

Microcontroller Technical Information

ID78K0R-QB 78K0R Integrated Debugger	Document No.	ZBG-CD-10-0021	1/1
	Date issued	May 12, 2010	
Usage Restrictions	Issued by	MCU Tool Product Marketing Department MCU Software Division MCU Business Unit Renesas Electronics Corporation	
	Notification classification	√	Usage restriction
		Upgrade	
		Document modification	
		Other notification	
Related documents: ID78K0R-QB Ver. 3.60 Integrated Debugger Operation User's Manual: U19613EJ1V0UM00 (1st edition) 78K0R Integrated Debugger ID78K0R-QB V3.60 Operating Precautions: ZUD-CD-09-0156			

1. Affected product

Product	Outline	Affected Version
ID78K0R-QB	78K0R integrated debugger	V3.60 or earlier

2. New restriction

New restriction No. 27 has been added. See the attachment for details.

3. Workarounds

See the attachment for details.

4. Modification schedule

Products in which restriction No. 27 is corrected will be posted on the Development Tools Download webpage on May 13, 2010 (ID78K0R-QB V3.61).

* Note that this schedule is subject to change without notice. For the detailed release schedule of modified products, contact a Renesas Electronics sales representative.

5. List of restrictions

See the attachment.

6. Document revision history

78K0R Integrated Debugger ID78K0R-QB - Usage Restrictions

Document Number	Issued on	Description
ZBG-CD-07-0010	January 18, 2007	Addition of restrictions (No. 8 to No. 19) Addition of specifications (No. 1 to No. 3)
ZBG-CD-07-0054	August 27, 2007	Addition of restrictions (No. 20 to No. 22) Addition of specification (No. 4)
ZBG-CD-08-0025	June 25, 2008	Addition of restrictions (No. 23 and No. 24) Addition of specifications (No. 5 to No. 10)
ZBG-CD-09-0044	July 31, 2009	Addition of restrictions (No. 25 and No. 26)
ZBG-CD-10-0021	May 12, 2010	Addition of restriction (No. 27)

1. Usage Restrictions

This section describes the restrictions on using the ID78K0R-QB.

The item numbers are the same as those described in *78K0R Integrated Debugger ID78K0R-QB V3.60 Operating Precautions* (ZUD-CD-09-0156).

1.1 List of usage restrictions

No.	Usage Restrictions	Affected Product	Version				
			V3.20	V3.30	V3.40	V3.50	V3.60
1	Source path cannot be added even if added using PM+	Common	×	×	×	×	×
2	Restriction on redraw	Common	×	×	○	○	○
3	Symbols with a <code>func#var</code> format cannot be converted to addresses	Common	Described in the user's manual				
4	Searching fails in the Assemble window	Common	×	×	○	○	○
5	The Refresh button appears dimmed in the Watch window	Common	×	○	○	○	○
6	Searching extending over a search-prohibited area is impossible	Common	×	×	○	○	○
7	Invalid values are displayed in the Memory window	Common	×	○	○	○	○
8	Restriction on hardware break before execution	IECUBE	×	○	○	○	○
9	Restrictions on coverage measurement	IECUBE	×	○	○	○	○
10	Operation becomes invalid if a specific stack area is used	MINICUBE2	×	○	○	○	○
11	A break does not occur if an access breakpoint is specified for an odd address in byte units	MINICUBE2	×	○	○	○	○
12	Restrictions on software break points	MINICUBE2	×	○	○	○	○
13	Restrictions on specifying access breakpoints	Common	×	×	○	○	○
14	Restrictions on adding SFRs in Add I/O Port dialog box	Common	×	×	○	○	○
15	Restriction on rewriting PSW bits in DMM dialog box	IECUBE	×	×	○	○	○
16	Restriction on changing font size	Common	×	×	○	○	○
17	Restriction on moving using Address Move dialog box	Common	×	×	○	○	○
18	Restriction on path name length	Common	×	×	○	○	○
19	Caution on <code>Return Out</code> in Source window	Common	×	×	○	○	○
20	Restriction on BCD correction result register	Common	×	×	○	○	○
21	Restriction on Code Coverage window	IECUBE	×	×	○	○	○
22	Restriction on accessing the external bus	Common	×	×	×	×	×
23	Restriction whereby a break occurs at a specific SP value	MINICUBE2	×	×	×	○	○
24	Restriction on reset during RRM or DMM	MINICUBE2	×	×	×	×	×
25	Restriction whereby an external expansion area value was not displayed in the Memory window	Common	○	○	○	×	○
26	Restriction whereby a pointer was displayed incorrectly in the Watch window	Common	×	×	×	×	○
27	Restriction whereby a step execution result of the specific instruction is incorrect	MINICUBE2	×	×	×	×	×

