

致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<http://www.renesas.com/inquiry>

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

M16C R8C PC7501 仿真调试器 V.1.03

瑞萨单片机开发环境系统

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的本公司产品的参考资料，对于本资料中所记载的技术信息，并非意味着对本公司或者第三者的知识产权及其他权利做出保证或对实施权力进行的承诺。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法及其他应用电路例而引起的损害或者对第三者的知识产权及其他权利造成侵犯，本公司不承担任何责任。
3. 不能将本资料所记载的产品和技术用于大规模破坏性武器的开发等目的、军事目的或其他的军需用途方面。另外，在出口时必须遵守日本的《外汇及外国贸易法》及其他出口的相关法令并履行这些法令中规定的必要手续。
4. 本资料所记载的产品数据、图、表、程序、算法以及其他应用电路例等所有信息均为本资料发行时的内容，本公司有可能在未做事先通知的情况下，对本资料所记载的产品或者产品规格进行更改。所以在购买和使用本公司的半导体产品之前，请事先向本公司的营业窗口确认最新的信息并经常留意本公司通过公司主页（<http://www.renesas.com>）等公开的最新信息。
5. 对于本资料中所记载的信息，制作时我们尽力保证出版时的精确性，但不承担因本资料的叙述不当而致使顾客遭受损失等的任何相关责任。
6. 在使用本资料所记载的产品数据、图、表等所示的技术内容、程序、算法及其他应用电路例时，不仅要对所使用的技术信息进行单独评价，还要对整个系统进行充分的评价。请顾客自行负责，进行是否适用的判断。本公司对于是否适用不负任何责任。
7. 本资料中所记载的产品并非针对万一出现故障或是错误运行就会威胁到人的生命或给人体带来危害的机器、系统（如各种安全装置或者运输交通用的、医疗、燃烧控制、航天器械、核能、海底中继用的机器和系统等）而设计和制造的，特别是对于品质和可靠性要求极高的机器和系统等（将本公司指定用于汽车方面的产品用于汽车时除外）。如果要用于上述的目的，请务必事先向本公司的营业窗口咨询。另外，对于用于上述目的而造成的损失等，本公司概不负责。
8. 除上述第7项内容外，不能将本资料中记载的产品用于以下用途。如果用于以下用途而造成的损失，本公司概不负责。
 - 1) 生命维持装置。
 - 2) 植埋于人体使用的装置。
 - 3) 用于治疗（切除患部、给药等）的装置。
 - 4) 其他直接影响到人的生命的装置。
9. 在使用本资料所记载的产品时，对于最大额定值、工作电源电压的范围、放热特性、安装条件及其他条件请在本公司规定的保证范围内使用。如果超出了本公司规定的保证范围使用时，对于由此而造成的故障和出现的事故，本公司将不承担任何责任。
10. 本公司一直致力于提高产品的质量和可靠性，但一般来说，半导体产品总会以一定的概率发生故障、或者由于使用条件不同而出现错误运行等。为了避免因本公司的产品发生故障或者错误运行而导致人身事故和火灾或造成社会性的损失，希望客户能自行负责进行冗余设计、采取延烧对策及进行防止错误运行等的安全设计（包括硬件和软件两方面的设计）以及老化处理等，这是作为机器和系统的出厂保证。特别是单片机的软件，由于单独进行验证很困难，所以要求在顾客制造的最终的机器及系统上进行安全检验工作。
11. 如果把本资料所记载的产品从其载体设备上卸下，有可能造成婴儿误吞的危险。顾客在将本公司产品安装到顾客的设备上时，请顾客自行负责将本公司产品设置为不容易剥落的安全设计。如果从顾客的设备上剥落而造成事故时，本公司将不承担任何责任。
12. 在未得到本公司的事先书面认可时，不可将本资料的一部分或者全部转载或者复制。
13. 如果需要了解关于本资料的详细内容，或者有其他关心的问题，请向本公司的营业窗口咨询。

概要

本高性能嵌入式工作区（High-performance Embedded Workshop，简称 HEW）是一种图形用户界面，使用它可以轻松开发和调试使用 C/C++ 编程语言和汇编语言编写的瑞萨单片机应用程序。它旨在提供一种高性能且直观的方式来存取、观察和修改运行应用程序的调试平台。

本帮助手册说明 HEW 作为一种“调试器”的功能。

目标系统

该调试器是在仿真器 PC7501 系统上进行操作的。

支持的 CPU

本帮助手册说明与以下 CPU 相对应的调试功能。

- M32C/80、M16C/70 系列
注意：在本帮助手册中，依赖于该 CPU 的信息被描述为“用于 M32C”。
- M16C/60、M16C/30、M16C/Tiny、R8C/Tiny 系列
注意：在本帮助手册中，依赖于该 CPU 的信息被描述为“用于 M16C/R8C”。

Active X、Microsoft、MS-DOS、Visual Basic、Visual C++、Windows 和 Windows NT 是 Microsoft Corporation 在美国和其他国家或地区的注册商标或商标。

IBM 和 AT 是 International Business Machines Corporation 的注册商标。

Intel 和 Pentium 是 Intel Corporation 的注册商标。

Adobe 和 Acrobat 是 Adobe Systems Incorporated 的注册商标。

所有其他品牌和产品名称均为其各自持有者的商标、注册商标或服务标志。

如需咨询本文档或产品的内容，请填写安装程序在以下目录中生成的文本文件，并通过电子邮件发送给您当地的经销商。

¥ SUPPORT ¥ Product-name ¥ SUPPORT.TXT

瑞萨工具主页：<http://www.renesas.com/en/tools>

目 录

第 1 部分 设置调试器部分	1
1 功能	3
1.1 实时 RAM 监控功能.....	3
1.1.1 RAM 监控功能	3
1.1.2 采样周期	3
1.1.3 相关窗口	4
1.2 断点功能.....	4
1.2.1 软件断点功能	4
1.2.2 硬件断点	4
1.2.3 关于中断保护功能	5
1.2.4 关于地址中断断点功能	6
1.3 实时跟踪功能.....	6
1.3.1 跟踪区域	7
1.3.2 跟踪条件设置	8
1.3.3 跟踪数据写条件	8
1.4 时间计量功能.....	8
1.4.1 计量条件	9
1.5 覆盖率功能.....	9
1.5.1 覆盖率计量区域	9
1.5.2 相关窗口	10
1.6 实时操作系统调试功能.....	10
1.7 GUI 输入 / 输出功能.....	10
2 关于仿真器 PC7501	11
2.1 通信方法.....	11
2.2 功能表.....	11
3 启动调试器之前	12
3.1 仿真器使用的通信方法.....	12
3.1.1 USB 接口	12
3.1.2 LAN 接口	12
3.1.3 LPT 接口	12
3.2 下载固件.....	12
3.3 在仿真器启动之前进行设置.....	13
3.3.1 USB 通信	13
3.3.2 LAN 通信	14

4	使用前的准备工作	16
4.1	工作区、工程和文件.....	16
4.2	启动 HEW.....	17
4.2.1	创建新的工作区（使用工具链）	18
4.2.2	创建新的工作区（不使用工具链）	23
4.3	启动调试器.....	28
4.3.1	连接仿真器	28
4.3.2	结束仿真器	28
5	设置调试器	29
5.1	Init（初始化）对话框	29
5.1.1	MCU 选项卡	30
5.1.2	Debugging Information（调试信息）选项卡	33
5.1.3	Emulator（仿真器）选项卡	35
5.1.4	Script（脚本）选项卡	37
5.2	设置通信接口.....	38
5.2.1	设置 USB 接口	38
5.2.2	设置 LAN 接口	38
5.2.3	设置 LPT 接口	40
5.3	设置 M32C 使用的调试器	41
5.3.1	Emem 对话框	41
5.4	设置 M16C/R8C 调试器.....	47
5.4.1	MCU Setting（MCU 设置）对话框	47
第 2 部分 教程		52
6	教程	54
6.1	简介.....	54
6.2	使用.....	55
6.2.1	步骤 1：启动调试器	55
6.2.2	步骤 2：检查 RAM 的运作情况	55
6.2.3	步骤 3：下载教程程序	56
6.2.4	步骤 4：设置断点	58
6.2.5	步骤 5：执行程序	59
6.2.6	步骤 6：查看断点	61
6.2.7	步骤 7：查看寄存器	62
6.2.8	步骤 8：查看存储器	63
6.2.9	步骤 9：监视变量	64
6.2.10	步骤 10：逐步执行程序	66
6.2.11	步骤 11：强制暂停程序执行	68
6.2.12	步骤 12：显示局部变量	69
6.2.13	步骤 13：堆栈跟踪功能	70
6.2.14	后续内容	70

7	窗口 / 对话框	71
7.1	[RAM Monitor] (RAM 监控) 窗口.....	73
7.1.1	扩展菜单	75
7.1.2	设置 RAM 监控区域	76
7.2	[ASM Watch] (ASM 监视) 窗口	78
7.2.1	扩展菜单	79
7.3	[C Watch] (C 监视) 窗口	80
7.3.1	扩展菜单	82
7.4	[Coverage] (覆盖率) 窗口.....	83
7.4.1	扩展菜单	84
7.4.2	查看源代码 / 已执行地址	85
7.5	[Script] (脚本) 窗口.....	86
7.5.1	扩展菜单	87
7.6	[S/W Break Point Setting] (软件断点设置) 窗口.....	88
7.6.1	命令按钮	89
7.6.2	从 [Editor(Source)] (编辑器 (源)) 窗口设置和删除断点	90
7.7	[H/W Break Point Setting] (硬件断点设置) 窗口.....	91
7.7.1	指定断点事件	92
7.7.2	指定组合条件	96
7.7.3	指定进程 ID	97
7.7.4	命令按钮	97
7.7.5	指定事件 (取指令)	98
7.7.6	指定事件 (存储器存取)	102
7.7.7	指定事件 (位存取)	123
7.7.8	指定事件 (中断)	125
7.7.9	指定事件 (外部触发信号)	126
7.7.10	指定事件组合条件	128
7.7.11	指定进程 ID	131
7.8	[Protect] (保护) 窗口.....	132
7.8.1	扩展菜单	132
7.9	[Address Interrupt Break Point Setting] (地址中断断点设置) 窗口.....	133
7.9.1	命令按钮	134
7.9.2	从 [Editor(Source)] (编辑器 (源)) 窗口设置和删除断点	135
7.10	[Trace Point Setting] (跟踪点设置) 窗口	136
7.10.1	指定跟踪事件	137
7.10.2	指定组合条件	140
7.10.3	指定进程 ID	141
7.10.4	指定跟踪范围	142
7.10.5	指定跟踪写条件	142
7.10.6	命令按钮	143
7.10.7	指定事件 (取指令)	143
7.10.8	指定事件 (存储器存取)	143

7.10.9	指定事件（位存取）	143
7.10.10	指定事件（中断）	143
7.10.11	指定事件（外部触发信号）	143
7.10.12	指定事件组合条件	144
7.10.13	指定进程 ID	144
7.10.14	指定写条件	144
7.11	[Time Measurement]（时间计量）窗口	148
7.11.1	指定时间计量事件	149
7.11.2	时间计量条件	153
7.11.3	命令按钮	153
7.11.4	指定事件（取指令）	154
7.11.5	指定事件（存储器存取）	154
7.11.6	指定事件（位存取）	154
7.11.7	指定事件（中断）	154
7.11.8	指定事件（外部触发信号）	154
7.11.9	设置计量条件	155
7.12	[Trace]（跟踪）窗口	159
7.12.1	配置“Bus Mode”（总线模式）	160
7.12.2	配置“Disassemble Mode”（反汇编模式）	161
7.12.3	配置“Data Access Mode”（数据存取模式）	162
7.12.4	配置“Source Mode”（源模式）	163
7.12.5	扩展菜单	164
7.12.6	显示 M32C 调试器的总线信息	165
7.12.7	显示 M16C/R8C 调试器的总线信息	167
7.13	[Data Trace]（数据跟踪）窗口	168
7.13.1	扩展菜单	169
7.14	[GUI I/O] 窗口	170
7.14.1	扩展菜单	171
7.15	[MR] 窗口	172
7.15.1	扩展菜单	173
7.15.2	显示任务状态	174
7.15.3	显示就绪队列状态	177
7.15.4	显示超时队列状态	178
7.15.5	显示事件标志状态	180
7.15.6	显示信标状态	182
7.15.7	显示邮箱状态	183
7.15.8	显示数据队列状态	185
7.15.9	显示周期处理程序状态	187
7.15.10	显示警报处理程序状态	188
7.15.11	显示存储池状态	189
7.15.12	显示任务上下文	191

7.16 [MR Trace] (MR 跟踪) 窗口.....	193
7.16.1 扩展菜单	195
7.16.2 查看任务的执行历史记录 ([MRxx] 窗口)	196
7.17 [MR Analyze] (MR 分析) 窗口	202
7.17.1 配置 “CPU Occupancy Status Display Mode” (CPU 占用状态显示模式)	202
7.17.2 配置 “Ready State Duration Display Mode” (就绪状态持续时间显示模式)	203
7.17.3 配置 “System Call History Display Mode” (系统调用历史记录显示模式)	203
7.17.4 扩展菜单	204
7.17.5 分析任务的执行历史记录	204
7.18 [MR Task Pause] (MR 任务暂停) 窗口	207
7.18.1 关于任务暂停功能	207
7.18.2 扩展菜单	208
7.18.3 暂停指定任务	209
7.19 [Task Trace] (任务跟踪) 窗口	214
7.19.1 扩展菜单	215
7.19.2 查看任务的执行历史记录 ([Taskxx] 窗口)	216
7.20 [Task Analyze] (任务分析) 窗口	221
7.20.1 扩展菜单	221
7.20.2 分析任务的执行历史记录	222
第 3 部分 参考	223
8 脚本命令表	225
8.1 脚本命令表 (按功能分类)	225
8.1.1 执行命令	225
8.1.2 文件操作命令	225
8.1.3 寄存器操作命令	226
8.1.4 存储器操作命令	226
8.1.5 汇编 / 反汇编命令	226
8.1.6 软件断点设置命令	227
8.1.7 地址中断断点设置命令	227
8.1.8 硬件断点设置命令	227
8.1.9 实时跟踪命令	227
8.1.10 覆盖率计量命令	227
8.1.11 脚本 / 日志文件命令	228
8.1.12 程序显示命令	228
8.1.13 映像命令	228
8.1.14 时钟命令	228
8.1.15 监视定时器命令	229
8.1.16 C 语言调试命令	229
8.1.17 实时操作系统命令	229
8.1.18 实用程序命令	229
8.2 脚本命令表 (依字母顺序)	230

9	编写脚本文件	233
9.1	脚本文件的结构元素	233
9.1.1	脚本命令	234
9.1.2	赋值语句	234
9.1.3	条件语句	234
9.1.4	循环语句（while 及 endw）和中断语句	235
9.1.5	注释语句	235
9.2	编写表达式	236
9.2.1	常数	236
9.2.2	符号和标签	237
9.2.3	Macro Variables 宏变量	238
9.2.4	寄存器变量	238
9.2.5	存储器变量	238
9.2.6	行号	239
9.2.7	字符常数	239
9.2.8	运算符	239
10	C/C++ 表达式	240
10.1	编写 C/C++ 表达式	240
10.1.1	直接值	240
10.1.2	作用域解析	241
10.1.3	数学运算符	241
10.1.4	指针	241
10.1.5	引用	241
10.1.6	取反	241
10.1.7	使用点运算符的成员引用	242
10.1.8	使用箭头运算符的成员引用	242
10.1.9	指向成员的指针	242
10.1.10	括号	243
10.1.11	数组	243
10.1.12	基本数据类型转换	243
10.1.13	typedef 类型转换	243
10.1.14	变量名	243
10.1.15	函数名	244
10.1.16	字符常数	244
10.1.17	字符串文字	244
10.2	C/C++ 表达式的显示格式	245
10.2.1	枚举类型	245
10.2.2	基本数据类型	245
10.2.3	指针类型	246
10.2.4	数组类型	247
10.2.5	函数类型	247
10.2.6	引用类型	247

10.2.7	位域类型	247
10.2.8	未找到任何 C 符号时	248
10.2.9	语法错误	248
10.2.10	结构类型和联合类型	248
11	显示程序停止的原因	249
12	注意事项	250
12.1	通用注意事项	250
12.1.1	与 Windows 相关的文件操作	250
12.1.2	可设置软件断点的区域	250
12.1.3	获取或设置 C 变量	251
12.1.4	C++ 中的函数名	251
12.1.5	下载模块的选项设置	251
12.1.6	调试多个模块	251
12.1.7	同步调试	252
12.1.8	下载固件	252
12.1.9	LPT 端口限制	252
12.1.10	覆盖率功能的注意事项	253
12.1.11	仿真器复位开关	253
12.1.12	仿真器上的调试资源	253
12.1.13	启动仿真器和调试器	253
12.1.14	使用任务暂停功能和地址匹配中断断点功能	253
12.1.15	覆盖率计量的结果	253
12.2	M32C 调试器的注意事项	254
12.2.1	仿真器使用的堆栈区域	254
12.2.2	目标程序复位时的中断堆栈指针	254
12.2.3	C 编译器 / 汇编器 / 连接器选项	254
12.2.4	目标 MCU HOLD 终端	254
12.2.5	硬件事件	254
12.2.6	CPU 重写	255
12.2.7	MR STK 脚本命令	255
12.3	M16C/R8C 调试器的注意事项	256
12.3.1	连接到仿真器	256
12.3.2	仿真器使用的堆栈区域映像	256
12.3.3	目标程序复位时的中断堆栈指针	256
12.3.4	编译器、汇编器和连接器选项	256
12.3.5	TASKING C 编译器	256
12.3.6	目标 MCU HOLD 终端	256
12.3.7	硬件事件	257
12.3.8	任务暂停功能的对应操作系统版本	257
12.3.9	存储空间扩展	258
12.3.10	CPU 重写	259
12.3.11	MR STK 脚本命令	259

12.4 C 编译器 / 汇编器 / 连接器选项.....	260
12.4.1 使用 NCxx	260
12.4.2 使用 IAR EC++ 编译器 (EW)	260
12.4.3 使用 IAR C 编译器 (EW)	260
12.4.4 使用 IAR C 编译器 (ICC)	261
12.4.5 使用 TASKING C 编译器 (CM)	262

设置调试器部分

1 功能

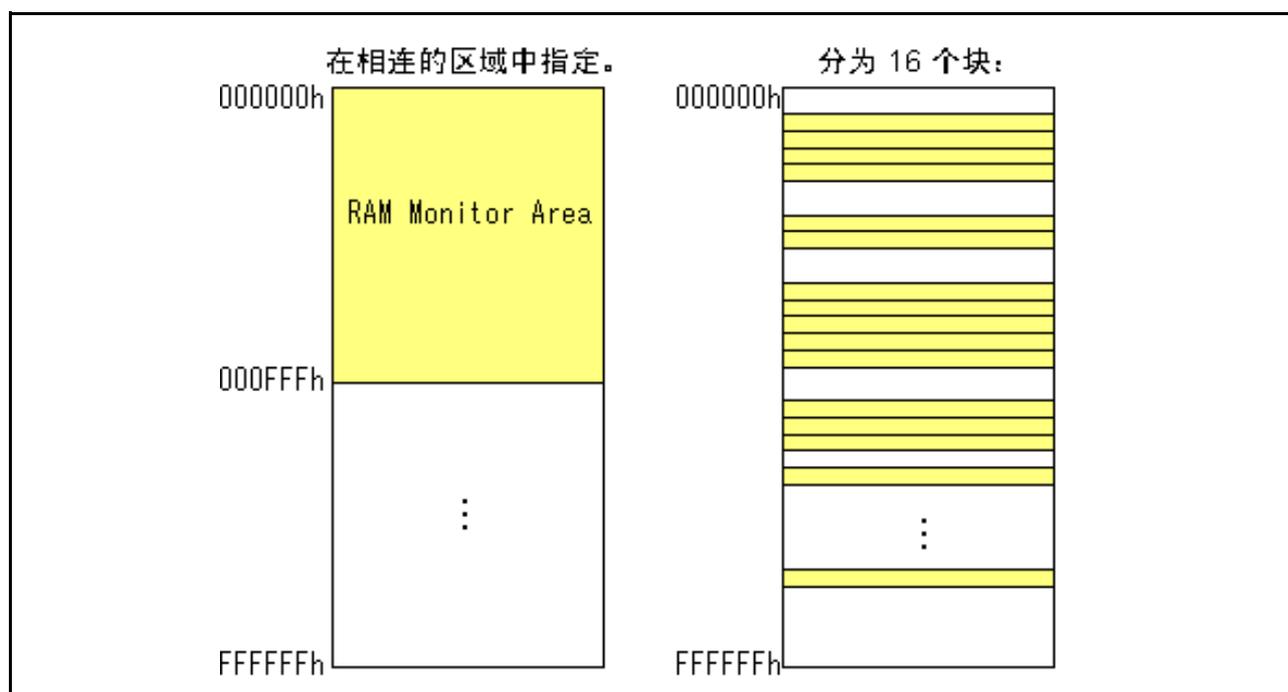
1.1 实时 RAM 监控功能

该功能可让您在不影响目标程序执行的实时性能的情况下，检查存储器内容的更改。

PC7501 仿真器系统具有 4 KB 的 RAM 监控区域，该区域可以位于任何邻接的地址位置或 16 个单独的块（每块由 256 个字节组成）中。

1.1.1 RAM 监控功能

该调试器具有 4 KB 的 RAM 监控区域，该区域可以位于任何相连的地址位置或 16 个单独的块（每块由 256 个字节组成）中。



1.1.2 采样周期

采样周期表示显示更新间隔。

您可以在支持 RAM 监控的任何窗口中指定该功能。（默认情况下，设置 100 ms 的间隔。）

根据不同的操作环境，实际采样周期可能比指定的周期更长。（采样周期取决于以下环境。）

- 通信接口
- 显示的 [RAM Monitor]（RAM 监控）窗口数
- 显示的 [RAM Monitor]（RAM 监控）窗口大小
- [ASM Watch]（ASM 监视）窗口的 RAM 监控区域内的 ASM 监视点数
- [C Watch]（C 监视）窗口的 RAM 监控区域内的 C 监视点数

1.1.3 相关窗口

下面显示可以使用实时 RAM 监控功能的窗口。

- [RAM Monitor] (RAM 监控) 窗口
- [ASM Watch] (ASM 监视) 窗口
- [C Watch] (C 监视) 窗口

1.2 断点功能

1.2.1 软件断点功能

软件断点在指定地址执行命令之前中断目标程序。该断点称为软件断点。

软件断点是在 [Editor (Source)] (编辑器 (源)) 窗口或 [S/W Breakpoint Setting] (软件断点设置) 窗口中进行设置 / 复位的。也可以暂时禁止 / 允许软件断点。

最多可以指定 64 个软件断点。指定两个或更多软件断点时, 根据 OR 逻辑来组合断点。(到达任一断点均会中断目标程序。)

1.2.1.1 设置软件断点

可以通过以下窗口设置软件断点。

- [Editor (Source)] (编辑器 (源)) 窗口
- [S/W Break Point Setting] (软件断点设置) 窗口

可以在 [Editor (Source)] (编辑器 (源)) 窗口中双击鼠标来设置 / 复位软件断点。

也可以改为在 [S/W Break Point Setting] (软件断点设置) 窗口中暂时禁止 / 允许软件断点。

1.2.1.2 可设置软件断点的区域

可以针对软件断点进行设置的区域会根据产品的不同而有所不同。

有关可用于软件断点的区域, 请参阅以下章节:

“12.1.2 可设置软件断点的区域”

1.2.2 硬件断点

该功能使目标程序在以下情况停止: 检测到存储器数据读 / 写、指令执行或外部跟踪电缆供给的输入信号的上升沿 / 下降沿。

对于各个目标 MCU 而言, 可以设置的事件内容均有所不同。

可以指定以下内容作为断点事件:

- 地址指定
 - 取指令
 - 存储器存取
 - 位存取
- 外部触发器指定
- 中断

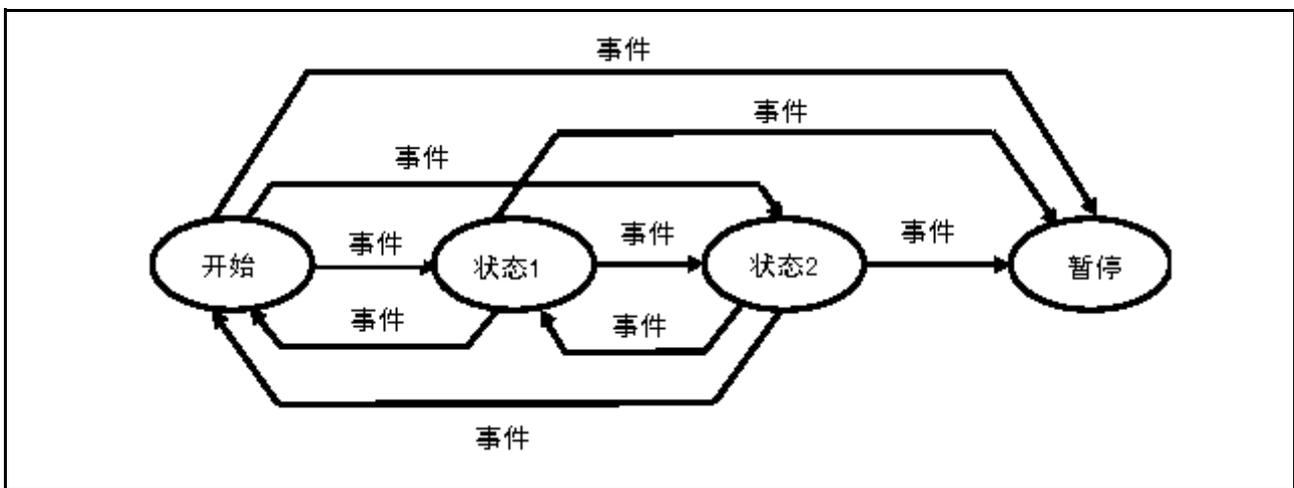
可以指定的事件数有 6 个。对于地址指定方法，除了普通的单地址指定之外，取指令和存储器存取还允许范围指定和逻辑条件指定。此外，取指令还允许您指定函数名称。

存储器存取允许您指定比较数据并在指定地址上进行数据读 / 写。它还允许对比较数据进行屏蔽指定。

这些断点事件可以按如下方式进行组合：

- 当所有的有效事件都已建立时进行跟踪（[AND]（与）条件）
- 当所有的有效事件都已同时建立时进行跟踪（同时的 [AND]（与）条件）
- 当其中一个有效事件已建立时进行跟踪（[OR]（或）条件）
- 在状态转换过程中进入暂停状态时进行跟踪（[State Transition]（状态转换）条件）

当满足在状态间传递中设置的转变条件时，会发生状态转换，在这种情况下，目标程序会在进入暂停状态时停止。下面显示的概念图描绘了状态和传递之间的关系。



状态转换断点允许您选择“specified task only”（仅指定任务）（或“other than specified task”（不同于指定任务））作为断点条件，以满足实时操作系统。

1.2.3 关于中断保护功能

中断保护是一种功能，用于检测向 ROM 区域写入数据的操作和存取未用区域的操作（读、写、命令执行），并停止目标程序。

1.2.3.1 保护属性

可以 256 字节为单位（256 字节边界）指定以下属性。

- “Access Disable”（禁止存取）
- “Read Only”（只读）
- “Write Only”（只写）
- “R/W Enable”（允许读/写）

1.2.3.2 存取保护区

整个存储器空间 (16 MB) 都是保护区。

默认情况下，启动仿真器时，整个保护区都设置为“R/W Enable”（允许读 / 写）。

1.2.3.3 中断保护的设置方法

提供了两种指定方法：

- 从目标程序段信息中取存储器属性
- 指定任何区域的存储器属性

不过，只能以 256 字节为单位设置属性。

将指定范围调整为与 256 字节边界对齐。

（示例：如果指定范围为 0x456 至 0x567，则将其调整为 0x400 至 0x5FF。）

1.2.4 关于地址中断断点功能

在指定地址上执行指令之前，该功能会及时停止目标程序。该功能是通过使用 MCU 的地址匹配中断来实现的。

只有在用户应用程序中未使用地址匹配中断时，才能使用地址中断断点功能。断点计数取决于连接的 MCU。

注意

只有在 [Init]（初始化）对话框的 MCU 选项卡下面选中了 [Enable the Address Match Interrupt Break Function]（允许地址匹配中断断点功能）复选框时，才能使用地址中断断点功能。（详细信息）。

但是，如果以下条件之一为真，则即使在这种情况下也无法设置任何地址中断断点：

- 在微处理器模式下操作时，包含地址中断向量的块映射为“外部”或“无用”区域。

1.2.4.1 设置和删除断点

可以通过以下窗口设置地址中断断点。

- [Editor (Source)]（编辑器（源））窗口
- [Address Interrupt Break Point Setting]（地址中断断点设置）窗口

可以在 [Editor (Source)]（编辑器（源））窗口中双击鼠标来设置 / 复位地址中断断点（与软件断点相同）。

也可以改为在 [Address Interrupt Break Point Setting]（地址中断断点设置）窗口中暂时禁止 / 允许地址中断断点。

1.3 实时跟踪功能

该功能记录目标程序执行历史记录。

最多可以记录 256K 执行周期历史记录。通过该记录，可以检查每个周期的总线信息、执行的指令和源程序执行路径。

实时跟踪功能记录目标程序的执行历史记录。

跟踪窗口会引用执行历史记录。

可以在以下模式中引用执行历史记录。

- “BUS mode”（总线模式）

该模式允许您按周期检查总线信息。显示内容取决于使用的 MCU 和仿真器系统。除总线信息外，该模式还允许以组合形式显示反汇编、源行或数据存取信息。

- “Disassemble mode”（反汇编模式）

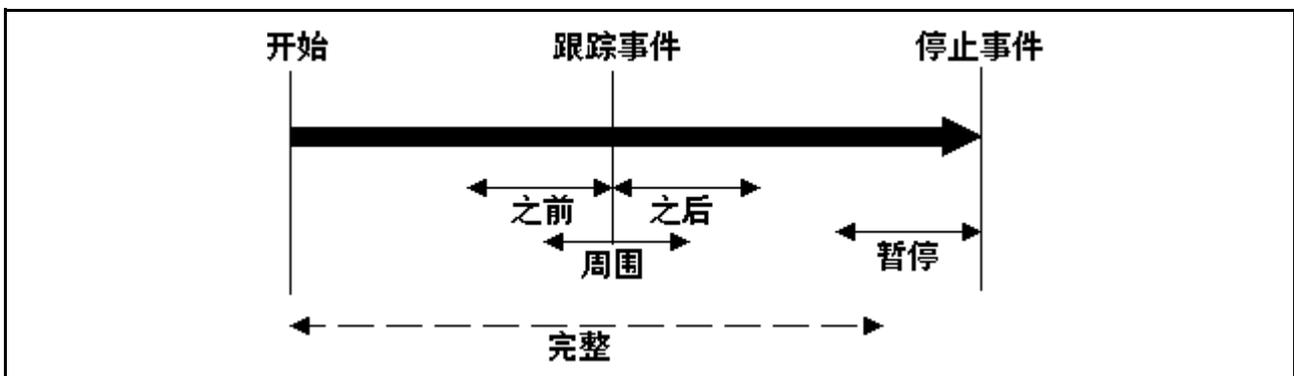
该模式允许您检查执行的指令。除反汇编信息外，该模式还允许以组合形式显示源行或数据存取信息。

- “Data access mode”（数据存取模式）
该模式允许您检查数据读/写周期。除数据存取信息外，该模式还允许以组合形式显示源行信息。
- “Source mode”（源模式）
该模式允许您检查源程序中的程序执行路径。

1.3.1 跟踪区域

可以使用此调试器引用 256K 个周期的执行历史记录。
支持下面 5 个模式的跟踪区域。

- “Break”（暂停）
目标程序停止前的 256K 个周期
- “Before”（之前）
跟踪点之前的 256K 个周期
- “About”（周围）
跟踪点两侧各 128K 个周期
- “After”（之后）
跟踪点之后的 256K 个周期
- “Full”（完整）
直到 256K 个周期写入跟踪存储器为止



“Break”（暂停）为默认设置。若要在停止目标程序之前引用执行历史记录，请使用“Break”（暂停）（不需要指定跟踪事件）。

若要在任何位置引用执行历史记录，或者继续执行目标程序，请指定跟踪事件并更改跟踪范围。

1.3.2 跟踪条件设置

可以指定以下内容作为跟踪事件：

- 地址指定
 - 取指令
 - 存储器存取
 - 位存取
- 外部触发器指定（八个事件）
- 中断

可以指定的事件数有 8 个。这些断点事件可以按如下方式进行组合：

- 当所有的有效事件都已建立时进行跟踪（[AND]（与）条件）
- 当所有的有效事件都已同时建立时进行跟踪（同时的 [AND]（与）条件）
- 当其中一个有效事件已建立时进行跟踪（[OR]（或）条件）
- 在状态转换过程中进入暂停状态时进行跟踪（[State Transition]（状态转换）条件）

可以选择“specified task only”（仅指定任务）或“other than specified task”（不同于指定任务）作为跟踪条件，以满足实时操作系统。

1.3.3 跟踪数据写条件

可以指定跟踪数据写条件。

可以指定以下写条件：

- 无限制的写条件（默认）
- 从建立开始事件到建立结束事件的周期
- 仅建立开始事件的周期
- 从建立开始事件到取消建立开始事件的周期
- 从建立开始事件到建立结束事件的周期之外
- 建立开始事件的周期之外
- 从建立开始事件到取消建立开始事件的周期之外

1.4 时间计量功能

该功能记录目标程序执行（存取）的地址（C0 覆盖率）。

该功能帮助记录程序停止运行之后未执行的地址。在测试过程中使用该覆盖率计量功能可以记录遗漏的测试项目。

时间计量功能允许您计量最长、最短和平均执行时间，以及指定区域的计量计数。

使用此调试器，最多可以同时计量四个点的时间。

1.4.1 计量条件

区域时间的计量条件可以指定每个计量区域的下列内容。

- 指定函数的执行时间
- 两个事件之间的时间
- 事件建立之间的时间
- 事件发生周期的时间

1.5 覆盖率功能

覆盖率计量的功能在于记录目标程序的执行（存取）的地址（C0 覆盖率）。

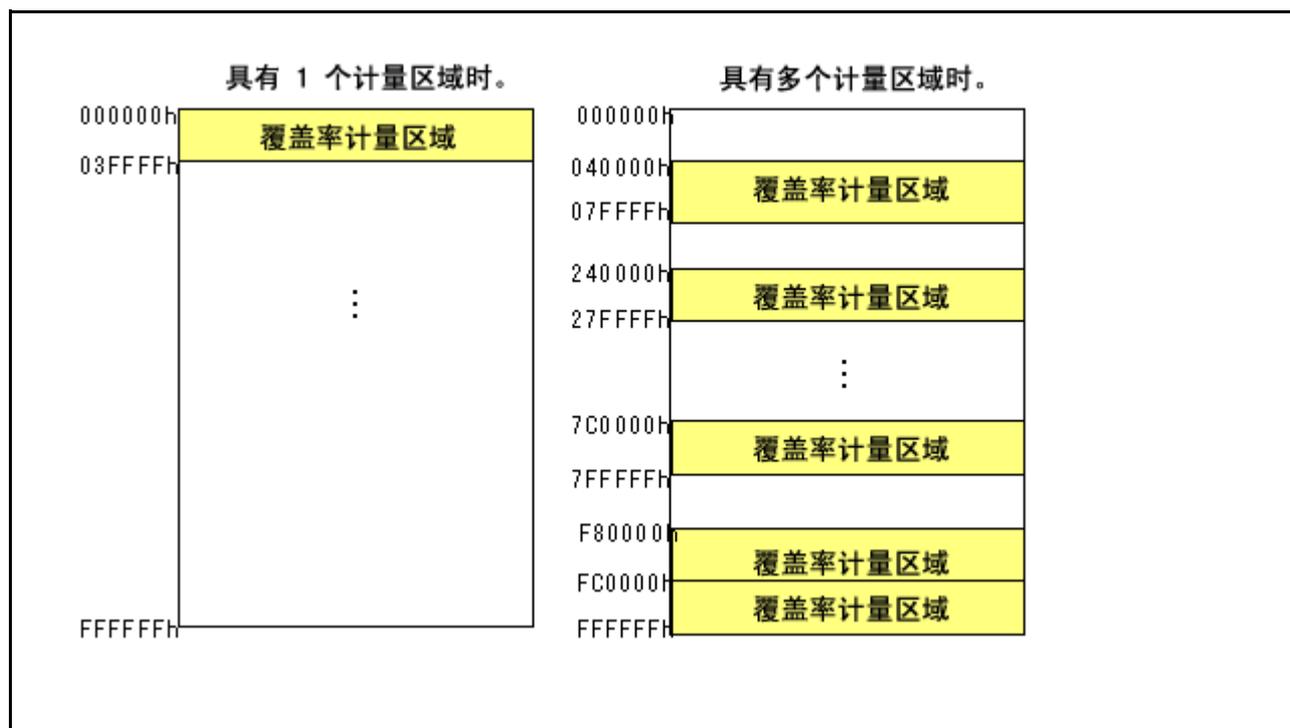
停止目标程序的执行之后，您可以了解到哪些地址尚未执行。

通过在测试过程中使用覆盖率计量功能，可以检查是否有遗漏的测试项目。

1.5.1 覆盖率计量区域

仿真器 PC7501 允许选择以 256 字节边界开始的、介于 1 到 32 之间的任意块（最多 8 MB）作为覆盖率计量区域。

请注意，覆盖率是以字（2 个字节）为单位进行计量的。



1.5.2 相关窗口

请在以下窗口中查看覆盖率计量结果。

- [Editor (Source)] (编辑器 (源)) 窗口
- [Memory] (存储器) 窗口
- [Coverage] (覆盖率) 窗口

1.6 实时操作系统调试功能

该功能调试目标程序中使用实时操作系统的相关源行部分。

该功能帮助显示实时操作系统的状态以及检查任务执行历史记录等等。

1.7 GUI 输入 / 输出功能

该功能在窗口上模拟用户目标系统的按键输入面板 (按钮) 和输出面板。

按钮可以用于输入面板, 标签 (字符串) 和 LED 可以用于输出面板。

2 关于仿真器 PC7501

PC7501 仿真器具有完整的总线跟踪功能，可用于围绕 M16C 处理器平台设计的系统中的内部电路仿真。该小型单元能够进行高达 66MHz 的仿真（即，可在 20MHz 以上频率情况下使用）。当与适合目标 MCU 的仿真探针搭配使用时，它可以帮助配置与使用的每个 MCU 匹配的仿真器系统。

2.1 通信方法

支持的通信方法如下。

I/F	仿真器
	PC7501
USB	支持
LAN	支持
LPT	支持

根据通信方法，在启动调试器之前，需要设置一些项目。请参阅“3.3 在仿真器启动之前进行设置”。

2.2 功能表

支持的功能如下。

功能	PC7501
软件断点	64 点
硬件断点	8 点
地址中断断点	4 点 *
实时跟踪	256K 个周期
RAM 监控	4K 字节（256 字节 x 16 块）区域
C0 覆盖率	256K 字节 x 32 块区域
时间计量	运行到停止 /4 点间隔
中断保护	存取中断保护

* 取决于使用的目标 MCU。

3 启动调试器之前

3.1 仿真器使用的通信方法

支持的通信方法如下。

USB、LAN、LPT

3.1.1 USB 接口

- 符合 USB 标准 1.1。
- 不支持通过 USB 集线器进行的连接。
- 通过使用 USB 电缆连接主机和仿真器，可以使用向导安装支持的设备驱动程序。
- 仿真器附带有所需的电缆。

3.1.2 LAN 接口

- 在 LAN 中连接仿真器之前，必须在仿真器中设置 IP 地址等信息。
- 可以通过路由器将 LAN 中的 PC7501 仿真器连接到与该 LAN 相连的另一个网络上的其他 PC7501。
- PC7501 仿真器使用市面上常见的 LAN 电缆（仅限 10BASE-T）。
- 主机和仿真器可以直接连接。

3.1.3 LPT 接口

- 该通信使用主机的并行（打印机）接口。
- 仿真器附带有所需的电缆。
- 支持包括“ECP”、“EPP”、“Byte”（字节）和“Nibble”（半字节）在内的四种通信方式。可以支持的通信方式取决于主机的 BIOS 设置。（这些通信方式并非总是可用，即使它们受 BIOS 支持也是如此。）

3.2 下载固件

Emulation Pod 需要在仿真器上下载固件，该固件对应于启动调试器时连接的仿真主机 (Emulation Pod)。

- 您更改了您的仿真主机。
- The firmware downloaded to the emulator is unknown one 下载到仿真器的固件是未知固件。
- 您首次设置了调试器。
- 您对仿真调试器进行了升级。

在给 PC4701 加电以后的两秒钟内，按系统复位开关，以建立维护模式。

该调试器在启动时会搜索下载到仿真器的固件的版本。同时，当下载到仿真器的固件是旧版本时，会设置一种模式，强制让该调试器下载固件。

如果仿真器设置为强制使调试器下载固件的模式，则当该调试器启动时，会打开以下对话框。

单击 [OK]（确定）按钮，下载固件。



注意事项

- 在 LAN 连接中的仿真器可以下载固件之前，必须首先在仿真器中注册 IP 地址等信息。

3.3 在仿真器启动之前进行设置

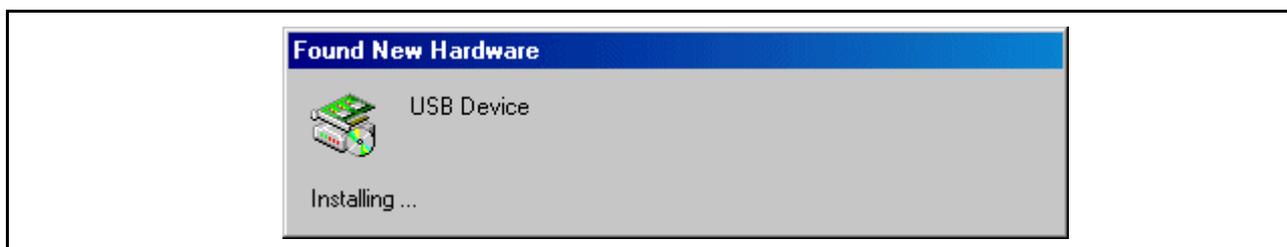
3.3.1 USB 通信

USB 设备的连接情况可以由 Windows 的即插即用功能检测到。已连接 USB 设备所需的设备驱动程序会自动进行安装。

3.3.1.1 安装 USB 设备驱动程序

已连接的 USB 设备可以由 Windows 的即插即用功能检测到。检测到设备以后，USB 设备驱动程序的安装向导会启动。下面说明安装 USB 设备驱动程序的过程。

- 使用 USB 电缆连接主机和仿真器。
- 将仿真器的通信接口开关设置到“USB”位置。然后接通仿真器的电源。
- 此时会出现下面显示的对话框。



继续按照向导的指示进行操作，此时会显示一个指定设置信息文件（inf 文件）的对话框。指定 musbdrv.inf 文件，该文件存储在该调试器安装目录下面。

注意事项

- 安装 USB 设备驱动程序之前，必须已安装您所用的调试器。首先安装该调试器。
- 安装 USB 设备驱动程序的用户需要管理员权限。
- 在安装过程中，会输出一条信息，指示找不到适当的设备驱动程序 musbdrv.sys。在这种情况下，指定存储在与 musbdrv.inf 文件相同的目录中的 musbdrv.sys。

3.3.2 LAN 通信

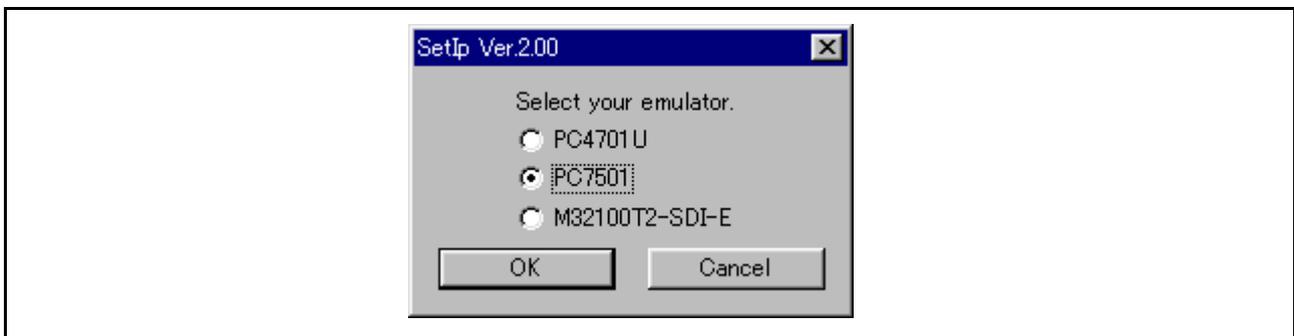
在 LAN 中连接仿真器之前，必须首先在仿真器中注册 IP 地址等信息。utilitydebugger 对于使用默认设置的仿真器，可以使用调试器附带的实用程序“setip.exe”在仿真器中设置 IP 地址等信息。

3.3.2.1 使用 SETIP.EXE 设置 LAN 接口

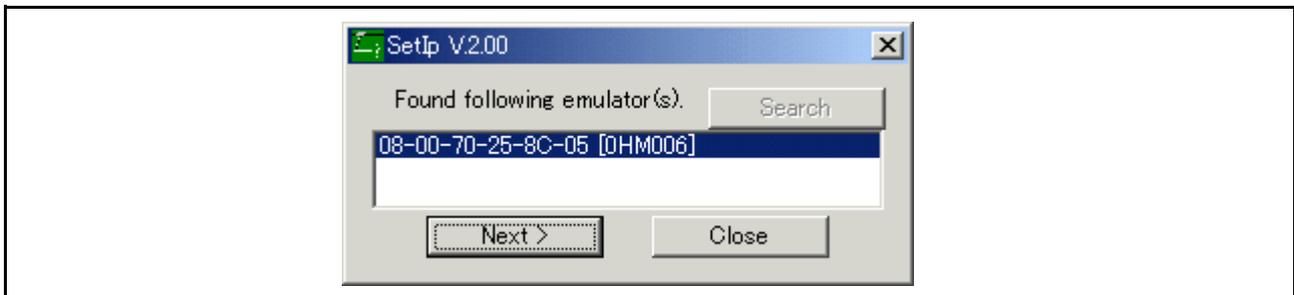
使用默认设置时，可以使用此调试器附带的实用程序“SETIP”在仿真器中设置 IP 地址等信息。SETIP 检测连接到相同网络的仿真器默认设置。SETIP 存储在该调试器安装目录下面。文件名为“setip.exe”。

若要在仿真器中注册 IP 地址，请按照下面说明的过程操作。

1. 使用 LAN 电缆将仿真器连接到主机所连的相同网络（相同子网）上。
2. 将仿真器的通信接口开关设置到“LAN”位置。然后接通仿真器的电源。
3. 启动 SETIP。SETIP 启动以后，会出现下面显示的对话框，选择所用的仿真器，然后单击 [OK]（确定）按钮。

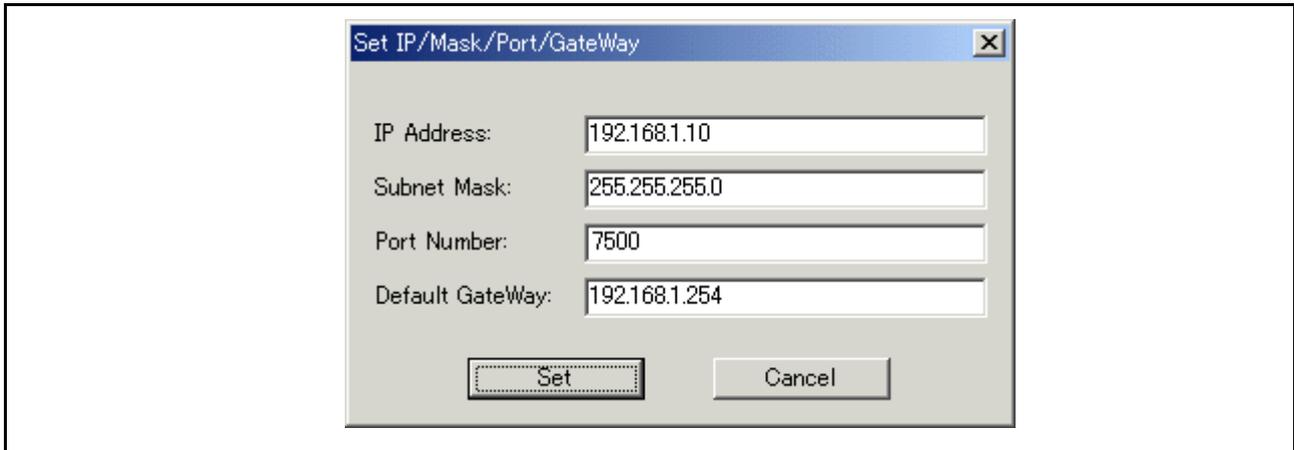


4. 单击 [OK]（确定）按钮，此时会显示有关连接到网络的仿真器的信息。（该信息由 MAC 地址以及随后的仿真器序列号组成。）



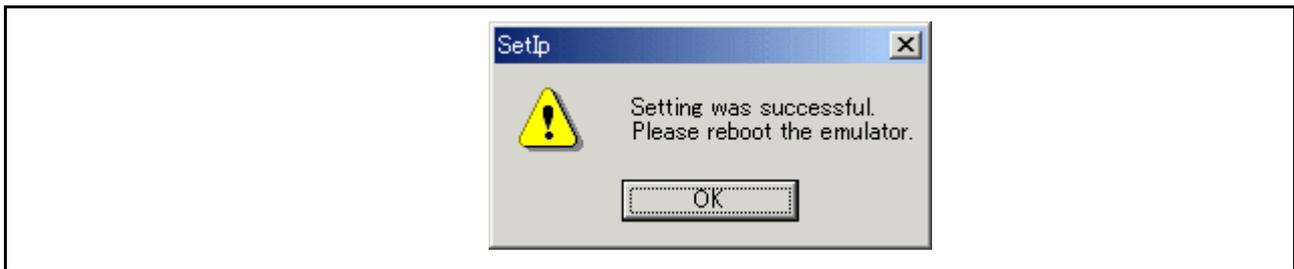
若要注册 IP 地址，请单击 [Next]（下一步）按钮。若要取消注册，请单击 [Close]（关闭）按钮。如果没有显示，请检查是否正确设置了通信接口开关；在暂时断开电源之后，再次接通电源。然后单击 [Search]（搜索）按钮。

- 单击 [Next]（下一步）按钮，此时会出现下面显示的对话框。设置 IP 地址、子网掩码、端口号和默认网关 IP 地址。在相同网络的相同子网掩码上使用仿真器时，可以省略默认网关 IP 地址。



使用任意 4 位数指定端口号。（启动调试器时需要输入该数字。）有关要指定的 IP 地址、子网掩码和默认网关内容的详细信息，请与网络管理员联系。

- 单击对话框上的 [Set]（设置）按钮。已设置的 IP 地址等信息会注册到仿真器中。正确注册以后，会出现下面显示的对话框。



检查对话框的内容以后，单击 [OK]（确定）按钮。

- 暂时断开仿真器电源，然后再次接通电源。仿真器再次加电以后，注册的 IP 地址就会生效。

注意事项

- 如果相同网络上连接了多个使用默认设置的仿真器，则只显示检测到的第一个仿真器。
- SETIP 无法检测已设置 IP 地址的仿真器。在这种情况下，请通过另一个通信接口连接到仿真器，然后从出现的 Init（初始化）对话框重新注册 IP 地址。有关如何从 Init（初始化）对话框设置 IP 地址的详细信息，请参阅“5.2.2 设置 LAN 接口”。

4 使用前的准备工作

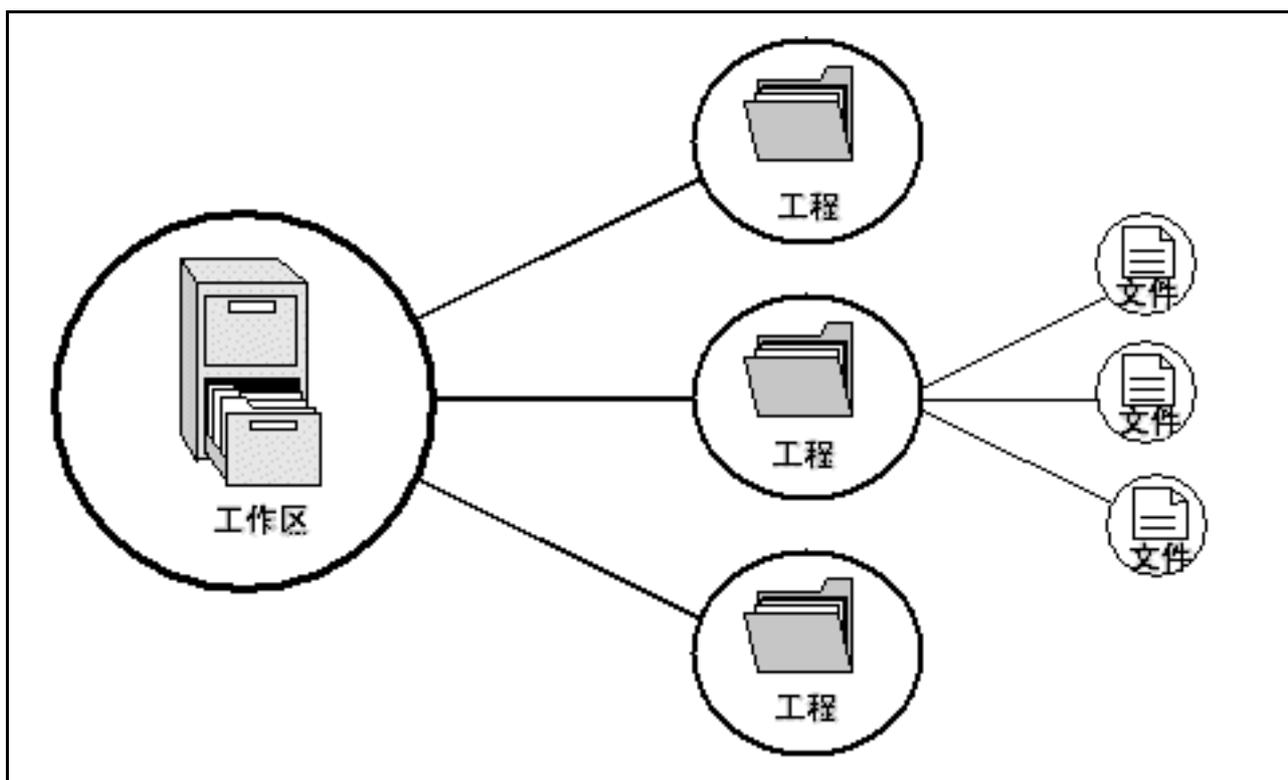
请运行 HEW 并连接仿真器。

此外，为了用该产品进行调试，需要创建工作区。

4.1 工作区、工程和文件

就像文字处理软件允许您创建和修改文档一样，该产品允许您创建和修改工作区。

可以将工作区视为工程的贮存箱；同样，可以将工程视为工程文件的贮存箱。因此，每个工作区可包含一个或多个工程，每个工程可包含一个或多个文件。

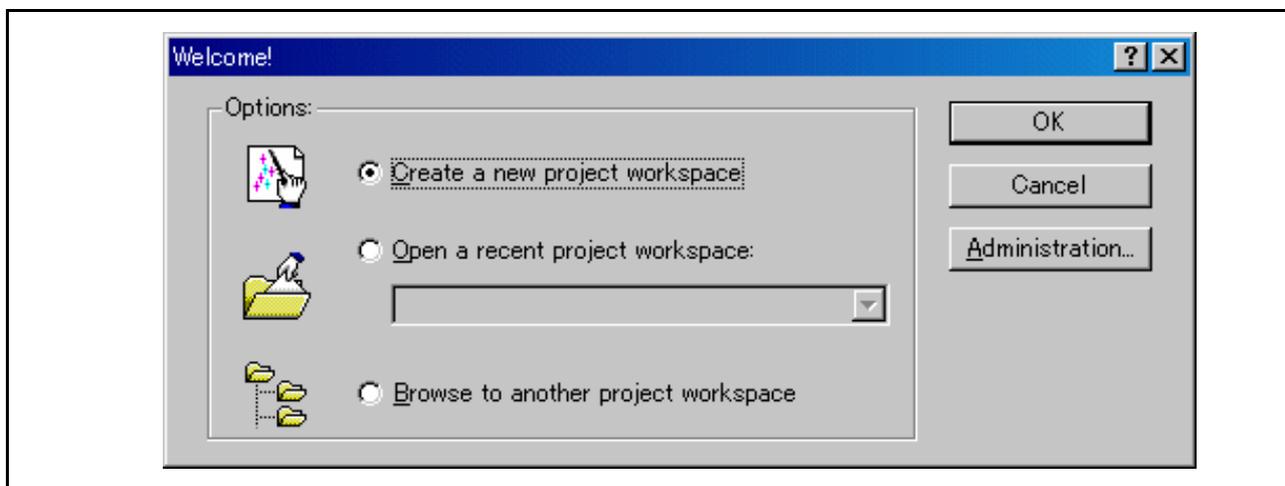


工作区允许您将相关工程组合在一起。例如，您可能有一个应用程序，该应用程序需要针对不同处理器进行创建，或者您可能正在同时开发一个应用程序和库。在工作区中，还可以分层连接工程，这意味着创建一个工程时，它的所有“子”工程都会首先进行创建。

不过，纯粹只是工作区并不能发挥很大的作用。要进行实际操作，需要将工程添加到工作区，然后再将文件添加到该工程。

4.2 启动 HEW

从 [Start] (中文系统: [开始]) 菜单的 [Programs] (中文系统: [程序]) 中激活 HEW。
此时显示 [Welcome!] (欢迎!) 对话框。



可以在该对话框中创建或显示工作区。

- [Create a new project workspace] (创建新的工程工作区) 单选按钮:
创建新的工作区。
- [Open a recent project workspace] (打开最近的工程工作区) 单选按钮:
使用现有工作区并显示所打开工作区的历史记录
- [Browse to another project workspace] (浏览到另一个工程工作区) 单选按钮:
使用现有工作区;
在所打开的工作区未保留历史记录时, 使用该单选按钮。

在选择现有工作区的情况下, 选择 [Open a recent project workspace] (打开最近的工程工作区) 或 [Browse to another project workspace] (浏览到另一个工程工作区) 单选按钮, 并选择工作区文件 (.hws)。

有关创建新工作区的方法, 请参阅以下章节。

参阅 “4.2.1 创建新的工作区 (使用工具链)”。

参阅 “4.2.2 创建新的工作区 (不使用工具链)”。

* 当使用该产品调试现有装入模块文件时, 该方法会创建一个工作区。

创建新工作区的方法取决于是否正在使用工具链。请注意, 此产品不包括工具链。在已安装适用于所用 CPU 的 C/C++ 编译器套件的环境中, 可以使用工具链。

有关详细信息, 请参阅 C/C++ 编译器套件附带的手册。

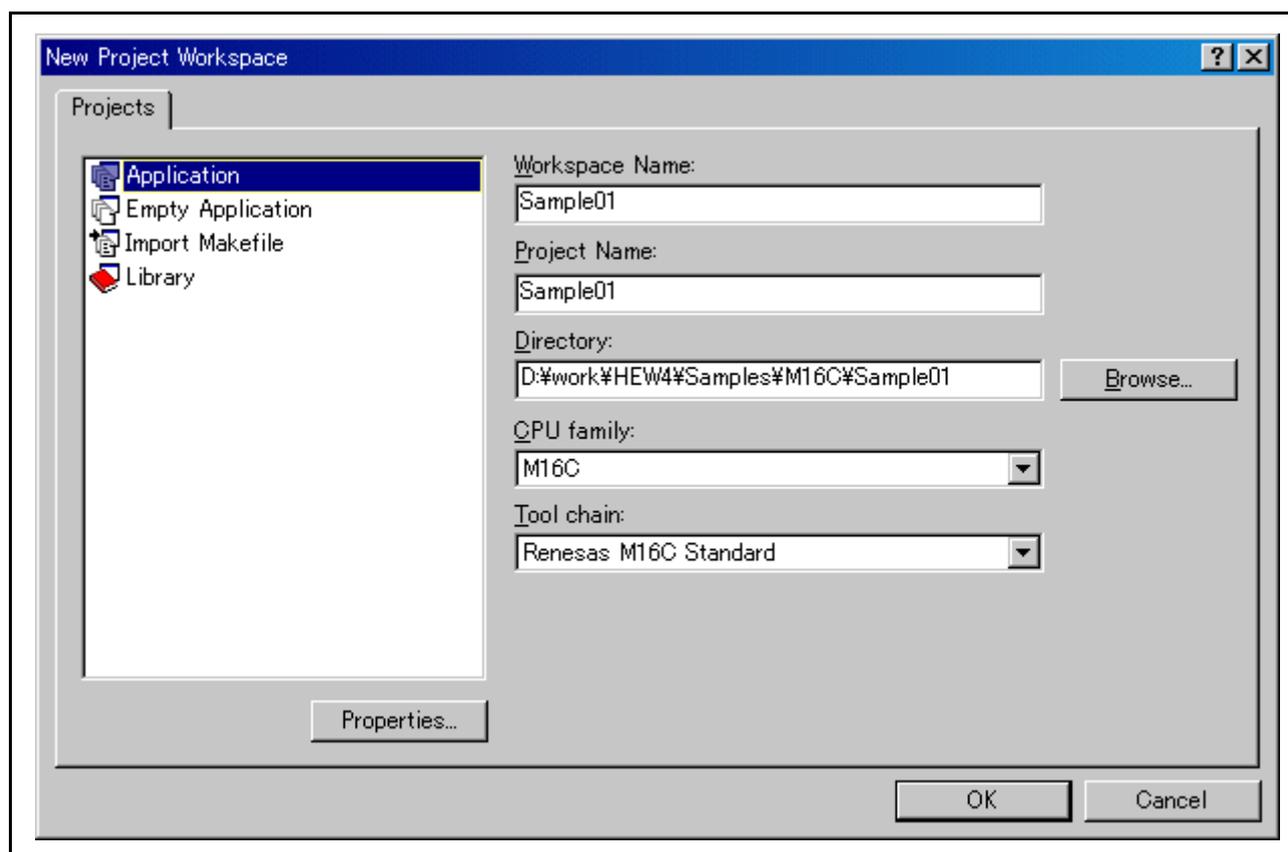
4.2.1 创建新的工作区（使用工具链）

4.2.1.1 步骤 1：创建新的工作区

在激活 HEW 时显示的 [Welcome!]（欢迎！）对话框中，选择 [Create a new project workspace]（创建新的工程工作区）单选按钮并单击 [OK]（确定）按钮。

此时开始创建新工作区。

此时显示以下对话框。

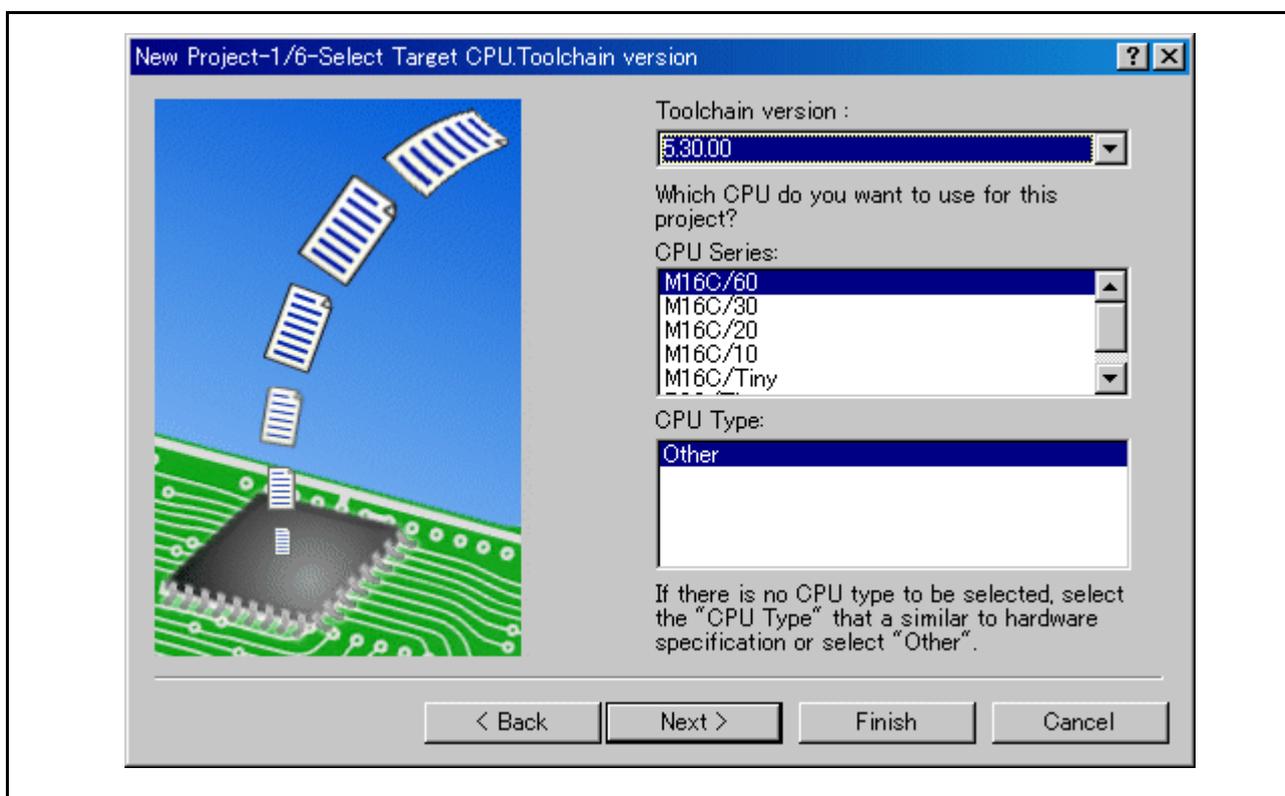


1. 选择目标 CPU 族
在 [CPU family]（CPU 族）组合框中，选择目标 CPU 族。
2. 选择目标工具链
在 [Tool chain]（工具链）组合框中，选择使用工具链时的目标工具链名称。
3. 选择工程类型
在 [Project type]（工程类型）列表框中，选择要使用的工程类型。
在本例中，选择“Application”（应用程序）。
（有关可以选择的工程类型的详细信息，请参阅 C/C++ 编译器套件附带的手册。）
4. 指定工作区名称和工程名称
 - 在 [Workspace Name]（工作区名称）编辑框中，输入新的工作区名称。
 - 在 [Project Name]（工程名称）编辑框中，输入工程名称。如果工程名称与工作区名称相同，则无需输入。
 - 在 [Directory]（目录）编辑框中，输入将在其中创建工作区的目录名称。单击 [Browse...]（浏览...）按钮可以选择目录。

设置以后，单击 [OK]（确定）按钮。

4.2.1.2 步骤 2: 设置工具链

此时会启动工程创建向导。



在这里设置以下内容。

- 工具链
- 实时操作系统（如果使用的话）的设置
- 启动文件、堆区、堆栈区等等的设置

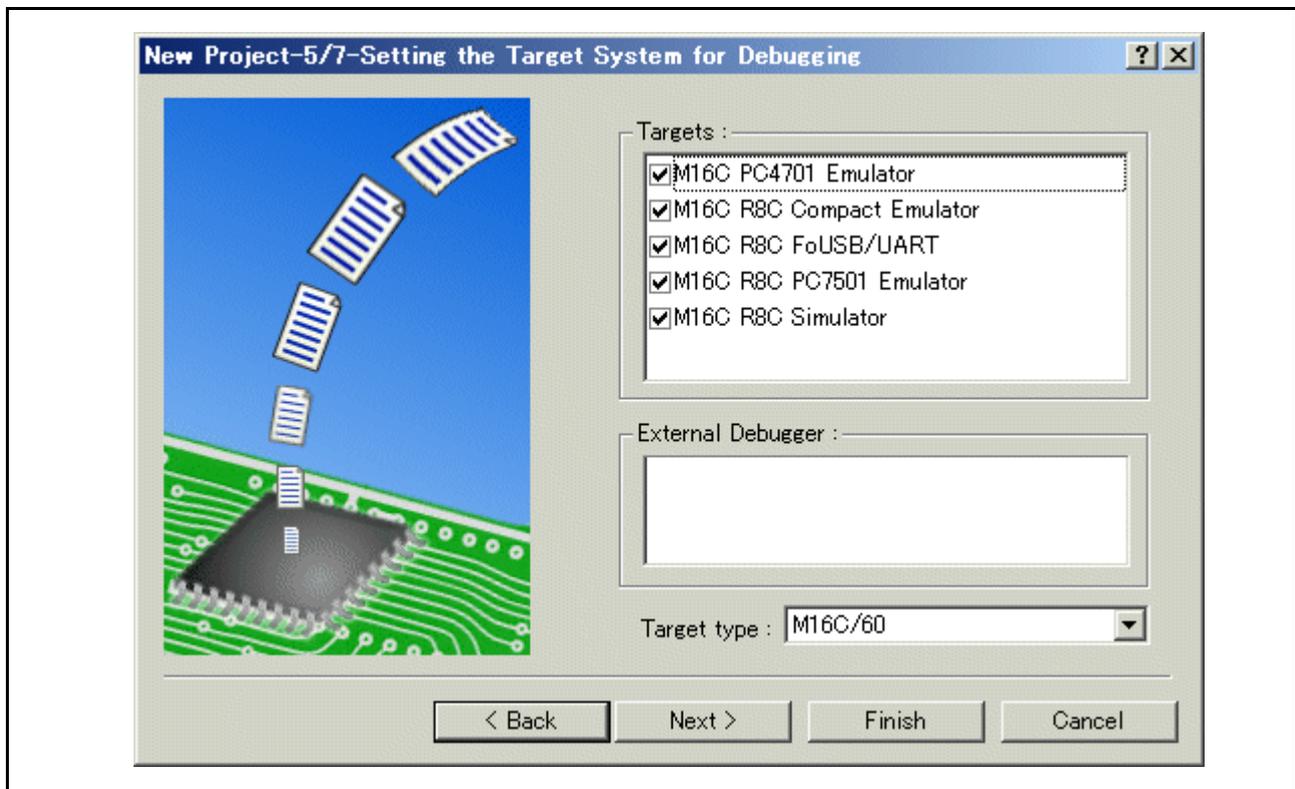
请设置所需的信息并单击 [Next]（下一步）按钮。

设置的内容会根据所用 C/C++ 编译器套件的不同而有所不同。有关设置内容的详细信息，请参阅 C/C++ 编译器套件附带的手册。

4.2.1.3 步骤 3: 设置目标平台

选择用于调试的目标系统（仿真器、模拟器）。

工具链设置完成以后，会显示以下对话框。



1. 选择目标类型

在 [Target type]（目标类型）列表框中，选择目标 CPU 类型。

2. 选择目标平台

在 [Targets]（目标）区域中，必须选择激活该调试器时所用会话文件的目标。

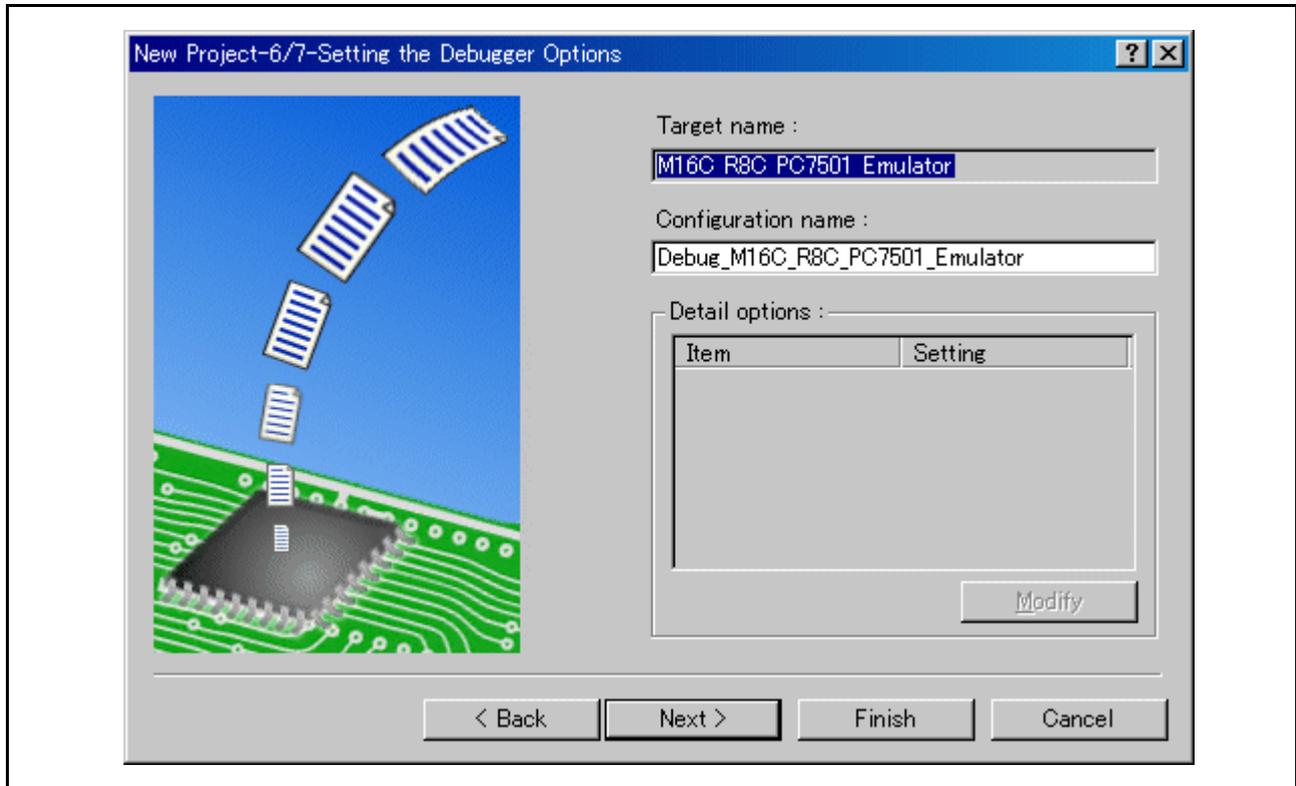
选中目标平台所对应的框。（根据需要选择其他目标。）

然后单击 [Next]（下一步）按钮。

4.2.1.4 步骤 4: 设置配置文件名

为选择的所有目标逐个设置配置文件名。

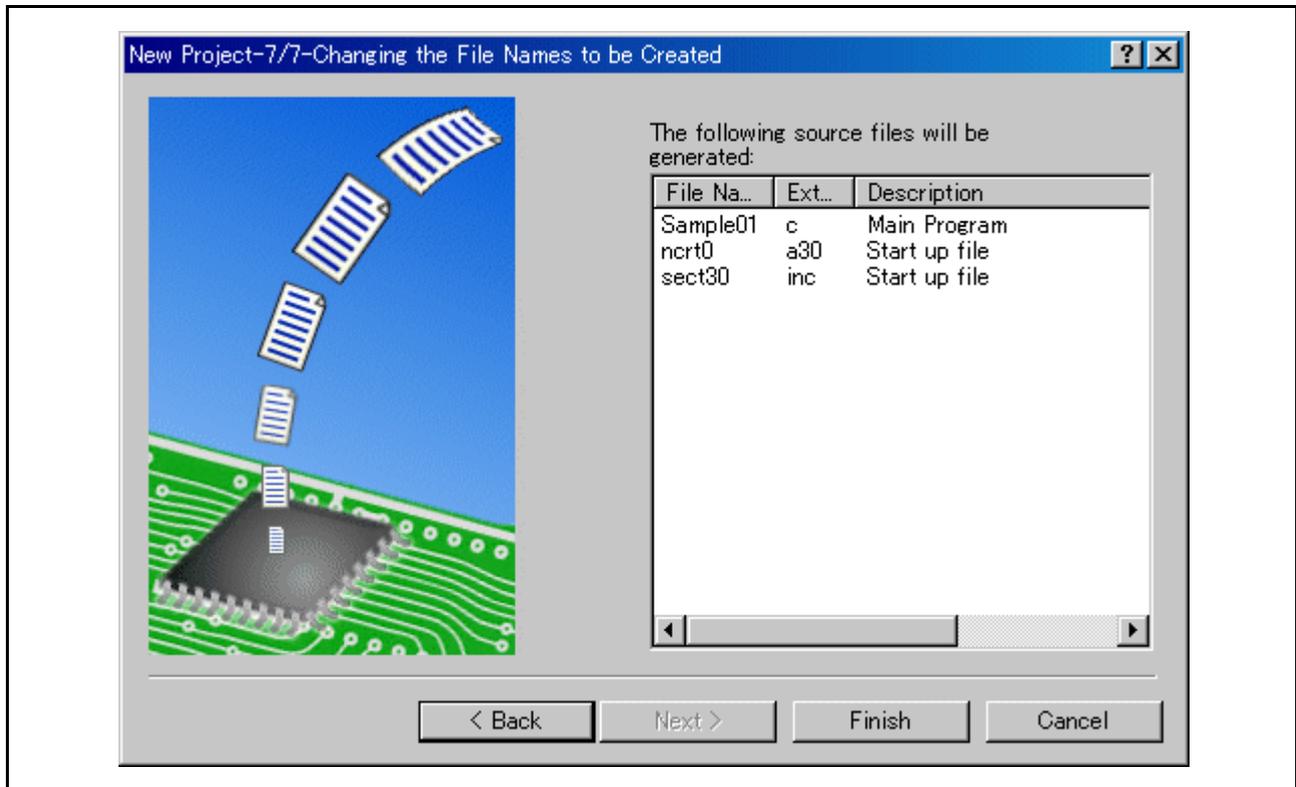
配置文件保存 HEW 的状态（目标仿真器、模拟器除外）。



默认名称已设置。如果不需要更改，请单击 [next]（下一步）按钮（使用默认名称）。

4.2.1.5 步骤 5: 检查创建的文件名

最后, 确认您创建的文件名。此时会显示将由 HEW 生成的文件; 如果您要更改文件名, 请选择并单击该文件名, 然后输入新名称。



仿真器设置到此结束。

按照屏幕上的指示退出工程生成器。

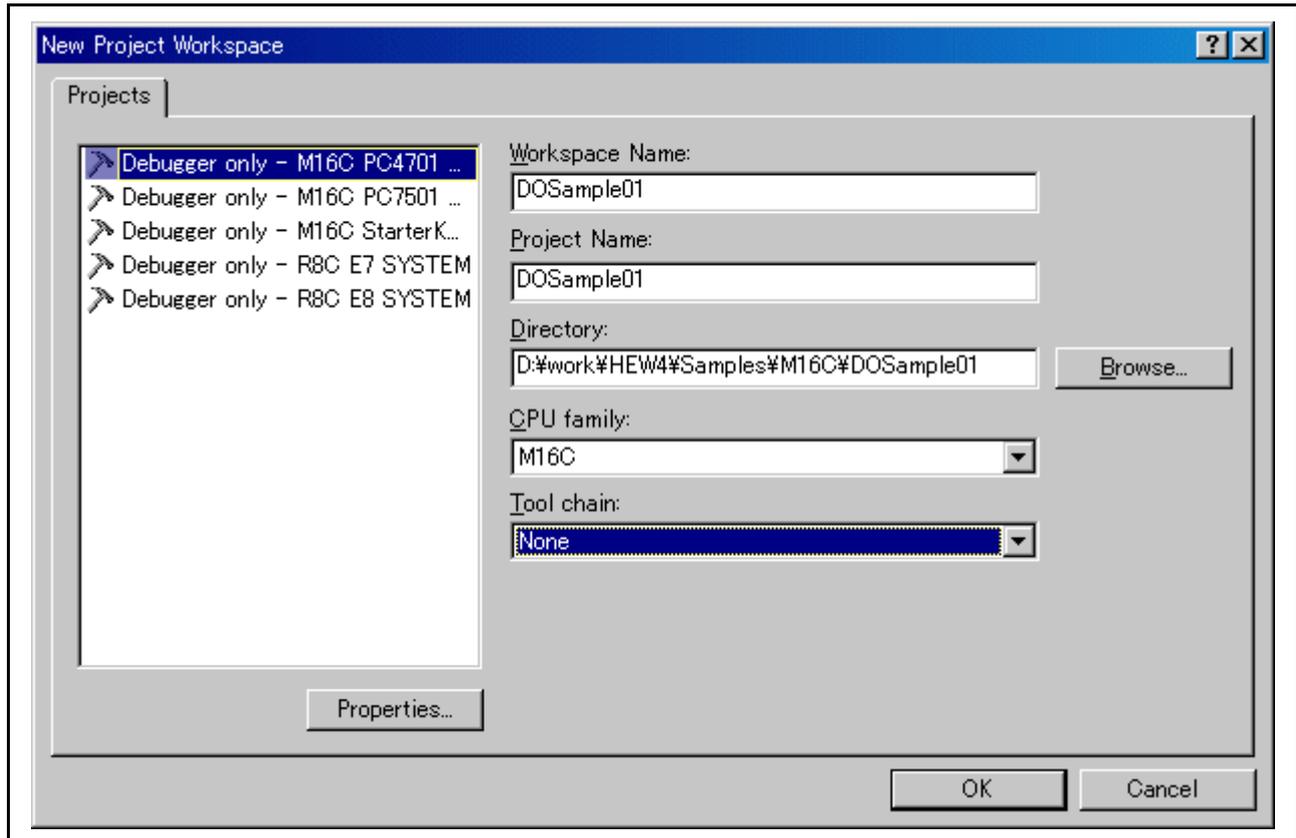
4.2.2 创建新的工作区（不使用工具链）

当使用该产品调试现有装入模块文件时，该方法会创建一个工作区。（即使未安装工具链，该方法也会创建一个工作区。）

4.2.2.1 创建新的工作区

在激活 HEW 时显示的 [Welcome!]（欢迎！）对话框中，选择 [Create a new project workspace]（创建新的工程工作区）单选按钮并单击 [OK]（确定）按钮。

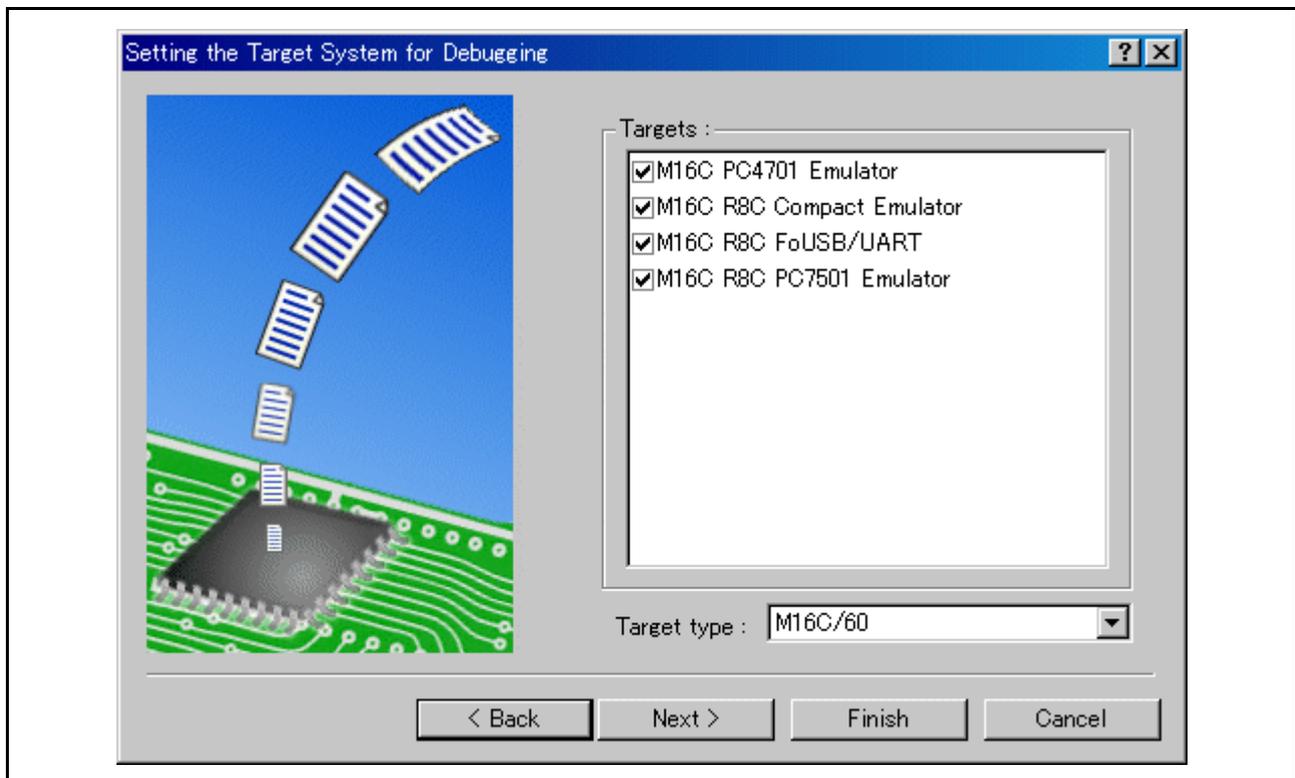
此时开始创建新工作区。此时显示以下对话框。



1. 选择目标 CPU 族
在 [CPU family]（CPU 族）组合框中，选择目标 CPU 族。
2. 选择目标工具链
在 [Tool chain]（工具链）组合框中，选择“None”（无）。在这种情况下，不使用工具链。
（如果未安装工具链，则该组合框中会显示固定信息。）
3. 选择工程类型
（如果不使用工具链，则 [Project Type]（工程类型）列表框上会显示“Debugger only - Target Name”（仅限调试器 - 目标名称）。选择该选项。（如果显示两个或更多工程类型，请选择其中的一个。）
4. 指定工作区名称和工程名称
 - 在 [Workspace Name]（工作区名称）编辑框中，输入新的工作区名称。
 - 在 [Project Name]（工程名称）编辑框中，输入工程名称。如果工程名称与工作区名称相同，则无需输入。
 - 在 [Directory]（目录）编辑框中，输入将在其中创建工作区的目录名称。单击 [Browse...]（浏览...）按钮可以选择目录。设置以后，单击 [OK]（确定）按钮。

4.2.2.2 步骤 2: 设置目标平台

选择用于调试的目标系统（仿真器、模拟器）。
此时会启动一个向导并显示以下对话框。



1. 选择目标类型

在 [Target type]（目标类型）列表框中，选择目标 CPU 类型。

2. 选择目标平台

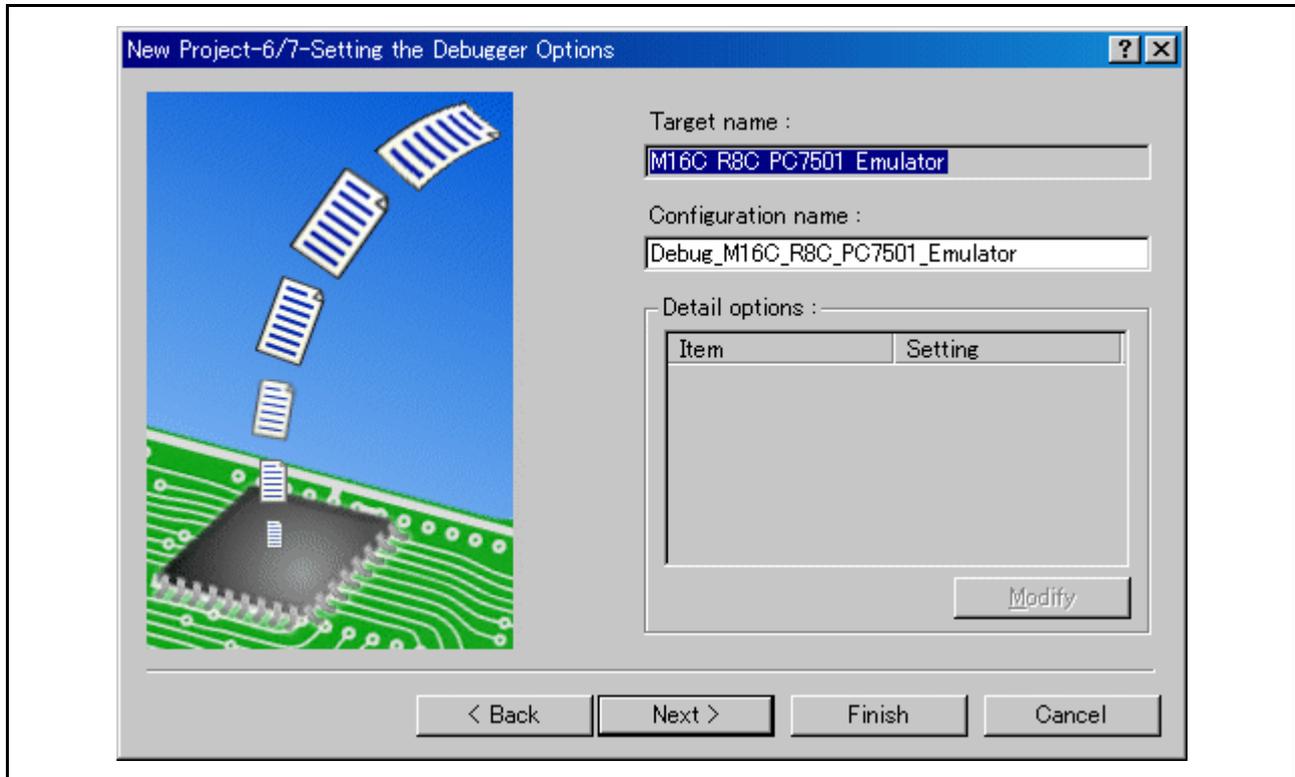
在 [Targets]（目标）区域中，必须选择激活该调试器时所用会话文件的目标。
选中目标平台所对应的框。（根据需要选择其他目标。）

然后单击 [Next]（下一步）按钮。

4.2.2.3 步骤 3: 设置配置文件名

为选择的所有目标逐个设置配置文件名。

配置文件保存 HEW 的状态（目标仿真器、模拟器除外）。



默认名称已设置。如果不需要更改，请单击 [next]（下一步）按钮（使用默认名称）。

仿真器设置到此结束。

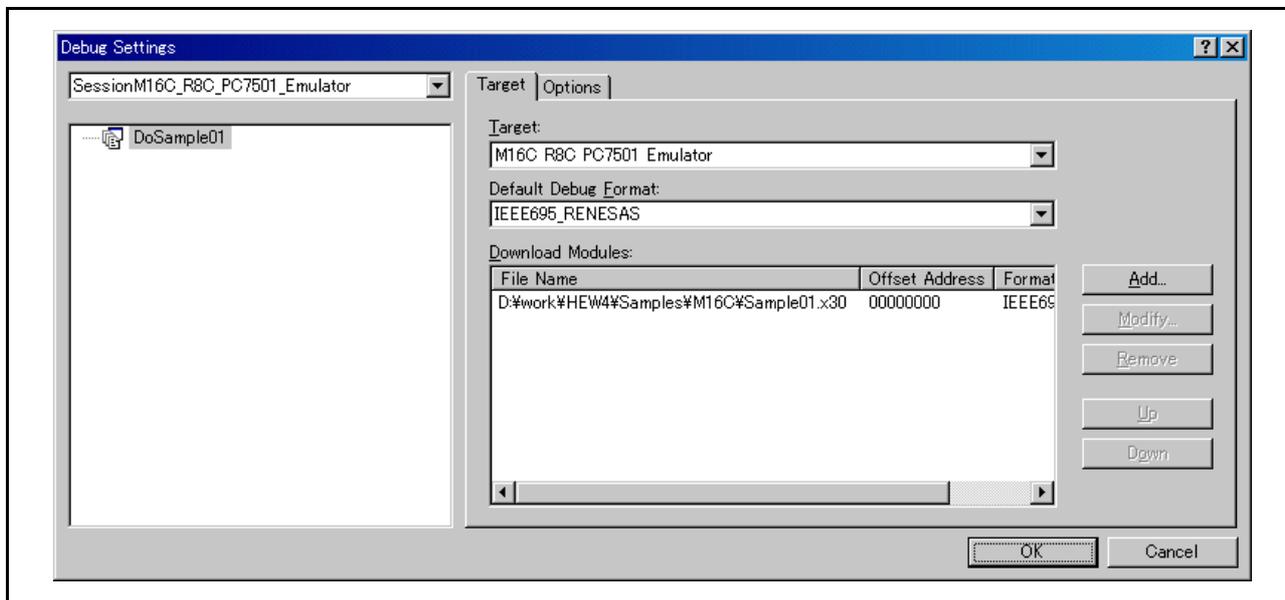
按照屏幕上的指示退出工程生成器。

此时还会显示调试器设置对话框。如果完成了仿真器的准备工作，请在该对话框中设置调试器并连接仿真器。

4.2.2.4 步骤 4: 注册要下载的装入模块

最后, 注册要使用的装入模块文件。

从 [Debug] (调试) 菜单中选择 [Debug Settings...] (调试设置...), 以打开 [Debug Settings] (调试设置) 对话框。

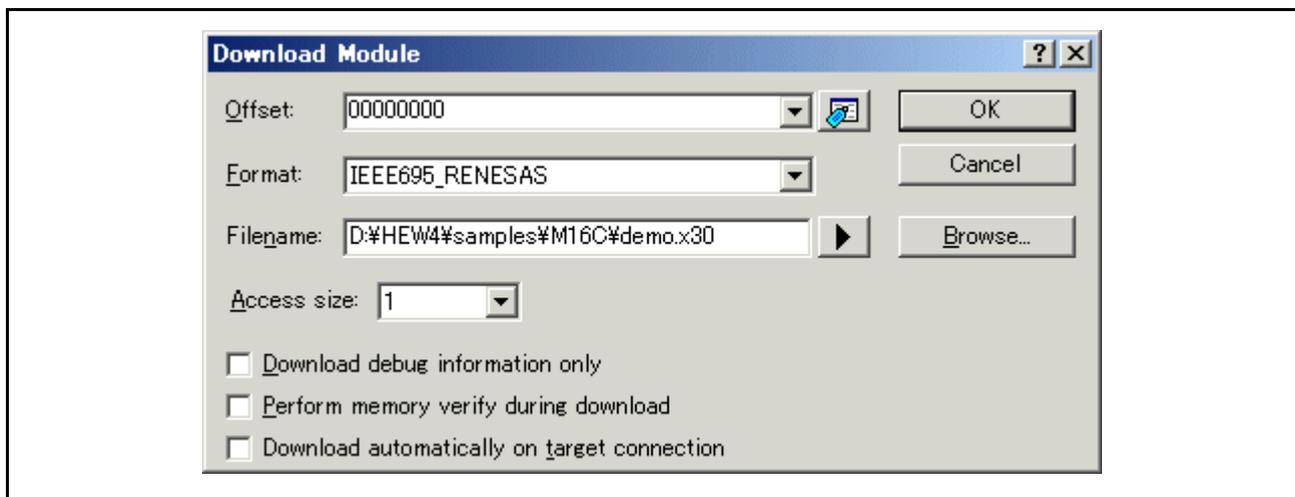


1. 在 [Target] (目标) 下拉式列表框中选择要连接的产品名称。
2. 在 [Default Debug Format] (默认调试格式) 下拉式列表框中选择要下载的装入模块的格式。

格式名称	内容
IEEE695_RENESAS	IEEE-695 格式文件 (使用瑞萨工具链时)
IEEE695_IAR	IEEE-695 格式文件 (使用 IAR 工具链时)
IEEE695_TASKING	IEEE-695 格式文件 (使用 Tasking 工具链时)
ELF/DWARF2	ELF/DWARF2 格式文件 (使用瑞萨工具链时)
ELF/DWARF2_IAR	ELF/DWARF2 格式文件 (使用 IAR 工具链时)
ELF/DWARF2_TASKING	ELF/DWARF2 格式文件 (使用 Tasking 工具链时)
ELF/DWARF2_KPIT	ELF/DWARF2 格式文件 (使用 KPIT 工具链时)

该调试器不支持目标格式, 这些格式未显示在下拉式列表中。

3. 然后在 [Download Modules]（下载模块）列表框中注册相应的下载模块。
可以在使用 [Add...]（添加...）按钮打开的对话框中指定下载模块。



- 在 [Format]（格式）编辑框中选择下载模块的格式。有关下载模块的格式名称，请参阅上面的表格。
- 在 [Filename]（文件名）编辑框中输入下载模块的完整路径和文件名。
- 在 [Access size]（存取大小）列表框中指定当前下载模块的存取大小。

然后单击 [OK]（确定）按钮。

注意事项

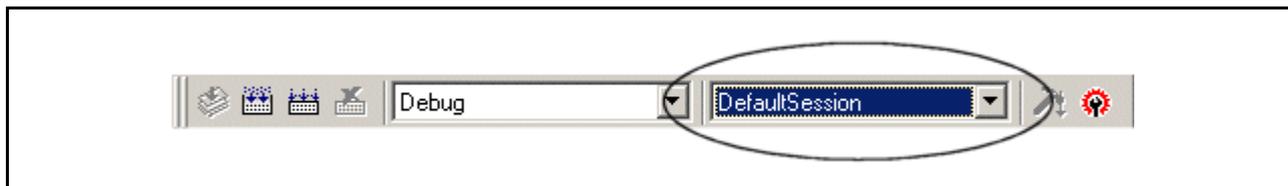
忽略“Offset”（偏移量）、“Access size”（存取大小）和“Perform memory verify during download”（下载期间执行存储器验证）。偏移量总是设置为 0，存取大小总是设置为 1，而验证则不起作用。

4.3 启动调试器

通过与仿真器相连接，可以启动调试器。

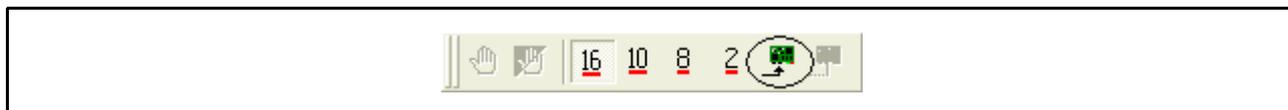
4.3.1 连接仿真器

只需将会话文件转换为已在其中注册仿真器使用设置的文件，即可连接仿真器。默认情况下，会创建会话文件。会话文件含有关于创建工程时所选目标的信息。在以下工具栏上圈定的列表框中，选择包括要连接的目标字符串的会话名称。



选择会话名称之后，会显示设置调试器的对话框，并且将连接仿真器。

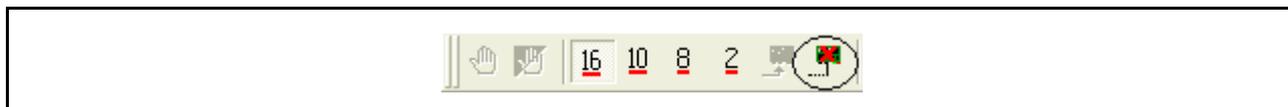
如果未显示该对话框，请从 [Debug]（调试）菜单中选择 [Connect]（连接）。



4.3.2 结束仿真器

使用以下方法，可以退出仿真器：

1. 选择“Disconnect”（断开）
从 [Debug]（调试）菜单中选择 [Disconnect]（断开）。



2. 选择“DefaultSession”（默认会话）
在列表框中选择连接仿真器时使用的“DefaultSession”（默认会话）。

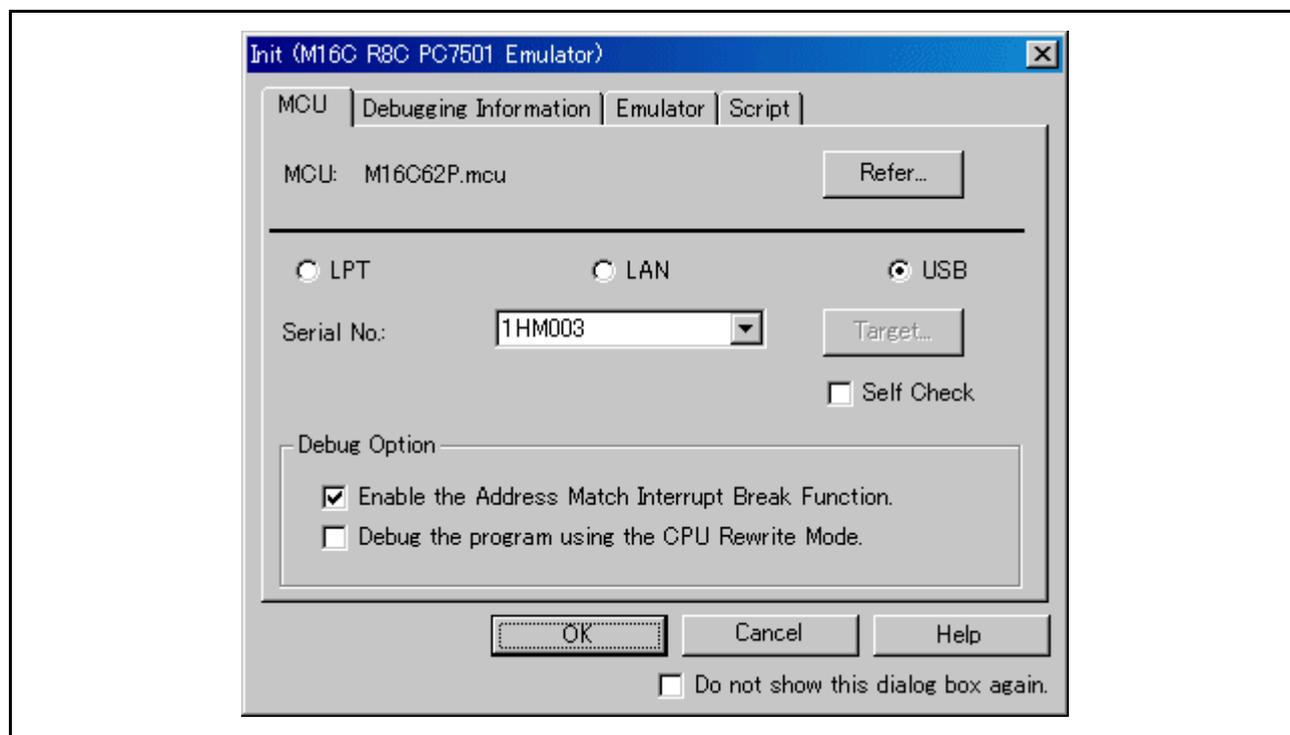
3. 退出 HEW
从 [File]（文件）菜单中选择 [Exit]（退出）。此时会结束 HEW。

退出仿真器时，会显示信息框，询问是否要保存会话。如果需要保存会话，请单击 [Yes]（是）按钮。如果不需要，请单击 [No]（否）按钮。

5 设置调试器

5.1 Init（初始化）对话框

[Init]（初始化）对话框用于设置调试器启动时需要设置的项目。从此对话框设置的内容在调试器下次启动时也是有效的。在此对话框中设置的数据在下次启动时保持有效。



此对话框上可用的选项卡因所用的产品而异。有关详细信息，请在下表中单击所需的选项卡名称。

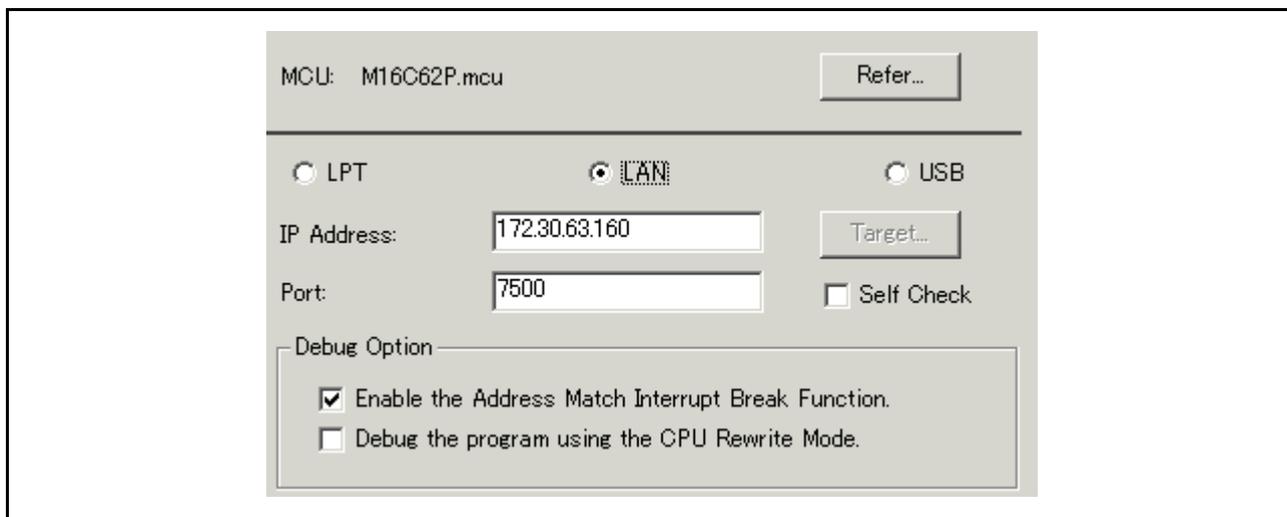
选项卡名称	产品名称	
	用于 M32C 的调试器	用于 M16C/R8C 的调试器
MCU	存在	存在
Debugging Information（调试信息）	存在	存在
Emulator（仿真器）	存在	存在
Script（脚本）	存在	存在

可以使用下列方法中的任意一种来打开 [Init]（初始化）对话框：

- 调试器启动以后，选择菜单 - [Setup]（设置）-> [Emulator]（仿真器）-> [System...]（系统...）。
- 在按住 Ctrl 键的同时启动调试器。

5.1.1 MCU 选项卡

指定的内容在下次启动时生效。



MCU: M16C62P.mcu Refer...

LPT LAN USB

IP Address: 172.30.63.160 Target...

Port: 7500 Self Check

Debug Option

Enable the Address Match Interrupt Break Function.

Debug the program using the CPU Rewrite Mode.

5.1.1.1 指定 MCU 文件



MCU: M30626.MCU Refer...

