

E1/E20 仿真器

用户手册附加文档（连接时的注意事项）

对象器件

RX族 / RX600系列

RX610、RX621、RX62N、RX62T群

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。
请通过瑞萨电子的主页确认发布的最新信息。

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.

目 录

1.	E1/E20 仿真器手册的构成	1
2.	用户系统的设计	2
2.1	E1/E20 仿真器和用户系统的连接	2
2.2	安装在用户系统上的仿真器连接器	3
2.3	仿真器连接器的引脚排列	6
2.4	仿真器连接器和单片机之间的推荐连接例子	9
2.4.1	推荐连接例子（14 引脚连接器）	9
2.4.2	推荐连接例子（38 引脚连接器）	11
2.4.3	连接时的注意事项	13
3.	调试的准备	22
3.1	工作环境	22
3.2	High-performance Embedded Workshop 的启动	23
3.3	新工作空间的建立（不使用工具链）	24
3.4	新工作空间的建立（使用工具链）	26
3.5	现有工作空间的打开	28
3.6	仿真器的连接	29
3.6.1	仿真器的连接	29
3.6.2	仿真器的重新连接	29
3.7	仿真器的断开	29
3.7.1	仿真器的断开	29
3.8	High-performance Embedded Workshop 的结束	29
3.9	调试设置	30
3.9.1	下载模块的指定	30
3.9.2	命令行批文件的自动执行的设定	31
3.10	E1/E20 仿真调试程序的启动步骤	32
4.	使用 RX610 群、RX621 群、RX62N 群、RX62T 群时的软件规格	39
4.1	E1/E20 仿真器规格一览表	39
4.2	E1/E20 仿真器和单片机的不同点	41
4.3	调试程序的设定	43
4.3.1	[Initial Settings]（初始设置）对话框	43
4.3.2	[Configuration Properties]（配置属性）对话框	46
5.	E1/E20 仿真器的功能	52
5.1	功能概要	52
5.2	下载功能	54
5.3	源文件的打开	54
5.3.1	源码的显示	54
5.3.2	汇编语言代码的显示	56
5.3.3	汇编语言代码的修改	56
5.4	存储器的存取功能	57
5.4.1	存储器的读写功能	57
5.5	暂停功能	58
5.5.1	强制暂停功能	58
5.5.2	S/W（软件）暂停功能	58
5.5.3	On-chip 暂停功能	58
5.6	事件功能	59
5.6.1	事件的使用	59

5.6.2	事件的添加	60
5.6.3	事件的删除	65
5.6.4	事件的注册	66
5.6.5	注册事件的删除	70
5.6.6	事件的随时输入 / 再次使用	71
5.6.7	事件的应用	72
5.7	On-Chip Break (On-chip 暂停) 功能	73
5.7.1	On-Chip Breakpoint (On-chip 断点) 的设定	73
5.7.2	On-chip 暂停设定内容的保存 / 加载	77
5.8	跟踪功能	78
5.8.1	跟踪信息的查看	79
5.8.2	跟踪信息的获取	79
5.8.3	跟踪结果的显示	82
5.8.4	选项菜单	86
5.8.5	跟踪信息获取条件的设定	87
5.8.6	跟踪信息的文件保存	91
5.8.7	跟踪信息的文件加载	91
5.8.8	跟踪信息获取的暂停	91
5.8.9	跟踪信息获取的重新开始	91
5.9	性能测量功能	92
5.9.1	性能测量	92
5.9.2	性能测量结果的显示	92
5.9.3	性能测量条件的设定	94
5.9.4	性能测量的开始	98
5.9.5	性能测量条件的解除	98
5.9.6	性能测量结果的清除	98
5.9.7	性能测量的最大次数	98
5.9.8	性能条件设定内容的保存和加载	98
5.10	实时 RAM 监视功能	99
5.10.1	RAM 监视功能的使用	99
5.10.2	RAM 监视范围的分配	99
5.10.3	监视内容的显示	101
5.10.4	有关丢失	102
5.10.5	丢失的抑止	103
5.11	Start/Stop (启动 / 停止) 功能	105
5.11.1	[Start/Stop function setting] (启动 / 停止功能设置) 对话框的打开	105
5.11.2	执行例行程序的指定	105
5.11.3	Start/Stop (启动 / 停止) 功能的限制事项	106
5.11.4	有关指定例行程序记述的限制事项	106
5.12	调试控制台功能	107
5.12.1	[Debug Console] (调试控制台) 窗口的打开	107
5.12.2	选项菜单	108
5.12.3	有关低水平接口例行程序	109
5.13	热插拔功能	111
5.13.1	启动步骤	111
5.13.2	热插拔的注意事项	115
5.14	堆栈跟踪功能	115
5.15	在线帮助	115
6.	教程	116
6.1	绪论	116
6.2	High-performance Embedded Workshop 的启动	116

6.3	仿真器的连接	117
6.4	教程程序的下载	117
6.4.1	教程程序的下载	117
6.4.2	源程序的显示	118
6.5	S/W 断点的设定	119
6.6	程序的执行	120
6.6.1	CPU 的复位	120
6.6.2	程序的执行	120
6.7	断点的确认	121
6.7.1	断点的确认	121
6.8	寄存器内容的更改	122
6.9	符号的参照	123
6.10	存储器内容的确认	124
6.11	变量的参照	125
6.12	局部变量的显示	127
6.13	程序的单步执行	128
6.13.1	Step In (步进) 的执行	128
6.13.2	Step Out (步出) 的执行	129
6.13.3	Step Over (跨步) 的执行	130
6.14	程序的强制暂停	131
6.15	On-chip 暂停功能	132
6.15.1	执行指定地址时的程序停止执行	132
6.16	存取存储器时的程序停止执行	133
6.17	跟踪功能	134
6.17.1	Fill until Stop (填充直至停止) 跟踪的跟踪信息显示	135
6.18	堆栈跟踪功能	137
6.19	结尾	138
7.	使用时的注意事项	139
7.1	存储器	139
7.1.1	I/O 寄存器区域	139
7.1.2	内部闪存 ROM 区	139
7.1.3	内部闪存 ROM 的下载	139
7.1.4	内部闪存 ROM 的改写	139
7.1.5	FCU-RAM、FCU 固件区	139
7.1.6	调试程序使用的作业 RAM 区	139
7.2	[Memory] (存储器) 窗口	140
7.2.1	复制、比较和检索功能	140
7.2.2	选项菜单	140
7.3	IO 窗口	140
7.3.1	I/O 寄存器文件的自定义	140
7.3.2	验证	140
7.4	执行操作	140
7.4.1	[Go To Cursor] (转至光标)	140
7.4.2	[Run Program] (执行程序)	140
7.4.3	[Step Out] (步出) 执行	140
7.5	复位	141
7.5.1	由仿真器系统进行的操作和单片机复位的竞争	141
7.5.2	使用跟踪功能时的单片机复位	141
7.5.3	使用实时 RAM 监视功能时的单片机复位	141
7.5.4	在内部 ROM 无效扩展模式中执行时的单片机复位	141
7.6	跟踪功能	141

7.6.1	能记录的存取	141
7.6.2	跟踪信息	141
7.7	事件功能	142
7.7.1	能检测的存取	142
7.7.2	事件组合	142
7.7.3	通过次数条件	142
7.7.4	数据存取事件的地址范围条件	142
7.7.5	跟踪获取条件设定	142
7.7.6	有关事件注册	142
7.8	暂停功能	143
7.8.1	设定断点时的注意事项	143
7.9	实时 RAM 监视功能	143
7.10	性能测量功能	144
7.10.1	有关性能测量过程中的复位	144
7.10.2	性能测量过程中的嵌套限制	144
7.10.3	选定 [Measure the performance only once.] (仅测量一次) 时的注意事项	144
7.10.4	有关事件的处理	144
7.11	下载功能	145
7.11.1	有关存取长度	145
7.11.2	内部 ROM 无效状态的下载	145
7.12	外部闪存的下载	146
7.13	执行时间	147
7.14	Start/Stop (启动 / 停止) 功能	147
7.15	监视功能	147
7.16	其他	147
7.16.1	改写闪存时的寄存器值	147
7.16.2	有关 DMAC 和 DTC	147
7.16.3	锁定位的解除	147
附录	148
附录 1.	菜单一览表	148
附录 2.	High-performance Embedded Workshop 的注意事项	151

1. E1/E20 仿真器手册的构成

E1/E20 仿真器手册由 E1/E20 仿真器用户手册以及对应各单片机的 E1/E20 仿真器用户手册附加文档（本手册）构成。

在使用 E1/E20 仿真器前，请务必阅读这 2 本用户手册。

(1) E1/E20 仿真器用户手册

在 E1/E20 仿真器用户手册中记载了硬件规格。

- E1/E20 仿真器的构成部件
- E1/E20 仿真器的硬件规格
- E1/E20 仿真器和主机、用户系统的连接

(2) E1/E20 仿真器用户手册附加文档

在 E1/E20 仿真器用户手册附加文档中记载了仿真调试程序的功能说明和操作方法以及取决于各单片机的内容和注意事项。

- E1/E20 仿真器的规格
- 设计硬件时所需的 E1/E20 仿真器连接例子和接口电路
- 仿真调试程序的各功能说明
- 仿真调试程序的操作方法
- 使用 E1/E20 仿真器时的注意事项
- 使用 RXxxx 群单片机时的软件规格等
- 从启动 E1/E20 仿真调试程序开始到调试作业为止的教程等

2. 用户系统的设计

2.1 E1/E20 仿真器和用户系统的连接

为了连接 E1/E20 仿真器，需要将用于连接用户接口电缆【注】的仿真器连接器安装到用户系统上。
在设计用户系统时，请务必仔细阅读本章节和所用单片机的硬件手册。

【注】 在从仿真器和用户系统的连接器拔插用户接口电缆时，必须拿着电缆两端连接器的外壳进行拔插。如果不拿连接器外壳而拉扯电缆部分，就会引起电缆断线。

E1/E20 仿真器产品型号及其对应的连接器类型、外部跟踪输出功能的使用 / 不使用的关系如表 2.1 所示。

表 2.1 产品型号、跟踪引脚功能和连接器类型的对应表

产品名	产品型号	连接器类型	外部跟踪输出功能
E1	R0E000010KCE00	14 引脚型	不能使用
E20	R0E000200KCT00	14 引脚型 (需要 38 引脚 /14 引脚转换适配器)	不能使用
E20	R0E000200KCT00	38 引脚型	能使用

仿真器连接器的类型有以下所示的 14 引脚型和 38 引脚型，请根据使用目的选择使用。当使用能进行大容量实时跟踪的外部跟踪输出功能时，需要 38 引脚型的连接器。

(1) 14 引脚型（不能使用外部跟踪输出功能）

只支持 JTAG 等通信基本功能而不对应外部跟踪输出功能，因此不能使用通过外部跟踪输出功能进行的大容量跟踪。此连接器是 2.54mm 节距的普通连接器。

(2) 38 引脚型（能使用外部跟踪输出功能）

这是对应 JTAG 等通信基本性能和外部跟踪输出功能的连接器，能进行大容量的实时跟踪，是大小与 14 引脚型同等的小型连接器。

2.2 安装在用户系统上的仿真器连接器

E1/E20 仿真器推荐的仿真器连接器如表 2.2 所示。

表 2.2 推荐的连接器

	型号	厂家	规格
14 引脚连接器	7614-6002	Sumitomo 3M Limited	14 引脚型直立型 (日本国内推荐)
	2514-6002	3M Limited	14 引脚直立型 (日本以外的国家推荐)
38 引脚连接器	2-5767004-2	Tyco Electronics Japan G.K.	38 引脚型

【注意事项】

- (1) 在使用 14 引脚连接器时，请不要在周围 5mm 以内安装高于 10mm 的部件。E1 的 14 引脚连接器和用户接口电缆的连接例子如图 2.1 所示，E20 的使用 38 引脚 /14 引脚转换适配器的 14 引脚连接器和用户接口电缆的连接例子如图 2.2 所示。

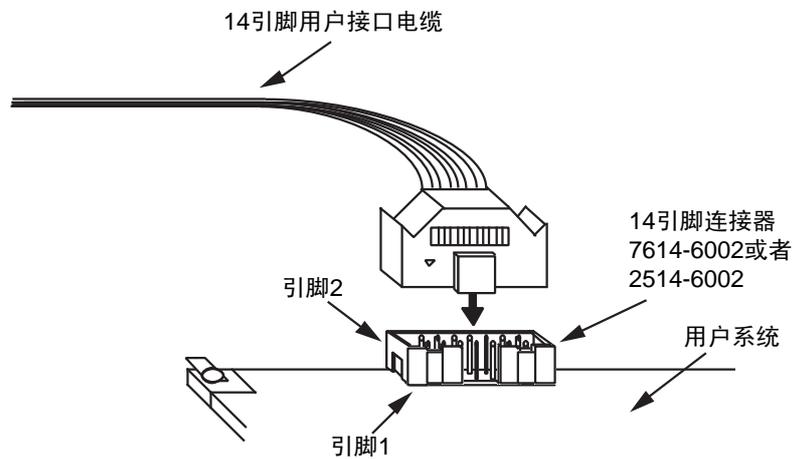


图 2.1 E1 仿真器的 14 引脚连接器和用户接口电缆的连接方法

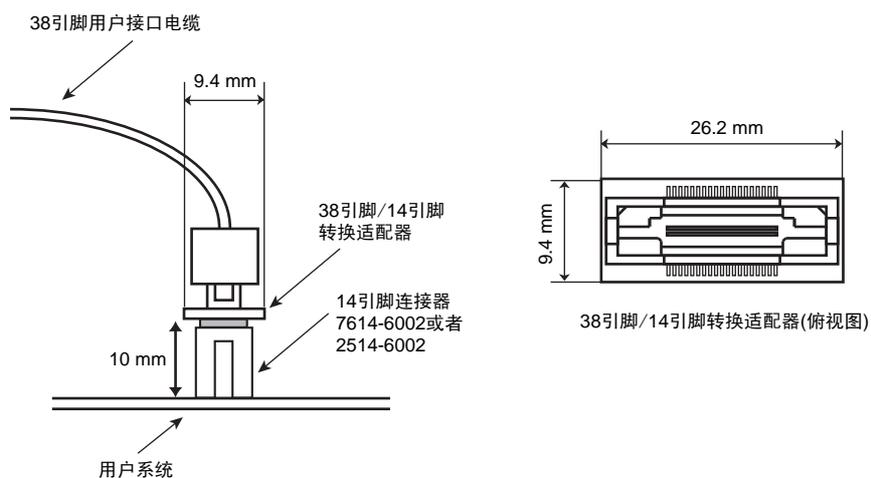


图 2.2 E20 仿真器的 14 引脚连接器和用户接口电缆的连接方法

- (2) 在使用 38 引脚连接器时，为了减少外来干扰噪声，请不要将其他信号线布在安装连接器的地方。如图 2.3 所示，在用户系统上的连接器周围（限制部件安装的区域）安装其他部件时，请不要安装高于 5mm 的部件。

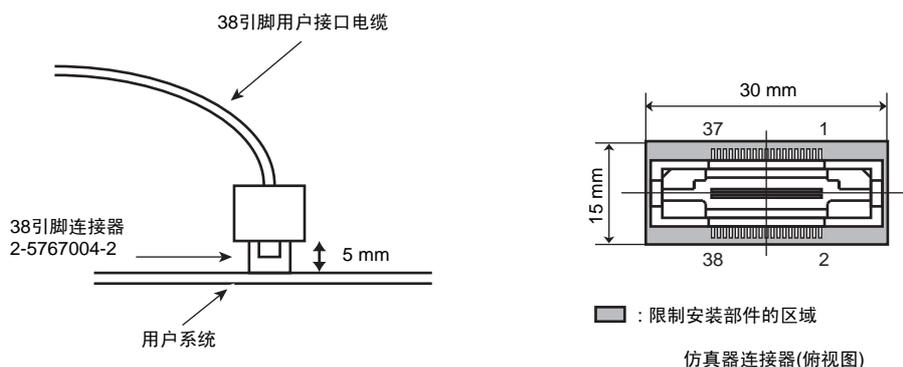


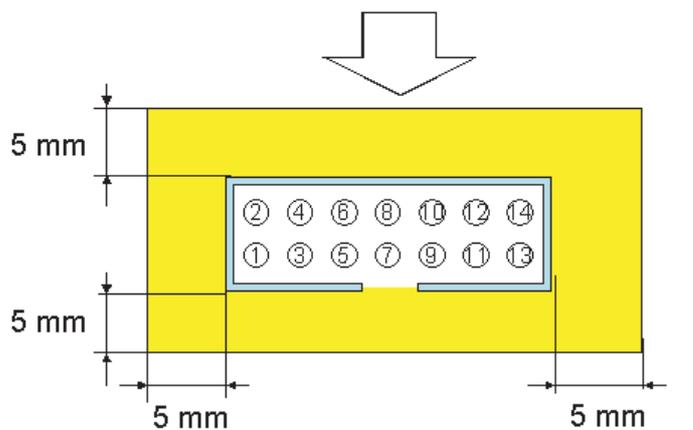
图 2.3 E20 仿真器的 38 引脚连接器和用户接口电缆的连接方法

注意

请注意：在仿真器连接器周围安装其他部件时有高度限制。

连接器型号：7614-6002(Sumitomo 3M Limited产)或者
2514-6002(3M Limited产)

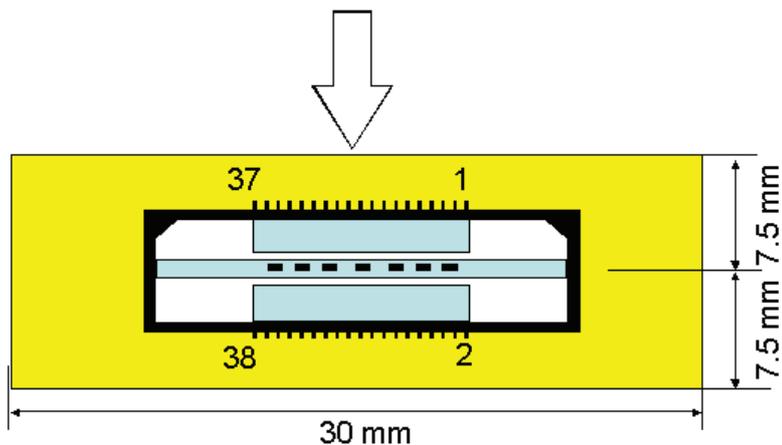
从此侧面连接仿真器



 限制安装部件的区域(高度限制为不超过10mm)

连接器型号：2-5767004-2(Tyco Electronics Japan G.K.产)

从此侧面连接仿真器



 限制安装部件的区域(高度限制为不超过5mm)

2.3 仿真器连接器的引脚排列

14 引脚连接器和 38 引脚连接器的引脚排列分别如图 2.4 和图 2.5 所示。

请注意：图 2.4 中记载的 14 引脚连接器的引脚序号排列与 E10A-USB 的引脚序号排列不同。

另外，在用 FDT 对 RX610 群单片机进行编程时，14 引脚连接器的引脚排列不同。

接线方法请参照 <http://www.renesas.com/fdt>。

(1) 14 引脚连接器的引脚排列

引脚序号 【注1】	信号名 【注2】 【注3】	输入/输出 【注4】	备注
1	TCK	输入	
2	GND 【注5】	—	
3	TRST# 【注6】	输入	
4	(EMLE) 【注7】	输入/输出	
5	TDO	输出	
6	— 【注8】	—	未连接
7	(MD1) 【注7】	输入/输出	
8	VCC	—	电源
9	TMS	输入	
10	(MD0) 【注7】	输入/输出	
11	TDI	输入	
12	GND 【注5】	—	
13	RES# 【注6】	输入/输出	用户系统的复位
14	GND 【注5】	—	用于检测电缆

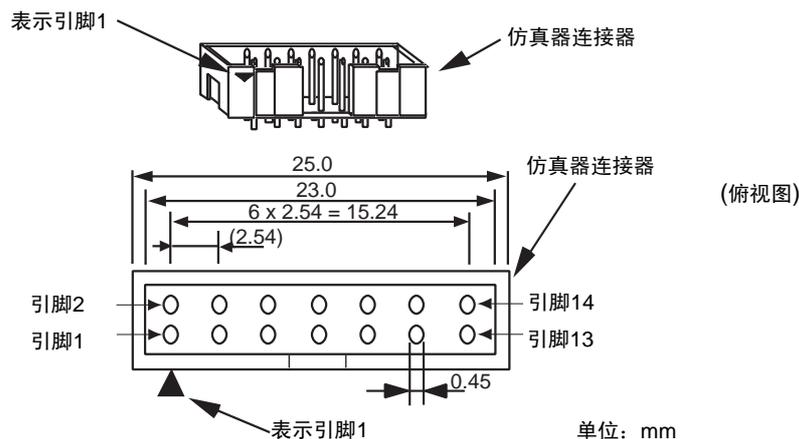


图 2.4 仿真器连接器的引脚排列（14 引脚型）

- 【注】
1. 和 E10A-USB 的引脚序号排列不同。
 2. 请注意：在用 FDT 对 RX610 群单片机进行编程时 14 引脚连接器的引脚排列不同。
接线方法请参照 <http://www.renesas.com/fdt>。
 3. 用连接 E1/E20 时（调试时）的各群单片机的引脚名称表示信号名。
 4. 这是从用户系统看 E1/E20 时的输入 / 输出方向。
 5. 必须在 PCB 上将引脚 2、引脚 12 和引脚 14 牢固地连接 GND。除了用作电气 GND 以外，还用于 E1/E20 仿真器监视连接器的连接。
 6. 信号名 # 表示 Low 电平有效信号。
 7. 即使不将（信号名）所示的引脚连接 E1/E20 也能正常工作。当不连接 E1/E20 时，（信号名）所示的引脚必须在用户系统上构成电路。
但是，如果不连接 EMLE、MD1、MD0 信号，就可能在将来扩展仿真器功能时受限制。
 8. 不能连接没有记载信号名的引脚。

(2) 38 引脚连接器的引脚排列

引脚序号	信号名【注1】	输入/输出【注2】	备注	引脚序号	信号名【注1】	输入/输出【注2】	备注
1	—【注3】	—	未连接	2	(MD0)【注4】	输入/输出	
3	(EMLE)【注4】	输入/输出		4	—【注3】		未连接
5	GND【注5】	—	用于检测电缆	6	TRCLK	输出	
7	—【注3】	—	未连接	8	(MD1)【注4】	输入/输出	
9	RES#【注6】	输入/输出	用户系统的复位	10	—【注3】	—	未连接
11	TDO	输出		12	VCC_TR	—	电源
13	—【注3】	—	未连接	14	VCC	—	电源
15	TCK	输入		16	—【注3】	—	未连接
17	TMS	输入		18	—【注3】	—	未连接
19	TDI	输入		20	—【注3】	—	未连接
21	TRST#【注6】	输入		22	—【注3】	—	未连接
23	—【注3】	—	未连接	24	TRDATA3	输出	
25	—【注3】	—	未连接	26	TRDATA2	输出	
27	—【注3】	—	未连接	28	TRDATA1	输出	
29	—【注3】	—	未连接	30	TRDATA0	输出	
31	—【注3】	—	未连接	32	TRSYNC#【注6】	输出	
33	—【注3】	—	未连接	34	—【注3】	—	未连接
35	—【注3】	—	未连接	36	—【注3】	—	未连接
37	—【注3】	—	未连接	38	—【注3】	—	未连接

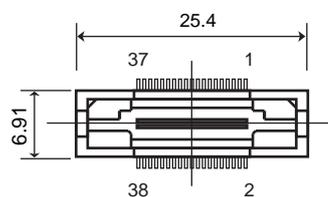


图 2.5 仿真器连接器的引脚排列 (38 引脚型)

必须将 38 引脚连接器中间的 GND 总线引线连接 GND。

- 【注】
1. 用连接 E1/E20 时 (调试时) 的各群单片机的引脚名称表示信号名。
 2. 这是从用户系统看 E1/E20 时的输入 / 输出方向。
 3. 不能连接没有记载信号名的引脚。
 4. 即使不将 (信号名) 所示的引脚连接 E1/E20 也能正常工作。当不连接 E1/E20 时, (信号名) 所示的引脚必须在用户系统上构成电路。
但是, 如果不连接 EMLE、MD1、MD0 信号, 就可能在将来扩展仿真器功能时受限制。
 5. 必须在 PCB 上将引脚 5 和连接器的 GND 总线引线牢固地连接 GND。除了用作电气 GND 以外, 还用于 E1/E20 仿真器监视连接器的连接。
 6. 表示 Low 电平有效信号。

2.4 仿真器连接器和单片机之间的推荐连接例子

2.4.1 推荐连接例子（14 引脚连接器）

使用 E1/E20 仿真器时的仿真器连接器（14 引脚型）和单片机之间的推荐连接例子如图 2.6 所示。有关图 2.6 中没有表示的各信号线的处理，请参照“2.4.3 连接时的注意事项”。

- 【注】
1. 因为在使用排电阻进行上拉时有可能受其他引脚噪声的影响，所以 TCK 引脚的电阻必须与其他引脚的电阻分开。
 2. 仿真器连接器和单片机之间的接线尽可能用短的布线（推荐在 50mm 以内）。另外，在电路板上不要将信号线的布线绕到仿真器连接器和单片机之间以外的地方。
 3. JTAG 引脚（TCK、TMS、TDI）必须尽可能用相同长度的布线（推荐在 $\pm 20\text{mm}$ 以内）。
 4. 必须用 GND 将仿真器连接器和单片机之间的 TCK 引脚布线围起来。
 5. TDO 引脚必须用最短的布线。
 6. 有关不使用 E1/E20 仿真器时的引脚处理，请参照相关的单片机硬件手册。

请注意：14 引脚连接器的引脚序号排列与 E10A-USB 的引脚序号排列不同（参照图 2.4）。

另外，在用 FDT 对 RX610 群单片机进行编程时，14 引脚连接器的引脚排列不同。

接线方法请参照 <http://www.renesas.com/fdt>。

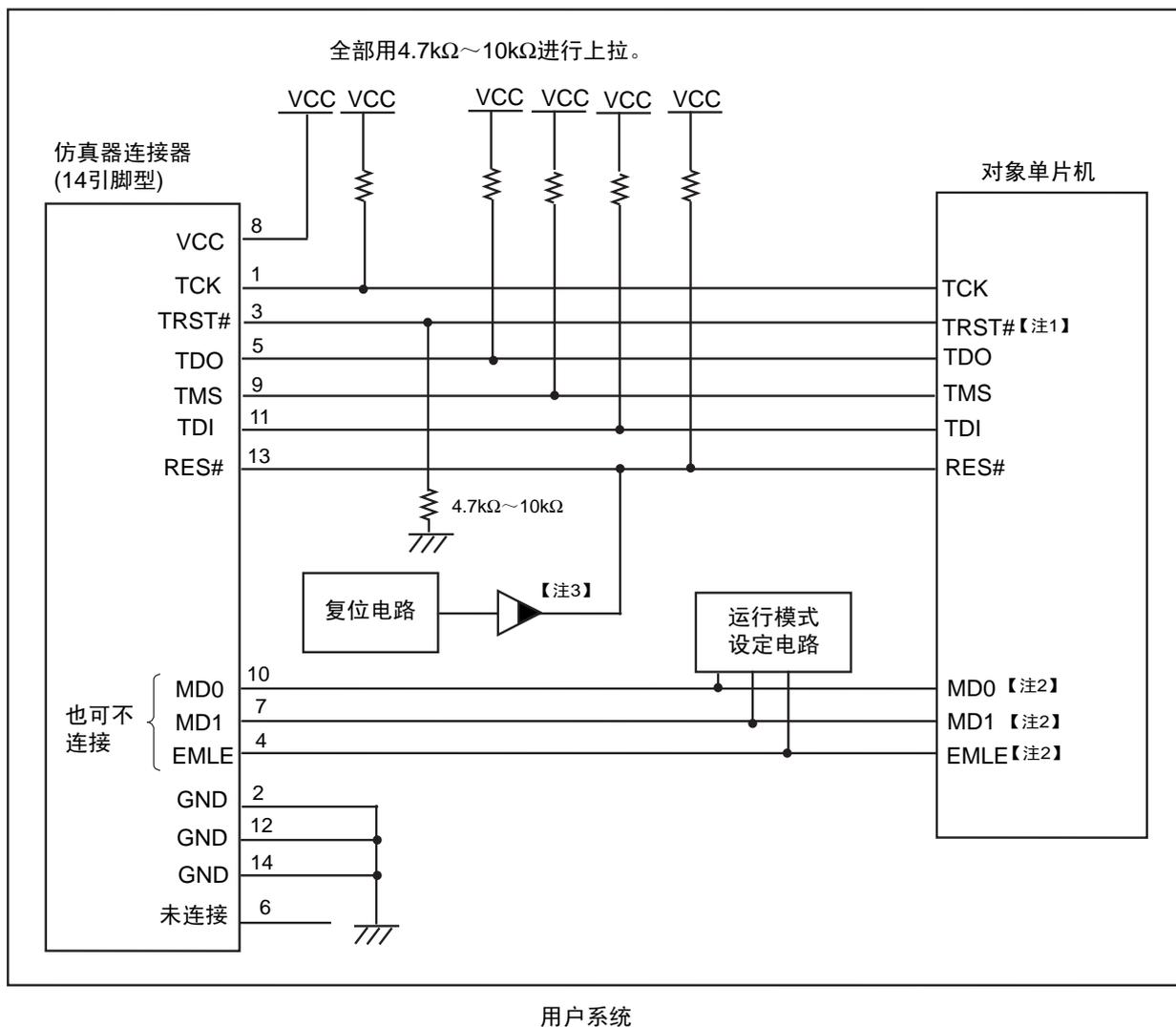


图 2.6 使用 E1/E20 时的仿真器连接器和单片机之间的推荐连接例子（14 引脚型）

- 【注】 1. 对于 TRST# 引脚的处理，即使用 4.7kΩ ~ 10kΩ 的电阻进行上拉也没有问题，但是请注意：如果进行了上拉处理，就不能使用热插拔。
2. 有关 MD0、MD1、EMLE 引脚的处理，请参照“2.4.3 连接时的注意事项”。
3. 必须将用户系统的复位电路置为漏极开路输出。

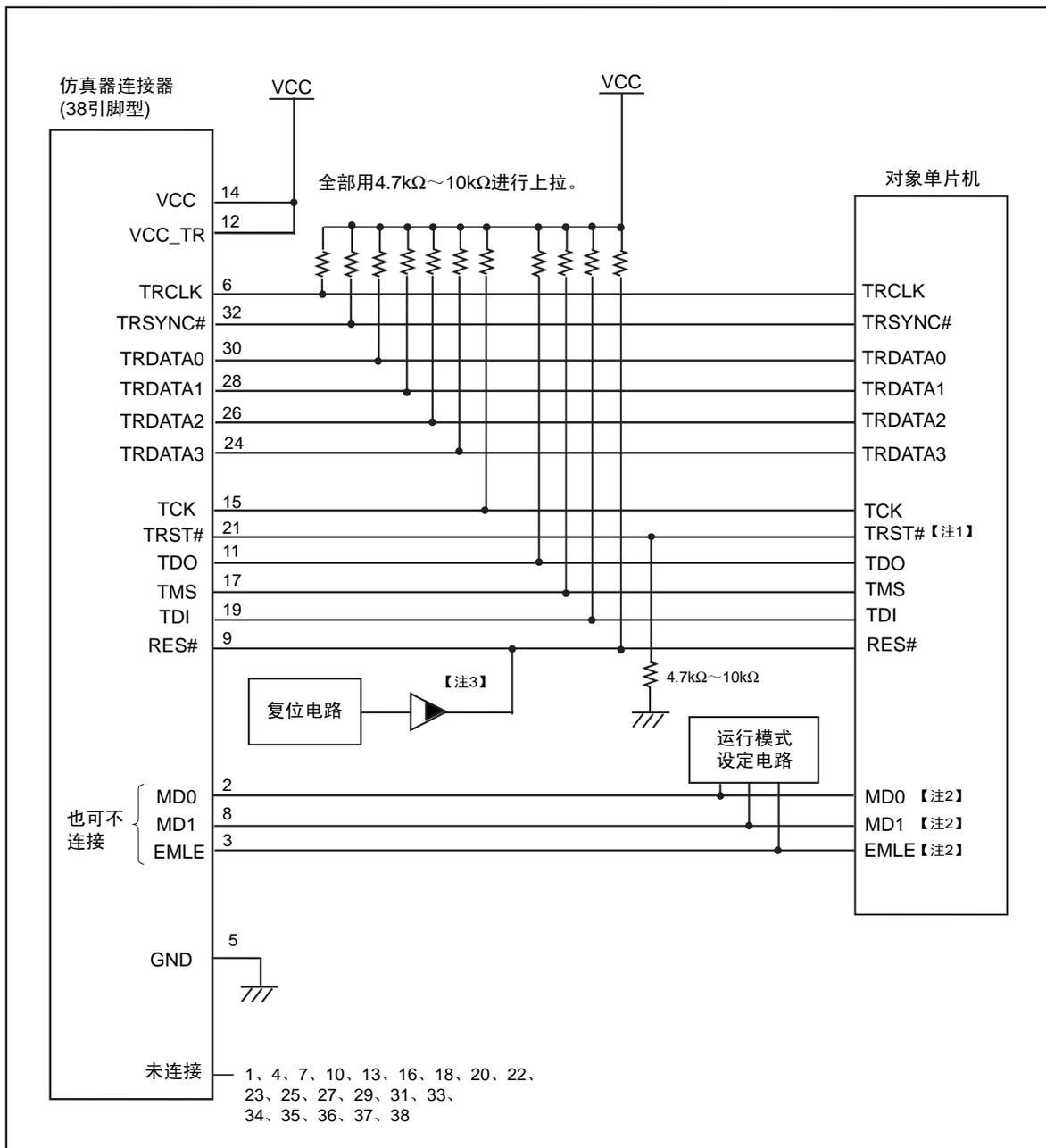
在使用热插拔的情况下，需要在接通用户系统的电源时（系统复位时）TRST# 引脚为“L”电平并且 EMLE 引脚为“H”电平的状态下连接仿真器，必须对 TRST# 引脚进行下拉处理并且为 EMLE 引脚设计能进行“H”电平和“L”电平切换的电路。

2.4.2 推荐连接例子（38 引脚连接器）

使用 E1/E20 仿真器时的仿真器连接器（38 引脚型）和单片机之间的推荐连接例子如图 2.7 所示。有关图 2.7 中没有表示的各信号线的处理，请参照“2.4.3 连接时的注意事项”。

必须将 38 引脚连接器中间的 GND 总线引线连接 GND。

- 【注】
1. 因为在使用排电阻进行上拉时有可能受其他引脚噪声的影响，所以 TCK 引脚的电阻必须与其他引脚的电阻分开。
 2. 仿真器连接器和单片机之间的接线尽可能用短的布线（推荐在 50mm 以内）。另外，在电路板上不要将信号线的布线绕到仿真器连接器和单片机之间以外的地方。
 3. JTAG 引脚（TCK、TMS、TDI）必须尽可能用同长度的布线（推荐在 ± 20mm 以内）。
 4. 必须用 GND 将仿真器连接器和单片机之间的 TCK 引脚布线围起来。
 5. TDO 引脚必须用最短的布线。
 6. 因为跟踪信号（TRCLK、TRSYNC#、TRDATA0 ~ 3）高速变化，所以必须尽可能用相同长度的布线（推荐在 ±10mm 以内），避免分叉布线并且不接近其他信号线。另外，不能在线上配置会引起噪声源的元件。
 7. 必须用 GND 将仿真器连接器和单片机之间的 TRCLK 引脚布线围起来。
 8. 有关不使用 E1/E20 仿真器时的引脚处理，请参照相关的单片机硬件手册。



用户系统

图 2.7 使用 E1/E20 时的仿真器连接器和单片机之间的推荐连接例子（38 引脚型）

- 【注】
1. 对于 TRST# 引脚的处理，即使用 $4.7\text{k}\Omega \sim 10\text{k}\Omega$ 的电阻进行上拉也没有问题，但是请注意：如果进行了上拉处理，就不能使用热插拔。
 2. 有关 MD0、MD1、EMLE 引脚的处理，请参照“2.4.3 连接时的注意事项”。
 3. 必须将用户系统的复位电路置为漏极开路输出。

在使用热插拔的情况下，需要在接通用户系统的电源时（系统复位时）TRST# 引脚为“L”电平并且 EMLE 引脚为“H”电平的条件下连接仿真器，必须对 TRST# 引脚进行下拉处理并且为 EMLE 引脚设计能进行“H”电平和“L”电平切换的电路。

2.4.3 连接时的注意事项

1. 因为E1/E20占用TCK、TDO、TMS、TDI引脚，所以这些引脚不能使用多路复用的引脚功能。4.7kΩ~10kΩ的上拉电阻在不连接E1/E20时起保护作用。

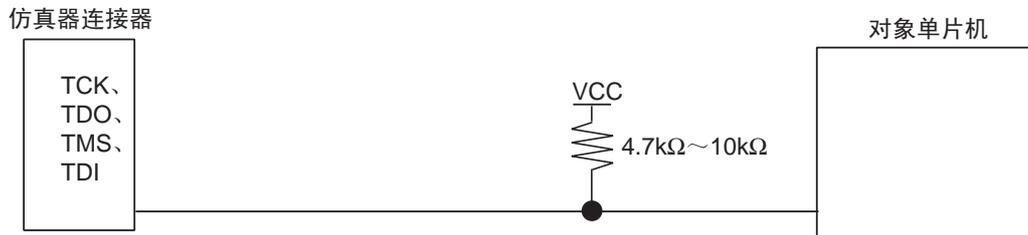


图 2.8 TCK、TDO、TMS、TDI 引脚和 E1/E20 的连接

2. 因为E1/E20占用TRST#引脚，所以此引脚不能使用多路复用的引脚功能。对于TRST#引脚的处理，即使通过4.7kΩ~10kΩ的电阻进行上拉也没有问题，但是请注意：如果进行了上拉处理，就不能使用热插拔。

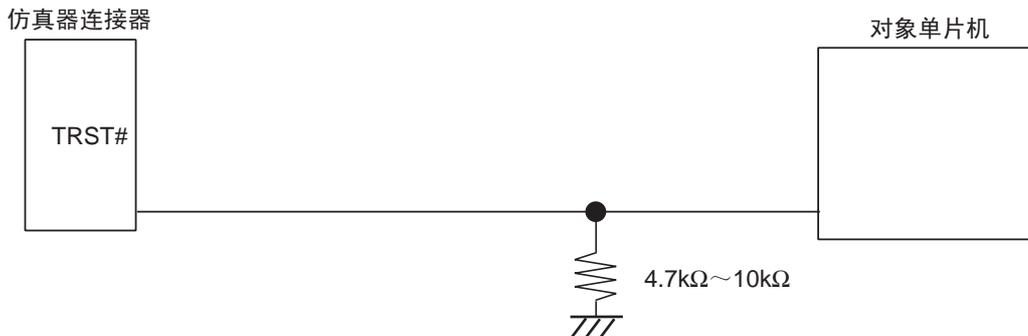


图 2.9 TRST# 引脚和 E1/E20 的连接

3. E1/E20使用RES#引脚。如下图所示，在用户系统中有复位电路（以下称作用户逻辑）的情况下，必须用漏极开路缓冲器将仿真器连接器的RES#信号和用户逻辑的输出信号连接。在没有用户逻辑的情况下，必须将仿真器连接器的RES#信号直接连接到各群单片机的RES#引脚。另外，在使用热插拔的情况下，为了抑止在连接仿真器时对RES#引脚产生的噪声，必须在RES#引脚和GND之间安装陶瓷电容器（0.1μF左右）。

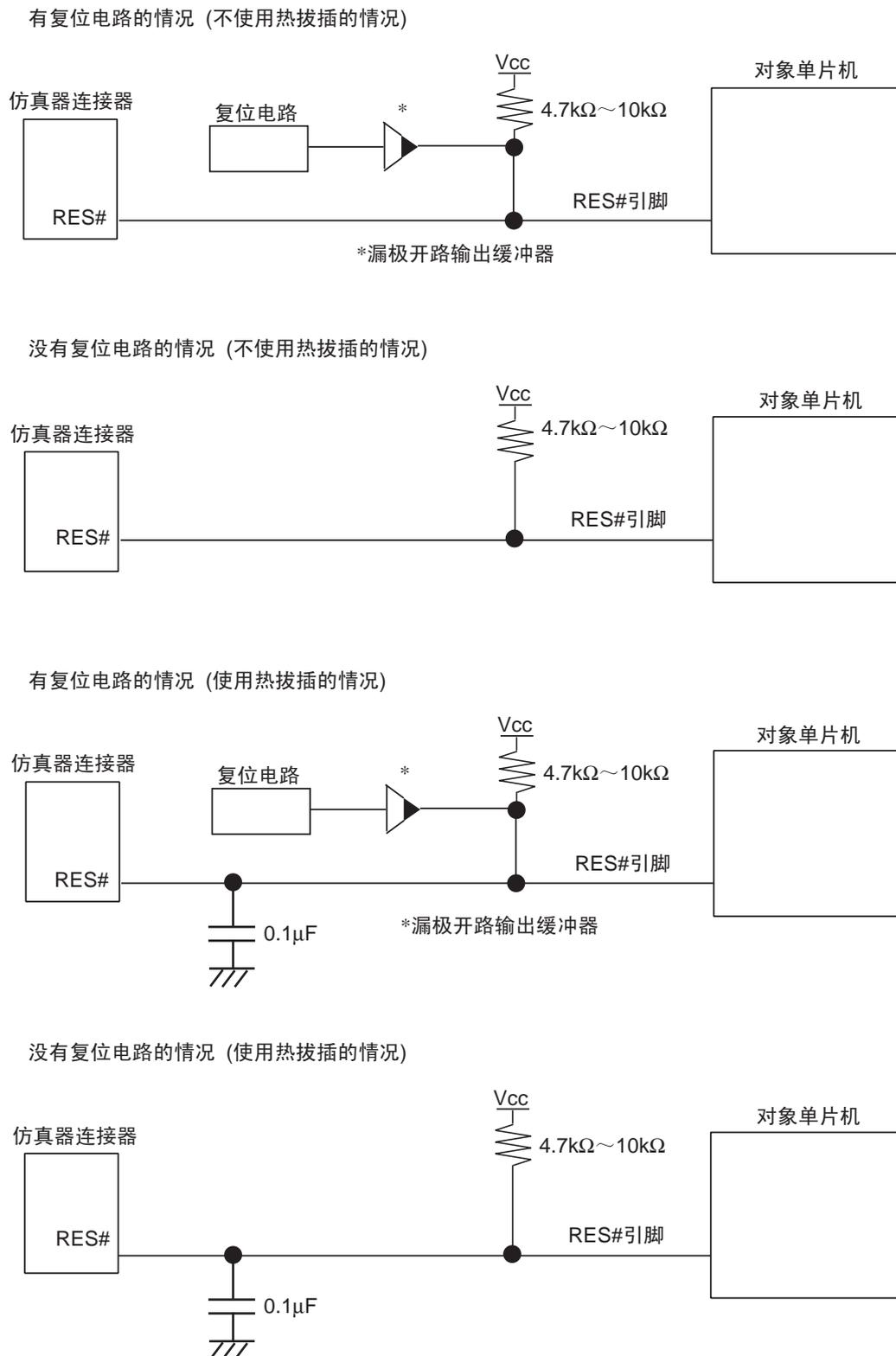


图 2.10 复位电路例子

4. MD0引脚和MD1引脚的连接有以下2种方法:

【将MD0引脚和MD1引脚连接E1/E20的情况】

必须用 $4.7\text{k}\Omega \sim 10\text{k}\Omega$ 的电阻将MD0引脚和MD1引脚上拉。

在用FDT对内部闪存进行编程时，必须通过FDT进行设定，使MD1能输出“L”电平。

【不将MD0引脚和MD1引脚连接E1/E20的情况】

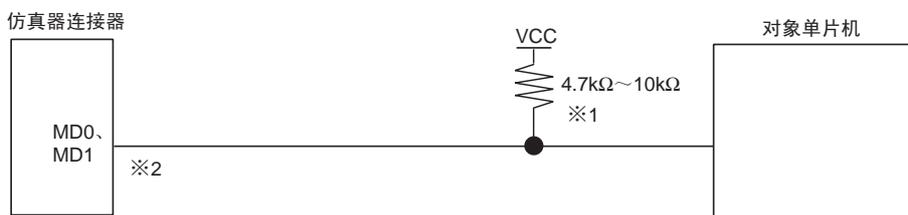
在调试时，必须用 $4.7\text{k}\Omega \sim 10\text{k}\Omega$ 的电阻将MD0引脚和MD1引脚上拉。

