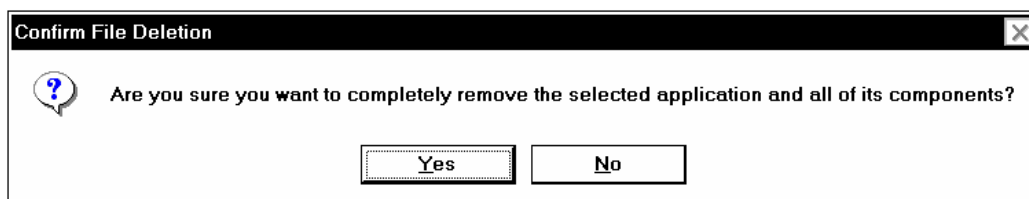
<1> Turn on power to the host machine and start Windows.

<2> Start "Add /Remove Programs" in the Control Panel.

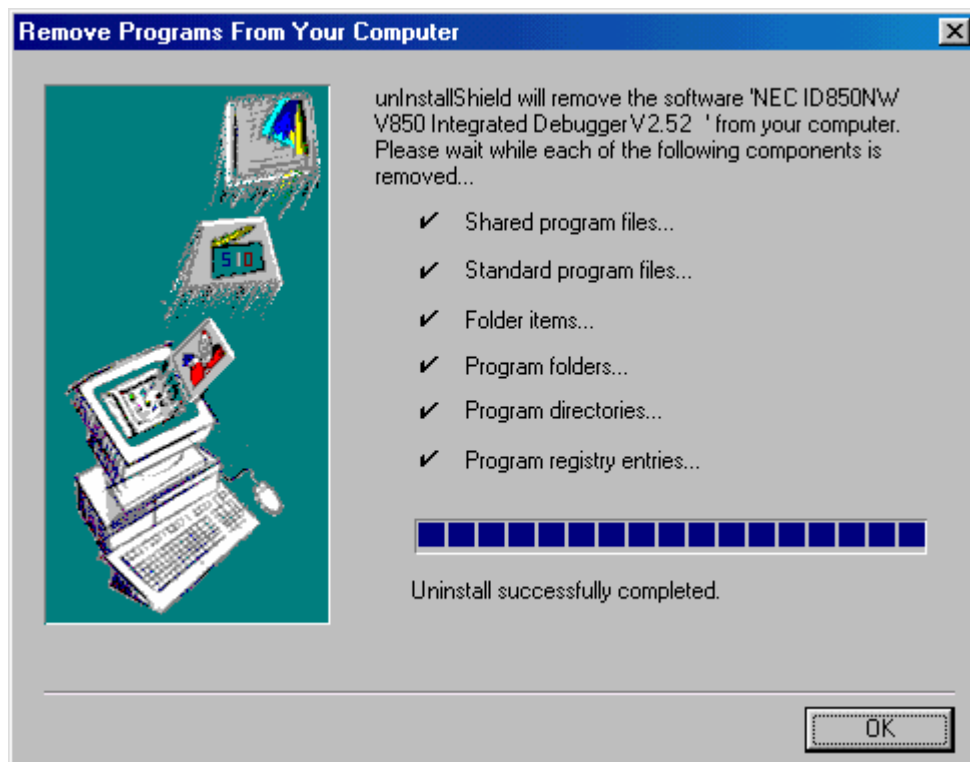


<3> Select "NEC ID850NW V850 Integrated Debugger V2.52" from the list displayed in "Install/Uninstall" and click .

<4> A dialog box for confirming deletion of files will be displayed. Click . Deleting of the files will be started.



<5> When the completion message is displayed, click .



<6> This completes uninstalling ID850NW V2.52.

5. MAJOR REVISIONS FROM ID850NW V1.11 TO V2.52

(1) Change of online help from the Windows help format to the HTML format

(2) Addition of Change Watch dialog box

This dialog box can be opened from [View] → [Change Watch]

(3) Addition of the Console window

This window can be opened from [Browse] → [Console...]

