

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ひ申し上げます。

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

M3XXXXT-PAC

《コントロールソフトウェアPDB38M編》

ユーザーズマニュアル

Simple Emulator System Package for 740 Family

《安全設計に関するお願い》

三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障または誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

《本資料ご利用に際しての留意事項》

本資料は、お客様が用途に応じた適切な三菱半導体製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング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、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング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全ての情報は本資料発行時点のものであり、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は、予告なしに、本資料に記載した製品または仕様を変更することがあります。三菱半導体製品のご購入に当たりましては、事前に三菱電機株式会社、三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社または特約店へ最新の情報をご確認頂きますとともに、三菱電機半導体情報ホームページ (<http://www.semicon.melco.co.jp/>) および三菱開発ツールホームページ (<http://www.tool-spt.maec.co.jp/>) などを通じて公開される情報に常にご注意ください。

本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社はその責任を負いません。

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。三菱電機株式会社および三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社は、適用可否に対する責任を負いません。

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、特殊用途へのご利用をご検討の際には、三菱電機株式会社、三菱電機セミコンダクタ・アプリケーション・エンジニアリング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および三菱電機セミコンダクタ・アプリケーション・エンジニアリング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・アプリケーション・エンジニアリング株式会社または特約店までご照会ください。

《製品の内容および本書についてのお問い合わせ先》

三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社 マイコンツール部

[技術サポート窓口] support@tool.maec.co.jp

[ユーザ登録窓口] regist@tool.maec.co.jp

[ホームページアドレス] <http://www.tool-spt.maec.co.jp/>

はじめに

PDB38Mは、三菱8ビットマイクロコンピュータ用簡易エミュレータシステムのデバッグソフトウェアです。

製品概要

PDB38Mは以下のものから構成されています。

1. フロッピーディスク
2. ユーザーズマニュアル
3. リリースノート
4. ソフトウェア使用権許諾契約書（リリースノートの最後に添付）
5. お客様登録カード

本製品に同封されていますリリースノートには、マニュアルの補足等が記載されていますので、必ずお読みください。また、お客様登録カードは、ご使用中のお客様への緊急の連絡が必要となった場合等に使用しますので、必ずご返却くださいますようお願い致します。もしこれらのものが含まれていない場合は、ご購入いただいた三菱電機営業又は特約店にご連絡ください。

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。PDB38M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

PDB38Mのマニュアルでは、以下のように用語を定義しています。

表記	意味
M38000T-SBI	エミュレータ本体を意味します。
SRA74M	アセンブラを意味します。

目次

セットアップ編		1
1.	製品パッケージの内容	3
1.1.	包装製品一覧表.....	3
2.	動作環境	4
3.	システム構成	5
3.1.	ホストマシン	6
3.2.	エミュレータ M38000T-SBI.....	6
3.3.	ターゲットシステム	6
3.4.	通信方法.....	6
4.	PDB38M が扱うファイル	7
4.1.	入力ファイル	8
4.1.1.	インテル HEX フォーマットファイル.....	8
4.1.2.	シンボルファイル	8
4.1.3.	定義体ファイル.....	8
4.1.4.	スクリプトファイル.....	8
4.1.5.	環境設定ファイル	9
4.2.	出力ファイル	10
4.2.1.	インテル HEX フォーマットファイル.....	10
4.2.2.	逆アセンブルファイル	10
4.2.3.	ログファイル.....	11
4.2.4.	ビューファイル.....	11
ウィンドウ操作編		13
5.	PDB38M ウィンドウの操作方法	15
5.1.	PDB38M ウィンドウの概要.....	15
5.2.	PDB38M ウィンドウの構成.....	16
5.2.1.	メニュー	17
5.2.2.	ツールバー	17

5.2.3. メイン表示領域	18
5.2.4. 各種ウィンドウ	18
5.2.5. ステータスバー	18
5.3. 基本メニューの操作方法	19
5.3.1. Fileメニュー	20
5.3.2. Editメニュー	26
5.3.3. Viewメニュー	28
5.3.4. Environメニュー	28
5.3.5. Debugメニュー	32
5.3.6. Optionメニュー	39
5.3.7. Windowメニュー	39
5.3.8. Helpメニュー	41
5.4. ツールバーの構成	42
6. プログラムウィンドウの操作方法	44
6.1. プログラムウィンドウの概要	44
6.2. プログラムウィンドウの構成	45
6.2.1. ツールバー	46
6.2.2. プログラム表示領域	46
6.2.3. 行番号表示領域	46
6.2.4. アドレス表示領域	47
6.2.5. ブレークポイント表示領域	47
6.2.6. オブジェクトコード表示領域	48
6.2.7. 垂直スクロールバー	48
6.2.8. 水平スクロールバー	48
6.3. プログラムウィンドウの表示モードについて	49
6.3.1. ソース表示モード	49
6.3.2. 逆アセンブル表示モード	50
6.4. 表示内容・表示モードの更新について	51
6.4.1. 表示内容の自動更新	52
6.4.2. 表示モードの自動更新	52
6.4.3. 手動による表示内容・表示モードの更新	52
6.5. 拡張メニューの操作方法	54
6.6. ツールバーの構成	60
7. ソースウィンドウの操作方法	62
7.1. ソースウィンドウの概要	62

8.	レジスタウィンドウの操作方法	63
8.1.	レジスタウィンドウの概要.....	63
8.2.	レジスタウィンドウの構成.....	63
8.2.1.	レジスタ領域.....	63
8.3.	レジスタ値の設定方法.....	65
8.3.1.	プロセッサステータスレジスタの設定.....	65
8.3.2.	その他のレジスタの設定.....	65
8.4.	レジスタウィンドウの表示内容の更新について.....	65
9.	メモリウィンドウの操作方法	66
9.1.	メモリウィンドウの概要.....	66
9.2.	メモリウィンドウの構成.....	66
9.2.1.	ツールバー.....	66
9.2.2.	メモリ表示領域.....	67
9.2.3.	アドレス表示領域.....	67
9.3.	拡張メニューの操作方法.....	68
9.4.	ツールバーの構成.....	74
9.5.	メモリウィンドウの表示内容の更新について.....	75
10.	ダンプウィンドウの操作方法	76
10.1.	ダンプウィンドウの概要.....	76
10.2.	ダンプウィンドウの構成.....	76
10.2.1.	ツールバー.....	77
10.2.2.	メモリ表示領域.....	77
10.2.3.	アドレス表示領域.....	77
10.3.	拡張メニューの操作方法.....	78
10.4.	ツールバーの構成.....	84
10.5.	ダンプウィンドウの表示内容の更新について.....	84
11.	ウォッチウィンドウの操作方法	85
11.1.	ウォッチウィンドウの概要.....	85
11.2.	ウォッチウィンドウの構成.....	86
11.2.1.	ツールバー.....	86
11.2.2.	アドレス/ビット番号表示領域.....	86
11.2.3.	アドレス式表示領域.....	86
11.2.4.	データ長表示領域.....	87
11.2.5.	基数表示領域.....	87
11.2.6.	データ表示領域.....	87

11.3.	拡張メニューの操作方法	89
11.4.	ツールバーの構成	95
11.5.	ウォッチウィンドウの表示内容の更新について	96
11.6.	ウォッチポイントのアドレス再計算機能について	96
11.7.	ウォッチポイントの保存機能について	96
12.	スクリプトウィンドウの操作方法	97
12.1.	スクリプトウィンドウの概要	97
12.1.1.	ツールバー	98
12.1.2.	ファイル名表示領域	98
12.1.3.	コマンド表示領域	98
12.1.4.	コマンド履歴表示領域	98
12.1.5.	コマンド入力領域	99
12.1.6.	スクリプトファイル表示領域	99
12.2.	スクリプトコマンドの実行中断方法について	100
12.3.	拡張メニューの操作方法	101
12.4.	ツールバーの構成	106
13.	S/Wブレークポイント設定ダイアログの操作方法	109
13.1.	S/W ブレークポイント設定ダイアログの概要	109
13.2.	S/W ブレークポイント設定ダイアログの構成	110
13.3.	S/W ブレークポイント設定ダイアログの操作方法	110
13.3.1.	ソフトウェアブレークポイントの参照	111
13.3.2.	ソフトウェアブレークポイントの設定	111
13.3.3.	ソフトウェアブレークポイントの削除	112
13.3.4.	ソフトウェアブレークポイントの無効化	113
13.3.5.	ソフトウェアブレークポイントの有効化	113

スクリプト仕様編 **115**

14.	スクリプトコマンドの概要	117
14.1.	コマンド一覧	117
14.2.	コマンドの入力形式	118
14.3.	コマンドの詳細説明	120
14.3.1.	コマンド詳細の記述内容	120
14.3.2.	コマンド入力書式の表記方法	121
14.3.3.	各コマンドの詳細説明	121

15. 式の記述方法	155
15.1. 式の構成要素	155
15.2. ラベルおよびシンボル.....	157
15.2.1. ラベルおよびシンボルの記述方法	157
15.2.2. ローカルラベル/シンボルとスコープ	157
15.2.3. ラベル/シンボルの優先順位	158
15.3. 定数.....	159
15.4. 演算子	160
15.5. 行番号	161
トラブルシューティング編	163
16. エラーメッセージ一覧	165
索引	181

図目次

図 3-1 エミュレータ本体(M38000T-SBI)のシステム構成図.....	5
図 4-1 逆アセンブルファイル例.....	11
図 5-1 PDB38M ウィンドウの構成図.....	16
図 5-2 メニューの表示例	17
図 5-3 PDB38M ウィンドウのツールバー表示例.....	17
図 5-4 ステータスバーの表示例.....	18
図 5-5 ファイルセクションダイアログ	21
図 5-6 ダウンロード中のダイアログ表示例.....	22
図 5-7 Upload ダイアログ	22
図 5-8 ファイルセクションダイアログ	23
図 5-9 アップロード中のダイアログ表示例.....	23
図 5-10 Save ダイアログ	24
図 5-11 ファイルセクションダイアログ	25
図 5-12 逆アセンブル結果保存中のダイアログ表示例	25
図 5-13 ウォーニングダイアログ.....	26
図 5-14 Find ダイアログ.....	26
図 5-15 プログラム表示領域のカーソル表示例	27
図 5-16 Init ダイアログ	29
図 5-17 Path ダイアログ	30
図 5-18 ファイルセクションダイアログ	31
図 5-19 クエスチョンダイアログ.....	31
図 5-20 StartUp ダイアログ.....	32
図 5-21 Go ダイアログ.....	33
図 5-22 ファイルセクションダイアログ	34
図 5-23 プログラム表示領域のカーソル表示例	35
図 5-24 Step ダイアログ.....	35
図 5-25 Over ダイアログ	36
図 5-26 S/W ブレークポイント設定ダイアログ.....	38
図 5-27 プログラム表示領域のカーソル表示例	39
図 5-28 About ダイアログ	41
図 5-29 PDB38M ウィンドウのツールバー構成	42
図 6-1 ソース表示モードのプログラムウィンドウ構成図	45
図 6-2 逆アセンブル表示モードのプログラムウィンドウ構成図	45

図 6.-3	プログラムカウンタの表示例.....	46
図 6.-4	ブレークポイント表示領域の表示例.....	47
図 6.-5	ソース表示モードでの表示.....	49
図 6.-6	逆アセンブル表示モードでの表示.....	50
図 6.-7	フォント指定ダイアログ.....	55
図 6.-8	TAB ダイアログ.....	55
図 6.-9	Source ダイアログ.....	56
図 6.-10	Address ダイアログ (ソースモードの場合).....	57
図 6.-11	Address ダイアログ (逆アセンブルモードの場合).....	57
図 6.-12	プログラムウィンドウのツールバーの構成.....	60
図 6.-13	Disp Area ダイアログ.....	60
図 8.-1	レジスタウィンドウの構成図.....	63
図 8.-2	Set Register ダイアログ (その2).....	65
図 9.-1	メモリウィンドウの構成図.....	66
図 9.-2	フォント指定ダイアログ.....	69
図 9.-3	Scroll Area ダイアログ.....	70
図 9.-4	Address ダイアログ.....	70
図 9.-5	Set ダイアログ.....	72
図 9.-6	Fill ダイアログ.....	73
図 9.-7	メモリウィンドウのツールバーの構成.....	74
図 10.-1	ダンプウィンドウの構成図.....	76
図 10.-2	フォント指定ダイアログ.....	79
図 10.-3	Scroll Area ダイアログ.....	80
図 10.-4	Address ダイアログ.....	80
図 10.-5	Set ダイアログ.....	82
図 10.-6	Fill ダイアログ.....	83
図 10.-7	ダンプウィンドウのツールバーの構成.....	84
図 11.-1	ウォッチウィンドウの構成図.....	86
図 11.-2	カーソル表示例.....	87
図 11.-3	カーソル表示例.....	88
図 11.-4	フォント指定ダイアログ.....	90
図 11.-5	Add ダイアログ.....	91
図 11.-6	ウォーニングダイアログ.....	92
図 11.-7	BitAdd ダイアログ.....	92
図 11.-8	Set ダイアログ.....	93
図 11.-9	クエスチョンダイアログ.....	94

図 11.-10	ウォッチウィンドウのツールバーの構成	95
図 12.-1	スクリプトウィンドウの構成図.....	98
図 12.-2	スクリプトファイルオープン時のスクリプトウィンドウ構成図	99
図 12.-3	スクリプトコマンド実行中のダイアログ表示例.....	100
図 12.-4	フォント指定ダイアログ.....	102
図 12.-5	ファイルセレクションダイアログ	102
図 12.-6	スクリプトファイル実行中のダイアログ表示例.....	103
図 12.-7	ファイルセレクションダイアログ	104
図 12.-8	ファイルセレクションダイアログ	105
図 12.-9	スクリプトウィンドウのツールバーの構成.....	106
図 13.-1	S/W ブレークポイント設定ダイアログ.....	110
図 13.-2	ファイルセレクションダイアログ	112
図 13.-3	クエスチョンダイアログ.....	113
図 14.-1	コマンドの入力形式.....	119
図 15.-1	式を使用したコマンド入力例図.....	155
図 15.-2	行番号の入力形式	161

表目次

表 1-1	PDB38M 包装内容一覧表	3
表 2-1	PDB38M の動作環境	4
表 3-1	ホストマシン	6
表 5-1	PDB38M ウィンドウのメニューの構成 (その 1)	19
表 5-2	PDB38M ウィンドウのメニューの構成 (その 2)	20
表 6-1	プログラムウィンドウの拡張メニュー	54
表 8-1	レジスタの種類	64
表 9-1	メモリウィンドウの拡張メニュー	68
表 10-1	ダンプウィンドウの拡張メニュー	78
表 11-1	ウォッチウィンドウの拡張メニュー	89
表 12-1	スクリプトウィンドウの拡張メニュー	101
表 14-1	コマンド一覧(実行と停止)	118
表 14-2	コマンド入力書式の表記方法	121
表 15-1	定数の入力一覧	159
表 15-2	コマンド一覧(実行と停止)	160
表 15-3	単項演算子一覧	160
表 16-1	エラーメッセージ (No.100 ~)	165
表 16-2	エラーメッセージ (No.150 ~)	166
表 16-3	エラーメッセージ (No.200 ~)	166
表 16-4	エラーメッセージ (No.300 ~)	167
表 16-5	エラーメッセージ (No.400 ~)	167
表 16-6	エラーメッセージ (No.600 ~)	167
表 16-7	エラーメッセージ (No.650 ~)	168
表 16-8	エラーメッセージ (No.700 ~)	168
表 16-9	エラーメッセージ (No.900 ~)	168
表 16-10	エラーメッセージ (No.1001 ~ (その 1))	169
表 16-11	エラーメッセージ (No.1001 ~ (その 2))	170
表 16-12	エラーメッセージ (No.1070 ~)	170
表 16-13	エラーメッセージ (No.1100 ~)	171
表 16-14	エラーメッセージ (No.1200 ~)	172
表 16-15	エラーメッセージ (No.1250 ~)	173
表 16-16	エラーメッセージ (No.1300 ~)	173
表 16-17	エラーメッセージ (No.1400 ~ (その 1))	174

表 16.-18	エラーメッセージ (No.1400 ~ (その2))	175
表 16.-19	エラーメッセージ (No.1500 ~)	175
表 16.-20	エラーメッセージ (No.1700 ~)	177
表 16.-21	エラーメッセージ (No.2400 ~)	177
表 16.-22	エラーメッセージ (No.5200 ~)	178
表 16.-23	エラーメッセージ (No.5500 ~)	178
表 16.-24	エラーメッセージ (No.5700 ~)	178
表 16.-25	エラーメッセージ (No.5800 ~)	178
表 16.-26	エラーメッセージ (No.5900 ~)	179
表 16.-27	エラーメッセージ (No.10045 ~)	179
表 16.-28	エラーメッセージ (No.10055 ~)	180

セットアップ編

1. 製品パッケージの内容

本章では、PDB38Mの製品パッケージの包装内容を示します。PDB38Mをインストールする前に確認してください。

1.1. 包装製品一覧表

本製品の包装内容を、表 1-1に示します。

表 1-1 PDB38M包装内容一覧表

製品名	数量
フロッピーディスク	1 セット
PDB38MV.1.00 ユーザーズマニュアル	1 冊
リリースノート (ソフトウェア使用権特許契約書を含む)	1 部
お客様登録カード	1 枚

2. 動作環境

PDB38Mは、表 2-1のホストマシンおよび OS バージョン上で動作します。

表 2-1 PDB38Mの動作環境

ホストマシン	IBM PC/AT シリーズ
OS	・ Microsoft Windows Version3.11 英語版 ・ Microsoft Windows Version3.1 日本語版 ・ Windows95 英語 / 日本語版
CPU	i386 以上 (i486DX4-100MHz または Pentium75MHz 以上を推奨)
メモリ	8M バイト (16M バイト以上を推奨)

3. システム構成

PDB38Mを動作させるためには、以下の装置が必要です。

1. ホストマシン
2. エミュレータ
3. エミュレーションポッド
4. ターゲットシステム

エミュレータ本体(M38000T-SBI)のシステム構成図を図 3-1に示します。

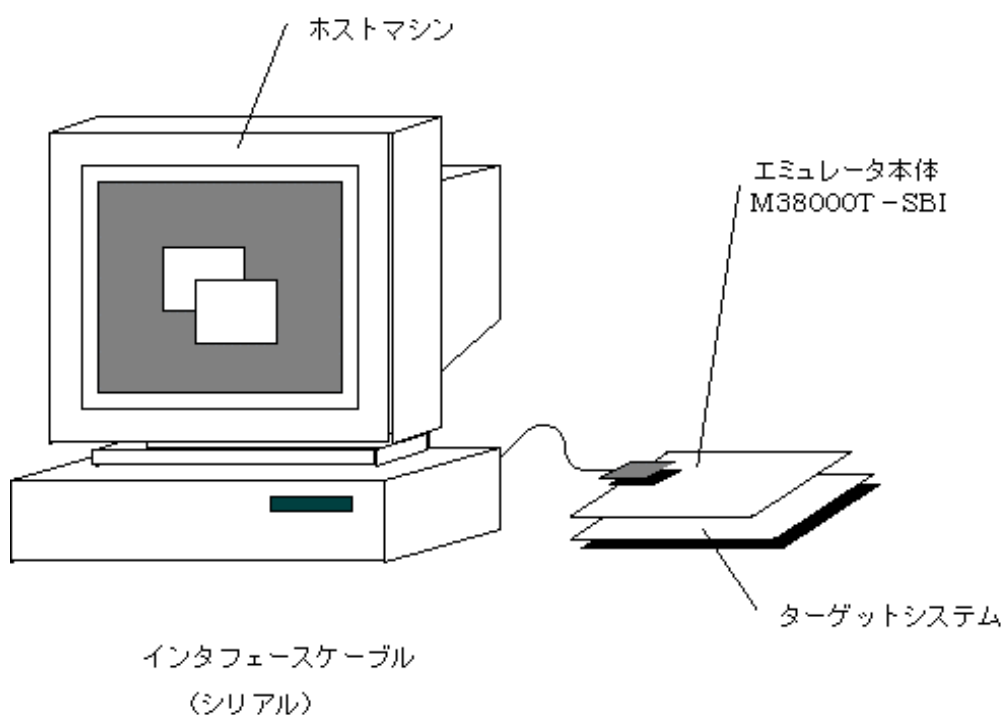


図 3-1 エミュレータ本体(M38000T-SBI)のシステム構成図

3.1.ホストマシン

PDB38Mは、表 3-1のホストマシン上で動作します。

表 3-1 ホストマシン

ホストマシン名	OS
IBM PC/AT シリーズ およびその互換機	・Microsoft Windows Version3.11 英語版 ・Microsoft Windows Version3.1 日本語版 ・Windows95 英語/日本語版

3.2.エミュレータ M38000T-SBI

M38000T-SBIは、三菱 8 ビットマイクロコンピュータ用簡易エミュレータシステムのエミュレータ本体です。M38000T-SBIについては、M38000T-SBIのユーザーズマニュアルを参照してください。

3.3.ターゲットシステム

お客様のターゲットシステムです。簡易エミュレータシステムには、端子処理を行ったターゲットシステムが付属しており、お客様のターゲットシステムの変わりに使用することができます。

3.4.通信方法

エミュレータ本体(M38000T-SBI)とホストマシン間は、RS-232C シリアルインタフェースで接続できます。

RS-232C シリアルインタフェースでは、9600bps の通信速度 (ボーレート) で通信を行います。

4. PDB38Mが扱うファイル

PDB38Mが扱うファイルは、以下の2種類に分類できます。

PDB38Mに入力するファイル（入力ファイル）

1. インテル HEX フォーマットファイル
2. シンボルファイル
3. 定義体ファイル
4. スクリプトファイル
5. 環境設定ファイル

PDB38Mが出力するファイル（出力ファイル）

1. インテル HEX フォーマットファイル
2. 逆アセンブルファイル
3. ログファイル
4. ビューファイル

以下より、各ファイルの詳細について説明します。

4.1.入力ファイル

4.1.1.インテル HEX フォーマットファイル

インテル HEX フォーマットファイルは、機械語を格納しているファイルであり、ファイル属性は、.hexです。アセンブラ SRA74M 用のリンカ LINK74M により生成されます。

ソースレベルのデバッグを行う場合は、シンボルファイルを読み込む必要があります。

4.1.2.シンボルファイル

シンボルファイルは、シンボル情報、行番号情報等のデバッグ情報を格納しているファイルです。ファイル属性は、.symです。シンボルファイルは、LINK74M 実行時に “-s” オプションを指定することにより生成されます。

4.1.3.定義体ファイル

ターゲット MCU の固有情報を持つ情報ファイルです。定義体ファイルは、製品に付属されています。ファイル名は、M3XXXX.I38です。

PDB38M起動時（環境設定を行う場合）に、ご使用になる MCU に対応した定義体ファイル名を指定していただく必要があります。この定義体ファイルの指定がない場合、PDB38M は起動できません。定義体ファイルは、環境設定の際に Init ダイアログで指定します。詳細は、セットアップ編の「Init ダイアログの操作方法」を参照してください。

4.1.4.スクリプトファイル

スクリプトコマンドをファイルから自動実行するためのファイルです。スクリプトコマンドとは、スクリプトウィンドウ内で実行することができるコマンドのことです。

スクリプトファイルは、お手持ちのエディタで作成してください。ファイル名およびファイル属性には、任意の名前が使用できますが、ファイル属性.scr をご使用いただくことをお勧めします（スクリプトファイルを選択するダイアログの表示では、ファイル属性.scr のファイルを優先して表示します）。

スクリプトファイルは、スクリプトウィンドウをアクティブにし、メニュー [Option] [Script] [Open] の選択でオープンします。その後、メニュー [Option] [Script] [Run] で実行できます。

4.1.5.環境設定ファイル

PDB38M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、PDB38Mが自動的に作成します。環境設定ファイルは、ユーザ自身が作成/編集することはできません。

4.2.出力ファイル

4.2.1.インテル HEX フォーマットファイル

PDB38Mは、指定したアドレス範囲の内容をインテル HEX フォーマットファイルに保存できます（アップロード）。

保存されたインテル HEX フォーマットファイルは、再ダウンロードすることができます。

4.2.2.逆アセンブルファイル

プログラムメモリの逆アセンブル結果を保存したテキストファイルです。メニュー [File] [Save Disasm] で保存します。

なお、逆アセンブルファイルは参照用のテキストファイルであり、再アセンブルや再ダウンロードはできません。

図 4-1に逆アセンブルファイル例を示します。

```

C000      78      RESET:   SEI
C001      58              CLI
C002      78              SEI
C003      32              SET
C004      F8              SED
C005      12              CLT
C006      D8              CLD
C007      3C04FB         LDM      #04H,FBH :CPUMOD
C00A      A2BF         LDX      #BFH
C00C      9A              TXS
C00D      3CFFC1         LDM      #FFH,C1H :P0D
C010      3C00C0         LDM      #00H :P_PORT,C0H :P0
C013      3CFFC3         LDM      #FFH,C3H :P1D
C016      3C01C2         LDM      #01H :TIME00,C2H :P1
C019      3C00C5         LDM      #00H :P_PORT,C5H :P2D
C01C      3CFFC9         LDM      #FFH,C9H :P4D
C01F      3C01C8         LDM      #01H :TIME00,C8H :P4

```

図 4-1 逆アセンブルファイル例

ファイルの内容は、左端から下記の内容を出力しています。

- アドレス
- オブジェクトコード
- アドレスに相当するラベル
- 逆アセンブル結果

4.2.3.ログファイル

スクリプトウィンドウでの、スクリプトコマンドの実行結果を保存したテキストファイルです。ファイル属性は、.log です

スクリプトウィンドウをアクティブにし、メニュー [Option] [Log] [On] でログファイル名を指定すると、それ以降のスクリプトコマンドの実行結果をログファイルに保存できます。メニュー [Option] [Log] [Off] でログファイルへの保存を終了します。

4.2.4.ビューファイル

スクリプトウィンドウの現在の表示内容を保存したテキストファイルです。スクリプトウィンドウ上でスクロール可能な表示内容のすべてを保存します。

ビューファイルの内容は、ログファイルと同様です。ログファイルはログファイルの出力

を指定した後のコマンドの実行結果を保存するのに対し、ビューファイルは既に実行したコマンドの実行結果が保存できます。ただし、スクロール表示可能な範囲に制限されます（それ以前の古い内容は捨てられますので保存できません）。

スクリプトウィンドウをアクティブにし、メニュー [Option] [View] [Save] でビューファイル名を指定すると、スクリプトウィンドウの表示内容をビューファイルに保存できます。

ウィンドウ操作編

5. PDB38Mウィンドウの操作方法

5.1.PDB38Mウィンドウの概要

PDB38Mウィンドウは、PDB38Mのメインウィンドウです。PDB38Mを起動した際に、オープンします。本ウィンドウから、各種ウィンドウの起動や表示、ターゲットプログラムの実行、ブレークポイントの設定などの様々なデバッグ機能が使用できます。

PDB38Mウィンドウの機能を以下に示します。

- 各ウィンドウの制御
プログラムウィンドウ、レジスタウィンドウなど、各種ウィンドウの制御を行います。
- エミュレータの制御
 - ターゲットプログラムのダウンロード
 - ターゲットプログラムの実行/停止
 - ステップ実行
 - ブレークポイントの設定/解除

5.2.PDB38Mウィンドウの構成

図 5-1にPDB38Mウィンドウの構成図を示します。

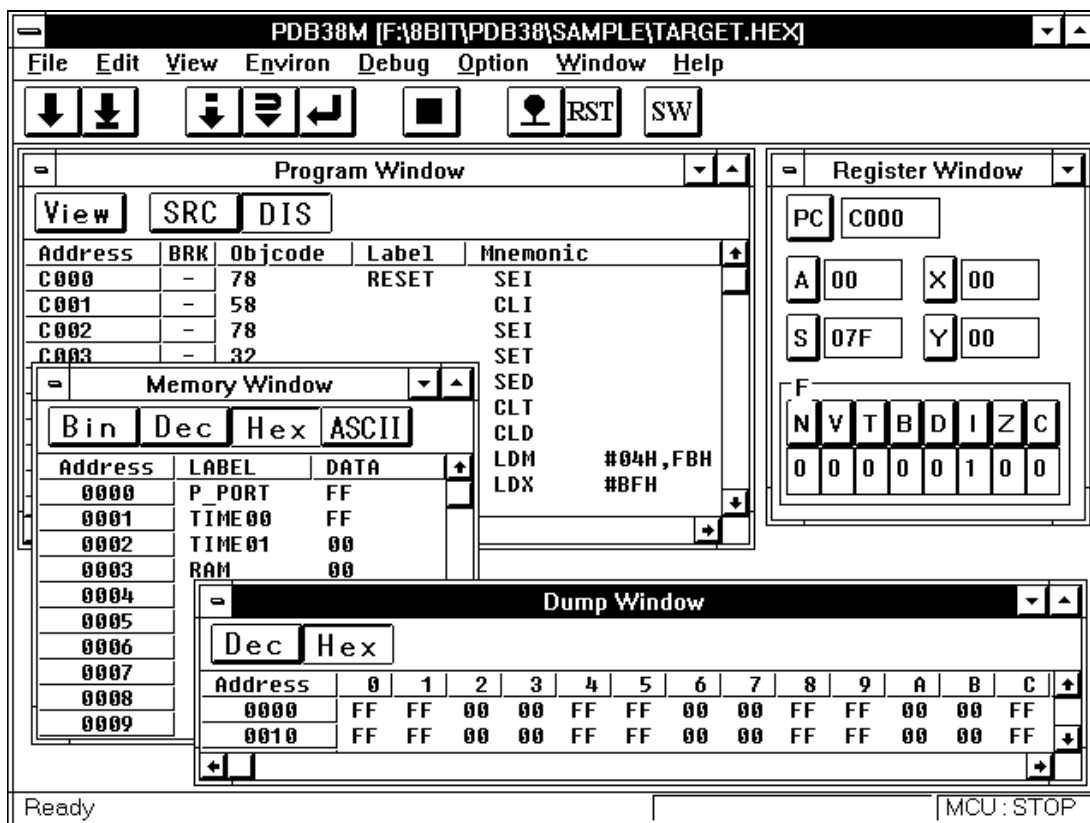


図 5-1 PDB38Mウィンドウの構成図

PDB38Mウィンドウの各部品について、以下で説明します。

5.2.1.メニュー

各種コマンドを発行するためのメニューを備えます。メニューの表示例を図 5-2に示します。

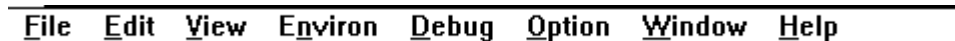


図 5-2 メニューの表示例

PDB38M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

PDB38Mのメニュー項目のうち Option メニューについては、PDB38M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、PDB38M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。