在用FDT对内部闪存进行编程时，必须将MD0上拉，并且用开关等将MD1下拉。

但是，如果MD0引脚和MD1引脚未被连接，就可能在将来扩展仿真器功能时受限制。

请注意：在用FDT对RX610群单片机进行编程时14引脚连接器的引脚排列不同。接线方法请参照<http://www.renesas.com/fdt>。

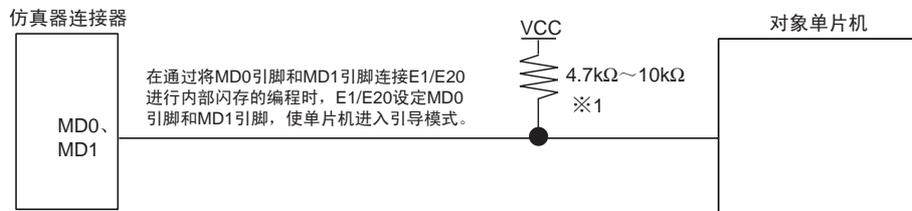
不考虑内部闪存的编程的情况



- ※1. 必须根据所用单片机的运行模式进行上拉或者下拉的处理。
- ※2. 在不通过E1/E20检查运行模式时，也可以不连接。

考虑内部闪存的编程的情况

(a) 连接E1/E20



- ※1. 必须根据所用单片机的运行模式进行上拉或者下拉的处理。

考虑内部闪存的编程的情况

(b) 不连接E1/E20

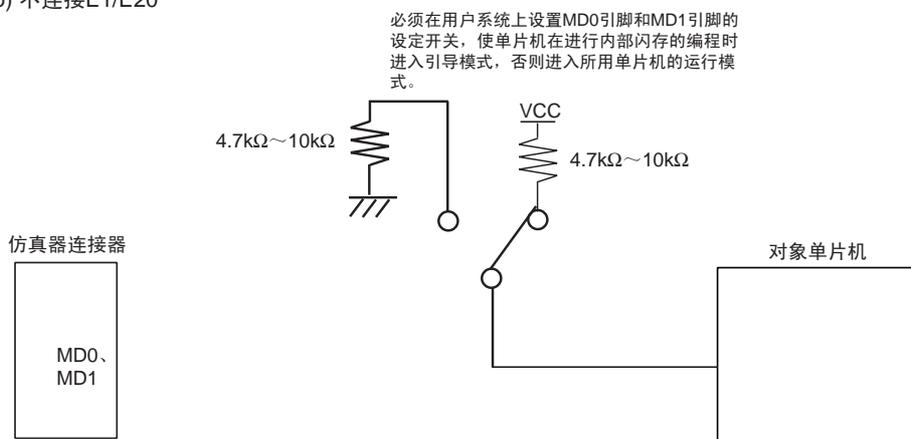


图 2.11 MD0 引脚、MD1 引脚和 E1/E20 的连接

5. 这是连接到EMLE引脚的仿真器连接器的选项。

当连接E1/E20后使用时，需要将EMLE引脚置为“H”电平；当单片机单独使用时，需要将EMLE引脚置为“L”电平。

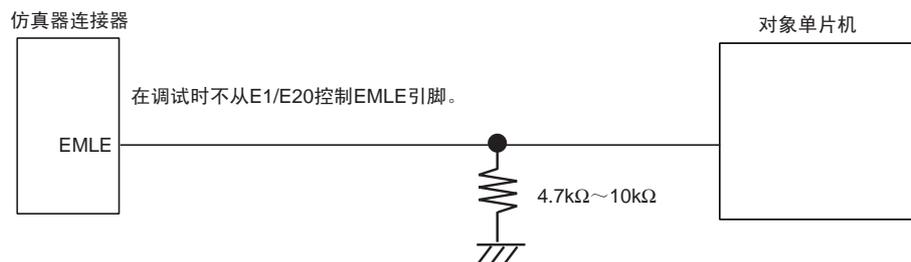
在连接E1/E20的EMLE引脚和单片机的EMLE引脚时，因为E1/E20控制EMLE引脚，所以必须在用户系统上对EMLE引脚进行下拉处理。

在不连接E1/E20的EMLE引脚和单片机的EMLE引脚时，必须在用户系统上为EMLE引脚设计能进行“H”电平和“L”电平切换的电路。

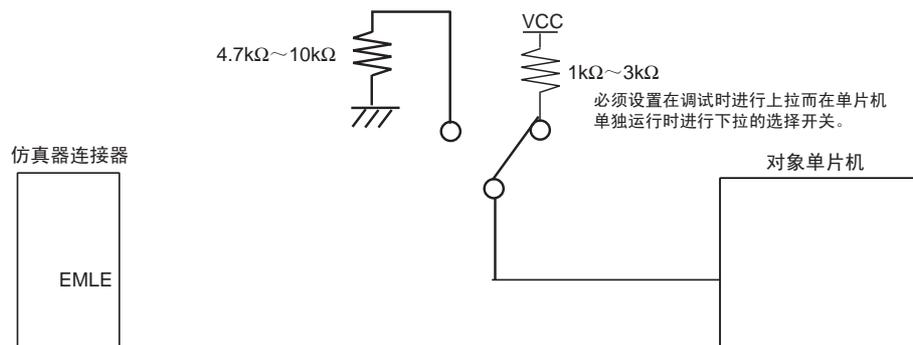
但是，如果EMLE引脚未被连接，就可能在将来扩展仿真器功能时受限制。

在使用热插拔的情况下，必须在用户系统上设计能使EMLE引脚进行“H”电平和“L”电平切换的电路。

E1/E20和EMLE引脚连接的情况（不使用热拔插）



E1/E20和EMLE引脚不连接的情况（不使用热拔插）



使用热拔插的情况【注】

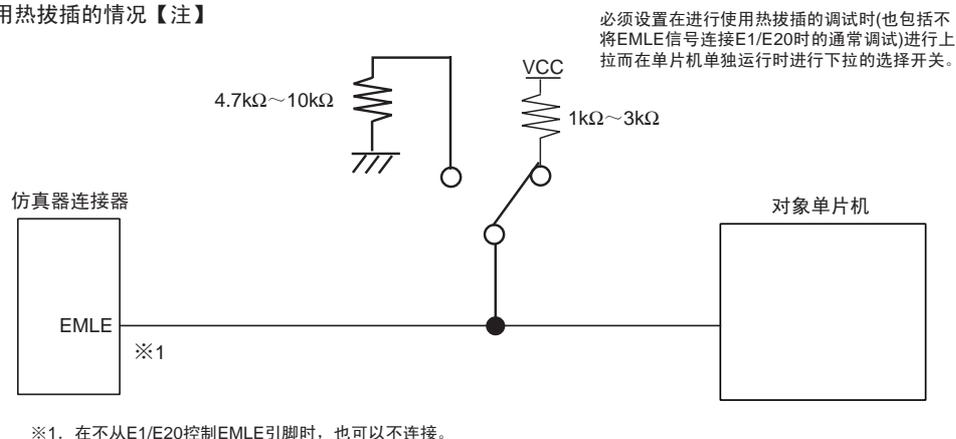


图 2.12 E1/E20 和 EMLE 引脚的连接

【注】 EMLE 引脚是否连接 E1/E20 都没有问题。但是，如果不连接 EMLE 引脚，就可能在将来扩展仿真器功能时受限制。

另外，在不使用热插拔并且不连接 E1/E20 而单片机单独运行时，必须将 EMLE 引脚置为“L”电平（下拉）。

6. 为了进行E20的大容量跟踪，需要将TRCLK、TRSYNC#、TRDATA0~TRDATA3引脚连接38引脚连接器。因为这些跟踪引脚的信号高速变化，所以尽可能用相同长度的布线，避免分叉布线并且不接近其他信号线。

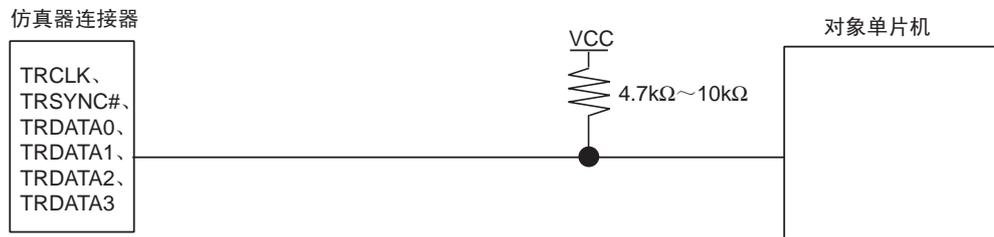


图 2.13 E1/E20 和跟踪引脚的连接

7. 仿真器连接器的GND必须和各群单片机的V_{SS}引脚的GND相同。
8. 必须将用户系统的VCC（电源）连接到仿真器连接器的VCC，可输入的电压是在单片机的保证范围内。另外，E1能给没有电源的简易评估系统提供电源，能提供的电源电流最大为200mA。

在使用本产品进行批量生产工程等需求可靠性的编程时，请不要使用本产品的供电功能，而将另外准备的符合单片机编程电压的电源提供给用户系统。因为本产品提供的电压取决于主机的USB电源性能，所以不能保证电压的精度。



在提供电源时，必须确认用户系统的电源电路没有短路。必须在确认仿真器连接器的引脚排列没有连接E1/E20。如果发生错误连接，主机、E1/E20和用户系统就可能冒烟起火。

9. E1 和 E20 的内部接口电路分别如图 2.14 和图 2.15 所示，E20 的内部接口电路的 TRACE 电路部分如图 2.16 所示。

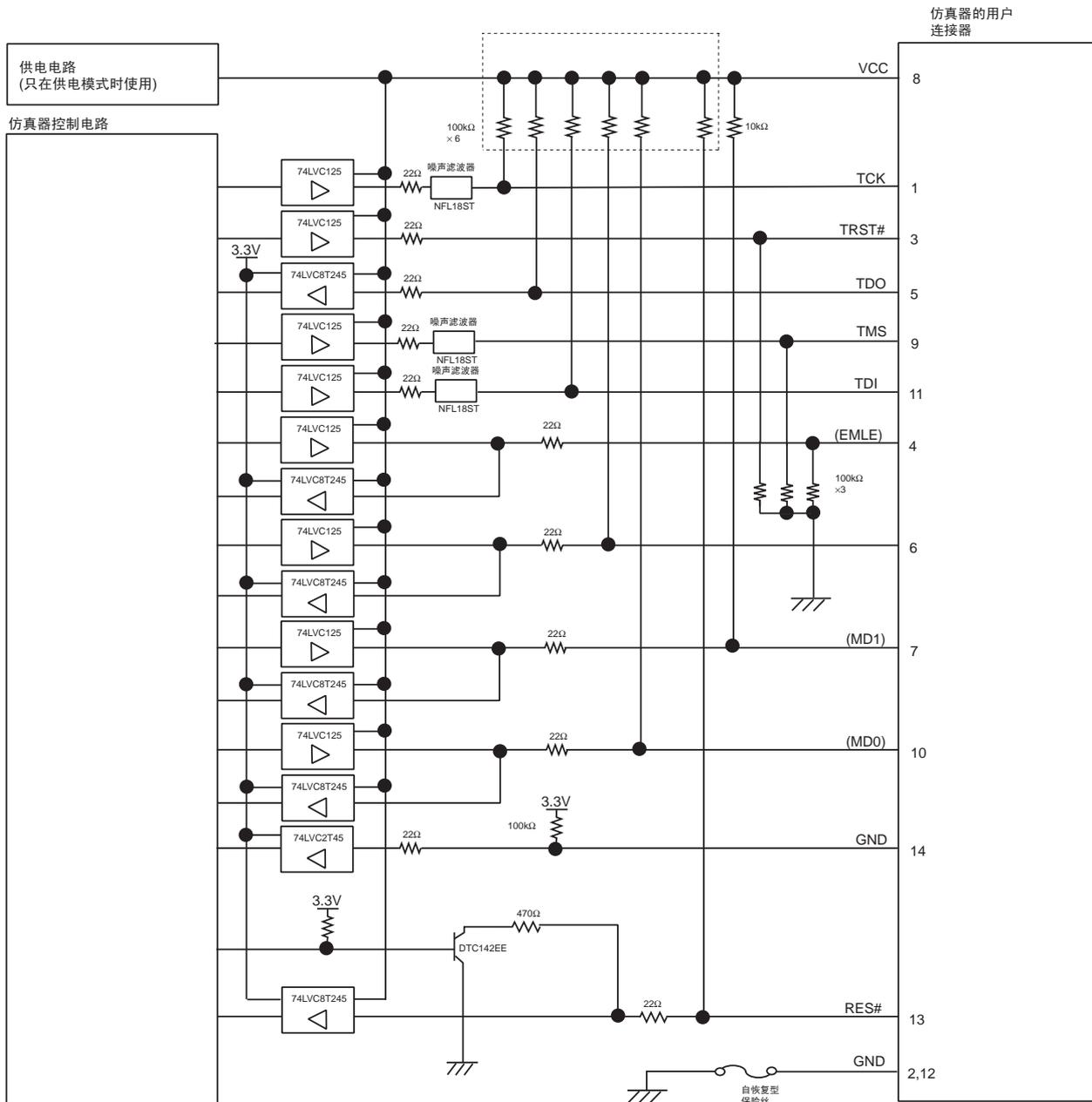


图 2.14 E1 的内部接口电路

【注】 () 内所示的信号即使不连接用户系统也能正常工作。

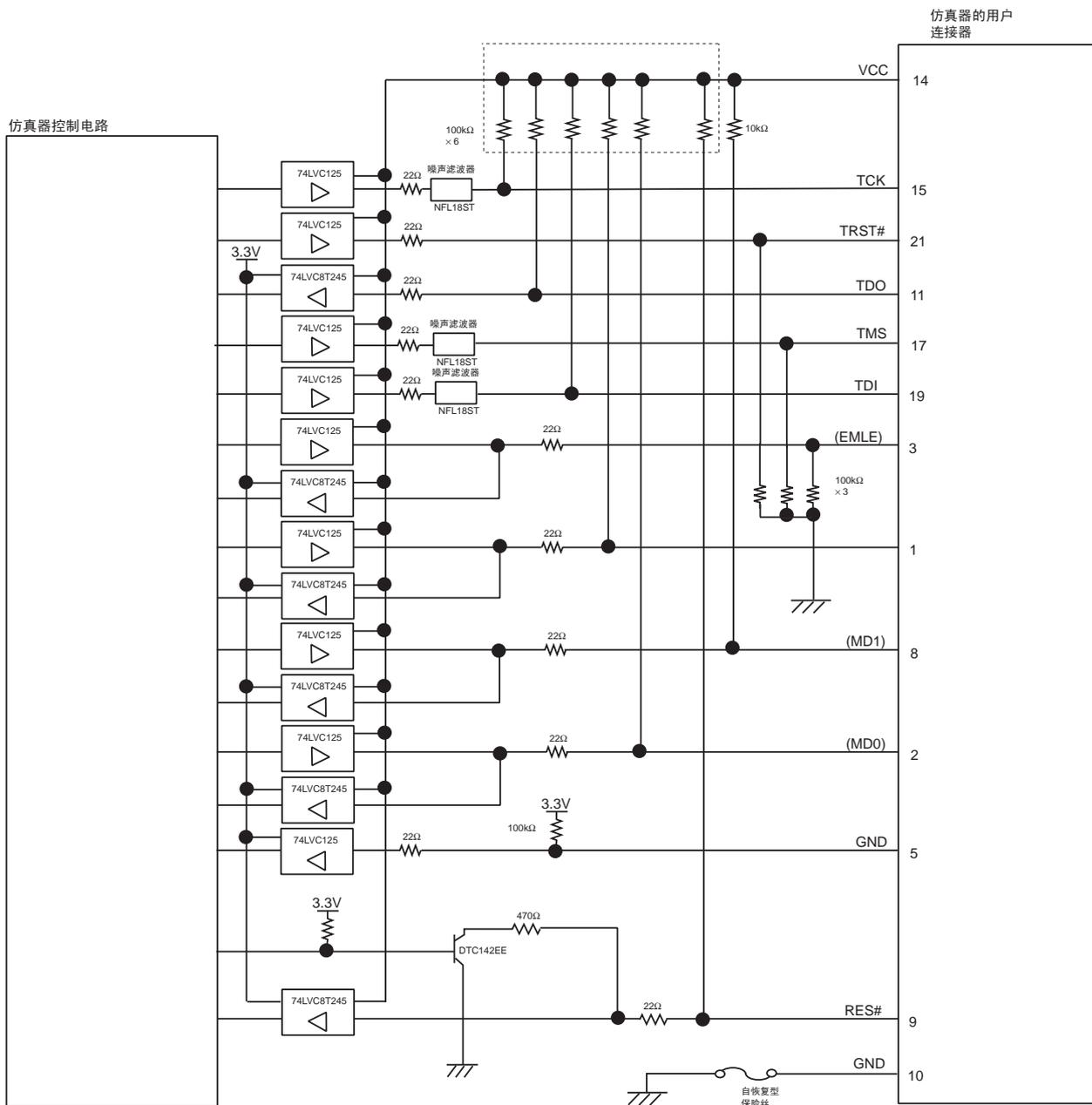


图 2.15 E20 的内部接口电路

【注】 () 内所示的信号即使不连接用户系统也能正常工作。

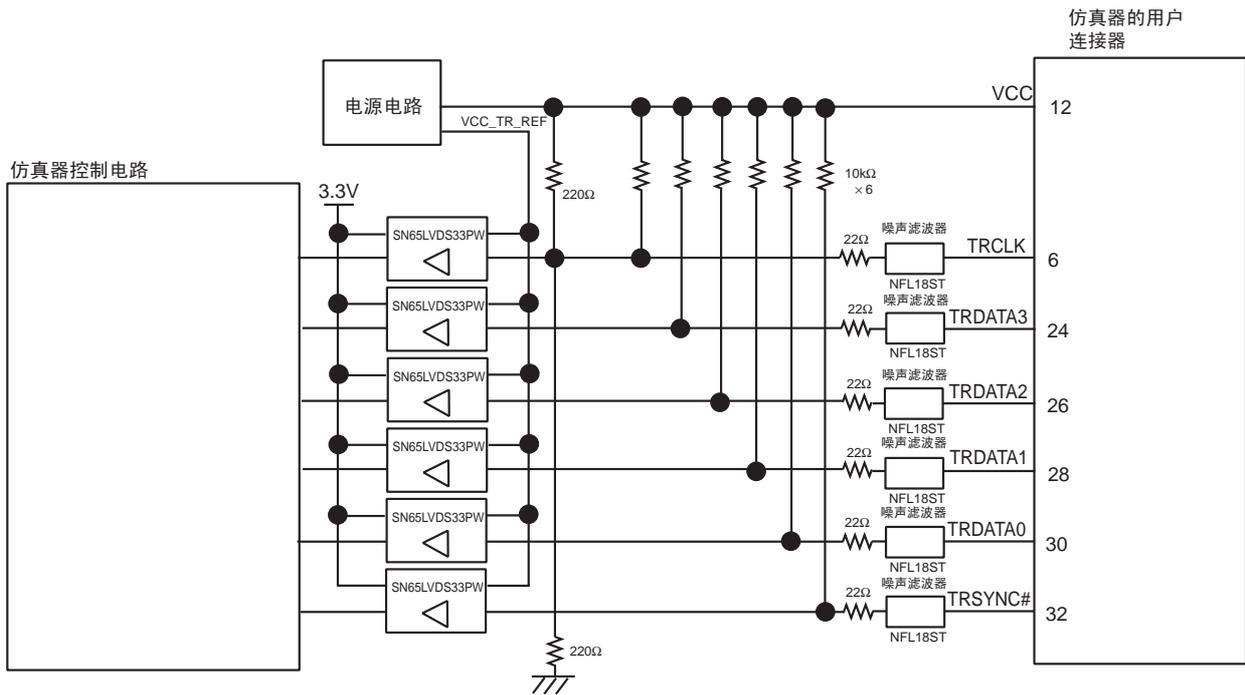


图 2.16 E20 的内部接口电路 (2) TRACE 电路部分

10. E1/E20 仿真器不对应 JTAG 链式连接。在设计用户系统时，不能将器件的 TDI 信号和 TDO 信号连接边界扫描环路，并且需要用开关等分离器件（参照图 2.17）。

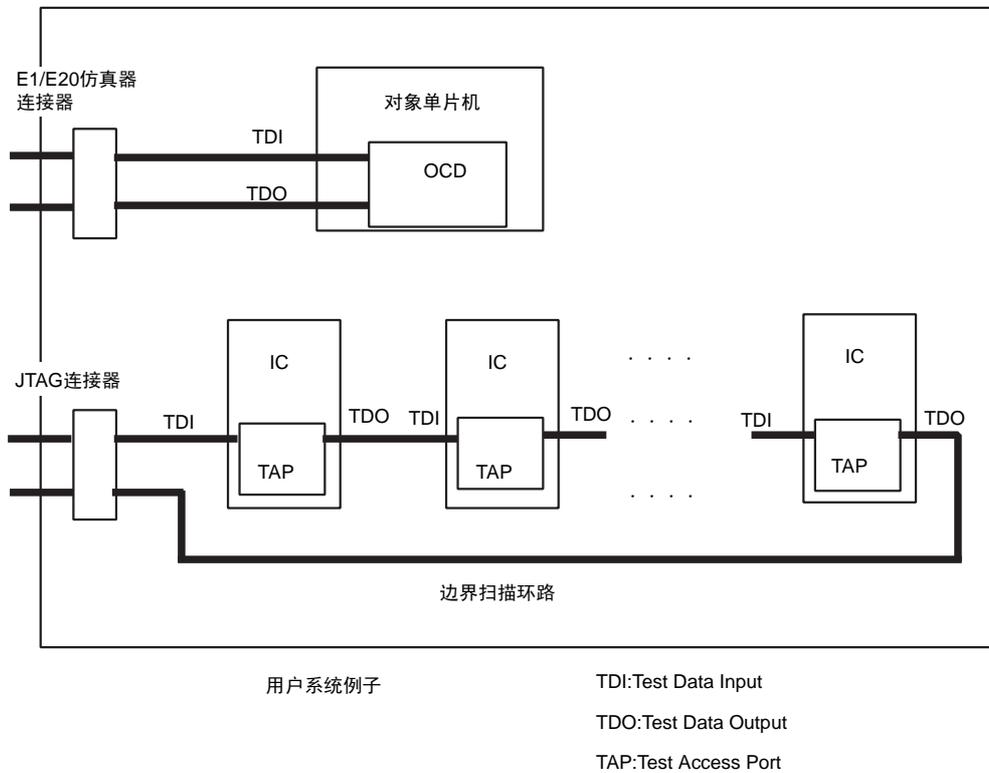


图 2.17 设计用户系统时的注意事项

11. 在使用另售的E1仿真器的小型连接器转换适配器（R0E000010CKZ11）时，引脚排列和E1仿真器的标准接口连接器的引脚排列不同。使用E1仿真器的小型连接器转换适配器时的14引脚连接器的引脚排列如表2.3所示。

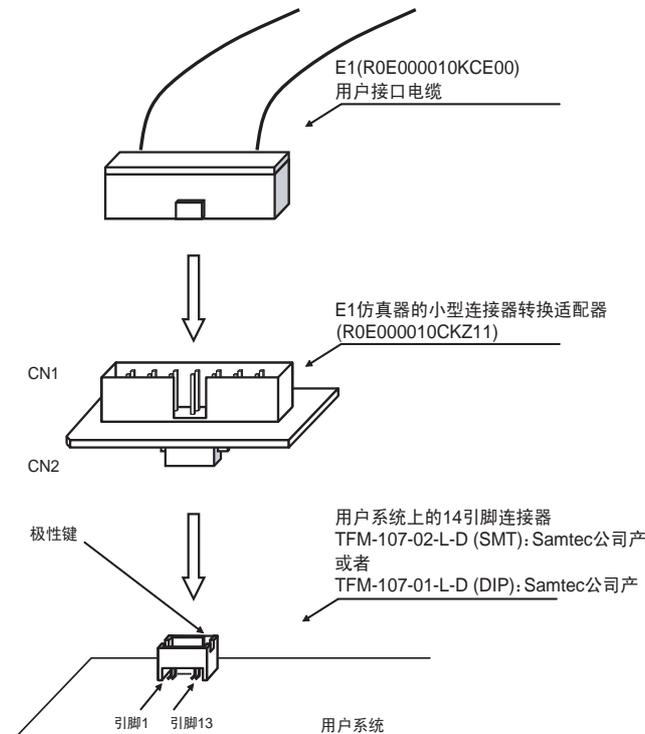


图 2.18 E1 仿真器的小型连接器转换适配器的使用方法

表 2.3 使用 E1 仿真器的小型连接器转换适配器时的连接器的引脚排列

引脚序号	信号名【注1】【注2】	输入/输出【注3】	备注
1	GND【注4】	—	
2	(MD1)【注6】	输入/输出	
3	VCC	—	电源
4	TCK	输入	
5	RES#【注5】	输入/输出	用户系统的复位
6	GND【注4】	—	用于检测电缆
7	(MD0)【注6】	输入/输出	
8	TRST#【注5】	输入	
9	TDI	输入	
10	TDO	输出	
11	(EMLE)【注6】	输入/输出	
12	—【注7】	—	未连接
13	TMS	输入	
14	GND【注4】	—	

- 【注】
1. 请注意：在用 FDT 对 RX610 群单片机进行编程时，14 引脚连接器的引脚排列不同。接线方法请参照 <http://www.renesas.com/fdt>。
 2. 用连接 E1/E20 时（调试时）的各群单片机的引脚名称表示信号名。
 3. 这是从用户系统看 E1/E20 时的输入/输出方向。
 4. 必须在 PCB 上将引脚 1、引脚 6 和引脚 14 连接 GND。除了用作电气 GND 以外，还用于 E1/E20 仿真器监视连接器的连接。
 5. 信号名 # 表示 Low 电平有效信号。
 6. 即使不将（信号名）所示的引脚连接 E1/E20 也能正常工作。当不连接 E1/E20 时，（信号名）所示的引脚必须在用户系统上构成电路。但是，如果不连接 EMLE、MD1、MD0 信号，就可能在将来扩展仿真器功能时受限制。
 7. 不能连接没有记载信号名的引脚。

3. 调试的准备

3.1 工作环境

在使用本仿真器时，请务必保证表 3.1 和表 3.2 所示的主机工作环境。

表 3.1 工作环境（Windows® XP、Windows® 2000）

PC 环境	
PC	IBM PC/AT 兼容机
OS	Windows® XP 32 位版 *1 *3 Windows® 2000 *1
CPU	推荐 Pentium4 1.6GHz 以上。
接口	USB2.0 *2
内存	推荐（1GB+ 装入模块文件大小的 10 倍）以上。
鼠标等点击设备	能连接主机的对应上述 OS 的鼠标等点击设备。
CD 驱动器	用于安装仿真调试程序或者参阅用户手册。
硬盘	在安装仿真调试程序时，需要 600MB 以上的可用空间（考虑到交换区，还需要准备 2 倍内存容量以上（推荐 4 倍以上）的可用空间）。
显示器的分辨率	推荐 1024×768 以上。

表 3.2 工作环境（Windows Vista®、Windows® 7）

PC 环境	
PC	IBM PC/AT 兼容机
OS	Windows Vista® 32 位版 *1 *4 Windows® 7 32/64 位版 *1
CPU	推荐 Pentium4 3GHz 或者 Core 2 Duo 1GHz 以上。
接口	USB2.0 *2
内存	推荐（2GB+ 装入模块文件大小的 10 倍）以上（32 位）。 推荐（3GB+ 装入模块文件大小的 10 倍）以上（64 位）。
鼠标等点击设备	能连接主机的对应上述 OS 的鼠标等点击设备。
CD 驱动器	用于安装仿真调试程序或者参阅用户手册。
硬盘	在安装仿真调试程序时，需要 600MB 以上的可用空间（考虑到交换区，还需要准备 2 倍内存容量以上（推荐 4 倍以上）的可用空间）。
显示器的分辨率	推荐 1024×768 以上。

【注】 *1 Windows 和 Windows Vista 是美国 Microsoft Corporation 在美国和其他国家的商标或者注册商标。其他所有的公司名称和产品名称是各公司的注册商标或者商标。

*2 对于 USB 接口，不保证所有主机、USB 设备、USB 集线器的组合运行。

*3 不对应 64 位版的 Windows® XP。

*4 不对应 64 位版的 Windows Vista®。

3.2 High-performance Embedded Workshop 的启动

按以下步骤启动 High-Performance Embedded Workshop:

1. 请将仿真器和用户系统连接到主机，接通仿真器和用户系统的电源。
2. 请从[Start]（开始）菜单的[Programs]（程序）中选择[Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop]。
显示[Welcome!]（欢迎!）对话框。

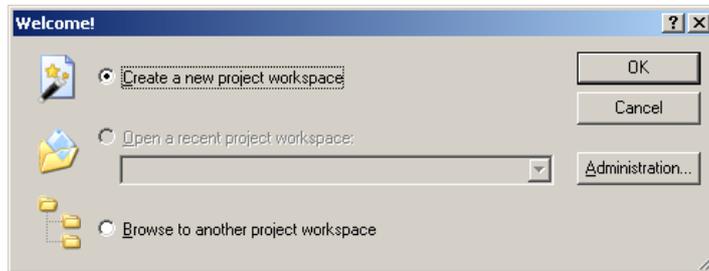


图 3.1 [Welcome!]（欢迎!）对话框

请选择以下的启动方法:

- [Create a new project workspace]（建立新的工程工作空间）
建立新的工作空间。
- [Open a recent project workspace]（打开最近使用过的工程工作空间）
显示打开过的工作空间的履历。
使用已用过的现有工作空间。
- [Browse to another project workspace]（浏览到另一个工作空间）
使用现有工作空间。

3.3 新工作空间的建立（不使用工具链）

新工程工作空间的建立步骤因使用或者不使用工具链而不同。

本产品不包括工具链。能在安装 C/C++ 编译程序包的环境中使用工具链。

请用以下方法建立新的工作空间：

1. 请在[Welcome!]（欢迎！）对话框中选择[Create a new project workspace]（建立新的工程工作空间）单选按钮，单击[OK]按钮。

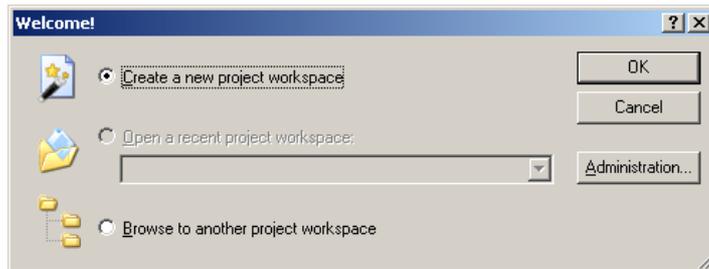


图 3.2 [Welcome!]（欢迎！）对话框

2. 开始 Project Generator。

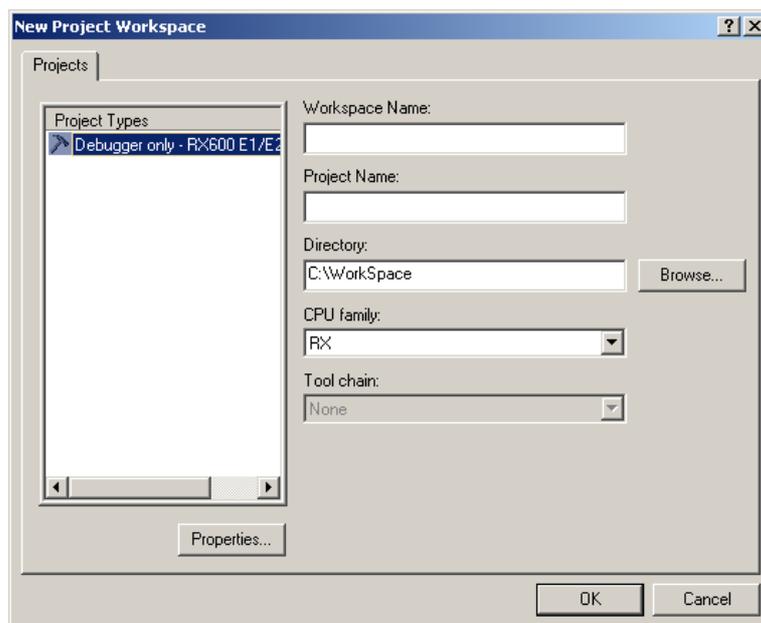


图 3.3 [New Project Workspace]（新的工程工作空间）对话框

[Workspace Name]（工作空间名）： 请输入工作空间名。

[Project Name]（工程名）： 请输入工程名。

如果可以和工作空间同名，就不需要输入。

[Directory]（目录）： 请输入要建立工作空间的目录。

能单击 [Browse]（浏览）按钮，选择要建立工作空间的目录。

[CPU family]（CPU 族）： 请选择相应的 CPU 族。

其他列表框用于设定工具链，在没有安装工具链的情况下显示固定信息。

请单击 [OK] 按钮。

3. 选择调试目标。

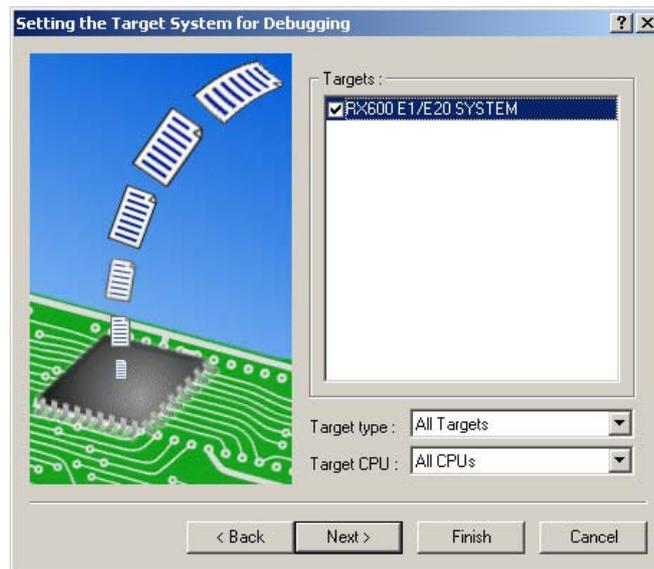


图 3.4 [Setting the Target System for Debugging] (设定调试的目标系统) 对话框

请选定要使用的目标平台，单击[Next] (下一步) 按钮。

4. 设定配置名。

配置文件是保存仿真器以外的高性能嵌入式工作室的状态文件。

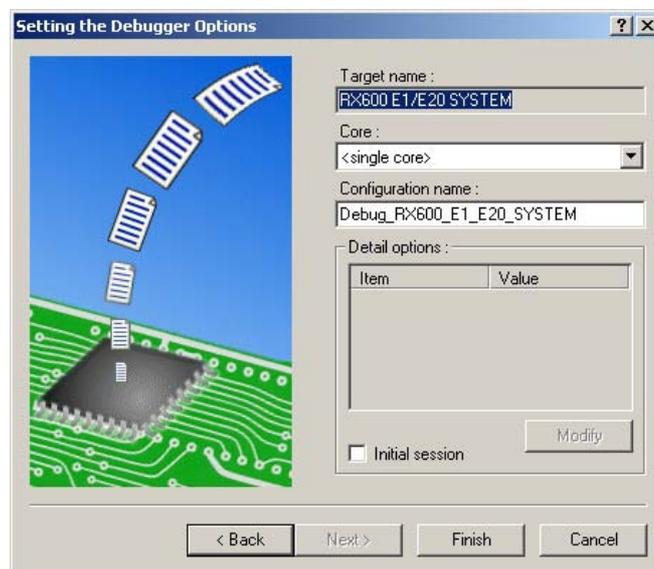


图 3.5 [Setting the Debugger Options] (设定调试的程序选项) 对话框

如果选定多个目标平台，就单击[Next] (下一步) 按钮，给每个选定的目标平台设定配置名。

一旦设定完配置名，就结束有关仿真调试程序的设定。

当单击[Finish] (完成) 按钮时，显示[Summary] (概要) 对话框。

当单击[OK]时，启动High-performance Embedded Workshop。

5. 请在启动后连接仿真器。

3.4 新工作空间的建立（使用工具链）

请用以下方法建立新的工作空间：

1. 请在[Welcome!]（欢迎！）对话框中选择[Create a new project workspace]（建立新的工程工作空间）单选按钮，单击[OK]按钮。

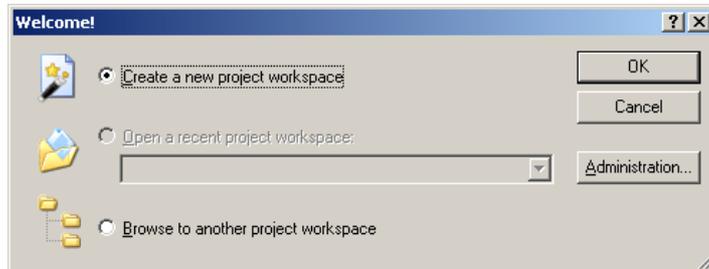


图 3.6 [Welcome!]（欢迎！）对话框

2. 开始Project Generator。

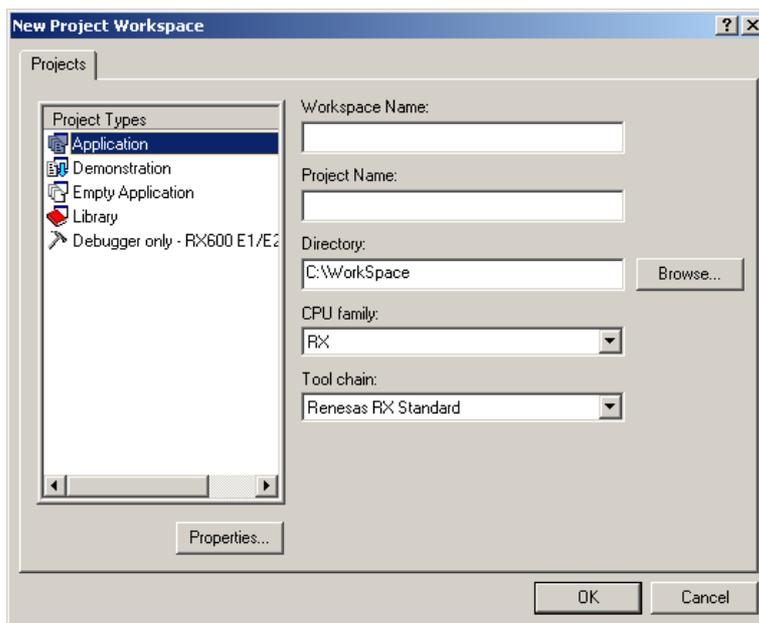


图 3.7 [New Project Workspace]（新的工程工作空间）对话框

- [Workspace Name]（工作空间名）： 请输入工作空间名。
- [Project Name]（工程名）： 请输入工程名。
如果可以和工作空间同名，就不需要输入。
- [Directory]（目录）： 请输入要建立工作空间的目录。
能单击[Browse]（浏览）按钮，选择要建立工作空间的目录。
- [CPU family]（CPU族）： 请选择相应的CPU族。
- [Tool chain]（工具链）： 当使用工具链时，请选择相应的工具链。
当不使用时，请选择[None]（无）。

请在结束上述项目的设定后按[OK]按钮。

3. 请进行工具链的CPU设定、选项设定以及其他需要的设定。
4. 选择调试目标。

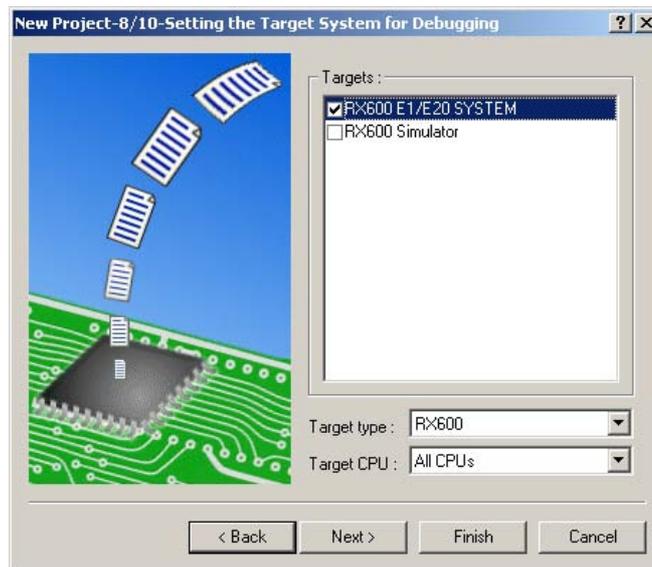


图 3.8 [New Project -8/10- Setting the Target System for Debugging]（新工程 8/10- 设定调试的目标系统）对话框

请选定要使用的目标平台，单击[Next]（下一步）按钮。

5. 设定配置名。

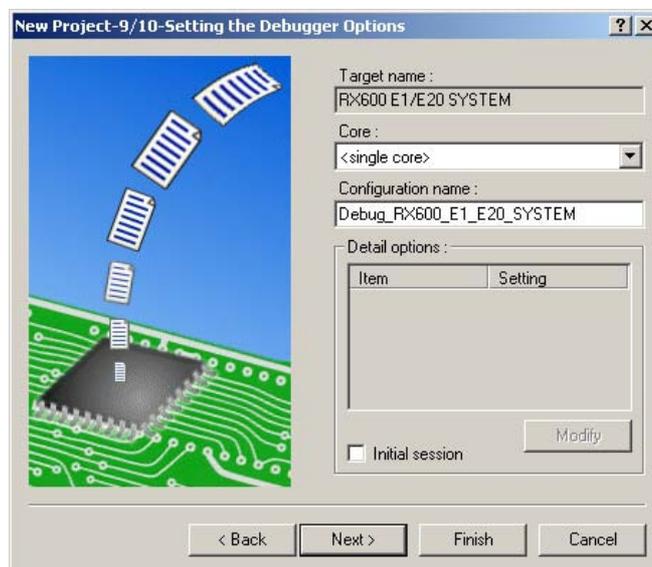


图 3.9 [New Project -9/10-Setting the Debugger Options]（新工程 9/10- 设定调试的程序选项）对话框

如果选定多个目标平台，就单击[Next]（下一步）按钮，给每个选定的目标平台设定配置名。一旦设定完配置名，就结束有关仿真调试程序的设定。当单击[Finish]（完成）按钮时，显示[Summary]（概要）对话框。当单击[OK]时，启动High-performance Embedded Workshop。

6. 请在启动后连接仿真器。

3.5 现有工作空间的打开

用以下方法打开现有工作空间：

1. 请在[Welcome!]（欢迎！）对话框中选择[Browse to another project workspace]（浏览到另一个工作空间）单选按钮，单击[OK]按钮。

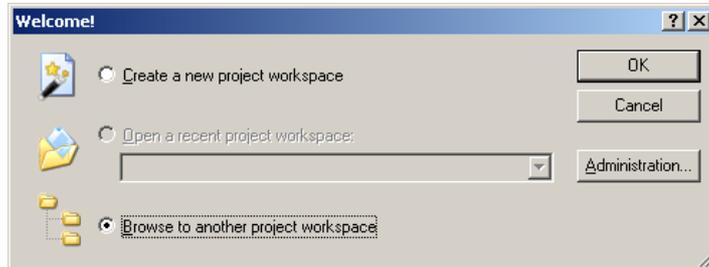


图 3.10 [Welcome!]（欢迎！）对话框

2. 打开[Open Workspace]（打开工作空间）对话框。

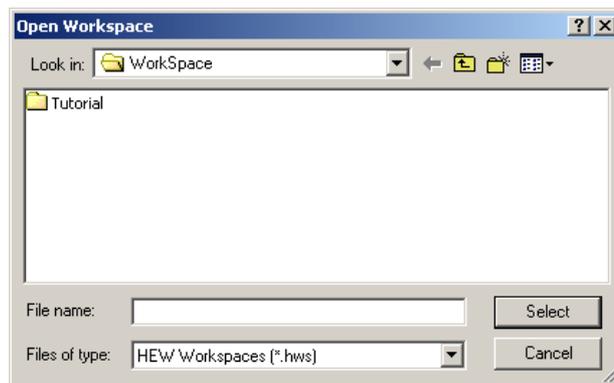


图 3.11 [Open Workspace]（打开工作空间）对话框

请指定建立工作空间的目录，选择工作空间文件（扩展名.hws），单击[Select]（选择）按钮。

3. 启动High-performance Embedded Workshop，恢复指定工作空间的保存状态。
如果指定的工作空间保存了连接仿真器的状态，就自动连接仿真器。
当指定的工作空间保存了没有连接仿真器的状态时，仿真器的连接方法请参照“3.6 仿真器的连接”。

3.6 仿真器的连接

3.6.1 仿真器的连接

仿真器的连接方法如下：

1. 启动仿真器时设定后的连接方法
请选择 [Debug]（调试）-> [Debug Settings]（调试设置），打开 [Debug Settings]（调试设置）对话框。在此，能选择目标并且注册下载模块以及自动执行的命令链等。
如果在 [Debug Settings]（调试设置）对话框中选择目标后按 [OK] 按钮，就连接仿真器。
2. 对话文件的加载
能通过切换到事先注册了使用仿真器的设定的对话文件，简单地连接仿真器。

3.6.2 仿真器的重新连接

在断开仿真器的状态下，能用以下任意一种方法重新连接：

1. 选择 [Debug]（调试）-> [Connect]（连接）。
2. 单击 [Connect]（连接）工具栏按钮（）。
3. 在 [Command Line]（命令行）窗口中输入“connect”命令。