(4) Addition of the Expansion window

This window can be opened from [Browse] → [Others]

(5) Addition of [Ignore Break and Go] and deletion of [Ignore Break Point] from the [Run] menu
<Deletion>


[Run] → [Ignore Break Point]

Function key	F12
--------------	-----

<Addition>

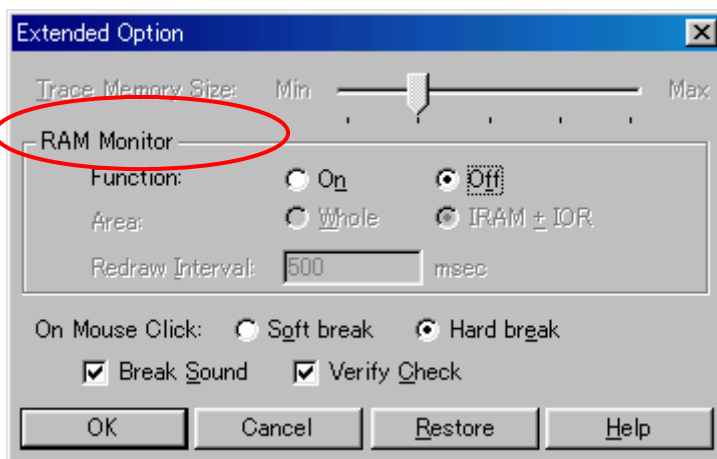
[Run] → [Ignore Break and Go]

[Ignore Break and Go] can also be executed by the button on the tool bar or by a function key.

Button on toolbar	
Function key	Ctrl + F5

(6) Addition of a shortcut key “Ctrl + E” to [Edit Source] in the [Edit] menu

(7) Modification of “real-time RAM monitor setting area” to “RAM monitor setting area” in the Extended Option dialog box.



(8) Addition of the following methods for specifying arrays in the Watch window.

- Variables E.g.) array-name[*i*]
- Octal and hexadecimal numbers E.g.) array-name[010], array-name[0x10]
- Minus symbol E.g.) array-name[-1], array-name[-*i*]

(9) Addition of the following methods for specifying the source file line number.

- prog\$file#no

- (10) Support for entering the selected register name or a variable name in the Name field when the Add Watch dialog box is opened from the following windows.
 - IOR window
 - Register window
 - Local Variable window
- (11) Variable search during emulation in the Source window is now supported.
- (12) Jump to the allocation address (to the Memory window) by variable selection in the Source window is now supported.
- (13) The same variable can now be registered in the Watch window multiple times.
- (14) The environment of the Watch window can now be saved and read.
- (15) Setting and canceling an access break can now be performed in the Watch window by right-clicking the mouse.
- (16) Rewriting the memory during execution in the Memory window has been made much easier.
- (17) The environment of the Peripheral I/O register window can now be saved and read.
- (18) The portion at which a value was changed in the Internal RAM can now be highlighted with a color in the Expansion window.
- (19) When an error message is output, the Help can now be displayed by pressing the [F1] key.
- (20) The displayed radix can now be changed per variable in the Local Variable window and Stack Trace window.
- (21) The Download dialog box is now displayed to indicate the progress of downloading.
- (22) Inclusion of documents can now be selected when ID850NW V2.52 is installed.
- (23) Bit IOR is now always displayed in the IOR window.
- (24) The integrated development environment PM plus included with CA850 V2.50 or later is now supported.
- (25) The Midas Lab emulator is now supported.

6. RESTRICTIONS AND CAUTIONS

This section explains the restrictions and cautions when using ID850NW V2.52.

6.1 Restrictions

The restrictions when using ID850NW V2.52 are described below.

No.	Description
1	Even if an area that is not the target memory is set to I/O protected, an error message is not displayed. [Workaround] None
2	When setting an I/O protected area, if an area is set that is not aligned with the access size, the protect function is not effective. [Workaround] Set an area whose size is aligned with the access size.
3	When the PC is in the prologue of a function during step execution in the mixed display mode in the source text window, the PC marker and current PC line (yellow) are not displayed. [Workaround] None
4	When the Source window is in mixed display mode, if the cursor is moved in the downward direction, it may inadvertently jump. Also, in mixed display mode, the end of the source line cannot be displayed without using the scroll. [Workaround] None
5	The setting of the radix display of the item in the nest of a variable with nest cannot be correctly downloaded from a file on the watch window. [Workaround] None
6	If the data in the Local Variable window is moved by the cursor while emulation is being executed, the display changes to "***". [Workaround] None
7	Non-map area and IOR area cannot be avoided in memory search. [Workaround] Specify a search range avoiding these areas.

