お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがな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て、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が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。

User's Manual

H8/388R E6000 エミュレータ

ユーザーズマニュアル

H8/388R E6000　　HS388REPI60HJ-U2

ルネサスマイクロコンピュータ開発環境システム

H8 ファミリ／ H8/300L Super Low Power シリーズ

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00　2004.11

安全設計に関するお願い

1. 弊社は品質、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又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留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジ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。
2. 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、ルネサス テクノロジは責任を負いません。
3. 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、ルネサス テクノロジは、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。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (http://www.renesas.com) などを通じて公開される情報に常にご注意ください。
4. 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が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、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及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。ルネサス テクノロジは、適用可否に対する責任は負いません。
6. 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、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 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ルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

重要事項

・当エミュレータをご使用になる前に、必ずユーザーズマニュアルをよく読んで理解してください。
・ユーザーズマニュアルは、必ず保管し、使用上不明な点がある場合は再読してください。

エミュレータとは：

ここでいうエミュレータとは、株式会社ルネサス テクノロジ（以下、「ルネサス」という。）が製作した次の製品を指します。

（1）E6000エミュレータ本体、（2）ユーザシステムインタフェースケーブル、（3）PCインタフェースボード、（4）オプションメモリボード、（5）オプションボード

お客様のユーザシステム及びホストコンピュータは含みません。

エミュレータの使用目的：

当エミュレータは、ルネサスマイクロコンピュータ（以下、MCUと略します）を使用したシステムの開発を支援する装置です。ソフトウェアとハードウェアの両面から、システム開発を支援します。

この使用目的に従って、当エミュレータを正しく使用してください。この目的以外に当エミュレータを使用することを堅くお断りします。

使用制限：

当エミュレータは、開発支援用として開発したものです。したがって、機器組み込み用として使用しないでください。また、以下に示す開発用途に対しても使用しないでください。

1　ライフサポート関連の医療機器用（人命にかかわる装置用）
2　原子力開発機器用
3　航空機開発機器用
4　宇宙開発機器用

このような目的で当エミュレータの採用をお考えのお客様は、当社営業窓口へ是非ご連絡頂きますようお願い致します。

製品の変更について：

ルネサスは、当エミュレータのデザイン、機能および性能を絶えず改良する方針をとっています。したがって、予告なく仕様、デザイン、およびユーザーズマニュアルを変更することがあります。

エミュレータを使う人は：

当エミュレータは、ユーザーズマニュアルをよく読み、理解した人のみが使用してください。

特に、当エミュレータを初めて使用する人は、当エミュレータをよく理解し、使い慣れている人から指導を受けることをおすすめします。

保証の範囲：

ルネサスは、お客様が製品をご購入された日から１年間は、無償で故障品を修理、または交換いたします。

ただし、（１）製品の誤用、濫用、またはその他異常な条件下での使用
（２）ルネサス以外の者による改造、修理、保守、またはその他の行為
（３）ユーザシステムの内容、または使用
（４）火災、地震、またはその他の事故

により、故障が生じた場合はご購入日から１年以内でも有償で修理、または交換を行います。また、日本国内で購入され、かつ、日本国内で使用されるものに限ります。

その他の重要事項：

１　本資料に記載された情報、製品または回路の使用に起因する損害または特許権その他権利の侵害に関しては、ルネサスは一切その責任を負いません。

２　本資料によって第三者またはルネサスの特許権その他権利の実施権を許諾するものではありません。

版権所有：

このユーザーズマニュアルおよび当エミュレータは著作権で保護されており、すべての権利はルネサスに帰属しています。このユーザーズマニュアルの一部であろうと全部であろうといかなる箇所も、ルネサスの書面による事前の承諾なしに、複写、複製、転載することはできません。

図について：

このユーザーズマニュアルの一部の図は、実物と異なっていることがあります。

予測できる危険の限界：

ルネサスは、潜在的な危険が存在するおそれのあるすべての起こりうる諸状況や誤使用を予見できません。したがって、このユーザーズマニュアルと当エミュレータに貼付されている警告がすべてではありません。お客様の責任で、当エミュレータを正しく安全に使用してください。

安全事項

・当エミュレータをご使用になる前に、必ずユーザーズマニュアルをよく読んで理解してください。
・ユーザーズマニュアルは、必ず保管し、使用上不明な点がある場合は再読してください。

シグナル・ワードの定義

これは、安全警告記号です。潜在的に、人に危害を与える危険に対し注意を喚起するために用います。起こり得る危害又は死を回避するためにこの記号の後に続くすべての安全メッセージに従ってください。

⚠ 危険 **危険**は、回避しないと、死亡又は重傷を招く差し迫った危険な状況を示します。
ただし、本製品では該当するものはありません。

⚠ 警告 **警告**は、回避しないと、死亡又は重傷を招く可能性がある潜在的に危険な状況を示します。

⚠ 注意 **注意**は、回避しないと、軽傷又は中程度の傷害を招くことがある潜在的に危険な状況を示します。

注意 安全警告記号の付かない**注意**は、回避しないと、財物損傷を引き起こすことがある潜在的に危険な状況を示します。

注、留意事項は、 例外的な条件や注意を操作手順や説明記述の中で、ユーザに伝達する場合に使用しています。

⚠警告

１．感電、火災等の危険防止および品質保証のために、お客様ご自身による修理や改造は行なわないでください。故障の際のアフターサービスにつきましては、ルネサスまたはルネサス特約店保守担当にお申し付けください。
２．エミュレータまたはユーザシステムのパワーオン時、すべてのケーブル類の抜き差しを行なわないでください。抜き差しを行なった場合、エミュレータとユーザシステムの発煙、発火の可能性があります。また、デバッグ中のユーザプログラムを破壊する可能性があります。
３．エミュレータまたはユーザシステムのパワーオン時、エミュレータとユーザシステムインタフェースケーブルおよびユーザシステムインタフェースケーブルとユーザシステム上のICソケットの抜き差しを行なわないでください。
抜き差しを行なった場合、エミュレータとユーザシステムの発煙、発火の可能性があります。また、デバッグ中のユーザプログラムを破壊する可能性があります。
４．ユーザシステムインタフェースケーブルとユーザシステム上のICソケットはピン番号を確かめて正しく接続してください。
接続を誤るとエミュレータとユーザシステムの発煙、発火の可能性があります。
５．電源給電については電源仕様に従って供給してください。使用する電源ケーブルは製品に添付のものを使用してください。仕様以外の電源電圧を加えないでください。

はじめに

E6000 エミュレータは、ルネサス MCU をサポートする高性能リアルタイムインサーキットエミュレータです。本 E6000 エミュレータは H8 ファミリマイクロコントーラ用のプログラムの開発とデバッグができます。
E6000 エミュレータは、ソフトウェア開発とデバッグのために単体で、あるいはユーザシステムのデバッグのためにユーザシステムインタフェースケーブルでユーザシステムに接続した状態で使用できます。

High-performance Embedded Workshop (HEW)は、ルネサスのマイクロコンピュータ用に、C/C++言語およびアセンブリ言語で書いたアプリケーションの開発およびデバッグを簡単に行うためのグラフィカルユーザインタフェースを提供します。アプリケーションを実行するエミュレータのアクセス、計測、および変更に関して、HEW は高機能でしかも直観的な手段を提供することを目的としています。

HEW は、ルネサスマイクロコンピュータの組み込み用アプリケーションの開発を強力にサポートするツールです。おもな特徴をまとめると次のようになります。

使い勝手の良いインタフェースを活用したコンパイラ、アセンブラ、リンケージエディタなどのオプションが設定できるカスタマイズ可能なプロジェクトビルドシステム。
プログラムを読みやすくするシンタックス色付け機能を持つ統合化テキストエディタ。
ユーザ独自のツールを実行するための環境設定。
同一アプリケーション内でのビルドおよびデバッグを可能にする統合化デバッガ。
バージョン管理サポート。

HEW は 2 つの目的で設計されています。一つはユーザに強力な開発ツールを提供すること、そしてもう一つは、それらのツール類を統合して使いやすくすることです。

このマニュアルについて

本マニュアルでは HEW 編およびデバッガ編の 2 部構成になっています。それぞれ以下の内容を説明しています。

HEW 編	HEW の基本的な使い方に関する情報、HEW 環境のカスタマイズ、HEW のビルド機能
エミュレータデバッガ編	使用前の準備、E6000 エミュレータの機能、デバッグ機能、チュートリアル、E6000 エミュレータのハード仕様、E6000 エミュレータのソフトウェア仕様

このマニュアルでは C/C++言語、アセンブリ言語の書き方や、オペレーティングシステムの使い方、個々のデバイスに適したプログラムの書き方などについては説明していません。それらについては、各々のマニュアルを参照してください。

Microsoft®, MS-DOS, Windows®, Windows NT®は米国 Microsoft 社の米国およびその他の国における登録商標です。
Visual SourceSafe は Microsoft 社の米国およびその他の国における商標です。
IBM は International Business Machines Corporation の登録商標です。
その他、記載されている製品名は各社の商標または登録商標です。

このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1: 記号一覧

記号	意味
[Menu->Menu Option]	**太字と** '->' **はメニューオプションを示します (例** [File->Save As...])
FILENAME.C	**大文字の名前はファイル名を示します**
"**文字列の入力**"	**下線は入力する文字列を示します** ("" **を省く**)
Key + Key	**キー入力を示します。例えば、**CTRL+N **キーでは** CTRL **キーと** N **キーを同時に押します**
➲ (**「操作方法」マーク**)	**このマークが左端にあるとき、その右の文章は何かの操作方法を示します**
バックスラッシュ文字'\'	**文中および図中で使用しているファイルパス名の文字列中のバックスラッシュ文字は日本語** Windows®**では円記号として表示されます。**

梱包品の確認

梱包を解いた後、納入品明細書に記入されている梱包品がそろっているか確認してください。確認した結果、梱包品に不足がありましたら、当エミュレータ購入元の営業担当までご連絡ください。

目次

HEW 編 1

1. **概要** 3

1.1 ワークスペース、プロジェクト、ファイル 3

1.2 メインウィンドウ 4

1.2.1 タイトルバー 4
1.2.2 メニューバー 4
1.2.3 ツールバー 4
1.2.4 ワークスペースウィンドウ 7
1.2.5 エディタウィンドウ 11
1.2.6 アウトプットウィンドウ 12
1.2.7 ステータスバー 12

1.3 ヘルプ機能 12

1.4 HEW を起動する 13

1.5 HEW を終了する 14

1.6 ツールシステム概要 14

2. **ビルドの基本** 15

2.1 ビルド処理 15

2.2 プロジェクトファイル 16

2.2.1 プロジェクトにファイルを追加する 16
2.2.2 プロジェクトからファイルを削除する 19
2.2.3 ビルドからプロジェクトファイルを除外する 20
2.2.4 ビルドへプロジェクトファイルを入れる 20

2.3 ファイル拡張子とファイルグループ 21

2.4 ファイルのビルド方法を設定する 27

2.5 ビルドのコンフィグレーション 27

2.5.1 ビルドコンフィグレーションを選択する 28
2.5.2 ビルドコンフィグレーションを追加、削除する 29

2.6 プロジェクトをビルド実行する 30

2.6.1 プロジェクトをビルド実行する 30
2.6.2 1 つのファイルをビルド実行する 30
2.6.3 ビルド実行を中止する 30
2.6.4 複数のプロジェクトをビルド実行する 31
2.6.5 アウトプットウィンドウ 31
2.6.6 アウトプットウィンドウの内容の制御 32
2.6.7 ビルド対象ファイルをワークスペースウィンドウにマーク表示する 33

2.7 ファイル依存関係 33

2.8 ワークスペースウィンドウの構成 34

2.8.1 各ファイルの下に依存を表示する 34
2.8.2 標準ライブラリファイルのインクルードを表示する 35
2.8.3 ファイルのパスを表示する 35
2.8.4 ファイルをタイムスタンプの順序でソートする 36

2.9 アクティブプロジェクトを設定する 36
2.10 ワークスペースにプロジェクトを追加する 36
2.11 プロジェクト間の依存関係を指定する 37
2.12 ワークスペースからプロジェクトを削除する 38
2.13 ワークスペースのプロジェクトをロード、アンロードする 39
2.14 ワークスペースの相対プロジェクトパス 39
2.15 ワークスペース内のユーザフォルダ 40
3. ビルドの応用 41
3.1 ビルド実行の復習 41
3.1.1 ビルドとは? 41
3.2 カスタムビルドフェーズを作成する 43
3.3 ビルドのフェーズ順序 47
3.3.1 ビルドのフェーズ順序 48
3.3.2 ビルドファイルのフェーズ順序 50
3.4 カスタムビルドフェーズのオプション設定 51
3.4.1 オプションタブ 52
3.4.2 Output Files タブ 53
3.4.3 Dependent Files タブ 54
3.5 ファイルのマッピング 55
3.6 ビルドを管理する 57
3.7 ビルドの出力のログを取る 58
3.8 ツールチェインのバージョンを変更する 58
3.9 外部デバッガを使う 60
3.10 メイクファイルの生成 62
3.11 HEW システム内部のメイクファイルを使用する 64
3.12 リンク順序をカスタマイズする 66
4. エディタの使用 67
4.1 エディタウィンドウ 67
4.2 複数のファイルを使う 68
4.2.1 エディタツールバー 68
4.2.2 エディタツールバーボタン 68
4.2.3 検索ツールバーボタン 70
4.2.4 ブックマークツールバーボタン 70
4.2.5 テンプレートツールバーボタン 71
4.3 標準のファイル操作 71
4.3.1 新規ファイルの作成 71
4.3.2 ファイルの保存 71
4.3.3 全ファイルの保存 71
4.3.4 ファイルを開く 72
4.3.5 ファイルを閉じる 72
4.4 ファイルを編集する 72
4.5 検索とファイル内の移動 73
4.5.1 テキストの検索 73
4.5.2 複数のファイル間でのテキスト検索 75
4.5.3 テキストを置換する 76
4.5.4 指定した行にジャンプする 77
4.6 ブックマーク 77
4.7 ファイルを印刷する 78
4.8 テキストのレイアウト 78
4.8.1 ページ設定 78
4.8.2 タブを変更する 80
4.8.3 自動インデント 81

4.9 ウィンドウを分割する 82
4.10 テキストの表示の変更方法 83
4.10.1 エディタのフォントを変更する 83
4.11 シンタックスを色づけする 84
4.12 テンプレート 86
4.12.1 テンプレートを設定する 86
4.12.2 テンプレートを削除する 89
4.12.3 テンプレートを挿入する 89
4.12.4 かっこの組み合わせ 89
4.13 Editor カラムの管理 90
4.14 ツールチップウォッチ 91
5. ツール管理 93
5.1 ツールの位置 94
5.2 HEW 登録ファイル (*.HRF) 94
5.3 ツールを登録する 95
5.3.1 ドライブのツール検索 95
5.3.2 ツールを一つ登録する 96
5.4 ツールの登録を取り消す 96
5.5 ツールのプロパティの参照と編集 96
5.6 ツールのアンインストール 99
5.7 テクニカルサポートについて 101
5.8 オンデマンドのコンポーネント 103
5.9 カスタムプロジェクトタイプ 104
6. 環境のカスタマイズ 111
6.1 ツールバーのカスタマイズ 111
6.2 ツールメニューのカスタマイズ 113
6.3 ヘルプシステムを構築する 116
6.4 ワークスペースオプションを指定する 117
6.4.1 起動時に最後に開いたワークスペースを開くチェックボックス 118
6.4.2 ワークスペースを開いたときにファイル表示チェックボックス 118
6.4.3 ワークスペースを開いたときにワークスペース情報の表示チェックボックス 118
6.4.4 ツール実行前にワークスペースの保存チェックボックス 119
6.4.5 ワークスペース保存前に確認チェックボックス 119
6.4.6 セッション保存前に確認チェックボックス 119
6.4.7 自動バックアップ時間間隔チェックボックス 119
6.4.8 新規ワークスペースのデフォルトディレクトリエディットボックス 120
6.5 HEW エディタ以外のエディタを使う 120
6.6 ファイルの保存をカスタマイズする 122
6.6.1 ツール実行前にワークスペースの保存チェックボックス 122
6.6.2 ワークスペース保存前に確認チェックボックス 122
6.7 カスタムプレースホルダを使う 123
6.8 ワークスペースやプロジェクトのログ機能を使う 124
6.9 バーチャルデスクトップを使用する 125
7. バージョン管理 127
7.1 バージョン管理システムを選択する 128
8. カスタムバージョン管理システム 131
8.1 バージョン管理メニューオプションを定義する 131
8.1.1 システムメニューオプションとツールバーボタン 134
8.1.2 ユーザ定義メニューオプション 135
8.2 バージョン管理コマンドを定義する 136
8.2.1 Executable return code オプション 137

8.3 変数を指定する 137
8.3.1 ファイルの位置を指定する 138
8.3.2 環境変数の設定 142
8.3.3 コメントを指定する 142
8.3.4 ユーザ名とパスワードを指定する 144
8.4 実行を制御する 146
8.4.1 Prompt before executing command チェックボックス 146
8.4.2 Run in DOS Window チェックボックス 146
8.4.3 Use forward slash '/' as version control directory delimiter チェックボックス 146
8.5 設定内容の保存と適用 147

9. Visual SourceSafe を使用する 149

9.1 ワークスペースに Visual SourceSafe を関連づける 149
9.1.1 Visual SourceSafe を選ぶ 149
9.1.2 Visual SourceSafe にファイルを追加する 150
9.2 Visual SourceSafe コマンド 151
9.2.1 バージョン管理からファイルを削除する 151
9.2.2 バージョン管理からファイルの読み取り専用コピーを取得する 151
9.2.3 バージョン管理からファイルの書き込み可能コピーをチェックアウトする 152
9.2.4 バージョン管理にファイルの書き込み可能コピーをチェックインする 152
9.2.5 チェックアウト操作を取り消す 152
9.2.6 ファイルの状態を表示する 153
9.2.7 ファイル履歴を表示する 153
9.3 Visual SourceSafe 統合化オプション 153

10. ネットワーク機能 155

10.1 概要 155
10.1.1 ネットワークアクセスを使う 156
10.1.2 アドミニストレータユーザのパスワードを設定する 156
10.1.3 新規ユーザを追加する 158
10.1.4 パスワードを変更する 159
10.1.5 ネットワーク HEW サービスを利用する 159

11. 相違点の表示 161

12. テクニカルサポート 165

13. ナビゲーション機能 167

13.1 C++ナビゲーション機能 170
13.2 C 関数と#define ナビゲーション機能 173

14. スマートエディタ 175

エミュレータデバッガ編 179

1 はじめに 181

1.1 特長 181
1.2 使用上の注意事項 182
1.3 使用環境条件 183
1.4 外形寸法と質量 184

2 使用前の準備 185

2.1 E6000 エミュレータ使用フローチャート 185
2.2 エミュレータソフトウェアのインストール 185
2.3 ユーザシステムへの接続 186
2.3.1 ユーザシステムインタフェースケーブル先端部とユーザシステムの接続例 186
2.3.2 ユーザシステムインタフェースケーブル本体部と E6000 エミュレータの接続 187
2.3.3 ユーザシステムインタフェースケーブル本体部と先端部の接続 187

2.4 電源供給 188
2.4.1 AC 電源アダプタ 188
2.4.2 極性 188
2.4.3 電源モニタ回路 188
2.5 ハードウェアインタフェース 188
2.5.1 クロック発振器 188
2.5.2 外部プローブ／トリガ出力 189
2.5.3 電源フォロワ回路 190
2.6 システムチェック 191
2.7 通信不良 196
2.8 その他の起動方法 196
2.9 アンインストール 196
3 E6000 エミュレータ機能 197
3.1 デバッグの特長 197
3.1.1 ブレークポイント 197
3.1.2 トレース 197
3.1.3 実行時間測定 197
3.1.4 バスモニタ 197
3.2 イベント検出システム（CES: Complex Event System） 198
3.2.1 イベントチャネル 198
3.2.2 範囲チャネル 198
3.2.3 ブレーク 198
3.2.4 イベント間実行時間測定 198
3.3 ハードウェアの特長 199
3.3.1 メモリ 199
3.3.2 エミュレーションクロック 199
3.3.3 外部プローブ 199
3.4 スタックトレース機能 199
3.5 オンラインヘルプ 199
4 デバッグの準備をする 201
4.1 ワークスペース、プロジェクト、ファイル 201
4.2 HEW の起動方法 202
4.2.1 新規にワークスペースを作成する場合(ツールチェイン未使用) 203
4.2.2 新規にワークスペースを作成する場合(ツールチェイン使用) 205
4.2.3 既存のワークスペースを指定する場合 208
4.3 E6000 エミュレータ起動時の設定 209
4.4 デバッグセッション 211
4.4.1 セッションを選択する 211
4.4.2 セッションの追加と削除 211
4.4.3 セッション情報を保存する 213
4.4.4 セッション情報の再ロード 214
4.5 エミュレータの接続 214
4.6 エミュレータの再接続 215
4.7 エミュレータの終了 215
5 デバッグ 217
5.1 エミュレーション環境を設定する 217
5.1.1 Configuration Properties ダイアログボックスを開く 217
5.1.2 MCU 一覧にない MCU を設定する 218
5.1.3 接続するインタフェースを選択する 219
5.1.4 Memory Mapping ダイアログボックスを開く 220
5.1.5 メモリマップ設定を変更する 221
5.2 プログラムをダウンロードする 222
5.2.1 プログラムをダウンロードする 222
5.2.2 ソースコードを表示する 223

5.2.3 アセンブリ言語コードを表示する 225
5.2.4 アセンブリ言語コードを修正する 226
5.2.5 特定のアドレスを見る 226
5.2.6 現在のプログラムカウンタアドレスを見る 226
5.3 コマンドラインインタフェースでデバッグする 227
5.3.1 コマンドラインウィンドウを開く 227
5.3.2 コマンドファイルを設定する 227
5.3.3 コマンドファイルを実行する 228
5.3.4 コマンド実行を中断する 228
5.3.5 ログファイルを設定する 228
5.3.6 ログファイルへの出力を開始/停止する 228
5.3.7 ファイルの full パスを入力する 228
5.3.8 プレースホルダを入力する 228
5.4 レジスタ内容を見る 229
5.4.1 レジスタウィンドウを開く 229
5.4.2 ビットレジスタを拡張する 229
5.4.3 表示するレジスタを選択する 230
5.4.4 ウィンドウを分割表示する 230
5.4.5 レジスタの内容を修正する 230
5.4.6 レジスタの内容を使用する 231
5.4.7 現在の表示内容をセーブする 231
5.5 メモリを操作する 231
5.5.1 メモリ領域を見る 231
5.5.2 異なるフォーマットでデータを表示する 232
5.5.3 ウィンドウを分割表示する 233
5.5.4 異なるメモリ領域を見る 233
5.5.5 メモリの内容を修正する 233
5.5.6 メモリ範囲を選択する 233
5.5.7 メモリ内の値を探す 234
5.5.8 メモリ範囲に値をフィルする 235
5.5.9 メモリ領域をコピーする 235
5.5.10 メモリ領域を保存、検証する 236
5.5.11 ウィンドウ内容更新を抑止する 236
5.5.12 ウィンドウ内容を更新する 237
5.5.13 メモリ内容を比較する 237
5.5.14 メモリ領域をファイルからロードする 237
5.6 I/O メモリを見る 238
5.6.1 IO ウィンドウを開く 238
5.6.2 I/O レジスタ 表示を拡張する 238
5.6.3 I/O ファイルの手動ロード 238
5.6.4 I/O レジスタの内容を修正する 239
5.6.5 現在の表示内容をセーブする 239
5.7 現在の状態を表示する 239
5.8 メモリ内容の日本語表示 240
5.8.1 メモリ内容を UNICODE 形式で表示する 240
5.8.2 メモリ内容を SJIS 形式または EUC 形式で表示する 240
5.9 エミュレータの情報を定期的に読み出し表示する 242
5.9.1 拡張モニタウィンドウを開く 242
5.9.2 表示項目を選択する 242
5.10 リアルタイムにメモリ内容を表示する 243
5.10.1 モニタウィンドウを開く 243
5.10.2 モニタの設定内容を変更する 245
5.10.3 モニタの更新を一時的に停止する 245
5.10.4 モニタ設定を削除する 245
5.10.5 変数の内容をモニタする 245
5.10.6 モニタウィンドウを非表示にする 245
5.10.7 モニタウィンドウを管理する 246

5.11 ラベルを見る 247
5.11.1 ラベルを一覧にする 247
5.11.2 ラベルを追加する 248
5.11.3 ラベルを編集する 248
5.11.4 ラベルを削除する 248
5.11.5 すべてのラベルを削除する 249
5.11.6 ラベルをファイルからロードする 249
5.11.7 ラベルをファイルに保存する 249
5.11.8 ラベルを検索する 250
5.11.9 次を検索する 250
5.11.10 ラベルに対応するソースプログラムを表示する 250
5.12 プログラムを実行する 251
5.12.1 リセットから実行を開始する 251
5.12.2 実行を継続する 251
5.12.3 カーソルまで実行する 251
5.12.4 開始アドレスを指定して実行する 252
5.12.5 シングルステップ 252
5.12.6 複数のステップ 253
5.13 プログラムを停止する 253
5.13.1 [停止]ツールバーボタンによる停止 253
5.13.2 標準のブレークポイント(PC ブレークポイント) 254
5.14 Elf/Dwarf2 のサポート 255
5.14.1 C/C++演算子 255
5.14.2 C/C++の式 255
5.14.3 複数ラベルをサポートする 256
5.14.4 オーバレイプログラムのデバッグ 257
5.15 変数の表示 259
5.15.1 ツールチップウォッチ 259
5.15.2 インスタントウォッチ 259
5.15.3 ウォッチウィンドウ 259
5.15.4 ローカルウィンドウ 264
5.16 イベントポイントを使用する 265
5.16.1 PC ブレークポイントとは 265
5.16.2 イベントポイントとは 265
5.16.3 イベント検出システムとは 265
5.16.4 バス状態およびエリア信号について 266
5.16.5 イベントポイントウィンドウを開く 267
5.16.6 PC ブレークポイントを設定する 267
5.16.7 イベントポイントを設定する 268
5.16.8 トリガポイントを設定する 274
5.16.9 イベントポイントの編集 275
5.16.10 イベントポイントの設定内容を変更する 275
5.16.11 イベントポイントを有効にする 275
5.16.12 イベントポイントを無効にする 275
5.16.13 イベントポイントを削除する 276
5.16.14 イベントポイントをすべて削除する 276
5.16.15 イベントポイントのソース行を表示する 276
5.17 トレース情報を見る 276
5.17.1 Trace ウィンドウを開く 276
5.17.2 トレース情報を取得する 276
5.17.3 トレース情報取得条件を設定する 277
5.17.4 Trace レコードを検索する 285
5.17.5 トレース情報をクリアする 290
5.17.6 トレース情報をファイルに保存する 290
5.17.7 ソースウィンドウを表示する 291
5.17.8 ソース表示を整形する 291
5.17.9 トレース情報のスナップショットを取得する 291
5.17.10 トレース情報の取得を一時的に停止する 291

5.17.11 トレース情報の取得を再開する 291
5.17.12 取得したトレース情報から必要なレコードを抽出する 291
5.17.13 タイムスタンプの差を計算する 299
5.17.14 統計情報を解析する 300
5.17.15 取得したトレース情報から関数呼び出し箇所を抽出する 301
5.18 関数呼び出し履歴を見る 302
5.18.1 スタックトレースウィンドウを開く 302
5.18.2 ソースプログラムを表示する 302
5.18.3 表示形式を設定する 303
5.19 メモリ内容を画像形式で表示する 304
5.19.1 画像ウィンドウを開く 304
5.19.2 ウィンドウを自動更新する 305
5.19.3 ウィンドウを更新する 305
5.19.4 ピクセル情報を表示する 306
5.20 メモリ内容を波形形式で表示する 307
5.20.1 波形ウィンドウを開く 307
5.20.2 ウィンドウを自動更新する 308
5.20.3 ウィンドウを更新する 308
5.20.4 拡大表示する 308
5.20.5 縮小表示する 308
5.20.6 最初のサイズに戻す 308
5.20.7 拡大/縮小倍率を設定する 308
5.20.8 横軸のサイズを設定する 308
5.20.9 カーソルを非表示にする 308
5.20.10 サンプリング情報を表示する 308
5.21 パフォーマンスを測定する 309
5.21.1 パフォーマンス解析ウィンドウを開く 309
5.21.2 実行効率測定条件を設定する 310
5.21.3 実行効率測定を開始する 310
5.21.4 測定条件を削除する 310
5.21.5 すべての測定条件を削除する 310
6 チュートリアル 311
6.1 はじめに 311
6.2 HEW の起動 311
6.3 エミュレータの接続 312
6.3.1 セッションを選択する 312
6.3.2 エミュレータの接続 312
6.4 チュートリアルプログラムのダウンロード 313
6.4.1 チュートリアルプログラムをダウンロードする 313
6.4.2 ソースプログラムを表示する 314
6.5 PC ブレークポイントの設定 315
6.6 レジスタ内容の変更 316
6.7 プログラムの実行 317
6.8 ブレークポイントの確認 318
6.9 シンボルの参照 319
6.10 メモリ内容の確認 320
6.11 変数の参照 321
6.12 ローカル変数の表示 323
6.13 プログラムのステップ実行 324
6.13.1 ステップインコマンドの実行 324
6.13.2 ステップアウトコマンドの実行 325
6.13.3 ステップオーバコマンドの実行 326

6.14 プログラムの強制ブレーク 327
6.15 MCU のリセット 327
6.16 ブレーク機能 328
6.16.1 PC ブレーク機能 328
6.16.2 イベントポイントによるブレーク機能 331
6.17 トレース機能 334
6.17.1 トレースの表示（タイムスタンプ無効時） 334
6.17.2 トレースの表示（タイムスタンプ有効時） 339
6.17.3 統計 342
6.17.4 関数コール 344
6.18 スタックトレース機能 346
6.19 モニタ機能 347
6.20 さてつぎは？ 350
7 本製品固有のハードウェア仕様 351
7.1 H8/388R E6000 エミュレータ仕様 351
7.1.1 サポート範囲 351
7.1.2 動作電圧および動作周波数 352
7.2 ユーザシステムインタフェース回路 353
7.2.1 信号保護 353
7.2.2 ユーザインタフェース回路 353
7.3 MCU と E6000 エミュレータの相違点 355
7.3.1 A/D コンバータ 355
7.3.2 未使用領域のアクセス 355
7.3.3 Go Reset コマンドによるプログラム実行 355
8 本製品固有のソフトウェア仕様 357
8.1 H8/388R E6000 エミュレータソフトウェア仕様 357
8.1.1 対応ハードウェア 357
8.1.2 選択可能プラットフォーム 357
8.1.3 Configuration Properties ダイアログボックス（General ページ） 358
8.1.4 Memory Mapping 機能 361
8.1.5 ステイタスウィンドウ 361
8.1.6 拡張モニタ機能 363
8.1.7 バス状態およびエリア信号 363
8.1.8 モニタ機能 364
8.1.9 トリガポイント 364
8.1.10 トレース情報 364
8.1.11 トレースレコードの検索 365
8.1.12 Trace Filter 機能 365
8.1.13 パフォーマンス解析ウィンドウ 365
8.2 H8/388R E6000 エミュレータ使用上の注意事項 366
8.2.1 チュートリアルプログラムの実行環境 366
8.2.2 サブクロック動作 366

付録 A トラブルシューティング 367

付録 B 正規表現 369

付録 C プレースホルダ 371

C.1 プレースホルダとは? 371

C.2 プレースホルダを挿入する 371

C.3 使用できるプレースホルダ 373

C.4 プレースホルダを使うにあたって 375

付録 D I/O ファイルフォーマット 377

D.1 ファイルフォーマット(ビットフィールド非対応) 377

D.2 ファイルフォーマット(ビットフィールド対応) 379

付録 E シンボルファイルフォーマット 381

付録 F HMAKE ユーザガイド 383

F.1 コマンドライン 383

F.1.1 基本構成 383

F.1.2 Exit コード 383

F.1.3 パラメータ 383

F.2 ファイルのシンタックス 384

F.2.1 変数宣言 384

F.3 記述部 384

F.3.1 概要 384

F.3.2 特殊なコマンド 385

F.3.3 サブコマンドファイル 385

F.4 コメント 386

F.5 メッセージコマンド 386

付録 G メニュー一覧 387

付録 H コマンドライン一覧 391

付録 I ハードウェア診断プログラムについて 393

I.1 テストプログラムを実行するためのシステムセットアップ 393

I.2 テストプログラムによる故障解析 394

I.3 エラー発生時の処理 397

HEW 編

1. 概要

本マニュアルでは HEW バージョン V.3.01.04 の機能について記載しています。
この章では HEW の基本概念を説明します。Windows®アプリケーションに慣れていないユーザのために、次章以降で必要となる情報を提供します。

1.1 ワークスペース、プロジェクト、ファイル

ワードプロセッサでドキュメントを作成、修正できるのと同じように、HEW ではワークスペースを作成、修正できます。
ワークスペースはプロジェクトを入れる箱と考えることができます。同じように、プロジェクトはプロジェクトファイルを入れる箱と考えることができます。したがって各ワークスペースにはプロジェクトが一つ以上あり、各プロジェクトにはファイルが一つ以上あります。この構成を図 1.1 に示します。

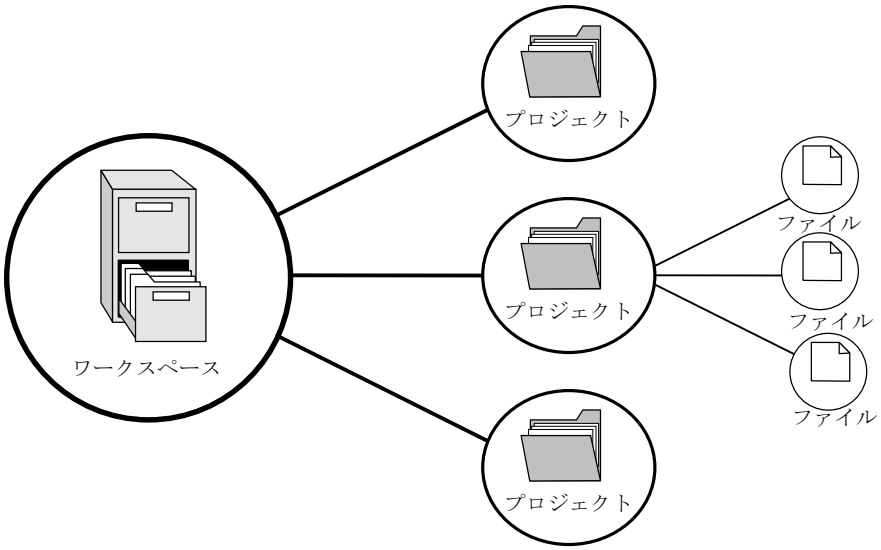


図 1.1: ワークスペース、プロジェクト、ファイル

ワークスペースでは関連したプロジェクトを一つにまとめることができます。例えば、異なるプロセッサに対して一つのアプリケーションを構築しなければならない場合、または、アプリケーションとライブラリを同時に開発している場合などに便利です。さらに、ワークスペース内でプロジェクトを階層的に関連づけることができます。つまり、一つのプロジェクトを構築すると、その子プロジェクトが最初に構築されます。

ワークスペースを活用するには、ユーザは、まずワークスペースにプロジェクトを追加して、そのプロジェクトにファイルを追加しなければなりません。

1.2 メインウィンドウ

HEW のメインウィンドウを図 1.2 に示します。

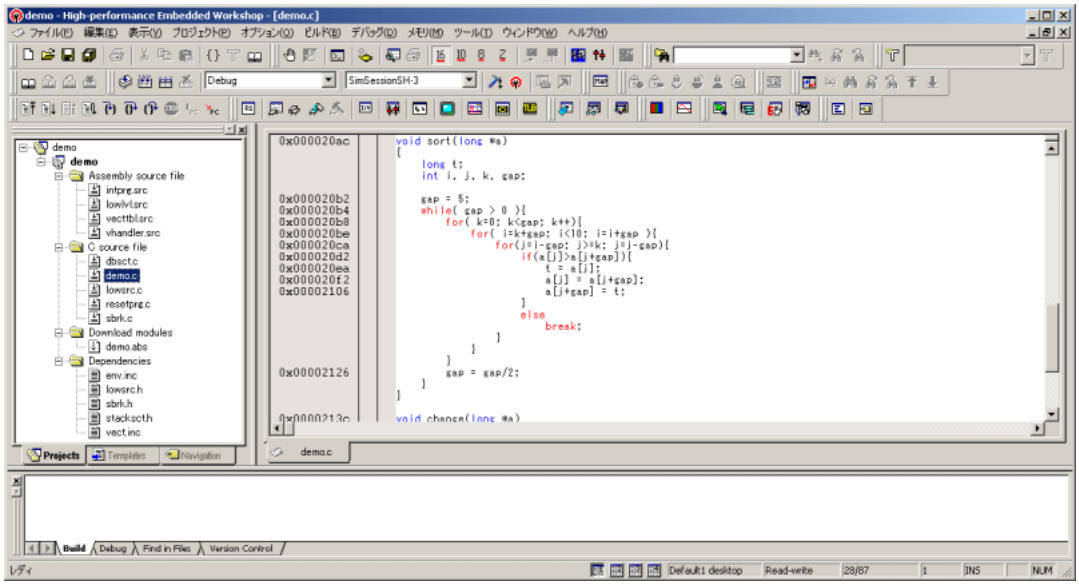


図 1.2: HEW **メインウィンドウ**

HEW にはメインウィンドウが 3 つあります。ワークスペースウィンドウ、エディタウィンドウ、アウトプットウィンドウです。ワークスペースウィンドウには現在そのワークスペースにあるプロジェクトやファイルを示します。エディタウィンドウではファイルを表示、編集できます。アウトプットウィンドウにはさまざまな処理結果（ビルド、バージョン管理コマンドなど）を表示します。

1.2.1 タイトルバー

タイトルバーには、現在開いている、プロジェクト名、ファイル名が表示されます。また、“最小化”ボタン、“最大化”ボタン、“閉じる”ボタンがあります。“最小化”ボタンをクリックすると Windows のタスクバー上に HEW が最小化されます。“最大化”ボタンをクリックすると HEW がフルスクリーンに表示されます。“閉じる”ボタンをクリックすると HEW を閉じることができます。（これは[ファイル->アプリケーションの終了] を選ぶか”ALT+F4”キーを押すのと同じです）。

1.2.2 メニューバー

メニューバーには次の 11 のメニューがあります。 [ファイル]、[編集]、[表示]、[プロジェクト]、[オプション]、[ビルド]、[デバッグ]、[メモリ]、[ツール]、[ウィンドウ]、[ヘルプ]です。メニューのオプションはすべてこれら 11 のメニューの下にグループ化されています。例えば、ファイルを開きたいときには[ファイル]メニューの下のオプションを選びます。ツールのセットアップをしたいときには、[ツール]メニューを選びます。メニューのオプションの機能については後の章で説明します。ここでは、各オプションの簡単な紹介をします。

1.2.3 ツールバー

ツールバーにより、使う頻度の高いオプションを簡単に利用できます。デフォルトでは[ブックマー

ク]、[デバッグ]、[デバッグラン]、[エディタ]、[検索]、[標準]、[テンプレート]、[バージョン管理]、[差分]、[Map]、[ツール]の 11 のツールバーがあります（図 1.3～図 1.13 参照）。ツールバーの作成や変更は [ツール->カスタマイズ...] メニューオプションで指定できます。 (詳細については 6 章「環境のカスタマイズ」を参照してください。)

図 1.3: **ブックマーク ツールバー**

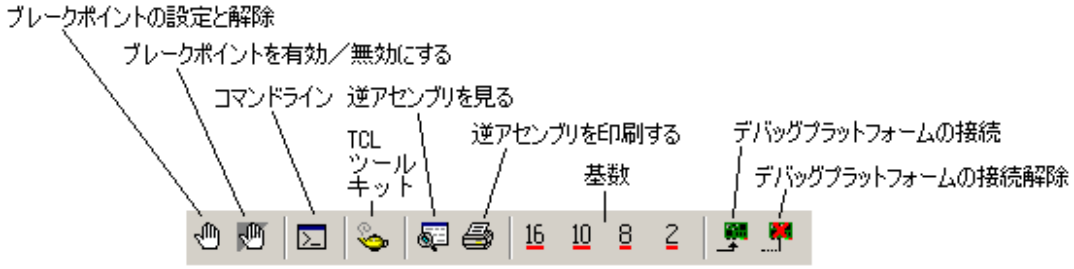


図 1.4: **デバッグ ツールバー**

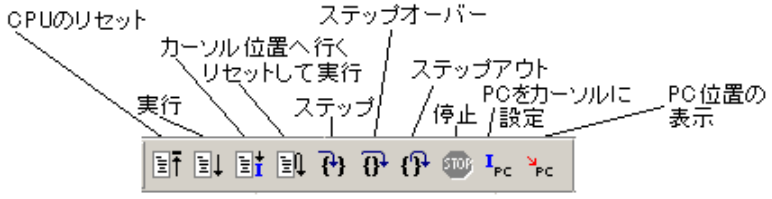


図 1.5: **デバッグランツールバー**

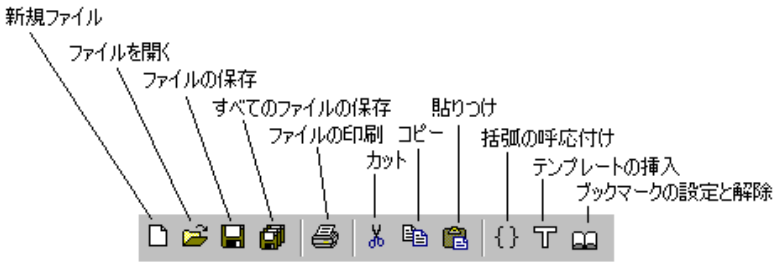


図 1.6: **エディタ ツールバー**

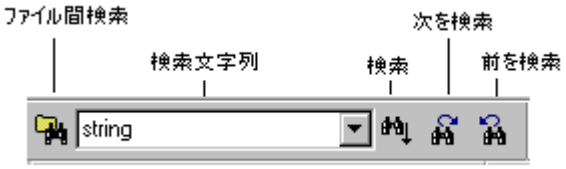


図 1.7: **検索ツールバー**

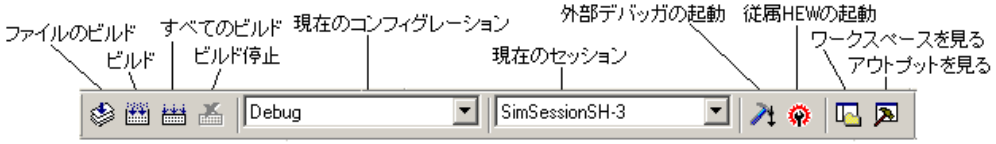


図 1.8: **標準ツールバー**

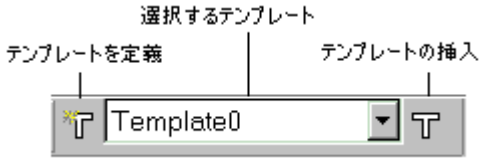


図 1.9: **テンプレートツールバー**

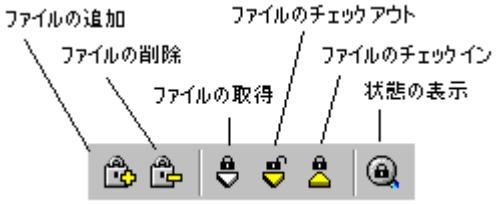


図 1.10: **バージョン管理ツールバー**

図 1.11: **差分ツールバー**

図 1.12: **マップツールバー**

図 1.13: **ツール ツールバー**

[標準]ツールバーがドッキング（連結）状態のとき、図 1.14.(i)に示すコントロールバーが表示されます。ドッキング状態の[標準]ツールバーの位置を移動したいときはコントロールバーを移動先までドラッグします。（ドラッグとは、マウスの左ボタンを押下したまま目的の場所まで移動してからボタンを離すことをいいます。）図 1.14.(i) がドッキング状態、1.14.(ii) がフローティング（浮遊）状態の[標準]ツールバーを示します。

(i)

(ii)

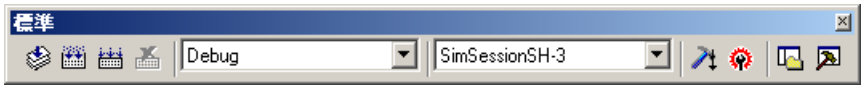


図 1.14: **標準ツールバーのドッキング／フローティング状態**

➲ ツールバーをドッキング状態にするには
フローティング状態のツールバーのタイトルバーをダブルクリックしてください。
または、フローティング状態のツールバーのタイトルバーをドッキング状態のウィンドウ、メニューバー、ツールバー、またはHEWメインウィンドウの端までドラッグします。バーの形が変わります。

➲ ツールバーをフローティング状態にするには
ドッキング状態のツールバーのコントロールバーをダブルクリックしてください。
または、ドッキング状態のツールバーのコントロールバーをHEWのメインウィンドウおよびその他のドッキング状態のウィンドウ、メニューバー、またはツールバーの端から外れるようにドラッグしてください。

1.2.4 ワークスペースウィンドウ

ワークスペースウィンドウにはタブが 3 つあります。 [Projects]タブには現在のワークスペース、プロジェクト、ファイルを示します（図 1.15）。アイコンをダブルクリックしてプロジェクトファイルや個々のファイルを開くことができます。

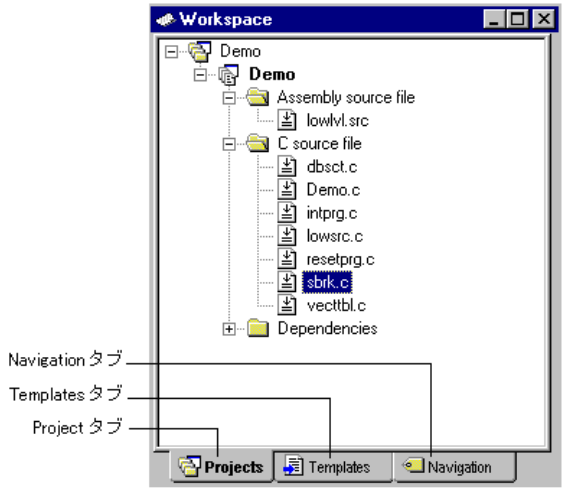


図 1.15: [**ワークスペース**] **ウィンドウ** [Projects] **タブ**

[Navigation]タブによりプロジェクトファイルの中のテキスト部へジャンプできます。[Navigation]タブに実際に表示される内容は、現在、何がインストールされているかによって異なります。図 1.16 には例えば ANSI 規格 の C 関数一覧を示します。ワークスペースウィンドウの詳細については 2 章「ビルドの基本」を参照してください。

[Templates]タブにはテンプレートの設定を表示します。テンプレートの詳細については 4.12 「テンプレート」を参照してください。

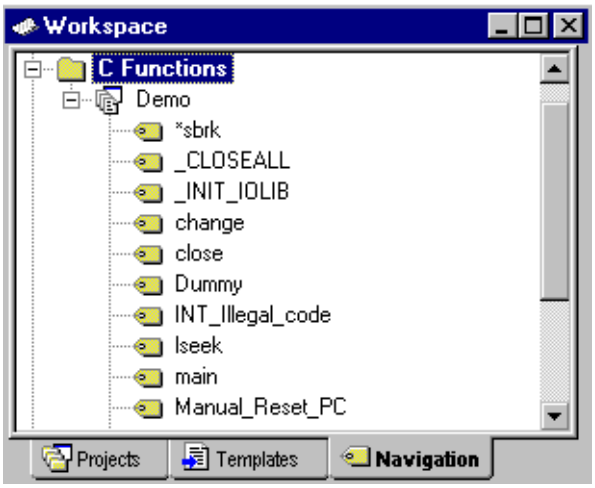


図 1.16: [**ワークスペース**] **ウィンドウ** [Navigation] **タブ**

➲ ワークスペースウィンドウやアウトプットウィンドウでドッキングを許すには

ウィンドウ上で右マウスボタンをクリックしてください。すると、ポップアップメニューが表示されます。ここで [ドッキングビュー] にチェック印が付いている場合、ドッキングが許されています。チェック印が外れている場合、ドッキングは許されていません。 [ドッキングビュー] を選択するとチェック印が付いたり外れたりします。

[ドッキングビュー] にチェック印が付いている場合、ウィンドウを HEW メインウィンドウや他のドッキング状態のウィンドウの端に連結できます。同じく[ドッキングビュー] にチェック印が付いている場合、ウィンドウを他の HEW のウィンドウ上や HEW メインウィンドウの外でフローティング状態にすることができます。図 1.17.(i) にはドッキング状態のワークスペースウィンドウ、図 1.17.(ii) にはフローティング状態のワークスペースウィンドウを示します。

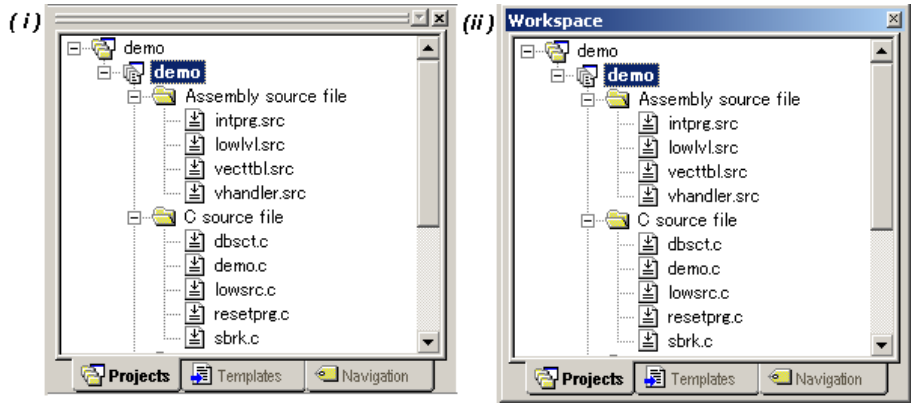


図 1.17: **ワークスペースウィンドウのドッキング／フローティング状態**

ワークスペースウィンドウやアウトプットウィンドウがドッキング状態のとき、図 1.18 に示すコントロールバーが表示されます。ドッキング状態のウィンドウを移動したいとき、コントロールバーを移動先までドラッグしてください。

図 1.18: **ドッキング状態ウィンドウのコントロールバー**

➲ ワークスペースウィンドウやアウトプットウィンドウをドッキング状態にするには

ワークスペースウィンドウやアウトプットウィンドウをドッキング状態にするにはポップアップメニューで[ドッキングビュー] にチェック印が付いている必要があります。（ポップアップメニューはウィンドウ上でマウスの右ボタンをクリックすると表示されます。）その上で、フローティング状態のウィンドウのコントロールバーをダブルクリックしてください。

または、フローティング状態のウィンドウのタイトルバーを、移動先のドッキング状態のウィンドウ、メニューバー、ツールバー、またはHEWのメインウィンドウの端までドラッグしてください。フローティング状態のウィンドウの形が変わります。

➲ ワークスペースウィンドウやアウトプットウィンドウをフローティング状態にするには

ワークスペースウィンドウやアウトプットウィンドウをフローティング状態にするにはポップアップメニューで[ドッキングビュー] にチェック印が付いている必要があります。（ポップアップメニューはウィンドウ上でマウスの右ボタンをクリックすると表示されます。）その上で、ドッキング状態のウィンドウのコントロールバーをダブルクリックしてください。

または、ドッキング状態のウィンドウのコントロールバーを、HEWのメインウィンドウや他のドッキング状態のウィンドウ、メニューバー、ツールバーの端から外れるようにドラッグしてください。

➲ ワークスペースウィンドウやアウトプットウィンドウを隠すには
ウィンドウの右上端にある“閉じる”ボタンをクリックしてください。または固定していないウィンドウの中で右マウスボタンをクリックし、ポップアップメニューから [非表示] を選んでください。

➲ ワークスペースウィンドウやアウトプットウィンドウを表示するには
ワークスペースウィンドウを表示するには[表示-> ワークスペース] を、 アウトプットウィンドウを表示するには[表示->アウトプット]を選んでください。

1.2.5 エディタウィンドウ

エディタウィンドウではプロジェクトのファイルを操作します。同時に複数のファイルを開いたり、任意の順序にファイルを切り替えたり、並べ替えたり、編集したりできます。デフォルトでは、エディタウィンドウはブック形式で表示されます。各テキストファイルにはタブがあります（図 1.19）。

エディタには、ウィンドウの左側に余白があります。これにより、ブックマークとブレークポイントの位置を速く簡単に設定できます。カラムの用途やその表示する情報について知りたい場合は、そのカラムにカーソルを置いてください。これを説明するツールチップが表示されます。

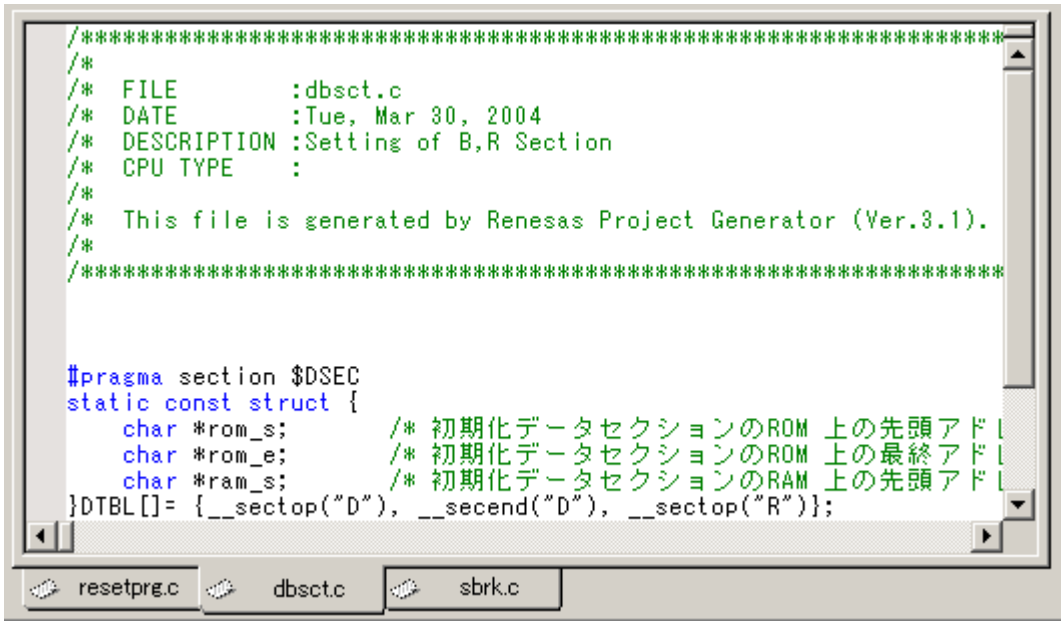


図 1.19: **エディタウィンドウ**

エディタウィンドウは[表示形式]ダイアログボックスでカスタマイズできます。[表示形式]ダイアログボックスは [ツール->表示形式...] メニューオプションから開くことができます。[表示形式]ダイアログボックスでは、エディタウィンドウのフォントやテキストの色、タブ文字の変更などができます。また、HEW でインストールした他のビューの外観も変更できます。HEW エディタ以外のエディタを使う場合は、使用するエディタを[オプション]ダイアログボックスで指定してください。[オプション]ダイアログボックスは、[ツール->オプション…] メニューオプションから開くことができます。エディタの使用方法や構築については、4 章「エディタの使用」を参照してください。

1.2.6 アウトプットウィンドウ

デフォルトではアウトプットウィンドウに 4 つのタブが表示されます。[Build]タブには任意のビルド実行（コンパイラ、アセンブラなど）の出力を示します。ソースファイルにエラーがある場合、[Build]タブにはエラーとソースファイル名と行番号が表示されます。エラーをダブルクリックすると、ソースファイルの行にジャンプするので、エラー箇所をすばやく発見できます。また、ダブルクリックにより、エラーまたは警告がステータスバーに表示されます。

注意　アウトプットウィンドウには、キーボードショートカット”SHIFT+ESC”があります。これを使用するとアウトプットウィンドウを閉じます。

図 1.20: **アウトプットウィンドウ**

[Debug]タブにはあらゆるデバッガ処理の出力を示します。情報を表示する必要のあるデバッグツールから、このウィンドウに出力が送られます。

[Find in Files]タブには最後の”Find in Files”操作の結果を示します。“Find in Files”を使用するには、[編集->ファイル内から検索…]メニューオプションを選ぶか、ツールバーの[ファイルの中から検索]ボタンをクリックしてください。“Find in Files”の使い方の詳細については、4 章「エディタの使用」を参照してください。

[Version Control]タブにはバージョン管理操作の結果を示します。このタブは、バージョン管理システムを使っているときだけ表示されます。バージョン管理の詳細については、7 章「バージョン管理」を参照してください。

1.2.7 ステータスバー

ステータスバーは 8 つの領域に分かれており、現在の HEW の状態を表示します。図 1.21 にステータスバーを示します。

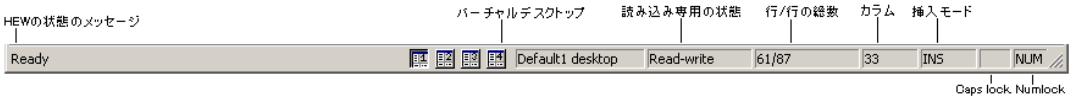


図 1.21:**ステータスバー**

1.3 ヘルプ機能

[ヘルプ]メニューは HEW メニューバーの右端にあります。[ヘルプ]メニューには[トピック]オプションがあります。[トピック]オプションを選ぶと、HEW ヘルプウィンドウのメイン画面が表示されます。

特定のダイアログボックスに関するヘルプを参照したいときは、各ダイアログボックスの右上端にあるコンテキスト依存ヘルプボタンをクリックしてください（図 1.22）。

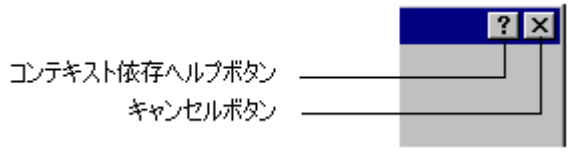


図 1.22: **ヘルプ ボタン**

コンテキスト依存ヘルプボタンをクリックすると、マウスポインタが？(クエスチョンマーク)付きのポインタに変わります。この状態で、ダイアログボックスの一部をクリックすると、その部分に関するヘルプを表示できます。

または、ある部分を選んで F1 キーを押下すると、その部分のヘルプを表示します。

1.4 HEW を起動する

HEW を起動するには Windows®の[スタート]メニューを開き、[プログラム]を選択し、 [Renesas High-performance Embedded Workshop]を選択し、HEW のショートカットを選びます。デフォルトで図 1.23 に示す[ようこそ!]ダイアログボックスが開きます。

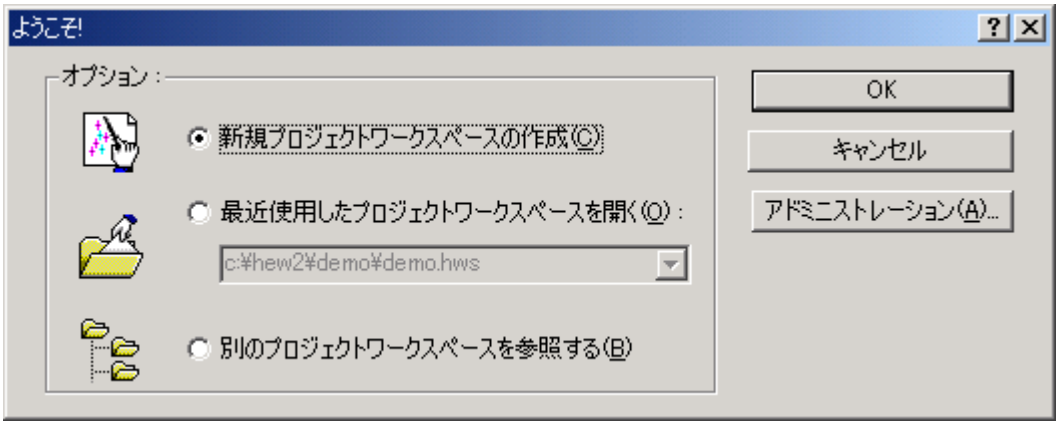


図 1.23: **ようこそ! ダイアログボックス**

新規ワークスペースを作成するには[新規プロジェクトワークスペースの作成]を選択し、[OK]ボタンをクリックしてください。最近開いたワークスペースを開くには[最近使用したプロジェクトワークスペースを開く]を選択し、ドロップダウンリストから開きたいワークスペースを選択し、[OK]ボタンをクリックしてください。最近開いたワークスペースのリストには、最近使ったワークスペースファイルリストで見るものと同じ情報が表示されます。このリストはファイルメニュー上に表示されます。ワークスペースファイルを指定してワークスペースを開くには[別のプロジェクトワークスペースを参照する]を選択し、[OK]ボタンをクリックしてください。HEW にツールを登録したり、HEW からツールの登録を外したりするには[アドミニストレーション]ボタンをクリックしてください。(詳細は、5 章、「ツール管理」を参照してください。) ワークスペースを開かないで HEW を使うには[キャンセル]をクリックしてください。

1.5 HEW を終了する

HEW を終了するには [ファイル->アプリケーションの終了]を選ぶか、“Alt+F4”キーを押下するか、システムメニューから[閉じる]オプションを選んでください。（システムメニューは HEW タイトルバーの最も左上側にあるアイコンをクリックすると開きます。）ワークスペースが開いているときは、前節で説明したワークスペースを閉じる操作を行います。

1.6 ツールシステム概要

ユーザは、更にツールを追加することによって、HEW の機能を拡張することができます。これを行うには、[ツールアドミニストレーション]ダイアログボックスでツールを登録しておくことが必要です。これらのツールを用いて、ウィンドウ、メニュー、およびツールバーを HEW システムに追加することができます。ツールの例としては、HEW のデバッガおよびビルダツールがあります。デバッガツールはデバッガに関連するすべてのメニューとツールバーを追加し、ビルダツールも同じようにビルド機能に関するすべてのメニューとツールバーを追加します。システムに登録したツールによって、HEW の使い方が変わります。そのため、このマニュアルに記載されたメニューのうち使用できないものもあります。例えば、デバッガツールがインストールされていない場合、HEW のメインウィンドウには[デバッグ]ツールがありません。

2. ビルドの基本

この章では HEW の一般的な機能を説明します。より高度な機能については 3 章 「ビルドの応用」を参照してください。

2.1 ビルド処理

ビルド処理の一般的な流れを図 2.1 に示します。HEW のインストール時に提供されるツールによってビルド処理は変わるので、図 2.1 の例とは少し異なるかもしれません（例えば、コンパイラが無いなど）。ビルドの各ステップまたはフェーズにおいて、1 セットのプロジェクトファイルについてビルド処理を行います。それが完了すると、次のステップまたはフェーズに移ります。

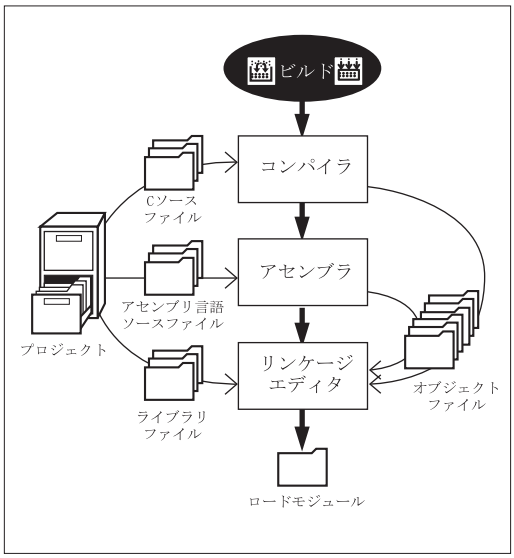


図 2.1: **ビルド処理の一般的な流れ**

図 2.1 に示す例では、第一のフェーズがコンパイラ、第二のフェーズがアセンブラ、そして最後のフェーズがリンケージエディタです。コンパイラのフェーズでは、プロジェクトの C/C++ソースファイルを順次コンパイルします。アセンブラのフェーズでは、アセンブリ言語のソースファイルを順次アセンブルします。リンケージエディタのフェーズでは、すべてのライブラリファイルと、コンパイラフェーズとアセンブラフェーズからの出力ファイルをリンクして、ロードモジュールを作成します。このモジュールは、HEW のデバッガ機能でダウンロードし、使用します。

ビルド処理をカスタマイズする方法はいくつかあります。例えば、独自のフェーズを追加したり、あるフェーズを無効にしたり、フェーズを削除できます。これらのビルド実行の応用については、3 章「ビルドの応用」を参照してください。

この章では、ビルドの一般的な原則や基本機能を説明します。

2.2 プロジェクトファイル

HEW を使ってアプリケーションをビルド処理するには、まず、どのファイルをプロジェクトに追加して、各ファイルをどのようにビルド処理すべきかを指定しなければなりません（図 2.2）。

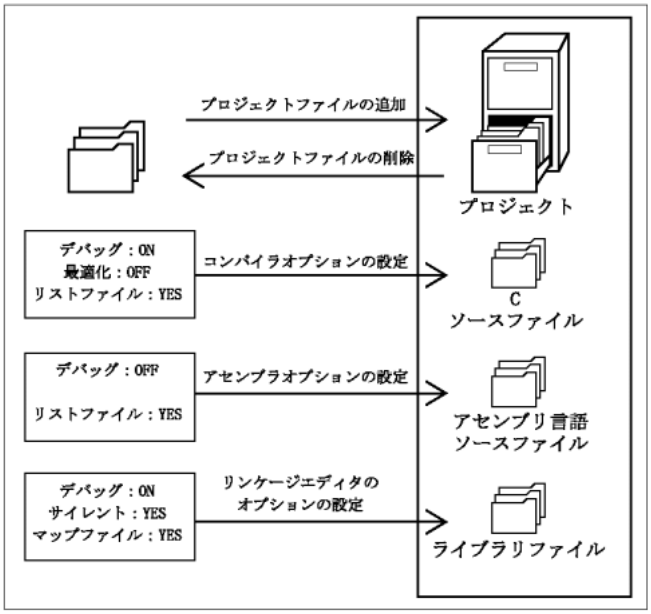


図 2.2: **プロジェクトの編集**

2.2.1 プロジェクトにファイルを追加する

アプリケーションをビルド実行する前に、まず、アプリケーションを構成するファイルを指定しなければなりません。

➲ プロジェクトにファイルを追加するには

1. **[プロジェクト->ファイルの追加...]**を選ぶか、ワークスペースウィンドウの[Projects]タブのプロジェクトアイコン上で右マウスボタンをクリックし、ポップアップメニューから**[ファイルの追加…]** を選ぶ（図2.3参照）か、ワークスペースウィンドウを選んで**INS**キーを押下してください。

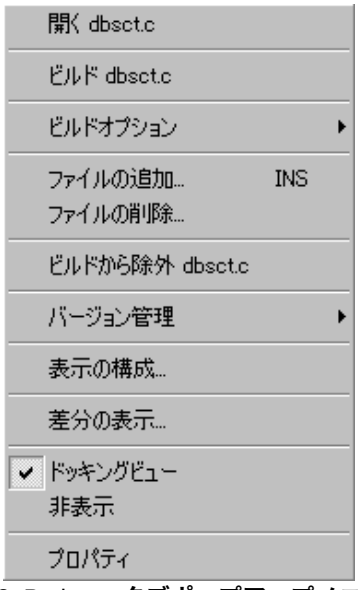


図 2.3: Projects **タブポップアップメニュー**

2. [ファイルの追加]ダイアログボックスが表示されます。
3. 追加するファイルを選び[追加]ボタンをクリックしてください。

プロジェクトに新しいファイルを追加するには、いくつか方法があります。これを以下で説明します。

エディタウィンドウ内で開いたファイルを右クリックすると、エディタウィンドウにポップアップメニューオプションが表示されます（図 2.4）。ファイルがすでにプロジェクト内にある場合、[プロジェクトにファイルの追加]メニューオプションは使用できません。[プロジェクトにファイルの追加]を選択すると、現在のプロジェクトにファイルを追加できます。

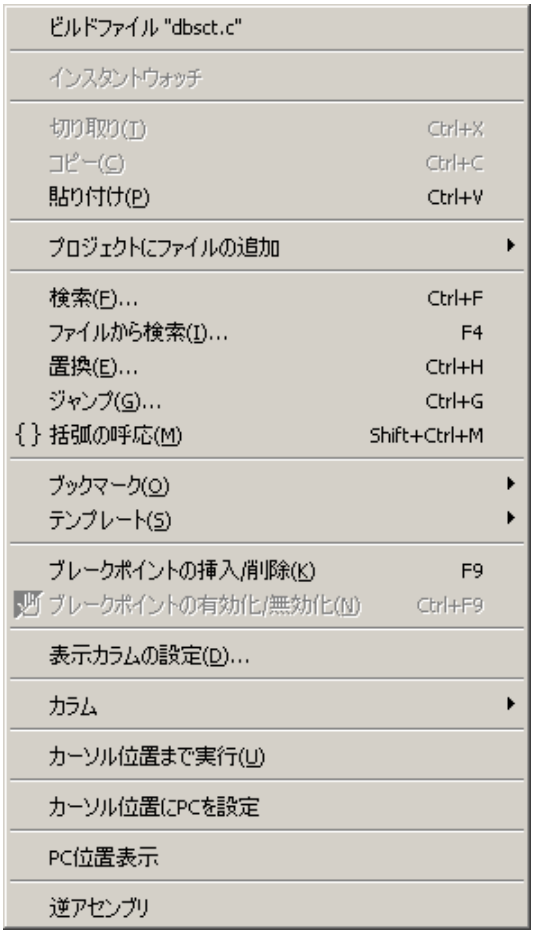


図 2.4: **エディタウィンドウポップアップメニュー**

HEW では、Windows® Explorer からワークスペースウィンドウにファイルを“ ドラッグしてドロップ ”することができます。こうしたファイルは自動的にプロジェクトに追加され、ドラッグされた先のフォルダに表示されます。

注意　プロジェクトに追加するファイルが、HEW の認識できない形式のファイルであっても、プロジェクトに追加されます。このファイルに関して、いくつかの機能が使用できなくなります。エディタ内でファイルを開く代わりにワークスペースウィンドウ内でこのファイルをダブルクリックすると、ファイルを開く動作が Windows オペレーティングシステムに受け渡されます。ファイルを Windows® Explorer 内で開いたかのように、ファイルを開くデフォルト動作が実行されます。現在、定義されている拡張子については、[ファイル拡張子]ダイアログボックスをご覧ください。（この章の後半にある「ファイルの拡張子とファイルグループ」を参照してください。）

2.2.2 プロジェクトからファイルを削除する

プロジェクトからファイルを削除できます。ファイルの削除は、一つでも、複数でも、すべてのファイルをまとめてでもできます。

➲ プロジェクトからファイルを削除するには

1. [プロジェクト->ファイルの削除...] を選択するか、ワークスペースウィンドウの[Projects]タブのプロジェクトアイコン上で右マウスボタンをクリックし、ポップアップメニューから [ファイルの削除…] を選択 (図2.5) してください。[プロジェクトファイルの削除]ダイアログボックスが表示されます（図2.6）。

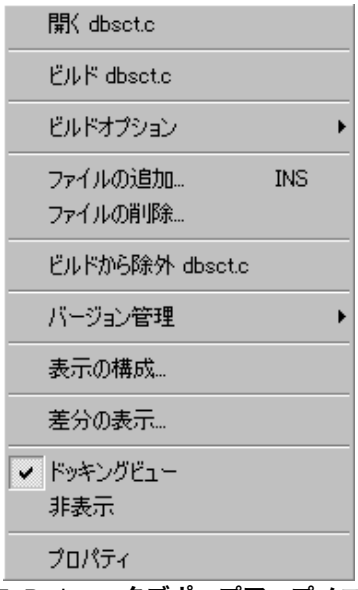


図 2.5: Projects **タブポップアップメニュー**

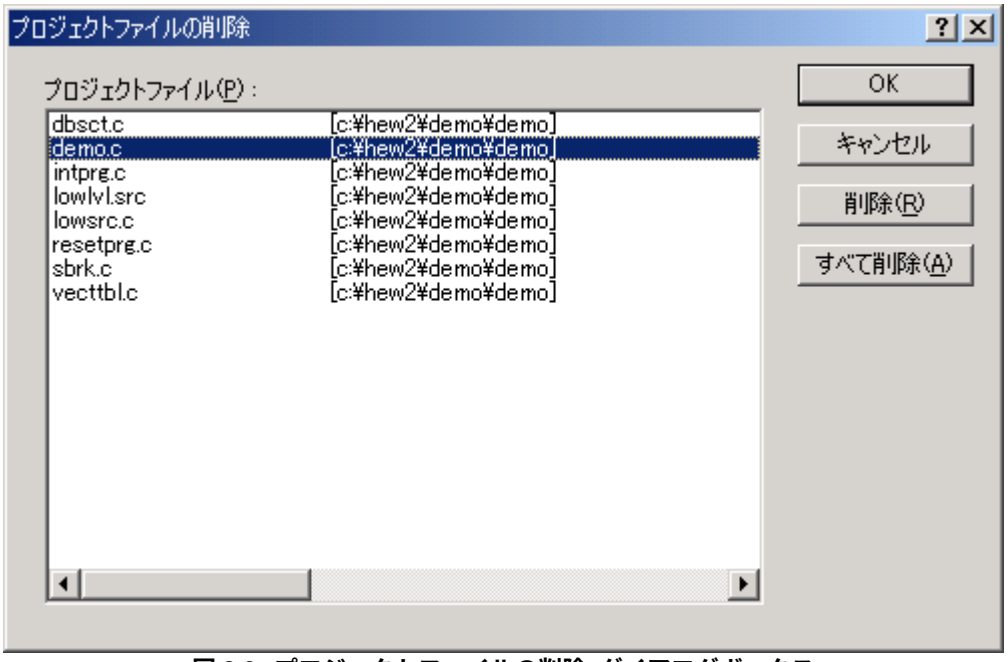


図 2.6: **プロジェクトファイルの削除 ダイアログボックス**

2. [プロジェクトファイル]リストから削除したいファイルを選んでください。複数のファイルを選ぶこともできます。
3. 選んだファイルを削除するには[削除]ボタンをクリックしてください。すべてのプロジェクトファイルを削除するには[すべて削除]ボタンをクリックしてください。
4. [OK]ボタンをクリックするとプロジェクトからファイルを削除します。

➲ 選んだファイルをプロジェクトから削除するには

1. ワークスペースウィンドウの[Projects]タブで削除したいファイルを選んでください。複数のファイルを選ぶときは”SHIFT”キーまたは”CTRL”キーを押下してください。
2. “DEL”キーを押してください。選んだファイルが削除されます。

2.2.3 ビルドからプロジェクトファイルを除外する

プロジェクトのファイルは、個々にビルドから除外することができます。

➲ ビルドからプロジェクトのファイルを除外するには

1. ワークスペースウィンドウの[Projects]タブで、ビルドから除外したいファイルを右マウスボタンでクリックしてください。
2. ポップアップメニュー（図2.5）から[ビルドから除外 ファイル名]を選んでください。すると、ファイルのアイコンに赤いバツ印がつけられ、ビルドから除外されます。

2.2.4 ビルドへプロジェクトファイルを入れる

除外したプロジェクトのファイルは再びビルドに入れることができます。

➲ 除外したファイルをビルドに再び入れるには

1. ワークスペースウィンドウの[Projects] タブでファイルを右マウスボタンで選んでください。
2. ポップアップメニューから[ビルドから除外の解除 ファイル名]を選んでください。 赤いバツ印が外され、ファイルがビルド可能になります。

2.3 ファイル拡張子とファイルグループ

HEW は拡張子でファイルを識別します。拡張子は使用するツールによって定義されます。例えば、コンパイラを使用すると拡張子.c が[C source file]グループに入り、コンパイラのフェーズに入力されます (図 2.1、ビルド処理の一般例)。さらに、独自の拡張子を定義することもできます。例えば、プロジェクトでアセンブリ言語ソースファイルを使っている場合、デフォルトの拡張子が.src だとします。.src の代わりに違う拡張子（例: .asm）を使うとき、新しい拡張子を定義してそれを.src ファイルと同様に扱うように指定できます。

ファイルの拡張子は、[ファイル拡張子]ダイアログボックスで表示、変更できます (図 2.7)。このダイアログボックスを表示するには **[プロジェクト->ファイルの拡張子...]**を選んでください。このダイアログボックスには現在のワークスペースで定義されたすべての拡張子とファイルグループを表示します。

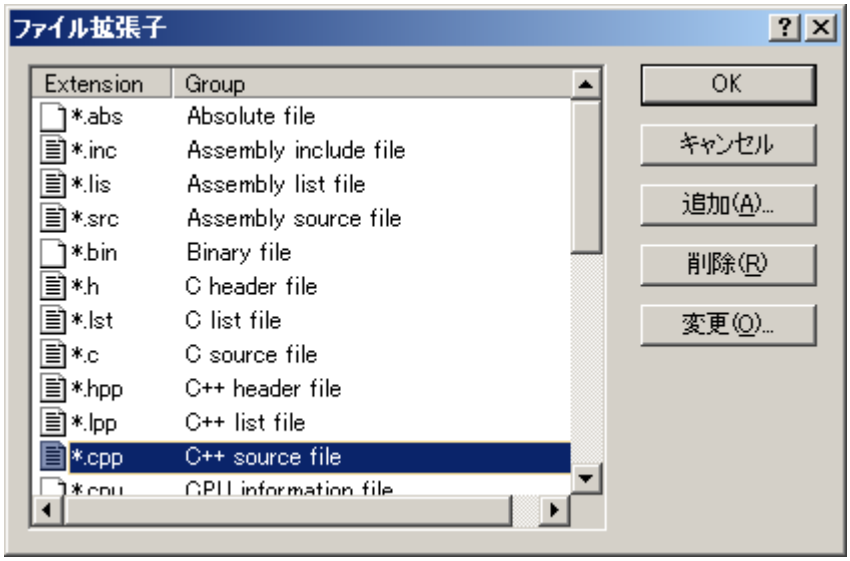


図 2.7: **ファイル拡張子 ダイアログボックス**

図 2.7 に示す[File Extensions]リストは 2 列に分かれています。左の列にはファイル拡張子、右の列にはファイルグループを表示します。図 2.8 に示すように、同じグループに多くのファイル拡張子が存在する場合があります。例えば、1 つのプロジェクト内でアセンブリ言語のソースファイルにいくつかの拡張子がある場合があります (例: .src, .asm, .mar など)。

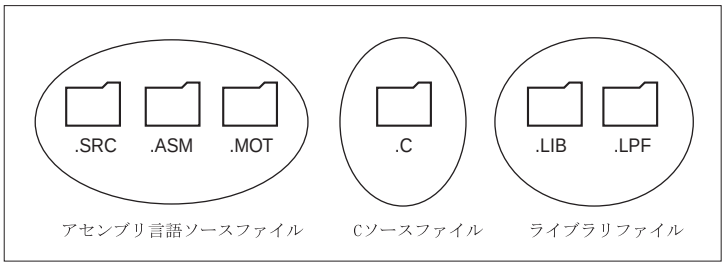


図 2.8:**ファイル拡張子とファイルグループ**

新しい拡張子を作成するときは、その拡張子がすでに定義されたファイルグループに属するのか、新しいファイルグループを作成する必要があるのか検討してください。新しい種類のファイルを追加するときには新しいファイルグループを作成してください。作成方法を以下に説明します。

➲ 新規ファイルグループに新規ファイル拡張子を作成するには

1. メニューバーから[プロジェクト->ファイルの拡張子...] を選んでください。[ファイル拡張子]ダイアログボックスが表示されます（図2.7）。
2. [追加…]ボタンをクリックしてください。[ファイル拡張子の追加]ダイアログボックスが表示されます（図2.9）。
3. [ファイル拡張子]フィールドに定義する拡張子を入力してください。ピリオド (.) の入力は不要です。ドロップリストには、現在のプロジェクトで未定義のすべての拡張子が含まれます。この拡張子のうち１つを選ぶと、ファイル拡張子フィールドにテキストが自動的に追加されます。
4. [新規拡張子グループ]オプションを選んで新しいファイルグループを定義する名前を入力してください。
5. この段階では、関連するアプリケーションの変更が可能です。“<拡張子グループ>を開くアプリケーション”のドロップダウンリストでは次の４つから選択が可能です。

   [Editor]
   [None]
   [Other]
   [Windows default]

   [Editor]を選択すると、ワークスペースウィンドウ内のファイルを開く機能によって、ファイルを HEW エディタ内で開くことができます。[None]を選択した場合、ファイルを開く機能を使用してもファイルを開くことはできません。[Other]を選択すると、ファイルを開くための他のツールを構築できます。詳細は、「アプリケーションとファイルグループを関連付けるには」を参照してください。[Windows default]を選択すると、ワークスペースウィンドウ内のファイルを開く機能によって、開いたファイルが Windows オペレーティングシステムに受け渡されます。また、Windows® Explorer で定義されたとして、このファイル拡張子のデフォルト動作を選択します。
6. [OK]ボタンをクリックすると[File Extension]リストに拡張子が追加されます。

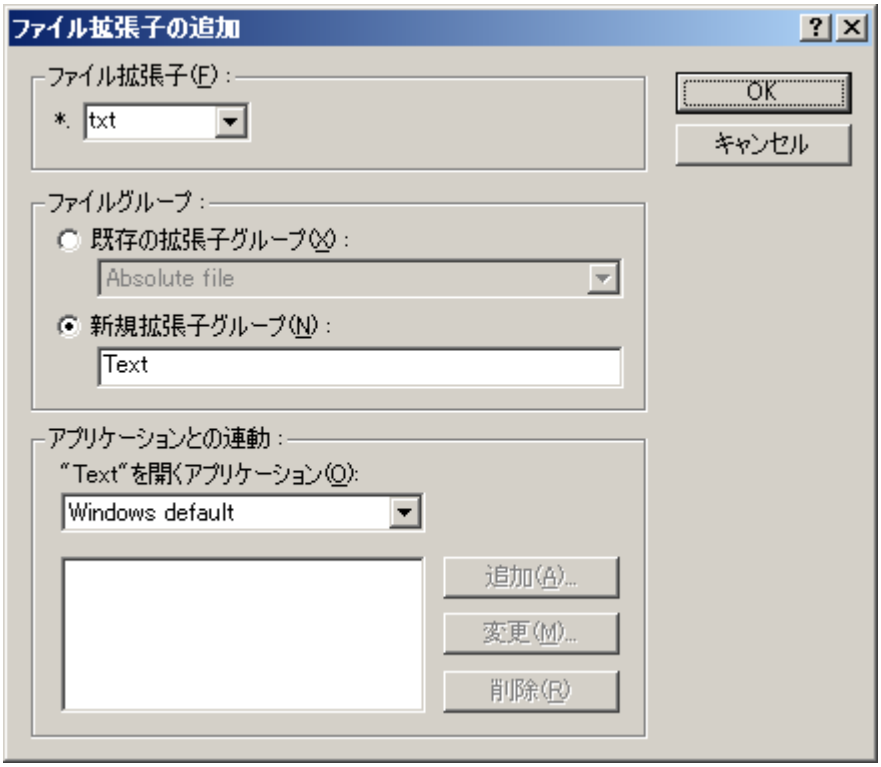


図 2.9: **ファイル拡張子の追加ダイアログボックス (新規グループ)**

HEW に登録されていない拡張子をもつプロジェクトを使っているときは、新しい拡張子を作成し、（例えば、フェーズ内のデフォルトの拡張子が.asm でビルダの認識する拡張子が.src のとき）新しい拡張子を既存のファイルグループに追加することが必要です。追加方法を下記に説明します。

➲ 新規ファイル拡張子を既存のファイルグループに追加するには

1. メニューバーから [プロジェクト->ファイルの拡張子...] を選んでください。[ファイル拡張子]ダイアログボックスが表示されます（図2.7）。
2. [追加…]ボタンをクリックすると [ファイル拡張子の追加]ダイアログボックスが表示されます(図2.10)。
3. [ファイル拡張子]フィールドに定義する拡張子を入力してください。 ピリオド (.) の入力は不要です。ドロップリストには、現在のプロジェクトで未定義のすべての拡張子が含まれます。この拡張子のうち１つを選ぶと、ファイル拡張子フィールドにテキストが自動的に追加されます。
4. [既存の拡張子グループ]オプションを選んでこの新しい拡張子をどのファイルグループに追加するか指定してください。
5. [OK]ボタンをクリックすると[File Extensions]リストに拡張子が追加されます。

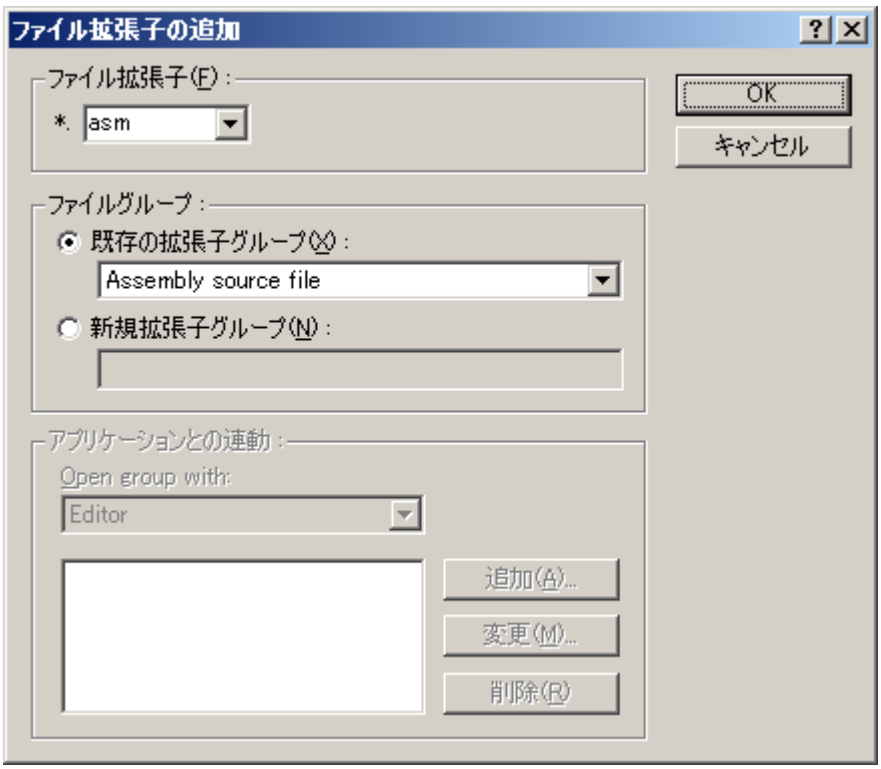


図 2.10: **ファイル拡張子の追加ダイアログボックス (既存グループ)**

– [ファイル拡張子]ダイアログボックスでは、エディタでファイルを開く指定だけでなく、ファイルグループとアプリケーションとの関連付けができます。これを行うと、ワークスペースウィンドウの[Projects]タブでファイルをダブルクリックすると、適切なアプリケー

ションでファイルを開きます。図 2.11 にワードプロセッサと拡張子.DOC の関連付けを示します。

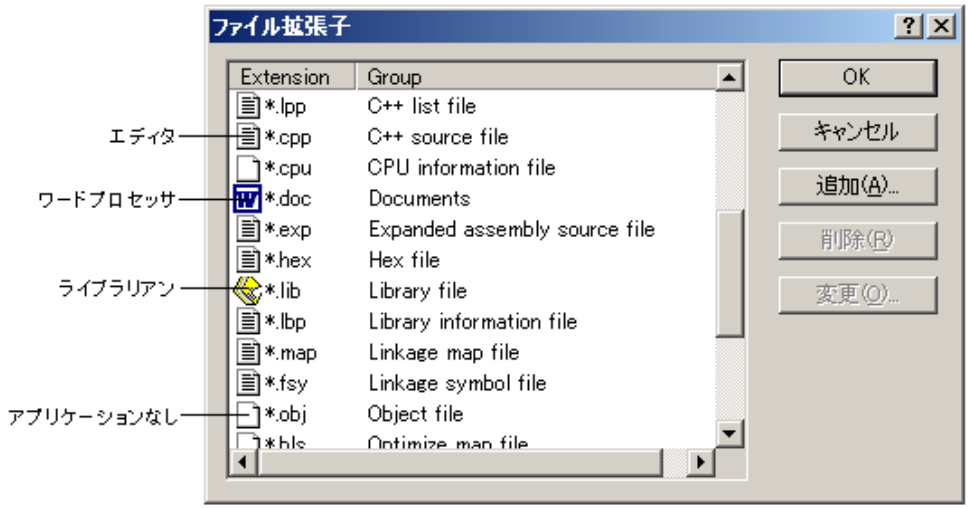


図 2.11:**ファイルグループとアプリケーション**

➲ アプリケーションとファイルグループを関連付けるには

1. [ファイル拡張子]ダイアログボックスで関連付けるファイルグループを選んでください（図2.11）。
2. [変更…]ボタンをクリックしてください。 [ファイル拡張子の変更]ダイアログボックスが表示されます（図2.12）。

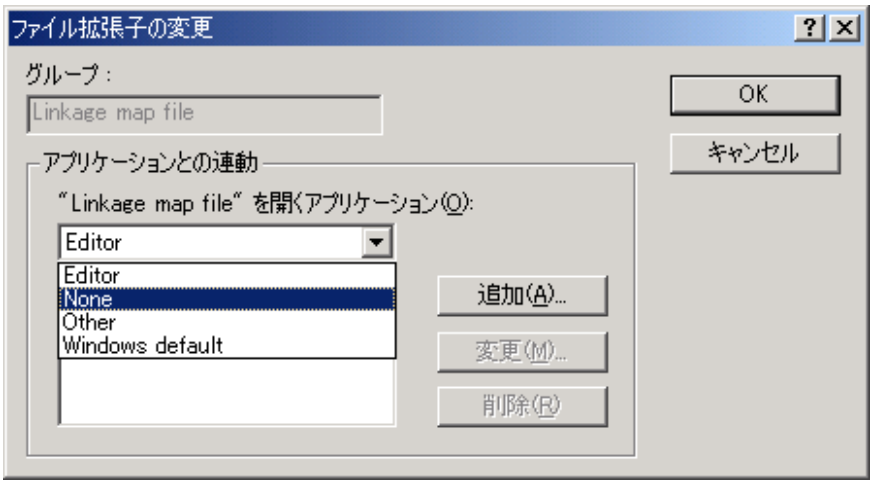


図 2.12: **ファイル拡張子の変更 ダイアログボックス**

3. 関連付けをしない場合は[None]を選びます。HEWのエディタまたは他のエディタでこの種類のファイルを開くには[Editor]を選びます。特定のアプリケーションでこの種類のファイルを開くには[Other]を選びます。

[Other]を選ぶと、すでに定義されたアプリケーションをドロップダウンリストから選ぶか、または、新しいアプリケーションを定義することができます。
新しいアプリケーションを定義する場合は[追加...]をクリックしてください。[アプリケーションの追加]ダイアログボックスが表示されます （図2.13）。
[名前]フィールドにアプリケーション名を入力します。[コマンド]フィールドにアプリケーションのフルパスを入力します(パラメータは含めません)。[パラメタ]フィールドにファイルを開くのに必要なパラメータを入力します。必ず$(FULLFILE) プレースホルダを使って入力ファイルを指定してください。 (プレースホルダの詳細と使用方法については、付録 C，「プレースホルダ」を参照してください。) [初期ディレクトリ]フィールドにアプリケーションを実行させる初期ディレクトリを入力します。[OK]ボタンをクリックするとアプリケーションが定義されます。

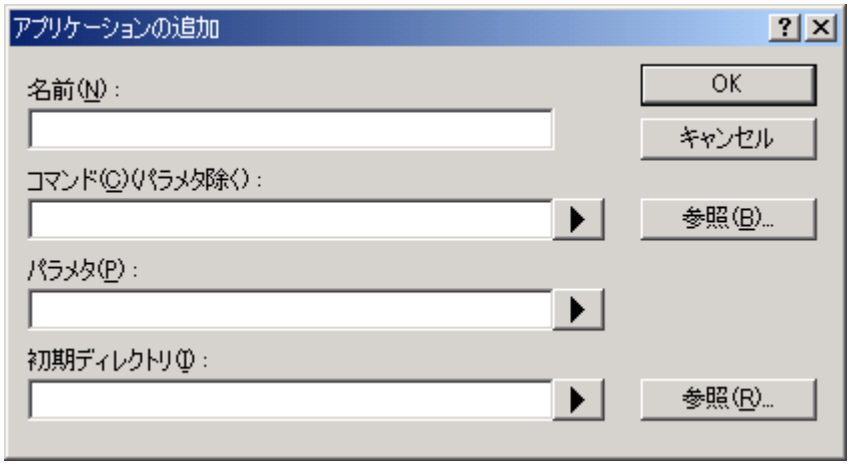


図 2.13: **アプリケーションの追加 ダイアログボックス**

アプリケーションを変更するには[変更…]ボタンを押してください。[アプリケーションの変更]ダイアログボックスが表示されます。このダイアログボックスは、[名前]フィールドが変更できないことを除いて[アプリケーションの追加]ダイアログボックスと同じです。設定を変更してから[OK]ボタンをクリックします。

4. [OK]ボタンをクリックすると選んだファイルグループに対するアプリケーションが関連付けられます。

2.4 ファイルのビルド方法を設定する

プロジェクトに必要なファイルを追加したら、次のステップは HEW に各ファイルのビルドを指示することです。このためには、[オプション]メニューからメニューオプションを選ばなければなりません。このメニューの内容は使用するツールによって異なります。

➲ ビルドフェーズにオプションを設定するには

1. オプションを変更するフェーズを[オプション]メニューから選んでください。
2. オプションを指定するダイアログボックスが表示されます。
3. オプションを設定して、[OK]ボタンをクリックしてください。

詳細は、コンテキスト依存のヘルプボタンをクリックするか、ヘルプを参照したい場所を選び"**F1**"キーを押下してください。

2.5 ビルドのコンフィグレーション

HEW では、ビルドのコンフィグレーションの中に、すべてのビルドのオプションを格納できます（図 2.14）。つまり、すべてのオプションを格納してそれらに名前を付けることができます。後にそのコンフィグレーションを選ぶと、各ビルドフェーズの各オプションを復帰させることができます。また、こうしたビルドのコンフィグレーションにより、ユーザはビルドのコンフィグレーション用にデバッガの設定を行うことができます。つまり、コンフィグレーションごとに異なるエンドプラットフォームをターゲットとすることができます。（詳細は、本マニュアルの「エミュレータ・デバッガ編」を参照してください。）

図 2.14 に"Default"、"MyDebug"、"MyOptimized"の 3 つのビルドコンフィグレーションを示します。"Default"ビルド構成では、各フェーズ（コンパイルとアセンブル）が標準設定されています。"MyDebug"ビルド構成では、各ファイルがデバッグ情報付きでビルドされています。"MyOptimized"ビルド構成では、各ファイルが最大限に最適化されデバッグ情報はありません。このプロジェクトの開発者は、オプションを設定するダイアログボックスに戻ってこれらを設定することなく、これらのビルドコンフィグレーションのうちどれでも選ぶことができます。

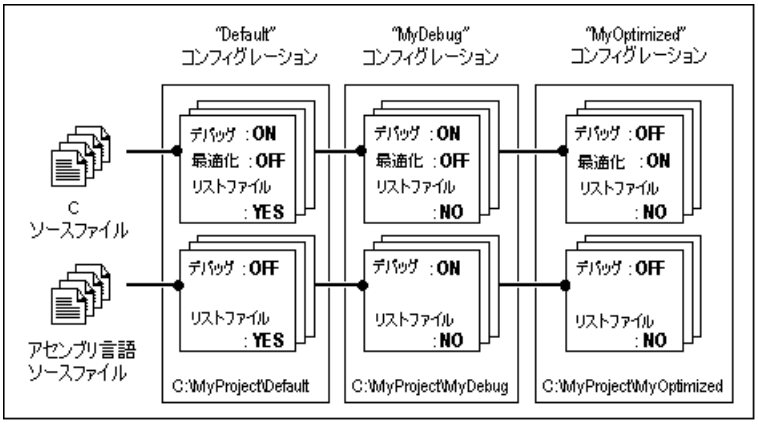


図 2.14: **ビルドコンフィグレーションとファイルオプション**

2.5.1 ビルドコンフィグレーションを選択する

使用するビルドコンフィグレーションを選択する方法は二つあります。

➲ ビルドコンフィグレーションを選択するには

1. ツールバーのドロップダウンリストボックス（図2.15）から選んでください。
2 ビルドコンフィグレーションが選ばれます。

図 2.15: **ツールバーの選択**

または

1. [オプション->ビルドの構成...]を選ぶと、[ビルドコンフィグレーション]ダイアログボックスが表示されます（図2.16）。

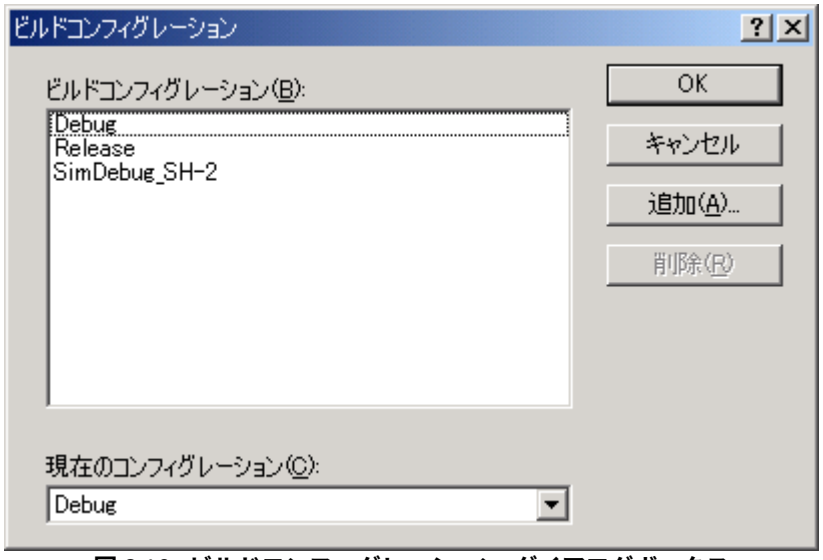


図 2.16: **ビルドコンフィグレーション ダイアログボックス**

2. [現在のコンフィグレーション]から使用するビルド構成を選んでください。
3. [OK]ボタンをクリックするとビルドコンフィグレーションが選ばれます。

2.5.2 ビルドコンフィグレーションを追加、削除する

ビルドコンフィグレーションの設定を他のビルドコンフィグレーションからコピーして追加したり、ビルドコンフィグレーションを削除したりできます。これらの操作を以下に説明します。

➲ ビルドコンフィグレーションを追加するには

1. [オプション->ビルドの構成...] を選ぶと[ビルドコンフィグレーション]ダイアログボックスが表示されます（図2.16）。
2. [追加...]ボタンをクリックすると[コンフィグレーションの追加]ダイアログボックスが表示されます （図2.17）。

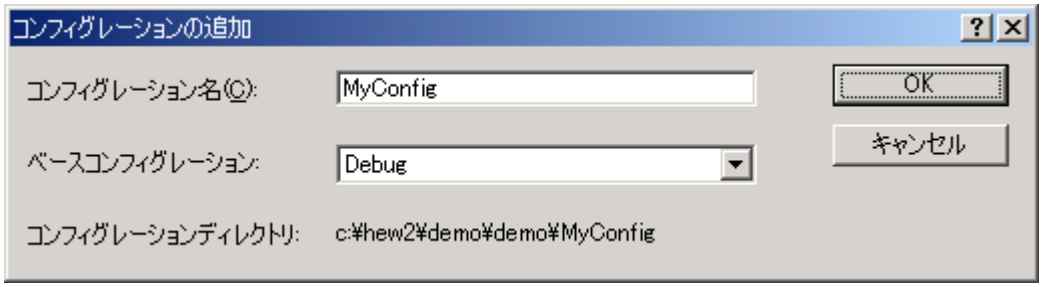


図 2.17: **コンフィグレーションの追加 ダイアログボックス**

3. [コンフィグレーション名]フィールドに新しいビルドコンフィグレーション名を入力してください。 入力すると、下に表示されるディレクトリがビルドコンフィグレーションに使われるディレクトリに変わります。[ベースコンフィグレーション]フィールドのドロップダウンリストにある既存コンフィグレーションの中から、コンフィグレーションの設定をコピーする元となるコンフィグレーションを選びます。両方のダイアログボックスの[OK]ボタンをクリックすると新しいビルドコンフィグレーションが作成されます。

➲ ビルドコンフィグレーションを削除するには

1. [オプション->ビルドの構成...] を選ぶと[ビルドコンフィグレーション]ダイアログボックスが表示されます（図2.16）。
2. 削除するビルドコンフィグレーションを選び[削除]ボタンをクリックしてください。
3. [OK]ボタンをクリックすると[ビルドコンフィグレーション]ダイアログボックスを閉じます。

2.6 プロジェクトをビルド実行する

ビルド実行の概要は図 2.1 を参照してください。

2.6.1 プロジェクトをビルド実行する

[ビルド]オプションでは前回のビルド実行後に変更のあったファイルだけをコンパイルまたはアセンブルします。さらに、前回のビルド実行以後に変更のあったファイルに依存するソースファイルを再ビルド実行します。例えば、 “test.c”にファイル “header.h”が含まれており“header.h”が前回のビルド実行以後に変更された場合、ファイル“test.c”が再コンパイルされます。

➲ ビルド実行するには

1. [ビルド->ビルド] を選ぶか、[ビルド]ツールバーボタン( ) をクリックするか、F7キーを押下してください。または、ワークスペースウィンドウの[Projects]タブのプロジェクトアイコン上で右マウスボタンをクリックし、ポップアップメニューから[ビルド]を選んでください。

[すべてをビルド]オプションでは変更の有無に関わらず、すべてのソースファイルをコンパイルまたはアセンブルして、新しく作成されたオブジェクトファイルをすべてリンクします。

➲ “すべてをビルド”処理を実行するには

1. [ビルド->すべてをビルド] を選ぶか、[すべてをビルド]ツールバーボタン ( )をクリックしてください。または、ワークスペースウィンドウの[Projects]タブのプロジェクトアイコン上で右マウスボタンをクリックし、ポップアップメニューの[ビルド]メニューからサブメニュー[すべてをビルド]を選んでください。

2.6.2 1 つのファイルをビルド実行する

プロジェクトにある 1 つのファイルをビルド実行できます。

➲ 1 つのファイルをビルド実行するには

1. プロジェクトウィンドウからビルド実行するファイルを選んでください。
2. **[ビルド->コンパイル]**を選ぶか、[ファイルのビルド]ツールバーボタン ( ) をクリックするか、**”CTRL+F7”**キーを押してください。または、ワークスペースウィンドウの[Projects]タブのファイルアイコン上で右マウスボタンをクリックして、ポップアップメニューから **[ビルド <ファイル名>]**を選んでください。

2.6.3 ビルド実行を中止する

ビルド実行を途中で中止できます。

➲ ビルド実行を中止するには

1. **[ビルド->ビルドの中止]** を選ぶか、“ビルド中止”ツールバーボタン ( )をクリックしてください。その時点のファイルのビルド実行を完了後、ビルド実行は中止されます。
2. アウトプットウィンドウに“Compiler Finished”というメッセージが表示されるのを確認してから操作を続けてください。

➲ ビルド実行中のツールを強制的に中止するには

1. [ビルド->ツールの終了]を選んでください。ビルダはツールの実行をすぐに中止します。

注意　中止したツールによって出力されたファイルは有効ではない場合があります。作成した出力ファイルをすべて削除して、そのフェーズを再実行してください。

2.6.4 複数のプロジェクトをビルド実行する

複数のプロジェクトやコンフィグレーションのビルド処理が行えます。

➲ 複数のプロジェクトをビルド実行するには

1. [ビルド->複数ビルド]を選択してください。図2.18に示します。
2. 複数のビルド実行では、どのプロジェクトまたはコンフィグレーションを処理するかを選択できます。どのプロジェクトまたはコンフィグレーションを実行するか選択するには、実行したいプロジェクトとコンフィグレーションの組み合わせの横にあるチェックボックスを選んでください。例えば、プロジェクト“hewtest2”全体にビルド実行したい場合、“hewtest2-Debug”、“hewtest2-Release”を選んでチェックし、その他のボックスのチェックを外してください。
3. 項目を選んだら、[ビルド]ボタンをクリックしてください。選んだプロジェクトやコンフィグレーションのビルド実行をHEWが行います。
4. また、選んだ項目に対して、全ビルド実行したい場合、[すべてをビルド]ボタンをクリックしてください。
5. 通常のビルド実行と同じ方法で、ビルドの結果がビルドウィンドウに表示されます。

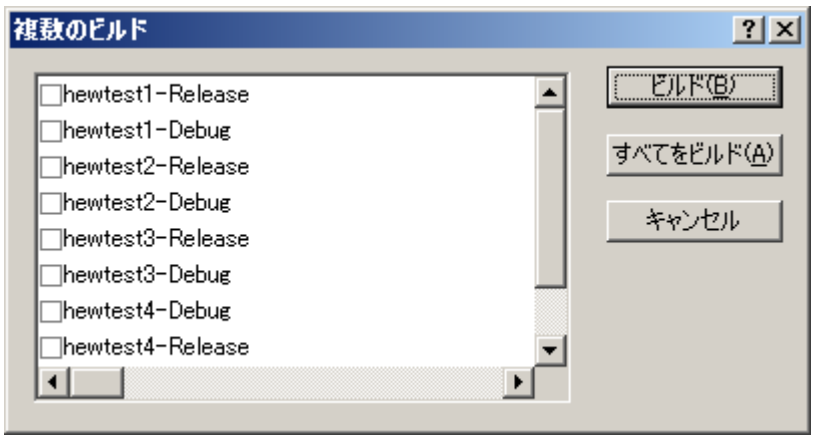


図 2.18　複数のビルドダイアログボックス

2.6.5 アウトプットウィンドウ

ツール（コンパイラ、アセンブラ、リンケージエディタなど）が実行されると、その出力がアウトプットウィンドウに表示されます。エラーまたはウォーニングが起きると、エラーメッセージまたはウォーニングメッセージと、ソースファイル名と行番号が表示されます。すぐにエラーまたはウォーニングが発生したところをエディタで見るには、表示されたエラーメッセージまたはウォーニングメッセージをダブルクリックします。

2.6.6 アウトプットウィンドウの内容の制御

ビルド実行の途中にビルド実行情報（ファイルに適用したコマンドラインオプションなど）を表示することができます。HEW では、“ビルド”、 “すべてをビルド”、 “ファイルのビルド”処理中、アウトプットウィンドウにそのオプションを表示するかどうかを[オプション]ダイアログボックスで指定できます。

➲ ビルド実行中にビルド実行情報の表示の有無を指定するには

1. **[ツール->オプション...]**を選ぶと[オプション]ダイアログボックスが表示されます。
2. [ビルド]タブ（図2.19）を選んでください。
3. [表示]グループのチェックボックスを以下のように設定します。 [コマンドライン]にはツール実行時のコマンドライン表示の有無を指定します。[環境]にはツール実行時の環境変数の表示の有無を指定します。[初期ディレクトリ]にはツールが起動されるディレクトリパスの表示の有無を指定します。

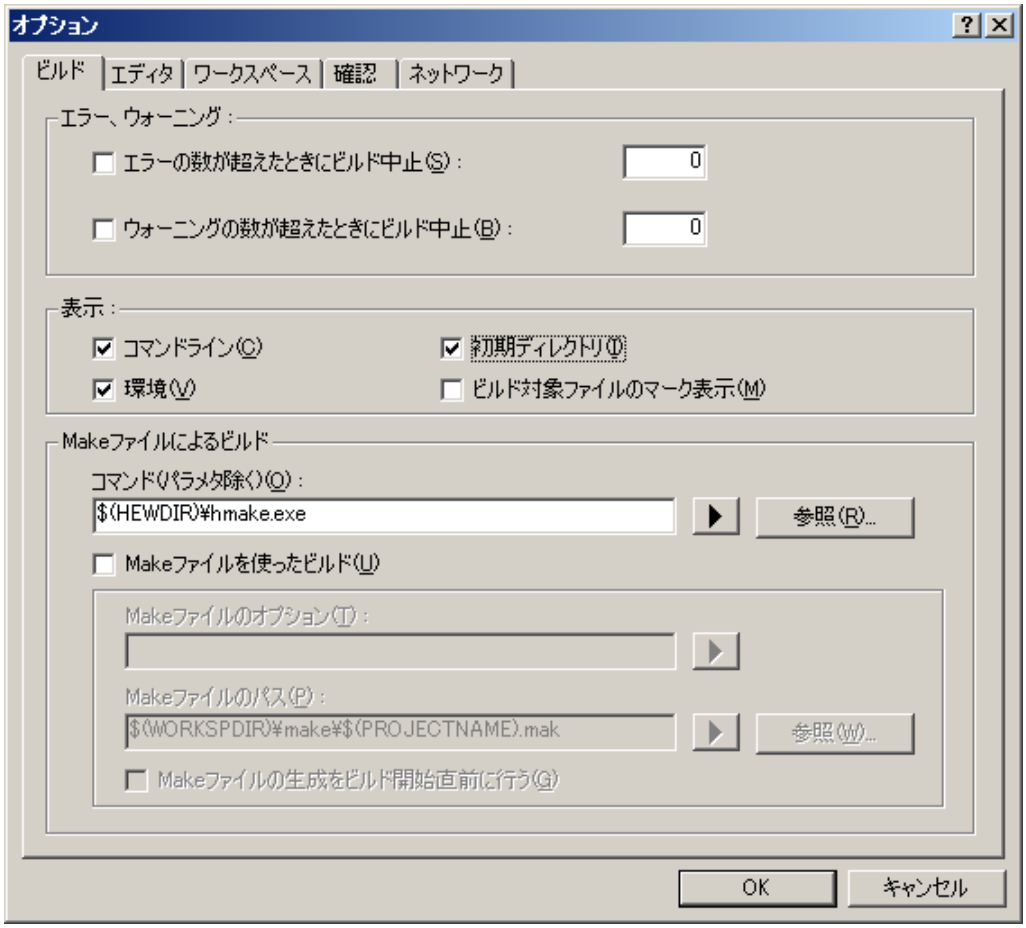


図 2.19: **オプションダイアログボックスビルドタブ**

2.6.7 ビルド対象ファイルをワークスペースウィンドウにマーク表示する

前回のビルドで生成されたファイルより後に更新されたファイル（ビルド対象ファイル）をワークスペースウィンドウ上にマーク表示します。図 2.20 では、ビルド対象ファイルは"demo.c"です。

次に[ビルド]をクリックしたとき、このファイルがビルドされます。アクティブプロジェクトの依存プロジェクトのファイルにもマーク表示します。

これらファイルのマーク表示は、ビルドに影響することが起こるたびに更新します。例えば、オプションの変更、ファイルの追加、依存関係の変更、ファイルの修正などです。

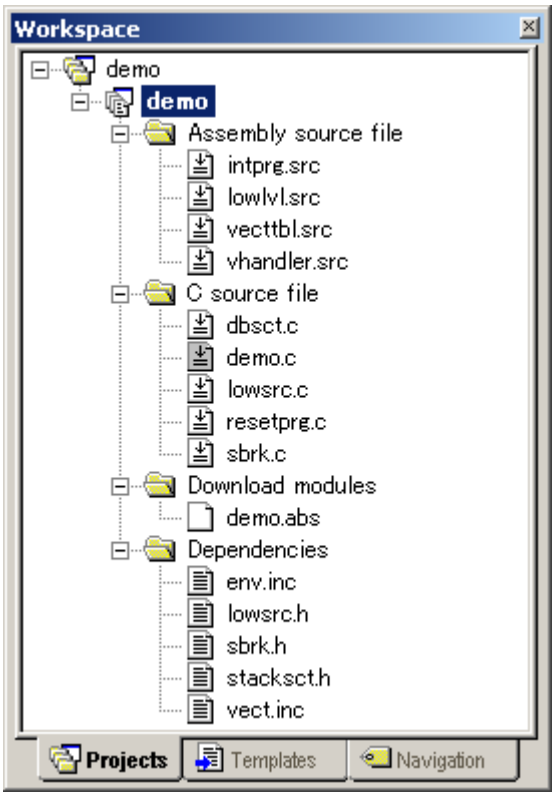


図 2.20: **ビルド対象ファイルのマーク表示**

➲ ビルド対象ファイルをワークスペースウィンドウにマーク表示するには

1. **[ツール->オプション...]**を選ぶと[オプション]ダイアログボックスが表示されます。
2. [ビルド]タブ（図2.19）を選んでください。
3. [ビルド対象ファイルのマーク表示]チェックボックスをチェックしてください。
4. [OK]ボタンをクリックしてください。

2.7 ファイル依存関係

多くの場合プロジェクトにはファイル間の依存関係があります。例えば、1 つの C ファイルはいくつかのヘッダファイルをインクルードします。複雑なプロジェクトでは、ソースファイルが他のインクルードファイルに依存するため、管理が複雑になります。しかし、HEW にはファイル依存関係をスキャンする機能があり、そのプロジェクトにあるすべてのファイルの依存関係をチェックできます。

スキャンが完了すると、プロジェクトのファイル依存関係を示す最新のリストをプロジェクトウィンドウに表示します。

➲ プロジェクトのファイル依存関係を更新するには

1. **[ビルド->すべての依存関係を更新]**を選んでください。または、ワークスペースウィンドウの[Projects]タブのプロジェクトアイコンを右マウスボタンでクリックし、ポップアップメニューの**[ビルド]**から**[すべての依存関係を更新]**サブメニューを選んでください。

最初に、すべての依存ファイルは[Dependencies]フォルダに表示されています（図 2.21.(i)）。

2.8 ワークスペースウィンドウの構成

ワークスペースウィンドウの[Projects]タブの中を右マウスでクリックすると、ポップアップメニューが表示されます。その中から、[表示の構成...]メニューオプションを選び、以下の情報の表示方法を設定してください。 以下に、[表示の構成]ダイアログボックスの各オプションについて説明します。

2.8.1 各ファイルの下に依存を表示する

[依存関係をファイルの下に表示]を選ぶと、依存ファイルがそれをインクルードするソースファイルの下に平坦な構造で表示されます（ファイル自体がフォルダになります）。これを図 2.21.(ii)に示します。このオプションを選ばないと、別のフォルダにすべての依存ファイルを示します（図 2.21.(i)）。

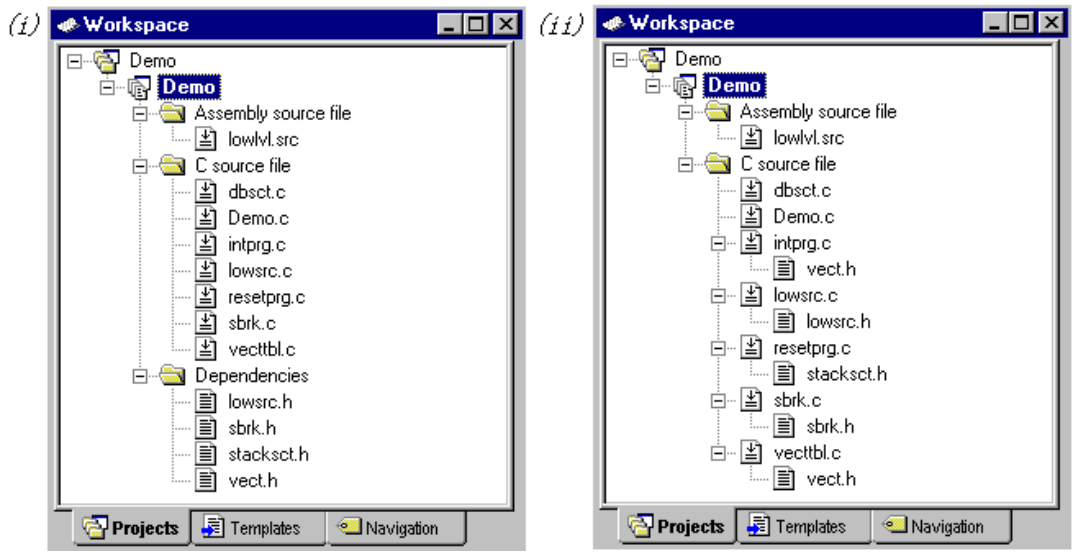


図 2.21: **各ファイルの依存**

2.8.2 標準ライブラリファイルのインクルードを表示する

デフォルトでは、標準インクルードパスの依存ファイルは表示されません（図 2.22.(i)）。例えば、C コードで#include <stdio.h>などのインクルード文を書くと、“stdio.h”は依存ファイルとして表示されません。そのようなシステムインクルードファイルを表示するには、[インクルードする標準ライブラリの表示]オプションを選んでください（図 2.22.(ii)）。

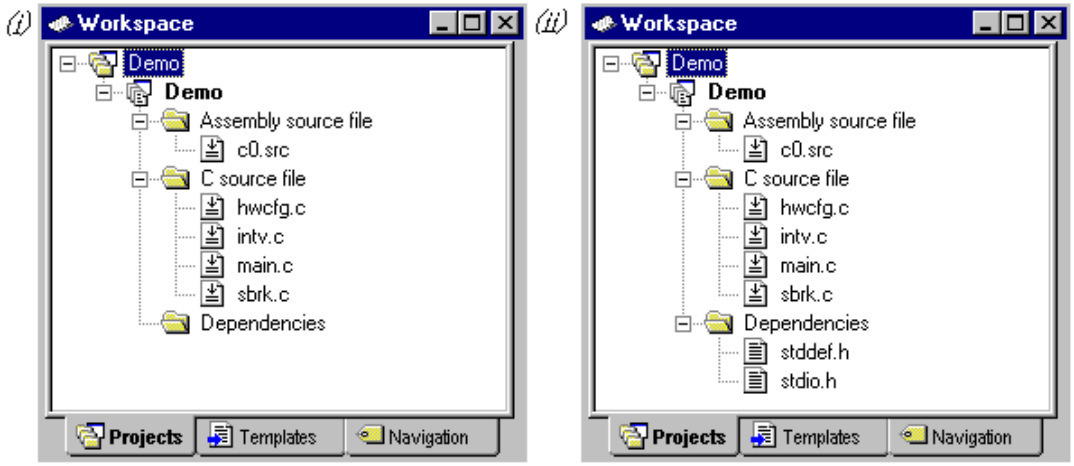


図 2.22: **標準ライブラリファイルのインクルード**

2.8.3 ファイルのパスを表示する

[ファイルパスの表示]を選ぶと、ワークスペースウィンドウのすべてのファイルがフルパス（ドライブ名からのパス）で表示されます （図 2.23）。

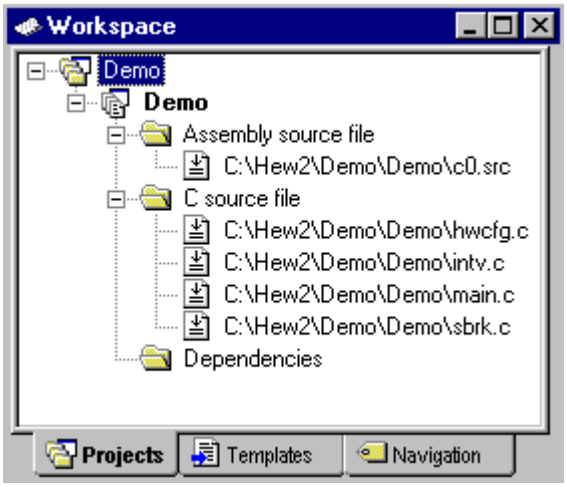


図 2.23: **ファイルのパスの表示**

2.8.4 ファイルをタイムスタンプの順序でソートする

[ファイルのタイムスタンプによるソート]を選ぶと、ワークスペースウィンドウのファイルがタイムスタンプの順序にソートして表示されます。

このオプションを設定した後でファイルを更新した場合は、手動で表示順を更新しなければいけません。

➲手動で表示順を更新するには

1. ワークスペースウィンドウの[Projects]タブのポップアップメニューから**[リフレッシュ]**を選んでください。

2.9 アクティブプロジェクトを設定する

ワークスペースには複数のプロジェクトを含めることができますが、ひとつだけがアクティブです。このアクティブプロジェクトでビルド動作とデバッグ動作が実行されます。そして、そのプロジェクト用のビルダやデバッガのオプションを変更できます。また、アクティブプロジェクトは太字で表示されます。

➲ プロジェクトをアクティブにするには

1. ワークスペースウィンドウの [Projects]タブからアクティブでないプロジェクトを選んでください。
2. マウスの右ボタンをクリックしてポップアップメニューを表示させ、**[アクティブプロジェクトに設定]** オプションを選んでください。

または

1. **[プロジェクト->アクティブプロジェクトに設定]** サブメニューからプロジェクトを選んでください。

2.10 ワークスペースにプロジェクトを追加する

ワークスペースを作成したとき、最初はプロジェクトが一つしかありません。しかし、後で、新しいプロジェクトや既存のプロジェクトを追加することができます。

➲ ワークスペースに新しいプロジェクトを追加するには

1. **[プロジェクト->プロジェクトの挿入...]**を選んでください。 [プロジェクトの挿入] ダイアログボックスが表示されます（図2.24）。
2. [新規プロジェクト] オプションを設定してください。
3. [OK]ボタンをクリックしてください。 [新規プロジェクトの挿入] ダイアログボックスが表示されます。
4. [名前] フィールドにプロジェクト名を入力してください。32文字以内で、半角英数字、半角下線が入力できます。プロジェクト名を入力すると、HEWは自動的にサブディレクトリを追加します。これは不要であれば削除できます。
5. [参照...]ボタンをクリックしてプロジェクトを作成するディレクトリを選んでください。または、 [ディレクトリ] フィールドにディレクトリを入力できます。
6. [プロジェクトタイプ] リストには使用可能なプロジェクトの種類を示します（アプリケーション、ライブラリなど）。このリストから作成するプロジェクトの種類を選んでください。
7. [OK]ボタンをクリックすると、プロジェクトが作成されワークスペースに追加されます。

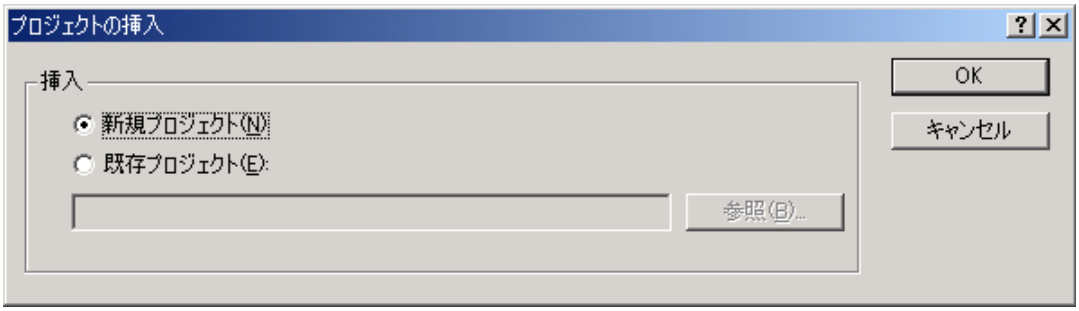


図 2.24: **プロジェクトの挿入ダイアログボックス**

➲ 既存のプロジェクトをワークスペースに追加するには

1. **[プロジェクト->プロジェクトの挿入...]**を選んでください。[プロジェクトの挿入] ダイアログボックスが表示されます。
2. [既存プロジェクト] オプションを設定してください。
3. プロジェクトデータベースファイル（.HWPファイル）へのフルパスを入力するか、 [参照...]ボタンをクリックしてプロジェクトデータベースファイルを指定してください。
4. [OK]ボタンをクリックするとそのプロジェクトがワークスペースに追加されます。

2.11 プロジェクト間の依存関係を指定する

ワークスペースのプロジェクトは、他のプロジェクトに依存することができます。ビルド処理をすると、依存プロジェクトが最初にビルドされます。これは、ワークスペースのプロジェクトを他のプロジェクトが使用しているときなどに使用します。例えば、ワークスペースに 2 つのプロジェクトがあるとします。1 つはアプリケーションプロジェクトに含まれたライブラリだとします。この場合 2 番目のアプリケーションのビルド前にライブラリは正確にビルドされ、また最新でなくてはなりません。そのため、ライブラリをアプリケーションプロジェクトの依存プロジェクトに指定します。こうすると、最新でないライブラリが先にビルドされます。

依存プロジェクトをビルドするとき、HEW は依存プロジェクトがアクティブプロジェクトのビルドコンフィグレーションになるようにします。上記の例では、アクティブプロジェクトのビルドコンフィグレーションが “Debug” であるとき、HEW は、依存プロジェクトで“Debug” ビルドコンフィグレーションが選択されるようにします。このような一致したコンフィグレーションが存在しない場合、HEW は依存プロジェクトで最近使われたコンフィグレーションを使用します。

➲ 依存プロジェクトを指定するには

1. **[プロジェクト->依存プロジェクト]**を選んでください。 [依存プロジェクト] ダイアログボックスが表示されます（図2.25）。
2. 依存させたいプロジェクトを選んでください。 [依存プロジェクト] リストに（選択したプロジェクト以外の）ワークスペース内のすべてのプロジェクトが表示されます。
3. [依存プロジェクト]リストには各プロジェクトにチェックボックスがあります。選んでプロジェクトが依存するプロジェクトのチェックボックスをチェックしてください。
4. [OK]ボタンをクリックしてください。

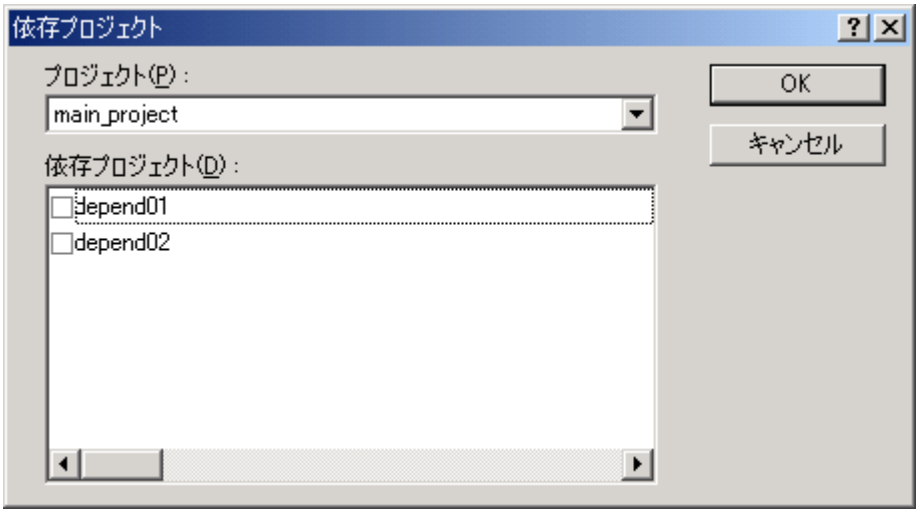


図 2.25: **依存プロジェクトダイアログボックス**

2.12 ワークスペースからプロジェクトを削除する

➲ ワークスペースからプロジェクトを削除するには

1. ワークスペースウィンドウの [Projects] タブを選び、右マウスボタンでポップアップメニューを表示してください。
2. **[プロジェクトの削除]** オプションを指定してください。
3. ダイアログボックスが表示され、この操作を誤って行っていないことを確認します。[ツール]メニューの[オプション]ダイアログボックスでは、この確認をオフにすることができます。

注意　ワークスペースからアクティブプロジェクトを削除することはできません。

2.13 ワークスペースのプロジェクトをロード、アンロードする

➲ ワークスペースのプロジェクトをロードするには

1. ワークスペースウィンドウの [Projects] タブから、アンロードしたプロジェクトを選んでください。
2. マウスの右ボタンをクリックしてポップアップメニューを表示させ、[プロジェクトのロード]オプションを選んでください。

➲ ワークスペースのプロジェクトをアンロードするには

1. ワークスペースウィンドウの [Projects] タブから、アクティブでないプロジェクトを選んでください。
2. マウスの右ボタンをクリックしてポップアップメニューを表示させ、[プロジェクトのアンロード]オプションを選んでください。

注意　一度に複数のプロジェクトを選び、それらをすべてロード、またはアンロードすることができます。

2.14 ワークスペースの相対プロジェクトパス

HEW では、プロジェクトをワークスペースに追加する場合、相対パスを使用してワークスペースに追加することが選択できます。これにより、プロジェクトをワークスペースディレクトリの上に置くことができ、HEW ワークスペースの再配置も正確にできます。プロジェクトは常にワークスペースと相対的であるため、プロジェクトがワークスペースの上のディレクトリにある場合、再配置のあと、HEW は、同じ相対場所でプロジェクトを見つけようとします。このことは、複数のワークスペース間で共有したプロジェクトを使用するとき、特に便利です。

HEW の古いバージョンでは、このプロジェクトは再配置されておらず、オリジナルのプロジェクトパスをアクセスしようとしました。また、ワークスペースディレクトリのサブディレクトリにあったプロジェクトを再配置することのみ可能でした。これは現在のバージョンでも、HEW の標準的な動作です。

➲ プロジェクト相対パスフラグを変更するには

1. ワークスペースウィンドウでプロジェクトを選択してください。
2. マウスの右ボタンをクリックしてプロパティを選択してください。
3. [プロジェクト相対パス]チェックボックスをクリックし、相対ファイルパスの特徴を切り替えてください（図2.26）。
4. [OK]ボタンをクリックしてください。

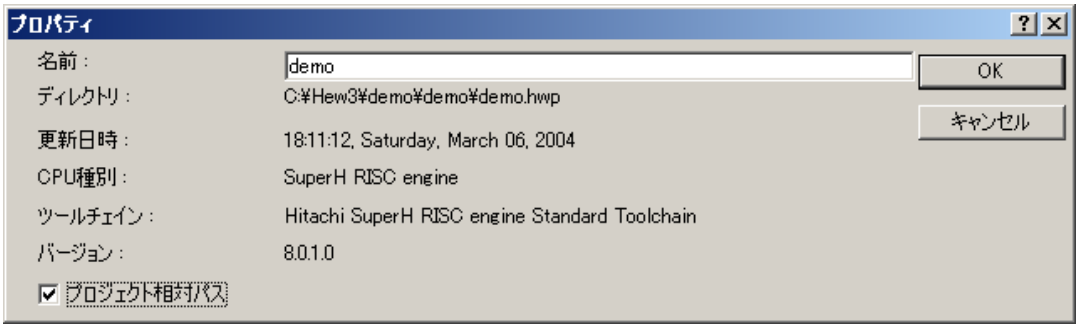


図 2.26: **プロパティダイアログボックス**

2.15 ワークスペース内のユーザフォルダ

HEW では、ユーザのワークスペースウィンドウにフォルダを追加することができます。これにより、プロジェクト内の特定の領域にファイルを論理的に分類することが可能です。フォルダ名として任意の名前をダイアログボックスで入力します。

➲ ユーザフォルダを追加するには

1. ワークスペースウィンドウの[Projects]タブでプロジェクトを選択してください。
2. マウスの右ボタンをクリックし、[フォルダの追加...]を選択してください。
3. 名前を入力して[OK]ボタンをクリックしてください。
4. ファイルをドラッグ＆ドロップして、論理的に分類することができます。

➲ ユーザフォルダを削除するには

1. ワークスペースウィンドウの[Projects]タブでフォルダを選択してください。
2. マウスの右ボタンをクリックし、[フォルダの削除]を選択してください。このときフォルダは空である必要があります。

➲ ユーザフォルダ名を変更するには

1. ワークスペースウィンドウの[Projects]タブで名前を変更したいフォルダを選択してください。
2. マウスの右ボタンをクリックし、[フォルダ名の変更]を選択してください。
3. ダイアログで新しい名前を入力してください。
4. [OK]ボタンをクリックしてください。

3. ビルドの応用

この章ではより高度なビルドの概念を説明します。

3.1 ビルド実行の復習

2 章「ビルドの基本」では、ビルド実行をコンパイラ、アセンブラ、リンケージエディタを用いて説明しました（図 2.1）。HEW ではこれが一般的な実行環境です。しかし、ビルド実行を変更（フェーズの追加や削除など）するためには、ビルドの機能についてさらに知ることが必要です。

3.1.1 ビルドとは?