单击 [Refer]（引用）按钮。

此时会打开 [File Selection]（文件选择）对话框。指定相应的 MCU 文件。

- MCU 文件包含特定于目标 MCU 的信息。
- 指定的 MCU 文件显示在 [MCU] 选项卡的 [MCU] 区域中。

5.1.1.2 设置通信接口

显示的数据会根据指定通信接口的不同而有所不同。

可用通信接口会根据产品的不同而有所不同。

下面显示各个通信接口的设置。

- 参阅“5.2.1 设置 USB 接口”
- 参阅“5.2.2 设置 LAN 接口”
- 参阅“5.2.3 设置 LPT 接口”

5.1.1.3 执行自检

指定此选项可以在调试器启动时在仿真器上执行自检*。



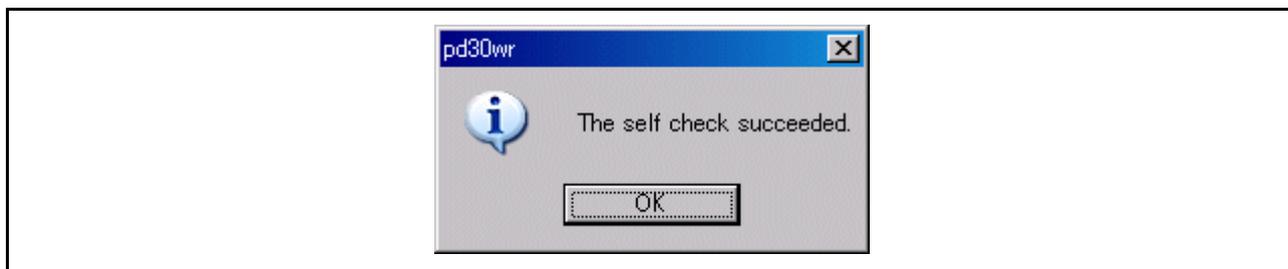
仅当要在启动时执行自检时，才有必要选中上面的复选框。出现以下情况时，请指定此选项：

- 无法下载固件
- 虽然成功下载了固件，但调试器无法启动
- MCU 出现异常或跟踪结果有错误，并且您希望检查仿真器是否正常操作。

选中复选框以关闭 [Init]（初始化）对话框。连接到仿真器并确认固件以后，调试器会在仿真器上立即启动自检。（自检大约花费 30 秒到 1 分钟时间。）

如果在此自检过程中发现错误，则调试器会显示错误的内容并结束。

自检正常结束时，会显示下面所示的对话框。单击 [OK]（确定）以后，调试器会在该状态下直接启动。

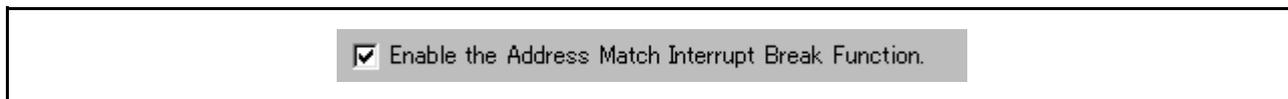


仅当调试器启动时此项指定才会生效。

* 自检功能将会检查仿真器内部电路板的存储器情况等等。有关自检功能的详细信息，请参阅仿真器的用户手册。

5.1.1.4 使用 / 不使用地址中断断点功能

指定是否要使用地址中断断点功能。



- 使用地址中断断点功能（默认）

选中上面显示的复选框。

在这种情况下，地址中断断点功能将由仿真器使用，并且不能在用户程序中使用。

- 不使用地址中断断点功能

取消选中上面显示的复选框。

在这种情况下，地址中断断点功能可以在用户程序中使用。

这里设置的内容只有在启动时才会反映出来。

5.1.1.5 使用 / 不使用看门狗定时器

指定是否要使用看门狗定时器。（默认情况下，不使用看门狗定时器。）
只有 M32C 调试器才有此项指定内容。

Debug the program using the Watchdog Timer.

在调试使用看门狗定时器的目标系统时，选中上面显示的复选框。

5.1.1.6 选择使用或不使用“CPU rewrite mode”（CPU 盖写模式）

指定是否要使用“CPU rewrite mode”（CPU 盖写模式）。（默认情况下，不使用“CPU rewrite mode”（CPU 盖写模式）。）

Debug the program using the CPU Rewrite Mode.

当您调试使用“CPU rewrite mode”（CPU 盖写模式）的目标系统时，请选中上面的复选框。
只有在启动调试器时才能设置或更改此项指定内容。

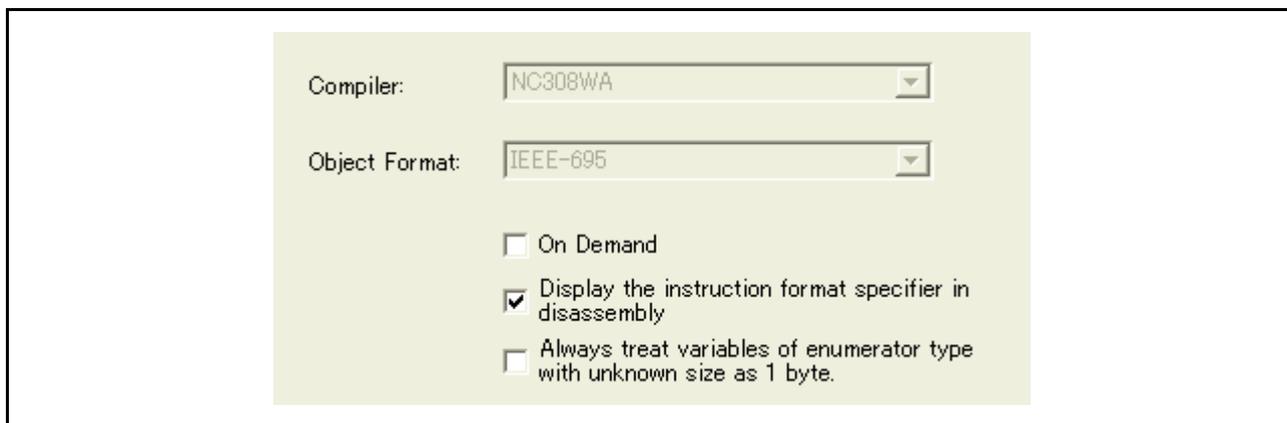
补充说明

如果允许在“CPU rewrite mode”（CPU 盖写模式）下调试，则存在以下限制：

- 无法设置地址匹配断点。
- 无法在内部 ROM 区设置软件断点。
- 无法在内部 ROM 区进行 COME 执行。

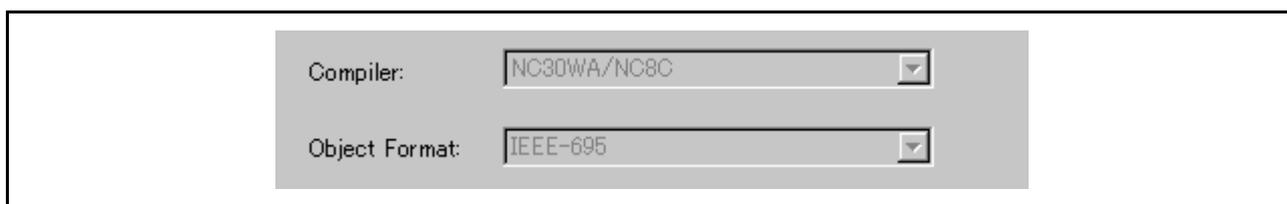
5.1.2 Debugging Information (调试信息) 选项卡

指定的内容在下次启动时生效。



5.1.2.1 显示使用的编译器及其目标格式

显示使用的编译器及其目标文件格式。



请在通过菜单 [Debug] (调试) -> [Debug Settings...] (调试设置...) 打开的对话框中, 指定使用的编译器及其目标文件格式。

5.1.2.2 指定调试信息的存储

调试信息的存储方法有两种: 存储器中存储和按需存储。

选择这两种方法中的一种。(默认情况下, 选择存储器中存储方法。)

若要选择按需存储方法, 请单击 [On Demand] (按需) 复选框。

指定的内容在下次下载时生效。

- 存储器中存储方法

调试信息存储在计算机的内部存储器中。

通常选择此方法。

- 按需存储方法

调试信息存储在计算机硬盘上的可重用临时文件中。

由于可以重用存储的调试信息, 因此下次下载相同装入模块时, 可以更快地进行下载。

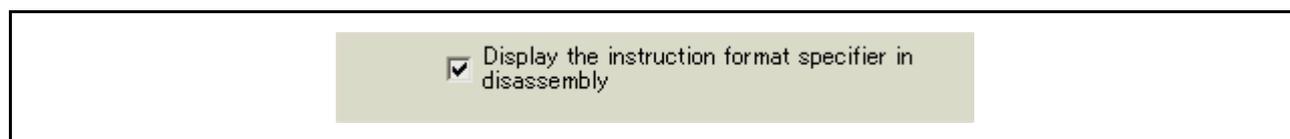
当下载调试信息需要花费很长时间时, 选择此方法较为合适, 因为 PC 存储器比装入模块文件大小要小。

注意

- 如果装入模块很大，则存储器中存储方法的效率可能较低，因为需要大量时间来进行下载。在这种情况下，请选择按需存储方法。
- 使用按需存储方法时，将在包含已下载的装入模块的文件夹中创建一个文件夹，用于存储可重用的临时文件。此文件夹的命名方法是在装入模块名称之前加上文字“~INDEX_”。例如，如果装入模块名称为“sample.abs”，则文件夹名称为“~INDEX_sample”。即使在退出调试器以后，此文件夹也不会被删除。

5.1.2.3 指定是否要显示指令格式说明符

指定是否要在反汇编显示中显示指令格式说明符。

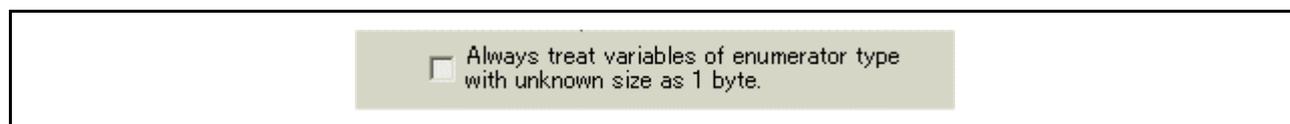


如果要显示指令格式说明符，请选中上面的复选框。
只有在启动调试器时才能设置或更改此项指定内容。

5.1.2.4 将枚举类型的大小视为 1 个字节

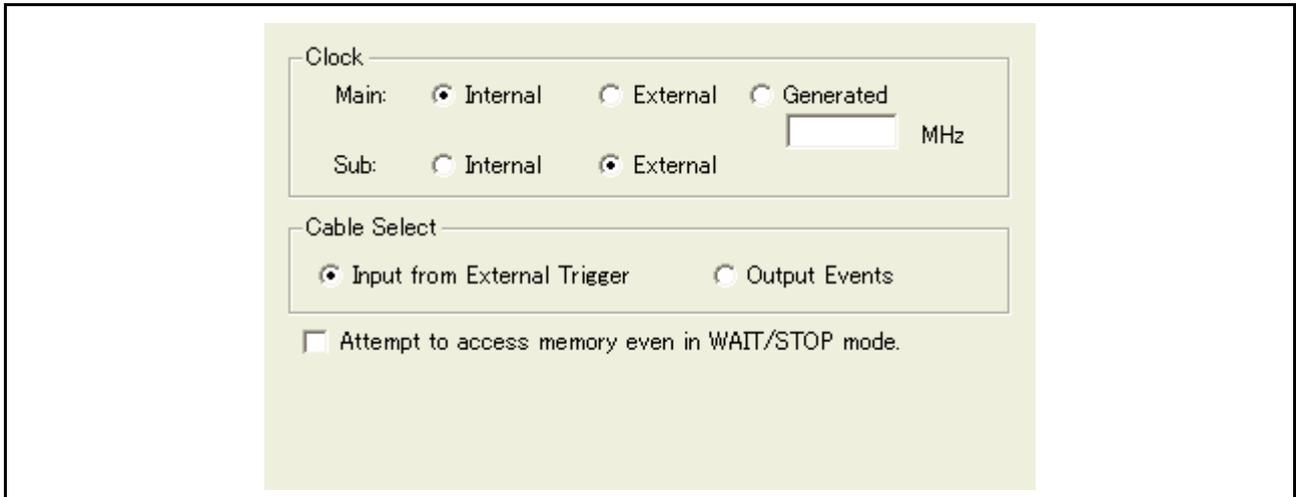
可以指定调试器是否将调试信息中大小未知的枚举类型的所有大小都视为 1 个字节。为了减少存储器的消耗，NC30 和 NC308 提供了一个选项，它可以将枚举符类型的大小视为 1 个字节（而不是与“int”相同的大小）。请注意，NC30 和 NC308 不会将枚举符类型的大小输出到调试信息中，调试器将该大小视为与“int”大小相同。

因此，在使用以上选项编译的目标程序中，可能无法正确引用枚举类型的值。此功能用于解决上面的问题。有关以上选项的详细信息，请参阅各个编译器的用户手册。



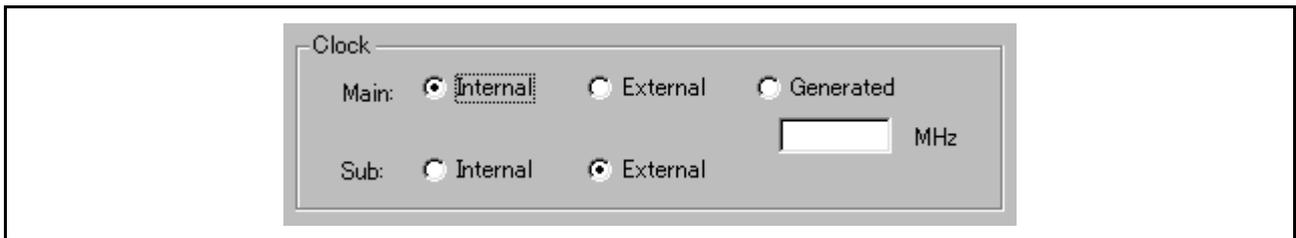
如果要将枚举类型的所有大小视为 1 个字节，请选中上面的复选框。为了反映此设置，需要再次装入调试信息。

5.1.3 Emulator（仿真器）选项卡



5.1.3.1 指定目标时钟

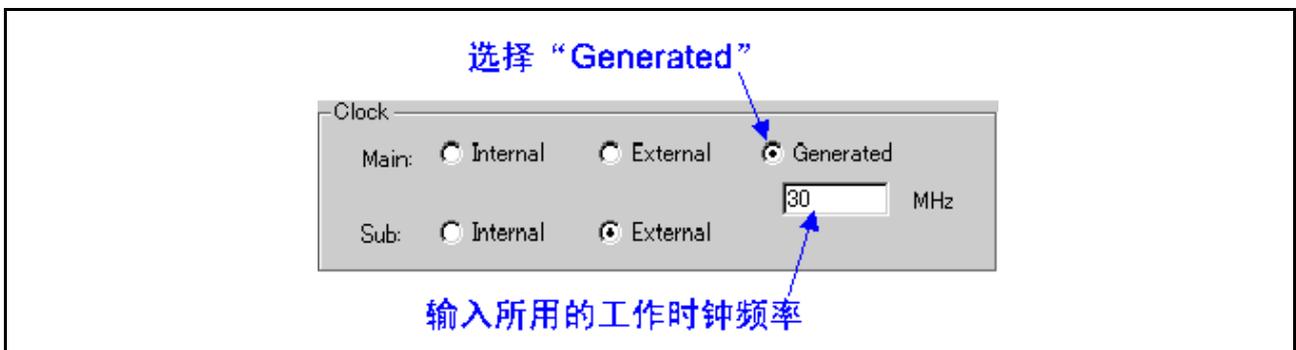
通过与目标单片机使用的时钟进行同步来更改设置。（[Internal]（内部）为默认设置。）



选择 [Internal]（内部）可设置内部时钟，选择 [External]（外部）可设置外部时钟。
指定的内容在下次启动时生效。

除了 [Internal]（内部）和 [External]（外部）之外，还可以选择主时钟作为在仿真器中根据指定时钟频率在内部产生的时钟（用户定义的时钟）。

若要使用用户定义的时钟，请选择 [Generated]（生成），并在 [Generated]（生成）频率输入区域中输入所用的时钟频率。



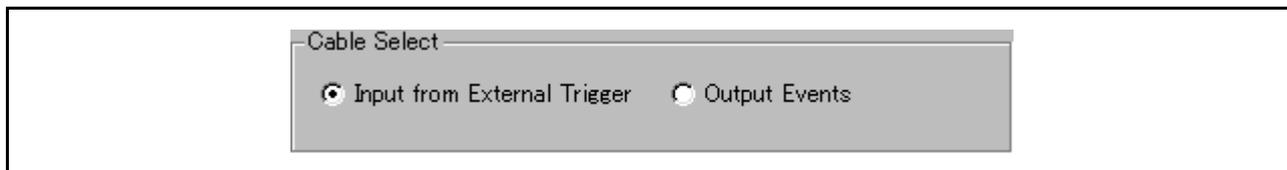
- 可输入的 [Generated]（生成）频率值范围为 1.0 MHz 到 99.9 MHz（增量为 0.1 MHz）。*
- 只有在启动调试器时才能设置或更改 [Generated]（生成）频率值。
- 除非在 [Generated]（生成）频率输入区域中设置了任何值，否则无法选择 [Generated]（生成）选项。

注意

* 时钟频率的精度为 +/-5%。我们建议，在安装谐振器或振荡器模块（其频率与实际使用的内部时钟匹配）之后进行最终计算。

5.1.3.2 选择事件输出 / 触发输入电缆

选择 PC7501 事件输出 / 触发输入电缆的输入 / 输出方向。



- 若要从电缆输入外部触发，请选择“Input from External Trigger”（从外部触发输入）。
- 若要将事件输出到电缆，请选择“Output Events”（输出事件）。

这里设置的内容只有在启动时才会反映出来。如果启动调试器之后在 [Init]（初始化）对话框中更改了设置，则更改不会反映出来。

请注意，启动时会选择“Input from External Trigger”（从外部触发输入）。（调试器上次启动时设置的内容没有任何作用。）

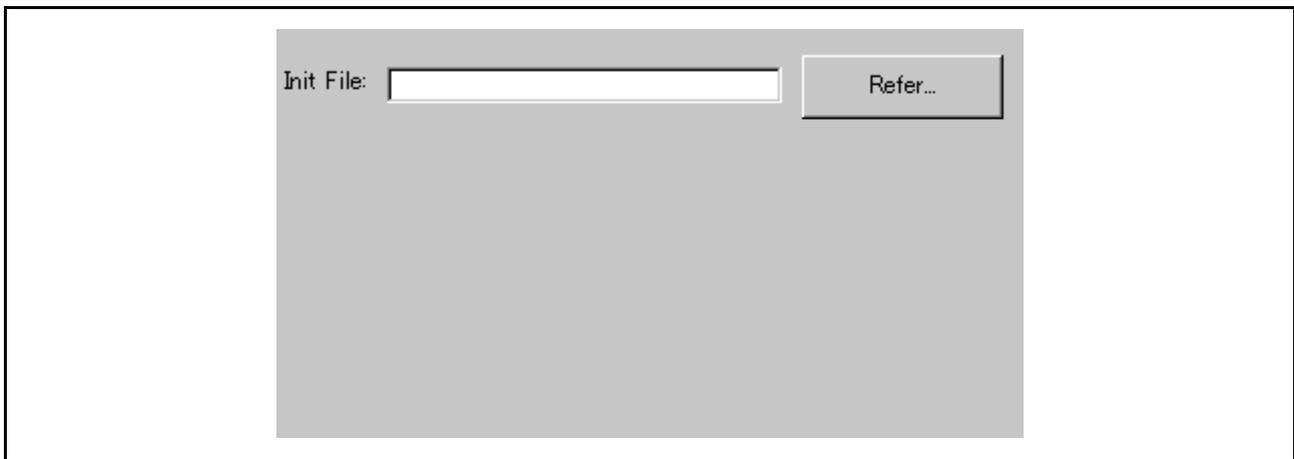
5.1.3.3 尝试在处于“WAIT/STOP mode”（等待 / 停止模式）期间存取存储器

如果 MCU 需要在处于“WAIT/STOP mode”（等待 / 停止模式）期间存取存储器，请选中此项。

选中此项后，调试器将等待大约 5 秒时间，直到 MCU 从“WAIT/STOP mode”（等待 / 停止模式）返回，然后尝试存取存储器。如果 MCU 在此期间保持“WAIT/STOP mode”（等待 / 停止模式），则操作将会出错。如果未选中此项，调试器在未真正存取存储器的情况下就会出错。

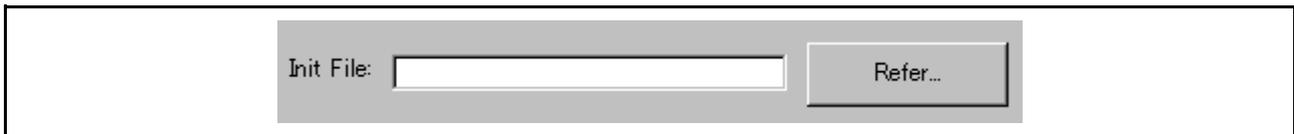
5.1.4 Script（脚本）选项卡

指定的内容在下次启动时生效。



5.1.4.1 自动执行脚本命令

若要在调试器启动时自动执行脚本命令，请单击 [Refer]（引用）按钮，指定要执行的脚本文件。



单击 [Refer]（引用）按钮后，会打开 [File Selection]（文件选择）对话框。

指定的脚本文件会显示在“Init File:”（初始化文件：）字段中。

若要禁止自动执行脚本命令，请擦除“Init File:”（初始化文件：）字段中显示的字符串。

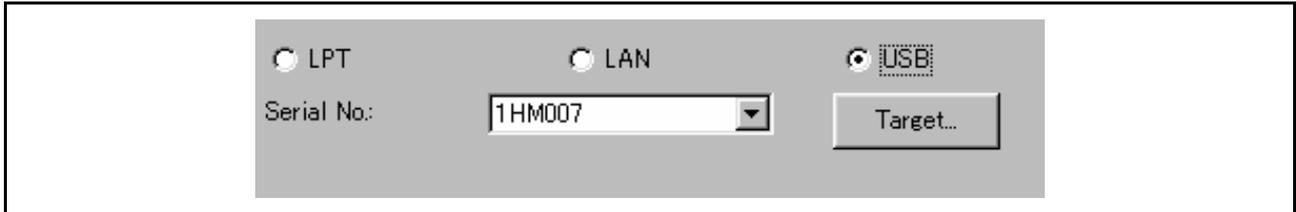
5.2 设置通信接口

5.2.1 设置 USB 接口

USB 通信使用个人计算机的 USB 接口。它符合 USB 1.1 标准。

执行 USB 通信之前，计算机必须已安装专用设备驱动程序。有关如何安装 USB 设备驱动程序的详细信息，请参阅“3.3.1.1 安装 USB 设备驱动程序”。

若要通过 USB 通信进行连接，请单击 [MCU] 选项卡上的“USB”单选按钮。



当前 USB 连接的仿真器列在 [Serial No.]（序列号）区域中。选择要连接的仿真器的序列号。

5.2.2 设置 LAN 接口

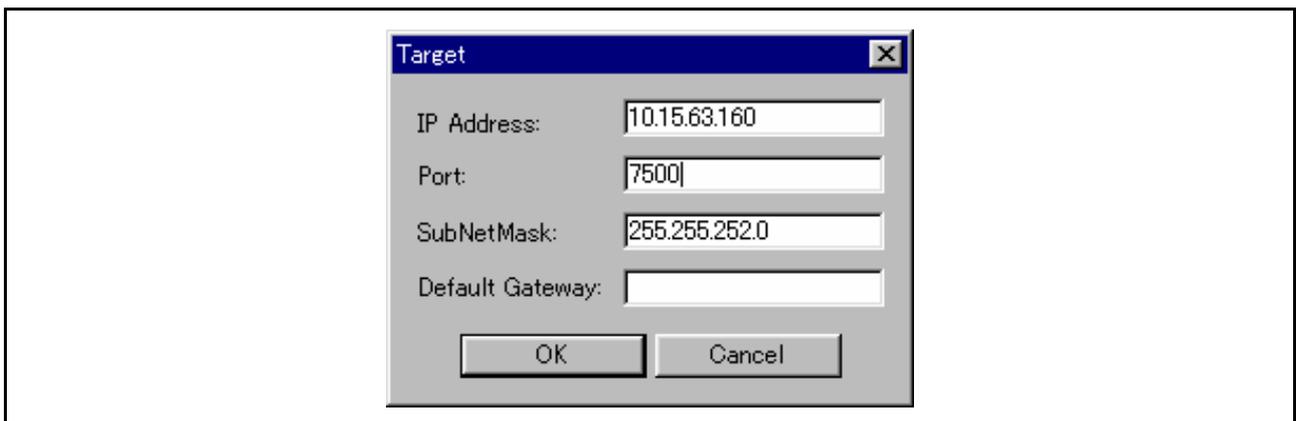
LAN 通信使用个人计算机的 LAN 接口。

使用 LAN 之前，必须向仿真器本身注册仿真器 IP 地址、端口号和子网掩码（否则将无法使用 LAN）。

然后设置 LAN 通信。

5.2.2.1 设置 IP 地址和子网掩码

使用其他通信方法启动调试器。调试器启动以后，选择菜单 - [Setup]（设置）-> [Emulator]（仿真器）-> [System...]（系统...），打开 [Init]（初始化）对话框。然后单击 [MCU] 选项卡的 [Target]（目标）按钮。此时会打开 [Target]（目标）对话框。



在 [IP Address] (IP 地址) 字段中指定仿真器 IP 地址, 在 [Port] (端口) 字段中指定端口号, 在 [SubNetMask] (子网掩码) 字段中指定子网掩码。(必须预先在网络环境中注册仿真器 IP 地址。)

使用 PC7501 或 PC4701U 时, [Default Gateway] (默认网关) 区域将会生效。请指定默认网关的 IP 地址。在相同网络的相同子网掩码上使用 PC7501 或 PC4701U 时, 可以省略默认网关的 IP 地址。

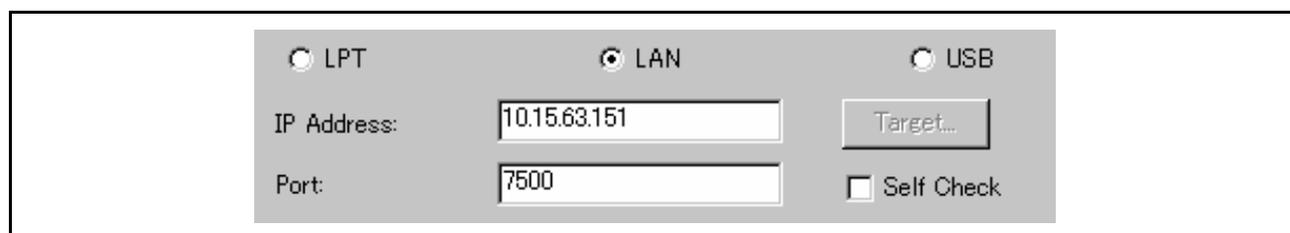
- 使用十进制数按字节指定 IP 地址、子网掩码和默认网关, 每 4 个字节以句点分隔。有关 IP 地址和子网掩码的详细信息, 请咨询网络管理员。
- [Port] (端口) 字段中设置的端口号用于标识 LAN (TCP/IP) 通信中服务器 (仿真器) 的通信进程。使用十六进制数指定仿真器中设置的端口号。(请勿添加表示基数的前缀。)

在 [Target] (目标) 对话框中单击 [OK] (确定) 按钮。[Target] (目标) 对话框将会关闭, 然后再次出现 [Init] (初始化) 对话框。单击 [OK] (确定) 按钮。

然后退出调试器。

5.2.2.2 设置 LAN 接口

若要设置 LAN 通信, 请单击 [Init] (初始化) 对话框中 [MCU] 选项卡下面的“LAN”单选按钮。设置如下图所示。



在 IP 地址字段中指定已连接仿真器的 IP 地址。

使用十进制数以字节为单位指定 IP 地址。每 4 个字节以句点分隔。端口号表示 LAN (TCP/IP) 上服务器 (仿真器) 通信进程的 ID。

使用十六进制数 (请勿填写表示基数的前缀) 指定仿真器上设置的端口号。

5.2.2.3 通过 1 个耦合器与仿真器建立 LAN 连接

使用 10BASE-T 交叉转换电缆 (市面有售), 可以通过 LAN (TCP/IP) 将仿真器连接到插在 PC 中的 LAN 卡 (市面有售)。在这种情况下, 不需要集线器。

10BASE-T 交叉转换电缆将仿真器附带的直连 LAN 电缆的 10BASE-T 公连接器转换为交叉 LAN 电缆的公连接器。

将交叉转换电缆连接到与仿真器相连的直连 LAN 电缆的 10BASE-T 公连接器; 然后, 将交叉转换电缆的公连接器连接到 LAN 卡。

可以按照与普通通信相同的设置方式来设置 LAN 通信。

5.2.3 设置 LPT 接口

LPT 通信使用个人计算机的并行接口（打印机接口）。

若要设置 LPT 通信，请单击 [Init]（初始化）对话框中 [MCU] 选项卡下面的“LPT”单选按钮。设置如下图所示。

在 [Type]（类型）字段中指定数据传送模式。

- LPT 接口有四种数据传送模式：[Nibble]（半字节）、[Byte]（字节）、[ECP] 和 [EPP]。这些模式在 IEEE-1284 标准中有说明。可以使用的模式取决于与调试器相连的 PC。
- 选择 [AUTO]（自动）时，调试器在启动时检测 LPT 接口，并且选择一种可以使用的、能够自动提供较佳性能的模式。在某些 PC 中，调试器无法检测最合适的模式。如果无法检测，则检查 PC 的 LPT 接口具有的模式，并且从 [Nibble]（半字节）、[Byte]（字节）、[ECP] 或 [EPP] 中选择一种。
- 启动 PC 的 BIOS 设置程序，检查可以使用的模式。如何启动和使用 BIOS 设置程序取决于各个 PC，因此请参阅 PC 手册。

BIOS 设置程序的显示内容	通信方式
SPP, Standard Parallel Port, Output Only	[Nibble]（半字节）
Bidirectional, Bi-directional	[Byte]（字节）
ECP, Extended Capabilities Port	[ECP]
EPP, Enhanced Parallel Port	[EPP]

并行端口基址字段中显示的地址是 I/O 地址。

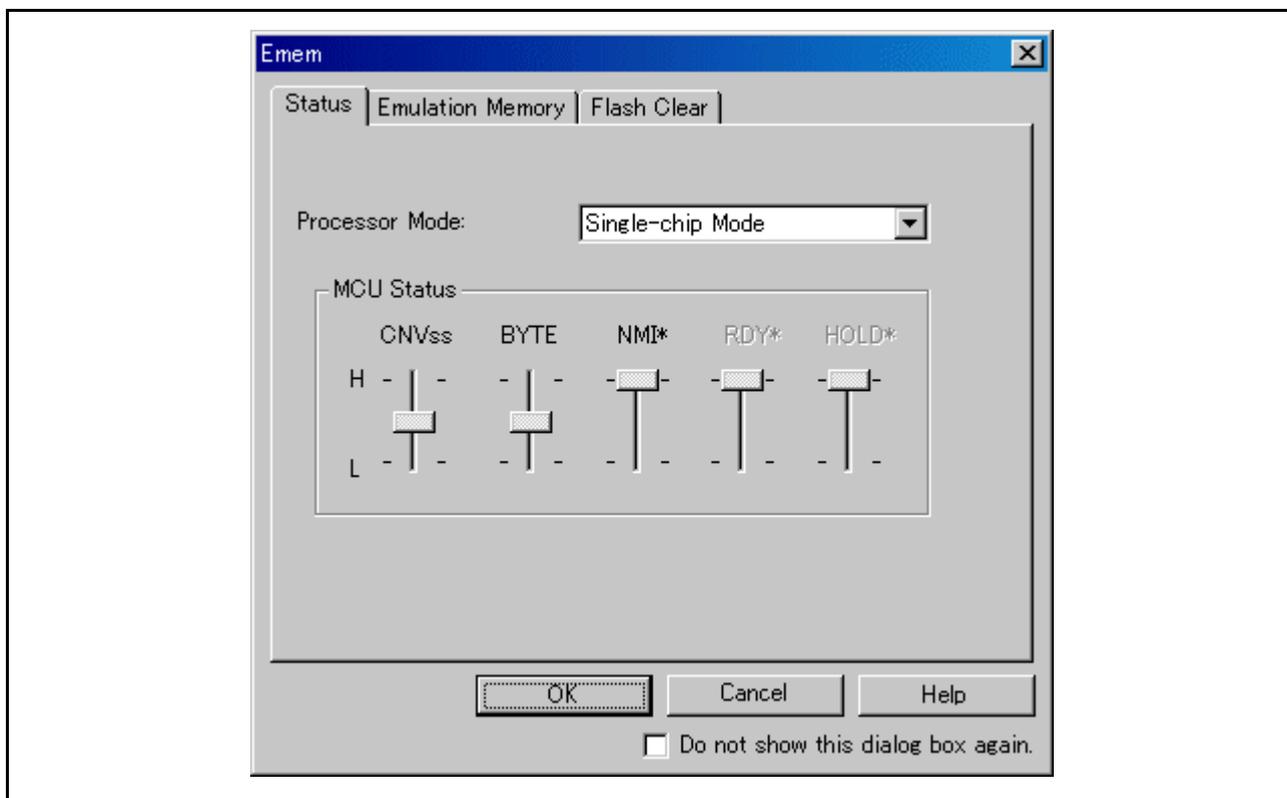
在 [I/O Address]（I/O 地址）字段中指定 BIOS 设置程序中设置的 I/O 地址。（可以指定以下地址）

- 378h
- 278h

5.3 设置 M32C 使用的调试器

5.3.1 Emem 对话框

在 [Emem] 对话框中，设置有关用户目标的信息。关闭 [Init]（初始化）对话框以后会打开 [Emem] 对话框。



此对话框上可用的选项卡因所用的产品而异。有关详细信息，请在下表中单击所需的选项卡名称。

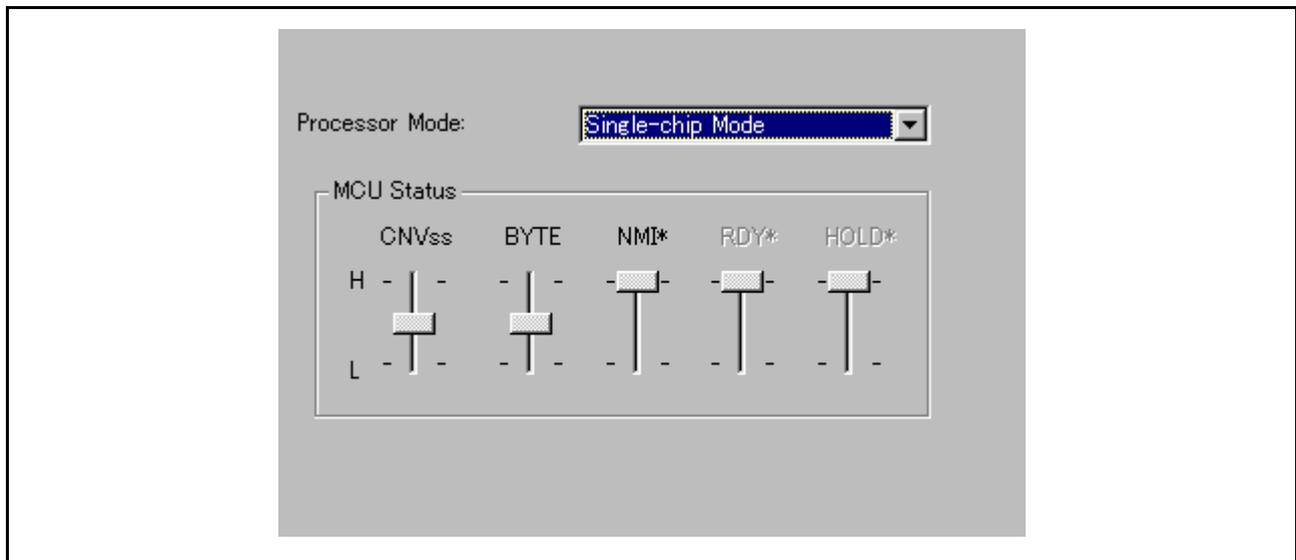
选项卡名称	内容
Status（状态）	指定处理器模式。
Emulation Memory（仿真存储）	指定仿真存储区。
Flash Clear（闪存清除）	指定是否要清除 MCU 内部闪存 ROM 的内容。

若要在下次启动调试器时让 [Emem] 对话框保持关闭状态，请在 [Emem] 对话框底部选择“Next Hide”（下次隐藏）。可以使用下列方法中的任意一种来打开 [Emem] 对话框：

- 调试器启动以后，选择菜单 - [Setup]（设置）->[Emulator]（仿真器）->[Target...]（目标...）。

5.3.1.1 Status（状态）选项卡

指定的内容在下次启动时生效。



5.3.1.1.1 选择处理器模式

指定目标系统的处理器模式。

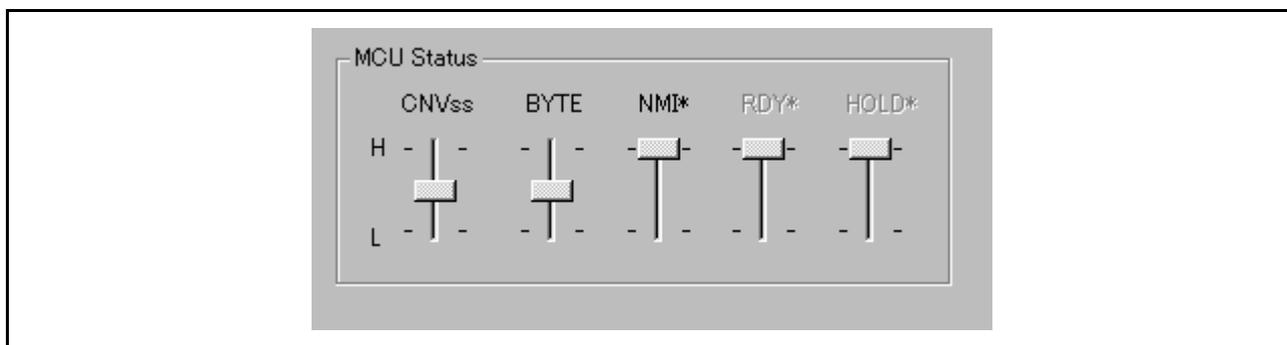


可以指定以下任意一项。

- Single-chip Mode
单片模式
- Memory Expansion 8 Bit
存储器扩展模式（8 位总线宽度）
- Memory Expansion 16 Bit
存储器扩展模式（16 位总线宽度）
- Microprocessor 8 Bit
微处理器模式（8 位总线宽度）
- Microprocessor 16 Bit
微处理器模式（16 位总线宽度）

5.3.1.1.2 检查 MCU 状态

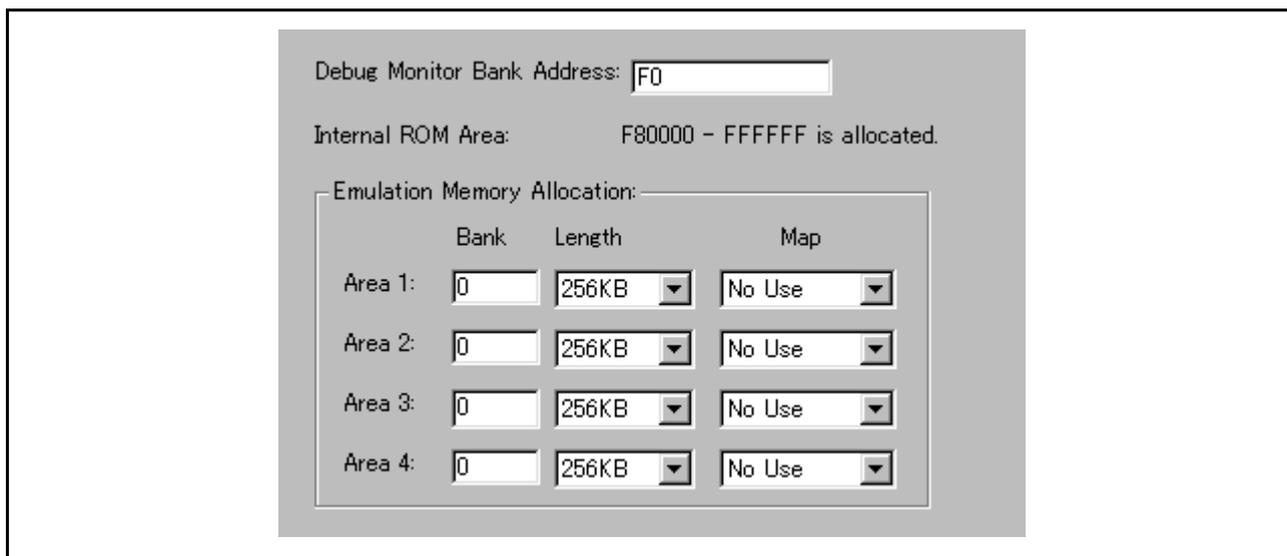
单击此选项卡可以显示各个 MCU 引脚的状态。它允许您检查 MCU 引脚状态是否与要设置的处理器模式匹配。



如果滑块处于中间位置，则表示值不确定。

5.3.1.2 Emulation Memory (仿真存储) 选项卡

指定的内容在下次启动时生效。



5.3.1.2.1 调试监视程序的存储体地址设置

此产品分配一个 64 KB 的连续地址区作为仿真器的工作区，以供调试监视程序使用。

指定目标系统不使用的任何存储体。调试监视程序使用从指定存储体起始地址开始的 64 KB 区域。

(示例：如果指定存储体为“F0”，则调试监视程序使用从地址 F000000h 开始的 64 KB 区域。)

Debug Monitor Bank Address: <input type="text" value="F0"/>

- 不能引用或设置这里指定的存储体的内容。

[Memory] (存储器) 窗口或 [Program/Source] (程序/源) 窗口的反汇编显示模式中所显示的此区域的内容可能不正确。

- 无法指定以下存储体地址：
 - MCU 内部资源 (例如, SFR 和 RAM 区)
 - DRAM 区和多路复用区
 - 中断向量区

5.3.1.2.2 内部 ROM 的自动仿真存储器分配

选择单片或存储器扩展模式时，仿真存储器会自动分配到内部 ROM 区。

自动分配的内部 ROM 地址范围显示在此字段中。

Internal ROM Area: F80000 - FFFFFFF is allocated.
--

5.3.1.2.3 扩展区的仿真存储器分配

选择存储器扩展或微处理器模式时，仿真存储器可以分配到要调试的扩展区 (最多四个区)。

在这里为调试目标区分配存储器并指定其映射信息。

Emulation Memory Allocation:			
	Bank	Length	Map
Area 1:	<input type="text" value="c0"/>	<input type="text" value="1MB"/>	<input type="text" value="INTERNAL"/>
Area 2:	<input type="text" value="c2"/>	<input type="text" value="256KB"/>	<input type="text" value="EXTERNAL"/>
Area 3:	<input type="text" value="0"/>	<input type="text" value="256KB"/>	<input type="text" value="No Use"/>
Area 4:	<input type="text" value="0"/>	<input type="text" value="256KB"/>	<input type="text" value="No Use"/>

按照下面说明的过程操作。

Bank (存储体) (设置存储体地址)	使用十六进制数指定要分配的调试目标区的存储体地址。 如果指定为 C0, 则 C0000h 为调试目标区的起始地址。
Length (长度) (指定区域大小)	指定调试目标区的大小 (256 字节或 1 MB)。 如果 [Length] (长度) 指定为 “256 字节”, 则存储体 00、04、08 到 FC (每 4 个存储体) 将指定用于 [Bank] (存储体); 如果 [Length] (长度) 指定为 “1 MB”, 则存储体 00、10、20, 直到 F0 (每 16 个存储体) 将指定用于 [Bank] (存储体)。
Map (映射) (指定区域映射)	为指定的区域指定映射信息 ([Internal] (内部) 或 [External] (外部))。 如果没有指定区域, 则选择 “No Use” (未使用)。 Internal (内部): 指定为 “Internal” (内部) 的区域会映射到内部区域 (仿真存储器)。 External (外部): 指定为 “External” (外部) 的区域会映射到外部区域 (目标系统中的外部资源)。

- 为 [Map] (映射) 选择 “No Use” (未使用) 的区域和这里未指定的区域会映射到外部区域。
与明确指定区域为 “External” (外部) 的情况相比, 唯一的区别是下载速度。(下载到这些区域的速度低于下载到指定为 “External” (外部) 的区域。)
 - 内部 ROM 区会自动映射到仿真存储器。因此无需在这里进行设置。
 - 请注意调试区不会重叠。
 - 请确保指定调试目标区总大小不超过所用仿真探针的仿真存储器大小。
可以分配的仿真存储器大小会根据各个仿真探针的不同而有所不同。(请参阅仿真探针的用户手册。)
- 仿真存储区的设置会根据指定处理器模式的不同而有所不同。

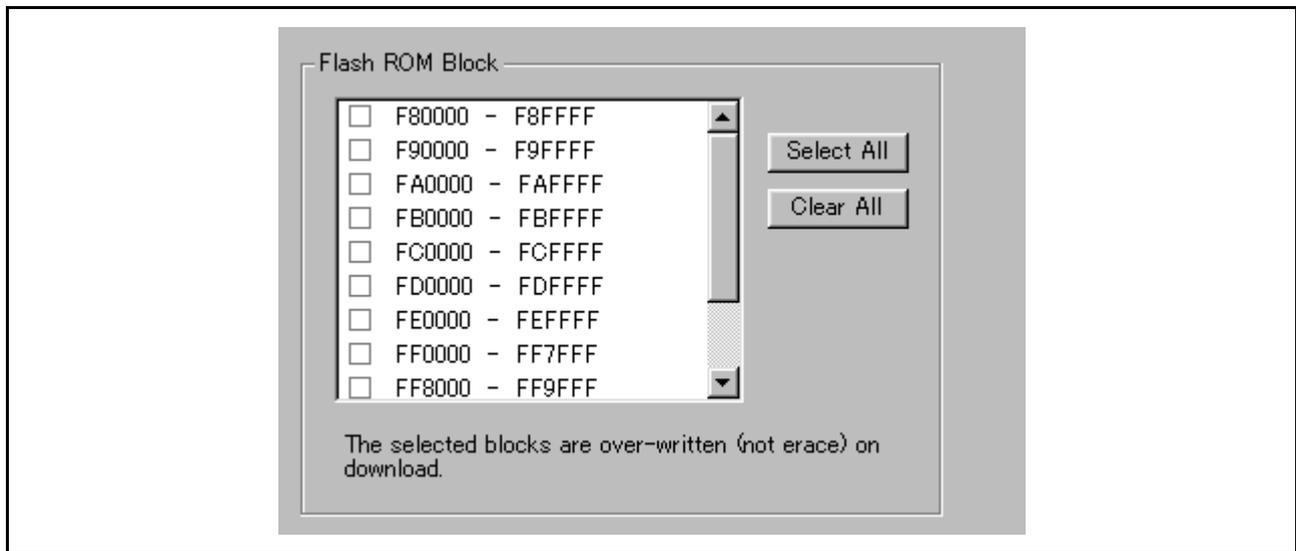
- “Single-chip Mode” (单片模式)
无需指定要分配作为仿真存储器的区域。
内部 ROM 区会自动映射到仿真存储器。自动映射区的地址范围显示在 [Internal ROM Area:] (内部 ROM 区:) 字段中。
- “Memory Expansion Mode” (存储器扩展模式) (8 位和 16 位)
如果除了内部 ROM 区之外还有一个要分配作为仿真存储器的区域, 请单独指定它。
内部 ROM 区会自动映射到仿真存储器。自动映射区的地址范围显示在 [Internal ROM Area:] (内部 ROM 区:) 字段中。
- “Microprocessor Mode” (微处理器模式) (8 位和 16 位)
指定要单独分配的区域。(没有自动分配的区域。)

注意事项

- 使用 Map 命令指定的映射设置数据不会在 [Emem] 对话框中反映。
- 按照使用优先级的顺序设置仿真存储区。
要通过 Map 命令设置的仿真存储区带有编号 (未使用的区域不算)。
因此, 在 [Emem] 对话框中设置的仿真存储区和 Map 命令设置的仿真存储区编号不匹配。

5.3.1.3 Flash Clear (闪存清除) 选项卡

指定的内容在下次启动时生效。



5.3.1.3.1 清除 MCU 内部闪存 ROM 的设置

指定当下载目标程序或数据时是否要清除 MCU 内部闪存 ROM 的内容。

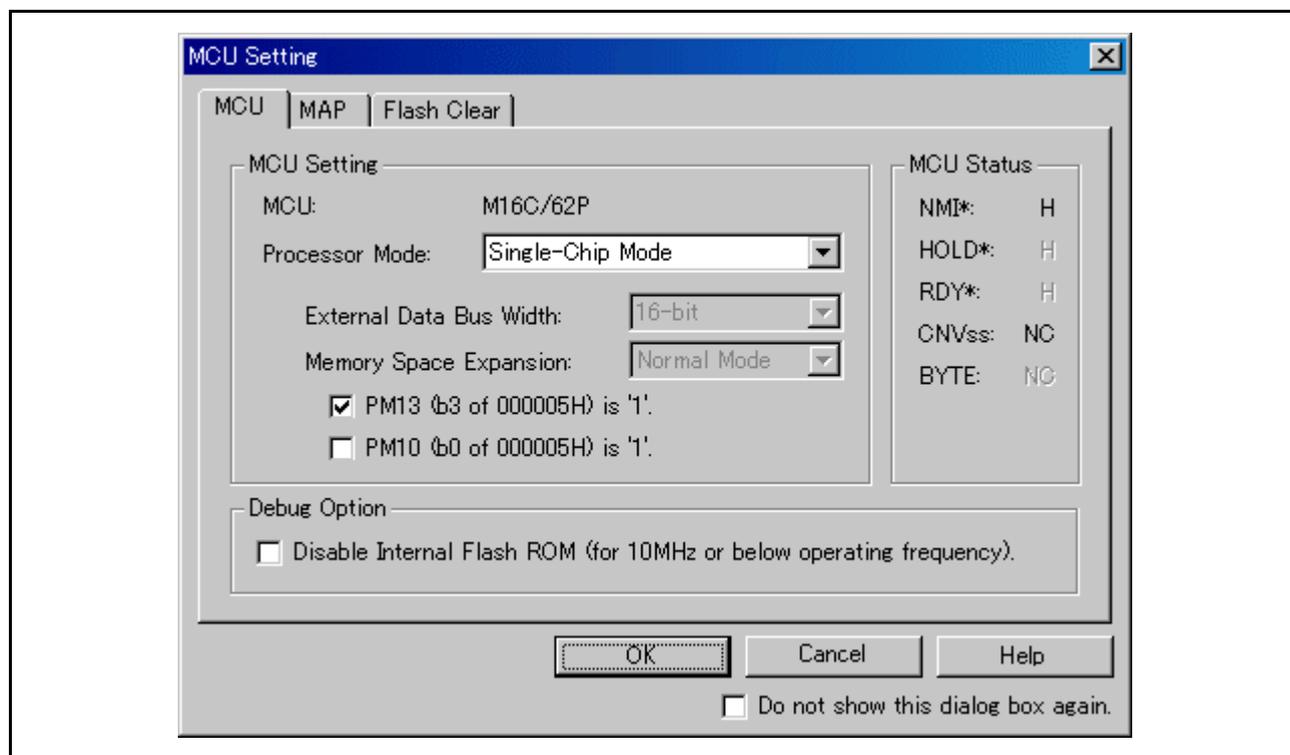
MCU 内部闪存 ROM 逐块显示在列表视图中。

- 打开选定标记的块不会在下载时清除闪存内容。处于不会被下载盖写的位置的存储内容将保持原样。
- 关闭选定标记的块会在下载时清除闪存内容。
- 按 [Select All] (全选) 按钮可以防止所有的块在下载时被清除。
- 按 [Clear All] (全部清除) 按钮可以在下载时清除所有的块。

5.4 设置 M16C/R8C 调试器

5.4.1 MCU Setting (MCU 设置) 对话框

在 [MCU Setting] (MCU 设置) 对话框中，设置有关用户目标的信息。关闭 [Init] (初始化) 对话框以后会打开 [MCU Setting] (MCU 设置) 对话框。



此对话框上可用的选项卡因所用的产品而异。有关详细信息，请在下表中单击所需的选项卡名称。

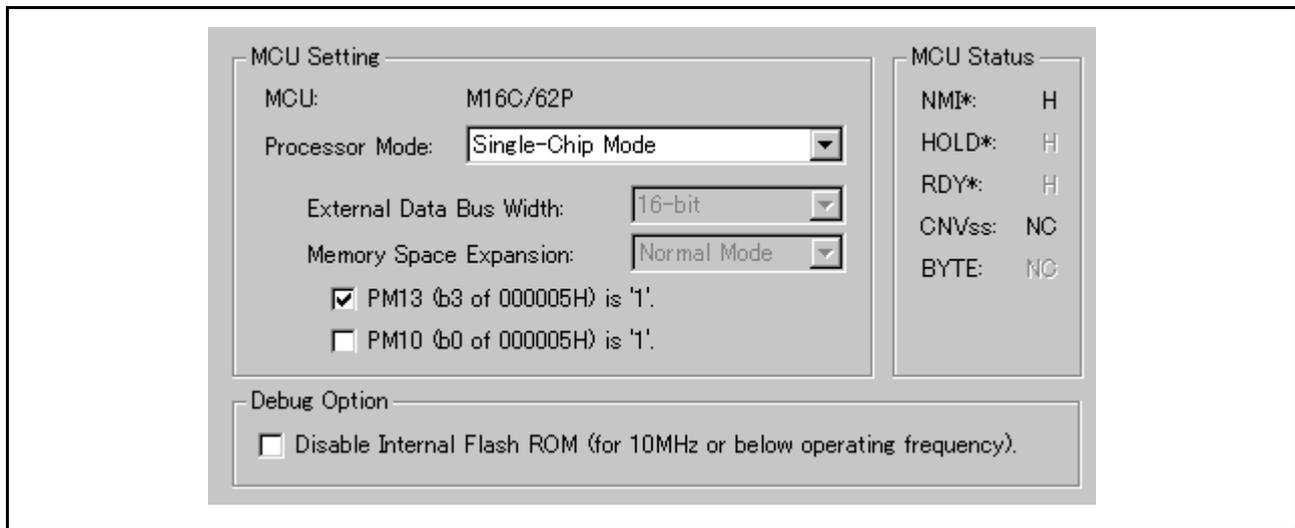
选项卡名称	内容
MCU	指定 MCU 的处理器模式、调试选项等等。
MAP (映射)	设置仿真存储器映射到的存储区。
Flash Clear (闪存清除)	指定是否要清除 MCU 内部闪存 ROM 的内容。

若要在下次启动调试器时让 [MCU Setting] (MCU 设置) 对话框保持关闭状态，请在 [MCU Setting] (MCU 设置) 对话框底部选择 “Next Hide” (下次隐藏)。可以使用下列方法中的任意一种来打开 [MCU Setting] (MCU 设置) 对话框：

调试器启动以后，选择菜单 - [Setup] (设置) -> [Emulator] (仿真器) -> [Target...] (目标...)

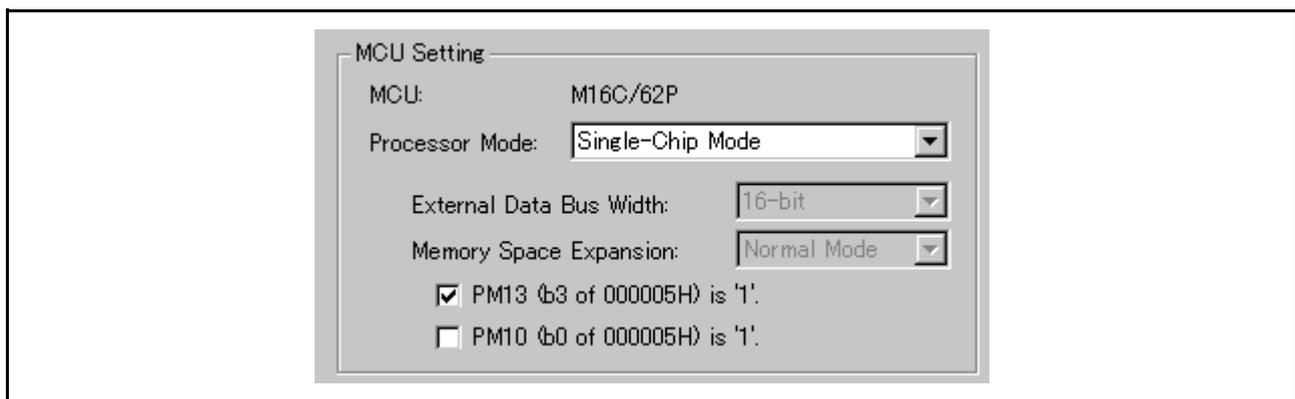
5.4.1.1 MCU 选项卡

指定的内容在下次启动时生效。



选择处理器模式

指定目标系统的处理器模式。



可以指定以下任意一项。

- Single-chip Mode
单片模式
- Memory Expansion Mode
存储器扩展模式
- Microprocessor Mode
微处理器模式

同时还需要根据选择的处理器模式指定以下信息。

- External Data Bus Width （外部数据总线宽度）

如果选择了存储器扩展或微处理器模式，请为外部总线宽度指定“16-bit”（16位）或“8-bit”（8位）。请确保指定的外部总线宽度与 BYTE 引脚的设置匹配。

- Memory Space Expansion （存储空间扩展）

如果选择了存储器扩展或微处理器模式，请指定是否要使用存储空间扩展工具。如果要使用存储空间扩展工具，请选择“4MB Mode”（4MB 模式）；如果不使用，请选择“Normal Mode”（正常模式）。

- PM13(b3 of 000005H)

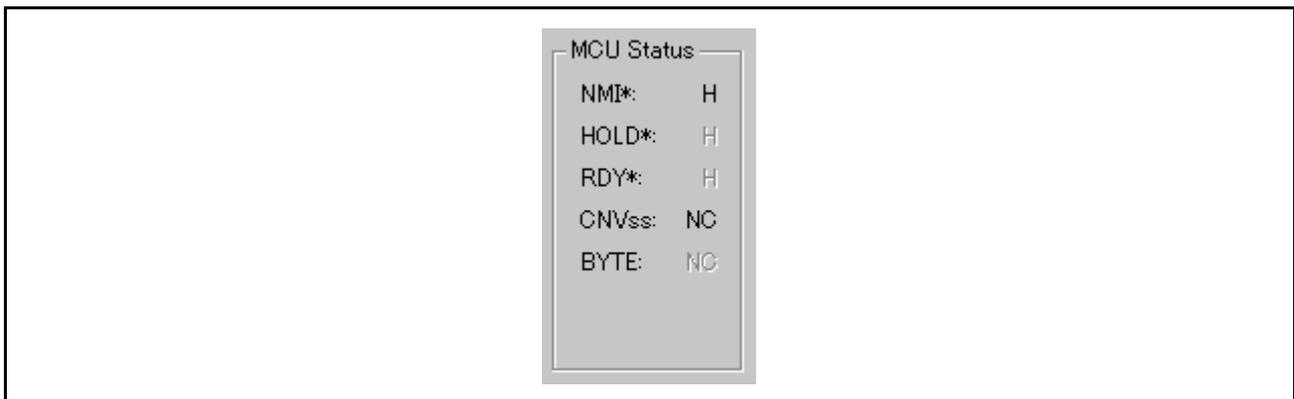
指定是否设置 PM13（000005H 的 b3）位。如果您在 PM13 为 1 的设置情况下使用目标系统，请选择此选项。

- PM10(b0 of 000005H)

指定是否设置 PM10（000005H 的 b0）位。如果您在 PM10 为 1 的设置情况下使用目标系统，请选择此选项。

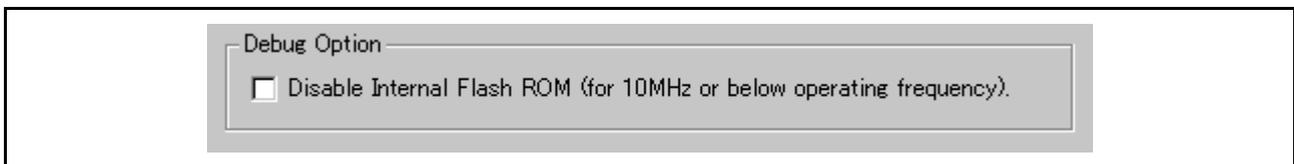
5.4.1.1.1 检查 MCU 状态

单击此选项卡可以显示各个 MCU 引脚的状态。它允许您检查 MCU 引脚状态是否与要设置的处理器模式匹配。



“NC”表示值不确定。

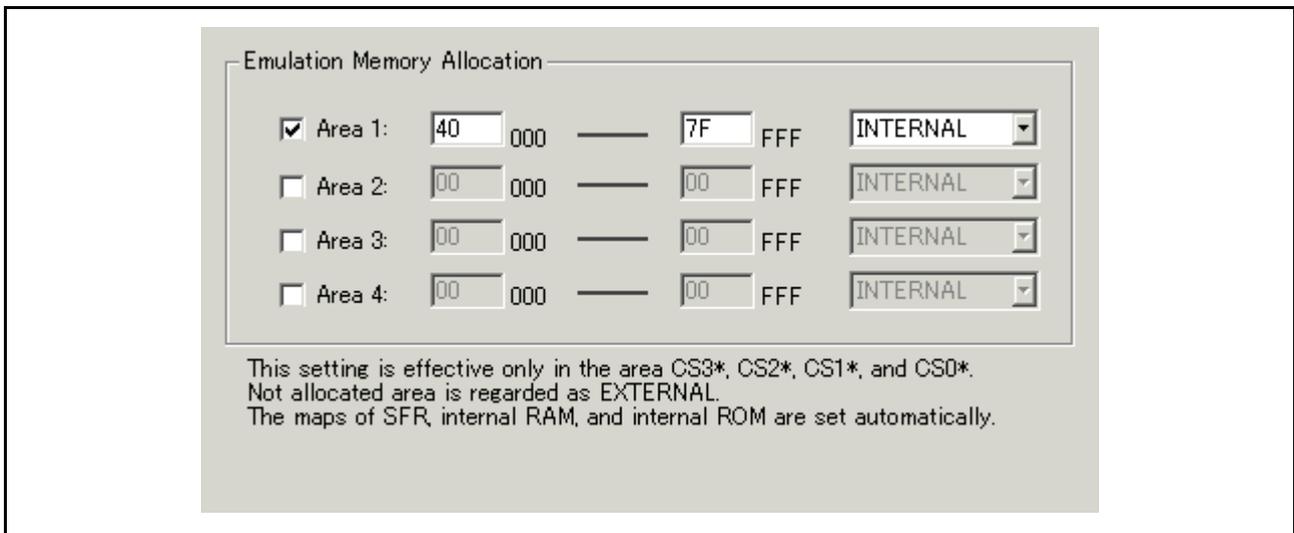
5.4.1.1.2 设置调试选项



如果您不想让程序下载到 MCU 的内部闪存 ROM，请单击此复选框。当无法改写内部闪存 ROM 时（例如在低电压情况下使用 MCU 时），可能需要选中此选项。最初，此选项处于取消选中状态。当您的系统使用 MCU 的工作频率超过 10MHz 时，如果选中该选项，则您的系统将不会正常工作。在这种情况下，请不要选中此选项。

5.4.1.2 [MAP] 选项卡

指定的内容在下次启动时生效。



5.4.1.2.1 仿真存储器分配

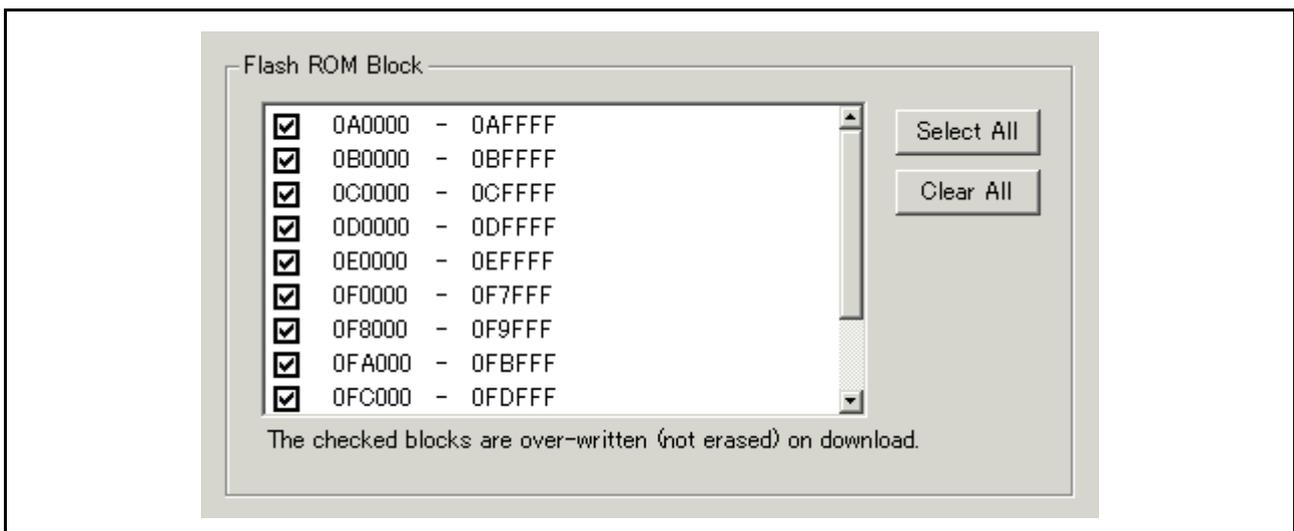
以 4 KB 为单位设置仿真存储器要映射到的存储区。可以设置四个这样的存储区。

仿真存储器映射到标志为“Internal”（内部）的区域。没有选择的区域和没有指定任何内容的区域会分配到外部区域。

请注意，[MAP]（映射）设置仅对 CS3*、CS2*、CS1* 和 CS0* 区域有效。SFR、内部 ROM 和内部 RAM 区会自动映射。

5.4.1.3 Flash Clear（闪存清除）选项卡

指定的内容在下次启动时生效。



5.4.1.3.1 清除 MCU 内部闪存 ROM 的设置

指定当下载目标程序或数据时是否要清除 MCU 内部闪存 ROM 的内容。MCU 内部闪存 ROM 逐块显示在列表视图中。

- 打开选定标记的块不会在下载时清除闪存内容。处于不会被下载盖写的位置的存储内容将保持原样。
- 关闭选定标记的块会在下载时清除闪存内容。
- 按 [Select All]（全选）按钮可以防止所有的块在下载时被清除。
- 按 [Clear All]（全部清除）按钮可以在下载时清除所有的块。

教程

6 教程

6.1 简介

本节通过一个教程程序对此调试器的主要功能进行介绍。该教程程序的安装目录为 \WorkSpace\Tutorial，位于已安装 HEW 的驱动器之上。每个目标和 MCU 均有各自的工作区。请针对您的系统选择相应的工作区，然后从 [Open Workspace...]（打开工作区...）菜单打开工作区文件 (*.hws)。

该教程程序基于对十个随机数据项进行升序或降序排序的 C 程序。

该教程程序执行以下操作：

- tutorial 函数生成要排序的随机数据。
- sort 函数按升序对生成的随机数据进行排序。
- 然后 change 函数按降序对这些数据进行排序。

注意

重新编译后的地址可能与本章所述的地址不同。

6.2 使用

请遵照以下说明进行操作：

6.2.1 步骤 1：启动调试器

6.2.1.1 使用前的准备工作

要运行 HEW 并连接到仿真器，请参阅“4 使用前的准备工作”。

6.2.1.2 设置调试器

如果调试器与仿真器相连，则会显示设置调试器的对话框。请在此对话框中设置调试器。

要在此对话框中设置调试器，请参阅“5 设置调试器”。

调试器设置完成后，即可实现调试器的所有功能。

6.2.2 步骤 2：检查 RAM 的运作情况

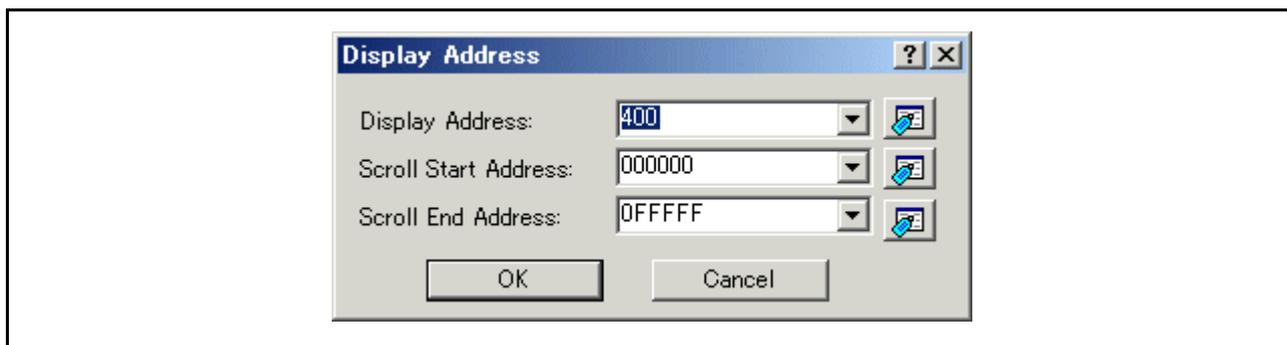
检查 RAM 是否正常运作。在 [Memory]（存储器）窗口中显示和编辑存储器所存储的内容，从而检查存储器是否正常运作。

注意

在某些单片机上，存储器可为板载安装方式。在这种情况下，通过上述方式可能无法检查存储器的工作情况。建议创建一个检查存储器的程序。

6.2.2.1 检查 RAM 的运作情况

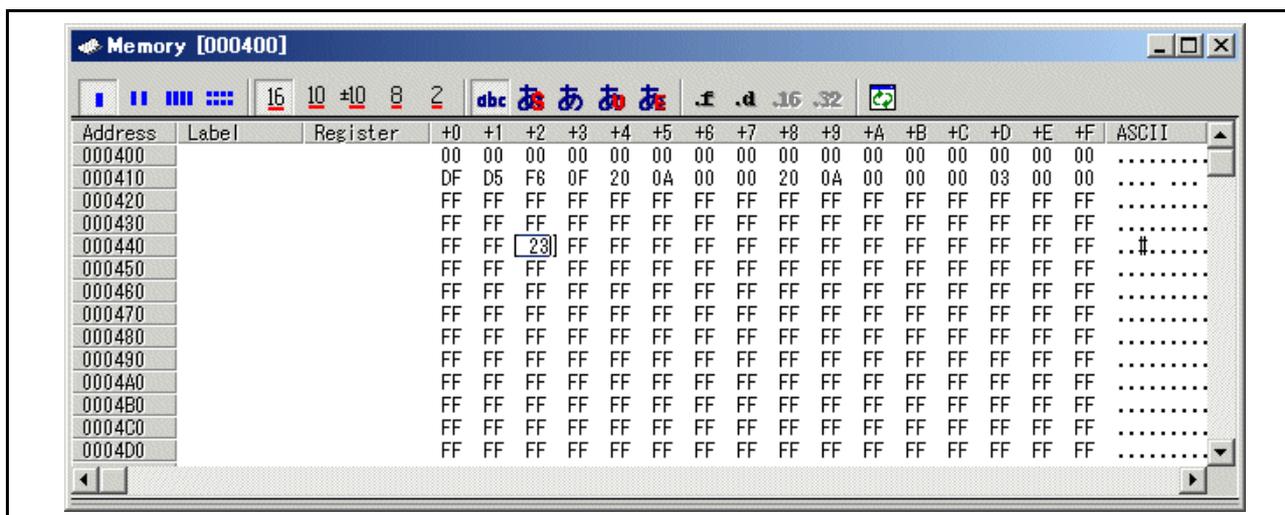
从 [View]（视图）菜单的 [CPU] 子菜单中选择 [Memory]（存储器），然后在 [Display Address]（显示地址）编辑框输入 RAM 地址（此处输入“400”）。在 [Scroll Start Address]（滚动起始地址）和 [Scroll End Address]（滚动结束地址）编辑框中保留默认设置。（默认情况下，滚动范围为 0h 至 MCU 的最大地址。）



注意

RAM 区域的设置因产品而异。有关详细信息，请参阅硬件手册。

单击 [OK]（确定）按钮。此时显示 [Memory]（存储器）窗口，窗口中显示指定的存储区。



将鼠标光标放在 [Memory]（存储器）窗口中数据显示区的某个点，然后双击即可更改该点处的值。

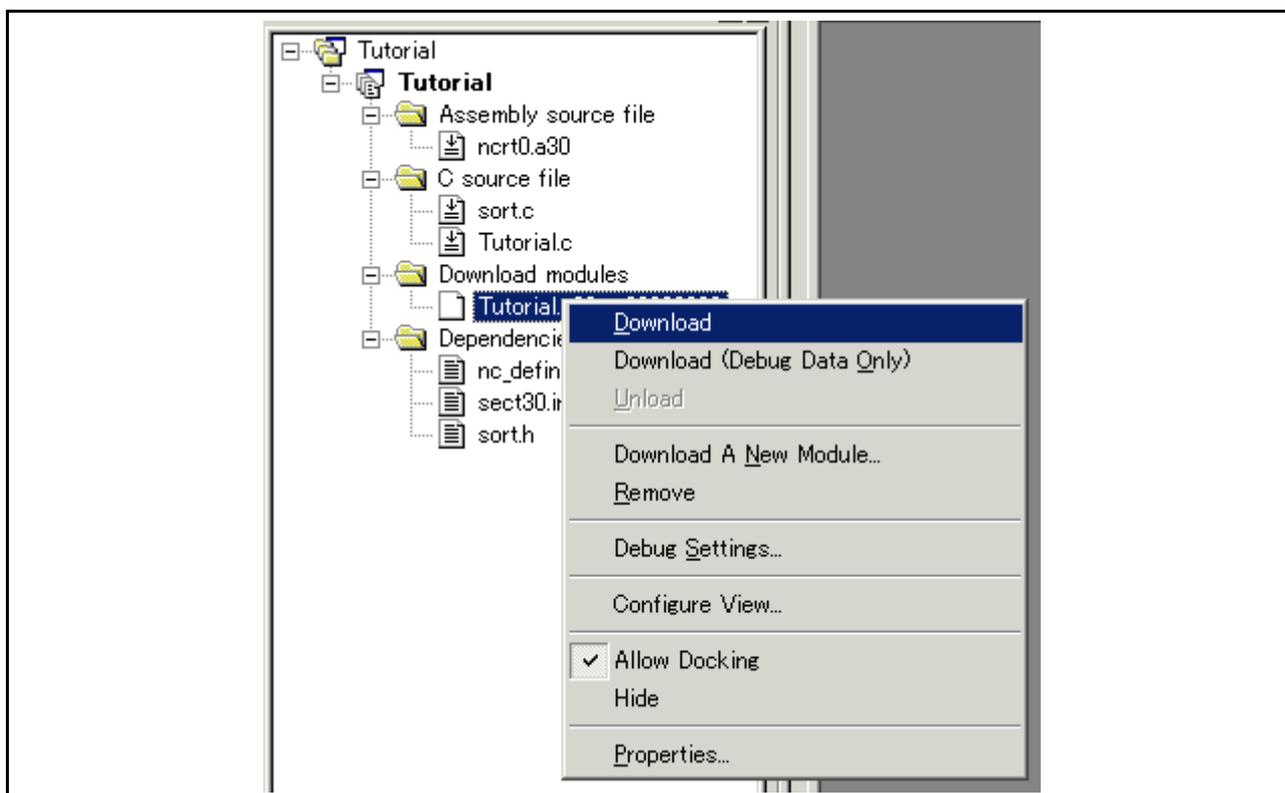
6.2.3 步骤 3: 下载教程程序

6.2.3.1 下载教程程序

下载要调试的目标程序。下载文件和下载地址取决于您所用的目标 MCU。请使用与您的目标 MCU 相应的图像和地址替换屏幕图像和地址。

- 用于 M16C/R8C 或 M32C 的调试器

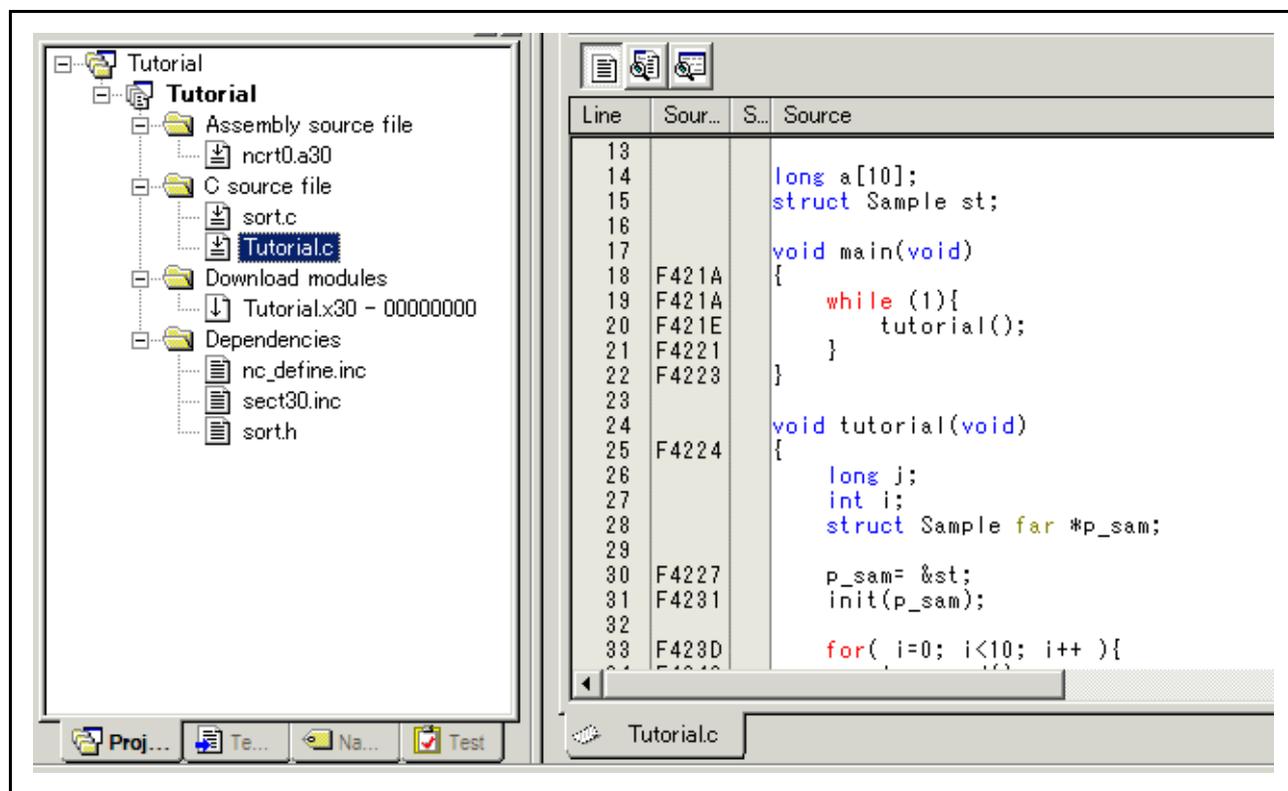
在 [Download modules]（下载模块）下，从 [Tutorial.x30] 选择 [Download module]（下载模块）。



6.2.3.2 显示源程序

使用此调试器，用户可以执行源码级用户程序调试。

双击 [C source file] (C 源文件) 下的 [tutorial.c]。此时会打开 [Editor(Source)] (编辑器 (源码)) 窗口，其中显示 “Tutorial.c” 文件的内容。



如果需要，从 [Setup] (设置) 菜单选择 [Format Views...] (格式视图...) 选项，设置字体和字号以改善显示效果。

[Editor(Source)] (编辑器 (源码)) 窗口初始显示用户程序的起始部分，但用户可以使用滚动条查看用户程序中的其他语句。

6.2.4 步骤 4: 设置断点

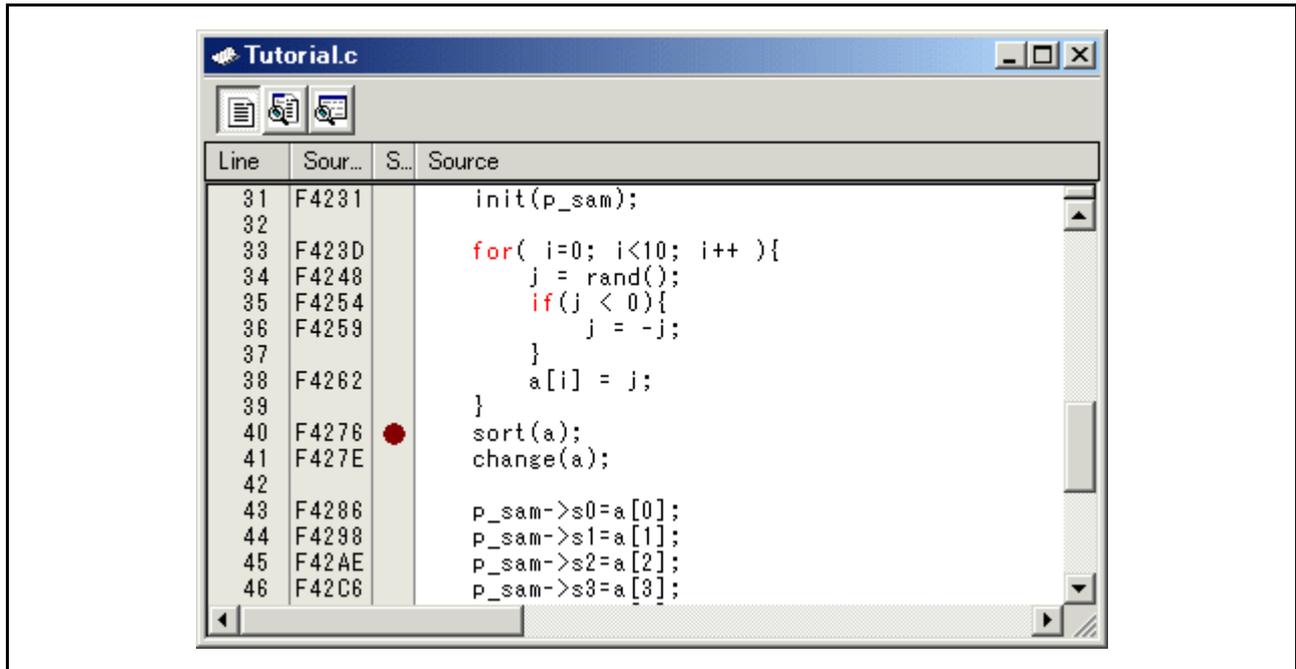
设置软件断点是一项基本调试功能。

通过 [Editor(Source)] (编辑器 (源码)) 窗口, 可在程序中的任意位置轻松设置软件断点。

6.2.4.1 设置软件断点

例如, 要在 sort 函数调用处设置一个软件断点, 请执行以下操作:

在调用 sort 函数的这一行, 双击 [S/W breakpoints] (软件断点) 列。



调用 sort 函数的行将显示红色符号。这表示已设置了一个软件断点。

6.2.5 步骤 5: 执行程序

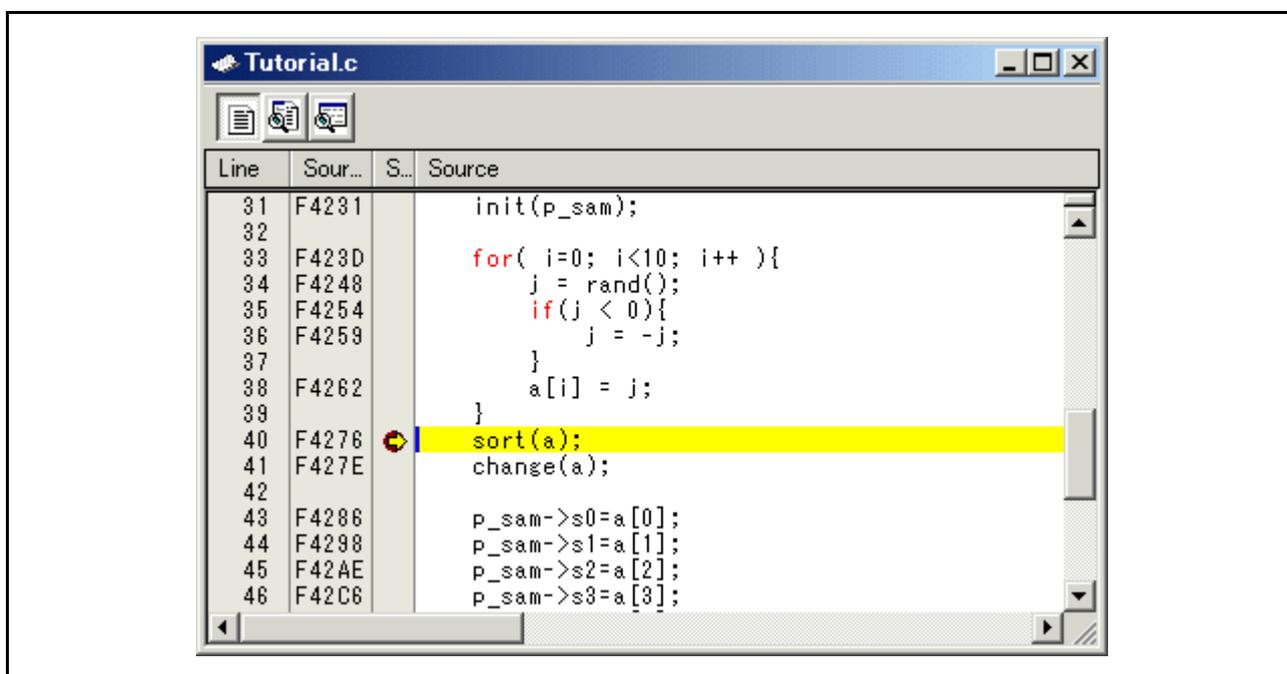
按以下过程执行程序:

6.2.5.1 复位 CPU

要将 CPU 复位, 请从 [Debug] (调试) 菜单选择 [Reset CPU] (复位 CPU), 或者单击工具栏上的 [Reset CPU] (复位 CPU) 按钮 。

6.2.5.2 执行程序

要执行程序, 请从 [Debug] (调试) 菜单选择 [Go] (开始), 或者单击工具栏上的 [Go] (开始) 按钮 。程序将执行至已设置的断点位置, [S/W Breakpoints] (软件断点) 列中显示一个箭头, 指示程序暂停的位置。

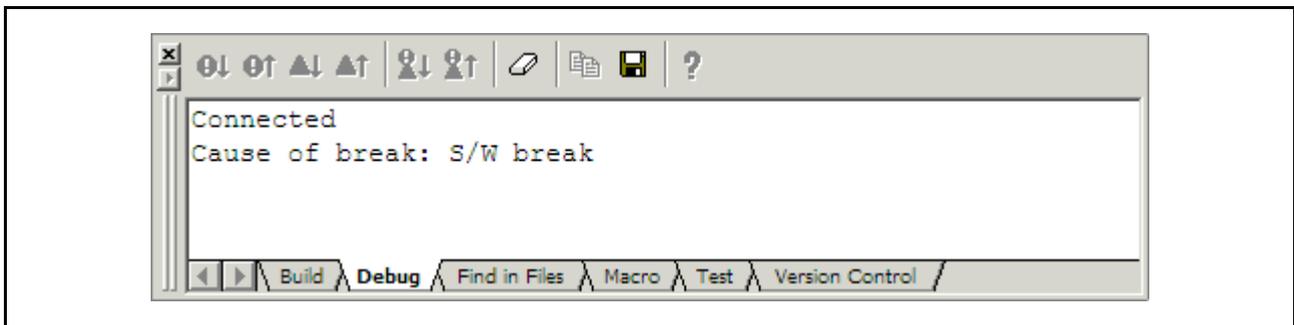


注意

暂停后要显示源文件时, 可能要求提供源文件的路径。如遇到这种情况, 请指定源文件的位置。

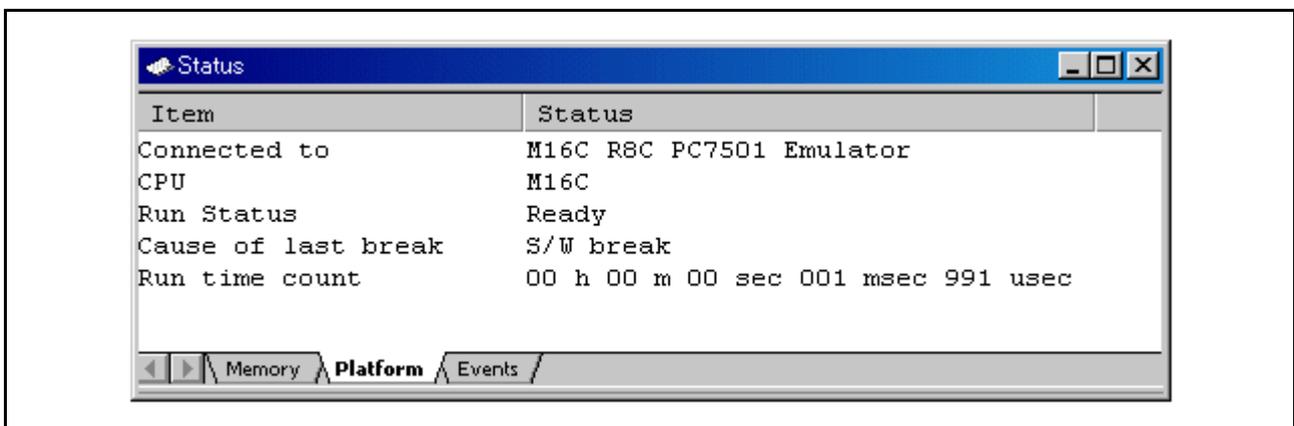
6.2.5.3 查看暂停原因

暂停原因显示于 [Output]（输出）窗口中。



用户也可以在 [Status]（状态）窗口中查看上一次出现暂停的原因。

从 [View]（视图）菜单的 [CPU] 子菜单中选择 [Status]（状态）。显示 [Status]（状态）窗口后，打开 [Platform]（平台）页，检查 Cause of last break（上次暂停的原因）的 Status（状态）。



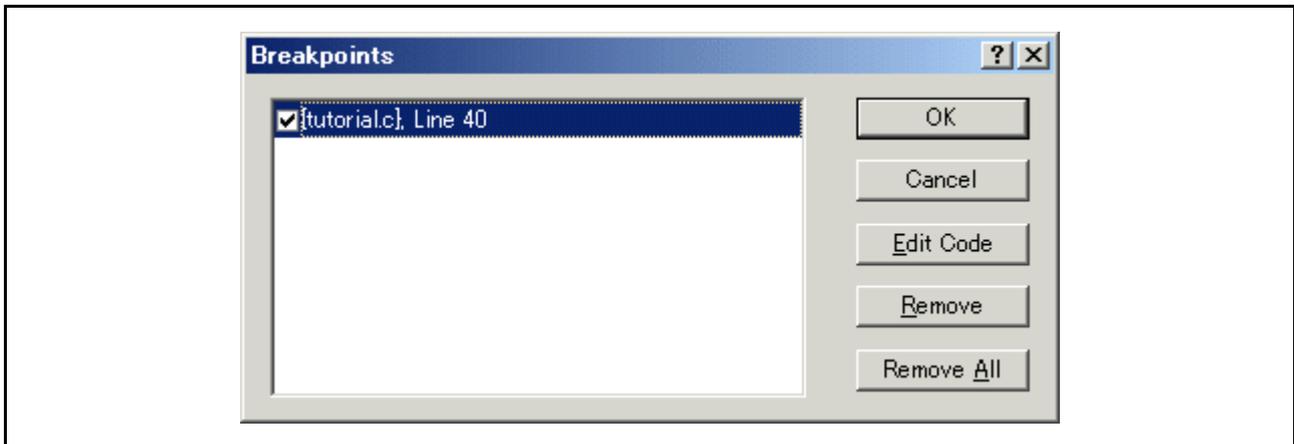
请参阅“11 显示程序停止的原因”了解暂停原因的表示法。

6.2.6 步骤 6: 查看断点

用户可以在 [Breakpoints] (断点) 对话框中看到程序中设置的所有断点。

6.2.6.1 查看断点

按 Ctrl+B 键, 此时显示 [Breakpoints] (断点) 对话框。



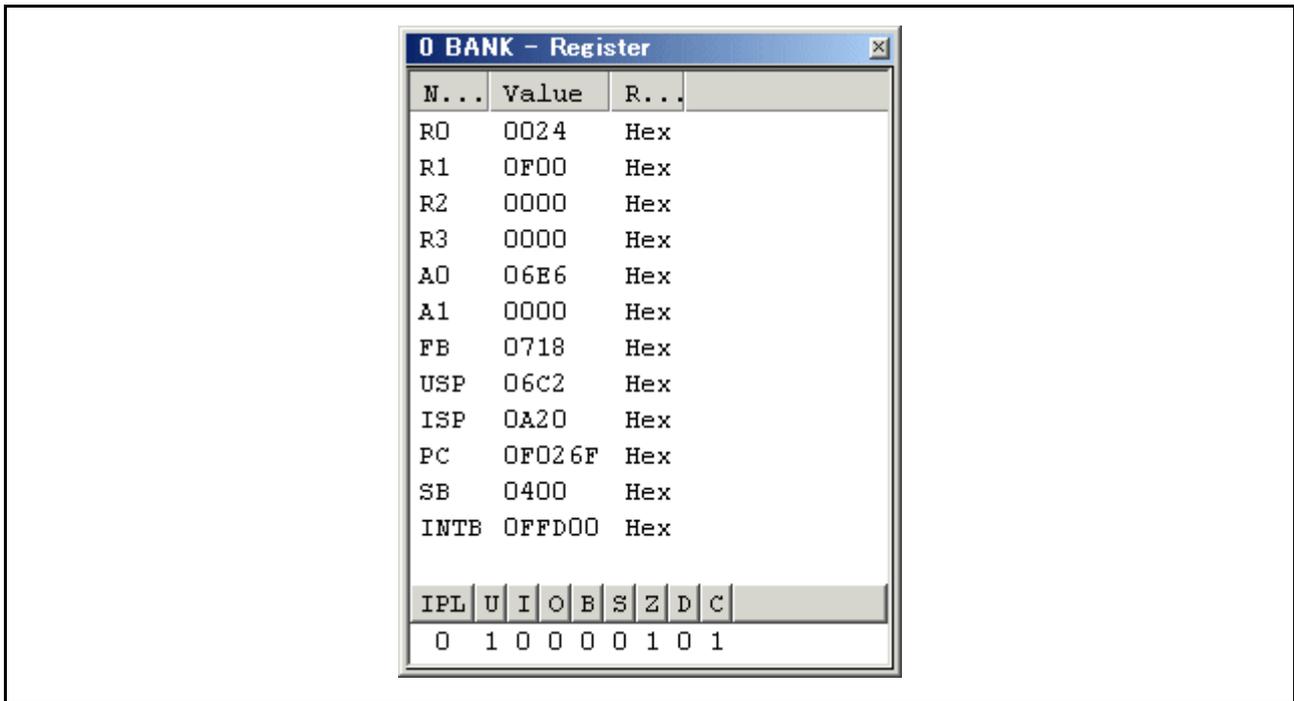
用户可通过此窗口删除、允许或禁止断点。

6.2.7 步骤 7: 查看寄存器

用户可以在 [Register]（寄存器）窗口中看到所有寄存器 / 标志值。

6.2.7.1 查看寄存器

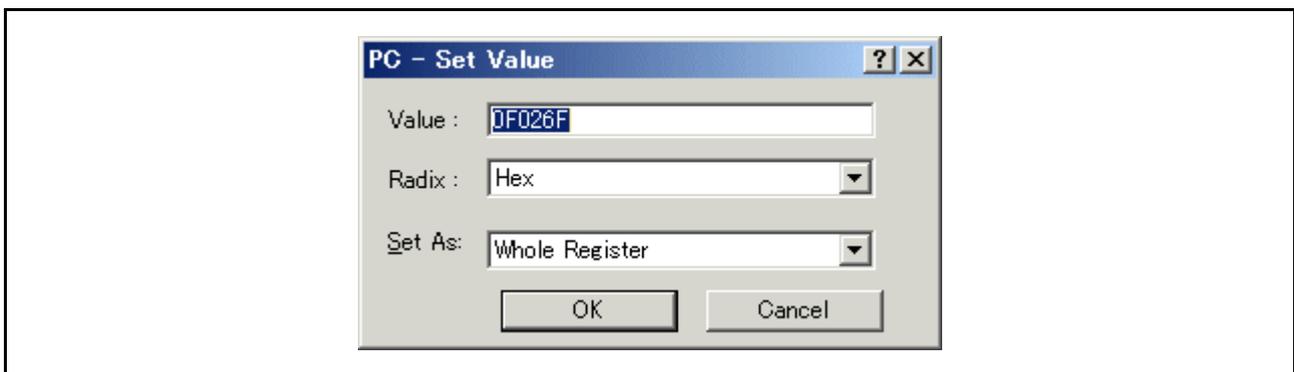
从 [View]（视图）菜单的 [CPU] 子菜单中选择 [Registers]（寄存器）。此时显示 [Register]（寄存器）窗口。下图所示为用于 M16C/R8C 的调试器的 [Register]（寄存器）窗口。



6.2.7.2 设置寄存器值

您可以从此窗口更改寄存器 / 标志值。

双击相关行以更改其寄存器。此时会打开对话框。输入要更改的值。



6.2.8 步骤 8: 查看存储器

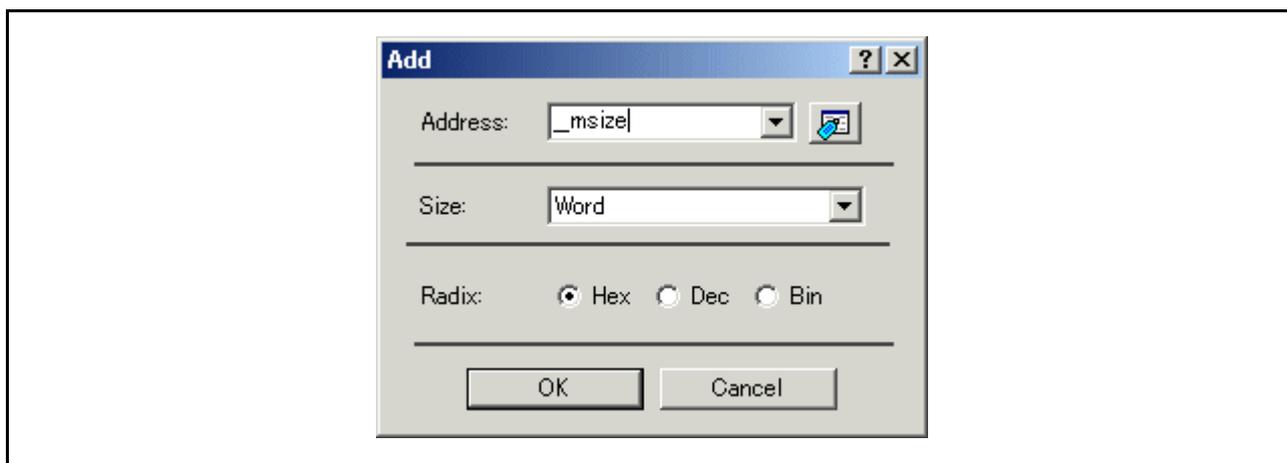
指定标签名称后，用户可以在 [ASM Watch]（ASM 监视）窗口中查看该标签已注册的存储内容。

6.2.8.1 查看存储器

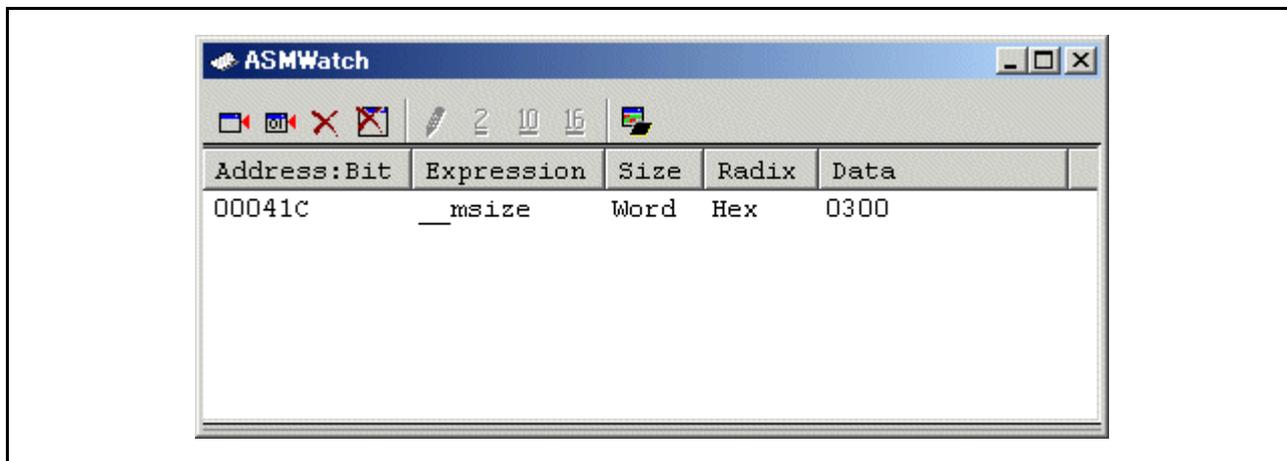
例如，要查看与 `__msize` 对应的、字大小的存储内容：

从 [View]（视图）菜单的 [Symbol]（符号）子菜单中选择 [ASM Watch]（ASM 监视），打开 [ASM Watch]（ASM 监视）窗口。

随后用鼠标右键单击 [ASM Watch]（ASM 监视）窗口，从弹出菜单中选择 [Add...]（添加...），在 [Address]（地址）编辑框中输入 `__msize`，并在 [Size]（大小）组合框中设置“Word”（字）。



单击 [OK]（确定）按钮。此时的 [ASM Watch]（ASM 监视）窗口会显示所指定的存储区域。



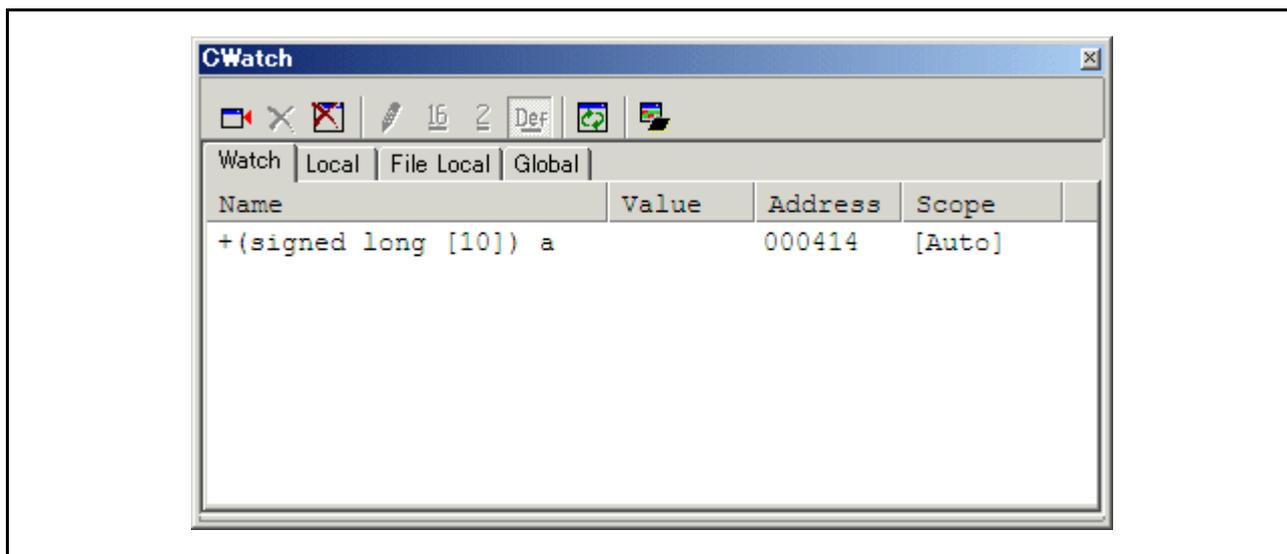
6.2.9 步骤 9: 监视变量

用户逐步执行程序时，可以监视到用户程序里所用变量中被更动的值。

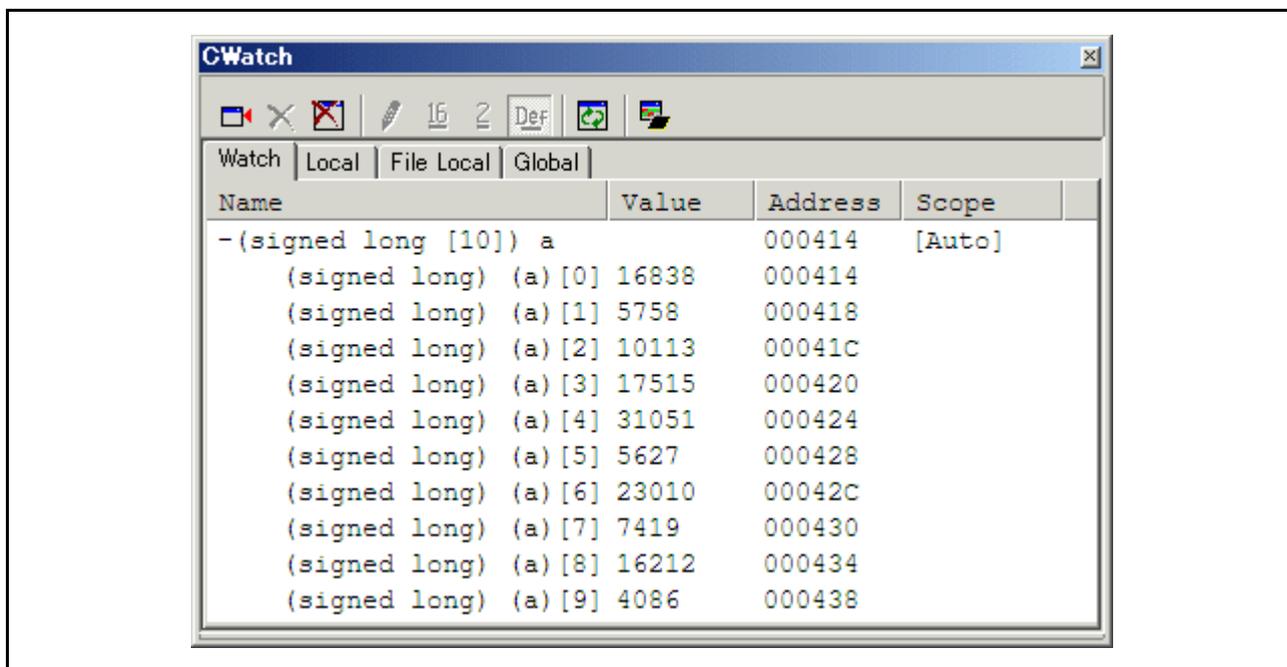
6.2.9.1 监视变量

例如，对于程序起始处声明的 long 类型的数组 a，可以通过以下步骤设置对其的监视：

在 [Editor(Source)]（编辑器（源码））窗口中，单击显示的数组 a 的左侧以定位光标，然后从鼠标右键菜单中选择 [Add C Watch...]（添加 C 监视...）。此时会打开 [C watch]（C 监视）窗口的 [Watch]（监视）选项卡，其中显示变量。



在 [C watch]（C 监视）窗口中，用户可以单击数组 a 左侧的“+”标记来监视所有元素。

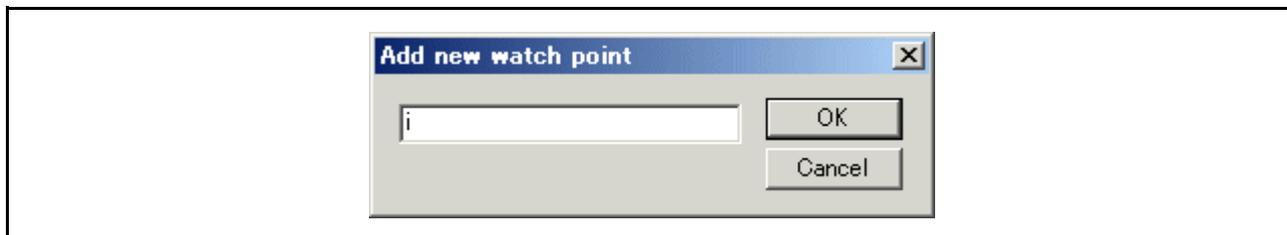


6.2.9.2 注册变量

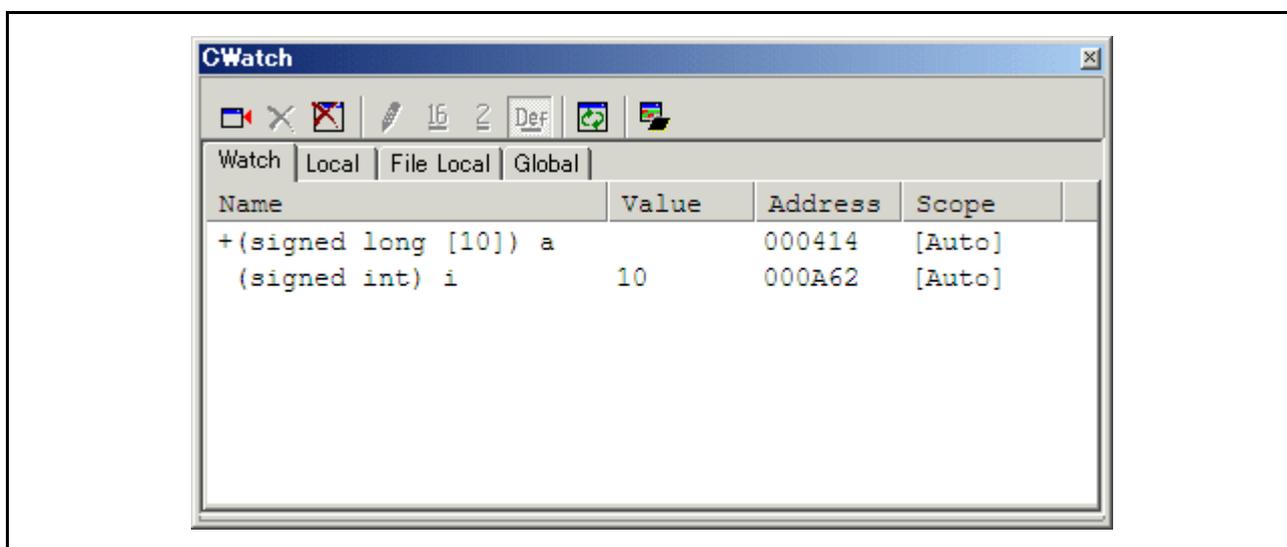
用户可以通过指定变量的名称将其添加到 [C watch] (C 监视) 窗口。

鼠标右键单击 [C Watch] (C 监视) 窗口，并从弹出菜单中选择 [Add...] (添加...)

