

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。



お客様各位

---

## 資料中の「三菱電機」、「三菱XX」等名称の株式会社ルネサス テクノロジへの変更について

---

2003年4月1日を以って株式会社日立製作所及び三菱電機株式会社のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。

従いまして、本資料中には「三菱電機」、「三菱電機株式会社」、「三菱半導体」、「三菱XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

注:「高周波・光素子事業、パワーデバイス事業については三菱電機にて引き続き事業運営を行います。」

2003年4月1日  
株式会社ルネサス テクノロジ  
カスタマサポート部

# M32R/Dシリーズ用 エミュレータデバッガ

PD32000 V.1.00 ユーザーズマニュアル

---

Microsoft、MS、MS-DOS、Windows、Windows NTは、米国Microsoft Corporationの米国およびその他の国における登録商標です。  
IBM、AT、PS/2、OS/2は、米国International Business Machines Corporationの登録商標です。

《本資料ご利用に際しての留意事項》

本資料中に記載の技術情報は、三菱電機株式会社・三菱電機セミコンダクタシステム株式会社が所有する知的財産権その他の権利の実施、使用を許諾するものではありません。

本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、三菱電機株式会社・三菱電機セミコンダクタシステム株式会社は責任を負いません。本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他すべての情報は本資料発行時点のものであり、三菱電機株式会社・三菱電機セミコンダクタシステム株式会社は特性改良などにより予告なしに変更することがあります。したがって、三菱半導体製品のご購入にあたりましては事前に三菱電機株式会社・三菱電機セミコンダクタシステム株式会社または特約店へ最新の情報をご確認ください。

本資料に記載の製品データ、図、表に示す技術的な内容、プログラムおよびアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。三菱電機株式会社・三菱電機セミコンダクタシステム株式会社は、適用可否に対する責任は負いかねます。

本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途をご検討の際には、三菱電機株式会社・三菱電機セミコンダクタシステム株式会社または特約店へご照会ください。

本資料の転載、複製については、文書による三菱電機株式会社・三菱電機セミコンダクタシステム株式会社の事前の承諾が必要です。

本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたら三菱電機株式会社・三菱電機セミコンダクタシステム株式会社または特約店へご照会ください。

製品の内容及び本書についてのご質問は、インストーラが生成する以下のテキストファイルをご使用の上、電子メールもしくはFAXにて最寄りの開発ツールサポート窓口までお問い合わせください。

Windows 95/Windows NT 4.0版 ¥SUPPORT¥製品名¥SUPPORT.TXT

EWS版 :/support/製品名/toolinfo.txt

また、マニュアル・ガイドブックなどの最後に添付されている「技術サポート連絡書」もご利用ください。

三菱電機セミコンダクタシステム株式会社  
マイコンソフトツール部 開発ツールサポート窓口

[大阪地区]

〒564-0051 大阪府吹田市豊津町1番18号 / エクラート江坂ビル

FAX番号 : 06-6338-4471

[東京地区]

〒105-6130 東京都港区浜松町2丁目4番1号 / 世界貿易センタービル30F

FAX番号 : 03-5403-0414

[中部地区]

〒460-0003 愛知県名古屋市中区錦2丁目15番15号 / 豊島ビル7F

FAX番号 : 052-221-7318

[九州地区]

〒812-0016 福岡市博多区博多駅南1-3-6 / 第3博多偕成ビル

FAX番号 : 092-452-1427

[電子メールアドレス]

E-mail :support@tool.mesc.co.jp

[ホームページアドレス]

http://www.tool-spt.mesc.co.jp

---

# はじめに

PD32000は、三菱 32 ビットマイクロコンピュータM32R シリーズ用エミュレータ **M32000T**をコントロールするデバッガソフトウェア [ Windows 対応版 ] です。本ユーザーマニュアルは、PD32000の特長、機能、セットアップ方法、操作方法等について説明しています。

## プログラムの使用権

本製品に含まれるプログラムの使用権は、「ソフトウェア使用権許諾契約書」に基づきます。PD32000のプログラムは、お客様の製品開発の目的でのみ使用できます。その他の目的では使用できませんのでご注意ください。

また、本マニュアルによってソフトウェアの使用権の実施に対する保証及び使用権の実施の許諾を行うものではありません。

## 用語の定義

**PD32000**のマニュアルでは、エミュレータの名称に関して、以下のように用語を定義しています。

表記	意味
M32000T	エミュレータM32000TB-2MB-Eを意味します。



# 目次

## 概要編 1

<b>1. PD32000 の概要</b>	<b>3</b>
<b>2. PD32000 の特長</b>	<b>4</b>
2.1 マルチウィンドウ機能.....	4
2.2 ブレーク機能.....	4
2.3 トレース機能.....	4
2.4 ソースレベルデバッグ機能.....	5
2.5 オンデマンド方式.....	5
<b>3. PD32000 の入出力ファイル</b>	<b>6</b>
3.1 入力ファイル.....	6
3.2 出力ファイル.....	8
3.3 テンポラリファイル.....	9

## セットアップ編 11

<b>1. セットアップ</b>	<b>13</b>
1.1 インストール.....	13
1.2 PD32000 の起動.....	13
1.3 PD32000 の動作環境の設定.....	14

## ウィンドウ機能編 19

<b>1 PD32000 のウィンドウ機能</b>	<b>21</b>
1.1 PD32000 ウィンドウ.....	21
1.2 プログラムウィンドウ.....	26
1.3 ソースウィンドウ.....	30
1.4 レジスタウィンドウ.....	33
1.5 メモリウィンドウ.....	34
1.6 ダンプウィンドウ.....	36
1.7 ASM ウォッチウィンドウ.....	38

1.8	C ウォッチウィンドウ .....	40
1.9	ローカルウィンドウ .....	42
1.10	ファイルローカルウィンドウ .....	44
1.11	グローバルウィンドウ .....	45
1.12	スクリプトウィンドウ .....	46
1.13	ブレークポイント設定ダイアログ .....	48
1.14	エミュレーションメモリダイアログ .....	50
1.15	トレースコンディションウィンドウ .....	51
1.16	トレースウィンドウ .....	56
1.17	区間時間測定ウィンドウ .....	63
1.18	コールスタックウィンドウ .....	65
1.19	MR ウィンドウ .....	66

## 基本操作方法編

69

<b>1</b>	<b>ターゲットプログラムの読み込み・表示</b>	<b>71</b>
1.1	ダウンロードするには.....	71
1.2	ダウンロード直後のプログラム表示位置を変更するには.....	72
1.3	アップロードするには.....	73
1.4	逆アセンブル結果を保存するには.....	73
1.5	プログラムの任意位置を常に表示するには .....	74
1.6	プログラムの表示位置を変更するには .....	75
1.7	他ディレクトリに存在するソースプログラムを参照するには .....	77
1.8	ソースと逆アセンブル結果を MIX 表示するには .....	78
1.9	逆アセンブル結果を表示するには.....	78
<b>2</b>	<b>ターゲットプログラムの実行 / 停止</b>	<b>80</b>
2.1	実行・停止するには .....	80
2.2	ステップ実行するには.....	82
2.3	現ルーチンから上位ルーチンへ戻るには.....	83
2.4	指定位置までプログラムを実行するには.....	83
2.5	プログラムをリセットするには .....	84
<b>3</b>	<b>ソフトウェアブレーク</b>	<b>85</b>
3.1	ブレークポイント設定ダイアログをオープンするには .....	85
3.2	ブレークポイントを設定するには.....	86
3.3	ブレークポイントを解除するには.....	87
3.4	ブレークポイントを一時的に無効化するには.....	87

3.5	ブレークポイントを一時的に有効化するには.....	88
3.6	ウィンドウ上からブレークポイントを設定するには.....	88
3.7	ツールバーからブレークポイントを設定するには.....	89
3.8	ブレークポイントを保存するには.....	89
3.9	ブレークポイントを読み込むには.....	89
<b>4</b>	<b>トレース</b>	<b>90</b>
4.1	バス情報を参照するには.....	90
4.2	実行した命令で実行経路を参照するには.....	91
4.3	ソース行レベルで実行経路を参照するには.....	91
4.4	トレース条件を設定するには.....	94
<b>5</b>	<b>レジスタ情報、メモリ内容の参照・設定</b>	<b>99</b>
5.1	レジスタの内容を参照するには.....	99
5.2	レジスタの内容を変更するには.....	99
5.3	任意アドレスの値を参照するには.....	100
5.4	指定アドレスにデータを設定するには.....	103
5.5	メモリ内容の表示を更新するには.....	104
<b>6</b>	<b>C 変数の参照・変更</b>	<b>105</b>
6.1	C 変数の値を参照するには.....	105
6.2	C 変数の値を変更するには.....	106
<b>7</b>	<b>スクリプトコマンド</b>	<b>107</b>
7.1	スクリプトコマンドを実行するには.....	107
7.2	スクリプトコマンドの実行結果を記録するには.....	108
7.3	スクリプトコマンドを一括して実行するには.....	110
<b>8</b>	<b>PD32000 の終了</b>	<b>113</b>
8.1	PD32000 を終了するには.....	113
<b>9</b>	<b>その他</b>	<b>114</b>
9.1	ラインアセンブルするには.....	114
9.2	Make を起動するには.....	116
9.3	ターゲットプログラムの文字列を検索するには.....	117
9.4	ウィンドウの表示領域の割合を変更するには.....	117
9.5	アクティブウィンドウを切替えるには.....	118
9.6	PD32000 のバージョンを表示するには.....	119

<b>1</b>	<b>区間時間測定</b>	<b>123</b>
1.1	概要.....	123
1.2	測定条件の設定.....	123
1.3	測定条件の削除.....	124
1.4	測定条件の変更.....	124
1.5	区間時間測定結果を参照するには.....	125
<b>2</b>	<b>カスタマイズ機能</b>	<b>126</b>
2.1	カスタマイズ機能とは.....	126

## リアルタイム OS デバッグ編 **131**

<b>1</b>	<b>リアルタイム OS 用のデバッグ機能</b>	<b>133</b>
1.1	リアルタイム OS の情報を参照するには.....	133

## リファレンス編 **149**

<b>1</b>	<b>スクリプトコマンド一覧</b>	<b>151</b>
1.1	入力書式.....	151
1.2	コマンド一覧.....	152
<b>2</b>	<b>スクリプトファイルの記述方法</b>	<b>157</b>
2.1	スクリプトファイルの構成要素.....	157
2.2	式の記述方法.....	160
<b>3</b>	<b>C 言語式について</b>	<b>164</b>
3.1	C 言語式の記述方法.....	164
3.2	C 言語式の表示形式.....	168
<b>4</b>	<b>エラーメッセージ一覧</b>	<b>171</b>

## 索引 **183**

# 概要編



## 1.PD32000の概要

PD32000は、三菱 32 ビットマイクロコンピュータM32R シリーズ用エミュレータ**M32000T** をコントロールするエミュレータデバッガソフト [ Windows 対応版 ] です。

## 2. PD32000の特長

### 2.1 マルチウィンドウ機能

PD32000は、オーバーラッピング形式のマルチウィンドウ機能をサポートし、様々な情報を同時に表示します。各ウィンドウは、メニュー、ボタン等を備えており、これらをマウスで操作することによってコマンドを実行できます。

### 2.2 ブレーク機能

PD32000は、以下のブレーク機能を用意しています。

#### 2.2.1 ソフトウェアブレーク

ソフトウェアブレークとは、指定アドレスの命令を実行する手前でブレークする機能のことです。このブレークが行われるポイントを、ソフトウェアブレークポイントといいます。ソフトウェアブレークポイントの設定は、ブレークポイント設定ダイアログで行います。また、プログラムウィンドウやソースウィンドウからも簡単にソフトウェアブレークポイントが設定できます。PD32000では、48点のソフトウェアブレークポイントが設定できます。

ブレークポイント設定ダイアログから設定したソフトウェアブレークポイントの保存や読み込みが可能です。

### 2.3 トレース機能

PD32000は、以下のトレース機能を用意しています。

#### 2.3.1 PC トレース機能

PC トレース機能とは、ターゲットプログラムの実行履歴を記録する機能です。**PD32000**では、128K 命令分の実行履歴を記録できます。PC トレースの設定は、トレースコンディションウィンドウで行うことができます。なお PC トレースは、リアルタイムトレースではありません。

#### 2.3.2 SIGNAL トレース機能

SIGNAL トレース機能とは、M32R/D のバスに出力されたデータの履歴を記録する機能です。**PD32000**では、256K サイクル分の履歴が記録できます。このトレース範囲を指定するための基点をトレースポイントといいます。トレースポイントの設定は、トレースコンディションウィンドウで行うことができます。なお SIGNAL トレースは、リアルタイムトレースです。

## 2.4 ソースレベルデバッグ機能

ソースファイルを表示し、ソース行でのブレークポイント指定やステップ実行など、ソースレベルのデバッグが行えます。PD32000では、C 言語及びアセンブリ言語レベルでのデバッグが可能です。

- ソースファイルは、プログラムウィンドウ及びソースウィンドウで参照できます。
- C 言語ソースファイルに記述した C 変数や C 言語式は、C ウォッチウィンドウ、ローカルウィンドウ、ファイルローカルウィンドウ、グローバルウィンドウで参照できます。
- アセンブラソースファイルに記述したラベルやシンボルは、ASM ウォッチウィンドウで参照できます。
- ブレークポイント等のアドレス指定には、ラベル名やシンボル名が指定できます。これらから値への変換は、ローカル、グローバルの順に行います。C 言語ソースファイルで定義した変数（関数）名を指定する場合は、ご使用の C コンパイラにより異なります。**CC32R** の場合は変数（関数）名の前にアンダーバー“\_”を追加します。なお、引数がレジスタ渡しとなる関数の場合は、“\$”を指定してください（レジスタ渡し関数の仕様については、**CC32R**ユーザーズマニュアルの関数呼び出し規則をご参照下さい）。**TW32R** および **DCC** の場合は C 言語ソースファイルで定義した変数（関数）名で参照できます。

## 2.5 オンデマンド方式

PD32000では、ターゲットプログラムをダウンロードした際に、テンポラリファイルを作成し、必要なデバッグ情報を必要になった時にメモリ上に読み込む「オンデマンド方式」をサポートしています。これにより、メモリの使用量を削減することができます。なおデフォルトは、メモリ上に全てのデバッグ情報を保持する「オンメモリ方式」です。

「オンデマンド方式」と「オンメモリ方式」のどちらを使用するかは、**PD32000**の動作環境の設定で選択することができます。**PD32000**の動作環境の設定については、本マニュアル セットアップ編の項目「1.3 **PD32000**の動作環境の設定」をご参照下さい。

テンポラリファイルは、pdb\_xxxx.tmp (xxxx は 16 進 4 桁の数値) というファイル名で生成します。このファイルは、ダウンロードの直前および**PD32000**終了時に削除されます。

テンポラリファイルを作成するディレクトリは、Init ダイアログで指定したディレクトリに作成します。指定したディレクトリにテンポラリファイルと同名のファイルがある場合は、テンポラリファイルを作成するディレクトリを変更、または「オンメモリ方式」を選択してください。

## 3. PD32000の入出力ファイル

### 3.1 入力ファイル

PD32000が取り扱う入力ファイルは、以下の通りです。

#### 3.1.1 アブソリュート形式ファイル

アブソリュート形式ファイルは、ソースファイルで使用している変数情報や行情報などのデバッグ情報及び機械語情報を格納しているファイルです。C コンパイラ**CC32R**、**TW32R**、**DCC**により生成されます。ファイル属性は、“.x”、“.exe”または“.abs”を指定してください。なお、ELF/DWARF 形式のファイルのみ任意のファイル属性が指定可能です。

#### 3.1.2 モトローラSフォーマットファイル

モトローラSフォーマットファイルは、機械語情報を格納しているファイルです。**CC32R**ではユーティリティツール**lmc32R**、**TW32R**ではユーティリティツール**m32r-elf-objcopy**、**DCC**ではユーティリティツール**ddump**により生成されます（アブソリュート形式ファイルを読み込んで変換します）。ファイル属性は、“.mot”を指定してください。

#### 3.1.3 バイナリファイル

バイナリファイルは、バイナリデータを格納しているファイルです。ファイル属性は、“.bin”です。PD32000のアップロード機能により生成します。

#### 3.1.4 MPU ファイル

ターゲットMPUの固有情報を持つ情報ファイルです（メモリマッピング情報）。MPUファイルは、製品に付属されています。ファイル名は、“M32000T.MPU”です。

#### 3.1.5 スクリプトファイル

スクリプトファイルは、スクリプトコマンドを自動実行するためのファイルです。スクリプトファイルは、スクリプトウィンドウから読み込みます。ファイル属性は、“.scr”です。

#### 3.1.6 ヘルプファイル

ヘルプファイルは、PD32000のヘルプメッセージを含んだファイルです。ファイル属性は、“.hlp”です。ヘルプファイルは、PD32000に添付しています。

#### 3.1.7 環境設定ファイル

環境設定ファイルは、PD32000の環境設定に関する情報を保持するファイルです。環境設定ファイルは、PD32000が自動的に生成します。ファイル名は、pd32000.iniです。このファイルは、Windows ディレクトリ（ご使用のWindowsがインストールされたディレクトリ）に保存され、ユーザ自身が作成/編集することはできません。

### 3.1.8 Cウォッチポイント情報格納ファイル

Cウォッチポイント情報格納ファイルは、Cウォッチウィンドウに登録されているCウォッチポイントの情報を保持するファイルです。Cウォッチポイント情報格納ファイルは、**PD32000**が自動的に生成します。ファイル属性は、“.cwp”です。このファイルは、Windowsディレクトリ（ご使用のWindowsがインストールされたディレクトリ）に保存され、ユーザ自身が作成/編集することはできません。

### 3.1.9 ソフトウェアブレイクポイントファイル

ソフトウェアブレイクポイントファイルは、ソフトウェアブレイクポイントの設定内容を保存しているファイルです。ファイル属性は、“.brk”です。

本ファイルを読み込みことにより、ソフトウェアブレイクポイントを設定することができます。本ファイルは、ソフトウェアブレイクポイント設定ダイアログから読み込みます。

### 3.1.10 トレースポイントファイル

トレースポイントファイルは、トレースポイントの設定内容を保存しているファイルです。ファイル属性は、“.tc”です。

本ファイルを読み込みことにより、トレースポイントを設定することができます。本ファイルは、トレースコンディションウィンドウから読み込みます。

### 3.1.11 エミュレーションメモリ状態ファイル

エミュレーションメモリ状態ファイルは、エミュレーションメモリの設定内容を保存しているファイルです。ファイル属性は、“.em”です。

本ファイルを読み込みことにより、エミュレーションメモリを設定することができます。本ファイルは、エミュレーションメモリ設定ダイアログから読み込みます。

## 3.2 出力ファイル

**PD32000**が取り扱う出力ファイルは、以下の通りです。

### 3.2.1 モトローラ S フォーマットファイル

モトローラ S フォーマットファイルは、機械語情報を格納しているファイルです。ファイル属性は、“`.mot`”です。

**PD32000**では、**モトローラ S フォーマットファイル**をエミュレータへダウンロード、およびエミュレータのメモリ内容を**モトローラ S フォーマットファイル**に保存（アップロード）することが可能です。

ただし、**PD32000**は、機械語のみではソースレベルのデバッグが行えません。ソースレベルのデバッグを行う場合は、**モトローラ S フォーマットファイル**と同時に**SYSROF アプソリュート形式ファイル**のデバッグ情報を読み込んでください。

### 3.2.2 バイナリファイル

バイナリファイルは、バイナリデータを格納しているファイルです。ファイル属性は、“`.bin`”です。**PD32000**のアップロード機能により保存できます。保存したバイナリファイルは、**PD32000**で再ダウンロードすることもできます。

### 3.2.3 逆アセンブルファイル

逆アセンブルファイルは、プログラムメモリの逆アセンブル結果を保存したファイルです。ファイル属性は、“`.txt`”です。逆アセンブルファイルは、参照用のテキストファイルです。逆アセンブルファイルの再アセンブル / 再ダウンロードはできません。

### 3.2.4 ログファイル

スクリプトコマンドの実行結果を保存するテキストファイルです。ログファイルは、ログオンからログオフまでのコマンド実行結果を格納しています。ファイル属性は、“`.log`”です。

### 3.2.5 ビューファイル

スクリプトウィンドウの表示内容を保存したテキストファイルです。**PD32000**では、スクリプトコマンド実行結果の最新 1000 行分をビューバッファといわれる領域に格納しています。ビューファイルは、このビューバッファの内容を格納するためのファイルです。ファイル属性は、“`.viw`”です。

### 3.2.6 ソフトウェアブレイクポイントファイル

ソフトウェアブレイクポイントファイルは、ソフトウェアブレイクポイントの設定内容を保存しているファイルです。ファイル属性は、“`.brk`”です。

本ファイルは、ソフトウェアブレイクポイント設定ダイアログから保存します。

### 3.2.7 トレースポイントファイル

トレースポイントファイルは、トレースポイントの設定内容を保存しているファイルです。ファイル属性は、“`.tc`”です。

本ファイルは、トレースコンディションウィンドウから保存します。

### 3.2.8 トレース計測結果保存ファイル

トレース計測結果保存ファイルは、**M32000T**のトレース機能により計測された結果を保存しているファイルです。ファイル属性は、".rtt"です。保存するトレース計測結果のファイル形式は、テキスト形式です(参照用)。

本ファイルは、トレースウィンドウから保存します。

### 3.2.9 エミュレーションメモリ状態ファイル

エミュレーションメモリ状態ファイルは、エミュレーションメモリの設定内容を保存しているファイルです。ファイル属性は、".em"です。

本ファイルは、エミュレーションメモリ設定ダイアログから保存します。

## 3.3 テンポラリファイル

### 3.3.1 オンデマンド方式指定時に作成するファイル

ターゲットプログラムをダウンロードする際に、デバッグ情報をオンデマンド方式で読み込むように指定していた場合、テンポラリファイルが作成されます。テンポラリファイルは、Initダイアログで指定されたディレクトリに作成します。ディレクトリの指定がない場合は、ダウンロードしたファイルがあるディレクトリに作成します。

テンポラリファイルはファイル名 `pdb_xxxx.tmp` (`xxxx` は 16 進 4 桁の数値になります) です。このファイルは、ダウンロードの直前および**PD32000**終了時に削除されます。

【 MEMO 】

# セットアップ編



# 1. セットアップ

## 1.1 インストール

PD32000のインストール方法については、製品に付属しているリリースノートをご参照下さい。

## 1.2 PD32000の起動

PD32000を起動するには、以下の操作を行ってください。

スタートボタンをクリックし、

プログラム(P)

[ MITSUBISHI TOOL ]

[ **PD32000** V.x.xx Release x ]

[ **PD32000** ] を選択してください。



上記操作を行うと**PD32000** (エミュレータデバugg) が起動します。

## 1.3 PD32000の動作環境の設定

PD32000を起動すると、Init ダイアログがオープンします。PD32000の動作環境は、この Init ダイアログで設定します。

### 1.3.1 Init ダイアログの構成

Init ダイアログの構成図を図 1に示します。

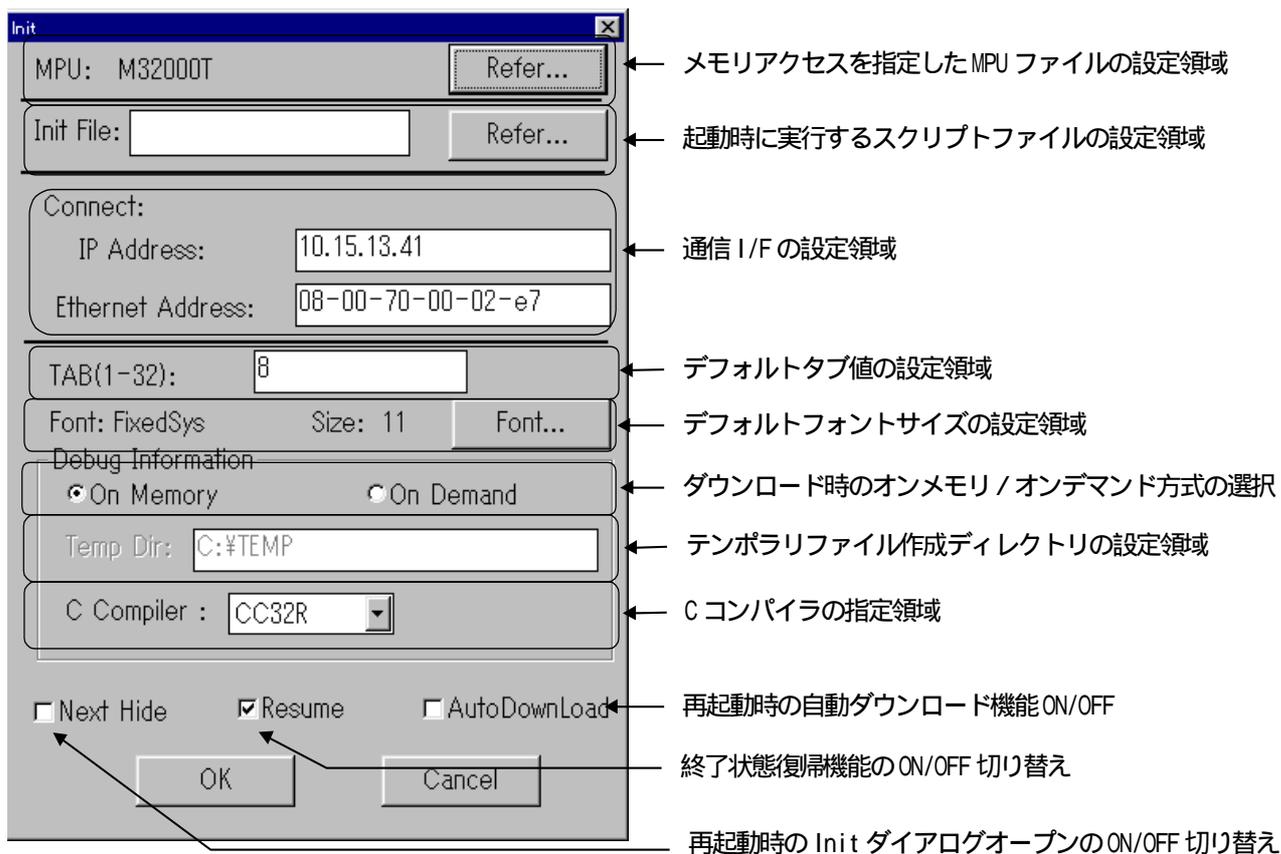


図 1 Init ダイアログの構成図

## 1.3.2 動作環境の設定

### メモリアクセス情報ファイル(MPU ファイル)の設定

ターゲットボード(MPU)のメモリアクセス情報を記述したファイルを指定します。“Refer”ボタンをクリックするとファイルセレクションダイアログがオープンします。ターゲットボードに対応した MPU ファイル名をマウスで選択してください。

なおターゲットボードに対応した MPU ファイルは、PD32000に添付している M32000T.MPU ファイルとは別に作成して頂く必要があります。

以下に MPU ファイルの記述例を示します。

<code>_MAP_START</code>	メモリアクセス情報開始の識別子
<code>4 00000000 001FFFFFF (*1)</code>	Word アクセス、0 ~ 1FFFFFF 番地
<code>2 00200000 009FFFFFF</code>	Hword アクセス、200000 ~ 9FFFFFF 番地
<code>0 00A00000 FEFFFFFF</code>	メモリなし、A00000 ~ FEFFFFFF 番地
<code>2 FF000000 FF1FFFFFF</code>	Hword アクセス、FF000000 ~ FF1FFFFFF 番地
<code>0 FF200000 FF5FFFFFF</code>	メモリなし、FF200000 ~ FF5FFFFFF 番地
<code>2 FF600000 FF7FFFFFF</code>	Hword アクセス、FF600000 ~ FF7FFFFFF 番地
<code>0 FF800000 FFBFFFFFF</code>	メモリなし、FF800000 ~ FFBFFFFFF 番地
<code>4 FFC00000 FFFFFFFF</code>	Word アクセス、FFC00000 ~ FFFFFFFF 番地
<code>_MAP_END</code>	メモリアクセス情報終了の識別子
<code>[EOF]</code>	

[補足：1行に記述する情報について(\*1)]

以下の2点の情報を1行に記述してください。

- ・メモリアクセスサイズ  
記述できる値は、0:メモリなし、1:Byte、2:Hword、4:Wordです。それ以外の値を記述した場合はエラーとなります。
- ・開始/終了アドレス  
開始アドレスの最下位8ビットは00、終了アドレスの最下位8ビットはFFとしてください。アドレス値は16進数で記述してください。

[注意1：M32000Tとターゲットボードを接続してデバッグする場合]

このファイルはターゲットボードの仕様に合わせて作成して頂く必要があります。上記の記述例を参考にして、MPU ファイルを作成してください。MPU ファイルの属性は.MPU としてください。

[注意2：指定アドレスに抜けがある場合]

アドレスの抜けとは、M32R/Dのメモリ空間で、アクセス情報を指定していない空間を意味します。MPU ファイルに抜けの空間がある場合、MPU ファイルエラーとなります。記述例のようにメモリがないアドレスも、メモリがないこと(0)を指定してください。

なお、**PD32000**ウィンドウのメニュー [ Environ ] [ Init ] を選択して MPU ファイルを設定した場合、その設定は次の起動時に有効になります。

### 起動時に実行するスクリプトファイルの設定

起動時、スクリプトコマンドを実行する場合に設定します。実行するスクリプトコマンドは、あらかじめスクリプトファイルに記述してください。“Refer”ボタンをクリックするとファイルセレクションダイアログがオープンします。起動時に実行するスクリプトファイル名をマウスで選択してください。選択されたスクリプトファイルは、Init ダイアログの Init File:領域に表示されます。

## 通信 I/F の設定

PD32000 起動時の LAN 通信で必要となる情報を設定します。

- IP アドレスには、接続するエミュレータの IP アドレスを指定します。IP アドレスは、10 進数で 1 バイトずつ、4 バイトをピリオドで区切って指定します。
- Ethernet アドレスには、接続するエミュレータの Ethernet アドレスを設定します。Ethernet アドレスは、16 進数で 1 バイトずつ XX-XX-XX-XX-XX-XX のフォーマットで入力してください。エミュレータは出荷時に、Ethernet アドレスを割り付けています。

## デフォルトタブ値の設定

プログラムウィンドウ、ソースウィンドウのデフォルトタブ値を設定します。タブ値には、1 ~ 32 までの数値が指定できます。

### 補足

タブ値は、ウィンドウ毎に設定することもできます。この場合、対象ウィンドウがアクティブな状態で PD32000 ウィンドウのメニュー [ Option ] [ TAB ] を選択してください。TAB ダイアログがオープンしますのでタブ値を指定してください。

## デフォルトフォントサイズの設定

PD32000 で表示する文字のデフォルトフォントを指定します。“Font”ボタンをクリックするとフォント指定ダイアログがオープンしますのでフォントとフォントサイズを指定してください。

### 補足

フォントサイズは、ウィンドウ毎に設定することもできます。この場合、対象ウィンドウがアクティブな状態で PD32000 ウィンドウのメニュー [ Option ] [ Font ] を選択してください。フォント指定ダイアログがオープンしますのでフォントとフォントサイズを指定してください。

## ダウンロード時のオンメモリ / オンデマンド方式の選択

ターゲットプログラムをダウンロードした際に、デバッグ情報をオンメモリ方式で読み込むか、オンデマンド方式で読み込むかを選択します。オンメモリ方式とは、メモリ上に全てのデバッグ情報を保持する方式です。オンデマンド方式とは、ターゲットプログラムをダウンロードした際に、テンポラリファイルを作成し、必要なデバッグ情報を必要になった時にメモリ上に読み込む方式です。

なお、PD32000 ウィンドウのメニュー [ Environ ] [ Init ] を選択して Init ダイアログから設定した場合、その設定は次のダウンロード時に有効になります。

### オンデマンド時のテンポラリディレクトリ指定

**PD32000**では、オンデマンド方式を選択してダウンロードした際、テンポラリファイルを作成します。**PD32000**では、指定したディレクトリにテンポラリファイルを作成します。ディレクトリの指定が無い場合は、ダウンロードしたファイルがあるディレクトリに作成します。

### C コンパイラの指定

M32R/D 用コンパイラは、**CC32R**、TW32R、及びDCCの3種類があります。各コンパイラが出力するファイルのデータ構造が異なるため、対象コンパイラの指定が必要です。デバッグするプログラムをコンパイルしたコンパイラを指定して下さい。

### 再起動時の Init ダイアログオープンの ON/OFF

PD32000再起動時に Init ダイアログをオープンするか否かを指定します。Next Hide をチェックした場合、次回から Init ダイアログがオープンしません。Init ダイアログを再オープンさせるには、**PD32000**ウィンドウのメニュー [ Environ ] [ Init ] を選択し、Init ダイアログから Next Hide のチェックを解除してください。または、Ctrl キーを押しながらPD32000を起動すると、起動時に強制的に Init ダイアログを表示することができます。

### 終了状態復帰機能の ON/OFF

前回終了したときのウィンドウの表示状態で**PD32000**を起動するか否かを指定します。Resume をチェックした場合、前回終了したときとおなじウィンドウの表示状態で**PD32000**が起動します。

### 起動時の自動ダウンロード機能

最後に読み込んだターゲットプログラムを**PD32000**起動時に読み込むか否かを指定します。AutoDownload をチェックした場合、**PD32000**起動時にターゲットプログラムを読み込みます。

【 MEMO 】

# ウィンドウ機能編

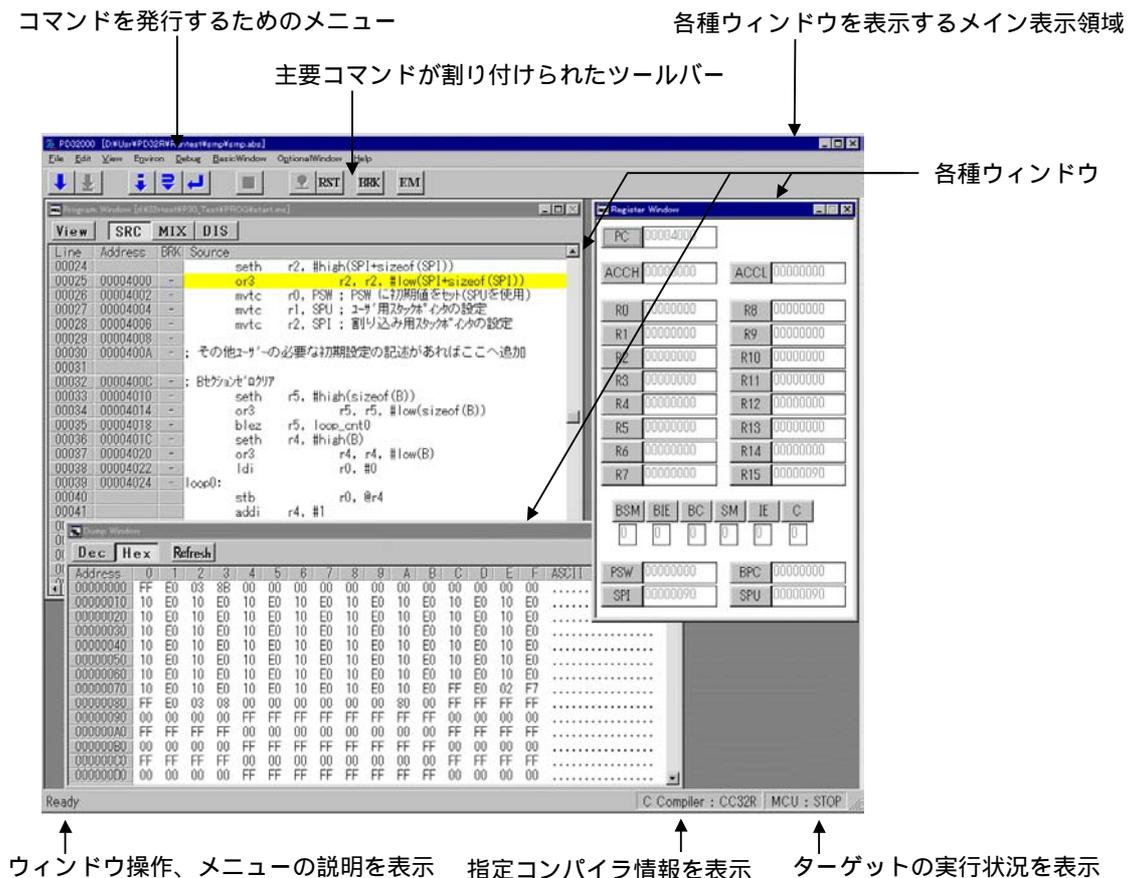


# 1 PD32000のウィンドウ機能

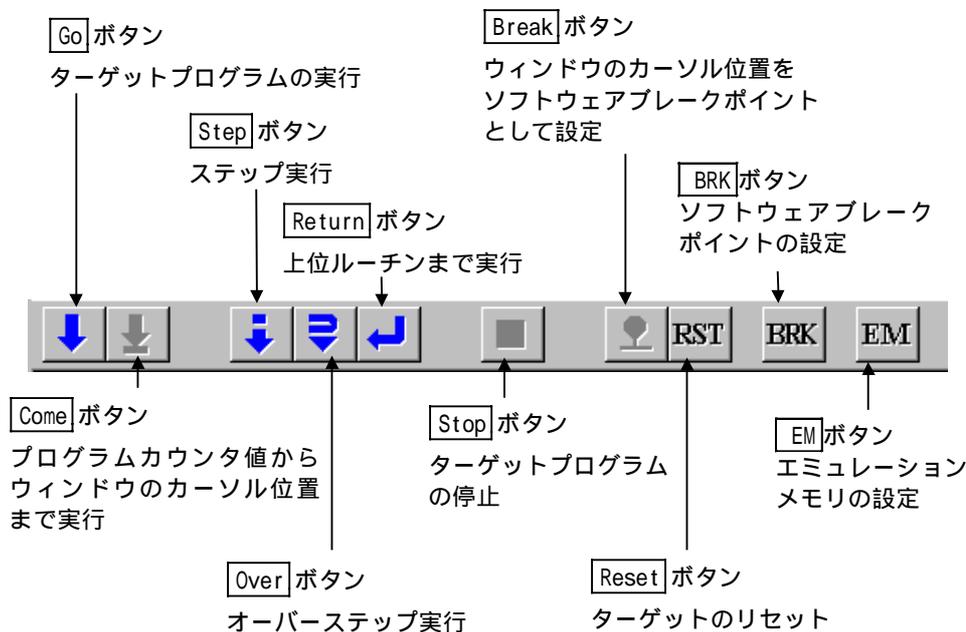
## 1.1 PD32000ウィンドウ

PD32000ウィンドウは、PD32000のメインウィンドウです。PD32000を起動した際に、最初にオープンします。PD32000ウィンドウでは、主要コマンドをツールバーに割り付けています。ツールバーのボタンをクリックすることにより、ターゲットプログラムの実行、ステップ実行等が容易に操作できます。また、メイン表示領域には、ターゲットプログラム表示用ウィンドウ等の各種ウィンドウを表示します。

### 1.1.1 PD32000ウィンドウの画面構成

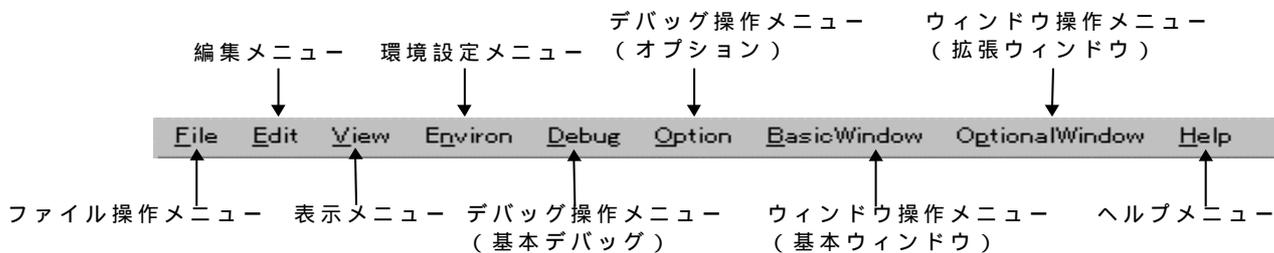


### 1.1.2 PD32000ウィンドウのツールバー



### 1.1.3 PD32000ウィンドウのメニュー

PD32000ウィンドウのメニューは、基本メニューと拡張メニューに分類することができます。



#### 「基本メニュー」と「拡張メニュー」について

PD32000ウィンドウのメニュー項目のうち Option メニューについては、PD32000ウィンドウのメイン表示領域に表示されたアクティブなウィンドウに応じてサブメニュー項目が自動的に切り替わります。この Option メニューのことを「拡張メニュー」と呼びます。

これに対し、Option メニュー以外のすべてのメニューは、アクティブなウィンドウの変化によらずメニュー項目が常に一定です。これらを「基本メニュー」と呼びます。基本メニューは、PD32000の基本操作およびデバッグ操作を行うためのメニューを備えています。

以下に、基本メニューの各項目の機能について説明します。なお、拡張メニューの各項目の機能については、各ウィンドウで説明します。

### 1.1.3.1 ファイル操作

[ File ] メニューには、PD32000の機能のうち、ファイルの読み込みや保存、PD32000の終了等、ファイル操作に関するメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
File	Download	ターゲットプログラムをダウンロード	Shift + F.1
	Load Module...	機械語データとデバッグ情報をダウンロード	
	Memory Image...	機械語データのみダウンロード	
	Binary...	バイナリデータのダウンロード	
	Symbol...	デバッグ情報のみダウンロード	
	Reload...	ターゲットプログラムの再ダウンロード	
	Upload...	ターゲットプログラムのアップロード	
	Memory Image...	モトローラ形式(S3)でのアップロード	
	Binary	バイナリデータでのアップロード	
	Save Disasm...	逆アセンブル結果の保存	
	Exit	PD32000を終了	

### 1.1.3.2 編集

[ Edit ] メニューには、PD32000の機能のうち、文字列のコピー、ペースト、検索等、編集操作に関するメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Edit	Copy	選択した文字列をクリップボードにコピー	Ctrl + C
	Paste	クリップボードの文字列を貼り付け	Ctrl + V
	Find...	文字列の検索	

### 1.1.3.3 表示

[ View ] メニューには、PD32000の機能のうち、ツールバーやステータスバーの表示切り替えに関するメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
View	Tool Bar	ツールバーの表示 / 非表示の切り替え	
	Status Bar	ステータスバーの表示 / 非表示の切り替え	

### 1.1.3.4 環境設定

[ Environ ]メニューには、PD32000の機能のうち、環境設定に関するメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Environ	Init...	環境設定	
	Path...	ソースファイルのサーチパス	
	Start Up...	スタートアップ回数数の設定	

### 1.1.3.5 デバッグ操作 (基本デバッグ)

[ Debug ]メニューには、PD32000の機能のうち、ターゲットの実行/停止やステップ実行等、基本的なデバッグ操作に関するメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Debug	Go	ターゲットプログラムの実行	
	Go	現PCからの実行	F.1
	Go Option...	指定アドレスからの実行	
	Come	カーソル位置までの実行	F.2
	Step	ステップ実行	
	Step	一回のステップ実行	F.3
	Step Option...	指定回数のステップ実行	
	Over	オーバーステップ実行	
	Oyer	一回のオーバーステップ実行	F.4
	Over Option...	指定回数のオーバーステップ実行	
	Return	現サブルーチンの復帰まで実行	F.5
	Break Point	ブレークポイントの設定	
	Break Point...	S/W ブレークポイント設定ダイアログをオープン	F.7
	Break	カーソル位置にソフトウェアブレークを設定/解除	
Reset	ターゲットのリセット	F.8	
Reset Go	ターゲットのリセット後 実行		
Stop	ターゲットプログラムの実行停止		
Emulation Memory	エミュレーションメモリの設定		
Entry	メイクファイルの登録		
Make	ターゲットプログラムのメイク		

### 1.1.3.6 デバッグ操作 (オプション)

拡張メニューには、PD32000が表示する各ウィンドウを操作するためのメニューが割り当てられます。拡張メニューの下は、アクティブなウィンドウの変化によって、メニュー項目が変化します。拡張メニューの各項目の機能については、各ウィンドウの説明をご参照下さい。

メニュー	メニュー項目	機能	ショートカットキー
Option		(各ウィンドウの拡張メニューが追加されます。)	

