

PARTNER-V30MZ-TPマニュアル

この度は、『RTE-V30MZ-TPシリーズ』（以下、『RTE-V30MZ-TP』と略します。）用ソースレベルデバッガ『PARTNER-V30MZ-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-V30MZ-TPマニュアル』はソースレベルデバッガ『PARTNER-V30MZ-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、改変等できません。

お客様に設定される使用権は、お客様が登録するPARTNER-V30MZ-TP用のモニタを搭載した1台のシステムにおいてのみ、PARTNER-V30MZ-TPを使用する権利を意味します。

PARTNER-V30MZ-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。

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。

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、一般に各メーカーの商標です。

空白ページ

改定履歴

日付 Y/M/D	Rev	内容
98.6.28	1.0	初版
98.11.10	1.1	PT-V30MZ-TP Ver.1.1対応 * ENV,NC,NCD,ROMコマンドの追加 *フラグレジスタ名(FLAGS)の追加 *CPU環境設定,エミュレーションROM設定ダイアログの追加

空白ページ

目次

はじめに.....	1
1 はじめてのPARTNER-V30MZ-TP.....	5
1.1 PARTNER-V30MZ-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-V30MZ-TPの基本的な操作手順.....	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-V30MZ-TPの設定.....	12
2.1.2 RTE-V30MZ-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-V30MZ-TPのインストール.....	13
2.2.3 PARTNER-V30MZ-TPのアンインストール.....	13
3 起動.....	15
3.1 PARTNER-V30MZ-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(RPTV30T.CFG).....	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-V30MZ-TP起動時に必要なファイル.....	23

3.4.2	PARTNER-V30MZ-TP終了時に作成するファイル	24
3.5	PARTNER-V30MZ-TPの起動.....	25
3.5.1	PARTNER-V30MZ-TPが正常に起動した場合	25
3.5.2	PARTNER-V30MZ-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	59
4.8.2	レジスタウインドウのローカルメニュー	60
4.8.3	レジスタウインドウでのマウス操作	60
4.9	スタックウインドウ.....	61
4.9.1	スタックウインドウのショートカットキー	61
4.9.2	スタックウインドウのローカルメニュー	61
4.10	ローカルウインドウ.....	62
4.10.1	ローカルウインドウのショートカットキー	62
4.10.2	ローカルウインドウのローカルメニュー	63
4.10.3	ローカルウインドウでのマウス操作	63
4.11	バックトレースウインドウ.....	64
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	64
4.11.2	バックトレースウインドウのローカルメニュー	65
4.11.3	バックトレースウインドウでのマウス操作.....	65
4.12	ウォッチウインドウ.....	66
4.12.1	ウォッチウインドウのショートカットキー	66
4.12.2	ウォッチウインドウのローカルメニュー	67
4.12.3	ウォッチウインドウでのマウス操作	67
4.13	ブレークウインドウ.....	68
4.13.1	ブレークウインドウのショートカットキー	68
4.13.2	ブレークウインドウのローカルメニュー	69
4.13.3	ブレークウインドウでのマウス操作	69
4.14	履歴ウインドウ.....	70
4.14.1	履歴ウインドウのショートカットキー	71
4.14.2	履歴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	ツールバー.....	77
4.17.1	ツールバーの設定.....	77
4.17.2	各ボタンの機能.....	78
4.18	ウインドウバー.....	80
4.19	ステータスバー.....	81
5	ダイアログコマンド.....	83
5.1	ダイアログボックス.....	84
5.1.1	ファイルを開くダイアログボックス.....	84
5.1.2	フォントの指定ダイアログボックス.....	85
5.1.3	ツールバーの設定ダイアログボックス.....	85
5.1.4	色指定ダイアログボックス.....	86
5.1.5	文字列検索ダイアログボックス.....	86
5.1.6	インスペクト設定ダイアログボックス.....	87
5.1.7	ウォッチ設定ダイアログボックス.....	87
5.1.8	モジュールダイアログボックス.....	88
5.1.9	行番号指定ダイアログボックス.....	88
5.1.10	アドレス指定ダイアログボックス(コード).....	89
5.1.11	コマンド履歴ダイアログボックス.....	89
5.1.12	シンボル拡張ダイアログボックス.....	90
5.1.13	アドレス指定ダイアログボックス(メモリ).....	90
5.1.14	データ設定ダイアログボックス.....	91
5.1.15	レジスタの設定ダイアログボックス.....	91
5.1.16	フラグの設定ダイアログボックス.....	92
5.1.17	ブレークポイントの設定ダイアログボックス.....	92
5.1.18	フレーム番号へジャンプダイアログボックス.....	93

5.1.19	イベントブレークの設定ダイアログボックス	93
5.1.20	各種状態の設定ダイアログボックス	93
5.1.21	イベント設定ダイアログボックス	94
5.1.22	リアルタイムトレースモードダイアログボックス	95
5.1.23	CPU環境設定ダイアログボックス	96
5.1.24	エミュレーションROM設定ダイアログボックス	97

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	レジスタ表示/変更	116
7.2.9	イベントの設定	116
7.2.10	メモリ表示/変更	117
7.2.11	式の表示	117
7.2.12	シンボル表示/設定	117
7.2.13	Cのデータ参照/変更	117
7.2.14	文字列の定義(Cの式評価)	118
7.2.15	コード表示	118
7.2.16	アセンブル	118
7.2.17	バックトレース	118
7.2.18	リアルタイムトレース	118
7.2.19	システムコール	119
7.2.20	システム制御	119
7.2.21	基数変更	119
7.2.22	ロギング/バッチ	119
7.2.23	I/Oポート入力/出力	119
7.2.24	オプション設定	119
7.2.25	画面コントロール/その他	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	MS-C Ver4.0/5.1/6.0/7.0	122
8.1.1	コンパイルとリンク方法	122
8.1.2	使用例	123
8.2	Turbo C Ver1.5	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	Turbo C Ver 2.0 Turbo C++ Ver1.0 Borland C++ Ver3.1	125
8.3.1	コンパイルとリンク方法	125
8.3.2	使用例	126

8.4 LSI-C86	127
8.4.1 コンパイルとリンク方法.....	127
8.4.2 使用例.....	128
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 エラーメッセージ	133

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-V30MZ-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-V30MZ-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-V30MZ-TPの主な機能と概要

PARTNER-V30MZ-TPは、『RTE-V30MZ-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-V30MZ-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-V30MZ-TPマニュアル(本書)

本マニュアルでは、PARTNER-V30MZ-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-V30MZ-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-V30MZ-TPを使用するには、ハードディスクが必要です。PARTNER-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-TP

『RTE-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-TP

PARTNER-V30MZ-TPをお使いになるに当たり、PARTNER-V30MZ-TPの特徴とデバッグの手順を簡単に説明します。

1.1 PARTNER-V30MZ-TPの特徴

PARTNER-V30MZ-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、ブレークポイントの設定、プログラムの実行、変数のインスペクトなど、日常使われる機能を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。

1.2 PARTNER-V30MZ-TPの基本的な操作手順

PARTNER-V30MZ-TPの基本的な操作手順を以下に示します。

セットアップ

PARTNER-V30MZ-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』(11頁)

環境設定

PARTNER-V30MZ-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-V30MZ-TP環境設定』(16頁)

環境設定は、PARTNER-V30MZ-TP環境設定用プログラムで行います。(図 1)



図 1 PARTNER-V30MZ-TP環境設定プログラム

起動

環境設定が完了した後、PARTNER-V30MZ-TPを起動します。(図 2)

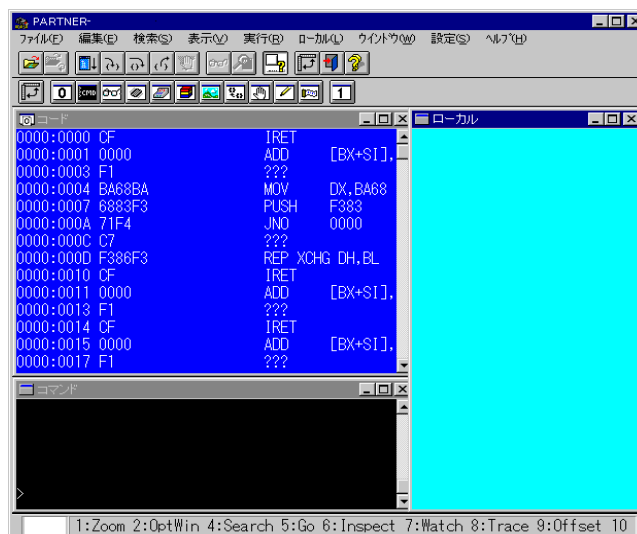



図 2 PARTNER-V30MZ-TP起動画面

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(78頁),Lコマンド,[ファイル]-[ロード]コマンド(33頁)

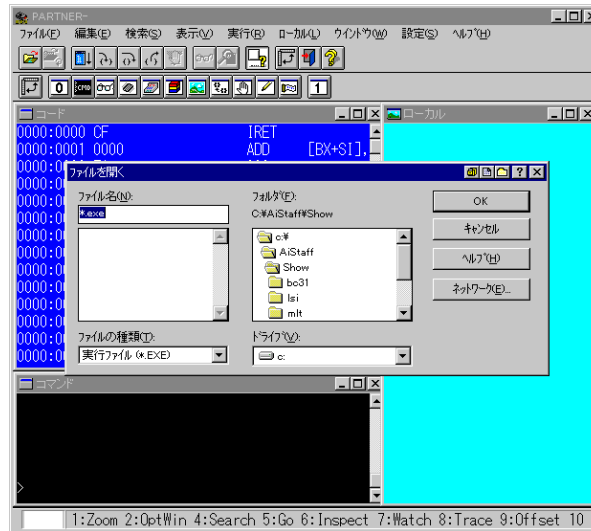


図 3 デバッグ対象ファイルのロード

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『4.5.3 コードウィンドウでのマウス操作』(53頁),[ブレーク]-[ブレーク設定](69頁)

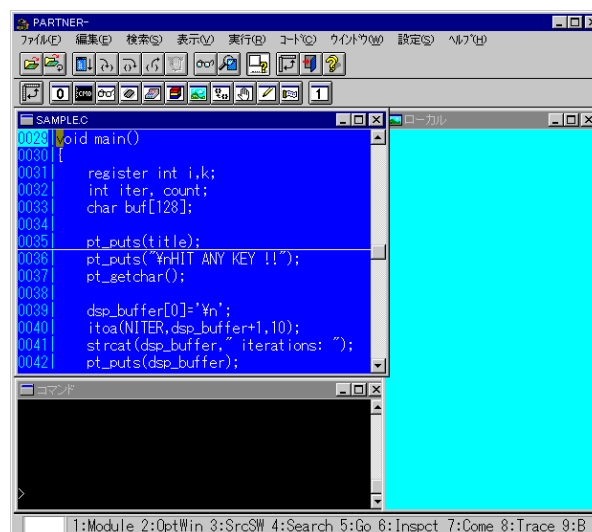


図 4 ブレークポイントの設定

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(78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

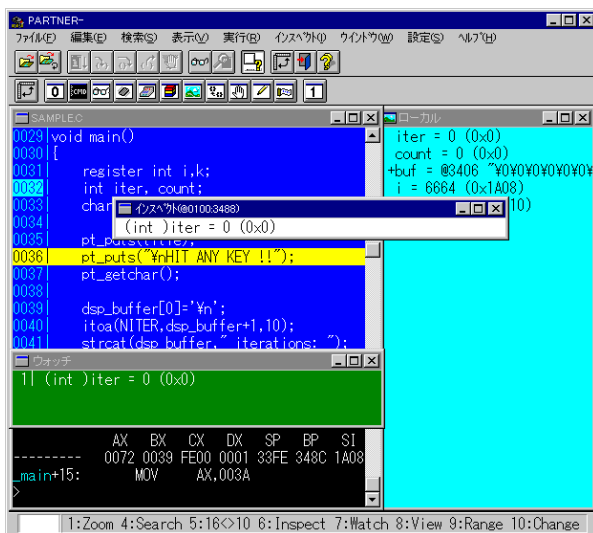


図 6 iterのウォッチ登録

PARTNER-V30MZ-TPの終了

PARTNER-V30MZ-TPを終了します。(図 7)

