

PARTNER ユーザーズ・マニュアル  
『ARMシリーズ共通編』

## ■ ユーザー登録について(重要)

- 同封していますユーザー登録書に必要な事項を漏れなくご記入の上、直ちにご返送ください。
- ユーザーサポート(技術的な問い合わせ、バージョンアップ等のお知らせ)は、このユーザー登録書に基づいて行います。

## ■ PARTNERのバージョンアップについて

- 最新のPARTNERは、以下のサイトよりダウンロードできます。

<http://www.midas.co.jp/products/download/program/partner.htm>

## ■ ご注意

- PARTNER(プログラム及びマニュアル)に関する著作権は株式会社マイダス・ラボ及び京都マイクロコンピュータ株式会社が所有します。
- 本プログラム及びマニュアルは著作権法で保護されており、弊社の文書による許可がない限り複製、転載、改変等できません。
- お客様に設定される使用権は、PARTNERをお客様が登録する1台のシステムにおいてのみ使用する権利を意味します。
- 本製品は、万全の注意を持って作製されていますが、ご利用になった結果については、販売会社、株式会社マイダス・ラボ及び京都マイクロコンピュータ株式会社は一切の責任を負いかねますのでご了承ください。
- 本プログラム及びマニュアルに記載されている事柄は、予告なく変更されることがあります。
- 本プログラムの変更により、このマニュアルに記載している画面のイメージと実際の画面が異なる場合があります。

## ■ 商標について

- MS-Windows、Windows、MS、MS-DOSは米国マイクロソフト・コーポレーションの商標です。
- そのほか本書で取り上げるプログラム名、システム名、CPU名などは、一般に各メーカーの商標です。

改訂履歴

日付 Y / M / D	Rev	内 容
2001.09.25	1.00	初版

空白ページ

# 目次

はじめに.....	1
<b>1 はじめてのPARTNER.....</b>	<b>5</b>
1.1 PARTNERの特徴.....	6
1.1.1 データ参照/変更およびリンク機能.....	6
1.1.2 ツールバー.....	6
1.1.3 ユーザによるカスタマイズ機能.....	6
1.2 PARTNERの基本的な操作手順.....	7
<b>2 セットアップ.....</b>	<b>11</b>
2.1 セットアップの前に.....	12
2.2 セットアップの方法.....	13
2.2.1 README.TXTを読む.....	13
2.2.2 PARTNERのインストール.....	13
2.2.3 PARTNERのアンインストール.....	13
<b>3 起動.....</b>	<b>15</b>
3.1 PARTNERの環境設定.....	16
3.1.1 RPTSETUPの使用法.....	16
3.2 関連ファイル.....	21
3.2.1 起動時に必要なファイル.....	21
3.2.2 PARTNER終了時に作成するファイル.....	22
3.3 コンフィギュレーションファイル(RPTARMxx.CFG).....	23
3.3.1 設定項目.....	23
3.4 初期設定コマンド.....	25
3.5 PARTNERの起動.....	26

3.5.1 PARTNERが正常に起動した場合 .....	26
3.5.2 PARTNERが起動しない場合 .....	27
<b>3.6 操作上の注意.....</b>	<b>28</b>
3.6.1 ウィンドウ操作上の注意.....	28
3.6.2 コマンド実行上の注意 .....	29
<b>4 ウィンドウコマンド.....</b>	<b>31</b>
<b>4.1 画面構成.....</b>	<b>32</b>
4.1.1 構成要素 .....	32
4.1.2 画面表示 .....	34
<b>4.2 メニュー .....</b>	<b>35</b>
4.2.1 ファイルメニュー .....	35
4.2.2 編集メニュー .....	36
4.2.3 検索メニュー .....	36
4.2.4 表示メニュー .....	37
4.2.5 実行メニュー .....	37
4.2.6 ローカルメニュー .....	38
4.2.7 ウィンドウメニュー.....	39
4.2.8 設定メニュー .....	40
4.2.9 ヘルプメニュー.....	41
<b>4.3 ショートカットキー.....</b>	<b>42</b>
4.3.1 各ウィンドウ共通のショートカットキー .....	42
4.3.2 ウィンドウ独自のショートカットキー .....	44
<b>4.4 マウス操作.....</b>	<b>45</b>
4.4.1 各ウィンドウ共通のマウス操作 .....	45
4.4.2 ウィンドウ独自のマウス操作.....	46
<b>4.5 ツールバー.....</b>	<b>47</b>
4.5.1 ツールバーの設定 .....	47
4.5.2 各ボタンの機能.....	48
<b>4.6 ウィンドウバー.....</b>	<b>50</b>
<b>4.7 ステータスバー.....</b>	<b>51</b>

<b>5</b>	<b>チャイルドウィンドウ.....</b>	<b>53</b>
5.1	コードウィンドウ.....	54
5.1.1	コードウィンドウのショートカットキー .....	55
5.1.2	コードウィンドウのローカルメニュー .....	57
5.1.3	コードウィンドウでのマウス操作 .....	58
5.2	コマンドウィンドウ.....	59
5.2.1	コマンドウィンドウのショートカットキー .....	59
5.2.2	コマンドウィンドウのローカルメニュー .....	61
5.3	メモリウィンドウ.....	62
5.3.1	メモリウィンドウのショートカットキー .....	62
5.3.2	メモリウィンドウのローカルメニュー .....	63
5.3.3	メモリウィンドウでのマウス操作 .....	63
5.4	レジスタウィンドウ.....	64
5.5	スタックウィンドウ.....	65
5.5.1	スタックウィンドウのショートカットキー .....	65
5.5.2	スタックウィンドウのローカルメニュー .....	65
5.6	ローカルウィンドウ.....	66
5.6.1	ローカルウィンドウのショートカットキー .....	66
5.6.2	ローカルウィンドウのローカルメニュー .....	67
5.6.3	ローカルウィンドウでのマウス操作 .....	67
5.7	バックトレースウィンドウ.....	68
5.7.1	バックトレースウィンドウのショートカットキー .....	68
5.7.2	バックトレースウィンドウのローカルメニュー .....	69
5.7.3	バックトレースウィンドウでのマウス操作 .....	69
5.8	ウォッチウィンドウ.....	70
5.8.1	ウォッチウィンドウのショートカットキー .....	70
5.8.2	ウォッチウィンドウのローカルメニュー .....	71
5.8.3	ウォッチウィンドウでのマウス操作 .....	71
5.9	ブレークウィンドウ.....	72

5.10	メモウィンドウ.....	73
5.10.1	メモウィンドウのショートカットキー .....	73
5.10.2	メモウィンドウのローカルメニュー .....	74
5.11	インスペクトウィンドウ.....	75
5.11.1	インスペクトウィンドウのショートカットキー .....	75
5.11.2	インスペクトウィンドウのローカルメニュー.....	76
5.11.3	インスペクトウィンドウでのマウス操作 .....	76
5.12	ヒストリウィンドウ.....	77
5.13	I/Oウィンドウ.....	78
<b>6 ダイアログコマンド.....</b>		<b>79</b>
6.1	ダイアログボックス.....	80
6.1.1	ファイルを開くダイアログボックス .....	80
6.1.2	フォントの指定ダイアログボックス .....	81
6.1.3	ツールバーの設定ダイアログボックス.....	81
6.1.4	色指定ダイアログボックス .....	82
6.1.5	文字列検索ダイアログボックス .....	82
6.1.6	インスペクト設定ダイアログボックス.....	83
6.1.7	ウォッチ設定ダイアログボックス.....	83
6.1.8	モジュールダイアログボックス .....	84
6.1.9	行番号指定ダイアログボックス .....	84
6.1.10	アドレス指定ダイアログボックス(コード).....	85
6.1.11	コマンド履歴ダイアログボックス.....	85
6.1.12	シンボル拡張ダイアログボックス.....	86
6.1.13	アドレス指定ダイアログボックス(メモリ).....	86
6.1.14	データ設定ダイアログボックス .....	87
6.1.15	レジスタの設定ダイアログボックス .....	87
6.1.16	ブレークポイントの設定ダイアログボックス.....	88
6.1.17	データ変更ダイアログボックス .....	88
6.1.18	範囲指定ダイアログボックス .....	89
6.1.19	フレーム番号指定ダイアログボックス.....	89
<b>7 データ表現.....</b>		<b>91</b>



---

<b>7.1 アドレスデータの表現</b> .....	<b>92</b>
7.1.1 扱えるシンボル.....	92
7.1.2 グローバルシンボル.....	92
7.1.3 ローカルシンボル.....	93
7.1.4 特殊シンボル.....	94
7.1.5 扱える数値.....	94
7.1.6 アドレス.....	94
7.1.7 行番号.....	95
7.1.8 文字列(ストリング).....	96
7.1.9 レジスタ名.....	96
7.1.10 演算式.....	96
<b>7.2 C言語レベルでのデータ表現</b> .....	<b>98</b>
7.2.1 <式>.....	98
7.2.2 変数.....	99
7.2.3 変数スコープ.....	99
7.2.4 定数.....	100
7.2.5 演算子.....	101
7.2.6 副作用のある式.....	102

空白ページ

## はじめに

このマニュアルでは、PARTNERの導入方法、起動手順、操作方法などARMシリーズの共通部分について説明します。CPUおよび利用形態に依存する操作方法については、『個別編』を参照してください。

なお、ダイアログコマンドの詳細については、オンラインヘルプを参照してください。また、CPUおよび利用形態で本マニュアルに記載された画面を構成する要素が異なる場合がありますのでご了承ください。

### PARTNERの概要

---

PARTNERは、京都マイクロコンピュータ株式会社が開発したPARTNER-Winをマイダス・ラボ製品用に移植したWindows版ソースレベルデバッガです。

ソースレベルデバッガとしての基本機能であるプログラムロード、実行、ブレークポイント管理、データ表示/変更、コード表示/変更、Cのデータ表示/変更やマイダス・ラボ製品用にカスタマイズされた機能があります。

### パッケージの内容

---

PARTNERのパッケージには、セットアップCD-ROMとマニュアルおよびユーザー登録書が入っています。

### 動作環境

---

セットアップを始める前に、PARTNERを使用する上でパッケージ以外に必要なものを説明します。

セットアップを実施する機器のハードウェアおよびソフトウェアの環境を確認してください。

## ■ 必要なハードウェア

- ホストコンピュータ

P5-90以上のCPUを搭載しWindows95/98/NT/2000が動作可能なコンピュータが必要です。

- ハードディスク

PARTNERを使用するには、ハードディスクが必要です。PARTNERをセットアップするには、使用するハードディスクに10MB以上の空き領域が必要です。

- CD-ROMドライブ

PARTNERをセットアップするために、CD-ROMドライブが必要です。

- その他

利用形態によって異なります。詳細は『個別編』を参照してください。

## ■ 必要なソフトウェア

- Windows

Windows95/98/NT/2000(基本OS)が必要です。PARTNERのセットアップを始める前に、Windowsが起動するように、ハードディスクにセットアップしておいてください。

- プログラム開発ツール

PARTNERがサポートする言語処理系のプログラムを作成するツールが必要です。

- その他

利用形態によって異なります。詳細は『個別編』を参照してください。

## マニュアルについて

---

『PARTNER ユーザーズ・マニュアル』は、CPUや利用形態に依存しない『共通編』と依存する『個別編』の二部で構成されます。これらのマニュアルは、次の表記規則に従って記述されています。

- |               |  |
|---------------|--|
| [メニュー]        | メニュー名は、角括弧([ ])で囲んで表記します。                          |
| [メニュー]-[コマンド] | コマンド名は、そのコマンドが含まれているメニューの名前を付けて、角括弧([ ])で囲んで表記します。 |
| [ダイアログ]       | ダイアログボックス名は、角括弧([ ])で囲んで表記します。                     |
| <ボタン>         | 各種ボタン名は、山括弧(<>)で囲んで表記します。                          |

## オンラインヘルプについて

---

PARTNERには、画面上で機能や操作方法を説明するオンラインヘルプが用意されています。

オンラインヘルプを表示するには、HELPキー(DOS/VではEndキー)、HELPコマンド、[ヘルプ]-[目次]コマンドもしくは<ヘルプ>ボタンで行います。

空白ページ

# 1 はじめてのPARTNER

PARTNERを使用するに当たり、PARTNERの特徴とデバッグの手順を簡単に説明します。

## 1.1 PARTNERの特徴

PARTNERは、MDI(Multi Document Interface)を採用し、各チャイルドウィンドウにデバッグ時に必要な様々な情報を表示します。

また、ショートカットキー、メニュー、ツールバー、ステータスバー、ダイアログボックスを採用し、操作環境の向上を図るために様々な機能が用意されています。

### 1.1.1 データ参照/変更およびリンク機能

コードウィンドウに表示されている変数にマウスカーソルを移動してダブルクリックするだけでインスペクトウィンドウが開き、簡単に変数のデータ参照/変更ができます。

また、ウォッチウィンドウ、ローカルウィンドウでも同様に変数のデータ参照/変更ができます。

さらに、マウス操作でレジスタウィンドウ、メモリウィンドウでもデータの変更ができます。

### 1.1.2 ツールバー

デバッグ対象プログラムのロード、ブレイクポイントの設定、プログラムの実行、変数のインスペクトなど、利用頻度の高い機能がツールバーに登録でき、ツールバーのボタンをマウスでクリックするだけで簡単に実行することができます。

### 1.1.3 ユーザによるカスタマイズ機能

ツールバー、表示フォント、表示色、ウィンドウレイアウト等をユーザで自由にカスタマイズできます。

ウィンドウレイアウトは、使用状況に応じて3種類まで登録できます。



## 1.2 PARTNERの基本的な操作手順

PARTNERの基本的な操作手順を以下に示します。

### セットアップ

PARTNERをハードディスクにセットアップ(インストール)します。

【参照】『2 セットアップ』(10頁)

### 環境設定

PARTNERを起動するために必要な環境設定を行います。

【参照】『3.1 PARTNERの環境設定16』(16頁)

環境設定には、PARTNER環境設定用プログラム(RPTSETUP)で行います。(図 1)



図 1 環境設定プログラム(RPTSETUP)

### 起動

環境設定が完了した後、PARTNERを起動します。(図 2)

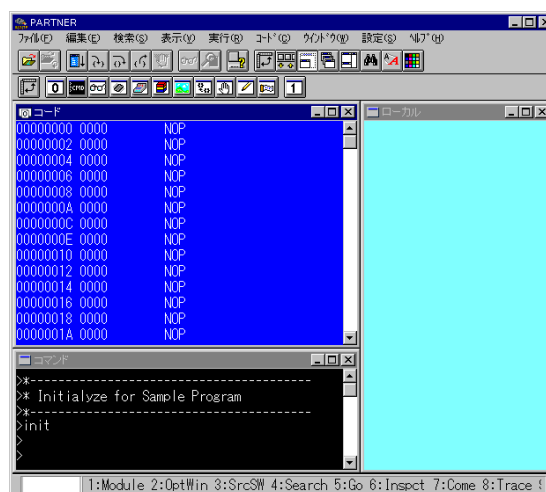



図 2 起動画面

## プログラムのロード

デバッグ対象のプログラムをロードします。(図 3)

【参照】  ボタン(48頁),Lコマンド,[ファイル]-[ロード]コマンド(35頁)

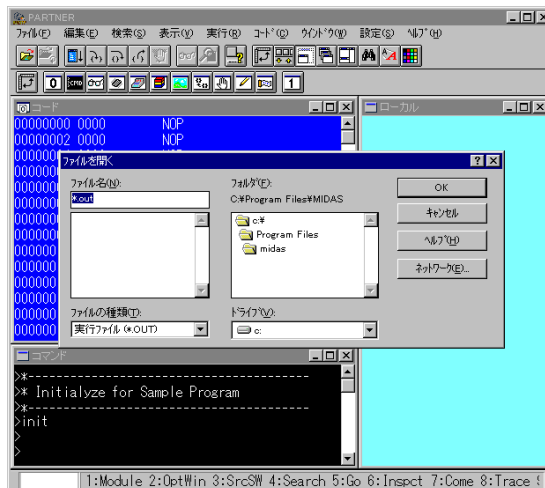


図 3 デバッグ対象ファイルのロード

## ブレークポイントの設定

コードウィンドウの行番号部分、アドレス部分をマウスでクリックしてブレークポイントを設定します。(図 4)

【参照】 BPコマンド 『5.1.3 コードウィンドウでのマウス操作』(58頁),[ブレーク]-[ブレーク設定](『個別編』のウィンドウコマンド)

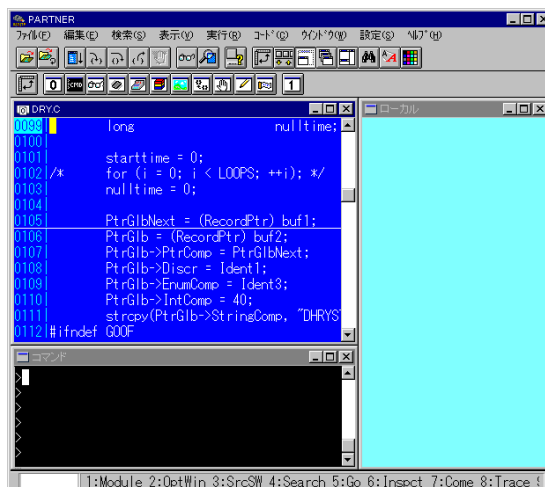




図 4 ブレークポイントの設定


## プログラムの実行

プログラムの実行を開始し、 で設定したブレークポイントで停止するか、ESCキーで強制的に停止します。

【参照】  ボタン(48頁),Gコマンド,[実行]-[プログラム実行] (37頁),  ボタン(48頁), ESCコマンド

## 変数のインスペクト

参照(変更)したい変数をマウスでダブルクリックしてインスペクトウィンドウに表示します。(図 5)