Common: Items common to IECUBE and MINICUBE2
 ○: Corrected or planned to be corrected, ×: Not corrected

1.2 Restriction details

No. 1 Source path cannot be added even if added using PM+ <Common>

Description:

When a project is saved and then a new source path is added using PM+, the new path is not added to the ID78K0R-QB.

Workaround:

On the **Option** menu, select **Debugger Options** to open the **Debugger Option** dialog box, and then add the source path manually.

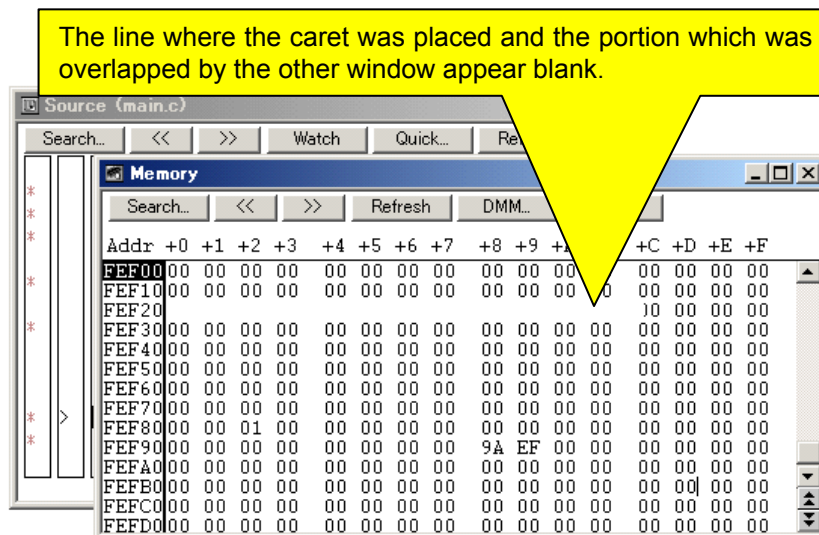
Correction:

Under study.

No. 2 Restriction on redraw <Common>

Description:

When the **Memory** window is displayed in front of another window, if the window behind the **Memory** window is clicked to the front and then the **Memory** window is clicked to the front again, the line where the caret was placed and the portion which was overlapped by the other window appear blank in the **Memory** window.



Workaround:

Click the **Refresh** button or click on the blanked line to restore the original display.

Correction:

This issue has been corrected in V3.40.

No. 3 Symbols with a func#var format cannot be converted to addresses <Common>

Description:

This item is related to the specification, so it is described in *CHANGES IN USER'S MANUAL* in *ID78K0R-QB Operating Precautions*.

No. 4 Searching fails in the **Assemble window <Common>****Description:**

Searching might not be performed for the last 0x400 space of the range subject to searching in the **Assemble** window. For example, if the range 0x1000 to 0x1FFF is searched in the down direction and a character string that matches the search target exists in the range 0x1C00 to 0x1FFF, the search might fail. In the same manner, if the range 0x2000 to 0x2FFF is searched in the up direction, the search in the range 0x2000 to 0x23FF might fail.

Workaround:

Add a 0x400 space to the search target range. (To search a character string in the range 0x1000 to 0x1FFF in the down direction, for example, specify 0x1000 to 0x23FF.)

Correction:

This issue has been corrected in V3.40.