- 基本メニューの各項目の機能と操作方法についての詳細は、本章の「基本メニューの操作方法」で説明します。
- 拡張メニューの各項目の機能と操作方法についての詳細は、各ウィンドウの操作方法の「拡張メニューの操作方法」で説明します。

5.2.2.ツールバー

PDB38Mでは、使用頻度の高いコマンドをより簡単に実行できるようにツールバーを備えています。

PDB38Mウィンドウのツールバーの表示例を図 5-3に示します。



図 5-3 PDB38Mウィンドウのツールバー表示例

ツールバーの各ボタンの機能についての詳細は、本章の5.4節「ツールバーの構成」で説明します。

5.2.3.メイン表示領域

メイン表示領域は、PDB38Mウィンドウから起動した各種ウィンドウを表示する領域です。各種ウィンドウはこの領域外に配置することはできません。

5.2.4.各種ウィンドウ

PDB38Mの Window メニューで起動したウィンドウです。これらのウィンドウはメイン表示領域内に表示されます。

5.2.5.ステータスバー

ステータスバーには、操作を補助するための情報、ターゲットの実行状態を表示します。ステータスバーの表示例を図 5-4に示します。

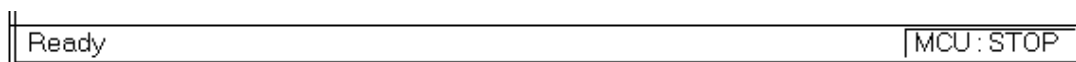


図 5-4 ステータスバーの表示例

ステータスバーの各領域の表示内容は以下のとおりです。

- a. 説明表示領域
ウィンドウ操作時の説明およびメニューの説明等を表示します。
- b. 実行状態表示領域
現在のターゲットの実行状況を表示します。
ターゲット実行中 (Come、Step、Over、Return 実行を含む) は、“RUN”と表示します。ターゲット停止中は、“STOP”と表示します。


5.3.基本メニューの操作方法

PDB38Mウィンドウのメニューの構成を、表 5-1および表 5-2に示します。

表 5-1 PDB38Mウィンドウのメニューの構成 (その1)

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 機械語データとデバッグ情報をダウ ンロード	Shift + F.1	
	Load Module... Memory Image... Symbol...	機械語データのみダウンロード デバッグ情報のみダウンロード		
	Upload...	ターゲットプログラムのアップロード		
	Save Disasm...	逆アセンブル結果の保存		
	Exit	PDB38Mを終了		
	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		
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		
Environ	Init...	環境設定		
	Path...	ソースファイルのサーチパス		
	Start Up...	スタートアップ関数の設定		
Debug	Go	ターゲットプログラムの実行	F.1	
	Go Go Option...	現 PC からの実行 指定アドレスからの実行		
	Come	カーソル位置までの実行		
	Step	ステップ実行	F.3	
	Step Step Option...	一回のステップ実行 指定回数のステップ実行		
	Over	オーバーステップ実行	F.4	
	Over Over Option...	一回のオーバーステップ実行 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	F.7	
	S/W Break Point...	S/W ブレークポイント設定ダイアロ グをオープン カーソル位置にソフトウェアブレー クを設定/解除		
	Break			
Reset	ターゲットのリセット	F.8		

表 5-2 PDB38Mウィンドウのメニューの構成 (その2)

メニュー	メニュー項目	機能	ショート カットキー
Debug	Stop	ターゲットプログラムの実行停止	
Option		(各ウィンドウの拡張メニューが追加されます。)	
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	アイコンを整理	
	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	ウォッチ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		
Help	About...	バージョン情報表示	

以下に、基本メニューの操作方法を説明します。

5.3.1. Fileメニュー

Fileメニューには、ファイル読み込み、ファイル保存、PDB38Mの終了に関連するメニューが割り当てられています。以下では、Fileメニューの下に割り当てられている各種メニューの機能について説明します。


□ Downloadメニュー

機械語データとデバッグ情報をダウンロードします。

Downloadメニューを選択すると、次の3つのサブメニューが表示されますので、目的に応じて選択してください。

Load Module サブメニュー

機械語データとデバッグ情報の両方をダウンロードします。

 Optionメニューのメニュー項目(拡張メニュー)については、各ウィンドウ操作方法の拡張メニューの表を参照してください。

通常のダウンロードでは、この Load Module サブメニューをご使用ください。

Memory Image サブメニュー

機械語データのみダウンロードします。

アップロード機能（メニュー [File] [Upload]）で保存した機械語データファイルを再ダウンロードする場合等にご使用ください。

この場合、ラベル・シンボル・ソースファイルは表示できませんのでご注意ください。

Symbol サブメニュー

デバッグ情報のみダウンロードします。

上記の何れかのサブメニューを選択すると、ファイルセレクションダイアログがオープンします。図 5-3にファイルセレクションダイアログの構成図を示します。ダウンロードするファイル名を選択してください。

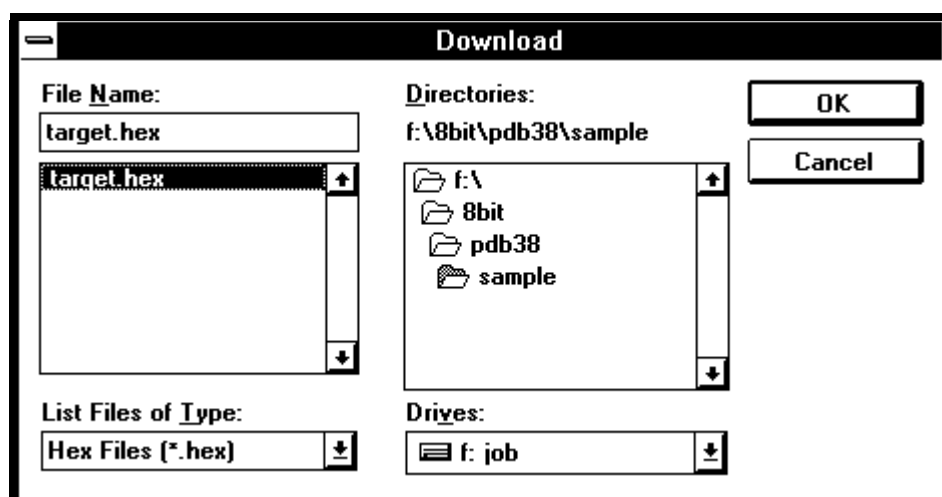


図 5-5 ファイルセレクションダイアログ

- Load Module サブメニューのダイアログでは、インテル HEX フォーマットファイル名（拡張子が.hexのファイル名）を選択してください。
- Memory Image サブメニューのダイアログでは、インテル HEX フォーマットファイル名（拡張子が.hexのファイル名）を選択してください。
- Symbol サブメニューのダイアログでは、シンボルファイル名（拡張子が .symのファイル名）を選択してください。
- OK ボタンをクリックすると、指定ファイルのダウンロードを開始します。ダウンロード中は、図 5-6のダイアログを表示します。このダイアログの Cancel ボタンをクリックすると、ダウンロードを中断できます。なお、ダウンロードが終了すれば、このダイアログは自動的に閉じます。

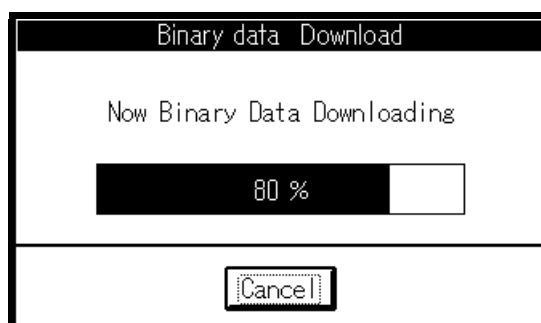


図 5-6 ダウンロード中のダイアログ表示例

注意事項

PDB38Mウィンドウのメイン表示領域にソースウィンドウを表示した状態でダウンロードを開始した場合、ダウンロード終了時にソースウィンドウのみクローズします。

□ Upload メニュー

プログラムのメモリ内容を機械語データファイルに保存します（アップロード）。

Upload メニューを選択すると Upload ダイアログがオープンします。図 5-7 にアップロードダイアログの構成図を示します。保存するファイル名、メモリ領域の開始・終了アドレスを指定してください。

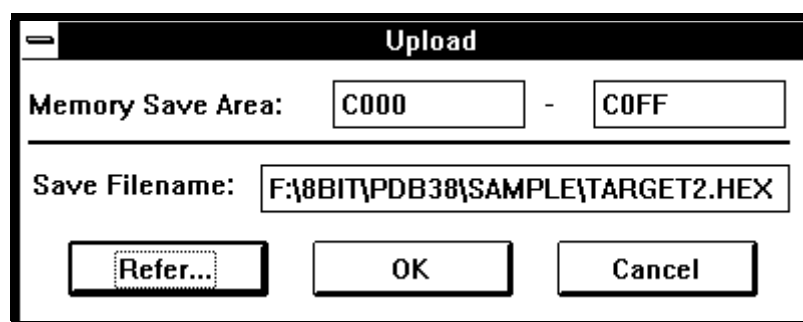


図 5-7 Upload ダイアログ

- Memory Save Area:領域に、メモリ領域の開始アドレスと終了アドレスを入力します。Save Filename:領域に、保存するファイル名を入力します。
- メモリ領域の開始・終了アドレスには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」を参照してください。

なお、このときの定数の基数は、16 進数です。

- ファイル名の指定には、パス名を付加してください。
- ファイル名の、ファイル属性には .hexを指定してください。インテル HEX フォーマットファイルが生成されます。
- 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。

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。ファイルセレクションダイアログによるファイル名指定も可能です。図 5-8にファイルセレクションダイアログの構成図を示します。

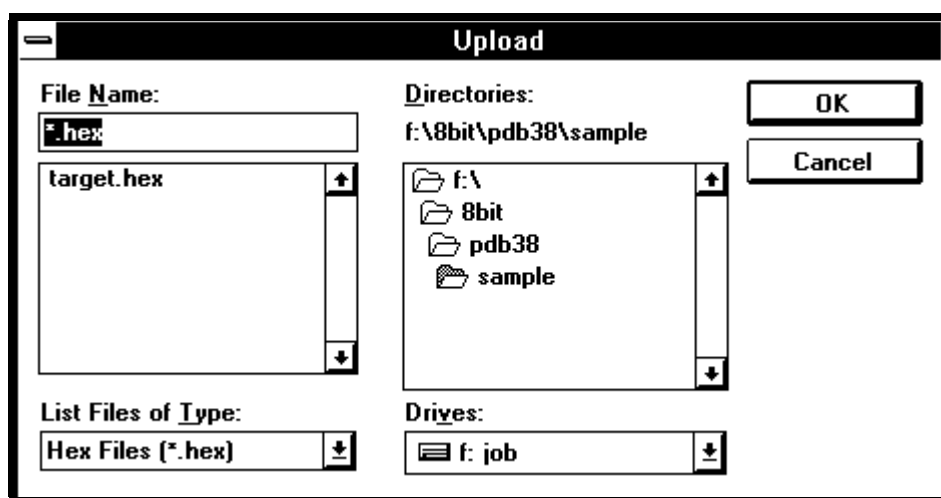


図 5-8 ファイルセレクションダイアログ

- OK ボタンをクリックすると、指定ファイルへのアップロードを開始します。アップロード中は、図 5-9のダイアログを表示します。アップロードが終了すれば、このダイアログは自動的に閉じます。

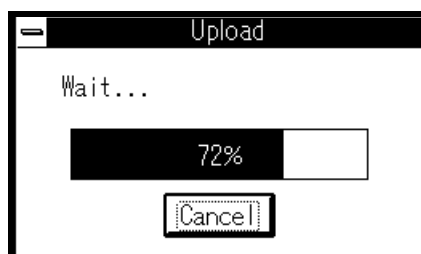


図 5-9 アップロード中のダイアログ表示例

- 保存するメモリ範囲が広範囲の場合、保存するために大きなディスク容量と時間を必要とします。この場合、図 5-9のダイアログの Cancel ボタンをクリックする

ことにより、保存を中止できます。このとき途中までファイルが作成されます。

□ Save Disasm メニュー

プログラムメモリの逆アセンブル結果を、逆アセンブルファイルとして、テキストファイルに保存します。逆アセンブルファイルは、参照用のテキストファイルであり、再アセンブルや再ダウンロードはできません。

Save Disasm メニューを選択すると、Save ダイアログがオープンします。図 5-10 に Save ダイアログの構成図を示します。保存するファイル名、開始・終了アドレスを指定してください。

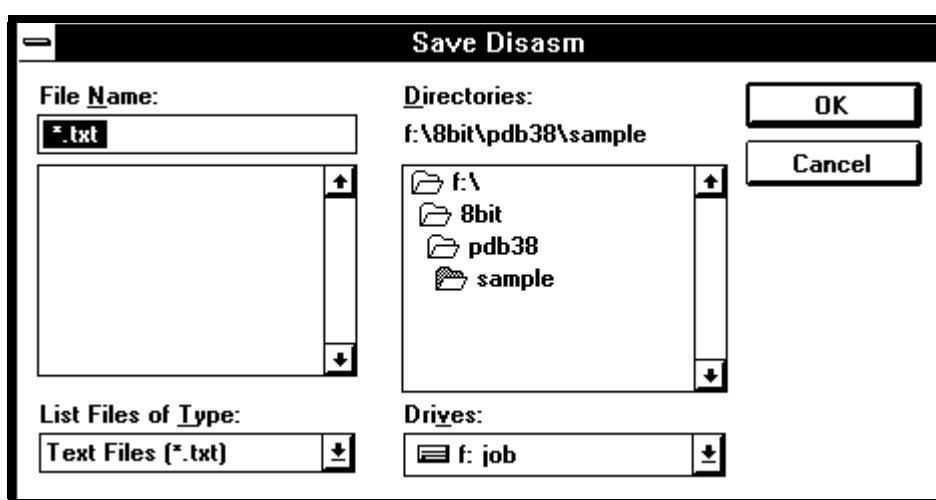


図 5-10 Save ダイアログ

- Memory Save Area:領域に、メモリ領域の開始アドレスと終了アドレスを入力します。Save Filename:領域に、保存するファイル名を入力します。
- メモリ領域の開始・終了アドレスには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」を参照してください。
なお、このときの定数の基数は、16進数です。
- ファイル名の指定には、パス名を付加してください。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。
- 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。
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。ファイルセレクションダイアログによるファイル名指定も可能です。図 5-11にファイルセレクションダイアログの構成図を示します。

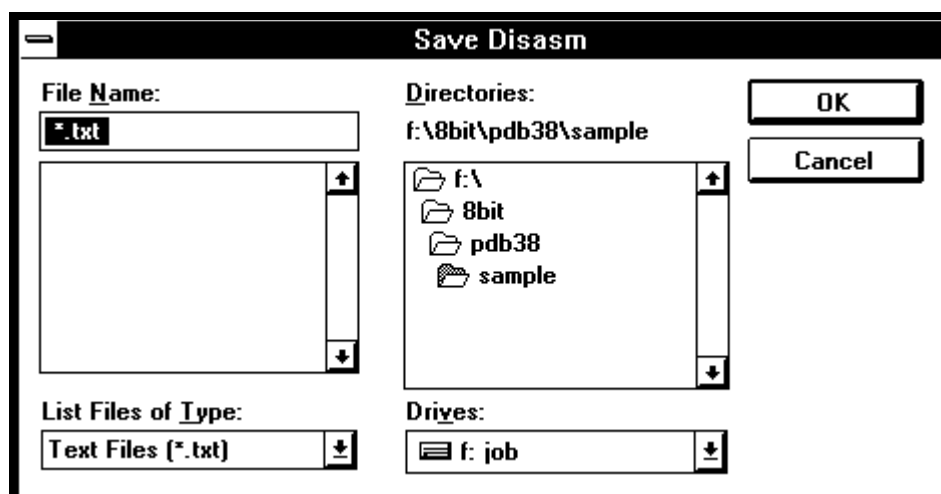


図 5-11 ファイルセレクションダイアログ

- OK ボタンをクリックすると、指定ファイルへ逆アセンブル結果の保存を開始します。保存中は、図 5-12のダイアログを表示します。保存が終了すれば、このダイアログは自動的に閉じます。

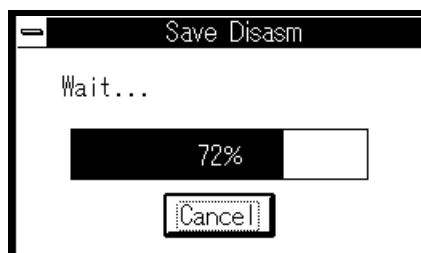


図 5-12 逆アセンブル結果保存中のダイアログ表示例

- 保存するメモリ範囲が広範囲の場合、保存するために大きなディスク容量と時間を必要とします。この場合、図 5-12のダイアログの Cancel ボタンをクリックすることにより、保存を中止できます。このとき途中までファイルが作成されます。

□ Exit メニュー

PDB38Mを終了します。

Exit メニューを選択すると、ウォーニング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。なお、ここで Cancel ボタンをクリックすると、PDB38Mは終了しません。図 5-13 にウォーニングダイアログの構成図を示します。

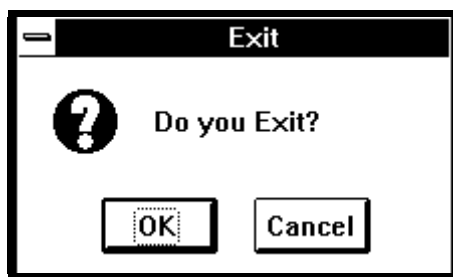


図 5-13 ウォーニングダイアログ

5.3.2. Editメニュー

Editメニューには、文字列のコピー、ペースト、検索を行うメニューが割り当てられています。以下では、Editメニューの下に割り当てられている各種メニューの機能について説明します。

□ Copyメニュー

マウスで選択した文字列をクリップボードにコピーします。

□ Pasteメニュー

クリップボードの文字列をスクリプトウィンドウに貼り付けます。

□ Findメニュー

プログラムウィンドウおよびソースウィンドウに表示しているソースファイルから指定した文字列を検索します。

Findメニューを選択すると、Findダイアログがオープンします。図 5-14 に、Findダイアログの構成図を示します。検索文字列および検索方向を指定してください。

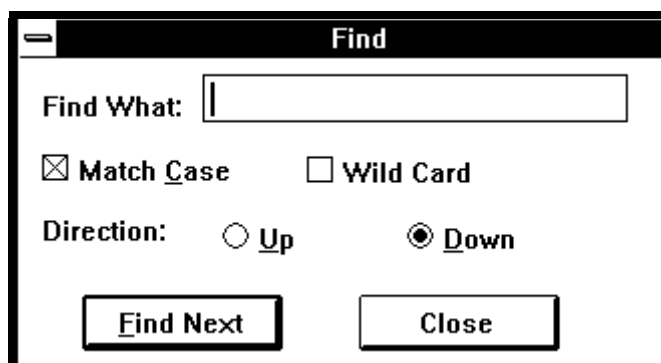


図 5-14 Findダイアログ

- Find What:領域に、検索文字列を入力します。
- [Match Case] チェックボックスは、大文字/小文字の区別を行うか否かを指定します。このチェックボックスをオンにすると、大文字/小文字を区別して検索します。例えば検索文字列として“ABC”を指定した場合、“abc”や“AbC”等は検索の対象となりません。“abc”や“AbC”等も検索の対象とする場合は、このチェックボックスをオフにしてください。
- [Wild Card] チェックボックスは、ワイルドカードを使用するか否かを指定します。このチェックボックスをオンにすると、ワイルドカードを使用した検索文字列の指定が可能になります。使用できるワイルドカードは、半角のアスタリスク(*)およびクエスチョンマーク(?)です。

ワイルドカード	説明
*	任意の文字列(0文字以上)を表します。
?	任意の1文字を表します。

なおこれらの特殊文字を通常の文字として検索したい場合は、特殊文字の前に‘\’文字を付加してください。

また、ワイルドカードを使用しない場合は、[Wild Card] チェックボックスをオフにしてください。

- 検索方向は、Direction:領域内の Up ラジオボタン(現在のカーソル位置から上(文頭)に向かって検索)、または Down ラジオボタン(現在のカーソル位置から下(文末)に向かって検索)が指定できます。
- 検索開始位置(カーソル位置)は、プログラムウィンドウまたはソースウィンドウのプログラム表示領域を、マウスでシングルクリックすることにより指定できます。

図 5-15にプログラム表示領域のカーソル表示例を示します。

Line	BRK	Source
00050	-	RESET: ;
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER
00059	-	LDX #0BFH ;STACK POINT SET

図 5-15 プログラム表示領域のカーソル表示例

なお、カーソル位置の指定がない場合は、ファイルの先頭行が検索開始位置になります。

- Find Next ボタンをクリックすると、指定方向に向かって、順次検索を開始します。

5.3.3. Viewメニュー

View メニューには、ツールバーおよびステータスバーの表示切り替えを行うメニューが割り当てられています。以下では、View メニューの下に割り当てられている各種メニューの機能について説明します。

□ Tool Bar メニュー

メイン表示領域内のアクティブなウィンドウが所有するツールバーの表示/非表示を切り替えます。

なお、PDB38Mウィンドウが所有するツールバーは非表示にすることはできません。

□ Status Bar メニュー

ステータスバーの表示/非表示を切り替えます。

5.3.4. Environメニュー

Environ メニューには、動作環境の設定等を行うメニューが割り当てられています。以下では、Environ メニューの下に割り当てられている各種メニューの機能について説明します。

□ Init メニュー

PDB38Mの動作環境を設定します。

Init メニューでは、以下の動作環境が設定できます。

- 定義体ファイルの設定
ターゲットの MCU に対応する定義体ファイルを指定します。
- 起動時に実行するスクリプトファイルの設定
PDB38M起動時に実行されるスクリプトファイルを指定します。この設定を行うと、PDB38M起動時にスクリプトウィンドウをオープンし、指定したスクリプトファイルを実行します。
- シリアル通信 I/F の設定
シリアル通信に関する設定を行います。

- デフォルトタブ値の設定
PDB38M起動時の、プログラムウィンドウおよびソースウィンドウのソース表示におけるタブ値を指定します。
- デフォルトフォントサイズの設定
PDB38M起動時の、各ウィンドウのフォントおよびサイズを指定します。

Init メニューを選択すると、Init ダイアログがオープンします。図 5-16 に Init ダイアログの構成図を示します。各動作環境を設定してください。

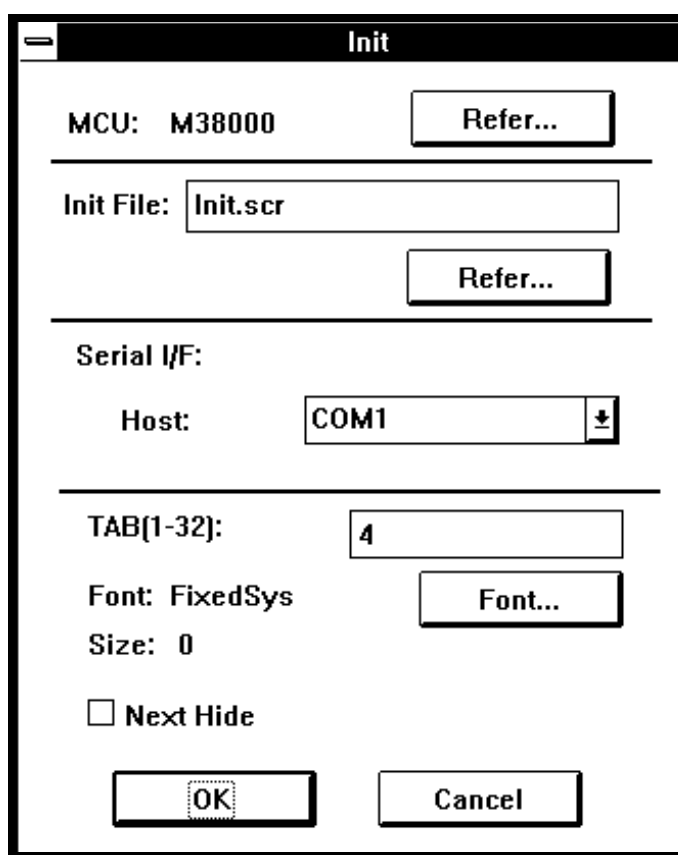


図 5-16 Init ダイアログ

- この Init ダイアログは、PDB38M起動時にオープンする Init ダイアログと同じものです。Init ダイアログの操作方法の詳細は、セットアップ編の「Init ダイアログの操作方法」を参照してください。
- 本メニューから Init ダイアログをオープンし、動作環境の設定を行った場合、その内容は、次回PDB38Mを起動した際に有効になります。ただし、タブ値とフォントサイズの変更内容は、OK ボタンをクリックした直後から有効になります。

□ Path メニュー

ソースファイルのサーチパス名を指定します。デバッグ対象の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無い場合、また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、サーチパス名を指定します。サーチパス名は、複数指定できます。

PDB38Mは、プログラムウィンドウおよびソースウィンドウにソースファイルを表示する際、下記の優先順位でディレクトリを検索します。

1. デバッグ情報に含まれるパス名
2. Path メニューで指定したパス名（指定順）
3. カレントディレクトリ

Path メニューを選択すると、Path ダイアログがオープンします。図 5-17 に Path ダイアログの構成図を示します。

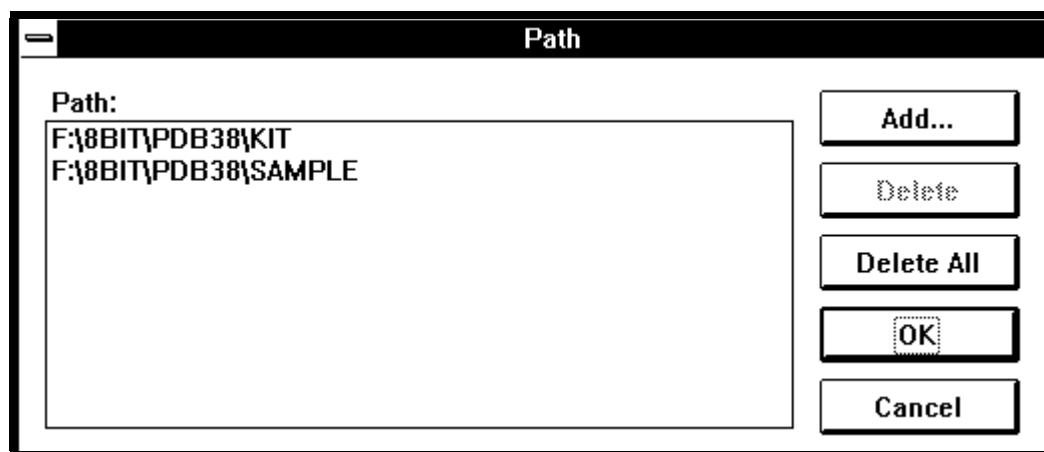


図 5-17 Path ダイアログ

- Path:領域には、現在設定されている全てのサーチパス名が表示されます。
- サーチパスを追加する場合は、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、任意のファイル名を選択してください。ファイルセレクションダイアログで指定したファイルの存在するディレクトリが、サーチパスとして追加されます。図 5-18にファイルセレクションダイアログを示します。

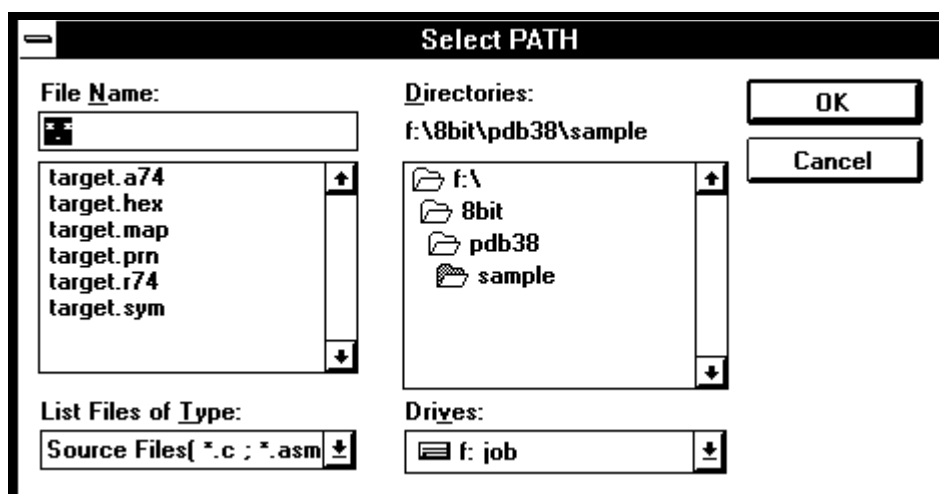


図 5-18 ファイルセレクションダイアログ

- サーチパスを削除する場合は、Path:領域に表示されているサーチパス名をマウスのシングルクリックで選択し、Delete ボタンをクリックしてください。
- また、すべてのサーチパスを削除する場合は Delete All ボタンをクリックしてください。Delete All ボタンをクリックすると、クエスチョン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。なお、ここで Cancel ボタンをクリックすると、サーチパスは削除しません。図 5-19にクエスチョンダイアログの構成図を示します。

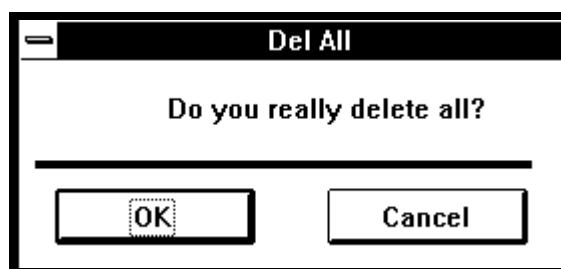


図 5-19 クエスチョンダイアログ

□ StartUp メニュー

デバッグ実行開始位置の設定を行います。

ターゲットプログラムをダウンロードした直後の PC 値に相当するソース行頭がない場合（例えば、スタートアッププログラムを記述したモジュールにデバッグ情報がない場合）に、StartUp メニューで示したラベルに対応するソース行を、プログラムウィンドウのプログラム表示領域に表示します。

ただし、プログラムウィンドウに StartUp メニューで指定したソースを表示しても、その時の PC 値はダウンロードした直後の PC 値のままです。その場合、表示中のソースまで PC 値を進めるには、カム実行（メニュー [Debug] [Come] の説明を参照）を行うようにしてください。

注意事項

ターゲットプログラムのダウンロード直後の PC 値に相当するソース行頭がある場合は、この StartUp の設定は無視されます。

StartUp メニューを選択すると、Startup ダイアログがオープンします。図 5-20に、Startup ダイアログの構成図を示します。デバッグ実行開始位置となるラベル、またはシンボルを設定してください。