プロジェクトのビルド実行とは、複数の特定の入力ファイルに複数のツールを適用して期待する出力を得ることです。つまり、オブジェクトファイルを得るために、C/C++ソースファイルにコンパイラを適用したりアセンブリ言語のソースファイルにアセンブラを適用したりします。ビルドの各ステップ、または、各「フェーズ」において、様々な入力ファイルの集まりに各種ツールを適用します。図 3.1 にビルド処理を示します。

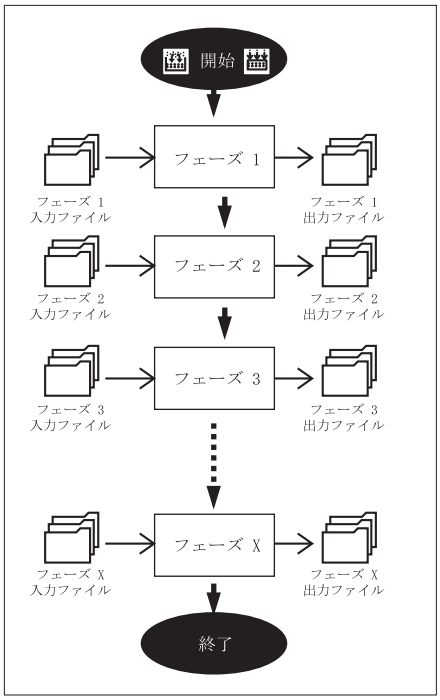


図 3.1: ビルド実行

HEW ではビルド処理を変更できます。 **[オプション->ビルドフェーズ...]**を選んで[ビルドフェーズ]ダイアログボックス （図 3.2） を表示します。左には現在のプロジェクトで定義されたフェーズを示します（図 3.2 では標準のビルドフェーズを示します）。この章では[ビルドフェーズ] ダイアログボックスが提供する様々な機能について説明します。

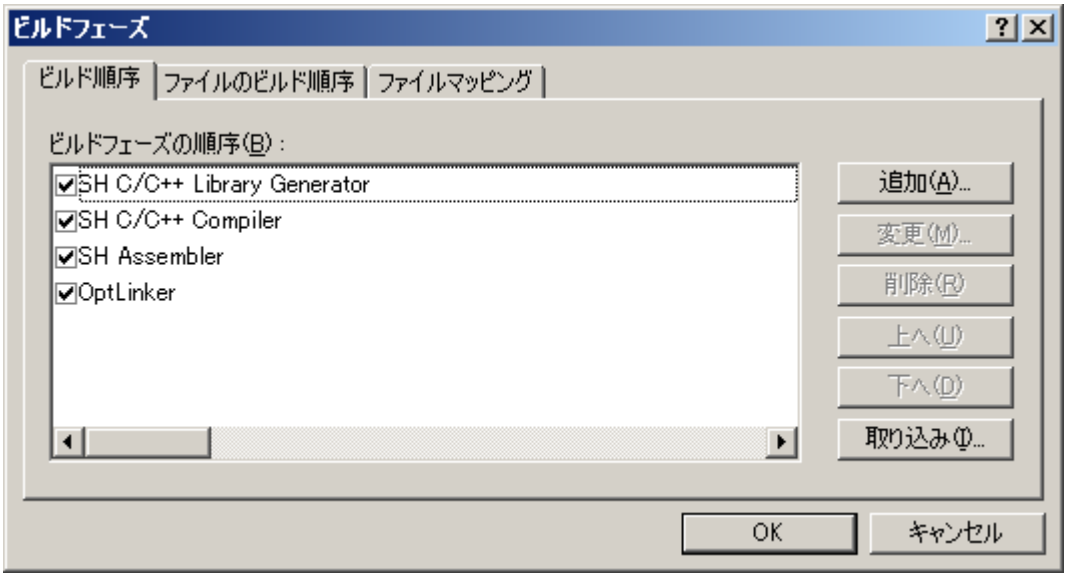


図 3.2: **ビルドフェーズ ダイアログボックス**

3.2 カスタムビルドフェーズを作成する

標準のビルド実行の前後または途中に他のツールを実行する場合、独自のビルドフェーズ（カスタムビルドフェーズ）を作成します。カスタムビルドフェーズを略してカスタムフェーズということがあります。

[オプション->ビルドフェーズ...] を選ぶと [ビルドフェーズ] ダイアログボックス （図 3.2） が表示されます。 [追加...] ボタンをクリックしてください。新しいビルドフェーズを作成するための[新規ビルドフェーズ]ダイアログボックス（図 3.3a）が表示されます。

ステップ 1（図 3.3a）では、カスタムビルドフェーズを新規に作成するか、システムビルドフェーズを追加するかを選択します。システムビルドフェーズは、使用しているツールチェイン（コンパイラ、アセンブラ、リンケージエディタ、ライブラリアンなど）内で定義済みのすぐに使用できるフェーズ、または、ユーティリティフェーズ（例えば、ファイルコピー、ソースコード複雑度解析ツールなど）です。システムフェーズがこれ以上ない場合、[既存のシステムフェーズの追加]ボタンが非アクティブになります。（システムビルドフェーズを略してシステムフェーズといいます。）

[新規カスタムフェーズの作成]を選んでカスタムビルドフェーズを作成してください。

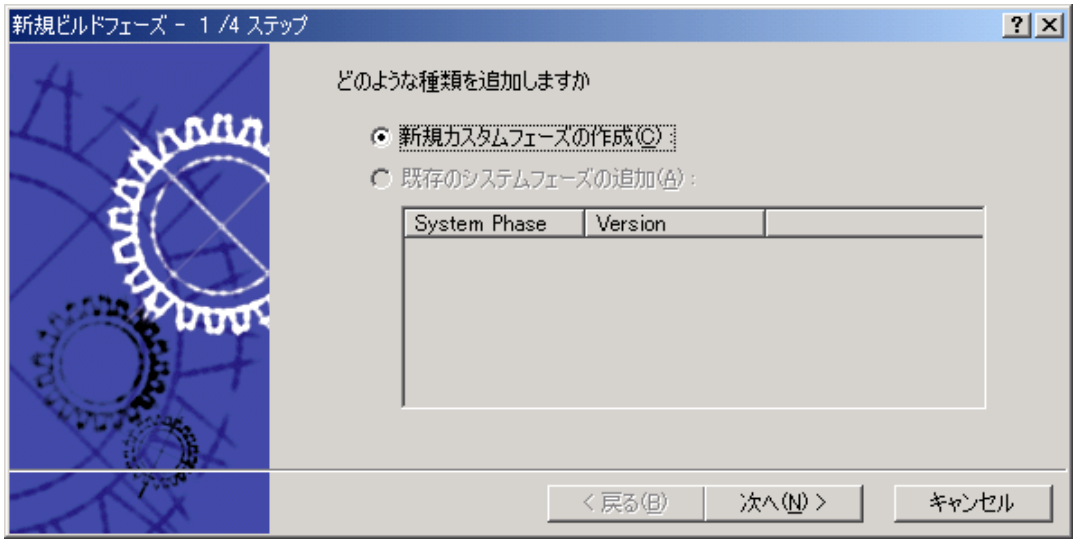


図 3.3a: **新規ビルドフェーズ ダイアログボックス (ステップ** 1)

ステップ 2（図 3.3b）では作成するフェーズの種類を選びます。2 つの選択肢（ [複数フェーズ]または [単一フェーズ]）があります。複数フェーズを実行すると特定のファイルグループに属するプロジェクト内の各ファイルにコマンドが適用されます。例えば、[入力ファイルの選択]フィールドに[C source file]を選ぶと、プロジェクト内の各ファイルに 1 回ずつコマンドが実行されます。単一フェーズを選ぶとビルド実行中に一度だけ実行されます。

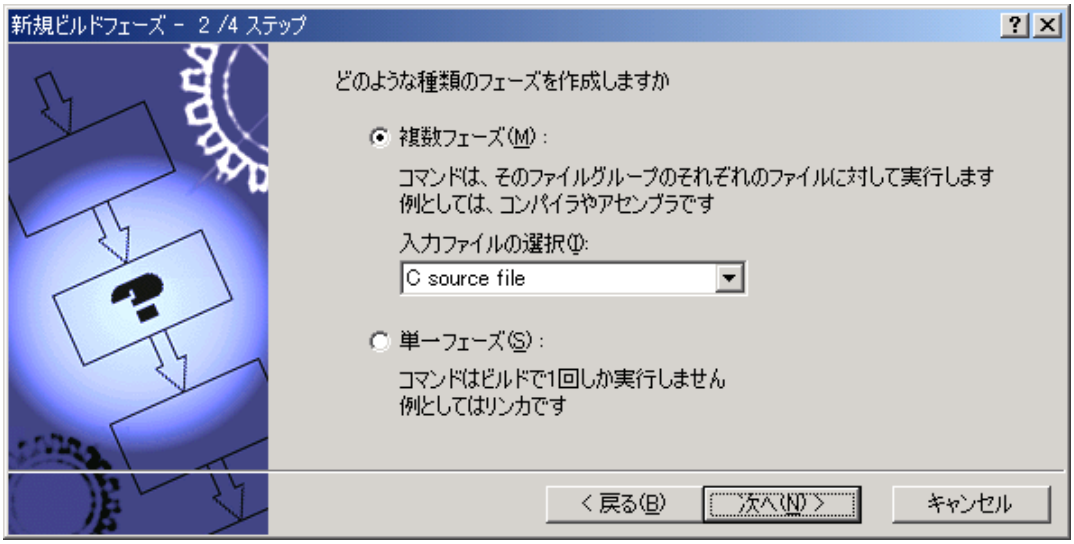


図 3.3b: **新規ビルドフェーズ ダイアログボックス（ステップ** 2)

入力ファイルグループリストは、そのプロジェクト用に定義された現在のファイルグループを含みます。入力ファイルグループリストの中の[Multiple Groups…]エントリを選択すると、複数の入力ファイルグループを定義することができます。このリストのエントリは、図 3.3c に示すダイアログボックスを表示します。

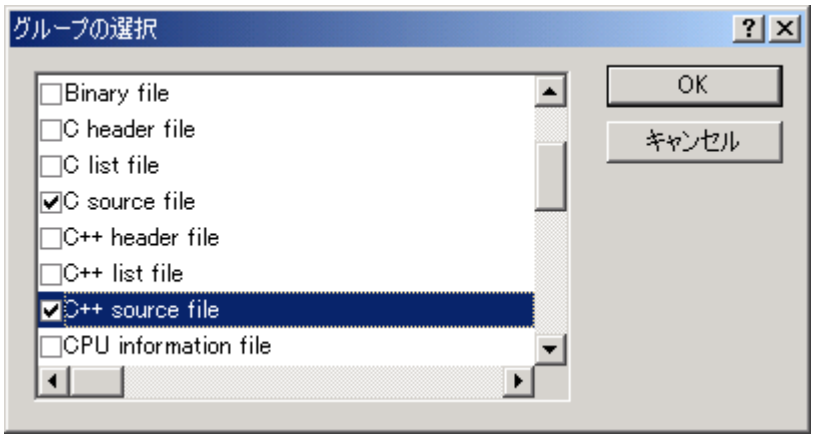


図 3.3c: **複数の入力ファイルグループの変更**

複数選択すると、入力ファイルグループは[Multiple Groups…]と表示されます。プロジェクトに追

加されているカスタムフェーズ用に、複数の入力ファイルグループをこのダイアログボックスで選ぶことができます。ファイルグループを選択するには、ファイルグループ名の隣にあるボックスをチェックしてください。このダイアログボックスでは、１つ以上のファイルグループが選択できます。

ステップ 3（図 3.3d）では、新しいビルドフェーズについての基礎的な情報を入力します。 [フェーズ名] フィールドにフェーズ名を入力します。 [コマンド] フィールドにプログラムファイルのパスを入力します （コマンドラインオプションは含めません。オプションは HEW のメニューバーの[オプション]メニューで指定します）。[デフォルトオプション]フィールドにフェーズのデフォルトのオプションを指定します。デフォルトオプションはプロジェクトに新しいファイルを追加するときに付加されるオプションです。[初期ディレクトリ] フィールドにはそのプログラムをどのディレクトリから実行するか（つまり、ツール実行前にどこにカレントディレクトリを設定するか）を入力します。

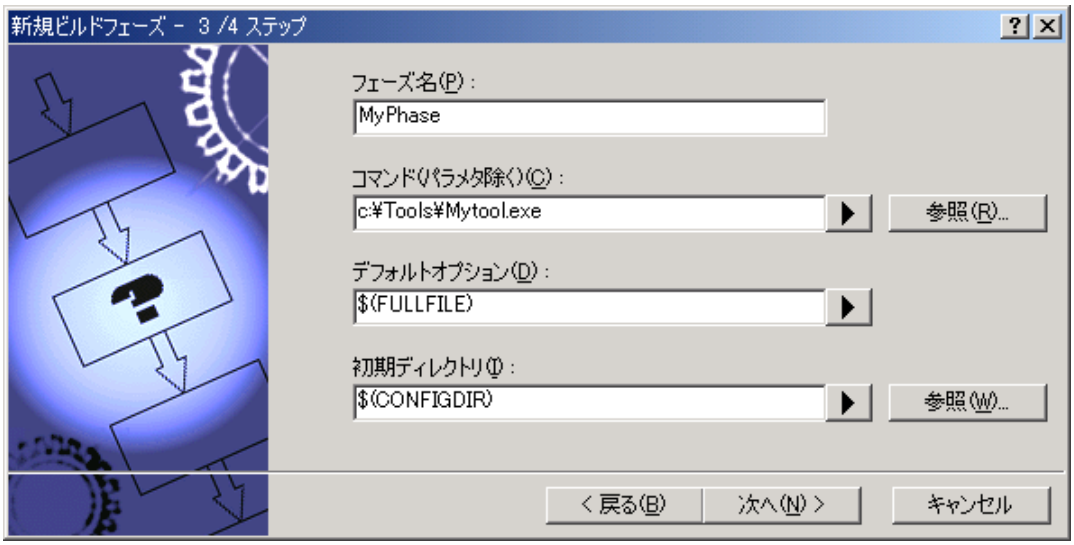


図 3.3d: **新規ビルドフェーズ ダイアログボックス（ステップ** 3)

最後のステップ 4（図 3.3e）では、そのフェーズに必要な環境変数を指定します。

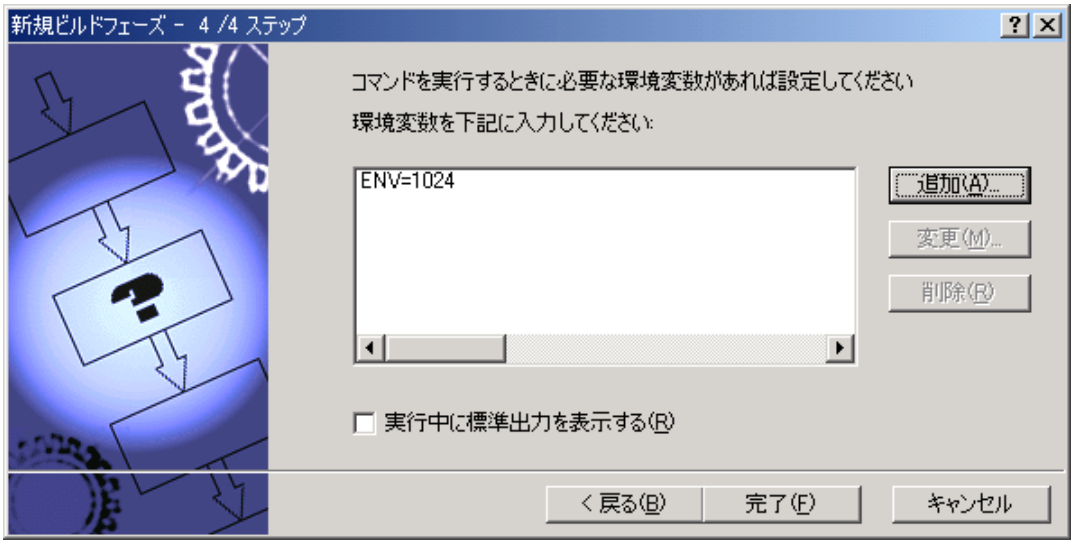


図 3.3e: **新規ビルドフェーズ ダイアログボックス (ステップ** 4)

新しい環境変数を追加するには、[追加...] ボタンをクリックしてください。図 3.4 のダイアログボックスが表示されます。新しい環境変数を追加するには[変数] フィールドに環境変数名を入力して [値] フィールドに環境変数の値を入力して [OK]ボタンをクリックします。

環境変数を変更するには、ステップ 4 のダイアログボックスのリストから環境変数を選んで、[変更...] ボタンをクリックします。 [変数] フィールドと [値] フィールドを変更して [OK]ボタンをクリックすると、リストに変更した変数が追加されます。

環境変数を削除するには、ステップ 4 のダイアログボックスのリストから削除する環境変数を選んで、[削除] ボタンをクリックします。

図 3.4: **環境変数 ダイアログボックス**

追加するツールが動作中に出力を表示できる場合、ステップ 4 のダイアログボックスの[実行中に標準出力を表示する]オプションを使用してください。出力が発生するごとにツールの出力を表示します。このオプションが off に設定されると、HEW はツールに表示されている全出力を保存し、ツールが動作を終了したときアウトプットウィンドウに表示します。ただ、ツールが長時間かかる作業を実行中である場合、実行の進行状況を見るのが難しいため、このオプションは問題となることがあります。

注意　[実行中に標準出力を表示する]を用いると、特定のオペレーティングシステムで特定のツールを使用するとき問題を引き起こすことがあります。もしツールが HEW の中でロックアップ、またはフリーズするといった問題がありましたら、[実行中に標準出力を表示する]オプションのチェックを外してください。

指定した内容で新しいフェーズを作るには、 [完了] ボタンをクリックしてください。デフォルトでは[ビルドフェーズ] ダイアログボックス（図 3.2）の[ビルド順序]タブの[ビルドフェーズの順序]リストの最後に新しいフェーズが追加されます。

3.3　ビルドのフェーズ順序

図 3.5 の標準的ビルド処理では、①コンパイラの前、②アセンブラの前、③リンケージエディタの前、④リンケージエディタの後、の四ヶ所にフェーズを追加できます。ビルドの順序の中で好きな場所にカスタムフェーズを追加したりシステムフェーズを移動したりできます。ビルド処理を正しく実行させるためには、カスタムフェーズの出力が他のフェーズに入力される場合、フェーズの順序を正しく設定することが必要です。

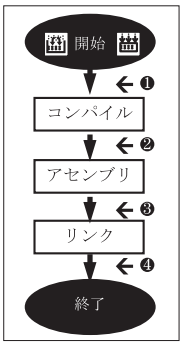


図 3.5: 標準的ビルド処理

[ビルドフェーズ] ダイアログボックスでビルドフェーズの順序を変更できます。このダイアログボックスにはフェーズの順序に関するタブが二つあります。 [ビルド順序] タブと[ファイルのビルド順序]タブです。変更後 [OK]ボタンをクリックしてください。

3.3.1 ビルドのフェーズ順序

[ビルド順序] タブ(図 3.6)は、[ビルド] ()または[すべてをビルド] () 操作で実行されるフェーズの現在の順序を示します。各フェーズの左にあるチェックボックスのチェックの有無はそのフェーズの有効/無効を示します。このチェックボックスをチェックするとそのフェーズが実行されます。

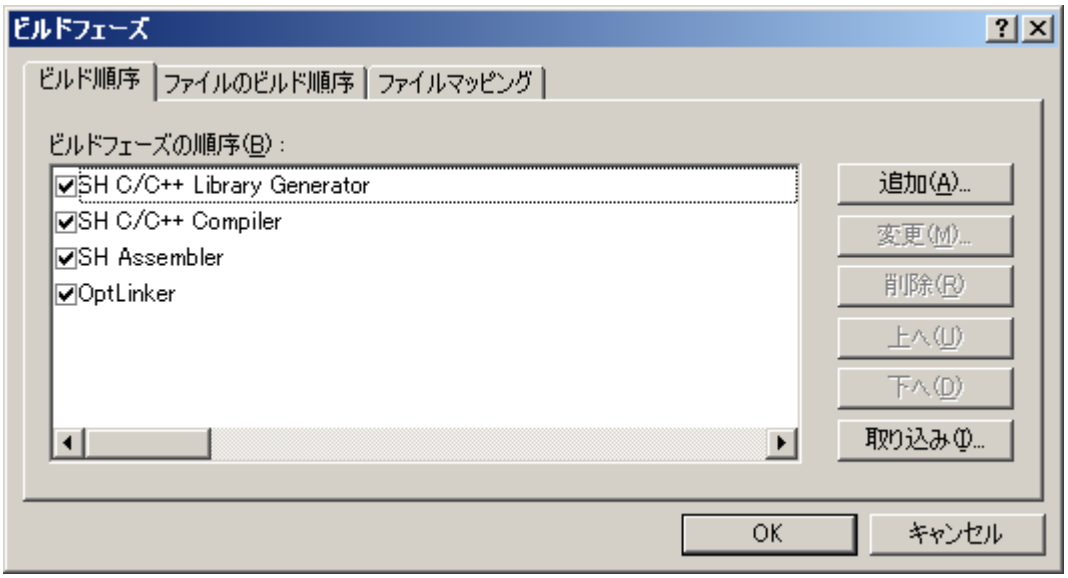


図 3.6: **ビルドフェーズ ダイアログボックス ビルド順序タブ**

また、以下の操作ができます。

➲ フェーズを削除するには

1. 削除するフェーズを選んでください。
2. [削除] ボタンをクリックしてください。
3. 確認ダイアログボックスが表示された場合は、 [はい] ボタンをクリックしてください。

➲ システムフェーズのプロパティを表示するには

1. プロパティを表示するシステムフェーズを選んでください。
2. [変更...]ボタンをクリックしてください。
3. ダイアログボックスで、システムフェーズのプロパティが表示されます。[OK]ボタンをクリックしてください。

➲ フェーズを移動するには

1. 移動するフェーズを選んでください。
2. [上へ] または [下へ] ボタンをクリックすると上下に移動します。

➲ フェーズを取り込むには

1. [取り込み] ボタンをクリックしてください。ダイアログボックスが表示され、カスタムフェーズを取り込むための既存のプロジェクトを見ることができます。
2. カスタムフェーズを取り込みたいプロジェクトの位置を選んでください。選択すると、取り

込み可能なプロジェクトのカスタムフェーズを並べたダイアログボックスが表示されます。

3 どのフェーズを取り込むか決定したら、そのフェーズをリスト上でハイライト表示し、[OK] ボタンをクリックしてください。ビルド順序で一番下にあるビルドフェーズダイアログボックスに、そのフェーズが追加されます。

➲ カスタムフェーズを変更するには

1. 変更するカスタムフェーズを選んでください。
2. [変更…] ボタンをクリックしてください。[MyPhaseの変更] ダイアログボックスの[コマンド] タブが表示されます（図3.7）。
3. 必要に応じてフィールドの内容を変更してください。
4. 入力ファイルがなくてもフェーズの実行を中断したくない場合、 [実行前に入力ファイルが存在するか否かのチェックを行わない] チェックボックスをチェックしてください。
   標準出力への出力をコマンド実行中に表示したい場合は、[実行中に標準出力を表示する] チェックボックスをチェックしてください。

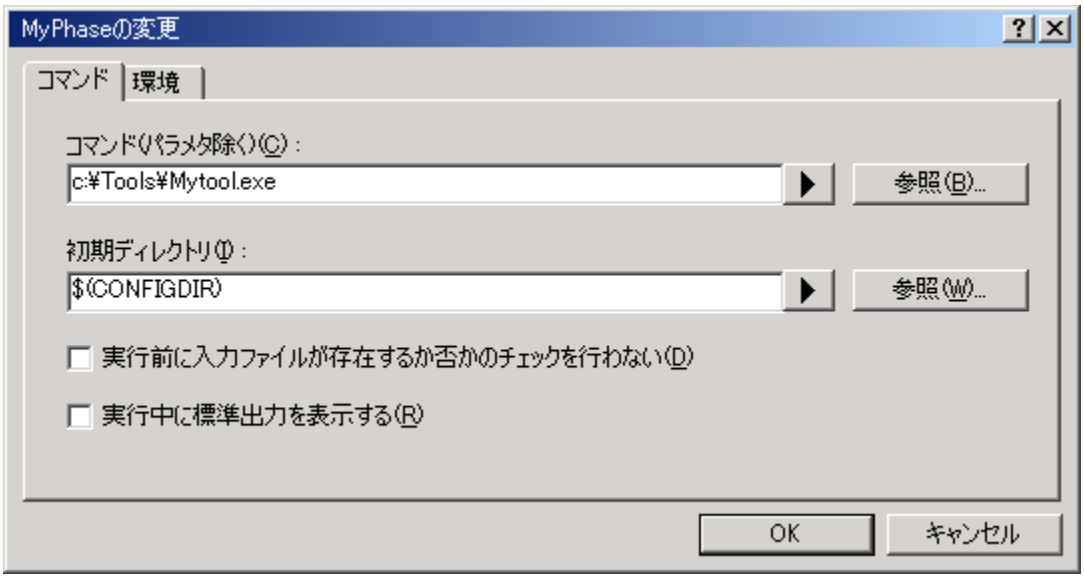


図 3.7: MyPhase **の変更 ダイアログボックスコマンド タブ**

5. [環境] タブ（図3.8）を選んでフェーズの環境設定を行ってください。
6. 環境変数の追加は [追加…] ボタン、変更は [変更…] ボタン、削除は [削除] ボタンを使用してください。操作は前節と同じです。
7. 変更後 [OK] ボタンをクリックしてください。

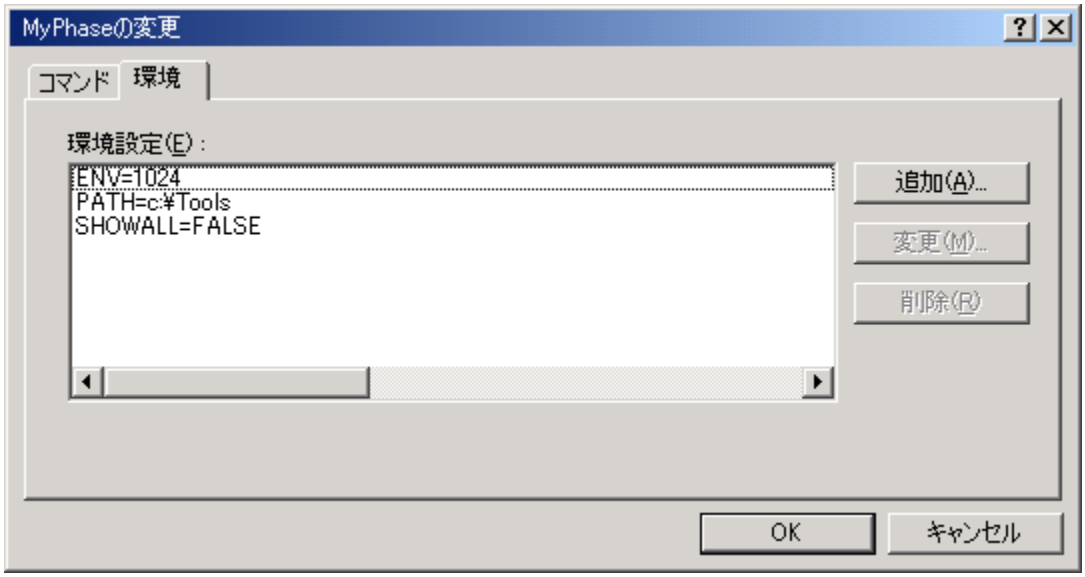


図 3.8: MyPhase **の変更 ダイアログボックス環境 タブ**

3.3.2 ビルドファイルのフェーズ順序

ワークスペースウィンドウから C ソースファイルを選んで**[ビルド->コンパイル]**を選ぶか、を押すと、ファイルがコンパイルされます。同じように、ワークスペースウィンドウからアセンブリ言語ソースファイルを選んで**[ビルド->コンパイル]**を実行すると、ファイルがアセンブルされます。ファイルグループと実行するフェーズの関係は[ビルドフェーズ]ダイアログボックスの [ファイルのビルド順序] タブ（図 3.9）で管理されています。

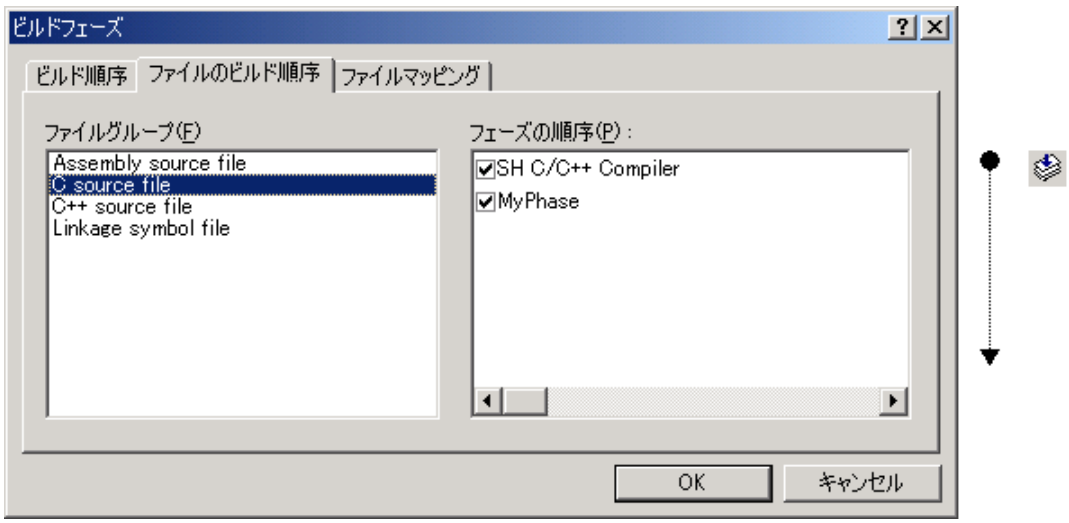


図 3.9: **ビルドフェーズ ダイアログボックス ファイルのビルド順序 タブ**

リストには、[ファイルグループ] リストボックスのファイルグループに対してビルドファイル操

作を選択したとき実行されるすべての現在のフェーズを表示します。図 3.9 では“C source file” ファイルグループが選ばれており、 “Compiler” フェーズと “MyPhase” フェーズが関連付けされています。

[ビルド順序]タブのリストに新しいエントリを追加すると、自動的に[ファイルのビルド順序]タブの[フェーズの順序]リストに新しいエントリが追加されます。例えば、“C source file”を入力とするフェーズを追加します。このフェーズは“ファイルのビルド”操作を“C source file”に適用する時に実行されるフェーズのリストに自動的に追加されます。**[ビルド->コンパイル...]**を選んだときに実行したくないフェーズがある場合、[フェーズの順序]リストのフェーズ名の左にあるチェックボックスのチェックを外してください。

3.4 カスタムビルドフェーズのオプション設定

カスタムフェーズを定義後、フェーズ実行時に使用するコマンドラインオプションを指定します。定義されたフェーズにはそれぞれ **[オプション]**メニューにオプション設定用のメニューがあります。指定するフェーズのオプションを選んでください。起動するダイアログボックスは、選んだカスタムフェーズが単一フェーズか複数フェーズかによって異なります（図 3.3b [単一フェーズ]/[複数フェーズ]ラジオボタン指定）。

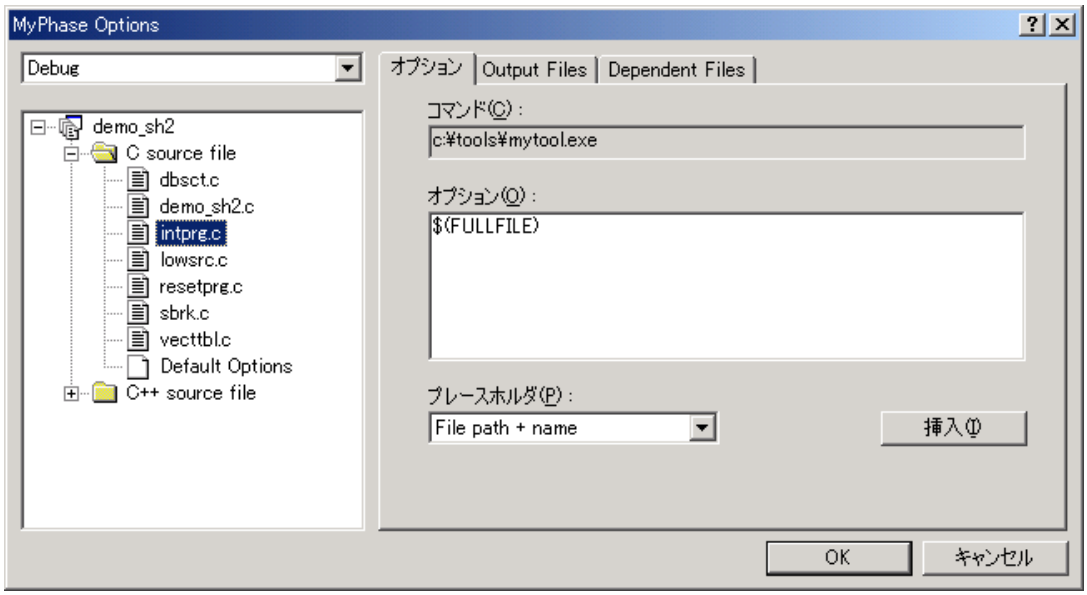


図 3.10: **カスタムフェーズのオプションダイアログボックス（複数フェーズ）**

図 3.10 にカスタムフェーズオプションダイアログボックス（複数フェーズ）を示します。左側にはプロジェクトとファイルのリストがあります。Windows® Explorer と同様の方法で、オプションを変更するプロジェクトとファイルを複数、一回以上選ぶことができます。右側には 3 つのオプションタブがあります。選んだファイルに適用するオプションをここで設定してください。

また、ダイアログボックス左上のコンフィグレーションリストで、どのコンフィグレーションの情報を表示するか選択できます。各コンフィグレーションは[Multiple configurations...]という名のエントリと一緒に並べられています。[Multiple configurations...]を選択すると、ダイアログボックスが表示され、複数のコンフィグレーションを選択できます。

3.4.1 オプションタブ

[Options] タブ（図 3.11）ではフェーズに渡されるコマンドラインオプションを定義できます。 [コマンド] フィールドではフェーズを定義したときに入力したコマンドを表示します （図 3.3d）。 [オプション] フィールドにはコマンドに渡すコマンドライン引数を入力してください。プレースホルダを挿入する場合は、対応するプレースホルダを [プレースホルダ] ドロップダウンリストボックスから選び、 [挿入] ボタンをクリックしてください。プレースホルダの詳細については、付録 C, 「プレースホルダ」を参照してください。

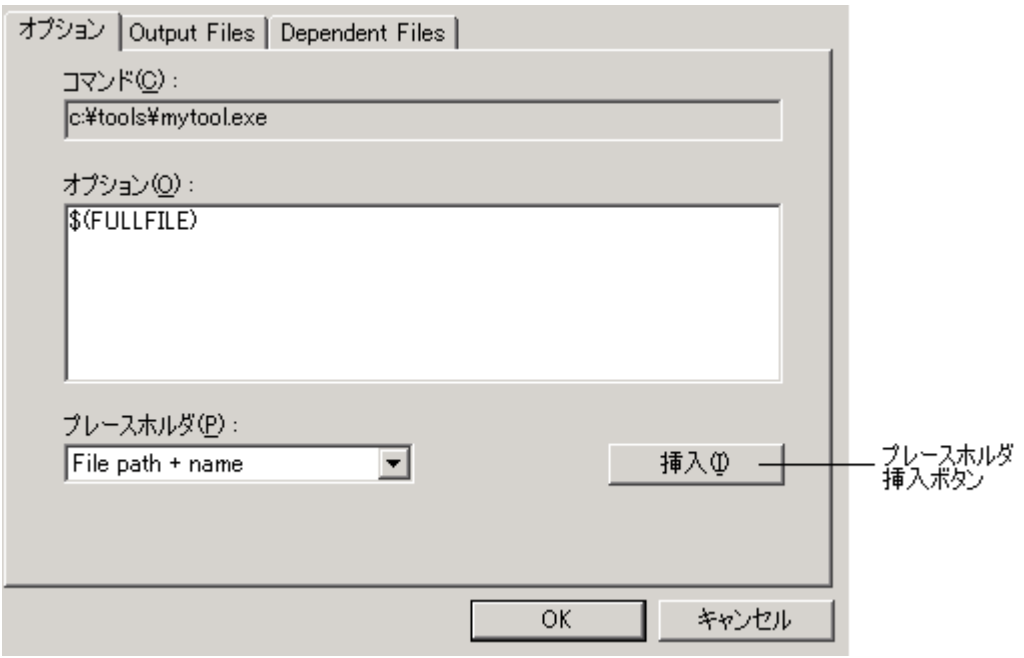


図 3.11: **カスタムフェーズのオプション オプション タブ**

3.4.2 Output Files タブ

“Output Files” タブ（図 3.12）ではフェーズで作成される出力ファイルを指定します。HEW では、ファイルがこのフェーズを通過する前に、出力ファイルの日付が入力ファイルの日付より古いことをチェックしています。出力ファイル作成後入力ファイルが変更された場合、入力ファイルに対してこのフェーズが実行されます。出力ファイルが最新の場合、入力ファイルに対してこのフェーズは実行されません。

注意　出力ファイルを指定しない場合フェーズは常に実行されます。

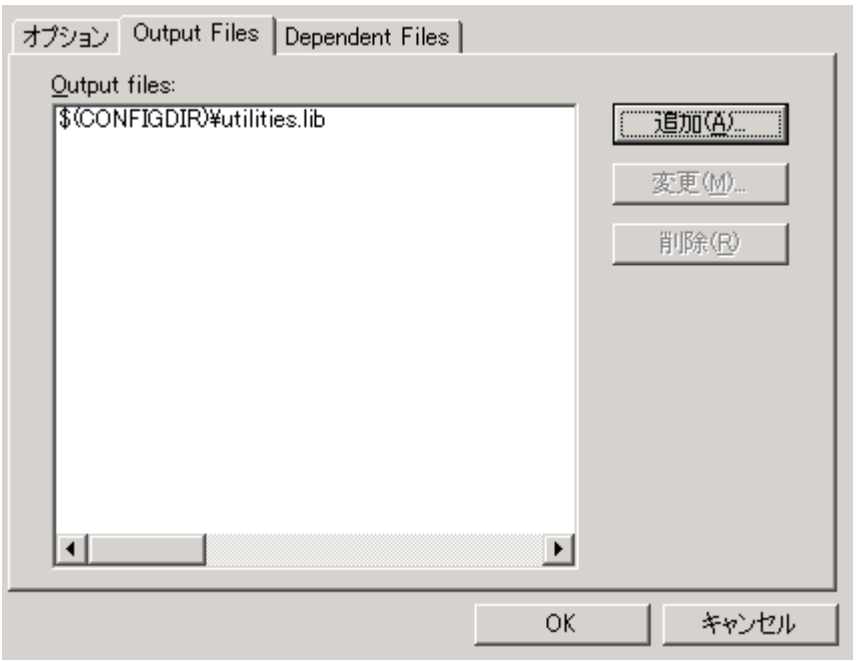


図 3.12: **カスタムフェーズのオプション** Output Files **タブ**

➲ 出力ファイルを追加するには

1. [追加…] ボタンをクリックしてください。[Add Output File] ダイアログボックスが表示されます（図3.13）。
2. ファイル名を入力するか、 [参照…] ボタンで選んでください。
3. [OK] ボタンをクリックすると、リストに出力ファイルを追加します。

図 3.13: Add Output File **ダイアログボックス**

➲ 出力ファイルを変更するには

1. 変更する出力ファイルを選んでください。
2. [変更…] ボタンをクリックすると [Modify Output File] ダイアログボックスが表示されます（タイトル以外は図3.13と同様）。
3. フィールドを変更して [OK] ボタンをクリックしてください。変更した項目がリストに追加されます。

➲ 出力ファイルを削除するには

1. 削除する出力ファイルを選んでください。
2. [削除] ボタンをクリックしてください。

3.4.3 Dependent Files タブ

“Dependent Files” タブ（図 3.14）ではフェーズに必要な依存ファイルを指定します。HEW では、各ファイルがこのフェーズを通過する前に、依存ファイルの日付が入力ファイルの日付より新しいか否かをチェックしています。チェック後、依存ファイルの日付が新しい場合（つまり、入力ファイル作成後に依存ファイルが変更された場合）このフェーズでファイルが実行されます。依存ファイルの日付が入力ファイルの日付より古い場合、このフェーズは実行されません。

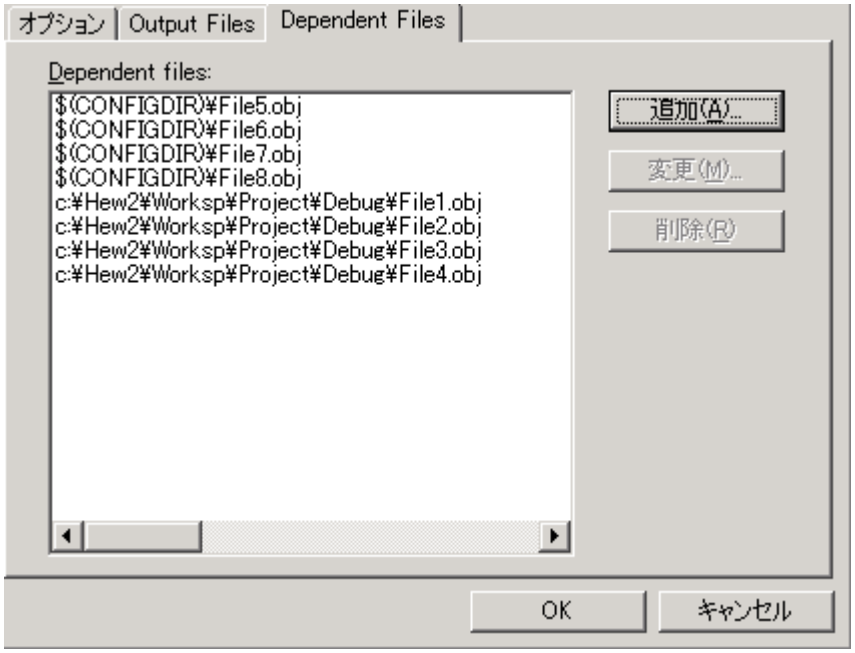


図 3.14: **カスタムフェーズのオプション** Dependent Files **タブ**

➲ 依存ファイルを追加するには

1. [追加…] ボタンをクリックしてください。[Add Dependent File]ダイアログボックスが表示されます（図3.15）。
2. ファイル名を入力するか、 [参照…] ボタンで選んでください。
3. [OK] ボタンをクリックすると、リストに出力ファイルを追加します。

図 3.15: Add Dependent File **ダイアログボックス**

➲ 依存ファイルを変更するには

1. 変更する依存ファイルを選んでください。
2. [変更…] ボタンをクリックすると [Modify Dependent File] ダイアログボックスが表示されます（タイトル以外は図3.15と同じ）。
3. フィールドを変更して [OK] ボタンをクリックすると変更した項目がリストに追加されます。

➲ 依存ファイルを削除するには

1. 削除するファイルを選んでください。
2. [削除] ボタンをクリックしてください。

3.5 ファイルのマッピング

デフォルトでは、フェーズに入力されるファイルはプロジェクトから取得したのものだけです。つまり、[新規ビルドフェーズ]ダイアログボックス（図 3.3b）の[入力ファイルの選択] ドロップダウンリストに指定した種類のプロジェクトファイルだけです。もし前のフェーズから出力されたファイル（中間ファイル）をフェーズで使いたい場合は、[ビルドフェーズ] ダイアログボックス （図 3.16）の[ファイルマッピング] タブで定義してください。

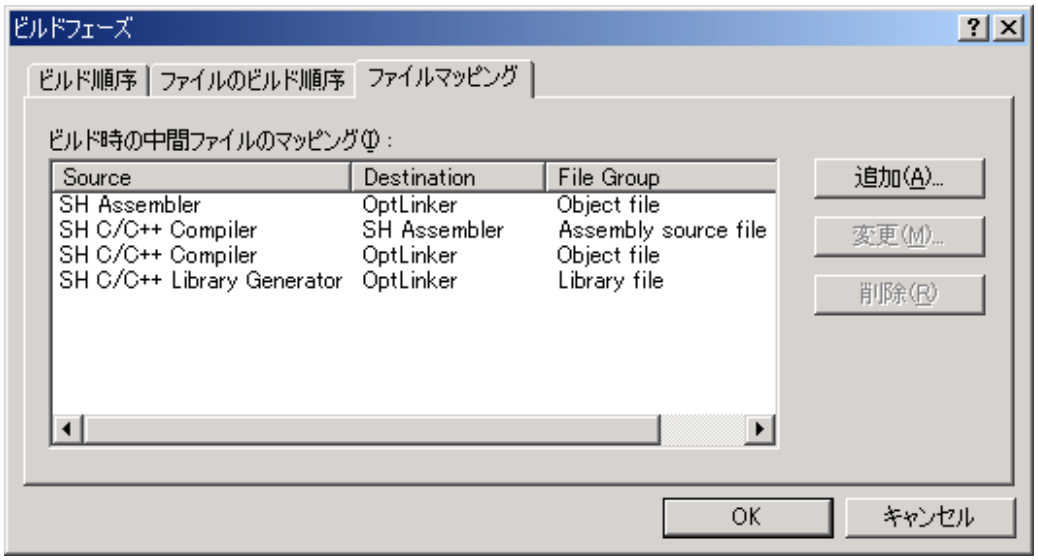


図 3.16: **ビルドフェーズ ダイアログボックス ファイルマッピング タブ**

ファイルマッピングでは、あるフェーズ（出力元フェーズという）で作成したある種類の出力ファイル（中間ファイル）を、他のフェーズ（出力先フェーズという）に渡すように指定します。プロジェクトファイルに加えて中間ファイルも渡されます。

➲ ファイルのマッピングを追加するには

1. [追加...] ボタンをクリックしてください。 [ファイルマッピングの設定] ダイアログボックスが表示されます（図3.17）。
2. [ファイルグループ] ドロップダウンリストボックスから、フェーズ間で渡したいファイルグループを選んでください。
3. [フェーズ元] ドロップダウンリストボックスから出力元フェーズ（ファイルを作成するフェーズ）を選んでください。
4. [フェーズ先] ドロップダウンリストボックスから出力先フェーズ（ファイルを渡す先のフェーズ）を選んでください。
5. [OK] ボタンをクリックすると新しいマッピングが追加されます。

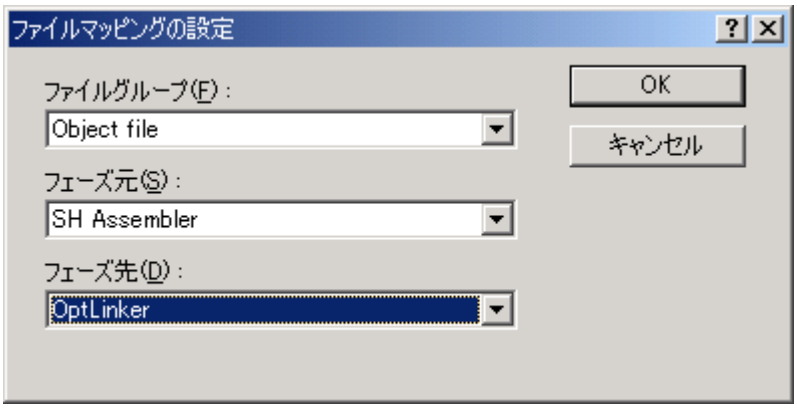


図 3.17: **ファイルマッピングの設定 ダイアログボックス**

➲ ファイルマッピングを変更するには

1. 変更するマッピングを選んでください。
2. [変更...] ボタンをクリックしてください。 [ファイルマッピングの設定] ダイアログボックスが表示されます（図3.17）。
3. 必要に応じてオプションを変更してください。
4. [OK] ボタンをクリックすると変更が有効になります。

3.6　ビルドを管理する

デフォルトでは、HEW はビルドのすべてのフェーズを実行し、途中で致命的なエラーが起こったときだけ中止します。これは [オプション] ダイアログボックスの [ビルド]タブ（図 3.18）で変更することができます。

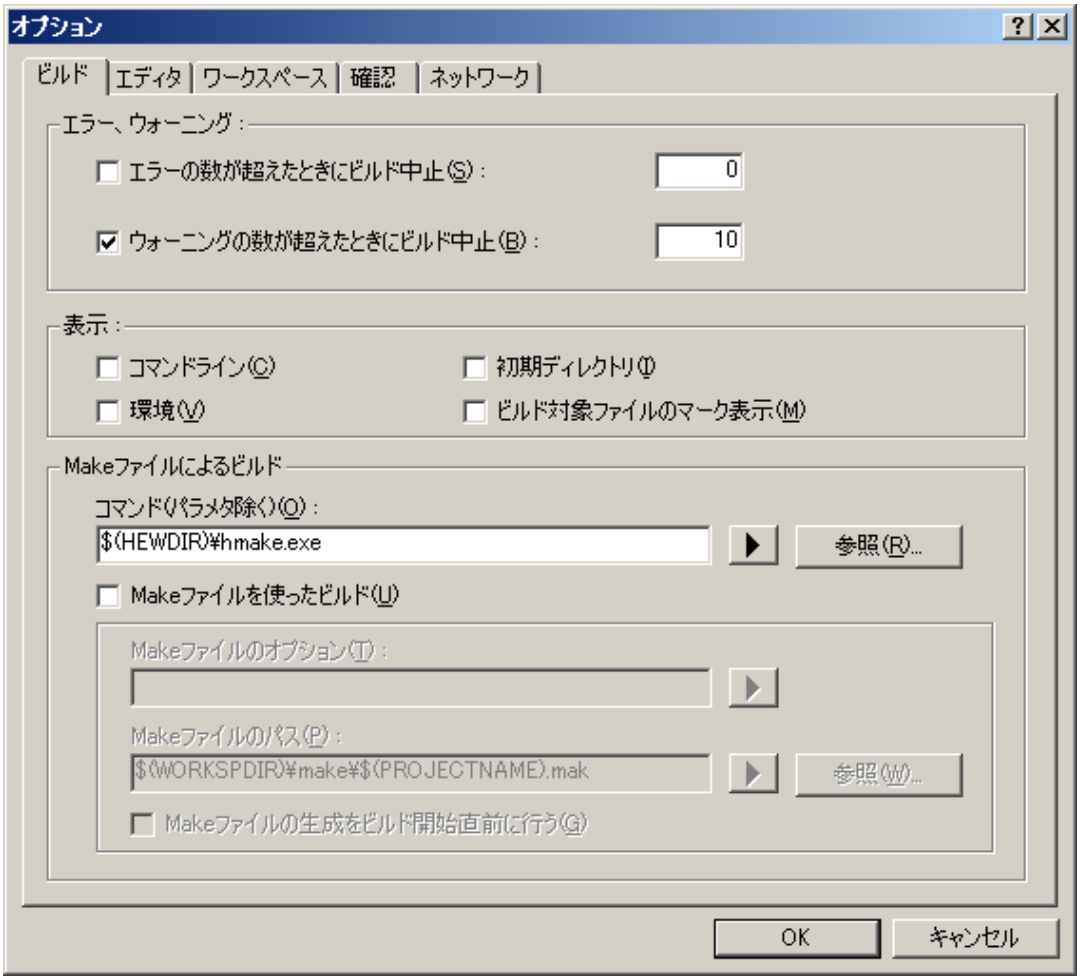


図 3.18: オプションダイアログボックス ビルド タブ

[ツール->オプション...] を選んでダイアログボックスを表示してください。ある一定の回数を超えるエラーが起きた後にビルドを中止したいとき、[エラーの数が超えたときにビルド中止]チェックボックスをチェックして右のフィールドにエラー数を入力してください。指定した数を超えるウォーニングが表示されたときにビルドを中止したいとき、[ウォーニングの数が超えたときにビルド中止]チェックボックスをチェックして右のフィールドにウォーニング数を入力してください。

注意　　上記設定にかかわらず、致命的エラーが発生した場合、ビルドは停止します。

[ビルド]タブでは、エラー数やウォーニング数の制限のほかに、コマンドライン、環境、初期ディ

レクトリの表示の有無を設定することができます。表示するには、それぞれのチェックボックスをチェックしてください。

3.7 ビルドの出力のログを取る

ファイルに各ビルドの結果を保存したいときには、**[ツール-> カスタマイズ...]**を選んで[カスタマイズ]ダイアログボックスを表示して [ログ] タブを選んでください （図 3.19）。 [ログファイル生成] チェックボックスをチェックして、[ディレクトリ] フィールドにログファイルのフルパスを入力するか、[参照...]ボタンをクリックしてパスを選択してください。

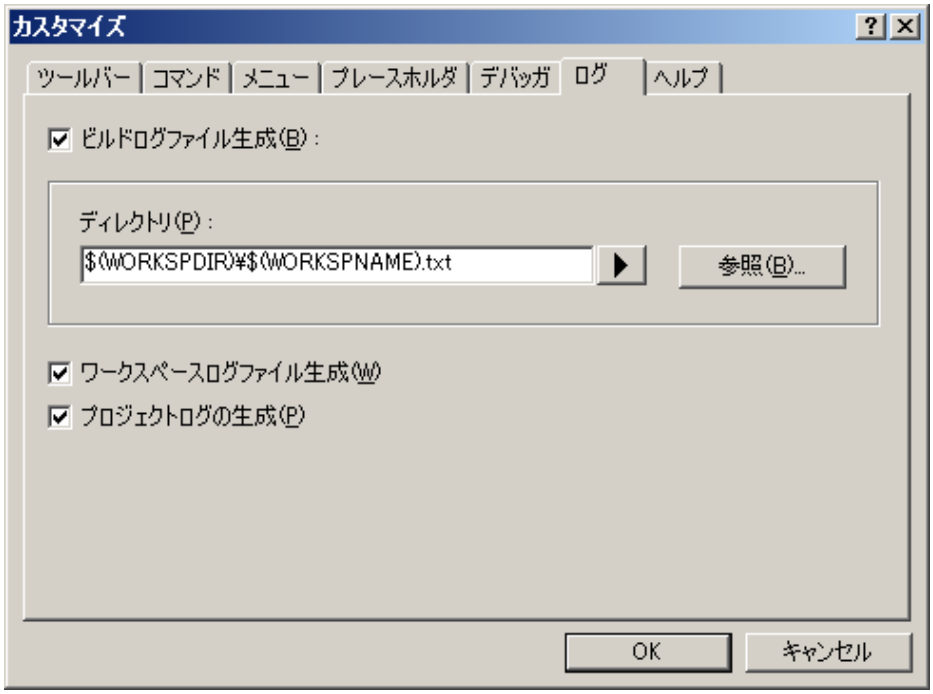


図 3.19: **カスタマイズダイアログボックス ログ タブ**

3.8 ツールチェインのバージョンを変更する

同じツールチェインの２つ以上のバージョンが HEW に登録されているとき図 3.20 に示す[ツールチェインのバージョンの変更]ダイアログボックスでバージョンを選択できます。このダイアログボックスを表示されるには**[ツール->ツールチェインバージョンを変更...]**を選択してください。このダイアログボックスの[ツールチェインバージョン]ドロップダウンリストからバージョンを選択し、[OK]ボタンをクリックしてください。

図 3.20: **ツールチェインのバージョンの変更 ダイアログボックス**

ツールチェインを構成するツールの情報を表示するには[ツールチェインのバージョンの変更]ダイアログボックスの[ツールチェインビルドフェーズ]リストからツールを一つ選択し、[情報...]ボタンをクリックしてください。ツール情報ダイアログボックス（図 3.21）にそのツールの情報が表示されます。このダイアログボックスを閉じるには[閉じる]ボタンをクリックしてください。

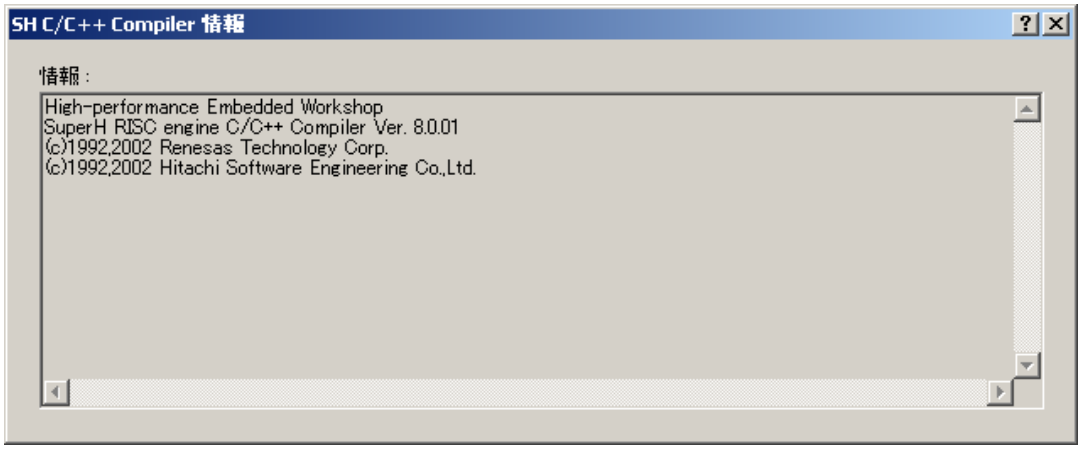


図 3.21: **ツール情報 ダイアログボックス**

3.9 外部デバッガを使う

HEW は外部デバッガツールを起動することができます。他のデバッガを使用する場合は、[ツール]メニューに追加する必要があります。

[カスタマイズ] ダイアログボックスの [デバッガ] タブに外部デバッガに関連する情報を設定します（図 3.22a、図 3.22b、図 3.22c）。一部のターゲットが新しい環境でサポートされていない場合は、古いバージョンのデバッガを使用することができます。**[ツール-> カスタマイズ...]**を選んでダイアログボックスを表示させ、[デバッガ]タブを選んでください。
[外部デバッガの選択]ドロップダウンリストで使用する外部デバッガを選択します。

[Hitachi Debugger Interface (version 4.x or greater)] ：HDIを使用する
[Mitsubishi PD debugger] ：PDデバッガを使用する
[Other external debugger] ：HDIまたはPDデバッガ以外の外部デバッガを使用する
[Non selected] ： 外部デバッガを使用しない

外部デバッガを使用する場合、そのデバッガ固有の情報を構築する必要があります。

➲ HDI を使用するには
1. HDIの場所を指定します。HDI 4.0以降のバージョンでない場合この動作は保証されません。
2. セッションファイルの場所を指定します。起動時にどのセッションをロードするかHDIに伝えます。
3. ダウンロードモジュールの場所を指定します。

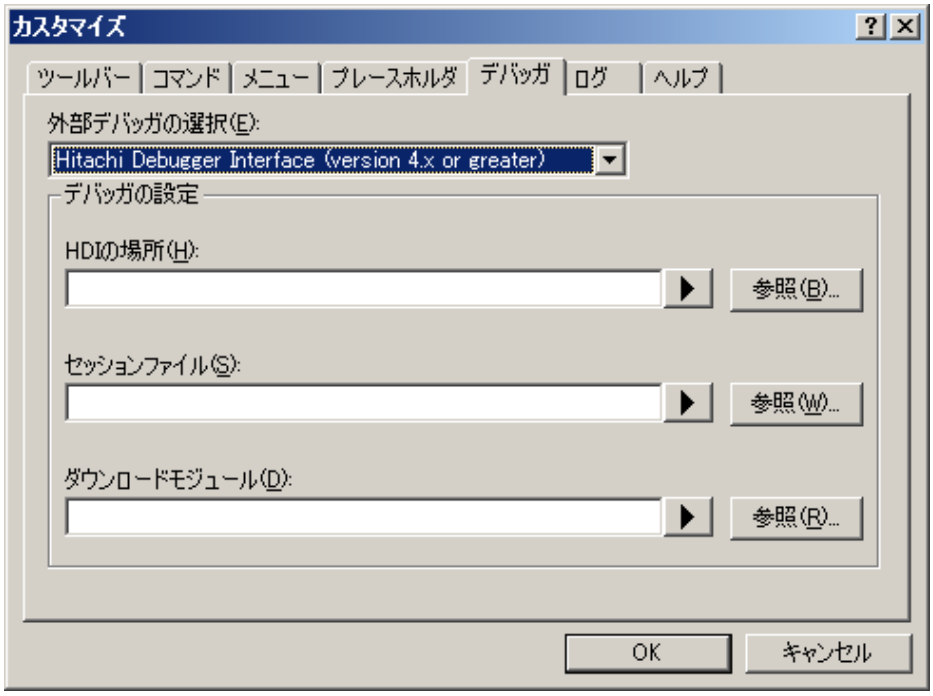


図 3.22a: **外部デバッガの選択** - HDI

➲ PD デバッガを使用するには

1. PDデバッガの場所を指定します。
2. PD Profileファイルの場所を指定します。起動時にどのセッションをロードするかPDデバッガに伝えます。
3. コマンドラインオプションです。PDデバッガの動作を変更するオプションを指定できます。
4. ダウンロードモジュールの場所を指定します。

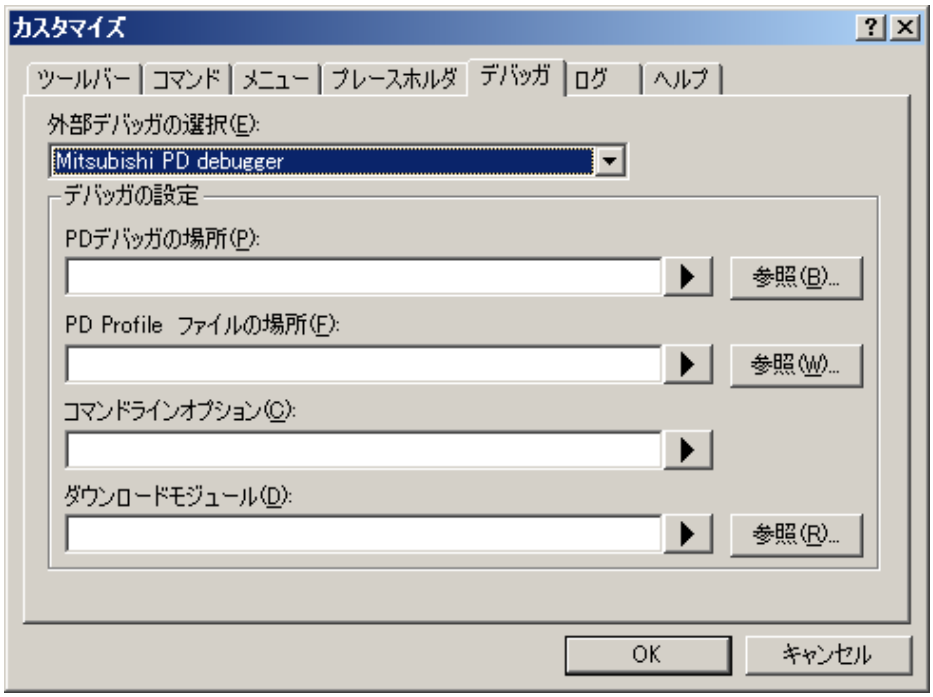


図 3.22b: **外部デバッガの選択** - PD **デバッガ**

➲ HDI または PD デバッガ以外の外部デバッガを使用するには

1. 外部デバッガの場所を指定します。
2. コマンドラインオプションです。外部デバッガの動作を変更するオプションを指定できます。
3. ダウンロードモジュールの場所を指定します。

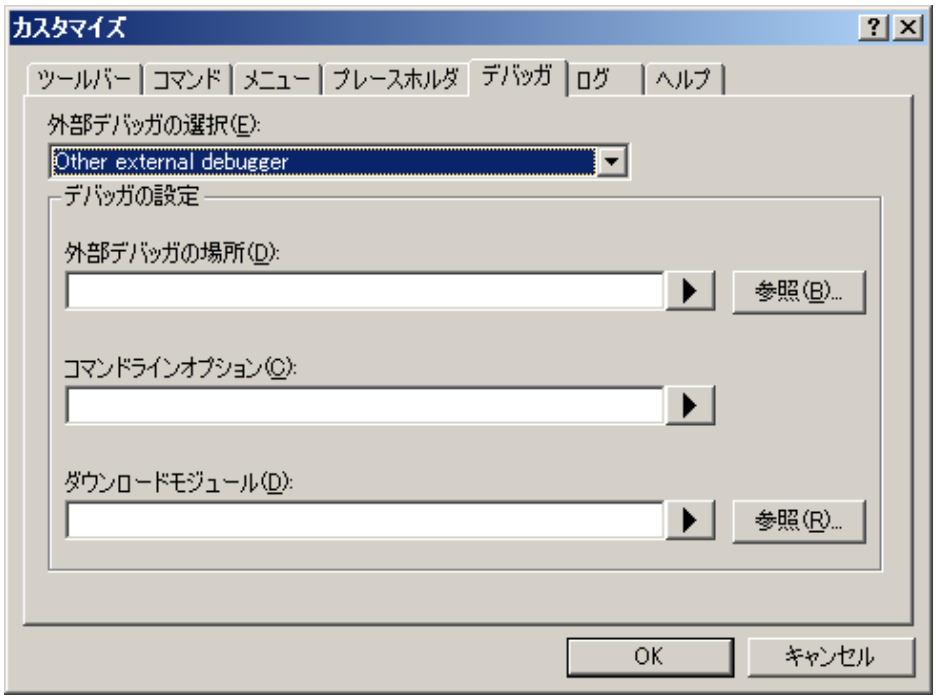


図 3.22c: **外部デバッガの選択** - HDI **または** PD **デバッガ以外の外部デバッガ**

外部デバッガを起動するには、[外部デバッガの実行]ツールバーボタンをクリックしてください。

ビルド後にダウンロードモジュールが変更された場合 HEW から外部デバッガに切り替わり、すぐデバッグできるようになります。

外部デバッガを使用しているときにソースウィンドウのどれかをダブルクリックすると、HEW に切り替わりダブルクリックした行のソースファイルを開いた状態になります。[Other external debugger]を選んだ場合は、この操作を行うことはできません。

3.10 メイクファイルの生成

HEW では、メイクファイルを生成することができます。メイクファイルを使用すると、完全に HEW をインストールしていなくても、現在のプロジェクトをビルドすることができます。HEW をインストールしていない相手にプロジェクトを送ったり、メイクファイルを含むビルド全体をバージョン管理したりする場合に便利です。

本バージョンの HEW では、生成したメイクファイルと Make ツールを使用することもできます。[ビルド]メニューまたはツールバーボタンをクリックし、HEW の標準のビルド処理ではなく、Make ツールを起動することができます。詳細は、「3.11 HEW システム内部のメイクファイルを使用する」を参照してください。

➲ メイクファイルを生成するには

1. メイクファイルを生成するプロジェクトが現在のプロジェクトであることを確認してください。
2. プロジェクトをビルドするビルドコンフィグレーションが現在のコンフィグレーションであることを確認してください。
3. [ビルド->Makeファイルの生成] を選んでください。
4. [Makeファイルの生成]ダイアログボックスが表示されます。（図3.23）
5. [Makeファイルの生成タイプ]で生成したいメイクファイルをラジオボタンで選んでください。
6. 選択した[Makeファイルのフォーマット] が適切であるか確認してください。HEWでは、HMakeおよびNMake、GNUMakeに対応したファイルを生成できます。
7. [サブコマンドファイルの生成]チェックボックスをチェックすると、メイクファイルが出力されたディレクトリ内にコマンドファイルを生成します。GNUMakeファイル形式の場合、このチェックボックスを有効にする必要があります。
8. [Makeファイルの生成時に依存関係をスキャン]チェックボックスをチェックすると、最新のメイクファイルが生成されるよう、依存性のスキャンを行います。
9. [OK]ボタンをクリックしてください。

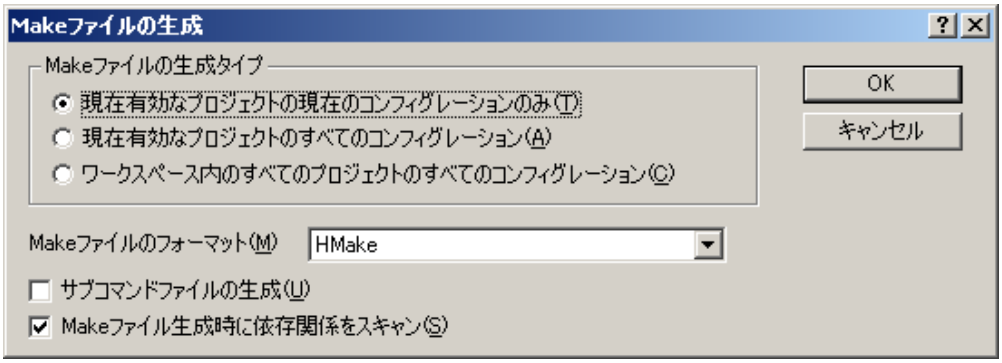


図 3.23: Make **ファイルの生成 ダイアログボックス**

HEW は現在のワークスペースディレクトリ内に“make”サブディレクトリを作り、その中にメイクファイルを生成します。メイクファイルの名前は、現在のプロジェクトやコンフィグレーションに拡張子.mak を付けたものです(例：project　debug.mak)。HEW により生成されたメイクファイルは、HEW をインストールしたディレクトリにある実行ファイル HMAKE.EXE で実行できます。ただし、ユーザが変更したメイクファイルは実行できません。HMAKE の詳細については、付録 F,「HMAKE ユーザガイド」を参照してください。

➲ メイクファイルを実行するには

1. コマンドウィンドウを開き、メイクファイルが生成された“make”ディレクトリに移動してく

ください。

2. HMAKEを実行してください。コマンドラインは”HMAKE.EXE△<メイクファイル名>”です。

注意　生成したメイクファイルが移動可能か否かは、プロジェクト自体が移動可能か否かに依存します。たとえば、出力ディレクトリやインクルードファイルディレクトリへのフルパスを含むコンパイラオプションがあると、異なるインストール環境下の別のユーザがビルドした場合、失敗する可能性があります。一般的に、できるだけプレースホルダを使用して、フルパス、または特定のパスの使用はなるべく避けてください。

3.11 HEW システム内部のメイクファイルを使用する

HEW では、[ビルド]をクリックすると内部ビルドを構築します。本機能は、標準の HEW のビルド処理ではなく外部のメイクファイルツールを使用することができます。

ユーザの必要とするビルドには標準の HEW のビルドで十分ですが、時には標準の HEW のビルド処理に依存せず、メイクファイルを使用したい場合もあります。この方法を以下に説明します。

➲ メイクファイルの実行を設定するには

1. メイクファイルでビルドするためのHEWのワークスペースを必要に応じて作成してください。そのときは使用するツールチェインをメイクファイルと同じにする必要があります。
2. **[ツール-> オプション]** をクリックしてください。
3. [ビルド] タブをクリックしてください。（図3.24）
4. [コマンド（パラメタ除く）] にメイクファイルツールを設定してください。デフォルトでは、HEWと一緒に出荷されるルネサス製Makeツールに設定されています。名称はHMAKE.exeで、格納先はHEWインストールディレクトリ“$(HEWDIR)”です。
5. [Makeファイルを使ったビルド] チェックボックスをチェックしてください。[ビルド]ボタンをクリックしたときに、標準のビルド処理ではなくメイクファイルを実行することをHEWに伝えます。
6. [Makeファイルのパス] を設定してください。HEWがメイクファイルを生成するディレクトリがデフォルト値として設定されています。本項目はコンフィグレーションがベースであるため、各コンフィグレーションに異なるメイクファイルを設定することもできます。

注意　必ずしも、HEW で生成したメイクファイルを使用する必要はありません。ご使用の Make ツールでサポートしていれば、どの形式でも使用することができます。

7. [Makeファイルのオプション] を設定してください。HMAKEを使用してプロジェクトまたはコンフィグレーションの選択オプションを指定することで、同一ファイルから異なるビルドを起動することができます。ここでも本項目はコンフィグレーションがベースであるため、各コンフィグレーションに異なるメイクファイルオプションを設定することができます。HMAKEのオプションについては、付録 F, 「HMAKEユーザガイド」の「F.1.3 パラメータ」を参照してください。
8. 最後のオプションは [Makeファイルの生成をビルド開始直前に行う] です。メイクファイルを実行する前に、デフォルトのメイクファイル生成機能を起動します。つまり、メイクファイルは常に最新の状態のHEWプロジェクトに対応しています。
9. [OK] ボタンをクリックしてください。

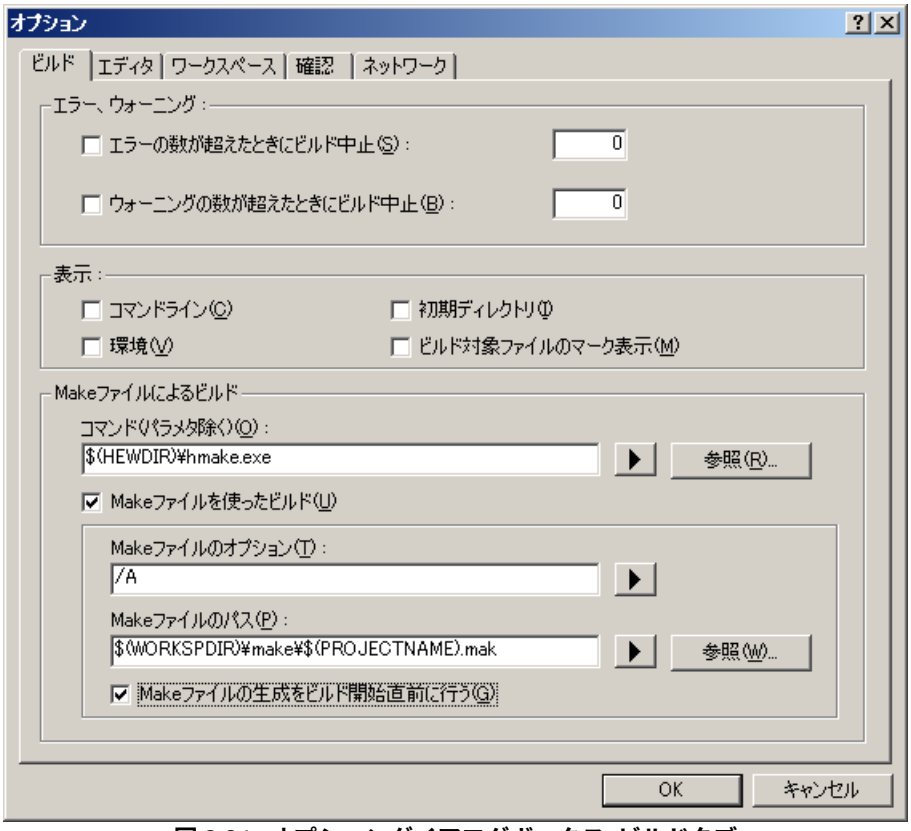


図 3.24: **オプションダイアログボックス ビルドタブ**

本システムを設定すると、[ビルド]ボタン、メニュー、およびキーボードのショートカットは Make ツールの実行とリンクします。標準のビルド処理同様、出力先はすべてアウトプットウィンドウです。HEW がサポートしているツールチェインを使用する場合、エラーおよび警告をダブルクリックするとソースファイルにジャンプします。エラー等の行を選択して F1 キーを押下すると、詳細を表示する機能をサポートしています。HMAKE.exe システムをご使用になる場合、[すべてをビルド]ボタンをクリックすると、HMAKE にコマンドを送り強制的に再ビルドすることにご注意ください。GNUMake ツールをご使用の場合には、本機能をサポートしていません。

3.12 リンク順序をカスタマイズする

HEW では、通常オブジェクトファイルは、ファイル名のアルファベット順にリンクしますが、リンクする順番をカスタマイズすることができます。

➲ リンク順序をカスタマイズするには

1. **[ビルド->リンク順の指定...]** を選んでください。
2. 表示されるダイアログボックスで [リンク順序のカスタマイズの使用] チェックボックスをチェックしてください。（図3.25）
3. オブジェクトの順序を並びかえることができます。モジュールを選択し、[上へ]ボタンまたは[下へ]ボタンをクリックして任意の位置に移動してください。
4. 各モジュールには、そのモジュールの属性によって異なるアイコンを表示しています。
5. [OK] ボタンをクリックしてください。

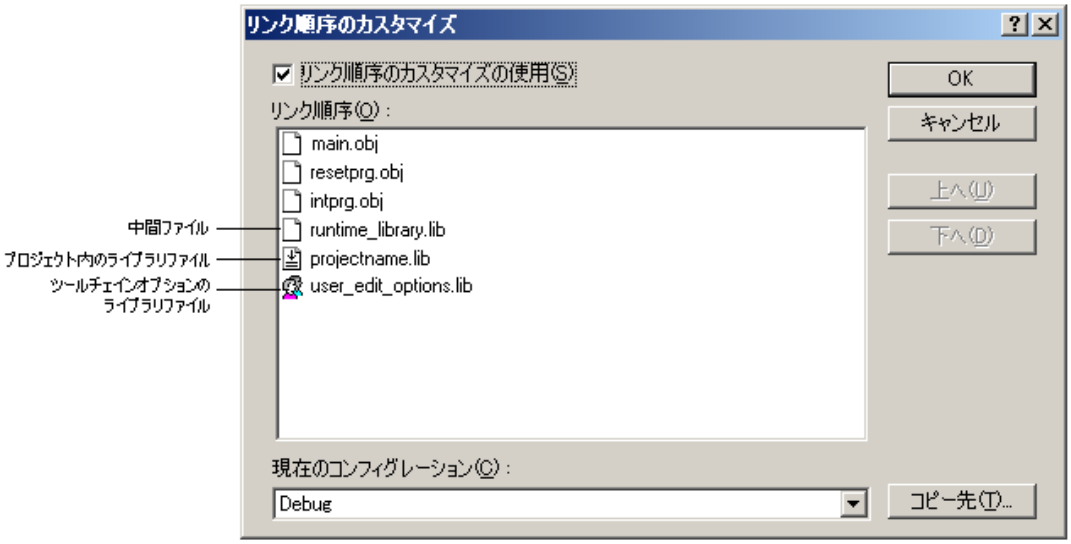


図 3.25: **リンク順序のカスタマイズ ダイアログボックス**

複数のコンフィグレーションを使用する場合、リンク順序を同じにする場合があります。現在のリンク順序の設定を他のコンフィグレーションにコピーする方法を以下に説明します。

➲ 他のコンフィグレーションにリンク順序をコピーするには

1. **[ビルド->リンク順の指定...]** を選んでください。
2. [現在のコンフィグレーション] リストで、コピー元のコンフィグレーションを選択してください。現在のコンフィグレーションがデフォルトで表示されます。
3. [コピー先...] ボタンをクリックしてください。ダイアログボックスが表示され、現在のリンク順序を現在のプロジェクトのどのコンフィグレーションにコピーするかを指定します。コンフィグレーションを選択し、[OK]ボタンをクリックしてください。
4. [OK] ボタンをクリックしてください。

4. エディタの使用

この章では HEW が提供するエディタの使い方を説明します。

4.1 エディタウィンドウ

エディタウィンドウには、表示中や編集中のファイルウィンドウが含まれます（図 4.1）。

常に 1 つのウィンドウだけがアクティブです。それをアクティブウィンドウ（または現在のウィンドウ）と呼びます。アクティブウィンドウのタイトルバーは他のウィンドウとは色が異なります。（図 4.1 では“dbsct.c”がアクティブウィンドウです。）文字入力やテキスト貼りつけなどのテキスト操作はアクティブウィンドウでのみ行うことができます。

アクティブウィンドウから他のソースファイルウィンドウへ切り替えるには（他のウィンドウをアクティブウィンドウにする場合）、以下の方法があります。

(a) そのウィンドウが表示されていればそれをクリックする。
(b) CTRL+TAB キーを押下して順次ウィンドウをアクティブにする。
(c) [ウィンドウ]メニューからウィンドウ名を選ぶ。
(d) エディタウィンドウの下の該当するタブを選ぶ。

ファイルが変更されている場合、ファイル名のタイトルバーにアスタリスク（＊）が追加されます。アスタリスクはファイルを保存するまで表示されます。現在のウィンドウですべての変更が戻された(Undo)場合には、アスタリスクは削除されます。

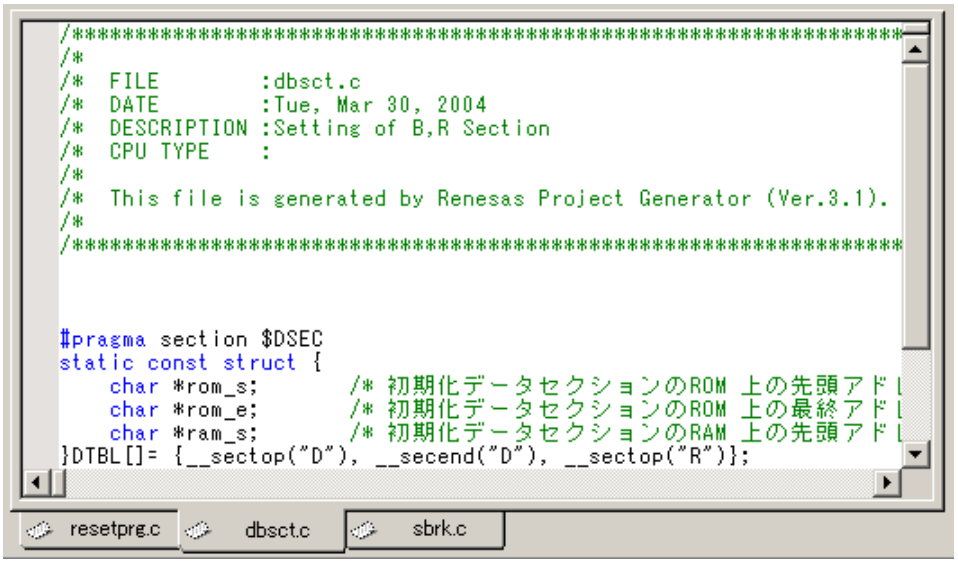


図 4.1: **エディタウィンドウ**

4.2 複数のファイルを使う

プロジェクトのファイルを操作するときにはエディタウィンドウを使います。エディタによって、一度に複数のファイルを開いたり、ファイル間の切り替えをしたり、異なる構成に並べ替えたり、任意の順序に編集できます。ウィンドウ上での操作はWindows®のアプリケーションの一般的な機能であり、[ウィンドウ]メニューから選ぶことができます。

[ウィンドウ->重ねて表示]
開いているすべてのウィンドウの左上が見えるように重ねて並べます。

[ウィンドウ->上下に並べて表示]
開いているすべてのウィンドウが互いに重ならずにエディタウィンドウの全体を占めるように並べます（水平方向）。

[ウィンドウ->左右に並べて表示]
開いているすべてのウィンドウが互いに重ならずにエディタウィンドウの全体を占めるように並べます（垂直方向）。

[ウィンドウ->アイコンの整列]
すべての最小化したウィンドウをエディタウィンドウの下に並べます。

[ウィンドウ->すべて閉じる]
開いているエディタウィンドウをすべて閉じます。

また、エディタのファイルはノートブック形式で表示することができます。つまり、各ファイルにタブが付いていてファイル間の行き来を容易にできます。

➲ ファイルをノートブック形式で表示するには
1.[ツール->オプション…] を選んでください。 [オプション] ダイアログボックスが表示されるので [エディタ] タブを選んでください。
2.[ブック形式でファイルを表示] チェックボックスをチェックしてください。
3.[OK] ボタンをクリックしてください。

4.2.1 エディタツールバー

エディタツールバーには、エディタ、検索、ブックマーク、テンプレートの４つの関連するツールバーがあります。これにより、頻繁に使うエディタの機能を簡単に選ぶことができます。以下に各ボタンの機能を説明します。

4.2.2 エディタツールバーボタン

新規ファイル
新規のソースファイルウィンドウをデフォルト名で作成します。ファイルを保存するときにファイルの名前を変えることができます。

ファイルを開く

ファイルを開きます。標準のファイル選択用ダイアログボックスが表示されます。開きたいファイルを選び“開く”ボタンをクリックします。

ファイルの保存

アクティブウィンドウのソースファイルを保存します。

すべてのファイルの保存

開いているすべてのファイルを保存します。

ファイルの印刷

クリックすると、現在のアクティブウィンドウの内容を印刷します。

カット

選択されたテキストを削除してWindows® のクリップボードにコピーします。（貼りつけ操作をすると、再びファイルに貼りつけることができます。）

コピー

選択されたテキストを Windows® のクリップボードにコピーします。

貼りつけ

アクティブウィンドウのカーソル位置にクリップボードの内容を貼りつけます。

かっこの組み合わせ

かっこ{ }, [ ] , ( )内のテキストをハイライト表示します。これは、開きかっこ（{）で始まり閉じかっこ（}）で終わるC/C++言語のコードの構造を知りたいときに便利です。始まりのかっこを選択するか、始まりのかっこの前にカーソルを置いて、このボタンをクリックしてください。かっこの組み合わせの詳細については、この章の後半の「かっこの組み合わせ」を参照してください。

テンプレートの挿入

あらかじめ決められたテンプレートを現在のカーソル位置に挿入します。[テンプレートの挿入]ダイアログボックスが表示されます。テンプレート名を選び、[OK]ボタンをクリックしてください。テンプレートの挿入の詳細については、この章の後半の「テンプレート」を参照してください。

ブックマークの設定と解除

HEWはブックマーク機能を提供します。ブックマークを設定するには、行を選び、このボタンをクリックしてください（エディタウィンドウの左余白に緑色のマークが付きます）。ブックマークを解除するには、行を選び、このボタンをクリックしてください（エディタウィンドウの左余白のマークが消えます）。ブックマークの詳細については、この章の後半の「ブックマーク」を参照してください。

4.2.3 検索ツールバーボタン

ファイル間検索

複数のファイルを指定された文字列で検索します。検索結果はすべてアウトプットウィンドウの“Find in Files”タブに表示します。詳細は、この章の後半の「複数のファイル間でのテキスト検索」を参照してください。

検索

現在のファイルを指定された文字列で検索します。[検索]ダイアログボックスが表示されるので、検索パラメータを指定してください。

次を検索

現在の検索文字列の次の出現を検索します。

前を検索

現在の検索文字列の直前の出現を検索します。

4.2.4 ブックマークツールバーボタン

ブックマークの設定と解除

現在の行にブックマークを設定します。または、現在の行のブックマークを解除します。

次のブックマーク

現在の行から現在のファイルの次のブックマークに飛びます。

前のブックマーク

現在の行から現在のファイルの前のブックマークに飛びます。

全ブックマーク解除

現在のファイルのすべてのブックマークを解除します。

4.2.5 テンプレートツールバーボタン

テンプレートの定義
挿入するテンプレートテキストを定義します。

テンプレートの挿入
ドロップダウンリストで選んだテンプレートを、現在のカーソル位置に挿入します。

4.3 標準のファイル操作

4.3.1 新規ファイルの作成

➲ 新しいエディタウィンドウを作成するには
[ファイル->新規作成] を選択するか、“新規ファイル”ツールバーボタン ( ) をクリックするか、”CTRL+N”キーを押下してください。

新しいウィンドウにはデフォルトで名前がつきます。ファイルを保存するときに名前を変更できます。

4.3.2 ファイルの保存

➲ エディタウィンドウの内容を保存するには
1. 内容を保存するウィンドウがアクティブであることを確認してください。
2. [ファイル->上書き保存] を選択するか、 [ファイルの保存] ツールバーボタン ( ) をクリックするか、”CTRL+S”キーを押下してください。
3. ファイルを新規に保存する場合、名前を付けて保存するダイアログボックスが表示されます。ファイル名とディレクトリを指定して、[保存]をクリックしてください。
4. すでに保存したことのあるファイルの場合は、ダイアログボックスを表示せずにファイルが更新されます。

➲ エディタウィンドウの内容を新しい名前で保存するには
1. 内容を保存するウィンドウがアクティブであることを確認してください。
2. [ファイル->名前を付けて保存...] を選んでください。
3. 名前を付けて保存するダイアログボックスが表示されます。ファイル名とディレクトリを指定して[保存]をクリックしてください。

4.3.3 全ファイルの保存

➲ すべての開いているエディタウィンドウの内容を保存するには
1. [ファイル->すべて保存] を選ぶか、 [すべて保存] ツールバーボタン ( ) をクリックしてください。
2. 新規にファイルを保存する場合、名前を付けて保存するためのダイアログボックスが表示されます。ファイル名とディレクトリを指定してください。

3. すでに保存したことのあるファイルの場合は、ダイアログボックスを表示せずにファイルが更新されます。

4.3.4 ファイルを開く

➲ ファイルを開くには

1. [ファイル->開く...] を選ぶか、 [ファイルを開く] ツールバーボタン ( ) をクリックするか、 “CTRL+O”キーを押下してください。
2. ファイルを開くダイアログボックスが表示されます。右のディレクトリブラウザを使って、開きたいファイルのあるディレクトリに移動してください。[ファイルの種類]コンボボックスでファイルの種類を選んでください。 (“All Files” (*.*) を指定すると、そのディレクトリのすべてのファイルが表示されます。)
3. ファイルを選んで [開く] ボタンをクリックしてください。

開いた最新のファイル 4 つは[ファイル]メニューの[最近使ったファイル]サブメニューに追加されます。この機能は最近開いたファイルを再び開きたいときに便利です。

➲ 最近利用したファイルを開くには
[ファイル->最近使ったファイル] メニューオプションを選択してこのサブメニューから開きたいファイルを選択してください。または、ワークスペースウィンドウの[Projects]タブで開きたいファイルをダブルクリックするか、ファイルを選び、右マウスボタンをクリックしてポップアップメニューから[開く <ファイル名>]オプションを選んでください。

4.3.5 ファイルを閉じる

個別にファイルを閉じるには、以下のいずれかの方法で閉じます。

- エディタウィンドウのシステムメニュー（[最大化]表示されていないときの各ウィンドウの左上）をダブルクリックする。
- エディタウィンドウのシステムメニュー（[最大化]表示されていないときの各ウィンドウの左上）をクリックして[閉じる]メニューオプションを選ぶ。
- アクティブウィンドウであることを確認後”CTRL+F4”キーを押下する。
- アクティブウィンドウであることを確認後[ファイル->閉じる]を選ぶ。
- “閉じる”ボタン（[最大化]表示されていないときの各ウィンドウの右上）をクリックする。

➲ すべてのウィンドウを閉じるには
[ウィンドウ->すべて閉じる]を選んでください。

4.4 ファイルを編集する

HEW エディタは標準的な編集機能をサポートします。通常の方法（メニュー、ツールバー、キーボードのショートカット）で編集できるほか、各エディタウィンドウにあるポップアップメニュー（またはローカルメニューという）でも編集することができます。ポップアップメニューを表示するには、開いたウィンドウにポインタを置き、マウスの右ボタンをクリックしてください。表 4.1 に編集の基本操作を示します。

表 4.1: 編集の基本操作

名称	機能	操作
切り取り	ハイライト表示されたテキストを削除し Windows®クリップボードに貼りつける	“カット”ツールバーボタンをクリック [編集->切り取り]を選択 [切り取り] -ローカルメニューを選択 CTRL+X キーを押下
コピー	ハイライト表示されたテキストをコピーし Windows®クリップボードに貼りつける	“コピー ”ツールバーボタンをクリック [編集->コピー] を選択 [コピー] -ローカルメニューを選択 CTRL+C キーを押下
貼り付け	Windows®クリップボードの内容をコピーしてアクティブウィンドウのカーソル位置に貼りつける	“貼り付け”ツールバーボタンをクリック [編集->貼り付け]を選択 [貼り付け] - ローカルメニューを選択 CTRL+V キーを押下
削除	ハイライト表示されたテキストを削除する（Windows®クリップボードに貼りつけない）	[編集->削除] を選択 Delete キーを押下
すべて選択	アクティブウィンドウの内容すべてを選択（ハイライト表示）する	[編集->すべて選択] を選択 CTRL+A キーを押下
元に戻す	最新の編集操作を取り消す	[編集->元に戻す] を選択 CTRL+Z キーを押下
やり直し	最新の取り消した編集操作をやり直す	[編集->やり直し] を選択 CTRL+Y キーを押下

4.5 検索とファイル内の移動

HEW には、検索、置換、ファイル間の操作の機能があります。次の 3 節にそれらの機能の使い方を示します。

4.5.1 テキストの検索

➲ 現在のファイルのテキストを検索するには

1. 内容を検索するウィンドウがアクティブウィンドウであることを確認してください。
2. 検索開始位置にカーソルを置いてください。
3. [編集->検索...] を選ぶか、”CTRL+F”キーを押下するか、エディタウィンドウのローカルメニューから [検索...] を選ぶか、 [検索] ツールバーボタン ( ) をクリックしてください。[検索]ダイアログボックスが表示されます （図4.2）。

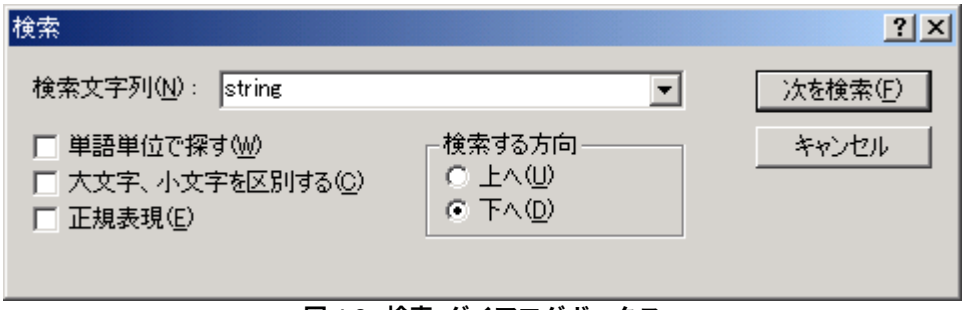


図 4.2: **検索 ダイアログボックス**

4. [検索文字列] フィールドに検索するテキストを入力するか、ドロップダウンリストボックスから以前検索した文字列を選んでください。ファイルウィンドウでテキストを選んでから[検索]ダイアログボックスを開くと、選んだテキストが自動的に[検索文字列]フィールドに入ります。
5. 単語単位で文字列を探す場合は、 [単語単位で探す] チェックボックスをチェックしてください。このオプションを選択しない場合は、一部でも一致する文字列が検索されます。
6. 大文字と小文字を識別したいときは [大文字、小文字を区別する] チェックボックスをチェックしてください。
7. 正規表現を使ってテキストを検索する場合 [正規表現] チェックボックスをチェックしてください。詳細は付録B, 「正規表現」を参照してください。
8. [検索する方向] ラジオボタンにより、検索する方向を指定できます。[下へ]を選ぶと、カーソル位置からファイルの下の方向に検索します。[上へ]を選ぶと、カーソル位置から上の方向に検索します。
9. [次を検索] ボタンをクリックすると検索を始めます。検索を終了するには[キャンセル] ボタンをクリックしてください。

また、複数のファイル間での検索もできます。

4.5.2 複数のファイル間でのテキスト検索

➲ 複数のファイルでテキストを検索するには

1. [編集->ファイル内から検索...] を選ぶか、エディタウィンドウのローカルメニューから [ファイル内から検索...] を選ぶか、 [ファイルの中から検索] ツールバーボタン ( ) をクリックしてください。[ファイルから検索]ダイアログボックスが表示されます（図4.3）。

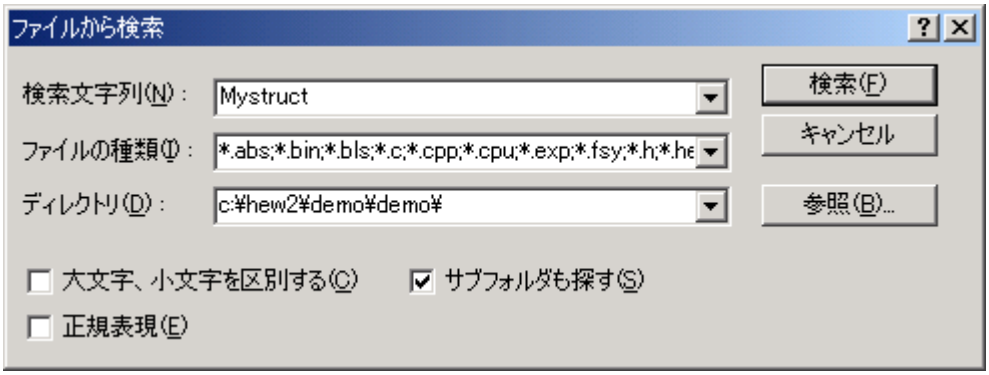


図 4.3: **ファイルから検索 ダイアログボックス**

2. [検索文字列] フィールドに検索したいテキストを入力するか、ドロップダウンリストボックスから以前検索した文字列を選んでください。ファイルウィンドウでテキストを選んでから [ファイルから検索]ダイアログボックスを開くと、そのテキストが自動的に[検索文字列] フィールドに入ります。
3. [ファイルの種類] フィールドに、検索するファイルの拡張子を指定してください。複数の拡張子を指定するときは、セミコロンで区切ってください（例：*.c;*.h）。
4. [ディレクトリ] フィールドに検索するファイルのディレクトリを指定してください。または、[参照...] ボタンをクリックして画面上で検索するファイルを指定してください。
5. 指定したディレクトリとその下のディレクトリすべてを検索したいときは、 [サブフォルダも探す] チェックボックスをチェックしてください。[ディレクトリ]フィールドで指定したディレクトリだけを検索したいときは、[サブフォルダも探す]チェックボックスがチェックされていないことを確認してください。
6. 大文字と小文字を識別したいときは [大文字、小文字を区別する] チェックボックスをチェックしてください。
7. 正規表現を使ってテキストを検索したいときは、 [正規表現] チェックボックスをチェックしてください。 詳細は付録 B,「正規表現」を参照してください。
8. [検索] ボタンをクリックすると検索を始めます。 検索結果はアウトプットウィンドウの [Find in Files]タブに表示されます。アウトプットウィンドウの文字列をダブルクリックすると、ファイルの当該文字列にジャンプします。

4.5.3 テキストを置換する

テキストの置換は前節で説明したテキストの検索に似ています。異なる点は、テキスト検索後、他のテキストに置き換えるところです。

➲ ファイルのテキストを置換するには

1. 内容を置換するウィンドウがアクティブであることを確認してください。
2. 検索を始める位置にカーソルを置いてください。
3. [編集->置換...] を選ぶか、”CTRL+H”キーを押下するか、エディタウィンドウのローカルメニューから [置換...] を選んでください。[置換]ダイアログボックスが表示されます（図4.4）。
4. [置換文字列] フィールドに置換前のテキストを入力してください。または、ドロップダウンリストボックスから以前に検索した文字列を選んでください。ファイルウィンドウでテキストを選んでから[置換]ダイアログボックスを開くと、選んだテキストが自動的に[置換文字列]フィールドに入ります。
5. [置換後の文字列] フィールドに置換後のテキストを入力してください。または、ドロップダウンリストボックスから以前に検索した文字列を選んでください。

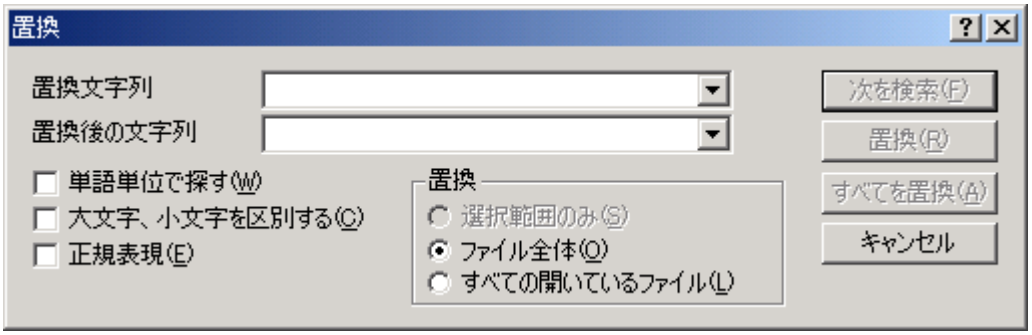


図 4.4: **置換 ダイアログボックス**

6. 単語単位で文字列を探す場合は、 [単語単位で探す] チェックボックスをチェックしてください。このオプションを選択しない場合は、一部でも一致する文字列が検索されます。
7. 大文字と小文字を識別したいときは [大文字、小文字を区別する] チェックボックスをチェックしてください。
8. 正規表現を使ってテキストを検索したいときは、 [正規表現] チェックボックスをチェックしてください。 詳細は付録 B, 「正規表現」を参照してください。
9. [次を検索] をクリックすると、検索文字列、条件に合致したものの次を検索します。置換したい場合は[置換]をクリックしてください。[すべてを置換]をクリックすると、該当するすべての文字列を置換します。[キャンセル]をクリックすると、置換動作を停止します。
   なお、[置換]で[選択範囲のみ]を選択している場合はテキストの選択範囲が置換対象となり、[ファイル全体]を選択している場合はファイル全体が置換対象となります。[すべての開いているファイル]を選択している場合は、エディタで現在開いているファイルがすべて置換対象となります。

4.5.4 指定した行にジャンプする

➲ ファイルの指定した行にジャンプするには

1. ファイルがアクティブであることを確認してください。
2. [編集->ジャンプ...] を選ぶか、”CTRL+G”キーを押下するか、エディタウィンドウのローカルメニューから [ジャンプ...] を選んでください。[ジャンプ]ダイアログボックスが表示されます（図4.5）。
3. ダイアログボックスに指定する行番号を入力して、 [OK] ボタンをクリックしてください。
4. カーソルが指定した行の先頭に移ります。

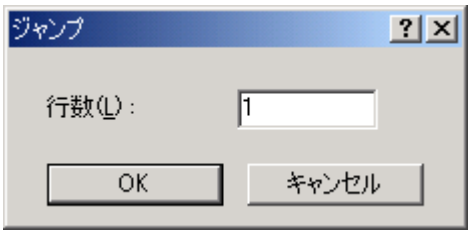


図 4.5: **ジャンプダイアログボックス**

4.6 ブックマーク

一度に多くの大容量のファイルを扱うとき、必要な行や領域を見つけるのが難しくなります。ブックマークをあらかじめ特定の行に設定しておくと、後でその行にジャンプできます。例えば、行数の多い C 言語ファイルの各関数の定義位置にブックマークを設定すると便利です。ブックマークは、解除するまでまたは、設定したファイルを閉じるまで有効です。

➲ ブックマークを設定するには

1. カーソルを設定する行に置いてください。
2. [編集->ブックマーク->ブックマークの挿入/削除] を選ぶか、”CTRL+F2”キーを押下するか、ローカルメニューから [ブックマーク->ブックマークの挿入/削除] を選ぶか、 [ブックマークの挿入/削除] ツールバーボタン ( ) をクリックしてください。
3. ブックマークが設定されると、その行の左余白に緑のマークが表示されます。

➲ ブックマークを解除するには

1. カーソルを解除する行に置いてください。
2. [編集->ブックマーク->ブックマークの挿入/削除] を選ぶか、”CTRL+F2”キーを押下するか、ローカルメニューから [ブックマーク->ブックマークの挿入/削除] を選ぶか、 [ブックマークの挿入/削除] ツールバーボタン ( ) をクリックしてください。
3. 解除されると左余白部分は通常のテキスト表示に戻ります。

➲ 次のブックマークにジャンプするには

1. カーソルが検索するファイルの中にあることを確認してください。
2. [編集->ブックマーク->次のブックマーク] を選ぶか、”F2”キーを押下するか、ローカルメニューから [ブックマーク->次のブックマーク] を選ぶか、または [次のブックマークへ] ツールバーボタン ( ) をクリックしてください。

➲ 同じファイルの一つ前のブックマークに戻るには

1. カーソルが検索するファイルの中にあることを確認してください。
2. [編集->ブックマーク->前のブックマーク] を選ぶか、”SHIFT+F2” キーを押下するか、ローカルメニューから [ブックマーク->前のブックマーク] を選ぶか、 [前のブックマークへ] ツールバーボタン ( ) をクリックしてください。

➲ ファイル内のすべてのブックマークを解除するには

1. すべてのブックマークを解除するファイルのウィンドウをアクティブにしてください。
2. [編集->ブックマーク->すべてのブックマークの削除] を選ぶか、ローカルメニューから [ブックマーク->すべてのブックマークの削除] を選ぶか、 [すべてのブックマークの削除] ツールバーボタン ( ) をクリックしてください。

4.7 ファイルを印刷する

➲ ファイルを印刷するには

1. 印刷するファイルのウィンドウをアクティブにしてください。
2. [ファイル->印刷...] を選ぶか、[印刷] ツールバーボタン ( ) をクリックするか、”CTRL+P” キーを押下してください。

4.8 テキストのレイアウト

この章では、エディタウィンドウの中でテキストのレイアウトを設定する方法を説明します。

4.8.1 ページ設定

HEW エディタからファイルを印刷するとき、印刷ダイアログボックスの設定により、印刷方法が変わります（例えば、片面印刷、両面印刷など）。また、[ページ設定]ダイアログボックスでは、印刷するテキストの余白（上下左右）を指定できます。プリンタによっては、A4 サイズの端まで印刷できない場合があるので、この指定が必要です。また、用紙の左端にとじしろを残したい場合などにも便利です。

➲ ページの余白を設定するには

1. [ファイル->ページレイアウトの設定...] を選んでください。[ページ設定]ダイアログボックスが表示されます（図4.6）。
2. [余白] フィールドに必要な余白を入力してください（ [インチ] または [ミリメートル] ラジオボタンを選んでください）。
3. [OK] ボタンをクリックすると余白が設定されます。

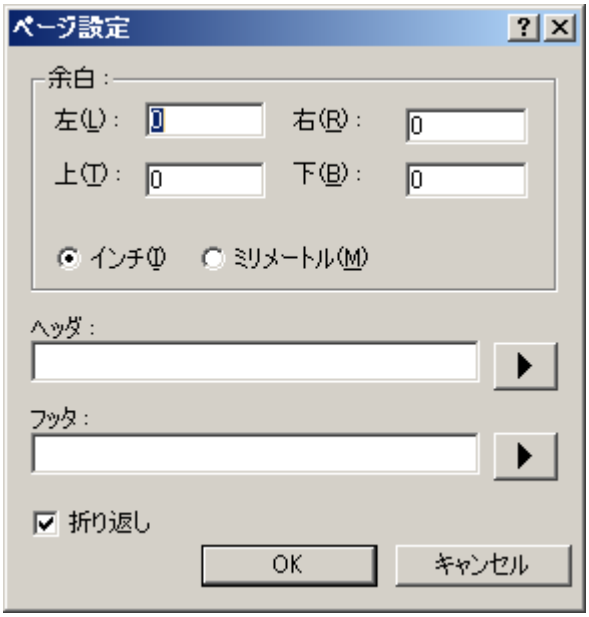


図 4.6: **ページ設定 ダイアログボックス**

➲ ページのヘッダとフッタを設定するには

1. [ファイル->ページレイアウトの設定...] を選んでください。[ページ設定]ダイアログボックスが表示されます（図4.6）。
2. ヘッダおよびフッタの編集フィールドに表示するテキストを入力してください。ページ番号、テキスト配置、日付の各フィールドとともに通常のプレースホルダが利用できます。これらは、ページが印刷される前にすべて拡張されます。
3. [OK] ボタンをクリックすると、新しい設定が有効になります。

➲ 印刷の折り返しを設定するには

1. [ファイル->ページレイアウトの設定...] を選んでください。[ページ設定]ダイアログボックスが表示されます（図4.6）。
2. [折り返し] チェックボックスをクリックしてください。折り返し機能が有効になり、印刷時にテキストが省略されずにすべて表示されます。
3. [OK] ボタンをクリックすると、新しい設定が有効になります。

4.8.2 タブを変更する

➲ タブの文字数を変更するには

1. [ツール->オプション...] を選んでください。[オプション]ダイアログボックスが表示されます。[エディタ]タブを選びます（図4.7）。
2. [タブ幅] フィールドに必要なタブの文字数を指定してください。
3. [OK] ボタンをクリックするとタブの文字数が設定されます。

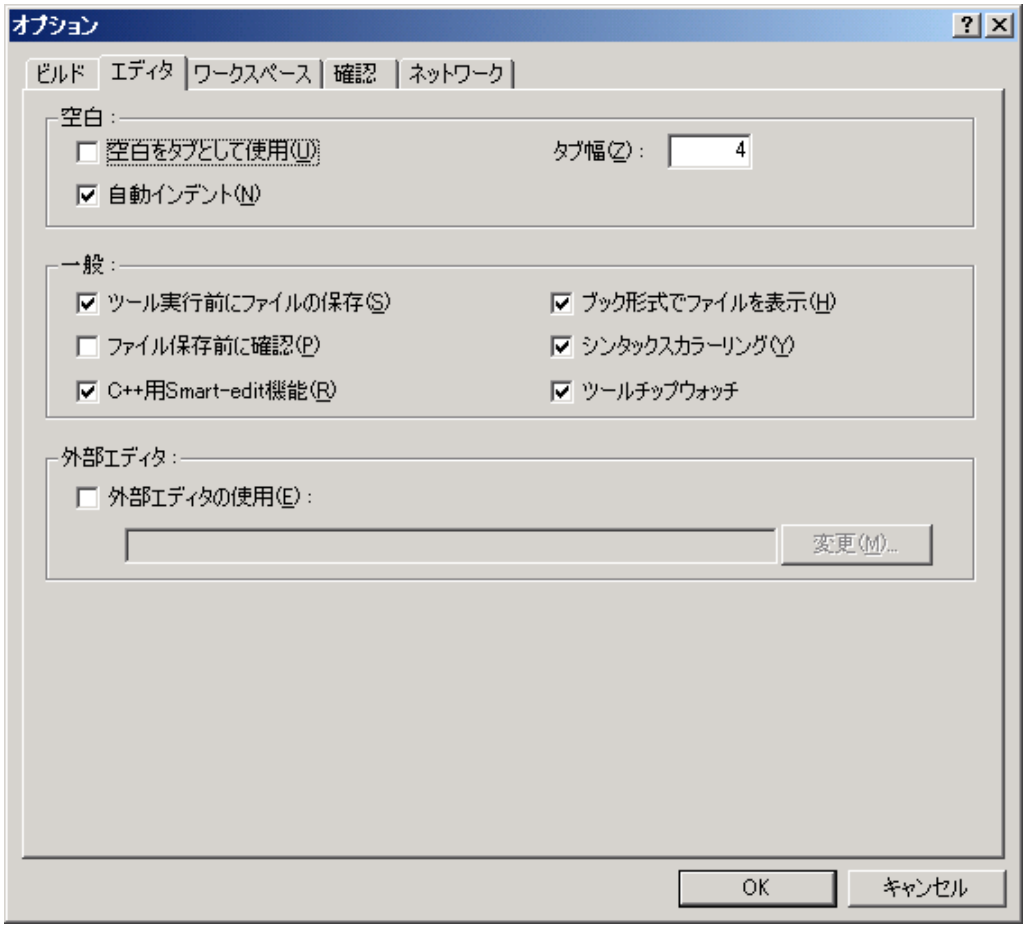


図 4.7: **オプション ダイアログボックス エディタ タブ**

“TAB”キーを押下すると、通常、タブ文字がファイルに入力されます。しかし、タブ文字のかわりに空白文字列を入力したほうがよい場合があります。[オプション]ダイアログボックスの[エディタ]タブで、タブ文字を空白文字列として指定できます。

➲ タブ文字の代わりに空白文字列を使用するには

1. [ツール->オプション...] を選んでください。[オプション]ダイアログボックスが表示されます。[エディタ] タブを選んでください（図4.7）。

2. タブ文字の代わりに空白文字列を使用する場合は [空白をタブとして使用] チェックボックスをチェックしてください。
3. [OK] ボタンをクリックすると設定内容が有効になります。

4.8.3 自動インデント

標準のエディタでリターンキーを押下すると、カーソルは次の行の左端に移ります。自動インデントを設定すると、カーソルは、改行時、前行の最初の文字の下に移ります。したがって、自分でタブや空白を入力することなく、C/C++言語またはアセンブリ言語のコードをより速く見やすく入力できます。

図 4.8 に 2 つの例を示します。 (i) は自動インデントの設定をしなかったときにリターンキーを押した場合の例を示します。カーソルは次の行の左端に移ります。int z=20 の行は、上の 2 行とそろっていません。(ii)は自動インデントを設定してリターンキーを押下した場合の例を示します。カーソルは前行の i の下に移ります。したがって、int z=20 行を入力すると、テキストは自動的に前の行と整列（つまり、自動インデント）します。

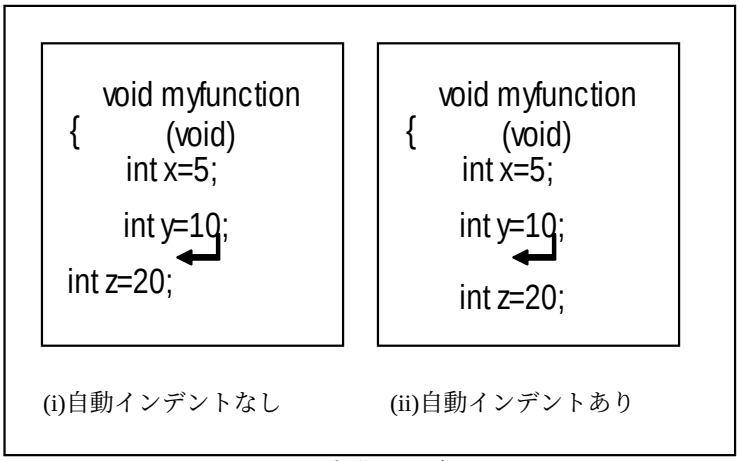


図 4.8: **自動インデント**

➲ 自動インデントを設定するには

1. [ツール->オプション...] を選んでください。[オプション]ダイアログボックスが表示されます。[エディタ]タブを選んでください（図4.7）。
2. [自動インデント] チェックボックスをチェックしてください。
3. [OK] ボタンをクリックすると、自動インデントが設定されます。

4.9 ウィンドウを分割する

HEW では、テキストウィンドウを 2 つに分割できます。図 4.9 に、テキストウィンドウの右上にある“閉じる”ボタンのすぐ下にある スプリットバーボタンを示します。

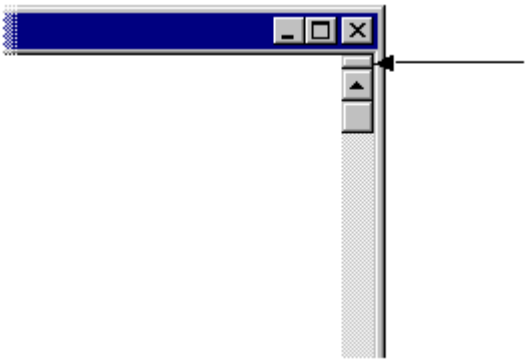


図 4.9: **スプリットバー ボタン**

➲ ウィンドウを分割するには
スプリットバーボタンをダブルクリックしてウィンドウを二つに分割するか、スプリットバーボタンを押したままマウスを下に移動して分割したい位置でマウスボタンを離してください。

➲ スプリットバーの位置を調節するには
スプリットバーボタンを押したままマウスを下に移動して、分割したい位置でマウスボタンを離してください。

➲ ウィンドウの分割を解除するには
スプリットバーボタンをダブルクリックするか、スプリットバーボタンをウィンドウの一番上か一番下に移動してください。

4.10　テキストの表示の変更方法

この節では、エディタウィンドウのテキスト表示の変更方法を説明します。

4.10.1　エディタのフォントを変更する

HEW ではエディタのフォントを指定できます。ファイルの種類にかかわらず、エディタウィンドウはすべて同じフォントを使用します。

➲ エディタのフォントを変えるには

1. [ツール->表示形式...] を選んでください。[表示形式]ダイアログボックスが表示されます。ツリーの中の[Source]アイコンを選んでください（図4.10）。
2. [フォント] リストからフォントの種類を選んでください。
3. [サイズ] リストからフォントの大きさを選んでください。
4. [OK] ボタンをクリックするとフォントの種類と大きさが設定されます。

図 4.10: **表示形式 ダイアログボックス フォント タブ**

4.11 シンタックスを色づけする

コードをより読みやすくするため、指定した文字列（キーワードなど）を異なる色で表示できます。例えば、C ソースコードのコメントを緑色で、C 言語の型（int など）を青色で表示できます。

色づけの方法は、ファイルグループ単位で指定できます。例えば、C ソースファイル、テキストファイル、マップファイル、自分のファイルでそれぞれ異なった色づけ方法を定義できます。

➲ 既存の色を変えるには

1. [ツール->表示形式...] を選んでください。[表示形式]ダイアログボックスが表示されます。
2. 色の変更をしたいアイテムを、ツリーの中のアイコンの下から選んでください。このアイテムはファイルタイプ（例：C言語のソースファイル）および適切なキーワードグループ（例：識別子またはプリプロセッサ）となります。
3. [カラー] タブを選んでください。
4. [前面] リストと [背景] リストの色を変更してください。[SYSTEM]を選択するとコントロールパネルで現在設定してある文字色と背景色になります。
5. [OK] ボタンをクリックすると新しい設定になります。

➲ 新規のキーワードグループを作るには

1. [ツール->表示形式...] を選んでください。[表示形式]ダイアログボックスが表示されます。
2. キーワードグループを追加したいファイルタイプをツリーから選んでください。
3. ツリー下の [追加…] をクリックすると [カテゴリ追加] ダイアログボックスが表示されます（図4.11）。[カテゴリ名]フィールドにキーワードグループを入力し、[OK]ボタンを押下すると新しいキーワードグループが追加されます。

図 4.11: **カテゴリ追加ダイアログボックス**

➲ 新規のキーワードを作るには

1. [ツール->表示形式...] を選んでください。[表示形式]ダイアログボックスが表示されます。
2. シンタックスのハイライト表示を変更したいアイテムを、ツリーの中のアイコンの下から選んでください。このアイテムはファイルタイプ（例：C言語のソースファイル）および適切なキーワードグループ（例：識別子またはプリプロセッサ）となります。
3. [キーワード] タブを選んでください（図4.12）。

図 4.12: **表示形式ダイアログボックス　キーワードタブ**

4. キーワードを追加するには [追加...] ボタンをクリックしてください。すると[キーワード追加]ダイアログボックスが表示されます（図4.13）。[キーワード]フィールドにキーワードを指定し[OK]ボタンを押してダイアログを閉じてください。キーワードを削除するにはキーワードを選択して[削除]ボタンをクリックしてください。

図 4.13: **キーワード追加ダイアログボックス**

新規ファイルを作成すると、デフォルトではファイルの拡張子がないためシンタックスの色付けは行いません。（エディタが自動的に新規ファイルに付ける名前には拡張子がありません）。新規ファイルにシンタックスの色付けをするには、上記の拡張子をもつ名前でファイルを保存してください。

➲ シンタックスの色付けを有効／無効にするには

1. [ツール->オプション...] を選んでください。[オプション]ダイアログボックスが表示されます。[エディタ]タブを選んでください。（図4.7）。
2. [シンタックスカラーリング] チェックボックスをチェックすると有効になり、チェックしないと無効になります。設定後、[OK]ボタンをクリックしてください。

4.12 テンプレート

ソフトウェア開発時、同じテキストを繰り返し入力する場合がよくあります。例えば、関数定義、ループ、関数の機能のコメント欄などです。HEW では、現在アクティブなエディタウィンドウに、定型テキスト（テンプレート）を挿入できます。テンプレート設定後は定型テキストを、手作業で入力するかわりに、簡単に挿入できるようになります。

4.12.1 テンプレートを設定する

➲ テンプレートを設定するには

1. [編集->テンプレート->テンプレートの定義...] を選ぶか、ローカルメニューから [テンプレート->テンプレートの定義...] を選ぶか、 [テンプレートの定義] ツールバーボタン ( ) を選んでください。図4.14 に示すダイアログボックスが表示されます。

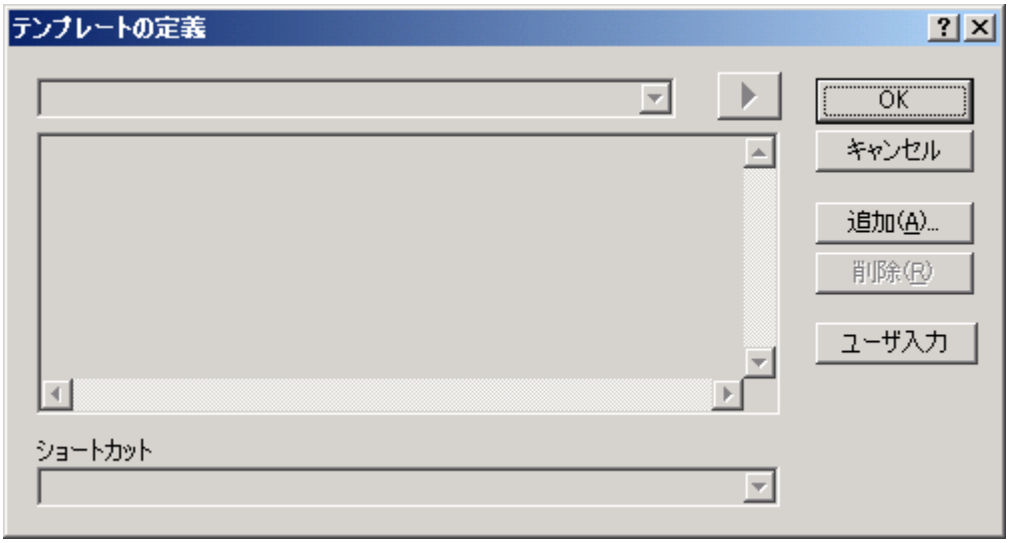


図 4.14: **テンプレートの定義 ダイアログボックス**

2. [追加…] ボタンをクリックしてください。ダイアログボックスが表示され、選択したテンプレート名を尋ねます。この場合、独自のテンプレート名を選択してください。そうしないと、複製したテンプレート名のメッセージが表示され、テンプレートは追加されません。
3. 既存のテンプレートを変更したいときは、 [テンプレート名] ドロップダウンメニューから変更するテンプレートを選んでください。
4. テンプレートテキストエリアにテキストを入力してください。他のエディタウィンドウからテキストをコピーして”CTRL+V”キーを押下してこのダイアログボックスに貼りつけることができます。
5. テンプレート用に確保されているショートカットキーは10個あります。このうち1つを使いたい場合、[テンプレートの編集]ダイアログ下部にあるドロップリストでキーを選択してください。CTRL+0 ～CTRL+9の中から選択できます。
6. テンプレートを設定するとき、次のプレースホルダを使うことができます。

表 4.2: テンプレートで使用するプレースホルダ

メニューエントリ	プレースホルダテキスト	説明
ファイルパス+ファイル名	$(FULLFILE)	ファイル名（フルパスを含む）
ファイル名	$(FILENAME)	ファイル名（パスを除き拡張子を含む）
ファイルリーフ	$(FILELEAF)	ファイル名（パスと拡張子を除く）
ワークスペース名	$(WORKSPNAME)	ワークスペース名
プロジェクト名	$(PROJECTNAME)	プロジェクト名
行番号	$(LINE)	テンプレートを挿入する最初の行番号
時間	$(TIME)	テンプレートが挿入される時間
日付，テキスト	$(DATE_TEXT)	現在の年月日をテキスト表示
日付，日/月/年	$(DATE_DMY)	現在の日/月/年
日付，月/日/年	$(DATE_MDY)	現在の月/日/年
日付，年/月/日	$(DATE_YMD)	現在の年/月/日
ユーザ名	$(USER)	現在のユーザ
カーソル位置	$(CURSOR)	挿入カーソル：テンプレートを挿入した後カーソルをこの位置に設定

7. ‘$(CURSOR)’ を入力すると、テンプレートが挿入された後のカーソルはこの位置になります。‘$(CURSOR)’ を設定しないと、テンプレートが挿入された後のカーソルはテンプレート最後の文字の後ろになります（通常の貼りつけ操作と同じ）。

ユーザ入力

テンプレートを定義する際、ユーザ入力フィールドも定義することができます。以下のプレースホルダを使って指定します。

$(USERINPUT<n:1-10>|”<*some text*>”)

n はユーザ入力識別子を表す数字です。これらのプレースホルダは手動で追加することも可能ですが、[テンプレートの定義]ダイアログの[ユーザ入力]ボタンを使うとプレースホルダを自動的に追加できます。テンプレートをファイルに挿入するとダイアログが表示され、ユーザは各フィールドにカスタムのテキストを入力することができます。その後、このテキストはプレースホルダの替わりに挿入されます。ユーザはこのようなユーザ入力フィールドを 10 個定義できます。

図 4.15 にテンプレート一覧を示します。この一覧は[Workspace]ウィンドウの[Templates]タブにあります。

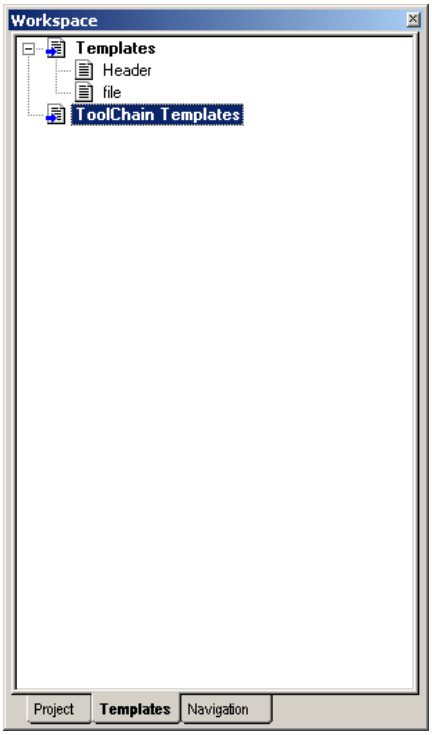


図 4.15: テンプレート一覧

HEW に追加された新しいテンプレートは、[Templates]フォルダの下に表示されます。[Toolchain Templates]フォルダには、HEW システムで使用するため現在のツールチェインによって提供される読み取り専用のテンプレートが表示されます。この一覧にあるテンプレートはドラッグしてエディタファイルに挿入できます。さらに、エディタからテキスト領域を[Templates]フォルダにドラッグして、簡単にテンプレートを作成することも可能です。この一覧でマウスの右ボタンをクリックするとポップアップメニューが表示され、ユーザは簡単に新しいテンプレートを追加したり、現在選択しているテンプレートを削除または編集したりすることができます。

4.12.2 テンプレートを削除する

➲ テンプレートを削除するには

1. [編集->テンプレート->テンプレートの定義...] を選ぶか、ローカルメニューから [テンプレート->テンプレートの定義...] を選ぶか、 [テンプレートの定義] ツールバーボタン （ ）をクリックしてください。図4.14 に示すダイアログボックスが表示されます。
2. [テンプレート名] ドロップダウンリストから削除したいテンプレート名を選び、 [削除] ボタンをクリックしてください。
3. 確認ダイアログボックスが表示された場合は、[はい]ボタンをクリックしてください。
4. [OK]ボタンをクリックすると、ダイアログボックスは終了されます。

4.12.3 テンプレートを挿入する

➲ テンプレートを挿入するには
ツールバーでテンプレートを選択し“テンプレートの挿入”ツールバーボタン( )をクリックするか、 [編集->テンプレート->テンプレートの挿入...]を選ぶか、ローカルメニューから [テンプレート->テンプレートの挿入...]を選んでください。選んだテンプレートが現在のエディタウィンドウに追加されます。

4.12.4 かっこの組み合わせ

複雑なソースコードは扱いにくいことがあります。特に、C 言語のブロックが互いに深いネスト構造になっている場合や、if 文で複雑な論理文が表現されている場合などです。HEW では、そのような場合のために、かっこの種類{ }, ( ) , [ ]ごとにかっこの間のテキストをハイライト表示できます。

➲ かっこの組み合わせを見つけるには

1. かっこの始めをハイライト表示するか、カーソルをかっこの前に置いてください。
2. [括弧の呼応] ツールバーボタン （ ） をクリックするか、 “Shift+CTRL+M”キーを押下するか、 [編集->括弧の呼応] を選ぶか、 ローカルメニューから [括弧の呼応] を選んでください。

ファイル全体の構造をチェックするために、カーソルをファイルの始めに置いて、かっこの組み合わせの操作を繰り返し行ってください。組み合わせがなくなるまで、かっこの組み合わせごとに次々とハイライト表示されます。

4.13 Editor カラムの管理

Editor のカラムは、デバッガの機能に応じて追加します。また、ユーザがカラムを表示、非表示を選択することが可能です。

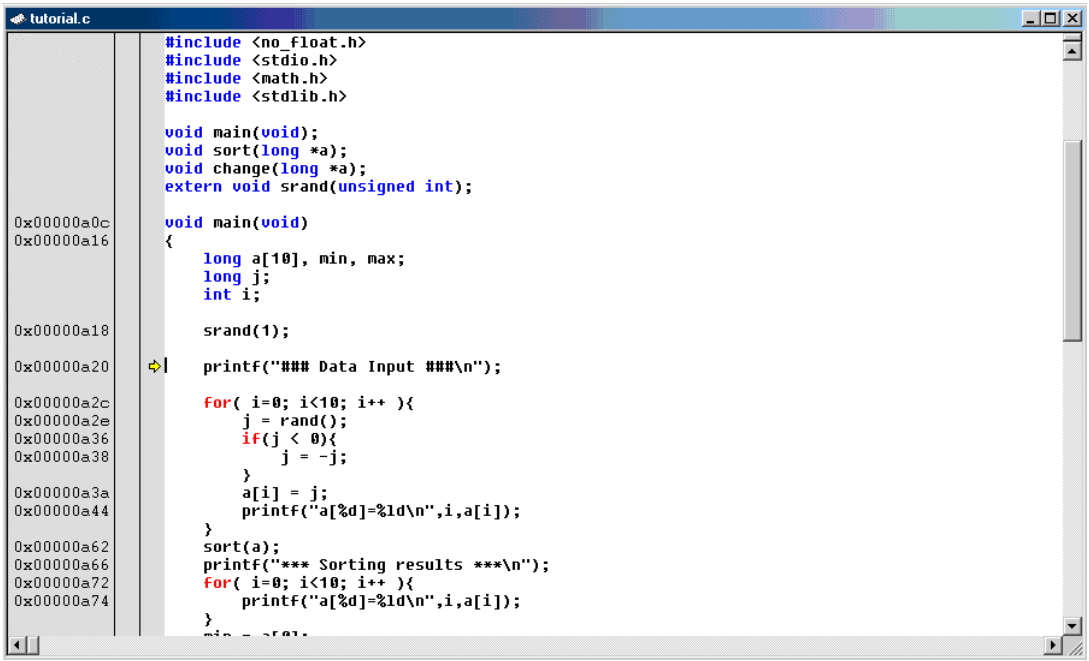


図 4.16: Editor **カラム**

➲すべてのソースファイルでカラムをオフにするには

1. [エディタ] ウィンドウを右クリックしてください。
2. [表示カラムの設定...] メニュー項目をクリックしてください。
3. [エディタ全体のカラム状態] ダイアログボックスが表示されます（図4.17）。
4. チェックボックスは、そのカラムが有効か無効かを示します。チェックされている場合は有効です。チェックボックスがグレー表示の場合、一部のファイルではカラムが有効で、別のファイルでは無効であることを意味します。
5. [OK] ボタンをクリックして、新しいカラム設定を有効にしてください。

➲1つのソースファイルでカラムをオフにするには

1. 削除したいカラムのある [エディタ] ウィンドウを右クリックしてください。
2. [カラム] メニュー項目をクリックしてください。カスケードされたメニュー項目が現れます。各カラムは、このポップアップメニューに表示されます。カラムが有効である場合、名前の横にチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。

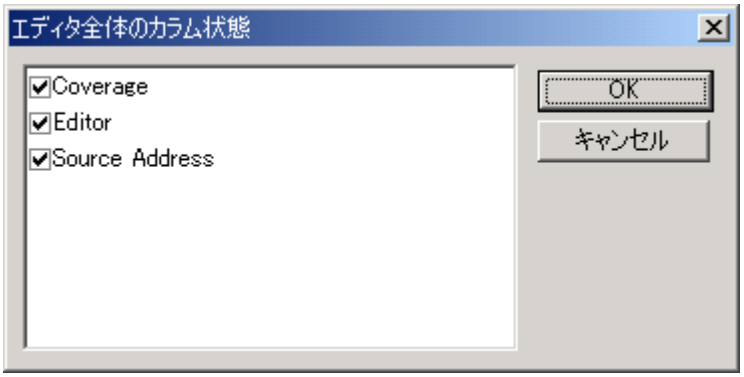


図 4.17: **エディタ全体のカラム状態ダイアログボックス**

4.14　ツールチップウォッチ

[ツール->オプション...]の[エディタ]タブを選んでください(図 4.7)。[ツールチップウォッチ]チェックボックスをチェックすると、ソースプログラム上の変数の値を早く確認するツールチップウォッチ機能を使用することができます。

➲ [エディタ]ウィンドウにツールチップウォッチを表示するには

1. 確認したい変数を含むソースファイルを[エディタ]ウィンドウに表示します（図4.18）。
2. 確認したい変数名の上にマウスのカーソルを静止させます。変数の近くに変数の値を表示したツールチップを表示します。

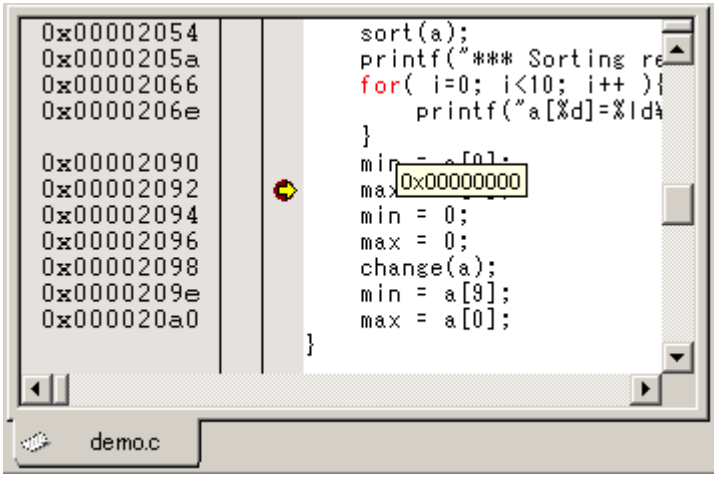


図 4.18: **ツールチップウォッチの表示**

5. ツール管理

[ツールアドミニストレーション]ダイアログボックスで、HEW で使うツールを管理します (図 5.1)。このダイアログボックスは[ツール-> アドミニストレーション…]で開きます。すべてのワークスペースが閉じているときは変更可能で、ワークスペースが開いているときは参照のみ可能です。

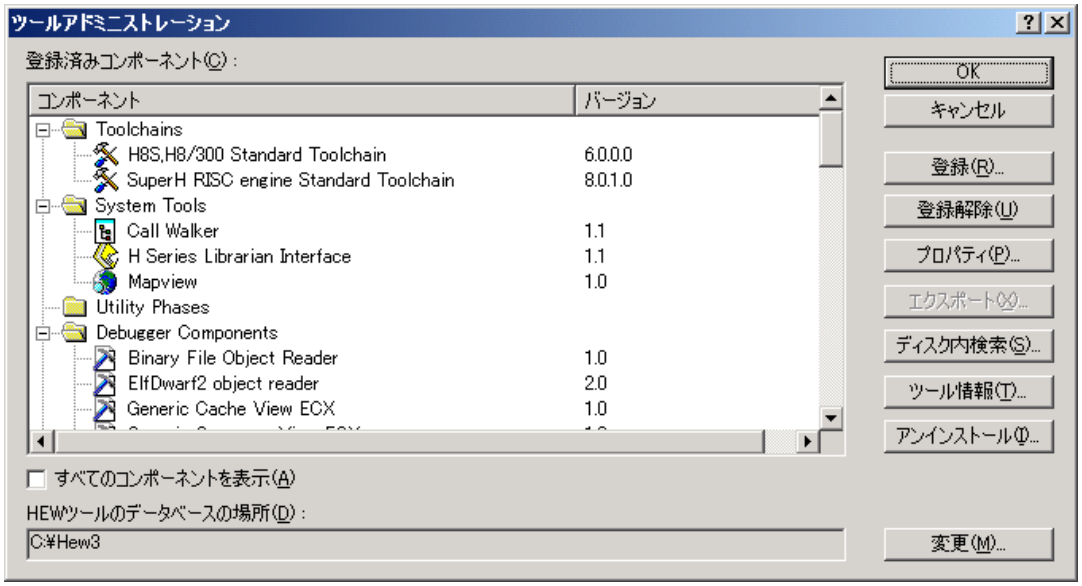


図 5.1: ツールアドミニストレーション ダイアログボックス

5 種類の標準ツールがあります。

Toolchains： 一連のビルドフェーズ （例:コンパイラ、アセンブラ、リンケージエディタ）。 ビルド機能を実現。

System Tools： [ツール]メニューから選ぶことのできるアプリケーション （.EXE）。ツールチェインをサポートする追加のアプリケーション（例: HDI などの外部デバッガまたは対話式グラフィカルライブラリアン）。

Utility Phases： 特定のビルド機能をサポートする、あらかじめ用意されたビルドフェーズ （例: ソースコードの複雑度解析、ソースコードの行カウントなど）。特定のツールチェインに依存しない追加のビルド機能。

Debugger Components： 特定のデバッガ機能をサポートするツール（例: ターゲットプラットフォーム、オブジェクトリーダなど）。

Extension Components： HEW システムのある領域における特定のキー機能をサポートするツールです。これらのツールはインストールすると、必ず登録されます（例: HEW ビルダ、デバッガ、フラッシュサポート）。

5.1 ツールの位置

HEW では、新しいツールがインストールされるたびに HEW との連動に必要なツールの位置を自動的に保持します。インストール後、HEW はそのツールに関する情報（位置を含む）を保持します。これを登録と呼びます。初期登録は自動で行いますが、開発の途中で、プロジェクトのツールをより効率良く利用するためにユーザ自身でツールを登録することが必要になることがあります。この章では登録について説明します。