3.7 仿真器的断开

3.7.1 仿真器的断开

能用以下任意一种方法断开连接中的仿真器：

1. 选择 [Debug]（调试）-> [Disconnect]（断开）。
2. 单击 [Disconnect]（断开）工具栏按钮（）。
3. 在 [Command Line]（命令行）窗口中输入“disconnect”命令。

3.8 High-performance Embedded Workshop 的结束

如果选择 [File]（文件）-> [Exit]（退出），就结束 High-performance Embedded Workshop。
在结束前，显示对话保存的确认信息框。要保存对话时，请单击 [Yes] 按钮。

3.9 调试设置

注册下载模块，设定命令行批文件的自动执行以及设定下载时的选项等。

3.9.1 下载模块的指定

选择 [Debug]（调试）-> [Debug Settings]（调试设置），打开 [Debug Settings]（调试设置）对话框。

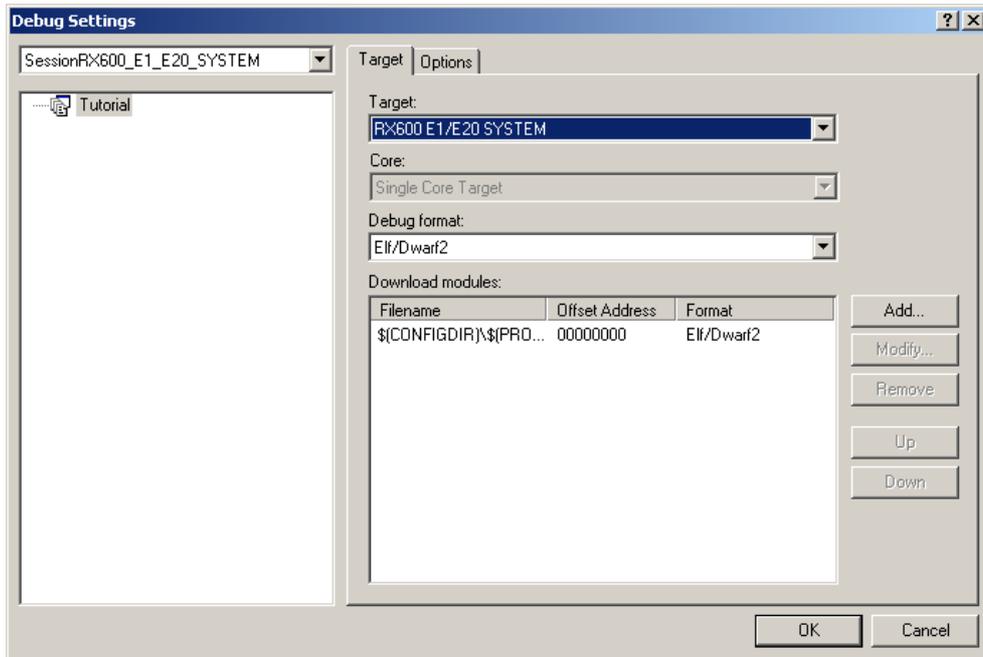


图 3.12 [Debug Settings]（调试设置）对话框

请在 [Target]（目标）下拉式列表框中选择要连接的产品名称。

请在 [Debug format]（调试格式）下拉式列表框中选择要下载的装入模块格式，并且将其对应的下载模块注册到 [Download modules]（下载模块）列表框。

【注】 此时不下载程序。有关下载方法请参照“5.2 下载功能”。

3.9.2 命令行批文件的自动执行的设定

请单击 [Options] (选项) 的页面。

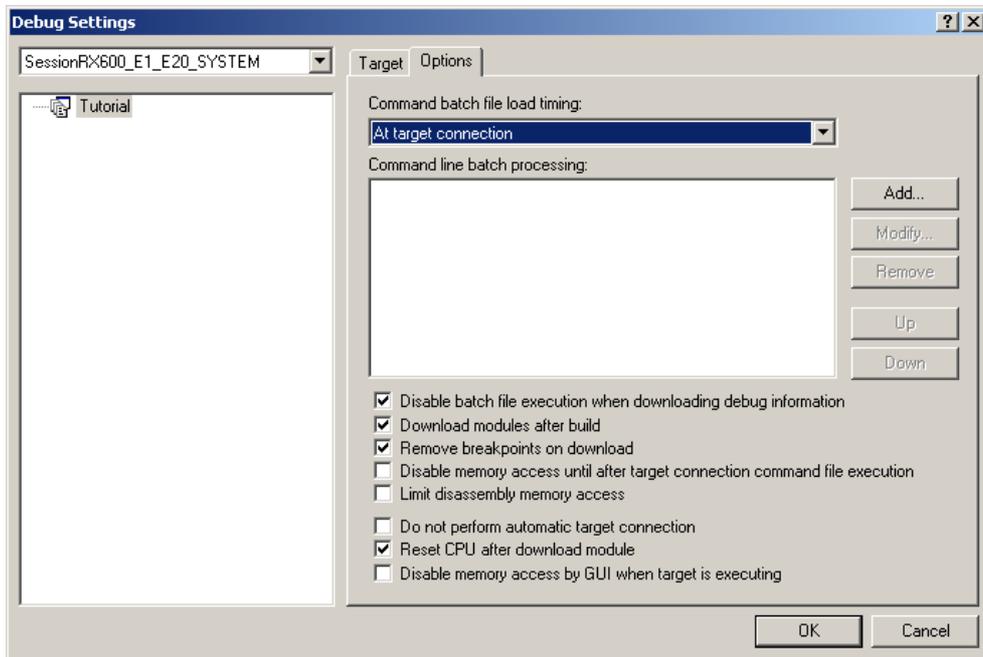


图 3.13 [Debug Settings] (调试设置) 对话框

在此，注册在指定时间自动执行的命令链。

能指定的时间有以下 4 种：

- 连接仿真器时 (At target connection)
- 下载模块前 (Before download of modules)
- 下载模块后 (After download of modules)
- 复位后 (After reset)

请在 [Command batch file load timing] (命令批文件加载时间) 下拉式列表框中指定命令链的执行时间。请在指定时间后将要执行的命令批文件注册到 [Command Line Batch Processing] (命令批处理) 列表框。

3.10 E1/E20 仿真调试程序的启动步骤

在本章节中，启动 High-performance Embedded Workshop，检查 E1/E20 仿真器是否正确连接用户系统上的单片机。

在此，使用本产品中附带的教程工作空间进行检查。

首先，请确认以下内容：

- 确认用户系统的电源是否切断。
- 将用户接口电缆的一端连接E1/E20仿真器的用户连接器，另一端连接用户系统的仿真器连接器。
- 用USB接口电缆连接E1/E20仿真器和主机。

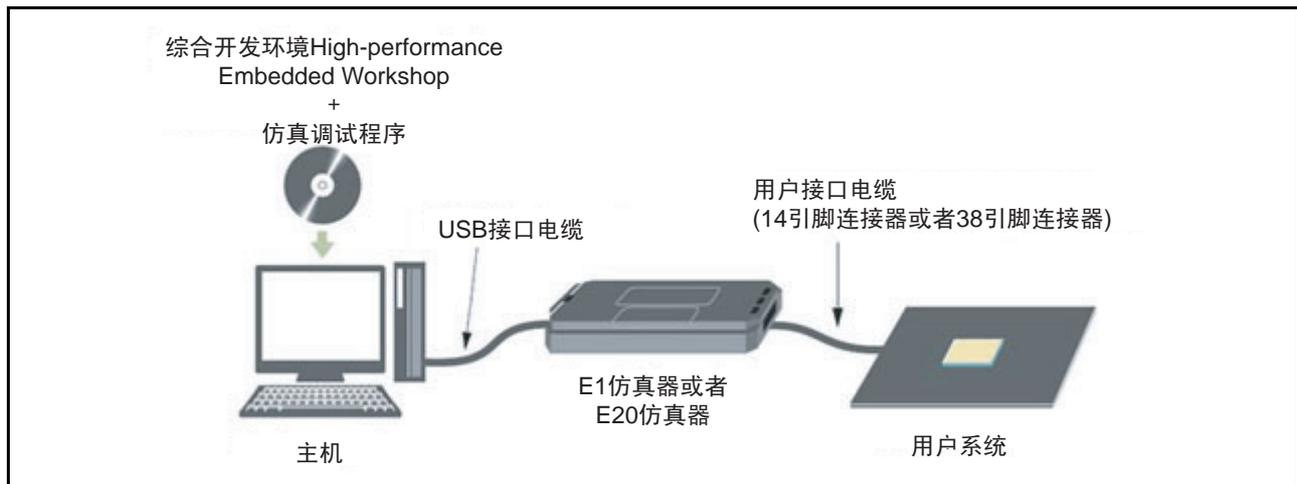


图 3.14 系统检查的构成

1. 请从[Start]（开始）菜单的[Programs]（程序）中选择[Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop]。

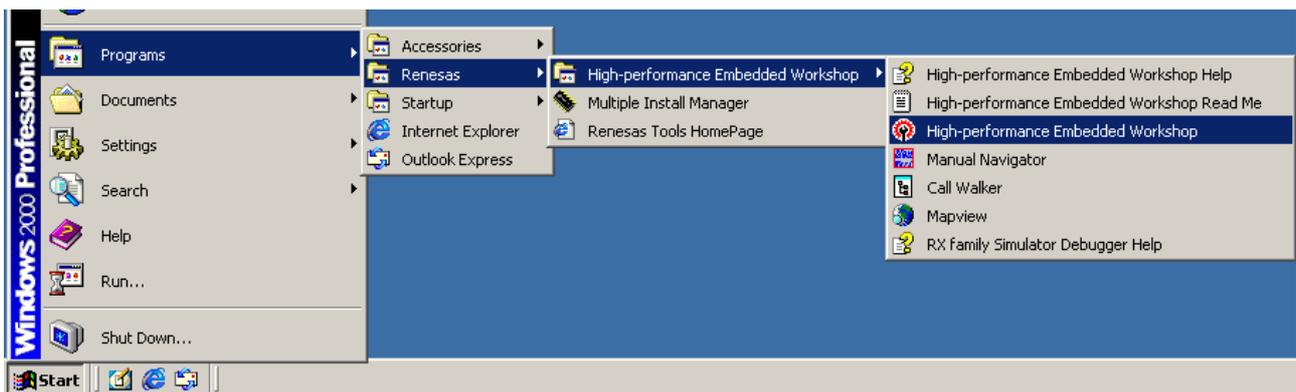


图 3.15 [Start]（开始）菜单

2. 显示[Welcome!]（欢迎！）对话框。

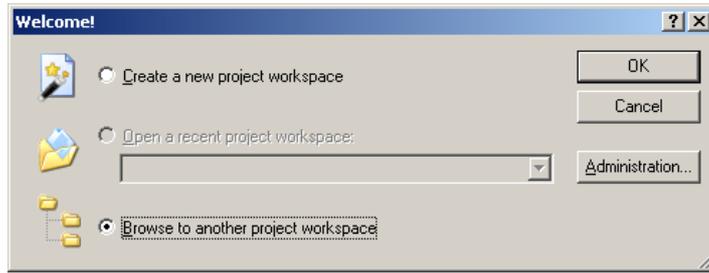


图 3.16 [Welcome!]（欢迎！）对话框

在此，因为使用教程工作空间，所以请选择[Browse to another project workspace]（浏览到另一个工作空间）单选按钮，单击[OK]按钮。

打开[Open Workspace]（打开工作空间）对话框，请指定以下目录：

<安装OS的驱动器>:\WorkSpace\Tutorial\E1E20\RXxxx\Tutorial_LittleEndian

xxx表示对象产品群。

请在指定目录后选择以下文件（Tutorial.hws），按[Select]（选择）按钮。

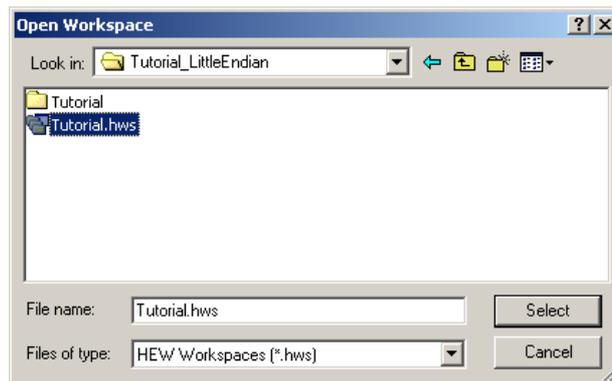


图 3.17 [Open Workspace]（打开工作空间）对话框

3. 显示[Initial Settings]（初始设置）对话框。在此对话框中进行目标单片机的设定以及通信设定。请在设定各项目后按[OK]按钮。

有关[Initial Settings]（初始设置）对话框的详细说明，请参照“4.3.1 [Initial Settings]（初始设置）对话框”。

有关热插拔请参照“5.13 热插拔功能”。

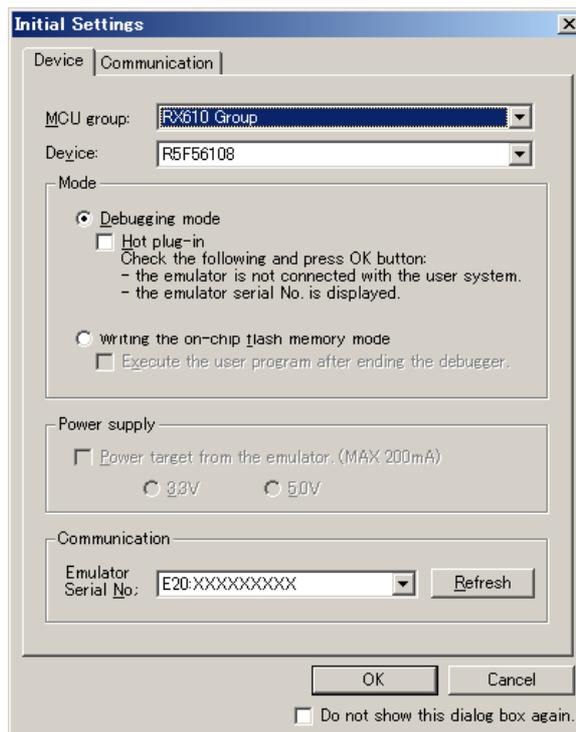


图 3.18 [Initial Settings] (初始设置) 对话框的 [Device] (设备) 页面

【注意事项】

[Initial Settings] (初始设置) 对话框显示的内容可能因各单片机而不同。

以下所示的2个注意事项只是针对 E1 仿真器的内容。E20 仿真器不支持供电功能。

【使用仿真器的供电功能给用户系统提供电源时的注意事项】

如果在没有从外部电源给用户系统提供电源的状态下忘记选定 [Power target from the emulator] (从仿真器给目标供电) 项目, 就显示以下对话框。

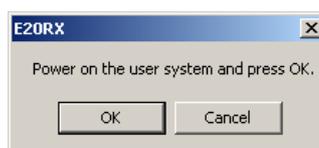


图 3.19 供电确认对话框

如果按 [OK] 按钮, 就显示 [Power Supply] (电源) 对话框。请在选定各项目后按 [OK] 按钮, 给用户系统供电。

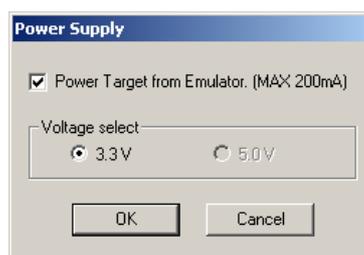


图 3.20 [Power Supply] (电源) 对话框

【使用外部电源给用户系统供电时的注意事项】

如果在从外部电源给用户系统提供电源的状态下选定了 [Power target from the emulator]（从仿真器给目标供电）项目，就显示以下对话框。

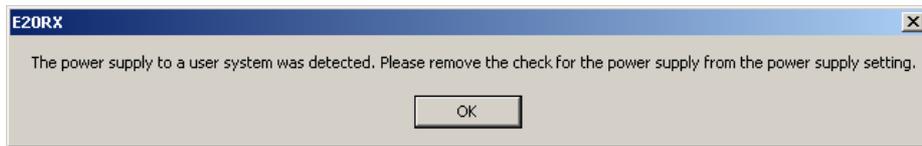


图 3.21 用户系统的供电确认对话框

如果按 [OK] 按钮，就显示 [Power Supply]（电源）对话框。请在撤销 [Power target from the emulator]（从仿真器给目标供电）复选框的选定后按 [OK] 按钮，不从仿真器供电。



图 3.22 [Power Supply]（电源）对话框

4. 如果在设定 [Initial Settings]（初始设置）对话框后按 [OK] 按钮，就显示 [Connecting...]（正在连接...）对话框。



图 3.23 [Connecting...]（正在连接...）对话框

5. 如果仿真器固件是旧版本，就显示[Confirm Firmware]（确认固件）对话框。
如果按[OK]按钮，就开始更新仿真器的固件。必须注意：一旦开始更新仿真器的固件，就不能中途取消。



图 3.24 [Confirm Firmware]（确认固件）对话框

在更新仿真器固件的过程中显示[Downloading]（下载）进度栏。更新结束后重新开始仿真器的启动处理。

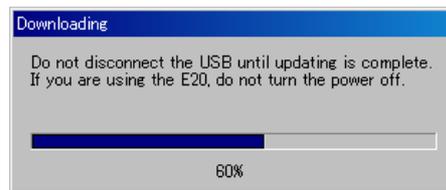


图 3.25 [Downloading]（下载）进度栏

注意

绝对不能在更新结束前拔掉 USB 接口电缆，否则会损坏 E1/E20 仿真器。

6. 如果在目标单片机的内部闪存中设定了ID码，就显示[ID Code verification]（ID码验证）对话框。请在输入所设定的ID码后按[OK]按钮。

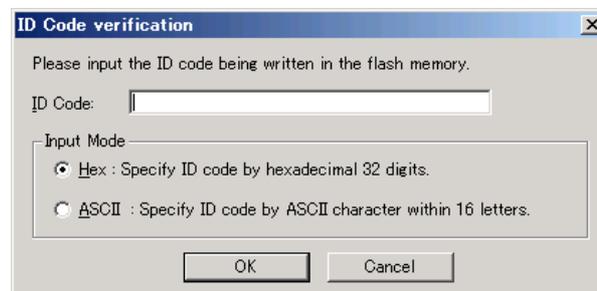


图 3.26 [ID Code verification]（ID码验证）对话框

7. 然后，显示 [Configuration Properties]（配置属性）对话框。在此对话框中进行有关仿真器和调试功能的设定。请在设定各项目后按 [OK] 按钮。

[Configuration Properties]（配置属性）对话框的详细说明，请参照“4.3.2 [Configuration Properties]（配置属性）对话框”。

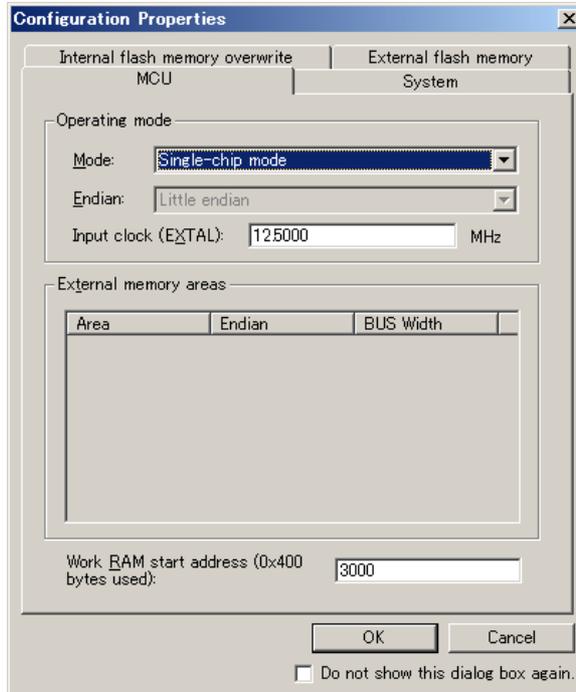


图 3.27 [Configuration Properties]（配置属性）对话框 [MCU] 页面

【注意事项】

[Configuration Properties]（配置属性）对话框显示的内容可能因各单片机而不同。

8. 在设定 [Configuration Properties]（配置属性）对话框并且按 [OK] 按钮后，如果在 High-performance Embedded Workshop 的 [Output]（输出）窗口中显示“Connected”（已连接），就表示 E1/E20 仿真器启动结束。

如果在 [Output]（输出）窗口中显示的内容不是“Connected”（已连接），请参照表 3.3。

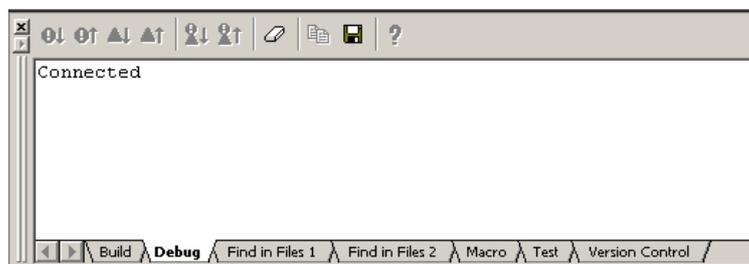


图 3.28 [Output] 窗口

【注】 在用户程序已经被下载到内部闪存的情况下，因为在启动 E1/E20 仿真器后没有用户程序的调试信息，所以不能进行源级调试。为了进行源级调试，请在启动 E1/E20 仿真器后下载含有调试信息的下载模块文件。

表 3.3 [Output] (输出) 窗口的错误信息

错误信息	对策
LEVEL0 的版本不同, 请更新调试程序的版本。	使用的仿真器需要更新调试程序软件的版本, 请从本公司主页下载。
The device ID code does not match the one for the selected device. Please check the device name. (设备 ID 码和所选设备的 ID 码不同, 请确认启动时指定的设备名。)	在初始设置中指定的设备名可能和连接对象的 MCU 不同, 请确认在初始设置中指定的设备名。
The power voltage has exceeded 5.9V. Please check the user system setting. (电源电压超过 5.9V, 请确认用户系统的设定。)	给用户系统提供的电源电压可能超过了规定值, 请确认用户系统的设定。
A JTAG communication error. Please retry with reducing the JTAG clock. (发生 JTAG 通信错误, 请降低 JTAG 时钟频率。)	请确认下述内容: <ul style="list-style-type: none"> • 用户接口电缆的接触不良 • 单片机引脚的接触不良 • JTAG 时钟频率过高或者过低。

4. 使用 RX610 群、RX621 群、RX62N 群、RX62T 群时的软件规格

4.1 E1/E20 仿真器规格一览表

用于 RX610 群、RX621 群、RX62N 群、RX62T 群的 E1/E20 仿真器规格一览表如表 4.1 所示。

表 4.1 用于 RX610 群、RX621 群、RX62N 群、RX62T 群的 E1/E20 仿真器规格

项目	E1	E20
对应单片机群	RX 族 RX600 系列 RX610 群、RX621 群、RX62N 群、RX62T 群	←
对应运行模式【注 1】【注 8】	单芯片模式、内部 ROM 有效扩展模式、 内部 ROM 无效扩展模式	←
S/W 暂停	最多 256 点	←
事件（不能检测由 DMAC/DTC 进行的存取）		
事件数【注 2】	执行地址：8 点 数据存取：4 点	←
通过次数【注 3】	最多 256 次	←
On-chip 暂停		
执行前 PC 暂停【注 2】	最多 8 点	←
事件组合【注 2】	OR / AND（累积）/ 顺序	←
其他	跟踪满暂停	←
跟踪（不能记录由 DMAC/ DTC 进行的存取）	内部跟踪（最多获取 256 个转移或者周期 转移信息和数据存取信息）	<ul style="list-style-type: none"> 内部跟踪（最多获取 256 个转移或者周期转移信息和数据存取信息） 外部跟踪输出（获取约 2M 个转移或者周期转移信息和数据存取信息）
性能测量	最多 2 点的执行周期或者执行次数（32 位 计数器 ×2 个或者 64 位计数器 ×1 个）	←
实时 RAM 监视（不能监视由 DMAC/DTC 进行的存取）	无	显示 4K 字节（1K 字节 ×4 块）读写的最后 存取履历。
调试控制台	在 [output]（输出）窗口中显示 printf 的输出。	←
用户接口	14 引脚连接器	38 引脚连接器或者 14 引脚连接器【注 4】

表 4.1 用于 RX610 群、RX621 群、RX62N 群、RX62T 群的 E1/E20 仿真器规格 (续)

项目	E1	E20
和主机的接口	USB2.0 (全速 / 高速) 【注 5】	←
和用户系统的连接	用附属的用户接口电缆进行连接。	←
仿真器电源	不需要 (由主机给 USB 总线供电)	←
供电功能 【注 6】	可给用户系统提供 3.3V 或者 5.0V (最大 200mA)。	无
外部闪存的下载	有	←
热插拔 【注 7】 【注 9】	有	←

- 【注】
1. 不对应用户引导模式。
 2. On-chip 暂停和跟踪共用事件，不能同时在多个功能中使用同一个事件。
 3. 只能对某一个事件指定通过次数。
 4. 在不使用外部跟踪输出功能和实时 RAM 监视功能时，E20 仿真器也能使用 14 引脚连接器。此时，需要 38 引脚 /14 引脚转换适配器。
 5. 也能连接对应 USB1.1 的主机。
 6. 在将 E1 仿真器用作批量生产工程中的闪存编程器时，请不要使用供电功能，而将另外准备的符合单片机规格的电源提供给用户系统。
 7. 在将热插拔功能用于 E1 仿真器时，请购买另售的热插拔适配器 (R0E000010ACB00)。
 8. RX62T 群只对应单芯片模式。
 9. 不对应内部 ROM 无效扩展模式。

4.2 E1/E20 仿真器和单片机的不同点

(1) 寄存器的初始值

请注意：E1/E20 仿真器在系统启动时对通用寄存器和部分控制寄存器进行初始化（表 4.2）。单片机的初始值为不定值。

表 4.2 E1/E20 仿真器的寄存器初始值

状态	寄存器名	E1/E20 仿真器
启动 E1/E20 仿真器时	R0(SP)	00000000h
	R1 ~ R15	00000000h
	USP	00000000h
	ISP	00000000h
	PSW	00000000h
	PC	上电复位向量表中的 PC 值
	INTB	00000000h
	BPSW	00000000h
	FINTV	00000000h
	FPSW	00000100h
	ACC【注】	00000000h

【注】 ACC 寄存器的 b15-b0 的读取值总是为“0”，忽视 b15-b0 的写操作。

【注意事项】

如果在[Registers]（寄存器）窗口中更改 PSW 寄存器的处理器中断优先级，被更改的内容就在开始执行用户程序前实际反映到 PSW 寄存器。通过 REGISTER_SET 命令进行的更改也一样。

(2) 低功耗状态

在使用 E1/E20 仿真器时，除了低功耗状态的解除源以外，也能通过 [Stop]（停止）按钮来解除状态并且暂停执行。

(3) 复位信号

在暂停期间，屏蔽单片机的复位信号。

【注意事项】

不能在 RES# 和 WAIT# 引脚为“Low”电平的状态下暂停用户程序，否则会发生超时错误。另外，如果在暂停期间 WAIT# 引脚固定为“Low”电平状态，就会在存取存储器时发生超时错误。

(4) 直接存储器存取控制器（DMAC）

即使在暂停期间，DMAC 也不停止。因此，如果发生传送请求，就执行 DMA 传送。请参照使用时的注意事项的“有关 DMAC 和 DTC”。

(5) 有关 WDT 的使用

在暂停期间，WDT 停止计数。在重新开始执行程序时，WDT 也重新开始计数。

(6) SCKCR 寄存器的变更和调试功能的竞争

在用户程序执行过程中，不能在 [Memory]（存储器）等窗口中更改 SCKCR 寄存器的值。

(7) 用户引脚的占有

因为以下引脚与用户端口多路复用，所以在调试时用户不能使用。有关多路复用的用户端口，请参照所用单片机的硬件手册。

- TRST#
- TCK
- TMS
- TDI
- TDO

对于 RX621 群和 RX62N 群的单片机，如果将分配为 TDO 的引脚设定为漏极开路输出，此引脚就变为漏极开路输出，与用作 TDO 无关。不能在调试时将 P26 设定为漏极开路输出。

在使用 E20 并且通过 38 引脚连接器连接用户系统时，用户系统不能使用以下引脚。当使用 38 引脚 /14 引脚转换适配器时，能作用用户端口。

- TRCLK
- TRSYNC#
- TRDATA[3:0]

(8) 有关可屏蔽中断

- 即使在用户程序停止期间（包括运行时间调试期间），定时器等功能仍在运行。因为在用户程序停止期间（包括运行时间调试期间）仿真器禁止中断，所以即使发生可屏蔽中断的请求也不接受。在开始执行用户程序后接受此中断请求。
- 请注意：在用户程序停止期间（包括运行时间调试期间），不接受外围 I/O 的中断请求。

4.3 调试程序的设定

在连接仿真器时，显示 [Initial Settings]（初始设置）对话框和 [Configuration Properties]（配置属性）对话框。在此，请选择有关仿真器的各选项。

请注意：对于调试对象的目标单片机等，只能在启动时设定一次。

4.3.1 [Initial Settings]（初始设置）对话框

[Initial Settings]（初始设置）对话框是进行有关目标单片机设定的对话框。设定的内容在下次启动时也有效（但是，热插拔和供电源的设定除外）。

在启动仿真器后，能通过选择 [Setup -> Emulator -> Device setting]（设置 -> 仿真器 -> 器件设置）重新打开此对话框，但是在启动后不能更改设定。

(1) 目标单片机的选择

在 [Initial Settings]（初始设置）对话框的 [Device]（器件）页面中指定要仿真的目标单片机。

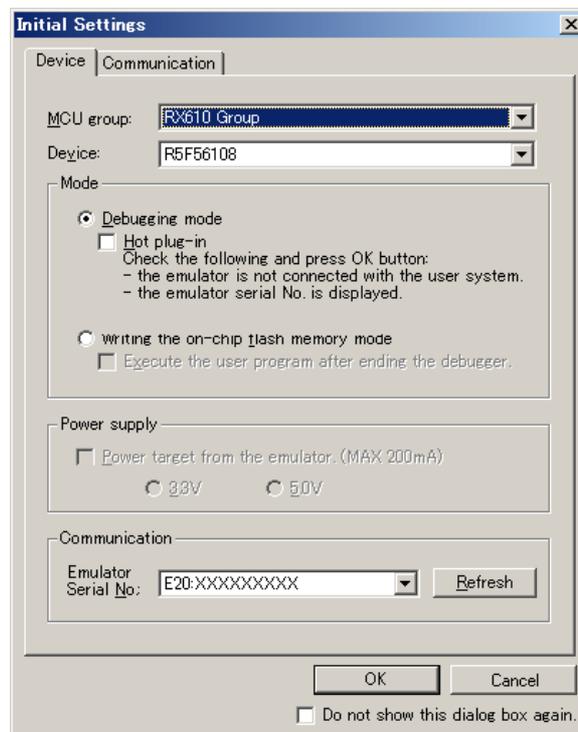


图 4.1 [Initial Settings]（初始设置）对话框的 [Device]（器件）页面

【注意事项】

[Initial Settings]（初始设置）对话框显示的内容可能因各单片机而不同。

在启动仿真器后不能更改在此设定的目标单片机。

要更改目标单片机时，需要解除一次仿真器的启动，然后重新启动仿真器。

面向 RX600 系列并且 E1/E20 仿真调试程序能选择的目标单片机如下所示。

表 4.3 目标单片机一览表

MCU 群	器件名
RX610 群	R5F56104、R5F56106、R5F56107、R5F56107、R5F56108
RX621 群	R5F56216、R5F56217、R5F56218
RX62N 群	R5F562N7、R5F562N8
RX62T 群	R5F562T6、R5F562T7、R5F562TA

(2) 运行模式的选择

从以下选项中选择运行模式：

“Debugging mode（调试模式）”、“Writing the on-chip flash memory mode（写入内部闪存编程模式）”

- **【Debugging mode（调试模式）】**

在将 E1/E20 用作调试器时选择此模式。

因此，在此模式中下载程序后不能拆除 E1/E20 而单独运行用户系统。

在此模式中，不能将用于禁止读内部闪存 ROM 的 ID 码写到单片机的内部闪存。

- (a) 选择热插拔。

如果选定 [Hot plug-in]（热插拔）的复选框，就能在执行用户程序的中途开始调试。有关热插拔请参照“5.13 热插拔功能”。

- **【Writing the on-chip flash memory mode（写入内部闪存编程模式）】**

在将 E1/E20 用作编程器时选择此模式。

因此，在此模式中不能调试程序。

在此模式中下载程序时，能将用于禁止读内部闪存 ROM 的 ID 码写到单片机的内部闪存。

(3) 供电设定的选择

指定从 E1 仿真器给用户系统提供电源的方法。要从 E1 仿真器给用户系统提供电源时，在选定 [Power target from the emulator. (MAX 200mA)]（从仿真器给目标供电（最大 200mA））的复选框后，选择“3.3V”或者“5.0V”。

能选择的供电因目标单片机而不同。有关单片机的工作电压，请参照单片机的硬件手册。

E20 仿真器不支持供电功能，因此不能选择此选项。

(4) 通信设定的选择

设定 USB 的连接目标。

在仿真器系列 No. 列表框中，显示通过 USB 连接的仿真器的固有识别信息一览表。

如果单击 [Refresh]（刷新）按钮，就更新固有识别信息一览表。

(5) JTAG 通信设定的选择

在 [Initial Settings]（初始设置）对话框的 [Communications]（通信）页面中，设定仿真器和用户系统之间的 JTAG 通信速度【注】。

从以下选项（单位：MHz）中选择通信速度。

“16.5”、“12.38”、“6.188”、“3.094”、“1.547”

【注】 根据用户系统上的 JTAG 信号的布线长度和布线方法，有可能不能以所选的 JTAG 时钟频率进行通信。通过降低频率，也许能进行通信。



图 4.2 [Initial Settings]（初始设置）对话框的 [Communications]（通信）页面

不能在启动仿真器后更改在此设定的通信信息。

要更改通信信息时，需要解除一次仿真器的启动，然后重新启动仿真器。

4.3.2 [Configuration Properties]（配置属性）对话框

[Configuration Properties]（配置属性）对话框是进行有关仿真器和调试功能设定的对话框。设定的内容在下次启动时也有效。

在启动仿真器后，能通过选择 [Setup -> Emulator -> System]（设置 -> 仿真器 -> 系统）重新打开此对话框。在启动后能更改部分选项的设定。

只有激活显示的设定内容是能更改的选项，而非激活（灰色）显示的设定内容是不能更改的选项。

(1) 有关仿真器和调试功能的设定

在 [Configuration Properties]（配置属性）对话框的 [MCU] 页面中进行有关目标单片机的设定。

在 [Initial Settings]（初始设置）对话框之后接着显示 [Configuration Properties]（配置属性）对话框。

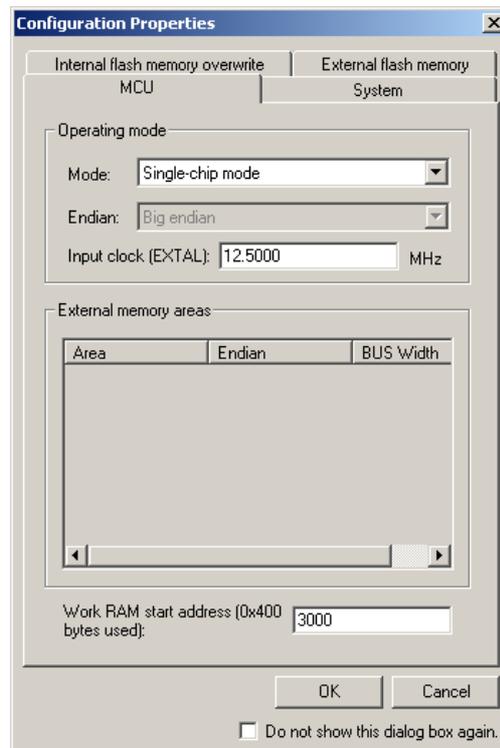


图 4.3 [Configuration Properties]（配置属性）对话框的 [MCU] 页面

【注意事项】

[Configuration Properties]（配置属性）对话框显示的内容可能因各单片机而不同。

(a) 运行模式的选择

从以下选项中选择运行模式。

“Single-chip mode（单芯片模式）”、“On-chip ROM enabled extended mode（内部 ROM 有效扩展模式）”、“On-chip ROM disabled extended mode（内部 ROM 无效扩展模式）”

如果在 [target MCU]（目标单片机）中选择 RX62T 群单片机，运行模式的选项就只能选择 “Single-chip mode（单芯片模式）”。

(b) 字节序

仿真器读用户系统的 MDE 引脚状态，显示以下字节序的设定信息：

“Big endian（小端法）”、“Little endian（大端法）”

(c) 输入时钟（EXTAL）的选择

给目标单片机设定所提供的运行时钟。

输入值的范围是 0.0001 ~ 99.9999。如果输入的值不在此范围内，就显示错误信息。

输入值的小数点第 5 位以后的数被舍去。

(d) 外部区域的选择（对象为 RX610 群、RX621 群、RX62N 群单片机）

设定外部区域的字节序和总线宽度。

能在运行模式为内部 ROM 有效扩展模式或者内部 ROM 无效扩展模式时进行设定。

如果选择并且双击外部区域的设定行，就显示 [External Area]（外部区域）对话框，能按各区域设定字节序和总线宽度。

- 能从以下选项中选择字节序：

“Same endian as the MCU（和 MCU 的字节序相同）”、“Different endian from the MCU（和 MCU 的字节序不同）”

- 能从以下选项中选择总线宽度：

RX610 群单片机：“8bit”、“16bit”

RX621 群、RX62N 群单片机：“8bit”、“16bit”、“32bit”

外部区域因目标单片机而不同。有关单片机的外部区域，请参照单片机的硬件手册。

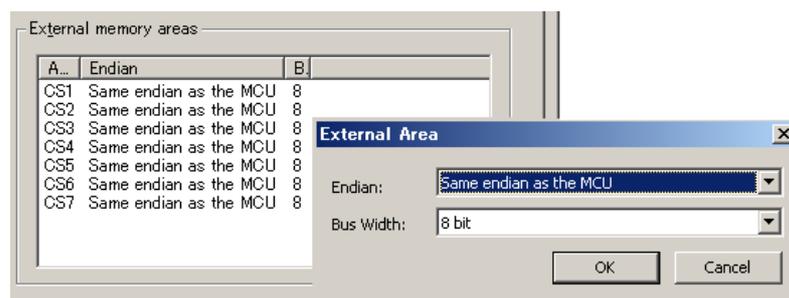


图 4.4 [External Area]（外部区域）设定对话框

(e) 作业 RAM 的选择

选择调试程序使用的作业 RAM 分配地址。

从指定作业 RAM 的分配地址开始，E1/E20 仿真调试器的固件使用所指定的字节数。为了进行存储器内容的压栈和退栈，用户程序也能使用此区域，但是不能将此区域指定为 DMA 或者 DTC 功能的传送源和传送目标。

(2) 有关系统的设定

在 [Configuration Properties] (配置属性) 对话框的 [System] (系统) 页面中进行有关仿真器系统总体的设定。



图 4.5 [Configuration Properties] (配置属性) 对话框的 [System] (系统) 页面

(a) 选择 [Exclusive Function] (专用功能)。

跟踪条件设定功能和实时 RAM 监视功能不能同时使用，只能选择其中一个。

初始设置为“Trace conditions”（跟踪条件）。

如果使用 E1 仿真器，就不能选择实时 RAM 监视功能（复选框灰色显示）。

(b) 使用 [Debugging the program re-writing the internal flash.] (调试有改写内部闪存功能的程序)。

在调试有改写内部闪存（程序 ROM 区）功能的程序时进行指定。

要调试有改写内部闪存（程序 ROM 区）功能的程序时，必须选定 [Debugging the program re-writing the internal flash.] (调试有改写内部闪存功能的程序) 复选框。

(3) 闪存 ROM 的盖写设定

在 [Configuration Properties]（配置属性）对话框的 [Internal flash memory overwrite]（内部闪存盖写）页面中，进行闪存 ROM 各块的盖写设定。

在启动后也能选择 [Setup -> Emulator -> System]（设置 -> 仿真器 -> 系统），打开此对话框并且更改设定。

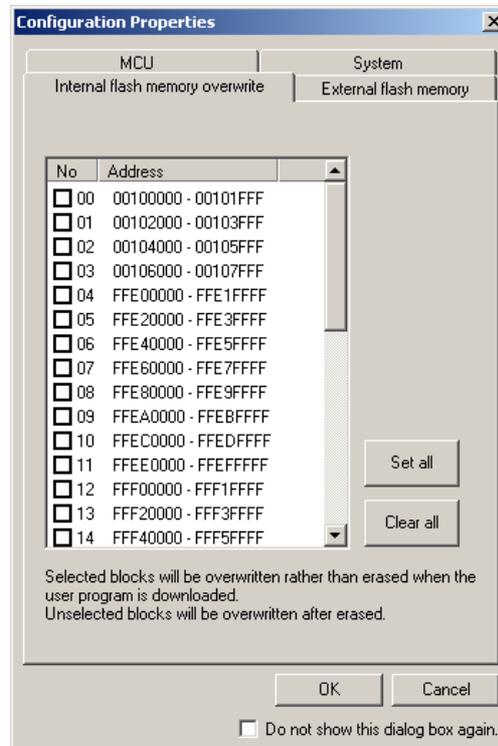


图 4.6 [Configuration Properties]（配置属性）对话框的 [Internal flash memory overwrite]（内部闪存盖写）页面

在列表中自动显示所选目标单片机的各块的设定。

如果选定复选框，就在下载时盖写被选择的块（合并）。而对于没有选定的块，因为在擦除后下载程序，所下载的程序中没有数据的区域为闪存的初始值。

(4) 外部闪存信息的注册（对象为 RX610 群、RX621 群、RX62N 单片机）

在 [Configuration Properties]（配置属性）对话框的 [External flash memory]（外部闪存）页面中进行外部闪存信息的注册。在将程序下载到连接单片机外部总线地址空间的闪存（称为外部内存）时进行设定。

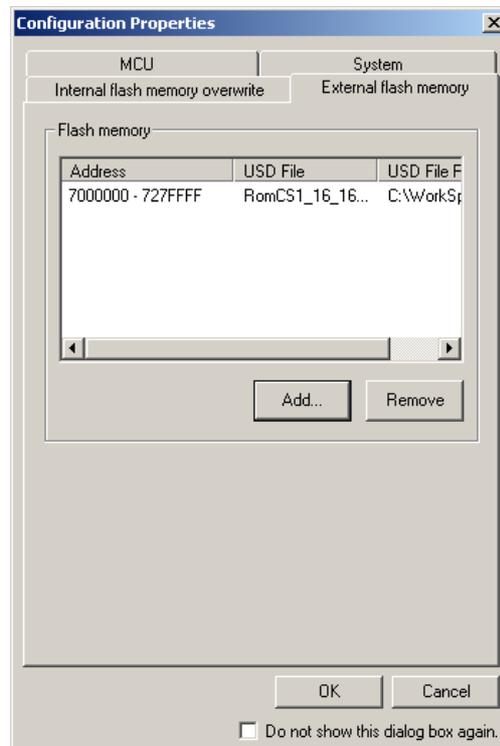


图 4.7 [Configuration Properties]（配置属性）对话框的 [External flash memory]（外部闪存）页面

通过在 [External flash memory]（外部闪存）页面中注册 USD 文件（用户系统的外部闪存连接形态等的定义文件），识别外部闪存。

最多能注册 4 个 USD 文件。

USD 文件由“External Flash Definition Editor”建立。有关 USD 文件的建立方法，请参照以下 URL 刊载的“External Flash Definition Editor”手册。

<http://www.renesas.com/efe>

(a) 外部闪存信息的注册

通过在 [MCU] 页面的 [Mode]（模式）中选择“On-chip ROM enabled extended mode（内部 ROM 有效扩展模式）”或者“On-chip ROM disabled extended mode（内部 ROM 无效扩展模式）”，[External flash memory]（外部闪存）页面的 [Add]（添加）按钮变为有效。

如果单击 [External flash memory]（外部闪存）页面的 [Add]（添加）按钮，就打开 USD 文件的文件选择对话框。如果在文件选择对话框中指定 USD 文件，就在 [Flash memory]（闪存）列表中显示指定 USD 文件的信息。

(b) 外部闪存信息的删除

请从 [Flash memory]（闪存）列表中选择要删除的外部闪存信息，单击 [Remove]（移除）按钮。

(c) 外部闪存的盖写设定

如果从 [Flash memory]（闪存）列表中双击外部闪存信息，就显示 [External flash memory overwrite]（外部闪存的盖写）对话框。

