

PARTNER-V831/2-TPマニュアル

この度は、『RTE-V831-TP』用ソースレベルデバッガ『PARTNER-V831/2-TP』をお買い上げいただきまして、誠にありがとうございます。

『PARTNER-V831/2-TPマニュアル』はソースレベルデバッガ『PARTNER-V831/2-TP』の導入、起動、使用方法について説明されています。

重要

同封していますユーザ登録書に必要事項を漏れなく記入して直ちにご返送いただきますようお願いいたします。

ユーザサポート(技術的な問い合わせ、バージョンアップなどのお知らせ)は、このカードに基づいて行います。

本プログラム及び説明書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、改変等一切できません。

PARTNER-V831/2-TP(プログラム及び説明書)に関する著作権は株式会社マイダス・ラボおよび京都マイクロコンピュータ株式会社が所有します。

お客様に設定される使用権は、PARTNER-V831/2-TP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。

本製品の内容及び仕様は予告なしに変更されることがありますのでご了承ください。

本製品は、万全の注意を払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・ラボおよ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。

MS-WINDOWS、Windows、MS、MS-DOSはマイクロソフト社の商標です。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、一般に各メーカの商標です。

空白ページ

改定履歴

日付 Y / M / D	Rev	内 容
1998.6.30	1.0	初版
1998.11.10	1.1	PT-V831/2-TP Ver.1.1対応 *環境ファイルの説明変更(MAPの追加)

空白ページ

目次

はじめに.....	1
1 はじめてのPARTNER-V831/2-TP.....	5
1.1 PARTNER-V831/2-TPの特徴.....	6
1.1.1 最強のデータ参照/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-V831/2-TPの基本的な操作.....	7
2 セットアップ.....	11
2.1 セットアップの前に.....	12
2.1.1 RTE-V831-TPの設定.....	12
2.1.2 RTE-V831-TP用DLLのインストール.....	12
2.1.3 ハードディスクの空き領域.....	12
2.1.4 Windowsの起動.....	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-V831/2-TPのインストール.....	13
2.2.3 PARTNER-V831/2-TPのアンインストール.....	13
3 起動.....	15
3.1 PARTNER-V831/2-TP環境設定.....	16
3.1.1 環境設定プログラムの使用方法.....	16
3.2 環境ファイル(RPTV831T.CFG).....	21
3.2.1 設定項目.....	21
3.3 初期設定コマンド.....	22
3.4 関連ファイル.....	23

3.4.1	PARTNER-V831/2-TP起動時に必要なファイル.....	23
3.4.2	PARTNER-V831/2-TP終了時に作成するファイル.....	24
3.5	PARTNER-V831/2-TPの起動.....	25
3.5.1	PARTNER-V831/2-TPが正常に起動した場合.....	25
3.5.2	PARTNER-V831/2-TPが起動しない場合.....	26
4	ウインドウコマンド.....	29
4.1	画面構成.....	30
4.1.1	構成要素.....	30
4.1.2	画面表示.....	32
4.2	メニュー.....	33
4.2.1	ファイルメニュー.....	33
4.2.2	編集メニュー.....	34
4.2.3	検索メニュー.....	35
4.2.4	表示メニュー.....	35
4.2.5	実行メニュー.....	36
4.2.6	各ウインドウのローカルメニュー.....	37
4.2.7	ウインドウメニュー.....	42
4.2.8	設定メニュー.....	43
4.2.9	ヘルプメニュー.....	43
4.3	ショートカットキー.....	44
4.3.1	各ウインドウ共通のショートカットキー.....	44
4.3.2	ウインドウ独自のショートカットキー.....	46
4.4	マウス操作.....	47
4.4.1	各ウインドウ共通のマウス操作.....	47
4.4.2	ウインドウ独自のマウス操作.....	48
4.5	コードウインドウ.....	49
4.5.1	コードウインドウのショートカットキー.....	50
4.5.2	コードウインドウのローカルメニュー.....	52
4.5.3	コードウインドウでのマウス操作.....	53
4.6	コマンドウインドウ.....	54

4.6.1	コマンドウインドウのショートカットキー	54
4.6.2	コマンドウインドウのローカルメニュー	56
4.7	メモリウインドウ.....	57
4.7.1	メモリウインドウのショートカットキー	57
4.7.2	メモリウインドウのローカルメニュー	58
4.7.3	メモリウインドウでのマウス操作	58
4.8	レジスタウインドウ.....	59
4.8.1	レジスタウインドウのショートカットキー	60
4.8.2	レジスタウインドウのローカルメニュー	61
4.8.3	レジスタウインドウでのマウス操作.....	61
4.9	スタックウインドウ.....	62
4.9.1	スタックウインドウのショートカットキー	62
4.9.2	スタックウインドウのローカルメニュー	62
4.10	ローカルウインドウ.....	63
4.10.1	ローカルウインドウのショートカットキー	63
4.10.2	ローカルウインドウのローカルメニュー	64
4.10.3	ローカルウインドウでのマウス操作.....	64
4.11	バックトレースウインドウ.....	65
4.11.1	バックトレースウインドウのショートカットキー	65
4.11.2	バックトレースウインドウのローカルメニュー	66
4.11.3	バックトレースウインドウでのマウス操作	66
4.12	ウォッチウインドウ.....	67
4.12.1	ウォッチウインドウのショートカットキー	67
4.12.2	ウォッチウインドウのローカルメニュー	68
4.12.3	ウォッチウインドウでのマウス操作.....	68
4.13	ブレークウインドウ.....	69
4.13.1	ブレークウインドウのショートカットキー	69
4.13.2	ブレークウインドウのローカルメニュー	70
4.13.3	ブレークウインドウでのマウス操作.....	70
4.14	履歴ウインドウ.....	71
4.14.1	履歴ウインドウのショートカットキー	71

4.14.2	ヒストリウインドウのローカルメニュー	72
4.15	インスペクトウインドウ	73
4.15.1	インスペクトウインドウのショートカットキー	73
4.15.2	インスペクトウインドウのローカルメニュー	74
4.15.3	インスペクトウインドウでのマウス操作	74
4.16	メモウインドウ	75
4.16.1	メモウインドウのショートカットキー	75
4.16.2	メモウインドウのローカルメニュー	76
4.17	I/Oウインドウ	77
4.17.1	I/Oウインドウのショートカットキー	77
4.17.2	I/Oウインドウのローカルメニュー	78
4.17.3	I/Oウインドウでのマウス操作	78
4.18	ツールバー	79
4.18.1	ツールバーの設定	79
4.18.2	各ボタンの機能	80
4.19	ウインドウバー	82
4.20	ステータスバー	83
5	ダイアログコマンド	85
5.1	ダイアログボックス	86
5.1.1	ファイルを開くダイアログボックス	86
5.1.2	フォントの指定ダイアログボックス	87
5.1.3	ツールバーの設定ダイアログボックス	87
5.1.4	色指定ダイアログボックス	88
5.1.5	文字列検索ダイアログボックス	88
5.1.6	インスペクト設定ダイアログボックス	89
5.1.7	ウォッチ設定ダイアログボックス	89
5.1.8	モジュールダイアログボックス	90
5.1.9	行番号指定ダイアログボックス	90
5.1.10	アドレス指定ダイアログボックス(コード)	91
5.1.11	コマンド履歴ダイアログボックス	91

5.1.12	シンボル拡張ダイアログボックス	92
5.1.13	アドレス指定ダイアログボックス(メモリ)	92
5.1.14	データ設定ダイアログボックス.....	93
5.1.15	レジスタの設定ダイアログボックス.....	93
5.1.16	ブレークポイントの設定ダイアログボックス.....	94
5.1.17	アクセスブレークの設定ダイアログボックス.....	94
5.1.18	各種状態の設定ダイアログボックス.....	95
5.1.19	トレースアドレス設定ダイアログボックス	95
5.1.20	トレースモード設定ダイアログボックス.....	96
5.1.21	エミュレーションROM設定ダイアログ	97
5.1.22	CPU環境設定ダイアログボックス.....	98
5.1.23	フレーム番号指定ダイアログボックス	98

6 データ表現..... 99

6.1	データ表現.....	100
6.1.1	PARTNERで扱えるシンボル.....	100
6.1.2	グローバルシンボル	100
6.1.3	ローカルシンボル	101
6.1.4	特殊シンボル	102
6.1.5	PARTNERで扱える数値.....	102
6.1.6	アドレス	102
6.1.7	行番号.....	103
6.1.8	文字列(ストリング).....	104
6.1.9	レジスタ名.....	104
6.1.10	演算式.....	105
6.2	C言語レベルでのデータ表現.....	107
6.2.1	Cの式.....	107
6.2.2	Cの変数	108
6.2.3	Cの変数スコープ	108
6.2.4	定数	109
6.2.5	演算子.....	110
6.2.6	副作用のある式.....	111

7 コマンドリファレンス..... 113

7.1 コマンド解説の規約.....	114
7.2 機能別コマンド.....	115
7.2.1 ハードウェアの初期化	115
7.2.2 CPUリセット	115
7.2.3 環境設定コマンド	115
7.2.4 プログラムロード	115
7.2.5 ファイルリード/ライト.....	115
7.2.6 プログラム実行.....	115
7.2.7 ブレークポイント.....	116
7.2.8 アクセスブレークポイント.....	116
7.2.9 レジスタ表示/変更	116
7.2.10 システムレジスタ表示/変更.....	116
7.2.11 メモリ表示/変更	116
7.2.12 I/Oポート入力/出力	117
7.2.13 アセンブル.....	117
7.2.14 バックトレース.....	117
7.2.15 リアルタイムトレース.....	117
7.2.16 式の表示	118
7.2.17 シンボル表示/設定.....	118
7.2.18 Cのデータ参照/変更	118
7.2.19 文字列の定義(Cの式評価).....	118
7.2.20 コード表示.....	119
7.2.21 システムコール.....	119
7.2.22 システム制御	119
7.2.23 基数変更	119
7.2.24 ログイング/バッチ.....	119
7.2.25 オプション設定.....	120
7.2.26 マクロコマンド.....	120
8 高級言語レベルデバッグとROM化	121
8.1 Green Hills C	122
8.1.1 コンパイルとリンク方法.....	122
8.1.2 使用例.....	122
8.2 GNU C(exeGCC)	124

8.2.1 コンパイルとリンク方法.....	124
8.2.2 使用例.....	124
8.3 NEC(CA830) C	126
8.3.1 コンパイルとリンク方法.....	126
8.3.2 使用例.....	127
付録1.....	129
コマンドラインによる起動オプションの設定.....	129
付録2.....	133
エラーメッセージ.....	133
索引.....	137

空白ページ

はじめに

マニュアルについて

このマニュアルは、『PARTNER-V831/2-TP』の導入方法、起動手順、操作方法について説明しています。PARTNER-V831/2-TP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。

PARTNER-V831/2-TPの主な機能と概要

PARTNER-V831/2-TPは、『RTE-V831-TP』用のWindows版ソースレベルデバッガです。本デバッガはデバッガ情報量の拡大、MDIウインドウ、メニュー、スピードバーの採用等、Windows用に機能拡張した製品です。

パッケージの内容

PARTNER-V831/2-TPのパッケージには、セットアップCD-ROMとこのマニュアルが入っています。

PARTNER-V831/2-TPマニュアル(本書)

本マニュアルでは、PARTNER-V831/2-TPの導入方法、操作方法について説明されています。

動作環境

セットアップを始めるにあたり、PARTNER-V831/2-TPパッケージ以外にご用意いただくものについて説明します。セットアップを始める前に、現在お使いになっているハードウェアとソフトウェアをご確認ください。

必要なハードウェア

パーソナルコンピュータ

CPUに80386以上のCPUを搭載し、Windows95/NTが動作し、下記の『インターフェースキット』が接続された、パーソナルコンピュータが必要です。

ディスプレイ

コンピュータ本体に接続可能あるいはコンピュータに内蔵されていて、Windows95/NTに対応したディスプレイをご使用ください。

ハードディスク

PARTNER-V831/2-TPを使用するには、ハードディスクが必要です。PARTNER-V831/2-TPをセットアップするには、使用するハードディスクに2MB以上の空き領域が必要です。

CD-ROMドライブ

PARTNER-V831/2-TPをセットアップするために、CD-ROMドライブが必要です。

マウス

PARTNER-V831/2-TPの機能を最大限に引き出すため、コンピュータ本体に接続可能で、Windows95/NTに対応したマウスが必要です。

RTE-V831-TP

『RTE-V831-TP』本体が必要です。

インターフェースキット(IF-98D,IF-PCD,IF-CARD,LAN-BOX)

パーソナルコンピュータとRTE-V831-TPを接続するためのインターフェースキットが必要です。インターフェースキットには、『PC Cardインターフェースキット』、『PC98 Desk top PC用インターフェースキット』、『DOS/V Desk Top PC用インターフェースキット』、『10Base-TのLAN-BOX』等があります。

RTE-PS01 『電源ユニット』

RTE-V831-TP用の電源ユニットが必要です。

必要なソフトウェア

Windows

日本語Windows95/NTが必要です。PARTNER-V831/2-TP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。

マニュアルについて

『PARTNER-V831/2-TP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-V831/2-TPには、画面上で機能や使い方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-V831/2-TP

PARTNER-V831/2-TPをお使いになるに当たり、PARTNER-V831/2-TPの特徴とデバッグの手順を簡単に説明します。

1.1 PARTNER-V831/2-TPの特徴

PARTNER-V831/2-TPには快適にデバッグ作業を行うため、数々の機能が用意されています。

1.1.1 最強のデータ参照/変更およびリンク機能

従来から定評があるPARTNERのデータ参照/変更機能(インスペクト等)が大幅に機能アップしております。例えば、ソースウイ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インドウが開き、簡単に変数のデータ参照/変更が可能です。また、ウォッチウインドウ、ローカルウインドウでも同様に変数のデータ参照/変更が可能になっています。さらに、マウス操作でレジスタウインドウ、メモリウインドウでもデータの変更が可能になりました。

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、日常使われる機能がツールバーに登録することが可能になっており、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、ウイ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。また、ウインドウレイアウトは3種類まで登録可能です。

の基本的な操作

セットアップ

PARTNER-V831/2-TPをハードディスクにセットアップします。

【参照】『2 セットアップ』(10頁)

環境設定

PARTNER-V831/2-TP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-V831/2-TP環境設定』(16頁)

環境設定には、PARTNER-V831/2-TP環境設定用プログラムで行います。(図 1)



図 1 PARTNER-V831/2-TP環境設定プログラム

起動

環境設定が完了した後、PARTNER-V831/2-TPを起動します。(図 2)

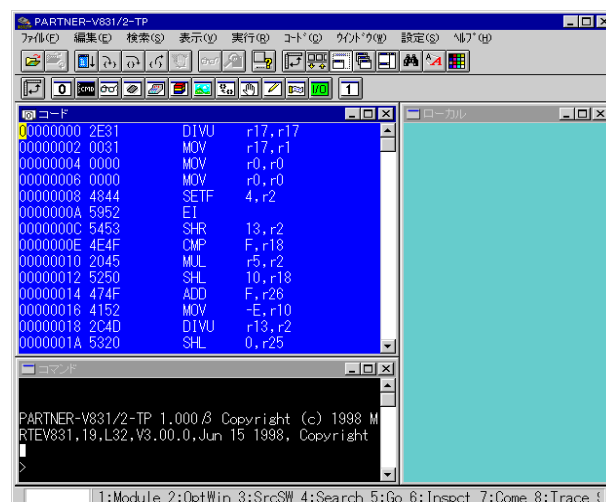


図 2 PARTNER-V831/2-TP起動画面

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(80頁),Lコマンド,[ファイル]-[ロード]コマンド(33頁)

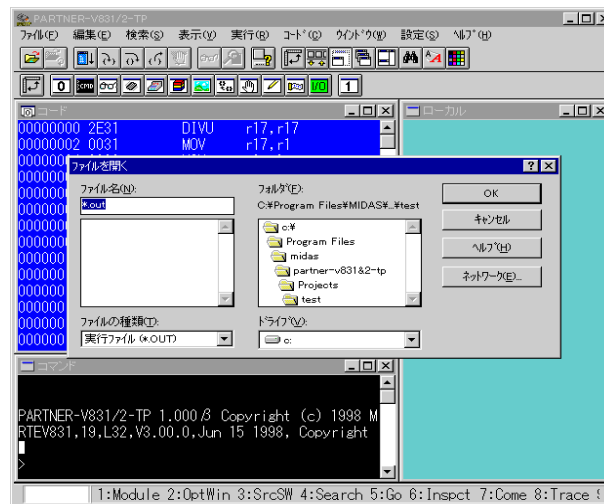


図 3 デバッグ対象ファイルのロード

ブレークポイントの設定

コードウイ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、『4.5.3 コードウインドウでのマウス操作』(53頁),[ブレーク]-[ブレーク設定](70頁)

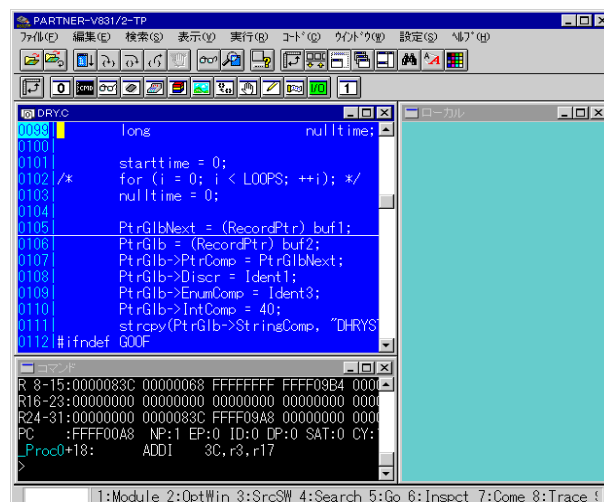




図 4 ブレークポイントの設定


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で強制的に停止します。

【参照】  ボタン(80頁),Gコマンド,[実行]-[プログラム実行] (36頁),  ボタン(80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インドウに表示します。(図 5)

【参照】  ボタン(80頁), 『4.5.1 コードウインドウのショートカットキー』 (50頁),INSコマンド

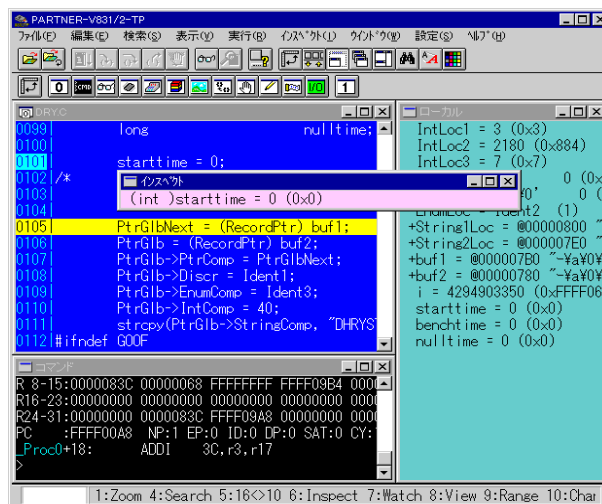


図 5 IntLoc3のインスペクト表示

変数のウォッチ登録

常時参照したい変数をウォッチウインドウに登録します。(図 6)