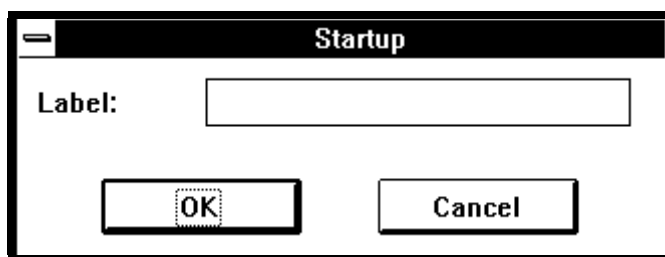


図 5-20 StartUp ダイアログ

- Label:領域に、ラベル、またはシンボルを入力します。
- 一度 Startup ダイアログでデバッグ開始位置の設定を行うと、その情報が環境設定ファイルに保存されます。その内容は、次回以降にPDB38Mを起動した際にも有効になります。

5.3.5. Debugメニュー

Debug メニューには、ターゲットプログラムの実行/停止やステップ実行などのデバッグに関連するメニューが割り当てられています。以下では、Debug メニューの下に割り当てられている各種メニューの機能について説明します。

□ Goメニュー

ターゲットプログラムを実行します。

Go メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Go サブメニュー

現在のプログラムカウンタから、ターゲットプログラムを実行します。

Go Option サブメニュー

指定したアドレスから、ターゲットプログラムを実行します。

Go Option サブメニューを選択すると、Go ダイアログがオープンします。図 5-21 に Go ダイアログの構成図を示します。ターゲットプログラムの実行開始位置を設定してください。

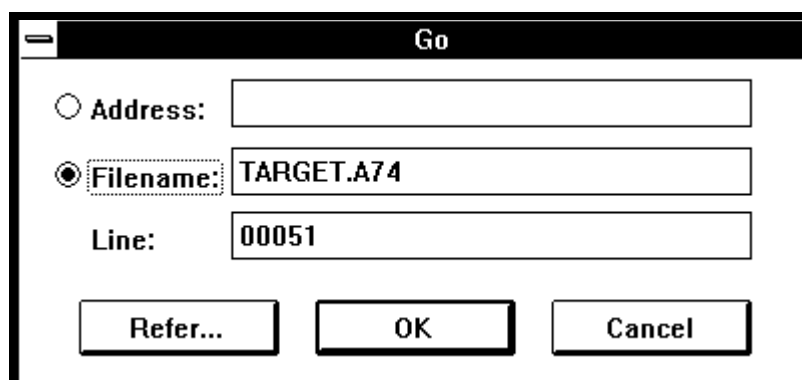


図 5-21 Go ダイアログ

- ターゲットプログラムの実行開始位置は、アドレスまたはソースファイル名 + 行番号で指定できます。
- ターゲットプログラムの実行開始位置をアドレスで指定する場合は、Address ラジオボタンをクリックし、Address:領域に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開始位置をソースファイル名 + 行番号で指定する場合は、Filename ラジオボタンをクリックし、Filename:領域にソースファイル名を、Line:領域に行番号を入力してください。
- Refer ボタンをクリックすると、ファイルセクションダイアログがオープンします。ソースファイル名の指定は、このファイルセクションダイアログから行うこともできます。図 5-22にファイルセクションダイアログの構成図を示します。

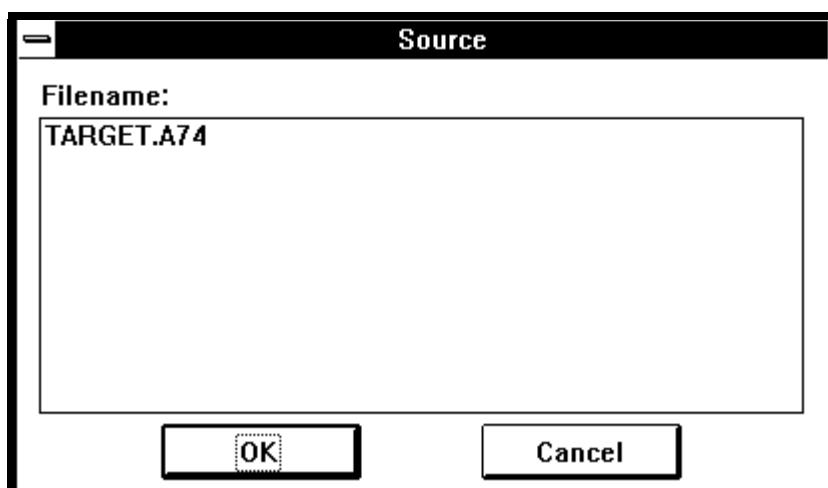


図 5-22 ファイルセレクションダイアログ

- 行番号は、アドレス情報付きの行番号を指定してください。
- アドレス、行番号の指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
なお、このときの定数の基数は、アドレスは 16 進数、行番号は 10 進数です。

Go メニューで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か、ストップコマンド（メニュー [Debug] [Stop] 等）を実行することによって停止します。

□ Come メニュー

現在のプログラムカウンタから、プログラムウィンドウまたはソースウィンドウのプログラム表示領域のカーソル位置までターゲットプログラムを実行します（カム実行）。プログラムカウンタがカーソル位置に達した時点で停止します。ただし、ソフトウェアブレー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。

また、現在設定されているブレークポイントは有効です。したがって、カーソル位置に達するまでにブレーク条件を満たした場合は、その時点で停止します。また、実行の途中で停止する場合は、ストップコマンド（メニュー [Debug] [Stop] 等）を実行してください。

なお、カーソル位置は、プログラムウィンドウまたはソースウィンドウのプログラム表示領域をマウスでシングルクリックすることにより指定できます。図 5-23 にプログラム表示領域のカーソル表示例を示します。

Line	BRK	Source
00050	-	RESET: ;
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER
00059	-	LDM #0BFH ;STACK POINT SET

図 5-23 プログラム表示領域のカーソル表示例

□ Step メニュー

ステップ実行を行います。

プログラムウィンドウがソースプログラムを表示している場合は、ソース行単位のステップ実行となります。逆アセンブルを表示している場合は、命令単位のステップ実行となります。サブルーチン呼び出し命令をステップ実行した場合は、サブルーチンの内部に入ってステップ実行を行います。

Step メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Step サブメニュー

一回分、ステップ実行を行います。

Step Option サブメニュー

指定した回数分、ステップ実行を行います。

Step Option サブメニューを選択すると、Step ダイアログがオープンします。図 5-24 に Step ダイアログの構成図を示します。ステップ回数を設定してください。

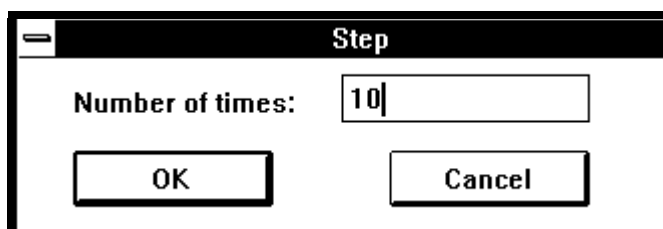


図 5-24 Step ダイアログ

- Number of times:領域に、ステップ回数を入力します。

- ステップ回数の指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
なお、このときの定数の基数は、10進数です。

なお、ステップ実行を途中で停止する場合は、ストップコマンド（メニュー [Debug] [Stop] 等）を実行してください。

□ Over メニュー

オーバーステップ実行を行います。

プログラムウィンドウがソースプログラムを表示している場合は、ソース行単位のオーバーステップ実行となります。逆アセンブルを表示している場合は、命令単位のオーバーステップ実行となります。ステップ実行と異なる点は、サブルーチン呼び出し命令をステップ実行した場合、サブルーチン内部をステップ実行せずに、次の命令までを1ステップとして実行する点です。

Over メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Over サブメニュー

一回分、オーバーステップ実行を行います。

Over Option サブメニュー

指定した回数分、オーバーステップ実行を行います。

Over Option サブメニューを選択すると、Over ダイアログがオープンします。図 5-25 に Over ダイアログの構成図を示します。オーバーステップ回数を設定してください。

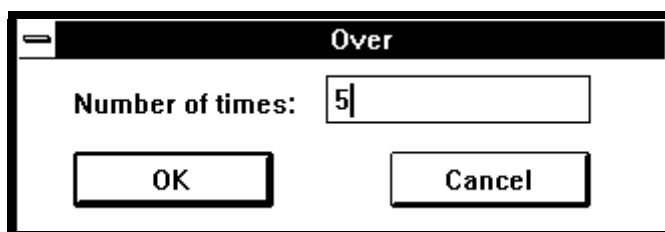


図 5-25 Over ダイアログ

- Number of times:領域に、オーバーステップ回数を入力します。
- オーバーステップ回数の指定には、式が記述できます。式の仕様は、スクリプ

ト仕様編の「式の記述方法」に記載されています。

なお、このときの定数の基数は、10 進数です。

なお、オーバーステップ実行を途中で停止する場合は、ストップコマンド（メニュー [Debug] [Stop] 等）を実行してください。

□ Return メニュー

現在のサブルーチンから復帰します。現在のサブルーチンの呼び出し元に戻るまで、ステップ実行を繰り返します。

プログラムウィンドウがソースプログラムを表示している場合は、ソース行単位のリターン実行となります。逆アセンブルを表示している場合は、命令単位のリターン実行となります。

なお、リターン実行を途中で停止する場合は、ストップコマンド（メニュー [Debug] [Stop] 等）を実行してください。

□ Break Point メニュー

ソフトウェアブレークポイントおよびハードウェアブレークポイントを設定します。Break Point メニューを選択すると、以下のサブメニューが表示されます。

S/W Break Point サブメニュー

S/W ブレークポイント設定ダイアログをオープンします。

S/W ブレークポイント設定ダイアログでは、以下の操作が行えます。

- ソフトウェアブレークポイントの参照
- ソフトウェアブレークポイントの設定
- ソフトウェアブレークポイントの解除
- ソフトウェアブレークポイントの有効化
- ソフトウェアブレークポイントの無効化

なお、ソフトウェアブレークポイントとは、指定したアドレスの命令を実行する手前でブレークするブレークポイントのことです。

S/W Break Point サブメニューを選択すると、S/W ブレークポイント設定ダイアログがオープンします。図 5-26に S/W ブレークポイント設定ダイアログの構成図を示します。

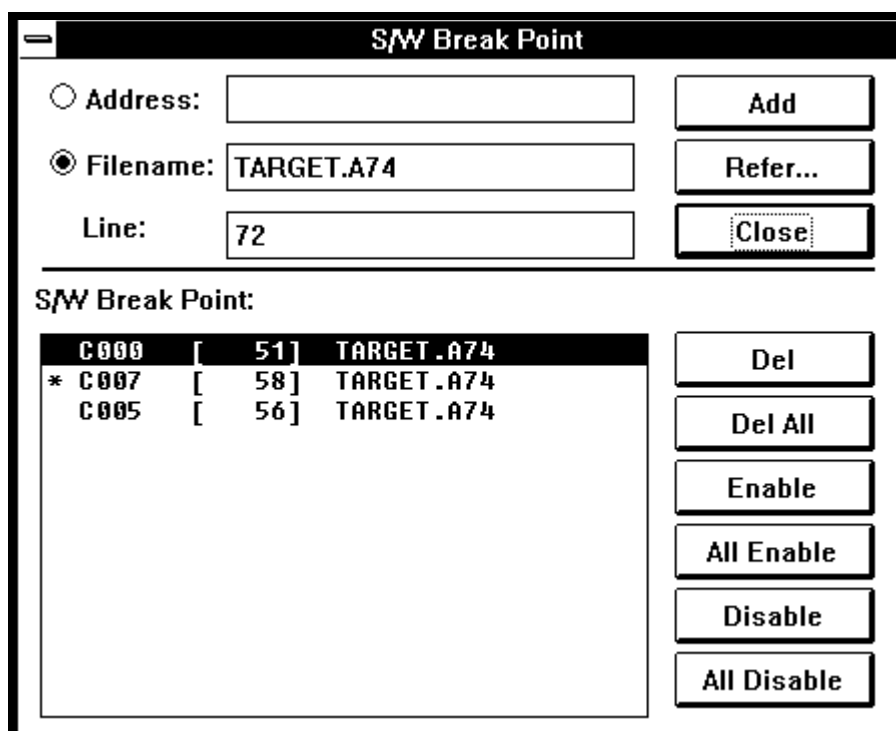


図 5-26 S/W ブレークポイント設定ダイアログ

- S/W ブレークポイント設定ダイアログの操作方法については、ウィンドウ操作編の「S/W ブレークポイント設定ダイアログの操作方法」を参照してください。

Break サブメニュー

プログラムウィンドウまたはソースウィンドウのプログラム表示領域のカーソル位置に、ソフトウェアブレークポイントを設定/解除します。

なお、カーソル位置は、プログラムウィンドウまたはソースウィンドウのプログラム表示領域をマウスでシングルクリックすることにより指定できます。図 5-27にプログラム表示領域のカーソル表示例を示します。

Line	BRK	Source
00050	-	RESET: ;
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER
00059	-	LDX #0BFH ;STACK POINT SET

図 5-27 プログラム表示領域のカーソル表示例

- ソフトウェアブレークポイントが設定されていない行にカーソルを移して本メニューを選択すると、ソフトウェアブレークポイントが設定されます。
- すでにソフトウェアブレークポイントが設定されている行にカーソルを移して本メニューを選択すると、設定されているソフトウェアブレークポイントが解除されます。
- ソフトウェアブレー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）にカーソルを移した場合、本メニューを選択することはできません。

□ Reset メニュー

ターゲットをリセットします。

□ Stop メニュー

ターゲットプログラムの実行を停止します。

5.3.6. Optionメニュー

Option メニューには、PDB38M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特有のメニュー（拡張メニュー）が追加されます。例えば、プログラムウィンドウがアクティブな場合はプログラムウィンドウ特有のメニュー、ソースウィンドウがアクティブな場合はソースウィンドウ特有のメニューが割り当てられます。

Option メニューの下に割り当てられる各種メニューについては、各ウィンドウの操作方法の「拡張メニューの操作方法」に記載していますので、そちらを参照してください。

5.3.7. Windowメニュー

Window メニューには、PDB38Mのメイン表示領域の表示形態を変更するメニューが割り当てられています。以下では、Window メニューの下に割り当てられている各種メニューの機能について説明します。

□ Cascade メニュー

メイン表示領域に表示されたウィンドウを、重ねて表示します。

□ Tile メニュー

メイン表示領域に表示されたウィンドウを、並べて表示します。

□ **Arrange Icon メニュー**

メイン表示領域に表示されたアイコンを、整列します。

□ **Program Window メニュー**

プログラムウィンドウをアクティブにします。

プログラムウィンドウの操作方法については、ウィンドウ操作編の「プログラムウィンドウの操作方法」を参照してください。

□ **Source Window メニュー**

ソースウィンドウを、メイン表示領域内にオープンします。

ソースウィンドウの操作方法については、ウィンドウ操作編の「ソースウィンドウの操作方法」を参照してください。

□ **Register Window メニュー**

レジスタウィンドウを、メイン表示領域内にオープンします。

既にレジスタウィンドウがオープンしている場合は、レジスタウィンドウをアクティブにします。

レジスタウィンドウの操作方法については、ウィンドウ操作編の「レジスタウィンドウの操作方法」を参照してください。

□ **Memory Window メニュー**

メモリウィンドウを、メイン表示領域内にオープンします。

メモリウィンドウの操作方法については、ウィンドウ操作編の「メモリウィンドウの操作方法」を参照してください。

□ **Dump Window メニュー**

ダンプウィンドウを、メイン表示領域内にオープンします。

ダンプウィンドウの操作方法については、ウィンドウ操作編の「ダンプウィンドウの操作方法」を参照してください。

□ **ASM Watch Window メニュー**

ウォッチウィンドウを、メイン表示領域内にオープンします。既にウォッチウィンドウがオープンしている場合は、ウォッチウィンドウをアクティブにします。

ウォッチウィンドウの操作方法については、ウィンドウ操作編の「ウォッチウィンドウの操作方法」を参照してください。

□ **Script Window メニュー**

スクリプトウィンドウを、メイン表示領域内にオープンします。

既にスクリプトウィンドウがオープンしている場合は、スクリプトウィンドウをアクティブにします。

スクリプトウィンドウの操作方法については、ウィンドウ操作編の「スクリプトウィンドウの操作方法」を参照してください。

5.3.8. Helpメニュー

Helpメニューには、PDB38Mの使用法やバージョン表示などに関連するメニューが割り当てられています。以下では、Helpメニューの下に割り当てられている各種メニューの機能について説明します。

□ Indexメニュー

ヘルプウィンドウをオープンし、ヘルプファイルの目次を表示します。

ヘルプウィンドウとは、PDB38Mのヘルプファイルを参照するためのウィンドウです。

□ Aboutメニュー

PDB38Mおよびエミュレータ本体(M38000T-SBI)のモニターバージョンを表示します。

Aboutメニューを選択すると、Aboutダイアログがオープンします。図 5-28に Aboutダイアログの構成を示します。

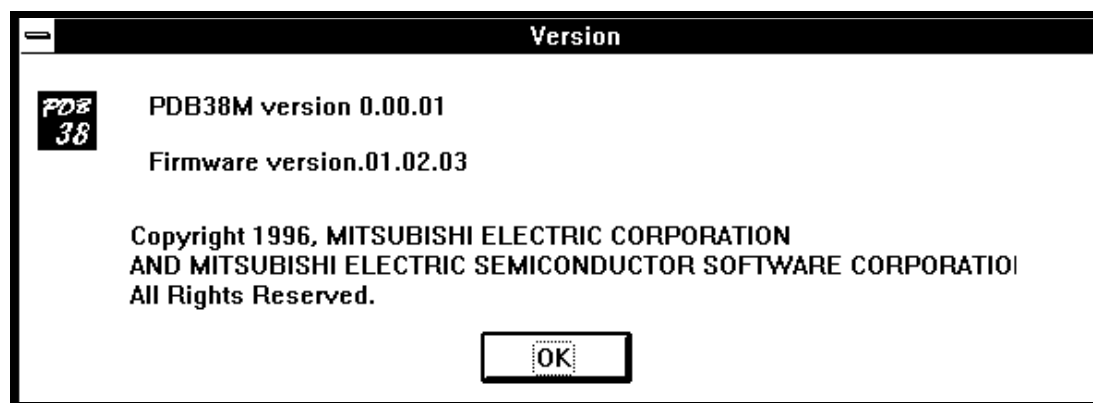


図 5-28 Aboutダイアログ

5.4. ツールバーの構成

PDB38Mウィンドウのツールバーは、図 5-29に示すボタンで構成されています。



図 5-29 PDB38Mウィンドウのツールバー構成

各ボタンの操作方法は以下の通りです。

○ Go ボタン

現在のプログラムカウンタから、ターゲットプログラムを実行します。

動作は、メニュー [Debug] [Go] [Go] を選択した場合と同じになります。

○ Come ボタン

現在のプログラムカウンタから、プログラム表示領域のカーソル位置までターゲットプログラムを実行します。

動作は、メニュー [Debug] [Come] を選択した場合と同じになります。

○ Step ボタン

1 回分、ステップ実行を行います。

動作は、メニュー [Debug] [Step] [Step] を選択した場合と同じになります。

○ Over ボタン

1 回分、オーバーステップ実行を行います。

動作は、メニュー [Debug] [Over] [Over] を選択した場合と同じになります。

○ Return ボタン

現在のサブルーチンから復帰します。

動作は、メニュー [Debug] [Return] を選択した場合と同じになります。

○ Stop ボタン

ターゲットプログラムの実行を停止します。

動作は、メニュー [Debug] [Stop] を選択した場合と同じになります。

○ Break ボタン

プログラム表示領域のカーソル位置に、ソフトウェアブレークポイントを設定/解除します。

動作は、メニュー [Debug] [Break Point] [Break] を選択した場合と同じになります。

す。

○ **Reset ボタン**

ターゲットをリセットします。

動作は、メニュー [Debug] [Reset] を選択した場合と同じになります。

○ **S/W ボタン**

S/W ブレークポイント設定ダイアログをオープンします。

動作は、メニュー [Debug] [Break Point] [S/W Break Point] を選択した場合と同じになります。

6. プログラムウィンドウの操作方法

6.1. プログラムウィンドウの概要

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表示します。プログラムウィンドウは、PDB38Mを起動した際に自動的にオープンします。カーソル位置までの実行、プログラムウィンドウ上でのソフトウェアブレークポイント設定/解除、現在のプログラムカウンタに相当するプログラムの表示などが行えます。ソフトウェアブレークポイントの設定/解除は、アドレス表示領域または行番号表示領域をマウスでダブルクリックすることによって行えます。

プログラムウィンドウの機能を以下に示します。

1. 現在のプログラムカウンタに相当するプログラムの表示
 - ソースファイルまたは逆アセンブル表示でプログラムを表示します。
 - プログラムカウンタに相当する行は、黄色のラインで表示します。
 - ソフトウェアブレークが設定されている行には、ブレークポイント表示領域に 'B' が表示されます。
 - ソースファイルの各行に対して、行番号とアドレスを表示します（デフォルトでは行番号のみの表示になります）。
 - 逆アセンブル表示の各行に対して、アドレス、オブジェクトコードを表示します。
 - 各種コマンド実行後、プログラムカウンタの移動に伴い表示を変更します。
 - 表示しているソースファイル名をプログラムウィンドウのタイトルバーに表示します。
2. ソフトウェアブレークポイントの設定/解除

6.2. プログラムウィンドウの構成

プログラムウィンドウには、ソース、逆アセンブルの2種類の表示モードがあります。

図 6-1にソース表示モード、図 6-2に逆アセンブル表示モードの場合のプログラムウィンドウの構成図を示します。

Line	BRK	Source
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER

図 6-1 ソース表示モードのプログラムウィンドウ構成図

Address	BRK	Objcode	Label	Mnemonic
C000	-	78	RESET	SEI
C001	-	58		CLI
C002	-	78		SEI
C003	-	32		SET
C004	-	F8		SED
C005	-	12		CLT
C006	-	D8		CLD
C007	-	3C04FB		LDM #04H,FBH :CPUMOD

図 6-2 逆アセンブル表示モードのプログラムウィンドウ構成図

プログラムウィンドウの各部品について、以下で説明します。

6.2.1. ツールバー

プログラムウィンドウは、表示領域の変更、表示モードの切り替えを行うボタンを備えています。

ツールバーの各ボタンの機能についての詳細は、本章の6.6節「ツールバーの構成」を参照してください。

6.2.2. プログラム表示領域

プログラムを表示する領域です。

ソース表示モードの場合は、ソースファイルの内容を表示します。逆アセンブル表示モードの場合は、逆アセンブル結果（「オブジェクトコード」「ラベル」「ニーモニック」）を表示します。現在のプログラムカウンタに相当する行は、黄色のラインで表示します。プログラムカウンタの表示例を、図 6-3に示します。

Address	BRK	Objcode	Label	Mnemonic
C000	-	78	RESET	SEI
C001	-	58		CLI
C002	-	78		SEI
C003	-	32		SET
C004	-	F8		SED
C005	-	12		CLT
C006	-	D8		CLD
C007	-	3C04FB		LDM #04H,FBH :CPUMOD

図 6-3 プログラムカウンタの表示例

6.2.3. 行番号表示領域

行番号を表示する領域です。

ソース表示モードの場合、ソースファイルの行番号を 10 進数で表示します。逆アセンブル表示モードの場合、行番号表示領域は存在しません。

また、行番号表示領域から、プログラム表示領域の表示内容の変更を簡単に行うことができます。行番号表示領域をマウスでダブルクリックすると、Source ダイアログがオープンします。Source ダイアログの操作方法については、本章の6.5節「拡張メニューの操作方法」のメニュー [Option] [View] [Source] の説明を参照してください。

6.2.4.アドレス表示領域

アドレスを表示する領域です。プログラム表示領域の各行のアドレスを 16 進数で表示します。

なお、ソース表示モードの場合は、デフォルトではアドレス表示領域を表示しません。ソース表示モードでアドレス表示領域を表示する場合は、メニュー [Option] [Layout] [Address Area] を選択してください。

また、アドレス表示領域から、プログラム表示領域の表示内容の変更を簡単に行うことができます。アドレス表示領域をマウスでダブルクリックすると、Address ダイアログがオープンします。Address ダイアログの操作方法については、本章の6.5節「拡張メニューの操作方法」のメニュー [Option] [View] [Address] の説明を参照してください。

6.2.5.ブレークポイント表示領域

ソフトウェアブレークポイントが設定されている行を表示する領域です。ソフトウェアブレークが設定されている行には 'B' が表示されます。ソフトウェアブレークが設定可能な行には '-' が表示されます。ソフトウェアブレークが設定できない行 (空白行、コメント行等) に対しては何も表示しません。ブレークポイント表示領域の表示例を図 6-4に示します。

BRK	Source
	;*****;
B	LDM #0FFH,P0D ;PORT1,1 OUTPUT
-	LDM #00H,P0 ;PORT1,1=0001H
-	LDM #0FFH,P1D ;PORT1,1 OUTPUT
-	LDM #001H,P1 ;PORT1,1=0001H
-	LDM #00H,P2D ;PORT1,1 OUTPUT
	LDM #001H,P2 ;PORT1,1=0001H

図 6-4 ブレークポイント表示領域の表示例

本領域をマウスでダブルクリックすることにより、ソフトウェアブレークポイントを設定/解除することができます。

- '-' の表示をダブルクリックすることによって、ソフトウェアブレークポイントが設定できます。その際、表示が 'B' に変わります。
- 'B' の表示をダブルクリックすることによって、ソフトウェアブレークポイントが解除できます。その際、表示が '-' に戻ります。

6.2.6.オブジェクトコード表示領域

オブジェクトコードを表示する領域です。

逆アセンブル表示モードの場合、逆アセンブル行のオブジェクトコードを 16 進数で表示します。ソース表示モードの場合、オブジェクトコード表示領域は存在しません。

6.2.7.垂直スクロールバー

全表示領域（プログラム表示領域、行番号表示領域、アドレス表示領域、ブレークポイント表示領域、オブジェクトコード表示領域）を垂直方向にスクロールします。

ソース表示モードの場合、表示しているソースファイルがスクロール範囲になります。逆アセンブル表示モードの場合は、逆方向へのスクロールは順方向へのスクロールの後でないと使用できません。逆方向へのスクロールは、順方向へのスクロールを元に戻すための機能です。順方向へのスクロールの際、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。

6.2.8.水平スクロールバー

プログラム表示領域を水平方向にスクロールします。

6.3. プログラムウィンドウの表示モードについて

プログラムウィンドウでは、ソース、逆アセンブルの2種類の表示モードをサポートしています。各表示モードについて、以下で説明します。

6.3.1. ソース表示モード

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。

図 6-5 にソース表示モードの表示例を示します。

Line	BRK	Source
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER

図 6-5 ソース表示モードでの表示

- PDB38Mでは、ソース表示モードにおいて、ソースレベルでのデバッグが行えます。
- ソース表示モードのプログラムウィンドウには、下記の表示領域があります。
 - a. 行番号表示領域
 - b. アドレス表示領域（デフォルトでは未表示）
 - c. ブレークポイント表示領域
 - d. プログラム表示領域
- プログラム表示領域にソースファイルの内容を表示します。
ソースファイルのタブのカラム数は、メニュー [Option] [TAB] で設定できます。
- 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。
- 垂直スクロールバーの操作によって、表示中のソースファイルの先頭行から最終行

までスクロールできます。このとき、他の表示領域も同時にスクロールします。

- 水平スクロールバーの操作によって、プログラム表示領域が水平方向にスクロールします。
- プログラムウィンドウがソース表示モードの時にPDB38Mウィンドウのメニュー等でステップ・オーバーステップ・リターンコマンドを発行すると、それぞれソース行単位のステップ・オーバーステップ・リターン実行を行います。

6.3.2.逆アセンブル表示モード

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。

図 6-6 に逆アセンブル表示モードの表示例を示します。

Address	BRK	Objcode	Label	Mnemonic
C000	-	78	RESET	SEI
C001	-	58		CLI
C002	-	78		SEI
C003	-	32		SET
C004	-	F8		SED
C005	-	12		CLT
C006	-	D8		CLD
C007	-	3C04FB		LDM #04H,FBH :CPUMOD

図 6-6 逆アセンブル表示モードでの表示

- PDB38M起動時の表示モードは、逆アセンブル表示モードです。
- プログラムウィンドウには、下記の表示領域があります。
 - アドレス表示領域
 - ブレークポイント表示領域
 - オブジェクトコード表示領域
 - プログラム表示領域
- オブジェクトコード表示領域（ Objcode ）にオブジェクトコードを表示します。
- プログラム表示領域に逆アセンブル結果を表示します。
表示内容は左から、ラベルを表示する Label 、ニーモニックを表示する Mnemonic になります。
- 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) の広さの割合を簡単に調節することができます。

Line	BRK	Source
00050	-	RESET: ;
00051	-	SEI ;INTERRUPT DISABLE
00052	-	CLI ;
00053	-	SEI ;INTERRUPT DISABLE
00054	-	SET ;INTERRUPT DISABLE
00055	-	SED ;INTERRUPT DISABLE
00056	-	CLT ;
00057	-	CLD ;
00058	-	LDM #04H,CPUMOD ;CPU MODE REGISTER
00059	-	LDX #0BFH ;STACK POINT SET

- 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。
- 垂直スクロールバーの操作によって、表示中のオブジェクトコードを垂直方向にスクロールできます。このとき、他の表示領域も同時にスクロールします。
 - 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の際、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。
- 水平スクロールバーの操作によって、プログラム表示領域が水平方向にスクロールします。
- 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。
- プログラムウィンドウが逆アセンブル表示モードの時にPDB38Mウィンドウのメニュー等でステップ・オーバーステップ・リターンコマンドを発行すると、それぞれ命令行単位のステップ・オーバーステップ・リターン実行を行います。

6.4.表示内容・表示モードの更新について

プログラムウィンドウでは、現在のプログラムカウンタに相当する行を表示するため、コマンド実行後に表示内容・表示モードを自動的に更新します。

以下に、この自動更新について説明します。

6.4.1.表示内容の自動更新

プログラムカウンタの変更に伴うコマンド（ターゲット実行のブレーク・ステップ実行・リセットなど）を実行した後、プログラムカウンタに相当する行がプログラム表示領域にならない場合、プログラムカウンタ行を表示するために表示内容を自動的に更新します。

- ソース表示モードでの表示更新の際は、プログラムカウンタに相当する行がプログラム表示領域の2行目に位置するように表示します。
- 逆アセンブル表示モードでの表示更新の際は、プログラムカウンタに相当する行がプログラム表示領域の1行目に位置するように表示します。

