PARTNER Users Manual

“V800 Series Common Edition”

Copyright (C) 2000 Midas Lab. Inc./Kyoto Micro Computer Co., Ltd.

* About User Registration (Important Notice)

- Please fill out user registration card attached to the product, and send it back immediately.
- User support (such as technical inquiry or notification about product update from vendor) is based on this user registration card.

* PARTNER version up

- The latest PARTNER can be down-loaded from following web page.
  ftp://caravan.net/midas/program/partner/update/

* Notices

- The copyright associated with PARTNER (including software and documentation) are proprietary to Midas Lab. Inc. and Kyoto Micro Computer Co., Ltd.
- This software and manual are protected under applicable copyright laws, and may not be copied, in whole or in part, in any way without explicit prior written permission from Midas Lab. Inc. and Kyoto Micro Computer Co., Ltd.
- The right of use granted for the customer means the right to use PARTNER only on one system that is registered.
- While this product was manufactured with all possible care, Midas Lab. Inc. and Kyoto Micro Computer Co., Ltd. assume no responsibility whatsoever for any result of using this product.
- The contents and specifications of this product and this document are subject to change without notice.
- By change of the product, the display image described in this document might vary from the actual one of the product.

* Trademarks

- MS-Windows, Windows, MS and MS-DOS are the trademarks of Microsoft Corporation, U.S.A.
- The names of the programs, systems, CPUs, and other products that appear in this document are usually trademarks of the manufacturer of the corresponding product.

Revision History

Date Y / M / D	Rev	Contents
1998.07.15	1.00	Initial Revision
1999.05.14	2.00	The 2nd Revision Edit to merge V800 series manuals
2000.03.06	2.10	Add "3.6 Notices on Operations"
2000.05.10	2.20	English version remake

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

Getting Started

This document explains the common part of V800 series, such as how to install, startup and operate PARTNER.

Regarding the operation depend on CPU and how to use the product, please refer to a PARTNER document for individual CPU.

Regarding detail for dialog command, please refer to an on-line help. Depend on CPU and how to use the product, display image and items describe in document might vary from the actual one.

PARTNER Overview

PARTNER is a Window based source level debugger, which is developed as PARTNER-Win by Kyoto Micro Computer Co., Ltd., and ported for the products of Midas lab Inc.

In addition to the basic functions as a source level debugger tool, such as program load, program execution, break point control, data display/change, code display/change, there are other functions customized for Midas lab products.

Package Contents

PARTNER product package contains setup CD-ROM, documents and user registration card in it.

Operation Environment

Followings are the explanations for necessary thing to use PARTNER in addition to the product package before starting setup.

Confirm the hardware and software environment of the computer equipment to install the product.

* Necessary Hardware

- Host computer

The one has a CPU higher than P5-90, and Windows95/98/NT workable.

- Hard drive

Hard drive is required to use PARTNER. To install PARTNER, more than 10MB of free area is required.

- CD-ROM drive

CD-ROM drive is required to install PARTNER

- Others

Other prerequisites might vary depend on how to use. For detail, refer to individual CPU dependent part.

* Necessary Software

- Windows

Windows95/98/NT (Operating System) is required. Before start to install PARTNER, install to the Hard drive so that Windows to be workable.

- Software development tools

Programming language tools are required to build the target software that PARTNER supports.

- Others

Depend on the case and situation, refer to the document "Part Edition" for detail.

About This Document

"PARTNER Users Manual" consists of two parts, "Common Edition" which does not depend on target CPU or environment of use and "Part Edition" which depends on them. The convention used in these documents is as follows.

[Menu] Menu name is denoted by brackets ([ ]) quoted character string.

[Menu]-[Command] Command name is denoted by brackets ([ ]) quoted character string preceded by menu name that the command belongs to.

[Dialog] Dialog box name is denoted by brackets ([ ]) quotation.

<Button> Various button names are denoted by angle brackets (< >) quotation.

Regarding On-line Help

PARTNER provides On-line Help that explains functions and operations on computer display.

The On-line Help is displayed by HELP Key (End key at PC/AT), HELP command, [Help]-[Help Contents] command or <Help> button.

This page is intentionally left blank.

1 Your First PARTNER

To start to use PARTNER, this chapter explains features of PARTNER and debugging procedures briefly.

1.1 PARTNER feature

PARTNER employs MDI (Multi Document Interface), which displays various necessary information in each child window during debugging. It also introduces Short-cut key, Menu, Tool Bar, Status bar and Dialog box, and various functions are provide to improve operation environment.

1.1.1 Data Reference/Change and Link Function

Just moving a mouse cursor to the variable displayed in Code window and double clicking on it, the Inspect window opens to refer and change the data easily. The data reference/change can be done also in Watch window and Local window in similar manner. Moreover, the data can be changed by mouse operation also in Register window and Memory window.

1.1.2 Tool Bar

Frequently used commands, such as loading debug program, setting break point, program execution and inspecting variables, can be registered to Tool Bar and can be invoked easily just by clicking on the Tool Bar buttons.

1.1.3 User Customization

User can customize Tool Bar, displayed font, display color and window layout as they want. Up to three types of window layout can be registered depend on operating environment.

1.2 Basic Operation Procedures in PARTNER

Basic operation procedures in PARTNER are explained as follows.

(1) Setup

Setup (install) PARTNER software to your hard drive.

cf.: page 11, "2 Setup"

(2) Configuration

Setup necessary configuration to invoke PARTNER.

cf.: page16, "3.1 PARTNER configuration"

The configuration is done by PARTNER configuration program (RPTSETUP). (Fig. 1)

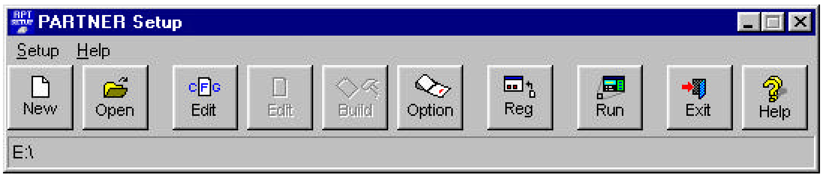
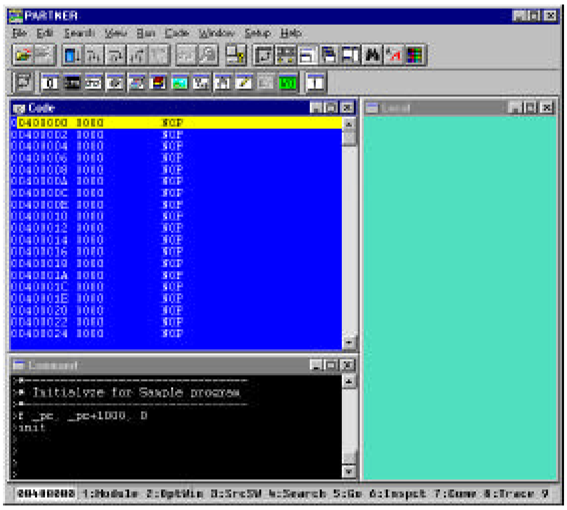


Fig. 1 Configuration Program (RPTSETUP)

(3) Invoking PARTNER

After finishing the configuration, invoke PARTNER software. (Fig. 2)



```
00400000 0000          NOP
00400002 0000          NOP
00400004 0000          NOP
00400006 0000          NOP
00400008 0000          NOP
0040000A 0000          NOP
0040000C 0000          NOP
0040000E 0000          NOP
00400010 0000          NOP
00400012 0000          NOP
00400014 0000          NOP
00400016 0000          NOP
00400018 0000          NOP
0040001A 0000          NOP
0040001C 0000          NOP
0040001E 0000          NOP
00400020 0000          NOP
00400022 0000          NOP
00400024 0000          NOP
```

```
>*------------------------------------
>* Initialyze for Sample program
>*------------------------------------
>f _pc, _pc+1000, 0
>init
>
>
>
>
```

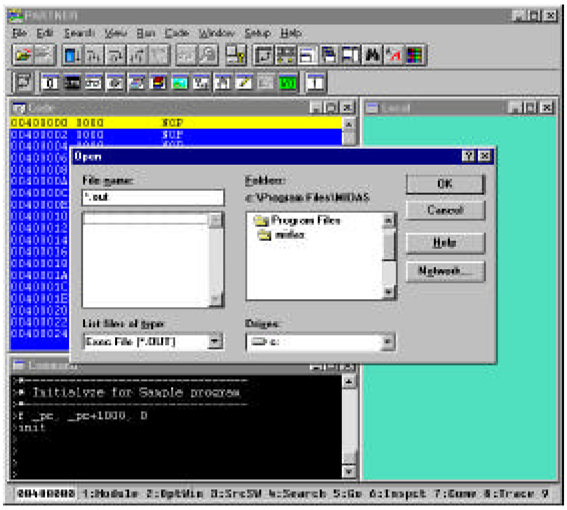
```
00400000 1:Module 2:OptWin 3:SrcSW 4:Search 5:Go 6:Inspct 7:Come 8:Trace 9
```

Fig. 2 Startup window

(4) Loading Program

Load the program to be debugged. (Fig. 3)

cf.: button (page 48), L command, [File]-[Load] command (page 35)



PARTNER
File Edit Search View Run Code Window Setup Help
Code
Local
Open
OK
Cancel
Help
Network...
Program Files
midas
Command
00400000 1:Module 2:OptWin 3:SrcSW 4:Search 5:Go 6:Inspect 7:Come 8:Trace 9

```
00400000 0000        NOP
00400002 0000        NOP
00400004
00400006
00400008
0040000A
0040000C
0040000E
00400010
00400012
00400014
00400016
00400018
0040001A
0040001C
0040001E
00400020
00400022
00400024
```

File name: *.out

Folders: c:\Program Files\MIDAS

List files of type: Exec File (*.OUT)

Drives: c:

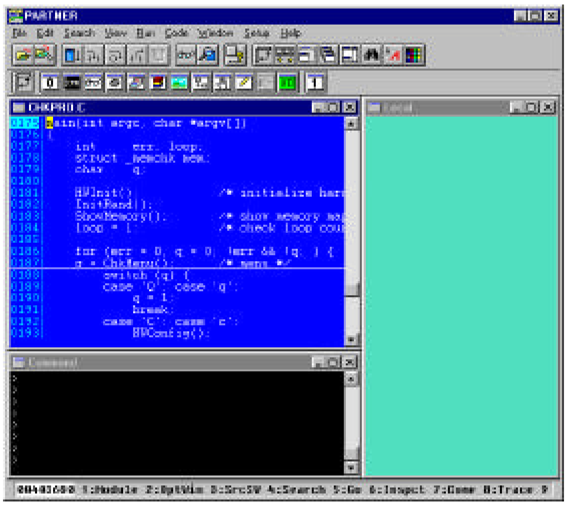
```
>*-------------------------------------
>* Initialyze for Sample program
>*-------------------------------------
>f _pc, _pc+1000, 0
>init
>
>
>
>
```

Fig. 3 Loading Program

(5) Set break point

Set a break point by clicking the mouse button on line number or address display in Code Window. (Fig. 4)

cf.: BP command "5.1.3 Mouse Operation in Code window ", [Break]-[Add] (Window command in "Part Edition")



PARTNER
File Edit Search View Run Code Window Setup Help
CHKPRO.C
Local
Command
00403680 1:Module 2:OptWin 3:SrcSW 4:Search 5:Go 6:Inspect 7:Come 8:Trace 9

```
main(int argc, char *argv[])
{
        int     err, loop;
        struct _memchk mem;
        char    q;

        HWInit();               /* initialize har
        InitRand();
        ShowMemory();           /* show memory ma
        loop = 1;               /* check loop cou

        for (err = 0, q = 0; !err && !q; ) {
        q = ChkMenu();          /* menu */
            switch (q) {
            case 'Q': case 'q':
                q = 1;
                break;
            case 'C': case 'c':
                HWConfig();
```

Fig. 4 Set Breakpoint

(6) Execute Program

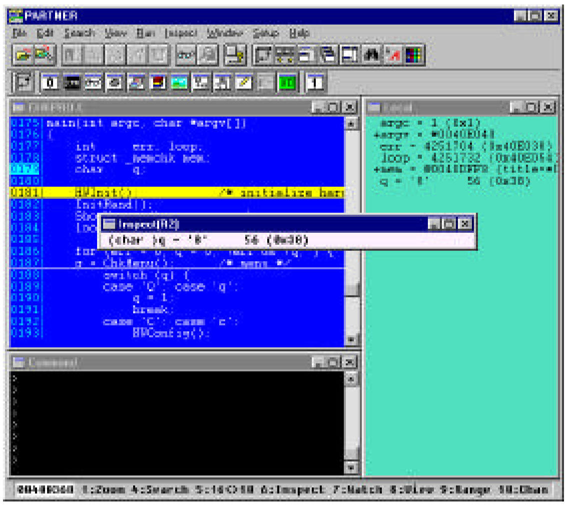
Starting program execution, that stops by the break points set in step (5), or forcibly stop by ESC key.

cf.: button (page 48), G command, [Run]-[Go] (page 37), button (page 48), ESC command

(7) Inspect Variables

Double clicking mouse button on a variable to refer (change) to display in Inspect window. (Fig. 5)

cf.: button (page 48, "5.1.1 Code window Short-cut Key" (page 55), INS command)


PARTNER
File Edit Search View Run Inspect Window Setup Help
Command
00400360 1:Zoom 4:Search 5:16<>8 6:Inspect 7:Watch 8:View 9:Range 10:Chan

```
main(int argc, char *argv[])
{
        int     err, loop;
        struct _memchk mem;
        char    q;

        HWInit()                /* initialize har
        InitRand();
        Sho
        Loc

        for
        q = ChkMenu();          /* menu */
            switch (q) {
            case 'Q': case 'q':
                q = 1;
                break;
            case 'C': case 'c':
                HWConfig();
```

```
 argc = 1 (0x1)
+argv = @0040E048
 err = 4251704 (0x40E038)
 loop = 4251732 (0x40E054)
+mem = @0018FFF8 (title=
 q = '8'      56 (0x38)
```

Inspect(R2)

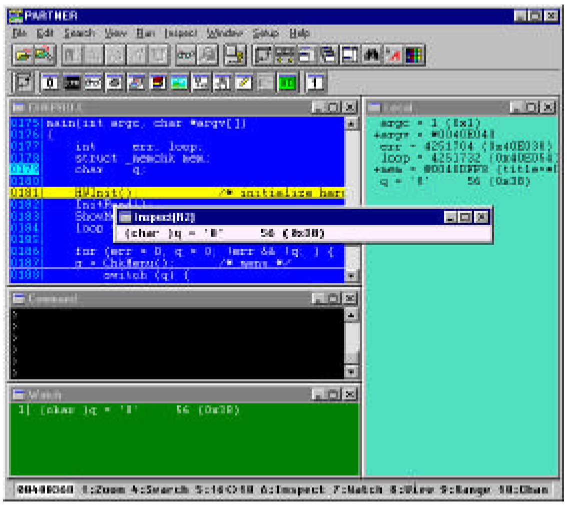
```
(char )q = '8'      56 (0x38)
```

Fig. 5 Inspect display

(8) Register Watch variables

Register the variables that always to be displayed to Watch window. (Fig. 6)

cf.: button (page 48), "5.1.1 Code window Short-cut Key" (page 55), W command



```
main(int argc, char *argv[])
{
    int     err, loop;
    struct _memchk mem;
    char    q;

    HWInit();              /* initialize har
    InitP
    ShowM
    Loop

    for (err = 0, q = 0; !err && !q; ) {
    q = ChkMenu();         /* menu */
        switch (q) {
```

```
 argc = 1 (0x1)
+argv = *0040E048
 err = 4251704 (0x40E038)
 loop = 4251732 (0x40E054)
+mem = @0018DFF8 {title=*
 q = '8'      56 (0x38)
```

```
(char )q = '8'      56 (0x38)
```

```
1| (char )q = '8'      56 (0x38)
```

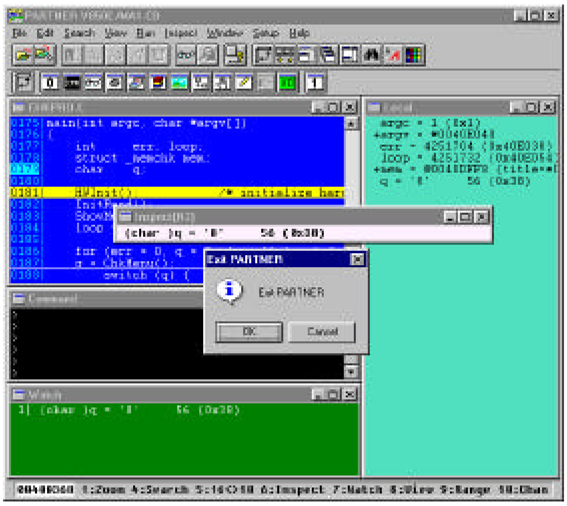
PARTNER
File Edit Search View Run Inspect Window Setup Help
Inspect
Command
Watch
1:Zoom 4:Search 5:16<>10 6:Inspect 7:Watch 8:View 9:Range 10:Chan

Fig. 6 Register Watch

(9) Terminate PARTNER

Terminate PARTNER software. (Fig. 7)

cf.: button (page 48), [File]-[Exit] (page 35), Q/EXIT command)



Exit PARTNER

Fig. 7 Confirmation of termination

2 Setup

To start to use PARTNER, setup to your host computer is required. This chapter explains the basic install procedures of PARTNER software.