【参照】  ボタン(80頁), 『4.5.1 コードウインドウのショートカットキー』(50頁), Wコマンド

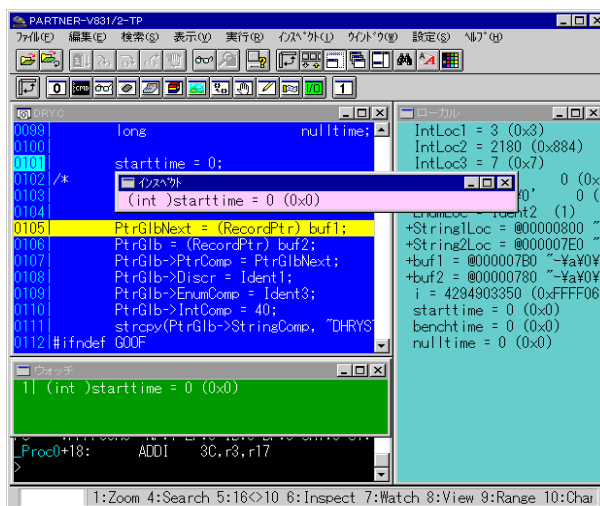


図 6 IntLoc3のウォッチ登録

PARTNER-V831/2-TPの終了

PARTNER-V831/2-TPを終了します。(図 7)

【参照】  ボタン(80頁), [ファイル]-[終了](33頁), Q/EXITコマンド

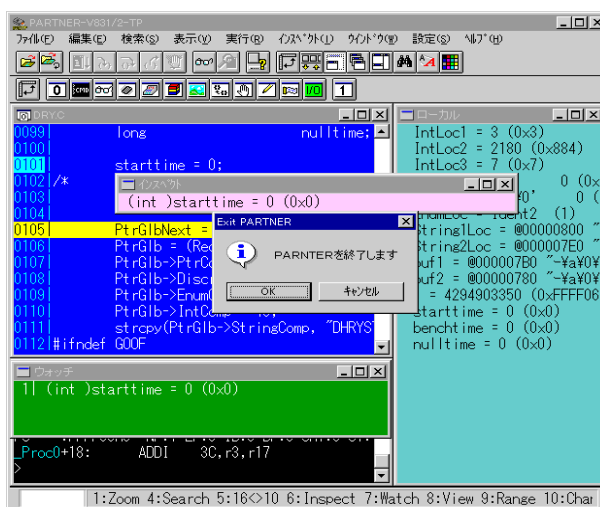


図 7 PARTNER-V831/2-TPの終了確認

2 セットアップ

PARTNER-V831/2-TPを使用するには、ご使用のホストパソコンにPARTNER-V831/2-TPをインストールする必要があります。この章では、PARTNER-V831/2-TP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-V831/2-TPのセットアップを始める前に、RTE-V831-TPの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。（デバッグソフトのセットアップだけであればRTE-V831-TPのインストールがされていなくてもセットアップ可能です。）

2.1.1 RTE-V831-TPの設定

RTE-V831-TPのハードウェア・ユーザズ・マニュアルを参照して接続と設定を行ってください。

2.1.2 RTE-V831-TP用DLLのインストール

RTE-V831-TPのインストールマニュアルを参照してDLLのインストールを行い、使用するICEの種類と接続ポートの設定を行い、接続の確認を実施してください。

2.1.3 ハードディスクの空き領域

PARTNER-V831/2-TPをセットアップする前に十分な空き領域があるかどうか確認してください。PARTNER-V831/2-TPをセットアップするには、ハードディスクに2MB以上の空き領域が必要です。

2.1.4 Windowsの起動

Windows95/NTが起動するかどうか確認してください。

2.2 セットアップの方法

Windows95/NTを起動してPARTNER-V831/2-TPのセットアップを行います。

【注意】PARTNER-V831/2-TPのファイルは、圧縮された状態でCD-ROMに格納されています。このため、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-V831/2-TP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-V831/2-TP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-V831/2-TP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-V831/2-TPのインストール

PARTNER-V831/2-TP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を起動することによって実行されます。

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(V831&2TP)」というショートカットが作成されます。

【注意】PARTNER-V831/2-TP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-V831/2-TP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-V831/2-TP」を一覧から選んで削除します。

空白ページ

3 起動

PARTNER-V831/2-TPを起動するには、PARTNER-V831/2-TPの環境を設定する必要があります。

スタートメニューのRPTSETUP(V831&2TP)をダブルクリックして起動し、PARTNER-V831/2-TPの環境を設定します。

セットアップが完了すればスタートメニューに「RPTSETUP.EXE」へのショートカットRPTSETUP(V831&2TP)が作成されています。

3.1 PARTNER-V831/2-TP環境設定

PARTNER-V831/2-TPを起動するに当たって次の設定を行う必要があります。

- 新規プロジェクトの作成(17頁)
- コンフィグファイルの設定(17頁)
- 起動オプションの設定(18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-V831/2-TPを起動することができます。
RPTSETUP(V831&2TP)は上記の設定を行うためのツールです。

3.1.1 環境設定プログラムの使用方法

RPTSETUP(V831&2TP)は、PARTNER-V831/2-TPの実行環境を設定するツールです。

スタートメニューのRPTSETUP(V831&2TP)を起動し、PARTNER-V831/2-TPの実行環境の作成、変更を行います。(図 8)



図 8 RPTSETUP(V831&2TP)起動画面

『新規に環境を作成する場合』及び『RPTSETUP(V831&2TP)で作成した環境を変更する場合』の説明をします。

なお、メニューにあるモニタの編集、モニタの作成はRPTSETUP(V831&2TP)では使用しません。

新規にPARTNER-V831/2-TPの環境を作成する場合

新規プロジェクトの作成



PARTNER-V831/2-TP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(RPTV831T.KPJ)、コンフィグファイル(RPTV831T.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ディレクトリのほかに、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に指定することも可能です。この場合はコンフィグファイル(RPTV831T.CFG)、プロジェクトファイル(RPTV831T.KPJ)はユーザのワークディレクトリに作成されます。(図 9)

以後、コンフィグファイルの設定、起動オプションの設定は、指定したディレクトリにあるファイルに対してのコマンドとなります。

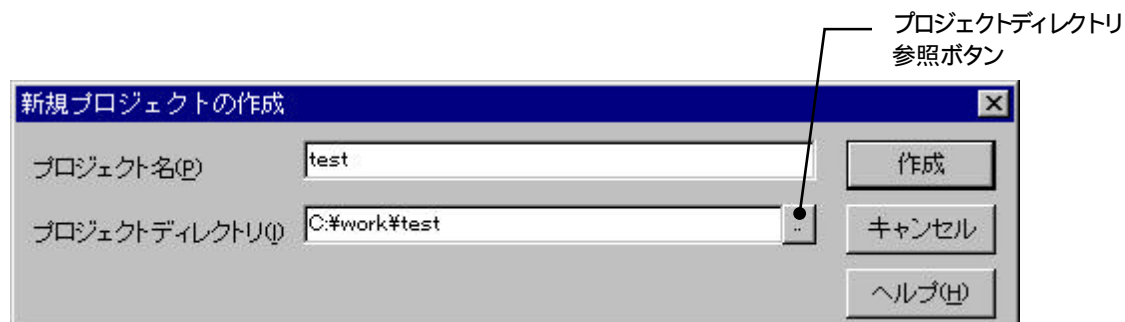


図 9 c:\work\testディレクトリを指定

コンフィグファイルの設定



使用するターゲットボードに合わせて作業ディレクトリにあるコンフィグファイルを編集します。コンフィグレーション項目については『3.2 環境ファイル(RPTV831T.CFG)』(21頁)を参照してユーザのシステムに合わせて変更してください。



起動オプションの設定

PARTNER-V831/2-TPの起動オプションを指定します。起動オプションでは、動作モードやメモリの使用状態を変更します。(図 10)



図 10 [起動オプション]ダイアログボックス



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 11)。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-V831/2-TPが起動します。



図 11 ショートカット名"test"でスタートメニューに登録

既存のプロジェクトを変更

以前に作成したプロジェクトの設定内容を変更するには、以下の手順で行ってください。

プロジェクトを開く



変更するプロジェクトファイル(RPTV831T.KPJ)を選択し、開きます。



設定の変更



『起動オプションの設定』によって、必要な設定を行います。

設定の変更が終われば、PARTNER-V831/2-TPは、変更した内容で起動します。

3.2 環境ファイル(RPTV831T.CFG)

デバッグプログラムPARTNER-V831/2-TPは、起動時に環境ファイル(RPTV831T.CFG)を読み込みます。環境ファイルには、RTE-V831-TPの動作環境を予め設定しておく必要があります。

環境ファイルは、アスキーファイルです。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にご注意ください。

3.2.1 設定項目

PARTNER-V831/2-TPの環境ファイル(RPTV831T.CFG)で設定できる項目は次のとおりです。

- PARTNER-V831/2-TPのコマンドでアクセス可能なメモリ領域

PARTNER-V831/2-TPのコマンドでアクセス可能なメモリ領域(MAP)

PARTNER-V831/2-TP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(最大20領域まで設定可能)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-V831/2-TPのコマンドでアクセス可能であることを宣言します。MAP範囲の指定は、最大20領域まで可能です。複数の範囲を指定する場合は、このフィールドを複数行記述してください。

3.3初期設定コマンド

環境ファイル(RPTV831T.CFG)の内容以外で、RTE-V831-TPの動作環境を予め設定しておく必要がある場合は、次のコマンドで初期化することができます。

&の付いたコマンドはスルーコマンドで、RTE-V831-TPの内部コマンドです。各コマンドについては、コマンドリファレンス(113頁)、または、ハードウェア・ユーザーズ・マニュアルの「内部コマンド」を参照してください。

初期化で使用するコマンド

ENV	各種入力信号のマスクなどのCPUの各種環境を設定します。
ROM	エミュレーションROMの種類、アドレス範囲を設定します。
NC	非メモリキャッシュの領域を設定します。
NCD	非メモリキャッシュの領域を解除します。

これらのコマンドはコマンドウィンドウで入力しますが、PARTNER-V831/2-TPの起動時に自動実行されるINIT.MCRファイル(23頁)に記述しておくくと便利です。

INIT.MCRでの初期化例

INIT.MCRファイルに、以下の例のような初期化コマンドを記述しておくことで、起動毎の初期化を自動で行うことができます。


```
init
    RTE-V831-TP、PARTNER-V831/2-TPを初期化します。
env !nmi
    NMI信号をマスクする、に設定します。
rom FFF80000 80000 4m rom16 bus16
    エミュレーションするROMを16ビットROM、16ビットバス、4Mビットサイズに、
    そしてエミュレーションアドレス範囲をFFF80000H ~ FFFFFFFFH番地に設定し
    ます。
nc 20000 1000
    20000番地から1000バイトの領域をメモリキャッシュの除外エリアに
    指定します。
L sample.exe
    デバッグを行うプログラムsample.exeを読み込みます。
```

3.4 関連ファイル

PARTNER-V831/2-TPが使用するファイルは以下の通りです。

3.4.1 PARTNER-V831/2-TP起動時に必要なファイル

PARTNER-V831/2-TPの起動に必要なファイルには、コンフィグファイル(.CFG)、プロジェクトファイル(.KPJ)があります。また、『3.4.2 PARTNER-V831/2-TP終了時に作成するファイル (24頁)』、INIT.MCRが存在すれば読み込みます。

これらのファイルは、『新規プロジェクトの作成  (17頁)』で指定されたディレクトリから読み込まれます。

コンフィグファイル(RPTV831T.CFG)

コンフィグファイルは、PARTNER-V831/2-TPの設定などを指定するファイルです。現在は設定する項目はありませんが、起動にはこのファイルが必要です

プロジェクトファイル(RPTV831T.KPJ)


プロジェクトファイルは、PARTNER-V831/2-TPの初期状態を設定するファイルです。このファイルには、ウインドウの配置、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-V831/2-TP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、ちょうどMS-DOSのAUTOEXEC.BATファイルに相当します。つまり、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-V831/2-TP起動時に自動的にそれらの処理を行うことができます。

3.4.2 PARTNER-V831/2-TP終了時に作成するファイル

PARTNER-V831/2-TP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、『新規プロジェクトの作成  (17頁)』で指定された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。

3.5 PARTNER-V831/2-TPの起動

PARTNER-V831/2-TPを起動するには、RPTSETUP(V831&2TP)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-V831/2-TPが正常に起動した場合

PARTNER-V831/2-TPを起動すると図 12の画面が表示されます。

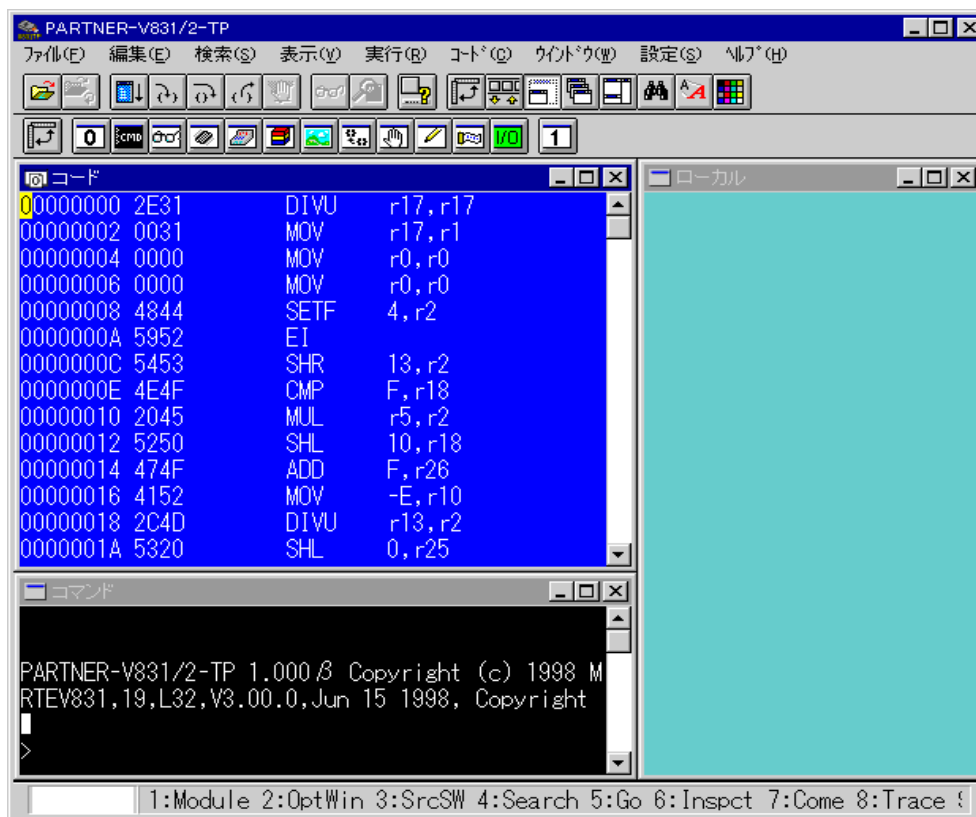


図 12 PARTNER-V831/2-TPの起動画面

3.5.2 PARTNER-V831/2-TPが起動しない場合

起動時に表示するエラーは以下のとおりです。



RTEの電源が入っていないか、ハードウェアの不良が考えられます。電源、接続ケーブル等を確認してください。その後、ChkRTEプログラムを実行してください。



他のアプリケーションがRTEを使用しています。または、異常終了によりRTEのコントロールプログラムは終了していません。その場合にはWindowsを再起動してください。



ホストカードまたはPCカードがインストールされていない可能性があります。正しくインストールされていることを確認しChkRTEプログラムを実行してください。



RTEが接続されていないか、RTEの電源が入っていないことが考えられます。電源、接続ケーブルなどを確認してください。その後、ChkRTEプログラムを実行してください。



システムの初期化ができません。RTEに電源が入っていない可能性があります。電源の接続を確認し、ChkRTEプログラムを実行してください。



何らかの異常終了によりRTEのコントロールプログラムは終了していません。Windowsを再起動してください。



インストールに不備があり、必要なDLLが見つかりません。ソフトウェアのインストールが正しくありませんので、RTEのSetUp CD-ROMで再インストールしてください。その後、必ずChkRTEプログラムを実行してください。

空白ページ

4 ウィンドウコマンド

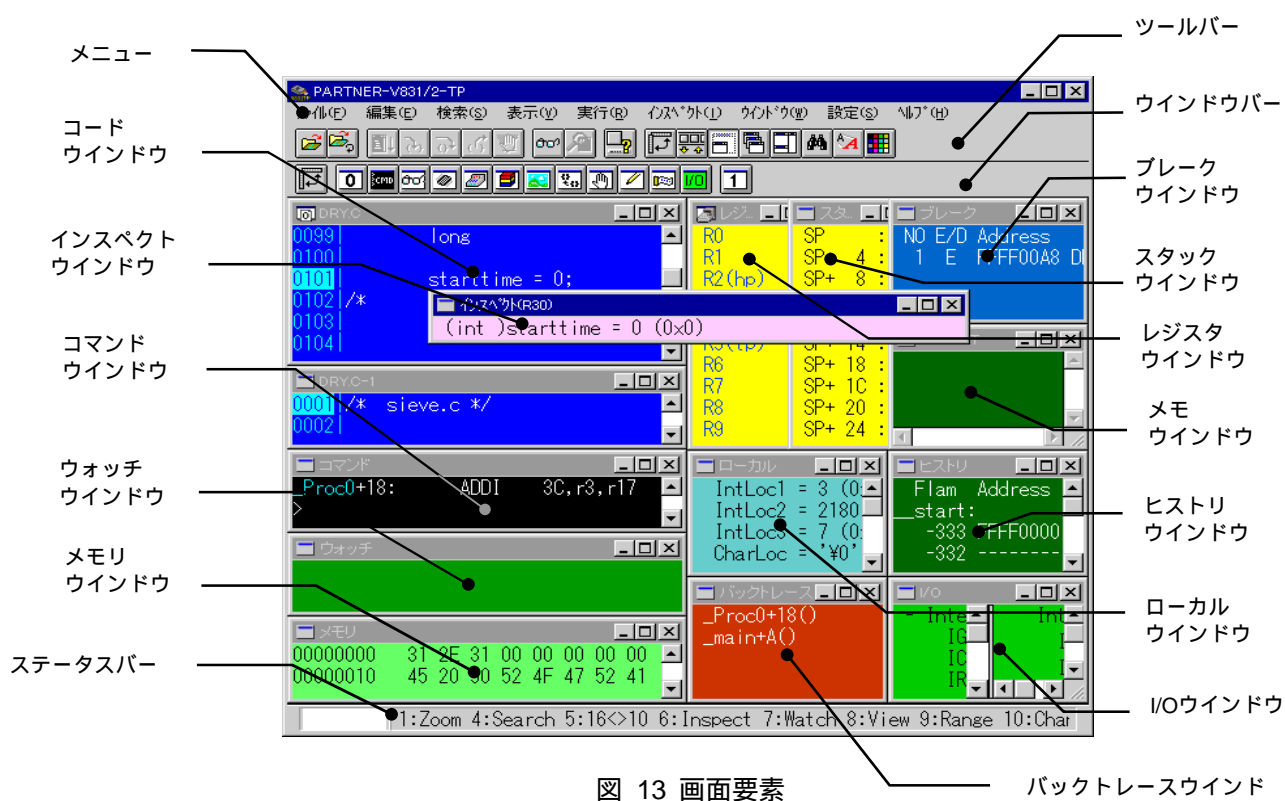
Windows版デバッガPARTNER-V831/2-TPは、MDI(Multi Document Interface)を採用し、各チャイルドウィンドウにデバッグ時に必要な情報を表示して、DOS版のPARTNER以上の情報量を提供します。また、ショートカットキー、メニュー、ツールバー、ステータスバー、ダイアログボックスの採用により、スピーディーな操作環境を提供します。