在 [External flash memory overwrite]（外部闪存的盖写）对话框中进行外部闪存各扇区（块）的盖写设定。

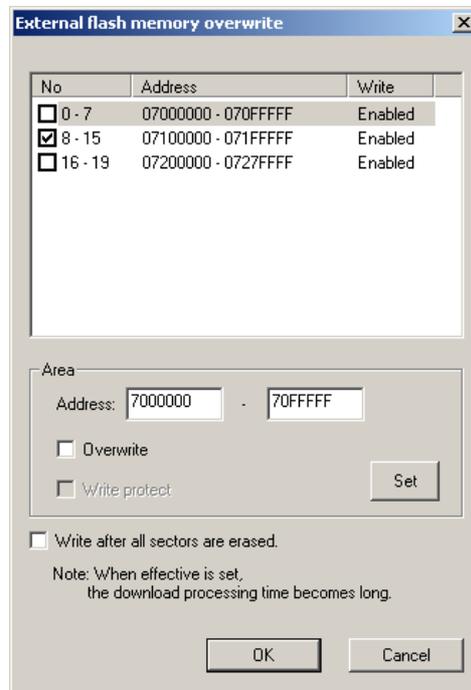


图 4.8 [External flash memory overwrite]（外部闪存的盖写）对话框

在列表中自动显示所选外部闪存信息的各扇区的设定。如果选定复选框，就在下载时盖写（合并）所选择的扇区。

要按扇区更改设定时，请在 [Address]（地址）编辑框中输入要更改设定的开始地址和结束地址，单击 [Set]（设置）按钮。此时，如果选定 [Overwrite]（盖写），就在下载时不擦除而盖写所设定的扇区。另外，不对应 [Write protect]（写保护）的设定。如果选定 [Write after all sectors are erased]（在擦除全部扇区后写），就在下载时擦除全部扇区。

5. E1/E20 仿真器的功能

本章节介绍 E1/E20 仿真器的功能。

各功能的使用方法请参照“6. 教程”。

5.1 功能概要

E1/E20 仿真器的功能概要如表 5.1 所示。

表 5.1 E1/E20 仿真器的功能

No.	项目	功能
1	用户程序的执行功能	<ul style="list-style-type: none"> • 在单片机保证的工作频率范围内执行程序。 • 复位仿真 • Step 功能 <ul style="list-style-type: none"> 单 Step 功能（1 Step: 1 条指令） 源级 Step 功能（1 Step: 1 行源码） Step Over 功能（不在子程序内暂停） 源级 Step Over 功能（不在子程序内暂停） Step Out 功能（执行返回到 PC 执行中的子程序调用源函数） • 转至光标。 • 复位执行 • 自由执行 • 暂停源的显示
2	复位功能	<ul style="list-style-type: none"> • 在暂停期间，从 High-performance Embedded Workshop 给单片机发行上电复位。
3	跟踪功能	<ul style="list-style-type: none"> • 内部跟踪功能（通过单片机的内部跟踪缓冲器来实现） • 外部跟踪输出功能 <ul style="list-style-type: none"> 跟踪输出优先模式（没有实时性） CPU 执行优先模式（有实时性，但是获取的数据可能会有欠缺（丢失）） • 跟踪内容 <ul style="list-style-type: none"> 转移（转移源 / 转移目标）跟踪、数据跟踪 • 检索功能 • 滤波器功能 • 用总线 / 反汇编 / C 源级显示
4	暂停功能	<ul style="list-style-type: none"> • S/W 暂停功能 • On-chip 暂停功能 <ul style="list-style-type: none"> 执行前 PC 暂停功能 事件暂停功能（OR/AND（累积） / 顺序 / 通过次数） 跟踪满暂停功能 • 强制暂停功能
5	性能测量功能	<ul style="list-style-type: none"> • 通过单片机的内部计数器进行 Point to Point 的经过周期数等的测量功能

No.	项目	功能
6	存储器存取功能	<ul style="list-style-type: none"> • 下载到 RAM • 下载到内部闪存 • 下载到外部闪存 • 1 行汇编 • 反汇编 • 读存储器 • 写存储器 • 用户程序执行中的变量内容的自动更新显示 • 填充 • 检索 • 移动 • 存储器数据的保存 / 读 • 存储器数据的验证 • 变量内容的显示 • 实时 RAM 监视 • 存储器数据的自动更新
7	通用 / 控制寄存器的存取功能	读写通用 / 控制寄存器
8	内部 I/O 寄存器的存取功能	读写内部 I/O 寄存器 【注 1】
9	源级调试功能	丰富的源级调试功能
10	命令行功能	支持命令输入。 如果建立将各命令按输入顺序排列的文件，就能进行批处理。
11	帮助功能	记载各功能的操作方法以及能从命令行窗口输入的命令语法。
12	热插拔功能	支持给正在执行的程序连接 E1/E20 并且能在执行程序中途开始调试的启动方式。

【注】 1. 在 [IO] 窗口中显示由 “xxxx.io” 定义的内容。能通过编辑 “xxxx.io” 的内容，添加或者删除显示的寄存器。有关 “xxxx.io” 中记载的全部内容，请参照 High-performance Embedded Workshop 用户手册的 “参考 6. I/O 文件格式”。

“xxxx.io” 被保存在以下目录中（xxxx 表示单片机群名）：

< High-performance Embedded Workshop 文件夹 >

\Tools\Renesas\DebugComp\Platform\E1E20\E20RX\IOFiles

5.2 下载功能

下载要调试的装入模块。

在下载程序时，从 [Debug -> Download]（调试 -> 下载）中选择装入模块，或者从用鼠标右键单击 [Workspace]（工作空间）窗口的 [Download modules] 的装入模块而显示的弹出式菜单中选择 [Download]（下载）。能通过选择并且双击装入模块进行下载。

【注】 要下载程序时，需要将装入模块注册到 High-performance Embedded Workspace。

5.3 源文件的打开

5.3.1 源码的显示

如果在选择源文件后单击 [Open]（打开）按钮，High-performance Embedded Workshop 就打开综合化编辑器的文件，或者能通过双击 [Workspace] 窗口的源文件进行显示。

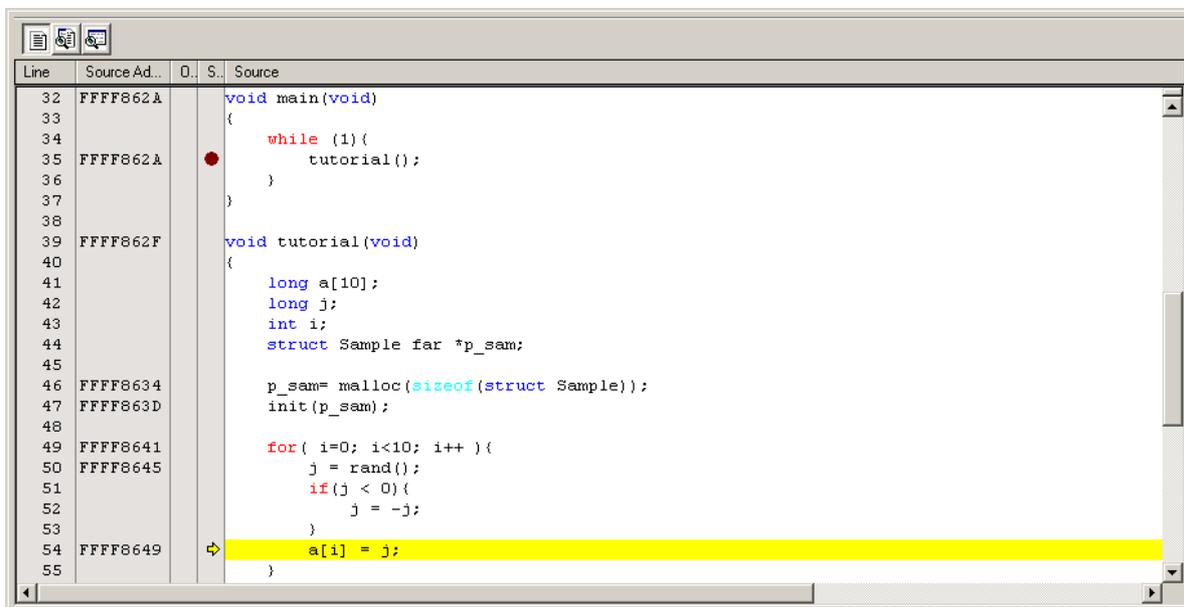


图 5.1 [Editor]（编辑器）窗口

在此窗口的左边显示以下的行信息：

- | | | |
|-------|-----------------------------------|--------------|
| 第 1 列 | [Source Address]（源地址）列 | 地址信息 |
| 第 2 列 | [On-Chip Breakpoint]（On-chip 断点）列 | On-chip 断点信息 |
| 第 3 列 | [S/W Breakpoint]（S/W 断点）列 | PC、书签、断点信息 |

源地址列

如果下载程序，就在源地址列中显示当前源文件的地址。
此功能便于决定在哪里设定 PC 值和断点。

On-chip 断点列

在 On-chip 断点列中显示以下标志：

- 设定 On-chip 断点的地址条件。
能对可设定的事件个数进行设定。
个数因单片机而不同。

通过双击 On-chip 断点列，出现上述位图。

S/W 断点列

在 S/W 断点列中显示以下标志：

-  设定了书签。
-  设定了 PC Break。
-  PC 位置

要关闭全部源文件的列时

1. 请用鼠标右键单击 [Editor]（编辑器）窗口。
2. 请单击 [Define Column Format...]（定义列的格式...）菜单项目。
3. 显示 [Global Editor Column States]（全局编辑器列状态）对话框。
4. 复选框表示该列有效或者无效，在有选定时有效。如果复选框是灰色显示，就表示一部分文件的列有效而另一部分文件的列无效。
请从要关闭的列的复选框中撤销选定。
5. 请单击 [OK] 按钮，使新设定的列有效。

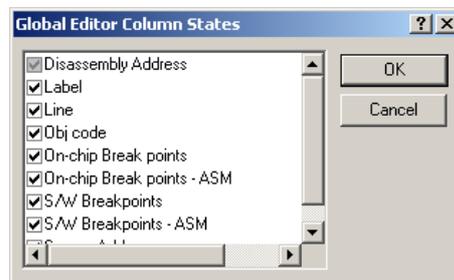


图 5.2 [Global Editor Column States]（全局编辑器列状态）对话框

要关闭 1 个源文件的列时

1. 请打开要删除的列的源文件，在 [Editor]（编辑器）窗口上单击鼠标右键，打开弹出式菜单。
2. 请单击 [Columns]（列）菜单项目，出现叠层的菜单项目，在此弹出式菜单中显示各列。当列有效时，在名称的边上有选定标记。如果单击该条目，就进行列的显示/不显示的转换。

5.3.2 汇编语言代码的显示

在打开源文件的状态下，请单击窗口上部的 [Disassembly]（反汇编）工具栏按钮。显示与打开的源文件对应的汇编语言代码。

当没有源文件也想用汇编语言级显示代码时，选择 [View -> Disassembly...]（视图 -> 反汇编 ...）或者单击 [Disassembly]（反汇编）工具栏按钮 。

在当前 PC 的位置打开 [Disassembly]（反汇编）窗口，并且显示表示反汇编助记符（也有可能一起显示标签）的 [Address] 和 [Code]（选项）。

如果选择 [Mixed display]（混合显示）工具栏按钮 ，就能显示源文件和代码。以下是选择 [Mixed display]（混合显示）时的显示例子。

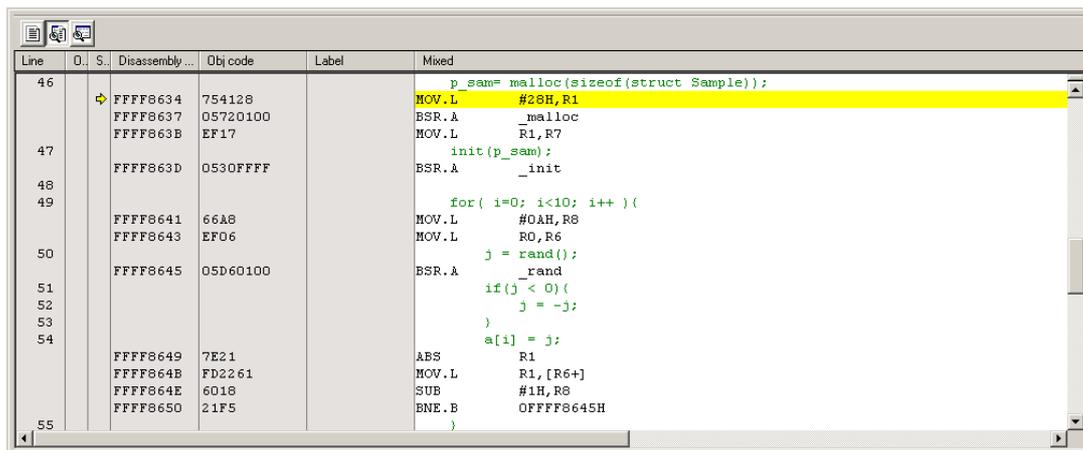


图 5.3 [Disassembly]（反汇编）窗口

5.3.3 汇编语言代码的修改

通过双击要修改的指令，能修改汇编语言代码。打开 [Assembler]（汇编）对话框。



图 5.4 [Assembler]（汇编）对话框

显示地址、机器码和反汇编指令。输入新的指令或者编辑 [Mnemonic]（助记符）域的当前指令。如果按“Enter”键，就在将指令汇编到存储器后移到下一条指令；如果单击 [OK] 按钮，就在将指令汇编到存储器后关闭对话框；如果单击 [Cancel]（取消）按钮或者按“Esc”键，就关闭对话框。

【注】 汇编语言的显示内容是对实际存储器中的机器码进行反汇编的内容。如果修改存储器的内容，就在对话框（和 [Disassembly]（反汇编）窗口）中显示新的汇编语言代码，但是 [Editor]（编辑器）窗口的显示内容不变。这与源文件中含汇编的情况相同。

5.4 存储器的存取功能

E1/E20 仿真器有以下的存储器存取功能。

5.4.1 存储器的读写功能

[Memory]（存储器）窗口：在窗口中显示存储器内容。

在打开 [Memory]（存储器）窗口时只读所指定的大小。

因为仿真器内没有高速缓存，所以总是发生读周期。

当从 [Memory]（存储器）窗口进行写操作时，因为更新窗口，所以在 [Memory]（存储器）窗口中读显示范围。

不想更新 [Memory]（存储器）窗口时，如果在窗口上单击鼠标右键，从显示的选项菜单中选择 [Lock Refresh]（锁定刷新），就能设定为不更新。

me 命令：这是命令行功能。

这是按指定的大小读写指定地址的功能。

(1) 用户程序的下载功能

能下载注册在工作空间内的装入模块。

在 [Debug]（调试）菜单的 [Download Module]（下载模块）中，能选择要下载的模块。

也能通过鼠标右键单击工作空间内的装入模块来打开弹出式菜单，从此弹出式菜单进行下载。

下载目标为 RAM 或者闪存。

也能通过此功能下载调试信息等源级调试所需的信息。

(2) 存储器数据的上载功能

能从指定地址开始将指定大小的数据保存到文件。

请从 [Memory]（存储器）窗口的弹出式菜单中选择 [Save]（保存）。

(3) 存储器数据的下载功能

能下载保存在文件中的存储器内容。

请从 [Memory]（存储器）窗口的弹出式菜单中选择 [Load]（加载）。

(4) 变量内容的显示

显示用户程序指定的变量内容。

(5) 实时 RAM 监视功能

能通过调试对象的单片机来监视用户程序执行过程中的存储器内容。

有关规格请参照在线帮助。

(6) 存储器数据的自动更新功能

在用户程序执行过程中，能自动更新 [Memory]（存储器）窗口中显示的存储器内容。能更改更新间隔。

(7) 其他存储器操作功能

其他功能如下：

- 存储器填充功能
- 存储器移动功能
- 存储器保存功能
- 存储器验证功能
- 存储器检索功能
- 内部 I/O 显示功能
- 标签名、变量名及其内容的显示功能

有关详细内容请参照在线帮助。

5.5 暂停功能

E1/E20 仿真器有强制暂停、S/W 暂停和 On-chip 暂停共 3 种暂停功能。

5.5.1 强制暂停功能

这是强制性地暂停用户程序的功能。

5.5.2 S/W（软件）暂停功能

这是在将指定地址的起始指令替换为调试专用指令后暂停的功能。

如果设定 S/W 断点，就因替换为调试专用指令而写存储器（如果解除 S/W 断点的设定，也因要恢复为替换调试专用指令前的指令而同样写存储器）。

能通过双击 High-performance Embedded Workshop 的 [Editor]（编辑器）窗口或者 [Disassembly]（反汇编）窗口的 [S/W Breakpoint]（S/W 断点）列，设定 S/W 断点。

5.5.3 On-chip 暂停功能

On-chip 暂停功能有使用事件功能（执行 PC 事件、操作数存取事件）的执行前 PC 暂停功能、事件暂停功能和跟踪满暂停功能。能最多设定 8 点使用执行 PC 事件的暂停以及 4 点使用操作数存取事件的暂停。

有关 On-chip 暂停功能的使用方法，请参照“5.7.1 On-Chip Breakpoint（On-chip 断点）的设定”。

(1) 执行前 PC 暂停功能

这是使用执行 PC 事件并且在执行指定地址（PC）的指令前暂停的功能。即使在用户程序执行过程中也能进行设定。

能通过双击 High-performance Embedded Workshop 的 [Editor]（编辑器）窗口或者 [Disassembly]（反汇编）窗口的 [On-Chip Breakpoint]（On-chip 断点）列，设定执行前 PC 暂停。有关 [On-Chip Breakpoint]（On-chip 断点）列，请参照“5.2 下载功能”。

也能通过 [On-Chip Break]（On-chip 暂停）对话框进行设定。

(2) 事件暂停功能

这是能组合事件的暂停功能，能给事件条件设定取指令（使用执行 PC 事件）/ 数据存取（使用操作数存取事件）。事件组合能设定 OR/AND（累积）/ 顺序 / 通过次数。即使在用户程序执行过程中也能进行设定。

能通过 [On-Chip Break]（On-chip 暂停）对话框进行设定。

(3) 跟踪满暂停功能

这是在跟踪缓冲器满时暂停的功能，不能在用户程序执行过程中更改设定。

暂停功能一览表如表 5.2 所示。

表 5.2 暂停功能一览表

暂停功能	设定点数	暂停条件	闪存的改写	程序执行过程中
S/W 暂停	256 点	指定地址	有	不能设定
执行前 PC 暂停	8 点	指定地址	无	能设定
事件暂停（取指令）	8 点	指定地址	无	能设定
事件暂停（数据存取）	4 点	数据存取	无	能设定
跟踪满暂停	—	跟踪缓冲器满	无	不能设定

5.6 事件功能

5.6.1 事件的使用

事件是指在程序执行过程中发生的事件组合。

在本调试程序中能将设定的事件用作暂停功能、跟踪功能和性能功能的条件。

事件设定数因单片机而不同。

为了以后再次使用，能注册已建立的事件。

(a) 事件类型

事件有以下类型。

表 5.3 事件类型一览表

执行地址	在 CPU 执行指定地址的指令时，检测到事件。 当指定为执行前 PC 断点时，在执行指定地址的指令前事件成立；当指定为其他用途时，在执行指定地址的指令后事件成立。
数据存取	在存取指定地址或者指定地址范围时，检测到事件。

(b) 事件组合

能至少使用 2 个事件来指定以下的组合条件。

表 5.4 事件组合一览表

OR	当指定事件中的任意一个事件成立时，条件成立。
AND (累积)	与时间轴无关，当指定的全部事件都成立时，条件成立。
顺序	按指定的顺序，当指定事件成立时，条件成立。

5.6.2 事件的添加

请用以下任意一种方法添加事件：

- 通过各设定（On-chip 暂停条件设定、跟踪条件设定和性能条件设定）对话框进行添加。
- 通过 [Editor]（编辑器）窗口的 [On-Chip Breakpoint]（On-chip 断点）列进行添加（只设定 On-chip 暂停条件）。
- 通过从其他窗口的拖放进行添加。
- 从命令行进行添加。

(a) 从各设定对话框添加事件。

在此说明从 [On-Chip Break]（On-chip 暂停）对话框添加事件的方法。

1. 选择 [View -> Event -> On-chip Break]（视图 -> 事件 -> On-chip 暂停），显示 [On-chip Break]（On-chip 暂停）对话框。在 [On-chip Break]（On-chip 暂停）选项卡中设定条件和组合后，显示与设定内容对应的选项卡。在此说明选择 [OR] 选项时的事件添加方法，请在选定 [On-Chip Break]（On-chip 暂停）对话框内的 [Event Break]（事件暂停）复选框后选择 “OR”。

除了 OR 以外，有关作为选项卡被显示的执行前 PC、AND 和顺序，请参照 “5.7.1 On-Chip Breakpoint（On-chip 断点）的设定”。

单击 [Add...]（添加）按钮，或者选择并且双击要输入的行。

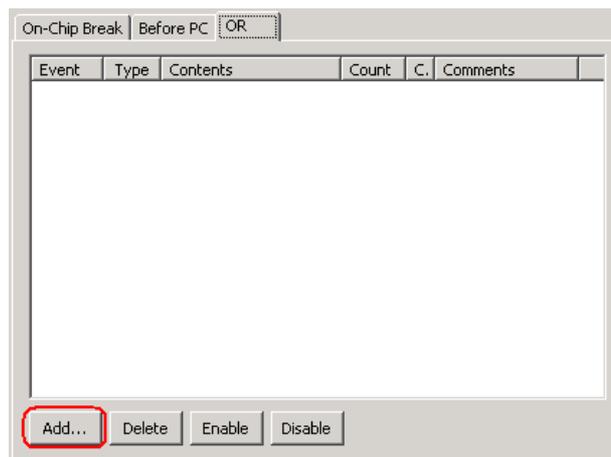


图 5.5 [On-chip Break]（On-chip 暂停）对话框

2. 显示[Event]（事件）对话框，在设定事件详细条件后单击[OK]按钮。

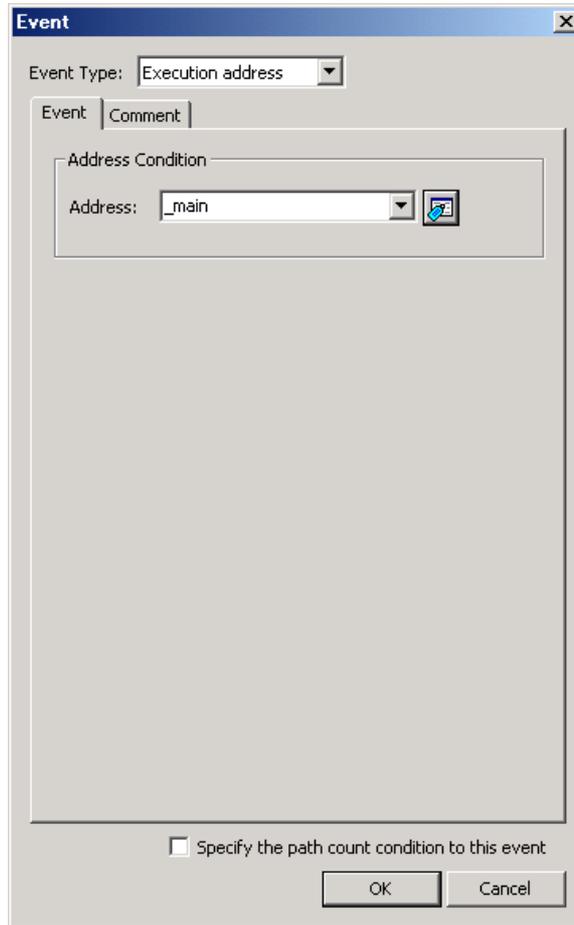


图 5.6 事件的添加

要设定事件的通过次数时，选定 [Specify the pass count condition to this event]（给此事件指定通过次数的条件）复选框。

显示 [Pass Count]（通过次数）页面，在设定事件的通过次数后单击 [OK] 按钮。

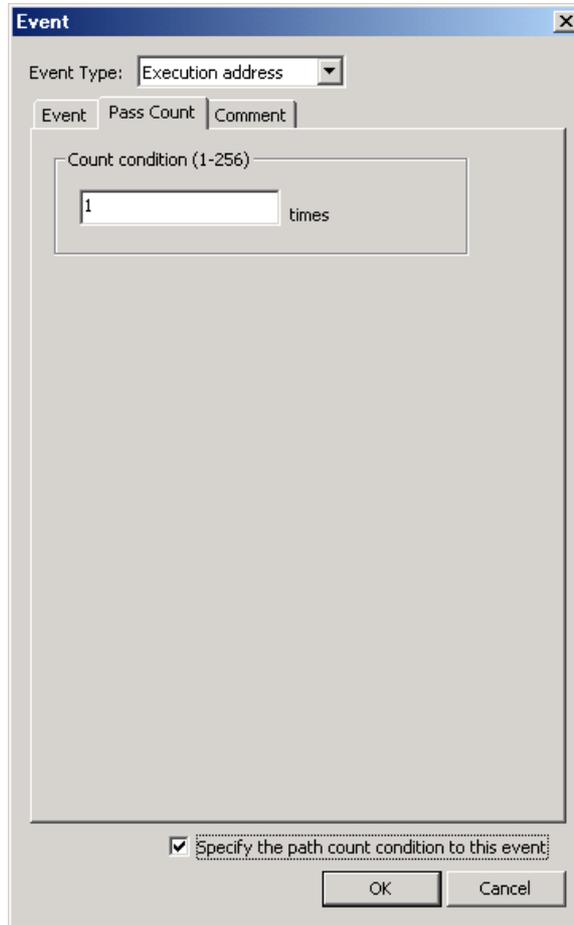


图 5.7 通过次数的设定

3. 给指定的位置添加事件。



图 5.8 事件的添加履历

4. 如果在建立事件时事件超过了能设定的最多点数，添加的事件就为无效状态。

(b) 从 [Editor] (编辑器) 窗口的 [On-chip Break points] (On-chip 断点) 列添加事件。

【添加 On-chip 断点的情况】

1. 如果双击 [Editor] (编辑器) 窗口的 [On-chip Break points] (On-chip 断点) 列, 就能设定以执行到指定地址为条件的 On-chip 断点。
=> 执行地址条件

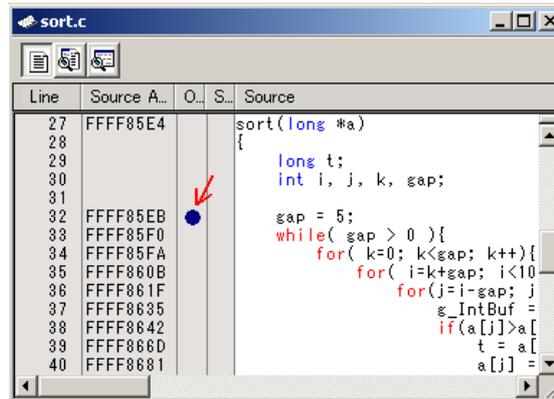


图 5.9 [Editor] (编辑器) 窗口

【注】 在 [On-Chip Break] (On-Chip 暂停) 对话框中更改设定后, 如果不单击 [Apply] (应用) 按钮, 设定内容就不会有效 (标题栏的最后有 “*”), 并且不能在 [Editor] (编辑器) 窗口的 [On-chip Break points] (On-chip 断点) 列中设定 On-chip 断点。

(c) 通过拖放添加事件。

【拖放 [Editor]（编辑器）窗口的变量名和函数名的情况】

1. 如果将变量名拖放到[On-Chip Break]（On-Chip 暂停）对话框的[Event]（事件）设定栏，就能设定以存取该变量为条件的事件。=> 数据存取条件
此时，将变量大小自动设定为数据存取事件的条件。
能注册为事件的变量是全局变量和静态变量，并且只能是1字节、2字节或者4字节大小的变量。函数内的静态变量不能注册为事件。
如果给条件和组合的设定选择执行前PC暂停（[Before PC]选项卡），就只能给事件类型设定执行地址条件，因此不能通过拖放来注册变量。
2. 如果将函数名拖放到[On-Chip Break]（On-Chip 暂停）对话框的[Event]（事件）设定栏，就能设定以执行到该函数起始地址为地址条件的事件。=> 执行地址条件

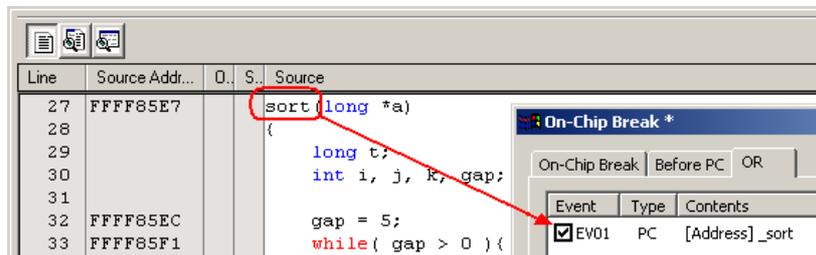


图 5.10 通过拖放添加事件

【拖放 [Memory]（存储器）窗口的地址范围的情况】

如果选择[Memory]（存储器）窗口的存储器内容并且拖放到[On-Chip Break]（On-Chip 暂停）对话框的[Event]（事件）设定栏，就能设定以所选存储器内容的地址范围为条件的数据存取事件。=> 数据存取条件

如果给条件和组合的设定选择执行前PC暂停（[Before PC]选项卡），就只能给事件类型设定执行地址条件，因此不能通过拖放[Memory]（存储器）窗口的地址范围来注册事件。

【拖放 [Label]（标签）窗口的标签的情况】

如果选择[Label]（标签）窗口的标签并且拖放到[On-Chip Break]（On-Chip 暂停）对话框的[Event]（事件）设定栏，就能设定以执行到所选标签为条件的事件。=> 执行地址条件

5.6.3 事件的删除

请用以下方法删除事件：

- 通过各设定（On-chip 暂停条件设定、跟踪条件设定和性能条件设定）对话框进行删除。

(a) 从各设定对话框删除事件。

在此说明从 [On-Chip Break]（On-Chip 暂停）对话框删除事件的方法。

1. 要删除 1 个事件（1 点）时，在 [On-Chip Break]（On-Chip 暂停）对话框的 [Event]（事件）设定栏（[OR] 选项卡内）中选择要删除的行，单击 [Delete]（删除）按钮（也能用 Ctrl + Del 键替代 [Delete]（删除）按钮进行删除）。

从 [Event]（事件）设定栏删除被选择的事件。

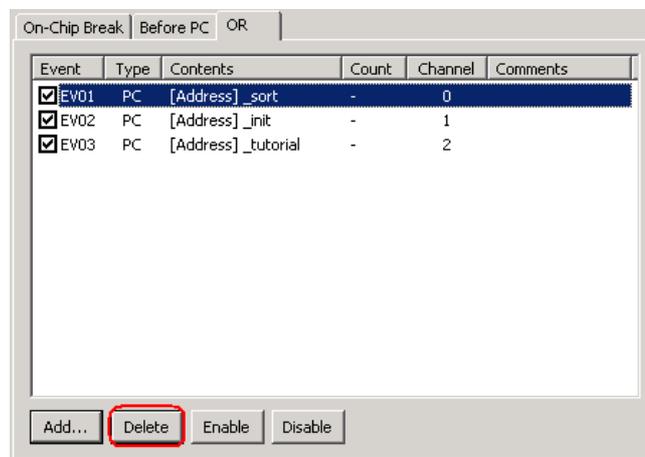


图 5.11 事件的删除

2. 要删除多个事件时，在 [On-Chip Break]（On-Chip 暂停）对话框的 [Event]（事件）设定栏中边按 Shift 键或者 Ctrl 键边选择要删除的行，单击 [Delete]（删除）按钮（也能用 Ctrl + Del 键替代 [Delete]（删除）按钮进行删除）。

从 [Event]（事件）设定栏删除被选择的事件。

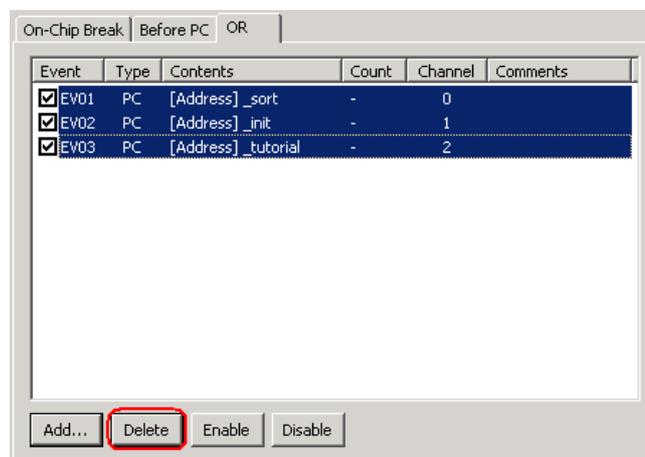


图 5.12 多个事件的删除

5.6.4 事件的注册

事件的注册是指将事件添加到 [Registered Events]（已注册事件）对话框。

注册的事件能在以后再次使用（Re-use）。

请用以下任意一种方法注册事件，最多能注册 256 个事件。

- 通过各设定（On-chip 暂停条件设定、跟踪条件设定和性能条件设定）对话框进行注册。
- 通过从各设定对话框的拖放进行添加。
- 通过 [Registered Events]（已注册事件）对话框进行注册。

(a) 从各设定对话框注册事件。

在此说明从 [On-Chip Break]（On-Chip 暂停）对话框注册事件的方法。

1. 单击 [On-Chip Break]（On-Chip 暂停）对话框的 [Event]（事件）设定栏（[OR] 选项卡内）的 [Add]（添加）按钮，打开 [Event]（事件）对话框。

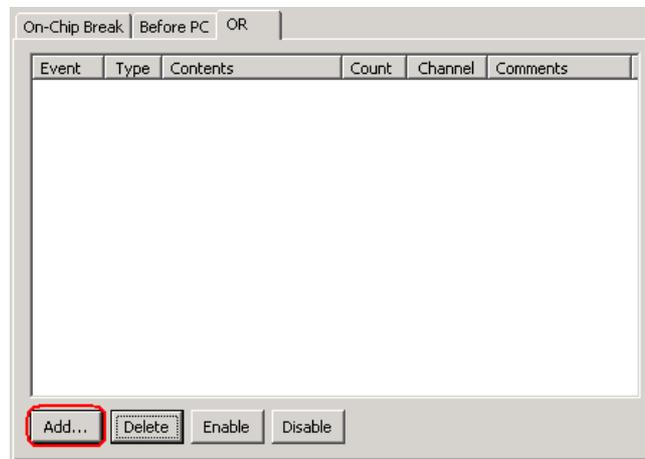


图 5.13 [On-Chip Break]（On-Chip 暂停）对话框

在 [Event]（事件）对话框中显示 [Comment]（注释）页面，请选定 [Add this event to the list]（将此事件添加到列表中）复选框。

请根据需要输入注释。

能在 [Registered Events]（已注册事件）对话框中确认注册的内容和注释。

单击 [OK] 按钮。

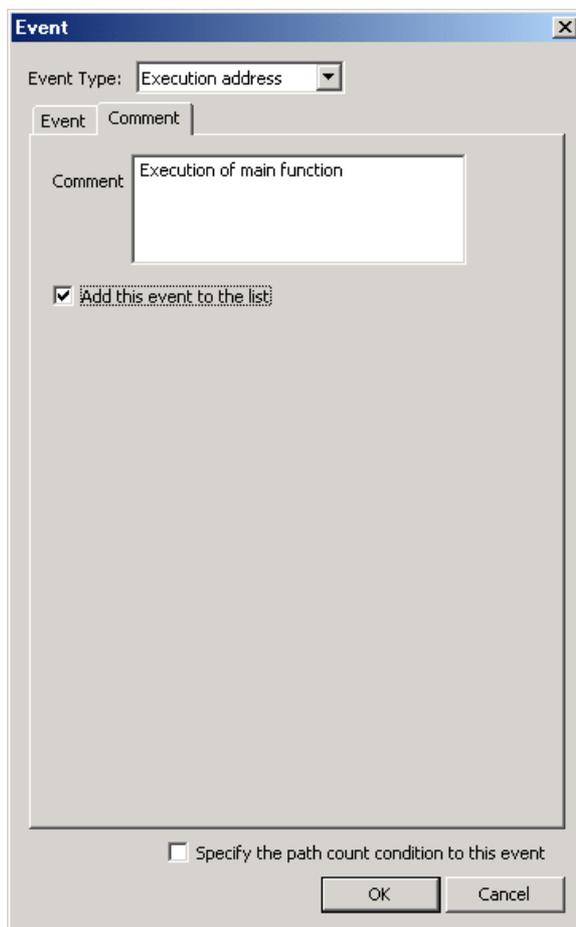


图 5.14 事件的注册

2. 将事件添加到指定的位置并且注册到 [Registered Events]（已注册事件）对话框。

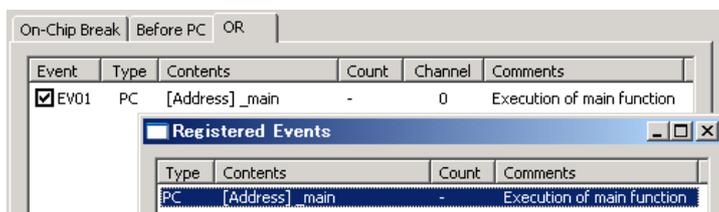


图 5.15 事件的注册履历

如果单击 [On-Chip Break] (On-Chip 暂停) 对话框下侧的 [Registered events...] (已注册事件 ...) 按钮, 就能打开 [Registered Events] (已注册事件) 对话框。

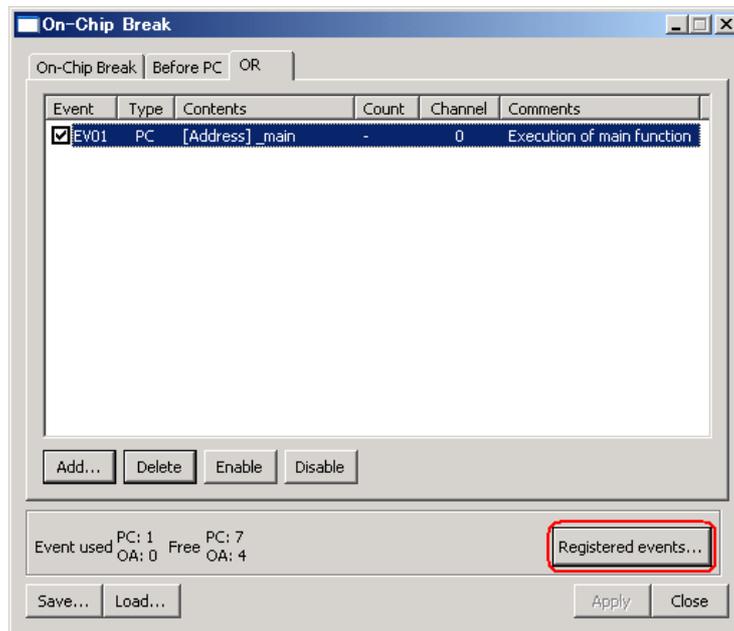


图 5.16 [Registered Events] (已注册事件) 对话框的显示方法

(b) 通过拖放注册事件。

能将建立的事件从 [On-Chip Break] (On-Chip 暂停) 对话框内的 [Event] (事件) 设定栏拖放到 [Registered Events] (已注册事件) 对话框进行注册。

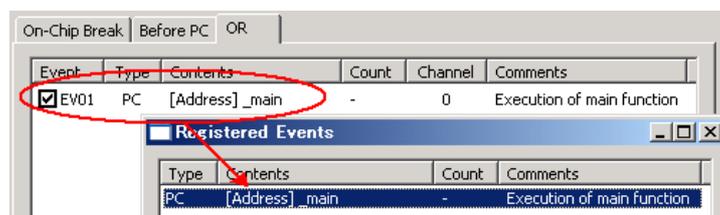


图 5.17 通过拖放注册事件

(c) 从 [Registered Events] (已注册事件) 对话框注册事件。

如果单击 [Add] (添加) 按钮, 就显示 [Event] (事件) 对话框, 建立事件。
在此建立的事件被添加到 [Registered Events] (已注册事件) 对话框。

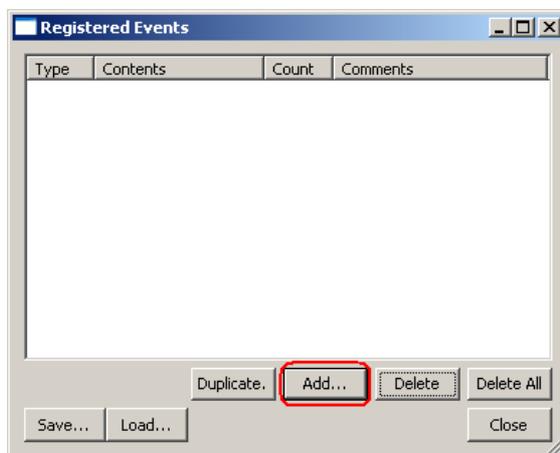


图 5.18 [Registered Events] (已注册事件) 对话框

5.6.5 注册事件的删除

请单击 [On-Chip Break] (On-Chip 暂停) 对话框下侧的 [Registered events...] (已注册事件 ...) 按钮, 打开 [Registered Events] (已注册事件) 对话框。

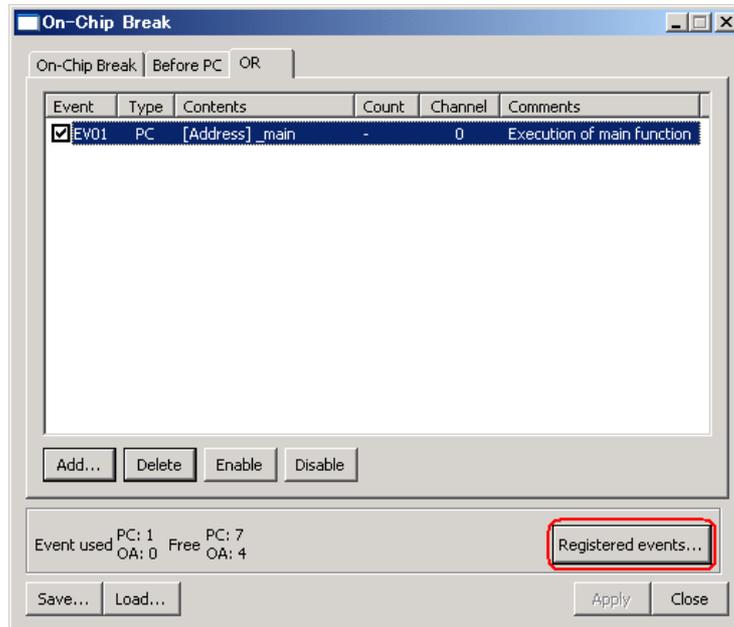


图 5.19 [Registered Events] (已注册事件) 对话框的显示方法

要删除 1 个事件时, 请在 [Registered Events] (已注册事件) 对话框中选择要删除的行, 单击 [Delete] (删除) 按钮 (也能用 Ctrl + Del 键替代 [Delete] (删除) 按钮进行删除)。

从 [Registered Events] (已注册事件) 对话框删除被选择的事件。

要删除全部事件时, 单击 [Delete All] (全部删除) 按钮。

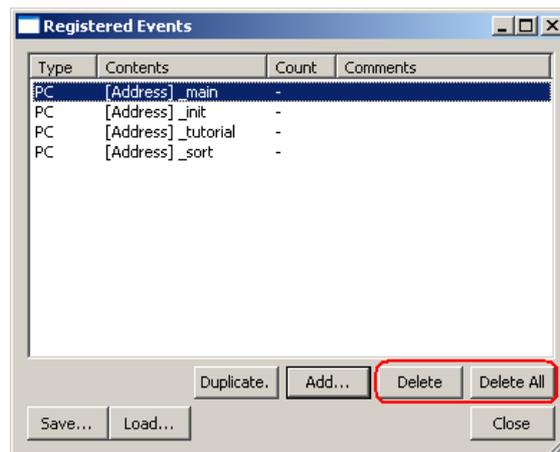


图 5.20 注册事件的删除

5.6.6 事件的随时输入 / 再次使用

给各功能设定事件的方法有 2 种。

第 1 种方法是在各设定对话框中随时建立事件。

第 2 种方法是从 [Registered Events]（已注册事件）对话框中选择 1 个要使用的条件并且拖放到要设定的条件区域。

前者称为随时输入而后者称为再次使用（Re-use）。

随时输入：

这是只能使用一次的条件。

不注册而使用已建立的事件。

使用（更改、删除）事件后不留下设定。

通过双击 [Editor]（编辑器）窗口的列等方法而简单地建成的事件属于随时输入。

再次使用：

通过将注册在 [Registered Events]（已注册事件）对话框中的事件拖放到各功能的条件设定区域，能再次使用。



图 5.21 事件再次使用的概要图

(a) 拖放到多个功能对话框。

能将 [Registered Events]（已注册事件）对话框中的 1 个事件拖放到多个功能对话框。

如果在拖放后更改事件内容，就不反映到 [Registered Events]（已注册事件）对话框。

(b) 重复注册到 [Registered Events]（已注册事件）对话框。

相同设定内容的事件能重复注册。

5.6.7 事件的应用

要在建立事件后将设定置为有效时，单击 [Apply]（应用）按钮。

在单击 [Apply]（应用）按钮前，设定的内容不会变为有效。

如果在 [On-Chip Break]（On-Chip 暂停）对话框、[Trace conditions]（跟踪条件）对话框或者 [Performance]（性能）对话框中标题栏的最后有“*”，就表示正在编辑设定。不能在编辑中途从 [Editor]（编辑器）窗口的 [Event]（事件）列或者命令行更改设定。