なお、ターゲットプログラムのダウンロード後は、ターゲットをリセットしますので、リセット実行と同様に表示内容が更新されます。

6.4.2.表示モードの自動更新

表示モードの自動更新には、ソース表示モードから逆アセンブル表示モードへの更新と、逆アセンブル表示モードからソース表示モードへの更新の2種類があります。それぞれ、下記の条件で自動更新を行います。

- ソース表示モードから逆アセンブル表示モードへの自動更新
ターゲット実行中にブレークした際（ストップコマンドの実行・ブレークポイント等で）、プログラムカウンタがソース行情報のない領域になった場合、表示モードを逆アセンブル表示モードに更新します。
- 逆アセンブル表示モードからソース表示モードへの自動更新
ターゲットプログラムのダウンロード後、ターゲットをリセットしますが、その際、プログラムカウンタの何れかのソース行頭に相当する場合、表示モードをソースモードに更新します。

6.4.3.手動による表示内容・表示モードの更新

プログラムウィンドウの表示内容・表示モードは、下記の操作によって手動でも変更できます。

- 表示内容の変更
 - 水平・垂直スクロールバー、ツールバーの表示領域変更ボタン、メニューによって変更できます。

- ソースファイル名の指定、アドレスの指定、文字列検索によって、変更します。

- 表示モードの変更
 - 表示モードは、メニューおよびツールバーのボタンによって切り替えられます。

6.5.拡張メニューの操作方法

プログラムウィンドウの拡張メニューには、プログラムウィンドウの表示領域の内容変更、モードの切り替えなどを行うためのメニューが備えられています。プログラムウィンドウの拡張メニューは、プログラムウィンドウがアクティブである場合にPDB38Mウィンドウの基本メニュー [Option] に追加されます。

プログラムウィンドウの拡張メニューの構成を表 6-1に示します。

表 6-1 プログラムウィンドウの拡張メニュー

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
Disasm mode	逆アセンブル表示モードへ変更		Ctrl + R
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		

以下に、プログラムウィンドウの拡張メニューの操作方法を説明します。

□ Font メニュー

表示領域のフォントを変更します。

Font メニューを選択すると、フォント指定ダイアログがオープンします。図 6-7にフォント指定ダイアログの構成図を示します。フォントを指定してください。

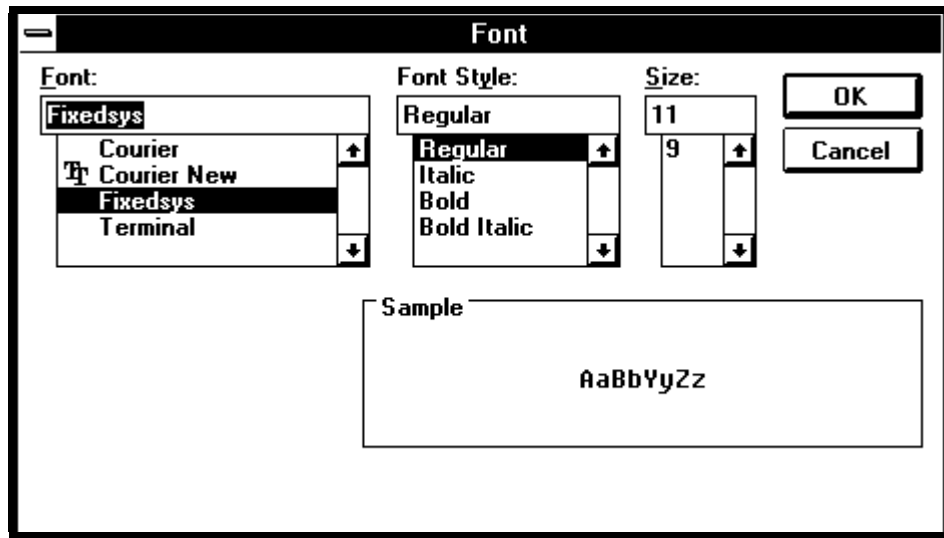


図 6-7 フォント指定ダイアログ

□ TAB メニュー

ソースファイル表示のタブ値を設定します。

TAB メニューを選択すると、TAB ダイアログがオープンします。図 6-8に TAB ダイアログの構成図を示します。タブ値を指定してください。

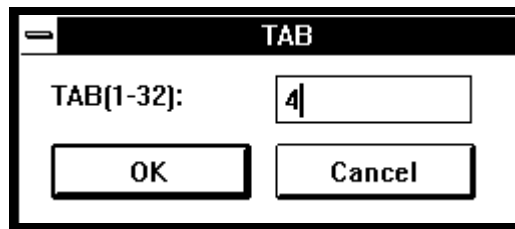


図 6-8 TAB ダイアログ

- TAB(1-32):領域に、タブ値を入力します。
- タブ値には、1 から 32 までの数値が指定できます。

□ View メニュー

表示領域の表示内容を変更するためのメニューを備えます。

View メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Source サブメニュー

プログラム表示領域に表示するソースファイル・関数を指定します。

Source サブメニューを選択すると、Source ダイアログがオープンします。図 6-9 に Source ダイアログの構成図を示します。ソースファイル名を選択してください。

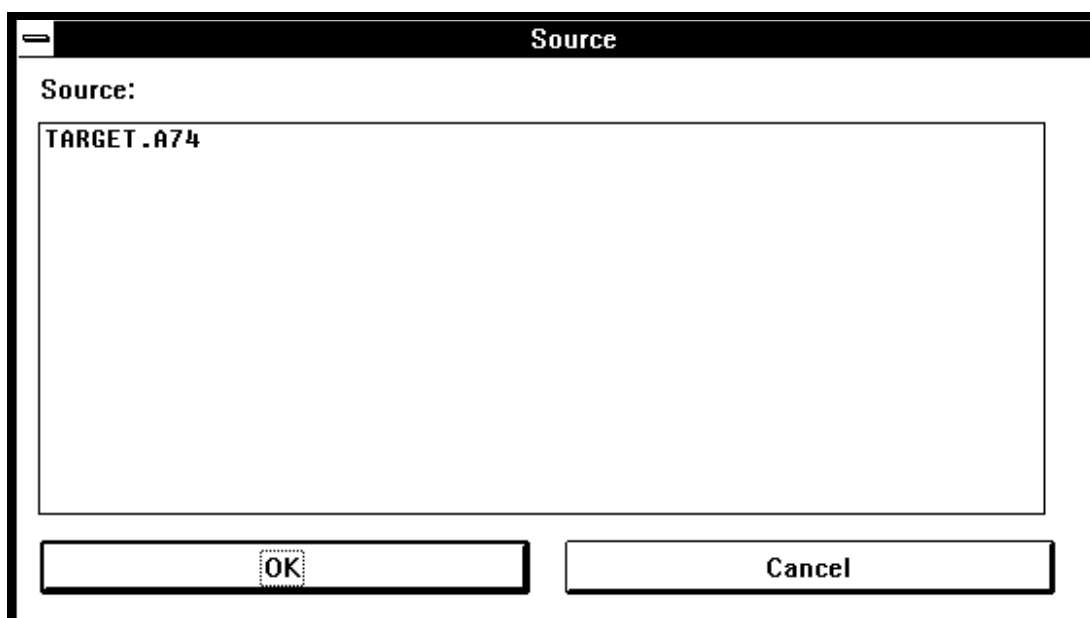


図 6-9 Source ダイアログ

- Source:領域には、ダウンロードしたロードモジュールファイルを構成するモジュール、および各モジュールに属するソースファイル（ヘッダファイルを含む）の一覧が表示されます。
- Source:領域の任意のソースファイル名をダブルクリック（またはシングルクリック + OK ボタン）すると、指定したソースファイルをプログラム表示領域に表示します。
- なお、本機能は、表示モードがソースモードの場合にのみ利用できます。

Address サブメニュー

プログラム表示領域に表示するプログラムのアドレスまたは行番号を指定します。

ソースモードにおいて Address サブメニューを選択すると、ソースモード用の Address ダイアログがオープンします。図 6-10にソースモード用の Address ダイアログの構成図を示します。アドレスを指定してください。

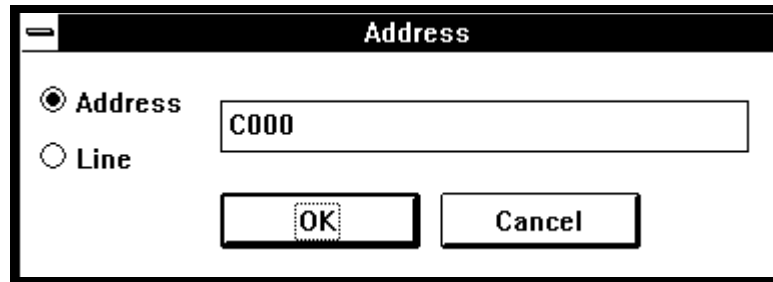


図 6-10 Address ダイアログ (ソースモードの場合)

- アドレス指定する場合は Address ラジオボタンをクリックし、右横の領域にアドレスを入力します。
 - 行番号指定する場合は Line ラジオボタンをクリックし、右横の領域に行番号を入力します。なお、入力した行番号は、現在プログラム表示領域に表示中のソースファイルの行番号として認識されます。
 - アドレス・行番号の指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
- なお、この時の定数の基数は、アドレスは 16 進数、行番号は 10 進数です。

また逆アセンブルモードにおいて Address サブメニューを選択すると、逆アセンブルモード用の Address ダイアログがオープンします。図 6-11に逆アセンブルモード用の Address ダイアログの構成図を示します。アドレスを指定してください。

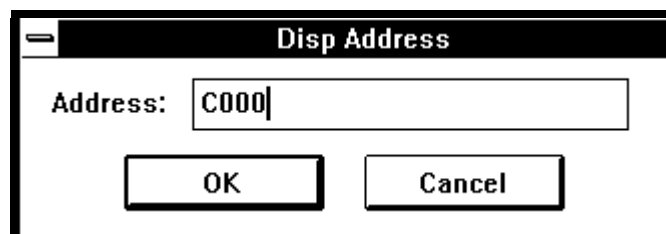


図 6-11 Address ダイアログ (逆アセンブルモードの場合)

- Address:領域に、アドレスを入力します。
- アドレスの指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
なお、この時の定数の基数は、16進数です。

Program Counter サブメニュー

プログラム表示領域の表示を、現在のプログラムカウンタの位置からの表示に変更します。

ソース表示モードの場合は、プログラムカウンタ相当行の1行手前からの表示になります。

□ Mode メニュー

表示領域の表示モードを変更するためのメニューを備えます。

Mode メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Source mode サブメニュー

プログラム表示領域の表示モードを、ソース表示モードに変更します。

Source mode サブメニューを選択すると、表示開始アドレスにソース行情報がある場合、ソースファイルの内容を表示します。ただし、ソース行情報がない場合は、ソース表示モードにはなりません。

Disasm mode サブメニュー

プログラム表示領域の表示モードを、逆アセンブル表示モードに変更します。

Disasm mode サブメニューを選択すると、ソース行情報の有無に関らず、常に逆アセンブル結果で表示します。

□ Layout メニュー

プログラムウィンドウのレイアウトを設定します。

Layout メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Line Area サブメニュー

行番号表示領域の表示/非表示を指定します。

Line Area サブメニューを選択すると、行番号表示領域を表示します。選択を解除

すると、行番号表示領域は表示しません。

なお本機能は、ソース表示モードの時のみ使用できます。

Address Area サブメニュー

アドレス表示領域の表示/非表示を指定します。

Address Area サブメニューを選択すると、アドレス表示領域を表示します。選択を解除すると、アドレス表示領域は表示しません。

Code Area サブメニュー

オブジェクトコード表示領域の表示/非表示を指定します。

Code Area サブメニューを選択すると、オブジェクト表示領域を表示します。選択を解除すると、オブジェクトコード表示領域は表示しません。

なお本機能は、逆アセンブル表示モードの時のみ使用できます。

6.6. ツールバーの構成

プログラムウィンドウのツールバーは、図 6-12に示すボタンで構成されています。

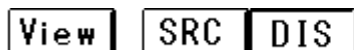


図 6-12 プログラムウィンドウのツールバーの構成

各ボタンの操作方法は以下の通りです。

○ View ボタン

プログラム表示領域の表示内容を変更するためのボタンです。

View ボタンをクリックすると、Disp Area ダイアログがオープンします。図 6-13 に、Disp Area ダイアログの構成図を示します。

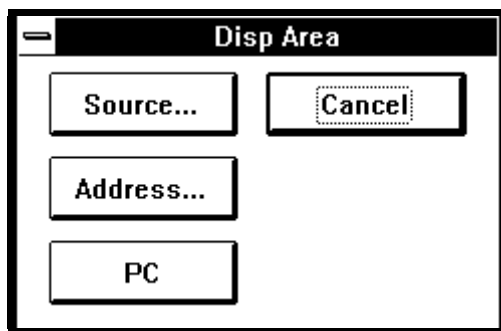


図 6-13 Disp Area ダイアログ

- Source ボタンを選択
指定したソースファイルの指定行からの表示に変更します。
Source ボタンをクリックした後の操作手順はメニュー [Option] [View] [Source] を選択した場合と同じになります。
- Address ボタンを選択
指定したアドレスからの表示に変更します。
Address ボタンをクリックした後の操作手順はメニュー [Option] [View] [Address] を選択した場合と同じになります。
- PC ボタンを選択
現在のプログラムカウンタの値からの表示に変更します。
PC ボタンをクリックした後の操作手順はメニュー [Option] [View] [Program Counter] を選択した場合と同じになります。

○ SRC ボタン

表示モードをソース表示モードに変更します。表示開始アドレスにソース行情報がある場合、ソースファイルの内容を表示します。ただし、ソース行情報がない場合は、ソース表示モードにはなりません。

動作は、メニュー [Option] [Mode] [Source mode] を選択した場合と同じになります。

○ DIS ボタン

表示モードを逆アセンブル表示モードに変更します。ソース行情報の有無に関らず、常に逆アセンブル結果で表示します。

動作は、メニュー [Option] [Mode] [Disasm mode] を選択した場合と同じになります。

7. ソースウィンドウの操作方法

7.1. ソースウィンドウの概要

ソースウィンドウは、特定のプログラム領域を継続して参照する場合に使用するウィンドウです。ソースウィンドウの表示は、プログラムウィンドウに似ています。異なる点は、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自動的に更新するのに対し、ソースウィンドウはユーザーが操作しない限り表示内容は更新しません。したがって、特定のサブルーチンやタスクなどを継続して参照できます。ソースウィンドウは、計 10 枚までオープンすることができます。ソースウィンドウの操作方法是、プログラムウィンドウと同等です。

操作方法については、「プログラムウィンドウの操作方法」を参照してください。

8. レジスタウィンドウの操作方法

8.1. レジスタウィンドウの概要

レジスタウィンドウは、レジスタ内容、フラグ内容を表示するウィンドウです。またレジスタ内容・フラグ内容を設定できます。

8.2. レジスタウィンドウの構成

図 8-1 にレジスタウィンドウの構成図を示します。

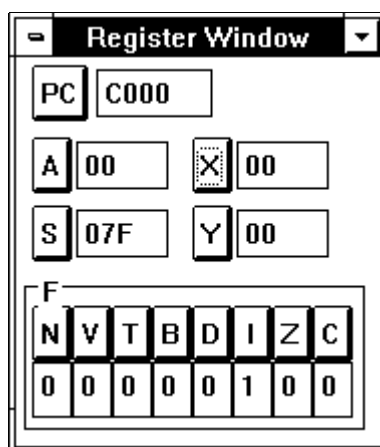


図 8-1 レジスタウィンドウの構成図

レジスタウィンドウの各部品について、以下で説明します。

8.2.1. レジスタ領域

レジスタやフラグの値を表示する領域です。各レジスタ（またはフラグ）名を示すボタンと、その値を表示する領域から構成されています。

PDB38Mのレジスタウィンドウには、表 8-1に示すレジスタ（またはフラグ）のボタンおよび表示領域が用意されています。

表 8-1 レジスタの種類

レジスタの種類	用意されているボタン
プログラムカウンタ	PC
アキュムレータ	A
インデックスレジスタ X、Y	X、Y
スタックポインタ	S
プロセッサステータスレジスタ	PS (プロセッサ割り込み優先レベル) N、V、T、B、D、I、Z、C (各フラグ)

値の表示形式は、プロセッサステータスレジスタのフラグ (N、V、T、B、D、I、Z、C) の値は 0 または 1 の数字表示、それ以外のレジスタの値は 16 進表示になります。

また、各レジスタやフラグの値を変更することもできます。変更方法は本章の 8.3 節「レジスタ値の設定方法」を参照してください。

8.3. レジスタ値の設定方法

8.3.1. プロセッサステータスレジスタの設定

フラグ (N、V、T、B、D、I、Z、C) のボタンをクリックすることにより、0 と 1 が交互に切り替わります。

8.3.2. その他のレジスタの設定

レジスタ (PC、A、X、Y、S、PC) および IPL のボタンをクリックすると、図 8-2 の Set Register ダイアログがオープンします。

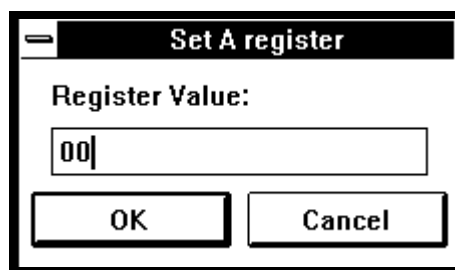


図 8-2 Set Register ダイアログ (その 2)

- Register Value:領域に、レジスタ値を入力してください。
- レジスタ値の入力には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なお、このときの定数の基数は 16 進数です。

8.4. レジスタウィンドウの表示内容の更新について

レジスタウィンドウでは、以下の条件でメモリ内容の表示を自動的に更新します。

- レジスタ内容の変更を伴うコマンドを実行した場合
レジスタ内容の変更を伴うコマンド (実行停止、ステップ実行など) を実行した際、レジスタ内容の表示を自動的に更新します。したがって、ステップ実行などのコマンド実行に伴うレジスタ内容の変化を確認することができます。

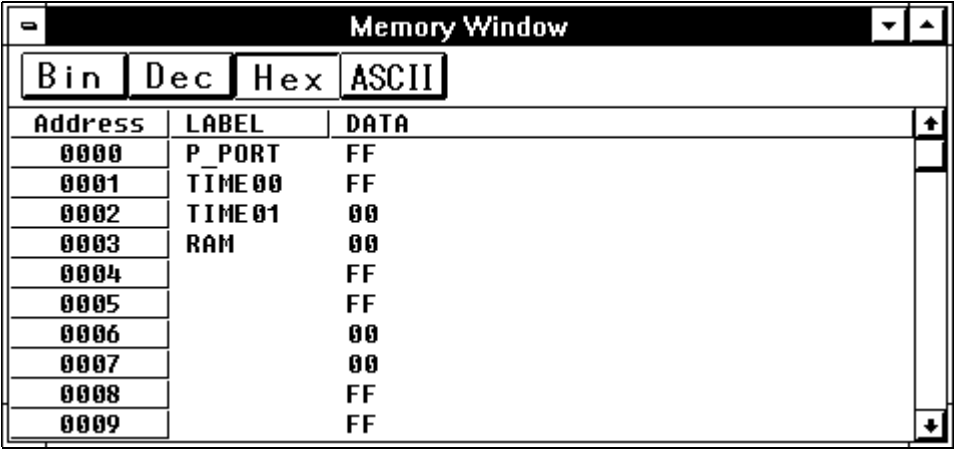
9. メモリウィンドウの操作方法

9.1. メモリウィンドウの概要

メモリウィンドウは、メモリ内容を「アドレス」「ラベル」「データ(メモリ内容)」の形式で表示するウィンドウです。また、メモリ内容の変更が可能です。メモリウィンドウは、10枚までオープンできます。

9.2. メモリウィンドウの構成

図 9-1 にメモリウィンドウの構成図を示します。



The screenshot shows a window titled "Memory Window" with a toolbar at the top containing buttons for "Bin", "Dec", "Hex", and "ASCII". Below the toolbar is a table with three columns: "Address", "LABEL", and "DATA". The table contains the following data:

Address	LABEL	DATA
0000	P_PORT	FF
0001	TIME00	FF
0002	TIME01	00
0003	RAM	00
0004		FF
0005		FF
0006		00
0007		00
0008		FF
0009		FF

図 9-1 メモリウィンドウの構成図

メモリウィンドウの各部品について、以下で説明します。

9.2.1. ツールバー

メモリ内容の変更、表示内容の変更を行うためのボタンを備えています。

ツールバーの各ボタンの機能についての詳細は、本章の9.4節「ツールバーの構成」を参照してください。

9.2.2.メモリ表示領域

メモリ内容を表示します。

- 表示内容は左から、ラベルを表示する LABEL、メモリ内容を表示する DATA になります。
- 2つの領域 (LABEL と DATA)の広さの割合を簡単に調節することができます。

Address	LABEL	DATA
0000	P_PORT	FF
0001	TIME00	FF
0002	TIME01	00
0003	RAM	00
0004		CC

- メモリ内容は、16進数、10進数、2進数またはASCII文字で表示します。
- 1行に1つのデータを表示します。
- データ長は、1バイト長、2バイト長の何れかが選択できます。
- メモリ表示領域から、メモリ内容の変更を簡単に行うことができます。メモリ表示領域に表示されている任意のデータをマウスでダブルクリックすると、Set ダイアログがオープンします。この場合は、データ値を入力するだけで、ダブルクリックした位置 (アドレス) のデータ内容が変更できます。Set ダイアログの操作方法については、本章の9.3節「拡張メニューの操作方法」のメニュー [Option] [Debug] [Set] の説明を参照してください。

9.2.3.アドレス表示領域

メモリ表示領域の各データのアドレスを表示します。

- アドレス表示領域は、メモリ表示領域の左側に存在します。垂直スクロールバーの操作によって、メモリ表示領域と共にスクロールします。
- アドレス表示領域から、メモリ表示の開始アドレスを簡単に変更することができます。アドレス表示領域をマウスでダブルクリックすると、Address ダイアログがオープンします。Address ダイアログの操作方法については、本章の9.3節「拡張メニューの操作方法」のメニュー [Option] [View] [Address] の説明を参照してください。

9.3.拡張メニューの操作方法

メモリウィンドウの拡張メニューには、表示領域の内容変更、メモリデータの入力などを行うためのメニューが備えられています。メモリウィンドウの拡張メニューは、メモリウィンドウがアクティブである場合にPDB38Mウィンドウの基本メニュー [Option] に追加されます。

メモリウィンドウの拡張メニューの構成を表 9-1に示します。

表 9-1 メモリウィンドウの拡張メニュー

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	
	<u>S</u>	表示開始アドレスを S レジスタ値に変更	
	Data Length	表示データ長の指定	
	<u>B</u> yte	1 バイト単位で表示	
	<u>W</u> ord	2 バイト単位で表示	
	Radix	表示基数の指定	
	<u>B</u> in	2 進数で表示	
	<u>D</u> ec	10 進数で表示	
	<u>H</u> ex	16 進数で表示	
	<u>A</u> scii	ASCII 文字で表示	
	Refresh	データの再表示	
Debug	メモリ内容の設定		
<u>S</u> et...	指定アドレスにデータを設定		
<u>F</u> ill...	指定したメモリブロックにデータ充填		

以下に、メモリウィンドウの拡張メニューの操作方法を説明します。

□ Font メニュー

表示領域のフォントを変更します。

Font メニューを選択すると、フォント指定ダイアログがオープンします。図 9-2にフォント指定ダイアログの構成図を示します。フォントを指定してください。

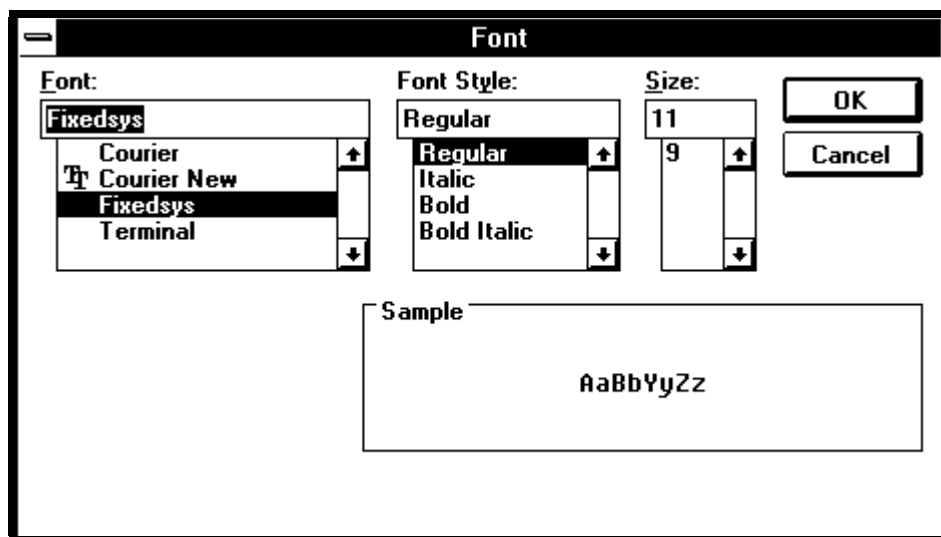


図 9-2 フォント指定ダイアログ

□ View メニュー

表示領域の表示内容を変更します。

View メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Scroll Area サブメニュー

メモリ表示領域のスクロール範囲を指定します。

Scroll Area サブメニューを選択すると、Scroll Area ダイアログがオープンします。

図 9-3に Scroll Area ダイアログの構成図を示します。開始アドレス、終了アドレスを入力してください。

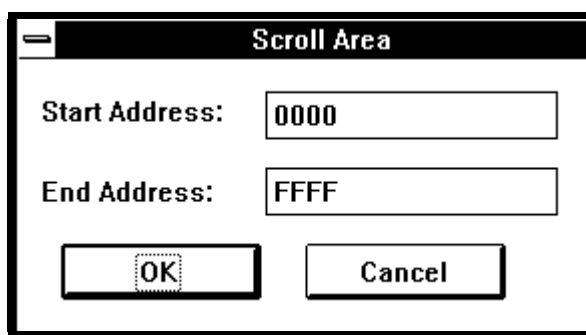


図 9-3 Scroll Area ダイアログ

- Start Address:領域に開始アドレス、End Address:領域に終了アドレスを入力してください。指定範囲のスクロールが可能になります。
- スクロール範囲の開始アドレス、終了アドレスの指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。なお、この時の定数の基数は、16進数です。
- スクロール範囲を指定した結果、現在の表示アドレスがスクロール範囲外になる場合、指定したスクロール範囲の先頭アドレスからの表示に変更します。

Address サブメニュー

メモリ表示の開始アドレスを指定します。開始アドレスを指定することによって、メモリ表示領域の表示を指定したアドレスからの表示に変更します。

Address サブメニューを選択すると、Address ダイアログがオープンします。図 9-4 に Address ダイアログの構成図を示します。アドレスを入力してください。

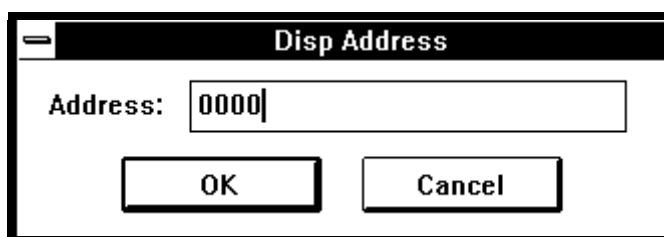


図 9-4 Address ダイアログ

- Address:領域に、アドレスを入力してください。指定したアドレスから、メモリ内容を表示します。
- アドレスには式が記述できます。式の仕様は、スクリプト仕様編の「式の記述

方法」に記載しています。

なお、この時の定数の基数は、16 進数です。

- スクロール範囲外のアドレスを指定した場合は、エラーになります。

S サブメニュー

S レジスタの内容を、表示開始アドレスに設定します。

Data Length サブメニュー

表示データのデータ長を指定します。

Data Length サブメニューを選択すると、さらに以下のサブサブメニューが表示されます。目的に応じて選択してください。

Byteサブサブメニュー

1 バイト単位でメモリ内容を表示します。

Wordサブサブメニュー

2 バイト単位でメモリ内容を表示します。

Radix サブメニュー

表示データの基数を指定します。

Radix サブメニューを選択すると、さらに以下のサブサブメニューが表示されます。目的に応じて選択してください。

Bin サブサブメニュー

2 進数でメモリ内容を表示します。

Dec サブサブメニュー

10 進数でメモリ内容を表示します。

Hex サブサブメニュー

16 進数でメモリ内容を表示します。

Ascii サブサブメニュー

ASCII 文字でメモリ内容を表示します。

Refresh サブメニュー

データ表示を更新します。

メモリウィンドウでは、通常、メモリ変更を伴うコマンド（メモリの設定・充填、実行停止、ステップ実行等）を実行した際に、表示を自動的に更新します。Refresh サブメニューは、これらのタイミング以外でデータ表示を更新したい場合に指定してください。

□ Debug メニュー

メモリ内容の設定を行います。

Debug メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Set サブメニュー

指定したアドレスに指定したデータを設定します。

Set サブメニューを選択すると、Set ダイアログがオープンします。図 9-5に Set ダイアログの構成図を示します。アドレス、データ、データ長を指定してください。

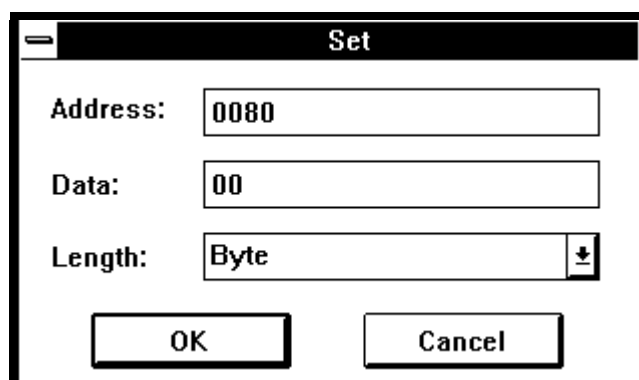


図 9-5 Set ダイアログ

- Address:領域にアドレス、Data:領域にデータ、Length:領域にデータ長を指定してください。
指定したアドレスに指定データを書き込みます。
- アドレスおよびデータには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なお、この時の定数の基数は、アドレスの場合は 16 進数になります。データの場合は、メモリウィンドウに表示している基数が 10 進の時は 10 進数、それ以外の時は 16 進数入力になります。
- データ長は、Byte (1 バイト長)、Word (2 バイト長) の何れかが選択できます。Length:領域横のドロップダウンリスト (下向きの矢印をマウスでクリックすると表示) により選択してください。なお、デフォルトは、メモリウィンドウに表示しているデータ長になります。

Fill サブメニュー

メモリを充填します。

Fill サブメニューを選択すると、Fill ダイアログがオープンします。図 9-6に Fill ダイアログの構成図を示します。開始アドレス、終了アドレス、データ、データ長を指定してください。