5.2 HEW 登録ファイル (*.HRF)

HEW と互換性のあるツール（ツールチェイン、システムツール、またはユーティリティフェーズ）をインストールすると、拡張子**.HRF**（図 5.2 i）のファイルもインストールされます。この拡張子は“**H**EW **R**egistration **F**ile”の略であり、.HRF ファイルには HEW への登録に必要な情報が記述されています。登録するには、そのツールの.HRF ファイルを[ツールアドミニストレーション]ダイアログボックスにロードします（図 5.2 ii）。

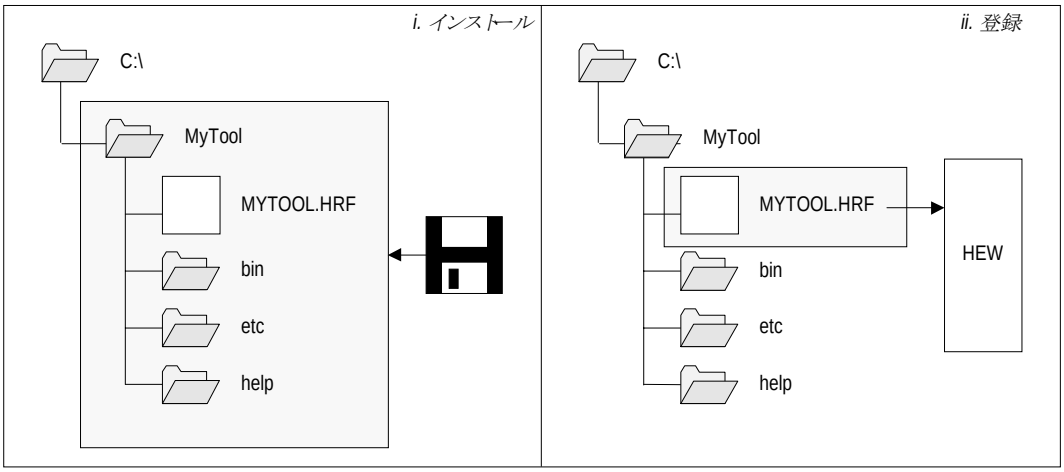


図 5.2: HRF **ファイルの位置と登録**

HEW でツールを使うには、まず登録が必要です。[ツールアドミニストレーション]ダイアログボックス（図 5.1）は現在登録されているツールを表示します。このダイアログボックスを開くには、ワークスペースがすべて閉じていることを確認して[ツール-> アドミニストレーション…]を選んでください。ワークスペースを開いた状態で[ツールアドミニストレーション]ダイアログボックスをアクセスすると、ダイアログが開きますが変更はできません。HEW がデフォルトでインストールされると、新しいツールは自動的に登録されます。

HEW はツールデータベースファイルに、ツール情報を格納します。デフォルトでは、このファイルは HEW アプリケーションディレクトリに作成されます。しかし、ネットワーク環境で作業を行っている場合は、このディレクトリは他の場所に設定されることがあります。ツールディレクトリの場所は変更が可能です。

➲ ツールの場所を変更するには

1. [ツール-> アドミニストレーション…] を選んでください。
2. [HEWツールのデータベースの場所] の [変更…] ボタンをクリックしてください。
3. 新しいツールがあるディレクトリのルートディレクトリを選択し、 [OK] ボタンをクリックしてください。
4. ディレクトリが切り替わり、ツールの場所が新しいディレクトリに移ります。この場所にある新しいツールはスキャンする必要があります。スキャンには、スキャンディスクまたはレジスタツール機能を使用します。

5.3 ツールを登録する

HEW は起動後にインストールしたすべてのツールを自動的に登録します。しかし、ときには、ユーザがツールを登録する必要があります。

5.3.1 ドライブのツール検索

ドライブを検索して HEW に互換性のあるツールを見つけることは、HEW のツールインストール情報が削除されたり破壊されたりしたときなどに有益です。なぜなら、ツール情報を再びすぐに作成することができるからです。

➲ ツールを検索するには

1. [ツールアドミニストレーション] ダイアログボックス（図5.1）の [ディスク内検索…] ボタンをクリックすると [コンポーネントのディスク内検索] ダイアログボックスが表示されます（図5.3）。

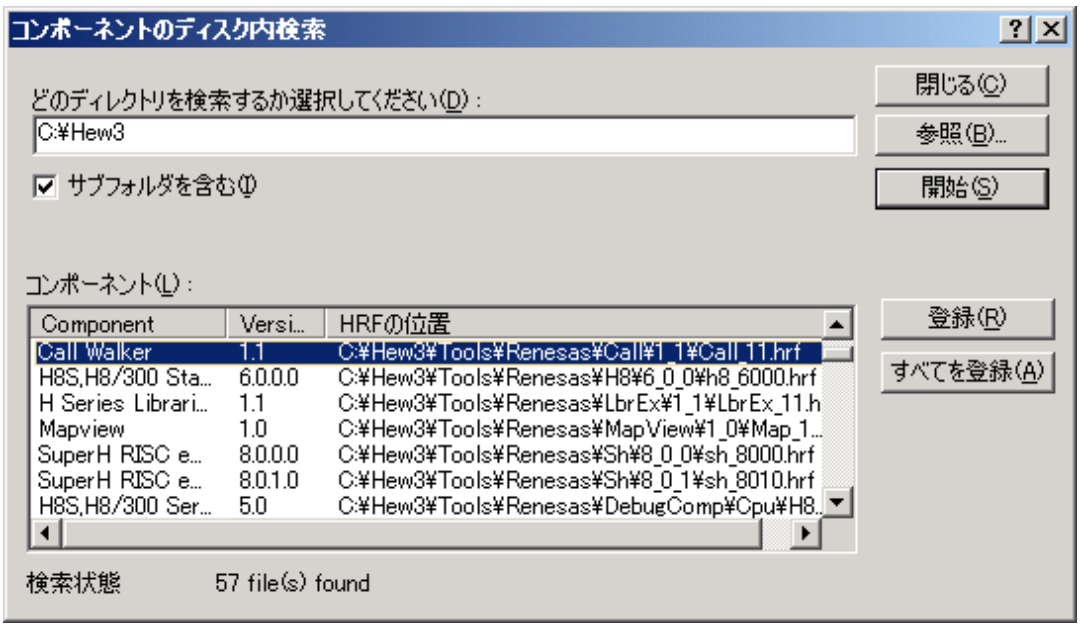


図 5.3: **コンポーネントのディスク内検索 ダイアログボックス**

2. [どのディレクトリを検索するか選択してください] フィールドに、検索するディレクトリを入力してください。または、 [参照…] ボタンをクリックしてディレクトリを選んでください。
3. [サブフォルダを含む] チェックボックスをチェックすると、指定したディレクトリとその下のディレクトリをすべて検索します。
4. [開始] ボタンをクリックすると検索を始めます。検索中、 [開始] ボタンは [中断] ボタンに変わります。検索を途中で止めるときには [中断] ボタンをクリックしてください。
5. [コンポーネント] リストに検索結果を表示します。個別にツールを登録するにはそのツールを選んで [登録] ボタンをクリックしてください。すべてのツールを登録するには [すべてを登録] ボタンをクリックしてください。
6. [閉じる] ボタンをクリックするとダイアログボックスを終了します。

5.3.2 ツールを一つ登録する

HEW では、ツールを検索しなくても、ツールを一つずつ登録できます。HEW 登録ファイル (*.HRF) はツールがインストールされたルートディレクトリにあります。

➲ ツールを登録するには

1. [登録…] ボタンをクリックすると標準のファイルを開くダイアログボックスが開きます。フィルタが [HEW Registration Files (*.hrf)] に設定されています。
2. 登録するファイルの.HRFファイルをアクセスして選び、 [選択] ボタンをクリックしてください。
3. 選んだツールに関する情報を示すダイアログボックスが表示されます。ツールを登録するには [登録] ボタンをクリックしてください。登録しない場合は [閉じる] ボタンをクリックしてください。

5.4 ツールの登録を取り消す

登録したツールによって、HEW は影響を受けます。例えば、新しいプロジェクト作成時に、登録された互換性のあるすべてのシステムツールが、ツールメニューに追加されます。ときにはユーザにとっては、これによって効率が下がり、使いにくいかもしれません。そのようなときは、登録を取り消すことができます。[ツールアドミニストレーション]ダイアログボックスでツールを選び [登録解除] ボタンをクリックしてください。確認のダイアログボックスが表示されます。 登録を取り消す場合は[はい] をクリックしてください。

注意 ツールの登録を取り消しても、ハードディスクからツールがなくなることはありません。単に、HEW に格納されているそのツールに関する情報を削除するだけです（HEW から切断されます）。この動作はいつでも元に戻すことができ、ツールを再登録できます（前節「ツールを登録する」参照）。ハードディスクからツールを削除（アンインストール）したいときはこの章後半の「ツールのアンインストール」を参照してください。

5.5 ツールのプロパティの参照と編集

ツールに関する情報を参照するには、ツールを [登録済みコンポーネント] リストから選んで [プロパティ…] ボタンをクリックしてください。[一般] タブを選んだ状態で （図 5.4）プロパティのダイアログボックスが表示されます。このタブでは、名前、バージョン、位置を示します。このタブは編集できません。

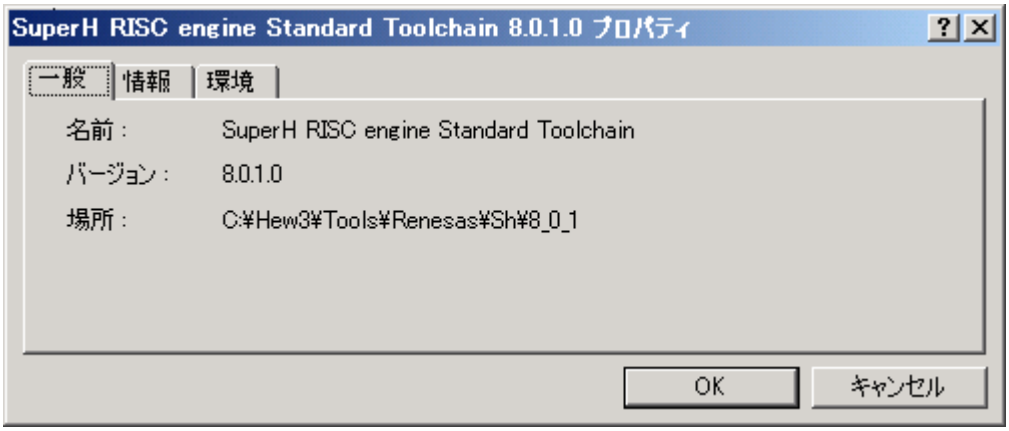


図 5.4: プロパティのダイアログボックス 一般 タブ

ツールについての情報を参照するには [情報] タブをクリックしてください（図 5.5）。例えば、著作権、履歴、バグの修正、ユーザへのお知らせなどが表示されます。

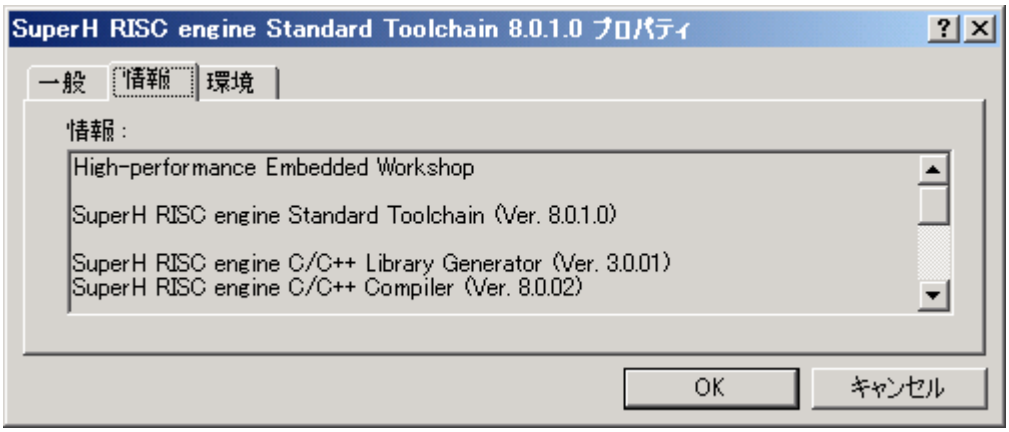


図 5.5: プロパティのダイアログボックス 情報 タブ

ツールの互換性に問題がある場合、この[情報]エディットフィールドで問題を報告することがあります。

ツールの環境設定を参照、編集するには [環境] タブを選んでください（図 5.6）。このダイアログボックスは、ツールチェインの環境を変更するとき、もっとも一般的に使われています。

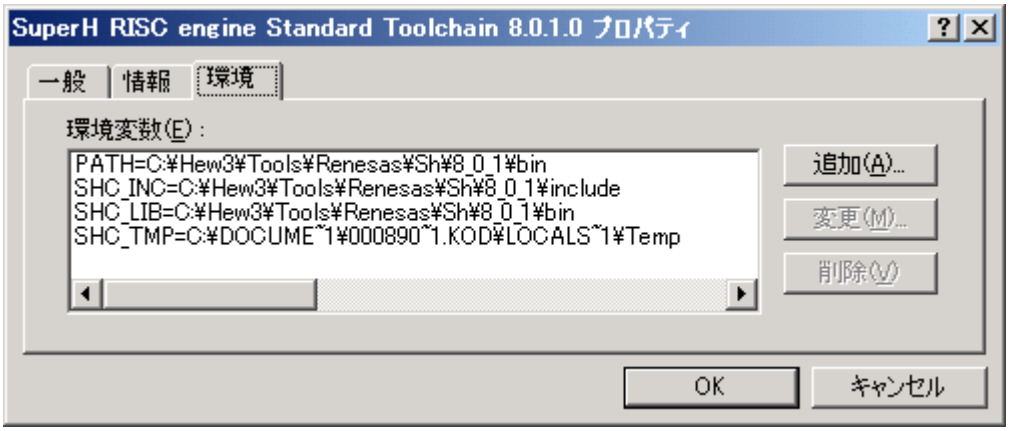


図 5.6: **プロパティのダイアログボックス 環境 タブ**

新しい環境変数を追加するには、[追加...]ボタンをクリックしてください。図 5.7 に示すダイアログボックスが表示されます。[変数]フィールドに変数名を入力して[値]フィールドに変数の値を入力して [OK]ボタンをクリックしてください。環境変数の値を柔軟に指定できるようにするため、プレースホルダポップアップメニューがあります。プレースホルダの詳細は、付録 C 「プレースホルダ」を参照してください。

環境変数を変更するには、[環境]タブで変更する環境変数を選び[変更...]ボタンをクリックしてください。[変数] フィールドと [値]フィールドを必要に応じて変更して [OK]ボタン をクリックすると、変更した環境変数が[環境]タブに加わります。環境変数を削除するには、その環境変数を選び [削除] ボタンをクリックしてください。

図 5.7: **環境変数 ダイアログボックス**

5.6 ツールのアンインストール

HEW には登録されていないツールをハードディスクから削除するための、アンインストール方法があります。

➲ ツールをアンインストールするには

1. [ツール->アドミニストレーション…] を選んでください。
2. [アンインストール] ボタンをクリックしてください。[HEWツールのアンインストール]ダイアログボックスが表示されます(図5.8)。

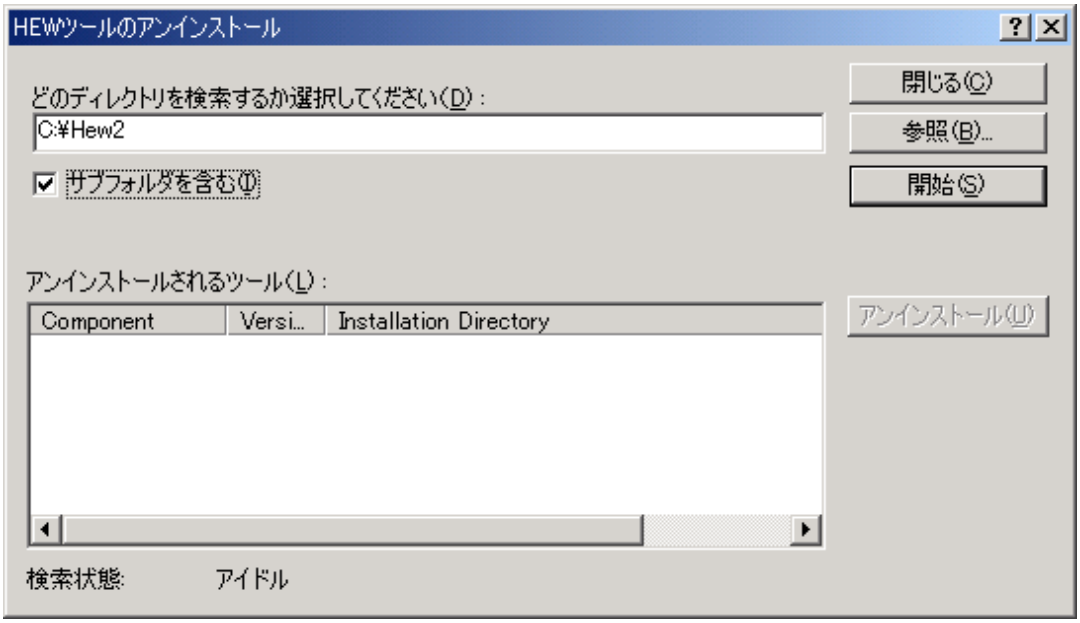


図 5.8: HEW **ツールのアンインストールダイアログボックス**

3. 一番上のフィールドに検索するディレクトリを入力するか、 [参照…] ボタンをクリックしてブラウズしてください。
4. [サブフォルダを含む] チェックボックスをチェックすると、指定したディレクトリの下のディレクトリをすべて検索します。
5. [開始] ボタンをクリックすると検索を始めます。検索中、 [開始] ボタンは [中断] ボタンに変わります。 [中断] ボタンをクリックすると、検索の途中でも検索を中止します。
6. 検索結果は [アンインストールされるツール] リストに表示されます。ツールを選んで [アンインストール] ボタンをクリックするとツールをアンインストールします。
7. [閉じる] ボタンをクリックしてダイアログボックスを終了してください。

現在 HEW で登録されていないツールのみアンインストールできます。登録されているツールをアンインストールしようとすると、図 5.9 に示すダイアログボックスが表示されます。このようなときは、[ツール->アドミニストレーション…]で HEW の[ツールアドミニストレーション]ダイアログボックスに戻り、ツールの登録を取り消してから、アンインストールを再実行してください。

図 5.9:**アンインストール不可のダイアログボックス**

HEW から登録をはずしたツールを選んで[登録解除] ボタンをクリックすると、図 5.10 に示す確認のダイアログボックスが表示されます。このダイアログボックスには削除されるすべてのファイルやフォルダが表示されます。これらのファイルやフォルダを削除してよいことを確認して [はい] ボタンをクリックしてください。アンインストールを中止するときは、 [いいえ] または [キャンセル] ボタンをクリックしてください。

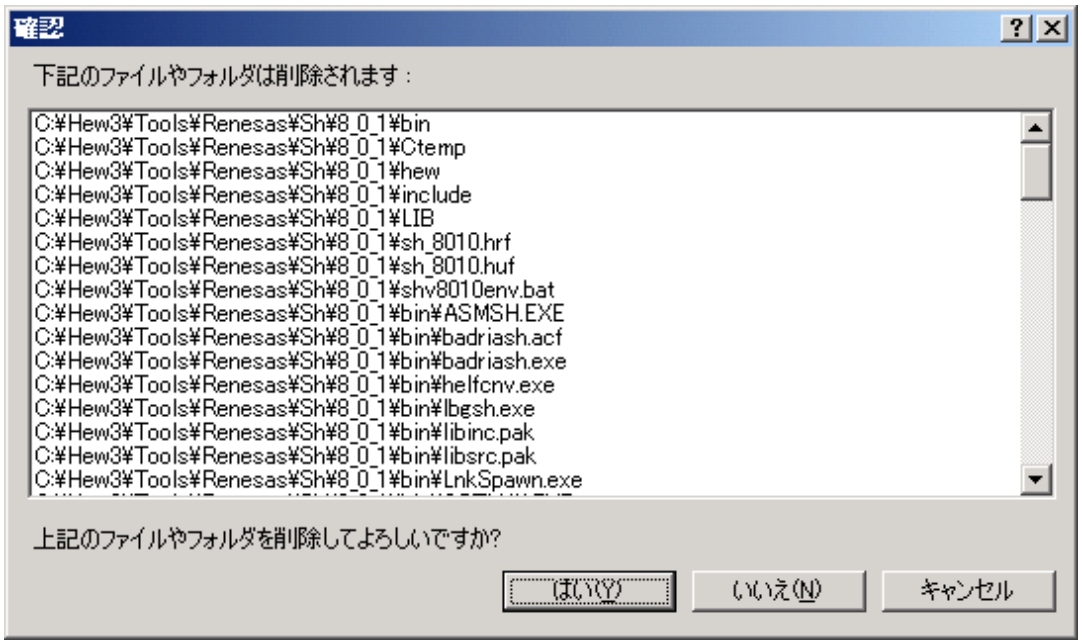


図 5.10: **確認ダイアログボックス**

5.7 テクニカルサポートについて

[ツールアドミニストレーション] ダイアログボックスでは、“隠れている”システムツールに関する情報を表示できます。これらのツールは HEW の一部であり、手動で登録または登録取り消しすることができません。[ツールアドミニストレーション]ダイアログボックスの [すべてのコンポーネントを表示]チェックボックスをチェックすると、隠れていたツールフォルダを表示します （図 5.11）。

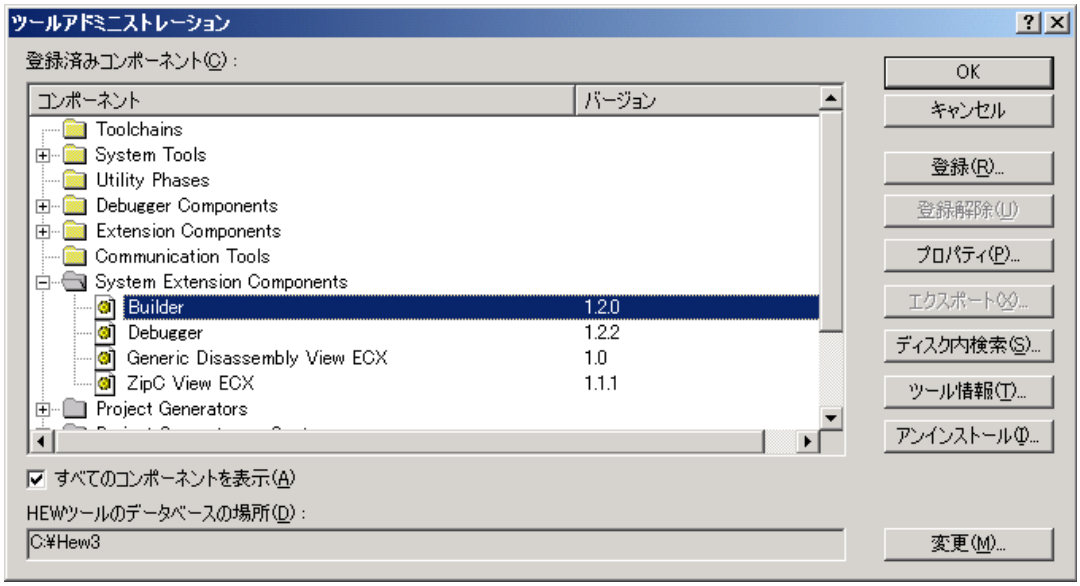


図 5.11: すべてのツールの表示

テクニカルサポートを受ける際、ツールに関する詳細をお尋ねすることがあります。そのときには、そのツールのフォルダを開き、ツールを選び、 [プロパティ] ボタンをクリックしてください。ここで表示されるダイアログボックスはこの章の前半で説明したものと同じように動作します。ただし、[環境]タブはありません。

HEW にはまた、登録されたツールに関する情報をファイルに出力する機能があるので、HEW システム全体の情報を得ることができます。もし HEW で問題がありましたら、テクニカルサポート担当宛にこの情報を送付してください。

➲ ツール情報を出力するには

1. [ツール->アドミニストレーション] を選んでください。
2. [ツール情報] ボタンをクリックすると、 [ツール情報ファイルの保存] ダイアログボックスが表示されます。
3. ファイルの場所を選び、 [保存] ボタンをクリックしてください。
4. HEWに現在登録されているツールの情報を選んだ場所にファイル出力します。

また、問題のあるコンポーネントは、[ツールアドミニストレーション]ダイアログボックスで確認できます。一つは登録済みコンポーネントが見つからない場合で、図 5.12 のアイコンを表示します。もう一つは、コンポーネントは見つかったが、古いバージョンであったり他の依存するコンポーネントが使用不可であったりして使用できない場合で、図 5.13 のアイコンを表示します。

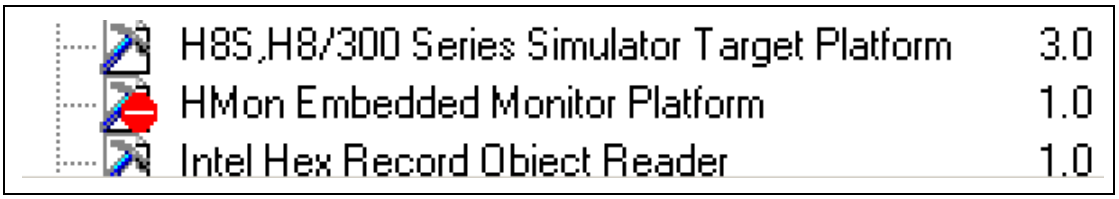


図 5.12: [Component not found] **アイコン**

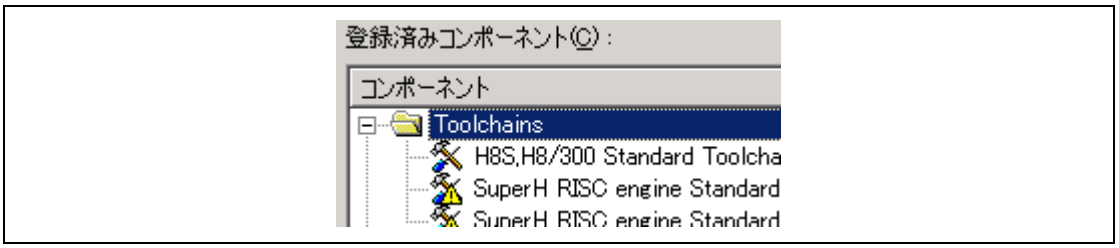


図 5.13: [Incompatible component found] **アイコン**

注意　ツールにいずれかのエラーがある場合、以下の方法でさらに情報を得ることができます。

➲ ツールエラーのフィードバックを取得するには

1. [ツール->アドミニストレーション] を選んでください。
2. 問題のあるツールをリストから選択してください。
3. [プロパティ] ボタンをクリックしてください。
4. [情報]タブを選択し、エディットフィールドを下までスクロールしてください。
5. この領域に問題の原因が表示されます。

5.8 オンデマンドのコンポーネント

HEW version 3.0 以降にはオンデマンドのコンポーネントという概念があります。これらのコンポーネントはアプリケーションやデバッガコンポーネントが自動的にロードするわけではありません。プロジェクト生成過程の一部として、ユーザがロードするものです。

➲ オンデマンドのコンポーネントを手動でロードまたはアンロードするには

1. [プロジェクト->コンポーネント...] を選んでください。
2. [コンポーネントギャラリー] ダイアログボックスが表示されます。これを図5.14に示します。
3. ロードしたいコンポーネントを選択して [ロード] ボタンをクリックしてください。コンポーネントのイメージはロードされた状態に変わります。
4. コンポーネントをアンロードしたい場合は、コンポーネントを選択して [アンロード] ボタンをクリックしてください。コンポーネントのイメージはアンロードの状態に変わります。
5. [OK] ボタンをクリックし、変更を確認してください。

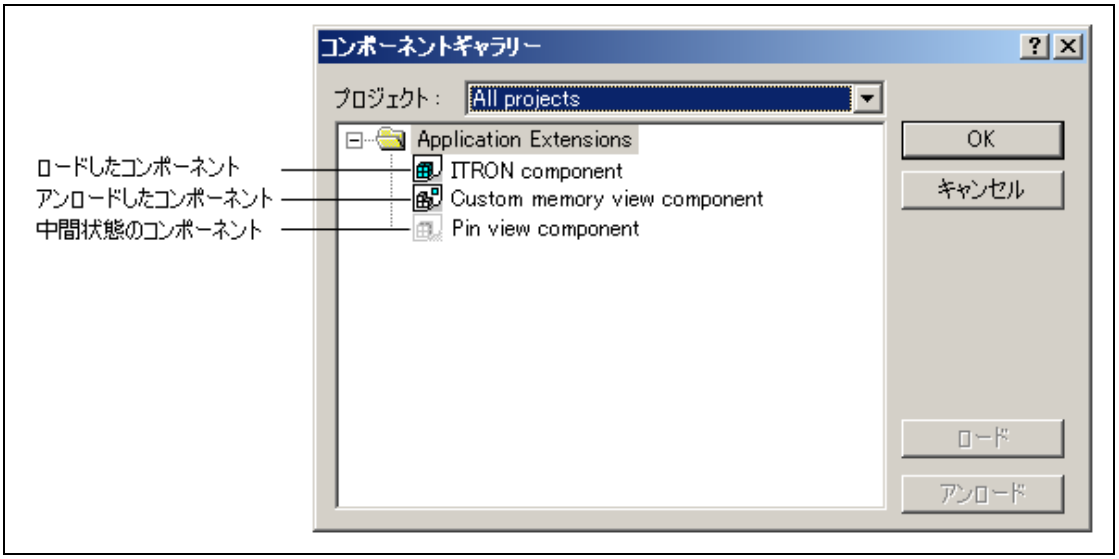


図 5.14: **[コンポーネントギャラリー] ダイアログボックス**

注意 **ユーザのワークスペースにある各プロジェクトには、ロード／アンロードされた異なるコンポーネントを格納できます。複数のプロジェクトがある場合は、[複数のプロジェクト]および[すべてのプロジェクト]を使うことで、複数のプロジェクトにわたってコンポーネントのロード状態を変更することができます。コンポーネントがあるプロジェクトでロードされ、他のプロジェクトではロードされない状態をユーザが選択した場合は、[中間状態]アイコンが表示されます。**

5.9 カスタムプロジェクトタイプ

HEW の [プロジェクト->プロジェクトタイプの作成...] メニュー機能で、現在のプロジェクトの設定を利用してプロジェクトのテンプレートを作成することができます。このテンプレートをカスタムプロジェクトジェネレータと呼びます。新しいプロジェクトタイプの名称とプロジェクト生成ウィザードのスタイルを指定できます。作成したカスタムプロジェクトジェネレータは、[ツールアドミニストレーション] ダイアログボックスで表示できます。カスタムプロジェクトジェネレータを他のユーザが使用するには、[ツールアドミニストレーション] ダイアログボックスで該当するカスタムプロジェクトジェネレータを選択して、[エクスポート...] ボタンをクリックしてください。カスタムプロジェクトジェネレータの実行環境がインストール可能な実行ファイルにまとまります。このファイルを目的とする他のマシンで実行すれば、カスタムプロジェクトジェネレータがインストールされます。

一度カスタムプロジェクトタイプを作成すると、その後[新規プロジェクトワークスペース]ダイアログボックスに表示されます。これを使用してプロジェクトの複製を作成することができます。

➲ カスタムプロジェクトジェネレータを作成するには

1. カスタムプロジェクトジェネレータに変換したいプロジェクトを含むワークスペースを開いてください。
2. 現在のプロジェクトとして変換したいプロジェクトを設定してください。
3. [プロジェクト->プロジェクトタイプの作成...] を選んでください。
4. 以下のダイアログボックスが表示されます。

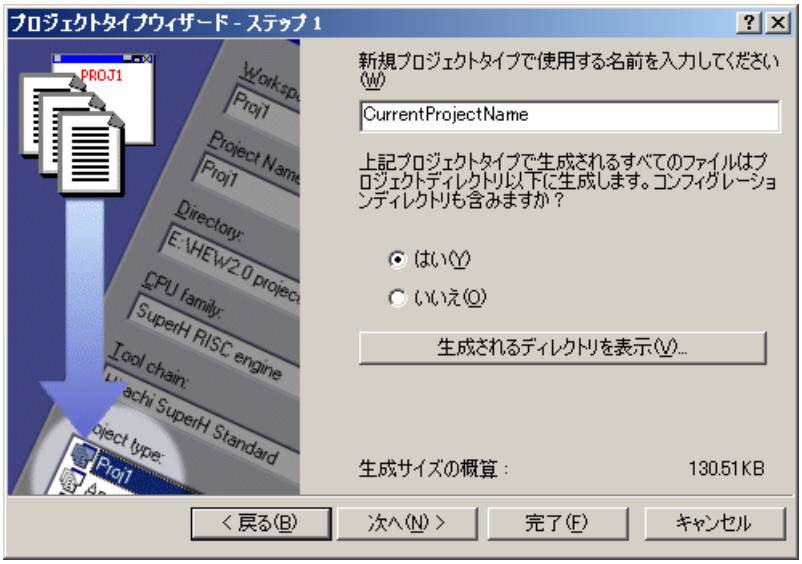


図 5.15: [プロジェクトタイプウィザード - ステップ 1] ダイアログボックス

5. 編集フィールドに現在のプロジェクト名が設定されます。これが、[新規プロジェクトワークスペース]ダイアログボックスに表示される、新しいプロジェクトジェネレータのプロジェクトタイプ名になります。必要に応じて変更してください。
6. コンフィグレーションディレクトリを含むかどうか選択することができます。コンフィグレーションディレクトリ内のファイルは単なる中間ファイルであるため、ここでは含めない

のが一般的です。

7. [次へ] ボタンをクリックしてください。以下のダイアログボックスが表示されます。

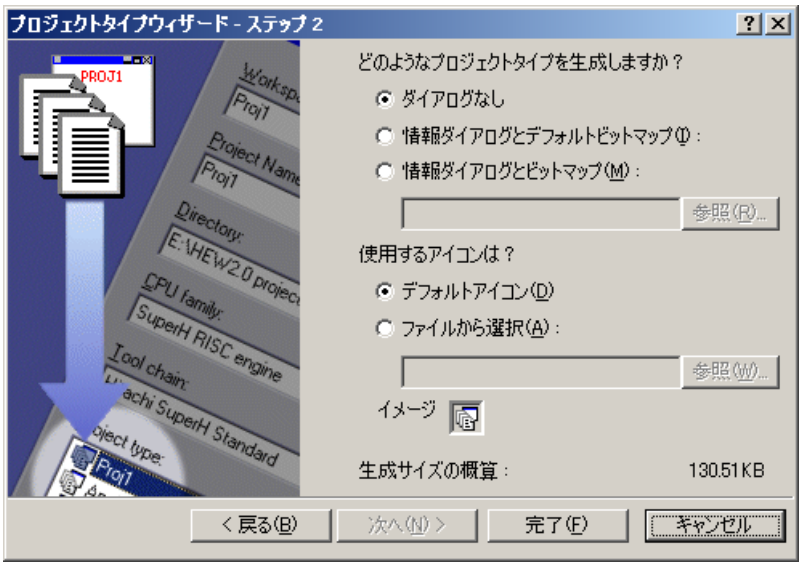


図 5.16: [**プロジェクトタイプウィザード - ステップ** 2] **ダイアログボックス**

8. ウィザードのグラフィックとプロジェクトタイプのアイコンを変更することができます。[ダイアログなし]を選ぶと、 [Project information] ダイアログボックス（図5.19参照）は表示しません。 [Project information] ダイアログボックスを表示する場合は、 [情報ダイアログとデフォルトビットマップ]と[情報ダイアログとビットマップ]のいずれかを選んでください。 [Project information] ダイアログボックス内のグラフィックにデフォルトのグラフィックを使用するか、カスタムのグラフィックを使用するかを選択できます。
9. プロジェクトタイプのアイコンを変更することもできます。デフォルトのアイコンが表示されていますが、カスタムのアイコンを使用することができます。これは [新規プロジェクトワークスペース] ダイアログボックスに表示されます。
10. ここで [Project information] ダイアログボックスを表示しない場合は、 [完了] ボタンをクリックしてください。表示する場合は、 [次へ] ボタンをクリックしてください。次のダイアログボックスが表示されます（図5.17）。

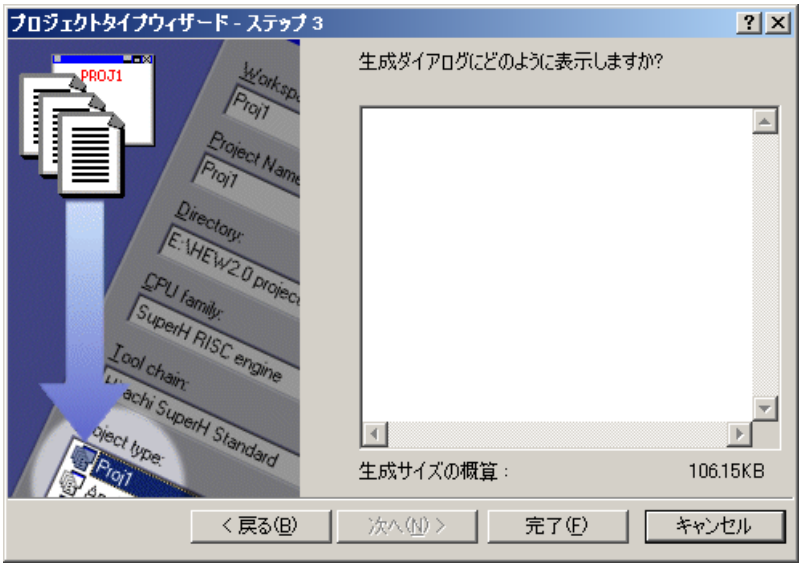


図 5.17: [**プロジェクトタイプウィザード - ステップ** 3] **ダイアログボックス**

11. プロジェクト情報を入力してください。 [Project information] ダイアログボックスの [プロジェクト情報] に表示されます。
12. [完了] ボタンをクリックしてください。

HEW インストールディレクトリ下の¥system¥pgc ディレクトリにカスタムプロジェクトジェネレータが作成されます。

カスタムプロジェクトジェネレータを作成すると、新しいプロジェクトタイプ（ここでは“CurrentProjectName”）が [新規プロジェクトワークスペース] ダイアログボックスに表示されます。

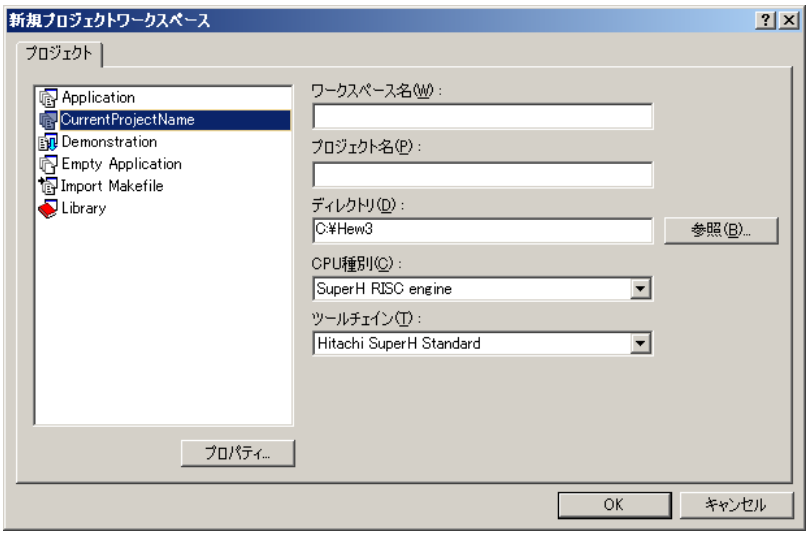


図 5.18: **[新規プロジェクトワークスペース] ダイアログボックス**

プロジェクトタイプ、ワークスペース名、プロジェクト名を設定し、 [OK] ボタンをクリックしてください。

[Project information] ダイアログボックスは、[プロジェクトタイプウィザード - ステップ 2] ダイアログボックスと[プロジェクトタイプウィザード - ステップ 3] ダイアログボックスで設定した内容を表示します。

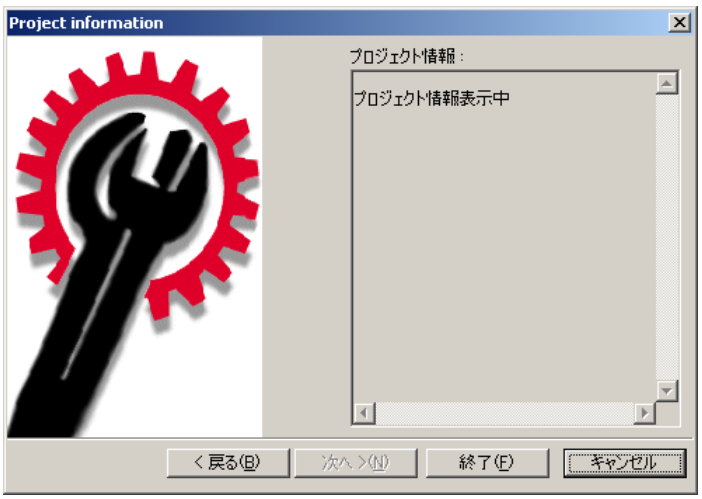


図 5.19: [Project information] **ダイアログボックス**

複製したプロジェクトを HEW システムからエクスポートし、そのプロジェクトを使用する他のユーザに渡すことができます。

➲ カスタムプロジェクトジェネレータを他のユーザにエクスポートするには

1. [ツール->アドミニストレーション] を選んでください。
   [すべてのコンポーネントを表示] チェックボックスをチェックしてください。
2. [Project Generators – Custom] フォルダを拡張してください。
3. エクスポートするカスタムプロジェクトジェネレータを選択してください。
4. [エクスポート…] ボタンをクリックしてください。

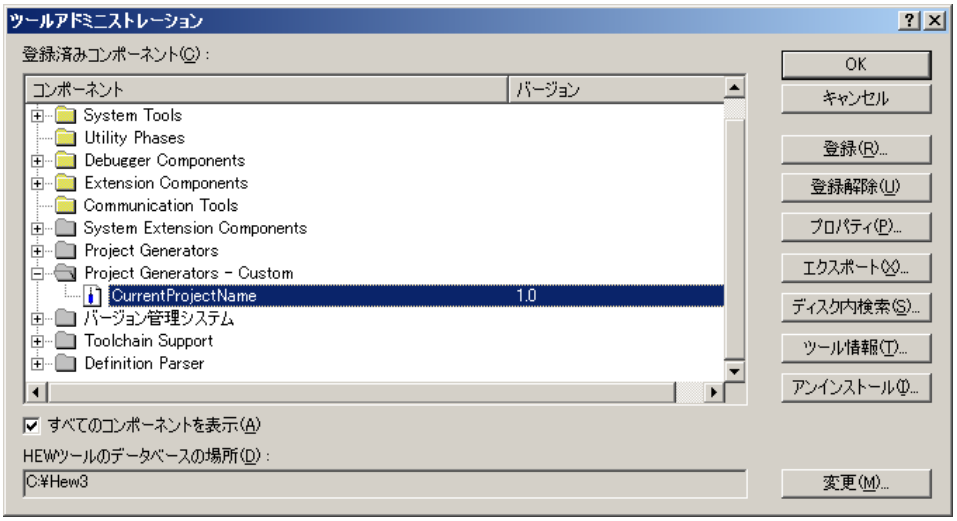


図 5.20: [ツールアドミニストレーション] ダイアログボックス

5. 表示されるダイアログボックスで、新しいカスタムプロジェクトジェネレータのエクスポート先のディレクトリを指定し、 [エクスポート] ボタンをクリックしてください。他の HEW 環境でカスタムプロジェクトジェネレータを使用するために必要なファイルセットを作成します。他のマシン環境へエクスポートする場合は、このダイアログボックスで指定したディレクトリ下のすべてのファイルが必要になります。

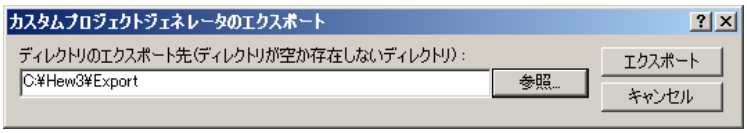


図 5.21: [カスタムプロジェクトジェネレータのエクスポート] ダイアログボックス

他の HEW 環境へカスタムプロジェクトジェネレータをエクスポートするには、エクスポートしたディレクトリ下にある“Setup.exe”を実行し、新しいユーザの HEW の場所を指定します。 [Install] ボタンをクリックすると、HEW インストールディレクトリ下の¥system¥pgc ディレクトリにカスタムプロジェクトジェネレータがインストールされます。

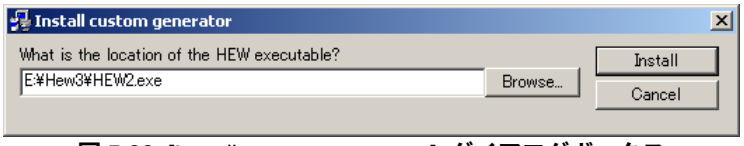


図 5.22: [Install custom generator] **ダイアログボックス**

エクスポートしたプロジェクトタイプが [新規プロジェクトワークスペース] ダイアログボックスに表示されます。

注意 **カスタムプロジェクトジェネレータは、同等もしくはそれ以上のバージョンの** HEW **でしか使用できません。古い** HEW **でこのプロジェクトジェネレータを使用すると、エラーメッセージが表示されます。**

使用しているプロジェクトジェネレータがエクスポート先の HEW **への異なるツールチェインを含んでいる場合、プロジェクトを最初に作成したときにアップグレードされます。**

6. 環境のカスタマイズ

6.1 ツールバーのカスタマイズ

HEW では 2 つのツールバーを標準で提供します。また、[カスタマイズ] ダイアログボックス（図 6.1）を使用して、新しいツールバーを作成することができます。

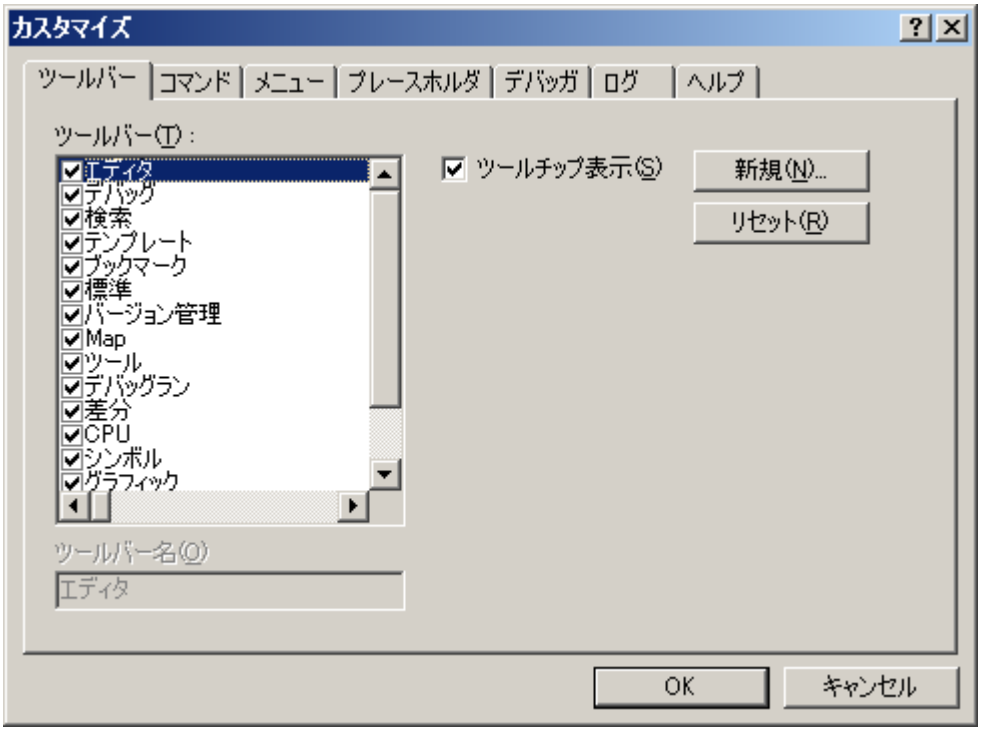


図 6.1: **カスタマイズ ダイアログボックス ツールバー タブ**

➲ 新しいツールバーを作るには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。
2. [新規...] ボタンをクリックしてください。図6.2に示すダイアログボックスが表示されます。
3. [ツールバー名] フィールドに新しいツールバー名を入力してください。
4. [OK] ボタンをクリックすると新しいツールバーが作成されます。

図 6.2: **新規ツールバー ダイアログボックス**

作成した新しいツールバーにはボタンがなく、位置を固定せずに表示されます。

➲ ツールバーにボタンを追加するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。 [コマンド] タブを選んでください（図6.3参照）。
2. [カテゴリ] リストからボタンのカテゴリーを選び、使用できるボタンを参照してください。[ボタン]エリアからボタンを選ぶとそのボタンの機能が表示されます。
3. ボタンをクリックしてダイアログボックスからツールバーにドラッグしてください。

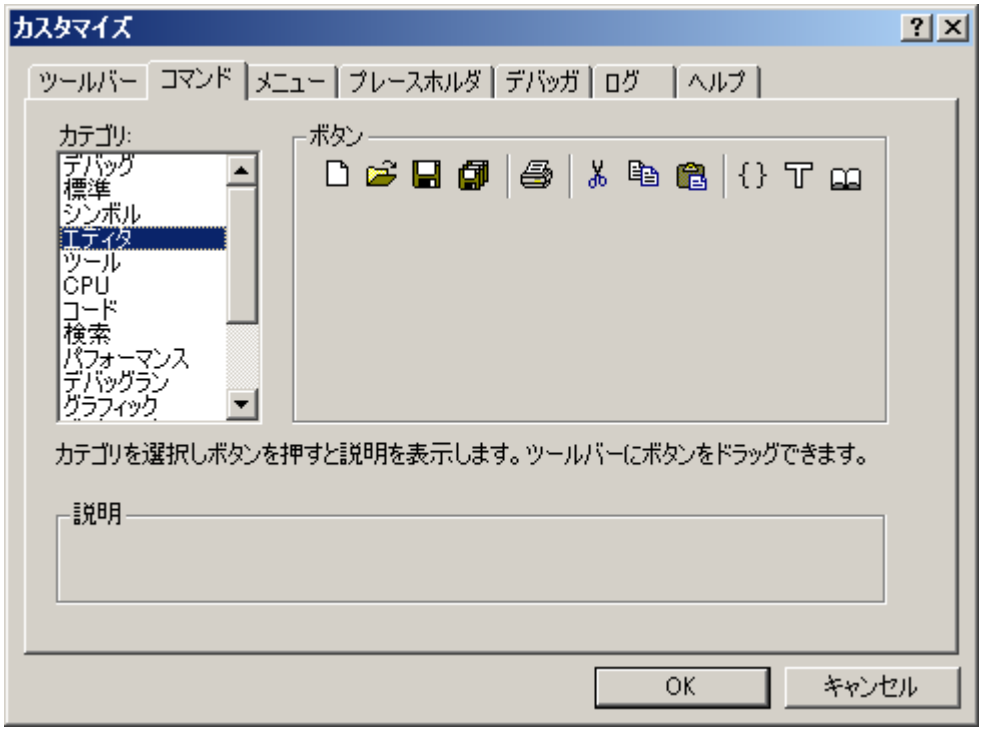


図 6.3: **カスタマイズ ダイアログボックス コマンド タブ**

➲ ツールバーからボタンを削除するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。 [コマンド] タブ （図6.3参照）を選んでください。
2. ボタンをツールバーから [ボタン] エリアへドラッグしてください。

➲ ユーザ定義のツールバーを削除するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。
2. [ツールバー] リストからユーザ定義のツールバーを選ぶと図6.1の [リセット] ボタンが [削除] ボタンに変化します。[削除] ボタンをクリックしてください。

➲ 標準のツールバーを初期状態に戻すには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。
2. [ツールバー] リストから標準のツールバーを選んで [リセット] ボタンをクリックしてください。

➲ ツールバーのツールチップを表示/非表示するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。
2. [ツールチップ表示] チェックボックスをチェックすると表示、チェックしないと非表示になります。

➲ ユーザが作成したツールバーのツールバー名を変えるには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。
2. [ツールバー] リストでユーザが作成したツールバーで名前を変更したいものを選んでください。
3. [ツールバー名] フィールドでツールバー名を変更してください。

6.2 ツールメニューのカスタマイズ

ツールメニューをカスタマイズして新しいメニューオプションを含めることができます。

➲ 新しいメニューオプションを追加するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。 [メニュー] タブを選んでください（図6.4）。最初に、全ワークスペースに適用できるグローバルアプリケーションワイドツールを追加する（[アプリケーション内有効:] に追加）か、または現在のワークスペースのみに適用できるワークスペースワイドツールを追加する（[ワークスペース内有効:] に追加）かどうかを決めてください。一度決定したら、ダイアログボックスの該当部分を選ぶようにしてください。

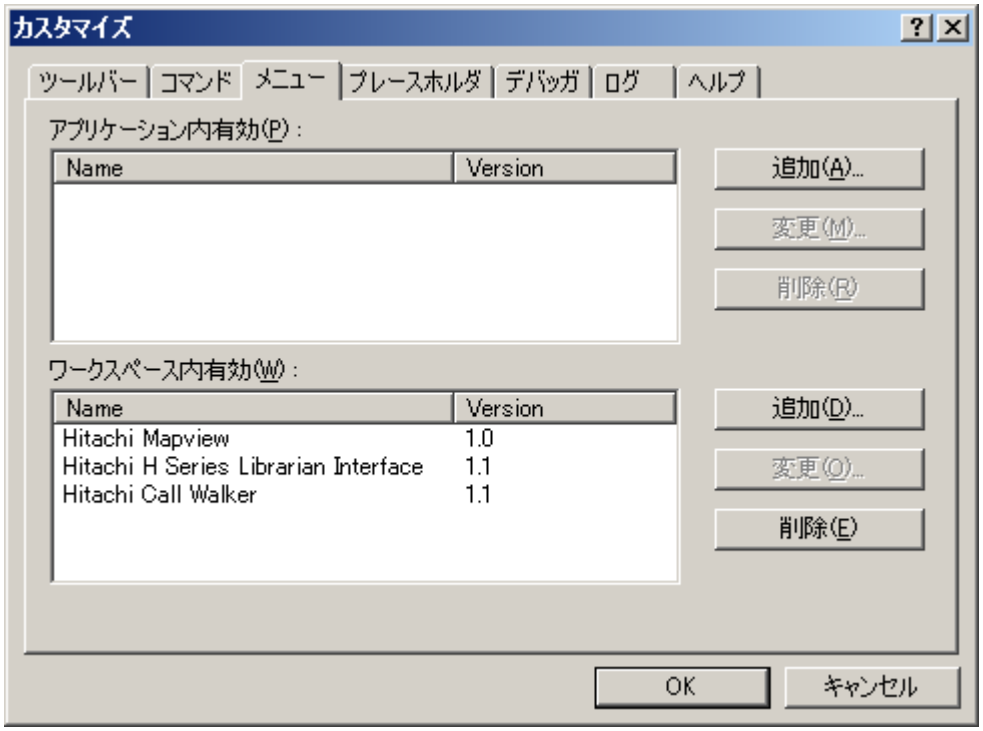


図 6.4: **カスタマイズ ダイアログボックス メニュー タブ**

2. [追加…] ボタンをクリックしてください。図6.5に示すダイアログボックスが表示されます。既存のシステムツールをメニューに追加するには、 [既存ツールからの選択] ラジオボタンを選び、ドロップダウンリストからシステムツールを選び、 [OK]ボタンをクリックしてください。
3. [名前] フィールドにツール名を入力してください。
4. [コマンド] フィールドにコマンドを入力してください。ただし、コマンドに渡す引数は入力しないでください。
5. [引数] フィールドにコマンドに渡す引数を入力してください。
6. [初期ディレクトリ] フィールドにツールを実行する初期ディレクトリを入力してください。
7. [OK] ボタンをクリックするとメニューオプションが [ツール] メニューに追加されます。

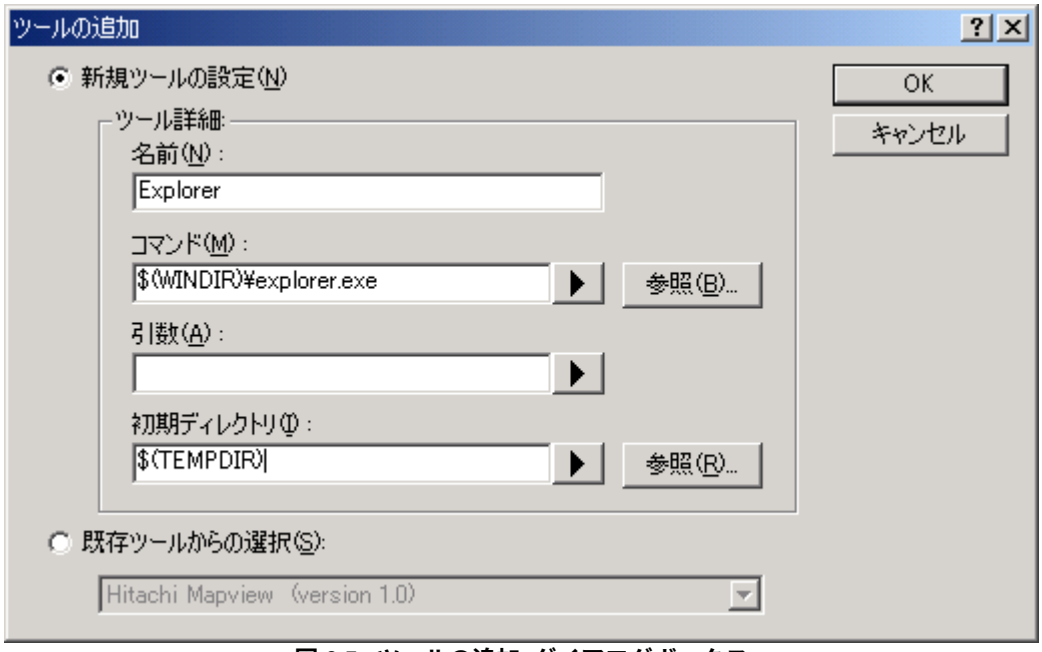


図 6.5: **ツールの追加 ダイアログボックス**

新しいメニューオプションはリストの最後に追加されます（ツールメニューの一番下）。

➲ メニューオプションを変更するには

1. [ツール->カスタマイズ...] を選んでください。図6.1に示すダイアログボックスが表示されます。[メニュー]タブ（図6.4参照）を選んでください。
2. 変更するメニューオプションを選んで [変更...] ボタンをクリックしてください。
3. [ツールの変更] ダイアログボックス（図6.6）を変更後、 [OK] ボタンをクリックしてください。

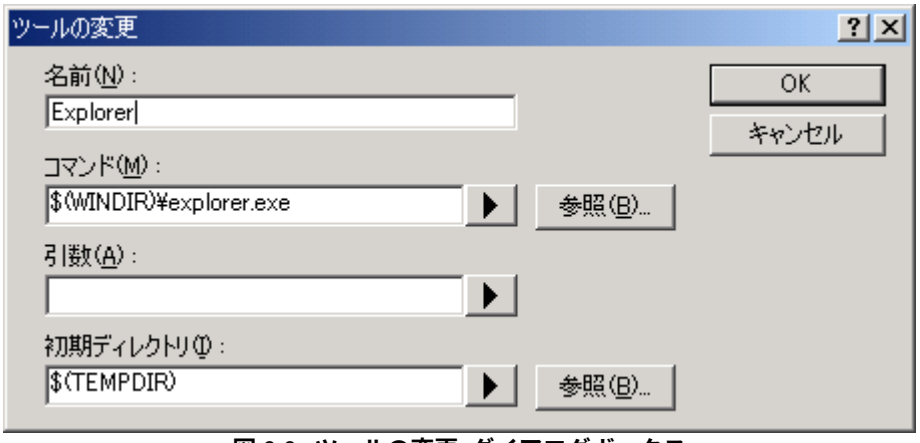


図 6.6: **ツールの変更 ダイアログボックス**

➲ メニューオプションを削除するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。[メニュー]タブ（図6.4参照）を選んでください。
2. 削除するメニューオプションを選んで [削除] ボタンをクリックしてください。

6.3 ヘルプシステムを構築する

HEW ではエディタウィンドウでコンテキスト依存ヘルプを提供します。エディタウィンドウでテキストを選び F1 キーを押下すると、選んだテキストに関するヘルプを検索します。検索するヘルプファイルは[カスタマイズ]ダイアログボックスの[ヘルプ]タブに表示されます。

➲ 新しいヘルプファイルを追加するには

1. [ツール->カスタマイズ…] を選んでください。図6.1に示すダイアログボックスが表示されます。 [ヘルプ] タブ （図6.7参照）を選んでください。

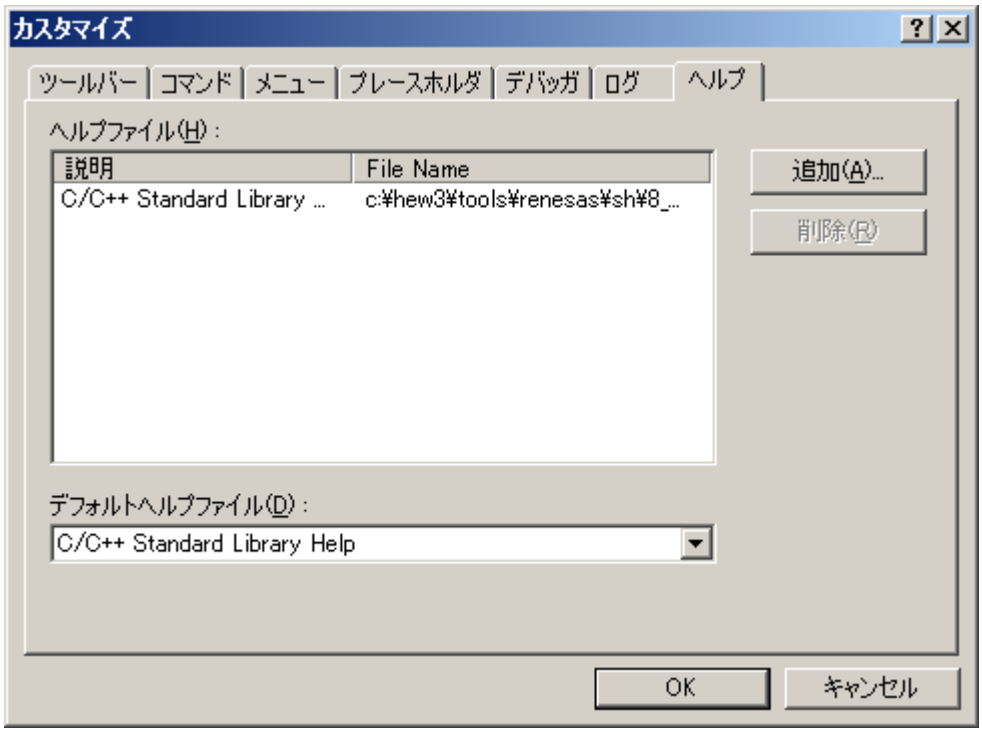


図 6.7: **カスタマイズ ダイアログボックス ヘルプ タブ**

2. [追加…] ボタンをクリックしてください。 [ヘルプファイルの追加]ダイアログボックス（図6.8）が表示されます。
3. [説明] フィールドにヘルプファイルの説明を入力してください。
4. [ファイル名]フィールドにヘルプファイルへのフルパスを入力してください（または[参照…]ボタンをクリックしてファイルを選んでください）。
5. [OK]ボタン をクリックすると新しいヘルプファイルが定義されます。

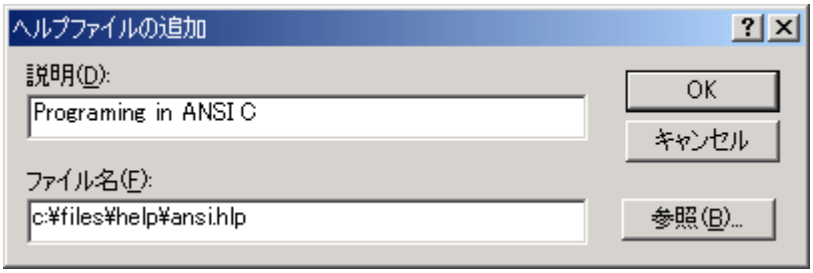


図 6.8: **ヘルプファイルの追加 ダイアログボックス**

デフォルトのヘルプファイルを設定する場合は[デフォルトヘルプファイル] ドロップダウンリストからヘルプファイルを選んでください。F1 キーを押下したときに任意のヘルプファイルを参照するには [(None)] を選んでください。

6.4　ワークスペースオプションを指定する

HEW では[オプション] ダイアログボックス（図 6.9）でワークスペースの様々なオプションを設定することができます。 [ツール->オプション...]の[ワークスペース]タブを選んでください。

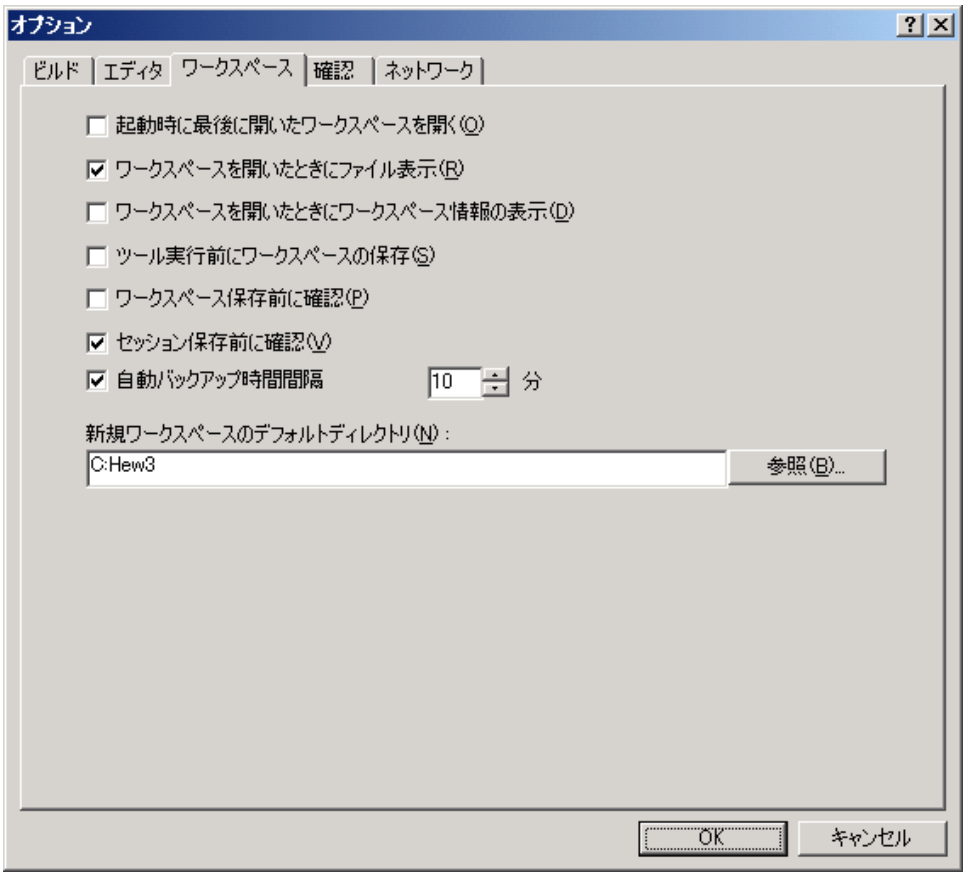


図 6.9: **オプション ダイアログボックス ワークスペース タブ**

次にこのタブで制御できるオプションを説明します。

6.4.1 起動時に最後に開いたワークスペースを開くチェックボックス

このチェックボックスをチェックすると、HEW 起動時に、最後に開いたワークスペースを自動的に開きます。

6.4.2 ワークスペースを開いたときにファイル表示チェックボックス

HEW は、ワークスペースを閉じるとき、開いていたファイルを記憶します。そして、再びワークスペースを開くとき、HEW は同じファイルを復帰させる（つまり開く）ことができます。これにより、中断したセッションを引き続き行うことができます。このチェックボックスをチェックすると、ワークスペースを開いたときに前回開いていたファイルを開きます。

6.4.3 ワークスペースを開いたときにワークスペース情報の表示チェックボックス

多くのワークスペースを使うと、各ワークスペースの内容を正確に覚えておくのは難しくなります。この問題を解決するために、HEW では、各ワークスペースの説明を入力しておくことができ、このチェックボックスをチェックすると、ワークスペースを開くときに表示することができます。

➲ ワークスペースの説明を入力するには

1. [ワークスペース] ウィンドウの [Projects] タブからワークスペースのアイコンを選んでください。
2. マウスの右ボタンをクリックしてポップアップメニューを表示させ、 [プロパティ] オプションを選んでください。図6.10に示すダイアログボックスが表示されます。
3. [情報] フィールドに説明を入力してください。
4. ワークスペースを開いたときにワークスペースプロパティダイアログボックスを開かせたいときは [ワークスペースを開いたときにワークスペース情報の表示] チェックボックスをチェックしてください。このチェックボックスは[オプション]ダイアログボックスの[ワークスペース] タブの[ワークスペースを開いたときにワークスペース情報の表示]チェックボックスと同じ役割を持っています。

図 6.10: ワークスペースプロパティダイアログボックス

HEW では、ワークスペースを開くときにこの説明を表示することができます。したがって、そのワークスペースが目的のワークスペースかどうかを判断することができます。この説明を表示するには、[ワークスペースを開いたときにワークスペース情報の表示]チェックボックスをチェックしてください。

6.4.4 ツール実行前にワークスペースの保存チェックボックス

このチェックボックスをチェックすると、ビルドフェーズを実行（[ビルド]、[すべてビルド]、[コンパイル]操作）する前や、バージョン管理コマンドを実行する前に、現在のワークスペースを保存します。

6.4.5 ワークスペース保存前に確認チェックボックス

上記の[ツール実行前にワークスペースの保存]チェックボックスに加えてこのチェックボックスをチェックすると、保存する前に確認の画面が表示されます。

6.4.6 セッション保存前に確認チェックボックス

このオプションをチェックすると、HEW はセッションをディスクに保存する前にプロンプトを表示します。

6.4.7 自動バックアップ時間間隔チェックボックス

このオプションをチェックすると、指定した時間間隔（デフォルト 10 分）で、ワークスペース、プロジェクト、およびセッションファイルをバックアップします。加えた変更はすべて、テンポラリファイルに保存されます。

ワークスペースを開いて以下のダイアログボックスが表示される場合、前回そのワークスペースを使用したときに問題が発生したことを意味します。ファイルを回復するには、回復するファイルの名前の横にあるチェックボックスをチェックし、[OK]ボタンをクリックしてください。[キャンセル]ボタンをクリックすると、自動バックアップファイルを削除し、元のファイルからロードを行います。

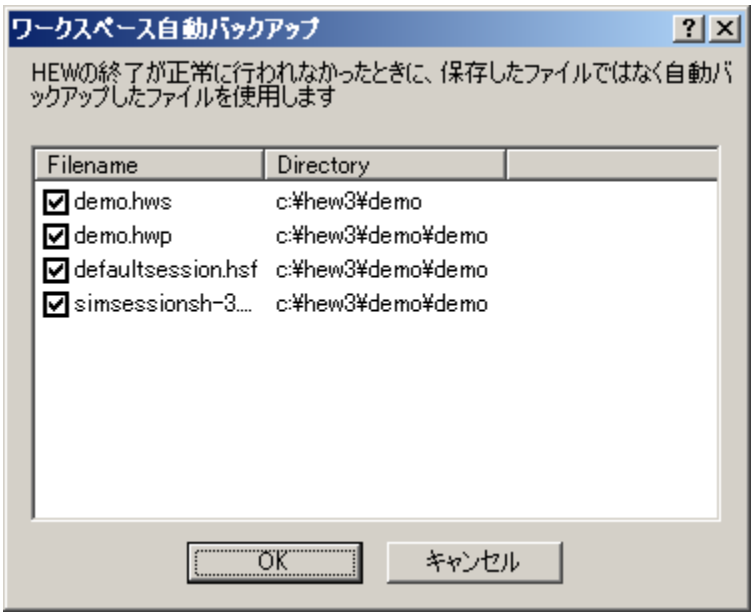


図 6.11: **ワークスペース自動バックアップダイアログボックス**

6.4.8 新規ワークスペースのデフォルトディレクトリエディットボックス

新しいワークスペースを作成すると [新規プロジェクトワークスペース] ダイアログボックスが起動します。このダイアログボックスにはその新しいワークスペースが作成されるディレクトリを入力するフィールドがあります。デフォルトでは、HEW インストールディレクトリが入力してあります。しかし、他のディレクトリ（例：“C:¥Workspaces”）にデフォルトを変更したい場合、このフィールドにそのディレクトリを入力するか、 [参照…] ボタンで位置を指定してください。

6.5 HEW エディタ以外のエディタを使う

HEW エディタ以外のエディタも使うことができます。外部のエディタを指定してあると、以下の操作をしたときにそれが起動します。

- ワークスペースウィンドウの[Projects]タブのファイルをダブルクリックしたとき
- ワークスペースウィンドウの[Navigation]タブのエントリをダブルクリックしたとき
- アウトプットウィンドウの[Build] タブのエラーやウォーニングをダブルクリックしたとき
- アウトプットウィンドウの[Find in Files] タブのエントリをダブルクリックしたとき
- ワークスペースウィンドウのポップアップメニューから[開く <ファイル名>] を選んだとき

➲ HEW エディタ以外のエディタを指定するには

1. [ツール->オプション…] を選んでください。 [オプション] ダイアログボックスが表示されるので [エディタ] タブ（図6.12）を選んでください。

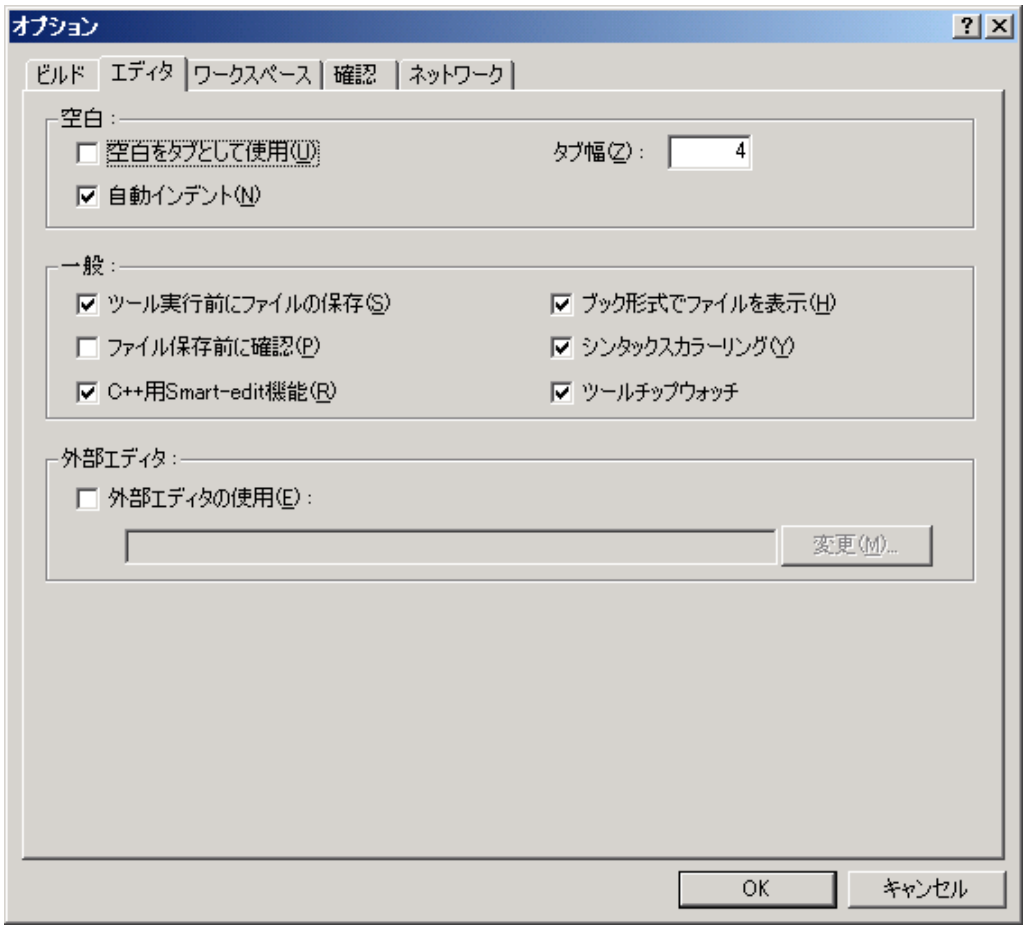


図 6.12: **オプション ダイアログボックス エディタ タブ**

2. [外部エディタの使用] チェックボックスをチェックしてください。 [外部エディタ] ダイアログボックスが表示されます （図6.13）。

図 6.13: **外部エディタ ダイアログボックス**

3. [コマンド] フィールドに実行可能ファイルのパス（引数を除く）を入力してください。
4. [ファイルを開く時の引数] フィールドにファイルを開くのに必要な引数を入力してください。開くファイルのパスには $(FULLFILE) プレースホルダを使ってください。
5. [行番号を指定してファイルを開く時の引数] フィールドにファイルの特定の行を開くのに必要な引数を入力してください。開くファイルのパスには $(FULLFILE) プレースホルダを使ってください。また、カーソルを最初に置く行の番号には $(LINE) プレースホルダを使ってください。
6. [OK] ボタンをクリックするとエディタが指定されます。

注意 HEW **エディタ以外のエディタを使う場合、以下のことに注意してください。**

- どのように起動しても、外部エディタを起動するたびに、エディタは新規に起動します。
- ビルド操作（[ビルド]、[すべてビルド]、[コンパイル]）を行う前にファイルを保存してください。
- デバッグ時には、外部エディタは使用できません。

6.6 ファイルの保存をカスタマイズする

[オプション] ダイアログボックス （図 6.12）の[エディタ] タブで、ファイルの保存方法をカスタマイズすることができます。[ツール->オプション…] で[エディタ] タブを選んでください。

ファイル保存に関するチェックボックスを以下に説明します。

6.6.1 ツール実行前にワークスペースの保存チェックボックス

このチェックボックスをチェックすると、ビルドフェーズ（[ビルド]、[すべてビルド]、[コンパイル]操作）またはバージョン管理コマンドを実行する前に編集したファイルを保存します。

6.6.2 ワークスペース保存前に確認チェックボックス

上記の[ツール実行前にワークスペースの保存]チェックボックスと、このチェックボックスをチェックすると、保存する前に確認メッセージを表示します。

6.7　カスタムプレースホルダを使う

HEW では、ディレクトリを定義するときに、以前に定義したプレースホルダを指定することができます。これによって、プロジェクトを再配置することができます。

また、HEW では、カスタムプレースホルダを定義することもできます。このことは、ユーザがカスタムプレースホルダを定義し、そのディレクトリの値を決められることを意味します。一度定義されると、このプレースホルダは HEW の別の場所で有効になります。

[アプリケーション内有効カスタムプレースホルダ]に定義したプレースホルダは、HEW で使用するすべてのワークスペースやプロジェクトに対して有効です。それに対して、[ワークスペース内有効カスタムプレースホルダ]に定義したプレースホルダは、現在のワークスペースにおいてのみ有効です。

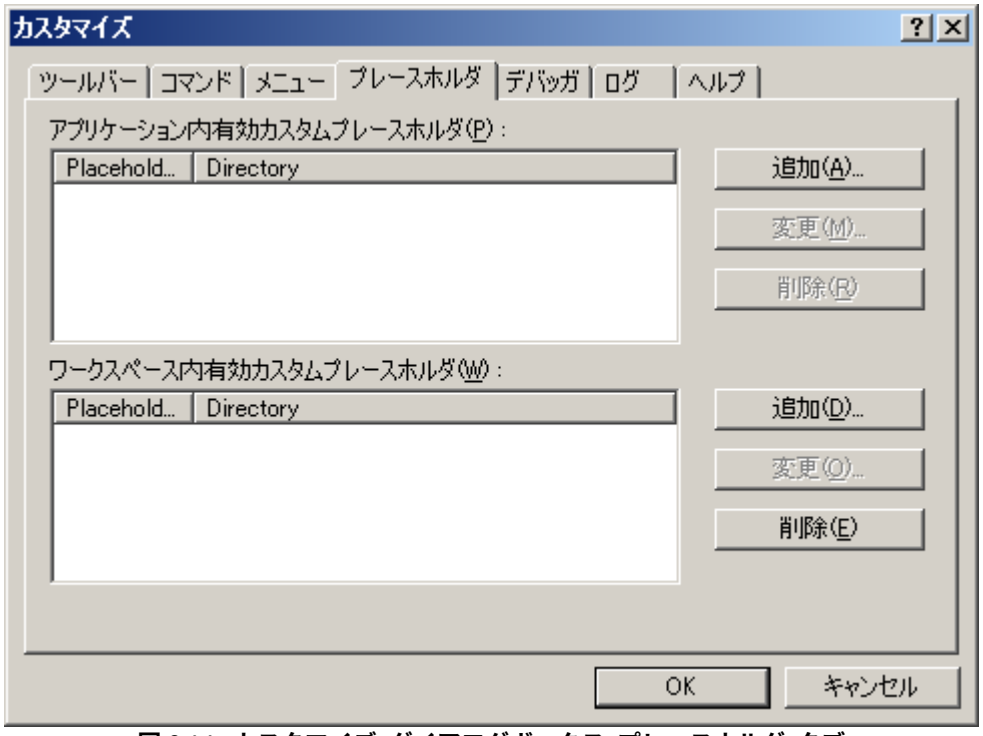


図 6.14: **カスタマイズ ダイアログボックス プレースホルダ タブ**

➲ カスタムプレースホルダを追加するには

1. [ツール->カスタマイズ...] を選んでください。図6.1に示すダイアログボックスが表示されます。
   [プレースホルダ]タブを選んでください（図6.14）。
2. [アプリケーション内有効カスタムプレースホルダ] 、あるいは [ワークスペース内有効カスタムプレースホルダ] のどちらを使いたいか、選んでください。リスト横にある[追加…]ボタンをクリックしてください。
3. [新規カスタムプレースホルダ] ダイアログボックスが表示されます（図6.15）。
4. このフィールドでは、プレースホルダの適切な名前、およびプレースホルダが意味する内容

の説明を選んでください。

5. 次に、このプレースホルダと関連するディレクトリを選んでください。$(PROJDIR)のように、オプション設定等で定義済みのプレースホルダとして使用することができます。

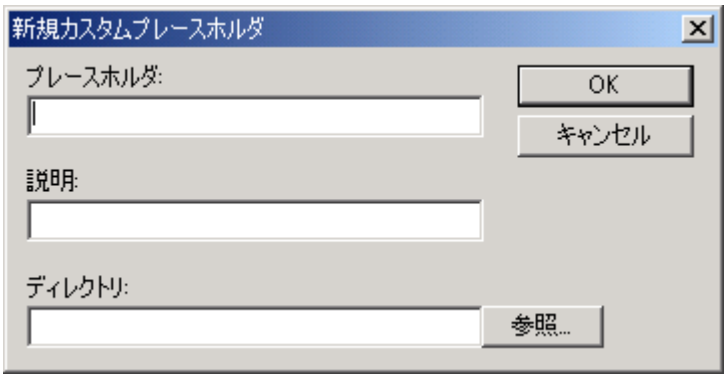


図 6.15: 新規カスタムプレースホルダダイアログボックス

6.8 ワークスペースやプロジェクトのログ機能を使う

HEW3.0 では、ワークスペースとプロジェクトのログ機能がアプリケーションの中に統合されています。この機能は、[カスタマイズ]ダイアログボックスの[ログ]タブで設定することができます。この設定により、ユーザ名や変更内容についてログファイルを作成できるため、ネットワークデータベースが動作しているときなどに特に便利です。このダイアログボックスを図 6.16 に示します。

[ワークスペースログファイル生成]をクリックすると、ワークスペースのすべての変更箇所がワークスペースと同じ名前で拡張子が.log のファイルにログされます。このファイルの位置は、ワークスペースファイルと同じディレクトリです。

[プロジェクトログの生成]をクリックすると、現在のワークスペースでのすべてのプロジェクトの変更箇所がプロジェクトと同じ名前で拡張子が.log のファイルにログされます。このファイルの位置は、ワークスペースファイルと同じディレクトリです。

ワークスペースを保存するとログファイルは更新されます。

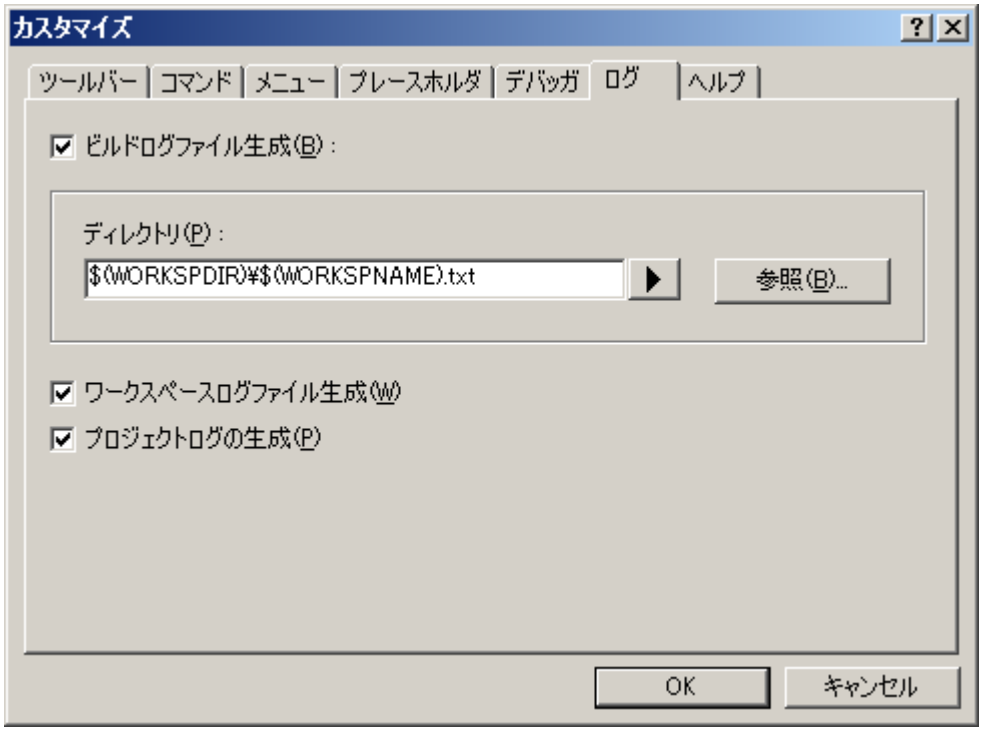


図 6.16: **[カスタマイズ]ダイアログボックスの[ログ]タブ**

6.9 バーチャルデスクトップを使用する

HEW ではバーチャルデスクトップの概念を導入しています。これにより、ボタンをクリックすることでウィンドウのコンフィグレーションの切り替えができるようになっています。ウィンドウのコンフィグレーションによって、ツールバーとウィンドウを表示または非表示にすることができます。

最大 4 つまでのウィンドウのコンフィグレーションを使用することが可能です。セッションを保存すると、各コンフィグレーションのウィンドウ位置がセッションファイルに保存されます。ウィンドウのコンフィグレーションを切り替えるだけで、他のウィンドウへアクセスすることができます。ツールバーとウィンドウは、コンフィグレーションに依存します。ソースファイルはバーチャルデスクトップシステムから独立しており、引き続き表示されます。

➲ コンフィグレーションに新しい名前を付けるには

1. [ウィンドウ->バーチャルデスクトップ] サブメニューから [デスクトップマネージャ] を選んでください。
2. [デスクトップマネージャ] ダイアログボックスで、名前を変更する [ウィンドウコンフィグレーション] を選んでください。
3. [名前の変更…] ボタンをクリックしてください。新しい名前を入力し、[OK]ボタンをクリックしてください。
4. [OK] ボタンをクリックしてください。

➲ コンフィグレーションを切り替えるには

ウィンドウコンフィグレーションの切り替えには、いくつかの方法があります。

もっとも簡単な方法は以下の通りです。

1. ステータスバー上のバーチャルデスクトップボタンをクリックします。

図 6.17 の例では、クリックしたバーチャルデスクトップボタンの番号は"1"で、コンフィグレーションに"Build"という名前が付けられています。ボタンの右側に " Build desktop " と表示しています。

別のバーチャルデスクトップボタンをクリックするとそのボタンが選択状態になり、ボタンの右側の表示も切り替わります。一度クリックすると、HEW は新しいコンフィグレーション形式でウィンドウをロードします。

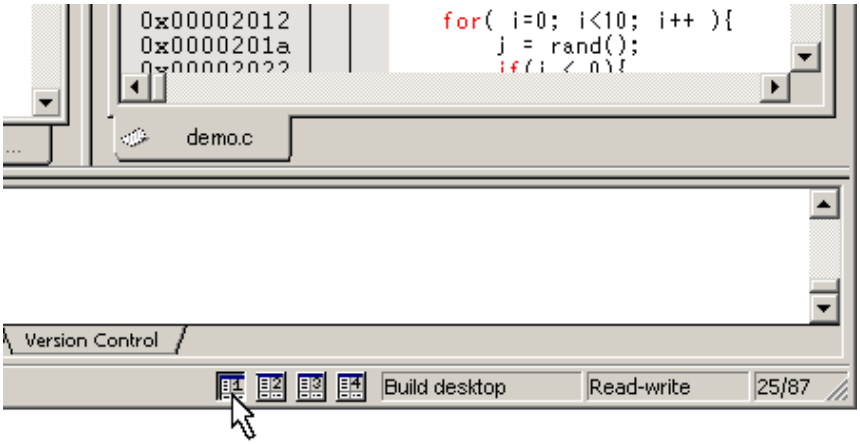


図 6.17: バーチャルデスクトップボタン

コンフィグレーションを切り替えるもう一つの方法は以下の通りです。

1. [ウィンドウ->バーチャルデスクトップ] サブメニューから表示したいコンフィグレーションを選んでください。

注意 コンフィグレーションが一度も使用されていない場合、HEW は現在のウィンドウコンフィグレーションを新しいコンフィグレーションにコピーします。ユーザはこの設定を変更することができます。セッションを保存すると、各ウィンドウコンフィグレーションでのウィンドウの位置を保存します。

7. バージョン管理

HEW はバージョン管理システムと接続することができます。プロジェクトでバージョン管理システムを使用する理由を以下に示します。

- プロジェクト開発環境の統合性を維持するため
- プロジェクトのバージョンを記録･保存するため
- ソースファイルに対するバージョン管理を行い、複数のユーザが一つのプロジェクトを共同開発できるようにするため

図 7.1 にバージョン管理システムを使用するプロジェクトの一般例を示します。ここでは 3 人のユーザがソースコードを相互参照するために同じ共有ネットワークドライブを使用しています。バージョン管理システムはソースファイルの参照や更新を管理するために使用します。

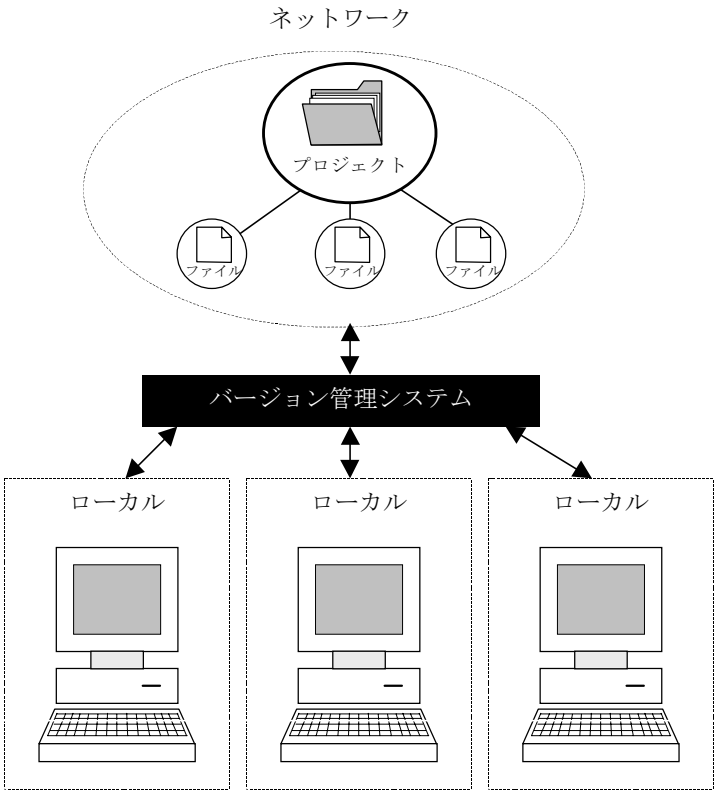


図 7.1: **バージョン管理**

7.1 バージョン管理システムを選択する

初期設定では、バージョン管理サブメニューが表示されます（図 7.2）。このとき、まだバージョン管理システムが現在のワークスペースで有効でないため、 [ツール->バージョン管理->選択...] オプションだけが利用できます。

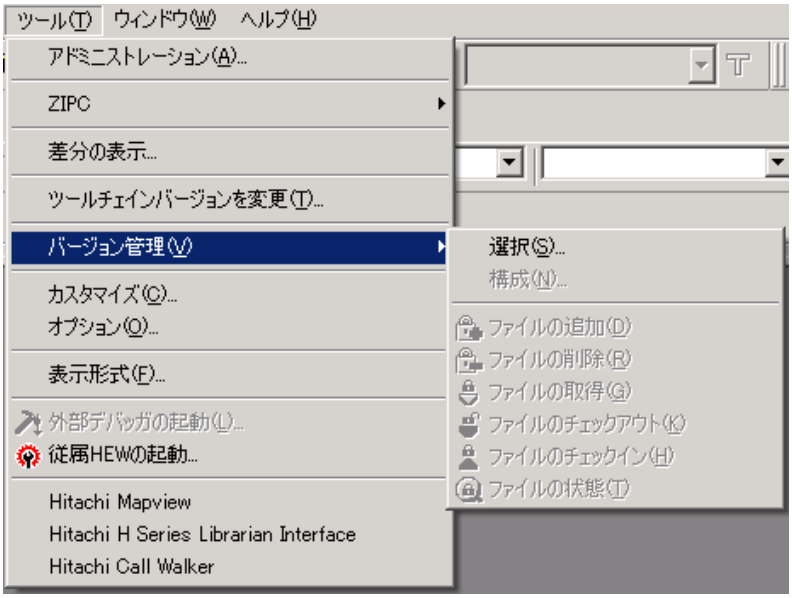


図 7.2: バージョン管理サブメニュー

➲ バージョン管理システムを選ぶには

1. [ツール->バージョン管理->選択...] を選んでください。図7.3に示すダイアログボックスが表示されます。このダイアログボックスにはサポートするバージョン管理システムがすべて表示されます。
2. [バージョン管理ツール] リストからバージョン管理システムを選んで [選択] ボタンをクリックしてください。 [現在のバージョン管理ツール] には新しい設定が表示されます。
3. [OK] ボタンをクリックしてください。

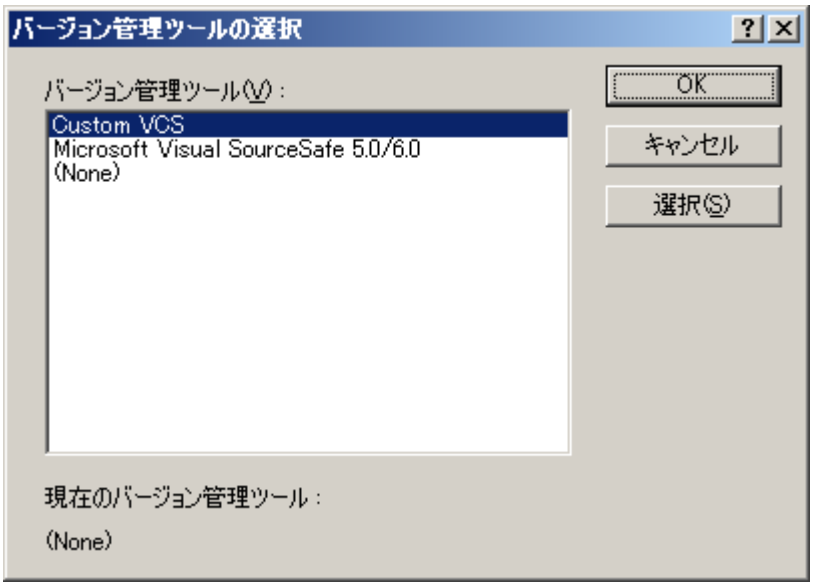


図 7.3: **バージョン管理ツールの選択 ダイアログボックス**

注意 HEW **と共にインストールされたバージョン管理システムだけが[バージョン管理ツールの選択] ダイアログボックスに表示されます（図** 7.3）。

バージョン管理システムを選択すると、[ツール->バージョン管理->構成...] オプションが使用できるようになります。

次の章ではバージョン管理システムの使用方法について説明します。

8. カスタムバージョン管理システム

HEW にカスタムのバージョン管理システムを接続できます。HEW は、すでにマシンにインストールされたバージョン管理システムと接続できます。つまり、HEW がバージョン管理ツール自体を提供するわけではなく、ワークスペースやプロジェクトで使用するバージョン管理システムを統合する手段を提供するだけです。

8.1 バージョン管理メニューオプションを定義する

カスタムのバージョン管理システムでは、[ツール->バージョン管理]サブメニューからオプションを選ぶか、バージョン管理ツールバーボタンでバージョン管理コマンドを起動することができます。そうすると、関連するコマンドを実行し、その出力が[アウトプット]ウィンドウの[Version Control] タブに表示されます。

➲ バージョン管理メニューオプションまたはツールバーボタンを実行するには

1. [ワークスペース] ウィンドウからバージョン管理コマンドを適用する項目（ワークスペース、プロジェクト、フォルダ、ファイルなど）を選んでください。コマンドが選択されると、すべてのファイルが選んだ項目から抽出され、バージョン管理コマンドに渡します。例えば、ワークスペースアイコンを選ぶと、そのプロジェクトのすべてのファイルがバージョン管理コマンドに渡します。これには、システムファイルも含まれます。
2. [ツール->バージョン管理] サブメニューからメニューオプションを選ぶか、バージョン管理ツールバーボタンを選んでください。

カスタムバージョン管理のサポートにより、バージョン管理システムの指定が任意に構築できます。[ツール->バージョン管理->構成…] を選んで [Version Control Setup] ダイアログボックスを表示してください （図 8.1）。

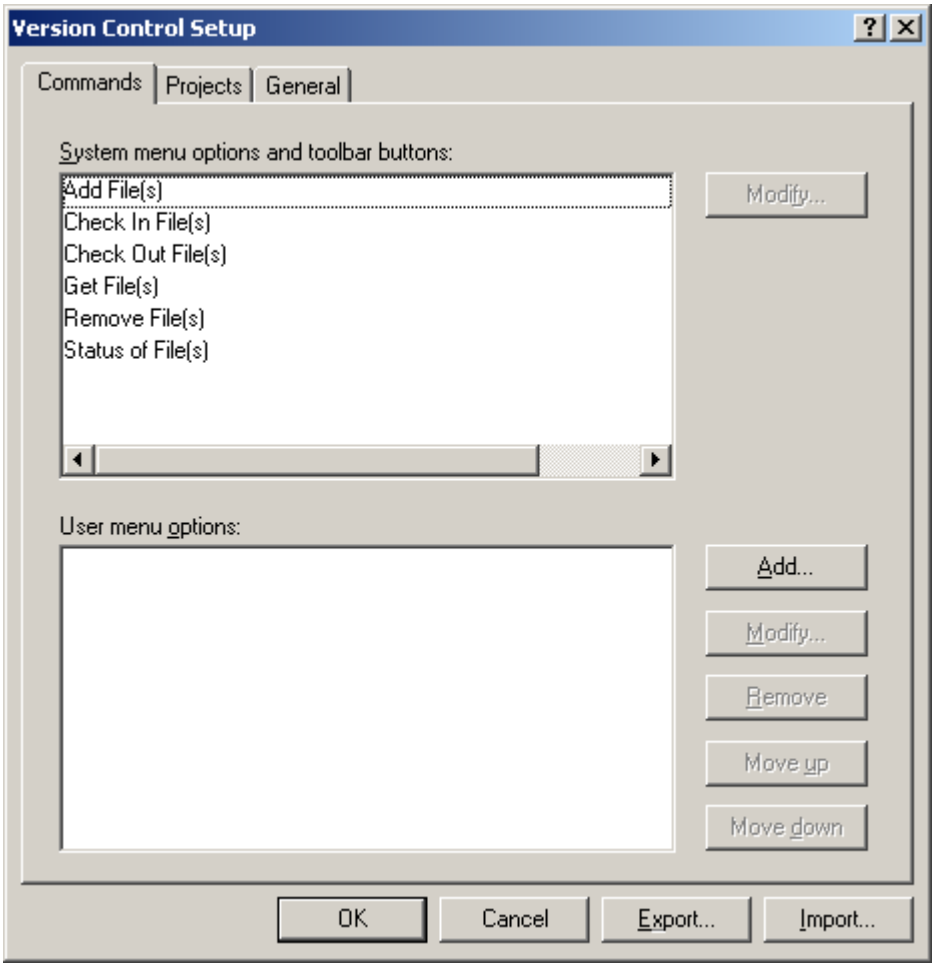


図 8.1: Version Control Setup **ダイアログボックス** Commands **タブ**

[Commands]タブには 2 つのメニューオプションのリストがあります。1 つめのリスト[System menu options and toolbar buttons]は常にバージョン管理サブメニューに表示されるメニューオプションです。これに対応するバージョン管理ツールバーがあります。2 つめのリスト [User menu options]はユーザ定義の追加メニューオプションで、バージョン管理サブメニューの最後に追加されます。図 8.2 にバージョン管理サブメニューの構成を示します。

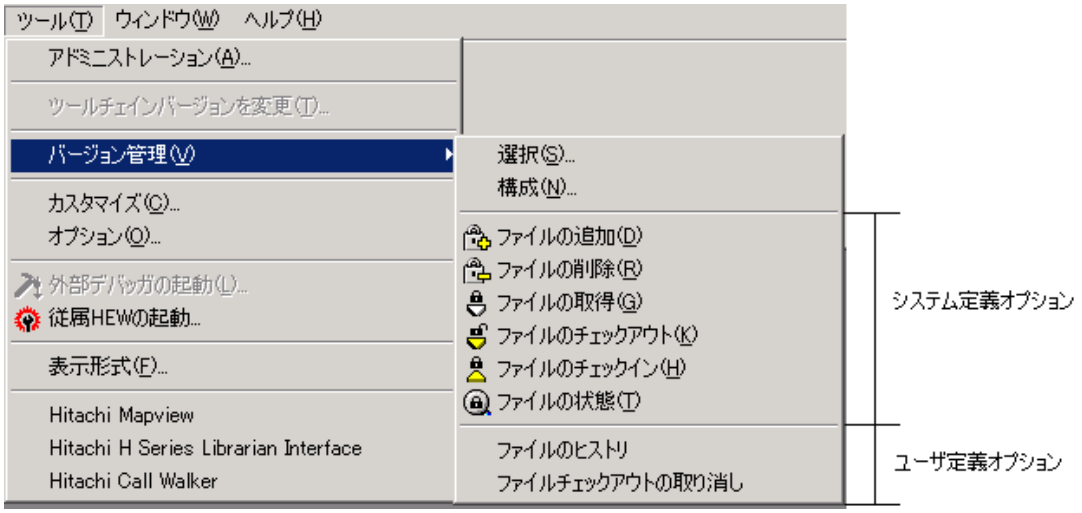


図 8.2: **バージョン管理 サブメニュー**

8.1.1 システムメニューオプションとツールバーボタン

ツールバーまたは[ツール->バージョン管理]サブメニューのシステム定義オプションからコマンドを起動するには、まず、起動したときに実行する関連したコマンドを定義しなければなりません。オプション名とその説明を表 8.1 に示します。

表 8.1: システムメニューオプションの説明

オプション	説明
Add File(s)	バージョン管理に選択したファイルを追加する
Remove File(s)	バージョン管理から選択したファイルを削除する
Get File(s)	バージョン管理から選択したファイルの読み取り専用ローカルファイルを取得する
Check In File(s)	バージョン管理に選択したファイルのローカルコピーを戻し更新する
Check Out File(s)	バージョン管理から選択したファイルの書き込み可能なローカルファイルを取得する
Status of File(s)	選択したファイルの状態を表示する

➲ システムメニューやツールバーボタンを変更するには

1. [バージョン管理->構成…] を選ぶと図8.1 に示すダイアログボックスが表示されます。
2. [System menu options and toolbar buttons] リストから変更するオプションを選んで [Modify…] ボタンをクリックすると 図8.3 に示すダイアログボックスが表示されます。図8.3は[Add File(s)]を選択した場合の例です。
3. [Add…] ボタンでコマンドを追加してください。詳細はこの章の後半の「バージョン管理コマンドを定義する」を参照してください。
4. [OK] ボタンをクリックして [Define Command for “<コマンド名>”] ダイアログボックスを閉じてください。
5. [OK] ボタンをクリックして [Version Control Setup] ダイアログボックスを閉じてください。

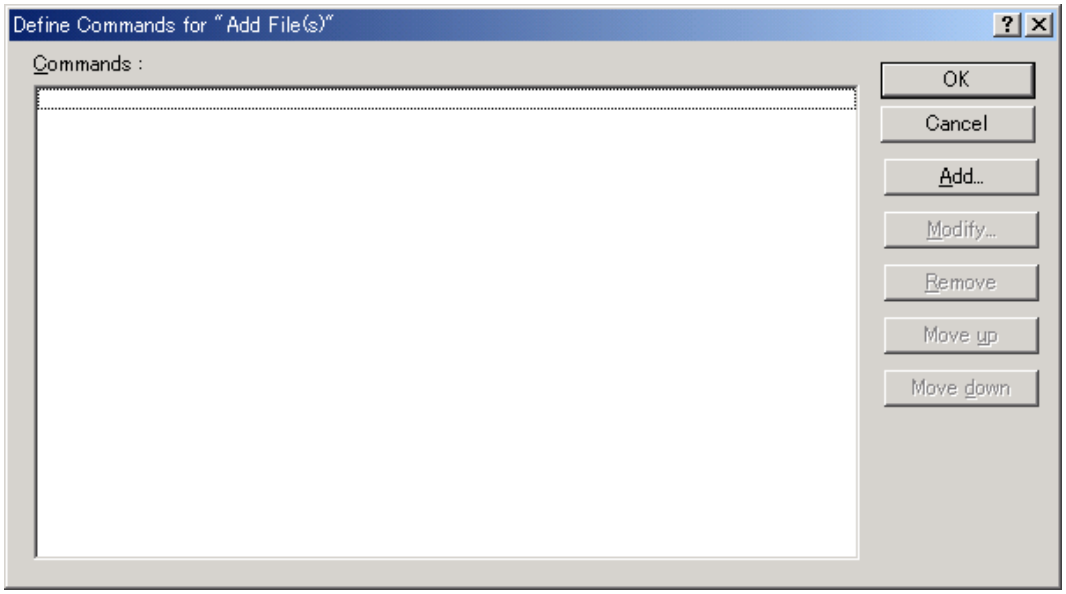


図 8.3: システムメニューオプションの変更（例）

8.1.2 ユーザ定義メニューオプション

ユーザ定義のメニューオプションはいくつでも作成できます。名前も自由につけられます。また、メニューに表示する順序も指定できます。しかし、ユーザ定義のメニューオプションはバージョン管理ツールバーには表示されません。

➲ 新しいバージョン管理メニューオプションを作成するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. [Add...] ボタンをクリックしてください。図8.4に示すダイアログボックスが表示されます。
3. [Option] フィールドにメニューオプション名を入力してください。
4. [Add…] ボタンでメニューオプションにコマンドを追加してください。詳細は、この章の後半の 「バージョン管理コマンドを定義する」を参照してください。
5. [OK] ボタンをクリックして [Add Menu Option] ダイアログボックスを閉じてください。
6. [OK] ボタンをクリックして [Version Control Setup] ダイアログボックスを閉じてください。

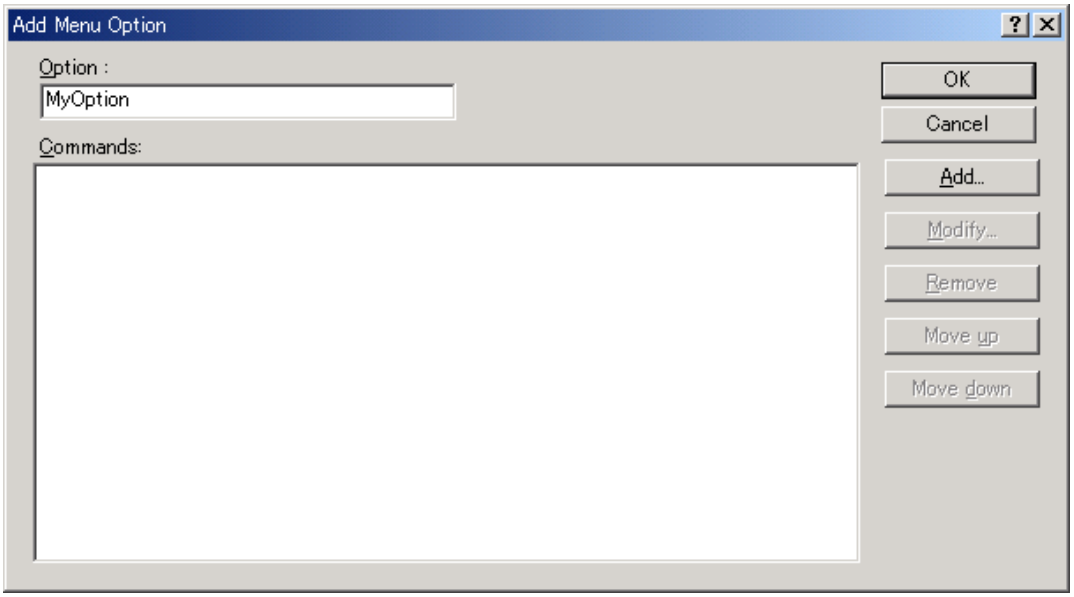


図 8.4: Add Menu Option **ダイアログボックス**

➲ 既存のバージョン管理メニューオプションを削除するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. [User menu options] リストから削除するメニューオプションを選んで [Remove] ボタンをクリックしてください。
3. [OK] ボタンをクリックして [Version Control Setup] ダイアログボックスを閉じてください。

➲ 既存のバージョン管理メニューオプションを変更するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. [User menu options] リストから変更するメニューオプションを選び、リストの横の

[Modify…] ボタンをクリックしてください。図8.4に示すダイアログボックスが表示されます（ただしタイトルは[Modify Menu Option]です）。

3. メニューオプション名を変更して [OK] ボタンをクリックしてください。
4. [OK] ボタンをクリックして [Version Control Setup] ダイアログボックスを閉じてください。

➲ バージョン管理メニューオプションの順序を変更するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. 移動するメニューオプションを選んで、上に移動するには [Move up] ボタンを、下に移動するには [Move down] ボタンをクリックしてください。
3. [OK] ボタンをクリックして [Version Control Setup] ダイアログボックスを閉じてください。

8.2 バージョン管理コマンドを定義する

図 8.3 や図 8.4 で [Add...] や [Modify…] ボタンをクリックすると、コマンドを定義できます。どちらの場合も、図 8.5 に示すダイアログボックスが表示されます。

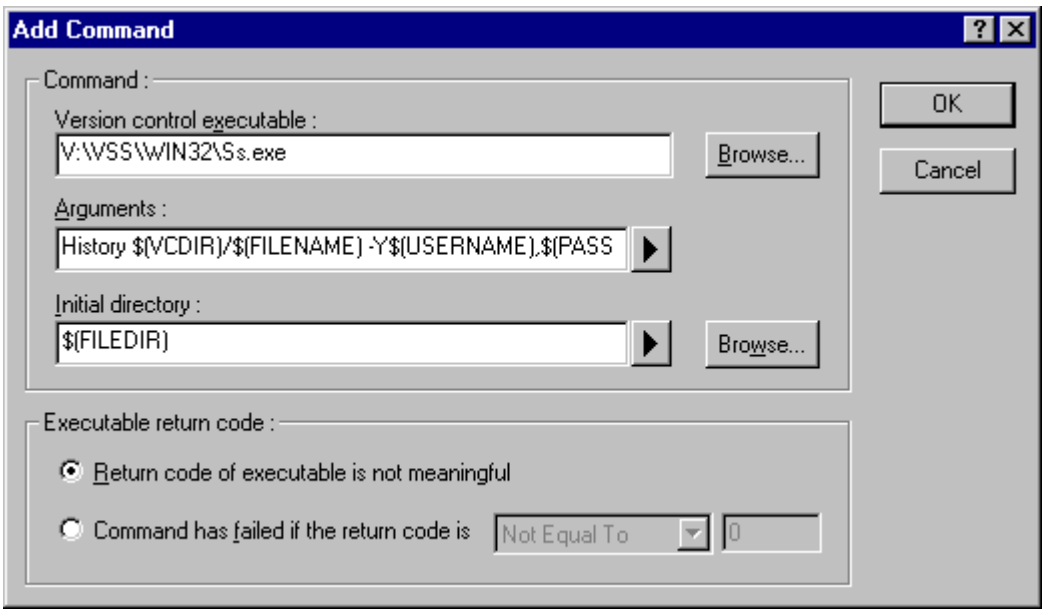


図 8.5: Add/Modify Command **ダイアログボックス**

➲ コマンドを定義するには

1. [Version control executable] フィールドにコマンド実行ファイルのフルパスを入力するか、[Browse…] ボタンをクリックしてコマンド実行ファイルの位置を指定してください。
2. [Arguments] フィールドにコマンドラインパラメータを入力してください。
3. [Initial directory] フィールドに実行ファイルを起動するディレクトリを入力するか、[Browse…] ボタンをクリックして起動ディレクトリを指定してください。ただし、特に起動ディレクトリを指定する必要がない場合は、 “$(FILEDIR)” プレースホルダを指定してファイルのディレクトリと同じディレクトリでコマンドを実行してください。
4. [Executable return code] オプションを設定してください。オプションの設定は次節を参照してください。

5. [OK] ボタンをクリックすると新しいコマンドが定義されます。

8.2.1 Executable return code オプション

コマンドのリターンコードでエラーを示す場合には、[Command has failed if the return code is] オプションを選択して右の 2 つのフィールドを設定してください。

[Command has failed if the return code is] オプションが選択されている場合、HEW は各コマンドのリターンコードをチェックしてエラーが起こったかどうか判定します。エラーが起こった場合、コマンド実行は停止してそれに続くコマンド実行やコマンドの後に続く処理（例：ビルド）は実行されません。

[Return code of tool is not meaningful] オプションが選択されている場合、HEW は各コマンドのリターンコードをチェックしません。したがって、すべてのコマンドが実行されます。

8.3 変数を指定する

変数は正しく指定しなければなりません。もし間違っていると、バージョン管理システムが正しく動作しません。また、1 つのバージョン管理コマンドを複数のファイルに適用する場合があるため、変数をフレキシブルな方法で指定することが重要です。これを行うために、[Arguments] フィールドにはプレースホルダボタンがあります（プレースホルダの詳細は、付録 C, 「プレースホルダ」を参照してください）。プレースホルダボタンをクリックすると、使用できるプレースホルダがポップアップメニューで表示されます（図 8.6）。表 8.2 に各プレースホルダの説明と実際の値を示します。

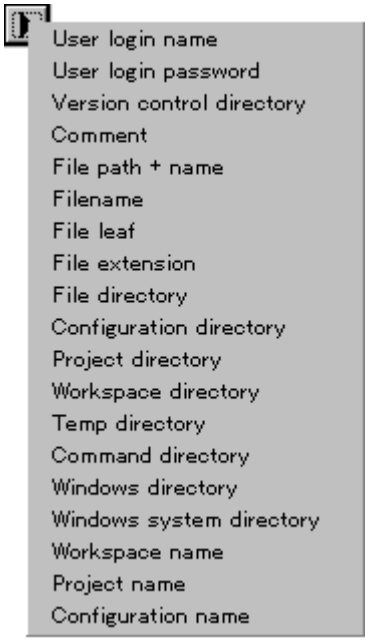


図 8.6: 変数フィールドプレースホルダのポップアップメニュー

表 8.2: 変数フィールドのプレースホルダ

プレースホルダ	実際の値
User login name	現在のユーザログイン (“General”タブ)
User login password	現在のユーザパスワード (“General”タブ)
Version control directory	「仮想的」バージョン管理マップ (“Projects”タブ)
Comment	コマンド実行前に指定したコメント
File path + name	操作するファイル名とそのフルパス
Filename	操作するファイル名（拡張子を含む）
File leaf	操作するファイル名（拡張子を含まない）
File extension	操作するファイルの拡張子
File directory	操作するファイルのディレクトリ
Configuration directory	現在のコンフィグレーションディレクトリ
Project directory	現在のプロジェクトのディレクトリ
Workspace directory	現在のワークスペースディレクトリ
Temp directory	テンポラリディレクトリ
Command directory	バージョン管理実行ディレクトリ
Windows directory	Windows® がインストールされているディレクトリ
Windows system directory	Windows®のシステムファイルがあるディレクトリ
Workspace name	現在のワークスペース名
Project name	現在のプロジェクト名
Configuration name	現在のコンフィグレーション名

8.3.1 ファイルの位置を指定する

ファイルの位置を指定するときには、プレースホルダを使用してください。そうでないと、そのコマンドは指定したファイルにしか適用できません。例えば、バージョン管理のアプリケーションに–GET コマンドが使われていて、ファイルの読み出し専用コピーを作成するとします。このとき、[Arguments] フィールドは以下のように指定できます。

```
-GET “c:¥vc¥files¥project¥main.c”
```

しかし、このコマンドを実行しても、ファイル MAIN.C しかコピーできません。この問題を解決するために、HEW にはプレースホルダとディレクトリのマッピングというシステムがあります。マッピングにより、どの「作業中の」ディレクトリ（作業中のソースファイルがあるディレクトリ）がどの「管理」ディレクトリ（バージョン管理システムに保存されているソースファイルのディレクトリ）に対応するかを指定します。これら 2 つのディレクトリ間のマッピングは、[Version Control Setup] ダイアログボックスの“Projects”タブ（図 8.7）で指定できます。

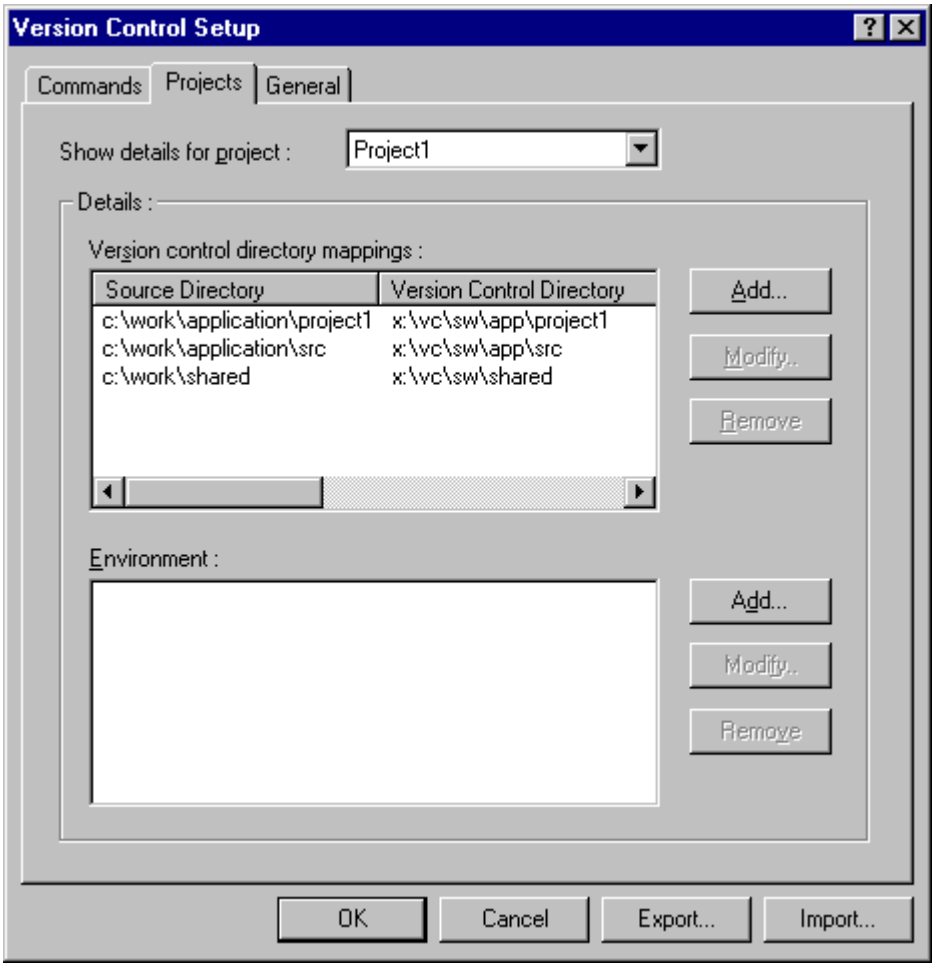


図 8.7: Version Control Setup **ダイアログボックス** Projects **タブ**

➲ 新しいマッピングを定義するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。 [Projects]タブを選んでください。図8.7に示すダイアログボックスが表示されます。
2. [Version control directory mappings] リストの横にある [Add…] ボタンをクリックしてください。図8.8に示すダイアログボックスが表示されます。
3. [Source directory] フィールドにもとの作業中のディレクトリを入力するか、 [Browse…] ボタンでディレクトリを選んでください。
4. [Version control directory] フィールドにバージョン管理ディレクトリを入力してください。

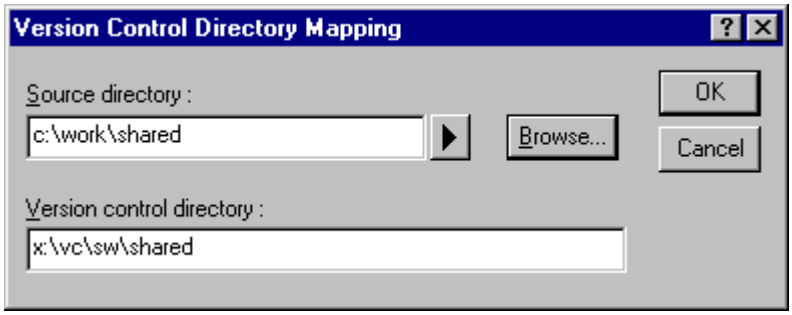


図 8.8: Version Control Directory Mapping **ダイアログボックス**

➲ 既存のマッピングを変更するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。 [Projects] タブを選んでください。図8.7に示すダイアログボックスが表示されます。
2. [Version control directory mappings] リストから変更するマッピングを選び、 [Modify…] ボタンをクリックしてください。 図8.8に示すダイアログボックスが表示されます。
3. 二つのディレクトリを変更して [OK] ボタンをクリックしてください。

➲ 既存のマッピングを削除するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。 [Projects] タブを選んでください。図8.7に示すダイアログボックスが表示されます。
2. [Version control directory mappings] リストから削除するマッピングを選んで [Remove] ボタンをクリックしてください。

マッピングを定義すると、[Version control directory] プレースホルダ $(VCDIR)を使用してプロジェクトファイルを保存するディレクトリを示すことができます。図8.9に例を示します。ここではネットワークを共有したバージョン管理ドライブ(X:¥) と開発が行われているローカルドライブ(C:¥)からマッピングされた3つのディレクトリがあります。

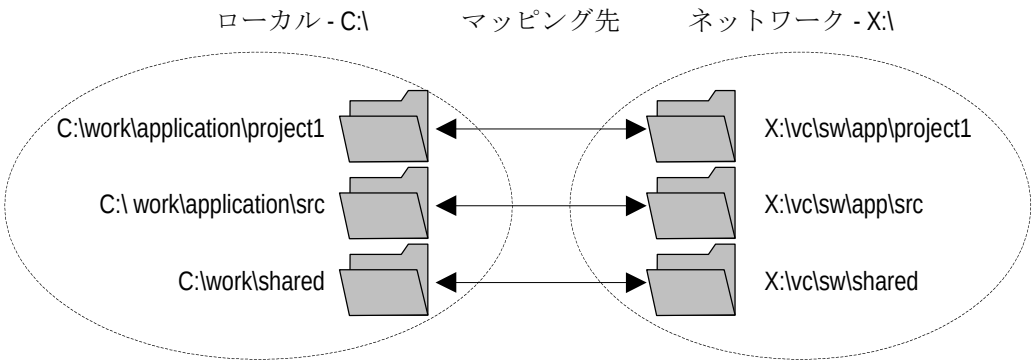


図 8.9:**マッピングの例**

例えば、–GET コマンドが使われていてファイルの読み出し専用コピーを作成するバージョンコントロールのアプリケーションが選ばれているとします。プロジェクトのすべてのファイルを取得するには、次のコマンドを使用します。

```
-GET “$(VCDIR)¥$(FILENAME)”
```

そのプロジェクトファイルのコマンドを実行すると、$(VCDIR) をファイルマッピングの中の対応するバージョン管理ディレクトリに置き換えます。

例えば、 FILE1.C が C:¥work¥application¥project1¥file1.c にあるとします。FILE1.C に-get コマンドが適用されると、次のようになります。

1. $(VCDIR)がx:¥vc¥sw¥app¥project1に置き換えられます。これはC:¥work¥application¥project1のバージョン管理ディレクトリマッピングに対応しているからです。
2. $(FILENAME) がFILE1.Cに置き換えられます。

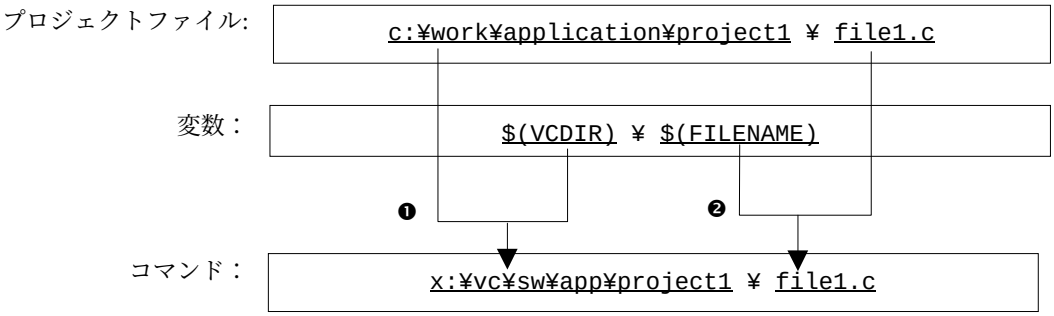


図 8.10: **置き換えの例**

8.3.2 環境変数の設定

現在の設定を確認するには[Version Control Setup]ダイアログボックスの[Projects] タブを表示してください（図 8.7）。

➲ 新しい環境変数を追加するには

1. [Environment] リストの横にある [Add…] ボタンをクリックしてください。図8.11に示すダイアログボックスが表示されます。 [Variable] フィールドに環境変数名を、[Value]フィールドに環境変数の値を入力して [OK]ボタン をクリックしてください。[Environment] リストに新しい環境変数が追加されます。

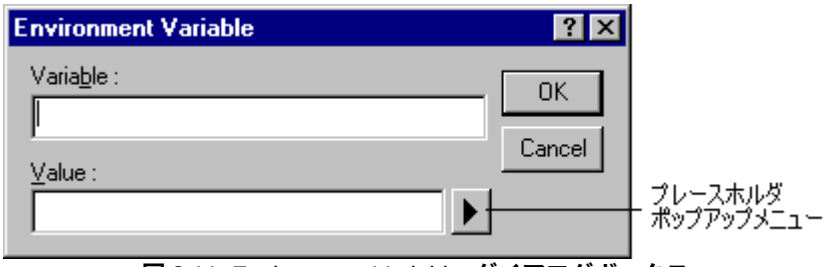


図 8.11: Environment Variable **ダイアログボックス**

➲ 環境変数を変更するには

1. 変更する環境変数を [Environment] リストから選んで、 [Modify…] ボタンをクリックしてください。[Variable] フィールドと [Value] フィールドを必要に応じて変更して[OK]ボタンをクリックすると、変更した環境変数がリストに追加されます。

➲ 環境変数を削除するには

1. 削除する環境変数を [Environment] リストから選んで、 [Remove] ボタンをクリックしてください。

8.3.3 コメントを指定する

コマンドに “$(COMMENT)” プレースホルダを含む場合、コマンド実行時に図 8.12 に示すダイアログボックスにコメントを入力してください。

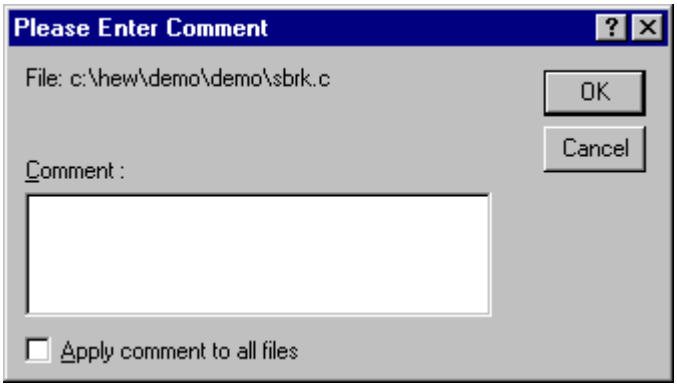


図 8.12: Please Enter Comment **ダイアログボックス**

コメントは各コマンドごとに指定できます。また、[Apply comment to all files] チェックボックスをチェックして [OK]ボタンをクリックすると、すべてのファイルに同じコメントが指定できます。

8.3.4 ユーザ名とパスワードを指定する

バージョン管理ツールでは一般的にユーザ名とパスワードの入力をコマンドラインで行う必要があります。これは、ファイルを保護し、どのファイルがどのユーザによって変更されたか記録するためです。バージョン管理システムでは 2 つのプレースホルダ “User login name”, $(USERNAME), と “User login password”, $(PASSWORD)をサポートします。コマンドを実行すると、これらのプレースホルダは[Version Control Setup]ダイアログボックスの[General] タブ（図 8.13）の現在の設定に置き換えます。

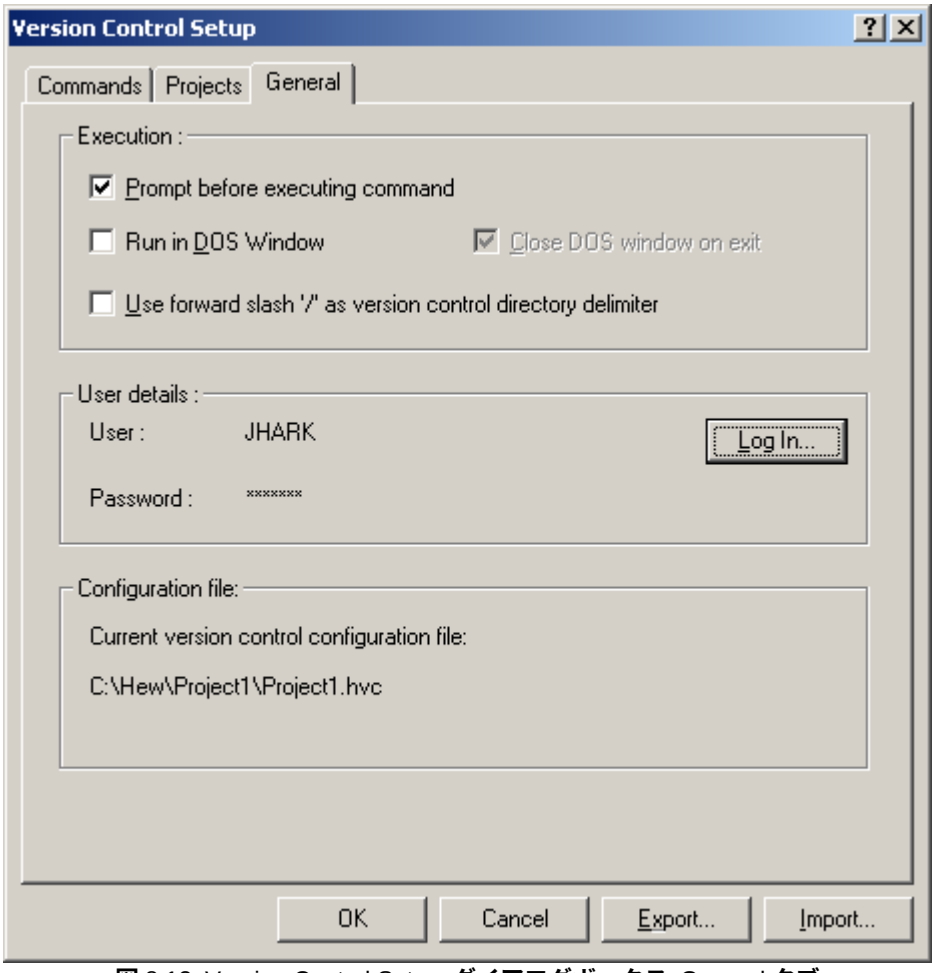


図 8.13: Version Control Setup **ダイアログボックス** General **タブ**

プレースホルダ$(USERNAME) と $(PASSWORD)に値を設定するには、最初にログインの指定をしておく必要があります。もしこれらのプレースホルダのうちのどちらかのプレースホルダを使うコマンドを実行する前にログイン指定をしていなかった場合、コマンド実行前にログインするよう要求されます。

➲ ユーザ名とパスワードを設定するには（ログインの指定）

1. [Log in…] ボタンをクリックしてください。図8.14に示すダイアログボックスが表示されます。
2. [User name] フィールドにユーザ名を入力してください。
3. [Password] フィールドにパスワードを入力してください。
4. [Confirm password by retyping it below] フィールドにパスワードを再入力してください。.
5. [OK] ボタンをクリックすると新しいユーザ名とパスワードが設定されます。[Password] フィールドと[Confirm password by retyping it below] フィールドで異なるパスワードが入力された場合はもう一度パスワードを入力するよう要求されます。

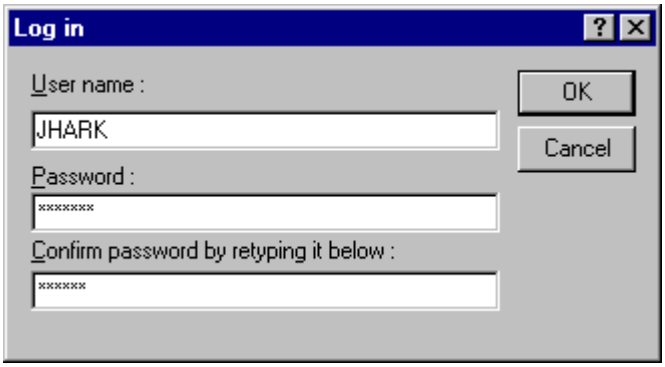


図 8.14: Log in **ダイアログボックス**

8.4 実行を制御する

[Version Control Setup]ダイアログボックスの [General] タブ（図 8.13）では、使用するバージョン管理ツールと HEW を制御することができます。また、現在のバージョン管理コンフィグレーションファイルへのフルパスを表示します。