An additional 0x400 space (1 KB) is now searched. Consequently, if target data exists in the search range, the search definitely returns a hit. As a result of this change, however, the search is performed on a wider range than specified, so be sure to check if the hit data actually exists in the specified address range.

No. 5 The **Refresh button appears dimmed in the **Watch** window <Common>****Description:**

If an item at the bottom in the **Watch** window is deleted, the **Refresh** button appears dimmed.

Workaround:

Close the **Watch** window and then open it again.

Correction:

This issue has been corrected in V3.30.

No. 6 Searching extending over a search-prohibited area is impossible <Common>**Description:**

If an area for which searching is prohibited (non-map area, SFR area, or I/O protect area) is included in the search target specified in the **Memory Search** dialog box and **Assemble Search** dialog box, searching is not performed for any area beyond the area for which searching is prohibited.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.40.

No. 7 Invalid values are displayed in the **Memory window <Common>****Description:**

Values in the general-purpose register area and the results of access monitoring are not displayed correctly in the **Memory** window.

Workaround:

Check these values in the **Register** window.

Correction:

This issue has been corrected in V3.30.

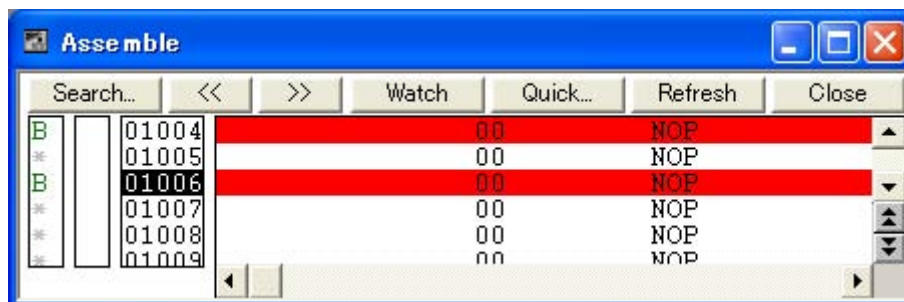
No. 8 Restriction on hardware break before execution <IECUBE>

Description:

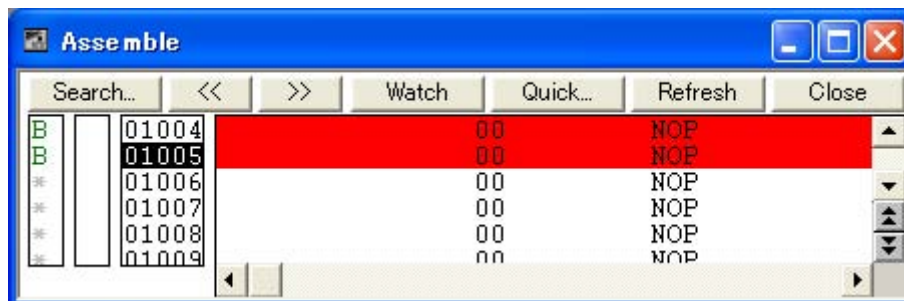
If multiple “break before execution” points are specified within a 4-byte boundary area^{Note}, the breaks do not occur as specified. The following two cases are assumed:

Note A 4-byte boundary area means the addresses ranging from $4n$ to $4n + 3$ (n is an integer).

Example 1: If a “break before execution” point is specified for addresses 1004 and 1006, in that order, the break does not occur at address 1006.



Example 2: If a “break before execution” point is specified for addresses 1004 and 1005, in that order, and the breakpoint at address 1005 is removed, the breakpoint at address 1005 remains valid.



Workaround:

There is no workaround.

Clear all of the “break before execution” points within a 4-byte boundary area and specify just one “break before execution” point, or use software breakpoints instead.

Correction:

This issue has been corrected in V3.30.

No. 9 Restrictions on coverage measurement <IECUBE>

Description:

The following bugs (a) to (c) concerning coverage measurement occur:

- Coverage measurement (C0 coverage) measures not only ROM fetch but also ROM read.
- When the read access status is displayed as a result of access monitoring in the **Memory** window, the window displays not only the ROM read status but also the ROM fetch status.
- The general-purpose register value displayed in the **Memory** window might become invalid

after accessing the RAM area to which the general-purpose register is assigned. In addition, the **Register** window display might be invalid during program execution.