【参照】  ボタン(78頁), [ファイル]-[終了](33頁), Q/EXITコマンド

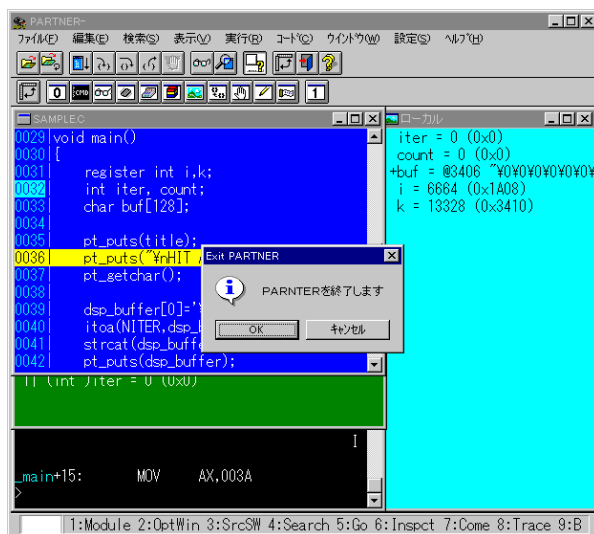


図 7 PARTNER-V30MZ-TPの終了確認

2 セットアップ

PARTNER-V30MZ-TPを使用するには、ご使用のホストパソコンにPARTNER-V30MZ-TPをインストールする必要があります。この章では、PARTNER-V30MZ-TPのインストールの方法について説明します。

2.1 セットアップの前に

PARTNER-V30MZ-TPのセットアップを始める前に、RTE-V30MZ-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-V30MZ-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-V30MZ-TPの設定

RTE-V30MZ-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-V30MZ-TP用DLLのインストール

RTE-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-TPのセットアップを行います。

【注意】PARTNER-V30MZ-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、各ファイルを単にハードディスクにコピーしただけでは、PARTNER-V30MZ-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-V30MZ-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-V30MZ-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のファイルをお読みください。

2.2.2 PARTNER-V30MZ-TPのインストール

PARTNER-V30MZ-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(V30MZ)」というショートカットが作成されます。

【注意】PARTNER-V30MZ-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-V30MZ-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-V30MZ-TP」を一覧から選んで削除します。

空白ページ

3 起動

PARTNER-V30MZ-TPを起動するには、PARTNER-V30MZ-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(V30MZ)をダブルクリックして起動し、PARTNER-V30MZ-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(V30MZ)が作成されています。

3.1 PARTNER-V30MZ-TP環境設定

PARTNER-V30MZ-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)
- 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-V30MZ-TPを起動することができます。RPTSETUP(V30MZ)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(V30MZ)は、PARTNER-V30MZ-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(V30MZ)を起動し、PARTNER-V30MZ-TPの実行環境の作成、変更を行います。(図 8)



図 8 RPTSETUP(V30MZ)起動画面

『新規に環境を作成する場合』及び『RPTSETUP(V30MZ)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(V30MZ)では使用しません。

新規にPARTNER-V30MZ-TPの環境を作成する場合

新規プロジェクトの作成



PARTNER-V30MZ-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(RPTV30T.KPJ)、コンフィグファイル(RPTV30T.CFG)が作成(Binディレクトリからのコピー)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(RPTV30T.CFG)、プロジェクトファイル(RPTV30T.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。

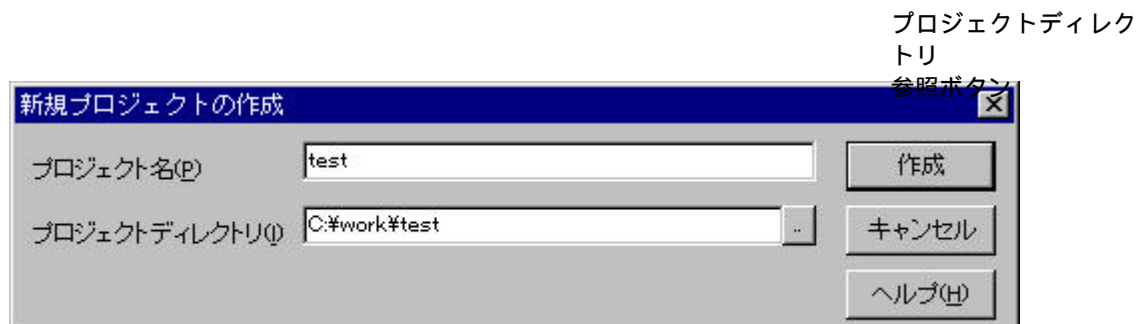


図 9 c:%work%testディレクトリを指定

コンフィグファイルの設定



このコマンドは、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が、PARTNER-V30MZ-TPでは、コンフィグレーション項目がありません。そのため、デフォルトのコンフィグファイルをそのまま使用し、編集する必要はありません。コンフィグファイルは将来の拡張のために用意されています。



起動オプションの設定

PARTNER-V30MZ-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)

デバッグ情報サイズ(B)	1024	了解
マクロバッファサイズ(M)	64	キャンセル
拡張子(E)	C	ヘルプ(H)
タブ(T)	8	
デバッグ情報モード(D)	Assembler	
メモリモデル(B)	Small	
ソースパス(P)	C:\work\test	
<input type="checkbox"/> 強制起動モード(F)		
-MS -X -B1024,64 -EC -TAB8 -SD "C:\work\test"		

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-V30MZ-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

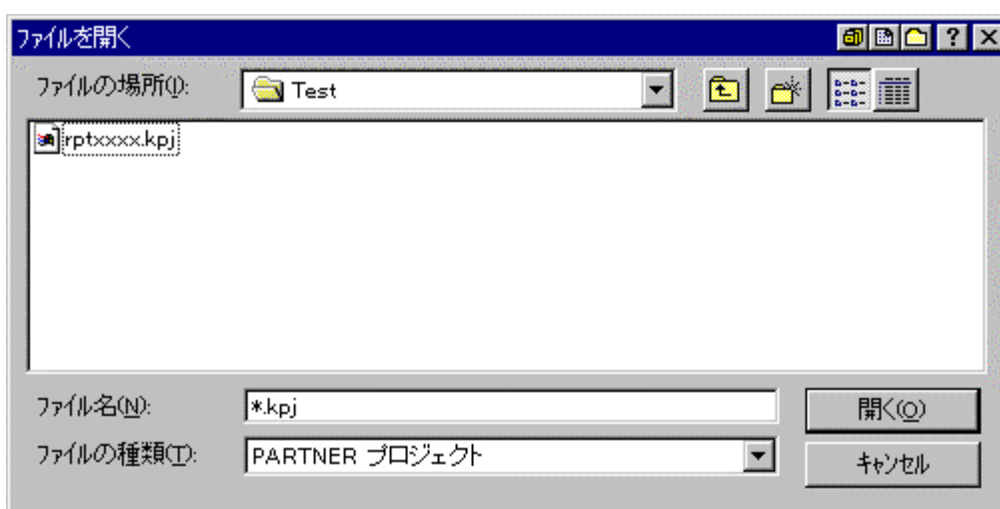
既存のプロジェクトを変更

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(RPTV30MZ.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-V30MZ-TPは、変更した内容で起動します。

3.2 環境ファイル(RPTV30T.CFG)

デバッガプログラムPARTNER-V30MZ-TPは、起動時に環境ファイル(RPTV30T.CFG)を読み込みます。この環境ファイルは、PARTNER-V30MZ-TPの動作環境を予め設定しておくためのものですが、現在は設定項目がないため、このファイルには、何も記述されていません。

PARTNER-V30MZ-TPでは、将来の拡張のために用意されているファイルです。

新規プロジェクトで作成されたデフォルトの状態で使用してください。

なお、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、コメント行になります。

3.3初期設定コマンド

RTE-V30MZ-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-V30MZ-TPの内部コマンドです。コマンドについては、ハードウェア・ユーザズ・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、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-V30MZ-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-V30MZ-TP、PARTNER-V30MZ-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 80000 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80000H~FFFFFFH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-V30MZ-TPが使用するファイルは以下の通りです。

3.4.1 PARTNER-V30MZ-TP起動時に必要なファイル

PARTNER-V30MZ-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-V30MZ-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(RPTV30T.CFG)

コンフィグファイルは、PARTNER-V30MZ-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(RPTV30T.KPJ)


プロジェクトファイルは、PARTNER-V30MZ-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-V30MZ-TPが起動された時、自動的に取り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-V30MZ-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-V30MZ-TP終了時に作成するファイル

PARTNER-V30MZ-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-V30MZ-TPの起動

PARTNER-V30MZ-TPを起動するには、RPTSETUP(V30MZ)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-V30MZ-TPが正常に起動した場合

PARTNER-V30MZ-TPを起動すると図 12の画面が表示されます。

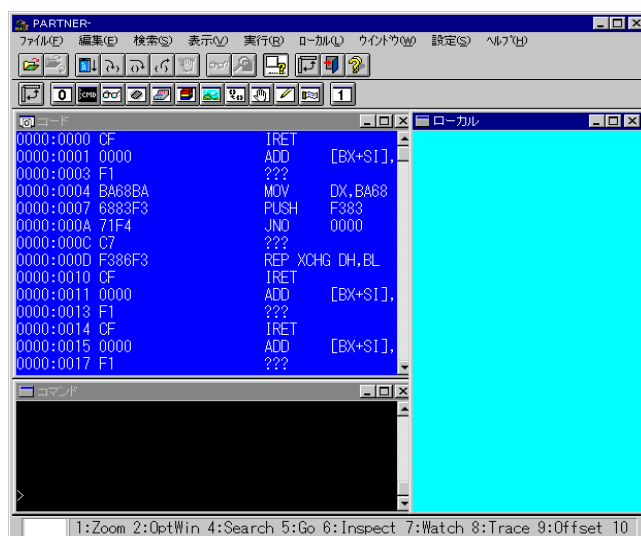


図 12 PARTNER-V30MZ-TPの起動画面

3.5.2 PARTNER-V30MZ-TPが起動しない場合

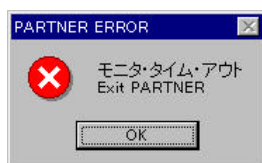
起動時に表示するエラーは以下のとおりです。



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。



初期化中にRTEからの応答がありません。ターゲットシステムが接続されている場合、プロセッサが動作できない状態にある可能性があります。障害を取り除いてから起動してください。



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。



パケット送受信のタイムアウトエラーです。RTEの接続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 DISK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-V30MZ-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、各デバッグウィンドウなど PARTNER-V30MZ-TPの主な画面の構成要素を説明します。

4.1.1 構成要素

ウィンドウ要素を図 13で表します。

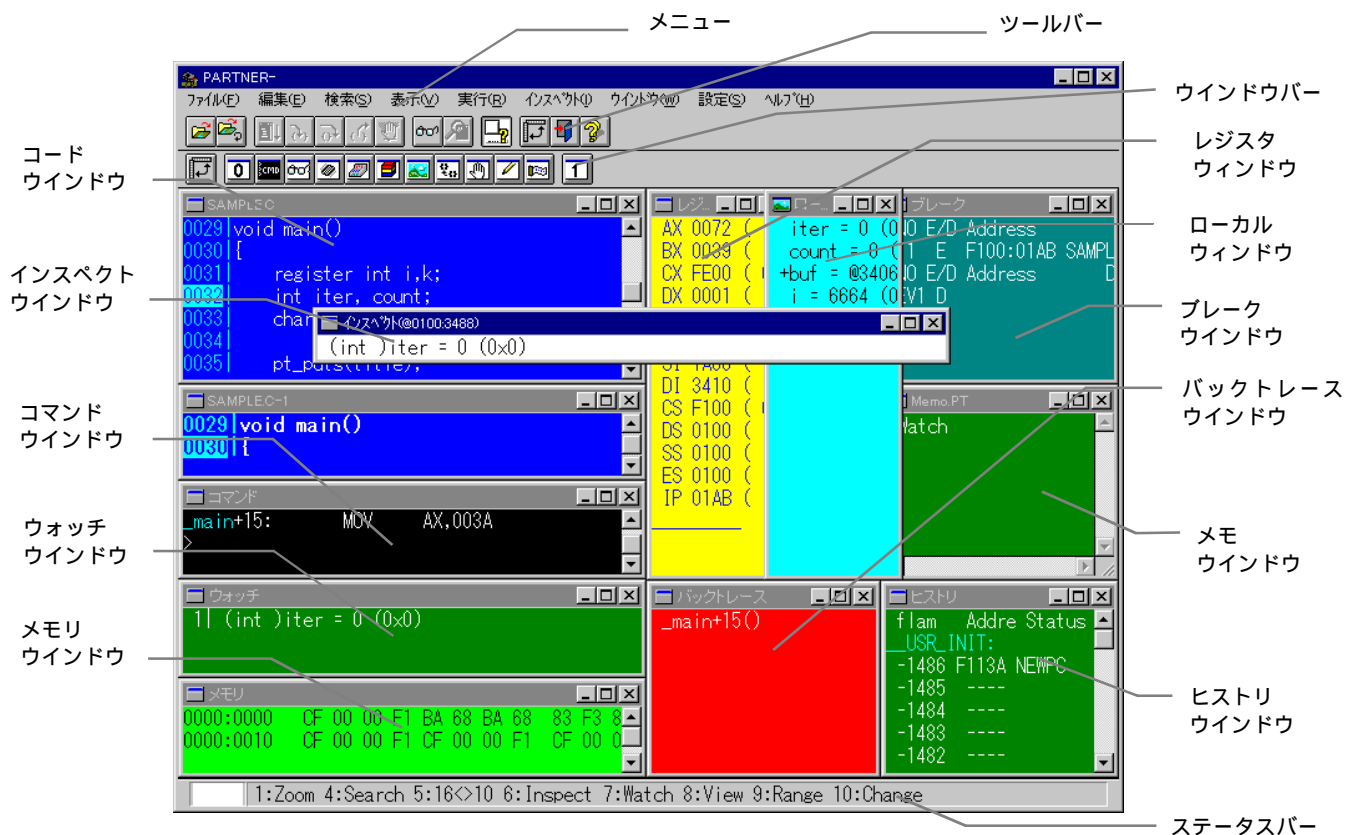
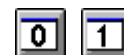