【参照】  ボタン(48頁), 『5.1.1 コードウィンドウのショートカットキー』(55頁),INSコマンド

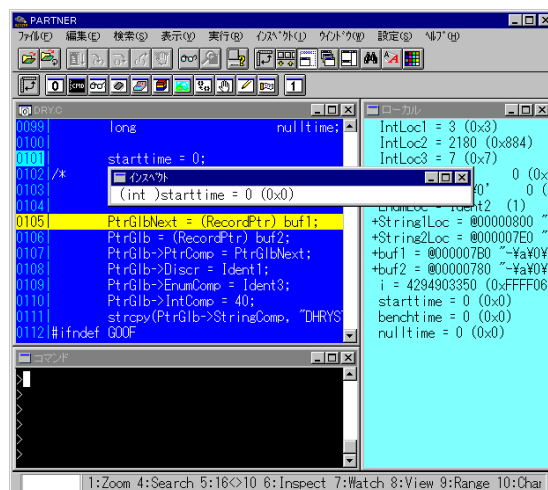



図 5 インスペクト表示

### 変数のウォッチ登録

常時参照したい変数をウォッチウィンドウに登録します。(図 6)

【参照】  ボタン(48頁), 『5.1.1 コードウィンドウのショートカットキー』(55頁), Wコマンド

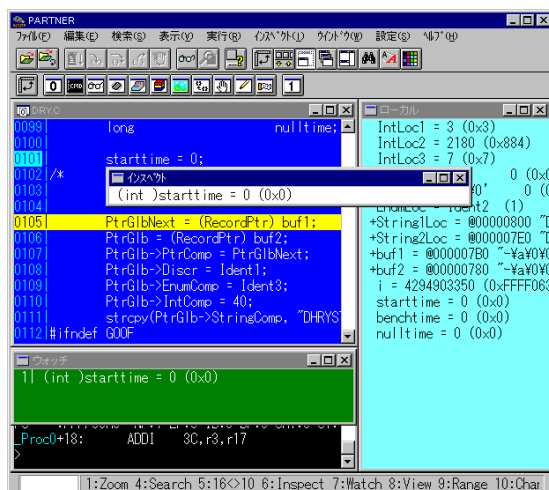



図 6 ウォッチ登録

### PARTNERの終了

PARTNERを終了します。(図 7)

【参照】  ボタン(48頁), [ファイル]-[終了](35頁), Q/EXITコマンド

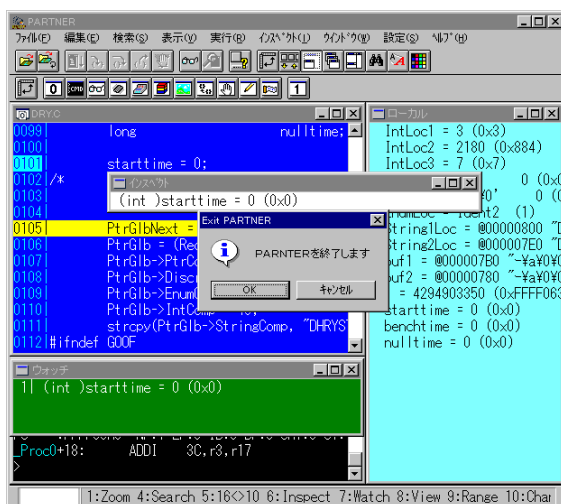


図 7 終了確認

## 2 セットアップ

PARTNERを使用するには、ホストコンピュータへのセットアップが必要です。この章では、PARTNERのインストールの方法が記述されています。

## 2.1 セットアップの前に

PARTNERのセットアップを始める前に、PARTNERに付属するハードウェアおよび、ソフトウェアのインストールをそれぞれのマニュアルにしたがって実施してください。

## 2.2 セットアップの方法

Windows95/98/NT/2000を起動してPARTNERのセットアップを行います。

**【注意】** PARTNERの各ファイルは、圧縮された状態でCD-ROMに格納されています。したがって、各ファイルを単純にハードディスクにコピーしただけでは、PARTNERを使用することができません。必ず、CD-ROMに納められているPARTNERのセットアッププログラムを使用して、適切なセットアップ作業を行ってください。

### 2.2.1 README.TXTを読む

PARTNERのセットアッププログラムを起動する前に、使用上の注意などについて書かれているREADME.TXTファイルをお読みください。

### 2.2.2 PARTNERのインストール

PARTNERをインストールするには、セットアップCD-ROMに入っているセットアッププログラム「SETUP.EXE」を起動します。

インストールの手順については、インストールプログラムに従ってください。

インストールが終了すると[スタート]-[プログラム]に「PARTNERフォルダ」が作成され、「RPTSETUP」というショートカットが作成されます。

**【注意】** PARTNERのセットアップ作業が終了しましたら、セットアップに使用したセットアップCD-ROMは大切に保管してください。

### 2.2.3 PARTNERのアンインストール

コントロールパネルの「アプリケーションの追加と削除」アイコンで、「PARTNER」を一覧から選んで削除します。

インストール直後の状態と内容が異なる場合は、完全に削除できませんので注意してください。

空白ページ



## 3 起動

PARTNERを起動するには、利用形態に合わせてPARTNERの環境設定が必要です。

スタートメニューの「RPTSETUP」をダブルクリックして起動し、PARTNERの環境を設定します。

## 3.1 PARTNERの環境設定

PARTNERを起動するに当たり、プロジェクト(デバッグ単位)に次の設定が必要です。

- 新規プロジェクトの作成(17頁)
- コンフィギュレーションファイルの設定(17頁)
- 起動オプションの設定(20頁)
- スタートメニューへの登録(19頁)

上記の設定を正常に行うことによってPARTNERを起動することができます。  
「RPTSETUP」は上記の設定を行うためのツールです。

### 3.1.1 RPTSETUPの使用方法

「RPTSETUP」は、プロジェクトの環境を設定するツールです。〈新規〉または、〈開く〉を選択すると環境設定用のボタンの使用が可能になります。

スタートメニューの「RPTSETUP」を起動し、プロジェクトの作成と設定(変更)を行います。(図 8)



図 8 RPTSETUP起動画面

メニューの〈編集〉および〈作成〉は、RPTSETUPでは選択できません。これらの機能は、将来の拡張のために用意されています。

## 新規にプロジェクトを作成する場合

新規にプロジェクトを作成する場合、<新規>のボタンを選択します。



## 新規プロジェクトの作成

PARTNERのプロジェクトを作成します。通常、「Projects」ディレクトリに作成します。

新規に作成したプロジェクトには、デフォルトのプロジェクトファイル(RPTARMxx.KPJ)とコンフィギュレーションファイル(RPTARMxx.CFG)が作成(Binディレクトリからコピー)されます。以後、これらのファイルに対してターゲットシステムのためのカスタマイズを行います。

プロジェクトを新規に作成するディレクトリは、「Projects」ディレクトリの他に、ユーザのソースプログラムなどが置かれているワークディレクトリを指定することもできます。この場合には、ユーザのワークディレクトリにコンフィギュレーションファイル(RPTARMxx.CFG)とプロジェクトファイル(RPTARMxx.KPJ)が作成されます(図 9)。

プロジェクトディレクトリ  
参照ボタン

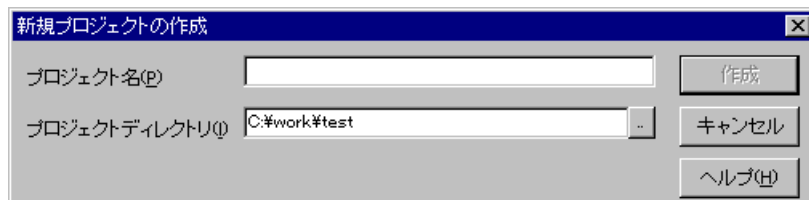


図 9 c:\work\testディレクトリを指定

### 既存のプロジェクトを変更

プロジェクトの設定内容を変更するには、<開く>のボタンを選択します。

### プロジェクトを開く



変更するプロジェクトファイル(RPTARMxx.KPJ)を選択して開きます。



図 10 プロジェクトの選択ダイアログボックス

## プロジェクト環境の設定(変更)

新規作成時または、既存のプロジェクトのプロジェクト環境を設定(変更)することができます。(図 11)

- 起動オプションの設定
- コンフィギュレーションファイルの設定
- スタートメニューへの登録



図 11プロジェクト設定(変更)画面

## 起動オプションの設定



PARTNERの起動オプションを指定します。起動オプションでは、PARTNERの動作環境を設定(変更)します。(図 12)



図 12 [起動オプション]ダイアログボックス

ダイアログボックスは、CPUおよび利用形態により異なります。詳細は『個別編』を参照してください。



## コンフィギュレーションの編集

使用するターゲットに合わせて作業ディレクトリにあるコンフィギュレーションファイルを編集します。

コンフィギュレーション項目については『3.3 コンフィギュレーションファイル (RPTARMxx.CFG)』(23頁)を参照してください。



## スタートメニューへの登録

設定したプロジェクトをスタートメニューに登録します (図 13)。登録されたショートカットを起動すると、設定したプロジェクトの内容でPARTNERが起動します。



図 13 ショートカット名"test"でスタートメニューに登録

## 3.2 関連ファイル

PARTNERが使用するファイルは以下の通りです。

### 3.2.1 起動時に必要なファイル

PARTNERの起動に必要なファイルには、コンフィギュレーションファイル、プロジェクトファイル、自動実行ファイルやPARTNERの終了時に作成するファイルがあります。これらのファイルは、プロジェクトディレクトリから読み込まれます。

自動実行ファイルやPARTNERの終了時に作成するファイル(3.2.2 PARTNER終了時に作成するファイル(22頁)参照)は、起動時に存在すれば読み込みます。

#### コンフィギュレーションファイル(RPTARMxx.CFG)

コンフィギュレーションファイルは、ターゲットの環境をPARTNERに通知するファイルです。このファイルには、ターゲットのメモリマップ、メモリマップに影響するレジスタの初期値等の設定が保存されています。

#### プロジェクトファイル(RPTARMxx.KPJ)

プロジェクトファイルは、PARTNERの環境を設定するファイルです。このファイルには、ウィンドウの配置、フォント、色の指定、ツールバーの設定、起動オプション等の設定が保存されています。

#### 自動実行ファイル(INIT.MCR)

自動実行ファイルは、PARTNERが起動された時、自動的に読み込まれて実行されます。これは、MS-DOSのAUTOEXEC.BATファイルに相当します。プログラムのデバッグに必要な前処理(ユーザプログラムのロードなど)をこのファイルに記述しておけば、PARTNER起動時に自動的にそれらの処理を行うことができます。

### 3.2.2 PARTNER終了時に作成するファイル

PARTNERは、終了時にいくつかのファイルを作成します。これらのファイルは、プロジェクトディレクトリに保存されます。

#### メモウインドウの内容

メモウインドウの内容を「memo.pt」ファイルに保存します。

#### コマンドヒストリ

コマンドヒストリの内容を「xxxx.dat」ファイルに保存します。



## 3.3 コンフィギュレーションファイル(RPTARMxx.CFG)

PARTNERは、起動時にコンフィギュレーションファイル(RPTARMxx.CFG)を読み込みます。コンフィギュレーションファイルには、ターゲットの動作環境を予め設定しておく必要があります。

コンフィギュレーションファイルは、テキスト形式のファイルです。セミコロン(;)で始まる行は、コメントとみなされます。

**【注意】**環境ファイルの変更の際は、記述順序や内部フォーマットを変更しないようご注意ください。

### 3.3.1 設定項目

コンフィギュレーションファイルで設定できる項目は次の通りです。

- プログラムカウンタ(PC)の初期値
- スタックポインタ(SP)の初期値
- コマンドでアクセス可能なメモリ領域

#### プログラムカウンタの初期値(INIT\_PC)

PARTNERが起動時にコードウィンドウでの不正なアクセスを防ぐため、プログラムカウンタ(PC)を初期化します。

書式：           INIT\_PC           アドレス

省略された場合は、ターゲットが持つデフォルト値になります。

#### スタックポインタの初期値(INIT\_SP)

PARTNERが起動時にスタックウィンドウでの不正なアクセスを防ぐため、スタックポインタ(SP)を初期化します。

書式：           INIT\_SP           アドレス

省略された場合は、ターゲットが持つデフォルト値になります。

アクセス可能なメモリ領域(MAP)
-------------------

PARTNERのコマンドや内部処理によりアクセス可能なメモリ領域を16進数で設定します。

書式：           MAP                   開始アドレス,終了アドレス

指定されたメモリ空間をPARTNERのコマンドでアクセス可能であることを宣言します。省略された場合は、0000-0000 ~ FFFF-FFFFの領域がアクセス可能と判断します。

MAP範囲の指定は、最大20領域まで設定できます。複数の範囲を指定する場合は、このフィールドを昇順並びで複数行記述します。

## 3.4 初期設定コマンド

PARTNERの動作環境を予め設定しておく必要がある場合は、コマンドで初期化することができます。

詳細は、『個別編』のコマンドリファレンス、または、ハードウェア・ユーザーズ・マニュアルを参照してください。

## 3.5 PARTNERの起動

PARTNERを起動するには、RPTSETUPでプログラムマネージャに登録したアイコンをダブルクリックします。

### 3.5.1 PARTNERが正常に起動した場合

PARTNERを起動すると図 14のような画面が表示されます。

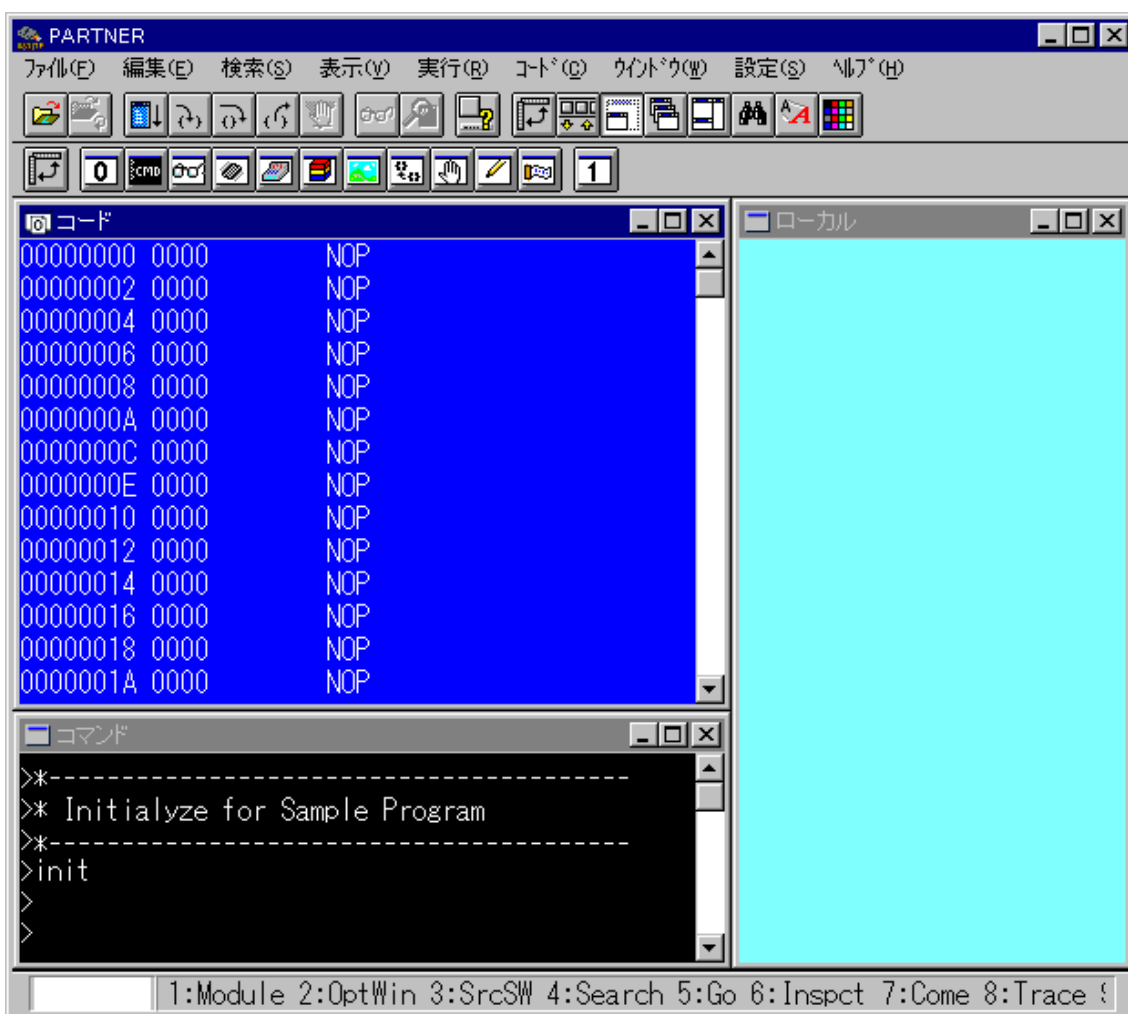


図 14 PARTNERの起動画面

### 3.5.2 PARTNERが起動しない場合

起動時にエラーが発生した場合は、メッセージボックスが表示されます。詳細に関しては、『個別編』の起動エラーメッセージを参照してください。

## 3.6 操作上の注意

PARTNERを快適に使用する上で操作上の注意点を説明します。

### 3.6.1 ウィンドウ操作上の注意

ウィンドウ操作に関して、以下に示す操作を行った場合に通信異常やモニタ・タイムアウトあるいはPARTNERの動作速度が著しく低下するなどの障害が発生する可能性があります。





メモリを参照するようなウィンドウを設定した状態でPARTNERを起動した場合





メモリを参照するようなウィンドウを閉じないでハードウェアをリセットした場合

このような障害は、バスコントローラやメモリコントローラなどの初期設定が必要なターゲットに依存します。初期設定の完了前にウィンドウに表示する情報をアドレス値やレジスタ値を基に取得するためのメモリアクセスが発生の要因となります。