8.4.1 Prompt before executing command チェックボックス

バージョン管理コマンドを実行する前にこのチェックボックスをチェックすると、図 8.15 に示すダイアログボックスが表示されます。このダイアログボックスではコマンドの実行の有無を確認することができます。チェックボックスのチェックを外すとそのコマンドは実行しません。[OK]ボタン をクリックすると、選んだコマンドを実行します。[Cancel]ボタン をクリックするとコマンドは実行しません。

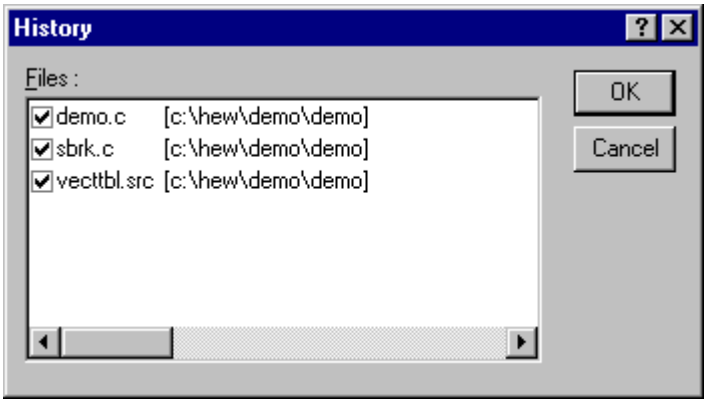


図 8.15: コマンドプロンプト ダイアログボックス

8.4.2 Run in DOS Window チェックボックス

デフォルトでは、バージョン管理コマンドの出力は[アウトプット]ウィンドウの[Version Control]タブに結果が表示されます。このチェックボックスをチェックすると、各コマンドを別の DOS ウィンドウで実行します。

8.4.3 Use forward slash ‘/’ as version control directory delimiter チェックボックス

デフォルトでは、HEW がプレースホルダ$(VCDIR)を置き換えるときバックスラッシュ文字 ‘¥’（日本語 Windows®では円記号）を使ってディレクトリを区切ります。しかし、お使いのバージョン管理システム(例 Visual SourceSafe)で、ディレクトリを区切るときにスラッシュ文字‘/’を使う場合は 、[Use forward slash ‘/’ as version control directory delimiter]チェックボックスをチェックしてスラッシュ文字‘/’でディレクトリを区切ってください。

8.5 設定内容の保存と適用

ワークスペースごとに異なるバージョン管理設定を行うことができます。HEW ではそれぞれのバージョン管理設定を保存して他のワークスペースに適用することができます。これにより、複数のワークスペースで何度も同じバージョン管理設定を行う必要がなく、時間を節約することができます。

➲ バージョン管理設定を保存するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. [Export…] ボタンをクリックしてください。標準のファイル保存ダイアログボックスが表示されます。設定内容を保存するディレクトリを選んでください。
3. ファイル名を入力して [OK] ボタンをクリックしてください。

➲ バージョン管理設定を適用するには

1. [バージョン管理->構成…] を選んでください。図8.1に示すダイアログボックスが表示されます。
2. [Import…] ボタンをクリックしてください。標準のファイルを開くダイアログボックスが表示されます。適用するファイル（*.HVC） を選んでください。
3. [OK] ボタンをクリックしてください。

9. Visual SourceSafe を使用する

HEW は Visual SourceSafe バージョン管理システムをサポートしています。現在、HEW は Visual SourceSafe のバージョン 5.0 と 6.0 をサポートしています。

Visual SourceSafe バージョン管理システムでは、Visual SourceSafe データベースのなかのプロジェクトとワークスペースのなかのプロジェクトを関連付けます。[ツール->バージョン管理] サブメニューからオプションを選ぶか、バージョン管理ツールバーボタンをクリックすることにより、標準コマンドをすばやく起動することができます。

9.1 ワークスペースに Visual SourceSafe を関連づける

以下の節では、Visual SourceSafe と現在のワークスペースとを関連付ける方法を説明します。

9.1.1 Visual SourceSafe を選ぶ

まず、Visual SourceSafe をバージョン管理システムとして選びます。

➲ Visual SourceSafe 5.0 または 6.0 を使うには

1. [ツール->バージョン管理->選択…] を選んでください。 [バージョン管理ツールの選択] ダイアログボックス（図7.3）に、サポートするバージョン管理システムを表示します。
2. バージョン管理システムのリストから [Visual SourceSafe 5.0/6.0] の項目を選び、 [選択] ボタンをクリックしてください。
3. [OK] ボタンをクリックしてください。[SourceSafe Login] ダイアログボックス（図9.1）が表示されます。
4. [Username] にVisual SourceSafeのログインを、 [Password] にパスワードを入力してください。
5. プロジェクトを追加するVisual SourceSafeデータベース（SRCSAFE.INI）へのフルパスをDatabase pathフィールドに入力してください。
6. [OK] ボタンをクリックしてください。 [Create SourceSafe Project] ダイアログボックス（図9.2）が表示されます。
7. [Project name] フィールドに、データベースに作成するプロジェクト名（フォルダ名）が表示されます。プロジェクト名は変更できます。
8. [Project name] フィールド下のツリーにはステップ6で指定したデータベースの構造を示します。[Project name]フィールドに指定したフォルダをどのフォルダ内に作成するかを選んでください。
9. [OK] ボタンをクリックしてください。
10. 現在のワークスペースにあるプロジェクトの数だけ、ステップ７～９を繰り返してください。

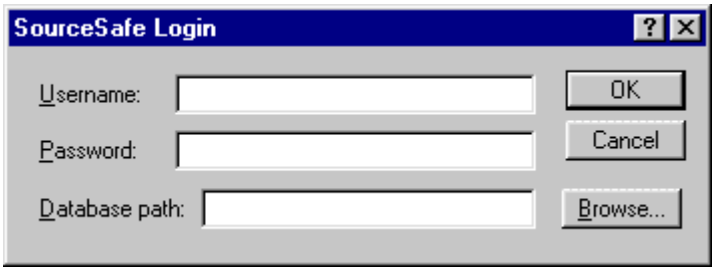


図 9.1: SourceSafe Login **ダイアログボックス**

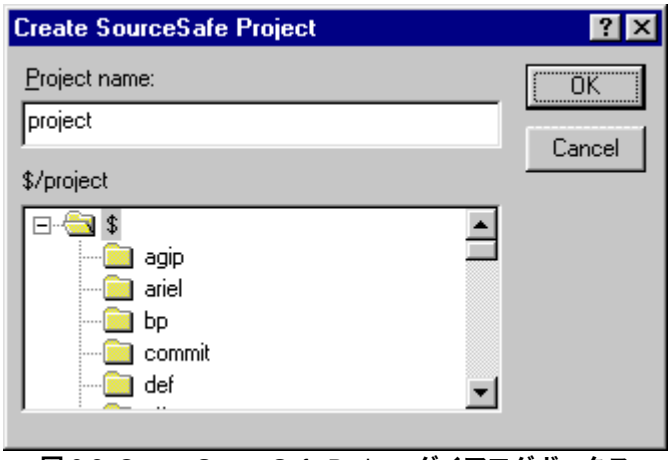


図 9.2: Create SourceSafe Project **ダイアログボックス**

HEW は、Visual SourceSafe 内に必要なプロジェクトを作成して、迅速にアクセスできるように、バージョン管理ツールバーやメニューを設定します。Visual SourceSafe プロジェクト自体は作成しましたが、まだファイルは追加していません。

9.1.2 Visual SourceSafe にファイルを追加する

前節では、ハードディスク上のプロジェクトディレクトリ（作業中のディレクトリ）と Visual SourceSafe のプロジェクトディレクトリ（Visual SourceSafe によって管理されたディレクトリ）との間でマッピングを確立しただけです。ハードディスクのプロジェクトディレクトリ（サブディレクトリを含む）に複数のソースファイルがあってもマッピング先の Visual SourceSafe のディレクトリにはファイルがありません。

まず、バージョン管理システムとして Visual SourceSafe を選びます。

➲ Visual SourceSafe にファイルを追加するには

1. Visual SourceSafeに追加するファイル（複数選択可）を選んでください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはそれらを混在して選ぶこともできます。プロジェクトフォルダ、またはワークスペースフォルダを選ぶと、システムファイルがファイルリストに追加されます。例えば、プロジェクトフォルダを選ぶと、プロジェクトファイルをファイルリストへ追加します。プロジェクトファイルがチェックアウトされ、

バージョンが最後にロードしたときより新しい場合、そのプロジェクトをリロードしたいかどうかをユーザに問い合わせます。

2. ファイル追加ツールボタン（ ）をクリックするか、 [ツール->バージョン管理->ファイルの追加] メニューオプションを選んでください。

Visual SourceSafe にファイルを追加すると、作業中のディレクトリのローカルバージョンは読み取り専用になります。ファイルが追加されたことの確認や、プロジェクト内のすべてのファイルの状態を表示するには、

1. チェックしたいファイルのプロジェクトフォルダを選んでください。
2. ファイルの状態ツールボタン（ ）をクリックするか、 [ツール->バージョン管理->ファイルの状態] メニューオプションを選んでください。
3. [アウトプット] ウィンドウの [Version Control] タブに各ファイルの状態が表示されます。表示された情報により、ファイルがプロジェクトに追加されたかどうか、ファイルがチェックアウトされたかどうか、また、誰によってチェックアウトされたかがわかります。

9.2 Visual SourceSafe コマンド

以下の 8 つの操作ができます。

- バージョン管理にファイルを追加する
- バージョン管理からファイルを削除する
- 読み取り専用ファイルを取得する
- 読み取り/書き込みファイルをチェックアウトする(編集するため)
- チェックアウトしたファイルをチェックインする（編集後 Visual SourceSafe を更新する）
- ファイルのチェックアウト操作をキャンセルする（編集結果をキャンセルする）*
- ファイルの状態を表示する
- ファイルの履歴を表示する*

*他のコマンドは、ツールバーまたはメニューからアクセスできますが、これらのコマンドは [ツール->バージョン管理] サブメニューからしかアクセスできません。

9.2.1 バージョン管理からファイルを削除する

HEW プロジェクトにファイルが表示（[ワークスペース]ウィンドウの[Projects]タブ）されても、それらのファイルが Visual SourceSafe によって管理されているとは限りません。

➲ Visual SourceSafe からファイルを削除するには

1. Visual SourceSafeから削除するファイルを選んでください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在させて選ぶこともできます。
2. ファイル削除ツールバーボタン（ ）をクリックするか、 [ツール->バージョン管理->ファイルの削除] メニューオプションを選んでください。

9.2.2 バージョン管理からファイルの読み取り専用コピーを取得する

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能なコピーを一度に一人のユーザだけが取得できるようにします。しかし、どのユーザもすべてのファイルの読み取り専用コピーを取得できます。

➲ Visual SourceSafe から読み取り専用コピーを取得するには

1. Visual SourceSafeから取得するファイルを選んでください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶこともできます。
2. ファイル取得ツールバーボタン（）をクリックするか [ツール->バージョン管理->ファイルの取得] メニューオプションを選んでください。

9.2.3 バージョン管理からファイルの書き込み可能コピーをチェックアウトする

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。これは、チェックアウトしようとするファイルがすでに他のユーザによりチェックアウトされていない場合のみ可能です。

➲ Visual SourceSafe からファイルの書き込み可能コピーをチェックアウトするには

1. Visual SourceSafeからチェックアウトしたいファイルを選んでください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶこともできます。
2. ファイルチェックアウトツールバーボタン（）をクリックするか [ツール->バージョン管理->ファイルのチェックアウト] メニューオプションを選んでください。
3. 動作が完了すると、ファイルの名前の横に印が付き、チェックアウトされたことを示します。

9.2.4 バージョン管理にファイルの書き込み可能コピーをチェックインする

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。チェックアウトしたファイルを編集してチェックインすると、編集結果を他のユーザが見られるようになります。

➲ 編集した Visual SourceSafe のファイルをチェックインするには

1. Visual SourceSafeに再びチェックインするファイルを選んでください。複数のファイルを選ぶことができます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶこともできます。
2. ファイルチェックインツールバーボタン（）をクリックするか [ツール->バージョン管理->ファイルのチェックイン] メニューオプションを選んでください。

9.2.5 チェックアウト操作を取り消す

Visual SourceSafe はソースファイルを保護して、管理しているファイルの書き込み可能コピーを一度に一人のユーザだけが取得できるようにします。チェックアウト操作をすると、Visual SourceSafe からファイルの書き込み可能コピーをローカルドライブに取得します。チェックアウトしたファイルを編集してチェックインすると、編集結果を他のユーザが見られるようになります。しかし、もしチェックアウト操作を誤って行った場合、または、必要なくなった場合、チェックアウト操作を取り消すことができます。

➲ Visual SourceSafe からのファイルのチェックアウト操作を取り消すには

1. 以前のチェックアウト操作を取り消したいファイルを選んでください。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶこともできます。
2. [ツール->バージョン管理->チェックアウトの取り消し] メニューオプションを選んでください。

9.2.6 ファイルの状態を表示する

HEW プロジェクトにファイルが表示されても（[ワークスペース] ウィンドウの[Projects] タブ）、ファイルが Visual SourceSafe に管理されているとは限りません。Visual SourceSafe に管理されているファイルのうち、チェックインされたり、チェックアウト（ユーザが編集するため）されたりするものがあります。状態コマンドは、現在のファイルの状態を表示します。

➲ Visual SourceSafe のファイルの状態を表示するには

1. 状態を表示するファイルを選んでください。複数のファイルを選ぶこともできます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶことができます。
2. ファイルの状態ツールバーボタン（ ）をクリックするか [ツール->バージョン管理->ファイルの状態] メニューオプションを選んでください。
3. ファイル名の横に青い印があるとき、そのファイルはチェックアウトされていることを示します。

9.2.7 ファイル履歴を表示する

Visual SourceSafe はプロジェクトのファイルへの編集を管理します。ファイルが最初にプロジェクトに追加されたときからの編集内容の完全なファイル履歴を表示できます。

➲ Visual SourceSafe のファイル履歴を表示するには

1. ファイル履歴を表示するファイルを選んでください。複数のファイルを選ぶことができます。ファイルフォルダ、プロジェクトフォルダ、ワークスペースフォルダ、またはこれらを混在して選ぶことができます。
2. [ツール->バージョン管理->Show History] メニューオプションを選んでください。

9.3 Visual SourceSafe 統合化オプション

[ツール->バージョン管理->構成…]を選ぶことにより、履歴コマンドや状態コマンドの表示方法を変更できます。

履歴コマンドの結果をダイアログボックスで表示するには、[Display dialog box for history] チェックボックスをチェックしてください。履歴コマンドの結果を[アウトプット] ウィンドウの[Version Control]タブで表示するには、[Display dialog box for history] チェックボックスのチェックをはずしてください。

状態コマンドの結果をダイアログボックスで表示するには、[Display dialog box for file status]チェックボックスをチェックしてください。状態コマンドの結果を[アウトプット] ウィンドウの[Version Control]タブで表示するには、[Display dialog box for file status]チェックボックスのチェックをはずしてください。

10. ネットワーク機能

10.1 概要

High-performance Embedded Workshop は、ネットワークを介してワークスペースやプロジェクトを共有できます。したがって、ユーザは共有したプロジェクトを同時に操作してお互いの変更を見ることができます。このシステムは、バージョン制御と連動して使用できます。このシステムの主な特長は、ほかのすべてのユーザがプロジェクトをリロードして変更を無効にすることなく、各ユーザがワークスペースやプロジェクトを変更したり更新したりできることです。

このシステムは、ネットワーク上のひとつのコンピュータをサーバマシンとして使うことで実現します。ほかのすべてのクライアントマシンはサーバマシンが提供するサービスを利用できます。例えば、クライアントマシンのひとつが新規ファイルを追加すると、サーバマシンに通知されます。すると、サーバマシンはほかのすべてのクライアントマシンに追加を通知します。この構成を図 10.1 に示します。

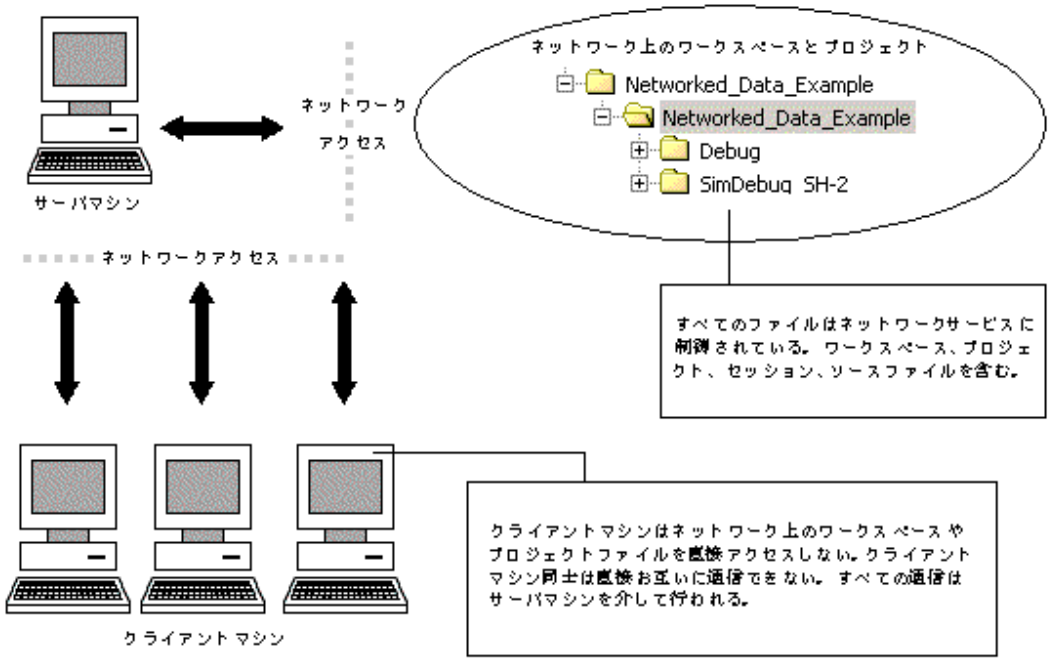


図 10.1: ネットワークデータベースの構成

ネットワーク機能によってユーザはファイルへのアクセス権を与えられます。これによって、プロジェクトアドミニストレータから書き込みを許可されたユーザのみが、プロジェクトやソースファイ

ルを変更できるようになります。アドミニストレータは、各ユーザの書き込み権を、プロジェクトの各自のエリアに制限して、ほかのエリアを読み出し専用にすることができます。こうして、各ユーザによるプロジェクトのほかの部分への競合や破壊を未然に防きます。これらの制限は異なるレベルに設定できます。詳細は、この章で後述します。

【注】 動作によっては、あるクライアントマシンが実行している間は、他のクライアントマシンでロックされていることがあります。例えば、ひとつのマシンがツールチェインオプションを変更している間、ほかのすべてのマシンではこのデータが読み出し専用になります。

【注】 ネットワーク機能を使うと、HEW の性能は低下します。小規模なチームで作業する場合は、単一ユーザモードとバージョン制御を使うことを推奨します。

10.1.1 ネットワークアクセスを使う

➲ ネットワークアクセスを使うには

1. [ツール->オプション] を選んでください。HEWの[オプション]ダイアログボックスが表示されます。
2. [ネットワーク] タブを選んでください（図10.2）。
3. [ネットワークデータアクセス有効化] チェックボックスをチェックしてください。パスワードなしでアドミニストレータがシステムに追加されます。
   アドミニストレータのみが、ユーザを追加したり、ユーザアクセス権を変更したりできます。アドミニストレータは最高のレベルのアクセス権を持ちます。
4. [ネットワーク] ダイアログボックスで、アドミニストレータはパスワードを設定しなければなりません。以下に説明します。

10.1.2 アドミニストレータユーザのパスワードを設定する

➲ パスワードを設定するには

1. 前述の手順に従って操作してください。
2. [パスワード] ボタンをクリックしてください。ネットワークデータアクセスを有効にしたとき、このボタンも有効になります。
3. [パスワード変更] ダイアログボックスが表示されます（図10.3）。
4. 上のフィールドでは、ユーザ名は読み出し専用です。この場合、Adminになります。
5. 両方のフィールドに新規パスワードを入力して、 [OK] ボタンをクリックしてください。
6. [ツール -> オプション] ダイアログボックスの [ネットワーク] タブにユーザとパスワードが設定されます。
7. これで、 [ツール -> オプション] ダイアログボックスを閉じることができます。
8. ダイアログボックスを閉じると、ワークスペースを保存して再び開くかどうか質問されます。これは、共有アクセスモードでワークスペースを再び開く必要があるためです。変更を保存しないと、変更は無効になります。
9. ワークスペースを再び開くと、システムに再びログインするかどうかを決めるダイアログボックスが開きます。ログインすると、現在のあなたのアクセス権を示すダイアログボックスが表示されます。例えば、アドミニストレータユーザの場合、レベルはアドミニストレータになります。このダイアログボックスを閉じると、HEWサーバウィンドウが開き、ネットワーク機能が使えるようになります。

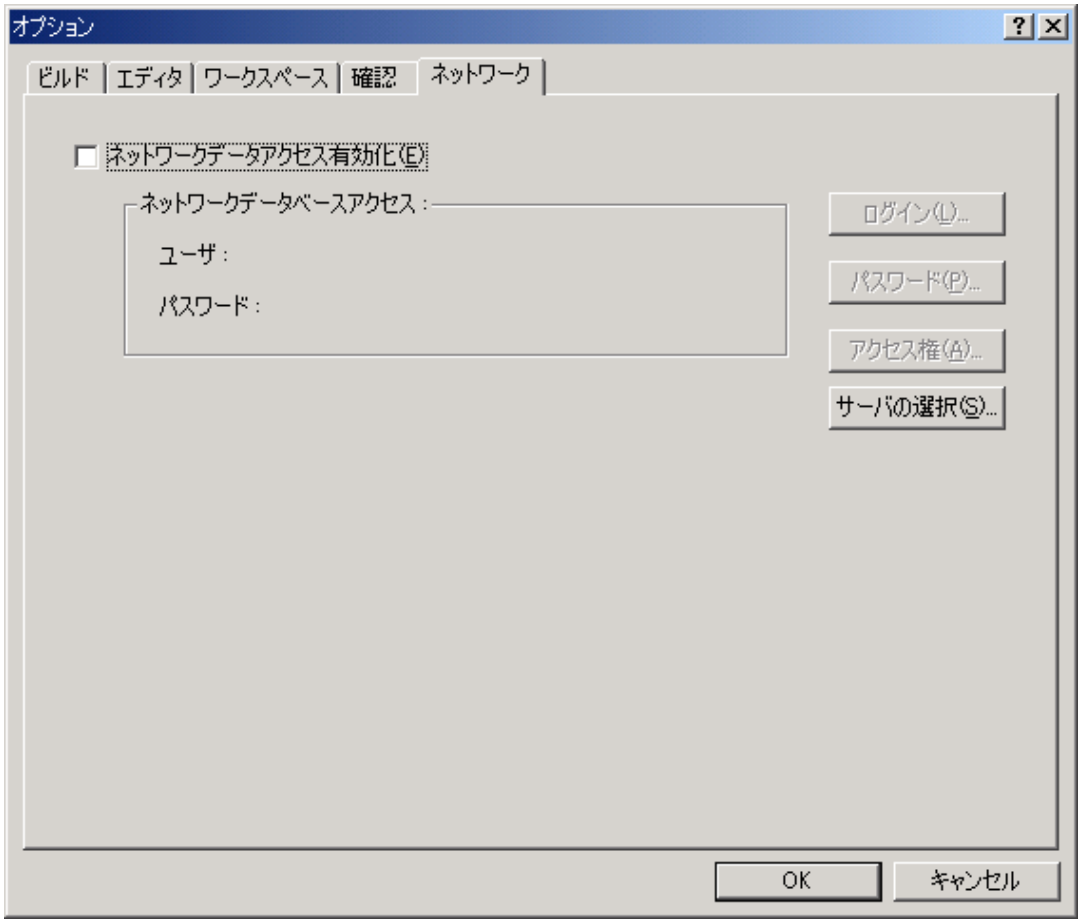


図 10.2: [**ネットワーク**]**タブの初期設定**

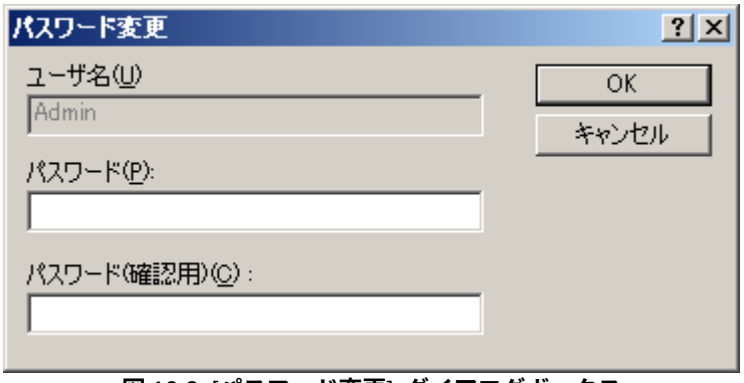


図 10.3: [**パスワード変更**] **ダイアログボックス**

10.1.3 新規ユーザを追加する

ネットワークデータベースの初期設定では、アドミニストレータユーザとゲストユーザをシステムに追加します。HEW システムでは、以下のアクセス権のレベルを設定できます。

- アドミニストレータ
  HEW のすべての機能にフルアクセスできます。プロジェクトからユーザを削除したり、アクセス権レベルを変更したりできます。ワークスペース、プロジェクトファイル、ソースファイルを変更できます。

- 読み出し/書き込みフルアクセス
  ワークスペース、プロジェクトファイル、ソースファイルを変更できます。ユーザのアクセス権レベルは変更できません。

- 読み出し/書き込みファイルアクセス
  ソースファイルのみ変更できます。すべてのプロジェクト設定を表示できますが、変更できません。

- 読み出し専用
  すべてのソースファイルとプロジェクトファイルが表示できます。変更はできません。

どのユーザも、ネットワーク上で使用できるプロジェクトを開くときにはユーザ名とパスワードを入力しなければなりません。入力しないと、アクセス権は与えられません。入力すると、ユーザには上記のアクセス権レベルのうちのいずれかのレベルを与えられます。

➲ 新規ユーザを追加するには

1. アドミニストレータのアクセス権をもつユーザがログインします。この詳細は、上記を参照してください。
2. [ツール->オプション] を選んでください。HEWの[オプション]ダイアログボックスが表示されます。
3. [ネットワーク] タブを選んでください（図10.2）。
4. [アクセス権...] ボタンをクリックしてください。図10.4に示すダイアログボックスが表示されます。
5. [追加…] ボタンをクリックしてください。[新規ユーザ追加]ダイアログボックスが表示されます。あなたをアドミニストレータユーザとして新規のログイン名とパスワードを追加できます。通常、パスワードはデフォルトのテキストか空白のままで設定します。次に、[OK] ボタンをクリックしてください。
6. [OK] ボタンをクリックすると、ユーザは読み出し専用のアクセス権レベルで追加されます。アクセス権レベルを変更するには、ユーザ名を選んで必要なラジオボタンをクリックしてください。[OK] ボタンをクリックすると、アクセス権レベルの変更が保存されます。

➲ 既存ユーザを削除するには

1. アドミニストレータのアクセス権をもつユーザがログインします。この詳細は、上記を参照してください。
2. [ツール->オプション] を選んでください。HEWの[オプション]ダイアログボックスが表示されます。
3. [ネットワーク] タブを選んでください（図10.2）。
4. [アクセス権…] ボタンをクリックしてください。図10.4に示すダイアログボックスが表示されます。

5.　削除するユーザ名を選んでください。
6.　[削除] ボタンをクリックしてください。
7.　[OK] ボタンをクリックするとアクセス権の変更が保存されます。

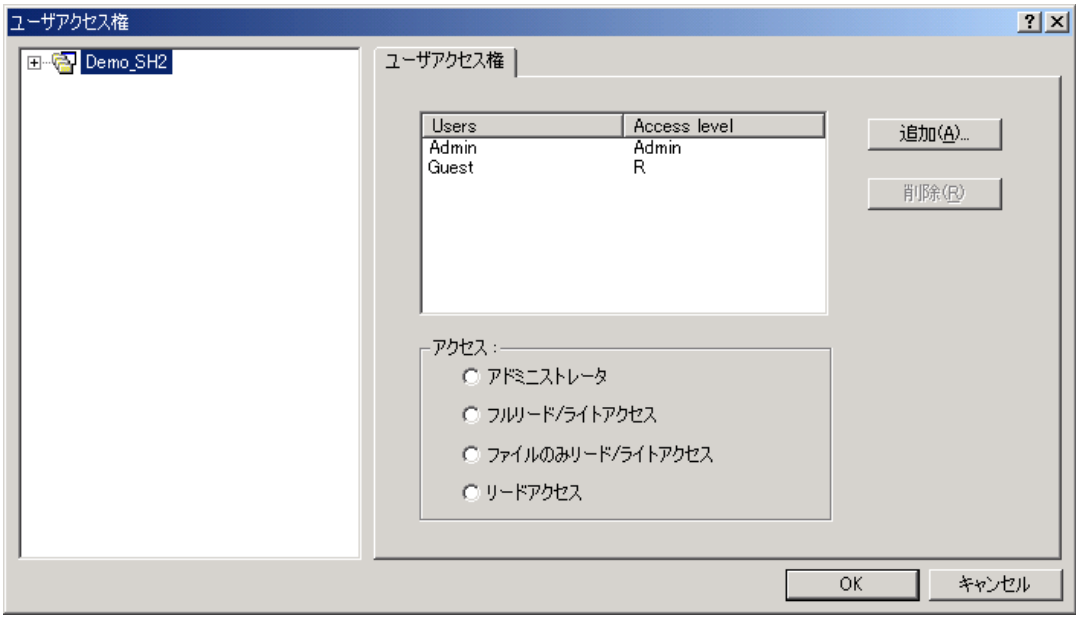


図 10.4: [**ユーザアクセス権**] **ダイアログボックス**

10.1.4　パスワードを変更する

➲ パスワードを変更するには

1.　パスワードを変更するHEWネットワークデータベースにログインして、 [ツール -> オプション] を選んでください。[オプション]ダイアログボックスが表示されます。
2.　[ネットワーク] タブを選んでください（図10.2）。
3.　[パスワード] ボタンをクリックしてください。
4.　新しいパスワードを入力して 2 番目のエディットボックスで確認してください。
5.　[OK] ボタンをクリックしてください。
6.　次に、 [OK] ボタンをクリックするとパスワードの変更が保存されます。

10.1.5　ネットワーク HEW サービスを利用する

最初にネットワーク上のプロジェクトに接続するとき、HEW は自動的に正しいネットワーク HEW サービスを探して接続します。これはマシン名で定義されます。マシン名でワークスペースのサービスが見つからないとき、図 10.5 に示すダイアログボックスが表示されます。サービスのある場所を入力するか参照して、[OK] ボタンをクリックしてください。設定するマシンをサーバマシンとして使用する場合は、ラジオボタンをデフォルトの設定(ローカル)のままにしてください。

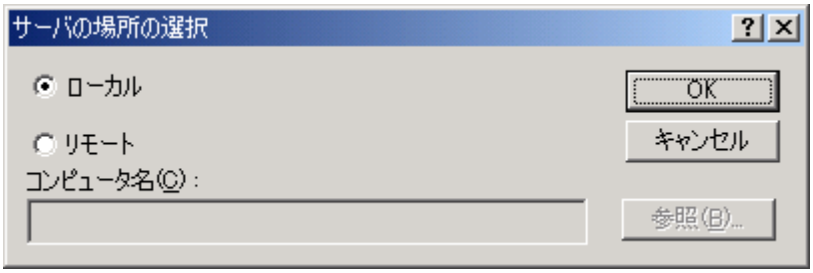


図 10.5: **マシン名の選択**

ワークスペースのサーバマシンとして使用していたマシンを他のマシンに接続しようとすると、図 10.6 のメッセージが表示されます。[OK] ボタンをクリックすると新しいマシンに接続します。

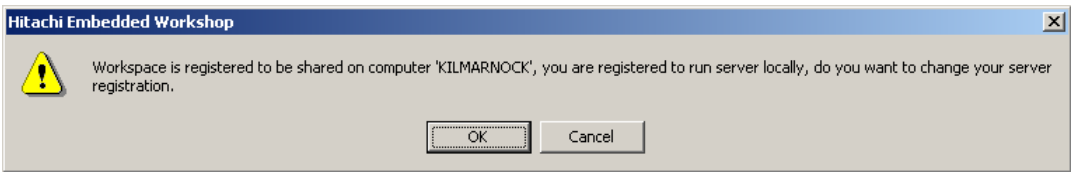


図 10.6: **マシン名の選択**

【注】 ネットワークサービスを有効にして複数の HEW **ワークスペースを実行している場合、ユーザは一度にひとつのワークスペースにしかアクセスできません。例外は、同じマシンがすべてのネットワークワークスペースを提供している場合です。**

11. 相違点の表示

High-performance Embedded Workshop には、相違点の表示があります。ドライブのローカルファイルやバージョン制御システムのファイルを比較して相違点を詳細に表示します。HEW のバージョン 3.0 以降の Visual SourceSafe にこの機能があります。

[差分]ウィンドウを起動するには、２通りの方法があります。１つめは、[ツール]メニューの[差分の表示...]メニュー項目を選びます。２つめは、[ワークスペース]ウィンドウのポップアップメニューの[差分の表示]メニュー項目です。ポップアップメニューを使うと、現在のファイルの選択が自動的にエディットボックスに追加されます。[差分の表示]メニュー項目をクリックすると、図 11.1 に示すダイアログボックスが表示されます。

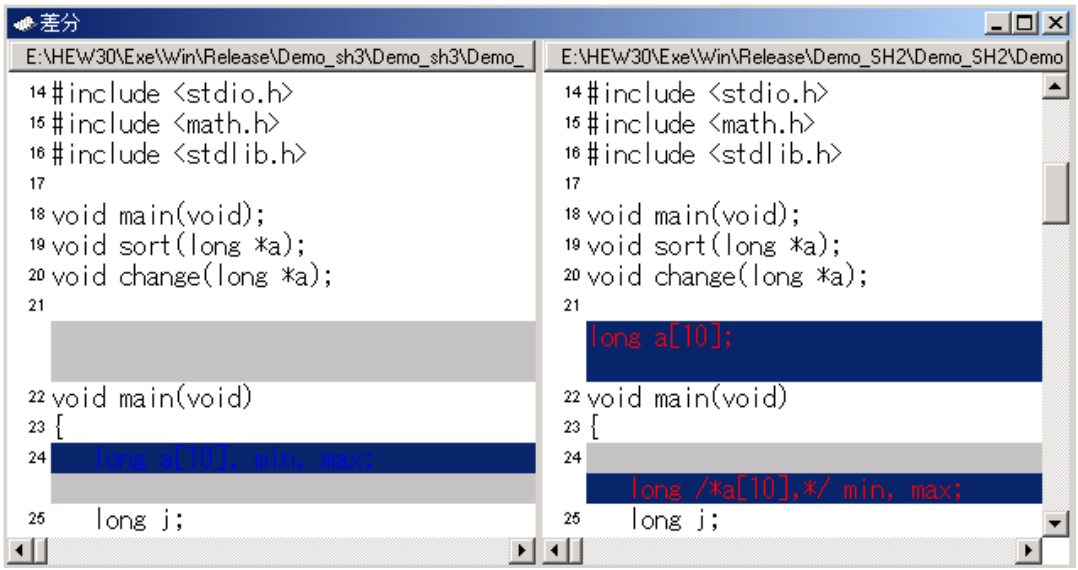


図 11.1: 相違点の表示

➲ ローカルドライブの２つのファイルの相違点を比較するには

1. [ツール->差分の表示...] を選んでください。図11.2の[ファイルの比較]ダイアログボックスが表示されます。
2. [ドライブ上のファイルと比較] ラジオボタンが有効であることを確認してください。
3. 比較する２つのファイルを入力してください。前回の相違点の比較を選ぶか新しいファイルをブラウズできます。
4. [詳細…] ボタンをクリックすると、図11.3のダイアログボックスが表示されます。空白を考慮せずに相違点の比較ができます。この[詳細オプション]ダイアログボックスを終了するには[OK]ボタンをクリックしてください。
5. [比較] ボタンをクリックしてください。
6. 相違点が表示されます。比較する２つのファイルは左右に分割された表示にロードされま

す。ファイル名は各ウィンドウの上に表示されます。

➲ ローカルファイルと SourceSafe のファイルの相違点を比較するには

1. SourceSafeが有効であることを確認してください。また、ファイルはバージョン制御システムに追加してください。
2. [ツール -> 差分の表示...] を選んでください。図11.2の[ファイルの比較]ダイアログボックスが表示されます。
3. [バージョン管理されているファイルとの比較] ラジオボタンが有効になっていることを確認してください。
4. 比較する１つめのファイルを入力してください。前回の相違点の比較を選ぶか新しいファイルをブラウズできます。
5. [詳細…] ボタンをクリックすると図11.3のダイアログボックスが表示されます。空白を考慮せずに相違点の比較ができます。この[詳細オプション]ダイアログボックスを終了するには[OK]ボタンをクリックしてください。
6. [比較] ボタンをクリックしてください。
7. 図11.1のように、相違点が表示されます。

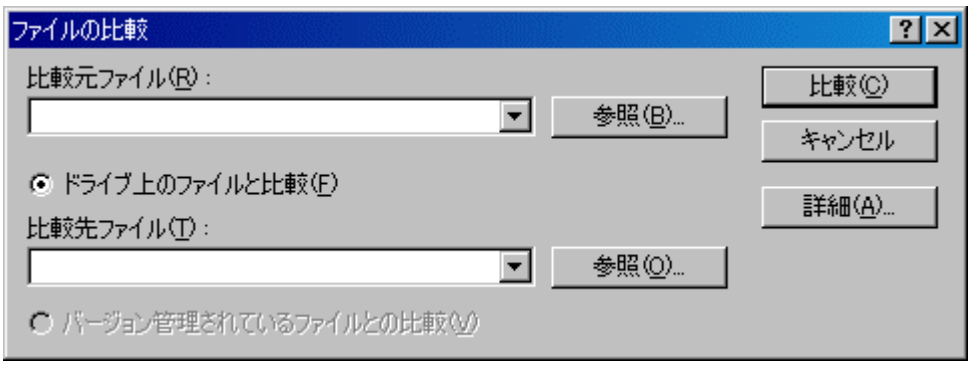


図 11.2: [**ファイルの比較**]**ダイアログボックス**

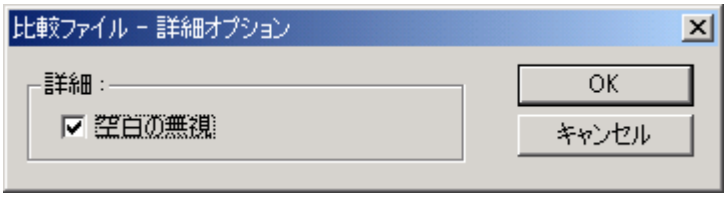


図 11.3: [**詳細オプション**] **ダイアログボックス**

相違点の表示のアクセスには２通りの方法があります。図 11.4 に示す相違点の表示用のツールバーを使用するか、または、[差分]ウィンドウ上でマウスを右クリックすると、図 11.5 のポップアップメニューが表示されます。

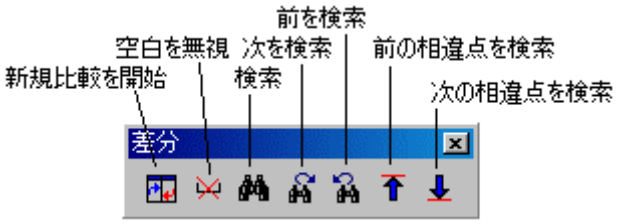


図 11.4: **[差分] ツールバー**

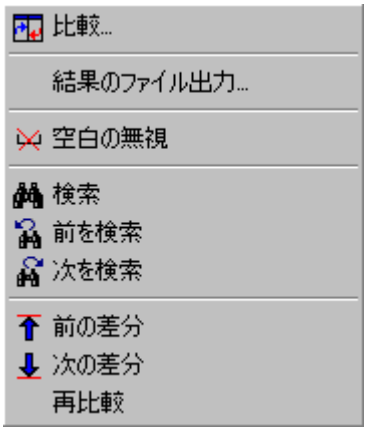


図 11.5: **[差分] ポップアップメニュー**

各メニュー項目の機能を以下に説明します。

- 比較
  新しい[ファイルの比較]ウィンドウを開きます。新しいファイルを比較して相違点を表示できます。
- 結果のファイル出力
  ダイアログボックスが開き、現在の相違点の結果をテキストフォーマットで保存することができます。
- 空白の無視
  空白文字を無視します。このメニュー項目は[詳細オプション]ダイアログボックスの[空白の無視]オプションから選択することもできます。
- 検索
  標準の[検索]ダイアログボックスを表示します。HEW エディタと同じダイアログボックスです。
- 次を検索
  検索項目に当てはまる次の前方の文字列を検索します。
- 前を検索
  検索項目に当てはまる次の後方の文字列を検索します。
- 前の差分
  次の後方の相違点に自動的にジャンプします。
- 次の差分
  次の前方の相違点に自動的にジャンプします。

- 再比較
  表示をリフレッシュして再び手動で相違点の比較を実行します。この機能は、どちらかのファイルを比較後、変更したときに便利です。

[差分]ウィンドウでは、表示の色を変えることができます。この手順を以下に示します。

➲ [差分]ウィンドウで表示する色を変更するには

1. [ツール -> 表示形式...] を選んでください。図11.6 に示す[表示形式]ダイアログボックスが表示されます。
2. [差分] を拡張表示してください。ダイアログボックスは、図11.6のようになります。
3. 変更するカテゴリを選んでください。[カラー]タブで変更できます。選択できるカテゴリは、左側で異なる行、左側で移動した行、右側で異なる行、右側で移動した行です。
4. [OK] ボタンを押すと変更が有効になります。

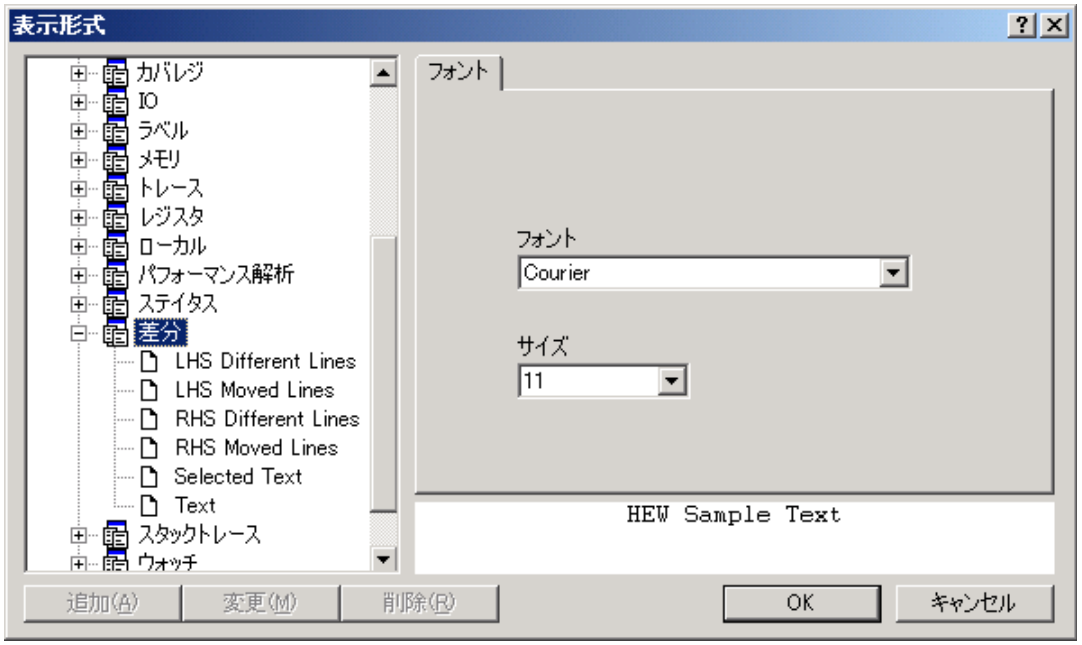


図 11.6: [表示形式] ダイアログボックス

12. テクニカルサポート

High-performance Embedded Workshop には、いくつかの統合されたテクニカルサポート機能があります。

➲ HEW 製品のバージョンアップやサービスパックを確認するには

1. [ヘルプ -> テクニカルサポート -> アップデートのWeb確認] を選んでください。
2. デフォルトのウェブブラウザが起動して、あなたの地域のHEWダウンロードページが表示されます。
3. HEWのバグの修正や新規機能の追加の更新情報を確認してください。

ときには、HEW アプリケーションで予期せぬ問題が起きるかもしれません。問題が起きてアプリケーションが破壊されると、自動的に HEW バグトラッキングプログラムが起動します。これにより、バグの情報を集約でき、様々な方法でテクニカルサポートに送ることができます。このバグトラッキングプログラムは手動で起動することもできます。この方法を以下に説明します。

➲ HEW バグ情報を作成するには

1. [ヘルプ -> テクニカルサポート -> 不具合レポートの作成] を選んでください。
2. HEWシステムから詳細情報が生成されます。これには多少，時間がかかるかもしれません。その後、図12.1のような、[Submit a Bug Report]ダイアログボックスが表示されます。
3. 大きなエディットボックスに、起きた問題に関する情報を追加することができます。
4. 報告を作成したら、[How would you like to submit the report?] ドロップリストで報告を提出する方法を選ぶことができます。印刷、電子メール、ディスクに保存の3種類があります。
5. [Submit] ボタンをクリックすると選んだ方法で報告を提出することができます。

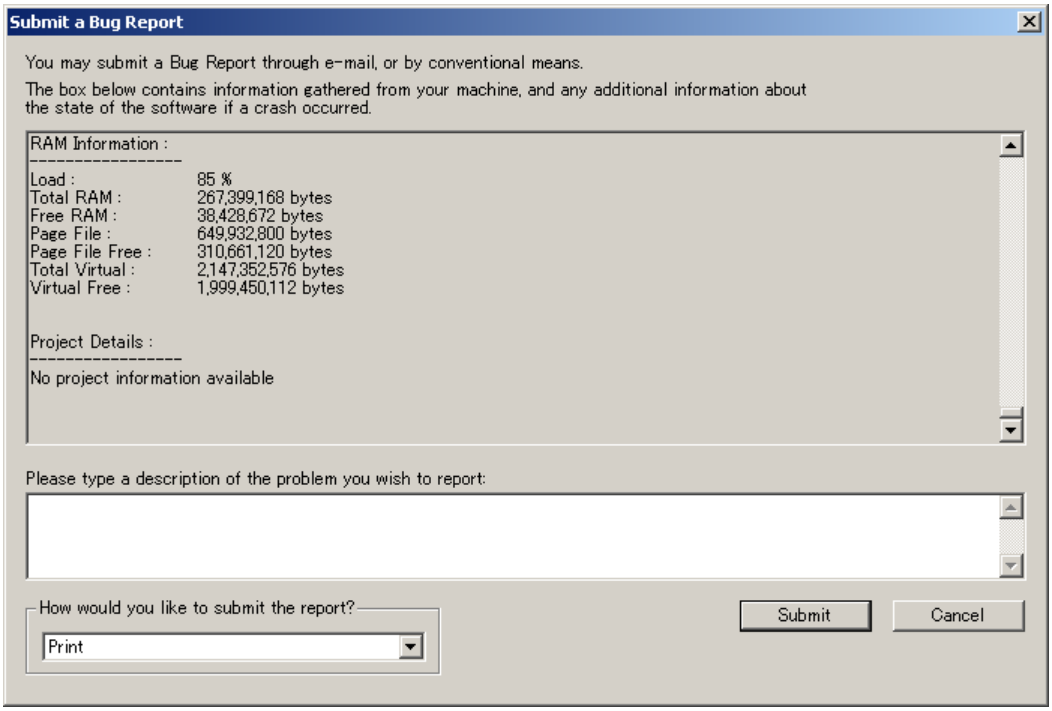


図 12.1: [Submit a Bug Report]**ダイアログボックス**

13. ナビゲーション機能

High-performance Embedded Workshop には、いくつかの統合されたナビゲーション機能があります。[Navigation]ウィンドウは[Projects]ウィンドウ、および[Templates]ウィンドウの横にあります（図 13.1）。

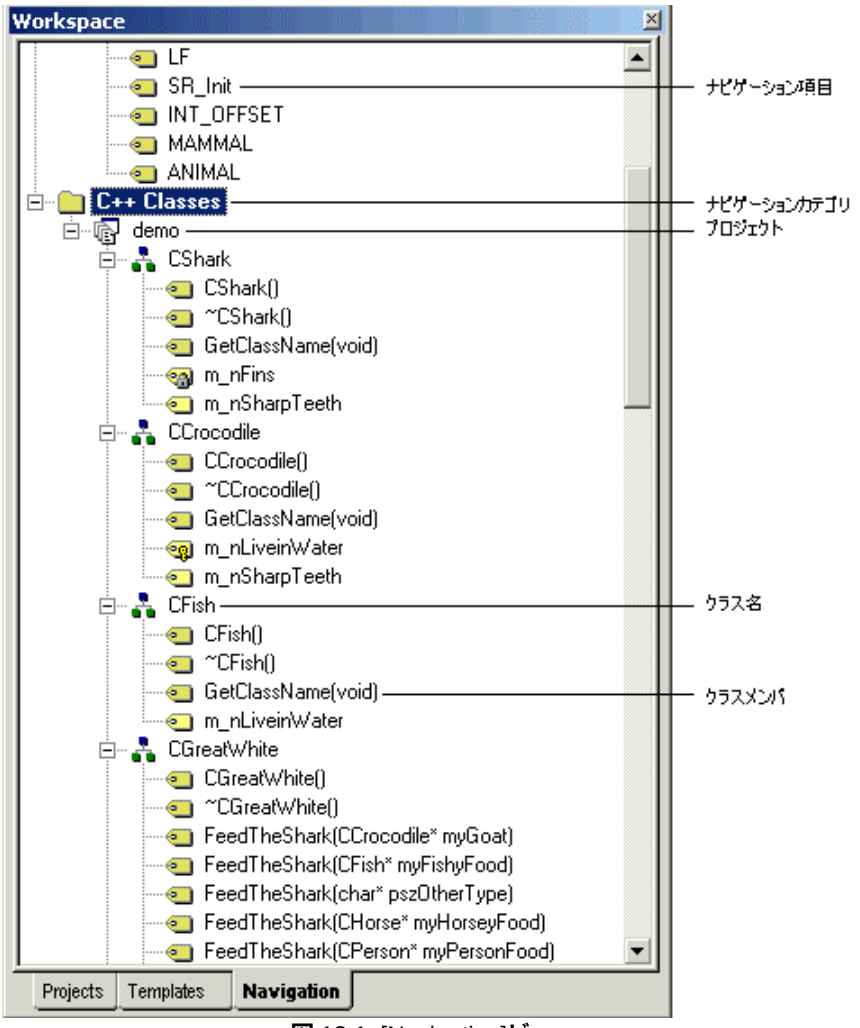


図 13.1: [Navigation]**ビュー**

[Navigation]ビューではサポートされているすべてのナビゲーションの種類を表示します。HEW3.0 以降では、以下のナビゲーション機能を標準でサポートしています。

- C Defines：C, C ソースファイルのすべての#define の表示

- C Functions：C ソースファイルのすべての ANSI C 標準関数の表示
- C++ Classes：C++ソースファイルのすべてのクラス、関数、メンバの表示

各カテゴリは、ビューの上位レベルに表示されます。各カテゴリの下にワークスペースの各プロジェクトが表示されます。特定のプロジェクトに属する項目はプロジェクトのアイコンの下に表示されます。

特定のナビゲーションカテゴリの情報を必要としない場合はスキャンを無効にすることができます。

➲ ナビゲーションカテゴリを無効にするには

1. [Navigation] ウィンドウ上でマウスの右ボタンをクリックしてください。図13.2のポップアップメニューが表示されます。
2. [カテゴリの選択…] メニュー項目を選んでください。
3. 図13.3に示すダイアログボックスが表示されます。
4. 定義を参照する必要のないカテゴリのチェックをはずしてください。
5. [OK] ボタンをクリックしてください。

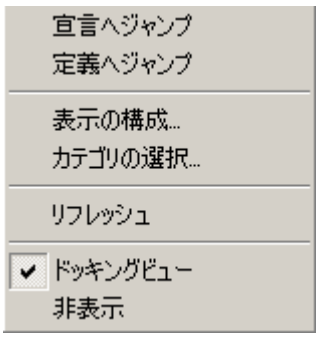


図 13.2: [C++ **ナビゲーション]ポップアップメニュー**

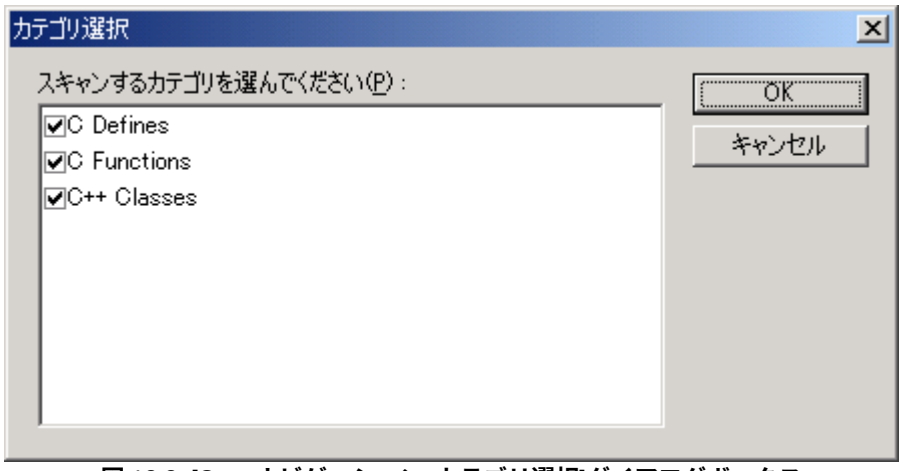


図 13.3: [C++ ナビゲーション カテゴリ選択]ダイアログボックス

【注】 ファイルがスキャンされるにしたがい、ナビゲーション項目が表示されます。つまり、ファイルが多数あると、ナビゲーションビューの更新を終了するのに時間がかかります。
ファイルは、保存するときに再びスキャンされます。つまり、ファイルを保存するまで、新しいクラスや関数のナビゲーション情報は表示されません。

13.1 C++ナビゲーション機能

C++ナビゲーション機能は、サポートする３つのナビゲーション機能の中で最も複雑です。C++ソースファイルの表示で以下の構成をサポートします。情報の基本構成を図 13.4 に示します。

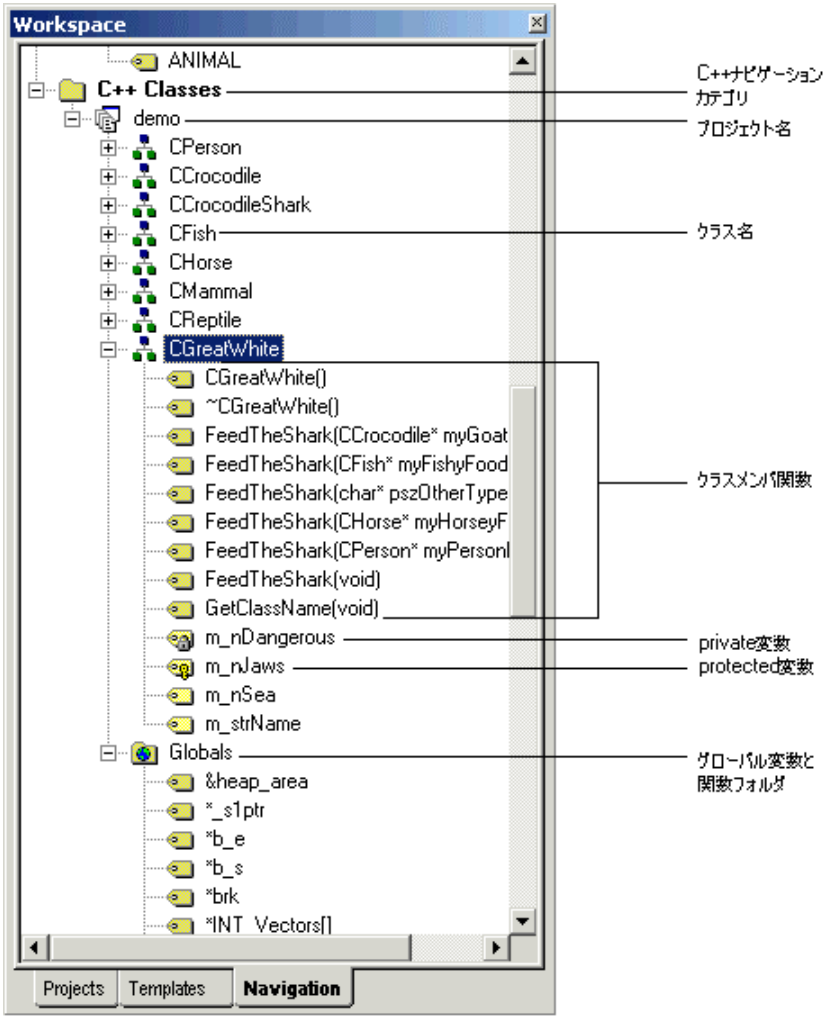


図 13.4: C++ナビゲーション情報

C++ナビゲーション表示では、アイコンの示す関数や変数の種類を説明するために多くのアイコンを使います。これらを図 13.5 に示します。

アイコン	説明
	public関数
	protected関数
	private関数
	public変数
	protected変数
	private変数

図 13.5: C++**ナビゲーションアイコン**

ナビゲーションビューでは、ソースコードを効率良く見て回ることができます。デフォルトで、ナビゲーション項目をダブルクリックすると、関連するナビゲーション項目の定義にジャンプします。通常、定義はソースファイルにあり、宣言はヘッダファイルにあります。

このデフォルト動作は、[表示の構成]ダイアログボックスで変更できます。[表示の構成]ダイアログボックスではナビゲーションビューでのデータの表示方法を変えることができます。

➲ C++ナビゲーション表示データを構成するには

1. [Navigation] ウィンドウのC++ナビゲーション項目上でマウスを右クリックしてください。図13.2に示すポップアップメニューが表示されます。
2. [表示の構成…] メニュー項目を選んでください。
3. 図13.6に示すダイアログボックスが表示されます。
4. 変更する項目を決めて [OK] ボタンをクリックしてください。

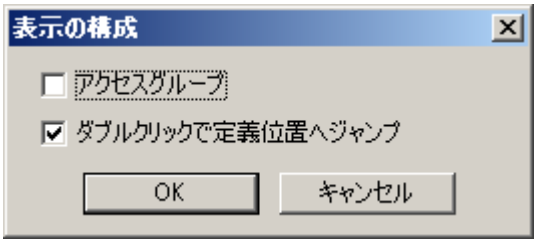


図 13.6: [**表示の構成**]**ダイアログボックス**

[表示の構成]ダイアログボックスで使用できる項目を以下に示します。

- アクセスグループ
  ナビゲーション表示で、public、private、protected メンバ変数と関数をグループ化します。
- ダブルクリックで定義位置にジャンプ
  デフォルトでこのオプションはチェックされています。このオプションをオフにすると、ダイアログボックスが表示され、どこにジャンプするかを質問します。HEW ではなくユーザが位置を決めます。このダイアログボックスを図 13.7 に示します。

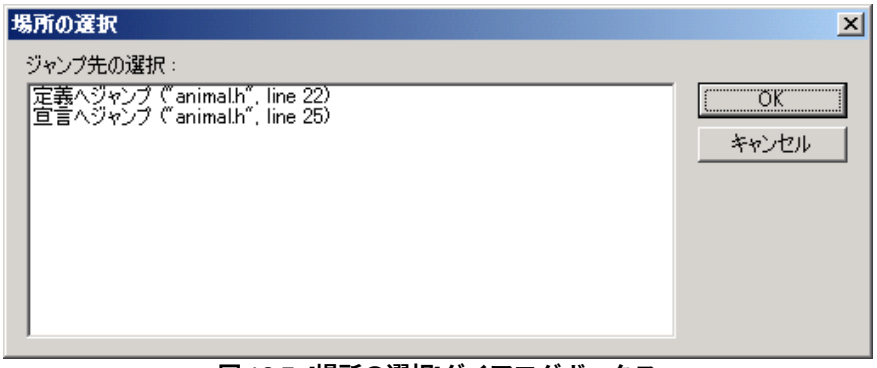


図 13.7: [場所の選択]ダイアログボックス

他の便利な機能としては、特定の範囲の基底クラス、または抽出クラスを表示します。

➲ 基底クラス、抽出クラスを表示するには

1. ナビゲーションビューでクラスを選んでください。
2. マウスを右クリックして、ポップアップメニューを表示してください。
3. [派生クラスの表示]メニュー項目をクリックすると、選択した範囲の導出クラスが表示されます。[基底クラス表示]メニュー項目をクリックすると、選択した範囲の基底クラスが表示されます。
4. 選択により、拡張ツリー形式で選択されたクラス構造がダイアログボックスに表示されます。
5. 必要な情報を取得後、[閉じる]ボタンをクリックしてダイアログボックスを閉じてください。

13.2 C 関数と#define ナビゲーション機能

この機能では、ナビゲーション表示に関数と#define 定義を追加します。ナビゲーション表示に追加すると、ラベルをダブルクリックしてこれらの定義にジャンプできるようになります。

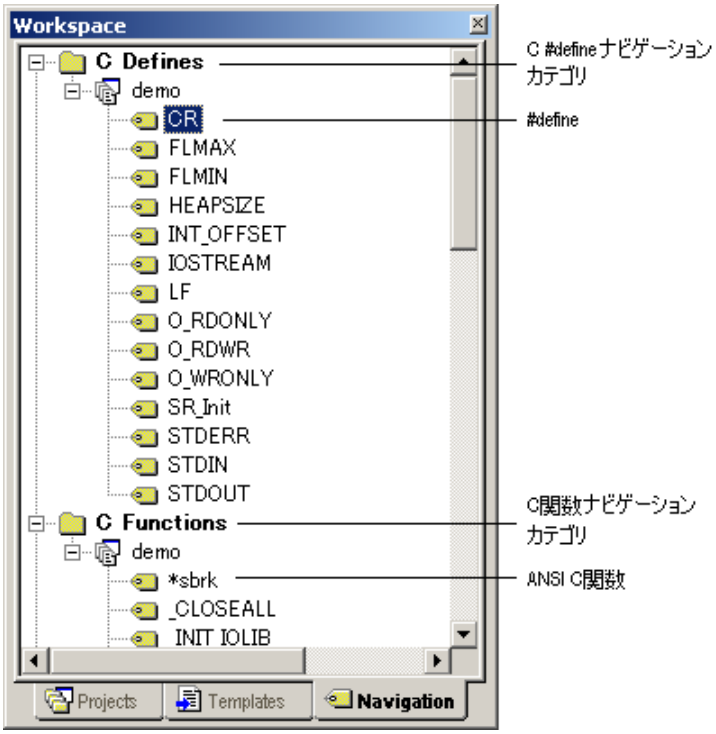


図 13.8: C 関数と#define ナビゲーション情報

14. スマートエディタ

High-performance Embedded Workshop のもうひとつの機能にスマートエディタがあります。これはデフォルトではすべてのC++ソースファイルで有効です。これにより、C++クラスやメンバ関数を使っているとき、HEW エディタは C++ナビゲーション情報にアクセスして自動補完ヘルプを提供します。

➲ スマートエディタの状態を表示するには

1. [ツール->オプション] を選んでください。
2. [オプション] ダイアログボックスの [エディタ] タブを選んでください。
3. 図14.1に示すダイアログボックスが表示されます。[C++用Smart-edit機能]チェックボックスをチェックしてください。
4. [OK] ボタンをクリックしてください。

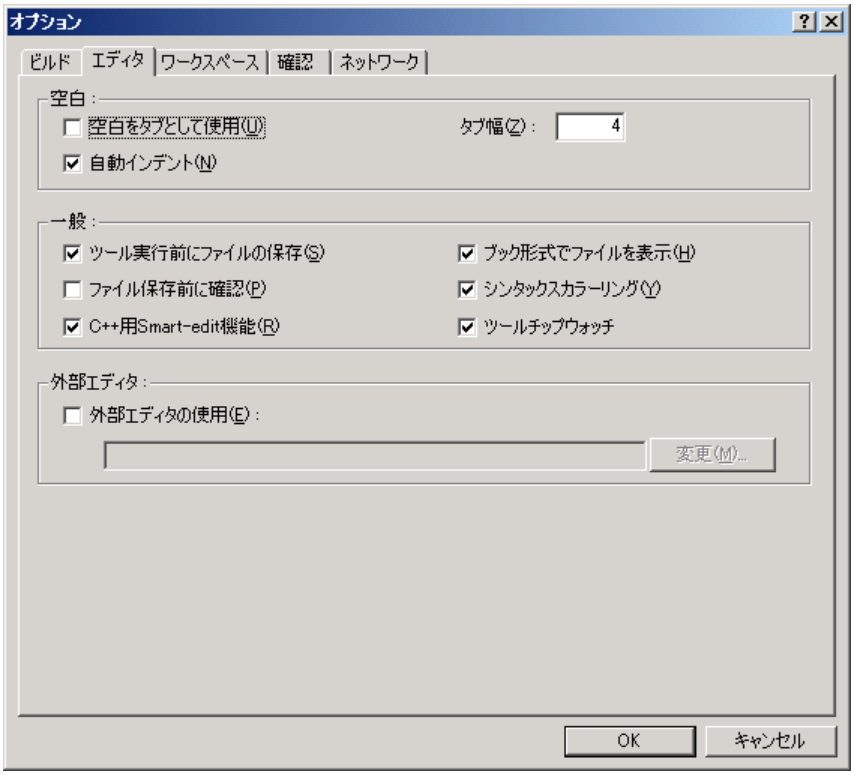


図 14.1: **オプションダイアログボックス　エディタ　タブ**

このオプションをオンにすると、操作している C++ファイルでスマートエディタ機能が有効になります。

【注】 **C++クラスナビゲーションカテゴリがオフのとき、HEW のスマートエディタ機能は無効です。C++クラスナビゲーションの詳細は「13. ナビゲーション機能」を参照してください。**

通常の使い方では、以下のエディタ動作によりスマートエディタ機能が表示されます。

- オブジェクトを使っていて「.」や「->」でメンバをアクセスしているときは、ポップアップメニューが表示され、入力するより効率良くメンバを選ぶことができます。入力中、ポップアップメニューはユーザが押下したキーを覚えているため、選びやすくなります。リターンキーを押すと、選択したメンバが追加されます。「::」を使っているときもこのポップアップメニューが表示されます（図 14.2 参照）。

- C++関数を使っているときは、最初に開きのかっこを入力すると、図 14.3 のダイアログボックスが表示されます。このダイアログボックスにより、現在のオブジェクトでどの関数が使用できるかわかります。関数を選ぶと自動的に残りのパラメータが入力されます。

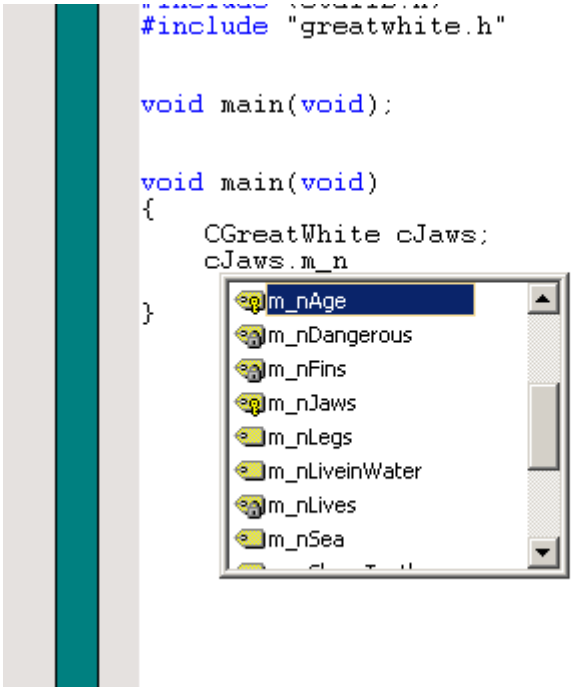


図 14.2: **スマートエディタのメンバの選択**

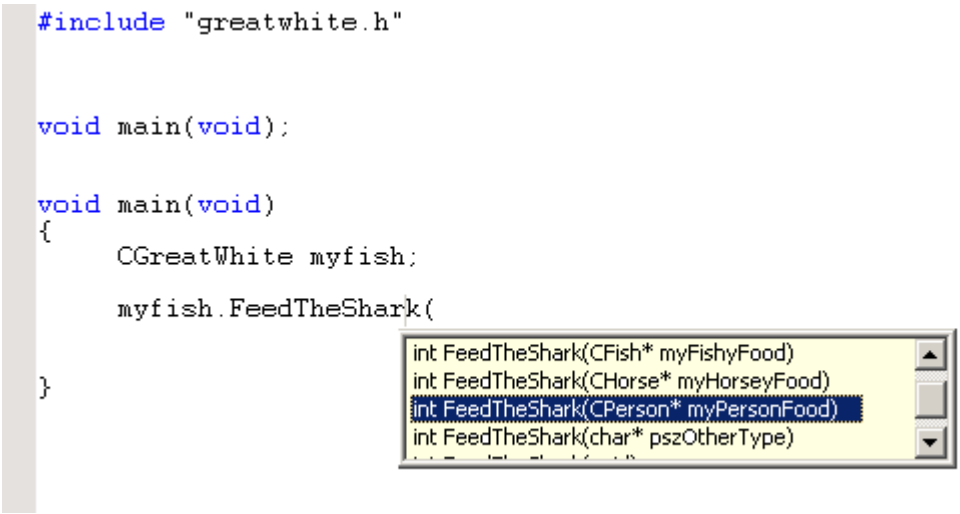


図 14.3:**スマートエディタの関数の選択**

【注】 **スマートエディタの自動補完は様々な理由で動作しないことがあります。例えば、ファイルがナビゲーション機能によりスキャンされていなければなりません。スキャンするには、ファイルが保存されている必要があります。また、**#define **が考慮されていないため、コードの冗長な部分が表示される可能性があります。さらに、コードの構文が不正確な場合、スマートエディタはコードを正しく解析できず、スマートエディタが機能しないことがあります。このとき、ポップアップメニューは表示されません。スマートエディタ機能はマクロの使用をサポートしないことに注意してください。**

エミュレータデバッガ編

1 はじめに

1.1 特長

本エミュレータには、次のような特長があります。

(1) ブレークポイント・メモリマップ・パフォーマンス・トレースをダイアログボックス上で設定することができます。
また、下記のような特徴を持ちます。
- 直観的なユーザインタフェース
- オンラインヘルプ
- 共通した表示と操作性

(2) 各種ホストインタフェースをサポート
ホストコンピュータと接続するためにPCIインタフェース、PC Card (PCMCIA) インタフェース、USBインタフェース、LANインタフェースが使用できます。

(3) リアルタイムエミュレーション
CPUの最高動作周波数でのリアルタイムエミュレーションができます。

(4) 優れた操作性を実現
HEWの使用により、マウスなどのポインティングデバイスを用いて、ユーザプログラムのデバッグが可能です。また、HEWを使用して、ロードモジュールファイルを高速にダウンロードできます。

(5) 充実したデバッグ機能
ブレーク、トレース機能の充実によりデバッグ効率が向上します。ブレークポイント、およびブレーク条件を専用のウィンドウで設定したり、トレース情報をウィンドウに表示できます。さらに、豊富なコマンドライン機能を備えています。

(6) エミュレーション実行中のメモリアクセス機能
エミュレーション実行中にメモリの内容を参照、変更することができます。

1.2 使用上の注意事項

注意

E6000 エミュレータをお使いになる前に、以下の注意事項を必ず確認してください。
誤った使い方は、E6000 エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

(1) 製品を梱包箱から取り出し、納入品明細書に示されているものがそろっているか、確認してください。
(2) 製品に重量物を上積みするなどして、無理な力を加えないでください。
(3) 次の条件を考慮して、E6000エミュレータを設置してください。
直射日光の当たる場所や、暖房機の近く等、高温となる場所に設置しないでください。
「1.3 使用環境条件」を参照してください。
温度や湿度が極端に変化する場所に設置しないでください。
チリやホコリが多い場所に設置しないでください。
振動が多い場所に設置しないでください。「1.3 使用環境条件」を参照してください。
(4) 製品に過大な物理的衝撃を与えないでください。
(5) E6000エミュレータに、指定された電圧、電源周波数以外の電源を供給しないでください。
(6) 設置場所を移動する場合は、本製品に強い振動、衝撃が加わらないように注意してください。
(7) ケーブルを接続した後は、接続位置が正しいことを再度確認してください。接続方法については、「2 使用前の準備」を参照してください。
すべてのケーブルを接続し終えてから、接続した各装置へ電源を投入してください。電源の投入順序は「2.6 システムチェック」を参照してください。また、電源が入っているときにケーブルの接続および取り外しをしないでください。

1.3 使用環境条件

注意

E6000 エミュレータを使用する場合、表 1-1に示す条件を守ってください。
この条件を満たさない状態で E6000 エミュレータを使用した場合、E6000 エミュレータ、ユーザプログラムおよびユーザシステムが正常に動作しない場合があります。

表1-1 使用環境条件

項番	項目	仕様
1	温度	動作時 ： 10～35℃ 非動作時 ： -10～50℃
2	湿度	動作時 ： 35～80%RH 結露なし 非動作時 ： 35～80%RH 結露なし
3	振動	動作時 ： 最大 2.45m/s^2 非動作時 ： 最大 4.9m/s^2 梱包輸送時 ： 最大 14.7m/s^2
4	周囲ガス	腐食性ガスのないこと

1.4 外形寸法と質量

表1-2 外形寸法および質量

項番	項目	仕様
1	外形寸法	219×160×54（mm）
2	質量	約 970（g）

2　使用前の準備

2.1　E6000 エミュレータ使用フローチャート

E6000 エミュレータを使用するにあたって、梱包を解いたあと下記の手順で準備を行ってください。

⚠警告

準備を行う前に図 2-1中のアミのかかっている参照先を全てよく読んで理解してください。
誤った使い方は、E6000 エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

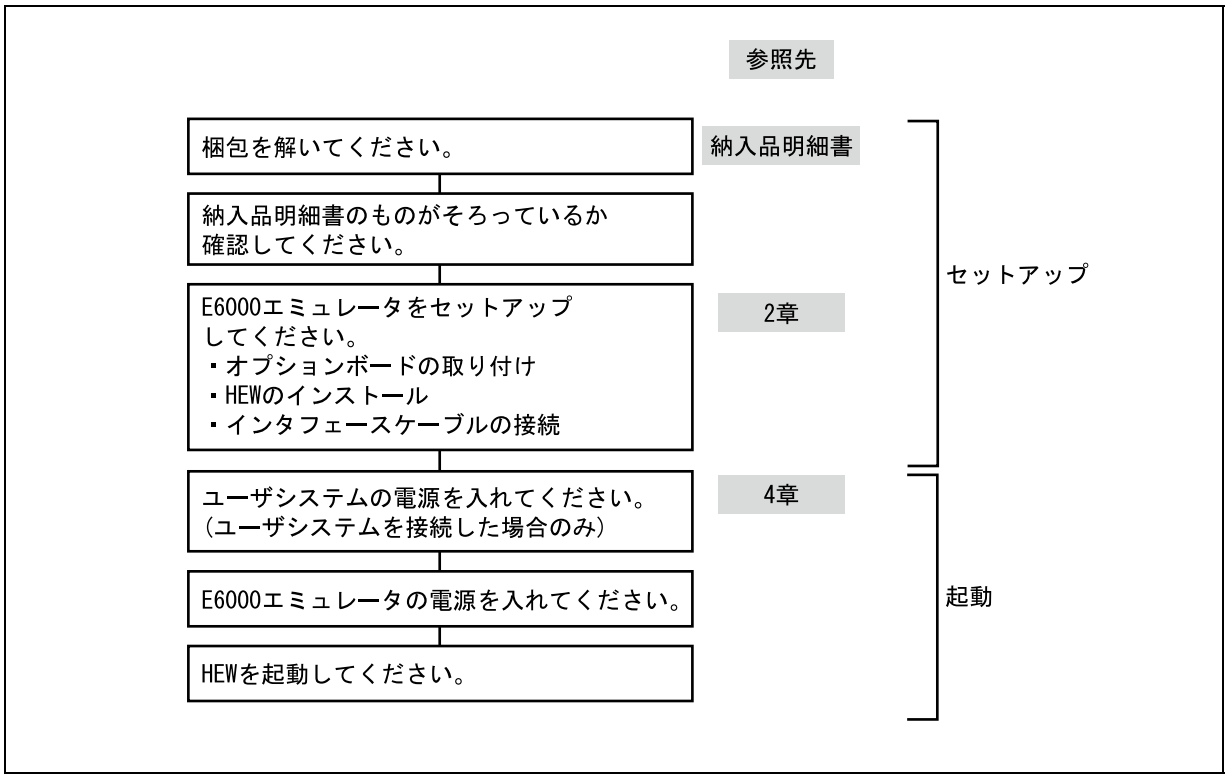


図 2-1 E6000 エミュレータ使用フローチャート

2.2　エミュレータソフトウェアのインストール

エミュレータソフトウェアのインストールにつきましては、同梱の E6000 エミュレータ用セットアップガイドをご参照ください。

2.3 ユーザシステムへの接続

E6000 エミュレータをユーザシステムへ接続するには、以下の手順に従ってください。

- ユーザシステムインタフェースケーブル先端部をユーザシステムへ接続します。
- ユーザシステムインタフェースケーブルのケーブル本体部を E6000 エミュレータへ接続します。
- ケーブル本体部を先端部へ接続する。

これらの手順詳細については、ユーザシステムインタフェースケーブル添付の取扱い説明書を参照してください。

以下に、E6000 エミュレータのコネクタを示します。

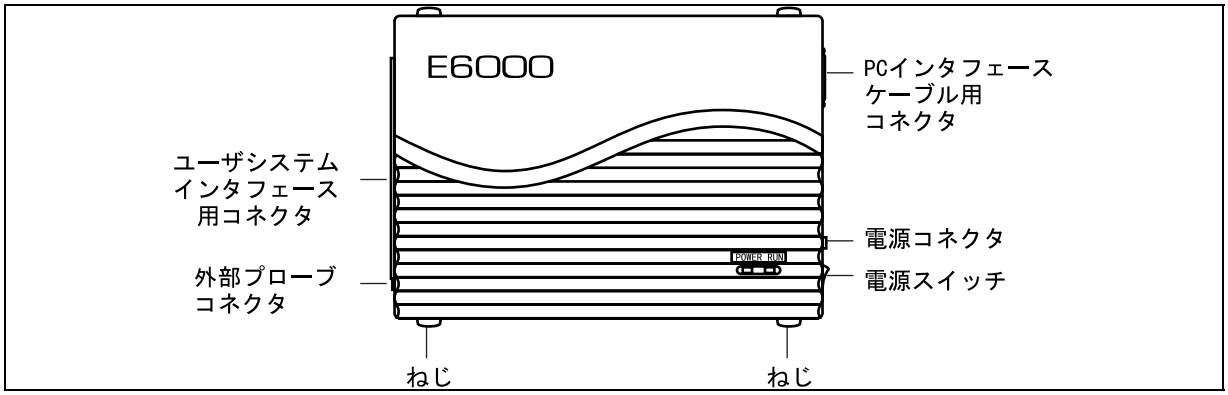


図 2-2 E6000 **コネクタの位置**

2.3.1 ユーザシステムインタフェースケーブル先端部とユーザシステムの接続例

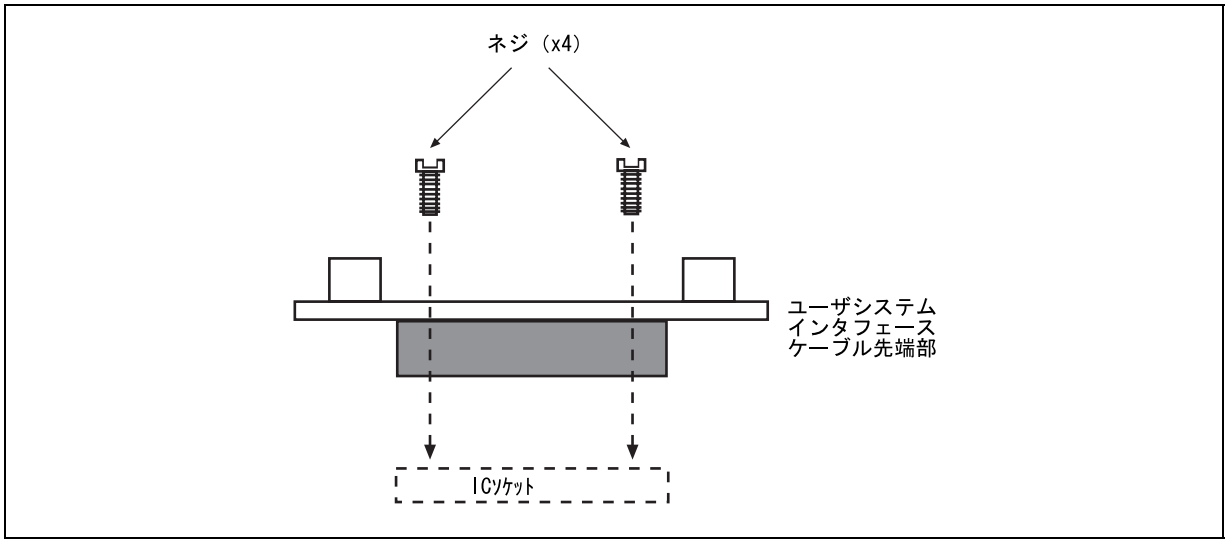


図 2-3 **ユーザシステムインタフェースケーブルの接続**

- 接続前に必ず、E6000 エミュレータ、ユーザシステムの電源を切ってください。
- ユーザシステムインタフェースケーブル先端部をユーザシステム上のソケットに挿入してください。

【注】 パッケージによっては、ユーザシステムインタフェースケーブル先端部の向きにかかわらず、ソケットに差し込むことができるものがあります。挿入の際には、E6000 **エミュレータ側とソケットの１ピンの位置を正しく一致させてください。**

- ユーザシステムインタフェースケーブルに付属のネジを使用して、ユーザシステムインタフェースケーブル先端部とソケットをネジ留めしてください。以下に示す順番で、対角に少しずつ締めつけ、１つのネジを集中して締めつけることは絶対に避けてください。

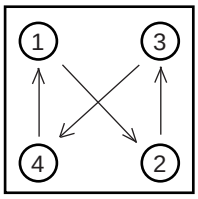


図 2-4 **ネジの締めつけ順序**

【注】 ネジを締めつけ過ぎないように注意してください。ユーザシステムの接続不良やユーザシステムインタフェースケーブル先端部が壊れる原因となります。QFP **ソケットに半田付け用固定金具が付いている場合は、これを使用して、**E6000 **エミュレータとユーザシステムの接続を強めることができます。**

2.3.2 ユーザシステムインタフェースケーブル本体部と E6000 エミュレータの接続

ユーザシステムインタフェースケーブル本体部ケーブルを E6000 エミュレータに接続してください。ケーブルは、まっすぐに、確実に接続されるまで押し込んでください。

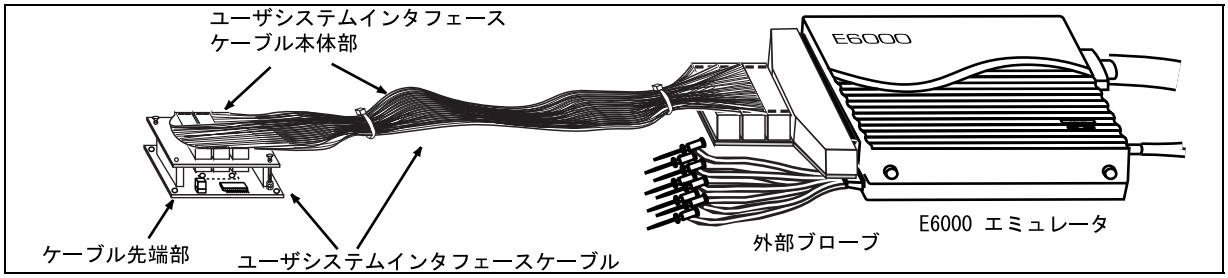


図 2-5 **ユーザシステムインタフェースケーブル外観図**

2.3.3 ユーザシステムインタフェースケーブル本体部と先端部の接続

ユーザシステムインタフェースケーブル本体部をユーザシステムに接続されている先端部に接続してください。

2.4 電源供給

2.4.1 AC 電源アダプタ

E6000 エミュレータに付属の AC 電源アダプタを常に使用してください。

2.4.2 極性

以下に電源プラグの極性を示します。

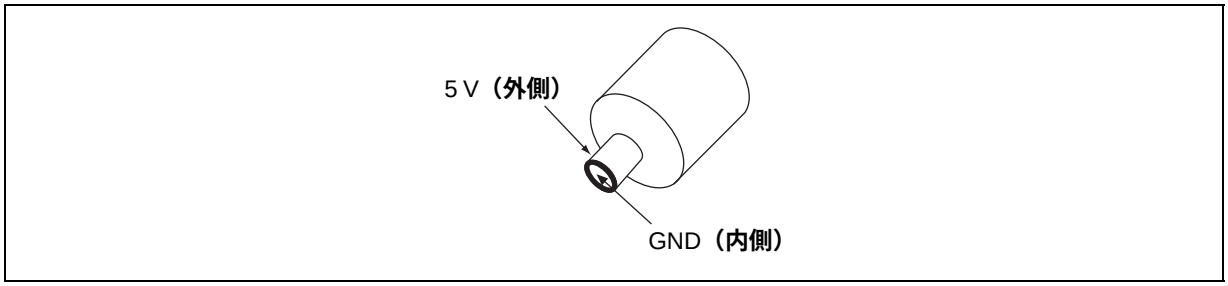


図 2-6 電源プラグ

2.4.3 電源モニタ回路

E6000 エミュレータには、ユーザシステムの電源モニタ回路があり、4.75V 以上の電源が供給されているとパワーLED が赤く点灯します。パワーLED が消えている場合は、E6000 エミュレータの電源レベルをチェックしてください。電源が 4.75V 未満の場合、E6000 エミュレータに必要な電流が供給されません。

【注】 必ず E6000 エミュレータに付属の AC 電源アダプタを使用してください。

2.5 ハードウェアインタフェース

E6000 エミュレータのユーザシステムインタフェース信号は、一部の信号を除いてバッファなしに直接エミュレータ上の MCU に接続されています。MCU に入力する前にエミュレータ制御回路が挿入される信号は各エミュレータにより異なります。詳細は「7 本製品固有のハードウェア仕様」を参照してください。

2.5.1 クロック発振器

ユーザシステムインタフェースケーブル先端部にはシステムクロック発振回路を実装しています。システムクロック発振回路の詳細については、各ユーザシステムインタフェースケーブル取扱い説明書を参照してください。

2.5.2 外部プローブ／トリガ出力

E6000 エミュレータ筐体側面にある EXT のマークが記された 8 ピンコネクタ（ユーザインタフェースコネクタの右下）は、外部プローブ入力 4 本とトリガ出力 2 本を備えています。以下にこのコネクタのピン配置を示します。

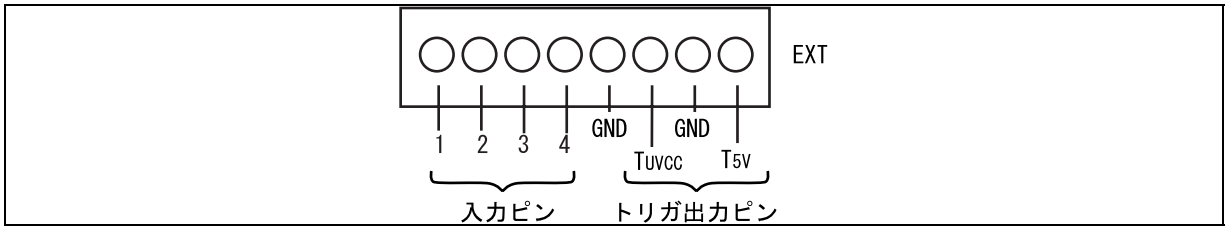


図 2-7 外部プローブコネクタ

以下に外部プローブのインタフェース回路を示します。

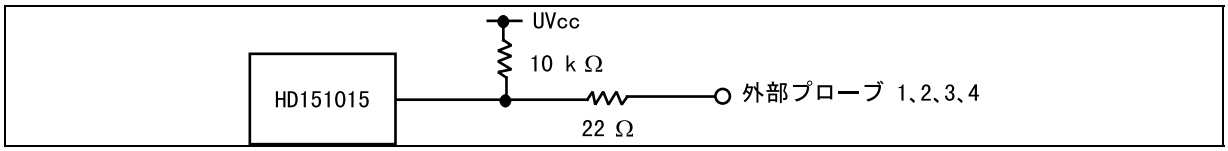


図 2-8 外部プローブインタフェース回路

トリガ出力はイベントチャネル 8 によって出力されるアクティブロー信号です。トリガ出力は T5V(プローブ色：白色、2.5V～5.0V の範囲でユーザシステムの電圧レベルに依存しません。)または TUvcc（プローブ色：黄色、ユーザシステム電源電圧）レベルの２つあります。

2.5.3 電源フォロワ回路

注意

1. **ユーザシステムを E6000 エミュレータに接続しない時は、ユーザシステムインタフェースケーブルをエミュレータに接続しないでください。**
2. **ユーザシステムを E6000 エミュレータに接続した時は、ユーザシステムの電源を入れてからエミュレータを起動してください。**

E6000 エミュレータに搭載されている電圧フォロワ回路は、ユーザシステムの電圧レベルをモニタしています。E6000 エミュレータの電源は、ユーザシステムの電源レベルを生成し、E6000 エミュレータ内に供給しているため MCU 電源がユーザシステムから供給されることはありません。

E6000 エミュレータにユーザシステムインタフェースケーブルが接続されていないと、E6000 エミュレータ上の MCU は一定電圧レベルで動作し、ユーザシステムインタフェースケーブルが接続されている場合は、ユーザシステムの電源電圧と同レベルの電圧で動作します。ユーザシステム Vcc が MCU の動作電圧よりも低い場合であっても、E6000 エミュレータは供給電圧をユーザシステム Vcc に一致させます。エミュレーションクロックの周波数が各 Vcc における最高動作周波数を超えないように注意してください。

Configuration ダイアログボックスを使用して、［User VCC Threshold］を Vcc max から 0V の範囲で設定できます。ユーザ Vcc がその値よりも下がった場合、[拡張モニタ]ウインドウの［User System Voltage］には［Down］が表示されます。User VCC Threshold 電源レベルよりも高い場合は、［OK］が表示されます。

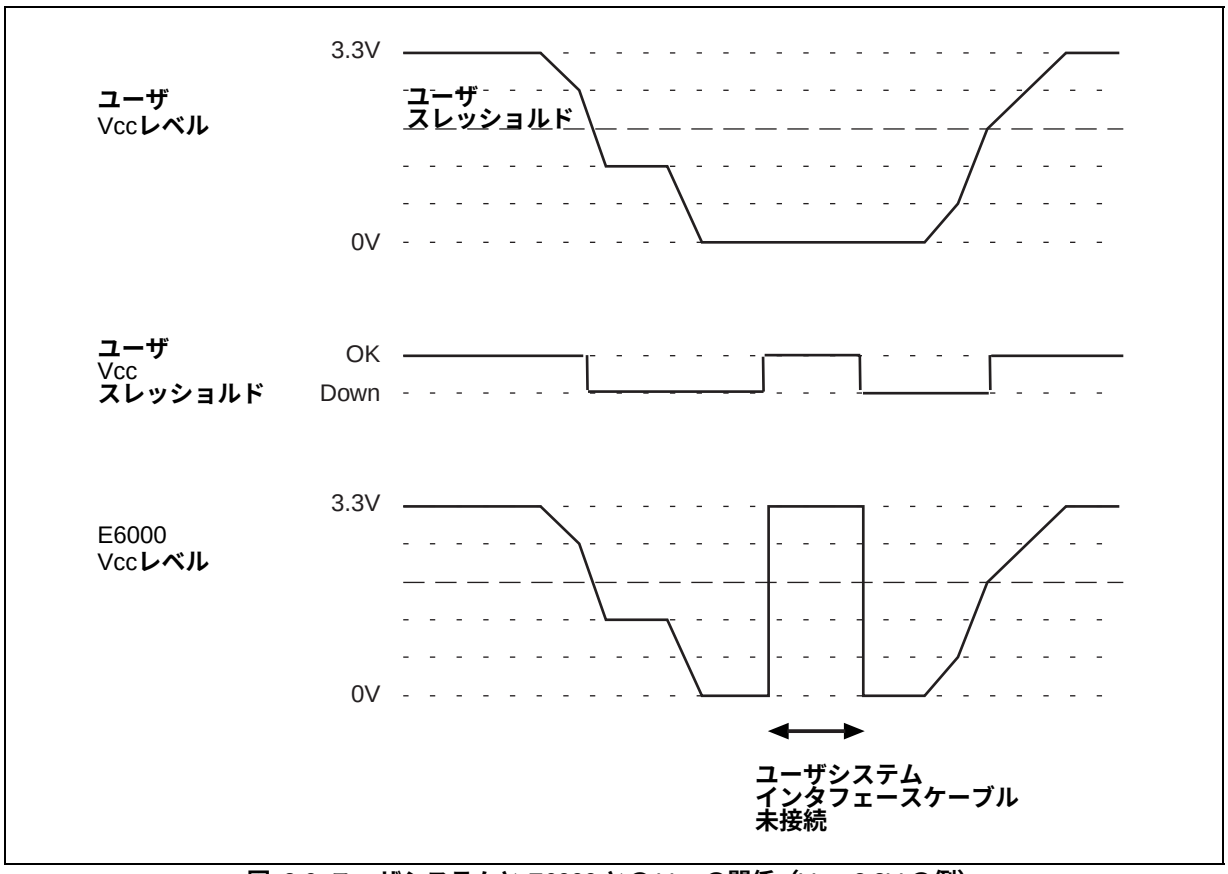


図 2-9 ユーザシステムと E6000 との Vcc の関係（Vcc=3.3V の例）

2.6　システムチェック

次に、ソフトウェアを実行し、E6000 エミュレータが正しく接続されていることをチェックします。
ここでは、製品に添付のチュートリアル用ワークスペースを使用して起動します。
新規にプロジェクトを作成して起動する方法や､旧バージョンの HEW 用のワークスペースを使用して起動する方法については、「2.8　その他の起動方法」を参照してください。

(1)　ホストコンピュータとE6000エミュレータを接続してください。
(2)　E6000エミュレータのコネクタとユーザインタフェースケーブルを接続します。
(3)　E6000エミュレータの電源を入れてください。
(4)　[スタート]メニューの[プログラム]からHEWを起動してください。（図 2-10）

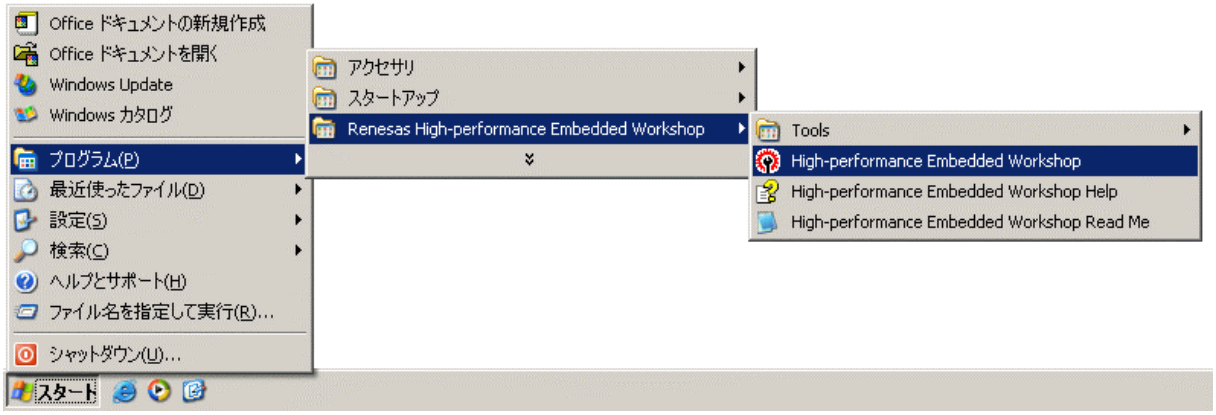


図 2-10　[スタート]メニュー

【留意事項】
[Tools] **は、インストール時、**”LAN Driver”**を選択しなかった場合は表示されません。**

(5) [ようこそ!]ダイアログボックスが表示されます。

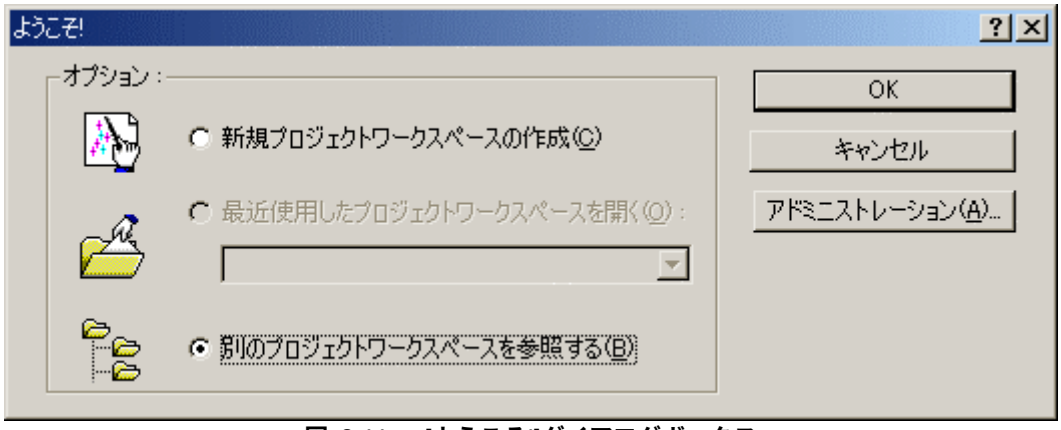


図 2-11 [ようこそ!]ダイアログボックス

ここでは、チュートリアル用ワークスペースを使用するため、[別のプロジェクトワークスペースを参照する]ラジオボタンを選択し、[OK]ボタンを押してください。

[ワークスペースを開く]ダイアログボックスが開きますので、以下のディレクトリを指定してください。
HEWインストール先ディレクトリ\Tools\Renesas\DebugComp\Platform\E6000\xxxx\Tutorial

ディレクトリの指定後、以下のファイルを選択し[Open]ボタンを押してください。

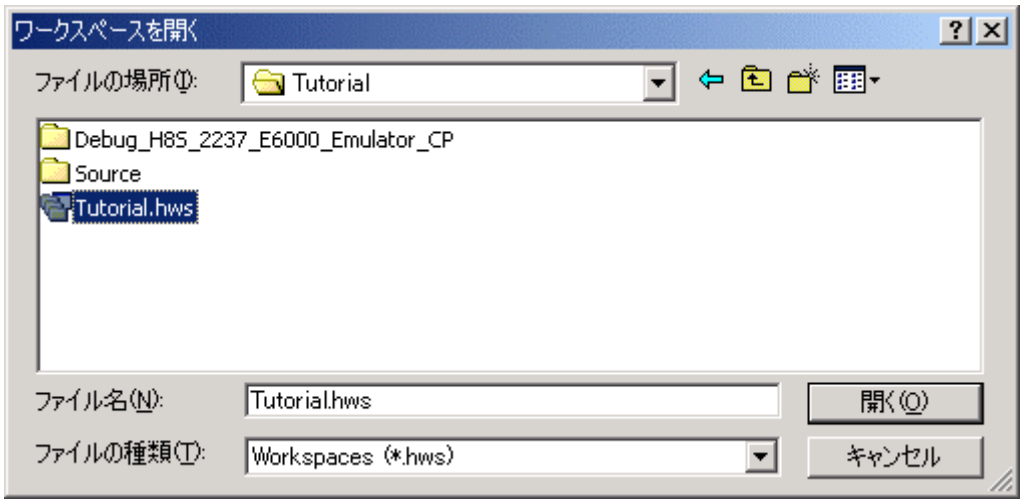


図 2-12　[ワークスペースを開く]ダイアログボックス

コンパイラパッケージがインストールされていない場合、もしくはバージョンの異なるコンパイラパッケージがインストールされている場合は以下のメッセージボックスが表示されます。

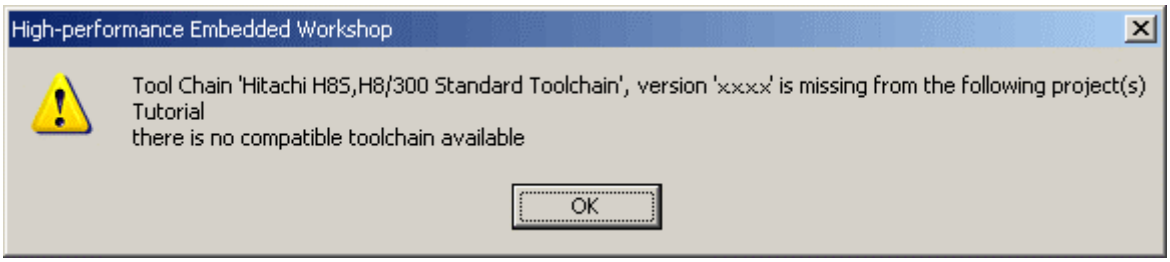


図 2-13　メッセージボックス

(6) [E6000 Driver Details]ダイアログボックスが表示されます。
本ダイアログボックスは、2度目の起動からは表示されません。
また、エミュレータソフトのインストール時にインタフェースドライバを1つのみ選択した場合にも表示されません。

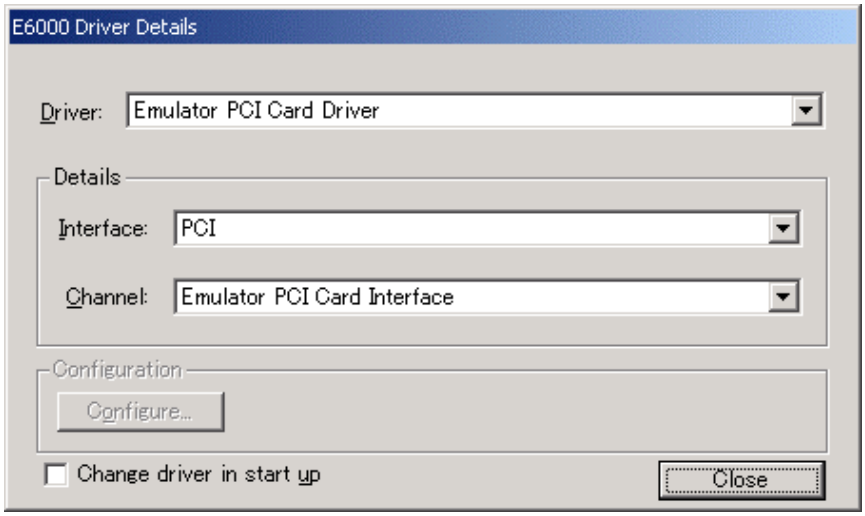


図 2-14 ［E6000 Driver Details］ダイアログボックス

- ［Driver］コンボボックスで、E6000 エミュレータを接続するドライバの選択を行います。
- ［Interface］には、接続するインタフェース名が表示されます。
- ［Close］ボタンをクリックしてください。

(7) E6000エミュレータのセットアップを行います。
セットアップ中は、以下のダイアログボックスが表示されます。

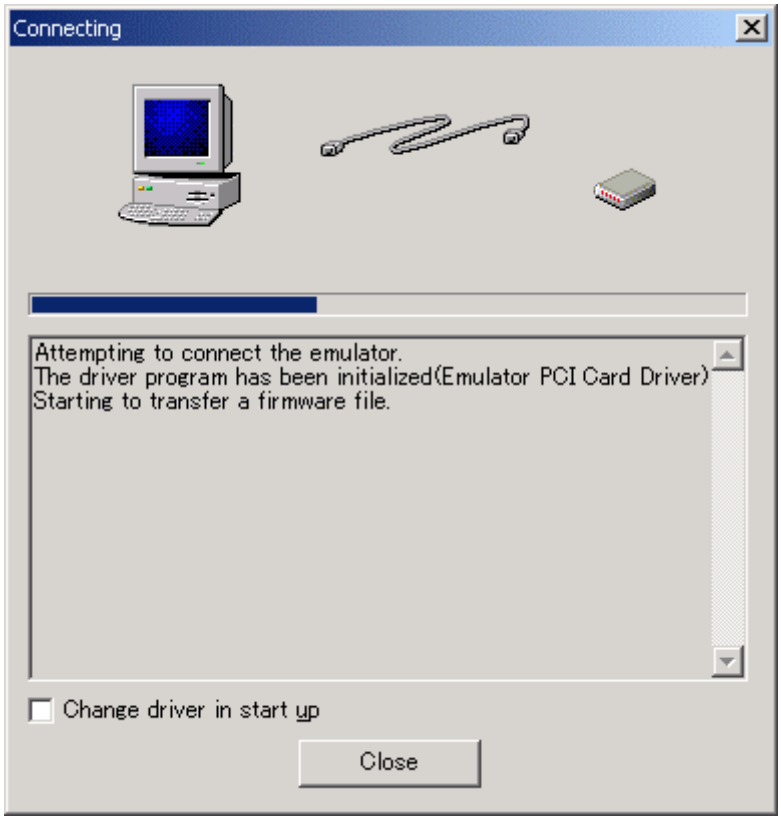


図 2-15 ［Connecting］ダイアログボックス

(8)　HEWの[Output]ウインドウに”Connected”と表示されたら、E6000エミュレータの起動は完了です。

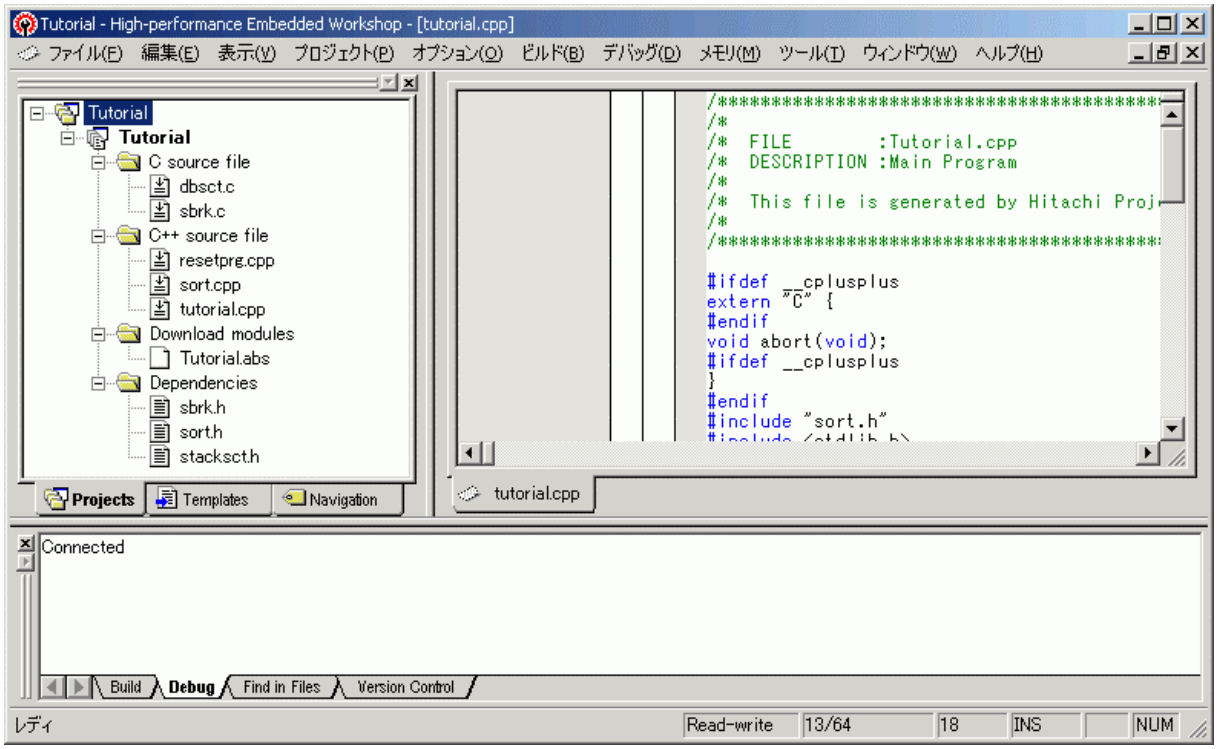


図 2-16　HEW **画面**

2.7　通信不良

E6000 エミュレータの電源が OFF の場合、または PC インタフェースケーブルが正しく接続されていない場合は以下のエラーメッセージが表示されます。

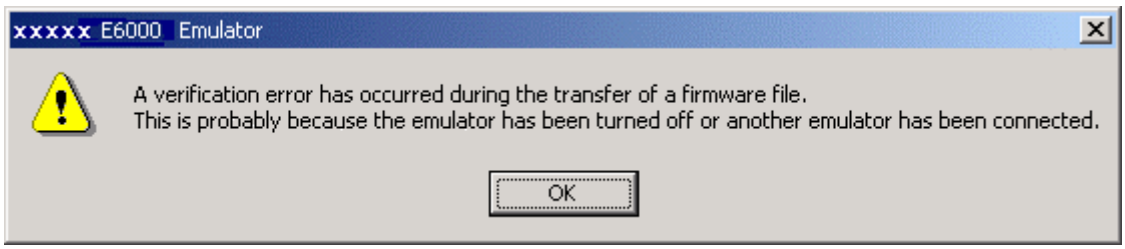


図 2-17　エラーメッセージ

その他のエラーについては、「E6000 エミュレータ用セットアップガイド」を参照してください。

2.8　その他の起動方法

その他の起動方法は、「4　デバッグの準備をする」を参照してください。

2.9　アンインストール

アンインストール方法は、「E6000 エミュレータ用セットアップガイド」を参照してください。

3 E6000 エミュレータ機能

3.1 デバッグの特長

3.1.1 ブレークポイント

E6000 エミュレータは、強力なハードウェアブレークおよびプログラムブレークを備えているので、ソフトウェアとユーザシステムのデバッグを効率よく実行できます。

ハードウェアブレークポイント

イベント検出システムのイベントチャネルと範囲チャネルを使って、最大 12 箇所のブレークポイントが設定できます。ハードウェアブレークポイントに関しては、「3.2 イベント検出システム（CES: Complex Event System）」を参照してください。

プログラムブレークポイント（PC ブレークポイント）

最大 256 箇所のプログラムブレークポイントが設定できます。プログラムブレークポイントは、ユーザ命令を BREAK 命令で置き換えることによって設定されるので、ユーザシステム上の ROM には 1 箇所のみ（On chip Break）設定できます。

3.1.2 トレース

E6000 エミュレータは、強力なリアルタイムトレース機能を備えていますので、MCU の動作を詳細に調べることができます。リアルタイムトレースバッファは、32768 までのバスサイクルを保持でき、実行中は常に更新されます。バッファはローリングバッファとして構成され、エミュレーションを中断することなく、トレースを中断しトレース内容を表示することができます。

トレースバッファ内の取得データは、デバッグを容易にするためにソースプログラムおよびアセンブリ言語の両方で表示されます。ただし、トレースフィルタリングが行われた場合は、アセンブリ言語だけが表示されます。

トレースバッファは、すべてのバスサイクルあるいは選択されたサイクルだけを記憶するように制御されます。イベント検出システムを使用して所望のトレース制御を選択します。

すべてのバスサイクルを記憶しておいて、選択されたサイクルだけを見ることも可能です。これをトレースフィルタリングといいます。

3.1.3 実行時間測定

E6000 エミュレータによって、総実行時間の測定、またはイベント検出システムで指定されたイベント間の実行時間の測定ができます。タイマの分解能は以下のいずれかの値に設定できます。

20ns, 125ns, 250ns, 500ns, 1µs, 2µs, 4µs, 8µs, 16µs

測定可能な最大時間は、分解能 20ns で約 6 時間、分解能 16µs で約 200 日です。

3.1.4 バスモニタ

E6000 エミュレータは、バスモニタ機能を備えています。プログラムの実行を中断することなくアクセスのあった領域の値をモニタし、ウィンドウ上に表示することができます。モニタ指定できる領域は 256 バイトで最大 8 ブロックです。また、指定したアドレス（4 箇所まで）に対するアクセスにより、トリガ信号を外部プローブ 2（EXT2）出力します。

製品によってはバスモニタ機能をサポートしていない場合があります。

3.2 イベント検出システム（CES: Complex Event System）

実際のデバッグの大部分において、デバッグしようとするプログラムの不具合またはハードウェアの不具合は、限定された状況においてのみ、発生します。たとえば、あるハードウェアエラーは、メモリの特定の領域がアクセスされた時のみ発生します。簡単なプログラムブレークポイントを使用してその問題を調べ上げるのは、非常に困難です。

E6000 エミュレータは、調べたい条件を正確に記述できるシステム（イベント検出システム）を備えています。これによって、MCU 信号の指定された組み合わせのイベントを定義できます。

イベント検出システムは、E6000 エミュレータのトレース、ブレーク、およびイベント間実行時間測定機能を制御します。

3.2.1 イベントチャネル

イベントチャネルによって、指定されたイベントの発生を検出できます。イベントは以下の項目の組み合わせで定義できます。

- アドレスまたはアドレス範囲
- アドレス範囲外
- リード、ライトまたは両方とも
- マスク条件指定付きデータ
- MCU アクセスタイプ（DMAC、命令プリフェッチなど）
- MCU アクセス領域（内蔵 ROM、内蔵 RAM など）
- 4 つの外部プローブ信号の値
- イベントの発生回数
- イベントの発生後のディレイサイクル数

また、最大 8 イベントがシーケンスで組み合わせできます。それぞれのイベントは、シーケンスにおける前のイベントの発生によって起動、あるいは停止します。たとえば、内蔵 RAM の指定された領域がアクセスされた後で I/O レジスタが書き込まれたときというブレーク条件を設定できます。

3.2.2 範囲チャネル

範囲チャネルは、以下の項目の組み合わせで定義できます。

- アドレスまたはアドレス範囲
- リード、ライトまたは両方とも
- マスク条件指定付きデータ
- MCU アクセスタイプ（DMAC、命令プリフェッチなど）
- MCU アクセス領域（内蔵 ROM、内蔵 RAM など）
- 4 つの外部プローブ信号の値
- イベントの発生後のディレイサイクル数

イベント検出システムは、E6000 エミュレータの以下の機能を制御するのに使われます。

3.2.3 ブレーク

指定されたイベントまたはイベントのシーケンスが発生したときに、プログラム実行を停止します。たとえば、プログラムがあるアドレスからデータ読み出し後、あるアドレスにデータを書き込んだときに実行を停止するように、ブレークを設定できます。また、ブレークは 65535 バスサイクルまで任意に遅らせることができます。

3.2.4 イベント間実行時間測定

2 つのイベントを設定し、最初のイベントの発生と 2 番目のイベントの発生間のプログラムの実行時間を測定できます。

3.3　ハードウェアの特長

3.3.1　メモリ

E6000 エミュレータは、エミュレーションメモリとして内蔵 ROM/内蔵 RAM 用代替メモリを標準装備しています。

エミュレーションメモリは、MCU アドレス空間の任意のサイズのメモリブロックに割り付けできます。各メモリブロックは、メモリマップ機能を使用して、ユーザシステム上のメモリに指定でき、それぞれの場合で、リードライトアクセス、リードオンリアクセス、またはアクセス禁止を指定できます。H8/300L シリーズの MCU では、内蔵 ROM、内蔵 RAM の指定しかできません。

エミュレーションメモリの各々のメモリタイプの定義を以下に示します。

表3-1　メモリタイプの定義

メモリタイプ	説明
オンチップ	MCU 内蔵メモリ
ユーザ	ユーザシステム上のメモリ（H8/300 シリーズのみ）
エミュレータ	エミュレーションボード上のメモリ（H8/300 シリーズのみ）

メモリアドレスの指定されたブロックの内容は、メモリ機能を使って表示されます。メモリの内容はいつでも（プログラム実行中であっても）変更でき、その結果は、他の関連するウインドウにすぐに反映されます。

3.3.2　エミュレーションクロック

エミュレーションクロックは E6000 エミュレータ内蔵クロックとターゲットクロックのいずれかの周波数に設定できます。選択できるクロックは製品により異なります。詳しくは、「8　本製品固有のソフトウェア仕様」を参照してください。

3.3.3　外部プローブ

ユーザシステム上の任意の信号をブレークもしくはトレースに使用するために、E6000 エミュレータには外部プローブが接続できます。外部プローブの信号はローまたはハイレベルに応じて、イベント検出システムの条件として設定できます。

3.4　スタックトレース機能

E6000 エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。本機能は、Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

本機能の使用方法については、「6.18　スタックトレース機能」を参照してください。

3.5　オンラインヘルプ

各機能の操作方法や、コマンドラインウィンドウから入力できるコマンドのシンタックスを記載している、オンラインヘルプ機能があります。

エミュレータ用機能のヘルプを見る場合、[ヘルプ]メニュー→[エミュレータヘルプ]を選択してください。

4　デバッグの準備をする

4.1　ワークスペース、プロジェクト、ファイル

ワードプロセッサでドキュメントを作成、修正できるのと同じように、HEW ではワークスペースを作成、修正できます。

ワークスペースはプロジェクトを入れる箱と考えることができます。同じように、プロジェクトはプロジェクトファイルを入れる箱と考えることができます。したがって各ワークスペースにはプロジェクトが一つ以上あり、各プロジェクトにはファイルが一つ以上あります。この構成を図 4-1に示します。

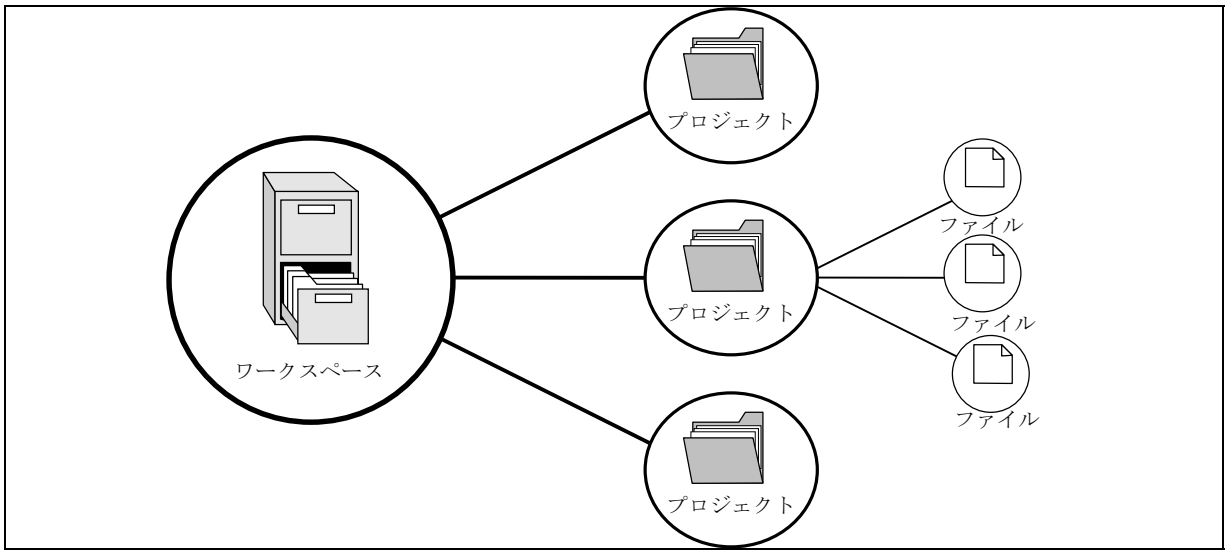


図 4-1　ワークスペース、プロジェクト、ファイルの構成

ワークスペースでは関連したプロジェクトを一つにまとめることができます。例えば、異なるプロセッサに対して 1 つのアプリケーションを構築しなければならない場合、または、アプリケーションとライブラリを同時に開発している場合などに便利です。さらに、ワークスペース内でプロジェクトを階層的に関連づけることができます。つまり、1 つのプロジェクトを構築すると、その子プロジェクトを最初に構築します。

ワークスペースを活用するには、ユーザは、まずワークスペースにプロジェクトを追加して、そのプロジェクトにファイルを追加しなければなりません。

4.2 HEW の起動方法

HEW は以下の手順で起動します。

(1) ホストコンピュータとE6000エミュレータを接続してください。

(2) ユーザシステムインタフェースケーブルをご使用の場合は、E6000エミュレータのコネクタとユーザシステムインタフェースケーブルを接続します。ユーザシステムインタフェースケーブル未使用の場合は本手順は不要です。
E6000エミュレータの電源を入れてください。ユーザシステムをご使用の場合は、E6000エミュレータの電源を入れる前に、ユーザシステムの電源を入れてください。

(3) [スタート]メニューの[プログラム]からHEWを起動してください。

(4) [ようこそ!]ダイアログボックスが表示されます。

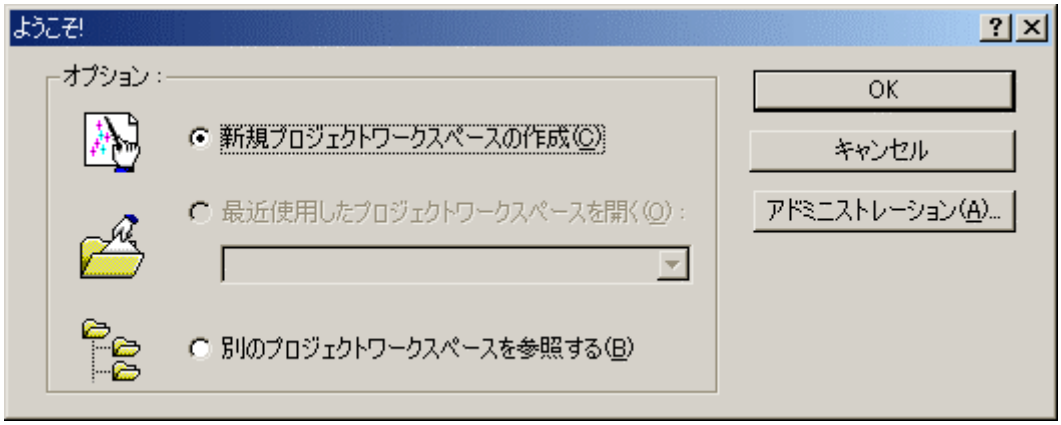


図 4-2 [ようこそ!]ダイアログボックス

・[新規プロジェクトワークスペースの作成]ラジオボタン
ワークスペースを新規作成する場合に選択します。
・[最近使用したプロジェクトワークスペースを開く]ラジオボタン
既存のワークスペースを使用する場合に選択します。
開いたワークスペースの履歴が表示されます。
・[別のプロジェクトワークスペースを参照する]ラジオボタン
既存のワークスペースを使用する場合に選択します。
開いた履歴が残っていない場合に使用します。

ここでは、下記の３つの方法を説明します。

・[新規プロジェクトワークスペースの作成] – ツールチェインを使用しない場合
・[新規プロジェクトワークスペースの作成] – ツールチェインを使用する場合
・[別のプロジェクトワークスペースを参照する]

ツールチェインを使用する場合と使用しない場合では新規プロジェクトワークスペースの作成手順が異なります。本製品には、ツールチェインは含まれていません。ツールチェインは H8S, H8/300 シリーズ C/C++コンパイラパッケージがインストールされている環境にて使用することができます。

ツールチェインを使用した新規プロジェクトワークスペースの作成についての詳細は、H8S, H8/300 シリーズ C/C++コンパイラパッケージ付属のマニュアルを参照してください。

4.2.1　新規にワークスペースを作成する場合(ツールチェイン未使用)

(1)HEW 起動時に表示される、[ようこそ!]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンを押してください。

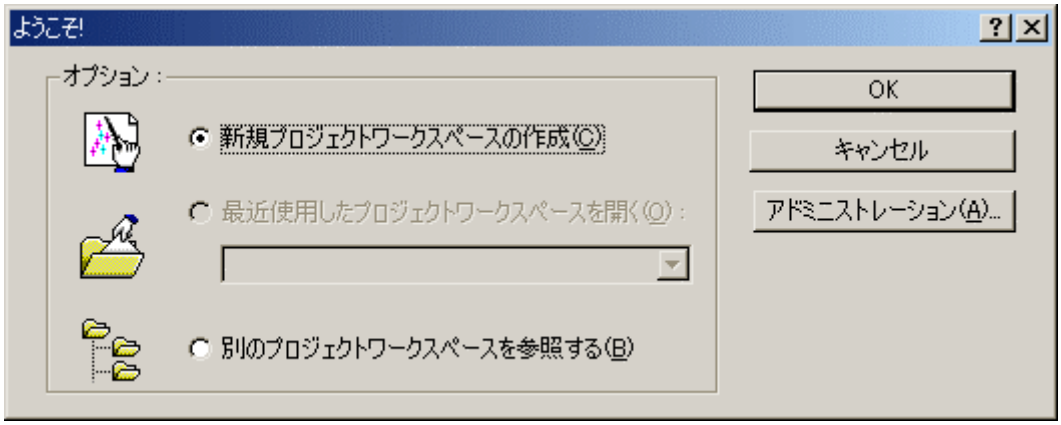


図 4-3　[ようこそ!]ダイアログボックス

(2)新規プロジェクトワークスペースの作成を開始します。
以下の画面が開きます。

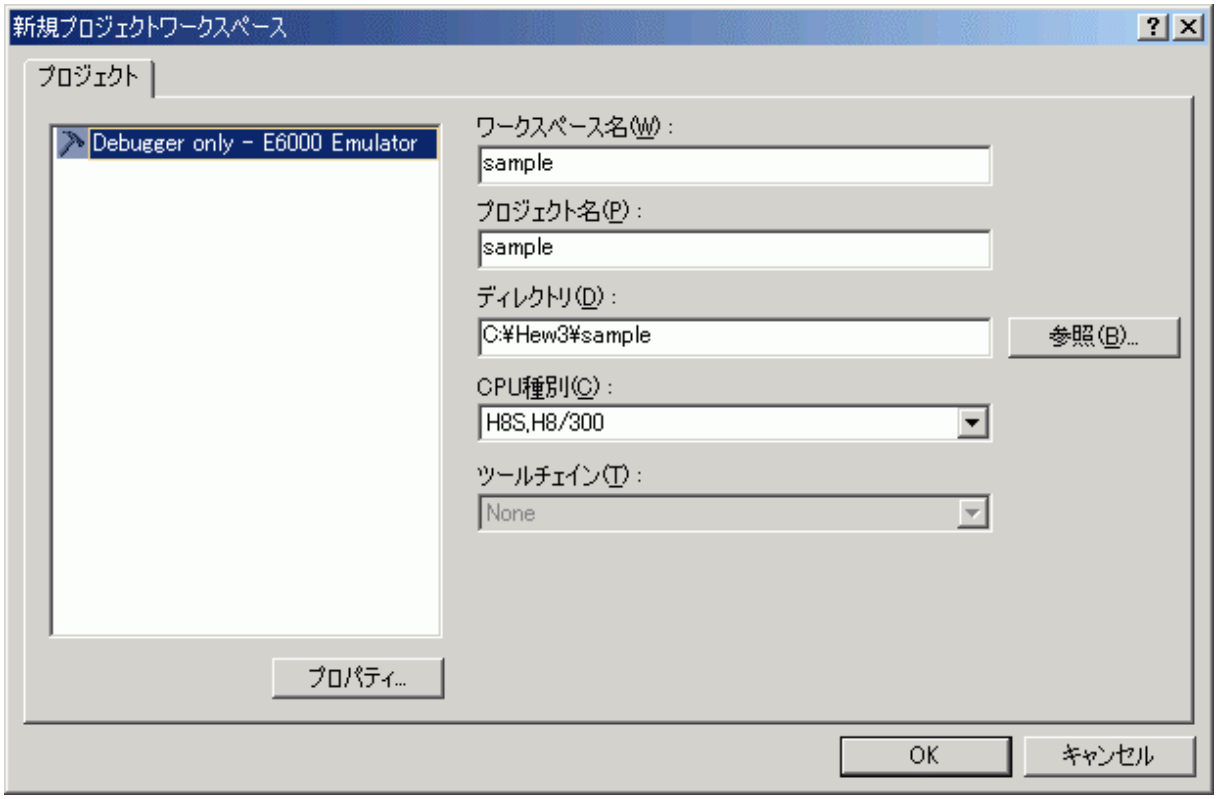


図 4-4　[新規プロジェクトワークスペース]ダイアログボックス

・[ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。

・[プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じでよろしければ、入力する必要はありません。

・[ディレクトリ]エディットボックス
ワークスペースが作成されるディレクトリが表示されます。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選んだり、[ディレクトリ]エディットボックスに、ワークスペースを作成するディレクトリを手入力することができます。

その他のリストボックスはツールチェイン設定用ですので、ツールチェインをインストールしていない場合は固定情報が表示されます。

(3)次に、ツールチェインの設定を行いますので、必要な設定を行ってください。
ツールチェインの設定が終了したら、以下の画面が表示されます。

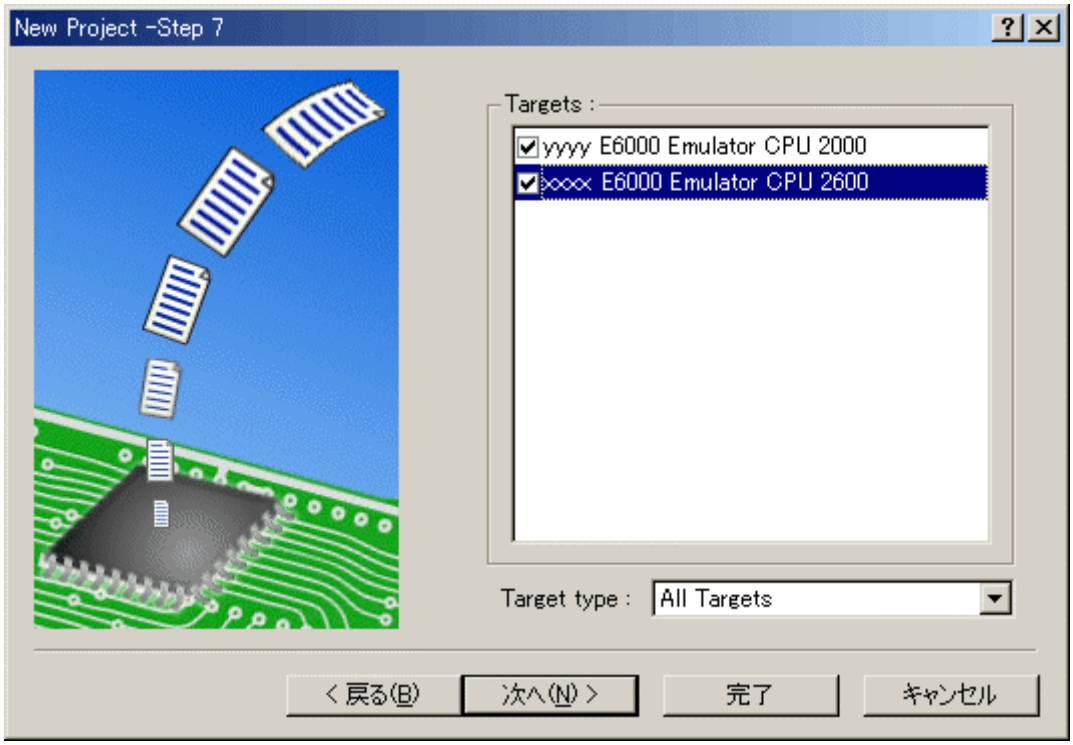


図 4-5　[New Project-Step 7]ダイアログボックス

ここでは、HEW 起動時に使用するセッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンを押してください。セッションファイルについての詳細は「4.4　デバッグセッション」を参照してください。

(4)次に、コンフィグレーションファイル名を設定します。
コンフィグレーションとは、エミュレータ以外の HEW の状態を保存するファイルです。

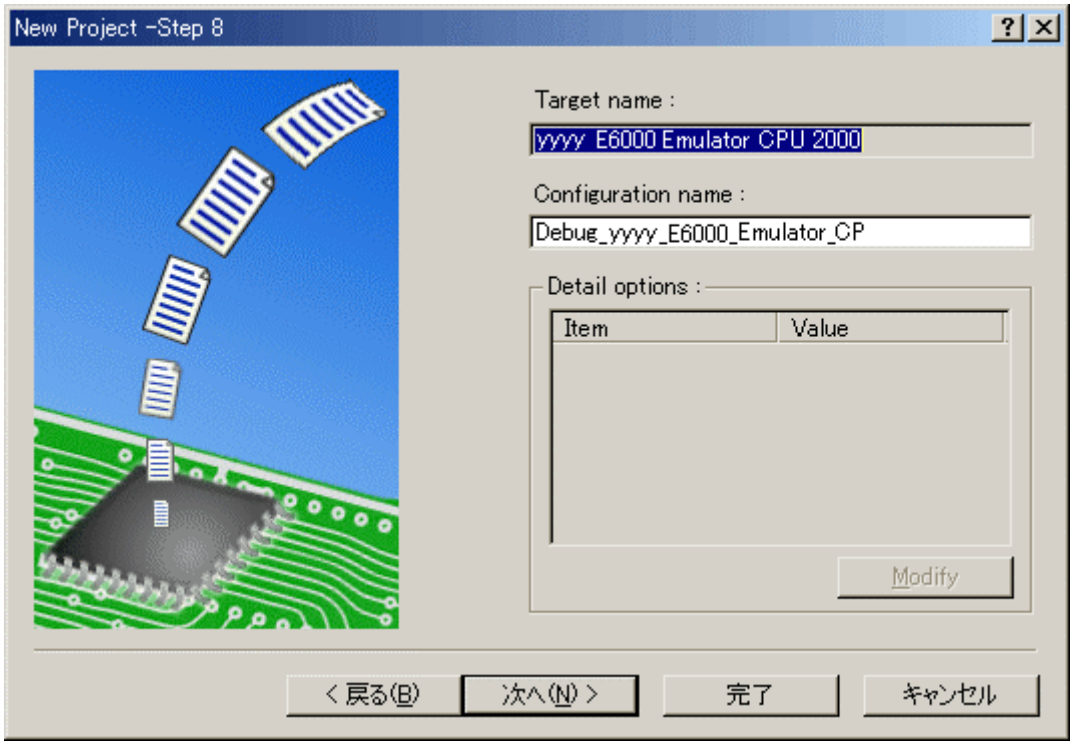


図 4-6　[New Project-Step 8]**ダイアログボックス**

[New Project-Step 7]ダイアログボックス（図 4-5）で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンを押し、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了しましたら、E6000 エミュレータに関する設定は終了です。
[完了]ボタンを押すと[Summary]ダイアログボックスが表示されます。[OK]ボタンを押すと、HEW が起動します。

