

# M16C R8C FoUSB/UART Debugger V.1.03.00

## Release Notes

This document describes the notes of this debugger, and please read before you start to use this debugger.

And also, please refer to the “High-performance Embedded Workshop Release Notes” about the notes of High-performance Embedded Workshop IDE.

### Contents

1	Notes .....	2
1.1	Line Assembly .....	2
1.2	RAM Monitor .....	2
1.3	Memory .....	2
1.4	Script .....	2
1.5	Real-time OS debugging functions .....	2
1.6	Macro recording function .....	2
1.7	Test facility function .....	2
1.8	Using cast operators for the member variable .....	2
1.9	Download module dialog box .....	2
1.10	The option “Always treat variables of enumerator type with unknown size as 1 byte” .....	2
1.11	Debugging for assembler macros .....	3
1.12	Debugging for inline functions .....	3
1.13	Automatic target connection on changing the session .....	3
1.14	Run program option .....	3
1.15	Selection of the object format for download module .....	3
1.16	Notes on Debugging (M16C R8C FoUSB/UART Debugger) .....	3
1.16.1	Variables assigned to registers .....	3
1.16.2	Limitations on Stop and Wait Modes .....	3
1.16.3	Real-time performance of Watchdog Timer .....	3
1.16.4	Realtime Capability of the User Program .....	3
1.16.5	On debugging of MCUs of the R8C/14, 15, 16, 17 group .....	4
1.16.6	The "Go to cursor" function when using FoUSB .....	4
1.16.7	The Pass counts for software break points .....	4
1.16.8	Communication Timeout .....	4
1.16.9	The "Free Go" function when using FoUSB .....	5
2	System Requirements .....	6
2.1	M16C R8C FoUSB/UART Debugger .....	6
3	Version Report .....	7
3.1	M16C R8C FoUSB/UART Debugger V.1.03.00 .....	7
3.1.1	Revisions to Restrictions .....	7
3.1.2	Functional Extensions and Modifications .....	7
3.2	M16C R8C FoUSB/UART Debugger V.1.02.00 .....	7
3.2.1	Revisions to Restrictions .....	7
3.2.2	Functional Extensions and Modifications .....	8
3.3	M16C R8C FoUSB/UART Debugger V.1.01.00 .....	8
3.3.1	Revisions to Restrictions .....	8
3.3.2	Functional Extensions .....	8
3.4	M16C R8C FoUSB/UART Debugger V.1.00.01 .....	8
3.4.1	Revisions to Restrictions .....	8
3.4.2	Functional Extensions .....	9

# 1 Notes

## 1.1 Line Assembly

Regardless of the Radix setting, the default for line assembly input is decimal. Specify H as the radix for a hexadecimal input.

## 1.2 RAM Monitor

### 1. Proportional Fonts

When a proportional font is selected, a part of the characters in the view may be hidden. Fixed fonts are recommended.

## 1.3 Memory

### 1. 8 bytes data operations

To set, fill, and copy 8 bytes data are not supported.

## 1.4 Script

### 1. Result of interactive command

When you invoke an interactive command, for example, Assemble and setMemoryByte, the running dialog box will appear and may hide the view of the results.

### 2. SCOPE Command

When you refer current scope name with SCOPE command after program execution, the scope of the start-up module will be returned even if scope has been changed to the other.

## 1.5 Real-time OS debugging functions

### 1. When several labels are allocated to the entry address of the tasks or handlers, the task name or the handler name displayed in the windows may be different from the actual function name.

### 2. When you use the feature to issue system-calls by the script command (MR SYS), the target program should be built with a specific option. For details, refer to the topic "Prepare the real-time OS debug" in the online help.

## 1.6 Macro recording function

The debug windows which support the macro recording function are memory, registers, IO, ASM watch, and C watch. And also, the debug operations which support this are Reset CPU, Go, Reset Go, Go To Cursor, Step In, Step Over, Step Out, Add/Delete break points, and Download the target program.

## 1.7 Test facility function

The contents to be compared by the test facility functions are memory, registers, I/O, Output, stack race, ASM watch, and C watch.

## 1.8 Using cast operators for the member variable

When you use cast operators for the member variable to refer to it as the pointer of the structure, you would not refer to it correctly.

