To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

User's Manual

M3T-PD308F V.3.00
M3T-PD30F V.2.00

User's Manual

Emulator Debugger for PC7501 System

Renesas Electronics
www.renesas.com

Rev.1.00 2003.05

Keep safety first in your circuit designs!

- Renesas Technology Corporation and Renesas Solutions Corporation put the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation, Renesas Solutions Corporation or a third party.
- Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation and Renesas Solutions Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corporation and Renesas Solutions Corporation by various means, including the Renesas home page (http://www.renesas.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation and Renesas Solutions Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation or Renesas Solutions Corporation for further details on these materials or the products contained therein.

For inquiries about the contents of this document or product, fill in the text file the installer generates in the following directory and email to your local distributor.

¥SUPPORT¥Product-name¥SUPPORT.TXT

Renesas Tools Homepage http://www.renesas.com/en/tools

Contents

Setup

1.Starting the debugger

1.1 Features

The PD308F and PD30F have the following functions.

1.1.1 Real-Time RAM Monitor Function

This function allows you to inspect changes of memory contents without impairing the realtime capability of target program execution. The Emulator PC7501 System has 4 Kbytes of RAM monitor area which can be located in any contiguous address location or in 16 separate blocks comprised of 256 bytes each.

1.1.2 Break Functions

- **Software Break**
  This function causes the target program to stop immediately before executing the instruction at a specified address. Up to 64 breakpoints can be set. If multiple breakpoints are set, the program breaks at one of the breakpoints that is reached.(Note)
- **Hardware Break**
  This function causes the target program to stop upon detecting a data read/write to memory, instruction execution, or the rising/falling edge of the input signal fed from an external trace cable. The contents of events that can be set vary with each target MCU. Specified hardware break events can be used in one of the following combinations:
  - Break when all specified break points are effected.(And)
  - Break when all specified break points are effected simultaneously.(And(Same Time))
  - Break when any one of the specified break points is effected.(Or)
  - Break on transition in state to a break state.(State Transition)
- **Protect Break**
  This function causes the target program to stop upon detecting a data write to the ROM area or an access to an unused area (read/write or instruction execution).
- **Address Interrupt Break**
  This function stops the target program immediately before executing an instruction at a specified address. This function is realized by using the MCU's address interrupt interrupt. When debugging the MCU's address interrupt interrupt function, deselect the Enable the Address Match Interrupt Break Function check box on the MCU tab of the Init dialog box. That way, the address interrupt break function is disabled.

1.1.3 Real-Time Trace Function

This function records a target program execution history. Up to 256K cycles of execution history can be recorded. This record allows inspecting the bus information, executed instructions, and source program execution path for each cycle.

1.1.4 Time Measurement Function

This function measures the minimum, maximum, and average execution time and the number of executions performed in a specified interval. Measurements can be taken in up to four intervals at the same time.

1.1.5 Coverage Function

This function records the addresses executed (accessed) by the target program (C0 coverage). This function helps to keep track of unexecuted addresses after the program has stopped running. Use of this coverage measurement function in the test process makes it possible to keep track of the test items that have been omitted.

1.1.6 Real-Time OS Debugging Function

This function debugs the realtime OS-dependent parts of the target program that uses the realtime OS. This function helps to show the status of the realtime OS and inspect a task execution history, etc.

1.1.7 GUI Input/Output Function

This function simulates the user target system's key input panel (buttons) and output panel on a window. Buttons can be used for the input panel, and labels (strings) and LEDs can be used for the output panel.

1.2 About the Emulator PC7501

The PC7501 is a full-bus trace emulator suitable for M16C family's operating clock frequencies exceeding 20 MHz. When combined with an emulation probe suitable for the target MCU, it helps to configure an emulator system matched to each MCU used.

1.2.1 Communication method

The supported communication methods are as follows.

I/F	Emulator
	PC7501
USB	O
LAN	O
LPT	O

Depending on communication methods, it is necessary to set up some items before the PDxx can be started. Please see "Before starting the debugger".

1.2.2 Function table

The supported functions are as follows.

Function	Emulator	
	PC7501	PC4701U(reference)
RAM Monitor	4K bytes (256bytes x 16blocks) area	1K bytes area
SW Break	64 points	
HW Break	8 points	6 points
Address Interrupt Break	4 points*	-
Real-Time Trace	256K Cycles	32K Cycles
C0 Coverage	256K bytes x 32blocks area	256K bytes area
Time Measurement	Go to Stop / 4 points interval	
Protect Break	Access Protect Break	

*Depends on the target MCU used.

1.3 Before starting the debugger

Before starting the debugger, check the following contents:

1.3.1 Communication method by emulator

The supported communication methods vary with the type of emulator used.

I/F	Emulator
	PC7501
USB	O

LAN	O
LPT	O

1.3.1.1 USB Interface

- The supported host computer OS is Windows Me/98/2000/XP. USB communication cannot be used in any other OS.
- Compliant with USB Standard 1.1.
- Connections via USB hub are not supported.
- By connecting the host computer and the PC7501 emulator with USB cable, it is possible to install the supported device drivers using a wizard (The PDxx that supports USB
- connections must be installed before this installation can be performed.). See "1.3.3.1 USB communication with PC7501" for details.
- The necessary cable is included with the PC7501 emulator.

1.3.1.2 LAN Interface

- The IP address, etc. must be set in the emulator before it can be connected in a LAN.
- The PC7501 emulator in a LAN can be connected to the PC7501s on another network connected to the LAN via a router. See "1.3.3.2 LAN communication with PC7501" for details.
- The emulators PC7501 uses LAN cable (10BASE-T only) generally available on the market.
- The host computer and the emulator can be connected directly. See "2.1.2.3 Setting of the LAN Interface" for details.

1.3.1.3 LPT Interface

- This communication uses the host computer's parallel (printer) interface.
- The necessary cable is included with the PC7501 emulator.
- Four communication modes are supported that include ECP, EPP, Byte, and Nibble. Communication modes that can be supported depend on the host computer's BIOS settings. (Communication modes may not always be used even when they are supported by BIOS.)

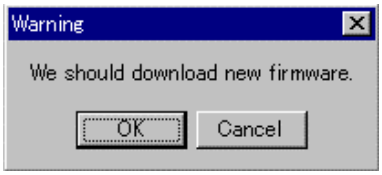
1.3.2 Download of Firmware

It is necessary to down-load the firmware which corresponds to connected Emulation Probe when the debugger is started to the emulator.

- You have changed your emulation probe.
- You have setup PDxx for the first time.
- You have upgraded emulator debugger PDxx.

Press the system reset switch within **two seconds** after powering up the PC7501 to establish the maintenance mode.
PDxx searches the version of the firmware downloaded to the emulator at start. Also when the firmware downloaded to the emulator is of old version, a mode which drives PDxx to download firmware is set.

When PDxx gets started while the emulator is set in the mode which drives PDxx to download firmware forcedly, the following dialog is opened at start.
Click the OK button to download the firmware.



ATTENTION

- Before the firmware can be downloaded by the PC7501 in a LAN connection, the IP address, etc. must first be registered in the PC7501. (Setup method)

1.3.3 Setting before emulator starts

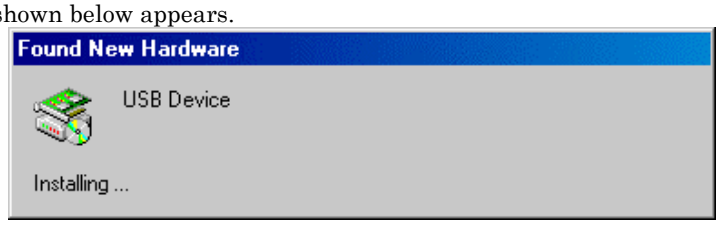
1.3.3.1 USB communication with PC7501

Connection of USB devices is detected by Windows' Plug & Play function. The device driver needed for the connected USB device is automatically installed. For details, see "Installing USB Device Driver".

<< Install of USB device driver >>

The USB devices connected are detected by Windows' Plug & Play function. The installation wizard for USB device drivers starts after the device had been detected. The following shows the procedure for installing the USB device drivers.

1. Connect the host computer and the PC7501 emulator with USB cable.
2. Set the PC7501 emulator's communication interface switch to the "USB" position. Then turn on the power to the emulator.
3. The dialog box shown below appears.



Go on following the wizard, and a dialog box for specifying the setup information file (inf file) is displayed. Specify the musbdrv.inf file stored in a location below the directory where the PDxx is installed (e.g., c:¥mtool¥pdxx¥drivers).

ATTENTION

- Before the USB device drivers can be installed, the PDxx you use must already be installed. Install the PDxx first.
- USB communication can be used only in Windows Me/98/2000/XP, and cannot be used in any other OSs.
- When using Windows 2000/XP, a user who install the USB device driver need **administrator** rights.
- During installation, a message may be output indicating that the device driver proper musbdrv.sys cannot be found. In this case, specify the musbdrv.sys which is stored in the same directory as is the musbdrv.inf file.

1.3.3.2 LAN communication with PC7501

Before the emulator can be connected in a LAN, the IP address, etc. must first be registered in the emulator. For the PC7501 emulator in default settings, the utility "setip.exe" included with the PDxx may be used to set the IP address, etc. in the emulator.

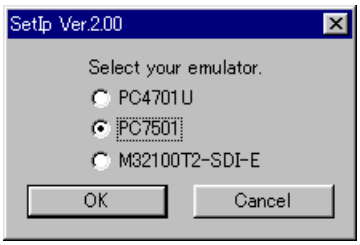
<< Setting of the LAN Interface using the SETIP.EXE >>

The utility "SETIP" included with the PDxx may be used to set the IP address, etc. in the PC7501 emulator while in default settings. SETIP detects the PC7501s in default settings that are connected to the same network. SETIP is stored in a location below the directory where the PDxx is installed (e.g., c:¥mtool¥pdxx¥utility). The file name is "setip.exe".

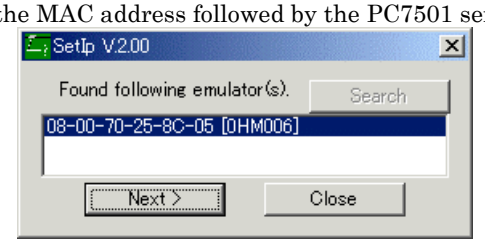
To register the IP address in the PC7501, follow the procedure described below.

1. Connect the PC7501 emulator with LAN cable to the same network (same subnet) as the host computer is connected.

2. Set the PC7501 emulator's communication interface switch to the "LAN" position. Then turn on the power to the emulator.
3. Start SETIP. When SETIP has started up, the dialog box shown below appears, choose PC7501, click the OK button.

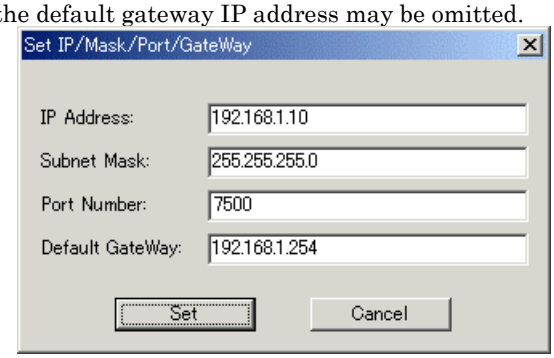


4. Click the OK button, and showing information on the PC7501 connected to the network. (This information consists of the MAC address followed by the PC7501 serial number.)



To register the IP address, click the Next button. To cancel registration, click the Close button. If not displayed, check whether the communication interface switch is set correctly and after temporarily turning off the power, turn it back on again. Then click the Search button.

5. Click the Next button, and the dialog box shown below appears. Set the IP address, subnet mask, port number, and default gateway IP address. When using the PC7501 on the same network's same subnet mask, the default gateway IP address may be omitted.



Use any 4-digit number to specify the port number. (Enter that number when starting the PDxx.) For details about the contents of the IP address, subnet mask, and default gateway to be specified, contact your network administrator.

6. Click the Set button on the dialog box. The IP address, etc. that have been set are registered in the PC7501 emulator. When registered correctly, the dialog box shown below appears.

After checking the contents of the dialog box, click the OK button.

7. Temporarily turn off the power to the PC7501 emulator and turn it back on again. The registered IP address becomes effective after the emulator is powered up again.

ATTENTION

- If multiple PC7501s in default settings are connected on the same network, only the first PC7501 detected is displayed.
- The PC7501s which have had an IP address already set cannot be detected by SETIP. In such a

case, connect to the emulator through another communication interface and re-register the IP address from the Init dialog box that appears. For details on how to set IP addresses from the Init dialog box, see "Setting of the LAN Interface."

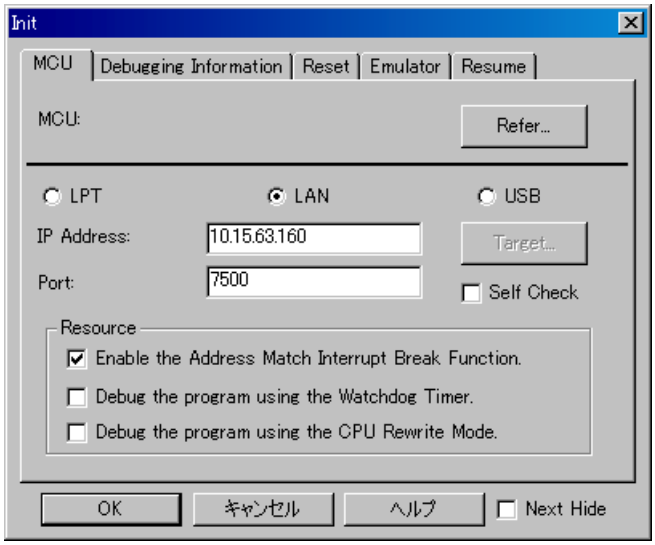
1.4 Starting the Debugger

Click the Windows start button, and then select menu:

Program (P) -> [RENESAS-TOOLS] -> [PDxxF V.x.xx Release x] -> [PDxxF]

2. Setup Debugger

The Init dialog box is provided for setting the items that need to be set when the debugger starts up. The contents set from this dialog box are also effective the next time the debugger starts. The data set in this dialog remains effective for the next start.



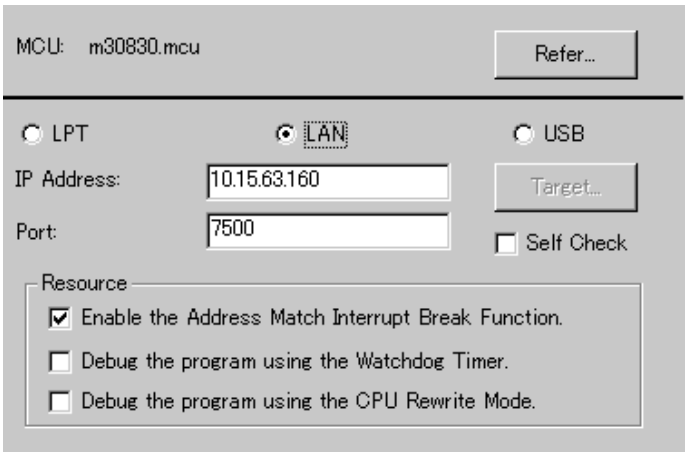
Tab Name	Contents	
	PD308F	PD30F
MCU	exist	exist
Debug Information	exist	exist
Reset	exist	exist
Emulator	exist	exist
Resume	exist	exist

To keep the Init dialog closed next time the debugger is started, check "Next Hide" at the bottom of the Init dialog. You can open the Init dialog using either one of the following methods:

- After the debugger gets started, select Menu - [Environment] -> [Init...].
- Start PDxx while holding down the Ctrl key.

2.1 MCU Tab

The specified content becomes effective when the next being start.



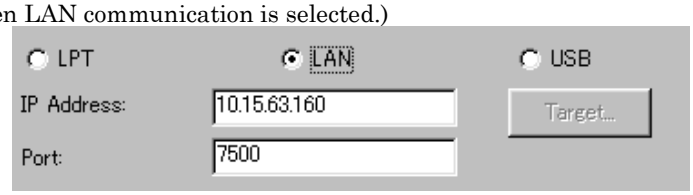
2.1.1 Specifying the MCU file

Click the "Refer" button. The File Selection dialog is opened. Specify the corresponding MCU file.
An MCU file is saved under the directory in which PDxx is installed.
(For example:c:¥mtool¥pdxx¥mcufiles).

- An MCU file contains the information specific to the target MCU.
- The specified MCU file is displayed in the MCU area of the MCU tab.

2.1.2 Setting of the Communication Interface

The displayed data varies depending on the specified communication interface. (The figure below shows the data when LAN communication is selected.)



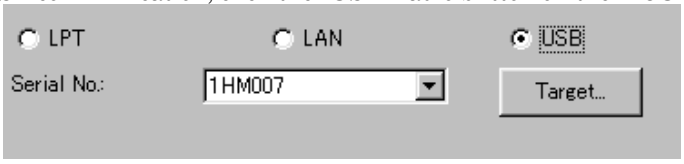
The available communication interface varies depending on the products.
The following shows the setting for each communication interface.

- USB Interface
- LPT Interface
- LAN Interface

2.1.2.1 Setting of the USB Interface

USB communication uses the personal computer's USB interface. It is compliant with USB 1.1.

<< Setting of the USB Interface >>
Before USB communication can be performed, the computer must have a dedicated device driver installed in it. For details on how to install USB device drivers, see "1.3.3.1 USB communication with PC7501"
For connection by USB communication, click the "USB" radio button on the MCU tab.

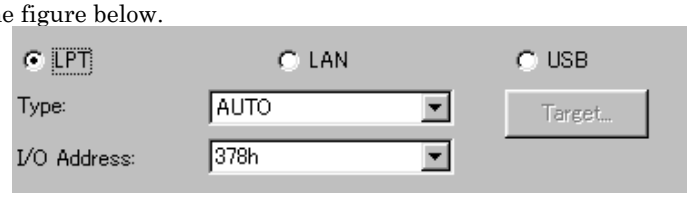


The currently USB-connected emulators are listed in the Serial No. area. Select the serial No. of the emulator you want to connect.

2.1.2.2 Setting of the LPT Interface

LPT communication uses a parallel interface (printer interface) of the personal computer.

<< Setting of the LPT Interface >>
To set the LPT communication, click the "LPT" radio button of the MCU tab in the Init dialog. The setting looks like the figure below.



Specify the mode for data transfers in the Type field.

- The LPT interface has four modes for data transfers, Nibble, Byte, ECP, and EPP. Their modes are documented in the IEEE-1284 standard. The mode possible to use depends on the PC with the PDxx.
- When selecting AUTO, the PDxx detect the LPT interface at the start and select a mode possible to use which give better performance automatically. In some PC, the PDxx cannot detect the most suitable mode. Check the mode which the LPT interface in the PC have and select it from Nibble, Byte, ECP or EPP, if cannot.
- Specify the I/O address of the used LPT port in the I/O Address field.
- Start the BIOS setup program of the PC for checking the mode possible to use. How to start and use the BIOS setup program depends on each PC, so refer the manuals of the PC.

Display of BIOS Setup	Communication Mode
SPP, Standard Parallel Port, Output Only	Nibble
Bidirectional, Bi-directional	Byte
ECP, Extended Capabilities Port	ECP
EPP, Enhanced Parallel Port	EPP

The address displayed in the parallel port base address field is the I/O address. Specify the I/O address set in the BIOS setup program, in the I/O Address field. (The following addresses are possible to be specified)

- 378h
- 278h

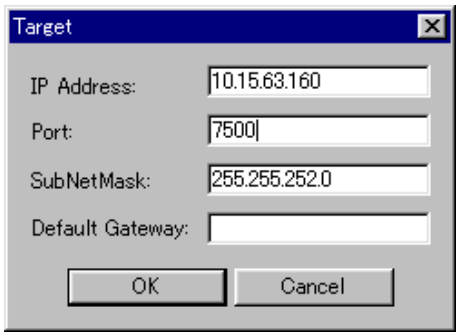
ATTENTION

The C compiler made by IAR also uses this printer (parallel) port.
When using PDxx and the emulator PC7501 in the ECP mode on the LPT communication, a problem that the data cannot be complied by the IAR C compiler will arise.
If this happens, take one of the following countermeasures:

- Connect PDxx to the emulator PC7501 in any mode other than the ECP mode.
- Start compilation when PDxx has been terminated.

2.1.2.3 Setting of the LAN Interface
LAN communication uses a LAN interface of the personal computer.
Before using LAN, you must register the emulator IP address, port number and subnet mask to the emulator itself (Otherwise, LAN is not available).
Then, set LAN communication.

<< Setting the IP Address and Subnet Mask >>
Start PDxx using other communication method. After it gets started, select Menu - [Environment]-> [Init ...] to open the Init dialog. Then, click the Target button of MCU tab. The Target dialog will be opened.

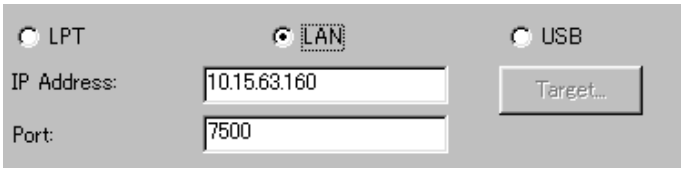


Specify the emulator IP address in the IP Address field, port number in the Port field, and subnet mask in the SubNetMask field. (The emulator IP address must be registered in the network environment in advance.)
When the PC7501 it is used, the Default Gateway area becomes effective. Please specify the IP address of the default gateway. When the PC7501 it is used on the identical sub net mask of identical network, it is possible to omit the IP address of the default gateway.

- Specify the IP address, subnet mask and Default Gateway in decimal byte by byte, by separating every 4 bytes with a period. For details on the IP address and subnet mask, consult with your network manager.
- A port number set in the Port field is used to identify the communication process of the server (emulator) in LAN (TCP/IP) communications. Specify the port number which has been set in the emulator in hexadecimal. (Do not add a prefix which shows a base.)

Click the "OK" button in the Target dialog. The Target dialog is then closed and the Init dialog appears again. Click the "OK" button.
Then, exit from PDxx.

<< Setting of the LAN Interface >>
To set the LAN communication, click the "LAN" radio button of the MCU tab in the Init dialog. The setting looks like the figure below.

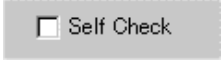


Specify the IP address of the connected emulator in the IP address field.
Specify the IP address, in bytes, in decimal. Delimit each 4 bytes with a period. The port No. is the ID No. for the communication process of the server (emulator) on the LAN (TCP/IP).
Specify, in hexadecimal (Don't describe prefix which shows a cardinal number), the port No. set on the emulator.

<< LAN connection by couple 1 with emulator >>
Emulators PC7501 can be connected by LAN (TCP/IP) to a commercially available LAN card inserted in a PC by using a cross conversion cable for 10BASE-T (also commercially available). A HUB is not necessary in this case.
The cross conversion cable for 10BASE-T converts the male connector of the 10BASE-T of a straight LAN cable that is included with the emulators to that of a cross LAN cable.
Connect a cross conversion cable to the male connector of the 10BASE-T of the straight LAN cable connected to the emulator; then, connect the male connector of the cross conversion cable to the LAN card.
The LAN communications can be set up the same way as normal one.

2.1.3 Executing Self-Check

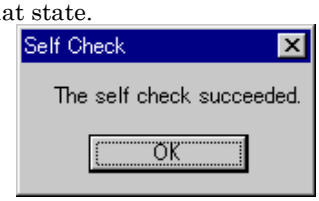
Specify this option to execute self-check* on the emulator when the debugger starts up.



Be sure to select the above check box only when you want to perform self-check at startup. Specify this option in the following cases:

- When the firmware cannot be downloaded
- When although the firmware is successfully downloaded, the debugger does not start
- When the MCU goes wild or something is wrong with the trace results and you want to check whether the emulator is operating normally.

Select the check box to close the Init dialog box. After connecting to the emulator and confirming the firmware, the debugger will immediately start self-check on the emulator. (Self-check takes about 30 seconds to 1 minute.)
If an error is found in this self-check, the debugger displays the content of the error and is finished.
When the self-check terminated normally, the dialog box shown below is displayed. When you click OK, the debugger starts up directly in that state.

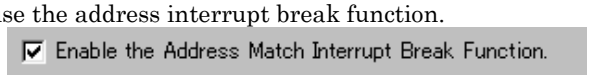


This specification is effective only when the debugger starts up.

* Self-check refers to the function to check the emulator's internal circuit boards for memory condition, etc. Refer to the user's manual of your emulator for details about the self-check function.

2.1.4 Using/unusing the address interrupt break function

Specify whether or not to use the address interrupt break function.



- **To use the address interrupt break function (default)**
  Select the check box shown above. In this case, the address interrupt break function is used by the emulator, and cannot be used in the user program.
- **Not to use the address interrupt break function**
  Deselect the check box shown above. In this case, the address interrupt break function can be used in the user program.

The contents set here are reflected at only startup time.

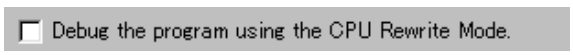
2.1.5 Using/unusing the watchdog timer (only PD308F)

Specify whether or not to use the watchdog timer. (By default, the watchdog timer is unused.)

When debugging the target system that uses a watchdog timer, select the check box shown above.

2.1.6 Choosing to use or not to use CPU rewrite mode

Specify whether or not you want to use CPU rewrite mode. (By default, CPU rewrite mode is unused.)



Select the above check box when you are debugging the target system that uses CPU rewrite mode.
This specification can only be set or changed when you start PD.

Supplementary explanation

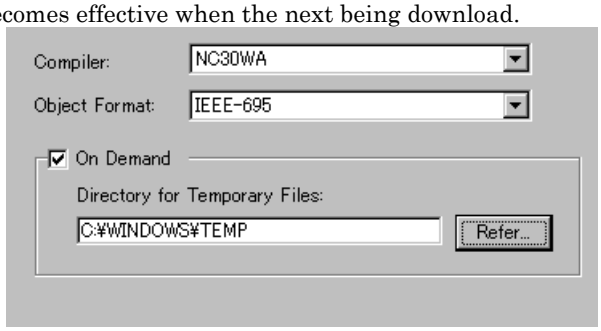
When debugging in CPU rewrite mode is enabled, the following limitations apply:

- Address match breakpoints cannot be set. (PD308F)

- No software breaks can be set in the internal ROM area. (PD308F, PD30F)
- The command Come cannot be executed in the internal ROM area. (PD308F, PD30F)

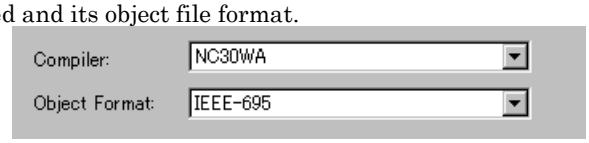
2.2 Debugging Information Tab

The specified content becomes effective when the next being download.



2.2.1 Specifying the compiler used and its object format

Specify the compiler used and its object file format.



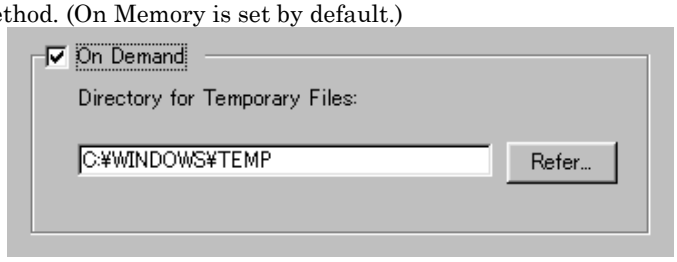
- **Compiler**
  Select the compiler used in your application. (By default, this is the C Compiler of our company.)
- **Object Format**
  Select the format of object files output by the compiler used.

2.2.2 Specify the Storing of Debugging Information

To save the debugging information, two methods are available: On Memory which saves the information in memory and On Demand which saves the information in the temporary file.

On Memory	Allows high-speed process because of use of memory.
On Demand	Minimizes use of memory.

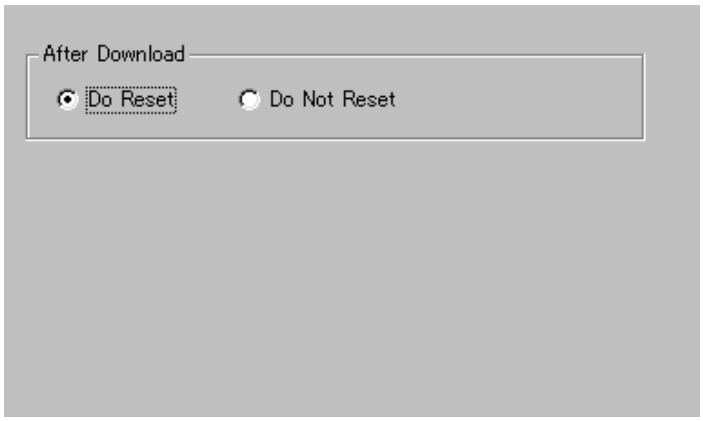
Select the saving method. (On Memory is set by default.)



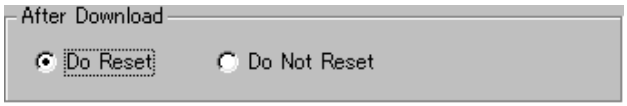
To select On Demand, specify the temporary file saving directory in the Temp Dir field.
If you do not specify the directory, the system creates a temporary file in the directory in which the downloaded load module file is saved.

2.3Reset Tab

The specified content becomes effective when the next being start.



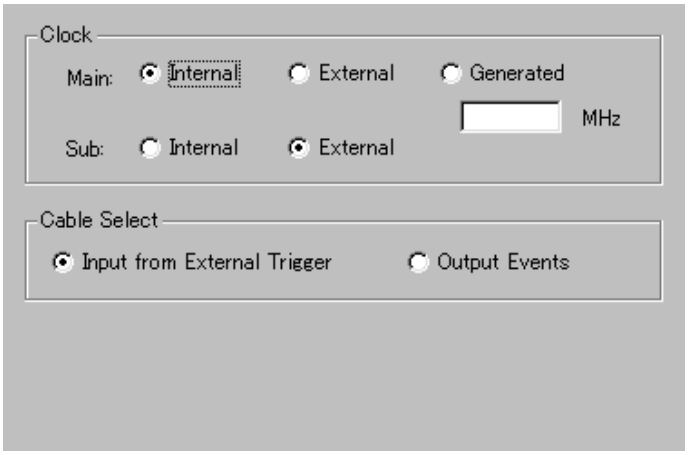
2.3.1 Target Reset after Down-loading



Specify whether you want to reset the target immediately after the target program is downloaded.

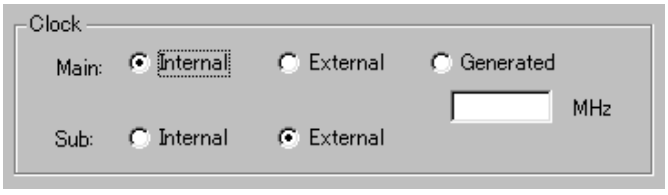
Do Reset	Reset.(Default)
Do Not Reset	Not Reset.

2.4 Emulator Tab



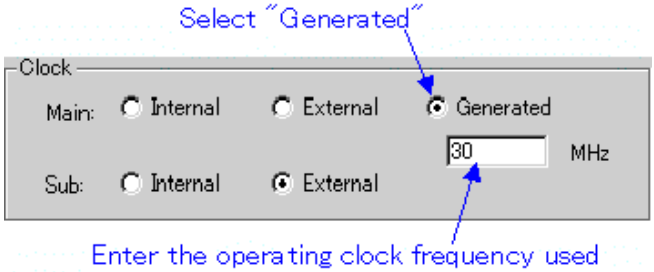
2.4.1 Specify the Target Clock

Change the setting by synchronizing with the clock used by the target microcomputer. (Internal is set by default.)



Select Internal to set the internal clock, and External to set the external clock.

In addition to Internal and External, the main clock can be chosen to be a clock which is internally generated in the emulator according to a specified clock frequency (user-defined clock).
To use a user-defined clock, select Generated and enter the clock frequency used in the Generated frequency input area.



- The Generated frequency value can be entered in the range of 1.0 MHz to 99.9 MHz in 0.1 MHz increments.*
- The Generated frequency value can only be set or changed when you start PD.
- Unless any value is set in the Generated frequency input area, the option "Generated" cannot be selected.

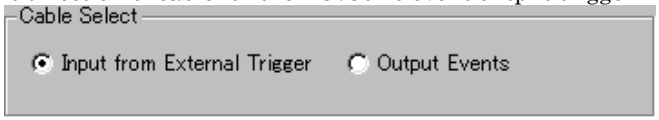
The specified content becomes effective when the next being start.

Note

* The accuracy of the clock frequency is ±5%. We recommend that final evaluation be made after installing the resonator or oscillator module whose frequency is matched to the actually used Internal clock.

2.4.2 Selecting event output/trigger input cable

Select the input/output direction of cable for the PC7501's event output/trigger input.

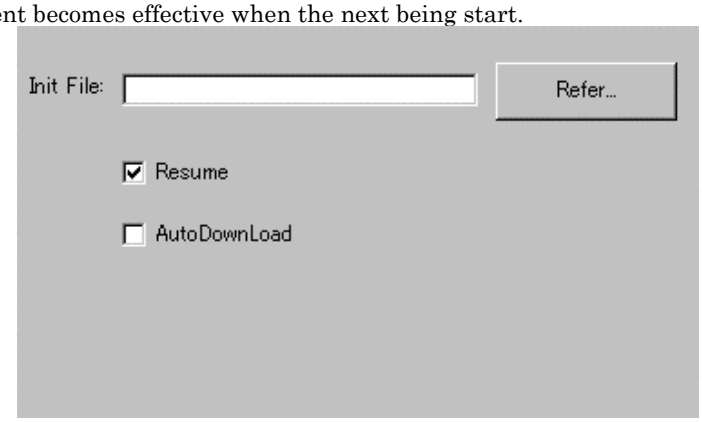


- To enter external triggers from the cable, select "Input from External Trigger."
- To output events to the cable, select "Output Events."

The contents set here are reflected at only startup time. If settings are changed in the Init dialog box after starting the debugger, the change is not reflected (until the PDxx is restarted).
Note that "Input from External Trigger" is selected at startup. (The contents that were set the last time the debugger started have no effect.)

2.5Resume Tab

The specified content becomes effective when the next being start.



2.5.1 Automatically Execute the Script Commands

To automatically execute the script command at start of Debugger, click the "Refer" button to specify

the script file to be executed.

By clicking the "Refer" button, the File Selection dialog is opened.
The specified script file is displayed in the "Init File:" field.
To disable auto-execution of the script command, erase a character string displayed in the "Init File:" field.

2.5.2 Restore the Window Status

To restore the window status (window position, window size) after the previous debugger program is terminated, check the "Resume" check box. (Resume is ON by default.)

2.5.3 Re-download a Load Module

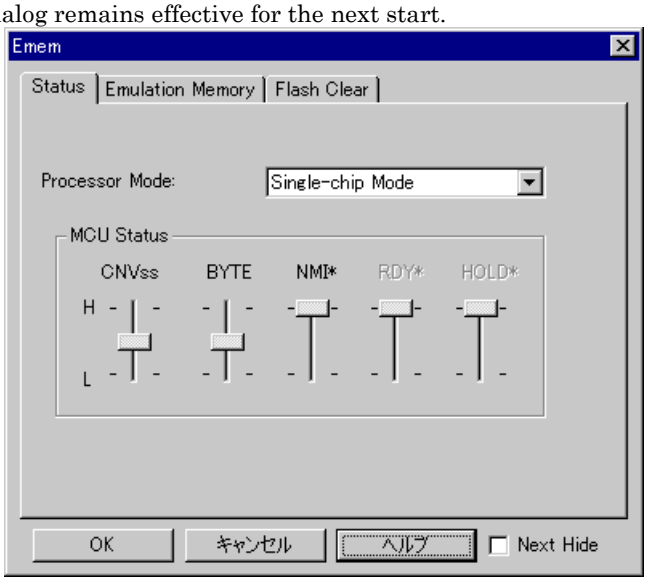
To re-download a load module (target program), check the "AutoDownLoad" check box. (Re-download is OFF by default.)