此时显示以下对话框。输入变量 i。



单击 [OK] (确定) 按钮。现在 [C Watch] (C 监视) 窗口会显示 int 类型的变量 i。



6.2.10 步骤 10: 逐步执行程序

此调试器提供多个逐步执行菜单命令，从而实现高效率的程序调试。

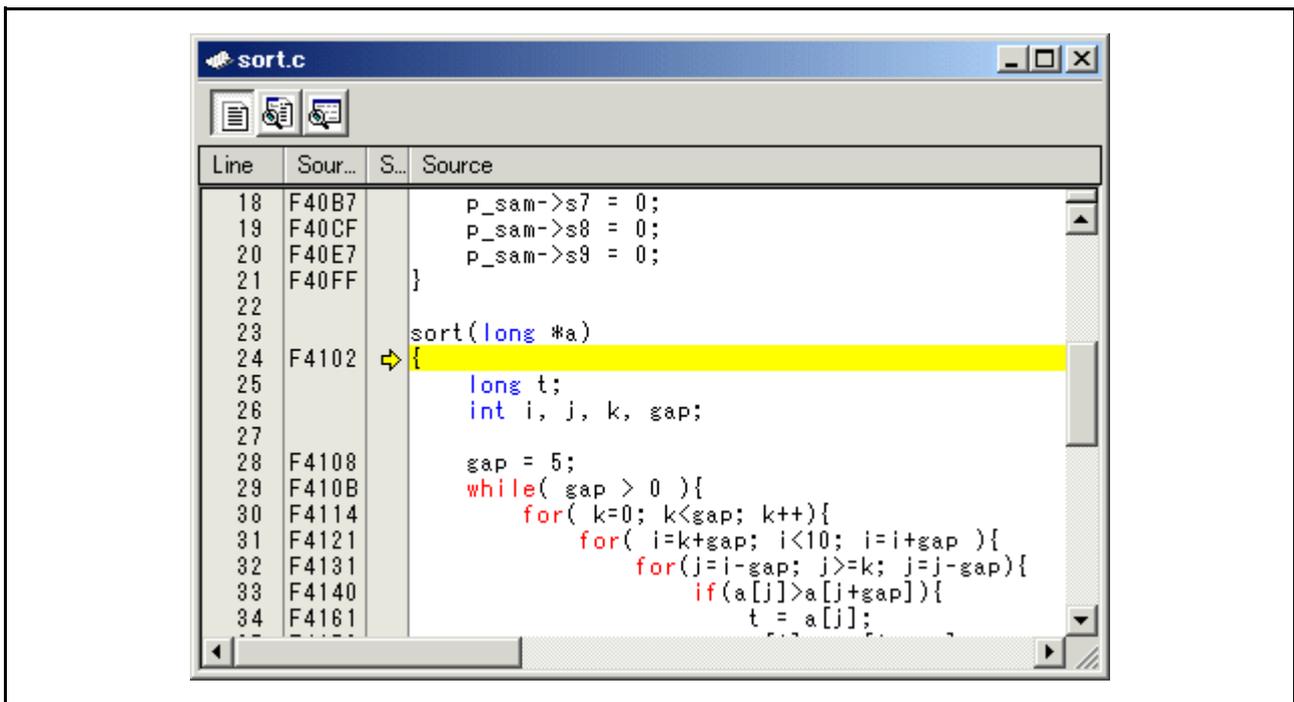
1. [Step In] (跳入)
执行每个语句，包括函数（子程序）内的语句。
2. [Step Out] (跳出)
跳出函数（子程序），并在程序中调用该函数（子程序）的语句的下一句处停止。
3. [Step Over] (跳过)
在单步中执行函数（子程序）调用。
4. [Step...] (单步...)
以指定的速率重复单步执行指定次数。

6.2.10.1 执行 [Step In] (跳入) 命令

[Step In] (跳入) 命令跳入被调用的函数（子程序），并在该函数（子程序）的第一个语句处停止。

要逐步执行 sort 函数，请从 [Debug] (调试) 菜单选择 [Step In] (跳入)，或者单击工具栏上的 [Step In] (跳入) 按钮 。

在 [Editor(Source)] (编辑器 (源码)) 窗口中，PC 光标将移动到 sort 函数的第一个语句。



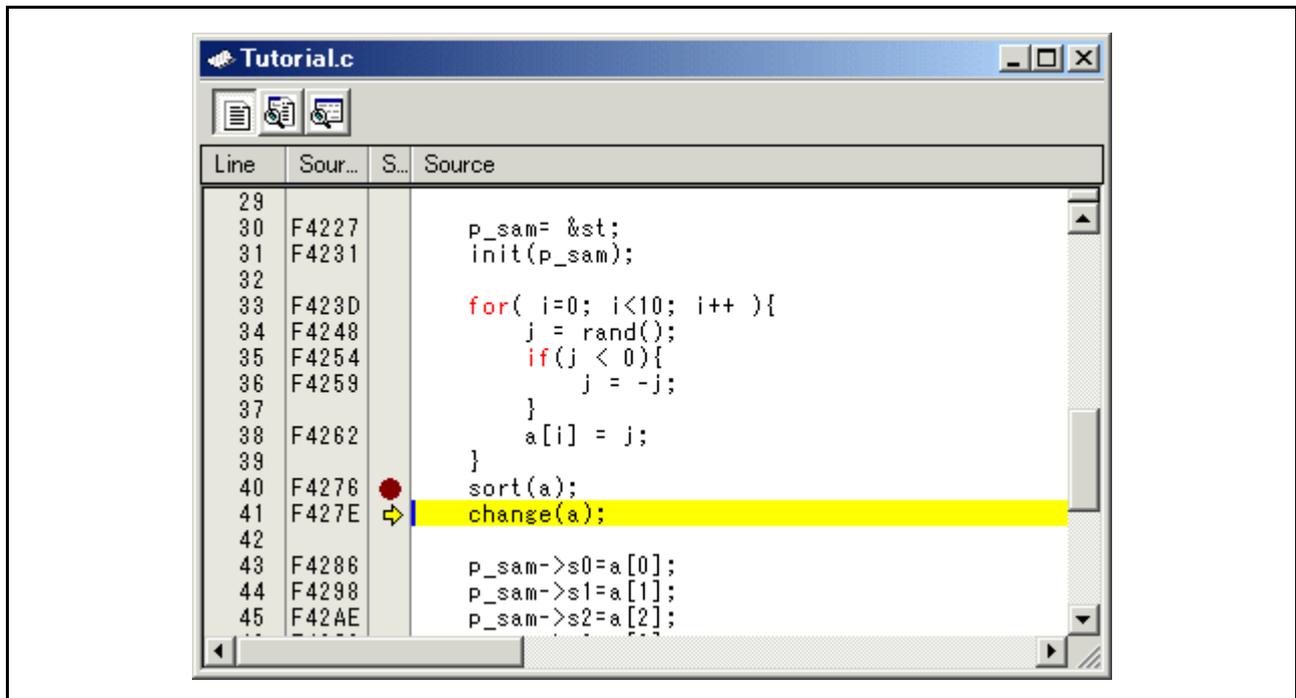
6.2.10.2 执行 [Step Out]（跳出）命令

[Step Out]（跳出）命令跳出所调用的函数（子程序），并在主程序中调用语句的下一语句处停止。

要跳出 sort 函数，请从 [Debug]（调试）菜单选择 [Step Out]（跳出），或者单击工具栏上的 [Step Out]

（跳出）按钮 。

PC 光标将移出 sort 函数，并移动到 change 函数前方的位置。



注意

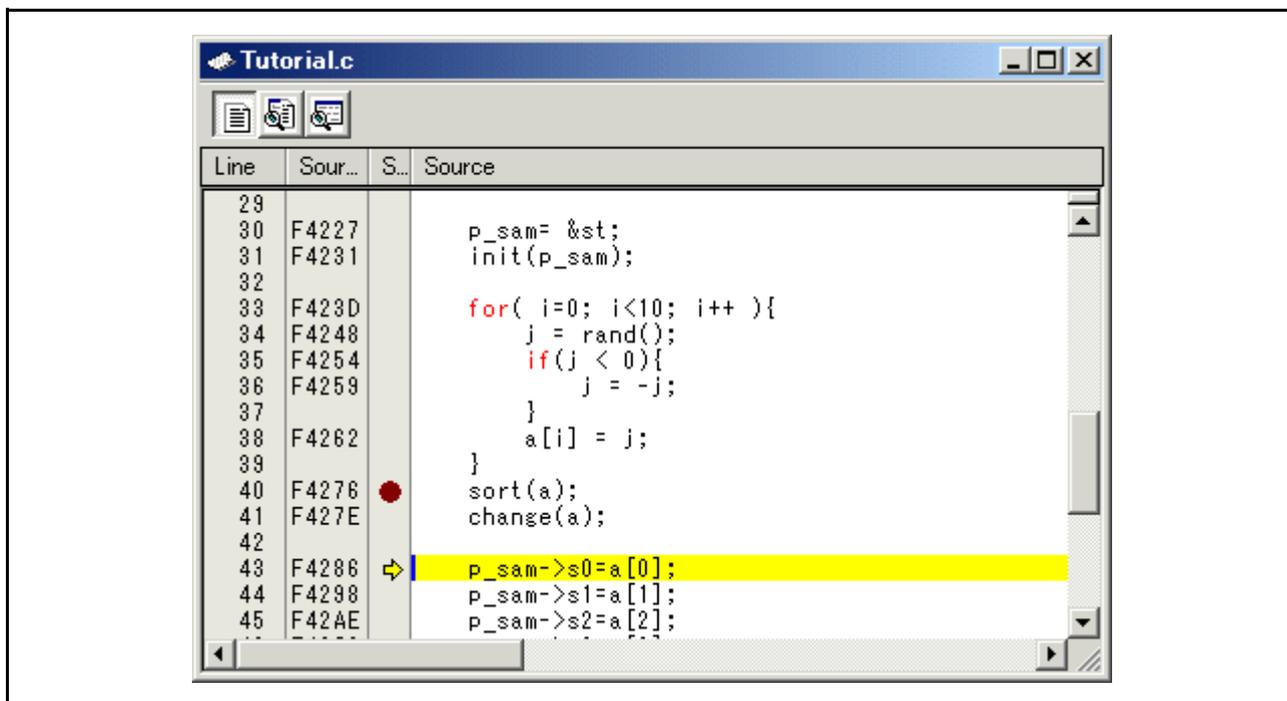
执行此函数需要一些时间。查明调用源后，使用 [Go To Cursor]（转至光标）。

6.2.10.3 执行 [Step Over]（跳过）命令

[Step Over]（跳过）命令将函数（子程序）调用作为单步执行，并在主程序的下一语句处停止。

要在单步中执行 change 函数中的所有语句，请从 [Debug]（调试）菜单选择 [Step Over]（跳过），或者单击工具栏上的 [Step Over]（跳过）按钮 。

PC 光标将移动到与 change 函数相邻的位置。



6.2.11 步骤 11：强制暂停程序执行

此调试器可以强制暂停程序执行。

6.2.11.1 强制暂停程序执行

取消所有断点。

要执行主函数的其他部分，请从 [Debug]（调试）菜单选择 [Go]（执行），或者单击工具栏上的 [Go]

（执行）按钮 。

程序进入死循环。要在执行时强制暂停，请从 [Debug]（调试）菜单选择 [Halt Program]（暂停程序），或者单击工具栏上的 [Halt]（暂停）按钮 。

6.2.12 步骤 12: 显示局部变量

用户可以通过 [C Watch] (C 监视) 窗口显示函数中的局部变量。

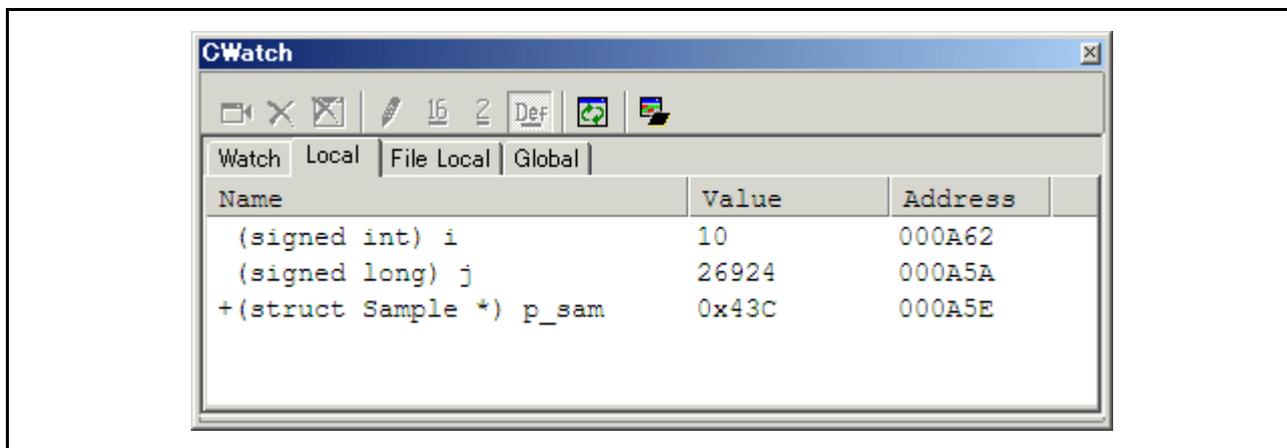
6.2.12.1 显示局部变量

例如, 我们将查看 tutorial 函数中的局部变量, 该函数声明了三个局部变量: i、j 和 p_sam。

从 [View] (视图) 菜单的 [Symbol] (符号) 子菜单中选择 [C Watch] (C 监视)。此时显示 [C Watch] (C 监视) 窗口。默认情况下, [C watch] (C 监视) 窗口有四个选项卡, 如下所示:

- [Watch] (监视) 选项卡:
仅显示用户已注册的变量。
- [Local] (局部) 选项卡
显示 PC 所在作用域可以引用的所有局部变量。如果作用域因程序的执行发生改变, 则 [Local] (局部) 选项卡的内容也会随之更改。
- [File Local] (文件局部变量) 选项卡
显示 PC 所在的文件作用域的所有文件局部变量。如果文件作用域因程序的执行发生改变, 则 [File Local] (文件局部变量) 选项卡的内容也会随之更改。
- [Global] (全局) 选项卡
显示已下载程序当前使用的所有全局变量。

显示局部变量时请选择 [Local] (局部) 选项卡。



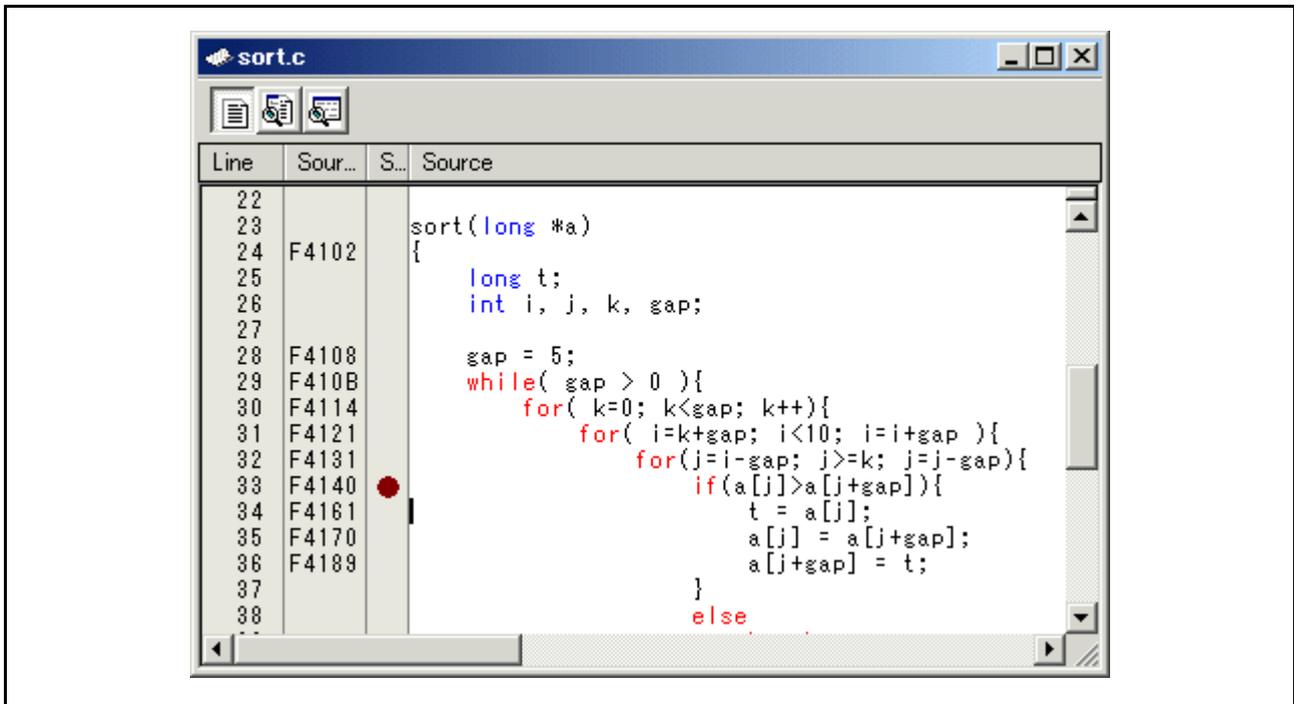
在 [Local] (局部) 窗口中, 双击指针 p_sam 左侧的 “+” 标记来显示 structure *(p_sam)。用户在 Tutorial 函数末尾引用这一结构的成员时, 随机数据将按降序排列。

6.2.13 步骤 13: 堆栈跟踪功能

调试器使用堆栈信息来显示调用序列中的函数名称，此调用序列最终到达程序计数器当前所指向的函数。

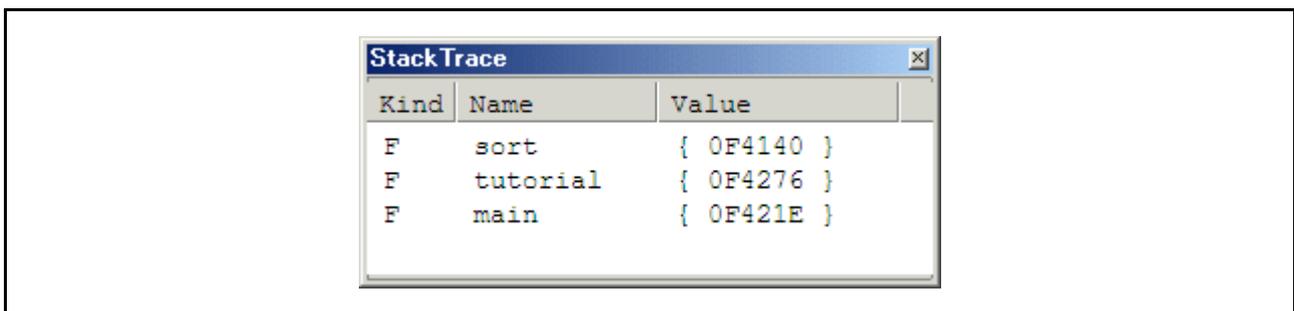
6.2.13.1 引用函数调用状态

在 sort 函数中，双击 [S/W Breakpoints]（软件断点）列设置一个软件断点。



要从复位向量地址执行用户程序，请从 [Debug]（调试）菜单选择 [Reset Go]（复位执行），或者单击工具栏上的 [Reset Go]（复位执行）按钮 。

程序执行暂停后，从 [View]（视图）菜单的 [Code]（代码）子菜单中选择 [Stack Trace]（堆栈跟踪）打开 [Stack Trace]（堆栈跟踪）窗口。



上图显示程序计数器当前位于 sort() 函数的选定行，而 sort() 函数从 tutorial() 函数中调用。

6.2.14 后续内容

此教程已对该调试器的使用进行了介绍。

使用仿真器提供的仿真功能可执行复杂调试。通过准确隔离和识别发生硬件和软件问题的条件，它可以有效调查此类问题。

7 窗口 / 对话框

此调试器的窗口如下所示。

单击窗口名称后会显示参考内容。

窗口名称	[View] (视图) 菜单
[RAM Monitor] (RAM 监控) 窗口	[View] (视图) ->[CPU]->[RamMonitor] (RAM 监控)
[ASM Watch] (ASM 监视) 窗口	[View] (视图) ->[Symbol] (符号) ->[ASMWatch] (ASM 监视)
[C Watch] (C 监视) 窗口	[View] (视图) ->[Symbol] (符号) ->[CWatch] (C 监视)
[Coverage] (覆盖率) 窗口	[View] (视图) ->[Code] (代码) ->[Coverage] (覆盖率)
[Script] (脚本) 窗口	[View] (视图) ->[Script] (脚本)
[S/W Break Point Setting] (软件断点设置) 窗口	[View] (视图) ->[Break] (断点) ->[S/W Break Points] (软件断点)
[H/W Break Point Setting] (硬件断点设置) 窗口	[View] (视图) ->[Break] (断点) ->[H/W Break Points] (硬件断点)
[Protect] (保护) 窗口	[View] (视图) ->[Break] (断点) ->[Protect] (保护)
[Address Interrupt Break Point Setting] (地址中断断点设置) 窗口	[View] (视图) ->[Break] (断点) ->[Address Interrupt Break Points] (地址中断断点)
[Trace Point Setting] (跟踪点设置) 窗口	[View] (视图) ->[Trace] (跟踪) ->[Trace Points] (跟踪点)
[Time Measurement] (时间计量) 窗口	[View] (视图) ->[Trace] (跟踪) ->[Time Measure] (时间计量)
[Trace] (跟踪) 窗口	[View] (视图) ->[Trace] (跟踪) ->[Trace] (跟踪)
[Data Trace] (数据跟踪) 窗口	[View] (视图) ->[Trace] (跟踪) ->[Data Trace] (数据跟踪)
[GUI I/O] 窗口	[View] (视图) ->[Graphic] (图形) ->[GUI I/O]
[MR] 窗口	[View] (视图) ->[RTOS]->[MR]
[MR Trace] (MR 跟踪) 窗口	[View] (视图) ->[RTOS]->[MR Trace] (MR 跟踪)
[MR Analyze] (MR 分析) 窗口	[View] (视图) ->[RTOS]->[MR Analyze] (MR 分析)
[MR Task Pause] (MR 任务暂停) 窗口	[View] (视图) ->[RTOS]->[MR Task Pause] (MR 任务暂停)
[Task Trace] (任务跟踪) 窗口	[View] (视图) ->[RTOS]->[Task Trace] (任务跟踪)
[Task Analyze] (任务分析) 窗口	[View] (视图) ->[RTOS]->[Task Analyze] (任务分析)

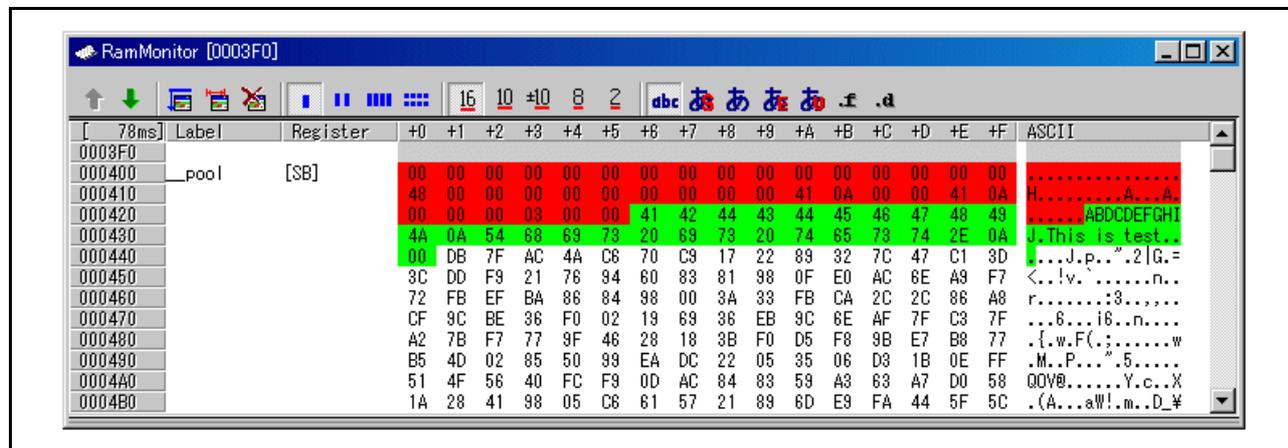
有关以下窗口的参考内容，请参阅 HEW 主部件的帮助文件。

- [Differences] (差异) 窗口
- [Map] (映像) 窗口
- [Command Line] (命令行) 窗口
- [Workspace] (工作区) 窗口
- [Output] (输出) 窗口
- [Disassembly] (反汇编) 窗口
- [Memory] (存储器) 窗口
- [IO] 窗口
- [Status] (状态) 窗口
- [Register] (寄存器) 窗口
- [Image] (图像) 窗口
- [Waveform] (波形) 窗口
- [Stack Trace] (堆栈跟踪) 窗口

7.1 [RAM Monitor] (RAM 监控) 窗口

目标程序运行时，[RAM monitor] (RAM 监控) 窗口会显示存储器内容的变化。

使用实时 RAM 监控功能，相关存储器内容将通过转储形式显示在 RAM 监控区域。目标程序运行时，将按指定间隔（默认情况下为 100 ms）对所显示的内容进行更新。



- 此系统具有 4 KB 大小的 RAM 监控区域，该区域可位于任何连续的地址区域，也可以由 16 个大小为 256 字节的单独块组成。

- RAM 监控区域可变换到任何需要的地址范围。

有关如何更改 RAM 监控区域的详细信息，请参阅 97 “7.1.2 设置 RAM 监控区域”。

默认 RAM 监控区域映射到一个 1 KB 大小的区域，该区域的起始地址为内部 RAM 的起始地址。

- 各个窗口可分别设置显示内容更新间隔。

目标程序运行时显示内容发生实际更新的实际更新间隔显示在地址显示区域的标题字段中。

- 数据显示区域和代码显示区域的背景颜色由“access”（存取）属性预先确定，如下所示。

“Access”（存取）属性	背景颜色
读地址	绿
写地址	红
非存取地址	白

背景颜色可以更改。

注意事项

- [RAM monitor] (RAM 监控) 窗口显示通过总线存取的数据。因此，显示的数据不反映所做的更改，除非目标程序对这些数据进行了存取，例如，从外部 I/O 直接盖写存储器时。
- 如果 RAM 监控区域中的数据不以字节长度显示，则该数据中各字节可能具有不同的存储器 “access” (存取) 属性。如果数据中的字节具有不同的 “access” (存取) 属性 (如在上述情况下)，则这类数据在窗口中显示时将置于括号中。在这种情况下，从背景颜色可以看出数据中第一个字节的 “access” (存取) 属性。

001B	00C8	00D2	0000	007C
0000	0000	0000	0000	0000
0000	(007C)	FF8C	0000	0000
0000	0000	0000	0050	0000

- 显示的 “access” (存取) 属性在下载目标程序时初始化。
- 在如下所示的某些工作条件下，更新显示的间隔时间可能会比指定间隔长。
 - 主机执行/装入条件
 - 通信接口
 - 窗口大小 (存储器显示范围) 或显示的窗口数

7.1.1 扩展菜单

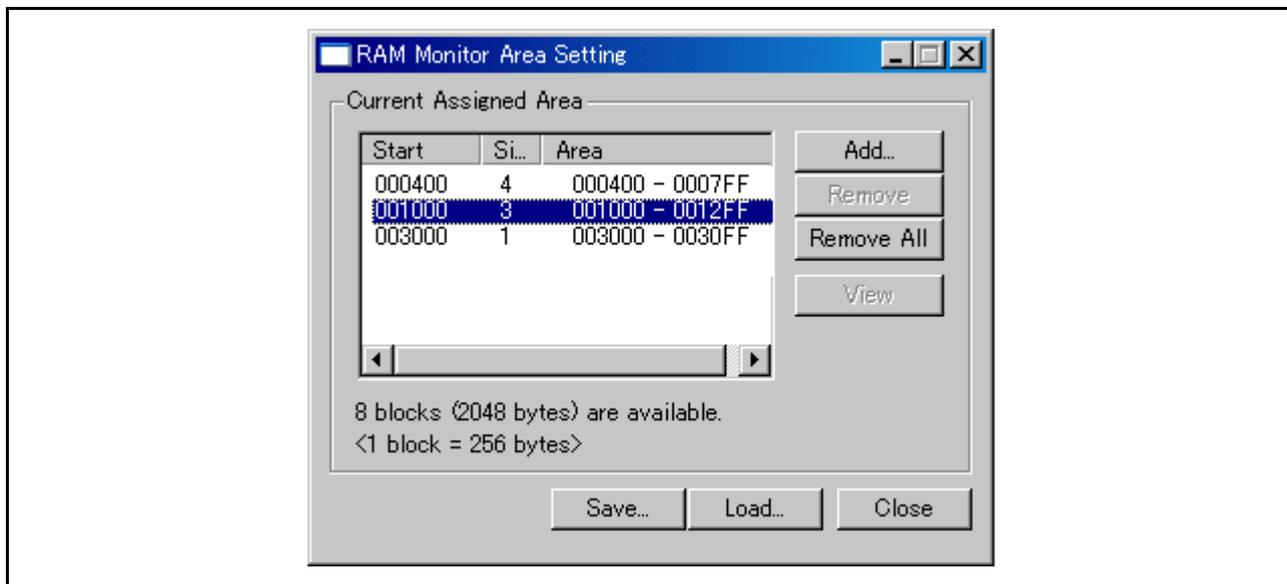
此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能	
[RAM Monitor Area...] (RAM 监控区域...)	设置 RAM 监控基址。	
[Sampling Period...] (采样周期...)	设置 RAM 监控采样周期。	
[Clear] (清除)	清除 “access” (存取) 属性。	
[Up] (上移)	将显示位置移至紧邻的上一 RAM 监控区域 (较小地址)	
[Down] (下移)	将显示位置移至紧邻的下一 RAM 监控区域 (较大地址)	
[Address...] (地址...)	从指定地址显示。	
[Scroll Area...] (滚动区域...)	指定滚动范围。	
[Data Length] (数据长度)	[1byte] (1 字节)	以 1 字节为单位显示。
	[2bytes] (2 字节)	以 2 字节为单位显示。
	[4bytes] (4 字节)	以 3 字节为单位显示。
	[8bytes] (8 字节)	以 8 字节为单位显示。
[Radix] (基数)	[Hex] (十六进制)	以十六进制显示。
	[Dec] (十进制)	以十进制显示。
	[Signed Dec] (带符号的十进制)	以带符号的十进制显示。
	[Oct] (八进制)	以八进制显示。
	[Bin] (二进制)	以二进制显示。
[Code] (代码)	[ASCII]	显示为 ASCII 字符。
	[SJIS]	显示为 SJIS 字符。
	[JIS]	显示为 JIS 字符。
	[UNICODE]	显示为 UNICODE 字符。
	[EUC]	显示为 EUC 字符。
	[Float] (浮点值)	显示为浮点值。
	[Double] (双精度值)	显示为双精度浮点值。
[Layout] (布局)	[Label] (标签)	在显示或不显示标签区域之间切换。
	[Register] (寄存器)	在显示或不显示寄存器区域之间切换。
	[Code] (代码)	在显示或不显示代码区域之间切换。
[Column...] (列...)	设置一行所显示的列数。	
[Split] (拆分)	拆分窗口。	
[Toolbar display] (工具栏显示)	显示工具栏。	
[Customize toolbar...] (自定义工具栏...)	打开工具栏自定义对话框。	
[Allow Docking] (允许入坞)	允许窗口入坞。	
[Hide] (隐藏)	隐藏窗口。	

7.1.2 设置 RAM 监控区域

在 [RAM monitor] (RAM 监控) 窗口中选择弹出菜单 [RAM Monitor Area...] (RAM 监控区域...)。

此时将出现如下所示的 [RAM monitor area setup] (RAM 监控区域设置) 窗口。此窗口中列出当前设置的 RAM 监控区域。



使用此窗口可添加、删除或更改 RAM 监控区域。

- 通过指定 RAM 监控区域的起始地址和大小 (后者通过块数指定) 对其进行指定。
- 起始地址可以按 0 x 100 字节单位指定。

如果指定了一个不符合上述要求的地址值, 该值会就近舍入为符合 0x100 字节单位的地址, 然后才能用于设置。

- 通过指定块的数目指定 RAM 监控区域的大小。

对于 PC7501, 一个块的大小为 256 字节。最多可指定 16 个块。

- 在使用的块数达到 16 之前可以继续添加 RAM 监控区域。

(当前可用的块的数目 (和大小) 显示于列表下方。)

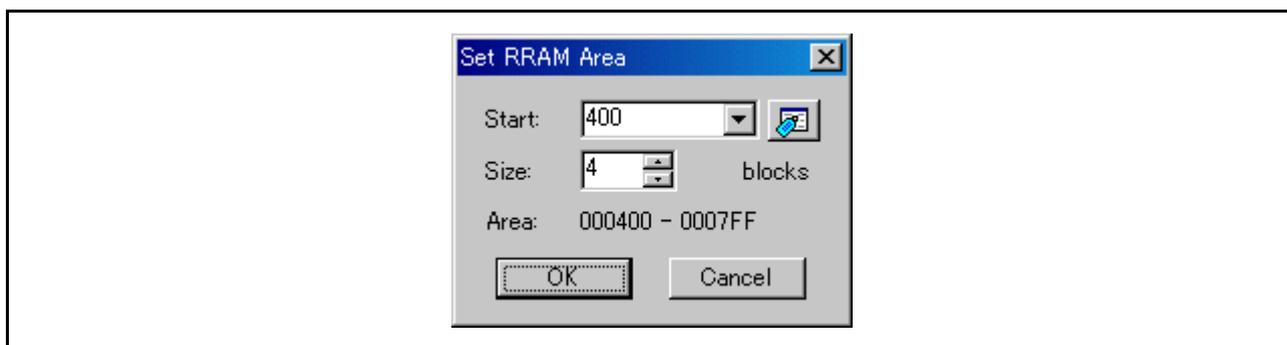
7.1.2.1 更改 RAM 监控区域

RAM 监控区域的起始地址和大小可以更改。

- 从对话框更改

从 RAM 监控区域列表中选择要更改的 RAM 监控区域，并在其上双击。

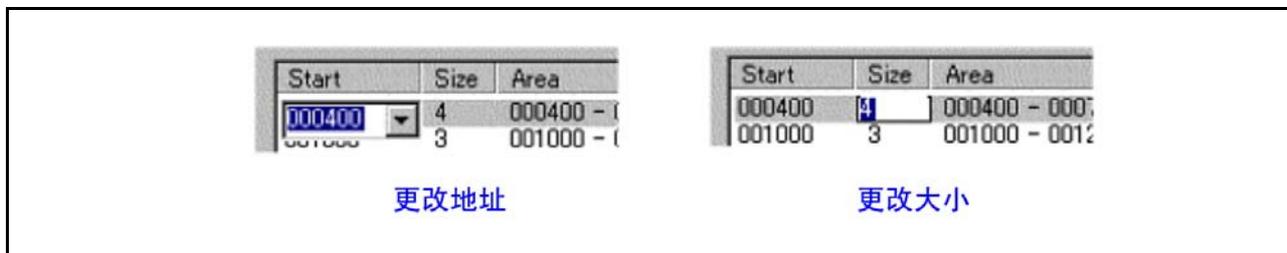
此时将出现如下所示的 [Set RRAM Area]（设置 RRAM 区域）对话框。在此对话框的“Start”（起始）和“Size”（大小）栏中指定 RAM 监控区域的起始地址和大小（通过块数指定）。



- 直接在窗口中更改

从 RAM 监控区域列表中选择要更改的 RAM 监控区域，并在其“Start”（起始）或“Size”（大小）显示列中再次单击。

在随后显示的编辑框中指定要更改的新起始地址或大小。按 Enter 键确认输入的更改，或按 Esc 键取消更改。



7.1.2.2 添加 RAM 监控区域

单击 [Add...]（添加...）按钮。

此时将出现 [Set RRAM Area]（设置 RRAM 区域）对话框。在此对话框的“Start”（起始）和“Size”（大小）栏中指定新的 RAM 监控区域的起始地址和大小（通过块数指定）。

7.1.2.3 删除 RAM 监控区域

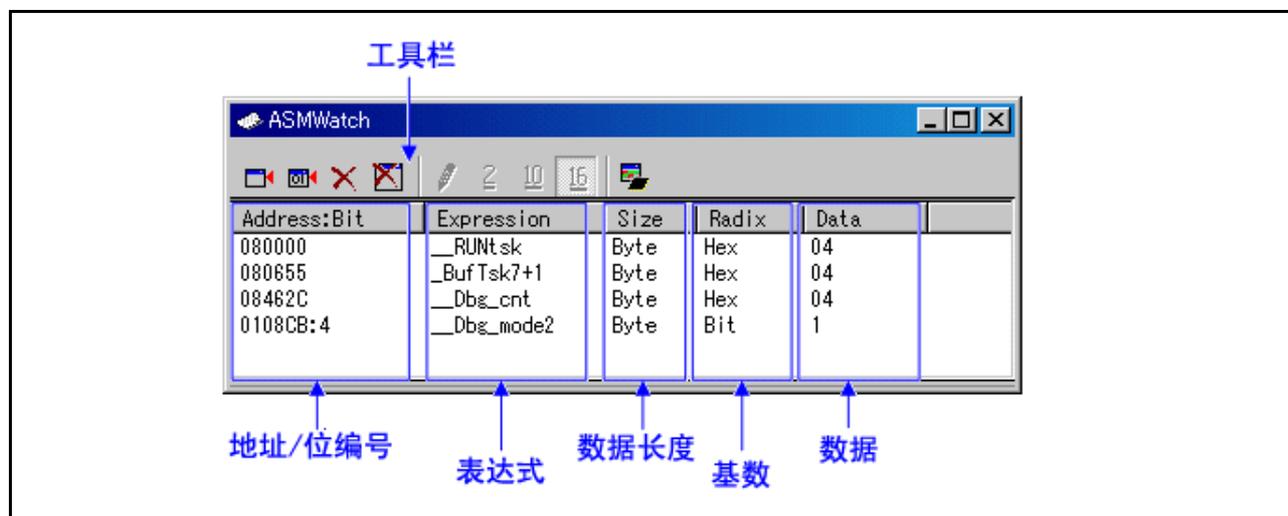
从 RAM 监控区域列表中选择要删除的 RAM 监控区域，然后单击 [Remove]（移除）按钮。

要删除所有 RAM 监控区域，请单击 [Remove All]（全部移除）按钮。

7.2 [ASM Watch] (ASM 监视) 窗口

在 [ASM watch] (ASM 监视) 窗口中，您可以将指定地址注册为监视点，并检查这些地址处的存储器内容。

如果已注册的地址驻留在 RAM 监控区域内，那么在程序执行过程中，该地址处的存储器内容将按给定间隔进行更新（默认情况下间隔时间为 100 ms）。



- 要注册的地址称为“监视点”。可通过以下方式之一指定：
 - 地址（可使用符号指定）
 - 地址 + 位号
 - 位符号
- 关闭 [ASM watch] (ASM 监视) 窗口时，已注册的监视点将保存在调试器中，并在再次打开窗口时自动完成注册。
- 如果为监视点指定了符号或位符号，则下载目标程序时会重新计算监视点地址。
- 屏幕上所显示的无效监视点将以 “-<not active>-” 标识。
- 监视点的列表顺序可通过拖放操作更改。
- 监视点的 “expressions” (表达式)、 “sizes” (大小)、 “radixes” (基数) 和 “datas” (数据) 可通过在原位置编辑来更改。

注意事项

- RAM 监控获取通过总线存取的数据。除了从目标程序进行存取外，其他任何更改均不会反映在此。
- 如果 RAM 监控区域的数据显示长度不是 1 字节，该数据中各字节可能具有不同的存储器 “access” (存取) 属性。如果一组数据中的 “access” (存取) 属性不统一，则该数据的 “access” (存取) 属性可能不会正确显示。在这种情况下，从背景颜色可以看出数据中第一个字节的 “access” (存取) 属性。

7.2.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

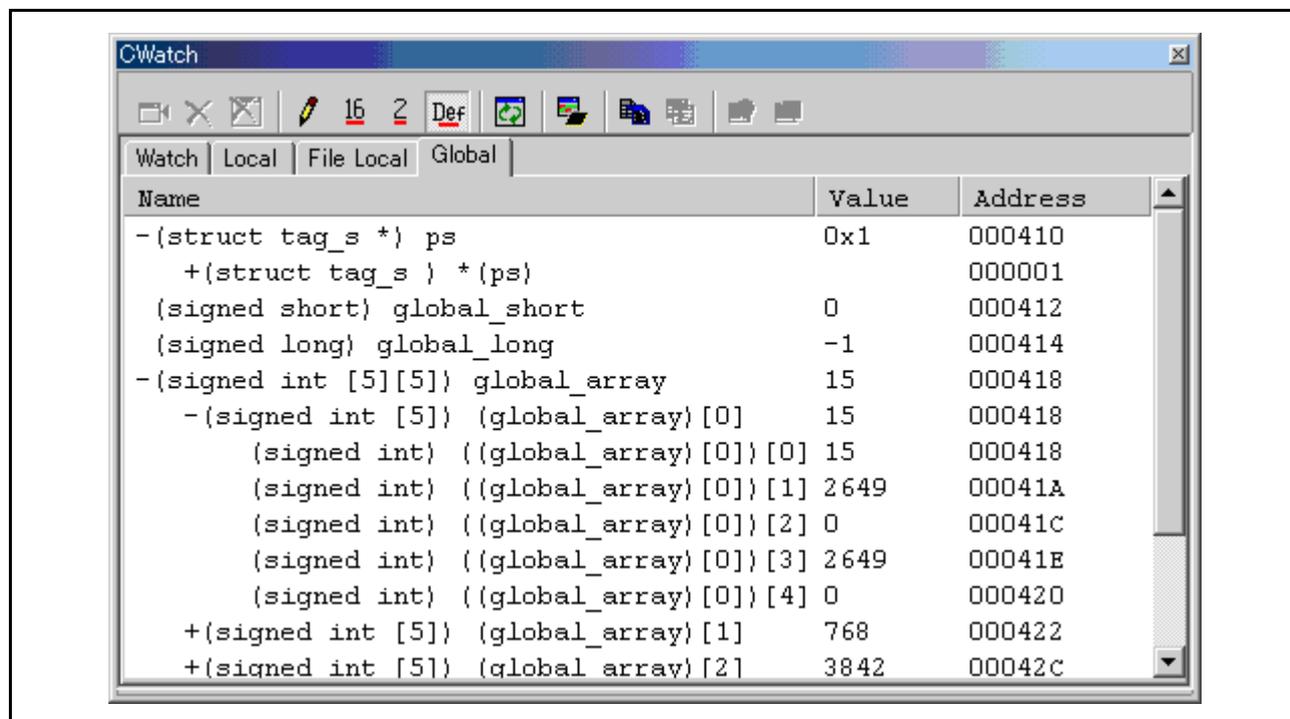
菜单		功能
[Add...] (添加...)		添加监视点。
[Add Bit...] (添加位...)		添加位标签监视点。
[Remove] (移除)		移除选定的监视点。
[Remove All] (全部移除)		移除全部监视点。
[Set...] (设置...)		对选定监视点设置新数据。
[Radix] (基数)	[Bin] (二进制)	以二进制显示。
	[Dec] (十进制)	以十进制显示。
	[Hex] (十六进制)	以十六进制显示。
[Refresh] (刷新)		刷新存储器数据。
[Layout] (布局)	[Address Area] (地址区域)	在显示或不显示地址区域之间切换。
	[Size Area] (大小区域)	在显示或不显示大小区域之间切换。
[RAM Monitor] (RAM 监控)	[Enable RAM Monitor] (允许 RAM 监控)	在允许或禁止 RAM 监控功能之间切换。
	[Sampling Period...] (采样周期...)	设置 RAM 监控采样周期。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

7.3 [C Watch] (C 监视) 窗口

[C Watch] (C 监视) 窗口将显示 C/C++ 表达式及其值 (计算结果)。

[C Watch] (C 监视) 窗口中显示的 C/C++ 表达式通常称为 C 监视点。C 监视点的计算结果显示在每次执行命令时更新。

如果 RAM 监控功能有效, 并且 C 监视点位于 RAM 监控区域内, 则显示的值在目标程序执行期间更新。



- 可以按作用域 (局部、文件局部或全局变量) 检查变量。
- 显示内容在 PC 值发生改变时自动更新。
- 变量值可以更改。
- 可分别更改各个变量的显示基数。
 - 初始显示基数可以更改。
 - 对于十六进制显示, 前导零抑制为可选。
- 任何变量均可注册到 [Watch] (监视) 选项卡, 这样该变量即可始终显示:
 - 各个工程的已注册内容将分别保存。
 - 同时打开两个或两个以上的 [C watch] (C 监视) 窗口时, [Watch] (监视) 选项卡中的注册内容将在多个窗口间共享。
 - 变量的引用作用域可从当前作用域、全局作用域以及各文件的作用域中选择。
- 通过添加 [Watch] (监视) 选项卡, 可将 C 监视点注册到不同的目标中。
- 通过拖放操作, 可从另一窗口或编辑器注册变量。
- C 监视点可以按名称或地址排序。
- 使用 RAM 监控功能, 在程序执行过程中可以对值进行实时检查。
- RAM 监控功能可置于指定变量的地址处。

注意事项

- 以下所列 C 监视点的值无法更改：
 - 寄存器变量
 - 未显示地址的 C 监视点（无效的 C 监视点）
- 如果某 C/C++ 语言表达式无法正确计算（例如，存在未定义的 C/C++ 符号），此表达式将注册为无效的 C 的监视点，
显示为 “--<not active>--”。如果第二次这个 C/C++ 语言表达式可以进行正确计算，那么它将变为有效的 C 监视点。
- [Local]（局部）、[File Local]（文件局部变量）和 [Global]（全局）选项卡的显示设置无法保存。
[Watch]（监视）选项卡的内容和新添加的选项卡将得以保存。
- RAM 监控获取通过总线存取的数据。除了从目标程序进行存取外，其他任何更改均不会反映在此。
- 实时更改的变量只有全局变量和文件局部变量。
- 如果 RAM 监控区域的数据显示长度不是 1 字节，该数据中各字节可能具有不同的存储器 “access”（存取）属性。如果一组数据中的 “access”（存取）属性不统一，则该数据的 “access”（存取）属性可能不会正确显示。在这种情况下，从背景颜色可以看出数据中第一个字节的 “access”（存取）属性。

有关 C 变量的更多信息，请参阅 “12.1.3 获取或设置 C 变量”。

7.3.1 扩展菜单

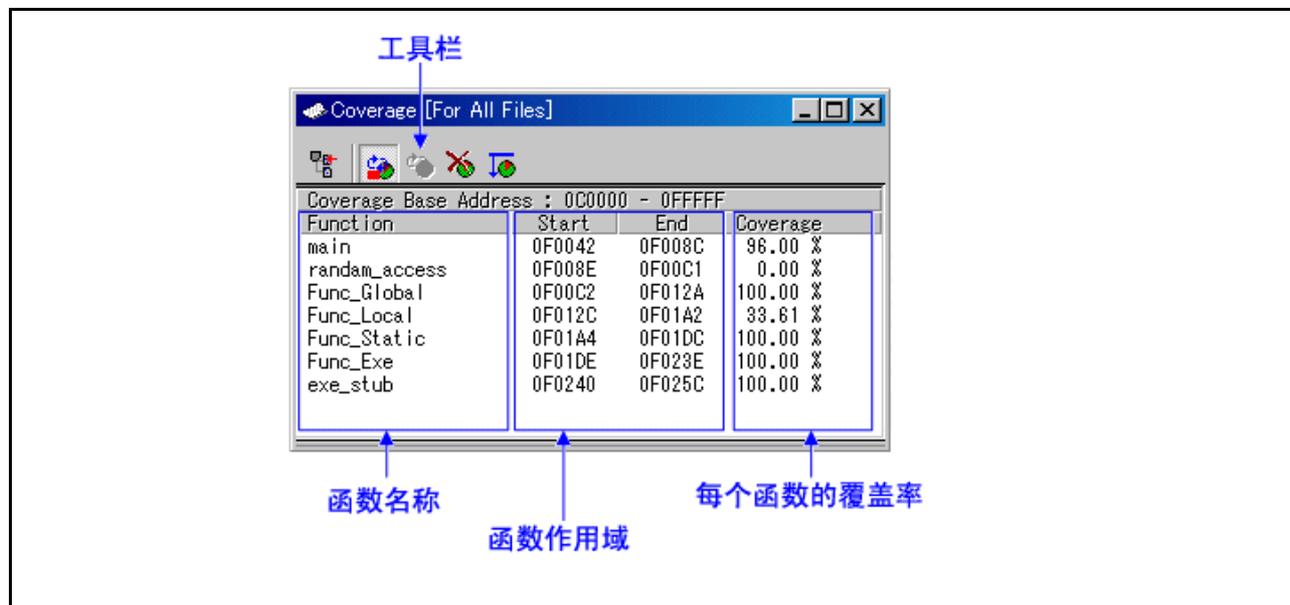
此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Add...] (添加...)		添加 C 监视点。
[Remove] (移除)		移除选定的 C 监视点。
[Remove All] (全部移除)		移除全部 C 监视点。
[[Initialize] (初始化)		对选定的 C 监视点重新求值。
[Set New Value...] (设置新值...)		对选定的 C 监视点设置新数据。
[Radix] (基数)	[Hex] (十六进制)	以十六进制显示。
	[Bin] (二进制)	以二进制显示。
	[Default] (默认)	显示默认基数。
	[Toggle(All Variables)] (切换 (全部变量))	更改基数 (切换)。
	[Set initial...] (设置初始基数...)	设置初始基数。
[Refresh] (刷新)		刷新存储器数据。
[Hide type name] (隐藏类型名称)		隐藏变量的类型名称。
[Show char* as string] (将 char* 显示为 string)		选择是否将 char* 显示为 string。
[Zero suppress in Hex display] (十六进制显示中消零)		十六进制显示中消零。
[Sort] (排序)	[Sort by Name] (按名称排序)	按变量名称排序。
	[Sort by Address] (按地址排序)	按变量地址排序。
[RAM Monitor] (RAM 监控)	[Enable RAM Monitor] (允许 RAM 监控)	在允许或禁止 RAM 监控功能之间切换。
	[Sampling Period...] (采样周期...)	设置 RAM 监控采样周期。
	[Arrange a RAM monitor area around this variable] (在此变量旁 布置 RAM 监控区域)	在此变量旁布置 RAM 监控区域。
	[Start Recording...] (开始记录...)	开始记录更新的值。
	[Stop Recording] (停止记录)	停止记录更新的值。
[Add New Tab...] (添加新选项卡...)		添加新选项卡。
[Remove Tab] (移除选项卡)		移除选定的选项卡。
[Copy] (复制)		将所选项目复制到剪贴板。
[Copy All] (全部复制)		将页中所有项目复制到剪贴板。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

7.4 [Coverage] (覆盖率) 窗口

您可以从 [Coverage] (覆盖率) 窗口引用已下载的目标程序中所含函数的覆盖率计量结果。可计量的覆盖率是 C0 覆盖率。

此时可使用两种窗口: [Coverage] (覆盖率) 窗口, 您可以在此窗口中查看函数起始地址 / 结束地址以及覆盖率计量结果; [Editor] (编辑器) 窗口, 您可以在此窗口中以源代码形式查看执行 / 非执行情况。



- 覆盖率计量区域的块数可在 1 至 32 之间任选, 每块最大 8MB, 其中的块均以 256 字节分界线为起始。
- 双击任一函数行, 相应的函数便出现在 [Editor(Source)] (编辑器 (源码)) 窗口中。
- 覆盖率计量过程中, 覆盖率显示区域中出现 “-%”。
- 您可以通过鼠标操作改变函数名显示区域和函数范围显示区域之间的显示比率。

7.4.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Select source file...] (选择源文件...)		选择要查看覆盖率的源文件。
[Auto Refresh] (自动刷新)		自动刷新覆盖率计量结果。
[Refresh] (刷新)		刷新覆盖率计量结果。
[Clear] (清除)		清除覆盖率计量结果。
[Base...] (基地址...)		更改覆盖率基地址。
[File] (文件)	[Save...] (保存...)	将覆盖率计量结果保存到文件。
	[Load...] (装入...)	从文件装入覆盖率计量结果。
[Layout] (布局)	[Address] (地址)	在显示或不显示地址区域之间切换。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

*: 不支持 M16C/R8C 系列的调试器，因为该调试器的整个存储器都属于覆盖率计量区域。

7.4.2 查看源代码 / 已执行地址

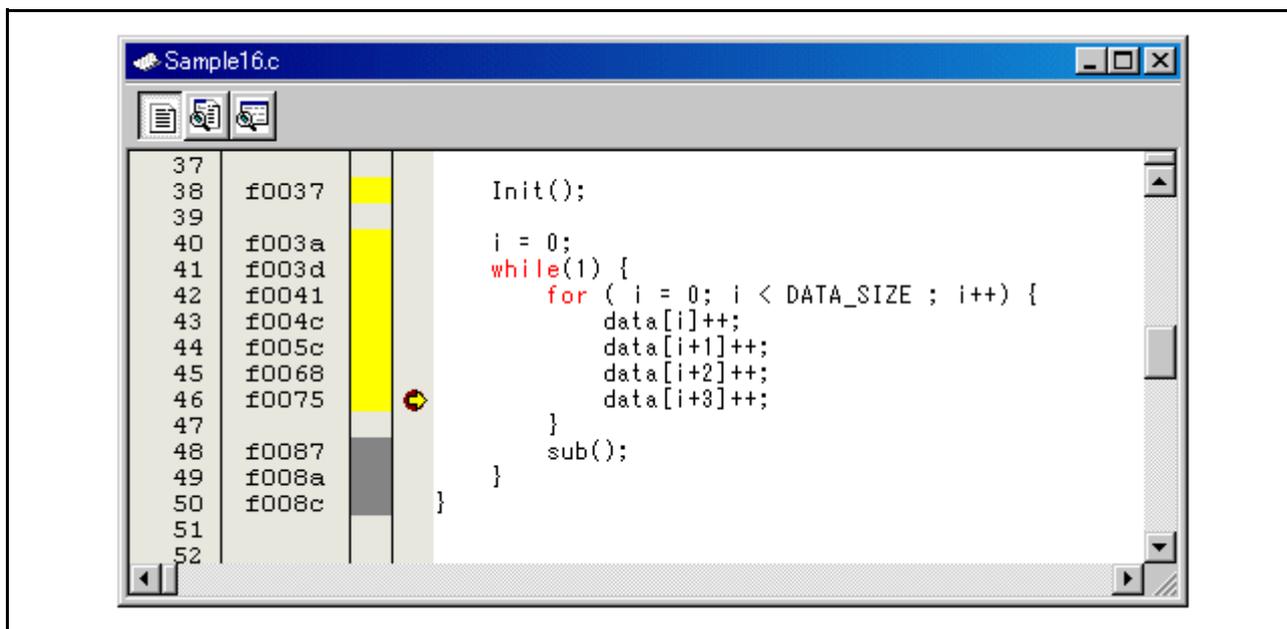
可以在 [Editor(Source)] (编辑器 (源码)) 窗口或 [Memory] (存储器) 窗口中进行查看。

7.4.2.1 在 [Editor(Source)] (编辑器 (源码)) 窗口中查看

在 [Editor(Source)] (编辑器 (源码)) 窗口中, “Coverage Measurement” (覆盖率计量) 显示在默认情况下设置为 “Disable” (禁止)。

要允许显示, 选择主菜单 - [Edit] (编辑) -> [Define Column Format] (定义列格式), 在出现的对话框中选中 [Coverage] (覆盖率) 复选框。此时所有 [Editor(Source)] (编辑器 (源码)) 窗口中都会出现一列, 用于显示覆盖率计量。

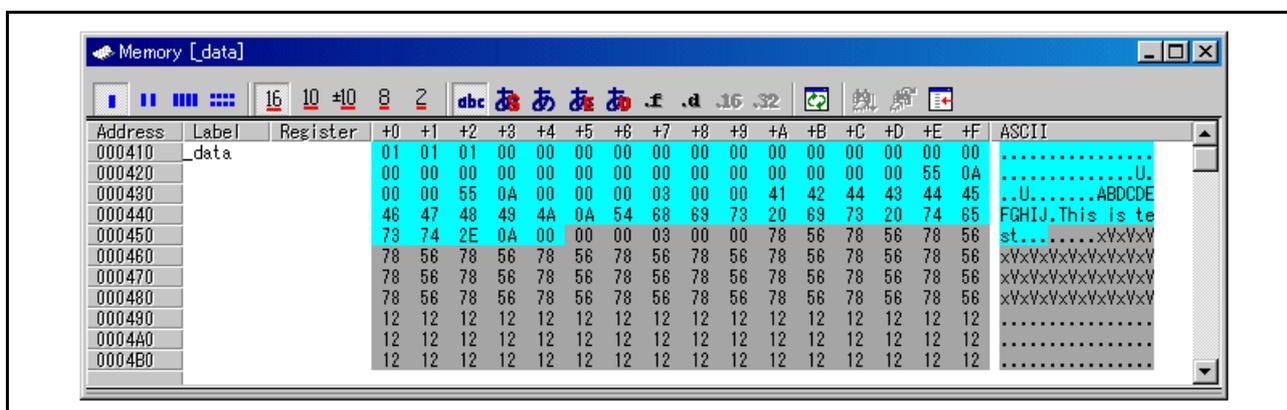
也可以在 [Editor(Source)] (编辑器 (源码)) 窗口中选择弹出菜单 - [Columns] (列) -> [Coverage] (覆盖率), 对各 [Editor(Source)] (编辑器 (源码)) 窗口添加这一列。



7.4.2.2 在 [Memory] (存储器) 窗口中查看

在 [Memory] (存储器) 窗口中, “Coverage Measurement” (覆盖率计量) 显示在默认情况下设置为 “Disable” (禁止)。

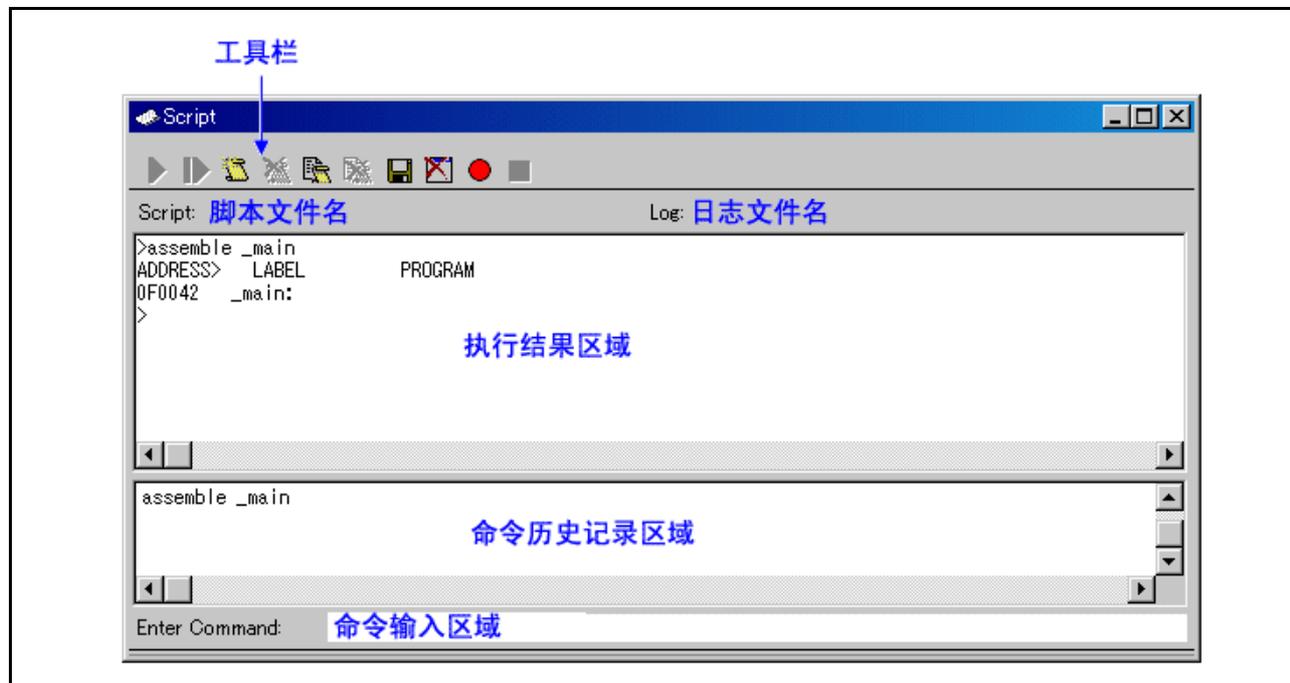
要允许该显示, 请在 [Memory] (存储器) 窗口中选择弹出菜单 - [Coverage] (覆盖率) -> [On/Off] (开/关)。



7.5 [Script] (脚本) 窗口

[Script] (脚本) 窗口显示文本形式脚本命令的执行及执行的结果。

脚本命令可通过脚本文件执行，或者以交互方式执行。您也可以将脚本命令写入脚本文件，从而自动执行脚本命令。脚本命令的执行结果还可以存储到之前指定的日志文件中。



- [Script] (脚本) 窗口有一个显示缓冲区，其中存储前 1000 行的执行结果。因此执行结果可存储在文件（显示文件）中，无需指定日志文件。
- 打开脚本文件后，命令历史区域将变为脚本文件显示区域，并显示脚本文件的内容。当脚本文件为嵌套文件时，则会显示最后打开的脚本文件。脚本文件显示区域将以反色形式显示当前正在执行的行。
- 打开脚本文件后，如果脚本文件尚未执行，您可以从命令输入区域调用脚本命令。
- [Script] (脚本) 窗口可将已执行的历史命令记录到文件。此功能与日志功能并不相同。它只记录已执行的命令，并不记录命令执行结果，因此所保存的文件可用作脚本文件。

7.5.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能	
[Script] (脚本)	[Open...] (打开...)	打开脚本文件。
	[Run] (运行)	运行脚本文件。
	[Step] (单步执行)	单步执行脚本文件。
	[Close] (关闭)	关闭脚本文件。
[View] (视图)	[Save...] (保存...)	将显示缓冲区保存到文件。
	[Clear] (清除)	清除显示缓冲区。
[Log] (日志)	[On...] (开...)	打开日志文件并开始记录 (开始输出到文件)。
	[Off] (关)	关闭日志文件并结束记录 (停止输出到文件)。
[Record] (记录)	[On...] (开...)	将已执行的命令记录到文件。
	[Off] (关)	停止记录已执行的文件。
[Copy] (复制)	复制所选内容并将其置于剪贴板。	
[Paste] (粘贴)	插入剪贴板内容。	
[Cut] (剪切)	剪切所选内容并将其置于剪贴板。	
[Delete] (删除)	删除所选内容。	
[Undo] (撤消)	撤消上一操作。	
[Toolbar display] (工具栏显示)	显示工具栏。	
[Customize toolbar...] (自定义工具栏...)	打开工具栏自定义对话框。	
[Allow Docking] (允许入坞)	允许窗口入坞。	
[Hide] (隐藏)	隐藏窗口。	

7.6 [S/W Break Point Setting] (软件断点设置) 窗口

您可以通过 [S/W Break Point Setting] (软件断点设置) 窗口设置软件断点。软件断点将立即停止执行指定断点之前的指令。



- 如果您设置了多个软件断点，则遇到其中任一软件断点地址（或条件）时都会停止执行程序。
- 在单击“Close”（关闭）按钮关闭 [S/W Break Point Setting] (软件断点设置) 窗口之前，您可以连续设置软件断点。
- 在软件断点显示区域可通过单击操作清除、允许或禁止所选软件断点。您也可以双击软件断点将其允许或禁止。
- 单击“Save”（保存）按钮，将软件断点保存到文件。要从已保存的文件再装入软件断点设置，请单击“Load”（装入）按钮。如果从文件装入软件断点，这些断点将被添加到现有断点。

7.6.1 命令按钮

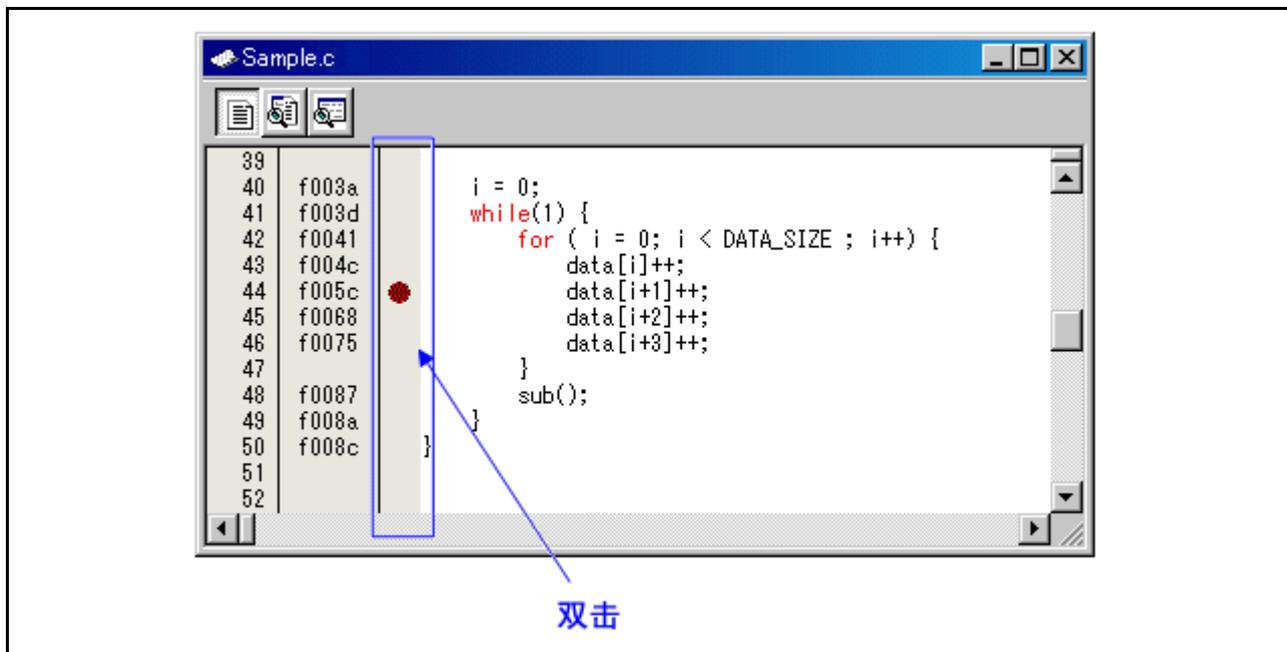
此窗口中的按钮具有以下含义。

按钮	功能
[Load...] (装入...)	从保存了设置信息的文件装入设置信息。
[Save...] (保存...)	将窗口中设置的内容保存到文件中。
[Help] (帮助)	显示此窗口的帮助信息。
[Add] (添加)	添加断点。
[Refer...] (引用...)	打开文件选择对话框。
[Close] (关闭)	关闭窗口。
[Delete] (删除)	移除选定断点。
[Delete All] (全部删除)	移除所有断点。
[Enable] (允许)	允许所选断点。
[All Enable] (全部允许)	允许所有断点。
[Disable] (禁止)	禁止所选断点。
[All Disable] (全部禁止)	禁止所有断点。
[View] (视图)	在 [Editor(Source)] (编辑器 (源)) 窗口中显示所选断点位置。

7.6.2 从 [Editor(Source)] (编辑器 (源)) 窗口设置和删除断点

可设置软件断点的区域因产品而异。有关详细信息，请参阅“12.1.2 可设置软件断点的区域”。

可以在 [Editor(Source)] (编辑器 (源)) 窗口中设置断点。为此，对于要在其中设置断点的行，双击对应的断点设置区域 (“S/W breakpoints” (软件断点) 列)。(设置了断点的行将显示一个红色标记。)

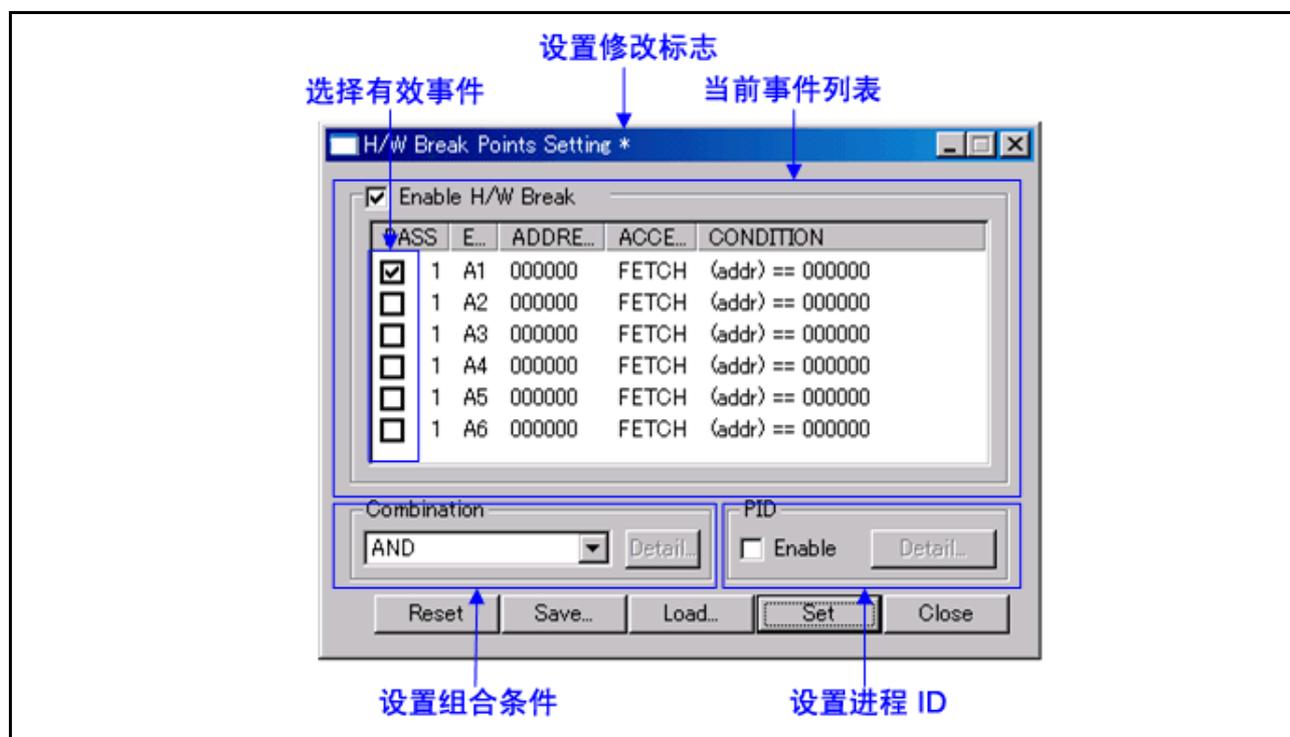


在断点设置区域 (“S/W breakpoints” (软件断点) 列) 中再次双击即可删除该断点。

在 [Editor(Source)] (编辑器 (源)) 窗口中，“S/W breakpoints” (软件断点) 列显示在默认情况下设置为“Enable” (允许)。要删除此列显示，选择主菜单 - [Edit] (编辑) -> [Define Column Format] (定义列格式)，在出现的对话框中取消选择 [S/W breakpoints] (软件断点) 复选框。此时将从所有 [Editor(Source)] (编辑器 (源)) 窗口删除 “S/W breakpoints” (软件断点) 列显示。也可以在 [Editor(Source)] (编辑器 (源)) 窗口中选择弹出菜单 - [Columns] (列) -> [S/W breakpoints] (软件断点)，对各 [Editor(Source)] (编辑器 (源)) 窗口添加这一列。

7.7 [H/W Break Point Setting] (硬件断点设置) 窗口

[H/W Breakpoint Setting] (硬件断点设置) 窗口用于为仿真器设置硬件断点。



- 下列事件可指定为断点事件。如果事件内容发生变化，则事件的标题栏上会标上一个星号 (*)。设置仿真器后不再显示星号 (*)。

Fetch (取)、Memory Access (存储器存取)、Bit Access (位存取)、Interrupt (中断)、External Trigger (外部触发)

- 最多可使用八个点处的事件。
- 可以通过下列方式之一组合这些事件：
 - 建立所有有效事件时暂停 ([AND] (与) 条件)
 - 同时建立所有有效事件时暂停 (同时符合 [AND] (与) 条件)
 - 建立有效事件之一时暂停 ([OR] (或) 条件)
 - 状态转换过程中进入断点状态时暂停 ([State Transition] (状态转换) 条件)
- 启动调试器时硬件断点不起作用。

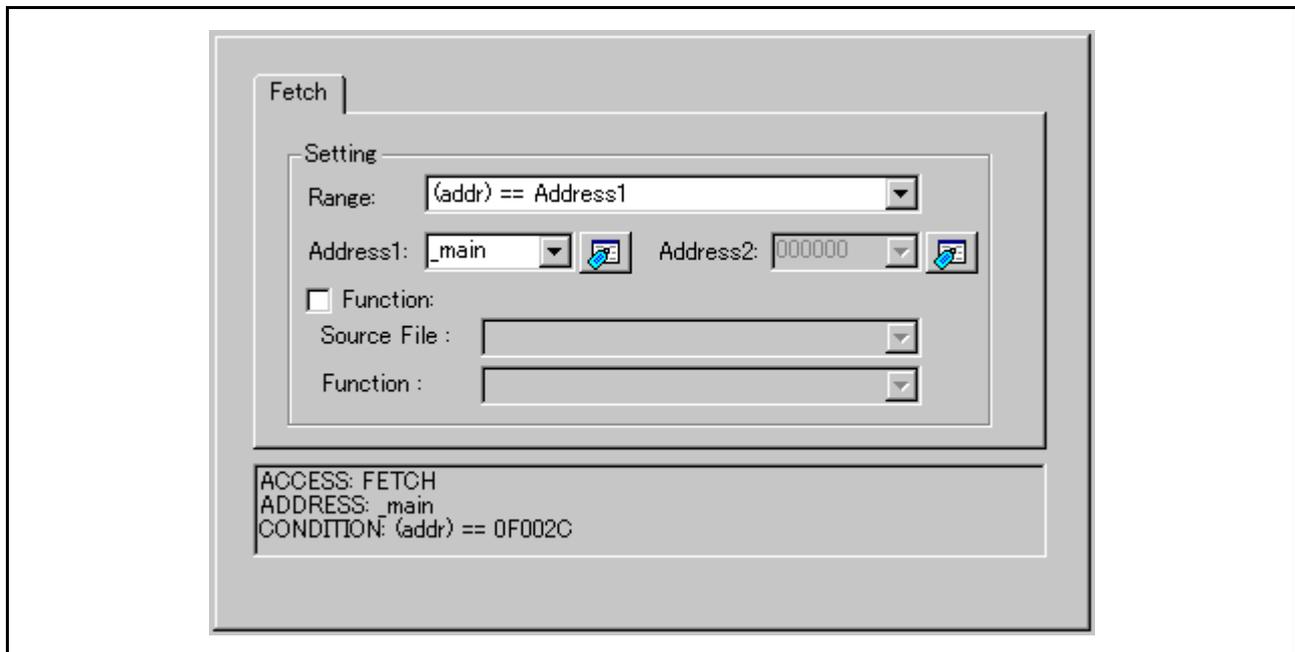
7.7.1 指定断点事件

要设置事件，请从 [H/W Break Point Setting]（硬件断点设置）窗口的事件设置区域双击以选中要设置的事件。此时会打开如下对话框。

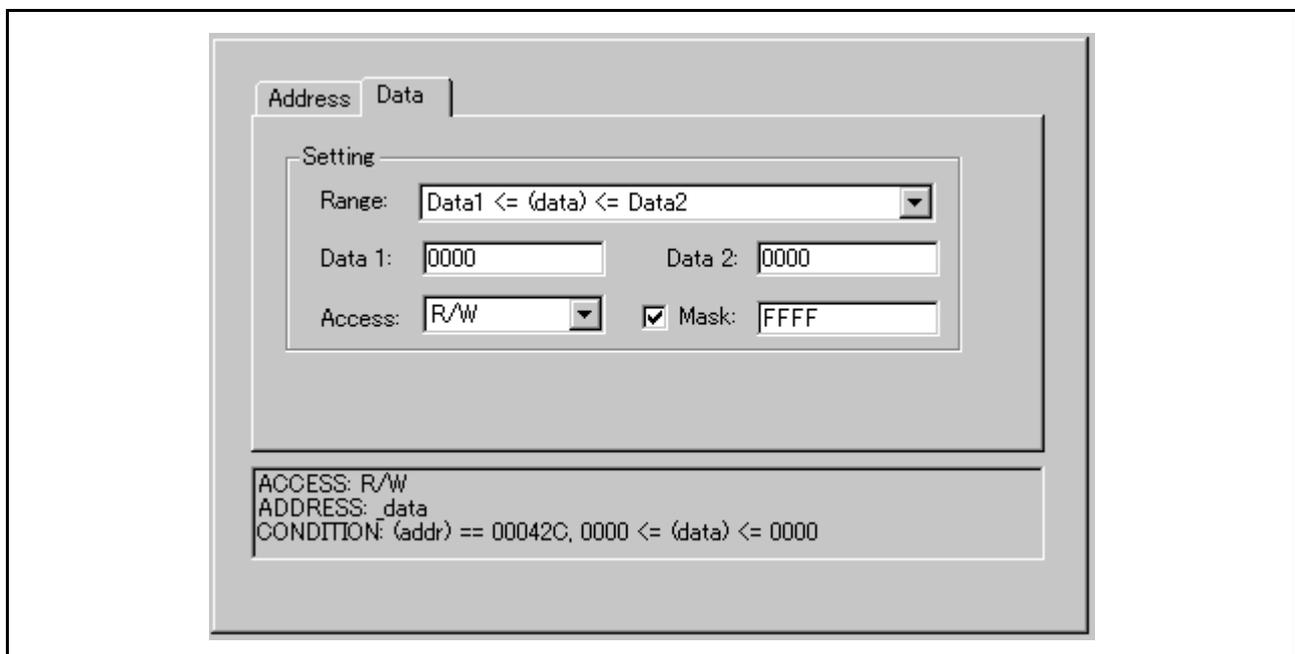


通过在此对话框中指定 [Event Type]（事件类型），可以设置以下事件。

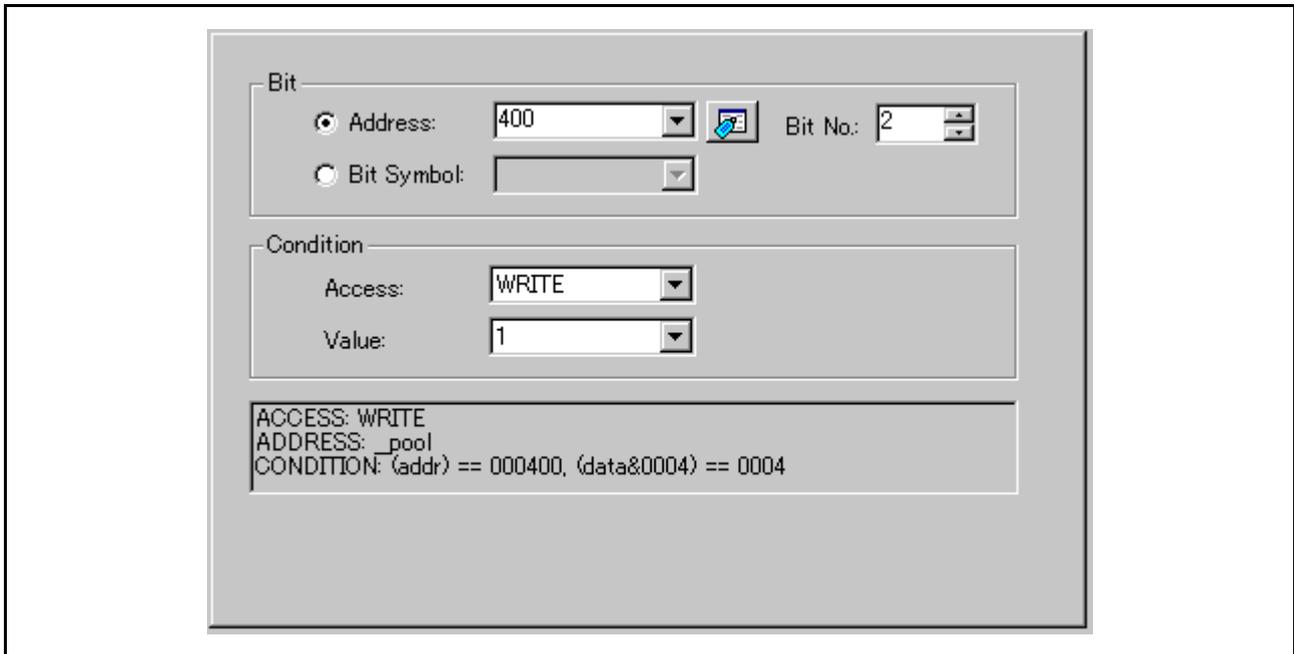
- 选择 FETCH 时
取指令的断点。



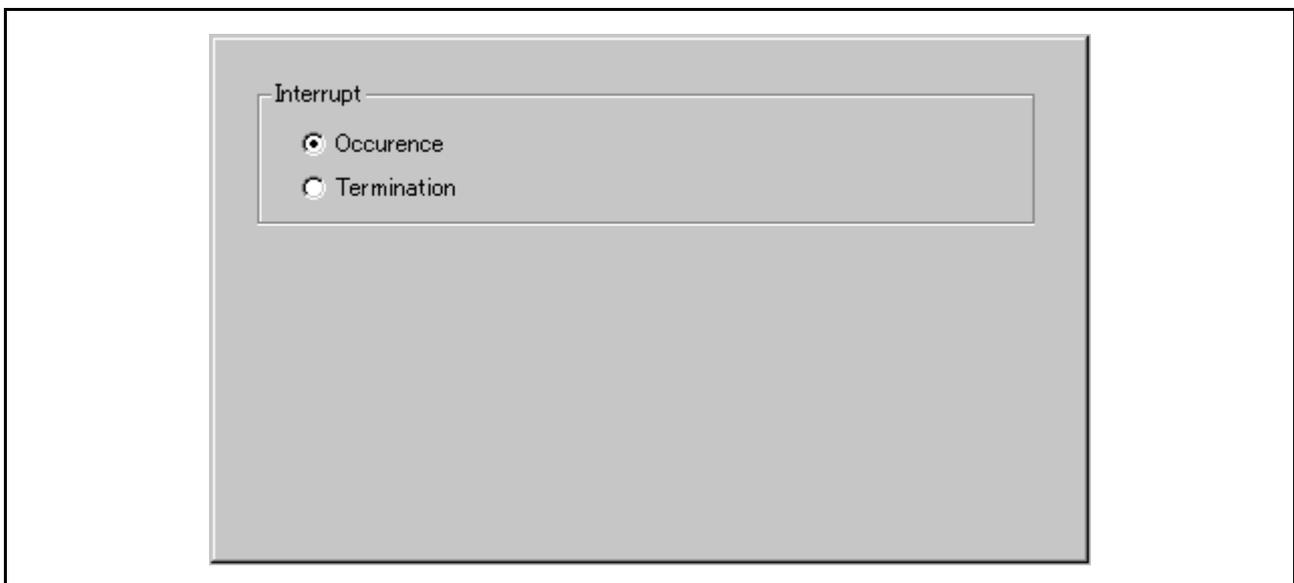
- 选择 DATA ACCESS 时
存储器存取的断点。



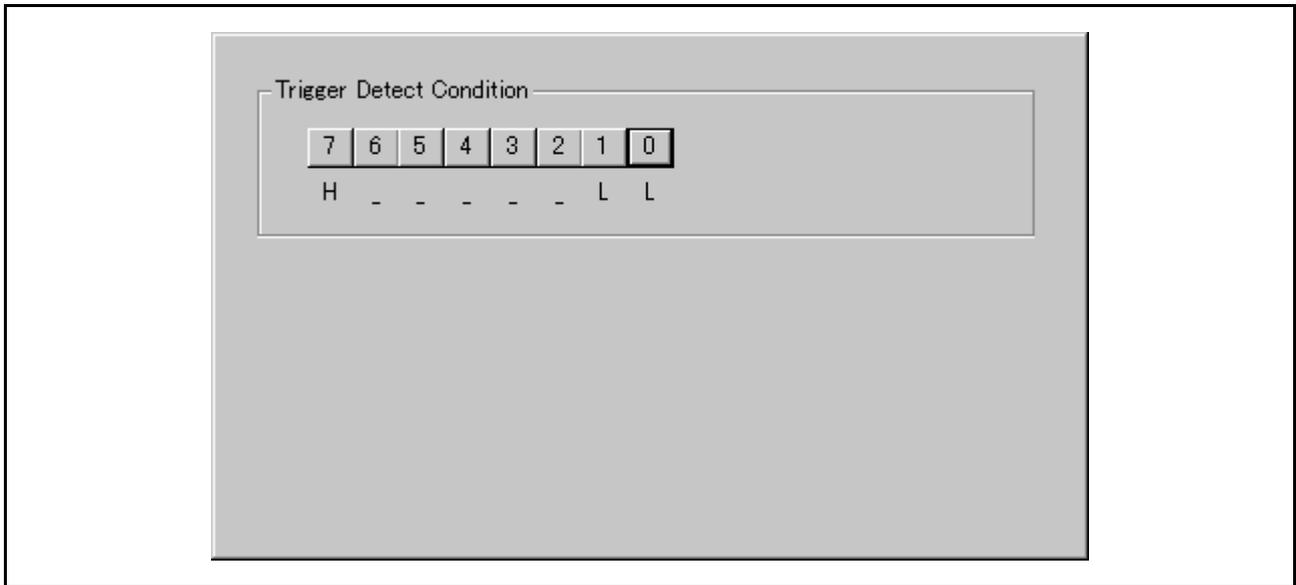
- 选择 BIT SYMBOL 时
位存取的断点。



- 选择 INTERRUPT 时
中断发生或终止的断点。



- 选择 TRIGGER 时
来自外部跟踪电缆的信号输入状态的断点。

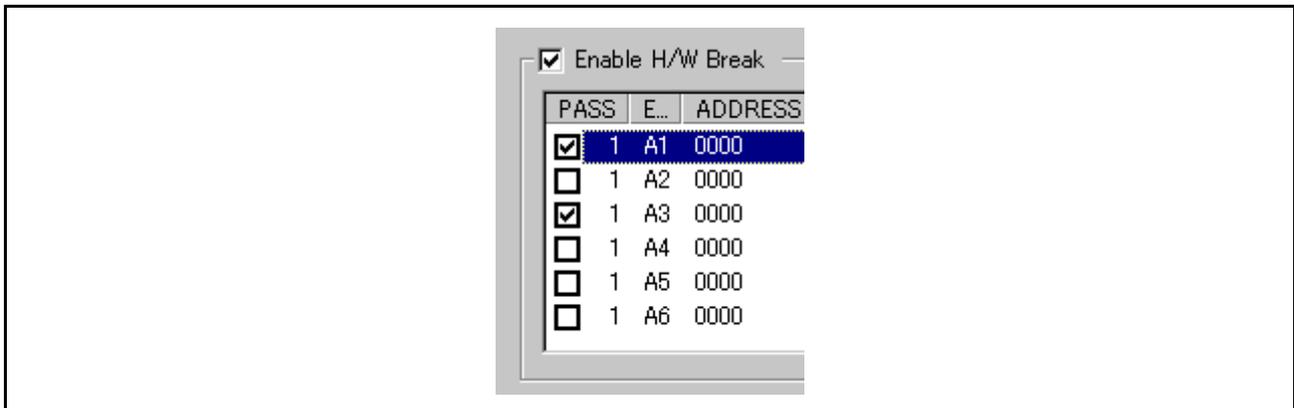


7.7.2 指定组合条件

要指定组合条件，请从组合条件指定区域选择所需条件。

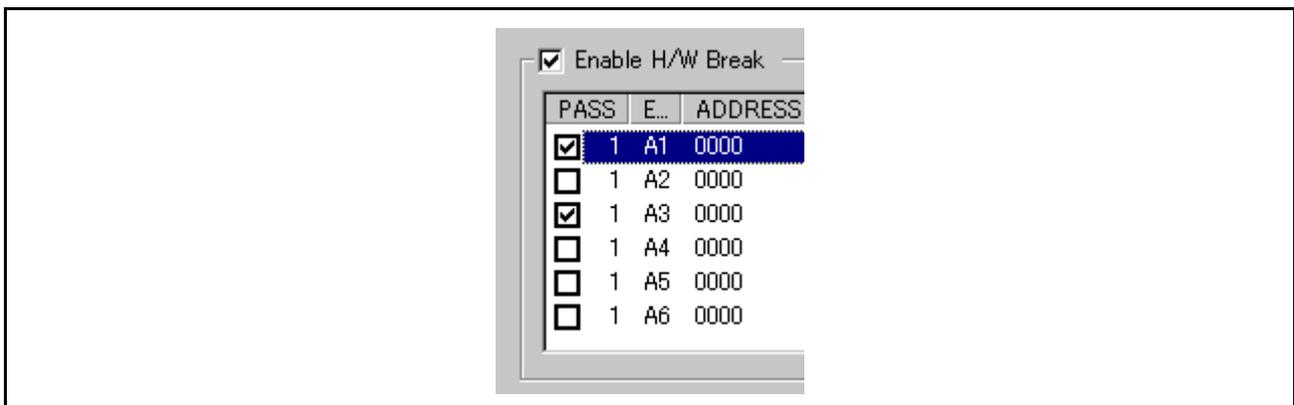
- 选择 [AND]（与）或 [OR]（或）时

在事件指定区域，可以指定所用事件和该事件的通过计数。要更改通过计数，请在要更改的事件处于选中状态时，单击该事件的通过计数值。



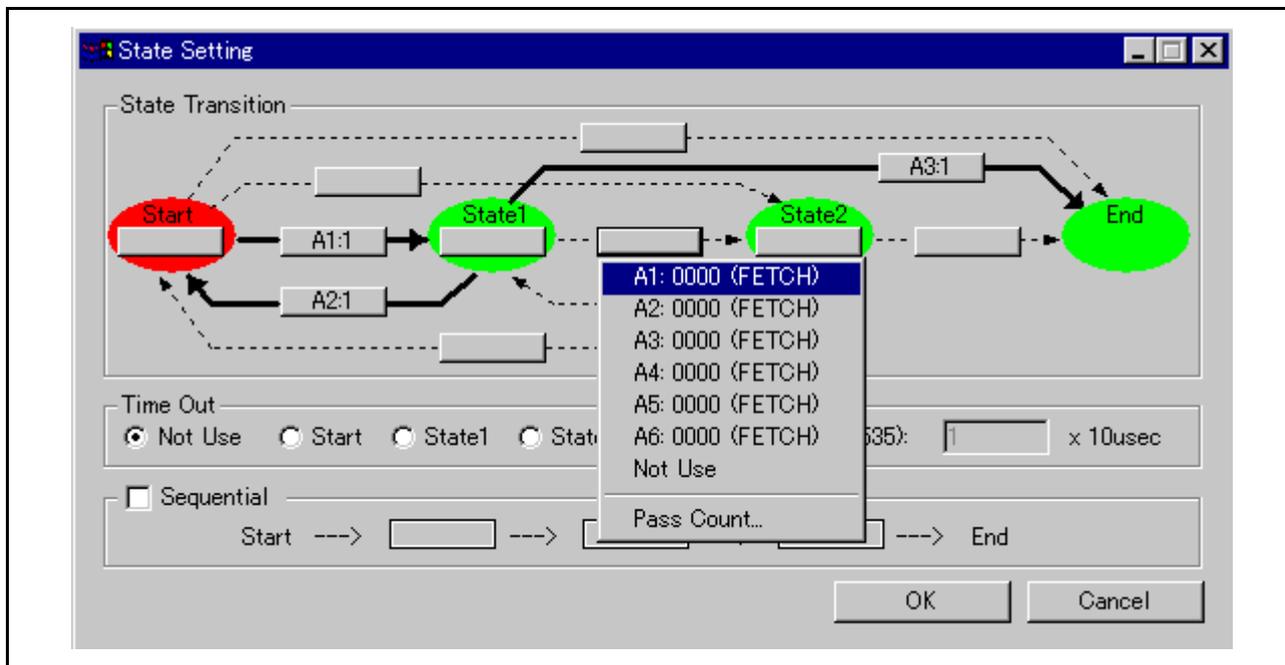
- 选择 [AND (Same Time)]（与（同时））时

在事件指定区域，可以指定所用事件。不能指定通过计数。



- 选择 [State Transition] (状态转换) 时

单击 [Details...] (详细信息...) 按钮，此时会出现如下对话框。可以通过状态转换图指定，也可以通过 [sequential] (顺序) 指定。如果任一事件的内容发生变化，则事件的标题栏上会标上一个星号 (*)。在仿真器中设置了条件后，就不再显示星号。还可以指定各个状态的超时时间。



7.7.3 指定进程 ID

通过指定进程 ID，可以只检测特定条件下的事件建立情况。



示例：仅允许使用实时操作系统时出现在特定任务中的事件。

7.7.4 命令按钮

此窗口中的按钮具有以下含义。

按钮	功能
[Reset] (复位)	删除窗口中显示的内容，并从设置了内容的仿真器装入内容。
[Save...] (保存...)	将窗口中设置的内容保存到文件中。
[Load...] (装入...)	从保存了事件信息的文件装入事件信息。
[Set] (设置)	将窗口中设置的内容发送到仿真器。
[Close] (关闭)	关闭窗口。

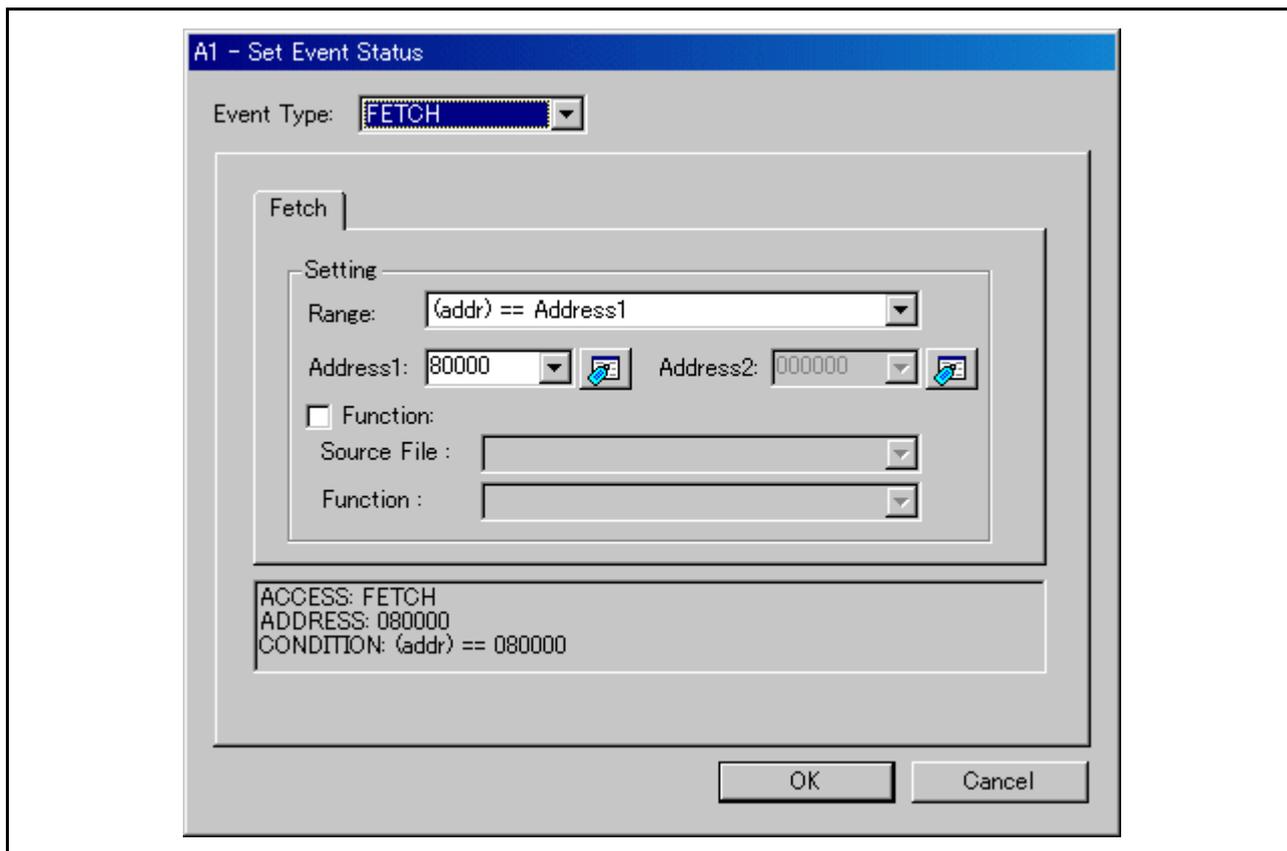
7.7.5 指定事件（取指令）

要指定取指令事件，请将对话框的 [Event Type]（事件类型）更改为“FETCH”。将在从指定地址或指定地址范围中的任意地址取指令时建立事件。

7.7.5.1 从指定地址取指令

设置如下所示。

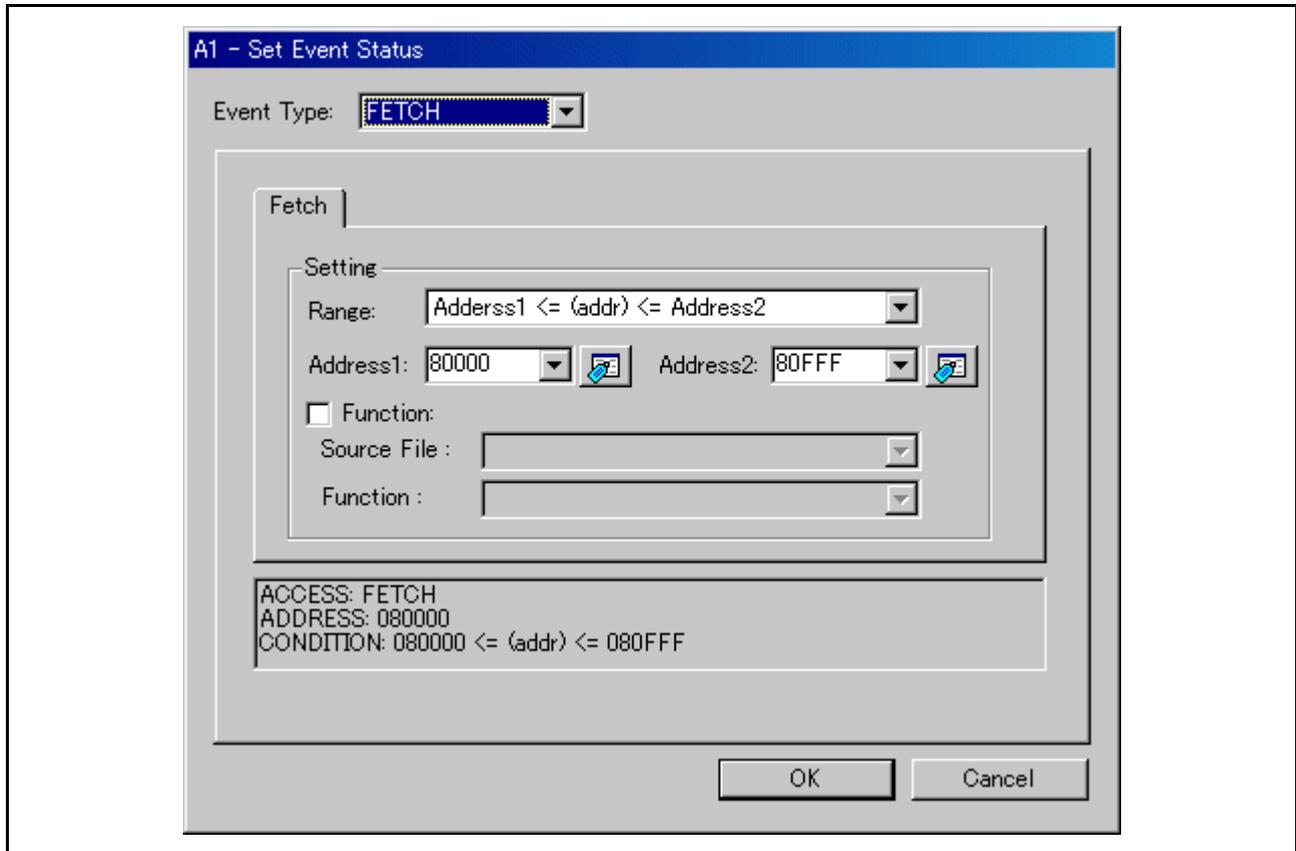
示例) 从地址 80000h 取指令



7.7.5.2 从指定地址范围内取指定

设置如下所示。

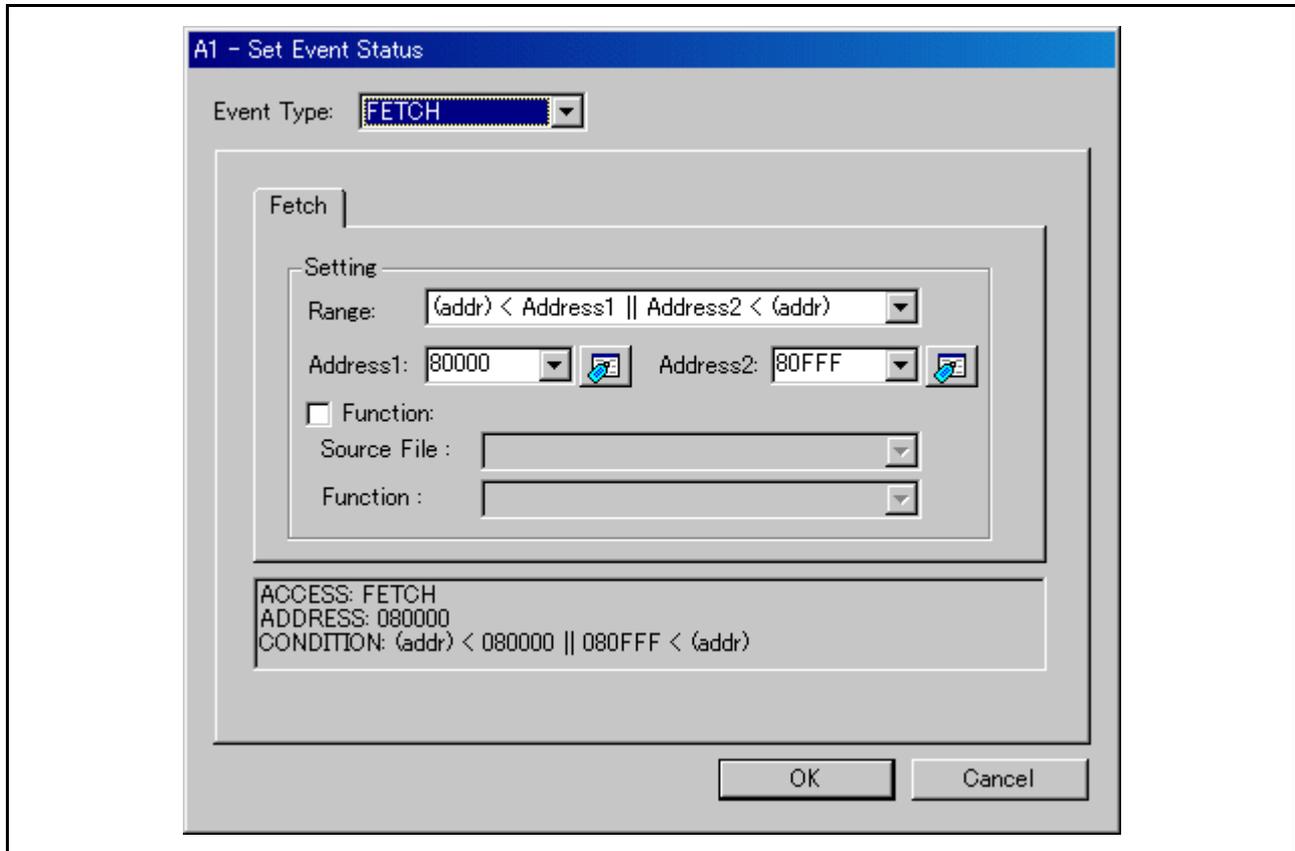
示例) 从地址 80000h 至 80FFFh 取指令



7.7.5.3 从指定地址范围以外取指令

设置如下所示。

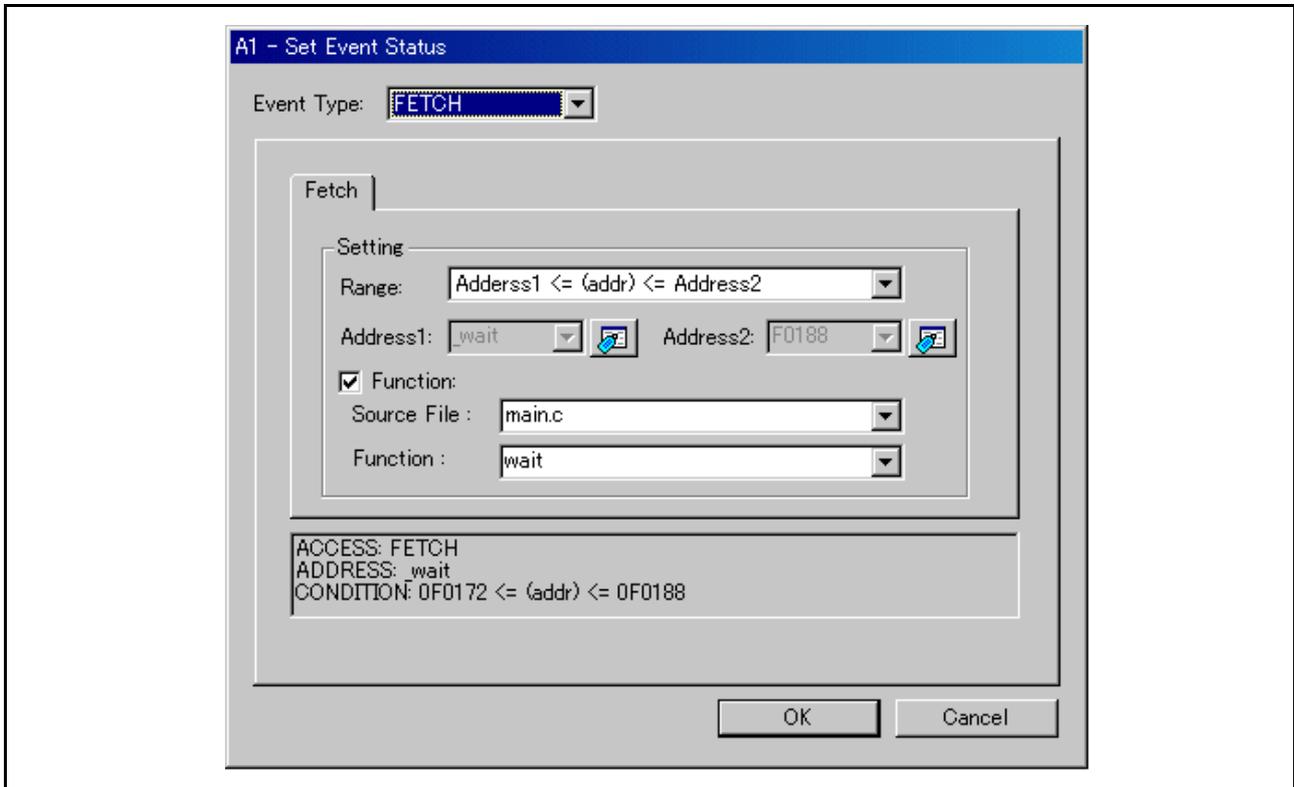
示例) 从 80000h 至 80FFFh 范围以外的任意地址取指令



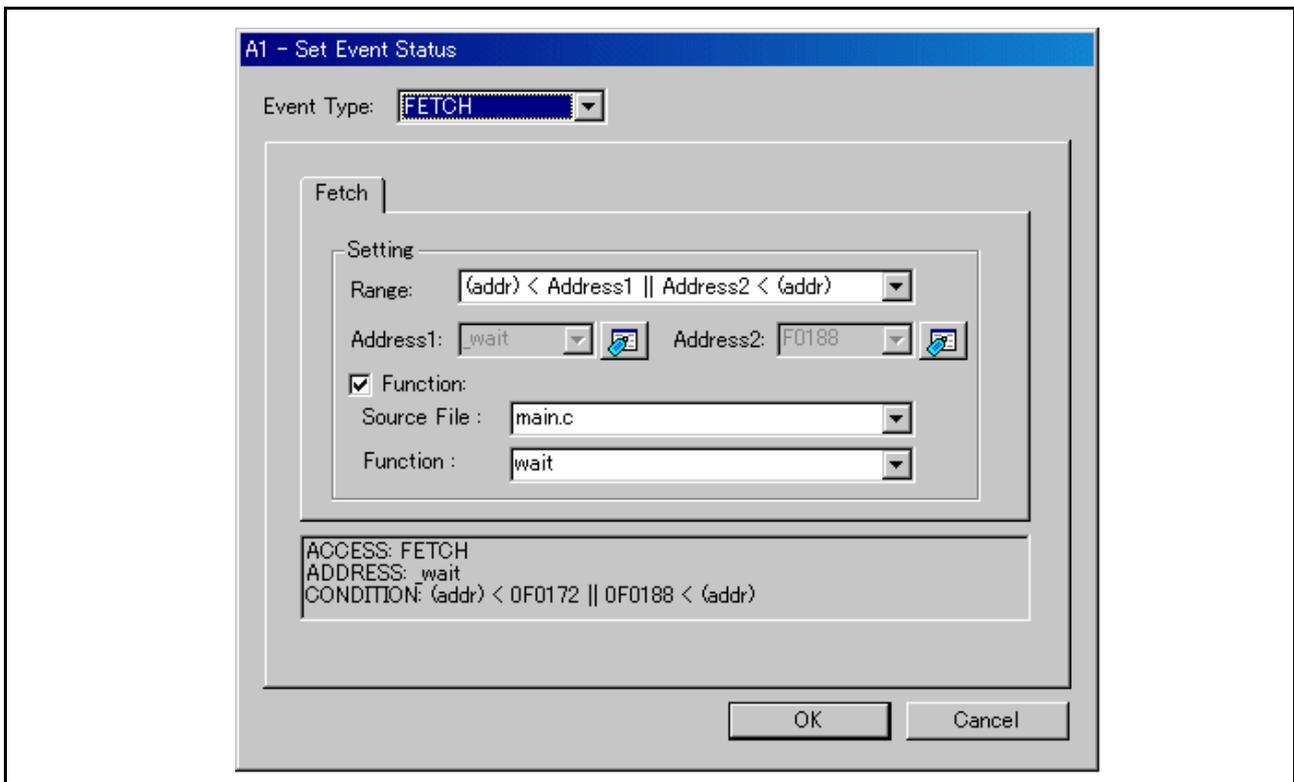
7.7.5.4 进入 / 退出指定函数

设置如下所示。

示例) 进入名为 “wait” 的函数



示例) 从名为 “wait” 的函数退出



7.7.6 指定事件（存储器存取）

要指定存储器存取事件，请将事件选择对话框的 [Event Type]（事件类型）更改为“DATA ACCESS”。在指定地址处存取存储器时，或在为指定地址范围设置的条件下存取存储器时建立事件。