図 13 画面要素

メインウィンドウ

PARTNER-V30MZ-TPのメインウィンドウです。このウィ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停止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停止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停止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停止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停止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-V30MZ-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。

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(>)の表示された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。

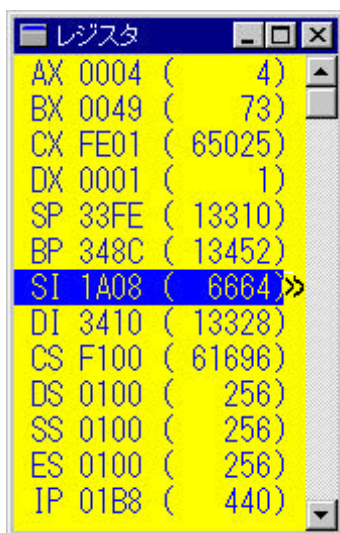
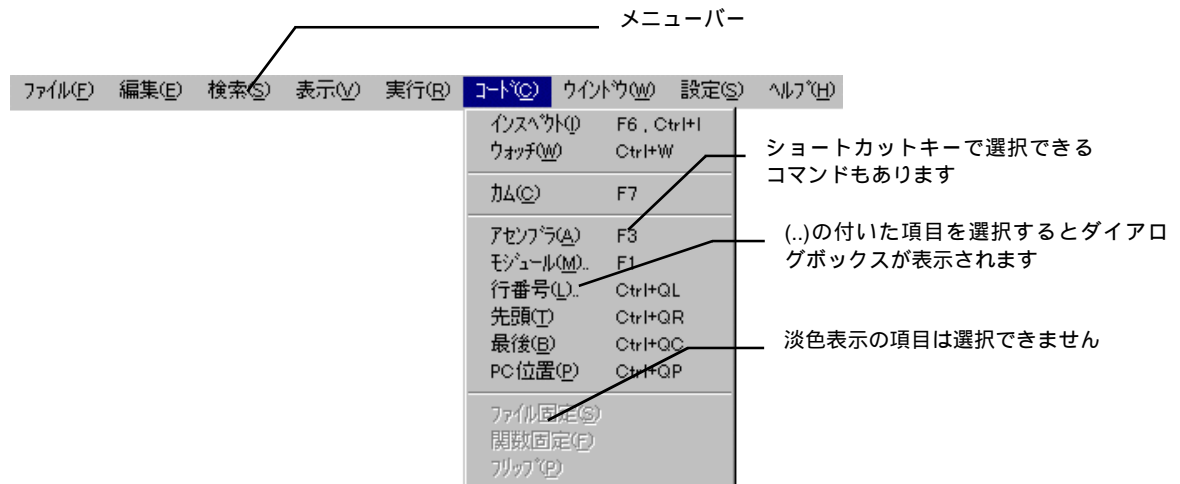


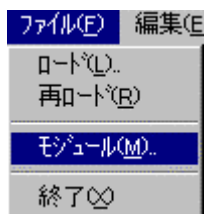
図 14 非表示部分に検索文字列がある場合

4.2 メニュー

メニューには、PARTNER-V30MZ-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無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』84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』88頁参照)
終了(X)	PARTNER-V30MZ-TPを終了します

4.2.2 編集メニュー



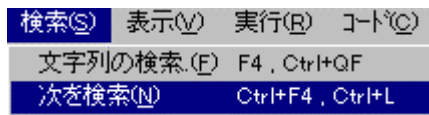
[編集]メニューにはウインドウ間でクリップボードを経由して文字列のやりとりを行うコマンドが収められています。



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(Q)	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

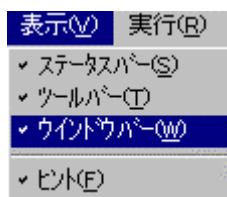
4.2.3 検索メニュー



[検索]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』86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]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(『4.19 ステータスバー』81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(『4.17 ツールバー』77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(『4.18 ウィンドウバー』80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(P)	F9
イベントブレーク(H)	
リアルタイムトレースモード(E)	
イベント設定(R)	
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[ブレークポイント]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、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
イベントブレーク(H)	イベントブレーク設定ダイアログボックスが表示されます。(『5.1.22 リアルタイムトレースモードダイアログボックス』95頁参照)
リアルタイムトレースモード(E)	リアルタイムトレースモード設定ダイアログボックスが表示されます。(『5.1.22 リアルタイムトレースモードダイアログボックス』95頁参照)
イベント設定(R)	イベントブレーク/リアルタイムトレースで使用するイベントの設定ダイアログボックスが表示されます。(『5.1.21 イベント設定ダイアログボックス』94頁参照)
強制ブレーク(B)	強制的にユーザプログラムを中断し、PARTNER-V30MZ-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)	ウインドウ(W)	設定(S)
インサバウト(I)	F6, Ctrl+I	
ウォッチ(W)	Ctrl+W	
加減(Q)	F7	
アセンブラ(A)	F8	
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。

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)	ウインドウ(W)	設定(S)
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。

メモリメニュー

メモリ(M)	ウインドウ(W)	設定(S)
✓ バイト表示(B)		B
ワード表示(W)		W
ダブルワード表示(D)		D
ショートポート(S)		S, F
ロングポート(L)		L
テンポラリポート表示(T)		T
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。

このメニューコマンドの内容は『4.7.2 メモリウイ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)	ウインドウ(W)	設定(S)
初期化(C)		F3
インクリメント(+)		
デクリメント(-)		
値の変更(O)		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『4.8.2 レジスタウインドウのローカルメニュー(60頁)』を参照してください。

スタックメニュー

スタック(S)	ウインドウ(W)
✓ 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『4.9.2 スタックウインドウのローカルメニュー(61頁)』を参照してください。

ローカルメニュー

ローカル(L)	ウインドウ(W)	設定(S)
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(63頁)』を参照してください。

ウォッチメニュー

ウォッチ(W)	ウインドウ(W)	設定(S)
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(67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)	ウインドウ(W)
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(65頁)』を参照してください。

ブレイクメニュー

ブレイク(B)	ウインドウ(W)	設定(S)
ブレイク設定(A)		Ins
禁止(D)		F9, Enter
削除(C)		Del
全て削除(Q)		F3
設定のセーブ(S)		F4
設定のリート(L)		F6

[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(69頁)』を参照してください。

メモメニュー

メモ(M)	ウインドウ(W)	設定(S)
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。

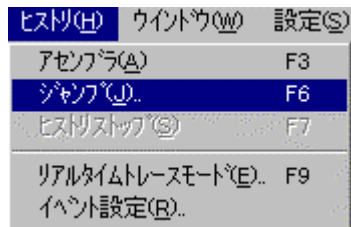
インスペクトメニュー

インスペクト(I)	ウインドウ(W)	設定(S)
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。

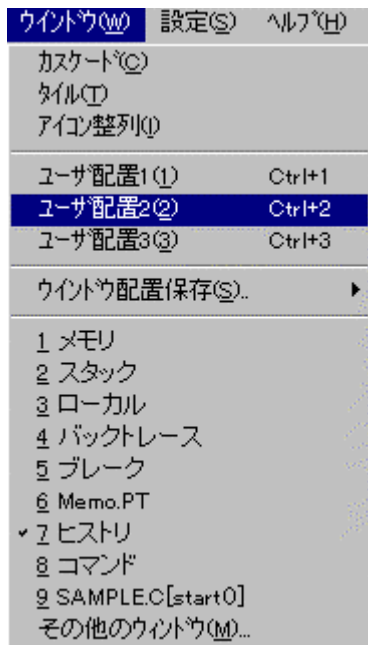
ヒストリメニュー



[ヒストリ]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 ヒストリウインドウ (70頁)』を参照してください。

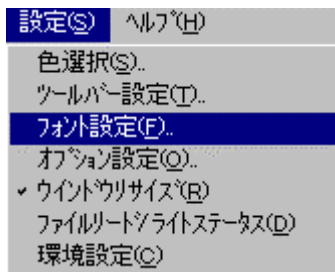
4.2.7 ウィンドウメニュー



[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-V30MZ-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

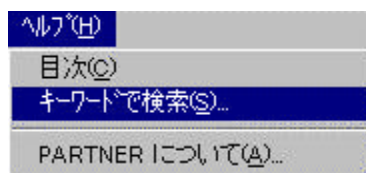
4.2.8 設定メニュー



[設定]メニューにはPARTNER-V30MZ-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』85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』85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』84頁参照)
オプション設定(O)	PARTNER-V30MZ-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-V30MZ-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-V30MZ-TPのヘルプの目次を表示します
キーワードで検索(S)	PARTNER-V30MZ-TPのヘルプをキーワードで検索します
PARTNERについて(A)	PARTNER-V30MZ-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-V30MZ-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 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
F2	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します
CTRL+O	
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	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』86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+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	一画面分上を表示します
ROLL DOWN	
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面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	61
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	62
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	64
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	66
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
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	68
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

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)

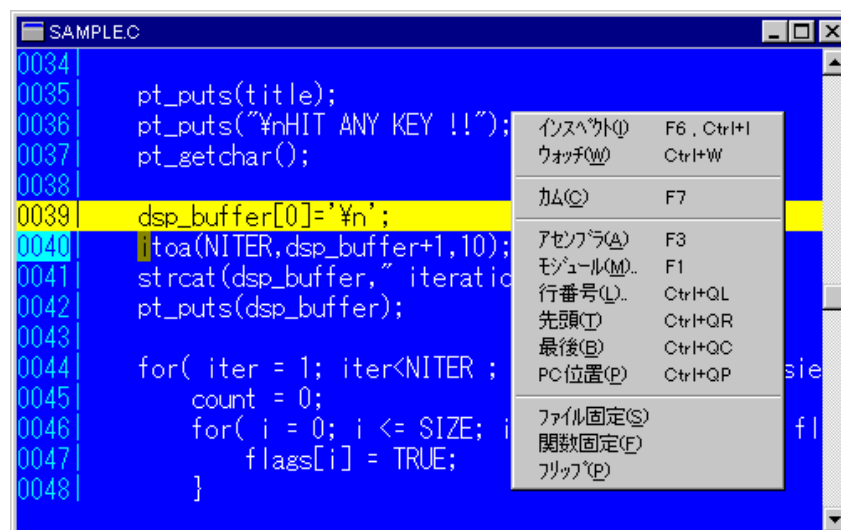


図 15 コードウインドウのローカルメニュー

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。

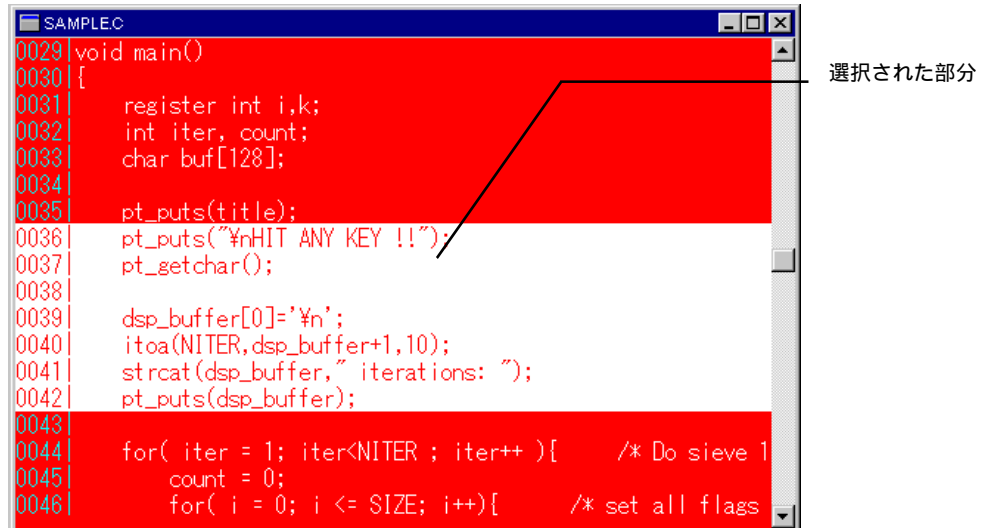


図 16 コピー範囲の選択中

【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 ウインドウ独自のマウス操作

『4.4.1 各ウインドウ共通のマウス操作』とは別に、ウインドウ独自のマウス操作が定義されています。それぞれのウイ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。