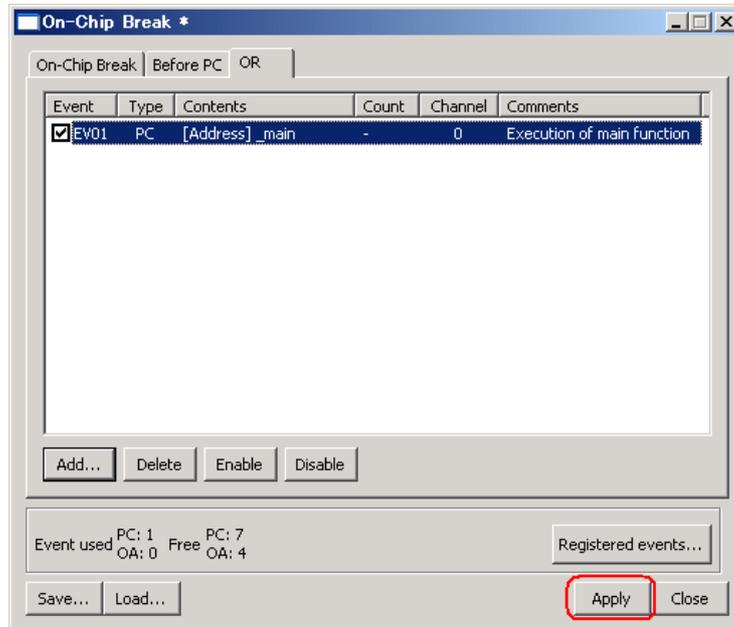


图 5.22 设定内容的应用

5.7 On-Chip Break (On-chip 暂停) 功能

On-chip 暂停是在检测到设定的事件时停止用户程序执行的功能。

5.7.1 On-Chip Breakpoint (On-chip 断点) 的设定

(a) 设定 On-Chip Breakpoint (On-chip 断点)。

On-chip 断点能设定执行前 PC、事件组合 (OR、AND (累积)、顺序) 和其他条件。

能同时设定执行前 PC、事件组合 (OR、AND (累积)、顺序) 和其他条件, 也能限定其中某一个。

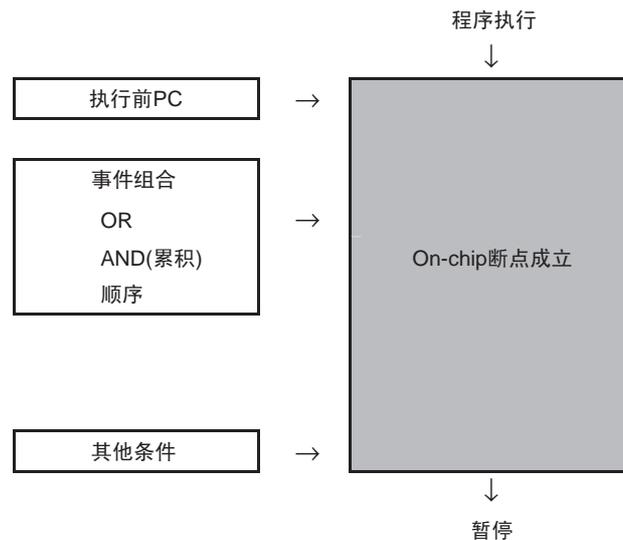


图 5.23 On-Chip Break (On-chip 暂停) 概要

要设定 On-chip 断点时, 请选择 [View -> Event -> On-Chip Break] (视图 -> 事件 -> On-chip 暂停), 打开 [On-Chip Break] (On-Chip 暂停) 对话框。

(b) 设定 Pre-PC Break (执行前 PC 暂停)。

执行前 PC 暂停总是有效。

要将执行前 PC 暂停设定为无效时, 删除已注册的事件或者将注册的事件设定为无效。

执行前 PC 暂停在执行事件指定地址的指令前进行暂停。

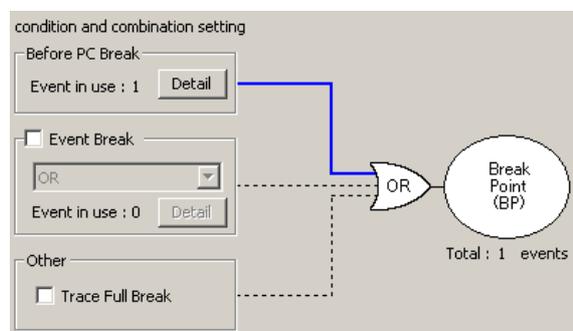


图 5.24 [On-Chip Break] (On-Chip 暂停) 对话框 _Pre-PC Break (执行前 PC 暂停)

如果单击 [Before PC Break] (执行前 PC 暂停) 栏的 [Detail] (详细) 按钮, 就显示事件设定栏 ([Before PC] (执行前 PC) 选项卡)。

如果单击 [Add] (添加) 按钮, 就显示 [Event] (事件) 对话框, 并能指定执行前 PC 暂停的事件 (执行地址条件)。(也能通过标签指定)

(执行前 PC 暂停的事件类型只能是“执行地址”。)

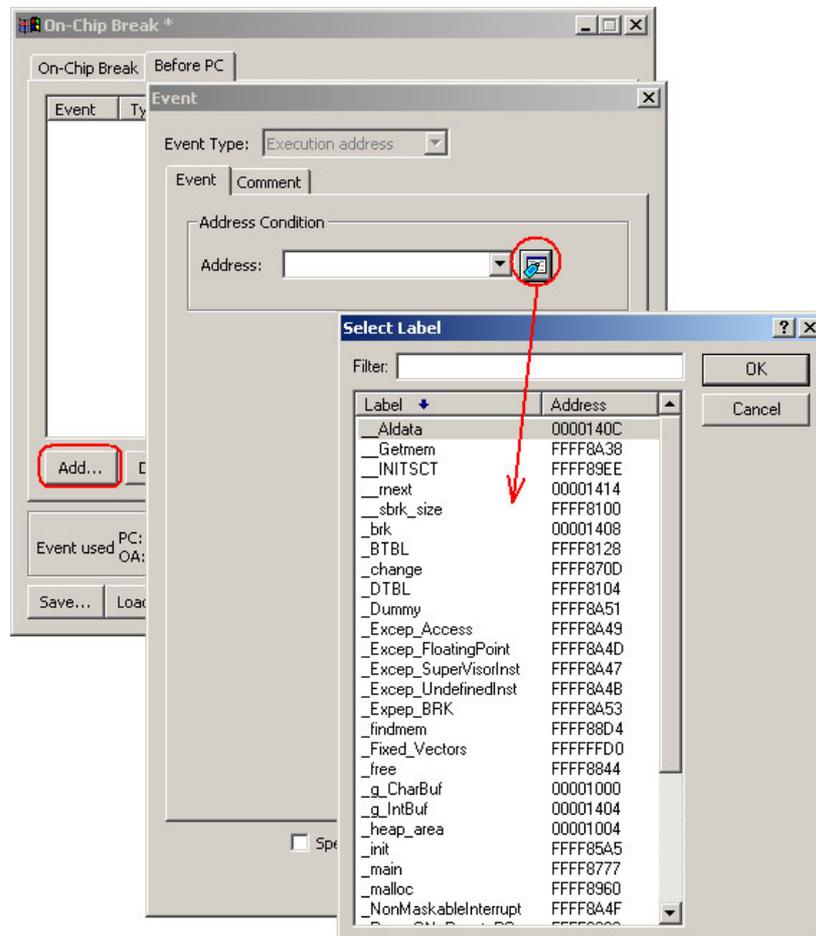


图 5.25 执行前 PC 暂停的设定

如果标题栏的最后有“*”，就表示设定还没有变为有效。请在设定事件后单击 [Apply] (应用) 按钮, 将设定为有效 (当设定变为有效时, “*” 就消失)。

(c) 设定 Event Break（事件暂停）。

能选择 [OR]、[AND(simultaneous)]（AND（累积））或者 [Sequential]（顺序）。

要将事件暂停设定为有效时，选定 [Event Break]（事件暂停）左边的复选框。要将事件暂停设定为无效时，撤销 [Event Break]（事件暂停）左边的选定。

默认值为事件暂停无效。

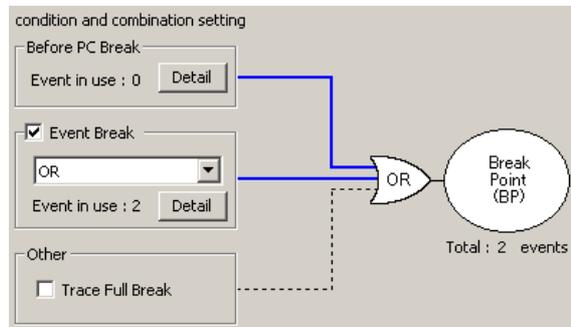


图 5.26 [On-Chip Break]（On-Chip 暂停）对话框 _Event Break（事件暂停）

表 5.5 事件暂停项目一览表

	种类	内容
1	OR	当设定事件中的任意一个事件成立时，断点成立。
2	AND (累积)	与时间轴无关，当设定的全部事件都成立时，断点成立。
3	顺序	7 个阶段（正向）+ 复位点 以指定的顺序，当设定的事件成立时，断点成立。

能用 Ctrl + Del 键删除各条件的列表中显示的事件。

如果单击 [Event Break]（事件暂停）栏的 [Detail]（详细）按钮，就显示 [Event Break]（事件暂停）栏（选择 OR 时为 [OR] 选项卡）。

如果单击 [Add]（添加）按钮，就显示 [Event]（事件）对话框，能指定事件暂停的事件（执行地址条件或者数据存取条件）。（也能通过标签指定地址条件）

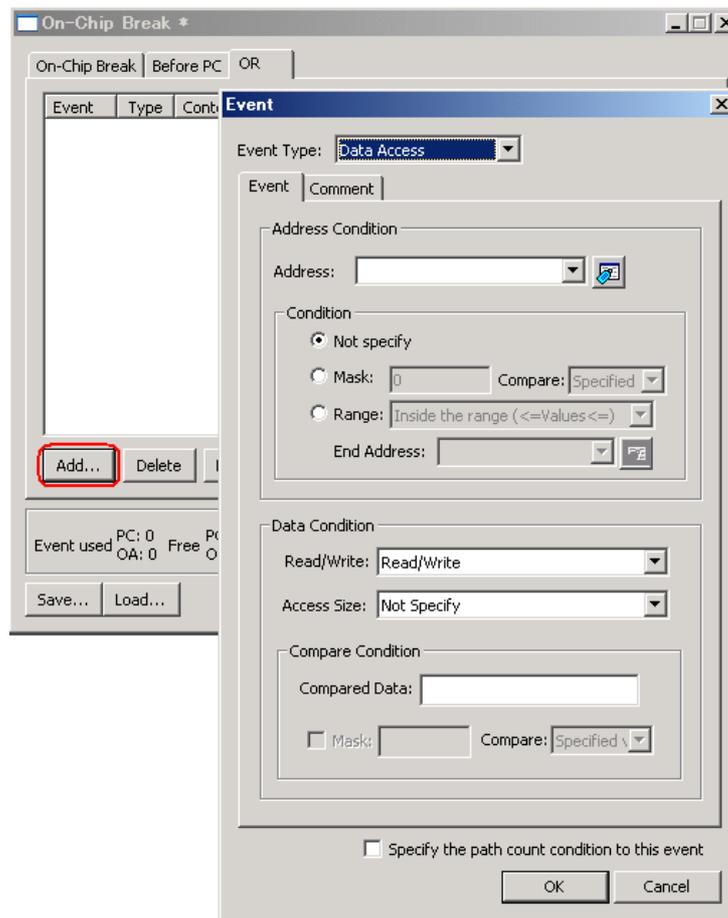


图 5.27 事件暂停的设定

如果标题栏的最后有“*”，就表示设定还没有变为有效。请在设定事件后单击 [Apply]（应用）按钮，将设定置为有效（当设定变为有效时，“*”就消失）。

(d) 设定其他条件。

设定是否将以下其他条件的检测用作断点。

- 跟踪满暂停

5.7.2 On-chip 暂停设定内容的保存 / 加载

(a) 保存 On-chip 暂停的设定内容。

请单击 [On-Chip Break] (On-Chip 暂停) 对话框的 [Save] (保存) 按钮。

打开 [Save As] (另存为) 对话框。

请指定要保存的文件名。文件的扩展名为 “.hev”，省略时自动添加扩展名 “.hev”。

(b) 加载 On-chip 暂停的设定内容。

请单击 [On-Chip Break] (On-Chip 暂停) 对话框的 [Load] (加载) 按钮。

打开 [Open] (打开) 对话框。

请指定要加载的文件名。

在加载时，取消加载前的 On-chip 暂停设定，重新设定已加载的内容。

如果单击 [On-Chip Break] (On-Chip 暂停) 对话框的 [Apply] (应用) 按钮，就确定已加载的 On-chip 暂停设定。

5.8 跟踪功能

E1/E20 仿真调试程序能使用如表 5.6 所示的内部跟踪和外部跟踪共 2 种跟踪功能。

表 5.6 跟踪功能一览表

功能	内部跟踪	外部跟踪输出
转移跟踪功能	能	能
数据跟踪功能	能	能

请注意：能使用外部跟踪输出功能的产品如下：

表 5.7 产品名称和外部跟踪输出功能的对应表

产品名称（产品型号）	外部跟踪输出功能的使用
E1（R0E000010KCE00）	不能使用
E20（R0E000200KCT00）	能使用

在 [Trace]（跟踪）窗口的 [Trace conditions]（跟踪条件）对话框中进行内部跟踪和外部跟踪的输出设定。

(1) 内部跟踪功能

通过使用单片机内置的跟踪缓冲器来实现内部跟踪功能。

通过事件功能（执行 PC 事件、操作数存取事件），能对 CPU 总线进行转移跟踪和数据跟踪，并且能获取 256 个转移或者周期的转移源 / 转移目标的地址信息和数据存取信息。能在 [Trace]（跟踪）窗口中将获取的跟踪结果进行总线 / 反汇编 / C 源级的显示。

(2) 外部跟踪输出功能

此功能使用单片机从跟踪引脚输出的跟踪信息。E20 通过记录单片机的跟踪引脚输出的信息来实现大容量跟踪功能，在将 E20 仿真器连接 38 引脚连接器时能使用此功能。跟踪内容与内部跟踪功能的情况相同，能进行转移跟踪和数据跟踪，并且能进行约 2M 个转移或者周期的大容量跟踪。

外部跟踪输出功能有跟踪输出优先模式（完全跟踪模式）和 CPU 执行优先模式（实时跟踪模式）共 2 种模式。

(a) 跟踪输出优先模式（完全跟踪模式）

这是能将全部跟踪信息完整地输出到外部的模式。

在跟踪输出优先模式中，为了将跟踪信息全部输出，如果超过外部输出量的跟踪信息被输入到跟踪电路时，就强制停止 CPU 和总线的运行而不让更多的跟踪信息输入。在将单片机内部处理中的全部跟踪信息输出到外部后再次允许跟踪输入。

因此，在跟踪输出优先模式中，可能会失去实时性。

(b) CPU 执行优先模式（实时跟踪模式）

这是将跟踪信息实时输出到外部的模式。

在 CPU 执行优先模式中，为了优先实时性，当内部输入量超过外部输出量时，跟踪信息可能会有欠缺（丢失）。

5.8.1 跟踪信息的查看

跟踪是指在用户程序执行过程中获取各周期的总线信息并且将该信息保持在跟踪存储器内的功能。能通过使用跟踪，追踪应用程序的执行流程，调查发生问题的关键。

5.8.2 跟踪信息的获取

在不设定跟踪信息获取条件的情况下，本调试程序按默认值无条件地跟踪并且获取全部转移信息（跟踪获取模式 = fill until stop（填充直至停止））

在“fill until stop”（填充直至停止）模式中，在用户程序开始执行的同时开始获取跟踪信息，在用户程序停止执行时停止获取跟踪。

在 [Trace]（跟踪）窗口中显示获取的跟踪信息。

Cycle	Label	Address	Source	Destination	Address	Data	Size	R/W	BUS Master	Type	ECM	Branch Type	Channel	TimeStamp (Count)
-00000040		FFFF87AB	-----	FFFF87AB					CPU	DESTINATION	11	-----	-	
-00000039														
-00000038	_rand	FFFF89D2	-----	FFFF89D2					CPU	DESTINATION	11	-----	-	
-00000037		FFFF87AB	-----	FFFF87AB					CPU	DESTINATION	11	-----	-	
-00000036														
-00000035	_rand	FFFF89D2	-----	FFFF89D2					CPU	DESTINATION	11	-----	-	
-00000034		FFFF87AB	-----	FFFF87AB					CPU	DESTINATION	11	-----	-	
-00000033														
-00000032	_rand	FFFF89D2	-----	FFFF89D2					CPU	DESTINATION	11	-----	-	
-00000031		FFFF87AB	-----	FFFF87AB					CPU	DESTINATION	11	-----	-	
-00000030														
-00000029	_rand	FFFF89D2	-----	FFFF89D2					CPU	DESTINATION	11	-----	-	
-00000028		FFFF87AB	-----	FFFF87AB					CPU	DESTINATION	11	-----	-	
-00000027														
-00000026	_rand	FFFF89D2	-----	FFFF89D2					CPU	DESTINATION	11	-----	-	

图 5.28 [Trace]（跟踪）窗口

总线显示时的显示项目一览表如下所示。

表 5.8 显示项目一览表

列	内容
Cycle	这是跟踪存储器内的周期号。最后获取的周期号为 0，前面的周期以逆向顺序分配为 -1、-2、……。
Label	显示地址的对应标签，只在设定了标签的情况下进行显示。
Address	显示地址值，当地址不存在时显示空白。
Source Address	显示转移源的地址值，当没有相应的值时显示“—”。
Destination Address	显示转移目标的地址值，当没有相应的值时显示“—”。
Data	用 16 进制显示字节、字和长字的数据。 没有被存取的数据总线显示“—”或者空白。
Size	显示存取大小。 BYTE：字节 WORD：字 LONG：长字 当没有相应的值时显示“—”。
R/W	显示数据总线的状态。 W：写存取（用红褐色字符显示总线写数据的行。） R：读存取（用绿色字符显示总线读数据的行。） 用黑色显示总线读写数据以外的行。 当没有相应的值时显示“—”。
BUS Master	显示发生跟踪事件的总线主控。 CPU：CPU 存取 当没有相应的值时显示“—”。
Type	显示跟踪信息的种类。 BCND：条件转移 BRANCH：转移指令的转移源 DESTINATION：转移指令的转移目标 BRANCH/DESTINATION：转移指令的转移源 / 转移目标 MEMORY：操作数存取 STANDBY：待机信息 LOST：表示丢失了跟踪信息。 当没有相应的值时显示“—”。
BCN	显示条件转移成立信息。当条件转移成立时显示 1，不成立时显示 0。 1 行最多显示 15 个转移信息。 当没有相应的值时显示“—”。

表 5.8 显示项目一览表（续）

列	内容
Branch Type	这是保留区，显示“—”。
Channel	显示是在哪个事件成立时输出了操作数存取。 （无）：事件不成立的操作数存取 0：Channel0 事件成立的操作数存取 1：Channel1 事件成立的操作数存取 2：Channel2 事件成立的操作数存取 3：Channel3 事件成立的操作数存取 当没有相应的值时显示“—”。
TimeStamp (Count)	显示时戳。 用 10 进制显示 0 ~ 19 位，不能检测上溢。 用以下频率对跟踪的 TimeStamp 值进行计数： EXTAL*8 <= 100 MHz 的情况 TimeStamp 的计数频率 = EXTAL*8 EXTAL*8 > 100MHz 的情况 TimeStamp 的计数频率 = EXTAL*4

能将 [Trace]（跟踪）窗口中不需要的列设定为不显示。

要将列设定为不显示时，请在页眉列上单击鼠标右键，从显示的弹出式菜单中选择不要显示的列。

5.8.3 跟踪结果的显示

在 [Trace]（跟踪）窗口中参照跟踪结果。

能用以下的显示格式进行跟踪结果的显示。

能在 [Trace]（跟踪）窗口的弹出式菜单的 [Display Mode]（显示模式）中转换显示格式。

能对跟踪结果进行总线显示、反汇编显示、源显示及其混合显示。

(a) 显示总线

能参照各跟踪周期的总线信息。显示内容由使用的单片机和仿真器系统决定。除了总线信息以外，能混合显示反汇编信息、源行信息和数据存取信息。

要显示总线时，从弹出式菜单中选择 [Display -> BUS]（显示 -> BUS）。

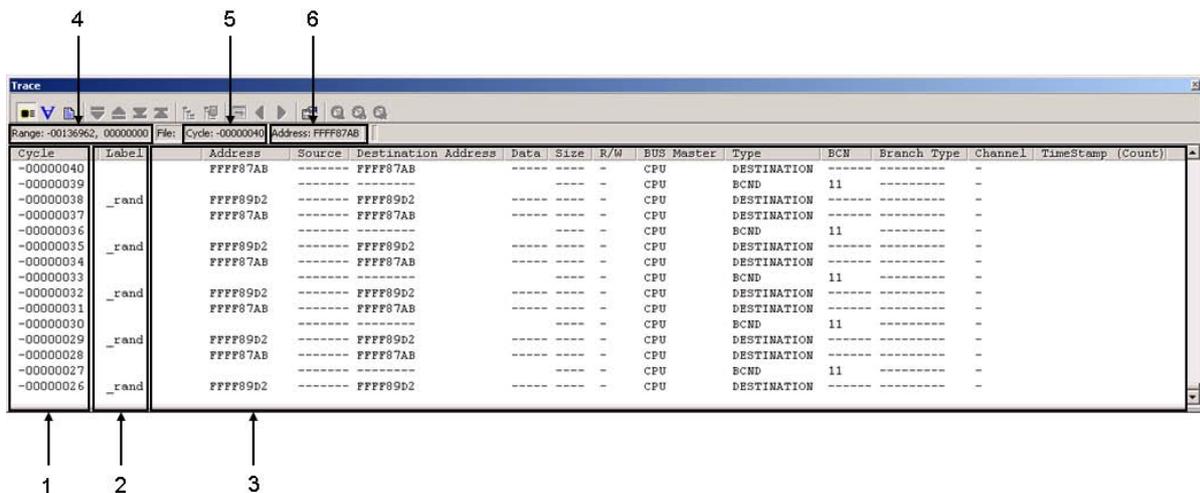


图 5.29 显示总线时的 [Trace]（跟踪）窗口

1. 周期显示区域：显示跟踪周期。如果双击此区域，就显示更改显示周期的对话框。
2. 标签显示区域：显示地址总线信息的对应标签。
3. 总线信息显示区域：显示跟踪总线信息。
4. 已获取的跟踪测量结果范围：显示当前获取的跟踪测量结果范围。
5. 起始行的周期：显示起始行的周期。
6. 起始行的地址：显示起始行的地址。

(b) 显示反汇编

能参照已执行的机器指令。除了反汇编信息以外，能混合显示源行信息和数据存取信息。

要显示反汇编时，从弹出式菜单中选择 [Display -> DIS]（显示 -> DIS）。

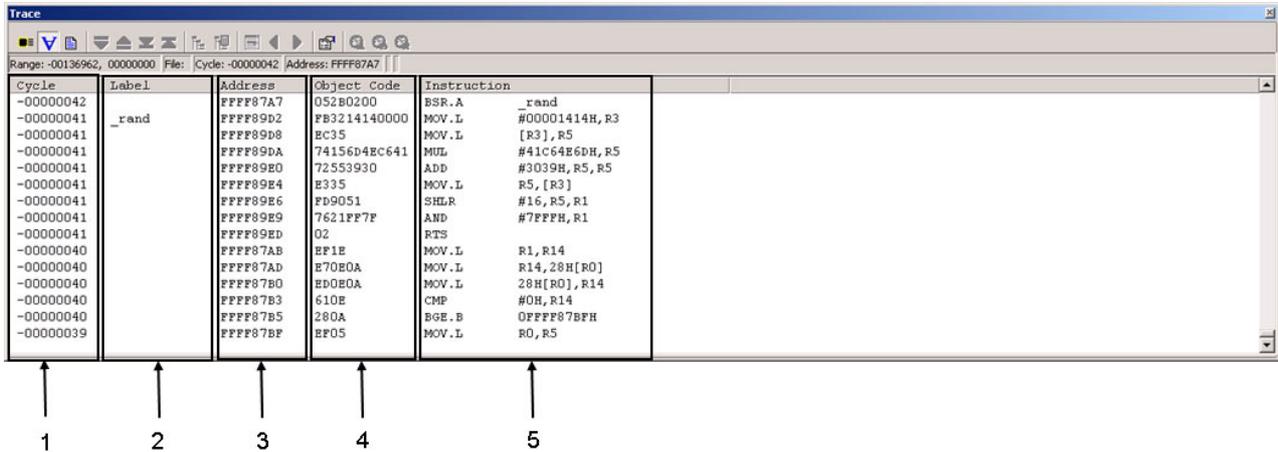


图 5.30 显示反汇编时的 [Trace]（跟踪）窗口

1. 周期显示区域：显示跟踪周期。如果双击此区域，就显示更改显示周期的对话框。
2. 标签显示区域：显示指令地址的对应标签。
3. 地址显示区域：显示指令的对应地址。如果双击此区域，就显示检索地址的对话框。
4. 目标码显示区域：显示指令的目标码。
5. 指令显示区域：显示指令。

(c) 显示源

能参照源程序的执行路径。

能从当前跟踪周期开始，在跟踪数据内通过正向或者反向的源步进（source step）来确认执行路径。

在跟踪测量结束时显示测量结果。如果重新开始跟踪测量，就清除窗口的显示。

要显示源时，从弹出式菜单中选择 [Display -> SRC]（显示 -> SRC）。

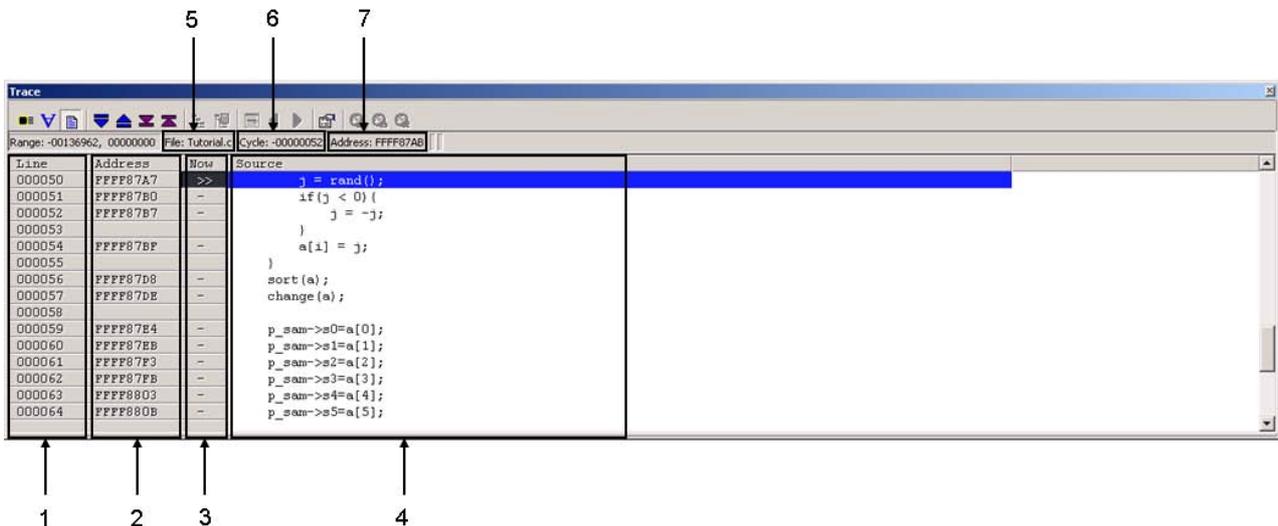


图 5.31 显示源时的 [Trace]（跟踪）窗口

1. 行号显示区域：显示文件的行号信息。
2. 地址显示区域：显示源行的对应地址。如果双击此区域，就显示检索地址的对话框。
3. 参照周期显示区域：对于当前参照的周期，显示“>>”。另外，当存在源行的对应地址时，显示“—”。
4. 源显示区域：显示源文件。
5. 文件名：显示当前的源文件名。
6. 参照周期：显示当前参照的周期。
7. 参照地址：显示当前参照周期的对应地址。

(d) 混合显示

能混合显示总线、反汇编和源。

如果在选择弹出式菜单的 [Display -> BUS] (显示 -> BUS) 后选择 [Display -> DIS] (显示 -> DIS)，就能混合显示总线和反汇编。

用同样的方法能进行总线和源、反汇编和源以及总线、反汇编和源的混合显示。

要在混合显示总线和反汇编后返回到只显示总线时，再次选择弹出式菜单的 [Display -> DIS] (显示 -> DIS)，变为总线显示。

Cycle	Label	Address	Source	Destination	Address	Data	Size	R/W	BUS Master	Type	BCM	Branch Type	Channel	TimeStamp (Count)
-0000053	_rand	FFFF89D2		FFFF89D2					CPU	DESTINATION				
		FFFF89D2	MOV.L	#00001414H, R3										
		FFFF89D8	MOV.L	[R3], R5										
		FFFF89DA	MUL	#41C64B6DH, R5										
		FFFF89E0	ADD	#3039H, R5, R5										
		FFFF89E4	MOV.L	R5, [R3]										
		FFFF89E6	SHLR	#16, R5, R1										
		FFFF89E9	AND	#7FFFH, R1										
		FFFF89ED	RTS											
-0000052		FFFF87AB		FFFF87AB					CPU	DESTINATION				
		FFFF87AB	MOV.L	R1, R14										
		FFFF87AD	MOV.L	R14, 28H[R0]										
		FFFF87B0	MOV.L	28H[R0], R14										
		FFFF87B3	CMP	#0H, R14										
		FFFF87B5	BGE.B	0FFFF87BPH										

图 5.32 混合显示总线和反汇编时的 [Trace] (跟踪) 窗口

5.8.4 选项菜单

如果在窗口中单击鼠标右键，就显示以下的弹出式菜单。这些菜单的主要功能也分配到工具栏的按钮。

表 5.9 选项菜单

菜单名	功能	
[Display Mode] (显示模式)	BUS	显示总线 (BUS) 信息。
	DIS	显示反汇编 (DIS) 信息。
	SRC	显示源 (SRC) 信息。
[Trace] (跟踪)	[Step Forward] (正向步进)	从当前周期开始正向步进。 (只在源模式时有效)
	[Step Backward] (反向步进)	从当前周期开始反向步进。 (只在源模式时有效)
	[Come Forward] (正向执行到指定的行)	从当前周期开始正向执行到指定的光标位置。 (只在源模式时有效)
	[Come Backward] (反向执行到指定的行)	从当前周期开始反向执行到指定的光标位置。 (只在源模式时有效)
	[Stop] (停止)	只在用户程序执行过程中获取跟踪信息时有效。中止跟踪并且更新跟踪的显示。 不能在内部跟踪模式中使用。
	[Restart] (重新开始)	重新开始跟踪。只在用户程序执行过程中暂停跟踪时有效。 不能在内部跟踪模式中使用。
[Find] (查找)	[Find...] (查找 ...)	显示 [Find] (查找) 对话框。检索特定的跟踪信息。
	[Find Previous] (查找上一个)	检索与 [Find] (查找) 中指定的模式相同的前一个周期。
	[Find Next] (查找下一个)	检索与 [Find] (查找) 中指定的模式相同的下一个周期。
[Auto Filter] (自动滤波器)	切换自动滤波器功能的有效和无效。	
[Acquisition...] (获取 ...)	显示设定跟踪条件的 [Trace conditions] (跟踪条件) 对话框。	
[Layout...] (布局 ...)	显示设定显示列的 [Layout] (布局) 对话框。	
[File] (文件)	[Edit Source] (编辑源)	在 [Editor] (编辑器) 窗口中显示用源模式显示的源文件。
	[Display Source] (显示源)	在源模式中，选择要显示的源文件。打开文件选择对话框。
	[Save] (保存)	将跟踪数据保存到文件。能选择 2 进制或者文本格式的保存。当是 2 进制格式时，保存全部周期。 当是文本格式时，保存窗口中显示的内容，也能指定周期范围进行保存。 另外，能用制表符分隔格式进行保存 (如混合模式中的总线、反汇编和源混合的情况，总线的显示用制表符分开，反汇编和源的显示用空格分开)。
	[Load] (加载)	加载被保存的跟踪数据。 只能加载用 2 进制格式保存的文件。
[Toolbar display] (工具栏显示)	切换工具栏的显示和不显示。	
[Customize toolbar] (自定义工具栏)	自定义工具栏。	
[Allow Docking] (允许入坞)	将窗口入坞。	
[Hide] (隐藏)	将窗口设定为隐藏。	

5.8.5 跟踪信息获取条件的设定

跟踪缓冲器的大小有限，当缓冲器满时从最早的跟踪数据开始按顺序盖写新的数据。
能通过设定跟踪信息的获取条件，只获取有用的跟踪信息，有效地利用跟踪缓冲器。

从弹出式菜单中选择 [Acquisition]（获取），在显示的 [Trace conditions]（跟踪条件）对话框中设定跟踪信息的获取条件。

(a) 设定跟踪获取条件。

请先设定跟踪获取条件。

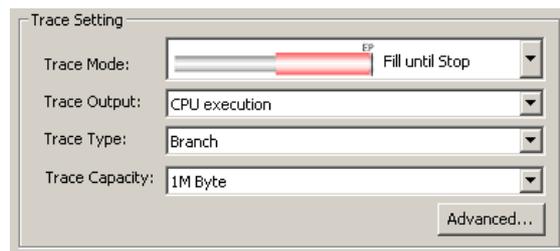


图 5.33 [Trace conditions]（跟踪条件）对话框

跟踪获取条件的设定有以下 4 种：

表 5.10 跟踪获取条件的设定一览表

	参数	内容
①	[Trace Mode] (跟踪模式)	<ul style="list-style-type: none"> Fill until stop (填充直至停止) Fill until full (填充至满)
		设定跟踪获取模式。 Fill until stop 在停止程序或者跟踪获取停止条件成立前连续获取跟踪信息。 Fill until full 在程序开始执行或者跟踪获取开始条件成立后跟踪缓冲器满时，停止获取跟踪信息。
②	[Trace Output] (跟踪输出)	<ul style="list-style-type: none"> CPU execution (CPU 执行) Trace output (跟踪输出) Do not output (不输出)
		设定跟踪获取数据的输出模式。 CPU 执行优先 CPU 的执行优先于跟踪信息的输出。输出的跟踪信息可能会丢失。 跟踪优先 跟踪信息的输出优先于 CPU 的执行。 不输出 因为输出跟踪信息而停止 CPU 的执行，所以失去实时性。不输出跟踪信息，只使用单片机的内部缓冲器。
③	[Trace Type] (跟踪类型)	<ul style="list-style-type: none"> Branch (转移) Branch + Data (转移 + 数据) Data (数据)
		设定跟踪获取数据的类型。 转移 在程序执行过程中，跟踪发生转移处理的转移源和转移目标的地址信息。 转移 + 数据 进行转移跟踪和数据存取跟踪。 数据 在程序执行过程中，跟踪成立的事件数据信息。
④	[Trace Capacity] (跟踪容量)	1M、2M、4M、8M、16M、32M Byte
		设定将跟踪获取数据保存到仿真器的缓冲器容量。 能设定 1M、2M、4M、8M、16M、32M Byte。

如果单击 [Advanced...]（高级 ...）按钮，就能设定 [Trace]（跟踪）窗口的显示内容。
请将要显示的信息复选框设定为有效。
不能使用 [Limit the data access range]（限制数据存取范围）。

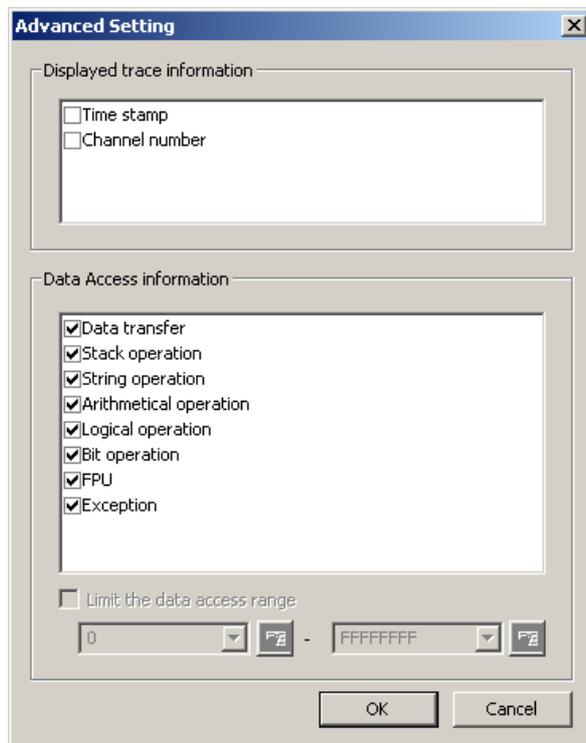


图 5.34 [Advanced Settings]（高级设置）对话框

(b) 设定跟踪条件。

能分别设定跟踪获取的开始条件、停止条件和抽出条件。当不设定跟踪条件时，跟踪从程序的执行到停止的内容。

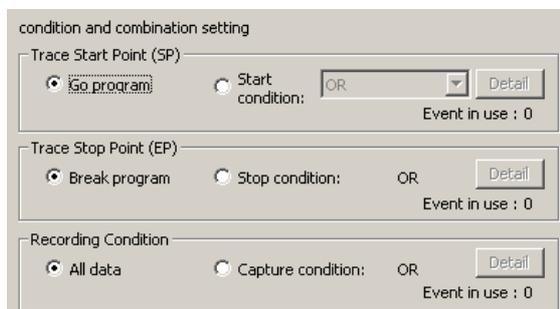


图 5.35 [Trace conditions] (跟踪条件) 对话框

跟踪条件的设定有以下 3 种：

表 5.11 跟踪条件的设定一览表

设定项目	内容
① Trace Start Point (SP) (跟踪开始点 (SP))	能通过事件组合来指定跟踪获取的开始触发。 当不设定跟踪获取开始条件时，就在程序开始执行的同时开始获取跟踪信息。 OR 当设定事件中的任意一个事件成立时，跟踪获取开始条件成立。 AND (累积) 与时间轴无关，当设定的全部事件都成立时，跟踪获取开始条件成立。 顺序 7 个阶段 (正向) + 复位点 以指定的顺序，当设定的事件成立时，跟踪获取开始条件成立。
② Trace Stop Point (EP) (跟踪停止条件 (EP))	能通过事件 (只限于 OR) 指定跟踪获取的停止触发。 当不设定跟踪获取停止条件时，就在程序停止执行的同时停止获取跟踪信息。
③ Recording Condition (记录条件)	能通过事件 (只限于 OR) 指定跟踪抽出的触发。

通过跟踪条件的设定和跟踪获取条件设定的获取模式 (“fill until stop” 或者 “fill until full”) 的组合，改变跟踪获取的内容。各设定时的不同跟踪获取内容如下所示。

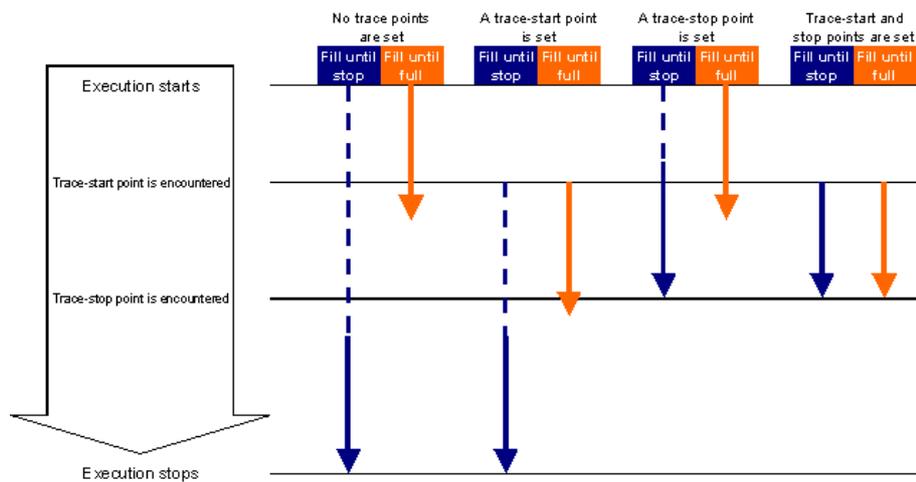


图 5.36 跟踪获取条件设定和跟踪条件设定的不同跟踪获取内容

(c) 保存或者加载跟踪设定内容。

1. 保存跟踪设定内容。

请单击 [Trace conditions] (跟踪条件) 对话框的 [Save] (保存) 按钮。

打开 [Save As] (另存为) 对话框。

请指定要保存的文件名。文件的扩展名为 “.tev”，省略时自动添加扩展名 “.tev”。

2. 加载跟踪设定内容。

请单击 [Trace conditions] (跟踪条件) 对话框的 [Load] (加载) 按钮。

打开 [Open] (打开) 对话框。

请指定要加载的文件名。

在加载时，取消加载前的跟踪设定，重新设定已加载的内容。

如果单击 [Trace conditions] (跟踪条件) 对话框的 [Apply] (应用) 按钮，就确定已加载的跟踪设定。

5.8.6 跟踪信息的文件保存

要将跟踪信息保存到文件时，从弹出式菜单中选择 [File -> Save]（文件 -> 保存）或者单击 [Save]（保存）工具栏按钮（）。

能用 2 进制格式或者文本格式保存 [Trace]（跟踪）窗口中显示的跟踪信息。

(a) 用 2 进制格式保存

要用 2 进制格式保存时，在通过 [File -> Save]（文件 -> 保存）菜单显示的对话框的 [Save As Type]（文件种类）列表框中选择 [Trace Data File: Memory Image (*.rtt)]。

当用 2 进制格式保存时，保存全部周期。此文件能加载到 [Trace]（跟踪）窗口。

(b) 用文本格式保存

要用文本格式保存时，在通过 [File -> Save]（文件 -> 保存）菜单显示的对话框的 [Save As Type]（文件种类）列表框中选择 [Text Files : Save Only (*.txt)]。

当用文本格式保存时，能指定要保存的周期范围。此文件只能保存而不能加载到 [Trace]（跟踪）窗口。

5.8.7 跟踪信息的文件加载

要从文件加载跟踪信息时，从弹出式菜单中选择 [File -> Load]（文件 -> 加载）或者单击 [Load]（加载）工具栏按钮（）。

请指定用 2 进制格式保存的跟踪信息文件。盖写当前跟踪结果。

当加载用 2 进制格式保存的文件时，请在通过 [Acquisition]（获取）菜单显示的 [Trace conditions]（跟踪条件）对话框中设定为保存时的跟踪模式，然后进行加载。

不能加载用文本格式保存的跟踪信息文件。

5.8.8 跟踪信息获取的暂停

要在用户程序执行过程中暂停获取跟踪信息时，从 [Trace]（跟踪）窗口的弹出式菜单中选择 [Trace -> Stop]（跟踪 -> 停止）或者单击 [Stop]（停止）工具栏按钮（）。

中止跟踪获取并且更新跟踪显示。

此功能用于不停止用户程序的执行而只停止获取跟踪信息来确认跟踪信息的情况。

不能在内部跟踪模式中使用。

5.8.9 跟踪信息获取的重新开始

在用户程序执行过程中暂停获取跟踪信息的情况下，要重新开始获取跟踪信息时，从 [Trace]（跟踪）窗口的弹出式菜单中选择 [Trace -> Restart]（跟踪 -> 重新开始）或者单击 [Restart]（重新开始）工具栏按钮（）。

不能在内部跟踪模式中使用。

5.9 性能测量功能

5.9.1 性能测量

性能测量功能是指对用户指定的最多 2 个区间分别测量总执行时间或者通过次数，并且显示执行时间和周期数的功能。

此功能通过单片机内置的计数器支持 Go-Break 之间或者计数开始事件成立时到计数结束事件成立时的测量，还支持用户程序执行过程中的再测量。能将取指令条件或者数据存取条件设定为计数开始事件和计数结束事件。

测量项目有无测量指定、经过周期、经过周期（管理模式周期）、异常和中断处理周期、异常处理周期、中断处理周期、有效指令发行次数、异常和中断接受次数、异常接受次数、中断接受次数、事件成立次数共 11 项。

支持 32 位计数器 ×2ch（ch0 和 ch1 独立使用）或者 64 位计数器 ×1ch（ch0 和 ch1 连接为 32 位计数器）的 2 种方式。

性能测量功能通过目标单片机的性能测量功能来测量执行时间，因此不妨碍用户程序的执行。

在程序执行过程中不能更改性能测量条件。

5.9.2 性能测量结果的显示

在 [Performance Analysis]（性能分析）窗口中显示测量结果。

要打开 [Performance Analysis]（性能分析）窗口时，请选择 [View -> Performance -> Performance Analysis]（视图 -> 性能 -> 性能分析）或者单击 [Performance Analysis]（性能分析）工具栏按钮（）。打开 [Select Performance Analysis Type]（选择性能分析类型）对话框，请按 [OK] 按钮。