(5)HEW 起動後、自動的に E6000 エミュレータが接続されます。
接続が完了すると、[Output]ウィンドウの[Debug]タブに「Connected」と表示されます。

4.2.2　新規にワークスペースを作成する場合(ツールチェイン使用)

(1)HEW 起動時に表示される、[ようこそ!]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンを押してください。

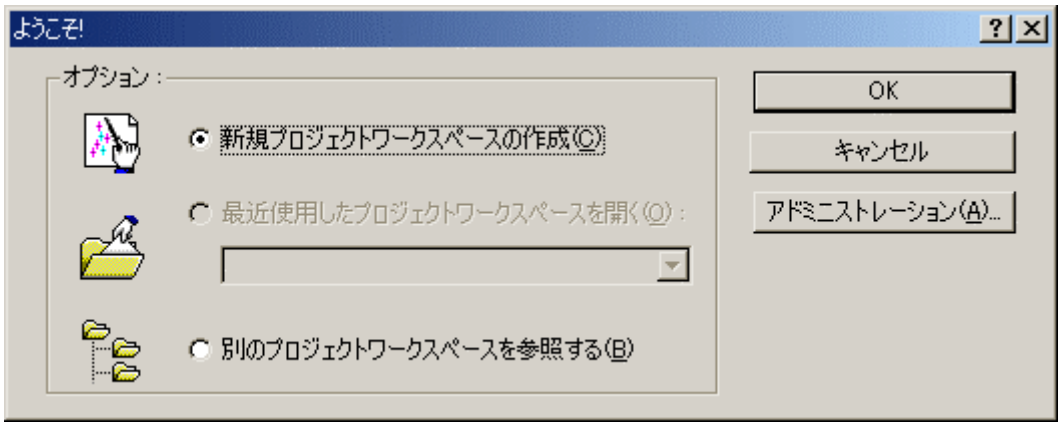


図 4-7　[**ようこそ!**]**ダイアログボックス**

(2)新規プロジェクトワークスペースを作成します。
以下の画面が開きます。

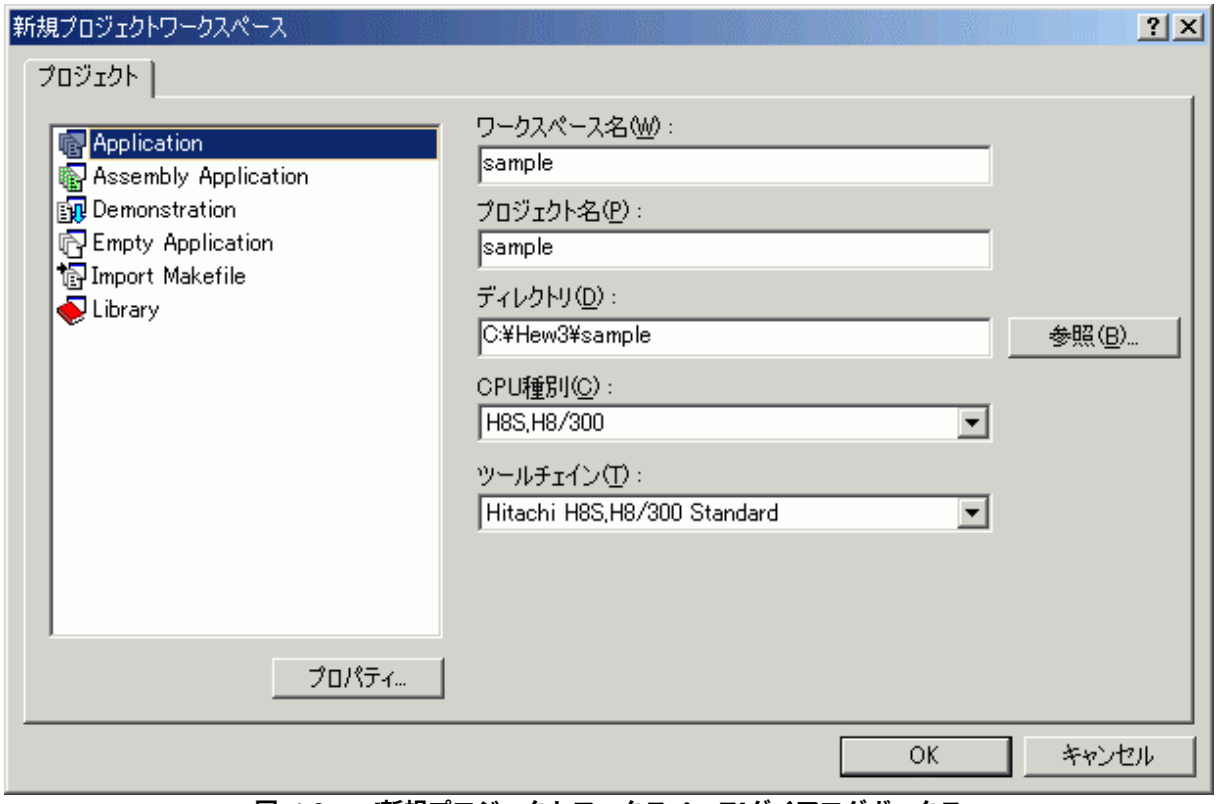


図 4-8　[新規プロジェクトワークスペース]ダイアログボックス

・[ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。

・[プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じでよろしければ、入力する必要はありません。

・[ディレクトリ]エディットボックス
ワークスペースが作成されるディレクトリが表示されます。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選んだり、[ディレクトリ]エディットボックスに、ワークスペースを作成するディレクトリを手入力することができます。

・[CPU 種別]ドロップダウンリストボックス
該当する CPU ファミリを選択してください。

・[ツールチェイン]ドロップダウンリストボックス
ツールチェインをご使用になる場合、該当するツールチェイン名を選択してください。
使用しない場合、[None]を選択してください。

・[プロジェクトタイプ]リストボックス
使用したいプロジェクトタイプを選択してください。

【留意事項】

1. E6000 エミュレータの場合、[Demonstration]を選択した場合に以下の注意事項があります。
   [Demonstration]は H8S, H8/300 コンパイラパッケージ付属のシミュレータ用のプログラムです。生成されたソースファイルを使用する場合、ソースファイル中の“Printf 文”を削除してください。

(3)次に、ツールチェインの設定を行いますので、必要な設定を行ってください。
ツールチェインの設定が終了したら、以下の画面が表示されます。

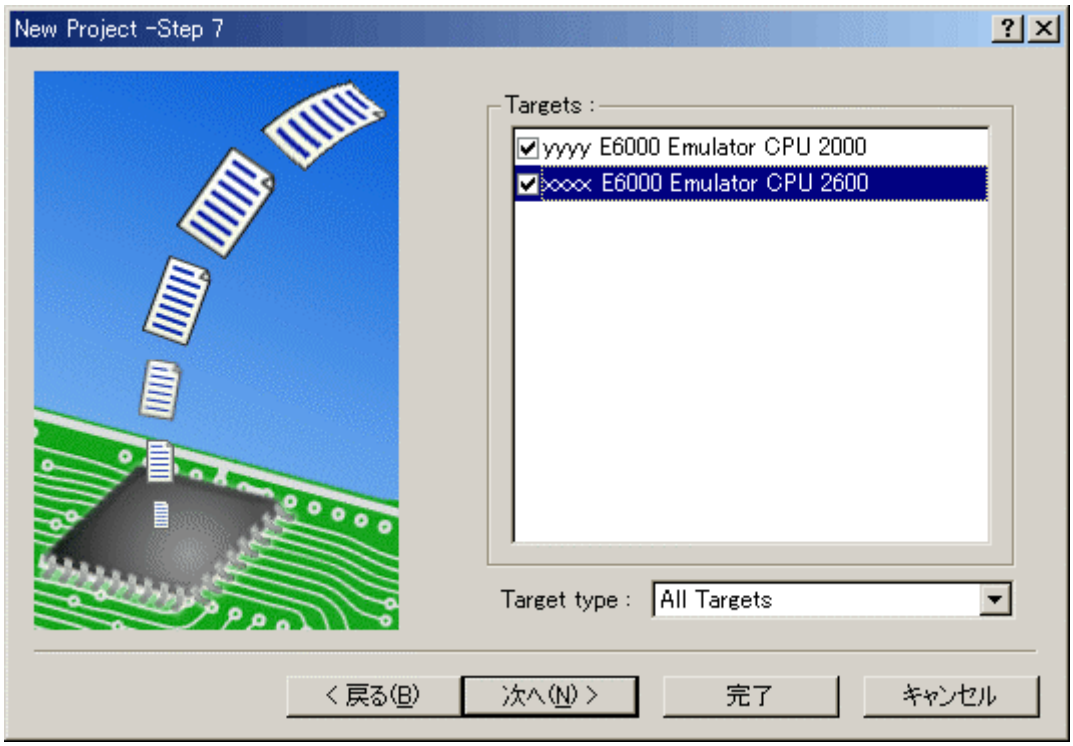


図 4-9　　[New Project-Step 7]**ダイアログボックス**

ここでは、HEW 起動時に使用するセッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンを押してください。セッションファイルについての詳細は「4.4　デバッグセッション」を参照してください。

(4)次に、コンフィグレーションファイル名を設定します。
コンフィグレーションとは、エミュレータ以外の HEW の状態を保存するファイルです。

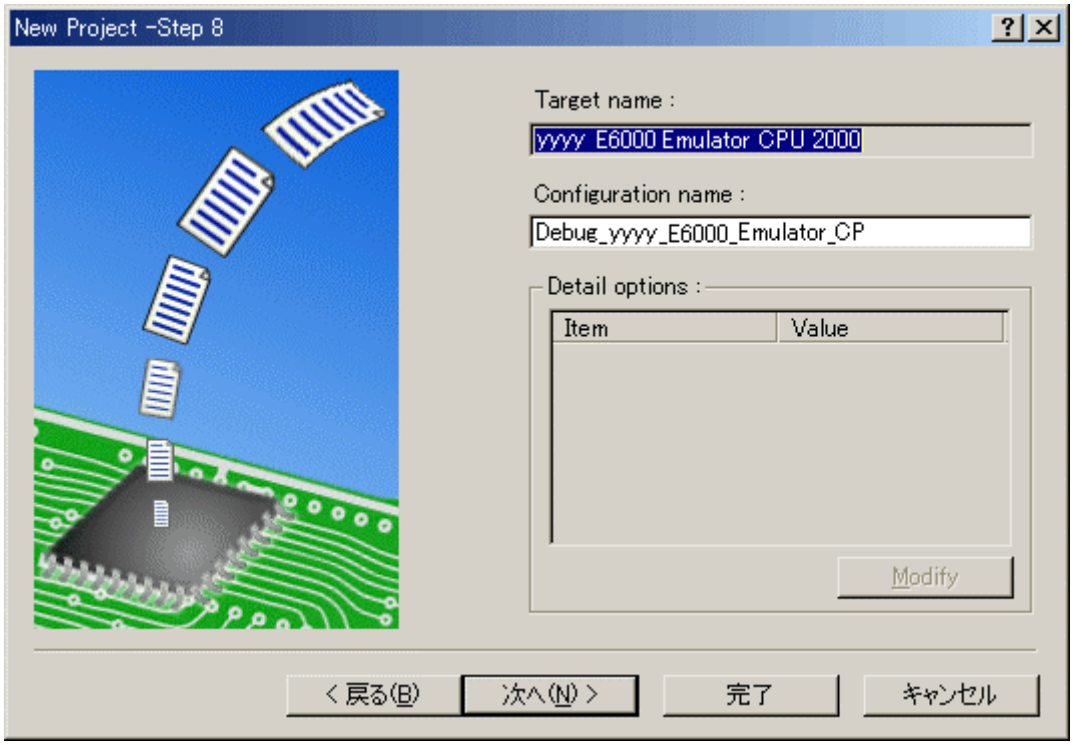


図 4-10　[New Project-Step 8]**ダイアログボックス**

[New Project-Step 7]ダイアログボックス（図 4-9）で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンを押し、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了しましたら、E6000 エミュレータに関する設定は終了です。
画面の指示に従い、新規ワークスペースの作成を完了してください。HEW が起動します。

(5)HEW 起動後、E6000 エミュレータを接続してください。
E6000 エミュレータは、HEW 起動後すぐに接続する必要はありません。
E6000 エミュレータを接続する場合は、E6000 エミュレータ起動時の設定を行ってから接続する方法と E6000 エミュレータ起動時の設定を行わずに簡単に接続する方法があります。エミュレータの接続についての詳細は「4.5　エミュレータの接続」を参照してください。

4.2.3　既存のワークスペースを指定する場合

(1)HEW 起動時に表示される、[ようこそ!]ダイアログボックスで、[**別のプロジェクトワークスペースを参照する**]ラジオボタンを選択し、[OK]ボタンを押してください。

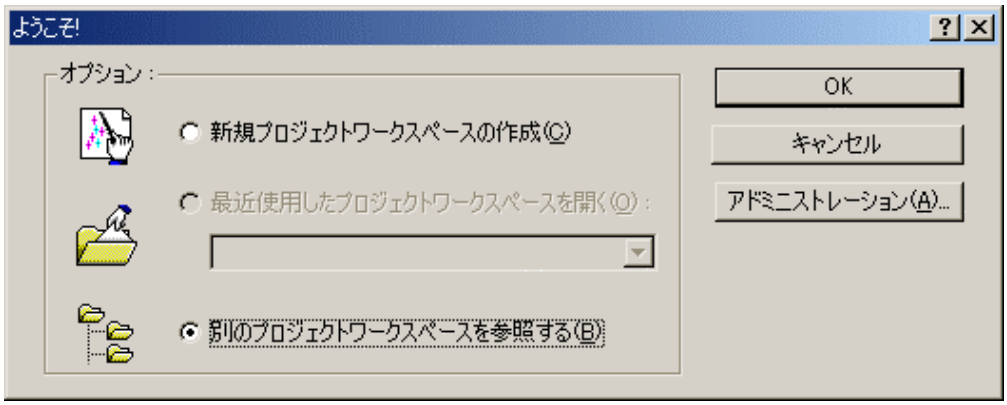


図 4-11　[**ようこそ!**]**ダイアログボックス**

(2)[ワークスペースを開く]ダイアログボックスが開きますので、ワークスペースが作成されているディレクトリを指定してください。

ディレクトリの指定後、ワークスペースファイル（拡張子 .hws）を選択し[開く]ボタンを押してください。

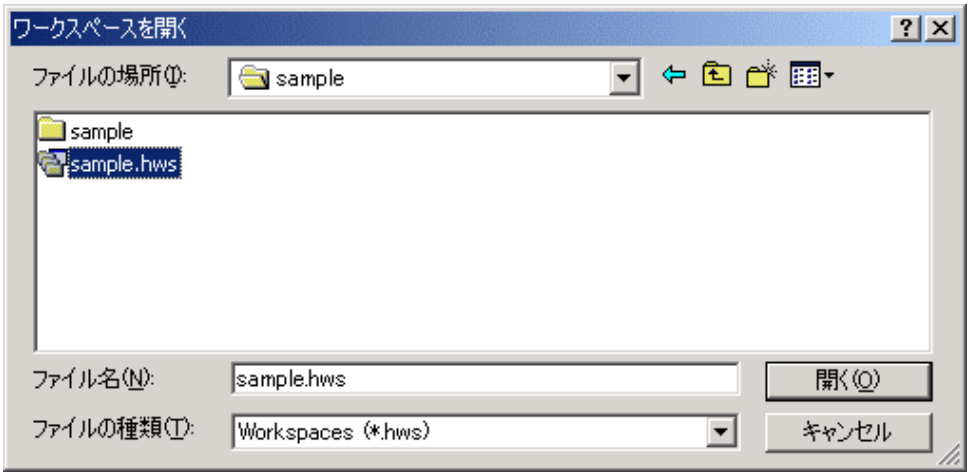


図 4-12　[ワークスペースを開く]ダイアログボックス

(3)HEW が起動され、指定したワークスペースの保存状態が復元されます。
指定したワークスペースがエミュレータに接続された状態を保存していた場合には、エミュレータへの接続が自動で行われます。指定したワークスペースがエミュレータに接続されていない状態を保存していた場合に、エミュレータの接続を行う場合は、「4.5　エミュレータの接続」を参照してください。

4.3　E6000 エミュレータ起動時の設定

E6000 エミュレータの起動時、コマンドチェインの実行を自動的に行うことができます。
また、ダウンロードするロードモジュールを複数登録することができます。
登録したロードモジュールは、ワークスペースウインドウに表示されます。

[オプション]メニューから[デバッグの設定]を選択してください。
[デバッグの設定]ダイアログボックスが開きます。

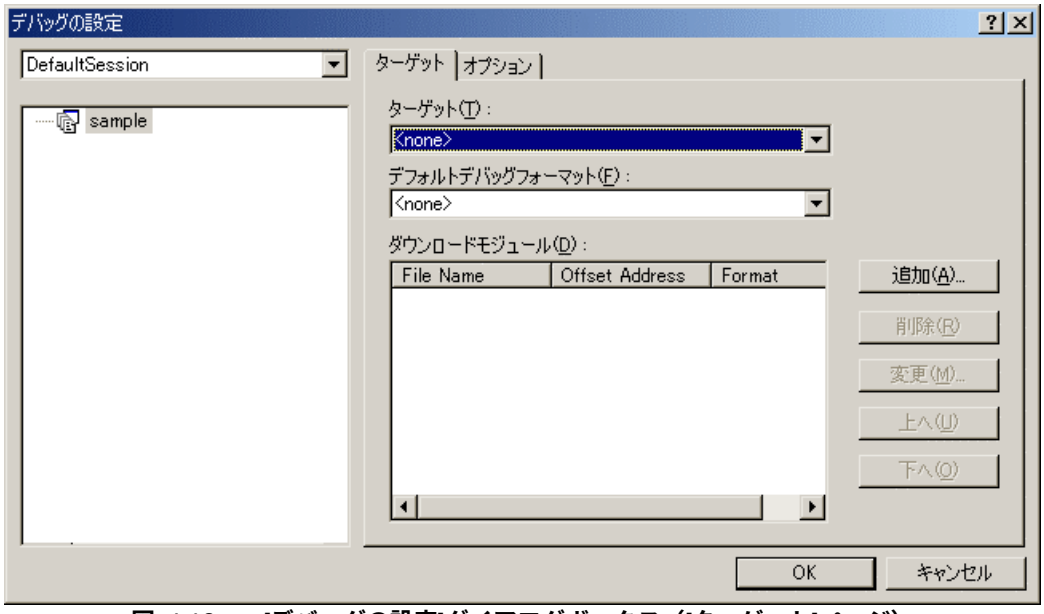


図 4-13　[デバッグの設定]ダイアログボックス（[ターゲット]ページ）

[ターゲット]ドロップダウンリストボックスで接続したい製品名を選択してください。
[デフォルトデバッグフォーマット] ドロップダウンリストボックスで、ダウンロードするロードモジュールの形式を選択し、それに対応するダウンロードモジュールを[ダウンロードモジュール]リストボックスに登録してください。

次に、[オプション]ページをクリックしてください。

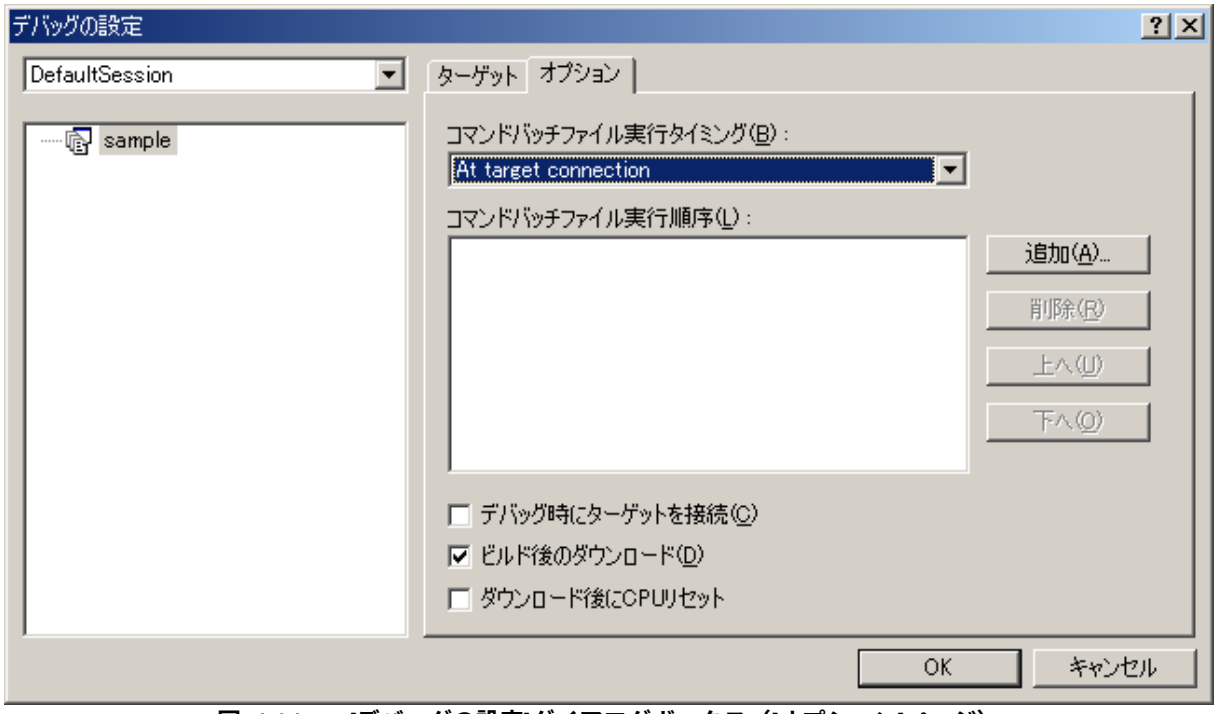


図 4-14　　**[デバッグの設定]ダイアログボックス（[オプション]ページ）**

ここでは、指定したタイミングで自動的に実行するコマンドチェインを登録します。
指定できるタイミングは以下 3 点です。

・エミュレータ接続時
・ダウンロード直前
・ダウンロード直後

[コマンドバッチファイル実行タイミング]ドロップダウンリストボックスで、コマンドチェインを実行するタイミングを指定してください。
また、[コマンドバッチファイル実行順序]リストボックスに、指定したタイミングで実行するコマンドチェインファイルを登録してください。

4.4　デバッグセッション

HEWは、ビルダオプションをコンフィグレーションへ保存することができます。同様に、HEWは、ターゲットオプションをセッションに保存することもできます。セッションには、デバッギングプラットフォーム、ダウンロードするプログラム、各デバッギングプラットフォームのオプションを保存することができます。

セッションは、コンフィグレーションとは直接関連がありません。これは、複数のセッションが同じダウンロードモジュールを共有し、プログラムの不要なリビルドを避けられることを意味します。

各セッションのデータは、別々のファイルで HEW プロジェクトに保存します。詳細については、以下で説明します。

4.4.1　セッションを選択する

セッション選択するには、次の2通りの方法があります。

・ツールバーから選択する

1. ツールバーのドロップダウンリストボックスからセッションを選んでください。

図 4-15　ツールバーの選択

・ダイアログボックスから選択する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します。

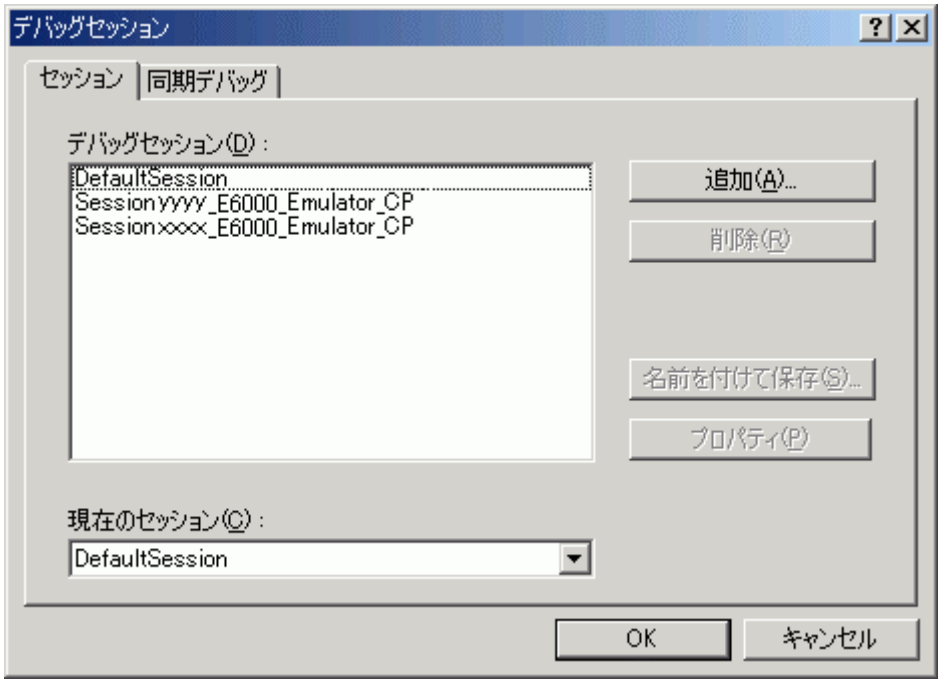


図 4-16　デバッグセッションダイアログボックス

2. [現在のセッション]ドロップダウンリストから使用したいセッションを選んでください。
3. [OK]ボタンをクリックして、セッションを設定してください。

4.4.2　セッションの追加と削除

別のセッションから設定をコピーしたり、セッションを削除したりして、新しいセッションを追加することができます。

・新しい空のセッションを追加する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. [追加...]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを表示します。

3. [新規セッションの追加]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。
6. 入力したセッション名のファイルを新しく作成します。ファイルが既に存在する場合は、エラーを表示します。

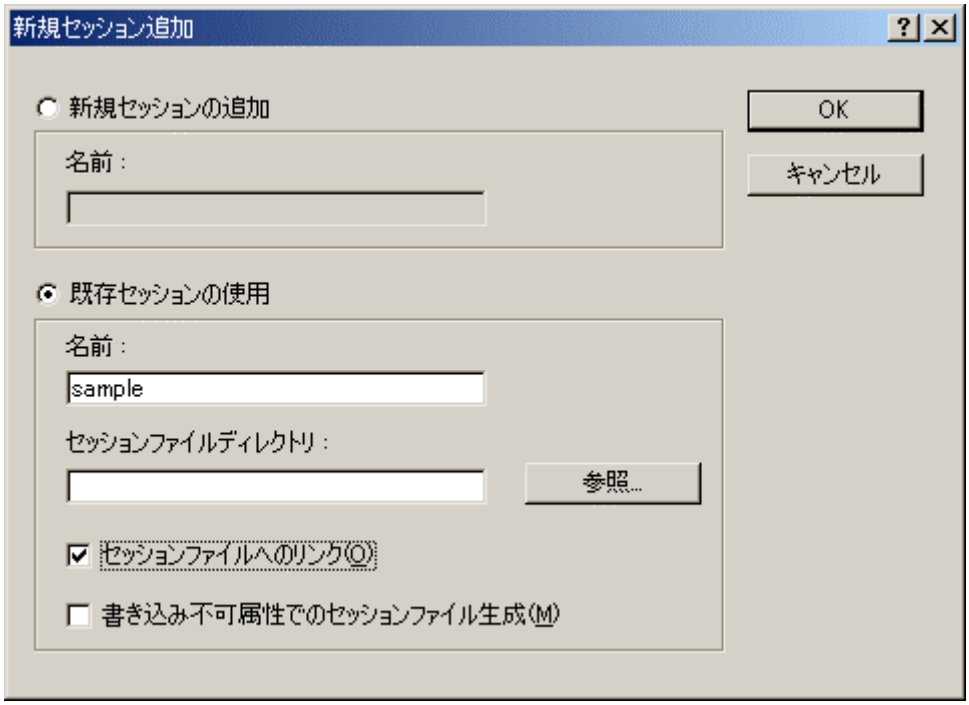


図 4-17　新規セッション追加ダイアログボックス

・既存のセッションを新しいセッションファイルにインポートする

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. [追加...]ボタンをクリックしてください。[新規セッション追加]ダイアログボックスを表示します（図 4-17）。
3. [既存セッションの使用]ラジオボタンをチェックしてください。
4. セッションの名前を入力してください。
5. 現在のプロジェクトにインポートしたい既存のセッションファイルを入力するか、[参照...]ボタンをクリックして選択してください。
   [セッションファイルへのリンク]チェックボックスをチェックしない場合、プロジェクトディレクトリにインポートした新しいセッションファイルを生成します。
   [セッションファイルへのリンク]チェックボックスをチェックした場合、プロジェクトディレクトリに新しいセッションファイルは生成せず、既存のセッションファイルにリンクします。
   [書き込み不可属性でのセッションファイル生成]チェックボックスをチェックした場合、リンクしたセッションファイルをリードオンリーで使用します。
6. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。

・セッションを削除する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. 修正したいセッションを選んでください。
3. [削除]ボタンをクリックしてください。
   現在のセッションを削除することはできません。
4. [OK]ボタンをクリックし、[デバッグセッション]ダイアログボックスを閉じてください。

・セッションのプロパティを見る

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. 見たいプロパティのあるセッションを選んでください。
3. [プロパティ]ボタンをクリックしてください。[セッションプロパティ]ダイアログボックスを表示します。

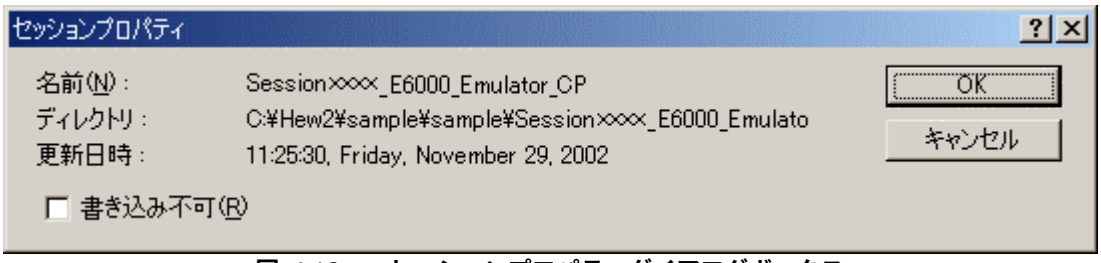


図 4-18　セッションプロパティダイアログボックス

・セッションをリードオンリーにする

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. リードオンリーにしたいセッションを選んでください。
3. [プロパティ]ボタンをクリックしてください。[セッションプロパティ]ダイアログボックスを表示します（図 4-18）。
4. [書き込み不可]チェックボックスをチェックしてください。リンクをリードオンリーにします。これは、デバッガ設定ファイルを共有する場合、およびデータを間違って修正したくない場合に便利です。
5. [OK]ボタンをクリックしてください。

・セッションを別名で保存する

1. [オプション->デバッグセッション...]を選んでください。[デバッグセッション]ダイアログボックスを表示します（図 4-16）。
2. 保存したいセッションを選んでください。
3. [名前を付けて保存]ボタンをクリックしてください。[セッションの保存]ダイアログボックスを表示します。
4. 新しいファイルを保存する場所を指定してください。
5. セッションファイルを別の場所へエクスポートしたい場合は、[プロジェクトとのリンク]チェックボックスをチェックしないでください。現在のセッションの場所の代わりに、この場所をHEWで使用したい場合は、[プロジェクトとのリンク]チェックボックスをチェックしてください。
6. [OK]ボタンをクリックしてください。

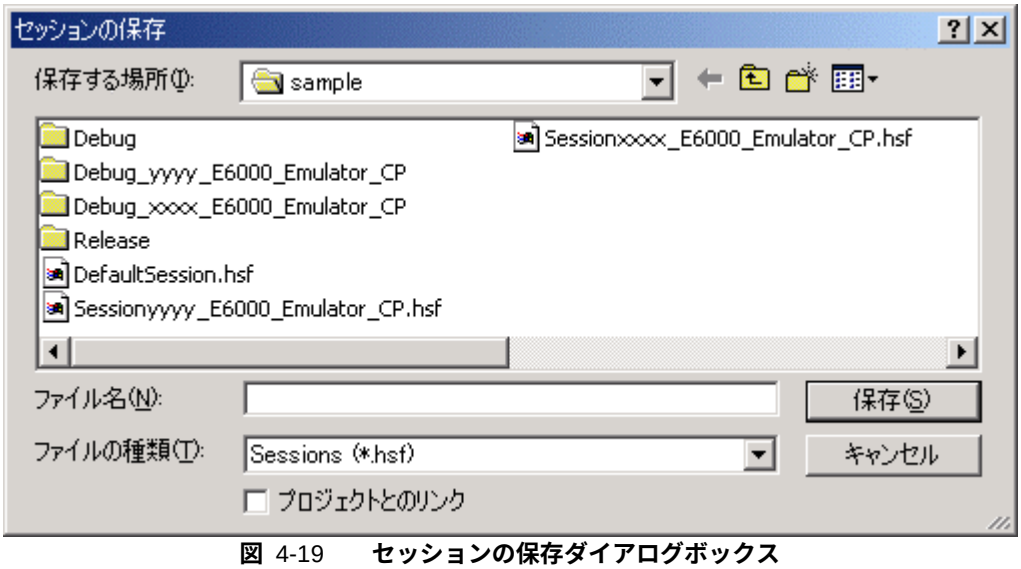


図 4-19　セッションの保存ダイアログボックス

4.4.3　セッション情報を保存する

➲セッションを保存するには

1. [ファイル->セッションの保存]を選んでください。

4.4.4 セッション情報の再ロード

[ファイル->セッションのリフレッシュ]を選択してください。これにより現在のセッションに加えられている変更を廃棄して、セッションを再ロードします。再ロードの前に確認ダイアログボックスを表示します。

4.5 エミュレータの接続

エミュレータの接続には、以下の方法があります。

(1) E6000 エミュレータ起動時の設定を行ってから接続する方法

[オプション->デバッグの設定...]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェインなどを登録することができます。

[デバッグの設定]ダイアログボックスの詳細については、「4.3 E6000 エミュレータ起動時の設定」を参照してください。

[デバッグの設定]ダイアログボックスの設定終了後、ダイアログボックスを閉じると、E6000 エミュレータが接続されます。

(2) E6000 エミュレータ起動時の設定を行わずに簡単に接続する方法

E6000 エミュレータを使用する設定があらかじめ登録されているセッションファイルに切り替えることにより、E6000 エミュレータを簡単に接続できます。

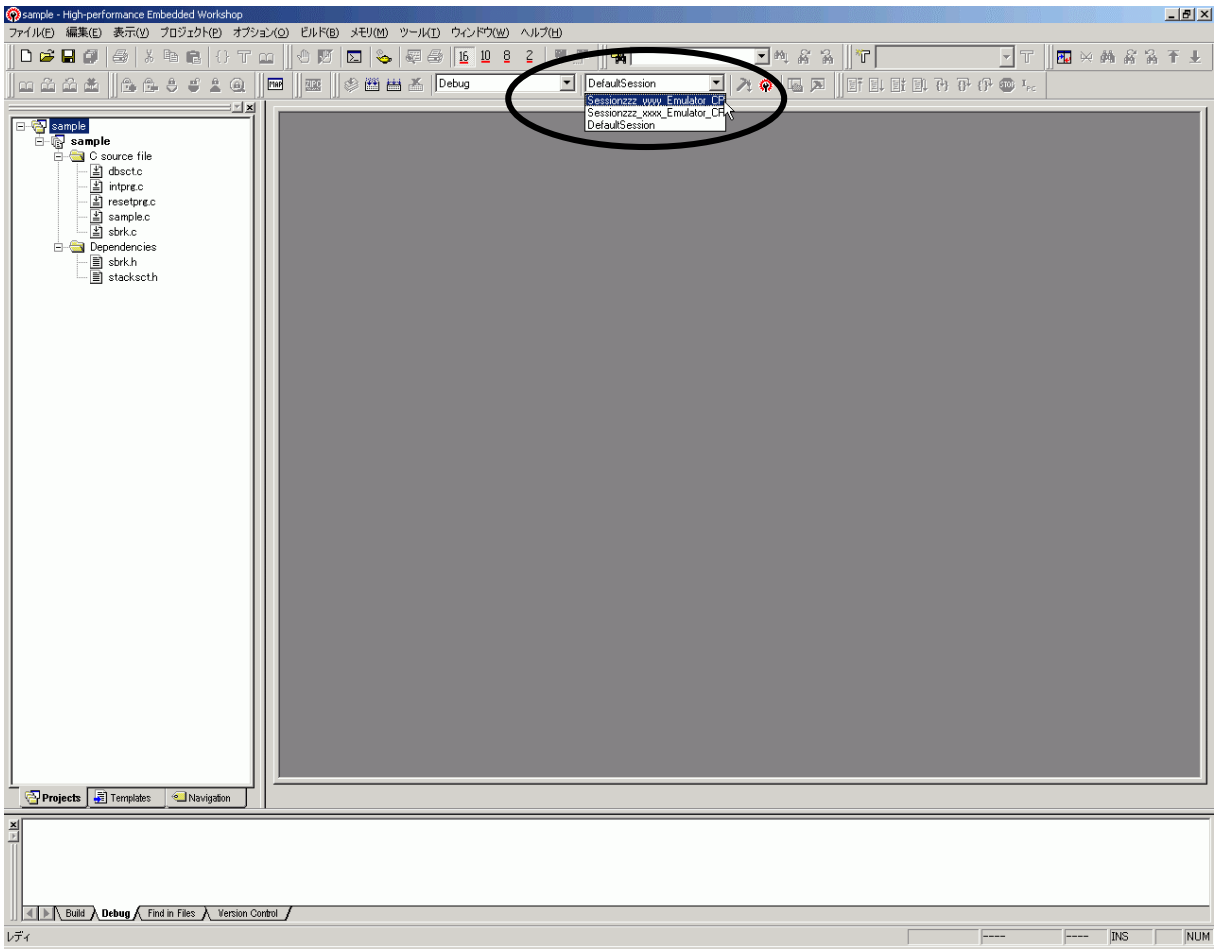


図 4-20 セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、「図 4-10 [New Project-Step 8]ダイアログボックス」の[Target name]テキストボックス内で設定されている文字列を含んだセッションファイル名を選択してください。

このセッションファイルには、E6000 エミュレータを使用する設定が登録されています。

選択終了後、E6000 エミュレータが自動的に接続されます。セッションファイルについての詳細は「4.4 デバッグセッション」を参照してください。

4.6　エミュレータの再接続

エミュレータ切断状態時に以下の方法で再接続を行うことができます。

[ビルド->デバッグ->接続]を選択するか、接続ツールバーボタン をクリックしてください。
エミュレータの接続が開始されます。

【注】　エミュレータの再接続を行う場合、あらかじめロードモジュールが HEW に登録されている必要があります。登録方法については「4.3 E6000 エミュレータ起動時の設定」を参照して下さい。

4.7　エミュレータの終了

ツールチェインをご使用の場合、エミュレータの終了方法は 2 通りあります。
・起動中のエミュレータの接続を解除する方法
・HEW 自体を終了する方法

(1)起動中のエミュレータの接続を解除する方法

[デバッグ]メニューから接続解除を選択するか、接続解除ツールバーボタン をクリックしてください。

接続解除後、E6000 エミュレータの電源を切ってください。

(2)HEW 自体を終了する方法
ファイルメニューから[アプリケーションの終了]を選択してください。

メッセージボックスが表示されます。必要なら、[はい]ボタンをクリックし、セッションをセーブしてください。セーブ後、HEW は終了します。不要なら、[いいえ]ボタンをクリックしてください。HEW は終了します。

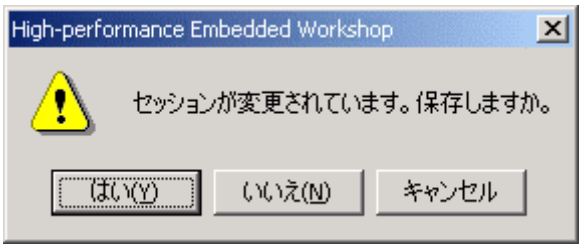


図 4-21　[セッションが変更されています]メッセージボックス

5　デバッグ

デバッグ操作と関連するウィンドウおよびダイアログボックス について説明します。

5.1　エミュレーション環境を設定する

この節では、エミュレーションを行うための環境を設定する方法を説明します。

5.1.1　Configuration Properties ダイアログボックスを開く

[オプション->エミュレータ->システム...]を選択するか、[Emulator System]ツールバーボタンをクリックすると、[Configuration Properties]ダイアログボックスが開きます。

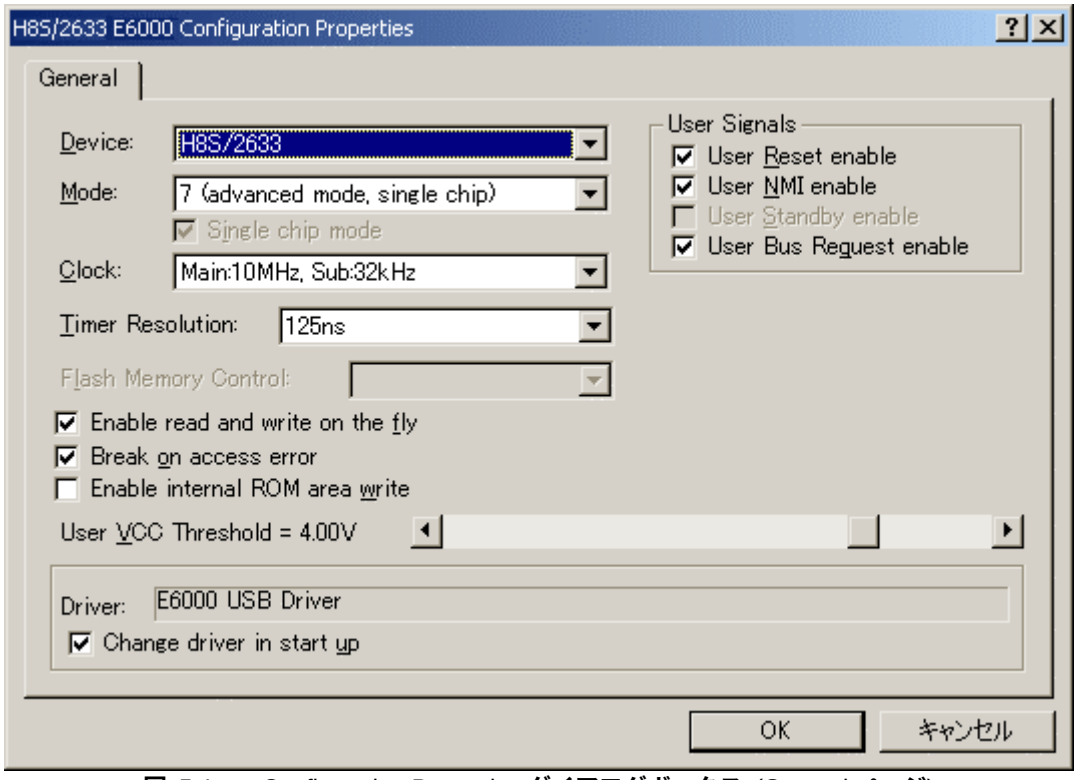


図 5-1　Configuration Properties **ダイアログボックス** (General **ページ**)

E6000 エミュレータにプログラムをダウンロードする前に、このダイアログボックスで対象 MCU 条件を設定します。

[General]

[Device]	エミュレーションする MCU を指定します。MCU 一覧にない MCU を指定する場合は、Custom を指定し、使用する MCU の機能を設定します。詳細に関しては、それぞれのハードウェアマニュアルを参照してください。
[Mode]	MCU の動作モードを指定します。
[Clock]	MCU のクロック速度とサブクロック速度を指定します。
[Timer Resolution]	実行時間の測定に使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 20ns, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us 実行時間測定用タイマは 40 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 20ns で約 6 時間、分解能 16us で約 200 日です。 カウンタがオーバフローした場合、オーバフローしたことを示すプロンプト”> ”と共に測定可能な最大時間を表示します。

[Enable read and write on the fly]	このチェックボックスをチェックすると、プログラム実行中にメモリにアクセスすることが可能です。 リアルタイム性はありませんので、リアルタイムでのエミュレーションを行いたい場合には、チェックしないでください。
[Break on access error]	このチェックボックスをチェックすると、プログラムでアクセス禁止エリアにアクセス、または書き込み禁止エリアに書き込みが生じると、ブレーク（ユーザプログラム停止）します。
[Enable internal ROM area write]	このチェックボックスをチェックすると、ユーザプログラムによる内蔵 ROM エリアへのライトが可能になります。ライトが行われたかどうかは、[拡張モニタ] ウィンドウによって知らされます。
[User VCC Threshold]	ユーザシステム電圧レベルを指定します。
[User Signals]	このチェックボックスをチェックすると、ユーザシステムからのリセット信号、NMI 信号、スタンバイ信号、バスリクエスト信号を有効にします。
[Driver]	現在使用している E6000 ドライバを表示しています。
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

【注】 本ダイアログボックスで設定可能な項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.2 MCU 一覧にない MCU を設定する

[Configuration Properties]ダイアログボックスの[Device]項目で[Custom]を選択すると、[Configuration Properties]ダイアログボックスに[Custom Device]ページが追加されます。

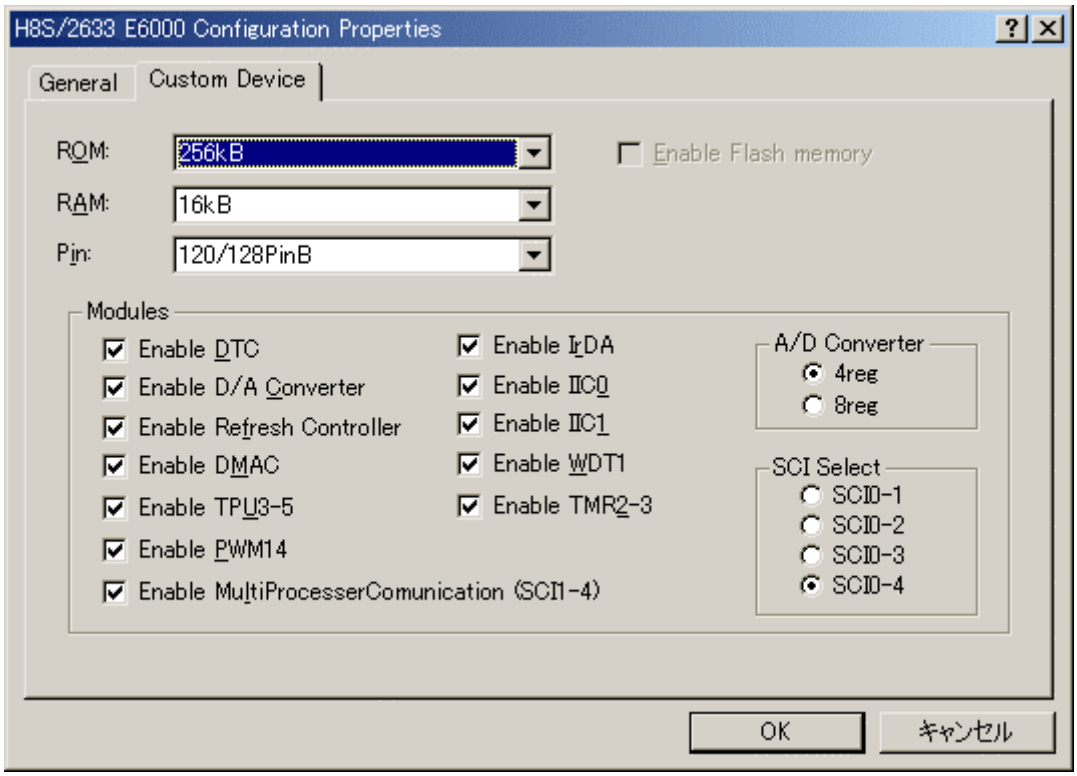


図 5-2 Configuration Properties **ダイアログボックス** (Custom Device **ページ**)

ここではDevice指定のMCU一覧にないMCUの機能を指定します。各項目は直前に選択されたデバイスの値が反映されています。

[Custom Device]

[ROM]	内蔵 ROM エリアサイズを指定します。
[RAM]	内蔵 RAM エリアサイズを指定します。
[Pin]	製品のパッケージを指定します。
[Modules]	チェックボックスをチェックし、内蔵周辺機能を指定します。

【注】　本ダイアログボックスで設定可能な項目はご使用のエミュレータにより異なります。
また、ご使用のエミュレータによっては[Custom Device]機能をサポートしていない場合があります。
詳細につきましては「8　本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.3　接続するインタフェースを選択する

[Configuration Properties]ダイアログボックスの[Change driver in start up]項目をチェックすると、次回 E6000 接続時にドライバを選択することができます。

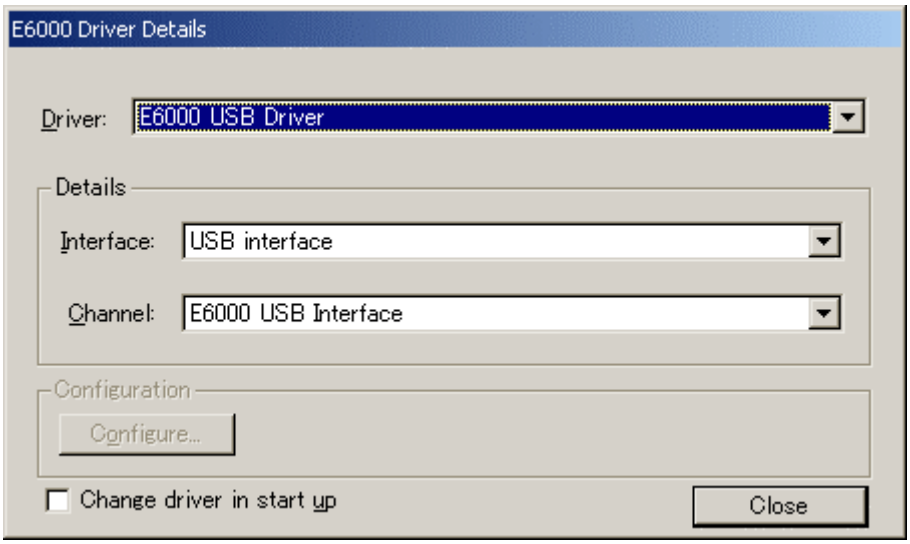


図 5-3 Driver Details **ダイアログボックス**

[Driver]	HEW と E6000 エミュレータを接続するドライバを選択します。
[Details]	接続するドライバの詳細を設定します。
[Interface]	選択したドライバがサポートしているインタフェースを選択します。 本エミュレータでは設定を変更する必要はありません。
[Channel]	選択したインタフェースのチャンネルを選択します。 本エミュレータでは設定を変更する必要はありません。
[Configuration]	ドライバの設定を行います。
[Configure...]	ドライバがコンフィグレーションダイアログをサポートしている場合、設定ダイアログボックスを表示します。（本エミュレータでは使用できません。）
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

5.1.4 Memory Mapping ダイアログボックスを開く

[オプション->エミュレータ->メモリリソース...]を選択するか、[Emulator Memory Resource]ツールバーボタン をクリックすると、[Memory Mapping]ダイアログボックスが開きます。

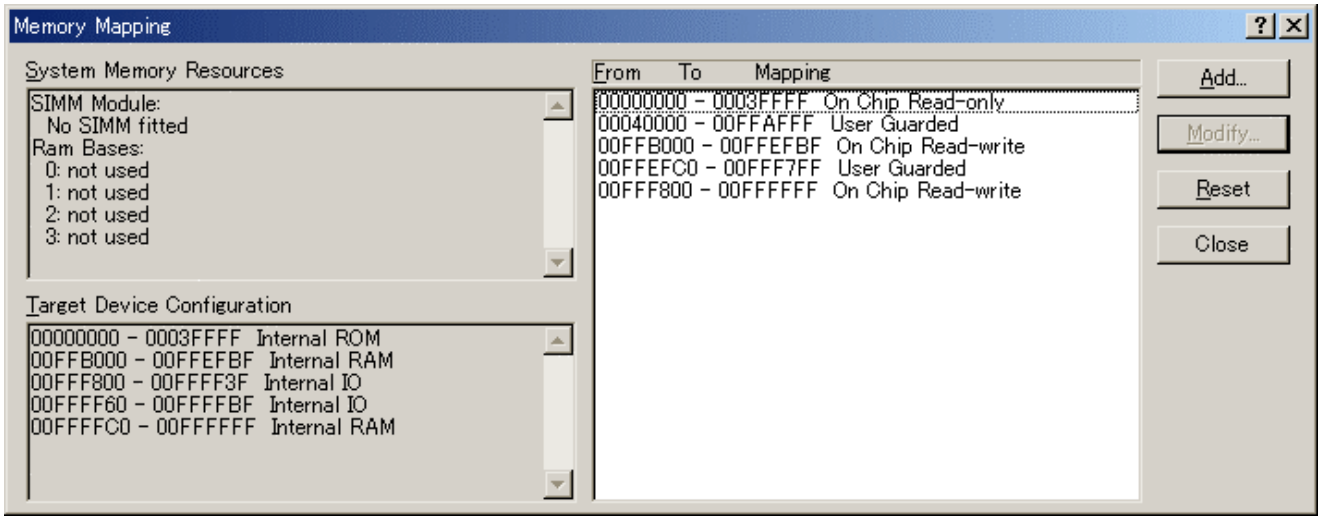


図 5-4 Memory Mapping **ダイアログボックス**

現在のメモリマップを表示しています。

H8S ファミリおよび H8/300H シリーズ E6000 は 4 ブロックのエミュレーションメモリをサポートしています。これらは実装した SIMM に従い、256kbyte または 1Mbyte 単位の指定ができます。各ブロックは、256kbyte または 1Mbyte 境界のアドレス空間に置くことができます。
メモリマップには、H'40(D'64)バイトのブロックがあります。各 64 バイトのブロックは、内部(エミュレーション)メモリまたは外部メモリ、ガーデッド(アクセス禁止)、書き込み禁止またはリード・ライトに設定することができます。

H8/300 シリーズ E6000 は標準でエミュレーションメモリを実装しています。
メモリマップは 1 バイト単位で内部(エミュレーション)メモリまたは外部メモリ、書き込み禁止またはリード・ライトに設定することができます。

[Add...]	メモリマップのアドレス範囲および属性を変更するため [Edit Memory Mapping] ダイアログボックスを開きます。
[Modify...]	メモリマップのアドレス範囲および属性を変更するため [Edit Memory Mapping] ダイアログボックスを開きます。
[Reset]	メモリマップをデフォルト設定にリセットします。
[Close]	ダイアログボックスを閉じます。

ターゲットマイコンのメモリマップ情報は、[ステイタス]ウィンドウの[Memory]シートに表示されます。

【注】 ご使用のエミュレータによりエミュレーションメモリをサポートしていない場合があります。
また、Memory Mapping 機能自体をサポートしていない場合もあります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.5 メモリマップ設定を変更する

[Memory Mapping]ダイアログボックスで[Add...]ボタンをクリックするか、または変更したいメモリマップ設定情報を選択し[Modify...]ボタンをクリックすると[Edit Memory Mapping]ダイアログボックスが開きます。

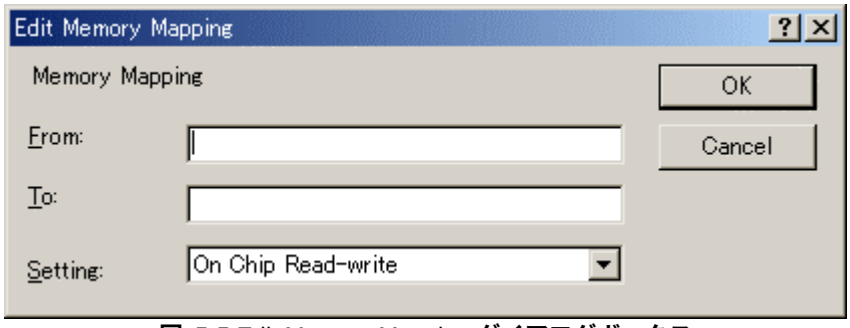


図 5-5 Edit Memory Mapping **ダイアログボックス**

メモリマップのアドレス範囲および属性を変更します。

[From] 範囲の開始アドレスを入力します。

[To] 範囲の終了アドレスを入力します。

[Setting] マップ設定を選択します。
マップ設定は以下 9 つの属性があり、User(外部メモリ)および Emulator(エミュレーションメモリ)属性については変更可能です。

On Chip Read-write	(変更できません)
On Chip Read-only	(変更できません)
On Chip Guarded	(変更できません)
User Read-write	(シングルチップモード時は設定できません)
User Read-only	(シングルチップモード時は設定できません)
User Guarded	
Emulator Read-write	
Emulator Read-only	
Emulator Guarded	

5.2 プログラムをダウンロードする

プログラムをダウンロードし、ソースコードおよびアセンブリ言語ニモニックとして見る方法を説明します。

【注】 ブレークが起こると、HEW はプログラムカウンタ(PC)の場所を表示します。多くの場合、例えば、ELF/DWARF2 をベースにしたプロジェクトが、ビルド時のパスから移動した場合、ソースファイルを自動的に見つけることができない場合があります。この場合、HEW はソースファイルブラウザダイアログボックスを開くので、ユーザは手動でファイルを探すことができます。

5.2.1 プログラムをダウンロードする

デバッグするロードモジュールをダウンロードします。

プログラムのダウンロードは、[デバッグ->ダウンロード]からロードモジュールを選択するか、[ワークスペース]ウィンドウの[Download modules]のロードモジュールを右クリックすると表示されるポップアップメニューより[ダウンロード]を選択します。

【注】 プログラムをダウンロードする場合、あらかじめロードモジュールとして HEW に登録されている必要があります。登録方法については「4.3 E6000 エミュレータ起動時の設定」を参照して下さい。

5.2.2 ソースコードを表示する

[ワークスペース]ウィンドウのソースファイル名を選択し、ポップアップメニューから[開く]を選択すると、HEWはエディタでソースファイルを表示します。[ワークスペース]ウィンドウのソースファイル名をダブルクリックすることによっても表示することができます。

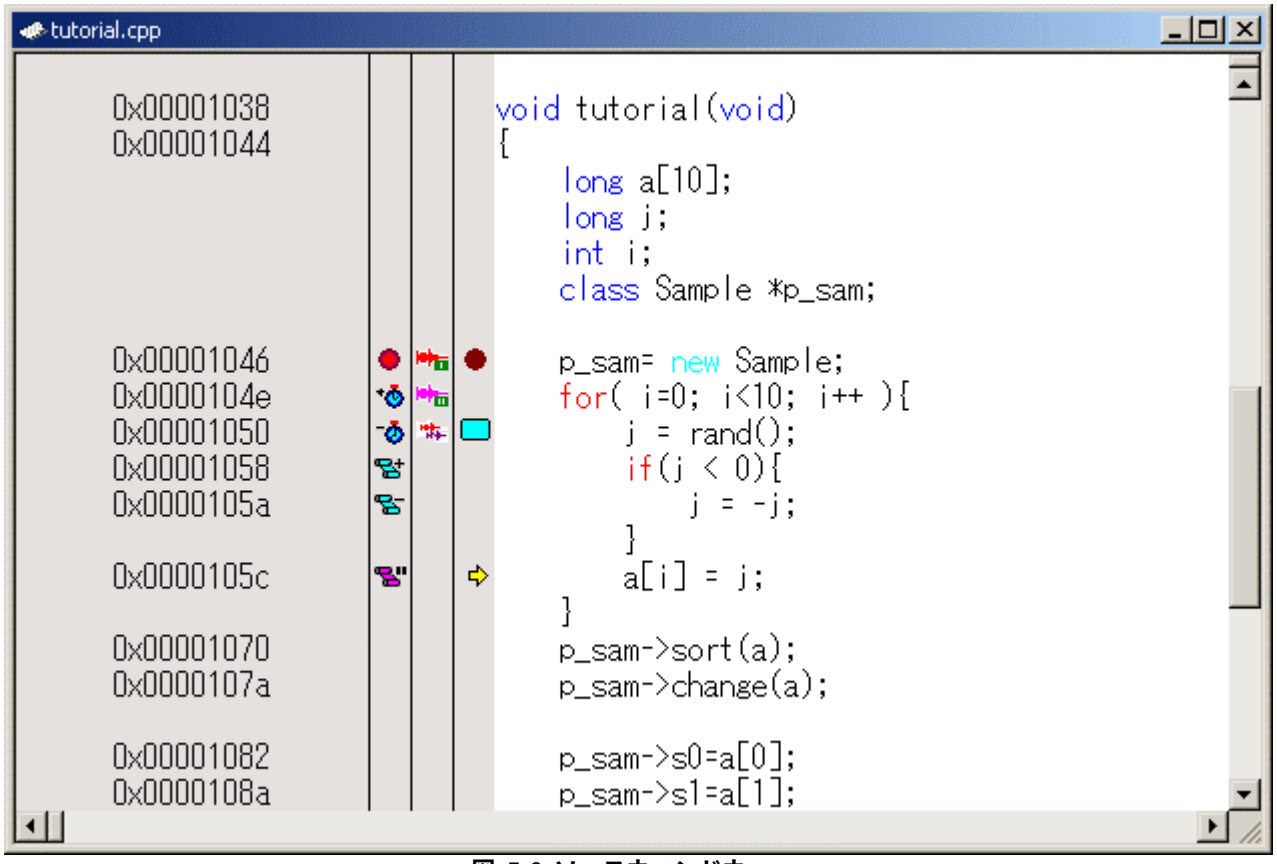


図 5-6 ソースウィンドウ

本ウィンドウでは左端に行情報として下記を表示します。

1 **列目**	(Source address **カラム**)	**ソース行に対応するアドレス情報の表示**
2 **列目**	(Event **カラム**)	**イベント情報(ブレーク)**
3 **列目**	(EXT.2 Trigger **カラム**)	EXT.2 Trigger **情報**
4 **列目**	(Editor **カラム**)	PC**、ブックマーク、ブレークポイント情報**

右側のソース表示画面をエディタウィンドウと呼びます。

Source address カラム

プログラムをダウンロードすると、Source address **カラムに現在のソースファイルに対するアドレスを表示します。**
本機能は PC **値やブレークポイントをどこに設定するかを決めるときに便利です。**

Event カラム

Event カラムには下記を表示します。

イベントチャネルまたは範囲チャネルによるアドレス条件ブレーク

イベントチャネルによる時間計測の開始

イベントチャネルによる時間計測の終了

Point to Point の範囲トレースの開始

Point to Point の範囲トレースの終了

トレース停止

これらの設定は、ポップアップメニューからも可能です。

EXT.2 Trigger カラム

EXT2. Trigger カラムには下記を表示します。

EXT.2-1 トリガの条件

EXT.2-2 トリガの条件

EXT.2-3 トリガの条件

EXT.2-4 トリガの条件

EXT.2　トリガの条件が２つ以上ある場合

これらの設定は、ポップアップメニューからも可能です。

Editor カラム

Editor カラムには下記を表示します。

ブックマークを設定している

PC Break を設定している

PC 位置

➲すべてのソースファイルでカラムをオフにするには

1. [エディタ]ウィンドウを右クリックしてください。または、[編集]メニューを選択してください。
2. [表示カラムの設定...]メニュー項目をクリックしてください。
3. [エディタ全体のカラム状態]ダイアログボックスを表示します。
4. チェックボックスは、そのカラムが有効か無効かを示します。チェックしている場合は有効です。チェックボックスがグレー表示の場合、一部のファイルではカラムが有効で、別のファイルでは無効であることを意味します。
   オフにしたいカラムのチェックボックスからチェックを外してください。
5. [OK]ボタンをクリックして、新しいカラム設定を有効にしてください。

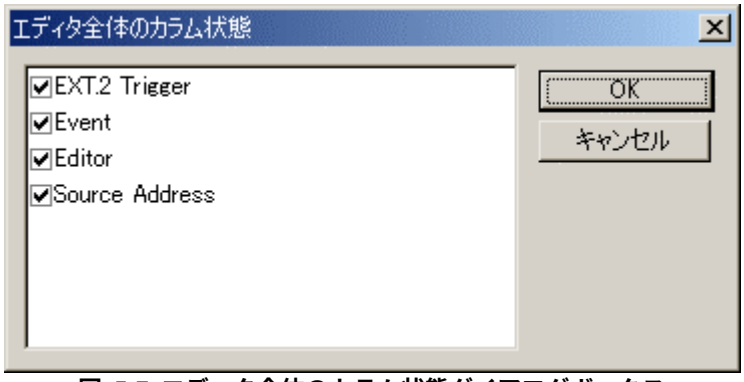


図 5-7 エディタ全体のカラム状態ダイアログボックス

➲1つのソースファイルでカラムをオフにするには

1. 削除したいカラムのあるソースファイルを開き、[編集]メニューをクリックしてください。
2. [カラム]メニュー項目をクリックしてください。カスケードしたメニュー項目が現れます。各カラムを、このポップアップメニューに表示します。カラムが有効である場合、名前の横にチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。

5.2.3 アセンブリ言語コードを表示する

[エディタ]ウィンドウのカーソル位置に対応するアドレス表示がある場合、ポップアップメニューから、 [逆アセンブリ]を選択することにより、[逆アセンブリ]ウィンドウを表示することができます。
[逆アセンブリ]ウィンドウの表示開始アドレスは、 [エディタ]ウィンドウのカーソル位置に対応するアドレスとなります。

ソースファイルが存在しなくてもアセンブリ言語レベルでコードを表示したい場合は、[表示->逆アセンブリ...]を選択するか、[逆アセンブリ]ツールバーボタン をクリックします。[逆アセンブリ]ウィンドウは現在の PC の場所で開きます。

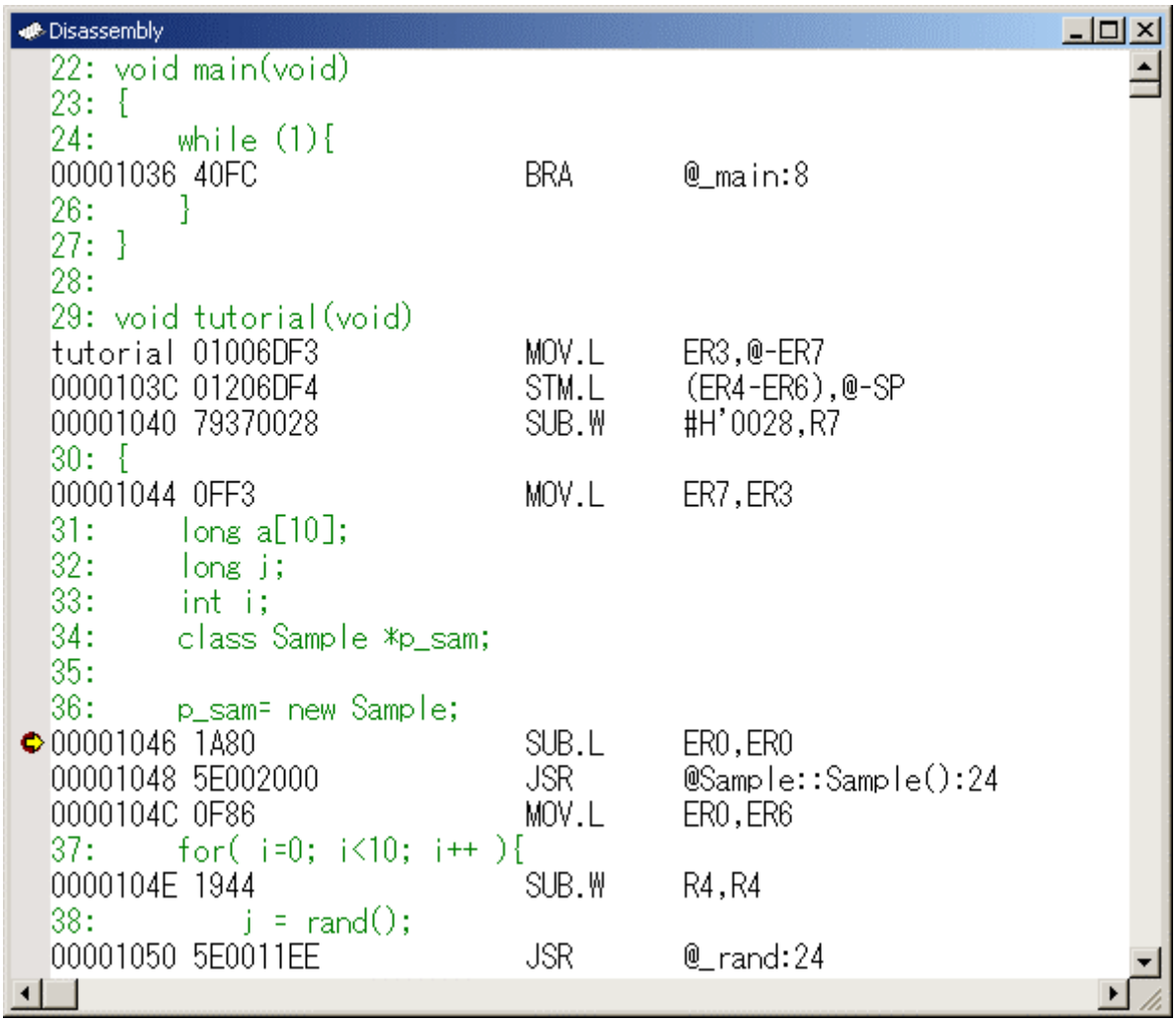


図 5-8 逆アセンブリウィンドウ

5.2.4 アセンブリ言語コードを修正する

修正したい命令をダブルクリックすることによって、アセンブリ言語コードを修正することができます。[アセンブル]ダイアログボックスが開きます。

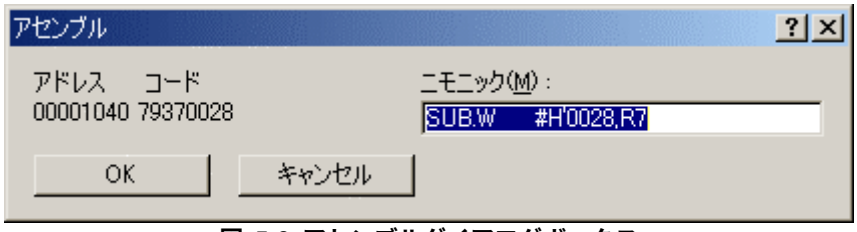


図 5-9 アセンブルダイアログボックス

アドレス、命令コード、およびニモニックを表示します。
[ニモニック]フィールドに新しい命令を入力（または古い命令を編集）します。
'Enter'キーを押すと、メモリ内容を新しい命令コードに書き換えて、次の命令に移ります。
[OK]ボタンをクリックすると、メモリ内容を新しい命令コードに書き換えて、ダイアログボックスを閉じます。
[キャンセル]ボタンをクリックするか"Esc"キーを押すと、メモリ内容を書き換えずに、ダイアログボックスを閉じます。

【注】 アセンブリ言語コードは、現在のメモリ内容から表示しています。メモリ内容を修正すると、[逆アセンブリ]ウィンドウおよび[アセンブル]ダイアログボックスでは、新しいアセンブリ言語コードを表示します。しかし、[エディタ]ウィンドウに表示しているソースファイルは変更しません。これはソースファイルにアセンブラを含む場合も同じです。

5.2.5 特定のアドレスを見る

[逆アセンブリ]ウィンドウを使って作成したプログラムを見ているとき、プログラム内のほかのところも見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスに直接行くことができます。ポップアップメニューから[表示アドレス設定...]を選択します。

図 5-10 アドレス設定ダイアログボックス

[アドレス]エディットボックスにアドレスを入力して、[OK]ボタンをクリックするか'Enter'キーを押します。アドレスは、ラベル名で入力することも可能です。
[逆アセンブリ]ウィンドウを更新して新しいアドレスコードを表示します。オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスを開くので、関数を選択してください。これについては、このマニュアルの「5.14.3 複数ラベルをサポートする」で詳細を説明します。

5.2.6 現在のプログラムカウンタアドレスを見る

HEW でアドレスまたは値を入力できるところでは、式も入力することができます。先頭にハッシュ文字をつけたレジスタ名を入力すると、そのレジスタ内容を式の値として使用します。従って、[表示アドレス設定]ダイアログボックスで、"#pc"という式を入力すると、[ソース]または[逆アセンブリ]ウィンドウには、現在の PC アドレスを表示します。例えば、"#PC+0x100"といった PC レジスタおよびオフセットの式を入力することにより現在の PC のオフセットも表示することができます。

5.3　コマンドラインインタフェースでデバッグする

ウィンドウメニューやウィンドウコマンドを使用しないで、テキストベースのコマンドを入力してデバッグするには、[コマンドライン]ウィンドウを使用します。

5.3.1　コマンドラインウィンドウを開く

[コマンドライン]ウィンドウを開くには、[表示->コマンドライン]を選択するか、[コマンドライン]ツールバーボタン をクリックします。

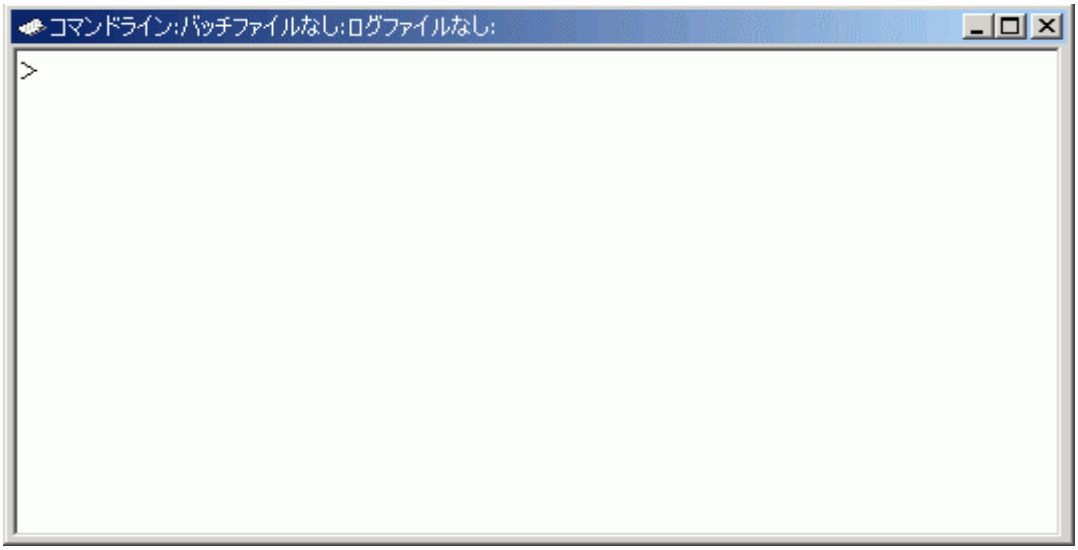


図 5-11 コマンドラインウィンドウ

テキストベースのコマンドを入力してデバッグプラットフォームを制御できるウィンドウです。コマンド行をファイルから呼び出して実行すること、および出力結果をファイルに記録することができます。

[コマンドライン]ウィンドウ最終行のコマンドプロンプト('>')に続けて入力後、"Enter"キーを押すとコマンドを実行します。使用できるコマンドについては、「付録 G　コマンドライン一覧」およびオンラインヘルプを参照してください。

ウィンドウタイトルとしてバッチファイル名とログファイル名をコロンで区切って表示します。

最終行で"Ctrl+↑"または"Ctrl+↓"を押すと過去に実行したコマンド行を呼び出すことができます。

本ウィンドウで、HEW コマンドおよび、TCL コマンドが入力できます。

5.3.2　コマンドファイルを設定する

あらかじめ定義した一連のコマンド行を実行する場合は、コマンドファイルを利用すると便利です。

コマンドファイルはテキストエディタで作成し、実行すべきコマンドラインを記述しておきます。コマンドファイル型名のデフォルトは".hdc"です。

コマンドファイルを設定するには、[バッチファイルを指定]ダイアログボックスを使用します。

ポップアップメニューから[バッチファイル指定...]を選択すると、[バッチファイルを指定]ダイアログボックスを表示します。コマンドファイル名(*.hdc)を入力できます。[OK]ボタンをクリックすると、設定したコマンドファイル名を、ウィンドウタイトルに表示します。[キャンセル]ボタンをクリックすると、設定を変更しないでダイアログボックスを閉じます。

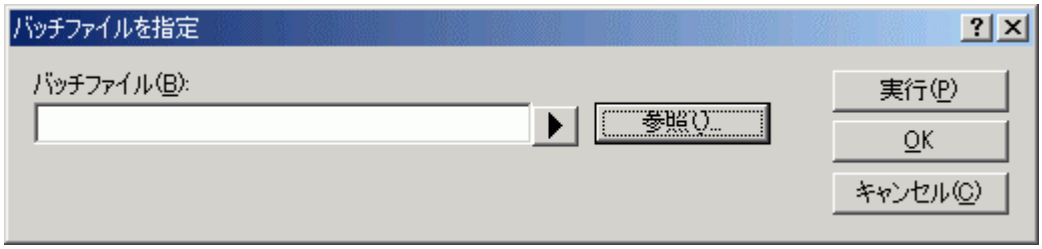


図 5-12 バッチファイルを指定ダイアログボックス

5.3.3 コマンドファイルを実行する

コマンドファイルを実行するには、[バッチファイルを指定]ダイアログボックスで[実行]ボタンをクリックするか、ポップアップメニューから[バッチファイルの実行を開始]を選択します。[バッチファイルの実行を開始/停止]メニューはコマンドファイル実行中にグレー表示となり、コマンドファイル実行が停止してユーザに制御が戻ったときに有効表示になります。

5.3.4 コマンド実行を中断する

コマンド実行を中断する場合は、ポップアップメニューから [バッチファイル停止]を選択します。[バッチファイル停止]メニューはコマンド実行中に有効表示になります。

5.3.5 ログファイルを設定する

コマンド実行結果を保存するログファイルは[ログファイルを開く]ダイアログボックスで設定します。
[ログファイルを開く]ダイアログボックスを開くには、ポップアップメニューから[ログファイル指定...]を選択します。

図 5-13 ログファイルを開くダイアログボックス

出力結果を記録するログファイル(“*.log”)の名前を入力します。ロギングオプションを自動的に設定し、ログファイル名をウィンドウのタイトルバーに表示します。
既に存在しているログファイル名を指定すると、ログを追加するか、以前のログを消去して、新しいログを上書きするかを確認します。

5.3.6 ログファイルへの出力を開始/停止する

ファイルへのロギング処理を実行するか、停止するかの切り替えは、ポップアップメニューの[ロギング開始/停止]で行います。ログファイルの内容は、ロギングが終了するか、チェックボックスをクリアしてロギングを一時的に停止しなければ表示できないことにご注意ください。ロギングを再び開始すると、ログファイルに追加します。

5.3.7 ファイルの full パスを入力する

カレントディレクトリは移動する可能性があるため、[コマンドライン]ウィンドウではファイル名をfullパスで指定することをお勧めします。しかし、fullパスのファイル名をキー入力するのは煩雑であるため、ファイルを参照して選択するだけでfullパスファイル名を入力できる機能をサポートしています。
ポップアップメニューで[参照...]を選択すると、[Browse]ダイアログボックスを表示します。ここでファイルを選択し[開く]をクリックすると、fullパスファイル名をカーソル位置に貼り付けます。カーソルが最終行にある場合のみ使用できます。

5.3.8 プレースホルダを入力する

ポップアップメニューから[プレースホルダ]のサブメニューで選択したプレースホルダをカーソル位置に貼り付けます。カーソルが最終行にある場合のみ使用できます。

5.4 レジスタ内容を見る

アセンブリ言語レベルでデバッグを行う場合に、CPU の汎用レジスタの内容を簡単に見ることができます。これは、[レジスタ]ウィンドウを使用して行います。

5.4.1 レジスタウィンドウを開く

[レジスタ]ウィンドウを開くには、[表示->CPU->レジスタ]を選択するか、[レジスタ]ツールバーボタン をクリックします。[レジスタ]ウィンドウが開き CPU 汎用レジスタおよびその値(16 進数)を表示します。

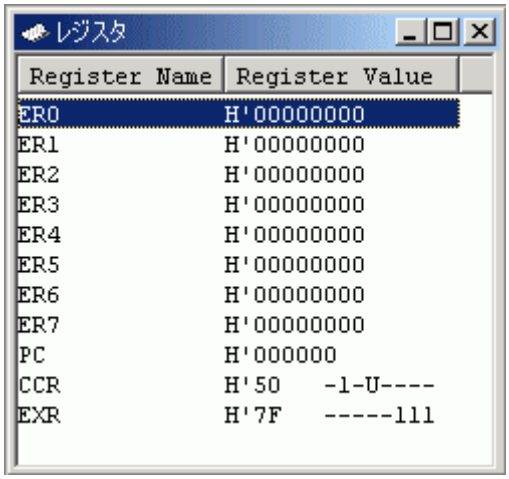


図 5-14 レジスタウィンドウ

5.4.2 ビットレジスタを拡張する

ビットレベルのフラグで制御するレジスタの場合、数値ではなく記号でビットの状態を表示します。また、そのレジスタをダブルクリックすると[レジスタ-[レジスタ名]]ダイアログを表示し、各ビット毎にオン/オフを設定できます。各ビットのチェックボックスをチェックしたときは 1 を、クリアしたときは 0 を設定します。

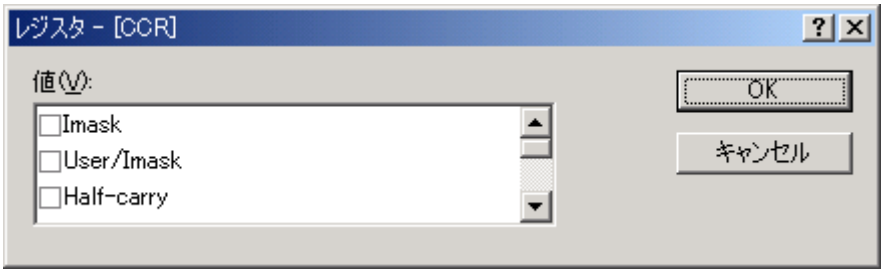


図 5-15 ビットレジスタの拡張

5.4.3 表示するレジスタを選択する

[レジスタ]ウィンドウに表示するレジスタを選択する場合は、ポップアップメニューから[設定...]を選択します。

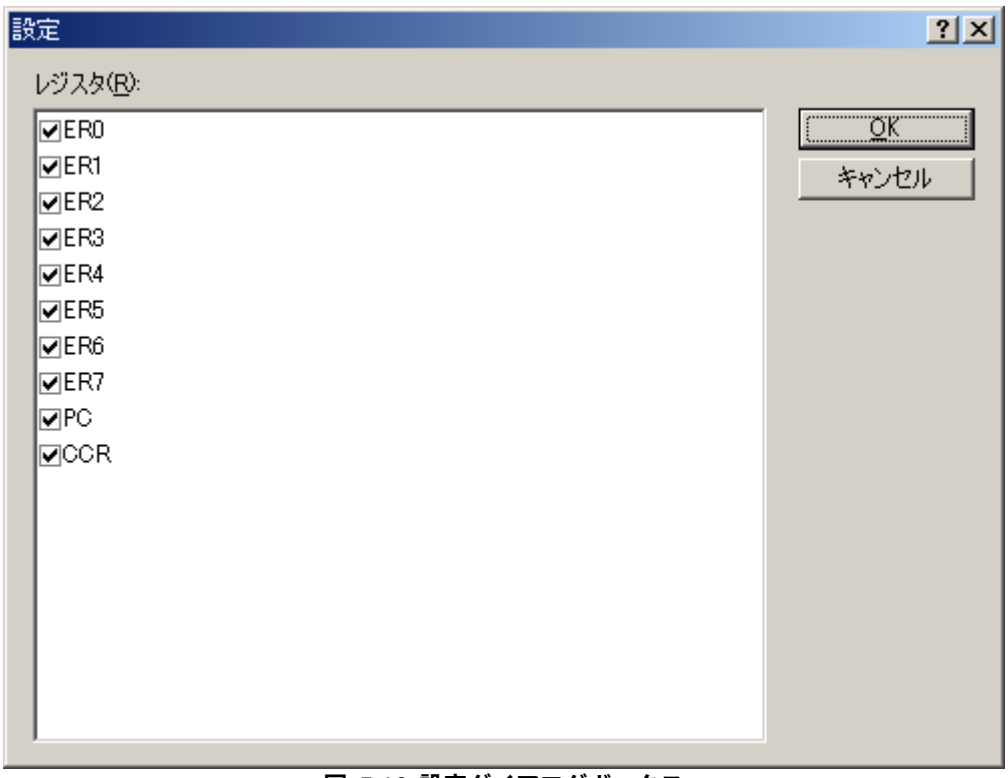


図 5-16 設定ダイアログボックス

5.4.4 ウィンドウを分割表示する

[レジスタ]ウィンドウを上下 2 分割で表示したいときは、ポップアップメニューから[分割]を選択後、分割バーを移動して分割します。

5.4.5 レジスタの内容を修正する

レジスタの内容を修正するには、以下のいずれかの方法でレジスタ編集ダイアログボックスを開きます。

修正したいレジスタをダブルクリックする。
修正したいレジスタを選択して、ポップアップメニューの[編集...]を選択する。

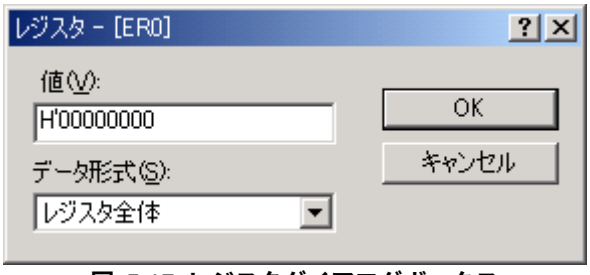


図 5-17 レジスタダイアログボックス

[値]フィールドには数字または C/C++の式を入力することができます。ドロップダウンリストからオプションを選択して、レジスタの内容のすべて、マスクした領域、フローティングビットまたはフラグビットを修正することができます(このリスト内容は、CPU のモデルおよび選択したレジスタによって異なります)。

新しい数字および式を入力したら[OK]ボタンをクリックするか”Enter”キーを押します。ダイアログボックスは閉じて、新しい値をレジスタに書き込みます。

5.4.6　レジスタの内容を使用する

[逆アセンブリ]または[メモリ]ウィンドウのアドレス指定など、HEW の別のところで値を入力する場合、CPU レジスタの中にある値を使用するためには、”#R1”、”#PC”、”#R6L”、または”#ER3”などのように、レジスタ名の先頭に “ # ” 記号をつけてください。

使用できるレジスタ名は、ご使用のターゲットマイコンにより異なります。

5.4.7　現在の表示内容をセーブする

現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアップメニューから[ファイルに保存...]を選択してください。

5.5　メモリを操作する

この節では、メモリ内容を操作する方法を説明します。操作の種類としては、色々なフォーマットでの表示、メモリブロックのフィル／移動、およびロードモジュールファイルのロード／ベリファイがあります。

5.5.1　メモリ領域を見る

メモリ領域を見るには、[表示->CPU->メモリ...]を選択するか、[メモリ]ツールバーボタン をクリックして[メモリ]ウィンドウを開きます。

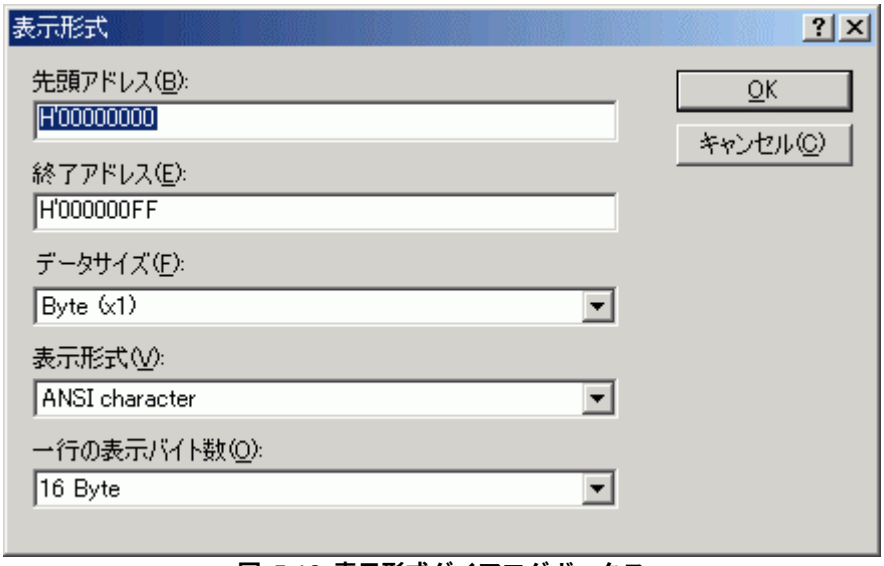


図 5-18 表示形式ダイアログボックス

[先頭アドレス] フィールドおよび[終了アドレス]フィールドに表示したい範囲をアドレス値または同等のシンボルで入力します。[データサイズ]および[表示形式]ドロップダウンリストから表示するデータサイズとフォーマットを選択します。また、[一行の表示バイト数]ドロップダウンリストから 1 行に表示するバイト数を選択します。[OK]ボタンをクリックするか、”Enter”キーを押すとダイアログボックスは閉じて[メモリ]ウィンドウが開きます。入力した先頭および終了アドレス内でスクロールすることができます。

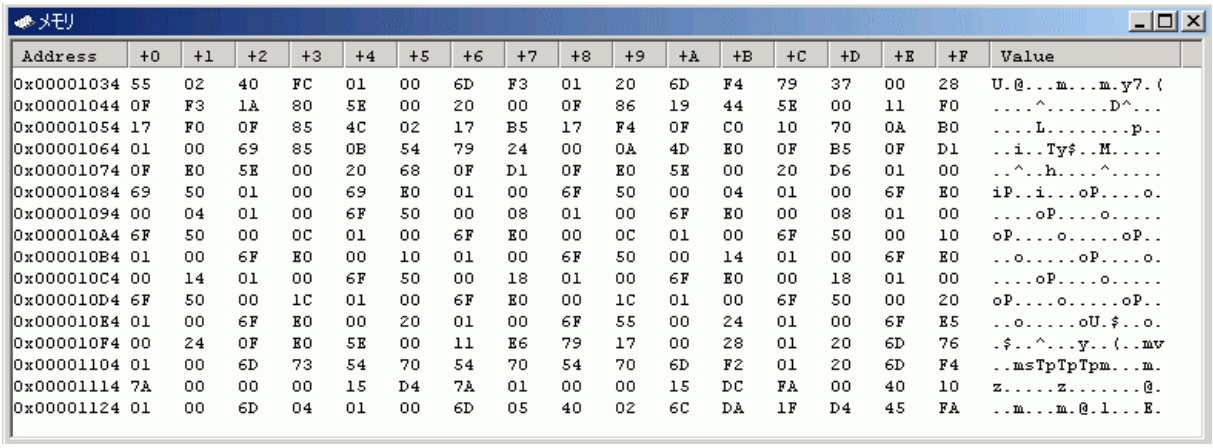


図 5-19 メモリウィンドウ

【注】 [+n]列の中の一組をメモリユニットと呼びます。

カラムを 3 つ表示します。

1. [Address] この行に表示したメモリデータの最初のアドレス
2. [+n] [Address]からのメモリデータで、nは行の最初のアドレスからのオフセット値
   デバッギングプラットフォーム物理メモリからアクセス幅でデータを読み出し、
   表示幅に変換します。
3. [Value] 他のフォーマットで表示するデータ

5.5.2 異なるフォーマットでデータを表示する

[メモリ]ウィンドウの表示フォーマットを変更する場合は、ポップアップメニューから[表示形式]を選択します。

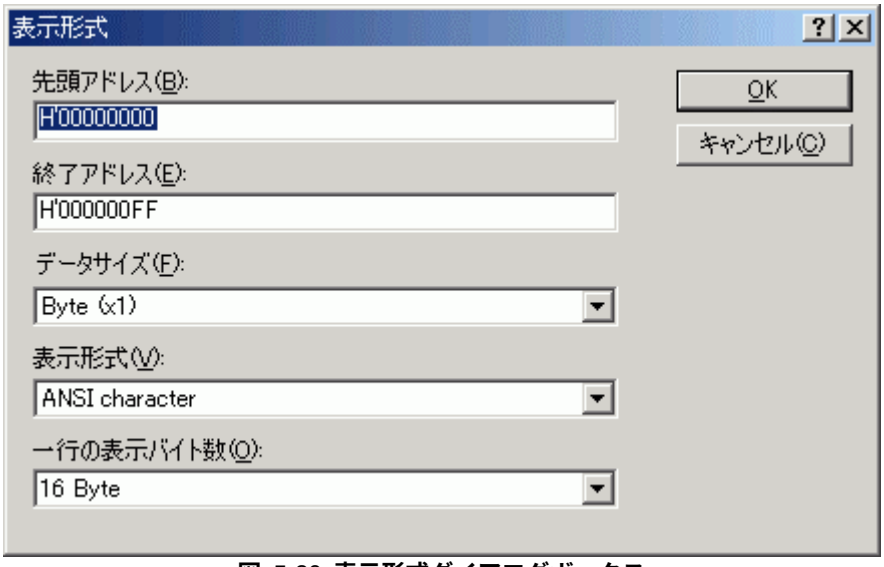


図 5-20 表示形式ダイアログボックス

メモリを異なる幅で表示して編集するには、[データサイズ]ドロップダウンリストを使用します。例えば、[Byte]オプションを選択すると、表示を更新して個々のバイトとしてメモリ領域を表示します。

データは、異なるフォーマットに変換することができます。これは[Value]カラムに表示します。フォーマットのリストは、データ選択に依存します。

1 行に表示するバイト数を変更するには、[一行の表示バイト数]ドロップダウンリストを使用します。例えば、[8 Byte]オプションを選択すると、1 行に 8 バイト分のデータを表示します。

5.5.3　ウィンドウを分割表示する

[メモリ]ウィンドウを上下 2 分割で表示したいときは、ポップアップメニューから[分割]を選択後分割バーを移動して分割します。

5.5.4　異なるメモリ領域を見る

[メモリ]ウィンドウの表示するメモリ領域を変更したいときは、スクロールバーを使用します。新しいアドレスをすぐに見たいときには、[表示形式]ダイアログボックスを使用します。これは、ポップアップメニューから[表示形式...]を選択することによって開くことができます。

新しいアドレスを入力して[OK]ボタンをクリックするか”Enter”キーを押します。ダイアログボックスは閉じ、[メモリ]ウィンドウの表示が新しいアドレスのデータに更新します。オーバーロード関数またはクラス名を入力すると、[関数選択]ダイアログボックスが開くので、関数を選択します。

5.5.5　メモリの内容を修正する

メモリ内容は、[メモリ編集]ダイアログボックスで変更します。変更したいメモリユニット上にカーソルを移動します（[メモリ]ウィンドウ表示選択にしたがって）。メモリユニットをダブルクリックするか、”Enter”キーを押します。

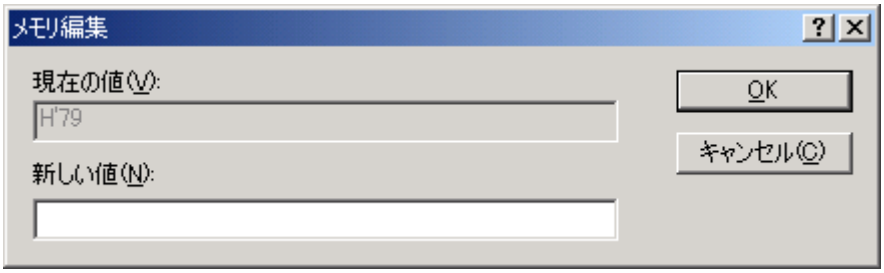


図 5-21 メモリ編集ダイアログボックス

[新しい値]フィールドには数字または C/C++の式を入力することができます。新しい数字または式を入力したら、[OK]ボタンをクリックまたは”Enter”キーを押すと、新しい値をメモリに書き込みます。

また、メモリユニット上にマウスカーソルを移動し、キーボードより 16 進数を入力することにより、メモリの内容を変更することもできます。

5.5.6　メモリ範囲を選択する

選択する範囲が[メモリ]ウィンドウにある場合、最初のメモリユニット（[メモリ]ウィンドウディスプレイでの選択に従い）をクリックして最後のユニットまでマウスをドラッグすることによって領域を選択することができます。選択した領域はハイライト表示します。

5.5.7 メモリ内の値を探す

メモリ内の値を探すには、[メモリ]ウィンドウを開き、ポップアップメニューから[検索...]を選択するか、[メモリ]ドロップダウンメニューの[検索...]を選択します。

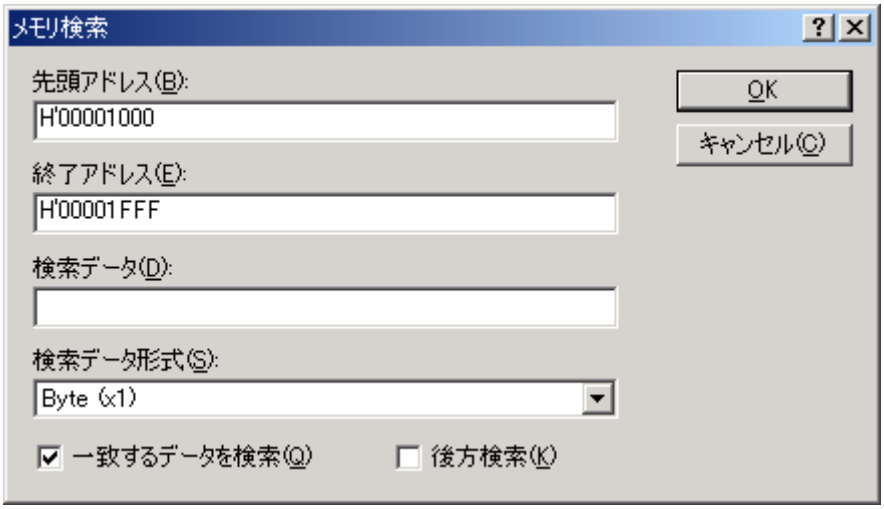


図 5-22 メモリ検索ダイアログボックス

検索するアドレス範囲の先頭および終了アドレス、および検索データ値を入力し、検索データ形式を選択します。検索データ形式に[パターン検索]を指定すると、最大 256 バイトのバイト列を検索できます。終了アドレスには先頭に'+'記号をつけることができ、この記号を入力すると先頭アドレス＋入力した値が終了アドレスになります。

また、パターン検索以外では検索条件として、データ一致/不一致、検索方向を指定できます。パターン検索はデータ一致および順方向のみの検索となります。

[OK]ボタンをクリックするか”Enter”キーを押します。ダイアログボックスは閉じて HEW は指定したデータの領域を検索します。データが見つかると、見つかったデータを反転表示します。

データを見つけることができなかった場合、[メモリ]ウィンドウの表示は以前と変わらず、データを見つけることができなかったことを知らせるメッセージボックスを表示します。

データが見つかった状態で、ポップアップメニューから[次を検索]を選択すると、次のアドレスから検索を続行します。

5.5.8 メモリ範囲に値をフィルする

メモリフィル機能を使って値をメモリアドレス範囲の内容に設定することができます。

同じ値でメモリ範囲に入れるには、[メモリ] ウィンドウのポップアップメニューの[フィル...]を選択するか、[メモリ]ドロップダウンメニューの[フィル...]を選択します。

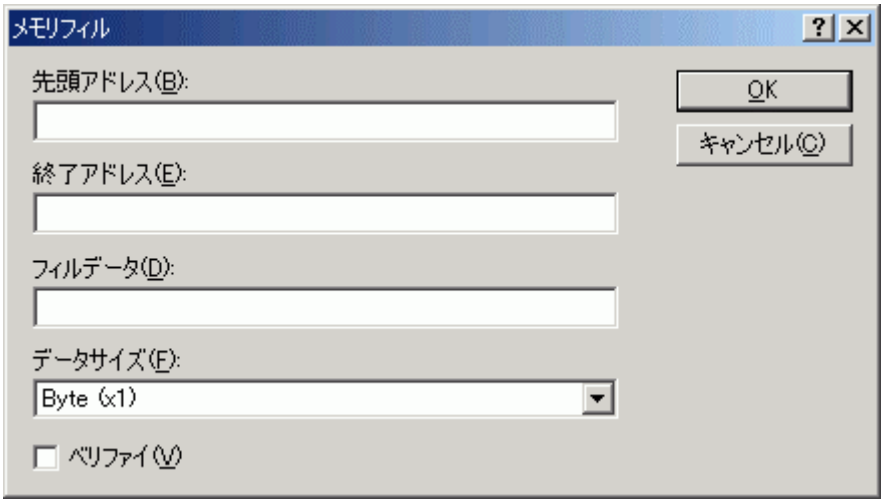


図 5-23 メモリフィルダイアログボックス

[メモリ]ウィンドウでアドレス範囲を選択した場合には、指定した開始および終了アドレスを表示します。

[データサイズ]ドロップダウンリストボックスからフォーマットを選択して[フィルデータ]フィールドにデータ値を入力します。[OK]ボタンをクリックするか”Enter”キーを押すと、ダイアログボックスが閉じて新しい値をメモリ領域に書き込みます。

5.5.9 メモリ領域をコピーする

メモリコピー機能を使用してメモリ領域をコピーすることができます。メモリ領域を選択してポップアップメニューから[コピー...]を選択するか、[メモリ]ドロップダウンメニューの[コピー...]を選択すると、[メモリコピー]ダイアログボックスを表示します。

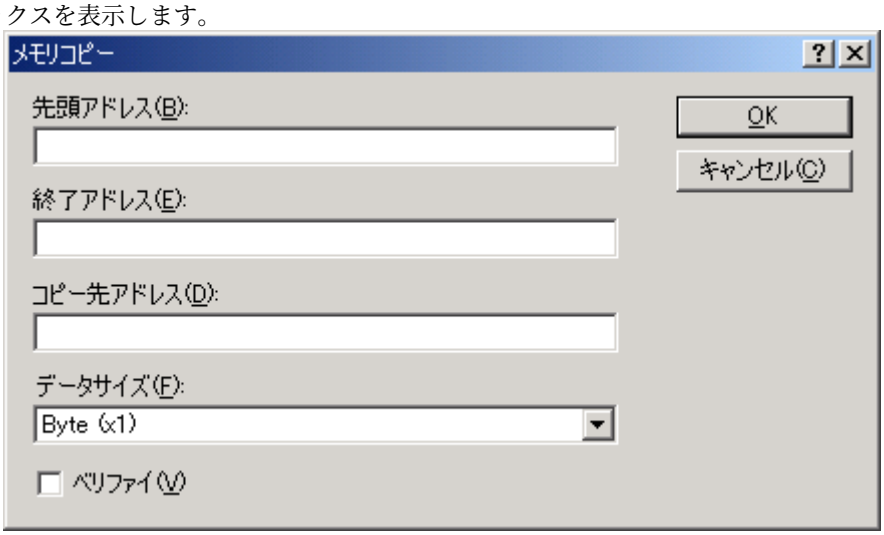


図 5-24 メモリコピーダイアログボックス

[メモリ]ウィンドウで選択したコピー元の開始アドレスおよび終了アドレスは、[先頭アドレス]および[終了アドレス]フィールドに表示します。[ベリファイ]チェックボックスをチェックすることによりコピー元とコピー先を比較しながらコピーすることもできます。[データサイズ]ドロップダウンリストボックスでコピー単位を選択することもできます。コピー先の開始アドレスを[コピー先アドレス]フィールドに入力して[OK]ボタンをクリックするか、”Enter”キーを押すと、ダイアログボックスを閉じてメモリブロックを新しいアドレスにコピーします。

5.5.10 メモリ領域を保存、検証する

メモリ保存機能を使用してアドレス空間のメモリ領域をディスクファイルに保存することができます。
[ファイル->メモリの保存...]を選択して[名前を付けて保存(メモリ)]ダイアログボックスを開きます。

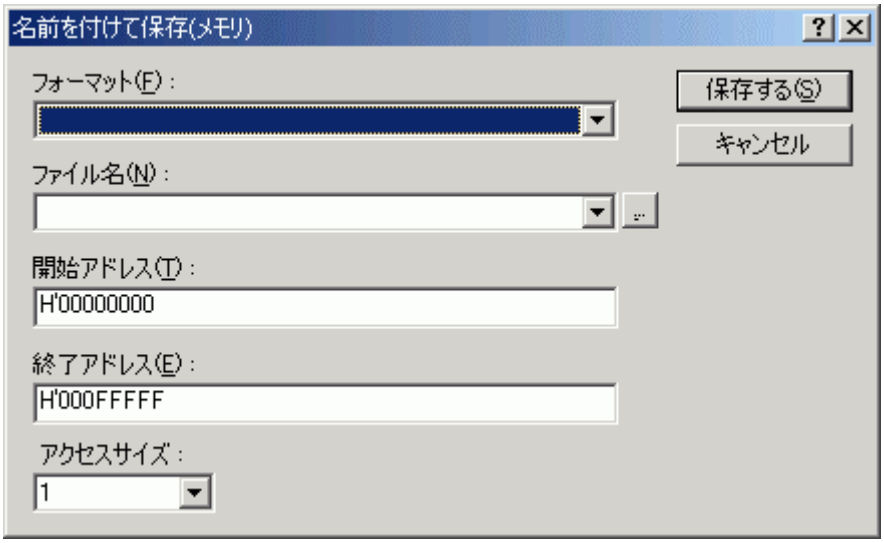


図 5-25 名前を付けて保存(メモリ)ダイアログボックス

保存するメモリブロックの開始および終了アドレス、ファイル名、ファイルフォーマットを入力します。[ファイル名]ドロップダウンリストボックスには、メモリを保存するために使用した過去 4 つのファイル名を表示します。

また[...]ボタンをクリックすると、標準の[名前を付けて保存]ダイアログボックスを開きます。[OK]ボタンをクリックするか”Enter”キーを押すと、ダイアログボックスを閉じて、メモリブロックを指定したフォーマットファイルとしてディスクに保存します。

[アクセスサイズ]ドロップダウンリストから、保存時のアクセスサイズを選択できます。

メモリベリファイ機能を使用してアドレス空間のメモリ領域を検証することができます。[ファイル->メモリのベリファイ...]を選択して[メモリベリファイ]ダイアログボックスを開きます。

[アクセスサイズ]ドロップダウンリストから、ベリファイ時のアクセスサイズを選択できます。

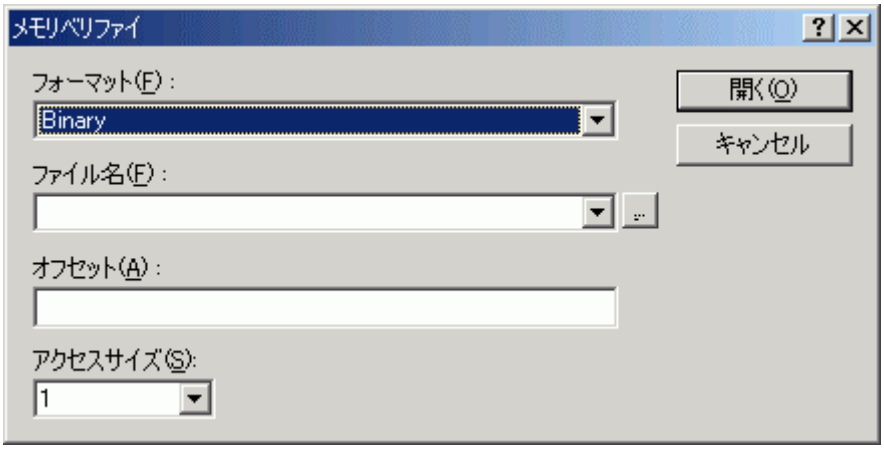


図 5-26 メモリベリファイダイアログボックス

5.5.11 ウィンドウ内容更新を抑止する

ユーザプログラム実行停止時などに、自動的に[メモリ]ウィンドウ内容を更新しないようにできます。
ポップアップメニューの[表示固定]をチェックします。[メモリ]ウィンドウの表示がグレー表示に変わります。

5.5.12　ウィンドウ内容を更新する

[メモリ]ウィンドウの内容を強制的にアップデートできます。
ポップアップメニューから[最新の情報に更新]を選択します。

5.5.13　メモリ内容を比較する

2 つのメモリブロック内容を比較することができます。メインメニューから[メモリ->比較...]を選択するか、[メモリ]ウィンドウのポップアップメニューから[比較...]を選択して[メモリ比較]ダイアログボックスを開きます。

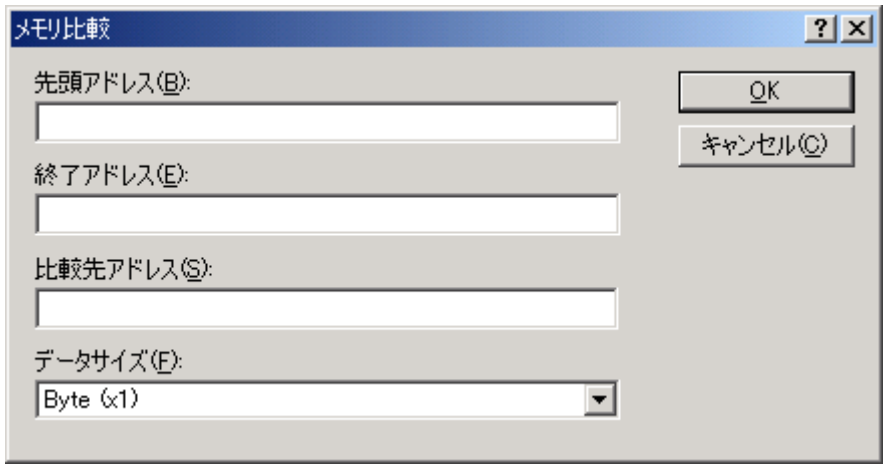


図 5-27 メモリ比較ダイアログボックス

比較フォーマット([データサイズ])、比較元メモリ領域の開始アドレス([先頭アドレス])と終了アドレス([終了アドレス])、および比較先メモリ領域の先頭アドレス([比較先アドレス])を入力します。[メモリ]ウィンドウでメモリブロックを反転表示していれば、ダイアログボックスを表示したときに開始アドレスと終了アドレスを自動的に設定します。
不一致個所があった場合は、そのアドレスをメッセージボックスに表示します。

5.5.14　メモリ領域をファイルからロードする

デバッグプラットフォームのメモリにファイルをロードできます。[メモリ]ウィンドウのポップアップメニューから[ロード...]を選択して、[ダウンロードプログラム]ダイアログボックスを開きます。

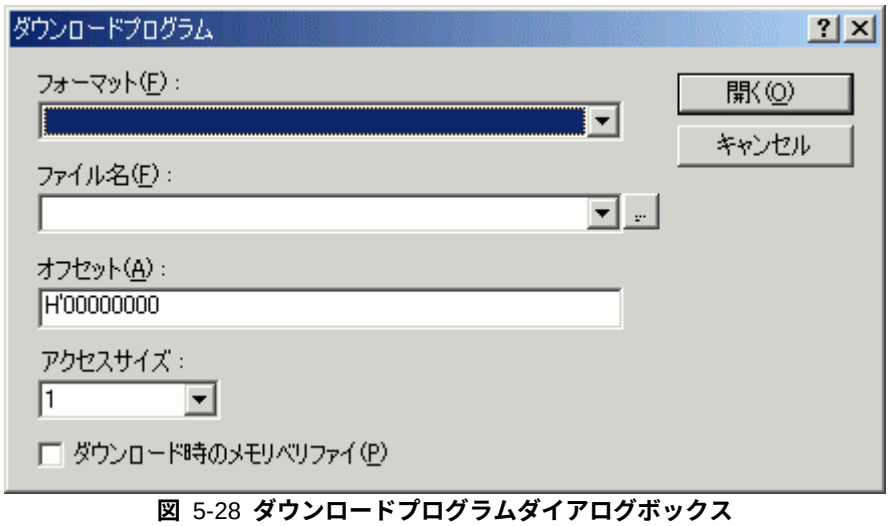


図 5-28 ダウンロードプログラムダイアログボックス

ファイルフォーマット([フォーマット])、ファイル名([ファイル名])を入力します。オフセットフィールド([オフセット])は、ロードアドレス値を変更するときはオフセット値、変更しないときは 0 を指定します。

5.6 I/O メモリを見る

マイクロコントローラには内蔵周辺モジュールがあります。デバイスによって周辺モジュールの数および型は異なりますが、代表的なモジュールとしては、DMA コントローラ、シリアルコミュニケーションインタフェース、A/D コンバータ、インテグレーテッドタイマユニット、バスステートコントローラおよびウォッチドッグタイマなどがあります。マイクロコントローラのアドレス空間にマッピングしたアクセスレジスタは、内蔵周辺モジュールを制御します。

[メモリ]ウィンドウは、連続したメモリアドレスのデータをバイト、ワード、ロングワード、単精度浮動小数点、倍精度浮動小数点、または ASCII 値として表示することがでます。これに対し I/O メモリは、非連続なメモリアドレスに異なるサイズでレジスタが割り付いているので、HEW はこれらのレジスタを簡単に確認したり設定したりすることができるように[IO]ウィンドウを提供します。

5.6.1 IO ウィンドウを開く

[IO]ウィンドウを開くには、[表示->CPU->IO]を選択するか、[IO の表示]ツールバーボタン をクリックします。内蔵周辺と一致するモジュールが I/O レジスタ情報を構成します。[IO]ウィンドウを最初に開くと、モジュール名の一覧表のみを表示します。

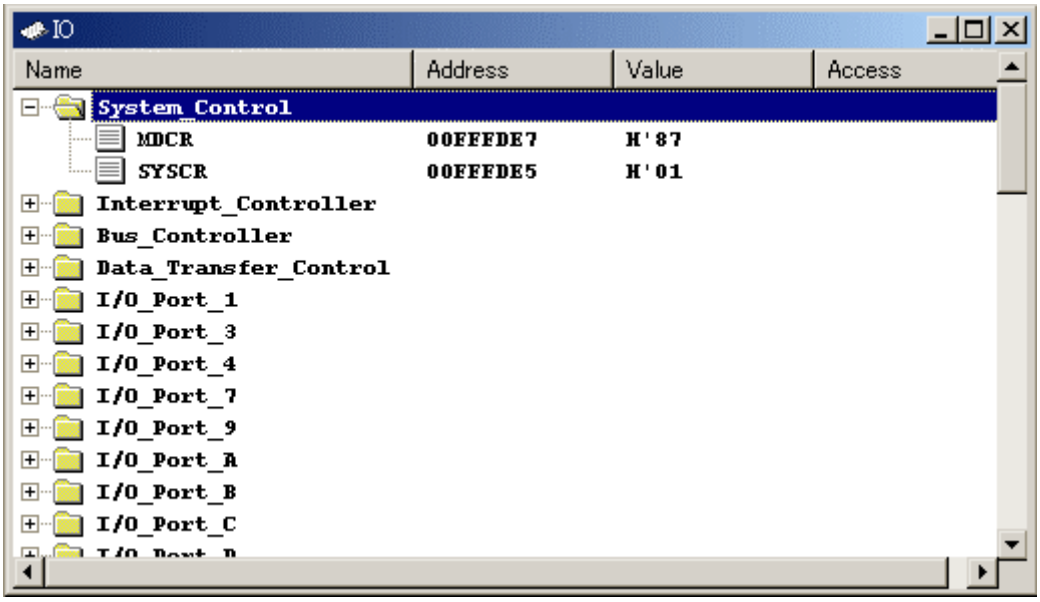


図 5-29 IO ウィンドウ

5.6.2 I/O レジスタ 表示を拡張する

I/O レジスタの名前、アドレス、および値を表示するには、モジュール名の左にある”＋”記号をクリックするか、クリックまたはカーソルを使用することによってモジュール名を選択し、”->”キーを押します。モジュールの表示が拡張し、その周辺モジュールのそれぞれのレジスタおよびその名前、アドレス、および値を表示します。拡張したモジュール名の左にある”－“記号をクリックするか、”<-”キーを押します。表示している I/O レジスタを閉じます。

ビットレベルで表示するには、レジスタ名の左にある”＋”記号をクリックするか、クリックまたはカーソルを使用することによってレジスタ名を選択し、”->”キーを押します。レジスタの表示が拡張し、レジスタビットの名前、および値を表示します。拡張したレジスタ名の左にある”－“記号をクリックするか、”<-”キーを押します。表示しているレジスタビットを閉じます。

5.6.3 I/O ファイルの手動ロード

I/O ファイルを手動でロードする場合は、[IO]ウィンドウ上での右クリックで表示されるポップアップメニューから[IO ファイルのロード...]を選択します。標準の[ファイルを探す]ダイアログボックスを表示します。ロードしたいファイルを選択して[OK]を選択します。これにより指定 I/O ファイルを[IO ウィンドウ]にロードします。I/O ファイルフォーマットは「付録 D I/O ファイルフォーマット」を参照してください。

5.6.4　I/O レジスタの内容を修正する

I/O レジスタの値を編集するには、ウィンドウに対して 16 進数を直接入力します。さらに複雑な式を入力するには、レジスタをダブルクリックするか”Enter”キーを押してレジスタの内容を修正するためのダイアログボックスを開きます。新しい数字または式を入力したら、[OK]ボタンをクリックするか”Enter”キーを押します。ダイアログボックスは閉じて新しい値をレジスタに書き込みます。

5.6.5　現在の表示内容をセーブする

現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアップメニューから[ファイルに保存]を選択してください。

5.7　現在の状態を表示する

デバッグプラットフォームの現在の状態を知るには[ステイタス]ウィンドウを表示します。

[ステイタス]ウィンドウを開くには、[表示->CPU->ステイタス]を選択するか、[ステイタスの表示]ツールバーボタン をクリックします。

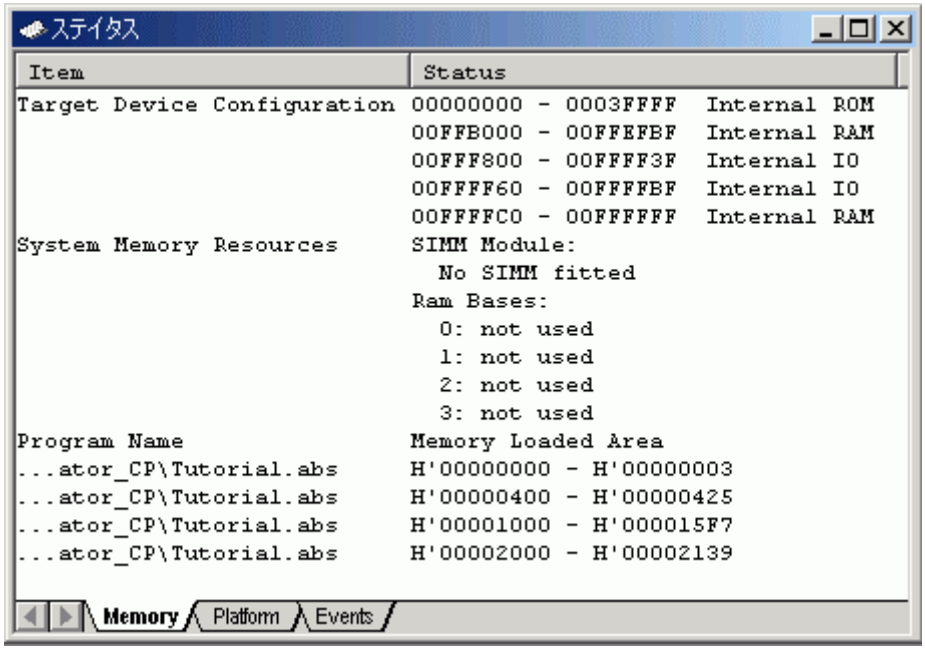


図 5-30 ステイタスウィンドウ

[ステイタス]ウィンドウには、3 枚のシートがあります。

[Memory]シート
メモリマッピングおよび現在ロードしたオブジェクト・ファイルが使用するメモリエリアなど、現在のメモリステータスに関する情報を含んでいます。

[Platform]シート
CPU種別および動作モードなど、デバッグプラットフォームのステータス情報、実行状態および実行統計情報を含んでいます。

[Events]シート
リソース情報およびブレークポイント等のイベント情報に関する情報を含んでいます。

【注】　本ウィンドウに表示する項目はご使用のエミュレータにより異なります。
詳細につきましては「8　本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.8 メモリ内容の日本語表示

5.8.1 メモリ内容を UNICODE 形式で表示する

メモリ内容を UNICODE 形式の日本語で表示する場合は[メモリ]ウィンドウを使用します。

[メモリ]ウィンドウに UNICODE 形式で表示するには、ポップアップメニューから[表示形式]を選択して[表示形式]ダイアログボックス(図 5-20参照)上で[データサイズ]に[Word]、[表示形式]に[Unicode character]を指定します。

5.8.2 メモリ内容を SJIS 形式または EUC 形式で表示する

メモリ内容を SJIS 形式または EUC 形式の日本語で表示および変更する場合は、[日本語メモリダンプ]ウィンドウを使用します。

(1) 日本語メモリダンプウィンドウを開く

[日本語メモリダンプ]ウィンドウを開くには[表示->CPU->日本語メモリダンプ...]を選択するか、[日本語メモリダンプ]ツールバーボタン をクリックして[日本語メモリダンプ設定]ダイアログボックスを開きます。

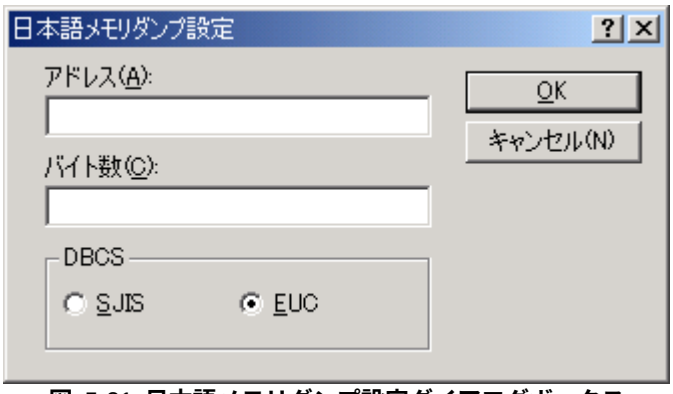


図 5-31 日本語メモリダンプ設定ダイアログボックス

本ダイアログでは下記項目を指定します。

[アドレス]	データのメモリ開始アドレスを指定します。（16 進表示）	
[バイト数]	データのバイト数を指定します。	
[DBCS]	データ形式を指定します。	
	[SJIS]	シフト JIS 形式で表示します。
	[EUC]	EUC 形式で表示します。

[日本語メモリダンプ設定]ダイアログボックスに設定後、[OK]ボタンをクリックすると[日本語メモリダンプ]ウィンドウが開きます。

図 5-32 日本語メモリダンプウィンドウ

(2)　日本語データを編集する

ポップアップメニューから[編集...]を選択すると、[文字列編集]ダイアログボックスが開きます。メモリ内容の日本語データを編集できます。

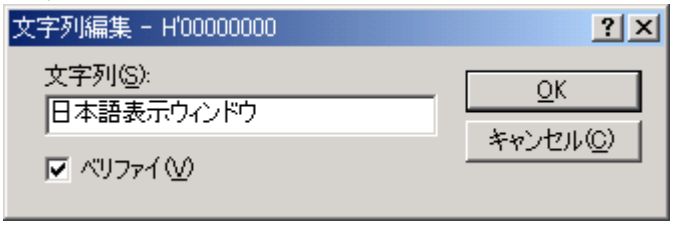


図 5-33 文字列編集ダイアログボックス

変更したい日本語データを入力して[OK]ボタンをクリックします。

(3)　日本語データを検索する

ポップアップメニューから[検索...]を選択すると、検索ダイアログボックスを開きます。日本語文字列を検索できます。

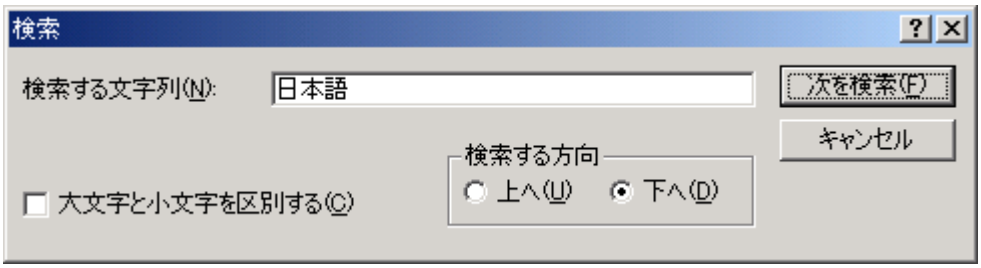


図 5-34 検索ダイアログボックス

検索したい日本語文字列をエディットボックスに入力し、[次を検索]ボタンまたは、”Enter”キーを押すと、ウィンドウ内で指定した文字列を検索します。一致した場合は強調表示します。
検索条件として文字列のほかに大文字/小文字の区別、および検索方向を指定できます。

(4)　表示形式を変更する

ポップアップメニューから[アドレス設定...]を選択すると、[アドレス設定]ダイアログボックスを開きます。ウィンドウに表示するメモリアドレスを変更できます。

(5)　表示バイト数を変更する

ポップアップメニューから[バイト数設定...]を選択すると、[バイト数設定]ダイアログボックスを開きます。ウィンドウに表示するバイト数を変更できます。

(6)　表示データ形式を変更する

ポップアップメニューから[EUC]または[SJIS]をチェックすると、ウィンドウに表示するデータ形式を変更できます。

5.9　エミュレータの情報を定期的に読み出し表示する

ユーザプログラム実行中/停止中にかかわらず変化するエミュレータの情報を知るには[拡張モニタ]ウィンドウを使用します。

【注】　拡張モニタ機能は E6000 エミュレータのハードウェア回路によりユーザシステムやエミュレータ内部の MCU から出力される信号をモニタするため、ユーザプログラムの実行に影響を与えることはありません。

5.9.1　拡張モニタウィンドウを開く

[拡張モニタ]ウィンドウを開くには、[表示->CPU->拡張モニタ]を選択するか、[拡張モニタ]ツールバーボタン をクリックします。表示項目の更新間隔は、ユーザプログラム実行中は約 100ms、ブレーク中は約 1000ms です。

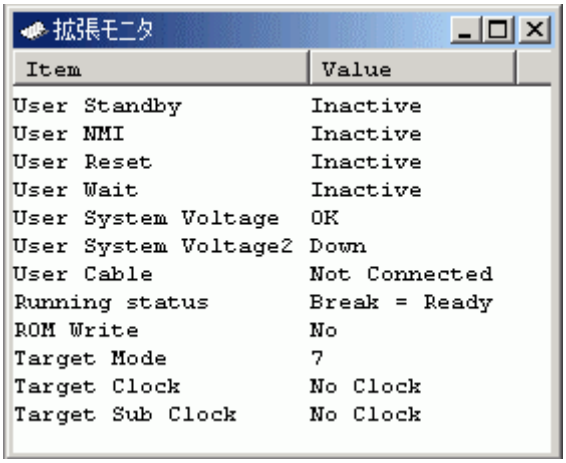


図 5-35 拡張モニタウィンドウ

5.9.2　表示項目を選択する

ポップアップメニューから[プロパティ...]を選択すると[拡張モニタコンフィギュレーション]ダイアログボックスを表示します。

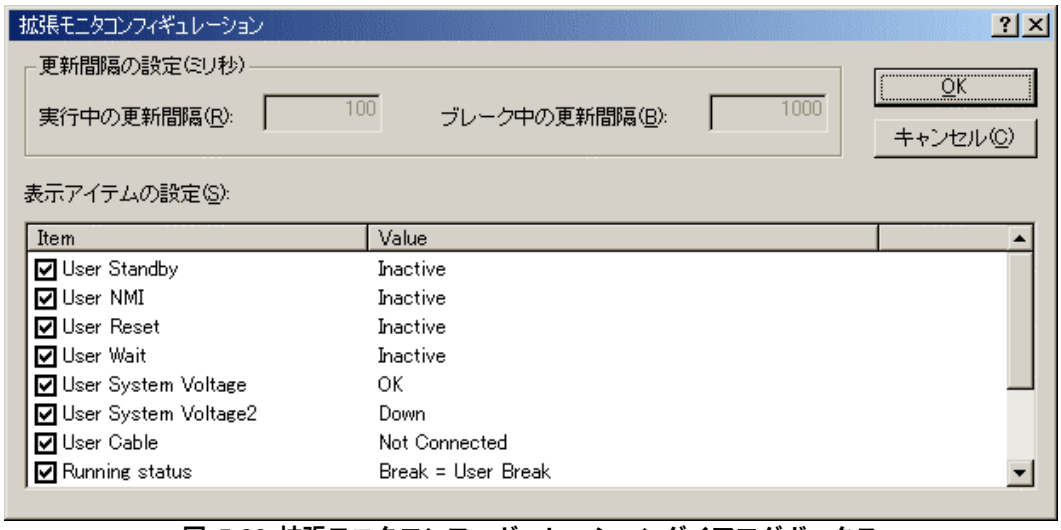


図 5-36 拡張モニタコンフィギュレーションダイアログボックス

[拡張モニタ]ウィンドウに表示する各項目を設定できます。

【注】　本ウィンドウに表示する項目はご使用のエミュレータにより異なります。
詳細につきましては「8　本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.10　リアルタイムにメモリ内容を表示する

ユーザプログラム実行中にメモリ内容をモニタするには[モニタ]ウィンドウを使用します。

モニタ機能は E6000 エミュレータのバスモニタ回路により MCU 内部のリード/ライト信号をトリガとしてアドレスバスおよびデータバスの値を保持し、該当するメモリの表示内容を更新するためリアルタイム性は損なわれません。

E6000 エミュレータのバスモニタ回路に実装されたモニタチャネル(8 チャネル)を使用し、最大 8 ポイント設定できます。

1 ポイントあたりのモニタ可能サイズは 1～256 バイトです。

各ポイントのモニタ範囲の一部または全部が重複する設定も可能です

【注】　MCU に内蔵のタイマカウンタなど、値の更新に MCU 内部のリード/ライト信号が発生しないエリアに対してはモニタできません。

5.10.1　モニタウィンドウを開く

[モニタ]ウィンドウを開くには、[表示->CPU->モニタ->モニタ設定...]を選択するか、[モニタ]ツールバーボタン をクリックして[Monitor Setting]ダイアログボックスを開きます。

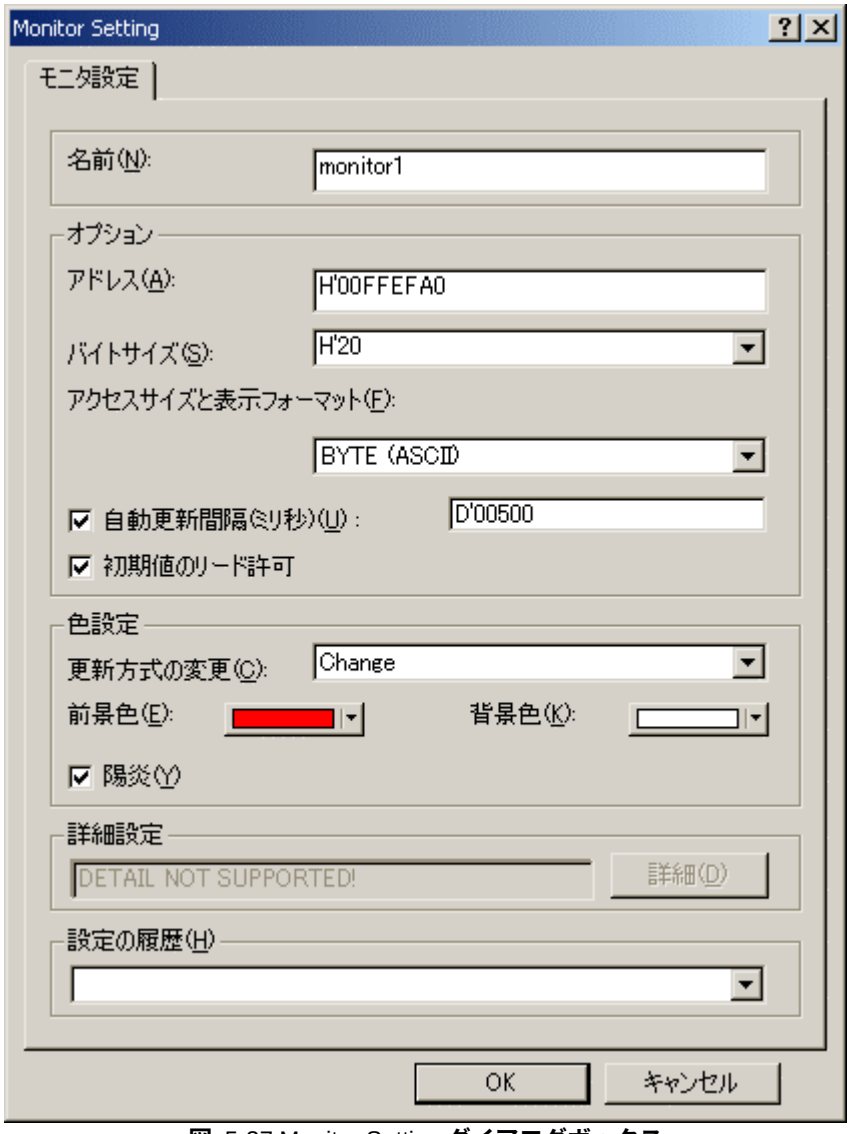


図 5-37 Monitor Setting **ダイアログボックス**

[名前] モニタウインドウの名称を設定します。
[オプション] モニタ条件を設定します。
[アドレス] モニタを行う先頭アドレスを設定します。
[バイトサイズ] モニタを行う範囲を設定します。
[アクセスサイズと表示フォーマット] モニタウインドウに表示するアクセスサイズを設定します。
[自動更新間隔(ミリ秒)] モニタ取得間隔を設定します。(最小値は 500ms)
[初期値のリード許可] モニタウインドウ OPEN 時に、モニタ表示エリアの値をリードします。
[色設定] モニタの更新方法および色属性を設定します
[更新方法の変更] モニタ中に変更があった値をどのように表示するかを設定します。
([初期値のリード許可]選択時有効)
No change: 色の変更は行いません。
Change: 色を変更します。
色は前景色オプション、背景色オプション、で設定します。
Gray: 値の変更のないデータを灰色表示します。
Appear: 値の変更があると表示します。変更なければ表示しません。
[前景色] 表示文字色を設定します。
([Change]選択時有効)
[背景色] 背景色を設定します。
([Change]選択時有効)
[陽炎] チェックボックスにチェックがある場合、一定間隔更新のないデータの色を背景色オプションで設定した色に戻します。一定間隔とは、モニタ取得間隔の一回分です。
([Change], [Gray], [Appear]選択時有効)
[詳細設定] エミュレータ固有の項目を設定します。本エミュレータでは設定できません。
[設定の履歴] 前回の設定内容を呼び出します。