2.1 Before Setup

Before starting setup PARTNER, finish the hardware and software installation described in each documentation.

2.2 How to setup

Start up Windows95/98/NT and setup the PARTNER software.

[Notice] Each file of PARTNER software is stored in distribution CD-ROM in compressed format. Thus, it is not enough to simply copy each file to hard drive for PARTNER installation. Be sure to setup properly using PARTNER setup program attached in the CD-ROM.

2.2.1 Read README.TXT

Before starting PARTNER setup program, read a README.TXT file that describes the notice to use and other information.

2.2.2 Install PARTNER

To install PARTNER, invoke a setup program "SETUP.EXE" contained in setup CD-ROM. For install procedures, follow the direction of setup program.

When the install has finished, "PARTNER folder" and "RPTSETUP" short-cut will be created in [Start]-[Program].

[Notice] After PARTNER setup finished, keep the setup CD-ROM to a safe place.

2.2.3 Un-install PARTNER

From "Add and Delete Applications" icon in Control panel, select the "PARTNER" from lists to delete. Please pay attention that if the status of the hard drive is changed after installation, the un-install might finish incompletely.

This page is intentionally left blank.

3 Startup

To startup PARTNER, the PARTNER configuration according to your operation environment is required.

Double clicking on "RPTSETUP" in Start menu to configure PARTNER environment.

3.1 PARTNER configuration

To startup PARTNER, following setups are required as a project (unit of debug).

- Create new project (page 17)
- Setup configuration file (page 17)
- Set start option (page 20)
- Register to Start menu (page 19)

With above setting properly, PARTNER can be invoked. "RPTSETUP" is a tool to configuration

3.1.1 How to use RPTSETUP

"RPTSETUP" is a tool to setup project configuration. Selecting <New> or <Open>, the button to setup configuration comes to be available.

Invoking "RPTSETUP" in Startup menu to create and setup (change) project. (Fig. 8)

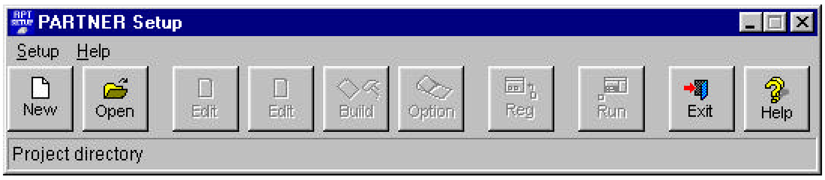


Fig. 8 RPTSETUP startup display

<Edit> and <Create> in menu cannot be selected by RPTSETUP. These functions are reserved for future expansion.

Create New Project

To create new project, select <New> button.

Create New project

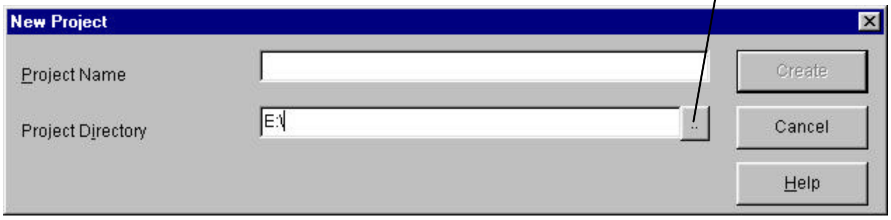
Create PARTNER project. It is usually created in "Projects" directory.

For newly created project, default project file (RPTV8xx.KPJ) and configuration file (RPTV8xx.CFG) will be created (copied from Bin directory). After then, these files will be used to customize for target system.

The directory where project is created, can be specified to working directory where user source program is located other than "project" directory.

In this case, configuration file (RPTV8xx.CFG) and project file (RPTV8xx.KPJ) will be created in user working directory. (Fig. 9)

Project Directory button



Project Name:

Project Directory: E:\

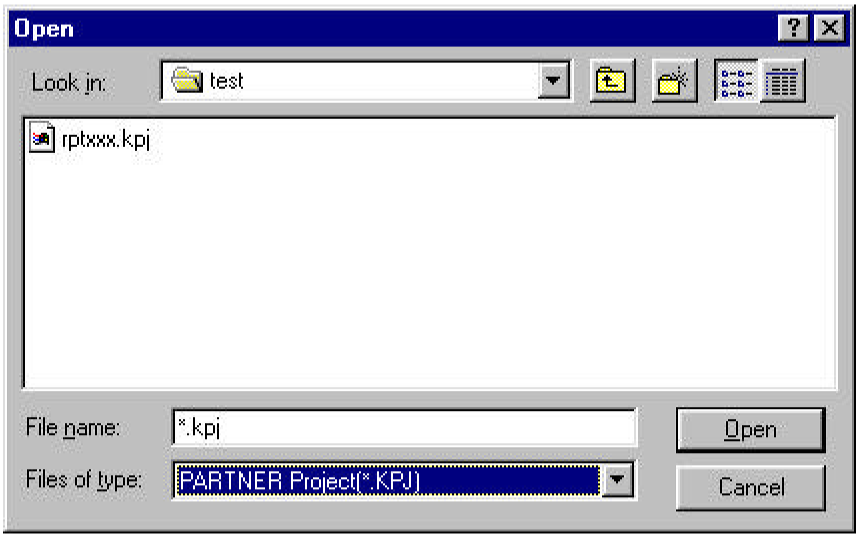
Fig. 9 Specify e:¥directory

Change Existing Project

To change the configuration of project, select the <Open> button.

Open Project

Select and open the project file (RPTV8xx.KPJ) to be changed.



Look in: test

File name: *.kpj

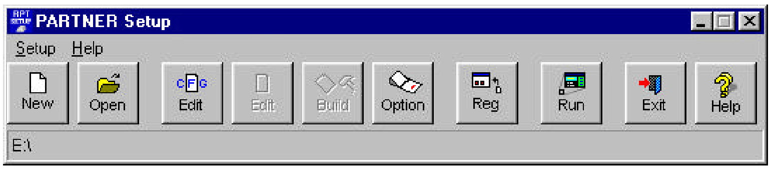
Files of type: PARTNER Project(*.KPJ)

Fig. 10 Select Project window

Set (change) project configuration

The project configuration can be set (changed) at newly created time or for existing project. (Fig. 11)

- Set start option
- Set configuration file
- Register to Start menu

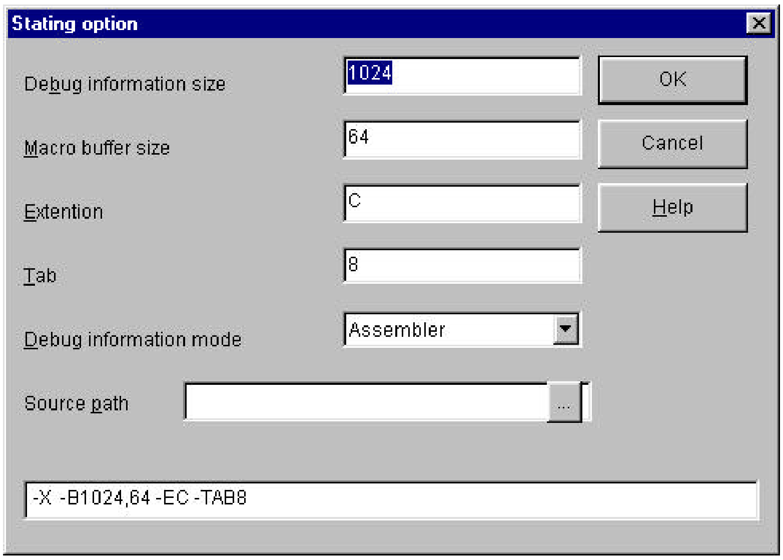


E:\

Fig. 11Set (change) Project display

Set Start option

Specifies PARTNER start option. The start options sets (changes) PARTNER operation configuration. (Fig. 12)



Debug information size: 1024

Macro buffer size: 64

Extention: C

Tab: 8

Debug information mode: Assembler

Source path:

-X -B1024,64 -EC -TAB8

Fig. 12 [Starting option] dialog box

The dialog box might vary depend on the CPU and operation environment.

For detail, refer to "Part Edition".

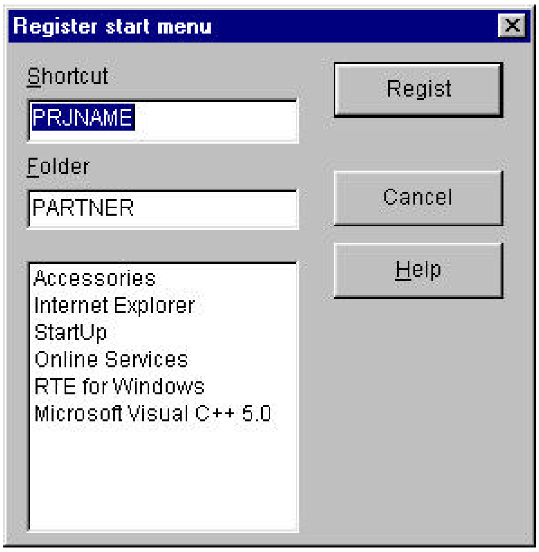
Edit configuration

Edit the configuration file located in working directory, according to the target CPU to be used.

For detail about the items of configuration, refer to page 23, "3.3 Configuration file(RPTV8xx.CFG)".

Register to Start menu

Register the project that has been configured to Start menu (Fig. 13). Invoking via registered short-cut, PARTNER will start with the contents of configured project.



Shortcut: PRJNAME

Folder: PARTNER

Fig. 13 Register to the Start menu with Short-cut key named "test".

3.2 Related Files

The files that PARTNER uses are as follows.

3.2.1 Necessary Files at Startup

For files required at PARTNER start up, there are configuration file, project file, automatic execution file and the one created at PARTNER termination. These files are read from the project directory.

Automatic execution file and the files created at PARTNER termination (refer to page 22,"3.2.2 Files created at PARTNER termination") will be read, if they exist at startup time

Configuration file(RPTV8xx.CFG)

The configuration file notifies the target environment to PARTNER. This file contains the configuration such as target memory map and initial value of registers related to the memory map.

Project file (RPTV8xx.KPJ)

The project file setups PARTNER configuration.

This file contains the configuration such as window arrangement, fonts, color setup, Tool Bar configuration and startup options.

Automatic execution file (INIT.MCR)

The automatic execution file will automatically be read and executed at PARTNER startup. This capability is similar to AUTOEXEC.BAT file in MS-DOS. If post processing required for program debugging (such as user program loading) is described in this file, they are automatically executed at PARTNER startup time.

3.2.2 Files created at PARTNER termination

PARTNER creates some files at its termination. These files will be saved in the project directory.

Contents of Memo window

Saves the contents of Memo window to "memo.pt" file.

Command history

Saves the contents of Command history to "xxxx.dat" file.

3.3 Configuration file(RPTV8xx.CFG)

PARTNER reads configuration file (RPTV8xx.CFG) at its startup. There needs to the specify target operation environment in the configuration file in advance.

The configuration file is a text file and the line start with semi-colon (;) is considered as comment.

[Notice] At the modification of configuration file, be careful not to change the order of description or the internal format.

3.3.1 Configuration Items

The items described in the configuration file are as follows.

- Initial value of program counter (PC)
- Initial value of stack pointer (SP)
- Memory region accessible by command

Initial value of program counter (INIT_PC)

Initializes the program counter (PC) to prevent invalid access in Code window at PARTNER startup.

Format: INIT_PC address

If omitted, default value of target CPU is assumed.

Initial value of stack pointer (INIT_SP)

Initializes the stack pointer (SP) to prevent invalid access in Stack window at PARTNER startup.

Format: INIT_SP address

If omitted, default value of target CPU is assumed.

Notes: This is not available for TP or IE series.

Memory region accessible by command (MAP)

Specifies the memory region accessible by PARTNER command and internal processing.

Format: MAP start_address,end_address

Declares that specified address area can be accessible by PARTNER command. If omitted, the area from 0000-0000 to FFFF-FFFF is assumed to be accessible.

Up to 20 MAP area can be specified. If plural areas need to be specified, describe these fields in ascending order and plural lines.

3.4 Initialize Command

If PARTNER operation environment needs to be set in advance, it can be initialized by commands.

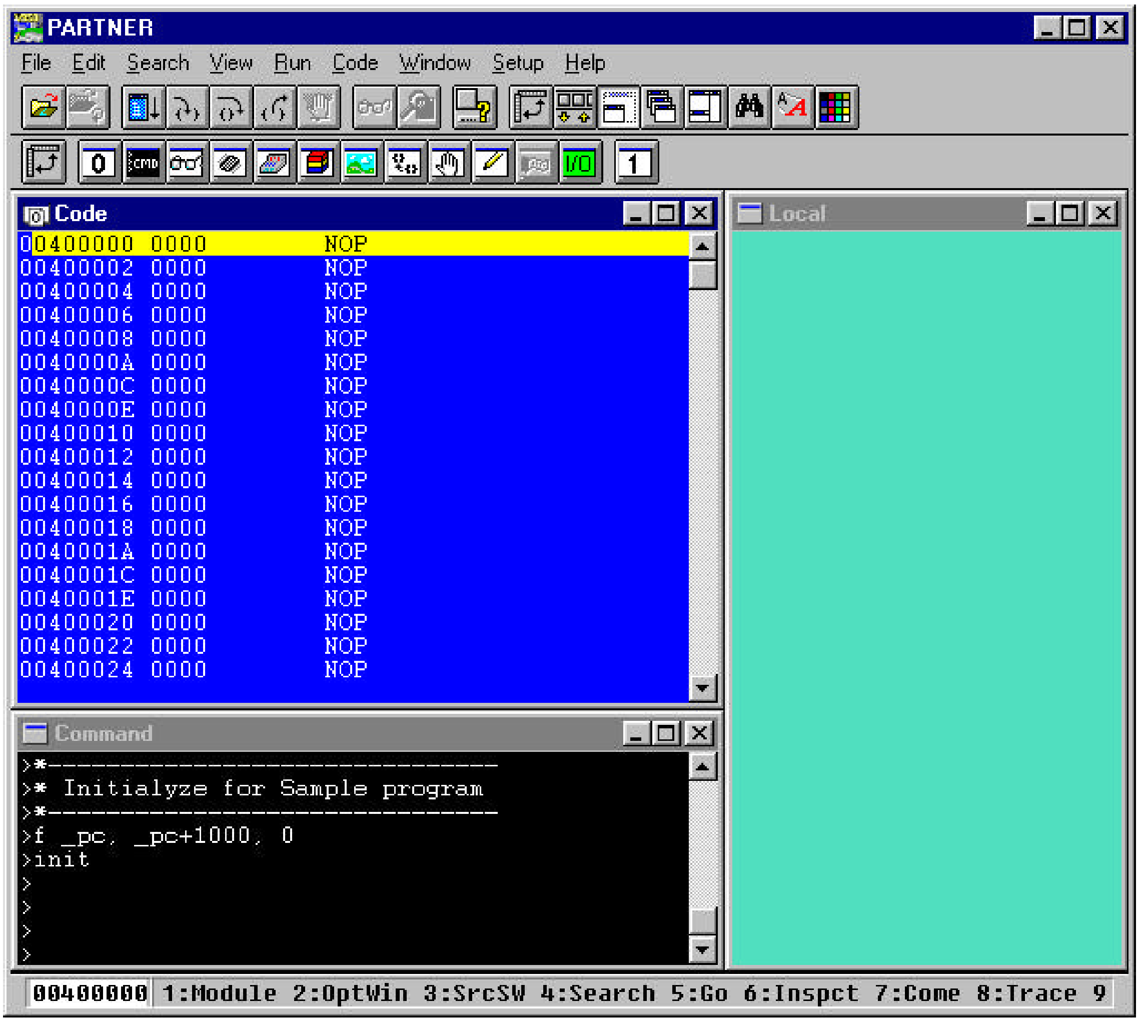
For detail, please refer to the Command Reference in "Part Edition" or Hardware User's Manual.

3.5 Invoking PARTNER

To invoke PARTNER, double click the icon registered to Program Manager by RPTSETUP.

3.5.1 PARTNER startup normally

When PARTNER is invoked, the display shown in Fig. 14 will appear.



```
00400000 0000         NOP
00400002 0000         NOP
00400004 0000         NOP
00400006 0000         NOP
00400008 0000         NOP
0040000A 0000         NOP
0040000C 0000         NOP
0040000E 0000         NOP
00400010 0000         NOP
00400012 0000         NOP
00400014 0000         NOP
00400016 0000         NOP
00400018 0000         NOP
0040001A 0000         NOP
0040001C 0000         NOP
0040001E 0000         NOP
00400020 0000         NOP
00400022 0000         NOP
00400024 0000         NOP
```

```
>*------------------------------
>* Initialyze for Sample program
>*------------------------------
>f _pc, _pc+1000, 0
>init
>
>
>
>
```

```
00400000 1:Module 2:OptWin 3:SrcSW 4:Search 5:Go 6:Inspct 7:Come 8:Trace 9
```

Fig. 14 PARTNER startup display

3.5.2 PARTNER startup trouble

If error occurs, a Message box will be displayed. For detail, please refer to the error messages at startup in "Part Edition".