Workaround:

- (a) There is no workaround.
- (b) There is no workaround.
- (c) There is no workaround.

Correction:

Issues (a) to (c) have been corrected in V3.30.

After correction of (a), only ROM fetch is measured.

After correction of (b), only ROM read is displayed.

The modification affects the IECUBE (QB-78K0RKX3) and IECUBE self-diagnostic tool IEQBUTL, so be sure use them in the following combination.

QB-78K0RKX3 Control Code	ID78K0R-QB	IEQBUTL
E, F, G, H	V3.20	V2.11
J and later	V3.30 and later	V2.13 and later

No. 10 Operation becomes invalid if a specific stack area is used <MINICUBE2>

Description:

If the area from 0xFB00 to 0xFCFF is used as a stack area, the stacked data might become invalid.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.30.

No. 11 A break does not occur if an access breakpoint is specified for an odd address in byte units <MINICUBE2>

Description:

If an access breakpoint is specified for an odd address in byte units, the break does not occur.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.30.

No. 12 Restrictions on software break points <MINICUBE2>

Description:

- (1) If a software breakpoint is specified for RAM, the first byte of the instruction becomes 0xff, and the rest of the bytes become 0x00.
- (2) If a software breakpoint is specified for the last address of a 64 KB boundary area (such as 0xffff or 0x1ffff) and execution stops at the specified point, the PC value indicates the last address of the next 64 KB boundary area (0x1ffff or 0x2ffff, in the above case).

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.30.

No. 13 Restrictions on specifying access breakpoints <Common>

Description:

- (1) In the **Watch** window, access breakpoints cannot be specified for variables other than global variables.
- (2) In the **Source** window, access breakpoints cannot be specified for variables other than global variables

Workaround:

To specify an access breakpoint to a static variable in a function, use the **Event** dialog box while the current PC line is in that function. To specify an access breakpoint to a static variable in a file, use the **Event** dialog box while the current PC line is in that file.

Correction:

This issue has been corrected in V3.40.

Specifying access breakpoints in the **Watch** and **Source** windows is now available for static variables in a function or a file.

Specifying access breakpoints in the **Watch** window is now available only for static variables registered in the **Add Watch** dialog box in the following formats:

- Static variables in a function: *file-name#function-name#variable-name*
- Static variables in a file: *file-name#variable-name*

No. 14 Restrictions on adding SFRs to **Add I/O Port** dialog box <Common>

Description:

The following restrictions apply when adding SFRs to the **Add I/O Port** dialog box (dialog box used when an SFR name is redefined with another name in a C source).

- (1) If a R/W-attribute SFR is added as a W-attribute register with another name, the SFR is added to the **Add Watch** window and its value is changed in the **Watch** window, values displayed in the **SFR** window are not updated.
- (2) Adding SFRs for the 2nd SFR area is not available (if attempted, an error occurs.)
- (3) If an SFR that satisfies the conditions below is added, the SFR is added but an error occurs when an attempt is made to display the added SFR by using the `MOVE` command on the shortcut menu in the **SFR** window (moving fails).
 - The name of the SFR to be added is used for a general-purpose register
 - The address to be added is the same as that of an SFR area (different from the address of the general-purpose register).

Workaround:

- (1) When changing values in the **Watch** window, input the value multiple times, or change the values in the **SFR** window.
- (2) There is no workaround.
- (3) There is no workaround.

Correction:

This issue has been corrected in V3.40.

After correction of (3), SFRs having the same names as general-purpose registers can no longer

be added.

No. 15 Restriction on rewriting PSW bits in **DMM** dialog box <IEUCBE>

Description:

When changing register values in the **DMM** dialog box, the error message `A9004: Too large register size.` is displayed if a PSW bit (IE, Z, RBS1, AC, RBS0, ISP1, ISP0, or CY) is specified in the **Register Name** area, but this error message is incorrect.