4.1 画面構成

ここでは、メニュー、ツールバー、ステータスバー、各デバッグウインドウなど PARTNER-V831/2-TPの主な画面の構成要素を説明します。

4.1.1 構成要素

ウインドウ要素を図 13で表します。



メインウインドウ

PARTNER-V831/2-TPのメインウインドウです。このウインドウにメニューバー、ツールバー、ステータスバーが存在します。

コードウインドウ

ソース表示、逆アセンブル表示をするウインドウです。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。このウインドウは2つあります。

コマンドウインドウ



ダイアログコマンドを入力、実行結果の表示を行うウインドウです。

メモリウインドウ



メモリ空間のダンプ表示を行うウインドウです。CPUブレーク時に最新のデータに変更されます。

レジスタウインドウ



CPUのレジスタを表示するウインドウです。CPUブレーク時に最新のデータに変更されます。

スタックウインドウ



スタックの内容を表示します。CPUブレーク時に最新のデータに変更されます。

ローカルウインドウ



Cの現在のプログラムカウンタのある関数のローカル変数の内容を表示します。CPUブレーク時に最新のデータに変更されます。

ウォッチウインドウ



ウォッチ登録データを表示します。CPUブレーク時に最新のデータに変更されま
す。

バックトレースウインドウ



Cの関数のバックトレース表示をします。CPUブレーク時に最新のデータに変更
されます。

ブレークウインドウ



現在、設定されているブレークポイントの表示をします。

メモウインドウ



デバッグ中に使用する簡易エディタです。PARTNER-V831/2-TP終了時に内容を保
存します。

インスペクトウインドウ

変数の内容をインスペクト(調査)し、表示します。

ヒストリウインドウ



リアルタイムトレースメモリをダンプ表示もしくは、逆アセンブル表示します。

I/Oウインドウ



I/Oレジスタの内容を変更できます。

4.1.2 画面表示

ウイ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インドウの大きさ、フォントの大きさを変更する必要があります。横スクロール可能なウインドウは、コードウインドウ、メモリウインドウ、メモウインドウです。

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(>)でその行にあることを示します。(図 14)検索結果を表示するには、ウインドウの大きさ、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。

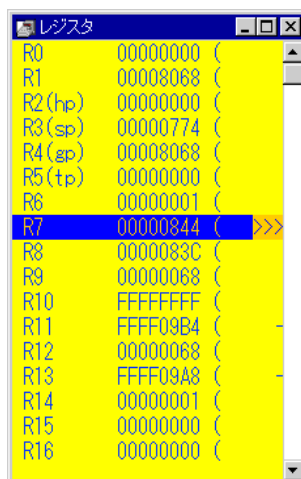
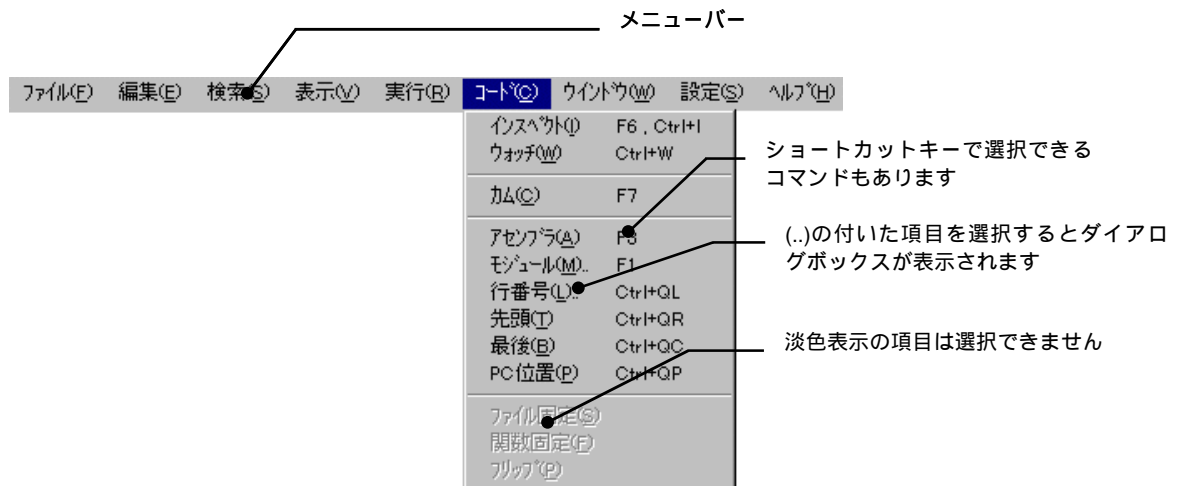


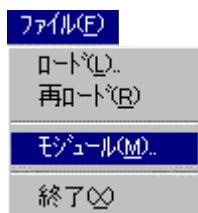
図 14 非表示部分に検索文字列がある場合

4.2 メニュー

メニューには、PARTNER-V831/2-TP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、コードウインドウもしくはコマンドウイ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)	デバッグプログラムのロードの為にダイアログボックスを表示 (『5.1.1 ファイルを開くダイアログボックス』86頁参照)
再ロード(R)	デバッグプログラムの再ロード
モジュール(M)	ロードされたデバッグプログラムに含まれるモジュールを選択する為にダイアログボックスを表示します (『5.1.8 モジュールダイアログボックス』90頁参照)
終了(X)	PARTNER-V831/2-TP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]メニューにはウイ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

メモウイ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)	ウイ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)	メモウイ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]メニューにはウイ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(『5.1.5 文字列検索ダイアログボックス』88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(『4.20 ステータスバー』83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(『4.18 ツールバー』79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(『4.19 ウィンドウバー』82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(B)

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(P)	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]はコードウインドウにフォーカスがある場合のみ実行できます。

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から現在カーソルのある行まで実行します
トレース(T)	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
ステップ(S)	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
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-V831/2-TP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。
CPU環境設定(V)	CPUの環境値を設定します。

4.2.6 各ウインドウのローカルメニュー

ローカルメニューには、各ウインドウで操作できるコマンドが納められています。現在フォーカスのあるウインドウのローカルメニューがメニューバーに登録されます。各ローカルメニューはマウスの右ボタンクリックによって表示/選択することができます。

コードメニュー

コード(C)	
インサ [^] 外(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インドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.5.2 コードウインドウのローカルメニュー(52頁)』を参照してください。

コマンドメニュー

コマンド(C)	
ポート(L)..	
再ポート(R)	
貼り付け(P)	Shift+Ins
履歴(H)..	Shift+F2
シンボル拡張(S)..	Shift+F6

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

このメニューコマンドの内容は『コマンドウインドウのローカルメニュー (56頁)』を参照してください。

メニュー

メモリメニュー

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
ショートワード(S)	S, F
ロングワード(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。

このメニューコマンドの内容は『4.7.2 メモリウィンドウのローカルメニュー(58頁)』を参照してください。

レジスタメニュー

レジスタ(R)

初期化(Q)	F3
インクリメント(+)	
デクリメント(-)	
値の変更(C)	Enter
10進数表示(D)	F9
シンボル表示(S)	F6
メモリ表示(M)	F7

[レジスタ]メニューにはレジスタ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。

このメニューコマンドの内容は『4.8.2 レジスタウィンドウのローカルメニュー(61頁)』を参照してください。

スタックメニュー

スタック(S)

✓ シンボル表示(S)	F6
✓ アドレス表示(A)	F7

[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。

このメニューコマンドの内容は『4.9.2 スタックウィンドウのローカルメニュー(62頁)』を参照してください。

ローカルメニュー

ローカル(L)

インスペクタ(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。

このメニューコマンドの内容は『4.10.2 ローカルウィンドウのローカルメニュー(64頁)』を参照してください。

ウォッチメニュー

ウォッチ(W)

インスペクタ(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(Q)	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。

このメニューコマンドの内容は『4.12.2 ウォッチウィンドウのローカルメニュー(68頁)』を参照してください。

バックトレースメニュー

バックトレース(B)

インスペクタ(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。

このメニューコマンドの内容は『4.11.2 バックトレースウィンドウのローカルメニュー(66頁)』を参照してください。

メニュー

ブレイクメニュー

ブレイク(B)	
ブレイク設定(A)..	Ins
許可(E)	F9,Enter
削除(D)	Del
全て削除(D)	F8
設定のセーブ(S)	F4
設定のリート(L)	F6
アクセスブレイク(E)..	F7

[ブレイク]メニューにはブレイクウインドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.13.2 ブレイクウインドウのローカルメニュー(70頁)』を参照してください。

メモメニュー

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インドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.16.2 メモウインドウのローカルメニュー(76頁)』を参照してください。

インスペクトメニュー

インスペクト(I)	
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インドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.15.2 インスペクトウインドウのローカルメニュー(74頁)』を参照してください。

ヒストリメニュー

ヒストリ(H)

アセンブラ(A)	F3
ジャンプ(J)..	F6
ヒストリストップ(S)	F7

トリガアドレス(T)..
トレースモート(M)..

[ヒストリ]メニューにはヒストリウインドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.14.2 ヒストリウインドウのローカルメニュー(72頁)』を参照してください。

I/Oメニュー

I/O(O)

✓ 略称(N)
名称(N)

説明文の表示(H)

定義ファイルの読み込み(L)

[I/O]メニューにはI/Oウインドウの制御に関するコマンドが収められています。

このメニューコマンドの内容は『4.17.2 I/Oウインドウのローカルメニュー(78頁)』を参照してください。

4.2.7 ウィンドウメニュー

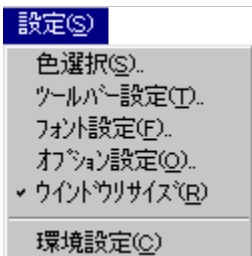
ウィンドウ(W)

カスケード(C)	
タイル(T)	
アイコン整列(I)	
ユーザ配置1(1)	Ctrl+1
ユーザ配置2(2)	Ctrl+2
ユーザ配置3(3)	Ctrl+3
ウィンドウ配置保存(S)...	
1 メモリ	
2 スタック	
3 ローカル	
4 バックトレース	
5 ブレーク	
6 Memo.PT	
7 ヒストリ	
8 コマンド	
9 SAMPLE.C[start0]	
その他のウィンドウ(M)...	

[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-V831/2-TP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記憶します

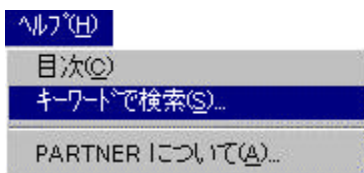
4.2.8 設定メニュー



[設定]メニューにはPARTNER-V831/2-TP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(『5.1.4 色指定ダイアログボックス』88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(『5.1.3 ツールバーの設定ダイアログボックス』87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(『5.1.2 フォントの指定ダイアログボックス』87頁参照)
オプション設定(O)	PARTNER-V831/2-TPの各種設定を指定するダイアログボックスを表示します
ウィンドウリサイズ(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-V831/2-TPは一旦終了します

4.2.9 ヘルプメニュー



[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-V831/2-TPのヘルプの目次を表示します
キーワードで検索(S)	PARTNER-V831/2-TPのヘルプをキーワードで検索します
PARTNERについて(A)	PARTNER-V831/2-TP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-V831/2-TPのコマンドの中で、頻繁に実行される操作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 各ウイ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、各ウイ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インドウの[最大表示]/[元のサイズに戻す]切り替えをします <small>[注]コードウインドウ,コマンドウインドウは別のコマンドが定義されています</small>
F2 CTRL+O	アクティブウインドウを1番下に移動し、次のウインドウをアクティブにします
CTRL+B	一つ前のウインドウをアクティブにします(F2,CTRL+Oの逆)
F4 CTRL+Q F	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(『5.1.5 文字列検索ダイアログボックス』88頁参照)
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 ROLL UP	一画面分下を表示します
CTRL+D	一単位分右にカーソルを移動します
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 ROLL DOWN	一画面分上を表示します
CTRL+S	一単位分左にカーソルを移動します
CTRL+X	一単位分(一行分)下にカーソルを移動します
CTRL+INS	フォーカスのあるウイ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	コードウインドウ0の表示/アイコン化をします
CTRL+GRPH+F2 CTRL+GRPH+C	コマンドウインドウの表示/アイコン化をします
CTRL+GRPH+F3 CTRL+GRPH+W	ウォッチウインドウの表示/アイコン化をします
CTRL+GRPH+F4 CTRL+GRPH+M	メモリウインドウの表示/アイコン化をします
CTRL+GRPH+F5 CTRL+GRPH+R	レジスタウインドウの表示/アイコン化をします
CTRL+GRPH+F6	スタックウインドウの表示/アイコン化をします
CTRL+GRPH+F7 CTRL+GRPH+L	ローカルウインドウの表示/アイコン化をします
CTRL+GRPH+F8 CTRL+GRPH+T	バックトレースウインドウの表示/アイコン化をします
CTRL+GRPH+F9 CTRL+GRPH+B	ブレークウインドウの表示/アイコン化をします
CTRL+GRPH+F10 CTRL+GRPH+E	メモウインドウの表示/アイコン化をします
CTRL+GRPH+H	ヒストリウインドウの表示/アイコン化をします
CTRL+1	ユーザ配置1の呼び出し
CTRL+2	ユーザ配置2の呼び出し
CTRL+3	ユーザ配置3の呼び出し
GRPH+F10	現在フォーカスのあるウインドウのローカルメニューを表示します

【注意】DOS/V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	4.5.1 コードウィンドウのショートカットキー	50
コマンドウィンドウ	4.6.1 コマンドウィンドウのショートカットキー	54
メモリウィンドウ	4.7.1 メモリウィンドウのショートカットキー	57
レジスタウィンドウ	4.8.1 レジスタウィンドウのショートカットキー	59
スタックウィンドウ	4.9.1 スタックウィンドウのショートカットキー	62
ローカルウィンドウ	4.10.1 ローカルウィンドウのショートカットキー	63
バックトレースウィンドウ	4.11.1 バックトレースウィンドウのショートカットキー	65
ウォッチウィンドウ	4.12.1 ウォッチウィンドウのショートカットキー	67
ブレークウィンドウ	4.13.1 ブレークウィンドウのショートカットキー	69
インスペクトウィンドウ	4.15.1 インスペクトウィンドウのショートカットキー	73
メモウィンドウ	4.16.1 メモウィンドウのショートカットキー	75
ヒストリウィンドウ	4.14.1 ヒストリウィンドウのショートカットキー	71

4.4 マウス操作

ウインドウ内でのマウス操作は、各ウインドウ共通の操作と独自の操作があります。

4.4.1 各ウインドウ共通のマウス操作

各ウイ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インドウのウインドウ範囲内で右ボタンをクリックすると、そのウイ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可能です。(『4.2.6 各ウインドウのローカルメニュー』37頁参照)

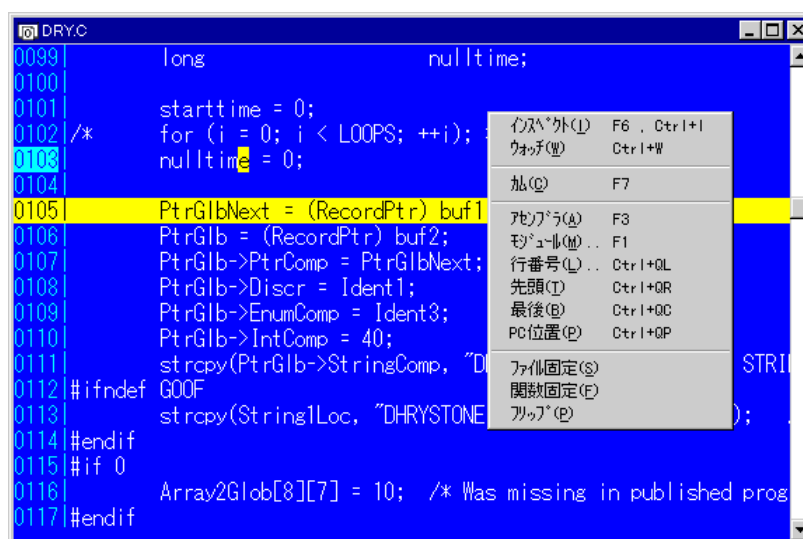


図 15 コードウインドウのローカルメニュー

左ボタンドラッグ

ウインドウ内で左ボタンで文字列をドラッグし、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。

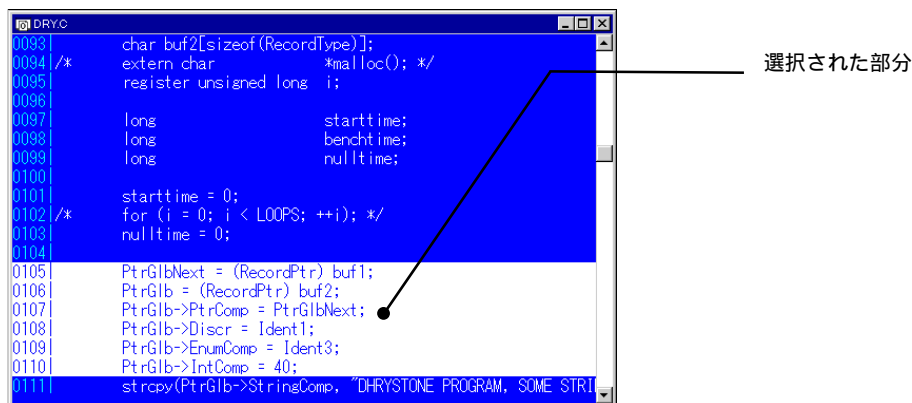


図 16 コピー範囲の選択中

【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	4.5.3 コードウィンドウでのマウス操作	53
メモリウィンドウ	4.7.3 メモリウィンドウでのマウス操作	58
レジスタウィンドウ	4.8.3 レジスタウィンドウでのマウス操作	61
ローカルウィンドウ	4.10.3 ローカルウィンドウでのマウス操作	64
バックトレースウィンドウ	4.11.3 バックトレースウィンドウでのマウス操作	66
ウォッチウィンドウ	4.12.3 ウォッチウィンドウでのマウス操作	68
ブレークウィンドウ	4.13.3 ブレークウィンドウでのマウス操作	70
インスペクトウィンドウ	4.15.3 インスペクトウィンドウでのマウス操作	74