3.6Notices on Operation

This section explains the notes to operate PARTNER smoothly.

3.6.1 Notices on Window Operation

Regarding window operation, following operation might cause troubles such as abnormal communication, monitor time-out or remarkable degrade in operation speed.

* PARTNER startup with displaying the window that refers memory

* Hardware reset without closing the window that refers memory

These troubles depend on the target system that requires initialization, such as bus controller and memory controller. The reason of trouble is a occurrence of memory access based on address value or register value before finishing the initialization. Thus, please consider that the target system that requires initialization for memory access, might cause this kind of trouble.

Followings are the windows that refer the memory contents.

button	window name	base value	description
0	Code window	address value	accesses memory contents based on displayed address value to get instruction code
1	Code window1	address value	accesses memory contents based on displayed address value to get instruction code
	Watch window	address value	accesses memory contents based on displayed symbol value to get data
	Memory window	address value	accesses memory contents based on displayed address value to get data
	Register window	register value	accesses memory contents based on each register value to get data (if memory display is selected)
	Stack window	register value	accesses memory contents based on stack pointer value to get data
	Local window	address value	accesses memory contents based on registered symbol value to get data
	History window	address value	accesses memory contents based on registered address as trace result to get data

Notes: Regarding the functions of above windows, please refer to page 53, "5 Child Window".

To suppress this kind of trouble, close the windows that refers the memory contents before terminating PARTNER or Hardware reset.

3.6.2 Notices on Command Execution

When the dialog commands that refers memory contents are invoked from Command window, the trouble similar to the one at window operation might occur for directly or indirectly.

* Direct Reason

The operation by memory display/change command (D, E, F, S,...)

* Indirect Reason

The operation specifies an expression that refer memory contents as parameter to specify the address or data of commands

Notes:

Regarding the <expression> specified as command parameter, please refer to page 91 "7 Data Expression", and regarding the commands available in Command window, refer to "Command Reference" in "Part Edition", respectively.

Thus, please consider that the target system that requires initialization for memory access, might cause this kind of trouble.

To suppress this kind of trouble, be sure to execute the commands that refer the memory contents after finishing the initialization.

This page is intentionally left blank.

4 Window command

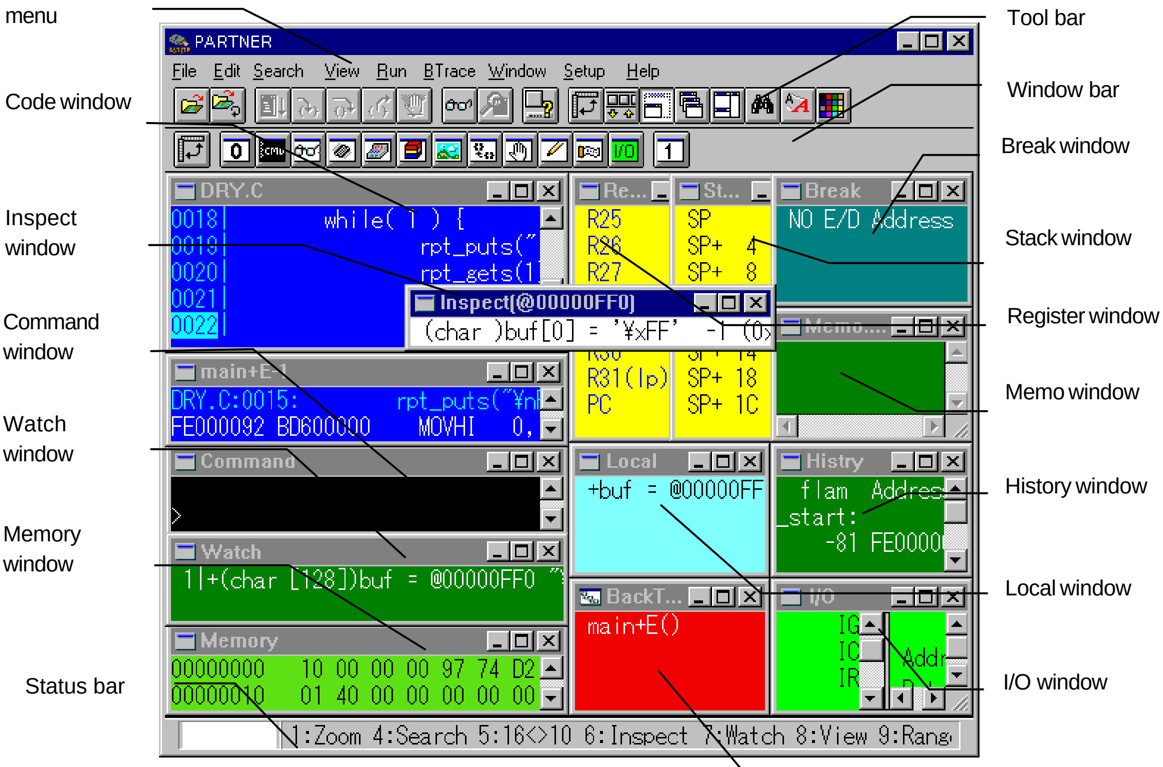
This chapter explains the basic operations that is characteristic of PARTNER debug capacities such as menu, Tool Bar, Status bar and child window.

4.1 Display Structure

This section explains the main display structures of PARTNER.

4.1.1 Components

The window components are shown in Fig. 15



```
0018|          while( 1 ) {
0019|                  rpt_puts("
0020|                  rpt_gets(1
0021|
0022|
```

```
(char )buf[0] = '¥xFF'   -1 (0x
```

```
DRY.C:0015:        rpt_puts("¥n
FE000092 BD600000     MOVHI    0,
```

```
1|+(char [128])buf = @00000FF0 "
```

```
00000000    10 00 00 00 97 74 D2
00000010    01 40 00 00 00 00 00
```

```
1:Zoom 4:Search 5:16<>10 6:Inspect 7:Watch 8:View 9:Rang
```

Fig. 15 Display Components

Code window

The child window displays source codes and disassembled code listings. The program counter is displayed in reverse video, and the break points are displayed with underline. There are two windows as the Code window.

Command window

The child window that handles dialog command input and displays the execution result and the console I/O characters in system call function.

Memory window

The child window displays the dump listing for specified memory area. It will be updated to the latest data contents at CPU execution stop.

Register window

The child window displays the CPU registers. It will be updated to the latest data contents at CPU execution stop.

Stack window

The child window displays the stack contents. It will be updated to the latest data contents at CPU execution stop.

Local window

The child window displays the local variable contents defined within the function that is corresponding to a program counter value. It will be updated to the latest data contents at CPU execution stop.

Watch window

The child window displays the data registered to Watch. It will be updated to the latest data contents at CPU execution stop.

Back Trace window

The child window displays the Back Trace of functions. It will be updated to the latest data contents at CPU execution stop.

Break window

The child window displays the break points currently set.

Memo window

The child window for the simple editor that is used during debugging. The window contents will be saved at PARTNER termination.

Inspect window

The child window that inspects (examines) and displays the variable contents.

History window

The child window displays the real-time trace memory in dump listing or dis-assembled listing.

I/O window

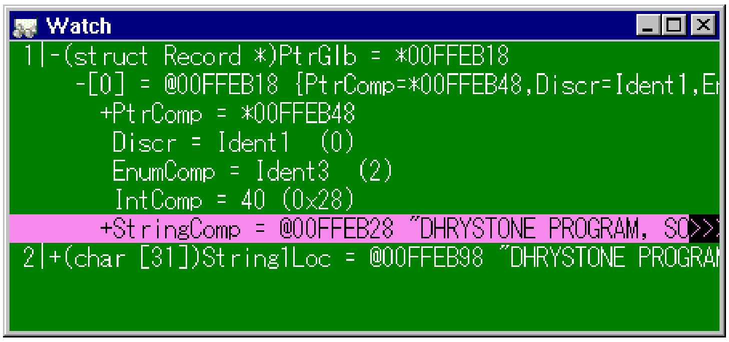
The child window displays the system I/O register contents.

4.1.2 Screen Display

The hidden part of data displayed in some windows cannot be displayed with horizontal scroll. Thus, to display the hidden portion, it is required to change the window size or font size.

The Code window, Memory window and Memo window can be scrolled horizontally.

If string search result is located in hidden portion, three inequality marks (>) will appear the line. (Fig. 16) To display the search result, change window or font size and execute [Search]-[Next] menu.

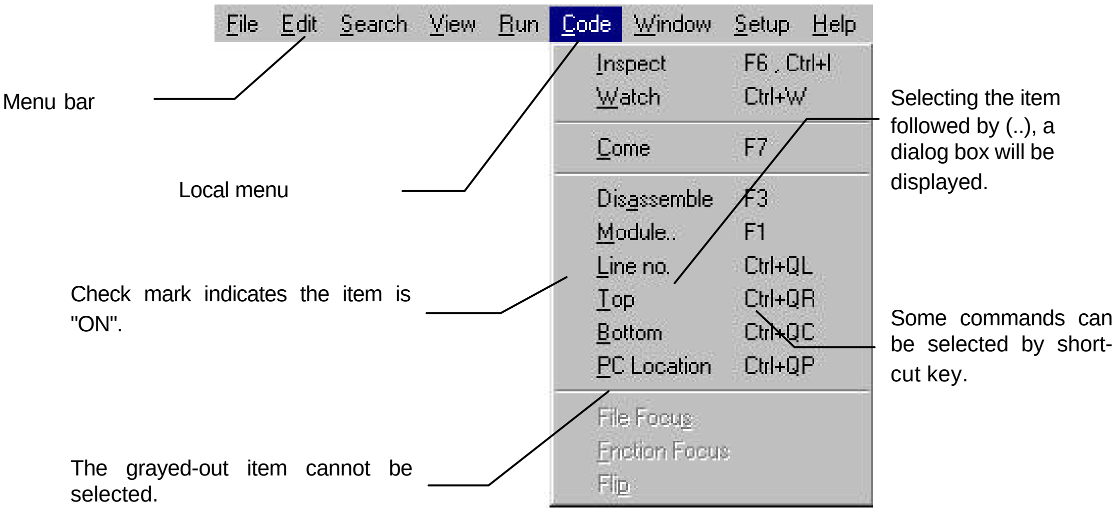


```
1|-(struct Record *)PtrGlb = *00FFEB18
    -[0] = @00FFEB18 {PtrComp=*00FFEB48,Discr=Ident1,E
      +PtrComp = *00FFEB48
       Discr = Ident1  (0)
       EnumComp = Ident3  (2)
       IntComp = 40 (0x28)
      +StringComp = @00FFEB28 "DHRYSTONE PROGRAM, SO>>>
2|+(char [31])String1Loc = @00FFEB98 "DHRYSTONE PROGRA
```

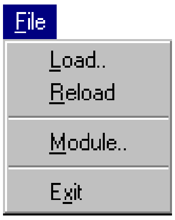
Fig. 16 If string search result is located in hidden portion of display.

4.2Menu

The Menu contains the PARTNER commands categorized in groups. There are two types of commands, the one executes immediately and the other displays a dialog box prompting the additional options.


File Edit Search View Run Code Window Setup Help
Inspect F6 , Ctrl+I
Watch Ctrl+W
Come F7
Disassemble F3
Module.. F1
Line no. Ctrl+QL
Top Ctrl+QR
Bottom Ctrl+QC
PC Location Ctrl+QP
File Focus
Fnction Focus
Flip
Menu bar
Local menu
Check mark indicates the item is "ON".
The grayed-out item cannot be selected.
Selecting the item followed by (..), a dialog box will be displayed.
Some commands can be selected by short-cut key.

4.2.1 File menu

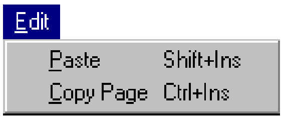


[File] menu contains the commands related to file operation.

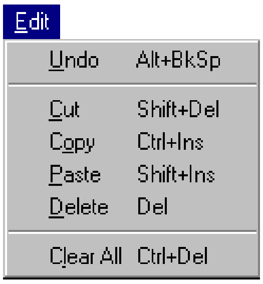
[Module] cannot be executed, if the focus is not on the Code window.

Menu Item	Function
Load	Displays the dialog box to select a file for user program. (refer to page 80, "6.1.1 Open dialog box")
Reload	Reloads the program previously loaded.
Module	Displays the dialog box to select modules contained in loaded user program. (refer to page 84, "6.1.8 Module dialog box)
Exit	Terminates PARTNER.

4.2.2 Edit menu



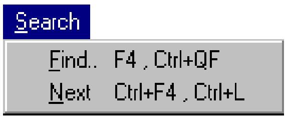
[Edit] menu contains commands that pass character string between windows via clipboard.



If the focus is on Memo window, [Edit] menu will have following items.

Menu Item	Function
Paste	Pastes the clipboard contents.
Copy Page	Copies the information displayed in the window to the clipboard.
Undo	Cancels previous editing.
Cut	Moves selected character string to the clipboard.
Copy	Copies selected character string to the clipboard.
Delete	Deletes selected character string.
Clear All	Clears all Memory window contents.

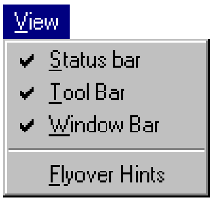
4.2.3 Search menu



[Search] menu contains commands to search character string in window.

Menu Item	Function
Find	Displays dialog box to specify search strings. (refer to page 82, "6.1.5 Find String dialog box")
Next	Searches specified string in specified direction.

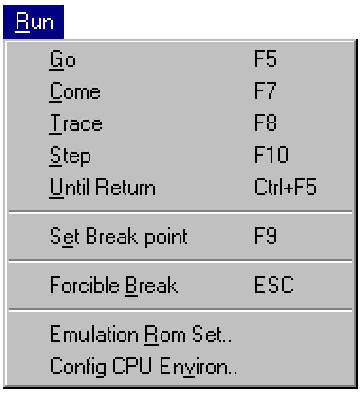
4.2.4 View menu



[View] menu contains commands to specify show/hide for Tool Bar.

Menu Item	Function
Status Bar	Selects show/hide of Status bar. (refer to page 51, "4.7 Status Bar")
Tool Bar	Selects show/hide of Tool Bar. (refer to page 46, "4.5 Tool Bar")
Window Bar	Selects show/hide of Window bar. (refer to page 50, "4.6 Window Bar")
Flyover Hints	Selects show/display of Hints about buttons.

4.2.5 Run menu



[Run] menu contains commands related to user program execution/stop.

[Come] and [Set Break point] can be executed, only if the focus is on the Code window.

As the menu structure differs depend on target CPU, refer to "dialog command" in "Part Edition" for detail.

Menu Item	Function
Go	Executes user program from current PC.
Come	Executes user program from current PC until cursor position.

(To be continued to next page)

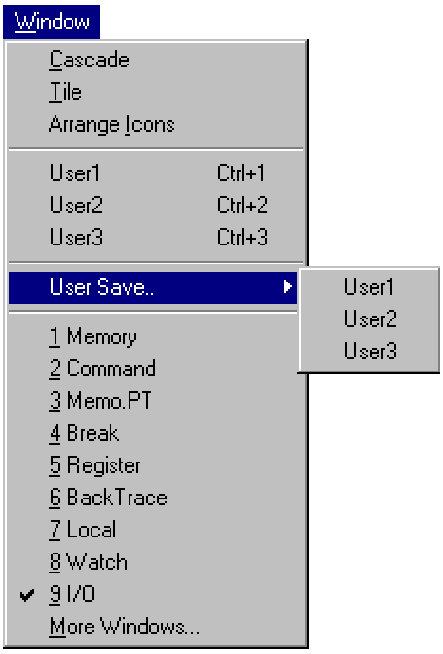
Menu Item	Function
Trace	Executes trace in source line or machine language basis.
Step	Executes step in source line or machine language basis.
Until Return	Executes current function until return to caller function.
Set Break point	Set or delete break points.
Forcible Break	Forcibly breaks executing user program, and returns control to PARTNER (during batch execution of command/macro, stop the batch execution).
Emulation ROM Set [*1]	Specifies emulation ROM area.
Config CPU Environ. [*1]	Specifies CPU environment.

[*1] not available for 1PC or CB series

4.2.6 Local menu

The child window menu focused currently, is registered to Local menu. It is also possible to be displayed and selected by right mouse button. For detail, please refer to page 53, "5 Child Window".

4.2.7 Window menu

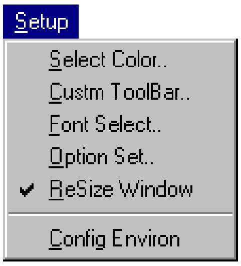


[Window] menu contains commands related to the overall control of windows.

The windows arrangement displayed in lower part of menu, might vary depend on the operation status.

Menu Item	Function
Cascade	Displays the windows not iconified in stack.
Tile	Displays the windows not iconified in side by side. (arrangement unique to PARTNER)
Arrange Icons	Arrange iconified windows.
User1	Restore the window arrangement saved by [User Save]-[User1].
User2	Restore the window arrangement saved by [User Save]-[User2].
User3	Restore the window arrangement saved by [User Save]-[User3].
User Save..	Saves current window arrangement.