### 1.1.3.7 ウィンドウ操作

[ Basic Window ] メニューには、PD32000の機能のうち、PD32000が表示する各ウィンドウの表示形態を操作するメニューおよび基本的なウィンドウをオープンするメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Basic Window	Cascade	ウィンドウを重ねて表示	
	Tile	ウィンドウを並べて表示	
	Arrange Icon	アイコンを整列	
	Program Window	プログラムウィンドウをアクティブ	
	Source Window	ソースウィンドウをオープン	
	Register Window	レジスタウィンドウをオープン	
	Memory Window	メモリウィンドウをオープン	
	Dump Window	ダンプウィンドウをオープン	
	ASM Watch Window	ASM ウォッチウィンドウをオープン	
	C Watch Window	C 言語レベルのウォッチウィンドウをオープン	
	C Watch Window	C ウォッチウィンドウをオープン	
	Local Window	ローカルウィンドウをオープン	
	File Local Window	ファイルローカルウィンドウをオープン	
	Global Window	グローバルウィンドウをオープン	
	Script Window	スクリプトウィンドウをオープン	

[ Optional Window ] メニューには、PD32000の機能のうち、より高機能なウィンドウをオープンするメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Optional Window	Trace Window	トレースウィンドウをオープン	
	Trace Condition	トレース設定ウィンドウをオープン	
	MR Window	MR ウィンドウをオープン	
	Time Measurement Window	区間時間計測ウィンドウをオープン	
	Call Stack Window	コールスタックウィンドウをオープン	
	Custom Window		
	Option	カスタムウィンドウの登録	
ユーザー定義メニュー	登録したカスタムウィンドウのオープン		

### 1.1.3.8 ヘルプ

[ Help ] メニューには、PD32000の機能のうち、PD32000のヘルプやバージョンを表示するためのメニューが割り当てられています。

メニュー	メニュー項目	機能	ショートカットキー
Help	Index	ヘルプ表示	
	About...	PD32000 のバージョン情報表示	

## 1.2 プログラムウィンドウ

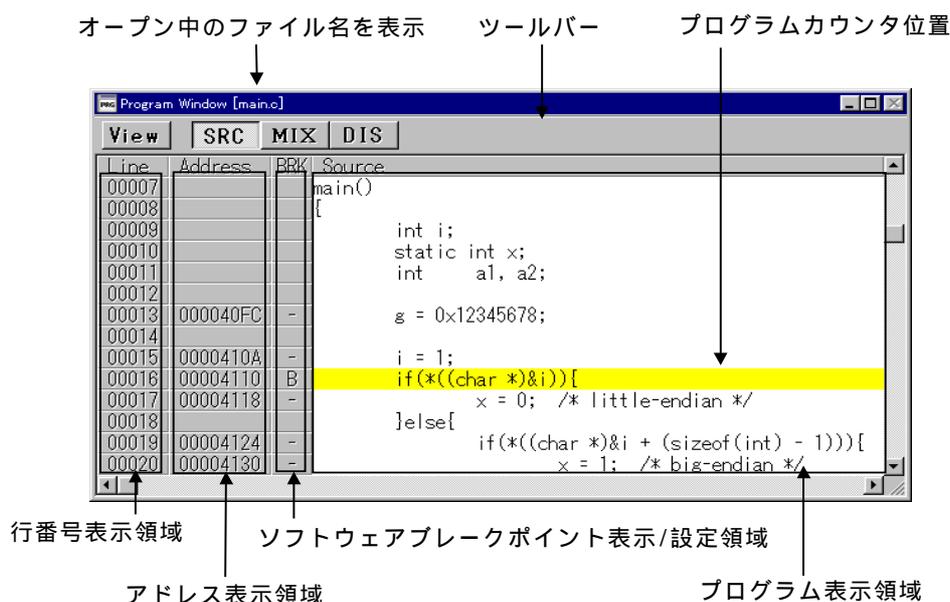
プログラムウィンドウは、現在のプログラムカウンタに相当するプログラムを常に表示するウィンドウです。プログラムカウンタに相当する行は、黄色の反転ラインで表示します。プログラムウィンドウは、PD32000を起動した際、PD32000ウィンドウのメイン表示領域内に自動的にオープンします。プログラムウィンドウでは、カーソル位置までの実行、マウスによるソフトウェアブレークポイントの設定 / 解除、逆アセンブル表示等が行えます。ソフトウェアブレークポイントの設定 / 解除は、ソフトウェアブレークポイント表示 / 設定領域をダブルクリックすることによって行えます。プログラム表示領域をクリックしてメニュー [ Option ] [ Line Assemble ] を選択することによりクリック位置からラインアセンブルを行うことができます。

### 1.2.1 プログラムウィンドウの画面構成

プログラムウィンドウには、ソース、逆アセンブルと両者を同時に表示する3種類の表示モードがあります。以下に3種類の表示モードにおけるプログラムウィンドウの画面構成を示します。

#### 1.2.1.1 ソース表示モード時のプログラムウィンドウの画面構成

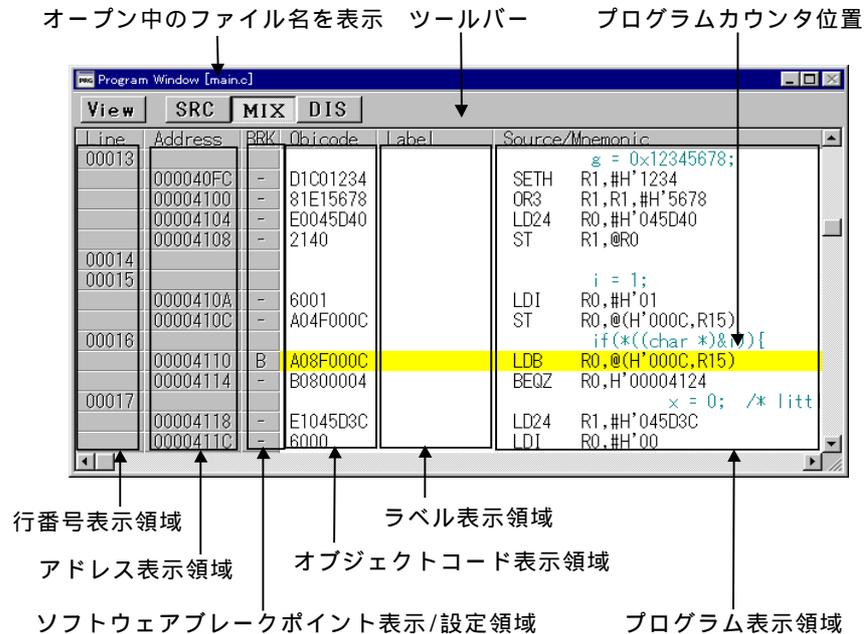
ソース表示モードは、ソースレベルでデバッグするためのモードです。ソース表示モードでは、ターゲットプログラムのソースファイルが参照できます。



- 行番号表示領域、アドレス表示領域は、それぞれメニュー [ Option ] [ Layout ] [ Line Area ]、[ Option ] [ Layout ] [ Address Area ] の選択 / 解除によって、表示 / 非表示にすることができます。なお、アドレス表示領域は、デフォルトでは非表示になっています。
- 行番号表示領域をダブルクリックすることで、表示するソースファイルを変更することができます。
- アドレス表示領域をダブルクリックすることで、表示開始アドレス、または表示開始行を変更することができます。

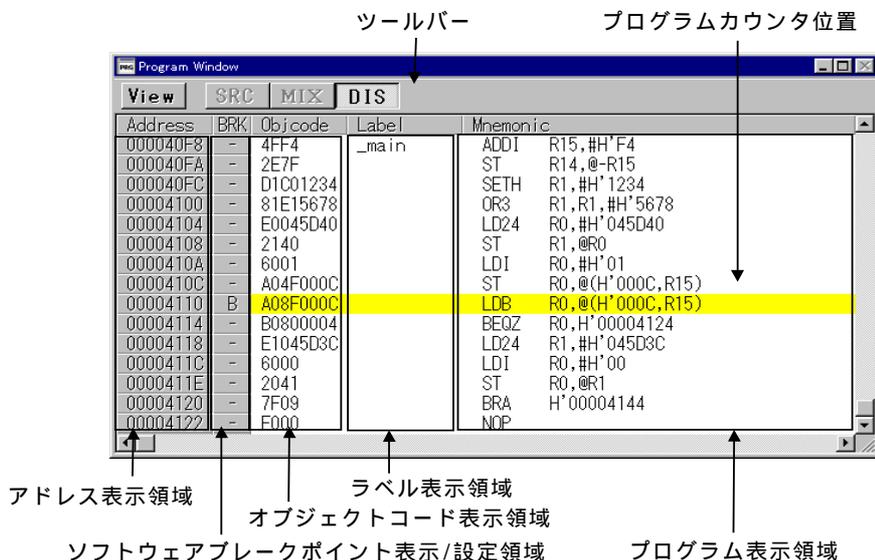
### 1.2.1.2 MIX 表示モード時のプログラムウィンドウの画面構成

MIX 表示モードは、ソースプログラムとその部分の逆アセンブル結果を混合して表示します。ソースプログラムと逆アセンブル結果は異なった色で表示します。



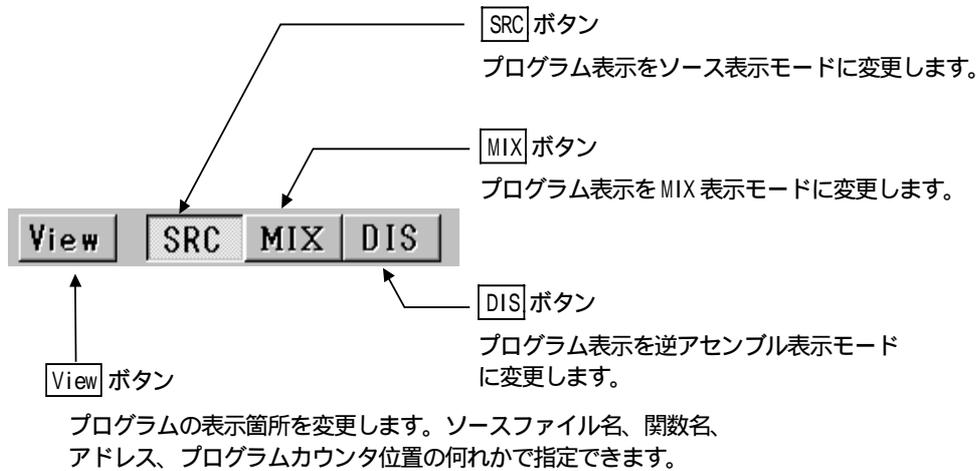
## 1.2.1.3 逆アセンブル表示モード時のプログラムウィンドウの画面構成

逆アセンブル表示モードは、命令レベルでデバッグするためのモードです。逆アセンブル表示モードでは、ターゲットプログラムの逆アセンブル結果が参照できます。



- アドレス表示領域、オブジェクトコード表示領域は、それぞれメニュー [ Option ] [ Layout ] [ Address Area ]、[ Option ] [ Layout ] [ Code Area ] の選択 / 解除によって、表示 / 非表示にすることができます。
- 垂直方向のスクロールをする場合、逆方向のスクロールは、順方向のスクロールの後でないと使用できません。順方向へのスクロールでは、以前の表示アドレスを内部バッファに保存します。逆方向へのスクロールは、このアドレス情報を利用して実現しています。なお、コマンド実行によって、先頭行アドレスが変更された場合、内部バッファの内容はクリアされます。
- プログラム以外の領域（データ・空き領域など）を逆アセンブルした場合、そのメモリ内容を命令コードとみなして逆アセンブル表示します。その際、未定義命令・未定義オペランドとなった場合、“???” を表示します。
- アドレス表示領域をダブルクリックすることで、表示開始アドレスを変更することができます。

## 1.2.2 プログラムウィンドウのツールバー



## 1.2.3 プログラムウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、プログラムウィンドウがアクティブな場合は、[ Option ]メニューには以下の拡張メニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	TAB...	ソースファイル表示のタブ設定	
	View	表示内容の変更	
	Source...	指定ソースファイル・関数からの表示	
	Address...	指定アドレスまたは行番号からの表示	
	Program Counter	現在のプログラムカウンタからの表示	
	Mode	表示モードの変更	
	Source mode	ソース表示モードへ変更	Ctrl+R
	Mix mode	MIX表示モードへ変更	Ctrl+R
	Disasm mode	逆アセンブル表示モードへ変更	Ctrl+R
Layout	Line Area	行番号表示領域の表示/非表示	
	Address Area	アドレス表示領域の表示/非表示	
	Code Area	オブジェクトコード表示領域の表示/非表示	
Line Assemble	ラインアセンブルダイアログのオープン	Ctrl+L	

## 1.3 ソースウィンドウ

ソースウィンドウは、特定のプログラムを継続して参照するためのウィンドウです。プログラムカウンタに相当する行は、黄色の反転ラインで表示します。前記のプログラムウィンドウがプログラムカウンタに追従して表示内容を更新するのに対し、ソースウィンドウは、ユーザが指定しない限り表示内容を更新しません。したがって、ソースウィンドウは、特定のサブルーチンやタスクを継続して参照される際にご使用下さい。ソースウィンドウは、計 10 枚までオープンできます。その他の機能は、プログラムウィンドウと同等です。

### 1.3.1 ソースウィンドウの画面構成

ソースウィンドウには、ソース、逆アセンブルと両者を同時に表示する 3 種類の表示モードがあります。以下に 3 種類の表示モードにおけるプログラムウィンドウの画面構成を示します。

なお、ソースウィンドウの画面構成は、プログラムウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.2.1 プログラムウィンドウの画面構成」をご参照下さい。

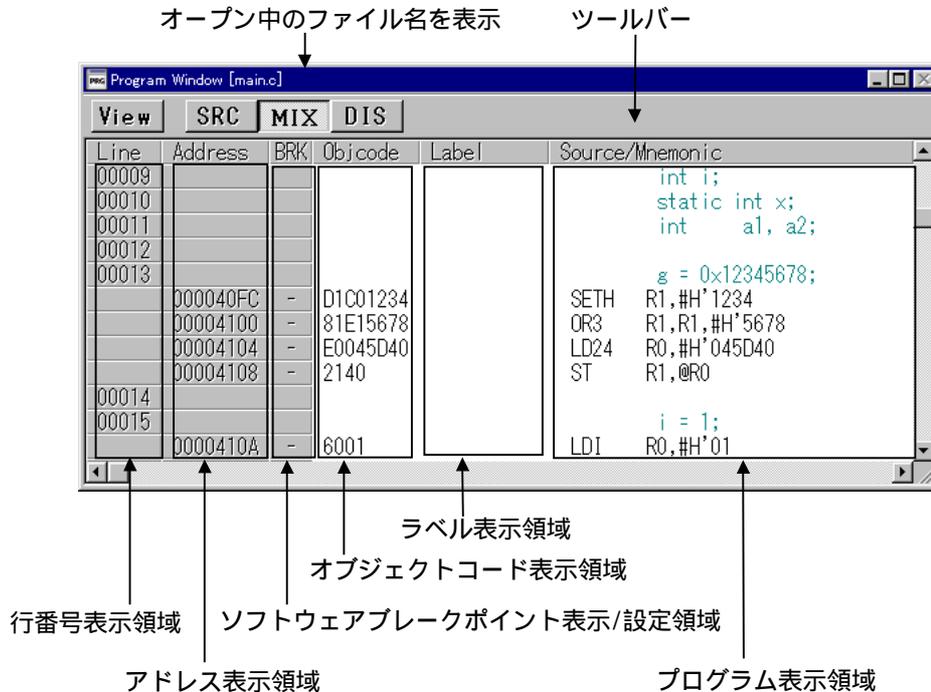
#### 1.3.1.1 ソース表示モード時のソースウィンドウの画面構成

ソース表示モードは、ソースレベルでデバッグするためのモードです。ソース表示モードでは、ターゲットプログラムのソースファイルが参照できます。



### 1.3.1.2 MIX 表示モード時のプログラムウィンドウの画面構成

MIX 表示モードは、ソースプログラムとその部分の逆アセンブル結果を混合して表示します。ソースプログラムと逆アセンブル結果は異なった色で表示します。



### 1.3.1.3 逆アセンブル表示モード時のソースウィンドウの画面構成

逆アセンブル表示モードは、命令レベルでデバッグするためのモードです。逆アセンブル表示モードでは、ターゲットプログラムの逆アセンブル結果が参照できます。



### 1.3.2 ソースウィンドウのツールバー

ソースウィンドウのツールバーは、プログラムウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.2.2 プログラムウィンドウのツールバー」をご参照下さい。

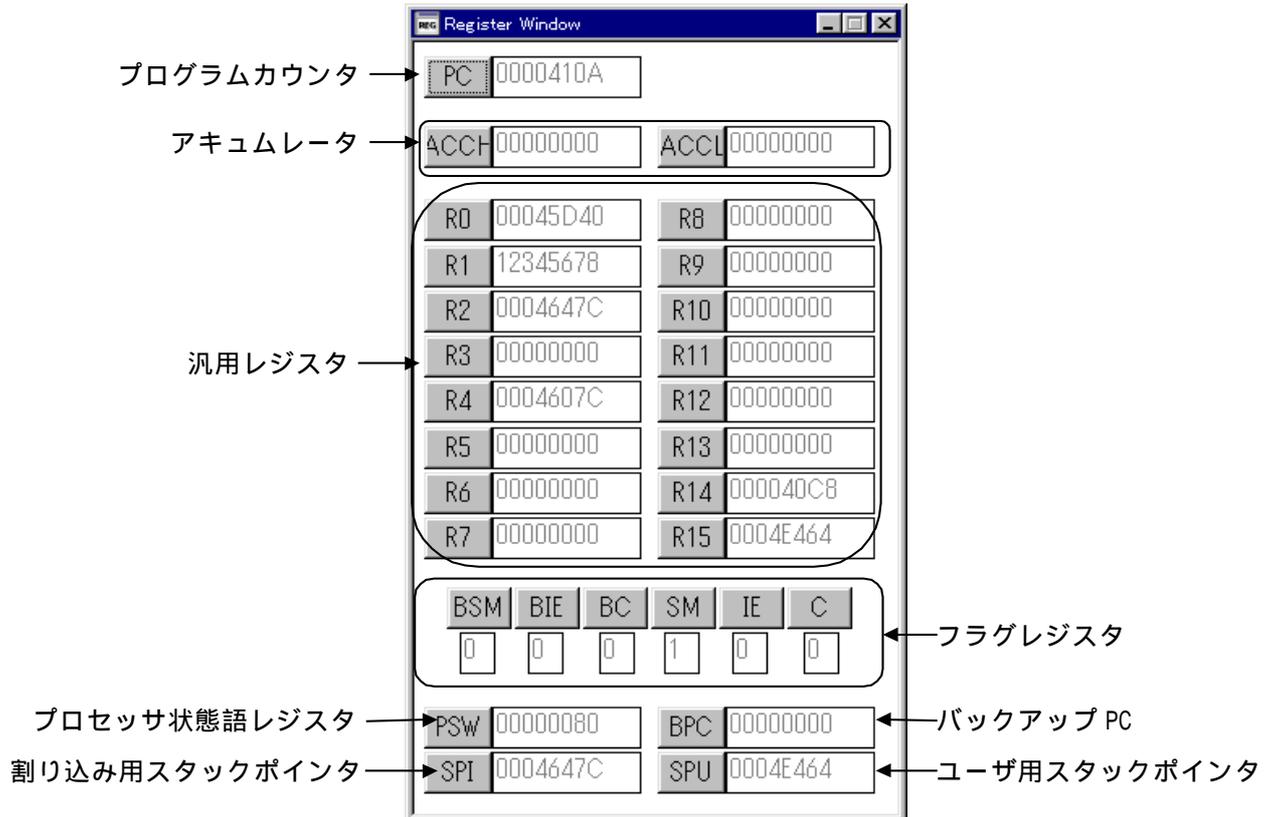
### 1.3.3 ソースウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、ソースウィンドウがアクティブな場合は、[ Option ]メニューにはソースウィンドウの拡張メニューが割り当てられます。ソースウィンドウの拡張メニューは、プログラムウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.2.3 プログラムウィンドウの拡張メニュー」をご参照下さい。

## 1.4 レジスタウィンドウ

レジスタウィンドウは、レジスタの内容やフラグの内容を表示するウィンドウです。表示内容は、各コマンド実行後に更新します。レジスタウィンドウでは、レジスタに対応したボタンをクリックすることにより、レジスタ値やフラグ値が容易に変更できます。

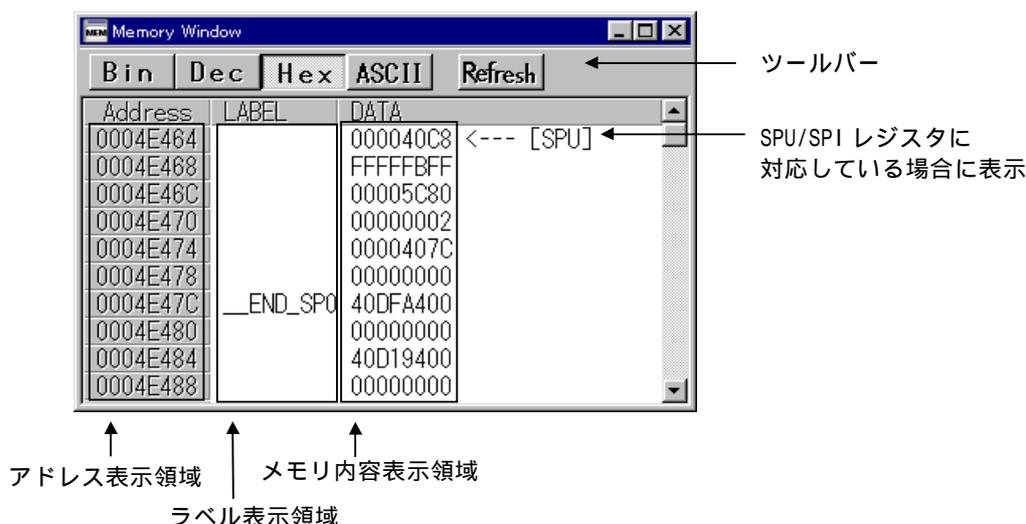
### 1.4.1 レジスタウィンドウの画面構成



## 1.5 メモリウィンドウ

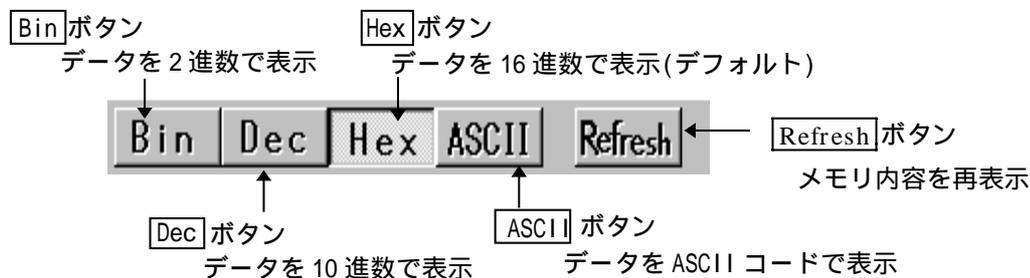
メモリウィンドウは、連続したメモリ内容を「アドレス」「ラベル」「データ(メモリ内容)」の書式で表示するウィンドウです。表示内容は、各コマンド実行後に更新します。データの表示は、2進数、10進数、16進数、ASCII表示が可能です。メモリウィンドウは、計10枚までオープンできます。メモリウィンドウでは、メモリ内容の変更や指定メモリ領域の充填/移動が容易に行えます。

### 1.5.1 メモリウィンドウの画面構成



- アドレス表示領域をダブルクリックすることで、表示開始アドレスを変更することができます。
- ラベル、またはメモリ内容表示領域をダブルクリックすることで、メモリの内容を変更することができます。

### 1.5.2 メモリウィンドウのツールバー



### 1.5.3 メモリウィンドウの拡張メニュー

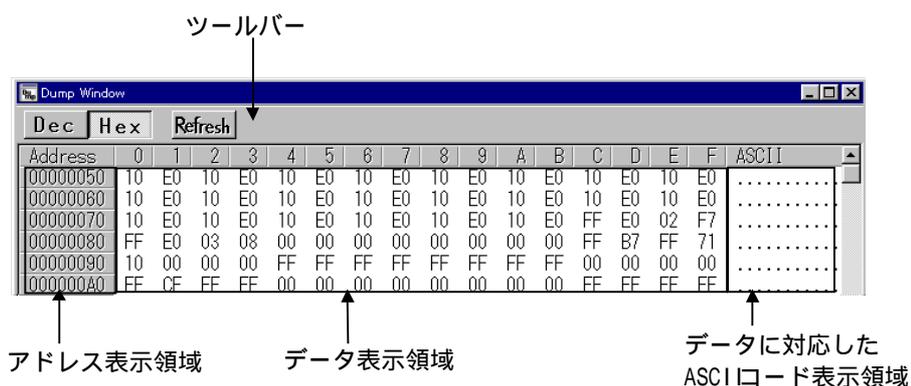
PD32000のメイン表示領域に表示されたウィンドウのうち、メモリウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	View	表示内容の変更	
	Scroll Area...	スクロール範囲の指定	
	Address...	表示開始アドレスの指定	
	SPU	表示開始アドレスをSPU レジスタ値に変更	
	SPI	表示開始アドレスをSPI レジスタ値に変更	
	Data Length	表示データ長の指定	
	Byte	1 バイト単位で表示	
	Hword	2 バイト単位で表示	
	Word	4 バイト単位で表示	
	Radix	表示基数の指定	
	Bin	2 進数で表示	
	Dec	10 進数で表示	
	Hex	16 進数で表示	
	Ascii	ASCII 文字で表示	
	Refresh	データの再表示	
	Debug	メモリ内容の設定	
	Set...	指定アドレスにデータを設定	
	Fill...	指定したメモリブロックにデータ充填	
Move...	指定したメモリブロックを指定アドレスに移動		

## 1.6 ダンプウィンドウ

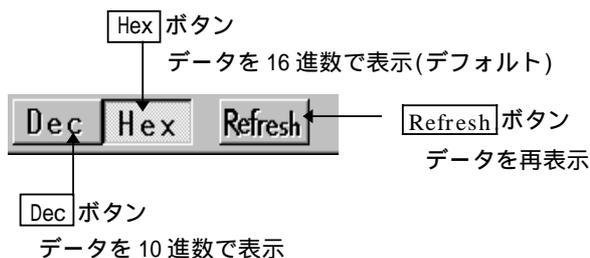
ダンプウィンドウは、連続したメモリ内容をダンプ形式で表示するウィンドウです。表示内容は、各コマンド実行後に更新します。ダンプウィンドウは、計 10 枚までオープンできます。ダンプウィンドウでは、メモリウィンドウと同様にメモリ内容の変更や指定メモリ領域の充填 / 移動が容易に行えます。

### 1.6.1 ダンプウィンドウの画面構成



- アドレス表示領域をダブルクリックすることで、表示開始アドレスを変更することができます。
- ラベル、またはメモリ内容表示領域をダブルクリックすることで、メモリの内容を変更することができます。

### 1.6.2 ダンプウィンドウのツールバー



### 1.6.3 ダンプウィンドウの拡張メニュー

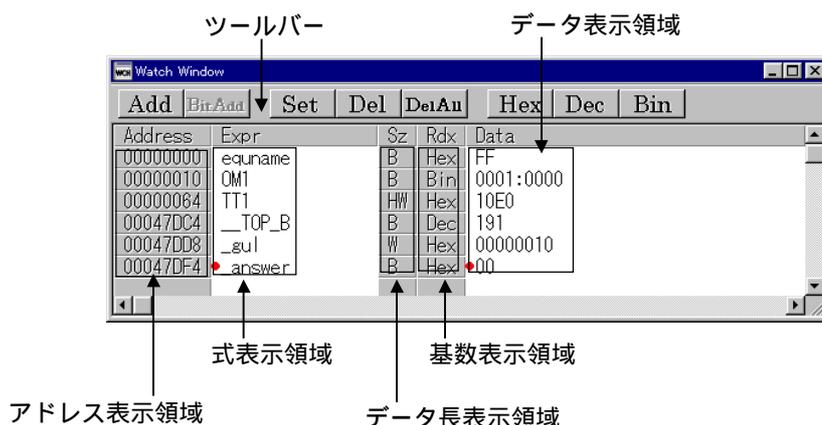
PD32000のメイン表示領域に表示されたウィンドウのうち、ダンプウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	View	表示内容の変更	
	Scroll Area...	スクロール範囲の指定	
	Address...	表示開始アドレスの指定	
	Data Length	表示データ長の指定	
	Byte	1 バイト単位で表示	
	Hword	2 バイト単位で表示	
	Word	4 バイト単位で表示	
	Radix	表示基数の指定	
	Dec	10 進数で表示	
	Hex	16 進数で表示	
	Refresh	データの再表示	
	Debug	メモリ内容の設定	
	Set...	指定アドレスにデータを設定	
Fill...	指定したメモリブロックにデータ充填		
Move...	指定したメモリブロックを指定アドレスに移動		

## 1.7 ASM ウォッチウィンドウ

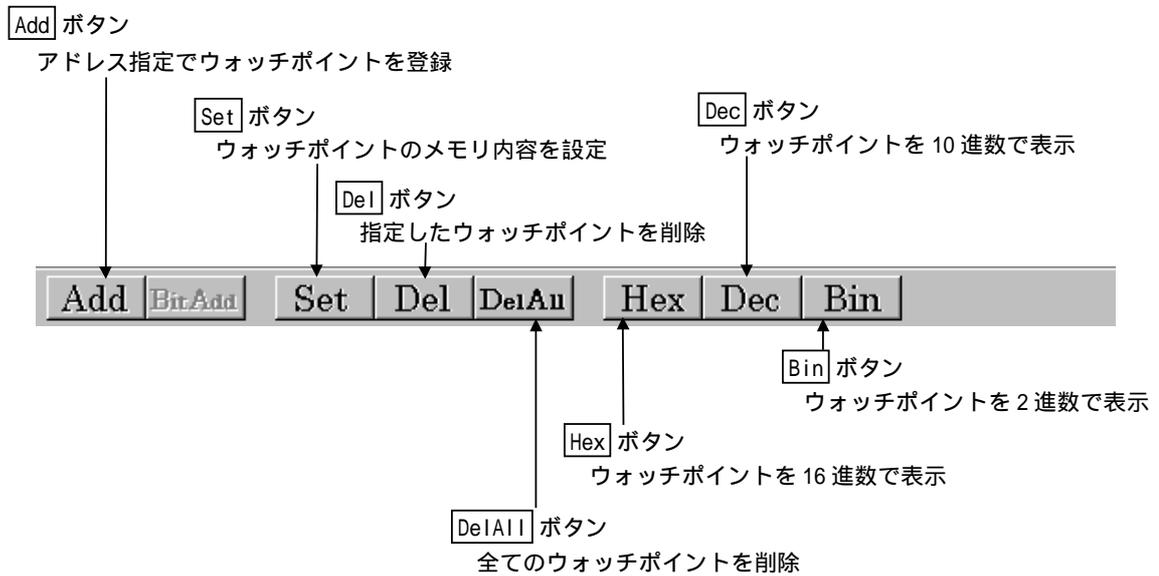
ASM ウォッチウィンドウは、任意アドレスの内容を参照するウィンドウです。この参照する任意アドレスをウォッチポイントと呼びます。ウォッチポイントは、アドレス(シンボルの記述可能)で指定することができます。表示内容は、各コマンド実行後に更新されます。

### 1.7.1 ASM ウォッチウィンドウの画面構成



- アドレス式表示領域とデータ表示領域には、赤いマークでカーソル位置が表示されます。カーソル位置は、どちらかの領域をクリックするか、または   キーで移動できます。
- 基数表示領域をダブルクリックすると、その位置のデータ表示の基数が、現在の表示基数から、  
 …… 16進数 10進数 2進数 16進数 ……  
 のローテーションで変更されます。
- 設定したウォッチポイントの情報は、ASM ウォッチウィンドウをクローズする際、またはPD32000を終了する際に初期化ファイルへ保存します。これにより、ASM ウォッチウィンドウを再オープンした際に、以前のウォッチポイントを自動的に登録します。
- ASM ウォッチウィンドウは、ターゲットプログラムをダウンロードした際、既に登録済みのウォッチポイントのアドレス式を再計算し、新たなアドレスでメモリ内容を参照します。これにより、プログラムの変更に伴ってウォッチポイントのアドレスが変化した場合でも、アドレスを再設定する必要がありません。
  - 無効なウォッチポイント(メモリ値が "--<not active>--" と表示)については、再計算の結果、アドレス式が正しく計算できた場合、有効なウォッチポイントになります。

## 1.7.2 ASM ウォッチウィンドウのツールバー



## 1.7.3 ASM ウォッチウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、ASM ウォッチウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	Watch	ウォッチポイントの登録/削除	
	Add...	ウォッチポイントの登録	Ctrl+A
	Set...	選択位置のウォッチポイントのメモリ内容の設定	Ctrl+S
	Del	選択位置のウォッチポイントの削除	Ctrl+D
	DelAll	全ウォッチポイントの削除	
	Radix	表示基数の変更	
	Bin	選択位置のウォッチポイントの値を2進数で表示	Alt+B
	Dec	選択位置のウォッチポイントの値を10進数で表示	Alt+D
	Hex	選択位置のウォッチポイントの値を16進数で表示	Alt+H
	Layout	レイアウト設定	
	Address Area	アドレス表示領域の表示/非表示	
	Size Area	データ長表示領域の表示/非表示	

## 1.8 C ウォッチウィンドウ

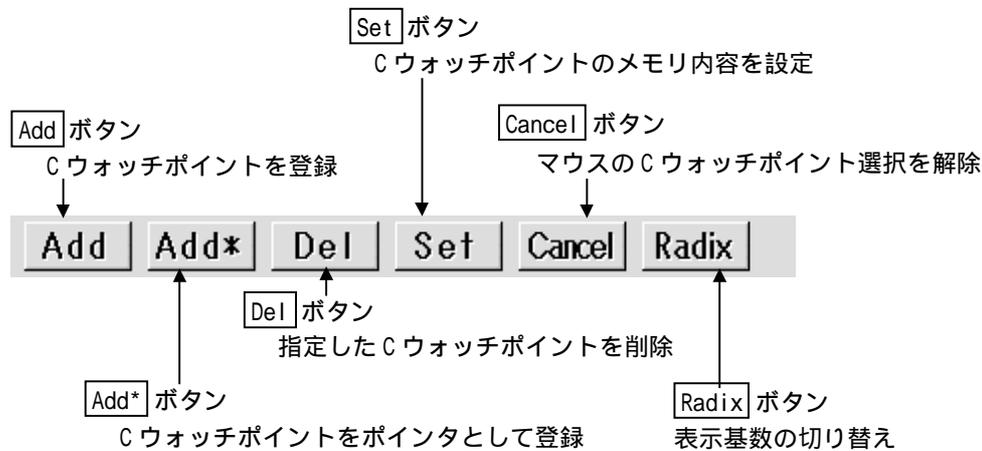
C ウォッチウィンドウは、C 言語の式（以下、C 言語式）とその値（計算結果）を表示するウィンドウです。このCウォッチウィンドウの表示対象となるC言語式のことをCウォッチポイントと呼びます。Cウォッチポイントの計算結果の表示は、各コマンド実行後に更新されます。

### 1.8.1 C ウォッチウィンドウの画面構成



- Cウォッチポイントには、以下のものが使用できます。
  - Cシンボル  
C言語ソースプログラムで定義されている変数名および関数名。
  - Cシンボルを含む式（C言語式）  
C言語式に使用できる字句（トークン）については、本マニュアル リファレンス編の項目「3.1 C言語式の記述方法」をご参照下さい。
- C言語式が正しく計算できなかった場合、（例えばシンボル未定義の場合）でも、無効なCウォッチポイントとして登録します。無効なCウォッチポイントは、計算結果表示の対象になりません。ただし、無効なCウォッチポイントは再計算を行った結果、正しく計算できた場合には有効なCウォッチポイントに変わります。
- 以下に示すCウォッチポイントには、値を代入することはできません。
  - 浮動小数点型変数
  - ビットフィールド型変数
  - レジスタ変数
  - その他、アドレスを示さないCウォッチポイント
- 設定したCウォッチポイントの情報は、Cウォッチウィンドウをクローズする際、またはPD32000を終了する際にCウォッチポイント情報格納ファイルへ保存します。これにより、Cウォッチウィンドウを再オープンした際に、以前のCウォッチポイントを自動的に登録します。
  - Cウォッチポイント情報格納ファイルは、ロードしたオブジェクトファイルごとに別々に作成されます（ロードしたオブジェクトファイル名の情報が含まれます）。  
Cウォッチポイントを再登録する際は、まず現在ロードされているオブジェクトファイルと同じオブジェクトファイル名の情報を持つCウォッチポイント情報格納ファイルを検索します。検索の結果、該当するファイルが見つければ、そのファイルの情報から得たCウォッチポイントを再登録します。

## 1.8.2 C ウォッチウィンドウのツールバー



- C ウォッチポイント表示領域上でクリックして選択した C ウォッチポイントに対して、削除や値の設定が行えます。なお、構造体の内容（メンバ）など C ウォッチポイントを登録する際に付加情報として表示されたものは、その付加情報だけを削除することはできません。
- ポインタなどのアドレス表示は、表示基数にかかわらず 16 進表示で固定です。なお、表示形式の詳細については、本マニュアル リファレンス編の項目「3.2 C 言語式の表示形式」をご参照下さい。

## 1.8.3 C ウォッチウィンドウの拡張メニュー

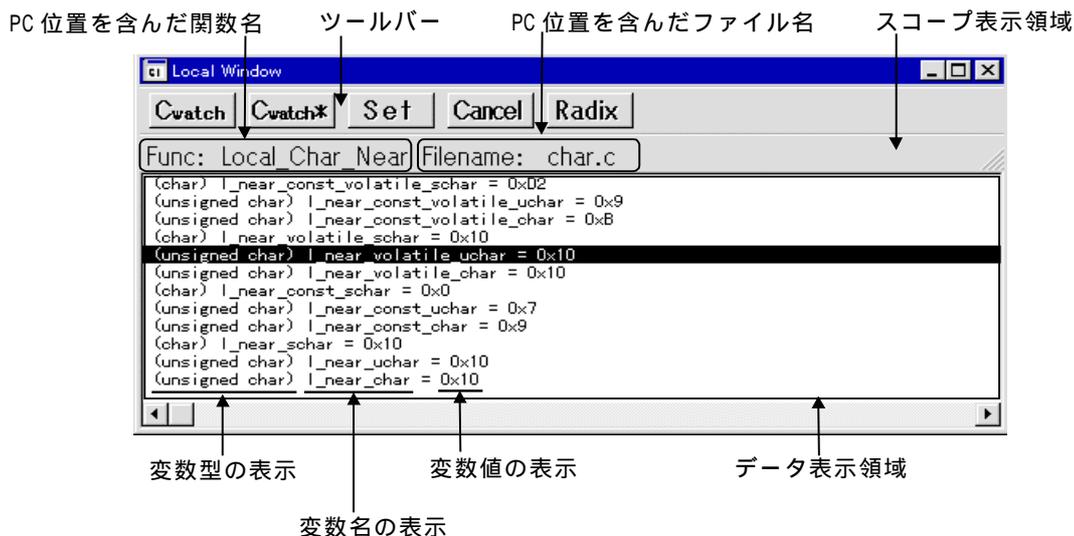
PD32000のメイン表示領域に表示されたウィンドウのうち、C ウォッチウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	W <u>atch</u>	C ウォッチポイントの登録/削除	
	<u>A</u> dd...	C ウォッチポイントの登録	
	A <u>dd</u> P <u>ointer</u> ...	C ウォッチポイントの登録 (ポインタ)	
	<u>D</u> el	選択位置のC ウォッチポイントの削除	
	<u>S</u> et...	選択位置のC ウォッチポイントの値を設定	
	<u>C</u> ancel	C ウォッチポイントの選択の解除	
	<u>D</u> el <u>A</u> ll	全C ウォッチポイントの削除	
	V <u>iew</u>	表示内容の変更	
	<u>R</u> adix	表示基数の変更	
<u>L</u> ayout	型名の表示/非表示		
<u>S</u> ort	アルファベット順並び替え		

## 1.9 ローカルウィンドウ

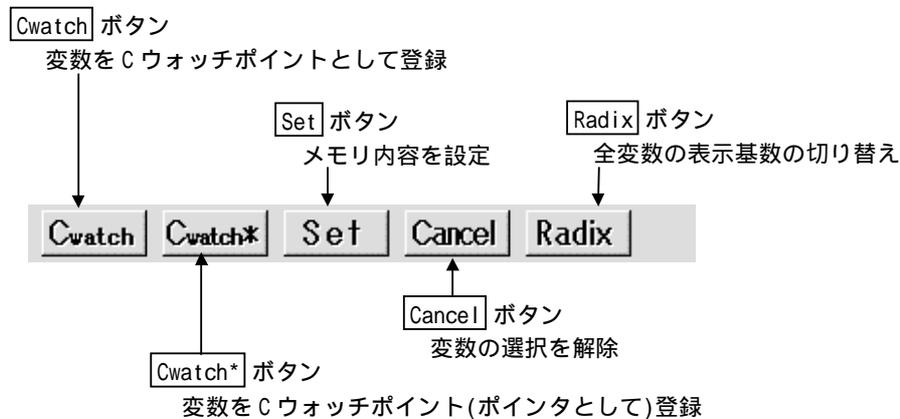
ローカルウィンドウは、C 言語の関数内ローカル変数の一覧とその値を表示するウィンドウです。ローカルウィンドウで表示される変数は、各コマンド実行後に更新されます。

### 1.9.1 ローカルウィンドウの画面構成



- ステップ実行等を行った際に、プログラムカウンタが含まれる関数に変更された場合は、それまで表示していた変数を削除し、新しい関数の関数内ローカル変数を自動登録します。

## 1.9.2 ローカルウィンドウのツールバー



- データ表示領域上でクリックして選択した変数に対して、C ウォッチポイントとして C ウォッチウィンドウに登録、及び値の設定が行えます。なお、値の設定には、C 言語式が使用できます。C 言語式に使用できる字句(トークン)については、本マニュアル リファレンス編の項目「3.1 C 言語式の記述方法」をご参照下さい。
- ポインタなどのアドレス表示は、表示基数にかかわらず 16 進表示で固定です。なお、表示形式の詳細については、本マニュアル リファレンス編の項目「3.2 C 言語式の表示形式」をご参照下さい。

## 1.9.3 ローカルウィンドウの拡張メニュー

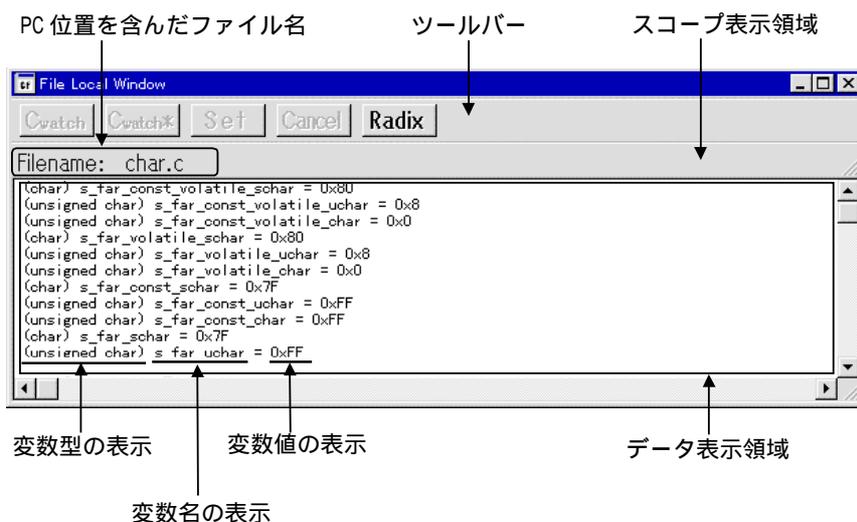
PD32000 のメイン表示領域に表示されたウィンドウのうち、ローカルウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	W <u>atch</u>	C 変数に関する操作	
	C <u>w</u> atch	選択位置の C 変数を C ウォッチポイントとして登録	
	C <u>w</u> atch <u>P</u> ointer	選択位置の C 変数のポインタを C ウォッチポイントとして登録	
	<u>S</u> et...	選択位置の C 変数の値を設定	
	<u>C</u> ancel	C 変数の選択を解除	
	V <u>iew</u>	表示内容の変更	
	<u>R</u> adix	表示基数の変更	
	<u>L</u> ayout	型名の表示/非表示	
	<u>S</u> ort	アルファベット順に並べ替え	