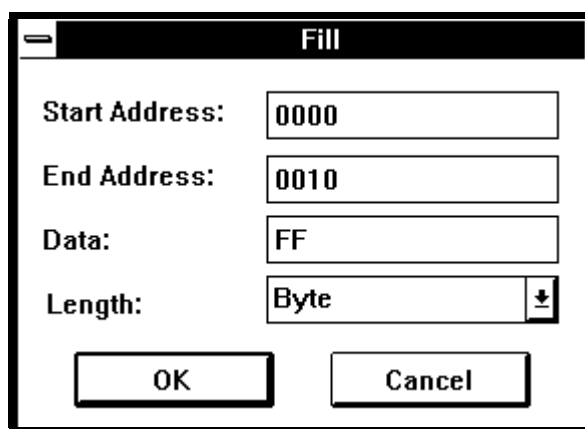


図 9-6 Fill ダイアログ

- Start Address:領域に開始アドレス、End Address:領域に終了アドレス、Data:領域にデータ、Length:領域にデータ長を指定してください。
開始アドレスから終了アドレスの範囲に、指定したデータを充填します。
- 開始アドレス、終了アドレス、データには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なお、この時の定数の基数は、開始アドレスおよび終了アドレスの場合は 16 進数になります。データの場合は、メモリウィンドウに表示している基数が 10 進の時は 10 進数、それ以外の時は 16 進数入力になります。
- データ長は、Byte (1 バイト長)、Word (2 バイト長) の何れかが選択できます。Length:領域横のドロップダウンリスト (下向きの矢印をマウスでクリックすると表示) により選択してください。
なお、デフォルトは、メモリウィンドウに表示しているデータ長になります。

9.4. ツールバーの構成

メモリウィンドウのツールバーは、図 12-9に示すボタンで構成されています。



図 9-7 メモリウィンドウのツールバーの構成

各ボタンの操作方法は以下の通りです。

○ Bin ボタン

2進数でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Bin] を選択した場合と同じになります。

○ Dec ボタン

10進数でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Dec] を選択した場合と同じになります。

○ Hex ボタン

16進数でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Hex] を選択した場合と同じになります。

○ ASCII ボタン

ASCII 文字でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Ascii] を選択した場合と同じになります。

9.5.メモリウィンドウの表示内容の更新について

メモリウィンドウでは、以下の条件でメモリ内容の表示を自動的に更新します。

- メモリ内容の変更を伴うコマンドを実行した場合、メモリ内容の変更を伴うコマンド（メモリの設定・充填、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。したがって、ステップ実行などのコマンド実行に伴うメモリ内容の変化を確認することができます。

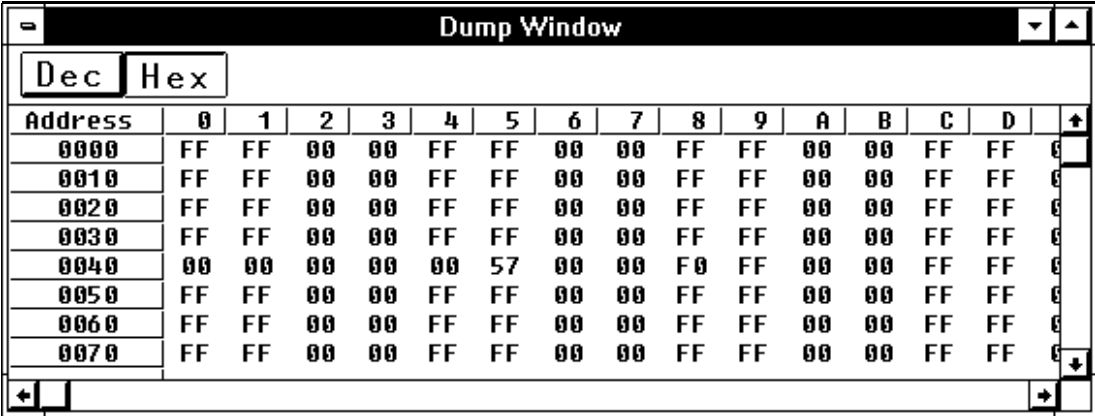
10. ダンプウィンドウの操作方法

10.1. ダンプウィンドウの概要

ダンプウィンドウは、メモリ内容をダンプ形式で表示するウィンドウです。また、メモリ内容の変更が可能です。ダンプウィンドウは、10枚までオープンできます。

10.2. ダンプウィンドウの構成

図 10-1にダンプウィンドウの構成図を示します。



The screenshot shows a window titled "Dump Window" with a tabbed interface. The "Hex" tab is selected. The table displays memory addresses from 0000 to 0070 in increments of 10. Each address is followed by 16 hexadecimal bytes, grouped into four columns of four bytes each. The values are mostly FF, with some 00 and one 57 at address 0040.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	
0000	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0010	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0020	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0030	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0040	00	00	00	00	00	57	00	00	F0	FF	00	00	FF	FF	0
0050	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0060	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0
0070	FF	FF	00	00	FF	FF	00	00	FF	FF	00	00	FF	FF	0

図 10-1 ダンプウィンドウの構成図

ダンプウィンドウの各部品について、以下で説明します。

10.2.1. ツールバー

メモリ内容の変更、表示内容の変更を行うためのボタンを備えています。ツールバーの各ボタンの機能についての詳細は、本章の10.4節「ツールバーの構成」を参照してください。

10.2.2. メモリ表示領域

メモリ内容を、ダンプ形式で表示します。

- メモリ内容は、16進数または10進数で表示します。
- 1行に16バイト分のメモリ内容を表示します。
- データ長は、1バイト長、2バイト長の何れかが選択できます。
- メモリ表示領域の上部には、行頭アドレスからのオフセットを示す値を表示します。水平スクロールバーの操作によって、メモリ表示領域と共にスクロールします。
- メモリ内容の右横に、各行のメモリ内容に対応したASCIIコード列(16バイト分)を表示します。
- メモリ表示領域から、メモリ内容の変更を簡単に行うことができます。メモリ表示領域に表示されている任意のデータをマウスでダブルクリックすると、Setダイアログがオープンします。この場合は、データ値を入力するだけで、ダブルクリックした位置(アドレス)のデータ内容が変更できます。Setダイアログについては、本章の10.3節「拡張メニューの操作方法」のメニュー[Option][Debug][Set]の説明を参照してください。

10.2.3. アドレス表示領域

メモリ表示領域の各行の先頭アドレスを表示します。

- アドレス表示領域は、メモリ表示領域の左側に存在します。垂直スクロールバーの操作によって、メモリ表示領域と共にスクロールします。
- メモリ表示領域の各行の先頭アドレスは、16バイト境界(下1桁が0のアドレス)になります。
- アドレス表示領域から、メモリ表示の開始アドレスを簡単に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と、Addressダイアログがオープンします。Addressダイアログの操作方法については、本章の10.3節「拡張メニューの操作方法」のメニュー[Option][View][Address]の説明を参照してください。

10.3.拡張メニューの操作方法

ダンプウィンドウの拡張メニューには、表示領域の内容変更、メモリデータの入力などを行うためのメニューが備えられています。ダンプウィンドウの拡張メニューは、ダンプウィンドウがアクティブである場合にPDB38Mウィンドウの基本メニュー [Option] に追加されます。

ダンプウィンドウの拡張メニューの構成を表 10-1に示します。

表 10-1 ダンプウィンドウの拡張メニュー

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	
	Word	2 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		

以下に、ダンプウィンドウの拡張メニューの操作方法を説明します。

□ Font メニュー

表示領域のフォントを変更します。

Font メニューを選択すると、フォント指定ダイアログがオープンします。図 10-2にフォント指定ダイアログの構成図を示します。フォントを指定してください。

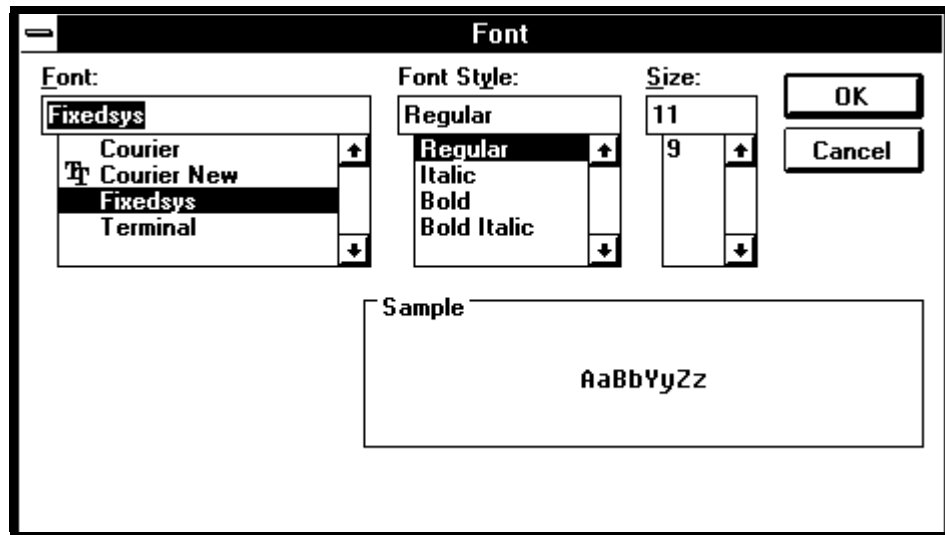


図 10-2 フォント指定ダイアログ

□ View メニュー

表示領域の表示内容を変更します。

View メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Scroll Area サブメニュー

メモリ表示領域のスクロール範囲を指定します。

Scroll Area サブメニューを選択すると、Scroll Area ダイアログがオープンします。

図 10-3に Scroll Area ダイアログの構成図を示します。開始アドレス、終了アドレスを入力してください。

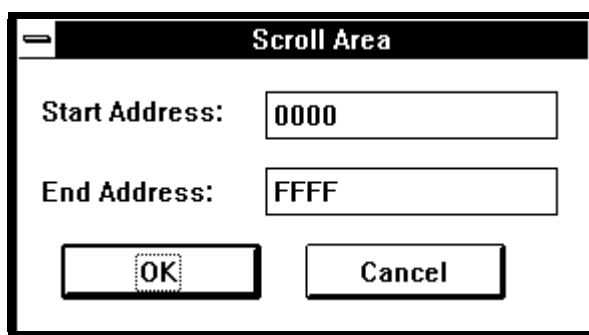


図 10-3 Scroll Area ダイアログ

- Start Address:領域に開始アドレス、End Address:領域に終了アドレスを入力してください。指定範囲のスクロールが可能になります。
- スクロール範囲の開始アドレス、終了アドレスの指定には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。なお、この時の定数の基数は、16進数です。
- スクロール範囲を指定した結果、現在の表示アドレスがスクロール範囲外になる場合、指定したスクロール範囲の先頭アドレスからの表示に変更します。

Address サブメニュー

メモリ表示の開始アドレスを指定します。開始アドレスを指定することによって、メモリ表示領域の表示を指定したアドレスからの表示に変更します。

Address サブメニューを選択すると、Address ダイアログがオープンします。図 10-4に Address ダイアログの構成図を示します。アドレスを入力してください。

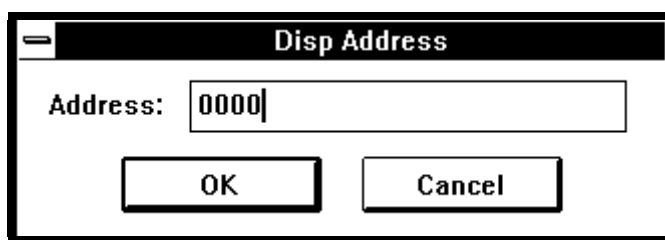


図 10-4 Address ダイアログ

- Address:領域に、アドレスを入力してください。指定したアドレスから、メモリ内容を表示します。
- ただし、指定したアドレスが16バイト境界でない場合(下位1桁が 0_{16} 以外の

場合)は、下位 1 桁の値を切り捨てた値を指定アドレスとして処理します。

- アドレスには式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。なお、この時の定数の基数は、16 進数です。
- スクロール範囲外のアドレスを指定した場合は、エラーになります。

Data Length サブメニュー

表示データのデータ長を指定します。

Data Length サブメニューを選択すると、さらに以下のサブサブメニューが表示されます。目的に応じて選択してください。

Byteサブサブメニュー

1 バイト単位でメモリ内容を表示します。

Wordサブサブメニュー

2 バイト単位でメモリ内容を表示します。この場合、1 行には 8 個のデータを表示します。

Radix サブメニュー

表示データの基数を指定します。

Radix サブメニューを選択すると、さらに以下のサブサブメニューが表示されます。目的に応じて選択してください。

Dec サブサブメニュー

10 進数でメモリ内容を表示します。

Hex サブサブメニュー

16 進数でメモリ内容を表示します。

Refresh サブメニュー

データ表示を更新します。

ダンプウィンドウでは、通常、メモリ変更を伴うコマンド(メモリの設定・充填、実行停止、ステップ実行等)を実行した際に、表示を自動的に更新します。Refresh サブメニューは、これらのタイミング以外でデータ表示を更新したい場合に指定してください。

□ Debug メニュー

メモリ内容の設定を行います。

Debug メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Set サブメニュー

指定したアドレスに指定したデータを設定します。

Set サブメニューを選択すると、Set ダイアログがオープンします。図 10-5に、Set ダイアログの構成図を示します。アドレス、データ、データ長を指定してください。

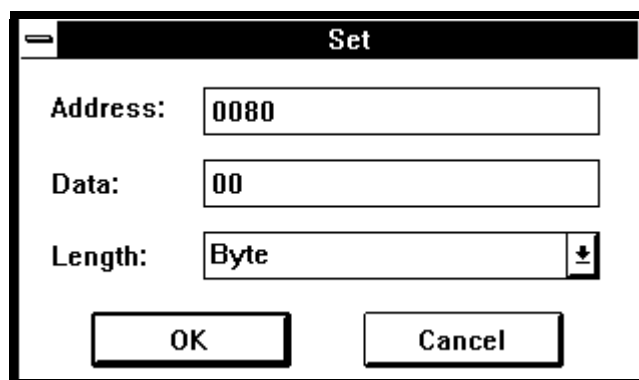


図 10-5 Set ダイアログ

- Address:領域にアドレス、Data:領域にデータ、Length:領域にデータ長を指定してください。
指定したアドレスに指定データを書き込みます。
- アドレスおよびデータには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なお、この時の定数の基数は、アドレスの場合は16進数、データの場合はダンプウィンドウに表示している基数になります。
- データ長は、Byte (1バイト長)、Word (2バイト長)の何れかが選択できます。Length:領域横のドロップダウンリスト (下向きの矢印をマウスでクリックすると表示) により選択してください。
なお、デフォルトは、ダンプウィンドウに表示しているデータ長になります。

Fill サブメニュー

メモリを充填します。

Fill サブメニューを選択すると、Fill ダイアログがオープンします。図 10-6に Fill ダイアログの構成図を示します。開始アドレス、終了アドレス、データ、データ長を指定してください。

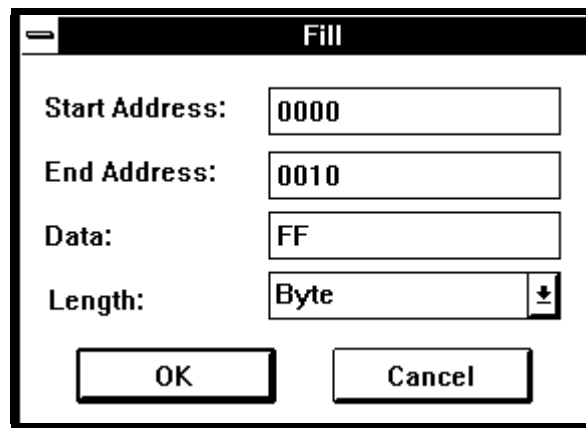


図 10-6 Fill ダイアログ

- Start Address:領域に開始アドレス、End Address:領域に終了アドレス、Data:領域にデータ、Length:領域にデータ長を指定してください。
開始アドレスから終了アドレスの範囲に、指定したデータを充填します。
- 開始アドレス、終了アドレス、データには、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なおこの時の定数の基数は、開始アドレスおよび終了アドレスの場合は 16 進数、データの場合はダンプウィンドウに表示している基数になります。
- データ長は、Byte (1 バイト長)、Word (2 バイト長) の何れかが選択できます。Length:領域横のドロップダウンリスト (下向きの矢印をマウスでクリックすると表示) により選択してください。
なお、デフォルトは、ダンプウィンドウに表示しているデータ長になります。

10.4. ツールバーの構成

ダンプウィンドウのツールバーは、図 12-9に示すボタンで構成されています。



図 10-7 ダンプウィンドウのツールバーの構成

各ボタンの操作方法は以下の通りです。

○ Dec ボタン

10 進数でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Dec] を選択した場合と同じになります。

○ Hex ボタン

16 進数でメモリ内容を表示します。

動作はメニュー [Option] [View] [Radix] [Hex] を選択した場合と同じになります。

10.5. ダンプウィンドウの表示内容の更新について

ダンプウィンドウでは、以下の条件でメモリ内容の表示を自動的に更新します。

- メモリ内容の変更を伴うコマンドを実行した場合
メモリ内容の変更を伴うコマンド（メモリの設定・充填、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。したがって、ステップ実行などのコマンド実行に伴うメモリ内容の変化を確認することができます。

11. ウォッチウィンドウの操作方法

11.1.ウォッチウィンドウの概要

ウォッチウィンドウは、アドレス式とそれぞれに対応するメモリ内容を表示するウィンドウです。指定したアドレス式とそれに対応するメモリ内容を並べて表示します。このウォッチウィンドウの表示対象となるメモリのことをウォッチポイントといいます。ウォッチポイントには、以下のものが使用できます。

- アドレス式（シンボル含む）
指定したアドレスのメモリ内容を表示します。
- ビットシンボル
ビットシンボルは、アドレスとビット番号情報を持ち、任意の1ビットを指します。ウォッチポイントがビットシンボルの場合、指定したビットの値を 0/1 で表示します。
- アドレス式とビット番号
指定したビットの値を 0/1 で表示します。

ウォッチポイントのメモリ内容の表示は、各コマンドの実行後に更新しますので、ステップ実行などのコマンド実行に伴うメモリ内容の変化を確認することができます。

11.2.ウォッチウィンドウの構成

図 11-1にウォッチウィンドウの構成図を示します。

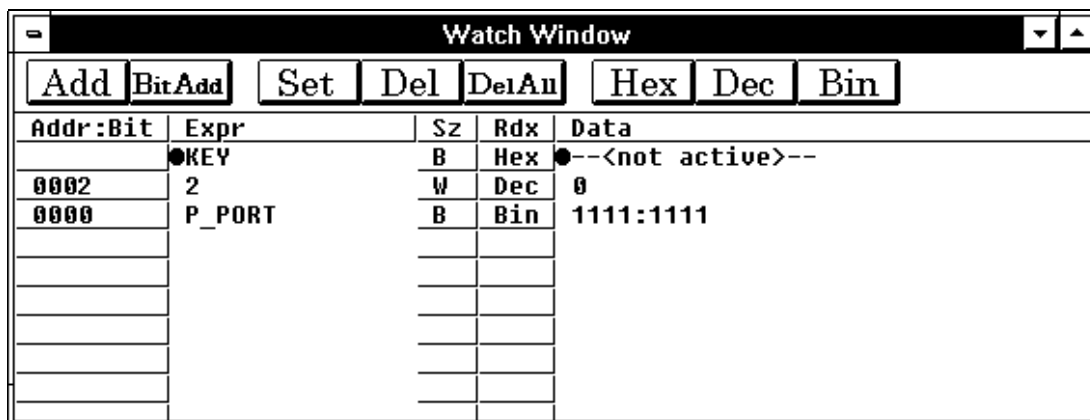


図 11-1 ウォッチウィンドウの構成図

ウォッチウィンドウの各部品について、以下で説明します。

11.2.1.ツールバー

ウォッチポイントの登録/削除などを行うためのボタンを備えています。

ツールバーの各ボタンの機能についての詳細は、「ツールバーの構成」を参照してください。

11.2.2.アドレス/ビット番号表示領域

ウォッチポイントのアドレスを 16 進数で表示します。

ウォッチポイントがビット(あるメモリの 1 ビット)の場合、アドレスの後にコロン(‘:’)とビット番号(0~7)を表示します。

11.2.3.アドレス式表示領域

ウォッチポイントのアドレス式を表示します。

- アドレス式表示領域の広さの割合を簡単に調節することができます。

Expr	Sz	Rdx	Data
KEY	B	Hex	<not active>
2	W	Dec	0
P_PORT	B	Bin	1111:1111

- 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、どちらかの領域をマウスでシングルクリックするか、または キーで移動できます。図 11-2に、カーソル表示例を示します。

Expr	Sz	Rdx	Data
KEY	B	Hex	--<not active>--
2	W	Dec	0
● P_PORT	B	Bin	● 1111:1111

図 11-2 カーソル表示例

- アドレス/ビット番号表示領域は、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 (デフォルトでは表示)。

11.2.4.データ長表示領域

ウォッチポイントのデータ長を表示します。

‘ B ’ は Byte (1 バイト長)、‘ W ’ は Word (2 バイト長) を示します。

- データ長表示領域は、メニュー [Option] [Layout] [Size 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 (デフォルトでは表示)。

11.2.5.基数表示領域

ウォッチポイントのデータ表示の基数を表示します。

‘ Hex ’ は 16 進数、‘ Dec ’ は 10 進数、‘ Bin ’ は 2 進数を示します。

- 基数表示領域から、データ表示の基数を簡単に変更することができます。基数表示領域をマウスで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していきます。

11.2.6.データ表示領域

ウォッチポイントのメモリ内容を表示します。

基数が ‘ Bin ’ (2 進数) の場合は、4 ビット毎にコロン ‘ : ’ で区切り、8 ビット毎に 1 文字の空白文字で区切ります。

- 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、どちらかの領域をマウスでシングルクリックするか

キーで移動できます。図 11-3に、カーソル表示例を示します。

Expr	Sz	Rdx	Data
KEY	B	Hex	--<not active>--
2	W	Dec	0
●P_PORT	B	Bin	●1111:1111

図 11-3 カーソル表示例

11.3.拡張メニューの操作方法

ウォッチウィンドウの拡張メニューには、ウォッチポイントの登録/削除などを行うためのメニューが備えられています。ウォッチウィンドウの拡張メニューは、ウォッチウィンドウがアクティブである場合にPDB38Mウィンドウの基本メニュー [Option] に追加されます。ウォッチウィンドウの拡張メニューの構成を表 11-1に示します。

表 11-1 ウォッチウィンドウの拡張メニュー

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	
	<u>A</u> dd...	ウォッチポイントの登録	Ctrl + A
	<u>B</u> itadd...	ビットレベルのウォッチポイントの登録	Ctrl + B
	<u>S</u> et...	選択位置のウォッチポイントのメモリ内容の設定	Ctrl + S
	<u>D</u> el	選択位置のウォッチポイントの削除	Ctrl + D
	<u>D</u> elAll	全ウォッチポイントの削除	
	Radix	表示基数の変更	
	<u>B</u> in	選択位置のウォッチポイントの値を 2 進数で表示	Alt + B
	<u>D</u> ec	選択位置のウォッチポイントの値を 10 進数で表示	Alt + D
<u>H</u> ex	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		
<u>A</u> ddress Area	アドレス/ビット表示領域の表示/非表示		
<u>S</u> ize Area	データ長表示領域の表示/非表示		

以下に、ウォッチウィンドウの拡張メニューの操作方法を説明します。

□ Font メニュー

表示領域のフォントを変更します。

Font メニューを選択すると、フォント指定ダイアログがオープンします。図 11-4にフォント指定ダイアログの構成図を示します。フォントを指定してください。

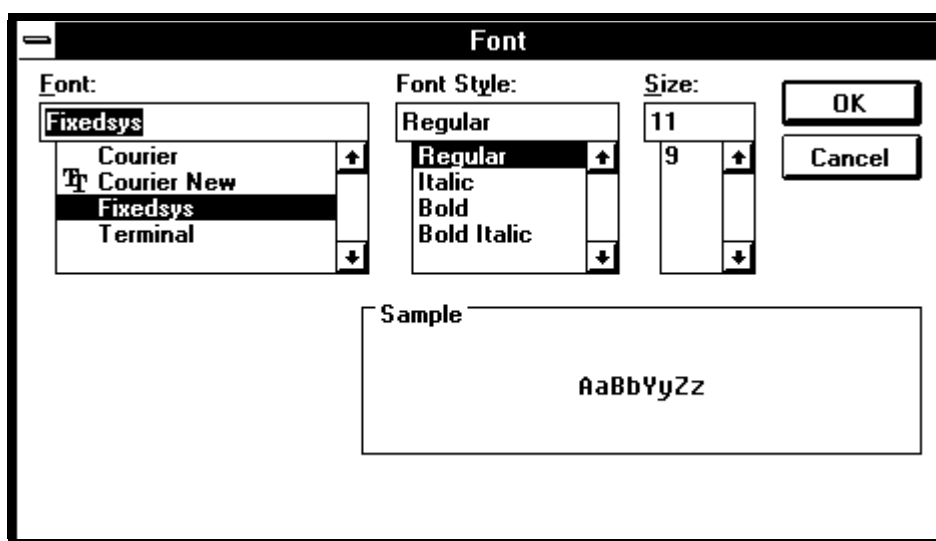


図 11-4 フォント指定ダイアログ

□ Watch メニュー

ウォッチポイントの登録/削除/値の設定などを行います。

Watch メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Add サブメニュー

ウォッチポイントを登録します。

Add サブメニューを選択すると、Add ダイアログがオープンします。図 11-5に Add ダイアログの構成図を示します。ウォッチポイントとなるアドレス式、データ長、基数を指定してください。

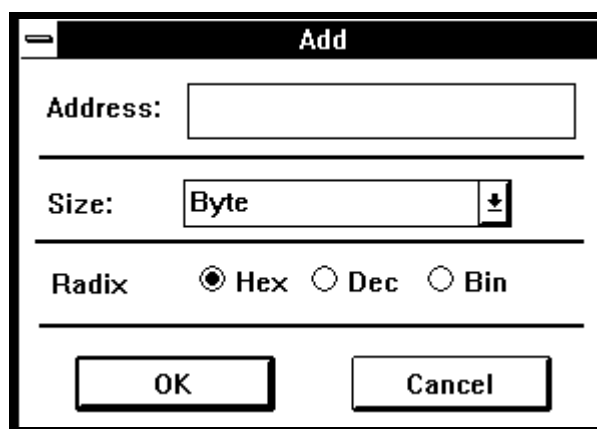


図 11-5 Add ダイアログ

- Address:領域にアドレス式、Size:領域にデータ長を指定してください。基数は Radix:領域内のラジオボタンから選択してください。
そのアドレス式をウォッチポイントとして登録し、指定データ長のメモリ内容を指定基数で表示します。
- アドレス式の記述の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
なおこのときの定数の基数は、16進数です。
- データ長は、Byte (1バイト長)、Word (2バイト長)の何れかが選択できます。Size:領域横のドロップダウンリスト(下向きの矢印をマウスでクリックすると表示)により選択してください。
- 基数は、Hex (16進数)、Dec (10進数)、Bin (2進数)が指定できます。Radix:領域内のラジオボタンをクリックして選択してください。
- アドレス式が正しく計算できなかった場合(例えばシンボルが未定義の場合)、図 11-6に示すウォーニングダイアログを表示します。このダイアログの Yes ボタンをクリックすると、そのアドレス式を無効なウォッチポイントとして登録します。無効なウォッチポイントは、メモリ表示の対象になりません(アドレス/ビット表示領域は空白、データ表示領域は常に “--<not active>--” で表示します)。
無効なウォッチポイントは、再ダウンロード、ステップ実行、ブレーク等の際にアドレス式を計算した結果、正しく計算できた場合には有効なウォッチポイントに変わります。

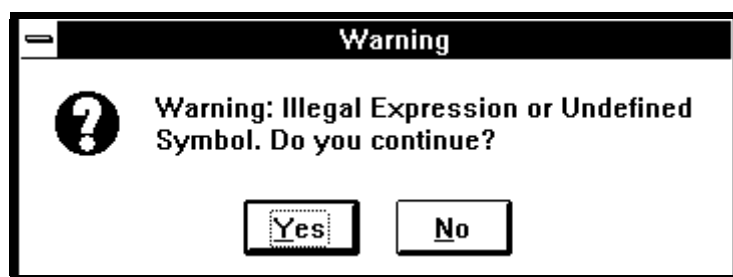


図 11-6 ウォーニングダイアログ

- 追加するウォッチポイントは、現在のカーソル位置に挿入します。カーソル位置は、アドレス式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。カーソル位置は、どちらかの領域をマウスでシングルクリックするか、または キーで移動できます。

Bitadd サブメニュー

ビットレベルのウォッチポイントを登録します。

Bitadd サブメニューを選択すると、BitAdd ダイアログがオープンします。図 11-7に BitAdd ダイアログの構成図を示します。ウォッチポイントとなるビットシンボル、またはアドレスとビット番号を指定してください。

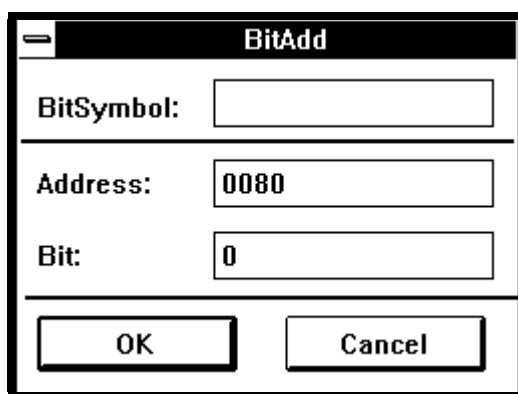


図 11-7 BitAdd ダイアログ

- ウォッチポイントをビットシンボルで指定する場合は、BitSymbol:領域にビットシンボルを入力してください。
アドレスとビット番号で指定する場合は、Address:領域にアドレス、Bit:領域にビット番号(0~7)を入力してください。アドレスとビット番号を指定する場合は、アドレス/ビット番号の両方の指定が必要です。
- アドレス、ビット番号の入力には、式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載されています。
なおこのときの定数の基数は、16進数です。

- 追加するウォッチポイントは、現在のカーソル位置に挿入します。
カーソル位置は、アドレス式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。カーソル位置は、どちらかの領域をマウスでシングルクリックするか、または□□キーで移動できます。

Set サブメニュー

カーソル位置のウォッチポイントのメモリ内容を設定します。

Set サブメニューを選択すると、Set ダイアログがオープンします。図 11-8に Set ダイアログの構成図を示します。ウォッチポイントに代入するデータを指定してください。