3. Set the Target Information

Setting information of the target for debugging is different according to the product.

3.1 PD308F

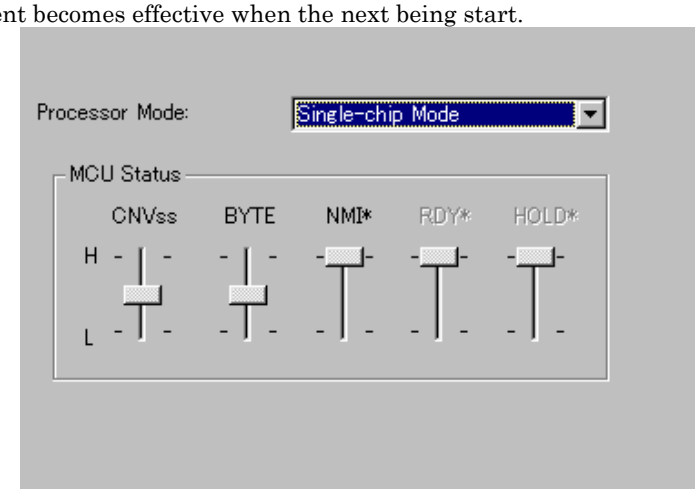
Set the target to be debugged in the Emem dialog, which is opened after closing the Init dialog opened after PDxx gets started.
The data set in this dialog remains effective for the next start.



To keep the Emem dialog closed next time the debugger is started, check "Next Hide" at the bottom of the Emem dialog.
You can open the Emem dialog using either one of the following methods: After the debugger gets started, select Menu - [Environment] -> [Emem...].

3.1.1 Status Tab (PD308F)

The specified content becomes effective when the next being start.



3.1.1.1 Select the Processor Mode

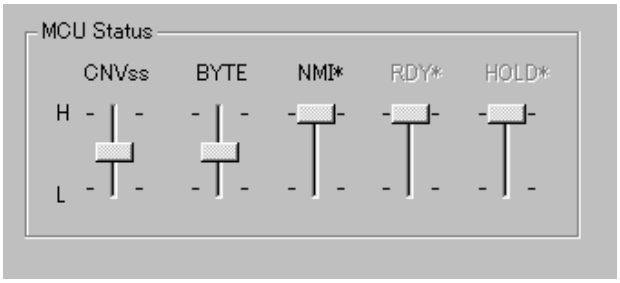
Specify the processor mode for the target system.

Either the following can be specified.

- **Single-chip Mode**
- **Memory Expansion 8 Bit**
- **Memory Expansion 16 Bit**
- **Microprocessor 8 Bit**
- **Microprocessor 16 Bit**

3.1.1.2 Inspecting the MCU status

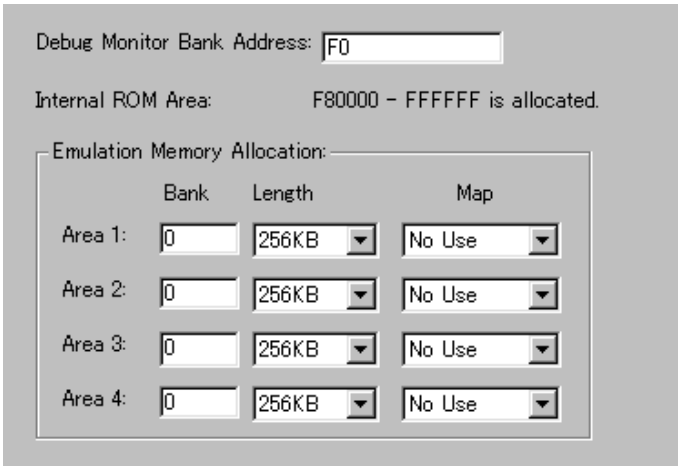
Clicking this tab displays the status of each MCU pin. It allows to check whether the MCU pin status matches the processor mode to be set.



If the slider is at the middle position, it means that the value is indeterminate.

3.1.2 Emulation Memory Tab(PD308F)

The specified content becomes effective when the next being start.



3.1.2.1 Debug monitor's bank address settings

This product allocates a 64-Kbyte contiguous address area as the emulator's work area for use by the debug monitor.
Specify any bank that the target system does not use. The debug monitor uses a 64-Kbyte area from the start address of the specified bank.
(Example: If the specified bank is "F0," then the debug monitor uses a 64-Kbyte area beginning with address F000000h.)

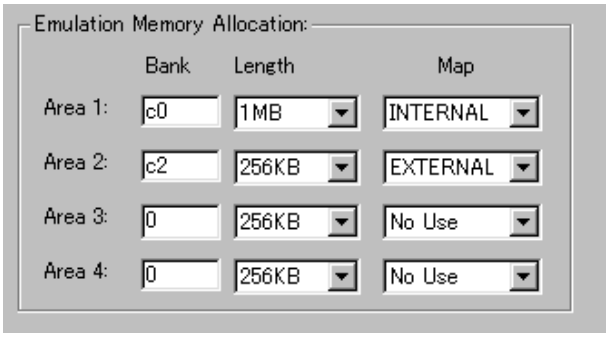
- The bank specified here cannot have its contents referenced or set. The contents of this area when displayed in the Memory window or the Program/Source window's disassemble display mode may not be correct.
- The following bank addresses cannot be specified:
- MCU internal resources (e.g., SFR and RAM areas)
- DRAM area and multiplexed area
- Interrupt vector area

3.1.2.2 Automatic emulation memory allocation for the internal ROM

When single-chip or memory extension mode is selected, emulation memory is automatically allocated to the internal ROM area.
The automatically allocated internal ROM address range is displayed in this field.

3.1.2.3 Emulation memory allocation for an extended area

When memory extension or microprocessor mode is selected, emulation memory can be allocated to the extended area to be debugged (in up to four areas).
Here, allocate memory for the debug target area and specify its mapping information.



Follow the procedure described below.

Bank (Set bank address)	Specify the bank address of the debug target area to be allocated in hexadecimal. If specified as C0, C00000h is the start address of the debug target area.
Length (Specify size of area)	Specify the size of the debug target area (256 bytes or 1 Mbytes). If Length is specified to be "256 bytes," banks 00, 04, 08, and up to FC (every four banks) are specified for Bank; if Length is specified to be "1 Mbytes," banks 00, 10, 20, and up to F0 (every 16 banks) are specified for Bank.
Map (Specify area map)	Specify the mapping information ("Internal" or "External") for the specified area. If no area is specified, select "No Use." • **Internal** The area specified to be "Internal" is mapped into the internal area (emulation memory). • **External** The area specified to be "External" is mapped into the external area (external resources in the target system).

- Areas for which "No Use" is selected for Map and those not specified here are mapped into external areas. If compared to the case where areas are explicitly specified to be "External," the only difference is a download speed. (Downloading into these areas is slower than downloading into the areas specified to be "External.")

- The internal ROM area is automatically mapped into the emulation memory. Therefore, there is no need to set here.
- Be careful that the debug areas will not overlap.
- Make sure the total size of the specified debug target areas does not exceed the emulation memory size of the emulation pod used. The size of emulation memory that can be allocated varies with each emulation pod. (Consult the user's manual of your emulation pod.)

The setting of the emulation memory area varies depending on the specified processor mode.

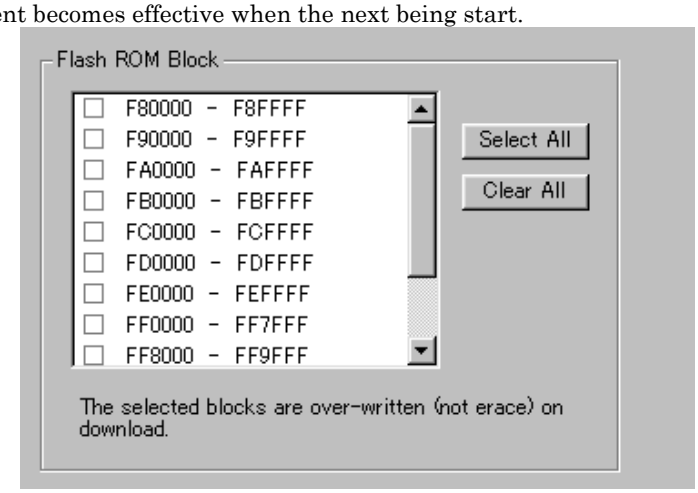
- **Single-chip Mode**
  You do not need to specify the area to be assigned as the emulation memory. The internal ROM area is automatically mapped into the emulation memory. The address range of the automatically mapped area is displayed in the Internal ROM Area: field.
- **Memory Expansion Mode(8bit and 16bit)**
  If you have an area to be assigned as the emulation memory in addition to internal ROM area, specify it specify it separately. The internal ROM area is automatically mapped into the emulation memory. The address range of the automatically mapped area is displayed in the Internal ROM Area: field.
- **Microprocessor Mode(8bit and 16bit)**
  Specify the area to be assigned separately. (There is no area which is automatically assigned.)

ATTENTION

- The mapping setting data specified using the Map command is not reflected to the Emem dialog box.
- Set the emulation memory areas in the order of usage priority. The emulation memory areas to be set by the Map command are numbered, ignoring the unused (Not Use) areas. Accordingly, the emulation memory areas set in the Emem dialog box and the emulation memory area numbers set by the Map command will be mismatched.

3.1.3 Flash Clear Tab(PD308F)

The specified content becomes effective when the next being start.



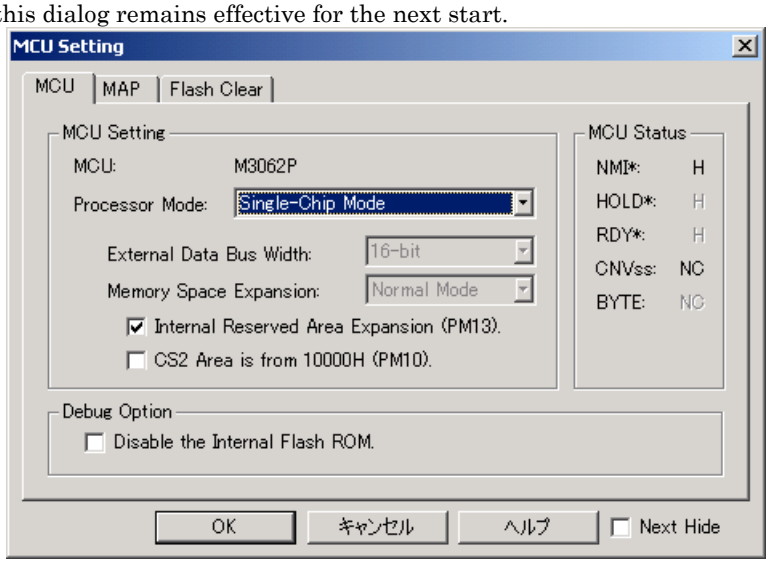
3.1.3.1Setting to clear the MCU's internal flash ROM

Specify whether or not to clear the contents of the MCU's internal flash ROM when downloading the target program or data.
The MCU's internal flash ROM is displayed block by block in the list view.

- The blocks whose check marks are turned on do not have their flash contents cleared when downloading. The memory contents in places not overwritten by downloading remain intact.
- The blocks whose check marks are turned off have their flash contents cleared when downloading.
- Pressing the Select All button keeps all blocks from being cleared when downloading.
- Pressing the Clear All button clears all blocks when downloading.

3.2 PD30F

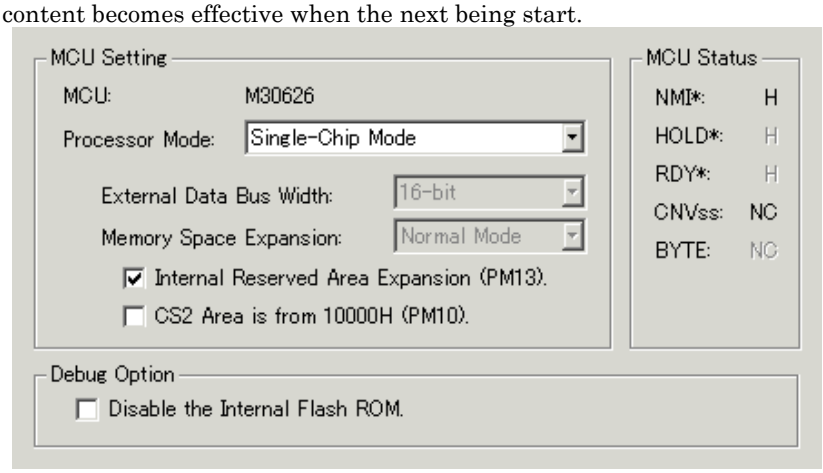
Set the target to be debugged in the MCU Setting dialog, which is opened after closing the Init dialog opened after PDxx gets started.
The data set in this dialog remains effective for the next start.



If you do not want the MCU Setting dialog box to be displayed the next time you start the debugger, select the check box "Next Hide" at the bottom of the MCU Setting dialog box. To open the MCU Setting dialog box, choose [MCU Setting...] from the [Environment] menu after starting the debugger.

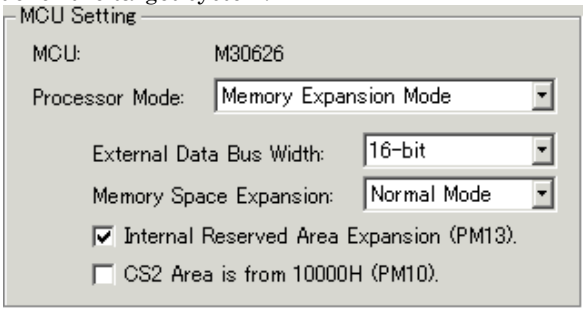
3.2.1 MCU Tab(PD30F)

The specified content becomes effective when the next being start.



3.2.1.1 Select the Processor Mode

Specify the processor mode for the target system.



Either the following can be specified.

- **Single-chip Mode**
- **Memory Expansion Mode**
- **Microprocessor Mode**

Also, you need to specify the following information according to the processor mode you've selected.

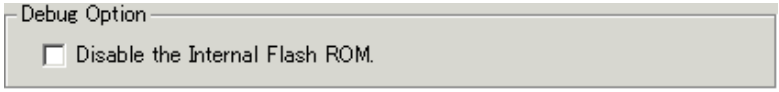
- **External Data Bus Width**
  If you selected memory extension or microprocessor mode, specify "16-bit" or "8-bit" for the external bus width. Make sure the specified external bus width matches settings of the BYTE pin.
- **Memory Space Expansion**
  If you selected memory extension or microprocessor mode, specify whether or not to use the memory space expansion facility. Select "4MB Mode" if you want to use the memory space expansion facility or "Normal Mode" if you do not.
- **Internal Reserved Area Expansion**
  Specify whether or not to extend the internal reserved area. Click this check box if you want the internal reserved area expansion bit (PM13) to be set to 1 while in use.
- **CS2 Area is from 10000H**
  Click this check box if you want the CS2 area select bit (PM10) to be set to 1 while in use.

3.2.1.2 Inspecting the MCU status

Clicking this tab displays the status of each MCU pin. It allows to check whether the MCU pin status matches the processor mode to be set.

"NC" means that the value is indeterminate.

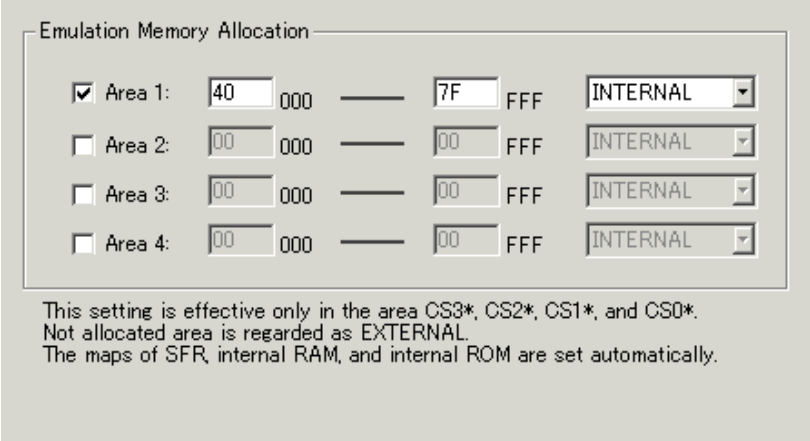
3.2.1.3 Setting the Debug Option



Click this check box if you do not want the program to be downloaded into the MCU's internal flash ROM. You may need to select this option when the internal flash ROM cannot be rewritten such as when using the MCU at low voltage. Initially, this option is deselected.

3.2.2 MAP Tab (PD30F)

The specified content becomes effective when the next being start.

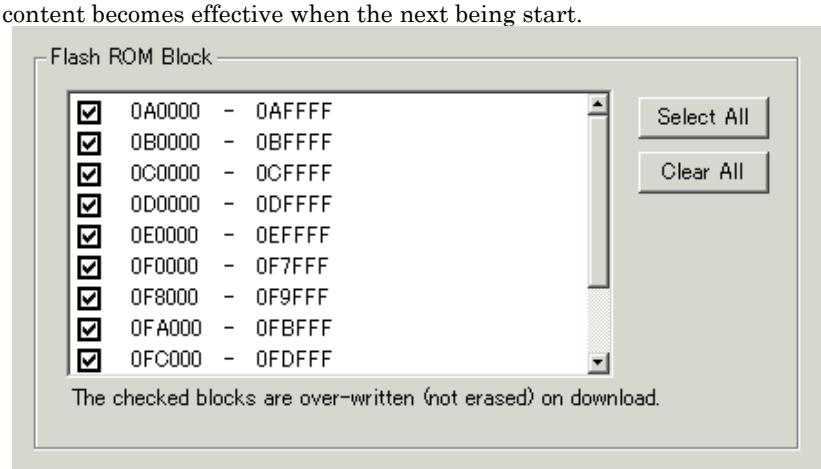


3.2.2.1Emulation memory allocation

Set the memory area in 4 KB units into which you want the emulation memory to be mapped. Four of such memory areas can be set.
The emulation memory is mapped into the areas marked "Internal." The unselected areas and the areas which have nothing specified are allocated to external areas.
Note that MAP settings are effective for only the areas CS3*, CS2*, CS1*, and CS0*. The SFR, internal ROM, and internal RAM areas are automatically mapped.

3.2.3 Flash Clear Tab (PD30F)

The specified content becomes effective when the next being start.



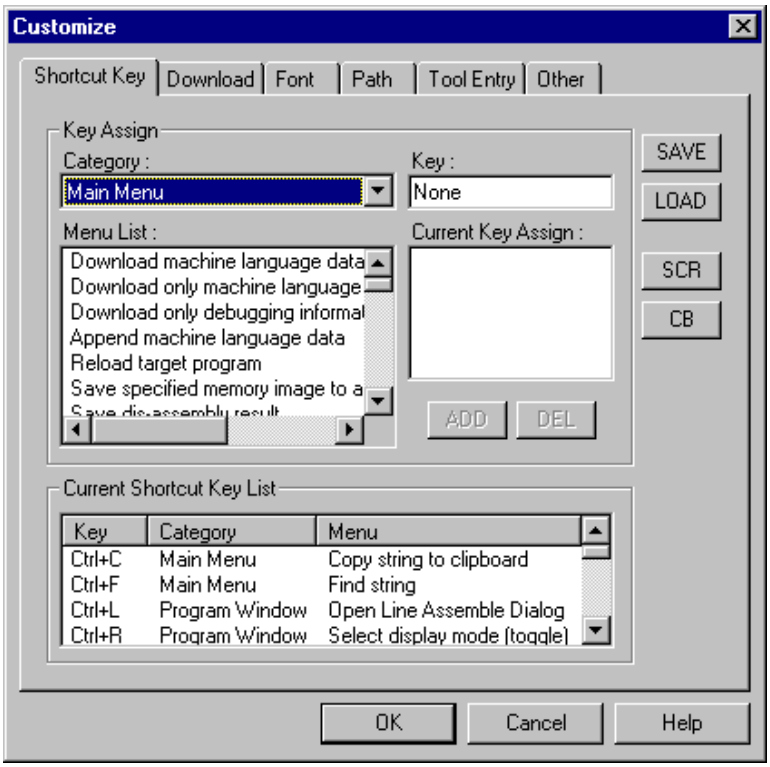
3.2.3.1 Setting to clear the MCU's internal flash ROM

Specify whether or not to clear the contents of the MCU's internal flash ROM when downloading the target program or data. The MCU's internal flash ROM is displayed block by block in the list view.

- The blocks whose check marks are turned on do not have their flash contents cleared when downloading. The memory contents in places not overwritten by downloading remain intact.
- The blocks whose check marks are turned off have their flash contents cleared when downloading.
- Pressing the Select All button keeps all blocks from being cleared when downloading.
- Pressing the Clear All button clears all blocks when downloading.

4. Environmental Setting of Debugger

Specify debugger environment setting in the Customize dialog. You can open this dialog by selecting menu - [Environment] -> [Customize...]. The data set in this dialog remains effective for the next start.



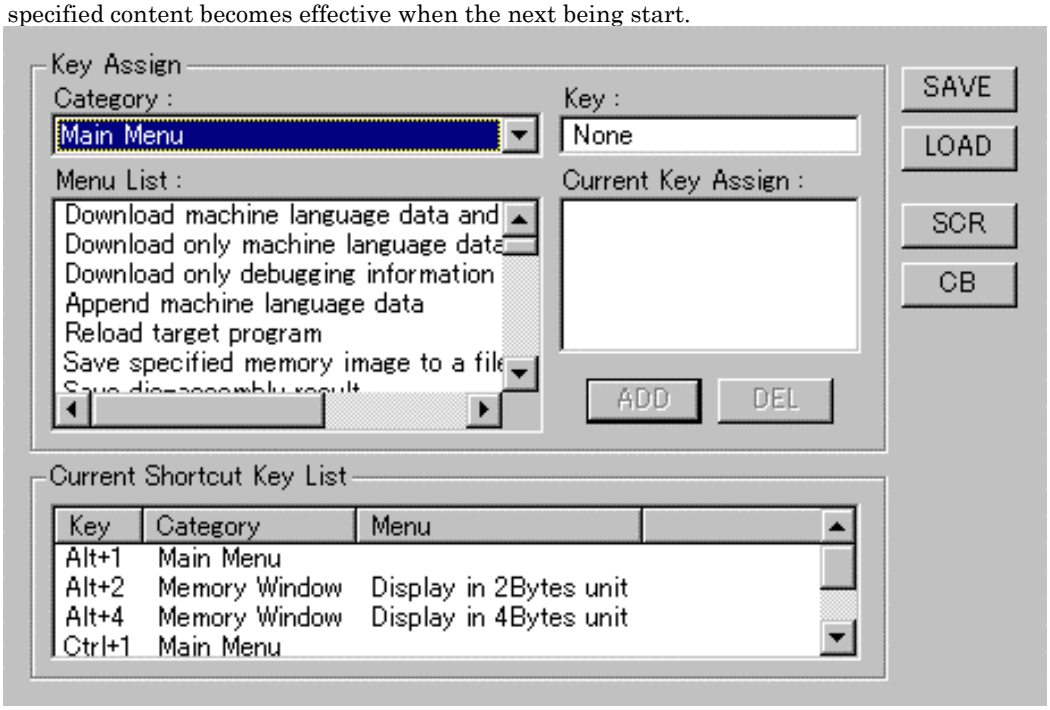
Please click the tab name about details.

Tab	Contents
Shortcut Key	● Register the menus to the shortcut keys
Download	● Automatically Down-load of the Load Module ● Setting the number of load module download histories
Font	● Specify the font ● Specify the Displaying Tab Width
Path	● Specify the Search Path of Source Files ● Specify the Saving Directory of Information File
Tool Entry	● Secify the make file ● Specify the Editor
Other	● Display the Termination Confirmation Dialog ● Debugger Forced Ending when Error Occurs ● Target Continuance Execution when Debugger Ends ● Display the Absolute Path of Source File ● Control the Display Mode Switching of Program Window ● Execution History of Script Command ● Number of Label Displays of Address Setting Area

You can also customize the buttons in the tool bar. To customize the buttons, see "Customizing the Tool Bar".

4.1Shortcut Key Tab

The specified content becomes effective when the next being start.



4.1.1 Register the menus to the shortcut keys

You can register the menus to the shortcut keys.
You can also register execution of the script file and opening of the Custom window to the shortcut keys.

- Assignable shortcut keys are any one key*, or combination of Shift/Ctrl/Alt keys + any one key*. *Any one key covers the following.
  - Alphabet key
  - Numeric key
  - Function key
  - Symbol key(",", "@", ":" etc.)
- When the shortcut key information is changed, the following dialog appears when exiting from the Customize dialog (when clicking the "OK" button). asking you whether you want to save the changed data or not.

When you save the changed data, the data is automatically loaded at the next start of PDxx.

<< Specification of Shortcut Key Tab >>

Key Assign Group

Category combo box

Displays the menu category. The enabled menus in the selected category are displayed in the Menu List list box.

- The category name [Main Menu] indicates all the menus except the option menus of each window.
- When the category of the window name is selected, the menu options available in that window become enabled.
- When the category name [Custom Window] is selected, the registered Custom windows become enabled. (PDxxF do not support this function.)
- When the category name [Script Command] is selected, the registered script commands become enabled.

Menu List list box

Lists the menus enabled in the menu category selected in the Category combo box. The listed menus are sorted in the alphabet order.

Key Edit box

Specifies the shortcut key to be assigned to the menu selected in the Menu List list box.

Current Key Assign list box

Displays the shortcut key to be assigned to the menu selected in the Menu List list box.

ADD button

Enables the shortcut key specified in the Key Edit box.

DEL button

Disables the shortcut key selected from the Current Key Assign list box.

Current Shortcut Key List group

Lists the preset shortcut keys.

SAVE button

Saves the shortcut key information displayed in the Current Shortcut Key List group in a file.

LOAD button

Reads the shortcut key information from a file.

SCR button

Registers a script to be assigned to the shortcut key.

CB button

Registers the Custom window to be assigned to the shortcut key.
(PDxxF do not support this function.)

<< Registering the shortcut key >>

1. Select the category of the menu to be registered in the Category combo box in the Key Assign group. The menus available for the category are displayed in the Menu List list box.
2. Select the menu to be registered from the Menu List list box and click the Key exit box. PDxx is now waiting for the entry of shortcut key.
3. Press the shortcut key to be assigned. The content of the shortcut key is displayed in the Key edit box.
4. Click the ADD button below the Current Key Assign list box.

<< Deleting the shortcut key >>

1. Select the shortcut key to be deleted using one of the following methods:
   - Select the shortcut key from the list in the Current Shortcut Key List group.
   - Select the Menu List list box in the Key Assign group.
2. Click the DEL button in the Current Shortcut Key List group.

<< Saving/reading the shortcut key >>

To use (save/read) the assigned shortcut key information separately, you need to specify the file.
Click the SAVE button and specify the file name.

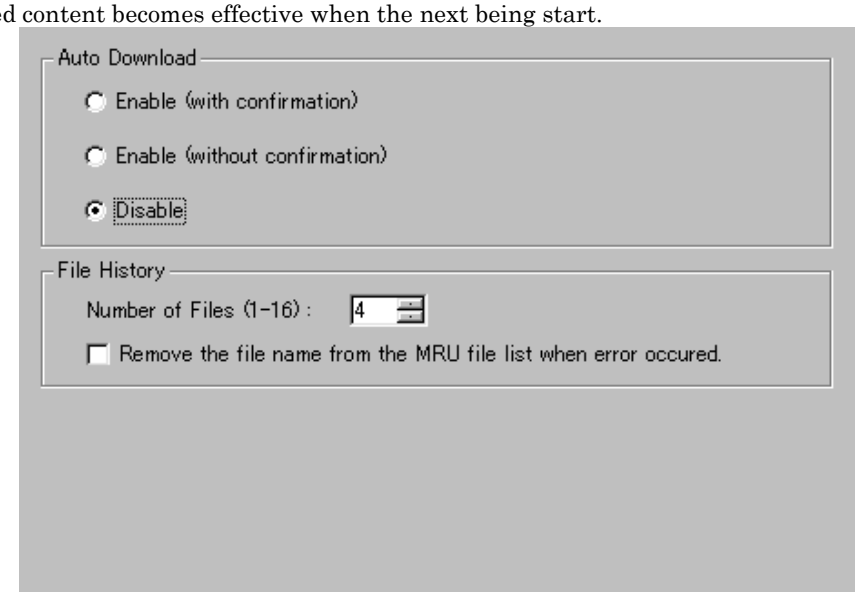
To read the shortcut key information, click the LOAD button and specify the file name.
All of the registered shortcut key information is deleted.

ATTENTION

- You cannot assign the same shortcut key to multiple menus. If you register the assigned key, the information on the previously assigned shortcut key is overwritten.
- The shortcut key is enabled only for the active window. If two or more same windows are opened, the shortcut key is not reflected to all of them.
- If the same menu (Change Font, etc.) exists between the windows, the menu is enabled in all the windows having that menu.

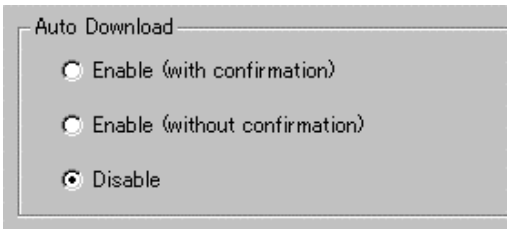
4.2 Download Tab

The specified content becomes effective when the next being start.



4.2.1 Automatically Down-load of the Load Module

When the downloaded load module is updated by re-compile assemble, the file can be auto-downloaded.
The load module is updated at timing when it is operated by a command of execution group (Go, Step, etc.).

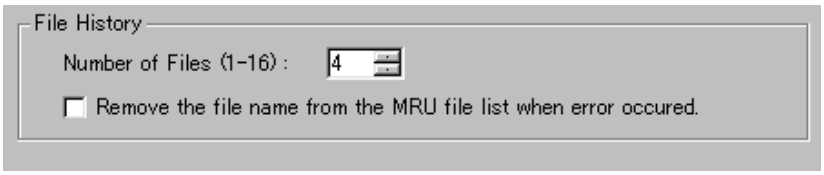


In the Auto Download group, select any one of the following ("Disable" is selected by default.)

Enable (with confirmation)	Asks for confirmation at auto-download.
Enable (without confirmation)	Does not ask for confirmation at auto-download.
Disable	Does not auto-download the load module file.

4.2.2 Setting the number of load module download histories

You can set the number of load module download histories ("4" is set by default).
Specify the number of histories in the File History Number field in the File History group. You can specify the number from 1 to 16.



Furthermore, if the debugger fails to redownload a file from the download history, you can choose whether or not to leave the history of that file. (By default, the file is left.) If you want to delete the history, select the check box shown above.

4.3 Font Tab

The specified content becomes effective when the next being start.

4.3.1 Specify the font

Specify the default font for the characters displayed by PDxx.

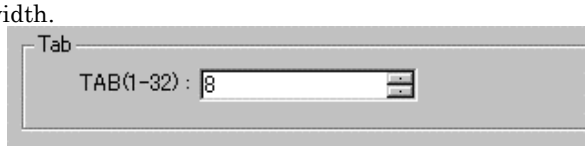
Click the "Font..." button. The font selection dialog opens. Enter the font and font size.

Note

You can set the font independently in each window. With the target window active,
select [Option]->[Font...] from the menu in the PDxx Window to open the font selection dialog.

4.3.2 Specify the Displaying Tab Width

In a window which displays the source files (Program Window, Coverage source window, etc.), you can specify the display tab width.



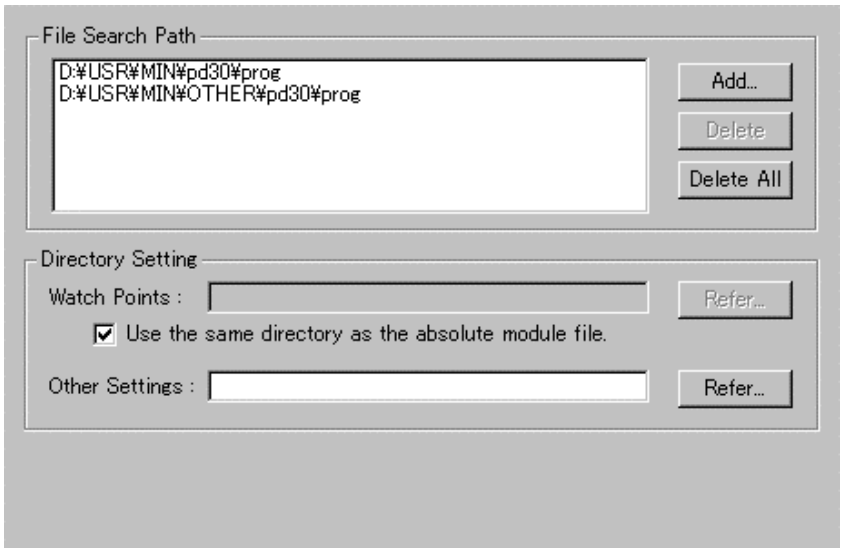
Specify the default tab values for the Program Window, Source Window. You can specify TAB values between 1 and 32.

Note

You can set the tab width by window.
Select the PDxx window Menu - [Option] -> [TAB] while the target window is active.
The TAB designation dialog is opened.

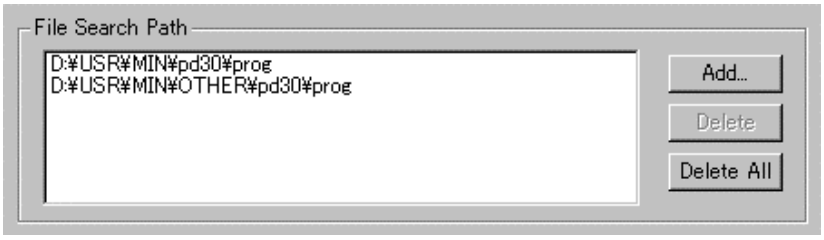
4.4 Path Tab

The specified content becomes effective when the next being start.



4.4.1 Specify the Search Path of Source Files

You can specify the directory position (search path) of the source file to be displayed in a window such as the Program Window.
This method is useful when the source file does not exist in the current directory or divided into multiple directories.



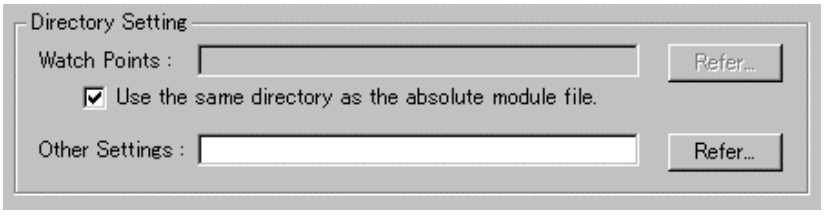
To register the search path, click the Add... button in the File Search Path group.
The folder selection dialog is opened.
Specify the directory in which the source file exists.
To delete a certain search path, click the target search path and click the Delete button.
To delete all the search paths, click the Delete All button.