4.2.8 Setup menu



[Setup] menu contains commands related to PARTNER setup.

Menu Item	Function
Select Color..	Displays dialog box to select the color of window or character (refer to page 82, "6.1.4 Color dialog box")
Custom ToolBar..	Displays dialog box to register Tool Bar buttons (refer to page 81, "6.1.3 Tool Bar dialog box").
Font Select..	Displays dialog box to select character size (refer to page 81, "6.1.2 Font dialog box").
Option Set..	Displays dialog box to specify various setting. *1
ReSize Window	Changes the size of each child window corresponding to main window size.
Config Environ	Invokes configuration program (RPTSETUP) from current project and changes start option and configuration file. However current PARTNER program terminates.

*1 Refer to "Child Window" in "Part Edition"

4.2.9 Help menu

[Help] menu contains commands related to help.

Menu Item	Function
Help Contents	Displays PARTNER help index.
Keyword Search..	Searches PARTNER help in key word.
About PARTNER..	Displays the information about PARTNER

4.3Short-cut Key

Short-cut key is the combination of keys defined for frequently used operations (commands), so that they can be executed more quickly.

4.3.1 Short-cut Keys Common to Each Windows

Following key operations are the common short-cut keys for each windows.

Key operation	Function
F1	changes the focused window display in between [Maximize] and [Restore]. Notes: For Code window and Command window, other command is defined.
F2 CTRL+O	Moves an active window to the bottom and change the active window to next one.
CTRL+B	Changes previous window to the active window. (reverse operation of F2 or CTRL+O)
F4 CTRL+Q F	Opens Find String dialog box and specifies the search string and direction. (refer to page 82, "6.1.5 Find String dialog box")
F5	Executes user program from current PC.
F8	Executes trace user program.
F10	Executes step user program.
CTRL+C ROLL UP	Displays lower part of one screen-full lines.
CTRL+E Up Arrow	Moves cursor to one line upper.
CTRL+L	Searches character string specified by F4 again from current cursor position.
CTRL+R ROLL DOWN	Displays upper part of one screen-full lines.
CTRL+X Down Arrow	Moves cursor to one line lower.

(To be continued to next page)

Key operation	Function
CTRL+INS	Copies the information of one screen-full lines in focused window to clipboard.
CTRL+GRPH+F1 CTRL+GRPH+0	Displays or iconifies Code window 0.
CTRL+GRPH+F2 CTRL+GRPH+C	Displays or iconifies Command window.
CTRL+GRPH+F3 CTRL+GRPH+W	Displays or iconifies Watch window.
CTRL+GRPH+F4 CTRL+GRPH+M	Displays or iconifies Memory window.
CTRL+GRPH+F5 CTRL+GRPH+R	Displays or iconifies Register window
CTRL+GRPH+F6	Displays or iconifies Stack window.
CTRL+GRPH+F7 CTRL+GRPH+L	Displays or iconifies Local window.
CTRL+GRPH+F8 CTRL+GRPH+T	Displays or iconifies Back Trace window.
CTRL+GRPH+F9 CTRL+GRPH+B	Displays or iconifies Break window.
CTRL+GRPH+F10 CTRL+GRPH+E	Displays or iconifies Memo window.
CTRL+1	Restores windows configuration User1.
CTRL+2	Restores windows configuration User2.
CTRL+3	Restores windows configuration User3.
GRPH+F10	Displays Local menu of currently focused window.

[Notice] For PC/AT compatible user, please substitute GRAPH key with Alt key, ROLL UP key with Page Down key, ROLL DOWN key with Page Up key, respectively to read above explanations.

4.3.2 Short-cut Keys Unique to Each Window

In addition to "4.3.1 Short-cut Key", short-cut keys unique to each window are defined. Regarding the short-cut keys for each windows, please refer to following page.

Window Name	Title	page
Code window	5.1.1 Code window Short-cut Key	55
Command window	5.2.1 Command window Short-cut Key	59
Memory window	Memory window Short-cut Key	*1
Register window	Register window Short-cut Key	*1
Stack window	5.5.1 Stack window Short-cut Key	65
Local window	5.6.1 Local window Short-cut Key	66
Back Trace window	5.7.1 Back Trace window Short-cut Key	68
Watch window	5.8.1 Watch window Short-cut Key	70
Break window	Break window Short-cut Key	*1
Inspect window	5.11.1 Inspect window Short-cut Key	75
Memo window	5.10.1 Memo window Short-cut Key	73
History window	History window Short-cut Key	*1,*2

*1refer to "Child Window" in "Part Edition"

*2 not available for PC or CB series

4.4 Mouse Operation

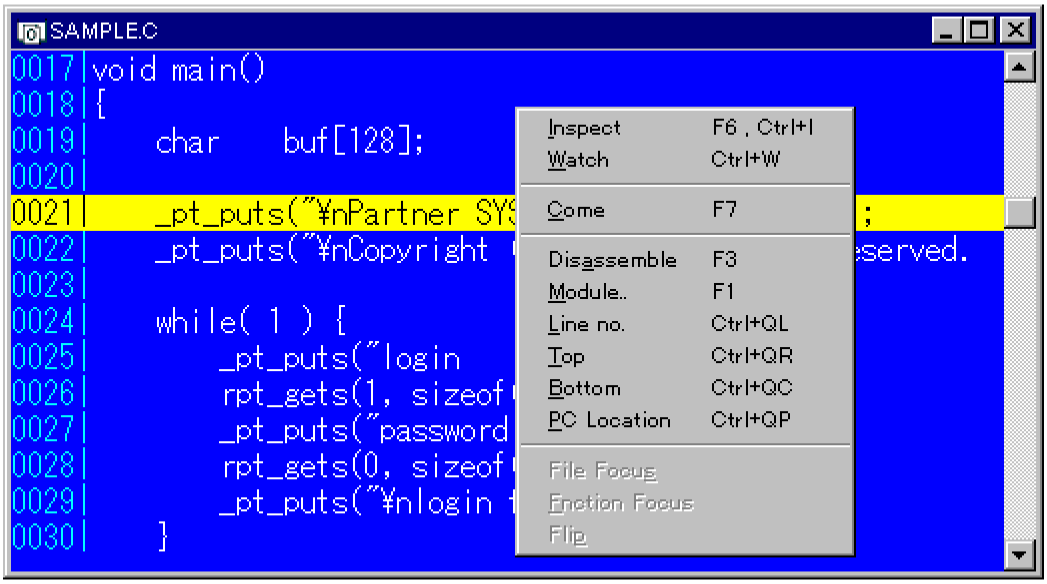
For mouse operations in window, there are common operation between all windows and unique one in each window.

4.4.1 Mouse Operations Common to Each Window

This section explains about mouse operation s common to each window.

Clicking Right Button

Clicking right button within focused window, the local menu of window will be displayed to select a command. (refer to page 38, "4.2.6 Local menu")

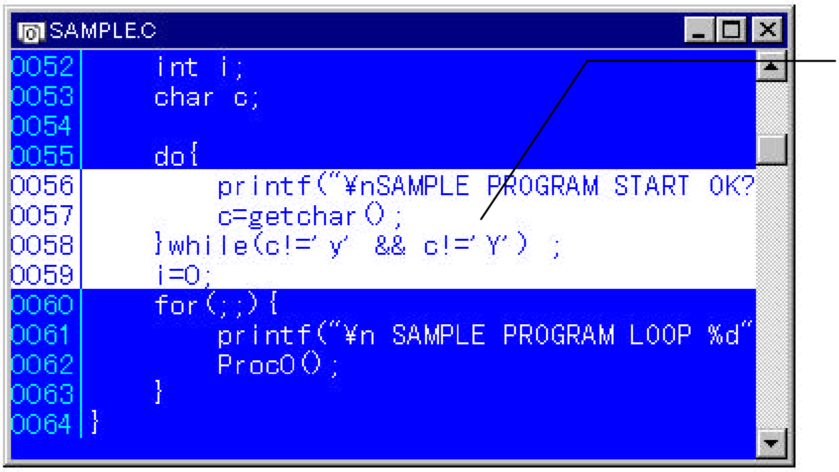


```
void main()
{
    char    buf[128];

    _pt_puts("¥nPartner SYS                    ;
    _pt_puts("¥nCopyright                      served.

    while( 1 ) {
        _pt_puts("login
        rpt_gets(1, sizeof
        _pt_puts("password
        rpt_gets(0, sizeof
        _pt_puts("¥nlogin
    }
```

Fig. 17 Local Menu in Code window

Dragging Left Button

Dragging left button within focused window and releasing the button after moving to a necessary position, the selected area displayed with reverse video will be copied to clipboard.



```
      int i;
      char c;

      do{
          printf("¥nSAMPLE PROGRAM START OK?
          c=getchar();
      }while(c!='y' && c!='Y');
      i=0;
      for(;;){
          printf("¥n SAMPLE PROGRAM LOOP %d"
          Proc0();
      }
}
```

selected region

Fig. 18 selecting copy region

Notes: Drag means to move mouse with pushing button.

4.4.2 Mouse Operations Unique to Each Window

In addition to "4.4.1 ", mouse operations unique to each window are defined. Regarding the mouse operations for each windows, please refer to following page.

Window	Title	Page
Code window	5.1.3 Mouse Operation in Code window	58
Memory window	Mouse Operation in Memory window	*1
Register window	Mouse Operation in Register window	*1
Local window	5.6.3 Mouse Operation in Local window	67
Back Trace window	5.7.3 Mouse Operation in Back Trace window	69
Watch window	5.8.3 Mouse Operation in Watch window	71
Break window	Mouse Operation in Break window	*1
Inspect window	5.11.3 Mouse Operation in Inspect window	76

*1refer to "Child Window" in "Part Edition"

4.5 Tool Bar

This section explains about PARTNER Tool Bar.

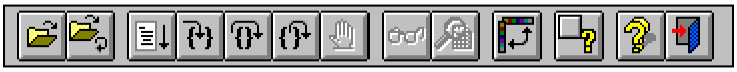


Fig. 19 Tool Bar example

Tool Bar is a convenience function that registers PARTNER commands to buttons and invokes the command by clicking mouse button on Tool Bar button.

Following section explains how to register buttons and functions of each button. [View]-[Tool Bar] command can select show/hide Tool Bar. (refer to page 37, "4.2.4 View menu")

4.5.1 Set Tool Bar

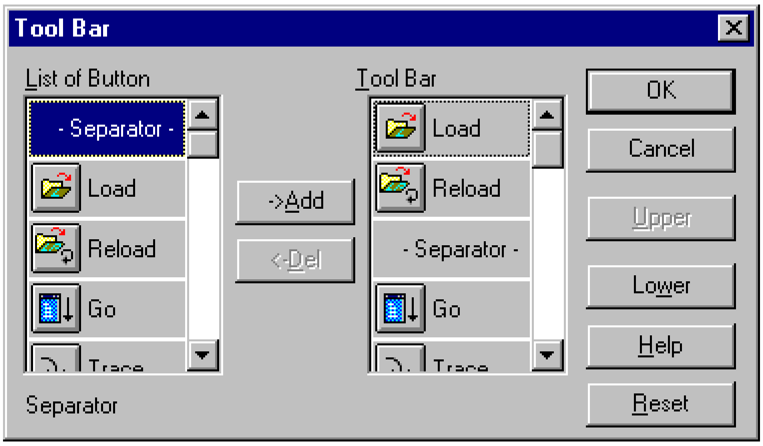
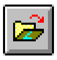
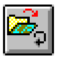
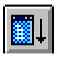
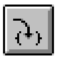
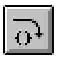
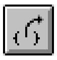
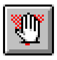
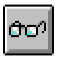
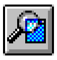
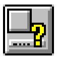
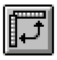
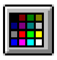


Fig. 20 [Tool Bar] dialog box

[Setup]-[Custom ToolBar] menu displays Set Tool Bar dialog box. Select the button to be registered to Tool Bar from [Button List] and add it to [Tool Bar definition]. (refer to page 81, "6.1.3 Tool Bar dialog box")

4.5.2 Function of Each Button

Following buttons can be registered to Tool Bar. The buttons and their functions are shown as follows.

Button	Function
	Loads user program. (refer to page 80, "6.1.1 Open dialog box")
	Reloads the program previously loaded.
	Executes user program from current PC.
	Executes trace in source line or machine language basis.
	Executes step in source line or machine language basis.
	Executes current function until return to caller function.
	Forcibly breaks executing user program, and returns control to PARTNER. (during batch execution of command/macro, stop the batch execution)
	Registers the variable at cursor position to Watch window. If there is no variable at cursor position to be registered as Watch, Watch dialog box will be displayed. (refer to page 83, "6.1.7 Set Watch data dialog box")
	Displays Inspect window for the variable at cursor position. If there is no variable at cursor position to be registered as Inspect, Set Inspect data dialog box will be displayed. (refer to page 83, "6.1.6 Set Inspect data dialog box")
	Changes Hint mode. If mouse cursor is on a Tool Bar buttons during Hint mode, the explanation of the button will be displayed in Status window.
	Changes Tool Bar display position.
	Displays the dialog box to set display color of currently focused window. (refer to page 82, "6.1.4 Color dialog box")

(To be continued to next page)

Button	Function
A	Displays the dialog box to set display font of currently focused window. (refer to page 81, "6.1.2 Font dialog box")
	Displays the dialog box to specify character string to be searched in currently focused window. (refer to page 82, "6.1.5 Find String dialog box")
	Searches the character string previously specified from current cursor position in currently focused window.
	Displays the dialog box to select a module in loaded user program. (refer to page 84, "6.1.8 Module dialog box")
1	Restores window arrangement 1.
2	Restores window arrangement 2.
3	Restores window arrangement 3.
	Displays the windows not iconified in stack.
	Displays the windows not iconified in side by side. (arrangement unique to PARTNER)
	Pastes character string from clipboard to currently focused window.
	Copies currently focused window display to clipboard.
	Displays the dialog box to set Tool Bar.(refer to page 81, "6.1.3 Tool Bar dialog box")
	Resize each child window corresponding to main window size.
	Exchanges display mode in Code window between source code and disassemble listing.
?	Displays PARTNER Help.
	Terminates PARTNER.

4.6 Window Bar

This chapter explains PARTNER Window Bar.

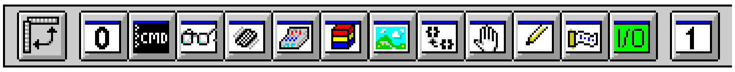


Fig. 21 Window Bar

The Window Bar is used to call each PARTNER window quickly. [View]-[Window Bar] command changes the show/hide of Window Bar. (refer to page 37, "4.2.4 View menu")

Followings show the assignment of each button.

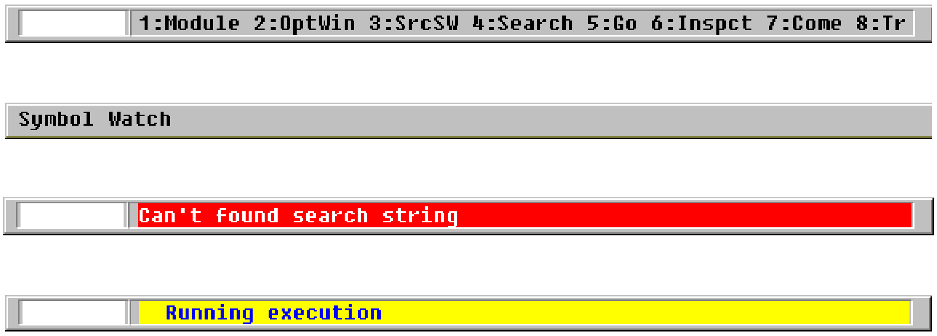
Button	Window
0	Code window
CMD	Command window
👓	Watch window
	Memory window
	Register window
	Stack window
	Local window
	Back Trace window
✋	Break window
✎	Memo window
	History window[*1]
I/O	I/O window
1	Code window1

[*1] not available for PC or CB series

[Notice] Configuration of Window Bar buttons cannot be changed.