4.5 コードウインドウ

コードウインドウは、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。また、コードウインドウは二つ存在し、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。

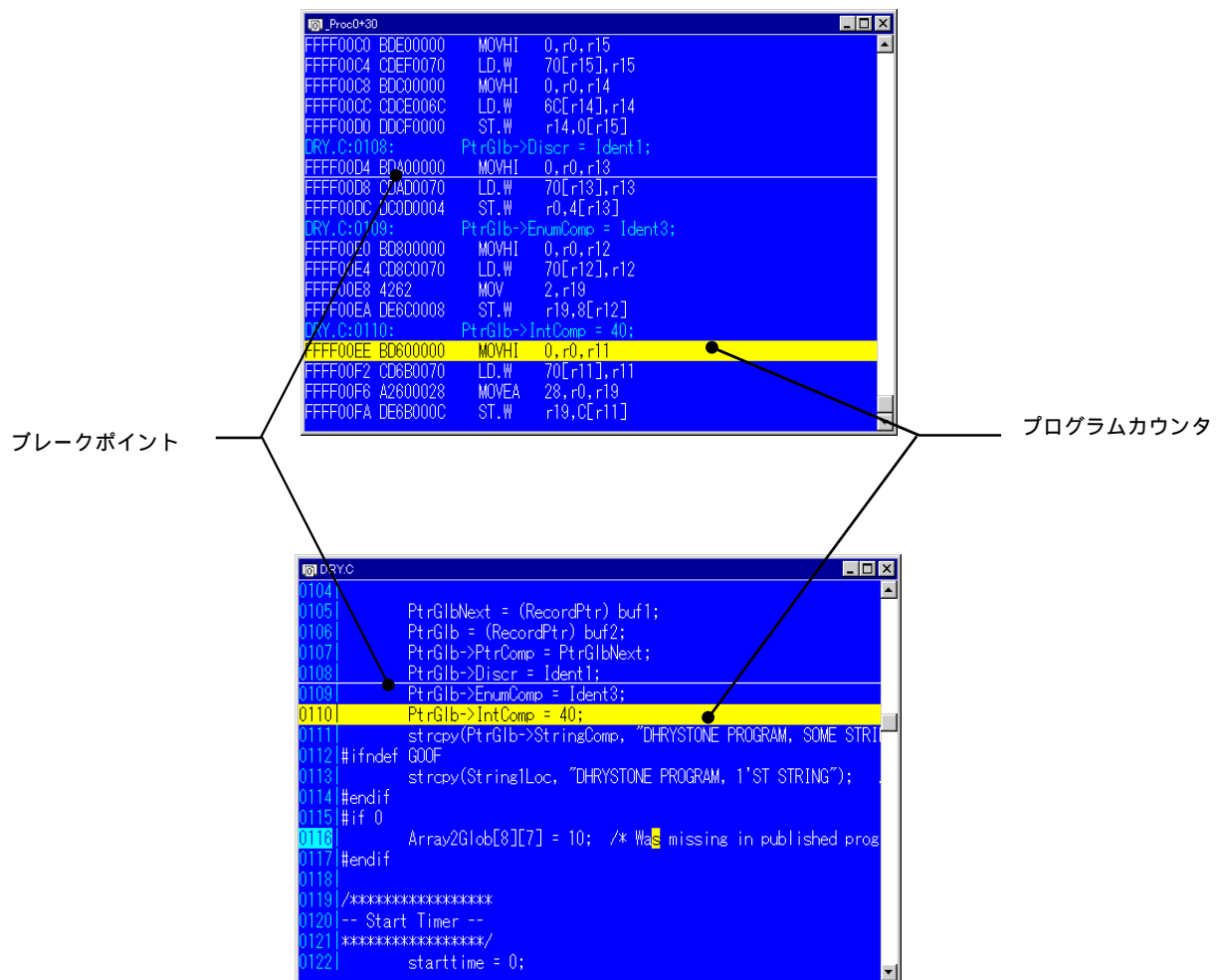


図 17 コードウインドウ

4.5.1 コードウインドウのショートカットキー

コードウイ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イ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(『5.1.8 モジュールダイアログボックス』90頁参照)	
F3	コードウイ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のインスペクトをします	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	コードウインドウの[最大化]/[元のサイズに戻す]を切り替えます	
CTRL+F2	次のウインドウにフォーカスを移動します	
CTRL+F3	コードウイ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 表示は現在のカーソル行を基準に表示するように切り替えます	Vコマンド Uコマンド
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イ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インドウにビュー表示します	VALコマンド ?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イ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	起動時のウイ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 ニーモニック」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	
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+???については『4.6.1 コマンドウインドウのショートカットキー(54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。

4.5.2 コードウインドウのローカルメニュー

コード(C)	
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)	
フリップ(P)	

[コード]メニューにはコードウイ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イ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(『5.1.6 インスペクト設定ダイアログボックス』89頁参照)
ウォッチ(W)	カーソル位置の変数をウォッチウイ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(『5.1.7 ウォッチ設定ダイアログボックス』89頁参照)
カム(C)	ユーザプログラムを現在のPCから現在カーソルのある行まで実行します
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)	ロードされたデバッグ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(『5.1.8 モジュールダイアログボックス』90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(『5.1.9 行番号指定ダイアログボックス』90頁, 『5.1.10 アドレス指定ダイアログボックス(コード)』91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

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イ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イ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イ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イ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インドウのフォーカスがもう一つのコードウイ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つのコードウインドウが表示されている場合にのみ有効になります。

4.5.3 コードウインドウでのマウス操作

コードウイ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(36頁参照)	CTRL+右クリック	

4.6 コマンドウインドウ

コマンドウインドウは、ダイアログコマンドの入力、実行結果の出力をします。

コマンドウイ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。

```

コマンド
>g
>
R0 / R8 R1 / R9 R2 /R10 R3 /R11 R4 /R12 R5 /R
R16/R24 R17/R25 R18/R26 R19/R27 R20/R28 R21/R
R 0- 7:00000000 4E070000 00000000 4E070A28 00000000 00000
R 8-15:00000000 00000000 00000003 C0000000 4FFF171A 00000
R16-23:00000001 00000003 00000000 00000001 00000000 00000
R24-31:00000000 00000000 00000000 45002000 00000008 00001
PC      :4FFF1330 NP:1 EP:0 ID:0 DP:0 SAT:0 CY:0 OV:0 S:0
      ADD      -1, r6
>
    
```

図 18 コマンドウインドウ

4.6.1 コマンドウインドウのショートカットキー

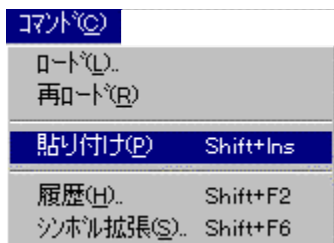
コマンドウインドウでは以下のショートカットキーにより各種コマンドが実行できます。このショートカットキーはコマンドウインドウにフォーカスがある場合に適用されます。

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(C1)
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(『5.1.11 コマンド履歴ダイアログボックス』91頁参照)
SHIFT+F3	直前に入力したコマンドの現在のカーソル位置以降の文字列をコピーします(CA)
SHIFT+F4 CTRL+A	カーソルを行頭に移動します(Ln Top)
SHIFT+F5 CTRL+F	カーソルを行末に移動します(Ln Bot)

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(『5.1.12 シンボル拡張ダイアログボックス』92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インドウのショートカットと同様です(『4.5.1 コードウインドウのショートカットキー(50頁)』参照)。ただし、カーソル位置に関するコマンドは使用できません。

4.6.2 コマンドウインドウのローカルメニュー



[コマンド]メニューにはコマンドウイ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)	デバッグプログラムのロードの為のダイアログボックスを表示
再ロード(R)	デバッグプログラムの再ロード
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(『5.1.11 コマンド履歴ダイアログボックス』91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(『5.1.12 シンボル拡張ダイアログボックス』92頁参照)

4.7 メモリウインドウ

メモリウインドウには、メモリの内容を各種表示形式で表示します。

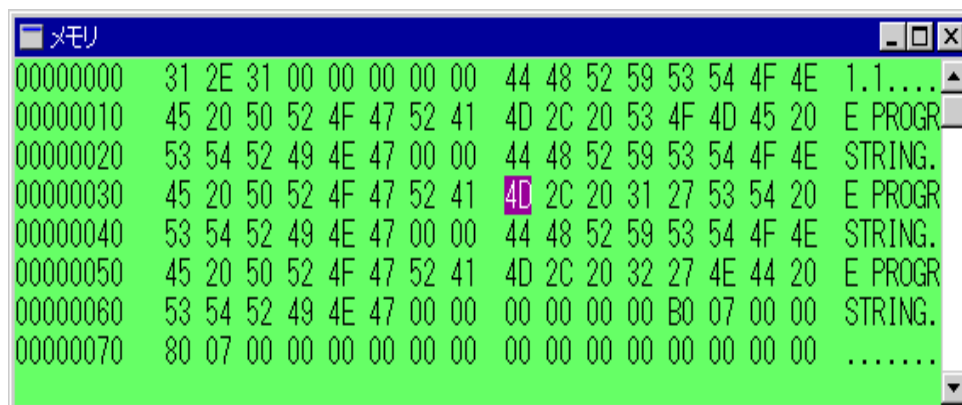


図 19 メモリウインドウ

4.7.1 メモリウインドウのショートカットキー

メモリウイ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イ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 (『5.1.13 アドレス指定ダイアログボックス(メモリ)』92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 (『5.1.14 データ設定ダイアログボックス』93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

4.7.2 メモリウイ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インドウの制御に関するコマンドが収められています。

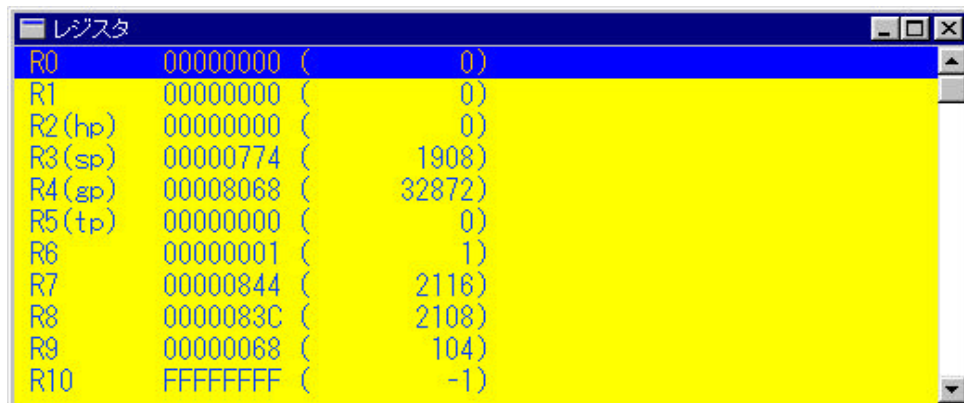
メニュー項目	機能
バイト表示(B)	メモリデータをバイト表示に指定します
ワード表示(W)	メモリデータをワード表示に指定します
ダブルワード表示(D)	メモリデータをダブルワード表示に指定します
ショートフロート(S)	メモリデータをショートフロート表示に指定します
ロングフロート(L)	メモリデータをロングフロート表示に指定します
ASCII表示(C)	メモリデータをASCII表示に指定します
16進数表示(H)	メモリデータを16進数表示に指定します
10進数表示(D)	メモリデータを10進数表示に指定します
アドレス設定(A)	表示開始アドレスを指定するダイアログボックスを表示します(『5.1.13 アドレス指定ダイアログボックス(メモリ)』92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(『5.1.14 データ設定ダイアログボックス』93頁参照)

4.7.3 メモリウイ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。(『5.1.14 データ設定ダイアログボックス』93頁参照)

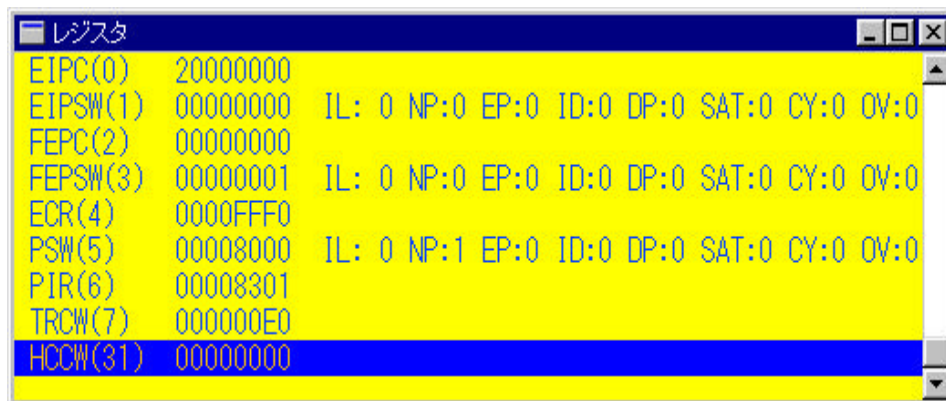
4.8 レジスタウインドウ

レジスタウインドウには、r0～r31,PCレジスタ(図 20)、システムレジスタ(図 21)を表示します。



レジスタ	値	コメント
R0	00000000	(0)
R1	00000000	(0)
R2 (hp)	00000000	(0)
R3 (sp)	00000774	(1908)
R4 (gp)	00008068	(32872)
R5 (tp)	00000000	(0)
R6	00000001	(1)
R7	00000844	(2116)
R8	0000083C	(2108)
R9	00000068	(104)
R10	FFFFFFFF	(-1)

図 20 CPUレジスタ



レジスタ	値	コメント
EIPC(0)	20000000	
EIPSW(1)	00000000	IL: 0 NP:0 EP:0 ID:0 DP:0 SAT:0 CY:0 OV:0
FEPC(2)	00000000	
FEPSW(3)	00000001	IL: 0 NP:0 EP:0 ID:0 DP:0 SAT:0 CY:0 OV:0
ECR(4)	0000FFF0	
PSW(5)	00008000	IL: 0 NP:1 EP:0 ID:0 DP:0 SAT:0 CY:0 OV:0
PIR(6)	00008301	
TRCW(7)	000000E0	
HCCW(31)	00000000	

図 21 システムレジスタ

4.8.1 レジスタウインドウのショートカットキー

レジスタウイ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レジスタウイ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	カーソル位置のレジスタまたはフラグの初期化をします
F6	レジスタ値のシンボル表示をします
F7	レジスタ値が指しているメモリの内容を表示します
F9	レジスタ値を10進数で表示します
Enter	カーソル位置のレジスタ値またはフラグの値を変更するダイアログボックスを表示します(『5.1.15 レジスタの設定ダイアログボックス』93頁参照)

4.8.2 レジスタウインドウのローカルメニュー

レジスタ(R)

初期化(Z)	F3
インクリメント(+)	
デクリメント(-)	
値の変更(C)	Enter
<hr/>	
10進数表示(D)	F9
シンボル表示(S)	F6
メモリ表示(M)	F7

[レジスタ]メニューにはレジスタウイ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
初期化(Z)	カーソル位置のレジスタの値を0にします
インクリメント(+)	カーソル位置のレジスタの値をインクリメントします
デクリメント(-)	カーソル位置のレジスタの値をデクリメントします
値の変更(C)	カーソル位置のレジスタの値を変更するダイアログボックスを表示します(『5.1.15 レジスタの設定ダイアログボックス』93頁参照)
10進数表示(D)	レジスタの値を10進数で表示 / 非表示の切り替えをします
シンボル表示(S)	レジスタの値のシンボルを表示 / 非表示の切り替えをします
メモリ表示(M)	レジスタの値が指しているメモリの内容の表示 / 非表示の切り替えをします

4.8.3 レジスタウインドウでのマウス操作

レジスタまたはフラグ部分を左ダブルクリックすると、値変更を行うダイアログボックスが表示されます。(『5.1.15 レジスタの設定ダイアログボックス』93頁参照)

4.9 スタックウインドウ

スタックウインドウには、現在のスタックメモリの内容を表示します。

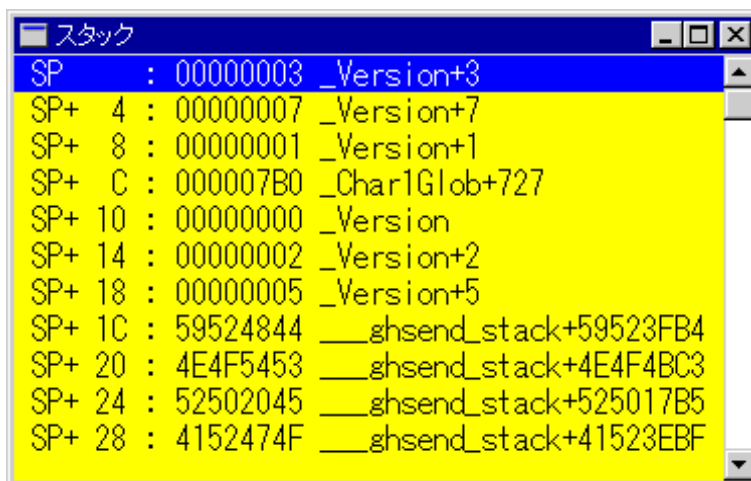


図 22 スタックウインドウ

4.9.1 スタックウインドウのショートカットキー

スタックウイ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イ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表示
F7	スタックの実メモリアドレスの表示

4.9.2 スタックウイ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イ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

4.10 ローカルウインドウ

ローカルウインドウには、現在プログラムカウンタ(PC)のあ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”-”が表示されています。

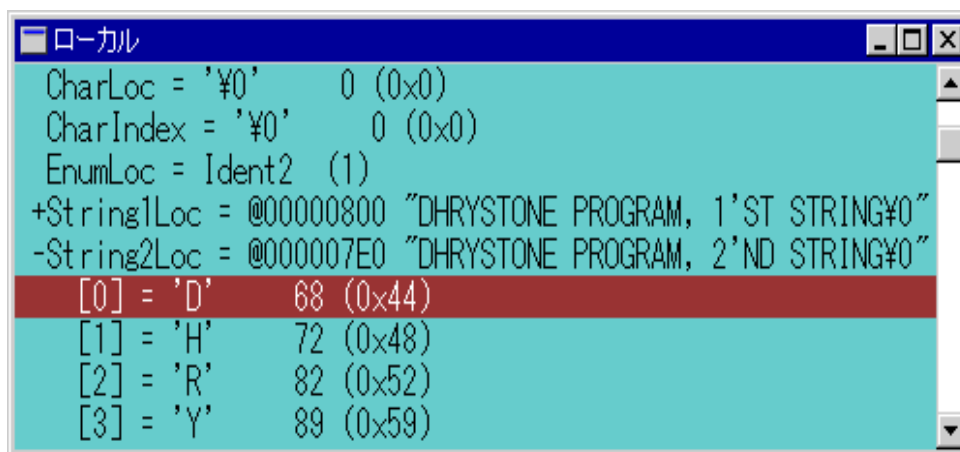


図 23 ローカルウインドウ

4.10.1 ローカルウインドウのショートカットキー

ローカルウイ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イ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イ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