## 1.10 ファイルローカルウィンドウ

ファイルローカルウィンドウは、C 言語のファイル内ローカル変数の一覧とその値を表示するウィンドウです。ファイルローカルウィンドウで表示される変数は、各コマンド実行後に更新されます。

### 1.10.1 ファイルローカルウィンドウの画面構成



- ステップ実行等を行った際に、プログラムカウンタが含まれるファイルが変更された場合は、それまで表示していた変数を削除し、新しいソースファイルのファイル内ローカル変数を自動登録します。

### 1.10.2 ファイルローカルウィンドウのツールバー

ファイルローカルウィンドウのツールバーは、ローカルウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.9.2 ローカルウィンドウのツールバー」をご参照下さい。

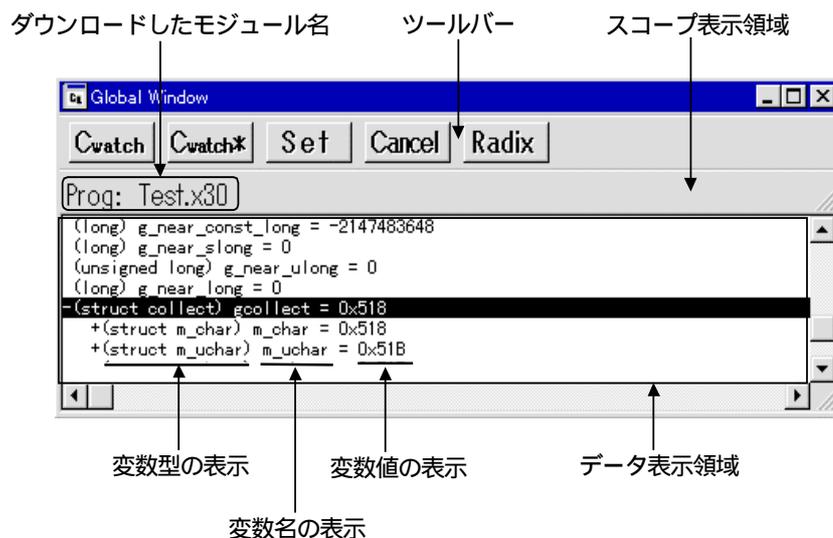
### 1.10.3 ファイルローカルウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、ファイルローカルウィンドウがアクティブな場合は、[ Option ]メニューにはファイルローカルウィンドウの拡張メニューが割り当てられます。ファイルローカルウィンドウの拡張メニューは、ローカルウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.9.3 ローカルウィンドウの拡張メニュー」をご参照下さい。

## 1.11 グローバルウィンドウ

グローバルウィンドウは、C 言語のグローバル変数の一覧とその値を表示するウィンドウです。グローバルウィンドウで表示される変数は、各コマンド実行後に更新されます。

### 1.11.1 グローバルウィンドウの画面構成



### 1.11.2 グローバルウィンドウのツールバー

グローバルウィンドウのツールバーは、ローカルウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.9.2 ローカルウィンドウのツールバー」をご参照下さい。

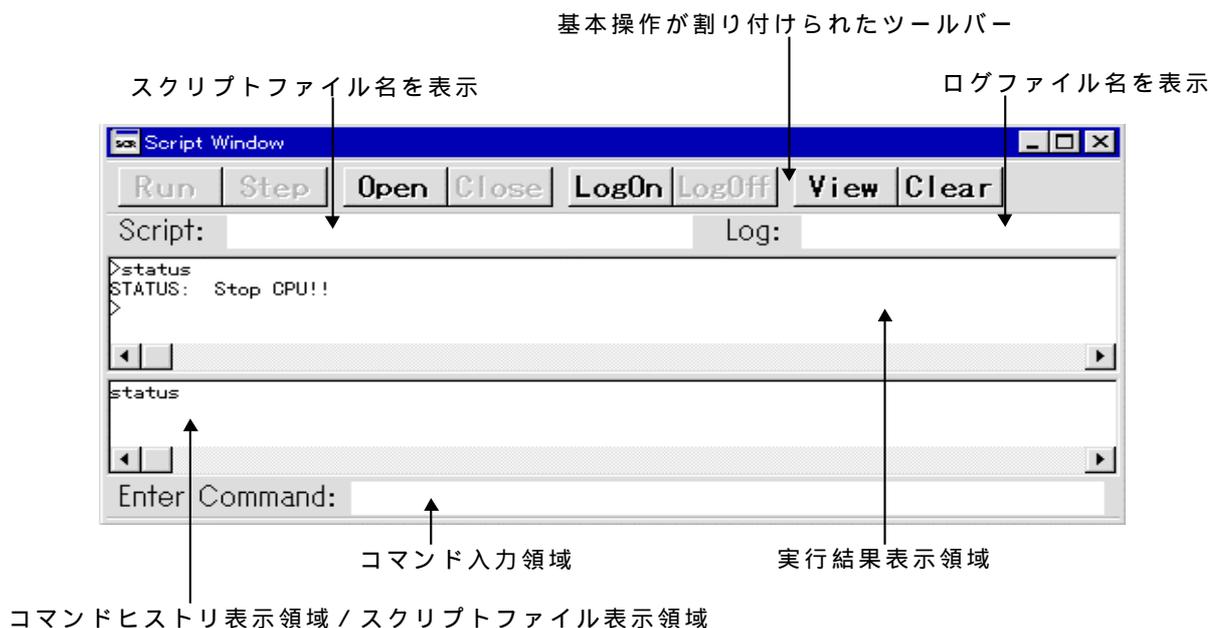
### 1.11.3 グローバルウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、グローバルウィンドウがアクティブな場合は、[ Option ]メニューにはグローバルウィンドウの拡張メニューが割り当てられます。グローバルウィンドウの拡張メニューは、ローカルウィンドウと同じです。詳細な説明については、本マニュアル ウィンドウ機能編の項目「1.9.3 ローカルウィンドウの拡張メニュー」をご参照下さい。

## 1.12 スクリプトウィンドウ

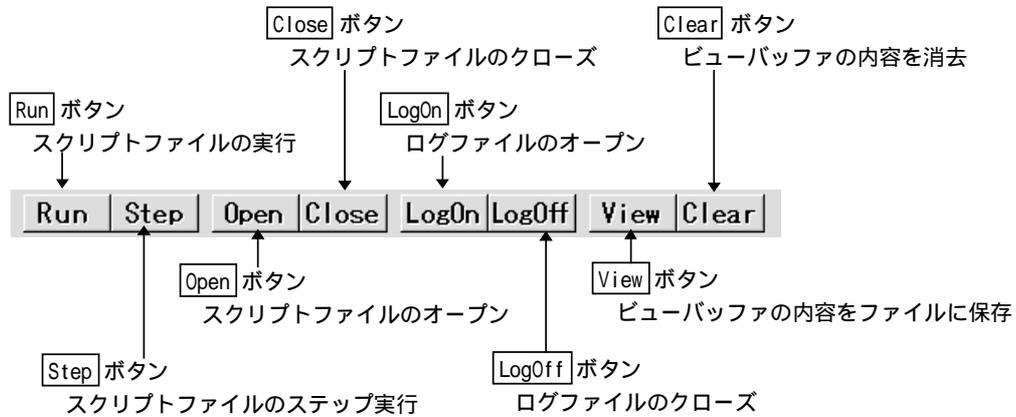
スクリプトウィンドウは、テキスト形式のスクリプトコマンドの実行、及び実行結果を表示するウィンドウです。スクリプトコマンドは、スクリプトファイルまたは対話入力によって実行できます。スクリプトコマンドをあらかじめスクリプトファイルに記述しておくことにより、スクリプトコマンドを自動実行できます。また、スクリプトコマンドの実行結果は、あらかじめ指定したファイル（ログファイル）に保存することができます。

### 1.12.1 スクリプトウィンドウの画面構成



- スクリプトウィンドウは、最新 1000 行分の実行結果を保存するビューバッファを持っており、ログファイルを指定していなくても実行結果をファイル（ビューファイル）に保存することができます。
- スクリプトファイルをオープンすると、コマンド履歴領域はスクリプトファイル表示領域に切り替わり、スクリプトファイルの内容を表示します。スクリプトファイルをネストオープンしている場合は、最後にオープンしたスクリプトファイルの内容を表示します。また、スクリプトファイル表示領域では、現在実行しているスクリプトファイル行を反転表示します。
- スクリプトファイルオープン時も、スクリプトファイルの実行が停止している時のみ、コマンド入力領域からスクリプトコマンドを発行することができます。

### 1.12.2 スクリプトウィンドウのツールバー



### 1.12.3 スクリプトウィンドウの拡張メニュー

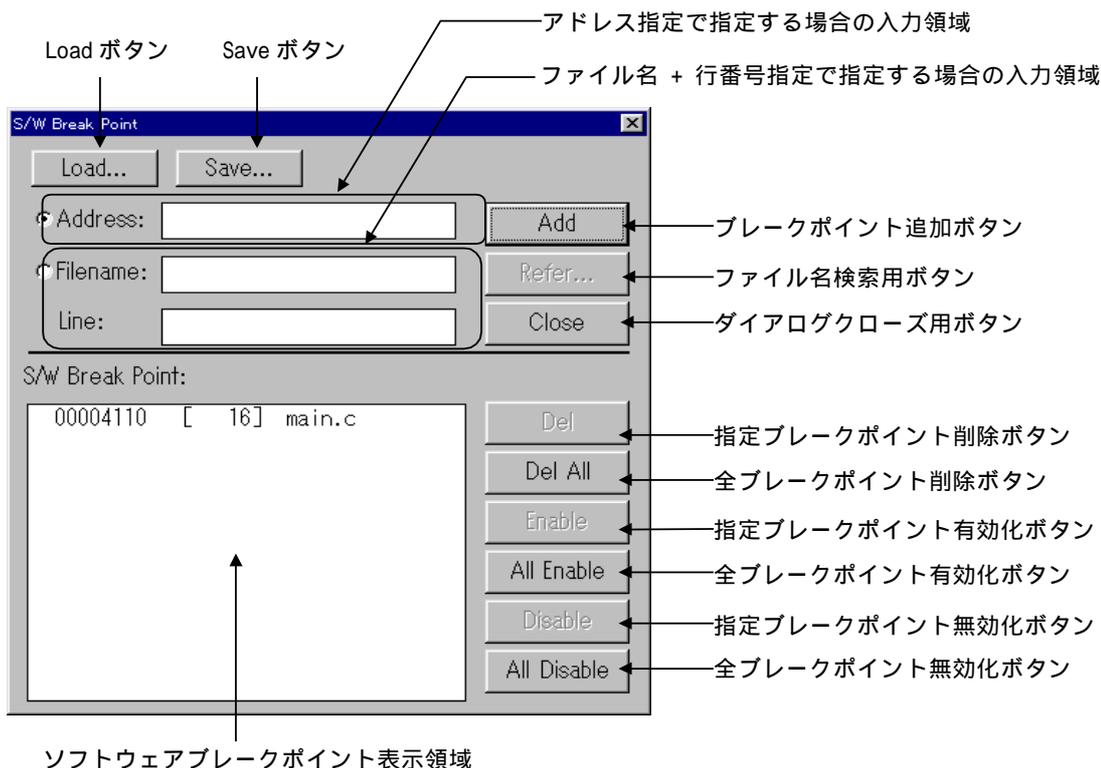
PD32000のメイン表示領域に表示されたウィンドウのうち、スクリプトウィンドウがアクティブな場合は、[ Option ]メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font....	フォントの変更	
	Script	スクリプトファイルの操作	
	Open...	スクリプトファイルのオープン	
	Run	スクリプトファイルの実行	
	Stop	スクリプトファイルの実行停止	
	Step	スクリプトファイルのステップ実行	
	Close	スクリプトファイルのクローズ	
	View	ビューバッファの操作	
	Save...	ビューバッファのファイル保存	
	Clear	ビューバッファのクリア	
	Log	ログファイルの操作	
	On...	ログファイルのオープン (出力開始)	
	Off	ログファイルのクローズ (出力終了)	

## 1.13 ブレークポイント設定ダイアログ

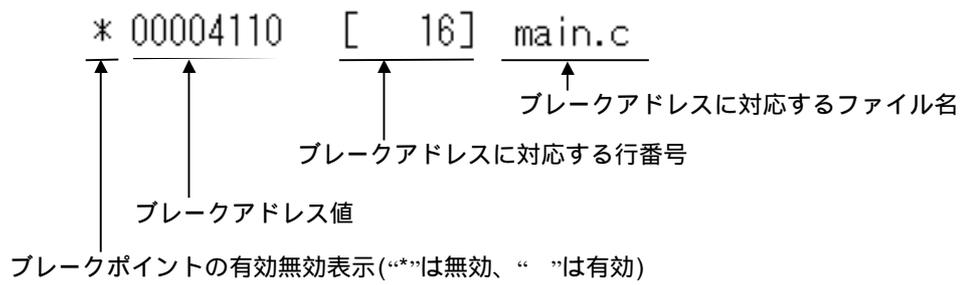
ブレークポイント設定ダイアログは、ソフトウェアブレークポイントを設定するためのダイアログです。ソフトウェアブレークは、設定したブレークポイントの命令を実行する手前でブレークします。それぞれのブレークポイントには、一時的に有効化/無効化を設定することができます。

### 1.13.1 ブレークポイント設定ダイアログの画面構成



- PD32000では、48点のソフトウェアブレークポイントが設定できます。
- ソフトウェアブレークポイントを複数設定した場合、組み合わせはOR条件になります。つまり、いずれか1点のソフトウェアブレークアドレスに到達するとプログラム実行を中止します。
- ソフトウェアブレークポイントの設定は、Close ボタンをクリックしてブレーク設定ダイアログをクローズするまで、連続して指定することができます。
- ソフトウェアブレークポイント表示領域上でクリックして選択したソフトウェアブレークポイントに対して、削除、有効化/無効化が行えます。また、有効化/無効化については、ソフトウェアブレークポイントをダブルクリックすることで、変更することができます。
- Save ボタンをクリックしてソフトウェアブレークポイントをファイルに保存できます。保存したファイルからソフトウェアブレークポイントの設定を読み込む場合は、Load ボタンをクリックします。ソフトウェアブレークポイントを読み込んだ場合、現在設定されているソフトウェアブレークポイントに追加してソフトウェアブレークポイントを設定します。
- 時間計測ポイントアドレスにブレークポイントを指定した場合、プログラムがブレークしません（エミュレータの仕様です）。

### 1.13.2 ソフトウェアブレークポイント一覧の記述

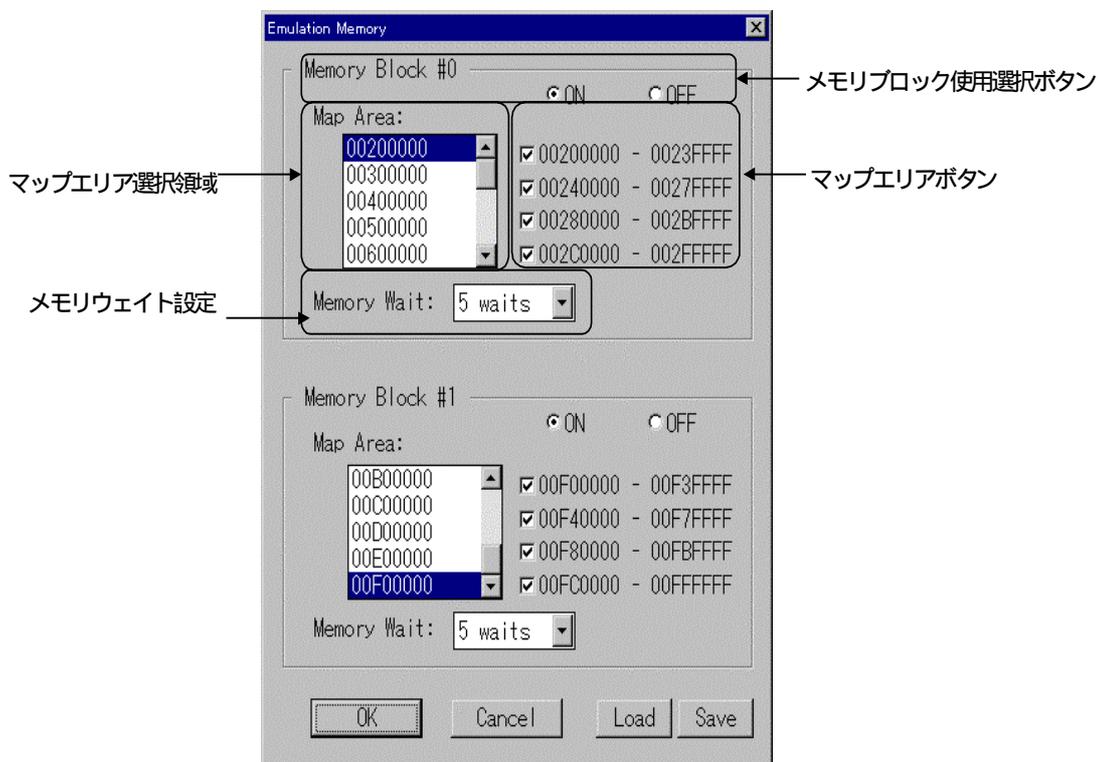


- アドレスに相当するソース行頭がない場合は、アドレス値のみ表示します。

## 1.14 エミュレーションメモリダイアログ

エミュレーションメモリダイアログは、エミュレータM32000Tのエミュレーションメモリの張り付けを行うためのダイアログです。

### 1.14.1 エミュレーションメモリダイアログの画面構成



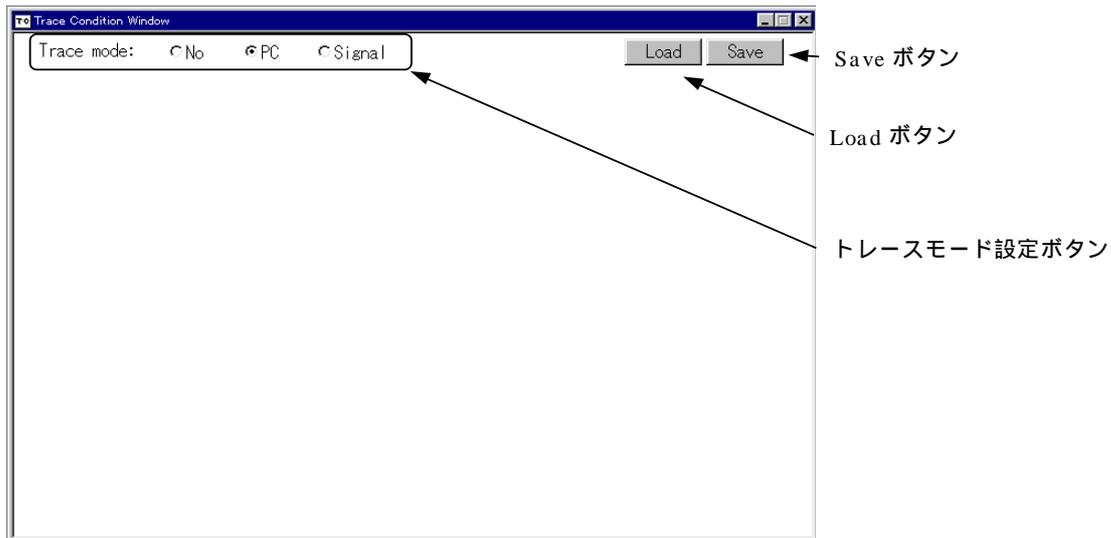
- M32000Tは1Mバイトのエミュレーションメモリを2つ持っています。
- メモリブロック使用選択ボタンでは、各メモリブロックの使用可否を選択できます。
- マップエリア選択領域では、1Mバイト単位でベースアドレスを設定できます。
- マップエリアボタンでは、256Kバイト単位でマッピングの可否を選択できます。
- メモリウェイト設定領域では、メモリのウェイト数を選択できます。選択できるウェイト数は2, 3, 4, 5です。

## 1.15 トレースコンディションウィンドウ

トレースコンディションウィンドウは、トレースモード、およびトレースイベントを設定するためのウィンドウです。トレースモードとしては、トレースなし、PC トレース、SIGNAL トレースの 3 種類から選択できます。

### 1.15.1 PC トレースモード時のトレースコンディションウィンドウ

以下に PC トレースモード時のトレースコンディションウィンドウの画面構成を示します。



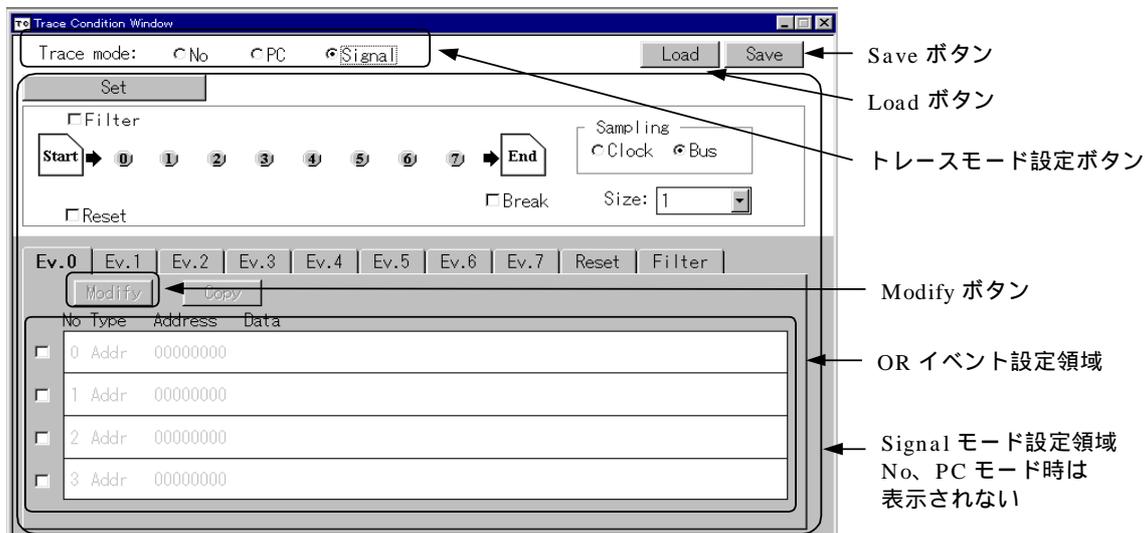
- トレースモードを PC トレースに設定した場合、その他の設定は不要です。

(注意)

PC トレースモード時のトレース結果は、逆アセンブルデータ形式で表示します。このデータは、リアルタイムな記録ではありません。

### 1.15.2 signal トレースモードのトレースコンディションウィンドウ

以下に signal トレースモード時のトレースコンディションウィンドウの画面構成を示します。



- Save ボタンをクリックしてトレースポイントの設定をファイルに保存できます。保存したファイルからトレースポイントの設定を読み込む場合は、Load ボタンをクリックします。設定を読み込んだ場合、現在の設定は削除されます。

#### 1. トレースポイントの組み合わせ条件

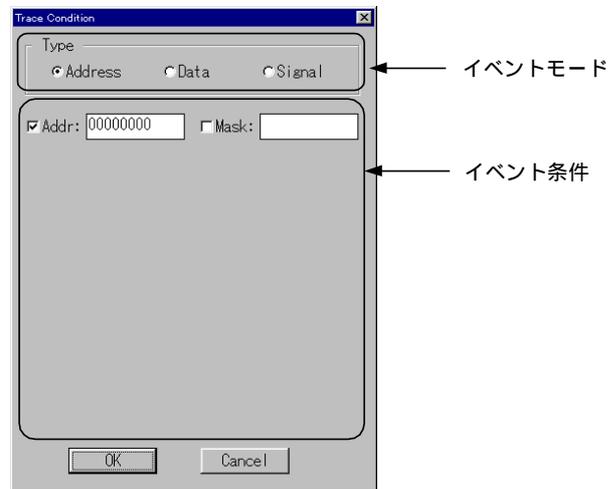
トレースコンディションウィンドウでは、トレースモードが signal の場合、トレースポイントの組み合わせ設定ができます。組み合わせは、1 ステートが OR 条件で最大 4 件まで、ステートの組み合わせがシーケンシャル（順序）で最大 8 ステートまで設定できます。

(注意)

signal トレースモード時のトレース結果は、MPU のバス形式で表示します。このデータは、リアルタイムな記録です。なお M32R/D の内部 DRAM の動作は記録できません。

## 2. トレースイベントの設定

OR イベント設定領域でダブルクリックまたは Modify ボタンをクリックすることにより、トレースイベント設定ダイアログがオープンします。このダイアログでトレースイベントが設定できます。

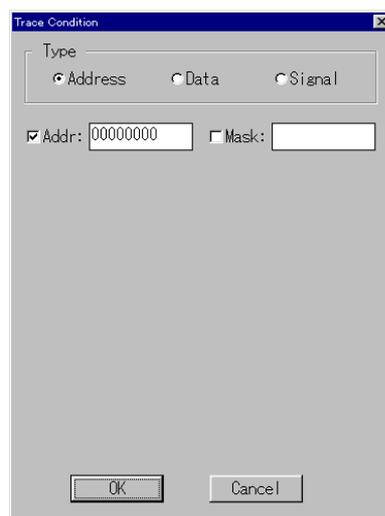


## 3. トレースイベント設定の詳細

トレースイベントは、Address モード、Data モード、Signal モードの 3 種類が設定できます。各モードの設定方法を以下に示します。

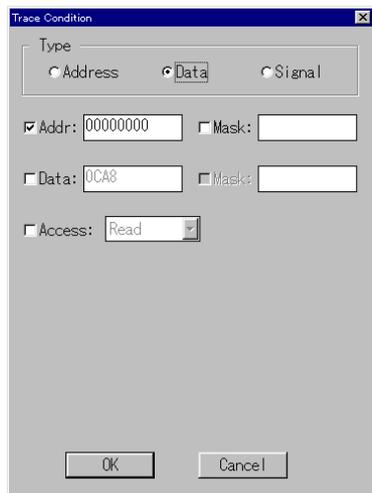
### - Address モード

イベントとしてアドレスを入力するモードです。イベント設定ダイアログの Address を選択することにより表示されます。(デフォルト)。

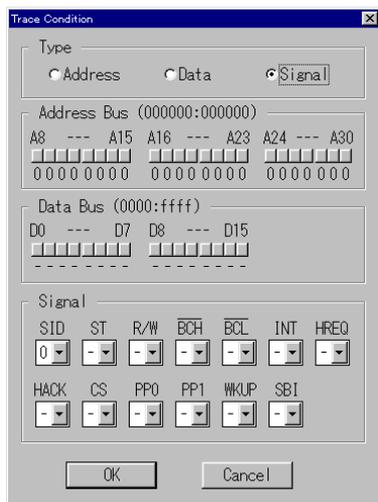


- Address モードでは、イベント対象となるアドレス、およびアドレスマスクが設定できます。

- Data モード  
 イベントとしてデータアクセスを入力するモードです。イベント設定ダイアログの Data を選択することにより表示されます。  
 Data モードでは、アドレス、アドレスマスク、データ、データマスク、およびアクセス条件が設定できます。アクセス条件を Write に設定した場合、書き込みサイズ (8bit、16bit) が選択できます。



- Signal モード  
 イベントとして MPU の信号状態を入力するモードです。イベント設定ダイアログの Signal を選択することにより表示されます。  
 Signal モードでは、MPU の信号線単位で設定できます。アドレスおよびデータのビット情報は、ビット単位で割り付けているボタンを押すことで '0','1','-' (マスク) が設定できます。また、他の信号線情報は、 ボタンを押すことで '0','1','-' (マスク) が設定できます。



### 1.15.3 トレースコンディションウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、トレースコンディションウィンドウがアクティブな場合は、[ Option ]メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Set	設定条件のエミュレータへの登録	
	L <u>o</u> ad...	設定条件をファイルから読み込む	
	S <u>a</u> ve...	設定条件をファイルに保存	

## 1.16 トレースウィンドウ

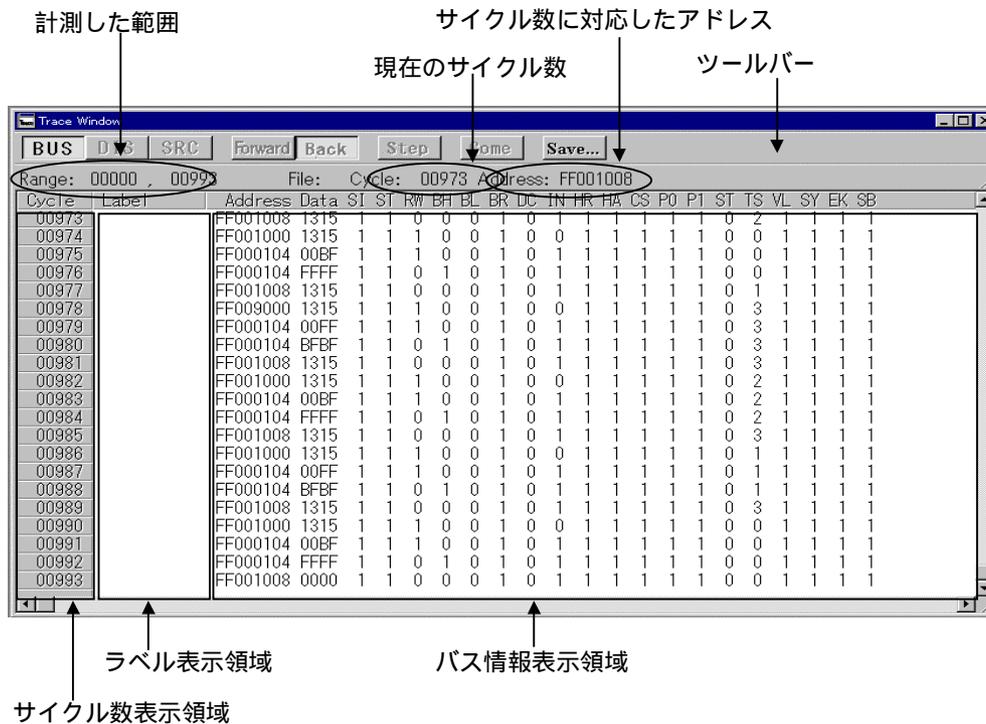
トレースウィンドウは、エミュレータ**M32000T**に搭載されているトレース機能（実行履歴、信号状態の記録）の計測結果を表示するウィンドウです。トレースウィンドウでは、以下の3つの表示モードをサポートしています。各表示モードの切り替えは、トレース結果表示ウィンドウのツールバーに割り付けられているボタンによって行います。

- バスモード(signal トレース時)  
サイクル毎のバス情報が参照できます。  
バスモードでは、トレース計測時の実行経路順に内容を表示します。そのため、トレース結果は、垂直スクロールバーのスクロールによって参照できます。  
なおバスモードは、トレースモードが signal トレースの時に表示可能です。
- 逆アセンブルモード(PC トレース時)  
命令毎のサイクル数が参照できます。実行した命令を逆アセンブル表示します。  
逆アセンブルモードでは、トレース計測時の実行経路順に内容を表示します。そのため、トレース結果は、垂直スクロールバーのスクロールによって参照できます。  
なお逆アセンブルモードは、トレースモードが PC トレースの時に表示可能です。
- ソースモード(PC トレース時)  
ソース行毎のサイクル数が参照できます。  
ソースモードでは、実行したソースファイルの内容をそのまま表示します（トレース計測時の実行経路順の表示ではありません）。そのため、トレース結果は、ツールバーに備えられたボタン操作によって参照します。  
なおソースモードは、トレースモードが PC トレースの時に表示可能です。

トレースウィンドウは、トレース計測が終了した時点で、計測結果を表示します。起動直後およびターゲット実行直後は、未だトレース計測が完了していませんので、空白表示になります。

トレース計測の開始/終了のタイミングを指定する機能として、トレースポイントが使用できます。トレースポイントは、トレースコンディションウィンドウ、およびスクリプトコマンドの TracePoint コマンドで設定できます。

### 1.16.1 バスモード時の画面構成



#### ツールバー

バスモードでは、トレース計測結果の保存を行うボタン（ Save... ボタン）が有効となります。なお、トレース計測結果が存在しない場合、トレース計測結果をファイルに保存するボタンは無効となります。

#### サイクル数表示領域

各行のサイクル数を表示します。

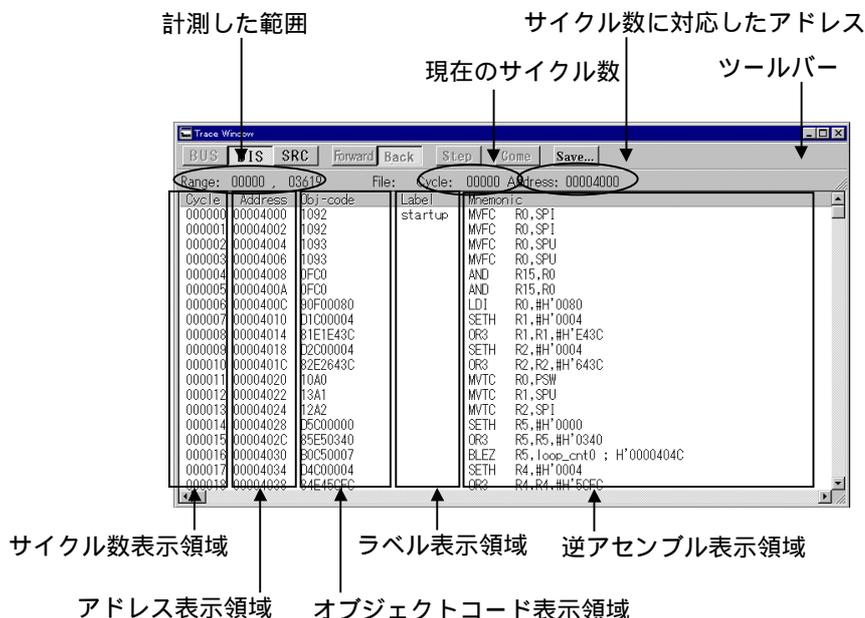
- トレースポイント(トレース計測の開始 / 終了の起点となるサイクル)を 0 サイクルとして、+10 進数で表示します。
- サイクル数表示領域をダブルクリックすることで、表示する開始サイクル数を変更することができます。

## バス情報表示領域

各サイクルのバス情報を表示しています。左端より以下の内容を意味します。

- Address  
アドレスバスの状態を示します。
- Data  
データバスの状態を示します。
- SI  
SID 信号の状態を示します。
- ST  
ST 信号の状態を示します。
- RW  
RW 信号の状態を示します。
- BH  
BCH 信号の状態を示します。
- BL  
BCL 信号の状態を示します。
- BU  
BURST 信号の状態を示します。
- DC  
DC 信号の状態を示します。
- IN  
INT 信号の状態を示します。
- HR  
HREQ 信号の状態を示します。
- HA  
HACK 信号の状態を示します。
- CS  
CS 信号の状態を示します。
- P0  
PP0 信号の状態を示します。
- P1  
PP1 信号の状態を示します。
- SY  
STBY 信号の状態を示します。
- WK  
WKUP 信号の状態を示します。
- SB  
SBI 信号の状態を示します。

## 1.16.2 逆アセンブルモード時の画面構成



### ツールバー

逆アセンブルモードでは、表示モードの変更を行うボタン ( DIS SRC ボタン)、トレース計測結果の保存を行うボタン ( Save... ボタン) が有効となります。ただし、デバッグ情報がダウンロードされていない場合、表示モードの変更を行うボタンのうち、ソースモードへの変更ボタンは無効となります。またトレース計測結果が存在しない場合、トレース計測結果をファイルに保存するボタンは無効となります。

### サイクル数表示領域

各行のサイクル数を表示します。

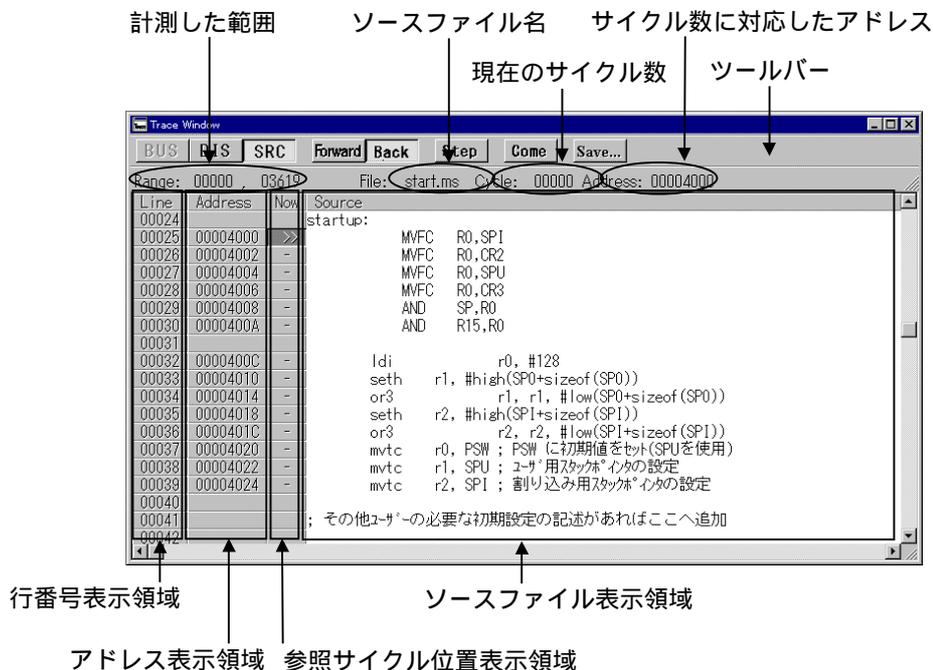
- トレースポイント(トレース計測の開始/終了の起点となるサイクル)を 0 サイクルとして、+10 進数で表示します。
- サイクル数表示領域をダブルクリックすることで、開始サイクル数を変更することができます。

### アドレス表示領域

命令の先頭アドレスを、16 進数で表示します。

- アドレス表示領域は、メニュー[Option] [Layout] [Address Area]の選択/解除によって、表示/非表示にすることができます。
- アドレス表示領域をダブルクリックすることで、アドレス指定によるサイクル検索を行うことができます。

### 1.16.3 ソースモード時の画面構成



#### ツールバー

ソースモードでは、ツールバーの「Come」ボタンを除く全てのボタンが有効になります。

「Come」ボタンは、Come 検索可能な行をクリックしたときに有効になります。

#### 行番号表示領域

ソースファイルの行番号を表示します。

- 行番号表示領域は、メニュー[Option] [Layout] [Line Area]の選択 / 解除によって、表示 / 非表示にすることができます。
- 行番号表示領域をダブルクリックすることで、表示するソースファイルを変更することができます。

#### アドレス表示領域

ソースファイル表示領域の各行のアドレスを、16 進数で表示します。

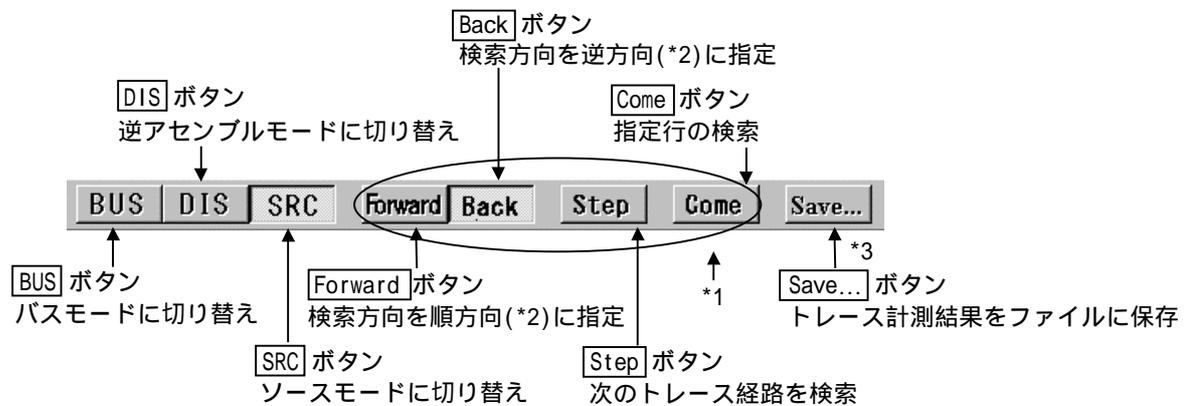
- アドレス表示領域は、メニュー[Option] [Layout] [Address Area]の選択 / 解除によって、表示 / 非表示にすることができます。なお、アドレス表示領域は、デフォルトでは非表示になっています。
- アドレス表示領域をダブルクリックすることで、アドレス指定によるサイクル検索を行うことができます。

#### 参照サイクル位置表示領域

トレース情報からアドレスを検索する際の現在の位置を示します。

- “>>” で示します。検索を続ける度に、この記号が移動します。またハイフン（‘-’）は、アドレス情報付きの行（Come 検索可能な行）を示します。

### 1.16.4 トレースウィンドウのツールバー



\*1 Forward, Back, Step, Come ボタンは、表示モードがソースモードのときのみ有効です。

\*2 順方向検索は、命令の実行順序で経路をトレースします。逆方向検索は、命令の実行順序を遡って経路をトレースします。

\*3 Save... ボタンは、トレース計測結果が存在するときのみ有効です。

### 1.16.5 トレースウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、トレースウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	TAB...	ソースファイル表示のタブ設定	
	View	表示内容の変更	
	Cycle...	サイクル指定による変更	
	Address Search...	アドレス指定によるサイクル検索	
	Source...	ソースファイル指定による変更	
	Mode	表示モードの変更	
	Bus	バスモードへの変更	
	Disasm	逆アセンブルモードへの変更	
	Source	ソースモードへの変更	
	Layout	レイアウト設定	
Line Area	行番号表示領域の表示/非表示		
Address Area	アドレス表示領域の表示/非表示		
Code Area	オブジェクトコード表示領域の表示/非表示		
Trace	トレース計測結果の検索		
Forward	検索方向(実行経路の順方向)		
Backward	検索方向(実行経路の逆方向)		
Step	1回の検索 (Step 検索)		
Come	指定行の検索 (Come 検索)		
Save...	トレース計測結果をファイルに保存		

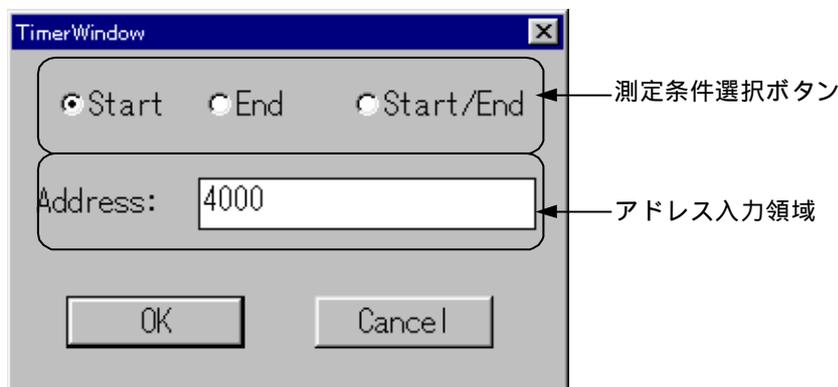
## 1.17 区間時間測定ウィンドウ

区間時間測定ウィンドウは、エミュレータ**M32000T**の区間時間測定機能の設定、および設定内容の参照を行なうウィンドウです。区間時間測定は指定された区間の最大、最小、平均実行時間、測定回数を測定する機能です。エミュレータ**M32000T**では、同時に最大 20 点の区間時間を測定することができます。

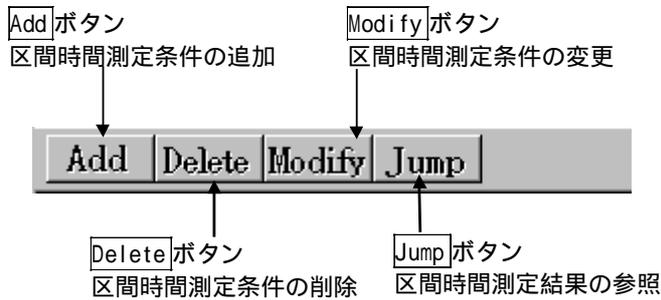
### 1.17.1 区間時間測定ウィンドウの画面構成



- 区間時間測定条件は Add ボタンまたは Modify ボタンをクリックすると、以下のダイアログがオープンしますので、ここで入力してください。



### 1.17.2 区間時間測定ウィンドウのツールバー



### 1.17.3 区間時間測定ウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、区間時間測定ウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの設定	
	Add	区間時間測定条件の設定	
	Delete	区間時間測定条件の削除	
	Modify	区間時間測定条件の変更	
	Jump	区間測定条件の結果参照	