したがって、メモリのアクセスに初期設定が必要なターゲットを使用する場合は、このような障害が発生する可能性があることを理解してください。

メモリを参照するウィンドウを以下に示します。

ボタン	ウィンドウ名称	基準値	備考
	コードウィンドウ	アドレス値	命令コードを取得するため、表示されたアドレス値を基に、メモリをアクセスします。
	コードウィンドウ1	アドレス値	命令コードを取得するため、表示されたアドレス値を基に、メモリをアクセスします。
	ウォッチウィンドウ	アドレス値	データを取得するため、登録されたシンボル値を基に、メモリをアクセスします。
	メモリウィンドウ	アドレス値	データを取得するため、表示されたアドレス値を基に、メモリをアクセスします。

	レジスタウィンドウ	レジスタ値	データを取得するため、各レジスタ値を基に、メモリをアクセスします。  メモリ表示が選択されている場合
	スタックウィンドウ	レジスタ値	データを取得するため、スタックポインタの値を基に、メモリをアクセスします。
	ローカルウィンドウ	アドレス値	データを取得するため、登録されたシンボル値を基に、メモリをアクセスします。
	履歴ウィンドウ	アドレス値	命令コードを取得するため、トレース結果として登録されたアドレス値を基に、メモリをアクセスします。

上記のウィンドウに関する機能は、『5 チャイルドウィンドウ(47頁以降)』を参照してください。

このような障害の発生を抑制したい場合は、PARTNERの終了、およびハードウェアをリセットする前に、メモリを参照するようなウィンドウを閉じる操作を行ってください。

### 3.6.2 コマンド実行上の注意

メモリを参照するようなダイアログコマンドをコマンドウィンドウから実行した場合にも、ウィンドウ操作時と同様な障害が、直接的または間接的に発生する可能性があります。

#### 直接的要因

メモリ表示 / 変更コマンド(D, E, F, S...)による操作

#### 間接的要因

コマンドのアドレスやデータを指定するパラメータに、メモリを参照するような<式>を指定した操作

コマンドのパラメータに記述する<式>に関しては、『7 データ表現 (91頁以降)』を、コマンドウィンドウで実行できるコマンドに関しては、『個別編』の「コマンドリファレンス」をそれぞれ参照してください。

したがって、メモリのアクセスに初期設定が必要なターゲットを使用する場合は、このような障害が発生する可能性があることを理解してください。

このような障害の発生を抑制したい場合は、必ず初期設定の完了後にメモリを参照するようなコマンドを実行してください。



## 4 ウィンドウコマンド

この章では、PARTNERのデバッグ機能の特徴づけるメニュー、ツールバー、ステータスバー、チャイルドウィンドウ等の基本的な操作について説明します。

## 4.1 画面構成

PARTNERの主な画面の構成要素を説明します。

### 4.1.1 構成要素

ウィンドウ要素を図 15で表します。

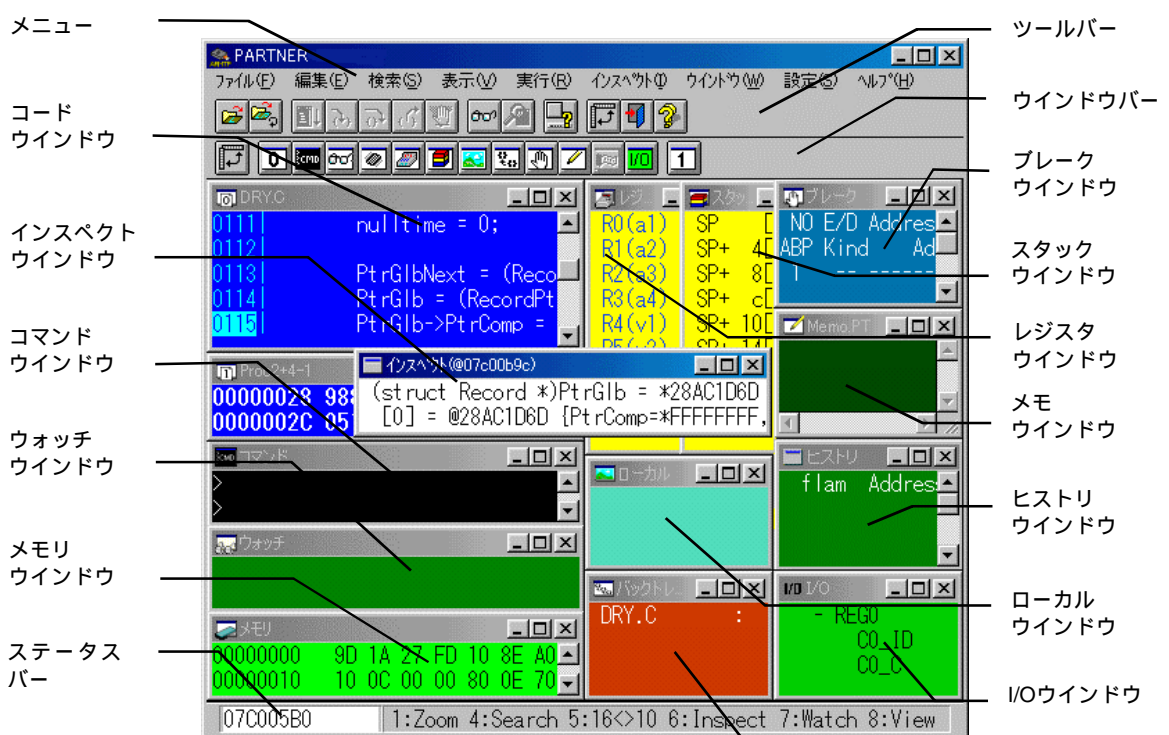


図 15 画面要素

バックトレースウィンドウ

### コードウィンドウ



ソース表示、逆アセンブル表示をするチャイルドウィンドウです。プログラムカウンタはリバース表示、ブレークポイントは下線表示されます。このウィンドウは2つあります。

---

## コマンドウィンドウ



ダイアログコマンドを入力、実行結果の表示やシスコール機能でのコンソール入出力文字の表示などを行うチャイルドウィンドウです。

---

## メモリウィンドウ



指定されるメモリ空間のダンプ表示をするチャイルドウィンドウです。CPU停止時に最新のデータに更新されます。

---

## レジスタウィンドウ



CPUのレジスタを表示するチャイルドウィンドウです。CPU停止時に最新のデータに更新されます。

---

## スタックウィンドウ



スタックの内容を表示するチャイルドウィンドウです。CPU停止時に最新のデータに更新されます。

---

## ローカルウィンドウ



プログラムカウンタに該当する関数内に定義されたローカル変数の内容を表示します。CPU停止時に最新のデータに変更されます。

---

## ウォッチウィンドウ



ウォッチ登録データをするチャイルドウィンドウです。CPU停止時に最新のデータに更新されます。

---

## バックトレースウィンドウ



関数のバックトレース表示をするチャイルドウィンドウです。CPU停止時に最新のデータに更新されます。

---

## ブレイクウィンドウ



設定されているブレイクポイントの表示をするチャイルドウィンドウです。

---

## メモウィンドウ



デバッグ中に使用する簡易エディタのするチャイルドウィンドウです。PARTNER終了時に内容を保存します。

## インスペクトウィンドウ

変数の内容をインスペクト(調査)し表示をするチャイルドウィンドウです。

## ヒストリウィンドウ



リアルタイムトレースメモリをダンプまたは、逆アセンブル表示をするチャイルドウィンドウです。

## I/Oウィンドウ



システムI/Oレジスタの内容を表示するチャイルドウィンドウです。

### 4.1.2 画面表示

ウィンドウに表示されるデータは、基本的に横スクロールして非表示部分を表示することができません。したがって、表示されていない部分を表示させるにはウィンドウの大きさ、フォントの大きさ等の変更が必要です。

横スクロール可能なウィンドウは、コードウィンドウ、メモリウィンドウ、メモウィンドウです。

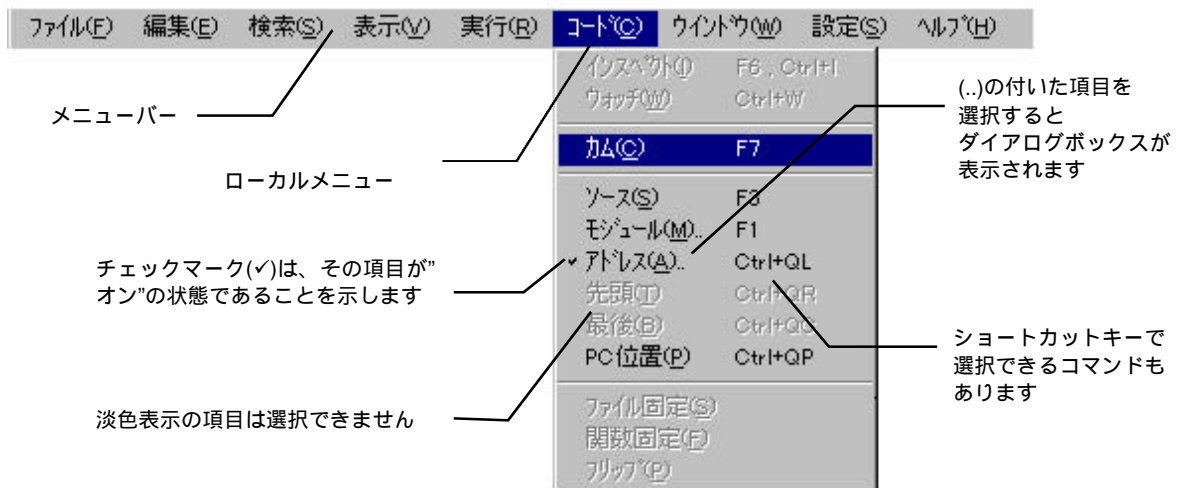
また、文字列検索結果が非表示部分にある場合は3つの不等号記号(>)が表示された行にあることを表示例として(図 16)に示します。検索結果を表示するには、ウィンドウの大きさ、フォントの大きさを変更し、[検索]-[次を検索]メニューを実行します。

```
ウォッチ
1|-(struct Record *)PtrG1b = *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 PROGRAM"
```

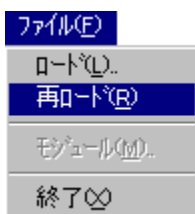
図 16 非表示部分に検索文字列がある場合

## 4.2 メニュー

メニューにはPARTNERのコマンドがグループ化して収められています。コマンドの中には、ただちに処理を実行するものと、ダイアログボックスを表示して追加のオプションを選択するものがあります。



### 4.2.1 ファイルメニュー



[ファイル]メニューにはファイル関連のコマンドが収められています。

[モジュール]は、コードウィンドウにフォーカスがないと実行できません。

メニュー項目	機能
ロード(L)	ユーザプログラムのファイルを選択するダイアログボックスを表示します。 (『6.1.1 ファイルを開くダイアログボックス』80頁参照)
再ロード(R)	以前にロードしたユーザプログラムの再ロードをします。
モジュール(M)	ロードしたユーザプログラムに含まれるモジュールを選択するダイアログボックスを表示します。 (『6.1.8 モジュールダイアログボックス』84頁参照)
終了(X)	PARTNERを終了します。

## 4.2.2 編集メニュー

### 編集(E)

貼り付け(P)	Shift+Ins
ページ複写(C)	Ctrl+Ins

[編集]メニューにはウィンドウ間でクリップボードを経由して文字列のやりとりを行うコマンドが収められています。

### 編集(E)

元に戻す(U)	Alt+BkSp
切り取り(C)	Shift+Del
コピー(O)	Ctrl+Ins
貼り付け(P)	Shift+Ins
削除(D)	Del
全て消去(L)	Ctrl+Del

メモウィンドウにフォーカスがある場合は、[編集]メニューは、左図のメニューになります。

メニュー項目	機能
貼り付け(P)	クリップボードの内容を貼り付けます。
ページ複写(C)	ウィンドウに表示されている情報をクリップボードにコピーします。
元に戻す(U)	前回の編集を取り消します。
切り取り(C)	選択されている文字列をクリップボードに移動します。
コピー(O)	選択されている文字列をクリップボードにコピーします。
削除(D)	選択されている文字列を削除します。
全て消去(L)	メモウィンドウの内容を全て削除します。

## 4.2.3 検索メニュー

### 検索(S)

文字列の検索(F)	F4, Ctrl+QF
次を検索(N)	Ctrl+F4, Ctrl+L

[検索]メニューにはウィンドウ内の文字列を検索するコマンドが収められています。

メニュー項目	機能
文字列の検索(F)	検索する文字列を指定するダイアログボックスが表示されます。 (『6.1.5 文字列検索ダイアログボックス』82頁参照)
次を検索(N)	指定した文字列を指定した方向に検索します。

## 4.2.4 表示メニュー

### 表示(V)

- ▼ ステータスバー(S)
- ▼ ツールバー(T)
- ▼ ウィンドウバー(W)
- ▼ ヒント(H)

[表示]メニューにはツールバーなどの表示 / 非表示の選択をするコマンドが収められています。

メニュー項目	機能
ステータスバー(S)	ステータスバーの表示 / 非表示を選択します。 (『4.7 ステータスバー』51頁参照)
ツールバー(T)	ツールバーの表示 / 非表示を選択します。 (『4.5 ツールバー』47頁参照)
ウィンドウバー(W)	ウィンドウバーの表示 / 非表示を選択します。 (『4.6 ウィンドウバー』50頁参照)
ヒント(H)	ボタンに関するヒントの表示 / 非表示を選択します。

## 4.2.5 実行メニュー

### 実行(R)

プログラム実行(G)	F5
カム(C)	F7
トレース(T)	F8
ステップ(S)	F10
リターン実行(U)	Ctrl+F5
ブレークポイント(B)	F9
強制ブレーク(B)	ESC
エミュレーションROM設定(M)	
CPU環境設定(V)	

[実行]メニューにはユーザプログラムの実行 / 停止に関するコマンドが収められています。

[カム]と[ブレーク]はコードウィンドウにフォーカスがある場合のみ実行できます。

CPUによってメニュー構成が異なりますので詳細に関しては、『個別編』の「ダイアログコマンド」を参照してください。

メニュー項目	機能
プログラム実行(G)	ユーザプログラムを現在のPCから実行します
カム(C)	ユーザプログラムを現在のPCからカーソルのある行まで実行します。

(次頁に続く)

メニュー項目	機能
トレース(T)	ソース行単位、機械語単位のトレース実行をします。
ステップ(S)	ソース行単位、機械語単位のステップ実行をします。
リターン実行(U)	現在の関数の実行を終了(リターン)して呼び出し関数に戻るまで実行します。
ブレークポイント(P)	ブレークポイントの設定 / 解除をします。
強制ブレーク(B)	強制的にユーザプログラムを中断し、PARTNERに制御を戻します。(コマンド/マクロの連続実行中は連続実行を中断します)
エミュレーションROM 設定(M)	エミュレーションROM領域を設定します。 <sup>1</sup>
CPU環境設定(V)	CPUの環境値を設定します。 <sup>1</sup>

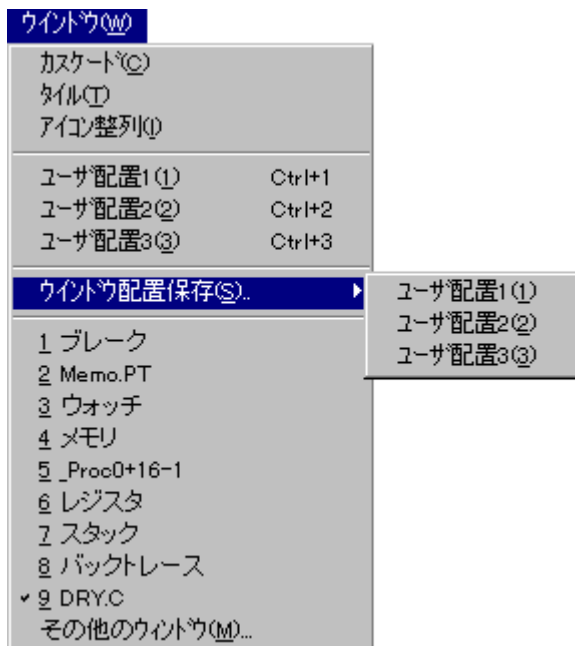
<sup>1</sup>PC、CBシリーズでは使用できません

#### 4.2.6 ローカルメニュー

ローカルメニューには、現在フォーカスのあるローカルウィンドウのメニューが登録されます。また、マウスの右クリックによって表示/選択することもできます。詳細に関しては、『5 チャイルドウィンドウ』53頁を参照してください。



## 4.2.7 ウィンドウメニュー



[ウィンドウ]メニューにはウィンドウ全体の制御に関するコマンドが収められています。

メニュー下部に表示されるウィンドウの並びは、使用状況に応じて変化します。

メニュー項目	機能
カスケード(C)	アイコン化されていないウィンドウを重ねて表示します。
タイル(T)	アイコン化されていないウィンドウを並べて表示します。 (PARTNER独自の配置)
アイコン整列(I)	アイコン化されているウィンドウを整列します。
ユーザ配置1(1)	[ウィンドウ配置保存]-[ユーザ配置1]で保存したウィンドウの配置を呼び出します。
ユーザ配置2(2)	[ウィンドウ配置保存]-[ユーザ配置2]で保存したウィンドウの配置を呼び出します。
ユーザ配置3(3)	[ウィンドウ配置保存]-[ユーザ配置3]で保存したウィンドウの配置を呼び出します。
ウィンドウ配置保存(S)	現在のウィンドウの配置を保存します。

## 4.2.8 設定メニュー

### 設定(S)

- 色選択(S)..
- ツールバー設定(T)..
- フォント設定(F)..
- オプション設定(O)..
- ウインドウリサイズ(R)
- 環境設定(E)..