4.7 Status Bar

In Status Bar, hints of short-cut keys and button/menu, error message and status display of PARTNER will be displayed. [View]-[Status bar] command changes the show/hide of Status Bar. (refer to page 37, "4.2.4 View menu")



```
1:Module 2:OptWin 3:SrcSW 4:Search 5:Go 6:Inspct 7:Come 8:Tr
```

```
Symbol Watch
```

```
Can't found search string
```

```
Running execution
```

Fig. 22 Status Bar example

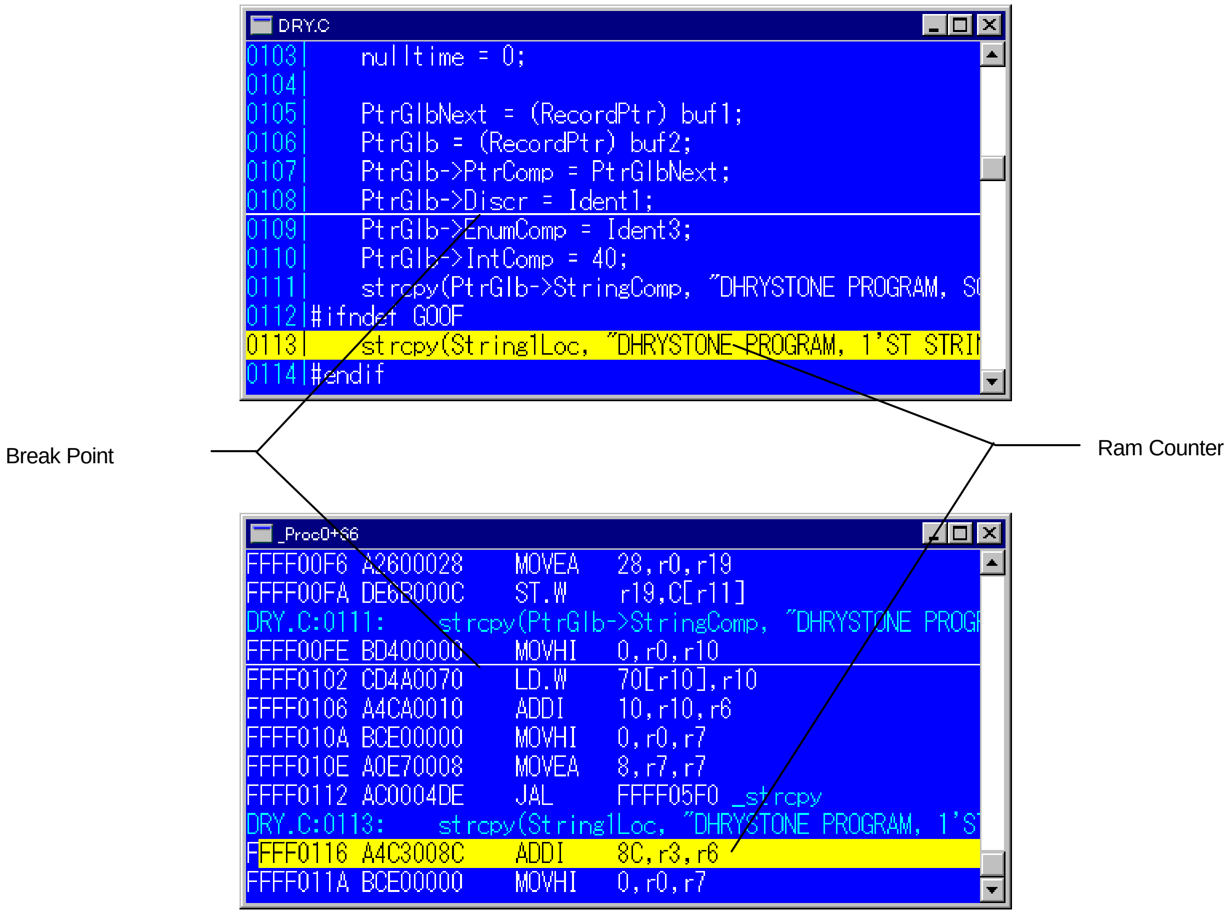
This page is intentionally left blank.

5 Child Window

Child window shows various information during debugging. And there are short-cut keys and menu and dialog box available for setting child window.

5.1 Code window

Code window displays source code or mixed display of assembler and source code. There are two Code windows that can be specified various display mode.



```
0103|     nulltime = 0;
0104|
0105|     PtrGlbNext = (RecordPtr) buf1;
0106|     PtrGlb = (RecordPtr) buf2;
0107|     PtrGlb->PtrComp = PtrGlbNext;
0108|     PtrGlb->Discr = Ident1;
0109|     PtrGlb->EnumComp = Ident3;
0110|     PtrGlb->IntComp = 40;
0111|     strcpy(PtrGlb->StringComp, "DHRYSTONE PROGRAM, SO
0112|#ifndef GOOF
0113|     strcpy(String1Loc, "DHRYSTONE PROGRAM, 1'ST STRIN
0114|#endif
```

```
FFFF00F6 A2600028    MOVEA   28,r0,r19
FFFF00FA DE6B000C    ST.W    r19,C[r11]
DRY.C:0111:     strcpy(PtrGlb->StringComp, "DHRYSTONE PROGR
FFFF00FE BD400000    MOVHI   0,r0,r10
FFFF0102 CD4A0070    LD.W    70[r10],r10
FFFF0106 A4CA0010    ADDI    10,r10,r6
FFFF010A BCE00000    MOVHI   0,r0,r7
FFFF010E A0E70008    MOVEA   8,r7,r7
FFFF0112 AC0004DE    JAL     FFFF05F0 _strcpy
DRY.C:0113:     strcpy(String1Loc, "DHRYSTONE PROGRAM, 1'ST
FFFF0116 A4C3008C    ADDI    8C,r3,r6
FFFF011A BCE00000    MOVHI   0,r0,r7
```

Fig. 23 Code window

5.1.1 Code window Short-cut Key

In Code window, various commands can be executed by following short-cut keys.

These short-cut keys will be applied, if the focus is on the Code window.

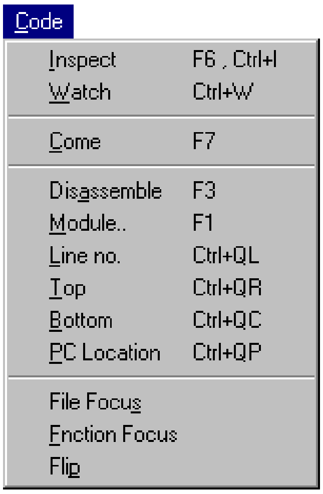
Key operation	Function	Dialog command
F1	Displays the dialog box to select module. (refer to page 84, "6.1.8 Module dialog box")	
F3 CTRL+F3	Exchanges display mode to disassembled code listing if the Code window is in source code listing, and to source code listing if in disassembled code listing. The line where current PC locates will be displayed.	V command U command
F6 CTRL+F6 CTRL+I	Sets variable at current cursor position to Inspect.	INS command
F7	Executes user program from current PC until current cursor position.	
F9	Sets or deletes break point at current cursor position.	BP command
CTRL+F1	Exchanges [Maximize]/[Restore] in Code window.	
CTRL+F2	Moves the focus to next window.	
CTRL+F4 CTRL+L	Searches again with string and direction that specified in previous search, from current cursor position.	
CTRL+F5	Executes user program until return to caller function from current function.	G command
CTRL+F7 CTRL+W	Registers the variable at current cursor position to Watch window.	W command
CTRL+F8 CTRL+V	Display view of variable at current cursor position in Command window.	VAL command ? command

(To be continued to next page)

Key operation	Function	Dialog command
CTRL+F9 CTRL+G	Copies the string at current cursor position to cursor position in Command window.	
CTRL+F10	Copies the string at cursor position to clipboard.	
CTRL+Q L	Displays the dialog box to specify the line number/address to be displayed.	V command U command
CTRL+Q R	Displays the top of source code.	
CTRL+Q C	Displays the bottom of source code.	
CTRL+Q J	Restore the window arrangement at startup.	
CTRL+Q P	Displays the line where current PC locates.	
CTRL+F	Moves the cursor to next strings if display is in source code listing, moves the cursor in the order of address, code and mnemonic if display is in disassembled code listing.	
CTRL+A	Moves the cursor to previous strings, if display is in source code listing.	
CTRL+S Left Arrow	Moves the cursor one unit left.	
CTRL+D Right Arrow	Moves the cursor one unit right.	
SHIFT+ ROLL UP	Moves the cursor to previous PC position.	
SHIFT+ ROLLDOWN	Moves the cursor to next PC position.	

[Notice]SHIFT key operations other than above are described in page 59, "5.2.1 Command window Short-cut Key ". The key operations, "SHIFT+ROLL UP" and "SHIFT+ROLL DOWN", only move cursor to 32 steps of reserved PC position. This capability is not anything like undo of execution or break, but used for checking the previous PC position, as such.

5.1.2 Local Menu in Code window



[Code] menu contains commands related to control of Code window.

Menu Item	Function
Inspect	Displays Inspect window for the variable at cursor position. If the string at cursor position is not the one to be inspected, Set Inspect data dialog box will be displayed. (refer to page 83, "6.1.6 Set Inspect data dialog box")
Watch	Registers the variable at cursor position to Watch window. If the string at cursor position is not the one to be registered to Watch, Set Watch data dialog box will be displayed. (refer to page 83, "6.1.7 Set Watch data dialog box")
Disassemble/ Source	Exchanges source listing and disassembled code listing display.
Module	Displays the dialog box to select modules contained in loaded user program. (refer to page 84, "6.1.8 Module dialog box")
Line no/Address	Displays the dialog box to select line number/address to be displayed. (refer to page 84, "6.1.9 Set line no. dialog box" and page 85, "6.1.10 Addressing dialog box (code)")
Top	Displays the top of displayed file.
Bottom	Displays the bottom of displayed file.

(To be continued to next page)

Menu Item	Function
PC Location	Displays current PC position.
File Focus	If the check is marked at file where current PC exists, when PC moves to the file other than checked one the focus will move to another Code window, and when PC moves to the checked file the focus will move to the Code window.
Function Focus	If the check is marked at function where current PC exists, when PC moves to the function other than checked one the focus will move to another Code window, and when PC moves to the checked function the focus will move to the Code window.
Flip	The focus of Code window moves to another Code window, every time the function where PC locates changes.

[Notice] The File Focus, Function Focus and Flip are valid, only if both of two Code windows are displayed.

5.1.3 Mouse Operation in Code window

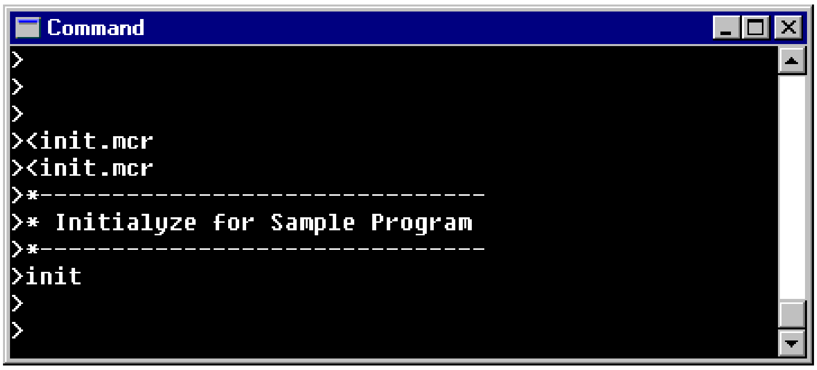
In Code window, frequently used commands are assigned to mouse operations.

Function	Mouse Operation	Dialog command
Inspect	Double click left button on the variable	INS command
Break Point	Click left button on line number/address	BP command
Executes Trace	SHIFT+Click left button	T command
Executes Step	SHIFT+Click right button	P command
[Run] menu (refer to page 37)	CTRL+Click right button	

5.2 Command window

Command window displays dialog command input, execution result and character I/O regarding system call function.

For dialog command to enter input to Command window, refer to On-line Help.



```
>
>
>
><init.mcr
><init.mcr
>*-------------------------------
>* Initialyze for Sample Program
>*-------------------------------
>init
>
>
```

Fig. 24 Command window

5.2.1 Command window Short-cut Key

In Command window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Command window.

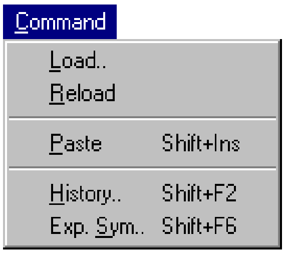
Key operation	Function
SHIFT+F1	Copies one character from the command previously entered.
SHIFT+F2	Displays the dialog box for command history previously entered. (refer to page 85, "6.1.11 Command History dialog box")
SHIFT+F3	Copies character string to current cursor position from the command previously entered.
SHIFT+F4 CTRL+A	Moves cursor to the beginning of line.

(To be continued to next page)

Key operation	Function
SHIFT+F5 CTRL+F	Moves cursor to the end of line.
SHIFT+F6	Displays the list of global symbols starts with the last strings at command input line, and completes command. (refer to page 86, "6.1.12 Expand Symbol dialog box")
SHIFT+F7 CTRL+U	Deletes all characters in current edit line.
SHIFT+F8	Deletes all characters in current edit line and all contents of command history.
BS CTRL+H	Deletes one character left at cursor.
Left Arrow SHIFT+ Left Arrow CTRL+S	Moves cursor to left.
Right Arrow SHIFT+Right Arrow CTRL+D	Moves cursor to right. If cursor is at end of line, operates the same as SHIFT+F3.
DEL CTRL+G	Deletes a character at cursor position.
INS CTRL+V	Toggles between Insert mode and Overwrite mode.
SHIFT+ Up Arrow CTRL+W	Displays the last command history. If there is string in command input line, searches and displays the command start with that string in the order of the latest comes the first. If this key is entered again, searches previous command history.
SHIFT+ Down Arrow	Displays the last command history. If there is string in command input line, searches and displays the command start with that string in the order of the oldest comes the first. If this key is entered again, searches next command history.
SHIFT+INS	Enters the character string in clipboard as commend input.

[Notice] Key operation of function key and CTRL+function key are similar to the short-cut key in Code window. (refer to page 55, "5.1.1 ") However, the commands depend on cursor position are not available.

5.2.2 Local Menu in Command window

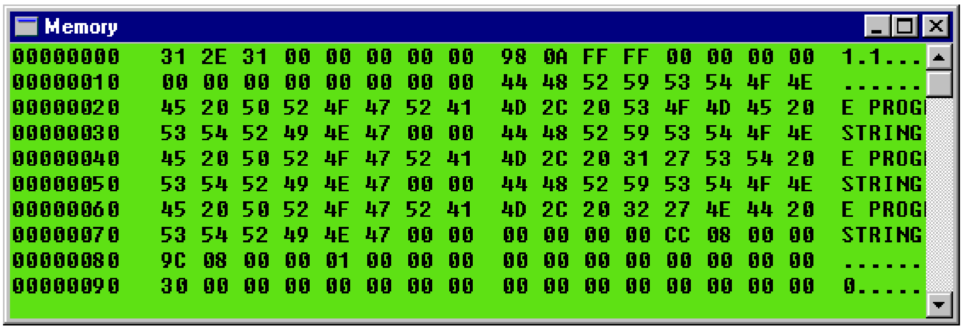


[Command] menu contains the commands related to Command window control.

Menu Item	Function
Load..	Displays dialog box to load user program.
Reload	Reloads the program previously loaded.
Paste	Pastes the clipboard contents.
History	Displays Command History dialog box. (refer to page 85, "6.1.11 Command History dialog box")
Exp. Sym..	Displays the dialog box of global symbols listing starts with the last strings at command input line. (refer to page 86, "6.1.12 Expand Symbol dialog box")

5.3 Memory window

Memory window displays memory contents in various format.



```
00000000   31 2E 31 00 00 00 00 00  98 0A FF FF 00 00 00 00  1.1...
00000010   00 00 00 00 00 00 00 00  44 48 52 59 53 54 4F 4E  ......
00000020   45 20 50 52 4F 47 52 41  4D 2C 20 53 4F 4D 45 20  E PROG
00000030   53 54 52 49 4E 47 00 00  44 48 52 59 53 54 4F 4E  STRING
00000040   45 20 50 52 4F 47 52 41  4D 2C 20 31 27 53 54 20  E PROG
00000050   53 54 52 49 4E 47 00 00  44 48 52 59 53 54 4F 4E  STRING
00000060   45 20 50 52 4F 47 52 41  4D 2C 20 32 27 4E 44 20  E PROG
00000070   53 54 52 49 4E 47 00 00  00 00 00 00 CC 08 00 00  STRING
00000080   9C 08 00 00 01 00 00 00  00 00 00 00 00 00 00 00  ......
00000090   30 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  0.....
```

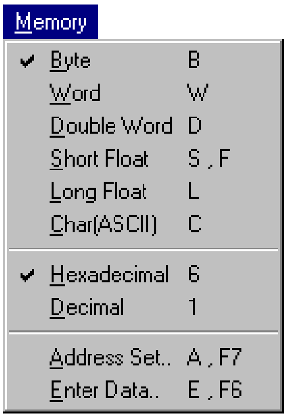
Fig. 25 Memory window

5.3.1 Memory window Short-cut Keys

In Memory window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Memory window.

Key operation	Function
F7 A	Displays dialog box to specify display start address. (refer to page 86, "6.1.13 Addressing dialog box (memory)")
F6 E Enter	Displays dialog box to change data. (refer to page 87, "6.1.14 Set Data dialog box")
B	Specifies memory data display in byte format.
W	Specifies memory data display in word format.
D	Specifies memory data display in double word format.
S,F	Specifies memory data display in short float format.
L	Specifies memory data display in long float format.
C	Specifies memory data display in ASCII format.
6,H	Specifies memory data display in hexadecimal format.
1	Specifies memory data display in decimal format.

5.3.2 Local Menu in Memory window



[Memory] menu contains the commands related to Memory window control.