4.4.2 Specify the Saving Directory of Information File

You can specify the directory in which the ASM/C watch point information file and other information file are saved.
Other files cover the following:

- Script command execution history file
- Break information file

The default saving destination directory of the ASM/C watch point information file is a directory in which the load module exists.
The default saving destination directory of other information file is a directory in which PDxx has been installed (example: c:¥mtool¥pdxx).



To change the directory in which the ASM/C watch point information file is saved, reset a check mark

from the "Use the same directory as the absolute module file" check box in the Directory Setting group. Then, the "Watch Points:" field is enabled.
Click the Refer... button on the right of the "Watch Points:" field and specify the saving destination directory from the Directory Selection dialog.

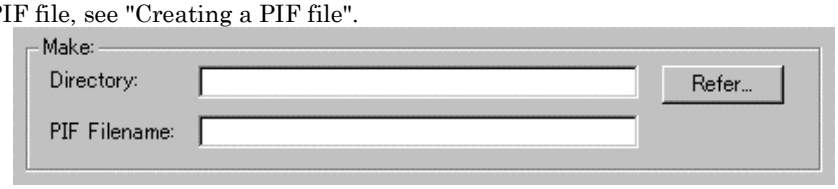
To change to directory in which other information file is saved, click the Refer... button on the right of the "Other Settings:" field and specify the saving destination directory from the Directory Selection dialog.

4.5 Tool Entry Tab

The specified content becomes effective when the next being start.

4.5.1 Starting the make command

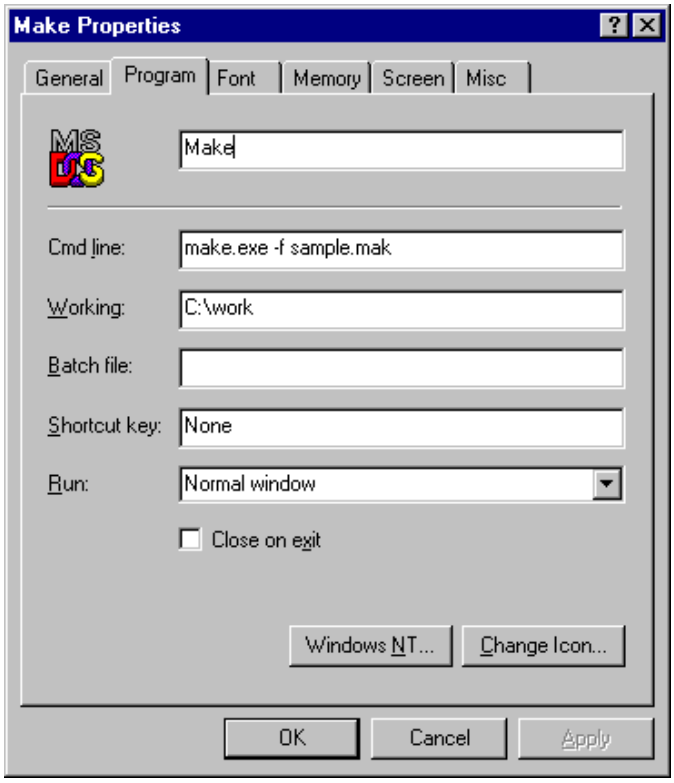
First, prepare a PIF file from which to start the make command.
To create a PIF file, see "Creating a PIF file".



Click the Refer button in the Make group. The Directory Selection dialog is opened.
Specify the directory in which the Make file exists.
Name the PIF file to be registered in the PIF Filename field.

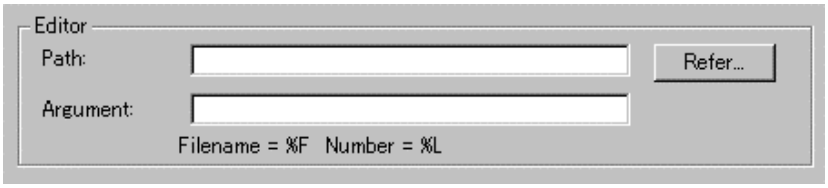
4.5.1.1Create a PIF File

1. Create a keyboard shortcut for command.com located in the Windows directory. Command.com is in the Windows directory in Windows95/98. It is in the system32 directory (The example:¥winnt¥system32) under the Windows directory in Windows NT 4.0/2000/XP.
2. For the keyboard shortcut thus created, assign a file name xxxxx.pif(xxxxx denotes a name specified by the user) and moves the file into the directory that contains makefile.
3. Open the property dialog box for PIF files and enter the make command to execute on the command line.



4.5.2 Specify the Editor

You can start the Editor in a window which displays the source file (Program window, Coverage source window).

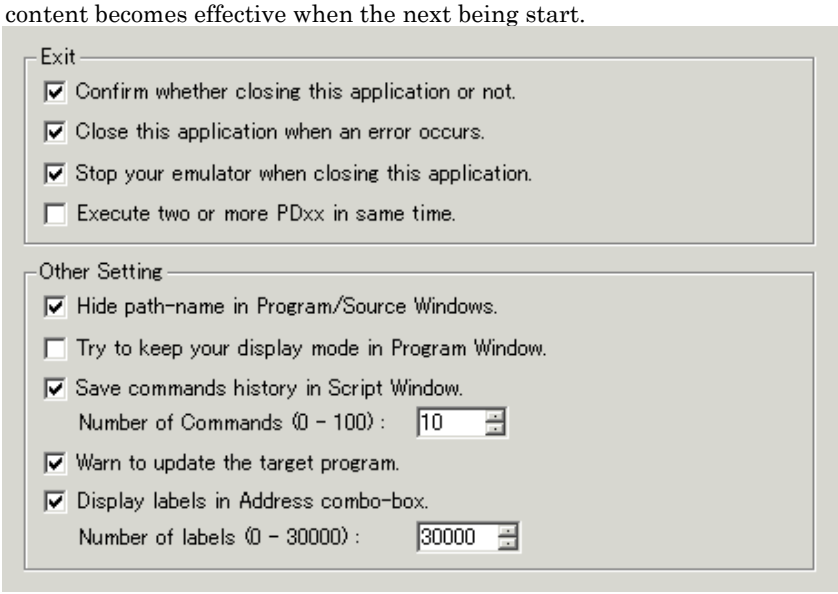


Click the Refer button in the Editor group. The File Selection dialog is opened.
Specify the item file of the editor to be used.

Specify the editor parameter in the Argument field.
File names are stored in "%F", and line numbers are stored in "%L".
To specify the editor options, see the Editor Manual/Help.

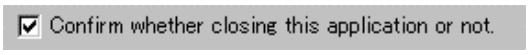
4.6 Other Tab

The specified content becomes effective when the next being start.



4.6.1 Display the Termination Confirmation Dialog

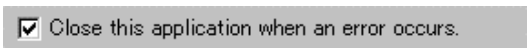
The on-completion confirmation dialog box can be disabled from being opened when the debugger is closed. (By default, it is opened.)



To keep the dialog closed, remove a check mark from the above check box in the Exit group.

4.6.2 Debugger Forced Ending when Error Occurs

You can set a parameter so that the debugger will not be forced to end when an communication error occurs. (The debugger is forced to end by default.)



To do this, remove a check mark from the above check box in the Exit group.

4.6.3 Target Continuance Execution when Debugger Ends

When exiting from the debugger during execution of the target program, you can select to continue execution or stop execution of the emulator. (The emulator is stopped by default.)

To continue execution, remove a check mark from the above check box in the Exit group.

ATTENTION

The target program which is executed continuously cannot be re-controlled next time the debugger gets started.
To start the debugger, press the system reset switch on the emulator to reset the target program.

4.6.4 Enabling multiple startup

Multiple PDxx startup can be enabled (By default, multiple startup is disabled.).

To enable multiple startup, check the above check box included in the Exit group.

4.6.5 Display the Absolute Path of Source File

If the file name is shown with a path on the title bar of the Program (Source) window, you can choose to omit the path and show only the file name.

To hide the file path, check the above check box in the Other Setting group.

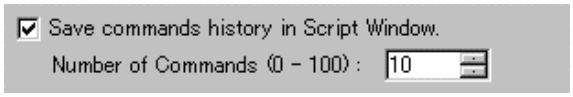
4.6.6 Control the Display Mode Switching of Program Window

You can set switching of the display mode at stop of the target program to "Suppress" (keep the current display mode) in the Program window. (However, the display mode may be switched depending on where the target program is stopped.)

To control the display mode switching, check the above check box in the Other Setting group.

4.6.7 Execution History of Script Command

You can save the execution history of the script command. (Ten sets of history data are saved by default.)

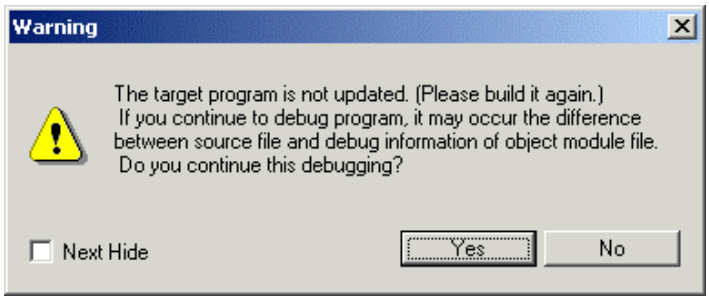


To change the history of script commands, specify the number of history in the Number of Commands field. (0 to 100) To clear history of script commands, remove a check mark from the above check box in the Other Setting group.

4.6.8 Source file update warning

If any source file exists that has been updated after creating the target program, an warning dialog box can be displayed when issuing the commands associated with target execution. (Warned, by default)

If source file update warnings are unnecessary, uncheck the above check box. If the check box is checked, the warning dialog box shown below appears.



Choosing "No" in this warning dialog box cancels the target execution command that was going to be issued. Build and download the target program.
Choosing "Yes" accepts the target execution command that was going to be issued, so that the command is processed normally. From the next time on (until the next time downloading is processed), no warnings will be displayed even when using target execution commands.
If the warning dialog box is closed by checking the Next Hide check box, no source file update warnings are displayed from the next time on (This is the same as when the Warn to update the target program check box is unchecked.).

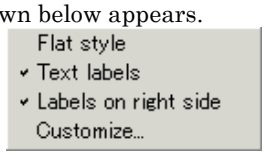
4.6.9 Label List Display in Address Setting Field

It is possible to cease to display the label list in the address setting field of each dialog. (For the default, up to 30,000 labels are displayed in the label list.)

To cease to display the label list, remove the check mark from the above check box. To change the number of label displays, designate the number of labels in the "Number of Labels" field (0-30,000).

4.7 Customizing of Toolbar

The toolbar buttons on each window can be customized. To customize any button, right-click on the window's toolbar. The popup menu shown below appears.

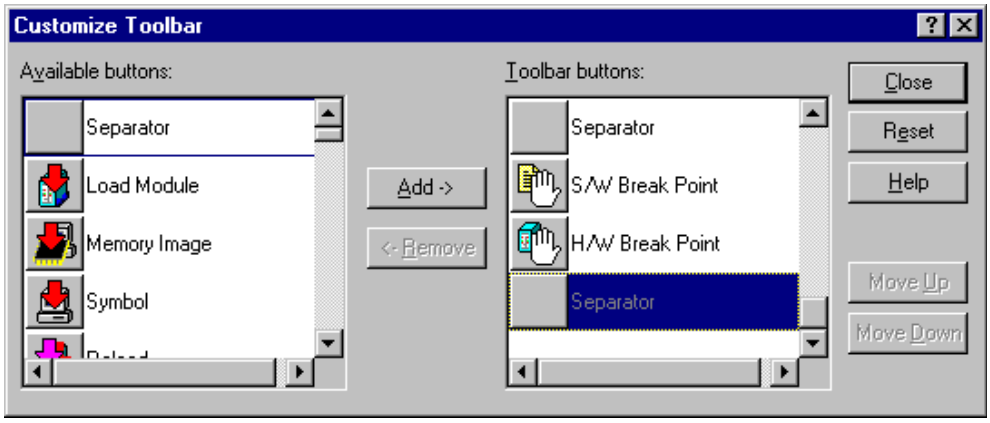


Flat Style	Flattens the button when checked.
Text Labels	Shows text below the button when checked.
Labels on right side	Shows text to the right of the button when checked.
Customize...	Opens a toolbar customize dialog box.

For details about the toolbar customize dialog box, see "Assigning Buttons to the Toolbar."

4.7.1 Assigning Buttons to the Toolbar

To do this, double-click an area in which no button is placed in the tool bar in the window. The Customize Tool Bar dialog opened.



- The buttons corresponding to the option menus in the window are provided.
- You can only add the buttons which are enabled in each window. You cannot add the buttons for other windows.

4.7.1.1 Adding a button

Click the buttons to be added in the "Available Button" list box at right of the Customize Tool Bar dialog. Then, click the "Add" button in the center of the dialog.

4.7.1.2 Deleting a button

Click the button to be deleted int "Tool Bar Button" list box at left of the Customize Tool Bar dialog. Then, click the "Delete" button in the center of the dialog.

4.7.1.3 Changing the button display order

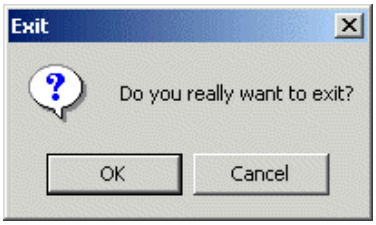
Use the "Up" button or "Down" button at right of the dialog to change the display order. Click the button for which the display order is to be changed in the "Tool Bar Button" list box at left of the Customize Tool Bar dialog. Then, click the "Up" or "Down" button to change the display position.

4.7.1.4 Resetting the display buttons

Click the "Help" button at right of the dialog. The display buttons are reset to the default settings.

5. Ending the Debugger

To ending the debugger, select Menu - [File] -> [Exit]. The Confirmation dialog opens.



When ending the PDxx, click the "OK" button.

To keep the dialog closed, refer to "4.6.1 Display the Termination Confirmation Dialog".

Reference

1. Windows/Dialogs

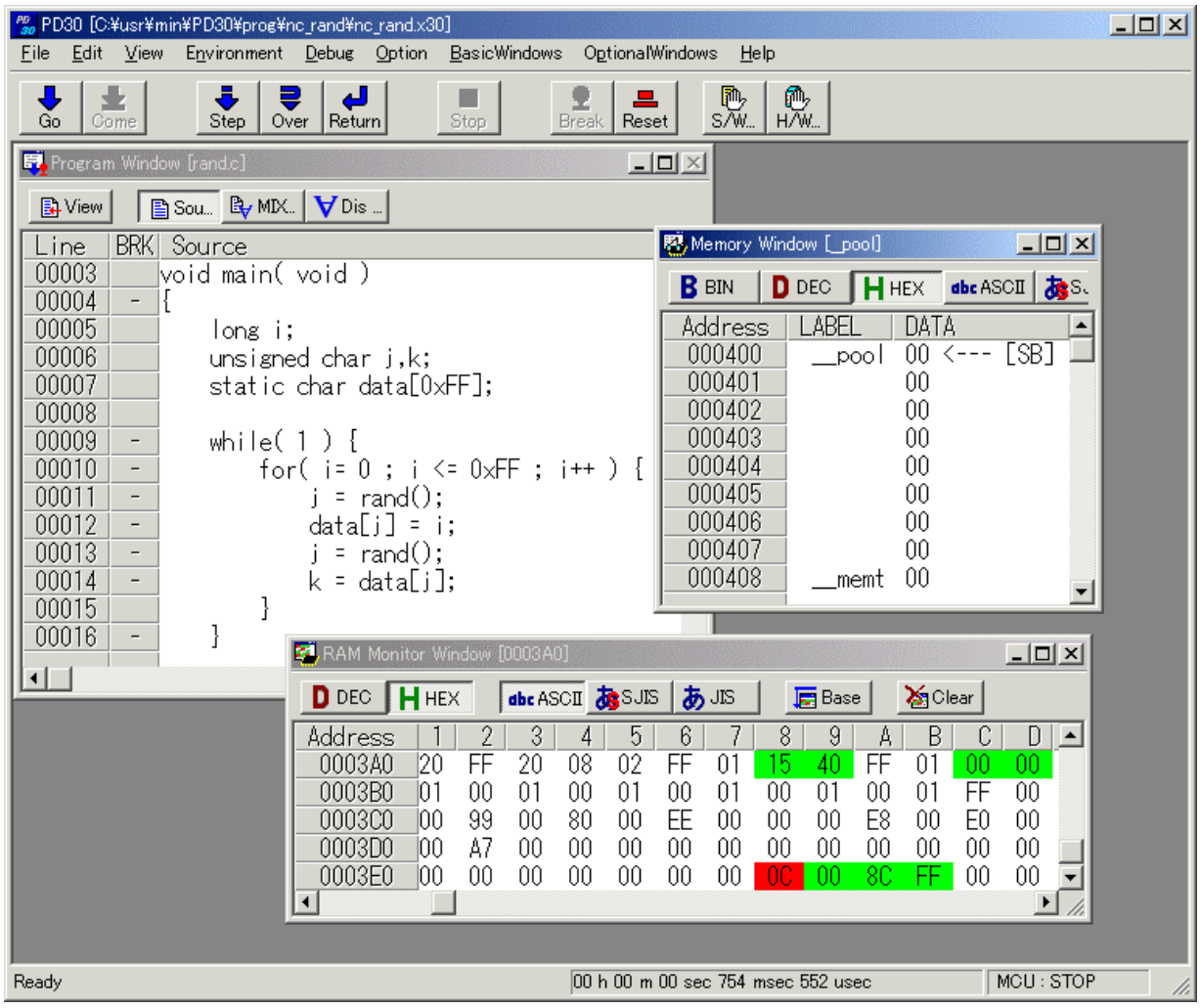
- Windows
  The window of this debugger is shown below.
  When the window name is clicked, the reference is displayed.

PDxx Window
Program Window
Source Window
Register Window
Memory Window
RAM Monitor Window
RAM Monitor Area Setting Window
ASM Watch Window
C Watch Window
Call Stack Window
Script Window
S/W Break Point Setting Window
Protect Window
Address Interrupt Break Window
H/W Break Point Setting Window
Trace Point Setting Window
Trace Window
Coverage Window
Time Measurement Window
MR Window
MR Trace Window
MR Analyze Window
MR Task Pause Window
Task Trace Window
Task Analyze Window
GUI Input Window
GUI Output Window

1.1 PDxx Window

The PDxx Window is the main window for PDxx. This window displays the main commands on a toolbar. You can click on the buttons on this toolbar to run the target program in normal or one-step mode. The main display area accommodates windows such as the Target Program Window.

1.1.1 Configuration of PDxx Window



- The main commands, such as execution/stop of the target program and step execution, are allocated to the tool bar.
- The Option menu is dependent on the active window. When the active window is changed, the Option menu is automatically changed.
- The status bar at the bottom of the PDxx window shows the following information:
- Explanation/display of menus and buttons
- Display the communication interface
- Execution time required from start to end of the target program execution
- Execution state of the target program (during execution or execution stopped)
- Display the Cause of the Program Stoppage

1.1.2 Tool Bar

A basic operation is allocated to the toolbar.

Button	Button Name	Contents
	Go	Execute target Program
	Come	Execute the target program from the value in the program counter to the position of the cursor
	Step	One-step execution of target program
	Over	Step over function/subroutine call
	Return	Run the program up to the higher routine
	Stop	Stop execution of the target program
	Break	Set the position of the cursor in the window as the software breakpoint
	Reset	Reset the target program
	SW	Set S/W breakpoint
	HW	Set H/W breakpoint
	ADIB	Set Address Interrupt breakpoint

1.1.3 Option

In the PDxx window, the following menus are prepared.

File Operation

Menu	Menu Options	Function
File	Download	Download target program.
	Load Module...	Download machine language data and debugging information.
	Memory Image...	Download only machine language data .
	Symbol...	Download only debugging information.
	Rom Data...	Additional download machine language data
	Reload...	Reload target program.
	Upload...	Upload target program.
	Save Disasm...	Save disassembly result.
	(Download File)	List the file name of target program downloaded.
	Exit	Terminate PDxx.

Editing

Menu	Menu Options	Function
Edit	Copy	Copy character strings specified to clipboard.
	Paste	Paste character strings of clipboard.
	Cut	Cut character strings specified to clipboard.
	Delete	Cut character strings specified
	Undo	Undo of edit
	Find...	Find character strings.

Display

Menu	Menu Options	Function
View	Tool Bar	Switch display or non-display of toolbar.
	Status Bar	Switch display or non-display of status bar.
	Tool Bar(Child)	Switch display or non-display of toolbar(child window).

Setup

Menu	Menu Options	Function
Environment	Init...	Environment setup(open the Init dialog box)
	Start Up...	Startup function settings
	Customize...	Open Customize dialog box.

Debugging (Basic)

Menu	Menu Options	Function
Debug	Go Go Go Option...	Start target program. Run from current program counter. Run from specified address.
	GoFree	Free-run target program.
	Come	Run to cursor position.
	Step Step Step Option...	Step execution. Execute one step. Execute specified No. of steps.
	Over Over Over Option...	Over-step execution. Execute one over-step. Execute specified No. of over-steps.
	Return	Execute until return from current subroutine.
	Reset	Reset target program.
	Stop	Stop target program.
	Break Point S/W Break Point... H/W Break Point... Address Interrupt Break Point... Break	Set break point. Open S/W Break Point Setting Window. Open H/W Break Point Setting Window. Open Address Interrupt Break Point Setting Window. Set/cancel software break at cursor.
	Trace Point...	Open Trace Point Setting Window.
	Scope...	Open Scope Setting dialog box
	Make	Make target program

Debugging (Option)

Menu	Menu Options	Function
Option	The content of option menu depends on the active window. The content of the menu changes automatically when an active window changes. Please refer to the reference of each window for the content of the menu of each window.	

Window Operations (Basic Window)

Menu	Menu Options	Function
Basic Windows	Cascade	Cascade windows.
	Tile	Tile windows.
	Arrange Icon	Arrange icons.
	Program Window	Make Program Window active.
	Source Window	Open Source Window.
	Register Window	Open Register Window.
	Memory Window	Open Memory Window.
	RAM Monitor Window	Open RAM Monitor Window.
	ASM Watch Window	Open ASM Watch Window.

	C Watch Window	Open C Watch Window.
	Call Stack Window	Open Call Stack Window
	Script Window	Open Script Window

Window Operations (Optional Window)

Menu	Menu Options	Function
Optional Windows	Protect Window	Open Protect Window.
	H/W Break Point Setting Window	Open H/W Break Point Setting Window.
	Trace Point Setting Window	Open Trace Point Setting Window.
	Trace Window	Open Trace Window.
	Coverage Window	Open Coverage Window
	Time Measurement Window	Open Time Measurement Window
	Realtime OS Windows MR Window MR Trace Window MR Analyze Window MR Task Pause Window Task Trace Window Task Analyze Window	Real-time OS Windows Open MR Window Open MR Trace Window Open MR Analyze Window Open MR Task Pause Window Open Task Trace Window Open Task Analyze Window
	GUI Windows GUI Input Window GUI Output Window	GUI Widows Open GUI Input Window Open GUI Output Window
	Custom Windows* Option* (Custom Window) *	Custom Windows Entry Custom Window Open the custom window

*PDxxF dose not support.

Help

Menu	Menu Options	Function
Help	Contents	Display Help.
	Active Window	Display Help of Active Window.
	About...	Display version information

1.2 Program Window

The Program window always displays the source file corresponding to the current program counter position.
This window is opened automatically at start. The background of the program counter position is displayed in yellow.
This window allows you to execute the source program up to the cursor position, set/reset the software breakpoint, and perform line assemble.

The Program window provides the three display modes as below:

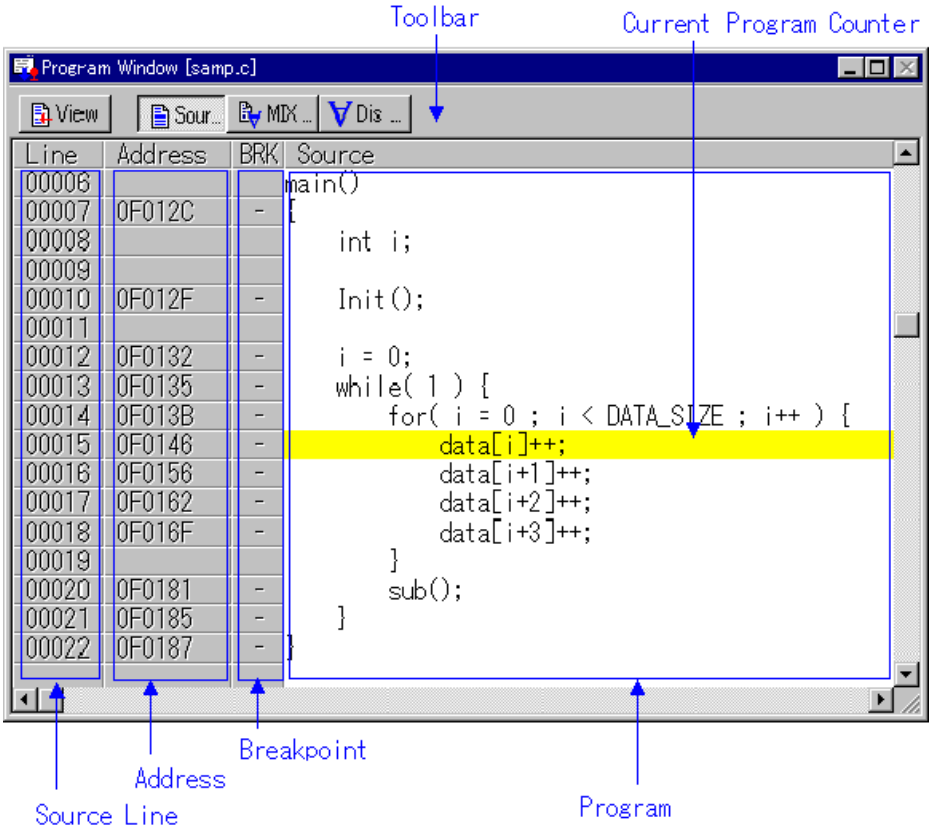
- **Source display mode**
  Displays the source file of the target program.Can also be used to edit the source file.
- **Disassemble Mode**
  Displays the disassemble result of the target program.
- **MIX display mode**
  Displays the source file of the target program and its disassemble result in a mixed style.

1.2.1 Configuration of Source Display Mode

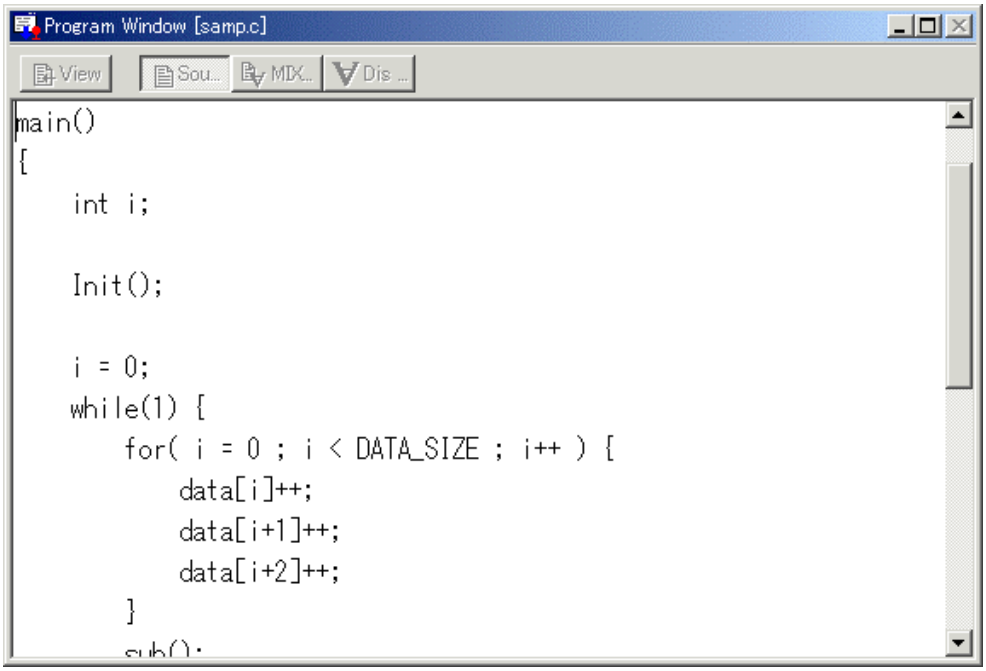
The program window has the following two source display modes. These display modes can be changed from menus on the program window.

- Debug mode This mode is used to debug (e.g., run or stop) the target program.

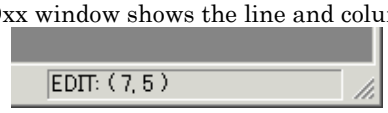
- Edit mode This mode is used to edit the source file.
  The program window's source display mode (debug mode) is configured as shown below.



- You can switch "Display/Hide" for the line number display area/address display area.
- You can change the source file to be displayed by double-clicking the line number display area.
- You can change the display start address/display start line by double-clicking the address display area.
- You can set/reset the breakpoint by clicking (or double-clicking) the breakpoint display area(Contents of breakpoint display area).
- By staying the mouse cursor on a C language variable for a given period of time (about 0.5 second), the variable data is popped up.
- You can drag the function name and then click the mouse right button to display the source file corresponding to the function.
- You can drag the C language variable and then click the mouse right button to register the variable as the C watch point.
- You can drag the assembler symbol and then click the mouse right button to register the symbol as the ASM watch point.
- You can open the displayed source file on the editor. (You must have registered the editor name.)
- The source file being displayed can be edited. The source file can also be edited on an external editor. (The editor name must be registered.)
- You can display the coverage measurement result by specifying the option. (It is not displayed by default.)
- You can line-assemble the clicked position.
- The program window's source display mode (edit mode) is configured as shown below.



- The line number display, address display, and breakpoint display areas are not shown.
- The right-click menu changes for exclusive use in edit mode.
- The status bar on the PDxx window shows the line and column numbers of the cursor position.



1.2.2 Contents of breakpoint display area

When a breakpoint is set, the following marks are displayed in the breakpoint field of the progam window (same for the source window).

Functions	Mark
Software Breaks	"B"
Address Interrupt Breaks	"A"
Hardware Breaks	"H"

These Breaks can be set from the Program or Source Window. Please refer to the following for the method of setting and clear.

1.2.2.1 Software Breaks, Address Interupt Breaks

You can also set break points in the Program or Source Window. To do so, double-click the break point setting display area (BRK column) (indicated by "-") for the line in which you want to set the break. Depending on the situation, the debugger behaves as follows.

- **If the Enable the Address Match Interrupt Break Function check box on the Inix dialog box is deselected**
  Software breakpoints are always set. No address interrupt breakpoints are set.
- **If the Enable the Address Match Interrupt Break Function check box on the Inix dialog box is selected**
  Up to a valid range of address interrupt breakpoints are set preferentially over other breakpoints. If the number of breakpoints set exceeds the valid range, software breakpoints are set thereafter.

When any address interrupt breakpoint is set, the "-" display changes to the letter "A." When a software breakpoint is set, the "-" display changes to the letter "B."
You can delete the break point by double-clicking again in the BRK column.
If the breakpoint display field is blank in the Program (Source) Window (data definition line, comment

line, null line), you cannot set a breakpoint.

About the Software Breaks

- You can set up to 64 software break points.
- The areas for which software breakpoints can be set varies with each product.

About the Address Interupt Breaks

- The number of address interrupt breakpoints that can be set varies with each product.
- The address interrupt break function can only be used when the Enable the Address Match Interrupt Break Function check box on the Init dialog box MCU tab is selected. Even in this case, however, no address interrupt breakpoints can be set if one of the following conditions hold true:
  - Target program execution is running.
  - When operating in microprocessor mode, the block that contains the address interrupt interrupt vector is mapped into an "External" or "NoUse" area.

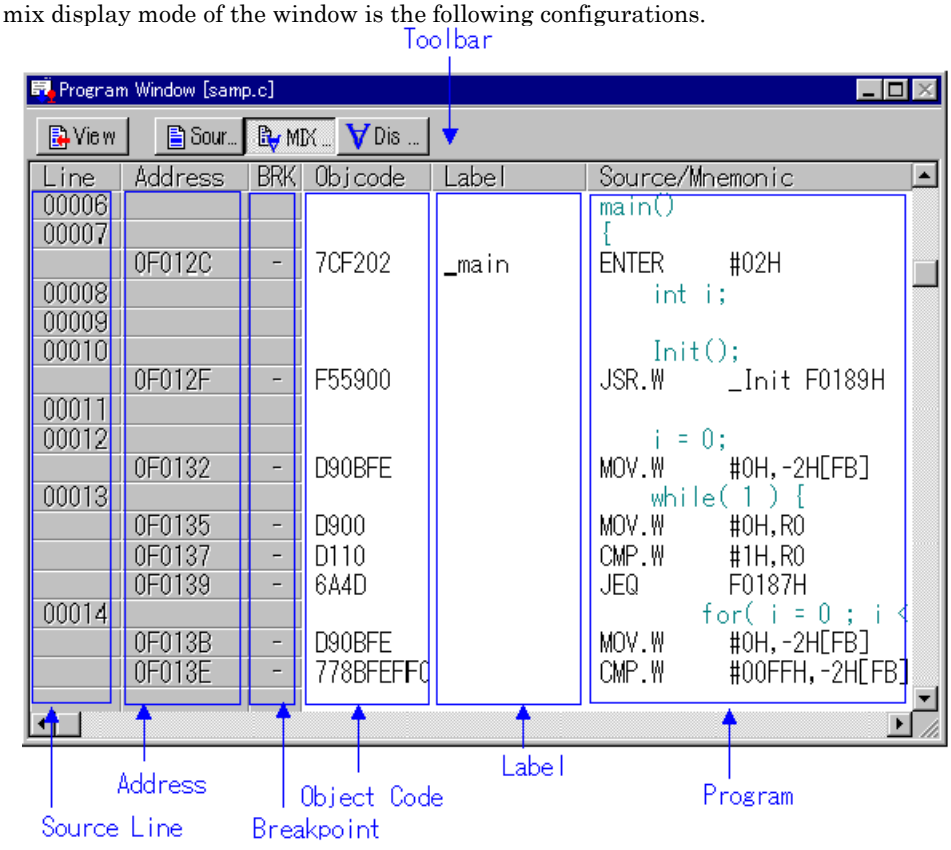
1.2.2.2 Hardware Breaks

You can also set break points in the Program or Source Window. To do so, **right-click** the break point setting display area for the line in which you want to set the break.(A display of "-" changes to "H") Lines in which a software break point have been set are marked by a "H"in place of the "-" in the break point setting display area (BRK column). You can delete the break point by right-clicking again in the BRK column. ("H" changes to "-")

- Breakpoints here can be set only when the selected trace type is Fetch.
- The condition for the combinatorial condition of hardware breaks is set to OR. (the program breaks when one of break conditions is met).
- No hardware breakpoints can be set if the breakpoint display column of the program (source) window is blank.