4.10.2 ローカルウイ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イ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イ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インドウに登録します
要素表示(+)	配列や構造体など要素があるローカル変数の内容の表示 / 非表示切り替えを行います
オフセット(O)	ローカル変数のオフセット値の表示 / 非表示切り替えを行います

4.10.3 ローカルウイ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。

4.11 バックトレースウインドウ

バックトレースウインドウには、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。

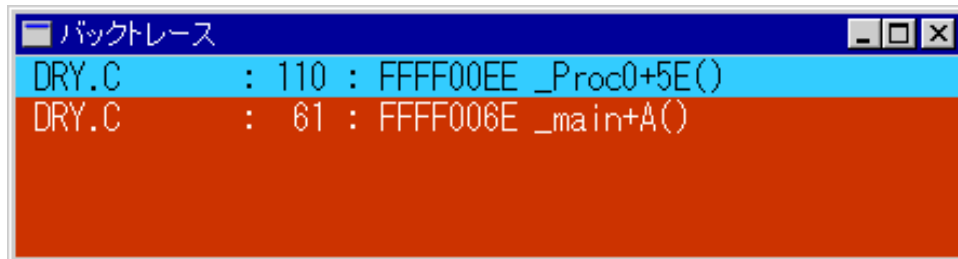


図 24 バックトレースウインドウ

4.11.1 バックトレースウインドウのショートカットキー

バックトレースウイ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イ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インドウに行います
F7	ウイ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	ウイ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

4.11.2 バックトレースウイ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イ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イ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

4.11.3 バックトレースウイ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表示をコードウインドウに行います。