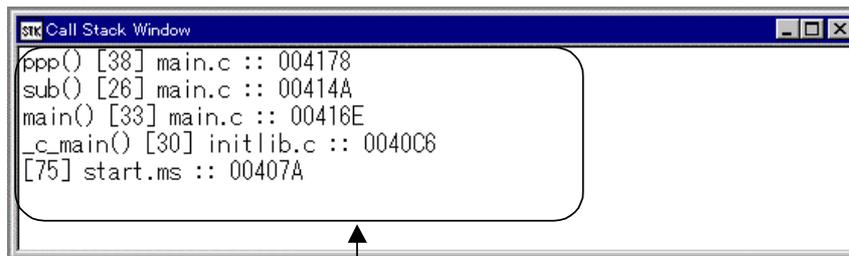
**(注意)**

- Start アドレスと End アドレスは同一の for, while などのツール中に設定しないで下さい。また StartEnd アドレスも同様です。
- 時間計測ポイントアドレスにブレークポイントを指定した場合、プログラムがブレークしません (エミュレータの仕様です)。

## 1.18 コールスタックウィンドウ

コールスタックウィンドウは、プログラムの関数呼び出し状況を参照するためのウィンドウです。

### 1.18.1 コールスタックウィンドウの画面構成



関数呼び出し状況

### 1.18.2 コールスタックウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、コールスタックウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの設定	
	Jump	指定した関数に対応するソースをPRG ウィンドウへ表示	
	New window	指定した関数に対応するソースを新SRC ウィンドウへ表示	

## 1.19 MR ウィンドウ

MR ウィンドウは、リアルタイム OS の状態を表示するウィンドウです。リアルタイム OS を使用したプログラムをダウンロードした場合にのみ使用することができます(ダウンロードしたプログラムが MR を使用していなかった場合、MR ウィンドウをオープンしても MR ウィンドウには何も表示されません)。

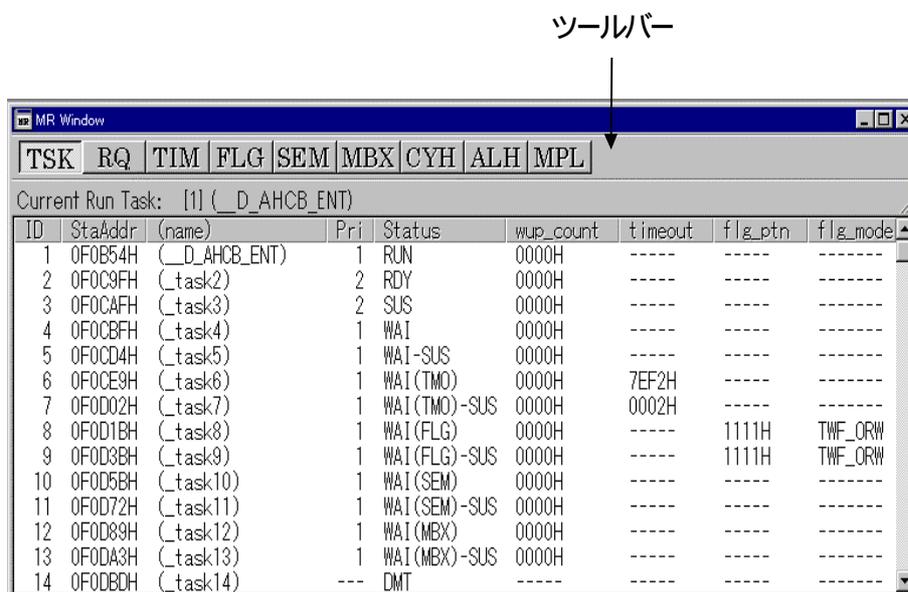
また MR ウィンドウは、表示モードの種類数分(現状 9 枚)までオープンすることができます。

### (注意)

- ダウンロードしたプログラムが MR を使用していなかった場合、表示モードを選択するメニューは全て選択できなくなります。

### 1.19.1 MR ウィンドウの画面構成

以下に MR ウィンドウの画面構成を示します。



### 1.19.2 MR ウィンドウのツールバー



各ボタンをクリックしたときの MR ウィンドウの表示内容を以下に示します。

ボタン名	表示内容	ボタン名	表示内容
TSK	タスクの状態を表示	MBX	メールボックスの状態を表示
RQ	レディキューの状態を表示	CYH	周期起動ハンドラの状態を表示
TIM	タイムアウトキューの状態を表示	ALH	アラームハンドラの状態を表示
FLG	イベントフラグの状態を表示	MPL	メモリプールの状態を表示
SEM	セマフォの状態を表示		

### 1.19.3 MR ウィンドウの拡張メニュー

PD32000のメイン表示領域に表示されたウィンドウのうち、MR ウィンドウがアクティブな場合は、[ Option ] メニューには以下のメニューが割り当てられます。

メニュー	メニュー項目	機能	ショートカットキー
Option	Font...	フォントの変更	
	Mode	表示モードの切り替え	
	Task	タスクの状態を表示	
	Ready Q	レディキューの状態を表示	
	Timeout Q	タイムアウトキューの状態を表示	
	Event Flag	イベントフラグの状態を表示	
	Semaphore	セマフォの状態を表示	
	Mailbox	メールボックスの状態を表示	
	Cyclic Handler	周期性ハンドラの状態を表示	
	Alarm Handler	アラームハンドラの状態を表示	
	Memory Pool	メモリプールの状態を表示	
	MR		
	Context...	コンテキストの表示	
	Layout	レイアウトの設定	
	Status Bar	MR ウィンドウのステータスバーの表示非表示切り替え	

【 MEMO 】

# 基本操作方法編

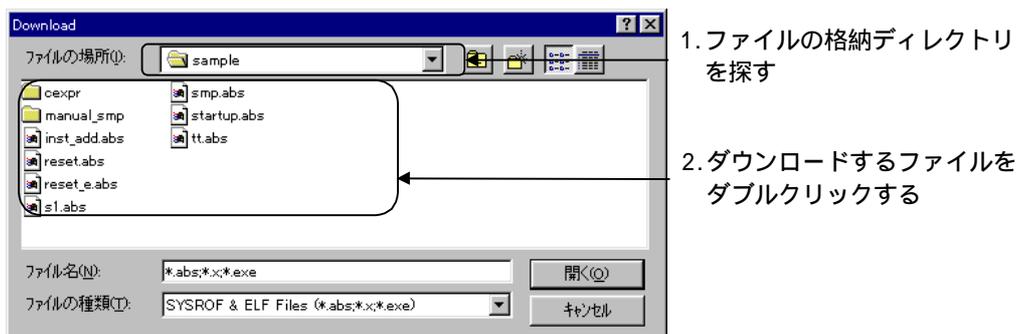


# 1 ターゲットプログラムの読み込み・表示

## 1.1 ダウンロードするには

ターゲットプログラムをダウンロードするには、**PD32000**ウィンドウのメニュー [ File ] [ Download ] [ Load module... ] を選択してください。ファイルセレクションダイアログがオープンします。ファイルセレクションダイアログからターゲットプログラムを選択してください。ダウンロードの対象ファイルは、アブソリュート形式ファイル（以下、Xファイルと記述）です。Xファイルのファイル属性は、“.x”、“.abs”、または“.exe”です。

ダウンロード用ファイルセレクションダイアログは、Shift + F1 キー入力でもオープンします。



### 注意事項

Xファイルをダウンロードしてもプログラム・ソースウィンドウがソース表示モードに切り替わらない場合は、Xファイル中にソース行情報が存在しない可能性があります。コンパイラ・アセンブル・リンク時のオプションをご確認ください。

### 機械語情報のみダウンロードするには・・・

機械語情報のみをダウンロードするには、**PD32000**ウィンドウのメニュー [ File ] [ Download ] [ Memory Image... ] を選択してください。ファイルセレクションダイアログがオープンします。ファイルセレクションダイアログからモトローラSフォーマットファイルを選択し、機械語情報のダウンロードを行ってください。

### バイナリデータをダウンロードするには・・・

バイナリデータをダウンロードするには、**PD32000**ウィンドウのメニュー [ File ] [ Download ] [ Binary... ] を選択してください。ダイアログにダウンロード開始アドレス、およびバイナリファイルを選択し、バイナリデータのダウンロードを行ってください。

### シンボル情報のみダウンロードするには・・・

シンボル情報のみをダウンロードするには、**PD32000**ウィンドウのメニュー  
[ File ] [ Download ] [ Symbol... ]  
を選択してください。ファイルセレクションダイアログから X ファイルを選択し、シンボル  
情報のダウンロードを行ってください。シンボル情報のみのダウンロードでは、X ファイル  
からシンボル情報だけを読み込みます。

### 再ダウンロードするには・・・

PD32000では、前回ダウンロードしたファイルを再度ダウンロードする際に、ファイル名  
を指定せずにダウンロードすることができます。再ダウンロードする際には、**PD32000**ウィ  
ンドウのメニュー

[ File ] [ Reload ]  
を選択してください。本メニューを選択するとダウンロードが実行されます。

#### 注意事項

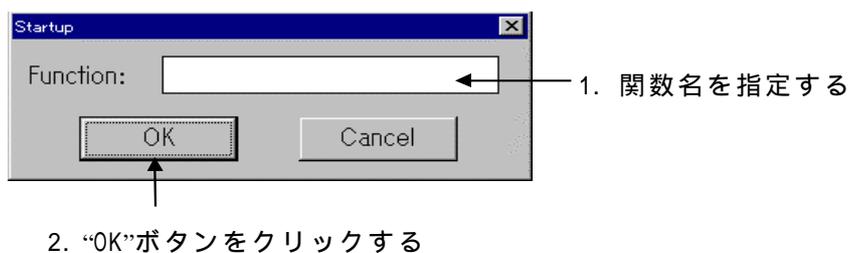
再ダウンロードした際に、SRC ウィンドウの表示モードがソース/MIX モードから逆アセ  
ンブルモードに変更できない場合は、基本操作方法編の項目「1.7 他ディレクトリに存在す  
るソースプログラムを参照するには」を参照し、サーチパスを設定してください。サーチパ  
スの設定は、D:¥xxx¥yyy のように絶対パスで指定してください。

## 1.2 ダウンロード直後のプログラム表示位置を変更するには

**PD32000**では、ターゲットプログラムをダウンロードするとターゲットリセット後のプログ  
ラムカウンタ位置のソースプログラムをプログラムウィンドウに表示します。このプログラムカ  
ウンタ位置にソース行情報が存在しない場合(スタートアッププログラムにソース行情報がない  
場合等)、逆アセンブル表示モードで表示されます。

ターゲットプログラムのダウンロード後、自動的に main 関数等のソースプログラムを表示し  
たい場合は、予め関数名を指定しておく必要があります。**PD32000**ウィンドウのメニュー

[ Environ ] [ StartUp... ]  
を選択してください。StartUp ダイアログがオープンします。StartUp ダイアログで表示する関  
数名を指定してください。



StartUp ダイアログで指定した関数がプログラムウィンドウに表示されてもプログラムカウ  
ンタ値はダウンロード直後のプログラムカウンタ値のままです。StartUp ダイアログで指定した  
関数までプログラムカウンタ値を進める場合は、カム実行を行ってください。カム実行につい  
ては、本マニュアル 基本操作方法編の項目「2.4 指定位置までプログラムを実行するには」を  
ご参照下さい。

#### 注意事項

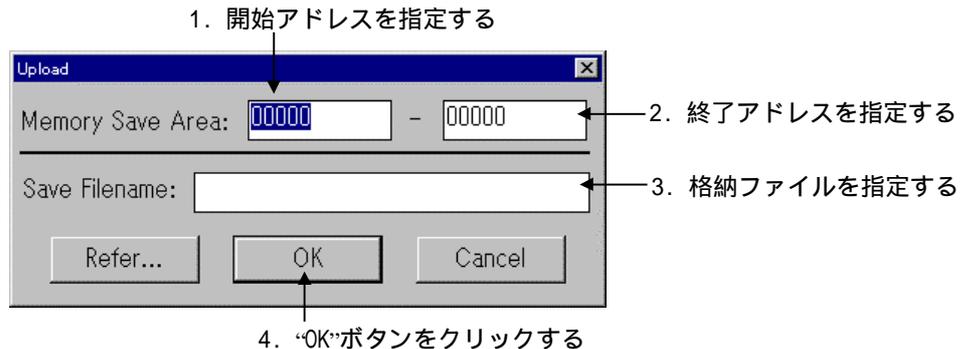
ダウンロード直後のプログラムカウンタ位置にソース行情報が存在する場合、この設定は無  
効になります。

## 1.3 アップロードするには

### PD32000ウィンドウのメニュー

[ File ] [ Upload... ]

を選択してください。アップロードダイアログがオープンします。アップロードする領域と保存するファイル名を入力してください。保存ファイルは、モトローラ S フォーマットファイルとバイナリファイルが指定できます。モトローラ S フォーマットファイル形式で保存する場合は、拡張子に“.mot”を指定してください。バイナリデータで保存する場合は、拡張子に任意の文字列が指定できます。なお、ファイル名に既存のファイル名を指定した場合は、上書きします。



### モトローラ S フォーマット形式でアップロードするには・・・

バイナリデータをアップロードするには、**PD32000**ウィンドウのメニュー

[ File ] [ Upload ] [ Memory Image... ]

を選択してください。ダイアログにアップロード開始/終了アドレス、およびモトローラ S フォーマット形式データを保存するファイル名を指定してください。

### バイナリデータをアップロードするには・・・

バイナリデータをアップロードするには、**PD32000**ウィンドウのメニュー

[ File ] [ Upload ] [ Binary... ]

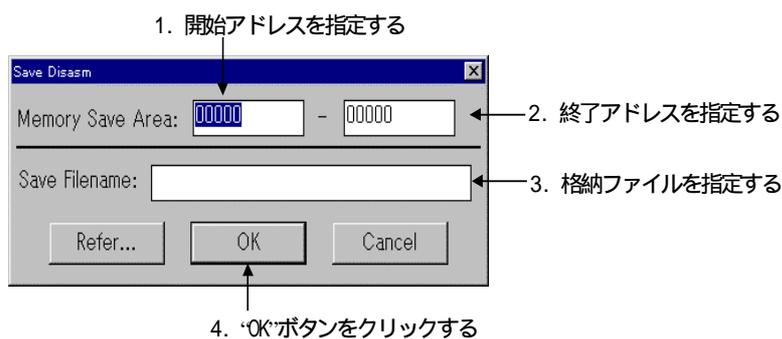
を選択してください。ダイアログにアップロード開始/終了アドレス、およびバイナリデータを格納するファイル名を指定してください。

## 1.4 逆アセンブル結果を保存するには

### PD32000ウィンドウのメニュー

[ File ] [ Save Disasm... ]

を選択してください。逆アセンブル結果保存ダイアログがオープンします。保存する領域とファイル名を入力してください。保存ファイル名には、任意のファイル名、ファイル属性が指定できます。なお、ファイル名に既存のファイル名を指定した場合は、上書きします。



## 1.5 プログラムの任意位置を常に表示するには

プログラムウィンドウは、プログラムカウンタ位置を追従して表示位置を切り替えますのでターゲットプログラムの任意位置を常に表示することはできません。ターゲットプログラムの任意位置を常に表示するには、ソースウィンドウをご使用ください。ソースウィンドウは、**PD32000** ウィンドウのメニュー

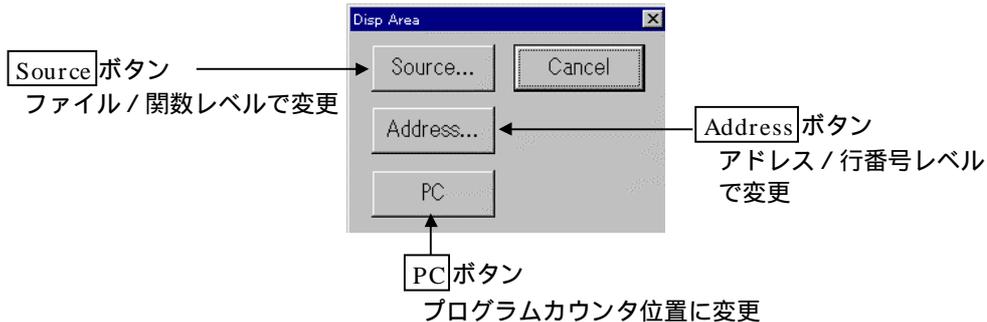
[ Window ] [ Source Window ]  
 の選択でオープンします。

## 1.6 プログラムの表示位置を変更するには

ソースプログラムの内容は、プログラムウィンドウ及びソースウィンドウで表示できます。プログラム（ソース）ウィンドウの表示位置を変更するには、プログラム（ソース）ウィンドウのツールバーから“View”ボタンをクリックしてください。表示位置の変更は、アクティブウィンドウのみ有効です。



“View”ボタンをクリックすると以下の Disp Area ダイアログがオープンします。

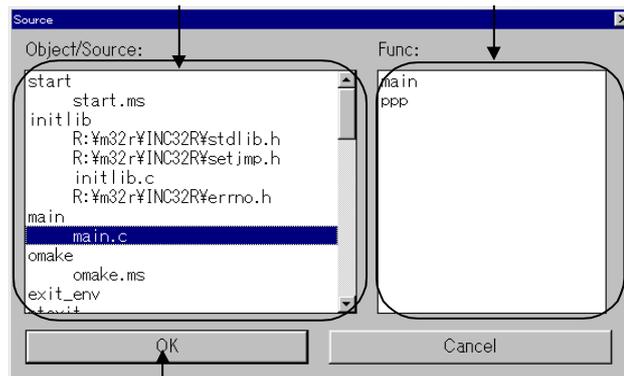


なお、デバッグ情報を読み込んでいない場合は、ファイル/関数レベル、及び行番号レベルでの変更は行えません。また、プログラム（ソース）ウィンドウの表示モードが逆アセンブル表示モードの場合、行番号レベルでの変更は行えません。

### ファイル/関数レベルで表示位置を変更するには・・・

Disp Area ダイアログから“Source”ボタンをクリックしてください（“Source”ボタンは、デバッグ情報を読み込んでいない場合はクリックできません）。以下の Source ダイアログがオープンします。Source ダイアログでは、読み込んだターゲットプログラムのファイル構成、及び関数情報を表示しています。変更するファイル及び関数名をクリックしてください。

1. ファイルを選択する
2. 関数を選択する



3. “OK”ボタンをクリックする

Source ダイアログは、PD32000ウィンドウのメニュー

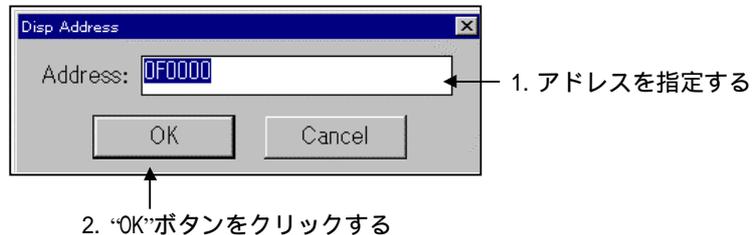
[ Option ] [ View ] [ Source... ]

の選択でもオープンします。

また、Source ダイアログは、プログラム（ソース）ウィンドウの行番号表示領域をダブルクリックすることでもオープンします。

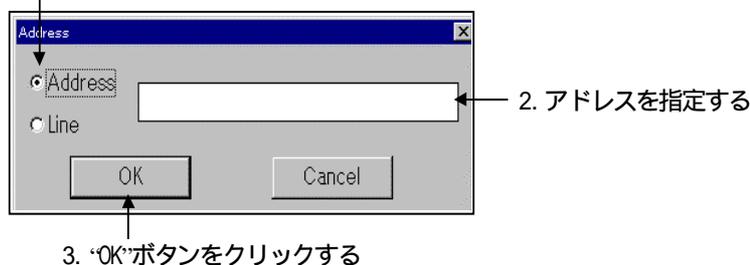
### アドレスレベルで表示位置を変更するには・・・

プログラム（ソース）ウィンドウの表示モードが逆アセンブル表示モードのとき、Disp Area ダイアログから“Address”ボタンをクリックしてください。“Address”ボタンをクリックすると以下の Disp Address ダイアログがオープンします。



なお、ソース表示モードまたは MIX 表示モードのときは、以下の Address ダイアログがオープンします。Address ダイアログの Address ラジオボタンをクリックして、変更アドレスを入力してください。

1. “Address”ラジオボタンをクリックをする



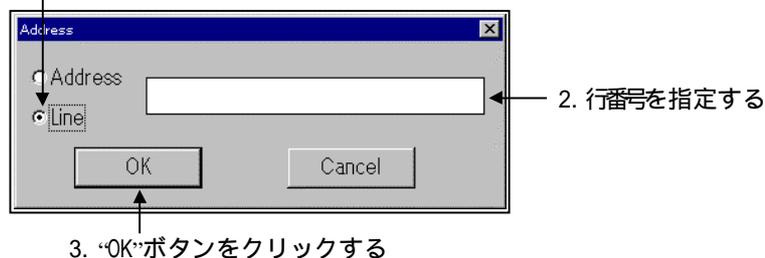
Disp Address ( Address ) ダイアログは、**PD32000**ウィンドウのメニュー  
[ Option ] [ View ] [ Address... ]  
の選択でもオープンします。

また、Disp Address ( Address ) ダイアログは、プログラム（ソース）ウィンドウのアドレス表示領域をダブルクリックすることでもオープンします。

### 行番号レベルで表示位置を変更するには・・・

プログラム（ソース）ウィンドウの表示モードがソース表示モードまたは MIX 表示モードのとき、Disp Area ダイアログから“Address”ボタンをクリックしてください。“Address”ボタンをクリックすると以下の Address ダイアログがオープンします。Address ダイアログの Line ラジオボタンをクリックして、変更する行番号を入力してください。

1. “Line”ラジオボタンをクリックをする



Address ダイアログは、**PD32000**ウィンドウのメニュー  
[ Option ] [ View ] [ Address... ]  
の選択でもオープンします。

また、Address ダイアログは、プログラム（ソース）ウィンドウのアドレス表示領域をダブルクリックすることでもオープンします。

プログラムカウンタ位置に変更するには・・・

Disp Area ダイアログから“PC”ボタンをクリックしてください。“PC”ボタンをクリックするとウィンドウの表示位置がプログラムカウンタ位置に変更されます。**PD32000**ウィンドウのメニュー

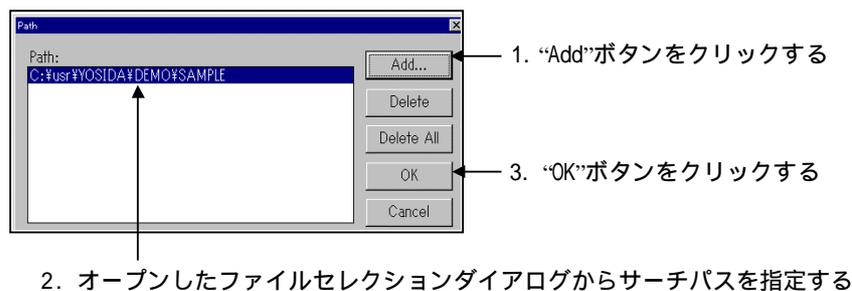
[ Option ] [ View ] [ Program Counter ]  
 の選択でも変更できます。

### 1.7 他ディレクトリに存在するソースプログラムを参照するには

ソースファイルのサーチパスを指定します。ターゲットプログラムのソースファイルがカレントディレクトリになく、複数のサブディレクトリに別れて格納されている場合に有効です。これにより、ソースプログラムの参照やウィンドウ上からのソフトウェアブレイクポイント設定等ができます。サーチパスを指定するには、**PD32000**ウィンドウのメニュー

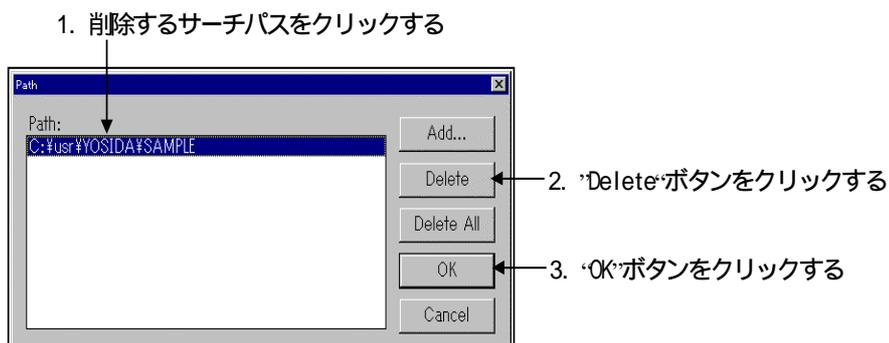
[ Environ ] [ Path... ]

を選択してください。Path ダイアログがオープンします。サーチパスを追加する場合は、Path ダイアログの“Add”ボタンをクリックしてください。ファイルセレクションダイアログがオープンしますので参照するファイル名を選択してください。



サーチパスを削除するには・・・

Path ダイアログから以下の操作を行ってください。



## 1.8 ソースと逆アセンブル結果を MIX 表示するには

ソースファイルとその逆アセンブル結果を混合して表示するには、プログラム(ソース)ウィンドウのツールバーから“MIX”ボタンをクリックしてください。MIX 表示モードの切り替えは、**PD32000**ウィンドウのメニュー

[ Option ] [ Mode ] [ MIX Mode ]

の選択でも行うことができます。表示モードの変更は、アクティブウィンドウのみ有効です。

ターゲットプログラム停止後のプログラムカウンタの停止位置が、ソース行情報が出力されている領域でかつソースの行頭アドレスに一致しない場合、プログラムウィンドウの表示モードが自動的に MIX 表示モードとなります。

MIX 表示をソースプログラム表示に戻すには・・・

プログラム(ソース)ウィンドウのツールバーから“SRC”ボタンをクリックしてください。**PD32000**ウィンドウのメニュー

[ Option ] [ Mode ] [ Source Mode ]

の指定でもソースプログラム表示となります。

### 注意事項

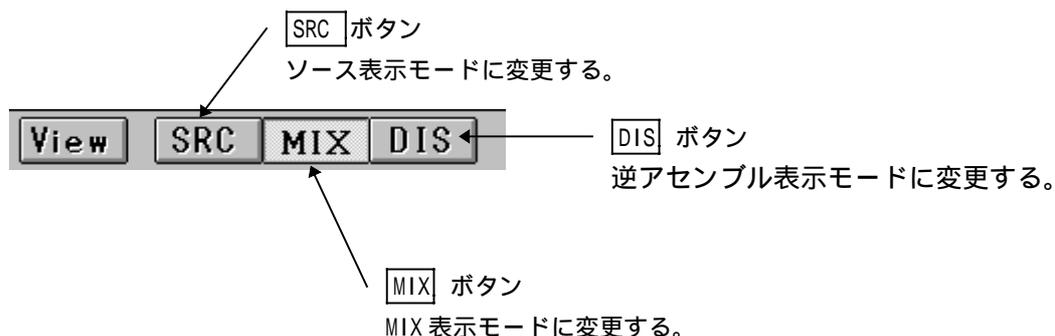
マクロの定義部に PC 値がある場合、PC 値を示す黄色のラインが表示されない場合があります。

## 1.9 逆アセンブル結果を表示するには

逆アセンブル結果を表示するには、プログラム(ソース)ウィンドウのツールバーから“DIS”ボタンをクリックしてください。逆アセンブル結果は、**PD32000**ウィンドウのメニュー

[ Option ] [ Mode ] [ Disasm Mode ]

の選択でも行うことができます。表示モードの変更は、アクティブウィンドウのみ有効です。



ターゲットプログラム停止後のプログラムカウンタ位置が、ソース行情報の存在しない領域であった場合、プログラムウィンドウの表示モードが自動的に逆アセンブル表示モードとなります。

**逆アセンブル表示をソースプログラム表示に戻すには・・・**

プログラム(ソース)ウィンドウのツールバーから“SRC”ボタンをクリックしてください。  
**PD32000**ウィンドウのメニュー

[ Option ] [ Mode ] [ Source Mode ]  
の指定でもソースプログラム表示となります。

**逆アセンブル表示を MIX 表示に戻すには・・・**

プログラム(ソース)ウィンドウのツールバーから“MIX”ボタンをクリックしてください。  
**PD32000**ウィンドウのメニュー

[ Option ] [ Mode ] [ MIX Mode ]  
の指定でもソースプログラム表示となります。

**注意事項**

プログラム(ソース)ウィンドウのプログラム表示領域の先頭行に、ソース行情報が存在しない場合、逆アセンブル表示モードからソース表示モードおよび MIX 表示モードに変更することができません(“SRC”、“MIX”ボタンが無効となります)。この場合、垂直スクロールバー、上下矢印キーを用いてプログラム表示領域の先頭アドレスをソース行情報が存在する位置に変更してください。

## 2 ターゲットプログラムの実行/停止

### 2.1 実行・停止するには

ターゲットプログラムを実行するには・・・

**PD32000**ウィンドウのツールバーから“Go”ボタンをクリックしてください。ターゲットプログラムは、F1 キーの入力でも実行できます。



Go ボタンをクリックする

また、**PD32000**ウィンドウのメニュー  
[ Debug ] [ Go ]

の選択でもターゲットプログラムを実行できます。

実行したターゲットプログラムは、ブレークポイントに到達すると停止します。

ターゲットプログラムをフリーランするには・・・

スクリプトウィンドウから GoFree (短縮名 GF) コマンドを実行します。スクリプトコマンドの使用方法については、本マニュアル 基本操作方法編の項目「7.1 スクリプトコマンドを実行するには」をご参照下さい。

#### [ 入力書式 ]

(書式1) GoFree

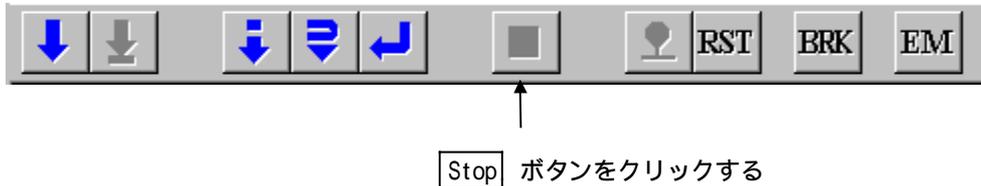
(書式2) GoFree 開始アドレス

#### [ 機能 ]

- 指定した開始アドレスからターゲットプログラムを実行します (書式2)。
- 開始アドレスを省略した場合は、現在のプログラムカウンタが示すアドレスから実行を開始します (書式1)。
- GoFree コマンドでは、全てのブレークポイントが無効になります。

ターゲットプログラムを停止するには・・・

PD32000ウィンドウのツールバーから“Stop”ボタンをクリックしてください。



また、PD32000ウィンドウのメニュー  
 [ Debug ] [ Stop ]  
 の選択でもターゲットプログラムを停止できます。

注意事項

ターゲットプログラム停止後のプログラムカウンタ位置が、ソース行情報の存在しない領域であった場合、プログラムウィンドウの表示モードが逆アセンブル表示モードとなります。

ターゲットプログラムを任意アドレスから実行するには・・・

PD32000ウィンドウのメニュー  
 [ Debug ] [ Go ] [ Go Option... ]  
 を選択してください。Go ダイアログがオープンしますので開始アドレスを入力してください。

ターゲットプログラムの実行状態(実行中か否か)を参照するには・・・

PD32000ウィンドウのステータスバー右部を参照してください。現在の実行状態を表示しています。

MCU : RUN      ターゲットプログラムが実行中の場合

MCU : STOP      ターゲットプログラムが停止中の場合

## 2.2 ステップ実行するには

PD32000ウィンドウのツールバーから“Step”ボタンをクリックしてください。ステップ実行は、F3 キー入力でも実行できます。



プログラムウィンドウがソース表示モードの場合は、ソース行単位でステップ実行します。プログラムウィンドウが逆アセンブル表示モードの場合は、命令単位でステップ実行します。

また、PD32000ウィンドウのメニュー  
 [ Debug ] [ Step ] [ Step ]  
 の選択でもステップ実行が行えます。

### サブルーチンを 1 命令としてステップ実行するには・・・

サブルーチンを 1 命令としてステップ実行することをオーバーステップ実行と呼びます。PD32000ウィンドウのツールバーから“Over”ボタンをクリックしてください。オーバーステップ実行は、F4 キー入力でも実行できます。



また、PD32000ウィンドウのメニュー  
 [ Debug ] [ Over ] [ Over ]  
 の選択でもオーバーステップ実行が行えます。

### ステップ実行の回数を指定したい場合は・・・

PD32000ウィンドウのメニュー  
 [ Debug ] [ Step ] [ Step Option... ]  
 (オーバーステップ実行の場合、[ Debug ] [ Over ] [ Over Option... ])  
 を選択してください。オープンした Step ダイアログ (Over ダイアログ) でステップ回数を指定してください。

### ステップ実行を停止したい場合は・・・

ツールバーの“Stop”ボタンをクリックしてください。PD32000ウィンドウのメニュー  
 [ Debug ] [ Stop ]  
 を選択することにより、ステップ実行を途中で停止させることもできます。オーバーステップ実行の場合も同様です。

## 2.3 現ルーチンから上位ルーチンへ戻るには

PD32000ウィンドウのツールバーから“Return”ボタンをクリックしてください（リターン実行と呼びます）。リターン実行は、F5 キーの入力でも実行できます。



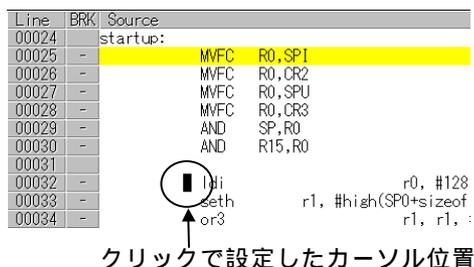
また、PD32000ウィンドウのメニュー  
[ Debug ] [ Return ]  
の選択でもリターン実行が行えます。

リターン実行を途中で停止するには・・・

ツールバーの“Stop”ボタンをクリックしてください。  
また、PD32000ウィンドウのメニュー  
[ Debug ] [ Stop ]  
の選択でもリターン実行を停止させることができます。

## 2.4 指定位置までプログラムを実行するには

指定位置までターゲットプログラムを実行する（カム実行と呼びます）には、まず、プログラム（ソース）ウィンドウのプログラム表示領域の停止させたい行をクリックして、カーソル位置を指定してください。ただし、ソフトウェアブレイクポイントが設定できない行（コメント行、データ定義行等）にカーソル位置を指定した場合は、カム実行は行えません。



カム実行を開始するには、PD32000ウィンドウのツールバーから“Come”ボタンをクリックしてください。カム実行は、F2 キーの入力でも実行できます。



また、PD32000ウィンドウのメニュー  
[ Debug ] [ Come ]  
の選択でもカム実行が行えます。

カム実行を途中で停止するには・・・

ツールバーの“Stop”ボタンをクリックしてください。  
また、**PD32000**ウィンドウのメニュー  
[ Debug ] [ Stop ]  
の選択でもカム実行を停止させることができます。

## 2.5 プログラムをリセットするには

ターゲットプログラムをリセットするには、**PD32000**のツールバーから“Reset”ボタンをクリックしてください。リセットは、F8 キーの入力でも実行できます。



また、**PD32000**ウィンドウのメニュー

[ Debug ] [ Reset ]

の選択でもターゲットプログラムをリセットできます。

## 3 ソフトウェアブレーク

ターゲットプログラムを指定行(アドレス)で停止するには、ソフトウェアブレークを使用します。ソフトウェアブレークポイントを設定するには、ブレークポイント設定ダイアログを使用します。ソフトウェアブレークは、指定したソフトウェアブレークポイントの実行直前でターゲットプログラムを停止します。

- ソフトウェアブレークポイントは、48点まで設定可能です。
- ソフトウェアブレークポイントを複数設定した場合、いずれか1点のソフトウェアブレークポイントに到達するとターゲットプログラムは停止します。

### 3.1 ブレークポイント設定ダイアログをオープンするには

**PD32000**ウィンドウのツールバーから“BRK”ボタンをクリックしてください。ブレークポイント設定ダイアログは、F7 キー入力でもオープンできます。



BRK ボタンをクリックする

また、**PD32000**ウィンドウのメニュー

[ Debug ] [ Break Point ] [ Break Point... ]

の選択でもブレークポイント設定ダイアログをオープンすることができます。

ブレークポイント設定ダイアログの構成については、本マニュアル ウィンドウ機能編の項目「1.13 ブレークポイント設定ダイアログ」をご参照下さい。

#### 注意事項

ブレーク設定アドレスの命令が並列 NOP 命令(Parallel NOP)である場合、ブレークポイントの設定はできません。

### 3.2 ブレークポイントを設定するには

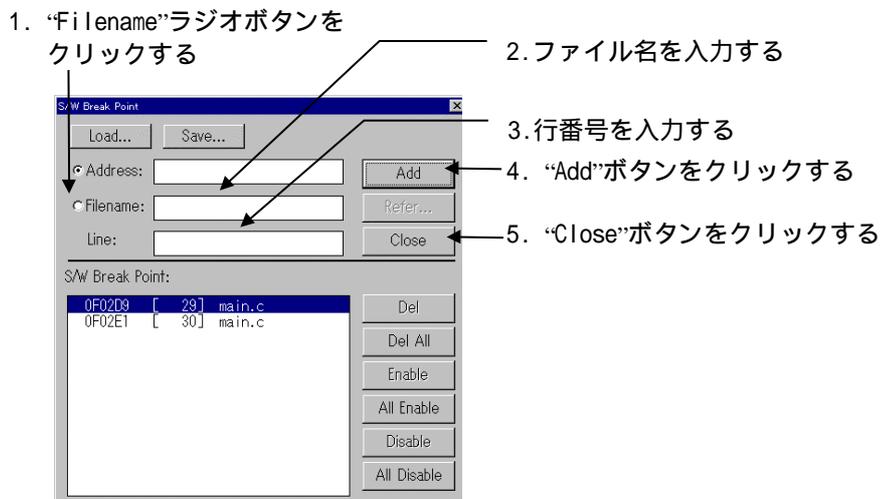
ブレークポイントをアドレスで指定するには・・・

ブレークポイント設定ダイアログから以下の操作を行ってください。アドレス入力には、ラベルを指定することも可能です。



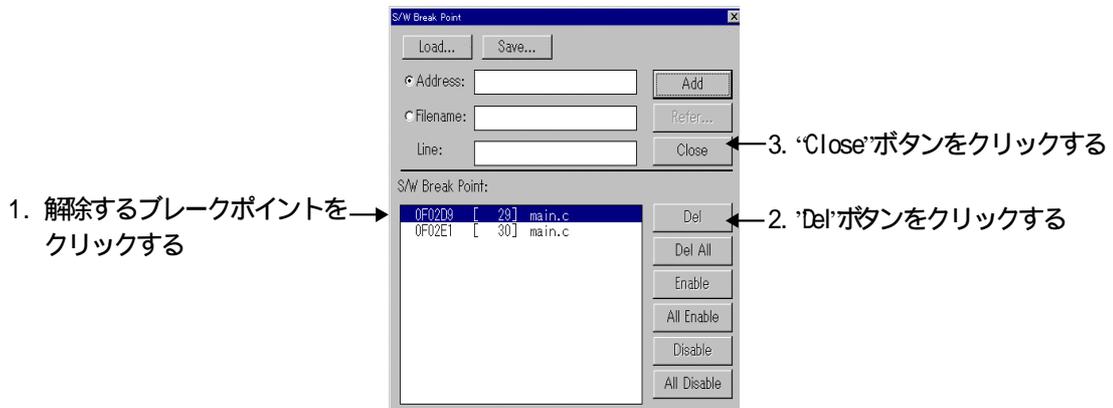
ブレークポイントを行番号で指定するには・・・

ブレークポイント設定ダイアログから以下の操作を行ってください。



### 3.3 ブレークポイントを解除するには

ブレークポイント設定ダイアログから以下の操作を行ってください。



全ブレークポイントを解除する場合は、'Del All'ボタンをクリックする

または、解除するブレークポイントをクリックして、Delete キーを入力することでも行えます。

### 3.4 ブレークポイントを一時的に無効化するには

ブレークポイント設定ダイアログから以下の操作を行います。

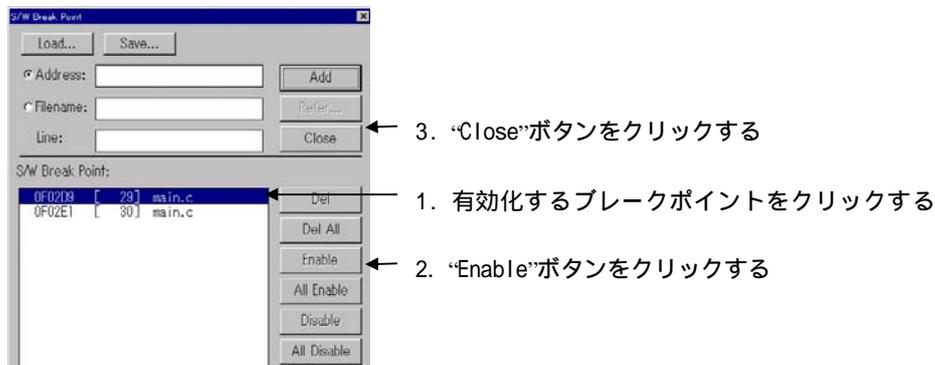


全ブレークポイントを無効化する場合は、'All Disable'ボタンをクリックする

または、無効化するブレークポイントをダブルクリックすることでも行えます（'\*\*\*'が表示されます）。

### 3.5 ブレークポイントを一時的に有効化するには

ブレークポイント設定ダイアログから以下の操作を行ってください。

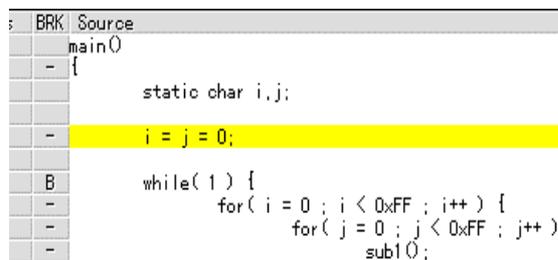


全ブレークポイントを有効化する場合、"All Enable"ボタンをクリックする

または、有効化するブレークポイントをダブルクリックすることでも行えます（"\*"の表示が削除されます）。

### 3.6 ウィンドウ上からブレークポイントを設定するには

プログラム（ソース）ウィンドウからブレークポイントを設定することもできます。プログラム（ソース）ウィンドウから、ブレークさせたい行のブレークポイント表示領域（“-”の表示部分）をダブルクリックしてください。



ここをダブルクリックする

ソフトウェアブレークポイントとして設定された行は、ウィンドウのブレークポイント表示領域が“-”から“B”に変化します。“B”と表示された部分を再度ダブルクリックすると、ソフトウェアブレークポイントが解除されます。

### 3.7 ツールバーからブレークポイントを設定するには

PD32000ウィンドウのツールバーからブレークポイントを設定することもできます。プログラム（ソース）ウィンドウから、ブレークさせたい行のプログラム表示領域（対応するブレークポイント表示領域に“-”の表示がある行）をクリックしてください。

ブレークポイントを設定するには、PD32000ウィンドウのツールバーから”Break”ボタンをクリックしてください。



また、PD32000ウィンドウのメニュー

[ Debug ] [ Break Point ] [ Break ]

の選択でもブレークポイントの設定が行えます。

ソフトウェアブレークポイントとして設定された行は、ウィンドウのブレークポイント表示領域が“-”から“B”に変化します。“B”と表示された行を再度クリックして”Break”ボタンをクリックすると、ソフトウェアブレークポイントが解除されます。

### 3.8 ブレークポイントを保存するには

ブレークポイント設定ダイアログの[Save] ボタンをクリックするとファイル選択ダイアログがオープンしますので、保存するファイル名を指定してください。拡張子を省略した場合、拡張子”.brk”が付加されます。