## 1.9 Download module dialog box

This debugger does not support the setting of "Offset", "Memory verify on download", and "Access Size" in the download module setting dialog box. These are always treated as "Offset: 0", "Memory verify: off", and "Access Size: 1".

## 1.10 The option "Always treat variables of enumerator type with unknown size as 1 byte"

The "Always treat variables of enumerator type with unknown size as 1 byte" option is effective after downloading the program. When the option status is changed, target program should be downloaded again.

And also, this option is effective for all variables of enumerator type in the program, even if the compiling options are different for each file.

## 1.11 Debugging for assembler macros

When the break points are set at the assembler macro codes, the break points would be set at the different address or not be displayed as the PC line.

## 1.12 Debugging for inline functions

When stepping the function including the call for a inline function, local variables would not be able to be referred.

## 1.13 Automatic target connection on changing the session

When the target connection is not performed on changing the session, select the menu [Debug] -> [Connect]. To perform automatic target connection, remove the check from the option "Do not perform automatic target connection" in the Option tab on Debug Setting dialog box which is invoked by the menu [Debug] -> [Debug Settings...].

## 1.14 Run program option

The "Run Program" dialog box enables to specify several temporary PC breakpoints, but this debugger only supports one breakpoint which is listed first in the "Temporary PC breakpoints" list box.

## 1.15 Selection of the object format for download module

When the specified file format in the debug setting dialog box is different from the format of the object module file, downloading the file may cause a freeze of the debugger. Please select the correct object format. And also, when selecting the object format for download module file, if there are two or more object format, whose name includes the vender name another ones do not include it, prioritize the file whose name includes vender name leading the object format name.

## 1.16 Notes on Debugging (M16C R8C FoUSB/UART Debugger)

### 1.16.1 Variables assigned to registers

When you build a target program, IAR EWM16C may generate the warning like following:

```
Warning [w23]: Cannot represent location of Register pair variable  
'R1H:R1L'
```

In this case, you can not see the variable, which is assigned to R1 in the emulator software. IAR will fix it by upgrading XLINK in near future. Please ask IAR about it for more details.

### 1.16.2 Limitations on Stop and Wait Modes

To use stop or wait modes in the User Program, start the M16C R8C FoUSB/UART software in free-run mode. If you want to debug, close the RAM, C Watch and ASM Watch Windows before actually debugging. Furthermore, set a breakpoint or take other necessary measures when getting out of stop or wait mode in order to ensure that no window operations will be performed until the program stops at the breakpoint.

### 1.16.3 Real-time performance of Watchdog Timer

If your target program uses watchdog timer, to get real time performance of watchdog timer on M16C R8C FoUSB / UART Software, start the software in free-running mode and close RAM monitor, C Watch and ASM Watch. Because USB monitor program initializes the watchdog timer.

### 1.16.4 Realtime Capability of the User Program

#### 1. SamplingRun mode

In SamplingRun mode, the FoUSB/UART software periodically monitors the execution status of the user program when executing the Go or Come command. This enables, for example, stoppage of the user program by a break to be detected. Select this mode when you perform

ordinary debugging.

## 2. FreeRun mode

In FreeRun mode, the FoUSB/UART software does not monitor the execution status of the user program while executing the Go or Come command. Therefore, although the real-time capability of the user program will be maintained, stoppage of the user program by a break point, etc. cannot be detected.

Consequently, even when the user program has stopped, the FoUSB/UART software does not stop running the Go or Come command. Press the STOP button to halt the FoUSB/UART software. Select this mode when the user program needs to be executed in real-time.

### Caution:

When using FreeRun mode, make sure the RAM Monitor, C Watch and ASM Watch Windows should be closed. This is because if these windows are open, the real-time capability of the user program will be impaired.

### 1.16.5 On debugging of MCUs of the R8C/14, 15, 16, 17 group

When a target is connected to a host PC with serial communication using UART of the MCUs of the R8C/14, R8C/15, R8C/16, and R8C/17 groups for debug, if three or more breakpoints are set, the program will stop at the point which was a breakpoint before, or will not stop at the breakpoints.

Use the following MCU files, when debugging a program under above-mentioned environment. The number in which the breakpoint can be set is limited to two.