7.7.6.1 存储器存取（用于 M32C 的调试器）

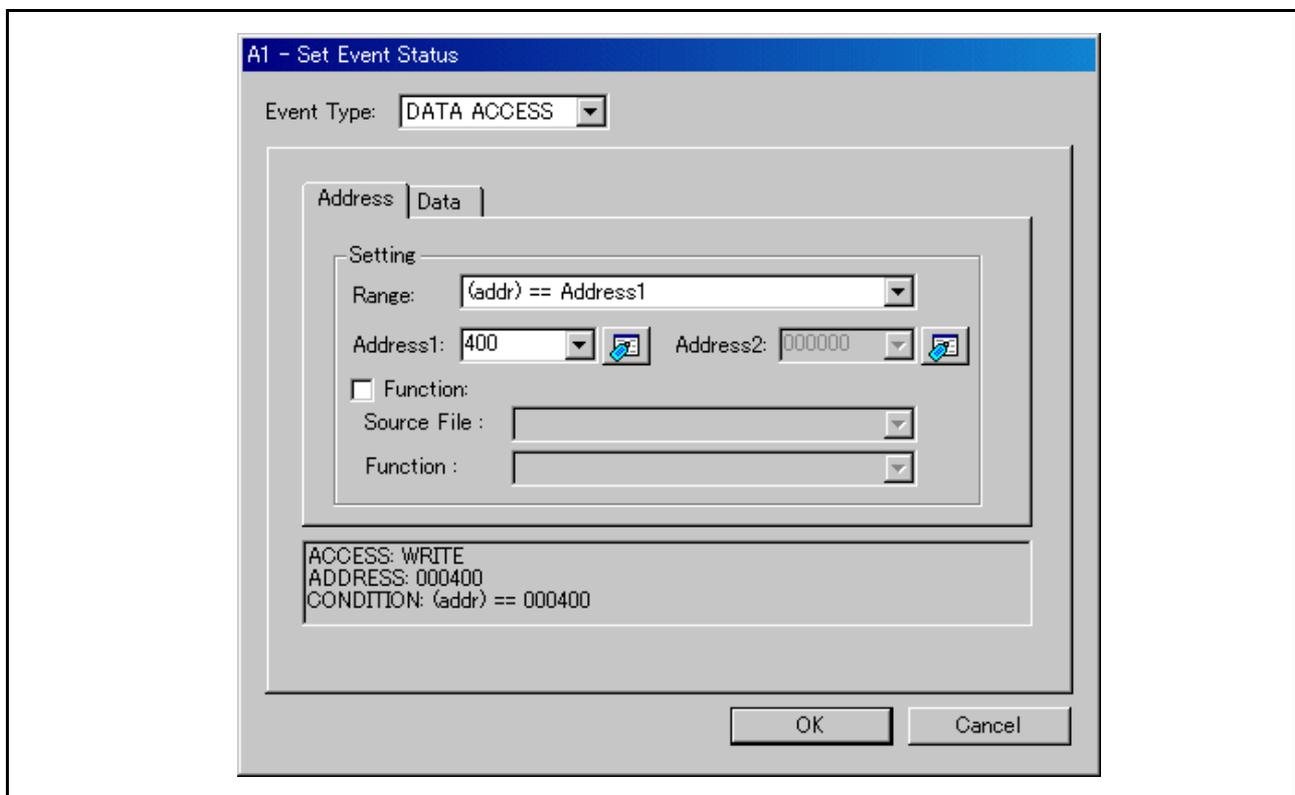
注意事项

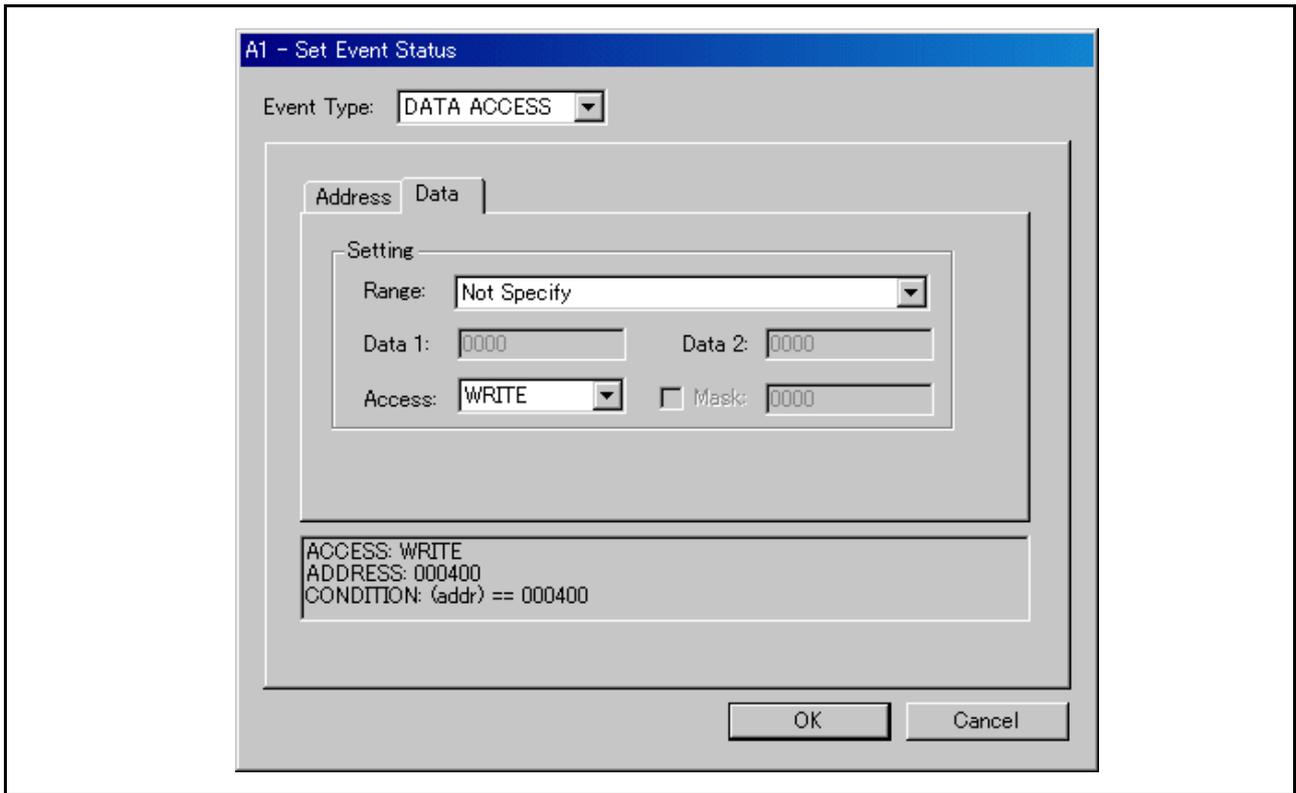
从奇地址以单字长大小存取数据时不会被检测到。

7.7.6.1.1 写入 / 读取指定地址

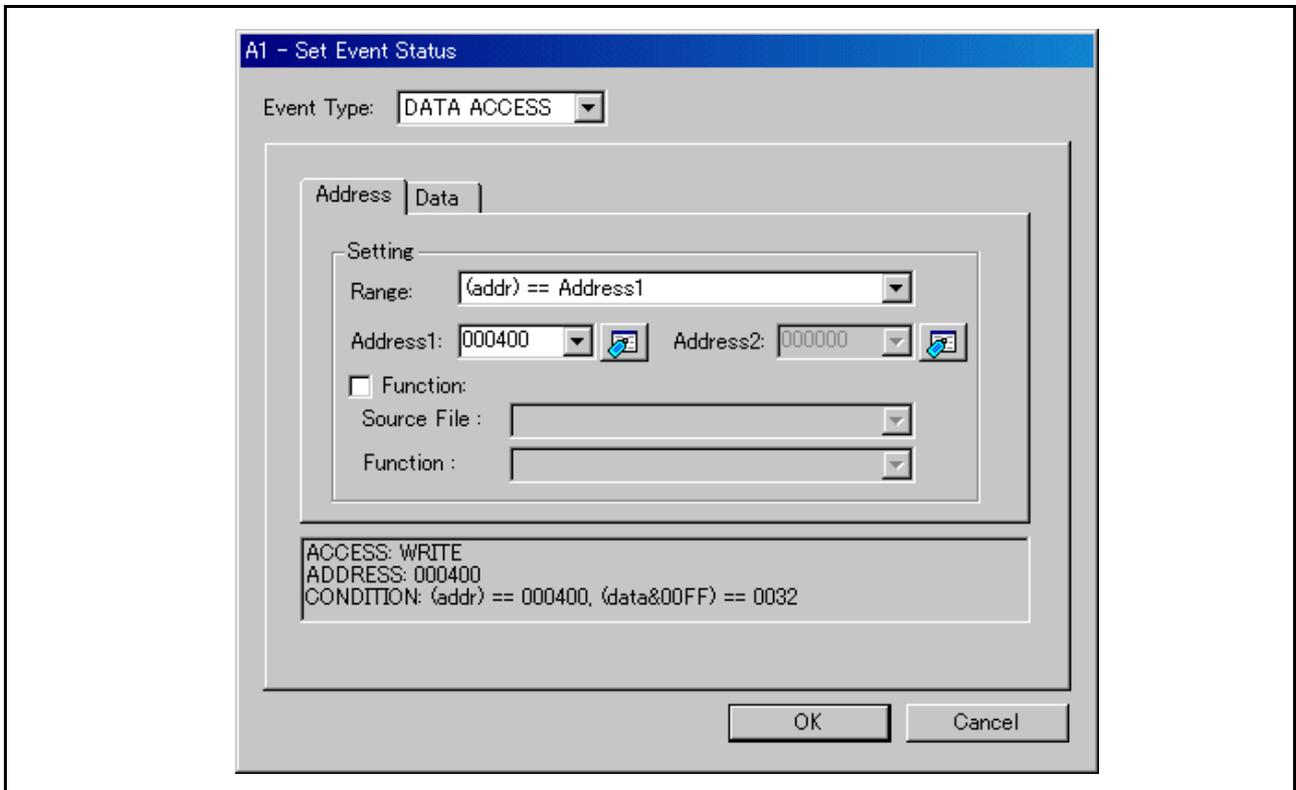
设置如下所示。

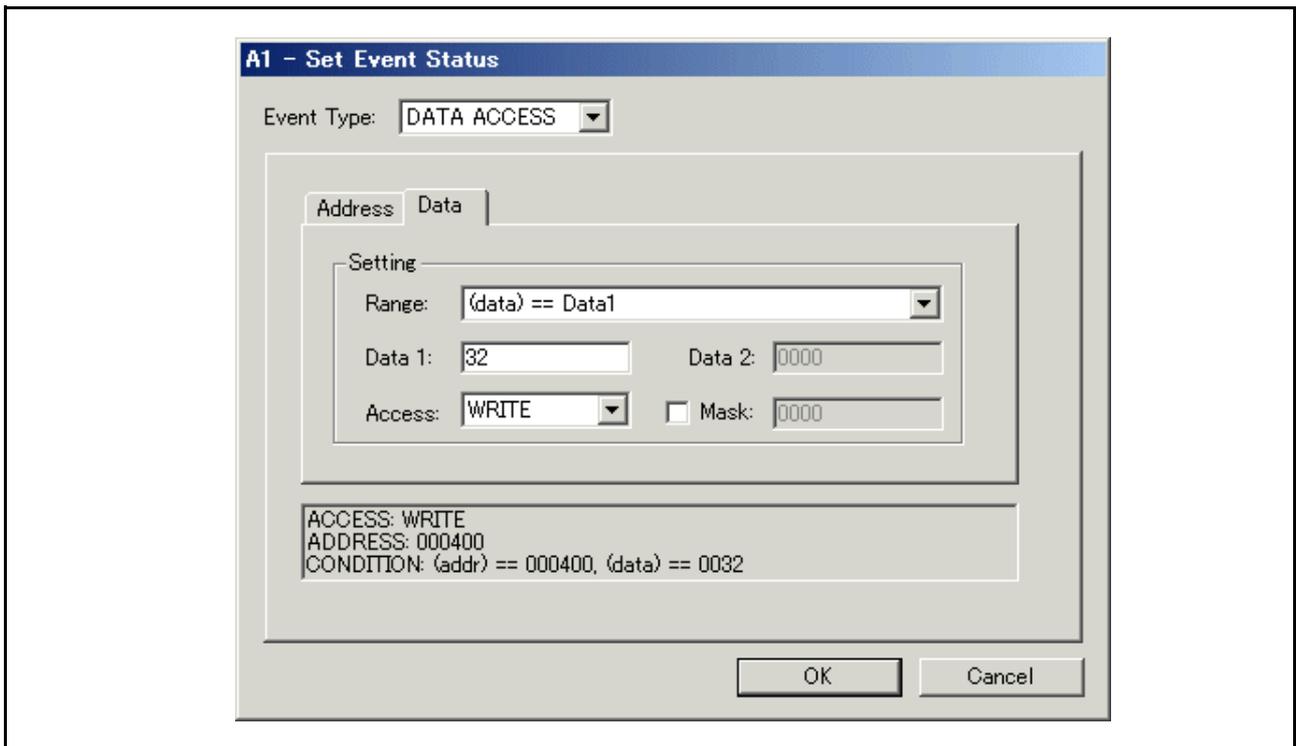
示例) 写入偶地址 400h





示例) 将字节长度的数据 32h 写入偶地址 400h

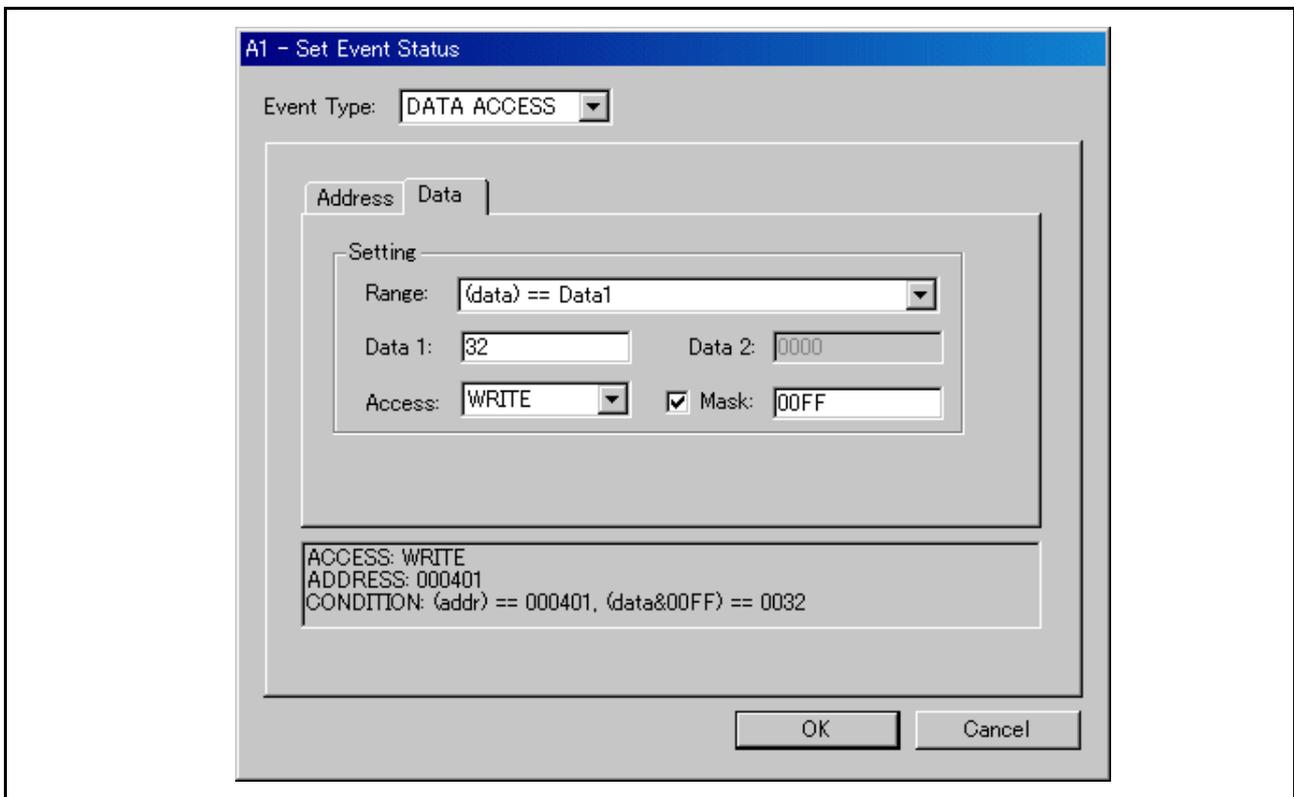
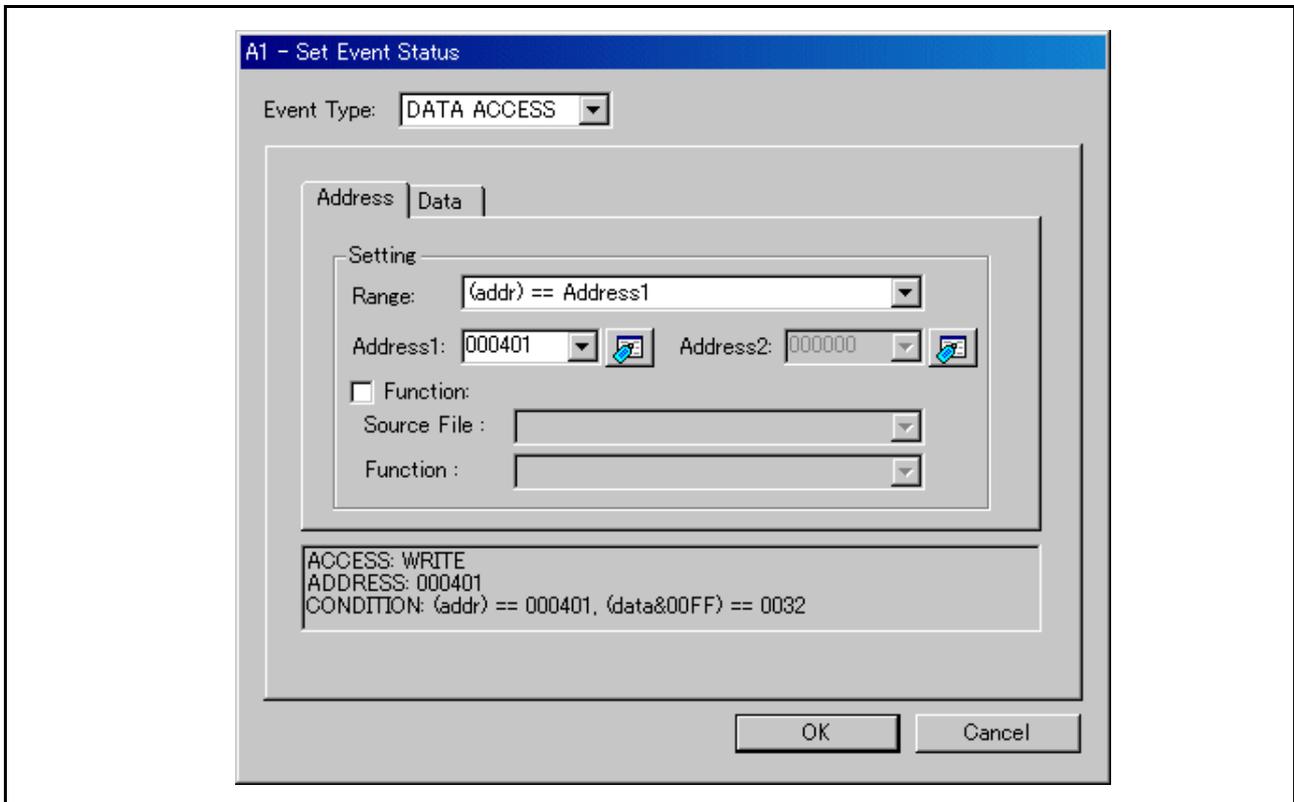




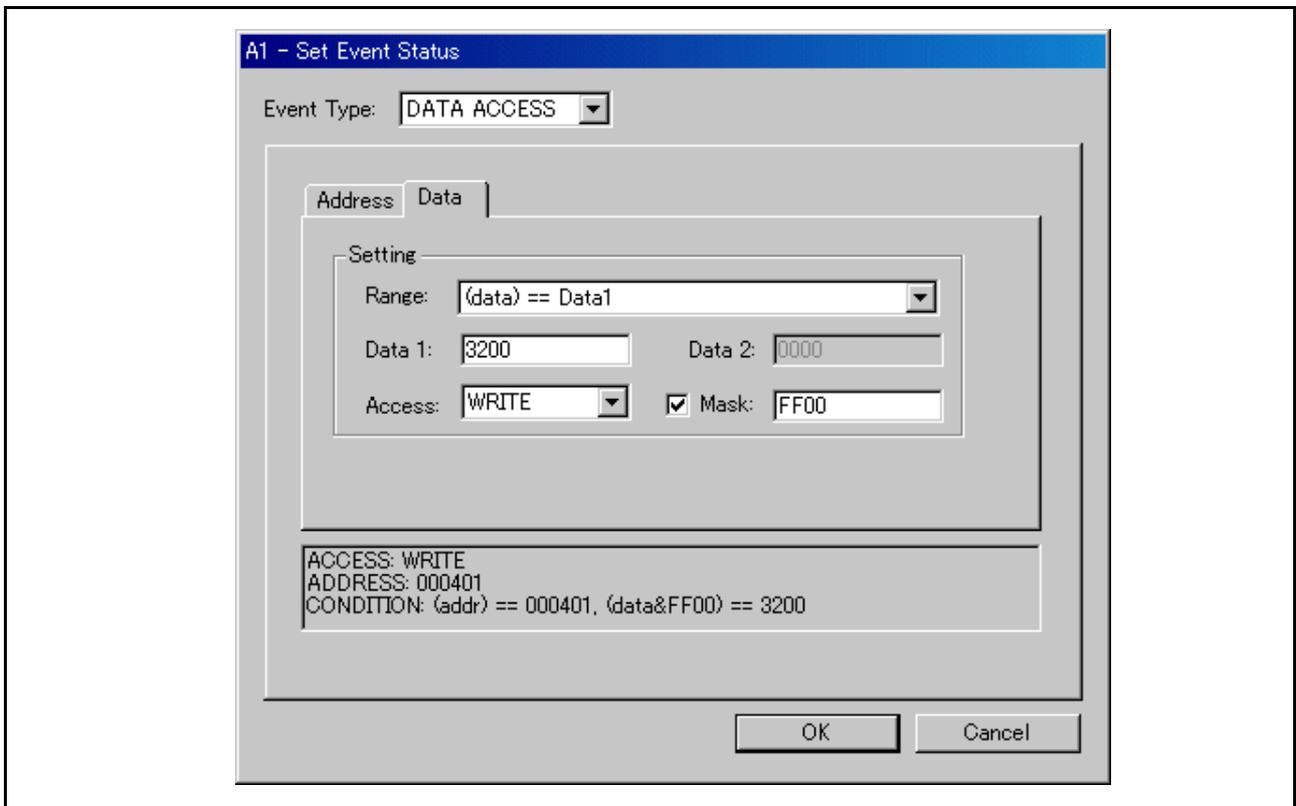
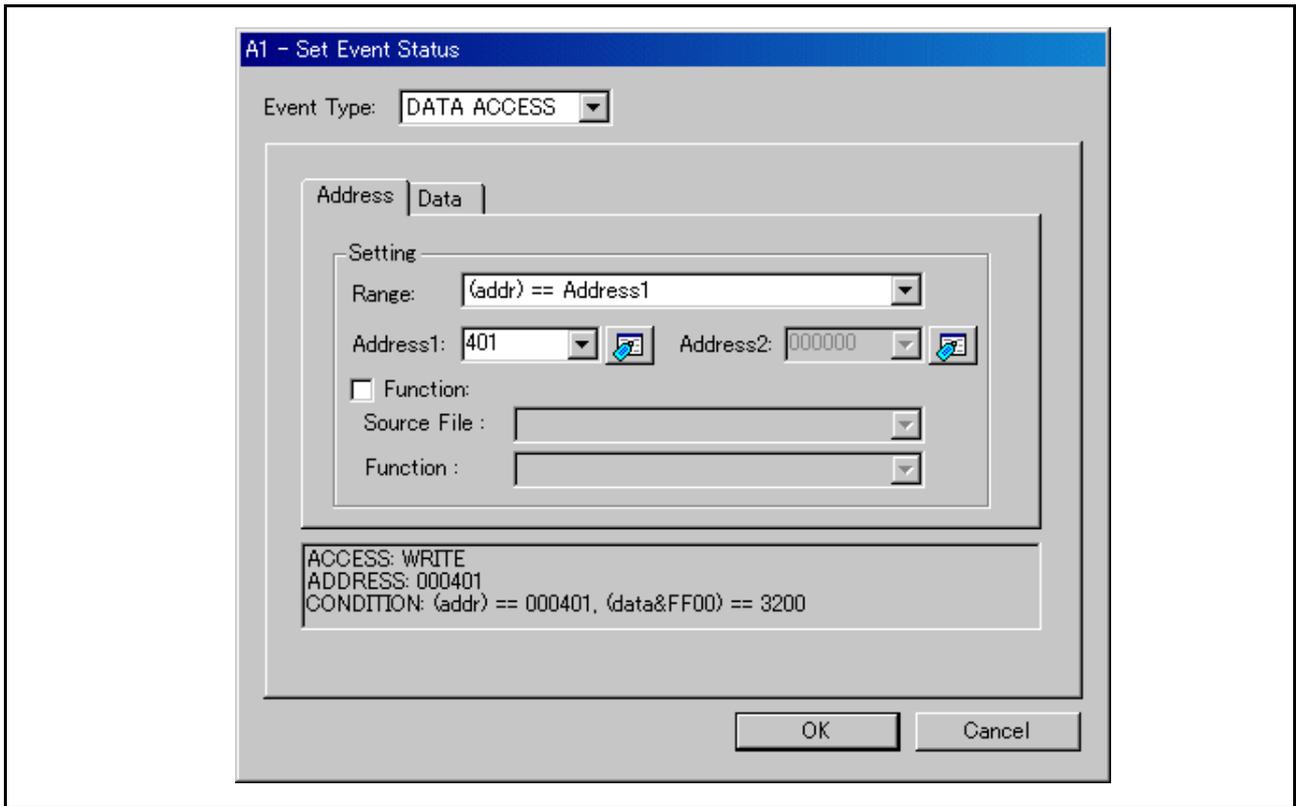
示例) 将字节长度的数据 32h 写入奇地址 401h

设置内容因产品和总线宽度而异。

(8 位总线宽度)



(16 位总线宽度)

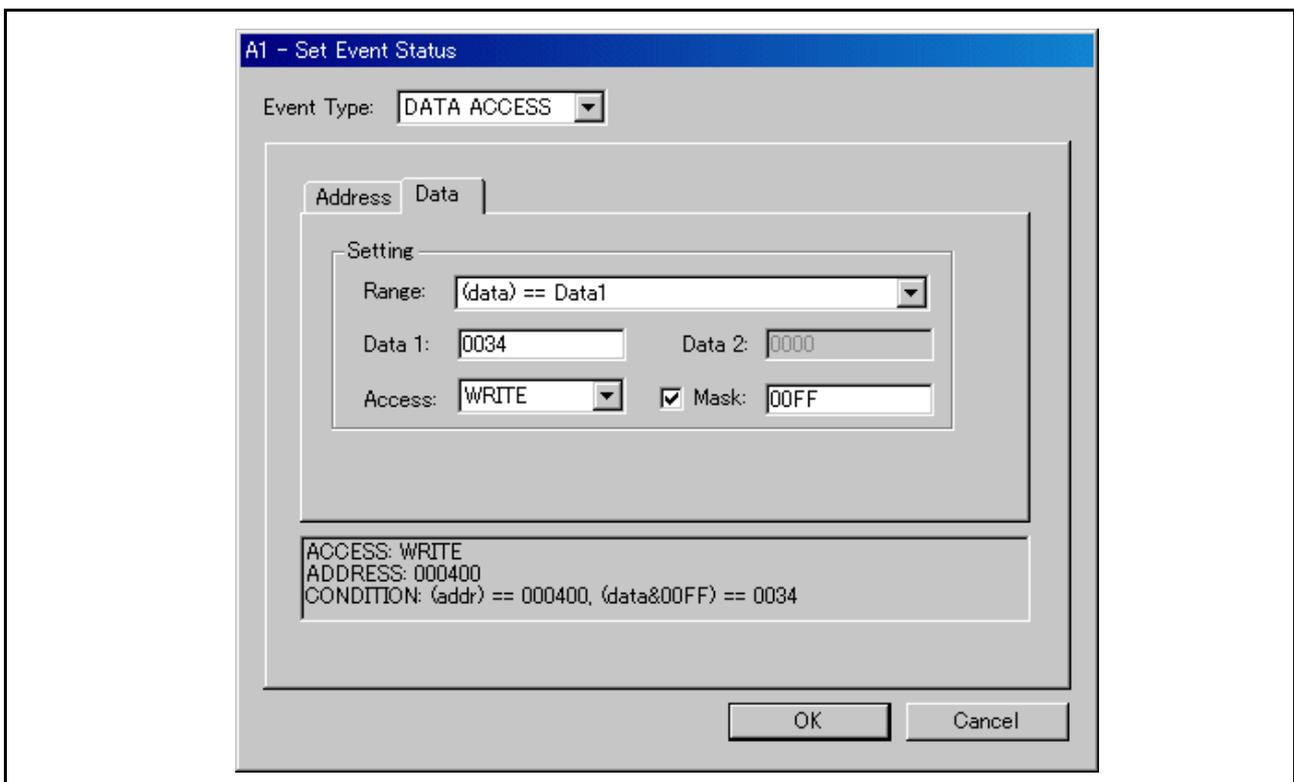
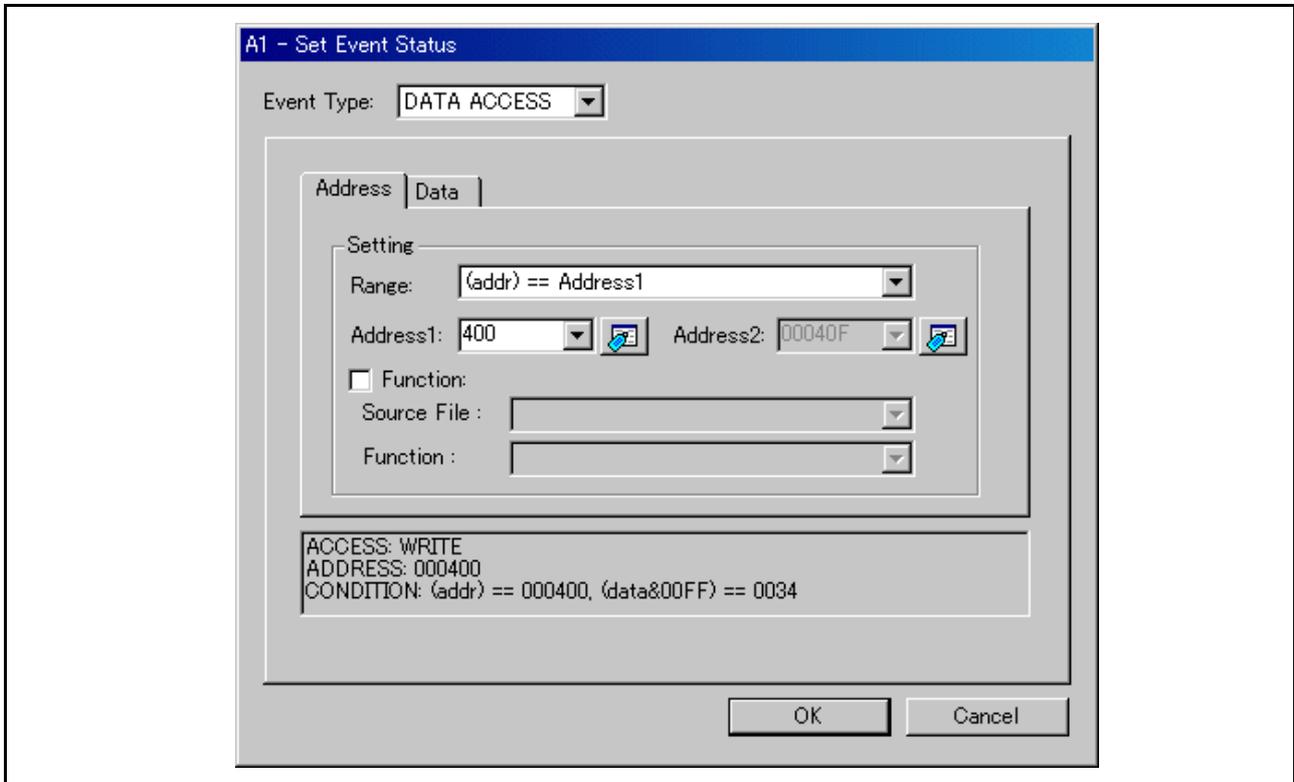


(示例) 将字长度的数据 1234h 写入偶地址 400h
设置内容因产品和总线宽度而异。

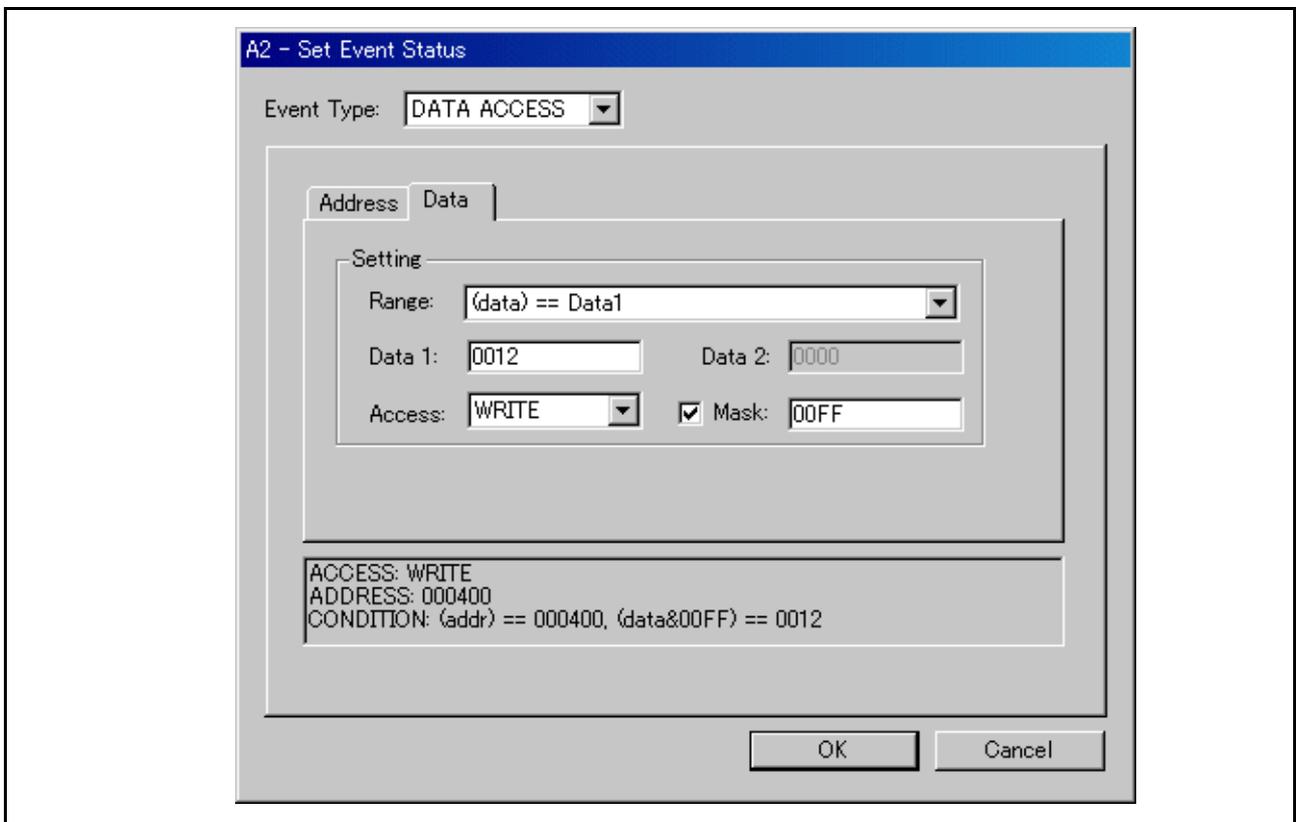
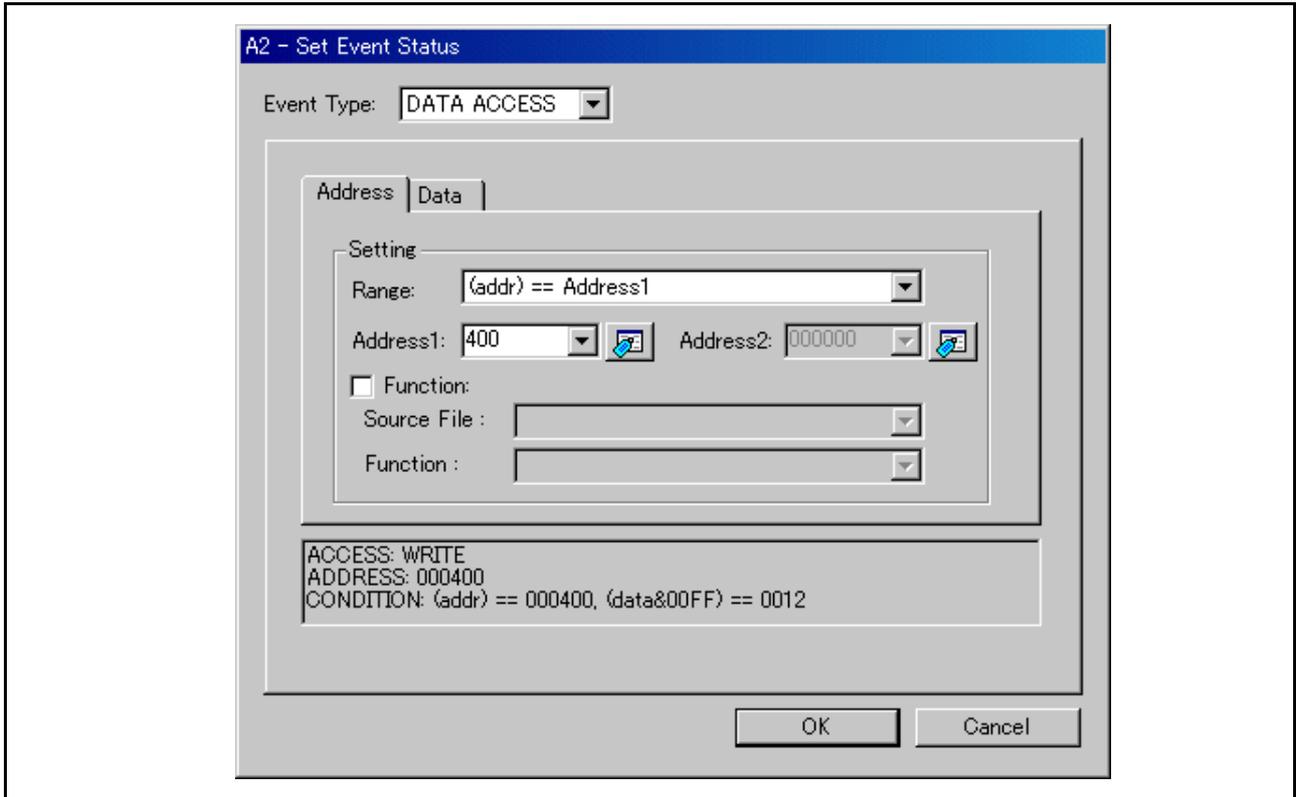
(8 位总线宽度)

请组合指定第一部分和第二部分的逻辑与。

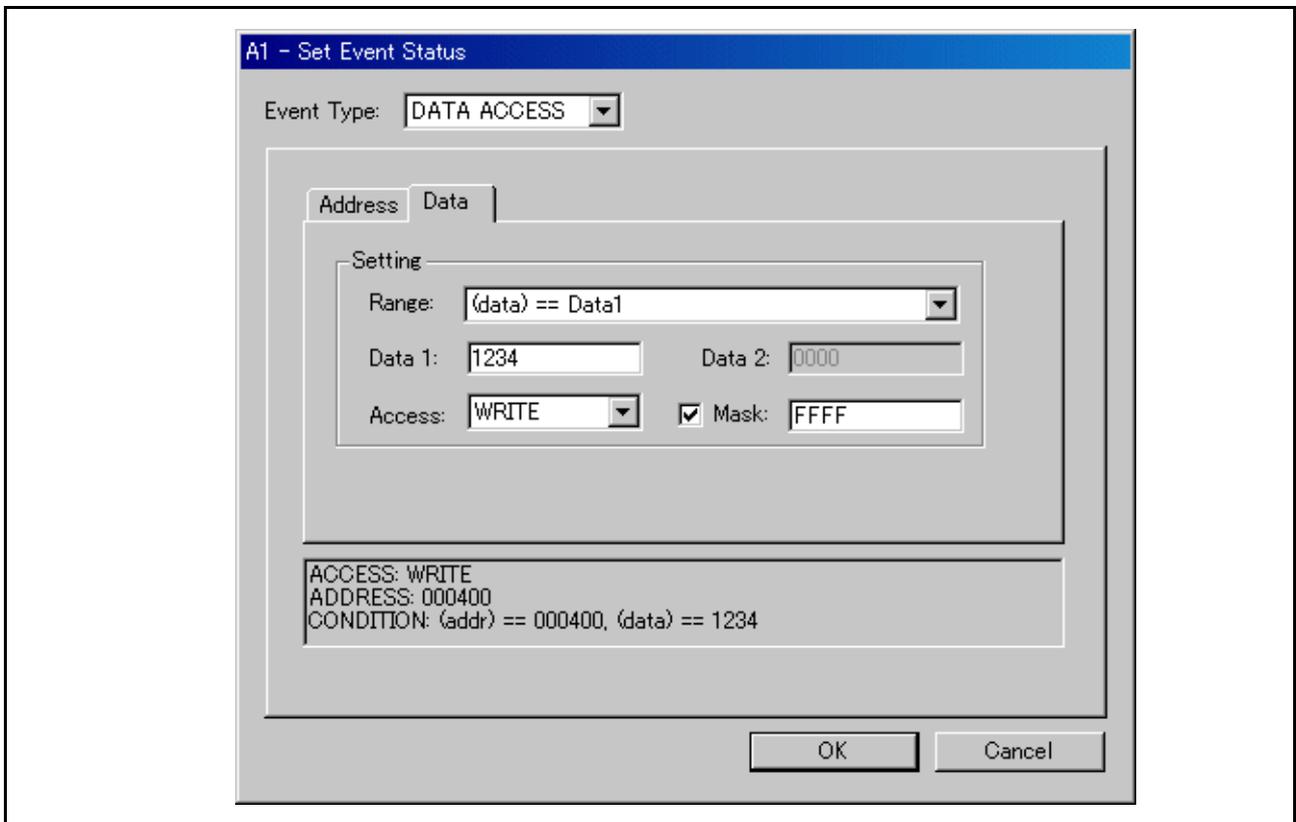
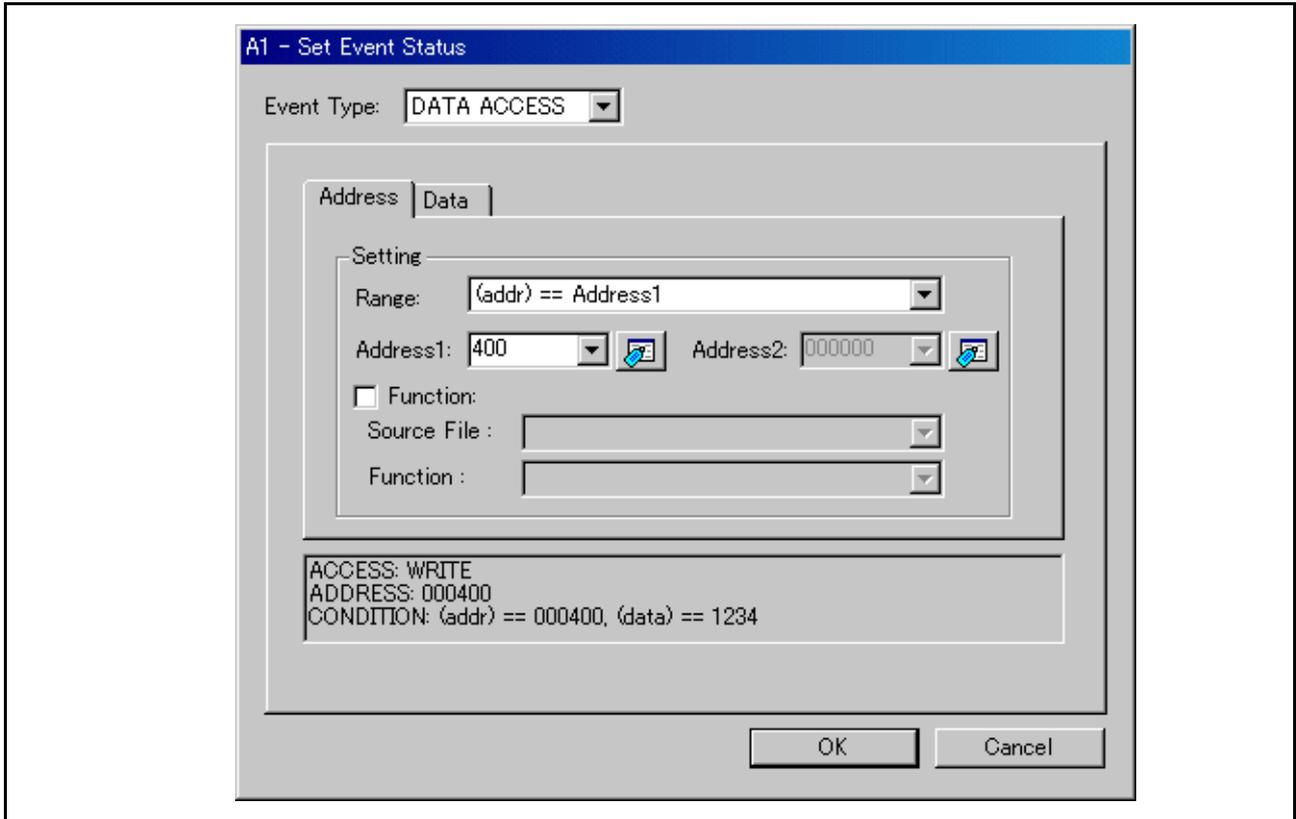
第一部分



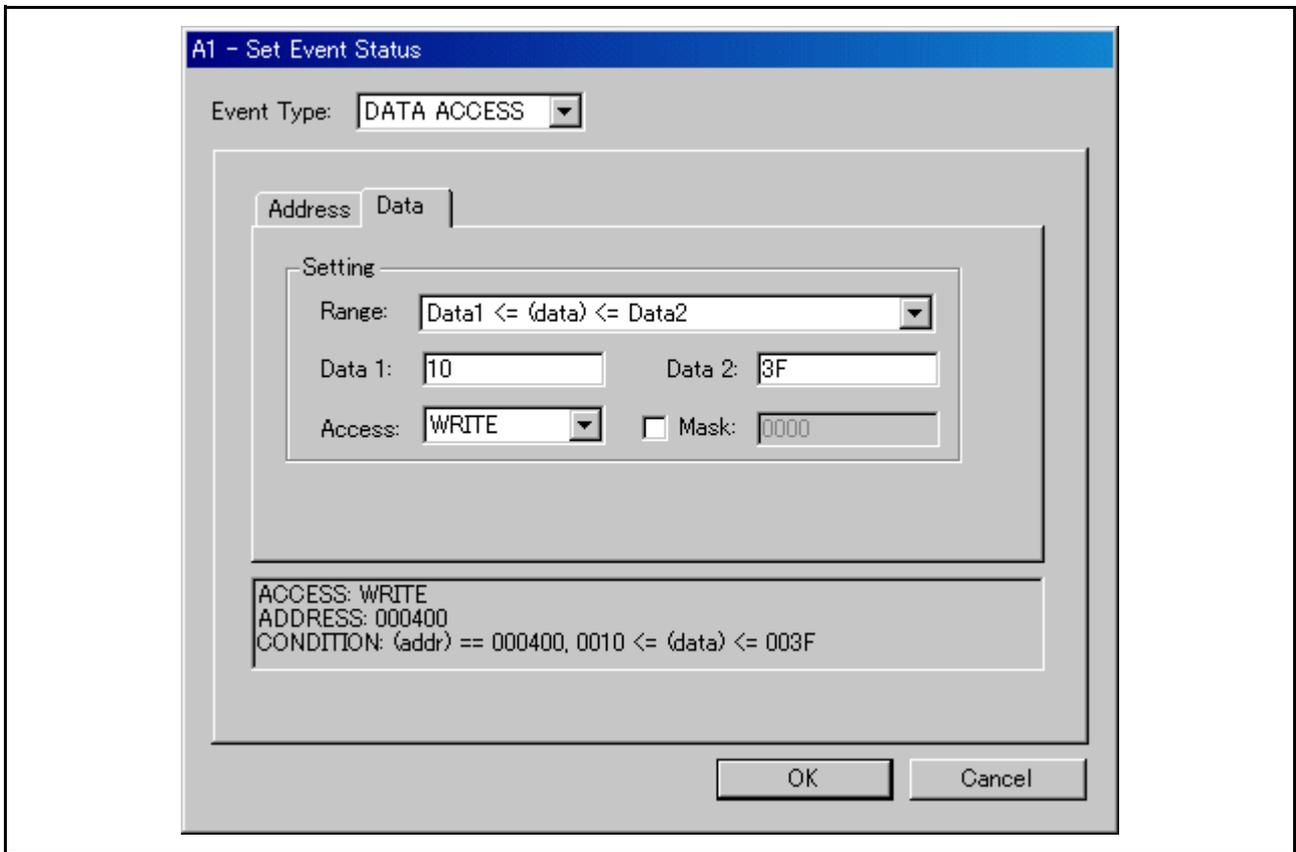
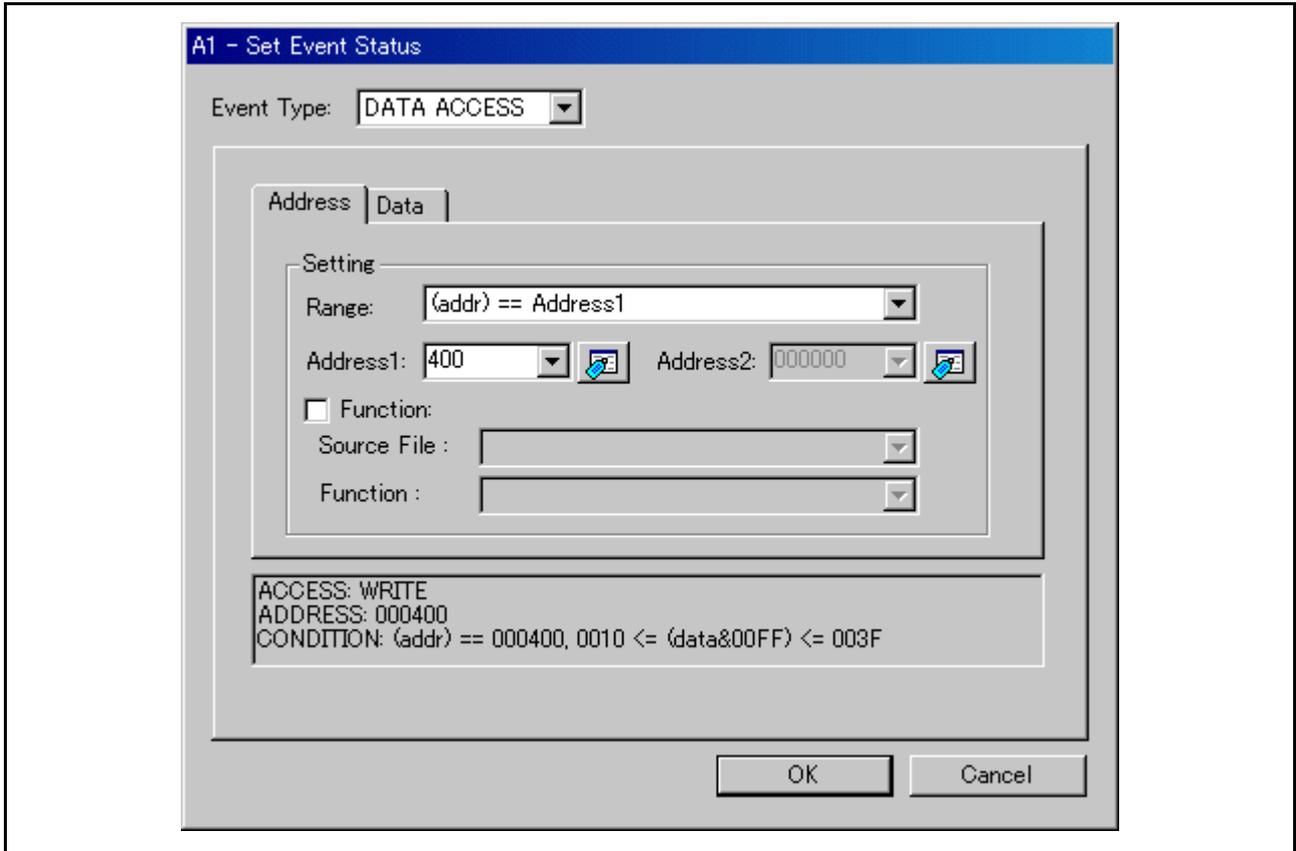
第二部分



(16 位总线宽度)



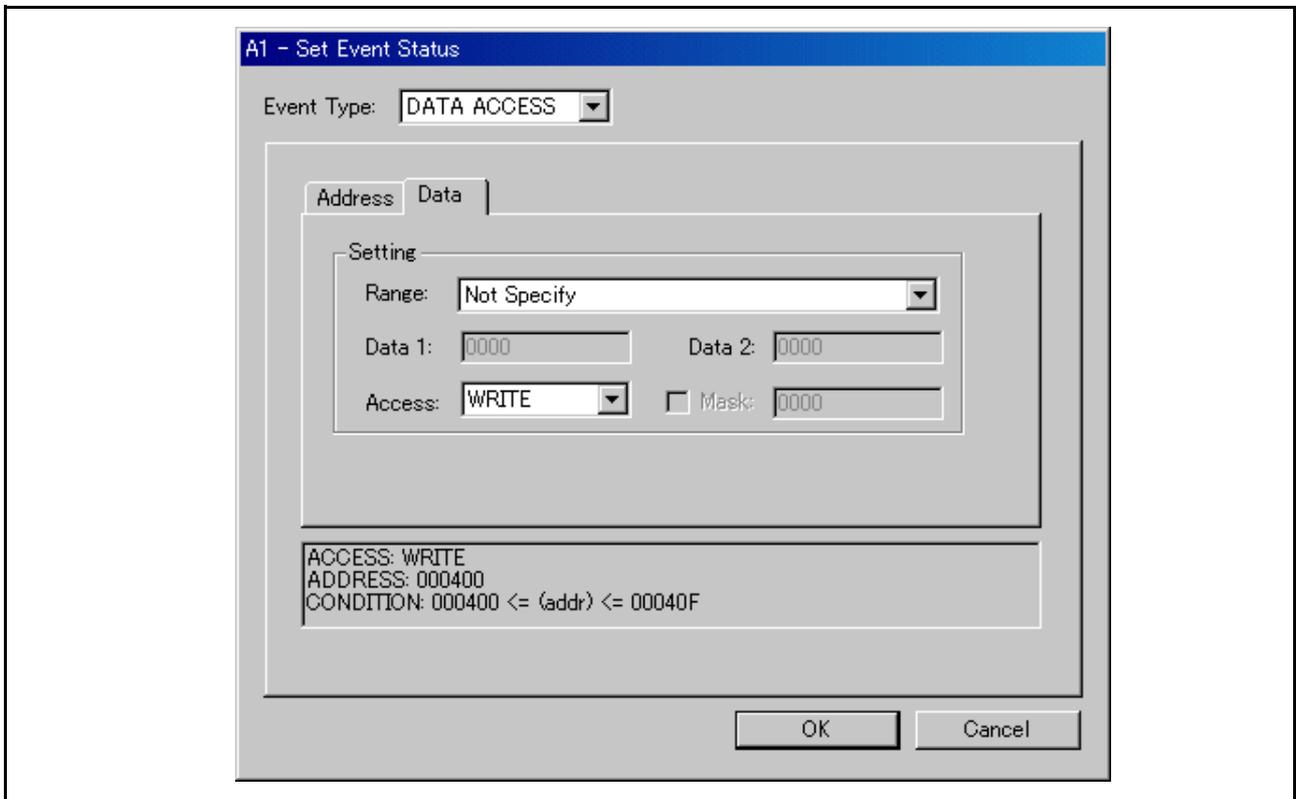
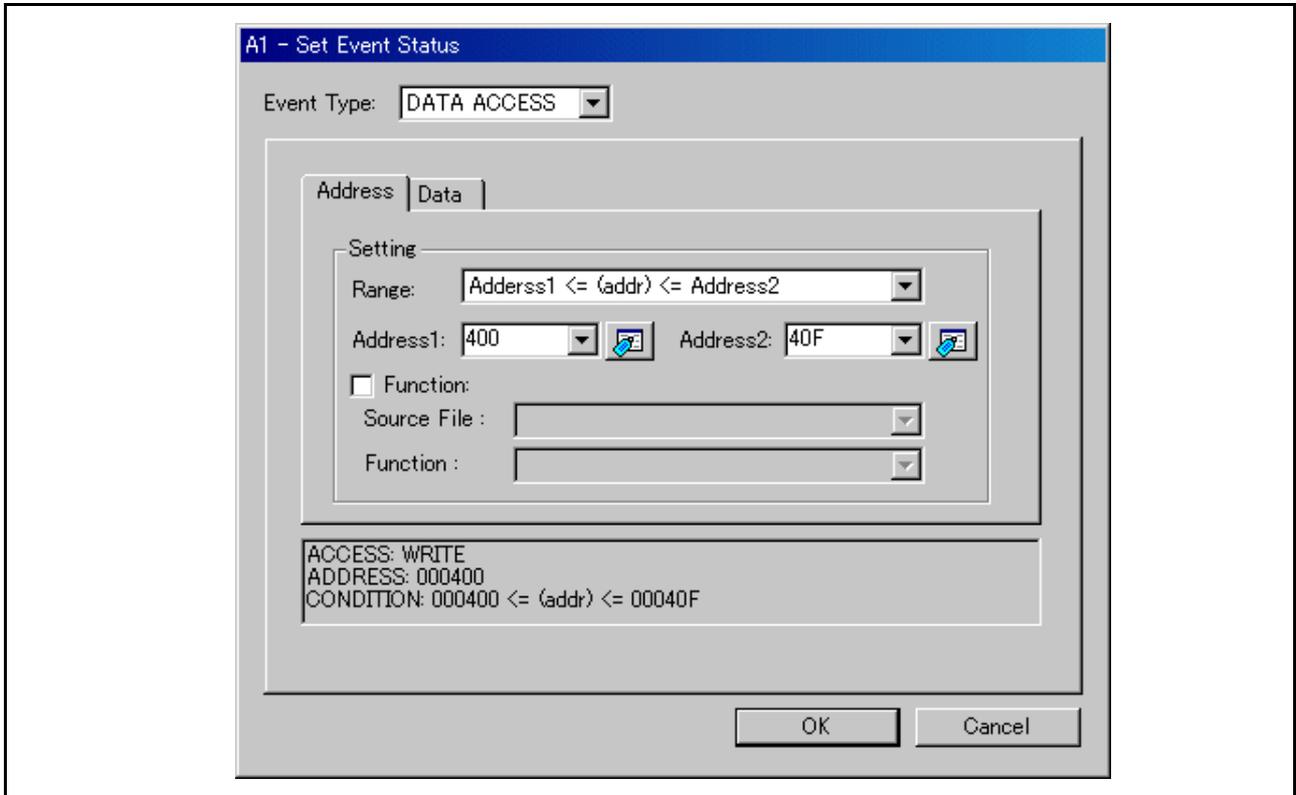
示例) 将数据 10h - 3Fh 写入偶地址 400h



7.7.6.1.2 从指定地址范围读取 / 写入数据

设置如下所示。

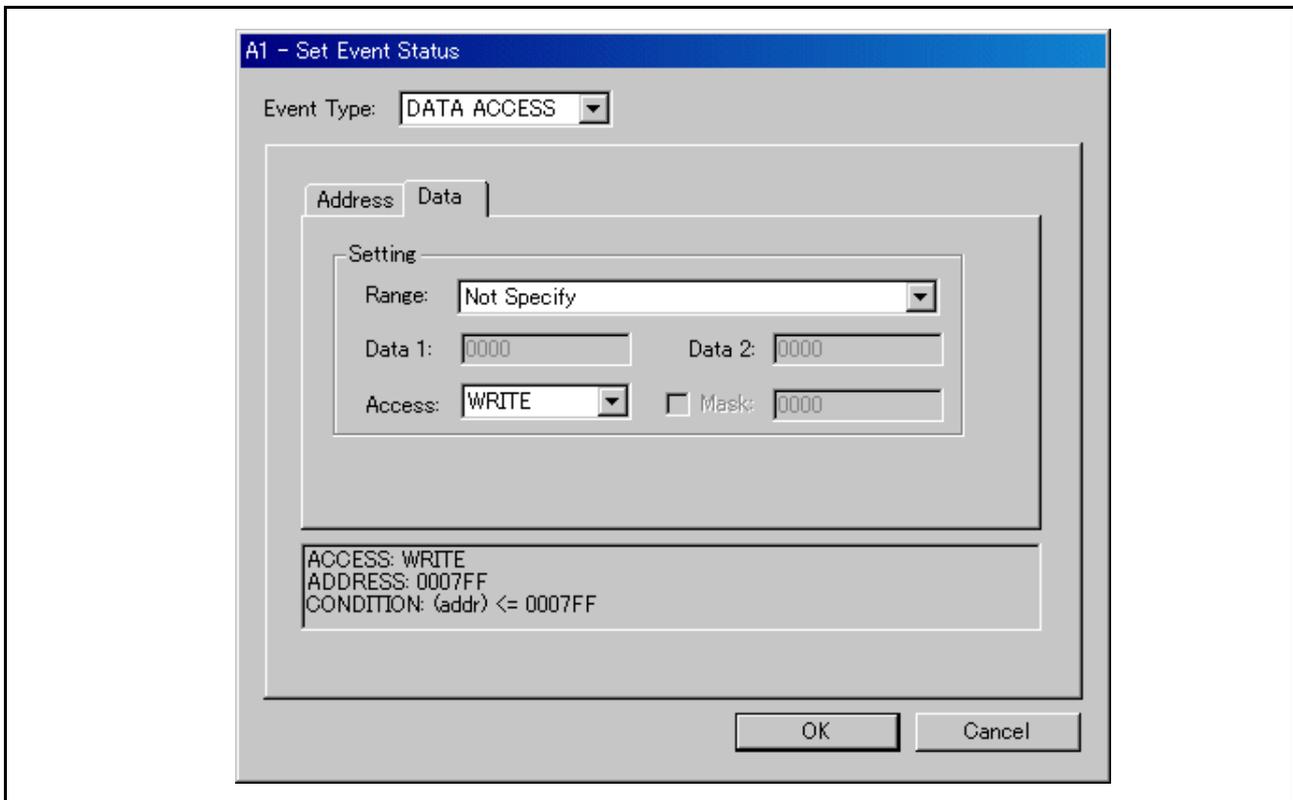
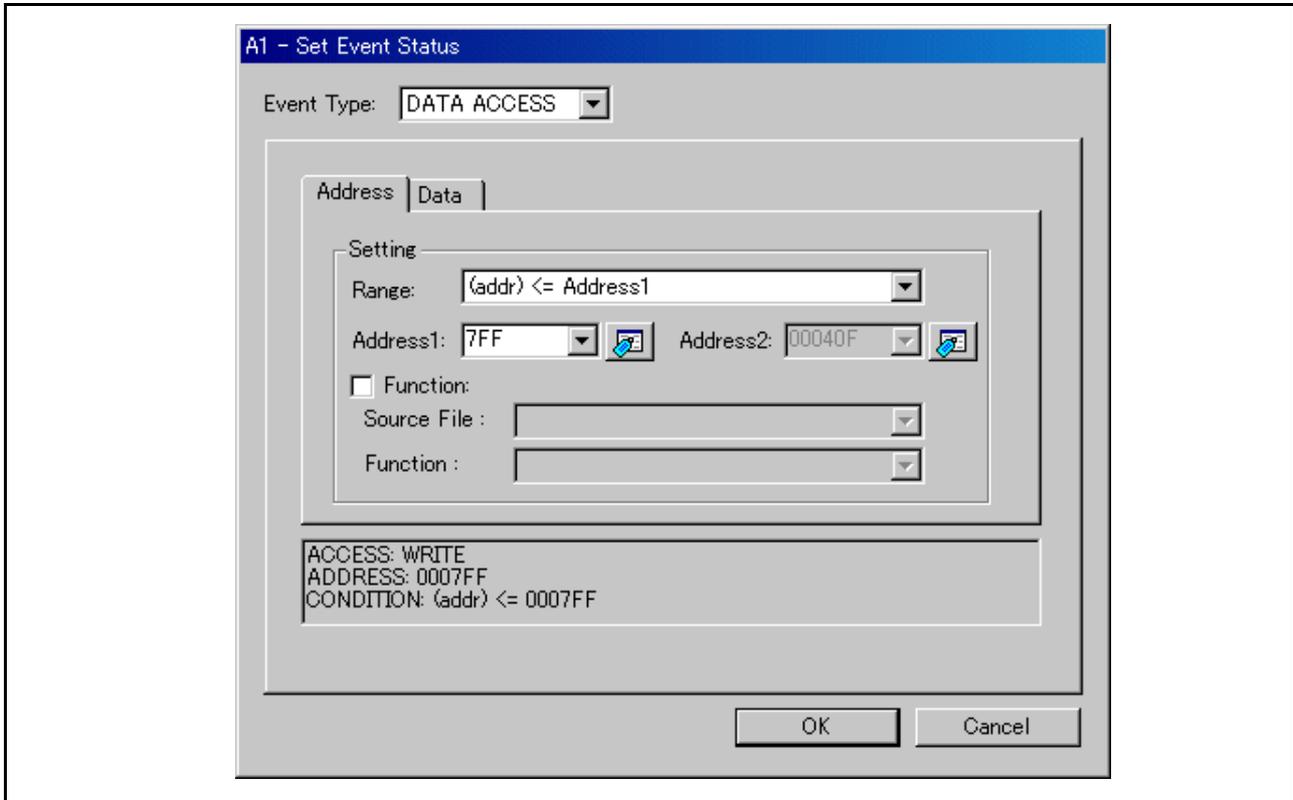
示例) 将数据写入 400h 至 40Fh 的地址范围



7.7.6.1.3 从指定地址范围以外读取 / 写入数据

设置如下所示。

示例) 将数据写入小于 7FFh 的地址



7.7.6.2 存储器存取（用于 M16C/R8C 的调试器）

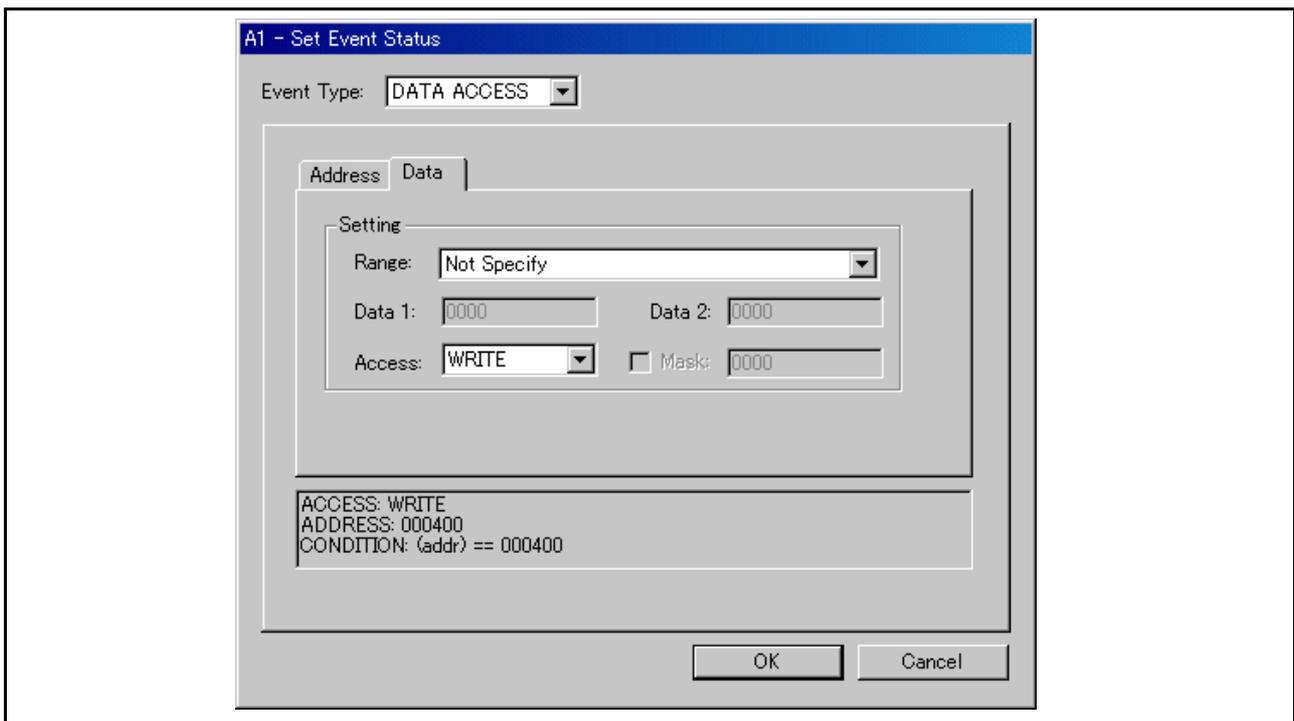
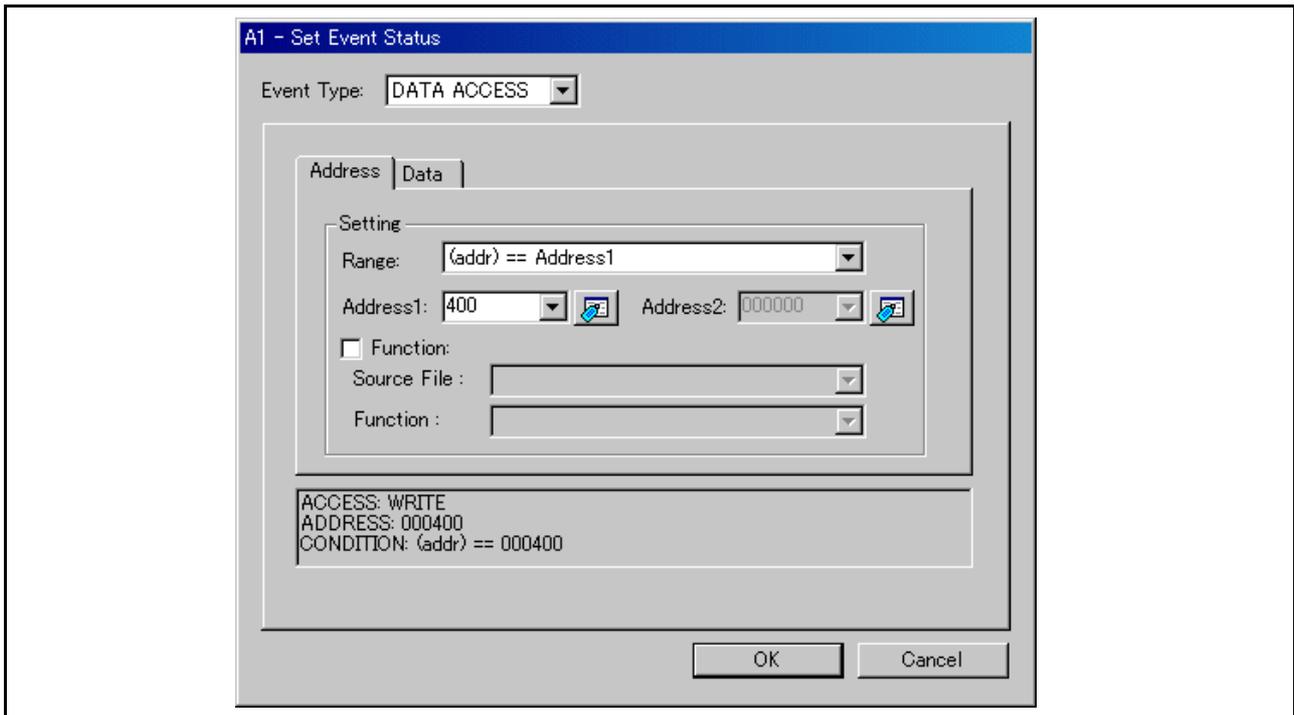
注意事项

从奇地址以单字长大小存取数据时不会被检测到。

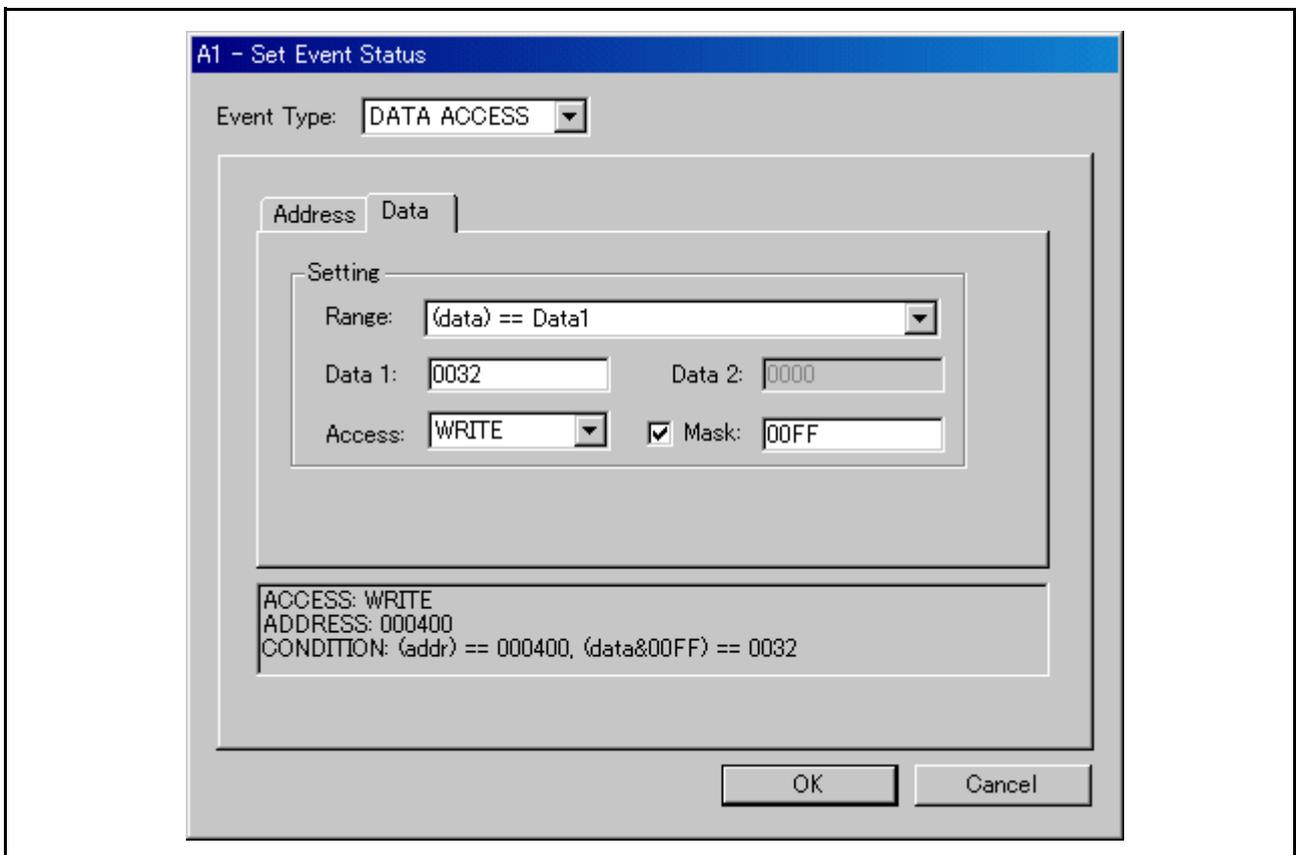
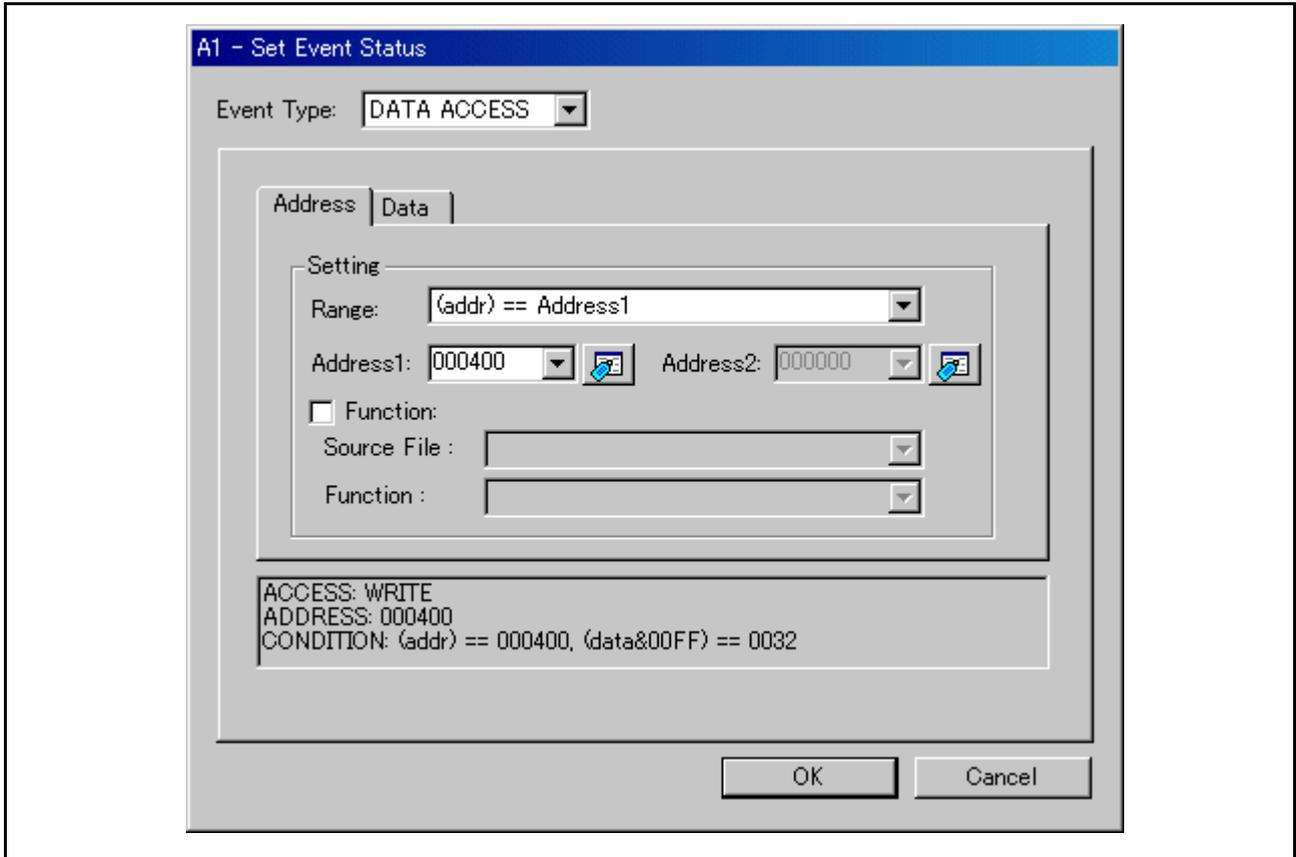
7.7.6.2.1 写入 / 读取指定地址

设置如下所示。

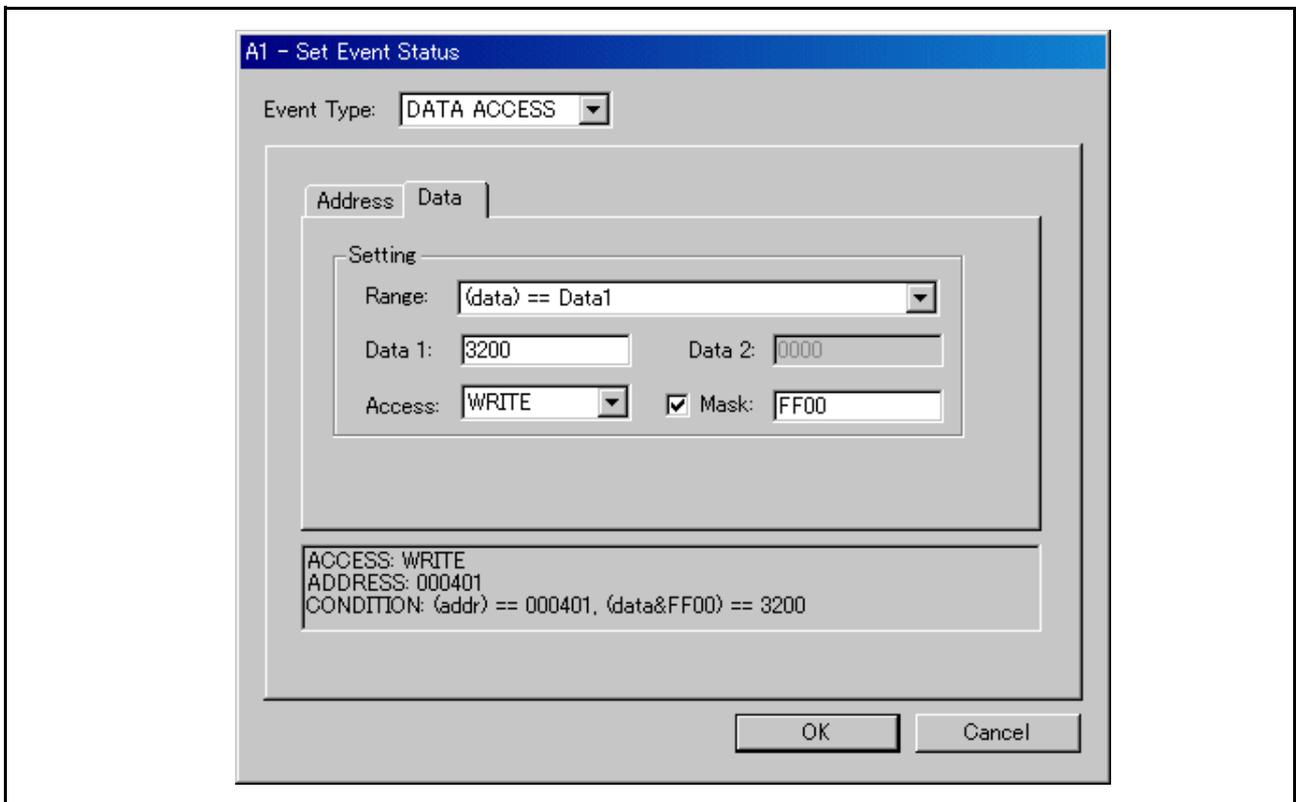
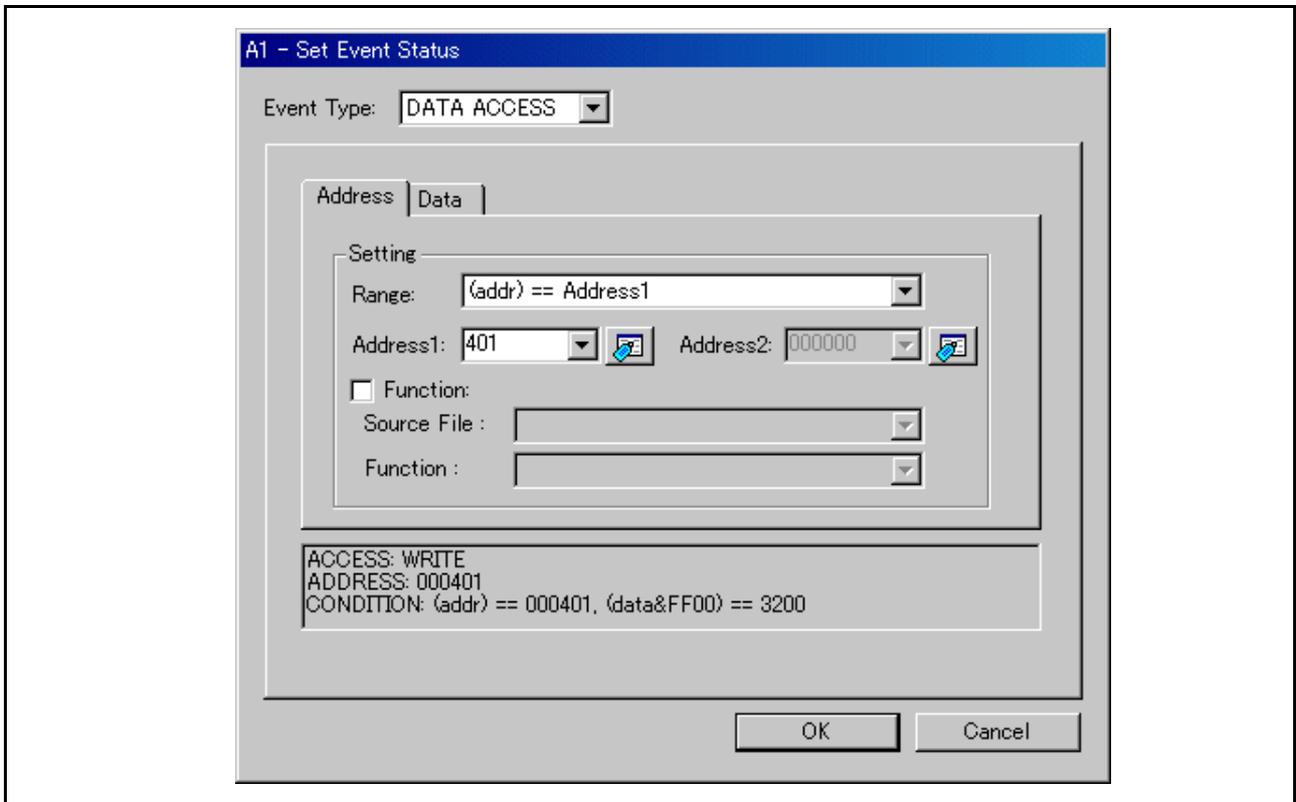
示例) 写入偶地址 400h



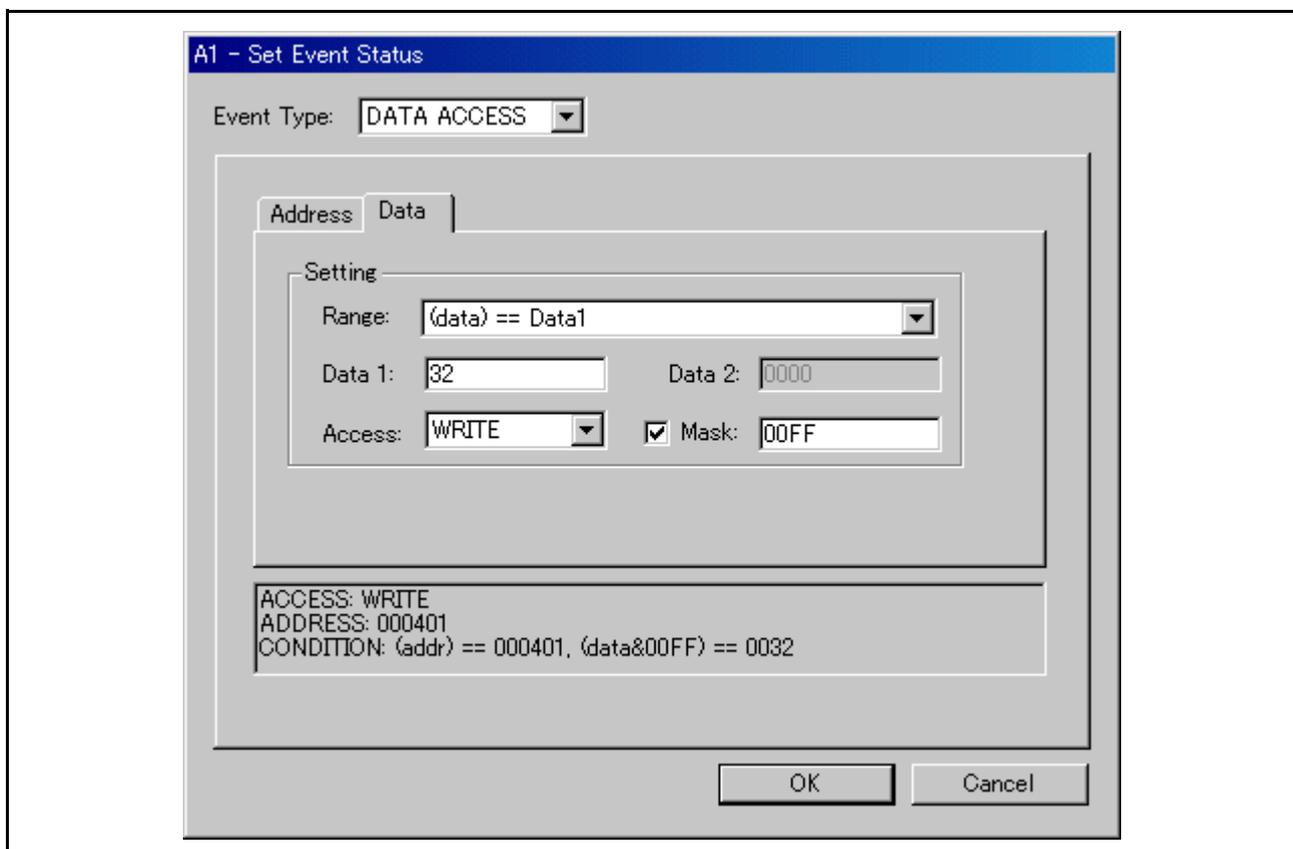
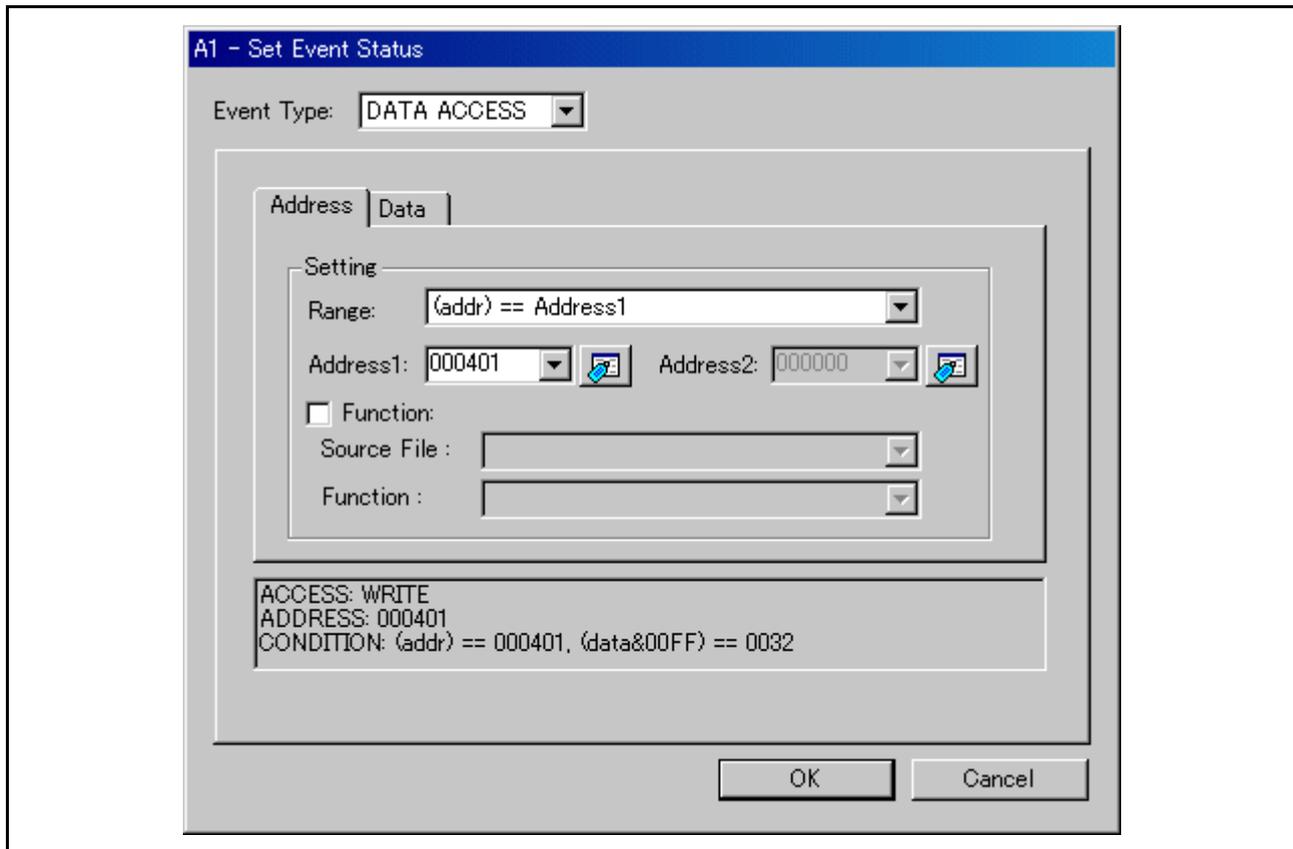
示例) 将字节长度的数据 32h 写入偶地址 400h



示例) 将字节长度的数据 32h 写入奇地址 401h
 设置内容因产品和总线宽度而异。
 (8 位总线宽度)



(16 位总线宽度)



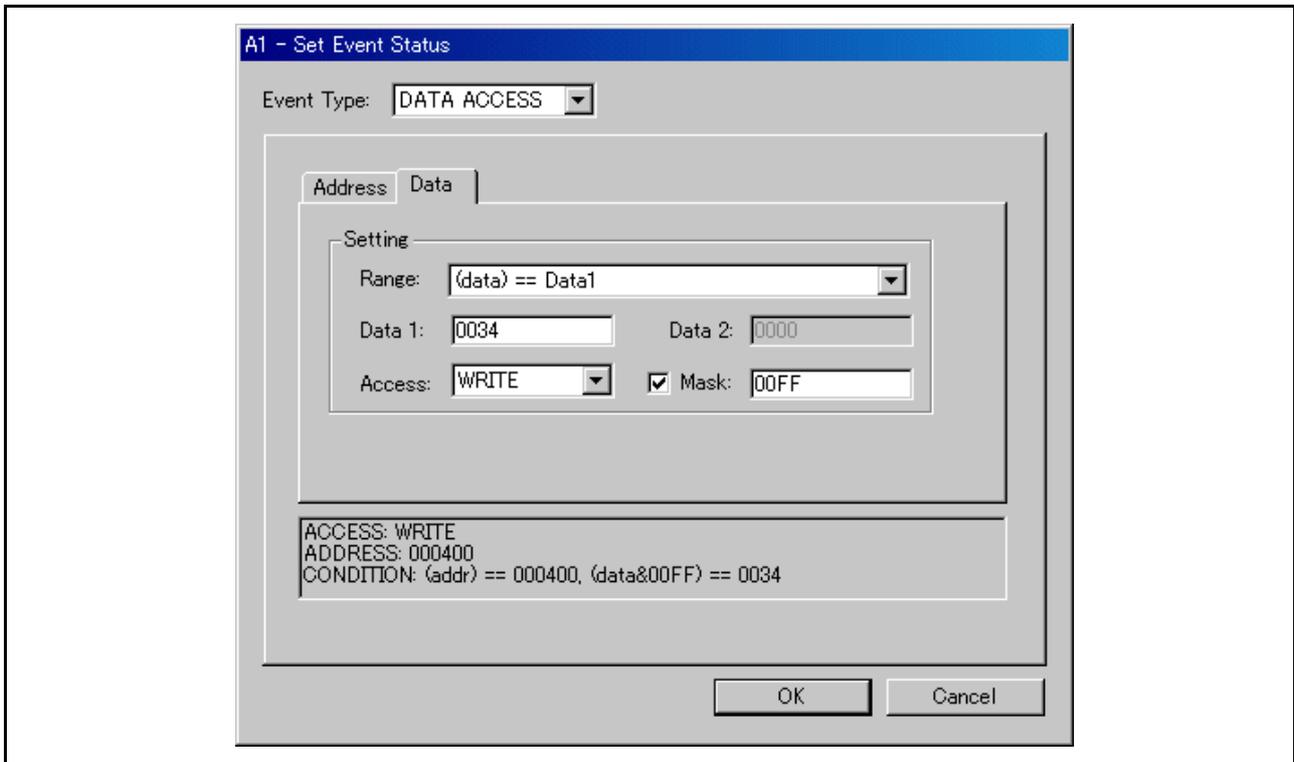
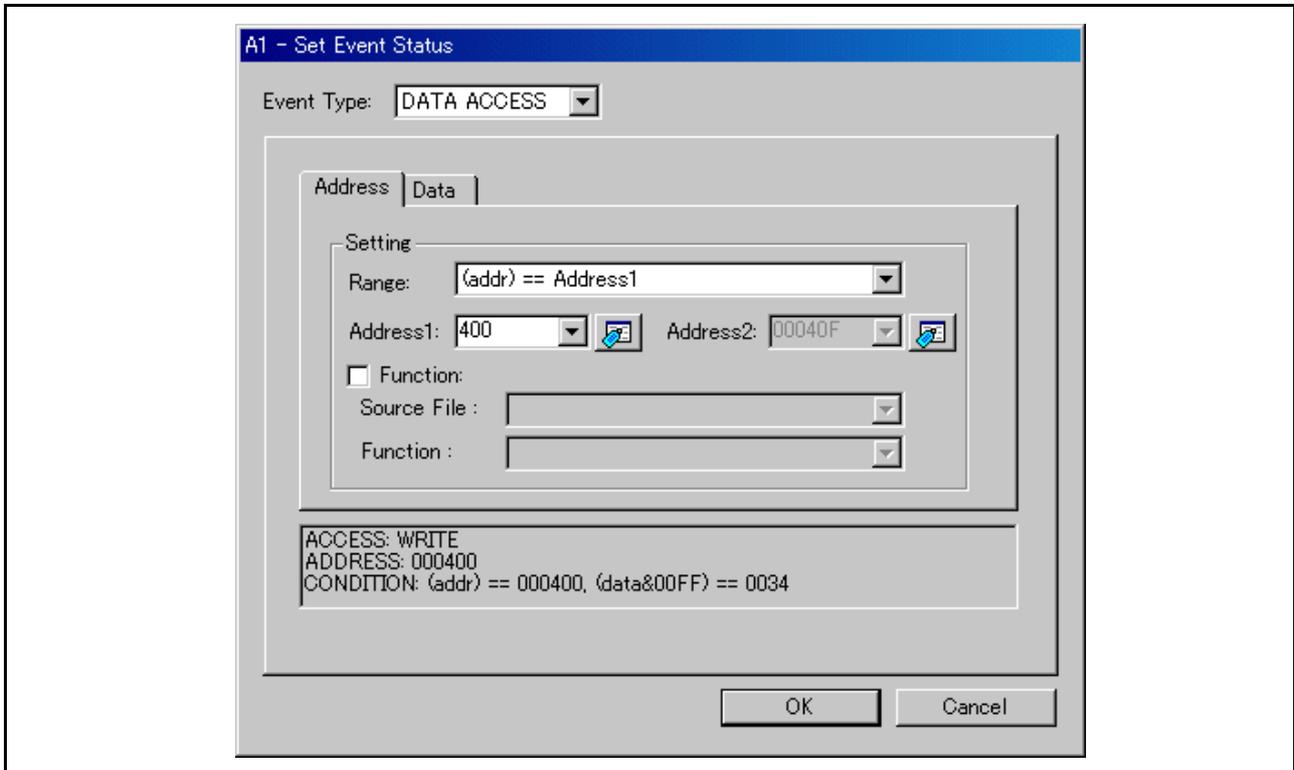
示例) 将字长度的数据 1234h 写入偶地址 400h