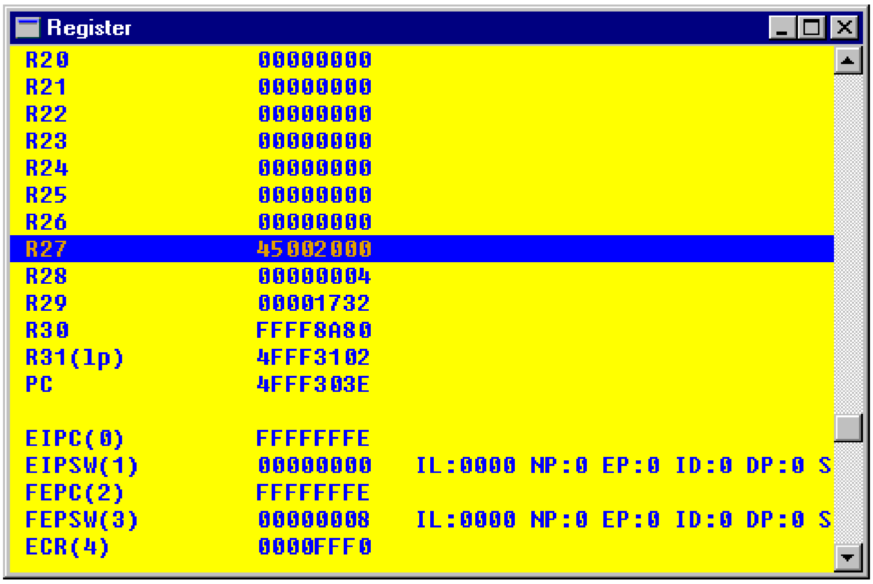
Menu Item	Function
Byte	Specifies memory data display in byte format.
Word	Specifies memory data display in word format.
Double Word	Specifies memory data display in double word format.
Short Float	Specifies memory data display in short float format.
Long Float	Specifies memory data display in long float format.
Char(ASC)	Specifies memory data display in ASCII format.
Hexadecimal	Specifies memory data display in hexadecimal format.
Decimal	Specifies memory data display in decimal format.
Address Set..	Displays dialog box to specify display start address. (refer to page 86, "6.1.13 Addressing dialog box (memory)")
Edit Data..	Displays dialog box to change data at cursor. (refer to page 87, "6.1.14 Set Data dialog box")

5.3.3 Mouse Operation in Memory window

Double clicking left mouse button on data portion, dialog box to change data will be displayed. (refer to page 87, "6.1.14 Set Data dialog box")

5.4 Register window

Register window displays program register and system register in the format shown in Fig. 26.



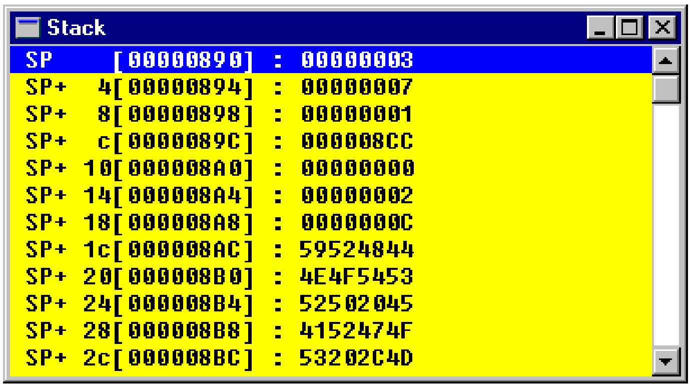
```
R20            00000000
R21            00000000
R22            00000000
R23            00000000
R24            00000000
R25            00000000
R26            00000000
R27            45002000
R28            00000004
R29            00001732
R30            FFFF8A80
R31(lp)        4FFF3102
PC             4FFF303E

EIPC(0)        FFFFFFFE
EIPSW(1)       00000000   IL:0000 NP:0 EP:0 ID:0 DP:0 S
FEPC(2)        FFFFFFFE
FEPSW(3)       00000008   IL:0000 NP:0 EP:0 ID:0 DP:0 S
ECR(4)         0000FFF0
```

Fig. 26 Register window display example

As the register components depend on CPU, please refer to Register window in "Part Edition" for detail.

5.5 Stack window

Stack window displays current stack memory contents.



```
SP      [00000890] : 00000003
SP+   4[00000894] : 00000007
SP+   8[00000898] : 00000001
SP+   c[0000089C] : 000008CC
SP+  10[000008A0] : 00000000
SP+  14[000008A4] : 00000002
SP+  18[000008A8] : 0000000C
SP+  1c[000008AC] : 59524844
SP+  20[000008B0] : 4E4F5453
SP+  24[000008B4] : 52502045
SP+  28[000008B8] : 4152474F
SP+  2c[000008BC] : 53202C4D
```

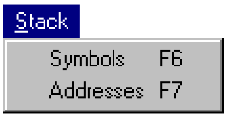
Fig. 27 Stack window

5.5.1 Stack window Short-cut Key

In Stack window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Stack window.

Key operation	Function
F6	Displays symbols of stack memory contents.
F7	Display real memory address of stack.

5.5.2 Local Menu in Stack window

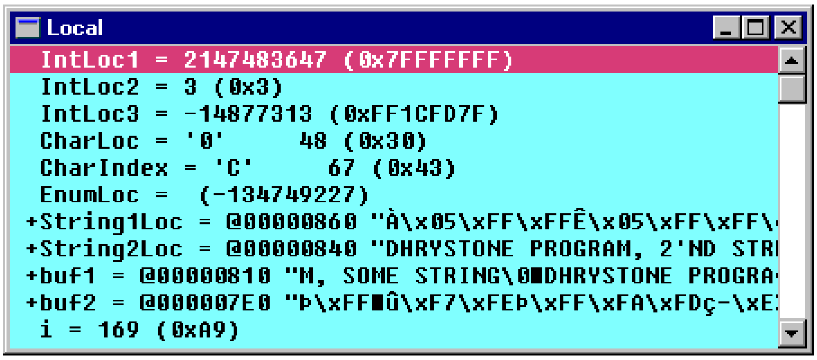


[Stack] menu contains the commands related to Stack window control.

Menu Item	Function
Symbols	Displays symbols of stack value.
Addresses	Displays address of stack memory.

5.6 Local window

Local window displays the contents of local variables which belongs to the function corresponding to current program counter (PC). The '+' mark at the left of local variable name indicates that the variable contains variable elements to be displayed. If the variable elements are displayed, '-' sign will displayed at the left of local variable name.



```
  IntLoc1 = 2147483647 (0x7FFFFFFF)
  IntLoc2 = 3 (0x3)
  IntLoc3 = -14877313 (0xFF1CFD7F)
  CharLoc = '0'      48 (0x30)
  CharIndex = 'C'      67 (0x43)
  EnumLoc =  (-134749227)
+String1Loc = @00000860 "À\x05\xFF\xFFÊ\x05\xFF\xFF\
+String2Loc = @00000840 "DHRYSTONE PROGRAM, 2'ND STR
+buf1 = @00000810 "M, SOME STRING\0■DHRYSTONE PROGRA
+buf2 = @000007E0 "Þ\xFF■Û\xF7\xFEÞ\xFF\xFA\xFDç-\xE
  i = 169 (0xA9)
```

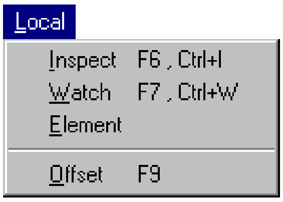
Fig. 28 Local window

5.6.1 Local window Short-cut Key

In Local window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Local window.

Key operation	Function
F6 CTRL+I	Inspects the local variable at cursor position.
F7 CTRL+W	Registers the local variable at cursor position to Watch window.
F9	Displays the local variable offset.
Enter	Exchanges show/hide of local variable which has variable elements (the one with '+' or '-' mark at the left of variable name).

5.6.2 Local Menu in Local window



[Local] menu contains the commands related to Local window control.

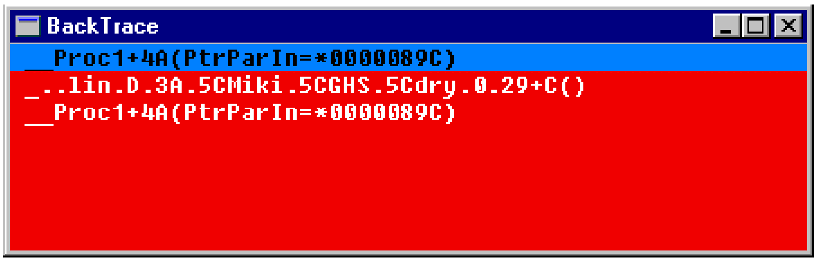
Menu Item	Function
Inspect	Displays Inspect window for the local variable selected by cursor.
Watch	Registers the local variable selected by cursor to Watch window.
Element	Exchanges the show/hide of the local variable contents for the one with elements such as array or structure.
Offset	Exchanges the show/hide of the local variable offset value.

5.6.3 Mouse Operation in Local window

Double clicking left mouse button on the local variable part with variable elements, show/hide of the variable elements will be exchanged.

5.7 Back Trace window

Back Trace window displays the process how current function is called from main() function.



```
__Proc1+4A(PtrParIn=*0000089C)
_..lin.D.3A.5CMiki.5CGHS.5Cdry.0.29+C()
__Proc1+4A(PtrParIn=*0000089C)
```

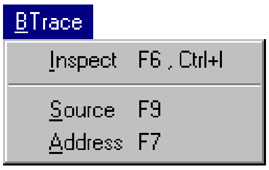
Fig. 29 Back Trace window

5.7.1 Back Trace window Short-cut Key

In Back Trace window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Local window.

Key operation	Function
F6 CTRL+I Enter	Displays source code of current cursor line to Code window.
F7	Exchange show/hide of address of symbol displayed in the window.
F9	Exchange show/hide of source name and line number of symbol displayed in the window.

5.7.2 Local Menu in Back Trace window



[BTrace] menu contains the commands related to Back Trace window control.

Menu Item	Function
Inspect	Displays Back Trace address selected by cursor in Code window.
Source	Exchanges show/hide of source name and line number.
Address	Exchanges show/hide of address.

5.7.3 Mouse Operation in Back Trace window

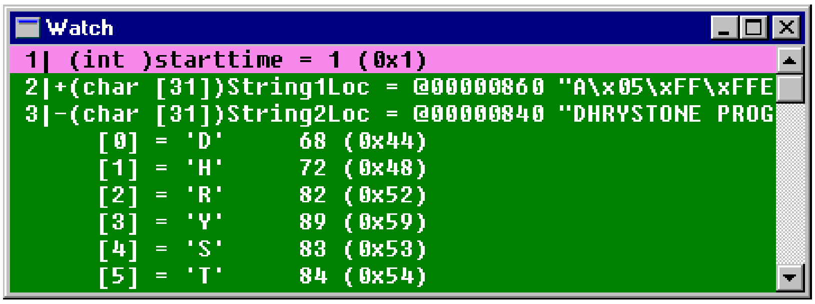
Double clicking left mouse button on Back Trace portion, source code of selected lines will be displayed in Code window.

5.8 Watch window

Watch window displays the variables registered to Watch.

The '+' mark at the left of variable name indicates that the variable contains variable element that can be displayed.

If the variable elements are displayed, '-' mark will be displayed at the left of variable name.



```
 1| (int )starttime = 1 (0x1)
 2|+(char [31])String1Loc = @00000860 "A\x05\xFF\xFFE
 3|-(char [31])String2Loc = @00000840 "DHRYSTONE PROG
     [0] = 'D'     68 (0x44)
     [1] = 'H'     72 (0x48)
     [2] = 'R'     82 (0x52)
     [3] = 'Y'     89 (0x59)
     [4] = 'S'     83 (0x53)
     [5] = 'T'     84 (0x54)
```

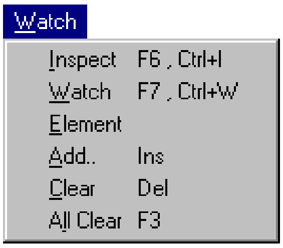
Fig. 30 Watch window

5.8.1 Watch window Short-cut Key

In Watch window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Watch window.

Key operation	Function
F3	Deletes all variables.
F6 CTRL+I	Inspects the variable selected by cursor.
Enter	Exchanges show/hide of local variable which has variable elements (the one with '+' or '-' mark at the left of variable name).
INS	Displays the dialog box to add Watch registration.
DEL	Deletes the variable selected by cursor.

5.8.2 Local Menu in Watch window



[Watch] menu contains the commands related to Watch window control.

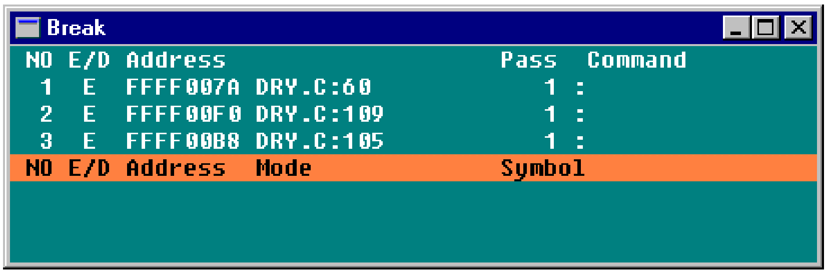
Menu Item	Function
Inspect	Displays Inspect window for the variable selected by cursor.
Watch	Registers variable selected by cursor to Watch widow.
Element	Exchanges the show/hide of the local variable contents for the one with elements such as array or struct.
Add.	Displays dialog box to enter the variable name that is newly registered to Watch window.
Clear	Deletes the variable selected by cursor from Watch window.
All Clear	Deletes all variables registered to Watch window.

5.8.3 Mouse Operation in Watch window

Double clicking left mouse button on the variable part which has variable elements, show/hide of the variable elements will be exchanged.

5.9 Break window

Break window displays the break points currently set.



```
NO E/D Address                        Pass  Command
 1  E  FFFF007A DRY.C:60                 1 :
 2  E  FFFF00F0 DRY.C:109                1 :
 3  E  FFFF00B8 DRY.C:105                1 :
NO E/D Address  Mode                  Symbol
```

Fig. 31 Break window

As Break window might vary depend on target CPU, please refer to Break window in "Part Edition" for detail.

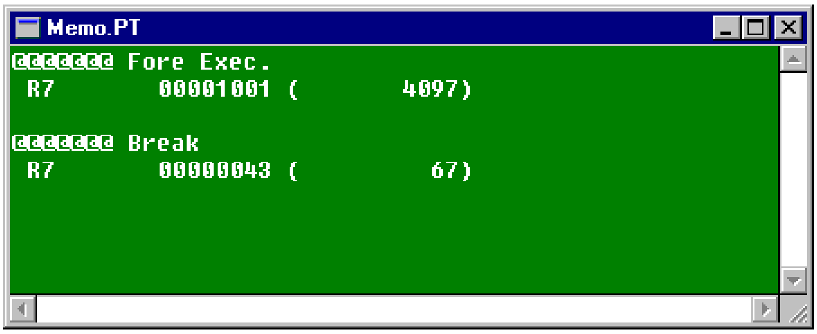
5.10 Memo window

Memo window is a simple text editor that replaces Window's notepad. As this window supports clipboard, the data transfer between each window is available.

For example, if register values before start execution and after execution break are copied from Register window, the comparison of register values will be available.

And PARTNER macro commands entered into Memo window also can be entered into Command window.

The contents of Memo window will be saved to file (memo.pt) at the termination of PARTNER and will be restored at startup time.



```
@@@@@@@ Fore Exec.
 R7        00001001 (        4097)

@@@@@@@ Break
 R7        00000043 (          67)
```

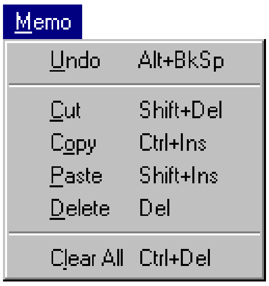
Fig. 32 Memo window

5.10.1 Memo window Short-cut Key

In Memo window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Memo window.

Key operation	Function
CTRL+INS	Copies selected character string to clipboard.
CTRL+DEL	Deletes all contents of Memo window.
SHIFT+INS	Pastes character string to cursor position from clipboard.
SHIFT+DEL	Copies selected character string to clipboard and deletes from Memo window.
GRPH+BS	Undo previous editing.

5.10.2 Local Menu in Memo window

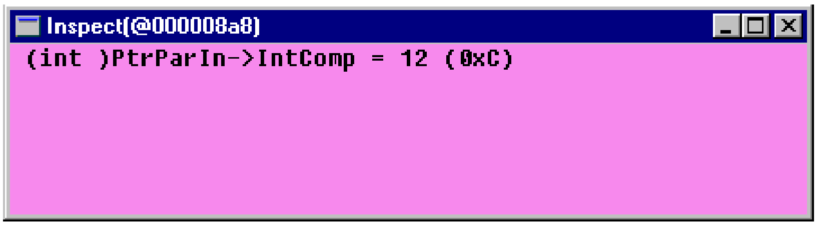


[Memo] menu contains the commands related to Memo window control.

Menu Item	Function
Undo	Cancels previous editing.
Cut	Moves selected character string to clipboard.
Copy	Copies selected character string to clipboard.
Paste	Pastes contents of clipboard.
Delete	Deletes selected character string.
Clear All	Deletes all contents of Memo window.