[設定]メニューにはPARTNERの設定に関するコマンドが収められています。

メニュー項目	機能
色選択(S)	ウインドウ / 文字の色を選択するダイアログボックスを表示します。 (『6.1.4 色指定ダイアログボックス』82頁参照)
ツールバー設定(T)	ツールバーのボタンを登録するダイアログボックスを表示します。 (『6.1.3 ツールバーの設定ダイアログボックス』81頁参照)
フォント設定(F)	文字の大きさを選択するダイアログボックスを表示します。 (『6.1.2 フォントの指定ダイアログボックス』81頁参照)
オプション設定(O)	各種状態を指定するダイアログボックスを表示します。 <sup>1</sup>
ウインドウリサイズ(R)	メインウインドウの大きさに合わせて各チャイルドウインドウの大きさを連動します。
環境設定(C)	環境設定プログラム(RPTSETUP)を、現プロジェクト内容で起動して、起動オプション、コンフィグファイルの再設定ができます。ただし、PARTNERは一旦終了します。

<sup>1</sup> 『個別編』のチャイルドウインドウを参照してください。

## 4.2.9 ヘルプメニュー

ヘルプ(H)

目次(C)

キーワードで検索(S)...

PARTNER について(A)...

[ヘルプ]メニューにはヘルプに関するコマンドが収められています。

メニュー項目	機能
目次(C)	PARTNERのヘルプの目次を表示します。
キーワードで検索(S)	PARTNERのヘルプをキーワードで検索します。
PARTNERについて(A)	PARTNERについての情報を表示します。

## 4.3 ショートカットキー

ショートカットキーは、PARTNERのコマンドの中で頻繁に実行される操作（コマンド）をよりすばやく行えるよう定義したキーの組み合わせです。

### 4.3.1 各ウィンドウ共通のショートカットキー

以下に示すキー操作は、各ウィンドウに共通なショートカットキーです。

キー操作	機能
F1	フォーカスのあるウィンドウの([最大表示]/[元のサイズに戻す])を切り替えます。  注)コードウィンドウ,コマンドウィンドウは別のコマンドが定義されています。
F2	アクティブウィンドウを1番下に移動し、次のウィンドウをアクティブにします。
CTRL+O	
CTRL+B	一つ前のウィンドウをアクティブにします。(F2,CTRL+Oの逆)
F4	文字列検索ダイアログボックスを開き、検索文字列と検索方向を指定します。 (『6.1.5 文字列検索ダイアログボックス』82頁参照)
CTRL+Q F	
F5	ユーザプログラムを現在のPCから実行します。
F8	ユーザプログラムをトレース実行します。
F10	ユーザプログラムをステップ実行します。
CTRL+C	一画面分下を表示します。
ROLL UP	
CTRL+E	一単位分(一行分)上にカーソルを移動します。
CTRL+L	F4で指定した文字列を現在のカーソル位置から再検索します。
CTRL+R	一画面分上を表示します。
ROLL DOWN	
CTRL+X	一単位分(一行分)下にカーソルを移動します。

(次頁に続く)

キー操作	機能
CTRL+INS	フォーカスのあるウィンドウの一画面分の情報をクリップボードにコピーします。
CTRL+GRPH+F1 CTRL+GRPH+0	コードウィンドウ0の表示/アイコン化をします。
CTRL+GRPH+F2 CTRL+GRPH+C	コマンドウィンドウの表示/アイコン化をします。
CTRL+GRPH+F3 CTRL+GRPH+W	ウォッチウィンドウの表示/アイコン化をします。
CTRL+GRPH+F4 CTRL+GRPH+M	メモリウィンドウの表示/アイコン化をします。
CTRL+GRPH+F5 CTRL+GRPH+R	レジスタウィンドウの表示/アイコン化をします。
CTRL+GRPH+F6	スタックウィンドウの表示/アイコン化をします。
CTRL+GRPH+F7 CTRL+GRPH+L	ローカルウィンドウの表示/アイコン化をします。
CTRL+GRPH+F8 CTRL+GRPH+T	バックトレースウィンドウの表示/アイコン化をします。
CTRL+GRPH+F9 CTRL+GRPH+B	ブレークウィンドウの表示/アイコン化をします。
CTRL+GRPH+F10 CTRL+GRPH+E	メモウィンドウの表示/アイコン化をします。
CTRL+1	ユーザ配置1の呼び出しをします。
CTRL+2	ユーザ配置2の呼び出しをします。
CTRL+3	ユーザ配置3の呼び出しをします。
GRPH+F10	現在フォーカスのあるウィンドウのローカルメニューを表示します。

【注意】DOS/V機をご使用の方は、GRPHキーをAltキーに、ROLL UPキーをPage Downキーに、ROLL DOWNキーをPage Upキーに置き換えて使用してください。

### 4.3.2 ウィンドウ独自のショートカットキー

『4.3.1 各ウィンドウ共通のショートカットキー』とは別に、ウィンドウ独自のショートカットキーが定義されています。それぞれのウィンドウのショートカットキーは以下のページを参照してください。

ウィンドウ名	見出し	ページ
コードウィンドウ	5.1.1 コードウィンドウのショートカットキー	55
コマンドウィンドウ	5.2.1 コマンドウィンドウのショートカットキー	59
メモリウィンドウ	メモリウィンドウのショートカットキー	<sup>1</sup>
レジスタウィンドウ	レジスタウィンドウのショートカットキー	<sup>1</sup>
スタックウィンドウ	5.5.1 スタックウィンドウのショートカットキー	65
ローカルウィンドウ	5.6.1 ローカルウィンドウのショートカットキー	66
バックトレースウィンドウ	5.7.1 バックトレースウィンドウのショートカットキー	68
ウォッチウィンドウ	5.8.1 ウォッチウィンドウのショートカットキー	70
ブレークウィンドウ	ブレークウィンドウのショートカットキー	<sup>1</sup>
インスペクトウィンドウ	5.11.1 インスペクトウィンドウのショートカットキー	75
メモウィンドウ	5.10.1 メモウィンドウのショートカットキー	73
ヒストリウィンドウ	ヒストリウィンドウのショートカットキー	<sup>1, 2</sup>

<sup>1</sup> 『個別編』のチャイルドウィンドウを参照してください。

<sup>2</sup> PC、CBシリーズでは使用できません。

## 4.4 マウス操作

ウィンドウ内でのマウス操作には、各ウィンドウ共通の操作と独自の操作があります。

### 4.4.1 各ウィンドウ共通のマウス操作

各ウィンドウ共通のマウス操作を説明します。

#### 右ボタンクリック

フォーカスのあるウィンドウのウィンドウ範囲内で右ボタンをクリックすると、そのウィンドウのローカルメニューが表示され、コマンドを選択することができます。



図 17 コードウィンドウのローカルメニュー

**左ボタンドラッグ**

ウィンドウ内で左ボタンをドラッグし、必要な位置まで移動後に左ボタンを離すとリバーズ表示された選択部分をクリップボードにコピーします。

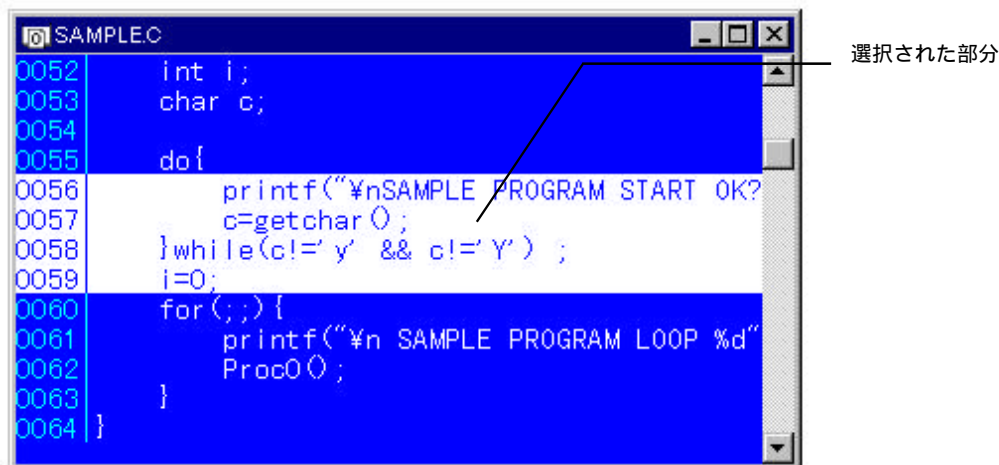


図 18 コピー範囲の選択中

【注意】ドラッグとはマウスのボタンを押したまま、マウスを動かすことです。

**4.4.2 ウィンドウ独自のマウス操作**

『4.4.1 各ウィンドウ共通のマウス操作』とは別に、ウィンドウ独自のマウス操作が定義されています。それぞれのウィンドウのマウス操作は以下のページを参照してください。

ウィンドウ	見出し	ページ
コードウィンドウ	5.1.3 コードウィンドウでのマウス操作	58
メモリウィンドウ	メモリウィンドウでのマウス操作	1
レジスタウィンドウ	レジスタウィンドウでのマウス操作	1
ローカルウィンドウ	5.6.3 ローカルウィンドウでのマウス操作	67
バックトレースウィンドウ	5.7.3 バックトレースウィンドウでのマウス操作	69
ウォッチウィンドウ	5.8.3 ウォッチウィンドウでのマウス操作	71
ブレークウィンドウ	ブレークウィンドウでのマウス操作	1
インスペクトウィンドウ	5.11.3 インスペクトウィンドウでのマウス操作	76

<sup>1</sup> 『個別編』のチャイルドウィンドウを参照してください。



## 4.5 ツールバー

ここでは、PARTNERのツールバーについて説明します。



図 19 ツールバーの例

ツールバーは、PARTNERのコマンドをボタンに登録しそのボタンをマウスでクリックすることで、PARTNERのコマンドを呼び出せる便利な機能です。

以下に、ボタンの登録方法と、各ボタンの機能を説明します。ツールバーは、[表示]-[ツールバー]コマンドで表示/非表示の選択ができます。(『4.2.4 表示メニュー』37頁参照)

### 4.5.1 ツールバーの設定



図 20 [ツールバーの設定]ダイアログボックス

[設定]-[ツールバー設定]メニューでツールバーの設定ダイアログボックスが表示されます。[ボタンリスト]からツールバーに登録するボタンを選択し、[ツールバー定義]に追加します。(詳しくは『6.1.3 ツールバーの設定ダイアログボックス』(81頁参照))

## 4.5.2 各ボタンの機能

ツールバーには以下のボタンが登録可能です。ボタンと機能を以下に示します。

### ボタン 機能

---



ユーザプログラムをロードします。

(『6.1.1 ファイルを開くダイアログボックス』80頁参照)



前回ロードしたプログラムを再ロードします。



ユーザプログラムを現在のPCから実行します。



ソース行単位あるいは機械語命令単位のトレース実行をします。



ソース行単位あるいは機械語命令単位のステップ実行をします。



現在の関数を呼び出し関数に戻る(リターン)まで実行します。



実行中のユーザープログラムを強制的に停止して、PARTNERに制御を戻します。

(コマンド/マクロの連続実行中は連続実行を中断します)



カーソル位置の変数をウォッチウィンドウに登録します。カーソル位置にウォッチ登録可能な変数名がない場合、ウォッチ登録ダイアログボックスを表示します。

(『6.1.7 ウォッチ設定ダイアログボックス』83頁参照)



カーソル位置の変数のインスペクトウィンドウを表示します。カーソル位置にインスペクト可能な変数名がない場合、インスペクトダイアログボックスを表示します。

(『6.1.6 インスペクト設定ダイアログボックス』83頁参照)



ヒントモードの切り替えをします。ヒントモードの時、ツールバーのボタンの上にマウスカーソルがあれば、ステータスウィンドウにボタンの説明が表示されます。



ツールバーの表示位置を切り替えます。



現在フォーカスのあるウィンドウの表示色を設定するダイアログボックスを表示します。

(『6.1.4 色指定ダイアログボックス』82頁参照)

(次頁に続く)

## ボタン 機能



現在フォーカスのあるウィンドウの表示フォントを設定するダイアログボックスを表示します。

(『6.1.2 フォントの指定ダイアログボックス』81頁参照)



現在フォーカスのあるウィンドウで検索する文字列を指定するダイアログボックスを表示します。

(『6.1.5 文字列検索ダイアログボックス』82頁参照)



現在フォーカスのあるウィンドウで前回指定した文字列を、現在のカーソル位置から検索します。



ロードされたユーザプログラムに含まれるモジュールを選択する為のダイアログボックスを表示します。

(『6.1.8 モジュールダイアログボックス』84頁参照)



ウィンドウ配置1を呼び出します。



ウィンドウ配置2を呼び出します。



ウィンドウ配置3を呼び出します。



アイコン化されていないウィンドウを重ねて表示します。



アイコン化されていないウィンドウを並べて表示します。(PARTNER独自の配置)



クリップボードから現在フォーカスがあるウィンドウに文字列を貼り付けます。



現在フォーカスのあるウィンドウの表示をクリップボードにコピーします。



ツールバーの設定のダイアログボックスを表示します。

(『6.1.3 ツールバーの設定ダイアログボックス』81頁参照)



メインウィンドウの大きさに合わせて各チャイルドウィンドウの大きさを連動します。



コードウィンドウのソースコード/逆アセンブル表示の切り替えをします。



PARTNERのヘルプを表示します。



PARTNERを終了します。

## 4.6 ウィンドウバー

ここでは、PARTNERのウィンドウバーについて説明します。



図 21 ウィンドウバー

ウィンドウバーは、PARTNERの各ウィンドウをすばやく呼び出すことができます。ウィンドウバーは、[表示]-[ウィンドウバー]コマンドでウィンドウバーの表示/非表示の選択を行えます。(『4.2.4 表示メニュー』37頁参照)

以下に各ボタンの対応を示します。

ボタン	ウィンドウ
	コードウィンドウ
	コマンドウィンドウ
	ウォッチウィンドウ
	メモリウィンドウ
	レジスタウィンドウ
	スタックウィンドウ
	ローカルウィンドウ
	バックトレースウィンドウ
	ブレークウィンドウ
	メモウィンドウ
	履歴ウィンドウ <sup>1</sup>
	I/Oウィンドウ
	コードウィンドウ1

<sup>1</sup> PC / CBシリーズ、また対象CPUによっては使用できません。

【注意】ウィンドウバーのボタン設定は変更できません。

## 4.7 ステータスバー

ステータスバーには、ショートカットキーのヒント、ボタン/メニューのヒント、エラーメッセージ、PARTNERの状態表示等が表示されます。ステータスバーは、[表示]-[ステータスバー]コマンドで表示/非表示の選択ができます。(『4.2.4 表示メニュー』37頁参照)

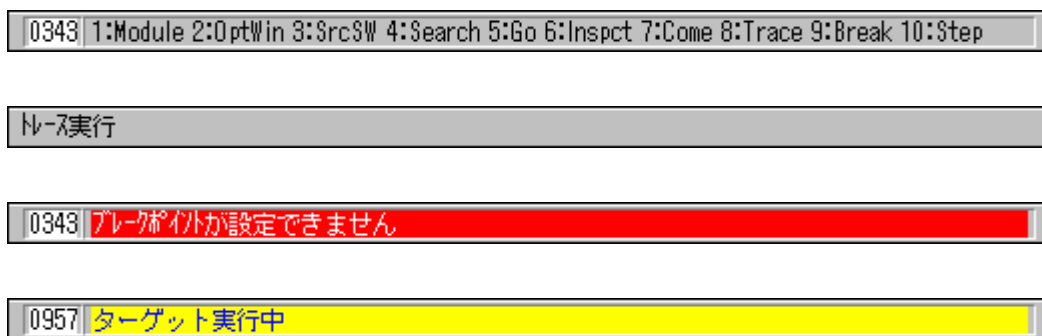


図 22 ステータスバーの例

空白ページ

## 5 チャイルドウィンドウ

チャイルドウィンドウは、デバッグ時に必要な様々な情報を表示します。また、操作のためのショートカットキーやメニュー、設定のためのダイアログボックスが用意されています。