4.12 ウォッチウインドウ

ウォッチウインドウには、ウォッチ登録した変数の表示を行います。

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”-”が表示されています。

```

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 25 ウォッチウインドウ

4.12.1 ウォッチウインドウのショートカットキー

ウォッチウイ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イ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

4.12.2 ウォッチウイ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イ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イ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イ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の内容の表示 / 非表示の切り替えを行います
ウォッチ追加(A)	新しくウォッチウイ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インドウから削除します
全て取り消し(L)	ウォッチウインドウに登録されている全ての変数を削除します

4.12.3 ウォッチウイ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。

4.13 ブ레이크ウインドウ

ブ레이크ウインドウには、現在設定されているブ레이크ポイントを表示します。

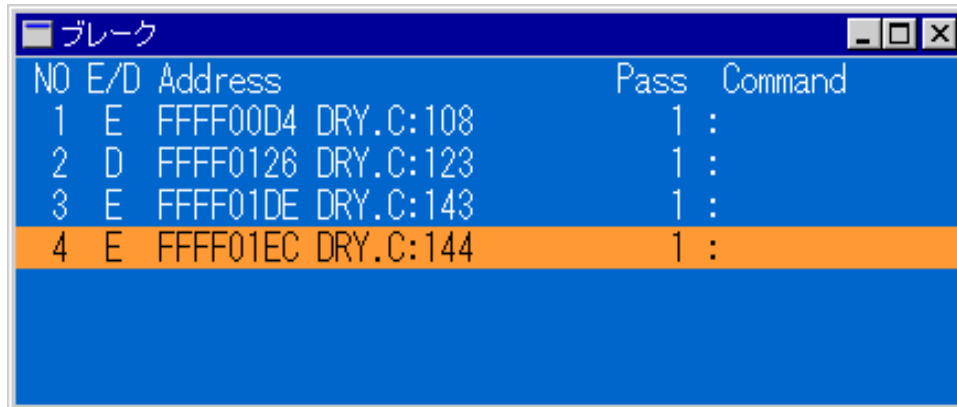


図 26 ブ레이크ウインドウ

4.13.1 ブ레이크ウインドウのショートカットキー

ブ레이크ウイ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ブ레이크ウイ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	全てのブ레이크ポイントを削除します
F4	現在のブ레이크ポイントの設定を保存します
F6	F4で保存したブ레이크ポイントの設定を呼び出します
F7	アクセスブ레이크の設定をします。(『5.1.17 アクセスブ레이크の設定ダイアログボックス』94頁参照)
F9 Enter	現在選択されているブ레이크ポイントの許可/禁止を切り替えます
INS	ブ레이크ポイントの設定ダイアログボックスを表示します(『5.1.16 ブ레이크ポイントの設定ダイアログボックス』94頁参照)
DEL	現在選択されているブ레이크ポイントを削除します

4.13.2 ブレークウインドウのローカルメニュー

ブレーク(B)

ブレーク設定(A)..	Ins
許可(E)	F9,Enter
削除(C)	Del
全て削除(I)	F8
設定のセーブ(S)	F4
設定のロード(L)	F6
アクセスブレーク(E)..	F7

[ブレーク]メニューにはブレークウイ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
ブレーク設定(A)	新規にブレークポイントを設定するダイアログボックスを表示します(『5.1.16 ブレークポイントの設定ダイアログボックス』94頁参照)
禁止(D)/許可(E)	カーソルのあるブレークポイントの許可/禁止の切り替えをします
削除(C)	カーソルのあるブレークポイントを削除します
全て削除(I)	全てのブレークポイントを削除します
設定のセーブ(S)	現在のブレークポイントの設定を保管します
設定のロード(L)	以前保管されていたブレークポイントの設定をロードします
アクセスブレーク(E)	アクセスブレークポイントの設定をします(『5.1.17 アクセスブレークの設定ダイアログボックス』94頁参照)

4.13.3 ブレークウインドウでのマウス操作

ブレークポイント部分を左ダブルクリックすると、ブレークポイントの有効/無効を切り替えます。

4.14 ヒストリウインドウ

ヒストリウインドウには、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。

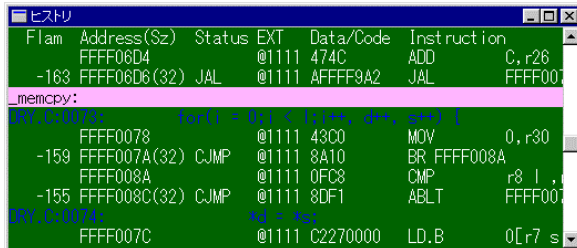


図 27 ヒストリ(逆アセンブル表示)

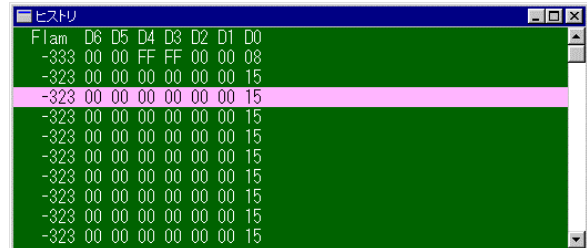


図 28 ヒストリ(ダンプ表示)

4.14.1 ヒストリウインドウのショートカットキー

ヒストリウイ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ヒストリウイ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	表示形式を16進ダンプ/逆アセンブルに切り替えます
F6	表示するフレーム番号のダイアログボックスを表示します(『5.1.23 フレーム番号指定ダイアログボックス』98頁参照)
F7	ユーザプログラム実行中、リアルタイムトレースの実行/停止を切り替えます

4.14.2 ヒストリウインドウのローカルメニュー

ヒストリ(H)	
アセンブラ(A)	F3
ジャンプ(J)	F6
ヒストリストップ(S)	F7
トリガアドレス(T)	
トレースモード(M)	

[ヒストリ]メニューにはヒストリウイ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
ダンプ表示(D)/逆アセンブル表示(A)	ヒストリの表示形式をダンプと逆アセンブルに切り替えます
ジャンプ(J)	表示フレームを指定するダイアログボックスを表示します(『5.1.19 トレースアドレス設定ダイアログボックス』95頁参照)
ヒストリ実行(S)/ヒストリストップ(T)	ヒストリの実行/停止を切り替えます
トリガアドレス(T)	トリガアドレス、データトレースアドレスを指定するダイアログボックスを表示します。(『5.1.19 トレースアドレス設定ダイアログボックス』95頁参照)
トレースモード(M)	トレースモードを指定するダイアログボックスを表示します。(『5.1.20 トレースモード設定ダイアログボックス』96頁参照)

4.15 インスペクトウインドウ

インスペクトウインドウには、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。

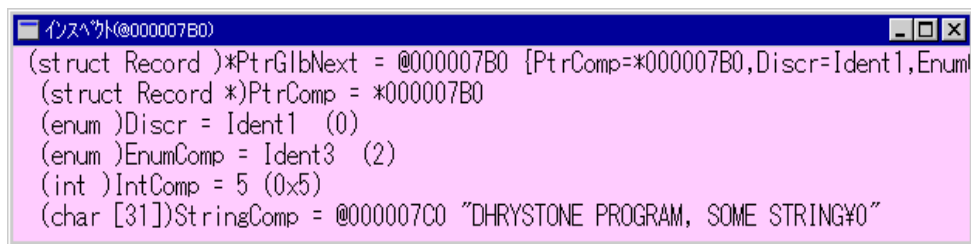


図 29 インスペクトウインドウ

4.15.1 インスペクトウインドウのショートカットキー

インスペクトウイ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イ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イ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イ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イ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

4.15.2 インスペクトウインドウのローカルメニュー

インスペクト	
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イ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インドウを開きます
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イ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インドウにビュー表示します
範囲(R)	インスペクトウイ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

4.15.3 インスペクトウイ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インドウに表示します。

4.16 メモウインドウ

メモウインドウは、簡易エディタです。このウインドウはクリップボードをサポートしており、各ウイ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イ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可能です。また、メモウインドウにPARTNERマクロコマンドを入力しておき、コマンドウインドウに入力することができます。

このウインドウの内容は、PARTNER-V831/2-TPの終了時にファイル(MEMO.PT)に保存され、起動時に読み込まれます。

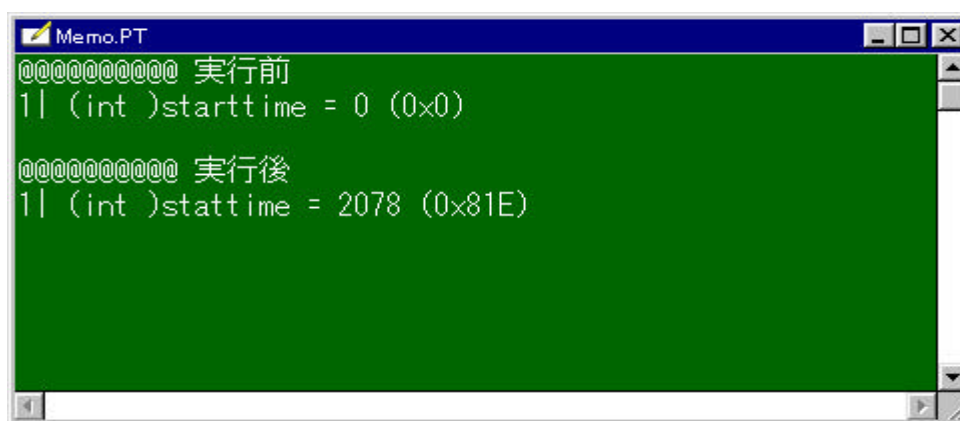


図 30 メモウインドウ

4.16.1 メモウインドウのショートカットキー

メモウイ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イ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	メモウイ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、削除します
GRPH+BS	直前の編集を取り消します

4.16.2 メモウイ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イ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
貼り付け(P)	クリップボードの内容を貼り付けます
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)	メモウインドウの内容を全て削除します

4.17 I/Oウインドウ

I/Oウインドウは、指定された内部周辺I/Oレジスタの表示と設定を行います。

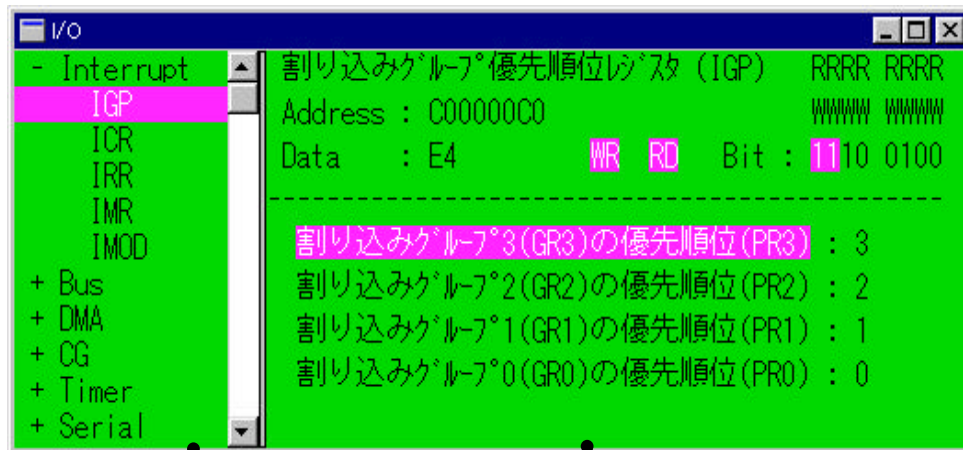


図 31 I/Oウインドウ

アイテムウインドウ

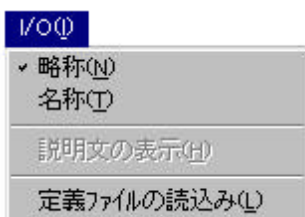
データウインドウ

4.17.1 I/Oウインドウのショートカットキー

I/Oウイ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I/Oウイ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
F1	このウインドウを最大化
F2	ウインドウのフォーカスを次のウインドウに移動
F4	文字列検索ダイアログボックスの表示
F5	ユーザプログラムの実行開始 (G コマンド)
F8	トレース実行 (T コマンド)
Tab	アイテム / データウインドウの切り替え
Enter	アイテムウインドウの場合、階層構造の表示 / 非表示 データウインドウの場合、一つ下階層に移動
ESC・BS	データウインドウの場合、一つ上階層に移動
右方向キー	設定値を変更

4.17.2 I/Oウインドウのローカルメニュー



[I/O]メニューにはI/Oウイ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
略称(N)	レジスタ名を表示します
名称(I)	レジスタの機能名称を表示します
説明文の表示(H)	レジスタの説明を表示します
定義ファイルの読み込み(L)	定義ファイル(RPTV831. I0)からレジスタの情報を読み込みます

4.17.3 I/Oウインドウでのマウス操作

設定値の変更：

変更したいデータの内容をクリックするとポップアップメニューが表示されます。メニュー項目を選択して値の変更、参照をします。

ポップアップメニューは、レジスタによって項目が異なります。

ローカルメニューの表示：

I/Oウインドウのアイテムウインドウ内で右ボタンをクリックします。

クリップボードへのコピー：

I/Oウインドウのアイテムウインドウ内で、任意の位置で左ボタンを押し、ボタンを押したまま選択領域を指定します。ボタンを放すと選択されていた領域がクリップボードにコピーされます。

4.18 ツールバー



図 32 ツールバーの例

ここでは、PARTNER-V831/2-TPのツールバーについて説明します。

ツールバーはPARTNER-V831/2-TP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だけでPARTNER-V831/2-TP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』35頁参照)

4.18.1 ツールバーの設定

















図 33 [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『5.1.3 ツールバーの設定ダイアログボックス』(87頁参照))


4.18.2 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


-  デバッグプログラムをロードします(『5.1.1 ファイルを開くダイアログボックス』86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
-  強制的にユーザプログラムを中断し、PARTNER-V831/2-TP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イ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無い場合、ウォッチ登録ダイアログボックスを表示します(『5.1.7 ウォッチ設定ダイアログボックス』89頁参照)
-  カーソル位置の変数のインスペクトウイ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無い場合、インスペクトダイアログボックスを表示します(『5.1.6 インスペクト設定ダイアログボックス』89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、ステータスウイ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イ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(『5.1.4 色指定ダイアログボックス』88頁参照)
-  現在フォーカスのあるウイ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(『5.1.2 フォントの指定ダイアログボックス』87頁参照)
-  現在フォーカスのあるウイ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(『5.1.5 文字列検索ダイアログボックス』88頁参照)


- 


現在フォーカスのあるウイ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
- 


ロードされたデバッグ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(『5.1.8 モジュールダイアログボックス』90頁参照)
- 


ウインドウ配置1を呼び出します
- 


ウインドウ配置2を呼び出します
- 


ウインドウ配置3を呼び出します
- 


アイコン化されていないウインドウを重ねて表示します
- 


アイコン化されていないウインドウを並べて表示します(PARTNER-V831/2-TP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イ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イ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(『5.1.3 ツールバーの設定ダイアログボックス』87頁参照)
- 

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
- 

コードウインドウのソースコード/逆アセンブル表示の切り替え
- 

PARTNER-V831/2-TPのヘルプの表示
- 

PARTNER-V831/2-TPを終了します

4.19 ウインドウバー



図 34 ウインドウバー

ここでは、PARTNER-V831/2-TPのウインドウバーについて説明します。ウインドウバーはPARTNER-V831/2-TPの各ウインドウをすばやく呼び出すことができます。ウインドウバーは、[表示]-[ウインドウバー]コマンドでウインドウバーの表示/非表示の選択を行えます。(『4.2.4 表示メニュー』35頁参照)

以下に各ボタンの対応を示します。

ボタン	ウインドウ
	コードウインドウ
	コマンドウインドウ
	ウォッチウインドウ
	メモリウインドウ
	レジスタウインドウ
	スタックウインドウ
	ローカルウインドウ
	バックトレースウインドウ
	ブレークウインドウ
	ヒストリウインドウ
	I/Oウインドウ
	メモウインドウ
	コードウインドウ1

【注意】ウインドウバーのボタン設定は変更できません。

4.20 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-V831/2-TP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』35頁参照)

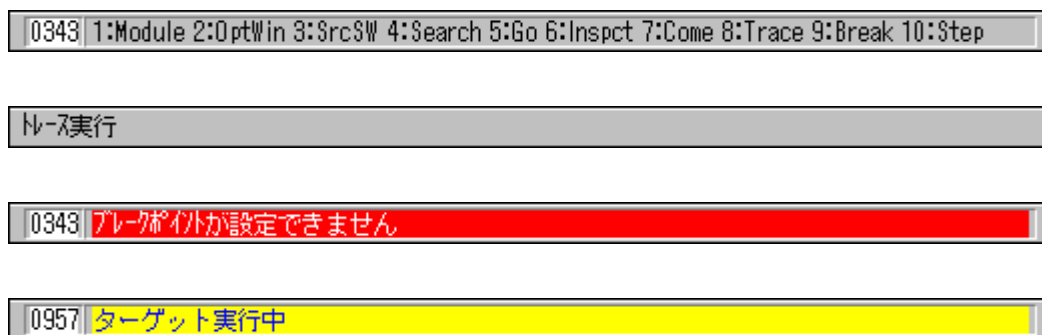


図 35 ステータスバーの例

空白ページ

5 ダイアログコマンド

PARTNER-V831/2-TP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、コマンドウイ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。

5.1 ダイアログボックス

PARTNER-V831/2-TP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。


5.1.1 ファイルを開くダイアログボックス

[ファイル]-[ロード] 

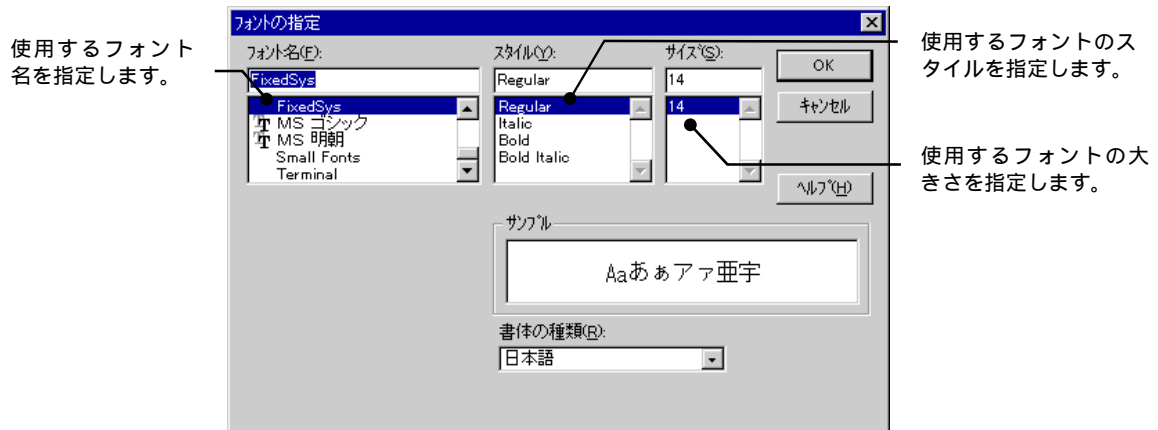
[ファイルを開く]ダイアログボックスは、デバッグ対象のプログラムを選択し、ロードします。




5.1.2 フォントの指定ダイアログボックス

[設定]-[フォント設定] 

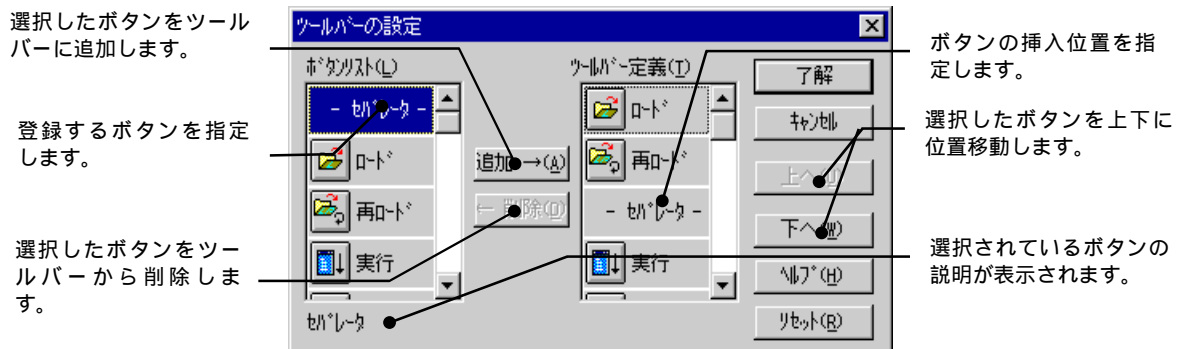
[フォントの指定]ダイアログボックスは、現在選択されているウインドウのフォントを指定します。



5.1.3 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] 

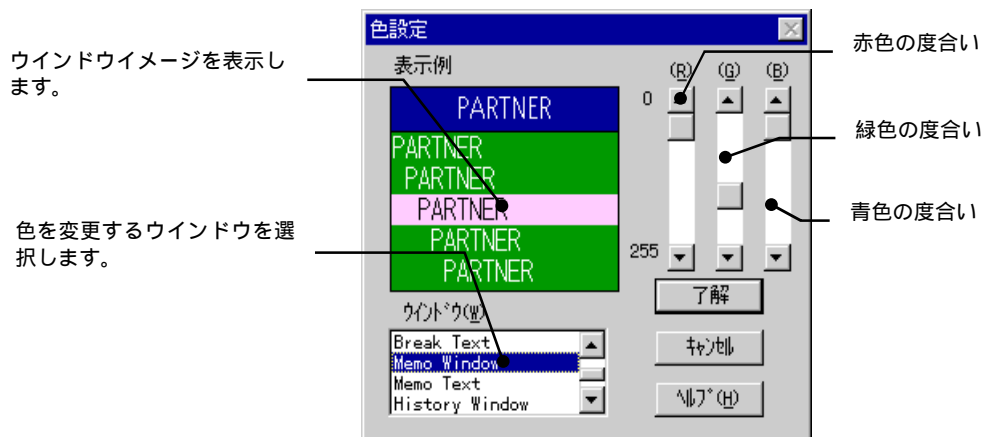
[ツールバーの設定]ダイアログボックスは、ツールバーに登録するボタンを設定します。[ツールバー定義]リストボックスでDELキーを押すと、選択されているボタンを削除します。



5.1.4 色指定ダイアログボックス

[設定]-[色選択]

[色指定]ダイアログボックスは、各ウインドウの背景色、文字色を指定します。



5.1.5 文字列検索ダイアログボックス

[検索]-[文字列の検索]

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



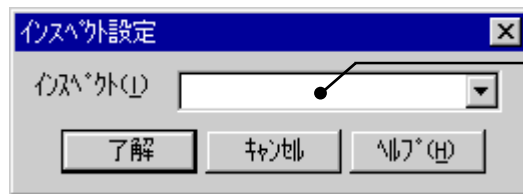
【注意】 検索方向は、コマンドウインドウは初期状態は[先頭へ]になっています。コマンドウインドウ以外の初期状態は[最後へ]です。

5.1.6 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]



インスペクトする変数名を指定するダイアログボックスです。



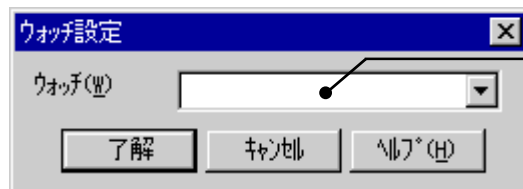
インスペクトする変数名を入力します。

5.1.7 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]



ウォッチ登録する変数名を指定するダイアログボックスです。



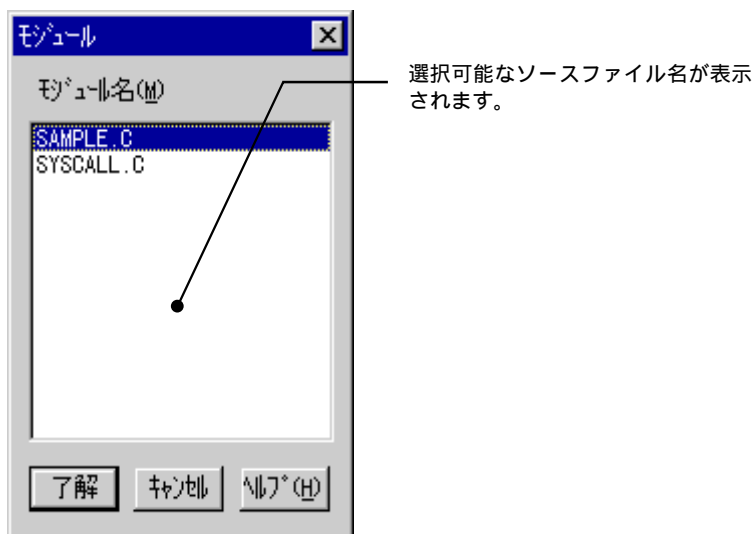
ウォッチ登録する変数名を入力します。

【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ヘルプもしくは『7.2.18 Cのデータ参照/変更』(118頁)を参照してください。

5.1.8 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]

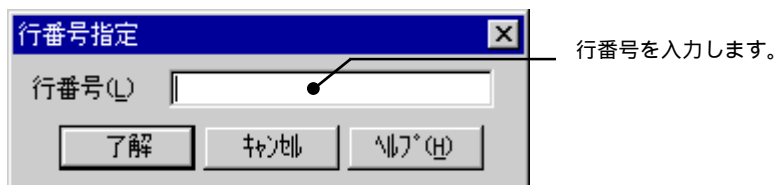
コードウインドウに表示するソースファイルを指定します。[モジュール名]リストボックスには、ロードしたデバッグ対象ファイルのデバッグ情報に定義されているソースファイル名と、Vコマンドで参照したファイル名が表示されます。



5.1.9 行番号指定ダイアログボックス

[コード]-[行番号]

コードウイ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。



5.1.10 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]

コードウイ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。



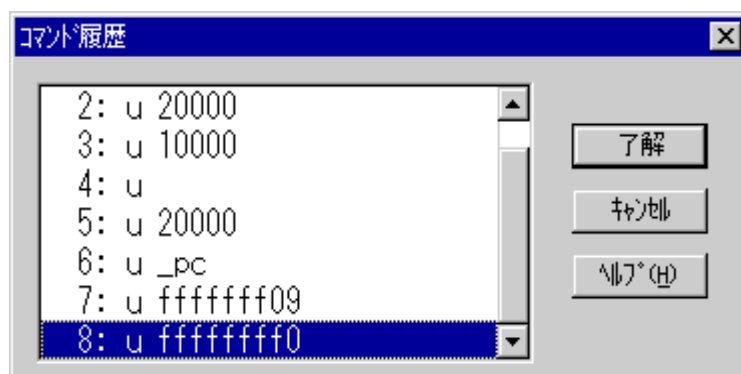
アドレスを入力します。

5.1.11 コマンド履歴ダイアログボックス

[コマンド]-[履歴]

コマンドウイ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。

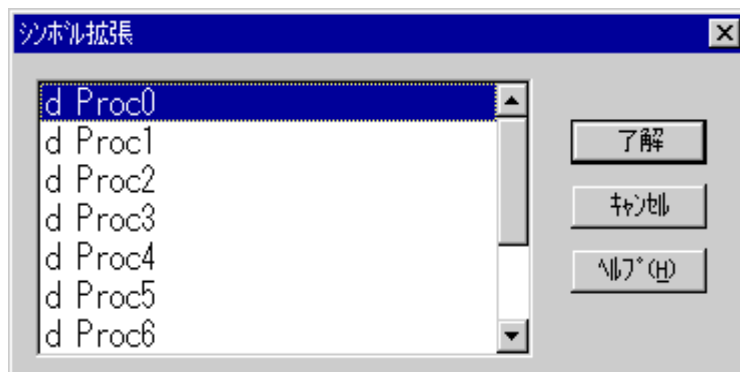
コマンドウインドウのコマンド入力行(現在のプロンプト)に文字列が入力されている場合は、その文字列で始まる文字列をリストボックスに表示します。



5.1.12 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]

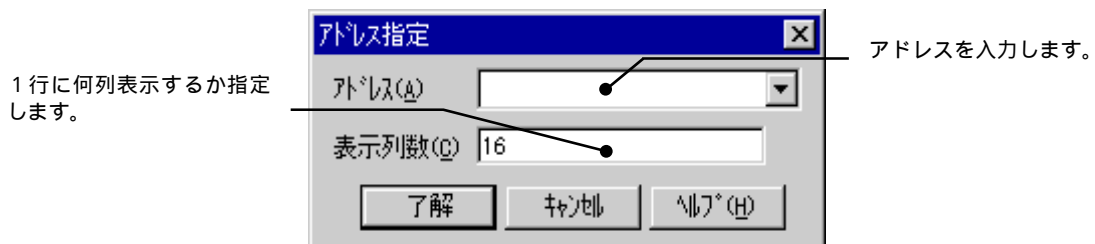
コマンド入力行の最後の文字列から始まるグローバルシンボルの拡張表示をリストボックスにします。(下の例はコマンド入力行に”ins T”と入力した場合です。)



5.1.13 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]

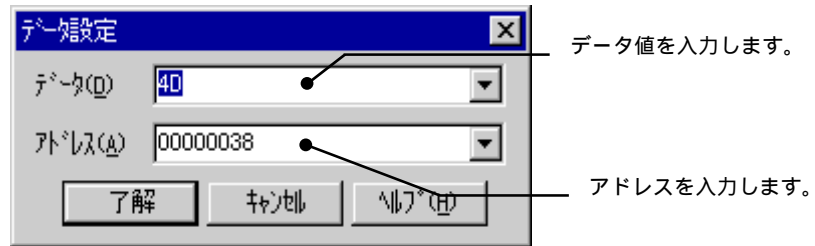
メモリウィンドウに表示するダンプリストの表示開始アドレスを指定します。



5.1.14 データ設定ダイアログボックス

[メモリ]-[データ変更]

メモリの内容を変更するダイアログボックスです。

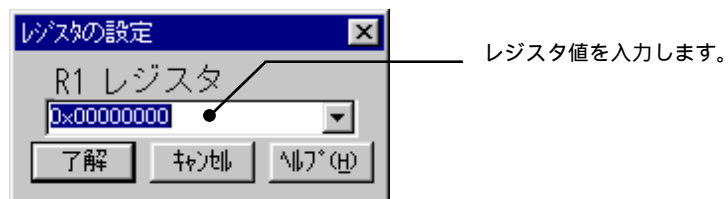


【注意】<了解>ボタンをクリックすると次のアドレスのデータ変更ができます。このダイアログボックスを閉じるには、<キャンセル>ボタンをクリックしてください。

5.1.15 レジスタの設定ダイアログボックス

[レジスタ]-[値の変更]

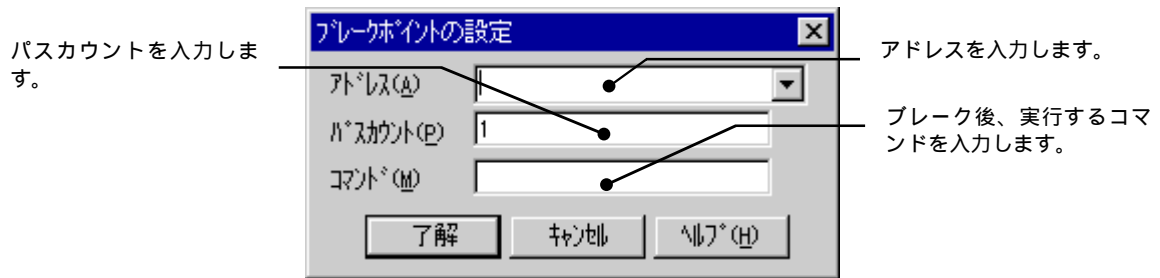
レジスタの値を変更するダイアログボックスです。



5.1.16 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]

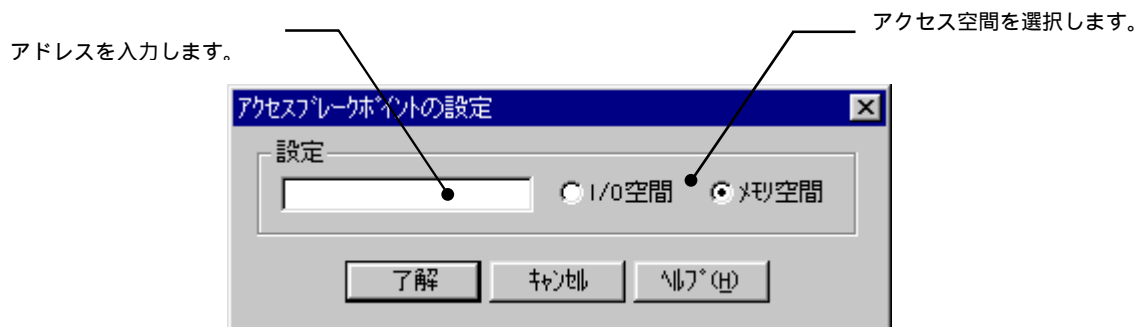
ブレークポイントを設定するダイアログボックスです。



5.1.17 アクセブレークの設定ダイアログボックス

[ブレーク]-[アクセブレーク設定]

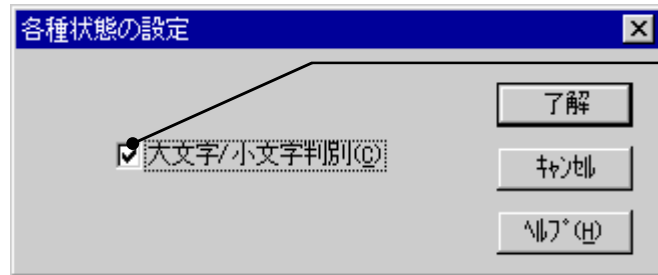
アクセブレークポイントを設定するダイアログボックスです。



5.1.18 各種状態の設定ダイアログボックス

[設定]-[オプション設定]

PARTNER-V831/2-TPの各種状態を設定します。



シンボルの大文字/
小文字の判断を指
定します。

5.1.19 トレースアドレス設定ダイアログボックス

[ヒストリ]-[トリガアドレス]

リアルタイムトレースのトリガポイント、データトレースの条件を設定します。

トリガアドレス値を設定します。



データトレース
の条件1を設定
します。

データトレース
の条件2を設定
します。

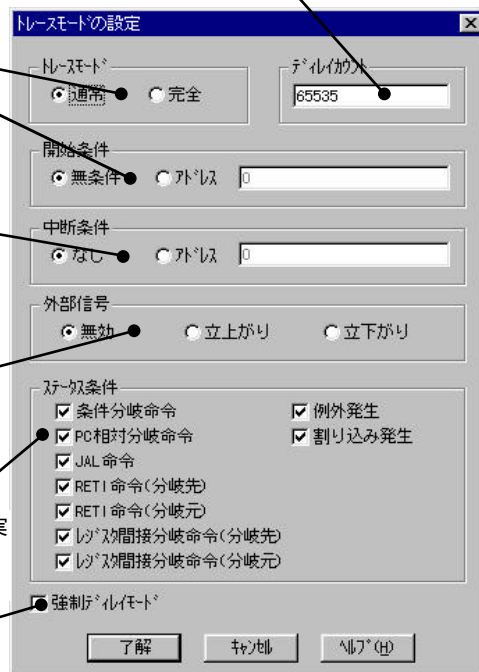
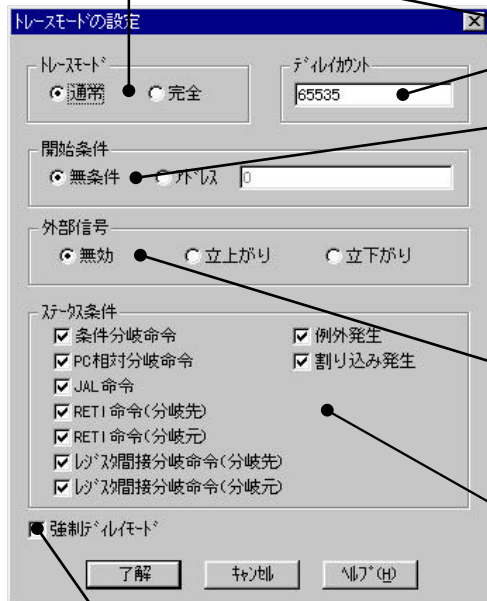
5.1.20 トレースモード設定ダイアログボックス

[ヒストリ][トレースモード]

リアルタイムトレースのモードを設定します。

トレースの取り込みモードを指定します。

トリガ成立後にメモリに取り込むサイクル数を指定します。



トレースの開始条件を選択します。

トレースの中断条件を指定します。

トレーストリガとする外部入力を指定します。

トレースに取り込む命令実行の種別を選択します。

V831の場合

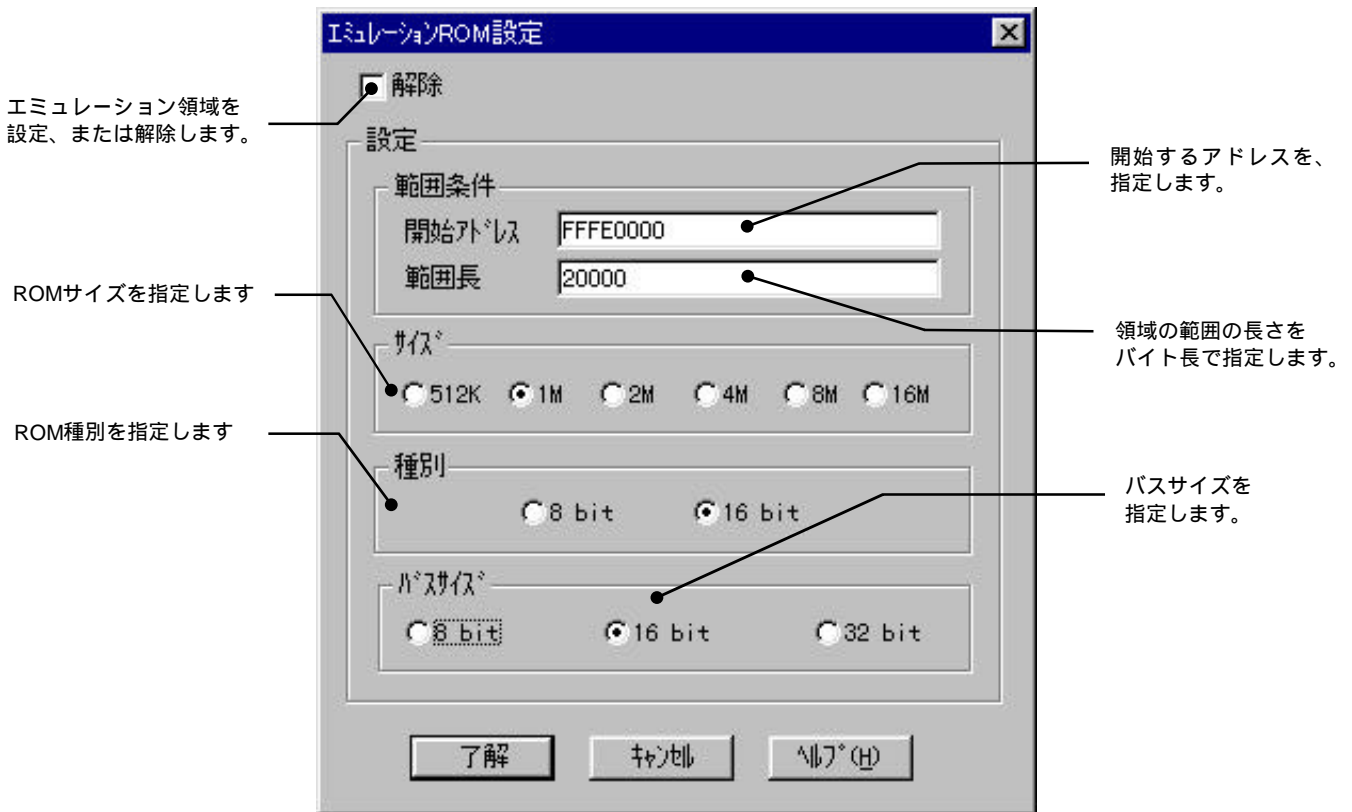
V832の場合

強制ディレイモードを指定するときチェックします。

5.1.21 エミュレーションROM設定ダイアログ

[実行]-[エミュレーションROM設定]

エミュレーション領域を設定します。



ダイアログボックス

5.1.22 CPU環境設定ダイアログボックス

[実行]-[CPU環境設定]

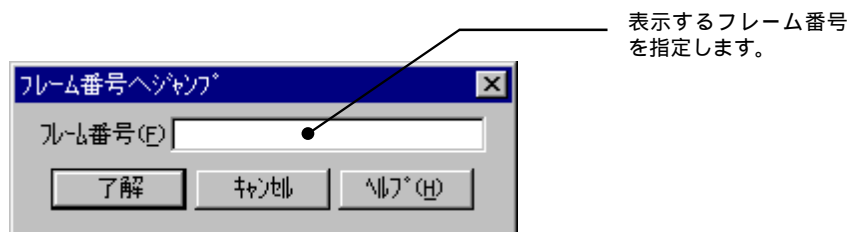
CPUの動作条件を設定します。



5.1.23 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]

表示するフレーム番号を指定します。



6 データ表現

この章では、PARTNER-V831/2-TP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。

6.1 データ表現

PARTNER-V831/2-TPでアドレスやデータ値として扱えるデータ表現はシンボルや演算式等、多岐にわたっています。

6.1.1 PARTNERで扱えるシンボル

PARTNER-V831/2-TP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(プログラムの全領域で有効な)シンボルです。もう一つはローカル(Cの関数内だけで有効な)シンボル(局所変数や静的変数のシンボル)です。

6.1.2 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-V831/2-TPでは、グローバルシンボルの前後のアンダーバーを省略可能です。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。(『OPTIONコマンド』, 『[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のシンボルがあった場合は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 36 [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。このとき、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。

6.1.3 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(宣言)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。

6.1.4 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。

6.1.5 PARTNERで扱える数値

PARTNER-V831/2-TP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。

6.1.6 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。

6.1.7 行番号

PARTNER-V831/2-TP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にのみ有効です。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デバッグ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(コードウインドウに表示されているファイル)の第何行目(行番号)という表示となります。