Workaround:

To change values of registers such as the PSW in the **DMM** dialog box, specify the actual register, not its bits. To change individual bits, use the **Register** window after the program has finished executing.

Correction:

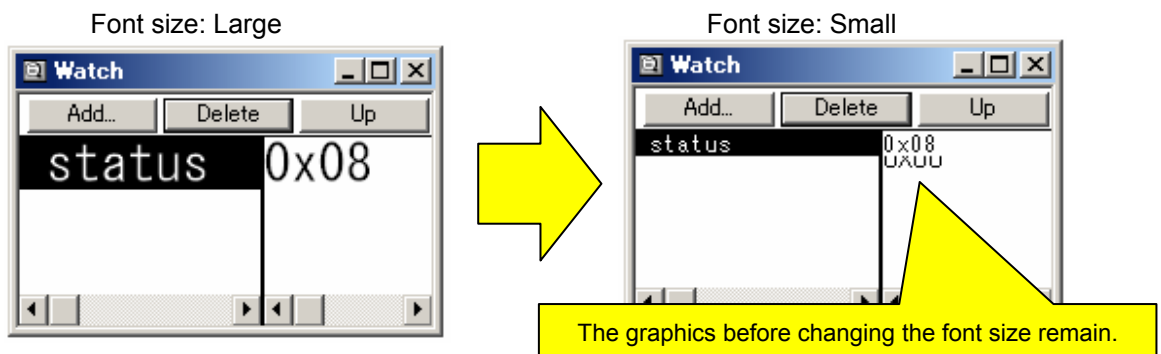
This issue has been corrected in V3.40. Specifying the PSW bits is now available.

No. 16 Restriction on changing font size <Common>

Description:

The font size in the **Watch** window and **Local Variable** window can be specified in the **Font** area in the **Debugger Option** dialog box, which is opened by clicking the **Option** menu and then **Debugger Option...**, but if the font size is reduced, the graphics before changing the font size remain in the window.

Example of Watch window:



Workaround:

Minimize the window and then restore the window size, or close the window and then open it again.

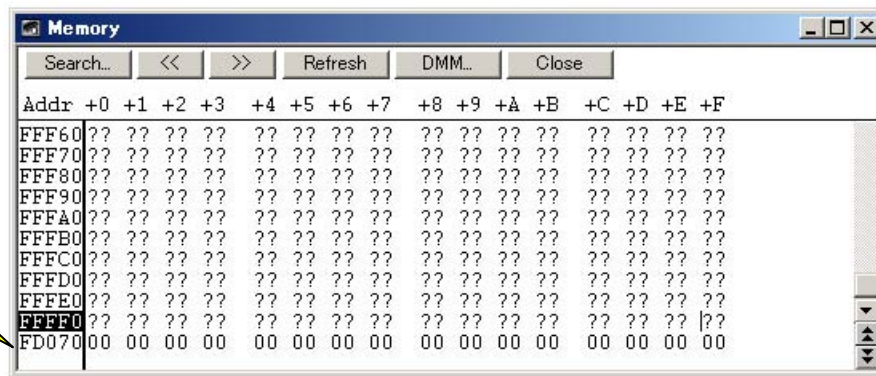
Correction:

This issue has been corrected in V3.40.

No. 17 Restriction on moving using **Address Move** dialog box <Common>

Description:

If the **Address Move** dialog box is opened by selecting the `Move` command on the shortcut menu in the **Memory** window, an extra line might be displayed when the end address (0xffff) is input.



An extra line might be displayed.

Workaround:

Do not move to address 0xffff. (Because this address is in the SFR area, displaying this area in the **Memory** window is meaningless.)

Correction:

This issue has been corrected in V3.40.

No. 18 Restriction on path name length <Common>

Description:

Windows supports path names (including file name) consisting of up to 259 single-byte characters, but the ID78K0R-QB cannot use 256 or more characters for path names. If 256 or more characters are used for path names, the operation becomes invalid. (For example, if the path for a load module is too long, the module cannot be downloaded, or if the path for a source file is too long, the source file cannot be opened.)

Workaround:

Specify path names using up to 255 characters, including the file name.