1.2.3 Configuration of MIX Display Mode

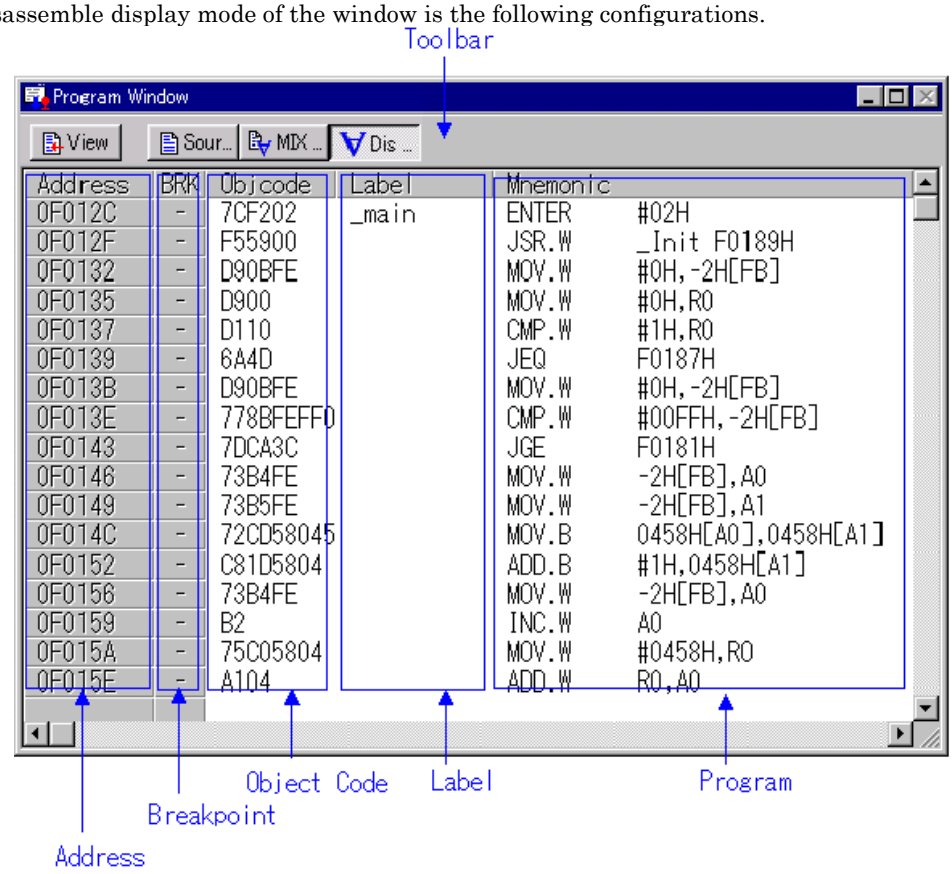
The mix display mode of the window is the following configurations.



- You can switch "Display/Hide" for the line number display area/address display area/object code display area.
- You can change the source file to be displayed by double-clicking the line number display area.
- You can change the display start address/display start line by double-clicking the address display area.
- You can set/reset the breakpoint by clicking (or double-clicking) the breakpoint display area.
- You can change the display ratio between the object code display area and the label display area, and between the label display area and the program display area, using the mouse.
- You can open the displayed source file on the editor. (You must have registered the editor name.)
- You can display the coverage measurement result by specifying the option. (It is not displayed by default.)
- The MIX display result can be saved as a text file.
- You can line-assemble the clicked position.
- You can scroll the display up/down in units of source line.

1.2.4 Configuration of Disassemble Display Mode

The disassemble display mode of the window is the following configurations.



- You can switch "Display/Hide" for the address display area/object code display area.
- You can change the display start address by double-clicking the address display area.
- You can set/reset the breakpoint by clicking (or double-clicking) the breakpoint display area.
- You can change the display ratio between the object code display area and the label display area, and between the label display area and the program display area, using the mouse.
- You can display the coverage measurement result by specifying the option. (It is not displayed by default.)
- You can line-assemble the clicked position.

1.2.5 Extended Menus

The Program window provides the following menu when being active. (This menu is called Program window option.)

Menu	Menu Options	Function
Option	Font...	Change font.
	TAB...	Set source file display tabs.
	Color...	Change display color
	View	Change contents of display.
	Source...	Display from specified source file or function.
	Address...	Display from specified address or line No.
	Program Counter	Display from current program counter.
	Mode	Switch display mode.
	Source Mode	Switch to source display mode.
	Mix Mode	Switch to MIX display mode.
	Disasm Mode	Switch to disassemble display mode.
	Layout	Set layout.
	Line Area	Switch display or non-display of line No. area.
	Address Area	Switch display or non-display of address area.
	Code Area	Switch display or non-display of object code area.
	Line Assemble...	Open Line Assemble dialog.
	Save Mix...	Saves MIX display result
	Coverage	Set Coverage measurement.
	On/Off	Switch display or non-display of Measurement result.
	Base...	Change coverage base address
	Clear	Initialize coverage measurement result
	Refresh	Update display of coverage measurement result
	Edit	Edit functions
	On	Turns editing on or off
	Save	Saves the edited contents by overwriting
	Save As...	Saves the edited contents with another name
	Save All	Saves all of the edited contents by overwriting

1.2.6 Shortcut Menu

The Program window provides the shortcut menu by clicking the mouse right button within the window. (This menu is called Program window right-click menu.)
The menu content varies depending on the clicked position.

- **When right-clicking the line number display area or address display area**
  The shortcut menu same as the option menu appears.
- **When right-clicking the breakpoint display area**
  The shortcut menu does not appear. Hardware break can be set.
- **When right-clicking other area**
  The following shortcut menu appears.

(Debug Mode)

Menu	Menu Options	Function
Right-Click	Jump to function	Display the selected function
	Open Source Window	Display the selected function (by Source Window)
	Set PC here	Sets the PC at the cursor position
	Add C Watch...	Register the C watch point on selected variable
	Add C Watch Pointer..	Register the C watch point on selected pointer variable
	Add ASM Watch	Register the ASM watch point on selected symbol
	BitAdd ASM Watch	Register the ASM watch point on selected bit symbol

	Open Editor Line Assemble...	Open the source file by the editor Open the Line Assemble dialog.
	Save Mix...	Saves MIX display result
	Edit On	Edit functions Turns editing on or off

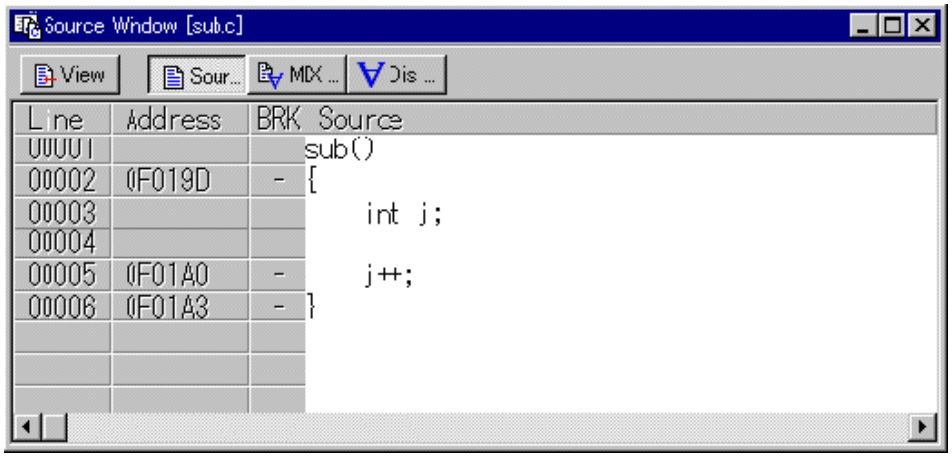
(Edit mode)

Menu	Menu Options	Function
Right-Click	Copy Paste Cut Delete Undo	Same [Edit]->[Copy] menu. Same [Edit]->[Paste] menu. Same [Edit]->[Cut] menu. Same [Edit]->[Delete] menu. Same [Edit]->[Undo] menu.
	Find	Find character strings.
	Font Tab	Change font. Set source file display tabs.
	Edit On Save Save As... Save All	Edit functions Turns editing on or off Saves the edited contents by overwriting Saves the edited contents with another name Saves all of the edited contents by overwriting

1.3 Source Window

The Source window continuously displays any position of the source file. (The Program window always displays the source file corresponding to the current program counter position.)
When the program counter points the displayed source file position, its background is displayed in yellow.
Like the Program window, the Source window allows you to execute the source program up to the cursor position, set/reset the software breakpoint and perform line-assemble.
You can open up to 30 Source windows.

1.3.1 Configuration of Source Window



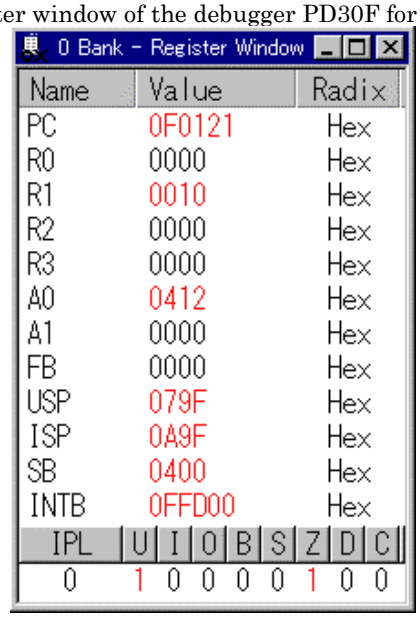
The Source Window configuration, toolbar and option menu is the same as that in the Program Window.
Please refer to Program Window.

1.4 Register Window

The Register window displays the register data and flag data.
You can change a register/flag value from the window.

1.4.1 Configuration of Register Window

The figure below shows a Register window of the debugger PD30F for M16C/60, 20 series.



- If a register/flag value is changed, the value is displayed in red.
- Double-clicking the register display line opens a dialog, which allows you to change a register value.
- You can change a flag value by clicking the button corresponding to the flag.
- The right-click menu allows you to change the display radix point and the register bank.
- You can change the display ratio between the register name display area and the register value display area, and between the register value display area and the radix point display area, using the mouse.

1.4.2 Extended Menus

The Register window provides the following menu when being active. (This menu is called Register window option.)

Menu	Menu Options	Function
Option	Bank0	Display registers of bank 0.
	Bank1	Display registers of bank 1.
	Layout	Set layout
	Hide Radix	Switch display or non-display of radix.
	Hide FLAGs	Switch display or non-display of flags display area.
	Font...	Change font.

1.4.3 Shortcut Menu

Press the right button on the register display area in Register Window to display shortcut menu.

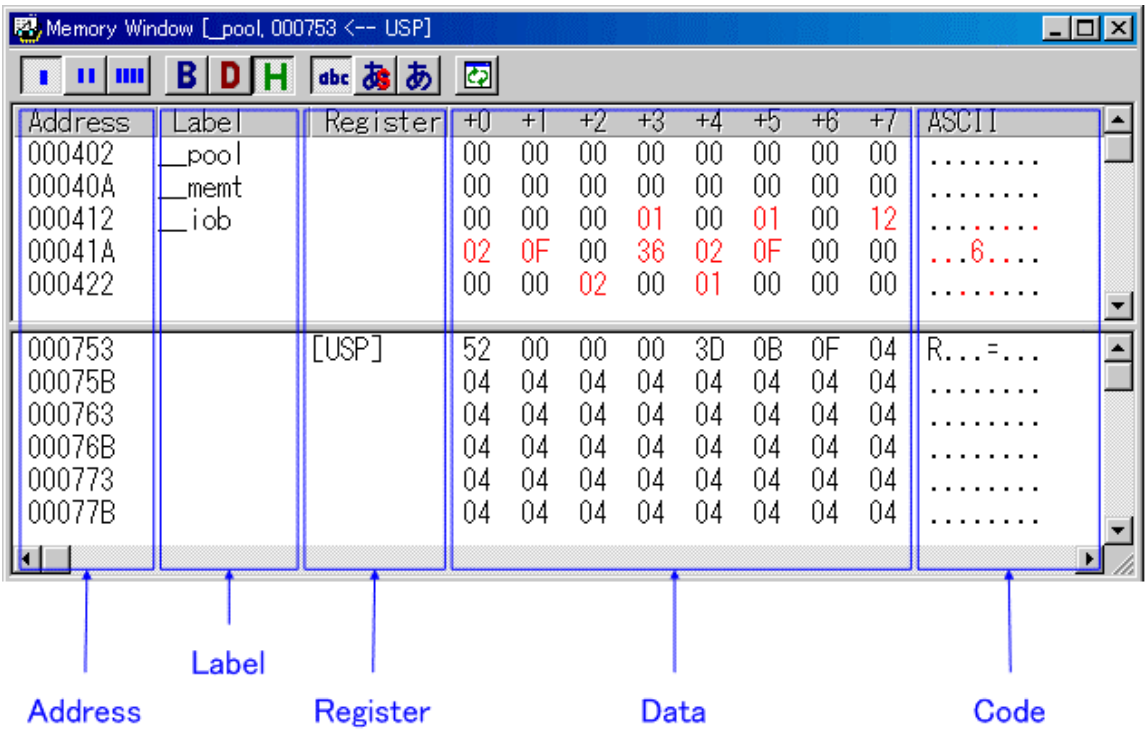
Menu	Menu Options	Function

Right Click	Hex Dec Bin	Display in hexadecimal. Display in decimal. Display in binary.
	Bank0 Bank1	Display registers of bank 0. Display registers of bank 1.
	Layout Hide Radix Hide FLAGs	Set layout. Switch display or non-display of radix Switch display or non-display of flags display area.
	Font...	Change font.

1.5 Memory Window

The Memory Window displays the contents of contiguous memory in "address", "label", and "data (contents of memory)" formats.
The display is updated after each command is executed. Data can be displayed in binary, decimal, hexadecimal, and ASCII. You can open up to 30 Memory Windows.
You can use the Memory Windows to modify the contents of memory, and also to fill and move specified blocks of memory.

1.5.1 Configuration of Memory Window



- The display start address and memory contents can be changed by in-place editing.
- Different memory areas can be inspected at the same time by dividing the window into halves.
- It is possible to keep track of the stack pointer position (by default, not tracked).
- Coverage measurement results can be displayed.
- Memory contents can be stored in a text file.

1.5.2 Option Menu

The Memory window provides the following menu when being active. (This menu is called Memory window option.)

Menu	Menu Options	Function
Option	Set...	Set data at specified address.
	Fill...	Fill specified memory block with data.
	Move...	Move specified memory block to
	Save Memory Contents...	Saving Memory Contents in a Text File
	Address...	Specify display starting address.
	Scroll Area...	Specify scroll range.
	Register	Starting address to value of the register.
	(xxxxx) *1	(Product dependence menu)
	Followed Stack Pointer...	Keep tracking of the stack pointer position.
	Set Start Up Symbol...	Startup label settings
	Refresh	Refresh display.
	Data Length	Specify data length.
	Byte	Display in 1-byte units.
	Word	Display in 2-byte units.
	Lword	Display in 4-byte units.
	Radix	Specify data radix.
	Hex	Display in hexadecimal.
	Dec	Display in decimal.
	Bin	Display in binary.
	Code	Specify data code.
	ASCII	Display as ASCII characters.
	SJIS	Display as SJIS characters.
	JIS	Display as JIS characters.
	Layout	Set layout.
	Label	Switch display or non-display of Label area.
	Register	Switch display or non-display of Register area.
	Column...	Changing the number of digits displayed
	Coverage	Set Coverage measurement.
	Enable	Switch display or non-display of Measurement result
	Base	Change coverage base address
	Clear	Initialize coverage measurement result
	Font...	Change font
	Color...	Change display color

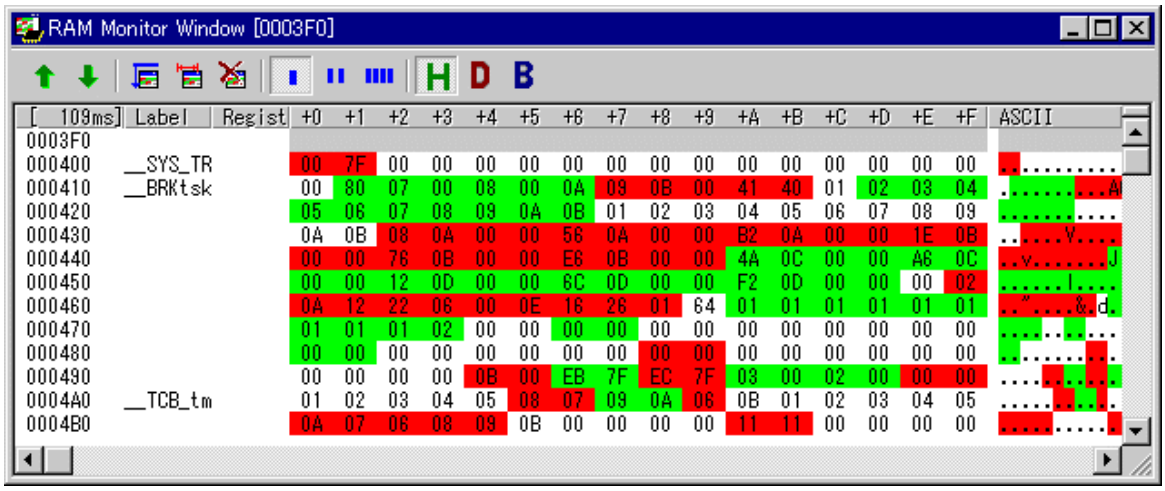
These menus can be selected even by the short cut menu by a right click in the window.

*1Product Dependence Menu

Product	Menu	Function
PD308F PD30F	FB	Starting address to value of FB register.
	SB	Starting address to value of SB register.
	USP	Starting address to value of USP register.
	ISP	Starting address to value of ISP register.

1.6 RAM Monitor Window

The RAM monitor window is a window in which changes of memory contents are displayed while running the target program.
The relevant memory contents are displayed in dump form in the RAM monitor area (in varying sizes depending on the emulator used) by using the realtime RAM monitor function. The displayed contents are updated at given intervals (by default, every 100 ms) while running the target program.
For the PC7501, up to 16 windows can be opened.



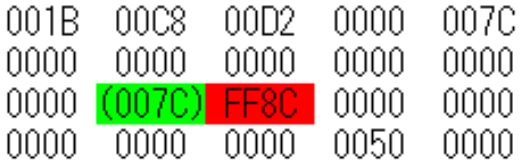
- The RAM monitor area can be changed to any desired address range. Refer to "Setting the RAM monitor area" for details on how to change the RAM monitor area. The default RAM monitor area is mapped into a 1-Kbyte area beginning with the start address of the internal RAM.
- The display content updating interval can be set for each window individually. The actual updating interval at which the display contents are actually updated while running the target program is shown in the title field of the Address display area.
- The background colors of the data display and code display areas are predetermined by access attribute, as shown below.

Access attribute	Background color
Read accessed address	Green
Write accessed address	Red
Non-accessed address	White

The background colors can be changed.

ATTENTION

- The RAM monitor window shows the data that have been accessed through the bus. Therefore, changes are not reflected in the displayed data unless they have been accessed via the target program as in the case where memory is rewritten directly from an external I/O.
- If the data in the RAM monitor area are displayed in lengths other than the byte, it is possible that the data will have different memory access attributes in byte units. If bytes in one data have a different access attribute as in this case, those data are enclosed in parentheses when displayed in the window. In that case, the background color shows the access attribute of the first byte of the data.



- The interval time at which intervals the display is updated may be longer than the specified interval depending on the operating condition (shown below).
- Host machine performance/load condition
- Communication interface
- Window size (memory display range) or the number of windows displayed
- The displayed access attributes are initialized by downloading the target program.

1.6.1 Extended Menus

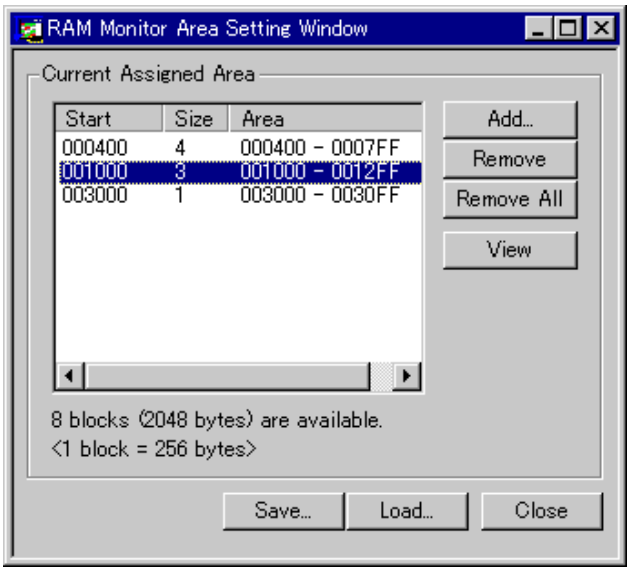
The RAM Monitor window provides the following menu when being active. (This menu is called RAM Monitor window option.)

Menu		Function
RAM Monitor Area...		Sets RAM monitor area A window or dialog box in which you can change the RAM monitor area is displayed.
Sampling Period...		Sets display updating interval Sets an interval time at which intervals you want the display to be updated while running the target program.
Clear		Clears access attribute The data in all RAM monitor areas are cleared, as are the displayed access attributes.
Up		Moves display position to the immediately preceding RAM area (smaller address) The display position is moved forward (toward smaller addresses) to the beginning of the RAM monitor area that immediately precedes the current display position.
Down		Moves display position to the immediately following RAM area (larger address) The display position is moved backward (toward larger addresses) to the beginning of the RAM monitor area that immediately follows the current display position.
Address...		Display from specified address.
ScrollArea...		Specify scroll range.
Data Length	Byte	Display in 1-byte units.
	Word	Display in 2-byte units.
	Lword	Display in 4-byte units.
Radix	Hex	Display in hexadecimal.
	Dec	Display in decimal.
	Bin	Display in binary.
Code	ASCII	Display as ASCII characters.
	SJIS	Display as SJIS characters.
	JIS	Display as JIS characters.
Layout	Label	Switch display or non-display of Label area.
	Register	Switch display or non-display of Register area.
Column...		Set up the number of column displayed on one line.
Font...		Change font.
Color...		Set color of access attribute display.

These menus can be selected even by the short cut menu by a right click in the window.

1.7 RAM Monitor Area Setting Window

The RAM monitor area setup window is a window in which you can set the RAM monitor area of the realtime RAM monitor function(by adding, deleting or modifying).
This window is brought by choosing the menu [Option]->[RAM Monitor Area...] of the window in which you use the realtime RAM monitor function.



- The currently set RAM monitor areas are displayed in order of start addresses beginning with the smallest.
- RAM monitor areas can be added or deleted.
- The start addresses or sizes of the RAM monitor areas set can be changed.
- The memory content of a selected RAM monitor area can be displayed in the RAM monitor window.
- Settings of RAM monitor areas can be saved to a file. Also, the saved information of RAM monitor areas can be loaded.
- The default RAM monitor area is mapped into a 1-Kbyte area beginning with the start address of the internal RAM area.
- The next time the debugger starts, the previously set contents become effective (when resume = ON).

ATTENTION

- The size and the number of RAM monitor areas that can be specified vary with each emulator used. Refer to About the realtime RAM monitor function for details.
- Refer to Setting the RAM monitor area for details on how to add or delete RAM monitor areas or how to change the start address or size of RAM monitor areas.

1.7.1 Command Button

The buttons at the bottom of the RAM Monitor Area Setting window have the following meanings.

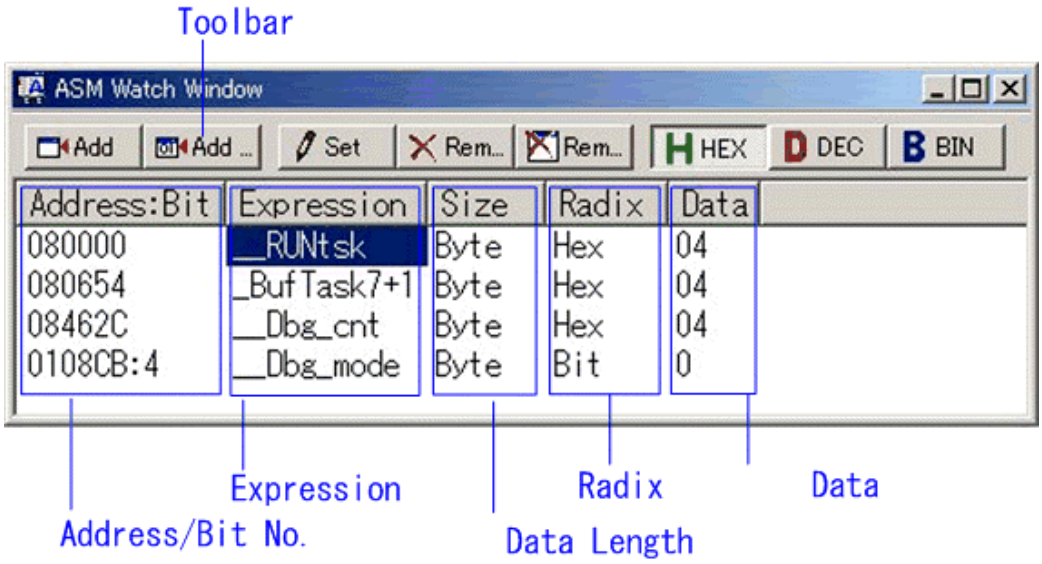
Button Name	Content
Add...	Add the RAM Monitor Area.
Remove	Delete the RAM Monitor Area.
Remove All	Delete all RAM monitor Areas.
View	Shows the selected RAM Monitor Area in the RAM Monitor window.
Save...	Saves the contents set in the window to a file.
Load...	Loads setting information from a file in which it was saved.
Close	Closes the window

1.8 ASM Watch Window

The ASM watch window is a window in which you can register specific addresses as watchpoints and inspect memory contents at those addresses.

If a registered address resides within the RAM monitor area, the memory content at that address is updated at given intervals (by default, every 100 ms) during program execution. The toolbar buttons may be used to perform the main functions.

1.8.1 Overview of ASM Watch Window



- The addresses to be registered are called the "watchpoints." One of the following can be registered:
  - Address (can be specified using a symbol)
  - Address + Bit number
  - Bit symbol
- The registered watchpoints are stored in an environment setup file when the ASM watch window is closed and are automatically registered in the debugger when the window is reopened.
- If symbols or bit symbols are specified for the watchpoints, the watchpoint addresses are recalculated when downloading the target program.
- The invalid watchpoints are marked by "-<not active>-" when displayed on the screen.
- The order in which the watchpoints are listed can be changed by a drag-and-drop operation.
- The watchpoint symbols, sizes and radixes can be changed by in-place editing.

ATTENTION

- The RAM monitor obtains the data accessed through the bus. Any change other than the access from the target program will not be reflected.
- If the display data length of the RAM monitor area is not 1 byte, the data's access attribute to the memory may varies in units of 1 byte. In such a case that the access attribute is not unified within a set of data, the data's access attribute cannot be displayed correctly. In this case, the background colors the access attribute color of the first byte of the data.

1.8.2 Extended Menus

When the ASM watch window is active, the following menus can be used.

Menu	Function
Add... Add Bit... Remove Remove All...	Add watch point. Add bit-level watch point. Remove selected watch point. Remove all watch points.
Set...	Set new data to selected watch point.

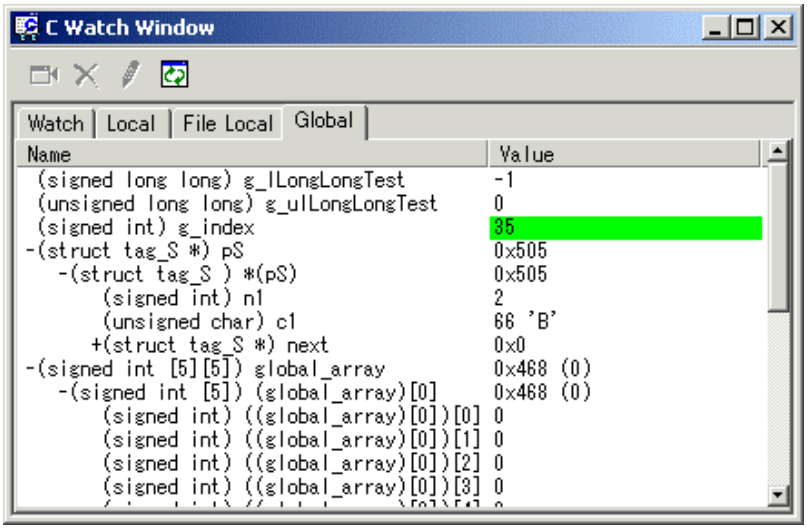
Radix Bin Dec Hex	Change display radix. Display value in binary. Display value in decimal. Display value in hexadecimal.
Refresh Layout Address Area Size Area	Refresh display. Set layout. Switch display or non-display of address/bit area. Switch display or non-display of data size area.
RAM Monitor RAM Monitor Area... Sampling period... Clear	Display RAM monitor. Set RAM monitor area. Set sampling period for RAM monitor. Clear the access attribute.
Save... Load...	Save the watch points. Load the watch points.
Font Color...	Change font. Set color of access attribute display.

These menus can be selected even by the short cut menu by a right click in the window.

1.9 C Watch Window

The C Watch Window displays C/C++ expressions and their values (results of calculations).
The C/C++ expressions displayed in the C Watch Window are known as C watchpoints. The displays of the results of calculating the C watchpoints are updated each time a command is executed.
When RAM monitor function is effective and the C watch points are within the RAM monitor area, the displayed values are updated during execution of the target program.

1.9.1 Configuration of C Watch Window



- Variables can be inspected by scope (local, file local or global).
- The display is automatically updated at the same time the PC value changes.
- Variable values can be changed.
- The display radix can be changed for each variable individually.
- Any variable can be registered to the Watch tab, so that it will be displayed at all times:
- The registered content is saved for each project separately.
- If two or more of the C watch window are opened at the same time, the registered
- The C watchpoints can be registered to separate destinations by adding Watch tabs.
- Variables can be registered from another window or editor by a drag-and-drop operation.
- The C watchpoints can be sorted by name or by address.

- Variable names can be searched.
- Values can be inspected in real time during program execution by using the RAM monitor function.

ATTENTION

- You cannot change the values of the C watch points listed below:
  - Bit field variables
  - Register variables
  - C watch point which does not indicate an address(invalid C watch point)
- If a C/C++ language expression cannot be calculated correctly (for example, when a C/C++ symbol has not been defined), it is registered as invalid C watch point. It is displayed as "--<not active>--". If that C/C++ language expression can be calculated correctly at the second time, it becomes an effective C watch point.
- The display settings of the Local, File Local and Global tabs are not saved. The contents of the Watch tab and those of newly added tabs are saved.
- The RAM monitor obtains the data accessed through the bus. Any change other than the access from the target program will not be reflected.
- The variables, which are changed in real-time, are global variables and file local variables only.
- If the display data length of the RAM monitor area is not 1 byte, the data's access attribute to the memory may varies in units of 1 byte. In such a case that the access attribute is not unified within a set of data, the data's access attribute cannot be displayed correctly. In this case, the background colors the access attribute color of the first byte of the data.

1.9.2 Extended Menus

The C Watch window provides the following menu when being active. (This menu is called C Watch window option.)

Menu Options	Function
Add...	Registers a new C watchpoint A new watchpoint can be registered by specifying its name in a dialog box. The result is reflected in all C watch windows.
Remove	Deletes a selected C watchpoint The watchpoint being selected by an active tab is deleted. The result is reflected in all C watch windows.
Initialize	Reevaluates a selected C watchpoint The currently selected watchpoint is reevaluated. Use this function in cases where variables although in the same name assume different types depending on scope, or where when expanding more than 100 arrays, you want to change a specified number of elements.
Set New Value...	Change value Change value of the selected C watch point.
Radix	Change radix Change radix of the selected C watch point.
Hex	Display in HEX Display the selected C watch ponit in hexadecimal.
Bin	Display in BIN Display the selected C watch point in Binary.
Default	Default radix Display the selected C watch point in default radix.
Refresh	Updates a variable value Variable values are updated (a memory access occurs).
Hide type name	Turns type name display on/off Type names are shown on the screen or hidden. The result affects all C watch windows.
Show char* as string	Selects whether to display char* type as a string

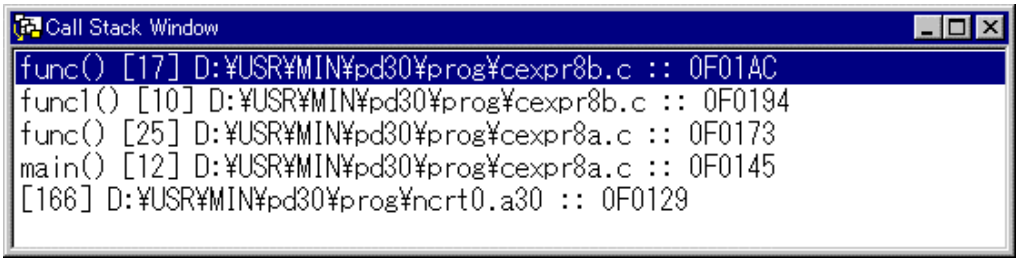
	The char* type is displayed as a string or as a pointer to char type. The result affects the entire PDxx
Sort	Sorts C watchpoints Watchpoints are sorted. The result affects all C watch windows.
Sort by Name	Sorts by name Variables are sorted by name.
Sort by Address	Sorts by address Variables are sorted by address.
RAM Monitor	Sets RAM monitor function The debugger is set to use the RAM monitor function to update values during program execution.
Enable RAM Monitor	Enables RAM monitor function The RAM monitor function is turned on or off.
RAM Monitor Area...	Set RAM monitor area Set or modify the RAM Monitor Area.
Sampling Period...	Set sampling period Change the sampling period for RAM monitor function.
Clear	Clear the access attribute Clear the access attribute for RAM Monitor function.
Add New Tab...	Adds a watch tab A new watch tab is added by specifying its name. The result is reflected in all C watch windows.
Remove Tab	Deletes a displayed watch tab The currently active watch tab is deleted. The result is reflected in all C watch windows.
Save...	Saves active watch tab content to a file The content of the active watch tab is saved to a file. The contents of the Local, File Local and Global tabs are not saved.
Load...	Loads saved content to an active watch tab The saved content is loaded from a file into the active watch tab. The result is reflected in all C watch windows.
Font...	Change font Change the display font. Each window has its own font.
Color...	Change color Change the display colors. All C watch window uses the common colors.

These menus can be selected even by the short cut menu by a right click in the window.

1.10 Call Stack Window

The Call Stack window displays the C language function call state of the target program.

1.10.1 Configuration of Call Stack Window



- The window displays the name of the called function and the function call position (file name, line number, address) sequentially from the current program counter position.
- The top line shows a function at the current PC position. The last line shows a function call source.

- By double-clicking the function name, the call position (line) of the function is displayed in the Program window.

1.10.2 Extended Menus

The Call Stack window provides the following menu when being active. (This menu is called Call Stack window option.)