5.11 Inspect window

Inspect window displays specified variable in the format that corresponding to data structure of the variable.



```
(int )PtrParIn->IntComp = 12 (0xC)
```

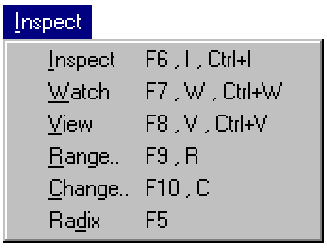
Fig. 33 Inspect window

5.11.1 Inspect window Short-cut Key

In Inspect window, various commands can be executed by following short-cut keys. These short-cut keys are applied, only if the focus is on Inspect window.

Key operation	Function
F5	Exchanges display format of array variable between decimal and hexadecimal.
F6 I CTRL+I Enter	Displays selected variables in Inspect window.
F7 W CTRL+W	Registers selected variables to Watch window.
F8 V CTRL+V	Displays selected variable in Command window.
F9 R	Displays dialog box to specify the display range of selected variable.
F10 C	Displays dialog box to change the value of selected variables.

5.11.2 Local Menu in Inspect window



Inspect

Inspect	F6 , I , Ctrl+I
Watch	F7 , W , Ctrl+W
View	F8 , V , Ctrl+V
Range..	F9 , R
Change..	F10 , C
Radix	F5

[Inspect] menu contains the commands related to Inspect window control.

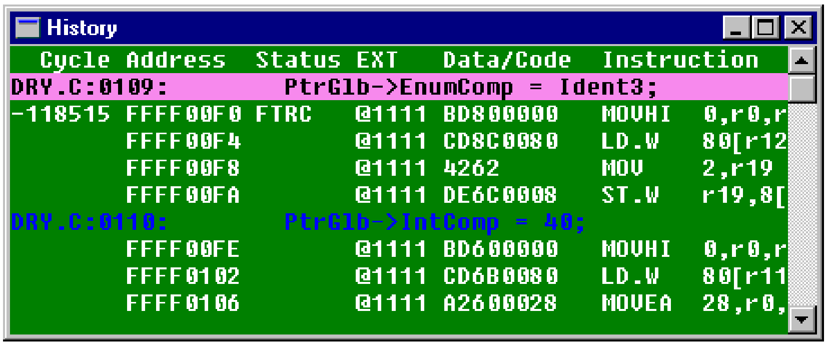
Menu Item	Function
Inspect	Opens Inspect window for variable selected by cursor.
Watch	Registers variable selected by cursor to Watch window.
View	Displays view of variable selected by cursor in Command window.
Range...	Displays dialog box to specify the range of variable element for displaying in Inspect window.
Change..	Displays dialog box to change value of variable selected by cursor.
Radix	Changes radix base of data displayed in array.

5.11.3 Mouse Operation in Inspect window

Double clicking left mouse button on the variable portion, specified variable will be displayed in Inspect window.

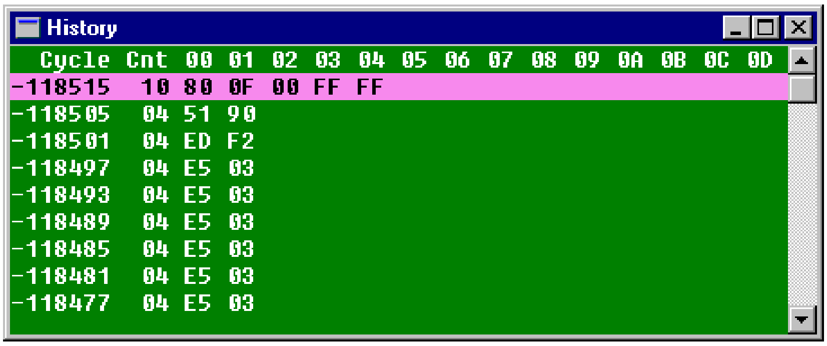
5.12 History window

History window displays the contents of Real-time Trace memory in hexadecimal dump or disassembled code listing.



```
  Cycle Address  Status EXT   Data/Code   Instruction
DRY.C:0109:          PtrGlb->EnumComp = Ident3;
-118515 FFFF00F0 FTRC     @1111 BD800000   MOVHI   0,r0,r
        FFFF00F4          @1111 CD8C0080   LD.W    80[r12
        FFFF00F8          @1111 4262       MOV     2,r19
        FFFF00FA          @1111 DE6C0008   ST.W    r19,8[
DRY.C:0110:          PtrGlb->IntComp = 40;
        FFFF00FE          @1111 BD600000   MOVHI   0,r0,r
        FFFF0102          @1111 CD6B0080   LD.W    80[r11
        FFFF0106          @1111 A2600028   MOVEA   28,r0,
```

Fig. 34 History (disassembled code listing) example



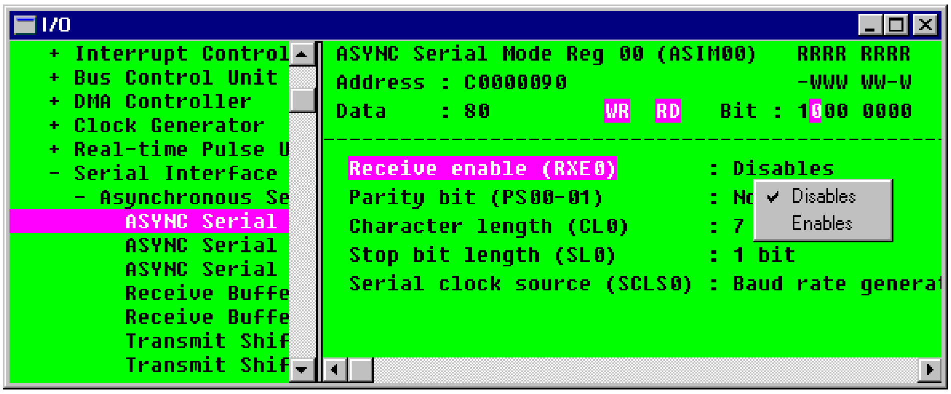
```
  Cycle Cnt 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D
-118515  10 80 0F 00 FF FF
-118505  04 51 90
-118501  04 ED F2
-118497  04 E5 03
-118493  04 E5 03
-118489  04 E5 03
-118485  04 E5 03
-118481  04 E5 03
-118477  04 E5 03
```

Fig. 35 History (dump listing) example

As History window display depends on target CPU, please refer to History window in "Part Edition" for detail. This function is not available for CP or CB series.

5.13 I/O window

I/O window displays and sets I/O registers which are implemented in target CPU. (Fig. 36)



```
  + Interrupt Control     ASYNC Serial Mode Reg 00 (ASIM00)    RRRR RRRR
  + Bus Control Unit      Address : C0000090                   -WWW WW-W
  + DMA Controller        Data    : 80          WR  RD   Bit : 1000 0000
  + Clock Generator       ----------------------------------------------
  + Real-time Pulse U
  - Serial Interface        Receive enable (RXE0)         : Disables
    - Asynchronous Se       Parity bit (PS00-01)          : No
        ASYNC Serial        Character length (CL0)        : 7
        ASYNC Serial        Stop bit length (SL0)         : 1 bit
        ASYNC Serial        Serial clock source (SCLS0)   : Baud rate generat
        Receive Buffe
        Receive Buffe
        Transmit Shif
        Transmit Shif
```

Fig. 36 I/O window display example

As I/O window display depends on target CPU, please refer to I/O window in "Part Edition" for detail.

6 Dialog command

There are two ways in debugging with PARTNER, the one is via dialog box or Tool Bar, and the other is entering dialog command in Command window.

6.1 Dialog Box

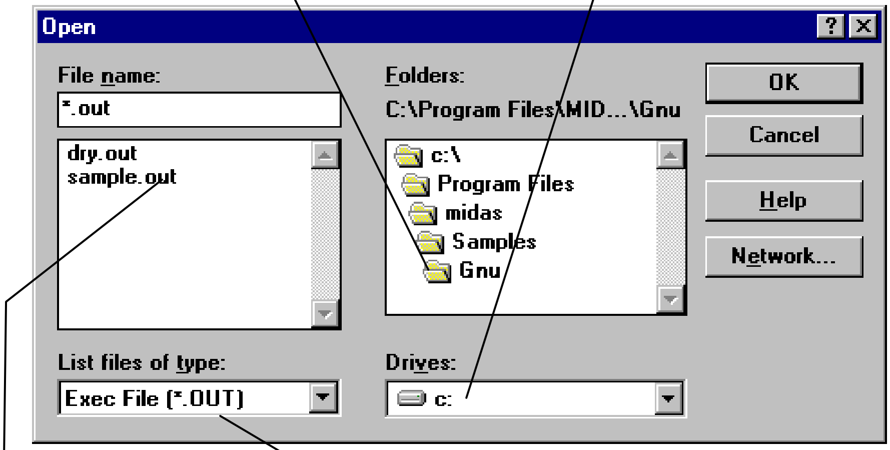
PARTNER has a lot of dialog boxes. Following section explains how to use each dialog box.

As they might depend on target CPU, please also refer to the section of Dialog Command in "Part Edition".

6.1.1 Open dialog box

[File]-[Load] ,

[Open] dialog box selects and loads user program to be debugged.



Double click mouse button on the directory contains program to be debugged.

Select a drive.

Select a directory contains a file to be debugged.

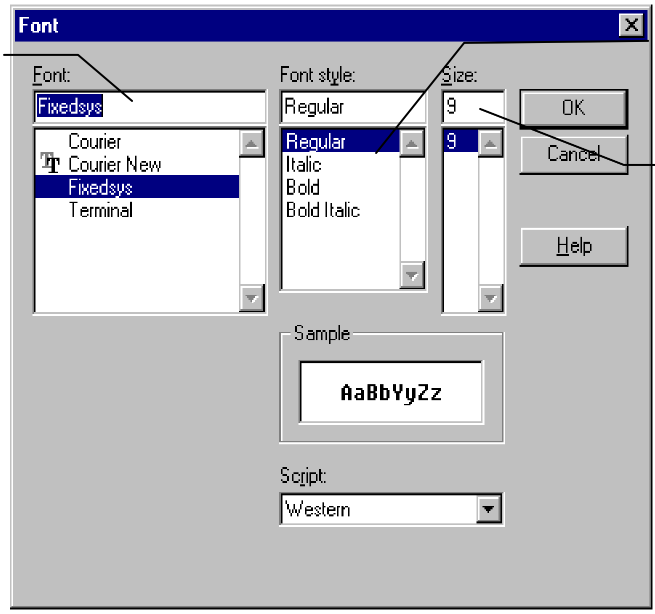
Select the file type to display in [file name:] box

6.1.2 Font dialog box

[Setup]-[Font select] ,

[Font] dialog box specifies the font of window currently selected.

Specifies font name to be used.



Font: Fixedsys

Font style: Regular

Size: 9

Script: Western

Specifies font type to be used.

Specifies font size to be used.

6.1.3 Tool Bar dialog box

[Setup]-[Custom ToolBar],

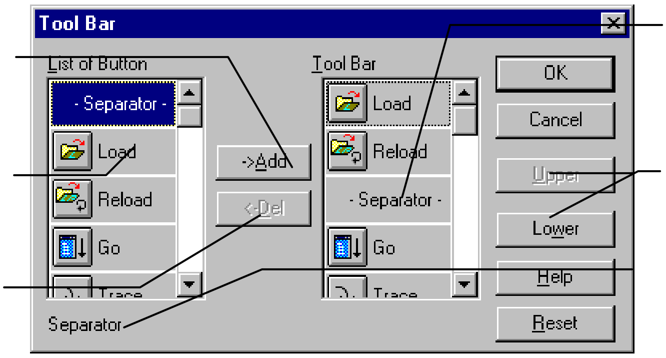
[Tool Bar] dialog box specifies the buttons to be registered to Tool Bar.

Entering DEL key in [Tool Bar] list box, the selected buttons will be deleted.

Adds selected button to Tool Bar.

Specifies the button to be registered.

Deletes selected button from Tool Bar.



Specifies the position where the button to be inserted.

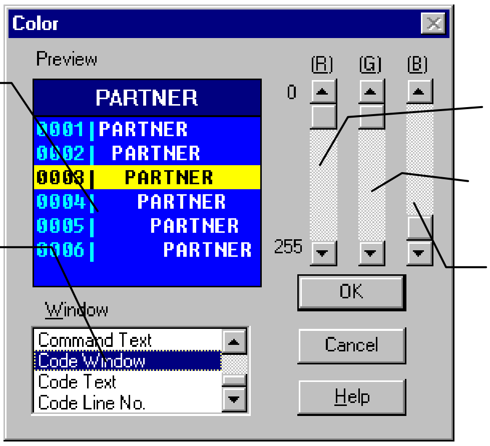
Moves selected button upward and downward.

Displays the explanation of selected button.

6.1.4 Color dialog box

[Setup]-[Select Color],

[Color] dialog box specifies the background and foreground color of each window.



Displays the window image.

Selects the window of which color to be changed.

Magnitude of red

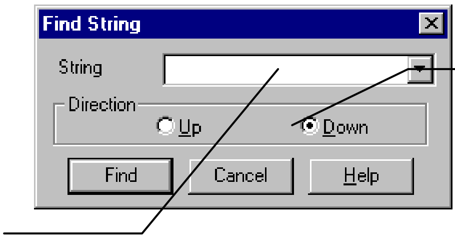
Magnitude of green

Magnitude of blue

6.1.5 Find String dialog box

[Search]-[Find],

[Find String] dialog box specifies the strings and direction to search in focused window.



Enter the string to be searched.

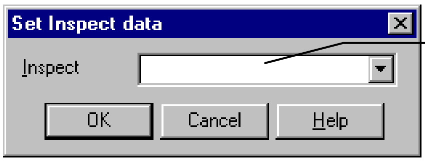
Specifies the direction to search.

[Notice] The default value of direction is [forward] in Command window, and [backward] in the others.

6.1.6 Set Inspect data dialog box

[Code]-[Inspect],[Local/Watch/Inspect]-[Inspect],

[Set Inspect data] dialog box specifies the variable name to be inspected.



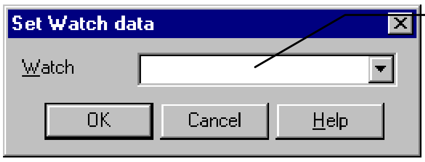
Enter the variable name to be inspected.

6.1.7 Set Watch data dialog box

[Code]-[Watch],[Inspect/Local]-[Watch],

[Set Watch data] dialog box specifies the variable to be registered to Watch.

Enter the variable name to be registered to Watch.



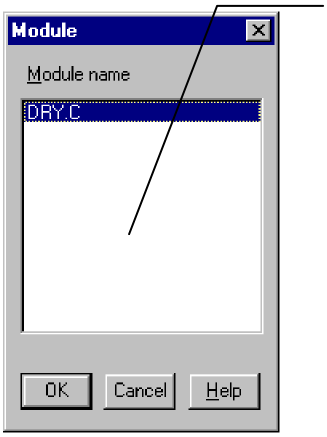
[Notice] As Watch registration in dialog box is limited only for C variables, instead, please use W command to register the contents of memory directly. Regarding W command, please refer to on-line help or Command Reference in "Part Edition".

6.1.8 Module dialog box

[File]-[Module],[Code]-[Module],

[Module] dialog box specifies the source file to be displayed in Code window.

In [Module name] list box, source file name defined in debug information of file loaded to be debugged and file name referred by V command will be displayed.

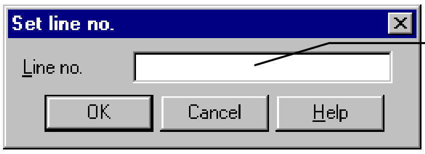


Displays the file name which can be selected.

6.1.9 Set line no. dialog box

[Code]-[Line no.]

[Set line no.] dialog box specifies the start line number to be displayed in Code widow.

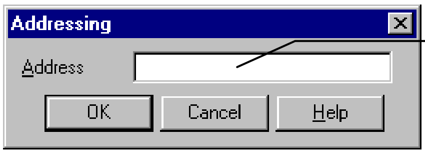


Enter the line number.

6.1.10 Addressing dialog box (code)

[Code]-[Address]

[Addressing (code)] dialog box specifies the start address of disassembled code listing to be displayed in Code window.



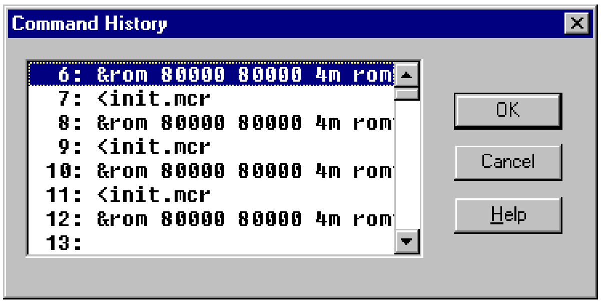
Enter the address.

6.1.11 Command History dialog box

[Command]-[History]

[Command History] dialog box selects and displays the command history that entered into Command window.

If any character string are entered into command input line (current prompt line) in Command window, the commands start with that strings will be displayed in list box.