Correction:

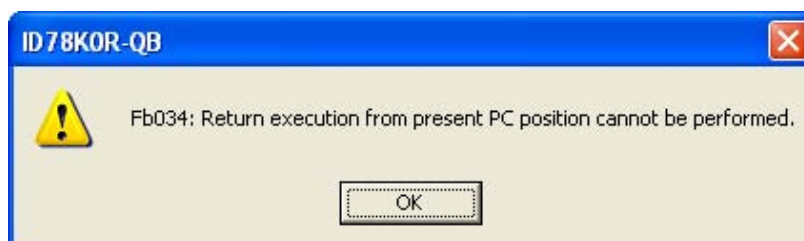
This issue has been corrected in V3.40.

The number of characters that can be used for path names is now extended to 259 characters (the maximum value supported by Windows).

No. 19 Caution on Return Out in **Source** window <Common>

Description:

`Return Out` is used to execute a program until execution returns to the address from which the function was called, and uses a stack to predict the call source. Due to this, execution cannot return from locations where the association between the stack and the stack pointer is shifted (location where the stack is currently being manipulated, such as before execution of a function prologue (`{`) or execution of an epilogue (`}`) is in progress, or where the program counter points to the middle, not the beginning, of a C source). If `Return Out` is attempted at a location where `Return Out` cannot be performed, the following error message is output:



Workaround:

When the above error message is output, step through the code and then execute `Return Out`. (After stepping through the code, the difference between the stack address and stack pointer is resolved.)

Correction:

The error message has been corrected in V3.40.

The above workaround is now displayed as a message.

Fb035: Return execution from present PC position cannot be performed. Perform step-wise execution once and then `Return Out`.

No. 20 Restriction on BCD correction result register <Common>**Description:**

When stepping through the code of an addition or subtraction instruction, the value in the BCD correction result register (BCDADJ) becomes invalid. In addition, the value in register A is not corrected in decimal units when using MINICUBE2.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.40.

No. 21 Restriction on Code Coverage window <IECUBE>**Description:**

The value **0.0** is displayed in the **Coverage(%)** column on the **Interrupt** tab in the **Code Coverage** window, regardless of whether an interrupt has occurred or not. In addition, **----** is displayed in the **Status** column, regardless of whether interrupt servicing is specified to be used or not.

Name	Type	Status	Address	Size	Fetch	Coverage(%)
INTLVI	maskable	----	0x6	2	0	0.0
INTPO	maskable	----	0x8	2	0	0.0
INTP1	maskable	----	0xa	2	0	0.0

Display in **Status** column becomes ----.

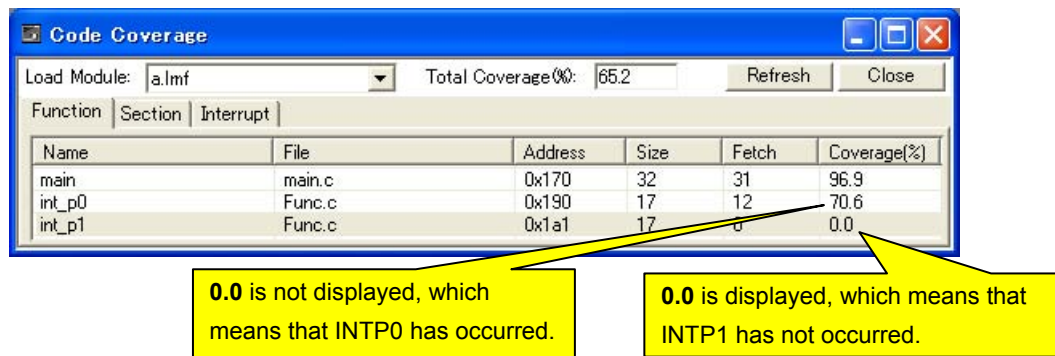
Display in **Coverage(%)** column becomes **0.0**.

Workaround:

The **Interrupt** tab is used to check whether an interrupt has occurred, but this can also be checked by referring to the **Function** tab to check the coverage of the function corresponding to the interrupt.