ウインドウ	見出し	ページ
コードウインドウ	4.5.3 コードウインドウでのマウス操作	53
メモリウインドウ	4.7.3 メモリウインドウでのマウス操作	58
レジスタウインドウ	4.8.3 レジスタウインドウでのマウス操作	60
ローカルウインドウ	4.10.3 ローカルウインドウでのマウス操作	63
バックトレースウインドウ	4.11.3 バックトレースウインドウでのマウス操作	65
ウォッチウインドウ	4.12.3 ウォッチウインドウでのマウス操作	67
ブレークウインドウ	4.13.3 ブレークウインドウでのマウス操作	69
インスペクトウインドウ	4.15.3 インスペクトウイ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。

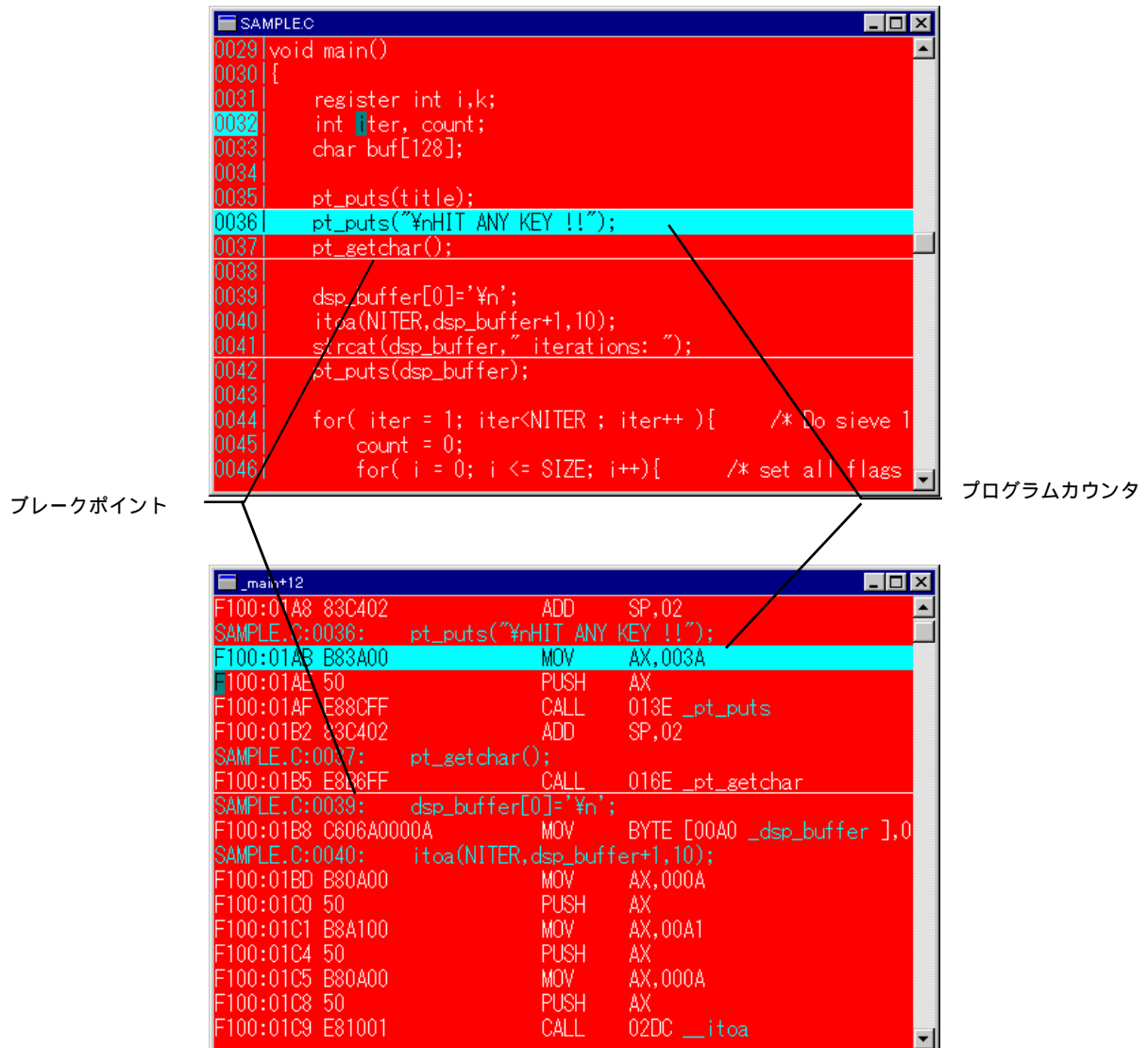


図 17 コードウインドウ

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』88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 / 解除をおこないます	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)	ウインドウ(W)	設定(S)
インスペクト(I)	F6, Ctrl+I	
ウォッチ(W)	Ctrl+W	
加減(C)	F7	
アセンブラ(A)	F8	
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』87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』87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』88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』88頁, 『5.1.10 アドレス指定ダイアログボックス(コード)』89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、常に使用するコマンドがマウス操作に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
>d 0 64
0100:0000 0A 50 41 52 54 4E 45 52 2D 45 54 32 20 53 41 4D .F
0100:0010 50 4C 45 20 50 52 4F 47 52 41 4D 20 28 63 29 20 PL
0100:0020 31 39 38 38 20 4B 79 6F 74 6F 20 4D 69 63 72 6F 19
0100:0030 20 43 6F 6D 70 75 74 65 72 00 0A 48 49 54 20 41 0
0100:0040 4E 59 20 4B 45 59 20 21 21 00 20 69 74 65 72 61 N
0100:0050 74 69 6F 6E 73 3A 20 00 0A 50 72 69 6D 65 3D 00 t
0100:0060 00 00 D0 00 D0 00 D0 00 00 00 00 00 35 F1 00 00 .
>
    