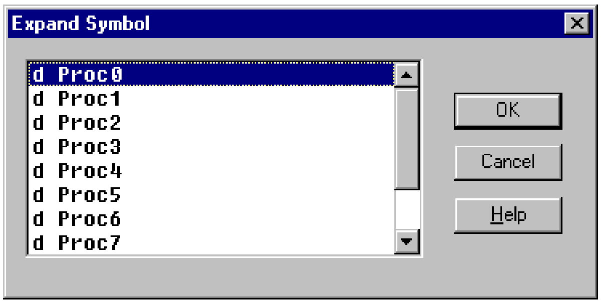


```
 6: &rom 80000 80000 4m rom
 7: <init.mcr
 8: &rom 80000 80000 4m rom
 9: <init.mcr
10: &rom 80000 80000 4m rom
11: <init.mcr
12: &rom 80000 80000 4m rom
13:
```

6.1.12 Expand Symbol dialog box

[Command]-[Exp.Sym]

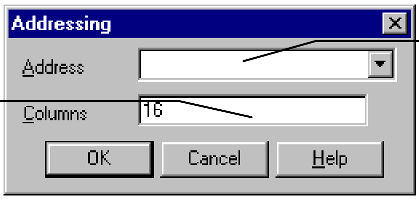
[Expand Symbol] dialog box displays the completion of global symbols, which start with last strings in command input line, in list box. (Example below is the case that "procX" is entered into command input line.)



6.1.13 Addressing dialog box (memory)

[Memory]-[Address Set]

[Addressing (memory)] dialog box specifies the start address of dump listing displayed in Memory window.



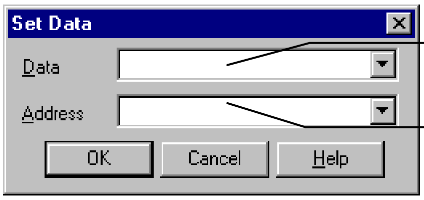
Specify the number of columns to be displayed per line.

Enter the address.

6.1.14 Set Data dialog box

[Memory]-[Edit Data]

[Set Data] dialog box changes the memory contents.



Enter the data value.

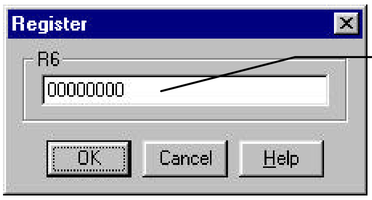
Enter the address.

[Notice] Clicking on <OK> button, the dialog box will prompt to change data of next address. To close this dialog box, click on <Cancel> button.

6.1.15 Register dialog box

[Register]-[Change]

[Register] dialog box changes the value of register.



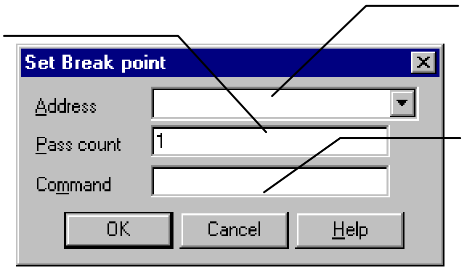
00000000

Enter the register value.

6.1.16 Set Break point dialog box

[Break]-[Add]

[Set Break point] dialog box sets break points.



Set Break point

Address

Pass count 1

Command

Enter the pass count.

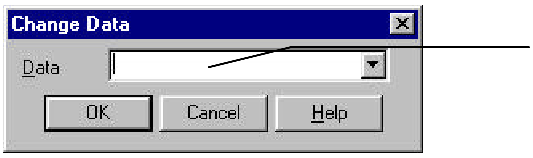
Enter the address.

Enter the command to be executed after program execution break.

6.1.17 Change Data dialog box

[Inspect]-[Change]

[Change Data] dialog box changes the value of variable currently inspected or element selected in Inspect window.



Change Data

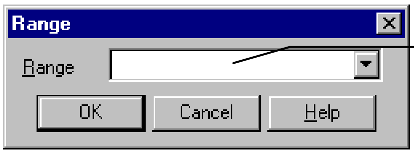
Data

Enter the value.

6.1.18 Range dialog box

[Inspect]-[Range]

[Range] dialog box changes the start number and the maximum element number to display an array or a pointer.

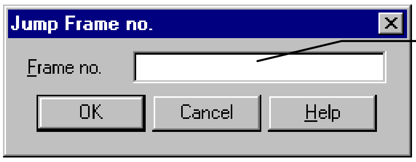


Enter the range into this edit box in the form of start number and maximum number.

6.1.19 Jump Frame no. dialog box

[History]-[Jump]

[Jump Frame no.] dialog box specifies the frame number to be displayed.



Specify the frame number to be displayed.

This page is intentionally left blank.

7 Data Expression

As an address or data notation, PARTNER supports various expressions, such as symbol and operation expression.

7.1 Address/Data Expression

As an address or data notation, PARTNER supports various expressions, such as symbol and operation expression.

7.1.1 Supported Symbol

PARTNER supports two types of symbol. The one is global symbol (which is valid for all region of program). And the other is local symbol such as local variable or static variable (which is valid only within a C function).

7.1.2 Global Symbol

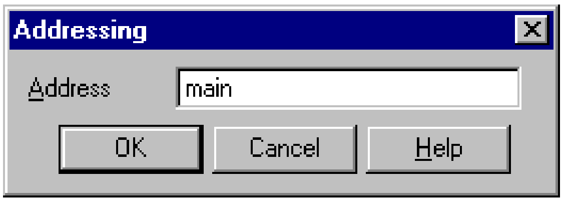
Global symbol can be specified instead of address value for label in disassembled code listing or in address input. C extern variable and function names are registered as the global symbols. The extern variable and function name in C program are registered as global symbol. Usually, C compiler employs the symbol name preceded or followed by under bar (_) as a variable or function name.

However, it will be very inconvenient, if global symbols always need to be specified with under bar. Thus in PARTNER, under bar at beginning or end of a global symbol can be omitted for users' convenience. And ignore case option for symbol name is also available. (refer to "Command Reference" in "Part Edition" and [Setup]-[Option Set])

However, if a symbol name is identical with CPU register name, the register name will have higher priority. Thus the symbol identical with register name cannot be referred.

Following example is the case that a global symbol is used in address specification.

```
>u _main   /* display disassembled code listing from symbol value _main */
>u main  /* the same as above */
```



Addressing

Address: main

Fig. 37 Specifying address with symbol value _main, in [Addressing] dialog box

Following example is the case that a global symbol is used in macro definition.

```
>i=5                    /* 1 */
symbol is registered
>while{ i!=0            /* 2 */
?T                      /* 3 */
?i=i-1                  /* 4 */
?}                      /* 5 */
>
```

(1) Assign 5 to symbol name i. (If i is undefined symbol, it will be registered here as a symbol.)

(2) Compare symbol value i with 0.

(3) If i is not equal to 0, executes Trace with T command.

(4) Assign i minus 1 to i.

(5)The end of WHILE{ command. This line cause to evaluate WHILE{ command condition.

If above macro is executed, it will execute T command 5 times, and then exit WHILE command loop.

In above example, the symbol i is treated the same as to a variable in high level programming language such as C or BASIC. The symbol name used as an variable should not conflict with already existing global or local symbol.

7.1.3 Local Symbol

The local symbol is the one of variable that is valid within a function such as automatic variables or function arguments, and static variable.

The local symbol is usually automatically registered at loading debug information.

The local symbol has address value and other information such as scope (valid range) and attribute (int, char, double, etc.).

7.1.4 Special Symbol

(1) __ERR__

The special symbol __ERR__ will be set to 1, if the last command executed occurs an error. And set to 0, if the command is normally executed.

__ERR__ can be used for error handling in macro commands.

(2) __RUN__

The special symbol __RUN__ will be set to 1 during user program execution, and set to 0 if user program breaks.

This symbol can be used for macro command processing such as waiting for user program break.

7.1.5 Available Numeric Value

PARTNER supports numeric value in binary/octal/decimal/hexadecimal digit, that are extinguished with the mark indicates radix base preceding to the digit. If the mark omitted, the radix specified with N command is assumed.

Notation	Radix Base
@ digit	binary
¥ digit	octal
_ digit	decimal
$ digit	hexadecimal
0x digit	hexadecimal
digit	according to radix specification (decimal, hexadecimal)

For a example, @11001010,312, _202, $CA, 0xCA all stand for the same value.

In addition to it, there are commands (DS, DL, DT, SS, SL, ST) to handle 4, 8 and 12 byte real number (IEEE format).

7.1.6 Address

If address is specified as command parameter, symbol name or line number is also available, instead of address value.

7.1.7 Line Number

PARTNER accepts the source level debugging with the line number of source files.

The line number is used to indicate a specific line in the source file. Debugging with line number is available, only if loaded debug information contains the line number information in it.

Following 3 types of input format are available for the line number.

```
format1   .[file_name:]line_number

format2   .+-line_number

format3   .symbol_name+-line_number
```

The line number designates a specific source line of user program, in the combination of a decimal number (line number) and file name or symbol name.

Format1 specifies an absolute line number. If file_name is specified, it stands for the specified number-th of line (line number) in the file. If the file name is omitted, the current file (the one displayed in the Code window) is assumed as a file name.

```
>u.100              /* specifies 100th line in current file */
>u.kmc:120          /* specifies 120th line in kmc.c */
```

Format2 specifies a relative position from the source line which current program counter indicates. With the form +line_number and -line_number, it specifies the number of lines after and before the current source lines, respectively.

If there is no source line corresponding to the current program counter, the command will fall to an input error.

```
>v.+10        /* specifies 10th line from the source line that program counter
              indicates */
```

Format3 specifies a relative position from the line number which specified symbol value (address) indicates. With the form +line_number and -line_number, it specifies the number of lines after and before the current source lines, respectively. If there is no source line corresponding to the specified symbol name, the command will fall to an input error.

```
>bp.main+10         /* specifies 10th line after the symbol main */
```

7.1.8 Character String

PARTNER accepts character string (sequence of ASCII characters) instead of numeric value. The character string is the characters quoted with single quotation (').

‘A’ = $41
‘AB’ = $4142
‘ABCD’ = $41424344

Especially, the Enter Input mode in E/EB command accepts character string up to 16 characters as input.

[Notice] If character string contains single quotation (') in it, the one up to the single quotation will be entered and the rest will be ignored. To avoid this, specify numerical value ($27) in place of single quotation.

7.1.9 Register Name

PARTNER can handle register contents as numeric value.

As register specification depend on the target CPU, please refer to Register Variables in "Part Edition".

7.1.10 Operation Expression

The operation expression is the combination of numeric value, symbol and registers by operator, and returns one value. PARTNER has a C-like numeric and logical operator. The operation expression is available everywhere in each command to specify a value (data or address). Followings are unary and binary operators available as an operator.

Unary Operator		
*		16bit data of specified address (pointer word)
+		unary plus
-		unary minus
~		NOT(complementary of 1)
!		logical negation
Binary Operator		
1	*	multiplication
1	/	division
1	%	module operation(modulo)
2	+	addition

(To be continued to next page)

2	-	subtraction
3	>>	right shift
3	<<	left shift
4	>=	comparison(if the left hand side is greater than or equal to the right hand side, returns 1, otherwise 0)
4	<=	comparison(if the left hand side is smaller than or equal to the right hand side, returns 1, otherwise 0)
4	>	comparison(if the left hand side is greater than the right hand side, returns 1, otherwise 0)
4	<	comparison(if the left hand side is smaller than the right hand side, returns 1, otherwise 0)
5	==	comparison(if the left hand side equals to the right hand side, returns 1, otherwise 0)
5	!=	comparison(if the left hand side does not equal to the right hand side, returns 1, otherwise 0)
6	&	AND
7	^	XOR
8	|	OR
9	&&	logical AND
10	||	logical OR
System Function		
VAL(C_expression)		evaluate as c_expression as C expression

The numeric value at the left hand of operators stand for priority of the operator. If the priority is equal among the operators in a expression, the expression will be valuated from left to right. However, quoting with paresis, priority of expression can be altered.

The comparison, logical AND, logical OR operators are used for condition control of macro (FOR, WHILE command) and condition command (IF command and others).

```
>h -(1+2*3)
           oct          dec           hex      asc        float
       37777777771              -7    FFFFFFF9   '....'    -6.805644e++38
>
```

7.2 Data Expression in C Language Level

Section "7.1 Address/Data Expression" explains how the expressions are processed (carried out operation) on global symbol, local symbol and line number, just as address or data value.

Though these expressions are available almost all of the dialog commands, it is not enough to handle the expressions in user program with the one described above, because they are written based on C language standard.

PARTNER newly provides dialog commands that can directly handle C expressions. More specifically, the commands related Inspect, Watch command, VAL command and ? command can handle C expression just the same as C syntax.

7.2.1 <Expression>

In the description in C language level and expression explained section "7.1 Address/Data Expression", sometimes the evaluation results might differ for the same expression. Followings example with C global variable abc explains these differences.

```
>d abc       /* display memory from address of variable abc */
00001000  00 01 02 03 .......
>d abc+10  /* display memory from address of variable abc + 10 */
00001010  AA BB CC DD .......
>? Abc       /* display value of variable abc (evaluate as C expression) */
(int ) 1 (0x1)
>? abc+10  /* display value of variable abc + 10 (evaluate as C expression) */
(int ) 11 (0xB)
>
```

This example shows that the same expression, such as abc or abc+10, changes its meaning depending on the context, ordinal expression (D command) and C expression (? command). The Inspect command, Watch command, VAL command and ? command evaluate abc as a variable in C language level. The other commands evaluate as an address of variable abc.

7.2.2 Variables

The variables or functions available for PARTNER in C expression, need to be declared in source file which is compiled with detail debug information option. And all register names can be used as pseudo variable, which has type (attribute) of unsigned int. Regarding available register variables, refer to the appendix of "Part Edition".

7.2.3 Variable Scope

In C language, a variable scope (visibility range) should be considered in programming or debugging. For examples, a variable declared as extern is valid for all program area, that is to say the scope is all program area. On the other hand, an automatic variable declared within a function, is valid only during the function is alive, and the scope is within the function.

If an extern variable and an automatic variable has identical name, only the automatic variable is valid within the function and the extern variable cannot be accessed. And an automatic variable in the function currently is not used, cannot be accessed, as it is not exists anywhere in memory space.

PARTNER automatically handles these issues by getting scope information from debug information.

7.2.4 Constants

The specifications of constants are equivalent to C syntax. And the default radix is always decimal number in spite of the setting of N command.

Notation	Radix Base
digit	decimal
0xdigit	hexadecimal
0Xdigi	hexadecimal
0digit	octal

For an example, all of 4096(decimal), 0x1000(hexadecimal) and 010000(octal) stand for the same value. The C escape sequences in character constant are also supported.

C character	value	meaning
‘¥a’	0x7	bell
‘¥b’	0x8	back space
‘¥f’	0xC	form feed
‘¥n’	0xA	line feed
‘¥r’	0xD	return
‘¥t’	0x9	horizontal tab
‘¥v’	0xB	vertical tab
‘¥¥’	0x5C	back slash
‘¥nnn’	Nnn	octal number (8bit)
‘¥xnn’	Nn	hexadecimal number (8bit)

7.2.5 Operators

Regarding operators, the one equivalent with C language is supported in the same syntax. However the operators other than = (assignment) are not available for floating point value in PARTNER.

The priority of each operator are as follows.

Priority	Operators
1	function(n) array[n] n.n n->n n++ n--
2	&n *n -n ~n !n ++n --n sizeof n #n
3	(cast)n
4	n%n n/n n*n
5	n+n n-n
6	n<<n n>>n
7	n>n n<n n>=n n<=n
8	n==n n!=n
9	n&n
10	n^n
11	n|n
12	n&&n
13	n||n
14	nn?nn:nn
15	n=n n*=n n/=n n%=n n+=n n-=n n<<=n n>>=n n&=n n^=n n|=n
16	n,n

If the priority is equal between adjacent operators, the expression will be evaluated from left to right, however, with the exception that the assignment operator (priority 16) is evaluated from right to left. Quoting with parentheses, priority of expression can be altered.

7.2.6 Expressions with Side Effects

The assignment operator such as ++, -- or = and function call have side effects

that might alter the variables of program to be debugged during evaluating the expression.

Though there might be some requirements to change data with assignment operator in expression, data change in such manner during debugging is considered to be rather rare case.

Thus PARTNER prohibits the use of operators with side effects in Watch, Inspect and ? command, so that avoid changing the data during evaluation of expression by mistake. Only the VAL command accepts the operators with side effects.

Pleas try to use ? command or Inspect command instead of VAL command for data reference only.

The VAL command is recommended, only if the operator with side effects is used for changing data.

And regarding function call, if global or static variables have been changed or some other data area has been changed via pointer, the user program execution might not work properly anymore. Thus, if function call is used in the VAL command, considerations are required.

```
>? abc=1234
operator with side effects cannot be used
>val abc=1234              /* assign 1234 to abc */
(int ) 1234 (0x4D2)
>val fnc(1,2,3)            /* call function fnc */
(int ) 10 (0xA)
>
```

Index

PARTNER Users Manual

"V800 Series Common Edition"

Revision 2.20 May 2000

Midas lab. Inc.

Copyright 2000 Midas lab. Inc./Kyoto Micro Computer Co.,Ltd.