设置内容因产品和总线宽度而异。

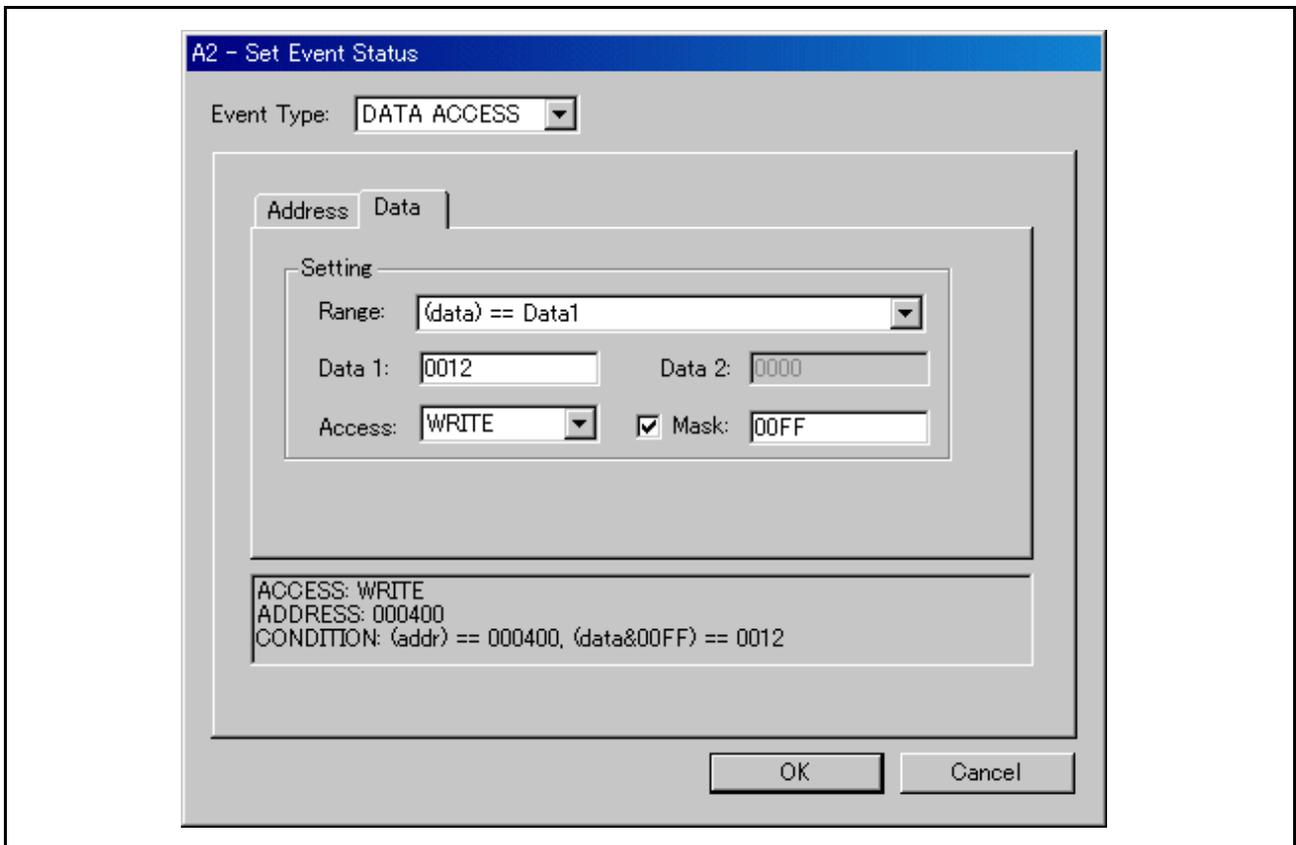
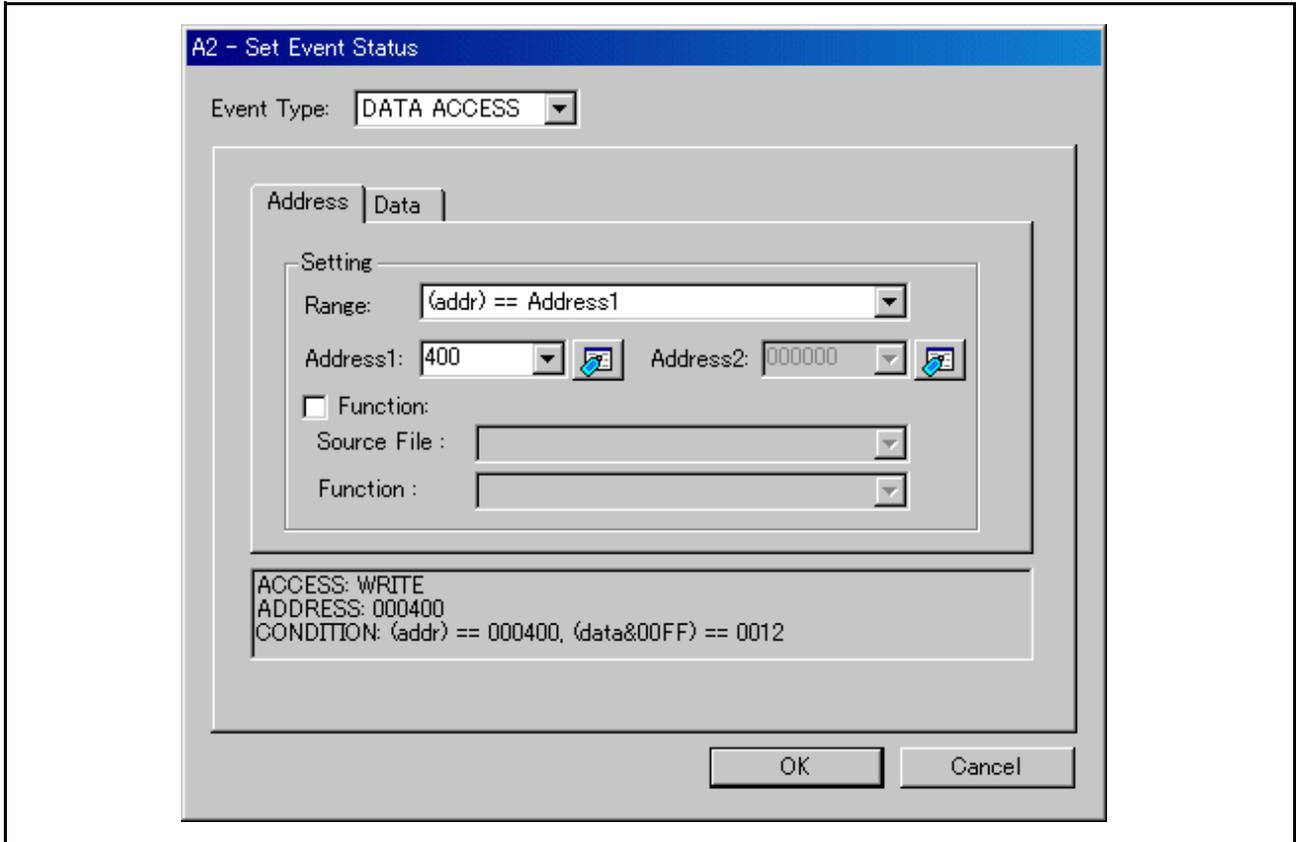
(8 位总线宽度)

请组合指定第一部分和第二部分的逻辑与。

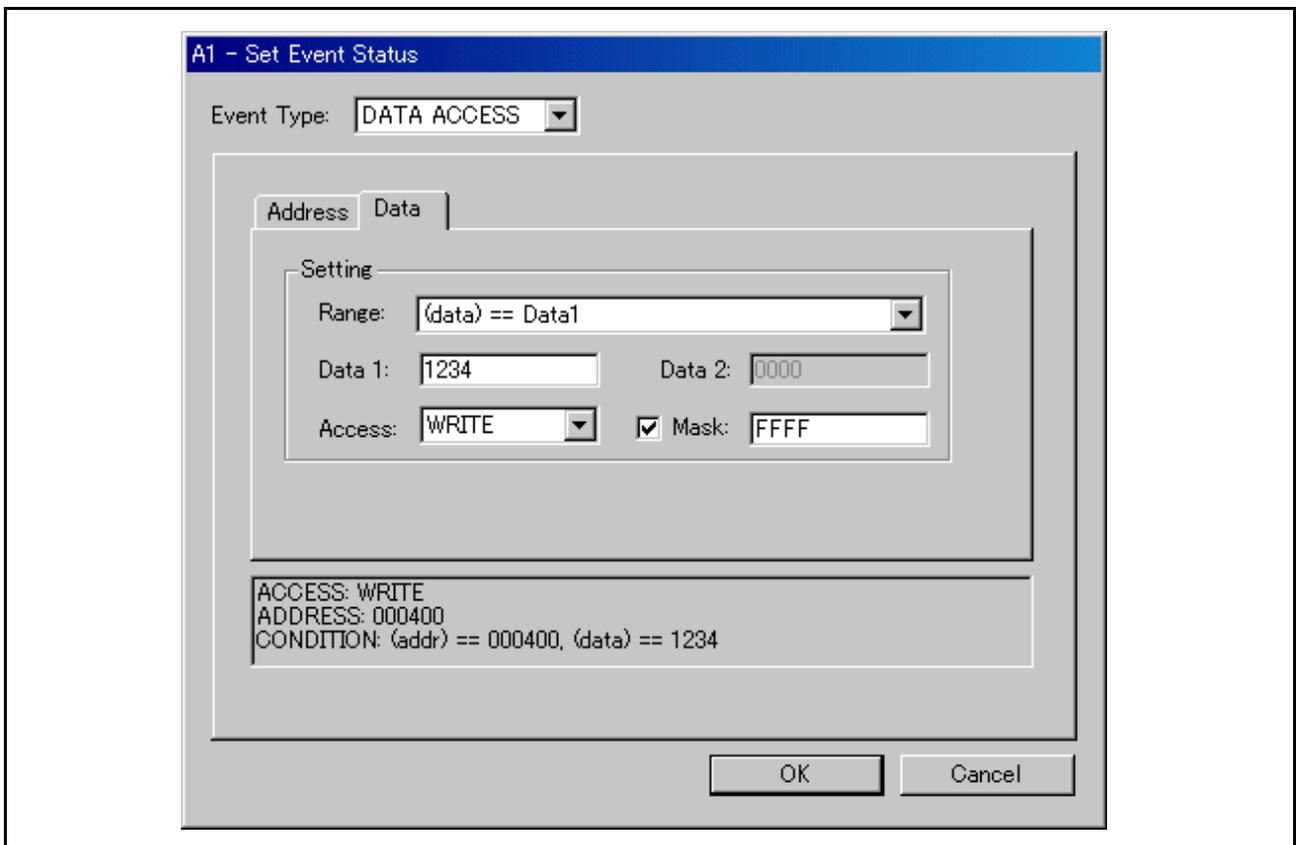
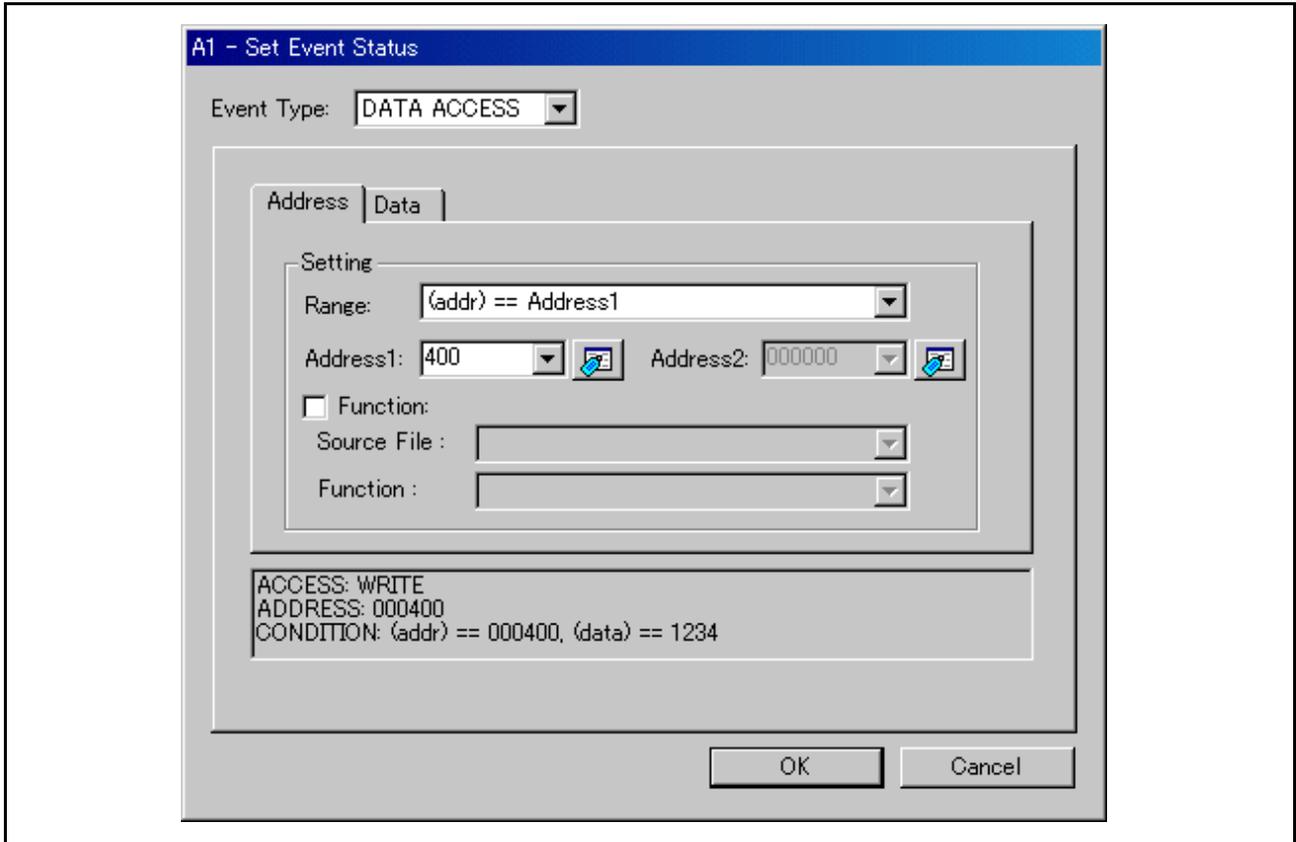
第一部分



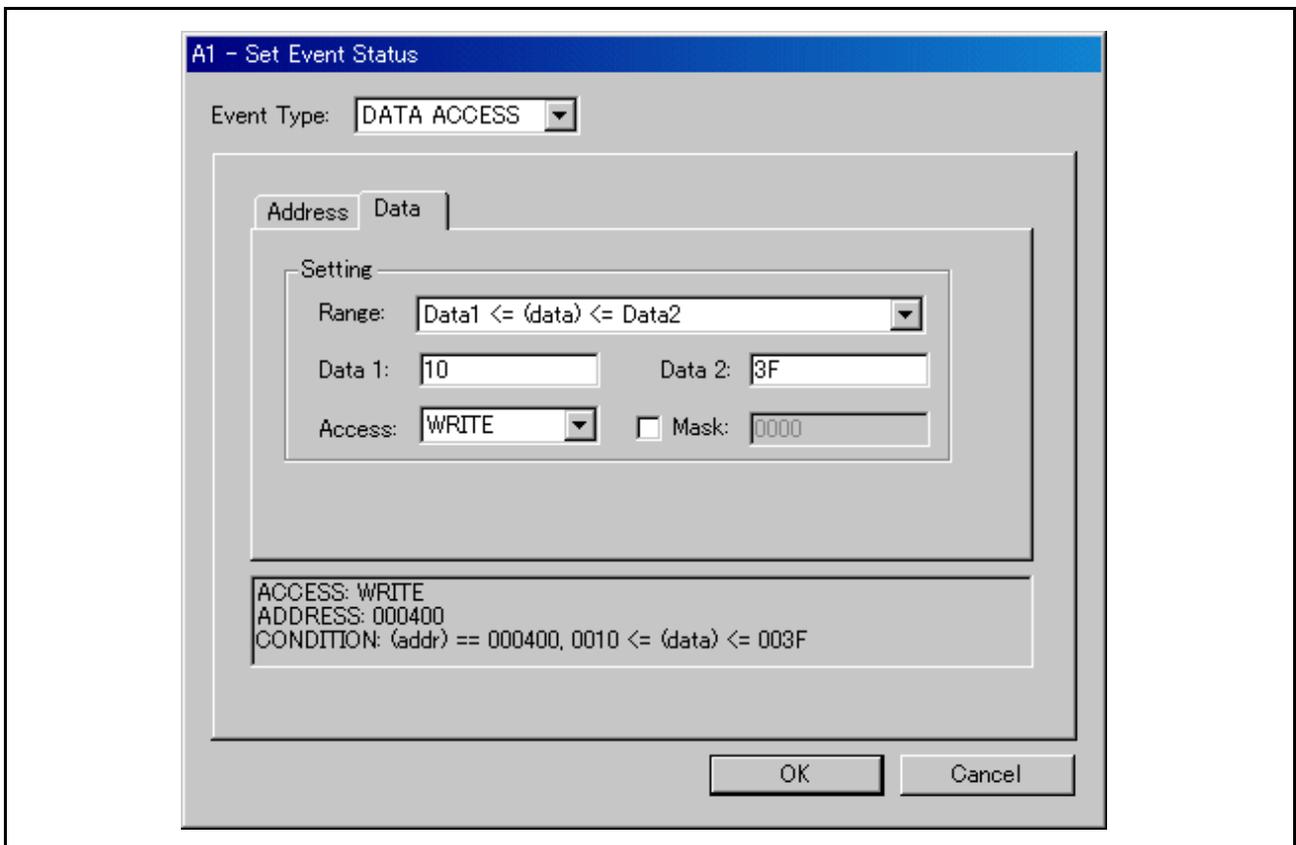
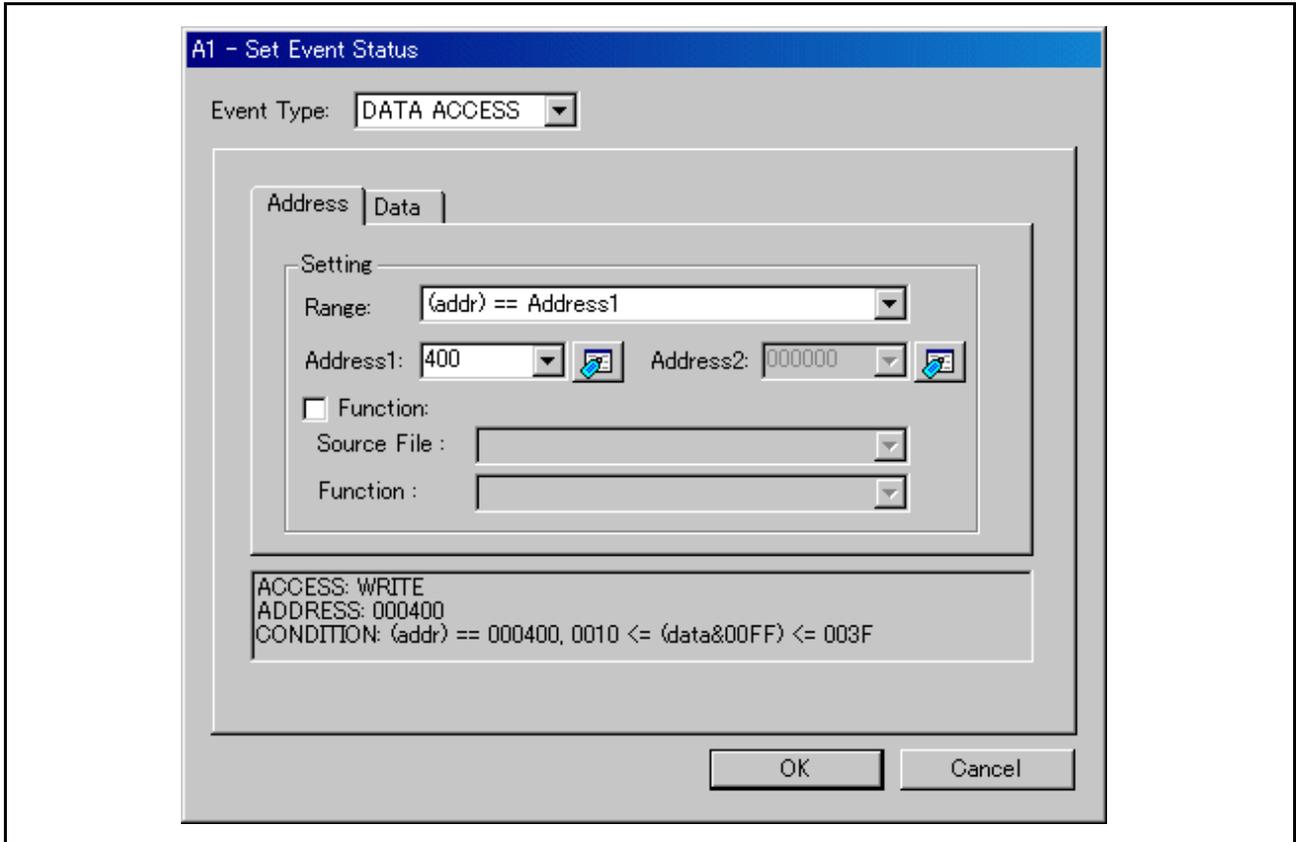
第二部分



(16 位总线宽度)



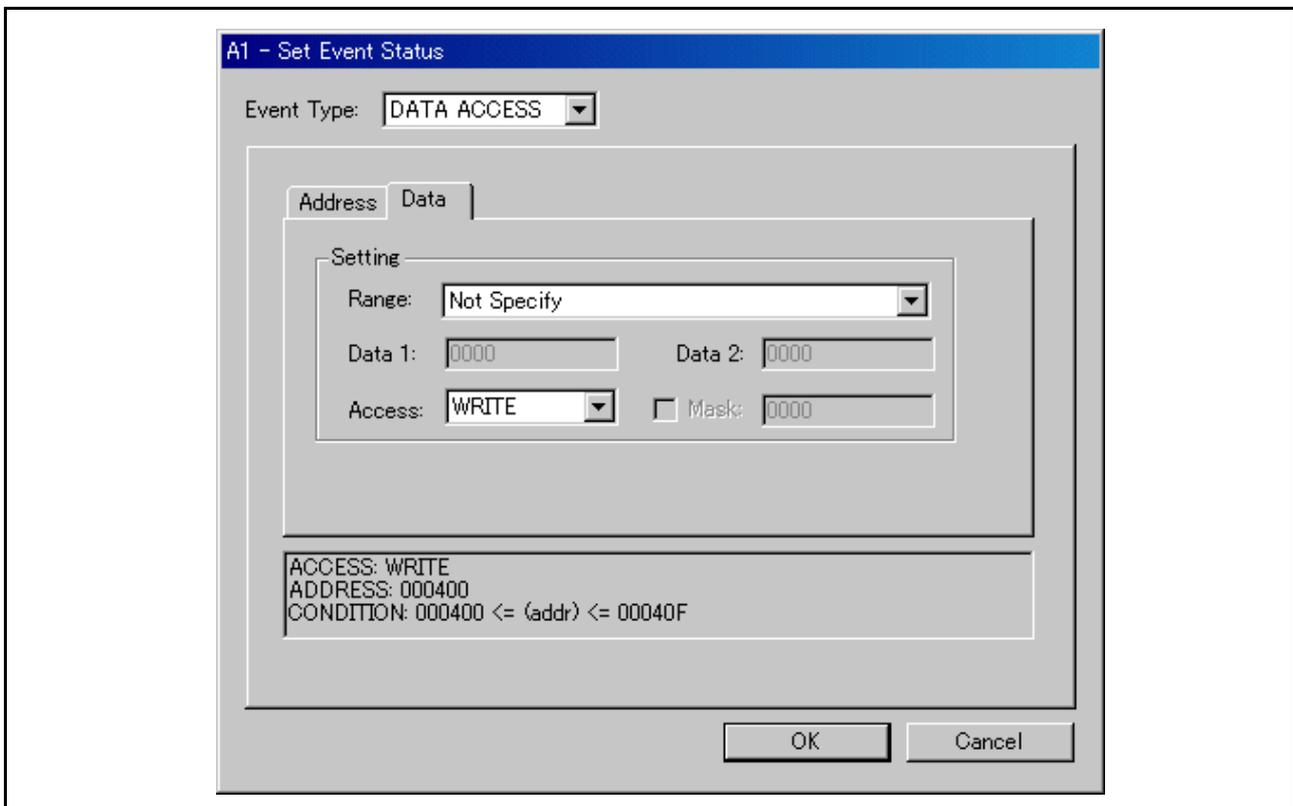
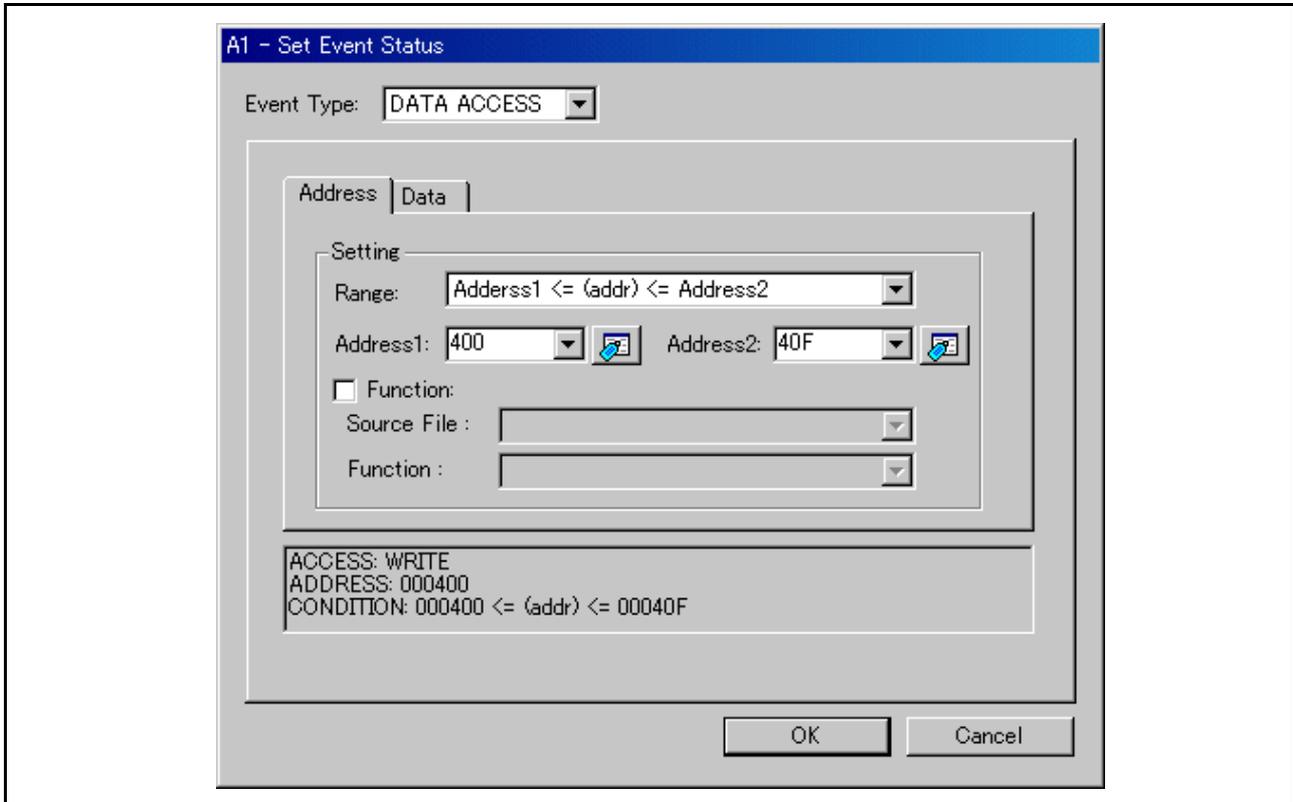
示例) 将数据 10h - 3Fh 写入偶地址 400h



7.7.6.2.2 从指定地址范围读取 / 写入数据

设置如下所示。

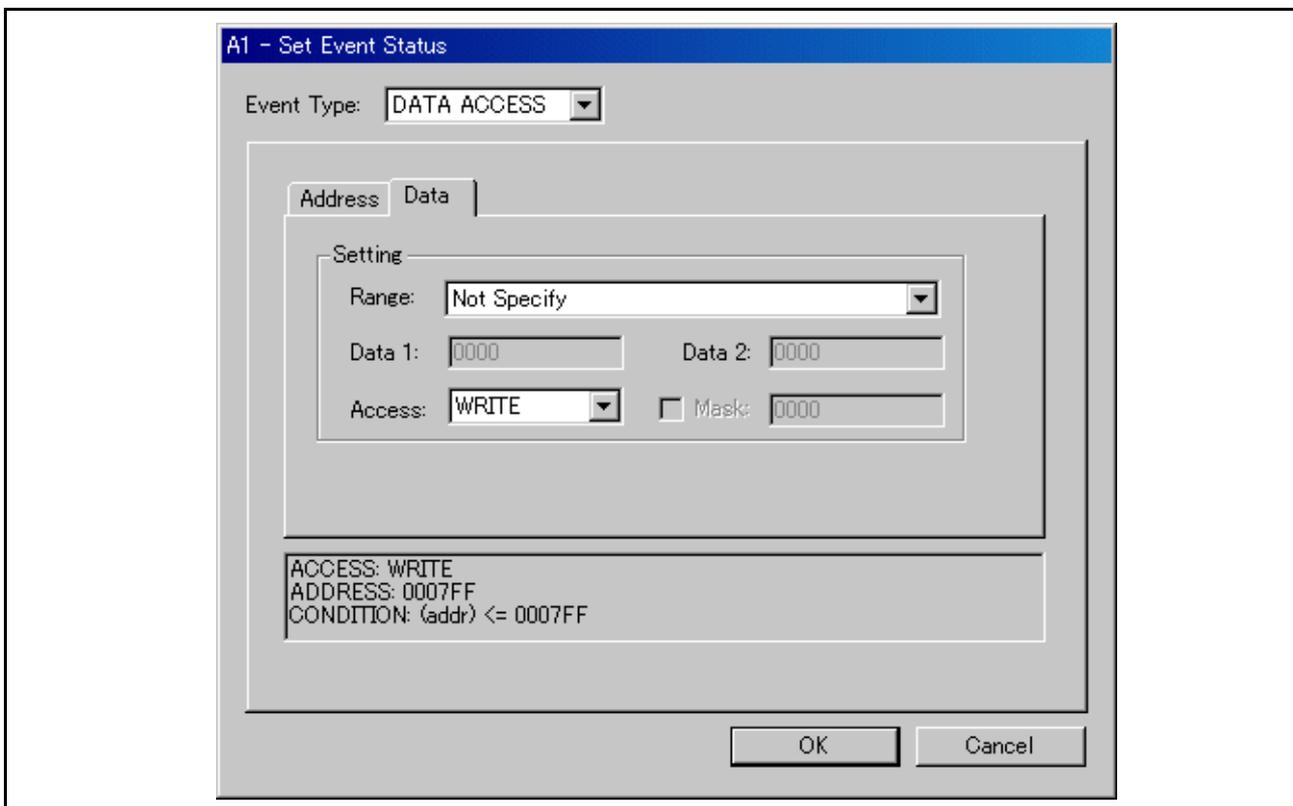
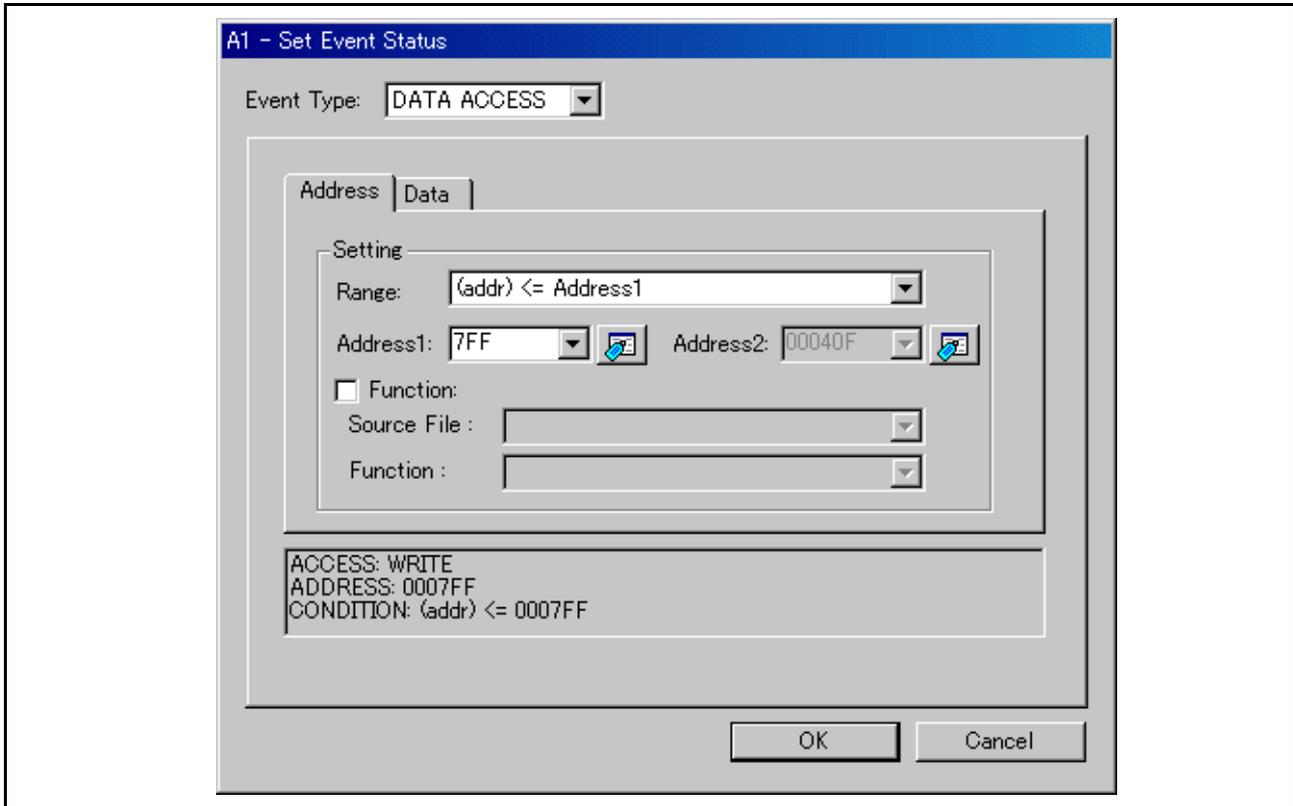
示例) 将数据写入 400h 至 40Fh 的地址范围



7.7.6.2.3 从指定地址范围以外读取 / 写入数据

设置如下所示。

示例) 将数据写入小于 7FFh 的地址



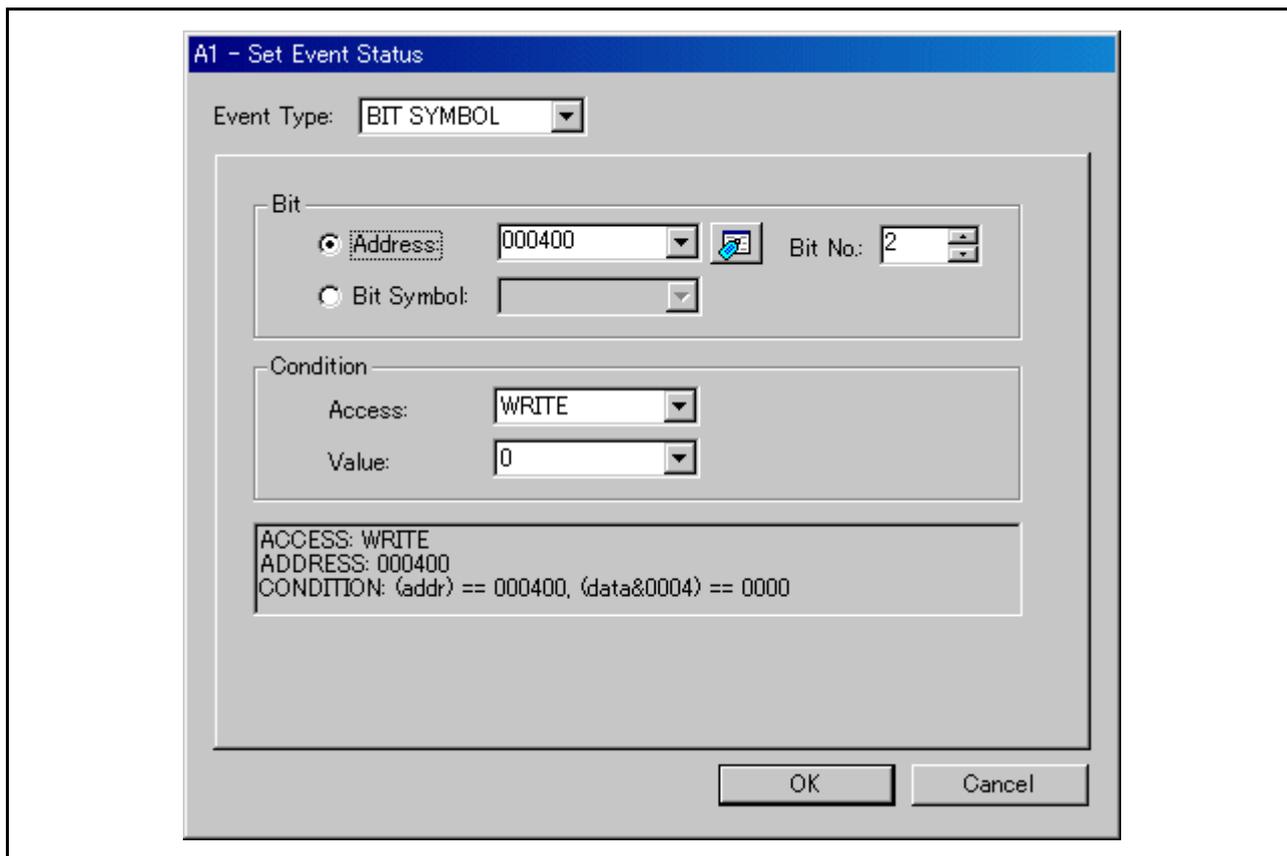
7.7.7 指定事件（位存取）

要指定位存取事件，请将事件选择对话框的 [Event Type]（事件类型）更改为“BIT SYMBOL”。在指定条件下对指定地址的指定位或指定位符号进行存取时建立事件。

7.7.7.1 写入 / 读取指定位

设置如下所示。

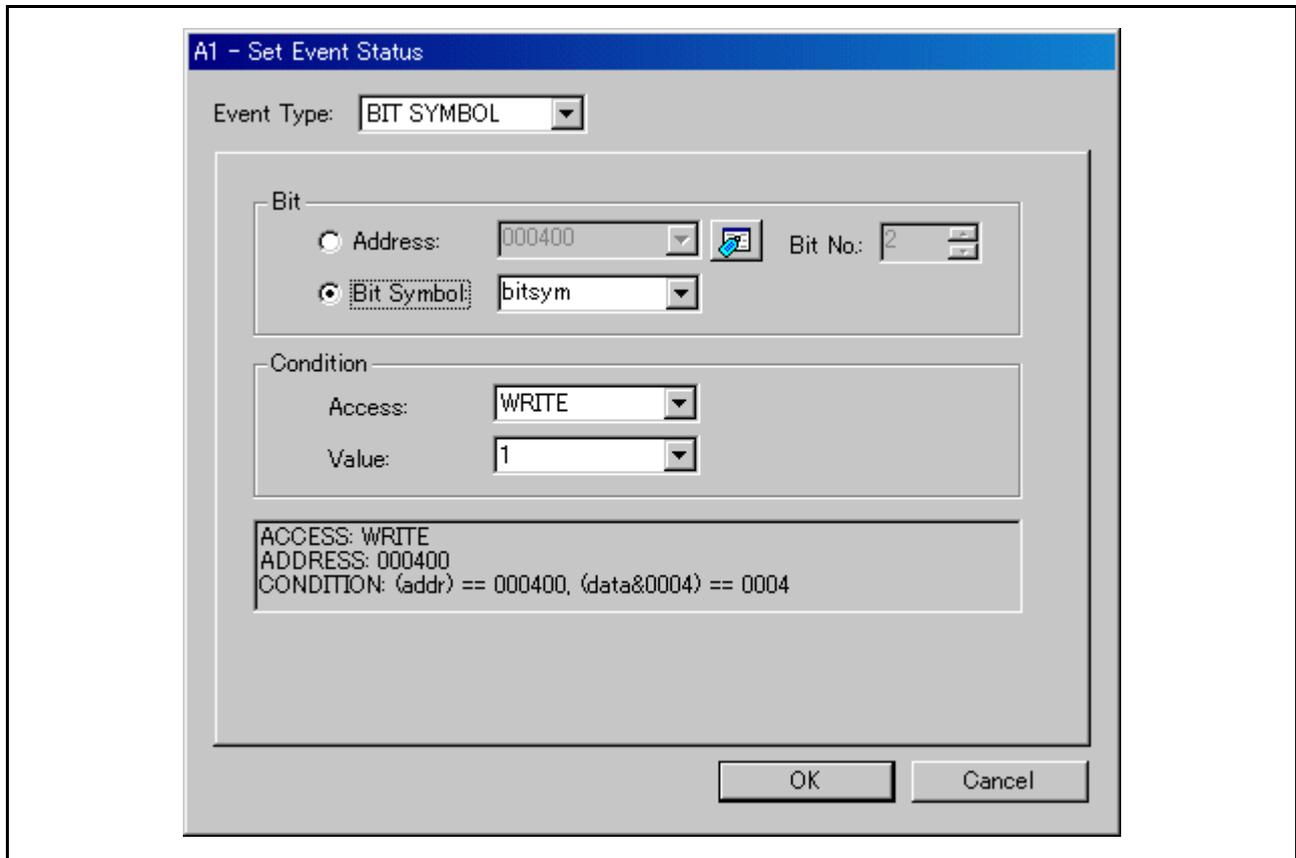
示例) 将“0”写入地址 400h 的第 2 位



7.7.7.1.1 写入 / 读取指定位符号

设置如下所示。

示例) 将 “1” 写入位符号 “bitsym”

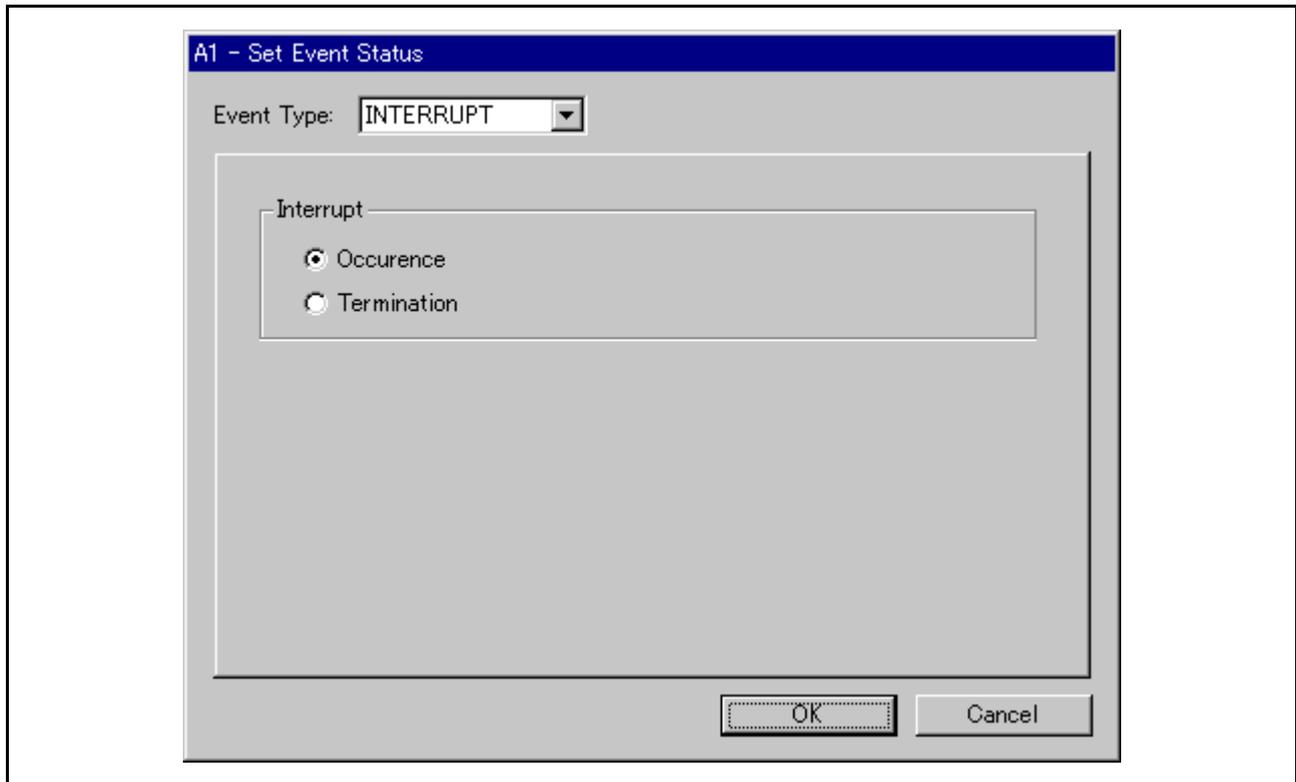


7.7.8 指定事件（中断）

要指定中断事件，请将事件选择对话框的 [Event Type]（事件类型）更改为“INTERRUPT”。将在中断发生或结束时建立事件。

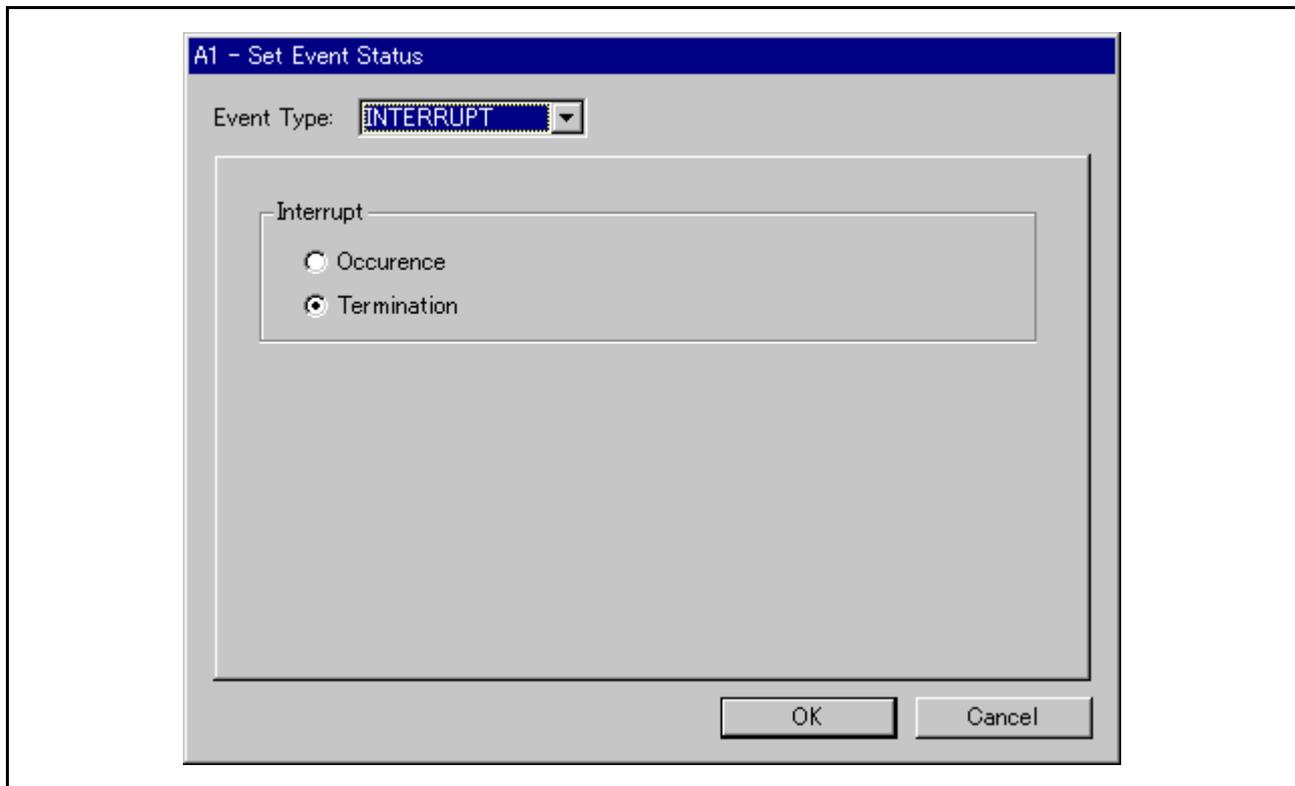
7.7.8.1 中断发生

设置如下所示。



7.7.8.2 中断终止

设置如下所示。



7.7.9 指定事件（外部触发信号）

要指定外部触发信号事件，请将事件选择对话框的 [Event Type]（事件类型）更改为“TRIGGER”。当来自外部跟踪信号输入电缆的信号处于指定状态时建立事件。随仿真器附有外部跟踪信号输入电缆（可传送 8 个信号），将从此电缆检测信号的上升 / 下降沿。

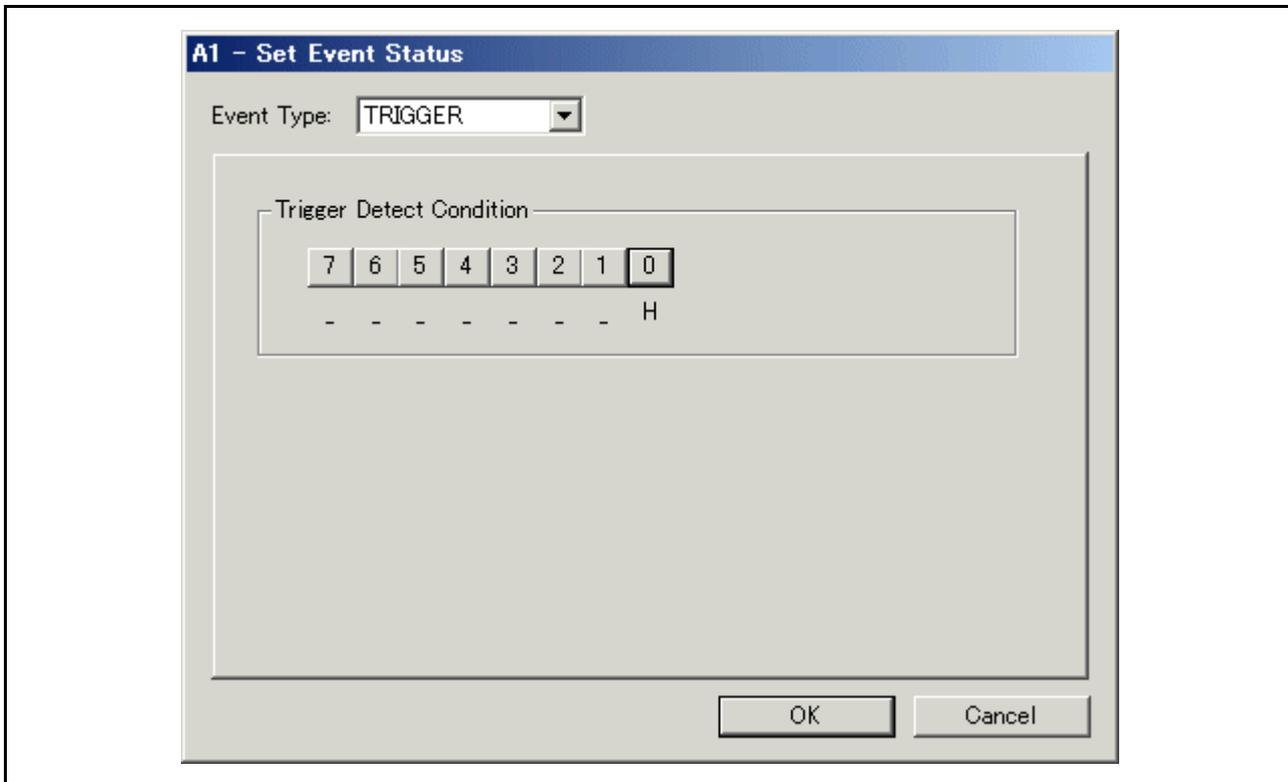
如下所示为来自外部跟踪信号输入电缆的信号名称及其对应的电缆颜色。

信号名称	电缆颜色
EXT0	白
EXT1	棕
EXT2	红
EXT3	橙
EXT4	黄
EXT5	绿
EXT6	蓝
EXT7	紫
GND	黑

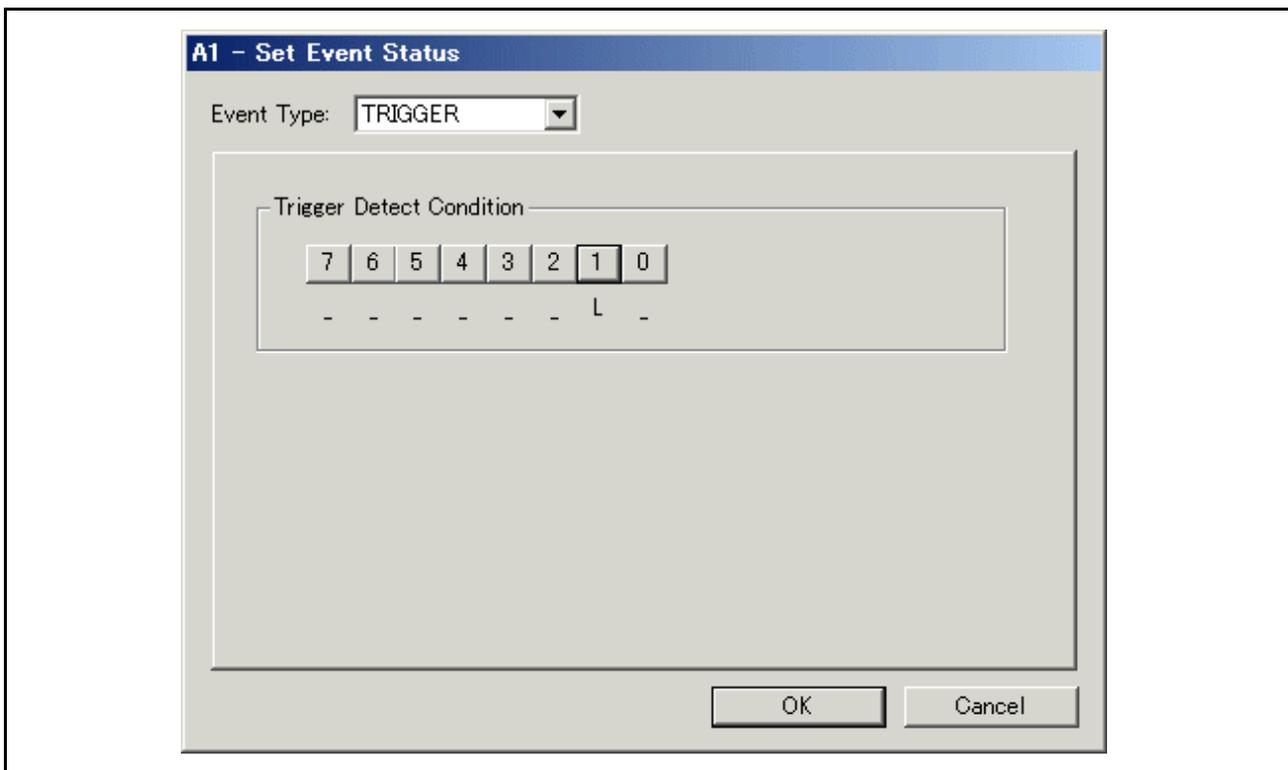
7.7.9.1 检测上升 / 下降沿

设置如下所示。单击各触发按钮，其触发设置将按如下顺序变换：“H”（上升沿）->“L”（下降沿）->“ ”（无设置）。

示例）EXT0（白）信号上升沿



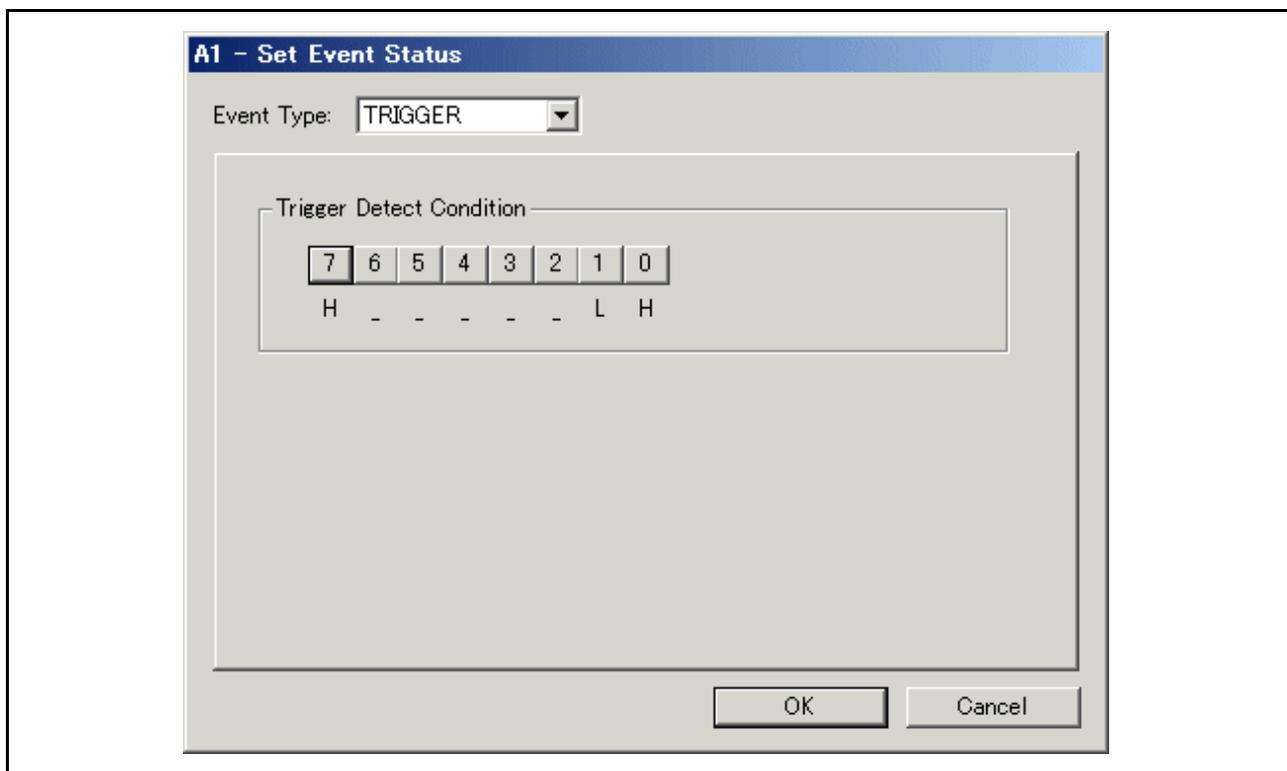
示例）EXT1（棕）信号下降沿



7.7.9.2 上升 / 下降沿组合

设置如下所示。

示例) EXT0 (白) /EXT7 (紫) 信号上升沿, EXT1 (棕) 信号下降沿



7.7.10 指定事件组合条件

使用事件设置窗口的 [Combination] (组合) 框指定事件的组合条件。

可应用对两个或两个以上的事件的组合。

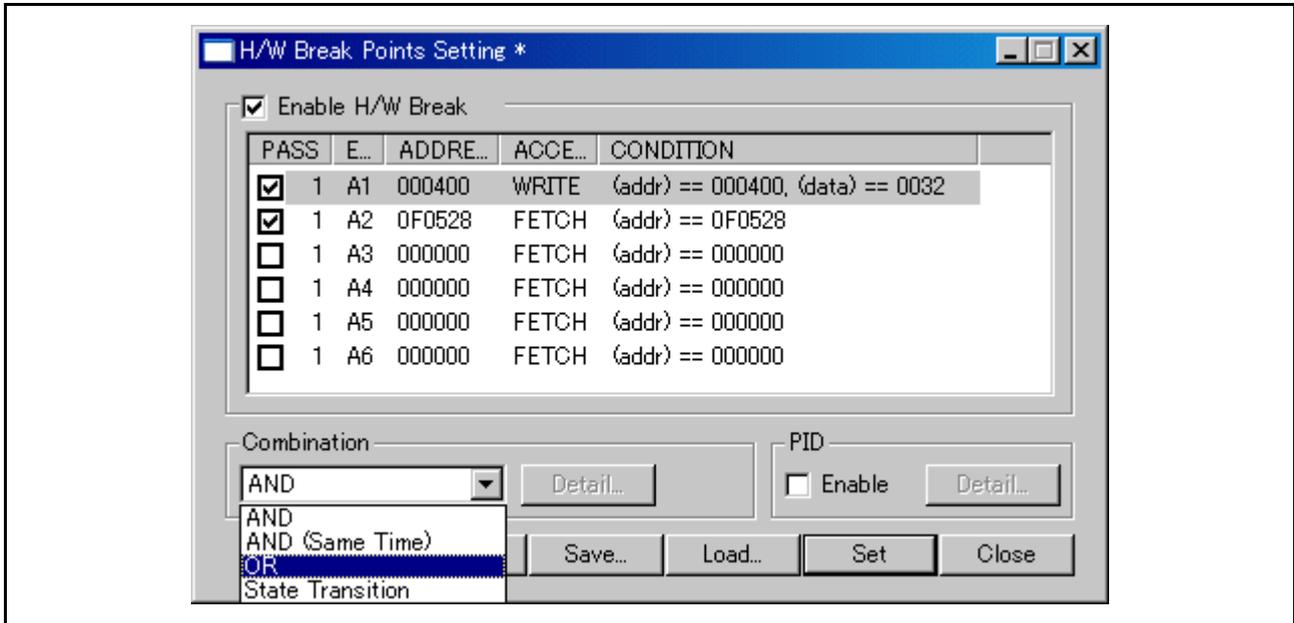
可选择以下组合条件之一。

AND (与)	建立所有指定事件。
AND(Same Time) (与 (同时))	指定的多个事件同时建立。
OR (或)	建立指定事件之一。
STATE TRANSITION (状态转换)	根据状态转换图中输入的断点状态建立。

可为每一事件指定通过计数 (已通过次数), 指定范围为 1-255。如果指定的组合条件为 AND (Same Time) (与 (同时)), 则无法设置通过计数 (值固定为 1)。

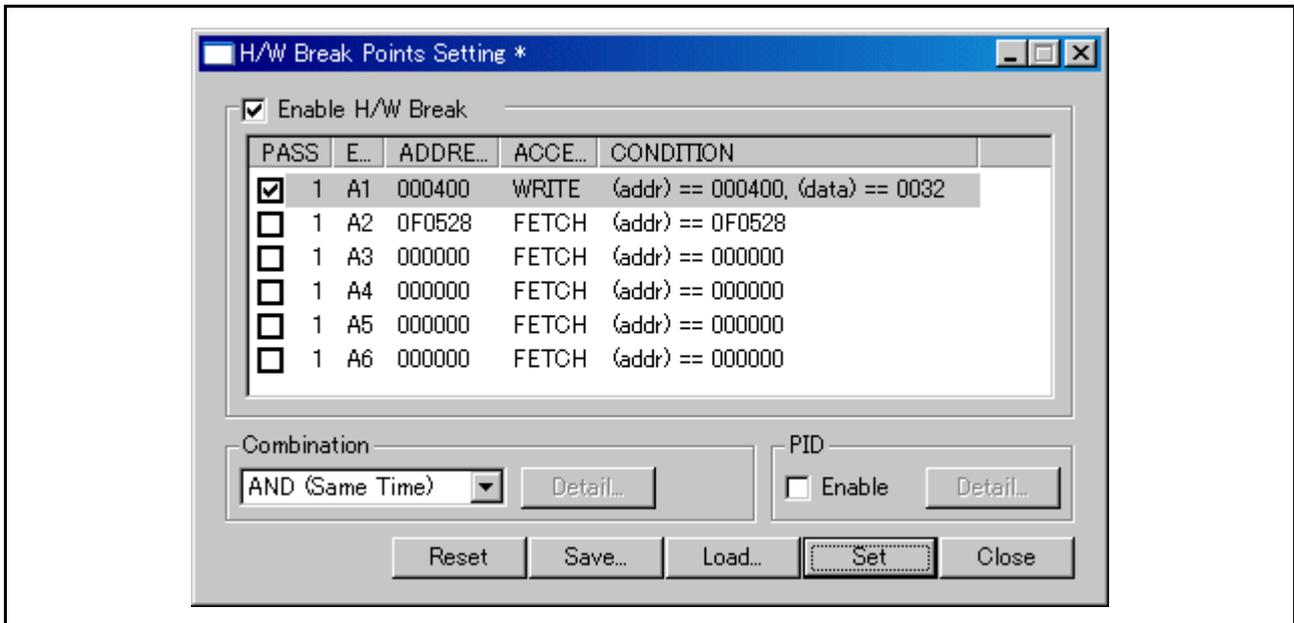
7.7.10.1 选择 AND（与）、OR（或）

将 [Combination]（组合）框改为“AND”（与），指定 AND（与）作为组合条件，或改为“OR”（或）指定组合条件为 OR（或）。接下来，从事件指定区域选取（打开）要使用的事件，并为该事件指定通过计数。要更改通过计数，请在要更改的事件处于选中状态时，单击该事件的通过计数值。



7.7.10.2 选择 AND(Same Time)（与（同时））

将 [Combination]（组合）框改为“AND (Same Time)”（与（同时））。接下来，从事件指定区域选取（打开）要使用的事件。无法指定通过计数（值固定为1）。

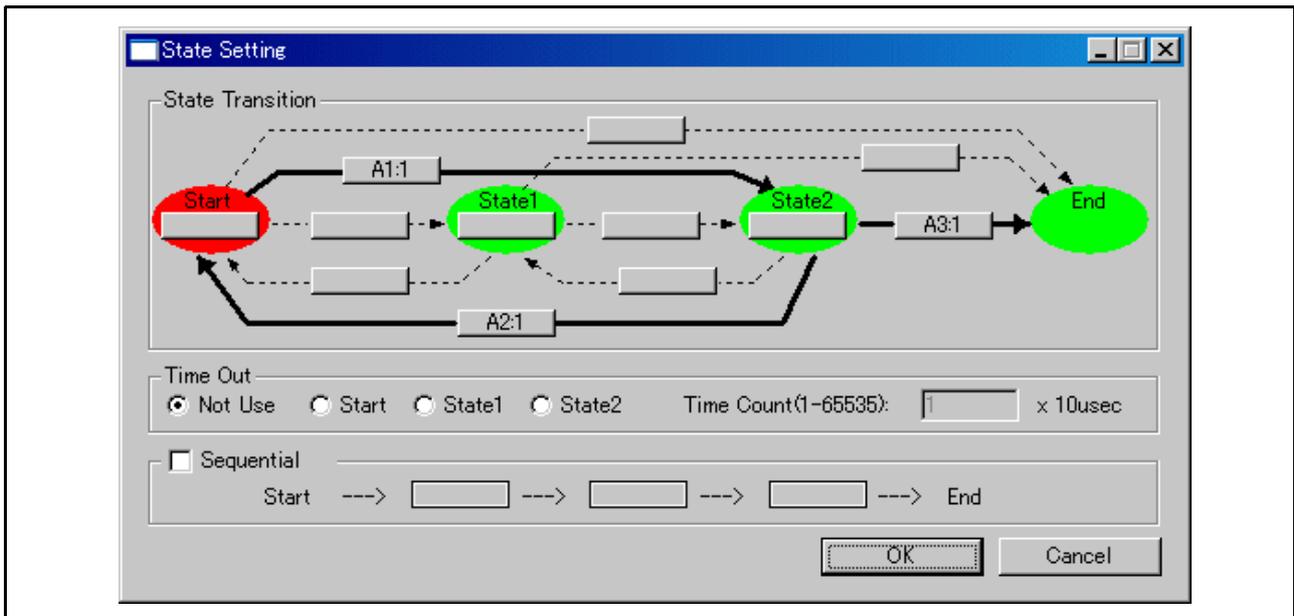


7.7.10.3 选择 State Transition (状态转换)

将 [Combination] (组合) 框改为 “State Transition” (状态转换)。
[Combination] (组合) 框的 [Detail...] (详细信息...) 按钮变为可用, 请单击该按钮。这将打开 [State Setting] (状态设置) 窗口。在此窗口中, 可以使用状态转换图或按顺序指定状态转换。还可以指定状态超时时间。

通过状态转换图指定使用 [State Transition] (状态转换) 栏的状态转换图。单击此处的任一按钮选择要使用的事件。对于箭头上方的按钮, 请指定必须转至箭头指示状态的事件。对于状态 (椭圆形显示部分) 上方的按钮, 请指定必须复位各状态通过计数的事件, 并将该状态的计数复位。选择某事件时, 可以从出现的弹出菜单指定通过计数。

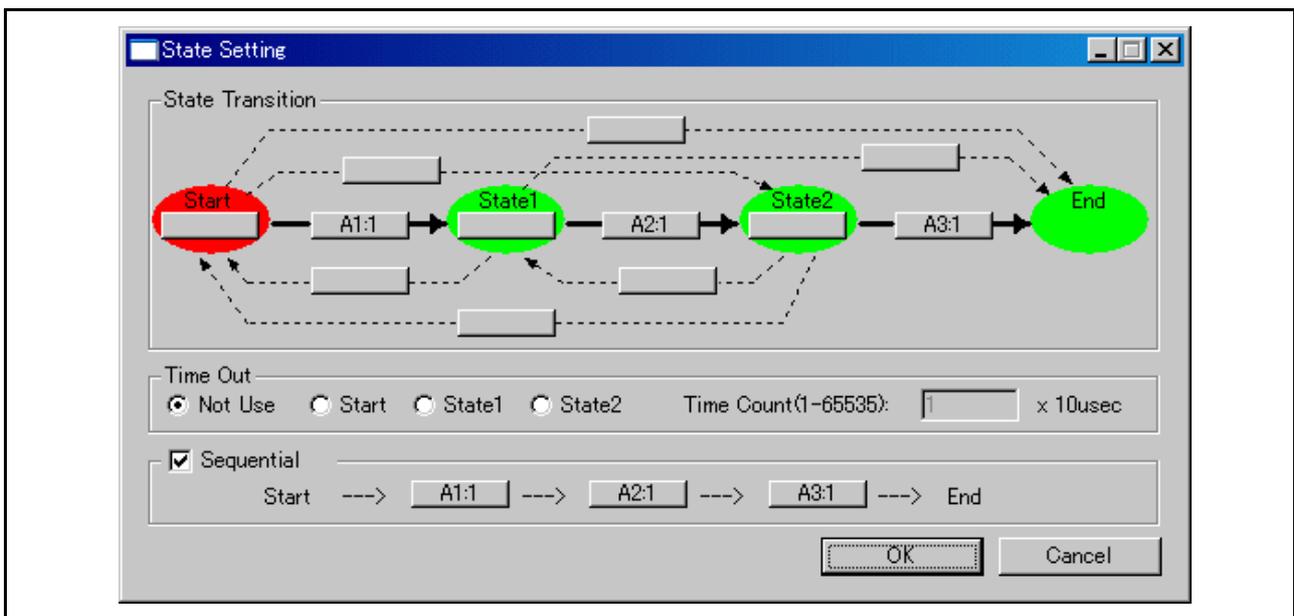
示例: 按事件 A1 到 A2 的顺序满足条件后, 事件 A3 未建立, 事件 4 建立。



按顺序指定

使用 [Sequential] (顺序) 栏中的按钮。选择某事件时, 可以从出现的弹出菜单指定通过计数。此处设置的内容将反映到状态转换图中。

示例: 事件 A1、A2 和 A3 依次出现, 并按此顺序建立。



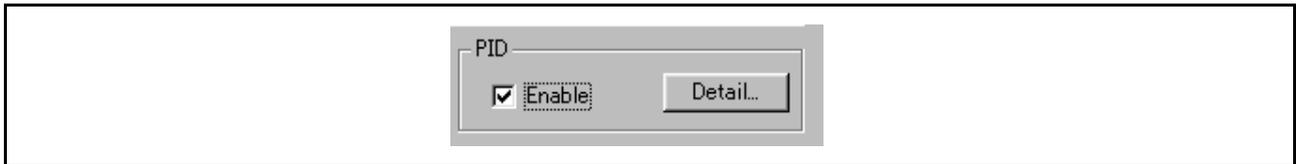
7.7.11 指定进程 ID

通过指定任务名称（任务编号），可以只允许在指定任务中出现的事件。

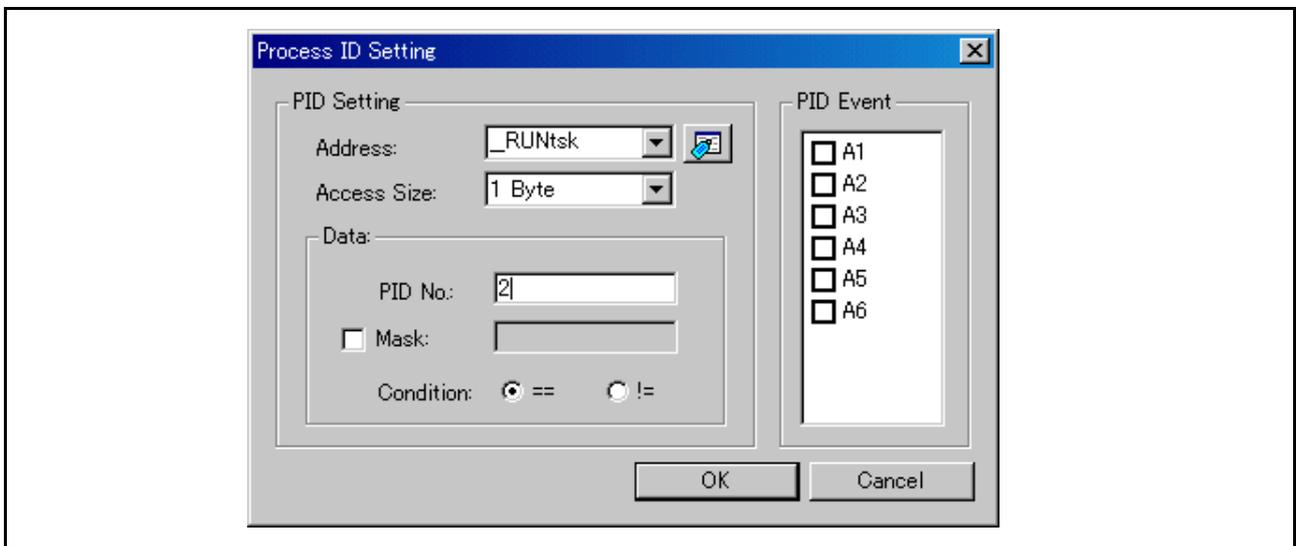
7.7.11.1 指定任务

仅能检测到来自指定任务的事件。无法检测到来自非指定任务的事件。

在事件设置窗口中，选中位于 [PID] 栏中的 [Enable]（允许）复选框。选中此复选框后，其右侧的 [Detail...]（详细信息...）按钮变为可用。



单击位于 [PID] 栏中的 [Detail...]（详细信息...）按钮。这会打开 [Process ID Setting]（进程 ID 设置）对话框。



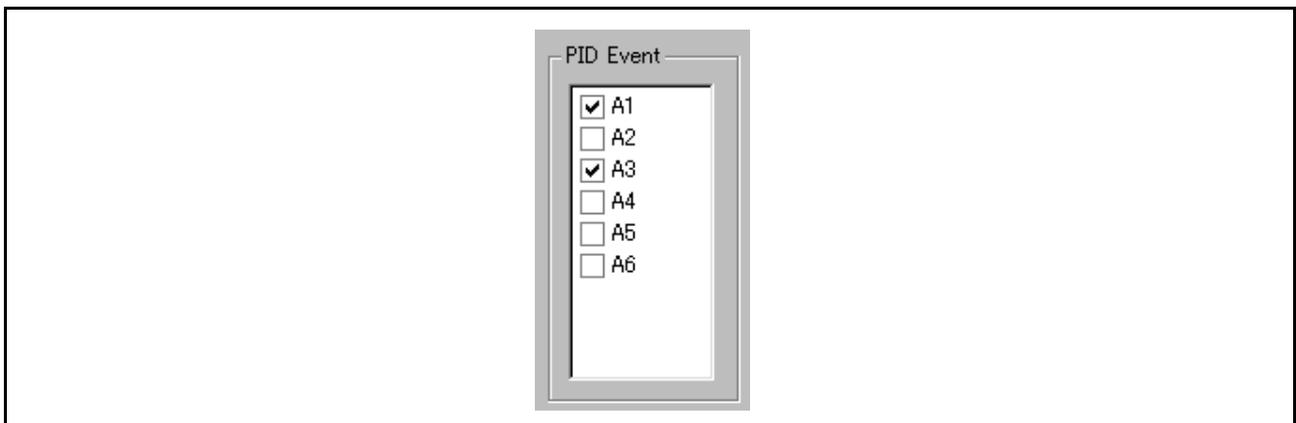
在 PID 设置区域中指定存储器地址，将在此处存储执行任务、数据大小和任务编号（或任务名称）。还可以为任务编号指定掩码。在 [Condition]（条件）栏中，单击单选按钮“==”。

要允许来自非指定任务的事件，请单击单选按钮“!=”。

注意事项

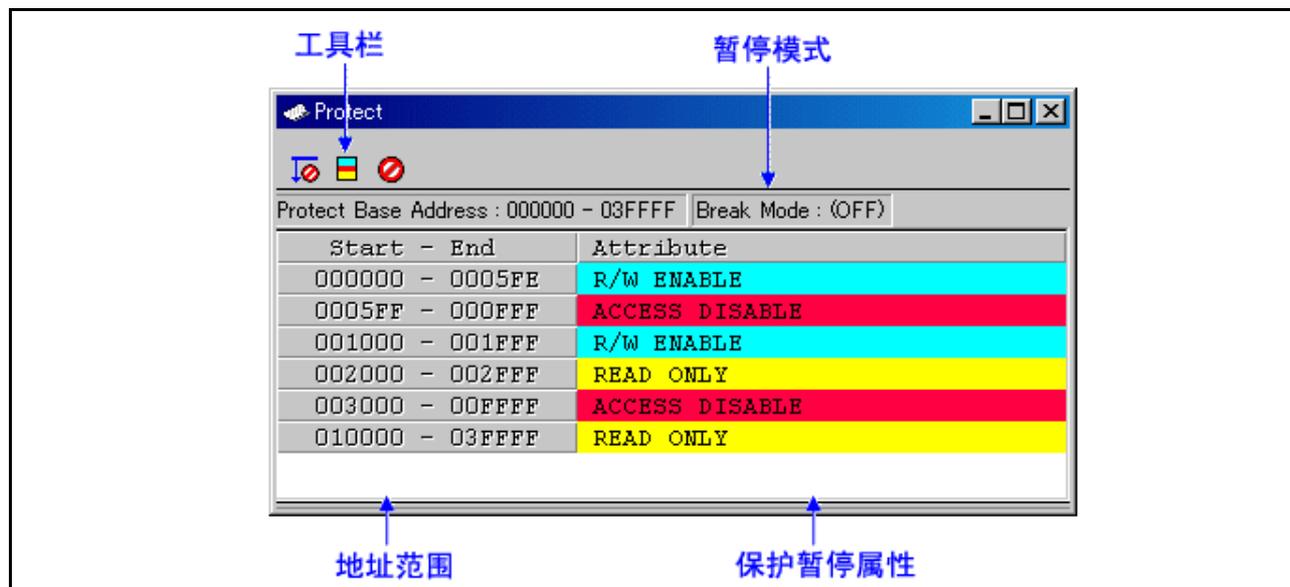
在 PID 设置中，总是确保为 [Address]（地址）区域指定一个偶地址。

在 [PID Event]（PID 事件）区域，选中要允许的事件对应的复选框。仅选中的事件成为要调试的目标。



7.8 [Protect] (保护) 窗口

[Protect] (保护) 窗口可设置仿真器的保护断点 (存储器保护) 功能。



- 启动调试器时保护断点功能处于禁止状态。
 - 共有四种保护断点属性可用，如下所示：
 - “Access Disable” (禁止存取) (读取/写入被禁止，显示为红色)
 - “Read Only” (只读) (写入被禁止，显示为黄色)
 - “Write Only” (只写) (读取被禁止，显示为绿色)
 - “R/W Enable” (允许读/写) (允许读取/写入，显示为天蓝色)
 - 您可以在从 256 字节单元至整个存储空间 (16 MB) 的范围内指定存储器存取属性。
 - 可以使用如下两种方式设置保护断点。
 - 从目标程序会话信息中指定。
 - 指定所需地址范围的存储器属性。
- * 指定的范围将调整为与 256 字节分界线对齐。
(示例：如果指定范围为 0x456 至 0x567，则将其调整为 0x400 至 0x5FF。)

7.8.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能
[Section] (节)	通过节信息设置保护断点属性。
[Base Address...] (基址...)	更改保护基址。
[Attribute...] (属性...)	设置保护断点属性。
[Mode] (模式)	在允许或禁止保护断点功能之间切换。
[Toolbar display] (工具栏显示)	显示工具栏。
[Customize toolbar...] (自定义工具栏...)	打开工具栏自定义对话框。
[Allow Docking] (允许入坞)	允许窗口入坞。
[Hide] (隐藏)	隐藏窗口。

7.9 [Address Interrupt Break Point Setting]（地址中断断点设置）窗口

通过 [Address Interrupt Break Point Setting]（地址中断断点设置）窗口可以设置地址中断断点。

在指定地址执行指令之前，该功能会及时停止目标程序。该功能是通过使用 MCU 的地址匹配中断来实现的。因此只有在用户应用程序中未使用地址匹配中断时，才能使用地址中断断点功能。



- 只有在使用地址中断断点功能时，此窗口才可用。

使用 [Init]（初始化）对话框中的 [MCU] 选项卡可以指定是否要使用地址中断断点功能。在此选项卡上，选中 [Enable the Address Match Interrupt Break Function]（允许地址匹配中断断点功能）复选框。（详细信息）。

- 可以设置的地址中断断点数因产品而异。
- 可以通过“Address”（地址）或“Filename”（文件名）+“Line No.”（行号）来指定断点。
- 如果设置了多个断点，则遇到任一断点地址时，程序停止执行（OR 条件）。
- 可以清除、允许或禁止断点显示区中通过单击选中的断点。也可以通过双击断点来允许和禁止断点。
- 单击“Save”（保存）按钮，将软件断点保存到文件。要从已保存的文件再装入软件断点设置，请单击“Load”（装入）按钮。
- 如果从文件中装入断点，这些断点会添加到任何现有的断点中。

7.9.1 命令按钮

此窗口中的按钮具有以下含义。

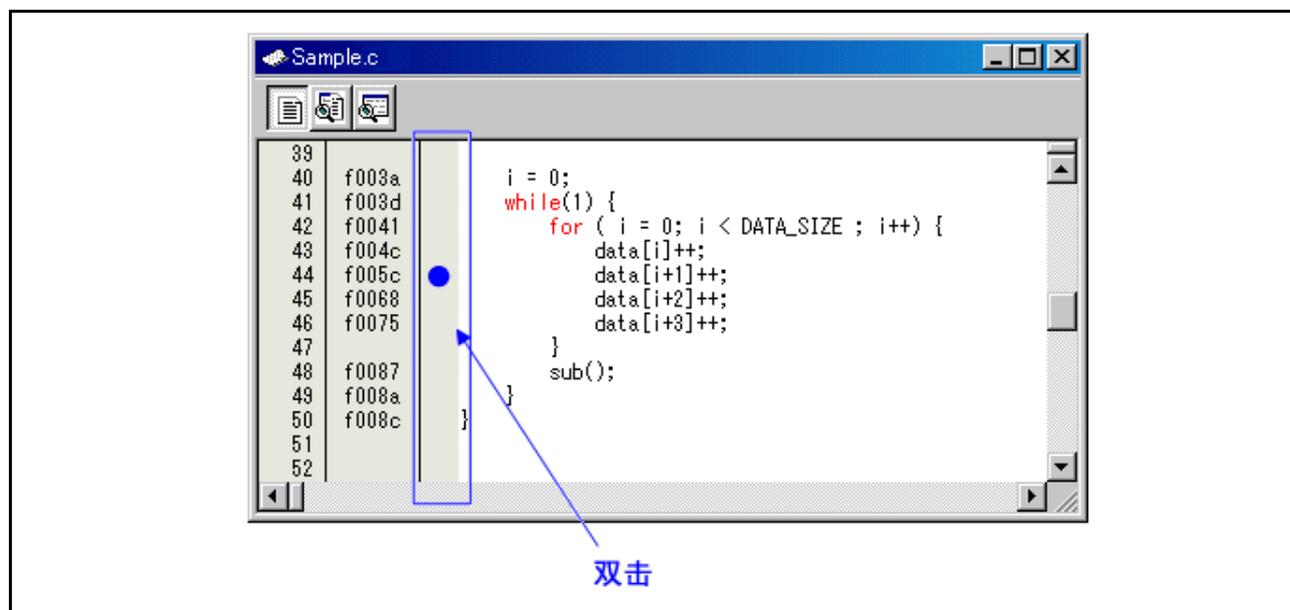
按钮	功能
[Load...] (装入...)	从保存了设置信息的文件装入设置信息。
[Save...] (保存...)	将窗口中设置的内容保存到文件中。
[Help] (帮助)	显示此窗口的帮助信息。
[Add] (添加)	添加断点。
[Refer...] (引用...)	打开文件选择对话框。
[Close] (关闭)	关闭窗口。
[Delete] (删除)	移除选定断点。
[Delete All] (全部删除)	移除所有断点。
[Enable] (允许)	允许所选断点。
[All Enable] (全部允许)	允许所有断点。
[Disable] (禁止)	禁止所选断点。
[All Disable] (全部禁止)	禁止所有断点。
[View] (视图)	在 [Editor(Source)] (编辑器 (源)) 窗口中显示所选断点位置。

7.9.2 从 [Editor(Source)] (编辑器 (源)) 窗口设置和删除断点

只有在用户应用程序中未使用地址匹配中断时，才能使用地址中断断点功能。

有关详细信息，请参阅“1.2.4 关于地址中断断点功能”。

可以在 [Editor(Source)] (编辑器 (源)) 窗口中设置断点。若要进行设置，请双击要设置断点的行的断点设置区 (“Address Match Interrupt Break” (地址匹配中断断点) 列)。(设置了断点的行上会显示一个蓝色标记。)

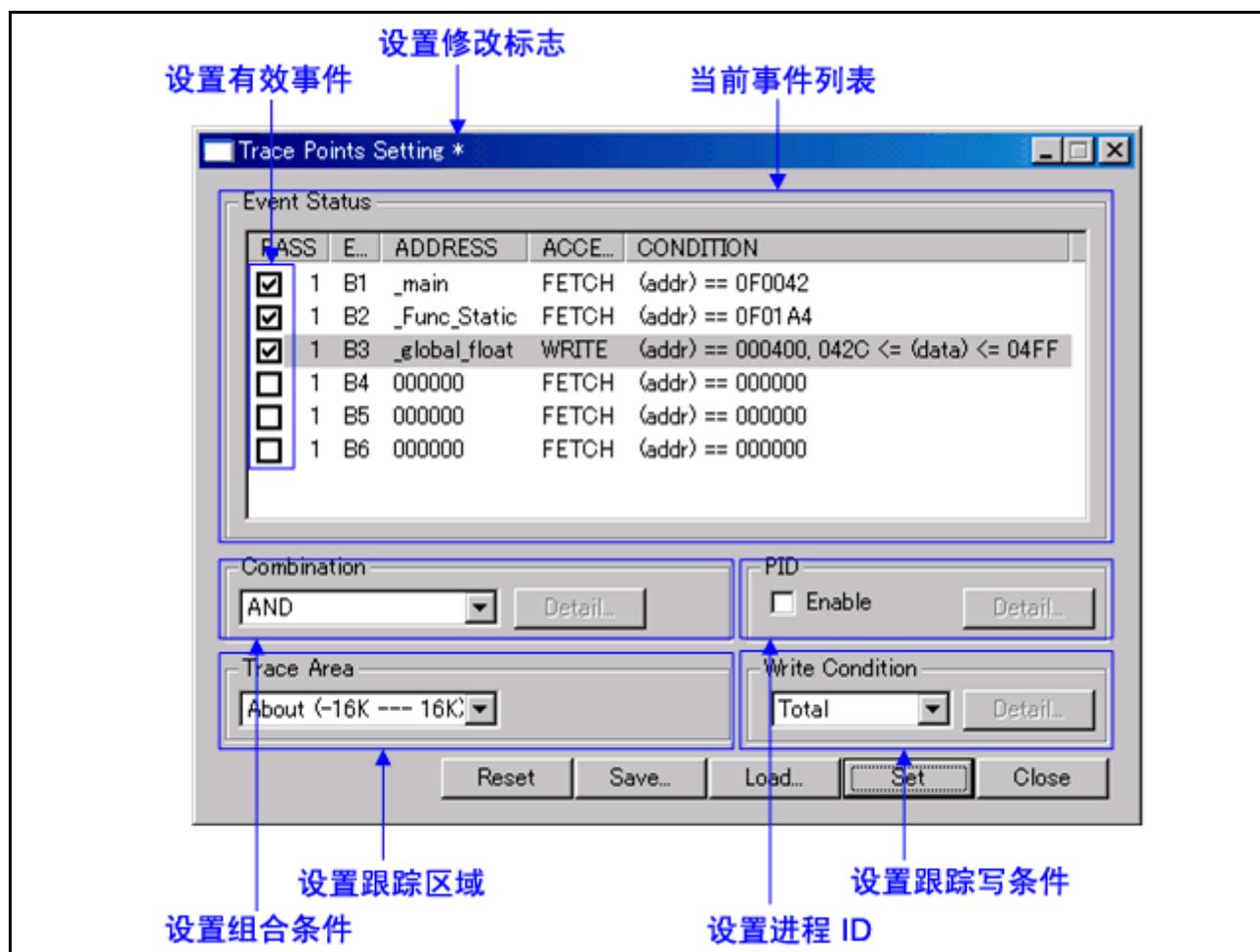


再次双击断点设置区 (“Address Match Interrupt Break” (地址匹配中断断点) 列) 即可删除断点。

在 [Editor(Source)] (编辑器 (源)) 窗口中，“Address Match Interrupt Break” 列默认设置为 “Enable” (允许) 显示。若要擦除此列，请选择主菜单 - [Edit] (编辑) -> [Define Column Format] (定义列格式) 打开对话框，在此对话框中取消选中 [Address Match Interrupt Break] (地址匹配中断断点) 复选框。“Address Match Interrupt Break” (地址匹配中断断点) 列会从所有的 [Editor (Source)] (编辑器 (源)) 窗口中擦除。在 [Editor (Source)] (编辑器 (源)) 窗口中选择弹出菜单 - [Columns] (列) -> [Address Match Interrupt Break] (地址匹配中断断点)，可以为每个 [Editor (Source)] (编辑器 (源)) 窗口设置一列。

7.10 [Trace Point Setting] (跟踪点设置) 窗口

[Trace Point Setting] (跟踪点设置) 窗口用于设置跟踪点。



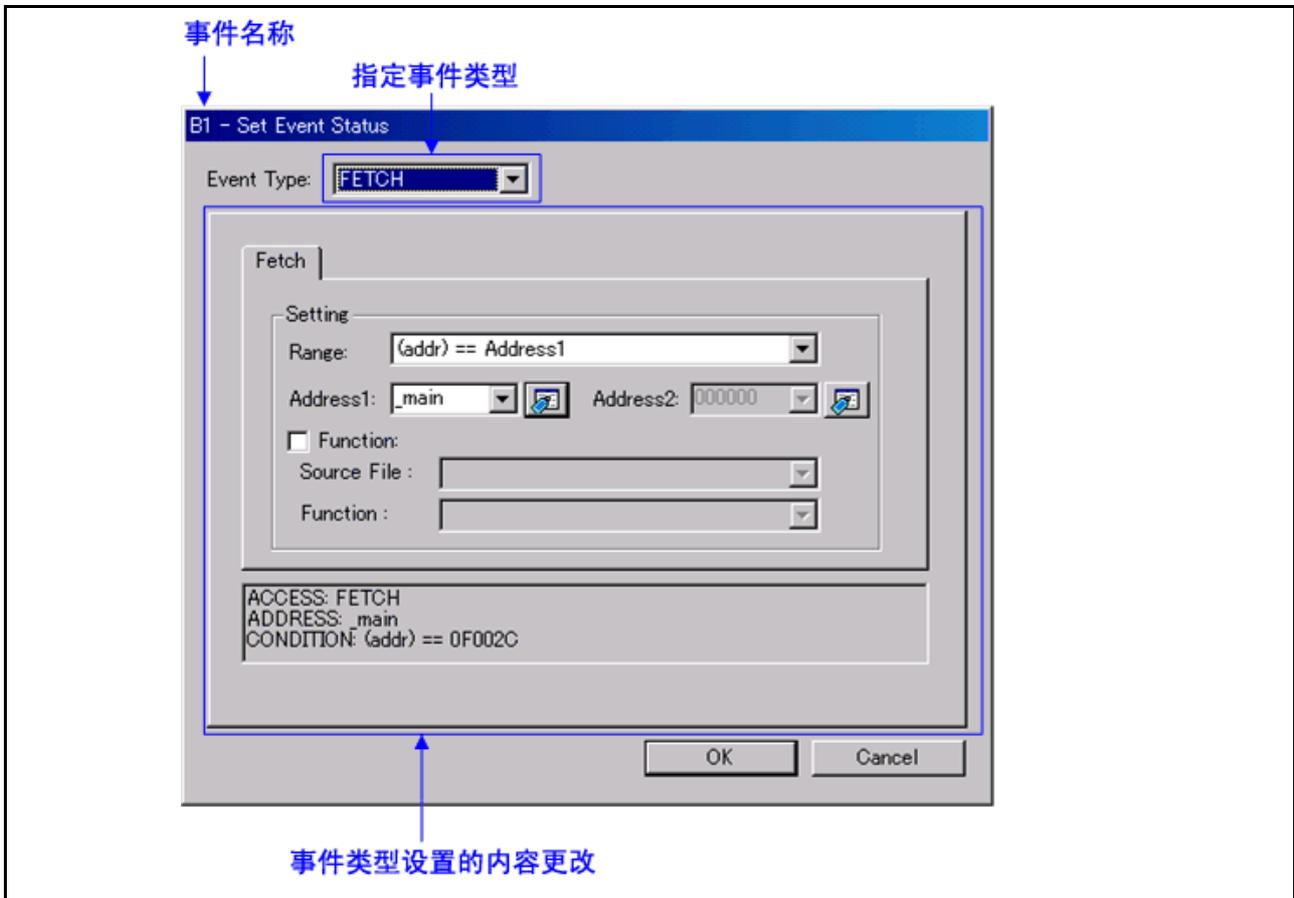
- 下列事件可以指定为跟踪事件。如果事件内容发生变化，则事件的标题栏上会标上一个星号(*)。设置仿真器后不再显示星号(*)。

Fetch (取)、Memory Access (存储器存取)、Bit Access (位存取)、Interrupt (中断)、Trigger (触发)

- 最多可使用八个点处的事件。
- 可以通过下列方式之一组合这些事件：
 - 当所有的有效事件都已建立时进行跟踪 ([AND] (与) 条件)
 - 当所有的有效事件都已同时建立时进行跟踪 (同时的 [AND] (与) 条件)
 - 当有效事件之一已建立时进行跟踪 ([OR] (或) 条件)
 - 在状态转换过程中进入断点状态时进行跟踪 ([State Transition] (状态转换) 条件)

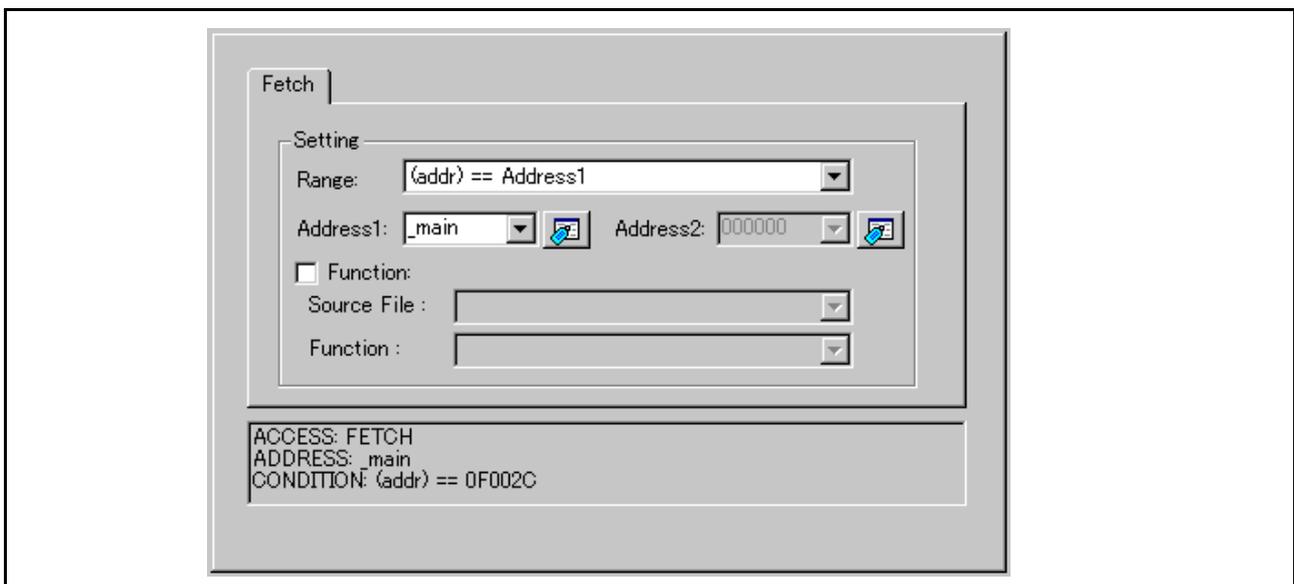
7.10.1 指定跟踪事件

若要设置事件，请从 [Trace Point Setting]（跟踪点设置）窗口的事件设置区双击以选择要设置的事件。此时会打开如下对话框。

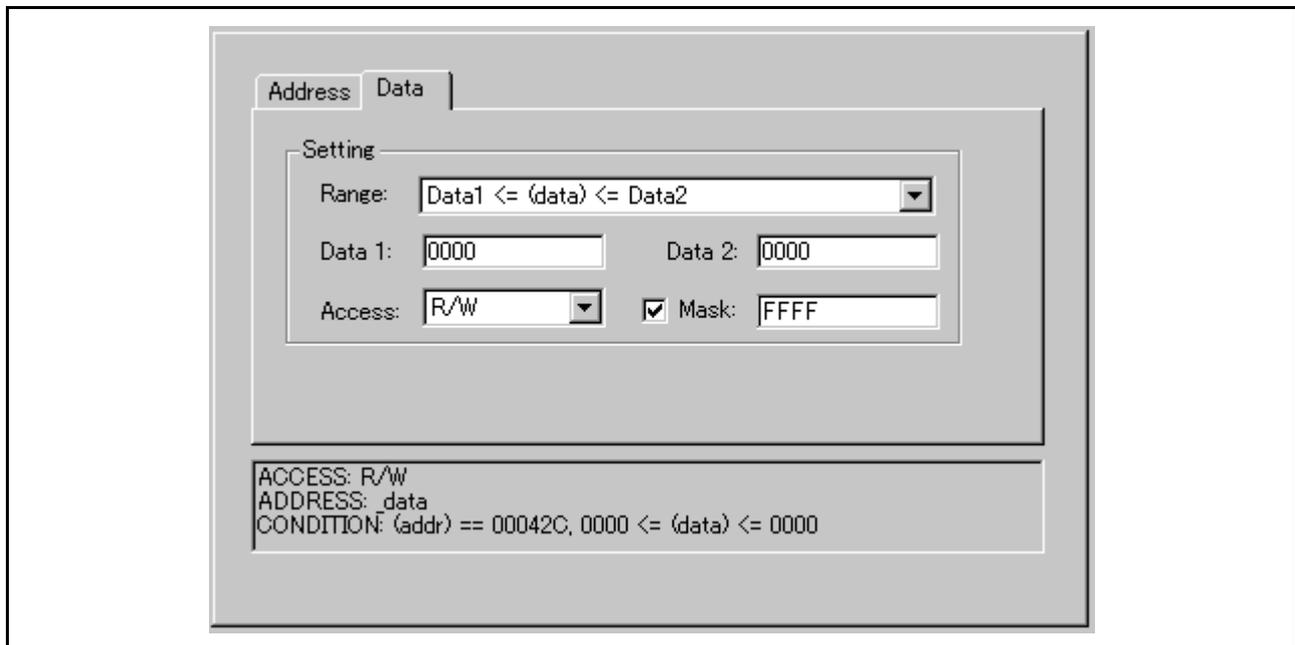


通过在此对话框中指定 [Event Type]（事件类型），可以设置以下事件。

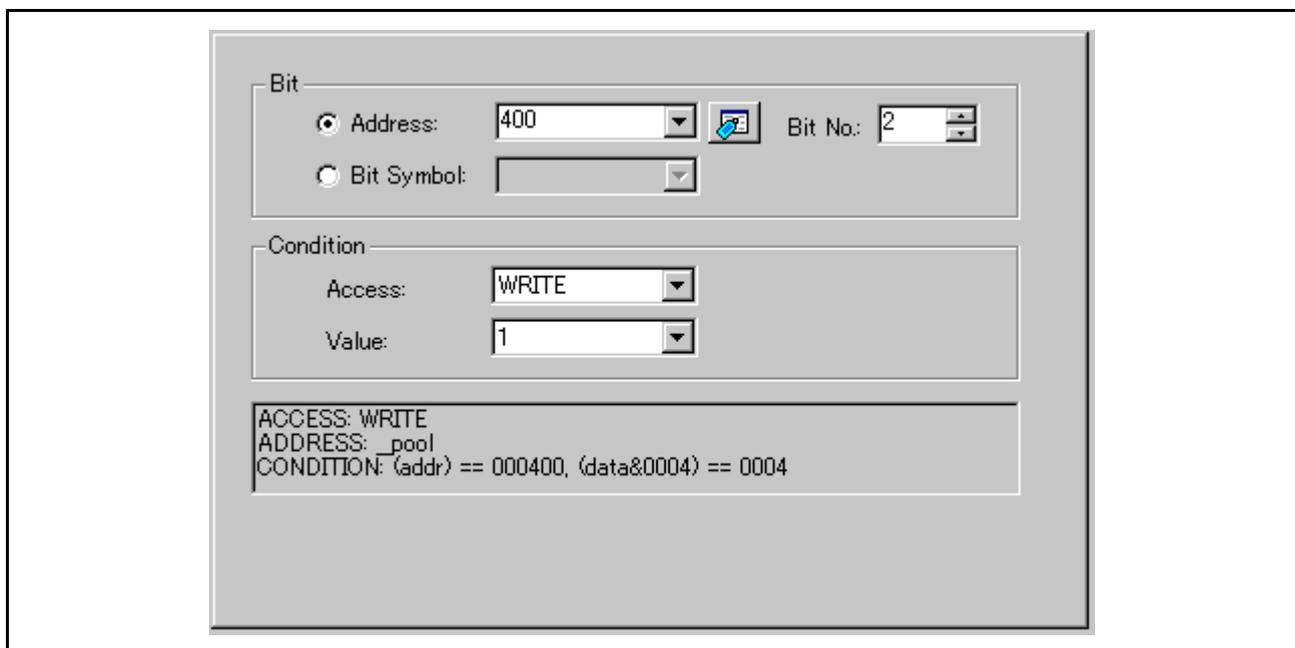
- 选择 [FETCH]（取）时跟踪取指令。



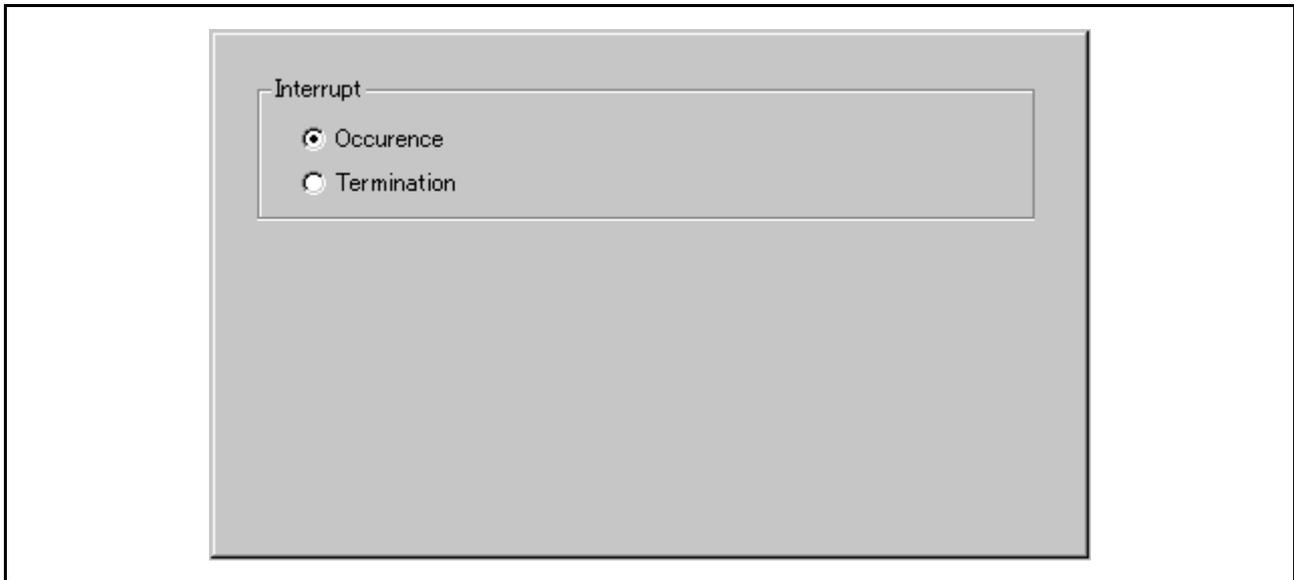
- 选择 [DATA ACCESS] (数据存取) 时跟踪存储器存取。



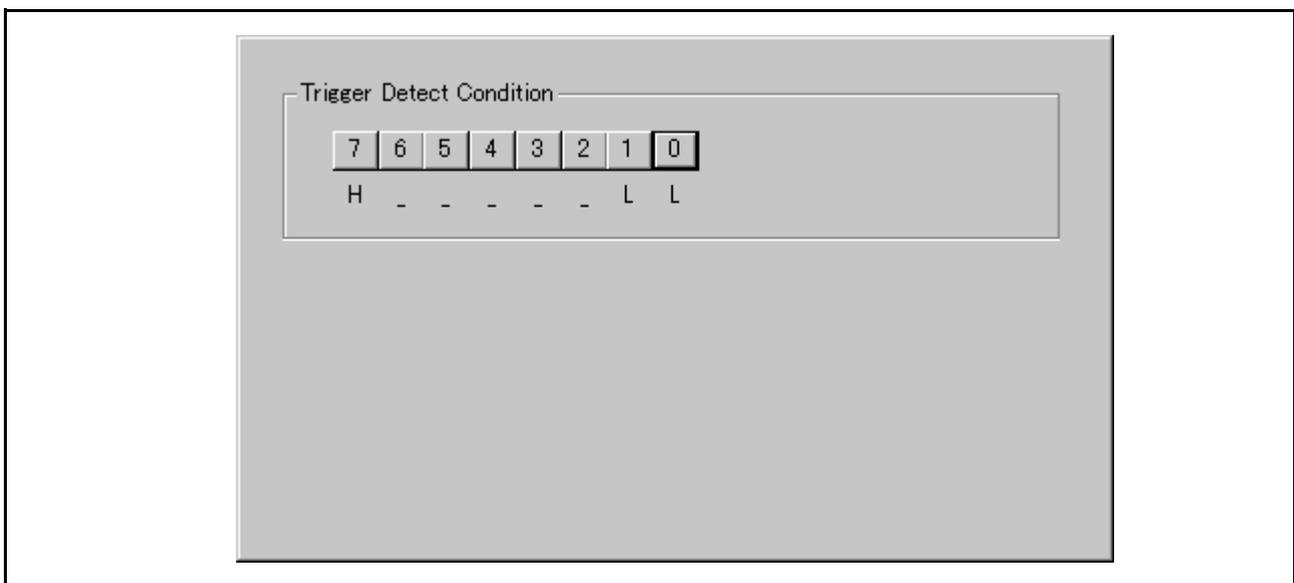
- 选择 [BIT SYMBOL] (位符号) 时跟踪位存取。



- 选择 [INTERRUPT]（中断）时
跟踪中断发生或终止。



- 选择 [TRIGGER]（触发）时
跟踪来自外部跟踪电缆的信号输入状态。

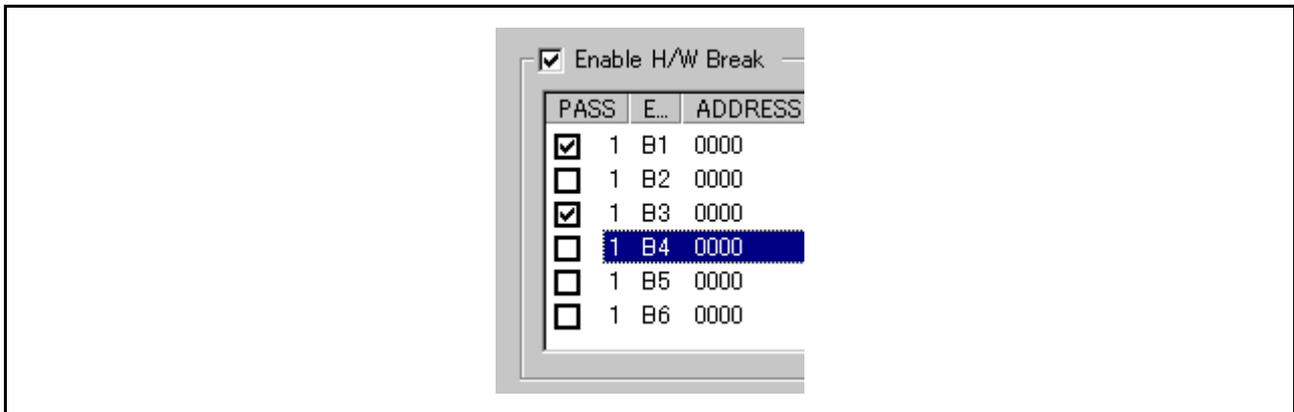


7.10.2 指定组合条件

要指定组合条件，请从组合条件指定区域选择所需条件。

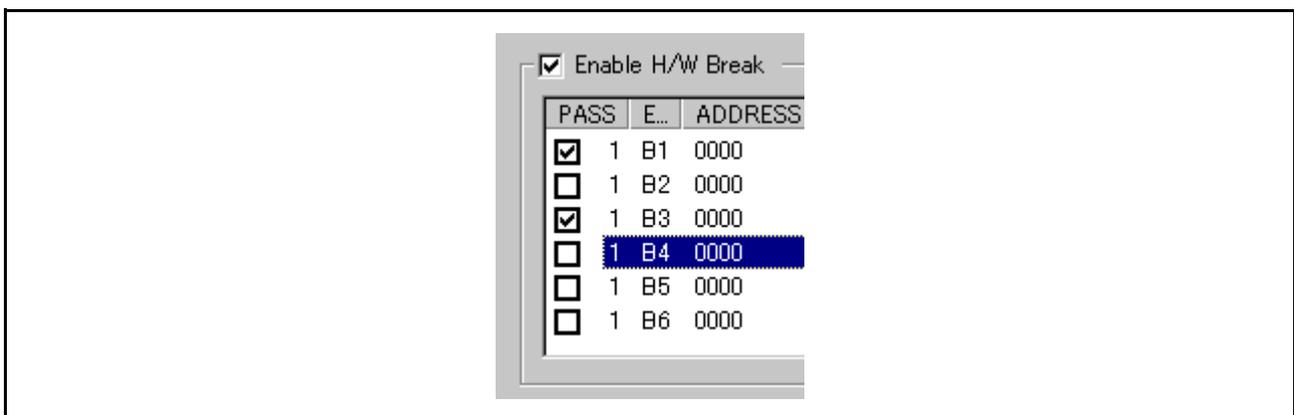
- 选择 [AND]（与）或 [OR]（或）时

在事件指定区域，可以指定所用事件和该事件的通过计数。要更改通过计数，请在要更改的事件处于选中状态时，单击该事件的通过计数值。



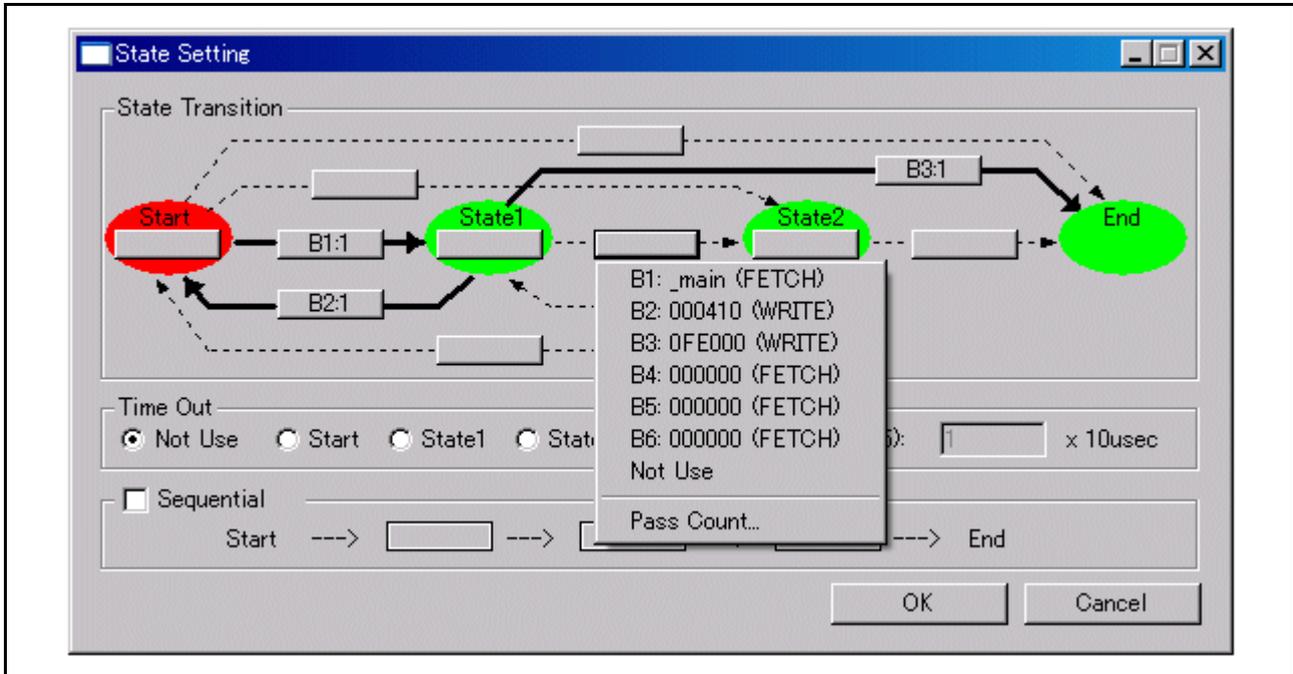
- 选择 [AND (Same Time)]（与（同时））时

在事件指定区域，可以指定所用事件。不能指定通过计数。



- 选择 [State Transition] (状态转换) 时

单击 [Details...] (详细信息...) 按钮, 此时会出现如下对话框。可以通过状态转换图指定, 也可以通过 [Sequential] (顺序) 指定。如果任一事件的内容发生变化, 则事件的标题栏上会标上一个星号 (*)。在仿真器中设置了条件后, 就不再显示星号。还可以指定各个状态的超时时间。