No.	Description
8	<p>If the specified copy destination start address is in the internal ROM area, the copy destination start address and copy source start address are aligned in 4-byte units. The copy destination end address is the address at which copying the memory of the size starting from the start address aligned in 4-byte units ends.</p> <p>Example To copy the following memory (access size: 1 byte)</p> <pre>0x100: 01 23 45 67 89 ab cd ef 01 23 45 67 89 ab cd ef 0x110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</pre> <p><Copy source start address> <Copy source end address> <Copy destination start address> 0x102 0x10a 0x112</p> <p>If the addresses are specified as above, the memory contents at the following addresses are copied.</p> <pre>0x100: 01 23 45 67 89 ab cd ef 01 23 45 67 89 ab cd ef 0x110: 01 23 45 67 89 ab cd ef 01 00 00 00 00 00 00 00</pre> <p>The copy source start address is aligned in 4-byte units and is 0x100. The copy source end address is 0x109 because the contents of the specified size, starting from the specified copy source start address, are copied. The copy destination start address is aligned in 4-byte units and is 0x110. [Workaround] None</p>
9	<p>If the cursor is placed in the ASCII display area of the Memory window and then control is shifted to another window, the cursor will inadvertently return to the data display area. [Workaround] None</p>
10	<p>Period (.) must not be used in a symbol name. [Workaround] None</p>
11	<p>When a hardware or software breakpoint has been set for the HALT instruction or an instruction that makes the debugger enter STOP mode, if the program is executed from that instruction, the relevant mode will be entered briefly but soon released, and execution will resume from the next instruction.</p>
12	<p>If step or execution is carried out on the last line^{Note} of the source, an error occurs. Note The last line of main() in C source or the last line of the code in an assembler source. [Workaround] None</p>
13	<p>If one line contains too many instructions (more than 1000 assembler instructions) in source line step, the processing may be aborted even in the middle of step execution. [Workaround] None</p>
14	<p>In the ROM/flash memory relink function supported by the compiler, it is not possible to perform step-in to a function on the flash memory side from the boot side. [Workaround] Cause a break in the function on the flash memory side by setting a breakpoint in that function.</p>
15	<p>Debugging of programs that use program code copied to the internal RAM after being passed through a ROMization processor is not supported. [Workaround] Do not pass programs through a ROMization processor when compiling.</p>

No.	Description
16	<p>Bug that a breakpoint cannot be set in the Source window</p> <p>[Description] A breakpoint cannot be set in the Source window if there is a source file that includes a "-" (hyphen) or "+" (plus) in its name.</p> <p>[Workaround] Modify the source file name so that it does not include a "-" (hyphen) or "+" (plus). Specify a file name using 0 to 9, a to z, A to Z, and _.</p> <p>[Action] Regard this as a usage restriction.</p>
17	<p>Restriction on setting access break in Watch window</p> <p>[Description] A variable other than global (structure/union member, array member, bit field member, etc.) cannot be specified for an access break in the Watch window.</p> <p>[Workaround] None</p>
18	<p>Restriction on setting access break in Source window</p> <p>[Description] A variable other than global (structure/union member, array member, bit field member, etc.) cannot be specified for an access break in the Source window.</p> <p>[Workaround] None</p>

6.2 Cautions

The cautions when using ID850NW V2.52 are described below.

1	The DBPC and DBPSW registers cannot be manipulated in the Register window because of the device specifications. [Workaround] None
2	The access event break is delayed because of the specifications of the DCU (the specified address is passed before execution stops). [Workaround] None
3	Disassembled program cannot be edited when both C and disassembled program are displayed together in the source text window. [Workaround] Edit in the Assemble window.
4	"File" specification is not supported in the Source Search dialog box.
5	Lines cannot be deleted and radices cannot be changed by selecting multiple lines on the Watch window. [Workaround] Select one line at a time.
6	The maximum number of lines that can be displayed in the watch window is 10000. [Workaround] None
7	Lines cannot be scrolled up/down by selecting multiple lines in the Peripheral IOR Select dialog box. [Workaround] Select one line at a time.
8	When a download or project file load is performed, the hardware breakpoints or events may shift in the middle of an instruction. [Workaround] Delete the hardware breakpoints or events and then re-set them.
9	When a software breakpoint has been set in a module file with no debug information and a download or project file load is performed, the software breakpoint will be deleted. [Workaround] Re-set the breakpoint.
10	The device selected at the ID850NW activation cannot be changed after activation. Even if the project file of a device different from the target device is downloaded, the device specified by the project file is not used.
11	The combo box of the type of a file is not in association with the extension of a file in a file-manipulating dialog box (the file is handled in the format selected as "File Type").
12	If one line in a window contains more than 400 characters (ANK characters), the 400th character and those that follow cannot be displayed.
13	The number of lines that can be specified for a C source file (one file) is 65536.
14	The maximum length of the character string that can be searched for in the Source window is 150 (ANK characters).
15	IORs with the same address but different names cannot be distinguished in the disassemble display.
16	If the Assemble window is scrolled up (toward the direction in which the addresses are decremented), illegal mnemonics may be displayed.
17	The maximum length of the character string that can be searched for in the assemble window is 150 (ANK).
18	Arrays with five dimensions or more are not supported.