图 5.37 [Select Performance Analysis Type]（选择性能分析类型）对话框

显示 [Performance Analysis]（性能分析）窗口。

请用鼠标右键单击此窗口，从打开的弹出式菜单中选择 [Set]（设置），打开 [Performance]（性能）对话框。能在此对话框中设定要测量的事件和测量条件。

如果在设定条件后按 [OK] 按钮并且执行用户程序，就在执行结束后显示用户设定条件的执行时间和周期数。

能将显示在 [Performance Analysis]（性能分析）窗口中不需要的列设定为不显示。

要将列设定为不显示时，请在页眉列上单击鼠标右键，从显示的弹出式菜单中选择不要显示的列。

要再次显示列时，请从弹出式菜单中再次选择相应的列。

No	Condition	Time (h:m:s.ms.us.ns)	Count (Decimal)
1	Not use	---:---:---.---.---	-
2	Not use	---:---:---.---.---	-

图 5.38 [Performance Analysis] (性能分析) 窗口

显示的内容如下所示。

表 5.12 列和显示内容一览表

列	内容
No	这是在 [Performance] (性能) 对话框中设置的 1 ~ 2 个测量区间号。通道号: 在 32bit×2ch 模式中显示 1 和 2, 在 64bit×1ch 模式中只显示 1。
Condition	显示 [Performance] (性能) 对话框的测量项目的内容。 <ul style="list-style-type: none"> • 不使用时: “Not use” • 经过周期: “Execution cycle” • 经过周期 (管理模式周期): “Execution cycle (Supervisor mode)” • 异常和中断处理周期: “Exception and interrupt cycle” • 异常处理周期: “Exception cycle” • 中断处理周期: “Interrupt cycle” • 有效指令发行次数: “Execution count” • 异常和中断接受次数: “Exception and interrupt count” • 异常接受次数: “Exception count” • 中断接受次数: “Interrupt count” • 事件成立次数: “Event match count”
Time (h:m:s.ms.us.ns)	显示被测量的执行时间的累积时间。 <ul style="list-style-type: none"> • 在测量对象是经过周期、经过周期 (管理模式周期)、异常和中断处理周期、异常处理周期、中断处理周期时, 将 Count 改为时间显示, 使用用户设定的频率和测量的 Count 决定显示值。 • 在测量对象是有效指令发行次数、异常和中断接受次数、异常接受次数、中断接受次数、事件成立次数时, 显示 “---:---:---.---.---
Count (Decimal)	用 10 进制显示所测量的 Count。如果计数器发生上溢, 就显示 “overflow”。

5.9.3 性能测量条件的设定

请在 [Performance Analysis]（性能分析）窗口中选择要设定条件的区间号的行，从弹出式菜单中选择 [Set]（设置）。

显示 [Performance]（性能）对话框。

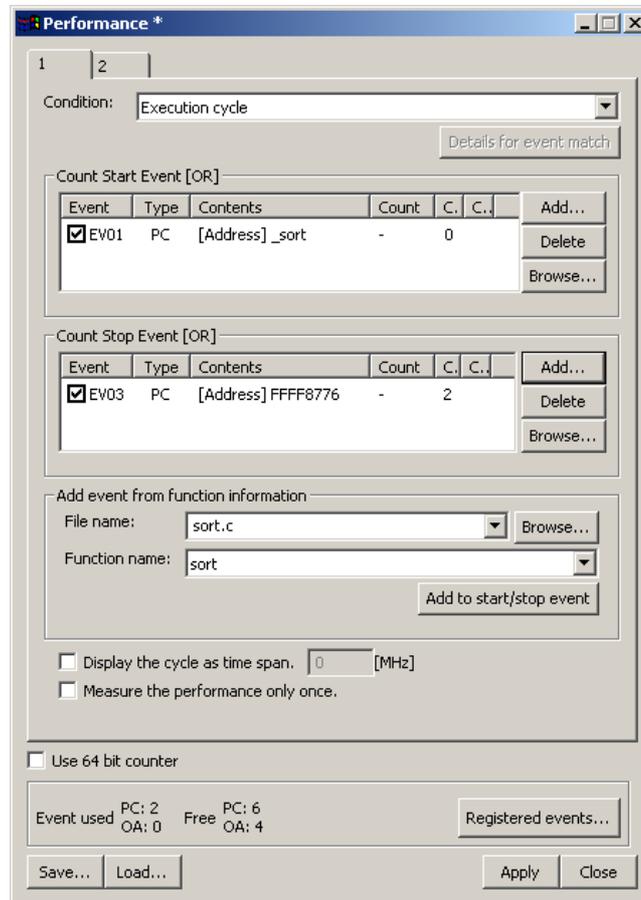


图 5.39 [Performance]（性能）对话框

如果标题栏的最后有“*”，就表示设定还没有变为有效。请在设定事件后单击 [Apply]（应用）按钮，将设定变为有效（当设定变为有效时，“*”就消失）。

(a) 设定测量条件。

对 1 个区间使用 32 位的次数测量计数器，能测量 2 个区间。另外，也能将 1 个区间设定为 64 位的次数测量计数器来使用。如果将其中一个区间设定为 64 位计数器，另一个区间的设定就无效。

1 个区间选择 1 个测量条件。

使用事件进行区间的设定。

表 5.13 测量条件的模式一览表

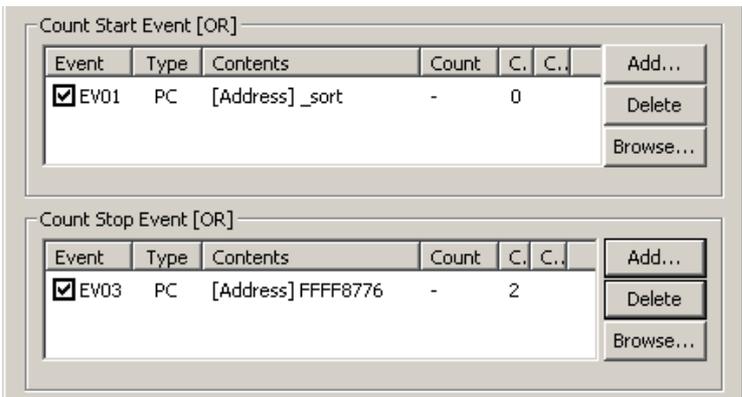
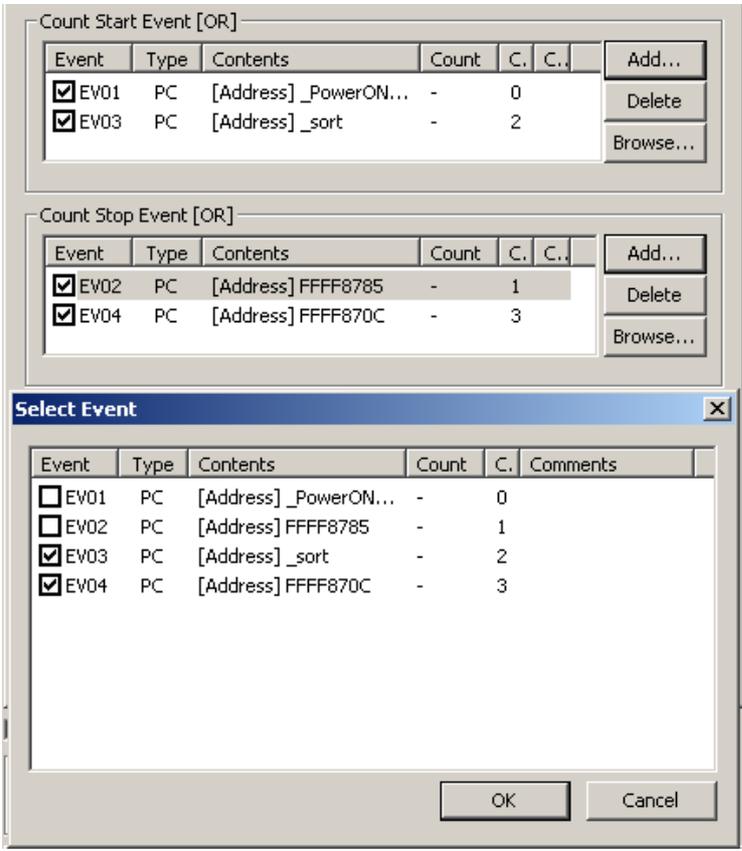
测量项目	测量内容
没有测量指定	不测量。
【测量处理周期】 经过周期、经过周期 (管理模式周期)、 异常和中断处理周期、 异常处理周期、 中断处理周期 【测量处理次数】 有效指令发行次数、 异常和中断接受次数、 异常接受次数、 中断接受次数	从开始事件的成立到结束事件的成立为止进行测量。 <div style="text-align: center;">  </div> <p style="text-align: center;">图 5.40 2 个事件之间</p> <p>测量由<开始事件>和<结束事件>设定的范围的执行周期和处理次数。 测量时间：在<开始事件>成立时开始测量，在<结束事件>成立时中止测量。 处理次数：对设定范围的<开始事件>和<结束事件>之间成立的处理次数进行计数。</p> <p>[计数开始事件]：能设定 1 个事件或者多个事件。如果没有设定计数开始事件，就在用户程序开始执行的同时开始测量。 [计数结束事件]：能设定 1 个事件或者多个事件。如果没有设定计数结束事件，就在用户程序停止执行的同时中止测量。</p>

表 5.13 测量条件的模式一览表（续）

测量项目	测量内容
事件成立次数	<p>测量从开始事件的成立到结束事件的成立为止的事件成立次数。</p>  <p>图 5.41 2 个事件之间的事件成立次数</p> <p>在由<开始事件>和<结束事件>设定的范围内，对指定为事件成立次数测量的事件进行成立次数的计数。</p> <p>[计数开始事件]：能设定 1 个事件或者多个事件。</p> <p>[计数结束事件]：能设定 1 个事件或者多个事件。</p> <p>从由跟踪条件、On-chip 暂停条件和性能条件设定的事件中选择用于事件成立次数测量的事件。</p> <p>能在单击 [Details for event match]（事件成立详细）按钮后显示的 [Select Event]（选择事件）对话框中设定指定为事件成立次数测量的事件。</p>

【注】 要测量函数的执行时间时，必须使用 2 点之间的事件。
 请将函数起始地址的执行设定为开始事件，而将函数出口（return 语句的位置）的执行设定为结束事件。如果有多个出口，就必须将全部出口的执行条件设定为结束事件。

(b) 测量项目

能通过性能测量功能选择的测量项目一览表如表 5.14 所示。

在 [Performance Analysis]（性能分析）窗口的 [Condition]（条件）中显示选择名。

表 5.14 测量项目一览表

选择名	Condition	选择项目
没有测量指定	Not use	不设定性能测量项目。
经过周期	Execution cycle	将指定期间的经过周期数（I ϕ ）设定为测量项目。
经过周期（管理模式周期）	Execution cycle (Supervisor mode)	将管理模式中的经过周期数设定为测量项目。
异常和中断处理周期	Exception and interrupt cycle	将包括异常处理在内的中断处理所需要的周期数设定为测量项目。
异常处理周期	Exception cycle	测量异常处理所需要的周期数。
中断处理周期	Interrupt cycle	测量中断处理所需要的周期数。
有效指令发行次数	Execution count	测量执行结束指令的个数。
异常和中断接受次数	Exception and interrupt count	测量包括异常在内的中断接受次数。
异常接受次数	Exception count	测量发生异常的次数。
中断接受次数	Interrupt count	测量接受中断的次数。
事件成立次数	Event match count	测量设定事件实际成立的次数。

性能测量中的异常处理 / 中断处理包括以下单片机的异常事件。

【异常处理】

- 未定义指令异常
- 特权指令异常
- 浮点异常
- 无条件陷阱（INT 指令和 BRK 指令）

【中断处理】

- 非屏蔽中断
- 中断（可屏蔽）
- 复位

(c) 从函数信息添加事件【注】。

能从源文件指定函数名，设定性能测量的开始事件和结束事件。

在单击 [Browse]（浏览）按钮并且选择源文件后，如果设定函数名并且单击 [Add to start/stop event]（添加开始 / 结束事件）按钮，就将函数的起始地址和结束地址的执行事件设定为开始事件和结束事件。

【注】 在此设定中，将函数的开始地址设定为开始事件并且将结束地址设定为结束事件。当结束地址不是函数的出口（通常是 RTS 指令）时，不能正确测量，此时必须手动将函数的出口地址设定为结束事件。

5.9.4 性能测量的开始

如果执行用户程序，就根据设定的性能测量条件自动开始性能测量。

当暂停用户程序的执行时，就在 [Performance Analysis]（性能分析）窗口显示测量结果。

如果在暂停用户程序的执行后不改变测量条件而重新开始执行，就将测量时间加到前次的测量值。

要从最初开始重新测量时，请在执行前从弹出式菜单中选择 [Clear Data]（清除数据），清除测量结果。

5.9.5 性能测量条件的解除

在 [Performance Analysis]（性能分析）窗口中选择要删除的测量条件，然后从弹出式菜单中选择 [Set]（设置），显示 [Performance]（性能）对话框。

请在 [Performance]（性能）对话框中将要解除的条件设定为“Not use”（不使用）。



图 5.42 性能测量条件的删除

5.9.6 性能测量结果的清除

如果在 [Performance Analysis]（性能分析）窗口中选择要清除的区间后从弹出式菜单中选择 [Clear Data]（清除数据），就清除所选区间的测量结果。

要清除全部测量结果时，从弹出式菜单中选择 [Clear All Data]（清除全部数据）。

5.9.7 性能测量的最大次数

由 32 位计数器测量各区间的性能次数，最多能测量 4,294,967,295 次。当将 1 个区间设定为 64 位计数器时，最多能测量 18,446,744,073,709,551,615 次。

如果在性能测量过程中计数器发生上溢，就在 [性能分析] 窗口显示“overflow”。

要使用 64 位计数器时，请在 [Performance]（性能）对话框中选定 [Use 64-bit counter]（使用 64 位计数器）复选框。

5.9.8 性能条件设定内容的保存和加载

(a) 保存性能条件的设定内容。

请单击 [Performance]（性能）对话框的 [Save]（保存）按钮。

打开 [Save As]（另存为）对话框。

请指定要保存的文件名。文件的扩展名为“.pev”，省略时自动添加扩展名“.pev”。

(b) 加载性能条件的设定内容。

请单击 [Performance]（性能）对话框的 [Load]（加载）按钮。

打开 [Open]（打开）对话框。

请指定要加载的文件名。

在加载时，取消加载前的性能条件设定，重新设定已加载的内容。

如果单击 [Performance]（性能）对话框的 [Apply]（应用）按钮，就确定已加载的跟踪设定。

5.10 实时 RAM 监视功能

5.10.1 RAM 监视功能的使用

要在用户程序执行过程中监视存储器内容时，使用 [RAM Monitor]（RAM 监视）窗口。

RAM 监视功能是指在不妨碍用户程序执行的情况下对分配在监视范围区域内的存储器内容和存取状况进行实时记录和参照的功能。

在 [RAM Monitor]（RAM 监视）窗口中用颜色显示各存取状况（读 / 写 / 丢失）。

因为通过使用调试程序的外部跟踪功能来实现 RAM 监视功能，而 E1 仿真器不能使用外部跟踪功能，所以 E1 仿真器不能使用 RAM 监视功能。

要使用 RAM 监视功能时，请将 [Configuration Properties]（配置属性）对话框的 [Switching function]（切换功能）设定为 [Realtime RAM monitor]（实时 RAM 监视）。

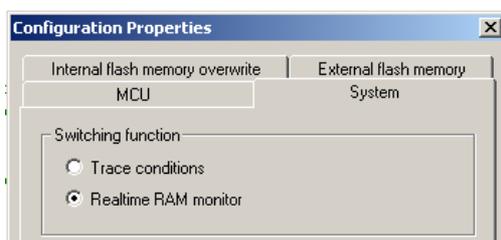


图 5.43 [Configuration Properties]（配置属性）对话框

如果将 [Configuration Properties]（配置属性）对话框的专用功能设定为实时 RAM 监视，就不能使用一部分跟踪功能。使用实时 RAM 监视时的跟踪功能的限制事项如下所示。

表 5.15 跟踪功能的限制事项一览表

功能名	限制事项内容
事件设定	不能使用开始和结束事件，只能使用跟踪抽出事件。
获取模式	不能使用 Fill until full 模式，只能使用 Fill until stop 模式。
数据种类	不能使用转移、转移 + 数据跟踪，只能使用数据跟踪。
跟踪容量	只能使用 1MB。
跟踪显示信息	不能指定时戳和通道号。

5.10.2 RAM 监视范围的分配

具有 4096 字节的 RAM 监视区域。

此 RAM 监视区域能分配任意的连续地址或者以 1024 字节为单位分成 4 块。

设定要监视的区域。因为在此状态下对全部数据的存取进行跟踪输出，所以有可能会丢失。能通过指定跟踪抽出条件进行最佳的 RAM 监视。

请参照“5.10.5 丢失的抑止”。

在 [RAM Monitor Area Setting]（RAM 监视区域的设定）对话框中设定 RAM 监视区域。

要打开 [RAM Monitor Area Setting]（RAM 监视区域的设定）对话框时，请选择 [View -> CPU -> RAM Monitor]（视图 -> CPU -> RAM 监视），显示 [RAM Monitor]（RAM 监视）窗口，然后单击 [RAM Monitor Area Setting]（RAM 监视区域的设定）工具栏按钮。

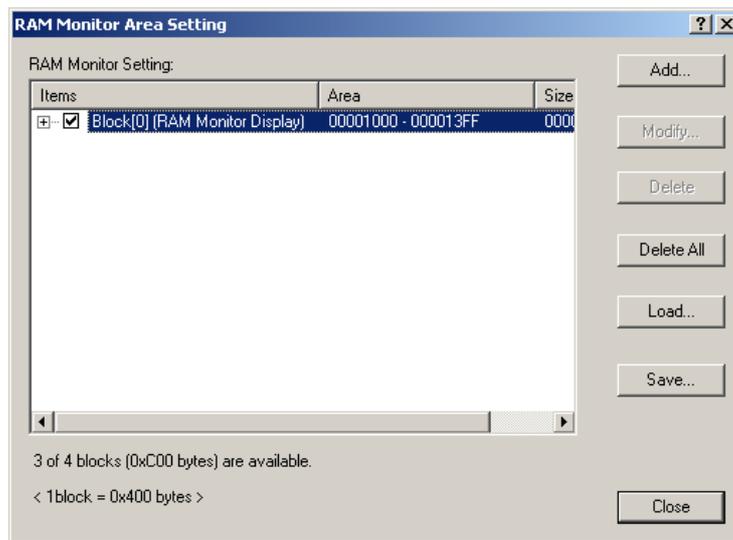


图 5.44 [RAM Monitor Area Setting] (RAM 监视区域的设定) 对话框

(a) 设定 RAM 监视块。

如果单击 [Add] (添加) 按钮, 就打开给 RAM 监视区域分配地址的设定对话框。

RAM 监视块的各块为 1024 字节。即使将分配给 RAM 监视区域的地址指定为小于 1024 字节的范围, 也给 RAM 监视区域分配从起始地址开始的 1024 字节区域。

如果从起始地址开始设定了超过 1024 字节范围的区域, 就自动分配多个 RAM 监视块。

在用户程序执行过程中进行此处理时, 因为发生存储器的读周期, 所以失去实时性。

(b) 更改 RAM 监视块。

请选择被注册的 RAM 监视块, 单击 [Modify] (更改) 按钮。

打开给 RAM 监视区域分配地址的变更对话框。

在用户程序执行过程中进行此处理时, 因为发生存储器的读周期, 所以失去实时性。

(c) 删除 RAM 监视块。

请选择被注册的 RAM 监视块, 单击 [Delete] (删除) 按钮。

要删除被注册的全部 RAM 监视块时, 请单击 [Delete All] (全部删除) 按钮。

(d) 保存 RAM 监视区域的设定内容。

请单击 [RAM Monitor Area Setting] (RAM 监视区域的设定) 对话框的 [Save] (保存) 按钮。

打开 [Save As] (另存为) 对话框。

请指定要保存的文件名。文件的扩展名为 “.rbi”, 省略时自动添加扩展名 “.rbi”。

(e) 加载 RAM 监视区域的设定内容。

请单击 [RAM Monitor Area Setting] (RAM 监视区域的设定) 对话框的 [Load] (加载) 按钮。

打开 [Open] (打开) 对话框。

请指定要加载的文件名。

在加载时, 取消加载前的 RAM 监视区域设定, 重新设定已加载的内容。

如果单击 [RAM Monitor Area Setting] (RAM 监视区域的设定) 对话框的 [Apply] (应用) 按钮, 就确定已加载的 RAM 监视区域设定。

5.10.3 监视内容的显示

要打开 [RAM Monitor] (RAM 监视) 窗口时, 请选择 [View -> CPU -> RAM Monitor] (视图 -> CPU -> RAM 监视) 或者单击 [CPU] 工具栏按钮 ()。

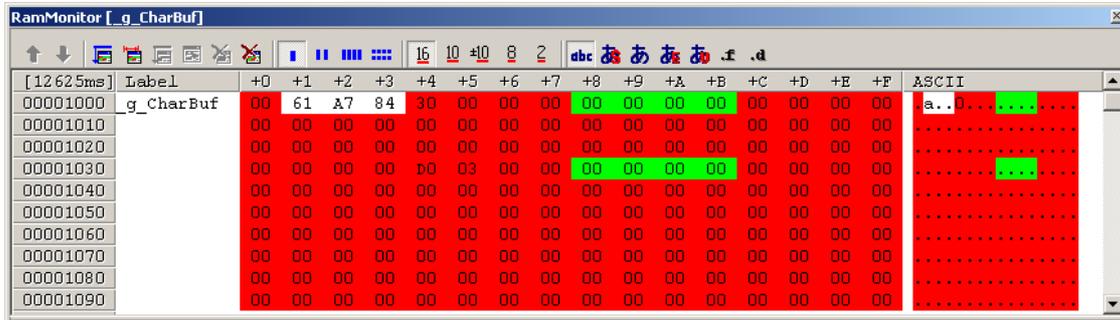


图 5.45 RAM Monitor] (RAM 监视) 窗口

通过存取属性用以下背景色进行显示。

读存取、写存取显示最后的履历。

RAM 监视功能获取并且显示 CPU 总线存取的数据。因此, 不显示用户程序不读写的区域 (例如, 从其他设备读写外部存储器的情况等) 变化以及 DMA/DTC 功能的存取变化。

如果 RAM 监视区域的显示数据的长度单位不是 1 字节, 就以 1 字节为单位的该数据的存储器存取属性可能不同。对于这种在 1 个数据中存取属性不同的情况, 用括号将数据括起来显示, 此时的背景色表示该数据的第 1 字节的存取属性。

(0001) E4BF 84BA 399F 68AD

根据运行状况 (以下的原因), 显示的更新间隔可能长于指定的更新间隔。

- 主机的性能/负载状况
- 通信接口
- 窗口的大小 (存储器显示范围) 和窗口的显示个数

表 5.16 存取属性的背景色一览表

存取属性	背景色
读存取	绿色
写存取	红色
不存取	白色
发生丢失后不存取	灰色

(a) 设定 RAM 监视的更新间隔

如果从 [RAM Monitor] (RAM 监视) 窗口的弹出式菜单中选择 [Update Interval Setting] (更新间隔设置), 就显示 [Update Interval Setting] (更新间隔设置) 对话框。

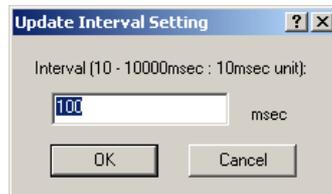


图 5.46 [Update Interval Setting] (更新间隔设置) 对话框

能按窗口指定更新间隔。初始值为 100ms。

(b) 清除 RAM 监视的存取履历。

如果从 [RAM Monitor] (RAM 监视) 窗口的弹出式菜单中选择 [Access Data Clear] (存取数据的清除), 就清除 RAM 监视区域的全部存取履历。

在用户程序执行过程中进行此处理时, 因为发生存储器的读周期, 所以失去实时性。

5.10.4 有关丢失

如果在跟踪输出时发生丢失, RAM 监视窗口中显示的内容就和实际的存储器内容不同。但是, 无法特定发生丢失的数据存取。

如果发生丢失, 就在 [RAM Monitor] (RAM 监视) 窗口中全部用灰色 (表示丢失) 显示存取履历, 保持在丢失后发生的数据存取到再次存取的履历。对于在小于 RAM 监视更新间隔的情况下发生丢失的程序, [RAM Monitor] (RAM 监视) 窗口的存取履历可能总是灰色。这种情况请参照“5.10.5 丢失的抑止”。

5.10.5 丢失的抑止

能通过以下设定来抑止 RAM 监视中存取数据的丢失。

(a) 选择跟踪优先输出。

请选择 [View -> Event -> Trace Conditions] (视图 -> 事件 -> 跟踪条件)，打开 [Trace conditions] (跟踪条件) 对话框，将输出模式设定为 [Trace output] (跟踪输出)。如果设定为跟踪优先输出，就不会在 RAM 监视中发生丢失，但是因为优先跟踪输出，CPU 可能会暂停而失去程序执行的实时性。

(b) 使用跟踪抽出功能。

如果只进行“5.10.2 RAM 监视范围的分配”的设定，因为对全部的存储器存取进行跟踪数据输出，所以即使在监视对象的范围内不发生丢失，也会在其他区域发生丢失。此时，在 [RAM Monitor] (RAM 监视) 窗口中显示“LOST”。

因此，通过跟踪抽出功能将跟踪输出的数据限制在要监视的数据范围以内，能减少丢失的发生。另外，在 [RAM Monitor] (RAM 监视) 中用灰色显示监视范围外的数据范围。

这种情况能维持程序执行的实时性，但是减小了能监视的范围。

- 但是，在频繁进行限制范围的数据存取并且发生丢失的情况下，这种方法也不能抑止丢失。

在通过 [View -> Event -> Trace Conditions] (视图 -> 事件 -> 跟踪条件) 显示的 [Trace conditions] (跟踪条件) 对话框中，能设定跟踪抽出条件。

设定例子如下所示。

设定例子：要监视地址 0x1020 ~ 0x103F 的数据时，通过“5.10.2 RAM 监视范围的分配”的设定方法设定地址 0x1000 ~ 0x13FF，然后通过跟踪抽出功能设定为只输出地址 0x1020 ~ 0x103F。

请在 [Trace conditions] (跟踪条件) 对话框中选择跟踪抽出条件的 [Capture condition] (捕获条件) 单选按钮。

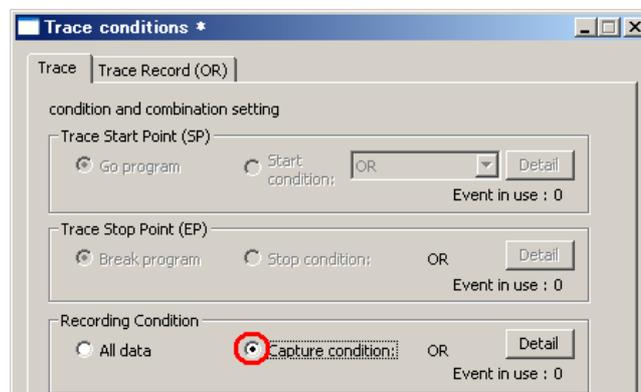


图 5.47 跟踪抽出功能

请选择 [Trace Record (OR)] (跟踪记录) 选项卡，按 [Add] (添加) 按钮。显示 [Event] (事件) 对话框，请在按下图进行设定后按 [OK] 按钮。必须在设定事件后按 [Apply] (应用) 按钮。

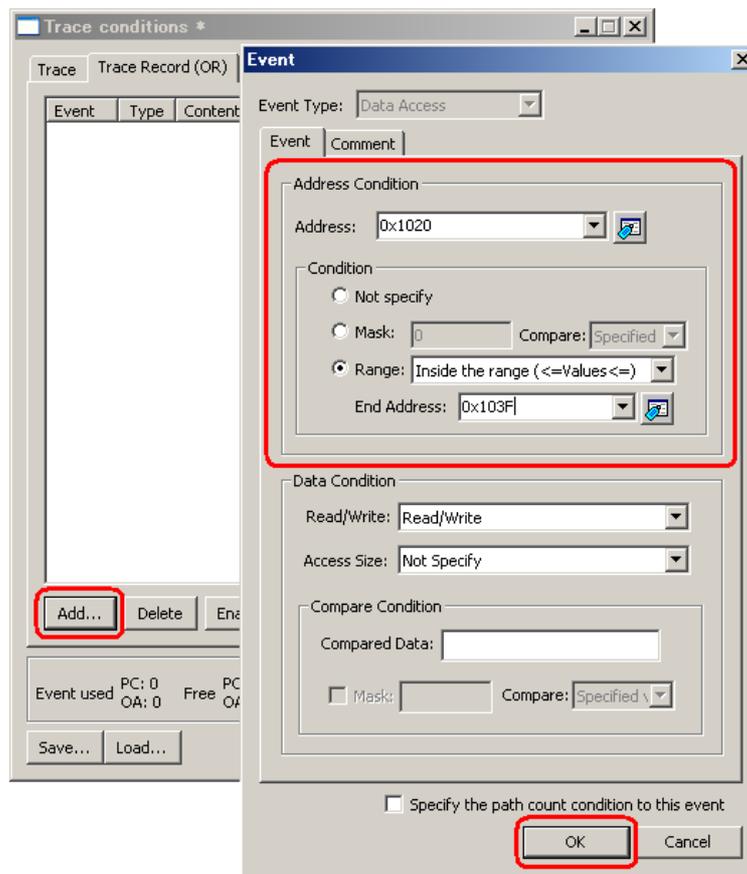


图 5.48 地址条件的设定

在 [RAM Monitor] (RAM 监视) 窗口中显示跟踪抽出条件指定的监视范围 (地址 0x1020 ~ 0x103F) 的跟踪数据。

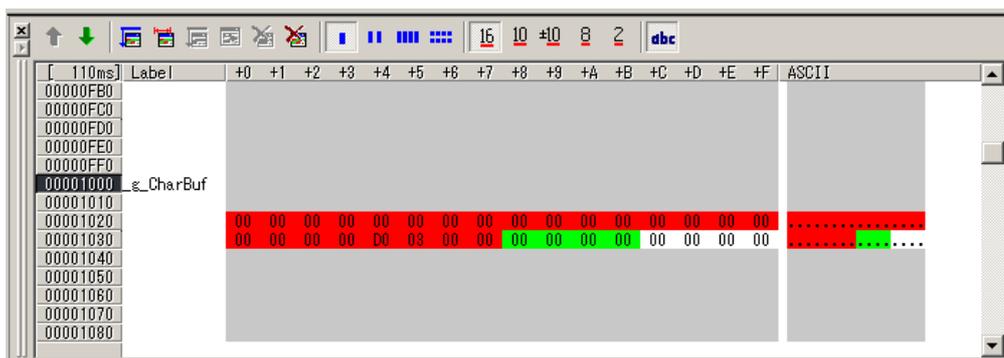


图 5.49 RAM Monitor] (RAM 监视) 窗口 (监视范围的指定)

5.11 Start/Stop（启动 / 停止）功能

Start/Stop（启动 / 停止）功能是在用户程序开始执行前以及停止执行后执行用户程序的指定例行程序的功能。

要与用户程序的执行和停止同步进行用户系统的控制时，设定任意的指定例行程序。

5.11.1 [Start/Stop function setting]（启动 / 停止功能设置）对话框的打开

在 [Start/Stop function setting]（启动 / 停止功能设置）对话框中，指定在用户程序开始执行前以及停止执行后要执行的例行程序。

如果从菜单中选择 [Setup -> Emulator -> Start/Stop function setting...]（设置 -> 仿真器 -> 启动 / 停止功能设置 ...），就打开 [Start/Stop function setting]（启动 / 停止功能设置）对话框。

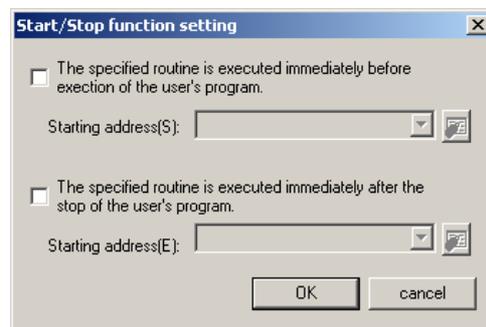


图 5.50 [Start/Stop function setting]（启动 / 停止功能设置）对话框

5.11.2 执行例行程序的指定

能分别指定在用户程序执行前以及停止执行后要执行的例行程序。

如果选定 [The specified routine is executed immediately before execution of the user's program]（在用户程序执行前执行指定的例行程序）复选框，就在用户程序执行前执行此复选框下面的 [Starting address]（起始地址）组合框指定的例行程序。

如果选定 [The specified routine is executed immediately after the stop of the user's program]（在用户程序停止之后执行指定的例行程序）复选框，就在用户程序停止执行后执行此复选框下面的 [Starting address]（起始地址）组合框指定的例行程序。

5.11.3 Start/Stop（启动/停止）功能的限制事项

Start/Stop（启动/停止）功能有以下限制：

- 不能在使用 Start/Stop（启动/停止）功能时使用以下调试功能：
 - a. 对指定例行程序的程序区域进行存储器设定和下载。
 - b. 对指定例行程序的程序区域进行断点设定。
- 在执行指定的例行程序时，通过仿真器的控制来使用中断堆栈指针指向的4字节。
- 指定例行程序中使用的通用寄存器和标志有以下限制：

表 5.17 寄存器和标志的限制事项

寄存器 / 标志名	限制事项
ISP 寄存器	在指定例行程序结束时，必须恢复为指定例行程序开始执行时的值。
U 标志	在指定例行程序的执行过程中，禁止切换到用户模式。
I 标志	在指定例行程序的执行过程中，禁止中断的发生。
PM 标志	在指定例行程序的执行过程中，禁止切换到用户模式。

- 在执行指定的例行程序时，以下调试功能无效：
 - a. 跟踪
 - b. 暂停关联
 - c. RAM 监视
 - d. 性能
 - e. 指定例行程序中的事件设定
- 在执行指定的例行程序时，始终禁止非屏蔽中断。
- 要在执行指定例行程序后开始执行用户程序时，单片机状态如下所示。

表 5.18 用户程序开始时的单片机状态

单片机资源	状态
单片机的通用寄存器	为用户程序停止执行时的状态或者用户在寄存器窗口中设定的寄存器值，不反映指定例行程序执行后的寄存器内容。
单片机空间的存储器	反映指定例行程序执行后的存储器存取。
单片机的外围功能	指定例行程序执行后的单片机外围功能继续运行。

5.11.4 有关指定例行程序记述的限制事项

指定例行程序的记述有以下限制：

- 在指定例行程序中使用堆栈时，必须使用中断堆栈。
- 在指定例行程序的处理结束位置必须记述子程序返回指令。
- 1次指定例行程序的处理时间必须在100ms以内。如果在指定例行程序中持续时钟的停止状态，仿真器就可能失控。
- 开始执行指定例行程序时的寄存器值为不定值，必须在指定例行程序中初始设置寄存器的值。
- 开始时，在管理模式中执行指定的例行程序，不能切换到用户模式。
- 在执行 Start 功能后大约需要20ms执行用户程序（CPU 时钟：25MHz，JTAG 时钟：16.5MHz）。
- 在用户程序开始执行后大约需要20ms执行 Stop 功能（CPU 时钟：25MHz，JTAG 时钟：16.5MHz）。

5.12 调试控制台功能

调试控制台功能是支持用户程序标准输入 / 输出的功能。

在 [Debug Console]（调试控制台）窗口中进行标准输入 / 输出。

此功能通过替换用户程序中的低水平接口例行程序（_charput、_charget）来实现。

【注意事项】

在利用输入/输出功能时，必须打开[Debug Console]（调试台控制）窗口。

5.12.1 [Debug Console]（调试控制台）窗口的打开

要打开 [Debug Console]（调试控制台）窗口时，选择 [View -> CPU -> Debug Console]（视图 ->CPU-> 调试控制台）或者单击 [Debug Console]（调试控制台）工具栏按钮（）。

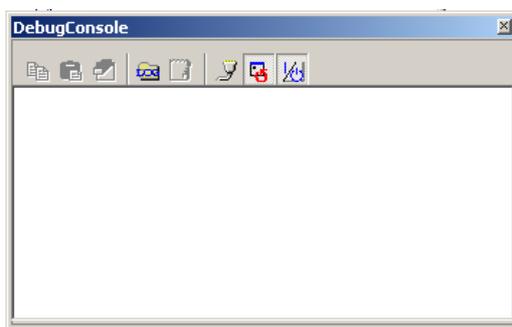


图 5.51 [Debug Console]（调试控制台）窗口

此窗口中的输出为用户程序的标准输出，此窗口中的键输入为用户程序的标准输入。

5.12.2 选项菜单

如果在窗口中单击鼠标右键，就显示以下的弹出式菜单。这些菜单的主要功能也分配了工具栏的按钮。

表 5.19 调试台控制的选项菜单

菜单名	功能
[Copy] (复制)	复制 [Debug Console] (调试台控制) 窗口中的指定范围。
[Paste] (粘贴)	粘贴到 [Debug Console] (调试台控制) 窗口。
[Erase] (删除)	清除 [Debug Console] (调试台控制) 窗口的全画面。
[Debug Console] (调试控制台)	交替选择调试控制台功能的有效和无效。
[Redirect Port Setting...] (通信端口设置 ...)	打开 COM 端口的指定对话框，指定用于通信的 COM 端口。请分别指定以下的值： COM_PORT: 用于通信的 COM 端口 BAUDRATE: 和 COM 端口进行通信的波特率 通过调试控制台功能，将用户程序的标准输入 / 输出和指定的 COM 端口进行通信。
[Set Log File...] (设置日志文件 ...)	打开日志文件的指定对话框。请指定日志文件。
[Logging] (记录)	交替指定记录的开始和结束。
[Local Echo Back] (本地回送)	交替指定本地回送的有效和无效。对 [Debug Console] (调试台控制) 对话框进行标准输入的本地回送。
[Toolbar display] (工具栏显示)	切换工具栏的显示和不显示。
[Customize toolbar] (自定义工具栏)	自定义工具栏。
[Allow Docking] (允许入坞)	将窗口入坞。
[Hide] (隐藏)	将窗口设定为隐藏。

5.12.3 有关低水平接口例行程序

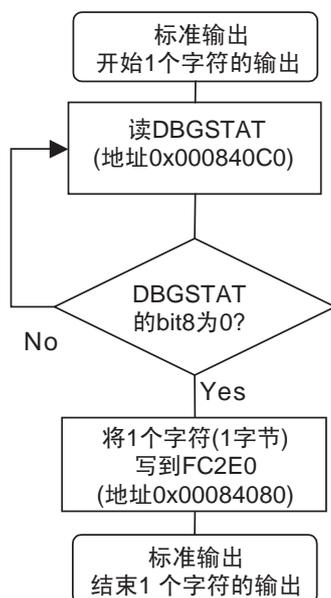
在 C/C++ 程序中使用标准输入 / 输出时，建立低水平接口例行程序。

为了实现调试控制台功能，需要将此低水平接口例行程序调用的 1 个字符输入 / 输出函数 `charput`、`charget` (`_charput`、`_charget`) 替换为用于调试控制台功能的程序。

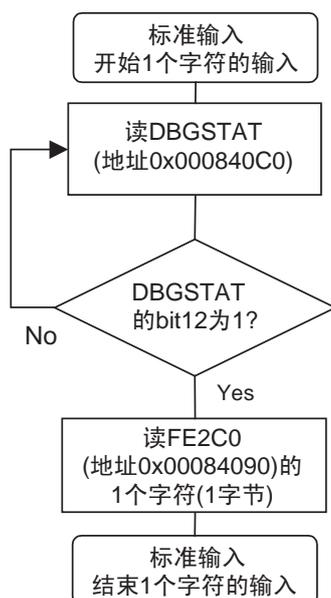
在修改为用于调试控制台功能的程序时，需要改为对特定的地址进行 1 个字符输入 / 输出的函数。处理流程和样本程序如下所示。

【注】 包括样本程序在内，符合此处理流程的程序是调试程序为调试控制台功能提供的程序。在不连接仿真器的情况下，不能在用户程序中包含调试控制台功能的程序。

`charput` 函数流程：



`charget` 函数流程：



用于调试控制台功能的样本程序

```

;-----
;
; FILE          :lowlv1.src
; DATE          :Wed, Jul 01, 2009
; DESCRIPTION   :Program of Low level
; CPU TYPE     :RX
;
;-----

.GLB          _charput
.GLB          _charget

FC2E0        .EQU 00084080h
FE2C0        .EQU 00084090h
DBGSTAT      .EQU 000840C0h
RXFLOEN      .EQU 00001000h
TXFLOEN      .EQU 00000100h
.SECTION     P,CODE
;-----
; _charput:
;-----
_charput:
        .STACK    _charput = 00000000h
__C2ESTART:  MOV.L   #TXFLOEN,R3
             MOV.L   #DBGSTAT,R4
__TXLOOP:   MOV.L   [R4],R5
             AND     R3,R5
             BNZ    __TXLOOP
__WRITEFC2E0:  MOV.L   #FC2E0,R2
             MOV.L   R1,[R2]
__CHARPUTEXIT: RTS
;-----
; _charget:
;-----
_charget:
        .STACK    _charget = 00000000h
__E2CSTART:  MOV.L   #RXFLOEN,R3
             MOV.L   #DBGSTAT,R4
__RXLOOP:   MOV.L   [R4],R5
             AND     R3,R5
             BZ     __RXLOOP
__READFE2C0:  MOV.L   #FE2C0,R2
             MOV.L   [R2],R1
__CHARGETEXIT: RTS
;-----
.END

```

5.13 热插拔功能

热插拔是为了在执行过程中给用户程序连接 E1/E20 仿真器而在执行过程中启动程序调试的方式。请注意：在通过 E1 仿真器使用热插拔时，需要在 E1 仿真器和用户系统之间连接另售的“热插拔适配器”（R0E000010ACB00）。热插拔功能的对应表如下所示。

表 5.20 产品名称和热插拔功能的对应表

产品名称（产品型号）	热插拔功能的使用
E1（R0E000010KCE00）	不能使用
E1（R0E000010KCE00）+ 热插拔适配器（R0E000010ACB00）	能使用
E20（R0E000200KCT00）	能使用

5.13.1 启动步骤

热插拔的启动步骤如下所示。在没有指示时，请不要将仿真器和用户系统连接。

1. 在没有将用户系统和仿真器连接的状态下，从[Start]（开始）菜单的[Programs]（程序）中选择 [Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop]，然后选择使用的工作空间。

2. 显示[Initial Settings]（初始设置）对话框。请选择[Mode]（模式）的[Debugging mode]（调试模式），选定[Hot plug-in]（热插拔）复选框。

在确认没有将用户系统和仿真器连接并且显示了[Communication]（通信）的仿真器系列No.的状态后，请单击[OK]按钮。

[Hot plug-in]（热插拔）复选框的状态在下次启动时无效。

有关其他设定请参照“4.3.1 [Initial Settings]（初始设置）对话框”。

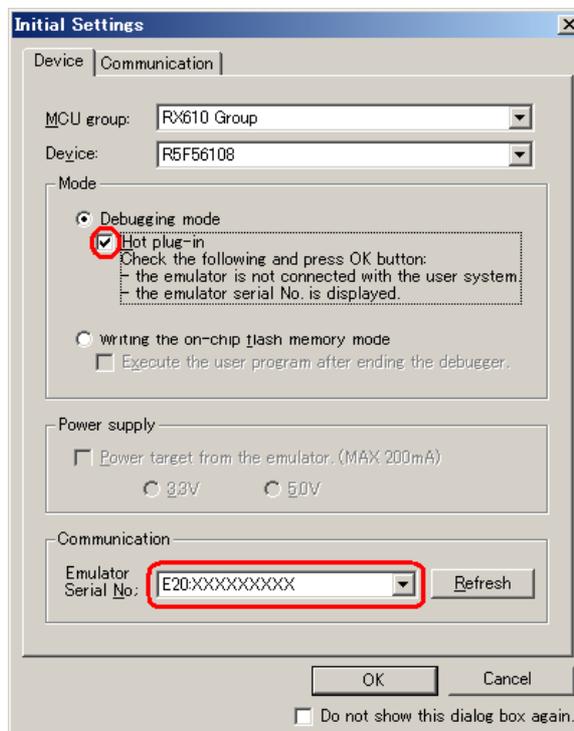


图 5.52 热插拔的 [Initial Settings]（初始设置）对话框的 [Device]（器件）页面

【注意事项】

[Initial Settings]（初始设置）对话框显示的内容可能因各单片机而不同。

3. 显示[Connect]（连接）对话框。请按照所用仿真器的连接方法将用户系统和仿真器连接。当发生改写固件时，在显示此对话框前进行改写。有关固件的改写请参照“3.10 (5) [Confirm Firmware]（确认固件）对话框”。