【注】 1. **本エミュレータではモニタを行う先頭アドレスとして奇数アドレスは指定できません。**
2. **前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。**

設定完了後、[OK]ボタンをクリックすると[モニタ]ウィンドウが開きます。

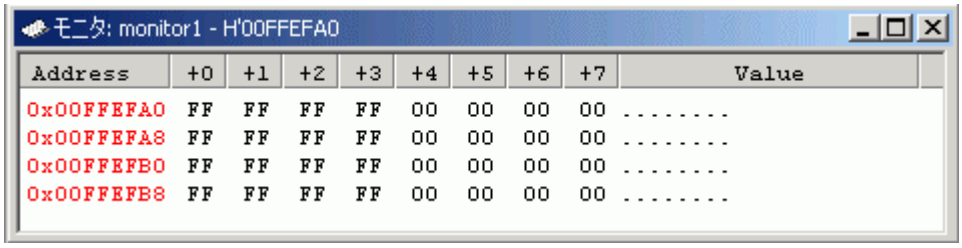


図 5-38 モニタウィンドウ

ユーザプログラム実行中、自動更新間隔の設定値に応じて表示を更新します。

【注】 アドレス変更時またはメモリ内容変更時、データ内容が正しく表示されない場合は、ポップアップメニューより [最新の情報に更新]を選択してください。

5.10.2　モニタの設定内容を変更する

変更したい[モニタ]ウィンドウのポップアップメニューより[モニタ設定...]を選択すると、[Monitor Setting]ダイアログボックスが開き、設定内容を変更することができます。
また、ポップアップメニューの[色設定]メニューおよび[アクセスサイズと表示フォーマット]メニューより簡単に色設定およびアクセスサイズと表示フォーマットを変更できます。

5.10.3　モニタの更新を一時的に停止する

ユーザプログラム実行中、[モニタ]ウィンドウは設定した自動更新間隔にしたがって自動的に表示を更新します。
表示更新を停止させたい[モニタ]ウィンドウのポップアップメニューより[表示固定]を選択してください。
アドレス部の表示文字が黒色となり、表示更新を停止します。
再びポップアップメニューより[表示固定]を選択することにより停止状態は解除できます。

5.10.4　モニタ設定を削除する

削除したい[モニタ]ウィンドウのポップアップメニューより[閉じる]を選択すると、[モニタ]ウィンドウを閉じ、モニタ設定を削除します。

5.10.5　変数の内容をモニタする

任意の変数の値を参照するには、ウォッチウィンドウを使用します。
[ウォッチ]ウィンドウに登録した変数のアドレスが、モニタ機能で設定したモニタ範囲に存在する場合、該当する変数の値をモニタ機能により更新し表示することができます。
この機能によりリアルタイム性を損なわずに変数の内容を確認できます。
[ウォッチ]ウィンドウについては「5.15.3 ウォッチウィンドウ」を参照してください。

5.10.6　モニタウィンドウを非表示にする

モニタ機能を使用し、[ウォッチ]ウィンドウより変数の値をモニタする場合、[モニタ]ウィンドウを非表示にしておくと画面を有効に活用できます。
現在設定しているモニタ情報は[表示->CPU->モニタ]のサブメニューとしてリストされます。
モニタ設定リストは[モニタ]ウィンドウ名およびモニタ開始アドレスで構成されています。
リストの左側にチェックがある場合は該当の[モニタ]ウィンドウが表示されていることを示します。

モニタ設定リストより非表示にしたい[モニタ]ウィンドウ項目を選択すると、該当の[モニタ]ウィンドウが非表示となり、リストの左側にあったチェックマークが消えます。
非表示にした[モニタ]ウィンドウを再び表示するにはモニタ設定リストより非表示にした[モニタ]ウィンドウ項目を選択してください。

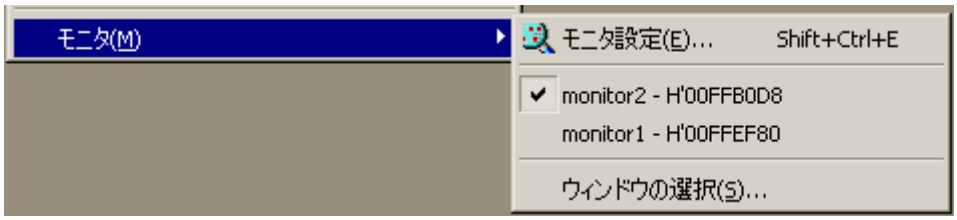


図 5-39 モニタ設定リスト

5.10.7 モニタウィンドウを管理する

[表示->CPU->モニタ->ウィンドウの選択...]を選択すると表示される､[ウィンドウの選択]ダイアログボックスより、現在設定されているモニタ条件の確認、新規モニタ条件の追加、編集、削除などの操作を連続的に行うことができます。

また、現在設定されているモニタ条件を複数選択することにより、更新の一時停止、非表示、削除を一括して操作できます。

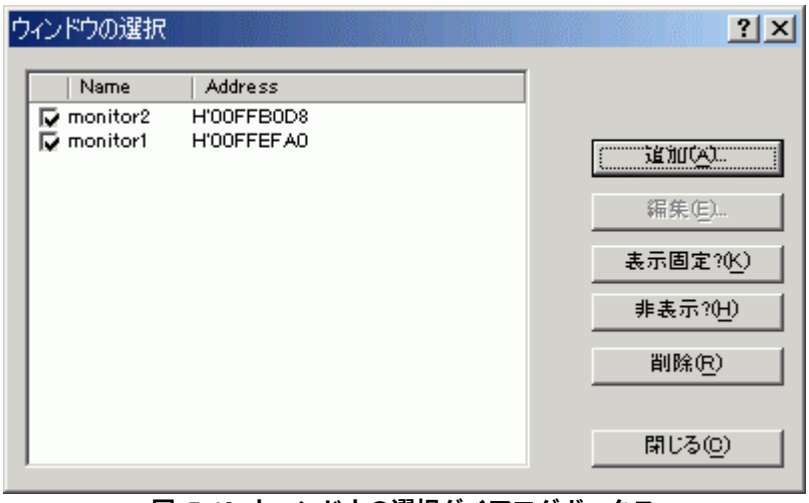


図 5-40 ウィンドウの選択ダイアログボックス

[追加]	新規にモニタ条件を追加します。
[編集]	選択している[モニタ]ウィンドウの設定を変更します。 (複数選択時無効)
[表示固定 / 表示固定解除]	選択している[モニタ]ウィンドウの表示を自動更新または更新停止にします。
[表示 / 非表示]	選択している[モニタ]ウィンドウを表示または非表示にします。
[削除]	選択しているモニタ条件を削除します。
[閉じる]	本ダイアログボックスを閉じます。

5.11　ラベルを見る

デバッグ情報を含んだユーザプログラムをロードした時に、ラベルを同時に登録します。ラベルを追加することも可能です。

[逆アセンブリ]ウィンドウにおいて、対応するアドレスの代わりとして、また命令オペランドの一部として、ラベルの最初の 8 文字を表示します。

【注】　オペランドとラベルの値が一致すれば、命令のオペランドはラベル名に置き換わります。同じ値を持つラベルが 2 つ以上ある場合、アルファベット順で先に来るラベルを表示します。

5.11.1　ラベルを一覧にする

現在のデバッガセッションに定義したラベルのすべてを見るには、[表示->シンボル->ラベル]を選択するか、[ラベルの表示]ツールバーボタン をクリックします。

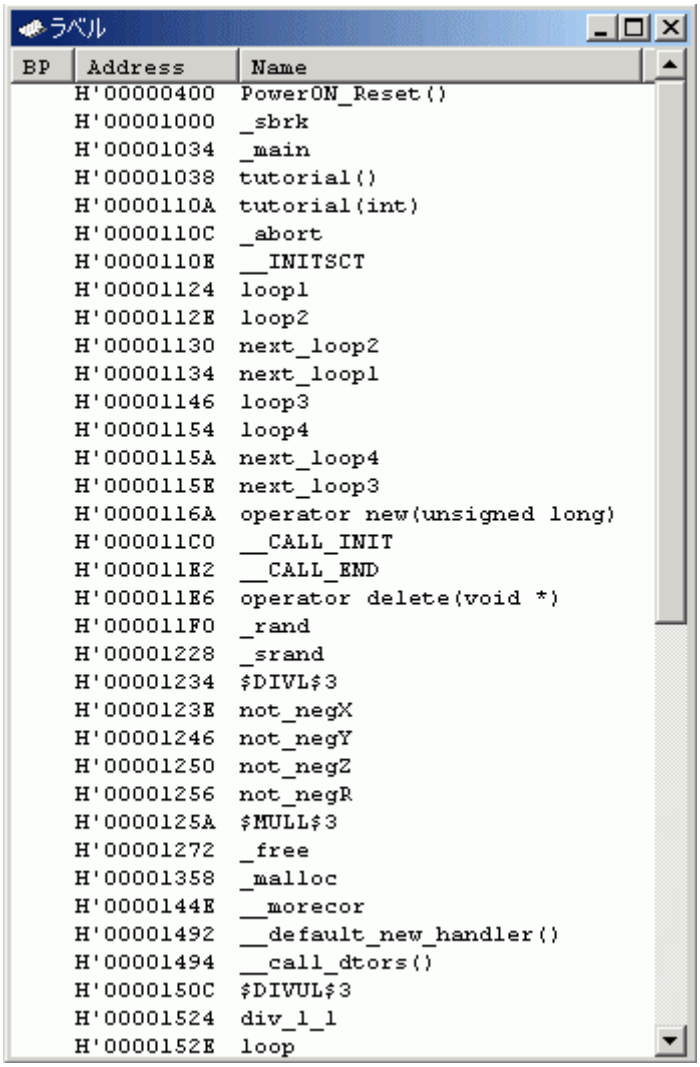


図 5-41 ラベルウィンドウ

それぞれのカラムのヘッダをクリックすることによりアルファベット順（ASCII コードによって）またはアドレス値でソートしたシンボルを表示させることができます。

[BP]カラムをダブルクリックすることにより関数の入り口でソフトウェアブレークポイントをすばやく設定したり解除したりすることができます。

5.11.2 ラベルを追加する

ラベルを追加するには、ポップアップメニューから[追加...]を選択して、[ラベル追加]ダイアログボックスを表示します。

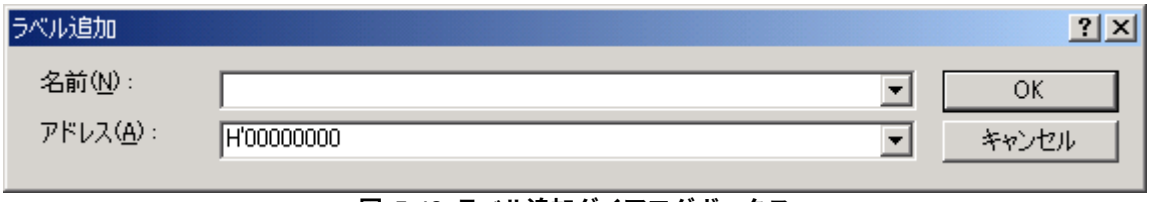


図 5-42 ラベル追加ダイアログボックス

新しいラベル名を[名前]フィールドに入力し、対応する値を[アドレス]フィールドに入力して[OK]ボタンを押します。[ラベル追加]ダイアログボックスがクローズし、ラベルリストに新しいラベルを追加、更新します。多重定義関数やクラス名を入力したときは、[関数選択]ダイアログボックスが開くので、関数を選択して[アドレス]フィールドを設定します。詳細は「5.14.3 複数ラベルをサポートする」を参照してください。

5.11.3 ラベルを編集する

ラベルを編集するにはポップアップメニューから[編集...]を選択します。[ラベルの編集]ダイアログボックスを表示します。

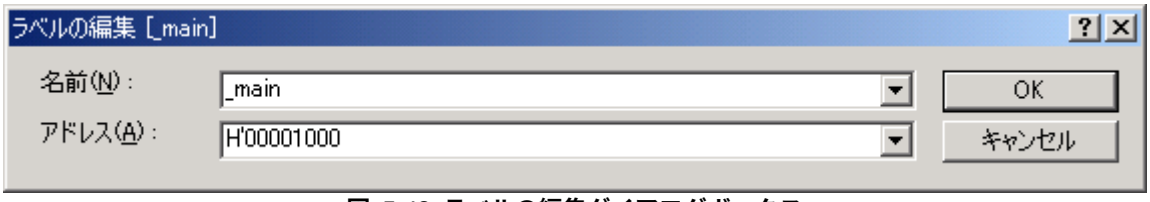


図 5-43 ラベルの編集ダイアログボックス

ラベル名と対応する値を編集して、[OK]ボタンを押すとラベルリストに編集を反映し、保存します。多重定義関数やクラス名を入力したときは、[関数選択]ダイアログボックスが開くので、関数を選択して[アドレス]フィールドを設定します。詳細は「5.14.3 複数ラベルをサポートする」を参照してください。

5.11.4 ラベルを削除する

削除したいラベルを選択した状態で、ポップアップメニューから[削除]を選択します。この際、図 5-44に示す確認メッセージボックスを表示します。

図 5-44 ラベル削除確認メッセージボックス

[はい]ボタンを押すとラベルリストから削除し、ウィンドウを更新します。メッセージボックスの表示が不要のときは、[今後表示しない]チェックボックスをチェックしてください。

5.11.5　すべてのラベルを削除する

ポップアップメニューから[すべてを削除]を選択すると、リストからすべてのラベルを削除します。この際、図5-45に示す確認メッセージボックスを表示します。

図 5-45 全ラベル削除確認メッセージボックス

[はい]ボタンを押すと、すべてのラベルを HEW のシンボルテーブルから削除し、リスト表示もクリアします。メッセージボックスの表示が不要のときは、[今後表示しない]チェックボックスをチェックしてください。

5.11.6　ラベルをファイルからロードする

シンボルファイルをロードして現在の HEW のシンボルテーブルに結合できます。ポップアップメニューから[ロード...]を選択すると、[ファイルを開く]ダイアログボックスをオープンします。

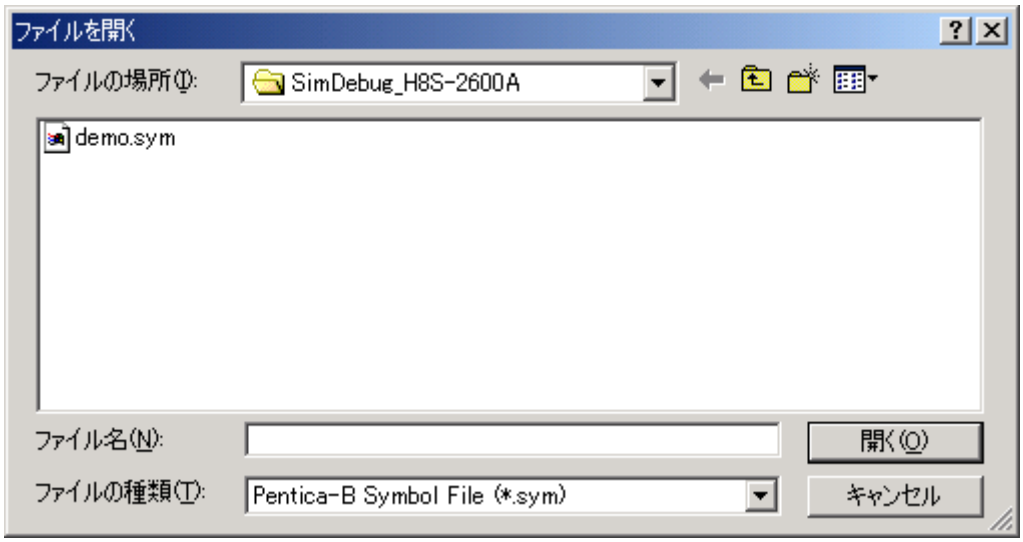


図 5-46 ファイルを開くダイアログボックス

ダイアログボックスは、Windows®標準の[ファイルを開く]ダイアログボックスと同様です。ファイルを選択し、[開く]ボタンを押すとロードを開始します。シンボルファイルの標準拡張子は".sym"です。

5.11.7　ラベルをファイルに保存する

ポップアップメニューから[名前を付けて保存...]を選択すると、[名前を付けて保存]ダイアログボックスを表示します。[名前を付けて保存]ダイアログボックスは Windows®標準の[名前を付けて保存]ダイアログボックスと同様に操作できます。[ファイル名]フィールドにファイル名を入力し[保存]ボタンを押すとシンボルファイルにラベルリストを保存します。標準ファイル拡張子は".sym"です。

フォーマットが「付録 E　シンボルファイルフォーマット」にあるので参照してください。

一度[名前を付けて保存...]メニューでファイルに保存すると、以後はポップアップメニューの[上書き保存]で、現在のシンボルテーブルを同一シンボルファイルに保存できます。

5.11.8 ラベルを検索する

ポップアップメニューから[検索...]を選択すると、[ラベルの検索]ダイアログボックスを表示します。

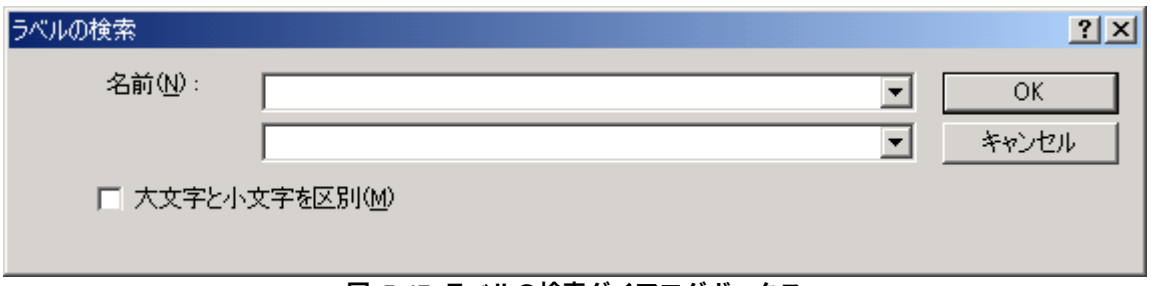


図 5-47 ラベルの検索ダイアログボックス

検索したいラベル名の一部、全部をエディットボックスに入力し、[OK]ボタンまたは、”Enter”キーを押すと、指定した文字列を含んでいるテキストファイルをサーチします。

【注】 ラベルは、はじめの 1024 文字分しか情報を保持していません。したがって、ラベル名のはじめの 1024 文字分は重複しないようにしてください。ラベルは、大文字、小文字を区別します。

5.11.9 次を検索する

ラベルが検索できた後、ポップアップメニューから[次を検索]を選択すると、検索条件に一致する次のラベルを検索します。

5.11.10 ラベルに対応するソースプログラムを表示する

ラベルを選択した状態で、ポップアップメニューから[ソースを表示]を選択すると、ラベルのアドレスに対応する[ソース]または、[逆アセンブリ]ウィンドウをオープンします。

5.12　プログラムを実行する

作成したプログラムコードの実行方法について説明します。ここでは、プログラムを連続して実行させたり、シングルステップ実行を行ったり、同時に複数の命令を実行させたりします。

5.12.1　リセットから実行を開始する

ターゲットマイコンをリセットしてリセットベクタアドレスプログラムを実行させるには、[デバッグ->リセット後実行]を選択するか、[リセット後実行]ツールバーボタン をクリックします。

プログラムは、ブレークポイントにヒットするまで、またはブレーク条件が成立するまで実行を続けます。プログラムの実行は手動で停止することができます。その方法としては、[デバッグ->プログラムの停止]を選択するか[停止]ツールバーボタン をクリックします。

【注】　プログラムはリセットベクタ位置に格納したアドレスから実行を開始します。したがって、この位置に自分のスタートアップコードのアドレスを含んでいることを確認することが重要です。

5.12.2　実行を継続する

作成したプログラムが停止すると、HEW は、CPU の現在のプログラムカウンタ(PC)値に対応する[エディタ]および[逆アセンブリ]ウィンドウの行の左余白に黄色の矢印を表示します。ステップ実行を行った場合、または実行を続けた場合、この命令を次に実行します。

現在の PC アドレスから実行を継続するには、[実行]ツールバーボタン をクリックするか、[デバッグ->実行]を選択します。

前回停止時と異なるアドレスから実行を継続するには、下記の方法で PC を変更後に[実行]ツールバーボタン をクリックするか、[デバッグ->実行]を選択します。

- [レジスタ]ウィンドウ上で変更する。詳しくは、「5.4.5　レジスタの内容を修正する」を参照してください。
- [ソース]ウィンドウまたは[逆アセンブリ]ウィンドウ上で、テキストカーソル(マウスカーソルではありません)を変更したい行に移動してポップアップメニューから[カーソル位置に PC 値を設定]を選択する。

5.12.3　カーソルまで実行する

アプリケーションを実行している途中で、シングルステップ実行を複数回行うだけの比較的小さいセクションコードのみを実行したいと考える場合があります。これは、カーソル位置まで実行機能を使用して行うことができます。

➲ カーソル位置まで実行を使用するには

1. [ソース]または[逆アセンブリ]ウィンドウが開いていて、プログラムを停止するアドレスを表示していることを確認します。
2. アドレスフィールドをクリックするかカーソルキーを使用してプログラムを停止するアドレス上にテキストカーソルを置きます。
3. ポップアップメニューから[カーソル位置まで実行]を選択します。

デバッギングプラットフォームは作成したコードを現在の PC 値から実行し、カーソル位置が示すアドレスまで実行します。

【注】 1.　作成したプログラムがこのアドレスのコードを決して実行しない場合、プログラムは停止しません。その場合、コードの実行を中止するには、"Esc"キーを押すか、[デバッグ ->プログラムの停止]を選択するか、[停止]ツールバーボタン をクリックします。

2.　カーソル位置まで実行機能は、PC ブレークポイント機能を利用しています。そのため、すでに PC ブレークポイントが最大数設定してある場合は、本機能は使用できません。

5.12.4 開始アドレスを指定して実行する

[プログラム実行]ダイアログボックスを利用すると、任意のアドレスから命令を実行することができます。
[プログラム実行]ダイアログボックスを開くには、[デバッグ->ラン...]を選択します。

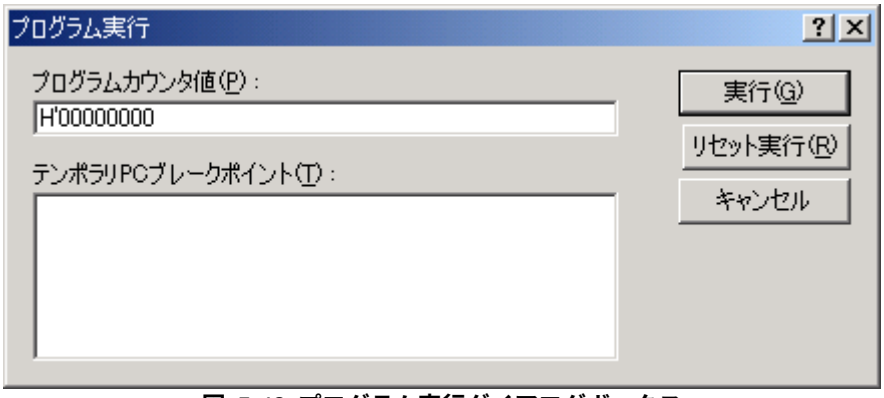


図 5-48 プログラム実行ダイアログボックス

本ダイアログボックスでは命令実行の条件を設定します。

[プログラムカウンタ値]	実行を開始する命令アドレスを設定します。 初期値は現在の PC 値となります。
[テンポラリ PC ブレークポイント]	一時的な PC ブレークポイントを設定します。 本ダイアログボックスによる命令実行が停止するとこのブレークポイントは解除されます。

【注】 [テンポラリ PC ブレークポイント]は PC ブレークポイント機能を利用しています。既に PC ブレークポイントを最大数利用している場合は本機能を利用できません。

[実行]ボタンをクリックすると、設定した内容に従って命令を実行します。
[リセット実行]ボタンをクリックすると、リセットベクタから命令を実行します。
[キャンセル]ボタンをクリックすると、命令を実行しないで、本ダイアログボックスを閉じます。

5.12.5 シングルステップ

作成したコードをデバッグするために、一度に一行だけまたは一つの命令だけステップ実行して、この命令がシステムにどのように影響するかを確認したい場合、[ソース]ウィンドウでは、ソースライン一行だけをステップ実行します。
[逆アセンブリ]ウィンドウにおいては、アセンブリ言語命令単位にステップ実行します。命令が他の関数またはサブルーチンをコールした場合、オプションでその関数にステップインまたはステップオーバーすることができます。その命令コールを行わない場合には、いずれのオプションでも、デバッガに命令を実行させ、次の命令で停止させることができます。

(1) 関数にステップイン実行する

関数にステップイン実行することを選択した場合にはデバッガは関数の行または命令でコールを実行します。関数にステップインするには、[ステップイン]ツールバーボタン をクリックするか、[デバッグ->ステップイン]を選択します。

(2) 関数コールをステップオーバ実行する

関数をステップオーバ実行することを選択した場合には、デバッガはコールおよび関数内のすべてのコード(および関数が行う可能性のある関数コールのすべて)を実行して、呼び出し元の関数の次の行または命令で停止します。関数をステップオーバするには、[ステップオーバ]ツールバーボタン をクリックするか、[デバッグ ->ステップオーバ]を選択します。

(3) 関数からステップアウト実行する

関数内の確認したい命令の実行が終了した場合や、誤って関数にステップインした場合に、ステップアウト機能を使用すると関数内の残りのコードをステップ実行せずに呼び出し元の関数に戻ることができます。

現在の関数からステップアウトするには、[ステップアウト]ツールバーボタンをクリックするか、[デバッグ->ステップアウト]を選択します。

5.12.6 複数のステップ

[プログラムステップ]ダイアログボックスを使用することにより、一度に複数のステップ実行ができます。このダイアログボックスでは、ステップ間の時間差を選択し、ステップ実行を自動的に行うよう設定できます。このダイアログボックスは、[デバッグ -> ステップ...]を選択して開きます。

[プログラムステップ]ダイアログボックスを表示します。

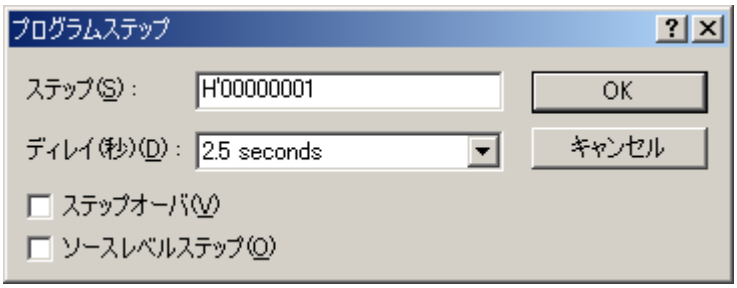


図 5-49 プログラムステップダイアログボックス

[ステップ]	連続実行するステップ数を設定します。
[ディレイ(秒)]	コードを自動的にステップ実行するときのステップ間の遅延を選択します。 0〜3 秒まで、0.5 秒単位で選択できます。
[ステップオーバステップ]	チェックすると関数呼び出しをステップオーバーします。
[ソースレベルステップ]	チェックするとソースレベルでステップします。

[OK]ボタンをクリックするか、”Enter”キーを押してステップ実行を開始します。

5.13 プログラムを停止する

この節では、作成したプログラムの実行を停止する方法を説明します。停止手段として、[停止]ボタンを使用して停止する方法、および作成したコードの特定の場所にブレークポイントを設定することによって停止する方法について説明します。

5.13.1 [停止]ツールバーボタンによる停止

作成したプログラムが実行中の場合、[停止]ツールバーボタン (赤い停止の印)を使用することができます。しかし、プログラムが停止している場合は、使用できません (STOP の印が灰色になります)。[停止]ツールバーボタンをクリックするか、[デバッグ->プログラムの停止]を選択することによりプログラムが停止します。

[停止]によりプログラムが停止したとき、[Output]ウィンドウの[Debug]タブに”User Break”というメッセージを表示します。

5.13.2 標準のブレークポイント(PC ブレークポイント)

作成したプログラムをデバッグする場合、PC ブレークポイントにより指定した行または命令でプログラムの実行を停止させることができます。標準的な PC ブレークポイントを設定したり解除したりする方法を以下に示します。より複雑な設定をする場合には、[イベントポイント]ウィンドウを使用します。[イベントポイント]ウィンドウは ボタンをクリックするか、

[表示->コード->イベントポイント]を選択することにより表示します。詳細は、「5.16 イベントポイントを使用する」を参照してください。

(1) [ソース]ウィンドウ上で PC ブレークポイントを設定する

1. PCブレークポイントを設定する位置の[逆アセンブリ]または[ソース]ウィンドウが開いていることを確認します。
2. プログラムを停止したい行でポップアップメニューの[ブレークポイントの挿入/削除] を選択するか、”F9”キーを押すかします。
3. 左余白に赤丸を表示します。これは、PC ブレークポイントを設定したことを示します。
4. ポップアップメニューの[ブレークポイントの有効化/無効化]を選択すると、現在設定しているブレークポイントの有効/無効の切り替えができます。

作成したプログラムを実行して PC ブレークポイントを設定したアドレスに達すると、[Output]ウィンドウの[Debug]タブに" PC Break"というメッセージを表示し、実行を停止し、[ソース]または[逆アセンブリ]ウィンドウを更新し、停止位置を左余白に矢印で表示します。

【注】 ブレーク発生時には、PC ブレークポイントを設定した行または命令を実行する直前で停止します。その PC ブレークポイントで停止した後に Go または Step を選択した場合、矢印で表示した行から実行します。

(2) [ブレークポイント]ダイアログボックスを使用して PC ブレークポイントを設定する

[編集->ソースブレークポイント...]を選択するとブレークポイントダイアログボックスを表示します。

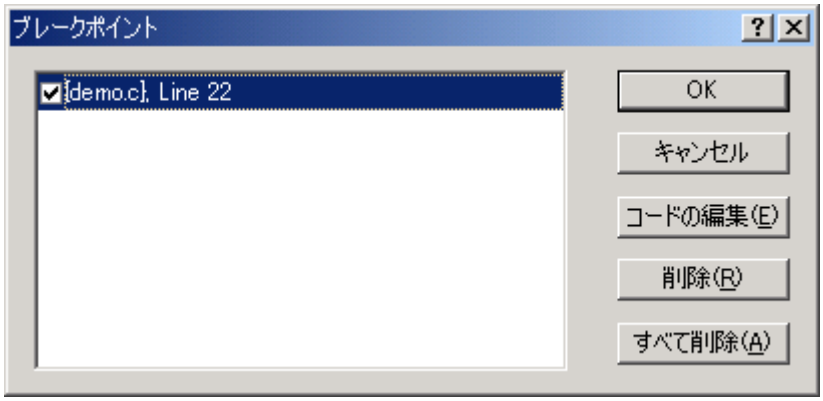


図 5-50 ブレークポイントダイアログボックス

現在設定しているブレークポイントを表示します。

[コードの編集]ボタンによりブレークポイントが存在するソースを見ることができます。 [削除]、[すべて削除]ボタンによりブレークポイントを１つまたはすべてを削除することができます。また、各ブレークポイントのチェックボックスにより有効/無効の切り替えができます。

(3) PC ブレークポイントを切り替える

PC ブレークポイントを設定している行の[BP]カラムをダブルクリックするか、その行にカーソルを置いて”F9”キーを使用すると、[PC ブレークポイント]の設定を切り替えることができます。切り替わる設定内容はデバッグプラットフォームによって異なります。

5.14 Elf/Dwarf2 のサポート

HEW は、C/C++およびアセンブリ言語で書いたアプリケーションのデバッグのために Elf/Dwarf2 オブジェクトファイルフォーマットをサポートします。

主な特長

- ソースレベルデバッグ
- C/C++演算子
- C/C++式(キャスト、ポインタ、参照)
- あいまいな関数名
- オーバーレイメモリロード
- ウオッチ-ローカル、およびユーザ定義
- スタックトレース

5.14.1 C/C++演算子

以下の C/C++ 言語演算子を使用することができます。

```
+, -, *, /, &, |, ^, ~, !, >>, <<, %, (, ), <, >, <=, >=, ==, !=, &&, ||
Buffer_start + 0x1000
#R1 | B'10001101
((pointer + (2 * increment_size)) & H'FFFF0000) >> D'15
!(flag ^ #ER4)
```

5.14.2 C/C++の式

式の例

```
Object.value                        //メンバの直接参照を指定します(C/C++)
p_Object->value                     //メンバの間接参照を指定します(C/C++)
Class::value                        //クラスを持つメンバの参照を指定します(C++)
*value                              //ポインタを指定します(C/C++)
&value                              //参照を指定します(C/C++)
array[0]                            //アレイを指定します(C/C++)
Object.*value                       //ポインタを持つメンバの参照を指定します(C++)
::g_value                           //グローバル変数の参照を指定します(C/C++)
Class::function(short)              //メンバ関数を指定します(C++)
(struct STR) *value                 //キャスト動作を指定します(C/C++)
```

5.14.3 複数ラベルをサポートする

プログラム言語の中の、例えば C++オーバーロード関数などでは、1 つのラベルが複数のアドレスを表す場合があります。各ダイアログボックスでこのようなラベル名を入力した場合、HEW は[関数選択]ダイアログボックスを使用してオーバーロード関数およびメンバ関数を表示します。

図 5-51 関数選択ダイアログボックス

[関数選択]ダイアログボックスでは、オーバーロード関数またはメンバ関数を選択します。通常、一度に一つの関数を選択します。ただし、ブレークポイントを設定する場合においてのみ、複数の関数を選択することができます。このダイアログボックスには 3 つの領域があります。

[関数名の選択]	同じ名前をもつ関数またはメンバ関数、およびその詳細情報を表示します。	
[関数名の指定]	設定する関数およびそれらの詳細情報を表示します。	
[カウンタ]	[全関数]	同じ名前をもつ関数かたはメンバ関数を表示します。
	[選択関数]	[関数名の選択] リストボックスに表示する関数の数を表示します。
	[指定関数]	[関数名の設定]リストボックスに表示する関数の数を表示します。

(1) 関数を選択する

[関数名の選択]リストボックスから選択したい関数をクリックして、[>]ボタンをクリックします。選択した関数を[関数名の設定]リストボックスに表示します。[関数名の選択]リストの関数すべてを選択するには、[>>]ボタンをクリックします。

(2) 関数の選択を解除する

[関数名の設定]リストボックスから選択を解除する関数をクリックして、[<]ボタンをクリックします。すべての関数の選択を解除するには、[<<]ボタンをクリックします。選択を解除した関数は、[関数名の設定]リストボックスから[関数名の選択]リストボックスへ戻します。

(3) 関数を設定する

[OK]ボタンをクリックして、[関数名の設定]リストボックスに表示した関数を設定します。関数を設定し、[関数選択]ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックすると、関数を設定せずにダイアログボックスを閉じます。

5.14.4 オーバレイプログラムのデバッグ

オーバレイ関数を使用するための設定について説明します。

(1) セクショングループを表示する

オーバレイ（いくつかのセクショングループを同じアドレス範囲に割り当てる）を使用すると、アドレス範囲およびセクショングループを[オーバレイ]ダイアログボックスに表示します。
[メモリ->オーバレイの構成]を選択して[オーバレイ]ダイアログボックスを開きます。

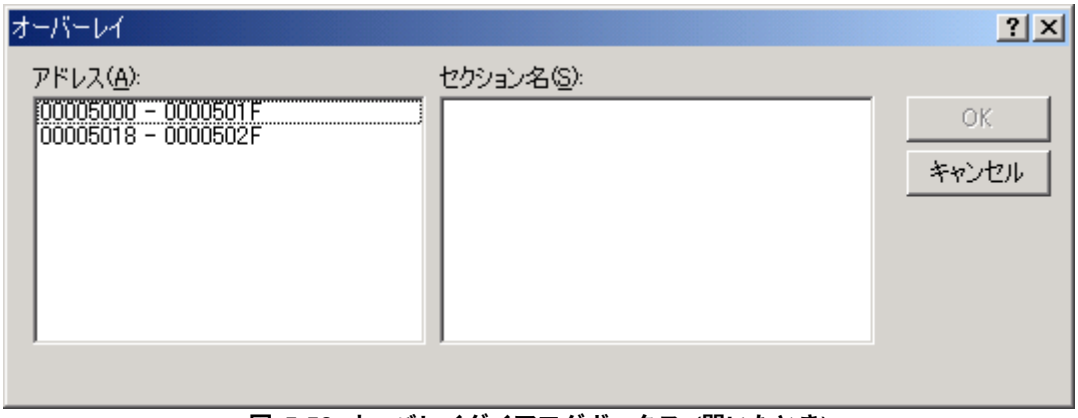


図 5-52 オーバレイダイアログボックス（開いたとき）

このダイアログボックスには 2 つの領域があります。[アドレス]リストボックスおよび[セクション名]リストボックスです。

[アドレス]リストボックスは、オーバレイプログラムが使用するアドレス範囲を表示します。アドレス範囲の 1 つをクリックして[アドレス]リストボックスのアドレス範囲を選択します。

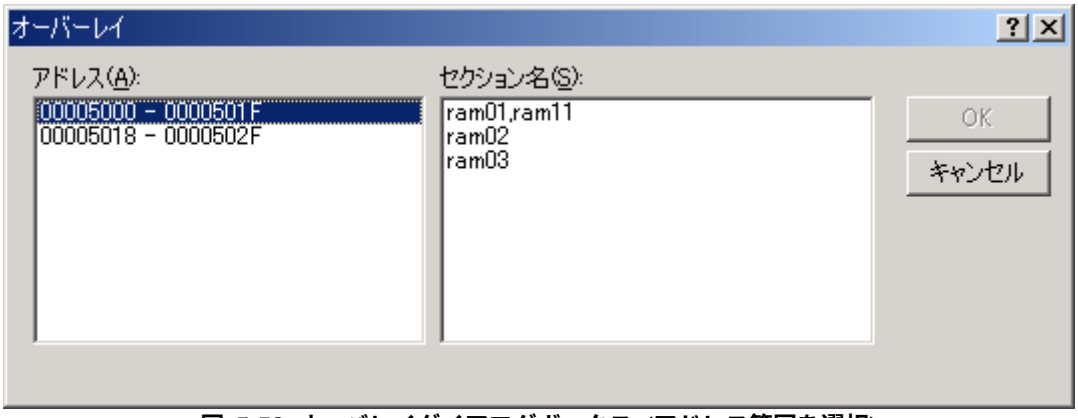


図 5-53 オーバレイダイアログボックス（アドレス範囲を選択）

[セクション名]リストボックスは、選択したアドレス範囲に割り当てたセクショングループを表示します。

➲セクショングループを設定するには
オーバレイ関数を使用するときは、最も優先度の高いセクショングループを[オーバレイ]ダイアログボックスで選択していなければ、HEW は正しく動作しません。
まず[アドレス]リストボックスに表示したアドレス範囲の一つをクリックします。選択したアドレス範囲に割り当てたセクショングループを[セクション名]リストボックスに表示します。
表示しているセクショングループの中から最も優先度の高いセクショングループをクリックします。

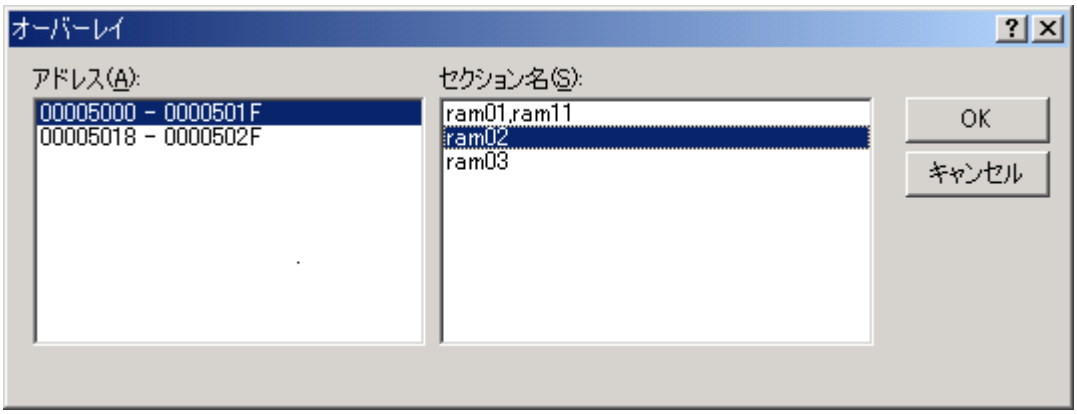


図 5-54 オーバレイダイアログボックス（最も優先度の高いセクショングループ選択時）

セクショングループを選択したら、[OK]ボタンをクリックして優先度の設定を保存して、ダイアログボックスを閉じます。[キャンセル]ボタンをクリックすると、優先度の設定を保存せずにダイアログボックスを閉じます。

【注】 オーバレイ関数が使用するアドレス範囲内では[オーバレイ]ダイアログボックスに指定したセクションのデバッグ情報を参照します。したがって、現在ロードしているプログラムと同じセクションを[オーバレイ]ダイアログボックスで選択しなければなりません。

5.15　変数の表示

本節では、ソースプログラム上の変数の値を表示する方法について説明します。

5.15.1　ツールチップウォッチ

作成したプログラムの変数を最もすばやく見るには、ツールチップウォッチ 機能を使用します。

➲ツールチップウォッチ を使用するには
確認したい変数を表示している[ソース]ウィンドウを開きます。
確認したい変数名の上にマウスのカーソルを静止させます。変数の近くにツールチップを表示し、その変数の基本的なウォッチ情報を表示します。

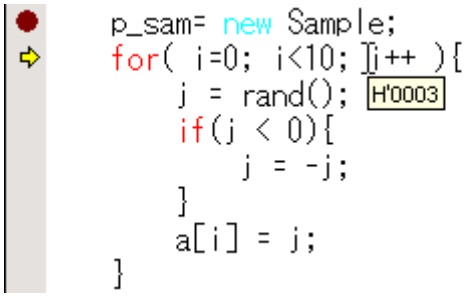


図 5-55 ツールチップウォッチ

5.15.2　インスタントウォッチ

確認したい変数を表示している[ソース]ウィンドウを開きます。
確認したい変数名の上にマウスのカーソルを置いてポップアップメニューから[インスタントウォッチ...]を選択します。
[インスタントウォッチ]ダイアログボックスが開き、カーソル上の変数を表示します。

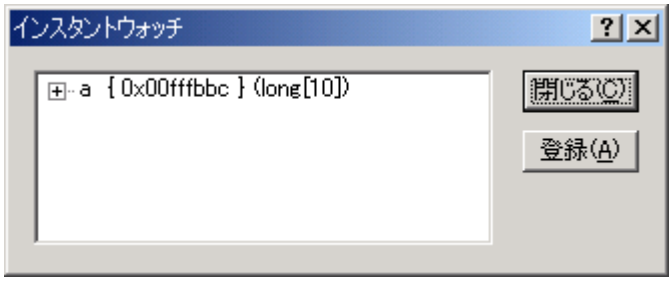


図 5-56 インスタントウォッチダイアログボックス

変数名の左側の'+'記号はクリックすれば情報を拡張表示できることを、'-'記号は情報を縮小表示できることを示します。[登録]ボタンを押すと、変数を[ウォッチ]ウィンドウに登録して、ダイアログボックスを閉じます。[閉じる]ボタンを押すと、変数を[ウォッチ]ウィンドウに登録しないで、ダイアログボックスを閉じます。

5.15.3　ウォッチウィンドウ

[ウォッチ]ウィンドウを開くことにより、任意の変数について値を参照することができます。

(1)　ウォッチウィンドウを開く

[ウォッチ]ウィンドウを開くには、[表示->シンボル->ウォッチ]を選択するか、[ウォッチ]ツールバーボタン が使用可能であれば、クリックします。[ウォッチ]ウィンドウが開きます。ウィンドウの中は最初空白です。

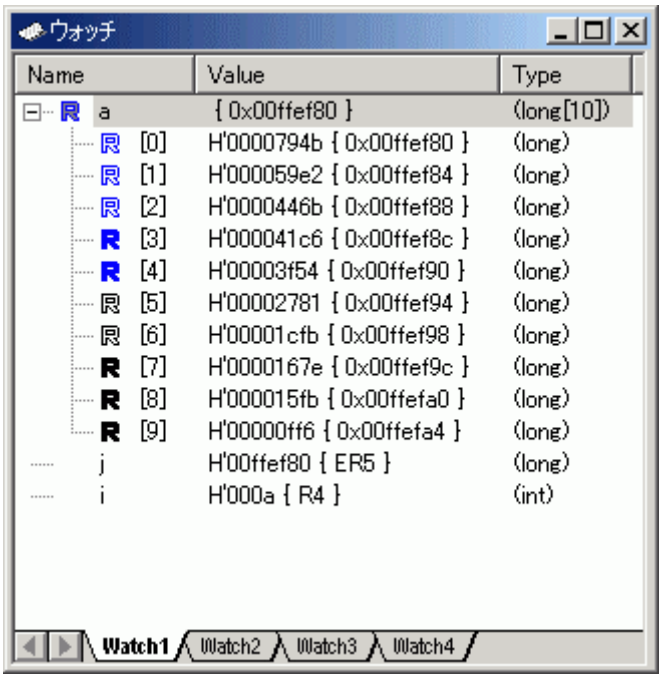


図 5-57 ウォッチウィンドウ

C/C++ソースレベルの変数を表示･変更することができるウィンドウです。本ウィンドウの内容は、アブソリュートファイル(*.abs)内のデバッグ情報から、C/C++ソースプログラムの情報がある場合にのみ表示します。コンパイラ等の最適化により、ソースプログラムの情報としてデバッグ情報にない場合は表示できません。また、マクロ宣言されたものについても表示できません。

表示する項目は以下の通りです。

[Name]	変数名を表示します
[Value]	変数の値、割り付け位置を表示します 割り付け位置は{}で囲んで表示します
[Type]	変数の型を表示します

R マークのある変数は、ユーザプログラム実行中に更新できることを示します。
E6000 エミュレータでは[ウォッチ]ウィンドウに登録した変数の内容をユーザプログラム実行中に更新する実現手段として下記 3 種類の方法があります。

1. モニタ機能を使用して、ユーザプログラムを停止せずに値の更新を行う。
   MCU内部のリード/ライト信号をトリガとしてアドレスバスおよびデータバスの値を保持し、該当する変数の値を更新します。

【注】 リアルタイム性は損なわれませんが、モニタ可能なサイズおよびポイントには限りがあります。
モニタ機能については「5.10 リアルタイムにメモリ内容を表示する」を参照してください。

2. エミュレータが強制的にバス権を確保することにより、ユーザプログラムを停止せずに、HEWから直接メモリ内容を読み出し値の更新を行う。

【注】 エミュレータがバス権を所有している間、CPU は動作を停止するため、リアルタイム性は損なわれます。
内蔵 ROM、内蔵 RAM、エミュレーションメモリに対するアクセス時のみ適応されます。

ご使用のエミュレータにより、適応されるエリアが異なる、もしくは本アクセス方法が適応できない場合があります。
詳細については「8 本製品固有のソフトウェア仕様」またはオンラインヘルプの[Configuration Properties]**ダイアログボックス**[General]**ページ**[Enable read and write on the fly]**項目をご参照ください。**

3. ユーザプログラムを一時的に停止し、メモリ内容を読み出し値の更新を行う。

【注】 ユーザプログラムを一時的に停止するためリアルタイム性は損なわれます。
上記項番 2 に該当するエリア以外のエリア(内蔵 I/O、DTCRAM、ユーザメモリ)に対するアクセス時に適応されます。

R マークの色によって、ユーザプログラム実行中における値更新の実現手段が判別できます。

中抜きの青色部	**変数のアドレスがモニタ機能で設定されているモニタ範囲内で、モニタ機能を使用したデータリードが可能であることを示します。**
青色部	**モニタ機能を使用したデータリードによる値の更新を行うことを示します。**
中抜きの黒色部	**変数のアドレスがモニタ機能で設定されているモニタ範囲外で、モニタ機能を使用したデータリードが不可能であることを示します。**
黒色部	**通常のデータリードにより値の更新を行うことを示します。**

【注】 1. **本機能は変数ごと、また構造体であれば指定された構造体一括/要素ごとに設定できます。**
2. **R マークの色は、モニタ設定を変更したときに変わります。**
3. **レジスタに割り付けられている変数には設定することができません。**

(2) ウォッチアイテムを追加する

ウォッチアイテムを[ウォッチ]ウィンドウに追加するには、[ウォッチ]ウィンドウの[シンボル登録]ダイアログボックスを使用ます。

➲ [ウォッチ]ウィンドウからシンボル登録を使用するには
[ウォッチ]ウィンドウを開きます。
ポップアップメニューから[シンボル登録]を選択します。
[シンボル登録]ダイアログボックスが開きます。

図 5-58 シンボル登録ダイアログボックス

見たい変数名を入力して、[OK]ボタンをクリックします。その変数を[ウォッチ]ウィンドウに追加します。
また、 [ソース]ウィンドウから[ウォッチ]ウィンドウへ変数をドラッグアンドドロップしても追加できます。

【注】 追加した変数がローカル変数で現在範囲外の場合には、HEW はその変数を[ウォッチ]ウィンドウに追加しますが、値には”Not available now”を表示します。

(3) ウォッチアイテムを拡張する

ウォッチアイテムがポインタ、アレイ、または構造体のとき、その名前の左側にプラス記号(+)の拡張指示子を表示します。つまり、ウォッチアイテムを拡張できるという意味です。ウォッチアイテムを拡張するには、プラス記号(+)をダブルクリックします。１つのタブによってインデントをつけたアイテムは拡張し、その要素(構造体またはアレイの場合)またはデータ値(ポインタの場合)を表示し、プラス記号がマイナス記号に変わります。ウォッチアイテムが要素にポインタ、構造体、またはアレイを含む場合、その横に拡張指示子を表示します。

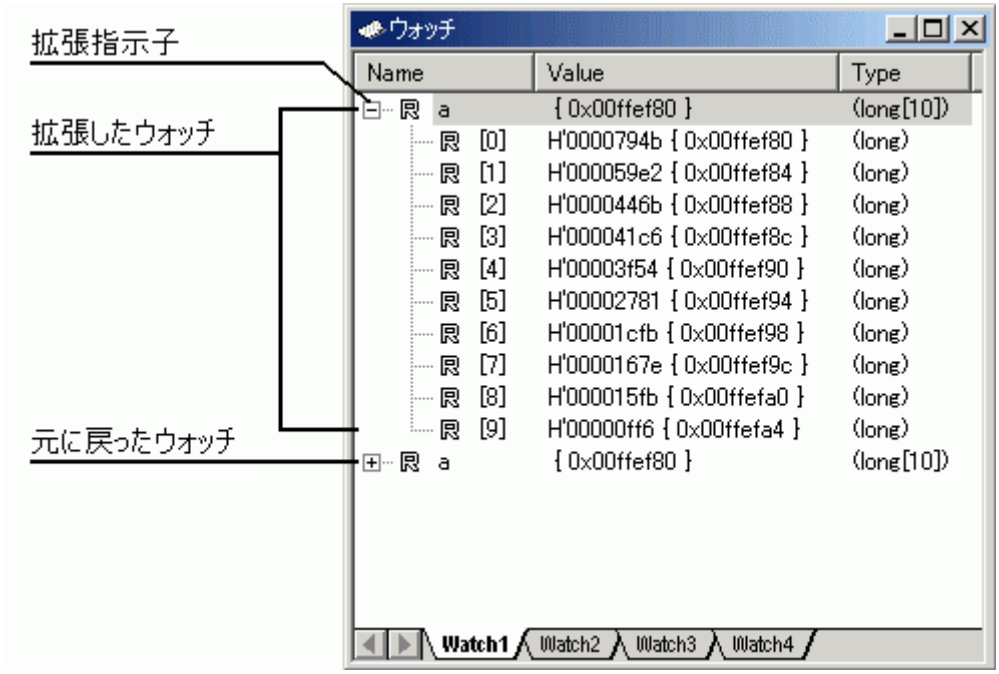


図 5-59 ウォッチを拡張する

拡張したウォッチアイテムを元に戻すには、再びアイテムをダブルクリックします。アイテムの要素は、元の単一のアイテムに戻り、マイナス記号はプラス記号に戻ります。

また、ウォッチアイテムを選択した状態で数字キー（１～９）を押すと、その数字の階層分ウォッチアイテムを拡張することができます。

(4) ウォッチアイテムの値を編集する

テストのためや、プログラムにバグがあるために値が正しくないときには、ウォッチ変数の値を変更することができます。ウォッチアイテムの値を変更するには、値の編集機能を使用します。

➲ ウォッチアイテムの値を編集するには
ウィンドウに対して値を直接入力します。
または、編集するアイテムをクリックにより選択すると、アイテム上のカーソルが点滅します。
ポップアップメニューから[値の編集]を選択します。
[値の編集]ダイアログボックスが開きます。

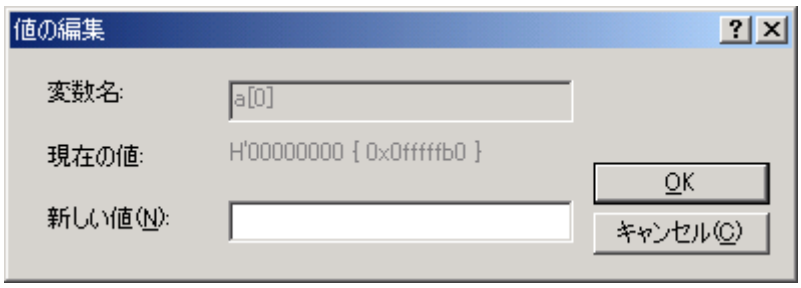


図 5-60 値の編集ダイアログボックス

[新しい値]フィールドに新しい値はまたは式を入力して[OK]ボタンをクリックします。[ウォッチ]ウィンドウを更新し、新しい値を表示します。

(5) ウォッチアイテムを削除する

特定のウォッチアイテムを削除するには、そのアイテムを選択してポップアップメニューから[削除]を選択します。アイテムを削除し、[ウォッチ]ウィンドウを更新します。

すべてのウォッチアイテムを削除するには、ポップアップメニューから[全シンボル削除]を選択します。すべてのアイテムを削除し、[ウォッチ]ウィンドウを更新します。

(6) リアルタイム更新を設定する

ウォッチアイテムで変数名の左に表示する”R”マークは、その変数がリアルタイムに更新できることを示します。R マークが太字のとき、その変数の値をプログラムの実行時に従ってリアルタイムに更新します。

リアルタイム更新は[ウォッチ]ウィンドウのポップアップメニューで設定します。

(a) 自動更新有効化

選択している変数の”R”マークが太字になり、リアルタイム更新します。

(b) 全シンボル自動更新有効化

すべての”R”マークが太字になり、リアルタイム更新します。

(c) 自動更新無効化

選択している変数の”R”マークが中抜きになり、リアルタイム更新を解除します。

(d) 全シンボル自動更新無効化

すべての”R”マークが中抜きになり、リアルタイム更新を解除します。

(7) 表示基数を変更する

変数を選択してポップアップメニューから[基数]を選択するとサブメニューで変数の表示基数を変更できます。

(8) 表示内容をファイルに保存する

[ウォッチ]ウィンドウの表示内容をファイルに保存するには、ポップアップメニューの[保存...]を選択します。

[保存...]を選択すると、[名前を付けて保存]ダイアログボックスを表示します。ファイル名を指定し、[ウォッチ]ウィンドウに表示している内容をセーブします。[Append]チェックボックスにチェックすると追加書きこみ、チェックしないと上書きします。

(9) メモリウィンドウを表示する

選択している変数が割り付いているメモリ領域を[メモリ]ウィンドウに表示することができます。ポップアップメニューの[メモリウィンドウへ移動...]を選択すると、[表示形式]ダイアログボックスを開きます。ダイアログの初期値には選択している変数の情報(先頭アドレス、終了アドレス、およびデータサイズ)を設定します。[OK]ボタンをクリックすることで[メモリ]ウィンドウを開きます。

5.15.4 ローカルウィンドウ

[ローカル]ウィンドウを開くことにより、ローカル変数の一覧とそれらの値を参照することができます。

(1) ローカルウィンドウを開く

[ローカル]ウィンドウを開くには、[表示->シンボル->ローカル]を選択するか、[ローカル]ツールバーボタン をクリックします。

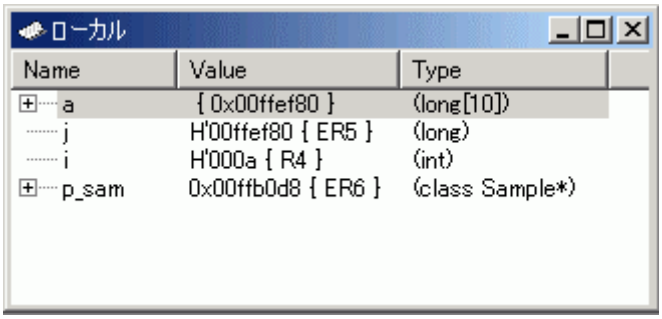


図 5-61 ローカルウィンドウ

ローカル変数を定義した時点で初期化していないと、ローカル変数に値を代入するまでローカル変数の値は不正となります。

ローカル変数の値およびローカル変数の表示は、[ウォッチ]ウィンドウと同じ方法で修正することができます。

5.16 イベントポイントを使用する

E6000 エミュレータは HEW 標準の PC ブレークポイントとは別に、より高度な条件指定によるブレーク、トレース、実行時間測定を行うイベントポイント機能を持っています。

5.16.1 PC ブレークポイントとは

PC ブレークポイントは指定アドレスの命令フェッチが行われた場合にユーザプログラムの実行を停止します。
最大 256 ポイントまで設定できます。
ただし、ユーザシステム上に実装された ROM エリアには 1 ポイントのみ PC ブレークポイントを設定することが可能です。この特殊な PC ブレークポイントのことを On Chip ブレークポイントと呼びます。
On Chip ブレークポイントは指定アドレスの命令実行後にユーザプログラムを停止します。

複数の PC ブレークポイントをユーザシステム上に実装された ROM エリアに設定する必要がある場合は、このエリアをエミュレーションメモリに割り付け、コードをコピーして、PC ブレークポイントを設定してください。

5.16.2 イベントポイントとは

イベントポイントは単一アドレス指定以外に、データ条件など、より高度な条件指定が可能なポイントです。
イベント検出システムのイベントチャネルと範囲チャネルを使って、最大 12 ポイント設定できます。
条件成立時の動作としてユーザプログラムの停止以外に実行時間測定の開始/終了、トレース取得の開始/終了条件として利用可能です。
複数のイベントポイントを組み合わせることにより、より複雑な条件設定が可能です。

【注】 イベントポイントはデータの取得、条件の判定、action(**ユーザプログラムの停止など)の実施を** E6000 **エミュレータのハードウェア回路にて行うため、条件成立から** action **の実施までに数サイクルの遅延が発生します。**

5.16.3 イベント検出システムとは

イベントは 8 ポイントのイベントチャネルと 4 ポイントの範囲チャネルに割り当てられます。
イベントチャネルには範囲チャネル以上の機能があります(シーケンス、カウントなど)。

(1) イベントチャネル(Ch1～Ch8)

E6000 は 8 ポイントのイベントチャネルを備えています。
イベントチャネルは以下の組合せで定義できます。

- アドレスまたはアドレス範囲
- アドレス範囲外
- リード、ライトまたは両方
- マスク条件指定付きデータ
- バス状態
- エリア
- 4 つの外部プローブ信号の値
- イベント発生回数
- イベント発生後のディレイサイクル数

また、最大 8 ポイントをシーケンスで組み合わせすることができます。
それぞれのシーケンスにおける前のイベントの発生によって起動、あるいは停止します。

(2) **範囲チャネル**(Ch9～Ch12)

E6000 は 4 ポイントの範囲チャネルを備えています。
範囲チャネルは以下の組合せで定義できます。

- アドレスまたはアドレス範囲
- リード、ライトまたは両方
- マスク条件指定付きデータ
- バス状態
- エリア
- 4 つの外部プローブ信号の値
- イベント発生後のディレイサイクル数

5.16.4 バス状態およびエリア信号について

イベント検出システムではイベント検出の条件として MCU のバス状態およびアクセスしたエリアを示す信号を指定できます。
これらの信号はエミュレータに実装された MCU より出力されるため、ご使用のエミュレータにより取得可能な信号が異なります。

バス状態およびエリア信号はイベントポイントの[Bus/Area]条件設定で使用します。
これらの信号はトレース情報としても取得可能です。
また、バス状態信号についてはトレース非取得条件設定([Suppress]オプション)およびハードウェアパフォーマンス測定の領域アクセス回数測定モード([Access Type]オプション)でも使用します。
トレース機能については「5.17 トレース情報を見る」を参照してください。
ハードウェアパフォーマンス機能については「5.21 パフォーマンスを測定する」を参照してください。

例として H8S/2633 E6000 エミュレータで取得可能なバス状態およびエリア信号を下記に示します。

表5-1 H8S/2633 E6000 **エミュレータで取得可能なバス状態信号**

バス状態	**トレース表示**(Status)	**詳細**
CPU Prefetch	PROG	CPU **プリフェッチサイクル**
CPU Data	DATA	CPU **データアクセスサイクル**
Refresh	REFRESH	**リフレッシュサイクル**
DMAC	DMAC	DMAC **サイクル**
DTC	DTC	DTC **サイクル**
Other	OTHER	**その他**

表5-2 H8S/2633 E6000 **エミュレータで取得可能なエリア信号**

エリア	**トレース表示**(Area)	**詳細**
On-chip ROM	ROM	ROM
On-chip RAM	RAM	RAM
On-chip I/O 16bit	I/O-16	16 **ビット** I/O
On-chip I/O 8bit	I/O-8	8 **ビット** I/O
External 16bit	EXT-16	16 **ビット** EXT(**外部**)
External 8bit	EXT-8	8 **ビット** EXT(**外部**)
DTC RAM	RAM/DTC	DTCRAM

【注】 取得可能なバス状態およびエリア信号はご使用のエミュレータにより異なります。
詳細につきましては「8 **本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。**

5.16.5 イベントポイントウィンドウを開く

[イベントポイント]ウィンドウを開くには、[表示->コード->イベントポイント]を選択するか、[イベントポイント]ツールバーボタンをクリックします。

[イベントポイント]ウィンドウには、3 枚のシートがあります。

[Breakpoint]シート
PCブレークポイントの設定内容を表示します。また、PCブレークポイントの設定、変更および解除を行うことができます。
[Event]シート
イベントポイントを表示、設定します。
[Trigger]シート
トリガポイントを表示、設定します。

5.16.6 PC ブレークポイントを設定する

[Breakpoint]シートでは PC ブレークポイントの設定内容の表示、変更および追加ができます。

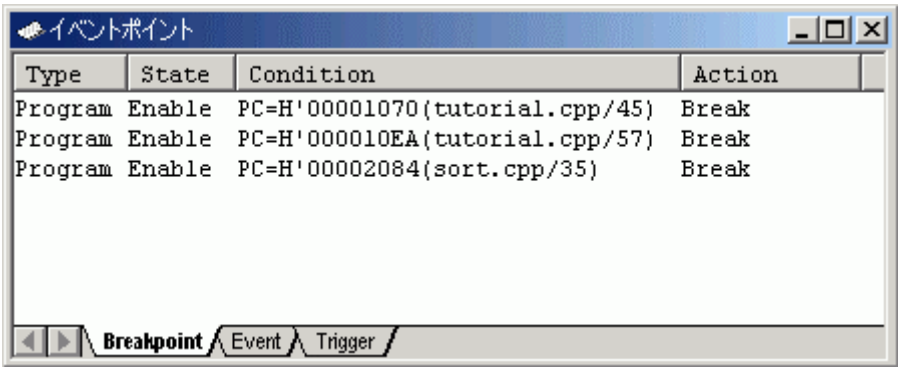


図 5-62 イベントポイントウィンドウ (Breakpoint シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されている PC ブレークポイントを選択しポップアップメニューから[編集...]を選択すると[Breakpoint/Event Properties]ダイアログボックスを表示します。

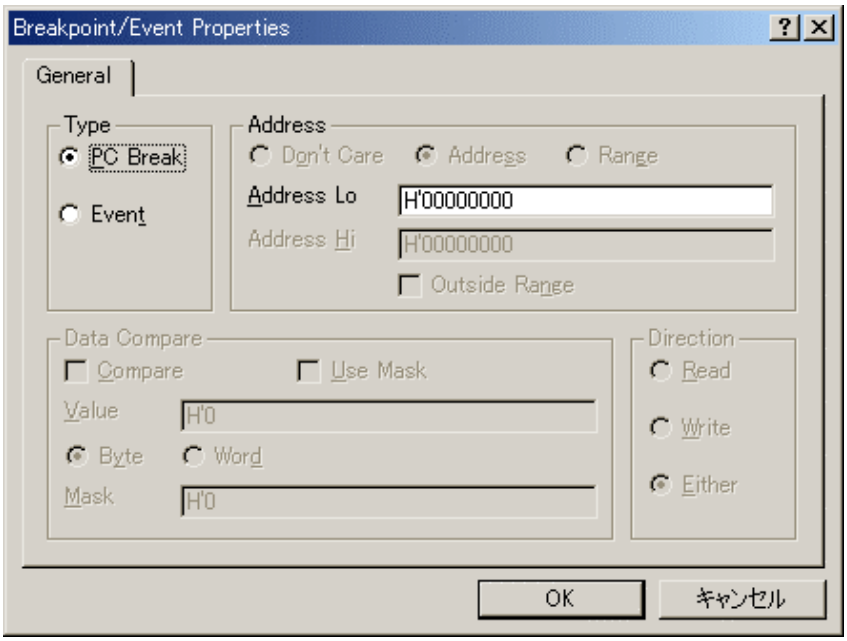


図 5-63 Breakpoint/Event Properties ダイアログボックス (PC Break 設定)

本ダイアログボックスより PC ブレークポイントを設定するアドレス条件を設定します。

[Type] ブレークポイントタイプを指定します。
[Breakpoint/Event Properties]ダイアログボックスは、PC ブレークポイントの設定とイベントポイントの設定とで共用になっています。
指定されたタイプに従い、ダイアログボックスの他の設定可能なオプションが有効表示されます。
表示されていないオプションについては設定できません。

[PC Break] プログラムフェッチによる単一アドレスのみ指定できます。
他のオプションはすべて無効です。

[Event] [General]ページの他のオプション、[Bus/Area]ページ、[Signals]ページまたは[Action]ページで詳細な検索条件を設定します。

[Address] アドレス条件を設定します。

[Address Lo] PC ブレークポイントを設定する単一アドレスを指定します。

5.16.7 イベントポイントを設定する

[Event]シートではイベントポイントの設定内容の表示、変更および追加ができます。

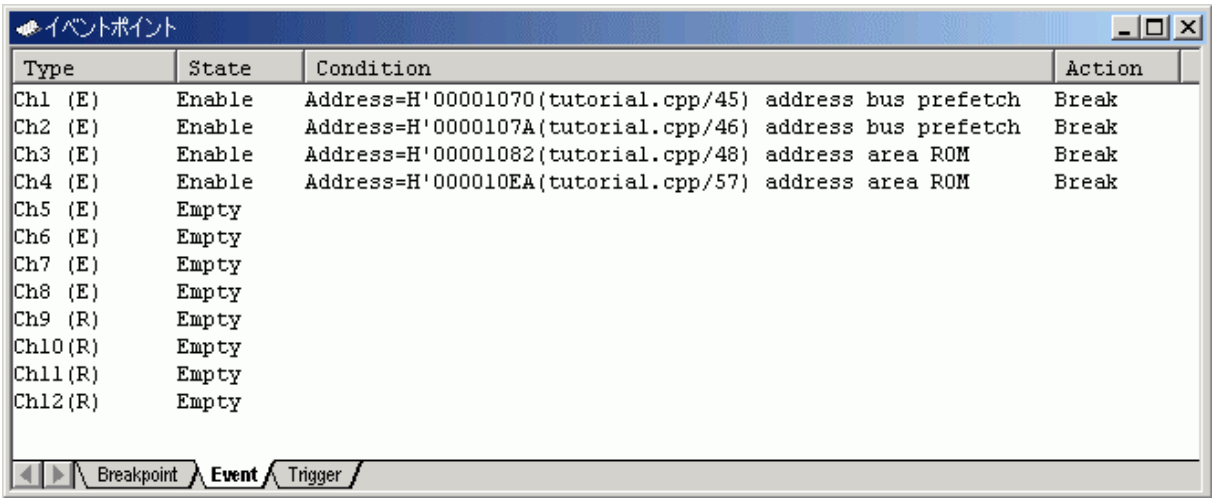


図 5-64 イベントポイントウィンドウ (Event シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているイベントポイントを選択しポップアップメニューから[編集...]を選択すると[Breakpoint/Event Properties]ダイアログボックスを表示します。

イベントポイントの条件設定は[General]ページ、[Bus/Area]ページ、[Signals]ページ、[Action]ページより構成されています。

各ページで設定した条件を複合したものが イベントポイントの検索条件を設定します。

【注】 1. **チャネル** 8 **はトリガ出力機能を備えています。**
チャネル 8 **の条件が成立した場合、外部プローブ** 1(EXT1)**より** 1 **バスサイクルの期間だけ** Low **レベルを出力します。**

2. **イベントポイントをトレース情報の取得条件として使用する場合はポップアップメニューより**[Trace Acquisition...]**を選択してください。**
トレース機能については「5.17 **トレース情報を見る」を参照してください。**

3. **範囲チャネル**(Ch9～Ch12)**編集時に範囲チャネルでは使用できない条件を設定した場合、設定チャネルは自動的に未使用のイベントチャネル**(Ch1～Ch8)**に変更されます。**

表5-3 **範囲チャネルでは使用できない条件**

条件	関連オプション
指定アドレス範囲外指定	[General]**ページ**[Outside Range]**項目**
実行時間計測開始/終了指定	[Action]**ページ**[Start Timer], [Stop Timer]**項目**
イベント発生回数指定 (2 **回以上指定時**)	[Action]**ページ**[Required number of event occurrences]**項目**
シーケンス指定	[Action]**ページ**[Enable Sequencing]**項目**

(1) General ページ

アドレス条件およびデータ条件を設定します。

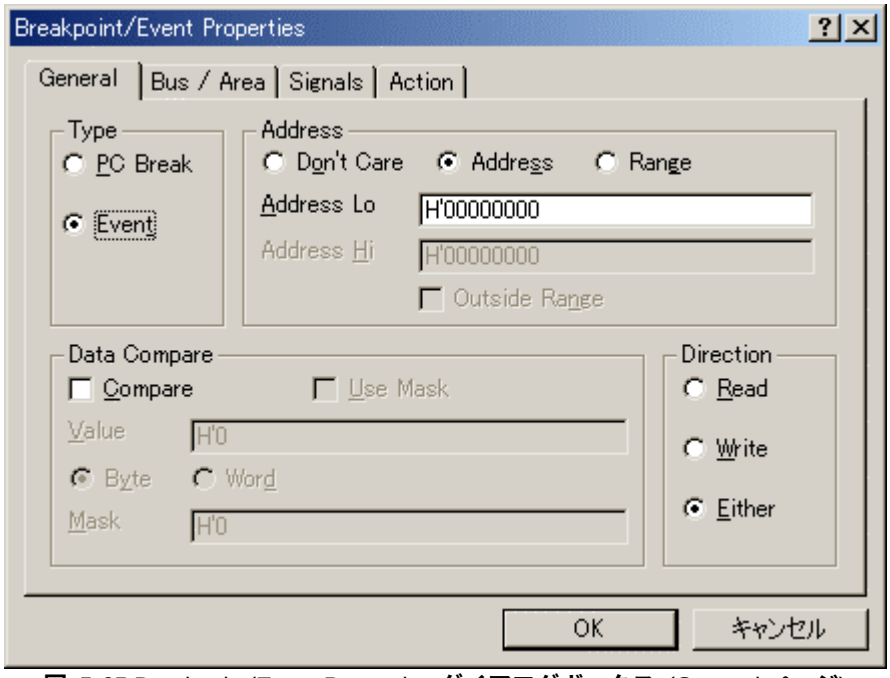


図 5-65 Breakpoint/Event Properties ダイアログボックス (General ページ)

[Type] ブレークポイントタイプを指定します。
[Breakpoint/Event Properties]ダイアログボックスは、PC ブレークポイントの設定とイベントポイントの設定とで共用になっています。
指定されたタイプに従い、ダイアログボックスの他の設定可能なオプションが有効表示されます。表示されていないオプションについては設定できません。

- [PC Break] プログラムフェッチによる単一アドレスのみ指定できます。
  他のオプションはすべて無効です。
- [Event] [General]ページの他のオプション、[Bus/Area]ページ、[Signals]ページまたは[Action]ページで詳細な検索条件を設定します。

[Address] アドレス条件を設定します。

- [Don't Care] アドレス条件を設定しません。
- [Address] 単一アドレスを指定します。
- [Range] 範囲アドレスを指定します。
- [Address Lo] 単一アドレスまたは範囲アドレスの開始アドレスを指定します。
  ([Address/Range]選択時有効)
- [Address Hi] 範囲アドレスの終了アドレスを指定します。
  ([Range]選択時有効)
- [Outside Range] アドレス範囲を否定します。(例 範囲外のアドレス)
  ([Range]選択時有効)

[Data Compare] データ条件を設定します。

- [Compare] チェックすると、データ比較を行います。
- [Use Mask] マスク条件を指定します。
  ([Compare]選択時有効)
- [Value] データバスの値を数値で設定します。
  また、データのアクセスサイズを選択します。
  ([Compare]選択時有効)
- [Byte] バイトアクセスを条件にします。
  ([Compare]選択時有効)
- [Word] ワードアクセスを条件にします。
  ([Compare]選択時有効)
- [Mask] マスクする値を設定します。
  マスク値を設定した場合、データバスの値およびデータ条件の値双方に対しマスク値で AND 演算を行い、その結果をもとにデータの比較を行います。
  ([Use Mask]選択時有効)

[Direction] リード、ライトサイクルの条件を選択します。
[Read] リードサイクルを条件にします。
[Write] ライトサイクルを条件にします。
[Either] リード、ライト両方のサイクルを条件にします。

(2) Bus/Area ページ

バス状態およびメモリアクセスエリアを指定します。

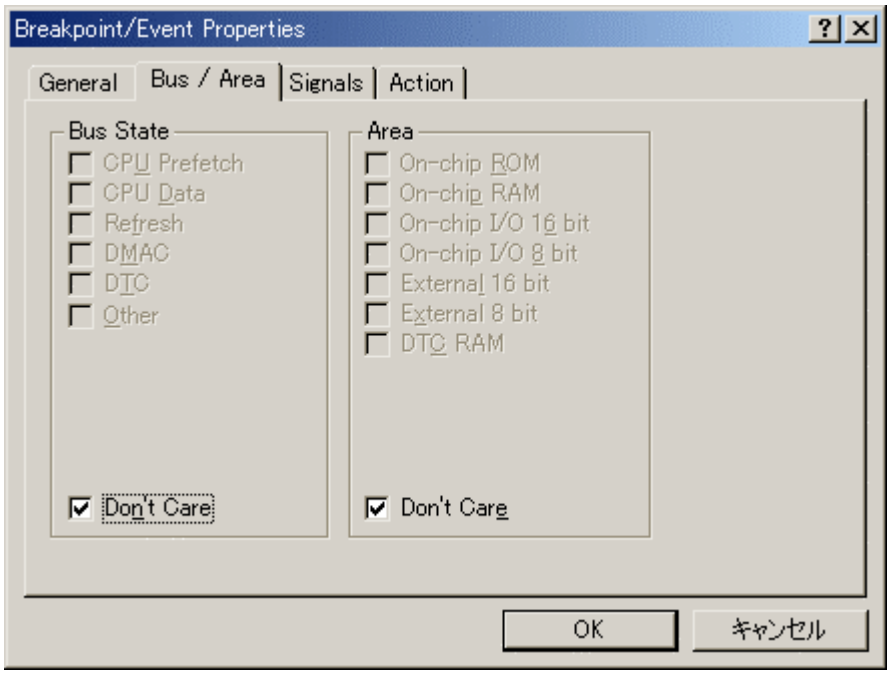


図 5-66 Breakpoint/Event Properties **ダイアログボックス** (Bus/Area **ページ**)

[Bus State] バス状態を指定します。
Don't Care チェックボックスをチェックすると、すべてのバス状態でイベントが成立します。
[Area] 検索するエリアを指定します。
Don't Care チェックボックスをチェックすると、すべてのエリアでイベントが成立します。

【注】 バス状態およびメモリアクセスエリアの設定項目についてはご使用のエミュレータにより異なります。詳細につきましては「5.16.4 バス状態およびエリア信号について」をご参照ください。

(3) Signals ページ

外部信号を指定します。

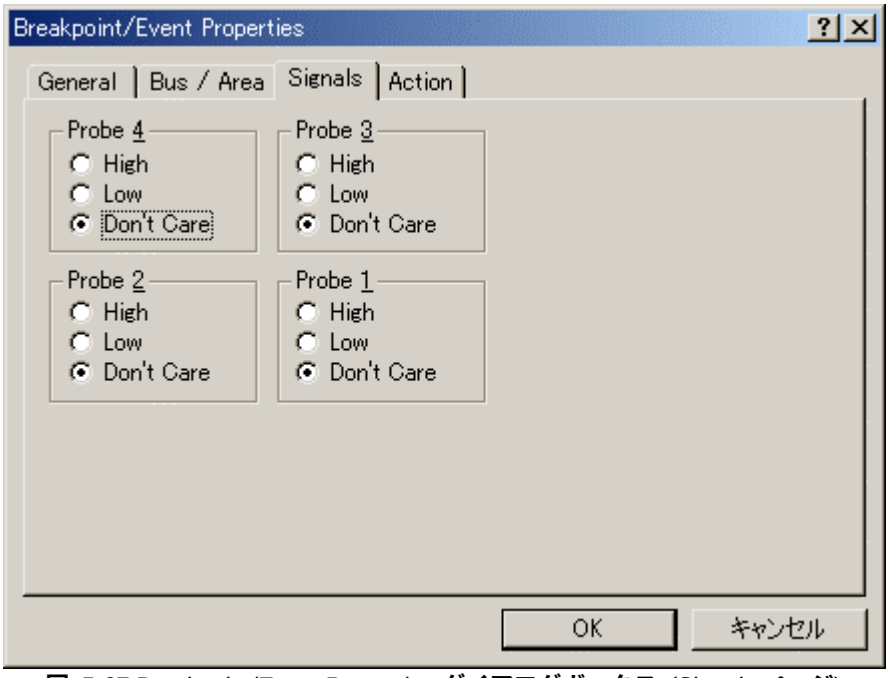


図 5-67 Breakpoint/Event Properties **ダイアログボックス** (Signals **ページ**)

[Probe4] 入力プローブ 4 の状態検出
- [High] 入力プローブが High の状態を検出します。
- [Low] 入力プローブが Low の状態を検出します。
- [Don’t Care] 入力プローブの状態を検出しません。

[Probe3] 入力プローブ 3 の状態検出
- [High] 入力プローブが High の状態を検出します。
- [Low] 入力プローブが Low の状態を検出します。
- [Don’t Care] 入力プローブの状態を検出しません。

[Probe2] 入力プローブ 2 の状態検出
- [High] 入力プローブが High の状態を検出します。
- [Low] 入力プローブが Low の状態を検出します。
- [Don’t Care] 入力プローブの状態を検出しません。

[Probe1] 入力プローブ 1 の状態検出
- [High] 入力プローブが High の状態を検出します。
- [Low] 入力プローブが Low の状態を検出します。
- [Don’t Care] 入力プローブの状態を検出しません。

(4) Action ページ

イベントが成立すると、エミュレータがどんな action をするかを決定します。

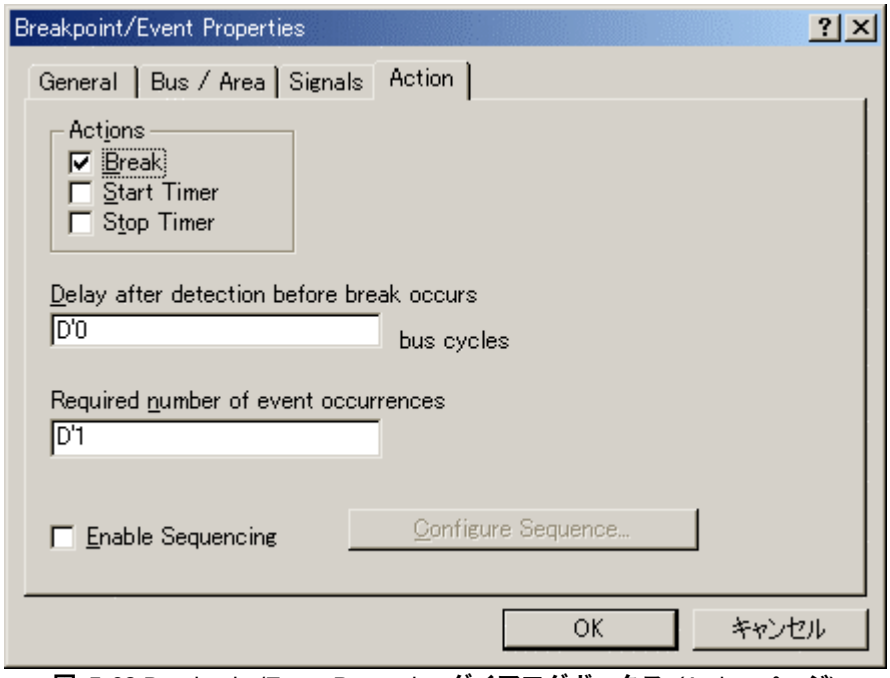


図 5-68 Breakpoint/Event Properties **ダイアログボックス** (Action **ページ**)

[Action]	イベント成立時の action を設定します。 トレース情報取得条件として使用するイベントポイントでは設定できません。
[Break]	イベントが成立したとき、ユーザプログラムを中断(停止)します。 これはデフォルト action です。
[Start Timer]	実行タイマを始動させます。(実行タイマ値は[ステイタス]ウィンドウに表示されます。)
[Stop Timer]	実行タイマを停止させます。(実行タイマ値は[ステイタス]ウィンドウに表示されます。)
[Delay after detection before break occurs]	イベント成立後の action が発生するまでの 16 ビットディレイ(バスサイクル)を設定します。 ディレイはイベントブレークのみで使用できるハードウェア上の 1 つだけのディレイカウンタです。したがって、1 つのブレークポイントのみディレイを指定することができます。 パラメータの指定可能範囲は D'0～D'65535 です。 ([Break]選択時有効) トレース情報取得条件として使用するイベントポイントでは設定できません。
[Required number of event occurrences]	16 ビットパスカウントを設定します。指定した回数イベント成立後、action が発生します。 パラメータの指定可能範囲は D'1～D'65535 です。
[Enable Sequencing]	イベントを順位づけします(イベントチャネルの設定が必要です)。
[Configure Sequence...]	[Event Sequencing]ダイアログボックスを表示し、イベントシーケンスを構成します。 ([Enable Sequencing]選択時有効)

(5) Event Sequencing ダイアログボックス

イベントが他のイベントのトリガとなることを定義します。

[Trace Acquisition…]からこのダイアログをアクセス(直接または間接)すると、トレース情報取得条件で使用するイベントポイントのみ表示します。

[イベントポイント]ウィンドウからアクセスすると、ブレークポイントまたはタイマイベントのみ表示します。

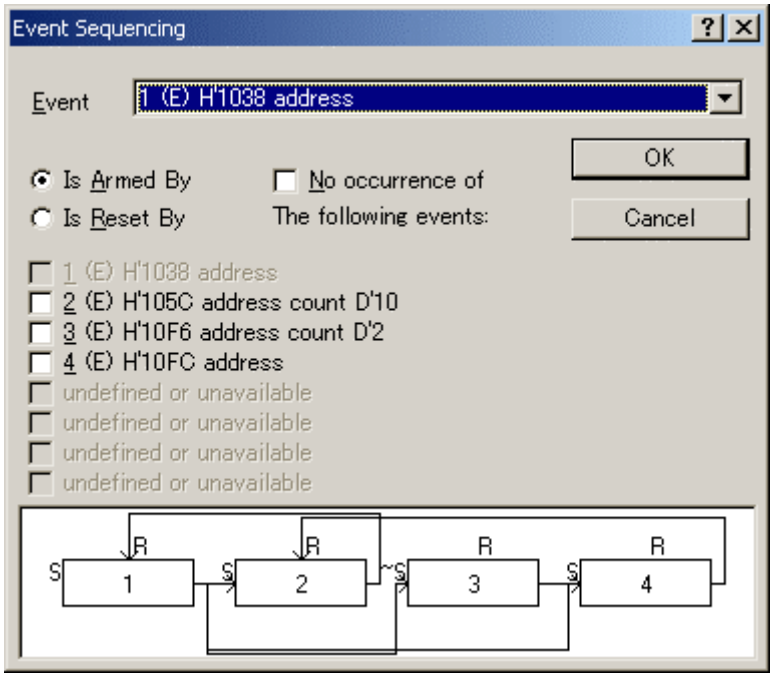


図 5-69 Event Sequencing **ダイアログボックス**

[Event]	設定の対象となるイベントポイントを選択します。
[Is Armed By]	イベントのシーケンス入力を設定します。
[Is Reset By]	イベントのリセット入力を設定します。
[No occurrence of]	イベントの非発生を条件にします。 ([Is Armed By]選択時有効)

イベントポイントの条件判定はユーザプログラムの実行と共に開始します。

ユーザプログラム実行開始直後、イベントポイントは条件非成立状態です。

イベントポイントの条件成立により設定された action を実施し条件成立状態に遷移します。

条件成立状態はユーザプログラムの停止またはイベントポイントのリセットが行われるまで維持されます。

イベントポイントが条件成立状態の場合、再び条件が成立しても action は実施されません。

条件成立により再度 action を実施したい場合はイベントポイントのリセットを行うことにより条件非成立状態に遷移させる必要があります。

ユーザプログラムの停止によりイベントポイントはすべて条件非成立状態に遷移します。

あるイベントポイントの成立条件として別のイベントポイントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要がある場合、後者のイベントポイントを arm イベントと呼びます。

イベントポイントは条件成立により他のイベントポイントまたは自分自身の条件判定状態をリセットできます。

このようなイベントポイントを reset イベントと呼びます。

reset イベントは条件成立状態のイベントポイント、条件非成立状態のイベントポイントにかかわらずイベントポイントのリセットを行います。(例: カウント途中のパスカウントをリセットするなど)。

[Event]ドロップダウンリストボックスより設定の対象となるイベントポイントを選択してください。

選択したイベントポイントに対して arm イベントを設定する場合は、[Is Armed By]を選択し、該当のイベントに対応したチェックボックスをチェックしてください。また、[No occurrence of]チェックボックスをチェックすることにより arm イベントの条件非成立状態を条件とすることができます。

選択したイベントポイントに対して reset イベントを設定する場合は、[Is Reset By]を選択し、該当のイベントに対応したチェックボックスをチェックしてください。

画面の下側にイベントの順序付けを示す図があります。(図 5-69)

S はシーケンス入力(arm イベント条件成立状態)、R はリセット入力(reset イベント)を示します。~S は非発生のシーケンス入力(arm イベント条件非成立状態)を示します。

図 5-69を例にすると、Ch1 は Ch2、Ch3、Ch4 に対する arm イベントとなります。Ch3 は Ch4 に対する arm イベントとなります。Ch2 は Ch1 に対する reset イベントとなります。Ch4 は Ch2 に対する reset イベントとなります。

arm イベントを持つイベントポイントの条件成立には、arm イベントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要があります。

一つのイベントポイントに対し複数の arm イベントが存在する場合、イベントポイントの条件成立には、いずれか一つの arm イベントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要があります。

一つのイベントポイントに対して arm イベントの条件は、条件成立状態または条件非成立状態のいずれか一方のみ設定できます。

reset イベントによるイベントポイントのリセットは、reset イベントの条件成立により実施されます。

reset イベントとなるイベントポイントが条件成立状態の場合、再び条件が成立してもイベントポイントのリセットは実施されません。

一つのイベントポイントに対し複数の reset イベントが存在する場合､いずれか一つの reset イベントが条件成立した場合にイベントポイントのリセットが実施されます。

5.16.8 トリガポイントを設定する

トリガポイントとは指定アドレスに対してアクセスが行われた際にトリガ出力を行うイベントです。

トリガポイントは E6000 エミュレータのバスモニタ回路に実装されたトリガ出力(4 チャネル)を使用し、最大 4 ポイント設定できます。

[Trigger]シートではトリガポイントの設定内容の表示および変更ができます。

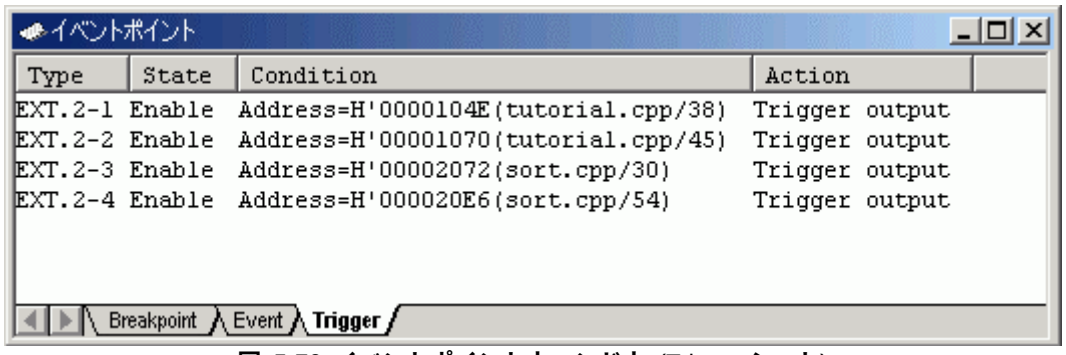


図 5-70 イベントポイントウィンドウ (Trigger シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているイベントポイントを選択しポップアップメニューから[編集...]を選択すると[Set Address For Trigger]ダイアログボックスを表示します。

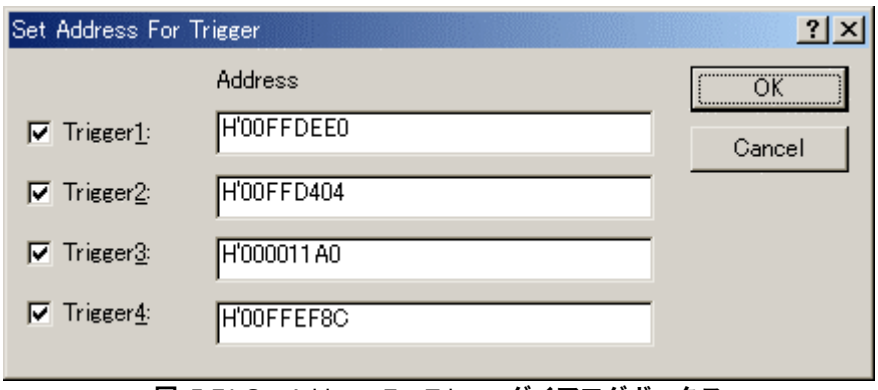


図 5-71 Set Address For Trigger ダイアログボックス

ユーザプログラム実行中のトリガ出力条件として、アクセスするアドレスを指定します。

トリガ出力ポイントの有効/無効は画面左側のチェックボックスで選択してください。

[Trigger1] トリガチャネル 1 の出力を有効にします。
[Trigger2] トリガチャネル 2 の出力を有効にします。
[Trigger3] トリガチャネル 3 の出力を有効にします。
[Trigger4] トリガチャネル 4 の出力を有効にします。
[Address] 該当チャネルのアドレス条件を設定します。

【注】1. **トリガ出力はトリガ設定**(1〜4)**の条件が一致した場合に外部プローブ** 2(EXT2)**の該当ピン**(1〜4)**よりリード/ライトの期間だけ** High **レベルを出力します。**
2. **ご使用のエミュレータによりトリガポイントをサポートしていない場合があります。詳細につきましては「**8 **本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。**

5.16.9 イベントポイントの編集

PC ブレークポイント、イベントポイント、トリガポイントに対する設定以外の操作方法はすべて共通となっています。

以下イベントポイントを例に設定以外の操作方法について説明します。

5.16.10 イベントポイントの設定内容を変更する

変更したいイベントポイントを選択後ポップアップメニューから[編集...]を選択すると、各イベントに対応した設定ダイアログボックスが開き、設定内容を変更することができます。[編集...]メニューはイベントポイントを 1 個選択しているときのみ有効となります。

5.16.11 イベントポイントを有効にする

イベントポイントを選択後ポップアップメニューから[有効]を選択すると、選択しているイベントポイントを有効にします。

5.16.12 イベントポイントを無効にする

イベントポイントを選択後ポップアップメニューから[無効]を選択すると、選択しているイベントポイントを無効にします。無効にした場合は、イベントポイントはリストには残りますが、指定した条件が一致してもイベントは発生しません。

5.16.13 イベントポイントを削除する

イベントポイントを選択後ポップアップメニューから[削除]を選択すると、選択しているイベントポイントを削除します。イベントポイントを削除しないで、詳細情報は保持したまま、条件が成立してもイベントを発生させないようにするには、[無効]オプションを使用します(「5.16.12　イベントポイントを無効にする」参照)。

【注】　トリガポイントは削除できません。設定を解除する場合は[無効]オプションを使用してください。

5.16.14 イベントポイントをすべて削除する

ポップアップメニューから[すべて削除]を選択すると、すべてのイベントポイントを削除します。

【注】　トリガポイントは削除できません。[すべて削除]を選択した場合、全チャネルの設定が無効となります。

5.16.15 イベントポイントのソース行を表示する

イベントポイントを選択後ポップアップメニューから[ソースを表示]を選択すると、ブレークポイントのある[ソース]または[逆アセンブリ]ウィンドウをオープンします。[ソースを表示]メニューは対応するソースファイルを持つイベントポイントを 1 個選択しているときのみ有効となります。

5.17 トレース情報を見る

E6000 エミュレータでは命令の実行結果をトレース情報としてトレースバッファに取得し、表示することができます。

トレース情報は[Trace]ウィンドウに表示します。トレース情報の取得条件は、[Trace Acquisition]ダイアログボックスで設定します。

トレース機能は E6000 エミュレータのハードウェア回路によりバスサイクル単位にトレース情報を取得し、トレースバッファに保持するためリアルタイム性は損なわれません。

リアルタイムトレースバッファは、32768 までのバスサイクルを保持でき、バッファがいっぱいになった場合は取得したトレース情報のうち最も古いデータを上書きすることによりトレース情報を取得し続けます。

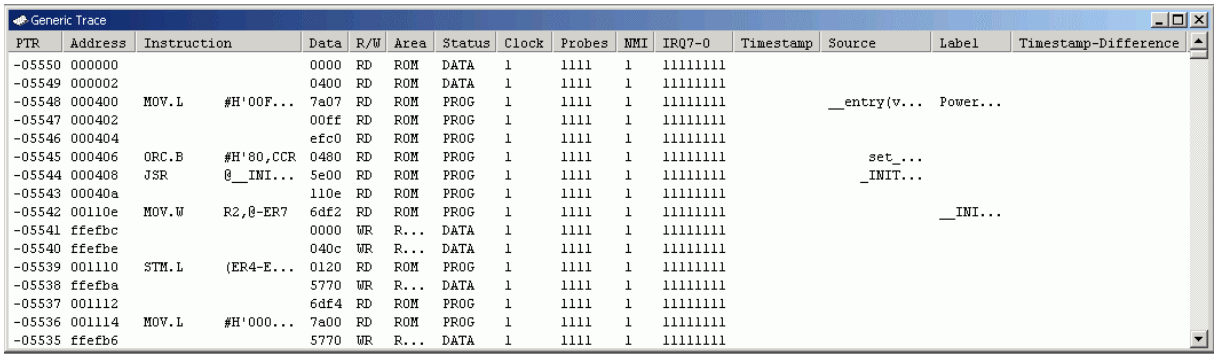
5.17.1 Trace ウィンドウを開く

[Trace]ウィンドウを開くには、[表示->コード->トレース]を選択するか、[トレース]ツールバーボタン をクリックします。

5.17.2 トレース情報を取得する

E6000 エミュレータはトレース情報の取得条件を設定しない場合、デフォルトで無条件に全バスサイクルをトレース取得します。(フリートレースモード)

フリートレースモードではユーザプログラムの実行開始と共にトレース取得を開始し、ユーザプログラムの停止によりトレース取得を停止します。取得したトレース情報は[Trace]ウィンドウに表示します。



PTR	Address	Instruction		Data	R/W	Area	Status	Clock	Probes	NMI	IRQ7-0	Timestamp	Source	Label	Timestamp-Difference
-05550	000000			0000	RD	ROM	DATA	1	1111	1	11111111				
-05549	000002			0400	RD	ROM	DATA	1	1111	1	11111111				
-05548	000400	MOV.L	#H'00F...	7a07	RD	ROM	PROG	1	1111	1	11111111		__entry(v...	Power...	
-05547	000402			00ff	RD	ROM	PROG	1	1111	1	11111111				
-05546	000404			efc0	RD	ROM	PROG	1	1111	1	11111111				
-05545	000406	ORC.B	#H'80,CCR	0480	RD	ROM	PROG	1	1111	1	11111111		set_...		
-05544	000408	JSR	@__INI...	5e00	RD	ROM	PROG	1	1111	1	11111111		_INIT...		
-05543	00040a			110e	RD	ROM	PROG	1	1111	1	11111111				
-05542	00110e	MOV.W	R2,@-ER7	6df2	RD	ROM	PROG	1	1111	1	11111111			__INI...	
-05541	ffefbc			0000	WR	R...	DATA	1	1111	1	11111111				
-05540	ffefbe			040c	WR	R...	DATA	1	1111	1	11111111				
-05539	001110	STM.L	(ER4-E...	0120	RD	ROM	PROG	1	1111	1	11111111				
-05538	ffefba			5770	WR	R...	DATA	1	1111	1	11111111				
-05537	001112			6df4	RD	ROM	PROG	1	1111	1	11111111				
-05536	001114	MOV.L	#H'000...	7a00	RD	ROM	PROG	1	1111	1	11111111				
-05535	ffefb6			5770	WR	R...	DATA	1	1111	1	11111111				

図 5-72 Trace ウィンドウ

表示する項目は以下の通りです。

[PTR]	トレースバッファ内のサイクル番号。 最後に取得されたサイクルの番号を 0 とし、古いサイクルにさかのぼって、順に-1、-2 と番号が小さくなります。 ディレイカウントが設定されている場合は、トレース停止条件が成立したサイクル番号を 0 とし、成立後停止するまでに実行されたサイクル(ディレイ期間中のサイクル)には、最後に取得されたサイクルに向かって順に+1、+2 と番号が大きくなります。
[Address]	アドレスバスの値(6 桁の 16 進数)。
[Instruction]	実行された命令の逆アセンブルコード。
[Data]	データバスの値。 それぞれ 16 進の 2 桁、4 桁で表示します。
[R/W]	アクセスサイクルの種類。読み出しは RD、書き込みは WR と表示します。
[Area]	アクセスしたエリア: ROM, RAM, 8 または 16 ビット I/O, 8 または 16 ビット EXT(外部), DTCRAM (タイムスタンプ取得時は無効)
[Status]	バス状態: DTC 動作, PROG(プリフェッチ), Data(CPU データアクセスサイクル), Refresh(リフレッシュサイクル), DMAC(DMAC サイクル) (タイムスタンプ取得時は無効)
[Clock]	バスサイクルのクロック数 1～8 を表示します。それ以上の場合は OVR を表示します。 (タイムスタンプ取得時は無効)
[Probes]	4 本のプローブ信号の状態(2 進数 4 桁)。 左から Probe4、Probe3、Probe2、Probe1 の順で表示します。 (タイムスタンプ取得時は無効)
[NMI]	NMI 入力の状態。 (タイムスタンプ取得時は無効)
[IRQ7-0]	8 本の IRQ 入力の状態。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプ。 タイムスタンプは、ユーザプログラム実行を開始するたびに 0 からカウントを始めます。測定時間の分解能は、Trace Acquisition で設定してください。 (タイムスタンプ取得時のみ有効)
[Source]	ソースコードの表示。
[Label]	アドレスに対応するラベル(ラベルが設定されている場合のみ表示します)。
[Timestamp-Difference]	前の行とのタイムスタンプの差分時間を表示。 (タイムスタンプ取得時のみ有効)

【注】 [PTR], [Address], [Instruction], [Data], [R/W], [Area], [Status], [Probes], [Timestamp], [Source], [Label], [Timestamp-Difference]**以外の項目はご使用のエミュレータにより異なります。**
詳細につきましては「8　本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

[Trace]ウィンドウ内の不要なカラムは非表示にすることができます。
カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。
カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

5.17.3　トレース情報取得条件を設定する

トレースバッファは有限であるため、バッファがいっぱいになった場合は最も古いトレース情報から順に上書きします。トレース情報の取得条件を設定することにより、有用なトレース情報のみを取得し、トレースバッファを有効に活用することができます。
トレース情報の取得条件はイベントポイントにて実現し、トレース情報取得の開始、停止、終了を制御します。
イベントポイントについては「5.16　イベントポイントを使用する」を参照してください。

トレース情報の取得条件はポップアップメニューから[設定...]を選択すると表示される[Trace Acquisition]ダイアログボックスで設定します。
[Trace Acquisition]ダイアログボックスは下記ページより構成されています。

表5-4 [Trace Acquisition]ダイアログボックスのページ構成

ページ	設定項目
[General]	トレース情報の取得条件を設定します。
[Stop]	トレース停止条件を設定します。(ディレイ設定なし)
[Delayed Stop]	トレース停止条件を設定します。(ディレイ設定あり)
[1]〜[4]	範囲トレースを設定します。 (フリートレースモード無効時のみ有効)

(1) General ページ

トレース情報の取得条件を設定します。

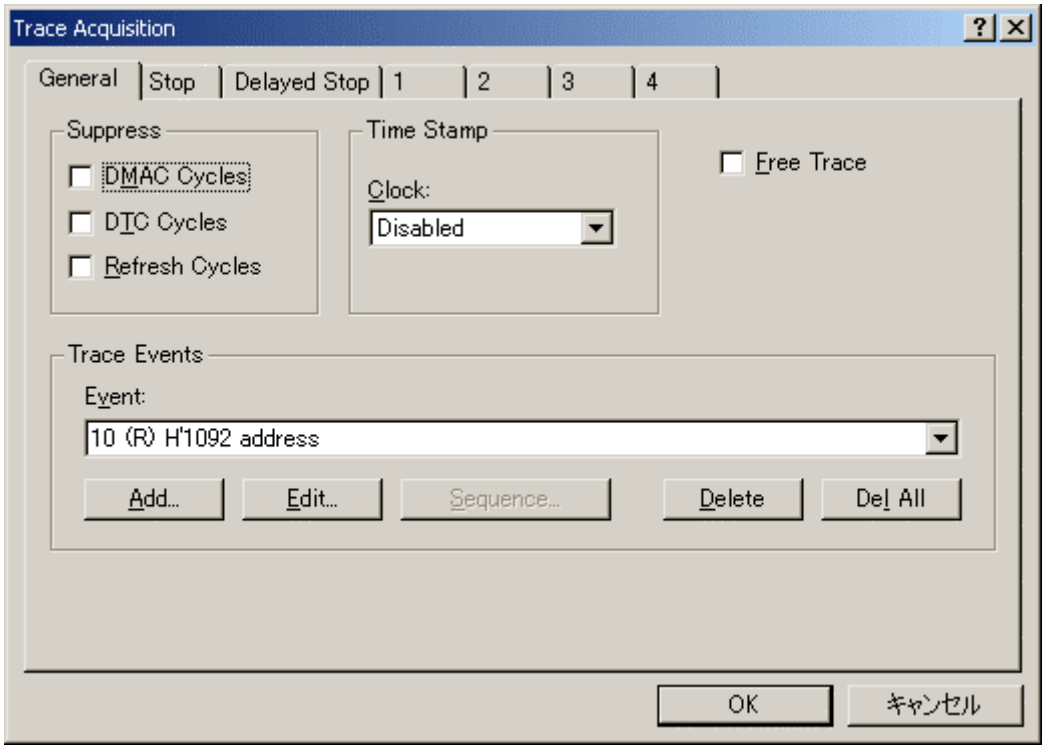


図 5-73 Trace Acquisition ダイアログボックス(General ページ)

[Suppress] 指定したバスサイクルタイプのトレース情報を取得しません。

[Time Stamp] タイムスタンプの条件を設定します。

[Clock] タイムスタンプの分解能を設定します。
分解能は以下のいずれかから選択できます。
Disabled, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us, 100us
タイムスタンプは 32 ビットのカウンタで構成されています。
測定可能な最大時間は分解能 125ns で約 9 分、分解能 100us で約 5 日です。
カウンタがオーバフローした場合、カウンタをクリアし、カウントアップを続けます。
Disabled を選択した場合はタイムスタンプ情報を取得しません。

[Free Trace] チェックするとフリートレースモードを有効にします。

フリートレースモード有効の場合

プログラム実行開始直後にデータ取得を開始します。フリートレースモードではトレース停止条件の設定のみ可能です。範囲トレースは利用不可となり、4 つの範囲トレースページ(1〜4)は無効になります。

フリートレースモード無効の場合

トレース取得の開始および停止条件を設定できます。

[Trace Events] トレース情報取得条件で使用するイベントポイントを設定します。
- [Event] トレース情報取得条件を設定したイベントポイントのリストです。
- [Add...] 新しいイベントポイントを追加します。
- [Edit...] Event で選択したイベントポイントの設定を変更します。
- [Sequence...] トレース情報取得条件で使用するイベントポイントに対しイベントシーケンスを構成します。先にイベントを設定していないと、シーケンスのセットアップはできません。
- [Delete] Event で選択したイベントポイントを削除します。
- [Del All] すべてのイベントポイントを削除します。

【注】 1. [Suppress]オプションで指定可能なバスサイクルはご使用のエミュレータにより異なります。詳細につきましては「5.16.4 バス状態およびエリア信号について」をご参照ください。
2. タイムスタンプ情報は一部のトレース情報と共用でトレースバッファを使用します。したがって、タイムスタンプ取得時は PTR, Address, Instruction, Data, R/W, Source, Label, Timestamp および Timestamp-Difference 以外のトレース情報は取得できません。
3. 範囲トレースまたはトレース停止で使用しているイベントが削除されると、範囲トレースまたはトレース停止機能は無効になります。

(2) Stop ページ

トレース停止条件を設定します。
トレース停止条件はディレイの有無にかかわらず、両方同時に指定できます。

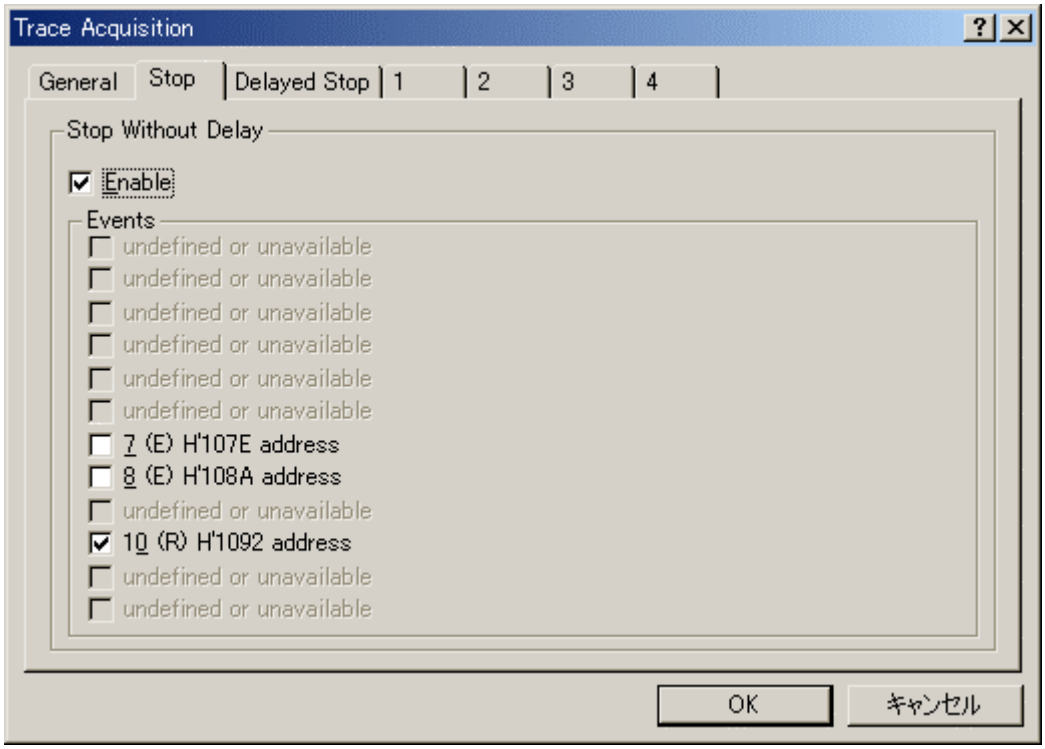


図 5-74 Trace Acquisition ダイアログボックス(Stop ページ)

[Stop Without Delay] トレース停止条件を設定します。
- [Enable] チェックするとトレース停止を有効にします。
- [Events] トレース情報取得条件を設定したイベントポイントを表示します。
イベントポイントに対応したチェックボックスをチェックした場合、該当のイベントが成立したときにトレース情報の取得を停止します。
([Enable]選択時有効)

(3) Delayed Stop ページ

トレース停止条件を設定します。
トレース停止条件はディレイの有無にかかわらず、両方同時に指定できます。

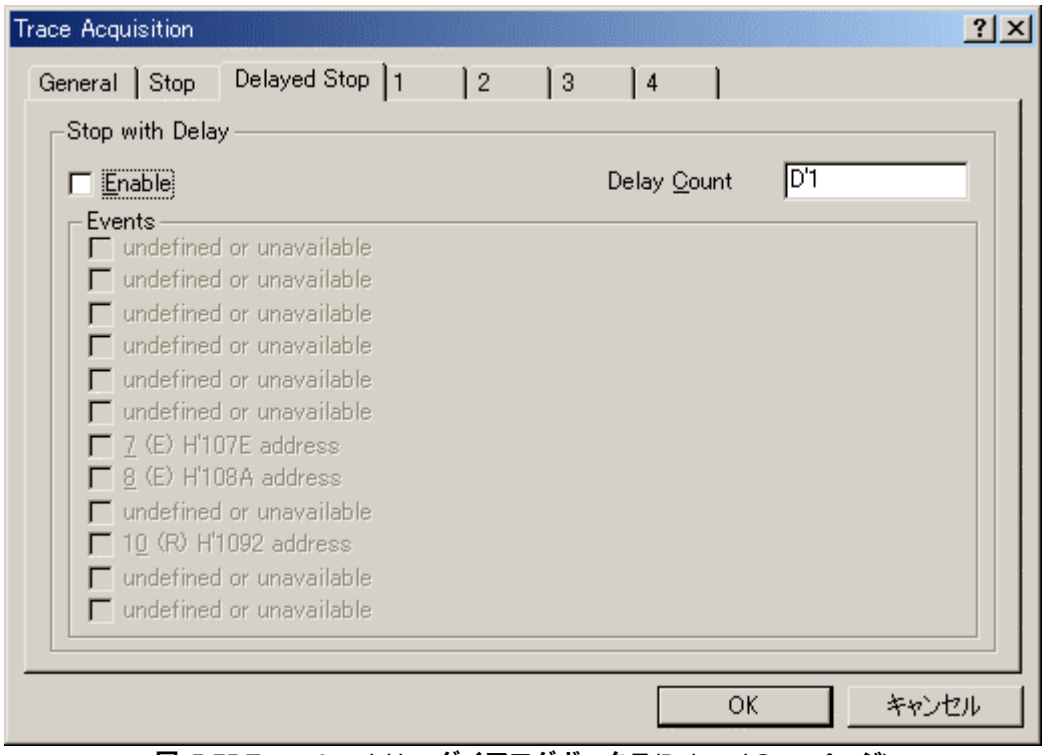


図 5-75 Trace Acquisition **ダイアログボックス**(Delayed Stop **ページ**)

[Stop With Delay] トレース停止条件を設定します。

[Enable] チェックするとトレース停止を有効にします。

[Delay Count] ディレイカウント(バスサイクル 1～65535 の範囲)を設定します。この機能により、イベントが成立した後にいくらかのトレースサイクルを取得します。

[Events] トレース情報取得条件を設定したイベントポイントを表示します。
イベントポイントに対応したチェックボックスをチェックした場合、該当のイベントが成立したときにトレース情報の取得を停止します。
([Enable]選択時有効)

(4) 1, 2, 3, 4 **ページ**

範囲トレースを設定します。フリートレースモード無効時のみ設定可能です。
範囲トレースは[Disabled], [Point to Point], [Range], [Event]の 4 つのモードを選択できます。

・Disabled

範囲トレースを無効にします。

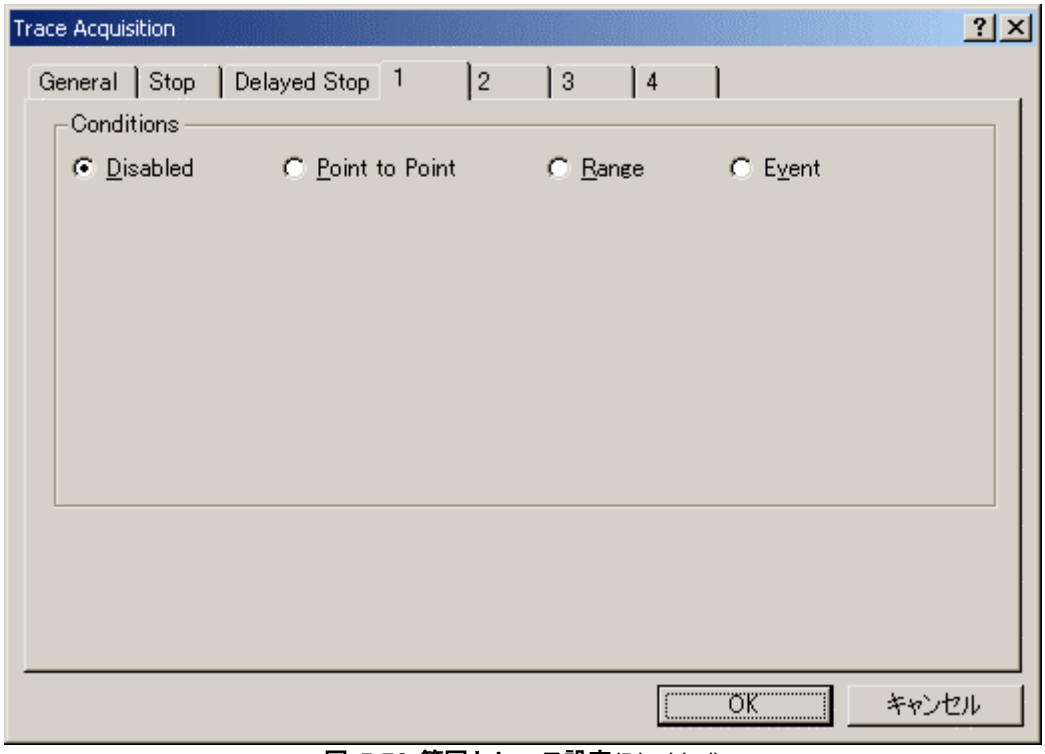


図 5-76 **範囲トレース設定**(Disabled)

• Point to Point

指定アドレス範囲のトレース情報を取得します。

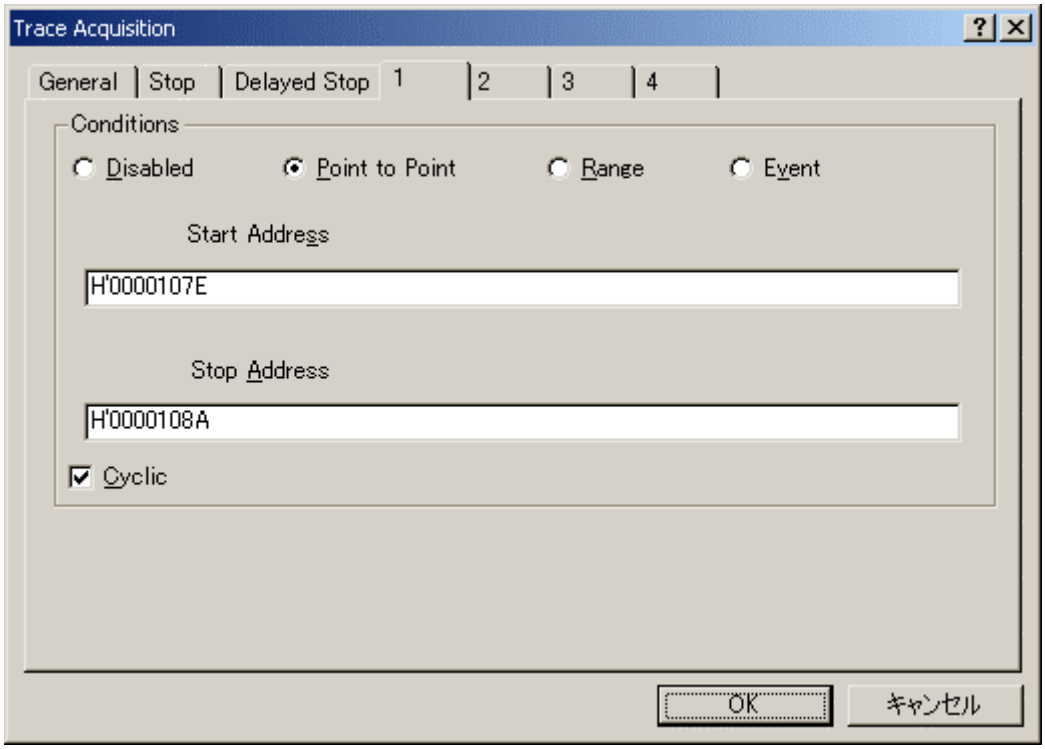


図 5-77 範囲トレース設定(Point to Point)

[Start Address]	トレース情報取得を開始するアドレスを指定します。
[Stop Address]	トレース情報取得を停止するアドレスを指定します。
[Cyclic]	チェックするとイベントシーケンスを構成し、イベント自身がリセットになり、停止イベントの後に開始イベントが成立すると再度トレースを開始します。

開始アドレスをアクセスしたときにトレース情報の取得を開始し、停止アドレスをアクセスしたときにトレース情報の取得を停止するために必要なイベントポイントを設定します。

Point to point モードは Event モードをセットアップする簡単な方法で、開始および停止イベントは単一アドレスをアクセスするイベントです。

[Cyclic]オプションを設定することにより指定アドレス範囲のトレース情報のみを取得し続けることができます。

【注】 本機能はイベントポイントのシーケンスを自動的に構成しますが、予期しない結果が起きる場合もあります。
この場合、[Event Sequencing]**ダイアログボックスにてシーケンス設定を修正してください。**

• Range

指定した条件に合うトレース情報のみを取得します。

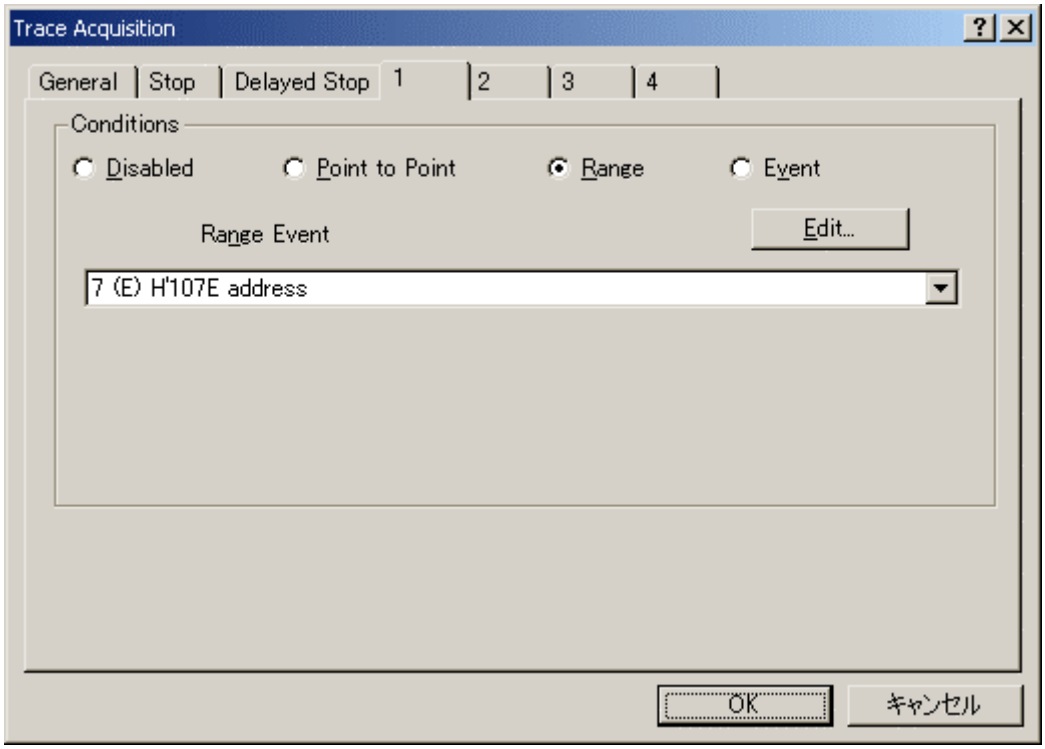


図 5-78 範囲トレース設定(Range)

[Range Event] トレース情報の取得条件を設定したイベントポイントを指定します。
[Edit...] 指定するイベントポイントの設定を変更します。

すべてのバスサイクルから選択したイベントの条件に合うトレース情報のみを取得します。
このモードは１つのイベントチャネルまたは範囲チャネルを使用します。

・Event

指定した条件によりトレース情報取得の開始と終了を制御しトレース情報を取得します。

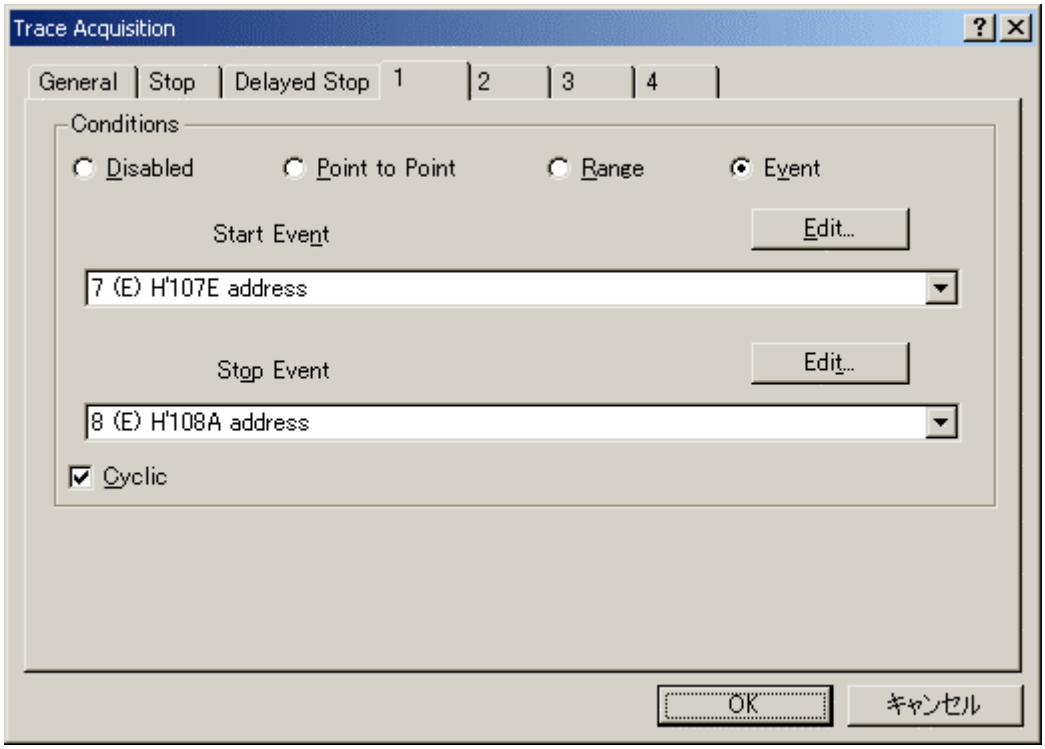


図 5-79 範囲トレース設定(Event)

[Start Event]	トレース情報取得を開始する条件を設定したイベントポイントを指定します。
[Stop Event]	トレース情報取得を停止する条件を設定したイベントポイントを指定します。
[Edit...]	指定するイベントポイントの設定を変更します。
[Cyclic]	チェックするとイベントシーケンスを構成し、イベント自身がリセットになり、停止イベントの後に開始イベントが成立すると再度トレースを開始します。

開始条件が成立したときにトレース情報の取得を開始し、停止条件が成立したときにトレース情報の取得を停止します。

[Cyclic]オプションを設定することにより指定条件により取得可能なトレース情報のみを取得し続けることができます。

5.17.4 Trace レコードを検索する

トレースレコードを検索するには[Trace Find]ダイアログボックスを使用します。
[Trace Find]ダイアログボックスを開くには、ポップアップメニューの[検索...]を選択します。

[Trace Find]ダイアログボックスは下記ページより構成されています。

表5-5 [Trace Find]**ダイアログボックスのページ構成**

ページ	設定項目
[General]	**検索範囲を指定します。**
[Address]	**アドレス条件を指定します。**
[Data]	**データ条件を指定します。**
[R/W]	**アクセスサイクルの種類を指定します。**
[Area]	**アクセスしたエリアを指定します。** **(タイムスタンプ取得時は無効)**
[Status]	**バス状態を指定します。** **(タイムスタンプ取得時は無効)**
[Probes]	4 **本のプローブ信号の状態を指定します。** **(タイムスタンプ取得時は無効)**
[IRQ7-0]	8 **本の** IRQ **入力の状態を指定します。** **(タイムスタンプ取得時は無効)**
[Timestamp]	**バスサイクルのタイムスタンプを指定します。** **(タイムスタンプ取得時のみ有効)**

【注】 [General], [Address], [Data], [R/W], [Area], [Status], [Probes], [Timestamp]**以外の項目はご使用のエミュレータにより異なります。**
詳細につきましては「8 **本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。**

各ページで条件を設定し、[OK]ボタンをクリックすることにより、サーチ条件を設定し、検索を開始します。[キャンセル]ボタンをクリックすると、設定しないでダイアログボックスを閉じます。
検索の結果一致するトレースレコードが見つかった場合は当該レコード行を強調表示します。一致するトレースレコードが見つからなかった場合は、メッセージダイアログボックスを表示します。
トレース情報の検索は各ページで設定した条件がすべて一致するトレース情報のみを検索します。

トレースレコードが検索できた場合は、ポップアップメニューで[次を検索]を選択すると、次のトレースレコードを検索できます。

(1) General ページ

検索範囲を指定します。

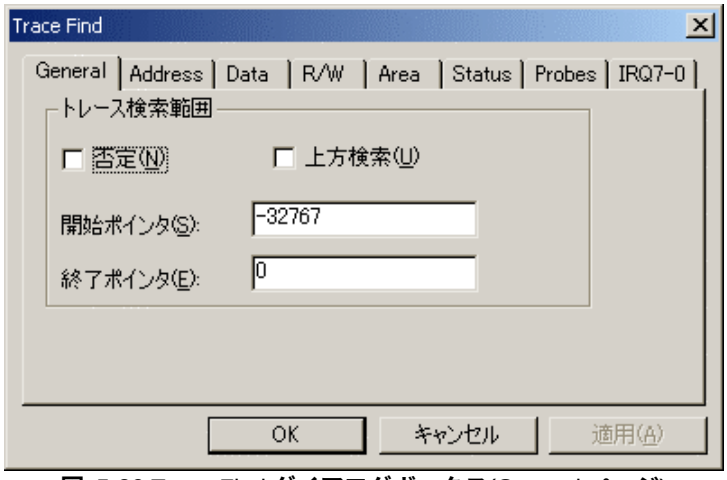


図 5-80 Trace Find ダイアログボックス(General ページ)

[トレース検索範囲] 検索範囲を指定します。
[否定] チェックすると他のページで設定した項目の否定条件で検索します。
[上方検索] チェックすると上方検索を行います。
[開始ポインタ] 検索を開始する PTR の値を入力します。
[終了ポインタ] 検索を終了する PTR の値を入力します。

【注】 検索範囲入力時、[開始ポインタ]オプションに検索を終了する PTR の値、 [終了ポインタ]オプションに検索を開始する PTR の値を指定することも可能です。

(2) Address ページ

アドレス条件を指定します。

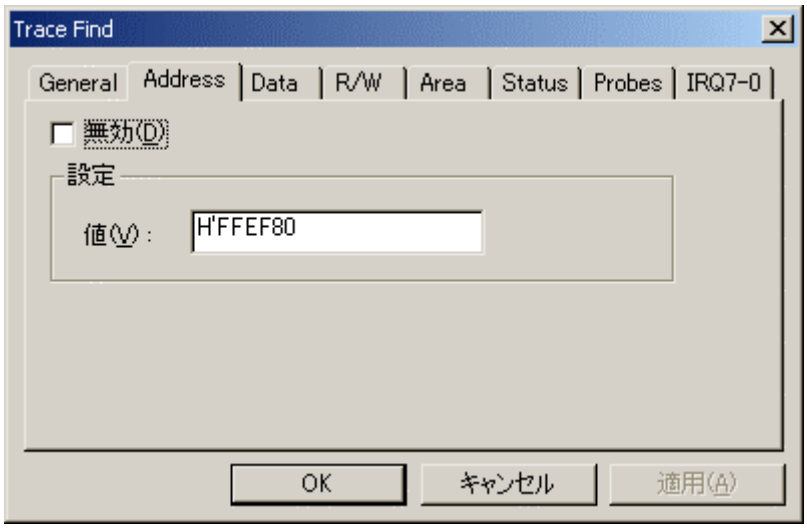


図 5-81 Trace Find ダイアログボックス(Address ページ)

[無効] チェックすると、アドレスを検出しません。
[設定] 指定したアドレスを検出します。
[値] アドレス値を入力します。
([無効]選択時無効)

(3) Data ページ

データ条件を指定します。

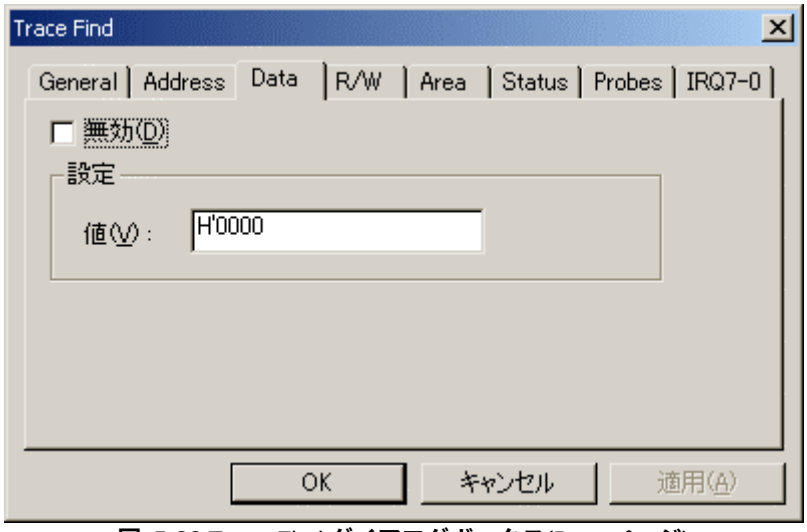


図 5-82 Trace Find **ダイアログボックス**(Data **ページ**)

[無効] チェックすると、データを検出しません。
[設定] 指定したデータを検出します。
[値] データ値を入力します。
([無効]選択時無効)

(4) R/W ページ

アクセスサイクルの種類を指定します。

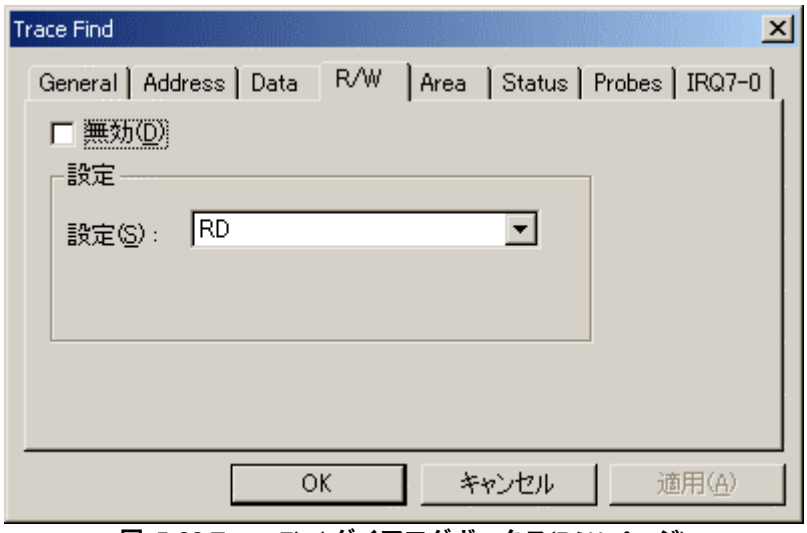


図 5-83 Trace Find **ダイアログボックス**(R/W **ページ**)

[無効] チェックすると、リード/ライト条件を検出しません。
[設定] 指定したリード/ライト条件を検出します。
[設定] リード/ライト条件を選択します。
([無効]選択時無効)
RD: リードサイクル
WR: ライトサイクル

(5) Area ページ

アクセスしたエリアを指定します。
タイムスタンプ取得時は無効です。

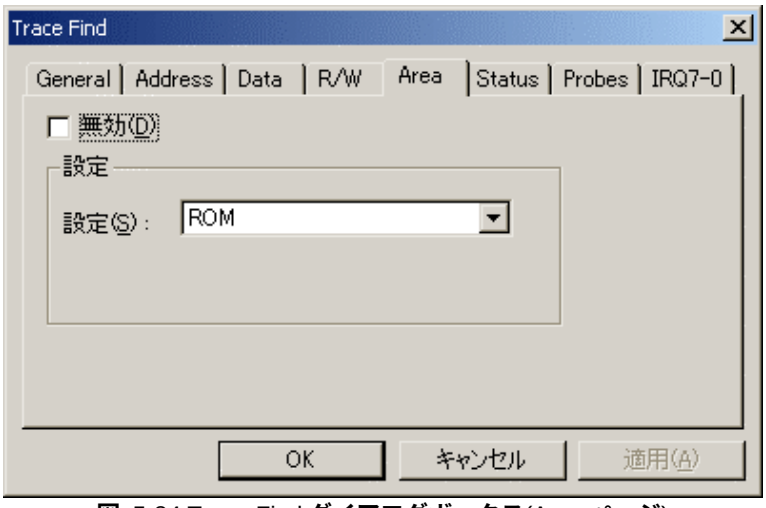


図 5-84 Trace Find ダイアログボックス(Area ページ)

[無効] チェックすると、エリア条件を検出しません。
[設定] 指定したエリア条件を検出します。
[設定] エリア条件を選択します。
([無効]選択時無効)