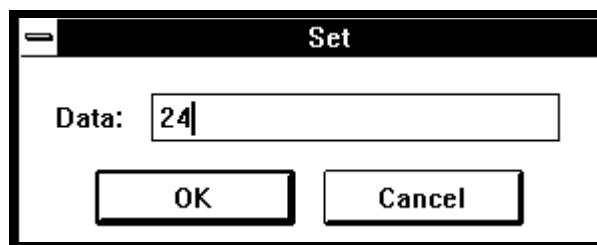


図 11-8 Set ダイアログ

- Data:領域にデータを入力してください。
カーソル位置のウォッチポイントのメモリ内容を、指定データに変更します。
- データの指定には、式が入力できます。式の仕様は、スクリプト仕様編の「式の記述方法」に記載しています。
なおこの時の定数の基数は、ウォッチポイントを 10 進数で表示している場合は 10 進数、その他の基数で表示している場合は 16 進数になります。

Del サブメニュー

カーソル位置のウォッチポイントを削除します。

Del All サブメニュー

現在登録されている全てのウォッチポイントを削除します。

Del All サブメニューを選択すると、クエスチョン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。なお、ここで Cancel ボタンをクリックすると、ウォッチポイントは削除しません。図 11-9にクエスチョンダイアログの構成図を示します。

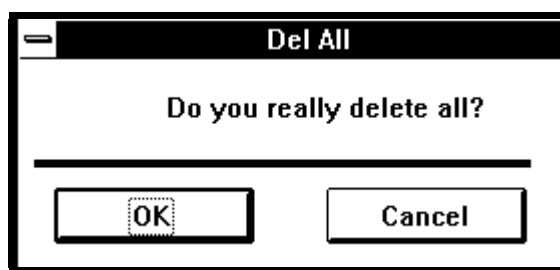


図 11-9 クエスチョンダイアログ

□ Radix メニュー

ウォッチポイントの基数を指定します。

Radix メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Bin サブメニュー

カーソル位置のウォッチポイントのメモリ内容を、2 進数で表示します。

Dec サブメニュー

カーソル位置のウォッチポイントのメモリ内容を、10 進数で表示します。

Hex サブメニュー

カーソル位置のウォッチポイントのメモリ内容を、16 進数で表示します。

□ Layout メニュー

ウォッチウィンドウのレイアウトを設定します。

Layout メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Address Area サブメニュー

アドレス/ビット番号表示領域の表示/非表示を指定します。

Address Area サブメニューを選択すると、アドレス/ビット番号表示領域を表示します。選択を解除すると、アドレス/ビット番号表示領域は表示しません。

Size Area サブメニュー

データ長表示領域の表示/非表示を指定します。

Size Area サブメニューを選択すると、データ長表示領域を表示します。選択を解除すると、データ長表示領域は表示しません。

11.4. ツールバーの構成

ウォッチウィンドウのツールバーは、図 11-10 に示すボタンで構成されています。

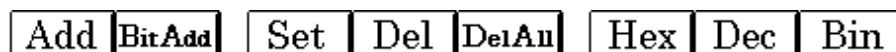


図 11-10 ウォッチウィンドウのツールバーの構成

各ボタンの操作方法は以下の通りです。

○ Add ボタン

ウォッチポイントを登録します。

動作はメニュー [Option] [Watch] [Add] を選択した場合と同じになります。

○ BitAdd ボタン

ビットレベルのウォッチポイントを登録します。

動作はメニュー [Option] [Watch] [Bitadd] を選択した場合と同じになります。

○ Set ボタン

マウスのシングルクリックで選択されたウォッチポイントのメモリ内容を設定します。

動作はメニュー [Option] [Watch] [Set] を選択した場合と同じになります。

○ Del ボタン

マウスのシングルクリックで選択されたウォッチポイントを削除します。

動作はメニュー [Option] [Watch] [Del] を選択した場合と同じになります。

○ DelAll ボタン

すべてのウォッチポイントを削除します。

動作はメニュー [Option] [Watch] [DelAll] を選択した場合と同じになります。

○ Hex ボタン

マウスのシングルクリックで選択されたウォッチポイントのメモリ内容を、16 進数で表示します。

動作はメニュー [Option] [Radix] [Hex] を選択した場合と同じになります。

○ Dec ボタン

マウスのシングルクリックで選択されたウォッチポイントのメモリ内容を、10 進数で表示します。

動作はメニュー [Option] [Radix] [Dec] を選択した場合と同じになります。

○ Bin ボタン

マウスのシングルクリックで選択されたウォッチポイントのメモリ内容を、2進数で表示します。

動作はメニュー [Option] [Radix] [Bin] を選択した場合と同じになります。

11.5.ウォッチウィンドウの表示内容の更新について

ウォッチウィンドウでは、以下の条件でウォッチポイントのメモリ内容の表示を自動的に更新します。

- メモリ内容の変更を伴うコマンドを実行した場合
メモリ内容の変更を伴うコマンド（メモリの設定・充填、実行停止、ステップ実行など）を実行した際、すべてのウォッチポイントのメモリ内容の表示を自動的に更新します。したがって、ステップ実行などのコマンド実行に伴うメモリ内容の変化を確認することができます。

11.6.ウォッチポイントのアドレス再計算機能について

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。

11.7.ウォッチポイントの保存機能について

ウォッチウィンドウをクローズする際、またはPDB38Mを終了する際、ウォッチウィンドウに登録されているすべてのウォッチポイントを環境設定ファイルに保存します。これにより、ウォッチウィンドウを再オープンした際、またはPDB38Mを再起動した際に、以前のウォッチポイントを自動的に登録することができます。

12. スクリプトウィンドウの操作方法

12.1. スクリプトウィンドウの概要

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を実行/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。スクリプトファイルから実行した場合は、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ログファイルに保存できます。

なお、PDB38Mで実行できるスクリプトコマンドについてはスクリプト仕様編の「スクリプトコマンドの概要」に、スクリプトファイルの記述方法についてはスクリプト仕様編の「スクリプトファイルの記述方法」に、それぞれ詳細を記載しています。あわせて参照してください。

スクリプトウィンドの構成

図 12-1にスクリプトウィンドウの構成図を示します。

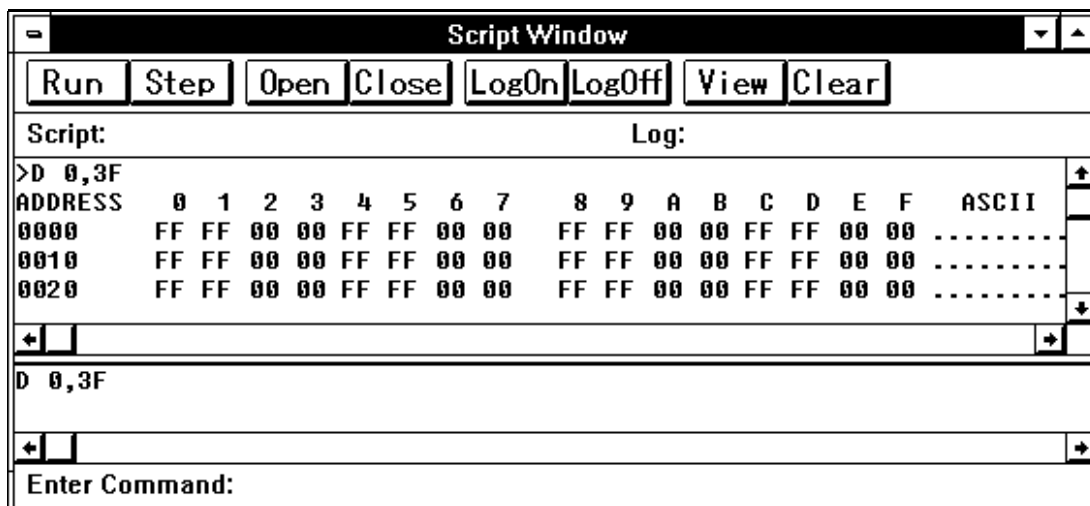


図 12-1 スクリプトウィンドウの構成図

スクリプトウィンドウの各部品について、以下で説明します。

12.1.1. ツールバー

ツールバーの各ボタンの機能についての詳細は、本章の12.4節「ツールバーの構成」を参照してください。

12.1.2. ファイル名表示領域

スクリプトファイル、ログファイルをオープンしているとき、そのスクリプトファイル名、ログファイル名を表示します。

12.1.3. コマンド表示領域

スクリプトコマンドの実行結果を表示します。

12.1.4. コマンドヒストリ表示領域

対話入力指定したスクリプトコマンド文字列を保存します（最新の100個までを保存します）。

マウスのシングルクリックでコマンド入力領域にコマンド文字列をコピーできます。また、マウスのダブルクリックで指定したコマンド文字列を実行できます。

12.1.5. コマンド入力領域

対話入力でスクリプトコマンドを入力するための領域です。

12.1.6. スクリプトファイル表示領域

スクリプトファイルをオープンしているときのみ現れ、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、一番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。図 12-2にスクリプトファイルをオープンしたときのスクリプトウィンドウの構成図を示します。

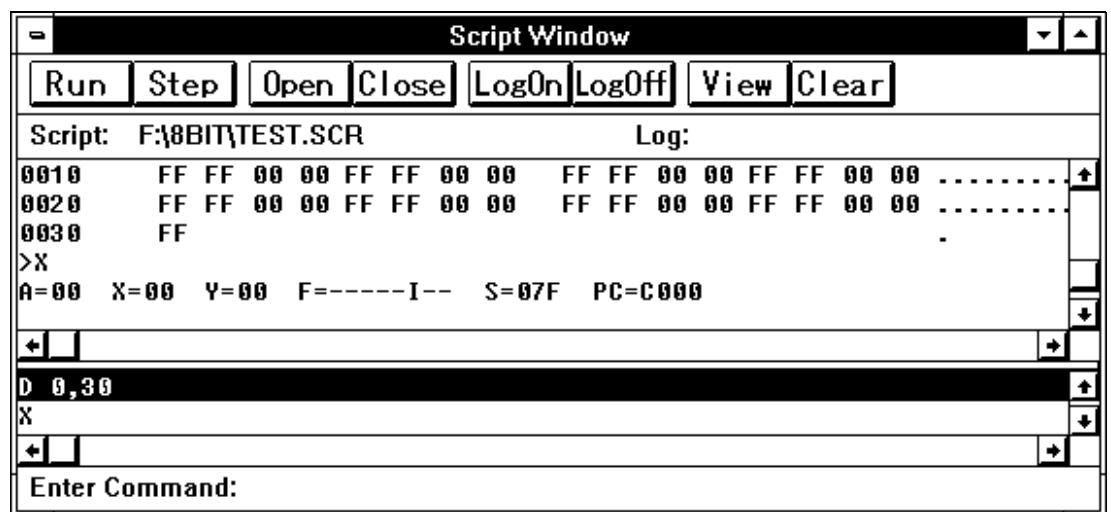


図 12-2 スクリプトファイルオープン時のスクリプトウィンドウ構成図

スクリプトファイルをオープンすると、かわりにコマンド履歴領域が消去されます。

この状態で、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) やメニュー [Option] [Script] [Step] (またはスクリプトウィンドウの Step ボタン) を選択することにより、スクリプトファイルを実行します。また、メニュー [Option] [Script] [Close] (またはスクリプトウィンドウの Close ボタン) を選択することにより、スクリプトファイルをクローズします。詳細は、各メニューの操作方を参照してください。

なお、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。

12.2.スクリプトコマンドの実行中断方法について

コマンド入力領域やスクリプトファイルからスクリプトコマンドを実行すると、そのコマンドの処理が終了するまで、PDB38Mは他の一切の操作を受け付けなくなります。

コマンド実行中は、図 12-3のダイアログを表示します(コマンド実行が終了すれば、このダイアログは自動的に閉じます)。このダイアログが表示している間は、PDB38Mの他の一切の操作が行えません。なお、コマンド実行を途中で中断する場合は、このダイアログの STOP ボタンをクリックしてください。



図 12-3 スクリプトコマンド実行中のダイアログ表示例

12.3.拡張メニューの操作方法

スクリプトウィンドウの拡張メニューには、スクリプトファイルの実行・停止・ステップ実行・クローズなどを操作するためのメニューが備えられています。スクリプトウィンドウの拡張メニューは、スクリプトウィンドウがアクティブである場合にPDB38Mウィンドウの基本メニュー [Option] に追加されます。

スクリプトウィンドウの拡張メニューの構成を表 12-1に示します。

表 12-1 スクリプトウィンドウの拡張メニュー

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	
	<u>O</u> pen...	スクリプトファイルのオープン	
	<u>R</u> un	スクリプトファイルの実行	
	<u>S</u> top	スクリプトファイルの実行停止	
	<u>S</u> tep	スクリプトファイルのステップ実行	
	<u>C</u> lose	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	
	<u>S</u> ave...	ビューバッファのファイル保存	
	<u>C</u> lear	ビューバッファのクリア	
	Log	ログファイルの操作	
	<u>O</u> n...	ログファイルのオープン（出力開始）	
<u>O</u> ff	ログファイルのクローズ（出力終了）		

以下に、スクリプトウィンドウの拡張メニューの操作方法を説明します。

□ Font メニュー

表示領域のフォントを変更します。

Font メニューを選択すると、フォント指定ダイアログがオープンします。図 12-4にフォント指定ダイアログの構成図を示します。フォントを指定してください。

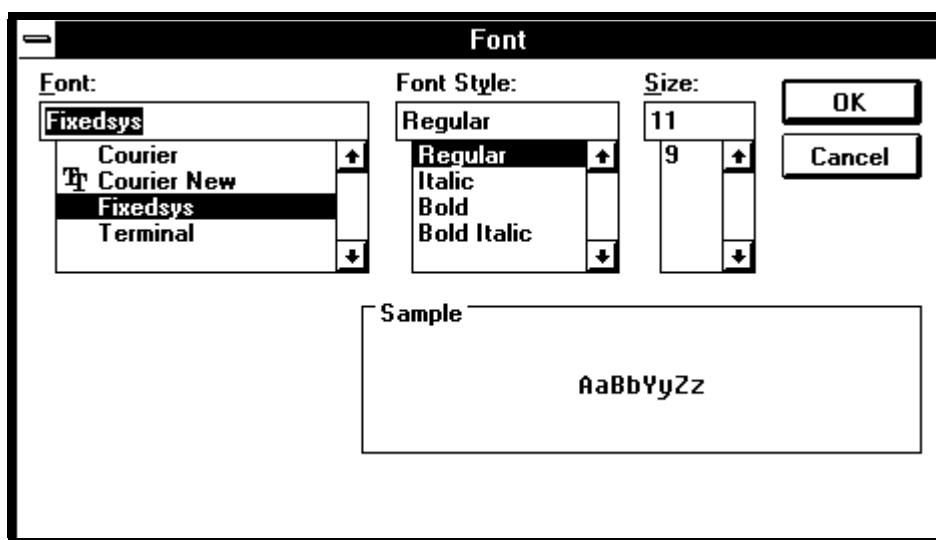


図 12-4 フォント指定ダイアログ

□ Script メニュー

スクリプトファイルのオープン/クローズ/実行関連の操作を行います。

Script メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Open サブメニュー

スクリプトファイルをオープンします。

Open サブメニューを選択すると、ファイルセレクションダイアログがオープンします。図 12-5にファイルセレクションダイアログの構成図を示します。オープンするスクリプトファイルを選択してください。

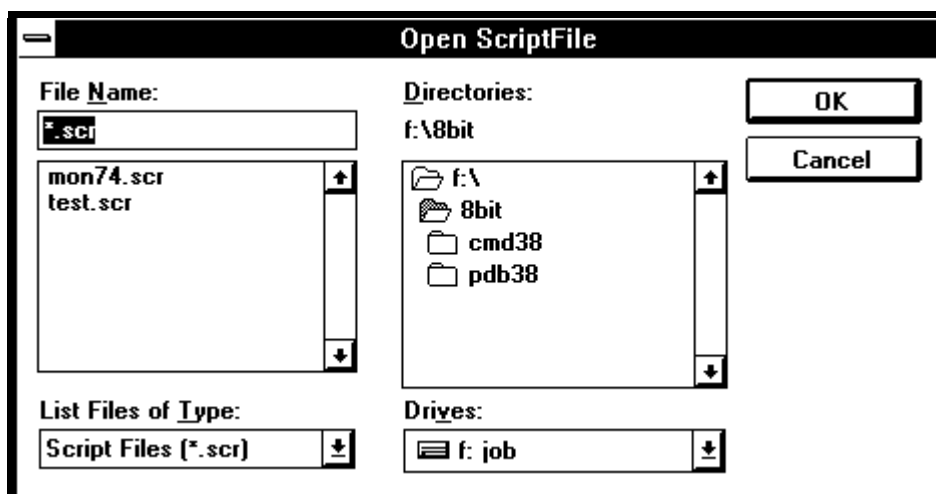


図 12-5 ファイルセレクションダイアログ

- スクリプトファイルには、ふつう拡張子が .scr のファイル名を指定します(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 以外のファイルを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名を選択し OK ボタンをクリックすると、指定したスクリプトファイルがオープンします。
スクリプトファイルがオープンすると、コマンド履歴表示領域が消去し、スクリプトファイル表示領域が現れます。
- スクリプトファイルは、ネストしてオープンできます。最大、5 段までネストできます。

Run サブメニュー

スクリプトファイルを実行します。

スクリプトファイルの実行は、スクリプトファイル実行中に表示する図 12-6のダイアログの STOP ボタンをクリックすることにより停止できます。

なお、このダイアログは、スクリプトコマンド実行中に表示するダイアログと同じものです。このダイアログでコマンド実行を中断した時、そのコマンドがスクリプトファイルから連続実行したものだ場合は、コマンド実行の中断と同時にスクリプトファイルの実行も、次の行が実行される手前で停止します。



図 12-6 スクリプトファイル実行中のダイアログ表示例

Stop サブメニュー

スクリプトファイルの実行を停止します。

Step サブメニュー

スクリプトファイルを 1 コマンドごとにステップ実行します。

スクリプトコマンドの実行結果を確認しながら、スクリプトファイルを少しずつ実行できます。

スクリプトファイル表示領域で反転表示されている行を 1 コマンド分、実行した後に停止します。

Close サブメニュー

スクリプトファイルをクローズします。

スクリプトファイルをクローズすると、スクリプトファイル表示領域が消去し、再びコマンド履歴表示領域が現れます。

□ View メニュー

ビューバッファの保存/クリアのためのメニューが割り当てられています。

View メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

Save サブメニュー

現在のビューバッファの内容をファイルに保存します。このファイルのことをビューファイルといいます。

Save サブメニューを選択すると、ファイルセレクションダイアログがオープンします。図 12-7にファイルセレクションダイアログの構成図を示します。ビューファイル名を指定してください。

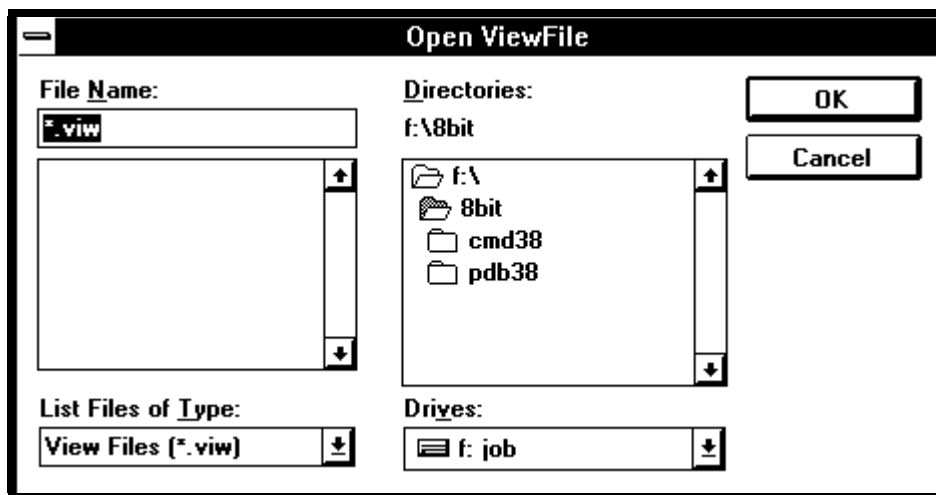


図 12-7 ファイルセレクションダイアログ

- ビューファイル名には、任意のファイル名が指定可能です。
- 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。
- ビューバッファは、最新の 1000 行分のコマンド実行結果を記憶しています。

Clear サブメニュー

ビューバッファの内容をクリアします。

□ Log メニュー

ログファイルのオープン/クローズ関連のメニューが割り当てられています。

Log メニューを選択すると、以下のサブメニューが表示されます。目的に応じて選択してください。

On サブメニュー

ログファイルをオープンし、それ以降、スクリプトコマンドの実行結果をログファイルに出力します。

On サブメニューを選択すると、ファイルセレクションダイアログがオープンします。図 12-8にファイルセレクションダイアログの構成図を示します。ログファイル名を指定してください。

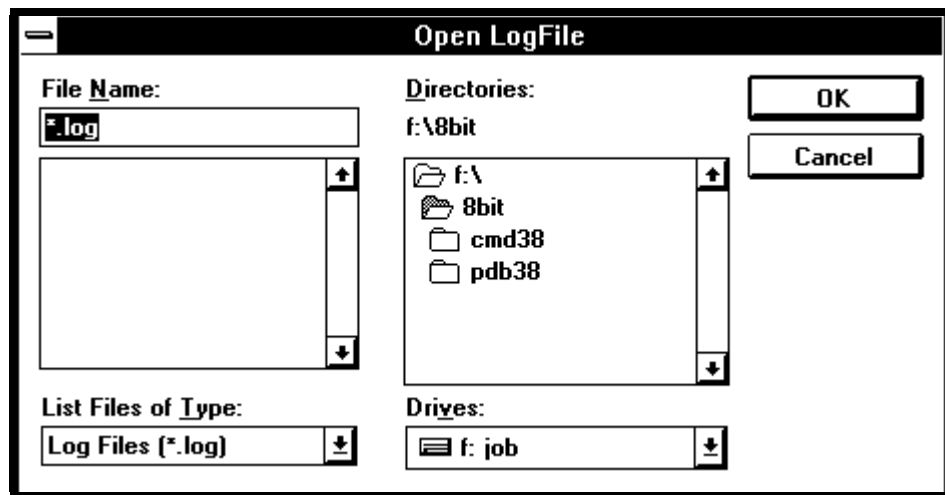


図 12-8 ファイルセレクションダイアログ

- ログファイルには、拡張子が .log のファイル名を指定してください。
- PDB38M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PDB38M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。
- ログファイルは、ネストしてオープンできます。最大、8段までネストできません。

Off サブメニュー

現在オープン中のログファイルをクローズします。

これ以降、ログファイルへの出力は行いません。ただし、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。

12.4. ツールバーの構成

スクリプトウィンドウのツールバーは、図 12-9に示すボタンで構成されています。

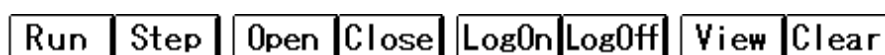


図 12-9 スクリプトウィンドウのツールバーの構成

各ボタンの操作方法は以下の通りです。

○ Run ボタン

スクリプトファイルを実行します。

動作はメニュー [Option] [Script] [Run] を選択した場合と同じになります。

○ Step ボタン

スクリプトファイルを1コマンドごとにステップ実行します。

動作はメニュー [Option] [Script] [Step] を選択した場合と同じになります。

○ Open ボタン

スクリプトファイルをオープンします。

動作はメニュー [Option] [Script] [Open] を選択した場合と同じになります。

○ Close ボタン

スクリプトファイルをクローズします。

動作はメニュー [Option] [Script] [Close] を選択した場合と同じになります。

○ LogOn ボタン

ログファイルをオープンし、それ以降の実行結果をログファイルに出力します。

動作はメニュー [Option] [Log] [On] を選択した場合と同じになります。

○ LogOff ボタン

現在オープン中のログファイルをクローズします。

動作はメニュー [Option] [Log] [Off] を選択した場合と同じになります。

○ View ボタン

現在のビューバッファの内容をファイルに保存します。

動作はメニュー [Option] [View] [Save] を選択した場合と同じになります。

○ **Clear ボタン**

ビューバッファの内容をクリアします。

動作はメニュー [Option] [View] [Clear] を選択した場合と同じになります。

13. S/Wブレークポイント設定ダイアログの操作方法

13.1.S/W ブレークポイント設定ダイアログの概要

S/W 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。PDB38Mウィンドウのメニュー[Debug] [Break Point] [S/W Break Point] を選択するとオープンします。

ソフトウェアブレークポイントとは、指定したアドレスの命令を実行する手前でブレークするブレークポイントのことです。

- PDB38Mでは、4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、何れか 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機能では、ソフトウェアブレークポイントの命令を実行する前にブレークします。

13.2.S/W ブレークポイント設定ダイアログの構成

図 13-1に S/W ブレークポイント設定ダイアログの構成図を示します。

S/W Break Point:		
C000	[51]	TARGET.A74
*C007	[58]	TARGET.A74
C003	[54]	TARGET.A74
C006	[57]	TARGET.A74

図 13-1 S/W ブレークポイント設定ダイアログ

13.3.S/W ブレークポイント設定ダイアログの操作方法

S/W ブレークポイント設定ダイアログでは、以下の操作が行えます。

1. ソフトウェアブレークポイントの参照
2. ソフトウェアブレークポイントの設定
3. ソフトウェアブレークポイントの削除
4. ソフトウェアブレークポイントの無効化
5. ソフトウェアブレークポイントの有効化

なお、ソフトウェアブレークポイントに関するこれらの指定は、Close ボタンをクリックして S/W 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。

次節より、各指定の際の操作方法を説明します。

13.3.1.ソフトウェアブレークポイントの参照

S/W Break Point:領域には、現在設定されているソフトウェアブレークポイントの一覧が表示されます。S/W Break Point:領域の垂直スクロールバーを操作することにより、全てのソフトウェアブレークポイントを参照することができます。

- 一覧の表示内容は、ソフトウェアブレークポイントのアドレスと、そのアドレスに相当するソース行頭の行番号およびソースファイル名です。

表示例: `0F0047 [62] TEST1.c` アドレス [行番号] ファイル名

- 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。

表示例: `0F00CD` アドレス

- さらに、そのソフトウェアブレークポイントが無効の場合は、その行の左端に無効化を示す「*」を表示します。

表示例: `* 0F0069 [90] TEST1.c` アドレス [行番号] ファイル名

13.3.2.ソフトウェアブレークポイントの設定

アドレス指定、または ソースファイル名 + 行番号指定で、ソフトウェアブレークポイントを設定することができます。

- アドレス指定でソフトウェアブレークポイントを設定する場合
Address ラジオボタンをクリックしてから、Address:領域にアドレスを入力し、最後に Add ボタンをクリックしてください。
 - アドレスの指定には式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」を参照してください。なお、このときの定数の基数は16進です。

- ソースファイル名 + 行番号指定でソフトウェアブレークポイントを設定する場合
Filename ラジオボタンをクリックしてから、Filename:領域にソースファイル名、Line:領域に行番号を入力し、最後に Add ボタンをクリックしてください。
- 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。このダイアログから、指定するソースファイルを選択することができます。図 13-2に、ファイルセレクションダイアログの構成図を示します。

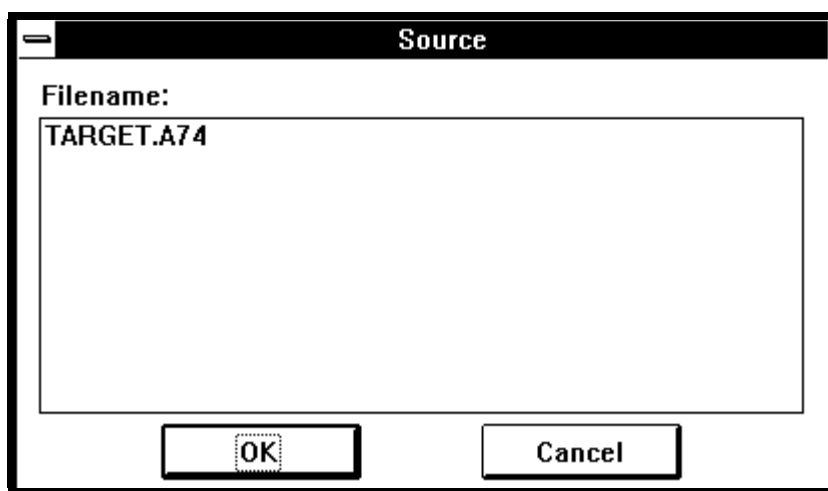


図 13-2 ファイルセレクションダイアログ

- 行番号の指定には式が記述できます。式の仕様は、スクリプト仕様編の「式の記述方法」を参照してください。なお、このときの定数の基数は 16 進です。

13.3.3.ソフトウェアブレークポイントの削除

任意のソフトウェアブレークポイントを削除する方法と、全てのソフトウェアブレークポイントを一度に削除する方法があります。

- 指定したソフトウェアブレークポイントを削除する場合
削除するソフトウェアブレークポイントを選択し、Del ボタンをクリックしてください。
なお、ソフトウェアブレークポイントの選択は、S/W Break Point:領域に表示されている任意のソフトウェアブレークポイントをマウスでシングルクリックすることで行います。
- 現在設定されている全てのソフトウェアブレークポイントを削除する場合
Del All ボタンをクリックしてください。
Del All ボタンをクリックすると、クエスチョン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。なお、ここで Cancel ボタンをクリックすると、ソフトウェアブレイクポイントは削除しません。図 13-3にクエスチョンダイアログの構成図を示します。

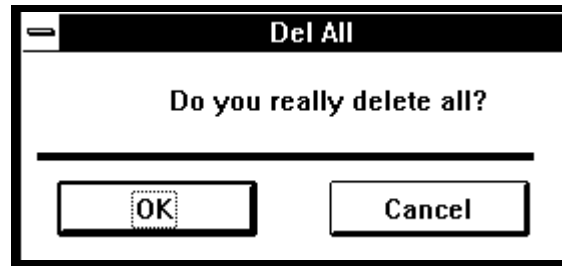


図 13-3 クエスチョンダイアログ

13.3.4.ソフトウェアブレイクポイントの無効化

任意のソフトウェアブレイクポイントを無効化する方法と、全てのソフトウェアブレイクポイントを一度に無効化する方法があります。

- 指定したソフトウェアブレイクポイントを無効化する場合
無効化するソフトウェアブレイクポイントを選択し、Disable ボタンをクリックしてください。
なお、ソフトウェアブレイクポイントの選択は、S/W Break Point:領域に表示されている任意のソフトウェアブレイクポイントをマウスでシングルクリックすることで行います。
 - 指定したソフトウェアブレイクポイントの左端に、無効化を示す ' * ' が表示されます。
 - すでに無効化されているソフトウェアブレイクポイントに対して本操作を行っても、無視されます。
- 現在設定されている全てのソフトウェアブレイクポイントを無効化する場合
All Disable ボタンをクリックしてください。