```
>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となります。

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

6.1.8 文字列(ストリング)

PARTNER-V831/2-TP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を指定するには数値(\$27)で入力してください。

6.1.9 レジスタ名

PARTNER-V831/2-TP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。使用できるレジスタ名は、次の通りです。

レジスタ名

```
_R0,_R1,_R2,_R3,_R4,_R5,_R6,_R7,_R8,_R9,_R10,_R11,_R12,_R13,_
R14,_R15,_R16,_R17,_R18,_R19,_R20,_R21,_R22,_R23,_R24,_R25,_
R26,_R27,_R28,_R29,_R30,_R31,_PC
```

レジスタの別名

上記レジスタ名の他に、以下のレジスタの別名も使用可能です。

```
_HP,_SP,_GP,_TP,_LP
```

```
>while{ _R0!=_R1      /* R0 レジスタと R1 レジスタの内容比較 */
? T                  /* トレース実行コマンド */
?}                  /* マクロ終わり */
>
```

上記の例ではR0とR1のレジスタ値が同じになるまでT(トレース)コマンドを実行します。レジスタと同名のシンボルがある場合にも、レジスタが優先されます。

6.1.10 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-V831/2-TP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
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か等しいとき 1, その他 0)
4	<=	比較演算(右辺が左辺より小さいか等しいとき 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されます。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)の条件判断の処理のために用意されています。

```
>h -(1+2*3)
      oct      dec      hex      asc      float
      3777777771      -7      FFFFFFF9      '....'      -6.805644e++38
>
```

6.2 C言語レベルでのデータ表現

『6.1 データ表現』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として処理する(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有効に使用できます。しかし、デバッグプログラムの中の式は、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。そこで、PARTNER-V831/2-TP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イ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。

6.2.1 Cの式

C言語レベルでの記述と『6.1 データ表現』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。

6.2.2 Cの変数

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。すべてのレジスタ擬似変数はunsigned intの型(属性)をもちます。

レジスタ擬似変数	レジスタ
_r0,,,_r31	R0 レジスタ,,,R31 レジスタ
_pc	PC レジスタ
_hp	R2 レジスタの別名
_sp	R3 レジスタの別名
_gp	R4 レジスタの別名
_tp	R5 レジスタの別名
_lp	R31 レジスタの別名

6.2.3 Cの変数スコープ

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はできません。

PARTNER-V831/2-TP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。

6.2.4 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)

6.2.5 演算子

演算子に関してもCと全く同じ演算子と同じ文法でサポートしています。ただし、PARTNER-V831/2-TP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されます。ただし、代入演算子(優先順位(16))のみ右から左に評価されます。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。

6.2.6 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。勿論、変数値を代入演算子を使ってデータ変更したい場合もあります。しかし、デバッグ中ではほとんどの場合、データ参照だけで変更は比較的少ないのが普通でしょう。

そこでPARTNER-V831/2-TP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はのみVALコマンドを使うようにしてください。

関数呼び出しはさらに危険です。関数を処理中に、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。そのことを意識して使用しないと、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よく注意して使用してください。

```
>? 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)
>
```

空白ページ

7 コマンドリファレンス

PARTNER-V831/2-TPのコマンドウインドウに入力可能なコマンドの説明をします。
詳しい内容(入力例など)はオンラインヘルプを参照してください。

7.1 コマンド解説の規約

PARTNER-V831/2-TPのコマンドは、コマンド名とパラメータリストによって構成されます。また、パラメータは省略できる場合もあります。省略できるパラメータに関しては鉤括弧([...])で示します。2通り以上の選択要素がある場合には中括弧({...})と|で要素内容を記述します。パラメータが省略された場合は、PARTNER-V831/2-TPの初期値や以前実行したコマンドの続きの値などが用いられます。

7.2 機能別コマンド

7.2.1 ハードウェアの初期化

INIT RTE-V831-TPを初期化します。

7.2.2 CPUリセット

RESET ターゲットCPUをリセットします。

7.2.3 環境設定コマンド

ENV [[!]AUTO] [[!]RESET] [[!]NMI] [[!]HLDRQ]
 [[!]INT{00|01|02|03}] [[!]INT{10|11|12|13}] [JTAG{25|12}]
 各種入力信号のマスクなどのCPUの各種環境を設定します。

ROM [アドレス[範囲]] [512K|1M|2M|4M|8M|16M]
 [ROM8|ROM16] [BUS8|BUS816|BUS32]
 エミュレーションROMの種類、アドレス範囲を設定します。

NC [アドレス[範囲]]
 非メモリアドレス領域を設定します。

NCD リスト
 非メモリアドレス領域を解除します。

7.2.4 プログラムロード

L [ファイル名] デバッグプログラムとデバッグ情報を読み込みます。

7.2.5 ファイルリード/ライト

RD ファイル名, 番地 指定ファイルを指定番地に書き込みます。
 WR ファイル名, 範囲 範囲指定されたメモリ内容を指定ファイルに書き込みます。

7.2.6 プログラム実行

T [回数] デバッグプログラムをトリス実行します。(F8)
 P [回数] デバッグプログラムをステップ実行します。(F10)
 G [=実行番地][,ブレーク番地][,/W]
 G@ [,/W] デバッグプログラムを実行します。(F5,F7)
 /W:実行中コマンド禁止
 ESC デバッグプログラムを強制ブレークします。(ESC)



R	レジスタの値を表示します。
_レジスタ=式	指定レジスタを式の値に変更します。
R レジスタ	レジスタ値の変更をします。

7.2.10 システムレジスタ表示/変更

STAT	システムレジスタ表示
STAT システムレジスタ名=データ	指定レジスタ値をデータに変更します。 <EIPC, EIPSW, FEPC, FEPSW, ECR, PSW, TRCW, HCCW>

7.2.11 メモリ表示/変更

D[型]範囲[,回数][,基数]	範囲内のメモリ内容を型書式と基数指定にしたがって表示します。
E[型] 番地	指定番地より型指定された形式でメモリ内容を変更します。
F[型] 範囲,リスト	指定範囲を型指定された形式でリストの値でフィル(Fill)します。
S[型] 範囲,リスト	指定範囲を型指定された形式でリストのメモリパターンを検索します。
C 範囲,番地	指定範囲を指定番地と比較(コパア)します。
M 範囲,番地	指定範囲を指定番地にブロック移動します。

K C の関数のバックトレース表示を行います。

7.2.15 リアルタイムトレース

TRC 現在のリアルタイムトレースモードを表示します。

TRC [デレイ値] [[!]REAL] [FORCE|PCADDR アドレス]
 [!ENDADDR|ENDADDR アドレス] [[!]DELAY
 [[!]ALL] [[!]EX] [[!]INT] [[!]CJMP] [[!]JMPR] [[!]JAL
 [[!]RET11] [[!]RET12] [[!]JREG1] [[!]JREG2
 [NOEXT|NEGA|POS]

リアルタイムトレースのモードを設定します。

TS リアルタイムトレースの停止

TG リアルタイムトレースの再開

TRG 現在のリアルタイムトレースのアドレス条件を表示します。

TRG [トリガアドレス][, [デレイ条件 1]][, [デレイ条件 2]]

リアルタイムトレースのアドレス条件を設定します。

TD[U] リアルタイムトレースの内容を表示するモードに入ります。'U'がついた場合は、逆アングル表示モードとなります。

リアルタイムトレース表示モードの内部コマンド

B 表示開始フレームをトレースメモリの先頭に移動します。

D [フレームアドレス S][, フレームアドレス E]
 表示モードをダンプ表示に変更し、表示します。

U [フレームアドレス S][, フレームアドレス E]
 表示モードを逆アングル表示に変更し、表示します。

機能別コマンド



INS C の式[,関数]	C の式を評価してウィンドウに表示します。(F6,Ctrl+I)
W? C の式	C の式をウィンドウに登録します。(Shift+F7,Ctrl+W)
W[型] 番地[,範囲][,基数]	番地と範囲で指定されたメモリ内容をウィンドウに登録します。
Y リスト	リストで指定されたウィッチ行を削除します。
VAL C の式[,関数]	
? C の式[,関数]	C の式を評価して表示します。

7.2.19 文字列の定義(Cの式評価)

DEF 文字列 1 文字列 2	
#DEFINE 文字列 1 文字列 2	ミプリの為の文字列を定義します。
DEF	現在登録されている文字列定義内容の表示をします。
DEF *	現在登録されているすべての文字列定義を無効にします。

EXIT	PARTNER を終了します。
Q	PARTNER を終了します。
HELP	ヘルプ表示を行います。
VER	PARTNER のバージョン表示をします。
!!	コマンド履歴の表示をします。
! 文字列	コマンド履歴の文字列サーチをします。
&	スクリプトへの移行/終了
& スクリプト	スクリプトの実行

7.2.23 基数変更

N 基数	入力基数を 10 進数または 16 進数に設定します。
------	-----------------------------

7.2.24 ロギング/バッチ

> ファイル名	コマンドウィンドウの表示/入力をファイルに出力します。
>> ファイル名	指定ファイルへのアペンド (APPEND) ログ出力します。
>	ログの中断 (ログファイルのクローズ) をします。
< ファイル名	コマンドウィンドウの入力をファイルから行います。ESC キーでバッチ処理の中断が可能です。

機能別コマンド





C言語で作成したモジュールのコンパイル時には、`-g -XDWARF`オプションを付加してください。このオプションを付加してコンパイルしたモジュールはソースレベルデバッグが可能です。さらに、`-o`オプションで実行ファイルのファイル名を指定する場合は、拡張子を `.out` にしてください。

```
>ccv830 -g -XDWARF ソースファイル名 -o 実行ファイル名.out
```

ソースファイル名	ソースファイル名
実行ファイル名	実行ファイル名

この結果ELFファイルフォーマット(.OUT)の実行ファイルが作成されます。

8.1.2 使用例

1) コンパイル

```
>ccv830 -g -XDWARF -c @cmpfiles
```

<i>cmpfiles</i>	コンパイル対象ソースファイルのリストファイル (複数のソースファイル名称を指定)
-----------------	---

この結果、拡張子".o"のオブジェクトファイルが作成されます。

2) リンク

```
>lx -map=chkpro.map -o chkpro.out @chkpro.lx @lnkfiles
```

<i>chkpro.lx</i>	リンクに引渡すオプション定義(リンクマップ)ファイル
<i>lnkfiles</i>	リンク対象オブジェクトファイルのリストファイル (複数のオブジェクトファイル名称を指定)

この結果、ELFファイルフォーマット(.OUT)の実行ファイルが作成されます。

3) リンクマップの例

実行プログラムで指定される各セクションをメモリ空間のどの位置に配置するのかを決定します。

```
-m 3 -sec
{
    .text      0xfe001000    :
    .rodata                    :
    .syscall                    :
    .secinfo                    :
    .fixaddr                    :
    .fixtype                    :
    .data align(4)              :
    .sdata align(4)            :
    .sdata                    :
    .rodata                    :
    .sbss                      :
    .bss                      :
    .heap align(8) pad(0x4000) :
    .stack align(8) pad(0x2000) :
}
-Mx
```

8.2 GNU C(exeGCC)

PARTNER-V831/2-TPはGNU Cの実行ファイルを読み込みます。実行ファイルの拡張子は".OUT"です。

8.2.1 コンパイルとリンク方法

C言語で作成したモジュールのコンパイル時には、`-g`オプションを付加してください。このオプションを付加してコンパイルしたモジュールはソースレベルデバッグが可能です。さらに、`-o`オプションで実行ファイルのファイル名を指定する場合は、拡張子を `.out` にしてください。以下にコンパイル及びリンク時の基本的な指定方法を明記します。

```
>gcc -g ソースファイル名 -o 実行ファイル名.out -T コマンドファイル名
```

ソースファイル名	ソースファイル名
実行ファイル名	実行ファイル名
コマンドファイル名	リンク用コマンドファイル

この結果COFFフォーマット(.OUT)の実行ファイルが作成されます。

8.2.2 使用例

1) コンパイル

```
>gcc -g -m830 -c @mpfiles
```

`mpfiles` コンパイル対象ソースファイルのリストファイル
(複数のソースファイル名称を指定)

この結果、拡張子".o"のオブジェクトファイルが作成されます。

2) リンク

```
>ld -Map chkpro.map -Tchkpro.cmd -o chkpro.out @lnkfiles
```

`chkpro.cmd` リンカに引渡すオプション定義(リンクマップ)ファイル
`lnkfiles` リンク対象オブジェクトファイルのリストファイル
(複数のオブジェクトファイル名称を指定)

この結果、COFFファイルフォーマット(.OUT)の実行ファイルが作成されます。

3) リンクマップの例

実行プログラムで指定される各セクションをメモリ空間のどの位置に配置するのかを決定します。

```
SECTIONS
{
  _xfer = 0 ;
  _stack_init = 0xFE010000 ;
  _heap_size = 0x4000 ;
  .text 0xFE001000 :
  {
    *(.text)
  }
  .rodata . :
  {
    *(.rodata)
    _erdata = . ;
  }
  .data . :
  {
    _data = . ;
    *(.data)
    _edata = . ;
  }
  .bss . :
  {
    _fbss = . ;
    *(.bss)
    *(COMMON)
    _end = . ;
    _heap = . ;
  }
  /* DWARF debug sections */
  .debug          0 : { *(.debug) }
  .debug_srcinfo  0 : { *(.debug_srcinfo) }
  .debug_aranges  0 : { *(.debug_aranges) }
  .debug_pubnames 0 : { *(.debug_pubnames) }
  .debug_sfnames  0 : { *(.debug_sfnames) }
  .line           0 : { *(.line) }
}
```