- R5F21144UART.MCU for R8C/14 group
- R5F21145UART.MCU for R8C/15 group
- R5F21146UART.MCU for R8C/16 group
- R5F21147UART.MCU for R8C/17 group

### 1.16.6 The "Go to cursor" function when using FoUSB

Usually, the "Go to cursor" function ignores break points existing between the current PC and the cursor position. However, when the following conditions are all satisfied, the target program stops at a break point before the pointed cursor position.

- Two or more break points are set.
- The second or later break point is set at the area between the current PC and the pointed cursor position.
- After setting the second or later break point, execute "Go to cursor" function immediately before executing "Go", "Reset", or "Step".

### 1.16.7 The Pass counts for software break points

When the running mode is "FreeRun Mode", the pass counts for software break points will be unavailable. If the pass count is specified, it is regarded as '1'.

### 1.16.8 Communication Timeout

If the operating frequency is so slow and it takes long time to respond from the target system to the debugger, the debugger may regard it as a communication timeout error. In this case, please change the timeout setting by the following script command in Script Window.

```
_SetTimeOut "timeout for except download", "timeout for download"
```

For example, to set the timeout for download to 120 seconds and set the timeout for other communication to 30 seconds, please input like below:

```
_SetTimeOut 30, 120
```

The default value for download is 15 seconds and the value for except download is 5 seconds. And also, the minimum value is 4 seconds. Please note that this setting is not available for FoUSB connection.

### **1.16.9 The "Free Go" function when using FoUSB**

The target program may stop at a software break point, even though the Free-Go is invoked. In this case, please invoke Reset-Go again.

## 2 System Requirements

### 2.1 M16C R8C FoUSB/UART Debugger

Target host PC	
PC	IBM PC/AT compatible with Pentium III 600MHz or higher
OS	Windows XP Windows 2000
Memory	128MB or more (and in addition, ten or more times as much size as the load module file).
HDD	Hard disk available capacity for installation: 100MB or more. Prepare an area at least double the memory capacity (four-times or more recommended) as the swap area.
Display Resolution	1024 x 768 or higher recommended

Please refer to the URL below about the supported MCU:

[http://www.renesas.com/m16c\\_r8c\\_fousb\\_uart\\_dbg](http://www.renesas.com/m16c_r8c_fousb_uart_dbg)

## 3 Version Report

This section describes the specification of the changed software.

### 3.1 M16C R8C FoUSB/UART Debugger V.1.03.00

In this version, the following specifications were changed from the previous version M16C R8C FoUSB/UART Debugger V.1.02.00.

This version supports all of the function extensions and the revisions to the restrictions in the High-performance Embedded Workshop V.4.02.00 and V.4.03.00. For more details, please refer to the RENESAS TOOL NEWS “061216/tn2” issued on December 16, 2006 and “070701/tn1” issued on September 1<sup>st</sup>, 2007.

#### 3.1.1 Revisions to Restrictions

1. A limitation has been corrected: if the On Demand check box is checked in the Debugging Information tab of the Init dialog box, which appears when you invoke your debugger, target programs may not be loaded successfully on MCUs and not run properly. (For more details, refer to the RENESAS TOOL NEWS “070416/tn8” issued on April 16, 2007).
2. A limitation has been corrected: If the size of the variable to be referred by the quick watch feature is larger than 256 bytes, High-performance Embedded Workshop would crash. (For more details, refer to the RENESAS TOOL NEWS “070601/tn5” issued on June 1<sup>st</sup>, 2007).

#### 3.1.2 Functional Extensions and Modifications

1. Displaying source files information automatically in the Workspace Window. When a download module file is downloaded, the High-performance Embedded Workshop obtains source files information contained in the download module file through its debug information, and displays it under "download module" node in the Projects tab on the Workspace Window. Note that this function is available only for debugging the debug-only projects.
2. Supports C watch window included in High-performance Embedded Workshop V.4.03. The feature Changing scope of the variable and suppression leading zero are supported.
3. The option to specify the size (1byte or 2byte) of enumerator type is added.
4. Supports “Disconnect target” feature.
5. When the instruction is modified by the line assemble feature, if the old instruction length is longer than the new instruction length, NOP instructions are inserted automatically so as to suit the old instruction length.
6. The instruction format specifier is displayed for each instruction in disassembly. And also, can be switched not to display.
7. Supports to substitute the bit field member variable.
8. The default value of “Reset CPU after download module” is changed to be checked.