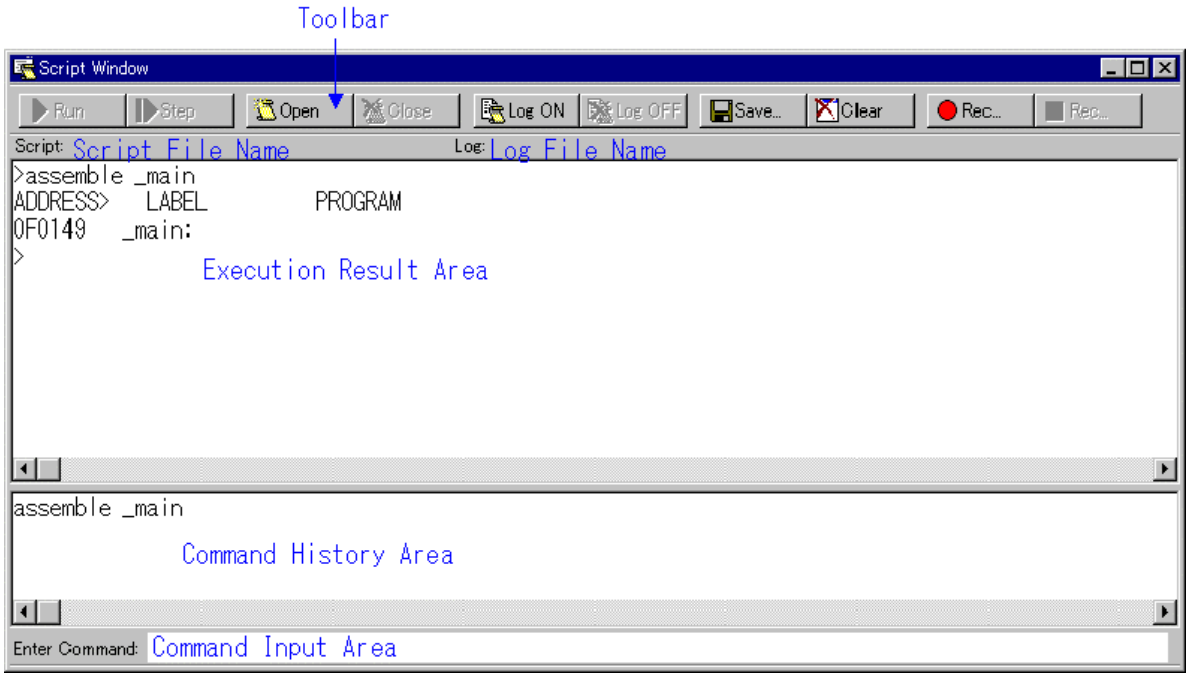
Menu	Menu Options	Function
Option	Font...	Change font.
	Jump New window	Displays the specified function on Program Window. Displays the specified function on a new Source Window.

These menus can be selected even by the short cut menu by a right click in the window.

1.11 Script Window

The Script Window displays the execution of text -format script commands and the results of that execution.
Script commands can be executed using a script file or interactively. You can also write script commands in the script file so that they are automatically executed. The results of script command execution can also be stored in a previously specified log file.

1.11.1 Configuration of Script Window



- The Script Window has a view buffer that stores the results of executing the last 1000 lines. The results of execution can therefore be stored in a file (view file) without specifying a log file.
- When a script file is opened, the command history area changes to become the script file display area and displays the contents of the script file. When script files are nested, the contents of the last opened script file are displayed. The script file display area shows the line currently being executed in inverse vide.
- When a script file is open, you can invoke script commands from the command input area provided the script file is not being executed.
- The Script Window can record the history of the executed commands to a file. This function is not the same as the log function. This function records not the result but only the executed

commands, so the saved files can be used as the script files.

1.11.2 Extended Menus

The Script window provides the following menu when being active. (This menu is called Script window option.)

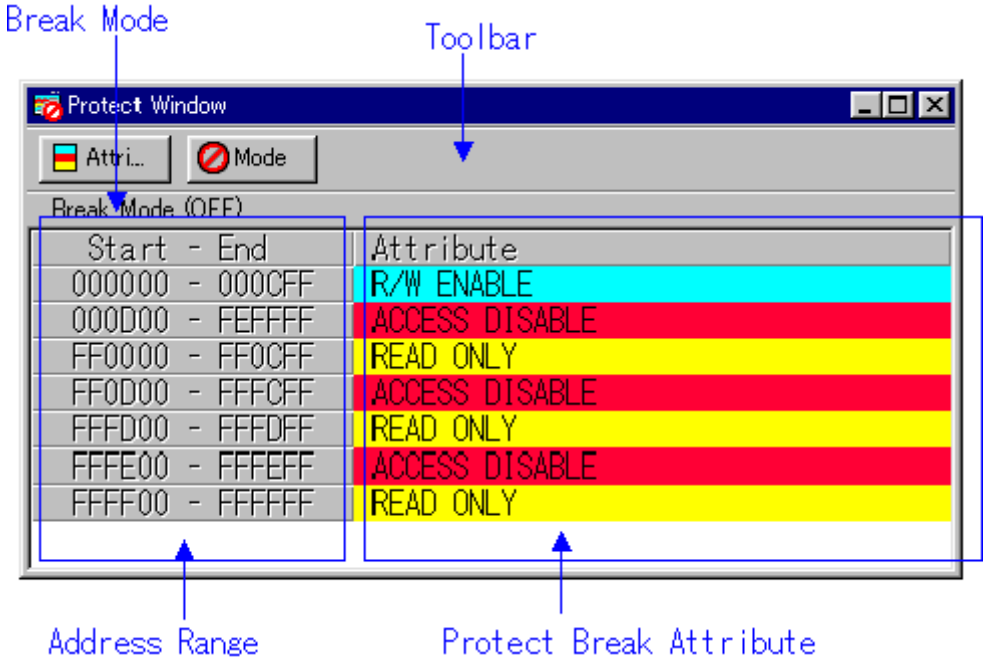
Menu	Menu Options	Function
Option	Font...	Change font.
	Script Open... Run Step Close	Script file operations. Open script file. Run script file. One-step execution of script file. Close script file.
	View Save... Clear	View buffer operations. Save view buffer file. Clear view buffer .
	Log On... Off	Log file operations. Open log file (start output to file). Close log file (stop output to file).
	Record On... Off	Record the executed commands Record the executed commands to a file. Stop Recording the executed commands.

These menus can be selected even by the short cut menu by a right click in the window.

1.12 Protect Window

The Protect window sets the protect break (memory protect) function of the emulator PC7501.
You can specify the memory access attribute in 256-byte units to the entire memory space (16 Mbytes).

1.12.1 Configuration of Protect Window



- The protect break function is disabled at start of the debugger.
- Three types of protect break attributes are provided as below:
  - Access Disable (read/write disabled, display in red)

- Read Only (write disabled, display in yellow)
- Write Only (read disabled, display in green)

- R/W Enable (read/write enabled, display in sky blue)
- You can use the following two methods to set protect break.
  - Specify from the target program session information.*
  - Specify the memory attribute of the desired address range.*

*The specified range is adjusted to be aligned to the 256-byte boundary. (Example: If the specified range is 0x456 through 0x567, it is adjusted to 0x400 through 0x5FF.)

1.12.2 Extended Menus

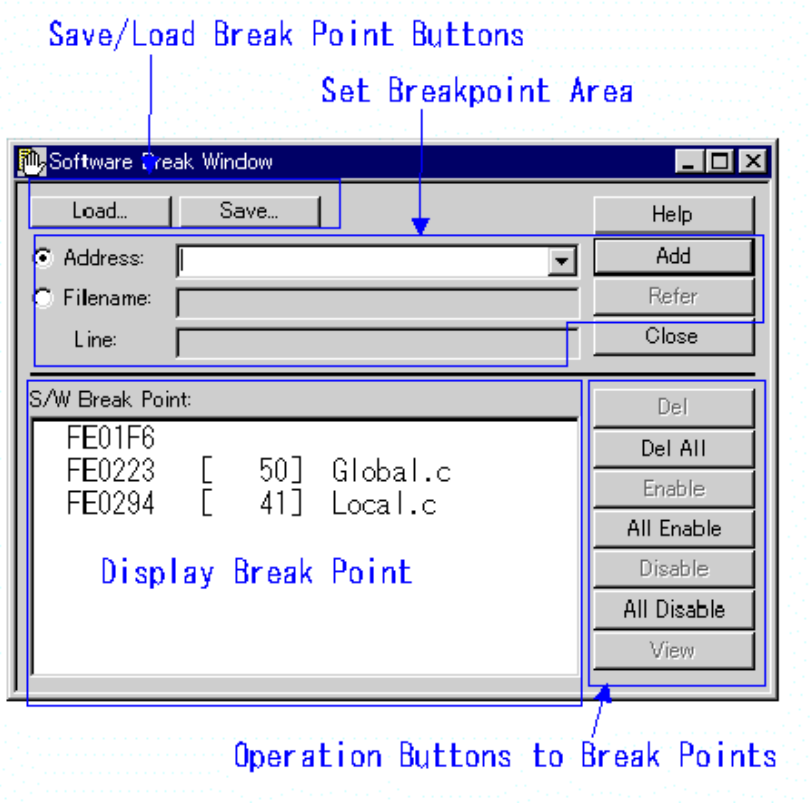
The Protect window provides the following menu when being active. (This menu is called Protect window option.)

Menu	Menu Options	Function
Option	Font...	Change font.
	Section	Set protect break attribute by the section information.
	Attribute...	Set protect break attribute.
	Mode	Switch break mode enable/disable.

These menus can be selected even by the short cut menu by a right click in the window.

1.13 Address Interrupt Break Window

The Address Interrupt Break Window allows you to set address interrupt break points.
This function stops the target program immediately before executing an instruction at a specified address. This function is realized by using the MCU's address interrupt interrupt. So that the address interrupt break function can only be used when the address match interrupt is not used in the user application.



- This window is available only when the address interrupt break function is used. Use the MCU

tab in the Init dialog box to specify whether or not to use the address interrupt break function. On this tab, deselect the Enable the Address Match Interrupt Break Function check box.

- The number of address interrupt breakpoints that can be set varies with each product. For details, refer to "Number of address interrupt breakpoints that can be set."
- Breakpoints can be specified by "Address" or "Filename + Line No.".
- If you have set multiple breakpoints, program execution stops when any one break address is encountered (OR conditions).
- You can clear, enable or disable breakpoints selected by clicking in the breakpoint display area. You can also enable and disable breakpoints by double-clicking on them.
- Click on the "Save" button to save the software break points in the file. To reload software break point settings from the saved file, click the "Load" button.
- If you load breakpoints from a file, they are added to any existing break points.

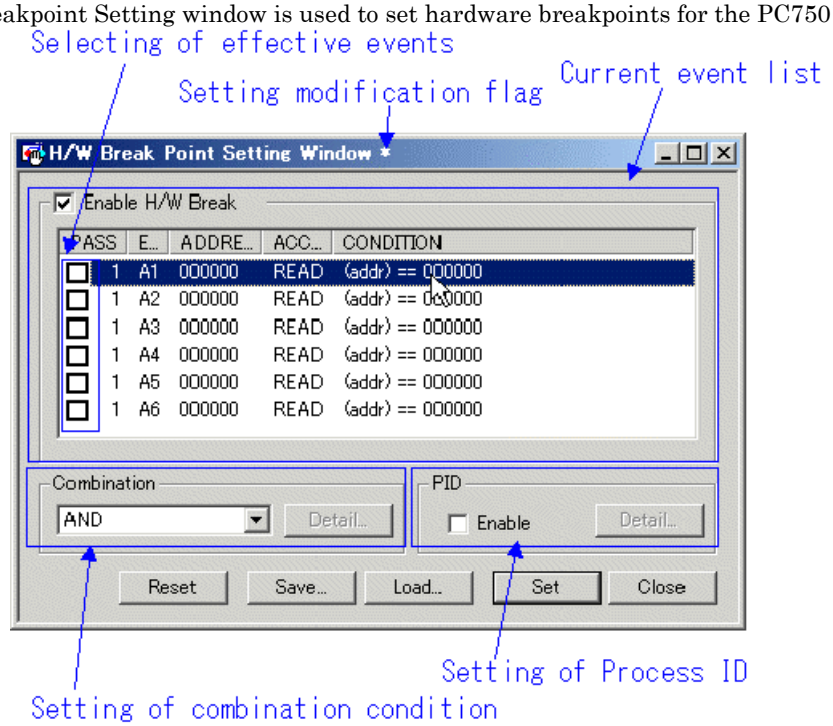
1.13.1 Command Button

The buttons at the right of the Address Interrupt Windowhave the following meanings.

Button Name	Content
Help	Display the help of this window
Load...	Load setting information from a file in which it was saved
Save...	Save the contents set in the window to a file
Add	Add address interrupt break
Refer	Open file selection dialog box
Close	Close the window
Del	Delete break point
Del All	Delete all break point
Enable	Enable break point
All Enable	Enable all break points
Disable	Disable break point
All Disable	Disable all break point
View	Shows software breakpoint positions in the Program Window

1.14 H/W Break Point Setting Window

The H/W Breakpoint Setting window is used to set hardware breakpoints for the PC7501 emulators.



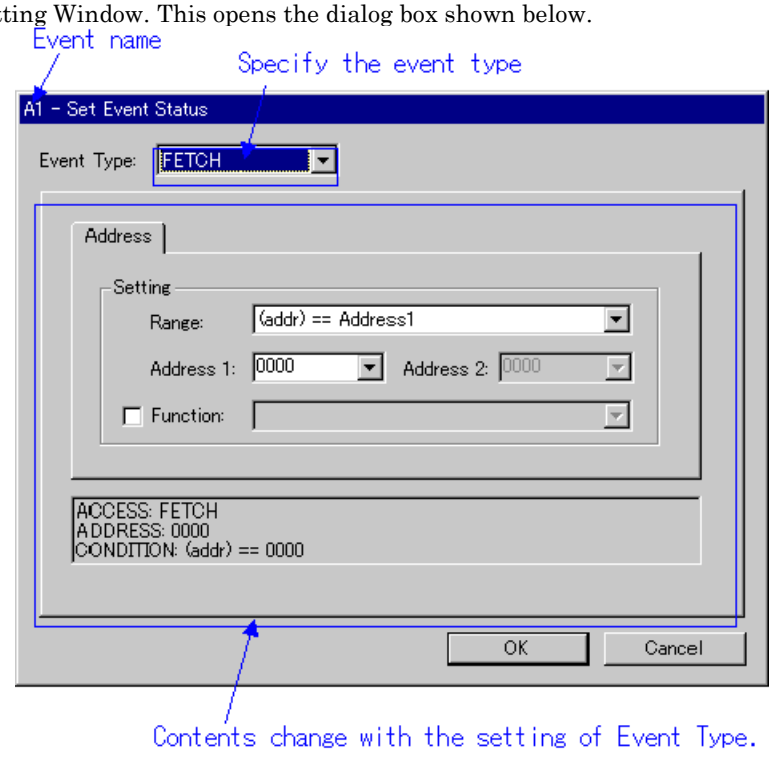
- The events listed below can be specified as break events. If the contents of events are altered, they are marked by an asterisk (*) on the title bar. The asterisks (*) are not displayed after setting up the emulator.

Event	Product Name	
	PD308F	PD30F
Fetch	Support	Support
Memory Access	Support	Support
Bit Access	Support	Support
Interrupt	Support	Support
External Trigger	Support	Support

- Events at up to eight points can be used. These eight events can be combined in one of the following ways:
- Break when all of the valid events are established (AND condition)
- Break when all of the valid events are established at the same time (simultaneous AND condition)
- Break when one of the valid events is established (OR condition)
- Break upon entering a break state during state transition (State Transition condition)
- At the time the debugger starts up, the hardware breaks have no effect.

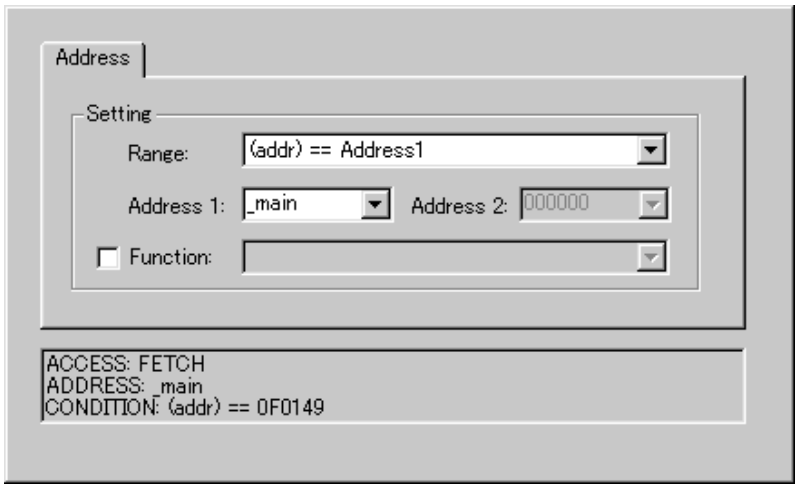
1.14.1 Specify the Break Event

To set events, double-click to select the event you want to set from the event setting area of the H/W Break Point Setting Window. This opens the dialog box shown below.

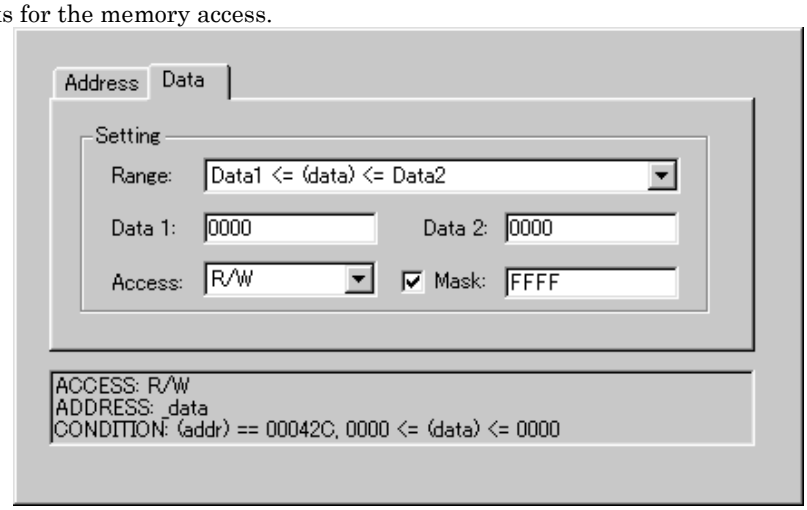


Following events can be set by specifying Event Type in this dialog box.

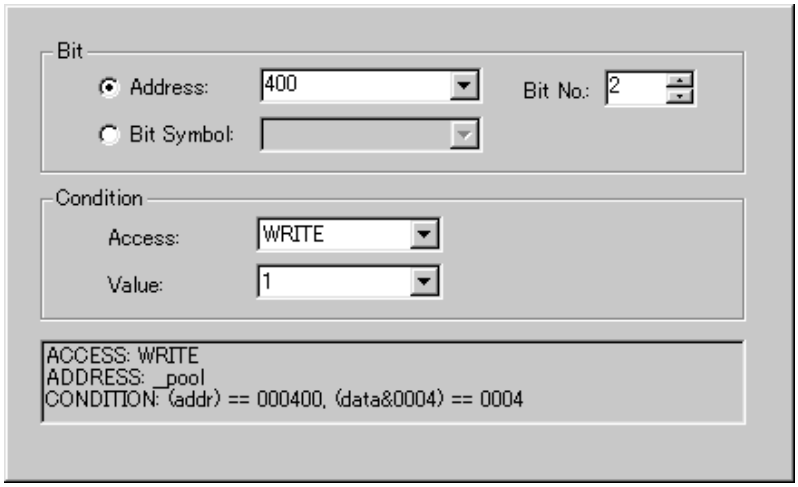
- **When FETCH is selected**
  Breaks for the instruction fetch.



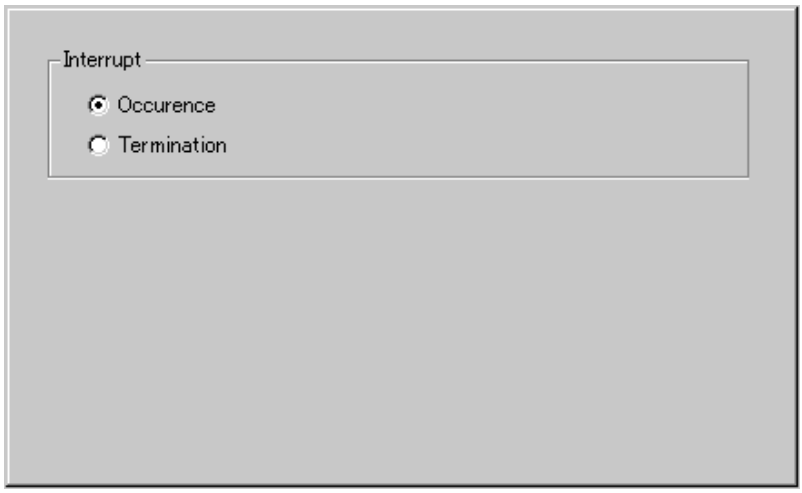
- **When DATA ACCESS is selected**
  Breaks for the memory access.



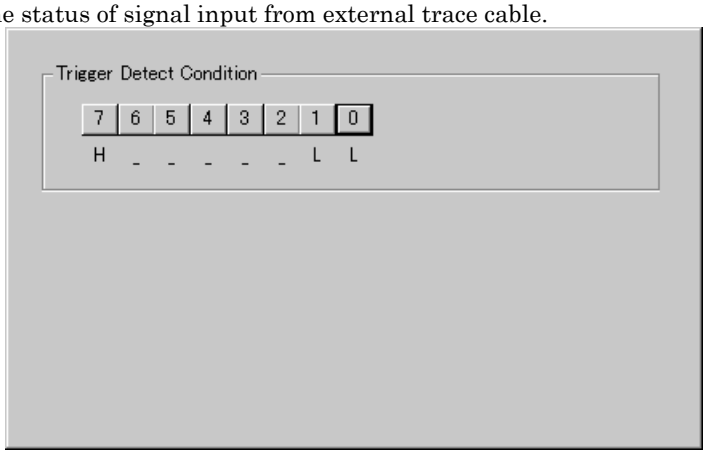
- **When BIT SYMBOL is selected**
  Breaks for the bit access.



- **When INTERRUPT is selected**
  Breaks for the interrupt occurrence or termination.



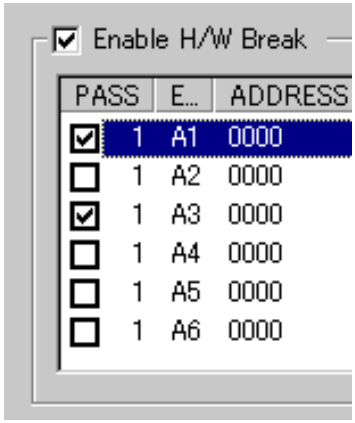
- **When TRIGGER is selected**
  Breaks for the status of signal input from external trace cable.



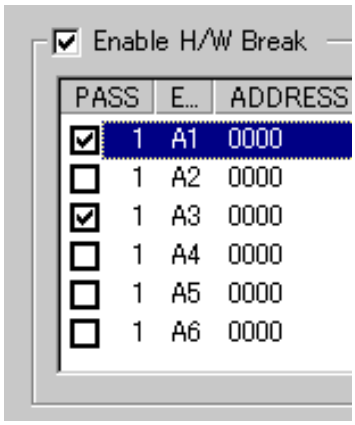
1.14.2 Specify the Combinatorial Condition

To specify a combinatorial condition, specify the desired condition from the combinatorial condition specification area.

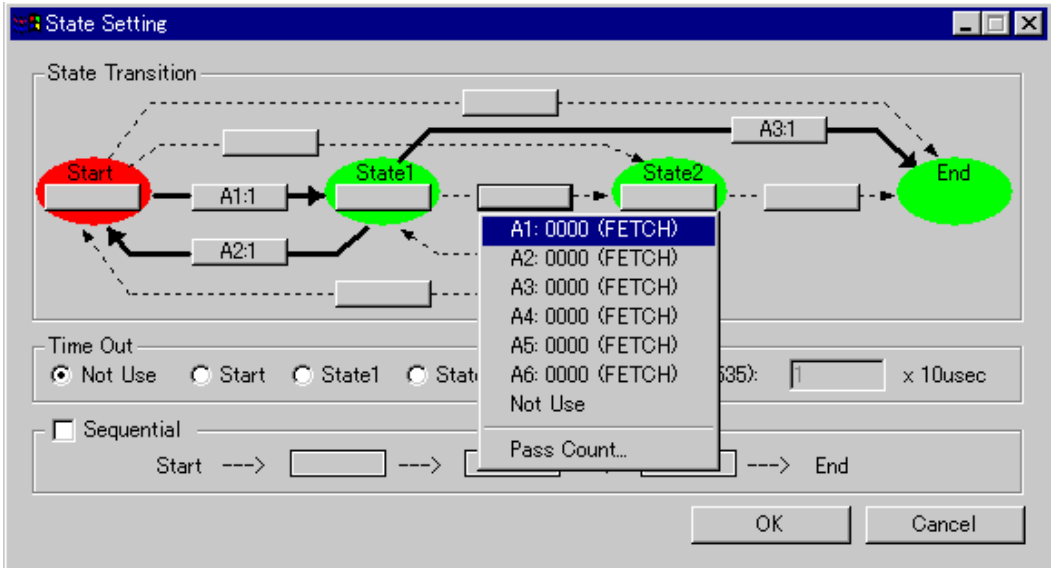
- **When AND or OR is selected**
  In the event specification area, the event used and a pass count for that event can be specified. To alter the pass count, while the event to alter is being selected, click the pass count value of that event.



- **When AND (Same Time) is selected**
  In the event specification area, the event used can be specified. No pass counts can be specified.

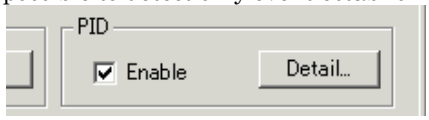


- **When State Transition is selected**
  Click the Details... button, and the dialog box shown below appears. Specification by a state transition diagram or sequential specification can be used. If the content of any event is altered, it is marked with an asterisk (*) on the title bar. Once conditions are set in the emulator, asterisks are not displayed. A time-out time in each state can also be specified.



1.14.3 Specify the Process ID

By specifying a process ID, it is possible to detect only event establishment under specific conditions.



Example: Enable only the event that occurs in a specific task when using the realtime OS.

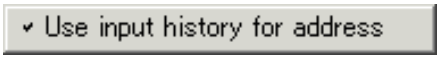
1.14.4 Command Button

The buttons at the bottom of the H/W Breakpoint Setting window have the following meanings.

Button Name	Content
Reset	Discards the contents being displayed in the window and loads contents from the emulator in which they were set.
Save...	Saves the contents set in the window to a file
Load...	Loads event information from a file in which it was saved
Set	Sends the contents set in the window to the emulator
Close	Closes the window

1.14.5 Extended Menus

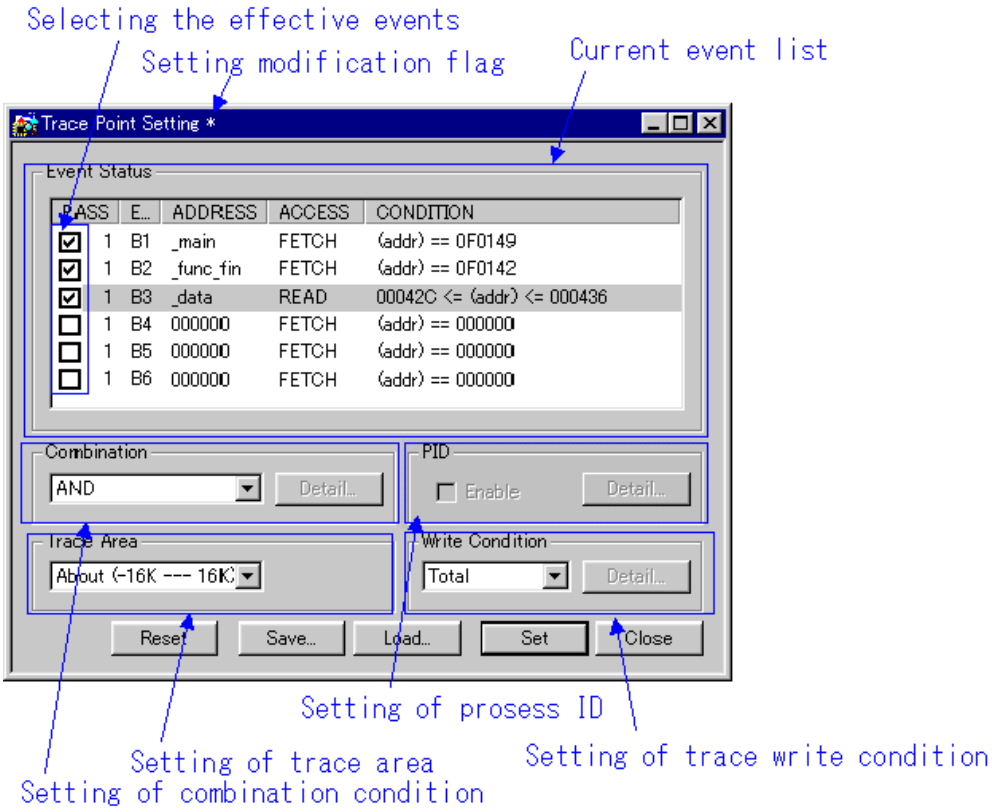
The H/W Breakpoint Setting Window has popup menus that can be brought up by right-clicking in the window.



If this menu is checked, input history for address input is available in an event setting dialog box opened from the H/W Breakpoint Setting Window. if not, the labels of program are listed for it.

1.15 Trace Point Setting Window

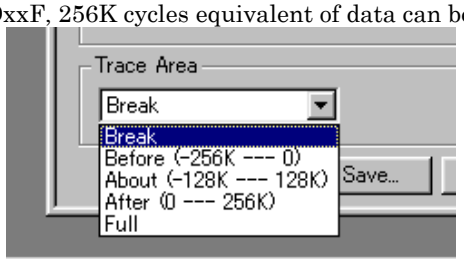
The Trace Point Setting window is used to set trace points.



Please refer to "1.14 H/W Break Point Setting Window" for the method of specifying the trace events and combination conditions.

1.15.1 Specify the Trace Range

For the emulator debugger PDxxF, 256K cycles equivalent of data can be recorded.

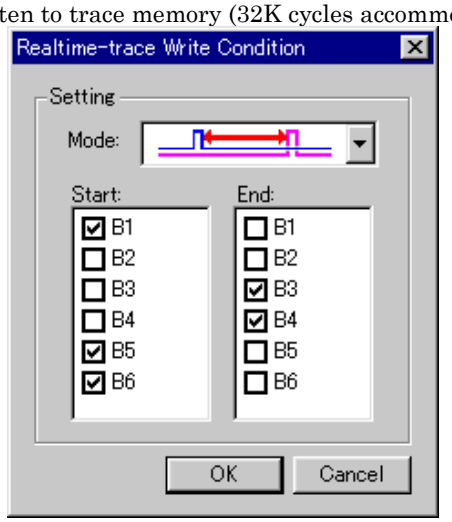


Break	Stores the 256K cycles (-256K to 0 cycles) to the point at which the target program stops.
Before	Stores the 256K cycles (-256K to 1 cycles) to the point at which the trace point is passed.
About	Stores the 256K cycles (-128K to 128K cycles) either side of the trace point.

After	Stores the 256K cycles (0 to 256K cycles) of trace data after the trace point.
Full	Stores the 256K cycles (-256K to 0 cycles) of trace data after the trace starts.

1.15.2 Specify the Trace Write Condition

Conditions for cycles to be written to trace memory (32K cycles accommodated) can be specified.



Total	Writes all cycles.
Pick up	Writes only the cycles where specified condition holds true.
Exclude	Writes only the cycles where specified condition does not hold true.

Also, following three write modes are supported.

	Only cycles where specified event is established
	Cycles from where specified event is established to where specified event is not established
	Cycles from where start event is established to where end event is established

1.15.3 Command Button

The buttons at the bottom of the Trace Point Setting window have the following meanings.

Button Name	Content
Reset	Discards the contents being displayed in the window and loads contents from the emulator in which they were set.
Save...	Saves the contents set in the window to a file
Load...	Loads event information from a file in which it was saved
Set	Sends the contents set in the window to the emulator
Close	Closes the window

1.15.4 Extended Menus

The Trace Point Setting Window has popup menus that can be brought up by right-clicking in the window.

If this menu is checked, input history for address input is available in an event setting dialog box opened from the Trace Point Setting Window. if not, the labels of program are listed for it.

1.16 Trace Window

The Trace window displays the measurement result of the real time trace.
The Trace window provides the three display modes as below:

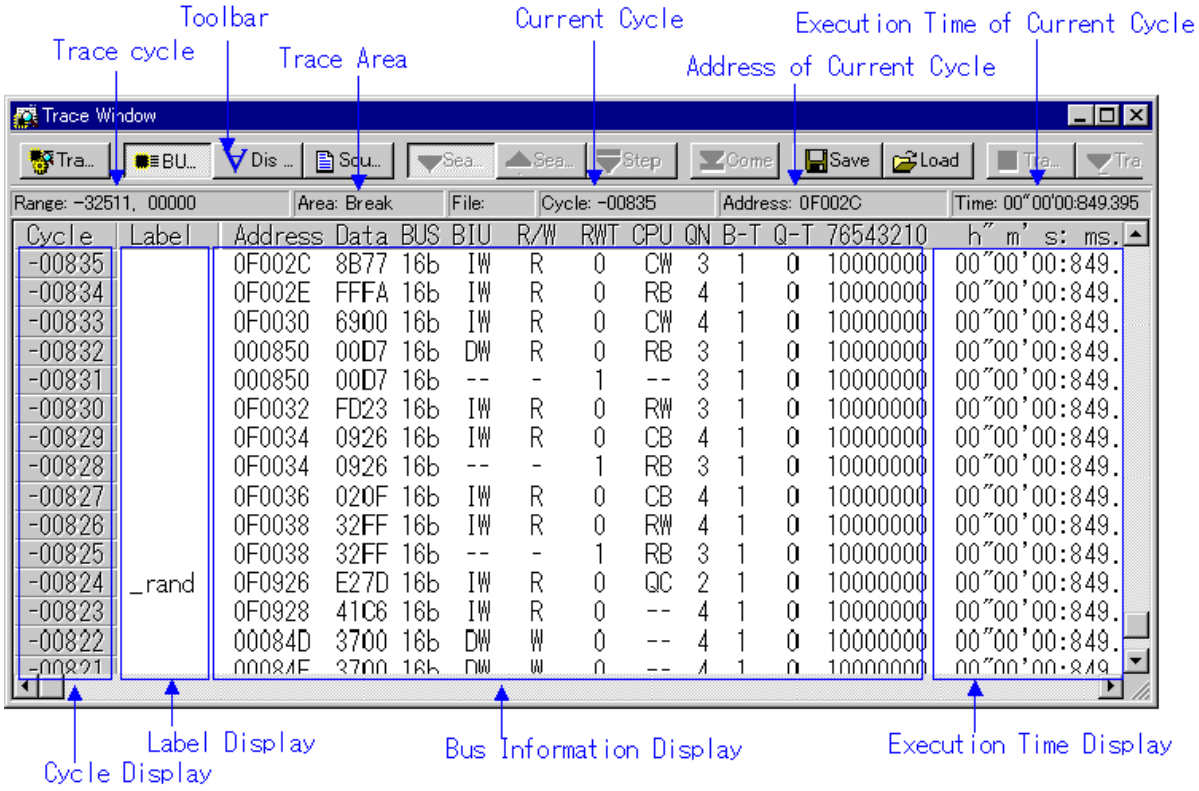
- **Bus Mode**
  Allows you to reference the bus information by cycle. The information is displayed in the order of execution path.
- **Disassemble + Data access Mixed Mode**
  The executed instructions and the accessed data contents can be referenced together. The contents are displayed in order of execution passes.
- **Disassemble Mode**
  Allows you to reference the executed command. The commands are displayed in the order of execution path.
- **Source Mode**
  Allows you to reference the source program execution path. Operate the buttons in the tool bar to reference the path.