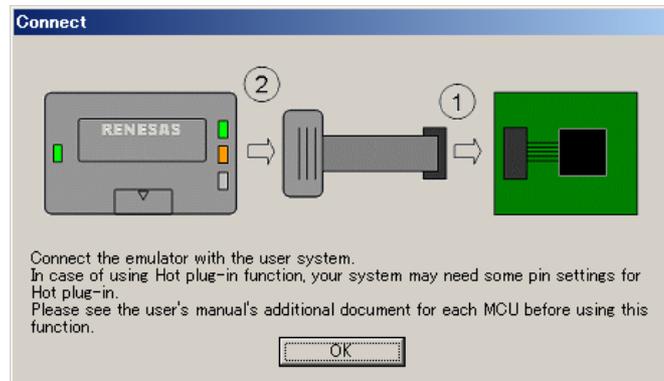


图 5.53 [Connect]（连接）对话框（E20 仿真器）

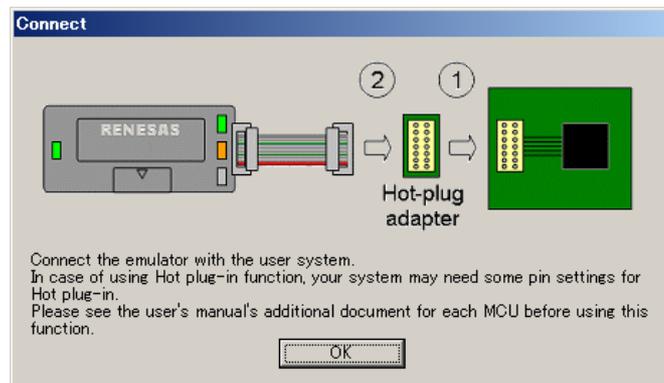


图 5.54 [Connect]（连接）对话框（E1 仿真器）

4. 显示[ID Code verification]（ID 码验证）对话框。为了保护目标单片机的内部闪存，请输入设定的 ID 码【注】。另外，在没有设定 ID 码的情况下，请按照输入模式（Hex/ASCII）输入指定位数的“0xFF”。

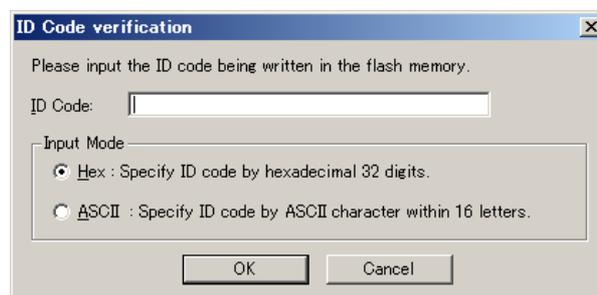


图 5.55 热插拔的 [ID Code verification]（ID 码验证）对话框

【注】 有关 ID 码请参照所用单片机的硬件手册。

5. 显示[Connecting...]（正在连接...）对话框。

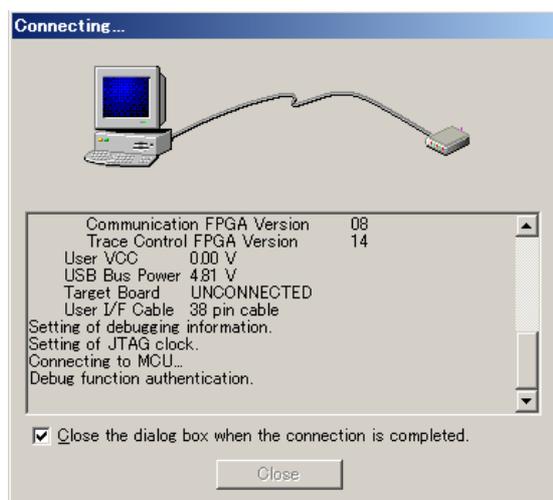


图 5.56 热插拔的 [Connecting...]（正在连接...）对话框

6. 显示[Configuration Properties]（配置属性）对话框。请进行有关仿真器和调试功能的设定，单击[OK]按钮。

有关各设定请参照“4.3.2 [Configuration Properties]（配置属性）对话框”。

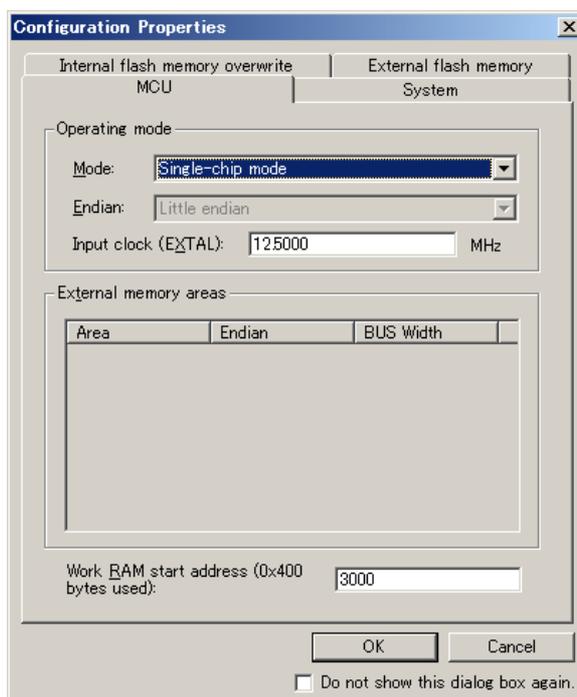


图 5.57 热插拔的 [Configuration Properties]（配置属性）对话框的 [MCU] 页面

【注意事项】

[Configuration Properties]（配置属性）对话框显示的内容可能因各单片机而不同。

7. 在 High-performance Embedded Workshop 的 [Output] 窗口中显示 “Connected” 并且显示以下信息框。
请在按 [OK] 按钮后进行下载模块文件的调试信息的下载并且按 [Go]（执行）按钮。



图 5.58 热插拔连接结束时的信息框

【注意事项】

1. 如果在将仿真器和用户系统连接的状态下选定 [Initial Settings]（初始设置）对话框的 [Hot plug-in]（热插拔）复选框后单击 [OK] 按钮，就显示以下的错误信息。



图 5.59 [User System Connection Error]（用户系统连接错误）对话框

2. 如果在没有将仿真器和用户系统连接的状态下单击 [Connect] 对话框的 [OK] 按钮，就显示以下的错误信息。



图 5.60 [User System No Connection Error]（用户系统未连接错误）对话框

3. 如果没有从外部给用户系统提供电源，就显示以下的错误信息。



图 5.61 [Command Processing]（命令处理错误）对话框

5.13.2 热插拔的注意事项

热插拔有以下注意事项：

- 在从热插拔连接结束后到按[Go]（执行）按钮的期间，不能更改事件的设定，也不能设定断点并且不能使用其他调试功能。另外，在热插拔连接时，不设定保存在对话信息中的软件断点。
- 在从热插拔连接结束后到按[Go]（执行）按钮的期间，不能在[Command Line]（命令行）窗口和[Memory]（存储器）窗口进行存储器比较。否则，不能显示正确的存储器比较结果。
- 在热插拔连接结束后从最初按[Go]（执行）按钮到停止的期间，不能使用跟踪功能。
- 在热插拔连接结束后，步进（step-in）、跨步（step-over）等执行操作和按[执行]按钮时的动作相同。
- 在通过热插拔进行启动后，[Register]（寄存器）窗口的显示内容和实际的寄存器值不同。
- 在通过热插拔进行启动的情况下，不能使用实时RAM监视（在[Configuration Properties]（配置属性）对话框中灰色显示）。
- 热插拔不对应内部ROM无效扩展模式。在[Configuration Properties]（配置属性）对话框的运行模式中，必须选择“单芯片模式”或者“内部ROM有效扩展模式”。
- 因为在热插拔时确认ID码，所以用户程序暂停大约800 μ s。
（CPU时钟：100MHz， JTAG时钟：16.5MHz）

5.14 堆栈跟踪功能

E1/E20 仿真器使用堆栈信息来显示是哪个函数调用了当前 PC 的函数。此功能只能用于加载了含有 Elf/Dwarf2 格式调试信息的装入模块的情况。

有关此功能的使用方法，请参照“6.18 堆栈跟踪功能”。

5.15 在线帮助

在线帮助功能记载各功能的操作方法以及能从命令行窗口输入的命令语法。

要查阅仿真器功能的帮助时，请选择 [Help]（帮助）菜单 -> [Emulator Help]（仿真器帮助）。

6. 教程

6.1 绪论

为了实际操作仿真器并且理解主要功能，提供了教程程序。在本章节中，使用此教程程序说明 E1/E20 仿真器的主要功能。

此教程程序是用 C 语言编写的，按照升降顺序将 10 个随机数据进行排序。

指南程序进行以下处理：

main 函数重复进行排序处理，因此重复调用 tutorial 函数。

tutorial 函数生成要排序的随机数据，按顺序调用 sort 函数和 change 函数。

sort 函数输入保存了由 tutorial 函数生成的随机数据的数组，并且按照升序进行排序。

change 函数输入由 sort 函数按照升序进行排序的数组，并且按照降序进行排序。

教程程序是为了理解仿真器功能的使用方法以及仿真调试程序的操作方法而编写的程序。在开发用户系统和用户程序时，请确认目标单片机的各种手册。

- 【注】**
1. 此程序以小端法运行。要进行大端法运行时，请重新编译。
 2. 如果重新编译，就可能和本章说明的地址有所不同。
 3. 本产品附带的教程程序的文件扩展名和运行地址因单片机而不同，因此显示的内容可能和本章节的说明有所不同。

6.2 High-performance Embedded Workshop 的启动

按照“3.10 E1/E20 仿真调试程序的启动步骤”的步骤，打开工作空间。

请指定以下目录：

安装 OS 的驱动器 \Workspace\Tutorial\E1E20\RX600\Tutorial_LittleEndian

请指定以下文件：

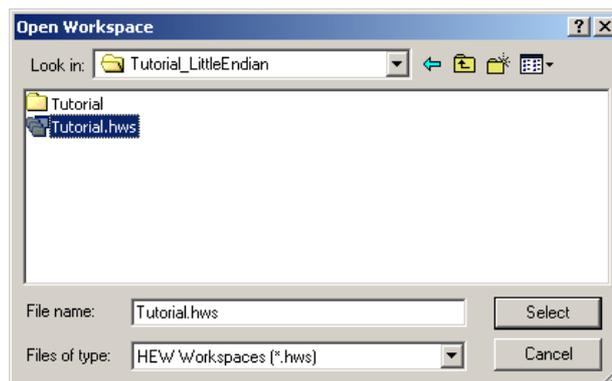


图 6.1 [Open Workspace] (打开工作空间) 对话框

6.3 仿真器的连接

如果连接仿真器，就显示设置调试程序的对话框。在此对话框中进行调试程序的初始设置。当结束调试程序的设置时，就进入能调试的状态。

6.4 教程程序的下载

6.4.1 教程程序的下载

下载要调试的目标程序。下载的程序和下载的地址因使用的单片机而不同。请按照所使用的单片机对画面的显示等进行适当的调整。

在选择 [Download modules] 的 [Tutorial.abs] 的状态下单击鼠标右键，从显示的弹出式菜单中选择 [Download] (下载)。

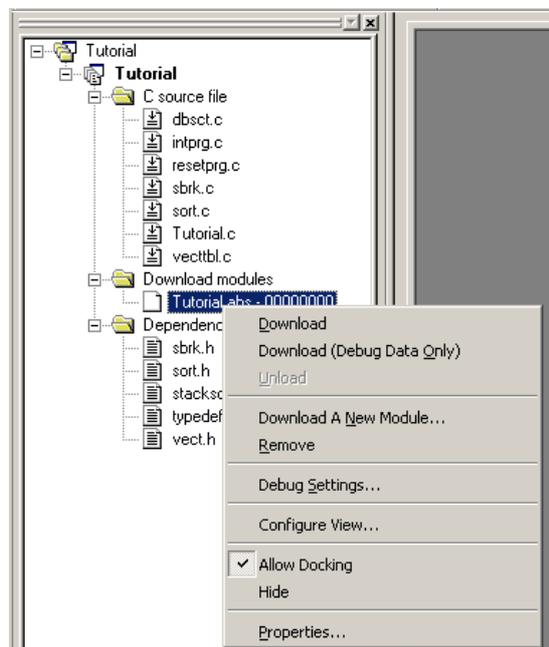


图 6.2 教程程序的下载画面

6.4.2 源程序的显示

High-performance Embedded Workshop 能以源级调试程序。

双击 [C source file] 的 [Tutorial.c]。

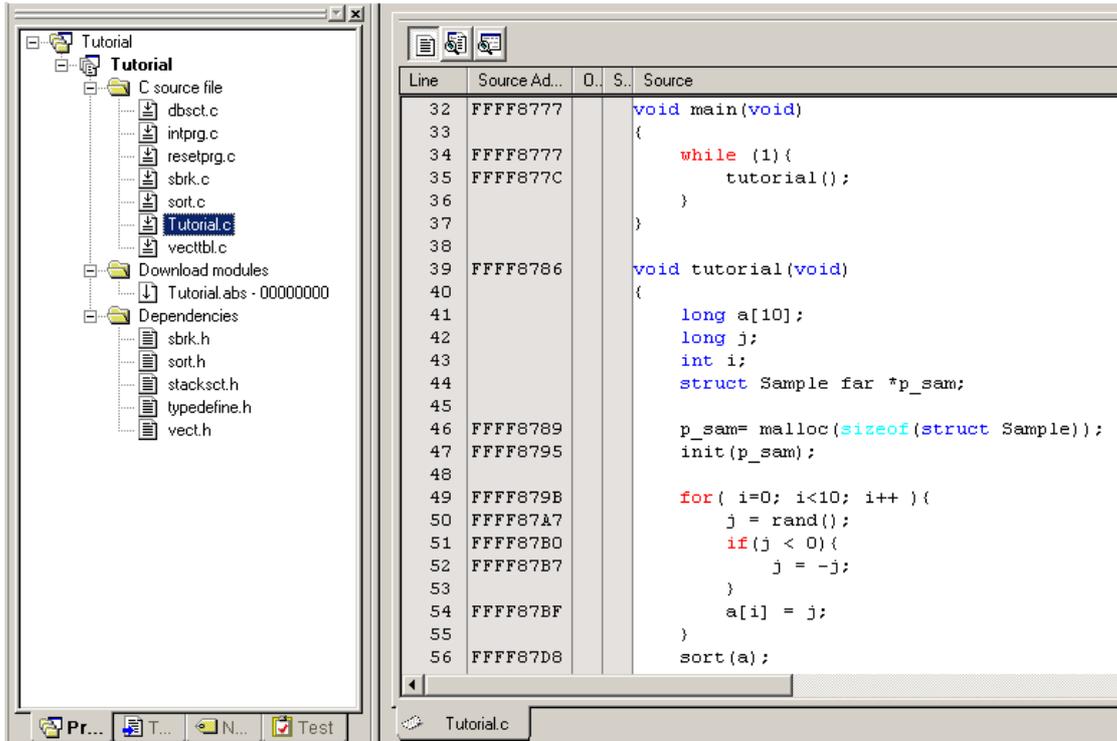


图 6.3 [Editor] (编辑器) 窗口 (源程序的显示)

如果需要，能更改易读的字体和大小。有关更改方法请参照 High-performance Embedded Workshop 用户手册。

在 [Editor] (编辑器) 窗口中最初显示程序的起始部分，但是能使用滚动栏查看其他部分。

6.5 S/W 断点的设定

S/W 断点（软件暂停）是基本调试功能中的一个功能。

能在 [Editor]（编辑器）窗口中简单地设定 S/W 断点。说明在调用 sort 函数的位置设定 S/W 断点的方法。

请双击 sort 函数调用行的 [S/W breakpoints]（S/W 断点）列。

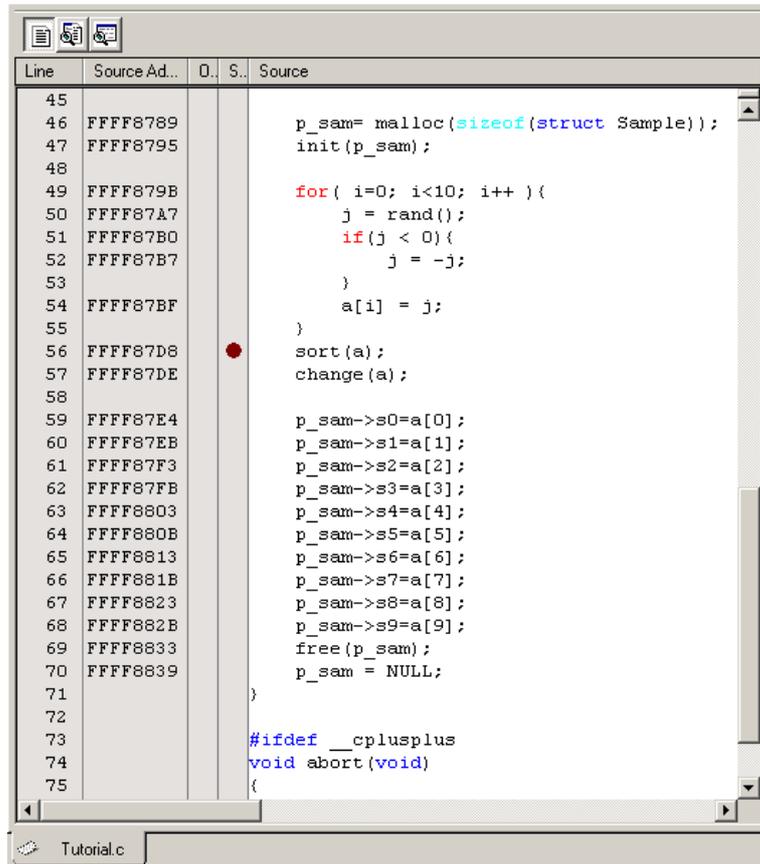


图 6.4 [Editor]（编辑器）窗口（S/W 断点的设定）

在 sort 函数行上显示红色圆点，表示 S/W 断点已被设定。

6.6 程序的执行

说明程序的执行方法。

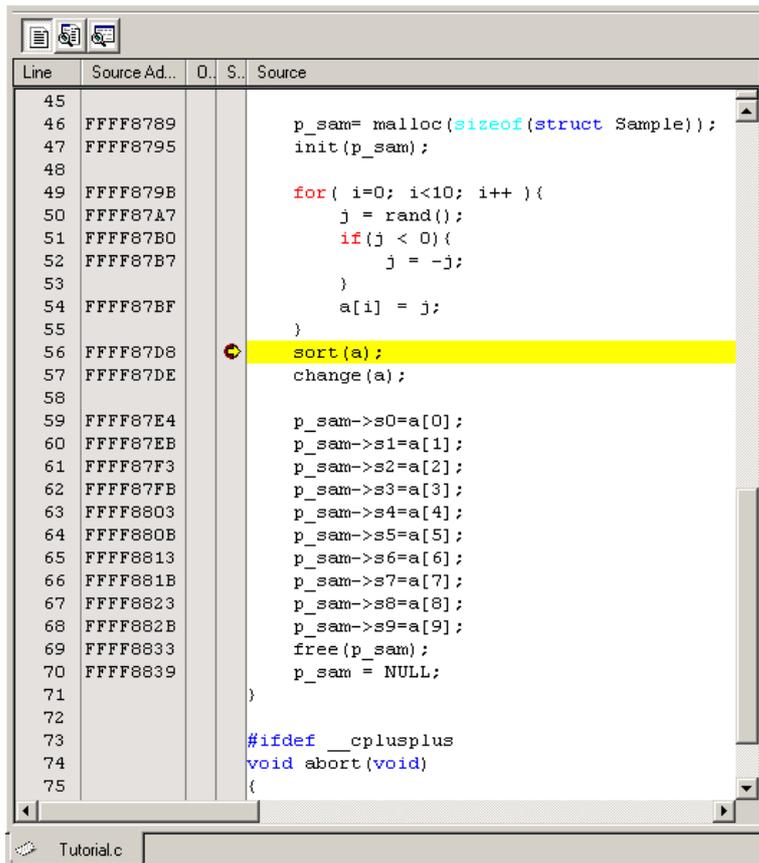
6.6.1 CPU 的复位

要对 CPU 进行复位时，从 [Debug]（调试）菜单中选择 [Reset CPU]（CPU 复位）或者选择工具栏上的 [Reset CPU]（CPU 复位）按钮 。

6.6.2 程序的执行

要执行程序时，从 [Debug]（调试）菜单中选择 [Go]（执行）或者选择工具栏上的 [Go]（执行）按钮 。

程序执行到设有断点的位置。在 [S/W breakpoints]（S/W 断点）列中显示箭头，表示程序停止的位置。



Line	Source Ad...	O.	S.	Source
45				
46	FFFF8789			p_sam= malloc(sizeof(struct Sample));
47	FFFF8795			init(p_sam);
48				
49	FFFF879B			for(i=0; i<10; i++){
50	FFFF87A7			j = rand();
51	FFFF87B0			if(j < 0){
52	FFFF87B7			j = -j;
53				}
54	FFFF87BF			a[i] = j;
55				}
56	FFFF87D8	●		sort(a);
57	FFFF87DE			change(a);
58				
59	FFFF87E4			p_sam->s0=a[0];
60	FFFF87EB			p_sam->s1=a[1];
61	FFFF87F3			p_sam->s2=a[2];
62	FFFF87FB			p_sam->s3=a[3];
63	FFFF8803			p_sam->s4=a[4];
64	FFFF880B			p_sam->s5=a[5];
65	FFFF8813			p_sam->s6=a[6];
66	FFFF881B			p_sam->s7=a[7];
67	FFFF8823			p_sam->s8=a[8];
68	FFFF882B			p_sam->s9=a[9];
69	FFFF8833			free(p_sam);
70	FFFF8839			p_sam = NULL;
71				}
72				
73				#ifdef __cplusplus
74				void abort(void)
75				{

图 6.5 [Editor]（编辑器）窗口（暂停状态）

能在 [Status]（状态）窗口中确认最后发生暂停的原因。

请选择 [View -> CPU -> Status]（视图 -> CPU -> 状态）或者单击 [View Status]（查看状态）工具栏按钮



显示 [Status]（状态）窗口，请打开 [Platform] 表进行确认。

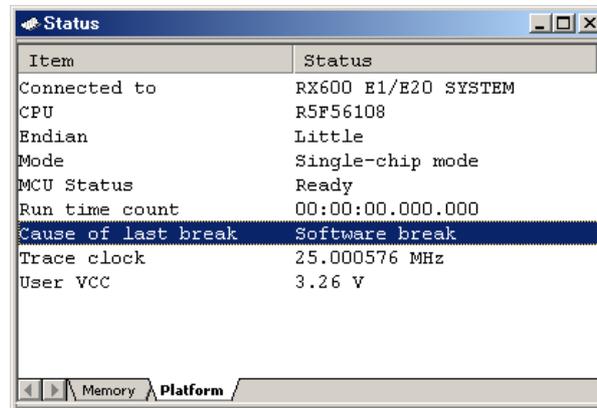


图 6.6 [Status]（状态）窗口

【注】 此窗口显示的内容因各仿真器而不同。有关显示的内容请参照在线帮助。

6.7 断点的确认

能在 [Breakpoints]（断点）对话框中确认设定的全部 S/W 断点。

6.7.1 断点的确认

请选择 [Edit]（编辑）菜单的 [Source Breakpoints...]（源断点...）。显示 [Breakpoints]（断点）对话框。

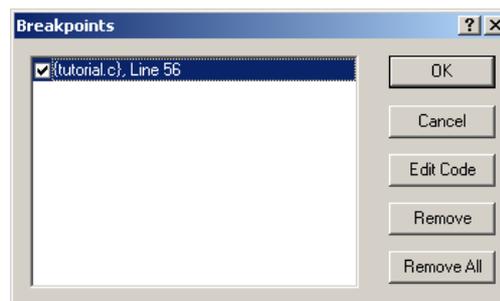


图 6.7 [Breakpoints]（断点）对话框

能使用此对话框删除断点以及选择断点的有效或者无效。

【注】 在没有设定 S/W 断点的情况下，不能打开 [Breakpoints]（断点）对话框。

6.8 寄存器内容的更改

如果选择 [View -> CPU -> Registers] (视图 -> CPU -> 寄存器) 或者单击 [Registers] (寄存器) 工具栏按钮



，就显示 [Register] (寄存器) 窗口。

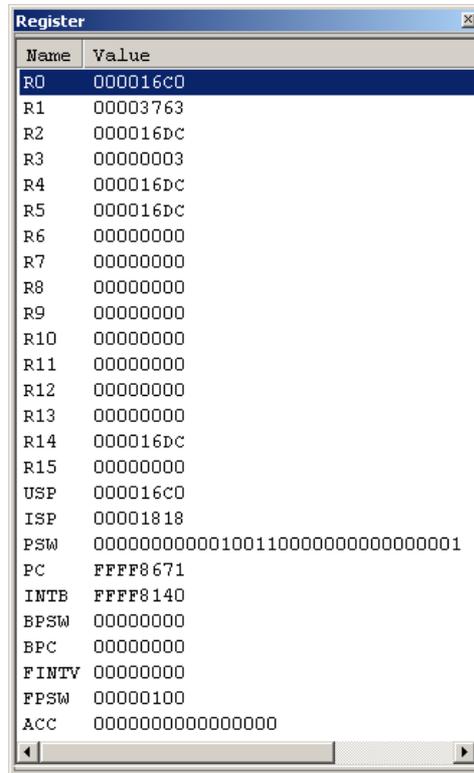


图 6.8 [Register] (寄存器) 窗口

能更改任意寄存器的内容。

请双击要更改的寄存器行。显示对话框，请输入要更改的值。

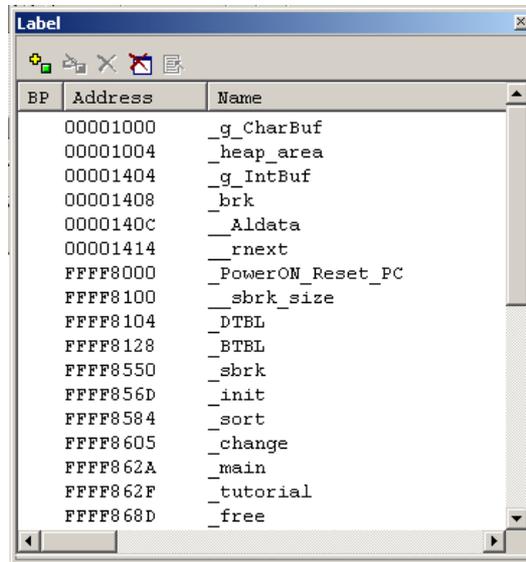


图 6.9 [Set Value] (设定值) 对话框 (PC)

6.9 符号的参照

能使用 [Labels] (标签) 窗口显示模块内的符号信息。

如果选择 [View -> Symbols -> Labels] (视图 -> 符号 -> 标签) 或者单击 [Labels] (标签) 工具栏按钮 , 就显示 [Labels] (标签) 窗口。能参照模块内的符号信息。



BP	Address	Name
	00001000	_g_CharBuf
	00001004	_heap_area
	00001404	_g_IntBuf
	00001408	_brk
	0000140C	_Aldata
	00001414	_rnext
	FFFF8000	_PowerON_Reset_PC
	FFFF8100	_sbrk_size
	FFFF8104	_DTBL
	FFFF8128	_BTBL
	FFFF8550	_sbrk
	FFFF856D	_init
	FFFF8584	_sort
	FFFF8605	_change
	FFFF862A	_main
	FFFF862F	_tutorial
	FFFF868D	_free

图 6.10 [Labels] (标签) 窗口

6.10 存储器内容的确认

通过指定标签名，能在 [Memory]（存储器）窗口中确认注册了标签的存储器内容。例如，按照以下的步骤，以字节长度确认对应 _main 的存储器内容。

请选择 [View -> CPU -> Memory]（视图 -> CPU -> 存储器）或者单击 [Memory]（存储器）工具栏按钮 ，打开 [Display Address]（显示地址）对话框。

请在 [Display Address]（显示地址）编辑框中输入 “_main”。



图 6.11 [Display Address]（显示地址）对话框

请单击 [OK] 按钮。显示指定存储区的 [Memory]（存储器）窗口。

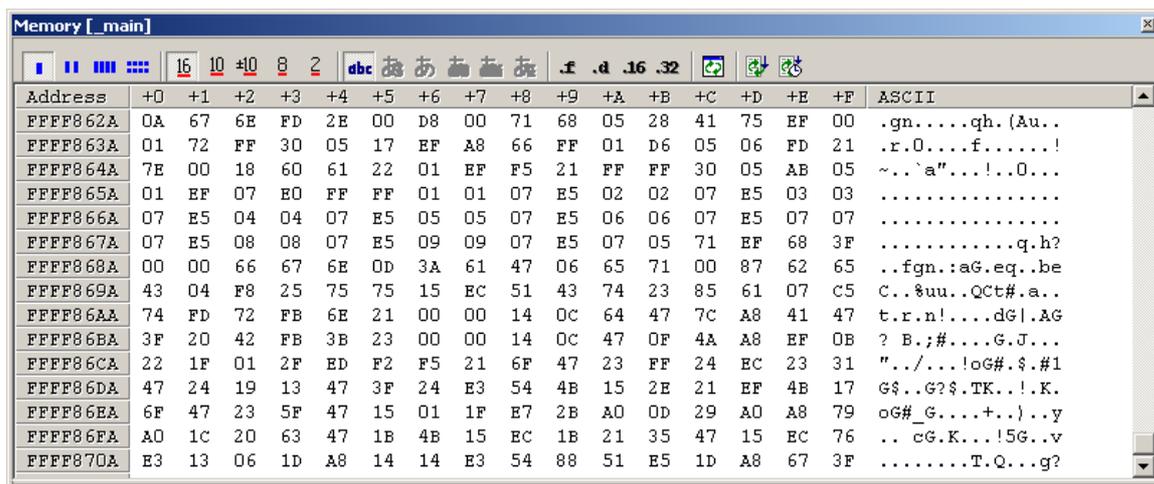


图 6.12 [Memory]（存储器）窗口

6.11 变量的参照

在对程序进行步进处理时，能确认程序使用的变量值的变化。例如，能按照以下的步骤查看在程序开始处声明的 long 型的数组 a。

请单击 [Editor]（编辑器）窗口中显示的数组 a 的左边，把光标放到该位置。

请用鼠标右键选择 [Instant Watch]（即时监视）。

显示以下对话框。



图 6.13 [Instant Watch]（即时监视）对话框

请单击 [Add]（添加）按钮，将变量添加到 [Watch]（监视）窗口。

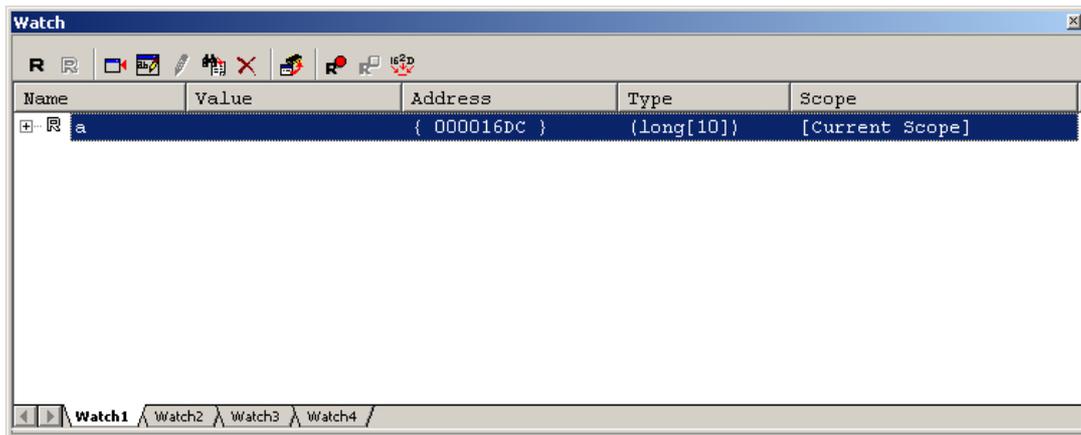


图 6.14 [Watch]（监视）窗口（数组的显示）

能指定变量名，将变量添加到 [Watch]（监视）窗口。

请用鼠标右键单击 [Watch]（监视）窗口，从弹出式菜单中选择 [Add Watch]（添加监视）。显示以下对话框。



图 6.15 [Add Watch]（添加监视）对话框

在 [Variable or expression]（变量或者表达式）编辑框中输入变量 `i`，单击 [OK] 按钮。
在 [Watch]（监视）窗口中显示 `int` 型的变量 `i`。

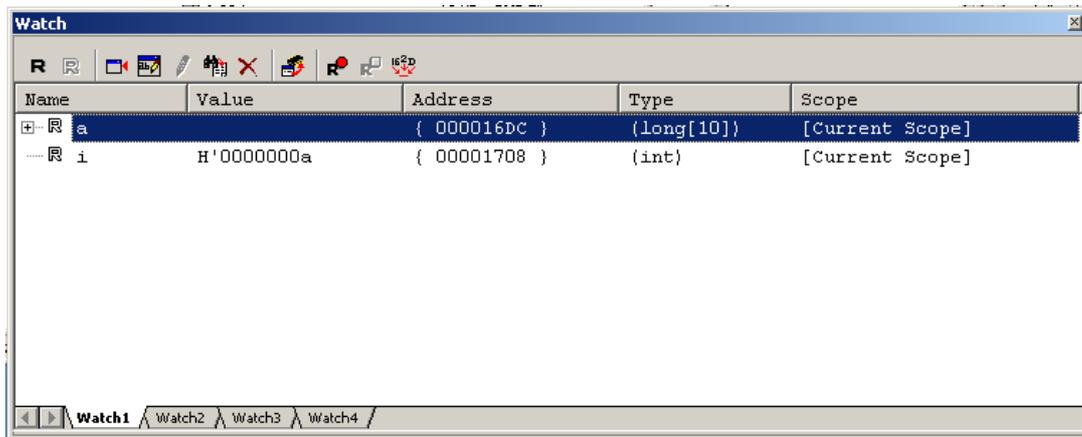


图 6.16 [Watch]（监视）窗口（变量的显示）

单击 [Watch]（监视）窗口的数组 `a` 左边的 “+” 标记，能参照数组 `a` 的各元素。

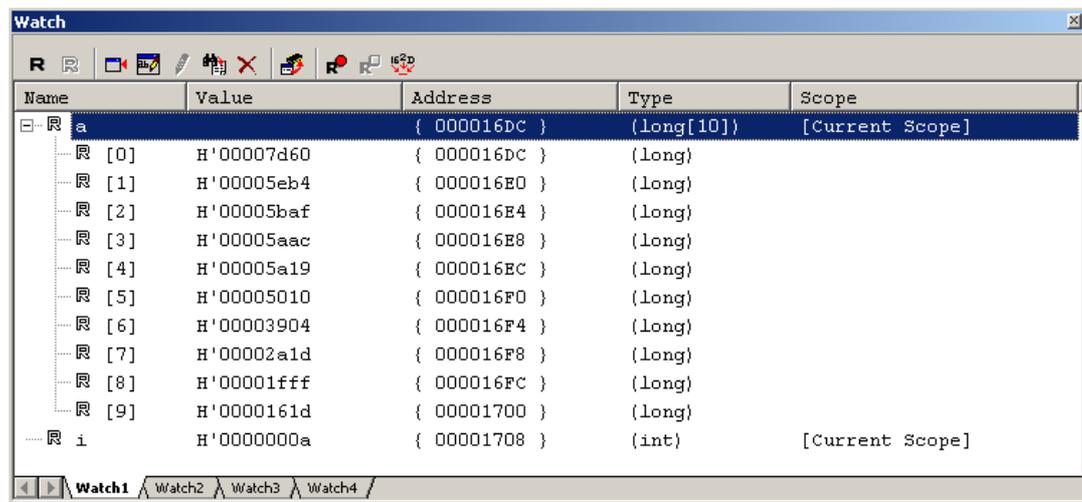


图 6.17 [Watch]（监视）窗口（数组元素的显示）

6.12 局部变量的显示

能使用 [Local] (局部) 窗口显示函数内的局部变量。例如, 检查 tutorial 函数的局部变量。此函数声明 4 个局部变量 a、j、i、p_sam。

如果选择 [View -> Symbols -> Local] (视图 -> 符号 -> 局部) 或者单击 [Local] (局部) 工具栏按钮 , 就显示 [Local] (局部) 窗口。

在 [Local] (局部) 窗口中显示当前程序计数器 (PC) 指向的函数局部变量和该变量的值。

当函数内没有局部变量时, [Local] (局部) 窗口中什么也不显示。

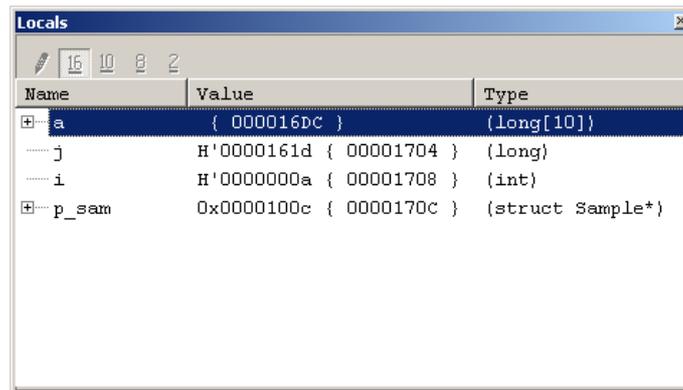


图 6.18 [Local] (局部) 窗口

请单击 [Local] (局部) 窗口的类实例 p_sam 左边的 “+” 标记, 显示类实例 p_sam 的结构元素。

请参照 sort 函数执行前后的类实例 p_sam 元素, 确认随机数据已按降序排序。

6.13 程序的单步执行

High-performance Embedded Workshop 为程序的调试提供了各种有效的步进命令。

表 6.1 步进选项

No.	命令	说明
1	Step In (步进)	执行各语句 (包括函数内的语句)。
2	Step Over (跨步)	将函数调用作为 1 步进行单步执行。
3	Step Out (步出)	跳出函数并且停止在调用该函数的程序的下一个语句。
4	Step... (单步)	以指定的速度进行指定次数的单步执行。

6.13.1 Step In (步进) 的执行

[Step In] (步进) 进入调用函数，并且停止在调用函数的第一个语句。

要进入 sort 函数时，请从 [Debug] (调试) 菜单中选择 [Step In] (步进) 或者单击工具栏的 [Step In] (步进) 按钮。



图 6.19 [Step In] (步进) 按钮

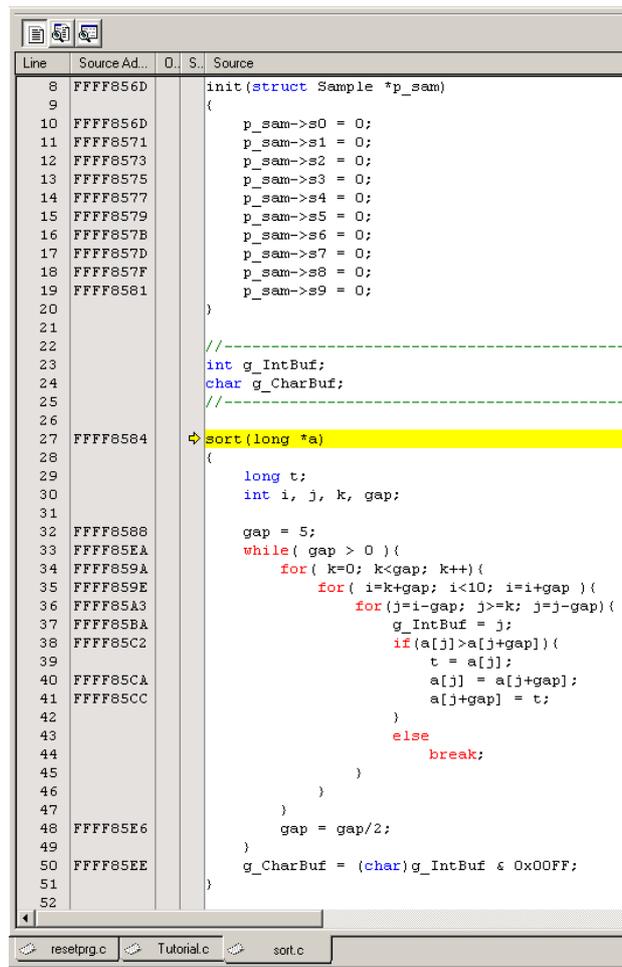


图 6.20 [Editor] (编辑器) 窗口 (Step In (步进))

[Editor] (编辑器) 窗口中的强调显示移动到 sort 函数的第一个语句。

6.13.2 Step Out（步出）的执行

[Step Out]（步出）从调用函数跳出，并且停止在调用源程序的下一个语句。

要从 sort 函数跳出时，请从 [Debug]（调试）菜单中选择 [Step Out]（步出）或者单击工具栏的 [Step Out]（步出）按钮。



图 6.21 [Step Out]（步出）按钮

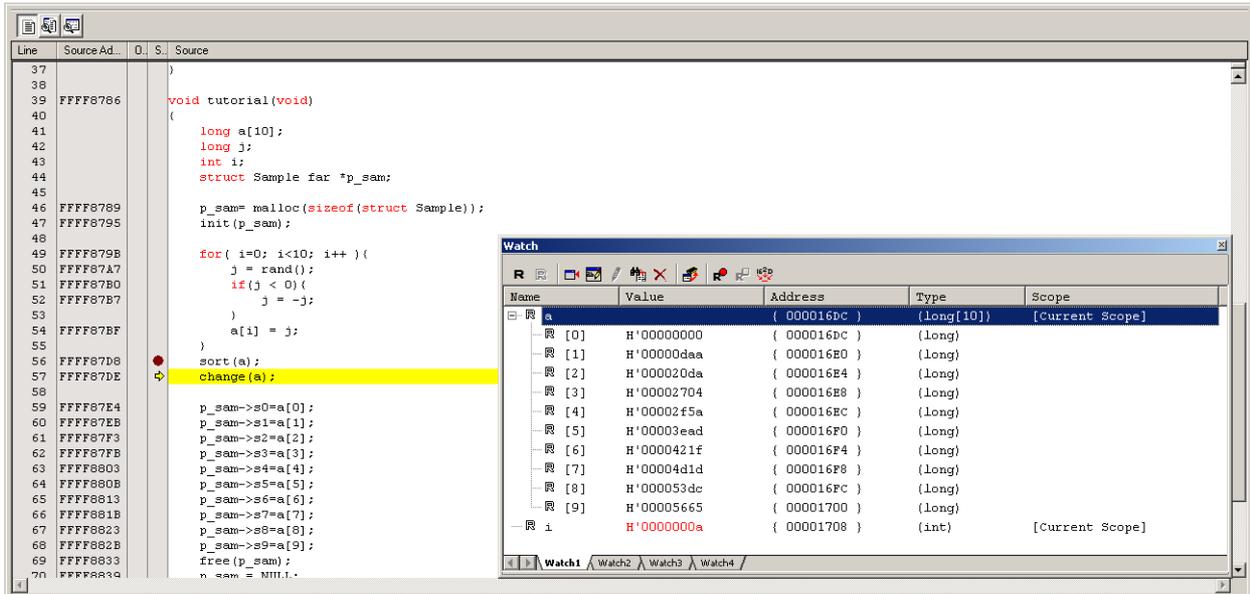


图 6.22 [Editor]（编辑器）窗口（Step Out（步出））

按升序将 [Editor]（编辑器）窗口中显示的变量 a 的数据进行排序。

6.13.3 Step Over（跨步）的执行

[Step Over]（跨步）将函数调用作为 1 步进行单步执行，并且停止在主程序的下一个语句。

要一次单步执行 change 函数中的语句时，请从 [Debug]（调试）菜单中选择 [Step Over]（跨步）或者单击工具栏的 [Step Over]（跨步）按钮。