13.3.5.ソフトウェアブレイクポイントの有効化

任意のソフトウェアブレイクポイントを有効化する方法と、全てのソフトウェアブレイクポイントを一度に有効化する方法があります。

- 指定したソフトウェアブレイクポイントを有効化する場合

有効化するソフトウェアブレイクポイントを選択し、 Enable ボタンをクリックしてください。

なお、ソフトウェアブレイクポイントの選択は、S/W Break Point:領域に表示されている任意のソフトウェアブレイクポイントをマウスでシングルクリックすることで行います。

- 指定したソフトウェアブレイクポイントの、左端の無効化を示す ‘ * ’ が消去します。
 - すでに有効化されているソフトウェアブレイクポイントに対して本操作を行っても、無視されます。
-
- 現在設定されている全てのソフトウェアブレイクポイントを有効化する場合
All Enable ボタンをクリックしてください。

スクリプト仕様編

14. スクリプトコマンドの概要

14.1. コマンド一覧

以下に、PDB38Mのスクリプトウィンドウ内で実行可能なコマンド（スクリプトコマンド）を、表 14-1に示しす。

表 14-1 コマンド一覧(実行と停止)

コマンド名	マンドの機能概要
A	一行単位でアセンブルを行う。
BIT	ビットシンボルに該当するビットを参照及び変更する。
CD	カレントディレクトリの変更/参照を行う。
D	メモリの内容をバイト単位で表示する。
F	指定したメモリ範囲に指定したバイトデータを書き込む。
G	ターゲットプログラムを実行する。
GB	ターゲットプログラムを実行する（PBで設定したブレークポイントが有効）。
I	HEXファイル、SYMファイルを読み込む。
L	プログラムメモリの内容を逆アセンブル表示する。
LOGOFF	PDB38Mの実行結果をログファイルに出力するのを停止する。
LOGON	PDB38Mの実行結果をログファイルに出力する。
O	現在のメモリ内容をファイル（HEXファイル）に出力する。
PB	プログラムブレークポイントを設定する。
RADIX	数値入力の既定値の表示及び変更を行う。
S	メモリの内容をバイト単位で表示及び変更する。
SCOPE	ローカルラベル/シンボルの有効範囲を指定する。
SCRIPT	スクリプトファイルをオープンする。
SI	セクション情報を表示する。
SRC	プログラムウィンドウに表示するソースファイル名を指定する。
STOP	ターゲットプログラムの実行を停止する。
T	プログラムを1命令単位で実行し、レジスタ内容等を表示する。
U	プログラムを指定命令数だけ実行する。（実行前にレジスタ内容を表示し、実行中は表示なし）
VER	PDB38Mのバージョン番号を表示する。
WAIT	ターゲットプログラムのブレークまで待機する。
X	レジスタ内容を表示及び変更する。
Z	ターゲットMCUをハードウェアリセットする。
;	コメント行

14.2.コマンドの入力形式

PDB38M では、図 14-1の書式でコマンド入力を受け付けます。

```
コマンド名 コマンド種別 パラメータ 1, 副パラメータ 1  
          , パラメータ 2, 副パラメータ 2 ..... <RET>
```

図 14-1 コマンドの入力形式

- コマンド名とパラメータ、コマンド名とコマンド種別、コマンド種別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。
- コマンド名及びコマンド種別には、大小の英数字が使用できます。
- コマンド種別の意味、各パラメータの意味、省略の可否等は、各コマンドによって異なります。

14.3.コマンドの詳細説明

14.3.1.コマンド詳細の記述内容

各コマンドの詳細説明の読み方は、以下のとおりです。

- **タイトル**
コマンド名、およびコマンドの機能概要を示します。

コマンド名

コマンドの機能概要

- **本文**
以下の5項目について説明します。

入力書式

コマンドの入力書式を示します。

入力書式での表記方法については、「コマンド入力書式の表記方法」で説明します。

ランタイム

プログラム実行中にコマンド実行が可能であるか否かを示します。

機能

コマンドの機能を説明します。

パラメータ

入力書式で示したパラメータについて説明します。

パラメータが存在しない場合、「なし」と記述します。

コマンド実行例

コマンドの実行例を示します。

14.3.2. コマンド入力書式の表記方法

コマンド入力形式の説明においては、以下の表記方法を用います。

表 14-2 コマンド入力書式の表記方法

表記方法	意味
XXXX	XXXX を入力する必要があります。
[XXXX]	入力を省略するか、または XXXX が入力可能です。
{ X1 X2 X3 }	X1、X2、X3 の内、何れか 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、または X1、X2、X3 の内、何れか 1 つが入力可能です。

14.3.3. 各コマンドの詳細説明

以下に、スクリプトウィンドウ内で実行するコマンドの詳細についてアルファベット順に記載します。

A (Assemble)

1 行単位アセンブル

入力書式

Assemble [アドレス]

ランタイム

実行不可

機能

- 指定したアドレスまたはラベルが示すアドレスから 1 行単位のアセンブルを行い、対応した機械語をプログラムメモリに書き込みます。(例 1)
- 入力されたニーモニックまたはオペランドの誤りを検出した場合、アSEMBルエラーメッセージを表示し同じアドレスでの入力を待ちます。(例 2)
- ジャンプ先のアドレスが、指定したアドレッシングモードの範囲外の場合、エラーメッセージを表示し、同じアドレスでの入力を待ちます。(例 3)
- ラベル欄に記述したラベルまたはシンボル、ビットシンボルは、グローバルラベルまたはグローバルシンボルとして登録されます。
- ラベル定義の際にはラベル名の最後に必ず '?' を指定してください。(例 4)
- コマンド入力時にアドレスが省略された場合、前回の A コマンドが処理した最終アドレスからアセンブルを始めます。(例 5)

ただし、前回の実行で FFFFh 番地に達した場合はエラーメッセージを表示します。(例 17)

なお、PDB38M 起動時はプログラムカウンタからアセンブルを始めます。

- 擬似命令として以下の 4 命令が使用できます。

擬似命令	機能
.BLKB	RAM 領域の 1 バイト確保 (範囲 1 ~ 0x10000) (例 8)
.BYTE	バイトデータの設定 (8 バイト分まで設定可能)
.EQU	シンボル定義 (例 8) ビットシンボル定義 (例 9)
.WORD	ワードデータの設定 (4 バイト分まで設定可能) (例 10)

- A コマンドを中止する場合は、“<RET>”のみを入力してください。(例 11)ま

た、空白文字だけ入力された状態で定義で “ <RET> ” を入力した場合も A コマンドは中止します。(例 14)

- ラベル、シンボル、ビットシンボルの定義で指定した名が既に定義されている場合、二重定義エラーメッセージを表示します。(例 12)
- ラベル、シンボル、ビットシンボルの定義で指定した名が異常な場合、エラーメッセージを表示します。(例 13)
- FFFFh 番地を越えた場合、文法が正常でもアセンブルエラーを表示します。(例 15)
- FFFFh 番地に達した場合、A コマンドを中止します。(例 16)
- A で始まる数値を入力する場合は、アキュムレータと区別するために先頭に '0' もしくは '\$'、または後尾に 'h' もしくは 'H' を付けて指定してください。(例 18)
- ビットシンボルの使用が可能です。(例 19)
- ビットシンボルはビット値のみまたはアドレス値のみとしても使用可能です(但し、アドレス値のみとしては LDA または STA 命令しか使用できません)。
- アドレッシングモードを以下のように指定することができます。
 - イミディットアドレッシング(例 20)
このアドレッシングを使用する場合は、イミディットデータの前に '#' を不可してください。
 - ゼロページアドレッシング(例 21)
このアドレッシングを使用する場合は、0h 番地から FFh 番地の間を指定してください。
 - アブソリュートアドレッシング(例 22)
このアドレッシングを使用する場合は、オペランドに対象アドレスを記述してください。
 - レラティブアドレッシング(例 23)
このアドレッシングを使用する場合は、オペランドに対象アドレスをそのまま記述してください。
 - スペシャルページアドレッシング(例 24)
このアドレッシングを使用する場合は、スペシャルページのアドレスを指定し、アドレスの前に '\$' を付加してください。
 - インダイレクトアドレッシング(例 25)
このアドレッシングを使用する場合は、アドレスの前後に '(' を付加してください。
- BRK 命令(1 バイト命令)をアセンブルする場合、メモリへはオペコード(0x00)と 0xEA(NOP)を書き込みます。アセンブルアドレスも 2 バイト更新します。(例

26)

パラメータ

- アドレスには、式が記述できます。アドレスが省略された場合は、前回実行した A コマンドの処理が終了したアドレスから処理が始まります。
- アドレスに指定できる値の範囲は、0000h ~ FFFFh までです。

コマンド実行例

```

>A E0000 (例 1)
ADDRESS> LABEL PROGRAM
E000 START: SEI
E001 BAT MNEMONIC (例 2)
ERROR 1101: アセンブリ言語の記述に誤りがあります。
E001 BRA E083 (例 3)
ERROR 1102: 指定したオペランドの値が範囲外です。
E001 Label: (例 4)
E001 Label:
>A (例 5)
ADDRESS> LABEL PROGRAM
E001 Label: .BLKB 5 (例 6)
E006 .BYTE 0,'0',FF (例 7)
E009 Symbol .Equ 0 (例 8)
E009 BitSymbol .Equ 0,0 (例 9)
E009 .WORD 0,'0',FFFF (例 10)
E00F (例 11)
>A
ADDRESS> LABEL PROGRAM
E00F Label: (例 12)
ERROR 1508: 既に同名のシンボル/ラベルが登録されています。
E00F Symbol .EQU 0
ERROR 1508: 既に同名のシンボル/ラベルが登録されています。
E00F BitSymbol .Equ 0,0
ERROR 1508: 既に同名のシンボル/ラベルが登録されています。
E00F Lab-el: (例 13)
ERROR 1505: 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。
E00F Sym-bol .EQU 0
ERROR 1505: 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。
E00F Bit-Symbol .EQU 0
ERROR 1505: 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。
E00F (例 14)
>A FFFF
ADDRESS> LABEL PROGRAM
FFFF LAST_ADD: LDA #10 (例 15)
ERROR 1101: アセンブリ言語の記述に誤りがあります。
FFFF LAST_ADD: NOP (例 16)
>A (例 17)

```


ERROR 1204: 前回の実行時に最大アドレスに達しました。

>A ACCUMULATOR

```
ADDRESS> LABEL      PROGRAM
E010     ACCUMULA:  INC A           (例 18)
E011                                     INC 0A
E013                                     INC $A
E015                                     INC Ah
E017                                     INC AH
```

>A BITSYMBOL

```
ADDRESS> LABEL      PROGRAM
E020     BITSYMBO:  SEB 0,10       (例 19)
E022                                     SEB BitSymbol
E024                                     SEB BitSymbol,10
E026                                     SEB 1,A
E027                                     SEB BitSymbol,A
E028                                     LDA BitSymbol
E02A                                     STA BitSymbol
E02C
```

>A IMMEDIATE

```
ADDRESS> LABEL      PROGRAM
E030     IMMEDIAT:  LDA #10        (例 20)
E032                                     LDA #Symbol
E034
```

>A ZEROPAGE

```
ADDRESS> LABEL      PROGRAM
E040     ZEROPAGE:  LDA 10         (例 21)
E042                                     LDA ZeroPage
E044
```

>A ABSOLUTE

```
ADDRESS> LABEL      PROGRAM
E050     ABSOLUTE:  LDA E090       (例 22)
E053                                     LDA Label
E056
```

>A RELATIVE

```
ADDRESS> LABEL      PROGRAM
E060     RELATIVE:  BRA E000       (例 23)
E062                                     BRA Label
E064
```

>A SPECIAL

```
ADDRESS> LABEL      PROGRAM
E070     SPECIAL:   JSR ¥FF70      (例 24)
E072                                     JSR ¥SPECIALPAGE
E074
```

>A INDIRECT

```
ADDRESS> LABEL      PROGRAM
E080     INDIRECT:  LDA (70,X)     (例 25)
E082                                     LDA (ZeroPage,X)
E084
```

>A BRK_OPERATION

```
ADDRESS> LABEL      PROGRAM
E090     BRK_OPER:  BRK           (例 26)
E092
```

BIT (set BIT)

ビットシンボルの参照/設定

入力書式

BIT ビットシンボル名

ランタイム

実行可能

機能

- 指定したビットシンボルの値を表示，変更します。

パラメータ

- ビットシンボル名には、ビットシンボルが記述できます。(式の記述はできません)
- 変更する値には、式が記述できます。変更する値の範囲は、0~1です。

コマンド実行例

```
>BIT BITSYMBOL<RET>  
0,0005h      1      0<RET>  
>
```

- 実行例では、ビットシンボル BITSYMBOL が示すビット(0005h 番地の第 0 ビット)の値を 0 に変更しています。

CD

カレントディレクトリの設定/参照

入力書式

CD [ディレクトリ名]

ランタイム

実行可能

機能

- カレントディレクトリの設定、及び参照を行います。

パラメータ

- パラメータを省略した場合、現在のカレントディレクトリを表示します。

コマンド実行例

```
>CD
c:¥usr
>cd ¥usr¥pdb38m
c:¥usr¥pdb38m
```

ダウンロードを行いたいプログラム(program.hex 等)が c:¥usr¥pdb38m¥program にある場合の例

```
>CD ¥usr¥pdb38m¥program          (例 1)
c:¥usr¥pdb38m¥program
>I program
program.hex Now Loading.
program.hex Load end.
program.sym Now Loading.
program.sym Load end.
```

補足

- i,o,logon,src,script コマンドでファイル名を指定する場合、本コマンドで指定したディレクトリからの相対パス名を使用することができます。(例 1 を参照)
- File-Download メニューのダウンロードダイアログでファイルを選択すると、そのファイルが存在するディレクトリに、カレントディレクトリが設定されます。

D (Dump)

メモリ内容の表示

入力書式

D [開始アドレス][終了アドレス]

ランタイム

実行可能

機能

- 指定したアドレスの範囲のメモリ内容と、その ASCII データを表示します。
- 実行中に STOP コマンドまたは PDB38M ウィンドウの <Stop> ボタンを入力することにより、コマンド実行を停止することができます。

パラメータ

- 開始アドレスには式が記述できます。開始アドレスが省略された場合は前回表示した終了アドレスの次のアドレスが開始アドレスとなります。(最初に実行した場合は E000h 番地が開始アドレスとなります。)
- 終了アドレスには式が記述可能です。終了アドレスが省略された場合は、開始アドレスから 128 バイト分のメモリ内容を表示して終了します。ただし表示アドレスが FFFFh 番地に達した場合、128 バイト分のメモリ内容を表示をしないでも実行を終了します。
- 開始アドレス、終了アドレスに指定できる値の範囲は 0000h ~ FFFFh までです。

コマンド実行例

```

>D E089, E0D0                                     (例 1)
ADDRESS:  0  1  2  3  4  5  6  7      8  9  A  B  C  D  E  F  ASCII
E080:          EA EA EA EA EA EA EA EA          .....
E090:    EA EA EA EA EA EA EA EA EA EA EA EA EA EA .....
E0A0:    EA EA EA EA EA EA EA EA EA EA EA EA EA EA .....
E0B0:    EA EA EA EA EA EA EA EA EA EA EA EA EA EA .....
E0C0:    EA EA EA EA EA EA EA EA EA EA EA EA EA EA .....
E0D0:    EA                                .
>

```

F (Fill)

メモリブロックへのデータ充填

入力書式

F 開始アドレス, 終了アドレス, データ

ランタイム

実行可能

機能

- 指定したアドレスの範囲に指定したデータを書き込みます。
- 実行中に STOP コマンドまたはPDB38Mウィンドウの<Stop>ボタンを入力することにより、コマンド実行を停止することができます。

パラメータ

- 開始アドレス、終了アドレス及びデータには式が記述できます。
- 開始アドレス、終了アドレスに指定できる値の範囲は、0000h ~ FFFFh までです。
- データに指定できる値の範囲は、0h ~ FFh までです。なお -128 ~ 255 の符号付き整数も指定できます。

コマンド実行例

```
>F 20,5F,74<RET>                                (例1)
>
DumpByte 2000, 200F<RET>
ADDRESS: +0 +1 +2 +3 +4 +5 +6 +7  +8 +9 +A +B +C +D +E +F  ASCII
002000:  74 74 74 74 74 74 74 74  74 74 74 74 74 74 74  tttttttttttttttt
>
```

- 例1では、20h 番地から 5h 番地のアドレス範囲にデータを 74h を書き込んでいます。

G (Go)

プログラムの実行

入力書式

G [開始アドレス][,ブレークアドレス 1]

ランタイム

実行不可

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。
- 本コマンドでは、他のブレーク機能を一時的に無効にします。
- 指定したブレークアドレス命令を実行するとブレークします。

パラメータ

- 開始アドレス、ブレークアドレスには式が記述できます。
- 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。
- ブレークアドレスを省略した場合は、ブレークを行いません。(ターゲットプログラムをブレークする場合は、STOP コマンドを入力してください。)
- 開始アドレス、ブレークアドレスに記述できる値の範囲は 0000h ~ FFFFh までです。

コマンド実行例

```
>G E000,E003 (例1)
>G (例2)
>STOP
>
```

(例1) ターゲットプログラムを E000h 番地から実行を開始し、E003h 番地の命令を実行した場合にターゲットプログラムの実行を停止するようにしています。

(例2) 現在のプログラムカウンタが示すアドレスからプログラムの実行をブレークなしで実行しています。

GB (Go with program)

プログラム実行

入力書式

GB [開始アドレス]

ランタイム

実行不可

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。開始アドレスの指定を省略した場合は、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。
- 本コマンドでは、現在の設定状況に応じて、プログラムブレーク機能（別名ソフトウェアブレーク機能）のブレークが対象になります。

パラメータ

- 開始アドレスには式が記述できます。
- 開始アドレスに指定できる値の範囲は、0000h ~ FFFFh までです。

コマンド実行例

```
>PB
ADDRESS label
E006
E009
>GB E000 (例1)
>GB (例2)
>
```

(例1) E000h 番地からブレーク付きでターゲットプログラムを実行させています。

(例2) プログラムカウンタが示すアドレスからブレーク付きでターゲットプログラムを実行します。

I (Input hex and sym file)

HEX,SYM ファイルの読み込み

入力書式

I ファイル名 [.hex | .sym]]

ランタイム

実行不可

機能

- HEX ファイルと SYM ファイルを読み込みます。ファイル名 (パス名が指定可能) を指定すると、HEX ファイルを読み込みます。そして同名の SYM ファイルが同じディレクトリ上に存在する場合、SYM ファイルも読み込みます。
- ファイル名の後にファイル属性.hexまたは.symを付けることにより HEX ファイルまたは SYM ファイルを単独で読み込むことができます。
- 実行中に STOP コマンドまたは PDB38M ウィンドウの <Stop> ボタンを入力することにより、コマンド実行を停止することができます。

パラメータ

- ファイル名には HEX ファイルと SYM ファイルが指定できます。ファイル属性を省略した場合は HEX ファイルと SYM ファイルの両方を読み込みます。

コマンド実行例

```
>I test (例1)
test.hex Load end
test.hex Load end
>I test.hex (例2)
test.hex Load end
>I test.sym (例3)
test.sym Load end
```

(例1) ファイル名が test の HEX ファイルと SYM ファイルを読み込んでいます。

(例2) ファイル名が test の HEX ファイルを読み込んでいます。

(例3) ファイル名が test の SYM ファイルを読み込んでいます。

L (List)

メモリ内容の 1 行単位逆アセンブル

入力書式

L [開始アドレス][, 終了アドレス]

ランタイム

実行可能

機能

- 指定した開始アドレスから終了アドレスまでを、逆アセンブル表示します。(例 1)
- 対応する機械語が無い場合には、“???” をプログラム欄に表示します。(例 2)
- オペランド値に該当するラベル、シンボル、ビットシンボルがある場合、オペランドと共に表示します。(例 5)
- FFFFh 番地に達した場合、L コマンドを中止します。(例 6)
- オペランドのデータが FFFFh 番地を越えた場合、超えた分のデータは“???” を表示します。(例 7)
- 実行中に STOP コマンドまたは PDB38M ウィンドウの <Stop> ボタンを入力することにより、コマンド実行を停止することができます。
- BRK 命令 (1 バイト命令) を逆アセンブルする場合、オブジェクトコードを 2 バイト表示し、逆アセンブルアドレスを 2 バイト更新します。(例 9)

パラメータ

- 開始アドレスには、式及びキーワード(_PC)が記述できます。終了アドレスには、式が記述できます。
開始アドレスに “_PC” を指定した場合、現在のプログラムカウンタから表示します。(例 4)
- 開始アドレスを省略した場合は、前回の L コマンド最終処理番地のアドレスから表示します。(例 3) ただし、前回の実行で FFFFh 番地に達した場合はエラーメッセージを表示します。(例 8) なお、PDB38M 起動時はプログラムカウンタから表示します。
- 終了アドレスを省略した場合、開始アドレスから 10 行分表示します。

コマンド実行例

```

>L E000, E001<RET> (例 1)
ADDRESS> OBJ-CODE LABEL PROGRAM
E000 A904 START_AD: LDA #04H

>L E001, E002<RET>
ADDRESS> OBJ-CODE LABEL PROGRAM
E001 04 ??? (例 2)
E002 3C1020 LDM # 10H,20H

>L E006<RET> (例 3)
ADDRESS> OBJ-CODE LABEL PROGRAM
E005 8D0001 STA 0100H
>X<RET>
A=72 X=74 Y=77 F=N----I-S=0BF PC=E000 START_ADD
>L _PC, E005<RET> (例 4)
ADDRESS> OBJ-CODE LABEL PROGRAM
E000 A904 START_AD: LDA #04H
E002 3C1020 LDM #10H,20H
E005 8D0001 STA 0100H

>L E020, E03D<RET>
ADDRESS> OBJ-CODE LABEL PROGRAM
E020 A930 LDA #30H :Immediate (例 5)
E022 2280 JSR ¥FF80H :SpacialPage
E024 B704E9 BBC 5H,04H,E010H
E027 B702E6 BBC 5H,02H :ZeroPage,E010H
E02A B7040D BBC 5H,04H,E03AH :BraAddr

>L FFFE<RET> (例 6)
ADDRESS> OBJ-CODE LABEL PROGRAM
FFFE E6A9 INC A9H

>L FFFF<RET>
ADDRESS> OBJ-CODE LABEL PROGRAM
FFFF A9?? LAST_ADD: LDA #??H (例 7)
>L<RET> (例 8)
ERROR 1204: 前回の実行時に最大アドレスに達しました。
>L E080, E080<RET>
ADDRESS> OBJ-CODE LABEL PROGRAM
E080 00EA BRK (例 9)
>

```

LOGON, LOGOFF

表示画面のログファイル出力/出力停止

入力書式

LOGON [ファイル名 [ファイル属性]]

LOGOFF

ランタイム

実行可能

機能

- LOGONコマンドは、表示画面の内容をログファイルへの書き込みます。
- LOGONコマンドは、複数のファイルのオープンが可能です。
なお、表示画面のログファイルへの書き込みは、一番最後にオープンしたログファイルに書き込まれます。
- LOGOFFコマンドは、表示画面のログファイルへの書き込みを中止します。
複数のログファイルをオープンしているときにLOGOFFコマンドを実行した場合、1番最後にLOGONコマンドでオープンしたログファイルをクローズします。
- ログファイルの最大ネスティングレベルは8です。

パラメータ

- ファイル名には、ログファイル名が記述できます。ファイル属性は、任意の名前をつけることができます。ファイル属性を省略した場合は、ファイル名にファイル属性 .log が付加されます。
ファイル名を省略した場合は、直前にLOGOFFしたファイルをオープンします。その時は、オープンしたファイルの最後尾から書き込みを行います。オープンするファイルがない場合は、LOGONを実行した時刻がログファイル名となります。
例えば、LOGONを実行した時刻が、12月16日15時03分であればログファイル名は、12161503.log となります。

コマンド実行例

>LOGON

ファイル名指定なし (12071333.log がオープンされる)

```
>LOGON d.log          ファイル名指定あり (d.log をオープン)
>D E000,E100          (d.log に出力される)
:
:
>LOGON 1.log          ファイル名指定あり (1.log をオープン)
>D E000,E100          (d.log に出力される)
:
:
>LOGOFF              1.log をクローズ
>LOGOFF              d.log をクローズ
>LOGON               ファイル名指定なし (d.log がオープンされる)
>D E300,E400         (d.log 最後尾から出力される)
:
:
>LOGON               ファイル名指定なし (1.log がオープンされる)
>D E300,E400         (1.log 最後尾から出力される)
:
:
>LOGOFF              1.log をクローズ
>LOGOFF              d.log をクローズ
>LOGOFF              12071333.log をクローズ
>
```

O (Output HEX file)

HEX ファイルの出力

入力書式

O ファイル名, 開始アドレス, 終了アドレス

ランタイム

実行不可

機能

- 指定したアドレス範囲のデータをインテルHEXファイルとして指定したファイル名で格納します。
- 実行中に STOP コマンドまたはPDB38Mウィンドウ<Stop>ボタンを入力することにより、コマンド実行を停止することができます。

パラメータ

- ファイル名には出力ファイル名が記述できます。ファイル属性を省略した場合は、.hex が指定されたと見なします。
- 開始アドレス、終了アドレスには式が記述できます。
- 開始アドレス、終了アドレスに指定できる値の範囲は 0000h から FFFFh までです。

コマンド実行例

```
>O TEST.HEX, E000,EFFF  
>
```

(例 1)

PB (Program Break)

プログラムブレークポイントの設定/参照

入力書式

PB

PB {SET | CLEAR} アドレス

ランタイム

実行不可

機能

- プログラムブレーク機能（別名ソフトウェアブレーク機能）のブレークポイントを設定/解除/参照します。
- プログラムブレーク機能はソフトウェアブレーク機能とも呼ばれ、プログラム領域に実行前ブレークが設定できるブレーク機能です。
指定したアドレスの命令を実行する前にブレークします。
- ブレークポイントを設定する場合は、PB SET アドレスと入力してください。解除する場合は、PB CLEAR アドレスと入力してください。参照する場合は、PB と入力してください。

パラメータ

- アドレスには式が記述できます。
- アドレスの指定できる値の範囲は 0000h から FFFFh までです。

コマンド実行例

```
>PB SET E000 (例 1)
```

```
>PB (例 2)
```

```
ADDRESS LABEL  
E000 START:
```

```
>PB CLEAR E000 (例 3)
```

```
>PB
```

```
ADDRESS LABEL
```

(例1) E000h 番地にブレークポイントを設定しています。

(例2) 現在設定されているブレークポイントを参照しています。

(例3) E000h 番地のブレークポイントを解除しています。

RADIX

数値入力の既定値の設定/参照

入力書式

```
RADIX [{ 10 | 16 }]
```

ランタイム

実行可能

機能

- 式における定数入力のデフォルトの基数を設定します。10 進、16 進の何れかが設定できます。デフォルトは 16 進です。
- 本コマンドで基数を設定した場合でも、定数入力の際に進数を示す '@'、'\$' 等を付加することによって、定数の進数を個別に指定できます。
- 本コマンドで設定した基数は、スクリプトコマンドでのみ有効です。

パラメータ

- 基数の既定値には、10, 16 の何れかが指定できます。
- パラメータを省略した場合は、現在設定されている基数の既定値を表示します。

コマンド実行例

```
>RADIX 16 (例 1)
>RADIX (例 2)
RADIX = 16
>RADIX 10 (例 3)
>RADIX
RADIX = 10
>
```

- (例 1) 基数の既定値を 16 進数に設定しています。
- (例 2) 現在設定されている基数の既定値を参照しています。
- (例 3) 基数の既定値を 10 進数に設定しています。

S (SET)

メモリ内容の参照/変更

入力書式

S [アドレス]	(16 進表示)
SD [アドレス]	(10 進表示)
SB [アドレス]	(2 進表示)

ランタイム

実行可能

機能

- 指定したアドレスの内容をバイト単位で表示、変更を行います。表示は、2進、10進、16進の何れで表示するかを選択できます。
- 前方のアドレスを参照する場合は、“-<RET>”（マイナス）を入力してください。
- コマンドの終了は、“.<RET>”（ピリオド）を入力してください。
- データの変更を行わない場合は、“<RET>”キーのみを入力してください。
- 開始アドレスを省略した場合は、直前のSコマンドで参照した最終処理番地から表示します。（初めて実行する場合はE000h番地から表示します。）

パラメータ

- アドレスには式が記述できます。省略した場合は、直前のSコマンドで参照した最終処理番地から表示します。
（初めて実行する場合はE000h番地から表示します。）
- 変更するデータには式が記述できます。
- アドレスに指定する値の範囲は、0000hからFFFFhまでです。変更するデータの範囲は0からFFhまでです。なお-128から255の符号付き整数も指定できます。

コマンド実行例

```
>S e000<RET>
ADDRESS LABEL OLD NEW
E000 start: EA 50<RET>
```

```
E001          EA      30+20<RET>
E002          EA      -<RET>          前の番地に戻る
E001          50      20<RET>
E002          EA      .<RET>          終了
>S<RET>
ADDRESS LABEL  OLD  NEW
E002          EA      00
E003          EA      .<RET>          終了
>SD F000
ADDRESS LABEL  OLD  NEW
E000          210   2D
E001          0     3E
E002          EA   .<RET>          終了
>SB 0000
ADDRESS LABEL  OLD  NEW
0000          00000000 5F
0001          00000000 -
0000          01011111 .<RET>          終了
>
```

SCOPE

ローカルシンボル有効範囲の設定/参照

入力書式

SCOPE [リロケータブルファイル名]

ランタイム

実行可能

機能

- スコープの設定を行います。
スコープとは、ローカルラベル/シンボルの有効範囲のことです。
スコープの単位は、オブジェクトファイル (R74 ファイル) です。
- リロケータブルファイル名を省略した場合、現在のスコープを表示します。

パラメータ

- モジュールファイル名にはファイル属性.r38を持つファイル名が記述できます。
ファイル属性は必ず指定してください。

コマンド実行例

```
>SCOPE test.r74 (例 1)  
>
```

(例 1) test.r74 ファイルをスコープの範囲に指定しています。

注意事項

スコープ (SCOPE) 処理

PDB38Mは、デフォルトではプログラムカウンタが示すスコープに設定します。また各種コマンド実行中は、コマンドが処理するアドレスに応じて、スコープを自動的に切り替え、コマンド実行終了時にカレントのスコープに戻します。

