Microcontroller Technical Information

ID850QB Integrated Debugger for V850 Microcontrollers Usage Restrictions	Document No.	ZBG-CD-08-0027	1/1
	Date issued	June 18, 2008	
	Issued by	Development Tool Solution Group Multipurpose Microcomputer Systems Division Microcomputer Operations Unit NEC Electronics Corporation	
Related documents ID850QB Ver. 3.40 Operation: U18604EJ1V0UM00 Integrated Debugger ID850QB Ver. 3.41 Operating Precautions: ZUD-CD-07-0208	Notification classification	√	Usage restriction
			Upgrade
			Document modification
			Other notification

1. Affected product

Affected Product	General Name	Outline	Affected Version
ID703000-QB	ID850QB	V850 microcontroller GUI C source debugger	V3.41 or earlier

2. New restrictions

The following restrictions have been added. See the attachment for details.

- No. 40 Restriction on using a load module made by Green Hills Software
- No. 41 Restriction on downloading a code which is filled with 0xFF

3. Workarounds

See the attachment for details.

4. Modification schedule

Restrictions No. 40 and No. 41 will be corrected in ID850QB Ver. 3.50, which is planned for release in mid-June, 2008.

* For the detailed release schedule of modified products, contact an NEC Electronics sales representative.

5. List of restrictions

A list of usage restrictions in the ID850QB, including the revision history and detailed information, is described on the attachment.

List of Usage Restrictions in ID850QB

The following table lists the restrictions on the ID850QB, tracing back from two older versions.
The restriction numbers are not in sequence because the restriction items that were corrected in the previous versions are omitted.

No.	Description	Version		
		3.40	3.41	3.50
1	Restriction on target memory I/O protect setting	Permanent restriction		
2	Restriction on access size when setting I/O protect area	Permanent restriction		
4	Restriction on Source window in mixed display mode	Permanent restriction		
5	Restriction on radix display in Watch window	Permanent restriction		
6	Restriction on data values displayed in the Local Variable window	Permanent restriction		
7	Restriction on memory search area	○	○	○
8	Restriction on ASCII display area of Memory window	Permanent restriction		
10	Restriction when a breakpoint is set to an instruction that sets standby mode	Permanent restriction		
12	Restriction when one line contains too many instructions in source line step	Permanent restriction		
13	Restriction on compiler’s external flash memory/flash memory relink function	Permanent restriction		
14	Restriction on debugging of program codes being passed through a ROMization processor	○	○	○
16	Restriction on setting access breaks in Watch window	Permanent restriction		
22	Restriction on timer overflow when using timer function	Permanent restriction		
23	Restriction on project file when using different emulators	Permanent restriction		
24	Restriction on standby and suspend functions of Windows	Permanent restriction		
28	Trace pickup function	Permanent restriction		
29	Restriction whereby RRM function monitor result display becomes invalid	Permanent restriction		
30	Restriction whereby complement frames displayed in Trace window become invalid	○	○	○
31	Restrictions on using pseudo real-time RAM monitor (pseudo RRM) or DMM	△	△	△
32	Restriction whereby symbol conversion or breakpoint setting becomes invalid	Permanent restriction		
33	Restriction on project files for IECUBE and MINICUBE	Permanent restriction		
34	Restriction on using XO850	○	○	○
35	Restriction on conflict between software break and hardware break	△	△	△
36	Restriction whereby the DWARF2 load module cannot be downloaded	○	○	○
37	Restriction on array variables displayed in Watch window	○	○	○
38	Restriction on a static variable displayed in Watch window	○	○	○
39	Restriction on using flash self-programming library Type04	×	○	○
40	Restriction on using a load module made by Green Hills Software	×	×	○
41	Restriction on downloading a code which is filled with 0xFF	×	×	○

○: Restriction does not apply, △: Restriction partially corrected, ×: Restriction applies, –: Not relevant

Remark “Permanent restriction” means a restriction for which correction is not planned.

ID850QB Restriction Details

No. 1 Restriction on target memory I/O protect setting

[Description]

An error message is not displayed even if an area that is not the target memory is set to I/O protected.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 2 Restriction on access size when setting I/O protect area

[Description]

The protect function does not take effect if an area is set that is not aligned with the access size when setting an I/O protected area.

[Workaround]

Set an area whose size is aligned with the access size.
Regard this issue as a permanent restriction.

No. 4 Restriction on Source window in mixed display mode

[Description]

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]

There is no workaround. Regard this issue as a permanent restriction.

No. 5 Restriction on radix display in Watch window

[Description]

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]

There is no workaround. Regard this issue as a permanent restriction.

No. 6 Restriction on data values displayed in the Local Variable window

[Description]

If the data values in the Local Variable window are moved by the cursor while emulation is being executed, the display changes to “**”.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 7 Restriction on memory search area

[Description]

Non-map area and I/O register area cannot be excluded from the memory search target.

[Workaround]

Exclude these areas when specifying the search range.

This issue has been corrected in V3.40 and later.