## 5.1 コードウィンドウ

コードウィンドウは、ソースコードあるいはアセンブラとソースコードのミックス表示をします。

コードウィンドウは2つ存在し、モードを変更することによりいろいろな表示形態を指定することができます。

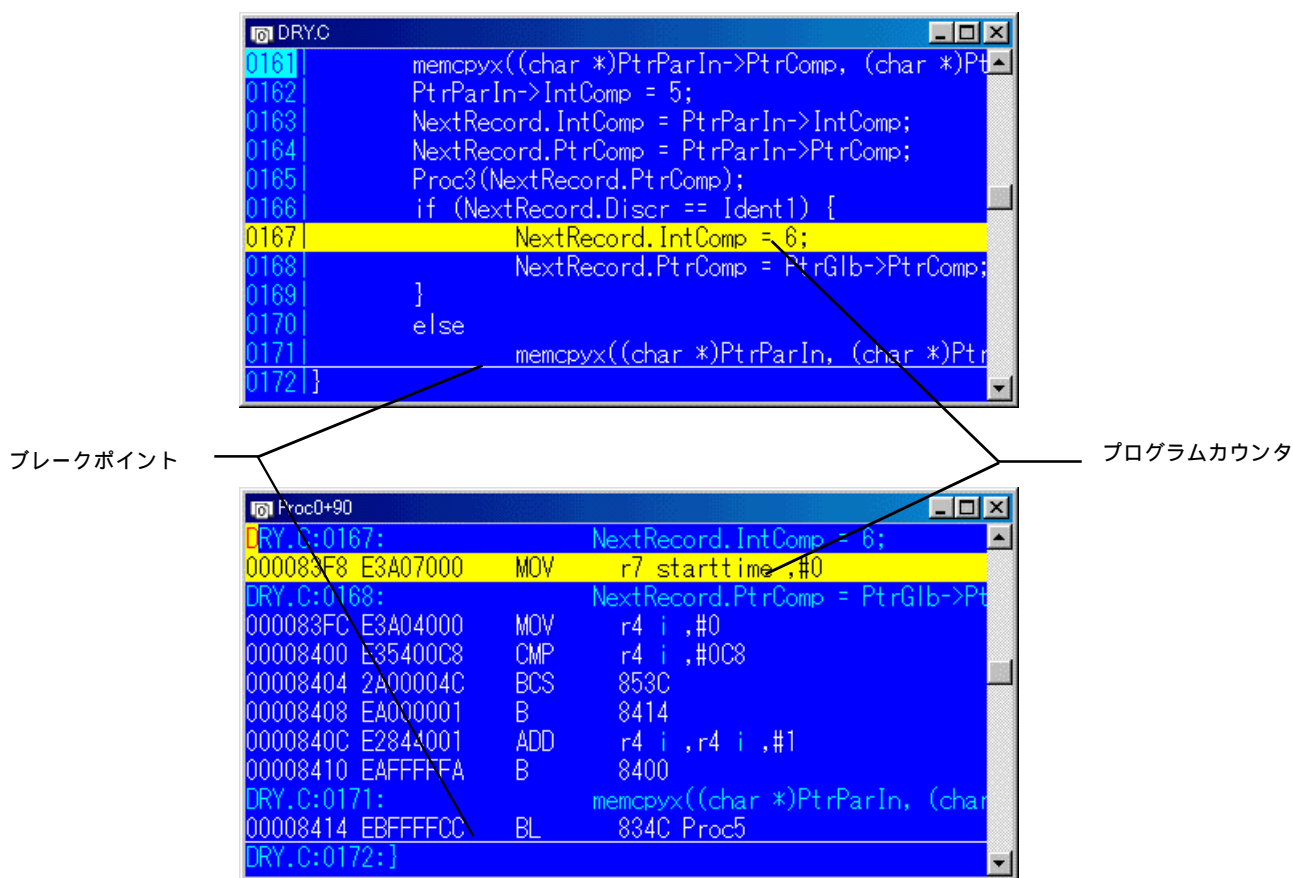


図 23 コードウィンドウ



### 5.1.1 コードウィンドウのショートカットキー

コードウィンドウには以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはコードウィンドウにフォーカスがある場合に適用されます。

キー操作	機能	ダイアログコマンド
F1	モジュールを選択する為のダイアログボックスを表示します。 (『6.1.8 モジュールダイアログボックス』84頁参照)	
F3 CTRL+F3	コードウィンドウがソースコード表示の場合は逆アセンブル表示、逆アセンブル表示の場合はソースコード表示に切り替えます。表示は現在のPCのある行を表示するように切り替えます。	Vコマンド Uコマンド
F6 CTRL+F6 CTRL+I	現在のカーソル位置の変数をインスペクトします。	INSコマンド
F7	現在のPCから現在のカーソル位置までユーザプログラムを実行します。	
F9	現在のカーソル位置にブレークポイントの設定 / 解除を行います。	BPコマンド
CTRL+F1	コードウィンドウの[最大化]/[元のサイズに戻す]を切り替えます。	
CTRL+F2	次のウィンドウにフォーカスを移動します。	
CTRL+F4 CTRL+L	現在のカーソル位置から以前に文字列検索で指定した方向と検索文字列を再検索します。	
CTRL+F5	現在の関数の実行を終了(リターン)して呼び出し関数に戻るまでユーザプログラムを実行します。	Gコマンド
CTRL+F7 CTRL+W	現在のカーソル位置の変数をウォッチウィンドウに登録します。	Wコマンド
CTRL+F8 CTRL+V	現在のカーソル位置の変数をコマンドウィンドウにビュー表示します。	VALコマンド ?コマンド

(次頁に続く)

キー操作	機能	ダイアログコマンド
CTRL+F9 CTRL+G	現在のカーソル位置の文字列をコマンドウィンドウのカーソル位置に取り込みます。	
CTRL+F10	カーソル位置の文字列をクリップボードにコピーします。	
CTRL+Q L	表示する行番号/アドレスを指定するダイアログボックスを表示します。	Vコマンド Uコマンド
CTRL+Q R	ソースコードの先頭を表示します。	
CTRL+Q C	ソースコードの最後を表示します。	
CTRL+Q J	起動時のウィンドウ配置に戻します。	
CTRL+Q P	現在のPCのある行を表示します。	
CTRL+F	ソースコード表示の時は次の文字列にカーソルを移動し、逆アセンブル表示のときは「アドレス コード モニック」の順にカーソルを移動します。	
CTRL+A	ソースコード表示のときは前の文字列にカーソルを移動します。	
CTRL+S	一単位分左にカーソルを移動します。	
CTRL+D	一単位分右にカーソルを移動します。	
SHIFT+ ROLL UP	以前のPCの位置にカーソルを移動します。	
SHIFT+ ROLLDOWN	次のPCの位置にカーソルを移動します。	

【注意】上記以外のSHIFTキーを押しながらの操作については『5.2.1 コマンドウィンドウのショートカットキー(59頁)』に記載されています。

SHIFT+ROLLUP,ROLLDOWNのキー操作は、保存している32ステップのPC位置にカーソルのみを移動します。実行やブレークを取り消すものではありません。以前のPC位置の確認等に使用します。

## 5.1.2 コードウィンドウのローカルメニュー

## コード

インスペクト(I)	F6, Ctrl+I
ウォッチ(W)	Ctrl+W
カム(C)	F7
アセンブラ(A)	F3
モジュール(M)	F1
行番号(L)	Ctrl+QL
先頭(T)	Ctrl+QR
最後(B)	Ctrl+QC
PC位置(P)	Ctrl+QP
ファイル固定(S)	
関数固定(F)	
フリック(P)	

[コード]メニューにはコードウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
インスペクト(I)	カーソル位置の変数のインスペクトウィンドウをオープンします。カーソル位置の文字列がインスペクトできる変数でない場合、インスペクト設定ダイアログを表示します。 (『6.1.6 インスペクト設定ダイアログボックス』83頁参照)
ウォッチ(W)	カーソル位置の変数をウォッチウィンドウに登録します。カーソル位置の文字列がウォッチ登録できない場合、ウォッチ設定ダイアログボックスを表示します。 (『6.1.7 ウォッチ設定ダイアログボックス』83頁参照)
アセンブラ(A)/ソース(S)	ソース表示と逆アセンブラ表示を切り替えます。
モジュール(M)	ロードされたユーザプログラムに含まれるモジュールを選択する為のダイアログボックスを表示します。 (『6.1.8 モジュールダイアログボックス』84頁参照)
行番号(L)/アドレス(A)	表示する行番号/アドレスの入力ダイアログボックスを表示します。 (『6.1.9 行番号指定ダイアログボックス』84頁参照, 『6.1.10 アドレス指定ダイアログボックス(コード)』85頁参照)
先頭(T)	表示中のファイルの先頭を表示します。
最後(B)	表示中のファイルの最後を表示します。

(次頁に続く)

メニュー項目	機能
PC位置(P)	現在のPC位置を表示します。
ファイル固定(S)	現在、PCの存在するファイルでチェックすると、チェックしたファイル以外にPCが移動した場合は、もう一つのコードウィンドウにフォーカスが移動します。また、チェックしたファイルにPCが移動した場合、チェックしたコードウィンドウにフォーカスが移動します。
関数固定(F)	現在、PCの存在する関数でチェックすると、チェックした関数以外にPCが移動した場合、もう一つのコードウィンドウにフォーカスが移動します。また、チェックした関数にPCが移動した場合、チェックしたコードウィンドウにフォーカスが移動します。
フリップ(P)	PCが存在する関数が変わる毎にコードウィンドウのフォーカスがもう一つのコードウィンドウに移動します。

【注意】ファイル固定,関数固定,フリップは、2つのコードウィンドウが表示されている場合にのみ有効になります。

### 5.1.3 コードウィンドウでのマウス操作

コードウィンドウではマウス操作に常に使用するコマンドが割り付けられています。

機能	マウス操作	ダイアログコマンド
インスペクト	変数を左ダブルクリック	INSコマンド
ブレークポイント	行番号/アドレスを左クリック	BPコマンド
トレース実行	SHIFT+左クリック	Tコマンド
ステップ実行	SHIFT+右クリック	Pコマンド
[実行]メニュー選択(37頁参照)	CTRL+右クリック	

## 5.2 コマンドウィンドウ

コマンドウィンドウは、ダイアログコマンドの入力、実行結果の出力及びシステムコール機能に関する入出力文字の表示をします。

コマンドウィンドウに入力するダイアログコマンドについてはオンラインヘルプを参照してください。

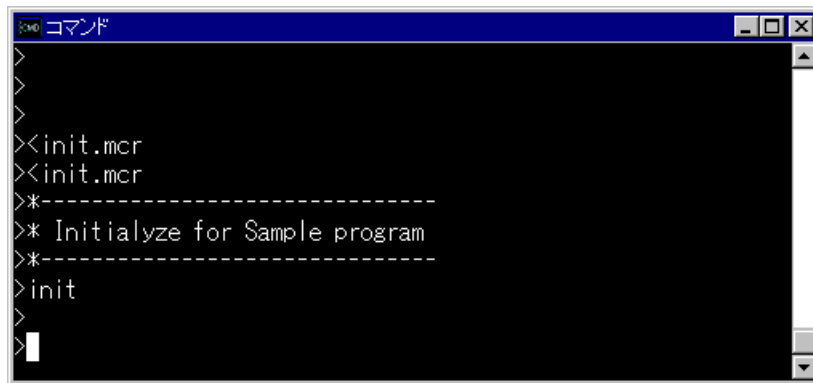


図 24 コマンドウィンドウ

### 5.2.1 コマンドウィンドウのショートカットキー

コマンドウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはコマンドウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
SHIFT+F1	直前に入力したコマンドの一文字コピーをします。
SHIFT+F2	以前入力したコマンド履歴のダイアログボックスの表示をします。 (『6.1.11 コマンド履歴ダイアログボックス』85頁参照)
SHIFT+F3	直前に入力したコマンドを現在のカーソル位置以降に文字列としてコピーします。
SHIFT+F4 CTRL+A	カーソルを行頭に移動します。

(次頁に続く)

キー操作	機能
SHIFT+F5 CTRL+F	カーソルを行末に移動します。
SHIFT+F6	コマンド入力行の最後の文字列から始まるグローバルシンボルの一覧をダイアログボックスに表示し、コマンドを拡張します。 (『6.1.12 シンボル拡張ダイアログボックス』86頁参照)
SHIFT+F7 CTRL+U	現在の編集行の全文字を削除します。
SHIFT+F8	現在の編集行の全文字とコマンド履歴の内容を全て削除します。
BS CTRL+H	カーソル左の1文字を削除します。
SHIFT+ CTRL+S	カーソルを左へ移動します。
SHIFT+ CTRL+D	カーソルを右へ移動します。カーソルが行末にある場合にはSHIFT+F3と同じ動作をします。
DEL CTRL+G	カーソル位置の文字を削除します。
INS CTRL+V	インサート(挿入)モード/オーバーライト(上書)モードを切り替えます。
SHIFT+ CTRL+W	直前のコマンド履歴の内容を表示します。また、コマンド入力行に文字列が入力されている場合は、その文字列で始まるコマンド新しい順に検索し、表示します。さらに、このキー入力をするると前のコマンド履歴を検索します。
SHIFT+	直前のコマンド履歴の内容を表示します。また、コマンド入力行に文字列が入力されている場合は、その文字列で始まるコマンド古い順に検索し、表示します。さらに、このキー入力をするると次のコマンド履歴を検索します。
SHIFT+INS	クリップボードにある文字列をコマンドとして入力します。

【注意】ファンクションキー、CTRL+ファンクションキーはコードウィンドウのショートカットと同様です(『5.1.1 コードウィンドウのショートカットキー(55頁)』参照)。ただし、カーソル位置に係るコマンドは使用できません。

## 5.2.2 コマンドウィンドウのローカルメニュー



[コマンド]メニューはコマンドウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
ロード(L)	ユーザプログラムをロードするためのダイアログボックスを表示します。
再ロード(R)	以前にロードしたユーザプログラムの再ロードをします。
貼り付け(P)	クリップボードの内容を貼り付けます。
履歴(H)	コマンド履歴ダイアログボックスを表示します。 (『6.1.11 コマンド履歴ダイアログボックス』85頁参照)
シンボル拡張(S)	コマンド入力行の最後の文字列から始まるグローバルシンボルの一覧ダイアログボックスを表示します。 (『6.1.12 シンボル拡張ダイアログボックス』86頁参照)

## 5.3 メモリウィンドウ

メモリウィンドウには、メモリの内容を各種表示形式で表示します。

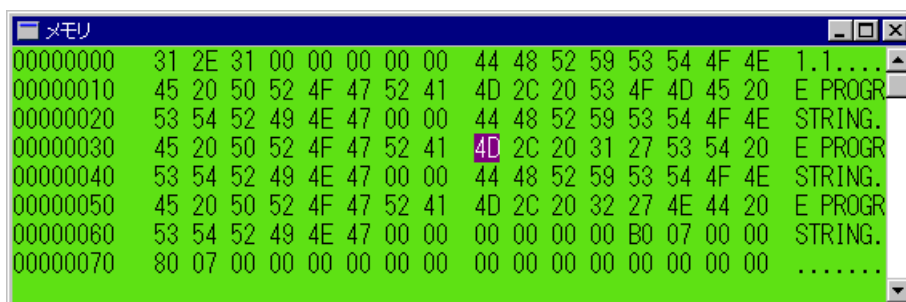


図 25 メモリウィンドウ

### 5.3.1 メモリウィンドウのショートカットキー

メモリウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはメモリウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F7 A	表示開始アドレスを指定するダイアログボックスが表示されます(『6.1.13 アドレス指定ダイアログボックス(メモリ)』86頁参照)。
F6 E Enter	データの変更を行うダイアログボックスが表示されます(『6.1.14 データ設定ダイアログボックス』87頁参照)。
B	メモリデータをバイト表示に指定します。
W	メモリデータをワード表示に指定します。
D	メモリデータをダブルワード表示に指定します。
S,F	メモリデータをショートフロート表示に指定します。
L	メモリデータをロングフロート表示に指定します。
C	メモリデータをASCII表示に指定します。
6,H	メモリデータを16進数表示に指定します。
1	メモリデータを10進数表示に指定します。



### 5.3.2 メモリウィンドウのローカルメニュー

メモリ(M)	
✓ バイト表示(B)	B
ワード表示(W)	W
ダブルワード表示(D)	D
ショートフロート(S)	S, F
ロングフロート(L)	L
ASCII表示(C)	C
✓ 16進数表示(H)	6, H
10進数表示(D)	1
アドレス設定(A)	A, F7
データ変更(E)	E, F6

[メモリ]メニューにはメモリウィンドウの制御に関するコマンドが収められています。

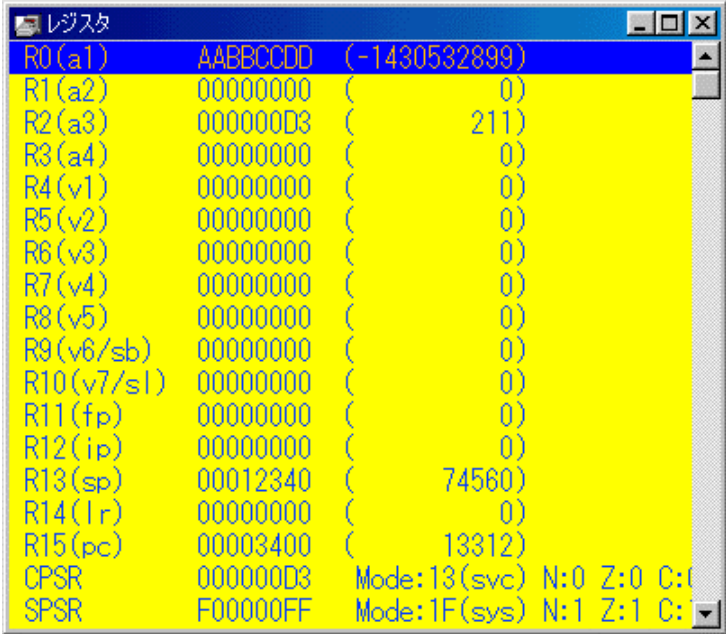
メニュー項目	機能
バイト表示(B)	メモリデータをバイト表示に指定します。
ワード表示(W)	メモリデータをワード表示に指定します。
ダブルワード表示(D)	メモリデータをダブルワード表示に指定します。
ショートフロート(S)	メモリデータをショートフロート表示に指定します。
ロングフロート(L)	メモリデータをロングフロート表示に指定します。
ASCII表示(C)	メモリデータをASCII表示に指定します。
16進数表示(H)	メモリデータを16進数表示に指定します。
10進数表示(D)	メモリデータを10進数表示に指定します。
アドレス設定(A)	表示開始アドレスを指定するダイアログボックスを表示します。(『6.1.13 アドレス指定ダイアログボックス(メモリ)』86頁参照)
データ変更(E)	カーソルのあるデータの内容を変更するダイアログボックスを表示します。(『6.1.14 データ設定ダイアログボックス』87頁参照)

### 5.3.3 メモリウィンドウでのマウス操作

データ部分を左ダブルクリックすると、データ変更を行うダイアログボックスが表示されます。(『6.1.14 データ設定ダイアログボックス』87頁参照)

## 5.4 レジスタウィンドウ

プログラムレジスタやシステムレジスタを(図 26)に示すような形式で表示します。



レジスタ名	値	コメント
R0(a1)	AABBCCDD	(-1430532899)
R1(a2)	00000000	(0)
R2(a3)	000000D3	(211)
R3(a4)	00000000	(0)
R4(v1)	00000000	(0)
R5(v2)	00000000	(0)
R6(v3)	00000000	(0)
R7(v4)	00000000	(0)
R8(v5)	00000000	(0)
R9(v6/sb)	00000000	(0)
R10(v7/sl)	00000000	(0)
R11(fp)	00000000	(0)
R12(ip)	00000000	(0)
R13(sp)	00012340	(74560)
R14(lr)	00000000	(0)
R15(pc)	00003400	(13312)
CPSR	000000D3	Mode: 13(svc) N:0 Z:0 C:0
SPSR	F00000FF	Mode: 1F(sys) N:1 Z:1 C:0