图 6.23 [Step Over]（跨步）按钮

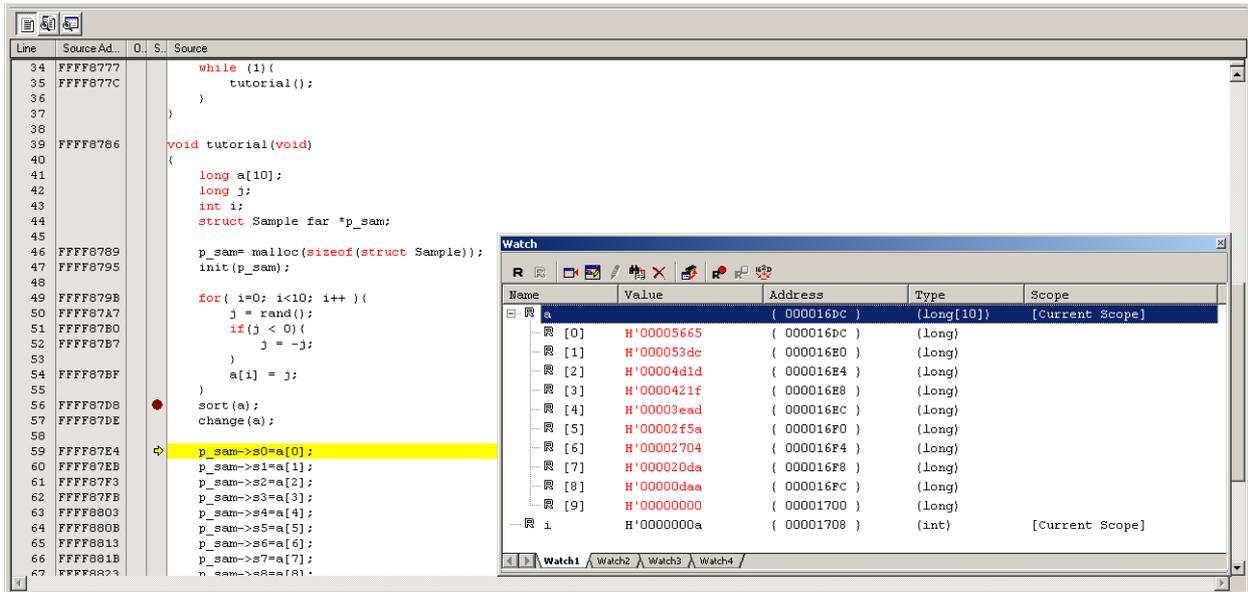


图 6.24 [Editor]（编辑器）窗口（Step Over（跨步））

按降序将 [Editor]（编辑器）窗口中显示的变量 a 的数据进行排序。

6.14 程序的强制暂停

High-performance Embedded Workshop 能强制暂停程序的执行。

请解除全部断点。

要执行 tutorial 函数的剩余部分时，请从 [Debug]（调试）菜单中选择 [Go]（执行）或者选择工具栏的 [Go]（执行）按钮。



图 6.25 [Go]（执行）按钮

因为程序正在执行无限循环处理，所以请从 [Debug]（调试）菜单中选择 [Stop Program]（停止程序）或者选择工具栏的 [Halt]（暂停）按钮，强制暂停程序的执行。



图 6.26 [Halt]（暂停）按钮

6.15 On-chip 暂停功能

能根据单片机的功能使用 On-chip 暂停功能。On-chip 暂停功能是在程序执行指定的地址时（取指令）或者读写指定的存储器时（数据存取）停止程序执行的功能。

6.15.1 执行指定地址时的程序停止执行

能在 [Editor]（编辑器）窗口中简单地设定取指令事件。例如，在 sort 函数的调用位置设定取指令事件。请双击 sort 函数调用行的 [On-chip breakpoints]（On-chip 断点）列。

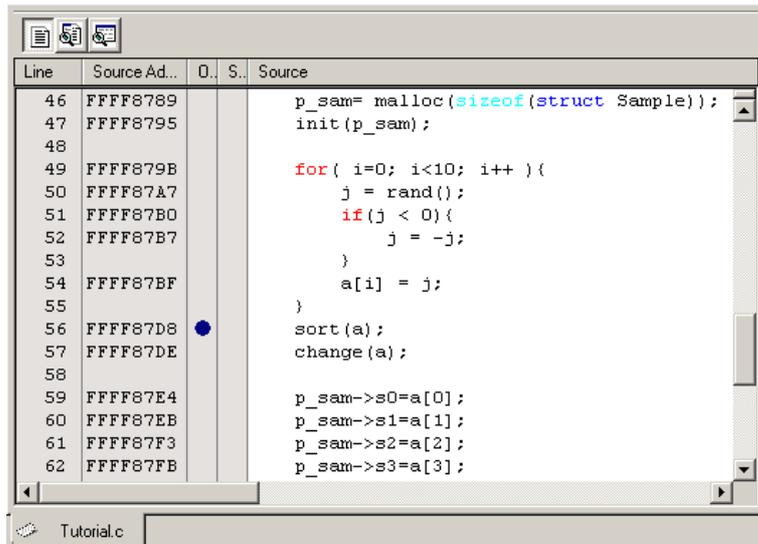


图 6.27 [Editor]（编辑器）窗口（On-chip 断点的设定）

在 sort 函数行上显示“●”，表示取指令时停止执行的 On-chip 断点已被设定。

6.16 存取存储器时的程序停止执行

如果要在读写全局变量值时停止程序执行，需要进行以下的设定。

选择 [View -> Event -> On-chip Break]（视图 -> 事件 -> On-chip 暂停），显示 [On-Chip Break]（On-chip 暂停）对话框。

显示 [On-Chip Break]（On-chip 暂停）对话框的 [OR] 选项卡。在 [Editor]（编辑器）窗口中选择要在读写值时停止程序执行的全局变量并且拖放到此 [OR] 选项卡，然后单击 [Apply]（应用）按钮。

如果执行程序，就在读写所设定的全局变量值时停止执行。

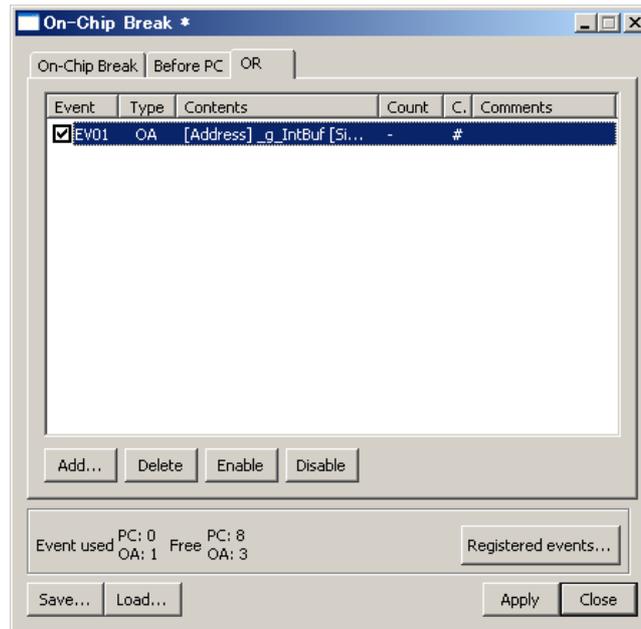


图 6.28 [On-Chip Break]（On-chip 暂停）对话框

- 【注】
1. 能设定的全局变量只能是 1 字节或者 2 字节的变量。
 2. 不能指定局部变量。

6.17 跟踪功能

根据单片机的功能，E1/E20 仿真器能使用以下的跟踪功能：

- 内部跟踪功能
通过使用单片机内置的跟踪缓冲器来实现此功能。能显示转移信息和数据存取信息，显示的内容和能获取的周期数因单片机而不同。
- 外部跟踪输出功能
只有E20仿真器对应此功能。在使用能进行外部跟踪输出的单片机并且将单片机的外部跟踪引脚连接E20仿真器时有效。显示的内容和能获取的周期数因单片机而不同。

RX600 系列的设定例子如下所示。

请选择 [View -> Code -> Trace]（视图 -> 代码 -> 跟踪）或者单击 [Trace]（跟踪）工具栏按钮 。显示 [Trace]（跟踪）窗口。



图 6.29 [Trace]（跟踪）窗口

跟踪功能的概要和设定方法说明如下。

6.17.1 Fill until Stop（填充直至停止）跟踪的跟踪信息显示

Fill until Stop（填充直至停止）跟踪功能连续获取用户程序开始执行到暂停为止的跟踪信息。

1. 请解除全部暂停条件。用鼠标右键单击[Trace]（跟踪）窗口，从打开的弹出式菜单中选择[Acquisition]（获取），显示[Trace conditions]（跟踪条件）对话框，请确认获取模式为[Fill until Stop]（填充直至停止）。请单击[Close]（关闭）按钮。

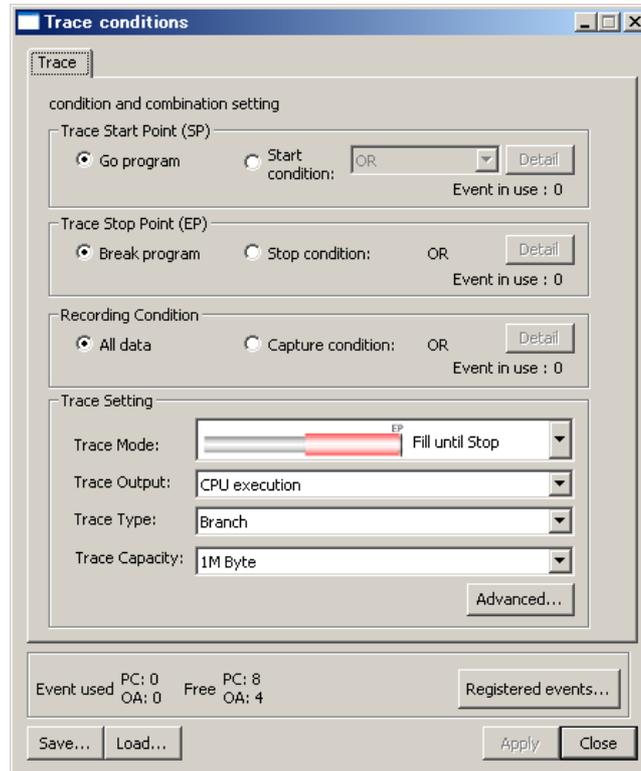


图 6.30 [Trace conditions]（跟踪条件）对话框（Fill until Stop（填充直至停止）跟踪）

2. 请在tutorial函数内记述“p_sam->s0=a[0];”的行设定S/W断点。

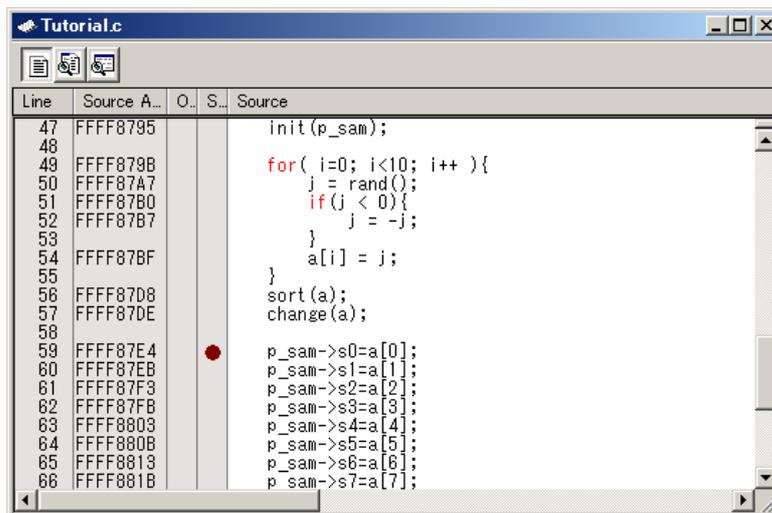


图 6.31 [Editor]（编辑器）窗口（在tutorial函数内设定S/W断点）

3. 请从[Debug]（调试）菜单中选择[Reset Go]（复位执行）。通过暂停来停止处理，在[Trace]（跟踪）窗口中显示暂停为止的跟踪信息。

Cycle	Label	Address	Source	Destination Address	Data	Size	R/W	BUS Master	Type	BCND	Branch Type	Channel	TimeStamp (Count)
-00000014		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000013		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	00110	-----	---	---
-00000012		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000011		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	010	-----	---	---
-00000010		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000009		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	010	-----	---	---
-00000008		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000007		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	00110	-----	---	---
-00000006		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000005		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	010	-----	---	---
-00000004		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000003		FFFF8620	-----	FFFF8620	-----	-----	---	CPU	BCND	001011	-----	---	---
-00000002		FFFF87DE	-----	FFFF87DE	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
-00000001	_change	FFFF870D	-----	FFFF870D	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
00000000		FFFF870D	-----	FFFF870D	-----	-----	---	CPU	BCND	011111	-----	---	---

图 6.32 [Trace]（跟踪）窗口（Fill until Stop（填充直至停止）跟踪）

4. 能进行总线、反汇编和源的混合显示。如果从弹出式菜单中选择[Display Mode -> DIS]（显示模式 -> DIS），就能混合显示总线和反汇编。

Cycle	Label	Address	Source	Destination Address	Data	Size	R/W	BUS Master	Type	BCND	Branch Type	Channel	TimeStamp (Count)
	FFFF8707		MOV.B	R14, [R1]									
	FFFF8709		ADD	#18H, R0, R0									
	FFFF870C		RTS										
-00000002		FFFF87DE	-----	FFFF87DE	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
	FFFF87DE		MOV.L	R0, R1									
	FFFF87E0		BSR.A	_change									
-00000001	_change	FFFF870D	-----	FFFF870D	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
	FFFF870D	_change	ADD	#-30H, R0, R0									
	FFFF8710		MOV.L	R1, 2CH[R0]									
	FFFF8712		MOV.L	#0H, R14									
	FFFF8714		MOV.L	R14, 28H[R0]									
	FFFF8717		MOV.L	28H[R0], R14									
	FFFF871A		CMP	#0AH, R14									
	FFFF871C		BGE.B	0FFFF8740H									
00000000		FFFF871C	-----	FFFF871C	-----	-----	---	CPU	BCND	011111	-----	---	---

图 6.33 [Trace]（跟踪）窗口（总线和反汇编的混合显示）

5. 如果从弹出式菜单中选择[Display Mode -> SRC]（显示模式 -> SRC），就能混合显示总线、反汇编和源。

Cycle	Label	Address	Source	Destination Address	Data	Size	R/W	BUS Master	Type	BCND	Branch Type	Channel	TimeStamp (Count)
-00000002		FFFF87DE	-----	FFFF87DE	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
	FFFF87DE		MOV.L	R0, R1									
	Tutorial.c	57	: change(a);										
	FFFF87E0		BSR.A	_change									
-00000001	_change	FFFF870D	-----	FFFF870D	-----	-----	---	CPU	DESTINATION	-----	-----	---	---
	FFFF870D	_change	ADD	#-30H, R0, R0									
	sort.c	53	: change(long *a)										
	FFFF8710		MOV.L	R1, 2CH[R0]									
	FFFF8712		MOV.L	#0H, R14									
	sort.c	58	: for(i=0; i<10; i++){										
	FFFF8714		MOV.L	R14, 28H[R0]									
	FFFF8717		MOV.L	28H[R0], R14									
	FFFF871A		CMP	#0AH, R14									
	FFFF871C		BGE.B	0FFFF8740H									
00000000		FFFF871C	-----	FFFF871C	-----	-----	---	CPU	BCND	011111	-----	---	---

图 6.34 [Trace]（跟踪）窗口（总线、反汇编和源的混合显示）

6.18 堆栈跟踪功能

使用堆栈信息，显示由哪个函数调用了当前 PC 所在的函数。

请双击 sort 函数内的行的 [S/W Breakpoints] (S/W 断点) 列，设定软件断点。

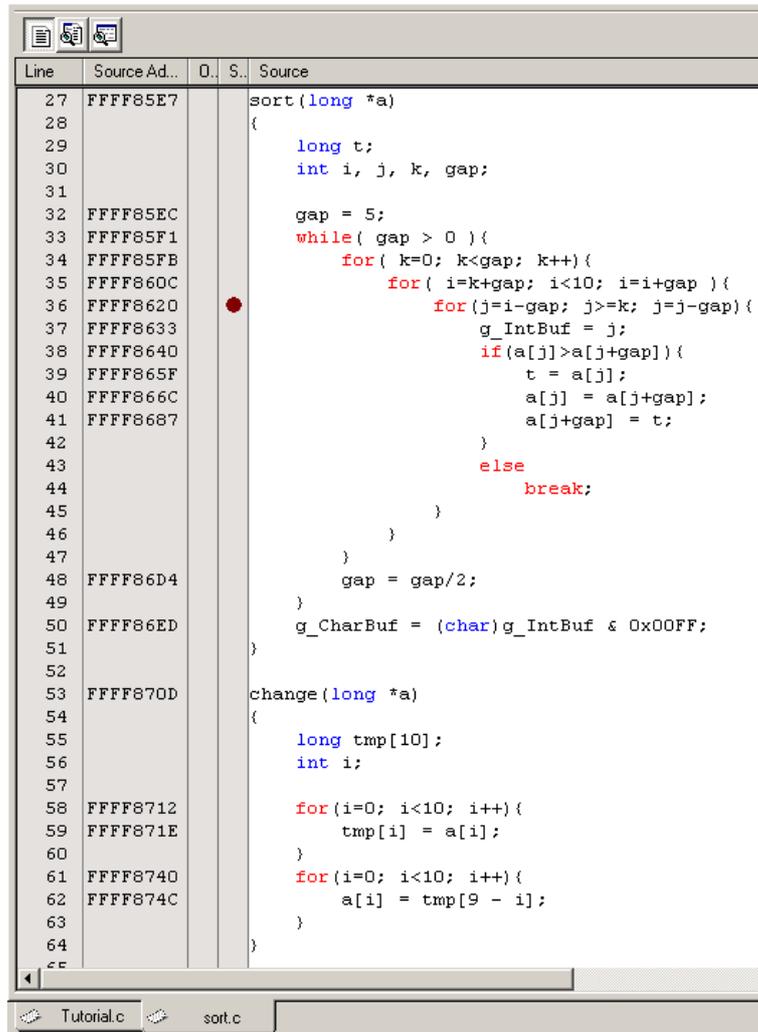
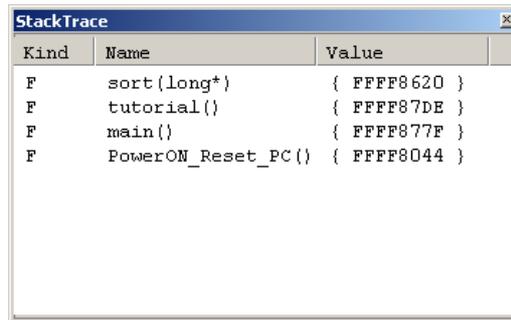


图 6.35 [Editor] (编辑器) 窗口 (S/W 断点的设定)

请从 [Debug] (调试) 菜单中选择 [Reset Go] (复位执行)。

请在暂停后从 [View]（视图）菜单中选择 [Code]（代码）子菜单并且选择 [Stack Trace]（堆栈跟踪），打开 [Stack Trace]（堆栈跟踪）窗口。



Kind	Name	Value
F	sort(long*)	{ FFFF8620 }
F	tutorial()	{ FFFF87DE }
F	main()	{ FFFF877F }
F	PowerON_Reset_PC()	{ FFFF8044 }

图 6.36 [Stack Trace]（堆栈跟踪）窗口

能知道当前 PC 在 sort() 函数内并且 sort() 函数被 tutorial() 函数调用。

再次双击 sort 函数内的行的 [S/W Breakpoints]（S/W 断点）列，解除 S/W 断点。

6.19 结尾

本教程介绍了 E1/E20 仿真器的几个主要特点和 High-performance Embedded Workshop 的使用方法。

能通过使用 E1/E20 仿真器提供的仿真功能进行高度的调试。因此，如果正确地地区和识别硬件和软件问题的发生条件，就能有效地检查这些问题。

7. 使用时的注意事项

7.1 存储器

7.1.1 I/O 寄存器区域

在通过 [Memory]（存储器）窗口等参照并且设定 IO 寄存器区域的内容时，存取长度需要符合对象单片机的硬件手册的指定。因此，建议使用 IO 窗口。

7.1.2 内部闪存 ROM 区

- 如果通过下载除外的操作（从 [Memory]（存储器）窗口的操作或者行汇编）对内部闪存 ROM 区进行编辑，编辑内容就在下次执行操作时被写到闪存 ROM。
- 如果参照被删除的数据闪存 ROM 的内容，就显示由单片机规格决定的不定值。另外，如果通过调试程序对数据闪存 ROM 的内容进行编程，就以 256 字节为单位进行编程，被编程后的区域不会变为不定值。
- 对于 [Internal flash memory overwrite]（内部闪存盖写）页面中没有选定的块，在擦除后进行下载。
- 在下载多个文件时，如果各区域重复在闪存的同一块中，就必须在 [Internal flash memory overwrite]（内部闪存盖写）中选定该块。
- 在仿真时，因为连接 E1/E20 仿真器进行调试的单片机被重复编程，被施加了很大的负荷，所以在调试中使用的单片机不能用于用户批量生产的产品。

7.1.3 内部闪存 ROM 的下载

不对应 [Download Module]（下载模块）对话框中的部分“存取长度”，存取长度必须为“1”、“2”或者“4”。

7.1.4 内部闪存 ROM 的改写

- 在由用户程序改写内部闪存 ROM（程序 ROM 区）后，如果要通过 [Memory]（存储器）窗口等参照被改写的内容，就必须选定 [Configuration properties]（配置属性）对话框中的 [Debugging the program re-writing the internal flash]（调试有改写内部闪存功能的程序）复选框，否则不能参照被改写的内容。
- 在用户程序执行过程中，不能通过 [Memory]（存储器）窗口等参照由用户程序改写的内部闪存 ROM（程序区、数据闪存区）的内容。
在用户程序执行过程中，如果参照被改写的区域，就读不到正确的值。

7.1.5 FCU-RAM、FCU 固件区

不能通过调试程序改写 FCU-RAM 区和 FCU 固件区的内容。

7.1.6 调试程序使用的作业 RAM 区

在改写内部闪存时，固件使用在 [Configuration properties]（配置属性）对话框的 [MCU] 页面中指定的从作业 RAM 地址开始的指定字节数的区域。为了进行存储器内容的压栈和退栈，用户程序也能使用此区域。但是，不能将此区域指定为 DMA 或者 DTC 功能的传送源和传送目标。

必须给作业 RAM 区指定内部 RAM 区，并且不能在用户程序中将内部 RAM 设定为无效。

7.2 [Memory]（存储器）窗口

7.2.1 复制、比较和检索功能

对于复制、比较和检索功能，不能指定 8 字节的数据长度，否则不能正常运行。
复制、比较和检索功能的范围最大为 16MB。

7.2.2 选项菜单

在 [Memory]（存储器）窗口功能的菜单项目中，灰色显示的项目不动作。

7.3 IO 窗口

7.3.1 I/O 寄存器文件的自定义

E1/E20 仿真调试程序能从 [IO] 窗口存取内部 I/O 寄存器。

因为单片机的规格可能会变更，所以在 I/O 寄存器文件的各 I/O 寄存器和目标单片机的硬件手册记载的地址不符时，请在按照硬件手册的记载内容进行修改后使用。能按照 I/O 寄存器文件的格式对 I/O 寄存器进行自定义。

7.3.2 验证

在 [IO] 窗口中，输入值的验证功能无效。

7.4 执行操作

7.4.1 [Go To Cursor]（转至光标）

- [Go To Cursor]（转至光标）的操作不在中途设定断点的位置停止执行。
- 如果将 PC 使用事件数设定为大于等于 4，就不能使用 [Go To Cursor]（转至光标）。

7.4.2 [Run Program]（执行程序）

[Run Program]（执行程序）的操作指定的临时 PC 断点只限最初的 1 点有效。

7.4.3 [Step Out]（步出）执行

对于使用子程序返回指令以外的指令（转移指令）返回到调用源函数的函数，[Step Out]（步出）不能正确地执行。

7.5 复位

7.5.1 由仿真器系统进行的操作和单片机复位的竞争

当引脚复位或者看门狗定时器复位等与仿真器系统对单片机的操作（在 [Memory]（存储器）窗口中参照存储器等）发生竞争时，就显示“A timeout error. The MCU is in the reset state. Is system reset issued?”（发生超时错误。MCU 为复位状态，发行系统复位吗？）的错误信息。

如果单击 [Yes] 按钮，就对仿真器进行初始化，然后停止用户程序的执行。在发行系统复位后跟踪记录也被初始化。如果单击 [No] 按钮，就不对仿真器进行初始化，也不停止用户程序的执行。

能在单击 [Yes] 或者 [No] 按钮后继续调试。

7.5.2 使用跟踪功能时的单片机复位

如果在用户程序执行过程中发生引脚复位或者看门狗定时器复位等，就不能正确记录复位前后的跟踪。

7.5.3 使用实时 RAM 监视功能时的单片机复位

如果在用户程序执行过程中发生引脚复位或者看门狗定时器复位等，就不能保证此后的 RAM 监视运行（可能无法显示正确的值）。

7.5.4 在内部 ROM 无效扩展模式中执行时的单片机复位

在内部 ROM 无效扩展模式中，不能在用户程序执行过程中进行引脚复位和看门狗定时器复位等。

7.6 跟踪功能

7.6.1 能记录的存取

只能记录 CPU 总线的存取，而不能记录 DMAC 或者 DTC 的存取。

7.6.2 跟踪信息

1. 当最新的跟踪信息种类为“BCND”时，显示的转移次数少于用户程序的实际条件转移次数。
2. 在输出“BRANCH”、“DESTINATION”和“BRANCH/DESTINATION”前，不显示“BCND: 条件转移”的跟踪信息。因此，当用户程序的执行结果全部为“BCND: 条件转移”时，不显示跟踪信息。
3. 如果指定跟踪获取开始条件和跟踪获取结束条件，就在每次开始条件和结束条件成立时获取跟踪信息。此时，在 [Trace]（跟踪）窗口中不正确显示反汇编。
4. [Trace]（跟踪）窗口中的反汇编显示对于 1 个转移间隔最多能显示 1024 条指令。如果转移间隔中的指令有 1025 条，就从转移后的指令开始显示 1024 条指令，而不显示第 1025 条指令目。如果转移间隔中的指令大于等于 1026 条，就不能进行反汇编的显示。

7.7 事件功能

7.7.1 能检测的存取

能检测 CPU 总线的存取，而不能检测 DMAC 或者 DTC 的存取。

7.7.2 事件组合

1. 如果将AND或者Sequential（顺序）指定为事件暂停条件的组合，跟踪获取开始条件的事件组合就为OR。如果将AND或者Sequential（顺序）指定为跟踪获取开始条件的事件组合，事件暂停条件的组合就为OR。
2. 在顺序条件中，能指定为数据存取事件的事件只能是复位事件和第1个~第3个事件。
3. 在顺序条件中，能注册为指定了地址范围的数据存取事件的事件只能是第1个事件。
4. 不能将通过次数条件指定为顺序条件的复位事件。

7.7.3 通过次数条件

能指定通过次数条件（[Pass Count]）的事件全部只有1点。

7.7.4 数据存取事件的地址范围条件

能指定地址范围条件的事件全部只有1点。

7.7.5 跟踪获取条件设定

在 [Advanced Settings]（高级设置）中，不能指定窗口跟踪地址范围。

7.7.6 有关事件注册

- 如果在 [Before PC]（执行前PC）事件列表中注册数据存取条件，就忽视此设定。
- 如果在 [Trace Record]（跟踪记录）事件列表中注册执行地址条件，就忽视此设定。

7.8 暂停功能

7.8.1 设定断点时的注意事项

1. S/W 断点通过替换指令来实现。
不能在内部ROM和内部RAM以外的区域中指定断点。
2. 在单步执行中，S/W断点和On-chip断点的指定无效。
3. 在S/W断点或者执行前PC断点停止后，如果从该地址重新开始执行，就在1次单步执行该地址后继续执行，因此失去实时性。
4. 不能在低功耗状态下设定On-chip暂停。
5. 如果在低功耗状态下从[Editor]（编辑器）窗口的On-chip断点列设定On-chip暂停，就在[Output]窗口中显示以下错误信息。

睡眠模式的情况：

“The internal clock is halted because the MCU is in the sleep mode.”（在MCU睡眠模式中内部时钟处于停止状态。）

全模块时钟停止模式、软件待机模式、深度软件待机模式的情况：

“The internal clock is halted because the MCU is in the standby mode.”（在MCU待机模式中内部时钟处于停止状态。）

此后，无法从On-chip断点列设定On-chip暂停。

在这种情况下，请再次打开并且关闭[On-Chip Break]（On-chip暂停）对话框（可以不更改设定）。

7.9 实时 RAM 监视功能

1. 只能监视CPU总线的存取，而不能监视DMAC或者DTC的存取。
2. 如果在用户程序执行过程中发生引脚复位或者看门狗定时器复位等，就不能保证此后的RAM监视运行（可能无法显示正确的值）。
3. 因为根据跟踪数据改写窗口的值，所以实时RAM监视功能与用户程序是在执行中还是在停止中无关，即使在[I/O]窗口或者[Memory]（存储器）窗口中改写值，也不反映到[RAM Monitor]（RAM监视）窗口。
4. 如果通过跟踪抽出功能设定监视范围，就不能对监视范围外的数据（灰色显示的位置）使用监视功能。

7.10 性能测量功能

7.10.1 有关性能测量过程中的复位

如果在性能测量过程中发生复位，即使在复位期间也进行测量。但是，因为用于测量的时钟设定被初始化，所以可能无法正确测量。

7.10.2 性能测量过程中的嵌套限制

以中断 / 异常的发生次数和 RTE/RTFI 指令的发行次数相同并且能一一对应为前提。如果中断 / 异常的发生次数和 RTE/RTFI 指令的发行次数不同，因为无法一一对应，所以不能测量正确的处理周期。

另外，嵌套只能保持 16 层。如果因 17 层以上（含 17 层）的中断和异常而发生嵌套，就不能正确测量。

7.10.3 选定 [Measure the performance only once.]（仅测量一次）时的注意事项

在以下的测量中，即使计数开始事件和计数结束事件成立，如果测量条件一次也没有成立，就不显示测量结果。

- 有效指令发行次数
- 异常和中断接受次数
- 异常接受次数
- 中断接受次数
- 事件成立次数

（例）

- 在测量区间中异常或者中断一次也没有发生。
- 在测量区间中事件成立一次也没有发生。

7.10.4 有关事件的处理

如果在性能测量中开始事件成立 2 次，即使结束事件没有成立也停止测量，在下次开始事件成立时重新开始测量。

7.11 下载功能

7.11.1 有关存取长度

在下载时能指定下载模块的存取长度。

通过指定存取长度，以指定的存取长度对存储器进行下载（写）。

请注意：能指定的下载模块的存取长度因各区域而不同。

内部 ROM/RAM 区只能指定“1”、“2”或者“4”（不能指定“8”），但是含有外部闪存 ROM 区数据的下载模块的存取长度只能选择“1”。

另外，在存取外部 RAM 区时，优先在启动时的 [Configuration Properties]（配置属性）对话框的外部区域中按各区域指定的总线宽度。

当需要给存取长度指定“2”时，请在启动时的 [Configuration Properties]（配置属性）对话框中对外部 RAM 区设定“16bit”总线宽度，将外部闪存 ROM 区的下载模块分为其他下载模块，并且给不含外部闪存 ROM 区的下载模块指定“2”。

另外，对外部闪存 ROM 区使用 USD 文件指定的存取长度。

详细内容请参照以下 URL 刊载的“External Flash Definition Editor”的手册。

<http://www.renesas.com/efe>

例如，在以下程序中，要将外部 RAM 的存取长度设定为“2”的情况。

内部 RAM
外部 RAM
外部 ROM
内部 ROM

请按以下方法分为 (a) 和 (b) 的模块进行下载。

(a) 将存取长度设定为“2”的下载模块（外部 RAM 的存取长度优先启动时的“16bit”总线宽度）

内部 RAM
外部 RAM
内部 ROM

(b) 将存取长度设定为“1”的下载模块

外部 RAM

7.11.2 内部 ROM 无效状态的下载

请注意：如果在下载到内部闪存时显示“An error has occurred in erasing of the internal flash ROM. The flash ROM may have been degraded.”（发生内部闪存 ROM 的擦除错误。闪存 ROM 可能已经老化。）的错误信息，就可能是在内部 ROM 无效状态（SYSCR0.ROME=0）下进行了下载。

7.12 外部闪存的下载

- 能注册扇区数不超过4096的闪存。
不保证扇区数超过4096的闪存编程。
- USD文件指定的脚本符合能从HEW调试程序的命令行执行的memory_fill命令。
请注意：不能指定部分选项。

能在脚本中使用的memory_fill命令格式如下：

```
mf <start> <end> <data> [<mode>]
```

<start> 起始地址

<end> 结束地址

<data> 数值

<mode> BYTE (1字节)

WORD (2字节)

LONG (4字节)

省略：和BYTE相同

※ USD文件能指定的脚本如下：

- “下载前执行脚本”
- “下载后执行脚本”
- “使用外部RAM前执行脚本”
- “使用外部RAM后执行脚本”
- 不能对外部地址空间设定S/W断点。
- 不能从[Memory]（存储器）窗口改写外部闪存。
- 必须给下载模块的存取长度选择“1”。
- 如果在下载到外部闪存时将外部RAM用作作业RAM，就必须设定和CPU相同的字节序。

7.13 执行时间

状态栏和 [Status] (状态) 窗口中显示的执行时间是程序开始执行到停止执行之间的时间, 舍去 100 μ s 未满足的值。另外, 不能正确显示单步 (Step) 执行、跨步 (Setp-over) 执行和步出 (Step-out) 执行时的执行时间。

7.14 Start/Stop (启动 / 停止) 功能

对于 Start/Stop (启动 / 停止) 功能, 如果不从 Start 功能转移到用户程序或者 Stop 功能不停止, 就显示 “A timeout error has occurred in START/STOP function processing. Is system reset issued?” (在启动 / 停止功能处理时发生超时错误。发行系统复位吗?) 的错误信息。

如果单击 [Yes] 按钮, 就对仿真器进行初始化, 然后停止用户程序的执行。在发行系统复位后跟踪记录也被初始化。如果单击 [No] 按钮, 就不对仿真器进行初始化, 而停止用户程序的执行。

能在单击 [Yes] 或者 [No] 按钮后继续调试。

7.15 监视功能

如果在用户程序执行过程中注册符号, 就在 [Watch] (监视) 窗口的 [Value] 列中显示 “Not available now.”。在此符号被注册的状态下, 不能在用户程序执行过程中将全符号的自动更新设定为有效。

7.16 其他

7.16.1 改写闪存时的寄存器值

请注意: 如果因下载程序或者设定 S/W 断点或者从存储器窗口设定值等而改写闪存, 闪存相关寄存器的值就被调试程序改写。

7.16.2 有关 DMAC 和 DTC

在用户程序停止执行的状态下向单片机的内部闪存 ROM (传送源) 发生 DMAC 或者 DTC 请求时, 不能进行以下操作:

- 对单片机的内部闪存 ROM 区设定 S/W 断点。
- 在 [Memory] (存储器) 窗口或者 [Command line] (命令行) 窗口中写单片机的内部闪存 ROM 区。
- 对单片机的内部闪存 ROM 区进行下载。

7.16.3 锁定位的解除

如果在设定锁定位后下载用户程序, 就解除锁定位, 与下载前的状态无关。

如果在启动仿真器时显示的 [Configuration Properties] (配置属性) 对话框的 [Internal flash memory overwrite] (内部闪存盖写) 选项卡中有选定, 就在选定的地址区域为用户程序的下载对象区域时解除该块的锁定位。

当没有选定时, 就在用户程序下载结束时解除该块的锁定位。

附录

附录 1. 菜单一览表

GUI 菜单一览表如附表 1.1 所示。

附表 1.1 GUI 菜单一览表

菜单	菜单选项	快捷键	工具栏按钮	备注
View (视图)	Difference (差分)			显示 [Difference] (差分) 窗口。
	Map (映像)			显示 [Map Section Information] 窗口。
	Command Line (命令行)	Ctrl+L		显示 [Command Line] (命令行) 窗口。
	TCL toolkit (TCL 工具包)	Ctrl+Shift+		显示 [Console] 窗口。
	Workspace (工作空间)	Alt+K		显示 [Workspace] 窗口。
	Output (输出)	Alt+O		显示 [Output] 窗口。
	Status bar (状态栏)	Alt+A		切换状态栏的显示和不显示。
	Disassembly (反汇编)	Ctrl+D		显示 [Disassembly] (反汇编) 窗口。
CPU	Registers (寄存器)	Ctrl+R		显示 [Registers] (寄存器) 窗口。
	Memory... (存储器 ...)	Ctrl+M		显示 [Memory...] (存储器 ...) 窗口。
	IO	Ctrl+I		显示 [IO] 窗口。
	Status (状态)	Ctrl+U		显示 [Status] (状态) 窗口。
	RAM Monitor (RAM 监视)			显示 [RAM Monitor] (RAM 监视) 窗口。
	Debug Console (调试控制台)			显示 [Debug Console] (调试控制台) 窗口。
Event (事件)	On-chip Break (On-chip 暂停)			显示 [On-chip Break] (On-chip 暂停) 对话框。
	Trace Conditions (跟踪条件)			显示 [Trace Conditions] (跟踪条件) 对话框。
	Performance (性能)			显示 [Performance] (性能) 对话框。
Symbol (符号)	Labels (标签)	Ctrl+Shift+A		显示 [Labels] (标签) 窗口。
	Watch (监视)	Ctrl+W		显示 [Watch] (监视) 窗口。
	Locals (局部)	Ctrl+Shift+W		显示 [Locals] (局部) 窗口。

附表 1.1 GUI 菜单一览表 (续)

菜单	菜单选项		快捷键	工具栏按钮	备注
View (视图)	Code (代码)	Trace (跟踪)	Ctrl+T		显示 [Trace] (跟踪) 窗口。
		Stack Trace (堆栈跟踪)	Ctrl+K		显示 [Stack Trace] (堆栈跟踪) 窗口。
	Performance (性能)	Performance Analysis (性能分析)	Ctrl+Shift+ P		显示 [Performance Analysis] (性能分析) 窗口。
Debug (调试)	Synchronized Debugging... (同步调试 ...)				显示能设定同步调试的 [Synchronized Debugging...] (同步调试 ...) 对话框。
	Debug Sessions... (调试对话 ...)				显示调试对话一览表以及能进行添加和删除等的 [Debug Sessions...] (调试对话 ...) 对话框。
	Debug Settings... (调试设置 ...)				显示设定调试时的条件和下载模块等的 [Debug Settings...] (调试设置 ...) 对话框。
	Reset CPU (CPU 的复位)				对目标单片机进行复位并且将 PC 设定为复位向量地址。
	Go (执行)		F5		从当前 PC 开始执行用户程序。
	Reset Go (复位执行)		Shift+F5		对目标单片机进行复位并且从复位向量地址开始执行用户程序。
	Free Go (自由执行)				忽视断点, 执行用户程序。
	Go To Cursor (转至光标)				用户程序从当前 PC 开始执行到文本光标位置。
	Set PC To Cursor (在光标位置设定 PC 值)				在文本光标位置设定 PC。
	Run... (执行 ...)				显示能设定执行时的 PC 和 PC 断点的 [Run Program] (执行程序) 对话框。
	Display PC (显示 PC)		Ctrl+Shift+Y		在 [Editor] (编辑器) 窗口显示 PC 位置。
	Step In (步进)		F11		执行用户程序的 1 个块后停止。
	Step Over (跨步)		F10		执行用户程序的 1 个块后停止, 但是在调用子程序时不进入子程序。
	Step Out (步出)		Shift+F11		将用户程序执行到当前函数的结束位置。
	Step... (单步 ...)				显示能设定步进的 [Step Program] (步进程序) 对话框。
	Step Mode (单步模式)		Auto (自动)		在 [Editor] (编辑器) 窗口有效时只对 1 行的源行进行单步执行, 在 [Disassembly] (反汇编) 窗口有效时以汇编语言指令为单位进行单步执行。
			Assembly (汇编)		以汇编语言指令为单位进行单步执行。
			Source (源)		只对 1 行的源行进行单步执行。
	Halt Program (暂停程序)		Esc		停止用户程序的执行。
Initialize (初始化)				断开并且重新启动调试平台。	

附表 1.1 GUI 菜单一览表 (续)

菜单	菜单选项		快捷键	工具栏按钮	备注
Debug (调试)	Connect (连接)				连接调试平台。
	Disconnect (断开)				断开调试平台。
	Save Memory... (保存存储器 ...)				将存储器内容保存到文件 (.bin、.hex、.mot 格式)。
	Verify Memory... (验证存储器 ...)				将存储器内容和保存的文件进行验证。
	Download Modules (下载模块)				加载目标程序。
	Unload Modules (卸载模块)				卸载目标程序。
Setup (设置)	Radix (基数)	Hexadecimal (16 进制数)			显示或者输入数值时的基数默认设定为 16 进制数。
		Decimal (10 进制数)			显示或者输入数值时的基数默认设定为 10 进制数。
		Octal (8 进制数)			显示或者输入数值时的基数默认设定为 8 进制数。
		Binary (2 进制数)			显示或者输入数值时的基数默认设定为 2 进制数。
	Emu-lator (仿真器)	Device setting... (器件设置 ...)			显示进行有关目标单片机设定的 [Initial Settings] (初始设置) 对话框。
		System... (系统 ...)			显示进行有关仿真器系统全体设定的 [Configuration Properties (配置属性)] 对话框。
		Start/stop Function Setting... (启动 / 停止功能设置)			显示 [Start/Stop function setting] (启动 / 停止功能设置) 对话框。

附录 2. High-performance Embedded Workshop 的注意事项

(1) 有关建立装入模块后移动源文件位置的注意事项

如果在建立装入模块后移动源文件，就可能显示 [Open]（打开）对话框，以便在调试建立的装入模块时指定源文件。请选择对应的源文件，按 [Open]（打开）按钮。

(2) 源级执行功能

- 源文件

在程序窗口中显示了不对应装入模块的源文件时，如果该源文件和装入模块对应的源文件同名，就在程序窗口中显示地址而不能操作该文件。

- 单步执行

也会进入标准 C 库程序等。在返回到上层函数时，必须使用 Step Out（步出）。

如果在菜单 [Setup] -> [Options]（设置 -> 选项）的 [Debug]（调试）页面中选定 “not to step into addresses where no debugging information exists.”（不步进到不存在调试信息的地址），就不进入没有调试信息的库程序。

在 for 语句或者 while 语句中，1 次单步执行不移动到下一行。要移动到下一行时，请再进行 1 次单步执行。

(3) 有关存取文件过程中的操作

在下载装入模块过程中，或者在 [Memory]（存储器）窗口中的验证处理过程中，或者在 [Save]（保存）和 [Trace]（跟踪）窗口中的保存处理过程中，不能进行其他操作。否则，可能无法正确地执行文件存取处理。

(4) 监视功能

- 优化时的局部变量

根据生成的目标码，可能无法正确地显示通过优化选项编译的源的局部变量。请显示 [Disassembly]（反汇编）窗口，确认生成的目标码。

可能不存在指定了局部变量的分配区域。此时，显示如下：

例) 变量名为 asc。

```
asc Not available now.
```

- 变量名的指定

如果不用变量名指定符号名（函数名），就不显示内容。

例) 函数名为 main。

```
main Not available now.
```

(5) 命令行界面

- 批文件

如果在批文件执行过程中显示 “Not currently available”（当前不可用），就必须插入 sleep 命令。

sleep 时间因工作环境而不同，必须调整。

例) 在执行 memory_fill 时显示 “Not currently available”（当前不可用）的情况。

```
sleep d'3000
```

```
memory_fill 0 ffff 0
```

- 命令文件中的文件指定

根据命令文件的指定方法，可能会移动当前目录。建议用绝对路径记述命令文件中的文件指定，以便不受当前目录移动的影响。

例) FILE_LOAD C:\Workspace\Tutorial\E1E20\RX600\Tutorial \Tutorial_LittleEndian
\Debug_RXxxx_E1_E20_SYSTEM\tutorial.abs

(6) Motorola S 格式的文件加载

High-performance Embedded Workshop 不支持记录末尾只为“CR 码”（0Dh）的 Motorola S 格式文件。要加载 Motorola S 格式文件时，必须使用记录末尾有“CR 码和 LF 码”（0D0Ah）的文件。

(7) 有关程序执行过程中的 [Register]（寄存器）窗口操作的注意事项

在程序执行过程中，不能在 [Register]（寄存器）窗口中更改寄存器的值。即使更改也不改变寄存器的内容。

(8) 有关 RUN-TIME 显示的注意事项

在状态栏和 [Status]（状态）窗口中显示用户程序的执行时间，因为使用 E1/E20 仿真器内部的 32 位计数器，所以舍去 100 μ s 未满足的值。另外，不能正确显示单步（Step）执行、跨步（Setp-over）执行和步出（Step-out）执行时的执行时间。

(9) memory test 功能

本产品不支持通过从 [Memory]（存储器）菜单中选择 [Test]（测试）而使用的 memory test 功能。

(10) 闪存编程器模式

- 当连续对多个单片机进行编程时，必须开关目标系统的电源。
- 在闪存编程器模式中，不能使用用户程序下载以外的调试功能。

(11) 有关反汇编显示的注意事项

在反汇编模式中显示 [Editor]（编辑器）窗口时，没被定义的汇编指令代码被显示为“???”。

修订记录	E1/E20 仿真器 用户手册附加文档（连接时的注意事项）
------	-------------------------------

Rev.	发行日	修订内容	
		页	修订处
3.00	2011.03.04	—	初版发行

E1/E20 仿真器
用户手册附加文档（连接时的注意事项）

Publication Date: Rev.3.00 Mar 4, 2011

Published by: Renesas Electronics Corporation

**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

E1/E20 仿真器



瑞萨电子株式会社

R20UT0399CJ0300