No. 8 Restriction on ASCII display area of Memory window

[Description]

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]

There is no workaround. Regard this issue as a permanent restriction.

No. 10 Restriction when a breakpoint is set to an instruction that sets standby mode

[Description]

When a hardware or software breakpoint has been set to an instruction that sets standby mode (such as HALT or STOP), if the program is executed from that instruction, the standby mode will be entered briefly but soon released, and execution will resume from the next instruction.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 12 Restriction when one line contains too many instructions

[Description]

If one line contains too many instructions (more than 1,000 assembler instructions) in source line step, the processing may be aborted even in the middle of step-wise execution.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 13 Restriction on compiler’s external flash memory/flash memory relink function

[Description]

With the external flash memory/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.

There is no workaround. Regard this issue as a permanent restriction.

No. 14 Restriction on debugging of program codes being passed through a ROMization processor

[Description]

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.

This debugging is supported in V3.30 and later.

No. 16 Restriction on setting access breaks in Watch window

[Description]

A variable other than *global* cannot be specified for an access break in the Watch window.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 22 Restriction on timer overflow when using timer function

[Description]

The time measurement counter is cleared to 0 when it overflows. The accumulated time counter is cleared to 0 before the count value becomes 0x1FFFFFFFF or higher, so indication of an overflow is not output. Consequently, the average value is undefined (the undefined value is displayed in red).

[Workaround]

Lower the division ratio in the Extended Option dialog box before performing time measurement.

There is no workaround. Regard this issue as a permanent restriction.

No. 23 Restriction on project file when using different emulators

[Description]

If a project file for IECUBE is opened in MINICUBE or the N-Wire CARD, or a project file for MINICUBE or the N-Wire CARD is opened in IECUBE, some information is lost and the event function, timer function, or the like may not operate normally when the project file is saved.

[Workaround]

Save project files for IECUBE and MINICUBE individually and use the one corresponding to the emulator used.

There is no workaround. Regard this issue as a permanent restriction.

No. 24 Restriction on standby and suspend functions of Windows

[Description]

The standby and suspend functions of Windows and the Switch User function in Windows XP are not supported.

[Workaround]

There is no workaround. Regard this issue as a permanent restriction.

No. 28 Trace pickup function

[Description]

The trace pickup function has not been supported. There is no plan for future support.

No. 29 Restriction whereby RRM function monitor result display becomes invalid

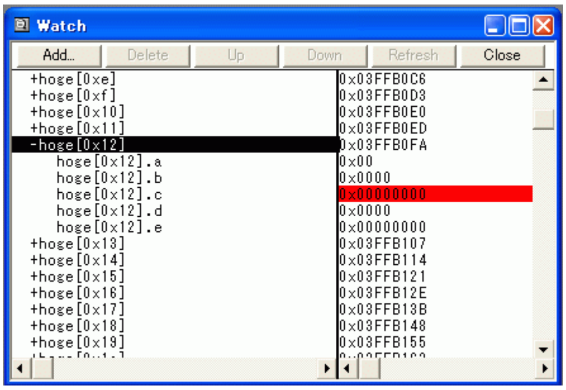
[Description]

With the RRM function, if the block areas to be monitored are set in succession and the access data extends over the boundary of the blocks, the invalid values are displayed in the Memory or Watch window.

[Workaround]

There is no workaround.

The specification has been modified so that variables that extend over the boundary of the blocks are highlighted in red in V3.11 and later.



```
+hoge[0xe]            0x03FFB0C6
+hoge[0xf]            0x03FFB0D3
+hoge[0x10]           0x03FFB0E0
+hoge[0x11]           0x03FFB0ED
-hoge[0x12]           0x03FFB0FA
   hoge[0x12].a       0x00
   hoge[0x12].b       0x0000
   hoge[0x12].c       0x00000000
   hoge[0x12].d       0x0000
   hoge[0x12].e       0x00000000
+hoge[0x13]           0x03FFB107
+hoge[0x14]           0x03FFB114
+hoge[0x15]           0x03FFB121
+hoge[0x16]           0x03FFB12E
+hoge[0x17]           0x03FFB13B
+hoge[0x18]           0x03FFB148
+hoge[0x19]           0x03FFB155
```

No. 30 Restriction whereby complement frames displayed in Trace window become invalid

[Description]

If the Trace area settings in the Extended Option dialog box satisfy both of the following conditions (1) and (2), complement frames between a frame in which a part of data is missed[Note] and a frame with the next branch instruction (BRM1 status) are displayed abnormally in the Trace window.

<Setting conditions>

(1) Either of the following settings is selected from the Trace Data drop-down list

- ALL PC + Access Data
- Branch PC + Access Data + Access PC

(2) Complement Data check box is selected

Note A frame in which a part of data is missed is marked with <Lost Data> in the Trace window.

[Workaround]

There is no workaround. This issue has been corrected in V3.20 and later.

No. 31 Restrictions on using pseudo real-time RAM monitor (pseudo RRM) or DMM

[Description]

There are following restrictions when using pseudo real-time RAM monitor (pseudo RRM)[Note] or DMM.