```

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』89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』90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、カーソル位置に関係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に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』89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』90頁参照)

4.7 メモリウインドウ

メモリウインドウは、メモリの内容を各種表示形式で表示します。

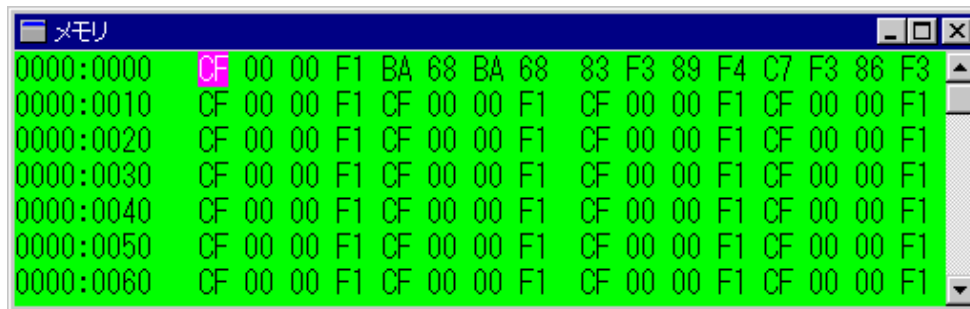


図 19 メモリウインドウ

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)』90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』91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)	ウインドウ(W)	設定(S)
▼ 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
テンポラリフロート表示(T)		T
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)』90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』91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』91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、レジスタの内容、フラグの内容を表示します。また、レジスタに関連する情報も表示します。

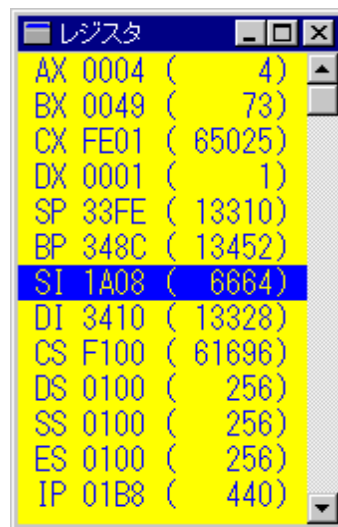


図 20 レジスタウインドウ

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』91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]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』91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』91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。

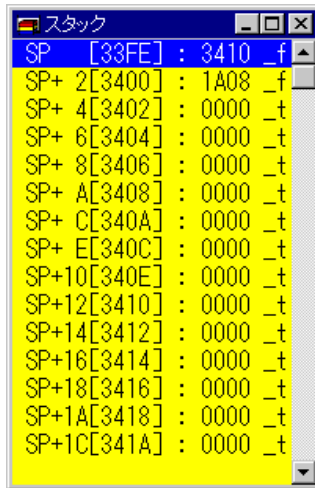


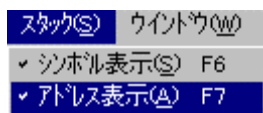
図 21 スタックウインドウ

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)	スタックメモリアドレス表示をします

4.10 ローカルウィンドウ

ローカルウィンドウは、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。

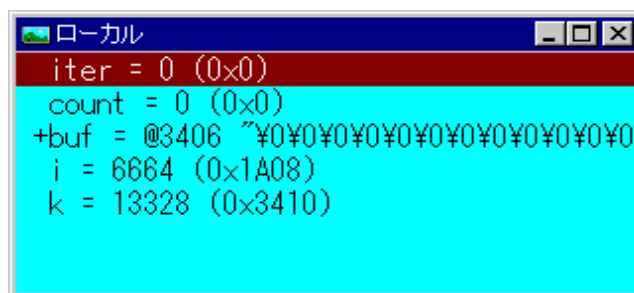


図 22 ローカルウィンドウ

4.10.1 ローカルウィンドウのショートカットキー

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

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]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 / 非表示の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 / 非表示の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。

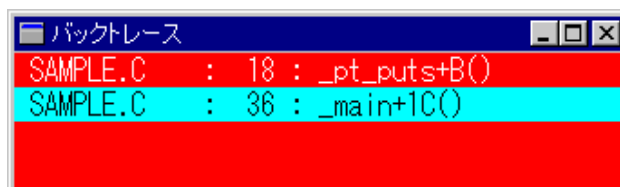


図 23 バックトレースウインドウ

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)	ウインドウ(W)
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。

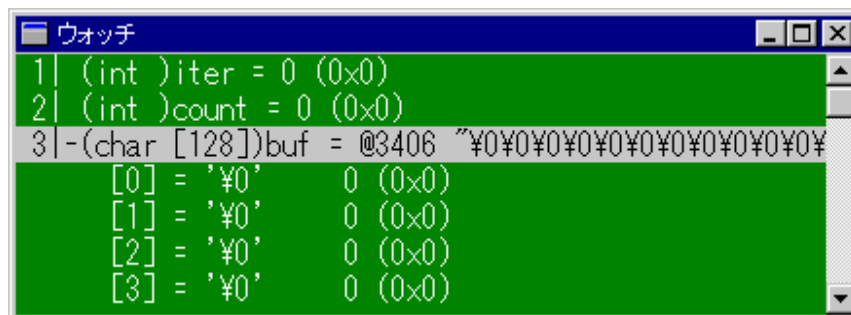


図 24 ウォッチウインドウ

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)	ウインドウ(W)	設定(S)
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 ブレイクウインドウ

ブレイクウインドウは、現在設定されているブレイクポイントを表示します。

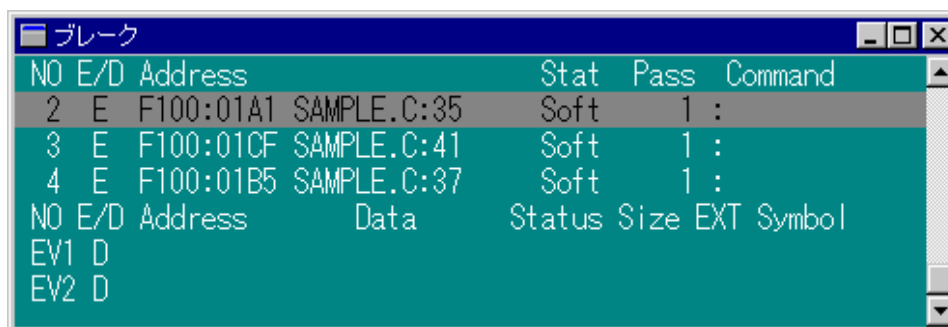


図 25 ブレイクウインドウ

4.13.1 ブレイクウインドウのショートカットキー

ブレイクウインドウでは、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、ブレイ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
F3	全てのブレイクポイントを削除します
F4	現在のブレイクポイントの設定を保存します
F6	F4で保存したブレイクポイントの設定を呼び出します
F9	現在選択されているブレイクポイントの許可/禁止を切り替えます
Enter	
INS	ブレイクポイントの設定ダイアログボックスを表示します(『5.1.17 ブレイクポイントの設定ダイアログボックス』92頁参照)
DEL	現在選択されているブレイクポイントを削除します

4.13.2 ブレイクウィンドウのローカルメニュー

ブレイク(B)	ウィンドウ(W)	設定(S)
ブレイク設定(A)	Ins	
禁止(D)	F9,Enter	
削除(C)	Del	
全て削除(L)	F3	
設定のセーブ(S)	F4	
設定のロード(L)	F6	

[ブレイク]メニューには、ブレイ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
ブレイク設定(A)	新規にブレイ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(『5.1.17 ブレイクポイントの設定ダイアログボックス』92頁参照)
禁止(D)/許可(E)	カーソルのあるブレイクポイントの許可/禁止の切り替えをします
削除(C)	カーソルのあるブレイクポイントを削除します
全て削除(L)	全てのブレイクポイントを削除します
設定のセーブ(S)	現在のブレイ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イクポイントの設定をロードします

4.13.3 ブレイクウィンドウでのマウス操作

ブレイクポイント部分を左ダブルクリックすると、ブレイクポイントの有効/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。

The screenshot shows a window titled 'ヒストリ' (History) with a green background. It displays a list of instructions in a table format. The columns are: flam, Adresse, Status, EXT, SEG, :OFS, DATA, and INSTRUCTION. The instructions are disassembled from their hexadecimal representation. The current instruction being executed is highlighted in white.

flam	Adresse	Status	EXT	SEG	:OFS	DATA	INSTRUCTION
SAMPLE.C:0018:							while(*p) _pt_dsp(*(p++));
-388	F1149	NEWPC	@1111	F100:0149	8B5E04		MOV BX,04[BP] p
-387	----						
-386	----						
-385	----						
-384	----						
-383	----						
-382	----						
-381	F114C	ISC(2)	@1111	F100:014C	803F00		CMP BYTE [BX],00
			@1111	F100:014F	7503		JNZ 0154
-380	F1154	NEWPC	@1111	F100:0154	8B5E04		MOV BX,04[BP] p
-379	----						
-378	----						
-377	F1157	ISC(5)	@1111	F100:0157	834604		ADD WORD 04[BP] p ,01
			@1111	F100:015B	8A07		MOV AL,[BX]
			@1111	F100:015D	98		CBW
			@1111	F100:015E	50		PUSH AX
			@1111	F100:015F	E8BFFF		CALL 0121 _pt_dsp
_pt_dsp:							
-376	F1121	NEWPC	@1111	F100:0121	55		PUSH BP

図 26 ヒストリ(逆アセンブル表示)

The screenshot shows a window titled 'ヒストリ' (History) with a green background. It displays a list of instructions in a table format. The columns are: flam, Adresse, Status, and EXT. The instructions are shown as hexadecimal values. The current instruction being executed is highlighted in white.

flam	Adresse	Status	EXT
-388	F1149	NEWPC	@1111
-387	----		
-386	----		
-385	----		
-384	----		
-383	----		
-382	----		
-381	F114C	ISC(2)	@1111
-380	F1154	NEWPC	@1111
-379	----		
-378	----		
-377	F1157	ISC(5)	@1111
-376	F1121	NEWPC	@1111
-375	----		
-374	----		
-373	F1122	ISC(4)	@1111
-372	F1138	NEWPC	@1111
-371	----		
-370	----		
-369	F1139	ISC(1)	@1111

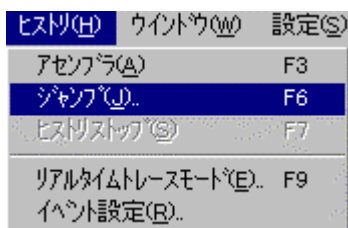
図 27 ヒストリ(ダンプ表示)

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(『5.1.18 フレーム番号へジャンプダイアログボックス』93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
F9	リアルタイムトレースイベントの設定ダイアログボックスを表示します(『5.1.22 リアルタイムトレースモードダイアログボックス』95頁参照)

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]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(『5.1.18 フレーム番号へジャンプダイアログボックス』93頁参照)
ヒストリ実行(S)/ヒストリストップ(I)	ヒストリの実行/停止を切り替えます
リアルタイムトレースモード (E)	リアルタイムトレースモード設定のダイアログボックスを表示します(『5.1.22 リアルタイムトレースモードダイアログボックス』95頁参照)
イベント設定(R)	イベント設定ダイアログボックスを表示します(『5.1.21 イベント設定ダイアログボックス』94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。

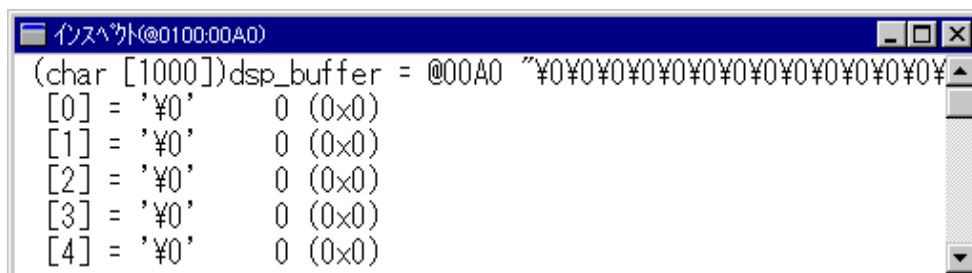


図 28 インスペクトウインドウ

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	ウインドウ	設定
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-V30MZ-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。

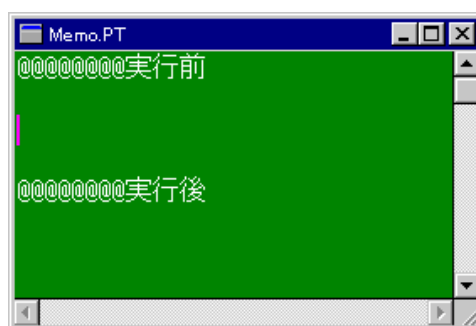


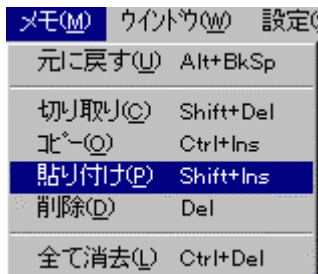
図 29 メモウインドウ

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]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 ツールバー