- When **0.0** is displayed in **Coverage(%)**: No interrupt has occurred.
- When a value other than **0.0** is displayed in **Coverage(%)**: An interrupt has occurred.

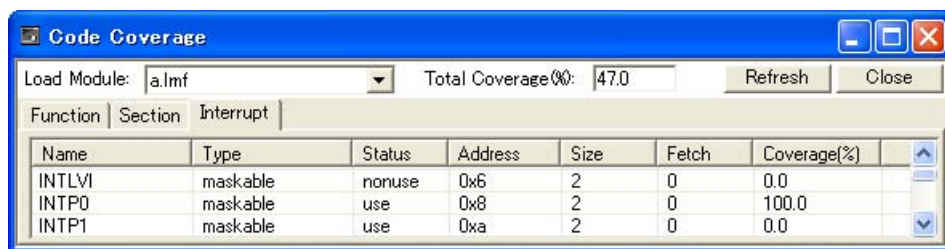
Example: When interrupt INTPO is defined as function `int_p0` and INTP1 as function `int_p1`



Correction:

This issue has been corrected in V3.40. The display after correction is as follows:

- **Status:** If the interrupt function is defined, **use** is displayed; otherwise, **nonuse** is displayed.
- **Coverage(%):** If an interrupt has occurred, **100.0** is displayed; otherwise, **0.0** is displayed.



No. 22 Restriction on accessing the external bus <Common>

Description:

When the external memory is written during a break, access is always performed in 8-bit units, regardless of the access size specified in the **Memory Mapping** area in the **Configuration** dialog box.

Workaround:

There is no workaround.

Correction:

Under study.

No. 23 Restriction whereby a break occurs at a specific SP value <MINICUBE2>

Description:

- (1) The program might hang up when a break occurs if the SP value is in the range of 0xffb02 to 0xffba6 while any of the **Assemble** window, the **Memory** window, or the **SFR** window is open, or is opened.
- (2) The program might hang up when a break occurs if the SP value is in the range of 0xffc02 to 0xffc04, if a debugger operation such as file downloading or software break setting, which rewrites the internal ROM data, is performed.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.50.

No. 24 Restriction on reset during RRM or DMM <MINICUBE2>**Description:**

If a reset is applied during RRM or DMM, the error message A0109: Can not communicate with ICE. Please terminate the debugger and check the power of ICE or the connection of cable then restart the debugger. might be displayed.

Workaround:

Close the error message and wait for approximately 5 seconds.

Correction:

Under study.

No. 25 Restriction whereby an external expansion area value was not displayed in the **Memory window <Common>****Description:**

When using the external bus interface, if the external expansion area is viewed in the **Memory** window, all of the values become 0. This problem is related only to the display of the ID78K0R-QB, and program execution is not affected.

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.60.

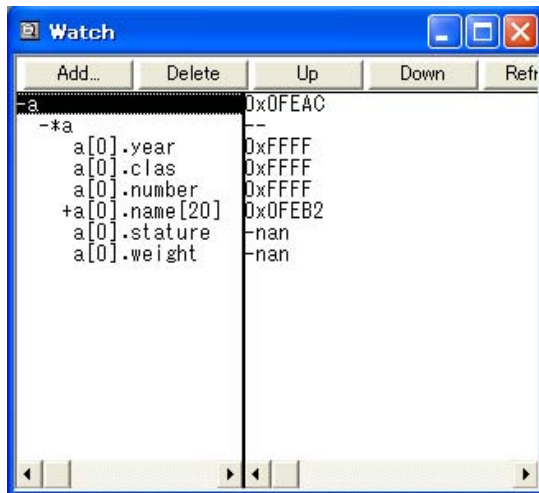
No. 26 Restriction whereby a pointer was displayed incorrectly in the **Watch window <Common>****Description:**

The two incorrect display problems below occur if a pointer used as a function parameter or an automatic pointer is registered to the **Watch** window. These problems are related only to the display of the ID78K0R-QB, and program execution is not affected.