19	If the PC is not in the body (other than prologue/epilogue) of the current function, the information on local variables in the function cannot be obtained.
20	The maximum length of the character string that can be searched for in the memory window is 150 (ANK characters).
21	The event settings are not cleared even if the load module is re-loaded.
22	A function that has been expanded in-line cannot be stepped in. Because the original function code is created separately from the part that has been expanded in-line, it is possible to set an event there, but the event does not occur because the original code is not executed (whether in-line expansion has been performed can be checked by the mixed display on source window).
23	Assembler instructions enclosed by '#pragma asm' and '#pragma endasm' can be executed on a step-by-step basis in the source mode. The instructions written in a '_asm()' statement cannot be executed.
24	If step execution is performed and '}' at the end of a function is executed, execution line moves to '{' to perform the epilogue processing of the function.
25	If step execution is performed in the source mode, it is judged whether an interrupt is serviced, based on the NP, EP, and DP flags of the PSW register. If the above flags or registers have been changed because nesting is used, return execution and stack display may not be correctly executed.
26	<p>If return execution is performed when a function is recursively called (stack is generated by the recursive processing), the PC moves to the position where the processing of the function called last (leaf function) ended. Even if this symptom appears, the subsequent operation is performed normally.</p> <p>Example If the same function is called five times because of recursive processing</p> <pre> 1:main() 2:fnuc01() Function is called. Execution exits from function. 3:func01() ↓ (*) ↑ 4:func01() 5:func01() 6:func01() </pre> <p>If return execution (that generates the stack) is performed when functions are called in the order of 2, 3, 4, 5, and 6 (*), the address moves the position at which processing 6 is completed. If return execution (which deletes the stack) is performed when execution exits from functions 6, 5, 4, 3, and 2, in that order, execution correctly returns to the main function, from 5 to 4, from 3 to 2, and so on.</p>
27	Step-in cannot be executed on the static function described in the header file in step execution. Step-in or breakpoints cannot be set in an #include statement.

28	<p>When two instructions are executed simultaneously</p> <p>Example 1. When a breakpoint is not set Address A: MOV r1, r2 Address A+2: XOR r1, r2 Two instructions are stepped from Address A, where one instruction should be stepped.</p> <p>Example 2. When a breakpoint is set at address A+2 Address A: MOV r1, r2 [B] Address A+2: XOR r1, r2 A break does not occur at address A+2 if instructions are “executed continuously” from address A.</p> <p>Example 3. When a hardware break is set at address A+2 before execution of instructions Address A: MOV r1, r2 [B] Address A+2: XOR r1, r2 Two instructions are executed instead of one instruction if instructions are “executed in steps” from address A</p>
29	<p>The Search menu of each window is dimmed during program execution.</p>
30	<p>A device file is not recognized if it is installed in more than one folder. Install the device file in one folder.</p>
31	<p>In the conditional statement of an if-else statement, a line that should not be executed may be passed. When such a case occurs, select the mixed display mode in the Source window and confirm that the else statement has not been executed.</p>
32	<p>Big endian is not supported. However, the big- or little-endian mode can be selected only in the Memory window.</p>
33	<p>An NMI and other interrupts are not acknowledged during step execution.</p>
34	<p>Line assemble in the Assemble window does not optimize instructions, as is performed in the CA850 assembler.</p>
35	<p>The PC indicates the address after halt if a break occurs in the HALT status. [Workaround] None</p>
36	<p>Before performing real-time execution, step execution is first performed on an instruction located at the PC. This causes an error in the time measurement result in the timer. In addition, when the program operation is checked using the oscillator or analog logic, the measured timing may differ between when Go is executed at a certain location and Go is executed one instruction before that location.</p>
37	<p>When NMI2 is input in a device with NMI2 (at present, only the SOC device applies), the NMI2 interrupt routine of the user program is executed even if a break is taking effect. [Workaround] Do not input NMI2 during a break period.</p>
38	<p>If an event break or software break occurs when data is overwritten by DMM, the break does not take effect. [Workaround] None</p>
39	<p>Setting/deletion of a hardware breakpoint during program execution is not possible. Setting/deletion of a software breakpoint is performed in the same manner.</p>
40	<p>Single-chip mode 1 is not supported.</p>