図 30 ツールバーの例

ここでは、PARTNER-V30MZ-TPのツールバーについて説明します。

ツールバーはPARTNER-V30MZ-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-V30MZ-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)

4.17.1 ツールバーの設定

















図 31 [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』85頁参照)


4.17.2 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』84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-V30MZ-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(『5.1.7 ウォッチ設定ダイアログボックス』87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(『5.1.6 インスペクト設定ダイアログボックス』87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』85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』84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』86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』88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-V30MZ-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』85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-V30MZ-TPのヘルプの表示
- 

PARTNER-V30MZ-TPを終了します

4.18 ウィンドウバー



図 32 ウィンドウバー

ここでは、PARTNER-V30MZ-TPのウィンドウバーについて説明します。ウィンドウバーはPARTNER-V30MZ-TP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』35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
	ブレイクウィンドウ
	ヒストリウィンドウ
	メモウィンドウ
	コードウィンドウ1

【注意】ウィンドウバーのボタン設定は変更できません。

4.19 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-V30MZ-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)

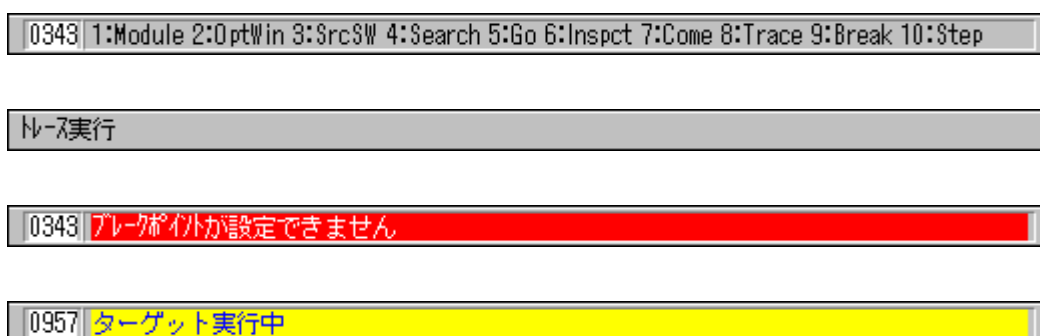


図 33 ステータスバーの例

空白ページ


5 ダイアログコマンド

PARTNER-V30MZ-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。

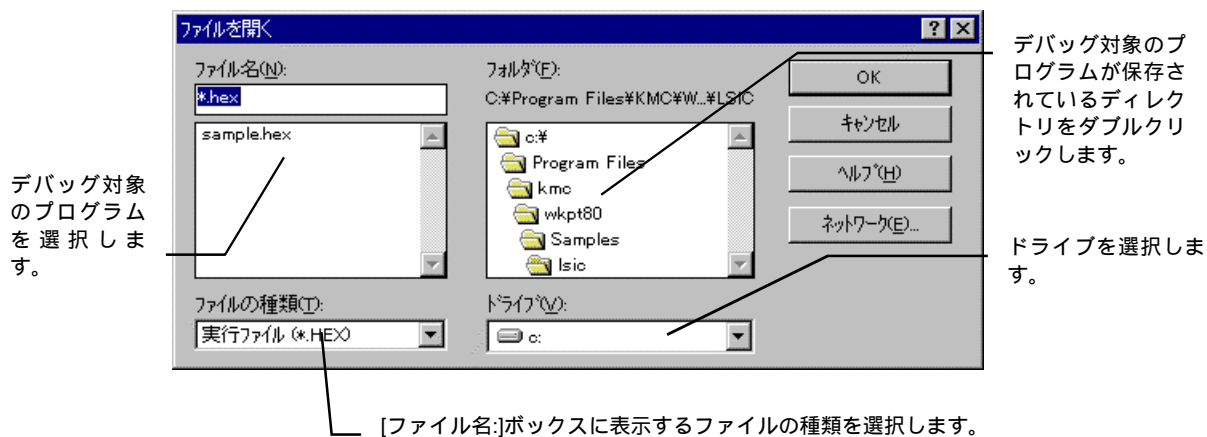
5.1 ダイアログボックス

PARTNER-V30MZ-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。


使用するフォント名を指定します。



使用するフォントのスタイルを指定します。

使用するフォントの大きさを指定します。

5.1.3 ツールバーの設定ダイアログボックス

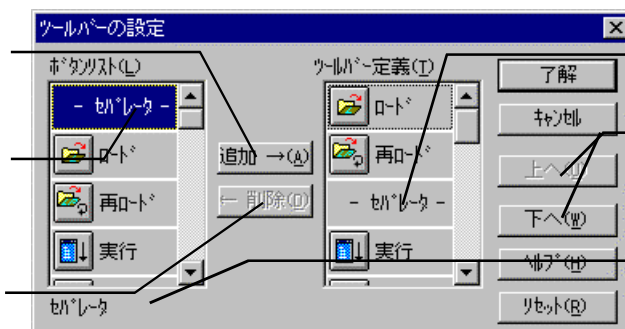
[設定]-[ツールバー設定], 

[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。

選択したボタンをツールバーに追加します。

登録するボタンを指定します。

選択したボタンをツールバーから削除します。



ボタンの挿入位置を指定します

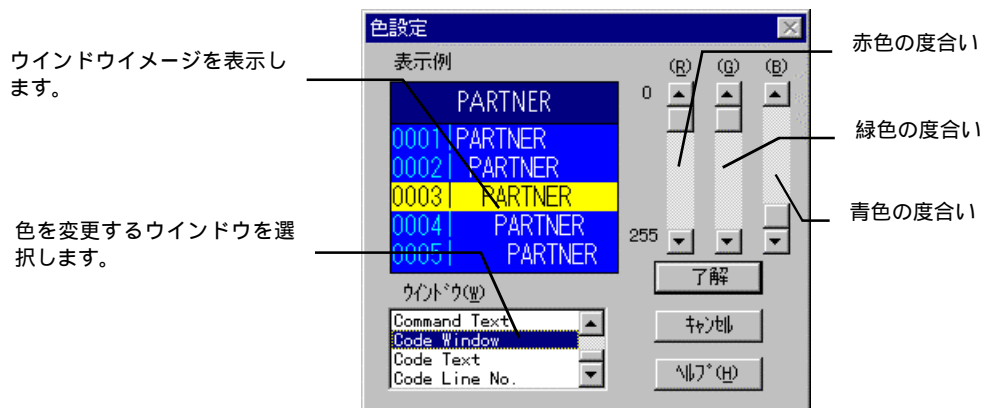
選択したボタンを上下に位置移動します。

選択されているボタンの説明が表示されます。

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は初期状態は[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], 

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。



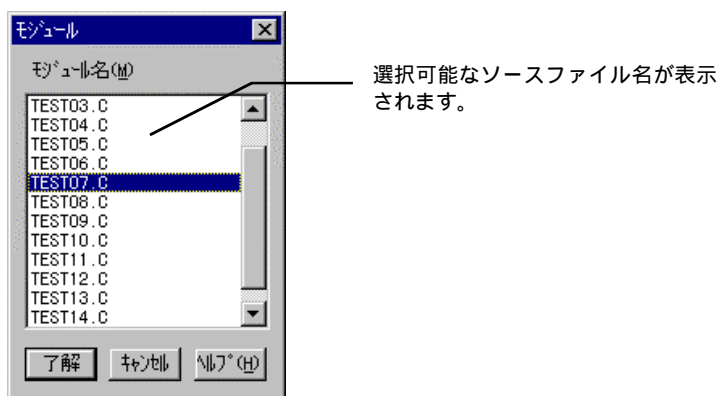
ウォッチ登録する変数名を入力します。

【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『7.2.13 Cのデータ参照/変更』117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]

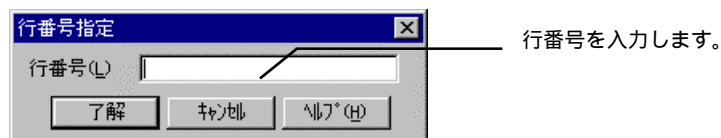
コードウインドウに表示するソースファイルを指定します。[モジュール名]リストボックスには、ロードしたデバッグ対象ファイルのデバッグ情報に定義されているソースファイル名と、Vコマンドで参照したファイル名が表示されます。



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]

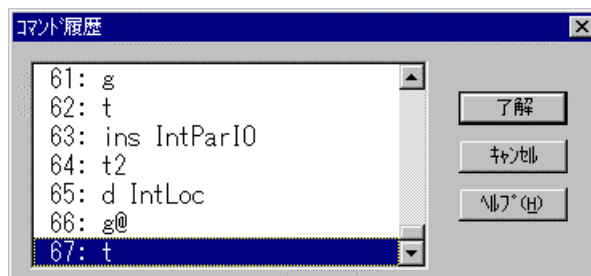
コードウインドウに表示する逆アセンブルリストの表示開始アドレスを指定します。



5.1.11 コマンド履歴ダイアログボックス

[コマンド]-[履歴]

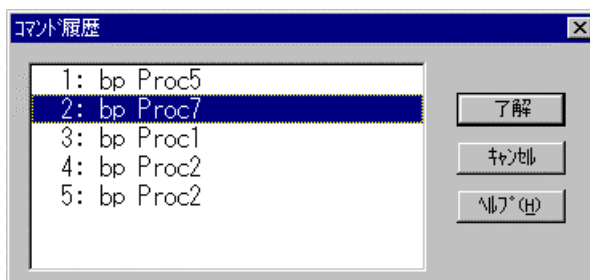
コマンドウインドウに入力されたコマンドの履歴を表示し、選択します。
 コマンドウインドウのコマンド入力行(現在のプロンプト)に文字列が入力されている場合は、その文字列で始まる文字列をリストボックスに表示します。



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]

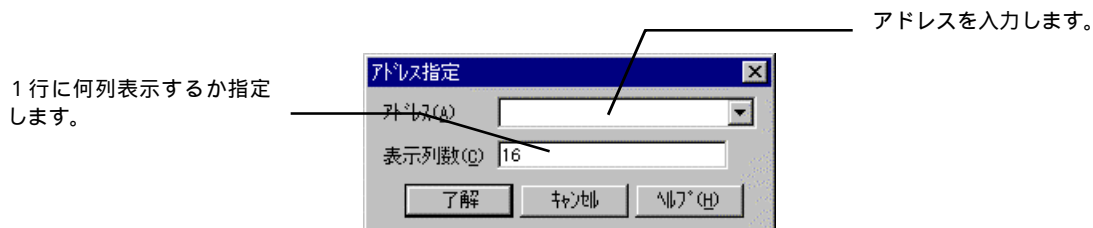
コマンド入力行の最後の文字列から始まるグローバルシンボルの拡張表示をリストボックスにします。(下の例はコマンド入力行に”bp Proc”と入力した場合です。)



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]

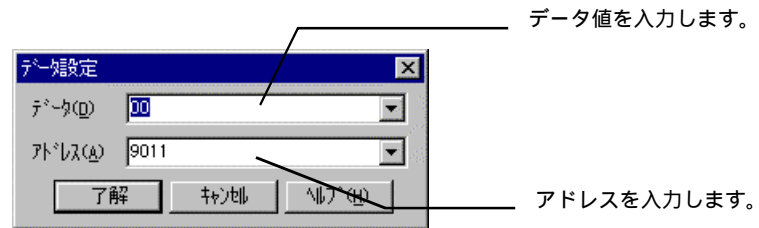
メモリウィンドウに表示するダンプリストの表示開始アドレスを指定します。



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]

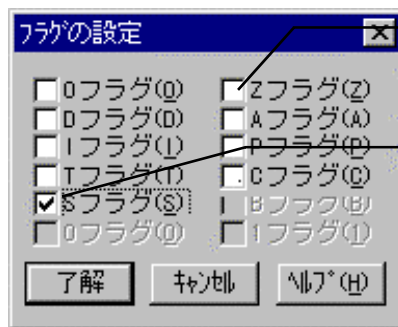
レジスタの値を変更するダイアログボックスです。



5.1.16 フラグの設定ダイアログボックス

[レジスタ]-[値の変更]

フラグの値を変更するダイアログボックスです。



チェックされていないフラグの値は0です。

チェックされているフラグの値は1です。

5.1.17 ブレークポイントの設定ダイアログボックス

[ブレーク]-[ブレークポイント]

ブレークポイントを設定するダイアログボックスです。

パスカウントを入力します。



アドレスを入力します。

ブレーク後、実行するコマンドを入力します。

5.1.18 フレーム番号へジャンプダイアログボックス

[履歴]-[ジャンプ]

履歴ウィンドウで表示するトレースバッファの先頭フレームを指定します。



フレーム番号を入力します。

5.1.19 イベントブレークの設定ダイアログボックス

[実行]-[イベントブレーク]

イベントブレークを設定するダイアログボックスです。

イベントブレーク
に使用するイベン
トを選択します。



イベント設定ダイア
ログボックスを表示
します。

5.1.20 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-V30MZ-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。

5.1.21 イベント設定ダイアログボックス

[実行]-[イベント設定],[履歴]-[イベント設定]

イベントブ레이크、リアルタイムトレースで使用するイベント条件を設定します。

設定するイベント番号を選択します。

データ条件を入力します。

CPUステータス条件を選択します。

The screenshot shows the 'イベント設定' (Event Setting) dialog box. It contains several sections for configuring event conditions:

- イベント番号 (Event Number):** Radio buttons for 'イベント1 (E)' (selected) and 'イベント2 (E)'.
- アドレス (Address):** A text input field containing 'P:XXXXX'.
- データ (Data):** A text input field containing 'XXXX'.
- サイズ (Size):** Radio buttons for '8ビット (B)' (selected) and '16ビット (B)'.
- CPUステータス (CPU Status):** A grid of radio buttons for various CPU events: 'メモリアクセス (M)', 'メモリード (R)', 'メモライト (W)', 'I/Oアクセス (I)', 'I/Oド (D)', 'I/Oライト (T)', '実行 (E)', '割り込みAck (A)', and '全て (A)'.
- AUX (AUX):** Radio buttons for '無効 (N)' (selected), 'Low (L)', and 'High (H)'.

At the bottom, there are three buttons: '設定' (Set), 'キャンセル' (Cancel), and 'ヘルプ (H)' (Help).

アドレス条件を入力します。

データ条件のデータサイズを指定します。

AUXのレベル条件を指定します。

5.1.22 リアルタイムトレースモードダイアログボックス

[実行]-[リアルタイムトレースモード],[ヒストリ]-[リアルタイムトレースモード]

リアルタイムトレースの条件を設定するダイアログボックスです。

トレーストリガとするイベントを指定します。

トレーストリガとする外部入力を指定します。

トレースの開始条件を選択します。

イベント2のトレースの取り込み条件を指定します。

強制ディレイモードを指定するときチェックします。

イベント設定ダイアログボックスを表示します。

トレースの取り込みモードを指定します。

イベント1のトレースの取り込み条件を指定します。

命令実行のトレースの取り込みを選択します。

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。

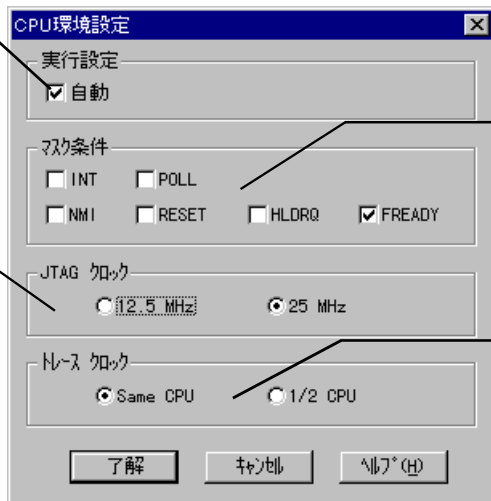
5.1.23 CPU環境設定ダイアログボックス

[実行]-[CPU環境設定]

CPUの動作環境を設定します。

PARTNERの内部処理のために、CPUをブレイクさせた場合、再実行を自動で行うか、行わないかの設定をします。

JTAGコントローラーのクロックを指定します



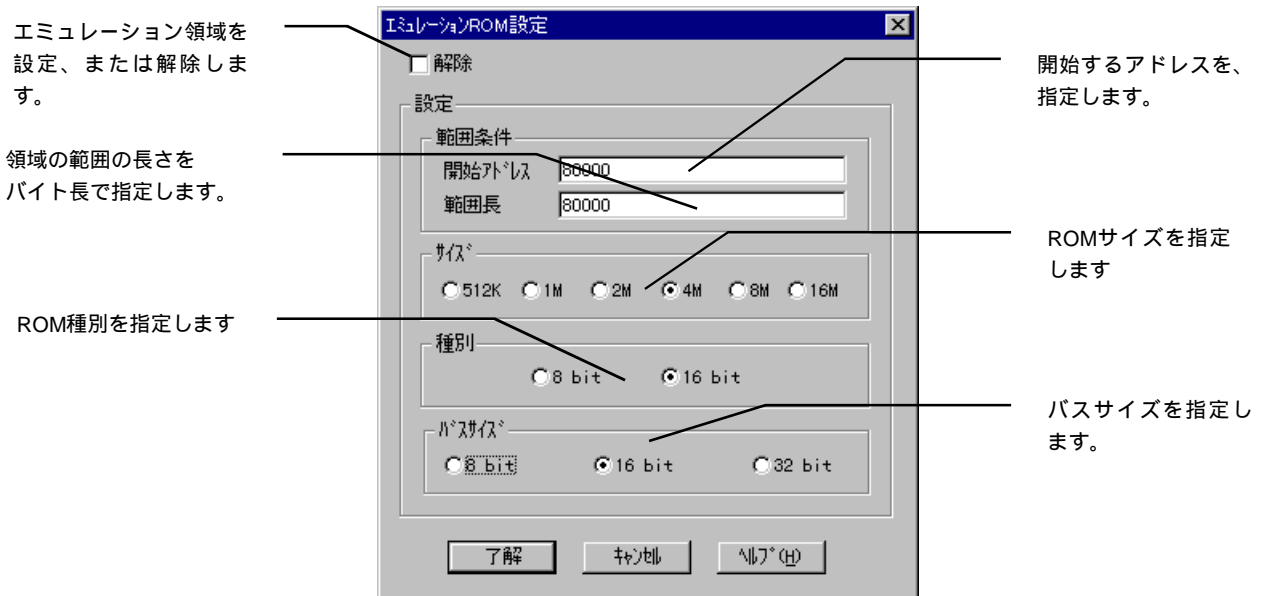
CPU信号のマスク条件を指定します。

トレースクロックの周波数を指定します。

5.1.24 エミュレーションROM設定ダイアログボックス

[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。



空白ページ

6 データ表現

PARTNER-V30MZ-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。

6.1 データ表現

PARTNER-V30MZ-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-V30MZ-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-V30MZ-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に設計してあります。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。(『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 34 [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-V30MZ-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-V30MZ-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 */
```