7.10.3 指定进程 ID

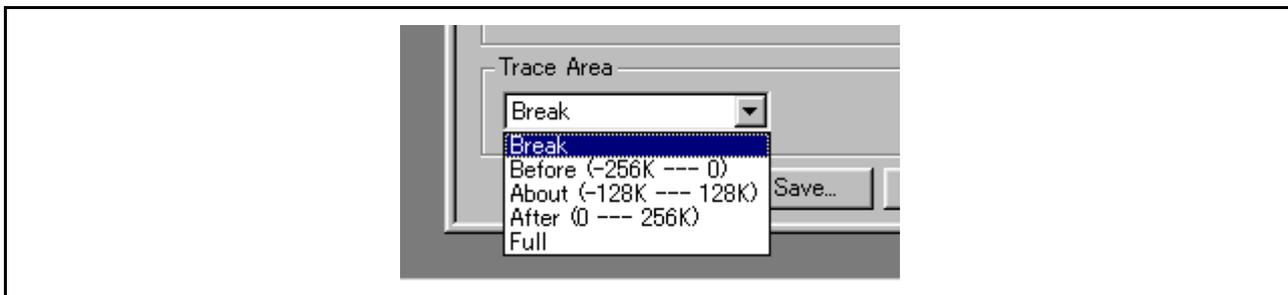
通过指定进程 ID, 可以只检测特定条件下的事件建立情况。



示例: 仅允许使用实时操作系统时出现在特定任务中的事件。

7.10.4 指定跟踪范围

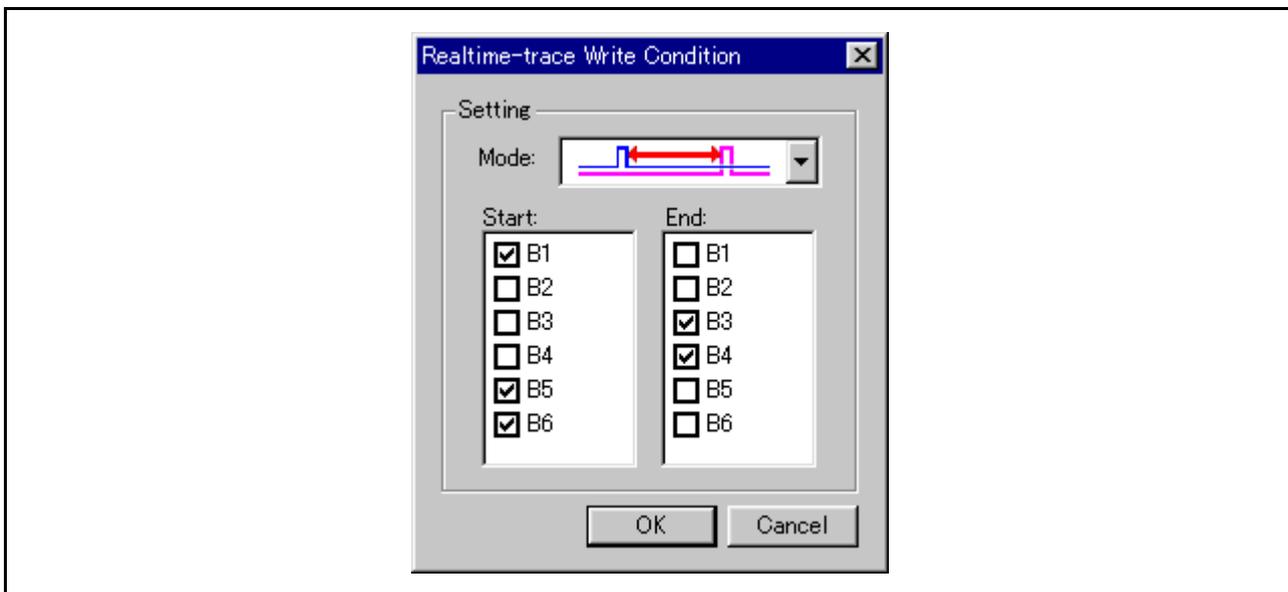
对于 PC7501 仿真调试器，可以记录相当于 256K 周期的数据。



[Break] (断点)	存储目标程序停止点之前的 256K 个周期 (-256K 到 0 周期)。
[Before] (之前)	存储通过跟踪点的点之前的 256K 个周期 (-256K 到 1 周期)。
[About] (周围)	存储跟踪点两侧的 256K 个周期 (-128K 到 128K 周期)。
[After] (之后)	存储跟踪点之后的 256K 个周期 (0 到 256K 周期) 的跟踪数据。
[Full] (完整)	存储跟踪开始之后的 256K 个周期 (-256K 到 0 周期) 的跟踪数据。

7.10.5 指定跟踪写条件

可以指定要写入跟踪存储器的周期的条件。



[Total] (全部)	写入所有周期。
[Pick up] (提取)	只写入指定条件为真的周期。
[Exclude] (排除)	只写入指定条件不为真的周期。

同时还支持以下三种写模式。

	仅建立指定事件的周期
	从建立指定事件到未建立指定事件的周期
	从建立开始事件到建立结束事件的周期

7.10.6 命令按钮

此窗口中的按钮具有以下含义。

按钮	功能
[Reset] (复位)	删除窗口中显示的内容，并从设置了内容的仿真器装入内容。
[Save...] (保存...)	将窗口中设置的内容保存到文件中。
[Load...] (装入...)	从保存了事件信息的文件装入事件信息。
[Set] (设置)	将窗口中设置的内容发送到仿真器。
[Close] (关闭)	关闭窗口。

7.10.7 指定事件 (取指令)

为取指令设置事件的方法与为 [H/W Break Point Setting] (硬件断点设置) 窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.5 指定事件 (取指令)”。

7.10.8 指定事件 (存储器存取)

为存储器存取设置事件的方法与为 [H/W Break Point Setting] (硬件断点设置) 窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.6 指定事件 (存储器存取)”。

7.10.9 指定事件 (位存取)

为位存取设置事件的方法与为 [H/W Break Point Setting] (硬件断点设置) 窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.7 指定事件 (位存取)”。

7.10.10 指定事件 (中断)

为位存取设置事件的方法与为 [H/W Break Point Setting] (硬件断点设置) 窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.8 指定事件 (中断)”。

7.10.11 指定事件 (外部触发信号)

为位存取设置事件的方法与为 [H/W Break Point Setting] (硬件断点设置) 窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.9 指定事件 (外部触发信号)”。

7.10.12 指定事件组合条件

设置事件组合的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.10 指定事件组合条件”。

7.10.13 指定进程 ID

设置事件组合的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.11 指定进程 ID”。

7.10.14 指定写条件

可以指定跟踪数据写条件。

可以指定以下写条件：

1. 无限制的写条件（默认）
2. 从建立开始事件到建立结束事件的周期
3. 仅建立开始事件的周期
4. 从建立开始事件到取消建立开始事件的周期
5. 从建立开始事件到建立结束事件的周期之外
6. 建立开始事件的周期之外
7. 从建立开始事件到取消建立开始事件的周期之外

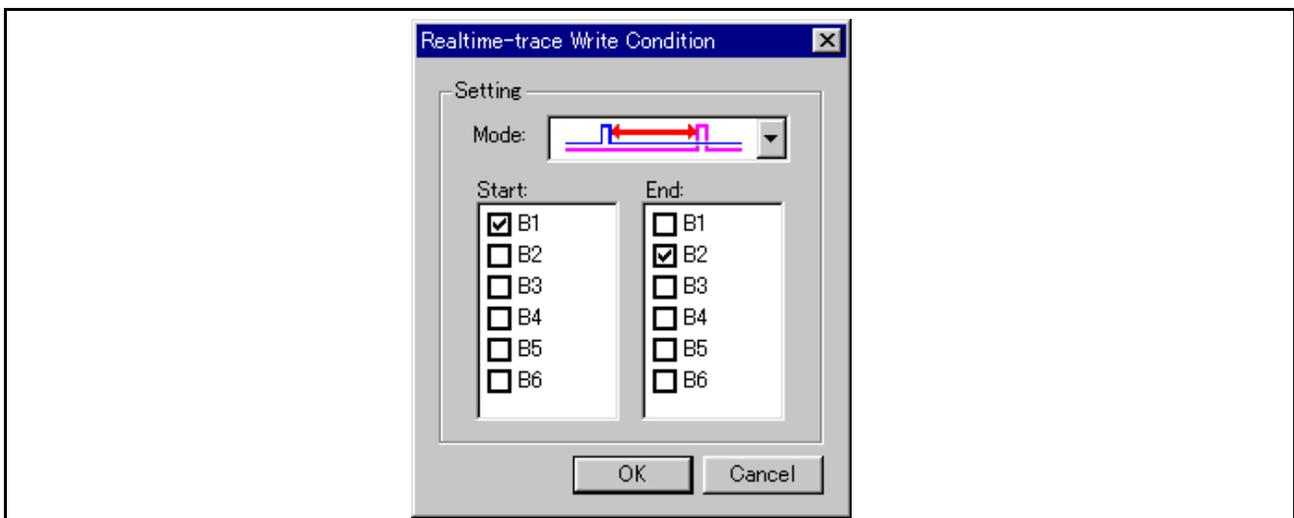
若要指定条件 1，请从窗口中“Write Condition”（写条件）项目的列表框中选择“Total”（全部）。



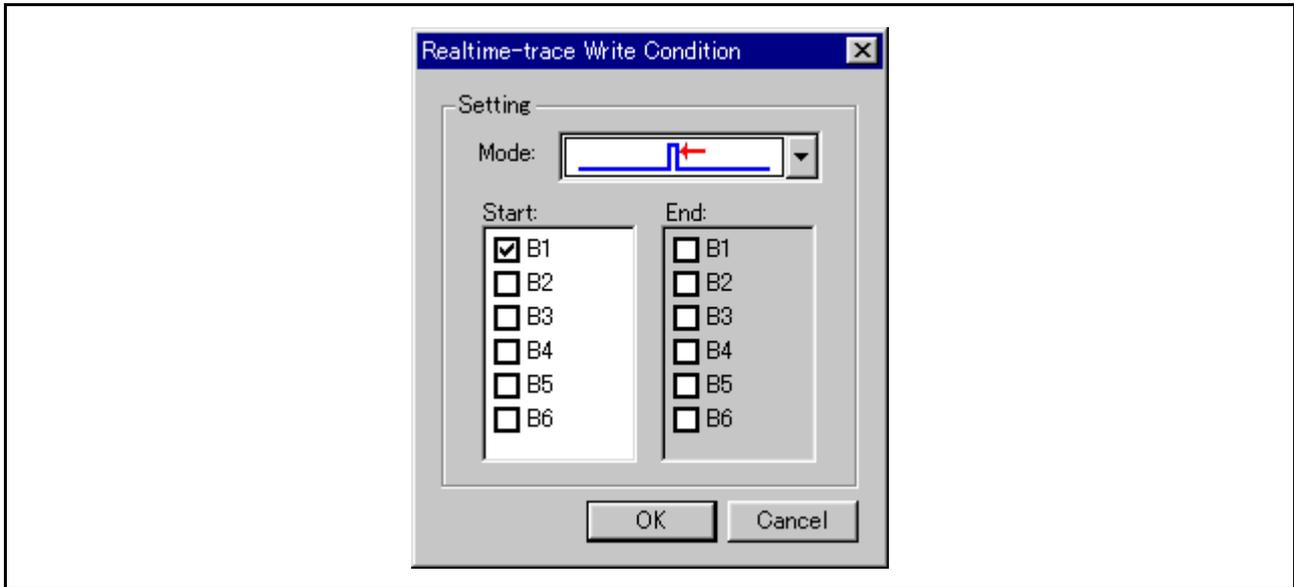
若要指定条件 2 到 4，请选择“Pick Up”（提取）并单击“Detail...”（详细信息…）按钮，以打开“Realtime-trace Write Condition”（实时跟踪写条件）对话框。



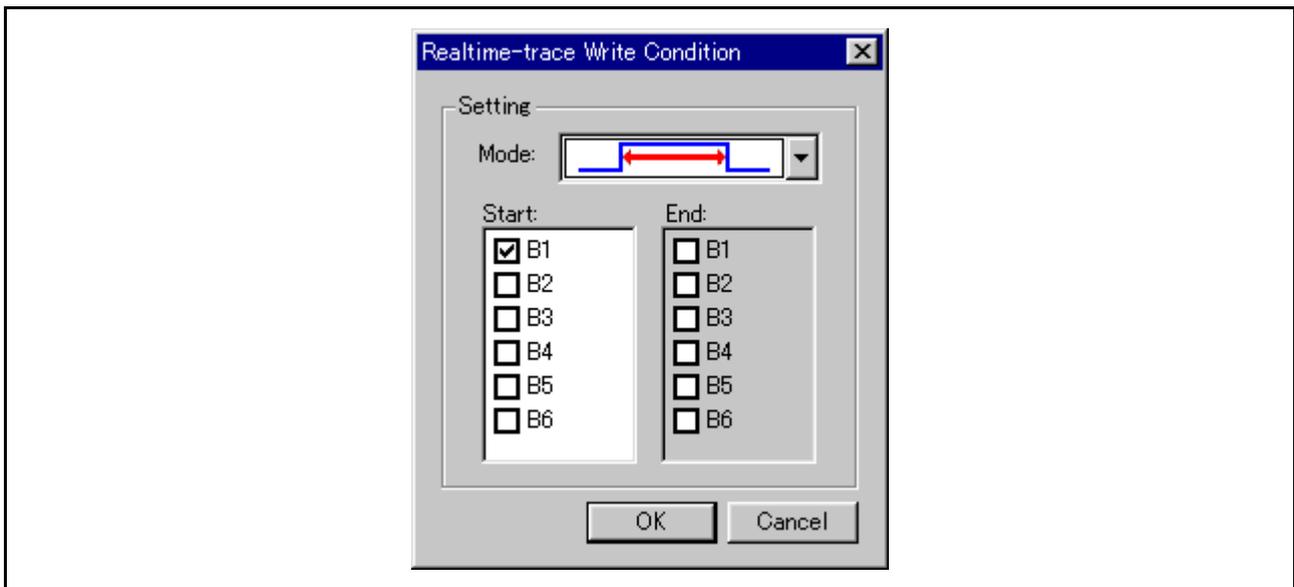
- 对于条件 2，请选择如下模式并设置开始和结束事件。



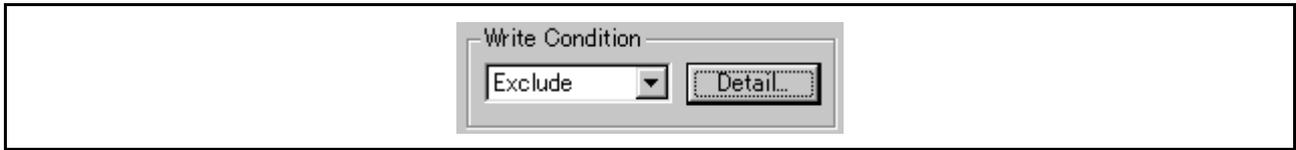
- 对于条件 3，请选择如下模式并设置开始事件。



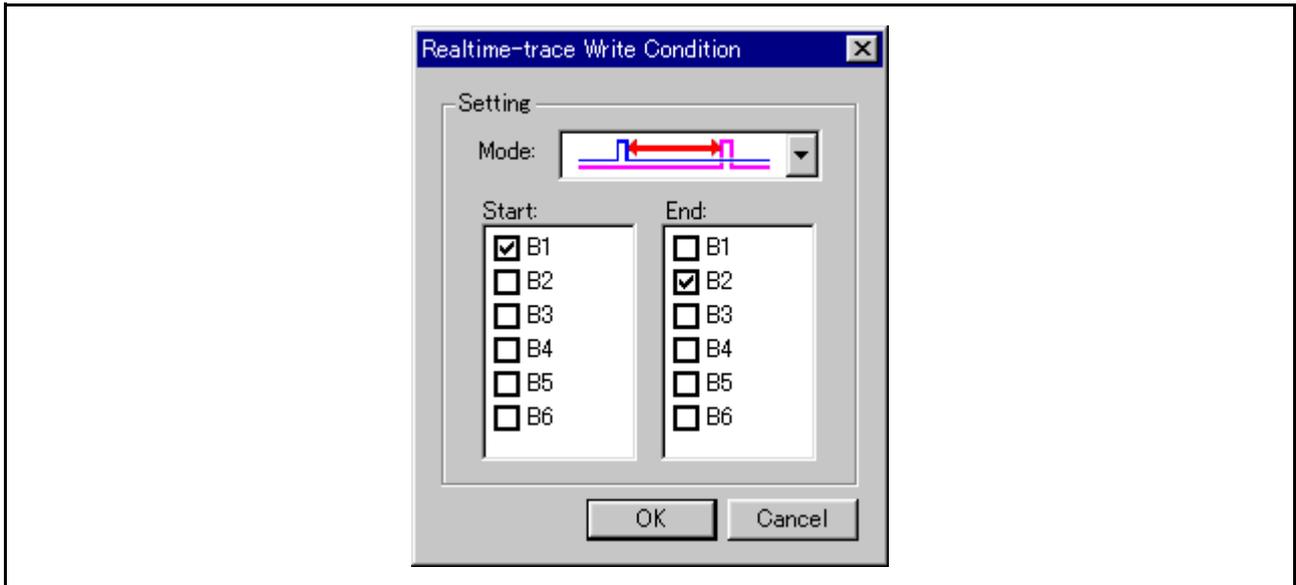
- 对于条件 4，请选择如下模式并设置开始事件。



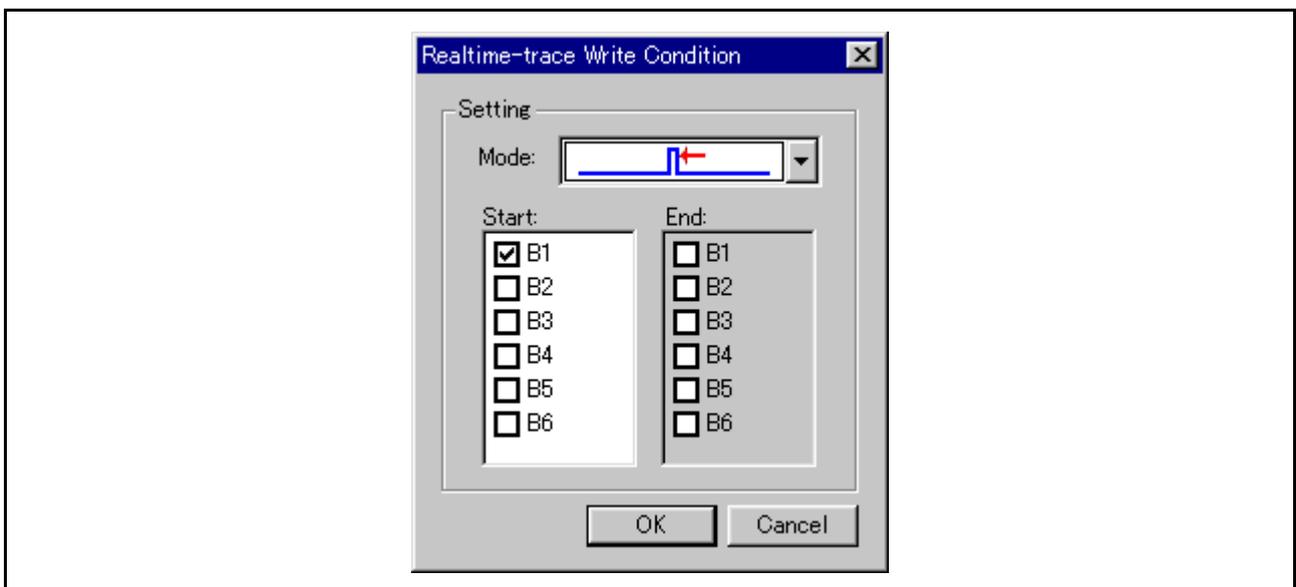
类似地，当指定条件 5 到 7 时，请选择 “Exclude”（排除）并单击 “Detail...”（详细信息…）按钮，以打开 [Realtime-trace Write Condition]（实时跟踪写条件）对话框。



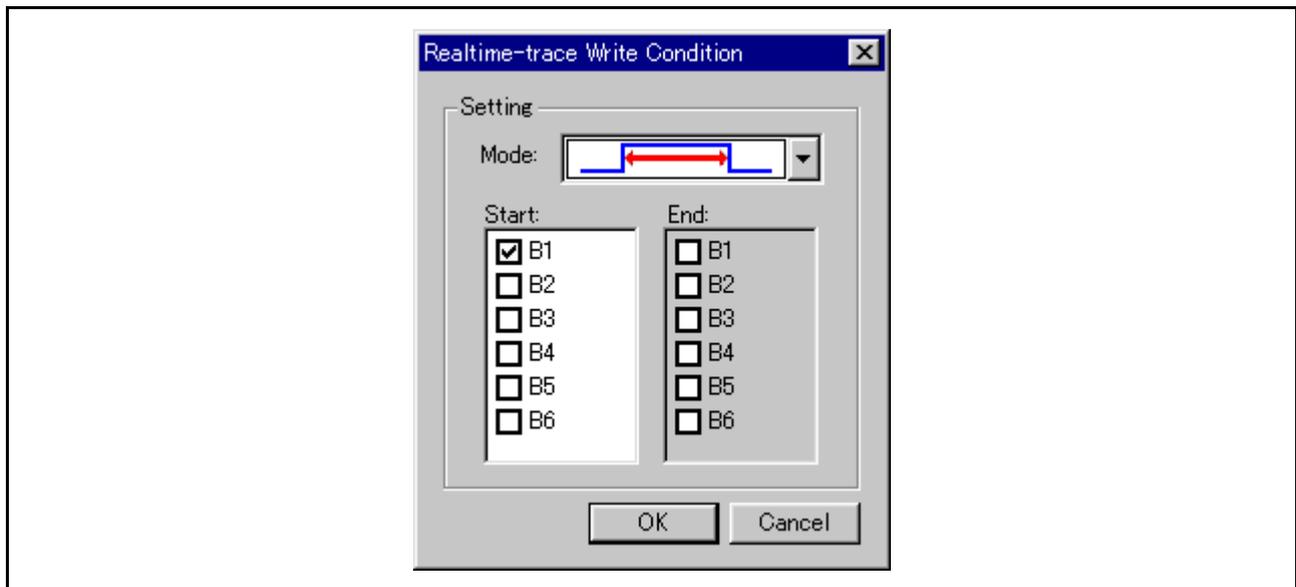
- 对于条件 5，请选择如下模式并设置开始和结束事件。



- 对于条件 6，请选择如下模式并设置开始事件。



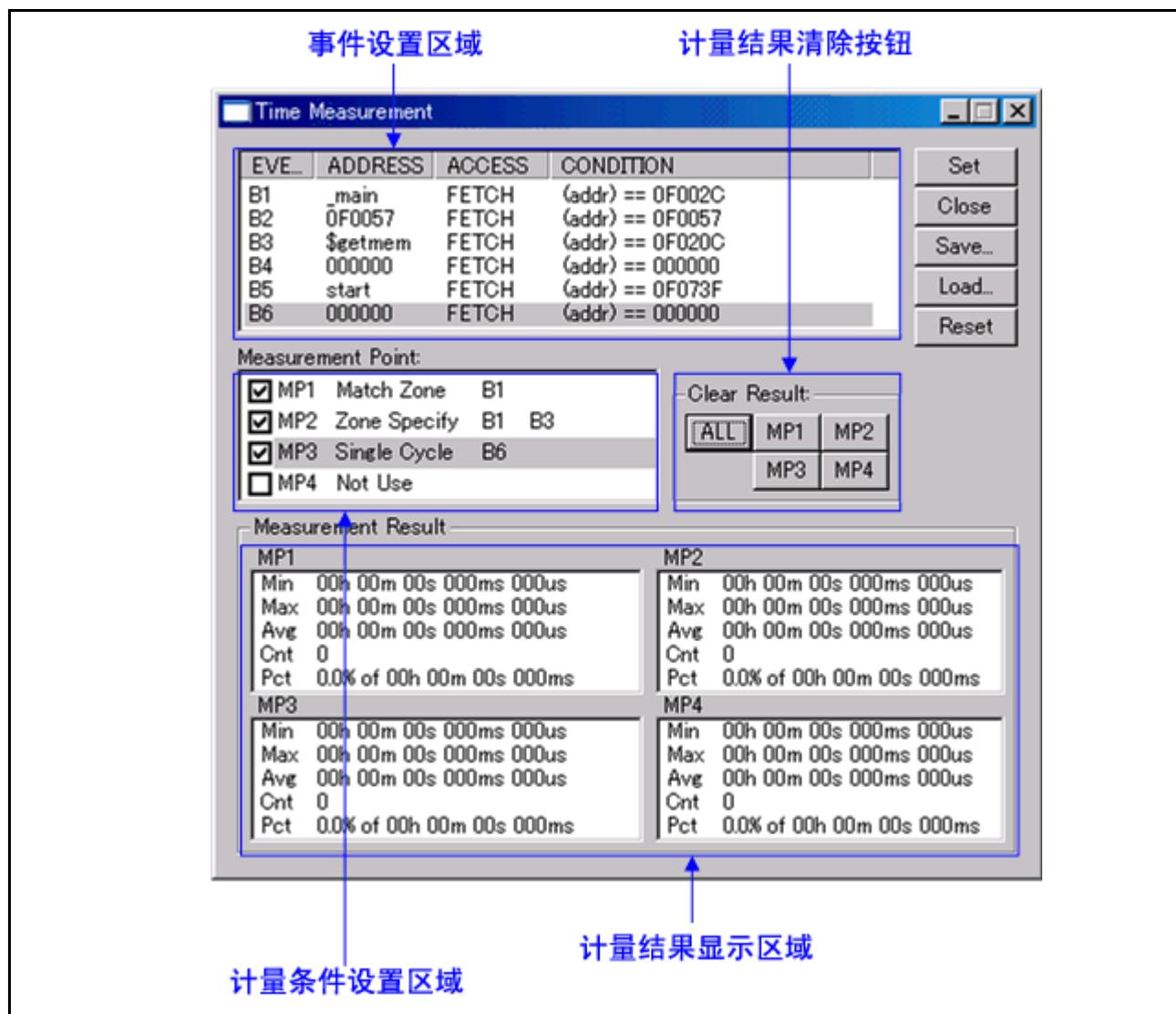
- 对于条件 7，请选择如下模式并设置开始事件。



7.11 [Time Measurement] (时间计量) 窗口

[Time Measurement] (时间计量) 窗口显示最短 / 最长 / 平均执行时间和任何计量点的计量计数。最多可以同时计量 4 个计量点的执行时间。

可以为计量条件指定事件，方法与在 [Trace Point Setting] (跟踪点设置) 窗口中指定事件的方法相同。



- 如果事件内容发生变化，则事件的标题栏上会标上一个星号 (*)。设置仿真器后不再显示星号 (*)。

注意事项

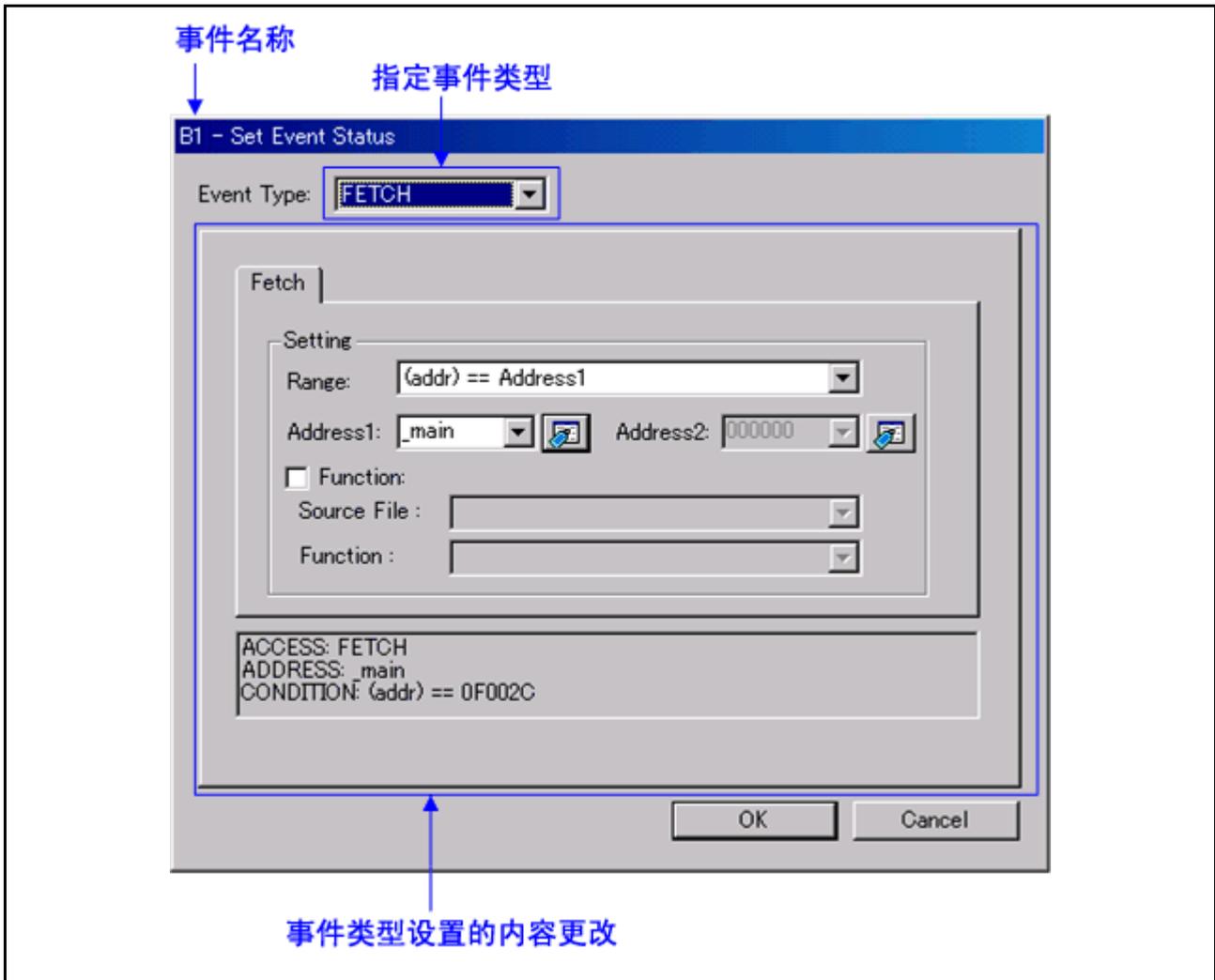
- [Trace Point Setting] (跟踪点设置) 窗口和 [Time Measure] (时间计量) 窗口使用相同的仿真器资源。如果在 [Time Measure] (时间计量) 窗口中修改了事件设置，则 [Trace Point Setting] (跟踪点设置) 窗口的设置也会被修改。

7.11.1 指定时间计量事件

下列事件可以指定为计量事件。

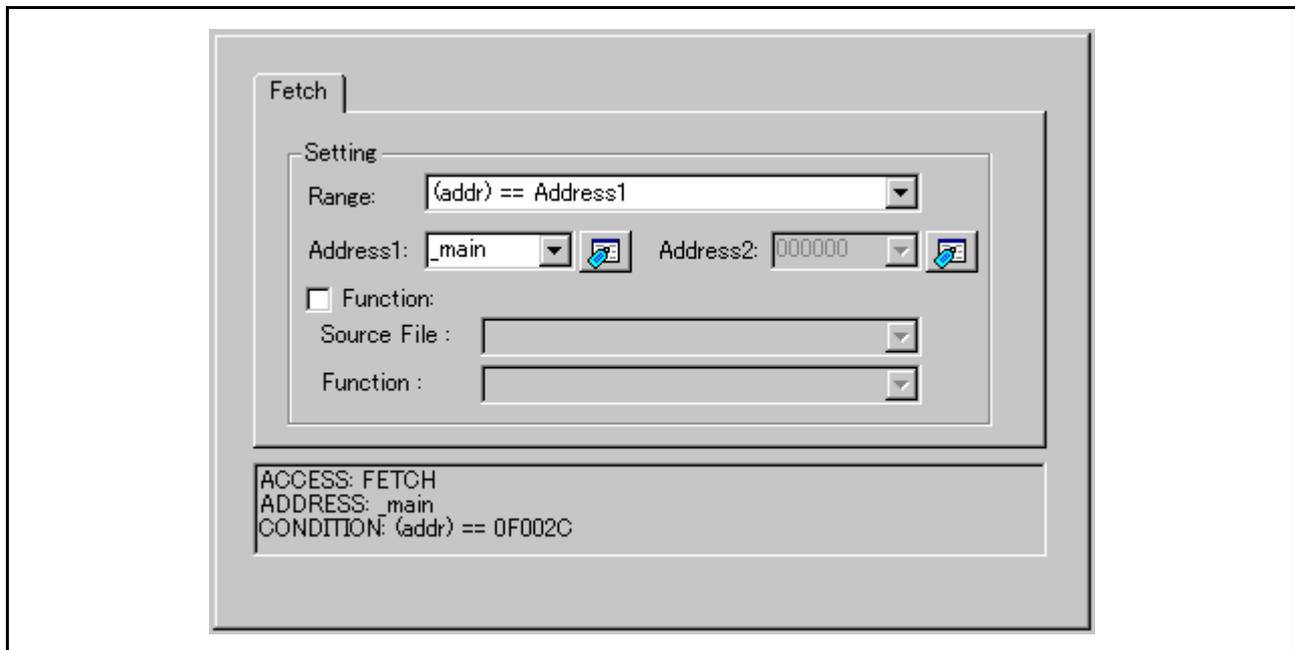
Fetch（取）、Memory Access（存储器存取）、Bit Access（位存取）、Interrupt（中断）、External Trigger（外部触发）

若要设置事件，请从 [Time Measurement]（时间计量）窗口的事件设置区双击以选择要设置的事件。此时会打开如下对话框。

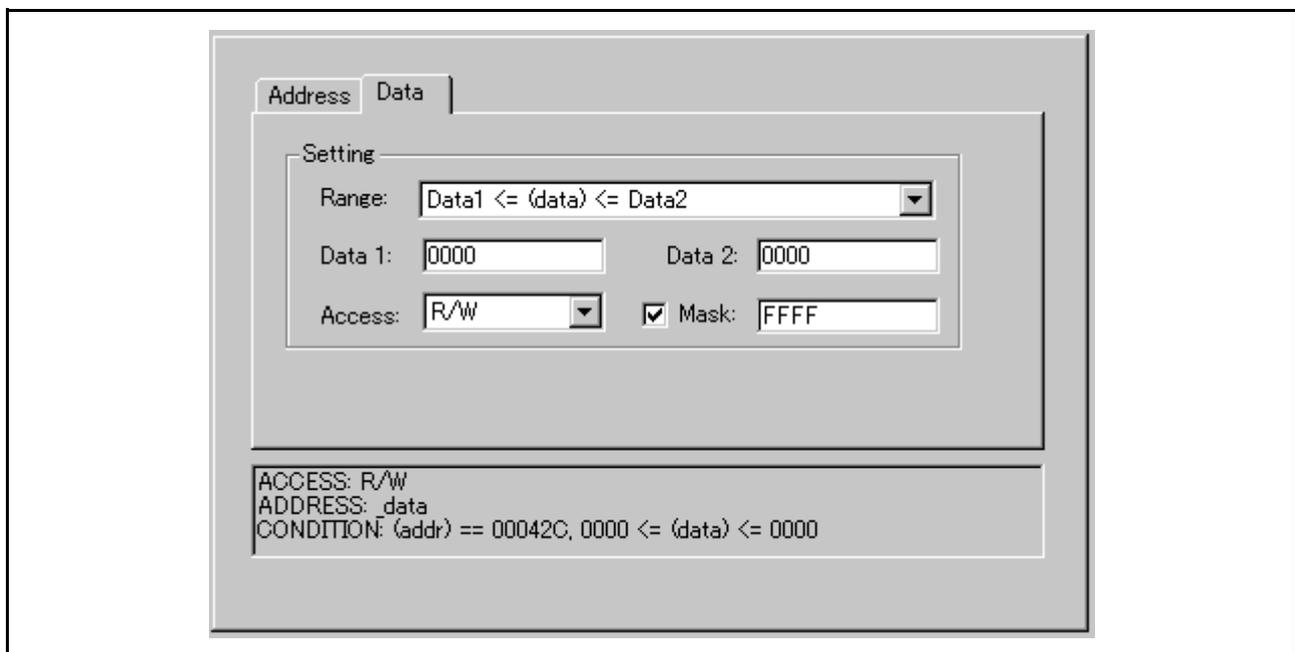


通过在此对话框中指定 [Event Type]（事件类型），可以设置以下事件。

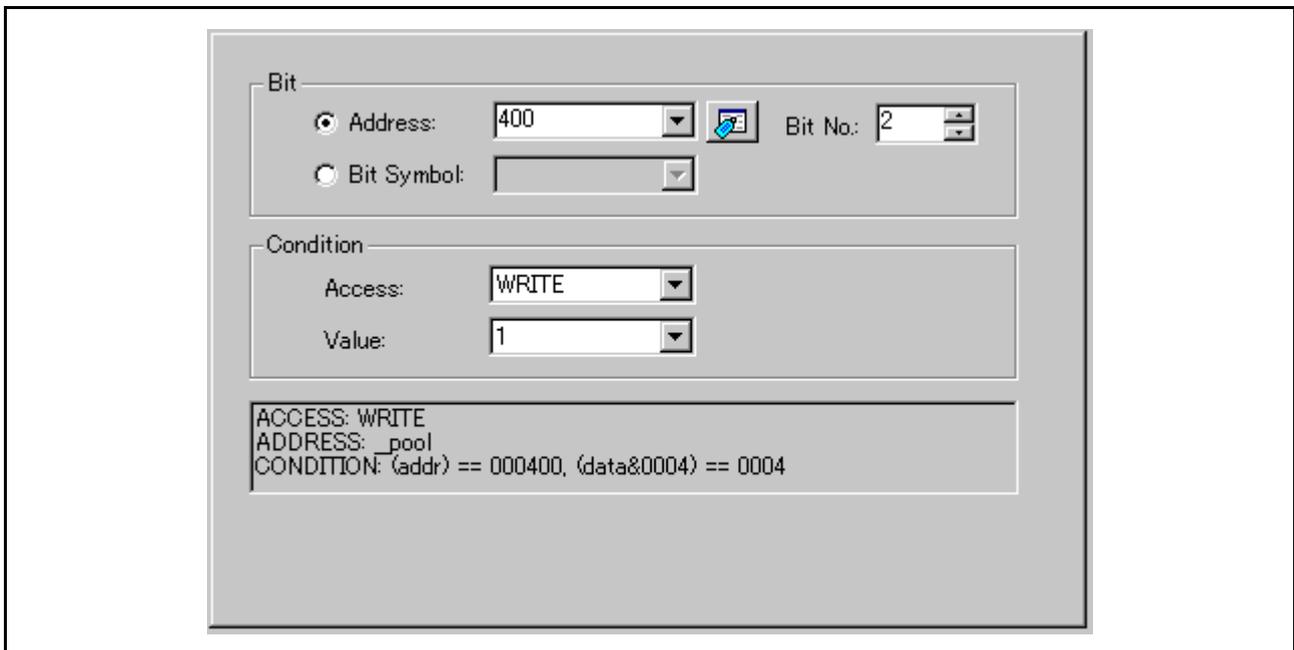
- 选择 [FETCH]（取）时
计量取指令。



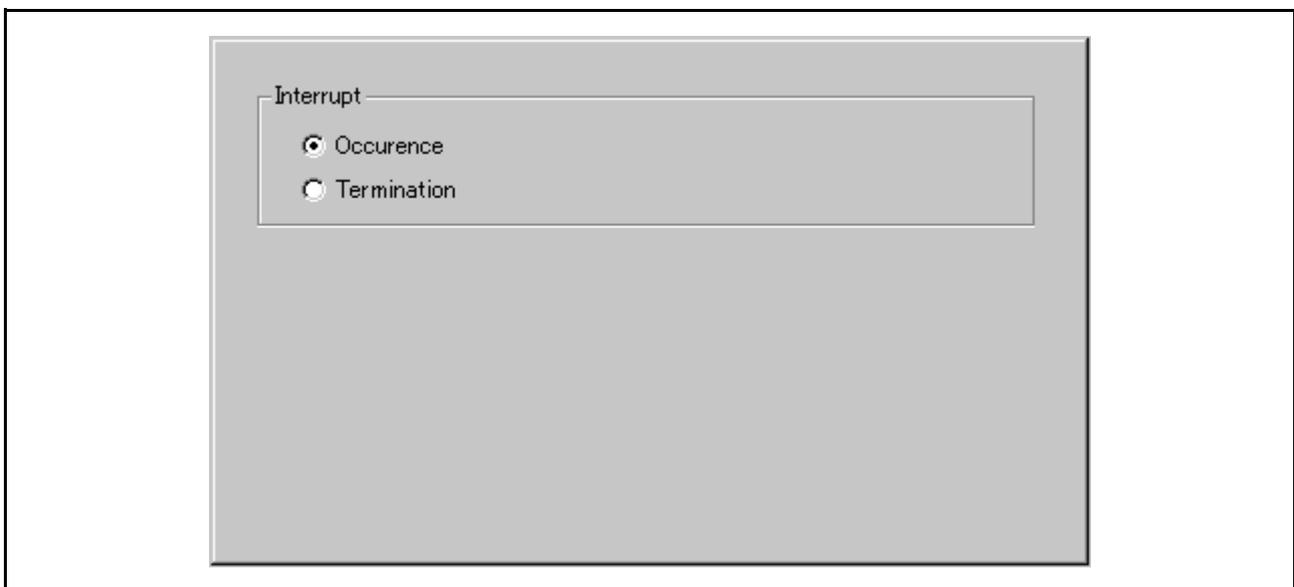
- 选择 [DATA ACCESS]（数据存取）时
计量存储器存取。



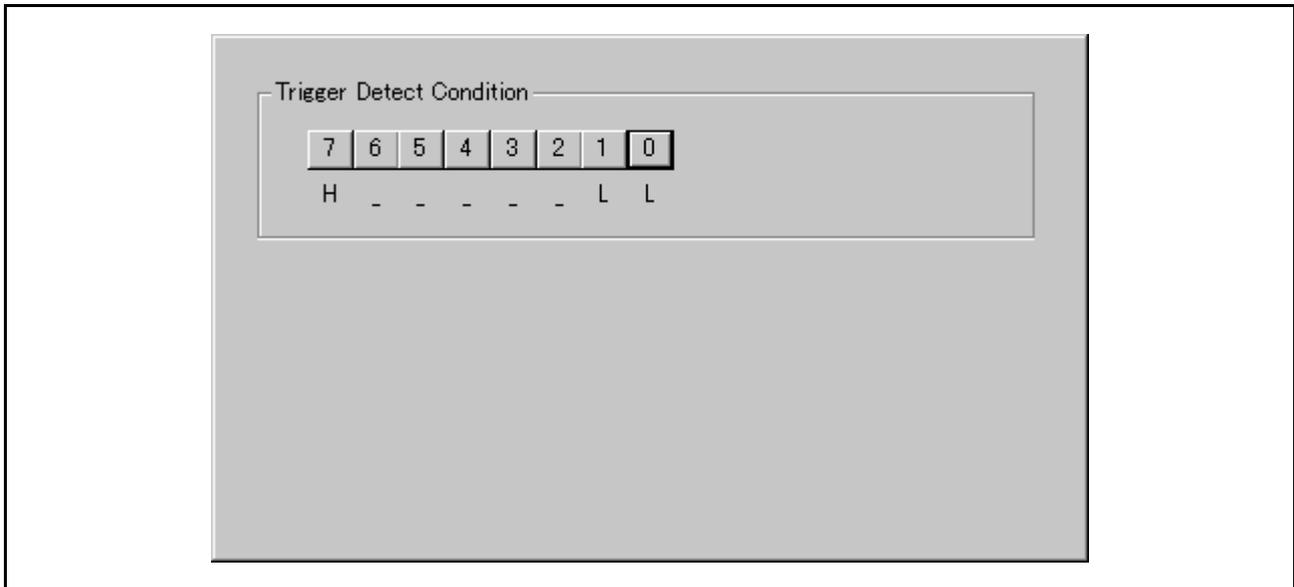
- 选择 [BIT SYMBOL]（位符号）时
计量位存取。



- 选择 [INTERRUPT]（中断）时
计量中断发生或终止。

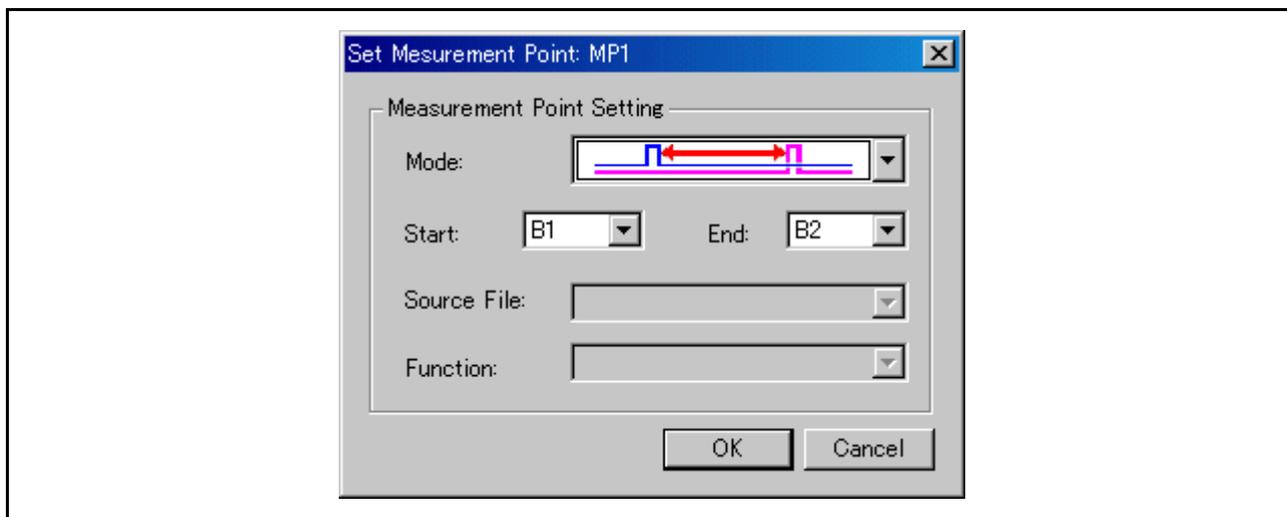


- 选择 [TRIGGER]（触发）时
计量来自外部跟踪电缆的信号输入状态。



7.11.2 时间计量条件

对于时间计量条件，可以为每个计量间隔指定以下内容。



	计量从建立开始事件到建立结束事件的间隔时间。
	计量从建立一个事件到建立下一个事件的时间。
	计量从建立一个事件到未建立该事件的时间。
	计量函数的执行时间。分别为开始事件和结束事件自动注册函数的起始地址和结束地址。 计量结果包括从指定函数内部调用的其他函数的执行时间。
	计量函数的执行时间。分别为开始事件和结束事件自动注册函数的起始地址和结束地址。计量结果不包括从指定函数内部调用的其他函数的执行时间。

7.11.3 命令按钮

此窗口中的按钮具有以下含义。

按钮	功能
[Reset] (复位)	删除窗口中显示的内容，并从设置了内容的仿真器装入内容。
[Save...] (保存...)	将窗口中设置的内容保存到文件中。
[Load...] (装入...)	从保存了事件信息的文件装入事件信息。
[Set] (设置)	将窗口中设置的内容发送到仿真器。
[Close] (关闭)	关闭窗口。

7.11.4 指定事件（取指令）

为取指令设置事件的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.5 指定事件（取指令）”。

7.11.5 指定事件（存储器存取）

为存储器存取设置事件的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.6 指定事件（存储器存取）”。

7.11.6 指定事件（位存取）

为位存取设置事件的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.7 指定事件（位存取）”。

7.11.7 指定事件（中断）

为位存取设置事件的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.8 指定事件（中断）”。

7.11.8 指定事件（外部触发信号）

为位存取设置事件的方法与为 [H/W Break Point Setting]（硬件断点设置）窗口设置事件的方法相同。有关设置的详细信息，请参阅“7.7.9 指定事件（外部触发信号）”。

7.11.9 设置计量条件

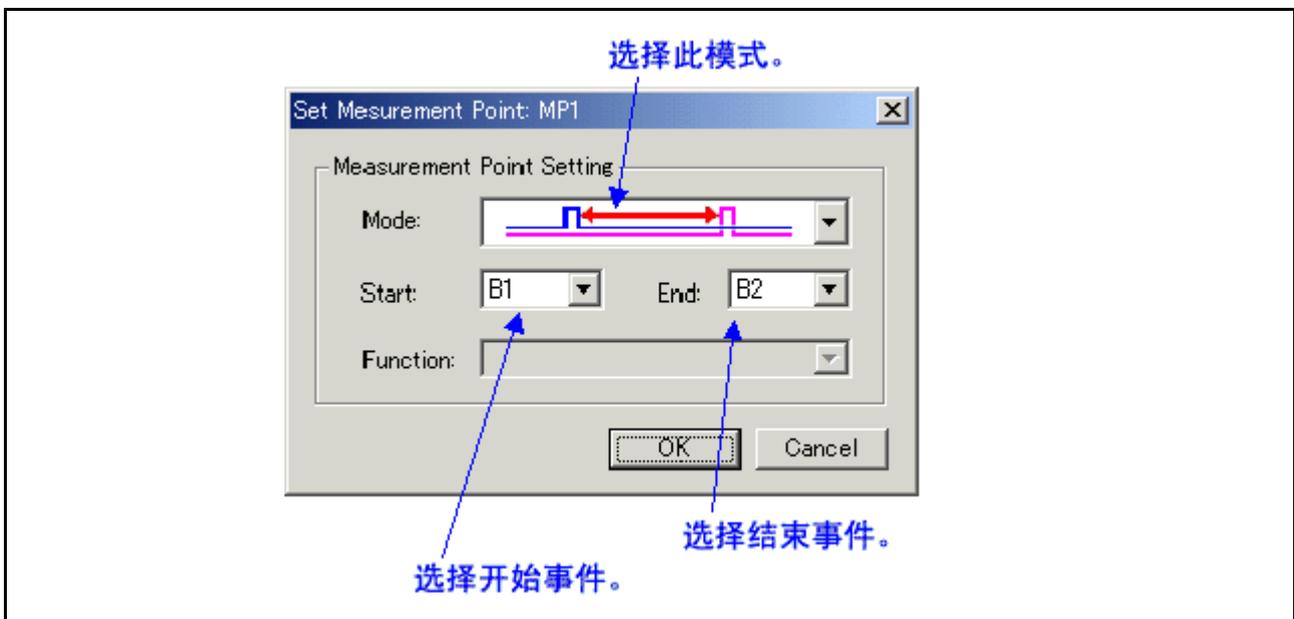
对于调试器，可以指定以下计量条件。

- 计量事件之间的执行时间。
- 计量发生间隔事件的周期。
- 计量建立事件的时间。

通过指定函数名称，可以计量该函数的执行时间。最多可以指定四个计量间隔。若要指定计量间隔，请单击 [Interval Time Measure]（间隔时间计量）窗口的 [Measurement Point]（计量点）组中的任何行 (MP1-MP4)。此时会打开指定计量条件的对话框。

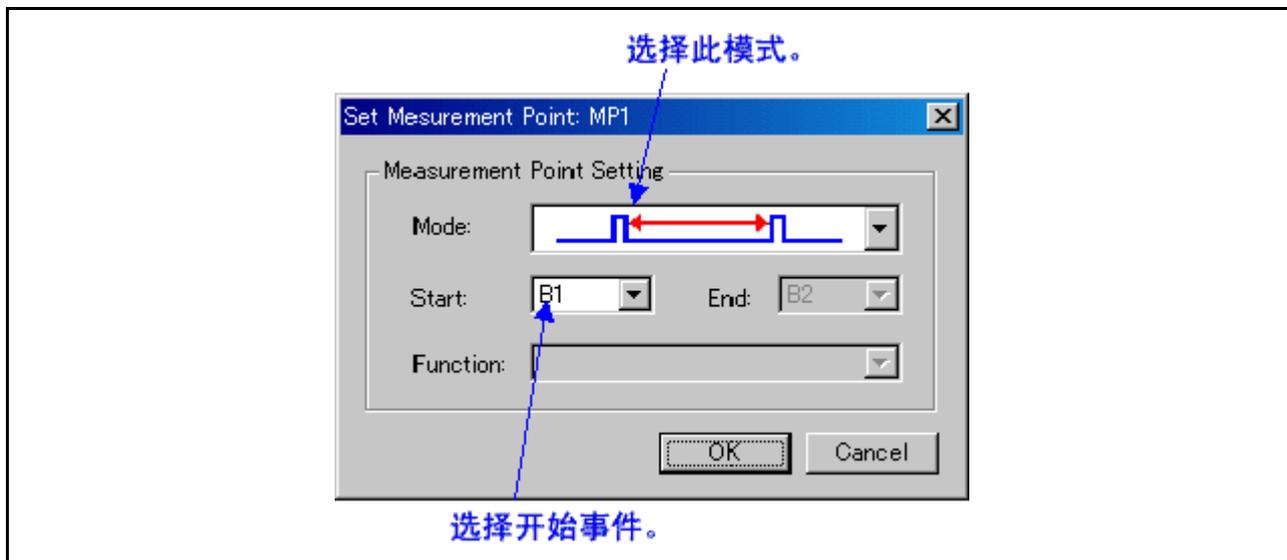
7.11.9.1 计量事件之间的执行时间

1. 设置计量事件（计量开始事件和计量结束事件）。
2. 在 [Measurement Condition Designation]（计量条件指定）对话框中指定以下内容。



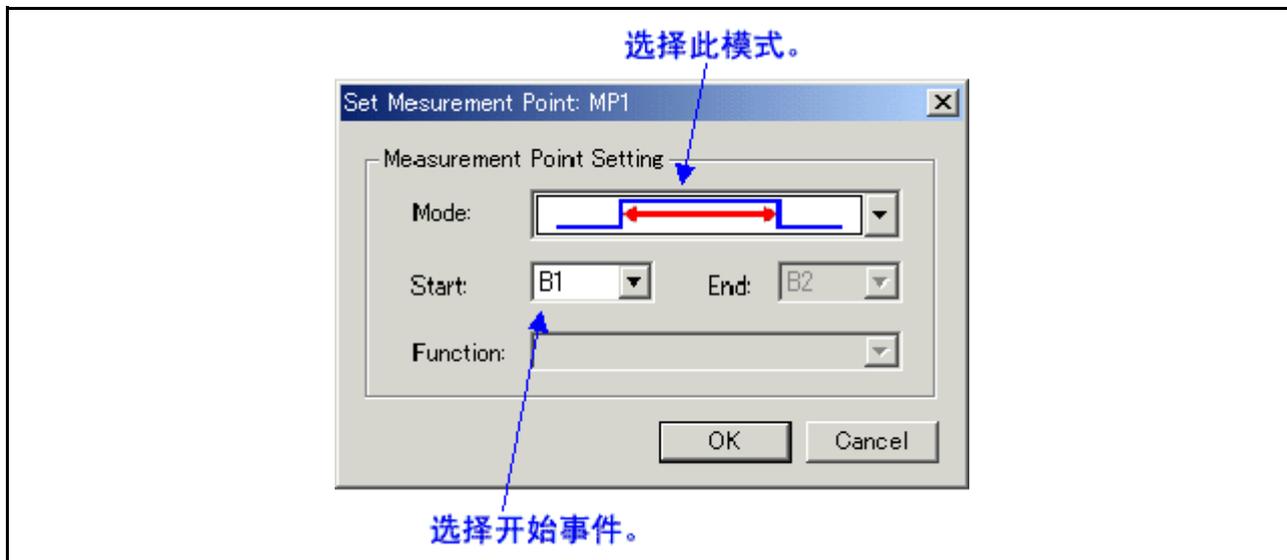
7.11.9.2 计量事件发生周期

1. 设置计量事件（仅计量开始事件）。
2. 在 [Measurement Condition Designation]（计量条件指定）对话框中指定以下内容。



7.11.9.3 计量事件建立时间

1. 设置计量事件（仅计量开始事件）。
2. 在 [Measurement Condition Designation]（计量条件指定）对话框中指定以下内容。



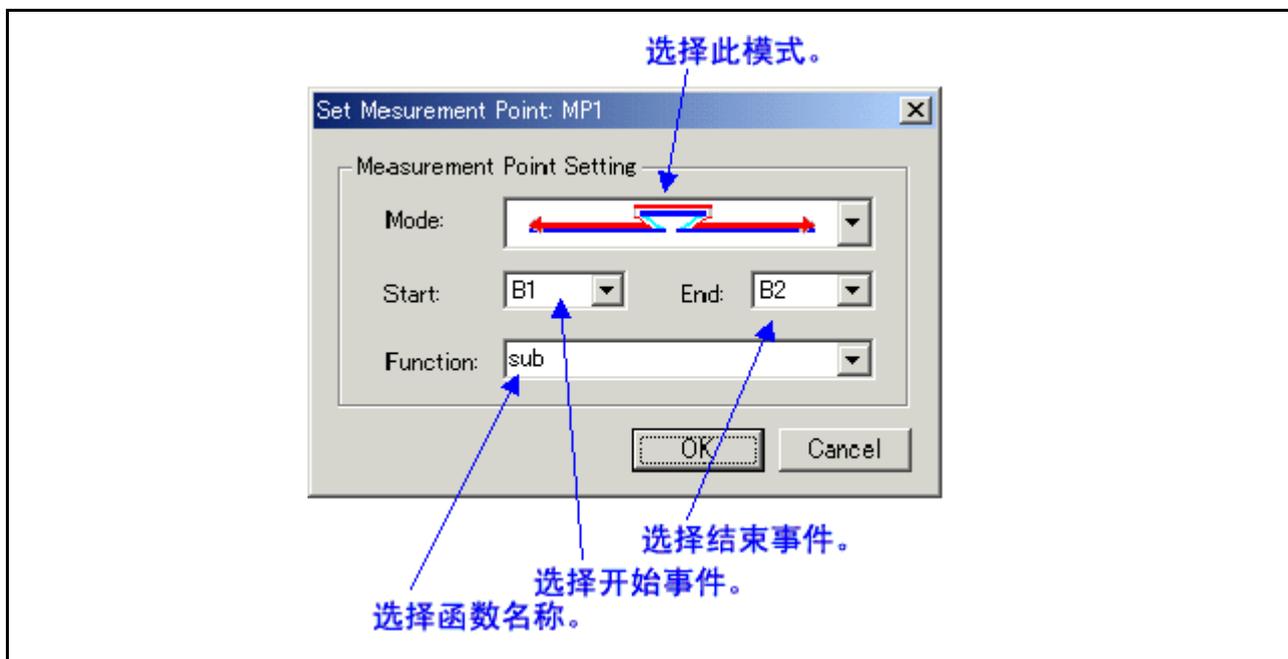
7.11.9.4 计量指定函数的执行时间 (1)

计量指定函数的执行时间。

在此模式下，函数顶部地址自动注册到计量开始事件，函数结束地址自动注册到计量结束地址。

计量结果包含指定函数内部调用的其他函数的执行时间。

在 [Measurement Condition Designation]（计量条件指定）对话框中指定以下内容。



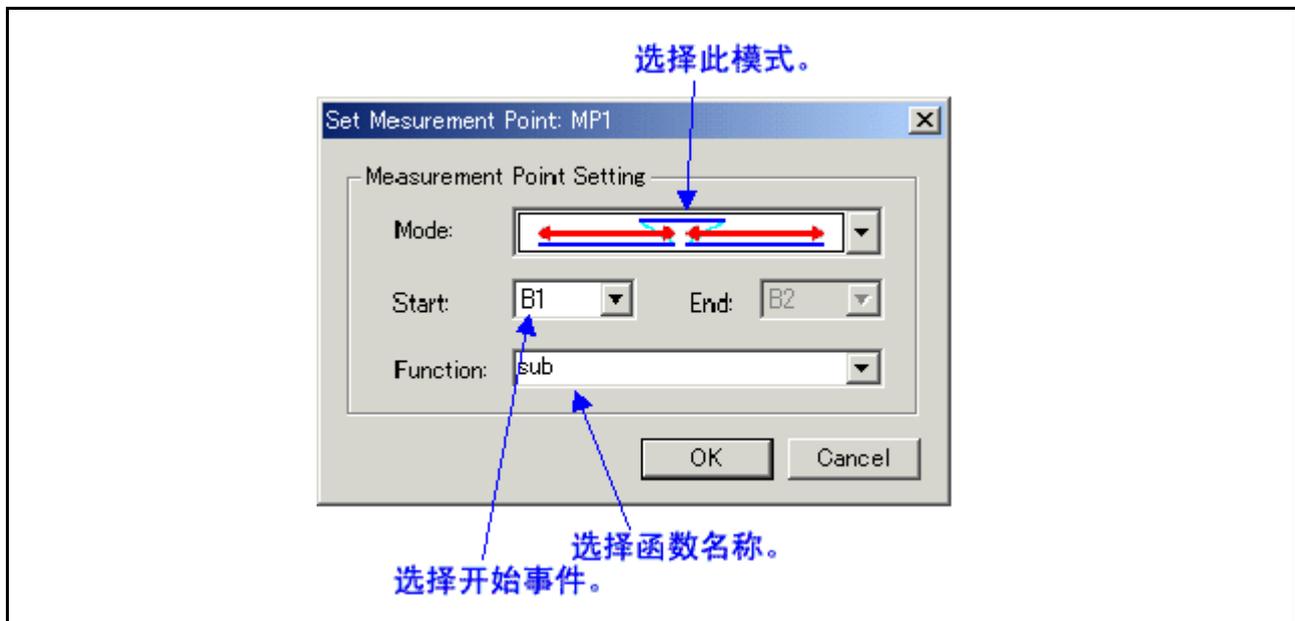
7.11.9.5 计量指定函数的执行时间 (2)

计量指定函数的执行时间。

在此模式下，函数地址范围自动注册到计量开始事件。

计量结果不包含指定函数内部调用的其他函数的执行时间。

在 [Measurement Condition Designation]（计量条件指定）对话框中指定以下内容。

**注意事项**

当目标函数调用其他函数时，第一个计量在第一次调用另一个函数时结束，第二个计量在从第一个计量返回到目标函数时开始。

7.12 [Trace] (跟踪) 窗口

[Trace] (跟踪) 窗口用于显示实时跟踪计量的结果。

- “Bus mode” (总线模式)

该模式允许您按周期检查总线信息。显示内容取决于使用的 MCU 和模拟器系统。除总线信息外，该模式还允许以组合形式显示反汇编、源行或数据存取信息。

- “Disassemble mode” (反汇编模式)

该模式允许您检查执行的指令。除反汇编信息外，该模式还允许以组合形式显示源行或数据存取信息。

- “Data access mode” (数据存取模式)

该模式允许您检查数据读/写周期。除数据存取信息外，该模式还允许以组合形式显示源行信息。

- “Source mode” (源模式)

该模式允许您检查源程序中的程序执行路径。

跟踪计量结束时，会显示计量结果。当跟踪计量再启动时，会清除窗口显示内容。

可以在 [Trace Point Setting] (跟踪点设置) 窗口中改变跟踪计量的范围。有关此窗口的详细信息，请参阅“7.10 [Trace Point Setting] (跟踪点设置) 窗口”。使用默认设置时，会记录程序停止之前的跟踪信息。

7.12.2 配置“Disassemble Mode”（反汇编模式）

当选择“disassemble mode”（反汇编模式）而取消选择“bus mode”（总线模式）时，跟踪信息会以反汇编模式显示。反汇编模式的配置如下所示。

Cycle	Address	Obj-code	Label	Mnemonic	h' m' s: ms. us
-18960	0F0199	730BFO		MOV.W RO, -10H[FB]	00"00'00:054.427
-18957	0F019C	C91BFD		ADD.W #1H, -3H[FB]	00"00'00:054.427
-18953	0F019F	FEC1		JMP.B F0161H	00"00'00:054.427
-18948	0F0161	778BFDOA00		CMP.W #000AH, -3H[FB]	00"00'00:054.428
-18943	0F0166	7DCA39		JGE F01A1H	00"00'00:054.428
-18938	0F01A1	7DF2		EXITD	00"00'00:054.428
-18929	0F0087	F50600		JSR.W _randam_ F008EH	00"00'00:054.429
-18920	0F008E	7CF204	_randam_access	ENTER #04H	00"00'00:054.429
-18916	0F0091	FDD40B0F		JSR.A _rand F0BD4H	00"00'00:054.430
-18906	0F0BD4	75C06D4E	_rand	MOV.W #4E6DH, RO	00"00'00:054.430
-18904	0F0BD8	75C2C641		MOV.W #41C6H, R2	00"00'00:054.430
-18902	0F0BDC	754F4004		PUSH.W 0440H	00"00'00:054.430
-18898	0F0BE0	754F3E04		PUSH.W 043EH	00"00'00:054.431
-18893	0F0BE4	FE01		JMP.B FOBE6H	00"00'00:054.431
-18889	0F0BE6	FD1C0COF		JSR.A _i4mulU F0C1CH	00"00'00:054.431
-18879	0F0C1C	EC50	_i4mulU	PUSHM R1, R3	00"00'00:054.432
-18875	0F0C1E	75B107		MOV.W 7H[SP], R1	00"00'00:054.432
-18870	0F0C21	7121		MULU.W R2, R1	00"00'00:054.432
-18865	0F0C23	7312		MOV.W R1, R2	00"00'00:054.433
-18863	0F0C25	75B109		MOV.W 9H[SP], R1	00"00'00:054.433
-18859	0F0C28	7101		MULU.W RO, R1	00"00'00:054.433
-18854	0F0C2A	A112		ADD.W R1, R2	00"00'00:054.433

(1) (2) (3) (4)

1. 地址显示区：显示与指令相对应的地址。双击这里可以打开搜索地址的对话框。
2. 目标代码显示区：显示指令的目标代码。
3. 标签显示区：显示与指令地址相对应的标签。双击这里可以打开搜索地址的对话框。
4. 助记符显示区：显示指令的助记符。

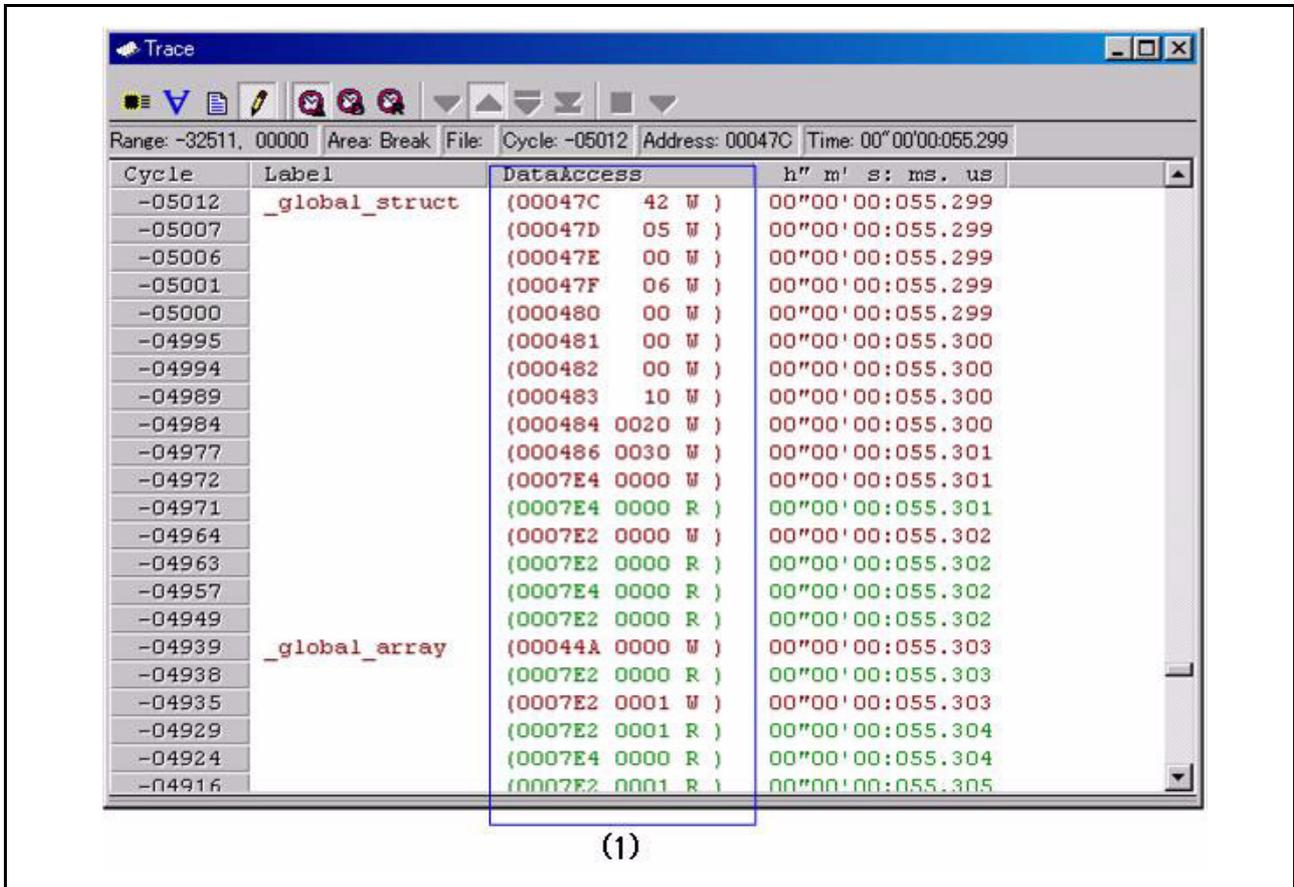
其他显示区与总线模式的显示区相同。

除反汇编信息外，该窗口还能够以组合形式显示源行或数据存取信息。在这种情况下，显示内容将与如下内容相似。

Cycle	Address	Obj-code	Label	Mnemonic	Data&access	h' m' s: ms. us
	LOCAL.C, 42:			for (i = 0; i < 10; i++) {		
-19026	0F0161	778BFDOA00		CMP.W #000AH, -3H[FB]	(0007E3 09 R)	00"00'00:054.423
					(0007E4 00 R)	
-19021	0F0166	7DCA39		JGE F01A1H		00"00'00:054.423
	LOCAL.C, 43:			char localScope_char = 'a';		
-19019	0F0169	C661FF		MOV.B #61H, -1H[FB]		00"00'00:054.423
	LOCAL.C, 44:			long localScope_long = 0;		
-19017	0F016C	D90BF9		MOV.W #0H, -7H[FB]	(0007E5 61 W)	00"00'00:054.423
-19014	0F016F	D90BFB		MOV.W #0H, -5H[FB]	(0007DF 00 W)	00"00'00:054.423

7.12.3 配置 “Data Access Mode”（数据存取模式）

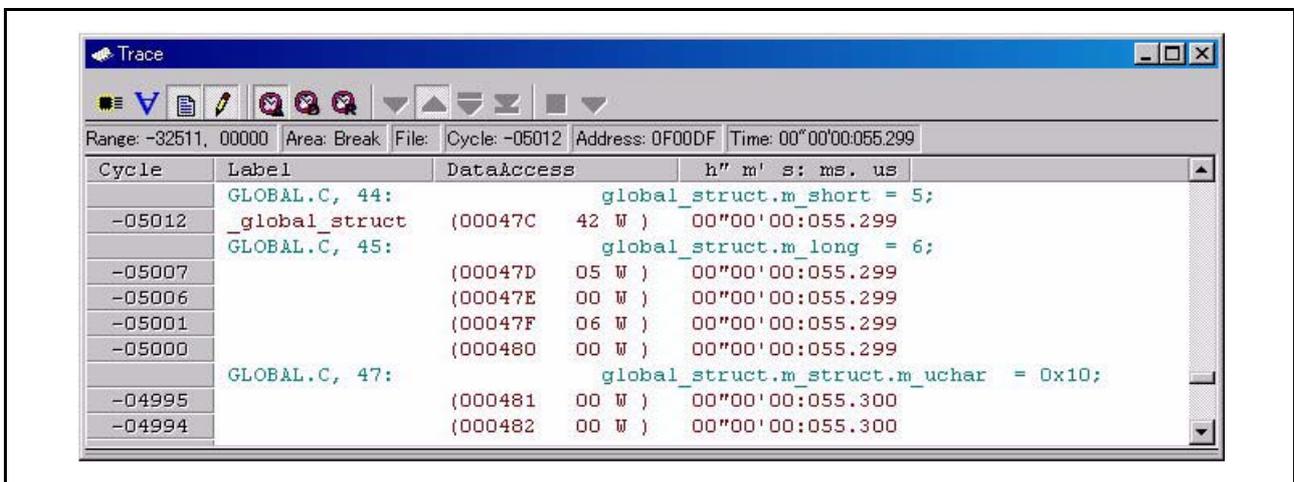
当选择 “data access mode”（数据存取模式）而取消选择 “bus mode”（总线模式）和 “disassemble mode”（反汇编模式）时，跟踪信息会以数据存取模式显示。数据存取模式的配置如下所示。



1. 数据存取显示区：显示数据存取信息。例如，如果这里显示的信息是 “000400 1234 W”，它表示数据 “1234H” 以 2 个字节的宽度写入地址 000400H。

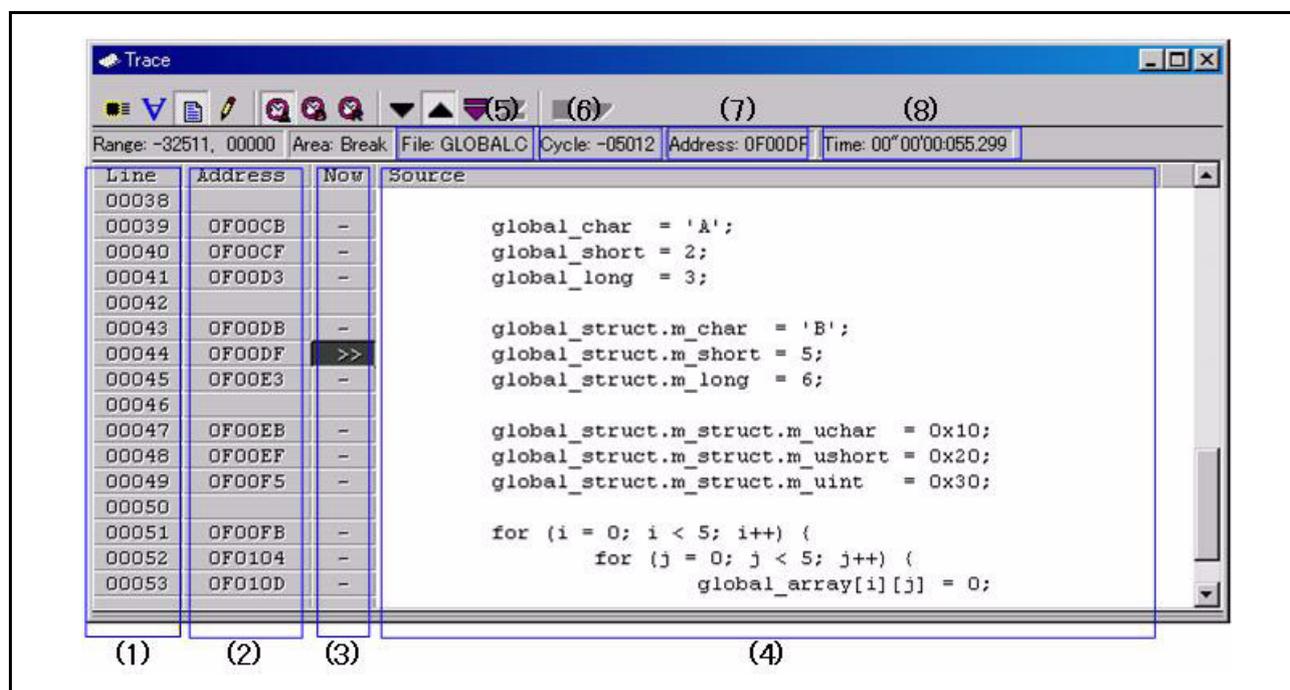
其他显示区与总线模式的显示区相同。

除数据存取信息外，该窗口还能够以组合形式显示源行信息。在这种情况下，显示内容将与如下内容相似。



7.12.4 配置“Source Mode”（源模式）

当仅选择“source mode”（源模式）时，跟踪信息会以源模式显示。源模式的配置如下所示。



1. 行号显示区：显示所示文件的行号信息。双击这里可以打开更改显示文件的对话框。
2. 地址显示区：显示与源行相对应的地址。双击这里可以打开搜索地址的对话框。
3. 引用周期显示区：显示当前引用的周期，该周期带有“>>”标志。此外，如果有与源行相对应的地址，则该地址会带有“-”标志。
4. 源显示区：显示源文件的内容。
5. 文件名：显示当前显示的源文件的文件名。
6. 引用周期：显示当前引用的周期。
7. 引用地址：显示与当前引用的周期相对应的地址。
8. 引用时间：显示与当前引用的周期相对应的的时间信息。

其他显示区与总线模式的显示区相同。

7.12.5 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[BUS] (总线)		显示“BUS mode”(总线模式)的信息。
[DIS]		显示“Disassemble mode”(反汇编模式)的信息。
[SRC]		显示“Source mode”(源模式)的信息。
[DATA] (数据)		显示“Data access mode”(数据存取模式)的信息。
[View] (视图)	[Cycle...] (周期...)	通过指定周期来更改显示的位置。
	[Address...] (地址...)	通过搜索地址来更改显示的位置。
	[Source...] (源...)	显示选择的源文件。
[Time] (时间)	[Absolute Time] (绝对时间)	按照绝对时间显示从程序开始运行的时间到现在所经过的时间。
	[Differences] (差异)	显示与相邻的前一个显示周期的时间差异。
	[Relative Time] (相对时间)	显示与当前选择的周期的相对时间。
[Trace] (跟踪)	[Forward] (正向)	将搜索方向更改为正向。
	[Backward] (反向)	将搜索方向更改为反向。
	[Step] (单步执行)	在指定方向进行单步执行。
	[Come]	将搜索指定行的执行周期。
	[Stop] (停止)	在中间停止跟踪计量并显示当前时间点的计量内容。
	[Restart] (再启动)	再启动跟踪计量。
[Layout...] (版图...)		更改当前视图的版图。
[Copy] (复制)		复制选择的行。
[Save...] (保存...)		将跟踪数据保存到文件。
[Load...] (装入...)		从文件装入跟踪数据。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

7.12.6 显示 M32C 调试器的总线信息

从左到右的内容如下：

- 地址
- 地址总线的状态
- 数据
数据总线的状态
- 总线

外部数据总线的宽度（“8b”表示 8 位数据总线，“16b”表示 16 位数据总线）

- BIU

此项显示 BIU（总线接口单元）和存储器以及 BIU 和 I/O 之间的状态。

表示法	BIU 状态
-	无存取
WAIT（等待）	正在执行 wait 指令
RBML	读操作（字节，ML 开启）
F	取存取
QC	不连续的取存取（队列缓冲器）
RWML	读操作（字，ML 开启）
INT	中断应答
RB	读操作（字节）
WB	写存取（字节）
DRB	DMA 读操作（字节）
DWB	DMA 写存取（字节）
RW	读操作（字）
WW	写存取（字）
DRW	DMA 读操作（字）
DWW	DMA 写存取（字）

- R/W

显示数据总线的状态（“R”表示读，“W”表示写，“-”表示无存取）。

- RWT

此信号显示总线周期中的有效位置（有效时为“0”）。当 RWT 为“0”时，地址、数据和 BIU 信号有效。

- CPU、OPC、OPR

此项显示 CPU 和 BIU 之间的信号。在“CPU”列中，数据显示 CPU 是否存取 BIU。在“OPR”列中，数据显示读操作数的字节大小。

表示法			BIU 状态	
CPU	OPC	OPR	操作码大小	操作数大小
-	-	-	无存取	
CPU	0	1	0 字节	1 字节
CPU	0	2	0 字节	2 字节
CPU	0	3	0 字节	3 字节
CPU	1	0	1 字节	0 字节
CPU	1	1	1 字节	1 字节
CPU	1	2	1 字节	2 字节
CPU	1	3	1 字节	3 字节
CPU	2	0	2 字节	0 字节
CPU	2	1	2 字节	1 字节
CPU	2	2	2 字节	2 字节
CPU	3	0	3 字节	0 字节
CPU	3	1	3 字节	1 字节
DMA	-	-	DMA 存取	
DMAT	-	-	DMA 存取 (终端计数)	

- QN
显示范围 0 到 8 的指令队列缓冲器中存储的字节数。
- B-T
显示外部中断触发器 (外部跟踪信号输入电缆的 EXTIN7 引脚) 的电平。高电平 = “1”, 低电平 = “0”。
- Q-T
显示外部跟踪触发器 (外部跟踪信号输入电缆的 EXTIN6 引脚) 的电平。高电平 = “1”, 低电平 = “0”。
- 76543210
显示 8 位外部信号 (外部跟踪信号输入电缆的 EXTIN0 到 EXTIN7 引脚) 的状态。高电平 = “1”, 低电平 = “0”。
- h" m' s: ms.us
显示目标程序开始以后所经过的时间。

7.12.7 显示 M16C/R8C 调试器的总线信息

从左到右的内容如下：

- 地址
地址总线的状态
- 数据
数据总线的状态
- 总线
外部数据总线的宽度（“8b”表示 8 位数据总线，“16b”表示 16 位数据总线）
- BIU

此项显示 BIU（总线接口单元）和存储器以及 BIU 和 I/O 之间的状态。

显示格式	状态
-	无更改
INT	启动 INTACK 序列
IB	由于 CPU 原因导致的指令码读操作（字节）
DB	由于 CPU 原因导致的数据存取（字节）
IW	由于 CPU 原因导致的指令码读操作（字）
DW	由于 CPU 原因导致的数据存取（字）
DMA	非 CPU 原因导致的数据存取，例如 DMA
EMA	由于仿真器系统原因导致的数据存取

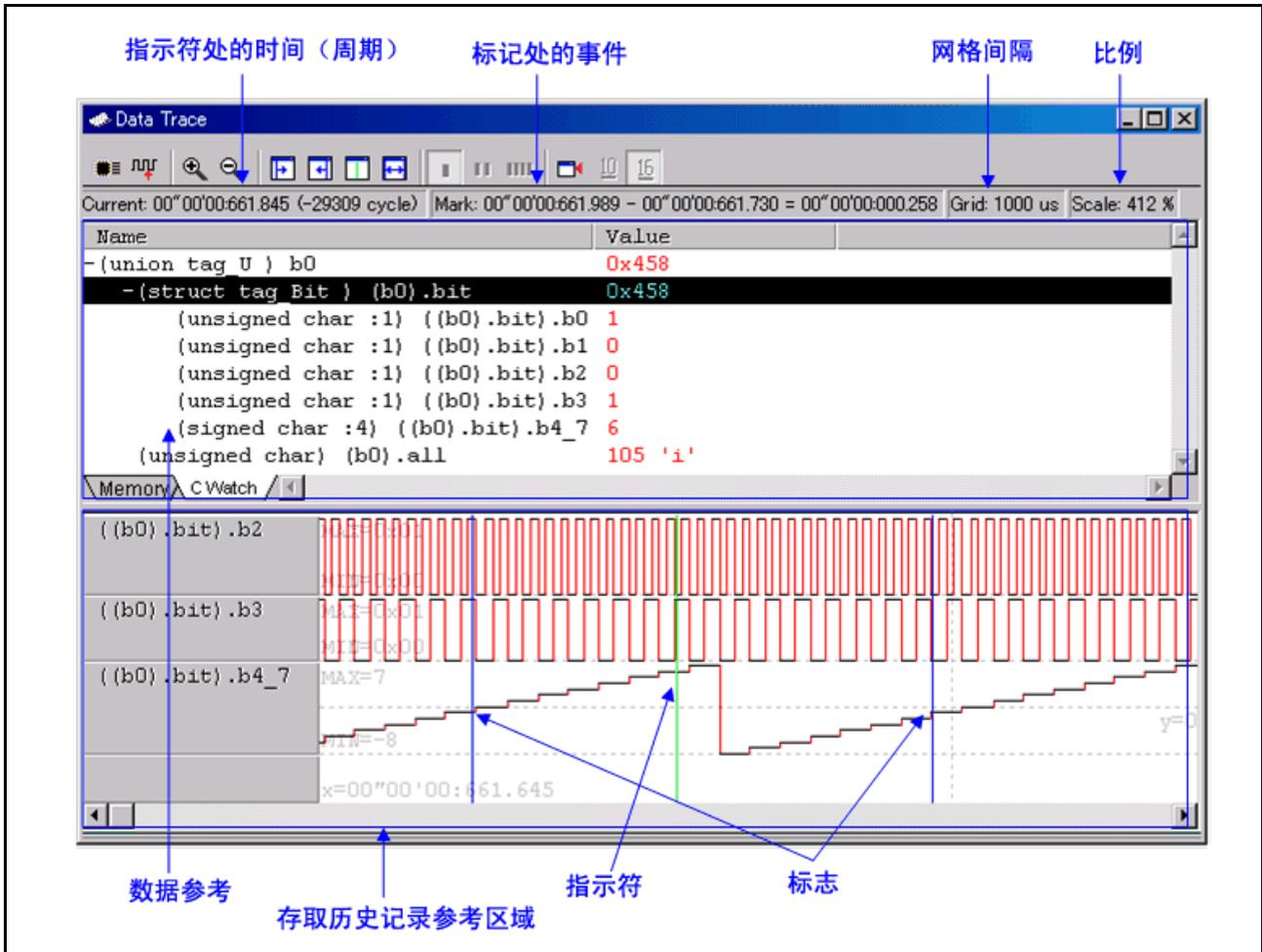
- R/W
显示数据总线的状态（“R”表示读，“W”表示写，“-”表示无存取）。
- RWT
此信号显示总线周期中的有效位置（有效时为“0”）。当 RWT 为“0”时，地址、数据和 BIU 信号有效。
- CPU
显示 CPU 和 BIU（总线接口单元）之间的状态

显示格式	状态
-	无更改
CB	操作码读（字节）
RB	操作数读（字节）
QC	指令队列缓冲器清除
CW	操作码读（字）
RW	操作数读（字）

- QN
显示范围 0 到 4 的指令队列缓冲器中存储的字节数。
- 76543210
显示 8 位外部信号（外部跟踪信号输入电缆的 EXTIN0 到 EXTIN7 引脚）的状态。高电平 = “1”，低电平 = “0”。
- h" m' s: ms.us
显示目标程序开始以后所经过的时间。

7.13 [Data Trace] (数据跟踪) 窗口

[Data Trace] (数据跟踪) 窗口用于分析实时跟踪计量结果，并用图形显示数据存取信息。该窗口与 [Trace] (跟踪) 窗口结合使用。



- 在数据引用区域，您可以检查当前涉及的周期点的存储器值，或检查已注册的 C 变量的值。
- 在存取历史记录引用区域，您可以查看图表形式的存取注册地址的历史记录。
- 与 [Trace] (跟踪) 窗口结合使用时，您可以在 [Trace] (跟踪) 窗口中查看正在监视的周期点的存储器值。反过来，您也可以您在 [Data Trace] (数据跟踪) 窗口中显示您正在监视的 [Trace] (跟踪) 窗口中的周期。

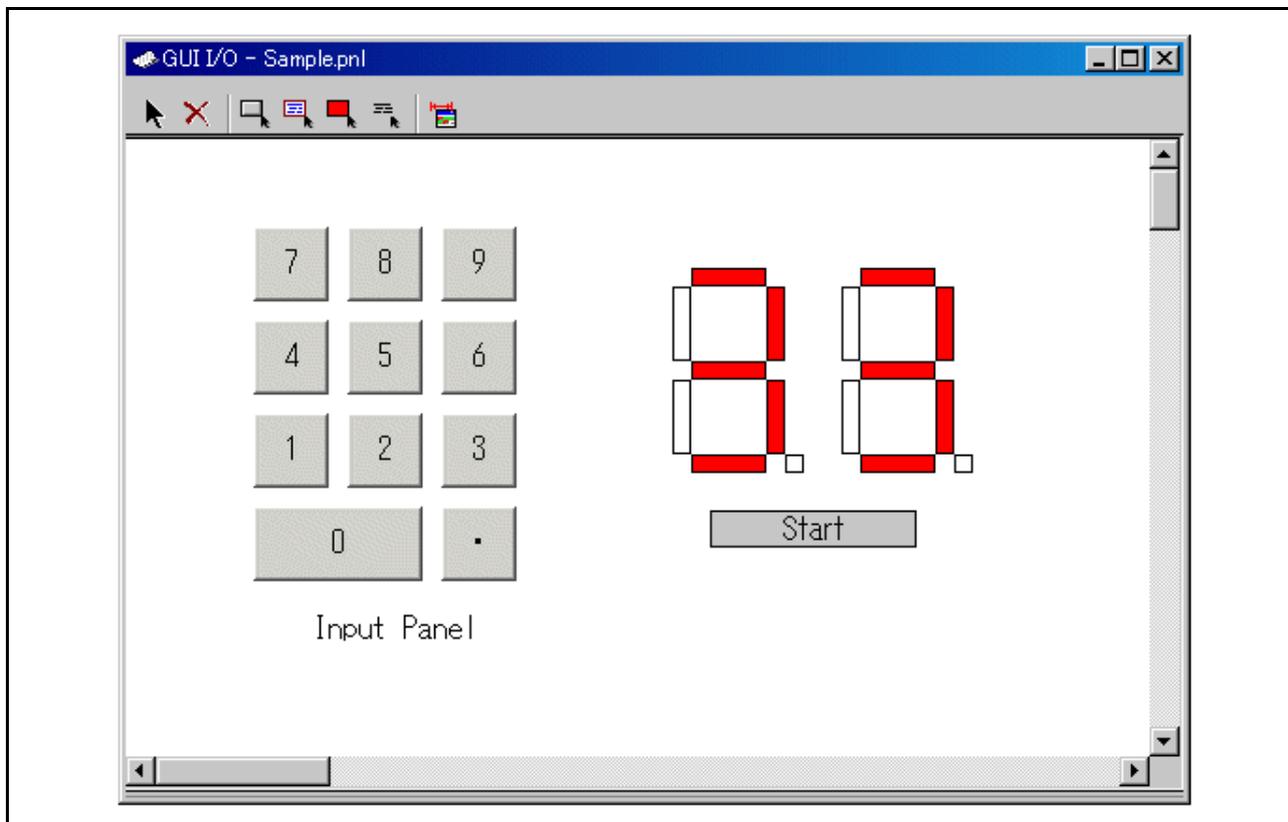
7.13.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Analyze Trace Data] (分析跟踪数据)		分析实时跟踪数据。
[Set Cycle...] (设置周期...)		指定显示周期。
[Sync with Trace Window] (与跟踪窗口同步)		与 [Trace] (跟踪) 窗口同步。
[Data Length] (数据长度)	[1byte] (1 字节)	以 1 字节为单位显示。
	[2bytes] (2 字节)	以 2 字节为单位显示。
	[4bytes] (4 字节)	以 4 字节为单位显示。
[Radix] (基数)	[Hex] (十六进制)	以十六进制显示。
	[Dec] (十进制)	以十进制显示。
[Address...] (地址...)		从指定地址显示。
[Add C Watch] (添加 C 监视)		添加 C 监视点。
[Remove C Watch] (移除 C 监视)		移除选定的 C 监视点。
[Hide Type Name] (隐藏类型名称)		隐藏变量的类型名称。
[Add...] (添加...)		向存取历史记录引用区域添加新的监视项目。
[Remove] (移除)		移除存取历史记录引用区域中的选定监视项目。
[Zoom] (缩放)	[Zoom In] (放大)	增大显示比例。
	[Zoom Out] (缩小)	缩小显示比例。
	[Zoom...] (缩放...)	指定显示比例。
[Marker] (标志)	[Start Marker] (开始标志)	在显示区中移动开始标志。
	[End Marker] (结束标志)	在显示区中移动结束标志。
	[Indicator] (指示器)	在显示区中移动指示器。
	[Adjust] (调整)	设置标志间的周期范围。
[Change Grid Interval...] (更改网格间隔...)		更改网格间隔。
[Change Row Setting...] (更改行设置...)		更改选定行的设置。
[Color...] (颜色...)		更改显示颜色。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

7.14 [GUI I/O] 窗口

在 [GUI I/O] 窗口中，您可以创建用户目标系统键盘输入面板（按钮），并单击所创建的按钮来进行端口输入。另外，您还可以在此窗口中创建并使用用户目标系统输出面板。



- 可以在窗口上布置下列部件。
 - 标签（字符串）
如果有任意值写入指定地址（位），则显示/擦除用户指定的字符串。
 - 指示灯
如果有特定值写入指定地址（或指定位），则更改显示的颜色。（取代“指示灯亮起”）
 - 按钮
可在按下按钮时执行虚拟端口输入。
 - 文本
显示文本字符串。
- 也可将创建的面板保存到文件中并重新装入。
- 最多可为创建的部件设置 200 个地址点。如果为各个部件设置不同的地址，则最多可布置 200 个部件。

7.14.1 扩展菜单

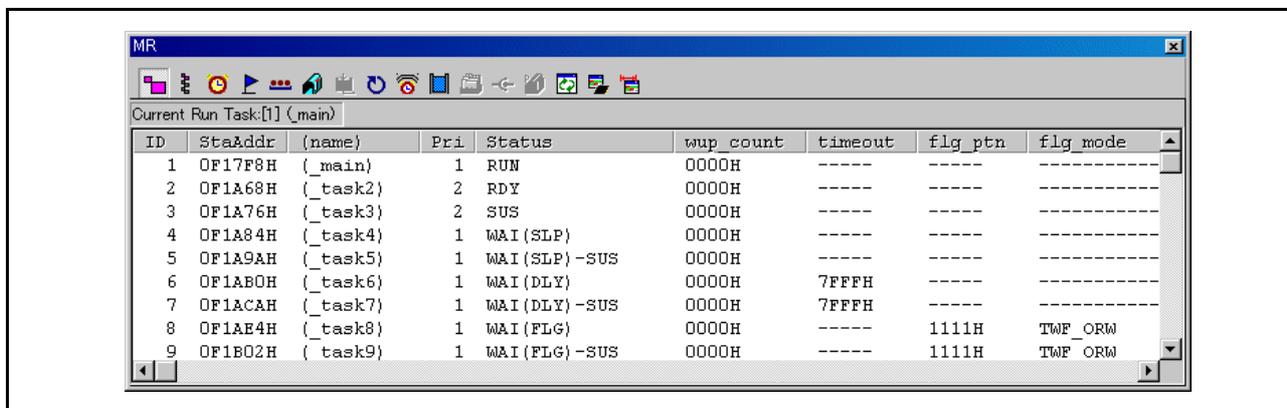
此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能
[Select Item] (选择项目)	选择 I/O 项目。
[Delete] (删除)	删除选定的 I/O 项目。
[Copy] (复制)	复制选定的 I/O 项目。
[Paste] (粘贴)	粘贴复制的 I/O 项目。
[Create Button] (创建按钮)	创建新按钮项目。
[Create Label] (创建标签)	创建新标签项目。
[Create LED] (创建指示灯)	创建新指示灯项目。
[Create Text] (创建文本)	创建新文本项目。
[Display grid] (显示网格)	显示网格线。
[Save...] (保存...)	保存 I/O 面板文件。
[Load...] (装入...)	装入 I/O 面板文件。
[Sampling Period...] (采样周期...)	设置 RAM 监控采样周期。
[Toolbar display] (工具栏显示)	显示工具栏。
[Customize toolbar...] (自定义工具栏...)	打开工具栏自定义对话框。
[Allow Docking] (允许入坞)	允许窗口入坞。
[Hide] (隐藏)	隐藏窗口。

7.15 [MR] 窗口

使用 [MR] 窗口可显示实时操作系统的状态。

只有下载使用实时操作系统的程序后，才能使用 [MR] 窗口（如果下载的程序不使用 MR，则打开 [MR] 窗口后，该窗口中不会显示任何内容。）



- 可以打开与显示模式数量同样多的 [MR] 窗口。
- 单击所需按钮，[MR] 窗口显示模式及显示数据会更改。
- 双击所需的任务行，可显示任务的上下文数据。
- 可以拖动光标来更改各个模式中显示区的宽度。
- 如果下载的程序不使用 MR，则无法选择用于选择显示模式的所有菜单。
- 可用的显示模式因 MRxx 而异。

注意事项

制作下载的程序时，请使用内容与 MRxx 版本匹配的启动文件 (crt0mr.axx/start.axx)。如果使用的启动文件与 MRxx 的版本不匹配，则 [MR] 窗口和 MR 命令将无法正常运行。

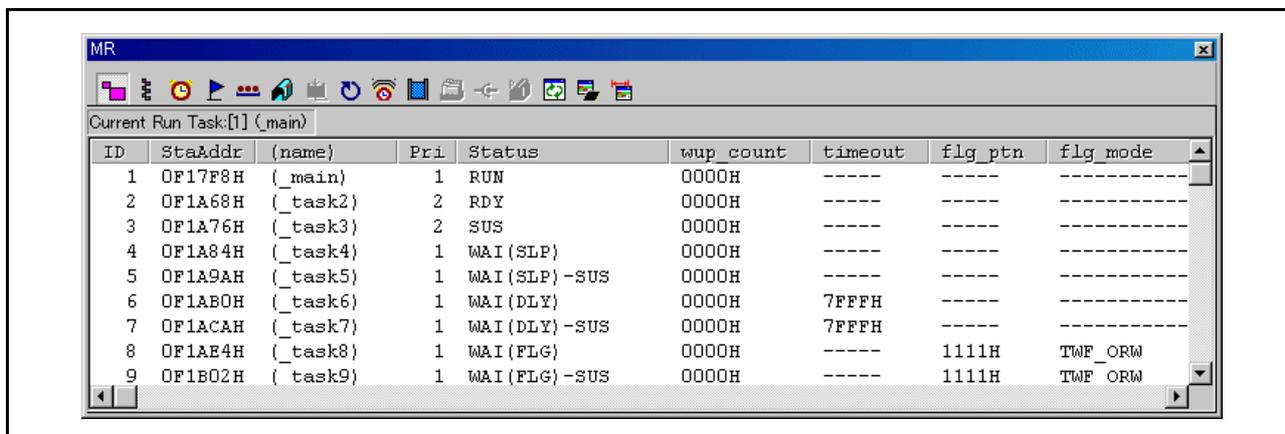
7.15.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Mode] (模式)	[Task] (任务)	显示任务状态。
	[Ready Queue] (就绪队列)	显示就绪状态。
	[Timeout Queue] (超时队列)	显示超时状态。
	[Event Flag] (事件标志)	显示事件标志状态。
	[Semaphore] (信标)	显示信标状态。
	[Mailbox] (邮箱)	显示邮箱状态。
	[Data Queue] (数据队列)	显示数据队列状态。
	[Cyclic Handler] (周期处理程序)	显示周期处理程序状态。
	[Alarm Handler] (警报处理程序)	显示警报处理程序状态。
	[Memory Pool] (存储池)	显示存储池状态。
	[Message Buffer] (消息缓冲器)	显示消息缓冲器状态。
	[Port] (端口)	显示端口状态。
	[Mailbox(with Priority)] (邮箱 (带优先级))	显示邮箱 (带优先级) 状态。
[Context...] (上下文...)		显示上下文。
[Layout] (布局)	[Status Bar] (状态栏)	在显示或不显示状态栏之间切换。
[Refresh] (刷新)		刷新存储器数据。
[RAM Monitor] (RAM 监控)	[Enable RAM Monitor] (允许 RAM 监控)	在允许或禁止 RAM 监控功能之间切换。
	[Sampling Period...] (采样周期...)	设置 RAM 监控采样周期。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

7.15.2 显示任务状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Task]（任务）。



双击任意行，[Context]（上下文）对话框中将显示任务上下文的相关信息。有关 [Context]（上下文）对话框的详细信息，请参阅“7.15.2 显示任务状态”状态栏中会显示下列数据。



7.15.2.1 显示任务状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有任务会按照 ID 号顺序列出。每个项目的功能描述如下。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	任务 ID
[StaAddr]	任务的起始地址
[(name)]（（名称））	任务名称
[Pri]	优先级
[Status]（状态）*1	任务状态
[wup_count]	唤醒计数
[timeout]（超时）	超时值
[flg_ptn]	事件标志的等待位模式（Wait bit pattern）
[flg_mode*2]	事件标志的等待取消条件

- *1 任务状态

显示	状态
RUN	运行状态
RDY	就绪状态
SUS	挂起状态
DMT	休眠状态
WAI(SLP)	睡眠状态
WAI(SLP)-SUS	睡眠状态 (挂起)
WAI(SLP-TMO)	超时睡眠状态
WAI(SLP-TMO)-SUS	超时睡眠状态 (挂起)
WAI(DLY)	dly_tsk 导致的延迟状态
WAI(DLY)-SUS	dly_tsk 导致的延迟状态 (挂起)
WAI(FLG-TMO)	事件标志的等待状态
WAI(FLG-TMO)-SUS	事件标志的等待状态 (挂起)
WAI(SEM)	信标资源的等待状态
WAI(SEM)-SUS	信标资源的等待状态 (挂起)
WAI(SEM-TMO)	超时信标资源的等待状态
WAI(SEM-TMO)-SUS	超时信标资源的等待状态 (挂起)
WAI(MBX)	邮箱的接收等待状态
WAI(MBX)-SUS	邮箱的接收等待状态 (挂起)
WAI(MBX-TMO)	超时邮箱的接收等待状态
WAI(MBX-TMO)-SUS	超时邮箱的接收等待状态 (挂起)

- *2 显示事件标志的等待取消条件

flg_mode	状态
TWF_ANDW	等待设置等待位模式 (Wait bit pattern) 中设置的所有位 (AND 等待)
TWF_ANDW+TWF_CLR	出现 AND 等待且任务等待状态已取消时, 将事件标志清零
TWF_ORW	等待设置等待位模式 (Wait bit pattern) 中设置的任一位 (OR 等待)
TWF_ORW+TWF_CLR	出现 OR 等待且任务等待状态已取消时, 将事件标志清零