【注】 指定可能なエリア条件はご使用のエミュレータにより異なります。
詳細につきましては「5.16.4 バス状態およびエリア信号について」をご参照ください。

(6) Status ページ

バス状態を指定します。
タイムスタンプ取得時は無効です。

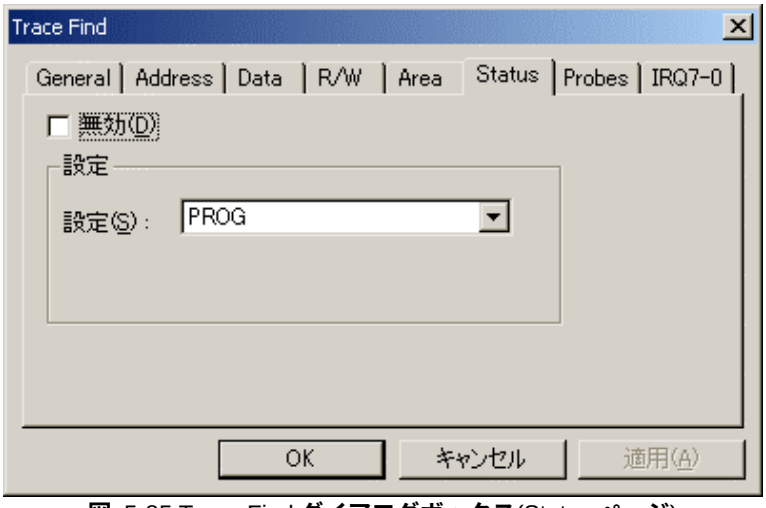


図 5-85 Trace Find ダイアログボックス(Status ページ)

[無効] チェックすると、バス条件を検出しません。
[設定] 指定したバス条件を検出します。
[設定] バス条件を選択します。
([無効]選択時無効)

【注】 指定可能なバス条件はご使用のエミュレータにより異なります。
詳細につきましては「5.16.4 バス状態およびエリア信号について」をご参照ください。

(7) Probes ページ

4 本のプローブ信号の状態を指定します。
タイムスタンプ取得時は無効です。

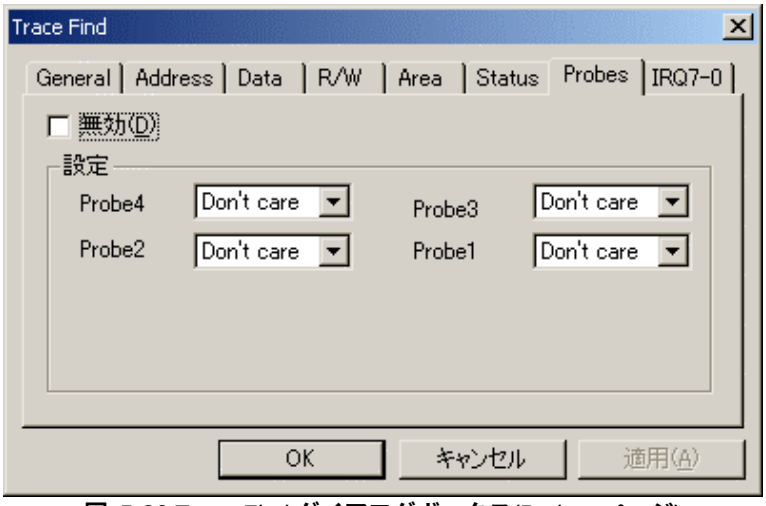


図 5-86 Trace Find **ダイアログボックス**(Probes **ページ**)

[無効] チェックすると、プローブ条件を検出しません。
[設定] 指定したプローブ条件を検出します。
[Probe4]～[Probe1] プローブ条件を選択します。
([無効]選択時無効)
Don't care: 選択したプローブ条件を検出しません
High: プローブ信号の状態が HIGH
Low: プローブ信号の状態が LOW

(8) IRQ7-0 ページ

IRQ 入力の状態を指定します。
タイムスタンプ取得時は無効です。

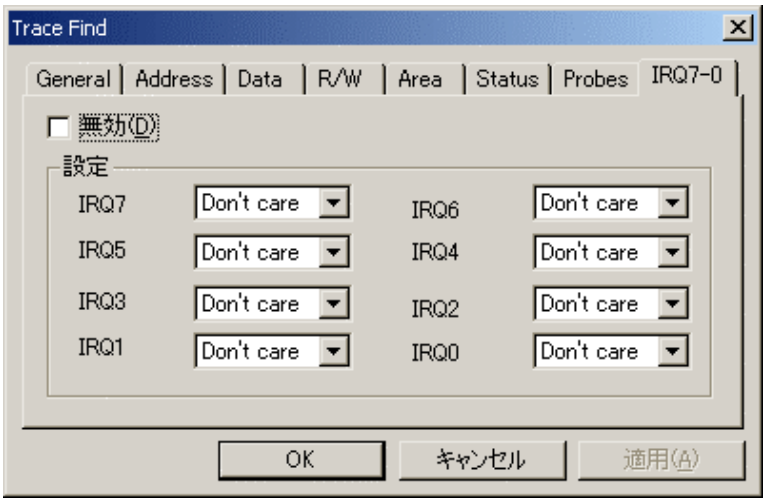


図 5-87 Trace Find **ダイアログボックス**(IRQ7-0 **ページ**)

[無効] チェックすると、IRQ 入力の条件を検出しません。
[設定] 指定した IRQ 入力の条件を検出します。
[IRQ7]～[IRQ0] IRQ 入力の条件を選択します。
([無効]選択時無効)
Don't care: 選択した IRQ 入力の条件を検出しません
High: IRQ 入力の状態が HIGH
Low: IRQ 入力の状態が LOW

(9) Timestamp ページ

バスサイクルのタイムスタンプを指定します。
タイムスタンプ取得時のみ有効です。

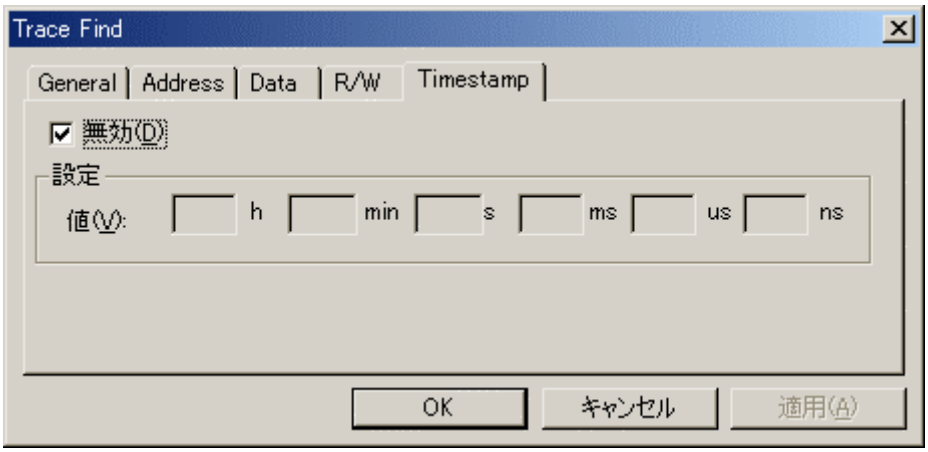


図 5-88 Trace Find ダイアログボックス(Timestamp ページ)

[無効] チェックすると、タイムスタンプ値を検出しません。
[設定] 指定したタイムスタンプ値を検出します。（全ての桁の入力が必須）
[値] タイムスタンプ値を入力します。
このフォーマットは以下の通りです。
[時] h [分] min [秒] s [ミリ秒] ms [マイクロ秒] us [ナノ秒] ns
([無効]選択時無効)

5.17.5 トレース情報をクリアする

トレース情報をクリアするには、ポップアップメニューから[クリア]を選択します。トレース情報を保持しているトレースバッファを空にします。複数の[Trace]ウィンドウが開いているときは、それらは同じバッファをアクセスしているため、すべての[Trace]ウィンドウをクリアすることになります。

5.17.6 トレース情報をファイルに保存する

トレース情報をファイルに保存するには、ポップアップメニューから[保存...]を選択します。
[名前を付けて保存]ダイアログボックスを表示します。[Trace]ウィンドウに表示しているトレース情報をテキストファイルとして保存します。保存する範囲を、[PTR]の範囲によって指定することができます(すべてのバッファをセーブするには、数分かかることがあります)。このファイルは保存のみ可能で、[Trace]ウィンドウへの読み込みはできないことに注意してください。

【注】 トレース情報をフィルタリングした場合、保存する範囲の指定はできません。フィルタリングした結果[Trace]ウィンドウに表示されたトレース情報すべてを保存します。保存する範囲を指定したい場合は[Trace Filter]ダイアログボックスの[General]ページよりフィルタ範囲を指定してください。
フィルタ機能については「5.17.12 取得したトレース情報から必要なレコードを抽出する」を参照してください。

5.17.7 ソースウィンドウを表示する

トレースレコードに対応する[ソース]ウィンドウを表示するには二通りの方法があります。

(1) **トレースレコードを選択した状態でポップアップメニューから[ソースファイル表示]を選択する**

(2) **トレースレコードをダブルクリックする**

上記の操作により、[ソース]ウィンドウあるいは[逆アセンブリ]ウィンドウを開いてソース表示し、選択した行をカーソルで示します。

5.17.8 ソース表示を整形する

ポップアップメニューで[ソーストリム]を選択すると、ソースプログラムの左側の空白を取り除きます。
取り除いた状態だと[ソーストリム]メニューの左にチェックが付きます。チェックありの状態で[ソーストリム]メニューを選択すると取り除いた空白を元に戻します。

5.17.9 トレース情報のスナップショットを取得する

ユーザプログラム実行中、一時的にトレース情報を確認したい場合はスナップショットを取得します。
タイムスタンプの確認や、プローブの入力チェックなどに便利です。
トレース情報のスナップショットを取得するにはポップアップメニューから[スナップショット]を選択します。
トレース情報の取得を一時中断し、最新のトレース情報 1 レコードを表示した後、トレース情報の取得を再開します。
トレース情報のスナップショットはユーザプログラム実行中のみ取得できます。

5.17.10 トレース情報の取得を一時的に停止する

ユーザプログラム実行中、一時的にトレース情報の取得を停止するにはポップアップメニューから[停止]を選択します。
トレース取得を中止し、トレース表示を更新します。
ユーザプログラムを停止せずにトレース情報の取得のみ停止し、トレース情報を確認する場合などに使用します。

5.17.11 トレース情報の取得を再開する

ユーザプログラム実行中、一時的にトレース情報の取得を停止した場合、再度トレース情報の取得を再開するにはポップアップメニューから[リスタート]を選択します。

5.17.12 取得したトレース情報から必要なレコードを抽出する

取得したトレース情報から必要なレコードのみを抽出するにはフィルタ機能を使用します。
フィルタ機能はハードウェアにより取得したトレース情報をソフトウェアによりフィルタリングします。
取得条件を設定してトレース情報を取得する[Trace Acquisition]設定と異なり、取得したトレース情報に対し何度もフィルタ設定を変更することで必要な情報が簡単に抽出でき、データの分析に役立ちます。
フィルタ機能を使用してもトレースバッファの内容は変更されません。
トレースバッファは有限ですので、[Trace Acquisition]設定により有用なトレース情報をより多く取得することで、より効果的にデータの分析が可能となります。

フィルタ機能を使用するには[Trace Filter]ダイアログボックスを使用します。
[Trace Filter]ダイアログボックスを開くには、ポップアップメニューの[フィルタ...]を選択します。

[Trace Filter]ダイアログボックスは下記ページより構成されています。

表5-6 [Trace Filter]ダイアログボックスのページ構成

ページ	設定項目
[General]	フィルタ範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Area]	アクセスしたエリアを指定します。 (タイムスタンプ取得時は無効)
[Status]	バス状態を指定します。 (タイムスタンプ取得時は無効)
[Probes]	4 本のプローブ信号の状態を指定します。 (タイムスタンプ取得時は無効)
[IRQ7-0]	8 本の IRQ 入力の状態を指定します。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプを指定します。 (タイムスタンプ取得時のみ有効)

【注】 [General], [Address], [Data], [R/W], [Area], [Status], [Probes], [Timestamp]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

各ページでフィルタ条件を設定し、[OK]ボタンをクリックすることにより、フィルタ条件にしたがいフィルタリングを行います。[キャンセル]ボタンをクリックすると、[Trace Filter]ダイアログボックスを開いた時点の設定のままダイアログボックスを閉じます。
フィルタリングは各ページで設定したフィルタ条件が 1 つ以上一致するトレース情報のみを[Trace]ウィンドウに表示します。

フィルタリングを行ってもトレースバッファの内容は変更されませんので、何度もフィルタ条件を変更しデータの分析ができます。

(1) General ページ

フィルタ範囲を指定します。

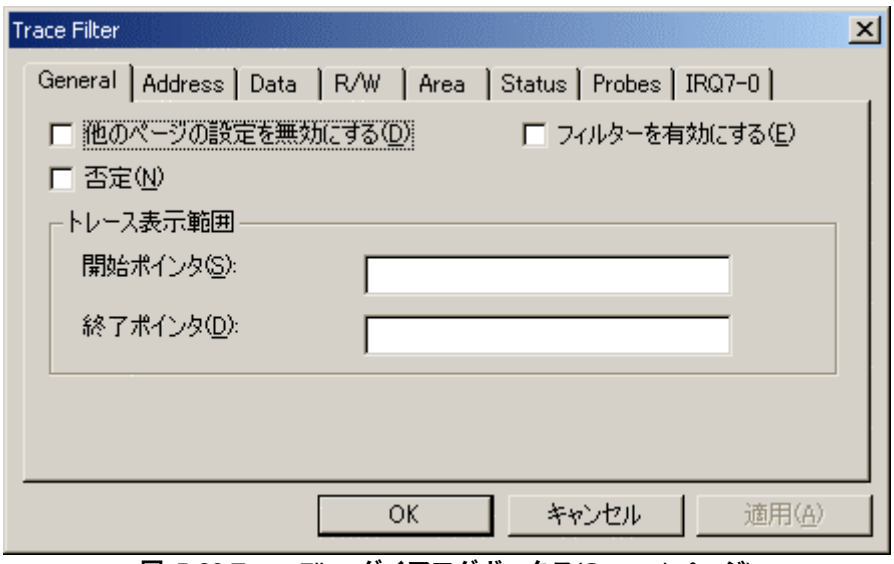


図 5-89 Trace Filter ダイアログボックス(General ページ)

[他のページ設定を無効にする] チェックすると、サイクル番号のみ指定できます。他のオプションはすべて無効になります。
[フィルターを有効にする] チェックすると、フィルタを有効にします。
[否定] チェックすると他のページで設定した項目の否定条件でフィルタリングします。
[トレース表示範囲] フィルタ範囲を指定します。
[開始ポインタ] フィルタを開始する PTR の値を入力します。
[終了ポインタ] フィルタを終了する PTR の値を入力します。

【注】 フィルタ範囲入力時、[開始ポインタ]オプションにフィルタを終了する PTR の値、 [終了ポインタ]オプションにフィルタを開始する PTR の値を指定することも可能です。

(2) Address ページ

アドレス条件を指定します。

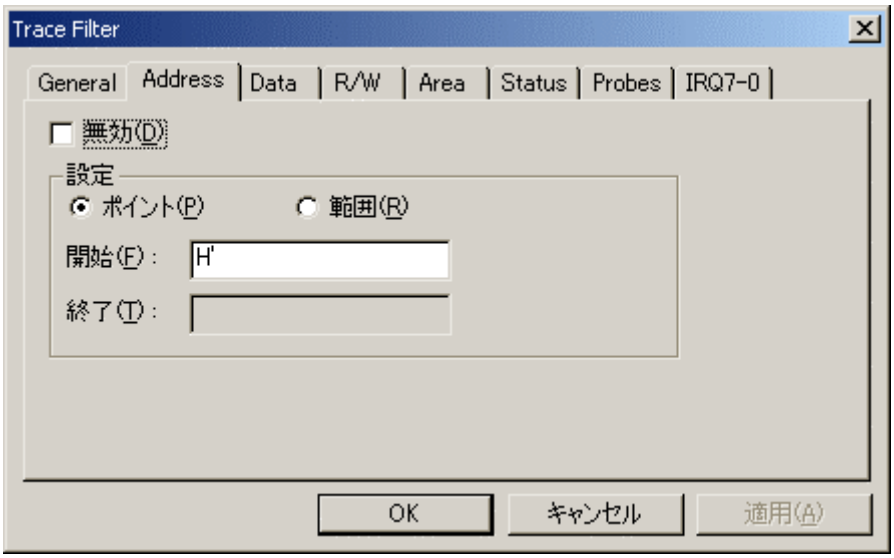


図 5-90 Trace Filter ダイアログボックス(Address ページ)

[無効] チェックすると、アドレスを検出しません。
[設定] 指定したアドレスを検出します。
[ポイント] 単一アドレスを指定します。
([無効]選択時無効)
[範囲] アドレス範囲を指定します。
([無効]選択時無効)
[開始] 単一アドレスまたはアドレス範囲の開始アドレスを入力します。
([無効]選択時無効)
[終了] アドレス範囲の終了アドレスを入力します。
([範囲]選択時有効)

【注】 アドレス範囲入力時、[開始]オプションにアドレス範囲の終了アドレス、[終了]オプションにアドレス範囲の開始アドレスを指定することも可能です。

(3) Data ページ

データ条件を指定します。

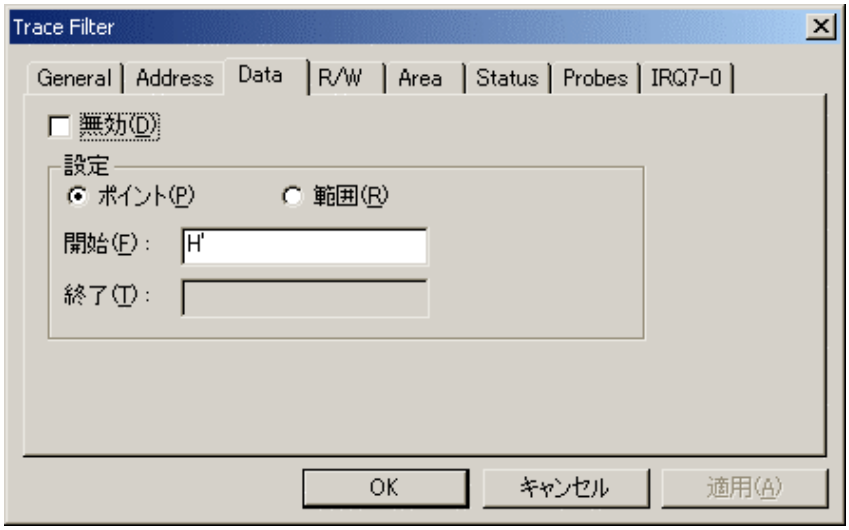


図 5-91 Trace Filter ダイアログボックス(Data ページ)

[無効] チェックすると、データを検出しません。

[設定] 指定したデータを検出します。

[ポイント] 単一データを指定します。
([無効]選択時無効)

[範囲] アドレス範囲を指定します。
([無効]選択時無効)

[開始] 単一データまたはデータ範囲の最小値を入力します。
([無効]選択時無効)

[終了] データ範囲の最大値を入力します。
([範囲]選択時有効)

【注】 データ範囲入力時、[開始]オプションにデータの最大値、[終了]オプションにデータの最小値を指定することも可能です。

(4) R/W ページ

アクセスサイクルの種類を指定します。

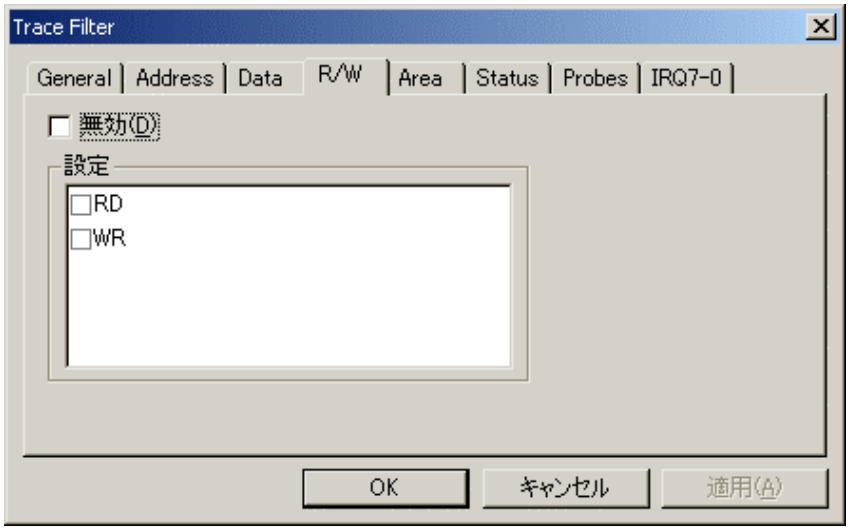


図 5-92 Trace Filter ダイアログボックス(R/W ページ)

[無効] チェックすると、リード/ライト条件を検出しません。
[設定] 指定したリード/ライト条件を検出します。
[RD] チェックすると、リードサイクルを検出します。
([無効]選択時無効)
[WR] チェックすると、ライトサイクルを検出します。
([無効]選択時無効)

(5) Area ページ

アクセスしたエリアを指定します。
タイムスタンプ取得時は無効です。

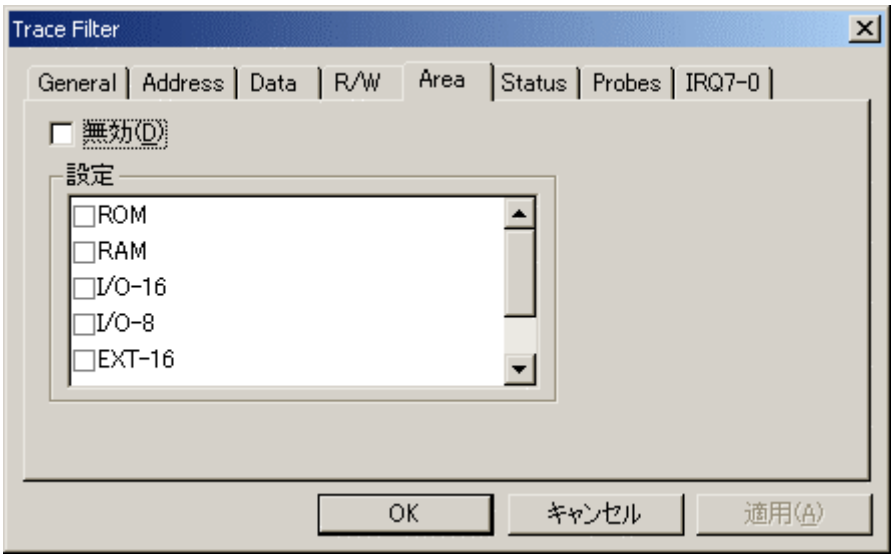


図 5-93 Trace Filter **ダイアログボックス**(Area **ページ**)

[無効] チェックすると、エリア条件を検出しません。
[設定] 指定したエリア条件を検出します。
([無効]選択時無効)

【注】 指定可能なエリア条件はご使用のエミュレータにより異なります。
詳細につきましては「5.16.4 **バス状態およびエリア信号について」をご参照ください。**

(6) Status ページ

バス状態を指定します。
タイムスタンプ取得時は無効です。

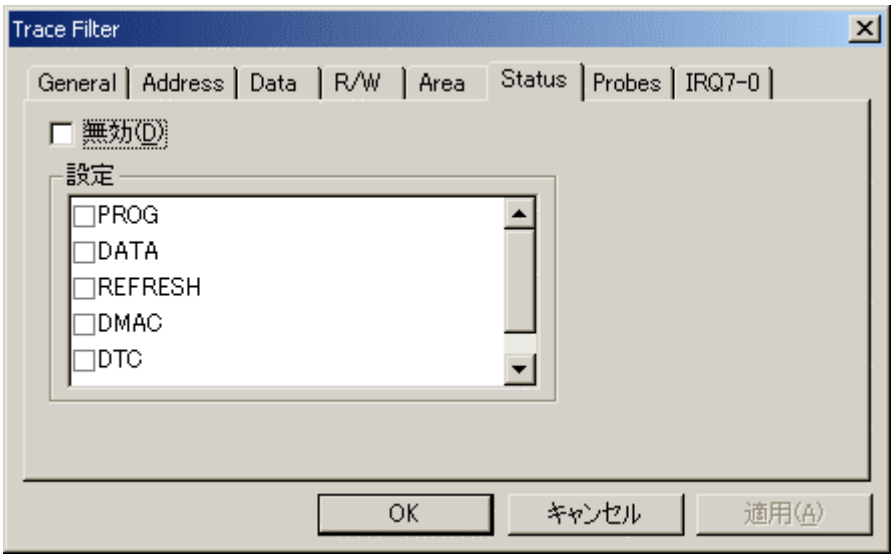


図 5-94 Trace Filter ダイアログボックス(Status ページ)

[無効] チェックすると、バス条件を検出しません。
[設定] 指定したバス条件を検出します。
([無効]選択時無効)

【注】 指定可能なバス条件はご使用のエミュレータにより異なります。
詳細につきましては「5.16.4 バス状態およびエリア信号について」をご参照ください。

(7) Probes ページ

4 本のプローブ信号の状態を指定します。
タイムスタンプ取得時は無効です。

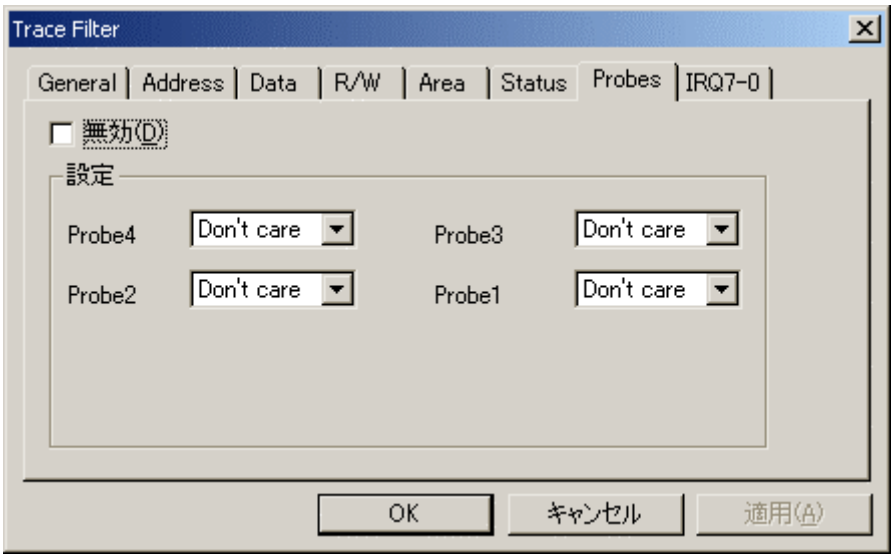


図 5-95 Trace Filter ダイアログボックス(Probes ページ)

[無効]　　チェックすると、プローブ条件を検出しません。
[設定]　　指定したプローブ条件を検出します。
　[Probe4]～[Probe1]　プローブ条件を選択します。
　([無効]選択時無効)
　　Don’t care:　選択したプローブ条件を検出しません
　　High:　プローブ信号の状態が HIGH
　　Low:　プローブ信号の状態が LOW

(8)　IRQ7-0 ページ

IRQ 信号の状態を指定します。
タイムスタンプ取得時は無効です。

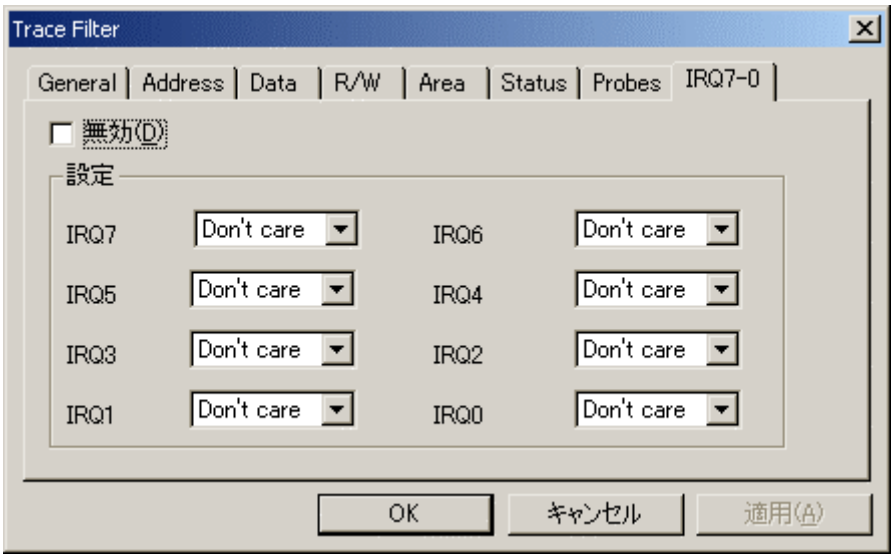


図 5-96 Trace Filter **ダイアログボックス**(IRQ7-0 **ページ**)

[無効]　　チェックすると、IRQ 入力の条件を検出しません。
[設定]　　指定した IRQ 入力の条件を検出します。
　[IRQ7]～[IRQ0]　IRQ 入力の条件を選択します。
　([無効]選択時無効)
　　Don’t care:　選択した IRQ 入力の条件を検出しません
　　High:　IRQ 入力の状態が HIGH
　　Low:　IRQ 入力の状態が LOW

(9) Timestamp ページ

バスサイクルのタイムスタンプを指定します。
タイムスタンプ取得時のみ有効です。

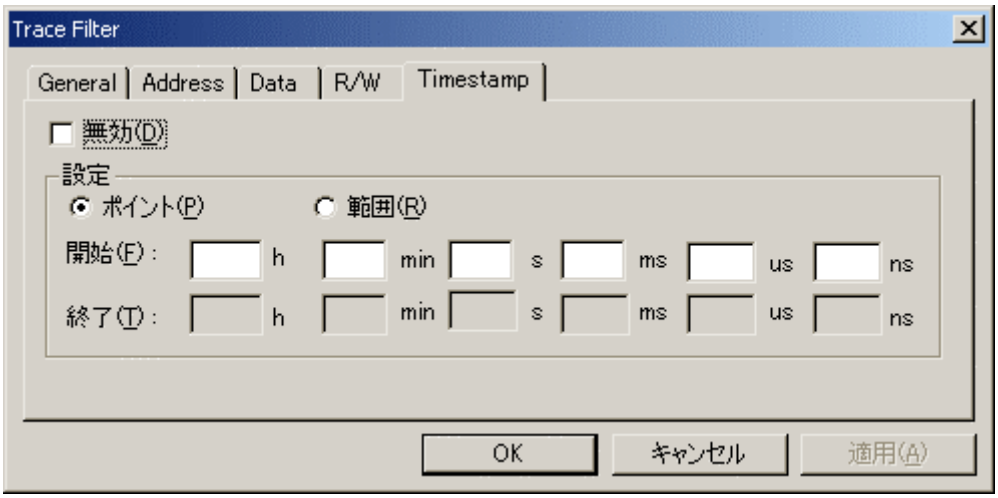


図 5-97 Trace Filter ダイアログボックス(Timestamp ページ)

[無効] チェックすると、タイムスタンプ値を検出しません。

[設定] 指定したタイムスタンプ値を検出します。

- [ポイント] 単一タイムスタンプを指定します。
  ([無効]選択時無効)
- [範囲] タイムスタンプ範囲を指定します。
  ([無効]選択時無効)
- [開始] 単一タイムスタンプまたはタイムスタンプ範囲の最小値を入力します。
  このフォーマットは以下の通りです。
  [時] h [分] min [秒] s [ミリ秒] ms [マイクロ秒] us [ナノ秒] ns
  ([無効]選択時無効)
- [終了] タイムスタンプ範囲の最大値を入力します。
  このフォーマットは以下の通りです。
  [時] h [分] min [秒] s [ミリ秒] ms [マイクロ秒] us [ナノ秒] ns
  ([範囲]選択時有効)

【注】 タイムスタンプ範囲入力時、[開始]オプションにタイムスタンプの最大値、[終了]オプションにタイムスタンプの最小値を指定することも可能です。

5.17.13 タイムスタンプの差を計算する

タイムスタンプ情報取得時、トレース結果より指定した 2 点間の時間差を計算するには、ポップアップメニューから[タイムスタンプ差...]を選択します。

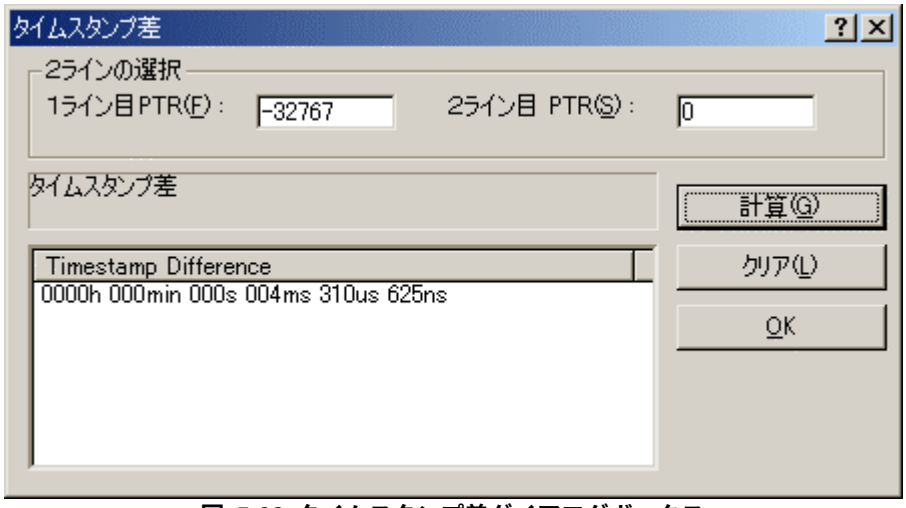


図 5-98 タイムスタンプ差ダイアログボックス

[2 ラインの選択]	タイムスタンプの差を計算するトレースレコードを指定します。
[1 ライン目 PTR]	差分を計測する 1 点目のポインタを指定します。 デフォルト値は、[Trace]ウィンドウ上で選択されているラインのポインタが表示されます。
[2 ライン目 PTR]	差分を計測する 2 点目のポインタを指定します。
[タイムスタンプ差]	2 点間の時間差を計算した結果を表示します。
[計算]	指定した 2 点間の差分を計算し、結果を[タイムスタンプ差]リストに表示します。
[クリア]	[タイムスタンプ差]リストのすべての結果を消去します。
[OK]	ダイアログボックスを閉じます。 この時、[タイムスタンプ差]リストのすべての結果は消去されます。

5.17.14 統計情報を解析する

指定された条件で統計情報の解析を実行するには、ポップアップメニューから[統計...]を選択します。
[統計]ダイアログボックスが開きます。

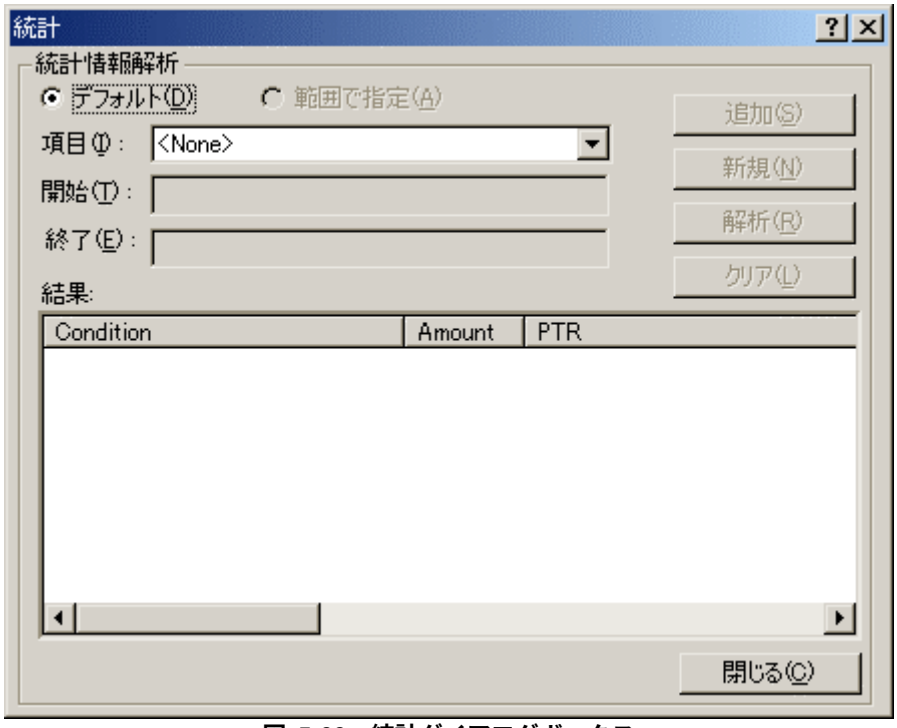


図 5-99 統計ダイアログボックス

[統計情報解析] 統計情報を解析するための設定を行います。
- [デフォルト] 単一の入力値または文字列を指定します。
- [範囲で指定] 入力値または文字列を範囲で指定します。
- [項目] 解析対象項目を指定します。
- [開始] 入力値または文字列を指定します。
  範囲で指定する場合は開始値を設定します。
- [終了] 範囲で指定する場合の終了値を設定します。
  ([範囲で指定]選択時有効)
- [追加] 現在の条件に追加設定します。
- [新規] 新しい条件を指定します。
- [解析] 統計情報解析の結果を取得します。
- [結果] すべての条件と統計情報解析結果を削除します。
- [閉じる] ダイアログボックスを閉じます。
  この時、[結果]リストのすべての結果は消去されます。

本ダイアログボックスは、トレース情報の統計情報解析に使用します。[項目]オプションで解析対象項目を指定し、[開始] オプションおよび[終了]オプションで入力値または文字列を指定します。
[新規]ボタンまたは[追加]ボタンにより条件を設定し[解析]ボタンをクリックすると、統計情報を解析し[結果]リストに解析結果を表示します。

【注】 本エミュレータでは[PTR]項目のみ範囲で指定可能です。それ以外の項目は単一の文字列で指定してください。
統計情報の解析における文字列の判定は[Trace]ウィンドウに表示される文字列と比較し、完全一致したものだけをカウントします。ただし、大文字小文字は区別しません。また、空白の数も考慮しません。

5.17.15　取得したトレース情報から関数呼び出し箇所を抽出する

取得したトレース情報から関数呼び出し箇所のみを抽出するには、ポップアップメニューから[関数コール...]を選択します。

[関数コール箇所の表示]ダイアログボックスが開きます。

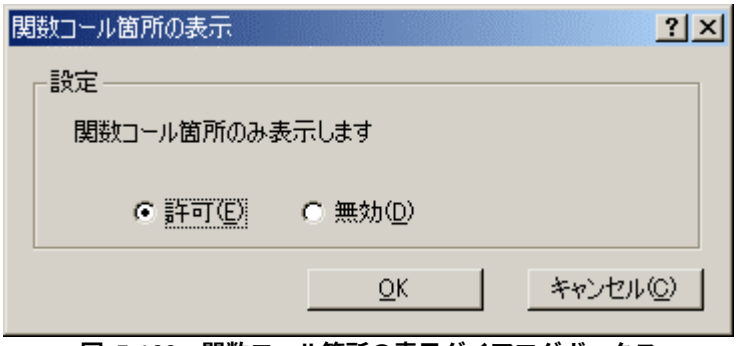


図 5-100　関数コール箇所の表示ダイアログボックス

[設定]　　関数呼び出し箇所の抽出を行うかどうか設定します。
　[許可]　　関数呼び出し箇所の抽出を行います。
　[無効]　　関数呼び出し箇所の抽出を行いません。

[許可]オプションを選択した場合、取得したトレース情報より関数呼び出しを行っているサイクルのみを抽出し表示します。関数呼び出し箇所の抽出を行ってもトレースバッファの内容は変更されません。

フリートレースのトレース結果または、関数の呼び出しを含んだトレース情報に対して本機能を使用することにより、関数の呼び出し順序を調べることができます。

5.18 関数呼び出し履歴を見る

関数呼び出し履歴を表示する場合は、[スタックトレース]ウィンドウを使用します。

5.18.1 スタックトレースウィンドウを開く

[スタックトレース]ウィンドウを開くには、[表示->コード->スタックトレース]を選択するか、[スタックトレース]ツールバーボタンをクリックします。

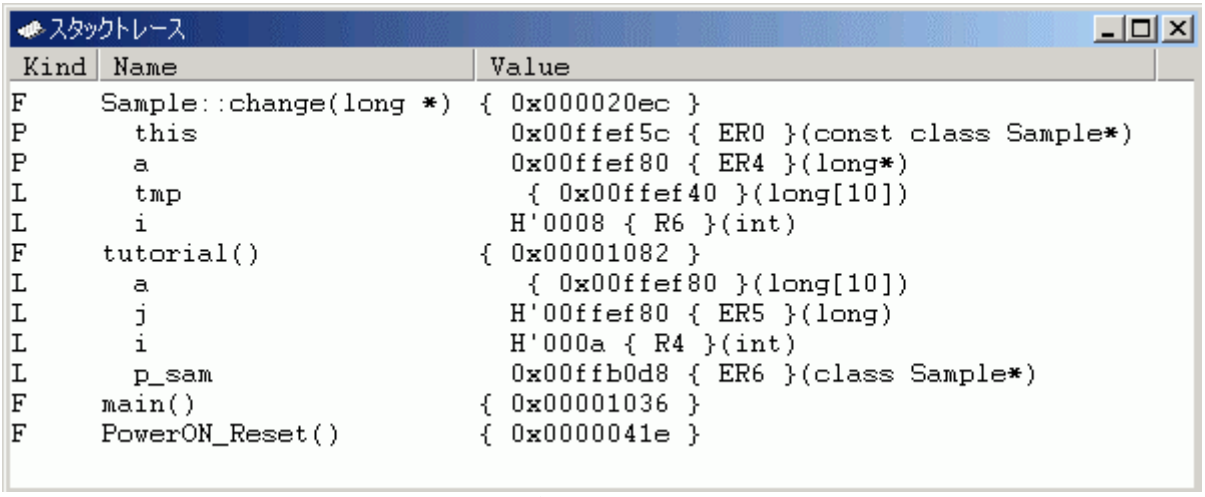


図 5-101 スタックトレースウィンドウ

表示する項目は以下の通りです。

[Kind]	該当シンボルのシンボル種別を示します。 F：関数 P：関数パラメータ L：ローカル変数
[Name]	シンボル名を示します。
[Value]	シンボルの値、アドレス、型を示します。

5.18.2 ソースプログラムを表示する

関数を選択した状態で、ポップアップメニューから[ソースファイル表示]を選択すると、選択した関数に該当するソースプログラムを[ソース]ウィンドウ上に表示します。

5.18.3　表示形式を設定する

ポップアップメニューから[表示設定...]を選択すると、[スタックトレース表示設定]ダイアログボックスを表示します。[スタックトレース]ウィンドウの表示形式を設定します。

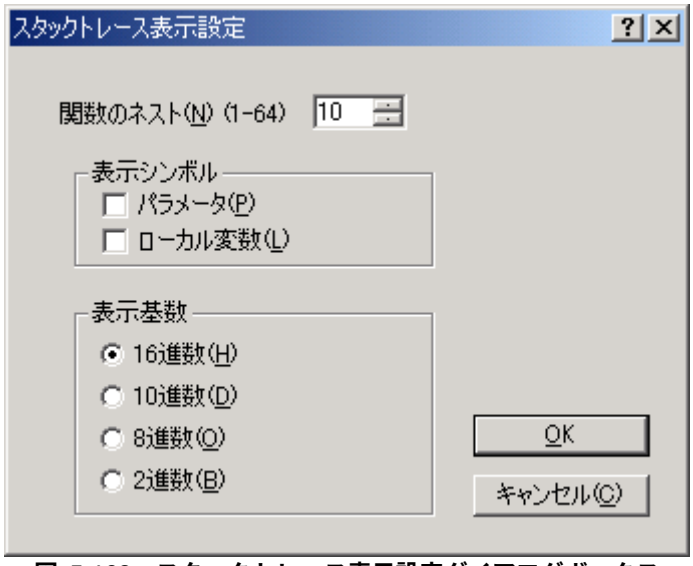


図 5-102　スタックトレース表示設定ダイアログボックス

[関数のネスト]　[スタックトレース]ウィンドウに表示する関数コールネスト数を指定します。
[表示シンボル]　関数以外に表示するシンボルを指定します。
[表示基数]　[スタックトレース]ウィンドウの表示基数を指定します。

5.19 メモリ内容を画像形式で表示する

[画像]ウィンドウを使用すると、メモリ内容を画像形式で表示することができます。

5.19.1 画像ウィンドウを開く

[表示->グラフィック->画像...]を選択するか、[画像]ツールバーボタン をクリックすると、[画像プロパティ]ダイアログボックスが開きます。

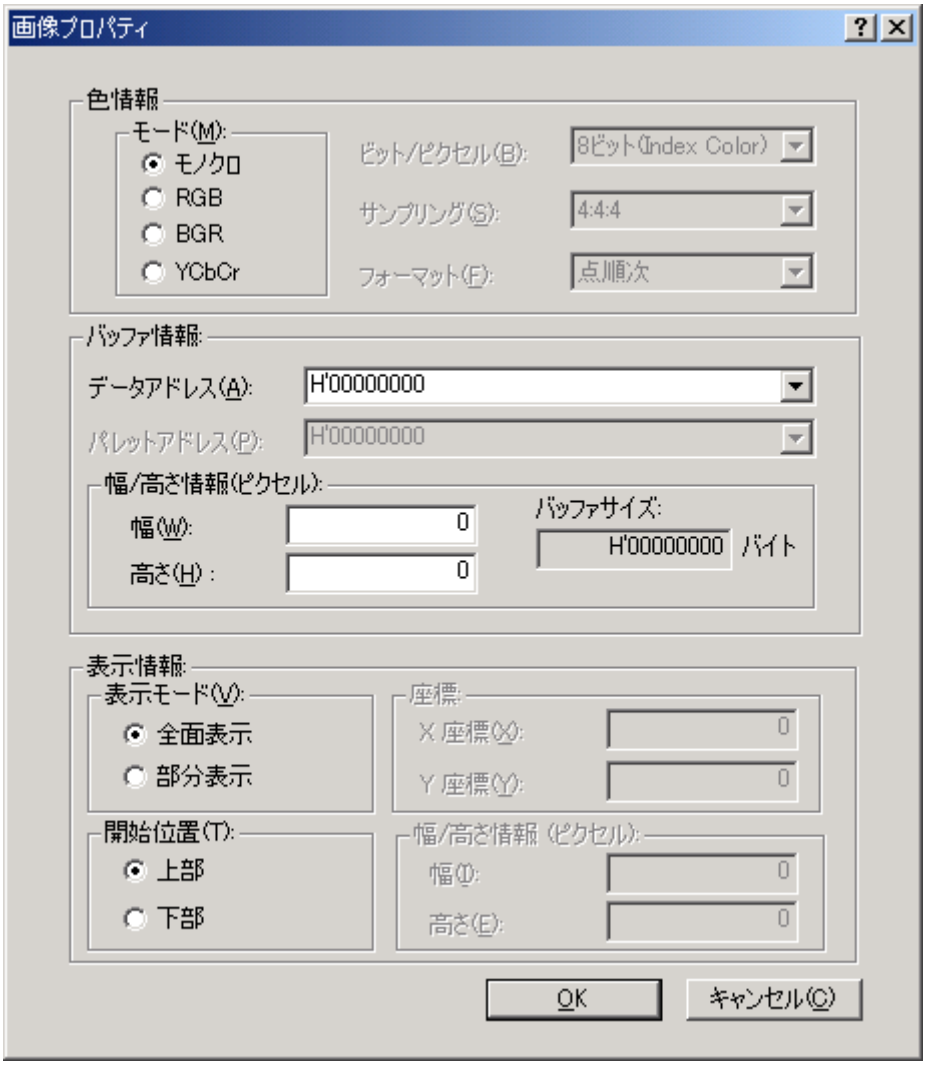


図 5-103 画像プロパティダイアログボックス

[画像プロパティ]ダイアログボックスでは[画像]ウィンドウの表示方法を指定します。

[色情報] 表示する画像のカラー情報を指定します。
- [モード] フォーマットを指定します。
  - [モノクロ] 白黒で表現します。
  - [RGB] R(赤)、G(緑)、B(青)で表現します。
  - [BGR] B(青)、G(緑)、R(赤)で表現します。
  - [YCbCr] Y(輝度)、Cb(青色の色差)、Cr(赤色の色差)で表現します。
- [ビット/ピクセル] 選択した[モード]によって、ビット/ピクセルを指定します。(RGB/BGR 選択時有効)
- [サンプリング] サンプリングのフォーマットを指定します。(YCbCr 選択時有効)
- [フォーマット] 点順次/面順次を指定します。(YCbCr 選択時有効)

[バッファ情報]　　データの格納場所、サイズ、パレットのアドレスを指定します。
[データアドレス]　　表示する画像データのメモリ開始アドレスを指定します。(16 進表示)
[パレットアドレス]　　カラーパレットデータのメモリ開始アドレスを指定します。(16 進表示)
(RGB/ BGR の 8Bit 選択時有効)
[幅/高さ情報(ピクセル)]　　画像の幅と高さを指定します。
[幅]　　画像の幅を指定します。(接頭辞省略時は 10 進で入力、10 進表示)
[高さ]　　画像の高さを指定します。(接頭辞省略時は 10 進で入力、10 進表示)
[バッファサイズ]　　幅と高さから画像のバッファサイズを表示します。(16 進表示)
[表示情報]　　画像全体中の表示部分の位置、サイズ、データ開始位置を指定します
[表示モード]　　画像の全体表示/部分表示を指定します。
[全面表示]　　画像を全体表示します。
[部分表示]　　画像を部分表示します。
[開始位置]
[上部]　　左上からデータを表示します。
[下部]　　左下からデータを表示します。
[座標]　　部分表示する画像の開始位置を指定します。([部分表示]選択時有効)
[X 座標]　　開始位置の X 座標を指定します。(接頭辞省略時は 10 進で入力、10 進表示)
[Y 座標]　　開始位置の Y 座標を指定します。(接頭辞省略時は 10 進で入力、10 進表示)
[幅/高さ情報(ピクセル)]　　部分表示する画像の幅と高さを指定します。
[幅]　　表示の幅を指定します。(接頭辞省略時は 10 進で入力、10 進表示)
[高さ]　　表示の高さを指定します。(接頭辞省略時は 10 進で入力、10 進表示)

[画像プロパティ]ダイアログボックスに設定後、[OK]ボタンをクリックすると[画像]ウィンドウが開きます。
[画像]ウィンドウ表示後もポップアップメニューの[プロパティ...]を選択することで本ダイアログボックスを表示して表示内容を変更できます。

図 5-104　画像ウィンドウ

メモリの内容を画像で表示します。

5.19.2　ウィンドウを自動更新する

ポップアップメニューから[自動更新]をチェックすると、ユーザプログラム実行停止時に自動的にウィンドウ内容を更新します。

5.19.3　ウィンドウを更新する

ポップアップメニューから[更新]を選択すると、直ちにウィンドウ内容を更新します。

5.19.4 ピクセル情報を表示する

ウィンドウ内をダブルクリックするとマウスポインタの位置のピクセル情報を[ピクセル情報]ダイアログボックスに表示します。

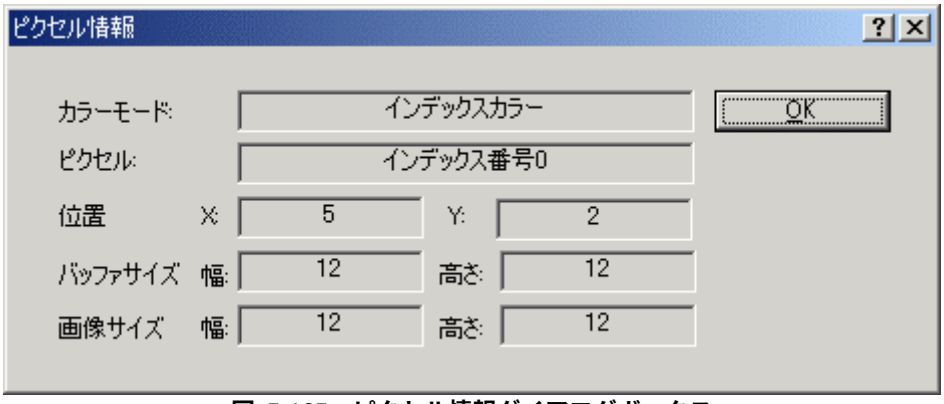


図 5-105 ピクセル情報ダイアログボックス

カーソル位置のピクセル情報を表示します。

[カラーモード]	画像のフォーマットを表示します。	
[ピクセル]	カーソル位置のカラー情報を表示します。（10 進表示）	
[位置]	カーソル位置を X 座標、Y 座標で表示します。（10 進表示）	
	[X]	カーソル位置の X 座標を表示します。
	[Y]	カーソル位置の Y 座標を表示します。
[バッファサイズ]	バッファサイズを表示します。（10 進表示）	
	[幅]	バッファの幅を表示します。
	[高さ]	バッファの高さを表示します。
[画像サイズ]	表示の幅と高さを表示します。（10 進表示）	
	[幅]	幅を表示します。
	[高さ]	高さを表示します。

5.20　メモリ内容を波形形式で表示する

[波形]ウィンドウを使用すると、メモリ内容を波形形式で表示します。

5.20.1　波形ウィンドウを開く

[表示->グラフィック->波形...]を選択するか、[波形]ツールバーボタン をクリックすると、[波形プロパティ]ダイアログボックスが開きます。

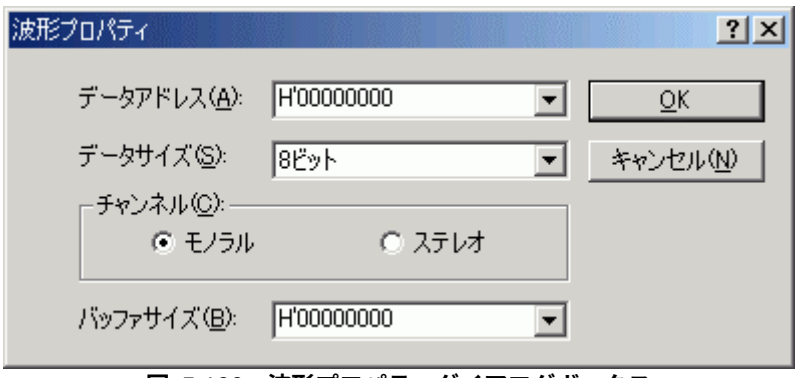


図 5-106　波形プロパティダイアログボックス

表示する波形形式を指定します。下記項目を指定できます。

[データアドレス]　データのメモリ開始アドレスを指定します。（16 進表示）
[データサイズ]　8 ビット / 16 ビットを指定します。
[チャンネル]　モノラル/ステレオ　を指定します。
[バッファサイズ]　データのバッファサイズを指定します。（16 進表示）

[波形プロパティ]ダイアログボックスに設定後、[OK]ボタンをクリックすると[波形]ウィンドウが開きます。
[波形]ウィンドウ表示後もポップアップメニューの[プロパティ...]を選択することで本ダイアログボックスを表示して表示内容を変更できます。

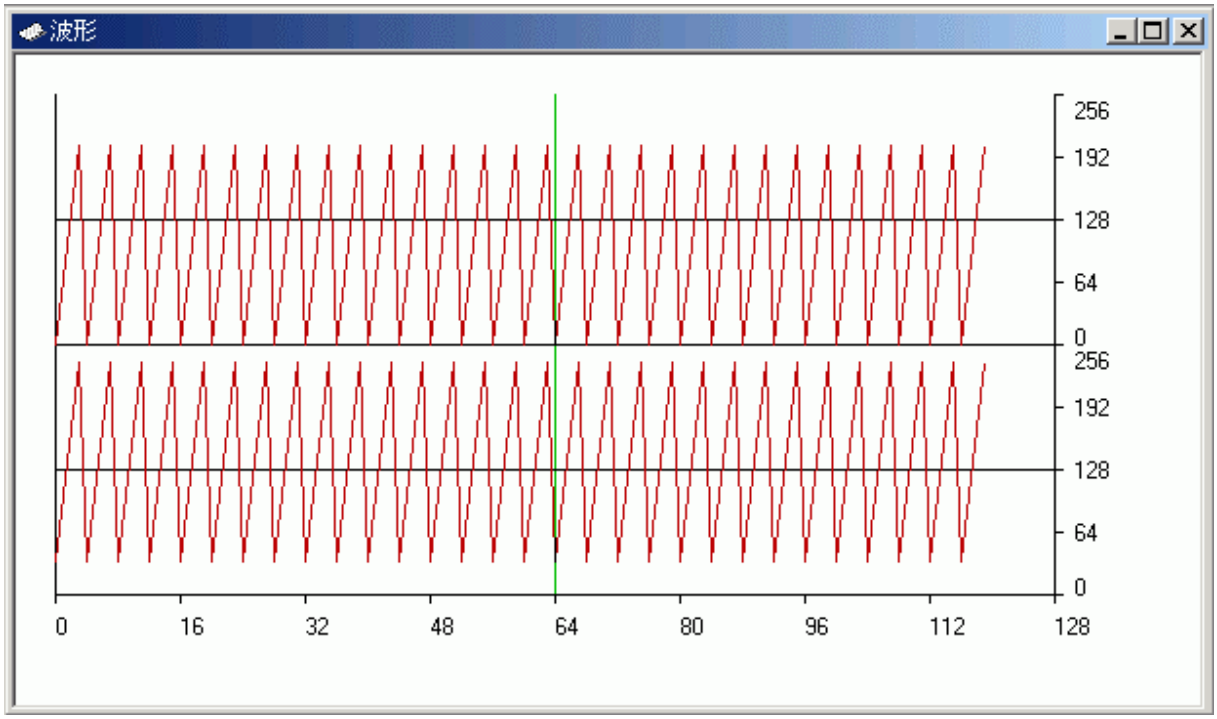


図 5-107　波形ウィンドウ

メモリ内容を波形で表示します。横軸(X)にサンプリングデータ数、縦軸(Y)にサンプリング値を表示します。

5.20.2 ウィンドウを自動更新する

ポップアップメニューの[自動更新]をチェックすると、ユーザプログラム実行停止時に自動的にウィンドウ内容を更新します。

5.20.3 ウィンドウを更新する

ポップアップメニューから[更新]を選択すると、直ちにウィンドウ内容を更新します。

5.20.4 拡大表示する

ポップアップメニューから[伸張]を選択すると、横軸を拡大して表示します。

5.20.5 縮小表示する

ポップアップメニューから[圧縮]を選択すると、横軸を縮小して表示します。

5.20.6 最初のサイズに戻す

ポップアップメニューから[元に戻す]を選択すると、最初のサイズに戻して表示します。

5.20.7 拡大/縮小倍率を設定する

ポップアップメニューの[圧縮・伸張倍率]サブメニューで拡大/縮小倍率を 2、4、8 倍から選択します。

5.20.8 横軸のサイズを設定する

ポップアップメニューの[スケール]サブメニューで横軸のサイズを 128、256、512 ピクセルから選択します。

5.20.9 カーソルを非表示にする

ポップアップメニューの[カーソル削除]をチェックすると、カーソルを非表示にします。

5.20.10 サンプリング情報を表示する

ポップアップメニューから[サンプリング情報...]を選択すると、[サンプリング情報]ダイアログボックスを表示します。

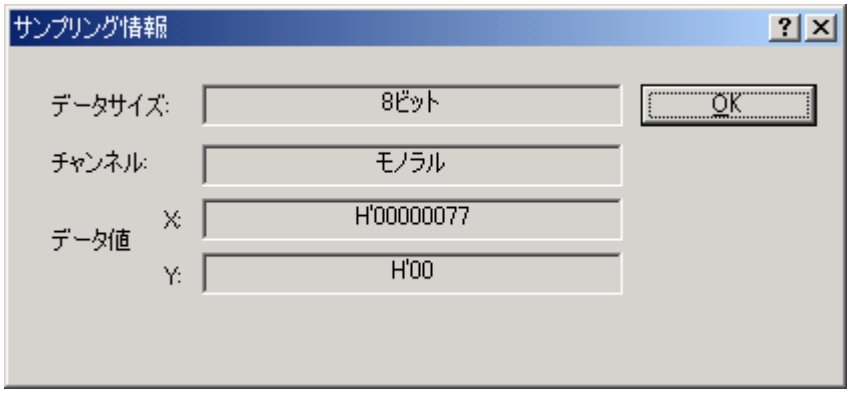


図 5-108 サンプリング情報ダイアログボックス

[波形]ウィンドウのカーソル位置のサンプリング情報を表示します。下記情報を表示します。

[データサイズ]	8 ビット / 16 ビットを表示します。	
[チャンネル]	データのチャンネルを表示します。	
[データ値]	[X]	カーソル位置の X 座標を表示します。
	[Y]	カーソル位置の Y 座標を表示します。 (ステレオ選択時は上下 2 つの Y 座標を表示します)

5.21　パフォーマンスを測定する

ユーザプログラムの実行効率を測定するにはパフォーマンス解析機能を使用します。
パフォーマンス解析機能は、直前のプログラム実行でユーザが選択したエリアの実行時間をパーセントおよびヒストグラムで表示します。

E6000 エミュレータは指定した性能分析範囲へのアクセスを全てカウントするわけではなく、ホスト PC の速度に依存し抽出した割合です。

5.21.1　パフォーマンス解析ウィンドウを開く

[パフォーマンス解析]ウィンドウを開くには、[表示->パフォーマンス->パフォーマンス解析]を選択するか、[パフォーマンス解析]ツールバーボタンをクリックして[パフォーマンス解析方式の選択]ダイアログボックスを開きます。

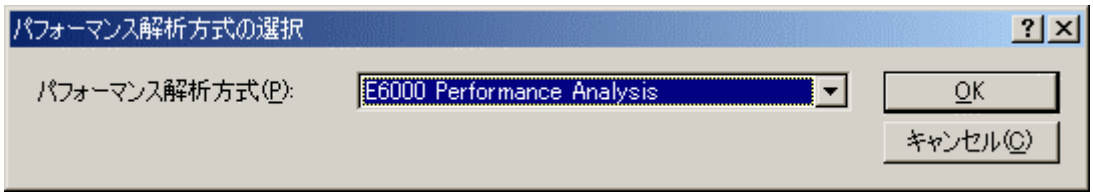


図 5-109 パフォーマンス解析方式の選択

[E6000 Performance Analysis]を選択し[OK]ボタンをクリックすると[パフォーマンス解析]ウィンドウが開きます。

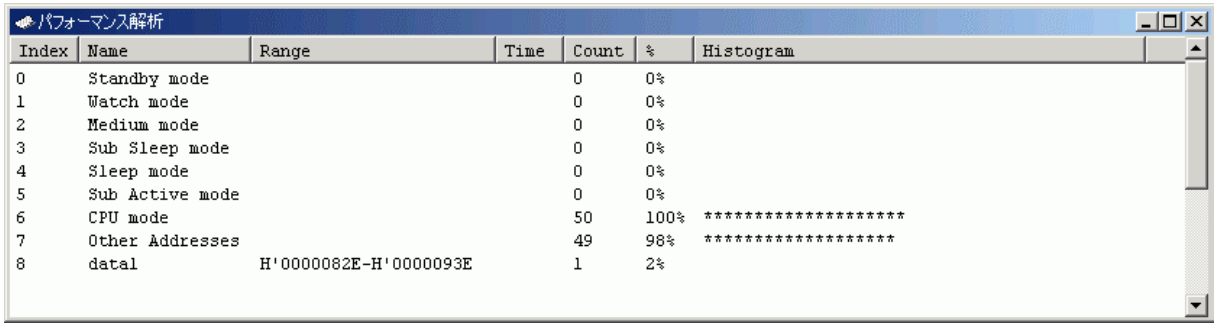


図 5-110 パフォーマンス解析ウィンドウ

直前のプログラム実行でユーザが選択したエリアの実行時間比率をパーセント、ヒストグラムおよび数値で表示します。

【注】　[パフォーマンス解析]ウィンドウには、削除することができない性能分析範囲があり、この性能分析範囲はご使用のエミュレータにより異なります。
詳細につきましては「8　本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

[パフォーマンス解析]ウィンドウ内の不要なカラムは非表示にすることができます。
カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。
カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

5.21.2 実行効率測定条件を設定する

[パフォーマンス解析]ウィンドウでは測定条件の設定内容の表示および変更ができます。
条件を設定するポイントを選択し、ポップアップメニューから[設定...]を選択すると[Add PA Range]ダイアログボックスを表示します。

[Add PA Range]ダイアログボックスにはアドレス範囲指定時、関数名を入力することにより関数のアドレス範囲を自動的に入力する入力補助機能があります。
[Add PA Range]ダイアログボックス上にある[...]ボタンをクリックすると表示される[Input Function Range]ダイアログボックスより、関数名を指定すると自動的に関数のアドレス範囲を入力することができます。

図 5-111 Input Function Range **ダイアログボックス**

【注】 1. **オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスが開きますので、関数を選択してください。[関数選択]ダイアログボックスについては「**5.14.3 **複数ラベルをサポートする」を参照してください。**

2. **算出されるアドレスは参考値です。場合により関数の終了アドレスが異なる場合があります。[逆アセンブリ]ウィンドウにより関数の最終命令を確認し、**[End Address]**の設定値を最終命令のアドレスに補正してください。**
(一般的に関数の最終命令は RTS **命令となります。)**
アドレス入力ではアドレス値以外にラベル名および式の指定も可能です。

5.21.3 実行効率測定を開始する

[コマンドライン]ウィンドウから[ANALYSIS]コマンドを実行し、実行効率測定を有効にします。
ユーザプログラムを実行すると実行効率測定を開始します。
ユーザプログラムを停止すると、測定結果を[パフォーマンス解析]ウィンドウに表示します。

【注】 [ANALYSIS]**コマンドの詳細につきましてはオンラインヘルプをご参照ください。**

5.21.4 測定条件を削除する

測定条件を選択した状態で、ポップアップメニューから[リセット]を選択すると、選択された測定条件を削除します。

5.21.5 すべての測定条件を削除する

ポップアップメニューから[全てリセット]を選択すると、設定している測定条件をすべて削除します。

6 チュートリアル

6.1 はじめに

E6000 エミュレータの主な機能を紹介するために、チュートリアルプログラムを提供しています。このプログラムを用いて説明します。

このチュートリアルプログラムは、C++言語で書かれており、10 個のランダムデータを昇順/降順にソートします。
チュートリアルプログラムでは、以下の処理を行います。
main 関数ではソート処理を繰り返し実行するため tutorial 関数の呼び出しを繰り返します。
tutorial 関数ではソートするランダムデータを生成し、sort 関数および change 関数を順に呼び出します。
sort 関数では tutorial 関数で生成したランダムデータを格納した配列を入力し、昇順にソートします。
change 関数では sort 関数で昇順にソートした配列を入力し、降順にソートします。

チュートリアルプログラムは、tutorial.cpp ファイルで提供しています。コンパイルされたロードモジュールは、Tutorial.abs ファイルとして Dwarf2 フォーマットで提供しています。

【注】 1. **再コンパイルを行った場合、本章で説明しているアドレスとずれることがあります。**
2. **本章は、一般的な E6000 エミュレータの使用例です。各製品の仕様については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。**
3. **各製品に添付される Tutorial.abs の動作アドレスは、製品によって異なります。本章で使用するアドレスについては、対応するソース行より各製品上で該当のアドレスを確認し、適宜読み替えて操作してください。**
4. **本チュートリアルでは H8S/2633 E6000 エミュレータを例に説明しています。ファイルのパス情報や、図の画面は各製品により異なりますので適宜読み替えて操作してください。**

6.2 HEW の起動

「4.2.3　既存のワークスペースを指定する場合」の手順に従ってワークスペースを開きます。

ディレクトリは以下を指定してください。
HEW インストール先ディレクトリ\Tools\Renesas\DebugComp\Platform\E6000\2633\Tutorial

【注】　パス情報は製品によって異なります。「8.2.1 チュートリアルプログラムの実行環境」を参照してください。

ファイルは以下を指定してください。

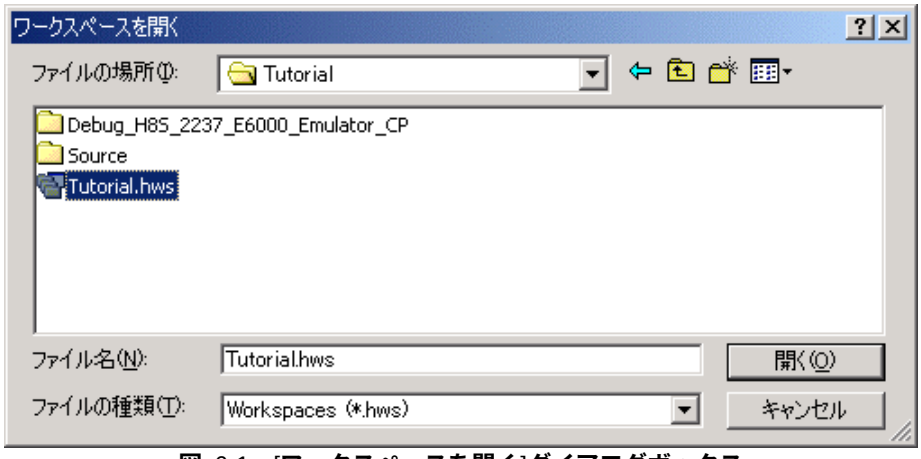


図 6-1　[ワークスペースを開く]ダイアログボックス

6.3 エミュレータの接続

6.3.1 セッションを選択する

「4.4.1 セッションを選択する」の手順に従ってセッションを選択します。選択するセッションはエミュレータ、及びエバリュエーションチップ（以降エバチップと略します）により異なります。表 6-1 に従い対応するセッションを選択してください。

表6-1 対応セッション一覧

E6000 エミュレータ	エバチップ	セッション
HS3008EPI60H	HS3308EVI60H	Sessionh8_3308_e6000_emulator
	HS3508EVI60H	Sessionh8_3508_e6000_emulator
	HS3398EVI60H	Sessionh8_3398_e6000_emulator
	HS3334EVI60H	Sessionh8_3334_e6000_emulator
	HS3257EVI60H	Sessionh8_3257_e6000_emulator
	HS3437EVI60H	Sessionh8_3437_e6000_emulator
	HS3318EVI60H	Sessionh8_3318_e6000_emulator
HS3L08EPI60H	HS3834EVI60H	Sessionh8_3834_e6000_emulator
	HS3876EVI60H	Sessionh8_3876_e6000_emulator
	HS3927EVI60H	Sessionh8_3927_e6000_emulator
HS3644EPI60H	-	Sessionh8_3644_e6000_emulator

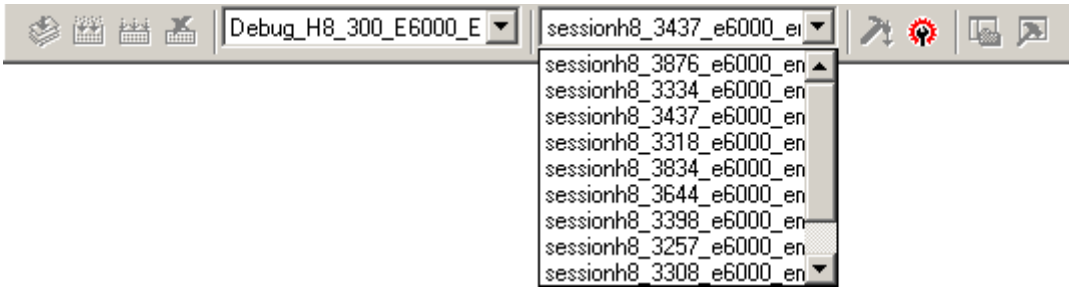


図 6-2 セッションの選択

6.3.2 エミュレータの接続

ツールバーボタンの[デバッグ]->[接続]をクリックし、エミュレータを接続します。

図 6-3 エミュレータの接続

6.4　チュートリアルプログラムのダウンロード

6.4.1　チュートリアルプログラムをダウンロードする

デバッグしたいオブジェクトプログラムをダウンロードできます。

[Download modules]の[Tutorial.abs]を右クリックすると表示されるポップアップメニューより[ダウンロード]を選択します。

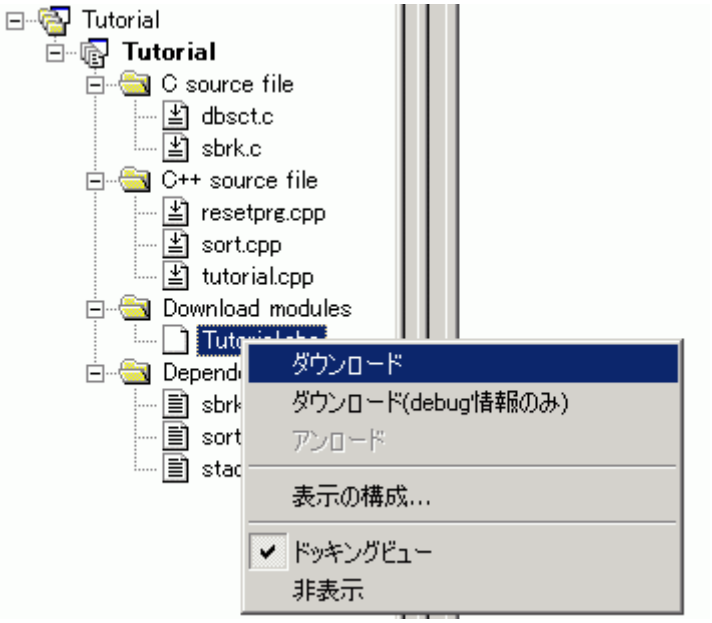


図 6-4　**チュートリアルプログラムのダウンロード**

6.4.2 ソースプログラムを表示する

HEW では、ソースレベルでプログラムをデバッグできます。

[C++ source file]の[Tutorial.cpp]をダブルクリックします。

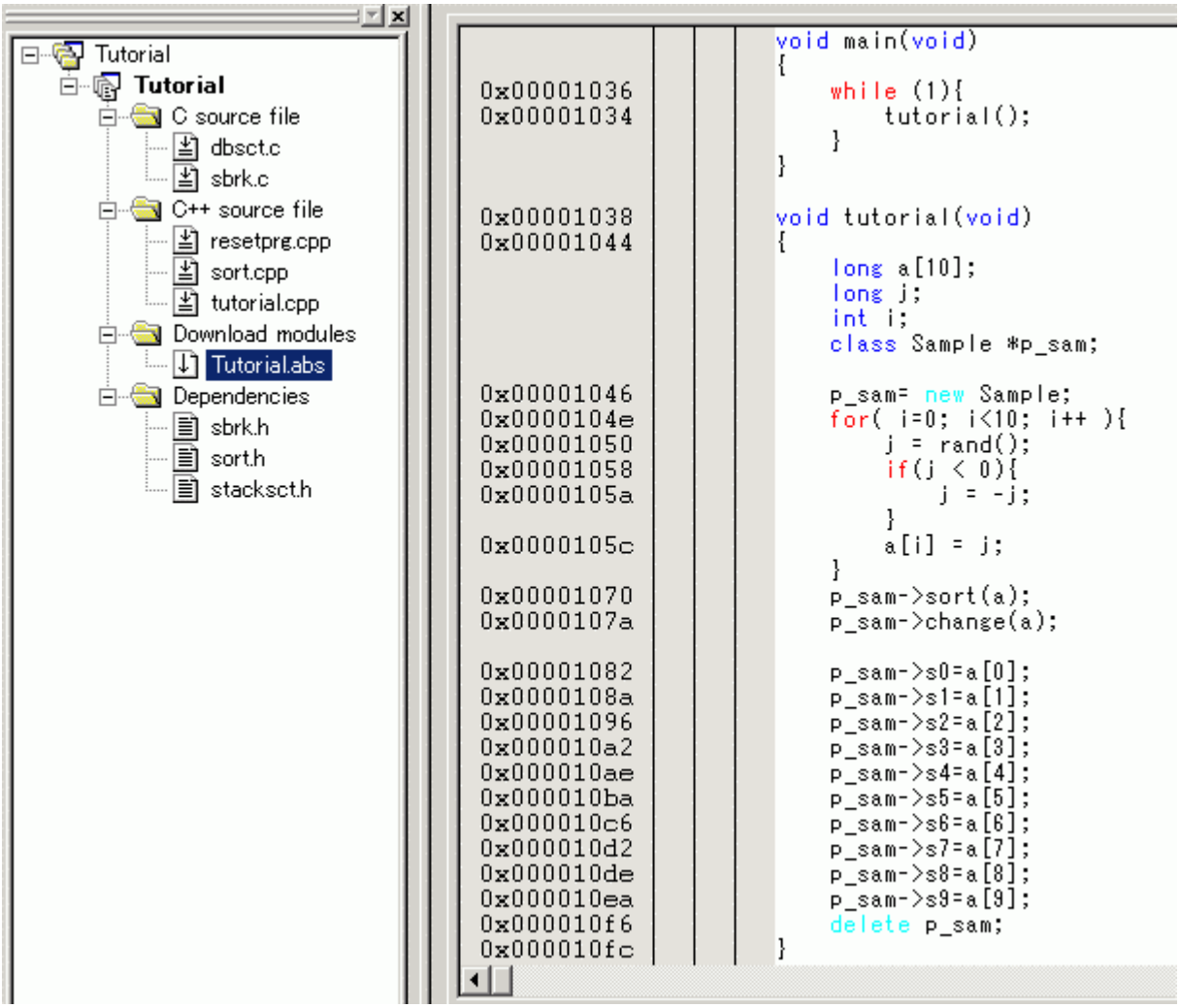


図 6-5 [ソース]ウィンドウ（ソースプログラムの表示）

必要であれば、見やすいフォントとサイズに変更することも可能です。変更方法については HEW 編 「4 エディタの使用」を参照してください。

[ソース]ウィンドウは、最初はプログラムの先頭を示しますが、スクロールバーを使って他の部分を見ることができます。

6.5　PC ブレークポイントの設定

簡単なデバッグ機能の 1 つに PC ブレークポイントがあります。

[ソース]ウィンドウにおいて、PC ブレークポイントを簡単に設定できます。たとえば、sort 関数のコール箇所に PC ブレークポイントを設定します。

sort 関数コールを含む行の[Editor]カラムをダブルクリックしてください。

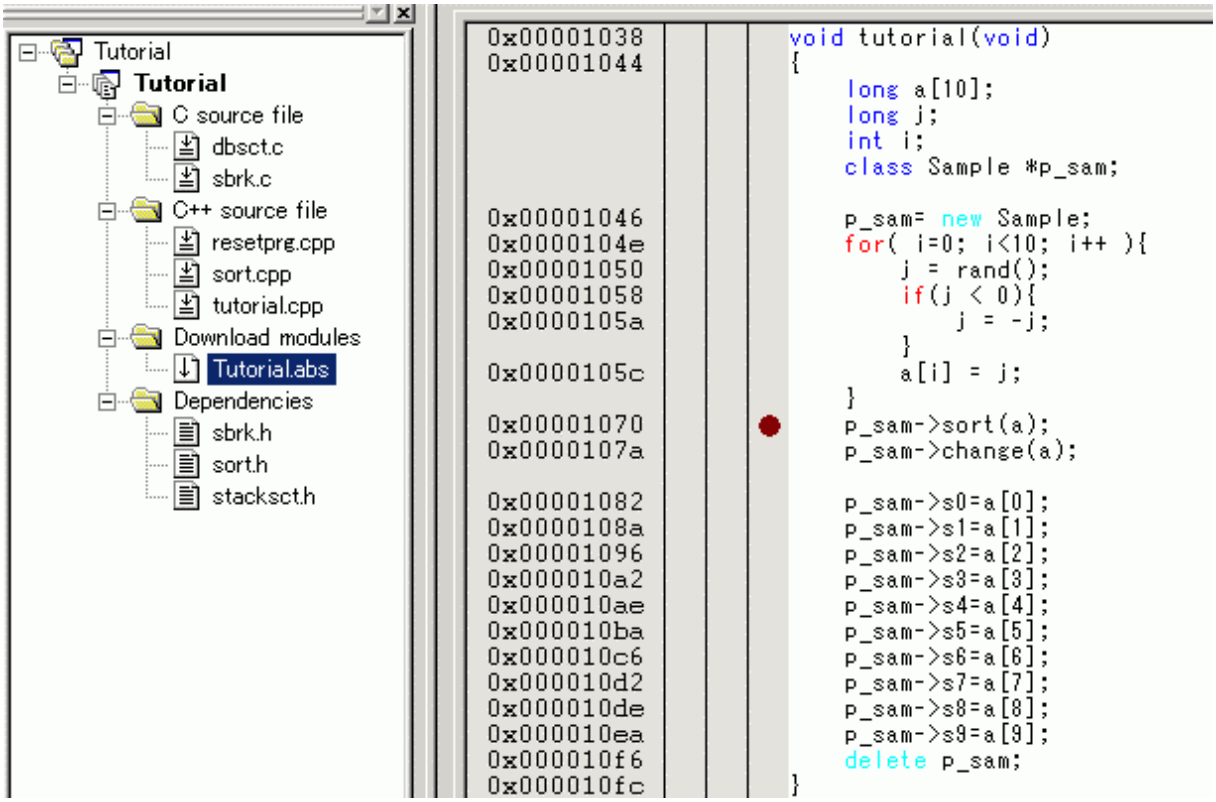


図 6-6　[ソース]ウィンドウ（PC ブレークポイントの設定）

sort 関数を含む行に”• ”と表示されます。この表示により PC ブレークポイントが設定されたことを示します。

6.6 レジスタ内容の変更

プログラムを実行する前に、プログラムカウンタの値を設定してください。

[表示]メニューから[CPU]サブメニューを選択し、[レジスタ]を選択してください。[レジスタ]ウィンドウが表示されます。

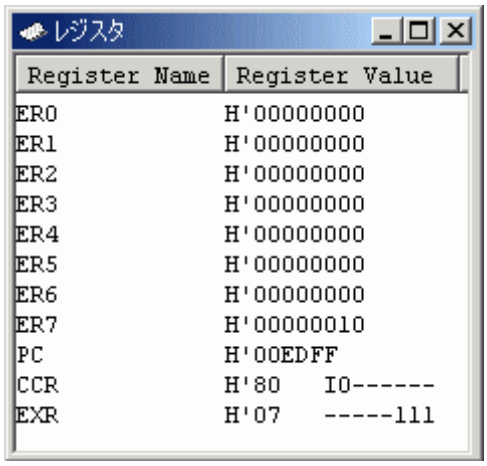


図 6-7 **[レジスタ]ウィンドウ**

プログラムカウンタ（PC）を変更する場合には、[レジスタ]ウィンドウで[PC]の数値エリアをマウスでダブルクリックすると、以下のダイアログボックスが表示され、値の変更が可能です。本チュートリアルプログラムでは、H´00000400 を設定し、[OK]ボタンをクリックしてください。

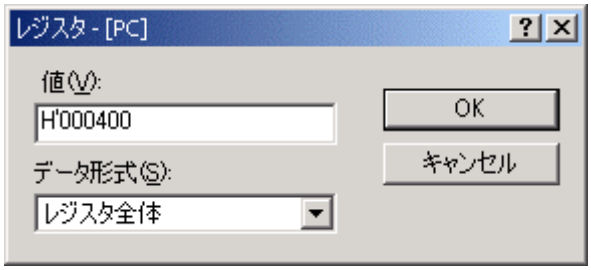


図 6-8 **[レジスタ]ダイアログボックス（PC）**

6.7　プログラムの実行

プログラムの実行方法について説明します。

プログラムを実行する場合は、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。

図 6-9　[実行]ボタン

実行を開始すると、ステータスバーに現在のアドレスバスの値、および MCU の動作状態を表示します。

プログラムはブレークポイントを設定したところまで実行されます。プログラムが停止した位置を示すために[ソース]ウィンドウの[Editor]カラムに矢印が表示されます。また、[Break = PC Break]メッセージがステータスバーに表示されます。

【注】　ブレーク後にソースファイルを表示する際に、ソースファイルパスを問い合わせる場合があります。ソースファイルの場所は以下です。
HEW **インストール先ディレクトリ**\Tools\Renesas\DebugComp\Platform\E6000\2633\Tutorial \source
パス情報は各製品により異なりますので「\2633」の部分を適宜変更してください。

```
0x00001038          void tutorial(void)
0x00001044          {
                        long a[10];
                        long j;
                        int i;
                        class Sample *p_sam;

0x00001046              p_sam= new Sample;
0x0000104e              for( i=0; i<10; i++ ){
0x00001050                  j = rand();
0x00001058                  if(j < 0){
0x0000105a                      j = -j;
                            }
0x0000105c                  a[i] = j;
                        }
0x00001070              p_sam->sort(a);
0x0000107a              p_sam->change(a);

0x00001082              p_sam->s0=a[0];
0x0000108a              p_sam->s1=a[1];
0x00001096              p_sam->s2=a[2];
0x000010a2              p_sam->s3=a[3];
0x000010ae              p_sam->s4=a[4];
0x000010ba              p_sam->s5=a[5];
0x000010c6              p_sam->s6=a[6];
0x000010d2              p_sam->s7=a[7];
0x000010de              p_sam->s8=a[8];
0x000010ea              p_sam->s9=a[9];
0x000010f6              delete p_sam;
0x000010fc          }
```

図 6-10　[ソース]ウィンドウ（ブレーク状態）

[ステイタス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示]メニューから[CPU]サブメニューを選択し、[ステイタス]を選択してください。
[ステイタス]ウィンドウが表示されますので、[Platform]シートを開いて Cause of last break の Status を確認してください。

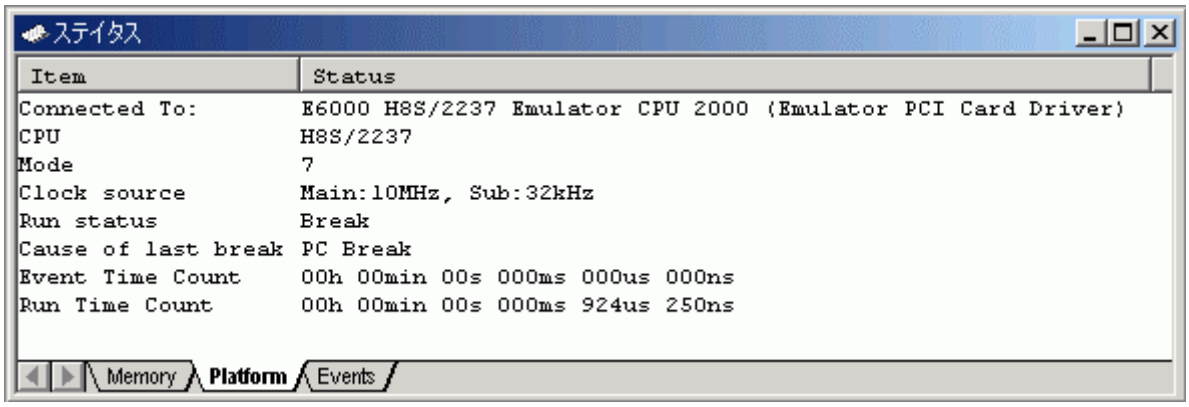


図 6-11 [ステイタス]ウィンドウ

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

6.8 ブレークポイントの確認

設定した全てのブレークポイントは、[イベントポイント]ウィンドウで確認することができます。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。[Breakpoint]シートを開きます。

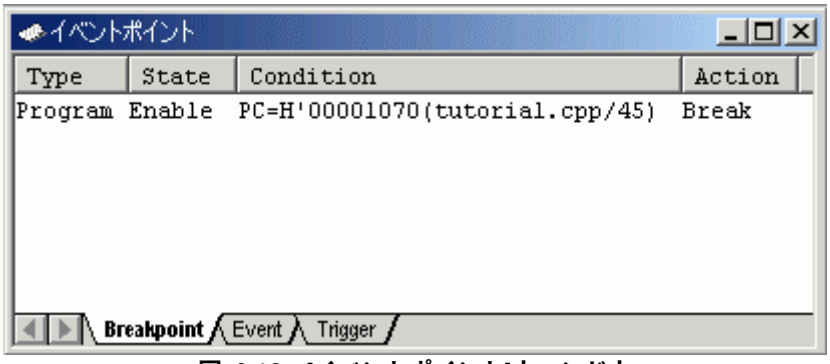


図 6-12 [イベントポイント]ウィンドウ

マウスの右ボタンで[イベントポイント]ウィンドウをクリックすると開くポップアップメニューにより、ブレークポイントの設定／変更、新しいブレークポイントの定義、およびブレークポイントの削除、有効／無効の選択ができます。

6.9 シンボルの参照

[ラベル]ウィンドウを使ってモジュール内のシンボル情報を表示させることができます。

[表示]メニューから[シンボル]サブメニューを選択し、[ラベル]を選択してください。[ラベル]ウィンドウが表示され、モジュール内のシンボル情報が参照できます。

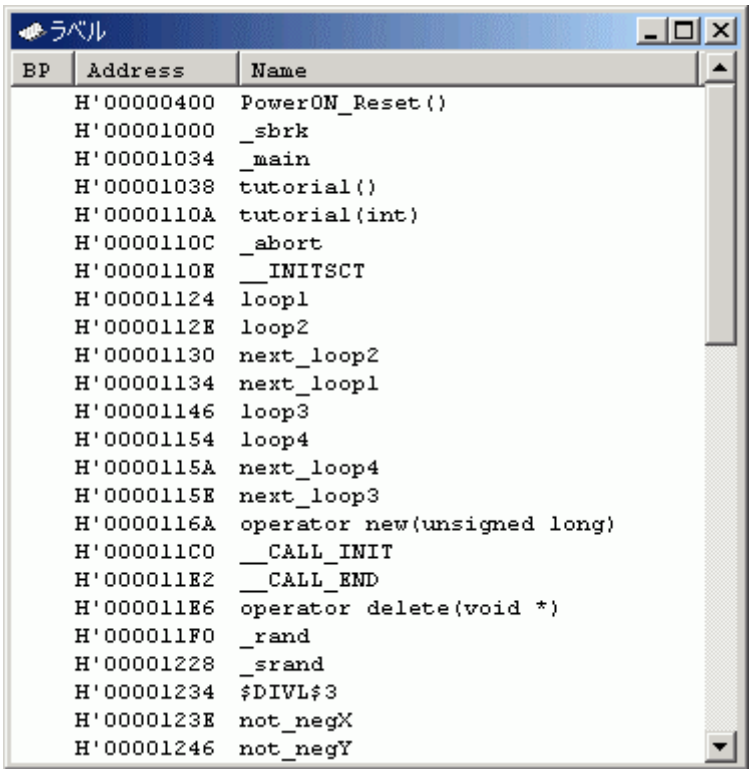


図 6-13 [ラベル]ウィンドウ

6.10 メモリ内容の確認

ラベル名を指定することによって、ラベルが登録されているメモリの内容を[メモリ]ウィンドウで確認することができます。たとえば、以下のように、バイトサイズで_main に対応するメモリ内容を確認します。

[表示]メニューから[CPU]サブメニューを選択し、[メモリ...]を選択し、[先頭アドレス]エディットボックスに"_main"を入力、[終了アドレス]エディットボックスに"+ff"を入力し、[データサイズ]コンボボックスを"Byte"に設定してください。

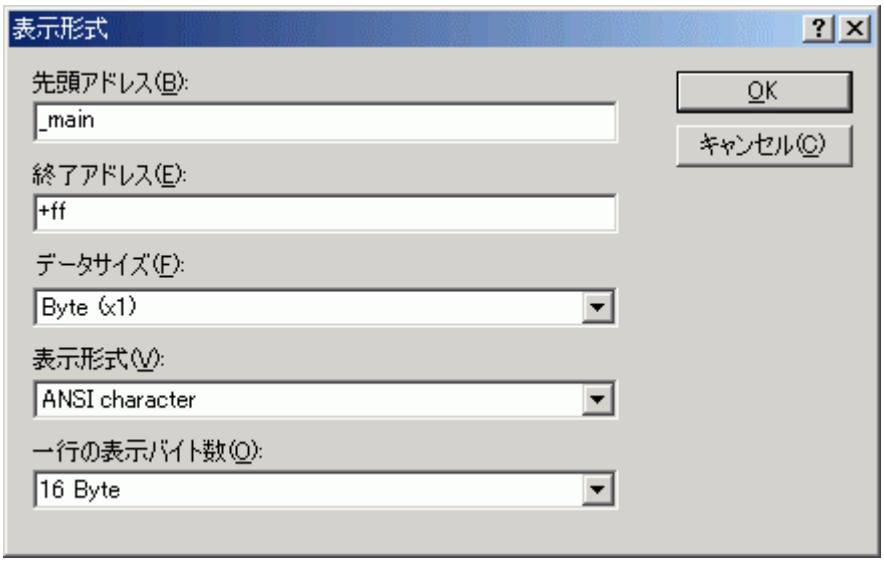
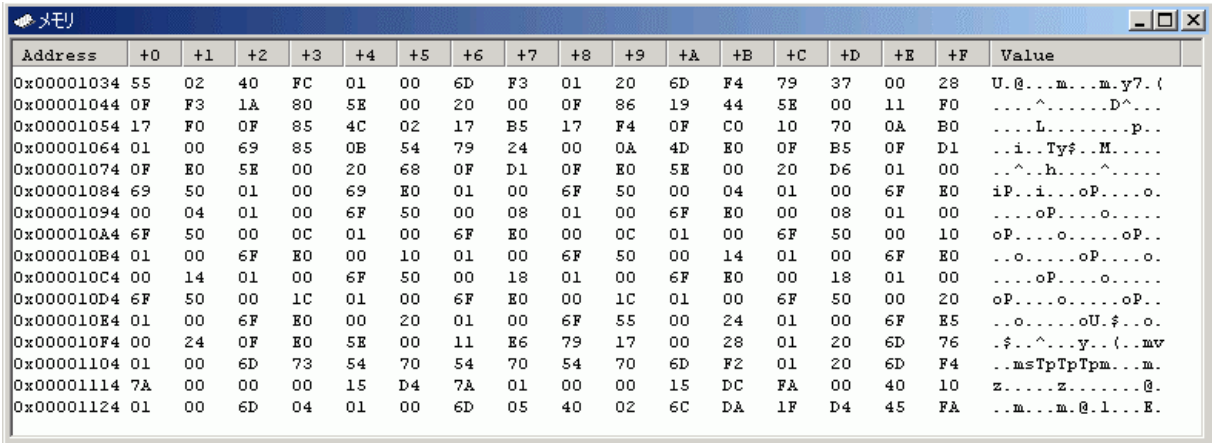


図 6-14 [表示形式]ダイアログボックス

[OK]ボタンをクリックしてください。指定されたメモリ領域を示す[メモリ]ウィンドウが表示されます。


メモリ

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Value
0x00001034	55	02	40	FC	01	00	6D	F3	01	20	6D	F4	79	37	00	28	U.@...m...m.y7.(
0x00001044	0F	F3	1A	80	5E	00	20	00	0F	86	19	44	5E	00	11	F0	....^.....D^...
0x00001054	17	F0	0F	85	4C	02	17	B5	17	F4	0F	C0	10	70	0A	B0	....L.....p..
0x00001064	01	00	69	85	0B	54	79	24	00	0A	4D	E0	0F	B5	0F	D1	..i..Ty$..M.....
0x00001074	0F	E0	5E	00	20	68	0F	D1	0F	E0	5E	00	20	D6	01	00	..^..h....^.....
0x00001084	69	50	01	00	69	E0	01	00	6F	50	00	04	01	00	6F	E0	iP..i...oP....o.
0x00001094	00	04	01	00	6F	50	00	08	01	00	6F	E0	00	08	01	00	....oP....o.....
0x000010A4	6F	50	00	0C	01	00	6F	E0	00	0C	01	00	6F	50	00	10	oP....o.....oP..
0x000010B4	01	00	6F	E0	00	10	01	00	6F	50	00	14	01	00	6F	E0	..o.....oP....o.
0x000010C4	00	14	01	00	6F	50	00	18	01	00	6F	E0	00	18	01	00	....oP....o.....
0x000010D4	6F	50	00	1C	01	00	6F	E0	00	1C	01	00	6F	50	00	20	oP....o.....oP..
0x000010E4	01	00	6F	E0	00	20	01	00	6F	55	00	24	01	00	6F	E5	..o.....oU.$..o.
0x000010F4	00	24	0F	E0	5E	00	11	E6	79	17	00	28	01	20	6D	76	.$..^...y..(..mv
0x00001104	01	00	6D	73	54	70	54	70	54	70	6D	F2	01	20	6D	F4	..msTpTpTpm...m.
0x00001114	7A	00	00	00	15	D4	7A	01	00	00	15	DC	FA	00	40	10	z.....z.......@.
0x00001124	01	00	6D	04	01	00	6D	05	40	02	6C	DA	1F	D4	45	FA	..m...m.@.l...E.

図 6-15 [メモリ]ウィンドウ

6.11　変数の参照

プログラムをステップ処理するとき、プログラムで使われる変数の値が変化することを確認できます。たとえば、以下の手順で、プログラムの初めに宣言した long 型の配列 a を見ることができます。

[ソース]ウィンドウに表示されている配列 a の左側をクリックし、カーソルを置いてください。
マウスの右ボタンで[インスタントウォッチ]を選択してください。

以下のダイアログボックスが表示されます。

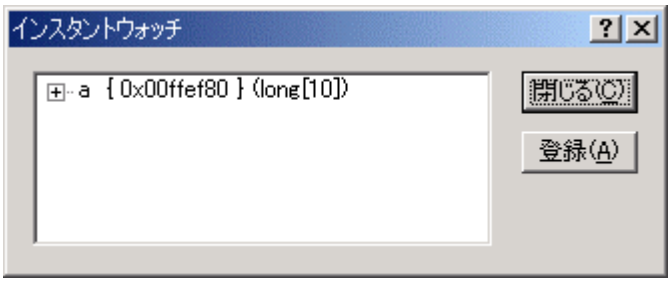


図 6-16　[インスタントウォッチ]ダイアログボックス

[登録]ボタンをクリックして、[ウォッチ]ウィンドウに変数を加えてください。

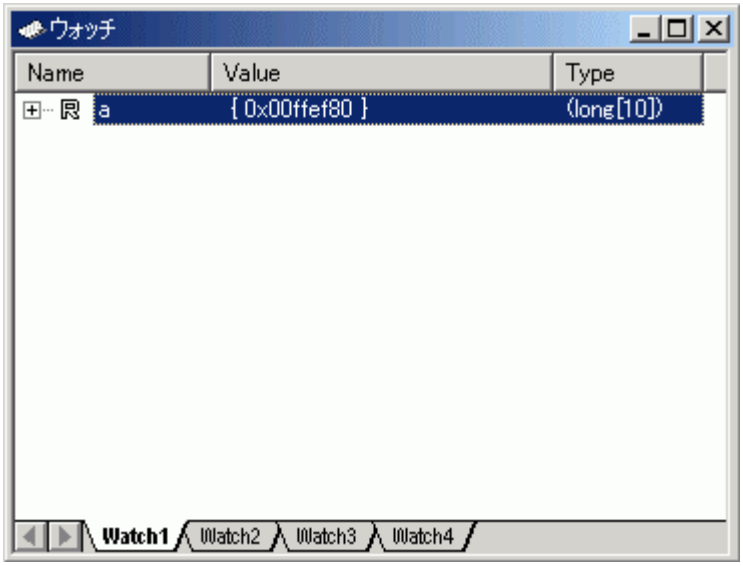


図 6-17　[ウォッチ]ウィンドウ（配列の表示）

また、変数名を指定して、[ウォッチ]ウィンドウに変数を加えることもできます。

マウスの右ボタンで[ウォッチ]ウィンドウをクリックし、ポップアップメニューから[シンボル登録]を選択してください。

以下のダイアログボックスが表示されます。

図 6-18 [シンボル登録]ダイアログボックス

[変数または式]エディットボックスに変数 i を入力し、[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、int 型の変数 i が表示されます。

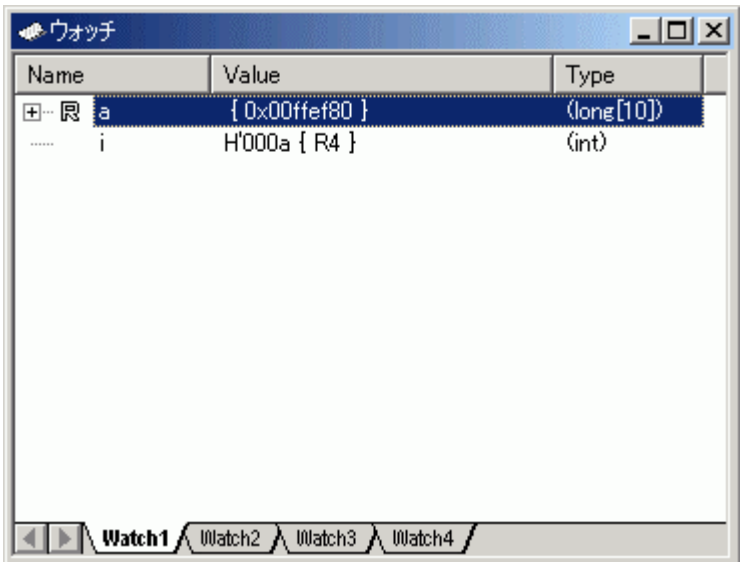


図 6-19 [ウォッチ]ウィンドウ（変数の表示）

[ウォッチ]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の各要素を参照することができます。

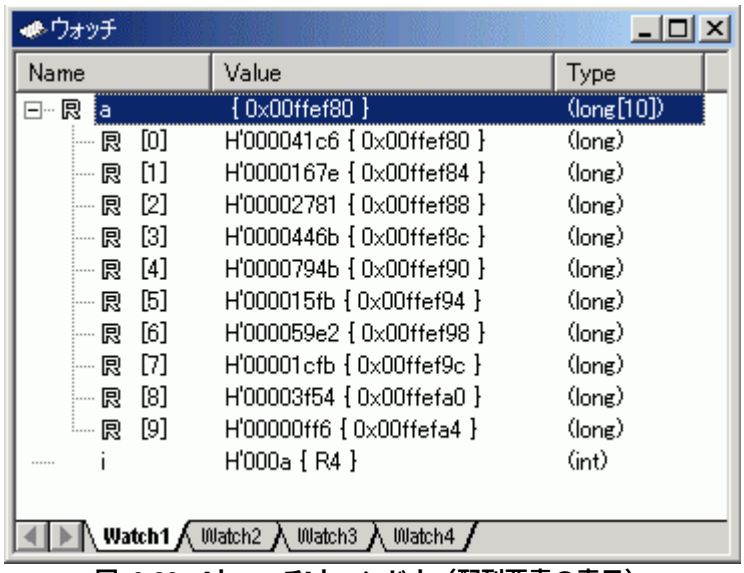


図 6-20　**[ウォッチ]ウィンドウ（配列要素の表示）**

6.12　ローカル変数の表示

[ローカル]ウィンドウを使って関数内のローカル変数を表示させることができます。例として、tutorial 関数のローカル変数を調べます。

この関数は、4 つのローカル変数 a, j, i, p_sam を宣言します。

[表示]メニューから[シンボル]サブメニューを選択し、[ローカル]を選択してください。[ローカル]ウィンドウが表示されます。

[ローカル]ウィンドウには、現在のプログラムカウンタ（PC）が指している関数のローカル変数とその値が表示されます。

関数内にローカル変数が存在しない場合、[ローカル]ウィンドウに何も表示されません。

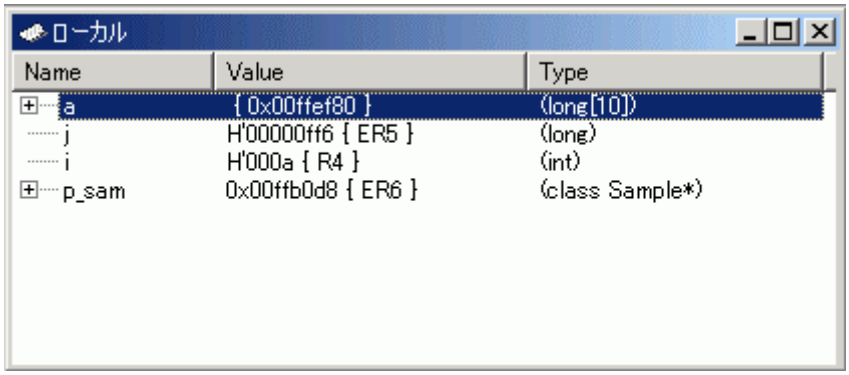


図 6-21　**[ローカル]ウィンドウ**

[ローカル]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の構成要素を表示させてください。

sort 関数実行前と実行後の配列 a の要素を参照し、ランダムデータが降順にソートされていることを確認してください。

6.13　プログラムのステップ実行

HEW は、プログラムのデバッグに有効な各種のステップコマンドを備えています。

表6-2　ステップオプション

項番	コマンド	説明
1	ステップイン	各ステートメントを実行します（関数内のステートメントを含む）。
2	ステップオーバ	関数コールを 1 ステップとして、ステップ実行します。
3	ステップアウト	関数を抜け出し、関数を呼び出したプログラムの次のステートメントで停止します。
4	ステップ...	指定した速度で指定回数分ステップ実行します。

6.13.1　ステップインコマンドの実行

[ステップイン]コマンドはコール関数の中に入り、コール関数の先頭のステートメントで停止します。

sort 関数の中に入るために、[デバッグ]メニューから[ステップイン]を選択するか、またはツールバーの[ステップイン]ボタンをクリックしてください。

図 6-22　[ステップイン]ボタン

```
0x00002000          Sample::Sample()
0x00002006          {
0x0000201c              s0=0;
0x00002026              s1=0;
0x0000202c              s2=0;
0x00002032              s3=0;
0x00002038              s4=0;
0x0000203e              s5=0;
0x00002044              s6=0;
0x0000204a              s7=0;
0x00002050              s8=0;
0x00002056              s9=0;
0x00002060          }

0x00002068          void Sample::sort(long *a)
0x00002070          {
                        long t;
                        int i, j, k, gap;

0x00002072              gap = 5;
0x00002076              while( gap > 0 ){
0x00002078                  for( k=0; k<gap; k++){
0x0000207c                      for( i=k+gap; i<10; i=i+gap ){
0x00002080                          for(j=i-gap; j>=k; j=j-gap){
0x00002084                              if(a[j]>a[j+gap]){
                                            t = a[j];
0x000020a4                                  a[j] = a[j+gap];
0x000020a8                                  a[j+gap] = t;
                                        }
                                        else
                                            break;
                                    }
                                }
                            }
0x000020c0                  gap = gap/2;
                        }
0x000020cc          }
```

図 6-23　[ソース]ウィンドウ（ステップイン）

[ソース]ウィンドウの強調表示が、sort 関数の先頭のステートメントに移動します。

6.13.2　ステップアウトコマンドの実行

[ステップアウト]コマンドはコール関数の中から抜け出し、コール元プログラムの次のステートメントで停止します。

sort 関数の中から抜け出すために、[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。

図 6-24　[ステップアウト]ボタン

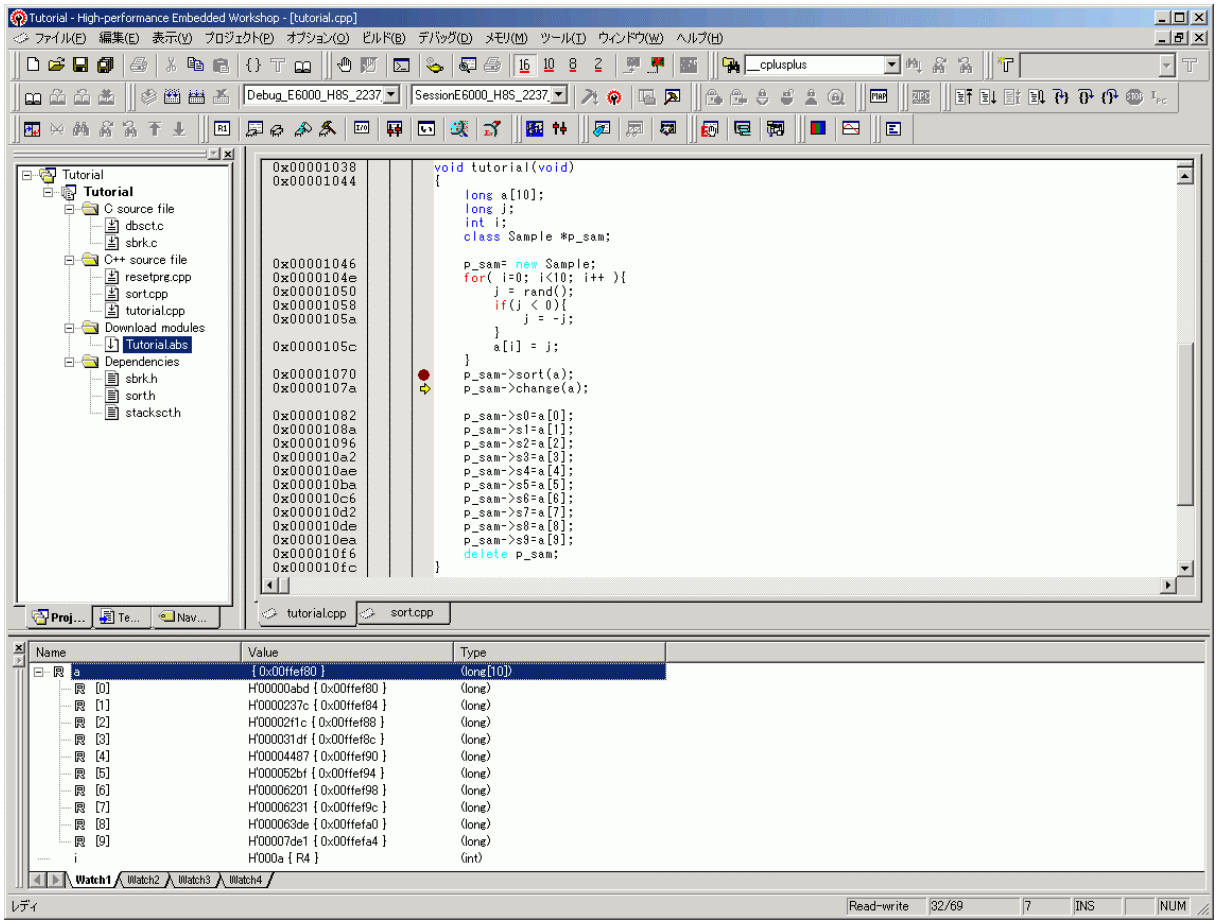


図 6-25　[HEW]ウィンドウ（ステップアウト）

[ウォッチ]ウィンドウに表示された変数 a のデータが昇順にソートされます。

6.13.3 ステップオーバコマンドの実行

[ステップオーバ]コマンドは関数コールを 1 ステップとして実行して、メインプログラムの次のステートメントで停止します。

change 関数中のステートメントを一度にステップ実行するために、[デバッグ]メニューから[ステップオーバ]を選択するか、またはツールバーの[ステップオーバ]ボタンをクリックしてください。

図 6-26 [ステップオーバ]ボタン

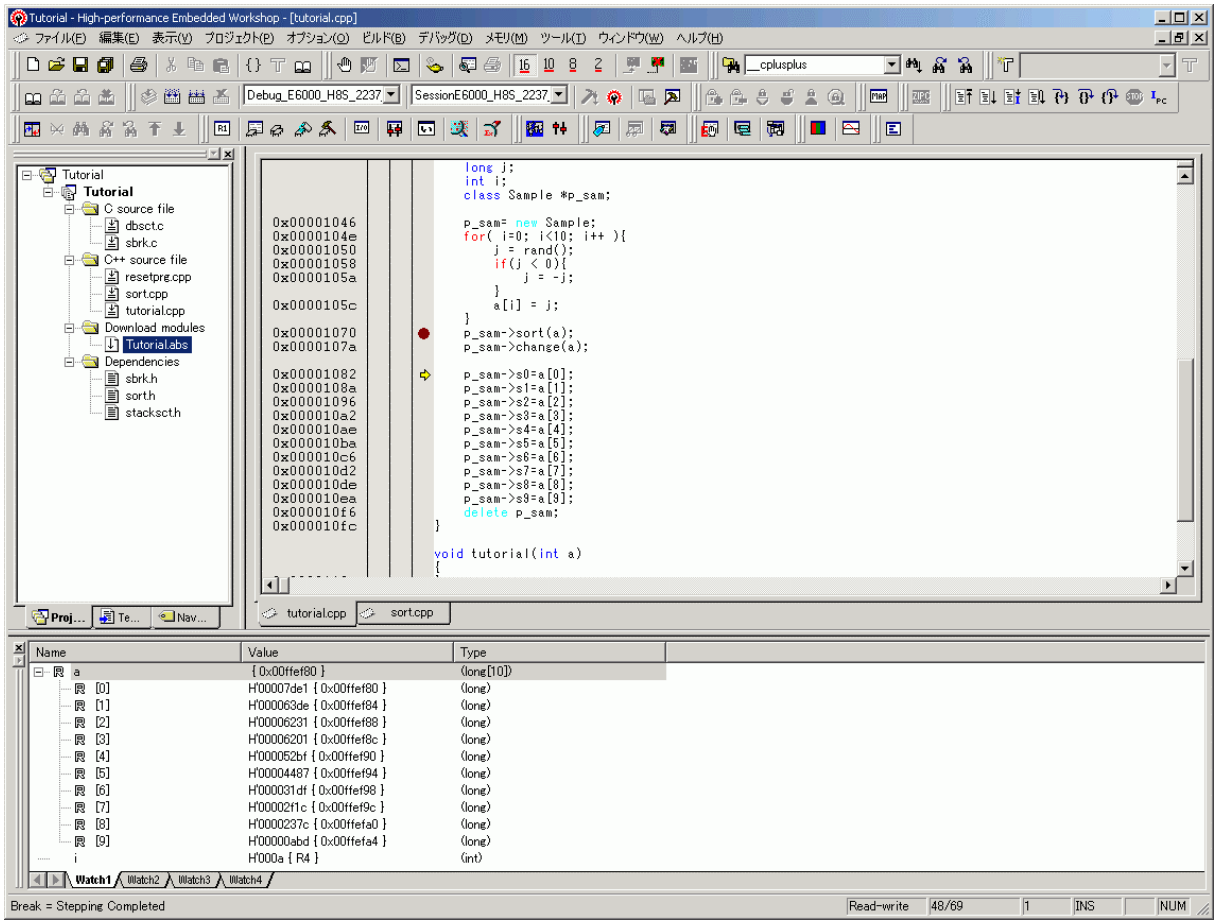


図 6-27 [HEW]ウィンドウ（ステップオーバ）

6.14　プログラムの強制ブレーク

HEW は、プログラムを強制的にブレークすることができます。

ブレークポイントを全て解除してください。

tutorial 関数の残り部分を実行するために、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。

図 6-28　[実行]ボタン

プログラムは無限ループ処理を実行していますので、強制ブレークするために、[デバッグ]メニューから[プログラムの停止]を選択するか、ツールバー上の[停止]ボタンを実行してください。

図 6-29　[停止]ボタン

6.15　MCU のリセット

MCU をリセットすることにより、内蔵 I/O レジスタの初期化、およびリセットベクタに設定されたアドレスを PC に設定します。

MCU のリセットを行うには、[デバッグ]メニューから[CPU のリセット]を選択するか、ツールバー上の[CPU リセット]ボタンを選択してください。

図 6-30　CPU リセットボタン

また、リセット直後からプログラムを実行する場合には、[デバッグ]メニューから[リセット後実行]を選択するか、ツールバー上の[リセット後実行]ボタンを選択してください。

図 6-31　リセット後実行ボタン

【注】　本チュートリアルプログラムは、リセットベクタからプログラムを実行しても問題なく実行できるように作成しています。

6.16 ブレーク機能

E6000 エミュレータは、PC ブレーク機能とイベントポイントによるブレーク機能を持っています。
PC ブレークポイントの設定、およびイベントポイントの設定は[イベントポイント]ウィンドウでそれぞれ行うことができます。

以下にブレーク機能の概要と設定方法について説明します。

6.16.1 PC ブレーク機能

E6000 エミュレータは、最大 256 ポイントまで PC ブレークポイントを設定することができます。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。
[Breakpoint]シートを開きます。

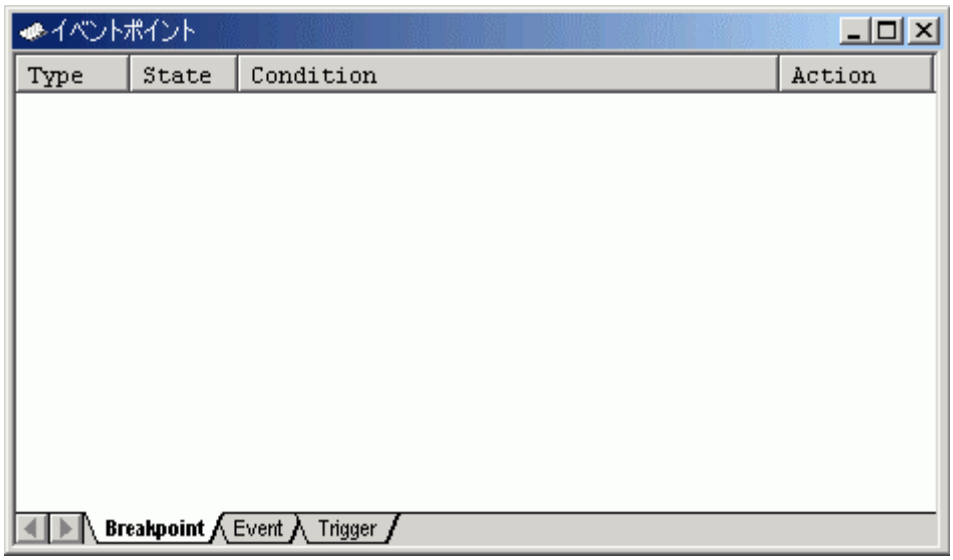


図 6-32 [イベントポイント]ウィンドウ（PC ブレーク設定前）

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加…]を選択してください。

[Breakpoint/Event Properties]ダイアログボックスが表示されます。

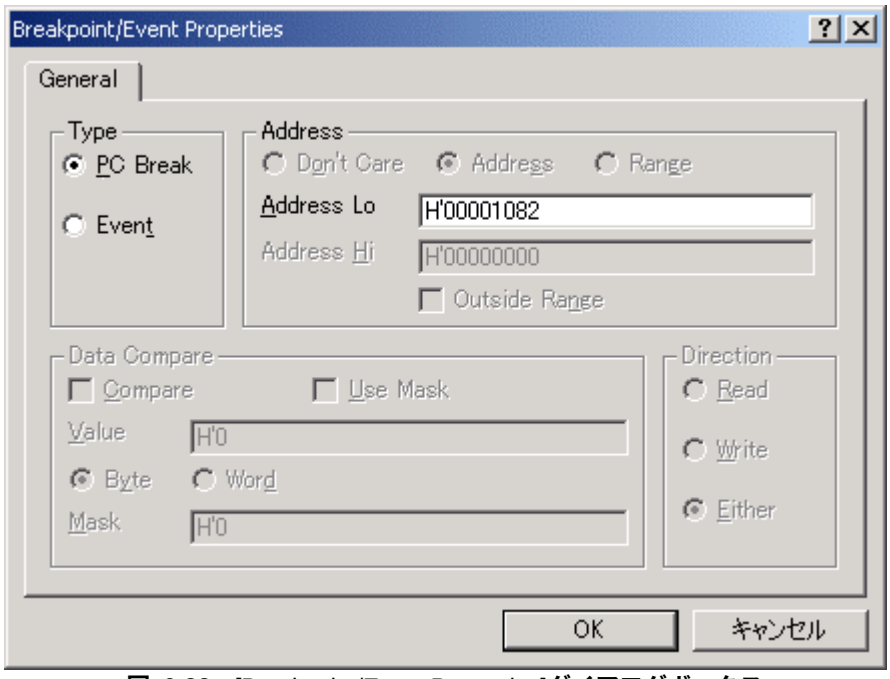


図 6-33　[Breakpoint/Event Properties]**ダイアログボックス**

[Type]グループボックスの[PC Break]ラジオボタンをチェックします。

[Address]グループボックスの[Address Lo]エディットボックスにtutorial関数内の「p_sam->s0=a[0];」と記述されている行のアドレスを[ソース]ウィンドウで参照し入力してください。本例ではH’00001082を入力します。

【注】　本ダイアログボックスは、製品ごとに異なります。各製品の内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

[OK]ボタンをクリックしてください。

[イベントポイント]ウィンドウには、設定された PC ブレークポイントが表示されます。

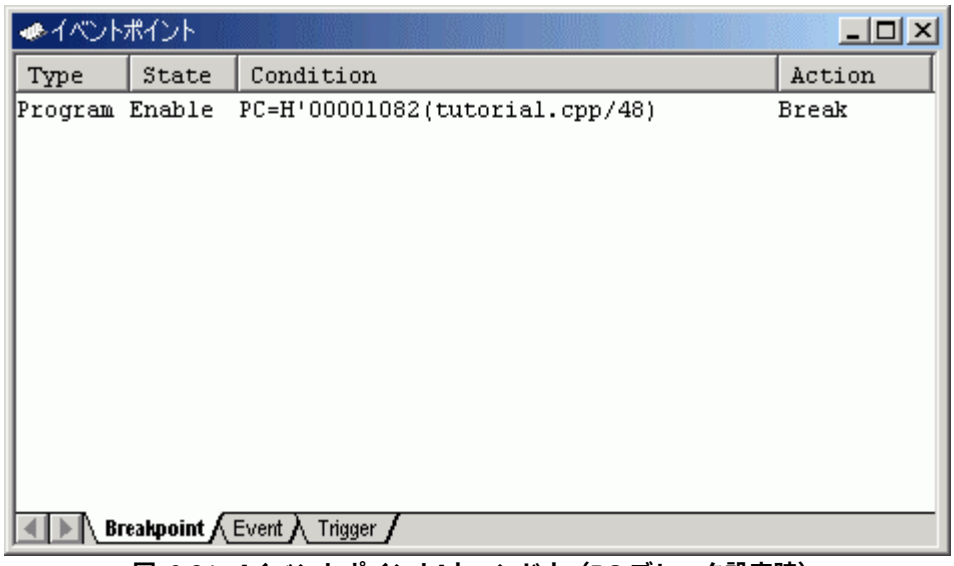


図 6-34　**[イベントポイント]ウィンドウ（PC ブレーク設定時）**

【注】　本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

[イベントポイント]ウィンドウを閉じてください。
チュートリアルプログラムをブレークポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。

設定したブレークポイントまで、プログラムを実行して停止します。

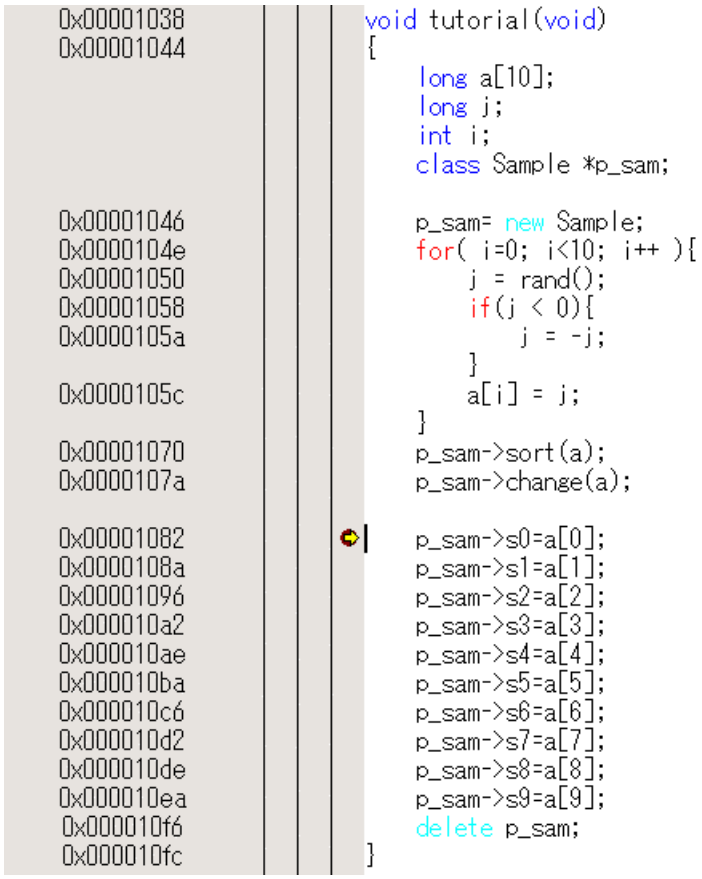


図 6-35　実行停止時の[ソース]ウィンドウ（PC ブレーク）

[ステイタス]ウィンドウの表示内容は、以下のようになります。

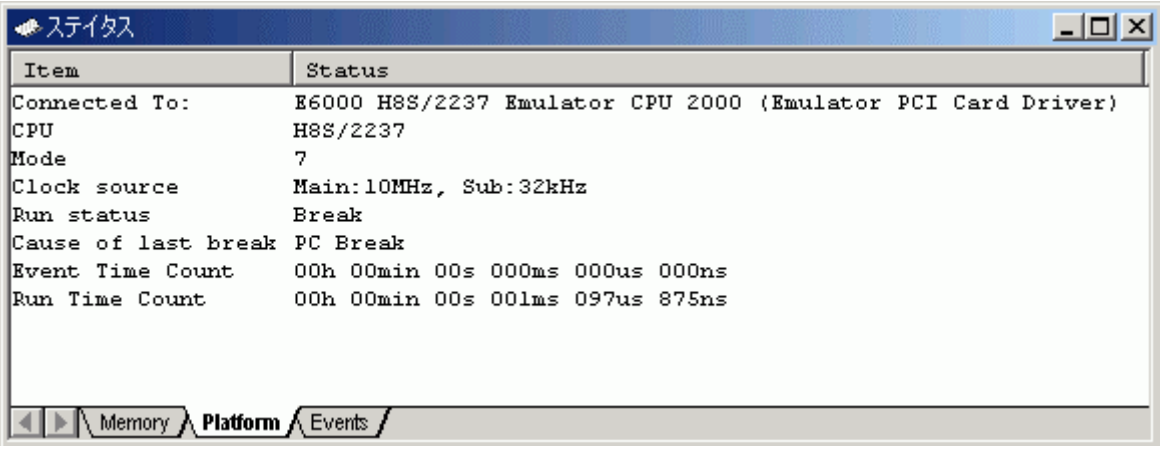


図 6-36　[ステイタス]ウィンドウの表示内容（PC ブレーク）

【注】　本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

6.16.2　イベントポイントによるブレーク機能

イベントポイントのイベントチャネル１（Ch1）を用いて、指定したイベントが５回成立した場合にブレークする方法を説明します。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。

先ほど設定したブレークポイントを削除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているブレークポイントをすべて解除してください。

次は Event を設定します。[Event]タブをクリックしてください。

イベントポイントは、イベントチャネル８箇所、範囲チャネル４箇所の最大１２箇所まで独立に条件を設定することができます。ここでは、イベントチャネル１（Ch1）を設定します。

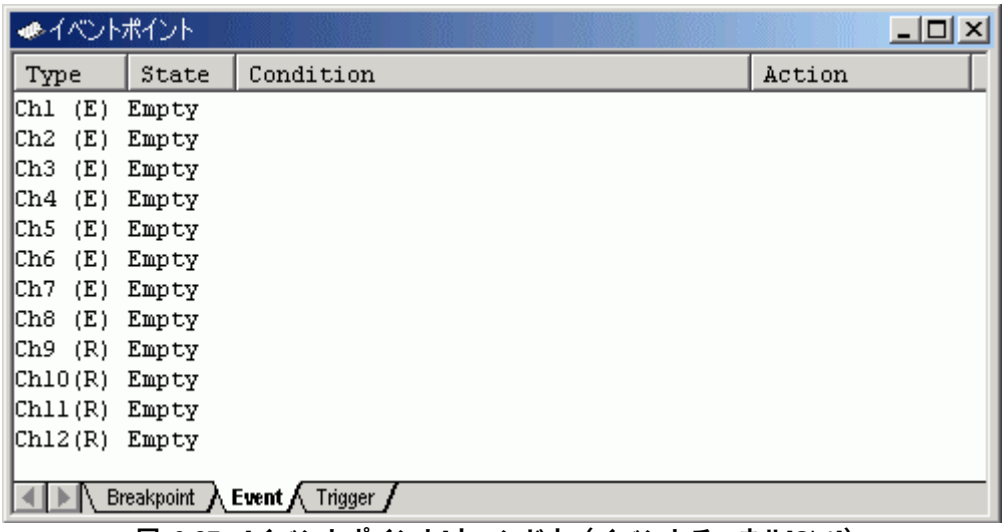


図 6-37　[イベントポイント]ウィンドウ（イベントチャネル[Ch1]）

[イベントポイント]ウィンドウ内の Ch1 行を選択してください。Ch1 行が強調表示されますので、ダブルクリックしてください。

[Breakpoint/Event Properties]ダイアログボックスが表示されます。

[General]ページの各グループボックスを以下のように設定します。

- [Type]グループボックスの[Event]ラジオボタンを選択します。
- [Address]グループボックスの[Address]ラジオボタンを選択し、[Address Lo]エディットボックスに tutorial 関数内の「a[i]=j;」と記述されている行のアドレスを[ソース]ウィンドウで参照し入力してください。本例では H'0000105c を入力します。

[Action]ページの[Required number of event occurrences]エディットボックスにイベント成立回数として D'5 を入力してください。

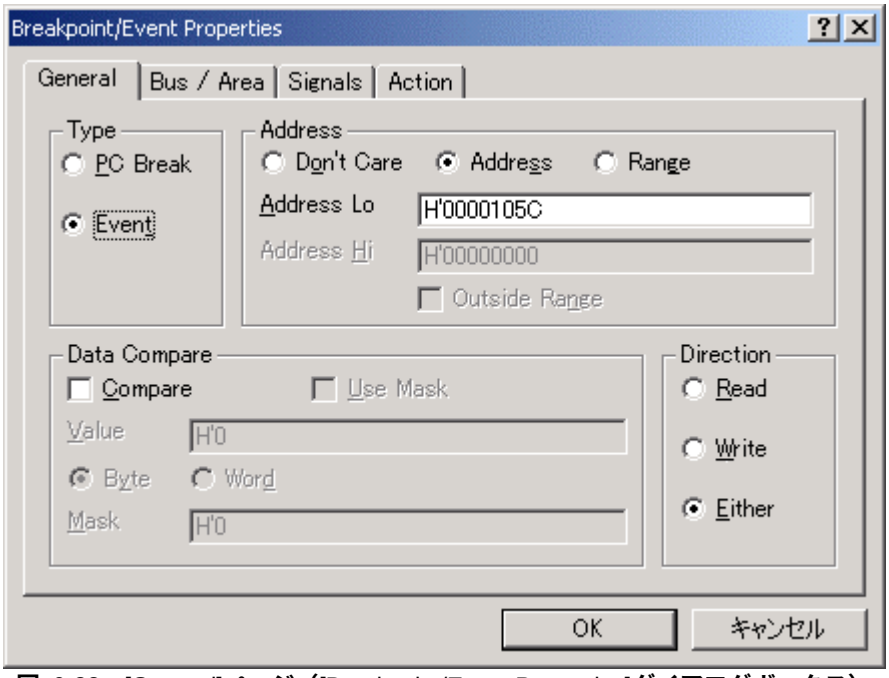


図 6-38 [General]ページ（[Breakpoint/Event Properties]ダイアログボックス）

[OK]ボタンをクリックしてください。
[イベントポイント]ウィンドウに以下が表示されます。

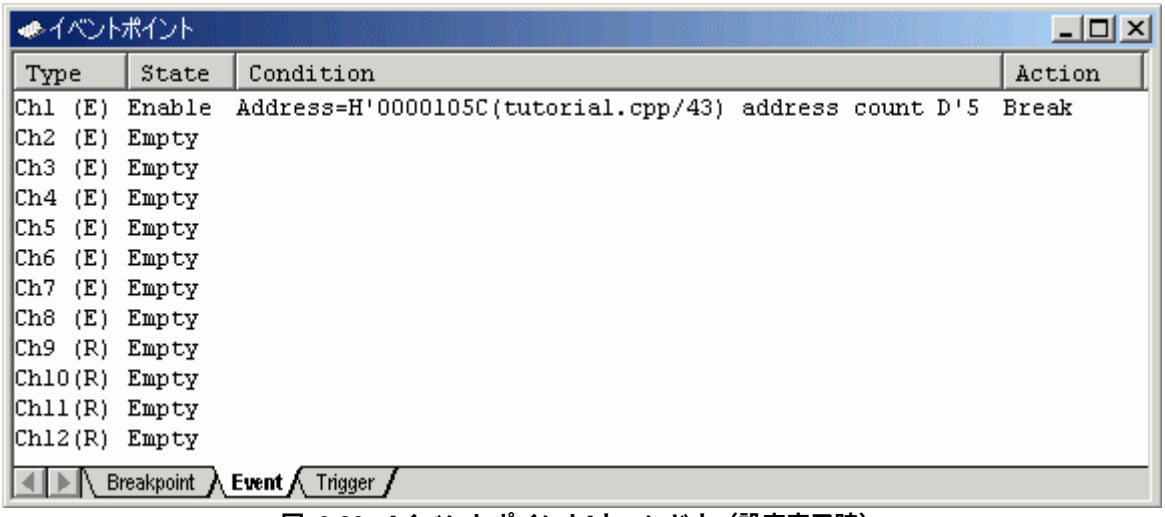


図 6-39 [イベントポイント]ウィンドウ（設定完了時）

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

チュートリアルプログラムをブレークポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。

Ch1 の条件まで、プログラムを実行して停止します。

```
0x00001038          void tutorial(void)
0x00001044          {
                        long a[10];
                        long j;
                        int i;
                        class Sample *p_sam;

0x00001046              p_sam= new Sample;
0x0000104e              for( i=0; i<10; i++ ){
0x00001050                  j = rand();
0x00001058                  if(j < 0){
0x0000105a                      j = -j;
                            }
0x0000105c                  a[i] = j;
                        }
0x00001070              p_sam->sort(a);
0x0000107a              p_sam->change(a);

0x00001082              p_sam->s0=a[0];
0x0000108a              p_sam->s1=a[1];
0x00001096              p_sam->s2=a[2];
0x000010a2              p_sam->s3=a[3];
0x000010ae              p_sam->s4=a[4];
0x000010ba              p_sam->s5=a[5];
0x000010c6              p_sam->s6=a[6];
0x000010d2              p_sam->s7=a[7];
0x000010de              p_sam->s8=a[8];
0x000010ea              p_sam->s9=a[9];
0x000010f6              delete p_sam;
0x000010fc          }
```