6.1.8 文字列(ストリング)

PARTNER-V30MZ-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-V30MZ-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

```
_AX,_BX,_CX,_DX,_SP,_BP,_SI,_DI,
_CS,_DS,_SS,_ES,_IP,_FLAGS
```

```
>while{ _AX!=_CX      /* AXレジスタとCX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AXとCXのレジスタ値が同じになるまでT(トレース実行)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、レジスタのほうが優先されます。

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-V30MZ-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)の条件判断の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
>
```

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。そこで、PARTNER-V30MZ-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)を持ちます。

レジスタ擬似変数	レジスタ
_AX	AX レジスタ
_BX	BX レジスタ
_CX	CX レジスタ
_DX	DX レジスタ
_SP	SP レジスタ
_BP	BP レジスタ
_SI	SI レジスタ
_DI	DI レジスタ
_CS	CS レジスタ
_DS	DS レジスタ
_SS	SS レジスタ
_ES	ES レジスタ
_IP	プログラムカウンタ
_FLAGS	フラグレジスタ

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、いくら参照したくとも見ることはできません。

PARTNER-V30MZ-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-V30MZ-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。

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、デバッグ中のほとんどの場合、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-V30MZ-TPでは、ウォッチ,インスペクト,?コマンドで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、データ参照のみの場合は意識的に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。

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-V30MZ-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-V30MZ-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-V30MZ-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-V30MZ-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**] [[!]**HLDREQ**
 [[!]**INT**] [[!]**FREADY**] [**JTAG**{25|12}] [**CLK**|**CLK2**]
 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]] 非メモリキャッシュの領域を設定します。

NCD リスト 非メモリキャッシュの領域を解除します。

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][, /C][, /W]

G@[, /C][, /W] デバッグプログラムを実行します。(F5, F7)
 /C:リアルタイムカウント継続, /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)

7.2.7 ブレークポイント

BP [番地[,回数[,コメント]]]	ブレークポイントを指定番地に設定します。(F9)
BC リスト	リストで指定されたブレークポイントを解除します。
BD リスト	リストで指定されたブレークポイントを無効にします。
BE リスト	リストで指定されたブレークポイントを有効にします。
BH [イベント番号][,イベント番号]	イベント番号で指定されたイベントをイベントブレークとして使用します。
BHD [イベント番号][,イベント番号]	イベント番号で指定されたイベントをイベントブレークとして使用しません。

7.2.8 レジスタ表示/変更

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。

7.2.9 イベントの設定

EV{1 2} [アドレス] [, [ALL INTACK IOREAD IOWRITE IOACCS MEMREAD MEMWRITE MEMACCS CODEFETCH READ WRITE HALT EXEC][, /{B W}][, /{EL EH}][, データ値]]	イベントブレーク、リアルタイムトレースで使用するイベント条件を設定します。
EV	現在設定されているイベント条件を表示します。

7.2.10 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(コンパ)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。

7.2.11 式の表示

H 式	式の値を 8, 10, 16 進数, ASCII および実数で表示します。
H 式 1, 式 2	式 1 と式 2 の和と差の値を表示します。
PRINTF 書式[,パラメータ]	
PF 書式[,パラメータ]	C の関数の printf() と同様なフォーマット表示をします。

7.2.12 シンボル表示/設定

X[シンボル名]	シンボル名(指定がなければすべてのシンボル)を表示をします。
[.]名前=アドレス	名前のシンボルを指定アドレスで登録(変更)します。

7.2.13 Cのデータ参照/変更

INS C の式[,関数]	C の式を評価してインサートポイントに表示します。 (F6, Ctrl+I)
W? C の式	C の式をウォッチポイントに登録します。(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ォッチポイントに登録します。
Y リスト	リストで指定されたウォ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。

7.2.14 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° のため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。

7.2.15 コード表示

V[.][ファイル名:][行]	指定ファイルの指定行をコードウィンドウに表示します。
V 関数名	指定関数のソースファイルをコードウィンドウに表示します。
U [番地]	コードウィンドウに指定番地から逆アングル表示を行います。
UPUSH [番地]	現在の表示アドレスをアドレススタック(8段の内部スタック)に PUSH して指定番地から逆アングル表示します。
UPOP	最後に UPUSH されたアドレスから逆アングル表示してアドレススタックを POP します。
UEND	最後に UPUSH されたアドレスから逆アングル表示します。

7.2.16 アセンブル

A 番地	指定番地よりアングルしてメモリに直接展開します。
------	--------------------------

7.2.17 バックトレース

K	C の関数のバックトレース表示を行います。
---	-----------------------

7.2.18 リアルタイムトレース

TRC	現在のリアルタイムトレースのモードを表示します。
TRC [デレイ値] [[!]DELAY] [[!]STARTEV1] [[!]EV1TRG] [[!]EV2TRG] [NOEXT NEGA POS1] [DT1OFF DT1EQ DT1TY] [DT2OFF DT2EQ DT2TY] [PCOFF PCJMP PCALL] [REAL NOREAL SAMPLE]	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
TD[U]	リアルタイムトレースの内容(ヒストリ)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。

7.2.19 システムコール

SYSC アドレス	アドレスで指定されたアドレスをエントリーとしてシステムコールを ON します。
SYSC OFF	システムコール機能を OFF します。
SYSC	システムコール機能の状態表示

7.2.20 システム制御

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。
!!	コマンドラインストリの表示をします。
! 文字列	コマンドラインストリの文字列サーチをします。
&	スルコマンドへの移行/終了
& スルコマンド	スルコマンドの実行

7.2.21 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。
------	-----------------------------

7.2.22 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ディック (APPEND) 時出力します。
>	ロギングの中断(ログファイルの閉鎖)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。

7.2.23 I/Oポート入力/出力

PI [型] 番地[, /C]	型指定された形式で指定アドレスの内容を表示します。
PO [型] 番地, データ[, /C]	型指定された形式で指定アドレスにデータを出力します。

7.2.24 オプション設定

OPTION {ON OFF}	大小文字判別をします。(SHIFT+F10)
-----------------	------------------------

7.2.25 画面コントロール/その他

CLS	コマンドウインドウのクリアをします。
HOME	コマンドウインドウのカーソルをホームポジションへ移動します。
LOCATE X 座標,Y 座標	コマンドウインドウのカーソルを指定した位置へ移動します。
LALL	マクロでの表示出力指定をします。
SALL	マクロでの表示出力抑制指定をします。
LIST	コマンドウインドウの表示出力指定をします。
NLIST	コマンドウインドウの表示出力抑制指定をします。
BEL	ベルを鳴らします。
TIME	現在時刻(時:分:秒)を表示します。
WAIT	一時停止します。
PROMPT 文字	プロンプトを指定された文字に変更します。
*	コメント行指定をします。

7.2.26 マクロコマンド

{ マクロ名	マクロ名でマクロ本体を登録します。
DO{ }WHILE 式	C の do..while 文と同様なマクロコマンド
FOR{ }	C の for 文と同様なマクロコマンド
WHILE{ 式	C の while 文と同様なマクロコマンド
REPEAT{ パラメータ	ループマクロコマンド
BREAK	マクロから抜け出します。
KILL マクロ名	マクロの削除をします。
MLIST [マクロ名]	マクロの表示をします。
MLIST >ファイル名	現在登録されているすべてのマクロを指定ファイルに書き込みます。
<ファイル名	指定されたマクロファイルからマクロを読み込みます。
IF{ 式	C の if,elseif,else 文と同様な条件制御コマンドです。

8 高級言語レベルデバッグとROM化

PARTNER-V30MZ-TPは各処理系が出力するデバッグ情報を含んだファイルをターゲットシステムにロードし、そのロードしたプログラムをソースレベルでデバッグする機能を持っています。これらの機能はデバッグ情報やオブジェクトが正しく生成されていないと正常なデバッグができません。

PARTNERで対応している処理系は以下の通りです。

- MS-C Ver4.0
- MS-C Ver5.1/6.0/7.0
- Turbo-C Ver1.5
- Turbo-C Ver2.0,Turbo-C++ Ver1.0 Borland C++ Ver3.1
- LSI-C86 Ver3.0
- LSI-C86 Ver3.3

通常、86用コンパイラで作成したプログラムはMS-DOS上で動作することを目的としており、そのままではPARTNERでロードできませんし、ROM化してターゲットシステム上で動作させることもできません。プログラムをROM化してターゲットシステム上で動作させるためには、ROM化サポートのスタートアップルーチンが必要です。また、MS-DOSやパソコンのBIOSおよびハードウェアに依存するライブラリ関数は使用できません。スタートアップルーチンでは、ハードウェアの初期化、スタックの初期化、初期化されたデータのRAMへの転送、割り込みベクタの初期化、main()関数へのジャンプなどの処理が行われます。

PARTNERには、各Cコンパイラに対応するスタートアップルーチンがソースで付属しています。まずターゲットシステムに合わせてスタートアップルーチンをカスタマイズしてください。各コンパイラ用スタートアップルーチンはサンプルディレクトリにあります。LSI-C86はコンパイラにスタートアップルーチンが含まれているため、このパッケージには含まれていません。

また、スタートアップルーチンのファイル名は"ROM?.ASM"で、?の部分がメモリモデルに対応した1文字です。

8.1 MS-C Ver4.0/5.1/6.0/7.0

PARTNER-V30MZ-TPはCodeView(MSC専用のデバッガ)用のデバッグ情報が出力された実行ファイルを読み込みます。実行ファイルの拡張子は".EXE"です。

8.1.1 コンパイルとリンク方法

コンパイル時に、/Ziオプションと/Odオプションを指定します。このオプションによって、ソース行情報がオブジェクトファイル(OBJファイル)に出力されるとともにデバッグの妨げとなる最適化を禁止します。

```
C>MSC /Zi /Od ソースファイル名
```

リンクは、MS-C付属のLINK.EXEを使用します。リンク時には、/COオプションを指定します。このオプションによって、CodeView(MSC専用のデバッガ)用のデバッグ情報が実行(EXE)ファイルの最後に付加されるかたちで出力されます。PARTNERはこのデバッグ情報をデバッグプログラムの読み込み時に一緒に読み込みます。また、あらかじめ作成したスタートアップルーチンを必ず最初にリンクしなければいけません。

```
C>LINK /CO スタートアップルーチン+ファイル名,ファイル名,ファイル名:
```

スタートアップルーチンは、使用するメモリモデルに合致したものをカスタマイズしてあらかじめアセンブルしておいてください。

この結果はCodeView(MSC専用のデバッガ)用のデバッグ情報が出力された実行ファイルが作成されます。

8.1.2 使用例

MS-C Ver4.0でのコンパイル,リンク例(スモールモデルの場合)

```
C>masm /mx roms; ..... スタートアップルーチンのアセンブル
Microsoft (R) Macro Assembler Version 4.00
Copyright (C) Microsoft Corp 1981, 1983, 1984, 1985. All rights reserved.
  48504 Bytes symbol space free
  0 Warning Errors
  0 Severe Errors
C>msc sample /AS /Zi /FPa; ..... サンプルプログラムのコンパイル
Microsoft (R) C Compiler Version 4.00
Copyright (C) Microsoft Corp 1984, 1985, 1986. All rights reserved.
C>link /CO roms sample,s_sample,s_sampl; ... ROM化のためのリンク
Microsoft (R) Overlay Linker Version 3.55
Copyright (C) Microsoft Corp 1984, 1985, 1986. All rights reserved.
C>
```

8.2 Turbo C Ver1.5

PARTNER-V30MZ-TPはパブリックシンボル情報とソース行情報が出力されたシンボルファイルを読み込みます。シンボルファイルの拡張子は".MAP"です。

8.2.1 コンパイルとリンク方法

CTCC(コマンドラインコンパイラ)の場合にはコンパイル時に `-y -r-` オプションの指定をします。

TC(統合化会話型コンパイラ)の場合には `Option-Compiler-Code Generation-Line numbers` を ON ,`Option-Compiler-Optimization-Use register variables` を Off ,`Linker-Map file` を Public に指定してください。このオプションによって、ソース行情報とシンボル情報がMAPファイルに出力されるとともにデバッグの妨げになるレジスタ変数割当てを禁止します。ユーザがカスタマイズしたスタートアップルーチンのファイル名をTurbo-C用に変更して所定のディレクトリに入れて置ってください。(例えばスモールモデルでは"ROMS.OBJ" "COS.OBJ"に変更してください。ただし、TCCを使用してコンパイルのみ行いリンクは、TLINKコマンドを使用して別々に行なう場合はスタートアップルーチンのファイル名を変更しなくても可能です。)

```
C>TCC -y -r -c ソースファイル名
```

Turbo-C付属のTLINK.EXEを使用する場合には、`/m/l/c`オプションを指定します。このオプションによって、パブリックシンボル情報とソース行情報がシンボルファイル(MAPファイル)に出力されます。また、あらかじめ作成したスタートアップルーチンを必ず最初にリンクしなければいけません。

```
C> tlink /m/l/c スタートアップルーチン+ファイル名,ファイル名,ファイル名;
```

この結果はパブリックシンボル情報とソース行情報が出力されたシンボルファイルが作成されます。

8.2.2 使用例

Turbo-C Ver1.5でのコンパイル,リンク例(スモールモデルの場合)

```
C>tasm roms
```

```
C>tcc -v -r -c sample.c
```

```
C>tlink /m/l/c roms sample,sample,sample,cs
```

```
C>
```


8.3 Turbo C Ver 2.0 Turbo C++ Ver1.0 Borland C++ Ver3.1

PARTNER-V30MZ-TPはTurbo Debugger(Turbo-C専用デバッガ)用のデバッグ情報が出力された実行ファイルを読み込みます。実行ファイルの拡張子は".EXE"です。

8.3.1 コンパイルとリンク方法

TCC(コマンドラインコンパイラ)の場合にはコンパイル時に `-v` オプションの指定をします。TC(統合化会話型コンパイラ)の場合には `Option-Compiler-Code Generation-Obj debug information` を `On` に指定してください。このオプションによってデバッグ情報がオブジェクトファイル(OBJファイル)に出力されます。

```
C>TCC -v -c ファイル名
```