7.15.2.2 显示任务状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有任务会按照 ID 号顺序列出。每个项目的功能描述如下。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	任务 ID
[Name] (名称)	任务名称
[Pri]	优先级
[Status] (状态) *1	任务状态
[Wupcnt]	唤醒计数
[Actcnt]	激活计数
[Tmout]	超时值
[Flgptn]	事件标志的等待位模式 (Wait bit pattern)
[Wfmode]*2	事件标志的等待取消条件

- *1 任务状态

显示	状态
RUN	运行状态
RDY	就绪状态
SUS	挂起状态
DMT	休眠状态
WAI(SLP)	睡眠状态
WAI(SLP)-SUS	睡眠状态 (挂起)
WAI(SLP-TMO)	超时睡眠状态
WAI(SLP-TMO)-SUS	超时睡眠状态 (挂起)
WAI(DLY)	dly_tsk 导致的延迟状态
WAI(DLY)-SUS	dly_tsk 导致的延迟状态 (挂起)
WAI(FLG)	事件标志的等待状态
WAI(FLG)-SUS	事件标志的等待状态 (挂起)
WAI(FLG-TMO)	事件标志的等待状态
WAI(FLG-TMO)-SUS	事件标志的等待状态 (挂起)
WAI(SEM)	信标资源的等待状态
WAI(SEM)-SUS	信标资源的等待状态 (挂起)
WAI(SEM-TMO)	超时信标资源的等待状态
WAI(SEM-TMO)-SUS	超时信标资源的等待状态 (挂起)
WAI(MBX)	邮箱的接收等待状态
WAI(MBX)-SUS	邮箱的接收等待状态 (挂起)
WAI(MBX-TMO)	超时邮箱的接收等待状态
WAI(MBX-TMO)-SUS	超时邮箱的接收等待状态 (挂起)
WAI(SDTQ)	数据队列的发送等待状态
WAI(SDTQ)-SUS	数据队列的发送等待状态 (挂起)
WAI(SDTQ-TMO)	超时数据队列的发送等待状态

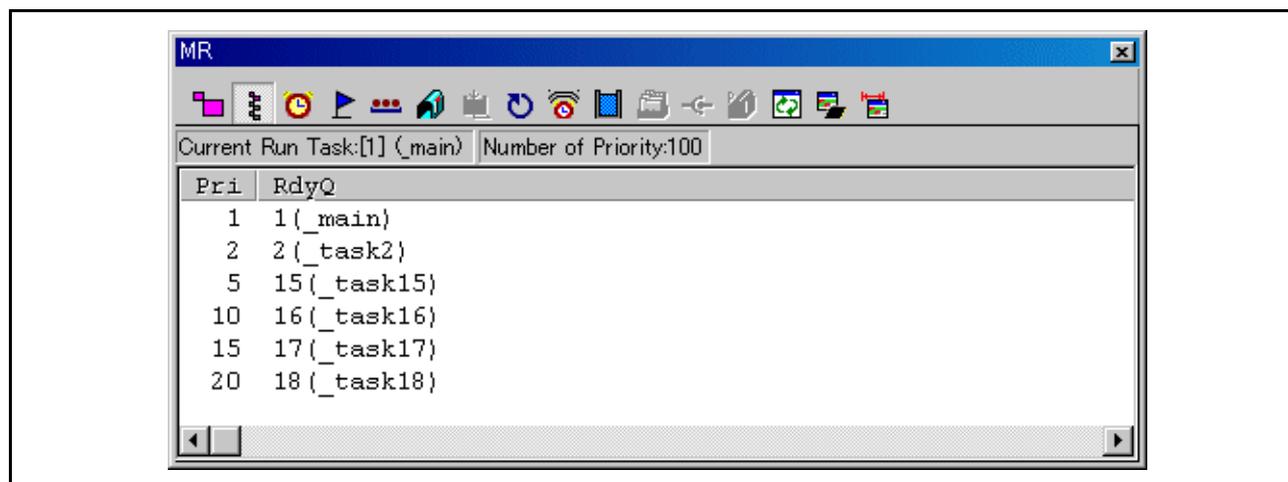
显示	状态
WAI(SDTQ-TMO)-SUS	超时数据队列的发送等待状态 (挂起)
WAI(RDTQ)	数据队列的接收等待状态
WAI(RDTQ)-SUS	数据队列的接收等待状态 (挂起)
WAI(RDTQ-TMO)	超时数据队列的接收等待状态
WAI(RDTQ-TMO)-SUS	超时数据队列的接收等待状态 (挂起)
WAI(VSDTQ)	扩展数据队列的发送等待状态
WAI(VSDTQ)-SUS	扩展数据队列的发送等待状态 (挂起)
WAI(VSDTQ-TMO)	超时扩展数据队列的发送等待状态
WAI(VSDTQ-TMO)-SUS	超时扩展数据队列的发送等待状态 (挂起)
WAI(VRDTQ)	扩展数据队列的接收等待状态
WAI(VRDTQ)-SUS	扩展数据队列的接收等待状态 (挂起)
WAI(VRDTQ-TMO)	超时扩展数据队列的接收等待状态
WAI(VRDTQ-TMO)-SUS	超时扩展数据队列的接收等待状态 (挂起)
WAI(MPF)	固定大小存储块的等待状态
WAI(MPF)-SUS	固定大小存储块的等待状态 (挂起)
WAI(MPF-TMO)	超时固定大小存储块的等待状态
WAI(MPF-TMO)-SUS	超时固定大小存储块的等待状态 (挂起)

- *2 显示事件标志的等待取消条件

Wfmode	状态
TWF_ANDW	等待设置等待位模式 (Wait bit pattern) 中设置的所有位 (AND 等待)
TWF_ORW	等待设置等待位模式 (Wait bit pattern) 中设置的任一位 (OR 等待)

7.15.3 显示就绪队列状态

在 [MR] 窗口中, 选择弹出菜单 - [Mode] (模式) -> [Ready Queue] (就绪队列)。



状态栏中会显示下列数据。



7.15.3.1 显示就绪队列状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

每个项目的功能描述如下。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[Pri]	显示优先级
[RdyQ]	显示就绪队列中任务的 ID 号和任务名称

- [RdyQ] 字段最多显示任务名称的 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.3.2 显示就绪队列状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

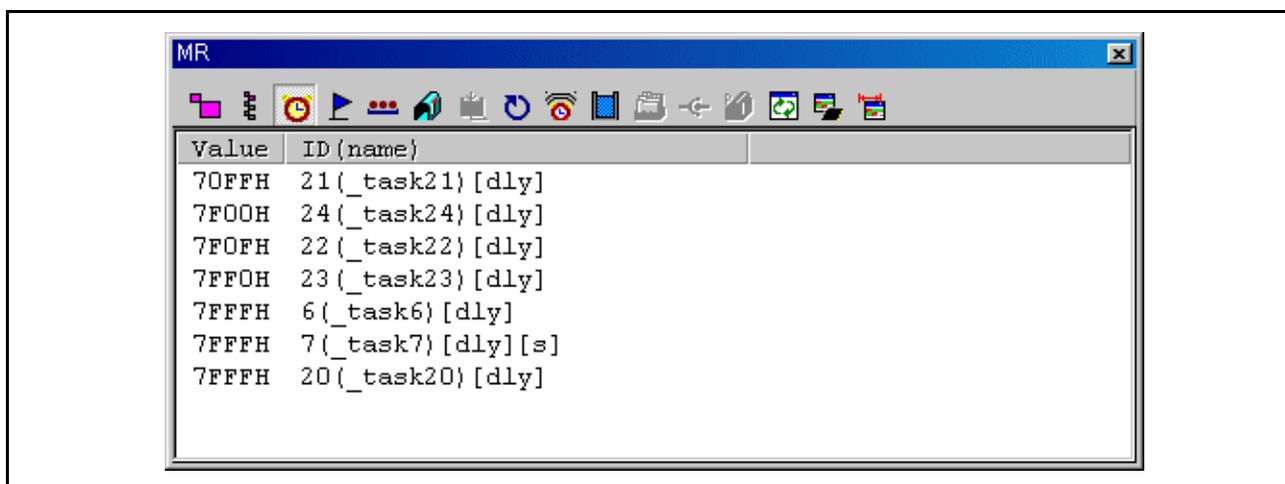
每个项目的功能描述如下。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[Pri]	显示优先级
[Ready Queue]（就绪队列）	显示就绪队列中任务的 ID 号和任务名称

- [Ready Queue]（就绪队列）字段最多显示任务名称的 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.4 显示超时队列状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Timeout Queue]（超时队列）。



7.15.4.1 显示超时队列状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

每个项目的功能描述如下。

当前处于等待状态的任务按照超时值的降序显示。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[Value]（值）	显示每个任务的超时值
[ID(name)]（ID（名称））	显示超时队列中任务的 ID 号和任务名称

- 下列字符串用于指示等待状态的类型。

字符串	等待状态
[slp]	tslp_tsk 导致的等待
[dly]	dly_tsk 导致的等待
[flg]	twai_flg 导致的等待
[sem]	twai_sem 导致的等待
[mbx]	trcv_msg 导致的等待

- 当连接到超时队列的任务处于强制等待（双重等待）状态时，[ID(name)]（ID（名称））字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]

7.15.4.2 显示超时队列状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

每个项目的功能描述如下。

当前处于等待状态的任务按照超时值的降序显示。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[Tmout]	显示每个任务的超时值 (ms)
[ID(Name)]（ID（名称））	显示超时队列中任务的 ID 号和任务名称

- 下列字符串用于指示等待状态的类型。

字符串	等待状态
[slp]	tslp_tsk 导致的等待
[dly]	dly_tsk 导致的等待
[flg]	twai_flg 导致的等待
[sem]	twai_sem 导致的等待
[mbx]	trcv_mbx 导致的等待
[mpf]	tget_mpf 导致的等待
[sdtq]	tsnd_dtq 导致的等待

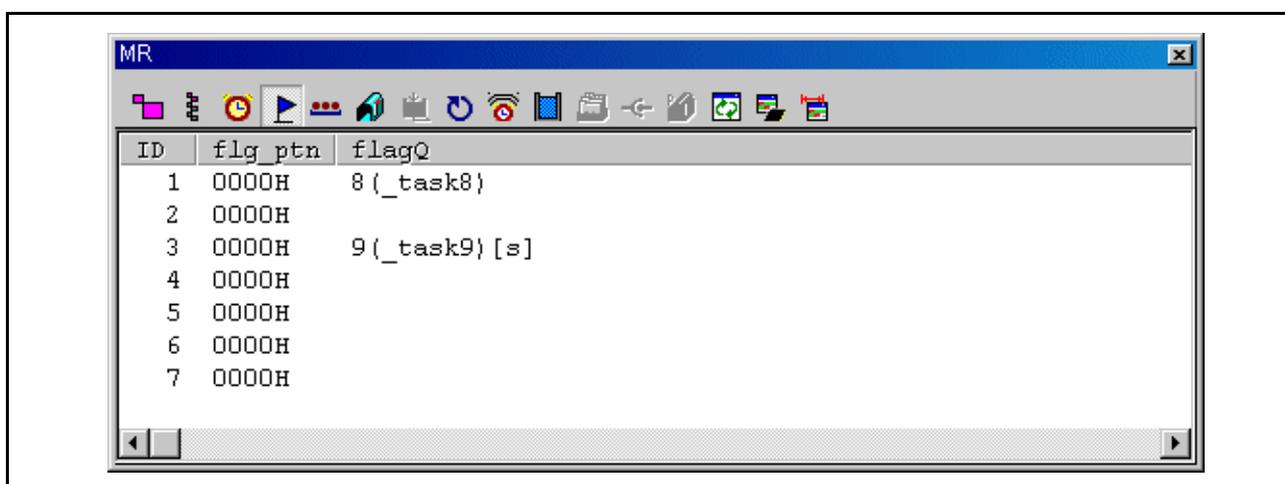
字符串	等待状态
[rdtq]	trcv_dtq 导致的等待
[vsdtq]	vtsnd_dtq 导致的等待
[vrdtq]	vtrcv_dtq 导致的等待

- 当连接到超时队列的任务处于强制等待（双重等待）状态时，[ID(Name)]（ID（名称））字段显示的字符串后面会追加一个字符串 “[s]”，表明处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]

7.15.5 显示事件标志状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Event Flag]（事件标志）。



7.15.5.1 显示事件标志状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有事件标志会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	事件标志的 ID 号
[flg_ptn]	每个事件标志的位模式（Bit pattern）
[flagQ]	事件标志队列中的任务 ID 号和任务名称

- 当连接到事件标志队列的任务处于允许超时等待（twai_flg 等待）状态时，[flag Q]（标志队列）字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时等待状态。

当连接到事件标志队列的任务处于强制等待（双重等待）状态时，[flag Q]（标志队列）字段显示的字符串后面会追加一个字符串 “[s]”，表明处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [flag Q]（标志队列）字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.5.2 显示事件标志状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有事件标志将按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	事件标志的 ID 号
[Flgatr]	每个事件标志的属性
[Flgptn]	每个事件标志的位模式（Bit pattern）
[Flag Queue]（标志队列）	事件标志队列中的任务 ID 号和任务名称

- Flgatr 区域显示内容如下：

TA_TFIFO	任务等待队列处于 FIFO 顺序
TA_TPRI	任务等待队列处于任务优先级顺序
TA_WSGL	只允许一个任务处于事件标志等待状态
TA_WMUL	允许多个任务处于事件标志等待状态
TA_CLR	任务脱离等待状态后，该事件标志的位模式（Bit pattern）将被清除

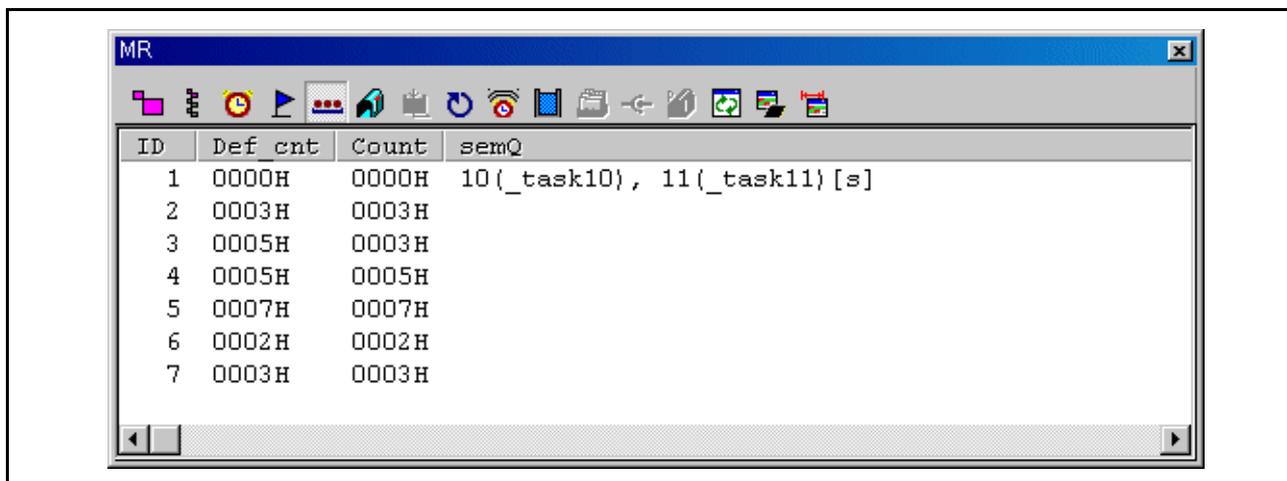
- 当连接到事件标志队列的任务处于允许超时等待（twai_flg 等待）状态时，[Flag Queue]（标志队列）字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时等待状态。
当连接到事件标志队列的任务处于强制等待（双重等待）状态时，[Flag Queue]（标志队列）字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [Flag Queue]（标志队列）字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.6 显示信标状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Semaphore]（信标）。



7.15.6.1 显示信标状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有信标将按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	信标的 ID 号
[Def_cnt]	信标计数器的默认值
[Count]（计数）	信标计数
[semQ]	信标队列中的任务 ID 号和任务名称

- 当连接到信标队列的任务处于允许超时等待（twai_sem 等待）状态时，[semQ] 字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时等待状态。

当连接到信标队列的任务处于强制等待（双重等待）状态时，[semQ] 字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [semQ] 字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.6.2 显示信标状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有信标会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	信标的 ID 号
[Sematr]	每个信标的属性
[Semcnt]	信标计数
[Semaphore Queue]（信标队列）	信标队列中的任务 ID 号和任务名称

- Sematr 区域显示内容如下：

TA_TFIFO	任务等待队列处于 FIFO 顺序
TA_TPRI	任务等待队列处于任务优先级顺序

- 当连接到信标队列的任务处于允许超时等待（twai_sem 等待）状态时，[Semaphore Queue]（信标队列）字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时等待状态。

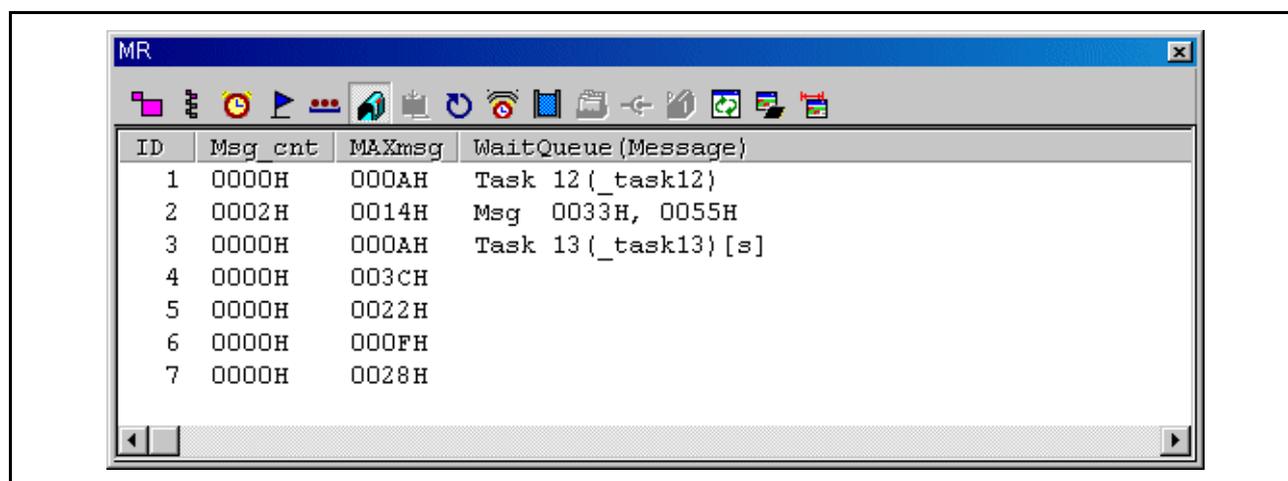
当连接到信标队列的任务处于强制等待（双重等待）状态时，[Semaphore Queue]（信标队列）字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [Semaphore Queue]（信标队列）字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.7 显示邮箱状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Mailbox]（邮箱）。



7.15.7.1 显示邮箱状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有邮箱会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	邮箱的 ID 号
[Msg_cnt]	每个邮箱中的邮件数量
[MAXmsg]	每个邮箱可以容纳邮件的最大数量
[Wait Queue(Message)]（等待队列（邮件））	存储在邮箱中的邮件或正在等待邮件的任务的 ID 号和任务名称

- [WaitQueue (Message)]（等待队列（邮件））字段会在存储邮件后（上述 [Msg_cnt] 不为零时）显示字符串“Msg”，然后显示存储的邮件。

未存储任何邮件时（[Msg_cnt] 为零时），如果存在等待邮件的任务，则 [WaitQueue]（等待队列）字段显示字符串“Task”（任务），然后显示正在等待邮件的任务的 ID 号和名称。

- 当连接到邮箱队列的任务处于允许超时等待（trcv_msg 等待）状态时，[WaitQueue (Message)]（等待队列（邮件））字段显示的字符串后面会追加一个字符串“[tmo]”，表示处于允许超时状态。

当连接到邮箱队列的任务处于强制等待（双重等待）状态时，[WaitQueue (Message)]（等待队列（邮件））字段显示的字符串后面会追加一个字符串“[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [WaitQueue (Message)]（等待队列（邮件））字段的任务名称最多可以显示 8 个字符。

如果任务名称超过 8 个字符，则忽略多余字符。

7.15.7.2 显示邮箱状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有邮箱会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	邮箱的 ID 号
[Mbxatr]	每个邮箱的属性
[Mailbox Queue (Wait)] （邮箱队列（等待））	正在等待邮件的任务的 ID 号和任务名称
[Mailbox Queue (Message)] （邮箱队列（邮件））	存储在邮箱中的邮件

- Mbxatr 区域显示内容如下：

TA_TFIFO	任务等待队列处于 FIFO 顺序
TA_TPRI	任务等待队列处于任务优先级顺序
TA_MFIFO	邮件队列处于 FIFO 顺序
TA_MPRI	邮件队列处于邮件优先级顺序

- 当连接到邮箱队列的任务处于允许超时等待（trcv_mbx 等待）状态时，[Mailbox Queue (Wait)]（邮箱队列（等待））字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时状态。

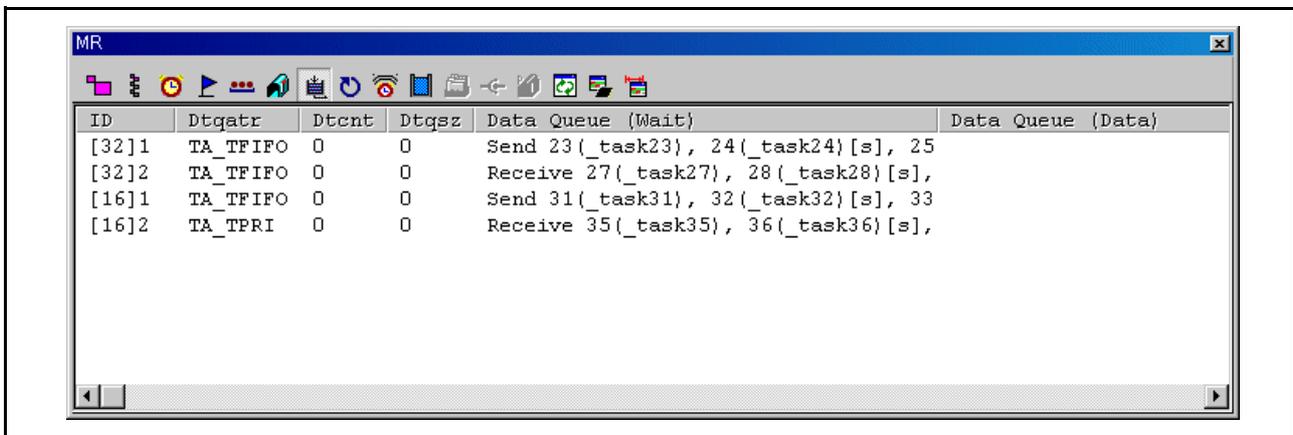
当连接到邮箱队列的任务处于强制等待（双重等待）状态时，[Mailbox Queue (Wait)]（邮箱队列（等待））字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [Mailbox Queue (Wait)]（邮箱队列（等待））字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.8 显示数据队列状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Data Queue]（数据队列）。



7.15.8.1 显示数据队列状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有数据队列会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	数据队列的 ID 号
[Dtqatr]	每个数据队列的属性
[Dtcnt]	每个数据队列中的邮件数量
[DtqsZ]	每个数据队列可以容纳邮件的最大数量
[Data Queue (Wait)]（数据队列（等待））	正在等待邮件传输等待或邮件接收等待的任务的 ID 号和任务名称
[Data Queue (Data)]（数据队列（数据））	存储在数据队列中的邮件

- [ID] 字段的显示内容取决于指定的是标准数据（32 位）还是扩展数据（16 位）。

MR308/4

- 如果指定的是标准数据（32 位），则 [ID] 字段显示字符串 “[32]” 和数据队列 ID 号。
- 如果指定的是扩展数据（16 位），则 [ID] 字段显示字符串 “[16]” 和数据队列 ID 号。

MR30/4

- 如果指定的是标准数据（16 位），则 [ID] 字段显示字符串 “[16]” 和数据队列 ID 号。
- 如果指定的是扩展数据（32 位），则 [ID] 字段显示字符串 “[32]” 和数据队列 ID 号。

- Dtqatr 区域显示内容如下：

TA_TFIFO	任务等待队列处于 FIFO 顺序
TA_TPRI	任务等待队列处于任务优先级顺序

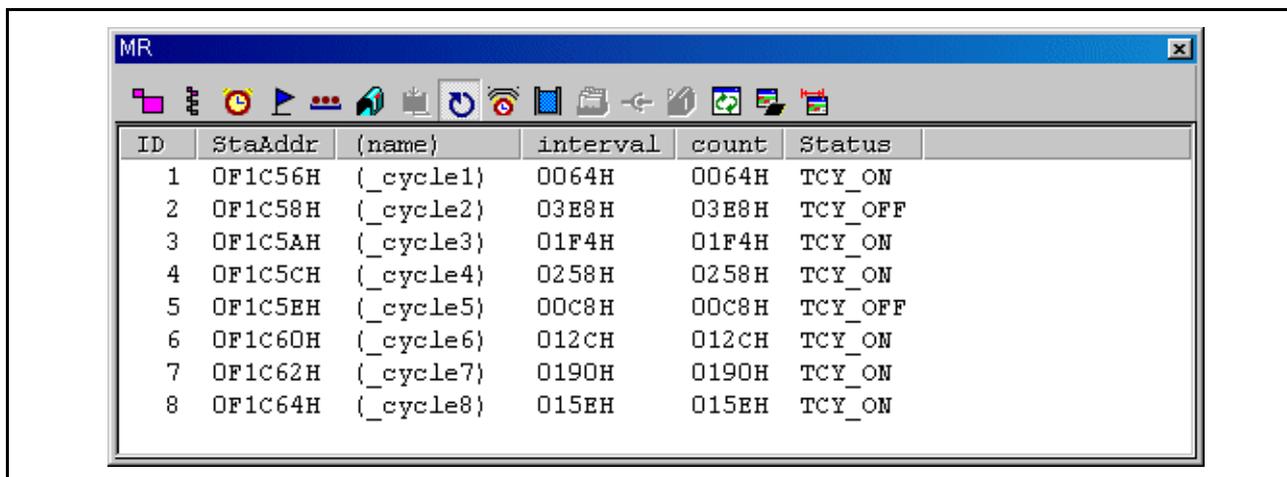
- 如果任务正在等待邮件发送，则 [Data Queue (Wait)]（数据队列（等待））字段显示字符串 “Send”（发送），然后显示正在等待邮件发送的任务的 ID 号和名称。同样，如果任务正在等待邮件接收，则显示字符串 “Receive”（接收），然后显示正在等待邮件接收的任务的 ID 号和名称。
- 当连接到数据队列的任务处于允许超时等待状态时，[Data Queue (Wait)]（数据队列（等待））字段显示的字符串后面会追加一个字符串 “[tmo]”，表示处于允许超时状态。
当连接到数据队列的任务处于强制等待（双重等待）状态时，[Data Queue (Wait)]（数据队列（等待））字段显示的字符串后面会追加一个字符串 “[s]”，表示处于双重等待状态。

普通显示	26(_task26)
处于 WAIT-SUSPEND 状态时的显示	26(_task26)[s]
处于超时 WAIT-SUSPEND 状态时的显示	26(_task26)[tmo][s]

- [Data Queue (Wait)]（数据队列（等待））字段的任务名称最多可以显示 8 个字符。如果任务名称超过 8 个字符，则忽略多余字符。

7.15.9 显示周期处理程序状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Cyclic Handler]（周期处理程序）。



7.15.9.1 显示周期处理程序状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有周期处理程序会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	周期处理程序的 ID 号
[StaAddr]	周期处理程序的起始地址
[(name)]（名称）	周期处理程序的名称
[interval]（间隔）	中断间隔
[count]（计数）	中断计数
[Status]（状态）	周期开始处理程序的活动状态

- [Status]（状态）区域显示内容如下：

TCY_ON	允许周期处理程序
TCY_OFF	禁止周期处理程序

7.15.9.2 显示周期处理程序状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

配置中定义的所有周期处理程序会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

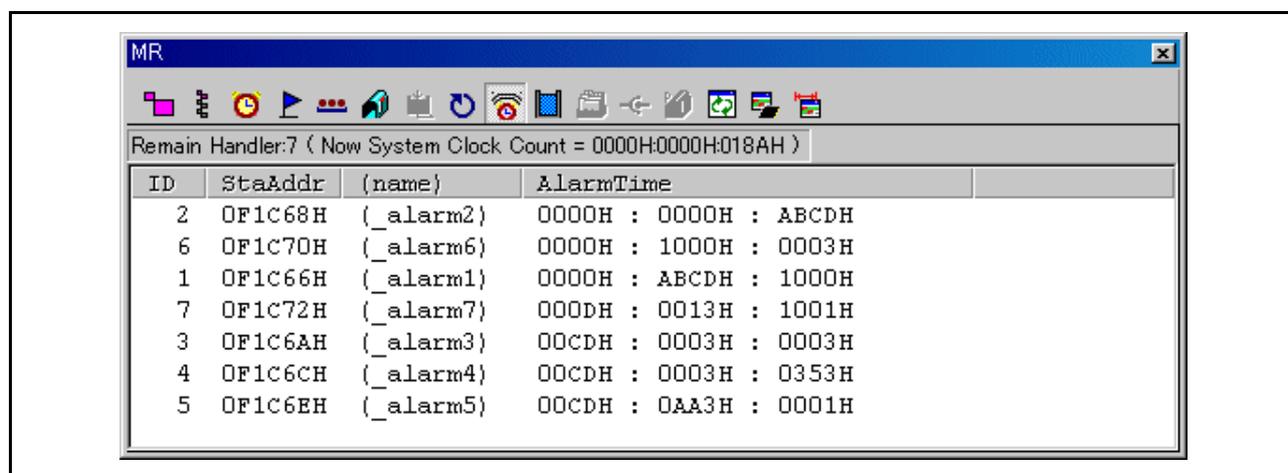
项目	内容
[ID]	周期处理程序的 ID 号
[Name] (名称)	周期处理程序的名称
[Cycphs]	激活阶段 (毫秒)
[CycTim]	激活周期时间 (毫秒)
[Tmout]	到周期处理程序下次激活时间之前剩余的时间量 (毫秒)
[Status] (状态)	周期开始处理程序的活动状态

- [Status] (状态) 区域显示内容如下:

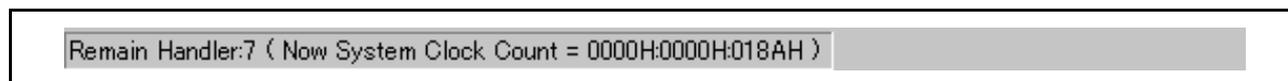
TCYC_STA	周期处理程序处于操作状态
TCYC_STP	周期处理程序处于非操作状态

7.15.10 显示警报处理程序状态

在 [MR] 窗口中, 选择弹出菜单 - [Mode] (模式) -> [Alarm Handler] (警报处理程序)。



当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时, 状态栏中会显示下列数据。



7.15.10.1 显示警报处理程序状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

在配置中定义的所有周期开始处理程序中，只有目前尚未开始的处理程序才会按照开始时间升序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	警报处理程序的 ID 号
[StaAddr]	警报处理程序的起始地址
[(name)]（(名称)）	警报处理程序的名称
[AlarmTime]（警报时间）	警报处理程序的开始时间

7.15.10.2 显示警报处理程序状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

在配置中定义的所有周期开始处理程序中，只有目前尚未开始的处理程序才会按照开始时间升序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

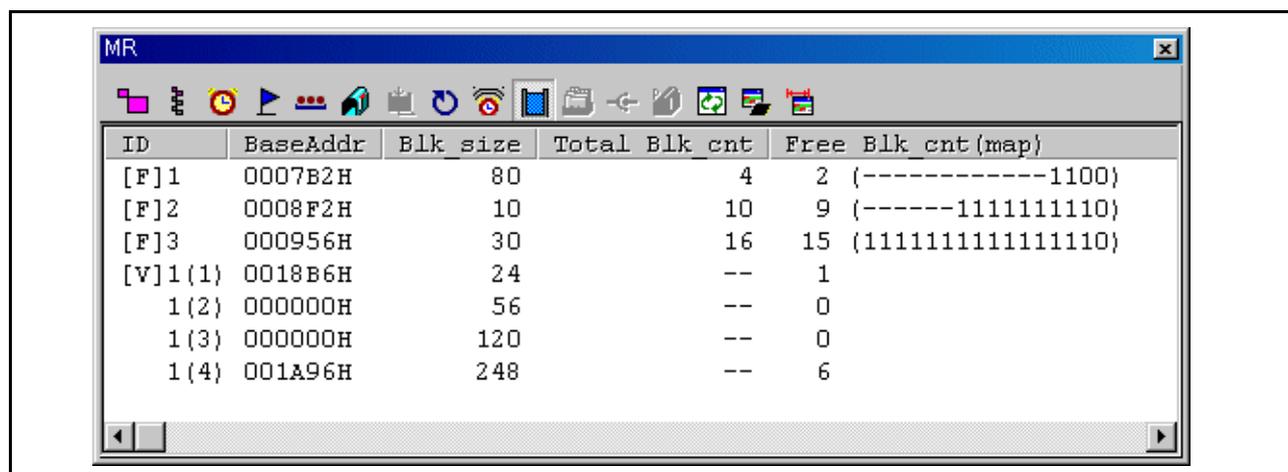
项目	内容
[ID]	警报处理程序的 ID 号
[Name]（名称）	警报处理程序的名称
[Almtim]	到警报处理程序激活时间之前剩余的时间量（毫秒）
[Status]（状态）	警报处理程序的活动状态

- [Status]（状态）区域显示内容如下：

TALM_STA	警报处理程序处于操作状态
TALM_STP	警报处理程序处于非操作状态

7.15.11 显示存储池状态

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Memory Pool]（存储池）。



7.15.11.1 显示存储池状态（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

配置中定义的所有存储池会按照 ID 号顺序列出。（固定长度数据在前，任意长度数据在后。）下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.3.0 的 MRxx 时。）

项目	内容
[ID]	存储池的 ID 号
[BaseAddr]	存储池的基址
[Blk_Size]	存储池的块大小
[Total Blk_cnt]（总块数）	存储池的总块数
[Free Blk_cnt(map)]（自由块数（映射））	未使用块的数量及未使用存储块的相关信息（位信息）

- [ID] 字段的显示内容取决于指定的是固定长度还是任意长度。
 - 如果数据为固定长度，则 [ID] 字段显示字符串 “[F]” 和存储池 ID 号。
 - 对于任意长度，第一行显示的内容是字符串 “[V]”、存储池 ID 号和块 ID 号。第二行至第四行显示存储池 ID 和块 ID 号。块 ID 号包含在括号中。
- 指定任意长度存储池时，[Total Blk_cnt]（总 Blk_cnt）字段中显示 “--”。
 - [Free Blk_cnt (map)]（自由块数（映射））字段中不显示任何位信息。
- 指定固定长度存储池时，[Free Blk_cnt (map)]（自由块数（映射））的存储块信息中每个位的显示格式如下所示：

项目	内容
'0'	正在使用的存储块（忙）
'1'	未使用的存储块（就绪）
'.'	无存储块

7.15.11.2 显示存储池状态（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

所有存储池会按照 ID 号顺序列出。下面列出了每个项目的功能。（当实时操作系统为符合 uITRON 规范 V.4.0 的 MRxx 时。）

项目	内容
[ID]	存储池的 ID 号
[Mplatr]	每个存储池的属性
[Mpladr]	存储池的基址
[Mplsz]	存储池的大小
[Blkcnt]	固定长度存储池的总块数
[Fblkcnt]	未使用块的数量及未使用存储块的相关信息
[Memory Pool Queue]（存储池队列）	显示存储池中正在等待的任务的 ID 号和名称。

- Mplatr 区域显示内容如下：

TA_TFIFO	任务等待队列处于 FIFO 顺序
TA_TPRI	任务等待队列处于任务优先级顺序

- [ID] 字段的显示内容取决于指定的是固定长度还是任意长度。
 - 如果数据为固定长度，则 [ID] 字段显示字符串 “[F]” 和存储池 ID 号。
 - 对于任意长度，第一行显示的内容是字符串 “[V]”、存储池 ID 号和块 ID 号。第二行至第四行显示存储池 ID 和块 ID 号。块 ID 号包含在括号中。

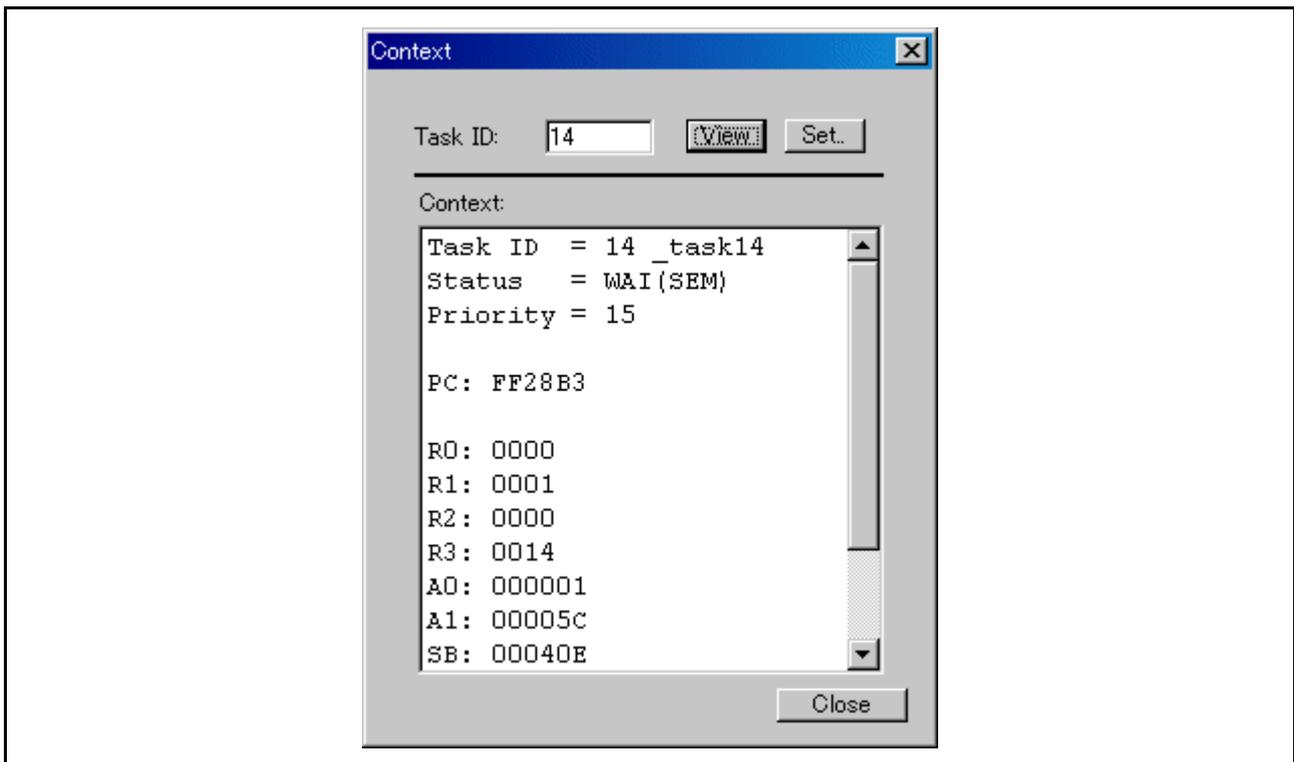
7.15.12 显示任务上下文

7.15.12.1 显示任务上下文

在 [MR] 窗口中，选择弹出菜单 - [Mode]（模式）-> [Context...]（上下文...）。

此时会打开 [Context]（上下文）对话框。[Context]（上下文）对话框用于引用 / 指定所指定任务的上下文信息。

也可通过双击任务状态显示模式的数据显示区来打开 [Context]（上下文）对话框。



在 [Task ID]（任务 ID）字段输入任务 ID 号并单击 [View]（视图）按钮（或按 Enter 键）。

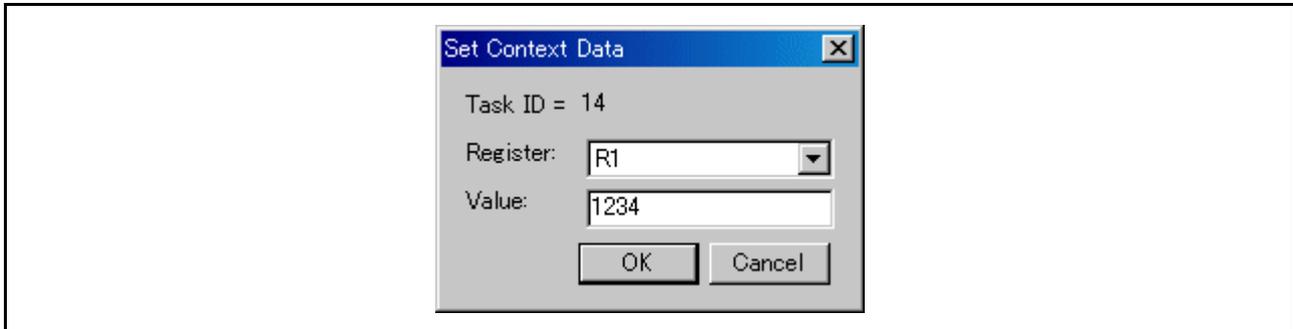
[Context]（上下文）字段中显示指定任务的上下文。

- 如果单击 [View]（视图）按钮时，在 [Task ID]（任务 ID）字段中输入的任务为 “RUN”（运行）或 “DMT”（休眠），则不显示上下文。（[Context]（上下文）字段仅显示任务 ID 和任务状态。）
- 如果单击 [View]（视图）按钮时，在 [Task ID]（任务 ID）字段中输入的任务 ID 号不存在，则会出现错误。

7.15.12.2 更改任务上下文

在 [Context]（上下文）对话框的 [Task ID]（任务 ID）字段中输入任务 ID 号并单击 [Set]（设置）按钮。此时会打开 [Set Context]（设置上下文）对话框。

[Set Context]（设置上下文）对话框用于设置指定任务的指定上下文寄存器值。



在 [Register]（寄存器）字段列表框中指定要更改的寄存器，然后在 “Value:”（值:）字段中输入要设置的值。

如果在 “Value:”（值:）字段中设置的表达式描述 有误，或指定的值超出指定寄存器允许设置的范围，则会出现错误。

7.16 [MR Trace] (MR 跟踪) 窗口

[MR Trace] (MR 跟踪) 窗口计量使用实时操作系统的程序的任务执行历史记录，并用图形显示结果。

除任务执行历史记录外，该窗口还可跟踪和显示各种其他操作的历史记录，包括中断处理、任务状态转换和系统调用发布。

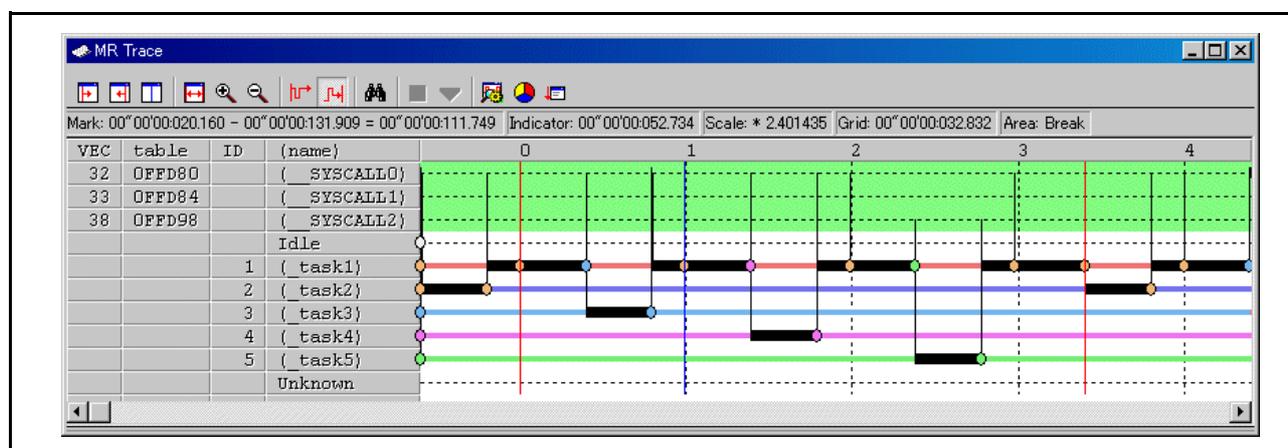
下载使用实时操作系统 (MRxx) 的目标程序后，才可使用此窗口。

对于 MR30

- 对于 MR30，此窗口可用于 V. 2.00 或更高版本。如果下载 MR30 V. 1.00 版本中的目标程序，则 [MR Trace] (MR 跟踪) 窗口不起作用且不显示任何数据。

对于 MR308

- 无法记录和显示高速中断的历史记录。

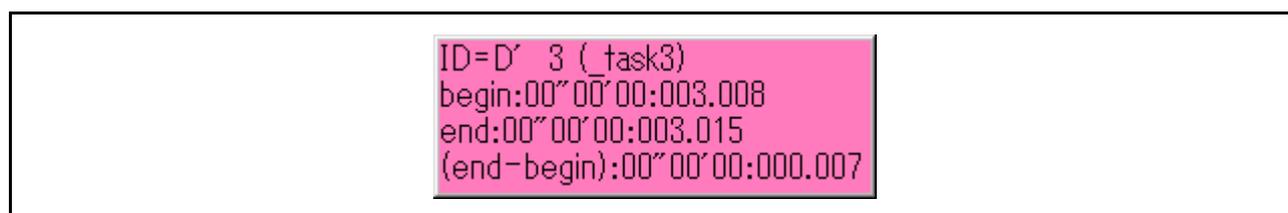


每个项目的内容如下。

项目	内容
[VEC]*1	表示软件中断编号。
[table] (表格)	表示中断向量表编号。
[ID]	表示任务 ID 号。
[(name)] ((名称))	表示中断程序名称、任务名称、空闲处理 (显示 “idle” (空闲))，以及未知名称 (显示 “unknown” (未知))。

将鼠标移动到窗口中显示的信息时，会打开以下弹出窗口，显示详细信息。

中断处理或任务执行历史记录



系统调用发出历史记录

```
rcv_msg
mbxid=D'1
E_OK
pk_msg(R1)=H'1234
pk_msg(R2)=H'5678
begin:00'00'00:002.861
```

任务状态转换历史记录

```
WAI(MBX)
begin:00'00'00:002.880
end:00'00'00:003.167
(end-begin):00'00'00:000.286
```

以下信息会显示在状态栏中。

- 位于开始标志位置的时间值
- 位于结束标志位置的时间值
- 开始标志和结束标志指示范围的时间宽度
- 位于指示器位置的时间值
- 显示的比例因子
- 网格线间隔的时间宽度
- 计量（跟踪）结果的范围

使用开始标志作为小数点显示网格线。

使用开始标志作为小数点显示网格线。显示比例，使用开始标志所在的时间作为 0，左边的（前面的时间）设置为“负”，右边的（后面的时间）设置为“正”。

网格线可让您粗略了解中断发生周期和处理时间。

显示的网格线的间隔时间宽度显示在状态栏的“Grid”（网格）区。

[MR Trace]（MR 跟踪）窗口中的时间值表示所有情况下使用程序执行开始时间作为 0 所经过的执行时间。相反，[MR Trace]（MR 跟踪）窗口中网格线（比例）上面的数值是使用开始标志作为 0 的相对值（网格间隔是在 [Value]（值）对话框中指定的）。

它与时间值无关。（提供此数值是为了让您能够轻松查看该窗口。）

注意

软件中断编号 *1 随产品的不同而不同。有关各系统调用分配的中断编号的详细信息，请参阅 MRxx Reference Manual（《MRxx 参考手册》）中的“Assemble Language Interface”（汇编语言接口）。

7.16.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Start Marker] (开始标志)		在显示区中移动开始标志。
[End Marker] (结束标志)		在显示区中移动结束标志。
[Indicator] (指示器)		在显示区中移动指示器。
[Adjust] (调整)		将开始和结束标志的范围调整到显示区调整的全部宽度。
[Expand] (放大)		放大显示的比例因子。
[Reduce] (缩小)		缩小显示的比例因子。
[Trace Stop] (跟踪停止)		停止计量。
[Trace Restart] (跟踪再启动)		再启动计量。
[Search...] (搜索...)		搜索系统调用历史记录。
[Trace Range] (跟踪范围)	[After] (之后)	将计量范围条件设置为“After”(之后)。
	[Break] (断点)	将计量范围条件设置为“Break”(断点)。
[Value...] (值...)		设置值。
[Color...] (颜色...)		更改显示颜色。
[Init Order...] (初始化顺序...)		复位任务显示顺序。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Allow Docking] (允许入坞)		隐藏窗口。

7.16.2 查看任务的执行历史记录 ([MRxx] 窗口)

可以在 [MR Trace] (MR 跟踪) 窗口中查看任务执行历史记录。可以在 [MR Analyze] (MR 分析) 窗口中查看执行历史记录统计处理结果。这些窗口对于使用实时操作系统 (MRxx) 的目标程序可用。

7.16.2.1 选择跟踪范围

要计量任务执行历史记录，可以使用实时跟踪功能。单击 [MR Trace] (MR 跟踪) 窗口中的 “After” (之后) 按钮 (或选择菜单 - [Trace Range] (跟踪范围) -> [After] (之后)) 或 “Break” (断点) 按钮 (或选择菜单 - [Trace Range] (跟踪范围) -> [Break] (断点))。

[After] (之后)	存储跟踪点之后的跟踪数据周期。
[Break] (断点)	存储跟踪点的通过点及其之前的周期。

执行目标程序。在跟踪存储器中，记录了解任务执行历史记录所需的信息。

注意事项

[Trace Point Setting] (跟踪点设置) 对话框中设置的跟踪点会被禁止。

7.16.2.2 停止任务执行历史记录计量

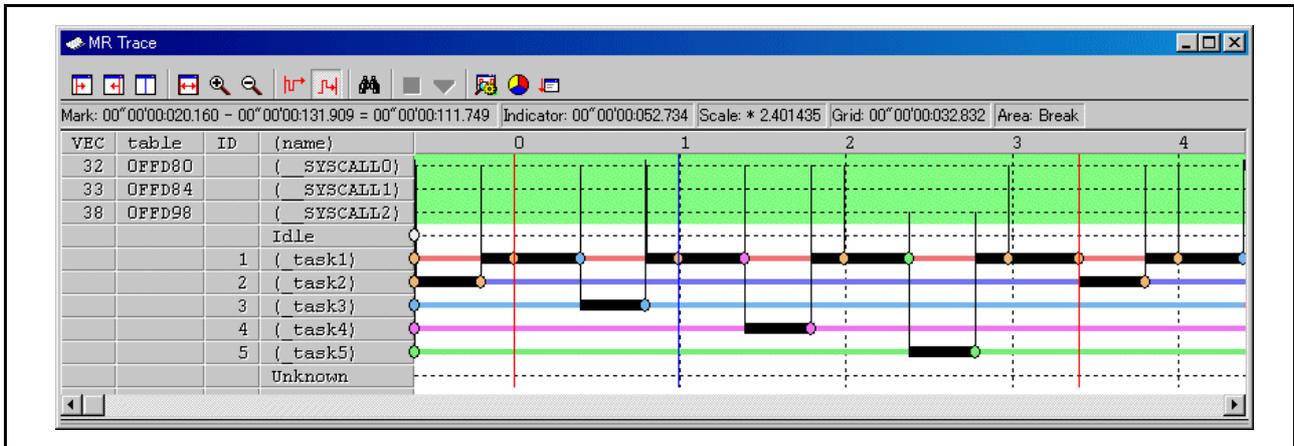
单击 [Task Trace] (任务跟踪) 窗口中的 “Stop” (停止) 按钮。(或者，选择菜单 - [Trace Stop] (跟踪停止)。) 迄今为止的计量结果显示在 [MR Trace] (MR 跟踪) 窗口中。

7.16.2.3 再启动任务执行历史记录计量

单击 [Task Trace] (任务跟踪) 窗口中的 “Restart” (再启动) 按钮。(或者，选择菜单 - [Trace Restart] (跟踪再启动)。) 再启动跟踪计量时，将删除迄今为止的所有计量结果。

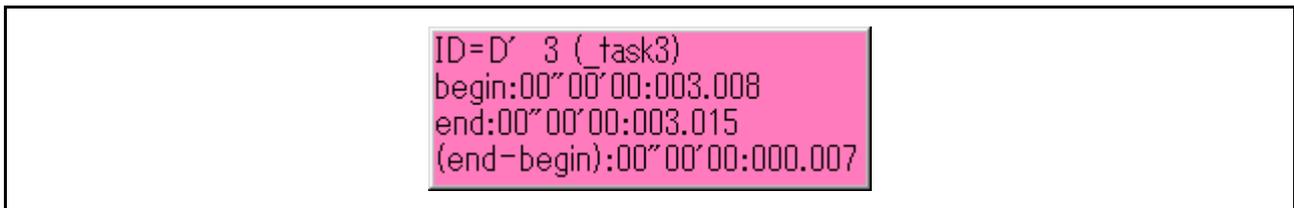
7.16.2.4 查看任务的执行历史记录

可以在 [MR Trace] (MR 跟踪) 窗口中查看任务执行转换。

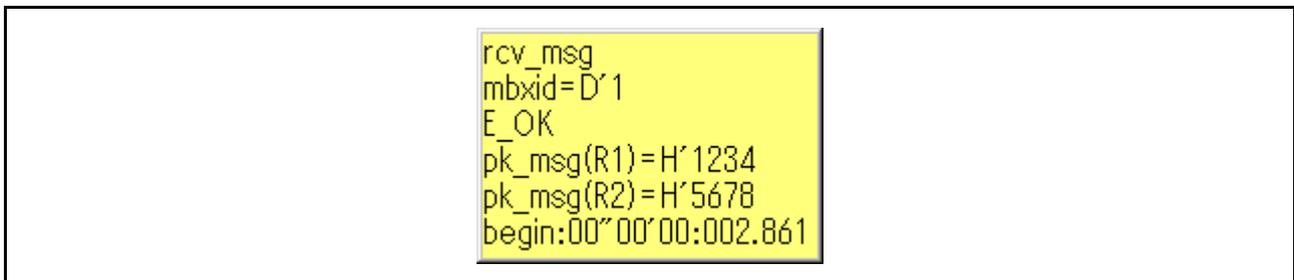


将鼠标移动到窗口中显示的任意信息上，会打开以下窗口，显示详细信息。

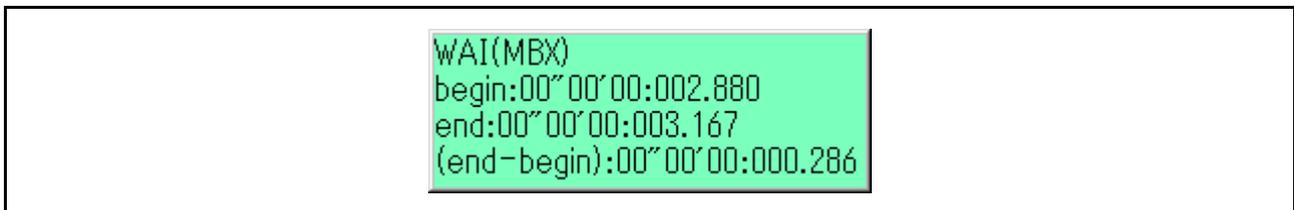
中断处理或任务执行历史记录



系统调用发出历史记录

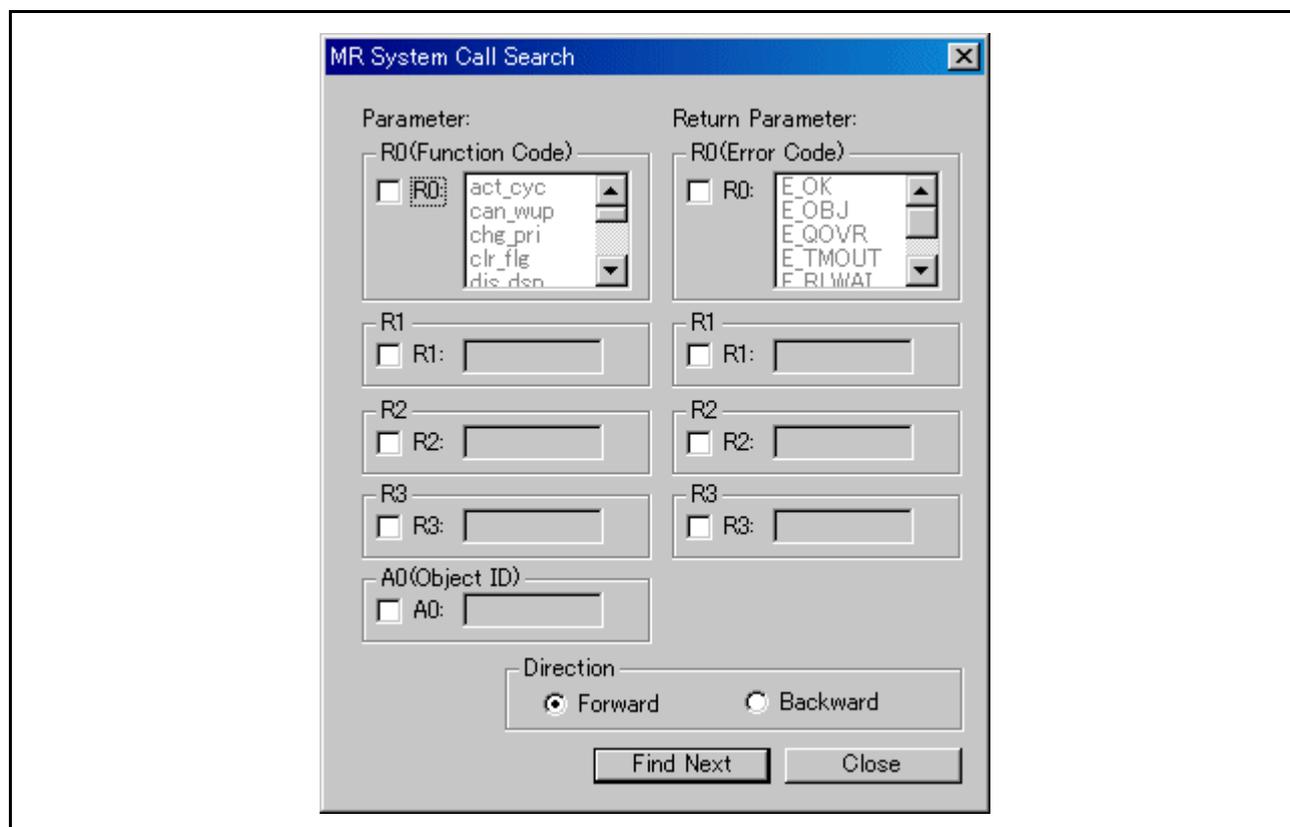


任务状态转换历史记录



7.16.2.4.1 搜索系统调用发出的历史记录

单击工具栏中的“Search”（搜索）按钮。此时会打开 [Search]（搜索）对话框。（或者，选择菜单 - [Search ...]（搜索...）。）



指定搜索条件。

使用功能代码（R0：功能代码）和错误代码（R0：错误代码），可以指定多个值（OR 条件）。其他项目根据 [AND]（与）条件搜索。

然后指定搜索方向。调试器使用指示器所指位置作为小数点，按照对话框中指定的方向搜索项目。

如果调试器不检查所有搜索项目，则搜索方向上随后的系统调用发布历史记录将作为搜索结果。单击 [Find Next]（查找下一个）按钮。调试器搜索与指定条件相对应的系统调用发布历史记录。使用 [AND]（与）条件搜索指定项目。

如果满足搜索条件，则将指示器移动到该点。

7.16.2.4.2 更改显示放大倍数

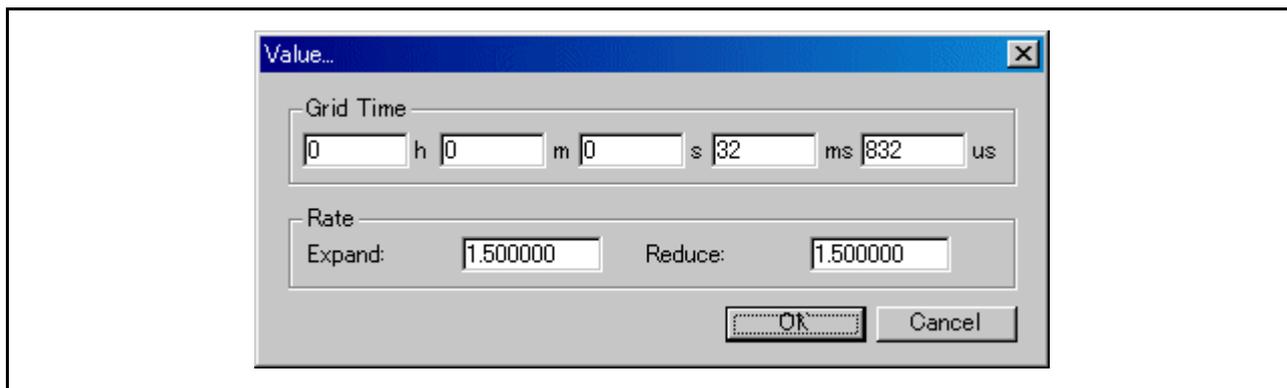
单击工具栏中的“Expand”（放大）按钮或“Reduce”（缩小）按钮。（或者，选择菜单 - [Expand]（放大）或 [Reduce]（缩小）。）

此时会使用图表区的左角作为小数点来放大或缩小显示内容。默认情况下，使用 1.5 的显示比例放大或缩小显示内容。

显示比例显示在状态栏的“Scale:*”（比例：*）字段中。

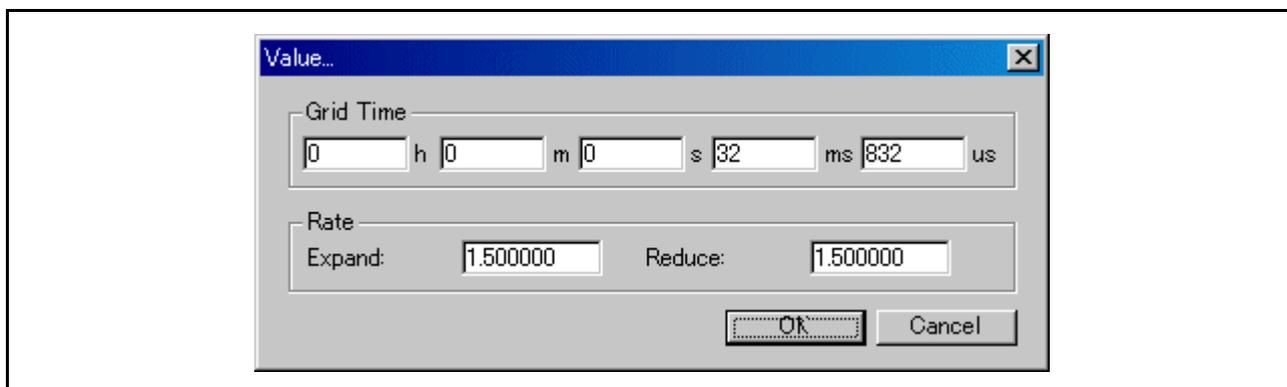
默认放大 / 缩小比例是 1.5。要更改比例，请选择菜单 - [Value ...]（值...）。

此时会打开 [Value]（值）对话框。指定显示放大 / 缩小比例。



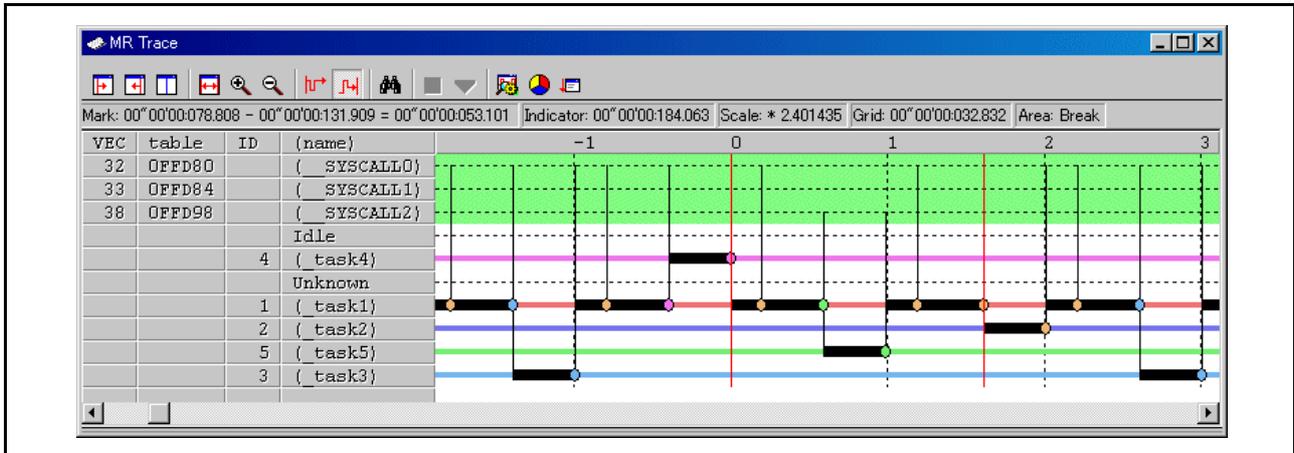
7.16.2.4.3 更改网格线显示间隔

选择菜单 - [Value ...]（值...）。此时会打开 [Value]（值）对话框。指定显示时间间隔。



7.16.2.4.4 更改任务显示顺序

将要移动的任务 / 中断程序（图形左边部分）拖动至目标位置。



要初始化显示顺序，请选择菜单 - [Init Order]（初始化顺序）。

7.16.2.4.5 仅显示指定任务

单击要隐藏的任务 / 中断程序（图形左边部分）。每次单击时，设置会在“Display”（显示）和“Hide”（隐藏）之间切换。

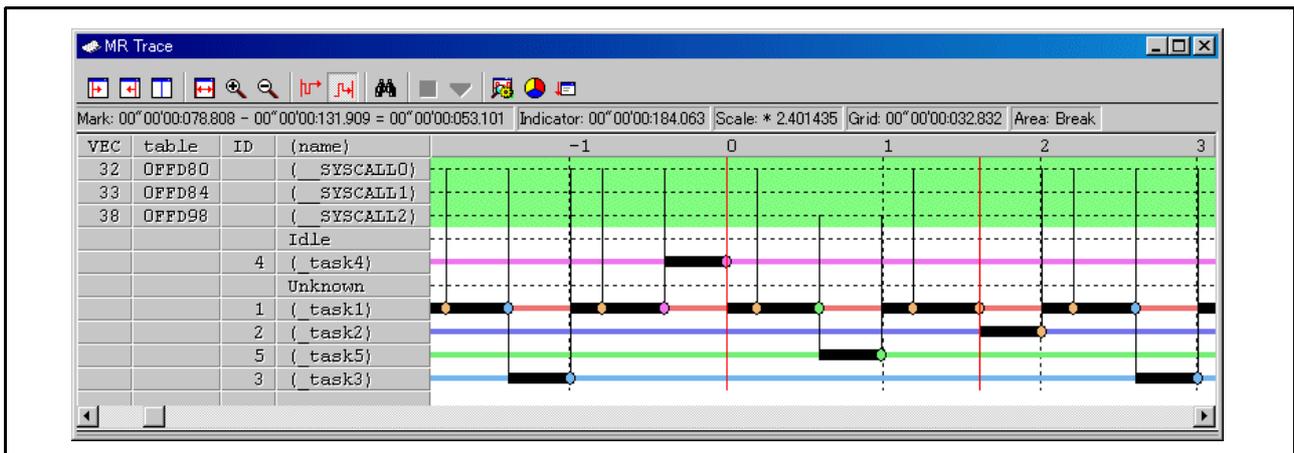
7.16.2.4.6 更改显示颜色

选择菜单 - [Color...]（颜色...）。此时会打开 [Color]（颜色）对话框。

单击与所需项目相对应的按钮。此时会打开 [Color Setting]（颜色设置）对话框。在对话框中更改显示颜色。

7.16.2.5 计量任务的执行时间

通过更改 [MR Trace]（MR 跟踪）窗口中开始标志和结束标志的位置，可以计量标志之间的执行时间。



拖动开始标志位置和结束标志位置。

标志之间的时间间隔会显示在状态栏中。

注意

[Definition of time value in the MR Trace window] (MR 跟踪窗口中时间值的定义)

[MR Trace] (MR 跟踪) 窗口中的时间值表示所有情况下将程序执行起始点设置为 0 所经过的执行时间。

相反, [MR Trace] (MR 跟踪) 窗口中网格线 (比例) 上面的数值是将开始标志设置为 0 的相对值 (网格间隔是在 [Value] (值) 对话框中指定的), 该值与时间值无关。 (提供此数值是为了让您能够轻松查看该窗口。)

7.16.2.5.1 移动标志

每个标志都可以通过拖动来移动。将鼠标移动到标志上时, 光标形状会更改。然后开始拖动。

单击工具栏中的 “Start Marker” (开始标志) 按钮, 开始标志会移动到窗口 (左边部分) 中。 (或者, 选择菜单 - [Start Marker] (开始标志)。)

单击 “End Marker” (结束标志) 按钮, 结束标志会移动到窗口 (右边部分) 中。 (或者, 选择菜单 - [End Marker] (结束标志)。)

单击 “Indicator” (指示器) 按钮, 指示器会移动到窗口 (中心) 中。 (或者, 选择菜单 - [Indicator] (指示器)。)

其他标志只能移动至下面列出的指定位置。

- 中断处理或任务执行转换到的位置
- 任务状态转换到的位置
- 显示系统调用发布历史记录的位置

7.17 [MR Analyze] (MR 分析) 窗口

[MR Analyze] (MR 分析) 窗口显示在 [MR Trace] (MR 跟踪) 窗口中开始标志和结束标志指定的范围内经过统计分析的计量数据的结果。

[MR Analyze] (MR 分析) 窗口支持如下三种显示模式：

- 根据中断/任务确定的 CPU 占用状态
- 根据任务确定的就绪时间
- 系统调用发布历史记录列表 (可以根据特定条件提取和显示历史记录。)

[MR Analyze] (MR 分析) 窗口与 [MR Trace] (MR 跟踪) 窗口搭配使用。

下载使用实时操作系统 (MRxx) 的目标程序后, 才可以使用此窗口。

7.17.1 配置 “CPU Occupancy Status Display Mode” (CPU 占用状态显示模式)

“CPU occupation state display mode” (CPU 占用状态显示模式) 用于显示中断 / 任务占用 CPU 的时间和比率。

[MR Trace] (MR 跟踪) 窗口显示开始标志和结束标志指定的范围内的统计结果。

VEC	table	ID	(name)	Num	Max Run Time	Min Run Time	Avg Run Time	Total Run Time	Ratio%	0	25	50	75	100
32	0FFD80		{ SYSCALL0 }	17	00"00"00:000.033	00"00"00:000.013	00"00"00:000.022	00"00"00:000.378	0.23					
33	0FFD84		{ SYSCALL1 }	5	00"00"00:000.020	00"00"00:000.019	00"00"00:000.019	00"00"00:000.099	0.06					
38	0FFD98		{ SYSCALL2 }	3	00"00"00:000.028	00"00"00:000.028	00"00"00:000.028	00"00"00:000.084	0.05					
			Idle	6	00"00"00:000.017	00"00"00:000.002	00"00"00:000.006	00"00"00:000.036	0.02					
		1	{ task1 }	11	00"00"00:014.003	00"00"00:000.001	00"00"00:008.957	00"00"00:098.528	60.02	█				
		2	{ task2 }	3	00"00"00:013.003	00"00"00:000.001	00"00"00:008.669	00"00"00:026.008	15.84	█				
		3	{ task3 }	2	00"00"00:013.006	00"00"00:000.003	00"00"00:006.504	00"00"00:013.009	7.92	█				
		4	{ task4 }	2	00"00"00:013.003	00"00"00:000.001	00"00"00:006.502	00"00"00:013.005	7.92	█				
		5	{ task5 }	2	00"00"00:013.007	00"00"00:000.003	00"00"00:006.505	00"00"00:013.011	7.93	█				
			Unknown	0	00"00"00:000.000	00"00"00:000.000	00"00"00:000.000	00"00"00:000.000	0.00					

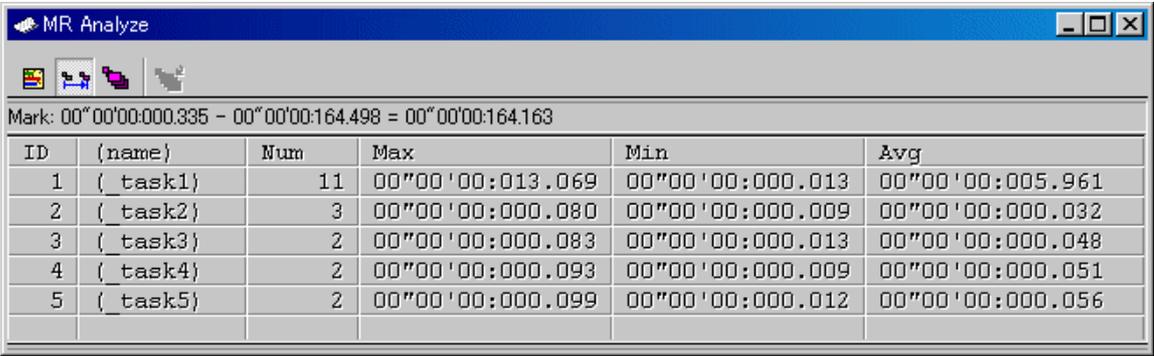
通过单击每行的最长执行时间 / 最短执行时间显示区, 可以搜索到单击行的中断或任务最长 / 最短执行时间的处理历史记录。

搜索结果由移动到 [MR Trace] (MR 跟踪) 窗口中目标位置的指示器指示。

7.17.2 配置 “Ready State Duration Display Mode”（就绪状态持续时间显示模式）

根据任务确定的 “ready state time display mode”（就绪状态时间显示模式）用于显示从执行就绪到转变到任务执行所需时间的统计处理所生成的结果。

统计结果显示在 [MR Trace]（MR 跟踪）窗口中开始标志和结束标志指定的范围内。



ID	(name)	Num	Max	Min	Avg
1	{_task1}	11	00"00'00:013.069	00"00'00:000.013	00"00'00:005.961
2	{_task2}	3	00"00'00:000.080	00"00'00:000.009	00"00'00:000.032
3	{_task3}	2	00"00'00:000.083	00"00'00:000.013	00"00'00:000.048
4	{_task4}	2	00"00'00:000.093	00"00'00:000.009	00"00'00:000.051
5	{_task5}	2	00"00'00:000.099	00"00'00:000.012	00"00'00:000.056

通过单击所需行的最长就绪时间 / 最短就绪时间显示区，可以搜索单击行对应任务的最长就绪时间 / 最短就绪时间的处理历史记录。

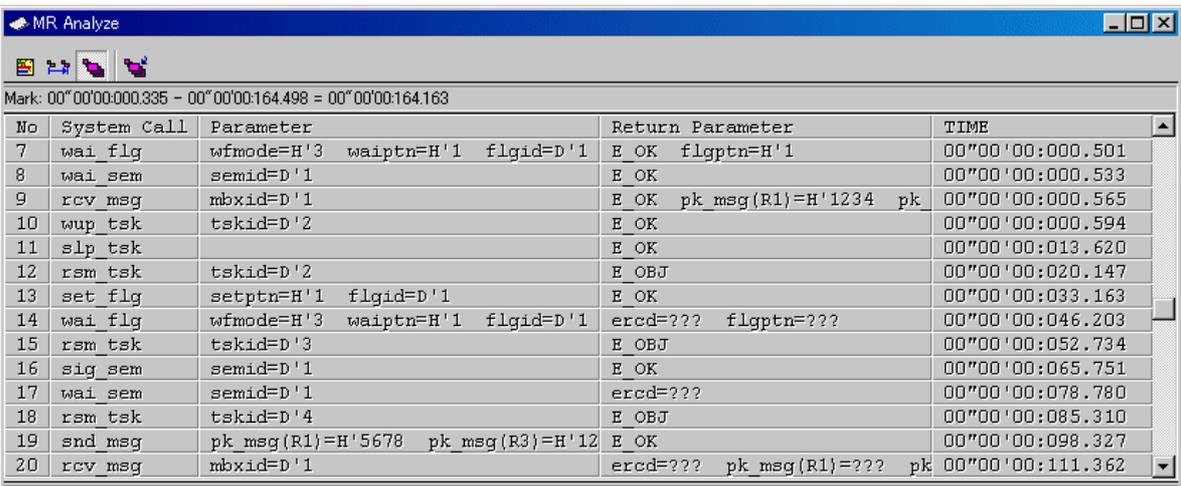
搜索结果由移动到 [MR Trace]（MR 跟踪）窗口中目标位置的指示器指示。

7.17.3 配置 “System Call History Display Mode”（系统调用历史记录显示模式）

“system call issuance history list mode”（系统调用发布历史记录列表模式）用于列出发出的系统调用。

系统调用发布历史记录列在 [MR Trace]（MR 跟踪）窗口中开始标志和结束标志指定的范围内。

数字表示在可计量范围内从顶部的系统调用的计数值。



No	System Call	Parameter	Return Parameter	TIME
7	wai_flg	wfmode=H'3 waiptn=H'1 flgid=D'1	E_OK flgpfn=H'1	00"00'00:000.501
8	wai_sem	semid=D'1	E_OK	00"00'00:000.533
9	rcv_msg	mbxid=D'1	E_OK pk_msg(R1)=H'1234 pk	00"00'00:000.565
10	wup_tsk	tskid=D'2	E_OK	00"00'00:000.594
11	slp_tsk		E_OK	00"00'00:013.620
12	rsm_tsk	tskid=D'2	E_OBJ	00"00'00:020.147
13	set_flg	setpfn=H'1 flgid=D'1	E_OK	00"00'00:033.163
14	wai_flg	wfmode=H'3 waiptn=H'1 flgid=D'1	ercd=??? flgpfn=???	00"00'00:046.203
15	rsm_tsk	tskid=D'3	E_OBJ	00"00'00:052.734
16	sig_sem	semid=D'1	E_OK	00"00'00:065.751
17	wai_sem	semid=D'1	ercd=???	00"00'00:078.780
18	rsm_tsk	tskid=D'4	E_OBJ	00"00'00:085.310
19	snd_msg	pk_msg(R1)=H'5678 pk_msg(R3)=H'12	E_OK	00"00'00:098.327
20	rcv_msg	mbxid=D'1	ercd=??? pk_msg(R1)=??? pk	00"00'00:111.362

通过单击所需的行，可以搜索到单击行的系统调用发布历史记录。

搜索结果由移动到 [MR Trace]（MR 跟踪）窗口中目标位置的指示器指示。

7.17.4 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能
[Run Time] (运行时)	“CPU occupancy status display mode” (CPU 占用状态显示模式)。
[Rdy->Run] (就绪 -> 运行)	“Ready state duration display mode” (就绪状态持续时间显示模式)。
[System Call] (系统调用)	“System call history display mode” (系统调用历史记录显示模式)。
[Pick Up System Call...] (提取系统调用...)	提取指定的 “system calls display mode” (系统调用显示模式)。
[Toolbar display] (工具栏显示)	显示工具栏。
[Customize toolbar...] (自定义工具栏...)	打开工具栏自定义对话框。
[Allow Docking] (允许入坞)	允许窗口入坞。
[Hide] (隐藏)	隐藏窗口。

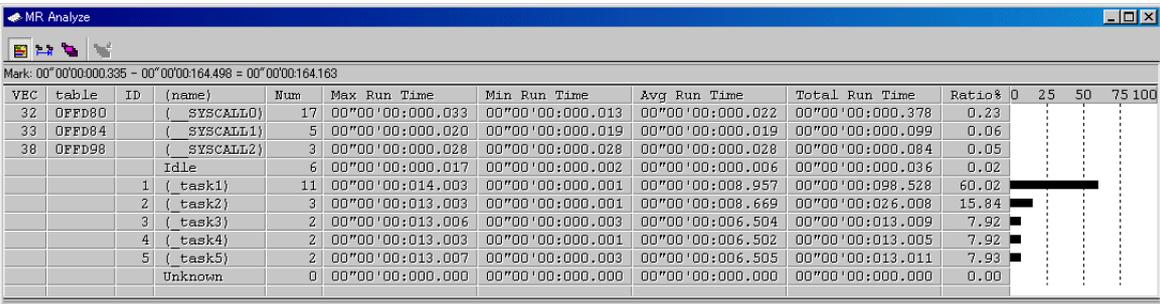
7.17.5 分析任务的执行历史记录

可以在 [MR Analyze] (MR 分析) 窗口中查看执行历史记录统计处理。[MR Analyze] (MR 分析) 窗口与 [MR Trace] (MR 跟踪) 窗口搭配使用。如果 [MR Trace] (MR 跟踪) 窗口未打开，或者 [MR Trace] (MR 跟踪) 窗口未显示任何数据，则 [MR Analyze] (MR 分析) 窗口将无法发挥作用。

执行历史记录统计处理功能允许您查看以下主题。

7.17.5.1 查看 CPU 占用状态

单击工具栏中的 [Run Time] (运行时) 按钮。(或者，选择 [Menu] (菜单) - [Run Time] (运行时)。) [MR Analyze] (MR 分析) 窗口更改为 “CPU occupation state display mode” (CPU 占用状态显示模式)。



VEC	table	ID	(name)	Num	Max Run Time	Min Run Time	Avg Run Time	Total Run Time	Ratio%
32	OFFD80		(SYSCALL0)	17	00"00'00:000.033	00"00'00:000.013	00"00'00:000.022	00"00'00:000.378	0.23
33	OFFD84		(SYSCALL1)	5	00"00'00:000.020	00"00'00:000.019	00"00'00:000.019	00"00'00:000.099	0.06
38	OFFD98		(SYSCALL2)	3	00"00'00:000.028	00"00'00:000.028	00"00'00:000.028	00"00'00:000.084	0.05
			Idle	6	00"00'00:000.017	00"00'00:000.002	00"00'00:000.006	00"00'00:000.036	0.02
		1	(task1)	11	00"00'00:014.003	00"00'00:000.001	00"00'00:008.957	00"00'00:098.528	60.02
		2	(task2)	3	00"00'00:013.003	00"00'00:000.001	00"00'00:008.669	00"00'00:026.008	15.84
		3	(task3)	2	00"00'00:013.006	00"00'00:000.003	00"00'00:006.504	00"00'00:013.009	7.92
		4	(task4)	2	00"00'00:013.003	00"00'00:000.001	00"00'00:006.502	00"00'00:013.005	7.92
			(task5)	2	00"00'00:013.007	00"00'00:000.003	00"00'00:006.505	00"00'00:013.011	7.93
			Unknown	0	00"00'00:000.000	00"00'00:000.000	00"00'00:000.000	00"00'00:000.000	0.00

窗口显示中断处理和任务占用的 CPU 时间和比率。

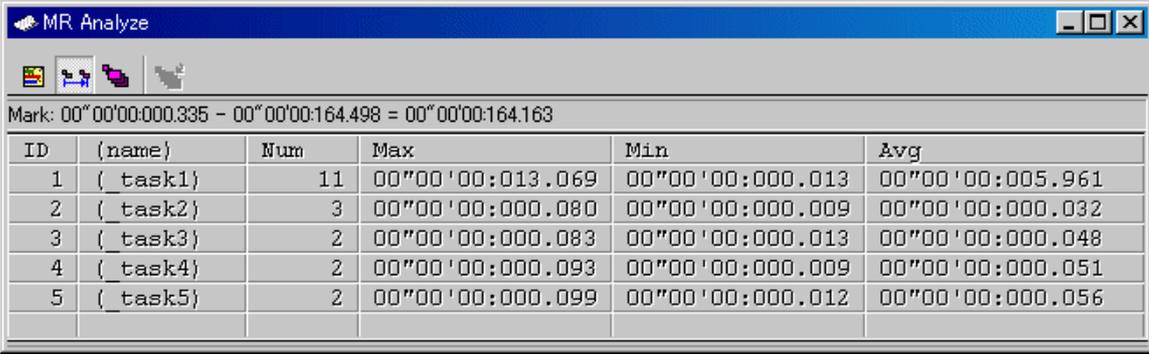
显示的数据是 [MR Trace] (MR 跟踪) 窗口中开始标志和结束标志指定范围的统计结果。

通过单击每行的最长执行时间 / 最短执行时间显示字段，可以搜索单击行对应任务的最长执行时间 / 最短执行时间的处理历史记录。

在 [MR Trace] (MR 跟踪) 窗口中的指示器移动到目标位置以后，搜索结果由该指示器指示。

7.17.5.2 查看就绪队列时间

单击工具栏中的 [Ready->Run] (就绪 -> 运行) 按钮。(或者, 选择 [Menu] (菜单) - [Rdy -> Run] (就绪 -> 运行)。)



Mark: 00"00'00:000.335 - 00"00'00:164.498 = 00"00'00:164.163

ID	(name)	Num	Max	Min	Avg
1	{ task1}	11	00"00'00:013.069	00"00'00:000.013	00"00'00:005.961
2	{ task2}	3	00"00'00:000.080	00"00'00:000.009	00"00'00:000.032
3	{ task3}	2	00"00'00:000.083	00"00'00:000.013	00"00'00:000.048
4	{ task4}	2	00"00'00:000.093	00"00'00:000.009	00"00'00:000.051
5	{ task5}	2	00"00'00:000.099	00"00'00:000.012	00"00'00:000.056

从执行就绪状态到任务执行状态所需的转变时间会经过统计处理并显示出来。

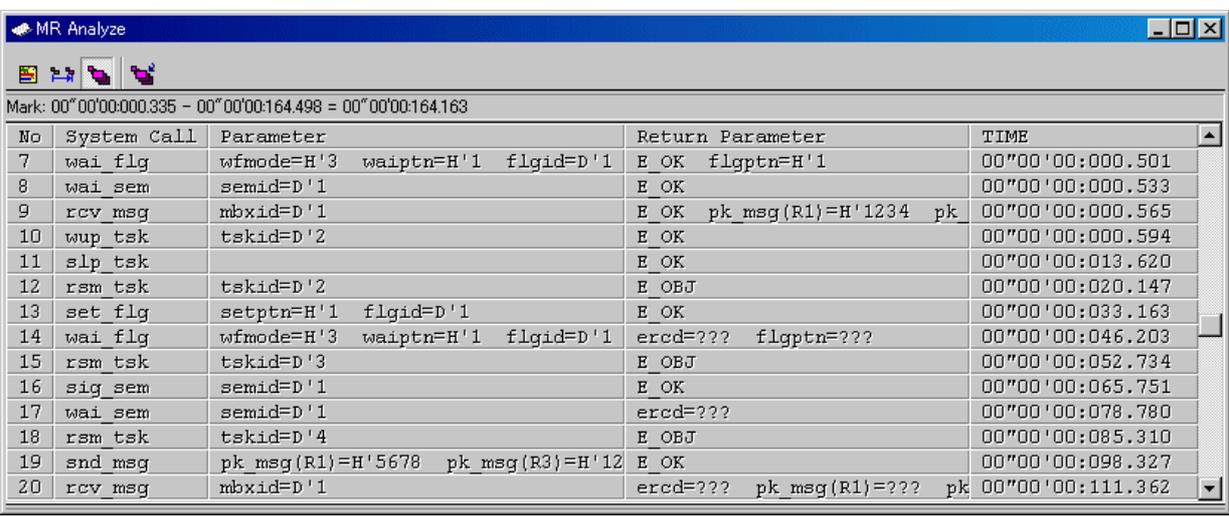
显示的数据是 [MR Trace] (MR 跟踪) 窗口中开始标志和结束标志指定范围的统计结果。

通过单击每行的最长就绪时间 / 最短就绪时间显示字段, 可以搜索单击行对应任务的最长就绪时间 / 最短就绪时间中被处理过的历史记录。

在 [MR Trace] (MR 跟踪) 窗口中的指示器移动到目标位置以后, 搜索结果由该指示器指示。

7.17.5.3 查看系统调用发布历史记录

单击工具栏中的 “System Call” (系统调用) 按钮。(或者, 选择 [Menu] (菜单) - [System Call] (系统调用)。)



Mark: 00"00'00:000.335 - 00"00'00:164.498 = 00"00'00:164.163

No	System Call	Parameter	Return Parameter	TIME
7	wai_flg	wfmode=H'3 waiptn=H'1 flgid=D'1	E_OK flgpfn=H'1	00"00'00:000.501
8	wai_sem	semid=D'1	E_OK	00"00'00:000.533
9	rcv_msg	mbxid=D'1	E_OK pk_msg(R1)=H'1234 pk	00"00'00:000.565
10	wup_tsk	tskid=D'2	E_OK	00"00'00:000.594
11	slp_tsk		E_OK	00"00'00:013.620
12	rsm_tsk	tskid=D'2	E_OBJ	00"00'00:020.147
13	set_flg	setpfn=H'1 flgid=D'1	E_OK	00"00'00:033.163
14	wai_flg	wfmode=H'3 waiptn=H'1 flgid=D'1	ercd=??? flgpfn=???	00"00'00:046.203
15	rsm_tsk	tskid=D'3	E_OBJ	00"00'00:052.734
16	sig_sem	semid=D'1	E_OK	00"00'00:065.751
17	wai_sem	semid=D'1	ercd=???	00"00'00:078.780
18	rsm_tsk	tskid=D'4	E_OBJ	00"00'00:085.310
19	snd_msg	pk_msg(R1)=H'5678 pk_msg(R3)=H'12	E_OK	00"00'00:098.327
20	rcv_msg	mbxid=D'1	ercd=??? pk_msg(R1)=??? pk	00"00'00:111.362

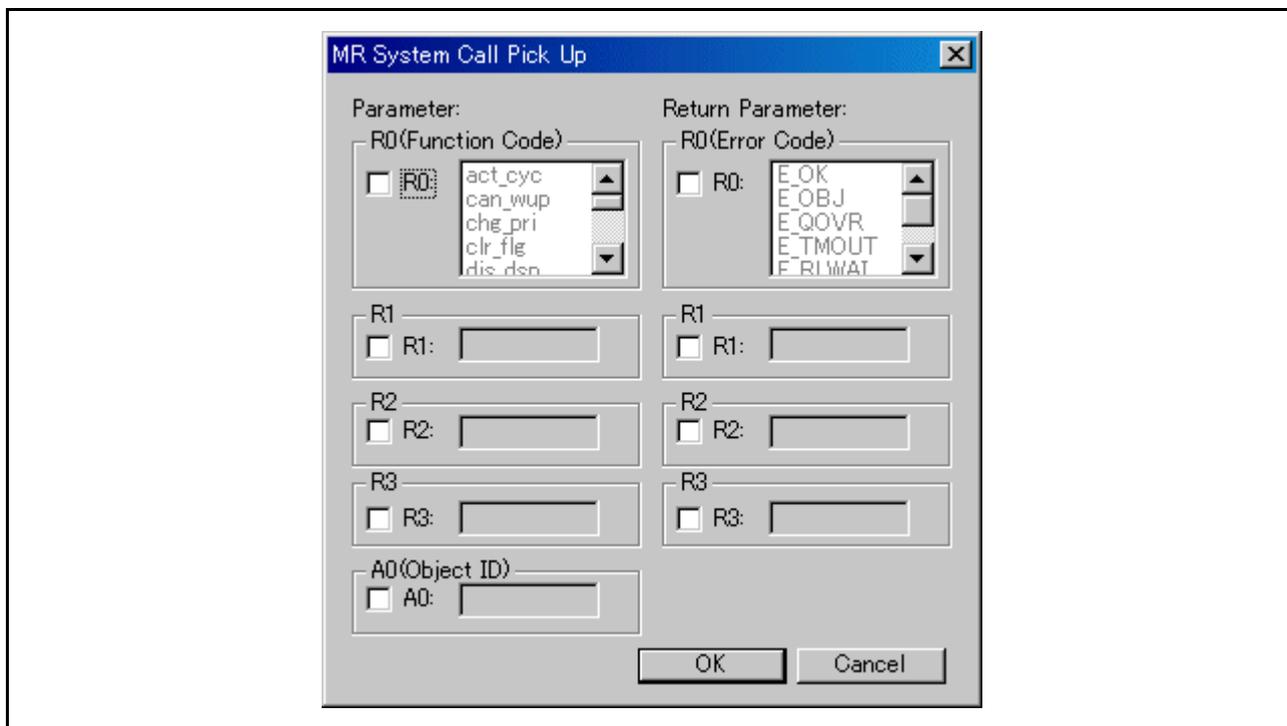
发出的系统调用会以系统调用的时间顺序列出来。

显示的数据是 [MR Trace] (MR 跟踪) 窗口中开始标志和结束标志指定范围的统计结果。

通过单击每一行, 可以搜索单击行对应的系统调用发布历史记录。在 [MR Trace] (MR 跟踪) 窗口中的指示器移动到目标位置以后, 搜索结果由该指示器指示。

7.17.5.3.1 提取发布历史记录

单击工具栏中的“Pick Up”（提取）按钮。（或者，选择菜单 - [Pick Up System Call...]（提取系统调用...）。）
此时会打开下面显示的对话框。指定要提取和显示的系统调用的搜索条件。



提取并显示满足指定条件的系统调用的发布历史记录。

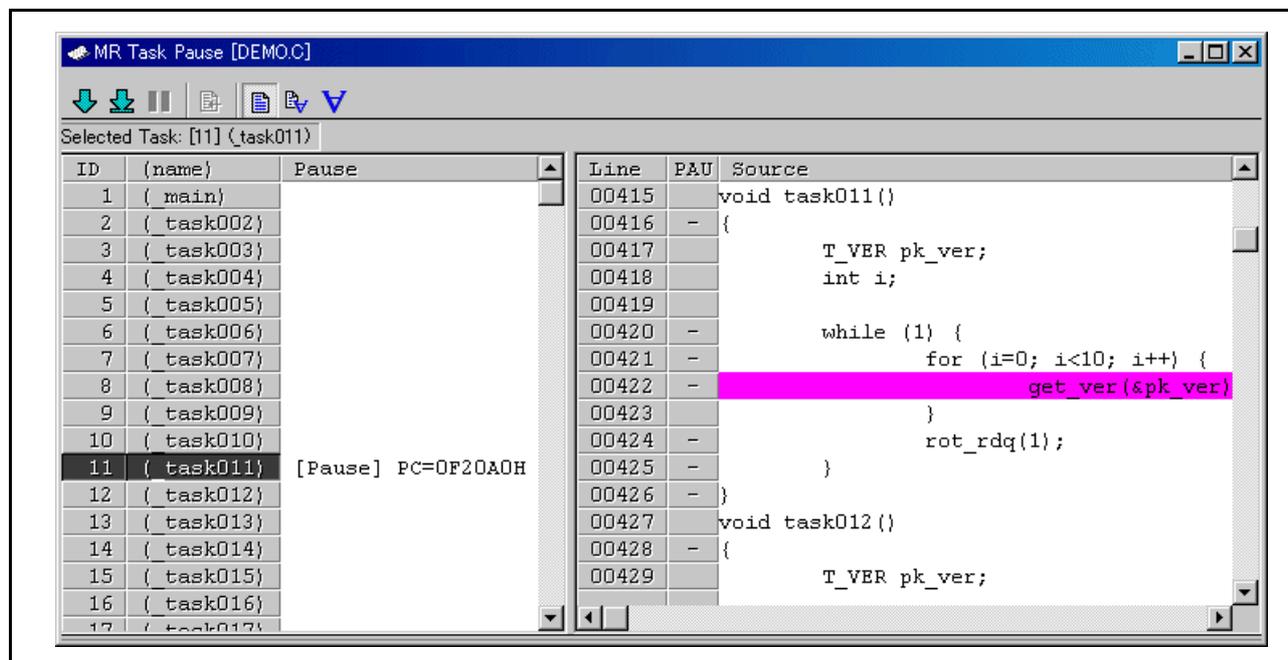
7.18 [MR Task Pause] (MR 任务暂停) 窗口

[MR Task Pause] (MR 任务暂停) 窗口实现实时操作系统 (MR308/MR30) 的任务暂停功能。

可以从此窗口暂停和取消暂停指定任务。

只有在下载了包含具有任务暂停功能和系统时钟定时器的 MR308/MR30 系统的程序以后, 才可以使用该窗口。

MR30 任务暂停功能受 MR30 V.3.00 或更高版本支持。以前安装的目标程序无法与此窗口一起使用。



- 当创建目标程序时在配置文件中定义的所有任务的信息 (ID 号、名称、“暂停”期间的上下文 PC 值) 显示在任务暂停显示区。从此显示区选择任务暂停的目标任务。
- 指定程序内容显示在任务源显示区。当执行“指定位置-暂停”时, 请在此显示区使用光标指定停止位置。

7.18.1 关于任务暂停功能

任务暂停功能只能在执行目标系统时暂停 / 取消暂停特定任务。

当使用任务暂停功能时, 可以在执行所有其他任务和中断时指定特定任务。

同时, 当调试功能可用时, 例如“指定位置 - 暂停”, 可以进行有效的调试, 而不影响由任务或中断控制的外围设备。

下面列出了本节使用的术语定义。

- 暂停状态
表示任务暂停功能在目标程序执行期间停止的任务的状态。
- 暂停
表示使特定任务处于暂停状态的处理。
- 取消暂停
表示使特定任务脱离暂停状态的处理。
- 指定位置-暂停
表示将特定任务暂停在指定地址的处理。

注意事项

任务暂停功能使用 MCU 的地址匹配中断。

当目标程序使用地址匹配中断时，请勿使用此功能。

如果使用 PC7501 仿真器，则当使用地址匹配中断断点功能时，无法使用此功能。

7.18.2 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单	功能	
[Task] (任务)	[Pause] (暂停)	使选择的任务处于暂停状态。
	[Come-pause] (指定位置 - 暂停)	在执行程序显示视图中光标处的代码之前，使选择的任务处于暂停状态。
	[Unpause] (取消暂停)	使选择的任务脱离暂停状态。
[Display] (显示)	[Source/Function...] (源 / 函数...)	显示指定文件 / 函数的开头。
	[Address...] (地址...)	从指定地址显示程序。
	[Program Counter] (程序计数器) *	从当前程序计数器值显示程序。
[Mode] (模式)	[Source] (源)	将视图切换为“source mode” (源模式)。
	[Mixed] (混合)	将视图切换为“mixed mode” (混合模式)。
	[Disassembly] (反汇编)	将视图切换为“disassembly mode” (反汇编模式)。
[Layout] (布局)	[Line] (行)	显示 / 隐藏行号列。
	[Address] (地址)	显示 / 隐藏地址列。
	[Code] (代码)	显示 / 隐藏行代码列。
[Toolbar display] (工具栏显示)		显示工具栏。
[Customize toolbar...] (自定义工具栏...)		打开工具栏自定义对话框。
[Allow Docking] (允许入坞)		允许窗口入坞。
[Hide] (隐藏)		隐藏窗口。

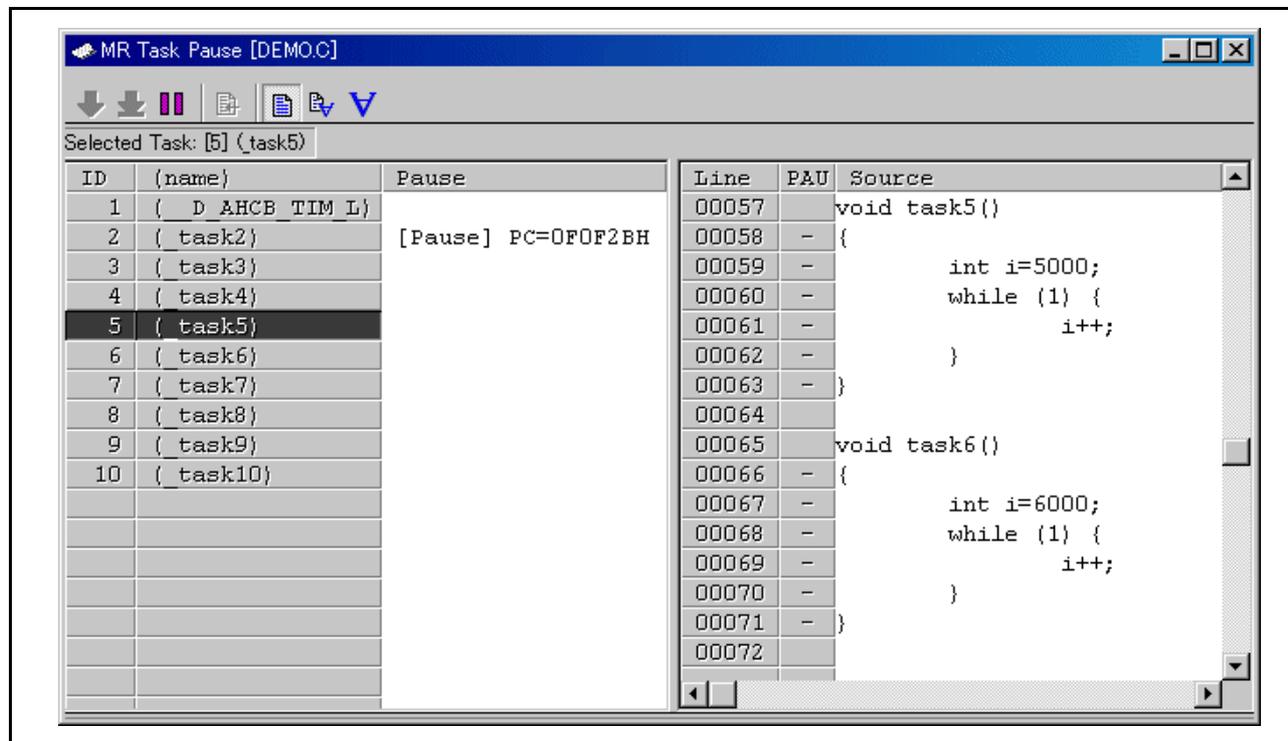
* 当通过指定 PC 位置更改程序显示位置时，MR 任务暂停窗口中的操作如下。

- 当任务暂停显示区中选择的目标任务暂停时
 - 它的显示位置会更改为任务的上下文 PC 位置。
- 当任务暂停显示区中选择的目标任务处于非“暂停”状态时
 - 显示位置不会更改。
- 当从任务暂停显示区选择目标任务时
 - 显示位置会更改为来自当前程序计数器的位置 (与 [Source] (源) 窗口中的操作方式相同)。

7.18.3 暂停指定任务

使用 [MR Task Pause] (MR 任务暂停) 窗口。

若要打开 [MR Task Pause] (MR 任务暂停) 窗口, 请选择菜单 - [View] (视图) -> [RTOS] -> [MR Task Pause] (MR 任务暂停)。



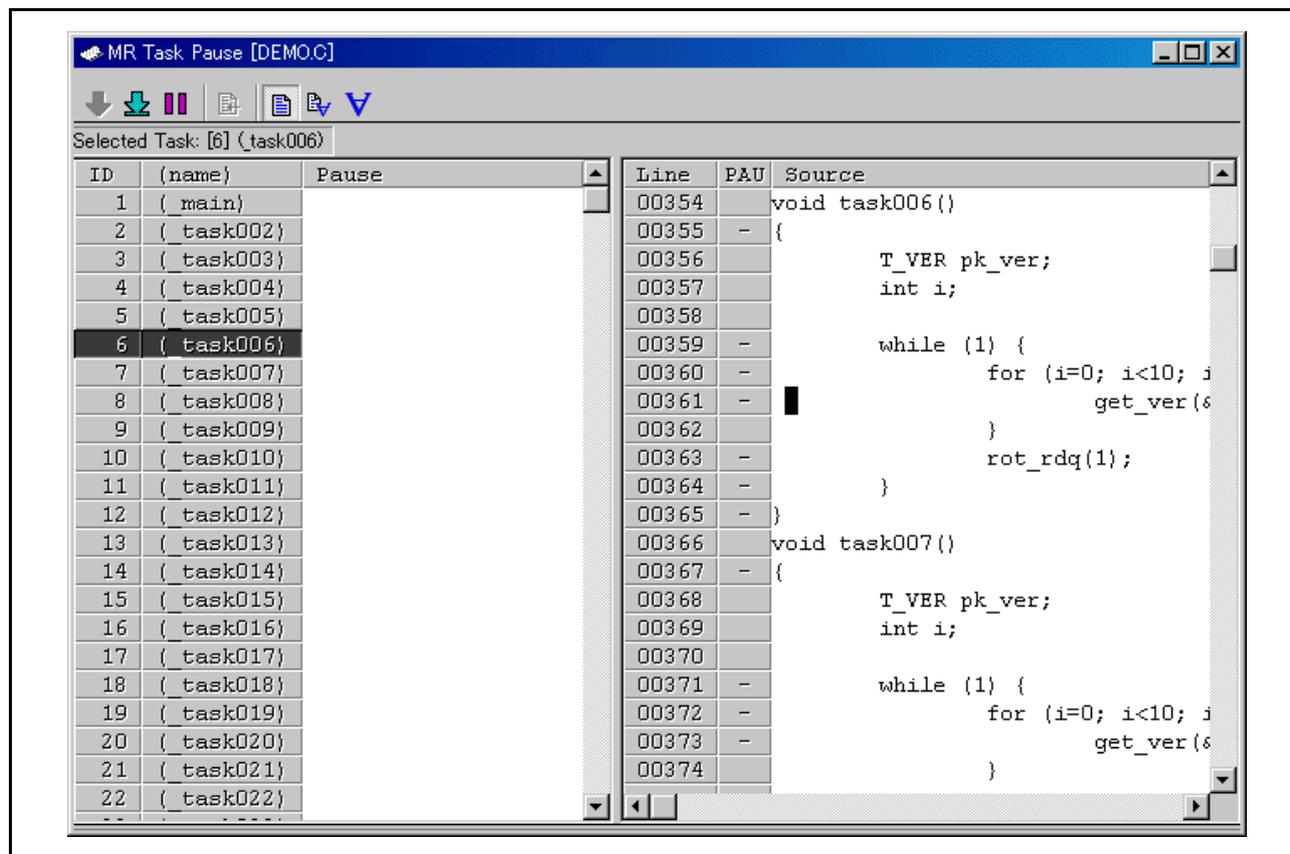
1. 在要暂停的任务行的任务暂停显示字段中, 单击 [ID] 字段或 [(name)] ((名称)) 字段。
选择目标任务以后, 任务 ID 号会显示在状态栏中。
2. 如果选择的任务处于非“暂停”状态, 则允许使用 [Pause] (暂停) 按钮。
单击 [Pause] (暂停) 按钮。在此时选择的任务会暂停。(只有在选择的任务处于执行状态时, 才允许这样做。)

注意事项

只能暂停处于执行状态的目标任务。如果目标任务处于非执行状态, 则暂停进程会被终止, 但目标任务不会被暂停。(如果发生这种情况, 会出现错误对话框。)