### 3.9 ブレークポイントを読み込むには

保存したファイルからブレークポイントを読み込むには、ブレークポイント設定ダイアログの[Load] ボタンをクリックするとファイル選択ダイアログがオープンしますので、読み込むファイルを指定してください。現在設定されているブレークポイントにファイルから読み込んだブレークポイントを追加します。ソフトウェアブレークポイントが48点を超える場合は、49点目以降は無視します。

## 4 トレース

トレース機能では、サイクル毎のバス情報参照、実行した命令の参照、ソースレベルで実行経路の参照が行えます。トレース結果は、トレースウィンドウで参照します。トレースウィンドウは、**PD32000**ウィンドウのメニュー

[ Optional Window ] [ Trace Window ]

の選択でオープンします。

また、トレース条件の設定は、トレースコンディションウィンドウで設定します。トレースコンディションウィンドウは、**PD32000**ウィンドウのメニュー

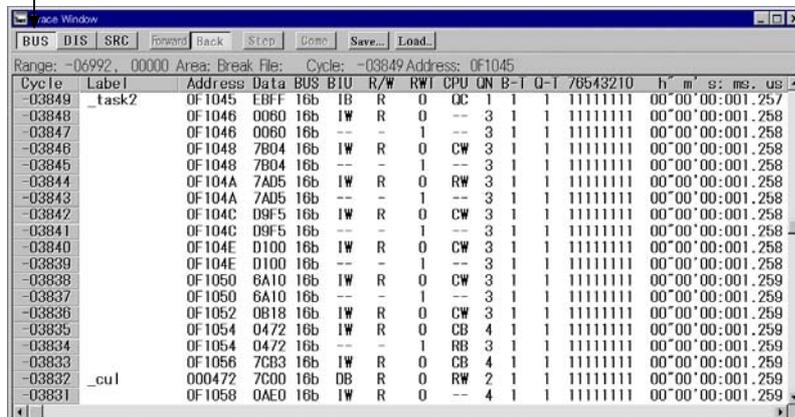
[ Optional Window ] [ Trace Condition ]

の選択でオープンします。

### 4.1 バス情報を参照するには

トレースコンディションウィンドウのトレースモードを signal トレースに設定した場合、トレースウィンドウの“BUS”ボタンが有効になります。他の表示モードへの変更はできません。

バスモードに変更する場合は、このボタンをクリックする



表示サイクル外のバス情報は、上下の矢印キー操作、Page Up/Page Down キー操作、垂直スクロールバーによるスクロールで参照してください。

バス情報の詳細については、本マニュアル ウィンドウ機能編の項目「1.16.1 バスモード時の画面構成」をご参照下さい。

トレースサイクル指定によって参照位置を変更するには・・・

トレースウィンドウのサイクル数表示領域をダブルクリックしてください。Cycle ダイアログがオープンしますのでサイクル数を入力してください。Cycle ダイアログは、**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Cycle ]

の選択でもオープンします。

指定したアドレスを検索するには・・・

トレースウィンドウのアドレス表示領域をダブルクリックしてください。Address Search ダイアログがオープンしますのでアドレスを入力してください。Address Search ダイアログは**PD32000**ウィンドウのメニュー

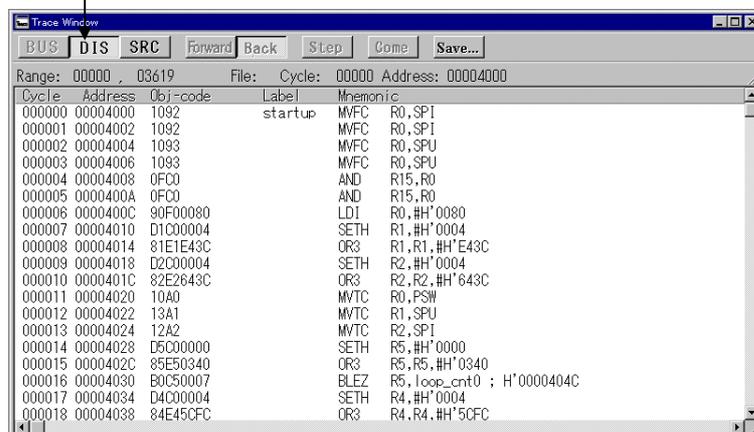
[ Option ] [ View ] [ Address Search ]  
 の選択でもオープンします。

## 4.2 実行した命令で実行経路を参照するには

トレースコンディションウィンドウのトレースモードをPC トレースに設定した場合、トレースウィンドウの“DIS”/“SRC”ボタンが有効になります。“DIS”ボタンをクリックすると、トレースウィンドウの表示モードが逆アセンブルモードに切り替わります。

なお PC トレース時は、“BUS”ボタンは無効になります。

逆アセンブルモードに切り替えるには、このボタンをクリックする



表示外の逆アセンブル情報は、上下の矢印キー操作、Page Up/Page Down キー操作、垂直スクロールバーによるスクロールで参照してください。

トレースサイクル指定によって参照位置を変更するには・・・

トレースウィンドウのサイクル数表示領域をダブルクリックしてください。Cycle ダイアログがオープンしますのでサイクル数を入力してください。Cycle ダイアログは、**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Cycle ]  
 の選択でもオープンします。

指定アドレスのサイクルを検索するには・・・

トレースウィンドウのアドレス表示領域をダブルクリックしてください。Address Search ダイアログがオープンしますのでアドレスを入力してください。Address Search ダイアログは、**PD32000**ウィンドウのメニュー

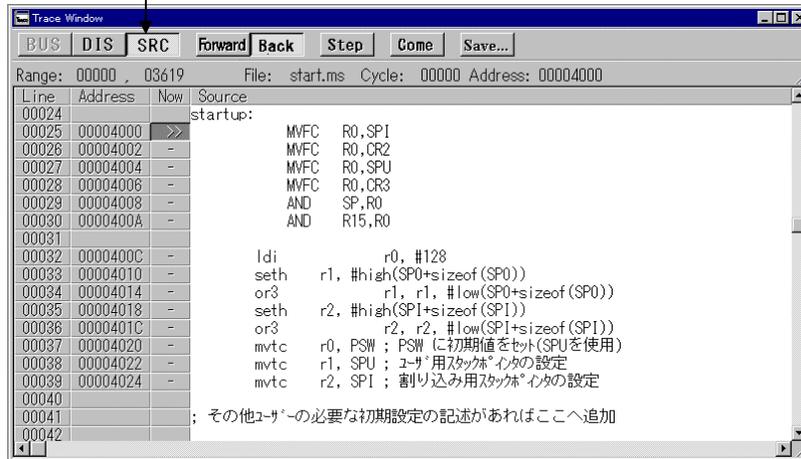
[ Option ] [ View ] [ Address Search ]  
 の選択でもオープンします。

## 4.3 ソース行レベルで実行経路を参照するには

トレースコンディションウィンドウのトレースモードをPC トレースに設定した場合、トレースウィンドウの“DIS”/“SRC”ボタンが有効になります。“SRC”ボタンをクリックすると、トレース

スウィンドウの表示モードがソースモードに切り替わります。  
 なお PC トレース時は、"BUS"ボタンは無効になります。

ソースモードに変更する場合は、このボタンをクリックする



実行経路を逆方向に参照するには、トレースウィンドウのツールバーから"Back"ボタンをクリックし、"Step"ボタンで次実行経路を参照します。実行経路を順方向に参照するには、トレースウィンドウのツールバーから"Forward"ボタンをクリックし、"Step"ボタンで次実行経路を参照します。

実行経路を順方向に指定するには、このボタンをクリックする



実行経路を逆方向に指定するには、このボタンをクリックする

Forward / Back ボタンの指定に従い、次の実行経路を参照するには、このボタンをクリックする

トレースサイクル指定によって参照位置を変更するには・・・

**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Cycle ]

を選択すると Cycle ダイアログがオープンしますのでサイクル数を入力してください。

指定アドレスのサイクルを検索するには・・・

トレースウィンドウのアドレス表示領域をダブルクリックしてください。Address Search ダイアログがオープンしますのでアドレスを入力してください。Address Search ダイアログは、**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Address Search ]

の選択でもオープンします。

任意ソース行に対応したサイクルを検索するには・・・

トレースウィンドウのソースファイル表示領域の任意ソース行をクリックした後、“Come”ボタンをクリックしてください。“Come”ボタンをクリックすると検索を始めます。検索方向は、現在の実行経路指定（順方向検索の“Forward”/逆方向検索の“Back”）に従います。

検索経路を順方向に指定するには、このボタンをクリックする



検索経路を逆方向に指定するには、このボタンをクリックする

任意ソース行実行サイクルを検索するには、このボタンをクリックする

トレース計測結果をファイルに保存するには・・・

トレースウィンドウのツールバーから“Save...”ボタンをクリックしてください。ファイル名を付けて保存ダイアログがオープンしますので、トレース計測結果を保存するファイル名を指定してください。ファイル名を指定した後、種類（サイクル毎のバス情報/逆アセンブル情報）と保存するトレース計測結果の範囲を指定してください。保存したトレースデータのファイルは読み込みできません。

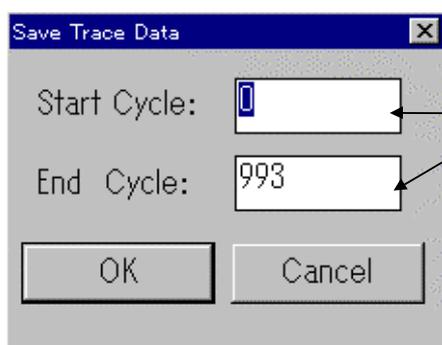
**PD32000**ウィンドウのメニュー

[ Option ] [ Save... ]

を選択してもトレース計測結果をファイルに保存できます。



トレース計測結果をファイルに保存するには、このボタンをクリックする



ファイルにテキスト形式で保存するトレース計測結果の範囲を指定する

## 4.4 トレース条件を設定するには

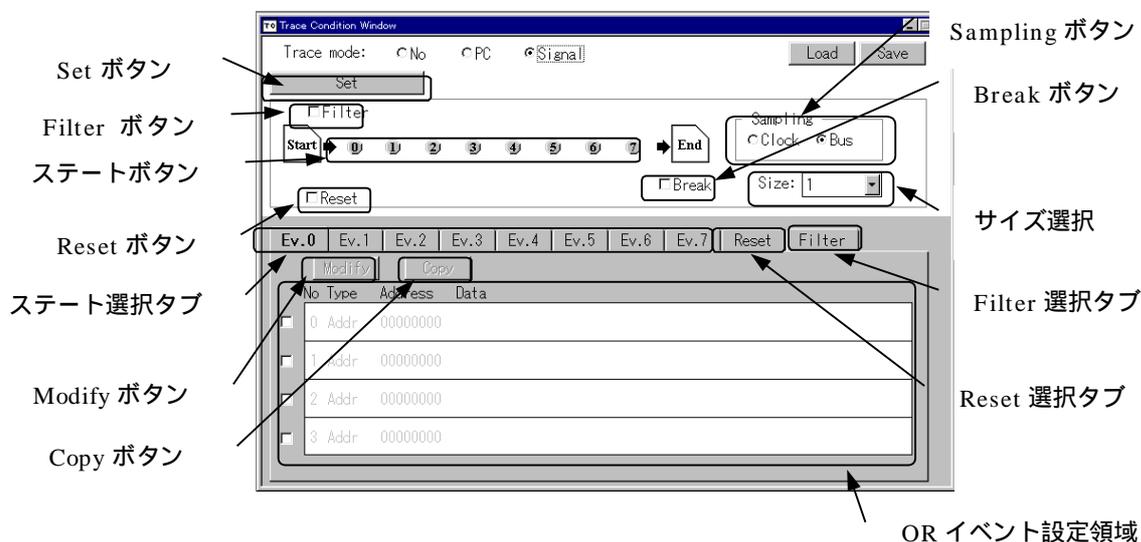
トレース条件として PC トレース、Signal トレースが設定できます。

### 4.4.1 PC トレースモードの設定

トレースコンディションウィンドウのトレースモードで PC トレースを選択してください。他の設定は不要です。

### 4.4.2 Signal トレースモードの設定

トレースコンディションウィンドウのトレースモードで Signal トレースを選択してください。トレースコンディションウィンドウが Signal モードに切り替わります。



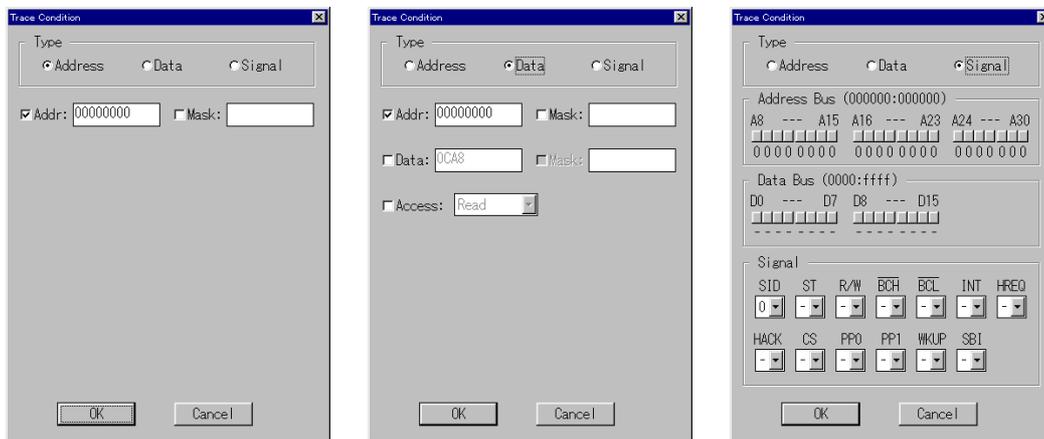
シーケンシャルイベントは最大 8 ステートまで設定することができます。また、各ステートにおいて、OR イベントを最大 4 ポイント設定することができます。

以下に各領域の説明をします。

- Set ボタン  
各設定を登録します。
- Filter ボタン  
トレースデータをフィルタリングするか指定するボタンです。フィルタイベントは Filter 選択タブで設定します。なお、フィルタイベントは全ステートの 4 番イベントを使用します。4 番イベントにイベントを設定していた場合、Filter ボタンをチェックすることにより、このイベントは無効になります。
- シーケンシャルボタン  
番号をクリックすることで、該当ステートの OR イベント領域が開きます。
- Reset ボタン  
リセットイベントを有効にするか指定するボタンです。リセットイベントを有効にすると、イベントが成立した場合、最初のステートに戻ります。リセットイベントは Reset 選択タブに記述します。なお、リセットイベントは全ステートの 3 番イベントを使用します。3 番イベントにイベントを設定していた場合、Reset ボタンをチェックすることにより、このイベントは無効になります。
- Break ボタン  
トレースイベントが成立し、トレースデータが指定のサイズ分取得した際に、実行を停止させるか指定するボタンです。チェックしている場合、停止します。
- Sampling ボタン  
サンプリングを選択するボタンです。Clock または Bus が選択できます。
- サイズ設定ボックス  
トレースイベント成立後、トレースメモリに格納するトレースデータのサイズを指定します。指定できるサイズは 16K バイト単位で 0 ~ 256K までです。
- シーケンシャル選択タブ  
各シーケンシャルの OR イベントを設定するタブです。
- Filter 選択タブ  
Filter イベントを設定するタブです。
- Reset 選択タブ  
Reset イベントを設定するタブです。
- Modify ボタン  
現在表示されているタブの OR イベントの変更を行うためのボタンです。クリックすることによりトレース設定ダイアログが開きます。
- Copy ボタン  
イベントのコピー、またはチェンジを行うボタンです。
- イベント設定領域  
各ステートのイベント、Filter イベント、および Reset イベントを設定する領域です。イベントの設定は、イベントを選択後、Modify ボタンまたはダブルクリックで設定ダイアログが開きます。また、Copy ボタンによりコピーおよびチェンジが可能です。設定されたイベントは、番号手前のチェックボックスをクリックすることで有効になります。

## イベントを登録するには・・・

イベント設定領域でイベントを選択して Modify ボタンをクリック、または該当イベント領域でダブルクリックすることにより、設定ダイアログが開きます。



Address モード (Default)

Data モード

Signal モード

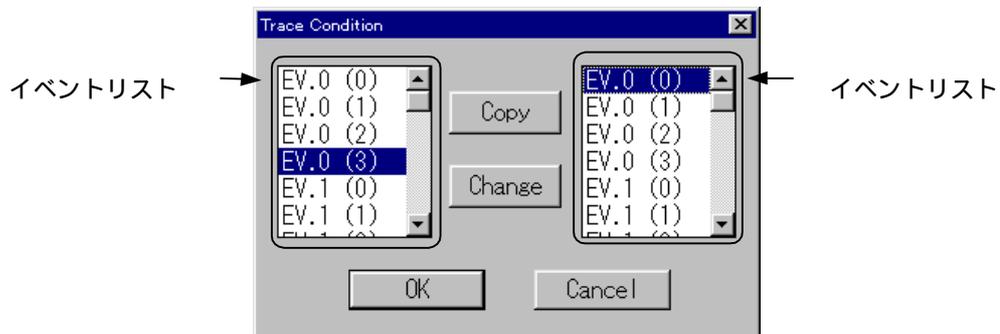
設定ダイアログは、設定モードにより表示および操作が異なります。

- Address モード
  - Addr をクリックすることにより、有効 / 無効が切り替わります。
  - Addr の値は 16 進数で入力します。
  - Mask をクリックすることにより、有効 / 無効が切り替わります。この Mask は Addr が有効な場合のみ、有効となります。
  - Mask の値は 16 進数で入力します。
  - OK ボタンをクリックすることにより登録されます。

- Data モード
  - Addr をクリックすることにより、有効 / 無効が切り替わります。
  - Addr の値は 16 進数で入力します。
  - Addr の右側の Mask をクリックすることにより、有効 / 無効が切り替わります。  
この Mask は Addr が有効な場合のみ、有効となります。
  - Mask の値は 16 進数で入力します。
  - Data をクリックすることにより、有効 / 無効が切り替わります。
  - Data の値は 16 進数で入力します。
  - Data の右側の Mask をクリックすることにより、有効 / 無効が切り替わります。  
この Mask は Data が有効な場合のみ、有効となります。
  - Mask の値は 16 進数で入力します。
  - Access をクリックすることにより、有効 / 無効が切り替わります。
  - Access の値は Read、Write より選択します。  
Write を指定した場合、アクセスサイズを 8bit、16bit より選択できます。
  - OK ボタンをクリックすることにより登録されます。
- Signal モード
  - Address bus 領域ではアドレスバス A8 - A30 の値を設定します。各ボタンはクリックすることにより、'0','1','-'に切り替わります。'0'は値 0 で、'1'は値 1 を示します。 '-'はマスクすることを示します。
  - Data bus 領域ではアドレスバス D0 - D15 の値を設定します。各ボタンはクリックすることにより、'0','1','-'に切り替わります。
  - Signal 領域では、各シグナルの値を設定します。各シグナルは'0','1','-'から選択します。
  - OK ボタンをクリックすることにより登録されます。

**イベントをコピーするには・・・**

イベントをコピーする場合、Copy ボタンをクリックしてください。Copy ダイアログが開きます。



左側のイベントリストからコピーするイベントを選択します。  
続いて右側のイベントリストからコピー先のイベントを選択します。  
Copy ボタンをクリック、OK ボタンをクリックすることにより、イベントがコピーされます。  
また、OK ボタンの代わりに Change ボタンをクリックすれば、イベントのチェンジができます。

## 5 レジスタ情報、メモリ内容の参照・設定

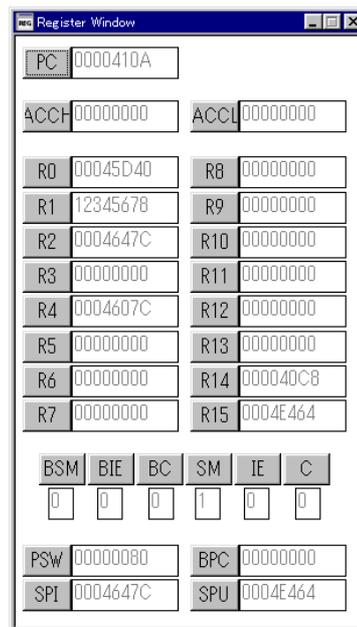
### 5.1 レジスタの内容を参照するには

レジスタの内容を参照するには、レジスタウィンドウをオープンします。レジスタウィンドウでは、CPU が持つレジスタの一覧を表示しています。レジスタウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ Register Window ]

の選択でオープンします。

レジスタウィンドウについては、本マニュアル ウィンドウ機能編の項目「1.4 レジスタウィンドウ」をご参照下さい。



### 5.2 レジスタの内容を変更するには

レジスタの内容を変更するには、レジスタウィンドウ内の変更するレジスタのボタンをクリックしてください。レジスタの値を設定する Set Register ダイアログがオープンします。変更する値を入力してください。

フラグの値を変更するには・・・

レジスタウィンドウ内の変更するフラグのボタンをクリックしてください。クリックするたびにフラグの状態が切り替わります。

## 5.3 任意アドレスの値を参照するには

任意アドレスの値を参照するには、ウォッチ機能を使用します。任意アドレスの値は、ASM ウォッチウィンドウで参照します。この参照するアドレスをウォッチポイントと呼びます。ASM ウォッチウィンドウは、**PD32000**ウィンドウのメニュー

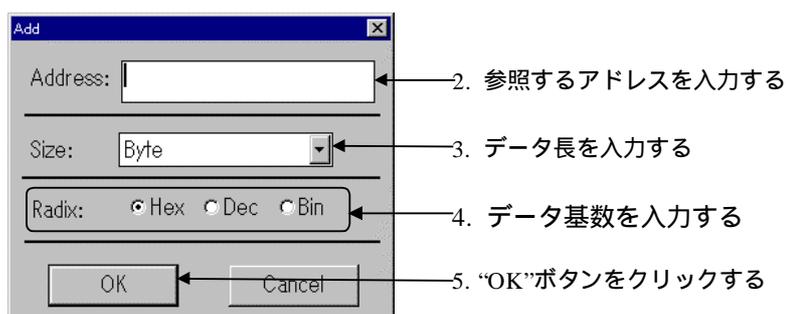
[ Basic Window ] [ ASM Watch Window ]

の選択でオープンします。ASM ウォッチウィンドウについては、本マニュアル ウィンドウ機能編の項目「1.8 ASM ウォッチウィンドウ」をご参照下さい。

### ウォッチポイントを登録するには・・・

ASM ウォッチウィンドウのメニューバーから“Add”ボタンをクリックしてください。“Add”ボタンをクリックすると Add ダイアログがオープンします。参照するアドレスを入力してください。

1. **Add** ボタンをクリックすると、下のダイアログがオープンする



登録したウォッチポイントは、ASM ウォッチウィンドウの現在のカーソル位置に追加されます。カーソル位置は、アドレス表示領域とデータ表示領域に、赤いマークで表示されます。また、カーソル位置は、どちらかの領域をクリックするか、または   キーで移動できます。

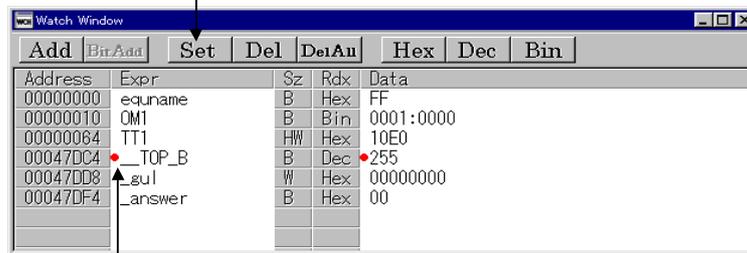
### 注意事項

Size で指定するデータ長は、表示サイズのデータ長を表しています。ターゲットボードおよびエミュレータのメモリへは、MPU ファイルで規定したメモリサイズでアクセスします。MPU ファイルの詳細については、セットアップ編「1.3.2 動作環境の設定」を参照ください。

### 任意アドレスの値を変更するには・・・

ASM ウォッチウィンドウから変更するウォッチポイントを選択し、ASM ウォッチウィンドウのツールバー“Set”ボタンをクリックしてください。

2. **Set** ボタンをクリックする



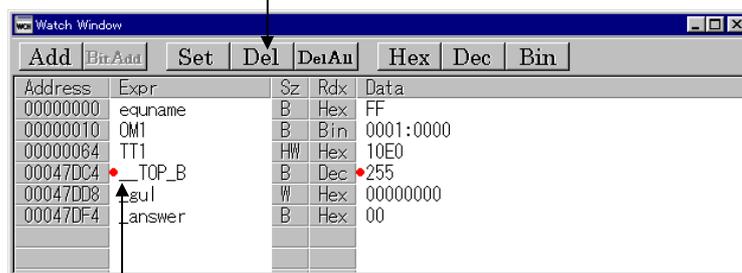
1. 変更するウォッチポイントをクリックする

Set ダイアログがオープンしますので変更値を入力してください。

### ウォッチポイントを削除するには・・・

ASM ウォッチウィンドウから削除するウォッチポイントを選択し、ASM ウォッチウィンドウのツールバー“Del”ボタンをクリックしてください。

2. **Del** ボタンをクリックする



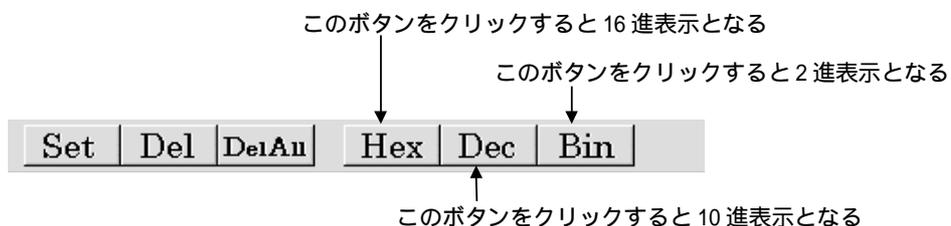
1. 削除するウォッチポイントをクリックする

\* 全ウォッチポイントを削除する場合は、**DelAll** ボタンをクリックする

または、削除するウォッチポイントをクリックして、Delete キーを入力することでも行えます。

**メモリの参照形式を変更するには・・・**

表示基数を変更することができます。ASM ウォッチウィンドウから変更するウォッチポイントを選択し、ASM ウォッチウィンドウのツールバー“Hex”、“Dec”、“Bin”ボタンのいずれかをクリックしてください。



また、ASM ウォッチウィンドウの基数表示領域をダブルクリックすることでも、変更が行えます。

## 5.4 指定アドレスにデータを設定するには

指定アドレスへのデータ設定は、メモリウィンドウ、及びダンプウィンドウ上から設定できます。メモリウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ Memory Window ]

の選択でオープンします。ダンプウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ Dump Window ]

の選択でオープンします。

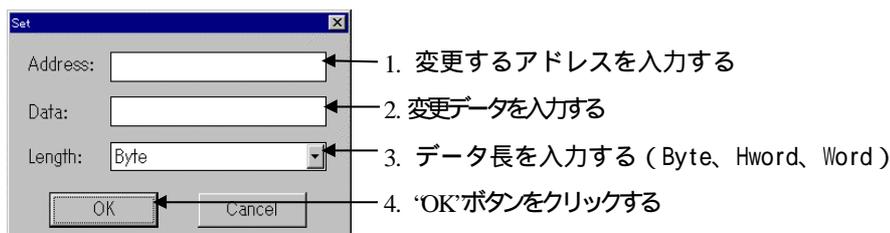
メモリウィンドウについては、本マニュアル ウィンドウ機能編の項目「1.5 メモリウィンドウ」、ダンプウィンドウについては、本マニュアル ウィンドウ機能編の項目「1.6 ダンプウィンドウ」をご参照下さい。

### 指定アドレスのデータを変更するには・・・

メモリウィンドウ、あるいはダンプウィンドウがアクティブな状態で、**PD32000**ウィンドウのメニュー

[ Option ] [ Debug ] [ Set... ]

を選択してください。選択すると Set ダイアログがオープンします。変更するアドレスとそのデータを入力してください。



また、Set ダイアログは、メモリウィンドウ、あるいはダンプウィンドウのデータ表示領域をダブルクリックすることでもオープンします。

### 注意事項

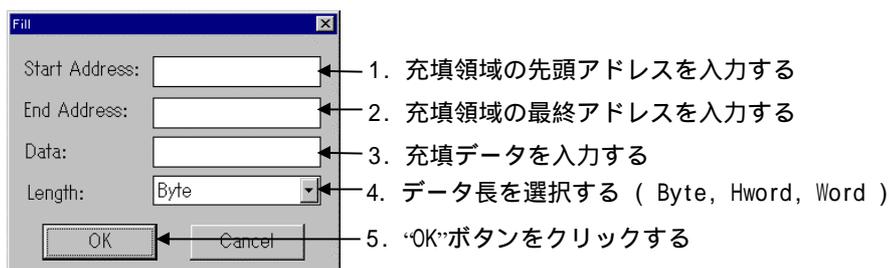
Length で指定するデータ長は、表示サイズのデータ長を表しています。ターゲットボードおよびエミュレータのメモリへは、MPU ファイルで規定したメモリサイズでアクセスします。MPU ファイルの詳細については、セットアップ編「1.3.2 動作環境の設定」を参照ください。

### 指定領域を一定データで充填するには・・・

メモリウィンドウ、あるいはダンプウィンドウがアクティブな状態で、**PD32000**ウィンドウのメニュー

[ Option ] [ Debug ] [ Fill... ]

を選択してください。選択すると Fill ダイアログがオープンします。充填するアドレス範囲と充填データを入力してください。



### 注意事項

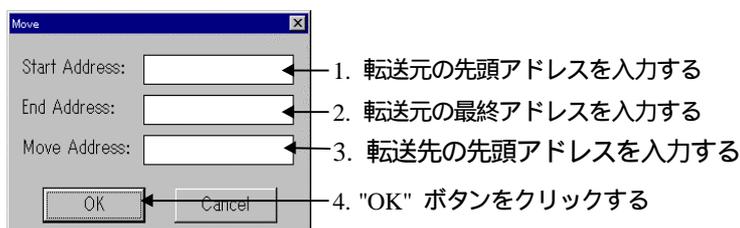
Length で指定するデータ長は、表示サイズのデータ長を表しています。ターゲットボードおよびエミュレータのメモリへは、MPU ファイルで規定したメモリサイズでアクセスします。MPU ファイルの詳細については、セットアップ編「1.3.2 動作環境の設定」を参照ください。

### 指定領域を他の領域へ移動するには・・・

メモリウィンドウ、あるいはダンプウィンドウがアクティブな状態で、**PD32000**ウィンドウのメニュー

[ Option ] [ Debug ] [ Move... ]

を選択してください。選択すると Move ダイアログがオープンします。移動するアドレス範囲と移動先のアドレスを入力してください。



## 5.5 メモリ内容の表示を更新するには

メモリウィンドウ、及びダンプウィンドウでは、メモリ内容の変更を伴うコマンド（メモリの設定・充填・転送、実行停止、ステップ実行など）を実行した際、メモリ内容の表示を自動的に更新します。しかし、I/O 領域など、MCU の実行とは無関係に変更される領域を表示している際に、表示しているデータと実際のメモリ内容が異なることがあります。

この場合、メモリウィンドウで最新のメモリ内容を表示するには、メモリウィンドウがアクティブな状態で、**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Refresh ]

を選択してください（ツールバーの Refresh ボタンでも可能）。

また、ダンプウィンドウで最新のメモリ内容を表示するには、ダンプウィンドウがアクティブな状態では、**PD32000**ウィンドウのメニュー

[ Option ] [ View ] [ Refresh ]

を選択してください（ツールバーの Refresh ボタンでも可能）。

## 6 C 変数の参照・変更

### 6.1 C 変数の値を参照するには

ターゲットプログラム中で宣言されている C 言語の変数（以下 C 変数と記述）を参照するために、**PD32000**では4つのウィンドウを用意しています。

- ローカルウィンドウ
- ファイルローカルウィンドウ
- グローバルウィンドウ
- C ウォッチウィンドウ

ローカルウィンドウは、関数内でのみ有効な変数の値を表示するウィンドウです。ローカルウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ C Watch Window ] [ Local Window ]

の選択でオープンします。

ファイルローカルウィンドウは、ファイル内でのみ有効な変数の値を表示するウィンドウです。ファイルローカルウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ C Watch Window ] [ File Local Window ]

の選択でオープンします。

グローバルウィンドウは、グローバルな変数の値を表示するウィンドウです。グローバルウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ C Watch Window ] [ Global Window ]

の選択でオープンします。

C ウォッチウィンドウは、任意変数の値を表示するウィンドウです。C ウォッチウィンドウは、**PD32000**ウィンドウのメニュー

[ Basic Window ] [ C Watch Window ] [ C Watch Window ]

の選択でオープンします。

ローカルウィンドウ、ファイルローカルウィンドウ、グローバルウィンドウは、参照する C 変数を選択することができません。各ウィンドウは、現在のターゲットプログラムの実行位置に応じて、表示する変数が変化します。

- ローカルウィンドウ  
現在実行されている関数に応じて、表示する変数が変化する。
- ファイルローカルウィンドウ  
現在実行されているソースファイルに応じて、表示する変数が変化する。
- グローバルウィンドウ  
実行位置に関係なく、C 言語のグローバル変数の一覧を表示する。

なお、任意の C 変数を参照する場合は、C ウォッチウィンドウを使用してください。

## C ウォッチポイントを登録するには・・・

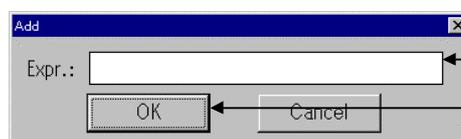
(方法1 C ウォッチウィンドウから登録)

C ウォッチウィンドウのツールバーから“Add”ボタンをクリックしてください。“Add”ボタンをクリックすると Add ダイアログがオープンします。Add ダイアログから C 言語式 (C 言語の変数、式 等) を指定してください。ポインタとして C 言語式を登録する場合は、ツールバーから“Add\*”ボタンをクリックしてください。これにより、C 言語式が C ウォッチポイントとして登録されます。

1. C ウォッチポイントを登録するには、Add ボタンをクリックします



1. C ウォッチポイントをポインタとして登録するには、Add\* ボタンをクリックします



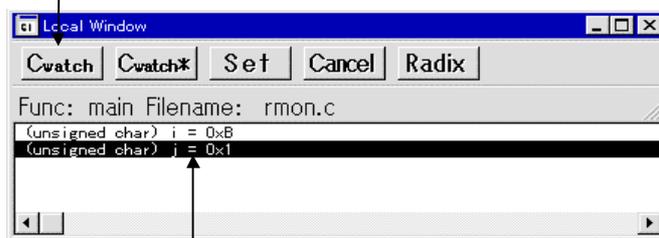
2. C 言語式を入力します

3. "OK" ボタンをクリックします

(方法2 ローカル/ファイルローカル/グローバルウィンドウから登録)

ローカルウィンドウ、ファイルローカルウィンドウ、グローバルウィンドウに表示されている変数を C ウォッチポイントとして登録することができます。登録したい変数をクリックしてください。その後、そのウィンドウのツールバーから“Cwatch”ボタンをクリックしてください。ポインタとして登録する場合は、“Cwatch\*”ボタンをクリックしてください。

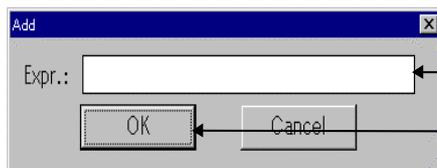
2. Cwatch ボタンをクリックする



1. 登録する C 変数をクリックする

## 6.2 C 変数の値を変更するには

ローカルウィンドウ、ファイルローカルウィンドウ、グローバルウィンドウ、C ウォッチウィンドウのいずれからも C 変数の値を変更することが可能です。変更したい C 変数をクリックし、ウィンドウのツールバーから“Set”ボタンをクリックしてください。“Set”ボタンをクリックすると Set ダイアログがオープンします。変更値を入力してください。



1. 変数値を入力する

2. "OK" ボタンをクリックする

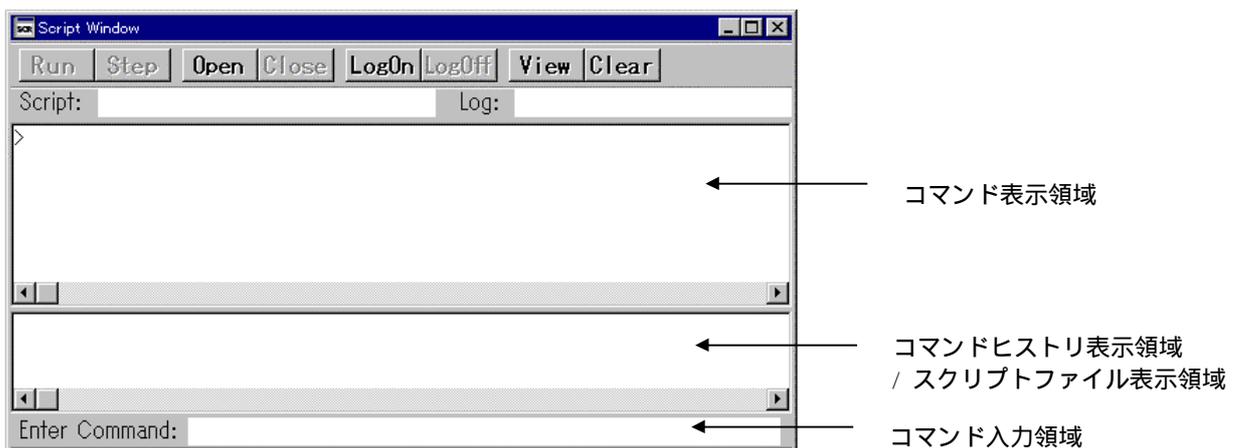
## 7 スクリプトコマンド

### 7.1 スクリプトコマンドを実行するには

スクリプトコマンドは、スクリプトウィンドウから実行します。スクリプトウィンドウは、PD32000ウィンドウのメニュー

[ Basic Window ] [ Script Window ]

の選択でオープンします。スクリプトコマンドは、スクリプトウィンドウのコマンド入力領域から入力します。コマンド入力領域をクリックし、フォーカスを与えてスクリプトコマンドを入力してください。コマンド実行結果は、コマンド表示領域に出力されます。

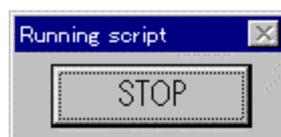


以前実行したスクリプトコマンドを再実行するには・・・

コマンド履歴表示領域にこれまでの実行履歴が表示されています。再実行するコマンドをダブルクリックしてください。

スクリプトコマンドを途中終了するには・・・

スクリプトコマンド実行中のみ表示されるダイアログの STOP ボタンをクリックしてください。



## 7.2 スクリプトコマンドの実行結果を記録するには

スクリプトコマンドの実行結果を記録するには、ロギング機能を使用します。スクリプトコマンドを実行する前に、スクリプトウィンドウのツールバーから“LogOn”ボタンをクリックしてください。

“LogOn”ボタンをクリックするとファイルセレクションダイアログがオープンしますので、記録するファイル名を指定してください。ファイルの拡張子を省略した場合は、“.log”となります。



記録を始めるには、**LogOn** ボタンをクリックする

### PD32000ウィンドウのメニュー

[ Option ] [ Log ] [ On... ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

- **PD32000**の起動後に一度オープン/クローズして保存したログファイルを再びオープンした場合は、既存のログファイルの最後に新たな内容を追加します。ただし、今回の**PD32000**起動以前に既に作成されていたログファイルを再びオープンした場合は、上書きになります。
- ファイルセレクションダイアログのファイルリストには、拡張子が“.log”のファイル名の一覧を優先して表示します。しかし、ファイル名を入力する領域に直接、フルネームでファイル名を入力することにより、ファイル属性“.log”以外のファイルをログファイルとしてオープンすることも可能です。
- ログファイルは、最大8段までネストしてオープンできます。

実行結果の記録を終了するには・・・

スクリプトウィンドウのツールバーから“LogOff”ボタンをクリックしてください。



記録停止は、**LogOff** ボタンをクリックする

### PD32000ウィンドウのメニュー

[ Option ] [ Log ] [ Off ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

- ログファイルがネストしている場合、現在のログファイルへの出力は、終了しますが、1つ前のログファイルへの出力を再開します。

### 実行結果をコマンド実行後に記録するには・・・

**PD32000**は、最新の1000行分のコマンド実行結果を保存するビューバッファを持っています。ビューバッファの内容を保存するには、スクリプトウィンドウのツールバーから“View”ボタンをクリックしてください。

“View”ボタンをクリックするとファイルセレクションダイアログがオープンしますので、記録するファイル名を指定してください。ファイルの拡張子を省略した場合は、“:viw”となります。



実行後記録するには、View ボタンをクリックする

#### PD32000ウィンドウのメニュー

[ Option ] [ View ] [ Save... ]

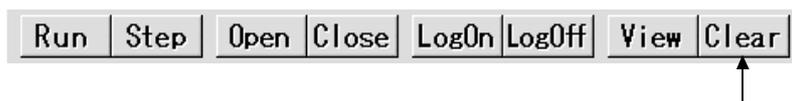
の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

- ファイル名に既存のファイル名を指定した場合は、既存ファイルの最後にビューバッファ内容を追加します。
- ファイルセレクションダイアログのファイルリストには、拡張子が“.viw”のファイル名の一覧を優先して表示します。しかし、ファイル名を入力する領域に直接、フルネームでファイル名を入力することにより、ファイル属性“.viw”以外のファイルをビューファイルとしてオープンすることも可能です。

### 実行結果の表示を消去するには・・・

コマンド表示領域の内容を消去するには、スクリプトウィンドウのメニューから“Clear”ボタンをクリックしてください。

“Clear”ボタンをクリックするとコマンド表示領域の内容が消去されます。また、ビューバッファの内容も消去されます。



コマンド表示領域の内容を消去するには、Clear ボタンをクリックする

#### PD32000ウィンドウのメニュー

[ Option ] [ View ] [ Clear ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

## 7.3 スクリプトコマンドを一括して実行するには

スクリプトコマンドは、一括して実行することもできます。あらかじめ、実行するコマンドをお手持ちのエディタでファイルに記述してください。ファイルの拡張子は、".scr"としてください。

次にスクリプトウィンドウからスクリプトファイルを読み込みます。スクリプトウィンドウのツールバーから“Open”ボタンをクリックしてください。

“Open”ボタンをクリックするとファイルセレクションダイアログがオープンしますので、実行するスクリプトファイルを選択してください。



スクリプトファイルを開くには、**Open** ボタンをクリックする

### PD32000ウィンドウのメニュー

[ Option ] [ Script ] [ Open ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

- ファイルセレクションダイアログのファイルリストには、拡張子が".scr"のファイル名の一覧を優先して表示します。しかし、ファイル名を入力する領域に直接、フルネームでファイル名を入力することにより、ファイル属性".scr"以外のファイルもスクリプトファイルとしてオープンすることも可能です。
- スクリプトファイルは、最大、5 段までネストしてオープンできます。

スクリプトファイルを読み込むとスクリプトウィンドウのコマンド履歴表示領域がスクリプトファイル表示領域に変化します。

スクリプトファイルの内容を一括実行するには、スクリプトウィンドウのツールバーから“Run”ボタンをクリックしてください。

“Run”ボタンをクリックすると一括実行を始めます。スクリプトファイルは、記述されたすべてのコマンド実行後にクローズします。



スクリプトファイルを一括実行するには、**Run** ボタンをクリックする

### PD32000ウィンドウのメニュー

[ Option ] [ Script ] [ Run ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

**スクリプトファイル実行を中断するには・・・**

スクリプトファイル実行中に表示されるダイアログの STOP ボタンをクリックしてください。



スクリプトファイルの実行は、その次の行が実行される手前で停止します。

**スクリプトファイルのコマンドを一コマンドずつ実行するには・・・**

スクリプトウィンドウのツールバーから“Step”ボタンをクリックしてください（スクリプトのステップ実行）。“Step”ボタンをクリックする度に一コマンドずつ実行します。



スクリプトファイルをステップ実行するには、Step ボタンをクリックする

**PD32000 ウィンドウのメニュー**

[ Option ] [ Script ] [ Step ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