リンクは、Turbo-C付属のTLINK.EXEを使用します。

リンク時には、`/v/c`オプションを指定します。このオプションによってTurbo Debugger(Turbo-C専用デバッガ)用のデバッグ情報が実行(EXE)ファイルの最後に付加されるかたちで出力されます。PARTNERはこのデバッグ情報をデバッグプログラムの読み込み時に一緒に読み込みます。

```
C> tlink /v/c スタートアップル-チン+ファイル名,ファイル名,ファイル名:
```

この結果はTurbo Debugger(Turbo-C専用デバッガ)用のデバッグ情報が出力された実行ファイルが作成されます。

8.3.2 使用例

Turbo-C Ver2.0でのコンパイル,リンク例(スモールモデルの場合)

```
C>tasm roms
```

```
Turbo Assembler Version 1.0 Copyright (c) 1988 by Borland International
```

```
Assembling file: ROMS.ASM
```

```
*Warning* ROMS.ASM(36) Reserved word used as symbol: CONST
```

```
*Warning* ROMS.ASM(42) Reserved word used as symbol: STACK
```

```
Error messages: None
```

```
Warning messages: 2
```

```
Remaining memory: 383k
```

```
C>tcc -v -c sample.c
```

```
Turbo C Version 2.0 Copyright (c) 1987, 1988 Borland International
```

```
sample.c:
```

```
    Available memory 331368
```

```
C>tlink /c/v roms sample,sample,sample,cs
```

```
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International
```

```
C>
```

8.4 LSI-C86

PARTNER-V30MZ-TPはパブリックシンボル情報とソース行情報が出力されたシンボルファイルを読み込みます。シンボルファイルの拡張子は".MAP"です。

8.4.1 コンパイルとリンク方法

コンパイル時に `-g` オプションを指定します。このオプションによって、ソース行情報がオブジェクトファイル(OBJファイル)に出力されます。

```
C>LCC -g -c ファイル名
```

リンクはMS-DOS標準のLINK.EXEがLCCによって起動されます。LINKのバージョンは、3.0以降のものがが必要です。リンク時には、`-g`オプションと`-a`オプションを指定します。このオプションによって、パブリックシンボル情報とソース行情報がシンボルファイル(MAPファイル)に出力されます。

```
C> LCC -g -a -o EXE 文件名 crom 文件名 文件名 文件名:
```

LSI-C86の場合は他のコンパイラと違いEXEファイルを直接読み込みません。HEXファイルとロケートされたMAPファイルを読み込みます。

LSI-C86 Ver3.0では、このHEXとMAPファイルはLSI-C86に付属しているROMIZEユーティリティを使用して作成します。

```
C>ROMIZE -m 出力MAP 文件名 EXE 文件名 HEX 文件名 ROM スタートアドレス RAM スタートアドレス
```

ROMIZEユーティリティはEXEファイルと同じ名前(ファイル名の.からは省く)のMAPファイルからアブソリュートな出力MAPファイルを作成します。このためEXEファイルと同名のMAPファイル名を指定できません。従ってEXEファイル名とHEXファイル名は違う名前にし、出力MAPファイル名はHEXファイル名と同じ名前にしてください。

8.4.2 使用例

```
C>lcc -v -g -c -o crom.obj crom.p
cpp -o 1.$$$ crom.p86
r86 -o crom.obj 1.$$$
C>lcc -v -g -c -o sample.obj sample.c
cpp -o 1.$$$ sample.c
cf -chcyt 1.$$$ 2.$$$ 3.$$$
cg86 -g 2.$$$ 3.$$$
r86 -o sample.obj 3.$$$
C>lcc -v -g -c -o pt_sub.obj ept_sub.p86    仮想コンソールのアセンブル
cpp -o 1.$$$ ept_sub.p86
r86 -o ept_sub.obj 1.$$$
C>lcc -g -v -a -o sample.exe crom.obj sample.obj ept_sub.obj
link @link.i
Microsoft (R) 8086 Object Linker Version 3.05
Copyright (C) Microsoft Corp 1983, 1984, 1985. All rights reserved.
Object Modules [.OBJ]: /LI/MAP CROM+
Object Modules [.OBJ]: SAMPLE+
Object Modules [.OBJ]: PT_SUB
Run File [CROM.EXE]: SAMPLE.EXE
List File [E:SAMPLE.MAP]: SAMPLE
Libraries [.LIB]: S%DOSLIB;
A>romize -m sample_a.map sample sample_a.hex f000 1000
ROMアドレスF000:0番地、RAMアドレス1000:0にロケート
PARTNERはsample_a.hex,sample_a.mapの両ファイルをLコマンドで読み込みます。
```

付録1

コマンドラインによる起動オプションの設定

PARTNER-V30MZ-TPの起動オプションは通常、RPTSETUP(V30MZ)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スタートメニューに登録したショートカットのコマンドラインに付加されます。RPTSETUP(V30MZ)を使用せずにショートカットの[プロパティ]で直接編集することも可能です。

PARTNER-V30MZ-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。デバッグ情報量粋にはグローバル、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-V30MZ-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(RPTV30T.CFG)、Windows用初期設定ファイル(RPTV30T.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-V30MZ-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。

-Mオプション

デバッグ対象となるプログラムのメモリモデルを指定します。

このオプションはCのデバッグモード(『-Xオプション(131頁参照)』)時に有効となります。

-MS(データ: 最大64Kバイト、コード: 最大64Kバイト)

スモールモデルの指定。(デフォルト)

-MM(データ: 最大64Kバイト、コード: 最大1Mバイト)

ミディアムモデルの指定。

-MC(データ: 最大1Mバイト、コード: 最大64Kバイト)

コンパクトモデルの指定。

-ML(データ: 最大1Mバイト、コード: 最大1Mバイト)

ラージモデルの指定。

-MH(データ: 最大1Mバイト、コード: 最大1Mバイト)

ヒュージモデルの指定。

-SDオプション

コードウィ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できます。

-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

コードウィ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-V30MZ-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-V30MZ-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。MSAM、TASMなどのアセンブラのデバッグはこのオプションを付けてください。

-XM

MS-C Ver 4.0/5.1/6.0/7.0のデバッグモード指定です。デバッグ情報はEXEファイルからCode Viewのデバッグ情報を読み込みます。

-XT

Turbo-C Ver1.5のデバッグモード指定です。デバッグ情報はMAPファイルから読み込みます。

-XT2

Turbo-C Ver2.0、Turbo-C++ Ver1.0及びBorland-C++ Ver3.1 のデバッグモード指定です。デバッグ情報はEXEファイルからTurbo Debuggerのデバッグ情報を読み込みます。

-XLSI

LSI-C86 Ver3.0/3.3のデバッグモード指定です。デバッグ情報はMAPファイルから読み込みます。

LSI-C86は、他の処理系にない高度な変数のレジスタ割り当てや、関数の引数レジスタ渡しなど非常に優れた機能を持っています。しかし、この高度な最適化のためにデバッガではローカル変数の参照やスタックのバックトレースなどの機能が使用できません。

-XOMF

INTERL OMF86のデバッグモード指定です。デバッグ情報はABSファイルから読

み込みます。

【注意】-XMSC、-XBCCに対応しているINTEL OMF86形式のファイル以外はこのオプションで読み込みます。このオプションを指定した時はシンボリックデバッグまたはソースラインデバッグのみとなります。

-XMSC

MS-C Ver6.0/7.0でコンパイルし、Link&Locate(Ver7.2f)でロケートしたINTEL OMF86のデバッグモード指定です。デバッグ情報はOMFファイルから読み込みます。

-XBCC

Borland-C Ver3.0/3.1でコンパイルし、Paradigm Locate(Ver3.2I)でロケートしたINTEL OMF86のデバッグモード指定です。デバッグ情報はOMF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:¥RPTV30T¥BIN¥RPTV30T -DC:¥SAMPLE @TEST.INF
```

ファイル内容(TEST.INF)

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


付録2 エラーメッセージ

PARTNER-V30MZ-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(V30MZ)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(V30MZ)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、多くの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、多くの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(V30MZ)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(V30MZ)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。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。なお、アドレス等を変更する前に、RESETコマンドを実行してCPUをリセットしてください。

ENVコマンドまたは、CPU環境設定メニューによって強制レディモードを設定することで対応することもできます。ただし、このモードの時は、レディ信号がアクティブにならない状態が発生した場合、ウィンドウの表示が非常に遅くなる場合があります。

索引

.

.DAT	
終了時に作成するファイル.....	24

B

Borland C++ Ver3.1	125
コンパイルとリンク方法	125
使用例	126

C

CPUリセット	115
Cの式.....	107
Cのデータ参照/変更.....	117
Cの変数	108
Cの変数スコープ	108

I

I/Oポート入力/出力	119
INIT.MCR	
起動時に必要なファイル	22, 23

L

LSI-C86.....	127
使用例	128

M

MEMO.PT	
終了時に作成するファイル.....	24
MS-C Ver4.0/5.1/6.0/7.0.....	122
コンパイルとリンク方法	122
使用例	123

P

PARTNERウインドウ	
画面構成.....	30

T

Turbo C Ver 2.0	125
コンパイルとリンク方法	125
使用例	126
Turbo C Ver1.5	124
コンパイルとリンク方法	124
使用例	124
Turbo C++ Ver1.0	125
コンパイルとリンク方法	125
使用例	126

あ

アセンブル.....	118
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.....	32
ウインドウバー.....	80
ウインドウメニュー	42
ウォッチウインドウ.....	66
ショートカットキー	66
マウス操作	67
ローカルメニュー.....	39, 67
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.....	119

か

画面構成.....	30
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
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
-M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
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
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.....	31
コード表示.....	118
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.....	117
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
インスペクトウインドウ.....	73
ウォッチウインドウ.....	66
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.....	61
バックトレースウインドウ.....	64
ヒストリウインドウ.....	71
ブレークウインドウ.....	68
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.....	59
ローカルウインドウ.....	62
シンボル表示/設定.....	117
スタックウインドウ.....	61
ショートカットキー.....	61
ローカルメニュー.....	38, 61
画面構成.....	31
ステータスバー.....	81
設定メニュー.....	43

セットアップ	
セットアップの前に	12
セットアップの方法	13
必要なソフトウェア	3
必要なハードウェア	2

た

ダイアログコマンド	83
C言語レベルでのデータ表現	107
データ表現	100
ダイアログボックス	84
アドレス指定(コード)	89
アドレス指定(メモリ)	90
イベントブレークの設定	93, 94
色指定	86
インスペクト設定	87
ウォッチ設定	87
各種状態の設定	93
行番号指定	88
コマンド履歴	89
シンボル拡張	90
ツールバーの設定	85
データ設定	91
ファイルを開く	84
フォントの指定	85
フラグの設定	92
ブレークポイントの設定	92
フレーム番号へジャンプ	93
モジュール	88
文字列検索	86
リアルタイムトレースモード設定	95, 96, 97
レジスタの設定	91
ツールバー	77
設定	77
定数	109

データ表現	99
特殊シンボル	102

は

ハードウェアの初期化	115
バクトレース	118
バクトレースウインドウ	64
ショートカットキー	64
マウス操作	65
ローカルメニュー	39, 65
画面構成	31
バッチ	119
ヒストリウインドウ	70
ショートカットキー	71
ローカルメニュー	41, 72
画面構成	32
ヒストリメニュー	
ヒストリウインドウ	41
表示メニュー	35
ファイルメニュー	33
ファイルリード/ライト	115
副作用のある式	111
ブレークウインドウ	68
ブレークウインドウ	
ショートカットキー	68
マウス操作	69
ローカルメニュー	40, 69
画面構成	31
ブレークポイント	116
プログラムマネージャへの登録	
新規に環境を作成	19
プログラムロード	115
プログラム実行	115
プロジェクトファイル	
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	78

ま

マウス操作	47
インスペクトウインドウ	74
ウォッチウインドウ	67
コードウインドウ	53
バックトレースウインドウ	65
左ボタンドラッグ	48
ブレークウインドウ	69
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	60
ローカルウインドウ	63
マクロコマンド	120
メニュー	33
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
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	117
文字列	104
文字列の定義	118

ら

リアルタイムトレース	118
レジスタウインドウ	59
ショートカットキー	59
マウス操作	60
ローカルメニュー	38, 60
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	62
ショートカットキー	62
マウス操作	63
ローカルメニュー	39, 63
画面構成	31
ローカルシンボル	101
ローカルメニュー	37
インスペクトウインドウ	40, 74
ウォッチウインドウ	39, 67
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, 61
バックトレースウインドウ	39, 65
ヒストリウインドウ	72
ブレークウインドウ	40, 69
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, 60
ローカルウインドウ	39, 63
ロギング	119

PARTNER-V30MZ-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.