41	<p>Restriction on internal ROM start address when using TEG chip</p> <p>Since there is no identification information in single-chip mode 0 or single-chip mode 1, the internal ROM start address is fixed to 0h.</p>				
42	<p>Events may not be detected depending on the event link setting.</p> <p>(1) Restriction on execution events (when using RCU0, RCU1, or RCU2)</p> <p>If the address of an execution event is set in the vicinity of another execution event, the second event cannot be detected normally. This condition does not apply when the event at the second address is executed again using a branch, etc.</p> <p>The second event cannot be detected under the following conditions.</p> <ul style="list-style-type: none"> • The interval between the first and second instruction is within 4 bytes (internal ROM, internal RAM) • The first and second instruction are executed consecutively (target) <p>[Execution example]</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <p>Program example (internal ROM)</p> <pre> 00FE nop 0100 nop ← 1st instruction 0102 nop 0104 nop 0106 nop ← 2nd instruction 0108 nop </pre> </td> <td style="width: 50%; vertical-align: top;"> <p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x106 </pre> </td> </tr> </table> <p style="text-align: center;">↕ Within 4 bytes ↕</p> <p>[Event link example]</p> <pre> Link Name :Lnk0001 Phase1 :Evt0001 Phase2 :Evt0002 </pre> <p>-----</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <p>Program example (target)</p> <pre> 0FFFFFFE 0100000 nop ← 1st instruction 0100002 nop ← 2nd instruction 0100004 nop </pre> </td> <td style="width: 50%; vertical-align: top;"> <p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x102 </pre> </td> </tr> </table> <p>[Event link example]</p> <pre> Link Name :Lnk0001 Phase1 :Evt0001 Phase2 :Evt0002 </pre> <p>(2) Restriction on access events (when using RCU0, RCU1, or RCU2)</p> <p>If the address of an access event is set in the vicinity of another access event, the second event cannot be detected normally. This condition does not apply when the event at the second address is executed again using a branch, etc.</p> <p>The second event cannot be detected under the following conditions.</p> <ul style="list-style-type: none"> • The interval between the first and second instruction is within 4 bytes (accessing the internal ROM or internal RAM) • The interval between the first and second instruction is within 28 bytes (target) 	<p>Program example (internal ROM)</p> <pre> 00FE nop 0100 nop ← 1st instruction 0102 nop 0104 nop 0106 nop ← 2nd instruction 0108 nop </pre>	<p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x106 </pre>	<p>Program example (target)</p> <pre> 0FFFFFFE 0100000 nop ← 1st instruction 0100002 nop ← 2nd instruction 0100004 nop </pre>	<p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x102 </pre>
<p>Program example (internal ROM)</p> <pre> 00FE nop 0100 nop ← 1st instruction 0102 nop 0104 nop 0106 nop ← 2nd instruction 0108 nop </pre>	<p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x106 </pre>				
<p>Program example (target)</p> <pre> 0FFFFFFE 0100000 nop ← 1st instruction 0100002 nop ← 2nd instruction 0100004 nop </pre>	<p>[Event example]</p> <pre> Event Name : Evt0001 Event Name : Evt0002 Event Status: Execution Event Status: Execution Address : 0x100 Address : 0x102 </pre>				