(1) If the STOP or IDLE mode is entered during program running while the IOR window or Memory window is open, error A0C03 is output and the speed of window update processing is degraded. This restriction does not apply even if the Memory window is open when the target device is a core other than the V850E2 core.

(2) The STOP, IDLE, or HALT mode is released if the IOR windows or Memory window is open.

(3) If a conflict occurs between the timing of a software break and the timing at which the CPU is temporarily stopped by pseudo RRM, the instruction to which a software break is set will not be executed. If the software break is set to a 4- or 6-byte instruction, the instruction different from the original one will be executed.

Note Pseudo RRM is a function used to read the values of I/O registers or the memory by generating a temporary break for the CPU during program running. This restriction applies when an item other than “Off” is selected for “Break When Readout” in the Extended Option dialog box.

[Workaround]

There is no workaround. This issue will be corrected (specification will be modified) in V3.20, as shown below.

(1) If the STOP, IDLE or HALT mode is entered during program running, the data in the IOR window or Memory window will be displayed as “*”.

(2) Modification will be made so as to minimize the possibility whereby the STOP, IDLE or HALT mode that is entered during program running while the IOR window or Memory window is open is released.

(3) Pseudo RRM and DMM can no longer be used with software breaks. An error will be output if an attempt is made to use them together.

No. 32 Restriction whereby symbol conversion or breakpoint setting becomes invalid

[Description]

If function names or variable names are distinguished with a prefixed underscore, the debugger may recognize them incorrectly. As a result, symbol conversion or breakpoint setting may become invalid.

Example When two function names _reset and __reset are used

[Workaround]

Do not distinguish similar function names or variable names only with a prefixed underscore.

There is no workaround. Regard this issue as a permanent restriction.

No. 33 Restriction on project files for IECUBE and MINICUBE

[Description]

A project file for IECUBE cannot be used with MINICUBE2 as is, because communication interface information is lacking.

[Workaround]

Save a project file using MINICUBE2, and then overwrite and save the project using IECUBE.

There is no workaround. Regard this issue as a permanent restriction.

No. 34 Restriction on using XO850

[Description]

Specifications for the RRM and DMM functions have been changed in ID850QB V3.20 so as not to read or write the memory contents, the operation of automatic verify system XO850 may become invalid when the CPU is in HALT mode.

[Workaround]

There is no workaround. This issue has been corrected in V3.21 and later.

No. 35 Restriction on conflict between software break and hardware break

[Description]

When execution is resumed after a break under the following cases, an unexpected instruction may be executed.

- When a software break and hardware break occur at the same time
- When a hardware break occurs at the address two addresses after a software break has occurred

[Workaround]

Do not set a software break near a fetch system hardware break. If an access system hardware break is set, do not set software breaks.

Specifications of ID850QB V3.30 and later have been corrected so that a warning is output upon execution of the unexpected instruction.

No. 36 Restriction whereby the DWARF2 load module cannot be downloaded

[Description]

The debugger is forcibly terminated if an attempt is made to download a load module for a program created with a GHS C compiler and includes *void* function pointer variables.

[Workaround]

Change the void function pointer variables to *int* function pointer variables.

This issue has been corrected in V3.21 and later.

No. 37 Restriction on array variables displayed in Watch window

[Description]

If an underscore is prefixed to the name of an array variable, the array variables expanded in the Watch window are displayed as “?”.

[Workaround]

Do not prefix underscore to the name of an array variable. Underscores automatically prefixed by the compiler will cause no problem.

This issue has been corrected in V3.30 and later.

No. 38 Restriction on a static variable displayed in Watch window

[Description]

If a static variable registered in the Watch window goes out of the scope, the variable is recognized as a label and an invalid result may be displayed.

[Workaround]

Register static variables to the Watch window in the format “*file-name#variable- name#*”.

This issue has been corrected in V3.30 and later.

No. 39 Restriction on using flash self-programming library Type04

[Description]

During flash self-programming with IECUBE, when any of the functions of flash self-programming library Type04 listed below is called under the relevant condition and then the *FlashStatusCheck* function is called, flash macro service processing does not end and the *FlashStatusCheck* function is not executed correctly.

Affected functions:

FlashBlockBlankCheck, FlashBlockErase, FlashBlockIVerify

Condition:

Block 128 or later is specified for the start block number or end block number.

(Except for the case where the block subject to specification consists of only one block.)

[Workaround]

This issue has been corrected in V3.41 and later.

No. 40 Restriction on using a load module made by Green Hills Software

[Description]

When using a load module made by Green Hills Software, stepwise execution in C source level may not be performed normally.

[Workaround]

There is no workaround. This issue has been corrected in V3.50 and later.

No. 41 Restriction on downloading a code which is filled with 0xFF

[Description]

If a code to be downloaded is filled with 0xFF for more than one block (in block units in flash memory), the subsequent download process may be skipped (when using MINICUBE or N-Wire CARD).

[Workaround]

Erase the flash memory before executing downloading. This issue has been corrected in V3.50 and later.