**スクリプトファイルをクローズするには・・・**

スクリプトウィンドウのツールバーから“Close”ボタンをクリックしてください。



スクリプトファイルをステップ実行するには、Step ボタンをクリックする

**PD32000 ウィンドウのメニュー**

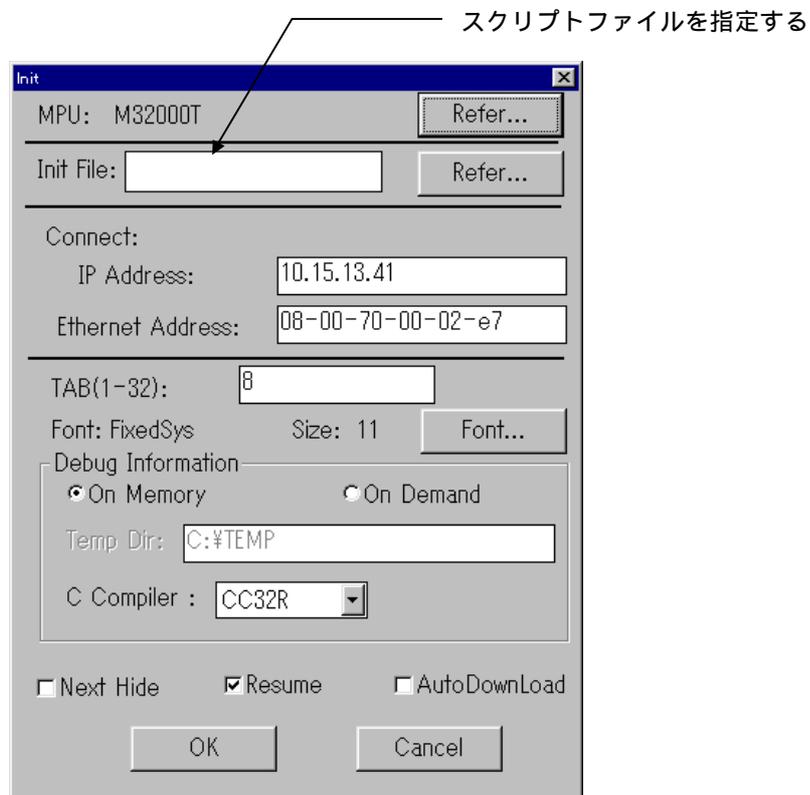
[ Option ] [ Script ] [ Close ]

の指定でも同じ動作を行います（スクリプトウィンドウがアクティブ時）。

- スクリプトファイルがネストしている場合、現在のスクリプトファイルはクローズしますが、1 つ前のスクリプトファイルがオープンします。

**スクリプトコマンドをPD32000起動時に実行するには・・・**

起動時に表示する Init ダイアログにおいて起動時に実行するスクリプトファイル名を指定してください。



## 8 PD32000の終了

### 8.1 PD32000を終了するには

PD32000ウィンドウのメニュー

[ File ] [ Exit ]

を選択してください。終了確認のためのダイアログがオープンしますのでOK ボタンをクリックしてください。



↑OKボタンをクリックする

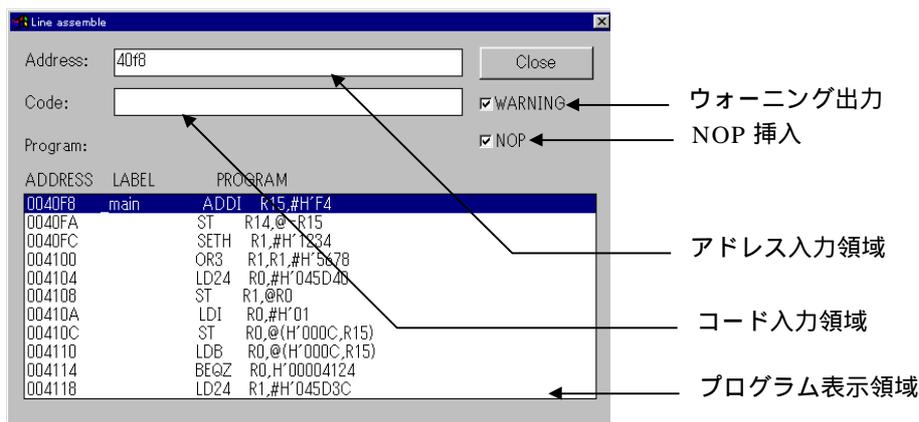
## 9 その他

### 9.1 ラインアセンブルするには

ラインアセンブルするには、ラインアセンブルダイアログで行う方法とスクリプトコマンドで行う2種類の方法があります。

#### 9.1.1 ラインアセンブルダイアログで行う方法

プログラム(ソース)ウィンドウでラインアセンブルを行いたい位置をクリックし、メニュー [ Option ] [ Line Assemble ] を選択すると以下に示すラインアセンブルダイアログがオープンします(位置を指定しない場合は、ダイアログをオープンした後にアドレスを入力してください)。



- アドレス入力領域、コード入力領域にラインアセンブルを行うアドレスとアセンブル命令を入力し、リターンキーを押します。本ダイアログのプログラム表示領域で、反転表示している行がラインアセンブルの対象となるアドレスです。
  - 入力した命令バイト数に変更前の命令バイト数より少ない場合は、NOP 命令を挿入してバイト数を合わせます。
  - 入力した命令バイト数に変更前の命令バイト数より多い場合、ウォーニングダイアログを表示します。OK ボタンを押すと、入力した命令を書き込みます。Cancel ボタンを押すと、書き込みを中止します
- NOP 挿入を OFF にすると、入力した命令バイト数に変更前の命令バイト数より短い場合でも NOP 命令を挿入しません。
- ウォーニング出力を OFF にすると、ウォーニングダイアログを表示せずに、強制的に入力した命令を書き込みます。
- 本ダイアログのプログラム表示領域の任意の行をクリックすることにより、クリックした行をラインアセンブルの対象にすることができます。

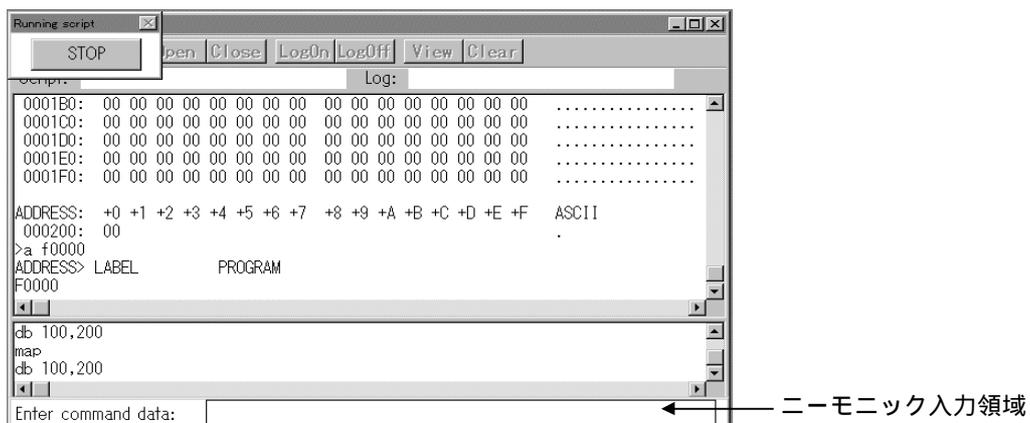
### 9.1.2 Assemble コマンドで行う方法

ラインアセンブルするには、スクリプトウィンドウからアセンブルコマンドを実行します。コマンド名は、Assemble（短縮名 A）です。コマンドの後にラインアセンブルするアドレスを指定してください。

>Assemble アセンブルアドレス

Assemble コマンドを実行するとスクリプトウィンドウのコマンド入力領域がニーモニック入力領域となります。

スクリプトコマンドの使用方法については、本マニュアル 基本操作方法編の項目「7.1 スクリプトコマンドを実行するには」をご参照下さい。ラインアセンブルするニーモニックを入力してください。以下にスクリプトウィンドウのニーモニック入力待ち状態を示します。



ラインアセンブルを終了するには・・・

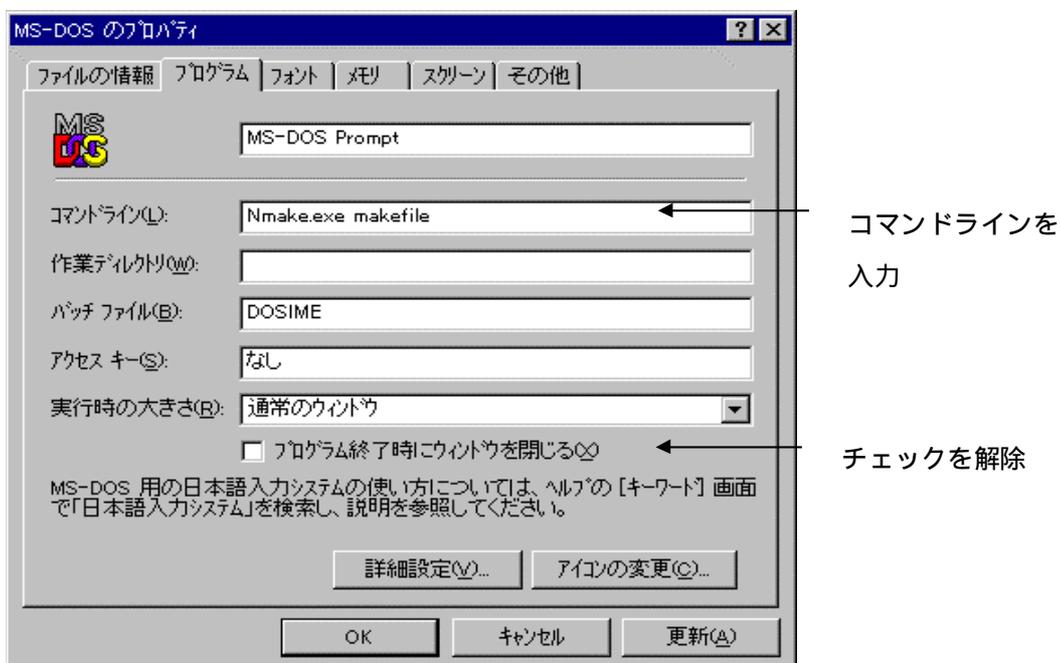
ニーモニック入力領域に“Enter”のみを入力してください。スクリプトウィンドウがスクリプトコマンド入力待ち状態となります。

## 9.2 Make を起動するには

従来、DOS 窓からコマンドを入力してターゲットプログラムを Make していた作業を PD32000 から行うことができます。

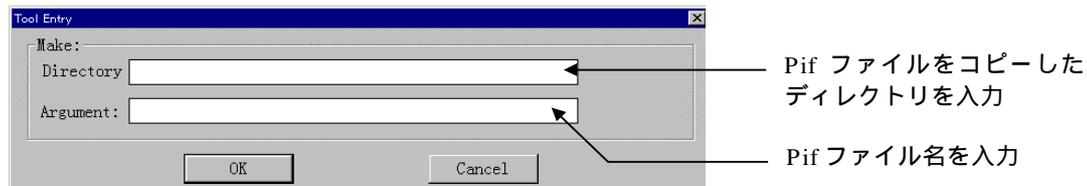
PD32000 で Make を行う手順を以下に示します。

- 1 pif ファイルを作成します。  
以下の手順で pif ファイルを作成してください。
  - (a) windows のディレクトリにある command.com のショートカットキーを作成します。
  - (b) 作成したショートカットキーのファイル名を xxxx.pif (xxxx の部分は、任意の名前を指定してください) にし、メイクファイルのあるディレクトリにコピーします。これが pif ファイルとなります。ファイル名変更時に拡張子.pif を指定しないでください。
  - (c) このファイルのプロパティダイアログをオープンし、コマンドラインに DOS 窓で入力していたコマンドを入力します。以下に、設定例を示します。



プロパティダイアログは、エクスプローラなどで pif ファイルを選択してマウスの右ボタンをクリックした際に表示されるメニューの [ プロパティ ] を選択することによりオープンします。

- 2 pif ファイルをPD32000に登録します。  
メニュー [ Debug ] [ Entry ] を選択し、以下のダイアログをオープンし、pif ファイルを登録します。



- 3 Make を起動します。  
メニュー [ Debug ] [ Make ] を選択すると、pif ファイルで指定した内容を実行します。

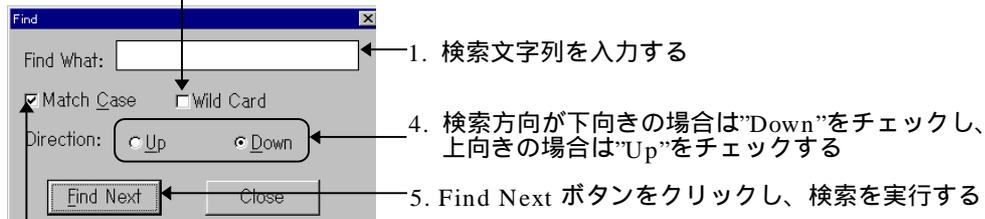
### 9.3 ターゲットプログラムの文字列を検索するには

プログラムウィンドウあるいはソースウィンドウがアクティブな状態の時、ターゲットプログラム内の文字列を検索することができます。対象とするプログラム(ソース)ウィンドウをアクティブにし、PD32000ウィンドウのメニュー

[ Edit ] [ Find... ]

を選択してください。選択すると以下の Find ダイアログがオープンします。検索文字列を入力して検索を開始してください。

3. 検索文字列にワイルドカードを指定する場合は、ここをチェックする



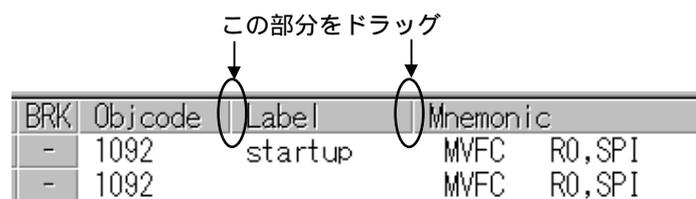
2. 検索文字列の大文字 / 小文字を区別する場合は、ここをチェックする

### 9.4 ウィンドウの表示領域の割合を変更するには

プログラムウィンドウ、ソースウィンドウ、メモリウィンドウ、ASM ウォッチウィンドウ、トレースウィンドウ、及びMR ウィンドウでは、各項目の表示領域の広さの割合をマウスで調節することができます。以下に各ウィンドウごとに、その方法を示します。

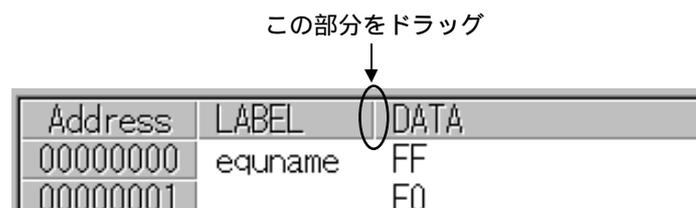
- プログラムウィンドウ、及びソースウィンドウ

MIX 表示モードおよび逆アセンブル表示モードのときに、オブジェクトコード表示領域 ( Objcode ) とプログラム表示領域の 2 つの領域 ( Label と Mnemonic ) の広さの割合が調節できます。



- メモリウィンドウ

ラベル表示領域 ( LABEL ) とメモリ内容表示領域の ( DATA ) の広さの割合が調節できます。



- ASM ウォッチウィンドウ  
式表示領域 ( Expr ) の広さの割合が調節できます。

この部分をドラッグ

Address	Expr	Sz	Rdx	Data
00000000	•equname	B	Hex	•FF
00000010	OM1	B	Bin	0001:

- Trace ウィンドウ  
BUS 表示モードのときに、ラベル表示領域 ( ( Label ) ) の広さの割合が調節できます。

この部分をドラッグ

Cycle	Label	Address	Data	SI	ST	RW
145468	_main	002000F8	4FB0	0	0	1
145469		002000FA	247F	0	0	1
145470		002000FC	257F	0	0	1
145471		002000FE	267F	0	0	1
145472		00200100	277F	0	0	1
145473		00200102	287F	0	0	1

- MR ウィンドウ  
タスク状態表示モード、周期起動ハンドラ状態表示モード、アラームハンドラ状態表示モードのとき、タスク名表示領域 ( ( name ) ) の広さの割合が調節できます ( 下図は、タスク状態表示モード時の例を示しています )。

この部分をドラッグ

ID	StaAddr	(name)	Pri	Status	wup_count	timeout	f
1	0F0B54H	(__D_AHCB_ENT)	1	RUN	0000H	-----	--
2	0F0C9FH	(_task2)	2	RDY	0000H	-----	--
3	0F0CAFH	(_task3)	2	SUS	0000H	-----	--
4	0F0CBFH	(_task4)	1	WAI	0000H	-----	--
5	0F0CD4H	(_task5)	1	WAI-SUS	0000H	-----	--

## 9.5 アクティブウィンドウを切換えるには

PD32000では、操作対象となるウィンドウがアクティブになっている必要があります。マウスでウィンドウをクリックしてアクティブウィンドウを切換える他に、キーボード操作でも切換えることができます。

キーボードからアクティブウィンドウを切換えるには以下のように入力します。

[ Ctrl ] + [ TAB ]

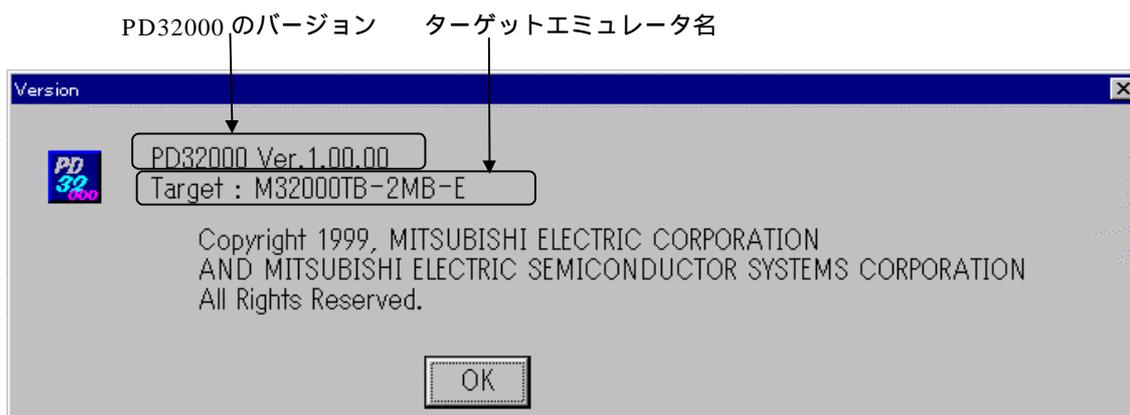
Shift キーを押しながら上記操作を行うと、逆順にアクティブウィンドウが切換わります。

## 9.6 PD32000のバージョンを表示するには

PD32000ウィンドウのメニュー

[ Help ] [ About... ]

を選択してください。About ダイアログがオープンします。About ダイアログでは、PD32000のバージョンを表示しています。



OK ボタンを押すと About ダイアログは、クローズします。

【 MEMO 】

# より高度なデバッグ編



# 1 区間時間測定

## 1.1 概要

区間時間を測定するには、区間時間測定ウィンドウを使用します。

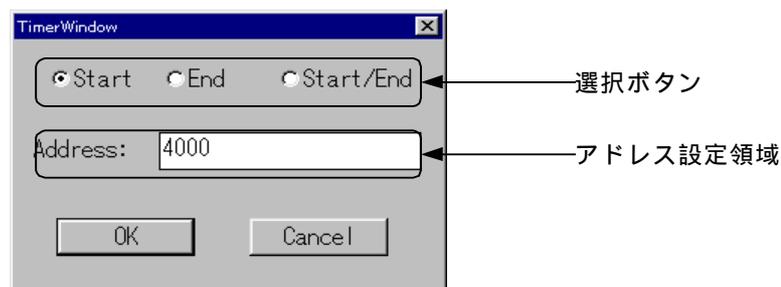
区間時間測定は指定された区間の最大、最小、平均実行時間、測定回数を測定する機能です。エミュレータM32000Tでは、同時に最大20点の区間時間を測定することができます。

## 1.2 測定条件の設定

測定条件を設定するには、Add ボタンをクリックします。



Add ボタンをクリックすると以下のダイアログがオープンします。このダイアログを区間設定ダイアログと呼びます。



選択ボタンでは次の中から選択します。

- Start 測定開始を示します。
- End 測定終了を示します。
- Start/End 測定開始および測定終了を示します。

アドレス設定領域にアドレスを入力し、OK ボタンをクリックすると登録されます。

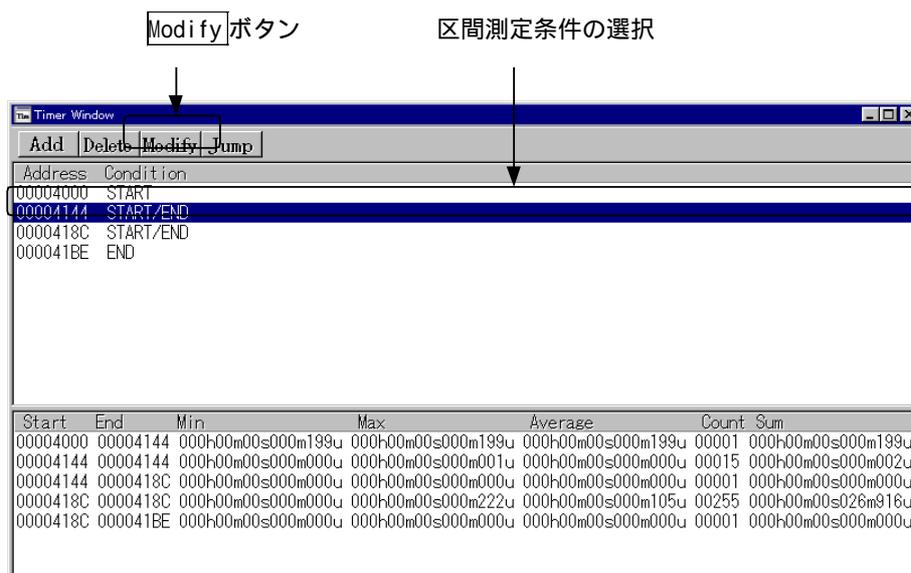
### 1.3 測定条件の削除

測定条件を削除するには、測定区間設定状況表示領域の削除項目をクリックして、Delete ボタンを押します。



### 1.4 測定条件の変更

区間時間測定ウィンドウにおいて、変更したい条件をクリックして、Modify ボタンを押します。



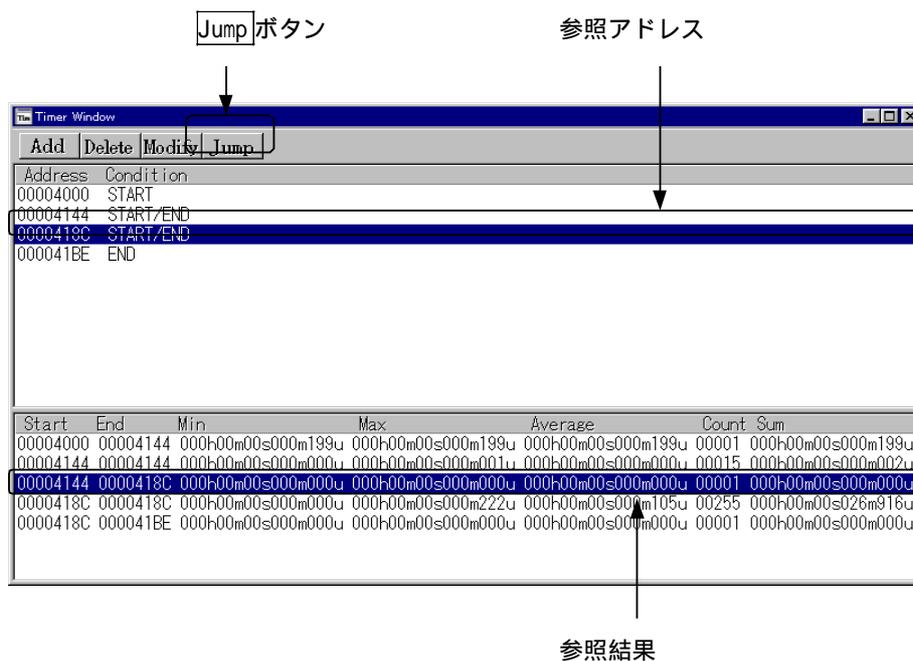
区間設定ダイアログがオープンしますので、ここで変更します。

## 1.5 区間時間測定結果を参照するには

区間時間測定結果は、プログラムを実行、停止されることにより、測定結果表示領域に表示されます。

区間時間測定条件に対応した測定結果を参照するには・・・

区間時間測定条件に対応した測定結果を参照するには、参照したいアドレスを選択して、Jump ボタンを押すことにより、参照結果が表示されます。



## 2 カスタマイズ機能

### 2.1 カスタマイズ機能とは

カスタマイズ機能とは、**PD32000**にユーザ独自の機能を追加する機能です。**PD32000**にカスタムコマンドプログラムおよびカスタムウィンドウプログラムをロードすることにより、**PD32000**の機能を拡張することが可能です。

カスタムコマンドプログラム/カスタムウィンドウプログラムを作成するには・・・

カスタムコマンドプログラム/カスタムウィンドウプログラムは、**PD32000**に付属のCB32000 (Custom Builder for **PD32000**)を用いて作成します。CB32000 で作成したカスタムコマンドプログラムおよびカスタムウィンドウプログラムは、MACRO コマンドを用いて**PD32000**に登録することにより使用可能となります。カスタムコマンドプログラム/カスタムウィンドウプログラムの作成方法は、「CB32000 ユーザーズマニュアル」を参照ください。

カスタムウィンドウプログラムを使用するには・・・

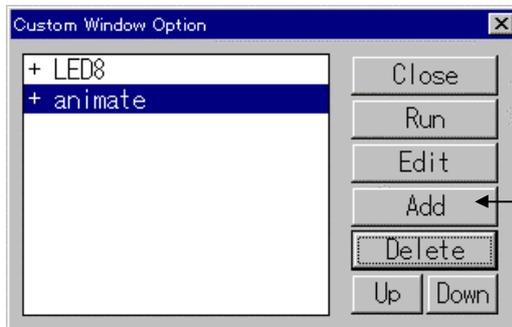
CB32000 で作成したカスタムウィンドウプログラムを使用するには、以下に示す2つの方法があります。

1. カスタムウィンドウをメニューに登録し、メニューから起動する。
2. スクリプトウィンドウから MACRO コマンドで登録および起動する。

メニューへの登録方法および起動方法を以下に示します。MACRO コマンドでの登録および起動方法は、後述の「カスタムコマンドプログラム/カスタムウィンドウプログラムを使用するには・・・」を参照ください。

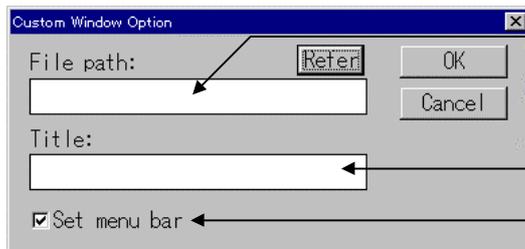
## メニューへの登録方法

カスタムウィンドウをメニューに登録するには、メニュー  
 [ Optional Window ] [ Custom Window ] [ Option ]  
 を選択し、以下のダイアログをオープンします。



1. カスタムウィンドウ登録  
 ダイアログのオープン

### カスタムウィンドウランチャー



2. カスタムウィンドウのプログラム  
 ファイル名(拡張子.pを含む)を絶  
 対パス付きで入力する。
3. メニューに表示する名称を入力する。
4. メニューへの表示/非表示を選択
5. **OK** ボタンをクリックする。

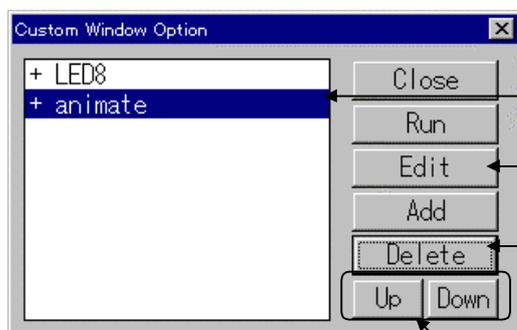
### カスタムウィンドウ登録ダイアログ

#### 注意

カスタムウィンドウをスクリプトウィンドウから起動する場合は、MACRO コマンドで  
 カスタムウィンドウを登録してください。

#### 登録内容の変更

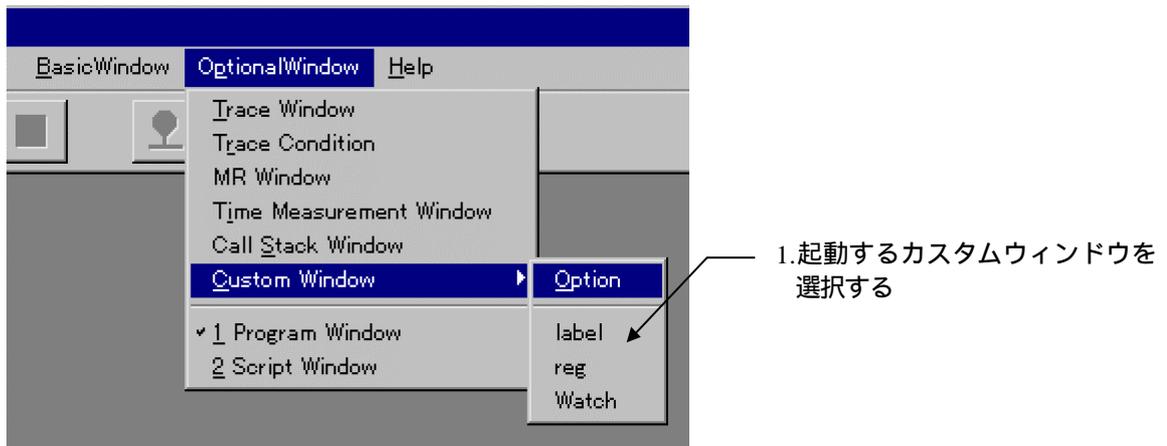
登録内容を変更するには、下記手順でカスタムウィンドウ登録ダイアログをオープンし、登録  
 内容を変更してください。



1. 登録内容を変更するカスタムウィンド  
 ウを選択する
2. カスタムウィンドウ登録  
 ダイアログのオープン
- 登録を削除する場合は、**Delete** ボタ  
 ンをクリックする
- メニュー表示の順番を変更する

### カスタムウィンドウの起動

カスタムウィンドウを登録すると、カスタムウィンドウを起動するためのメニューが追加されます。



メニューを非表示にしているカスタムウィンドウを起動する場合は、カスタムウィンドウランチャーで起動するカスタムウィンドウを選択して、**Run** ボタンをクリックします。

### カスタムコマンドプログラム/カスタムウィンドウプログラムを使用するには・・・

CB32000 で作成したカスタムコマンドプログラム/カスタムウィンドウプログラムは、スクリプトウィンドウから MACRO コマンドを使用して登録します。パラメータとしてカスタムコマンド/ウィンドウのプログラムファイル名( 拡張子.p は省略する )を指定して**PD32000** に登録することにより使用可能となります。

(例) >MACRO custom<RET>

MACRO コマンドで登録した、カスタムコマンドプログラムの実行およびカスタムウィンドウプログラムのウィンドウオープンは、スクリプトコマンドの実行と同様の手続きで行えます。MACRO コマンドで登録したプログラムファイル名をコマンド名として実行します。

(例) >custom<RET>

MACRO コマンドで登録した、カスタムコマンドプログラムおよびカスタムウィンドウプログラムは、DELMACRO コマンドおよびDELMACROALL コマンドを用いて削除されるまで、削除されません。( **PD32000**を終了しても、保持されます。)

なお、**PD32000**は、カスタムコマンドプログラム/カスタムウィンドウプログラムを**PD32000**のカレントディレクトリ(スクリプトコマンド CD で設定したディレクトリ)から検索します。もし別ディレクトリに格納される場合は、後記の検索ディレクトリの説明に従って、カスタムコマンド/ウィンドウプログラムの格納ディレクトリを指定してください。

### カスタムコマンドプログラム/カスタムウィンドウプログラムを削除するには・・・

MACRO コマンドで登録した、カスタムコマンドプログラムおよびカスタムウィンドウプログラムの削除は、DELMACRO コマンドに MACRO コマンドで登録したファイル名( 拡張子は除く )を指定するか、DELMACROALL コマンドを用います。DELMACRO コマンドでは、指定されたカスタムコマンドプログラムまたはカスタムウィンドウプログラムのみを削除します。DELMACROALL コマンドでは、登録されている全カスタムコマンドプログラムおよびカスタムウィンドウプログラムを削除します。

(例) >DELMACRO custom<RET>

>DELMACROALL<RET>

### カスタムコマンドプログラム/カスタムウィンドウプログラムの検索ディレクトリを設定するには・・・

MACRO コマンドで登録した、カスタムコマンドプログラムおよびカスタムウィンドウプログラムは、実行される際に**PD32000**にロードされます。ロードするカスタムコマンドプログラムおよびカスタムウィンドウプログラムは、MACROPATH(MPATH)コマンドで設定されたディレクトリから検索されます。MACROPATH(MPATH)で指定可能なディレクトリは1つです。

(例) >MACROPATH c:¥usr¥project¥custom<RET>

【 MEMO 】

# リアルタイム OS デバッグ編

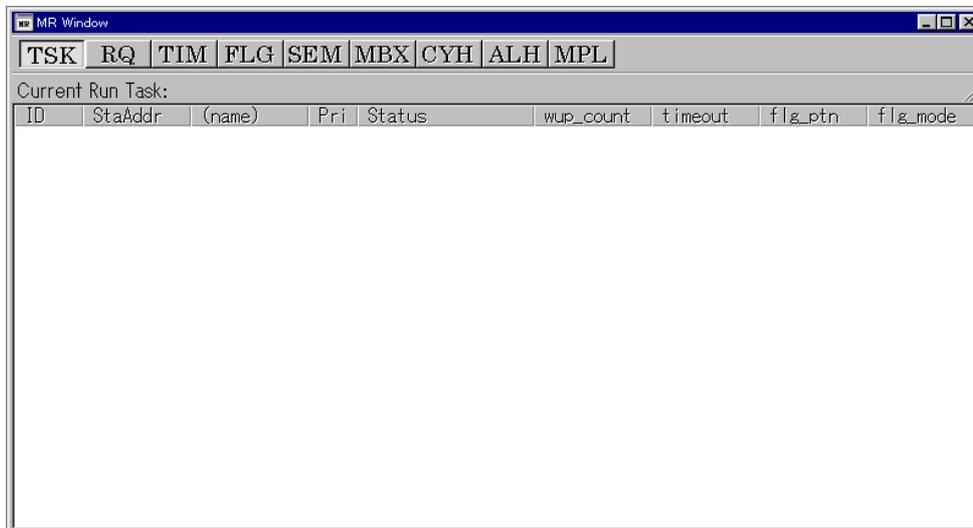


# 1 リアルタイム OS 用のデバッグ機能

## 1.1 リアルタイム OS の情報を参照するには

リアルタイム OS 情報を参照するには、MR ウィンドウをオープンするか、スクリプトウィンドウから MR コマンドを実行します。スクリプトコマンドの使用方法については、本マニュアル基本操作方法編の項目「7.1 スクリプトコマンドを実行するには」をご参照ください。

以下に MR ウィンドウの構成図を示します。



### (--注意事項--)

ターゲットプログラム作成の際、ご使用の MR32R のバージョンに対応したスタートアップファイル (CC32R用 crt0mr.ms/start.ms、TW32R 用 crt0mr.s/start.s) をご使用下さい。スタートアップファイルの内容が MR32R のバージョンに対応していない場合、MR ウィンドウ・MR コマンドは正常に動作しません。

### タスクの状態を表示するには・・・

MR ウィンドウのツールバーから “ TSK ” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Task ] を選択してください。

以下に MR ウィンドウのタスク状態表示モードの構成図を示します。

The screenshot shows a window titled "MR Window" with a menu bar containing TSK, RQ, TIM, FLG, SEM, MBX, CYH, ALH, and MPL. Below the menu bar, it says "Current Run Task: [1] (\_TOP\_P)". A table displays the following data:

ID	StaAddr	(name)	Pri	Status	wup_count	timeout	flg_ptn	flg_mode
1	00066114H	(_TOP_P)	1	RUN	0000H	-----	-----	-----
2	000666F4H	(_task2)	2	RDY	0000H	-----	-----	-----
3	000666FCH	(_task3)	2	SUS	0000H	-----	-----	-----
4	00066704H	(_task4)	1	WAI(SLP)	0000H	-----	-----	-----
5	00066718H	(_task5)	1	WAI(SLP)-SUS	0000H	-----	-----	-----
6	0006672CH	(_task6)	1	WAI(DLY)	0000H	00007528H	-----	-----
7	0006674CH	(_task7)	1	WAI(DLY)-SUS	0000H	00007528H	-----	-----
8	0006676CH	(_task8)	1	WAI(FLG)	0000H	-----	00001111H	TWF_ORW
9	0006679CH	(_task9)	1	WAI(FLG)-SUS	0000H	-----	00001111H	TWF_ORW
10	000667CCH	(_task10)	1	WAI(SEM)	0000H	-----	-----	-----
11	000667E8H	(_task11)	1	WAI(SEM)-SUS	0000H	-----	-----	-----
12	00066804H	(_task12)	1	WAI(MBX)	0000H	-----	-----	-----
13	00066828H	(_task13)	1	WAI(MBX)-SUS	0000H	-----	-----	-----
14	0006684CH	(_task14)	---	DMT	-----	-----	-----	-----

TSK モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全てのタスクについて、ID 番号順に表示します。

ID	タスクの ID 番号を表示します。
StaAddr	タスクの開始アドレスを表示します。
(name)	タスク名を表示します。
Pri	タスクの優先度を表示します。
Status	タスクの状態を表示します。
wup_count	タスクのウェイクアップカウント値を表示します。
timeout	タスクが時間待ち状態の場合、そのタイムアウト値を表示します。
flg_ptn	タスクがイベントフラグ待ち状態の場合、その待ちビットパターンを表示します。
flg_mode	タスクがイベントフラグ待ち状態の場合、その待ち解除条件を表示します。

TSK モードの場合、ステータスバーの表示内容は次のようになります。

**Current Run Task**----- 現在実行中のタスク ID とタスク名  
(アイドル状態の時は“nothing...” と表示します。)

Status 領域の表示内容には、以下の種類があります。

Status	状態
RUN	実行状態
RDY	実行可能状態
SUS	強制待ち状態
DMT	休止状態
WAI(SLP)	スリープ状態
WAI(SLP)-SUS	スリープ状態 (二重待ち)
WAI(DLY)	dly_tsk による時間待ち状態
WAI(DLY)-SUS	dly_tsk による時間待ち状態 (二重待ち)
WAI(FLG)	イベントフラグ待ち状態
WAI(FLG)-SUS	イベントフラグ待ち状態 (二重待ち)
WAI(SEM)	セマフォ待ち状態
WAI(SEM)-SUS	セマフォ待ち状態 (二重待ち)
WAI(MBX)	メッセージ待ち状態
WAI(MBX)-SUS	メッセージ待ち状態 (二重待ち)
WAI(SLP-TMO)	タイムアウト付スリープ状態
WAI(SLP-TMO)-SUS	タイムアウト付スリープ状態 (二重待ち)
WAI(FLG-TMO)	タイムアウト付イベントフラグ待ち状態 <sup>2</sup>
WAI(FLG-TMO)-SUS	タイムアウト付イベントフラグ待ち状態 (二重待ち)
WAI(SEM-TMO)	タイムアウト付セマフォ待ち状態 <sup>2</sup>
WAI(SEM-TMO)-SUS	タイムアウト付セマフォ待ち状態 (二重待ち)
WAI(MBX-TMO)	タイムアウト付メッセージ待ち状態 <sup>2</sup>
WAI(MBX-TMO)-SUS	タイムアウト付メッセージ待ち状態 (二重待ち)
NON	未登録状態

flg\_mode 領域の表示内容には、以下の種類があります。

flg_mode	状態
TWF_ANDW	待ちビットパターンで設定されているビットの全てが、イベントフラグにセットされるのを待ちます (AND 待ち)。
TWF_ANDW+ TWF_CLR	AND 待ちが発生し、タスクが待ち解除になった場合に、イベントフラグの値を 0 クリアします。
TWF_ORW	待ちビットパターンで設定されているビットのいずれかが、イベントフラグにセットされるのを待ちます (OR 待ち)。
TWF_ORW+ TWF_CLR	OR 待ちが発生し、タスクが待ち解除になった場合に、イベントフラグの値を 0 クリアします。

各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

TSK モードでは、水平スクロールバーは表示されません。

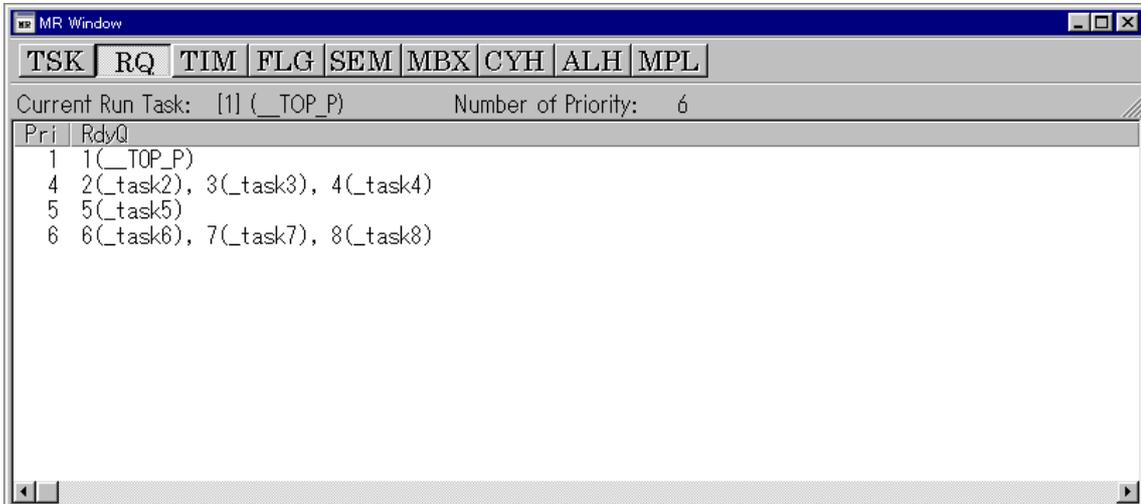
TSK モードの場合、リアルタイム OS 状態表示領域をマウスでダブルクリックすると、Context ダイアログが表示されます。この時 Context ダイアログには、ダブルクリックした行のタスクのコンテキスト情報が表示されています。

Context ダイアログの詳細については、「タスクのコンテキストを表示するには…」を参照してください。

レディキューの状態を表示するには・・・

MR ウィンドウのツールバーから“RQ” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Ready Q ] を選択してください。

以下に MR ウィンドウのレディキュー状態表示モードの構成図を示します。



RQ モードでは、以下に示した項目を表示します。コンフィグレーションで定義された優先度のうちレディキューにタスクがつながっているものについてのみ、優先度の高い順に表示します。

Pri	優先度を表示します。
RdyQ	レディキューに並んでいるタスク ID 番号とタスク名を表示します。

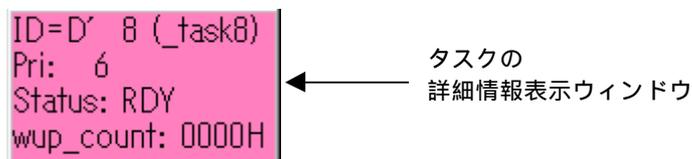
RQ モードの場合、ステータスバーの表示内容は次のようになります。

**Current Run Task**----- 現在実行中のタスク ID とタスク名  
(アイドル状態の時は“nothing...” と表示します。)

**Number of Priority**----- 最大優先度数

RdyQ 領域に表示されるタスク名の表示文字数は、最大 8 文字までです。タスク名が 8 文字を超える場合、それ以降は省略されます。

RdyQ 領域に表示されたデータ位置にマウスを移動すると、以下の例のようなウィンドウがオープンし、そのタスクに関する情報 (内容は TSK モードの情報と同じ) を表示します。



各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

RQ モードでは、データが表示されると同時に水平スクロールバーが表示されます。水平スクロールバーで、RdyQ 領域のデータ内容をスクロールすることができます。

### タイムアウトキューの状態を表示するには・・・

MR ウィンドウのツールバーから “TIM” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Timeout Q ] を選択してください。