8.3 NEC(CA830) C

PARTNER-V831/2-TPはELFフォーマットの実行ファイルを読み込みます。実行ファイルの拡張子は".OUT"です。

8.3.1 コンパイルとリンク方法

C言語で作成したモジュールのコンパイル時には、-gオプションを付加してください。このオプションを付加してコンパイルしたモジュールはソースレベルデバッグが可能です。さらに、-oオプションで実行ファイルのファイル名を指定する場合は、拡張子を.out にしてください。

```
>ca830 -c -g ソースファイル名.c
>ld830 -D リンクディレクティブ -o 実行ファイル名.out ソースファイル名.o -Llib -lc
```

ソースファイル名	ソースファイル名
実行ファイル名	実行ファイル名
リンクディレクティブ	リンクディレクティブファイル

リンクディレクティブファイルの内容

```
DATA : !LOAD ?RW V データアドレス {
    .data      = $PROGBITS ?AW;
    .sdata     = $PROGBITS ?AWG;
    .sbss      = $NOBITS ?AWG;
    .bss       = $NOBITS ?AW;
};
TEXT : !LOAD ?RX V コードアドレス {
    .text      = $PROGBITS ?AX;
};
__tp_TEXT @ %TP_SYMBOL;
__gp_DATA @ %GP_SYMBOL & __tp_TEXT;
```

コードアドレス	コード先頭アドレス指定
データアドレス	データ先頭アドレス指定

この結果ELFフォーマット(.OUT)の実行ファイルが作成されます。

8.3.2 使用例

```
C>ca830 -g -c dry.c  
C>ld830 -D dfile -o dry.out lib¥crtn830.o dry.o -Llib -lc
```

dfile の内容

```
DATA : !LOAD ?RW V0x2000 {  
    .data          = $PROGBITS ?AW;  
    .sdata         = $PROGBITS ?AWG;  
    .sbss          = $NOBITS ?AWG;  
    .bss           = $NOBITS ?AW;  
};
```

```
TEXT : !LOAD ?RX V0xffe00000 {  
    .text          = $PROGBITS ?AX;  
};
```

```
__tp_TEXT @ %TP_SYMBOL;  
__gp_DATA @ %GP_SYMBOL &__tp_TEXT;
```

空白ページ

付録1

コマンドラインによる起動オプションの設定

PARTNER-V831/2-TPの起動オプションは通常、RPTSETUP(V831&2TP)の[起動オプション]ダイアログボックスで指定します。指定された起動オプションはプロジェクトファイル(RPTV831T.KPJ)に書き込まれます。RPTSETUP(V831&2TP)を使用せずに、PARTNER-V831/2-TP(RPTV831T.EXE)のショートカットを作成し、[プロパティ]で直接編集することも可能です。

PARTNER-V831/2-TPの起動オプションを以下に示します。

-Bオプション

デバッグ情報およびマクロ領域のサイズ指定を行います。デバッグ情報量にはグローバル、ローカルシンボル、行番号情報、関数/変数の属性情報などの多くの情報が格納されます。

-B[サイズD],[サイズM]

サイズD デバッグ情報を登録する領域のサイズ(16K バイト単位)
省略時には 512K バイトの領域を確保します。

サイズM マクロ定義情報を保存する領域のサイズ(1K バイト単位)
省略時には 2K バイトの領域を確保します。

-Dオプション

PARTNER-V831/2-TPのカレントディレクトリを指定します。カレントディレクトリにはコンフィグファイル(RPTV831T.CFG)、Windows用初期設定ファイル(RPTV831T.KPJ)が存在する必要があります。

-D ディレクトリ

ディレクトリがPARTNER-V831/2-TP起動時のカレントディレクトリになります。このオプションが省略された場合は[プロパティ]ダイアログボックスの[作業フォルダ]に指定されているディレクトリ、[コマンドライン]に指定されている実行プログラムの存在するディレクトリの順にディレクトリが指定されます。

-Eオプション

コードウインドウで表示/参照するソースファイルの拡張子のデフォルトを追加指定します。

-E 拡張子

拡張子がソースファイルのデフォルトの拡張子となります。**拡張子**にはピリオドを除く拡張子部分を指定します。このオプションが省略された場合はfile.Cまたはfile.ASMがデフォルトの指定になります。

-SDオプション

コードウインドウに参照するソースファイルのあるディレクトリを指定します。ディレクトリ指定はセミコロン(;)で区切ることにより複数指定が可能となります。

-SD ディレクトリ

ディレクトリがソースファイルのあるディレクトリになります。このオプションが省略された場合は-Dオプションで指定したディレクトリになります。

-TABオプション

コードウインドウのソース表示をするときのタブのサイズを指定します。エディタでタブサイズを変更したファイルを表示するのに便利な機能です。

-TAB タブサイズ

タブサイズで指定される数にタブを合わせます。このオプションを省略した場合のタブサイズは8に設定されます。

-Xオプション

アセンブラまたは各種Cのデバッグモードの指定をします。PARTNER-V831/2-TPでは多くの処理系をサポートしているために各処理系を明確にする必要があります。この指定を正しく行っていない場合には、正常なデバッグができないかあるいは特定の機能が使用できない場合があります。各処理系の使用方法は『8 高級言語レベルデバッグとROM化(121頁)』、『各処理系のマニュアル』を参照してください。

-X

アセンブラのデバッグモードの指定です。(デフォルト)このモードでPARTNER-V831/2-TPを起動するとC関連(スタックのバックトレースやローカル変数などに関連した)コマンドは利用できません。

-XC

Green Hills Cのデバッグモード指定。デバッグ情報は.OUTファイルから読み込みます。

-XG

GNU C(exeGCC V800)のデバッグモード指定。デバッグ情報は.OUTファイルから読み込みます。

-XN

NEC Cのデバッグモード指定。デバッグ情報は.OUTファイルから読み込みます。

@オプション

-Dオプション以外のオプションを@で指定したファイルから読み込むことが可能です。

【例】

コマンドライン

```
C:¥RPTV831T¥BIN¥RPTV831T -DC:¥SAMPLE @TEST.INF
```

ファイル内容(TEST.INF)

```
-B1024 -SDC:¥SAMPLE¥SOURCE -XC
```

空白ページ

付録2

エラーメッセージ

PARTNER-V831/2-TPのエラーメッセージについて以下に説明します。

コマンドエラー

指定されたコマンドがPARTNERの内部コマンドおよびマクロコマンドとして認識できません。

アドレス指定が誤りです

アドレス入力部に不適当なアドレスが指定された場合や、スタートアドレスとエンドアドレスが逆転している場合に発生します。また、シンボル登録されていないシンボル名が使用された場合にも発生します。

データ指定が誤りです

データ入力部に不適当なデータが指定された場合や、範囲外のデータが指定された場合に発生します。

コマンドのフォーマットが不正です

コマンドの入力書式、パラメータの指定方法や数に間違いがあります。

Verifyエラー

メモリに正しくデータが書き込めませんでした。メモリの実装されていないアドレスやROM領域に書き込み動作をしたときに発生します。

マクロ内でのマクロの定義はできません

マクロの定義(登録)は、PARTNERのコマンドレベルで行ってください。マクロコマンド内での定義はできません。

マクロ内でのマクロ削除はできません

マクロの削除は、PARTNERのコマンドレベルで行ってください。マクロコマンド内での削除はできません。

マクロ名が内部コマンドと重複しています

マクロコマンド定義を行おうとしたマクロコマンド名がPARTNERの内部コマンドと重複しています。違うマクロコマンド名で定義してください。

マクロバッファがいっぱいです

マクロコマンド定義のためのバッファがいっぱい、または定義したマクロコマンドの数が多すぎます。RPTSETUP(V831&2TP)を使用してマクロバッファサイズを広げてPARTNERを起動してください。

マクロ定義が正しく終了していません{ }

マクロコマンドの定義で中括弧({ })の数が一致していません。

マクロが16レベル以上入れ子になっています

マクロの入れ子(ネスティング)は最大15レベルまで許されます。

指定された設定がありません

ブレークポイント、ウォッチ等の指定番号の設定が存在しません。

ブレークポイントが設定できません(最大15ポイント)

ブレークポイントを15点を越えて設定しようとした。ブレークポイントは最大15点まで指定できます。

ウォッチの設定ができません

ウォッチを16点を越えて設定しようとした。ウォッチ設定は最大16点まで指定できます。

ウォッチの指定が重複しています

Wコマンドでウォッチ登録しようとした内容はすでにウォッチ登録されています。

ファイルフォーマットが不正です

ロードしようとした実行形式のファイルのフォーマットが異常です。RPTSETUP(V831&2TP)で指定したデバッグモードとロードしようとするファイルが一致していません。

指定ファイルがありません

コマンド行で指定されたファイルが見つかりません。

指定ファイルがオープンできません

指定されたファイルがオープンできません。そのファイルが存在するか確認してください。また、ファイルをオープンしすぎている場合にも発生します。

ファイルがクリエートできません

ディスクがいっぱいか、ファイルを同時にオープンしすぎています。不要なファイルをクローズするなどしてください。

ディスクがいっぱいです

指定されたディスクの空き容量が足りません。

デバッグ情報がありません

Lコマンドでプログラムをロードするとき、デバッグ情報がプログラムファイル内にありませんでした。

デバッグ情報のフォーマットが不正です

デバッグ情報のフォーマットが不正です。RPTSETUP(V831&2TP)で指定したデバッグモードとロードしようとするファイルを確認してください。

デバッグ情報領域がいっぱいです(起動時の-Bオプション参照)

デバッグ情報登録用のバッファに空き領域がありません。RPTSETUP(V831&2TP)で指定したデバッグ情報バッファサイズを拡大してください。

ローカルシンボルの設定はできません

既に登録されているローカルシンボルと同名のグローバルシンボルを登録しようとしています。

Cの変数が見つかりません

?コマンドやVALコマンドで指定したCの変数が見つかりません。

Cの式計算エラー

?コマンドやVALコマンドでのCの式でエラーがあります。

副作用のある演算子は使用できません

?コマンドなどで副作用のある演算子(=,+=,-=等)は使用できません。副作用のある演算子はVALコマンドで使用してください。

PARTNERの内部エラー

PARTNERの内部処理で不都合が生じた場合に表示されます。このエラーは通常起こらないように設計されています。もし、このエラーが発生し、かつ再現性がある場合には、販売会社にご連絡ください。

ターゲット実行中は使用できません

指定したコマンドは、ユーザプログラム実行中には使用できません。ブレイクしてからコマンドを実行してください。

モニタプログラムが正しく実行できません

これは、CPUへのレディ信号がアクティブにならない場合に発生します。通常、そのような状態になるメモリ、I/Oアドレスへのアクセスで発生します。パートナーは、エラーが発生した時点で各ウィンドウの処理を中断します。ウィンドウがアクティブになった場合は、再度発生しますので、各ウィンドウ内のアドレスを変更してください。レジスタウィンドウでは、メモリ表示を解除することでも対応できます。ヒストリウィンドウでは、トレース出力中にそのようなアドレスがあるため、同じトレース条件でリアルタイムトレースを行うと同様にエラーが発生しますので、リアルタイムトレースの設定を変えてください。

また、環境ファイル(RPTV831T.CFG)でのMAP指定によって対応することができます。これは、コマンドからアクセス可能なメモリ領域を指定するものです。コマンドからのアクセスを制限します。ただし、パートナーのコマンドでのアクセスを制限するだけで、ヒストリ等のRTE-V831-TPによるアクセスは制限できません。

索引

.

.DAT
終了時に作成するファイル..... 24

C

CPUリセット 115
Cの式..... 107
Cのデータ参照/変更 118
Cの変数 108
Cの変数スコープ 108

G

GNU C..... 124
コンパイルとリンク方法 124
Green Hills C..... 122
コンパイルとリンク方法 122

I

I/Oウインドウ 77
ショートカットキー 77
マウス操作 78
ローカルメニュー 78
I/Oポート入力/出力 117
INIT.MCR
起動時に必要なファイル22, 23

M

MEMO.PT
終了時に作成するファイル..... 24

N

NEC(CA830) C..... 126
コンパイルとリンク方法 126

P

PARTNERウインドウ
画面構成..... 30
PARTNERで扱えるシンボル.....100

あ

アクセスブレイクポイント116
アSEMBル.....117
アドレス102
以前の環境を開く 20
インスペクトウインドウ 73
ショートカットキー 73
マウス操作 74
ローカルメニュー40, 74
画面構成..... 31
ウインドウバー 82
ウインドウメニュー 42
ウォッチウインドウ 67
ショートカットキー 67
マウス操作 68
ローカルメニュー39, 68
画面構成..... 31
エラーメッセージ 26
演算子.....110
演算式.....105
オプション設定.....120

か

画面構成 30
構成要素..... 30
表示..... 32
画面コントロール120
環境設定 16

以前に設定した環境の変更.....	20
新規に環境を作成.....	17
環境設定プログラムの使用方法.....	16
環境ファイル(RPTV831T.CFG).....	21
関連ファイル.....	23
基数変更.....	119
起動.....	25
起動オプション	
-Bオプション.....	129
-Dオプション.....	129
-Eオプション.....	130
-SDオプション.....	130
-TABオプション.....	130
-Xオプション.....	131
コマンドライン.....	129, 133
起動オプションの設定	
新規に環境を作成.....	18
起動時に必要なファイル.....	23
行番号.....	103
グローバルシンボル.....	100
検索メニュー.....	35
高級レベル言語デバック	
GNU C.....	124
Green Hills C.....	122
NEC(CA830) C.....	126
コードウインドウ.....	49
ショートカットキー.....	50
マウス操作.....	53
ローカルメニュー.....	37, 52
画面構成.....	30
コード表示.....	119
コマンドウインドウ.....	54
ショートカットキー.....	54
ローカルメニュー.....	37, 56
画面構成.....	31
コマンドリファレンス.....	113

コンフィグファイル	
コンフィグファイルの設定.....	17
起動時に必要なファイル.....	23

さ

作業ディレクトリの指定	
新規に環境を作成.....	17, 23, 24
式の表示.....	118
システムコール.....	119
システム制御.....	119
システムレジスタ表示/変更.....	116
実行メニュー.....	36
終了時に作成するファイル.....	24
ショートカットキー.....	44
I/Oウインドウ.....	77
インスペクトウインドウ.....	73
ウォッチウインドウ.....	67
コードウインドウ.....	50
コマンドウインドウ.....	54
スタックウインドウ.....	62
バックトレースウインドウ.....	65
ヒストリウインドウ.....	71
ブレークウインドウ.....	69
メモウインドウ.....	75
メモリウインドウ.....	57
レジスタウインドウ.....	60
ローカルウインドウ.....	63
初期設定コマンド.....	22
シンボル表示/設定.....	118
スタックウインドウ.....	62
ショートカットキー.....	62
ローカルメニュー.....	38, 62
画面構成.....	31
ステータスバー.....	83
設定メニュー.....	43
セットアップ	

セットアップの前に	12	データ表現	100
セットアップの方法	13	特殊シンボル	102
必要なソフトウェア	3		
必要なハードウェア	2		
		は	
た		ハードウェアの初期化	115
ダイアログコマンド	85	バックトレース.....	117
ダイアログボックス.....	86	バックトレースウインドウ	65
CPU環境設定	98	ショートカットキー	65
アクセスブレイクの設定	94	マウス操作	66
アドレス指定(コード).....	91	ローカルメニュー	39, 66
アドレス指定(メモリ).....	92	画面構成.....	31
色指定	88	バッチ.....	119
インスペクト設定.....	89	ヒストリウインドウ	71
ウォッチ設定.....	89	ショートカットキー	71
エミュレーションROM設定.....	97	ローカルメニュー	41, 72
各種状態の設定	95	画面構成.....	32
行番号指定	90	ヒストリメニュー	
コマンド履歴.....	91	ヒストリウインドウ	41
シンボル拡張.....	92	表示メニュー	35
ツールバーの設定	87	ファイルメニュー	33
データ設定	93	ファイルリード/ライト	115
トレースアドレス設定.....	95	副作用のある式.....	111
トレースモード設定	96	ブレイクウインドウ	69
ファイルを開く	86, 96	ブレイクウインドウ	
フォントの指定	87	ショートカットキー	69
ブレイクポイントの設定	94	マウス操作	70
フレーム番号指定.....	98	ローカルメニュー	40, 70
モジュール	90	画面構成.....	31
文字列検索	88	ブレイクポイント	116
レジスタの設定	93	プログラムマネージャへの登録	
ツールバー	79	新規に環境を作成.....	19
設定.....	79	プログラムロード	115
定数	109	プログラム実行.....	115
データ表現	99	プロジェクトファイル	
C言語レベルでのデータ表現	107	起動時に必要なファイル	23
		ヘルプメニュー	43

編集メニュー	34
ボタンの機能	80

ま

マウス操作	47
I/Oウインドウ	78
インスペクトウインドウ	74
ウォッチウインドウ	68
コードウインドウ	53
バックトレースウインドウ	66
左ボタンドラッグ	48
ブレークウインドウ	70
右ボタンクリック	47
メモリウインドウ	58
レジスタウインドウ	61
ローカルウインドウ	64
マクロコマンド	120
メニュー	33
ウインドウメニュー	42
検索メニュー	35
実行メニュー	36
設定メニュー	43
表示メニュー	35
ファイルメニュー	33
ヘルプメニュー	43
編集メニュー	34
ローカルメニュー	37
メモウインドウ	75
ショートカットキー	75
ローカルメニュー	40, 76
画面構成	31, 32
メモリウインドウ	57
ショートカットキー	57
マウス操作	58
ローカルメニュー	38, 58

画面構成	31
メモリ表示/変更	116
文字列	104
文字列の定義	118

ら

リアルタイムトレース	117
レジスタウインドウ	59
ショートカットキー	60
マウス操作	61
ローカルメニュー	38, 61
画面構成	31
レジスタ名	104
レジスタ表示/変更	116
ローカルウインドウ	63
ショートカットキー	63
マウス操作	64
ローカルメニュー	39, 64
画面構成	31
ローカルシンボル	101
ローカルメニュー	37
I/Oウインドウ	78
インスペクトウインドウ	40, 74
ウォッチウインドウ	39, 68
コードウインドウ	37, 52
コマンドウインドウ	37, 56
スタックウインドウ	38, 62
バックトレースウインドウ	39, 66
ヒストリウインドウ	72
ブレークウインドウ	40, 70
メモウインドウ	40, 76
メモリウインドウ	38, 58
レジスタウインドウ	38, 61
ローカルウインドウ	39, 64
ロギング	119

PARTNER-V831/2-TPマニュアル

初版 発行日 1998年6月

株式会社 マイダス・ラボ

Copyright 1998 Midas lab Inc./Kyoto Micro Computer Co.,LTD.