本コマンドは、スコープを一時的に変更したい場合にお使いください。

SCRIPT

スクリプトファイルのオープン/実行

入力書式

SCRIPT スクリプトファイル名

ランタイム

実行可能

機能

- スクリプトファイルをオープンします。
スクリプトファイルは、スクリプトコマンドを記述したテキストファイルです。スクリプトファイルは、お手持ちのエディタで作成してください。
- 本コマンドでスクリプトファイルをオープンした後、スクリプトウィンドウの Run ボタン等を入力することにより、スクリプトファイルに記述したスクリプトコマンドを自動実行できます。
- 本コマンドは、スクリプトファイル内にも記述できます。スクリプトファイルのネスティングレベルは5です。

パラメータ

- スクリプトファイル名には、任意のスクリプトファイル名が使用できます。ファイル属性は必ず指定してください。

コマンド実行例

```
>SCRIPT test.scr
```

(例1)

(例1) test.scr ファイルをスクリプトファイルとしてオープンします。

SI (Section Information)

セクション情報の表示

入力書式

SI

ランタイム

実行可能

機能

- セクション情報を表示します。
- セクション情報は、左から順に
NAME,OBJECT,TYPE,START,LENGTH,SOURCE,LIBRARY の内容を表示
します。
 - セクション名 (項目名 : NAME)
 - オブジェクトファイル名 (.r74) (項目名 : OBJECT)
 - 物理属性 (ROM/RAM) (項目名 : TYPE)
 - 先頭アドレス (項目名 : START)
 - セクション領域のバイト数 (項目名 : LENGTH)
 - ソースファイル名 (.a74) (項目名 : SOURCE)
 - ライブラリファイル名 (項目名 : LIBRARY)

パラメータ

- なし

コマンド実行例

```
>SI<RET>
NAME      OBJECT      TYPE      START  LENGTH  SOURCE      LIBRARY
RAMAREA   sample0.r74  RAM       0000   0100   sample0.a74
PROG1     sample1.r74  ROM       E000   0074   sample1.a74
PROG1     sample2.r74  ROM       E074   0077   sample2.aA4
PROG2     sample1.r74  ROM       F000   0072   sample1.a74
>
```

SRC (set SouRCe file)

ソースファイル名によるプログラムウィンドウの表示変更

入力書式

SRC ソースファイル名

ランタイム

実行可能

機能

- プログラムウィンドウに表示するソースファイル名を変更します。

パラメータ

- パラメータには、ソースファイル名(.a74)を設定して下さい。

コマンド実行例

```
>SRC sample.a74 (例1)  
>
```

(例1) プログラムウィンドウに表示するソースファイル名を sample.a74 に変更しています。

STOP

プログラムの実行停止

入力書式

STOP

ランタイム

実行可能

機能

- ターゲットプログラムの実行を強制停止します。

パラメータ

なし

コマンド実行例

```
>G E000  
>STOP  
>
```

T (Trace)

ステップ実行

入力書式

T [ステップ回数]

ランタイム

実行不可

機能

- 現在のプログラムカウンタから指定したステップ数だけ命令を実行します。
- ステップ数を省略した場合は、1ステップ実行します。
- T コマンドの最終行には ' * ' の後に、プログラムカウンタの値が表示されます。

パラメータ

- ステップ回数には式が記述できます。
- ステップ回数に指定できる値の範囲は 1 から 65535 までです。

コマンド実行例

```

>T 3                                     (例1)
A=04 X=7F Y=EB F=-----I-- S=07F PC=E000 START
A=04 X=7F Y=EB F=----B-I-- S=07F PC=E001
A=04 X=7F Y=EB F=----B-I-- S=07F PC=E002
*E003
>

```

(例1) ステップ回数を 3 回に指定してステップ実行をしています。(T コマンド実行後のプログラムカウンタの値は E003h 番地です。)

U (Untrace)

ステップ実行

入力書式

U [ステップ回数]

ランタイム

実行不可

機能

- シングルステップを開始する前のステータスを表示し、現在のプログラムカウンタから指定したステップ数を実行します。
- U コマンドの最終行には ' * ' の後に、プログラムカウンタの値が表示されます。

パラメータ

- ステップ回数には式が記述できます。
- ステップ回数に指定できる値の範囲は 1 から 65535 までです。
- ステップ回数を省略したときのデフォルトの値は 1 回です。

コマンド実行例

```
>U 3 (例1)
A=04 X=7F Y=EB F=-----I-- S=07F PC=E000 START
*E003
>
```

(例1) ステップ回数を 3 回に指定してステップ実行をしています。(U コマンド実行後のプログラムカウンタの値は E003h 番地です。)

VER (VERsion)

バージョンの表示

入力書式

VER

ランタイム

実行可能

機能

- PDB38Mのバージョン番号を表示します。

パラメータ

なし

コマンド実行例

```
>VER<RET>
PDB38M Ver.V.1.00.00
>
```

WAIT (WAIT commands till break)

プログラムのブレークまでコマンド入力待機

入力書式

WAIT

ランタイム

実行可能

機能

- ターゲットプログラムがブレークするまでコマンド入力を待機します。
- 本コマンドは、スクリプトファイルからターゲットプログラムをブレークポイント付きで実行し、ブレーク後に続けてコマンドを実行したいときに使用します。
- ターゲットプログラムが停止している場合は、本コマンドは無視されます。
- 実行中に STOP コマンドまたはPDB38Mウィンドウの<Stop>ボタンを入力することにより、コマンドの実行を停止することができます。

パラメータ

- なし

コマンド実行例

```
>WAIT<RET>  
>
```

X (eXeamine register)

レジスタ値の参照/変更

入力書式

- 書式 1 : X
書式 2 : XA 設定値
 : XX 設定値
 : XY 設定値
 : XS 設定値
 : XF 設定値
 : XP 設定値
 : XA 設定値
書式 3 : X レジスタ名

(レジスタ名: {A | X | Y | S | F | P})

ランタイム

実行不可

機能

- レジスタの内容を参照又は変更します。
- 書式 1 では、コマンド名 X のみ入力します。この場合、全レジスタの内容を参照します。
- 書式 2 では、コマンド名として X とレジスタ名を続けて入力します (例:XA)。そして第一パラメータにはレジスタへの設定値を入力します。
- 書式 3 では、コマンド名として X、第一パラメータにはレジスタ名を入力します。(コマンド名とレジスタ名は空白文字で区切ります)。この場合、その後、レジスタへの設定値の入力待ちとなります (対話入力)

パラメータ

- レジスタ名には A(A レジスタ)、X(X レジスタ)、Y(Y レジスタ)、F(F プロセッサステータスレジスタ)、S(S スタックポイント)、P(P プログラムカウンタ)の何れかが記述できます。
- 設定値には、式が記述できます。

- 設定値に指定する値の範囲は設定するレジスタにより異なります。
以下に設定値の範囲を示します。

レジスタ	最小値	最大値
A,X,Y,F	0h	FFh
S	000h	1FFh
P	0000h	FFFFh

なおレジスタ A,X,Y,F では、-128 から 255 の符号付き整数も指定できます。

コマンド実行例

```

>X A                                (例 1)
04 72
>X X                                (例 2)
7F 74
>XY FF                              (例 3)
>X                                  (例 4)
A=72      X=74      Y=FF      F=---B-I--      S=07F      PC=E006
>

```

- (例 1) A レジスタの内容を 04h から 72h に変更しています。
- (例 2) X レジスタの内容を 7Fh から 74h に変更しています。
- (例 3) Y レジスタの内容を FFh に変更しています。
- (例 4) 全てのレジスタの内容を参照しています。

Z (reset)

MCU のハードウェアリセット

入力書式

Z

ランタイム

実行不可

機能

- ターゲット MCU をリセットします。
- プログラムカウンタの値は、リセットベクトルが示すアドレスに変更されます。

パラメータ

なし

コマンド実行例

```
>Z  
>
```

; (comment line)

コメント行

入力書式

:[文字列]

ランタイム

実行可能

機能

- スクリプトファイル用のコメントです。PDB38Mはコメントを無視します。

パラメータ

- 文字列に記述できる文字の制限はありません。
- 文字列に記述できる文字数は、コマンド入力行の最大サイズ(160 カラム)に依存します。

コマンド実行例

```
>; データテーブル TABLE1 の表示<RET>  
>D TABLE1, TABLE1+8<RET>  
>
```

15. 式の記述方法

PDB38Mでは、各種コマンド入力に式が使用できます。図 15-1 に式を使用したコマンド入力を示します。

```
>D TABLE1  
>D TABLE1+20  
>F TABLE1, LABEL1, 'C'  
>F TABLE, LABEL1, 10  
>PB SET #10.TEST.A74  
>D TABLE1+20-#10*20-5
```

図 15-1 式を使用したコマンド入力例図

式の記述方法について説明します。

15.1.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。

1 . ラベル

アセンブラSRA74Mの行頭で定義したラベルが使用できます。ラベルは SYM ファイルを読み込んだ後に使用できます。またラインアセンブルコマンド(A コマンド)で定義したラベルも使用できます。

2 . シンボル

アセンブラSRA74Mの擬似命令.EQU で定義したシンボルが使用できます。シンボルはSYM ファイルを読み込んだ後に使用できます。またラインアセンブルコマンド(A コマンド)の擬似命令.EQU で定義したシンボルも使用できます。

3 . 定数

2進 10進 16進の定数が使用できます。

4 . 演算子

加減乗除、ビット演算子等の演算子が使用できます。

5 . 行番号

プログラム領域(実行可能な命令コードを含む領域)の行番号が使用できます。

以下では、各構成要素について説明します。

15.2. ラベルおよびシンボル

15.2.1. ラベルおよびシンボルの記述方法

ラベル及びシンボル名をそのまま記述します。

- 英数字、アンダーライン(_)、ピリオド(.)、クエッションマーク(?)が使用できます。ただし先頭文字に、数字は使用できません。
- 大文字/小文字を区別します。
- レジスタ名(A,X,Y,S,PC,PS,P)は使用できません。
- 文字数の制限は、アセンブラSRA74Mの仕様に依存します。(255 文字)
- アセンブラSRA74Mの構造化命令、擬似命令、マクロ命令、オペコードは使用できません。(.SECTION, .BYTE, .switch, if 等)
- 以下示すものはラベル、シンボルには使用できません。

.D0	~	.D65535
.F0	~	.F65535
.I0	~	.I65535
.S0	~	.S65535
..0	~	..65535
??0	~	??65535

コマンド名、およびコマンドの機能概要を示します。

15.2.2. ローカルラベル/シンボルとスコープ

PDB38Mでは、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(.R74 ファイル)です。PDB38Mでは、下記の場合に応じて、スコープを切り替えます。

1 . 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。

2. コマンド実行時

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。

15.2.3. 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。

1. アドレス値を変換する場合

- (a) ローカルラベル
- (b) グローバルラベル
- (c) ローカルシンボル
- (d) グローバルシンボル
- (e) スコープ範囲外のローカルラベル
- (f) スコープ範囲外のローカルシンボル

2. データ値を変換する場合

- (a) ローカルシンボル
- (b) グローバルシンボル
- (c) ローカルラベル
- (d) グローバルラベル
- (e) スコープ範囲外のローカルラベル
- (f) スコープ範囲外のローカルシンボル

3. ビット値を変換する場合

- (a) ローカルビットシンボル
- (b) グローバルビットシンボル
- (c) スコープ範囲外のローカルビットシンボル

15.3.定数

2 進、10 進、16 進が入力可能です。数値の基数は、数値の先頭または末尾に表に示す記号を付加して区別します。

ラベル及びシンボル名をそのまま記述します。

表 15-1 定数の入力一覧

進数	即定値が 10 進数の場合の表記	即定値が 16 進数の場合の表記
2 進数	%10010 または 10010B	%10010
10 進数	1234	@1234
16 進数	\$AB24 または 0AB24H	0AB24,AB24,\$AB24 または 0AB24H

記号を省略した場合、16 進数として処理します。ただしスクリプトコマンドでの式入力では、RADIX コマンドで設定した即定値 (10 進数または 16 進数) として処理します。なお 16 進数の先頭が “A” ~ “F” で始まり、ラベルやシンボル名と同じならばラベルまたはシンボルとして扱います。

15.4.演算子

式に必要な可能な演算子の一覧を表および表に示します。

表 15-2 コマンド一覧(実行と停止)

演算子	意味	優先度レベル
(左括弧	レベル 1
)	右括弧	
単項演算子		レベル 2
*	乗算	レベル 3
/	除算	
	加算	レベル 4
--	減算	
>>	右	レベル 5
<<	左	
&	ビットごとの AND	レベル 6
^	ビットごとの EOR	レベル 7
	ビットごとの OR	レベル 8

表 15-3 単項演算子一覧

演算子	意味	優先度レベル
*	正	レベル 2
--	負	
~	ビットごとの NOT	
!	論理 NOT	

- 演算の優先度は、レベル 1 が最も高く、レベル 8 が最も低くなります。
- 演算式の評価は、優先度の高い順に行います。

15.5.行番号

行番号の入力形式を図 15-2 に示します。

```
#行番号  
#行番号.”ファイル名.ファイル属性”
```

図 15-2 行番号の入力形式

- 行番号には、10 進数が使用できます。
- ソースファイル行番号は、省略することができます。省略した場合、現在、プログラムウィンドウに表示しているソース行番号の行番号と見なします。
- 行番号、ファイル名、記号の間には、空白文字は挿入できません。

トラブルシューティング編

16. エラーメッセージ一覧

以下に、エラーメッセージの一覧を示します。

表 16-1 エラーメッセージ (No.100 ~)

番号	エラーメッセージ	補足・対応
100	Can't find Source File (%s).	
101	Unable to read Load Module File (%s).	
102	INTERNAL ERROR : File size information is illegal.	
103	Line number of Source File (%s) is over 65000.	
104	INTERNAL ERROR:Widget ID is illegal.	
105	INTERNAL ERROR : Can't create Bit Map.	
106	INTERNAL ERROR : Window ID is illegal.	
107	INTERNAL ERROR : Memory overflow.	
108	Name of Save File (%s) is illegal.	
109	Environment value 'MXDB38_HOME' isn't defined.	
110	Environment File is illegal.	
111	Can't write Environment File.	
112	Environment value 'MXDB38_HELP' isn't defined.	
113	Environment value 'MXDB38_MCU' isn't defined.	
114	Environment value 'MXDB38_LANG' isn't defined.	
115	Can't open more source window.	
116	Necessary environment definition isn't done.	
117	Can't execute new command during command processing.	

表 16-2 エラーメッセージ (No.150 ~)

番号	エラーメッセージ	補足・対応
150	Can't open more %s window.	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	Can't Create %s window.	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	Can't open %s window, when the target program is running.	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	Value is out of range.	指定したアドレスがMCUの最大アドレスFFFFFFhを越えています。

表 16-3 エラーメッセージ (No.200 ~)

番号	エラーメッセージ	補足・対応
200	Can't change view mode.	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	Can't find source file (%s).	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	Can't find search string (%s).	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	Line number of Source File (%s) is over %d.	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。

表 16-4 エラーメッセージ (No.300~)

番号	エラーメッセージ	補足・対応
300	Illegal endi. (%s %d line)	endi に対応する if がありません。
301	Illegal endw. (%s %d line)	endw に対応する while がありません。
302	INTERNAL ERROR : ER_BAT_EOF	
303	Script File is already exist.	
304	Can't find endi. (%s %d line)	if に対応する endi がありません。
305	Line length is overflow. (%s %d line)	一行に記述できる文字数の制限を超えました。
306	Nest level is overflow. (%s %d line)	
307	Can't find Script File (%s).	
308	Can't read Script File (%s).	スクリプトファイルの続きが読み込めません。
309	Description is illegal. (%s %d line)	
310	Can't find endw. (%s %d line)	while に対応する endw がありません。
311	Nest level is overflow. (%d line)	
312	INTERNAL ERROR : ER_BAT_NONE	

表 16-5 エラーメッセージ (No.400~)

番号	エラーメッセージ	補足・対応
400	Address value is out range for scroll area.	

表 16-6 エラーメッセージ (No.600~)

番号	エラーメッセージ	補足・対応
600	Can't add new watch point because it exceeds limit of watch point number. Maximum number is (%d).	
601	Address value is out of range.	
602	Data value is out of range.	
603	Bit value is out of range	

表 16-7 エラーメッセージ (No.650 ~)

番号	エラーメッセージ	補足・対応
650	There are no symbol information.	ロードモジュールファイルがロードされていません。
651	The expression is too long.	

表 16-8 エラーメッセージ (No.700 ~)

番号	エラーメッセージ	補足・対応
700	Can't find file (%s)	
701	Message File (%s) is broken.	
702	Can't get enough memory.	

表 16-9 エラーメッセージ (No.900 ~)

番号	エラーメッセージ	補足・対応
900	SYMBOL file is illegal.	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	Loading is canceled.	
902	Can't find SYMBOL file(%s).	ロードモジュールファイルが存在しません。

表 16-10 エラーメッセージ (No.1001 ~ (その1))

番号	エラーメッセージ	補足・対応
1001	Can't find symbol.	指定したシンボルは、存在しません。
1002	Description of expression is illegal.	
1004	Description is illegal.	式の記述に誤りがあります。
1005	Can't find scope.	指定された変数は、スコープ内にありません。
1006	Can't find symbol.	
1007	Can't find function.	指定された関数はありません。
1008	Right hand side of the expression is illegal.	
1009	The Type of structure(union) are not same.	
1010	Can't assign.	
1011	Can't find type.	指定された型はありません。
1012	Not supported float(double) operation.	
1013	The operation does not be allowed to pointers.	
1014	The operation does not be allowed to the pointer.	
1015	Can't decrease by pointer.	

表 16-11 エラーメッセージ (No.1001 ~ (その2))

番号	エラーメッセージ	補足・対応
1016	Divided by 0.	
1017	The operator is not supported.	
1018	Type information is broken.	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	Left value must be the pointer.	
1020	Left value must be a structure or an union.	
1021	Can't find member.	
1022	Left value must be reference of a structure or an union.	
1023	Left value is illegal.	
1024	The operand must be a value.	
1025	The operand is able to be opposite sign.	
1026	Can't get address value.	
1027	The array variable is illegal.	
1028	The essential number of array is illegal.	
1029	The operand must be an address value.	
1030	Type casting for register variable is not be supported.	
1031	The type of type casting is illegal.	
1032	Type casting for that type is not be supported.	

表 16-12 エラーメッセージ (No.1070 ~)

番号	エラーメッセージ	補足・対応
1078	Can't open more memory window.	
1080	The address area is illegal.	

表 16-13 エラーメッセージ (No.1100~)

番号	エラーメッセージ	補足・対応
1100	Address value is out of range.	指定したアドレスがMCUの扱える最大値 FFFFh を越えています。
1101	Description of Assembly language is illegal.	
1102	Address value for JUMP is out of range.	
1103	Operand value is out of range.	
1104	Description of expression is illegal.	

表 16-14 エラーメッセージ (No.1200~)

番号	エラーメッセージ	補足・対応
1200	Start address is larger than end address.	
1201	Address value is out of range.	
1202	Data value is out of range.	
1203	Parameter (%d) is illegal.	
1204	Reached to maximum address in the previous execution.	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	Can't find File (%s).	
1206	File (%s) is broken.	
1207	File Name is illegal.	
1208	Can't execution because it exceeds limit nest level of log file. Maximum level is (%d)."	
1209	File (%s) is already log on.	
1210	Can't open File (%s).	
1211	Can't get enough memory for LOGON command.	
1212	Can't execute because it exceeds limit nest level of script file. Maximum level is (%d).	指定したファイルを保存することができません。
1213	Can't write File (%s) because there is not enough disk space.	8ビットバス幅で、16ビットアクセスはできません。
1214	Can't get enough memory.	M38000T-SBIに対して、M38000T-SBI用のコマンドは実行できません。
1215	Address data is wrong.	
1216	Register value is out of range.	
1217	Can't execute this command because target type is unexpected.	
1218	Start cycle is larger than end cycle.	
1219	Can't open source file, or can't find source line.	

表 16-15 エラーメッセージ (No.1250~)

番号	エラーメッセージ	補足・対応
1250	The number of parameter is too many.	
1251	Unknown command.	
1252	Parameter (%d) value is out of range.	
1253	Parameter (%d) is illegal.	
1254	Input command line is too long.	
1255	Can't execute that command, when the target program is running.	
1256	Description of expression is illegal.	
1257	Target program is already stopped.	

表 16-16 エラーメッセージ (No.1300~)

番号	エラーメッセージ	補足・対応
1300	Line number is illegal.	
1301	Can't find right bracket '}'.	
1302	The Number of Macro constant is over the limit (%d).	
1303	Immediate value is out of range.	
1304	Prefix which gives radix of the constant is illegal.	
1305	Description of indirect reference is illegal.	
1306	Can't find end of strings (%s).	
1307	Description of expression is illegal.	
1308	Macro constant (%s) isn't defined.	
1309	Symbol (%s) isn't defined.	
1310	Immediate value is illegal.	
1311	Divide by 0.	
1313	The value is over the maximum value of which can be treated by MCU.	

表 16-17 エラーメッセージ (No.1400 ~ (その1))

番号	エラーメッセージ	補足・対応
1400	Address value is out of range.	指定したアドレスがMCUの扱える最大値 FFFFFh を越えています。
1401	Target program is already stopped.	
1402	The number of break point is over the limit (%d).	
1403	The break point isn't defined at that address.	
1404	Data value is out of range.	
1406	Can't read/write, because there are no memory at that area.	メモリの存在しないアドレスに対する参照書き込みは行なえません。
1407	Can't get enough memory.	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。
1408	Register value is out of range.	
1409	Can't execute that command, when the target program is running.	
1410	Start address is larger than end address.	
1411	STOP execution.	
1412	Can't find source lines which include that address.	指定したアドレスには、ソース行情報がありません。
1413	That command has not yet supported.	
1450	Address value is out of range.	
1451	That baud rate has not yet supported.	
1452	Bit number is out of range.	
1453	STOP execution.	
1454	Data value is out of range.	
1455	Monitor File (%s) is broken.	モニタファイル filename を再インストールして下さい。
1456	Can't find File (%s).	モニタファイルは、PDB38M.EXE と同じディレクトリに格納してください。
1457	Target system is not constructed properly.	PDB38M, M38000T-SBIと POD の組合せが正しくありません。
1458	INTERNAL ERROR:ER_IN2_ILLEGAL_MODE has happen. (in %s)	
1459	Mask value is out of range.	

表 16-18 エラーメッセージ (No.1400 ~ (その2))

番号	エラーメッセージ	補足・対応
1460	Counter of measurement time is overflow.	
1461	The version of PDB38 and the firmware on the target are not same.	PDB38Mより新しいファームウェアが搭載されています。PDB38Mに対応したファームウェアをダウンロードしてください。
1462	Pass count value is out of range.	
1463	Can't execute that command, when the target program is running.	
1464	Target MCU is reset state. Please reset target systems.	ターゲットシステムをリセットして下さい。
1465	Target MCU is unable to reset. Please reset target systems.	ターゲットシステムをリセットして下さい。
1466	Target MCU is HOLD state. Please reset target systems.	
1467	Target MCU is not given clock. Please reset target systems.	
1468	Target MCU is not given power. Please reset target systems.	ターゲットシステムをリセットして下さい。
1469	INTERNAL ERROR:Break point number is illegal.	
1470	Please download the firmware to target.	自動的にファームウェアをバージョンアップします。
1471	Can't download firmware.	M38000T-SBIをメインテナンスモードで起動した後に、PDB38Mを起動してください。自動的にファームウェアをバージョンアップします。
1472	Download firmware is finished. Please restart PDB38M.	PDB38Mを再起動してください。
1473	Can't find trace data which is able to refer.	
1474	Cycle value is out of range.	
1475	Target MCU is not under control. Please reset target systems.	ターゲットシステムをリセットして下さい。
1476	First data is larger than second data.	
1477	First address is larger than second address.	
1478	No event set on the state transition path.	
1479	Time out value is out of range.	
1480	Process ID value is out of range.	
1481	Cannot execute PDB38M because Emulator's version is old. Please version up Emulator.	
1482	The MCU file does not match POD. Please check MCU file name.	

表 16-19 エラーメッセージ (No.1500 ~)

番号	エラーメッセージ	補足・対応
1500	Address value is out of range.	指定したアドレスがMCUの扱える最大値 FFFFh を越えています。
1501	Bit number is out of range.	
1502	File (%s) is broken.	
1503	Can't find File (%s).	
1504	Can't find sub routine information.	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。
1505	Illegal character in the strings.	
1506	INTERNAL ERROR:ER_IN2_ILLEGAL_MODE has happen. (in %s)	
1507	Can't find that line number.	
1508	Multiple definition of symbol/label.	
1509	There are no code at that line.	指定した行番号に対応するアドレスには機械語が生成されていません。
1510	Can't get enough memory.	
1511	Can't find scopes.	
1512	Can't find section information.	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。
1513	Can't find source lines which correspond to that address.	
1514	Can't find symbol (%s).	
1515	Can't find the scopes which include that address.	
1516	Loading is canceled.	

表 16-20 エラーメッセージ (No.1700~)

番号	エラーメッセージ	補足・対応
1700	INTERNAL ERROR:Already connected with the target.	
1701	INTERNAL ERROR:Fork error has happen.	
1702	Can't find Host Name (%s).	
1703	INTERNAL ERROR:The Baud rate is illegal.	
1704	The connection with the target isn't created.	
1705	Can't connect with the target.	
1706	INTERNAL ERROR:The Time of time out is out of range.	
1707	Time Out ERROR.	ターゲットシステムとの通信中にタイムアウトエラーが発生しました。
1708	INTERNAL ERROR:Can't disconnect with the target.	
1709	INTERNAL ERROR:Can't send given size data.	
1710	INTERNAL ERROR: Parameter is illegal.	
1711	Illegal Host Name.	
1712	Communication ERROR. The connection with the target is closed.	ターゲットシステムとの通信中にターゲットとの接続が切断されました。
1713	Communication ERROR. Can't send data.	ターゲットシステムへのデータ転送中に通信エラーが発生しました。
1714	Communication ERROR. Can't accept data.	ターゲットシステムからのデータ受信中に通信エラーが発生しました。
1715	Target is already used.	

表 16-21 エラーメッセージ (No.2400~)

番号	エラーメッセージ	補足・対応
2400	Address value is out of range.	
2401	Data value is out of range.	
2402	Start Address value is out of range.	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	Value is under (%d).	num 以下の値を指定してください。
2404	Data value is out of range.	

表 16-22 エラーメッセージ (No.5200~)

番号	エラーメッセージ	補足・対応
5200	Can't execute Come command, without unused break point.	何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。

表 16-23 エラーメッセージ (No.5500~)

番号	エラーメッセージ	補足・対応
5500	Value is out of range.	

表 16-24 エラーメッセージ (No.5700~)

番号	エラーメッセージ	補足・対応
5700	The data value is too large.	
5701	The address area is illegal.	
5702	Address value is out range for scroll area.	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 FFFFFFFh を越えています。

表 16-25 エラーメッセージ (No.5800~)

番号	エラーメッセージ	補足・対応
5800	Sampling period value is out of range.	
5801	Address value is out of range.	
5802	Can't change RAM monitor area, when the target program is running.	ターゲットプログラムを停止してから、RAM モニタ領域を変更してください。

表 16-26 エラーメッセージ (No.5900~)

番号	エラーメッセージ	補足・対応
5900	Can't open Script File (%s).	
5901	Script File is not open.	
5902	Can't open Log File (%s).	
5903	Can't open more Log File.	
5904	Can't open Log File.	
5905	File (%s) is already log on.	
5906	Can't open View File (%s) for new/add.	

表 16-27 エラーメッセージ (No.10045~)

番号	エラーメッセージ	補足・対応
10045	INTERNAL ERROR:ER_ENV_END	
10046	Can't find directory (%s).	
10047	Can't create environment definition directory.	
10048	Can't open environment definition file.	
10049	Can't find environment definition directory.	
10050	Can't find MCU File (%s).	
10051	Can't find Watch File (%s).	
10052	Can't find Script File (%s).	

表 16-28 エラーメッセージ (No.10055~)

番号	エラーメッセージ	補足・対応
10055	Can't open View File (%s) for new/add.	
10056	Can't write View/Log File because there is no enough disk space.	
10057	Can't find Script File (%s).	
10058	Can't execute because it exceeds limit nest level of log file. Maximum level is (%d).	
10059	File (%s) is already log on.	
10060	Can't open File (%s).	
10061	Can't get enough memory.	
10062	Can't write File (%s) because there is no enough disk space.	
10063	Script window is already opened.	
10064	Can't open more dump window.	
10066	There is no data.	
10068	There is no address.	
10069	There is no start address.	
10070	There is no end address.	
10071	Write area is illegal.	
10072	There is no Move address.	
10073	Unknown address size.	
10074	The area of display isn't mapped.	
10075	Can't open window because Map definition denied.	

索引

索引

— P —

PDB38M ウィンドウ 15

— あ —

アップロード 10, 22

アドレス式 85

— い —

インテル HEX フォーマットファイル 8, 10, 21, 23

— う —

ウォッチウィンドウ 85

ウォッチポイント 85

— お —

オーバーステップ実行 36

— か —

カム実行 34

環境設定ファイル 9

— き —

逆アセンブルファイル 10, 24

— し —

シリアルインタフェース 6

シリアル通信 6

シンボル 85

シンボルファイル 8, 21

— す —

スクリプトウィンドウ 97, 117

スクリプトコマンド 8, 97, 117, 143

スクリプトファイル 8, 97, 102, 143

スコープ 142

ステップ実行 35

— そ —

ソースウィンドウ 62

ソフトウェアブレークポイント 37, 47, 109

— た —

ダウンロード 20

ダンプウィンドウ 76

— は —

ハードウェアブレークポイント 37

— ひ —

ビットシンボル 85

ビューバッファ 104

ビューファイル 11, 104

— ふ —

プログラムウィンドウ 44

— め —

メモリウィンドウ 66

— り —

リターン実行 37

— れ —

レジスタウィンドウ 63

— ろ —

ローカルシンボル 142

ローカルラベル 142

ローカルラベルシンボル 142
ログファイル 11, 97, 105, 135

—漢字—

定義体ファイル 8

M3xxxxT-PAC ユーザーズマニュアル《コントロールソフトウェア PDB38M 編》

第 1 版:1996 年 11 月 16 日発行

第 2 版:2001 年 7 月 26 日発行

資料番号:MSD-740PAC-UC-010726

Copyright © 1996, 2001 三菱電機株式会社

Copyright © 1996, 2001 三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社

三菱電機株式会社

三菱電機セミコンダクタ・アプリケーション・エンジニアリング株式会社

M3XXXXT-PAC 《コントロールソフトウェア PDB38M 編》
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