図 26 レジスタウィンドウ表示例

CPUにより構成レジスタが異なりますので詳細に関しては、『個別編』のレジスタウィンドウを参照してください。

## 5.5 スタックウィンドウ

スタックウィンドウは、現在のスタックメモリの内容を表示します。

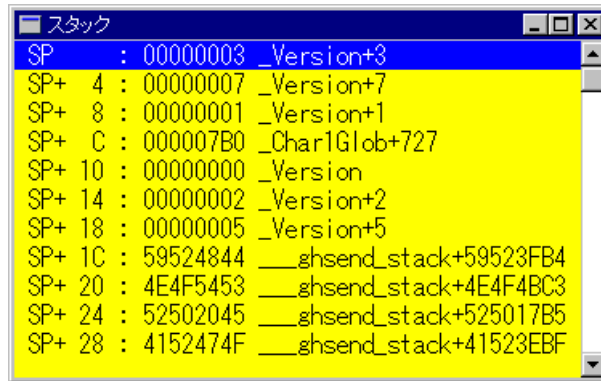


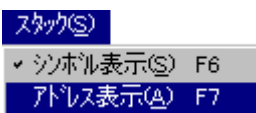
図 27 スタックウィンドウ

### 5.5.1 スタックウィンドウのショートカットキー

スタックウィンドウでは、以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはスタックウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F6	スタックのメモリ内容のシンボルを表示します。
F7	スタックの実メモリアドレスを表示します。

### 5.5.2 スタックウィンドウのローカルメニュー



[スタック]メニューにはスタックウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
シンボル表示(S)	スタックの値のシンボル表示をします。
アドレス表示(A)	スタックメモリのアドレス表示をします。

## 5.6 ローカルウィンドウ

ローカルウィンドウは、現在のプログラムカウンタ(PC)に該当する関数のローカル変数の内容を表示します。ローカル変数名の左に”+”が付いている変数は、その変数に表示可能な変数要素が含まれています。変数要素を表示している場合は、ローカル変数名の左に”-”が表示されます。

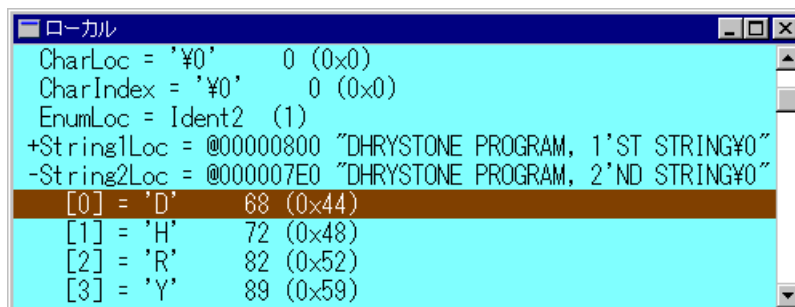


図 28 ローカルウィンドウ

### 5.6.1 ローカルウィンドウのショートカットキー

ローカルウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはローカルウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F6 CTRL+I	カーソルのあるローカル変数をインスペクトします。
F7 CTRL+W	カーソルのあるローカル変数をウォッチウィンドウに登録します。
F9	ローカル変数のオフセットを表示します。
Enter	変数要素があるローカル変数(変数名の左に”+”もしくは”-”が付いている変数)の表示/非表示を切り替えます。

## 5.6.2 ローカルウィンドウのローカルメニュー

ローカル(L)	
インスペクト(I)	F6, Ctrl+I
ウォッチ(W)	F7, Ctrl+W
要素表示(+)	Enter
オフセット(O)	F9

[ローカル]メニューにはローカルウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
インスペクト(I)	カーソルで選択されているローカル変数のインスペクトウィンドウを表示します。
ウォッチ(W)	カーソルで選択されているローカル変数をウォッチウィンドウに登録します。
要素表示(+)	配列や構造体などの要素があるローカル変数に対する内容の表示 / 非表示を切り替えます。
オフセット(O)	ローカル変数のオフセット値に対する表示 / 非表示を切り替えます。

## 5.6.3 ローカルウィンドウでのマウス操作

変数要素があるローカル変数部分を左ダブルクリックすると、変数要素の表示/非表示を切り替えます。

## 5.7 バックトレースウィンドウ

バックトレースウィンドウは、現在の関数がmain()関数から呼ばれてきた過程を表示します。

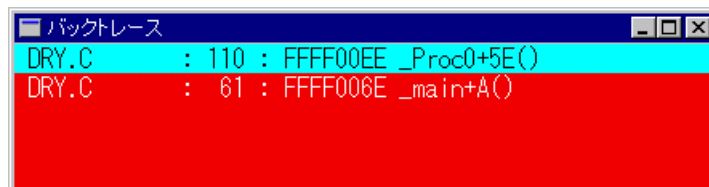


図 29 バックトレースウィンドウ

### 5.7.1 バックトレースウィンドウのショートカットキー

バックトレースウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはバックトレースウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F6 CTRL+I Enter	現在カーソルのある行のソース表示をコードウィンドウに行います。
F7	ウィンドウに表示されているシンボルのアドレスの表示/非表示を切り替えます。
F9	ウィンドウに表示されているシンボルのソース名、行番号の表示/非表示を切り替えます。

## 5.7.2 バックトレースウィンドウのローカルメニュー

バックトレース(B)	
インスペクト(I) F6, Ctrl+I	
ソース(S) F9	
アドレス(A) F7	

[バックトレース]メニューにはバックトレースウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
インスペクト(I)	カーソルで選択されているバックトレースアドレスをコードウィンドウに表示します。
ソース(S)	ソース名と行番号の表示 / 非表示の切り替えをします。
アドレス(A)	アドレスの表示 / 非表示の切り替えをします。

## 5.7.3 バックトレースウィンドウでのマウス操作

バックトレース部分を左ダブルクリックすると、選択した行のソースをコードウィンドウに表示します。

## 5.8 ウォッチウィンドウ

ウォッチウィンドウは、ウォッチ登録した変数の表示をします。

変数名の左に”+”が付いている変数は、その変数に表示可能な変数要素が含まれています。変数要素を表示している場合は、変数名の左に”-”が表示されます。

```

ウォッチ
1 | (int )starttime = 0 (0x0)
2 | +(struct Record *)PtrGlb = *00000780
3 | -(struct Record *)PtrGlbNext = *000007B0
   -[0] = @000007B0 {PtrComp=*000007B0,Discr=Ident1,EnumComp=
     +PtrComp = *000007B0
     Discr = Ident1 (0)
     EnumComp = Ident3 (2)
     IntComp = 5 (0x5)
     +StringComp = @000007C0 "DHRYSTONE PROGRAM, SOME STRING"
  
```

図 30 ウォッチウィンドウ

### 5.8.1 ウォッチウィンドウのショートカットキー

ウォッチウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはウォッチウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F3	登録されている変数を全て削除します。
F6	カーソルで選択されている変数をインスペクトします。
CTRL+I	
Enter	変数要素がある変数(変数名の左に”+”もしくは”-”が付いている変数)の表示/非表示を切り替えます。
INS	ウォッチ登録を追加するためのダイアログボックスを表示します。
DEL	カーソルで選択されている変数を削除します。



## 5.8.2 ウォッチウィンドウのローカルメニュー

ウォッチ(W)	
インスペクト(I)	F6, Ctrl+I
ウォッチ(W)	F7, Ctrl+W
要素の表示(+)	Enter
ウォッチ追加(A)	Ins
取り消し(C)	Del
全て取り消し(L)	F3

[ウォッチ]メニューにはウォッチウィンドウの制御に関するコマンドが収められています。

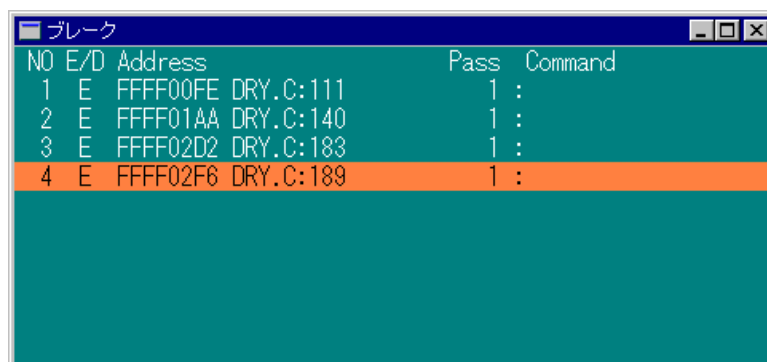
メニュー項目	機能
インスペクト(I)	カーソルで選択されている変数のインスペクトウィンドウを表示します。
ウォッチ(W)	カーソルで選択されている変数をウォッチウィンドウに登録します。
要素の表示(+)	配列や構造体など要素があるローカル変数に対する内容の表示 / 非表示の切り替えを行います。
ウォッチ追加(A)	新しくウォッチウィンドウに登録する変数名を入力するダイアログボックスの表示を行います。
取り消し(C)	カーソルで選択されている変数をウォッチウィンドウから削除します。
全て取り消し(L)	ウォッチウィンドウに登録されている全ての変数を削除します。

## 5.8.3 ウォッチウィンドウでのマウス操作

変数要素がある変数部分を左ダブルクリックすると、変数要素の表示/非表示を切り替えます。

## 5.9 ブレイクウィンドウ

ブレイクウィンドウは、現在設定されているブレイクポイントを表示します。



NO	E/D	Address	Pass	Command
1	E	FFFF00FE DRY.C:111	1	:
2	E	FFFF01AA DRY.C:140	1	:
3	E	FFFF02D2 DRY.C:183	1	:
4	E	FFFF02F6 DRY.C:189	1	:

図 31 ブレイクウィンドウ

CPUにより異なりますので詳細は、『個別編』のブレイクウィンドウを参照してください。

## 5.10 メモウィンドウ

メモウィンドウは、メモ帳に代わる簡易エディタです。このウィンドウはクリップボードをサポートしており、各ウィンドウとのデータのやりとりが可能です。

例えば、ユーザプログラムの実行前とブレーク後のレジスタの値をレジスタウィンドウからコピーしておき、レジスタ値の比較を行うことができます。

また、メモウィンドウにPARTNERマクロコマンドを入力しておき、コマンドウィンドウに入力することができます。

このウィンドウの内容は、PARTNERの終了時にファイル(memo.mt)に保存され、起動時に読み込まれます。

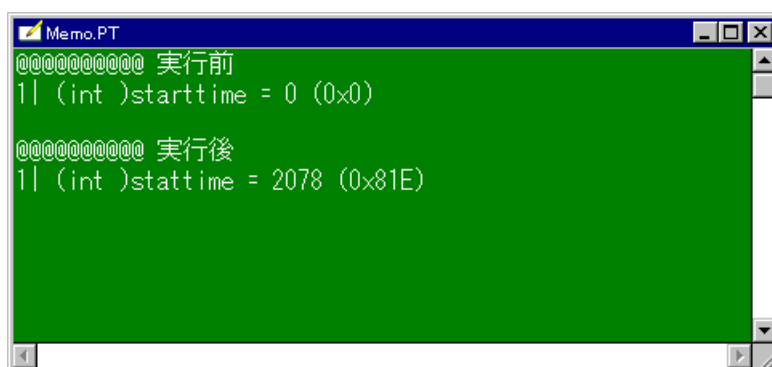


図 32 メモウィンドウ

### 5.10.1 メモウィンドウのショートカットキー

メモウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはメモウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
CTRL+INS	選択された文字列をクリップボードにコピーします。
CTRL+DEL	メモウィンドウの内容を全て削除します。
SHIFT+INS	クリップボードからカーソル位置に文字列を貼り付けます。
SHIFT+DEL	選択された文字列をクリップボードにコピーして削除します。
GRPH+BS	直前の編集を取り消します。

## 5.10.2 メモウィンドウのローカルメニュー

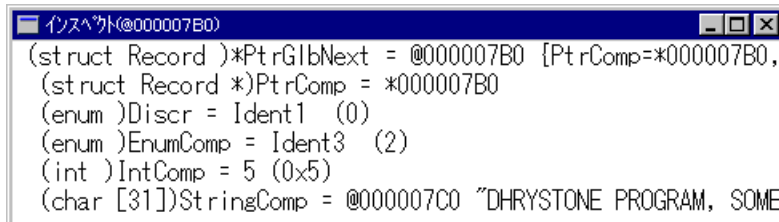
メモ(M)	
元に戻す(U)	Alt+BkSp
切り取り(C)	Shift+Del
コピー(O)	Ctrl+Ins
貼り付け(P)	Shift+Ins
削除(D)	Del
全て消去(L)	Ctrl+Del

[メモ]メニューにはメモウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
元に戻す(U)	前回の編集を取り消します。
切り取り(C)	選択されている文字列をクリップボードに移動します。
コピー(O)	選択されている文字列をクリップボードにコピーします。
貼り付け(P)	クリップボードの内容を貼り付けます。
削除(D)	選択されている文字列を削除します。
全て消去(L)	メモウィンドウの内容を全て削除します。

## 5.11 インスペクトウィンドウ

インスペクトウィンドウは、指定された変数のデータ構造に従った形式で表示を行います。



```

インスペクト(@000007B0)
(struct Record)*PtrGlbNext = @000007B0 [PtrComp=*000007B0,
(struct Record*)PtrComp = *000007B0
(enum)Discr = Ident1 (0)
(enum)EnumComp = Ident3 (2)
(int)IntComp = 5 (0x5)
(char [31])StringComp = @000007C0 "DHRYSTONE PROGRAM, SOME
    
```

図 33 インスペクトウィンドウ

### 5.11.1 インスペクトウィンドウのショートカットキー

インスペクトウィンドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはインスペクトウィンドウにフォーカスがある場合に適用されます。

キー操作	機能
F5	配列変数のデータ表示を10進数/16進数に切り替えます。
F6 I CTRL+I Enter	選択された変数をインスペクトウィンドウに表示します。
F7 W CTRL+W	選択された変数をウォッチウィンドウに登録します。
F8 V CTRL+V	選択された変数をコマンドウィンドウに表示します。
F9 R	選択された変数の表示範囲を指定するダイアログボックスを表示します。
F10 C	選択された変数の値を変更するダイアログボックスを表示します。

### 5.11.2 インスペクトウィンドウのローカルメニュー

インスペクト(I)	F6, I, Ctrl+I
ウォッチ(W)	F7, W, Ctrl+W
ビュー(V)	F8, V, Ctrl+V
範囲(R)	F9, R
変更(C)	F10, C
基数変更(D)	F5

[インスペクト]メニューにはインスペクトウィンドウの制御に関するコマンドが収められています。

メニュー項目	機能
インスペクト(I)	カーソルで選択されている変数のインスペクトウィンドウを開きます。
ウォッチ(W)	カーソルで選択されている変数をウォッチウィンドウに登録します。
ビュー(V)	カーソルで選択されている変数をコマンドウィンドウにビュー表示します。
範囲(R)	インスペクトウィンドウで表示される変数要素の範囲を指定するダイアログボックスを表示します。
変更(C)	カーソルで選択されている変数の値を変更するダイアログボックスを表示します。
基数変更(D)	配列で表示されるデータの基数を変更します。

### 5.11.3 インスペクトウィンドウでのマウス操作

変数部分を左ダブルクリックすると、指定した変数をインスペクトウィンドウに表示します。

## 5.12 ヒストリウィンドウ

ヒストリウィンドウには、リアルタイムトレースメモリの内容を16進ダンプ表示または、逆アセンブル表示で表示します。

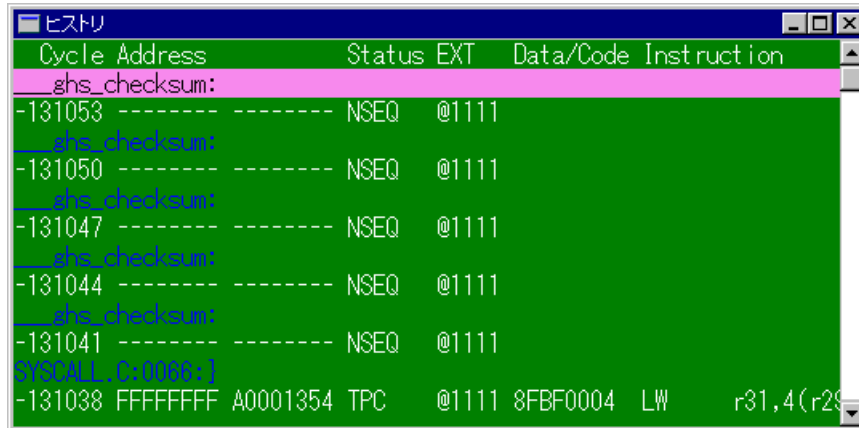


図 34 ヒストリ(逆アセンブル表示例)

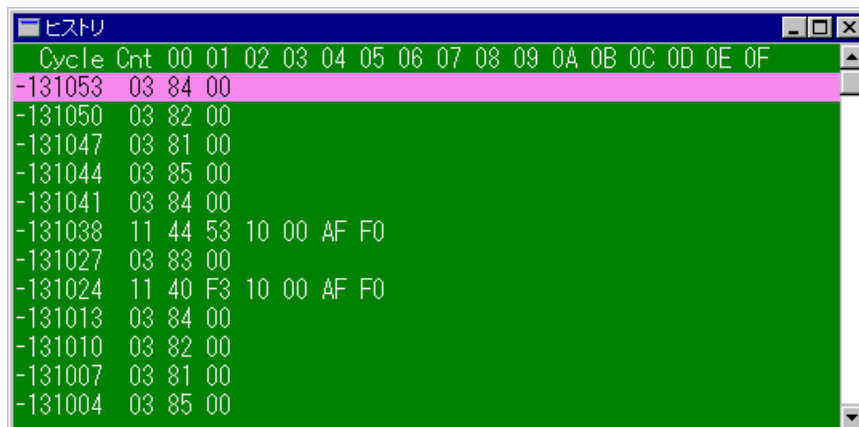


図 35 ヒストリ(ダンプ表示例)