42	<p>[Execution example]</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Program example (internal ROM)</p> <pre> 0100 mov 0x1000, gp 0106 ld.b 0x10[gp], r6 ← 1st instruction 010a nop 010c ld.b 0x12[gp], r7 ← 2nd instruction 0110 nop </pre> <p style="text-align: center;">↑ Within 2 bytes ↓</p> </div> <div style="width: 45%;"> <p>[Event example]</p> <pre> Event Name :Evt0001 Event Name :Evt0002 Event Status :R/W Event Status:R/W Access Size :Byte Access Size :Byte Address :0x1010 Address :0x1012 </pre> <p>[Event link example]</p> <pre> Link Name :Lnk0001 Phase1 :Evt0001 Phase2 :Evt0002 </pre> </div> </div> <p>-----</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Program example (target)</p> <pre> 0100 mov 0x100000, gp 0106 ld.b 0x10[gp], r6 ← 1st instruction 010a nop : : 0110 nop 0120 ld.b 0x12[gp], r7 ← 2nd instruction </pre> <p style="text-align: center;">↑ Within 28 bytes ↓</p> </div> <div style="width: 45%;"> <p>[Event example]</p> <pre> Event Name : Evt0001 Event Name:Evt0002 Event Status : R/W Event Status:R/W Address : 0x1010 Address :0x1012 </pre> <p>[Event link example]</p> <pre> Link Name :Lnk0001 Phase1 :Evt0001 Phase2 :Evt0002 </pre> </div> </div>																																																					
43	<p>Restriction on event detection during misalign access</p> <p>(1) Restriction on write access event No events can be detected.</p> <p>(2) Restriction on read access event Events can be detected by setting the read access event as shown below.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th colspan="2">Operation of Execution Instruction</th> <th colspan="3">Event Condition</th> <th rowspan="2">Detection Status</th> </tr> <tr> <th>Size</th> <th>Address</th> <th>Address</th> <th>Access Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td rowspan="4" style="text-align: center;">Word</td> <td>(Multiple of 4)+0</td> <td>(Multiple of 4)+0</td> <td>Word</td> <td>0x44332211</td> <td>Detected</td> </tr> <tr> <td>(Multiple of 4)+1</td> <td>(Multiple of 4)+1</td> <td>Byte</td> <td>0x22</td> <td>Detected (a)</td> </tr> <tr> <td>(Multiple of 4)+2</td> <td>(Multiple of 4)+2</td> <td>Halfword</td> <td>0x4433</td> <td>Detected (b)</td> </tr> <tr> <td>(Multiple of 4)+3</td> <td>(Multiple of 4)+3</td> <td>Byte</td> <td>0x44</td> <td>Detected</td> </tr> <tr> <td rowspan="4" style="text-align: center;">Half Word</td> <td>(Multiple of 4)+0</td> <td>(Multiple of 4)+0</td> <td>Halfword</td> <td>0x2211</td> <td>Detected</td> </tr> <tr> <td>(Multiple of 4)+1</td> <td>(Multiple of 4)+1</td> <td>Byte</td> <td>0x22</td> <td>Detected (c)</td> </tr> <tr> <td>(Multiple of 4)+2</td> <td>(Multiple of 4)+2</td> <td>Halfword</td> <td>0x4433</td> <td>Detected</td> </tr> <tr> <td>(Multiple of 4)+3</td> <td>(Multiple of 4)+3</td> <td>Byte</td> <td>0x44</td> <td>Detected</td> </tr> </tbody> </table> <p>[Memory status]</p> <pre> +0 +1 +2 +3 3FF8000 11 22 33 44 </pre>	Operation of Execution Instruction		Event Condition			Detection Status	Size	Address	Address	Access Size	Data	Word	(Multiple of 4)+0	(Multiple of 4)+0	Word	0x44332211	Detected	(Multiple of 4)+1	(Multiple of 4)+1	Byte	0x22	Detected (a)	(Multiple of 4)+2	(Multiple of 4)+2	Halfword	0x4433	Detected (b)	(Multiple of 4)+3	(Multiple of 4)+3	Byte	0x44	Detected	Half Word	(Multiple of 4)+0	(Multiple of 4)+0	Halfword	0x2211	Detected	(Multiple of 4)+1	(Multiple of 4)+1	Byte	0x22	Detected (c)	(Multiple of 4)+2	(Multiple of 4)+2	Halfword	0x4433	Detected	(Multiple of 4)+3	(Multiple of 4)+3	Byte	0x44	Detected
Operation of Execution Instruction		Event Condition			Detection Status																																																	
Size	Address	Address	Access Size	Data																																																		
Word	(Multiple of 4)+0	(Multiple of 4)+0	Word	0x44332211	Detected																																																	
	(Multiple of 4)+1	(Multiple of 4)+1	Byte	0x22	Detected (a)																																																	
	(Multiple of 4)+2	(Multiple of 4)+2	Halfword	0x4433	Detected (b)																																																	
	(Multiple of 4)+3	(Multiple of 4)+3	Byte	0x44	Detected																																																	
Half Word	(Multiple of 4)+0	(Multiple of 4)+0	Halfword	0x2211	Detected																																																	
	(Multiple of 4)+1	(Multiple of 4)+1	Byte	0x22	Detected (c)																																																	
	(Multiple of 4)+2	(Multiple of 4)+2	Halfword	0x4433	Detected																																																	
	(Multiple of 4)+3	(Multiple of 4)+3	Byte	0x44	Detected																																																	