### 3.2 M16C R8C FoUSB/UART Debugger V.1.02.00

In this version, the following specifications were changed from the previous version M16C R8C FoUSB/UART Debugger V.1.01.00.

This version supports all of the function extensions and the revisions to the restrictions in the High-performance Embedded Workshop V.4.01.00 and V.4.01.01. For more details, please refer to RENESAS TOOL NEWS “060701/tn1” issued on July 1, 2006 and “060801/tn1” issued on August 1, 2006.

#### 3.2.1 Revisions to Restrictions

A limitation has been corrected: The structure member variables, union member variables, or class member variables whose name begins with a letter of ‘e’ or ‘E’ immediately followed by a numeral, are not referenced.

For more details, refer to the RENESAS TOOL NEWS RSO-M3T-PD32RM-060116D issued on January 16, 2006.

### 3.2.2 Functional Extensions and Modifications

1. The real-time OS, M3T-MR30/4 which is compliant to  $\mu$ ITRON 4.0 specifications, has been supported.
2. The ELF/DWARF2 format files generated using KPIT GNUM16C have been supported. These commands, which can be invoked in Command Line, have been supported:  
breakpoint, breakpoint\_disable, breakpoint\_display, breakpoint\_clear  
register\_display, register\_set  
disassemble, assemble
3. The default value of the communication time out for UART connection has been modified.

### 3.3 M16C R8C FoUSB/UART Debugger V.1.01.00

In this version, the following specifications were changed from the previous version M16C R8C FoUSB/UART Debugger V.1.00.01.

This version supports all of the function extensions and the revisions to the restrictions in the High-performance Embedded Workshop V.4.00.02. For more details, please refer to RENESAS TOOL NEWS RSO-HEW\_1-050701D issued on July 1, 2005.

#### 3.3.1 Revisions to Restrictions

1. A limitation has been corrected: To keep quickly pushing the step button on the toolbar switches the source mode of the source window into the disassembly mode.
2. A limitation has been corrected: when the target program was executed with the free-run mode, you might not be able to stop the target program.

#### 3.3.2 Functional Extensions

1. The following windows for the real-time OS debugging are added to this version.
  - (a) MR Window

### 3.4 M16C R8C FoUSB/UART Debugger V.1.00.01

In this version, the following specifications were changed from the previous version M16C R8C FoUSB/UART Debugger V.1.00.00.

#### 3.4.1 Revisions to Restrictions

1. A limitation has been corrected: The automatic backup function of workspaces does not operate properly.  
For more details, please refer to RENESAS TOOL NEWS RSO-HEW\_1-050216D issued on February 16, 2005.
2. A limitation has been corrected: When connecting with emulator PC7501 or compact emulator in a debugging session of them, High-performance Embedded Workshop might load break points using address match interrupt that are deleted before.
3. A limitation has been corrected: When a setting of RAM monitoring area is changed with RAM Monitor Area Setting window, the setting displayed on-screen may not match with the actual ones, resulting in improper operation.  
For more details, refer to RENESAS TOOL NEWS RSO-M3T-PD308MF-050216D issued on February 16, 2005.
4. A limitation has been corrected: If any of the following operations is performed with the C Watch window being undocked, the High-performance Embedded Workshop abnormally shut down.
  - (a) Switching between sessions
  - (b) Closing a workspace
  - (c) Terminating an application programFor more details, please refer to RENESAS TOOL NEWS RSO-HEW\_2-050301D issued on March 1, 2005.
5. A limitation has been corrected: C watch window may not load correct variable names which

are registered in the watch tab.

This problem occurs if a variable name is entered in any of the following ways:

- (a) An element of an array is specified using [ ] operator.  
Example: a[0]
- (b) A member of a structure or union is specified using . operator.  
Example: str.a
- (c) The destination of a pointer is pointed to using -> operator.  
Example: pstr->a
- (d) An Address or indirection operator (& or \*) is used.  
Example: &a or \*a

For more details, please refer to RENESAS TOOL NEWS RSO-HEW\_3-050301D issued on March 1, 2005.

6. A limitation has been corrected: The drag and drop operation in memory window may cause the general protection failure.

### **3.4.2 Functional Extensions**

1. Script window is modified to be able to change the display font.
2. The menu to open the online help for the emulator is added.