The Trace window displays the measurement result when the real time measurement is completed. If the real time measurement has not been completed, the Trace window displays nothing.

By default, 256 K cycles before the target program is stopped are recorded.
To change the trace measurement range to the desired event position, set the required data in the Trace Point Setting window which enables more precise settings for trace events.
For details on the Trace Point Setting Window, see "1.15 Trace Point Setting Window".

1.16.1 Configuration of Bus Mode

The bus mode of the trace window is the following configurations.



- By double-clicking the cycle count display area, you can change the start cycle to be displayed.
- By double-clicking the Address line in the bus information display area, you can search the execution address.
- You can change the display ratio between the label display area and the bus information

display area, using the mouse.

- The lines in the bus information display area vary depending on the debugger that you are using.

1.16.1.1 Display of bus information on PD308F

From left to right, the contents are as follows:

- **Address**
  The status of the address bus
- **Data**
  The status of the data bus
- **BUS**
  The width of the external data bus ("8b" for an 8-bit data bus, and "16b" for a 16-bit data bus)
- **BHE**
  Indicates the status (0 or 1) of the BHE (Byte High Enable) signal.
  If BHE = 0, it means that the CPU is accessing an odd address.
- BIU
  This shows the status between the BIU (bus interface unit) and memory, and BIU and I/O.
  Representation BIU status

-	No access
WAIT	Executing wait instruction
RBML	Read access (bytes, ML on)
F	Fetch access
QC	Discontinuous Fetch access (queue buffer)
RWML	Read access (words, ML on)
INT	Interrupt acknowledge
RB	Read access (bytes)
WB	Write access (bytes)
DRB	Read access by DMA (bytes)
DWB	Write access by DMA (bytes)
RW	Read access (words)
WW	Write access (words)
DRW	Read access by DMA (words)
DWW	Write access by DMA (words)

- **R/W**
  Shows the status of the data bus ("R" for r ead, "W" for wr it e, "-" for no access).
- **RWT**
  This signal shows the effective position in the bus cycle ("0" when effective. Address, Data, and BIU signals are valid when RWT is "0".
- **CPU, OPC, OPR**
  This shows the signal between CPU and BIU. In the column “CPU”, the data shows whether CPU accesses BIU or not . In the Column “OPC”, the data shows the byte size of read operat ion code. In the Column “OPR”, the data shows the byte size of read operand.

Representation			Status	
CPU	OPC	OPR	Operation code size	Operand size
-	-	-	No accessing	
CPU	0	1	0byte	1byte
CPU	0	2	0byte	2bytes
CPU	0	3	0byte	3bytes
CPU	1	0	1byte	0byte
CPU	1	1	1byte	1byte
CPU	1	2	1byte	2bytes
CPU	1	3	1byte	3bytes

CPU	2	0	2bytes	0byte
CPU	2	1	2bytes	1byte
CPU	2	2	2bytes	2bytes
CPU	3	0	3bytes	0byte
CPU	3	1	3bytes	1byte
DMA	-	-	DMA accessing	
DMAT	-	-	DMA accessing(terminal count)	

- **QN**
  Shows the number of bytes stored in the instruction queue buffer in the range 0 to 8.
- **B-T**
  Shows the level of the external break trigger (the EXTIN7 pin of the external trace signal input cable). High level = "1", Low level = "0".
- **Q-T**
  Shows the level of the external trace trigger (the EXTIN6 pin of the external trace signal input cable). High level = "1", Low level = "0".
- **76543210**
  Shows the status of the 8-bit external signal (pins EXTIN0 to EXTIN7 of the external trace signal input cable). High level = "1", Low level = "0".
- **h" m' s: ms.us**
  Show the elapsed time from the target program beginning.

1.16.1.2 Display of bus information on PD30F

From left to right, the contents are as follows:

- **Address**
  The status of the address bus
- **Data**
  The status of the data bus
- **BUS**
  The width of the external data bus ("8b" for an 8-bit data bus, and "16b" for a 16-bit data bus)
- **BHE**
  Indicates the status (0 or 1) of the BHE (Byte High Enable) signal. If BHE = 0, it means that the CPU is accessing an odd address.
- **BIU**
  This shows the status between the BIU (bus interface unit) and memory, and BIU and I/O.

Representation	BIU status
-	No access
DMA	Data access other than a CPU cause such as DMA
INT	Start of INTACK sequence
IB	Instruction code read due to CPU cause (bytes)
DB	data access due to CPU cause (bytes)
IW	Instruction code read due to CPU cause (words)
DW	data access due to CPU cause (words)

- **R/W**
  Shows the status of the data bus ("R" for r ead, "W" for wr it e, "-" for no access).
- **RWT**
  This signal shows the effective position in the bus cycle ("0" when effective. Address, Data, and BIU signals are valid when RWT is "0".
- **CPU**
  This shows the signal between CPU and BIU.

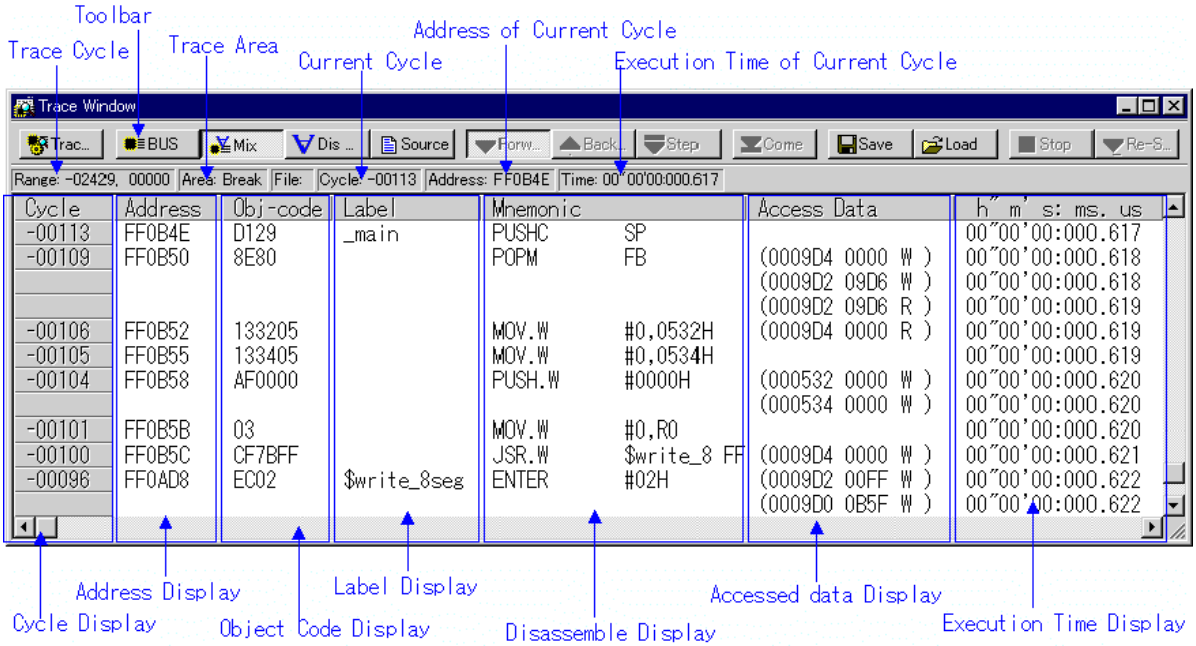
Representation	Status

-	No accessing
CB	Opecode read (bytes)
RB	Operand read (bytes)
QC	Instruction queue buffer clear
CW	Opecode read (words)
RW	Operand read (words)

- **QN**
  Shows the number of bytes stored in the instruction queue buffer in the range 0 to 4.
- **76543210**
  Shows the status of the 8-bit external signal (pins EXTIN0 to EXTIN7 of the external trace signal input cable).
  High level = "1", Low level = "0".
- **h" m' s: ms.us**
  Show the elapsed time from the target program beginning.

1.16.2 Configuration of Disassemble + Data access Mixed Mode

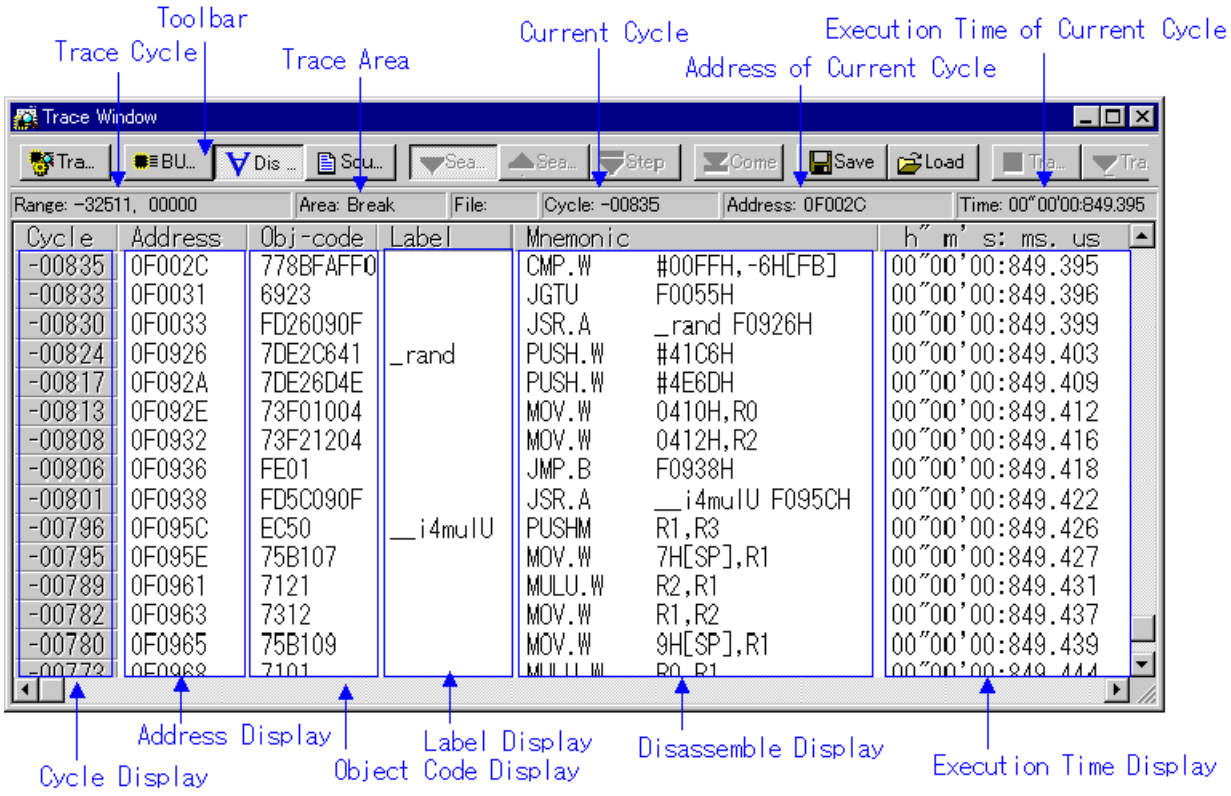
The disassemble + data access mixed mode of the trace window is the following configurations.



- The accessed memory contents are displayed in the accessed data display area. The contents are displayed in order of address, accessed data, and type of access (R for read, W for write), from left to right.
- The rest is the same as in disassemble mode.

1.16.3 Configuration of Disassemble Mode

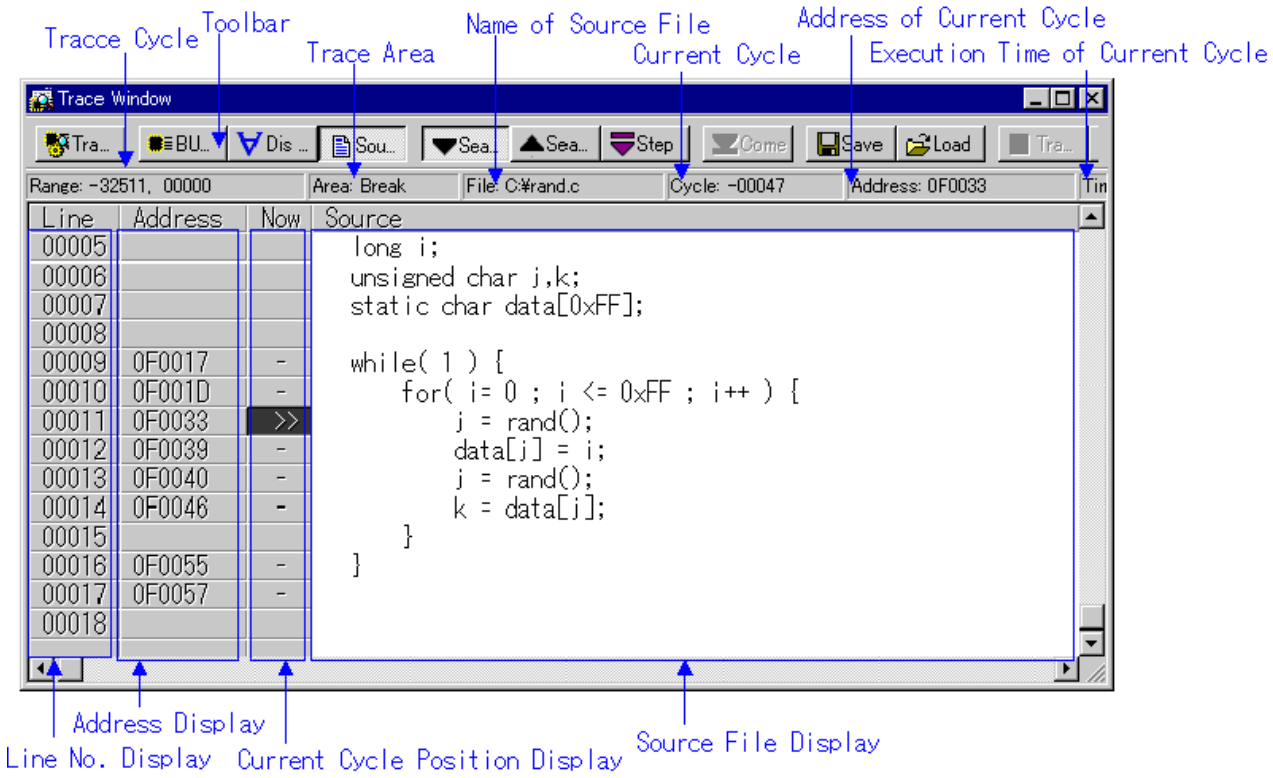
The disassemble mode of the trace window is the following configurations.



- By double-clicking the cycle count display area, you can change the start cycle to be displayed.
- By double-clicking the address display area, you can search the execution address.
- You can change the display ratio between the object code display area and the label display area, between the label display area and the inverted-assemble result display area, and between the inverted-assemble result display area and the execution time display area, using the mouse.

1.16.4 Configuration of Source Mode

The source mode of the trace window is the following configurations.



- You can switch "Display/Hide" for the line number display area/address display area/object code display area.
- By double-clicking the line number display area, you can change the source file to be displayed.
- By double-clicking the address display area, you can search the execution address.
- By clicking the source file display area and then clicking the Come button, you can search the address at the clicked position (Come search).
- In the reference cycle position display area, the current cycle position is displayed as ">>". A display of "-" indicates a line with the address information (a line for which Come search can be executed).

1.16.5 Extended Menus

The Trace window provides the following menu when being active. (This menu is called Trace window option.)

Menu	Menu Options	Function
Option	Font...	Change font.
	TAB...	Set tabs for source file display.
	View Cycle... Address Search... Source...	Change contents of display. Specify cycle. Search cycle by specifying address. Change by specifying source file.
	Mode Bus Mix Disasm Source	Change display mode. Select bus mode. Select disassemble + data access mixed mode. Select disassemble mode. Select source mode.

	Layout Line Area Address Area	Set layout. Switch display or non-display of line No. area. Switch display or non-display of address area.
	Trace Forward Backward Step Come	Search trace results. Search forward (in direction of execution). Search backward (in reverse direction of execution). Search one step (Step search). Search specified line (Come search).
	Save...	Save real-time trace data to file.
	Load...	Load real-time trace data to file.
	Trace Stop	Stop tracing.
	Trace Restart	Restart tracing.

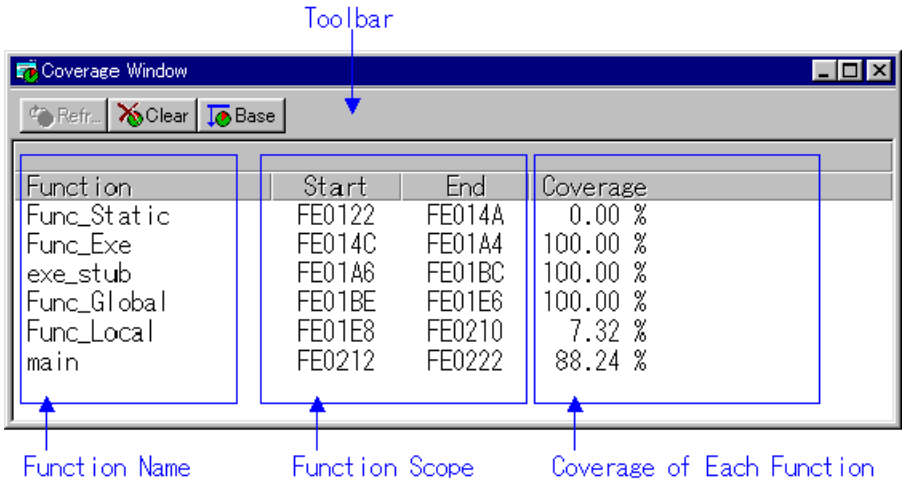
These menus can be selected even by the short cut menu by a right click in the window.

1.17 Coverage Window

The Coverage window allows you to reference the coverage measurement result of the functions of the target program downloaded.
Two types of windows are provided: the Coverage window in which you can check the start address/end address of the functions and coverage measurement results; and the Program window or Source window in which you can check execution/non-execution by source line.

- The coverage which can be measured is C0 coverage.
- The coverage measurement area is an any any blocks 1 to 32 (up to 8 Mbytes) beginning with the 256-byte boundarya.

1.17.1 Configuration of Coverage Window



- By double-clicking any function line, the corresponding function appears in the Source window.
- During coverage measurement, "-%" appears in the coverage display area.
- You can change the display ratio between the function name display area and the function range display area, using the mouse.

1.17.2 Extended Menus

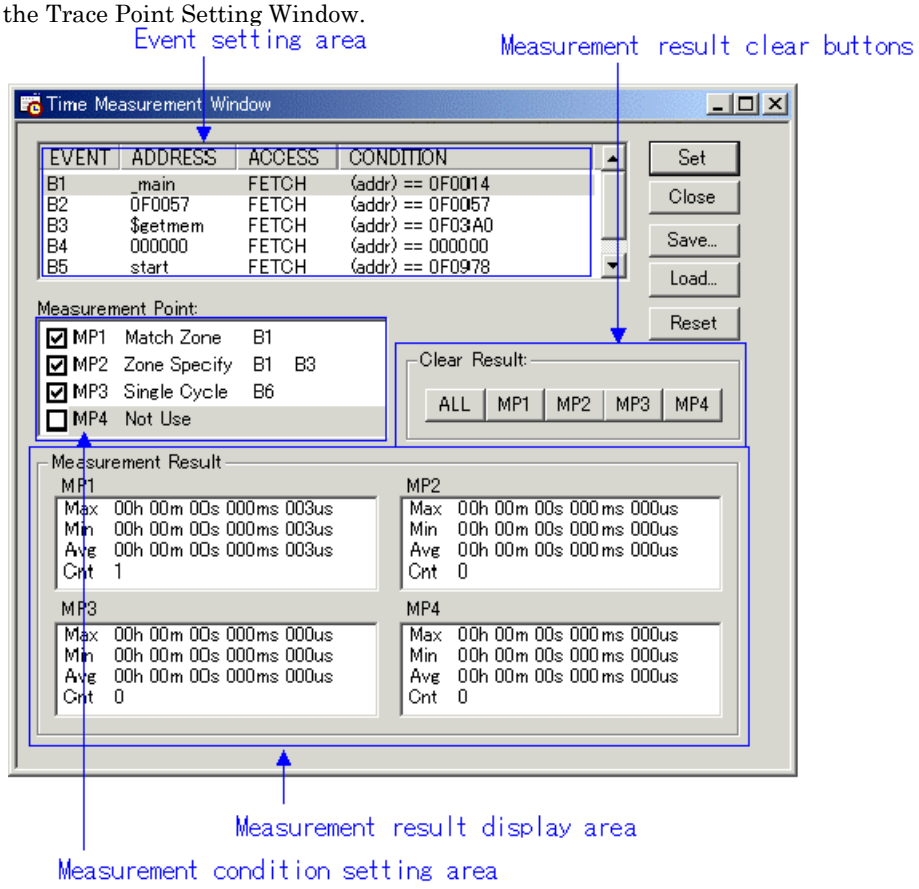
The Coverage window provides the following menu when being active. (This menu is called Coverage window option.)

Menu	Menu Options	Function
Option	Font...	Change font.

Select source file	Specify the source file to see the coverage
Refresh	Update display of coverage measurement result
Clear	Initialize coverage measurement result
Base	Change coverage measurement area
File Save... Load...	Input/output coverage measurement result file Save coverage measurement result file Load coverage measurement result file
Layout Address Area	Set Layout Turn address range display area on or off

1.17.3 Time Measurement Window

The Time Measurement window displays the minimum/maximum/average execution time and measurement count at any measurement point. The execution time of up to 4 measurement points can be measured simultaneously.
You can specify the event for the measurement condition in the same manner as when specifying events in the Trace Point Setting Window.



- The events listed below can be specified as measurement events. If the contents of events are altered, they are marked by an asterisk (*) on the title bar. The asterisks (*) are not displayed after setting up the emulator.

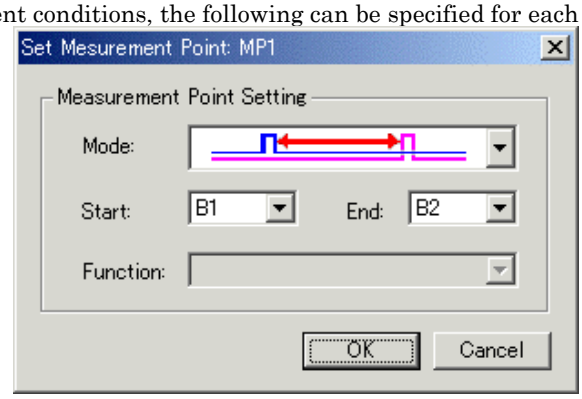
Event	Product Name	
	PD308F	PD30F
Fetch	Support	Support
Memory Access	Support	Support
Bit Access	Support	Support
Interrupt	Support	Support
External Trigger	Support	Support

ATTENTION

- The Trace Point Setting Window and the Time Measure Windows use the same resource of the emulator. If the event settings are modified in Time Measure Window, settings of the Trace Point Setting Window are modified, too.

1.17.4 Time Measurement Condition

For the time measurement conditions, the following can be specified for each measurement interval.



	Measures the time in an interval from where the start event is established till where the end event is established.
	Measures the time from where an event is established till where the next event is established.
	Measures the time from where an event is established till where the event is not established.
	Measures the execution time of functions. The start address and the end address of the function are automatically registered for the start event and the end event, respectively. The measurement result includes the execution time of other functions that have been called from within the specified function.
	Measures the execution time of functions. The start address and the end address of the function are automatically registered for the start event and the end event, respectively. The measurement result does not include the execution time of other functions that have been called from within the specified function.

1.17.5 Command Button

The buttons at the right side of the Time Measurement window have the following meanings.

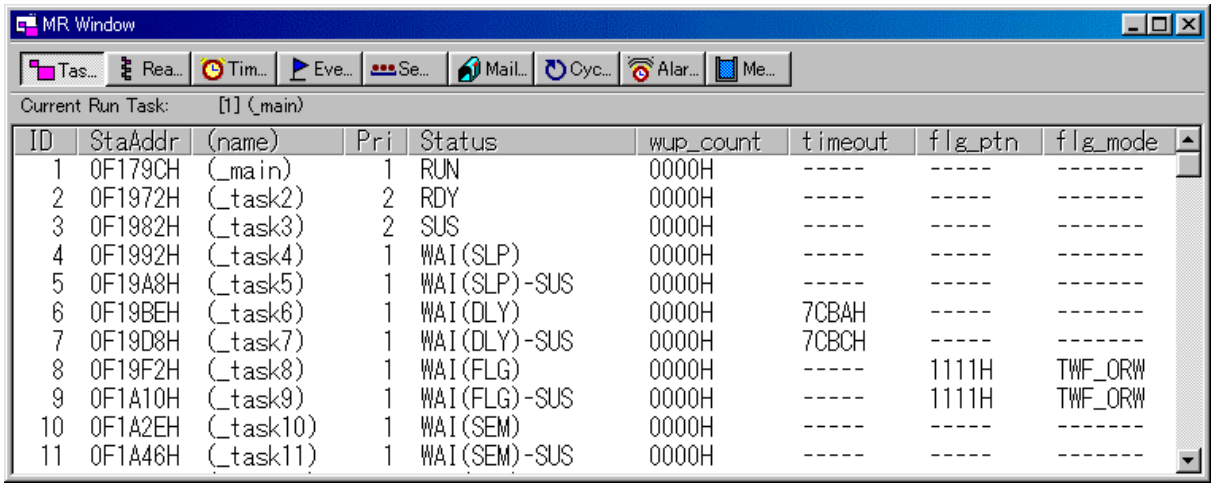
Button Name	Content
Reset	Discards the contents being displayed in the window and loads contents from the emulator in which they were set.
Save...	Saves the contents set in the window to a file
Load...	Loads event information from a file in which it was saved
Set	Sends the contents set in the window to the emulator
Close	Closes the window

1.18 MR Window

Use the MR Window to display the status of the realtime OS.
You can only use the MR Window when you have downloaded a program that uses the realtime OS (if the downloaded program does not use the MR, nothing is displayed in the MR Window when it is opened.)

1.18.1 Configuration of MR Window

You can open the MR window as many as the number of display modes (9 at maximum currently).



ID	StaAddr	(name)	Pri	Status	wup_count	timeout	flg_ptn	flg_mode
1	0F179CH	(_main)	1	RUN	0000H	-----	-----	-------
2	0F1972H	(_task2)	2	RDY	0000H	-----	-----	-------
3	0F1982H	(_task3)	2	SUS	0000H	-----	-----	-------
4	0F1992H	(_task4)	1	WAI(SLP)	0000H	-----	-----	-------
5	0F19A8H	(_task5)	1	WAI(SLP)-SUS	0000H	-----	-----	-------
6	0F19BEH	(_task6)	1	WAI(DLY)	0000H	7CBAH	-----	-------
7	0F19D8H	(_task7)	1	WAI(DLY)-SUS	0000H	7CBCH	-----	-------
8	0F19F2H	(_task8)	1	WAI(FLG)	0000H	-----	1111H	TWF_ORW
9	0F1A10H	(_task9)	1	WAI(FLG)-SUS	0000H	-----	1111H	TWF_ORW
10	0F1A2EH	(_task10)	1	WAI(SEM)	0000H	-----	-----	-------
11	0F1A46H	(_task11)	1	WAI(SEM)-SUS	0000H	-----	-----	-------

By clicking the desired button, the MR window display mode changes and the display data also changes.

By double-clicking the desired task line, you can display the context data of the task.
You can drag the cursor to change the width of the display area in each mode.

If the downloaded program does not use MR, you cannot select all the menu which will select the display mode.
The MR window supports the displays listed below.

Button	Contents
TSK	Task status
RQ	Ready queue status
TIM	Timeout queue status
FLG	Event flag status
SEM	Semaphore status
MBX	Mailbox status
CYH	Cyclic handler status
ALH	Alarm handler status
MPL	Memory pool status

If a target program created on MR30 V.1.00 is downloaded, the MPL mode cannot be used on MR30. (You cannot select the menu which changes the current mode to the MPL mode.)

1.18.2 Extended Menus

The MR window provides the following menu when being active. (This menu is called MR window option.)

Menu	Menu Options	Function
Option	Font...	Change font.

	Mode	Switch display mode.
	Task	Display Task status.
	Ready Queue	Display Ready queue status.
	Timeout Queue	Display Timeout queue status.
	Event Flag	Display Event flag status.
	Semaphore	Display Semaphore status.
	Mailbox	Display Mailbox status.
	Cyclic Handler	Display Cycle handler status.
	Alarm Handler	Display Alarm handler status.
	Memory Pool	Display Memory pool status.
	MR	
	Context...	Display Context.
	Layout	Set Layout .
	Status Bar	Switch display or non-display of status bar.

1.19 MR Trace Window

The MR Trace window measures the task execution history of a program using the real time OS and displays the result graphically.
In addition to the task execution history, a history of various other operations each are traced and displayed, including interrupt processing, task state transition, and system call issuance.
This window is available only when a target program which uses our real time OS (MRxx) is downloaded.

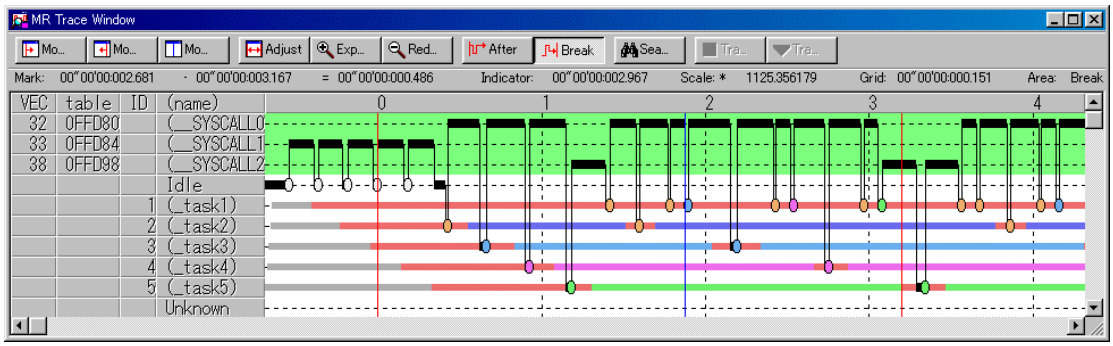
For MR308

- The history of the high-speed interrupt can not record and display.
- The recording and displaying the history of the OS-independent interrupt processing has the limitation. This function can detect the point of the OS-independent interruption, but not the end of it. This window regard the end of the OS-independent interruption as other point of interruption. The other words, when an OS-independent interrupt occur, the information of the interrupt and the tasks under the influence of the interrupt (the task interrupted by the OS-independent interrupt and so on) is not correct. The history of the OS-independent interrupt is indicated in hatch style wide line.

For MR30

- For MR30, this window is available for V. 2.00 or later version. If a target program crated on MR30 V. 1.00 is downloaded, the MR Trace window will not function and not display any data.

1.19.1 Configuration of MR Trace Window



The content of each item is as follows.

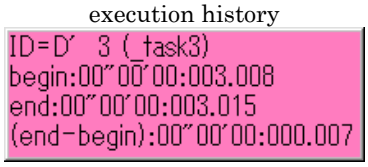
For PD308F/PD30F

Items	Contents
VEC*1	Indicates a software interrupt number.

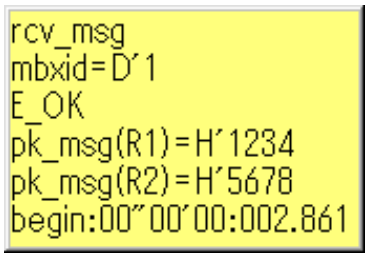
table	Indicates the interrupt vector table number.
ID	Indicates a task ID number.
(name)	Indicates an interrupt routine name, task name, idle processing (display "idle"), and unknown name(displayed "unknown").

When moving the mouse to the information displayed in the window, the pop up window as below is opened, showing the detailed information.

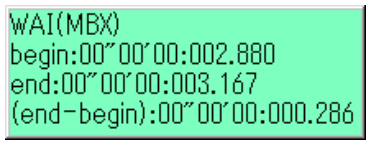
Interrupt handling or task execution history



System call issue history



Task state transition history



Following information is displayed in the status bar.

- Time value at which start marker is positioned
- Time value at which end marker is positioned
- Time width of a range indicated by start and end markers
- Time value at which indicator is positioned
- Scale factor of display
- Time width of grid line interval
- Range of measurement (trace) result

The grid lines are displayed using the start marker as the radix point.
The grid lines are displayed using the start marker as the radix point. The scale is displayed, using the time at which the start marker is positioned as 0, with the left (forward in time) set to "minus" and the right (backward in time) set to "plus".
The grid lines allow you to roughly understand the interrupt occurrence cycle and process time.
The interval time width of the displayed grid lines appears in the "Grid" area of the status bar.

The time value in the MR Trace window means the execution elapsed time using the program execution start time as 0 in all the cases. On the contrary, the numeric value above the grid lines (scale) in the MR Trace window is a relative value using the start marker as 0 (the grid interval is specified in the Value dialog).
It has nothing to do with the time value. (This is provided so that you can see the window easily.)

Note

The software interrupt number[*1] is different according to product.

PD308F

MR308 has interrupt numbers (48 to 63) for the INT instruction reserved for issuing a system call. The interrupt routine names displayed for interrupt numbers 48 to 63 are shown below.

Interrupt Number	Interrupt Routine Name
63	(_SYSCALL0)
62	(_SYSCALL1)
61	(_sys_ret_int)
60	(_sys_dis_dsp)
59	(_sys_loc_cpu)
58	(_sys_ext_tsk)
57	(_SYSCALL2)
56	(_SYSCALL3)

55	(_SYSCALL4)
54 to 48	-

For details about which interrupt number is assigned to which system call, refer to the MR308 Reference Manual, "Assemble Language Interface."

PD30F

MR30 has interrupt numbers (32 to 47) for the INT instruction reserved for issuing a system call. The interrupt routine names displayed for interrupt numbers 32 to 47 are shown below.

Interrupt Number	Interrupt Routine Name
32	(_SYSCALL0)
33	(_SYSCALL1)
34	(_sys_ret_int)
35	(_sys_dis_dsp)
36	(_sys_loc_cpu)
37	(_sys_ext_tsk)
38	(_SYSCALL2)
39	(_SYSCALL3)
40	(_SYSCALL4)
41 to 47	-

For details about which interrupt number is assigned to which system call, refer to the MR30 Reference Manual, "Assemble Language Interface."

1.19.2 Extended Menus

The MR Trace window provides the following menu when being active. (This menu is called MR Trace window option.)

Menu	Menu Options	Function
Option	Font...	Change font
	Mark S	Move start marker into display screen area
	Mark E	Move end marker into display screen area
	Indicator	Move indicator marker into display screen area
	Adjust	Adjust display (by expanding range of start and end markers to full width of display area)
	Expand	Increase scale factor of display
	Reduce	Reduce scale factor of display
	After	Set measurement range condition to After
	Break	Set measurement range condition to Break
	Trace Stop	Stop measuring
	Trace Restart	Restart measuring
	Value...	Set various values
	Color...	Set various display colors
	Search...	Search for history of system calls issued
	Init Order	Initialization of the display order

These menus can be selected even by the short cut menu by a right click in the window.