CPUにより異なりますので詳細は、『個別編』のヒストリウィンドウを参照してください。

PC、CBシリーズでは表示されません。また、CPUによっても表示されないものがあります。

## 5.13 I/Oウィンドウ

I/Oウィンドウは、コプロセッサレジスタの表示と設定を行います(図 36)。

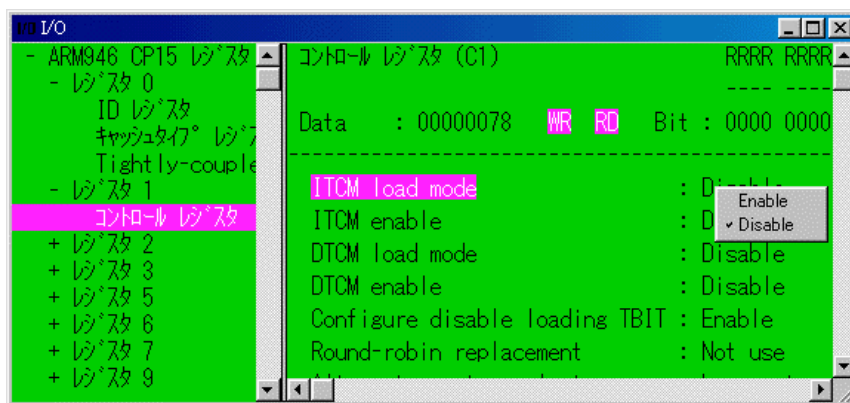


図 36 I/Oウィンドウの表示例

CPUにより異なりますので詳細は、『個別編』のI/Oウィンドウを参照してください。



## 6 ダイアログコマンド


PARTNERには、ダイアログボックスやツールバーによってデバッグを行う方法と、コマンドウィンドウにダイアログコマンドを入力してデバッグを行う方法の2通りあります。この章ではダイアログボックスによる操作に関して説明します。

## 6.1 ダイアログボックス

PARTNERには、多くのダイアログボックスがあります。以下に各ダイアログボックスの説明をします。

CPUにより異なる場合がありますので、『個別編』のダイアログコマンドの章も合わせて参照してください。

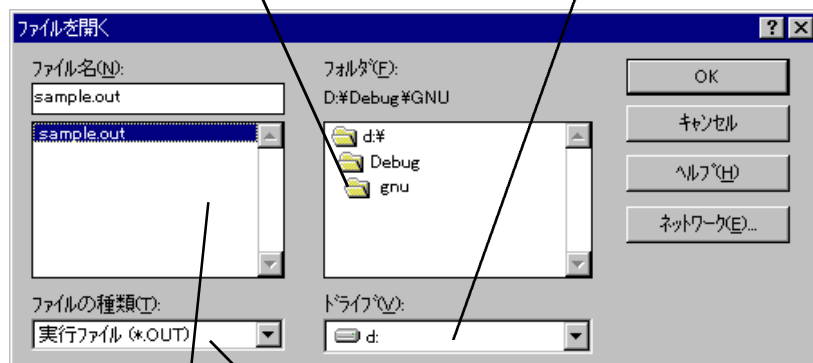
### 6.1.1 ファイルを開くダイアログボックス

[ファイル]-[ロード] 

[ファイルを開く]ダイアログボックスは、デバッグ対象のユーザプログラムを選択しロードします。

デバッグ対象のプログラムが保存されているディレクトリをダブルクリックします


ドライブを選択します



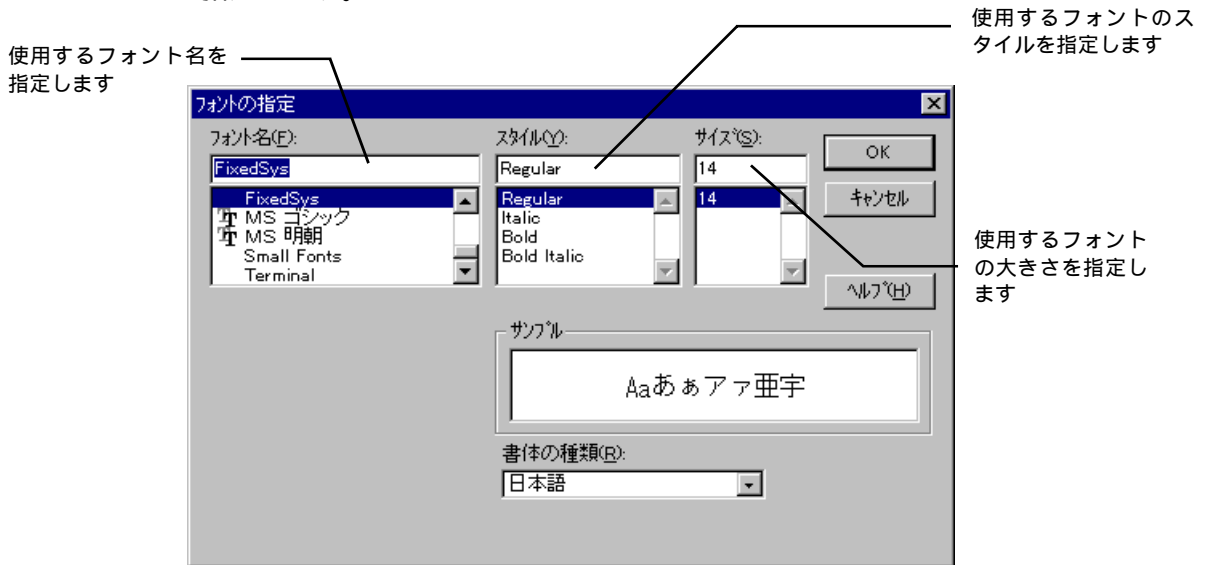
デバッグ対象のプログラムを選択します

[ファイル名:]ボックスに表示するファイルの種類を選択します


### 6.1.2 フォントの指定ダイアログボックス

[設定]-[フォント設定] 

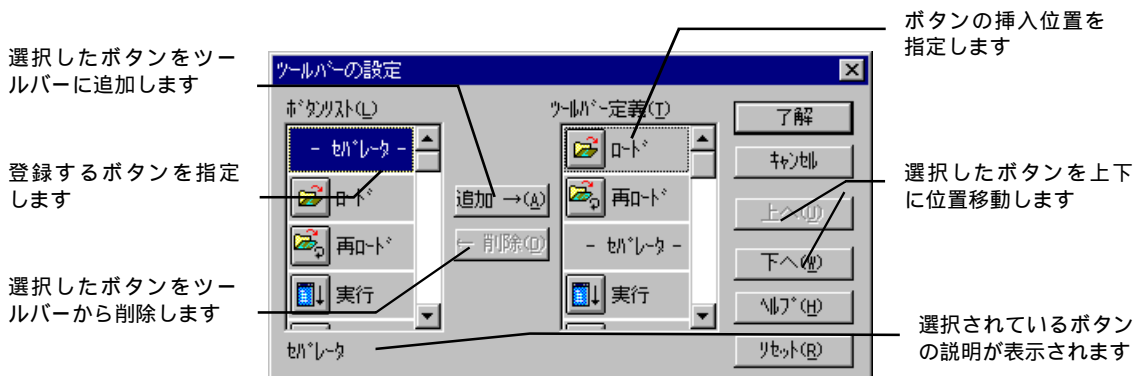
[フォントの指定]ダイアログボックスは、現在選択されているウィンドウのフォントを指定します。



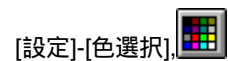
### 6.1.3 ツールバーの設定ダイアログボックス

[設定]-[ツールバー設定] 

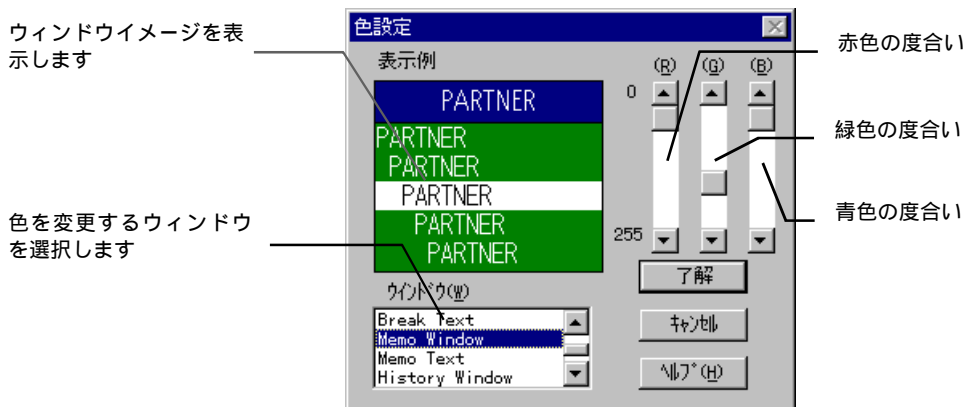
[ツールバーの設定]ダイアログボックスは、ツールバーに登録するボタンを設定します。[ツールバー定義]リストボックスでDELキーを押すと、選択されているボタンを削除します。



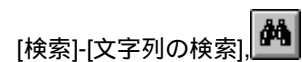
### 6.1.4 色指定ダイアログボックス



[色指定]ダイアログボックスは、各ウィンドウの背景色、文字色を指定します。



### 6.1.5 文字列検索ダイアログボックス



[文字列検索]ダイアログボックスは、フォーカスのあるウィンドウで検索するための文字列と方向を指定します。



**【注意】** 検索方向の初期状態はコマンドウィンドウでは[先頭へ]です。コマンドウィンドウ以外では[最後へ]です。

## 6.1.6 インスペクト設定ダイアログボックス

[コード]-[インスペクト],[ローカル/ウォッチ/インスペクト]-[インスペクト]

[インスペクト設定]ダイアログボックスは、インスペクトする変数名を指定します。



インスペクトする変数名を入力します

## 6.1.7 ウォッチ設定ダイアログボックス

[コード]-[ウォッチ],[インスペクト/ローカル]-[ウォッチ]

「ウォッチ設定」ダイアログボックスは、ウォッチ登録する変数名を指定します。



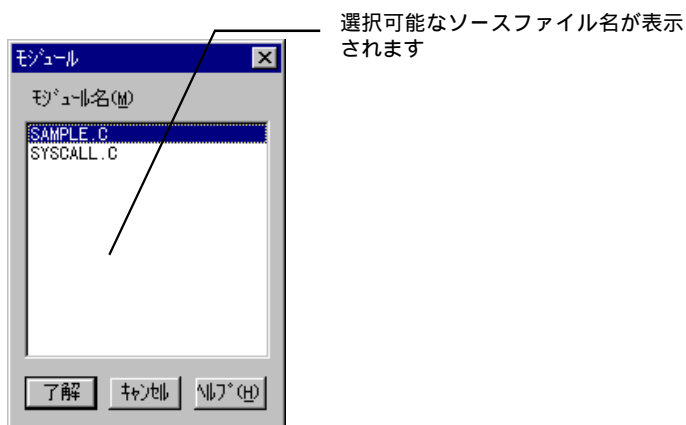
ウォッチ登録する変数名を入力します

**【注意】**ダイアログボックスでウォッチ登録できるのは、Cの変数のみです。直接メモリ内容を登録するには、Wコマンドを使用してください。Wコマンドについては、オンラインヘルプ、または、『個別編』(コマンドリファレンス)を参照してください。

## 6.1.8 モジュールダイアログボックス

[ファイル]-[モジュール],[コード]-[モジュール]

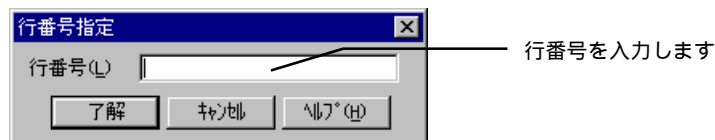
「モジュール」ダイアログボックスは、コードウィンドウに表示するソースファイルを指定します。[モジュール名]リストボックスには、ロードしたデバッグ対象ファイルのデバッグ情報に定義されているソースファイル名と、Vコマンドで参照したファイル名が表示されます。



## 6.1.9 行番号指定ダイアログボックス

[コード]-[行番号]

「行番号指定」ダイアログボックスは、コードウィンドウに表示するソースファイルの表示開始行番号を指定します。



### 6.1.10 アドレス指定ダイアログボックス(コード)

[コード]-[アドレス]

「アドレス指定(コード)」コードウィンドウに表示する逆アセンブルリストの表示開始アドレスを指定します。

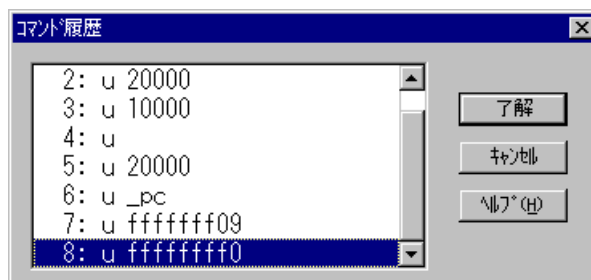


### 6.1.11 コマンド履歴ダイアログボックス

[コマンド]-[履歴]

「コマンド履歴」ダイアログボックスは、コマンドウィンドウに入力されたコマンドの履歴を表示し、選択します。

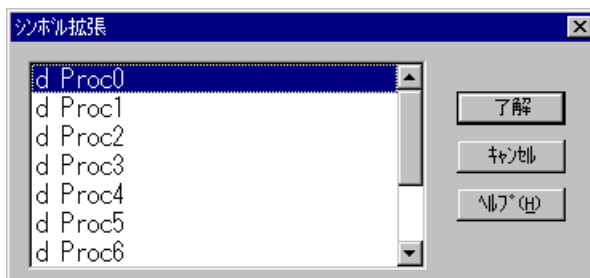
コマンドウィンドウのコマンド入力行(現在のプロンプト)に文字列が入力されている場合は、その文字列で始まる文字列をリストボックスに表示します。



### 6.1.12 シンボル拡張ダイアログボックス

[コマンド]-[シンボル拡張]

「シンボル拡張」コマンド入力行の最後の文字列から始まるグローバルシンボルの拡張表示をリストボックスにします。(下の例はコマンド入力行に”ProcX”と入力した場合です。)



### 6.1.13 アドレス指定ダイアログボックス(メモリ)

[メモリ]-[アドレス設定]

「アドレス指定(メモリ)」ダイアログボックスは、メモリウィンドウに表示するダンプリストの表示開始アドレスを指定します。

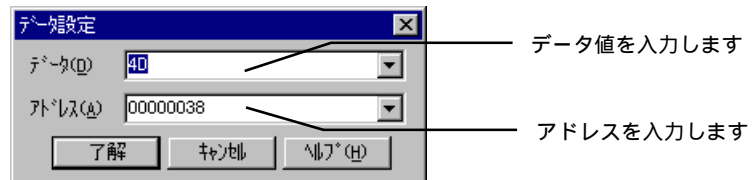




### 6.1.14 データ設定ダイアログボックス

[メモリ]-[データ変更]

「データ設定」ダイアログボックスは、メモリの内容を変更します。

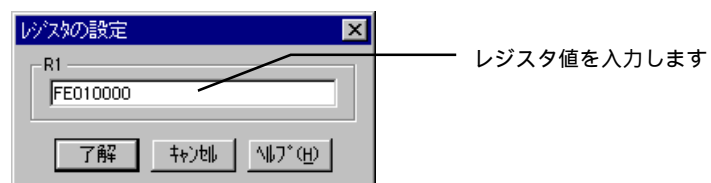


【注意】<了解>ボタンをクリックすると次のアドレスのデータ変更ができます。このダイアログボックスを閉じるには、<キャンセル>ボタンをクリックしてください。

### 6.1.15 レジスタの設定ダイアログボックス

[レジスタ]-[値の変更]

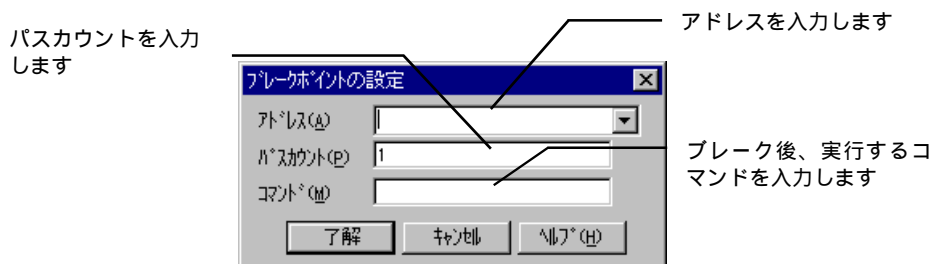
「レジスタの設定」ダイアログボックスは、レジスタの値を変更します。



### 6.1.16 ブレークポイントの設定ダイアログボックス

[ブレーク]-[ブレーク設定]

「ブレークポイントの設定」ダイアログボックスは、ブレークポイントを設定します。



### 6.1.17 データ変更ダイアログボックス

[インスペクト]-[変更]

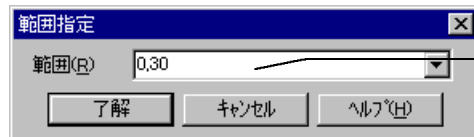
「データ変更」ダイアログボックスは、現在インスペクトされている変数あるいはインスペクトウィンドウで選択されている要素の値を変更します。



## 6.1.18 範囲指定ダイアログボックス

[インスペクト]-[範囲]