<p>43</p>	<p>[Execution example] (a)</p> <pre> 00FFE nop 01000 mov 0x3FF8001, gp 01006 nop 01008 ld.w 0x0[gp], r6 </pre> <p>[Event example]</p> <pre> Event Name :Evt0001 Event Status :R Access Size :Byte Address :0x3FF8001 Data :0x22 </pre> <p>[Execution example] (b)</p> <pre> 00FFE nop 01000 mov 0x3FF8002, gp 01006 nop 01008 ld.w 0x0[gp], r6 </pre> <p>[Event example]</p> <pre> Event Name :Evt0001 Event Status :R Access Size :Half Word Address :0x3FF8002 Data :0x4433 </pre> <p>[Execution example] (c)</p> <pre> 00FFE nop 01000 mov 0x3FF8001, gp 01006 nop 01008 ld.w 0x0[gp], r6 </pre> <p>[Event example]</p> <pre> Event Name :Evt0001 Event Status :R Access Size :Byte Address :0x3FF8001 Data :0x22 </pre>
<p>44</p>	<p>Restrictions on event detection using bit manipulation instruction</p> <p>(1) When the access size of an event is set to Byte and the event is set at an address other than a multiple of 4, if the address is accessed by a bit manipulation instruction, an incorrect event may be detected or no event may be detected. [Workaround] There is no workaround.</p> <p>(2) When the access size of an event is set to Bit and the event is set at an address other than a multiple of 4, if the address is accessed by a bit manipulation instruction, an incorrect event may be detected or no event may be detected. [Workaround] There is no workaround.</p>
<p>45</p>	<p>The following restrictions are applicable when using the AZ850.</p> <ul style="list-style-type: none"> • The hardware trace method cannot be used. Use the software trace method instead. • A function level trace cannot be performed.
<p>46</p>	<p>A pin reset during a break is masked and the CPU or peripheral I/O cannot be reset. In addition, the CPU or peripheral I/O may not be reset normally if a pin reset or internal reset occurs when data is overwritten by DMM or read using the RAM monitor function during user program execution. [Workaround] Do not use the DMM or RAM monitor function when a pin reset or internal reset may occur.</p>

47	No break occurs if an event break or software break occurs as soon as the program is read using the RAM monitor function. [Workaround] There is no workaround.
48	The source cannot be displayed in the Trace window.
49	Trace cannot be searched in the Trace window.
50	The Trace window may not be displayed normally when trace complement is performed. [Workaround] There is no workaround.
51	Devices incorporating RCU1 or RCU2 are not supported.
52	The WAIT signal cannot be masked.
53	When a function to assign external variables to the registers of CA850 V2.60 is used, the variable cannot be browsed or set in the Watch window. [Workaround] Manipulate such variables in the Register window or register the relevant register in the Watch window before manipulating the variables.
54	See the user's manual when using the KIT-NB85E-TP, KIT-V850E/ME2-TP, KIT-V850E/MA3-TP, or KIT-V850E/MA3-IE. Particularly confirm the set values in the peripheral I/O registers when using the external memory. (Example) Confirm the set values in the following peripheral I/O registers when using the KIT-V850E/MA3-IE. <ul style="list-style-type: none"> • Port AL mode control register • Port AH mode control register • Port DL mode control register • Port CS mode control register • Port CT mode control register • Bus cycle type configuration register • Data wait control register • Chip area select control register