以下に MR ウィンドウのタイムアウトキュー状態表示モードの構成図を示します。

Value	ID(name)
000058BAH	21(_task21)[dly]
000066BBH	24(_task24)[dly]
000066CAH	22(_task22)[dly]
000067ABH	23(_task23)[dly]
000067ACH	32(_task32)[mbx][s]
000067ACH	31(_task31)[mbx]
000067ACH	30(_task30)[sem][s]
000067ACH	29(_task29)[sem]
000067ACH	28(_task28)[flg][s]
000067ACH	27(_task27)[flg]
000067ACH	26(_task26)[slp][s]
000067ACH	25(_task25)[slp]
000067BAH	20(_task20)[dly]
000067BAH	7(_task7)[dly][s]
000067BAH	6(_task6)[dly]

TIM モードでは、以下に示した項目を表示します。現時点で時間待ち状態になっているタスクについて、タイムアウト値の小さい順に表示します。

Value	各タスクの現時点からのタイムアウト値を表示します。
ID(name)	タイムアウトキューに並んでいるタスク ID 番号とタスク名、および待ち状態の種類を表示します。

なお、待ち状態の種類を示す文字列には、以下の種類があります。

文字列	待ち状態
[slp]	tslp_tsk による待ち。
[dly]	dly_tsk による待ち。
[flg]	twai_flg による待ち。
[sem]	twai_sem による待ち。
[mbx]	trcv_msg による待ち。

TIM モードの場合、ステータスバーは表示されません。

タイムアウトキューにつながったタスクが さらに強制待ち状態 (二重待ち状態) の場合は、ID(name)領域に表示される文字列の後ろに二重待ち状態を示す文字列 “ [s] ” が付加されます。

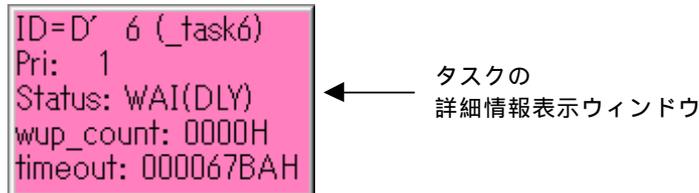
普通の場合の表示

**26(\_task26)[slp]**

二重待ち状態の場合の表示

**26(\_task26)[slp][s]**

ID(name)領域に表示されたデータ位置にマウスを移動すると、以下の例のようなウィンドウがオープンし、そのタスクに関する情報(内容はTSKモードの情報と同じ)を表示します。

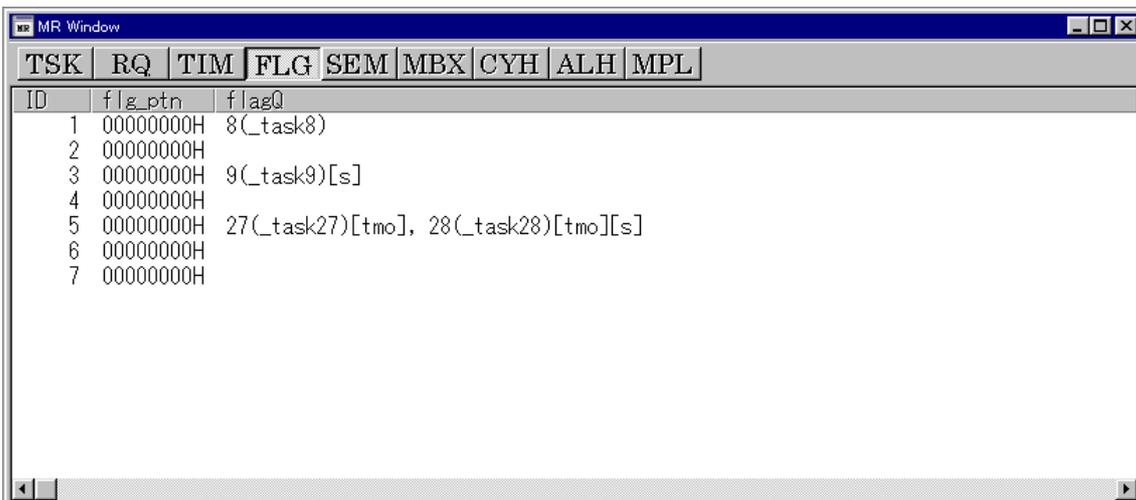


各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

TIMモードでは、水平スクロールバーは表示されません。

**イベントフラグの状態を表示するには・・・**

MR ウィンドウのツールバーから“FLG”ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Event Flag ] を選択してください。  
 以下にMR ウィンドウのイベントフラグ状態表示モードの構成図を示します。



FLGモードでは、以下に示した項目を表示します。コンフィグレーションで定義された全てのイベントフラグについて、ID番号順に表示します。

ID	イベントフラグのID番号を表示します。
flg_ptn	イベントフラグの値(ビットパターン)を表示します。
flagQ	イベントフラグキューに並んでいるタスクID番号とタスク名を表示します。

FLGモードの場合、ステータスバーは表示されません。

イベントフラグキューにつながったタスクが さらにタイムアウト有りの待ち状態 (twai\_flg による待ち状態) の場合は、flagQ 領域に表示される文字列の後ろにタイムアウト有りの待ち状態を示す文字列 “[tmo]” が付加されます。

普通の場合の表示

**27(\_task27)**

タイムアウト有りの待ち状態の場合の表示

**27(\_task27)[tmo]**

イベントフラグキューにつながったタスクが さらに強制待ち状態 (二重待ち状態) の場合は、flagQ 領域に表示される文字列の後ろに二重待ち状態を示す文字列 “[s]” が付加されます。

普通の場合の表示

**28(\_task28)**

二重待ち状態の場合の表示

**28(\_task28)[s]**

タイムアウト有りの待ち状態 + 二重待ち状態の場合の表示

**28(\_task28)[tmo][s]**

flagQ 領域に表示されるタスク名の表示文字数は、最大 8 文字までです。タスク名が 8 文字を超える場合、それ以降は省略されます。

flagQ 領域に表示されたデータ位置にマウスを移動すると、以下の例のようなウィンドウがオープンし、そのタスクに関する情報 (内容は TSK モードの情報と同じ) を表示します。

```
ID=D' 8 (_task8)
Pri: 1
Status: WAI(FLG)
wup_count: 0000H
wait_flg_ptn: 00001111H
wait_flg_mode: TWf_ORw
```

← タスクの  
詳細情報表示ウィンドウ

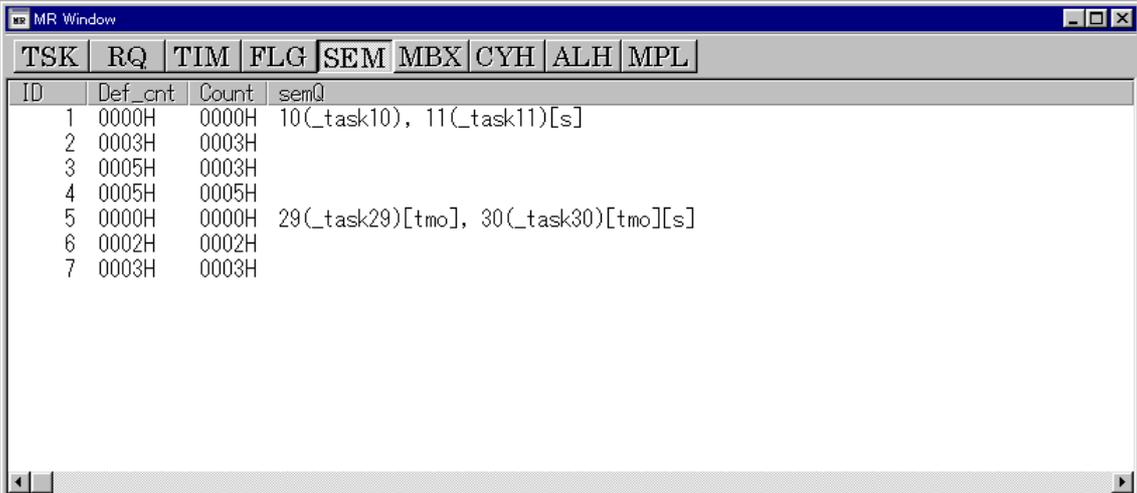
各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

FLG モードでは、データが表示されると同時に水平スクロールバーが表示されます。水平スクロールバーで、flagQ 領域のデータ内容をスクロールすることができます。

### セマフォの状態を表示するには・・・

MR ウィンドウのツールバーから “SEM” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Semaphore ] を選択してください。

以下に MR ウィンドウのセマフォ状態表示モードの構成図を示します。



ID	Def_cnt	Count	semQ
1	0000H	0000H	10(_task10), 11(_task11)[s]
2	0003H	0003H	
3	0005H	0003H	
4	0005H	0005H	
5	0000H	0000H	29(_task29)[tmo], 30(_task30)[tmo][s]
6	0002H	0002H	
7	0003H	0003H	

SEM モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全

でのセマフォについて、ID 番号順に表示します。

ID	セマフォの ID 番号を表示します。
Def_cnt	セマフォカウンタの初期値を表示します。
Count	現時点のセマフォカウンタを表示します。
semQ	セマフォキューに並んでいるタスク ID 番号とタスク名を表示します。

SEM モードの場合、ステータスバーは表示されません。

セマフォキューにつながったタスクが さらにタイムアウト有りの待ち状態( twai\_sem による待ち状態 ) の場合は、semQ 領域に表示される文字列の後ろにタイムアウト有りの待ち状態を示す文字列 “ [tmo] ” が付加されます。

普通の場合の表示

**29(\_task29)**

タイムアウト有りの待ち状態の場合の表示

**29(\_task29)[tmo]**

セマフォキューにつながったタスクが さらに強制待ち状態 ( 二重待ち状態 ) の場合は、semQ 領域に表示される文字列の後ろに二重待ち状態を示す文字列 “ [s] ” が付加されます。

普通の場合の表示

**30(\_task30)**

二重待ち状態の場合の表示

**30(\_task30)[s]**

タイムアウト有りの待ち状態 + 二重待ち状態の場合の表示

**30(\_task30)[tmo][s]**

semQ 領域に表示されるタスク名の表示文字数は、最大 8 文字までです。タスク名が 8 文字を超える場合、それ以降は省略されます。

semQ 領域に表示されたデータ位置にマウスを移動すると、以下の例のようなウィンドウがオープンし、そのタスクに関する情報 ( 内容は TSK モードの情報と同じ ) を表示します。

```
ID=D' 11 (_task11)
Pri: 1
Status: WAI(SEM)-SUS
wup_count: 0000H
```

← タスクの  
詳細情報表示ウィンドウ

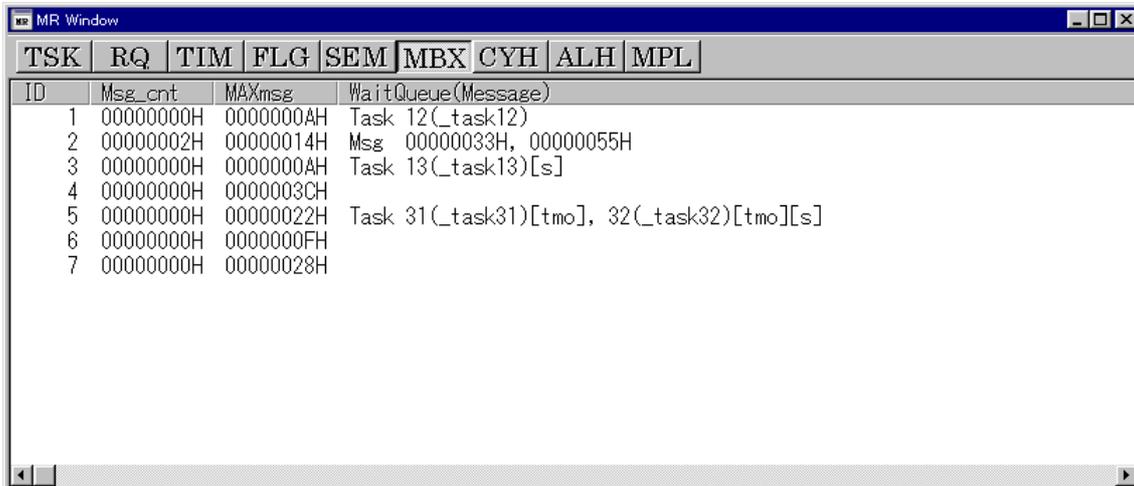
各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

SEM モードでは、データが表示されると同時に水平スクロールバーが表示されます。水平スクロールバーで、semQ 領域のデータ内容がスクロールできます。

### メールボックスの状態を表示するには・・・

MR ウィンドウのツールバーから“MBX”ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Mailbox ] を選択してください。

以下に MR ウィンドウのメールボックス状態表示モードの構成図を示します。



MBX モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全てのメールボックスについて、ID 番号順に表示します。

ID	メールボックスの ID 番号を表示します。
Msg_cnt	メールボックスに格納されているメッセージ数を表示します。
MAXmsg	メールボックスに格納可能なメッセージ数を表示します。
Wait Queue(Message)	メールボックスに格納されているメッセージ、またはメッセージ待ちのタスク ID 番号とタスク名を表示します。

MBX モードの場合、ステータスバーは表示されません。

WaitQueue ( Message ) 領域の表示内容は、次のようになります。

- メッセージが格納されている場合（上記の Msg\_cnt が 0 以外の場合）には、文字列 “Msg” を表示し、続いて格納されているメッセージを表示します。
- メッセージが格納されていない場合（上記の Msg\_cnt が 0 の場合）で、メッセージ待ちのタスクが存在している場合には、文字列 “Task” を表示し、続いてメッセージ待ちのタスク ID 番号とタスク名を表示します。

メールボックスキューにつながったタスクが さらにタイムアウト有りの待ち状態（trcv\_msg による待ち状態）の場合は、WaitQueue ( Message ) 領域に表示される文字列の後ろにタイムアウト有りの待ち状態を示す文字列 “[tmo]” が付加されます。

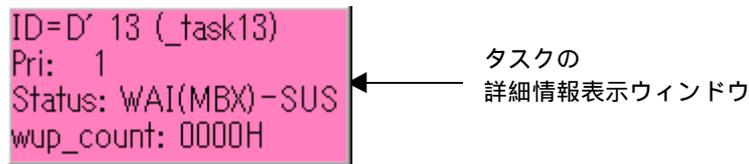
普通の場合の表示	<b>31(_task31)</b>
タイムアウト有りの待ち状態の場合の表示	<b>31(_task31)[tmo]</b>

メールボックスキューにつながったタスクが さらに強制待ち状態（二重待ち状態）の場合は、 WaitQueue ( Message ) 領域に表示される文字列の後ろに二重待ち状態を示す文字列 “ [s] ” が付加されます。

普通の場合の表示	<b>32(_task32)</b>
二重待ち状態の場合の表示	<b>32(_task32)[s]</b>
タイムアウト有りの待ち状態 + 二重待ち状態の場合の表示	<b>32(_task32)[tmo][s]</b>

WaitQueue ( Message ) 領域に表示されるタスク名の表示文字数は、最大 8 文字までです。タスク名が 8 文字を超える場合、それ以降は省略されます。

WaitQueue ( Message ) 領域に表示されたデータ位置にマウスを移動すると、以下の例のようなウィンドウがオープンし、そのタスクに関する情報（内容は TSK モードの情報と同じ）を表示します。



各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

MBX モードでは、データが表示されると同時に水平スクロールバーが表示されます。水平スクロールバーで、 WaitQueue ( Message ) 領域のデータ内容をスクロールすることができます。

**周期起動ハンドラの状態を表示するには・・・**

MR ウィンドウのツールバーから “ CYH ” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Cyclic Handler ] を選択してください。

以下に MR ウィンドウの周期起動ハンドラ状態表示モードの構成図を示します。

ID	StaAddr	(name)	interval	count	Status
1	00066A6CH	(_cycle1)	00000064H	00000056H	TCY_ON
2	00066A70H	(_cycle2)	000003E8H	00000312H	TCY_OFF
3	00066A74H	(_cycle3)	000001F4H	0000011EH	TCY_ON
4	00066A78H	(_cycle4)	00000258H	00000182H	TCY_ON
5	00066A7CH	(_cycle5)	000000C8H	000000BAH	TCY_OFF
6	00066A80H	(_cycle6)	0000012CH	00000056H	TCY_ON
7	00066A84H	(_cycle7)	00000190H	000000BAH	TCY_ON
8	00066A88H	(_cycle8)	0000015EH	00000056H	TCY_ON

CYH モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全ての周期起動ハンドラについて、ID 番号順に表示します。

ID	周期起動ハンドラの ID 番号を表示します。
StaAddr	周期起動ハンドラの開始アドレスを表示します。
(name)	周期起動ハンドラ名を表示します。
interval	周期起動ハンドラの周期起動間隔を表示します。
count	周期起動ハンドラが次に起動するまでの割り込み回数 (残数) を表示します。
Status	周期起動ハンドラの活性状態を表示します。

CYH モードの場合、ステータスバーは表示されません。

Status 領域の表示内容には、以下の種類があります。

TCY_ON	周期起動ハンドラが有効です。
TCY_OFF	周期起動ハンドラが無効です。

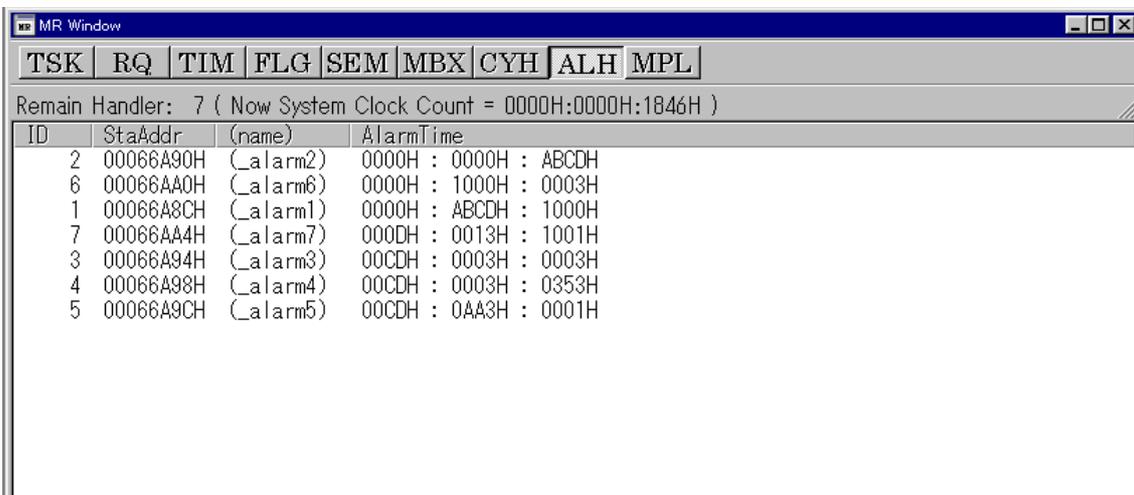
各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

CYH モードでは、水平スクロールバーは表示されません。

### アラームハンドラの状態を表示するには・・・

MR ウィンドウのツールバーから “ALH” ボタンをクリック、またはメニュー [ Option ] [ Mode ] [ Alarm Handler ] を選択してください。

以下に MR ウィンドウのアラームハンドラ状態表示モードの構成図を示します。



ALH モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全てのアラームハンドラのうち現時点で未起動のものについてのみ、起動時刻の早い順に表示します。

ID	アラームハンドラの ID 番号を表示します。
StaAddr	アラームハンドラの開始アドレスを表示します。
(name)	アラームハンドラ名を表示します。
Alarm Time	アラームハンドラの起動時刻を表示します。

ALH モードの場合、ステータスバーの表示内容は次のようになります。

**Remain Handler** 起動待ちのアラームハンドラ数、および 現在のシステムクロック  
カウント

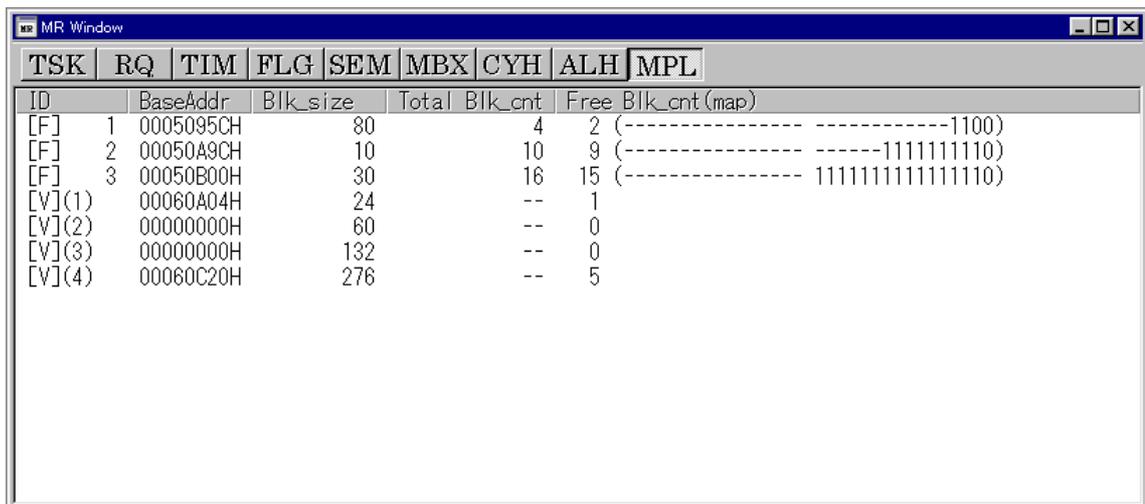
各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

ALH モードでは、水平スクロールバーは表示されません。

**メモリプールの状態を表示するには・・・**

MR ウィンドウのツールバーから “MPL” ボタンをクリック、またはメニュー [ Option ]  
[ Mode ] [ Memory Pool ] を選択してください。

以下にMR ウィンドウのメモリプール状態表示モードの構成図を示します。



MPL モードでは、以下に示した項目を表示します。コンフィグレーションで定義された全  
てのメモリプールについて、（固定長・任意長の順で）ID 番号順に表示します。

ID	メモリプールの ID 番号を表示します。
BaseAddr	メモリプールのベースアドレスを表示します。
Blk_size	メモリプールのブロックサイズを表示します。
Total Blk_cnt	メモリプールの全ブロック数を表示します。
Free Blk_cnt(map)	未使用のブロック数、およびメモリブロック情報（ビット情報）を表示します。

MPL モードの場合、ステータスバーは表示されません。

ID 領域は、固定長・任意長の違いにより以下のように表示内容が異なります。

- 固定長の場合、文字列“[F]”とメモリプールの ID 番号を表示します。
- 任意長の場合、文字列“[V]”とブロック ID 番号を表示します。この場合、ブロック ID はカッコ（ “ ( ) ” ）で囲んで表示します。

任意長メモリーブールの場合、Total Blk\_cnt 領域には何も表示されません（“--”文字列を表示します）。

任意長メモリーブールの場合、Free Blk\_cnt(map)領域のビット情報は表示されません。

また固定長メモリーブールの場合、Free Blk\_cnt(map)領域のメモリブロック情報の各ビットの表示形式は次のようになります。

'0'	メモリブロック使用不可（使用中）。
'1'	メモリブロック使用可能（未使用）。
'-'	もともとメモリブロックが存在しない。

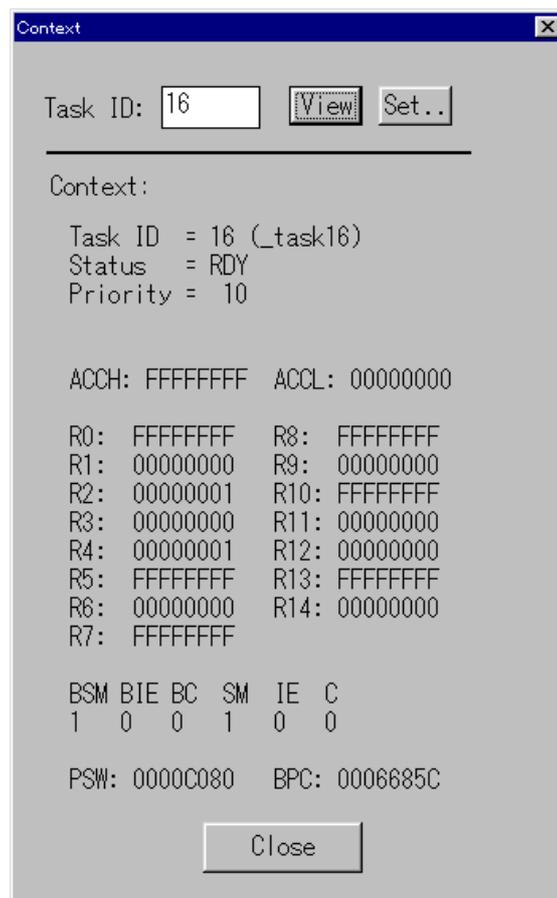
各表示領域は、マウスのドラッグ操作により表示幅を変更することができます。

MPL モードでは、水平スクロールバーは表示されません。

### タスクのコンテキストを表示するには・・・

メニュー [ Option ] [ MR ] [ Context ] を選択、または TSK モードのデータ表示部分をマウスでダブルクリックすると、Context ダイアログがオープンします。Context ダイアログは、指定タスクのコンテキスト情報を参照・設定するためのダイアログです。

以下に Context ダイアログの構成図を示します。



Task ID:領域 にタスク ID 番号を入力し、<View>ボタンをクリック (またはEnter キー入力) すると、Context:領域 に指定タスクのコンテキストが表示されます。

<View>ボタンクリック時、Task ID:領域に入力したタスクが“RUN”または“DMT”状態の場合は、コンテキストは表示されません (Context: 領域には、タスク ID とタスクの状態のみが表示されます)。

<View>ボタンクリック時、Task ID:領域 に存在しないタスク ID 番号を入力した場合は、エラーダイアログが表示されます。

Task ID:領域 にタスク ID 番号を入力し、<Set..>ボタンをクリックすると、Set Context ダイアログが表示されます。Set Context ダイアログでは 指定タスクのコンテキスト値を設定できます。Set Context ダイアログを閉じると、Set Context ダイアログで設定した値を反映したコンテキスト情報が Context:領域に表示されます。Set Context ダイアログの詳細については、「タスクのコンテキスト内容を変更するには...」を参照してください。

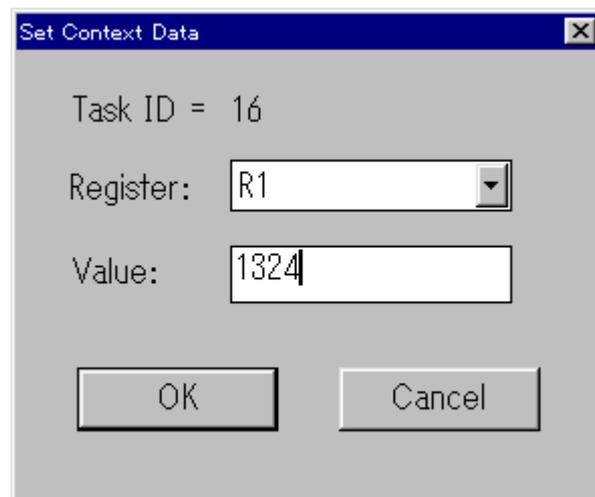
<Set..>ボタンクリック時、Task ID:領域に入力したタスクが“RUN”または“DMT”状態の場合、または存在しないタスク ID 番号を指定した場合は、エラーダイアログが表示されます。

<Close>ボタンをクリックすると、Context ダイアログはクローズします。

#### タスクのコンテキスト内容を変更するには・・・

Context ダイアログの Task ID:領域 にタスク ID 番号を入力し<Set..>ボタンをクリックすると、Set Context ダイアログがオープンします。Set Context ダイアログは、指定タスクの指定コンテキストレジスタ値を設定するためのダイアログです。

以下に Set Context ダイアログの構成図を示します。



---

Task ID:領域 には、Context ダイアログの Task ID:領域で入力したタスク ID 番号が、本ダイアログオープン時に既に表示されています。

Register:領域に、値を設定するレジスタ名を指定します。コンボボックスのリストには、指定タスクが保持するコンテキストレジスタ名が表示されますので、この中から選択してください。

Value:領域に、コンテキストレジスタに設定する値を入力します。

各領域に必要な事項を設定の上 <OK>ボタンをクリック（またはEnter キー入力）すると、指定タスクの指定コンテキストレジスタに 指定したデータが設定され、同時に Set Context ダイアログはクローズします。

Value:領域に設定した式の記述に誤りがあった場合、指定レジスタに設定できる値の範囲を超えた場合などには、エラーダイアログが表示されます。

<Cancel>ボタンをクリックすると、Set Context ダイアログの各領域に設定されたデータは無効となり、Set Context ダイアログはクローズします。

【 MEMO 】

# リファレンス編



# 1 スクリプトコマンド一覧

## 1.1 入力書式

- PD32000のスクリプトコマンドの入力形式
  1. コマンド名とパラメータの間には、1文字以上の空白文字またはタブが必要です。
  2. コマンド名は、大小の英数字が使用できます。
- コマンド書式の記号説明

パラメータ	コマンド書式
XXXX	XXXXを入力する必要があります。
[XXXX]	入力を省略するか、又はXXXXが入力可能です。
{ X1   X2   X3 }	X1,X2,X3の内、いずれか1つを入力する必要があります。
[{ X1   X2   X3 }]	入力を省略するか、X1,X2,X3の内、いずれか1つが入力可能です。

## 1.2 コマンド一覧

以下にコマンド一覧を示します。網掛けがされているコマンドはランタイム実行可能です。コマンド名欄の()内はコマンド名の省略形です。

なお、各コマンドの詳細な説明は、PD32000のヘルプをご参照下さい。ヘルプを表示するには、PD32000ウィンドウのメニュー  
[ Help ] [ Index ]  
を選択してください。

### 1.2.1 実行関連

コマンド名	書式	機能概要
Go(G)	Go [ 開始アドレス ]	ターゲットプログラムを実行します。
GoFree(GF)	GoFree [ 開始アドレス ]	ターゲットプログラムをフリーラン実行します。
ResetGo(RG)	ResetGo	ターゲット MCU のリセット検知後、ターゲットプログラムを実行します。
STOP	STOP	ターゲットプログラムを停止します。
STATUS	STATUS	ターゲットプログラムの実行状態を表示します。
Step(S)	Step [ 実行回数 ]	ソースレベルでステップ実行します。
StepInstruction(SI)	StepInstruction [ 実行回数 ]	機械語レベルでステップ実行します。
OverStep(O)	OverStep [ 実行回数 ]	ソースレベルでオーバーステップ実行します。
OverStepInstruction(OI)	OverStepInstruction [ 実行回数 ]	機械語レベルでオーバーステップ実行します。
RETurn(RET)	RETurn	ソースレベルでリターン実行します。
RETurnInstruction(RET I)	RETurnInstruction	機械語レベルでリターン実行します。
RESET	RESET	ターゲットプログラムをリセットします。

### 1.2.2 ファイル操作関連

コマンド名	書式	機能概要
Load(L)	Load ファイル名[ .x, .exe, .abs ]	ロードモジュールファイルをダウンロードします。
LoadMot(LM)	LoadMot ファイル名[ .mot ]	S3 ファイルをダウンロードします。
LoadBin(LB)	LoadBin ファイル名[ .bin ]	バイナリファイルをダウンロードします。
LoadSymbol(LS)	LoadSymbol ファイル名[ .x, .exe, .abs ]	デバッグ情報のみをダウンロードします。
UploadBin(UB)	UploadBin 開始アドレス, 終了アドレス, ファイル名	指定範囲のメモリ内容をバイナリファイルに出力します。
UploadMot(UM)	UploadMot 開始アドレス, 終了アドレス, ファイル名	指定範囲のメモリ内容を mot ファイルに出力します。

### 1.2.3 レジスタ操作関連

コマンド名	書式	機能概要
Register(R)	Register [ レジスタ名 ] Register レジスタ名, 設定値	指定レジスタの値を参照します。 指定レジスタに値を設定します。

## 1.2.4 メモリ操作関連

コマンド名	書式	機能概要
DumpByte(DB)	DumpByte [ 開始アドレス [ , 終了アドレス ] ]	メモリ内容を1バイト単位で表示します。
DumpHword(DH)	DumpHword [ 開始アドレス [ , 終了アドレス ] ]	メモリ内容を2バイト単位で表示します。
DumpWord(DW)	DumpWord [ 開始アドレス [ , 終了アドレス ] ]	メモリ内容を4バイト単位で表示します。
setMemoryByte(MB)	setMemoryByte アドレス [ , データ [ , ... ] ]	指定アドレスのメモリ内容を1バイト単位で変更します。終了するには、“.”を入力してください。
setMemoryHword(MH)	setMemoryHword アドレス [ , データ [ , ... ] ]	指定アドレスのメモリ内容を2バイト単位で変更します。終了するには、“.”を入力してください。
setMemoryWord(MW)	setMemoryWord アドレス [ , データ [ , ... ] ]	指定アドレスのメモリ内容を4バイト単位で変更します。終了するには、“.”を入力してください。
FillByte(FB)	FillByte 開始アドレス, 終了アドレス, データ	指定したアドレス範囲に指定したデータを1バイト単位で書き込みます。
FillHword(FH)	FillHword 開始アドレス, 終了アドレス, データ	指定したアドレス範囲に指定したデータを2バイト単位で書き込みます。
FillWord(FW)	FillWord 開始アドレス, 終了アドレス, データ	指定したアドレス範囲に指定したデータを4バイト単位で書き込みます。
MOVeByte(MOVB)	MOVeByte 開始アドレス, 終了アドレス, 転送先アドレス	指定範囲のメモリ内容を転送先アドレス以降に1バイト単位で転送します。
MOVeHword(MOVH)	MOVeHword 開始アドレス, 終了アドレス, 転送先アドレス	指定範囲のメモリ内容を転送先アドレス以降に2バイト単位で転送します。
MOVeWord(MOVW)	MOVeWord 開始アドレス, 終了アドレス, 転送先アドレス	指定範囲のメモリ内容を転送先アドレス以降に4バイト単位で転送します。

## 1.2.5 アセンブル/逆アセンブル関連

コマンド名	書式	機能概要
Assemble(A)	Assemble [ アドレス ]	指定したアドレスから1行単位でアセンブルします。
DisAssemble(DA)	DisAssemble [ 開始アドレス [ , 終了アドレス ] ]	指定した範囲の逆アセンブル結果を表示します。
MODuLe(MOD)	MODuLe	全モジュール(オブジェクト名)を表示します。
SCOPE	SCOPE SCOPE モジュール名	現在のスコープを表示します。 指定モジュールにスコープを設定します。
SECTion(SEC)	SECTion	セクション情報を表示します。
SYMBOL(SYM)	SYMBOL SYMBOL { GLOBAL   G } [ , シンボル名 ] SYMBOL { LOCAL   L } [ , シンボル名 ]	全シンボルを表示します。 グローバルシンボルを表示します。 ローカルシンボルを表示します。
LABel(LAB)	LABel LABel { GLOBAL   G } [ , ラベル名 ] LABel { LOCAL   L } [ , ラベル名 ]	全ラベルを表示します。 グローバルラベルを表示します。 ローカルラベルを表示します。
EXPRESS(EXP)	EXPRESS アセンブラ式	指定したアセンブラ式の値を表示します。

## 1.2.6 ソフトウェアブレイク設定関連

コマンド名	書式	機能概要
SoftwareBreak(SB)	SoftwareBreak SoftwareBreak アドレス	現在設定されているソフトウェアブレイクポイントを表示します。 指定したアドレスにソフトウェアブレイクポイントを設定します。
SoftwareBreakClear(SBC)	SoftwareBreakClear アドレス	指定アドレスのソフトウェアブレイクポイントを削除します。
SoftwareBreakClearAll(SBCA)	SoftwareBreakClearAll	全ソフトウェアブレイクポイントを削除します。
SoftwareBreakDisable(SBD)	SoftwareBreakDisable アドレス	指定アドレスのソフトウェアブレイクポイントを無効にします。
SoftwareBreakDisableAll(SBDA)	SoftwareBreakDisableAll	全ソフトウェアブレイクポイントを無効にします。
SoftwareBreakEnable(SBE)	SoftwareBreakEnable アドレス	指定アドレスのソフトウェアブレイクポイントを有効にします。
SoftwareBreakEnableAll(SBEA)	SoftwareBreakEnableAll	全ソフトウェアブレイクポイントを有効にします。
BREAKAT	BREAKAT 行番号 [, ソースファイル名]	指定した行番号にソフトウェアブレイクポイントを設定します。
BREAKIN	BREAKIN 関数名 [, モジュール名]	指定した関数の先頭にソフトウェアブレイクポイントを設定します。

## 1.2.7 トレース関連

コマンド名	書式	機能概要
TracePoint(TP)	TracePoint TracePoint {0~7[:{0~3}] Filter Reset} TracePoint {0~7[:{0~3}] Filter Reset}, Addr [, アドレス(8桁)[:アドレスマスク(8桁)] TracePoint {0~7[:{0~3}] Filter Reset}, Data [, アドレス(8桁)[:アドレスマスク(8桁)] [, 比較データ(4桁)[:データマスク(4桁)] [, {Read WriteWord WriteByte}]]] TracePoint {0~7[:{0~3}] Filter Reset}, Signal [, アドレス(6桁)[:アドレスマスク(6桁)] [, 比較データ(4桁)[:データマスク(4桁)] [, 信号線]]] TracePoint {0~7[:{0~3}] Filter Reset}, {ON OFF} TracePoint Mode, {No Pc Signal} TracePoint Sample, {Clock Bus} TracePoint Break, {ON OFF} TracePoint Size, {1~256K}	トレースポイントを参照します。 ~ トレースポイントを設定します。 ~ トレースポイントのモードを設定します。
TraceData(TD)	TraceData [開始サイクル] [, 終了サイクル]	トレースメモリの内容をバス信号形式で表示します。表示内容は、トレースウィンドウのバス表示モードと同じです。(SIGNALモード時)
TraceList(TL)	TraceList [開始サイクル] [, 終了サイクル]	トレースメモリの内容を逆アセンブリリスト形式で表示します。(PCモード時)



### 1.2.12 リアルタイム OS 関連

コマンド名	書式	機能概要
MR	MR [TSK] MR { RQ   R } MR { TIM   TOUT   T } MR { FLG   FLAG   F } MR { SEM   S } MR { MBX   M } MR { CYH   CYC   C } MR { ALH   ALM   A } MR { RTSK   RT } MR { STIM   ST } MR { CTX   CT }, タスク番号  MR MPL	指定したタスクの状態を表示します。 レディーキューの状態を表示します。 タイムアウトキューの状態を表示します。 イベントフラグの状態を表示します。 セマフォの状態を表示します。 メールボックスの状態を表示します。 周期ハンドラの状態を表示します。 アラームハンドラの状態を表示します。 実行中のタスクを表示します。 システムクロックカウントを表示します。 指定されたタスク番号のタスクのコンテキストを表示します。 メモリプールを表示します。

### 1.2.13 タイマ関連

コマンド名	書式	機能概要
TimerSet(TS)	TimerSet TimerSet {Start End StartEnd}, アドレス TimerSet Clear, アドレス TimerSet Result	タイマの状態を表示します。 タイマを設定します。 タイマを削除します。 タイマの測定結果を表示します。

### 1.2.14 ユーティリティ関連

コマンド名	書式	機能概要
RADIX	RADIX RADIX { 2   8   10   16 }	定数入力の既定値を参照します。 定数入力の既定値を設定します。
ALIAS	ALIAS ALIAS 別名 , コマンド名	コマンドの別名定義を参照します。 コマンドに別名を定義します。
UNALIAS	UNALIAS 別名	指定した別名定義を削除します。
UNALIASALL	UNALIASALL	すべての別名定義を削除します。
HELP(H)	HELP HELP コマンド名	スクリプトコマンド一覧を表示します。 指定したコマンドの説明を表示します。
VERsion(VER)	VERsion	PD32000 のバージョンを表示します。
DATE	DATE	現在の日時(年月日、曜日、時間)を表示します。
ECHO	ECHO “メッセージ”	指定したパラメータを表示します。
QUIT	QUIT	PD32000 を終了する。
CD	CD CD ディレクトリ名	現在のカレントディレクトリを参照します。 カレントディレクトリを設定します。
MACRO	MACRO  MACRO カスタムウィンドウ/コマンドのプログラムファイル名	登録しているカスタムウィンドウ/コマンド名を参照します。 カスタムウィンドウ/コマンド名を登録します。
DELMACRO	DELMACRO カスタムウィンドウ/コマンドのプログラムファイル名	指定したカスタムウィンドウ/コマンドの登録を削除します。
DELMACROALL	DELMACROALL	カスタムウィンドウ/コマンドの登録をすべて削除します。
MacroPATH (MPATH)	MacroPATH [ サーチパス [ ;サーチパス ;.... ] ]	カスタムウィンドウ/コマンドのプログラムファイルが存在するパスを設定します。

## 2 スクリプトファイルの記述方法

PD32000では、スクリプトウィンドウでスクリプトファイルを実行することができます。スクリプトファイルは、スクリプトコマンドを自動実行するために、その制御などを記述したファイルです。

### 2.1 スクリプトファイルの構成要素

スクリプトファイルに以下の文が記述できます。

- スクリプトコマンド
- 代入文
- 判断文 (if, else, endi)  
式の結果を判断して、実行する文を分岐します。
- 繰り返し文 (while, endw)  
式の結果を判断して、文を繰り返し実行します。
- break 文  
最も内側の繰り返し実行から抜けます。
- コメント文  
スクリプトファイルにコメント(注釈)を記述できます。スクリプトコマンド実行の際、コメント文は無視されます。

スクリプトファイルには、一行につき1文を記述してください。一行に複数の文を記述したり、1つの文を複数行にまたがって記述することはできません。

#### 2.1.1 スクリプトコマンド

スクリプトウィンドウで入力するコマンドを、そのまま記述することができます。またスクリプトファイルからスクリプトファイルを呼び出すこともできます(ネストは5段まで)。

#### 2.1.2 代入文

代入文は、マクロ変数の定義や初期化、および代入を行います。以下に記述書式を示します。

**%マクロ変数名 = 式**

- マクロ変数名には、英数字と'\_'が使用できます。ただし、マクロ変数名の先頭には、数字を記述することはできません。
- マクロ変数に代入する式が扱える値の範囲は、 $0_{16}$  から  $FFFFFFFF_{16}$  までの整数です。負の数を指定した場合は2の補数として扱います。
- マクロ変数は、式の中で使用することができます。
- マクロ変数は、先頭に%を付加して使用します。

### 2.1.3 判断文 (if,endi,else)

判断文は、式の結果を判断し、実行する文を分岐します。以下に記述書式を示します。

```
if ( 式 )
  文 1
else
  文 2
endi
```

- 式が真 ( 0 以外 ) のとき文 1 を実行します。式が偽 ( 0 ) のとき文 2 を実行します。
- else 文は省略することができます。else 文を省略時に式が偽の場合、endi 文の次の行から実行します。
- if 文は、32 段までネストすることができます。

### 2.1.4 繰り返し文 (while,endw)と break 文

繰り返し文は、式の結果を判断し、文を繰り返し実行します。以下に記述書式を示します。

```
while ( 式 )
  文
endw
```

- 式が真の場合、文を繰り返し実行します。式が偽の場合、ループから抜けます (endw の次の文から実行します)。
- while 文は、32 段までネストすることができます。
- while 文を強制的に抜ける場合は、break 文を使用します。while 文がネストしている場合は、最も内側のループから抜けます。

### 2.1.5 コメント文

コメント文は、スクリプトファイルにコメント (注釈) を記述する場合に使用します。以下に記述書式を示します。

```
; 文字列
```

- セミコロン ( ‘ ; ’ ) から文を記述します。セミコロンの前には、空白文字とタブのみ記述可能です。
- コメント文の行は、スクリプトファイル実行時に無視されます。

## 注意事項

- スクリプトコマンドのコメントとして同一行に記述することはできません。
- スクリプトファイルのネストは5段までです。
- if 文と while 文のネストはそれぞれ32段までです。
- 一つのスクリプトファイルで if と endi 文、while と endw が対になっていなければいけません。
- スクリプトファイルに記述する式は、unsigned 型で計算します。したがって、if 文、while 文の式で負の値を比較した場合の動作は不定になります。
- 1行に記述できる文字数は、4096文字までです。これを越える行を実行した場合、エラーになります。
- 不適当な記述のあるスクリプトファイルを自動実行した場合（スクリプトウィンドウでスクリプトファイルをオープン後メニュー [Option] [Script] [Run]（またはスクリプトウィンドウの Run ボタン）を選択した場合）、スクリプト行自身を読み込めない場合を除いて、エラー検出後もスクリプトファイルの終わりまで実行処理は続けられます。  
ただしこの場合、エラー検出後の動作は不定であり、したがってエラー検出後の実行結果は信頼性がありません。