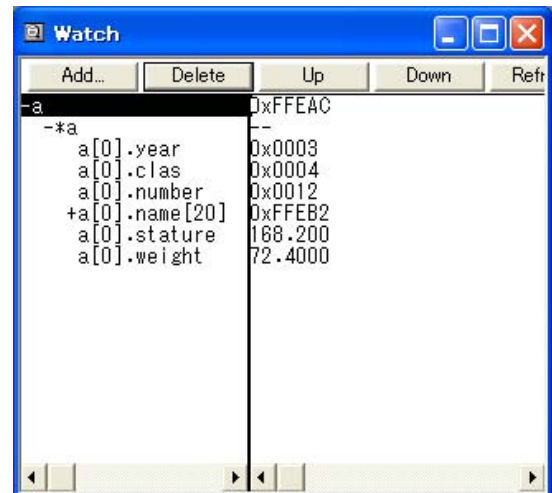
Problem (1) The data for an address other than the one specified for the pointer is displayed.

Example when the address 0x0FFEAC is assigned to the pointer a

When the pointer *a is registered to the **Watch** window, the data for 0x0FFEAC is supposed to be displayed, but the fifth least significant digit becomes 0 and the data for 0x00FEAC is displayed instead.



Incorrect display example
(ID78K0R-QB V3.50 or earlier)

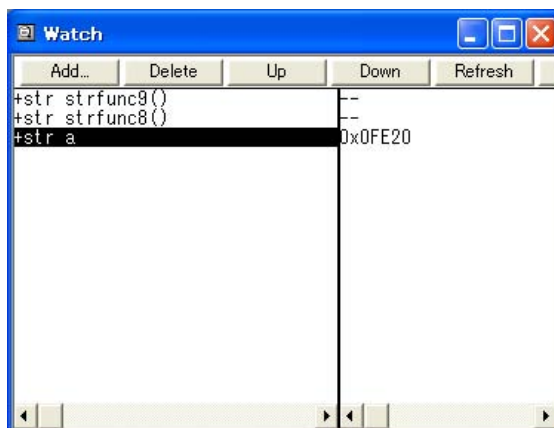


Correct display example

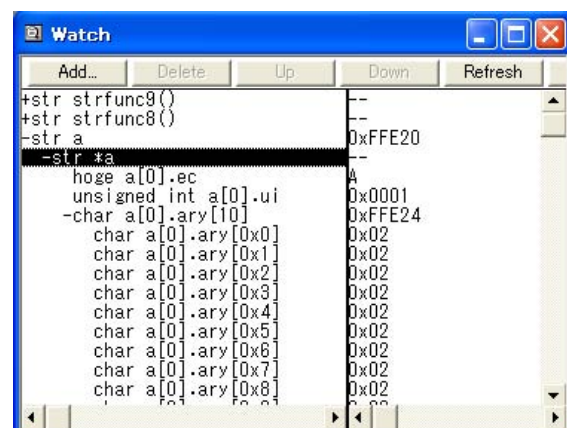
Problem (2) The pointer cannot be expanded for display.

Example for the pointer *a*, which points to a structure that has the members *ec*, *ui*, and *ary*

When the structure pointer *a* is registered to the **Watch** window, the members (*ec*, *ui*, and *ary*) cannot be expanded for display.



Incorrect display example
(ID78K0R-QB V3.50 or earlier)



Correct display example

Workaround:

There is no workaround.

Correction:

This issue has been corrected in V3.60.

No. 27 Restriction whereby a step execution result of the specific instruction is incorrect <MINICUBE2>

Description:

When an assembler instruction that has "general-purpose register + offset"^{note} as the access addressing operand is executed, and this access addressing operand exceeds 10000H, the result of step execution is incorrect. This restriction is only on a step execution of MINICUBE2.

Note In case the instruction which includes below as the operand is executed.

[HL+byte], [DE+byte], [SP+byte], word [B], word [C], word [BC], [HL+B], [HL+C] (not including ES:[HL+byte], ES:[DE+byte], ES:word[B], ES:word[C], ES:word[BC], ES:[HL+B], ES:[HL+C]).

Workaround:

There is no workaround.

Correction:

This issue will be corrected in the next version.