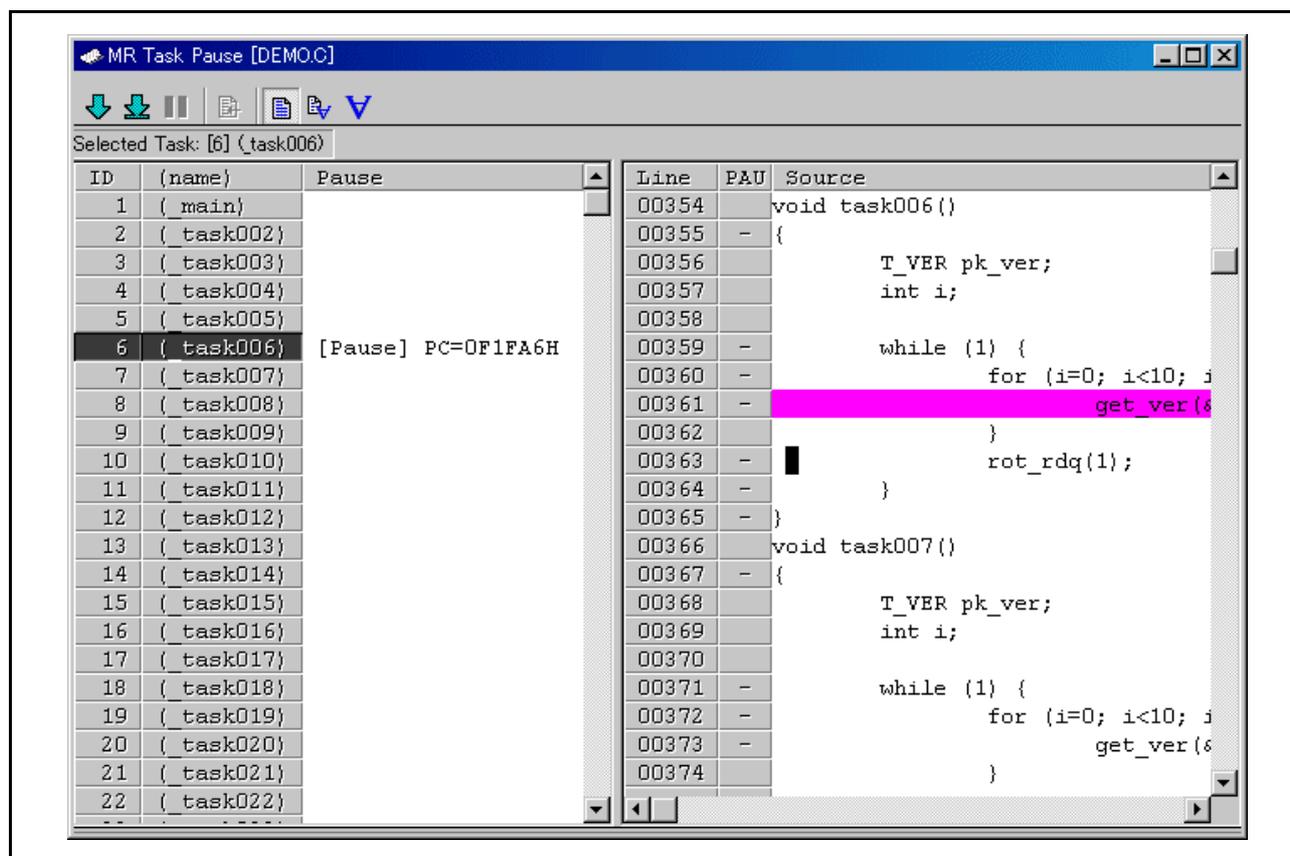
如果目标任务进入执行状态的时间很难判断, 则使用任务的“指定位置 - 暂停”(一种仅在指定位置暂停特定任务的功能)。

7.18.3.1 在指定位置暂停正在执行的任务



1. 在要暂停的任务行的任务暂停显示字段中，单击 [ID] 字段或 [(name)] ((名称)) 字段。
选择目标任务以后，任务 ID 号会出现在状态栏中。
在任务源显示区中，会显示从任务起始地址开始的程序（如果该程序处于非暂停状态）。
2. 在任务源显示区中，单击要暂停的行。允许使用 [Come-pause]（指定位置-暂停）按钮和 [Pause]（暂停）按钮。
3. 单击 [Come-pause]（指定位置-暂停）按钮。选择的任务会在光标位置之前执行任务的位置暂停。

7.18.3.2 在暂停任务执行到指定位置以后暂停



1. 在目标任务的任务暂停显示区中，单击 [ID] 字段或 [(name)] ((名称)) 字段。
选择目标任务以后，任务 ID 号会显示在状态栏中。
在任务源显示区中，会显示从任务暂停位置（上下文 PC 位置）开始的程序（如果该程序处于非暂停状态）。
2. 在任务源显示区中，单击要暂停的行。允许使用 [Come-pause]（指定位置-暂停）按钮和 [Pause]（暂停）按钮。
3. 单击 [Come-pause]（指定位置-暂停）按钮。选择的任务会暂时从暂停状态中解除。任务会执行到光标之前的位置，然后再次暂停。

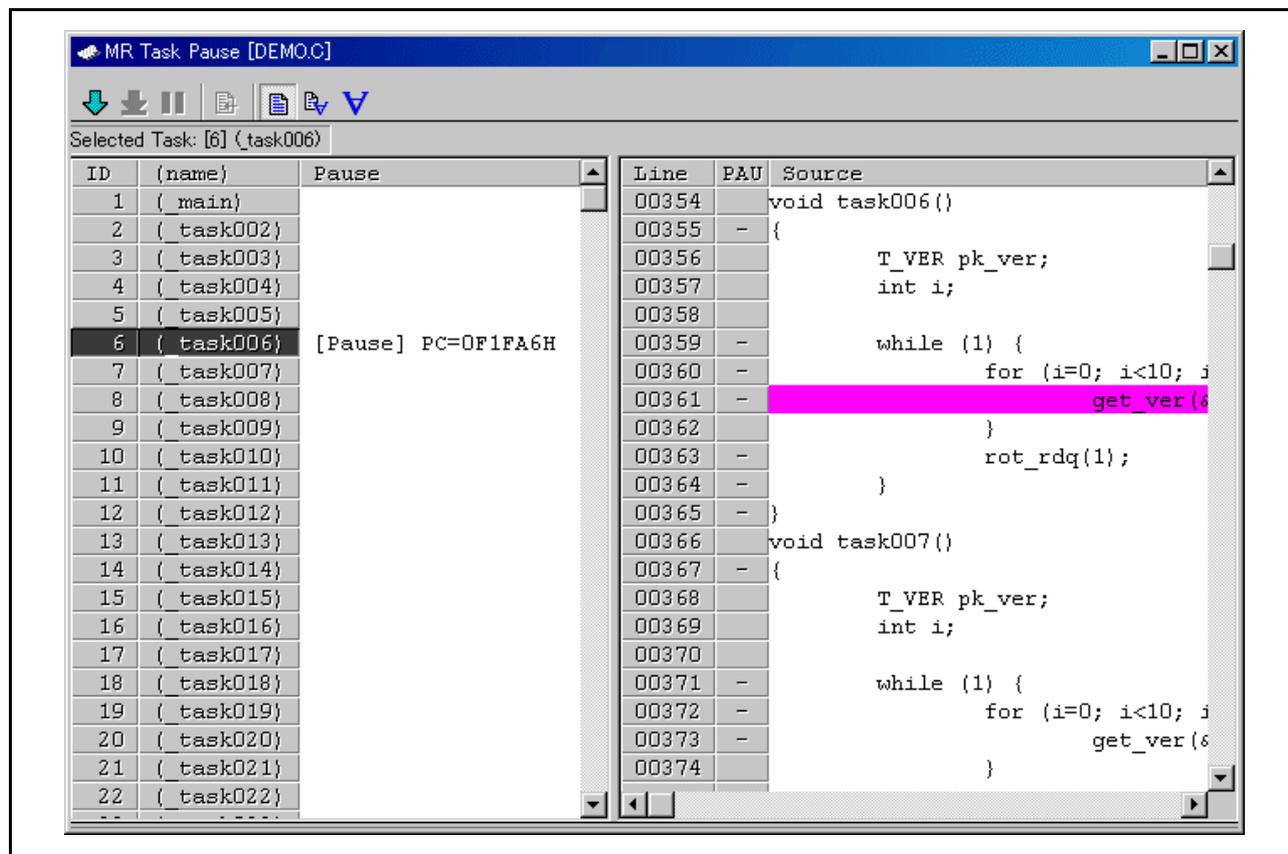
注意事项

当执行“指定位置 - 暂停”时，请务必在目标任务执行过的位置指定光标位置。

如果光标不是设置在上面所述的位置，而是设置在不适当的位置，则系统不会正常运作，而且在某些情况下，命令进程可能永远也不会终止。（由于调试器无法判断光标位置是否正确，因此它只会继续等待，直到目标任务执行光标位置。）

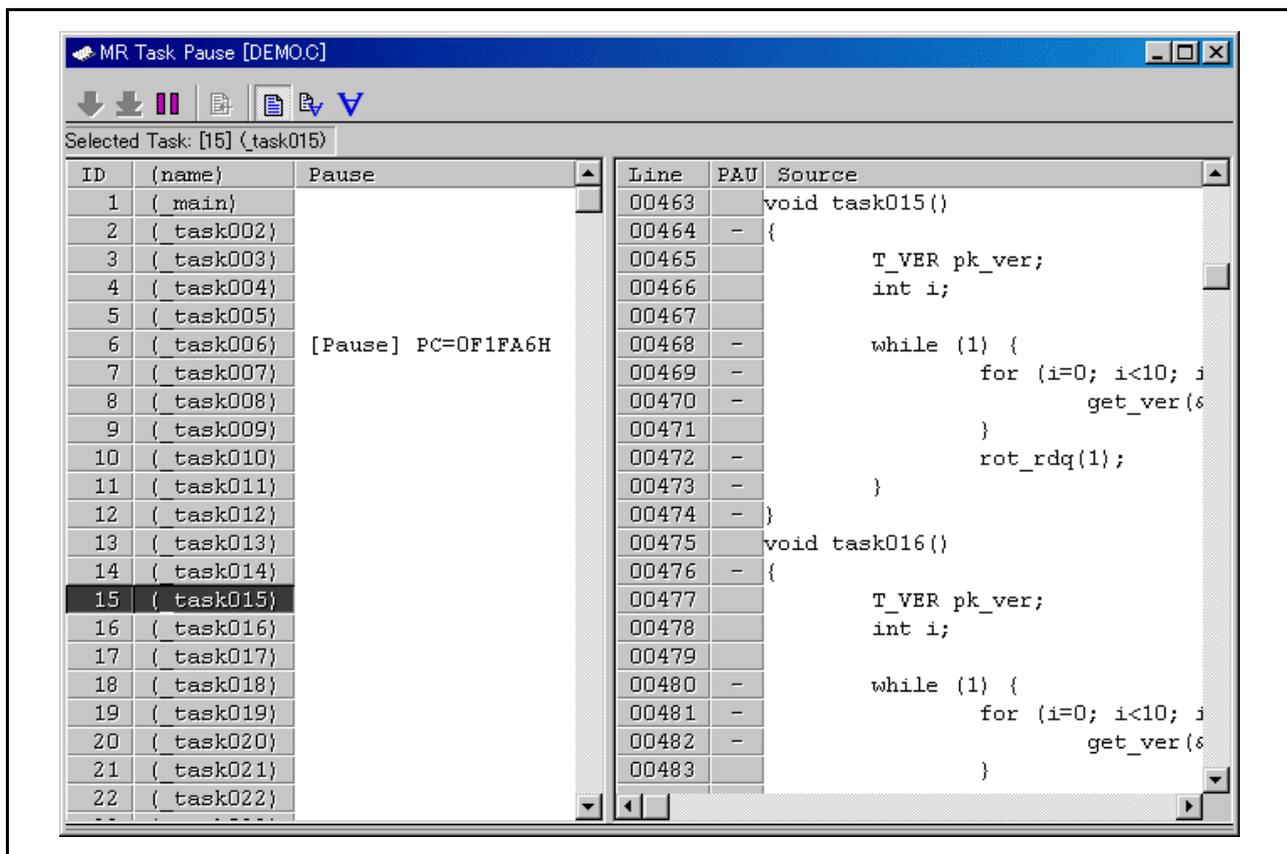
如果发生这种情况，请单击随后在“指定位置 - 暂停”执行过程中出现的对话框中的 [Stop]（停止）按钮，以停止命令处理。

7.18.3.3 取消暂停某个任务



1. 要从暂停状态中解除，须在相关任务行的任务暂停显示区中单击 [ID] 字段或 [(name)] ((名称)) 字段。选择目标任务以后，任务 ID 号会显示在状态栏中。允许使用 [Unpause] (取消暂停) 按钮 (如果任务已暂停)。
2. 单击 [Unpause] (取消暂停) 按钮。选择的任务会从暂停状态中解除。

7.18.3.4 在任务源显示区显示某个任务的程序内容



在要显示的任务行的任务暂停显示区中，单击 [ID] 字段或 [(name)] ((名称)) 字段。

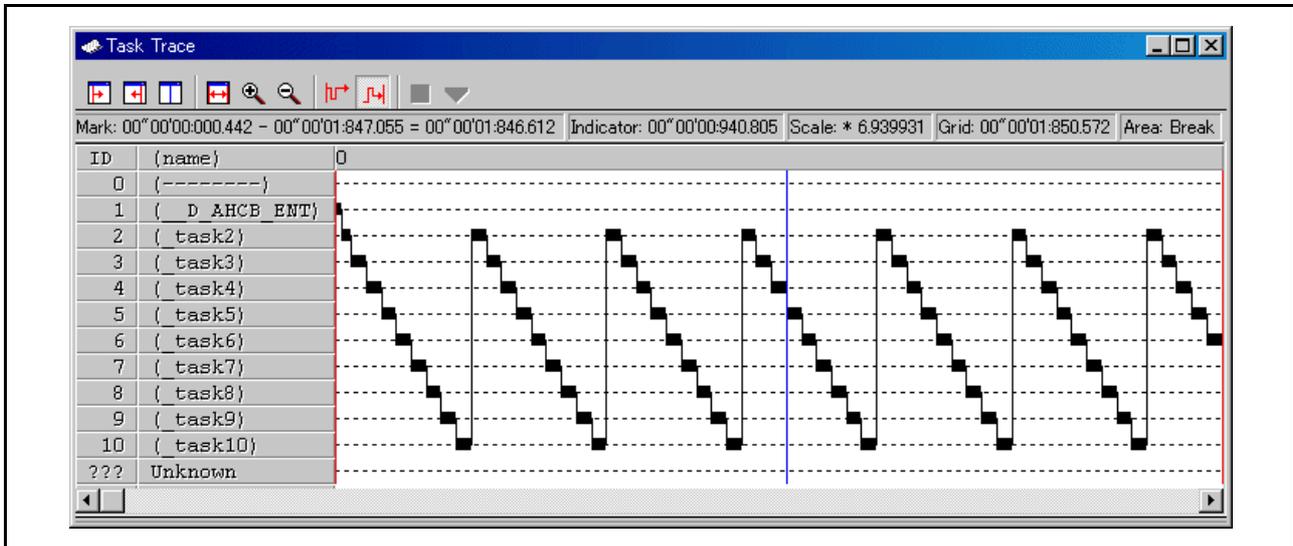
如果目标任务已暂停，则会显示任务暂停位置 (上下文 PC 位置)。

如果目标任务处于非暂停状态，则会显示任务起始地址。

也可以使用 [View] (视图) 按钮和 [View] (视图) 菜单来显示任何位置。

7.19 [Task Trace]（任务跟踪）窗口

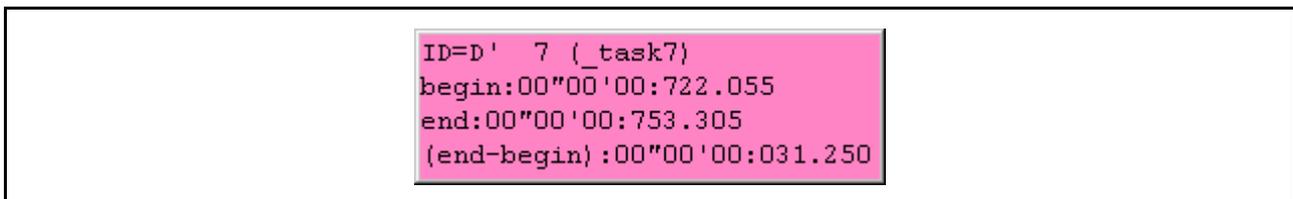
[Task Trace]（任务跟踪）窗口计量使用实时操作系统的程序的任务执行历史记录，并用图形显示出来。即使下载的目标程序使用的是实时操作系统 (MRxx) 以外的操作系统，也可以使用此窗口。



每个项目的内容如下。

项目	内容
[ID]	表示任务 ID 号。
[(name)] ((名称))	表示中断程序名称、任务名称、空闲处理（显示“idle”（空闲）），以及未知名称（显示“unknown”（未知））。

将鼠标移动到窗口中显示的信息时，会打开以下弹出窗口，显示详细信息。



以下信息会显示在状态栏中。

- 位于开始标志位置的时间值
- 位于结束标志位置的时间值
- 开始标志和结束标志之间的时间间隔
- 位于指示器位置的时间值
- 显示比例
- 网格线间隔的时间宽度
- 计量（跟踪）范围

使用开始标志作为小数点显示网格线。

显示比例，使用开始标志所在的时间作为 0，左边的（前面的时间）设置为“负”，右边的（后面的时间）设置为“正”。

网格线可让您粗略了解中断发生周期和处理时间。

显示的网格线的间隔时间宽度显示在状态栏的“Grid”（网格）区。

[Task Trace]（任务跟踪）窗口中的时间值表示所有情况下使用程序执行开始时间作为 0 所经过的执行时间。

相反，[Task Trace]（任务跟踪）窗口中网格线（比例）上面的数值是使用开始标志作为 0 的相对值（网格间隔是在 [Value]（值）对话框中指定的）。它与时间值无关。（提供此数值是为了让您能够轻松查看该窗口。）

7.19.1 扩展菜单

此窗口具有如下弹出菜单，在窗口中右键单击便可看到。

菜单		功能
[Start Marker]（开始标志）		在显示区中移动开始标志。
[End Marker]（结束标志）		在显示区中移动结束标志。
[Indicator]（指示器）		在显示区中移动指示器。
[Adjust]（调整）		将开始和结束标志的范围调整到显示区调整的全部宽度。
[Expand]（放大）		增大显示的比例因子。
[Reduce]（缩小）		减小显示的比例因子。
[Trace Stop]（跟踪停止）		停止计量。
[Trace Restart]（跟踪再启动）		再启动计量。
[Trace Range] （跟踪范围）	[After]（之后）	将计量范围条件设置为“After”（之后）。
	[Break]（断点）	将计量范围条件设置为“Break”（断点）。
[Value...]（值...）		设置值。
[Color...]（颜色...）		更改显示颜色。
[RTOS...]		设置目标 RTOS 信息。
[Toolbar display]（工具栏显示）		显示工具栏。
[Customize toolbar...]（自定义工具栏...）		打开工具栏自定义对话框。
[Allow Docking]（允许入坞）		允许窗口入坞。
[Hide]（隐藏）		隐藏窗口。

7.19.2 查看任务的执行历史记录 ([Taskxx] 窗口)

可以在 [Task Trace] (任务跟踪) 窗口中查看任务执行历史记录。

可以在 [Task Analyze] (任务分析) 窗口中查看执行历史记录统计处理结果。

即使目标程序使用的是实时操作系统 (MRxx) 之外的操作系统, 也可以使用这些窗口。

7.19.2.1 准备计量

若要计量使用实时操作系统的程序的任务执行历史记录, 必须在 [Task Trace] (任务跟踪) 窗口中选择跟踪范围, 然后执行目标程序。

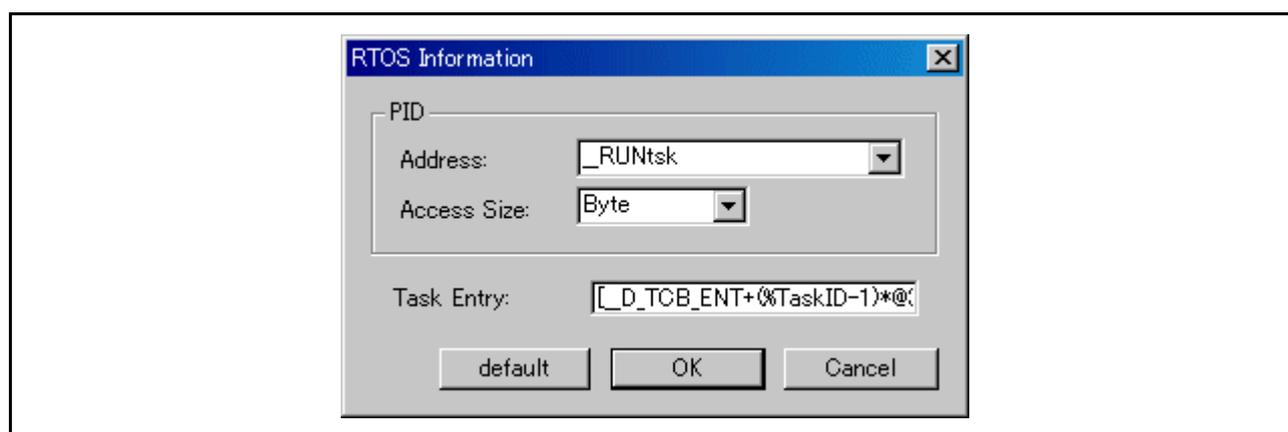
7.19.2.1.1 设置实时操作系统信息。

若要使用 [Task Trace] (任务跟踪) 窗口, 必须设置下面关于已下载程序所使用的实时操作系统 (目标实时操作系统) 的信息。

- 执行任务 ID 存储区的标签名称 (地址值) 及其大小
- 任务起始地址表达式

打开 [Task Trace] (任务跟踪) 窗口。选择菜单 - [View] (视图) -> [RTOS] -> [Task Trace] (任务跟踪)。

如果在启动 PDxx 以后第一次选择此菜单, 则在打开 [Task Trace] (任务跟踪) 窗口之前, 会打开 [RTOS Information] (RTOS 信息) 对话框。



- 使用实时操作系统 (MRxx) 时
 1. 单击 “default” (默认) 按钮。设置 MRxx 信息。
 2. 单击 [OK] (确定) 按钮。此时会打开 [Task Trace] (任务跟踪) 窗口。
- 使用非 MRxx 的实时操作系统时
 1. 在 [PID Address] (PID 地址) 字段中, 指定执行任务 ID 存储区的标签 (也可以使用地址); 在 [Size] (大小) 列表框中, 指定执行任务 ID 存储区的大小。
如果未正确设置此信息, 则无法使用 [Task Trace] (任务跟踪) 窗口。
 2. 在 [Task Entry] (任务入口) 字段中, 指定任务起始地址表达式。
使用与描述规则一致的格式描述表达式。
在假设要分配任务 ID 号的地址中使用宏变量 [% TaskID]。如果未正确设置此信息, 则任务名称不会显示在 [Task Trace] (任务跟踪) 窗口中。
 3. 单击 [OK] (确定) 按钮。此时会打开 [Task Trace] (任务跟踪) 窗口。

在 740 调试器上，只需单击默认按钮，即可设置 OSEK 操作系统信息。

一旦在此对话框中设置了实时操作系统信息，该信息就会从下次开始生效。

若要更改设置数据，请右击窗口，从弹出菜单中选择 [RTOS...]。此时会重新打开 [RTOS Information] (RTOS 信息) 对话框。

注意事项

在 [RTOS Information] (RTOS 信息) 对话框中进行 PID 设置时，如果在存取大小中指定 [WORD] (字)，则必须遵守以下限制。（如果不满足这些条件，则系统不会正常运作。）

- PID 信息存储区已分配到偶数地址。
- PID 信息存储区已分配到一个使用 16 位总线宽度存取的区域。

7.19.2.1.2 选择跟踪范围

实时跟踪功能用于任务执行历史记录计量。

在 [Task Trace] (任务跟踪) 窗口中，单击 [After] (之后) 按钮（或者右击窗口，从弹出菜单中选择 [After] (之后)）或 [Break] (断点) 按钮（或者从弹出菜单中选择 [Break] (断点)）。

[After] (之后)	记录任务执行历史记录，直到记录数据填满跟踪存储器为止。
[Break] (断点)	记录任务执行历史记录（最多可以达到可用跟踪存储器的容量），直到目标程序停止之前为止。

在跟踪存储器中，只需通过一个特定周期，便可了解其中的任务执行历史记录。

注意事项

[Trace Point Setting] (跟踪点设置) 对话框中设置的跟踪点会被禁止。

7.19.2.1.3 启动目标程序

执行目标程序。在跟踪存储器中，记录了解任务执行历史记录所需的信息。

为跟踪范围选择 [After] (之后) 时，执行历史记录会在填满跟踪存储器或目标程序停止之后，立即显示在 [Task Trace] (任务跟踪) 窗口中。

为跟踪范围选择 [Break] (断点) 时，执行历史记录会在目标程序停止之后，立即显示在 [Task Trace] (任务跟踪) 窗口中。

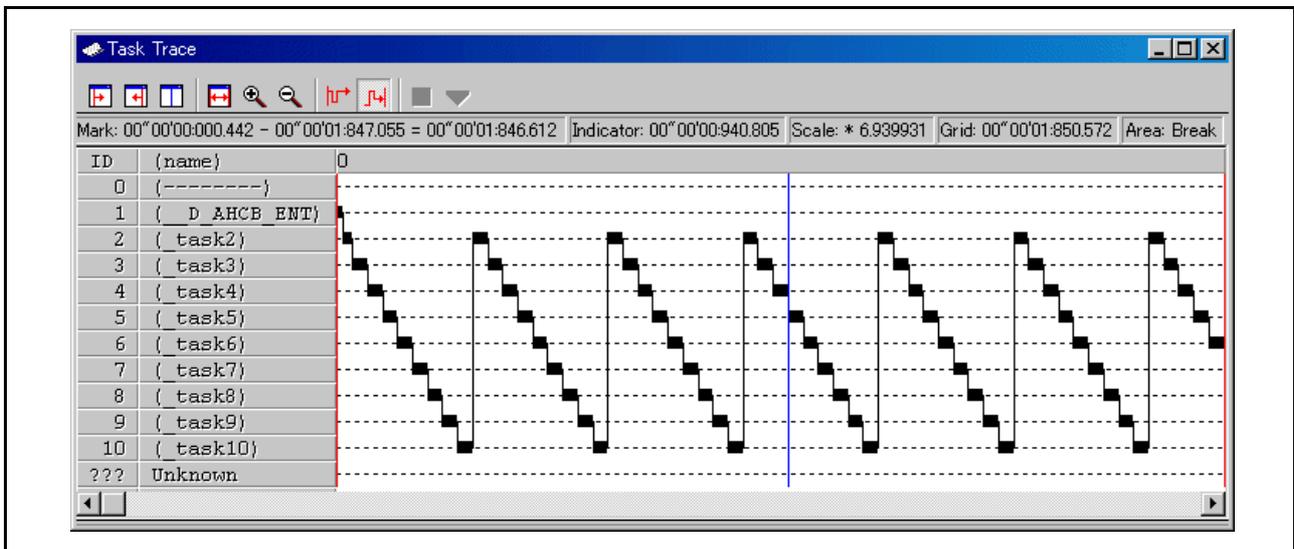
可以停止任务执行历史记录计量。

若要这样做，请在 [Task Trace] (任务跟踪) 窗口中单击 [Stop] (停止) 按钮。（或者，右击窗口，从弹出菜单中选择 [Trace Stop] (跟踪停止)。）

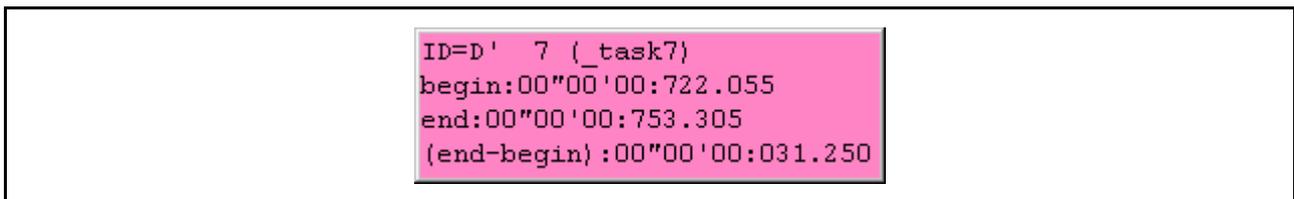
若要再启动任务执行历史记录计量，请在 [Task Trace] (任务跟踪) 窗口中单击 [Restart] (再启动) 按钮。（或者，右击窗口，从弹出菜单中选择 [Trace Restart] (跟踪再启动)。）

7.19.2.2 查看任务的执行历史记录

可以在 [Task Trace]（任务跟踪）窗口中查看任务跟踪转移情况。



将鼠标移动到窗口中显示的信息上，会打开以下示例窗口，显示详细信息。



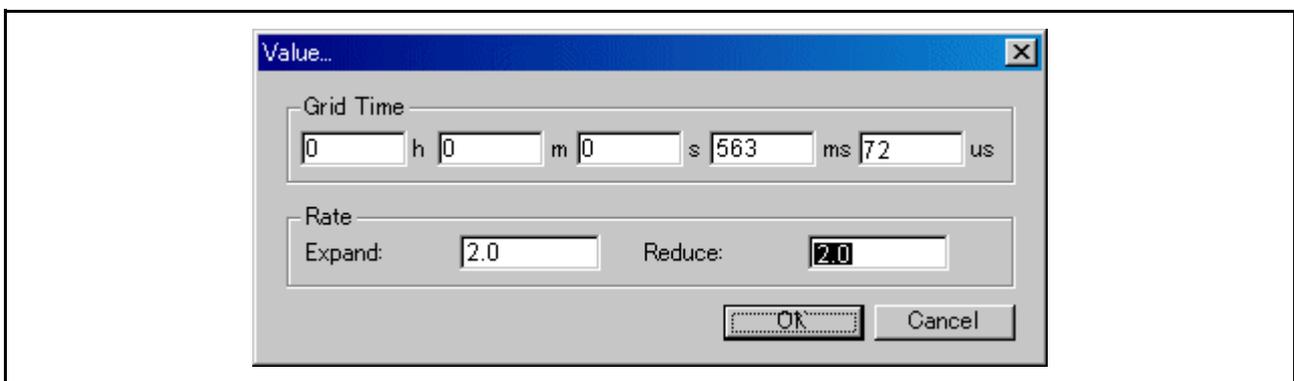
7.19.2.2.1 更改显示放大倍数

单击工具栏中的 [Expand]（放大）按钮或 [Reduce]（缩小）按钮。（或者，右击窗口，从弹出菜单中选择 [Expand]（放大）或 [Reduce]（缩小）。）

此时会使用图表区的左角作为小数点来放大或缩小显示内容。默认情况下，使用 1.5 的显示比例放大或缩小显示内容。

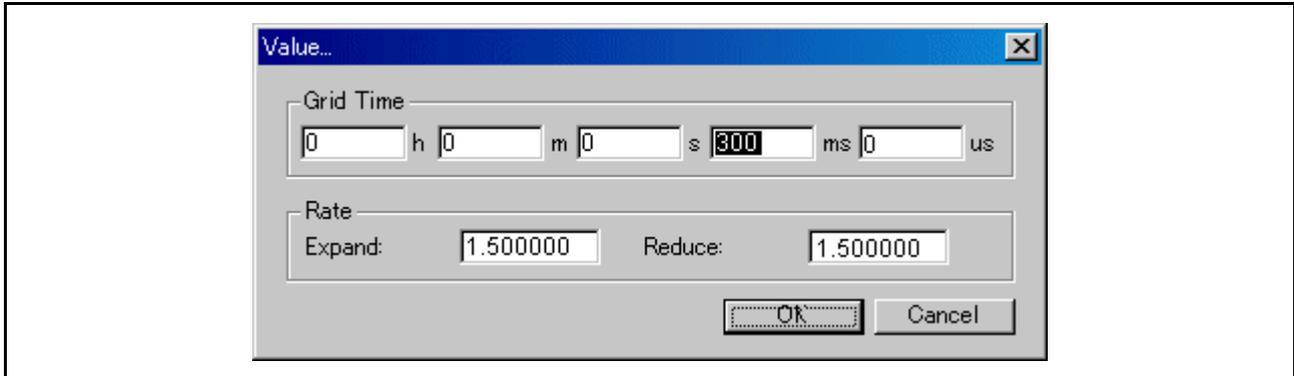
显示比例显示在状态栏的“Scale:*”（比例：*）字段中。

默认放大 / 缩小比例是 1.5。若要更改比例，请右击窗口，从弹出菜单中选择 [Value ...]（值...）。此时会打开 [Value]（值）对话框。指定显示放大 / 缩小比例。



7.19.2.3 更改网格线显示间隔

右击窗口，从弹出菜单中选择 [Value ...]（值...）。此时会打开 [Value]（值）对话框。指定显示中的时间间隔。



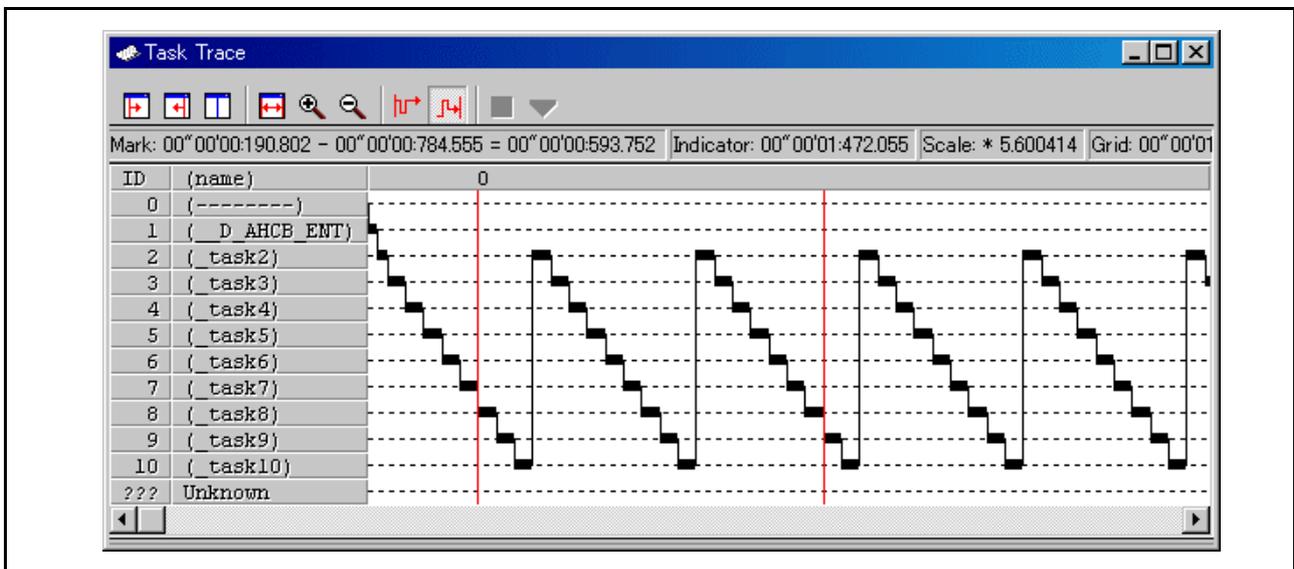
7.19.2.4 更改任务显示顺序

右击窗口，从弹出菜单中选择 [Color ...]（颜色...）。此时会打开 [Color]（颜色）对话框。

单击与所需项目相对应的按钮。此时会打开 [Color Setting]（颜色设置）对话框。在对话框中更改显示颜色。

7.19.2.5 计量任务的执行时间

通过在 [Task Trace]（任务跟踪）窗口中更改开始标志位置和结束标志位置，可以计量标志之间的执行时间。



拖动开始标志位置和结束标志位置。

标志之间的时间间隔会显示在状态栏中。

注意

在 [Task Trace]（任务跟踪）窗口中定义时间值

[Task Trace]（任务跟踪）窗口中的时间值表示所有情况下将程序执行起始点设置为 0 所经历的执行时间。

相反，[Task Trace]（任务跟踪）窗口中网格线（比例）上面的数值是将开始标志设置为 0 的相对值（网格间隔是在 [Value]（值）对话框中指定的），该值与时间值无关。（提供此数值是为了让您能够轻松查看该窗口。）

7.19.2.5.1 移动标志

每个标志都可以通过拖动来移动。在标志上设置并移动鼠标时，光标的形状会更改。然后开始以鼠标拖动该标志。

单击工具栏中的 [Start Marker]（开始标志）按钮，开始标志会移动到窗口（左边部分）中。（或者，右击窗口，从弹出菜单中选择 [Start Marker]（开始标志）。）

单击 [End Marker]（结束标志）按钮，结束标志会移动到窗口（右边部分）中。（或者，从弹出菜单中选择 [End Marker]（结束标志）。）

单击 “Indicator”（指示器）按钮，指示器会移动到窗口（中心）中。（或者，从弹出菜单中选择 [Indicator]（指示器）。）

请注意，每个标志只能移动到建立了事件的点上。

7.20 [Task Analyze] (任务分析) 窗口

[Task Analyze] (任务分析) 窗口显示在 [Task Trace] (任务跟踪) 窗口中开始标志和结束标志指定的范围内经过统计分析的计量数据的结果。

[Task Analyze] (任务分析) 窗口显示 CPU 占用状态。

[Task Analyze] (任务分析) 窗口与 [Task Trace] (任务跟踪) 窗口搭配使用。

即使下载的目标程序使用的是实时操作系统 (MRxx) 以外的操作系统, 也可以使用此窗口。

“CPU occupation state display mode” (CPU 占用状态显示模式) 用于显示任务占用 CPU 的时间和比率。

此模式显示 [Task Trace] (任务跟踪) 窗口中开始标志和结束标志指定的范围内的统计结果。

ID	(name)	Num	Max Run Time	Min Run Time	Avg Run Time	Total Run Time	Ratio%	0	25	50	75	100
0	{-----}	1	00"00"00:005.195	00"00"00:005.195	00"00"00:005.195	00"00"00:005.195	0.33					
1	{ main }	115	00"00"00:007.305	00"00"00:000.767	00"00"00:001.541	00"00"00:177.287	11.30	█				
2	{ task002 }	12	00"00"00:012.067	00"00"00:006.915	00"00"00:011.552	00"00"00:138.630	8.84	█				
3	{ task003 }	12	00"00"00:012.597	00"00"00:006.892	00"00"00:012.111	00"00"00:145.332	9.27	█				
4	{ task004 }	12	00"00"00:012.170	00"00"00:006.505	00"00"00:011.604	00"00"00:139.252	8.88	█				
5	{ task005 }	12	00"00"00:012.277	00"00"00:006.577	00"00"00:011.795	00"00"00:141.540	9.02	█				
6	{ task006 }	11	00"00"00:013.435	00"00"00:006.490	00"00"00:012.353	00"00"00:135.885	8.66	█				
7	{ task007 }	11	00"00"00:013.020	00"00"00:006.790	00"00"00:012.431	00"00"00:136.745	8.72	█				
8	{ task008 }	11	00"00"00:014.080	00"00"00:008.055	00"00"00:013.232	00"00"00:145.552	9.28	█				
9	{ task009 }	11	00"00"00:013.642	00"00"00:007.277	00"00"00:012.663	00"00"00:139.295	8.88	█				
10	{ task010 }	11	00"00"00:013.710	00"00"00:008.070	00"00"00:012.795	00"00"00:140.752	8.97	█				
11	{ task011 }	11	00"00"00:011.892	00"00"00:006.290	00"00"00:011.193	00"00"00:123.132	7.85	█				
???	Unknown	0	00"00"00:000.000	00"00"00:000.000	00"00"00:000.000	00"00"00:000.000	0.00					

通过单击每行的最长执行时间 / 最短执行时间显示区, 可以搜索最长 / 最短执行时间时单击行任务的历史记录处理。

搜索结果由移动到 [Task Trace] (任务跟踪) 窗口中目标位置的指示器指示。

7.20.1 扩展菜单

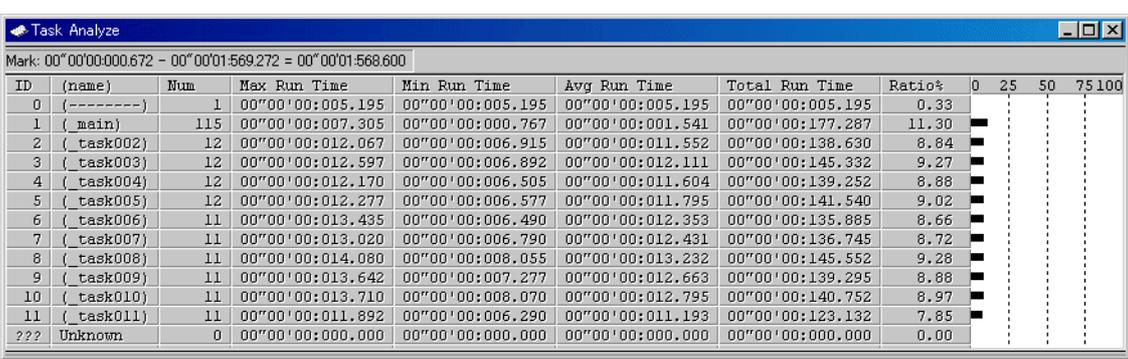
此窗口具有如下弹出菜单, 在窗口中右键单击便可看到。

菜单	功能
[Allow Docking] (允许入坞)	允许窗口入坞。
[Hide] (隐藏)	隐藏窗口。

7.20.2 分析任务的执行历史记录

可以在 [Task Analyze]（任务分析）窗口中查看执行历史记录统计处理。此窗口显示任务占用的 CPU 时间和 CPU 比率。

[Task Analyze]（任务分析）窗口与 [Task Trace]（任务跟踪）窗口搭配使用。如果 [Task Trace]（任务跟踪）窗口未打开，或者 [Task Trace]（任务跟踪）窗口未显示任何数据，则 [Task Analyze]（任务分析）窗口将无法发挥作用。



ID	(name)	Num	Max Run Time	Min Run Time	Avg Run Time	Total Run Time	Ratio%	0	25	50	75	100
0	{-----}	1	00"00'00:005.195	00"00'00:005.195	00"00'00:005.195	00"00'00:005.195	0.33					
1	{ main}	115	00"00'00:007.305	00"00'00:000.767	00"00'00:001.541	00"00'00:177.287	11.30	█				
2	{ task002}	12	00"00'00:012.067	00"00'00:006.915	00"00'00:011.552	00"00'00:138.630	8.84	█				
3	{ task003}	12	00"00'00:012.597	00"00'00:006.892	00"00'00:012.111	00"00'00:145.332	9.27	█				
4	{ task004}	12	00"00'00:012.170	00"00'00:006.505	00"00'00:011.604	00"00'00:139.252	8.88	█				
5	{ task005}	12	00"00'00:012.277	00"00'00:006.577	00"00'00:011.795	00"00'00:141.540	9.02	█				
6	{ task006}	11	00"00'00:013.435	00"00'00:006.490	00"00'00:012.353	00"00'00:135.885	8.66	█				
7	{ task007}	11	00"00'00:013.020	00"00'00:006.790	00"00'00:012.431	00"00'00:136.745	8.72	█				
8	{ task008}	11	00"00'00:014.080	00"00'00:008.055	00"00'00:013.232	00"00'00:145.552	9.28	█				
9	{ task009}	11	00"00'00:013.642	00"00'00:007.277	00"00'00:012.663	00"00'00:139.295	8.88	█				
10	{ task010}	11	00"00'00:013.710	00"00'00:008.070	00"00'00:012.795	00"00'00:140.752	8.97	█				
11	{ task011}	11	00"00'00:011.892	00"00'00:006.290	00"00'00:011.193	00"00'00:123.132	7.85	█				
???	Unknown	0	00"00'00:000.000	00"00'00:000.000	00"00'00:000.000	00"00'00:000.000	0.00					

显示的数据是 [Task Trace]（任务跟踪）窗口中开始标志和结束标志所指定范围的统计结果。

通过单击每行的最长就绪时间 / 最短就绪时间显示字段，可以搜索单击行对应任务的最长就绪时间 / 最短就绪时间的处理历史记录。

在 [Task Trace]（任务跟踪）窗口中的指示器移动到目标位置以后，搜索结果由该指示器指示。

参考

8 脚本命令表

我们为您准备了以下脚本命令。

带黄色背景的命令可以在运行时执行。

该产品不支持后面带有“*”的命令。

8.1 脚本命令表（按功能分类）

8.1.1 执行命令

命令名称	短名	内容
Go	G	带有断点的程序执行
GoFree	GF	自由运行程序执行
GoProgramBreak*	GPB	运行带有软件断点的目标程序
GoBreakAt*	GBA	运行带有软件断点的目标程序
Stop	-	停止程序执行
Status	-	检查 MCU 的操作状态
Step	S	暂停以便用户输入，直到指定的时间过去为止
Step	SI	单步执行（机器）指令
OverStep	O	源行的执行跳过
OverStepInstruction	OI	（机器）指令的执行跳过
Return	RET	源行的执行返回
ReturnInstruction	RETI	（机器）指令的执行返回
Reset	-	复位目标 MCU
Time	-	设置运行时显示并检查当前设置

8.1.2 文件操作命令

命令名称	短名	内容
Load	L	下载目标程序
LoadHex	LH	下载 Intel HEX 格式文件
LoadMot*	LM	下载 Motorola S 格式文件
LoadSymbol	LS	装入源行 /ASM 符号信息
Reload	-	重新下载目标程序
UploadHex	UH	将数据输出到 Intel HEX 格式文件
UploadMot*	UM	将数据输出到 Motorola S 格式文件

8.1.3 寄存器操作命令

命令名称	短名	内容
Register	R	检查和设置寄存器值

8.1.4 存储器操作命令

命令名称	短名	内容
DumpByte	DB	显示存储器的内容 (以 1 个字节为单位)
DumpWord*	DW	显示存储器的内容 (以 2 个字节为单位)
DumpLword*	DL	显示存储器的内容 (以 4 个字节为单位)
SetMemoryByte	MB	检查和更改存储器内容 (以 1 个字节为单位)
SetMemoryWord*	MW	检查和更改存储器内容 (以 2 个字节为单位)
SetMemoryLword*	ML	检查和更改存储器内容 (以 4 个字节为单位)
FillByte	FB	用指定数据填充存储块 (以 1 个字节为单位)
FillWord*	FW	用指定数据填充存储块 (以 2 个字节为单位)
FillLword*	FL	用指定数据填充存储块 (以 4 个字节为单位)
Move	-	移动存储块 (以 1 个字节为单位)
MoveWord*	MOVEW	移动存储块 (以 2 个字节为单位)

8.1.5 汇编 / 反汇编命令

命令名称	短名	内容
Assemble	A	逐行汇编
DisAssemble	DA	逐行反汇编存储器内容
Module	MOD	显示模块名称
Scope	-	设置和检查有效的局部符号作用域
Section	SEC	检查段信息
Bit*	-	检查和设置位符号
Symbol	SYM	检查汇编器符号
Label	-	检查汇编器标签
Express	EXP	显示汇编器表达式

8.1.6 软件断点设置命令

命令名称	短名	内容
SoftwareBreak	SB	设置和检查软件断点
SoftwareBreakClear	SBC	清除软件断点
SoftwareBreakClearAll	SBCA	清除所有软件断点
SoftwareBreakDisable	SBD	禁止软件断点
SoftwareBreakDisableAll	SBDA	禁止所有软件断点
SoftwareBreakEnable	SBE	允许软件断点
SoftwareBreakEnableAll	SBEA	允许所有软件断点
BreakAt	-	通过指定行号设置软件断点
BreakIn	-	通过指定函数设置软件断点

8.1.7 地址中断断点设置命令

命令名称	短名	内容
ADdressInterruptBreak	ADIB	设置和检查地址中断断点

8.1.8 硬件断点设置命令

命令名称	短名	内容
HardwareBreak	HB	设置和检查硬件断点
Protect	PT	设置和检查中断保护
BreakMode	BM	设置和检查硬件断点模式

8.1.9 实时跟踪命令

命令名称	短名	内容
TracePoint	TP	设置和检查跟踪点
TraceData*	TD	实时跟踪数据显示
TraceList*	TL	显示反汇编的实时跟踪数据

8.1.10 覆盖率计量命令

命令名称	短名	内容
Coverage	CV	指定和显示覆盖率计量

8.1.11 脚本 / 日志文件命令

命令名称	短名	内容
Script	-	打开和执行脚本文件
Exit	-	退出脚本文件
Wait	-	在输入命令之前等待事件发生
Pause	-	等待用户输入
Sleep	-	暂停以便用户输入，直到指定的时间过去为止
Logon	-	将屏幕显示内容输出到日志文件
Logoff	-	停止将屏幕显示内容输出到日志文件
Exec	-	执行外部应用程序

8.1.12 程序显示命令

命令名称	短名	内容
Func	-	检查函数名称和显示函数内容
Up*	-	显示调用函数
Down*	-	显示被调用函数
Where*	-	显示函数调用状态
Path	-	设置和检查源文件搜索路径
AddPath	-	添加源文件路径
File	-	检查文件名和显示该文件的内容

8.1.13 映像命令

命令名称	短名	内容
Map*	-	检查和设置映像数据

8.1.14 时钟命令

命令名称	短名	内容
Clock	CLK	检查和更改时钟
ClockControl*	CLKC	检查和更改 XOUT、P86/XCOUT 引脚

8.1.15 监视定时器命令

命令名称	短名	内容
WatchDogTimer*	WDT	设置和检查监视定时器的使用情况

8.1.16 C 语言调试命令

命令名称	短名	内容
Print	-	检查指定 C 变量表达式的值
Set	-	在指定 C 变量表达式中设置指定数据

8.1.17 实时操作系统命令

命令名称	短名	内容
MR*	-	显示实时操作系统的状态 (MRxx)

8.1.18 实用程序命令

命令名称	短名	内容
Radix	-	设置和检查数字输入的基数
Alias	-	指定和检查命令别名定义
UnAlias	-	取消为命令定义的别名
UnAliasAll	-	取消为命令定义的所有别名
Version	VER	显示版本号
Date	-	显示日期
Echo	-	显示消息
CD	-	窗口打开

8.2 脚本命令表（依字母顺序）

命令名称	短名	内容
AddPath	-	添加源文件路径
ADdressInterruptBreak	ADIB	设置和检查地址中断断点
Alias	-	指定和检查命令别名定义
Assemble	A	逐行汇编
Bit*	-	检查和设置位符号
BreakAt	-	通过指定行号设置软件断点
BreakIn	-	通过指定函数设置软件断点
BreakMode	BM	设置和检查硬件中断模式
CD	-	指定和检查当前目录
Clock	CLK	检查和更改时钟
ClockControl*	CLKC	检查和更改 XOUT、P86/XCOUT 引脚
Coverage	CV	指定和显示覆盖率计量
Date	-	显示日期
DisAssemble	DA	逐行反汇编存储器内容
Down*	-	显示被调用函数
DumpByte	DB	显示存储器的内容（以 1 个字节为单位）
DumpLword*	DL	显示存储器的内容（以 4 个字节为单位）
DumpWord*	DW	显示存储器的内容（以 2 个字节为单位）
Echo	-	显示消息
Exec	-	执行外部应用程序
Exit	-	退出脚本文件
Express	EXP	显示汇编器表达式
File	-	检查文件名和显示该文件的内容
FillByte	FB	用指定数据填充存储块（以 1 个字节为单位）
FillLword*	FL	用指定数据填充存储块（以 4 个字节为单位）
FillWord*	FW	用指定数据填充存储块（以 2 个字节为单位）
Func	-	检查函数名称和显示函数内容
Go	G	带有断点的程序执行
GoBreakAt*	GBA	运行带有软件断点的目标程序
GoFree	GF	自由运行程序执行
GoProgramBreak*	GPB	运行带有软件断点的目标程序
HardwareBreak	HB	设置和检查硬件断点
Label	-	检查汇编器标签
Load	L	下载目标程序
LoadHex	LH	下载 Intel HEX 格式文件
LoadMot*	LM	下载 Motorola S 格式文件

命令名称	短名	内容
LoadSymbol	LS	装入源行 /ASM 符号信息
Logoff	-	停止将屏幕显示内容输出到日志文件
Logon	-	将屏幕显示内容输出到日志文件
Map*	-	检查和设置映像数据
Module	MOD	显示模块名称
Move	-	移动存储块 (以 1 个字节为单位)
MoveWord*	MOVEW	移动存储块 (以 2 个字节为单位)
MR*	-	显示实时操作系统的状态 (MRxx)
OverStep	O	源行的执行跳过
OverStepInstruction	OI	(机器) 指令的执行跳过
Path	-	设置和检查源文件路径
Pause	-	等待用户输入
Print	-	检查指定 C 变量表达式的值
Protect	PT	设置和检查中断保护
Radix	-	设置和检查数字输入的基数
Register	R	检查和设置寄存器值
Reload	-	重新下载目标程序
Reset	-	复位目标 MCU
Return	RET	源行的执行返回
ReturnInstruction	RETI	(机器) 指令的执行返回
Scope	-	设置和检查有效的局部符号作用域
Script	-	打开和执行脚本文件
Section	SEC	检查段信息
Set	-	在指定 C 变量表达式中设置指定数据
SetMemoryByte	MB	检查和更改存储器内容 (以 1 个字节为单位)
SetMemoryLword*	ML	检查和更改存储器内容 (以 4 个字节为单位)
SetMemoryWord*	MW	检查和更改存储器内容 (以 2 个字节为单位)
Sleep	-	暂停以使用户输入, 直到指定的时间过去为止
SoftwareBreak	SB	设置和检查软件断点
SoftwareBreakClear	SBC	清除软件断点
SoftwareBreakClearAll	SBCA	清除软件断点
SoftwareBreakDisable	SBD	禁止软件断点
SoftwareBreakDisableAll	SBDA	禁止所有软件断点
SoftwareBreakEnable	SBE	允许软件断点
SoftwareBreakEnableAll	SBEA	允许所有软件断点
Status	-	检查 MCU 的操作状态
Step	S	单步执行源行

命令名称	短名	内容
StepInstruction	SI	单步执行（机器）指令
Stop	-	停止程序执行
Symbol	SYM	检查汇编器符号
Time	-	设置运行时显示并检查当前设置
TraceData*	TD	实时跟踪数据显示
TraceList*	TL	显示反汇编的实时跟踪数据
TracePoint	TP	设置和检查跟踪点
UnAlias	-	取消为命令定义的别名
UnAliasAll	-	取消为命令定义的所有别名
Up*	-	显示调用函数
UploadHex	UH	将数据输出到 Intel HEX 格式文件
UploadMot*	UM	将数据输出到 Motorola S 格式文件
Version	VER	显示版本号
Wait	-	在输入命令之前等待事件发生
WatchDogTimer*	WDT	设置和检查监视定时器的使用情况
Where*	-	显示函数调用状态

9 编写脚本文件

此调试器允许您在 [Script]（脚本）窗口中运行脚本文件。脚本文件包含自动执行脚本命令所需的控制。

9.1 脚本文件的结构元素

可以在脚本文件中包括以下内容：

- 脚本命令
- 赋值语句
- 条件语句（if、else、endi）
程序执行分支到要根据条件表达式结果执行的语句。
- 循环语句（while、endw）
根据表达式，重复执行一条或多条语句构成的块。
- 中断语句
从最里面的循环退出。
- 注释语句
可以在脚本文件中包括注释。执行脚本命令时，会忽略注释语句。

在脚本文件的每一行上，只能指定一条语句。不能在一行上指定多条语句，也不能编写跨越两行或更多行的语句。

注意

- 不能在与脚本命令相同的行上加插注释。
- 最多可以将脚本文件嵌套五层。
- 最多可以将 if 语句和 while 语句嵌套 32 层。
- 在每个脚本文件中，if 语句必须与 endi 语句成对出现，while 语句必须与 endw 语句成对出现。
- 脚本文件中包含的表达式将作为无符号类型进行计算。因此，如果在 if 或 while 语句中使用负值来进行比较，将无法保证正常运算。
- 每行最多可以指定 4096 个字符。如果某行超过此字符数，会发生错误。
- 当自动执行某个包含非法命令的脚本文件时（打开脚本文件后，从 [Script]（脚本）窗口菜单中选择 [Option]（选项）->[Script]（脚本）->[Run]（运行），或者在 [Script]（脚本）窗口中单击该按钮），即使在检测到错误以后，脚本文件也会继续执行，除非是无法读取脚本行本身。如果检测到错误并且脚本文件继续执行，则无法保证检测到错误以后的运算。因此，在检测到错误以后，执行的结果并不可靠。

9.1.1 脚本命令

可以使用在 [Script]（脚本）窗口中输入的相同脚本命令。也可以在其他脚本文件内部调用脚本文件（最多嵌套 10 层）。

9.1.2 赋值语句

赋值语句定义并初始化宏变量以及进行赋值。以下所示为使用的格式。

```
%macro-variable = expression
```

- 可以在宏变量名称中使用字母数字和下划线 (_)。不过，不能在宏变量名称开头使用数字。
- 可以在宏变量中指定分配任何表达式，该表达式的值是一个介于 0h 和 FFFFFFFFh 之间的整数。如果指定负数，它会被当作二的补码来处理。
- 可以在表达式内部使用宏变量。
- 请始终在宏变量之前添加 “%” 符号。

9.1.3 条件语句

在条件语句中，会根据条件真伪来执行不同的语句。以下所示为使用的格式。

```
if ( expression )  
    statement 1  
else  
    statement 2  
endi
```

- 如果表达式为真（非 0），则执行 statement 1。如果为伪 (0)，则执行 statement 2。
- 可以省略 else 语句。如果省略并且表达式为伪，则执行会跳到 endi 语句之后的行。
- if 语句可以嵌套（最多 32 层）。

9.1.4 循环语句（while 及 endw）和中断语句

在循环语句中，当表达式为真时，重复执行一组语句。以下所示为使用的格式。

```
while ( expression )  
    statement  
endw
```

- 如果表达式为真，则重复该组语句。如果为伪，则退出循环（执行 endw 语句之后的语句）。
- 最多可以将 while 语句嵌套 32 层。
- 使用中断语句可以强制退出 while 循环。如果 while 语句是嵌套的，则中断会从最里面的循环退出。

9.1.5 注释语句

可以在脚本文件中包括注释。请使用以下格式。

```
;character string
```

- 在分号 (;) 后面编写语句。分号前面只能包括空格和制表符
- 执行脚本文件时，会忽略注释语句行。

9.2 编写表达式

此调试器允许您使用表达式来指定地址、数据和传递数等等。以下所示为使用了表达式的示例命令。

```
>DumpByte TABLE1
```

```
>DumpByte TABLE1+20
```

可以在表达式中使用以下元素：

- 常数
- 符号和标签
- 宏变量
- 寄存器变量
- 存储器变量
- 行号
- 字符常数
- 运算符

9.2.1 常数

可以使用二进制、八进制、十进制或十六进制数。附加到数值上的前缀或后缀符号指示使用的是哪一种基数。
M32C、M16C/R8C 和 740 的调试器

	十六进制	十进制	八进制	二进制 *
前缀	0x、0X	@	无	%
后缀	h、H	无	o、O	b、B
示例	0xAB24 AB24h	@1234	1234o	%10010 10010b

* 当预先确定的基数是十六进制数时，只能指定 %。

如果您输入的是一个与预先确定的基数匹配的基数，则可以省略指示该基数（二进制数除外）的符号。

- 使用 RADIX 命令可以设置预先确定的基数值。不过，在下面显示的情况中，无论在 RADIX 命令中指定什么内容，基数都是固定的。

类型	基数
地址	十六进制
行号 执行数 传递数	十进制

9.2.2 符号和标签

可以包括目标程序中定义的符号和标签，或者包括使用汇编命令定义的符号和标签。

- 可以在符号和标签中包括字母数字、下划线 (_)、句点 (.) 和问号 (?)。不过，请不要以数字开头。
- 符号和标签最多可以由 255 个字符组成。
- 有大写字母和小写字母的区别。

产品名称	注意
用于 M32R 的调试器、 用于 M32C 的调试器、 用于 M16C/R8C 的调试器	<ul style="list-style-type: none"> • 不能包括汇编器结构化指令、伪指令、宏指令、操作码或保留字 (.SECTION、.BYTE、switch、if 等等)。 • 不能使用以两个句点 (..) 开头的字符串作为符号或标签。

9.2.2.1 局部标签符号和作用域

此调试器支持全局标签符号（可以从整个程序区引用全局标签符号）和局部标签符号（只能在声明局部标签符号的文件内部引用局部标签符号）。局部标签符号的有效范围就是通常所说的作用域，它是以目标文件为单位计量的。在以下情况下，作用域会在此调试器中发生转变：

- 输入命令时
包括程序计数器所指示地址的目标文件会成为当前作用域。当使用 SCOPE 命令设置作用域时，指定的作用域为活动作用域。
- 在命令执行过程中
根据命令正在处理的程序地址，当前作用域会自动转变。

9.2.2.2 标签和符号的优先级

将值转换为标签或符号（反之亦然）时，应遵从以下优先级：

- 转换地址值
 1. 局部标签
 2. 全局标签
 3. 局部符号
 4. 全局符号
 5. 作用域之外的局部标签
 6. 作用域之外的局部符号
- 转换数据值
 1. 局部符号
 2. 全局符号
 3. 局部标签
 4. 全局标签
 5. 作用域之外的局部符号
 6. 作用域之外的局部标签
- 转换位值
 1. 局部位符号
 2. 全局位符号
 3. 作用域之外的局部位符号

9.2.3 Macro Variables 宏变量

宏变量是由脚本文件中的赋值语句定义的。有关详细信息，请参阅参考部分中的“9.1.2 赋值语句”一节。请在变量之前添加“%”，以便用作宏变量。

- 可以在百分比符号 (%) 后面的变量名称中指定字母数字和/或下划线 (_)。不过，请不要以数字作为名称的开头。
- 不能使用寄存器的名称作为变量名称。
- 变量名称区分大写字母和小写字母。
- 最多可以定义 32 个宏变量。一旦定义，宏变量将保持有效，直到退出调试器。

宏变量用于指定 while 语句的重复次数。

9.2.4 寄存器变量

如果使用寄存器变量，则可以在表达式中使用寄存器的值。在寄存器名称之前添加“%”可以将其用作寄存器变量。请使用以下格式。

产品名称	寄存器名称
用于 M32C 的调试器	PC、USP、ISP、INTB、FLB、SVF、SVP、VCT、 DMD0、DMD1、DCT0、DCT1、DRC0、DRC1、 DMA0、DMA1、DCA0、DCA1、DRA0、DRA1、 0R0、0R1、0R2、0R3、0A0、0A1、0FB、0SB <- 存储体 0 寄存器 1R0、1R1、1R2、1R3、1A0、1A1、1FB、1SB <- 存储体 1 寄存器
用于 M16C/R8C 的调试器	PC、USP、ISP、SB、INTB、FLG 0R0、0R1、0R2、0R3、0A0、0A1、0FB <- 存储体 0 寄存器 1R0、1R1、1R2、1R3、1A0、1A1、1FB <- 存储体 1 寄存器

在寄存器名称中不区分大小写字母。可以指定任一形式的字母。

9.2.5 存储器变量

如果使用存储器变量，则可以在表达式中使用存储器值。格式如下：

[Address].data-size

- 可以在地址中指定表达式（也可以指定存储器变量）。
- 如下表所示指定数据长度。

数据长度	调试器	规范
1 个字节	全部	B 或 b
2 个字节	用于 M32R 的调试器	H 或 h
	其他	W 或 w
4 个字节	用于 M32R 的调试器	W 或 w
	用于 M32R、M16C/R8C 的调试器	L 或 l

示例：引用位于地址 8000h 的存储器的内容（数据长度为 2 个字节）

[0x8000].w

- 如果没有指定，则默认数据长度为字。

9.2.6 行号

指源文件行号。行号格式如下：

```
#line_no
#line_no."source file name"
```

- 使用十进制数指定行号。
- 只能指定可以在其中设置软件断点的行号。不能指定未生成任何汇编器指令的行，包括注释行和空白行。
- 如果省略源文件名称，则对应源文件中的行号会显示在活动 [Editor(Source)]（编辑器（源））窗口中。
- 在源文件名称中包含文件属性。
- 请不要在行号和源文件名称之间添加任何空格。

9.2.7 字符常数

指定的字符或字符串会转换为 ASCII 码并作为常数处理。

- 将字符括在单引号中。
- 将字符串括在双引号中。
- 字符串必须由一个或两个字符组成（最多 16 位）。如果指定的字符超过两个，则该字符串的首两个字符后的字符将会被处理。例如，“ABCD”会当作“CD”或值 4344h 来处理。

9.2.8 运算符

下表列出可以在表达式中使用的运算符。

- 运算符的优先级由级别表示，级别 1 最高，级别 8 最低。如果两个或更多个运算符具有相同优先级，则从表达式左边开始依次进行计算。

运算符	功能	优先级
()	括号	级别 1
+, -, ~	一元正、一元负、一元逻辑非	级别 2
*, /	二元乘法、二元除法	级别 3
+, -	二元加法、二元减法	级别 4
>>, <<	右移、左移	级别 5
&	二元逻辑“与”	级别 6
, ^	二元逻辑“或”、二元逻辑“异（或）”	级别 7
<, <=, >, >=, ==, !=	二元比较	级别 8

10 C/C++ 表达式

10.1 编写 C/C++ 表达式

使用由如下所示标记组成的 C/C++ 表达式，您可以将 C 监视点写入寄存器，并指定要赋给这些监视点的值。

标记	示例
直接值	10、0x0a、012、1.12、1.0E+3
作用域	::name、classname::member
数学运算符	+、-、*、/
指针	*、**...
引用	&
取反	-
使用点运算符的成员引用	Object.Member
使用箭头运算符的成员引用	Pointer->Member、this->Member
指向成员的指针	Object.*var、Pointer->*var
括号	(、)
数组	Array[2]、DArray[2][3]...
基本数据类型转换	(int)、(char*)、(unsigned long *)...
typedef 类型转换	(DWORD)、(ENUM)...
变量名和函数名	var、i、j、func...
字符常数	'A'、'b'...
字符串文字	"abcdef"、"I am a boy." ...

10.1.1 直接值

您可以将十六进制数、十进制数和八进制数用作直接值。以 0x 开头的值将视为十六进制数进行处理，以 0 开头的值被视为八进制数，不使用这两种前缀的值被视为十进制数。

还可以使用浮点数给变量赋值。

注意

- 不能只注册直接值作为 C 监视点。
- 仅当直接值在用于指定 C/C++ 监视点的 C/C++ 语言表达式中使用，或者用于指定要赋给这些表达式的值时，该直接值才有效。无法对类似于 1.0+2.0 这样的使用浮点数的表达式执行运算。

10.1.2 作用域解析

作用域解析运算符 :: 的使用方式如下。

全局作用域: ::variable name

::x, ::val

类作用域: class name::member name, class name::class name::member name。例如:

T::member, A::B::member

10.1.3 数学运算符

您可以使用加号 (+)、减号 (-)、乘号 (*) 和除号 (/) 数学运算符。如下所示为上述数学运算符用于求值时的优先级顺序。

(*), (/), (+), (-)

注意

- 目前不支持对浮点数使用数学运算符。

10.1.4 指针

指针由星号 (*) 指示。您可以使用指向指针的指针 **, 以及指向指针的指针的指针 ***, 依次类推。

示例: ****variable_name**、*****variable_name** 等

注意

- 不能把直接值当作指针进行处理。例如, 您不能指定 *0xE000。

10.1.5 引用

引用由 “&” 符指示。只能指定 “&variable_name” 这种形式。

10.1.6 取反

取反由减号 (-) 指示。只能指定 “-immediate_value” 或 “-variable_name” 这种形式。如果指定了 2 个 (或任意偶数个) 减号, 则不会执行取反。

注意

- 目前不支持对浮点数取反。

10.1.7 使用点运算符的成员引用

使用点运算符时只能通过“variable_name.member_name”的形式引用结构及联合的成员。

示例:

```
class T {
public:
int member1;
char member2;
};
class T t_cls;
class T *pt_cls = &t_cls;
```

在这种情况下，t_cls.member1 和 (*pt_cls).member2 均可正确引用成员。

10.1.8 使用箭头运算符的成员引用

使用箭头运算符时只能通过“variable_name->member_name”的形式引用结构及联合的成员。

示例:

```
class T {
public:
int member1;
char member2;
};
class T t_cls;
class T *pt_cls = &t_cls;
```

在这种情况下，(&t_cls)->member1 和 pt_cls->member2 均可正确引用成员。

10.1.9 指向成员的指针

对于使用“*”或“->*”运算符指向成员的指针，只能通过如下形式进行引用：variable name .* member name 或 variable name ->* member name。

示例:

```
class T {
public:
int member;
};
class T t_cls;
class T *pt_cls = &t_cls;

int T::*mp = &T::member;
```

在这种情况下，t_cls.*mp 和 tp_cls->*mp 可正确引用成员指针类型的变量。

注意

- 请注意，不能将表达式 *mp 视为成员指针类型的变量。

10.1.10 括号

使用 “(” 和 “)” 可指定表达式中计算的优先级。

10.1.11 数组

可以使用 “[” 和 “]” 指定数组的元素。您可以通过如下方式指定数组：“variable_name [(element_No 或变量)]”、“variable_name [(element_No 或变量)] [(element_No 或变量)]”，等等。

10.1.12 基本数据类型转换

您可以对 C 基本数据类型（如 char、short、int 和 long）执行转换，还可以将指针类型转换成这些基本数据类型。执行指针数据类型转换时，您可以使用的指针数据类型包括指向指针的指针，以及指向指针的指针的指针，等等。

请注意，如果未指定是否带符号，则默认值如下所示：

基本数据类型	默认值
char	无符号
short	带符号
int	带符号
long	带符号

注意

- 在 C++ 的基本数据类型中，无法对 bool 类型、wchar_t 类型和 floating-point 类型（浮点值或双精度值）数据进行转换。
- 无法执行对寄存器变量的转换。

10.1.13 typedef 类型转换

可以执行对 typedef 类型（除 C 基本数据类型以外的类型）的转换，以及将指针类型转换为 typedef 类型。执行指针数据类型转换时，您可以使用的指针数据类型包括指向指针的指针，以及指向指针的指针的指针，等等。

注意

- 无法执行对 struct 类型或 union 类型的转换，无法将指针类型转换成这些类型。

10.1.14 变量名

可以根据 C/C++ 约定使用以

英文字母开头的变量名。

变量名允许的最大字符数为 255。

可以使用 “this” 指针。

10.1.15 函数名

可以根据 C 约定使用以
英文字母开头的函数名。

对于 C++，不能使用任何函数名。

10.1.16 字符常数

可将由单引号 (') 引住的字符用作字符常数。例如，'A'、'b' 等等。这些字符常数将转换为 ASCII 码，并用作单字节的直接值。

注意

- 不能只是注册字符常数作为 C 监视点。
- 仅在指定 C 监视点的 C/C++ 表达式中作为字符常数使用，或者用于指定要赋予的值时，这些字符常数才有效（处理字符常数的方式与处理直接值的方式相同）。

10.1.17 字符串文字

可将由双引号 (" ") 引住的字符串用作字符串文字。例如，“abcde”、“I am a boy.” 等等。

注意

- 在表达式中，字符串文字只能位于赋值运算符的右侧。使用字符串文字的前提是赋值运算符左侧必须为 char 数组类型或 char 指针类型。否则会导致语法错误。

10.2 C/C++ 表达式的显示格式

在 [C Watch] (C 监视) 窗口的数据显示区域中, C/C++ 表达式将显示为类型名称、C/C++ 表达式 (变量名) 以及计算结果 (值), 如下所示。

下文将分别介绍各类型的显示格式。

10.2.1 枚举类型

- 如果计算结果 (值) 已确定, 则显示其名称。

(DATE) date = Sunday (所有基数)

- 如果计算结果 (值) 尚未确定, 则其显示如下:

(DATE) date = 16 (如果基数为初始状态)

(DATE) date = 0x10 (如果基数为十六进制数)

(DATE) date = 0000000000010000B (如果基数为二进制数)

10.2.2 基本数据类型

- 当计算结果为除 char 类型或浮点类型以外的基本数据类型时, 其显示如下:

(unsigned int) i = 65280 (如果基数为初始状态)

(unsigned int) i = 0xFF00 (如果基数为十六进制数)

(unsigned int) i = 1111111100000000B (如果基数为二进制数)

当计算结果为 char 类型时, 其显示如下:

(unsigned char) c = 'J' (如果基数为初始状态)

(unsigned char) c = 0x4A (如果基数为十六进制数)

(unsigned char) c = 10100100B (如果基数为二进制数)

- 当计算结果为浮点类型时, 其显示如下:

(double) d = 8.207880399131839E-304 (如果基数为初始状态)

(double) d = 0x10203045060708 (如果基数为十六进制数)

(double) d = 000000010.....1000B (如果基数为二进制数)

(..... 表示省略的内容)

10.2.3 指针类型

- 当计算结果为非 char* 类型的指针数据类型时，其将以十六进制形式显示如下：

```
(unsigned int *) p = 0x1234 (所有基数)
```

- 当计算结果为 char* 类型时，您可以在 [C Watch] (C 监视) 窗口的 [Display String] (显示字符串) 菜单中选择字符串或字符形式的显示格式。

— 字符串类型

```
(unsigned char *) str = 0x1234 "Japan" (所有基数)
```

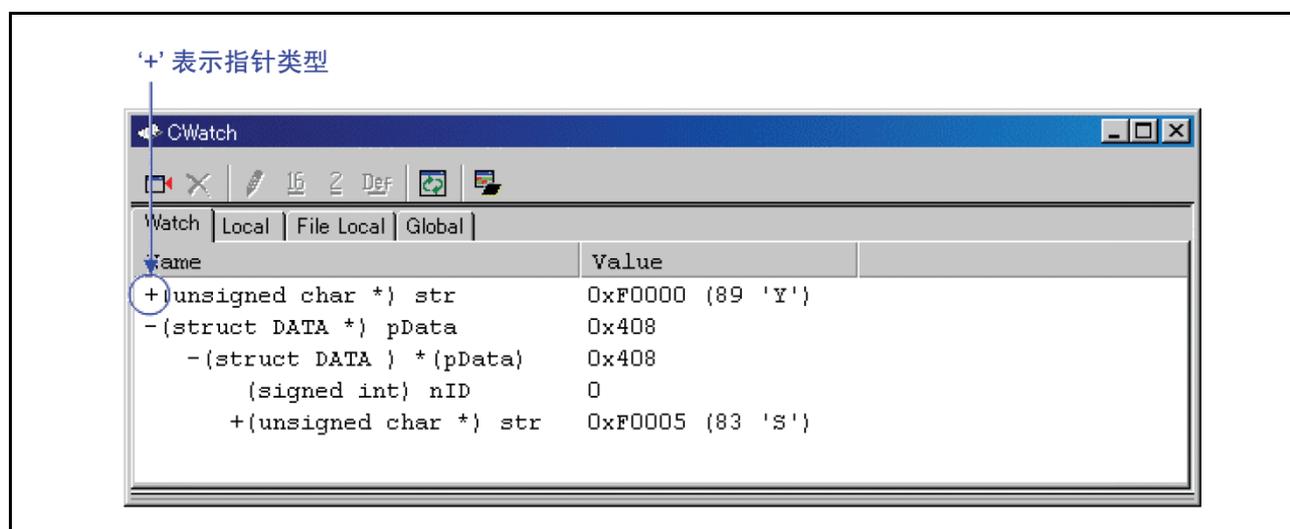
— 字符类型

```
(unsigned char *) str = 0x1234 (74 'J') (所有基数)
```

当计算结果为 char* 类型时，其显示如下：

```
(unsigned char *) str = 0x1234 "Jap (所有基数)
```

如果字符串中某个符号前包含一个非打印符号，用以显示字符串结束符 (0)，则字符串仅显示到此非打印字符位置，并且不显示右引号。



您可以双击由“+”指示的行，查看该结构或联合的成员。显示成员时“+”变为“-”。要恢复初始显示，双击现由“-”指示的那一行。

10.2.4 数组类型

- 当计算结果为非 char [] 类型的数组数据类型时，起始地址将按以下形式以十六进制数显示：

(signed int [10]) z = 0x1234 (所有基数)

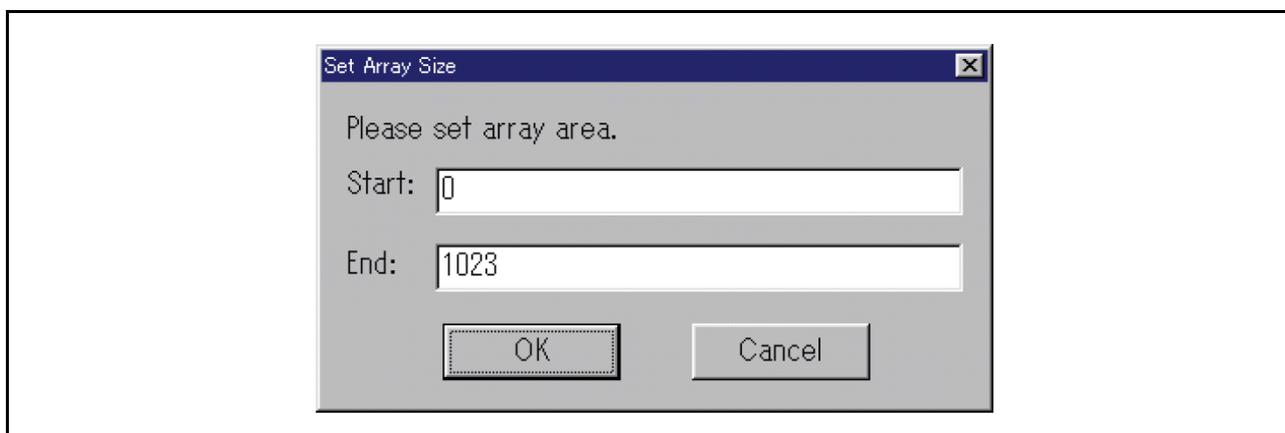
- 当计算结果为 char [] 类型时，其显示如下：

(unsigned char [10]) str = 0x1234 "Japan" (所有基数)

如果字符串中某个符号前包含一个非打印符号，用以显示字符串结束符 (0)，则字符串仅显示到此非打印字符位置，并且不显示右引号。

(unsigned char [10]) str = 0x1234 "Jap (所有基数)

如果字符串字符个数超过 80，同样也不会显示右引号。当 C/C++ 表达式是数组类型时，与指针类型一样，类型名称左侧将显示 “+”。您可以使用此指示符号查看数组的元素。（有关详细信息，请参阅 “10.2.3 指针类型”）当数组元素个数超过 100 时将打开如下对话框。请在对话框中指定元素个数。



将会显示此处指定的从 “Start”（开始）至 “End”（结束）的下标所对应的元素。如果您指定的值超出该数组的最大下标，则该值将被视作数组的最大下标。如果您单击 “Cancel”（取消）按钮，则不会显示元素。

10.2.5 函数类型

- 当计算结果为函数数据类型时，起始地址将以十六进制形式显示如下：

(void()) main = 0xF000 (所有基数)

10.2.6 引用类型

- 当计算结果为引用数据类型时，引用地址将以十六进制形式显示如下：

(signed int &) ref = 0xD038 (所有基数)

10.2.7 位域类型

- 当计算结果为位域数据类型时，其显示如下：

(unsigned int :13) s.f = 8191 (如果基数为初始状态)

(unsigned int :13) s.f = 0x1FFF (如果基数为十六进制数)

(unsigned int :13) s.f = 11111111111111B (如果基数为二进制数)

10.2.8 未找到任何 C 符号时

如果所计算的表达式包含找不到的 C 符号，其显示如下：

() x = <not active> (所有基数)

10.2.9 语法错误

当所计算的表达式包含语法错误时，其显示如下：

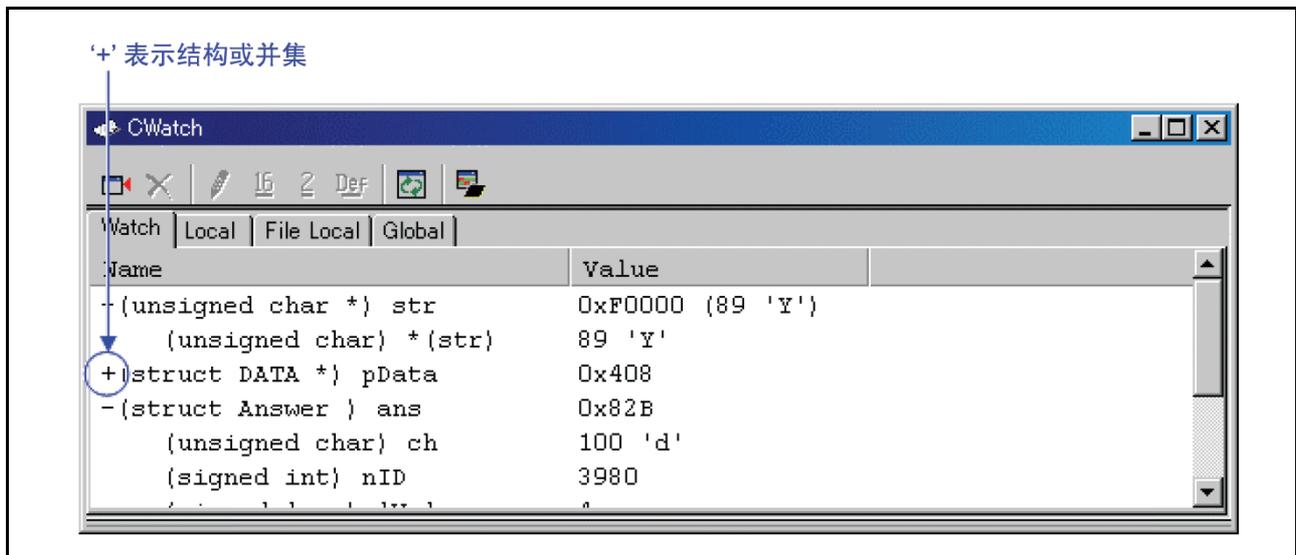
() str*(p = <syntax error> (所有基数)
(其中 str*(p 为该语法错误)

10.2.10 结构类型和联合类型

当计算结果为结构或联合数据类型时，地址将以十六进制形式显示如下：

(Data) v = 0x1234 (所有基数)

如果 C/C++ 表达式像结构或联合一样包含成员，则类型名称（标记名称）左侧会显示“+”。



您可以双击由“+”指示的行，查看该结构或联合的成员。显示成员时“+”变为“-”。要恢复初始显示，双击现由“-”指示的那一行。您可以使用此功能查看结构或联合的成员。

注意事项

如果声明的变量与通过 typedef 声明的类型定义同名，则您无法引用该变量。

- 寄存器变量

当计算结果为寄存器变量时，类型名称左侧将显示“register”，如下所示：

(register signed int) j = 100

11 显示程序停止的原因

如果程序被调试功能停止，则停止原因将显示在 [Output]（输出）窗口或 [Status]（状态）窗口（[Platform]（平台）页）中。

显示内容及“停止原因”说明如下所示：

显示内容	停止原因
Halt	由 [Halt Program]（暂停程序）按钮 / 菜单停止
S/W break	软件断点
Address match interrupt break	地址中断断点
H/W event, Combination	硬件断点，符合逻辑组合 [AND]（与）或 [AND(same time)]（与（同时））条件
H/W event, Combination, Ax	硬件断点，符合逻辑组合 [OR]（或）条件（Ax: 符合条件的事件编号。）
H/W event, State transition, from xx	硬件断点，符合状态转换条件（from xx: 上一状态（start、state1、state2））
H/W event, State transition, Timeout	硬件断点，符合 [State Transition]（状态转换）和 [Time Out]（超时）条件
H/W event, Access protect error	中断保护

注意

能否显示中断原因取决于所连接的目标。一些目标可能始终显示“Halt”（暂停）或者显示“---”。

12 注意事项

12.1 通用注意事项

12.1.1 与 Windows 相关的文件操作

1. 文件名和目录名称
 - 如果目录名和文件名中包含日文汉字，则不保证操作正常。
 - 文件名中只能包含一个句点。
2. 指定文件和目录
 - 不能使用“...”指定两层以上的目录。
 - 不能使用网络路径。必须分配一个驱动器号。

12.1.2 可设置软件断点的区域

可设置软件断点的区域因 MCU 类型的不同而异。

12.1.2.1 用于 M32C 的调试器

可设置软件断点的区域因处理器模式的不同而异。

处理器模式	可设置区域
单片	内部 RAM、内部 ROM
内存扩展	内部 RAM、内部 ROM 仿真存储器（仅限内部区域）
微处理器	内部 RAM 仿真存储器（仅限内部区域）

要停止仿真存储区域以外的目标程序，请使用 COME 执行。

12.1.2.2 用于 M16C/R8C 的调试器

可设置软件断点的区域因处理器模式的不同而异。

处理器模式	可设置区域
单片	内部 RAM 区域 内部 ROM 区域
内存扩展	内部 RAM 区域 内部 ROM 区域 仿真存储器（仅限内部区域）
微处理器	内部 RAM 区域 仿真存储器（仅限内部区域）

注意事项

在目标程序执行期间，不能在内部 ROM 区域设置软件断点。

12.1.3 获取或设置 C 变量

- 如果声明的变量与通过 `typedef` 声明的类型定义同名，则您无法引用该变量。
- 无法更改寄存器变量的值。
- 无法更改 64 位宽变量（`long long`、`double` 等）的值。
- 无法更改未指定存储地址和大小的 C/C++ 表达式的值。
- 出于优化考虑，C 编译器可能在相同地址放置不同的变量。在这种情况下，C 变量的值可能无法正常显示。
- 文字字符串只能代换 `char` 数组和 `char` 指针类型的变量。
- 对浮点类型无法执行任何数学运算。
- 对浮点类型无法执行任何取反操作。
- 对浮点类型无法执行转换。
- 对寄存器变量无法执行转换。
- 无法转换结构类型或联合类型，无法将指针类型转换成结构类型或联合类型。
- 字符常数和文字字符串不能包含转义序列。
- 以下值可代换位域。
 - 整数常数、字符常数和枚举数
 - 布尔类型、字符类型、整数类型以及枚举类型变量。
 - 位域

如果所代换的值超出位域大小，则会将该值截断。

- 在 SFR 区域中分配的位域成员可能无法转换为正确的值。
- 目标程序正在运行时，无法修改局部变量和位域的值。

12.1.4 C++ 中的函数名

- 进行显示地址、断点等设置时，如果您使用函数名输入地址，那么您不能指定类的成员函数、运算符函数和重载函数。
- 不能在 C/C++ 表达式中使用函数名
- 为参数所指定的函数名中不能使用任何脚本命令（例如，`breakin` 和 `func`）。
- 在对话框的地址值指定列中，任何地址均不能使用函数名进行指定。
- 在 [C watch]（C 监视）窗口中，用于成员函数的指针无法正确引用。

12.1.5 下载模块的选项设置

以下可在 [Debug Settings]（调试设置）对话框中设置的选项对此调试器无效：

- Offset（偏移量）：已指定的值将被视作“0”
- Access size（存取大小）：已指定的值将被视作“1”
- Perform memory verify during download（下载期间执行存储器验证）：不支持。

12.1.6 调试多个模块

如果在单个会话中注册两个或两个以上绝对模块文件，您一次只能下载一个文件。

如果在单个会话中注册一个绝对模块文件和一个或多个机器语言文件，您可以同时下载所有文件。

12.1.7 同步调试

同步调试功能不可用。

12.1.8 下载固件

要启动调试器，必须在仿真器上下载对应于已连接仿真主机或仿真探针的固件。

- 更换了仿真主机和仿真探针。
- 下载到仿真器的固件是未知固件。
- 首次使用调试器。
- 调试器软件已升级。

遇到上述情况之一时，您必须执行以下操作才能启动调试器。

打开仿真器电源开关后按住仿真器系统复位开关 2 秒。

随后仿真器将进入一种强制下载固件的模式。

12.1.9 LPT 端口限制

1. 仿真器使用打印机（并行）端口，通过 LPT 与个人计算机通信。

IAR C 编译器也使用此打印机（并行）端口。

进行 LPT 通信期间，当使用个人计算机和仿真器时会出现一个问题，即您无法使用 IAR C 编译器来编译程序。

如果出现这种情况，您必须采取下列应对措施之一：

- 通过非“ECP”的其他模式将个人计算机连接到仿真器。
 - 终止调试器后再开始对程序进行编译。
2. 使用运行 Windows XP 的主 PC 上的 LPT 并行接口调试器时，可能会出现以下现象：
 - 调试器不运作。
 - 调试器的操作异常缓慢。
 - 出现通信错误且仿真调试器停止操作。
 - 接连出现除通信错误以外的各类错误。但仿真调试器仍可在这种情况下继续工作。

导致这一问题的原因在于，当 Windows XP 的标准驱动程序 Parport.sys 正与 LPT 端口通信时，如果此时调用调试器，则调试器无法与仿真器通信。可采取以下方法之一避免这种情况（如果您的 PC 正在进行以上所述的通信但并未出现问题，则无需采取任何措施）：

- 执行实用程序 (LptFix.exe) 进行修复。
LptFix.exe 将禁止 Parport.sys 驱动程序。执行 LptFix.exe 后，如果除仿真器以外的其他器件连接到 LPT 端口，该器件可能无法正常操作。在这种情况下，建议您采取以下方法。
- 如果是启动 PC 后第一次启动仿真器，请在仿真器开始操作一分钟或更长时间后再调用调试器。
也可以在仿真器启动后立即调用调试器。

下面介绍 LptFix.exe 的使用：

1. 调用命令提示符，并转到 LptFix.exe 文件所在的文件夹。

在命令行中输入以下命令：

```
-----  
>LptFix  
-----
```

2. 重新启动 PC。

如果执行了 LptFix.exe，请按以下所述的方式，确保每次结束调试操作并断开仿真器连接后，该实用程序无法启用 Parport.sys 驱动程序。要使 LptFix.exe 无法启用 Parport.sys，请执行以下命令行：

```
-----
>LptFix /U
-----
```

12.1.10 覆盖率功能的注意事项

覆盖率计量功能将记录目标程序所存取的地址。

这些存取的地址将沿微型计算机的地址总线传送。

这可能导致计算覆盖率时将未执行的地址视作已执行的地址。

12.1.11 仿真器复位开关

如果仿真器的系统复位功能不正常，请终止调试器，再次打开仿真器的开关，然后重新启动调试器。重新下载程序。

12.1.12 仿真器上的调试资源

仿真器上的调试资源在窗口和对话框间是共享的。

因此，一次只能使用一个窗口或对话框。

跟踪事件	[Trace Point Setting] (跟踪点设置) 窗口 (TracePoint 命令) [MR Trace] (MR 跟踪) 窗口 / [MR Analyze] (MR 分析) 窗口 [Task Trace] (任务跟踪) 窗口 / [Task Analyze] (任务分析) 窗口 [Time Measurement] (时间计量) 窗口
------	--

12.1.13 启动仿真器和调试器

请在 PC7501 的目标状态 LED (错误) 熄灭后再启动调试器，因为 PC7501 在刚打开电源后开始运行，此时 PC7501 不会与调试器进行通信。

12.1.14 使用任务暂停功能和地址匹配中断断点功能

任务暂停功能需使用地址匹配中断断点功能。因此，当目标程序使用地址匹配中断时，您便无法使用任务暂停功能。

同样，当您使用地址匹配中断断点功能时，无法使用任务暂停功能。

12.1.15 覆盖率计量的结果

在 [Coverage] (覆盖率) 窗口和 CoVerage FUNC 脚本命令中，如果某函数的地址范围超出覆盖率计量区域的界限，则该函数的覆盖率计量结果不会正常显示 (总是显示为 "****")。

此外，当 CoVerage TOTAL 命令指定的地址范围对应于前面提到的条件时，将显示每一个块的覆盖率结果。

12.2 M32C 调试器的注意事项

12.2.1 仿真器使用的堆栈区域

仿真器将中断堆栈区域用作工作区域（20 字节）。

开始调试之前，请确保保留用户堆栈区域，再外加 20 字节区域。

12.2.2 目标程序复位时的中断堆栈指针

仿真器在复位目标程序时将中断堆栈指针 (ISP) 设置为 0500h。

在实际的仿真器上，中断堆栈指针 (ISP) 将被设置为 0000h。

12.2.3 C 编译器 / 汇编器 / 连接器选项

根据编译器、汇编器和连接器的选项指定，有时无法正常下载 / 调试信息。

请参阅以下内容了解选项规范。

请参阅“12.4 C 编译器 / 汇编器 / 连接器选项”。

M32C 调试器可使用的编译器：

- NCxx
- IAR EC++ 编译器
- IAR C 编译器

12.2.4 目标 MCU HOLD 终端

MCU HOLD 终端设置为 LOW 时，您无法停止执行目标程序。请将 HOLD 终端设置为 High 并再次尝试停止目标程序。

即使 HOLD 终端设置为 LOW 的时间很短，也可以在这种情况下停止目标程序。

如果出现此情况，请尝试再次停止目标程序。

12.2.5 硬件事件

- 调试器不支持将取命令 (Fetch) 或中断视作硬件事件。
- 如果从奇地址指定单字长（2 字节长度）数据作为后续数据存取的事件，则该事件不会被检测到。而且，在位存取过程中，对包含指定地址的其他位进行存取时，该事件才可能生效。
 - 硬件断点事件
 - 实时跟踪事件
 - 时间计量事件
- 在 PID 设置中，始终确保为 [Address]（地址）区域指定一个偶地址。
- 关于指定数据存取事件的示例及其值的对比情况

地址	存取大小	16 位数据总线	8 位数据总线
偶地址	WORD 示例: mov.w #1234h, 8000h	地址: 8000h 数据: 1234h 数据掩码: 无关	<-
偶地址	BYTE 示例: mov.b #34h, 8000h	地址: 8000h 数据: 34h 数据掩码: 00FFh	<-
奇地址	WORD 示例: mov.w #1234h, 8001h	不支持	<-
奇地址	BYTE 示例: mov.b #34h, 8001h	地址: 8001h 数据: 3400h 数据掩码: FF00h	地址: 8001h 数据: 34h 数据掩码: 00FFh

12.2.6 CPU 重写

- 使用 M32C/8x 系列 CPU 时, 请不要重写块 0。否则仿真器将不受控制。
- 使用 M16C/70 系列的 CPU 时, 只有最高有效地址, 即末尾块 4KB (FFF000h - FFFFFFFh) 的地址变为 FFFFAh 才能使用仿真器。请在使用软件命令时加以注意。并且不要对 FFFFFCh - FFFFFFFh 使用软件命令 “program”。
- 启用 CPU 重写模式后, 您将无法使用以下调试功能:
 - 设置地址匹配中断断点
 - 向内部闪存 ROM 设置软件断点
 - 对内部闪存 ROM 进行 COME 执行
- 您不能对 CPU 重写控制程序使用以下功能 (从设置 CPU 重写模式选择位到将其清除)。否则可能无法切换到 CPU 重写模式, 或者闪存 ROM 的内容可能无法正常读取。
 - 单步执行
 - 软件断点
 - 硬件断点
 - COME 执行

要查看由程序重写的的数据, 请将程序与 CPU 重写控制程序断开, 并通过转储、存储器窗口等查看。

12.2.7 MR STK 脚本命令

- MR STK BASE 命令会自动设置可在其中计量堆栈使用量的区域。由此命令设置的区域大小为 256KB, 包括系统堆栈的起始地址。如果您想设置其他区域, 则不要使用 MR STK BASE 脚本命令, 请使用 CoVerage BASE 脚本命令。
- 在 MR STK SYS 和 MR STK TSK 脚本命令中, 如果堆栈区域的地址范围超出覆盖率计量区域的界限, 那么堆栈区域的覆盖率计量结果可能无法正确显示 (总是显示为 “*****”)。

12.3 M16C/R8C 调试器的注意事项

12.3.1 连接到仿真器

通过 [MCU Setting] (MCU 设置) 对话框设置 MCU 之后, 需要 10-30 秒时间启动调试器。设置了 MCU 之后, 请稍候片刻。

12.3.2 仿真器使用的堆栈区域映像

仿真器将中断堆栈区域用作其工作区域 (20 字节)。

进行调试时, 请分配足够的中断堆栈区域, 该区域为常规使用的大小外加 20 字节。

12.3.3 目标程序复位时的中断堆栈指针

仿真器在复位目标程序时将中断堆栈指针 (ISP) 设置为 0500h。

在实际的仿真器上, 中断堆栈指针 (ISP) 将被设置为 0000h。

12.3.4 编译器、汇编器和连接器选项

根据编译器、汇编器和连接器的选项指定, 有时无法正常下载 / 调试信息。

请参阅以下内容了解选项规范。

请参阅 “12.4 C 编译器 / 汇编器 / 连接器选项”。

M16C/R8C 调试器可使用的编译器:

- NCxx
- IAR EC++ 编译器
- IAR C 编译器
- TASKING C 编译器

12.3.5 TASKING C 编译器

调试由 TASKING C 编译器 “CCM16” 编译的程序时, 位域的类型固定为 “unsigned short int” (无符号短整数), 因为 CCM16 将以 “unsigned short int” 的形式输出位域类型的调试信息。

12.3.6 目标 MCU HOLD 终端

MCU HOLD 终端设置为 LOW 时, 您无法停止执行目标程序。请将 HOLD 终端设置为 High 并再次尝试停止目标程序。即使 HOLD 终端设置为 LOW 的时间很短, 也可以在这种情况下停止目标程序。如果出现此情况, 请尝试再次停止目标程序。

12.3.7 硬件事件

- 如果从奇地址指定单字长（2 字节长度）数据作为后续数据存取的事件，则该事件不会被检测到。而且，在位存取过程中，对包含指定地址的其他位进行存取时，该事件才可能生效。
 - 硬件断点事件
 - 实时跟踪事件
 - 时间计量事件
- 在 PID 设置中，始终确保为 [Address]（地址）区域指定一个偶地址。
- 关于指定数据存取事件的示例及其值的对比情况

地址	存取大小	16 位数据总线	8 位数据总线
偶地址	WORD 示例: mov.w #1234h, 8000h	地址: 8000h 数据: 1234h 数据掩码: 无关	<-
偶地址	BYTE 示例: mov.b #34h, 8000h	地址: 8000h 数据: 34h 数据掩码: 00FFh	<-
奇地址	WORD 示例: mov.w #1234h, 8001h	不支持	<-
奇地址	BYTE 示例: mov.b #34h, 8001h	地址: 8001h 数据: 3400h 数据掩码: FF00h	地址: 8001h 数据: 34h 数据掩码: 00FFh

12.3.8 任务暂停功能的对应操作系统版本

如果在相关系统（装有支持任务暂停功能的 MR30 V.3.00 或更高版本）中下载被编译目标程序，则任务暂停功能可用。

如果 MR30 的版本为 V.3.00 Release 1，可能出现以下现象。

- 如果程序将诸如 rel_wai 或 irel_wai 之类的系统调用发出到任务，任务暂停功能将该任务更改为“Pause”（暂停），而使任务的“Pause”状态复位。

在这种情况下，[MR Task Pause]（MR 任务暂停）窗口所显示的任务信息与实际任务状态有异。在装有 MR30 V.3.10 Release 1 或更高版本的目标程序中不会出现此现象。

12.3.9 存储空间扩展

存储器映像

在 [MCU setting] (MCU 设置) 对话框中, 如果对 Memory Space Expansion (存储空间扩展) 选择 “4MB Mode” (4MB 模式), 则存储器映像取决于 mcu 的其他设置:

处理器模式	PM13	OFS	MCU 的存取区域	Bank0 - Bank5	Bank6	Bank7
存储器扩展模式	1	0	40000h - 7FFFFh	EXT	EXT	MAP
		1	40000h - 7FFFFh	EXT	EXT	MAP
	0	0	40000h - 7FFFFh	EXT	EXT	MAP
			80000h - BFFFFh	EXT	EXT	MAP
		1	40000h - 7FFFFh	EXT	EXT	MAP
			80000h - BFFFFh	EXT	EXT	---
微处理器模式	---	0	40000h - 7FFFFh	EXT	EXT	MAP
			80000h - BFFFFh	EXT	EXT	---
			C0000h - FFFFFh	---	---	MAP
		1	40000h - 7FFFFh	EXT	EXT	MAP
			80000h - BFFFFh	EXT	EXT	---

PM13: 处理器模式 register1 (00005h) 的 bit3

OFS: 偏移位 (数据存储体寄存器 (0000Bh) 的 bit2)

EXT: 对目标系统的存储存取

MAP: 对区域的存储存取, 取决于 [MCU Settings] (MCU 设置) 对话框中的 MAP setting (MAP 设置) (INT 代表 PC7501 中的仿真存储器, EXT 代表目标系统)

对扩展区域的存储存取

[C watch] (C 监视) 窗口、[Memory] (存储器) 窗口以及其他调试窗口在显示存储空间扩展功能的扩展区域时, 其中显示的值可能不正确。调试窗口不会考虑到寄存器体的状态。

要查看此区域, 请使用下面列出的脚本命令, 并为 M16C 指定存储体、偏移位状态和地址。

DumpByte2、DumpWord2 或 DumpLword2

您还可以使用以下命令:

DumpByte2、DumpWord2、DumpLword2、SetMemoryByte2、SetMemoryWord2、SetMemoryLword2、FillByte2、FillWord2、FillLword2、Move2、MoveWord2

注意

下列功能在对扩展区域进行存取时可能无法按预期运行。调试器无法区分存储体信息和 BUS 信息。

- RAM 监控功能 ([RAM Monitor] (RAM 监控) 窗口、[C watch] (C 监视) 窗口, 等等)
- 覆盖率计量功能 ([Coverage] (覆盖率) 窗口、Coverage 命令, 等等)
- 存储保护功能 ([Protect] (保护) 窗口、Protect 命令, 等等)
- 硬件事件 (硬件断点事件、实时跟踪事件、时间计量事件, 等等)

关于硬件事件, 您可以指定所需事件的 SameAND 组合以及存储体选择寄存器的状态, 以检测对收集存储体的存取。

12.3.10 CPU 重写

- 请勿重写最后一个块。否则，仿真器将不受控制。
- 启用 CPU 重写模式后，您将无法使用以下调试功能：
 - 向内部闪存 ROM 设置软件断点
 - 对内部闪存 ROM 进行 COME 执行
- 您不能对 CPU 重写控制程序使用以下功能（从设置 CPU 重写模式选择位到将其清除）。否则可能无法切换到 CPU 重写模式，或者闪存 ROM 的内容可能无法正常读取。
 - 单步执行
 - 软件断点
 - 地址匹配中断断点
 - 硬件断点
 - COME 执行

要查看由程序重写的的数据，请将程序与 CPU 重写控制程序断开，并通过转储、存储器窗口等查看。

12.3.11 MR STK 脚本命令

- 在 MR STK SYS 和 MR STK TSK 脚本命令中，如果堆栈区域的地址范围超出覆盖率计量区域的界限，那么堆栈区域的覆盖率计量结果可能无法正确显示（总是显示为“*****”）。

12.4 C 编译器 / 汇编器 / 连接器选项

我们未对其他设置进行评估，因此无法建议附加其他选项。

12.4.1 使用 NCxx

编译时如果指定了 -O、-OR 或 -OS 选项，出于优化考虑，可能不会正常生成源代码行信息，从而导致单步执行出现异常。

要避免此问题，请在指定 -O、-OR 或 -OS 选项的同时指定 -ONBSD（或 -Ono_Break_source_debug）选项。

12.4.2 使用 IAR EC++ 编译器 (EW)

请按如下步骤指定工程设置：

1. IAR Embedded Workbench 中的设置

选择 [Project]（工程）->[Options...]（选项...）菜单时，将打开 [Options For Target “target”]（“target” 目标选项）对话框。在此对话框中，请选择 “XLINK” 类别，然后设置工程设置。

— Output（输出）选项卡

在 “Format”（格式）区域中，选择 “Other”（其他）选项，然后选择 “elf/dwarf” 作为 “Output Format”（输出格式）。

— Include（包含）选项卡

在 “XCL File Name”（XCL 文件名）区域中，指定您的 XCL 文件（例如，lnkm32cf.xcl）。

2. 编辑 XCL 文件

将命令行选项 “-y” 添加到 XCL 文件。对 “-y” 选项的指定因产品而异。

产品名称	-y 选项
用于 M32C 的调试器	-yspc
用于 M16C/R8C 的调试器	-yspc

3. 完成上述设置后构建程序。

我们未对其他设置进行评估，因此无法建议附加其他选项。

12.4.3 使用 IAR C 编译器 (EW)

请按如下步骤指定工程设置：

1. IAR Embedded Workbench 中的设置

选择 [Project]（工程）->[Options...]（选项...）菜单时，将打开 [Options For Target “target”]（“target” 目标选项）对话框。在此对话框中，请选择 “XLINK” 类别，然后设置工程设置。

— Output（输出）选项卡

在 “Format”（格式）区域，选中 “Other”（其他）选项，然后选择 “ieee-695” 作为 “Output Format”（输出格式）。

— Include（包含）选项卡

在 “XCL File Name”（XCL 文件名）区域中，指定您的 XCL 文件（例如，lnkm16c.xcl）。

2. 编辑 XCL 文件

将命令行选项 “-y” 添加到 XCL 文件。对 “-y” 选项的指定因产品而异。

产品名称	-y 选项
用于 M32C 的调试器	-ylmb
用于 M16C/R8C 的调试器	-ylmb

3. 完成上述设置后构建程序。

在除上述选项之外的选项中，不会进行操作检查。请注意，除上述介绍的选项外不推荐使用其他选项。

12.4.4 使用 IAR C 编译器 (ICC)

12.4.4.1 指定选项

请根据以下步骤进行编译和连接。

- 编译时

指定“-i”选项

- 连接前

进行连接时，请打开要读取的连接选项定义文件 (extension .xcl) 并添加“-FIEEE695”和“-y”选项。

对“-y”选项的指定因产品而异。

产品名称	-y 选项
用于 M32C 的调试器	-ylmb
用于 M16C/R8C 的调试器	-ylmb

- 连接时

使用“-f”选项指定连接器的选项定义文件。

在除上述选项之外的选项中，不会进行操作检查。请注意，除上述介绍的选项外不推荐使用其他选项。

12.4.4.2 命令执行示例

以下示例显示如何针对不同的产品输入不同的命令

- 用于 M32C 的调试器

```
>ICCMC80 -r file1.c<Enter>
>ICCMC80 -r file2.c<Enter>
>XLINK -o filename.695 -f lnkm80.xcl file1 file2<Enter>
```

- 用于 M16C/R8C 的调试器

```
>ICCM16C -r file1.c<Enter>
>ICCM16C -r file2.c<Enter>
>XLINK -o filename.695 -f lnkm16c.xcl file1 file2<Enter>
```

XCL 文件名因不同的产品和存储器型号而异。有关详细信息，请参阅 ICCxxxx 手册。

12.4.5 使用 TASKING C 编译器 (CM)

12.4.5.1 指定选项

请在编译时指定“-g”和“-O0”选项。

在除上述选项之外的选项中，不会进行操作检查。请注意，除上述介绍的选项外不推荐使用其他选项。

12.4.5.2 命令执行示例

下面显示了输入命令的示例。

```
>CM16 -g -O0 file1.c<Enter>
```

12.4.5.3 使用 TASKING C 编译器 (EDE)

请按如下步骤指定工程设置：

1. 选择菜单 - [EDE]->[C Compiler Option] (C 编译器选项) ->[Project Options...] (工程选项...)。随后将打开 [M16C C Compiler Options [Project Name]] (M16C C 编译器选项 [工程名称]) 对话框。

请对此对话框进行如下设置。

— Optimize (优化) 选项卡

请对 Optimization level (优化级别) 指定 “No optimization” (无优化)。

— Debug (调试) 选项卡

仅选中 “Enable generation of any debug information(including type checking)” (允许生成所有调试信息 (包括类型检查)) 和 “Generate symbolic debug information” (生成符号调试信息)。

2. 选择菜单 - [EDE]->[Linker/Locator Options...] (连接器/定位器选项...)。随后将打开 [M16C Linker/Locator Options [Project Name]] (M16C 连接器/定位器选项 [工程名称]) 对话框。

请对此对话框进行如下设置。

— Format (格式) 选项卡

请对 Output Format (输出格式) 指定 “IEEE 695 for debuggers(abs)”。

3. 完成上述设置后构建程序。

在除上述选项之外的选项中，不会进行操作检查。请注意，除上述介绍的选项外不推荐使用其他选项。

[备注]

M16C R8C PC7501 仿真调试器 V.1.03 用户手册

Publication Date: Rev.1.00, Sep. 28, 2007

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIACenter, No.1233 Lujiiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



M16C R8C PC7501 仿真调试器 V.1.03



瑞萨电子株式会社