## 2.2 式の記述方法

PD32000では、アドレス、データ、通過回数などの指定に式を記述することができます。以下に式を使用したコマンド例を示します。

```
>DB TABLE1
>DB TABLE1+20
```

### 2.2.1 式の構成要素

式の構成要素として、以下のものが使用できます。

- 定数
- シンボル、ラベル
- マクロ変数
- レジスタ変数
- メモリ変数
- 行番号
- 文字定数
- 演算子

以下に、各構成要素について説明します。

### 2.2.2 定数

2進数、8進数、10進数、16進数が入力可能です。数値の基数は、数値の先頭または、末尾に基数を示す記号を付けて区別します。

	16進数	10進数	8進数	2進数 <sup>1</sup>
先頭	0x,0X	@	なし	%
末尾	h,H	なし	o,0	b,B
例	0xAB24 AB24h	@1234	1234o	%10010 10010b

<sup>1</sup> 基数の既定値が16進数のときは、'%'のみ指定可能

- 既定値と同じ基数で入力する場合は、基数を示す記号は省略可能です(2進数は除く)
- 基数の既定値は、RADIX コマンドで設定します。ただし、以下のデータに関する入力を行う場合は、RADIX コマンドの設定に関係なく、基数は固定です。

種別	基数
アドレス	16進
行番号 実行回数 通過回数 限定回数	10進

### 2.2.3 シンボル、ラベル

ターゲットプログラムで定義しているシンボル/ラベル、および Assemble コマンドで定義したシンボル/ラベルが使用できます。

- シンボル/ラベル名には、英数字、アンダスコア(‘\_’)、ピリオド(‘.’)、クエスチョンマーク(‘?’)が使用可能です。ただし、先頭文字に数字は使用できません。
- シンボル/ラベル名は、255 文字まで記述できます。
- 大文字/小文字は区別します。
- アセンブラ as32R の構造化命令、擬似命令、マクロ命令、オペコード予約語は使用できません(.SECTION, .BYTE, switch, if など)。
- “..”で始まる文字列は、シンボル/ラベル名には使用できません。

#### [ 参照 1 ] ローカルラベルシンボルとスコープ

PD32000では、CC32R を使用した場合、プログラムの全領域から参照可能なグローバルラベルシンボルと、宣言したファイル内でのみ参照可能なローカルラベルシンボルの 2 種類をサポートしています。

ローカルラベルシンボルの有効範囲をスコープといいます。スコープの単位は、オブジェクトファイル(“.o”ファイル)です。PD32000では、下記の場合に応じて、スコープを切り替えます。

- コマンド入力時  
プログラムカウンタが示すアドレスを含むオブジェクトファイルが、現在のスコープとなります。また SCOPE コマンドでスコープを設定した場合、設定したスコープが有効になります。
- コマンド実行中  
コマンドが扱うプログラムアドレスによって現在のスコープを自動的に切り替えます。

#### [ 参照 2 ] ラベル/シンボルの優先順位

値からラベル/シンボルへの変換、ラベル/シンボルから値への変換は、下記の優先順位で行います。

- アドレス値を変換する場合
  1. ローカルラベル
  2. グローバルラベル
  3. ローカルシンボル
  4. グローバルシンボル
  5. スコープ範囲外のローカルラベル
  6. スコープ範囲外のローカルシンボル
- データ値を変換する場合
  1. ローカルシンボル
  2. グローバルシンボル
  3. ローカルラベル
  4. グローバルラベル
  5. スコープ範囲外のローカルシンボル
  6. スコープ範囲外のローカルラベル

## 2.2.4 マクロ変数

マクロ変数は、スクリプトファイル中の代入文で定義します。詳細については、本マニュアルリファレンス編の項目「2.1 スクリプトファイルの構成要素」をご参照下さい。

マクロ変数は、変数名の先頭に`%`を付加して使用します。

- パーセント文字(`%`)の後の変数名には、英数字と`\_`が使用可能です。ただし、マクロ変数名の先頭には、数字を記述することはできません。
- 変数名には、レジスタ名は使用できません。
- 変数名の大文字 / 小文字を区別します。
- マクロ変数は、32 個まで定義できます。一度定義したマクロ変数は、PD32000が終了するまで有効です。

while 文の繰り返し回数を指定する際に、マクロ変数を利用すると便利です。

## 2.2.5 レジスタ変数

レジスタの値を式中で利用する場合に使用します。レジスタ変数は、レジスタ名の前に`%`を付加します。以下に記述形式を示します。

### %レジスタ名

使用できるレジスタ名を以下に示します。

PC, SPU, SPI, PSW, BPC, ACCH, ACCL,  
R0, R1, R2, R3, R4, R5, R6, R7,  
R8, R9, R10, R11, R12, R13, R14, R15

レジスタ名の大文字 / 小文字は区別しません。どちらで指定しても結果は同じです。

## 2.2.6 メモリ変数

メモリの値を式中で利用する際に使用します。メモリ変数の書式を以下に示します。

[アドレス].データサイズ

- アドレスには、式が記述できます（メモリ変数も指定可能）。
- データサイズは、以下のように指定します。

バイト長の場合	Bまたはb
ハーフワード(2バイト)長の場合	Hまたはh
ワード(4バイト)長の場合	Wまたはw

- データサイズの指定を省略した場合、ハーフワード長を指定したことになります。  
例 1 : 8000<sub>16</sub>番地のメモリ内容をバイトで参照する場合  
[8000h].B  
例 2 : 8000<sub>16</sub>番地のメモリ内容をハーフワードで参照する場合  
[8000h].H  
例 3 : 8000<sub>16</sub>番地のメモリ内容をワードで参照する場合  
[8000h].W

## 2.2.7 行番号

ソースファイルの行番号です。行番号の書式を以下に示します。

#行番号  
#行番号."ソースファイル名"

- 行番号は、10進数で指定します。
- 行番号に指定できるのは、ソフトウェアブレークが設定できる行だけです。コメント行や空白行などのアセンブラの命令が生成されない行を指定することはできません。
- ソースファイル名を省略した場合、プログラムウィンドウに表示しているソースファイルの行番号になります。
- ソースファイル名は、ファイル属性も指定してください。
- 行番号とソースファイル名の間には空白文字を挿入することはできません。

## 2.2.8 文字定数

指定された文字または文字列を ASCII コードに変換し、定数として扱います。

- 文字は、シングルクォーテーションで囲みます。
- 文字列は、ダブルクォーテーションで囲みます。
- 文字列は2文字以内（16ビット長）でなければなりません。2文字を越えた場合も、記述した文字列の最後の2文字が処理の対象となります。例えば、"ABCD" と記入した場合、文字列の最後の2文字 "CD" が処理対象となり、値は 4344<sub>16</sub> となります。

## 2.2.9 演算子

式に記述可能な演算子を以下に示します。

- 演算子の優先度は、レベル1が最も高く、レベル8が最も低くなります。優先順位が同じ場合は、式の左から順番に計算します。

演算子	意味	優先度
( )	括弧	レベル1
+, =, ~	単項正、単項負、単項論理否定	レベル2
*, /	二項乗算、二項除算	レベル3
+, -	二項加算、二項減算	レベル4
>>, <<	右シフト、左シフト	レベル5
&	二項論理積	レベル6
, ^	二項論理和、二項排他的論理和	レベル7
<, <=, >, >=, ==, !=	二項比較	レベル8

## 3 C 言語式について

### 3.1 C 言語式の記述方法

C ウォッチポイントの登録、および C ウォッチポイントに代入する値の指定には、以下の字句（トークン）で構成された式（C 言語式）が使用できます。

字句（トークン）	例
即値	10, 0x0a, 012
四則演算子	+, -, *, /
ポインタ	*, **, ...
参照	&
符号反転	-
コンマによるメンバ参照	Struct.Member
矢印によるメンバ参照	Struct->Member
括弧	(, )
配列	Array[2], DArray[2][3], ...
基本型へのキャスト	(int), (char*), (unsigned long *), ...
typedef された型へのキャスト	(DWORD), (ENUM), ...
変数名および関数名	var, i, j, func, ...
文字定数	'A', 'b', ...
文字列リテラル	"abcdef", "I am a boy.", ...

#### 3.1.1 即値

即値としては、16 進数、10 進数、および 8 進数を使用できます。0x で始めれば 16 進数、0 で始めれば 8 進数として認識します。それ以外の数値は、10 進数として認識します。

##### 注意事項

- 即値のみの C ウォッチポイントへの登録はできません。
- 即値は、C ウォッチポイントを指定する C 言語式の中に用いる場合、および代入する値を指定する場合にのみ有効です。

### 3.1.2 四則演算子

四則演算子は、加算 (+)、減算 (-)、乗算 (\*)、除算 (/) が使用できます。以下に、計算の優先順位を示します。

(\*)、(/) > (+)、(-)

#### 注意事項

- 浮動小数点に対する四則計算は、現在サポートしておりません。

### 3.1.3 ポインタ

ポインタは、\* で表され、ポインタのポインタ \*\*、ポインタのポインタのポインタ \*\*\*、... が使用できます。

「\*変数名」、「\*\*変数名」、... という記述で使います。

#### 注意事項

- 即値をポインタとして扱うことはできません。つまり、\*0xE000 などは、使用することができません。

### 3.1.4 参照

参照は、& で表され、「&変数名」のみが使用できます。

「&&変数名」等は使用することができません。

### 3.1.5 符号反転

符号反転は、- で表され、「-即値」、「-変数名」のみが使用できます。

- を 2 つ以上偶数個続けた場合には、符号反転は行なわれません。

#### 注意事項

- 浮動小数点に対する符号反転は、現在サポートしておりません。

### 3.1.6 コンマによるメンバ参照

コンマによる構造体、共用体のメンバ参照は、「変数名.メンバ名」のみが使用できます。

(例)

```
struct S{
    int    member1;
    char   member2;
};
```

```
struct S STRUCT;
struct S *STRUCT_P;
```

この場合、STRUCT.member1、(\*STRUCT\_P).member2 は、正しくメンバを参照することができます。

### 3.1.7 矢印によるメンバ参照

矢印による構造体、共用体のメンバ参照は、「変数名->メンバ名」のみが使用できます。  
(例)

```
struct S{
    int    member1;
    char   member2;
};

struct S STRUCT;
struct S *STRUCT_P;
```

この場合、(&STRUCT)->member1、STRUCT\_P->member2 は、正しくメンバを参照することができます。

### 3.1.8 括弧

式の途中に、計算の優先順位を指定する括弧として、「(」と「)」を使用することができます。

### 3.1.9 配列

配列の要素を指定する表現に「[」と「]」を使用することができます。  
配列は、「変数名[ (要素番号または変数) ]」、「変数名[ (要素番号または変数) ] [ (要素番号または変数) ]」、「・・・という記述で使します。

### 3.1.10 基本型へのキャスト

Cの基本型のうち、char型、short型、int型、long型へのキャスト、およびこれらの基本型へのポインタ型へのキャスト演算が使用できます。ポインタ型へのキャストは、ポインタのポインタ、ポインタのポインタのポインタ、・・・なども使用できます。

なお、signed、unsignedの指定がない場合のデフォルトは、以下のとおりです。

基本型	デフォルト
char	signed
short	signed
int	signed
long	signed

#### 注意事項

- Cの基本型のうち、浮動小数点型(float、double型)へのキャストは使用できません。
- レジスタ変数に対するキャストは使用できません。

### 3.1.11 typedef された型へのキャスト

typedef された型(Cの基本型以外の型)、およびそれらへのポインタ型へのキャスト演算が使用できます。ポインタ型へのキャストは、ポインタのポインタ、ポインタのポインタのポインタ、・・・なども使用できます。

#### 注意事項

- struct 型、union 型、およびそれらのポインタ型へのキャストは使用できません。

### 3.1.12 変数名および関数名

変数名および関数名は、Cの規約通りアルファベットで始まる文字列が使用できます。最大文字数は、255文字です。

### 3.1.13 文字定数

文字定数として、シングルクォーテーション ( ' ) で囲まれた文字が使用できます。例えば、 'A'、 'b' 等です。これらは、ASCIIコードに変換され、1バイトの即値として使用されま

#### 注意事項

- 文字定数のみのCウォッチポイントへの登録はできません。
- Cウォッチポイントを指定するC言語式の中に用いる場合、および代入する値を指定する場合にのみ有効です(文字定数は即値と同じ扱いになります)。

### 3.1.14 文字列リテラル

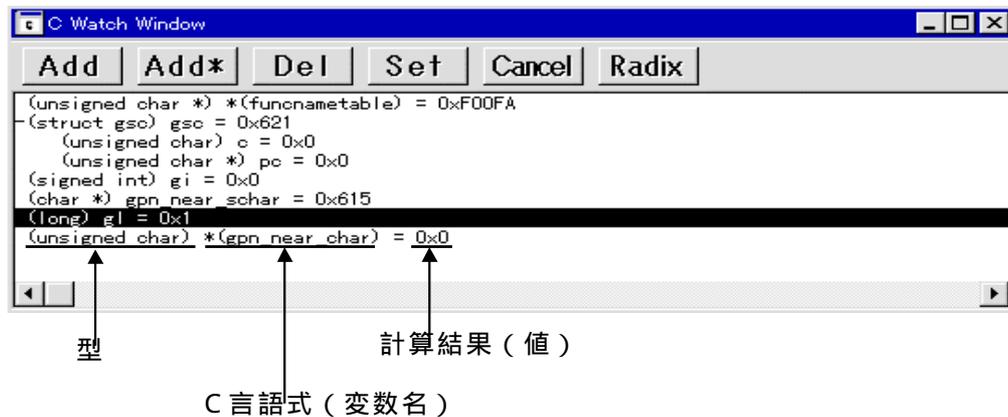
文字列リテラルとして、ダブルクォーテーション ( " ) で囲まれた文字列が使用できます。例えば、 "abcde"、 "I am a boy." 等です。

#### 注意事項

- 文字列リテラルは、右辺式(代入演算子の右辺)にのみ記述することができ、左辺式(代入演算子の左辺)が char 配列、または char ポインタ型の場合にのみ使用することができます。それ以外の場合には、文法エラーとなります。

## 3.2 C 言語式の表示形式

C ウォッチ、ファイル、ファイルローカル、及びグローバルウィンドウのデータ表示領域における C 言語式の表示は、以下に示すように、その型名、C 言語式（変数名）、計算結果（値）から構成されています。



以下に、型別に表示形式を説明します。

### 3.2.1 列挙型の場合

- 計算結果の値が定義されているものであれば、その名前で表示します。  
(DATE) date = Sunday (全 Radix)
- 計算結果の値が定義されているものでなかった場合には、以下のように表示します。  
(DATE) date = 16 (Radix が初期状態の場合)  
(DATE) date = 0x10 (Radix が 16 進数の場合)  
(DATE) date = 000000000010000B (Radix が 2 進数の場合)

### 3.2.2 基本型の場合

- 計算結果が char 型および浮動小数点以外の基本型の場合には、以下のように表示します。  
(unsigned int) i = 65280 (Radix が初期状態の場合)  
(unsigned int) i = 0xFF00 (Radix が 16 進数の場合)  
(unsigned int) i = 1111111100000000B (Radix が 2 進数の場合)
- 計算結果が char 型の場合には、以下のように表示します。  
(unsigned char) c = 'J' (Radix が初期状態の場合)  
(unsigned char) c = 0x4A (Radix が 16 進数の場合)  
(unsigned char) c = 10100100B (Radix が 2 進数の場合)
- 計算結果が浮動小数点の場合には、以下のように表示します。  
(double) d = 8.207880399131839E-304 (Radix が初期状態の場合)  
(double) d = 0x10203045060708 (Radix が 16 進数の場合)  
(double) d = 000000010....1000B (Radix が 2 進数の場合)  
(...は省略を表す)

### 3.2.3 ポインタ型の場合

- 計算結果が char\* 型以外のポインタ型の場合には、以下のように内容を 16 進数表示します。

```
(unsigned int *) p = 0x1234 (全 Radix)
```

- 計算結果が char\* 型の場合には、以下のように表示します。

```
(unsigned char *) str = 0x1234 "Japan" (全 Radix)
```

文字列の終わりを表すコード (0) までに、文字表示できないコードが格納されていた場合には、以下のように、閉じ (") を出力しません。

```
(unsigned char *) str = 0x1234 "Jap (全 Radix)
```

また、文字列の長さが 80 文字を越えた場合も同様に、閉じ (") を出力しません。

### 3.2.4 配列型の場合

- 計算結果が char[] 型以外の配列型の場合には、以下のように先頭アドレスを 16 進数表示します。

```
(signed int [10]) z = 0x1234 (全 Radix)
```

- 計算結果が char[] 型の場合には、以下のように表示します。

```
(unsigned char [10]) str = 0x1234 "Japan" (全 Radix)
```

文字列の終わりを表すコード (0) までに、文字表示できないコードが格納されていた場合には、以下のように、閉じ (") を出力しません。

```
(unsigned char [10]) str = 0x1234 "Jap (全 Radix)
```

また、文字列の長さが 80 文字を越えた場合も同様に、閉じ (") を出力しません。

### 3.2.5 関数型の場合

- 計算結果が関数型の場合には、以下のように関数の開始アドレスを 16 進数表示します。

```
(void()) main = 0xF000 (全 Radix)
```

### 3.2.6 参照型の場合

- 計算結果が参照型の場合には、以下のように参照するアドレスを 16 進数表示します。

```
(signed int &) ref = 0xD038 (全 Radix)
```

### 3.2.7 ビットフィールド型の場合

- 計算結果がビットフィールド型の場合には、以下のように表示します。

```
(unsigned int :13) s.f = 8191 (Radix が初期状態の場合)
```

```
(unsigned int :13) s.f = 0x1FFF (Radix が 16 進数の場合)
```

```
(unsigned int :13) s.f = 1111111111111B (Radix が 2 進数の場合)
```

### 3.2.8 C シンボルが見つからなかった場合

- 計算した式の中に発見できなかったC シンボルがあった場合には、以下のように表示します。  
 ( ) x = <not active> (全 Radix)

### 3.2.9 文法エラーの場合

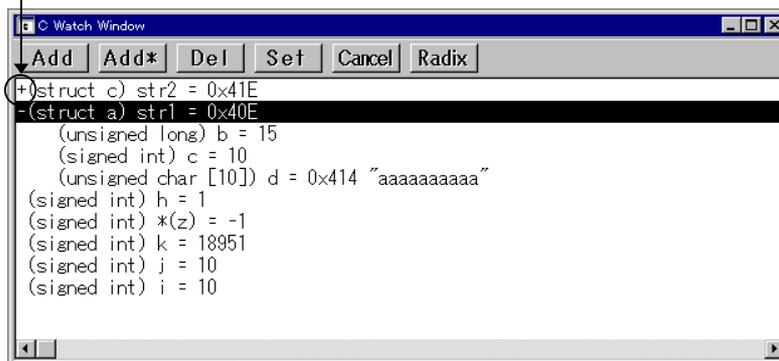
- 計算した式が文法的に間違っていた場合には、以下のように表示します。  
 ( ) str\*(p = <syntax error> (全 Radix)  
 (str\*(p は間違った記述)

### 3.2.10 構造体・共用体型の場合

- 計算結果が構造体・共用体型の場合には、以下のようにアドレスを 16 進数表示します。  
 (Data) v = 0x1234 (全 Radix)

なお、C 言語式が構造体・共用体型のようにメンバを持つ場合は、以下に示すように、型名(タグ名)の左側に '+' マークが現れます。

構造体・共用体を示す '+' マーク



この '+' マークが表示されている行をダブルクリックすると、その構造体(または共用体)のメンバが現れます。メンバを表示すると、 '+' マークは '-' マークにかわります。なお、 '-' マークが表示されている行をダブルクリックすると、もとの状態に戻ります。このようにして、メンバを参照することができます。

### 3.2.11 レジスタ変数の場合

- 計算結果がレジスタ変数の場合には、以下のように型名の先頭に "register" と表示します。  
 (register signed int) j = 100

## 4 エラーメッセージ一覧

以下に、PD32000のエラーメッセージ一覧を示します。

番号	エラーメッセージ	補足・対応
150	これ以上 ( name ) Window はオープンできません。	指定したウィンドウは既に最大枚数オープンしています。
151	( name ) Window のオープンに失敗しました。	指定したウィンドウを開くことができません。メモリ不足が原因と考えられます。他のアプリケーションを終了するか、メモリを増設してください。
152	ターゲット実行中のため ( name ) window はオープンできません。	ターゲットプログラムを停止してからウィンドウをオープンしてください。
153	指定した値が範囲外です。	指定したアドレスがMCUの最大アドレス 0xFFFFFFFF を越えています。
154	既に PD32000 は起動しています。	
156	指定されたファイル ( filename ) が見つかりません。	
157	指定されたパス ( path ) が見つかりません。	
158	メモリが不足しているため実行できませんでした。	
159	実行できませんでした。	

番号	エラーメッセージ	補足・対応
200	表示モードを切り替えることができません。	表示開始アドレスがソースの行頭アドレスと一致していない、または該当するソースファイルが見つかりません。
201	ソースファイル ( filename ) が見つかりません。	指定されたソースファイルが見つかりません。PATH コマンドまたは、メニュー[Environ][Path]でソースファイルがあるディレクトリを指定してください。
202	検索文字列 ( name ) が見つかりません。	検索開始位置から最後まで指定文字列を検索しましたが見つかりませんでした。
203	ソースファイル ( filename ) の行数が ( line ) 行を超えています。	ソースファイルが表示可能な行数を越えているために、ソース表示できません。表示モードを逆アセンブル表示モードに切り替えて表示します。

番号	エラーメッセージ	補足・対応
300	endi が多すぎます ( filename line ) 。	endi に対応する if がありません。
301	endw が多すぎます ( filename line ) 。	endw に対応する while がありません。
302	既にスクリプトファイルの最終行まで読みだしました。	
303	既にスクリプトファイルは起動しています。	
304	endi が足りません ( filename line ) 。	if に対応する endi がありません。
305	一行が長すぎます ( filename line ) 。	一行に記述できる文字数の制限を超えました。
306	スクリプトファイルのネストが制限 ( filename line ) を越えました。	
307	スクリプトファイルが見つかりません ( filename ) 。	
308	スクリプトファイルが読み込めません ( filename ) 。	スクリプトファイルの続きが読み込めません。
309	スクリプト文法エラー ( filename line ) 。	
310	endw が見つかりません ( filename line ) 。	while に対応する endw がありません。
311	スクリプトファイルのネストが制限 ( num ) を越えました。	

番号	エラーメッセージ	補足・対応
400	スクロール範囲外のアドレスが指定されました。	

番号	エラーメッセージ	補足・対応
600	ウォッチポイントの個数が制限数 ( num ) を超えるので追加できません。	
601	指定したアドレスが範囲外です。	
602	指定した値が範囲外です。	
603	指定したビット値が範囲外です。	

番号	エラーメッセージ	補足・対応
650	シンボル情報がロードされていません。	ロードモジュールファイルがロードされていません。
651	文字列が長すぎます。	

番号	エラーメッセージ	補足・対応
900	シンボルファイルフォーマットが異常です。	ロードモジュールファイルのフォーマットに誤りがあります。
901	シンボルファイルのロードを中断しました。	
902	シンボルファイル ( filename ) が見つかりません。	ロードモジュールファイルが存在しません。
903	必要なメモリが確保できません。	メモリが不足しています。他のアプリケーションを終了するか、メモリを増設してください。
904	テンポラリファイルがオープンできません。	オンデマンド方式でダウンロードする際のテンポラリファイルがオープンできません。。

番号	エラーメッセージ	補足・対応
1001	シンボルが見つかりません。	指定したシンボルは、存在しません。
1002	指定した式は、ウォッチポイントとして登録できません。	
1004	文法エラーです。	式の記述に誤りがあります。
1005	スコープが見つかりません。	指定された変数は、スコープ内にありません。
1006	シンボルが見つかりません。	
1007	関数が見つかりません。	指定された関数はありません。
1008	右辺式が不適切です。	
1009	型の異なる構造体 ( 共用体 ) をコピーしようとしました。	
1010	代入できません。	
1011	型が見つかりません。	指定された型はありません。
1012	浮動小数点型の演算はサポートしていません。	
1013	指定の演算はポインタ型同士に対してはできません。	
1014	指定の演算はポインタ型に対してはできません。	
1015	ポインタ変数によって減算しようとしました。	
1016	0 で除算しようとしました。	
1017	不正な演算子を用いています。	
1018	型情報が壊れています。	ロードモジュールファイルのシンボル情報に誤りがあります。
1019	左辺値は、ポインタ変数でなければなりません。	
1020	左辺値は、構造体 ( 共用体 ) 型でなければなりません。	
1021	メンバが見つかりません。	
1022	左辺値は、構造体 ( 共用体 ) 型への参照でなければなりません。	
1023	左辺値が不適切です。	

番号	エラーメッセージ	補足・対応
1024	被演算子は数値でなければなりません。	
1025	指定の被演算子は符号反転できません。	
1026	アドレス値を求めることができません。	
1027	配列変数が不適切です。	
1028	配列の要素番号が不適切です。	
1029	被演算子がアドレスではありません。	
1030	レジスタ変数に対するキャスト演算はサポートしておりません。	
1031	キャストする型の指定が不適切です。	
1032	指定の型に対するキャスト演算はサポートしておりません。	

番号	エラーメッセージ	補足・対応
1100	指定したアドレス値が範囲外です。	指定したアドレスが MCU の扱える最大値 0xFFFFFFFF を越えています。
1101	アセンブリ言語の記述に誤りがあります。	
1102	ジャンプ先のアドレスが範囲外です。	
1103	指定したオペランドの値が範囲外です。	
1104	式の記述に誤りがあります。	
1105	アドレッシングモード指定子の記述に間違いがあります。	
1107	オペランドの値が未定義です。	
1108	式中にビットシンボルがあります。	
1109	無効なビットシンボルがあります。	
1110	値がアセンブル時確定値ではありません。	
1111	オペランドの同一項目を複数指定しています。	
1112	オペランドの同種の項目を複数指定しています。	
1113	命令または式中に余分な文字があります。	
1114	フォーマット指定子の記述に間違いがあります。	
1115	シンボルの定義に間違いがあります。	
1116	オペランド中に予約語が記述されています。	
1118	予約語の記述がありません。	
1119	ニーモニック、アセンブル指示命令の直後に空白文字がありません。	
1123	オペランドの値が未定義です。	
1124	オペランドのサイズが間違っています。	
1125	オペランドの種類が間違っています。	
1131	サイズ指定子、又はフォーマット指定子の記述に間違いがあります。	

番号	エラーメッセージ	補足・対応
1132	サイズ指定子がありません。	
1133	式中に文字列式が記述されています。	
1134	シンボルの記述がありません。	
1135	シンボルはすでに定義されています。	
1136	' .EQU '、' .BTEQU ' で定義されるシンボル名の記述がありません。	
1137	シンボルはすでに定義されています。	
1138	命令に無効なオペランドがあります。	
1139	式の記述に間違いがあります。	
1140	命令に無効なオペランドがあります。	
1141	オペランド記述に不足があります。	
1142	オペランドが余分にありませう。	
1143	オペランドのデータが多すぎます。	
1144	未定義のシンボルがあります。	
1145	値が範囲外です。	
1146	0 除算が行われています。	
1148	' # ' の記述がありません。	
1149	' , ' の記述がありません。	
1150	' ] ' の記述がありません。	
1151	' ) ' の記述がありません。	
1153	命令に無効なオペランドがあります。	
1154	文字列に対する引用符の記述がありません。	
1155	右側の引用符がありません。	
1156	メモリの獲得ができません。	
1157	奇数アドレスは指定できません。	

番号	エラーメッセージ	補足・対応
1200	コマンド行の文法エラーです。	
1201	コマンド名に誤りがあります。	
1202	alias の登録が多すぎます。	alias の最大登録数は、256 個です。
1203	alias にはコマンド名のみ登録できます。	
1204	ターゲットプログラム実行中のため、指定したコマンドは使用できません。	指定されたコマンドは、プログラムを実行しているときには使用できません。
1205	これ以上 up できません。	
1206	これ以上 down できません。	
1207	この関数にブレークをかけることはできません。	
1208	すでにハードウェアブレークポイントが設定されています。	
1209	現在このコマンドはサポートされていません。	
1210	開始アドレスが終了アドレスよりも大きいアドレス値になっています。	
1211	別名にコマンド名および予約語は指定できません。	
1212	セーブファイル名 ( filename ) が不正です。	指定したファイルを保存することができません。
1214	ターゲットのタイプが異なるため、本コマンドは実行できません。	
1215	開始サイクルが終了サイクルよりも大きいサイクル値になっています。	
1216	MR32Rのシンボル情報 ( symbol ) が見つかりません。	リアルタイム OS ( MR32R ) のシンボルが見つかりません。
1218	MR32Rの初期化ルーチンが実行されていません。	リアルタイム OS ( MR32R ) の初期化ルーチンが実行されていない時には、MR コマンドは実行できません。
1219	指定されたタスク番号のタスクが見つかりません。	

番号	エラーメッセージ	補足・対応
1220	優先度が範囲外です。	指定した優先度が範囲外です。
1221	タスク ID が範囲外です。	指定したタスク ID が範囲外です。
1222	フラグ ID が範囲外です。	指定したフラグ ID が範囲外です。
1223	セマフォ ID が範囲外です。	指定したセマフォ ID が範囲外です。
1224	メールボックス ID が範囲外です。	指定したメールボックス ID が範囲外です。
1225	メモリプール ID が範囲外です。	指定したメモリプール ID が範囲外です。
1226	周期起動ハンドラ ID が範囲外です。	指定した周期起動ハンドラ ID が範囲外です。
1227	アドレスが範囲外です。	指定したアドレスが範囲外です。
1228	システムコール発行できません。	
1229	システムコール発行されていません。	
1230	システムコール発行が終了していません。	
1232	逆アセンブル表示できないトレースデータです。	
1234	ターゲットの停止はできません。	
1235	奇数アドレスは指定できません。	
1236	ファイル( filename )が見つかりません。	
1237	MR のデータが壊れています。	
1238	必要なメモリが確保できませんでした。	

番号	エラーメッセージ	補足・対応
1300	行番号の指定に誤りがあります。	
1301	右括弧 ' ) ' が見つかりません。	
1302	マクロ定数の個数が制限数 ( limit ) を越えています。	
1303	指定した定数値が範囲外です。	
1304	定数の基数を示すプレフィクスの記述に誤りがあります。	
1305	間接参照の記述に誤りがあります。	
1306	文字列の終わりを示す ( str ) が見つかりません。	
1307	式の記述に誤りがあります。	
1308	マクロ定数 ( macro ) が定義されていません。	
1309	シンボル ( symbol ) が定義されていません。	
1310	定数値の記述に誤りがあります。	
1311	0 で除算を行いました。	
1313	解析結果が MCU の扱える最大値を越えています。	
1314	マクロ変数名にレジスタ名を使用しています。	

番号	エラーメッセージ	補足・対応
1400	指定したアドレス値が範囲外です。	指定したアドレスが MCU の扱える最大値 0xFFFFFFFF を越えています。
1402	ブレークポイントの個数が制限数( limit )を超えています。	
1403	ブレークポイントが設定されていません。	
1404	指定したデータ値が範囲外です。	
1406	指定した領域にメモリがないので、参照/書き込みができません。	メモリの存在しないアドレスに対しての参照/書き込みは行なえません。
1408	指定したレジスタ値が範囲外です。	
1409	ターゲットプログラム実行中のため、指定したコマンドは使用できません。	
1410	開始アドレスが終了アドレスよりも大きいアドレス値になっています。	
1411	実行を中断しました。	
1417	これ以上のスタックの検索はできません。	
1418	指定回数が 65535 回を越えています。	
1420	メモリマップ情報(.mpu File)が不正です。	
1421	転送元(先)のメモリサイズがMPU ファイルで規定したサイズと異なります。	
1422	アライメントの設定が不正です。	
1423	奇数値は指定できません。	

番号	エラーメッセージ	補足・対応
1552	ファイル( filename )が壊れています。	
1553	ファイル( filename )が見つかりません。	
1554	関数/サブルーチン情報が見つかりません。	デバッグ情報を出力するオプションをつけて、ターゲットプログラムを再度作成して下さい。
1555	シンボル/ラベルとして記述できない文字が文字列中にあります。	
1557	指定した行番号が見つかりません。	
1558	既に同名のシンボル/ラベルが登録されています。	
1560	必要なメモリが確保できません。	
1561	スコープが見つかりません。	
1562	セクション情報が見つかりません。	デバッグ情報を出力するオプションをつけて、ターゲットプログラムを再度作成して下さい。
1563	指定したアドレスに該当するソース行が見つかりません。	
1564	シンボル( symbol )が見つかりません。	
1565	指定したアドレスを含むスコープが見つかりません。	
1566	ロードが中断されました。	
1569	レジスタ名に誤りがあります。	
1570	これはELF/DWARF形式のオブジェクトファイルではありません。	

番号	エラーメッセージ	補足・対応
1704	ターゲットと接続されていません。	
1705	ターゲットに接続できません。	
1707	タイムアウトエラーが発生しました。	ターゲットとの通信中にタイムアウトエラーが発生しました。
1712	通信エラーが発生しました。ターゲットとの接続が切断されました。	ターゲットとの通信中にターゲットとの接続が切断されました。
1713	通信エラーが発生しました。ターゲットにデータを転送できません。	ターゲットへのデータ転送中に通信エラーが発生しました。
1714	通信エラーが発生しました。ターゲットよりデータを受信できません。	ターゲットからのデータ受信中に通信エラーが発生しました。
1715	既にターゲットは使用されています。	

番号	エラーメッセージ	補足・対応
2400	指定したアドレスが不正です。	
2401	範囲外のデータ値が指定されました。	
2402	指定したアドレスが不正です。	開始アドレスより小さい値を終了アドレスに指定しています。
2403	指定回数は ( num ) 以上を指定してください。	num 以上の値を指定してください。
2404	範囲外の値が指定されました。	

番号	エラーメッセージ	補足・対応
5200	ブレークポイントが制限数まで設定されているため、Come 実行ができません。	いずれかのブレークポイントを削除してから、Come 実行してください。

番号	エラーメッセージ	補足・対応
5500	指定した値が範囲外です。	

番号	エラーメッセージ	補足・対応
5700	範囲外のデータ値が指定されました。	
5701	指定したアドレスが不正です。	
5702	スクロール範囲外のアドレスが指定されました。	スクロール範囲に指定したアドレスが、MCU の最大アドレス 0xFFFFFFFF を越えています。

番号	エラーメッセージ	補足・対応
5900	指定されたスクリプトファイル ( filename ) がオープンできません。	
5901	指定されたスクリプトファイル ( filename ) は既にオープンされています。	
5902	スクリプトファイルがオープンされていません。	
5903	指定されたログファイル ( filename ) がオープンできません。	
5904	これ以上ログファイルをオープンすることができません。	
5905	ログファイルがオープンされていません。	
5906	指定されたログファイル ( filename ) は既にオープンされています。	
5907	ビューファイル ( filename ) がオープンできません。	

番号	エラーメッセージ	補足・対応
7000	指定されたタスク番号のタスクが見つかりません。	指定したタスクは存在しません。
7001	指定されたタスク番号のコンテキストが見つかりません。	指定したタスクのコンテキストは存在しません。
7002	MR のデータが壊れています。	MR データの一部が壊れています。このため MR ウィンドウに正しいデータを表示することができません。
7003	必要なメモリが確保できません。	メモリが不足しています。他のアプリケーションを終了するか、メモリを増設してください。

番号	エラーメッセージ	補足・対応
8000	ファイル( filename )の内容に誤りがあります。	
8001	ファイル( filename )が見つかりません。	
8002	ファイル( filename )がオープンできません。	

番号	エラーメッセージ	補足・対応
10000	トレースデータが不足もしくは異常のためBUSモード以外に変更できません。	

番号	エラーメッセージ	補足・対応
10022	これ以上 Come 検索できません。	
10024	これ以上 Step 検索できません。	
10025	指定したサイクル値が範囲外です。	
10026	指定したアドレス値が見つかりません。	

番号	エラーメッセージ	補足・対応
11450	指定したアドレス値が範囲外です。	
11452	指定したビット番号が範囲外です。	
11453	実行を中断しました。	
11454	指定したデータ値が範囲外です。	
11456	ファイル ( filename ) が見つかりません。	
11457	ターゲットシステムが正しく構成されていません。	PD32000, M32000T, と POD の組合せが正しくありません。
11459	指定したマスク値が範囲外です。	

番号	エラーメッセージ	補足・対応
11462	指定したパスカウント値が範囲外です。	
11463	ターゲットプログラム実行中のため、指定したコマンドは使用できません。	
11473	参照可能なトレースデータが見つかりません。	
11474	指定したサイクル値が範囲外です。	

番号	エラーメッセージ	補足・対応
11500	エミュレータから未定義のステータスが送信されました。	
11501	Parallel NOP 命令へのブレークポイント設定はできません。	
11502	時間計測ポイントの個数が制限数( limit )を超えています。	
11503	時間計測ポイントが設定されていません。	
11504	時間計測ポイントが設定できません。	トレースモードをNO にして下さい。
11508	アドレス例外処理が発生しました。	
11509	予約命令例外処理が発生しました。	
11510	システム保護空間への不正アクセスが発生しました。	
11511	プログラムの最後まで実行しました。	
11512	M32R/D が暴走しているなどの理由で、エミュレータの応答がありません。	リセットボタンを押すか、Reset コマンドを実行してください。
11513	リセット後、エミュレータの応答がありません。	エミュレータシステムの電源を再投入してください。
11514	受信データの形式が異常です。	リセットボタンを押すか、Reset コマンドを実行してください。
11515	指定したイベント番号(:OR 番号)が範囲外です。	
11516	指定したデータタイプは設定できません。	
11517	データタイプとアドレス値が異なります。	
11518	指定したパラメータが間違っています。	
11519	トレースデータが見つかりません。	

# 索引



<b>A</b>		<b>あ</b>	
ASM ウォッチウィンドウ.....	38, 39	アップロード.....	73
<b>B</b>		アブソリュート形式ファイル .....	6, 71
break 文.....	157, 158	<b>い</b>	
<b>C</b>		インストール.....	13
C ウォッチウィンドウ .....	40, 41	<b>う</b>	
C ウォッチポイント.....	40	ウォッチウィンドウ.....	38
C ウォッチポイント情報格納ファイル 7		ウォッチポイント .....	38, 100
C 言語式.....	40, 106, 164, 168	ウォッチ機能.....	100
<b>E</b>		<b>え</b>	
else 文 .....	157, 158	エミュレーションメモリダイアログ..	50
endi 文 .....	157, 158	エミュレーションメモリ状態保存ファイ	
endw 文 .....	157, 158	ル .....	7, 9
<b>I</b>		演算子 .....	163
if 文.....	157, 158	<b>お</b>	
Init ダイアログ .....	14, 112	オーバーステップ実行 .....	82
<b>L</b>		オンデマンド方式 .....	6, 16
LAN 通信 .....	15	オンメモリ方式 .....	6, 16
<b>M</b>		<b>か</b>	
Make.....	116	カスタマイズ機能 .....	126
MPU ファイル.....	15	カスタムウィンドウプログラム.....	126
MIX 表示モード.....	78	カスタムコマンドプログラム .....	126
MR ウィンドウ .....	66	カム実行.....	83
<b>P</b>		環境設定.....	24
PC トレースモード.....	51	環境設定ファイル .....	6
PD32000 ウィンドウ .....	21	関数内ローカル変数 .....	42
<b>S</b>		<b>き</b>	
S/W ブレークポイント設定ダイアログ		起動.....	13
.....	21	逆アセンブルファイル .....	8
signal トレースモード .....	52	逆アセンブル結果保存 .....	73
<b>W</b>		行番号 .....	163
while 文.....	157, 158	<b>く</b>	

グローバルウィンドウ .....	45	トレース機能.....	4
グローバルラベルシンボル.....	163	トレースコンディションウィンドウ51,	
グローバル変数.....	45	55, 90	
区間時間測定ウィンドウ .....	63, 64,123	トレースポイント .....	56
繰り返し文 .....	159, 160	トレースポイント保存ファイル.....	7, 8
<b>こ</b>		トレース計測結果保存ファイル.....	9
コールスタックウィンドウ.....	65	<b>は</b>	
コメント文 .....	157, 158	バイナリファイル .....	6, 73
<b>し</b>		判断文 .....	157, 158
シンボル.....	161	<b>ひ</b>	
<b>す</b>		ビューバッファ .....	8, 46, 109
スクリプトウィンドウ .....	46, 47	ビューファイル .....	8, 46
スクリプトコマンド... ..	46, 107, 151, 157	<b>ふ</b>	
スクリプトファイル.. ..	6, 15, 46, 110,157	ファイルローカルウィンドウ .....	44
スコープ.....	161	ファイル内ローカル変数.....	44
ステップ実行.....	82	プログラムウィンドウ .....	26, 29, 114
<b>そ</b>		<b>へ</b>	
ソースウィンドウ .....	30, 32, 114	ヘルプファイル .....	6
ソフトウェアブレーク .....	4, 85	<b>ま</b>	
ソフトウェアブレークポイント. ..	4,48,85	マクロ変数 .....	162
ソフトウェアブレークポイント保存フ		<b>め</b>	
イル .....	7, 8	メモリウィンドウ .....	34, 35
<b>た</b>		メモリ変数 .....	162
代入文 .....	157	<b>も</b>	
ダウンロード.....	71	文字定数.....	163
ダンプウィンドウ .....	36, 37	モトローラS フォーマットファイル6, 8,	
<b>て</b>		71, 73	
定数 .....	160	<b>ら</b>	
テンポラリファイル.....	9, 17	ラインアセンブル .....	114, 115
<b>と</b>		ラインアセンブルダイアログ .....	114
動作環境.....	14	ラベル .....	34, 161
トレース.....	90	<b>り</b>	
トレースウィンドウ.....	56, 62, 90		

---

リアルタイム OS.....	133	ローカルウィンドウ.....	42, 43
リセット.....	84	ローカルラベルシンボル.....	161
リターン実行.....	83	ロギング機能.....	108
		ログファイル.....	8, 46

**れ**

レジスタウィンドウ.....	33
レジスタ変数.....	162

**ろ**

【 MEMO 】

# 技術サポート連絡書

年 月 日 (合計 枚)

三菱電機セミコンダクタシステム株式会社  
マイコンソフトツール部

## 開発ツールサポート窓口行

[電子メール] support@tool.mesc.co.jp  
[大阪地区] FAX : 06-6338-4471  
[東京地区] FAX : 03-5403-0414  
[中部地区] FAX : 052-221-7318  
[九州地区] FAX : 092-452-1427

インストーラが生成する以下のテキストファイルもサポート連絡書としてご利用できます。  
Windows 98/95/Windows NT 4.0版の場合 ※SUPPORT※製品名※SUPPORT.TXT  
EWS版の場合 :/support/製品名/toolinfo.txt

ご連絡先	製品情報
会社名 :	ソフトウェア名 :
部署名 :	バージョン番号 V.
担当者名 :	ライセンスID :
電話番号 :	. . . .
FAX番号 :	ホストマシン名 :
電子メールアドレス :	OS名 : V.
通信欄 :	

太枠内の項目にご記入ください。

受付日 : 年 月 日	回答日 : 年 月 日
担当名 :	
MEMO:	

整理番号 No.

お問い合わせ内容 (リスト等がある場合は、添付してください。)

三菱電機セミコンダクタシステム株式会社  
マイコンソフトツール部

[ 電子メール ] support@tool.mesc.co.jp

[ 大阪地区 ] FAX : 06-6338-4471

[ 東京地区 ] FAX : 03-5403-0414

[ 中部地区 ] FAX : 052-221-7318

[ 九州地区 ] FAX : 092-452-1427

( / )

**PD32000 V.1.00ユーザーズマニュアル**  
第1版 :1999年4月16日発行  
資料番号 **MSD-PD32000-U-990416**

---

**Copyright ©1999 三菱電機株式会社**  
©1999 三菱電機セミコンダクタシステム株式会社

三菱電機株式会社

三菱電機セミコンダクタシステム株式会社

M32R/D シリーズ用 エミュレータデバッグ  
PD32000 V.1.00 ユーザーズマニュアル



ルネサスエレクトロニクス株式会社  
神奈川県川崎市中原区下沼部1753 〒211-8668