「範囲指定」ダイアログボックスは、配列あるいはポイントの表示要素開始番号と最大要素番号を変更します。



範囲をこのエディットボックスに開始番号、最大番号の形式で入力します

## 6.1.19 フレーム番号指定ダイアログボックス

[ヒストリ]-[ジャンプ]

「フレーム番号へジャンプ」ダイアログボックスは、表示するフレーム番号を指定します。



表示するフレーム番号を指定します

空白ページ

## 7 データ表現

この章では、PARTNERで扱うことのできるデータ表現について記述しています。

## 7.1 アドレス/データの表現

PARTNERでアドレスやデータ値として扱える表現はシンボルや演算式等、多岐にわたっています。

### 7.1.1 扱えるシンボル

PARTNERは、2種類のシンボルを扱うことができます。その一つは、グローバルシンボル(プログラムの全領域で有効な)です。もう一つはローカルシンボル(Cの関数内だけで有効な局所変数や静的変数のシンボル)です。

### 7.1.2 グローバルシンボル

グローバルシンボルは、逆アセンブルのラベルやアドレス入力時のアドレス値の代わりに利用できます。Cのextern変数や関数名がこのグローバルシンボルに登録されます。Cでは、一般に変数名や関数名の前あるいは後ろにアンダーバー(\_)を付けたシンボル名になっています。しかし、グローバルシンボルを入力するたびにアンダーバーを入力するのは大変不便です。PARTNERでは、グローバルシンボルの前後のアンダーバーは省略可能になっています。また、大文字と小文字を区別するかしないかを指定できます。(『個別編』の「コマンドリファレンス」、『[設定]-[オプション設定]』参照)

ただし、CPUのレジスタ名と同じ名前前のシンボルがあった場合レジスタ名が優先されます。したがって、レジスタ名と同名のシンボルは参照できません。

#### アドレス指定でグローバルシンボルを使用する場合

```
>u _main          /* _main シンボル値からの逆アセンブル表示 */
>u main          /* 上と同じ */
```



図 37 [アドレス指定]ダイアログボックスでアドレスを\_main シンボル値に指定

## マクロ内でグローバルシンボルを使用する場合

```
>i=5          /* 1 */
シンボルを登録しました
>while{ i!=0  /* 2 */
?T           /* 3 */
?i=i-1      /* 4 */
?}          /* 5 */
>
```

シンボル名 *i* に 5 を代入(*i* が未定義のシンボル名の場合には *i* をシンボル登録します)します。

シンボル *i* の値と 0 との比較を行います。

*i* が 0 以外の場合、T コマンドでトレース実行をします。

*i* の値から 1 を引いた値を *i* に代入します。

while{ コマンドの終了を意味しています。この行が実行されると再び while{ コマンドの条件判断 を行います。

上記のコマンドを実際に行うと、T コマンドを 5 回実行して while{ コマンドのループを抜け出します。

上記の例では、シンボル *i* は C や BASIC 等の高級言語の変数と同様な扱いとなっています。使用するシンボル名はすでに登録されているグローバルシンボル名やローカルシンボル名と重複してはいけません。

### 7.1.3 ローカルシンボル

ローカルシンボルとは、C の自動変数や関数の実引数などでその関数内だけで有効な変数とスタティック(静的)宣言された変数のシンボルです。ローカルシンボルは、通常デバッグ情報の読み込み時に自動的に登録されます。

ローカルシンボルは、アドレス値以外にローカルシンボルのスコープ(有効範囲)、属性(int,char,,,doubleなど)の情報を持っています。

### 7.1.4 特殊シンボル

#### \_\_ERR\_\_

特殊シンボル `_ERR_` は前回実行したコマンドでエラーが発生した場合には1となり、正常にコマンドを実行した場合には0となります。

`_ERR_` はマクロコマンド内でエラー処理に利用することができます。

#### \_\_RUN\_\_

特殊シンボル `_RUN_` はユーザプログラム実行中は1になります。ブレーク中は0になります。このシンボルは、マクロ内でユーザプログラムのブレーク待ちなどの処理に利用できます。

### 7.1.5 扱える数値

PARTNERは2/8/10/16進数を扱うことができ、数値の前に基数を示す記号を付けて区別します。基数を示す記号を省略した数値はNコマンドで指定した基数指定により扱いが変わります。

表記	基数
@数値	2進数
¥数値	8進数
_数値	10進数
\$数値	16進数
0x数値	16進数
数値	基数指定に従う(16進数,10進数)

例えば、@11001010,¥312,\_202,\$CA,0xCAはすべて同じ数値を表しています。

このほかに、4,8,12バイトの実数(IEEEフォーマット)を扱うためのコマンド(DS,DL,DT,SS,SL,ST)もあります。

### 7.1.6 アドレス

各コマンドのパラメータとしてアドレスを指定する部分には、シンボル名や行番号が入力できます。



### 7.1.7 行番号

PARTNERでは、ソースファイルの行番号を使ってソースレベルでデバッグ作業を行えます。

行番号は、ソースファイル中の目的の行を指定するために使用されます。行番号は、ロードされたデバッグ情報にソースファイル行情報が含まれている場合のみ有効です。行番号の入力形式には、次に示す3通りがあります。

形式1 .[ファイル名:]行番号

形式2 .±行番号

形式3 .シンボル±行番号

行番号は、10進数(行番号)とファイル名あるいはシンボル名を組み合わせた形でユーザプログラムの特定のソース行を指定します。

形式1は、絶対行番号を示します。ファイル名が入力された場合は、指定ファイルの第何行目(行番号)といった指定となります。ファイル名が省略されたときには、現在のカレントファイル(コードウィンドウに表示されているファイル)の第何行目(行番号)という表示になります。

```
>u.100          /* 現在選択されているソースの 100 行目の指定 */
>u.kmc:120      /* kmc.c の 120 行目の指定 */
```

形式2は、現在のプログラムカウンタの示すソース行からの相対位置を示す指定方法です。+行番号の場合は、現在のソース行よりも行番号分後ろの行を指し、-行番号の場合は、現在のソース行よりも行番号分前の行を指します。ただし、形式2の行番号指定で現在のPCに対応する行がない場合には入力エラーとなります。

```
>v.+10         /* PC の示すソースから 10 行目の指定 */
```

形式3は、指定したシンボルの値(アドレス)に対応するソース行からの相対位置を示す指定方法です。+行番号の場合は、対応するソース行よりも行番号分後ろの行を指し、-行番号の場合は、対応するソース行よりも行番号分前の行を指します。このとき指定されたシンボルに対応するソース行がない場合は入力エラーになります。

```
>bp.main+10    /* main シンボルから 10 行目の指定 */
```

### 7.1.8 文字列(ストリング)

PARTNERでは、数値の代わりに文字列(ASCIIコード列)を扱えます。文字列はシングルコーテーション(')で文字を囲んだ形で表現します。

```
'A' = $41
'AB' = $4142
'ABCD' = $41424344
```

特に、E/EBコマンドのエンター入力時には最大16文字までの文字列を一度に設定できます。

**【注意】**文字列中にシングルコーテーション(')を含む場合はシングルコーテーションまでの文字のみ入力され後は無視されます。シングルコーテーションは数値(\$27)を指定してください。

### 7.1.9 レジスタ名

PARTNERでは、レジスタの内容を数値として取り扱うことができます。

CPUにより異なりますので詳細は、『個別編』のレジスタ変数を参照してください。

### 7.1.10 演算式

演算式は、数値、シンボル及びレジスタを演算子で組み合わせたもので、一つの値を持ちます。PARTNERはCライクな数値、論理演算子を持っています。演算式は、各コマンドで値(データ、アドレス)を指定するすべてのところで利用できます。演算式に利用できる単項演算子、2項演算子は以下のものがあります。

単項演算子		
*		指定アドレスの 32 ビットデータ
+		単項プラス
-		単項マイナス
~		NOT(1 の補数)
!		論理否定
2 項演算子		
1	*	乗算
1	/	除算
1	%	モジュロ演算(剰余)
2	+	加算

(次頁に続く)

2	-	減算
3	>>	右シフト
3	<<	左シフト
4	>=	比較演算(右辺が左辺より大きいとき 1,その他 0)
4	<=	比較演算(右辺が左辺より小さいとき 1,その他 0)
4	>	比較演算(右辺が左辺より大きいとき 1,その他 0)
4	<	比較演算(右辺が左辺より小さいとき 1,その他 0)
5	==	比較演算(右辺が左辺と等しいとき 1,その他 0)
5	!=	比較演算(右辺が左辺と等しくないとき 1,その他 0)
6	&	AND
7	^	XOR
8		OR
9	&&	論理 AND
10		論理 OR
システム関数		
	VAL(C の式)	括弧内を C の式として評価(『』頁参照)

演算子の左の数字は演算子の優先順位です。隣り合った演算子の優先順位が同じ場合には、式は左から右へ評価を行います。ただし、括弧を付けることにより式の優先順位を変えることができます。

また、比較演算や論理AND,論理OR演算はマクロ(for{,while{コマンドなど)や条件判断処理コマンド(if{コマンドなど)の条件判断処理のために用意されています。

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

## 7.2 C言語レベルでのデータ表現

『7.1 アドレス/データの表現』では、グローバルシンボルやローカルシンボルあるいは行番号を単にアドレス値やデータ値として処理（演算）する式に関して説明しました。これらの式はほとんどのダイアログコマンドで使用できますが、ユーザプログラムの中の式は当然、Cの規約で記述されています。これらの式を扱うには先の式の処理では十分ではありません。

PARTNERでは新たにCの式をそのまま扱えるダイアログコマンドとウィンドウコマンドを用意しました。具体的には、インスペクト関連のコマンド、ウォッチ登録コマンド、VALコマンド、?コマンドでCの式がCの文法そのままに取り扱えます。

### 7.2.1 <式>

C言語レベルでの記述と『7.1 アドレス/データの表現』で説明した式の表現では、同じ式を記述してもその評価が異なります。この違いをCのグローバル変数abcを例に説明します。

```
>d abc                /* 変数 abc のアドレスよりメモリ表示 */
00001000 00 01 02 03 .....
>d abc+10             /* 変数 abc のアドレス+10 よりメモリ表示 */
00001010 AA BB CC DD .....
>? abc                /* 変数 abc の値を表示(Cの式として評価) */
(int ) 1 (0x1)
>? abc+10             /* 変数 abc の値+10 を表示(Cの式として評価) */
(int ) 11 (0xB)
>
```

この例からもわかるように同じabcやabc+10の記述をしても通常の式(上の例ではDコマンド)とCの式(上の例では?コマンド)ではその意味が異なります。インスペクト、ウォッチ、VALコマンド、?コマンドでは、abcをC言語レベルの変数として評価します。その他のコマンドでは、変数abcのアドレスとして評価します。

## 7.2.2 変数

Cの式の中で使用可能な変数または関数は、詳細なデバッグ情報を付けるオプションでコンパイルされたソースファイル内で宣言されたものに限られます。また、レジスタは擬似変数として使用できます。すべてレジスタ擬似変数はunsigned intの型(属性)を持ちます。なお、使用できるレジスタ変数に関しては、『個別編』の付録を参照してください。

## 7.2.3 変数スコープ

Cのプログラム作成やデバッグは、変数のスコープ(使用可能範囲)を意識しなくてはなりません。例えば、extern宣言された変数はすべてのプログラム領域で有効です。すなわち、スコープは全プログラム領域です。一方、関数内で宣言された自動変数はその関数処理の間だけ有効な変数です。スコープはその関数内となります。extern宣言された変数と関数で宣言された自動変数が同じ名前の場合、関数内では自動変数のみ有効でextern変数はアクセスできません。あるいは、現時点で使用していない関数の自動変数はメモリ上のどの空間にも存在しないため、いくら参照したくても見ることができません。

PARTNERでは、こういった処理をデバッグ情報から得たスコープ情報により自動的に処理します。

## 7.2.4 定数

Cの文法と全く同じ仕様です。また、デフォルトの基数はNコマンド(基数変更コマンド)の設定にかかわらず常に10進数となります。

表記	基数
数字	10進定数
0x 数字	16進定数
0X 数字	16進定数
0 数字	8進定数

例えば、4096(10進数),0x1000(16進数),010000(8進数)はすべて同じ値を表しています。また、文字定数についてもCのエスケープシーケンスをサポートしています。

Cの文字	数値	意味
'\a'	0x7	ベル
'\b'	0x8	バックスペース
'\f'	0xC	フォームフィード
'\n'	0xA	ラインフィード
'\r'	0xD	リターン
'\t'	0x9	水平タブ
'\v'	0xB	垂直タブ
'\¥'	0x5C	¥(円)記号
'\¥nnn'	nnn	8進数(8ビット)
'\¥xnn'	nn	16進数(8ビット)

## 7.2.5 演算子

演算子についてもCと全く同じ演算子を同じ文法でサポートしています。ただし、PARTNERでは、浮動小数点に関しては=演算子(代入演算)以外の演算子を使うことができません。

各演算子の優先順位は以下のようになります。

優先順位	演算子
1	関数(n) 配列[n] n.n n->n n++ n--
2	&n *n -n ~n !n ++n --n sizeof n #n
3	(キャスト)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

隣り合った演算子の優先順位が同じ場合には、式は左から右へ評価を行います。ただし、代入演算子(優先順位(16))のみ右から左に評価します。括弧を付けることにより式の優先順位を変えることができます。

## 7.2.6 副作用のある式

++,--,=などの代入演算子および関数呼び出しはその演算処理中にデバッグ対象のプログラムの変数内容などを変更してしまう副作用があります。変数値に代入演算子を使ってデータ変更したい場合もありますが、デバッグ中での変更は比較的少ないと思われます。

そこでPARTNERではウォッチ、インスペクト、?コマンドでは誤って式の評価の途中でデータ変更がおきないように副作用のある演算子を禁止しています。唯一、VALコマンドのみ副作用のある演算子の使用を許しています。したがって、データ参照のみのときは意識的にVALコマンドを使用せずに、?コマンドやインスペクトを使用してください。データ変更など副作用のある演算子を使用する場合にのみ、VALコマンドを使うようにしてください。

また、関数呼び出しは、関数が処理中にグローバル変数やスタティック変数を変更したり、あるいはポインタでその他のデータ領域を変更したりするかもしれません。そのため、次に実行を再開できなくなる場合もあります。VALコマンドで関数呼び出しを使う場合には注意してください。

```
>? abc=1234
副作用のある演算子は使用できません
>val abc=1234          /* abc に 1234 を代入 */
(int ) 1234 (0x4D2)
>val fnc(1,2,3)       /* fnc 関数コール */
(int ) 10 (0xA)
>
```



# 索引

## .

.CFG .....	21, 23
.DAT .....	22
.KPJ.....	21
.MCR .....	21

## I

I/Oウインドウ .....	78
INIT.MCR.....	21
INIT_PC.....	23
プログラムカウンタの初期化 .....	23
INIT_SP .....	23
スタックポインタの初期化.....	23

## M

MAP.....	24
コマンドでアクセス可能なメモリ領域	24
MEMO.PT .....	22

## P

PARTNERの起動	
PARTNERが起動した場合.....	26
PARTNERが起動しない場合 .....	27

## R

RPTSETUPの使用方法 .....	16
プロジェクト環境の設定 .....	19
既存のプロジェクト環境の変更 .....	19
既存のプロジェクト変更 .....	18
新規プロジェクトの作成 .....	17

## あ

インスペクトウインドウ .....	75
ショートカットキー .....	75
マウス操作 .....	76
ローカルメニュー .....	76
ウインドウコマンド.....	31
ウインドウバー.....	50
ウォッチウインドウ.....	70
ショートカットキー .....	70
マウス操作 .....	71
ローカルメニュー .....	71

## か

画面構成 .....	32
構成要素.....	32
画面表示.....	34
構成要素	
インスペクトウインドウ.....	34
ウォッチウインドウ.....	33
コードウインドウ .....	32
コマンドウインドウ.....	33
スタックウインドウ.....	33
バックトレースウインドウ .....	33
ヒストリウインドウ.....	34
ブレイクウインドウ.....	33
メモウインドウ .....	33, 34
メモリウインドウ .....	33
レジスタウインドウ.....	33
ローカルウインドウ.....	33
環境設定 .....	16
関連ファイル .....	21
起動 .....	15





編集メニュー.....	36	画面構成.....	33
<b>ま</b>		メモリウインドウ.....	62
マウス操作.....	45	ショートカットキー.....	62
インスペクトウインドウ.....	76	マウス操作.....	63
ウインドウ独自のマウス操作.....	46	ローカルメニュー.....	63
ウォッチウインドウ.....	71	<b>ら</b>	
コードウインドウ.....	58	レジスタウインドウ.....	64
バックトレースウインドウ.....	69	ローカルウインドウ.....	66
左ボタンドラッグ.....	46	ショートカットキー.....	66
メモリウインドウ.....	63	マウス操作.....	67
ローカルウインドウ.....	67	ローカルメニュー.....	67
各ウインドウ共通のマウス操作.....	45	ローカルメニュー	
マウス操作		インスペクトウインドウ.....	76
右ボタンクリック.....	45	ウォッチウインドウ.....	71
メニュー.....	35	コードウインドウ.....	57
ウインドウメニュー.....	39	コマンドウインドウ.....	61
ファイルメニュー.....	35	スタックウインドウ.....	65
ヘルプメニュー.....	41	バックトレースウインドウ.....	69
ローカルメニュー.....	38	メモウインドウ.....	74
メモウインドウ.....	73	メモリウインドウ.....	63
ショートカットキー.....	73	ローカルウインドウ.....	67
ローカルメニュー.....	74		

PARTNER ユーザーズ・マニュアル  
『ARMシリーズ共通編』

第二版 発行日 2001年12月

株式会社 マイダス・ラボ

Copyright 2001 Midas lab Inc./Kyoto Micro Computer Co.,LTD.