1.20 MR Analyze Window

The MR Analyze window displays the result of the measurement data statistically analyzed within the range specified by the start marker and the end marker in the MR Trace window.
The MR Analyze window supports three display mode as below:

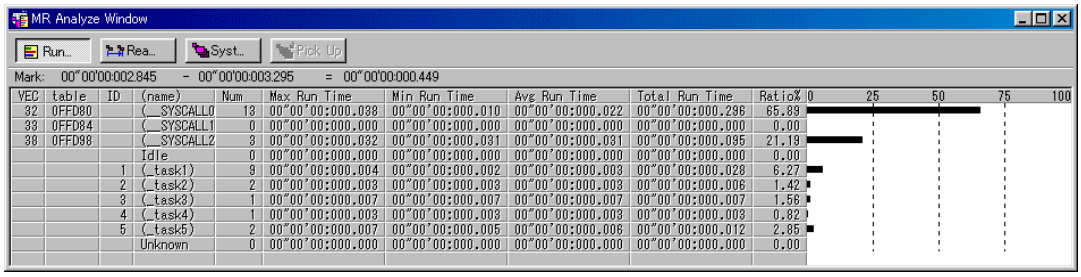
- CPU occupation state by interrupt/task
- Ready time by task
- List of system call issuance histories (You can extract and display the history based on the specific condition.)

The MR Analyze window functions together with the MR Trace window.

This window is available only when a target program using our real time OS (MRxx) is downloaded.

1.20.1 Configuration of CPU Occupancy Status Display Mode

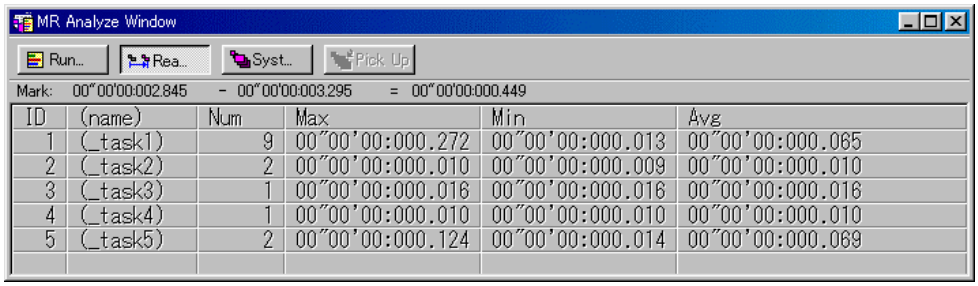
The CPU occupation state display mode is used to display the CPU occupation time and ratio by interrupt/task.
The MR Trace window shows the statistical results within the range specified by the start marker and end marker.



By clicking the maximum execution time/minimum execution time display area of each line, you can search interrupt to the clicked line or process history at the maximum/minimum execution time of the task.
The search result is pointed by the indicator which moves to the target position in the MR Trace window.

1.20.2 Configuration of Ready State Duration Display Mode

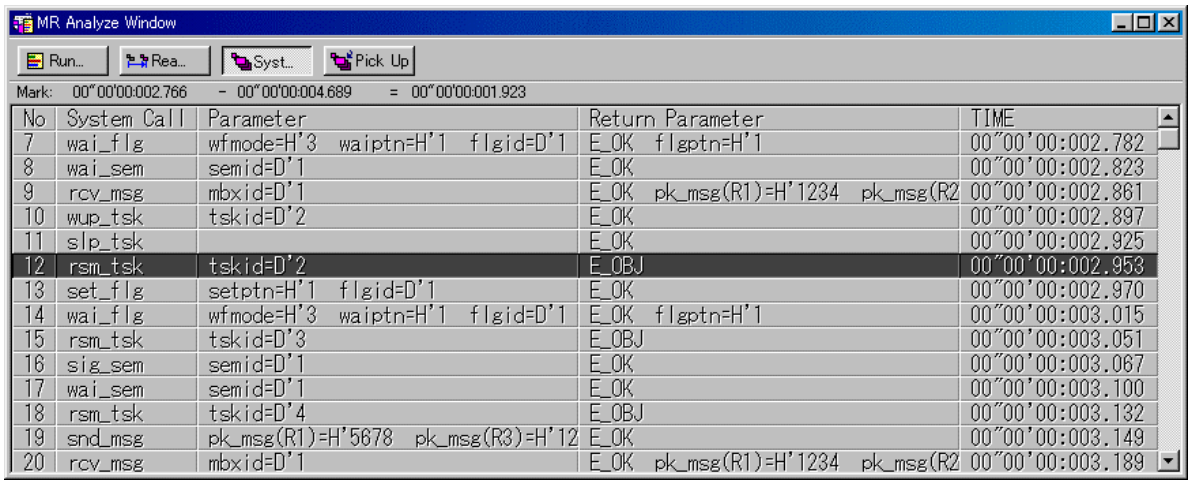
The ready state time display mode by task is used to display the results generated from statistical process of the time required from execution ready to transition to execution by task.
The statistical result is displayed within the range specified by the start marker and end marker in the MR Trace window.



By clicking the maximum ready time/minimum ready time display area of the desired line, you can search the process history of the maximum ready time/minimum ready time of the task corresponding to the clicked line.
The search result is pointed by the indicator which moves to the target position in the MR Trace window.

1.20.3 Configuration of System Call History Display Mode

The system call issuance history list mode is used to list the system calls issued.
The system call issuance history is listed within the range specified by the start marker and end marker in the MR Trace window.
The number indicates a numeric value counted from the top system call within the measurable range.



By clicking the desired line, you can search the system call issuance history to the clicked line.
The search result is pointed by the indicator which moves to the target position in the MR Trace window.

1.20.4 Extended Menus

The MR Analyze window provides the following menu when being active. (This menu is called MR Analyze window option.)

Menu	Menu Options	Function
Option	Font...	Change font.
	Run Time	Go to CPU occupancy status display mode
	Rdy->Run	Go to ready state duration display mode
	System Call	Go to system call history display mode
	Pick Up System Call...	Go to mode where history of system call issued is listed after extracting information according to specified conditions

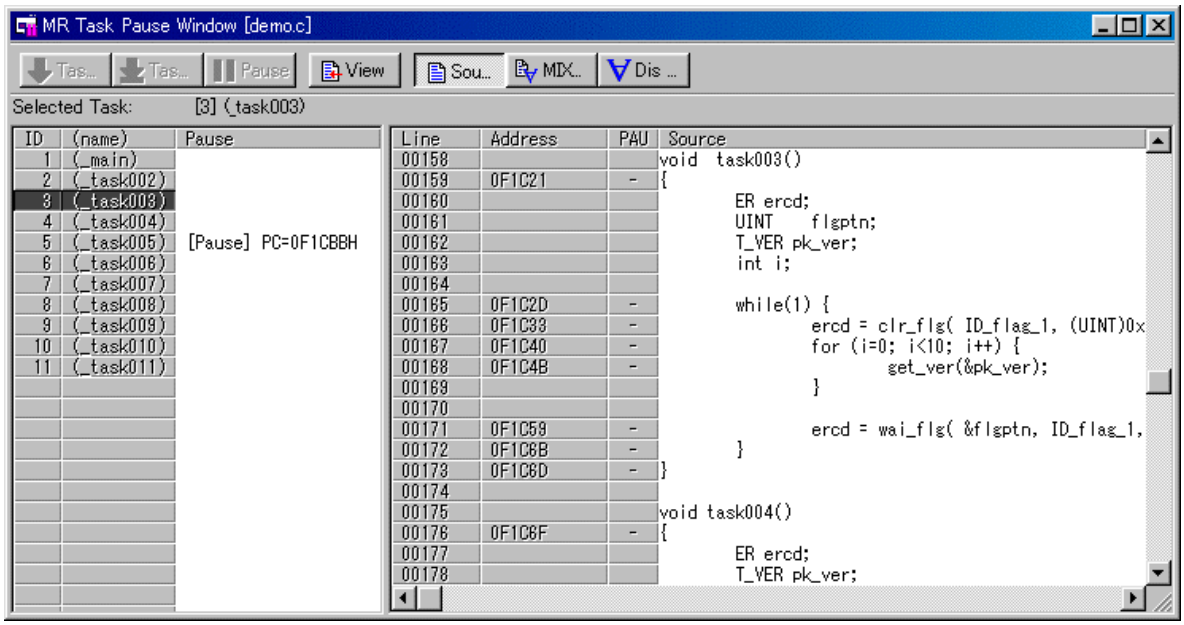
These menus can be selected even by the short cut menu by a right click in the window.

1.21 MR Task Pause Window

The MR task pause window realizes the task pause function of the real time OS (MR308/MR30).
You can pause and restart the specified task from this window. The window can be used only if the program containing the system and system programmer for the MR308/MR30 task pause function is downloaded.

The MR30 task pause function is supported with MR30 V.3.00 or later. Previously installed target programs cannot be used with this window.

1.21.1 Configuration of MR Task Pause Window



- The information (ID number, name, context PC value during "Pause") on all the tasks defined in the configuration file when creating the target program is displayed in the task pause display area. Select the target task for task pause from this display area.
- The specified program content is displayed in the task source display area. When performing task pause Come, specify the stop position using the cursor in this display area.

About Task Pause Function

The task pause function is to stop/restart only the specific task while executing the target system.
When using the task pause function, you can specify the specific task while executing all the other tasks and interrupts.
Also, as debugging is available, such as execution of Come, effective debugging is provided without affection to peripheral devices controlled by tasks or interrupts.
The following lists terminology definitions used in this section.

- **Pause Status**
  Indicates a state of the specific task when stopping the task during execution of the target using the MR Task Pause window.
- **Task Pause Pause Status**
  Indicates a process to pause the specific task during execution of the target using the MR Task Pause window.
- **Task Pause Go Status**
  Indicates a process to reset Pause for the specific task during execution of the target using the MR Task Pause window.
- **Task Pause Come Status**
  Indicates a process to pause the specific task during execution of the target using the MR Task Pause window.

1.21.2 Extended Menus

The MR Task Pause window provides the following menu when being active. (This menu is called MR Task Pause window option.)

Menu	Menu Options	Function

Option	Font...	Font change
	Pause Go Come Pause	Task pause function Task pause Go processing for target task Task pause Come processing for target task Task pause Pause processing for target task
	TAB... Color...	TAB setting for source file display of task source display area Display color setting for task source display area
	View Source... Address... Program Counter	Task source display area display contents change Display beginning from specified source file Display beginning from specified address or line number Display beginning from PC position*
	Mode Source Mode Mix Mode Disasm Mode	Task source display area display mode change Change to source display mode Change to mixdispla ymode Change to disassemble display mode
	Layout Line Area Address Area Code Area	Task source display area layout setting Line No. display area show/hide Address display area show/hide Object code display area show/hide

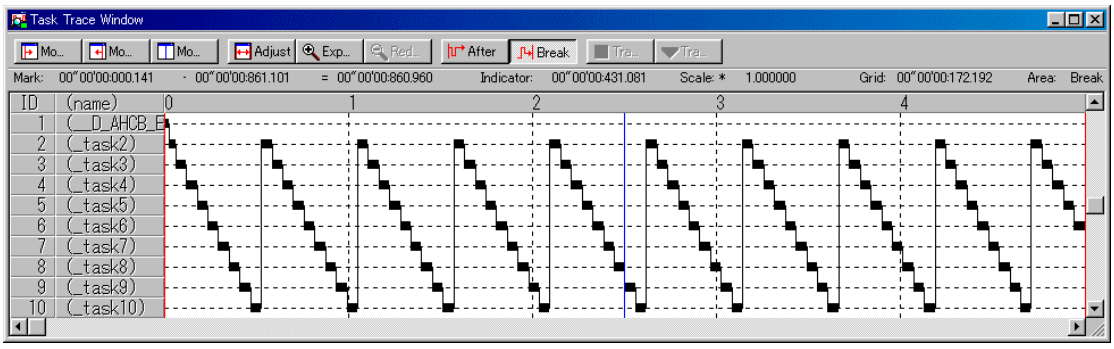
*Operation in the MR task window is as follows when the program display location is changed by PC position specification.

- **When the target task selected in the task pause display area is paused**
  -> Its display position is changed to the context PC position of the task.
- **When the target task selected in the task pause display area is in a state other than "pause"**
  -> The display position is not changed.
- **When the target task is selected from the task pause display area**
  -> The display position is changed to the position from the current program counter (in the same manner as the operation in the Source window).

1.22 Task Trace Window

The Task Trace window measures the task execution history of a program using the real time OS and display it graphically.
This window is available even when a target program using an OS other than our real time OS (MRxx) is downloaded.

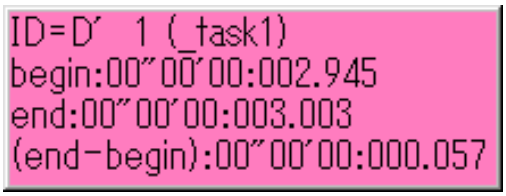
1.22.1 Configuration of Task Trace Window



The content of each item is as follows.

Items	Contents
ID	Indicates a task ID number.
(name)	Indicates an interrupt routine name, task name, idle processing (display "idle"), and unknown name(displayed "unknown").

When moving the mouse to the information displayed in the window, the pop up window as below is opened, showing the detailed information.



The following information is displayed in the status bar.

- Time value at the start marker position
- Time value at the end marker position
- Time interval between the start marker and the end marker
- Time value at the indicator position
- Display scale
- Time width at grid line interval
- Measurement (trace) range

The grid lines are displayed using the start marker as the radix point.
The scale is displayed, using the time at which the start marker is positioned as 0, with the left (forward in time) set to "minus" and the right (backward in time) set to "plus".
The grid lines allow you to roughly understand the interrupt occurrence cycle and process time.
The interval time width of the displayed grid lines appears in the "Grid" area of the status bar.

The time value in the Task Trace window means the execution elapsed time using the program execution start time as 0 in all the cases.
On the contrary, the numeric value above the grid lines (scale) in the Task Trace window is a relative value using the start marker as 0 (the grid interval is specified in the Value dialog). It has nothing to do with the time value. (This is provided so that you can see the window easily.)

1.22.2 Extended Menus

The Task Trace window provides the following menu when being active. (This menu is called Task Trace window option.)

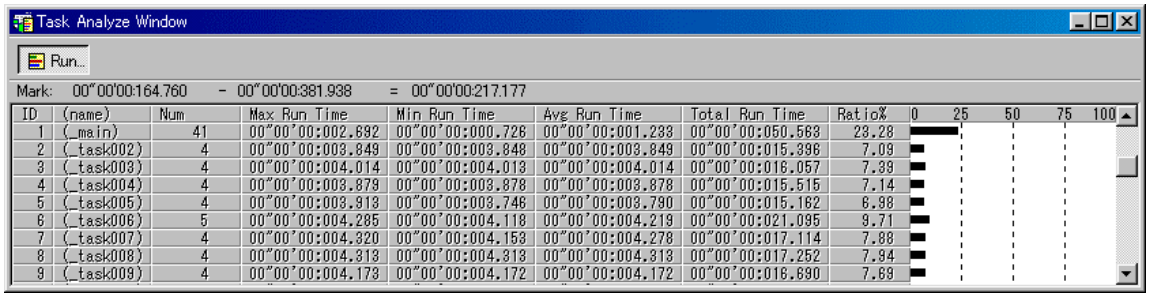
Menu	Menu Options	Function
Option	Font...	Change font.
	Mark S	Move start marker into display screen area
	Mark E	Move end marker into display screen area
	Indicator	Move indicator marker into display screen area
	Adjust	Adjust display (by expanding range of start and end markers to full width of display area)
	Expand	Increase scale factor of display
	Reduce	Reduce scale factor of display
	After	Set measurement range condition to After
	Break	Set measurement range condition to Break
	Trace Stop	Stop measuring
	Trace Restart	Restart measuring
	Value...	Set various values
	Color...	Set various display colors
	RTOS...	Set target RTOS information

These menus can be selected even by the short cut menu by a right click in the window.

1.23 Task Analyze Window

The Task Analyze window displays the result of the measurement data statistically analyzed within the range specified by the start marker and the end marker in the Task Trace window.
The Task Analyze window displays the CPU occupation state.
The Task Analyze window functions together with the Task Trace window.
This window is available even when a target program using an OS other than our real time OS (MRxx) is downloaded.
The CPU occupation state display mode is used to display the CPU occupation time and ratio by task.
This mode shows the statistical result within the range specified by the start marker and end marker in the Task Trace window.

1.23.1 Configuration of Task Analyze Window



By clicking the maximum execution time/minimum execution time display area of each line, you can search process history of the task for the clicked line at the maximum/minimum execution time.
The search result is pointed by the indicator which moves to the target position in the Task Trace window.

1.23.2 Extended Menus

The Task Analyze window provides the following menu when being active. (This menu is called Task Analyze window option.)

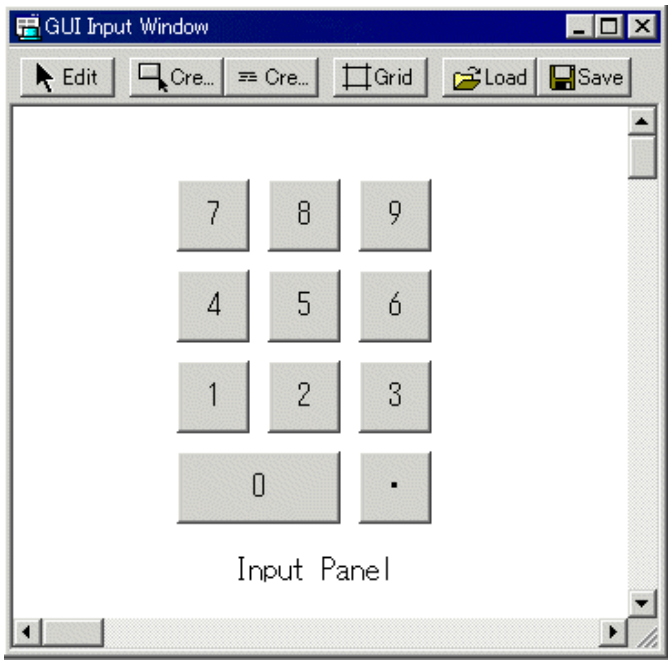
Menu	Menu Options	Function
Option	Font...	Change font.
	Run Time	Go to CPU occupancy status display mode

These menus can be selected even by the short cut menu by a right click in the window.

1.24 GUI Input Window

The GUI Input window allows you for port input by creating a user target system key input panel (button) in the window and clicking the created button.

1.24.1 Configuration of GUI Input Window



You can arrange the following parts on the input panel.

- Button
  A virtual port input or virtual interrupt (PDxxSIM only for the latter) can be executed at the time the button is pressed.
- Text
  Display the text string.

You can label (name) the created button.
You can also save the created input panel in a file and reload it.

1.24.2 Extended Menus

The GUI Input window provides the following menu when being active. (This menu is called GUI Input window option.)

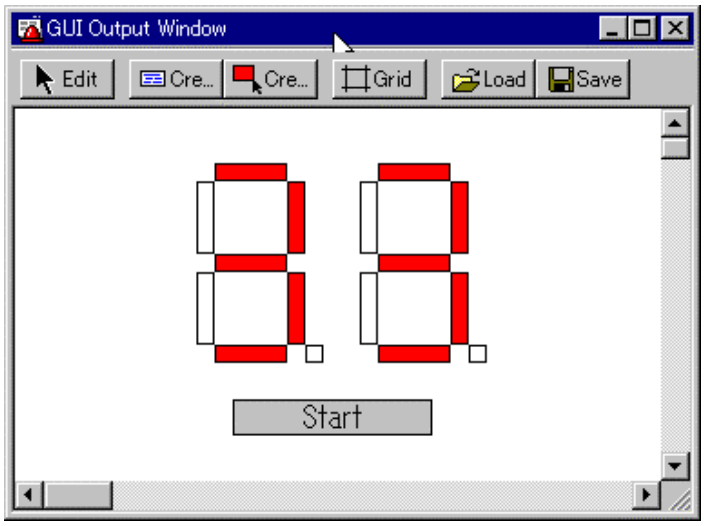
Menu	Menu Options	Function
Option	Set	Edits or moves button.
	Del	Deletes button.
	Copy	Copies button.
	Paste	Pastes button.
	Make Button	Creates button.
	Make Text	Create text label.
	Display Grid Line	Shows/hides grid line.
	Load...	Loads GUI input file.
	Save...	Saves GUI input file.

These menus can be selected even by the short cut menu by a right click in the window.

1.25 GUI Output Window

The GUI Output window allows you to implement the user target system output panel in the window.

1.25.1 Configuration of GUI Output Window



You can arrange the following parts on the output panel.

- Label (character string)
  Displays/erases a character string specified by the user when any value is written to the specified address (bit).
- LED
  Changes the display color of any area when any value is written to the specified address (bit). (Substitution for LED ON)
- Text
  Display the text character.

You can label (name) the created button.
You can also save the created output panel in a file and reload it.

1.25.2 Extended Menus

The GUI Output window provides the following menu when being active. (This menu is called GUI Output window option.)

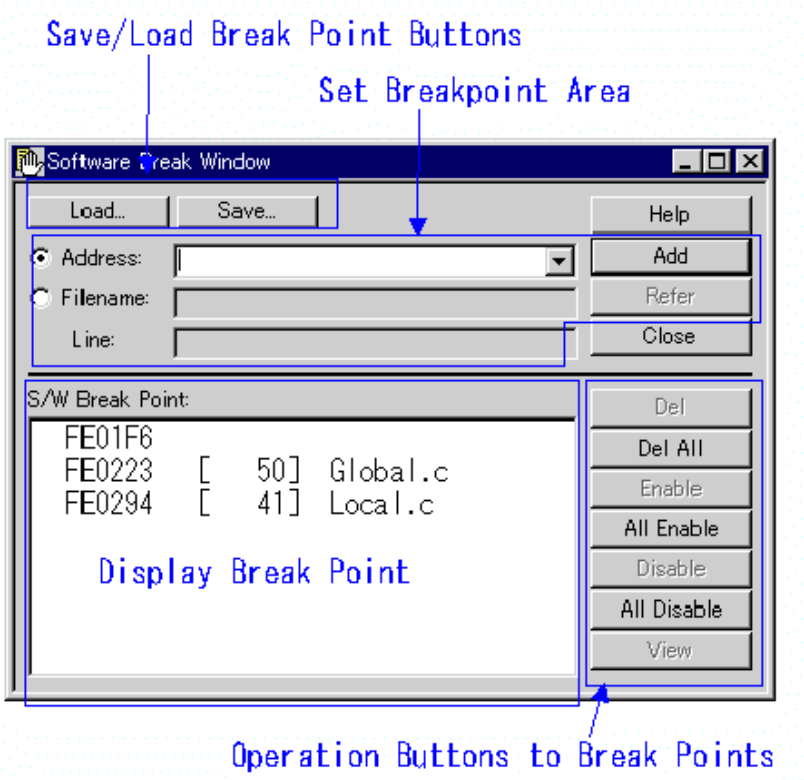
Menu	Menu Options	Function
Option	Set	Edits or moves parts.
	Del	Deletes parts.
	Copy	Copies parts.
	Paste	Pastes parts.
	Make Label	Creates label.
	Make LED	Creates LED.
	Make Text	Create text label.
	Display Grid Line	Shows/hides grid line.
	Load...	Loads GUI output file.
	Save...	Saves GUI output file.
	RAM Monitor	Display RAM monitor
	Ram Monitor Area...	Set RAM monitor area.
	Sampling Period...	Set sampling period for RAM monitor.

These menus can be selected even by the short cut menu by a right click in the window.

1.26 S/W Break Point Setting Window

The S/W Break Point Setting window allows you to set software break points.
Software breaks stop the execution of instructions immediately before the specified break point.
You can also enable and disable each of those break points.

1.26.1 Configuration of S/W Break Point Setting Window



- You can set up to 64 software break points.
- If you have set multiple software breakpoints, program execution stops when any one software break address is encountered (OR conditions).
- You can continue to set software breakpoints until you click the "Close" button to close the S/W Break Point Setting Window.
- You can clear, enable or disable software breakpoints selected by clicking in the software breakpoint display area. You can also enable and disable software breakpoints by double-clicking on them.
- Click on the "Save" button to save the software break points in the file. To reload software break point settings from the saved file, click the "Load" button. If you load software break points from a file, they are added to any existing break points.

1.26.2 Command Button

The buttons at the right of the S/W Break Point Setting window has the following meanings.

Button Name	Content
Help	Display the help of this window
Load...	Load setting information from a file in which it was saved
Save...	Save the contents set in the window to a file
Add	Add the break point
Refer	Open file selection dialog box
Close	Close the window
Del	Delete the break point
Del All	Delete all break points
Enable	Enable the break point
All Enable	Enable all break points
Disable	Disable the break point
All Disable	Disable all break points
View	Shows software breakpoint positions in the Program Window

2. Table of Script Commands

The following script commands are prepared.
The commands with yellow color displaying can be executed at run time.
The command to which "*" adheres behind is not supported according to the product.

2.1 Table of Script Commands

2.1.1 Execution Commands

Command Name	Short Name	Contents
Go	G	Program execution with breakpoints
GoFree	GF	Free run program execution
GoProgramBreak*	GPB	Run target program with software break point
GoBreakAt*	GBA	Run target program with software break point
Stop	-	Stops program execution
Status	-	Checks the operating status of the MCU
Step	S	Halts for user input until the specified time has elapsed
StepInstruction	SI	Step execution of instructions
OverStep	O	Overstep execution of source lines
OverStepInstruaction	OI	Overstep execution of instructions
Return	RET	Executes a source line return
ReturnInstruction	RETI	Executes an instruction return
Reset	-	Resets the target MCU
Time	-	Sets the run time display and checks the current setting

2.1.2 File Operation Commands

Command Name	Short Name	Contents
Load	L	Downloads the target program
LoadHex	LH	Downloads an Intel HEX-format file
LoadMot*	LM	Downloads a Motorola S-format file
LoadSymbol	LS	Loads source line/ASM symbol information
LoadIeee*	LI	Downloads IEEE-695 absolute-format files
Reload	-	Re-downloads the target program
UploadHex	UH	Outputs data to an Intel HEX-format file
UploadMot*	UM	Outputs data to a Motorola S-format file

2.1.3 Register Operation Commands

Command Name	Short Name	Contents
Register	R	Checks and sets a register value

2.1.4 Memory Operation Commands

Command Name	Short Name	Contents
DumpByte	DB	Displays the contents of memory (in 1-byte units)
*DumpWord**	DW	Displays the contents of memory (in 2-byte units)
*DumpLword**	DL	Displays the contents of memory (in 4-byte units)
*DumpDword**	DD	Displays the contents of memory (in 4-byte units)
SetMemoryByte	MB	Checks and changes memory contents (in 1-byte units)
*SetMemoryWord**	MW	Checks and changes memory contents (in 2-byte units)
*SetMemoryLword**	ML	Checks and changes memory contents (in 4-byte units)
*SetMemoryDword**	MD	Checks and changes memory contents (in 4-byte units)
FillByte	FB	Fills a memory block with the specified data (in 1-byte units)
*FillWord**	FW	Fills a memory block with the specified data (in 2-byte units)
*FillLword**	FL	Fills a memory block with the specified data (in 4-byte units)
*FillDword**	FD	Fills a memory block with the specified data (in 4-byte units)
Move	-	Moves memory blocks
MoveWord*	MOVEW	Moves memory blocks(in 2-byte units)

2.1.5 Assemble/Disassemble Commands

Command Name	Short Name	Contents
Assemble	A	Line-by-line assembly
DisAssemble	DA	Disassembles memory contents line by line
Module	MOD	Displays modules names
Scope	-	Sets and checks the effective local symbol scope
Section	SEC	Checks section information
*Bit**	-	Checks and sets bit symbols
Symbol	SYM	Checks assembler symbols
Label	-	Checks assembler labels
Express	EXP	Displays an assembler expression

2.1.6 Software Break Setting Commands

Command Name	Short Name	Contents
SoftwareBreak	SB	Sets and checks software breaks
SoftwareBreakClear	SBC	Clears software breaks
SoftwareBreakClearAll	SBCA	Clears all software breaks
SoftwareBreakDisable	SBD	Disables software breakpoints
SoftwareBreakDisableAll	SBDA	Disables all software breaks
SoftwareBreakEnable	SBE	Enables software breakpoints
SoftwareBreakEnableAll	SBEA	Enables all software breaks
BreakAt	-	Sets a software breakpoint by specifying a line No.
BreakIn	-	Sets a software breakpoint by specifying a function

2.1.7 Address Interrupt Break Setting Commands

Command Name	Short Name	Contents
ADdressInterruptBreak	ADIB	Sets and checks the address interrupt break

2.1.8 Hardware Break Setting Commands

Command Name	Short Name	Contents
HardwareBreak	HB	Sets and checks a hardware break
Protect	PT	Sets and checks protect breaks
BreakMode	BM	Sets and checks hardware break mode

2.1.9 Real-time Trace Commands

Command Name	Short Name	Contents
TracePoint	TP	Sets and checks a trace points
TraceData	TD	Realtime trace data display
TraceList	TL	Displays disassembled realtime trace data

2.1.10 Coverage Measurement Commands

Command Name	Short Name	Contents
Coverage	CV	Specifies and displays coverage measurement

2.1.11 Script/Log File Commands

Command Name	Short Name	Contents
Script	-	Opens and executes a script file
Exit	-	Exits the script file
Wait	-	Waits for an event to occur before command input
Pause	-	Waits for user input
Sleep	-	Halts for user input until the specified time has elapsed
Logon	-	Outputs the screen display to a log file
Logoff	-	Stops the output of the screen display to a log file

2.1.12 Program Window Control Commands

Command Name	Short Name	Contents
Func	-	Checks function names and displays the contents of functions
Up*	-	Displays the calling function
Down*	-	Displays a called function
Where*	-	Displays a function call status
Path	-	Sets and checks the search path
File	-	Checks a filename and displays the contents of that file

2.1.13 Map Commands

Command Name	Short Name	Contents
Map	-	Checks and sets mapping data

2.1.14 Clock Command

Command Name	Short Name	Contents
Clock	CLK	Checks and changes the clock
ClockControl*	CLKC	Checks and changes the XOUT, P86/XCOUT pin

2.1.15 WatchDog Timer Commands

Command Name	Short Name	Contents
WatchDogTimer	WDT	Sets and checks the usage condition of the watchdog timer

2.1.16 C Language Debugging Commands

Command Name	Short Name	Contents
Print	-	Check value of specified C variable expression
Set	-	Set specified data in specified C variable expression

2.1.17 Real-time OS Command

Command Name	Short Name	Contents
MR*	-	Displays status of realtime OS (MRxx)

2.1.18 Utility Commands

Command Name	Short Name	Contents
Radix	-	Sets and checks the radix for numerical input
Alias	-	Specifies and checks command alias definitions
UnAlias	-	Cancels the alias defined for a command
UnAliasAll	-	Cancels all aliases defined for commands
Version	VER	Displays the version No.
Date	-	Displays the date
Echo	-	Displays messages
Quit	-	Quits Debugger
CD	-	Window open
OpenWindow	-	Window open

2.2 Table of Script Commands (alphabetical order)

Command Name	Short Name	Contents
ADdressInterruptBreak	ADIB	Sets and checks the address interrupt break
Alias	-	Specifies and checks command alias definitions
Assemble	A	Line-by-line assembly
*Bit**	-	Checks and sets bit symbols
BreakAt	-	Sets a software breakpoint by specifying a line No.
BreakIn	-	Sets a software breakpoint by specifying a function
BreakMode	BM	Sets and checks hardware break mode
CD	-	Specifies and checks the current directory
Clock	CLK	Checks and changes the clock
ClockControl*	CLKC	Checks and changes the XOUT, P86/XCOUT pin
Coverage	CV	Specifies and displays coverage measurement
Date	-	Displays the date
DisAssemble	DA	Disassembles memory contents line by line
Down*	-	Displays a called function
DumpByte	DB	Displays the contents of memory (in 1-byte units)
*DumpDword**	DD	Displays the contents of memory (in 4-byte units)
*DumpLword**	DL	Displays the contents of memory (in 4-byte units)
*DumpWord**	DW	Displays the contents of memory (in 2-byte units)
Echo	-	Displays messages
Exit	-	Exits the script file
Express	EXP	Displays an assembler expression
File	-	Checks a filename and displays the contents of that file
FillByte	FB	Fills a memory block with the specified data (in 1-byte units)
*FillDword**	FD	Fills a memory block with the specified data (in 4-byte units)
*FillLword**	FL	Fills a memory block with the specified data (in 4-byte units)
*FillWord**	FW	Fills a memory block with the specified data (in 2-byte units)
Func	-	Checks function names and displays the contents of functions
Go	G	Program execution with breakpoints
GoBreakAt*	GBA	Run target program with software break point
GoFree	GF	Free run program execution
GoProgramBreak*	GPB	Run target program with software break point

HardwareBreak	HB	Sets and checks a hardware break
Label	-	Checks assembler labels
Load	L	Downloads the target program
LoadHex	LH	Downloads an Intel HEX-format file
LoadIeee*	LI	Downloads IEEE-695 absolute-format files
LoadMot*	LM	Downloads a Motorola S-format file
LoadSymbol	LS	Loads source line/ASM symbol information
Logoff	-	Stops the output of the screen display to a log file
Logon	-	Outputs the screen display to a log file
Map	-	Checks and sets mapping data
Module	MOD	Displays modules names
Move	-	Moves memory blocks
MoveWord*	MOVEW	Moves memory blocks(in 2-byte units)
MR*	-	Displays status of realtime OS (MRxx)
OpenWindow	-	Window open
OverStep	O	Overstep execution of source lines
OverStepInstruaction	OI	Overstep execution of instructions
Path	-	Sets and checks the search path
Pause	-	Waits for user input
Print	-	Check value of specified C variable expression.
Protect	PT	Sets and checks protect breaks
Quit	-	Quits Debugger
Radix	-	Sets and checks the radix for numerical input
Register	R	Checks and sets a register value
Reload	-	Re-downloads the target program
Reset	-	Resets the target MCU
Return	RET	Executes a source line return
ReturnInstruction	RETI	Executes an instruction return
Scope	-	Sets and checks the effective local symbol scope
Script	-	Opens and executes a script file
Section	SEC	Checks section information
Set	-	Set specified data in specified C variable expression
SetMemoryByte	MB	Checks and changes memory contents (in 1-byte units)
*SetMemoryDword**	MD	Checks and changes memory contents (in 4-byte units)
*SetMemoryLword**	ML	Checks and changes memory contents (in 4-byte units)
*SetMemoryWord**	MW	Checks and changes memory contents (in 2-byte units)
Sleep	-	Halts for user input until the specified time has elapsed
SoftwareBreak	SB	Sets and checks software breaks
SoftwareBreakClear	SBC	Clears software breaks
SoftwareBreakClearAll	SBCA	Clears software breaks
SoftwareBreakDisable	SBD	Disables software breakpoints
SoftwareBreakDisableAll	SBDA	Disables all software breaks
SoftwareBreakEnable	SBE	Enables software breakpoints
SoftwareBreakEnableAll	SBEA	Enables all software breaks
Status	-	Checks the operating status of the MCU
Step	S	Step execution of source line
StepInstruction	SI	Step execution of instructions
Stop	-	Stops program execution
Symbol	SYM	Checks assembler symbols
Time	-	Sets the run time display and checks the current setting
TraceData	TD	Realtime trace data display
TraceList	TL	Displays disassembled realtime trace data
TracePoint	TP	Sets and checks a trace points

UnAlias	-	Cancels the alias defined for a command
UnAliasAll	-	Cancels all aliases defined for commands
Up*	-	Displays the calling function
UploadHex	UH	Outputs data to an Intel HEX-format file
UploadMot*	UM	Outputs data to a Motorola S-format file
Version	VER	Displays the version No.
Wait	-	Waits for an event to occur before command input
WatchDogTimer	WDT	Sets and checks the usage condition of the watchdog timer
Where*	-	Displays a function call status

3. Error Messages

No.	Error Message	Notes and Action
0	INTERNAL ERROR:Unset err number	Contact your nearest distributor.

No.	Error Message	Notes and Action
200	Can't open more xxxxx window.	The maximum number of the specified window is already open.
201	Can't Create xxxxx window.	
202	PDxx is already exist.	
203	Project file (xxxxx) is broken.	
204	File not found (xxxxx).	
205	Path not found (xxxxx).	
206	Not enough memory.	
207	Can't execute.	
209	Failed to read/write data to the archive xxxxx (CODE: n).	
210	Failed to read/write data to the file xxxxx (CODE: n).	

No.	Error Message	Notes and Action
400	Can't change view mode.	The display starting address does not match the first line of the source file, or the specified source file cannot be found.
401	Can't find source file (xxxxx).	Specified source file was not found. Use the PATH command, or the [Environment] -> [Customize] menu items to specify the directory containing the source file.
402	Can't find search string (xxxxx).	The specified search string was not found between the starting position and end.
403	Line number of Source File (xxxxx) is over 2.	Because the source file has more lines than can be displayed, the file cannot be displayed in the Source Window. Switch to disassemble display mode.

No.	Error Message	Notes and Action
600	The address value is out of range.	
601	Can not open file(xxxxx).	
602	Can't find file (xxxxx).	
603	Can not save because the line number is over xxxxx.	
604	Can not save as the file (xxxxx). [system error: xxxxx]	
605	Can not edit this file (xxxxx) because it is being used by another process.	
606	The number of base addresses is over the limit	

	(num).	

No.	Error Message	Notes and Action
800	Value is out of range.	
801	Can't find the register information file.	
802	There's incorrect line in register information file.	Contact your nearest distributor.
803	Not enough memory.	
804	Description of expression is illegal.	

No.	Error Message	Notes and Action
1200	Address value is out range for scroll area.	
1201	The length of the set data is different from the length of the displayed data.	
1202	The value is out of range.The value which can be specified is 1 to 2.	
1203	Can not open file(string1).	
1204	Internal Error:Memory buffer is null.	
1205	Address value is out of range.	
1206	Start address is larger than end address.	

No.	Error Message	Notes and Action
1550	There is not enough memory to load the target program.	
1551	Can't open the target file	
1552	Failed to read or to load the target file. string1	
1553	The loading has stopped as your request.	
1554	The target file has not the specified format or it is broken.	
1555	Not found the debugging information.	
1556	Not found the debuging information.	
1557	The target file has wrong information. So can't read the file.	
1560	Can't find the scope.	
1561	Can't find the appropriate symbols.	
1562	Can't find the appropriate functions.	
1563	Can't find the appropriate sections.	
1564	Can't find the appropriate line information.	
1565	Can't find the appropriate source file.	
1566	Can't find the search paths.	
1567	There is no more symbols.	
1568	There is no more functions.	
1569	There is no more sections.	
1570	The name is invalid for registers.	
1571	The word (string1) is one of the resereved words. You can not specify it as symbol name.	
1572	The word (string1) has been still defined. You can not specify it as symbol name.	
1573	There is no information for the source files and the line numbers.	
1574	Bit symbols are not supported.	
1575	The word (string1) contains some illegal	

	characters. You can not specify it as symbol name.	
1580	Internal Error : unexpected symbol type has been specified.	
1581	Internal Error : an unexpected searching order has been specified.	
1582	Internal Error : not found the class where the download data is stored.	
1583	Internal Error : an unexpected file format has been specified.	
1584	Internal Error : The information for downloading has not been obtained. string1	
1585	Internal Error : Failed to regist the information to the debugging information data base. string1	

No.	Error Message	Notes and Action
1600	Can't add new watch point because it exceeds limit of watch point number. Max number is (num).	
1601	Address value is out of range.	
1602	Data value is out of range.	
1603	Bit value is out of range.	
1604	Can't save watch points.	

No.	Error Message	Notes and Action
1800	There are no symbol information.	
1801	The expression is too long.	
1802	Can't save c watch points.	
1803	Can't load c watch points.	
1804	Load is terminated because the file extension is different.	

No.	Error Message	Notes and Action
2000	Can't open Script File (filename).	
2001	Script File is not open.	
2002	Can't open Log File (filename).	
2003	Can't open more Log File.	
2004	Can't open Log File.	
2005	File (filename) is already log on.	
2006	Can't open View File (filename) for new/add.	
2007	Can't save command history.	

No.	Error Message	Notes and Action
2200	Address value is out of range.	
2201	Data value is out of range.	
2202	Start address is larger than end address.	
2203	Value is under (num).	
2204	Data value is out of range.	
2205	Data is not set.	
2206	Sampling period value is out of range.	
2207	Please fill in the blanks, and attach the contents	

	to the technical support sheet.	

No.	Error Message	Notes and Action
2400	Illegal endi. (filename line)	
2401	Illegal endw. (filename line)	
2402	INTERNAL ERROR:ER_BAT_EOF	
2403	Can't find endi. (filename line)	
2404	Line length is overflow. (filename line)	
2405	Nest level is overflow. (filename line)	
2406	Can't find Script File (filename).	
2407	Can't read Script File (filename).	
2408	Description is illegal. (filename line)	
2409	Can't find endw. (filename line)	
2410	The nest level exceeds the limit (num).	
2411	INTERNAL ERROR:ER_BAT_NONE	Contact your nearest distributor
2412	Illegal break. (filename line)	

No.	Error Message	Notes and Action
2600	Syntax error.	
2601	Command name is wrong.	
2602	Too many aliases.	
2603	You can register the only command name for alias.	
2604	Can't use the command now.	
2605	Can't up more.	
2606	Can't down more.	
2607	Can't set break point in this function.	
2608	The start address larger than the end address.	
2609	Can't register that token for alias.	
2610	Can't register that token for alias.	
2611	Can't find File (filename).	
2612	Data value is out of range.	
2613	Can't find the specified directory.	
2614	Can't open the window.	

No.	Error Message	Notes and Action
6000	INTERNAL ERROR:ER_ENV_END	Contact your nearest distributor.

No.	Error Message	Notes and Action
6200	SYMBOL file is illegal.	
6201	Loading is canceled.	
6202	Can't find SYMBOL file (filename).	
6203	Can't get enough memory.	
6204	Cannot open temporary file.	

No.	Error Message	Notes and Action
6402	Can't find symbol.	
6403	Description of expression is illegal.	
6404	Description is illegal.	

6405	Can't find scope.	
6406	Can't find symbol.	
6407	Can't find function.	
6408	Right hand side of the expression is illegal.	
6409	The Type of structure (union) are not same.	
6410	Can't assign.	
6411	Can't find type.	
6412	Not supported float (double) operation.	
6413	The operation does not be allowed to pointers.	
6414	The operation does not be allowed to the pointer.	
6415	Can't decrease by pointer.	
6416	Divided by 0.	
6417	The operator is not supported.	
6418	Type information is broken.	
6419	Left value must be the pointer.	
6420	Left value must be a structure or an union.	
6421	Can't find member.	
6422	Left value must be reference of a structure or an union.	
6423	Left value is illegal.	
6424	The operand must be a value.	
6425	The operand is able to be opposite sign.	
6426	Can't get address value.	
6427	The array variable is illegal.	
6428	The essential number of array is illegal.	
6429	The operand must be an address value.	
6430	Type casting for register variable is not be supported.	
6431	The type of type casting is illegal.	
6432	Type casting for that type is not be supported.	
6433	This expression can not be exchanged for some address value.	

No.	Error Message	Notes and Action
6601	Address value is out of range.	
6602	Target program is already stopped.	
6603	The number of break point is over the limit (num).	
6604	The break point isn't defined at that address.	
6605	Data value is out of range.	
6606	INTERNAL ERROR: ER_IN1_ILLEGAL_MODE has happen. (in xxxxx)	Contact your nearest distributor.
6607	Can't read/write, because there are no memory at that area.	
6608	Register value is out of range.	
6609	Can't execute that command, when the target program is running.	
6610	Start address is larger than end address.	
6611	STOP execution.	
6612	Can't search more on the stack.	

6613	Specified times of number is over than 65535.	
6614	INTERNAL ERROR: The memory of the odd number byte cannot be dumped by the Word access.	Contact your nearest distributor.
6615	Memory alignment error.	
6616	Illegal register is specified.	
6617	Already set address interrupt break.	
6618	The block number is out of range.	

No.	Error Message	Notes and Action
6800	The process is canceled.	
6801	Can't execute this command while some source windows are in editor mode.	

No.	Error Message	Notes and Action
10000	Cannot find source file (filename).	
10001	The number of lines of source file (filename) is over the limit (num).	
10002	The address value is out of range.	
10003	Cannot open file (filename).	
10004	Illegal file format.	
10005	Cannot read the file saved by simulator debugger.	
10006	Cannot read the file saved by emulator debugger.	
10007	Not enough memory for display all function.	

No.	Error Message	Notes and Action
10200	Operation code (code) not found.	
10201	File (filename) not found.	
10202	Duplicate event set in xxxxx.	
10203	File format error (filename).	

No.	Error Message	Notes and Action
10400	Can't execute more come instruction.	
10401	Can't execute more step instruction.	
10402	Cycle value is out of range.	
10403	Can't find that address.	
10404	Can not open file (filename).	
10405	Can not read file (filename).	
10406	The display mode is not able to change except the BUS mode. Trace data is not enough or is abnormal.	

No.	Error Message	Notes and Action
10600	Can't open BUTTON file (filename).	
10601	BUTTON file is illegal.	

No.	Error Message	Notes and Action
10800	Illegal file format.	
10801	Address value is out of range.	
10802	Data value is out of range.	

No.	Error Message	Notes and Action
11000	File format error (filename).	
11001	File (filename) not found.	
11002	Can't file (filename) open.	
11003	Failed to read/write data to the file (filename).	
11004	Failed to read/write data to the archive (filename).	
11005	Data value is out of range.	
11006	Function not found.	
11007	Bit Symbol not found.	
11008	Can not set trace points while program is running.	
11009	Specify BYTE access for ODD address.	

No.	Error Message	Notes and Action
11200	Already set hard ware break.	
11201	Combination of bus width and access condition.	
11202	Can't execute this command with PC4700L.	
11203	The start cycle larger than the end cycle.	
11204	HardwareBreak command cannot be used while H/W Break Point Setting Window opens.	
11205	These trace data can't disassemble.	
11206	TracePoint command cannot be used while Trace Point Setting Window, Time Measurement Window, MR Trace/Analyze Window or Task Trace/Analyze Window opens.	
11207	Cycle value is out of range.	
11208	The bit number is out of the range.	
11209	Address Interrupt Break is invalid.	
11210	ADdressInterruptBreak command cannot be used while ADI Break Point Setting Window opens.	
11211	No base address is set.	
11212	The number of base addresses is over the limit (num).	

No.	Error Message	Notes and Action
11400	Can't open temporary file.	
11401	Can't delete temporary file.	

11402	Can't open I/O data file(filename).	
11403	The I/O data not set.	
11404	The Output file of the same already set.	
11405	Data not found.	
11406	The start cycle larger than the end cycle.	
11407	The Output port already set.	
11408	There is no data in the Input file.	
11409	Illegal file format.	
11410	Can't open file.	
11411	Can't open (filename).	
11412	Address value is out of range.	

No.	Error Message	Notes and Action
11600	Can't execute this command.	
11601	Already set hard ware break.	
11602	Combination of bus width and access condition.	
11603	The start cycle larger than the end cycle.	
11604	HardwareBreak command cannot be used while state transition break window opens.	
11605	TracePoint command cannot be used while State Transition Trace Window, Time Measurement Window, MR Trace/Analyze Window or Task Trace/Analyze Window opens.	
11606	These trace data can't disassemble.	
11607	Cycle value is out of range.	

No.	Error Message	Notes and Action
11800	The I/O data not set.	
11801	Can't open (filename).	
11802	Can't open temporary file.	
11803	Address value is out of range.	
11804	Can't delete temporary file.	
11805	Can't open Log File (filename).	
11806	Address value is out of range.	

No.	Error Message	Notes and Action
12000	Address Interrupt Break is invalid.	
12001	Address value is out of range.	

No.	Error Message	Notes and Action
12200	Size value is out of range.	
12201	Base Address value is out of range.	
12202	The specified area is out of range.	
12203	failed to save data.	
12204	failed to load data.	

12205	he number of RAM monitor area cannot be changed.	

No.	Error Message	Notes and Action
12400	Sampling period value is out of range.	

No.	Error Message	Notes and Action
16000	INTERNAL ERROR: Already connected with the target.	Contact your nearest distributor.
16001	INTERNAL ERROR: Fork error has happen.	Contact your nearest distributor.
16002	Can't find Host Name (xxxxx).	
16003	INTERNAL ERROR: The Baud rate is illegal.	Contact your nearest distributor.
16004	The connection with the target isn't created.	
16005	Can't connect with the target.	
16006	INTERNAL ERROR: The Time of time out is out of range.	Contact your nearest distributor.
16007	Time Out ERROR.	Contact your nearest distributor.
16008	INTERNAL ERROR: Can’t disconnect with the target.	
16009	INTERNAL ERROR: Can’t send given size data.	Contact your nearest distributor.
16010	INTERNAL ERROR: Parameter is illegal.	Contact your nearest distributor.
16011	Illegal Host Name.	
16012	Communication ERROR. The connection with the target is closed.	
16013	Communication ERROR. Can't send data.	
16014	Communication ERROR. Can't accept data.	
16015	Target is already used.	
16016	Specified communications interface doesn't support.	
16017	LAN I/F can't be used on Windows3.1.	
16018	Parallel connection doesn't support on Windows NT.	
16019	Setting of the communications interface is illegal.	
16020	OverRun ERROR with serial communications.	

No.	Error Message	Notes and Action
16200	Address value is out of range.	
16201	That baud rate has not yet supported.	
16202	Bit number is out of range.	
16203	STOP execution.	
16204	Data value is out of range.	
16205	Monitor File (filename) is broken.	
16206	Can't find File (filename).	
16207	Target system is not constructed properly.	
16208	INTERNAL ERROR: ER_IN2_ILLEGAL_MODE	Contact your nearest distributor.

	has happen. (in xxxxx)	
16209	Mask value is out of range.	
16210	Counter of measurement time is overflow.	
16211	The version of string1 and the firmware on the target are not same.	
16212	Pass count value is out of range.	
16213	Can't execute that command, when the target program is running.	
16214	Target MCU is reset state. Please reset target systems.	
16215	Target MCU is unable to reset. Please reset target systems.	
16216	Target MCU is HOLD state. Please reset target systems.	
16217	Target MCU is not given clock. Please reset target systems.	
16218	Target MCU is not given power. Please reset target systems.	
16219	INTERNAL ERROR: Break point number is illegal.	Contact your nearest distributor.
16220	Please download the firmware to target.	
16221	Can't download firmware.	
16222	Can't find trace data which is able to refer.	
16223	Cycle value is out of range.	
16224	Target MCU is not under control. Please reset target systems.	
16225	First data is larger than second data.	
16226	First address is larger than second address.	
16227	No event set on the state transition path.	
16228	Time out value is out of range.	
16229	Process ID value is out of range.	
16230	Communication protocol error. (Argument error)	Contact your nearest distributor.
16231	There was sent undefined data from Emulator.	Contact your nearest distributor.
16232	Check sum error of the received data occurred.	Contact your nearest distributor.
16233	The specified data do not exist.	
16234	The target program is running.	
16235	The target program is not running.	
16236	The measurement has already been stopping.	
16237	The measurement has already been being executed.	
16238	The measurement is not completed.	
16239	There is no trace data of the specified cycle.	
16240	There is no trace data.	
16241	The measurement counter of time overflowed.	
16242	POF state was released by compulsory reset.	
16243	A number of setting points exceeds the range.	

16244	The program break is not set.	
16245	Source line information is not loaded.	
16246	The trigger mode is not a software output mode.	
16247	The exception processing was detected while executing the step.	
16248	Function range error.	
16249	The writing error to EEPROM occurred.	
16252	Unexecutable command code was specified.	
16253	The processor mode and the target system are the disagreements. xxxxx mode is used.	
16254	The specified bank isn't defined in the expansion memory.	
16255	The bank set up is duplicated.	
16256	The specified area includes the debugging monitor memory area.	
16257	The specified area includes the debugging monitor work area.	
16258	Flash ROM deletion error occurred. Flash ROM deletion error occurred.	
16259	Flash ROM verify error occurred.	
16260	Specification area includes the internal (flash) ROM area.	
16261	When Word is specified for a size, the odd number address cannot be specified.	
16262	Can not spesify the larger total bank size than the total emulation memory size.	
16263	The bank specified is defined as EXTERNAL.	
16264	The setting value is invalid in this processor mode.	
16265	RDY signal of MCU is Low.	
16266	HOLD signal of MCU is Low.	
16267	All program break points in the spesified bank is cleared.	
16268	Please specify the address in the emulation memory area.	
16269	The mistake is found in setting the emulation memory area.	
16270	The specified area has already been used in the debugging monitor bank address.	
16271	Too many emulation memory area specification.	
16272	The bank from 0 to 3 cannot be specified.	
16273	The mistake is found in the specification of the debugging monitor bank address.	
16274	The mistake is found in the specification of the debugging monitor work address.	
16275	Cannot specifiy to extend more than two banks.	
16276	Please specify the address in the emulation memory area.	

16277	Too many ROM area specification.	
16278	Start address is larger than end address.	
16279	Too many DMA area specification.	
16281	The mistake is found in the specification of the DMA area.	
16282	When Word is specified for a size, the odd number address cannot be specified.	
16283	Too many memory mapping specification.	
16284	The mistake is found in the specification of the memory mapping.	
16285	Please specify the address in the emulation memory area.	
16286	The mistake is found in setting the emulation memory area.	
16287	The specified area has already been used in the debugging monitor bank address.	
16288	Too many emulation memory area specification.	
16289	The bank from 0 to 3 cannot be specified.	
16290	The mistake is found in the specification of the debugging monitor bank address.	
16291	The mistake is found in the specification of the debugging monitor work address.	
16292	Cannot specifiy to extend more than two banks.	
16293	Please specify the address in the emulation memory area.	
16294	Too many ROM area specification.	
16295	Start address is larger than end address.	
16296	Too many DMA area specification.	
16298	The mistake is found in the specification of the DMA area.	
16299	Too many 8 bits bus mode area specification.	
16300	The mistake is found in the specification of the 8-bit bus mode area.	
16301	When Word is specified for a size, the odd number address cannot be specified.	
16302	The S/W breakpoint cannot be set in the SFR area and the RAM area.	
16303	The S/W breakpoint cannot be set in the flash ROM area.	
16304	The S/W breakpoint cannot be set.	
16305	The H/W breakpoint cannot be set in the SFR area and the RAM area.	
16306	The H/W breakpoint cannot be set in the flash ROM area.	
16307	The H/W breakpoint cannot be set.	
16308	Too many memory mapping specification.	
16309	The mistake is found in the specification of the memory mapping.	

16310	The target MCU is SLEEP mode.	
16311	The target MCU is STANDBY/STOP mode.	
16312	The target MCU is NO REFRESH STANDBY mode.	
16313	The MCU is HOLD state.	
16314	Work Address value is out of range.	
16315	The received data is illegal. The received data must be 'xxxxx'. But 'yyyyy' is received.	
16316	INIT code is received.	
16317	The sent command cannot be executed in this H/W environment.	
16318	The specified event is used in an another mode.	
16319	The chip break 0 is used in an another mode.	
16320	An uninitialized interrupt vector was detected.	
16321	This break function can't be set up in the ROM area or the memory area which doesn't exist.	
16322	This break function can't be set up in the odd number address.	
16323	This break function can't be set up in the middle of 32bit instruction.	
16324	A memory area which doesn't exist was manipulated. Or, A memory area was manipulated on the condition which wasn't forgiven.(address=H'xxxxx)	
16325	A specified reference section number is outside the range.	
16326	Tracing data file can't be open.	
16327	Tracing data can't be read from the file.	
16328	The specified break condition does not correspond to the trace output mode.	
16329	This break function can't be set up in the LSB side parallel instruction.	
16330	Can't execute from the LSB side parallel instruction.	
16347	Specification area includes not only the internal (flash) ROM area but also other area.	
16351	A request to an unavailable RAM monitor or coverage area has been maid.	
16352	Not in the output mode is the event output terminal.	
16353	Address Interrupt Break is invalid.	
16354	Remove Address Interrupt Break Point(s).	
16355	Remove Area (num) S/W Break Point(s). Remove Address Interrupt Break Point(s).	
16370	The S/W breakpoint cannot be set.	
16371	The break point isn't defined at that address.	
16372	The number of break point is over the limit (num).	

16373	Warning : The specified range was regulated into string1.	
16374	Software breakpoint is already set.	
16375	The number of base addresses is over the limit (num).	
16376	The block number is out of range.	
16377	The Generated mode cannot be specified. Please set the frequency of the Emulator Generation Clock in the Init dialog box at the next startup.	
16381	The target system may not work correctly, because the input level of string1 pin is 'L'. Check the pin level and the setting of the debugger.	
16382	The target system may not work correctly, because the input level of string1 pin is 'H'. Check the pin level and the setting of the debugger.	
16383	The target system may not work correctly, because the input level of string1 pin does not correspond to the setting. Check the pin level and the setting of the debugger.	
16384	Check the processor mode and the string1 pin level.	
16385	Self Check Error Please turn off the emulator. ([1]: string1.)	

No.	Error Message	Notes and Action
16400	INTERNAL ERROR:Already connected with the target.	Contact your nearest distributor.
16401	INTERNAL ERROR:Fork error has happen.	Contact your nearest distributor.
16402	Can't find Host Name (hostname).	
16403	INTERNAL ERROR:The Baud rate is illegal.	Contact your nearest distributor.
16404	The connection with the target isn't created.	
16405	Can't connect with the target.	
16406	INTERNAL ERROR:The Time of time out is out of range.	Contact your nearest distributor.
16407	Time Out ERROR.	
16408	INTERNAL ERROR:Can't disconnect with the target.	Contact your nearest distributor.
16409	INTERNAL ERROR:Can't send given size data.	Contact your nearest distributor.
16410	INTERNAL ERROR: Parameter is illegal.	Contact your nearest distributor.
16411	Illegal Host Name.	
16412	Communication ERROR. The connection with the target is closed.	
16413	Communication ERROR.Can't send data.	
16414	Communication ERROR.	

	Can't send data.	
16415	Target is already used.	
16416	Parallel connection doesn't support on Windows NT.	
16417	Can't find Simulator Engine.	

No.	Error Message	Notes and Action
16600	Address value is out of range. Address value is out of range.	
16601	That baud rate has not yet supported.	
16602	Bit number is out of range.	
16603	STOP execution.	
16604	Data value is out of range.	
16605	Monitor File (filename) is broken.	
16606	Can't find File (filename).	
16607	Target system is not constructed properly.	
16608	INTERNAL ERROR:ER_IN2_ILLEGAL_MODE has happen(in string1).	Contact your nearest distributor.
16609	Mask value is out of range.	
16610	Counter of measurement time is overflow.	
16611	The version of PD and the firmware on the target are not same.	
16612	Pass count value is out of range.	
16613	Can't execute that command, when the target program is running.	
16614	Target MCU is reset state. Please reset target systems.	
16615	Target MCU is unable to reset. Please reset target systems.	
16616	Target MCU is HOLD state.	
16617	Target MCU is not given clock. Please reset target system.	
16618	Target MCU is not given power.	
16619	INTERNAL ERROR:Break point number is illegal.	Contact your nearest distributor.
16620	Please download the firmware to target	
16621	Can't download firmware.	
16622	Download firmware is finished. Please restart PD.	
16623	Can't find trace data which is able to refer.	
16624	Cycle value is out of range.	
16625	Target MCU is not under control. Please reset target systems.	
16626	First data is larger than second data.	
16627	First address is larger than second address.	
16628	First address is larger than second address.	

16629	No event set on the state transition path.	
16630	Process ID value is out of range.	
16631	Communication protocol error.(Argument error)	Contact your nearest distributor.
16632	Check sum error of the received data occurred.	Contact your nearest distributor.
16633	The specified data do not exist.	
16634	The target program is running.	
16635	The target program is not running.	
16636	The measurement has already been stopping.	
16637	The measurement has already been being executed.	
16638	The measurement is not completed.	
16639	There is no trace data of the specified cycle.	
16640	There is no trace data.	
16641	The measurement counter of time overflowed.	
16642	POF state was released by compulsory reset.	
16643	A number of setting points exceeds the range.	
16644	The program break is not set.	
16645	Source line information is not loaded.	
16646	Source line information is not loaded.	
16647	The exception processing was detected while executing the step.	
16648	Function range error.	
16649	The writing error to EEPROM occurred.	
16650	There was sent undefined data from simulator.	Contact your nearest distributor.
16651	The received data is illegal. The received data must be (data). But (data) is received.	Contact your nearest distributor.
16652	INIT code is received.	Contact your nearest distributor.
16653	Can't read/write, because there are no memory at that area.	
16654	Number of points exceeds the limit (num).	
16655	Point already set.	
16656	Breakpoint of other type already set.	
16657	No hardware breakpoint set at specified address.	
16658	Can't get enough memory.	
16659	Can't set more I/O script file.	
16660	Can't set more virtual output.	
16661	Specified vector No. out of range.	
16662	Specified level of priority out of range.	
16663	Stack trace mode is not enabled.	
16664	The simulator engine execution error occurred.	
16665	Undefined instruction was executed.	
16666	Software break point can't be set up in the address.	

16667	Software break point can't be set up in the odd number address.	
16668	Software break point can't be set up in the middle of 32bit instruction.	
16669	Software break point can't be set up in the LSB side parallel instruction.	
16670	A memory territory which doesn't exist was manipulated. Or, A memory territory was manipulated on the condition which wasn't forgiven.	
16671	Can't execute from the LSB side parallel instruction.	

No.	Error Message	Notes and Action
16800	. Can't find '{'.(line: num)	
16801	Can't find '}'. (line: num)	
16802	Can't find '('.(line: num)	
16803	Symbol isn't defined. (line: num , token: string)	
16804	Can't find ')'.(line: num)	
16805	Description of expression is illegal. (line: num , token: string)	
16806	Nest level of the if statement is overflow. (line: num)	
16807	Nest level of the while statement is overflow. (line: num)	
16808	Too many the break statement. (line: num)	
16809	There is no if statement corresponding to the else statement. (line: num)	
16810	Unknown token. (line: num , token: string)	
16811	Can't open the (filename) file	
16812	The (filename) file is not a file made in the I/O window.	
16813	The description of the memory variable is illegal. (line: num)	

No.	Error Message	Notes and Action
17000	INTERNAL ERROR:External frash memory rewrite module parameter is wrong.	
17001	Can't find FTD file.	
17002	The FTD file is broken.	
17003	The number of External flash rom is over.	
17004	INTERNAL ERROR:The device number is illegal.	
17005	An Error was detected in work ram area activate commands.	
17006	An Error was detected in work ram area activate commands.	
17007	An Error was detected in external flash rom area	

	activate commands.	

No.	Error Message	Notes and Action
20000	Task with specified task No. not found.	
20001	Context of specified task No. not found.	
20002	Corrupted MR data.	
20003	Can't get enough memory.	

No.	Error Message	Notes and Action
20200	History of the system call issue that conforms to the search condition cannot be found.	

No.	Error Message	Notes and Action
20400	Can't use Task Pause function.	
20401	Task Pause function (xxxxx) was failed.	

No.	Error Message	Notes and Action
20600	Can't use Task Trace Window without setting real-time OS information.	

No.	Error Message	Notes and Action
20800	The save file name (filename) is wrong.	
20801	Can't find symbol (xxxxx) of MR.	
20802	Initialization routine of MR is not executed.	
20803	Can't find the task of the specified task number.	
20804	Priority out of range.	
20805	Task ID out of range.	
20806	Flag ID out of range.	
20807	Semaphore ID out of range.	
20808	Mailbox ID out of range.	
20809	Memory pool ID out of range.	
20810	Cyclic handler ID out of range.	
20811	Address out of range.	
20812	Cannot invoke system call.	
20813	System call not invoked.	
20814	System call not completed.	
20815	Address value is out of range.	
20816	File Name is illegal.	
20817	Corrupted MR data.	
20818	Can't get enough memory.	

No.	Error Message	Notes and Action
26000	Address value is out of range.	
26001	Description of Assembly language is illegal.	

26002	Address value for JUMP is out of range.	
26003	Operand value is out of range.	
26004	Description of expression is illegal.	
26005	Addressing mode specified is not appropriate.	
26006	INTERNAL ERROR: 'ALIGN' is multiple specified in '.SECTION'.	Contact your nearest distributor.
26007	Operand value is undefined.	
26008	Bit-symbol is in expression.	
26009	Invalid bit-symbol exist.	
26010	Symbol value is not constant.	
26011	Same items are multiple specified.	
26012	Same kind items are multiple specified.	
26013	Characters exist in expression.	
26014	Format specified is not appropriate.	
26015	Invalid symbol definition.	
26016	Invalid reserved word exist in operand.	
26017	INTERNAL ERROR: 'JMP.S' operand label is not in the same section.	Contact your nearest distributor.
26018	Reserved word is missing.	
26019	No space after mnemonic or directive.	
26020	INTERNAL ERROR: No '.FB' statement.	Contact your nearest distributor.
26021	INTERNAL ERROR: No '.SB' statement.	Contact your nearest distributor.
26022	INTERNAL ERROR: No '.SECTION' statement.	Contact your nearest distributor.
26023	Operand value is not defined.	
26024	Operand size is not appropriate.	
26025	Operand type is not appropriate.	
26026	INTERNAL ERROR:Section attribute is not defined.	Contact your nearest distributor.
26027	INTERNAL ERROR: Section has already determined as attribute.	Contact your nearest distributor.
26028	INTERNAL ERROR: Section name is missing.	Contact your nearest distributor.
26029	INTERNAL ERROR: Section type is not appropriate.	Contact your nearest distributor.
26030	INTERNAL ERROR: Section type is multiple specified.	Contact your nearest distributor.
26031	Size or format specified is not appropriate.	
26032	Size specified is missing.	
26033	String value exist in expression.	
26034	Symbol is missing.	
26035	Symbol is multiple defined.	
26036	Symbol is missing.	
26037	Symbol is multiple defined.	
26038	Invalid operand exist in instruction.	
26039	Syntax error in expression	
26040	Invalid operand exist in instruction.	

26041	Operand expression is not completed.	
26042	Too many operand.	
26043	Too many operand data.	
26044	Undefined symbol exist.	
26045	Value is out of range.	
26046	Division by zero.	
26047	INTERNAL ERROR:'.VER' is duplicated.	Contact your nearest distributor
26048	'#' is missing.	
26049	',' is missing.	
26050	']' is missing.	
26051	')' is missing.	
26052	INTERNAL ERROR: Symbol defined by external reference data is defined as global symbol.	Contact your nearest distributor.
26053	Invalid operand exist in instruction.	
26054	Quote is missing.	
26055	Right quote is missing.	
26056	Can't get enough memory.	
26057	Invalid chip mode.	
26058	':' is missing.	
26059	Absolute addressing is not avail.	
26060	Direct addressing is not avail.	
26061	Invalid addressing mode declaration included.	
26062	Syntax error in indexed addressing expression.	
26063	'(' is missing.	
26064	Internal error.	
26065	Operand value of direct addressing is out of range.	
26066	Operand value of absolute addressing is out of range.	
26067	Operand value of absolute long addressing is out of range.	
26068	Operand value of stack relative addressing is out of range.	
26069	Operand value is illegal.	
26070	The indirect addressing you expressed is illegal.	
26071	An odd number address can't be specified.	

No.	Error Message	Notes and Action
26200	Line number is illegal.	
26201	Can't find right bracket ')'.	
26202	The Number of Macro constant is over the limit (num).	
26203	Immediate value is out of range.	
26204	Prefix which gives radix of the constant is illegal.	
26205	Description of indirect reference is illegal.	

26206	Can't find end of strings (xxxxx).	
26207	Description of expression is illegal.	
26208	Macro constant (macro) isn't defined.	
26209	Symbol (symbol) isn't defined.	
26210	Immediate value is illegal.	
26211	Divide by 0.	
26212	The value is over the maximum value of which can be treated by MCU.	
26213	Register name is using for macro variable name.	

No.	Error Message	Notes and Action
26400	Address value is out of range.	
26401	Bit number is out of range.	
26402	File (filename) is broken.	
26403	Can't find File (filename).	
26404	Can't find sub routine information.	
26405	Illegal character in the strings.	
26406	INTERNAL ERROR: ER_IN2_ILLEGAL_MODE has happen. (in xxxxx)	Contact your nearest distributor
26407	Can't find that line number.	
26408	Multiple definition of symbol/label.	
26409	There are no code at that line.	
26410	Can't get enough memory.	
26411	Can't find scopes.	
26412	Can't find section information.	
26413	Can't find source lines which correspond to that address.	
26414	Can't find symbol (symbol).	
26415	Can't find the scopes which include that address.	
26416	Loading is canceled.	
26417	INTERNAL ERROR: The end of section information.	Contact your nearest distributor.
26418	INTERNAL ERROR: The end of section information.	Contact your nearest distributor.
26419	The register name is wrong.	
26420	Can't find Source File (filename).	
26421	Unable to read Load Module File (filename).	
26422	The PATH name is incorrect.	
26423	Cannot open the save file (filename).	
26424	Can't open SYSROF file.	
26425	Can't read SYSROF file.	
26426	Illegal file format. (no absolute format file)	
26427	Illegal file format.	
26428	Can't get enough memory.	

26429	Can't find file.	
26430	There are no address at that line.	
26431	Can't find the function which correspond to that source line.	
26432	Can't find the scopes which include that address.	
26433	Can't find symbol.	
26434	Can't find the function which correspond to that source line.	
26435	Loading is canceled.	
26436	INTERNAL ERROR: ER_LOAD_SYMSCOPE has happen.	Contact your nearest distributor.
26437	File Name is illegal.	
26438	Display source codes.	
26439	The path name is too long.	

No.	Error Message	Notes and Action
26600	Can't open file (filename).	
26601	Can't create file (filename).	
26602	Can't close file (filename).	
26603	File seek error (in xxxxx).	
26604	Out of disk space.	
26605	Illegal file format (xxxxx --> xxxxx). (filename)	
26606	Out of heap space.	
26607	Not yet implemented (xxxxx).	

No.	Error Message	Notes and Action
30200	Comfirm the processor mode and the CNVss terminal level.	
30201	Comfirm the emulation memory allocation, or the mapping.	

No.	Error Message	Notes and Action
30400	MCU file is old format.	
30401	MCU file is illegal format.	

No.	Error Message	Notes and Action
30600	In connected emulation-pod, the target clock is external fixation.	

No.	Error Message	Notes and Action
30201	Comfirm the emulation memory allocation, or the mapping.	

No.	Error Message	Notes and Action
38000	The value of Bank is wrong.	

[MEMO]

Emulator Debugger for PC7501 System User's Manual

Rev. 1.00
May 1, 2003
REJ10J0023-0100Z

M3T-PD308F V.3.00
M3T-PD30F V.2.00
User's Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ10J0023-0100Z