図 6-40　実行停止時の[ソース]ウィンドウ

[ステイタス]ウィンドウの表示内容は、以下のようになります。

ステイタス

Item	Status
Connected To:	E6000 H8S/2237 Emulator CPU 2000 (Emulator PCI Card Driver)
CPU	H8S/2237
Mode	7
Clock source	Main:10MHz, Sub:32kHz
Run status	Break
Cause of last break	Complex Event System
Event Time Count	00h 00min 00s 000ms 000us 000ns
Run Time Count	00h 00min 00s 000ms 826us 500ns

Memory | Platform | Events

図 6-41　[ステイタス]ウィンドウの表示内容

[ウォッチ]ウィンドウから変数iの値を参照します。変数iの値は4となっており、イベントが５回成立した事によってブレークが発生したことがわかります。

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

設定したイベントポイントを解除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除してください。

6.17 トレース機能

E6000 エミュレータのトレース機能がもつリアルタイムトレースバッファでは、32768 までのバスサイクルを保持でき、実行中は常に更新されます。リアルタイムトレースバッファの内容は[Trace]ウィンドウに表示されます。

[表示]メニューから[コード]サブメニューを選択し、[トレース]を選択してください。[Trace]ウィンドウが表示されます。

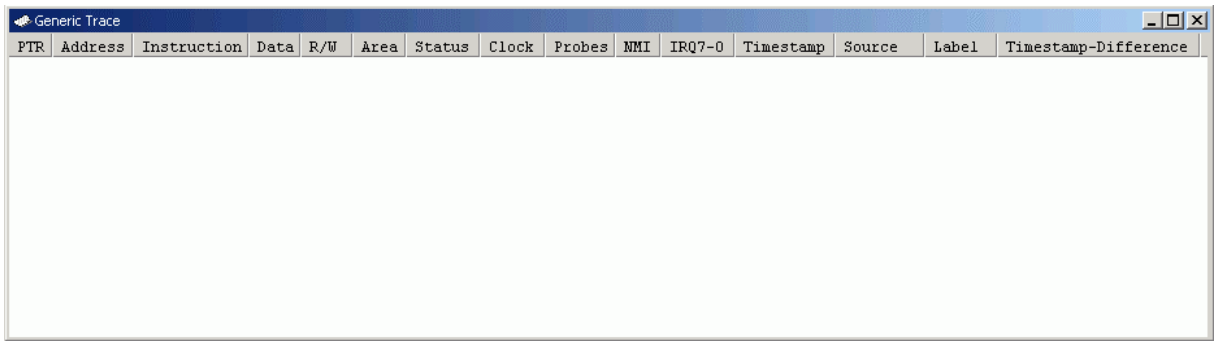


図 6-42 [Trace]ウィンドウ

[Trace]ウィンドウにトレース情報が表示されている場合は、マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

以下にトレース機能の概要と設定方法について説明します。

6.17.1 トレースの表示（タイムスタンプ無効時）

指定したアドレスに対するリード／ライトサイクルをトレース表示します。

(1)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[Trace Acquisition]ダイアログボックスが表示されます。

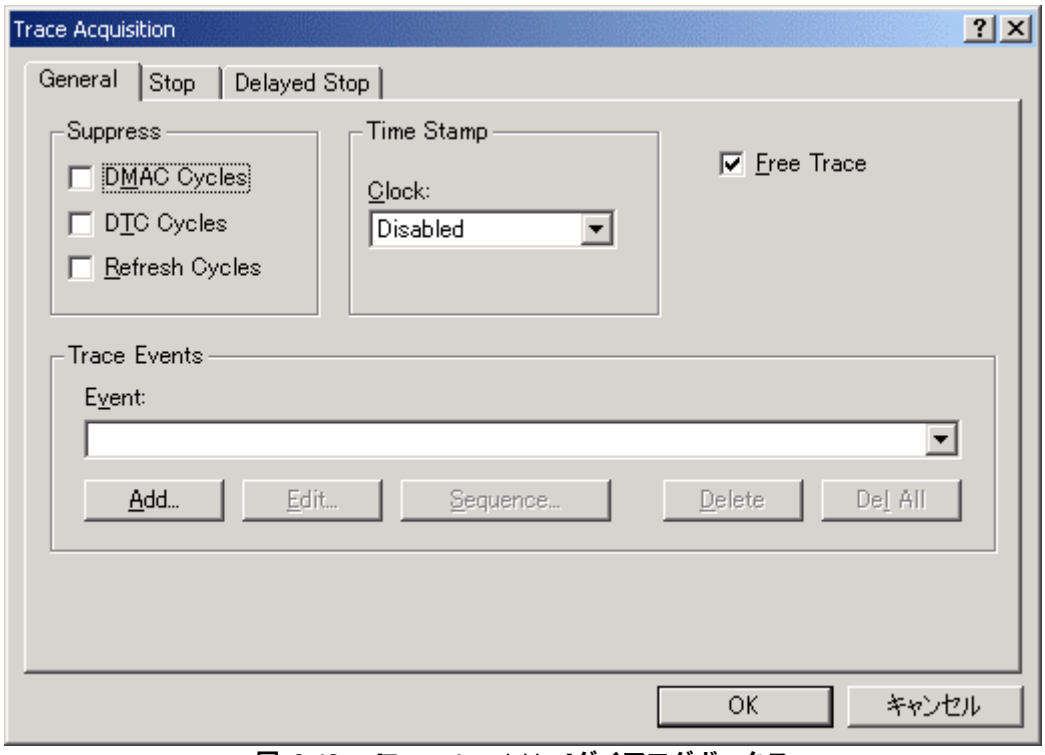


図 6-43 [Trace Acquisition]**ダイアログボックス**

(2)トレースを取得するアドレスを指定する為に、イベントを登録します。[General]ページの[Trace Events]グループボックスにある[Add…]ボタンをクリックして[Breakpoint/Event Properties]ダイアログボックスを表示します。

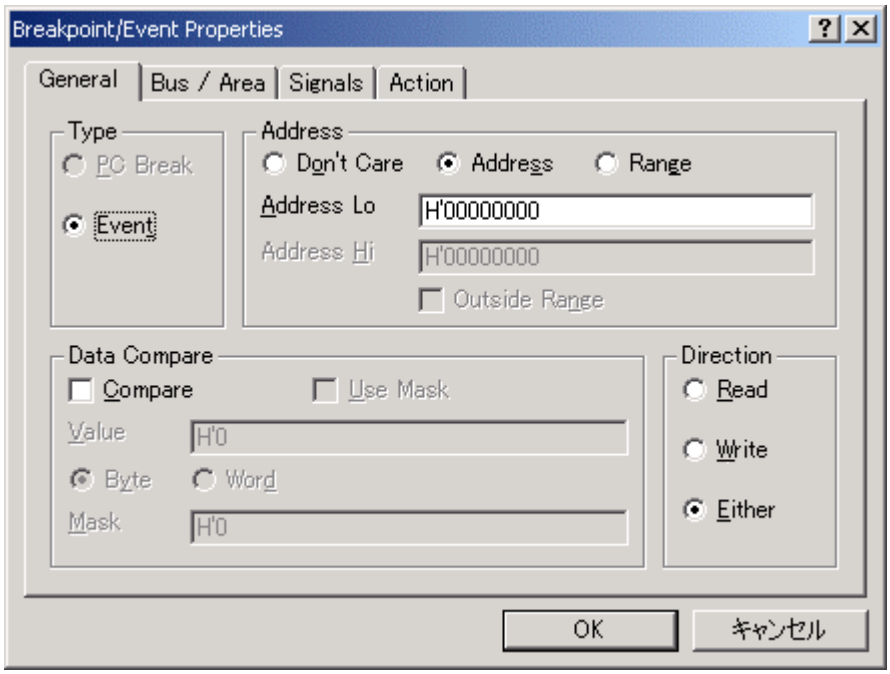


図 6-44 [Breakpoint/Event Properties]**ダイアログボックス**

(3)[Breakpoint/Event Properties]ダイアログボックス-[General]ページの[Address]グループボックスの[Address Lo]エディットボックスに tutorial 関数内の「a[i]=j;」と記述されている行のアドレスを[ソース]ウィンドウで参照し入力してください。本例では H'0000105c を入力します。これでアドレスの指定は終了です。[OK]ボタンをクリックし、[Breakpoint/Event Properties]ダイアログボックスを閉じます。

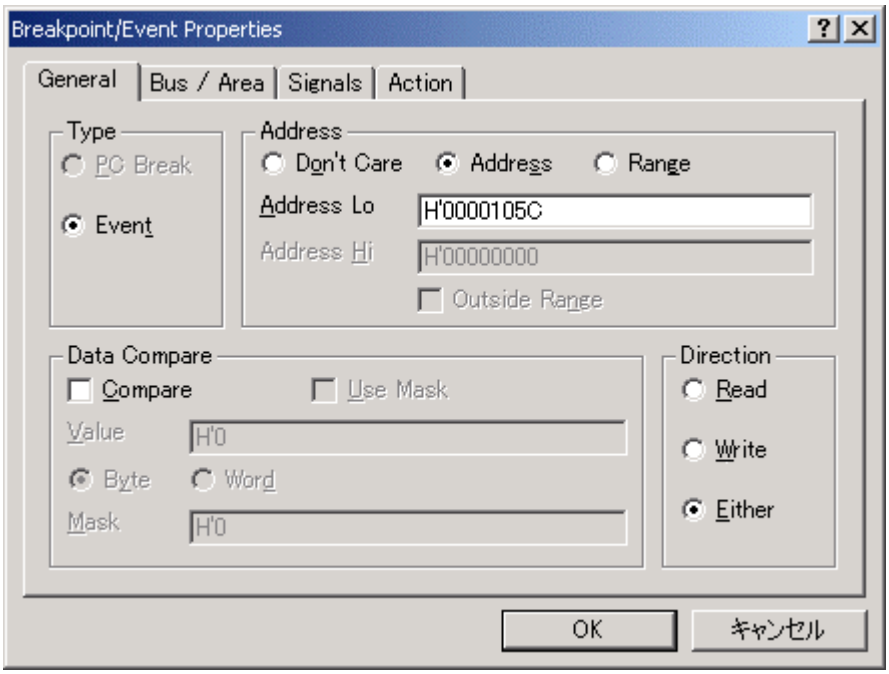


図 6-45 [Breakpoint/Event Properties]**ダイアログボックス（イベント設定後）**

(4)[Trace Acquisition]ダイアログボックス-[General]ページの[Trace Events]グループボックスにある[Event]ドロップダウンリストボックスに設定したイベントの内容が表示されます。

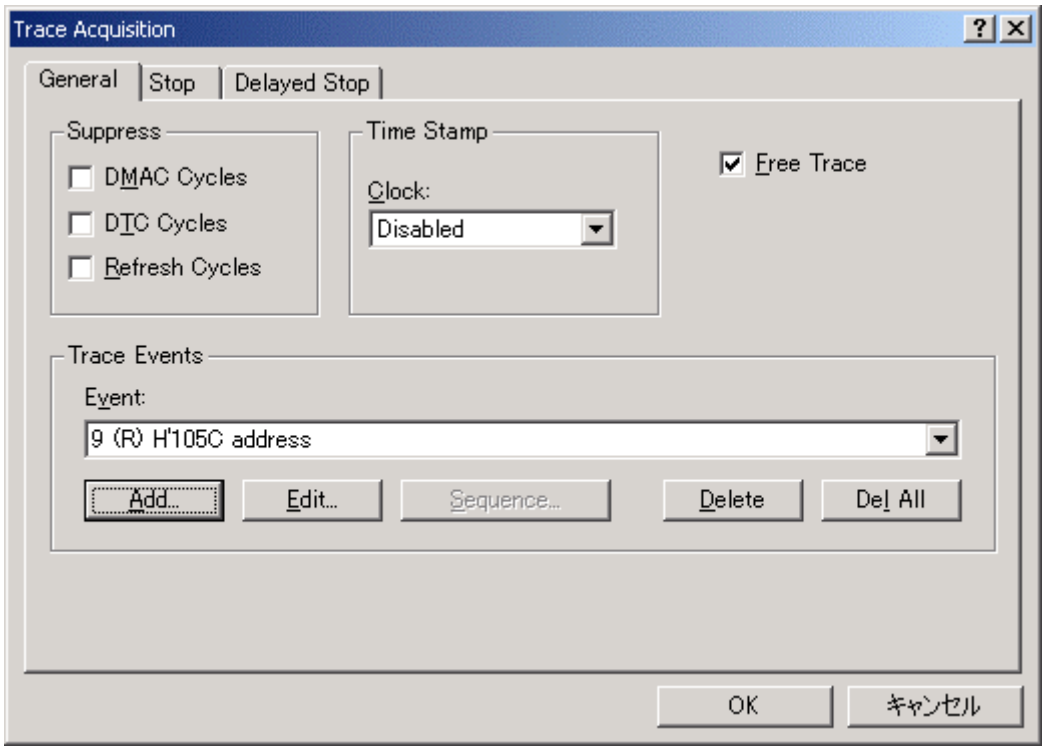


図 6-46 [Trace Acquisition]**ダイアログボックス（イベント追加）**

(5)設定したイベントを有効とする為に、[General]ページの[Free Trace]チェックボックスのチェックを解除します。チェックを解除すると新たに[1]～[4]のページが追加されます。

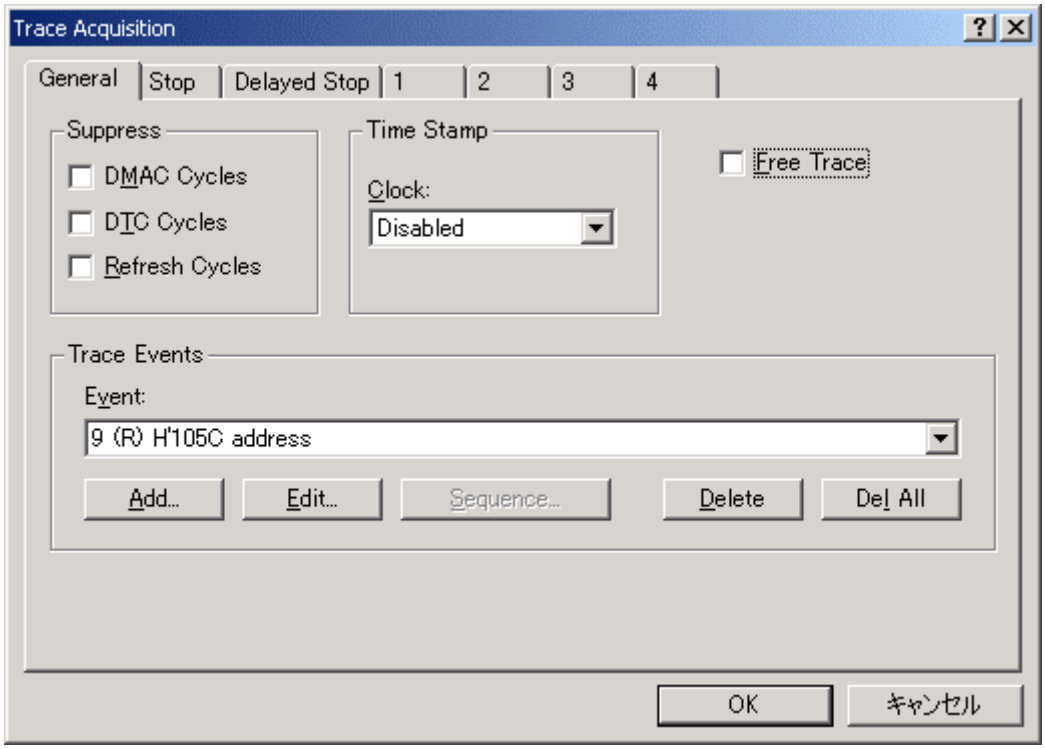


図 6-47　[Trace Acquisition]**ダイアログボックス（ページ追加）**

(6)[1]ページを選択し、[Conditions]グループボックスの[Range]ラジオボタンをクリックします。クリックと同時に[Range Event]ドロップダウンリストボックスと[Edit…]ボタンが表示されます。

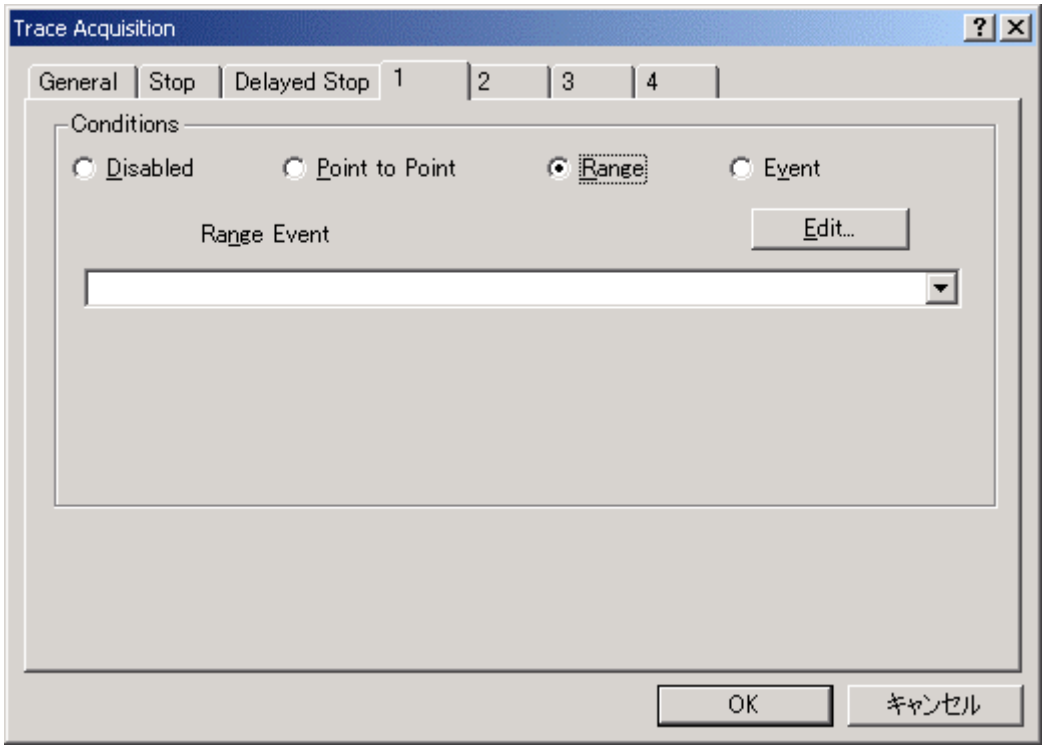


図 6-48　[Trace Acquisition]**ダイアログボックス（１ページ表示）**

(7)[Range Event]ドロップダウンリストボックスから、先ほど登録したイベントを選択します。これでイベントが有効となりました。[OK]ボタンをクリックし、トレースの設定は終了です。

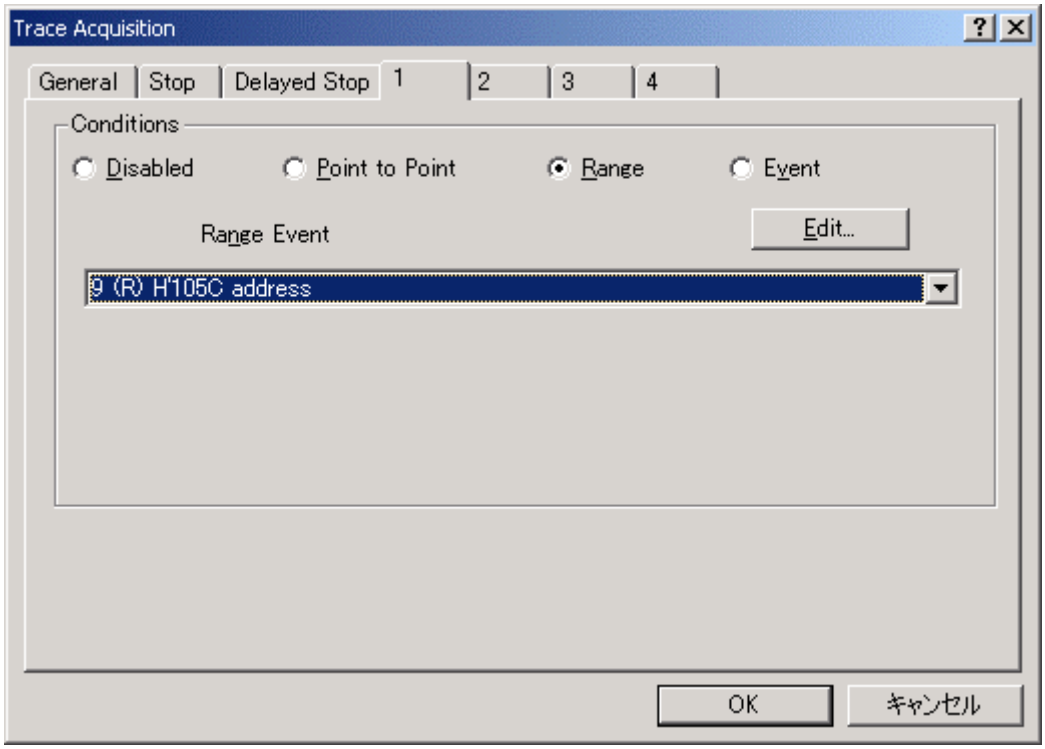


図 6-49 [Trace Acquisition]**ダイアログボックス（設定完了時）**

(8)指定した tutorial 関数内の「a[i]=j;」と記述されている行のアドレス（本例では H'0000105c）が５回実行された時点でブレークするように設定します。（「6.16.2 イベントポイントによるブレーク機能」参照）

(9)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウに以下の内容が表示されます。

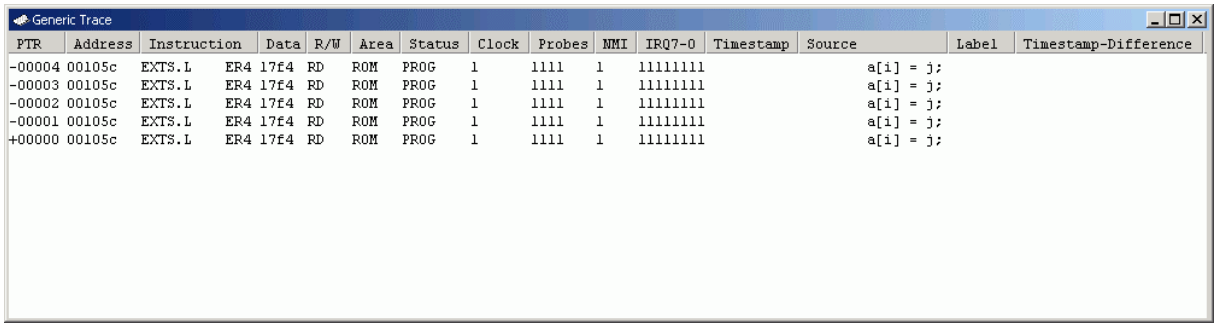


図 6-50 [Trace]**ウィンドウ（結果表示）**

必要ならば、タイトルバーの下のヘッダバーをドラッグして、カラムの幅を調節してください。

(10)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.17.2　トレースの表示（タイムスタンプ有効時）

指定した範囲のメモリに対するライトサイクルを、タイムスタンプを設定し、トレース表示します。

(1)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[設定…]を選択してください。[Trace Acquisition]ダイアログボックスが表示されます。（図 6-43　[Trace Acquisition]ダイアログボックス）

(2)トレースを取得するメモリの範囲を指定する為に、イベントを登録します。[General]ページの[Trace Events]グループボックスにある[Add…]ボタンをクリックして[Breakpoint/Event Properties]ダイアログボックスを表示します。（図 6-44　[Breakpoint/Event Properties]ダイアログボックス）

(3) [Breakpoint/Event Properties]ダイアログボックス-[General]ページの[Address]グループボックスの[Range]ラジオボタンをクリックし、[Address Lo]エディットボックスに tutorial 関数で定義している変数 a が割りついているアドレスを[ローカル]ウィンドウで参照し入力してください。本例では H’00FFEF80 を入力します。また、[Address Hi]エディットボックスには[Address Lo]エディットボックスに入力したアドレスに H’27 を加算したアドレスを入力してください。本例では、H’00FFEFA7 を入力します。この設定により tutorial 関数の変数 a のメモリ領域を指定したことになります。

(4)同様に[Direction]グループボックスの[Write]ラジオボタンをクリックし、設定した範囲に対するライトサイクルを指定します。これでメモリの範囲指定は終了です。[OK]ボタンをクリックし、[Breakpoint/Event Properties]ダイアログボックスを閉じます。

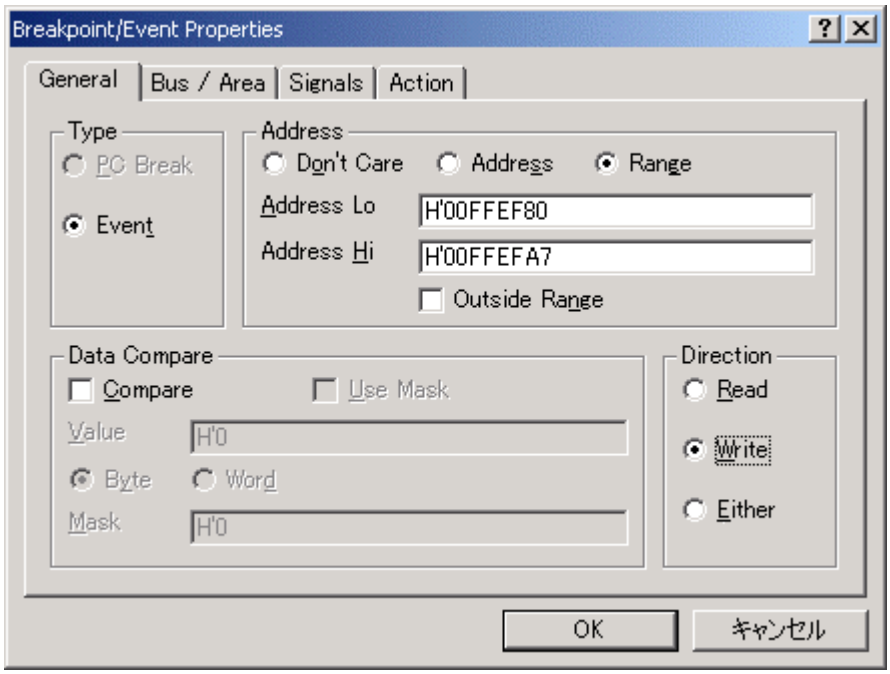


図 6-51　[Breakpoint/Event Properties]**ダイアログボックス（イベント設定後）**

(5)[Trace Acquisition]ダイアログボックス-[General]ページの[Trace Events]グループボックスにある[Event]ドロップダウンリストボックスに設定したイベントの内容が表示されます。

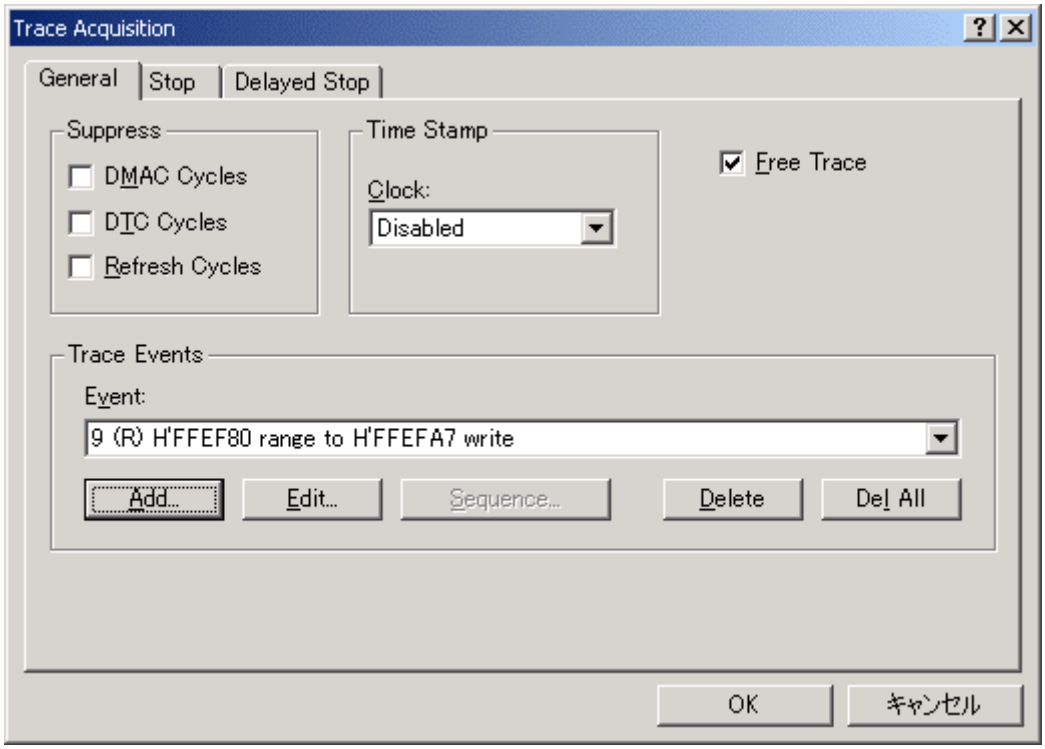


図 6-52 [Trace Acquisition]**ダイアログボックス（イベント追加時）**

(6)タイムスタンプを有効とする為に、[Time Stamp]グループボックスの[Clock]ドロップダウンリストボックスから 125ns を選択します。

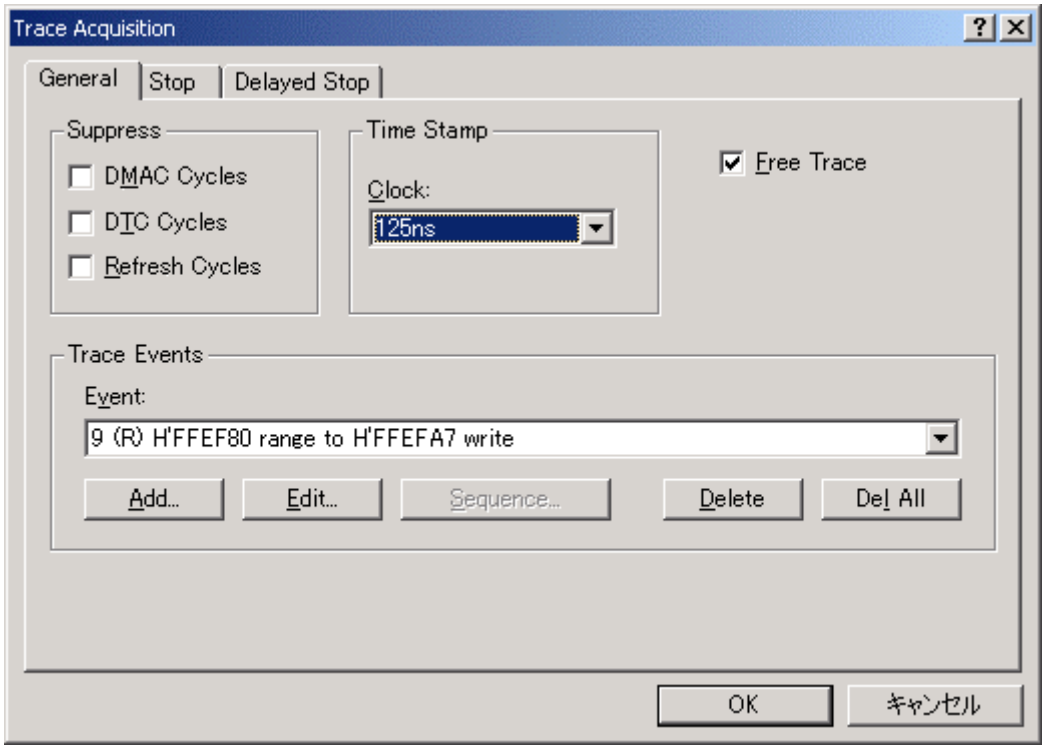


図 6-53 [Trace Acquisition]**ダイアログボックス（タイムスタンプ有効時）**

(7)設定したイベントを有効とする為に、[General]ページの[Free Trace]チェックボックスのチェックを解除します。チェックを解除すると新たに[1]～[4]のページが表示されます。（図 6-47　[Trace Acquisition]ダイアログボックス）

(8)[1]ページを選択し、[Conditions]グループボックスの[Range]ラジオボタンをクリックします。クリックと同時に[Range Event]ドロップダウンリストボックスと[Edit…]ボタンが表示されます。（図 6-48 [Trace Acquisition]ダイアログボックス）

(9)[Range Event]ドロップダウンリストボックスをクリックし、先ほど登録したイベントを選択します。これでイベントが有効となりました。[OK]ボタンをクリックし、トレースの設定は終了です。

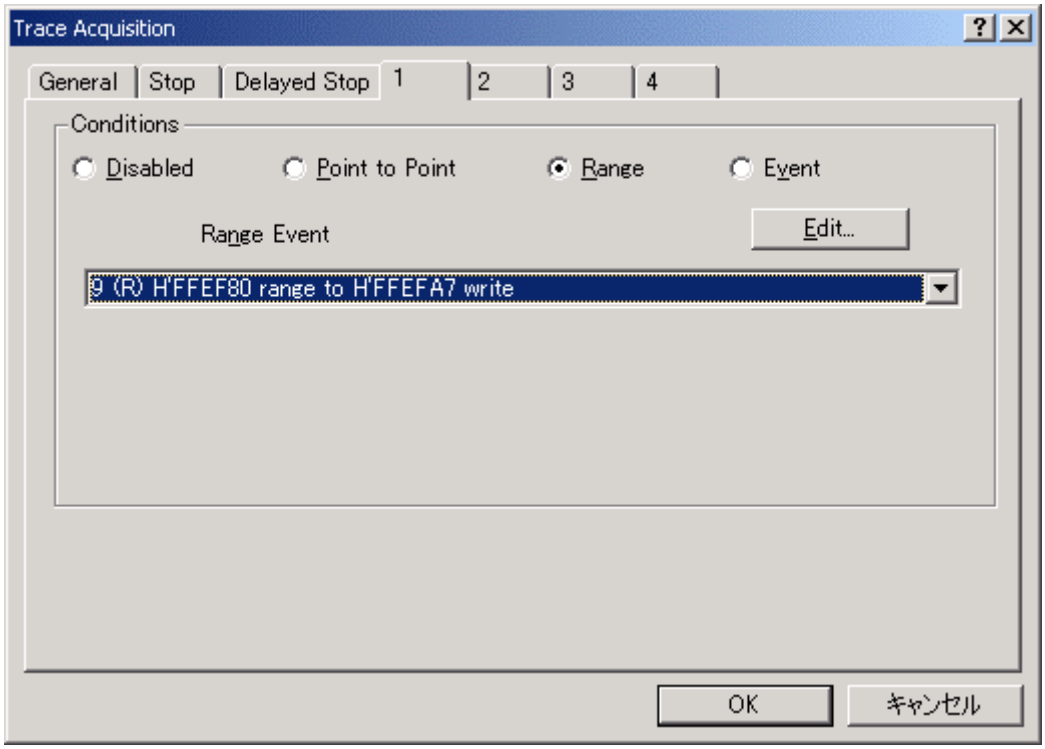


図 6-54　[Trace Acquisition]**ダイアログボックス（設定完了時）**

(10)tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス（本例では H’00001082）でブレークするように設定します。（「6.16.1 PC ブレーク機能」参照）

(11)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウに以下の内容が表示されます。

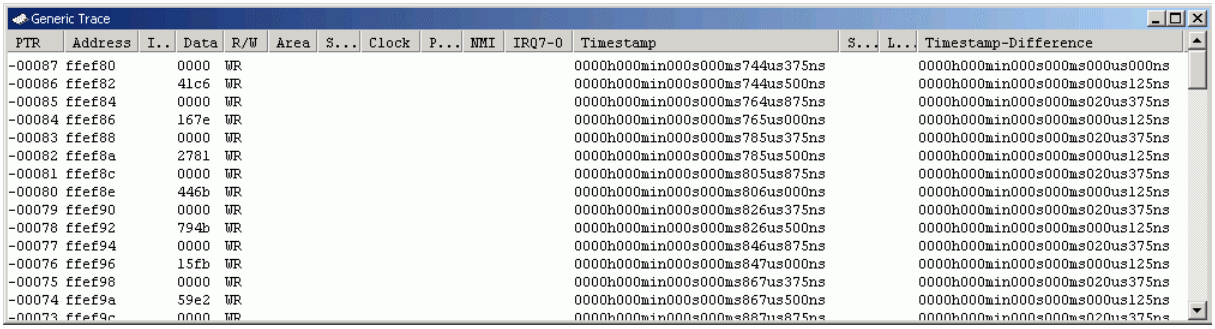


図 6-55　[Trace]**ウィンドウ（結果表示）**

必要ならば、タイトルバーの下のヘッダバーをドラッグして、カラムの幅を調節してください。

(12)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。また、タイムスタンプを無効とする場合には、[Trace Acquisition]ダイアログボックス-[General]ページの[Time Stamp]グループボックスの[Clock]ドロップダウンリストボックスからDisabledを選択します。

6.17.3 統計

取得したトレース情報から、内蔵 RAM に対する書き込み回数を収集します。

(1) tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス（本例では H’00001082）でブレークするように設定します。（「6.16.1 PC ブレーク機能」参照）

(2)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウにトレース情報が表示されます。

(3)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[統計…]を選択してください。トレースデータのロードを示すダイアログが表示された後に、[統計]ダイアログボックスが表示されます。

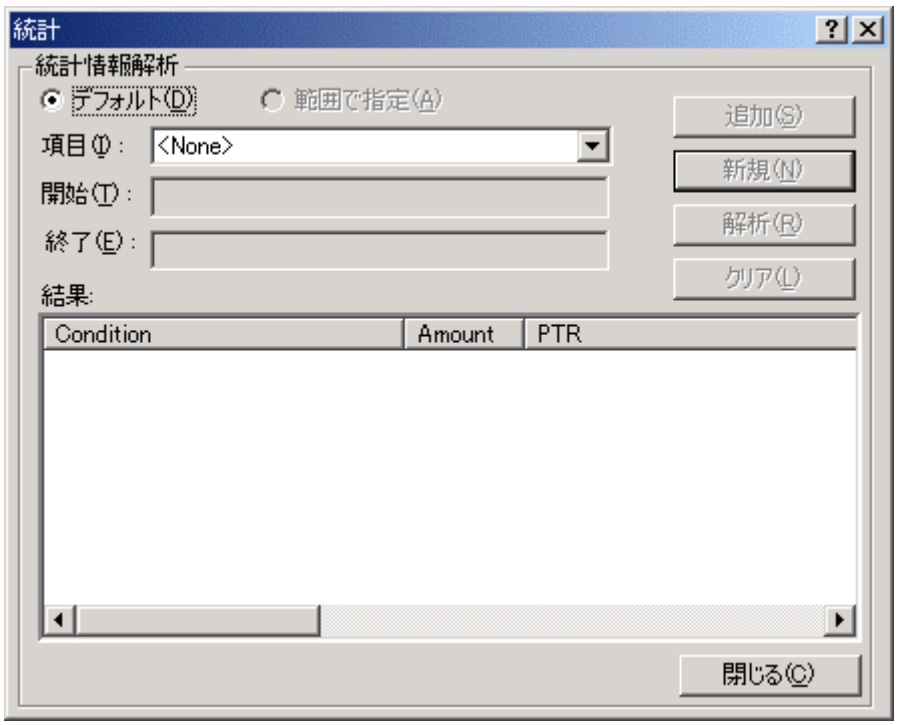


図 6-56 [統計]ダイアログボックス

(4)[項目]ドロップダウンリストボックスから R/W を選択し、[開始]エディットボックスに WR を入力します。入力終了後、[新規]ボタンをクリックすると[結果]リストボックスの[Condition]列に”R/W=WR”と表示されます。

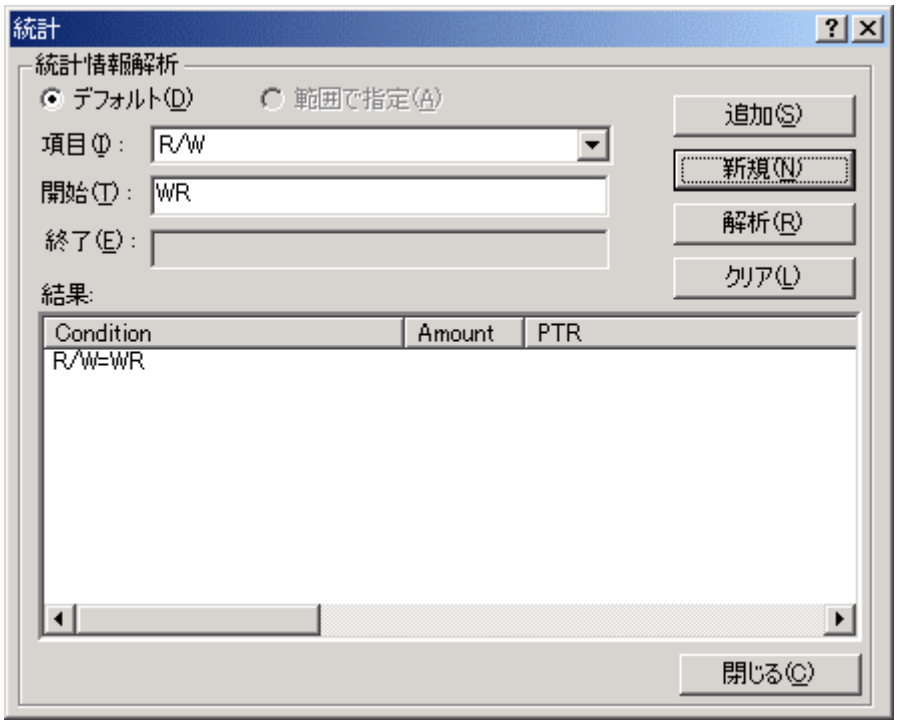


図 6-57　**[統計]ダイアログボックス（新規条件）**

(5)続いて[項目]ドロップダウンリストボックスから Area を選択し、[開始]エディットボックスに RAM を入力します。入力終了後、[追加]ボタンをクリックすると[結果]リストボックスの[Condition]列に表示されていた”R/W=WR”に条件が追加され、”R/W=WR & Area=RAM”と表示されます。これで条件の入力は終了です。

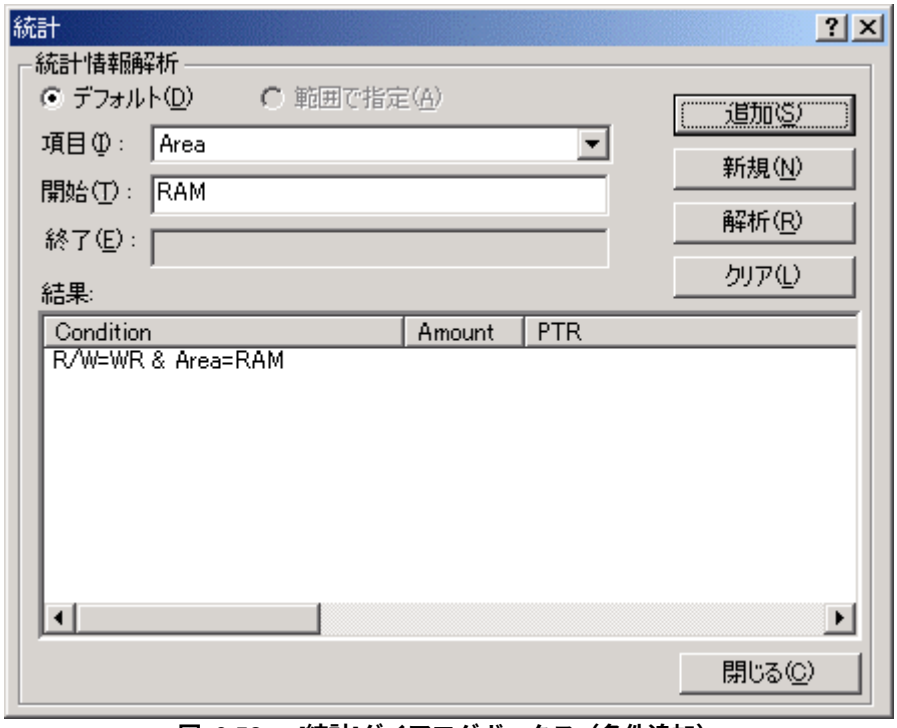


図 6-58　**[統計]ダイアログボックス（条件追加）**

(6)入力した条件による解析を行います。[解析]ボタンをクリックしてください。条件に該当する件数とその PTR が表示されます。

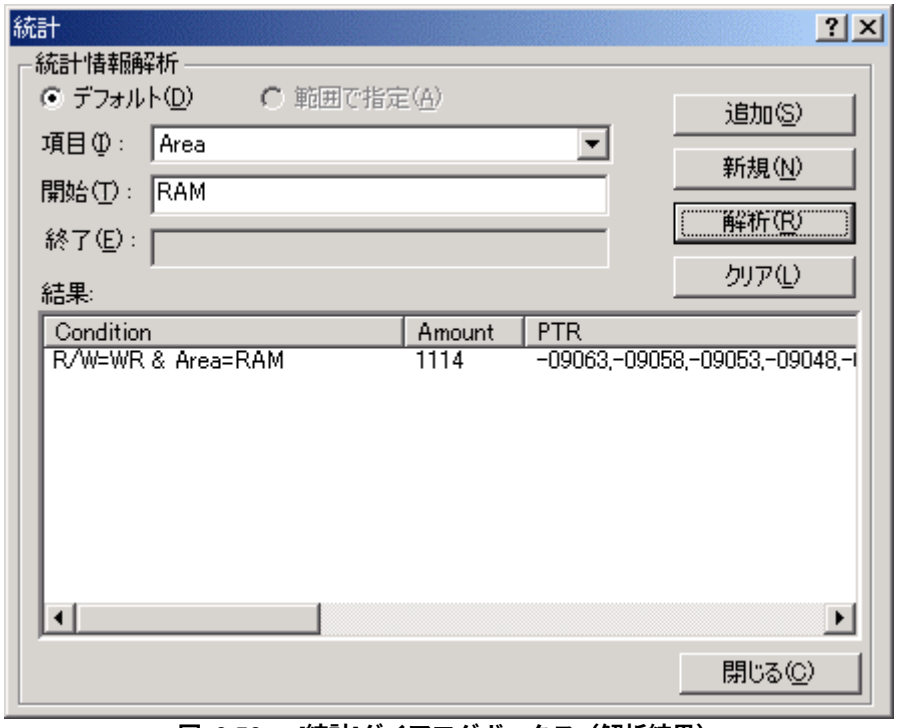


図 6-59 [統計]ダイアログボックス（解析結果）

(7)[閉じる]ボタンをクリックし、[統計]ダイアログボックスを閉じます。

(8)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.17.4 関数コール

取得したトレース情報から、関数をコールしている情報のみを収集します。

(1) tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス（本例では H'00001082）でブレークするように設定します。（「6.16.1 PC ブレーク機能」参照）

(2)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウにトレース情報が表示されます。

(3)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[関数コール…]を選択してください。[関数コール箇所の表示]ダイアログボックスが表示されます。

図 6-60 [関数コール箇所の表示]ダイアログボックス

(4)[許可]ラジオボタンをクリックし、[OK]ボタンをクリックしてください。[Trace]ウィンドウの表示が関数をコールしている情報のみに変更されます。（関数コールを確認する為、[Label]列を[Trace]ウィンドウの左端に移動しています）

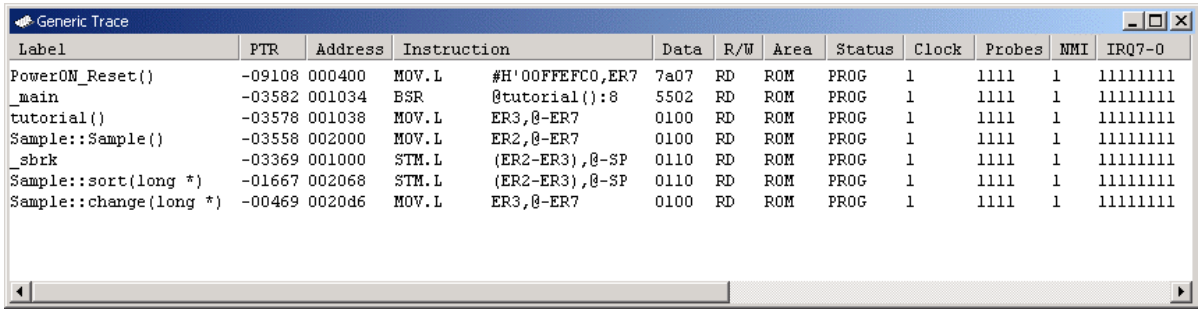


図 6-61 [Trace]ウィンドウ（関数コール）

(5)[Trace]ウィンドウの表示を元の状態に戻します。(3)の手順にて[関数コール箇所の表示]ダイアログボックスを表示し、[無効]ラジオボタンをクリック、その後[OK]ボタンをクリックします。

(6)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.18 スタックトレース機能

E6000 エミュレータでは、スタック情報を用いて、関数呼び出し履歴を表示します。

【注】 1. **本機能は、Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。Dwarf2 形式のデバッグ情報を持ったロードモジュールは、H8S,H8/300 C/C++コンパイラ V4.0 以降でサポートしています。**

2. **本機能の詳細については、オンラインヘルプを参照してください。**

sort 関数内の行の[Editor]カラムをダブルクリックして、PC ブレークポイントを設定してください。

```
0x00002068          void Sample::sort(long *a)
0x00002070          {
                        long t;
                        int i, j, k, gap;

0x00002072              gap = 5;
0x00002076              while( gap > 0 ){
0x00002078                  for( k=0; k<gap; k++){
0x0000207c                      for( i=k+gap; i<10; i=i+gap ){
0x00002080                          for(j=i-gap; j>=k; j=j-gap){
0x00002084                              if(a[j]>a[j+gap]){
                                            t = a[j];
0x000020a4                                  a[j] = a[j+gap];
0x000020a8                                  a[j+gap] = t;
                                        }
                                        else
                                            break;
                                    }
                                }
                            }
0x000020c0                  gap = gap/2;
                        }
0x000020cc          }
```

図 6-62 [ソース]ウィンドウ（PCブレークポイントの設定）

[デバッグ]メニューから[リセット後実行]を選択してください。

ブレーク後、[表示]メニューから[コード]サブメニューを選択し、[スタックトレース]を選択し[スタックトレース]ウィンドウを開いてください。

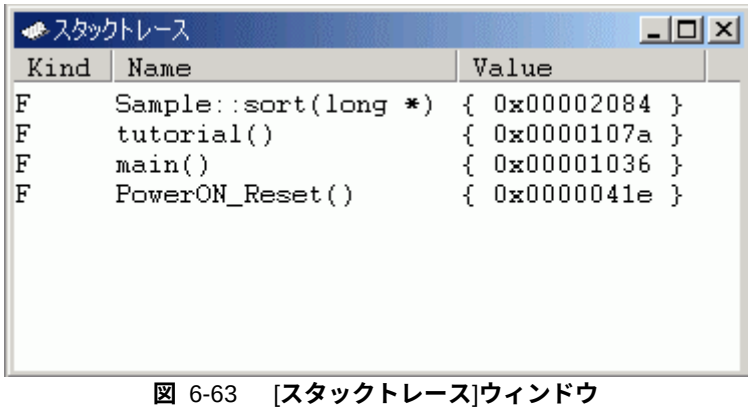


図 6-63 [スタックトレース]ウィンドウ

現在 PC が sort()関数内にあり、sort()関数は、tutorial()関数からコールされていることがわかります。

sort 関数内の行の[Editor]カラムを再度ダブルクリックして、PC ブレークポイントを解除します。

6.19 モニタ機能

E6000 エミュレータには、指定したアドレスのメモリ内容を、ユーザプログラム実行中にモニタすることができます。

ここでは tutorial 関数の変数 a が割り当てられたアドレス近辺の内容をモニタします。

(1)[表示]メニューから[CPU]サブメニューを選択、さらに[モニタ]サブメニューを選択し、[モニタ設定...]を選択します。[Monitor Setting]ダイアログボックスが表示されます。

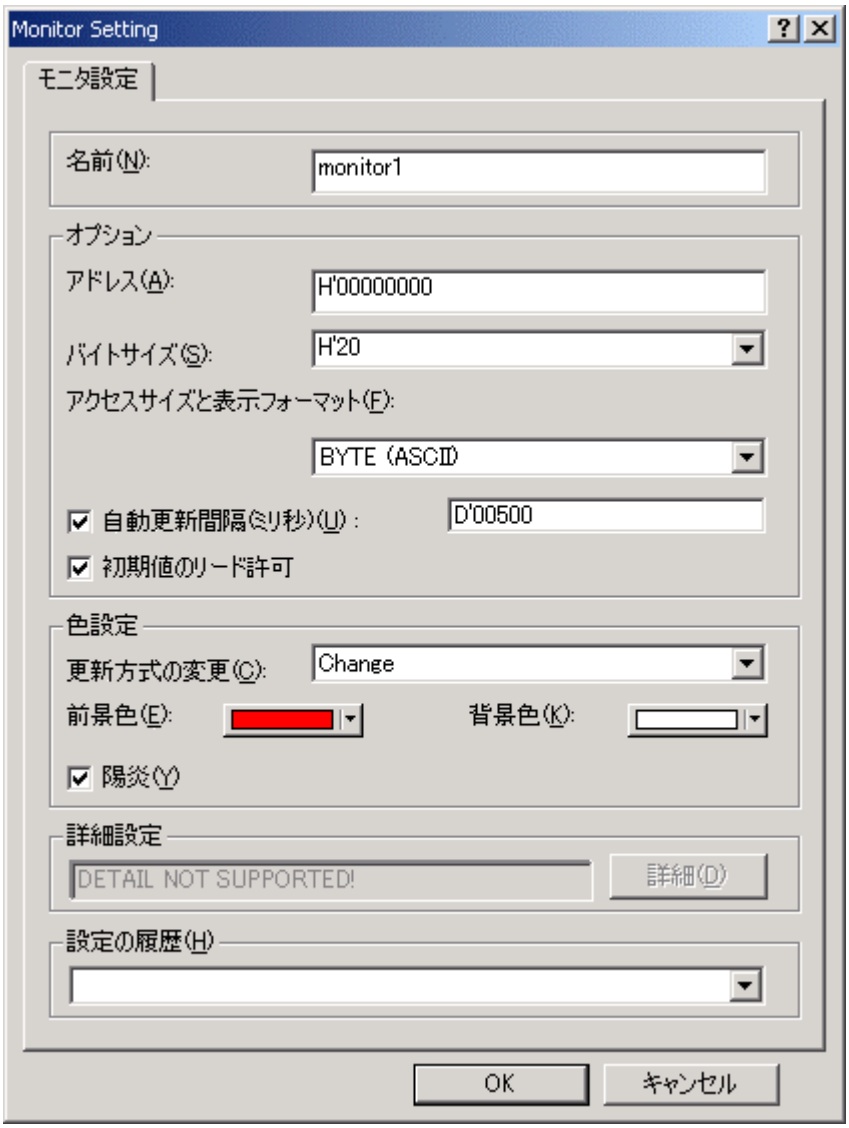


図 6-64 [Monitor Setting]ダイアログボックス

(2)[モニタ設定]ページの内容を以下のように設定します。

- [名前]エディットボックスに monitor1 と入力します。
- [オプション]グループボックスの各パラメータを以下の設定とします。
  - ① [アドレス]エディットボックスに tutorial 関数で定義している変数 a が割りついているアドレスを[ローカル]ウィンドウで参照し入力してください。本例では H'00FFEF80 を入力します。
  - ② [バイトサイズ]ドロップダウンリストボックスに H'50 と入力します。
  - ③ [アクセスサイズと表示フォーマット]ドロップダウンリストボックスから BYTE（HEX）を選択します。
  - ④ [自動更新間隔(ミリ秒)]のチェックボックスをチェックし、エディットボックスに D'00500 と入力します。
  - ⑤ [初期値のリード許可]チェックボックスをチェックします。

・ [色指定]グループボックスの各パラメータを以下の設定とします。
① [更新方式の変更]ドロップダウンリストボックスから Change を選択します。
② [前景色]ドロップダウンリストボックスを赤に、[背景色]ドロップダウンリストボックスを白に選択します。
③ [陽炎]チェックボックスをチェックします。

【注】前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

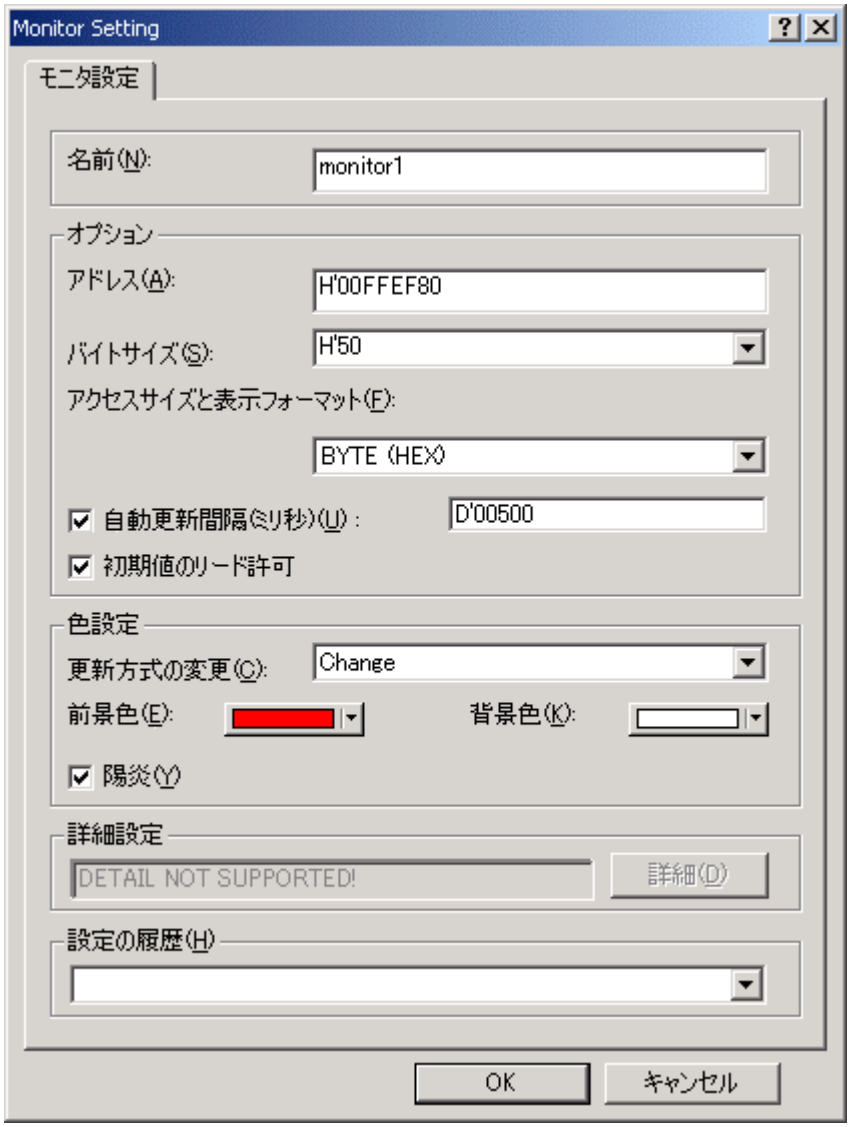
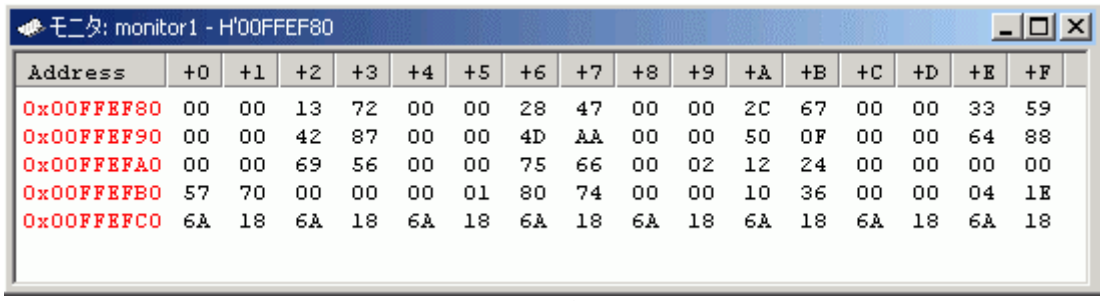


図 6-65 [Monitor Setting]**ダイアログボックス（設定完了時）**

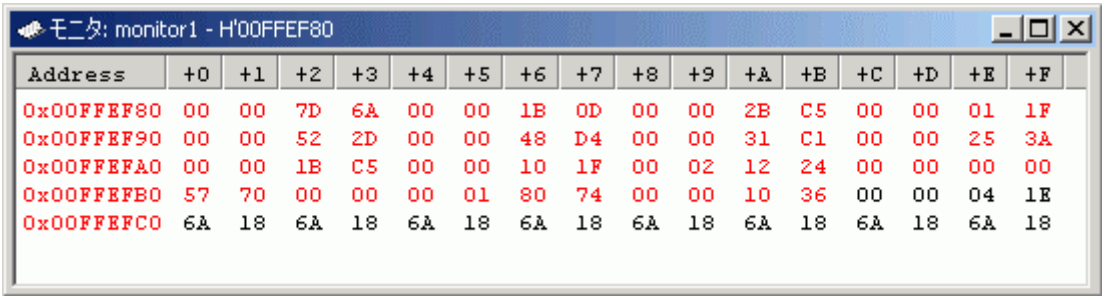
(3)[OK]ボタンをクリックします。[モニタ]ウィンドウが開きます。



Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x00FFEF80	00	00	13	72	00	00	28	47	00	00	2C	67	00	00	33	59
0x00FFEF90	00	00	42	87	00	00	4D	AA	00	00	50	0F	00	00	64	88
0x00FFEFA0	00	00	69	56	00	00	75	66	00	02	12	24	00	00	00	00
0x00FFEFB0	57	70	00	00	00	01	80	74	00	00	10	36	00	00	04	1E
0x00FFEFC0	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18

図 6-66 [モニタ]ウィンドウ

(4)[デバッグ]メニューから[リセット後実行]を選択してください。処理が実行されることによりメモリの内容が更新された場合は、更新されたメモリの値が赤（[モニタ設定]ページの[前景色]ドロップダウンリストボックスと[背景色]ドロップダウンリストボックスで設定した色）に変更されます。また、更新が行われないメモリの値および、更新後一定間隔更新が行われなかったメモリの値は黒で表示されます。



Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x00FFEF80	00	00	7D	6A	00	00	1B	0D	00	00	2B	C5	00	00	01	1F
0x00FFEF90	00	00	52	2D	00	00	48	D4	00	00	31	C1	00	00	25	3A
0x00FFEFA0	00	00	1B	C5	00	00	10	1F	00	02	12	24	00	00	00	00
0x00FFEFB0	57	70	00	00	00	01	80	74	00	00	10	36	00	00	04	1E
0x00FFEFC0	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18

図 6-67 [モニタ]ウィンドウ（実行中）

(5)[モニタ]ウィンドウの表示状況が確認できましたら、[デバッグ]メニューから[プログラムの停止]を選択し、実行を停止します。

6.20 さてつぎは？

このチュートリアルでは、E6000 エミュレータのいくつかの主な特徴と、HEW の使い方を紹介しました。

E6000 エミュレータで提供されるエミュレーション機能を使用することによって、高度なデバッグを行うことができます。それによって、ハードウェアとソフトウェアの問題が発生する条件を正確に分離し、識別すると、それらの問題点を効果的に調査することができます。

7　本製品固有のハードウェア仕様

この章では、H8/388R E6000 エミュレータに関するハードウェア仕様について説明します。

7.1　H8/388R E6000 エミュレータ仕様

H8/388R E6000 エミュレータは下記のマイクロコンピュータを使用したシステムの開発をサポートします。

サポートデバイス

- H8/3847R グループ
- H8/3847S グループ
- H8/3827R グループ
- H8/3827S グループ
- H8/3937R グループ
- H8/3887 グループ
- H8/3867 グループ
- H8/3847 グループ
- H8/3827 グループ
- H8/3937 グループ

7.1.1　サポート範囲

H8/3937R グループのエミュレーションを行う場合には、拡張 I/O ボード（HS3937REIO60H）が必要です。
H8/3937 グループのエミュレーションを行う場合には、拡張 I/O ボード（HS3937EIO60H）が必要です。
各ユーザケーブル、拡張 I/O ボードの注意事項についてはユーザケーブル、拡張 I/O ボードに添付されている取り扱い説明書を参照してください。
本 E6000 エミュレータがサポートする MCU 型名と対応するパッケージ、E6000 ユーザシステムインタフェースケーブルおよびオプションボードの組み合わせにつきましては、環境開発カタログを参照してください。

7.1.2 動作電圧および動作周波数

以下に本 E6000 エミュレータがサポートする MCU の動作電圧および動作周波数仕様について示します。MCU の保証する動作電圧・動作周波数を越えた状態で E6000 エミュレータを使用した場合、E6000 エミュレータは正常に動作しないため注意してください。

また、MCU によっては低電圧動作、高周波数動作を保証しないものがありますので注意してください。

表7-1 動作電圧および動作周波数

No.	MCU 型名	動作電圧（V）	動作周波数範囲φ（MHz）
1	H8/3847R グループ H8/3827R グループ	1.8-5.5	1.0-2.0
		2.7-5.5	1.0-5.0
		4.5-5.5	1.0-8.0
2	H8/3847S グループ H8/3827S グループ	1.8-3.6	0.5-2.0
		2.7-3.6	0.5-5.0
3	H8/3887 グループ H8/3867 グループ H8/3847 グループ H8/3827 グループ	1.8-5.5	0.2-0.5
		2.2-5.5	0.2-1.0
		2.6-5.5	0.2-1.6
		3.0-5.5	0.2-2.0
		4.5-5.5	0.2-3.0
4	H8/3937R グループ H8/3937 グループ	1.8-3.6	1.0-2.0
		2.7-3.6	1.0-5.0

留意事項

動作電圧および動作周波数範囲の詳細については、各MCUのハードウェアマニュアルでご確認ください。

7.2　ユーザシステムインタフェース回路

E6000 エミュレータのユーザシステムインタフェース信号は、バッファなしに直接エミュレータ上の MCU に接続されています。

7.2.1　信号保護

ユーザシステムインタフェース信号は、ダイオードによって、過大／過小電圧から保護されています。ただし、AVcc とアナログポートには、この保護回路がありません。

アナログポート以外のポートには、プルアップ抵抗が接続されています。

ユーザシステムインタフェースケーブル先端部の Vcc 端子（AVcc 端子を除く）は、すべて１つに接続されています。E6000 エミュレータは、Vcc 端子の電圧レベルを監視して[拡張モニタ]ウィンドウに電源状態を表示します。

なお、ユーザシステムインタフェースケーブル未接続の場合、MCU の Uvcc は 5V になります。

7.2.2　ユーザインタフェース回路

E6000 エミュレータのユーザインタフェースには、ケーブルにより約 8ns の信号の遅れが生じます。また、プルアップ抵抗により信号がハイインピーダンス状態でもハイレベルになります。このことを考慮してユーザシステムのハードウェアを調整してください。

(1)　**以下に記述のない信号**

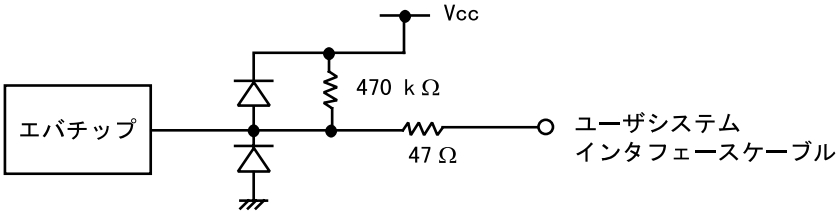


図 7-1　**ユーザインタフェース信号回路**

(2)　OSC1、X1

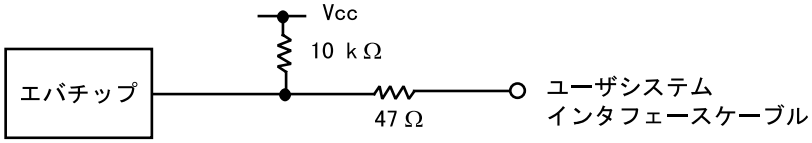


図 7-2　OSC1 **および** X1 **信号**

(3) PB0/AN0〜PB7/AN7、PC0〜PC3、
P50/WKP0/SEG1〜P57/WKP7/SEG8、
P60/SEG9〜P67/SEG16、
P70/SEG17〜P77/SEG24、
P80/SEG25〜P87/SEG32、
P90/SEG33〜P97/SEG40/CL1、
PA0/COM1〜PA3/COM4

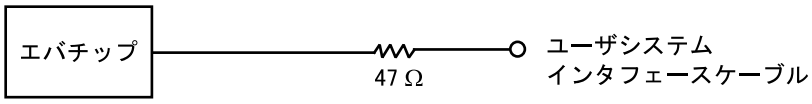


図 7-3 PB0/AN0〜PB7/AN7、PC0〜PC3、P50/WKP0/SEG1〜P57/WKP7/SEG8、P60/SEG9〜P67/SEG16、P70/SEG17〜P77/SEG24、P80/SEG25〜P87/SEG32、P90/SEG33〜P97/SEG40/CL1、PA0/COM1〜PA3/COM4 **信号**

(4) AVcc、AVss

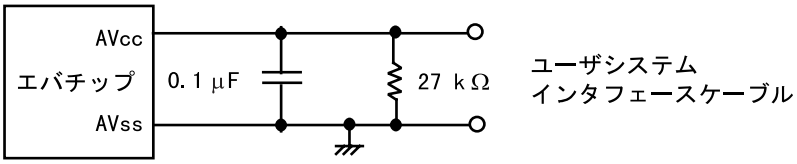


図 7-4 AVcc、Avss **電源端子**

(5) CVcc、TEST

CVcc が GND に接続されている場合、または TEST が Vcc レベルに接続されている場合は、エミュレータ接続時にメッセージが出力されます。

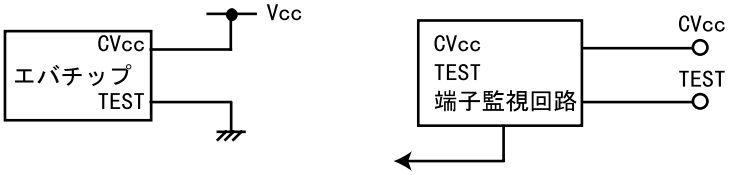


図 7-5 CVcc、TEST **信号**

(6) V0、V1、V2、V3

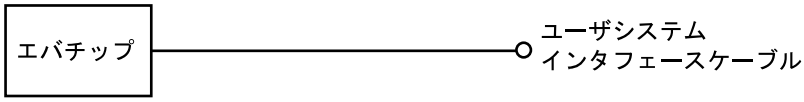


図 7-6 V0、V1、V2、V3 **電源端子**

(7) P30/PWM〜P37/AEVL、P40/RCK32〜P43/IRQ0

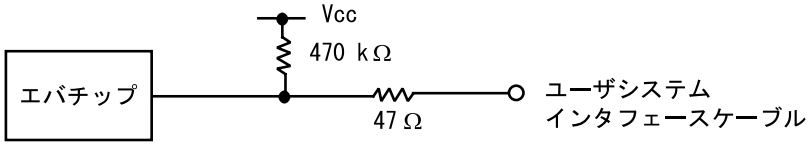


図 7-7 P30/PWM〜P37/AEVL、P40/RCK32〜P43/IRQ0 **信号**

7.3　MCU と E6000 エミュレータの相違点

E6000 エミュレータの電源投入後、初期化後およびリセット後の、MCU と E6000 エミュレータのレジスタの初期値の相違を以下に示します。

表7-2　MCU と E6000 エミュレータの相違

状態	レジスタ	E6000 エミュレータ	MCU
電源投入後／初期化後	PC	不定	リセットベクタ値
	R0 to R6	H'00	不定
	R7（SP）	H'10	不定
	CCR	I マスクは 1 その他は不定	I マスクは 1 その他は不定
リセット後	PC	リセットベクタ値	リセットベクタ値
	R0 to R6	不定	不定
	R7（SP）	H'10	不定
	CCR	I マスクは 1 その他は不定	I マスクは 1 その他は不定

7.3.1　A/D コンバータ

ユーザシステムインタフェースケーブルで接続されているため、A/D 変換の精度は、MCU のハードウェアマニュアルに記載の精度より劣下します。

7.3.2　未使用領域のアクセス

未使用領域 H'FF80～H'FF8F はエミュレータシステムで使用しています。MAP 設定でエミュレータに割り付けた場合、正常動作できませんので、本領域は使用しないでください。

また、MCU が使用していないレジスタについてはエミュレータでアクセスしないでください。

7.3.3　Go Reset コマンドによるプログラム実行

Go Reset コマンドを用いてプログラムを実行する際、E6000 は約 500μs のリセット信号をエバチップに入力します。実行時間測定結果には、このリセット信号入力時間が加算されます。

8　本製品固有のソフトウェア仕様

この章では、H8/388R E6000 エミュレータに関するソフトウェア仕様について説明します。

8.1　H8/388R E6000 エミュレータソフトウェア仕様

本エミュレータ固有の情報を以下に示します。

8.1.1　対応ハードウェア

本エミュレータソフトウェアは H8/388R E6000 エミュレータ（HS3880EPI60H, HS388REPI60H）に対応したソフトウェアです。

8.1.2　選択可能プラットフォーム

本エミュレータで選択可能なデバッギングプラットフォームは以下の通りです。

表8-1　**選択可能ターゲット**

デバッギングプラットフォーム	備考
H8/3827R E6000 Emulator CPU 300L	H8/300L CPU **をコアとする以下のデバイスのエミュレーションを行います。** H8/3822R, H8/3823R, H8/3824R, H8/3825R, H8/3826R, H8/3827R, H8/3824S, H8/3825S, H8/3826S, H8/3827S
H8/3847R E6000 Emulator CPU 300L	H8/300L CPU **をコアとする以下のデバイスのエミュレーションを行います。** H8/3842R, H8/3843R, H8/3844R, H8/3845R, H8/3846R, H8/3847R H8/3844S, H8/3845S, H8/3846S, H8/3847S
H8/3867 E6000 Emulator CPU 300L	H8/300L CPU **をコアとする以下のデバイスのエミュレーションを行います。** H8/3862, H8/3863, H8/3864, H8/3865, H8/3866, H8/3867, H8/3822, H8/3823, H8/3824, H8/3825, H8/3826, H8/3827
H8/3887 E6000 Emulator CPU 300L	H8/300L CPU **をコアとする以下のデバイスのエミュレーションを行います。** H8/3882, H8/3883, H8/3884, H8/3885, H8/3886, H8/3887, H8/3842, H8/3843, H8/3844, H8/3845, H8/3846, H8/3847
H8/3937 E6000 Emulator CPU 300L	H8/300L CPU **をコアとし、拡張** I/O **ボード** HS3937EIO60H**、**HS3937REIO60H **を使用する以下のデバイスのエミュレーションを行います。** H8/3935R, H8/3935, H8/3936R, H8/3936, H8/3937R, H8/3937

8.1.3 Configuration Properties ダイアログボックス（General ページ）

本ダイアログボックスで設定可能な項目は以下の通りです。

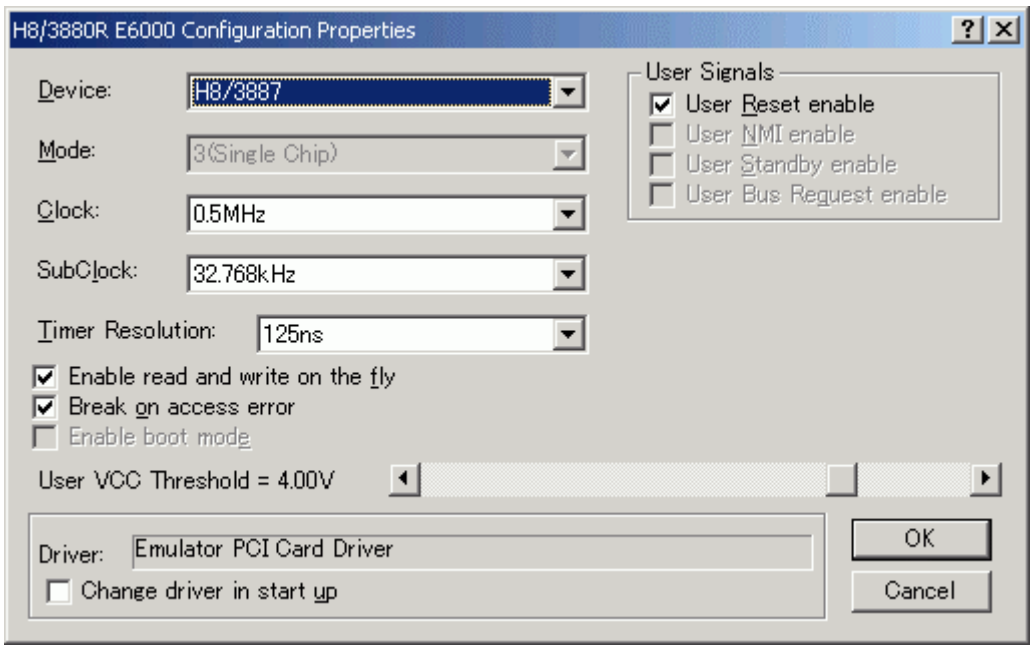


図 8-1 Configuration Properties ダイアログボックス（General ページ）

[General]

[Device]	エミュレーションする MCU を指定します。詳細に関しては、それぞれのハードウェアマニュアルを参照してください。
[Mode]	CPU 動作モードを指定します。本エミュレータではモード 3 のみの指定となります。
[Clock]	MCU のクロック速度を指定します。
[SubClock]	MCU のサブクロック速度を指定します。
[Timer Resolution]	実行時間の測定に使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 20ns, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us 実行時間測定用タイマは 40 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 20ns で約 6 時間、分解能 16us で約 200 日です。 カウンタがオーバフローした場合、オーバフローしたことを示すプロンプト”> ”と共に測定可能な最大時間を表示します。
[Enable read and write on the fly]	このチェックボックスをチェックすると、プログラム実行中にメモリにアクセスすることが可能です。 リアルタイム性はありませんので、リアルタイムでのエミュレーションを行いたい場合には、チェックしないでください。 • アクセスするメモリが内蔵 ROM、内蔵 RAM の場合 ユーザプログラムをブレークさせ、HEW から直接メモリにアクセスします。また、処理終了後プログラムを復帰させます。 • アクセスするメモリが内蔵 I/O の場合 ユーザプログラムをブレークして行います。 内蔵ＲＡM無効時、プログラム実行中にこの領域に対してのアクセスは不可となります。

【注】 ユーザプログラム実行中、メモリ内容を変更した場合（例: [メモリ]ウィンドウ、MEMORY_EDIT コマンド等による変更）、HEW は変更した値を表示するためメモリ内容のリードを行います。
また[メモリ（M）-> 最新の情報に更新（R）] 等のメモリ内容を更新する操作を行った場合も、メモリ内容のリードを行います。
このとき、メモリ内容を表示している各ウィンドウに対し、それぞれメモリ内容のリードを行い表示内容を更新します。
ユーザプログラム実行中に不要なメモリ内容のリードを抑止する場合はメモリ内容を表示しているウィンドウ（[メモリ]ウィンドウ、[逆アセンブリ]ウィンドウ等）を閉じるか、または更新を行わない設定としてください。
ただし､[モニタ]ウィンドウまたは下記条件を満たす[ウォッチ]ウィンドウはメモリ内容を表示していますが、メモリ内容更新の実現方法が異なるため、ウィンドウを開いていてもリアルタイム性を阻害しません。

[条件]

1. **登録したシンボルが汎用レジスタにのみ割りついている**
2. **登録したシンボルが[モニタ]機能により設定したモニタ範囲にのみ割りついている（R マークが青色となります）**
3. **登録したシンボルが上記 1 および 2 で構成されている**

[Break on access error]	このチェックボックスをチェックすると、プログラムでアクセス禁止エリアにアクセス、または書き込み禁止エリアに書き込みが生じると、ブレーク（ユーザプログラム停止）します。
[User VCC Threshold]	ユーザシステム電圧レベルを指定します。 ユーザ VCC が指定値よりも下がった場合、[拡張モニタ]ウィンドウの[User System Voltage]に[Down]が表示されます。
[User Signals]	このチェックボックスをチェックすると、ユーザシステムからのリセット信号を有効にします。
[Driver]	現在使用している E6000 ドライバを表示しています。
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

[Device]オプションで選択可能なデバイスおよびデバイスに依存したオプションは以下の通りです。

拡張ハードウェア欄に記載があるデバイスのエミュレーションを行う場合は必ず該当の拡張ハードウェアを接続してください。

表8-2 H8/3827R E6000 Emulator CPU 300L デバッギングプラットフォーム環境

ハードウェア	[Device]オプション	[Clock]オプション	[SubClock]オプション	拡張ハードウェア
HS388REPI60H	H8/3822R H8/3823R H8/3824R H8/3825R H8/3826R H8/3827R	0.5MHz 2MHz 8MHz Target/2	38.4kHz 307.2kHz 32.768kHz Target	-

【注】 1. [Clock]オプションの Target/2、[SubClock]オプションの Target はユーザシステム接続時のみ選択可能です。

2. H8/3827S グループのデバイスのエミュレーションを行う場合は、[Device]オプションから H8/3827R グループのデバイスを選択してください。（H8/3827S の場合は H8/3827R を選択等）

表8-3 H8/3847R E6000 Emulator CPU 300L デバッギングプラットフォーム環境

ハードウェア	[Device]オプション	[Clock]オプション	[SubClock]オプション	拡張ハードウェア
HS388REPI60H	H8/3842R H8/3843R H8/3844R H8/3845R H8/3846R H8/3847R	0.5MHz 2MHz 8MHz Target/2	38.4kHz 307.2kHz 32.768kHz Target	-

【注】 1. [Clock]オプションの Target/2、[SubClock]オプションの Target はユーザシステム接続時のみ選択可能です。

2. H8/3847S グループのデバイスのエミュレーションを行う場合は、[Device]オプションから H8/3847R グループのデバイスを選択してください。（H8/3847S の場合は H8/3847R を選択等）

表8-4 H8/3867 E6000 Emulator CPU 300L デバッギングプラットフォーム環境

ハードウェア	[Device]オプション	[Clock]オプション	[SubClock]オプション	拡張ハードウェア
HS3880EPI60H	H8/3862 H8/3863 H8/3864 H8/3865 H8/3866 H8/3867	0.5MHz 1MHz 2MHz Target/2	38.4kHz 307.2kHz 32.768kHz Target	-
HS388REPI60H	H8/3822 H8/3823 H8/3824 H8/3825 H8/3826 H8/3827	0.5MHz 2MHz Target/2		

【注】 1. [Clock]オプションの Target/2、[SubClock]オプションの Target はユーザシステム接続時のみ選択可能です。

表8-5 H8/3887 E6000 Emulator CPU 300L デバッギングプラットフォーム環境

ハードウェア	[Device]オプション	[Clock]オプション	[SubClock]オプション	拡張ハードウェア
HS3880EPI60H	H8/3882 H8/3883 H8/3884 H8/3885 H8/3886 H8/3887	0.5MHz 1MHz 2MHz Target/2	38.4kHz 307.2kHz 32.768kHz Target	-
HS388REPI60H	H8/3842 H8/3843 H8/3844 H8/3845 H8/3846 H8/3847	0.5MHz 2MHz Target/2		

【注】 1. [Clock]オプションの Target/2、[SubClock]オプションの Target はユーザシステム接続時のみ選択可能です。

表8-6 H8/3937 E6000 Emulator CPU 300L デバッギングプラットフォーム環境

ハードウェア	[Device]オプション	[Clock]オプション	[SubClock]オプション	拡張ハードウェア
HS388REPI60H	H8/3935R H8/3936R H8/3937R H8/3935 H8/3936 H8/3937	2MHz Target/2	Target	HS3937EIO60H または HS3937REIO60H

【注】 1. [Clock]オプションの Target/2 を指定した場合は、拡張 I/O ボード（HS3937EIO60H または HS3937REIO60H）より 5MHz が供給されます。
2. [SubClock]オプションの Target はユーザシステム接続時のみ選択可能です。

8.1.4 Memory Mapping 機能

メモリマップは、リザーブエリア（アクセス禁止）を１バイト単位で設定することができます。各メモリブロックは、ユーザシステム上のメモリに指定でき、それぞれの場合で、リードライトアクセス、リードオンリアクセス、またはアクセス禁止を指定できます。

デバイス内蔵メモリ（Internal ROM、RAM、I/O）の属性変更はできません。

8.1.5 ステイタスウィンドウ

[表示 -> CPU -> ステイタス] を選択するか、[ステイタスの表示]ツールバーボタンをクリックすると[ステイタス]ウィンドウを表示します。[ステイタス]ウィンドウには、3 枚のシートがあります。

本エミュレータでは下記項目を表示します。

(1) Memory シート

[ステイタス]ウィンドウの[Memory]タブを選択すると表示されます。表示内容は以下となります。

表8-7 Memoryシート内容一覧

Item カラムの項目	Status カラムの内容
Target Device Configuration	デバイスのメモリ構成を表示
System Memory Resources	エミュレータハードウェアのメモリ資源を表示
Program Name	プログラムファイル名を表示

(2) Platform シート

[ステイタス]ウィンドウの[Platform]タブを選択すると表示されます。表示内容は以下となります。

表8-8 Platformシート内容一覧

Item カラムの項目	Status カラムの内容
Connected To:	エミュレータの品名を表示（使用しているドライバ名）
CPU	選択しているデバイス名を表示
Mode	選択しているモードを表示
Clock source	選択しているクロックを表示
Subclock source	選択しているサブクロックを表示
Run status	ユーザプログラムの実行状態を表示 Break ブレーク中 Running プログラム実行中
Cause of last break	プログラムのブレーク要因を表示 サブアクティブの状態でブレークした場合は、要因の後に"（SubActive）"を表示します。 Ready ユーザプログラム未実行（HEW 起動直後など） User Break ユーザによる停止 PC Break プログラムブレークポイントによる停止 On Chip Break B ハードウェア PC Break による停止 Complex Event System イベント検出システムによる停止 Stepping Completed ステップ実行の完了 Stepping Aborted ステップのユーザによる停止 ROM Write Access Break ROM にライトした Write-protect Access Break リードオンリメモリにライトした Unused Area Access Break アクセス禁止メモリにアクセスした Stack In I/O Area I/O エリアへのスタックが発生した Invalid breakpoint PC Break 以外のブレーク命令による停止
Event Time Count	イベント間タイマの測定結果を表示
Run Time Count	プログラムの総実行時間を表示

(3) Events シート

[ステイタス]ウィンドウの[Events]タブを選択すると表示されます。表示内容は以下となります。

表8-9 Eventsシート内容一覧

Item カラムの項目	Status カラムの内容
Resources	ブレークポイントの種類とリソースを表示

8.1.6　拡張モニタ機能

[表示 -> CPU -> 拡張モニタ] を選択するか、[拡張モニタ]ツールバーボタンをクリックすると[拡張モニタ]ウィンドウを表示します。

本エミュレータでは下記項目を表示します。

表8-10　拡張モニタウィンドウ内容一覧

Item カラムの項目	Value カラムの内容
User Reset	リセット端子の状態を表示
User System Voltage	ユーザ VCC が[Configuration Properties]ダイアログボックス-[General]ページの [User VCC Threshold]で指定している値を満たしているかを表示
User Cable	ユーザケーブルが接続されているかを表示
Running status	ユーザプログラム実行中は MCU 内部のアドレスバスの値および CPU ステータスを表示 ユーザプログラム停止中はユーザプログラムの停止要因を表示 Break = <ブレーク要因>　ブレーク要因を表示します。 Address = <アドレスバス値>　実行中のアドレスバスの値を表示します。 サブアクティブモードの時は、アドレスバス値の後に （SubActive）を表示します。 Status = <CPU のステータス>　実行中の CPU ステータスを表示します。 CPU　CPU 命令プリフェッチサイクル、 CPU データアクセスサイクル MEDIUM　中速モード SUB_ACTIVE　サブアクティブモード SLEEP　スリープモード SUB_SLEEP　サブスリープモード WATCH　ウォッチモード STANDBY　スタンバイモード
Target Mode	ユーザシステムから入力されるモードを表示

【注】 1.　本エミュレータでは[拡張モニタ]ウィンドウの更新間隔の設定および変更はできません。

8.1.7　バス状態およびエリア信号

本エミュレータで取得可能なバス状態およびエリア信号を下記に示します。

表8-11　本エミュレータで取得可能なバス状態信号

バス状態	トレース表示（Status）	詳細
CPU Prefetch	CPU_PREFETCH	CPU プリフェッチサイクル（高速モード）
CPU Data	CPU	CPU データアクセスサイクル（高速モード）
CPU Medium Prefetch	CPU_MEDIUM_PREFETCH	CPU プリフェッチサイクル（中速モード）
CPU Medium Data	CPU_MEDIUM_DATA	CPU データアクセスサイクル（中速モード）
Sub Active Prefetch	SUB_ACTIVE_PREFETCH	サブアクティブモードプリフェッチサイクル
Sub Active Data	SUB_ACTIVE_DATA	サブアクティブモードデータアクセスサイクル

表8-12　本エミュレータで取得可能なエリア信号

エリア	トレース表示（Area）	詳細
On-chip ROM	IN-ROM	内蔵 ROM
On-chip RAM	IN-RAM	内蔵 RAM
On-chip I/O	I/O	I/O
LCD RAM	LCD-RAM	LCDRAM 領域

【注】　バス状態およびエリア信号はイベントポイントの[Bus/Area]条件設定で使用します。
これらの信号はトレース情報としても取得可能です。
また、バス状態信号についてはトレース非取得条件設定（[Suppress]オプション）でも使用します。

8.1.8 モニタ機能

本エミュレータはバスモニタ回路を実装しておりません。
モニタ機能を使用する場合には、E6000 用バスモニタオプションボード（HS6000EBR01H）が必要です。

8.1.9 トリガポイント

本エミュレータはバスモニタ回路を実装しておりません。
E6000 用バスモニタオプションボード（HS6000EBR01H）を装着することによって、[イベントポイント]ウィンドウの[Trigger]シートで設定可能なトリガポイントが使用できます。

8.1.10 トレース情報

[表示 -> コード -> トレース] を選択するか、[トレース]ツールバーボタンをクリックすると[Trace]ウィンドウを表示します。
本エミュレータで取得可能なトレース情報、及び表示する項目は以下の通りです。

[PTR]	トレースバッファ内のサイクル番号。 最後に取得されたサイクルの番号を 0 とし、古いサイクルにさかのぼって、順に-1、-2 と番号が小さくなります。 ディレイカウントが設定されている場合は、トレース停止条件が成立したサイクル番号を 0 とし、成立後停止するまでに実行されたサイクル（ディレイ期間中のサイクル）には、最後に取得されたサイクルに向かって順に+1、+2 と番号が大きくなります。
[Address]	アドレスバスの値（6 桁の 16 進数）。
[Instruction]	実行された命令の逆アセンブルコード。
[Data]	データバスの値。 それぞれ 16 進の 2 桁、4 桁で表示します。
[R/W]	アクセスサイクルの種類。読み出しは RD、書き込みは WR と表示します。
[Area]	アクセスしたエリア: 内蔵 ROM, 内蔵 RAM, I/O, LCDRAM （タイムスタンプ取得時は無効）
[Status]	バス状態: CPU（CPU データアクセスサイクル）, CPU_PREFETCH（プリフェッチ）, SUB_ACTIVE_DATA（サブアクティブデータアクセスサイクル）, SUB_ACTIVE_PREFETCH（サブアクティブプリフェッチ）, CPU_MEDIUM_DATA（中速データアクセスサイクル）, CPU_MEDIUM_PREFETCH（中速プリフェッチ） （タイムスタンプ取得時は無効）
[Probes]	4 本のプローブ信号の状態（2 進数 4 桁）。 左から Probe4、Probe3、Probe2、Probe1 の順で表示します。 （タイムスタンプ取得時は無効）
[Timestamp]	バスサイクルのタイムスタンプ。 タイムスタンプは、ユーザプログラム実行を開始するたびに０からカウントを始めます。測定時間の分解能は、Trace Acquisition で設定してください。 （タイムスタンプ取得時のみ有効）
[Source]	ソースコードの表示。
[Label]	アドレスに対応するラベル（ラベルが設定されている場合のみ表示します）。
[Timestamp-Difference]	前の行とのタイムスタンプの差分時間を表示。 (タイムスタンプ取得時のみ有効)

8.1.11　トレースレコードの検索

本エミュレータ使用時、[Trace Find]ダイアログボックスは下記ページより構成されます。

表8-13　[Trace Find]**ダイアログボックスのページ構成**

ページ	設定項目
[General]	**検索範囲を指定します。**
[Address]	**アドレス条件を指定します。**
[Data]	**データ条件を指定します。**
[R/W]	**アクセスサイクルの種類を指定します。**
[Area]	**アクセスしたエリアを指定します。** **（タイムスタンプ取得時は無効）**
[Status]	**バス状態を指定します。** **（ タイムスタンプ取得時は無効）**
[Probes]	4 **本のプローブ信号の状態を指定します。** **（タイムスタンプ取得時は無効）**
[Timestamp]	**バスサイクルのタイムスタンプを指定します。** **（タイムスタンプ取得時のみ有効）**

8.1.12　Trace Filter **機能**

本エミュレータ使用時、[Trace Filter]ダイアログボックスは下記ページより構成されます。

表8-14　[Trace Filter]**ダイアログボックスのページ構成**

ページ	設定項目
[General]	**フィルタ範囲を指定します。**
[Address]	**アドレス条件を指定します。**
[Data]	**データ条件を指定します。**
[R/W]	**アクセスサイクルの種類を指定します。**
[Area]	**アクセスしたエリアを指定します。** **（タイムスタンプ取得時は無効）**
[Status]	**バス状態を指定します。** **（ タイムスタンプ取得時は無効）**
[Probes]	4 **本のプローブ信号の状態を指定します。** **（タイムスタンプ取得時は無効）**
[Timestamp]	**バスサイクルのタイムスタンプを指定します。** **（タイムスタンプ取得時のみ有効）**

8.1.13　パフォーマンス解析ウィンドウ

H8/388R には下記に示す状態のための性能分析範囲があり、削除することはできません。

Standby、Watch、Medium、Sub Sleep、Sleep、Sub Active、CPU、Other Addresses

8.2 H8/388R E6000 エミュレータ使用上の注意事項

本エミュレータを使用するにあたり以下の注意事項があります。

8.2.1 チュートリアルプログラムの実行環境

チュートリアルプログラムを実行する場合は、以下のディレクトリに格納されている「Tutorial.hws」を指定して下さい。

HEW インストール先ディレクトリ\Tools\Renesas\DebugComp\Platform\E6000\3880R\Tutorial

8.2.2 サブクロック動作

ターゲット MCU である H8/3887 グループ、H8/3847 グループ、H8/3867 グループ、H8/3827 グループ、H8/3847R グループ、H8/3827R グループは、内部サブクロック(φw)として、307.2kHz を選択することはできません。

本エミュレータでサブクロック(32.768kHz および 38.4kHz)動作中にブレークが検出された場合、本クロックにて動作、表示反応してしまうため、サブクロック評価時は、38.4kHz の 8 倍周波数 307.2kHz を選択していただき評価することをお勧めします。

付録 A　　トラブルシューティング

? “ビルド中止”ボタンをクリックするか [ビルド->ビルドの中止]を選んでビルドを中止しようとしたが、ビルドの実行が中止できない

HEWは現在のファイルをビルド実行（または現在のフェーズを実行）後、ビルド実行を中止します。もしも、プロジェクトビルダから長い間応答がない場合、 [ビルド->ツールの終了] を選んでください。現在の処理を強制終了します。強制終了したツールから出力されたファイルは無効です。それらの出力ファイルを削除してフェーズを再実行してください。

? エディタにテキストファイルが表示されているが、シンタックス色付けが表示されない

ファイルに名前が付いている（保存した）ことを確認してください。また、[ツール->オプション...]を選んで“オプション” ダイアログボックスを開き、“エディタ”タブの“シンタックスカラーリング” チェックボックスがチェックされていることを確認してください。HEW ではファイルの拡張子の属するファイルグループを調べてファイルを色付けするかどうか判断します。現在定義されている拡張子とファイルグループを参照するには、 [プロジェクト ->ファイルの拡張子...]を選んで“ファイル拡張子”ダイアログボックスを表示してください。色付け情報を確認するには、[ツール-> 表示形式...]を選んで“表示形式” ダイアログボックスの“カラー” タブを参照してください。詳細は、HEW編の4章, 「エディタの使用」のシンタックスの色付けに関する説明をお読みください。

? ツールの設定を変えたいが、[ツール->アドミニストレーション…] メニューオプションを選べない

ワークスペースを開いている間は[ツール->アドミニストレーション...] を選ぶことはできません。“ツールアドミニストレーション”ダイアログボックスを開くには、現在のワークスペースを閉じてください。

? 日本語版 Windows® 98SE/Me 、Windows NT®、Windows® 2000 上で日本語入力ができない、または日本語の文字が正しく表示されない

[ツール->表示形式…] を選んで“フォント”タブをクリックして、“フォント”フィールドで日本語のフォントを選んでください。

? プロジェクトファイルの編集をしていないのに、[ビルド-> ビルド] を選んだらいくつかのファイルが再ビルドされた

ファイルは以下のいずれかの条件で再ビルドされます。

- ビルド後に現在のファイルのオプションが変更されたとき
- 出力ファイルのいずれかがないとき
- いずれかのソースファイルの日付がそのフェーズの出力ファイルの日付より新しいとき
- いずれかの依存ファイルの日付がそのフェーズの出力ファイルの日付より新しいとき
- ユーザ定義のビルドフェーズで、 “実行前に入力ファイルが存在するか否かのチェックを行わない” チェックボックスがチェックされているとき。このチェックボックスを参照するには、 [オプション-> ビルドフェーズ…]を選び、 “ビルド順序” タブの“ビルドフェーズの順序” リストからフェーズを選び、“変更…” ボタンをクリックしてください。すると、 “コマンド” タブで “実行前に入力ファイルが存在するか否かのチェックを行わない”チェックボックスを参照できます。
- 最適化リンケージエディタでサブコマンドファイルが選択されているとき

? プロジェクトのファイルを一時的にビルドから除外したい。

ワークスペースウィンドウの“Projects”タブのファイル上でマウスの右ボタンを押下し、[ビルドから除外 <file>]を選択してください。すると、そのファイルがビルドから除外されます。再びファイルをビルドに戻すには、ワークスペースウィンドウの“Projects”タブの当該ファイル上でマウスの右ボタンを押下し、[ビルドから除外の解除 <file>]を選択してください。

? 自分の PC でワークスペースを開いた。同時に、他の人が他の PC から同じワークスペースを開いた。自分でワークスペースの設定を変えて保存した。その後、他の人がワークスペースの設定を変えて保存した。自分が再びワークスペースを開くと、設定が自分の行った設定とは異なっていた。

最後に保存した設定が有効です。HEWはワークスペースを開くとメモリ内で更新します。ユーザが意識的に設定をファイルに保存しない限り、設定はファイルに保存されません。

? プロジェクトにファイルを追加するとき、ファイルに自動的にデフォルトオプションを指定したい

フェーズが複数ビルドフェーズであれば、フェーズにデフォルトオプションを指定できます。 [オプション] メニューからフェーズを選んでください。フェーズが複数ビルドフェーズの場合、オプションダイアログボックスの左側にファイルのリストが表示されます（図A-1）。 ファイルリストで、デフォルトオプションを指定したいファイルグループのフォルダを開いてください。 フォルダ内に“Default Options” アイコンが表示されます。アイコンを選択して、オプションダイアログボックスの右側でオプションを指定して“OK ”ボタンをクリックしてください。このオプションは、プロジェクトにそのファイルグループのファイルを初めて追加するときに適用されます。

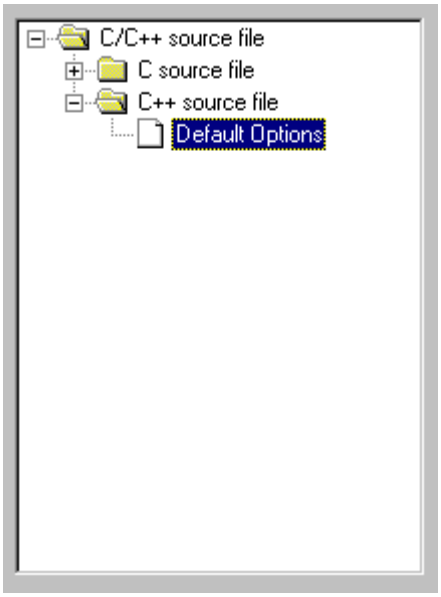


図 A-1: **オプションダイアログボックス、ファイルリスト**

? [ビルド>ビルド]を選択するとワークスペースウィンドウの “Projects”タブのファイル依存関係が更新されるが、[ビルド-> すべてをビルド]を選択してもファイル依存関係が更新されない。

[ビルド->すべてをビルド]はファイルの依存関係を更新しません。ファイル保存関係を更新するには[ビルド->すべての依存関係を更新]を選択してください。

付録 B　　正規表現

HEW エディタでは検索・置換操作の際、文字列に特殊文字を指定できます。指定できる特殊文字を表 B-1 に示し、その詳細を以下に示します。

表 B-1: 正規表現の文字

文字	意味
+	1 個以上の指定文字（ブラケット表現を除く）に一致します。例えば、a+は a、aa、aaa などに一致します。
*	0 個以上の指定文字（ブラケット表現を除く）に一致します。例えば、a*は空白文字列、a、aa などに一致します。
?	0 個以上の指定文字（ブラケット表現を除く）に一致します。例えば、a?は空白文字列と a に一致します。
{ }	{m,n}の形式で濃度の範囲を指定します。この構文は、m 個から n 個の指定文字に一致します。例えば、a{2,3}は aa と aaa に一致します。 この構文はまた、{m.}や{m}を使用して構成することもできます。1 つ目は m 個以上の指定文字に一致します。例えば、a{2.}は aa、aaa、aaaa などに一致します。2 つ目はちょうど m 個の指定文字に一致します。例えば、a{2}は aa に一致します。
[ ]	ブラケット表現を作成します。ブラケット表現は、どれもが一致するような文字セットを作成します。例えば、[abc]は a、b、または c に一致します。 ブラケット表現では、以下を除き、あらゆる正規表現の特殊文字は通常の文字として扱われます。 - は、ビットパターンに基づき文字値の範囲を指定します。例えば、[A-Za-z]はすべての大文字および小文字の英文字に一致します。- をブラケット表現の文字として表すには、文字セットの最初または最後の文字でなくてはなりません（例：[-a-z]、[A-Z-]）。 ^ は、かっこ内の最初の文字にあるときだけ特別です。^ を最初の位置に置くと、文字セットを補って一致させます。例えば、[^a-z]は小文字の英文字を除くすべての文字に一致します。 かっこ内の文字として] を含めるには、文字セットの最初の文字として含めなくてはなりません（例：[]abc）または[^]abc））。
( )	正規表現を、一単位として扱うことのできる部分正規表現に分類します。例えば、ab*は a、ab、abb などに一致しますが、(ab)*は空白文字列、ab、abab などに一致します。（ と ）は、ブラケット表現の中では特殊文字として扱われません。
\	正規表現文字を無視し、通常の文字として扱うようにします。例えば、(ab)は ab から成る部分正規表現を表しますが、\(ab\)は連続した文字（ 、a、b、 ）を示します。 注意：C++ソースプログラムで\の文字を指定する場合、\\と指定する必要があります。これは、C++コンパイラが\を C++ソースプログラムに実装したエスケープシーケンスの始まりを示す特殊文字として扱うためです。ただし、データファイルまたはダイアログボックスのテキストコントロールでは、二重のバックスラッシュは必要ありません。
^	入力文字列の先頭に正規表現または部分正規表現があることを示します。例えば、^ab は ab と abc に一致しますが、cab には一致しません。^ の扱いがブラケット表現では異なることに注意してください。
$	入力文字列の最後に正規表現または部分正規表現があることを示します。例えば、ab$は ab と cab に一致しますが、abc には一致しません。
|	同等に有効な選択肢である表現または部分表現の交代、または作成を示し、それぞれが一致します。例えば、ab|cd は ab または cd に一致します。
.	行の論理の終わりを示すものを除き、どのようなコード単位にも一致します。

付録 C　　プレースホルダ

プレースホルダは、HEW の複数のツールによって提供される機能です。この章ではプレースホルダの使い方を説明します。

C.1　プレースホルダとは?

プレースホルダとは一時的にテキストに挿入される特殊文字列です。後に実際の値に置き換えます。例えば、HEW のプレースホルダのひとつに、$(FULLFILE)があります。これは、すべてのパス付きのファイルを示します。パラメータとしてファイルを編集できるエディタが c: \myedit\myeditor.exe であるとします。c: \files ディレクトリのファイル FILE1.C を開き、このエディタを起動するには、以下のように直接指定することもできます。

```
c: \myedit\myeditor.exe c: \files\file1.c
```

しかし、このエディタを介して任意のファイルを開きたいとき、上記コマンドは“c:\files\file1.c”を開くだけのものであるため、問題が起きます。指定したエディタを使うときにその時点で選んだファイルを開くことができるようにするには、特定のファイル名を一般的なプレースホルダに置き換えます。以下に例を示します。

```
c:\myedit\myeditor.exe $(FULLFILE)
```

これで、HEW がエディタでファイルを開くとき、$(FULLFILE)を選ばれたファイル名で置き換えます。

C.2　プレースホルダを挿入する

プレースホルダは HEW の 3 つの編集フィールドで指定できます (図 C-1 、C-2、C-3)。プレースホルダの指定方法には 4 種類あります。

例 1

プレースホルダを挿入したい場所にカーソルを置いてください。次に、必要なプレースホルダを編集フィールドの右のポップアップメニューから選んでください。

図 C-1: **プレースホルダポップアップメニュー**

例 2

“Custom directory”以外のプレースホルダをドロップダウンリストボックスから選び、プレースホルダによって示されたディレクトリからの相対サブディレクトリを指定してください。 “Custom directory”を選んだ場合、“Sub-Directory”フィールドに絶対ディレクトリパスを指定してください。

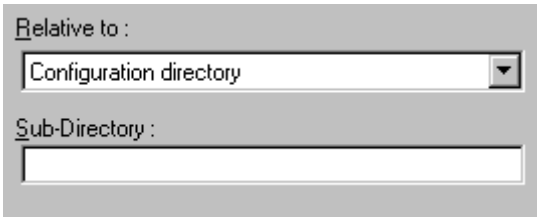


図 C-2: **プレースホルダドロップダウンリストと** Sub-Directory **フィールド**

例 3

プレースホルダを挿入したい場所にカーソルを置いてください。次に、必要なプレースホルダをドロップダウンリストボックスから選んでください。そして、“Insert”ボタンをクリックしてください。

図 C-3: **プレースホルダドロップダウンリストボックス**

例 4

フィールドにプレースホルダを直接入力してください。大文字で入力して、“$(” で始めて“) ”で終わってください。

正しい

```
$(FILEDIR)
```

誤り

```
$(Filedir)
$(  FILEDIR   )
$FILEDIR
```

C.3　使用できるプレースホルダ

表 C-1 にプレースホルダと意味を示します。

表 C-1: プレースホルダ

プレースホルダ	意味
$(FULLFILE)	ファイル名（フルパスを含む）
$(FILEDIR)	ファイルディレクトリ
$(FILENAME)	ファイル名（パスを除き拡張子を含む）
$(FILELEAF)	ファイル名（パスと拡張子を除く）
$(EXTENSION)	ファイルの拡張子
$(WORKSPDIR)	ワークスペースディレクトリ
$(WORKSPNAME)	ワークスペース名
$(PROJDIR)	プロジェクトディレクトリ
$(PROJECTNAME)	プロジェクト名
$(CONFIGDIR)	コンフィグレーションディレクトリ
$(CONFIGNAME)	コンフィグレーション名
$(HEWDIR)	HEW インストールディレクトリ
$(TCINSTALL)	ツールチェインインストールディレクトリ（オプションダイアログ上）
$(TOOLDIR)	ツールインストールディレクトリ（Tools Administration 上）
$(TEMPDIR)	テンポラリディレクトリ
$(WINDIR)	Windows® ディレクトリ
$(WINSYSDIR)	Windows® システムディレクトリ
$(EXEDIR)	コマンドディレクトリ
$(USERNAME)	ユーザログイン（バージョン管理）
$(PASSWORD)	ユーザパスワード（バージョン管理）
$(VCDIR)	「仮想的」バージョン管理ディレクトリ
$(COMMENT)	コメント（バージョン管理）
$(LINE)	エラー/ウォーニングの行番号

プレースホルダの使用例を以下に示します。

表 C-2: **プレースホルダの展開（例）**

プレースホルダ	プレースホルダの展開例
$(FULLFILE)	c:\hew\workspace\project\file.src
$(FILEDIR)	c:\hew\workspace\project
$(FILENAME)	file.src
$(FILELEAF)	file
$(EXTENSION)	src
$(WORKSPDIR)	c:\hew\workspace
$(WORKSPNAME)	workspace
$(PROJDIR)	c:\hew\workspace\project
$(PROJECTNAME)	project
$(CONFIGDIR)	c:\hew\workspace\project\debug
$(CONFIGNAME)	debug
$(HEWDIR)	c:\hew
$(TCINSTALL)	c:\hew\toolchains\renesas\sh\511
$(TOOLDIR)	c:\hew\toolchains\renesas\sh\511
$(TEMPDIR)	c:\Temp
$(WINDIR)	c:\Windows
$(WINSYSDIR)	c:\Windows\System
$(EXEDIR)	v:\vc\win32
$(USERNAME)	JHARK
$(PASSWORD)	214436
$(VCDIR)	“c:\project” **は** “x:\vc\project” **へマッピングされている**
$(COMMENT)	“Please Enter Comment” **ダイアログボックスが表示される**
$(LINE)	12

表 C-2 では以下のことを仮定しています。

- ファイルパスは “c:\hew\workspace\project\file.src”
- ワークスペース名 “workspace”の位置は “c:\hew\workspace”
- プロジェクト名 “project” の位置は “c:\hew\workspace\project”
- コンフィグレーション名 “debug” にはコンフィグレーションディレクトリがあり、位置は “c:\hew\workspace\project\debug”
- HEW.EXE が“c:\hew”にインストールされている
- ツールチェイン（コンパイラ、アセンブラ、リンケージエディタ）の*.HRF ファイル の位置は “c:\hew\toolchain\renesas\sh\511”
  このディレクトリは[オプション]メニューのオプション設定ダイアログボックス上では $(TCTINSTALL)として参照され、“ツールアドミニストレーション”ダイアログボックス上では $(TOOLDIR)として参照される
- Windows® オペレーティングシステムが c:\Windows にインストールされており、Windows® システムファイルが c:\Windows\System にインストールされている
- バージョン管理実行可能パスが v:\vc\win32\ss.exe である。バージョン管理システムにログインするユーザ名は JHARK でパスワードが 214436 であり、バージョン管理実行可能ファイルへのコマンドラインには$(COMMENT)が指定されている。c:\project は[ツール->バージョン管理->構成...]で選ぶと表示される“Version Control Setup”ダイアログボックスの“Projects”タブの x:\vc\project にマッピングされている
- コンパイラまたはアセンブラのエラーが 12 行目で起きた

注意　**どのフィールドでもすべてのプレースホルダを使用できるとは限りません。例えば、プレースホルダ** $(LINE) **は依存ファイル位置を指定するときには意味をもちません。プレースホルダ**$(USERNAME)**、** $(PASSWORD)**、**$(VCDIR)**、**$(COMMENT)**はバージョン管理でのみ受け付けられます。各編集フィールドで、使用できないプレースホルダを指定すると、警告メッセージが表示される場合があります。**

C.4　プレースホルダを使うにあたって

プレースホルダによって、システムが使用する様々なファイルへのパスを任意に指定できます。

- パスまたはファイル名を入力する編集フィールドのとなりにプレースホルダのポップアップメニュー( ) がある場合、プレースホルダをどのように使ってパスやファイルの指定をフレキシブルにできるかご考慮ください。
- いくつかの構成を使うとき、プレースホルダ$(CONFIGDIR) を使うと、現在の構成のディレクトリからファイルへの書き込みやそのファイルから現在の構成のディレクトリへの書き込みができるので、便利です。
- できるだけプレースホルダを利用してください。プレースホルダは後で削除したり追加したりできるので、気軽に試すことができます。

付録 D　I/O ファイルフォーマット

HEWは、I/Oレジスタ定義ファイルで取得する情報に基づいて、[IO]ウィンドウをフォーマットします。デバッギングプラットフォームを選択すると、HEWは、選択したデバイスに対応する“<device>.IO”ファイルを検索し、存在する場合にはこのファイルをロードします。これは、I/Oモジュール、およびそのレジスタのアドレスやサイズを記述するフォーマット済みテキストファイルです。ユーザはテキストエディタでこのファイルを編集し、ユーザアプリケーションに特有のメモリマップレジスタや周辺レジスタ(例えば、マイコンのアドレス空間にマップしたASICデバイスのレジスタ)のサポートを追加することができます。“<device>.IO”ファイルには、ビットフィールド対応／非対応により二種類のフォーマットがあります。それぞれのフォーマットについて説明します。

D.1　ファイルフォーマット(ビットフィールド非対応)

各モジュール名を[Modules]定義セクションで定義し、モジュールの番号を、順番に付けていなければなりません。各モジュールはレジスタ定義セクションに対応しており、セクション内のエントリは、I/Oレジスタを定義します。

“BaseAddress”はデバイスのための定義であり、そのデバイスでは、CPUモードによってアドレス空間のI/Oレジスタの場所が移動します。この場合、”BaseAddress”値は、ある特有モードのI/Oレジスタのベースアドレスです。また、レジスタ定義で使用するアドレスは、同じモードにおけるレジスタのアドレス位置です。I/Oレジスタファイルを実際に使用する場合、定義したレジスタアドレスから”BaseAddress”値を引き、その結果のオフセットを選択したモードのベースアドレスに加算します。

[Register]定義エントリは、<name>= <address> [<size>[<absolute>]]のフォーマットで入力します。

1. <name>は表示するレジスタ名です。
2. <address>はレジスタのアドレスです。
3. <size>は、Bがバイトサイズ、Wがワードサイズ、Lがロングワードサイズを意味します(デフォルトはバイトです)。
4. <absolute>は、レジスタが絶対アドレスにある場合、Aと設定します。これは、異なるモードのCPUによってI/O空間アドレス範囲が移動する場合のみ関連します。レジスタが絶対アドレスにあると定義すると、ベースアドレスオフセットは計算せず、指定したアドレスを直接使用します。

コメント行を入れる場合、“;”で始めなければなりません。

次に例を示します。

例:

コメント

```
; H8S/2655 Series I/O Register Definitions File
```

モジュールの定義

```
[Modules]
BaseAddress=0
Module1=Power_Down_Mode_Registers
Module2=DMA_Channel_Common
Module3=DMA_Channel_0
...
Module42=Bus_Controller
Module43=System_Control
Module44=Interrupt_Controller

...
```

レジスタの定義

```
[DMA_Channel_Common]
DMAWER=0xffff00 B A
DMATCR=0xffff01 B A
DMACR0A=0xffff02 B A
DMACR0B=0xffff03 B A
DMACR1A=0xffff04 B A
DMACR1B=0xffff05 B A
DMABCRH=0xffff06 B A
DMABCRL=0Xffff07 B A

...

[DMA_Channel_0]
MAR0AH=0xfffee0 W A
MAR0AL=0xfffee2 W A
IOAR0A=0xfffee4 W A
ETCR0A=0xfffee6 W A
MAR0BH=0xfffee8 W A
MAR0BL=0xfffeea W A
IOAR0B=0xfffeec W A
ETCR0B=0xfffeee W A
```

レジスタ名

アドレス

サイズ

絶対アドレスラグ

D.2　ファイルフォーマット(ビットフィールド対応)

各モジュール名を[Modules]定義セクションで定義し、モジュールの番号を、順番に付けていなければなりません。各モジュールはレジスタ定義セクションに対応しており、セクション内のエントリは、I/Oレジスタを定義します。

このセクションの最初で、”FileVersion=2”と宣言する必要があります。これは、このI/Oレジスタファイルがビットフィールド対応のバージョンで記述されていることを示します。

“BaseAddress”はデバイスのための定義であり、そのデバイスでは、CPUモードによってアドレス空間のI/Oレジスタの場所が移動します。この場合、”BaseAddress”値は、ある特有モードのI/Oレジスタのベースアドレスです。また、レジスタ定義で使用するアドレスは、同じモードにおけるレジスタのアドレス位置です。I/Oレジスタファイルを実際に使用する場合、定義したレジスタアドレスから”BaseAddress”値を引き、その結果のオフセットを選択したモードのベースアドレスに加算します。

各モジュールにはセクションがあり、オプションの依存性によって形成するレジスタを定義します。依存性は、モジュールがイネーブルかどうかを確認するためにチェックします。各レジスタ名をセクションで定義し、レジスタの番号を、順番に付けていなければなりません。依存性は、dep=<reg> <bit> <value>のようにセクションに入力します。

1. <reg>は依存性のレジスタ ID です。
2. <bit>はレジスタのビット位置です。
3. <value>は値で、ビットは、イネーブルであるモジュールに使用しなければなりません。

[Register]定義エントリは、id=<name> <address> [<size>[<absolute>[<format>[<bitfields>]]]]のフォーマットで入力します。

1. <name>は表示するレジスタ名です。
2. <address>はレジスタのアドレスです。
3. <size>は、B がバイトサイズ、W がワードサイズ、L がロングワードサイズを意味します(デフォルトはバイトです)。
4. <absolute>は、レジスタが絶対アドレスにある場合、A と設定します。これは、異なるモードの CPU によって I/O 空間アドレス範囲が移動する場合のみ関連します。レジスタが絶対アドレスにあると定義すると、ベースアドレスオフセットは計算せず、指定したアドレスを直接使用します。
5. <format>はレジスタを出力するためのフォーマットです。有効な値は、16 進数の場合は H、10 進数は D、2 進数は B です。
6. <bitfields>セクションは、レジスタのビットを定義します。

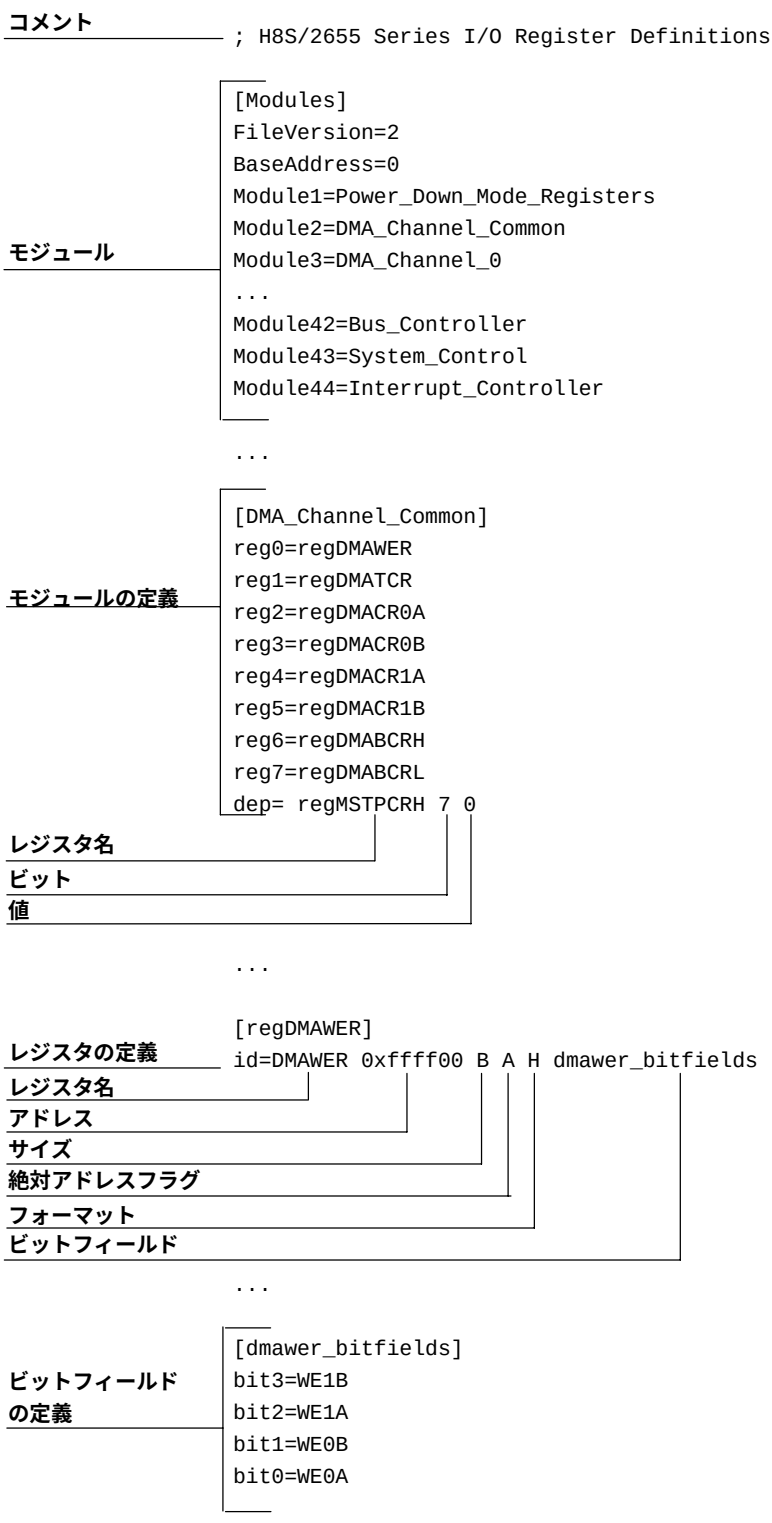
ビットフィールドセクションは、各エントリがbit<no>=<name>タイプのレジスタ内のビットを定義します。

1. <no>はビット番号です。
2. <name>はビットのシンボル名です。

コメント行を入れる場合、“;”で始めなければなりません。

次に例を示します。

例:


コメント
; H8S/2655 Series I/O Register Definitions File
[Modules]
FileVersion=2
BaseAddress=0
Module1=Power_Down_Mode_Registers
Module2=DMA_Channel_Common
Module3=DMA_Channel_0
...
Module42=Bus_Controller
Module43=System_Control
Module44=Interrupt_Controller
モジュール
...
[DMA_Channel_Common]
reg0=regDMAWER
reg1=regDMATCR
reg2=regDMACR0A
reg3=regDMACR0B
reg4=regDMACR1A
reg5=regDMACR1B
reg6=regDMABCRH
reg7=regDMABCRL
dep= regMSTPCRH 7 0
モジュールの定義
レジスタ名
ビット
値
...
[regDMAWER]
id=DMAWER 0xffff00 B A H dmawer_bitfields
レジスタの定義
レジスタ名
アドレス
サイズ
絶対アドレスフラグ
フォーマット
ビットフィールド
...
[dmawer_bitfields]
bit3=WE1B
bit2=WE1A
bit1=WE0B
bit0=WE0A
ビットフィールド
の定義

付録 E　シンボルファイルフォーマット

HEWを理解し、シンボルファイルを正確にデコードするためには、ファイルをPentica-Bファイルとしてフォーマットしなければなりません。

1. ファイルは、簡単なASCIIテキストファイルでなければなりません。
2. ファイルは、ワード[BEGIN]で始めなければなりません。
3. 各シンボルは、個々の行で、まず、“H”で終了する16進数の値から始まり、次にスペース、シンボルテキストの順でなければなりません。
4. ファイルは、ワード[END]で終了しなければなりません。

例：

```
BEGIN
11FAH Symbol_name_1
11FCH Symbol_name_2
11FEH Symbol_name_3
1200H Symbol_name_4
END
```

付録 F　HMAKE ユーザガイド

F.1　コマンドライン

次の節では、用意されたオプションの使用または不使用にかかわらず、ファイルで hmake プログラムを実行するときに使うコマンドラインを説明します。

F.1.1　基本構成

コマンドラインのシンタックスを以下に示します。

hmake <メイクファイル> <パラメータリスト>

拡張子なしでファイルを指定した場合、.mak が付加されます。パラメータリストには、次の節に示す複数のパラメータを指定できます。また、パラメータは省略できます。パラメータリストはメイクファイル名の前に指定してもかまいません。各パラメータは、1 つ以上の空白文字で区切ってください。パラメータは、大文字と小文字を区別しません。パラメータとファイル名を指定しなかった場合、ヘルプ情報が表示されます。

F.1.2　Exit コード

実行中のメイクファイルでシンタックスエラー発生時、またはメイクファイル実行中の処理で不当エラーコードが返されたとき、hmake はコード 1 で終了します。その他の場合、hmake はコード 0 で終了します（ファイルのシンタックスや exit コード条件の指定方法は、以下を参照してください）。

F.1.3　パラメータ

表 F.1 に指定できるパラメータとその機能を示します。

表 F.1 hmake のパラメータとその機能

パラメータ	機能
/A	入出力ファイル状態にかかわらず全コマンドを実行する。全ビルドと同じ
/N	入出力ファイルの使用状態により、（ノーマルで）実行すべきコマンドを判断して、表示するが、実行はしない
/?	ヘルプ情報の表示

F.2　ファイルのシンタックス

hmake ファイルでは、基本的に 4 種類の文（変数宣言、説明部分、コメント、メッセージコメント）が使用されます。hmake ファイルの作成にあたって、これらの文はどのような順序でも使用できます。ただし、変数を説明部分や他の変数宣言に使用する前に、変数宣言をしてください。nmake ファイルで使用する最初の “ all ” 文は、hmake ファイルでは必要ありません。メイクファイルでは、コマンドは出現する順序で実行されます。 “ → ” 文字は、メイクファイルのシンタックスを正しくするためにタブ文字を使用すべきところを示すことに注意してください。

F.2.1　変数宣言

変数宣言をした後、その変数は、hmake ファイルのすべての部分ですべての文で使うことができます。変数宣言のシンタックスを以下に示します。

```
<変数名> = <値>
```

変数名と ‘ = ’ 符号の間や、値と ‘ = ’ 符号の間に、複数の空白文字を入れることができます。値は、‘ \ ’ 文字で区切って、数行にわたって書くことができます。メインテキスト内で値に ‘ \ ’ 文字を含む場合は文字どおり解釈されます。‘ \ ’ 文字の後に新しい行が続くときのみ、値が次の行にわたっているとみなされます。変数宣言の例をいくつか示します。

```
EXECUTABLE = c:\dir\prog.exe
OUTPUT = c:\dir2\file1.out
INPUT = c:\dir2\file1.c
DEPEND = c:\dir2\file2.h \
         c:\dir2\file3.h \
         c:\dir2\file4.h
```

hmake ファイルの後半で変数を使うには、変数名の前に “ $(” を、後ろに “) ” を加えてください。 “ $() ” 文字付きの変数名は、変数の値で置き換えられます。次の説明部分の節にその例を示します。変数名には、半角英数文字と半角下線を使用することができます。別の変数宣言のなかで変数を使うことができますが、すべての変数は、使う前に宣言しなければなりません。

F.3　記述部

F.3.1　概要

説明部分には、1 つ以上のターゲット、0 以上の依存ファイル、そして最新のターゲットが最新の依存ファイルより新しい場合に実行するコマンド一覧を指定します。ターゲット、依存ファイルのどちらか、またはいずれもない場合、コマンドは常に実行されます。コマンドを常に実行したいときには、依存ファイルを指定する必要はありません。説明部分のシンタックスを以下に示します。

```
<ターゲット1> <ターゲット2> … : <依存ファイル1> <依存ファイル2> …
→<コマンド1>
→<コマンド2>
→…
→<コマンドn>
```

最後のターゲットと ‘ : ’ 文字の間や、最初の依存ファイルと ‘ : ’ 文字の間に、複数の空白文字を入れることができます。最初のターゲットの前に空白文字を入れないでください。各ターゲットと各依存ファイルは、少なくとも 1 つの空白文字で区切ってください。タブ文字はコマンドを含む行のはじめに必要です。変数宣言の後で、変数を上記シンタックスで説明部分に使うことができます。

記述部の例（変数宣言の後に変数を使う例）を以下に示します。

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→gcc c:\dir1\file1.c
$(OUTPUT) : $(INPUT) $(DEPEND)
→$(EXECUTABLE) $(INPUT)
```

F.3.2　特殊なコマンド

記述ブロックで使用できる特殊なコマンドが 2 つあります。cd コマンドはカレントディレクトリを変更し、set コマンドはメイクファイル実行の間に使用する環境変数を設定します。両方とも、相当する DOS コマンドと同じ方法で使用されます。

これらのコマンドを使用した有効な記述ブロックの例を、以下に示します。

```
CHANGEDIR :
→ cd c:\dir1\dir2
SETENV:
→ set VAR1=value1
→ set VAR2=value2
→ set VAR3=value3
```

$(EXECUTABLE)を置き換える CHANGEDIR と SETENV がファイル名でなくてもかまいません。これらは存在しないファイルとして扱われてコマンドは常に実行されます。

F.3.3　サブコマンドファイル

hmake でサブコマンドファイルを生成するには、説明部分のコマンド部分を以下のように指定してください(上記の<コマンド n>を置き換えます)。

```
→<コマンド開始> <<
→<サブコマンド1>,
→<サブコマンド2>,
→…
→<サブコマンドn>
→<<<コマンド終了>
```

これで、ウィンドウズの一時的ファイルに、<サブコマンド 1>、<サブコマンド 2>などの行を含むサブコマンドファイルが生成されます。このコマンドファイルはメイク処理が終了すると削除されます。コマンドファイル名は 2 つの”<<”の間のすべてのテキストで置き換えられます。サブコマンドファイル名は、自動的に hmake により付けられます。

例：

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→gcc @”<<
→-c -o c:\dir1\file1.obj c:\dir1\file1.c
<<”
```

生成されたサブコマンドファイル名が “ c:\temp\hmk111.cmd ” の場合、hmake は以下を実行します。(c:\dir1\file1.obj が更新されるとき)：

```
gcc @”c:\temp\hmk111.cmd”
```

サブコマンドファイル(c:\temp\hmk111.cmd)には以下を含みます。

```
-c -o c:\dir1\file1.obj c:\dir1\file1.c
```

記述部に 1 つ以上のコマンドを含むことができ、標準とサブコマンドファイルコマンドを混在して使用できます。

F.4　コメント

‘ # ’ 文字はコメントを示します。この文字が行の最初にあると、（次の改行文字があるまで）その行は無視されます。コメントの例を次に示します。

```
# My hmake file

# Variable declaration
OUTPUT= c:\dir1\file1.obj
# Descriptor
$(OUTPUT) : c:\dir1\file1.c c:\dir1\file1.h
→set VAR1=value1
→gcc c:\dir1\file1.c
```

hmake ファイルではコメントには専用の行が必要です。コメントを他の文の最後に付けることはできません。

F.5　メッセージコマンド

メッセージコマンドは、メイクファイル実行中にテキストの行を標準出力に出力します。これらのテキスト行は、適切に実行されている実行ファイルの出力のなかで、メイクファイルに出現する順序で出力されます。出力テキストのバッファリングは行いません。メッセージコマンドのシンタックスを以下に示します。

```
!MESSAGE <出力するテキスト>
```

改行文字が<出力するテキスト>の最後の文字の後にあると想定します。!MESSGE と<出力するテキスト>の間の空白文字は無視されます。メッセージコマンドの例を以下に示します。

```
!MESSAGE Executing C Compiler
```

付録 G　メニュー一覧

GUI メニューの一覧を表 F-1 に示します。

表 F-1　GUI メニュー一覧

メニュー	メニューオプション			ショートカットキー	ツールバーボタン	備考
表示	逆アセンブリ			Ctrl+D		[逆アセンブリ]ウィンドウを表示します
	コマンドライン			Ctrl+L		[コマンドライン]ウィンドウを表示します
	TCL ツールキット			Ctrl+Shift+L		[Console]ウィンドウを表示します
	ワークスペース			Alt+K		[ワークスペース]ウィンドウを表示します
	アウトプット			Alt+U		[Output]ウィンドウを表示します
	差分					[差分]ウィンドウを表示します
	CPU	レジスタ		Ctrl+R		[レジスタ]ウィンドウを表示します
		メモリ...		Ctrl+M		[メモリ]ウィンドウを表示します
		IO		Ctrl+I		[IO]ウィンドウを表示します
		ステイタス		Ctrl+U		[ステイタス]ウィンドウを表示します
		日本語メモリダンプ...		Shift+Ctrl+D		[日本語メモリダンプ]ウィンドウを表示します
		拡張モニタ				[拡張モニタ]ウィンドウを表示します
		モニタ	設定...	Shift+Ctrl+E		[モニタ]ウィンドウを表示します
			ウィンドウの選択...			[モニタ]ウィンドウの一覧表示、および追加／編集等を行う[ウィンドウの選択]ダイアログボックスを表示します
		EEPROM...				[EEPROM]ウィンドウを表示します
	シンボル	ラベル		Shift+Ctrl+A		[ラベル]ウィンドウを表示します
		ウォッチ		Ctrl+W		[ウォッチ]ウィンドウを表示します
		ローカル		Shift+Ctrl+W		[ローカル]ウィンドウを表示します
	コード	イベントポイント		Ctrl+E		[イベントポイント]ウィンドウを表示します
		トレース		Ctrl+T		[Trace]ウィンドウを表示します
		スタックトレース		Ctrl+K		[スタックトレース]ウィンドウを表示します
	グラフィック	画像...		Shift+Ctrl+G		[画像]ウィンドウを表示します
		波形...		Shift+Ctrl+V		[波形]ウィンドウを表示します
	パフォーマンス	パフォーマンス解析		Shift+Ctrl+P		[パフォーマンス解析]ウィンドウを表示します

メニュー	メニューオプション		ショートカットキー	ツールバーボタン	備考
オプション	デバッグセッション...				デバッグセッションの一覧表示、および追加／削除等が可能な[デバッグセッション]ダイアログボックスを表示します
	デバッグの設定...				デバッグ時の条件やダウンロードモジュール等の設定を行う[デバッグの設定]ダイアログボックスを表示します
	基数	16 進数			数値の表示／入力時の基数のデフォルト設定を 16 進数とします
		10 進数			数値の表示／入力時の基数のデフォルト設定を 10 進数とします
		8 進数			数値の表示／入力時の基数のデフォルト設定を 8 進数とします
		2 進数			数値の表示／入力時の基数のデフォルト設定を 2 進数とします
	エミュレータ	システム...			デバッグプラットフォームの設定を行う[Configuration Properties]ダイアログボックスを表示します
		メモリリソース...			デバッグプラットフォームのメモリマップの表示、および設定を行う[Memory Mapping]ダイアログボックスを表示します
デバッグ	CPU のリセット				ターゲットハードウェアをリセットし、PC をリセットベクタアドレスに設定します
	実行		F5		現在の PC からユーザプログラムを実行します
	リセット後実行		Shift+F5		ターゲットハードウェアをリセットし、リセットベクタアドレスからユーザプログラムを実行します
	カーソル位置まで実行				現在の PC からテキストカーソルの位置までユーザプログラムを実行します
	カーソル位置を PC 値に設定				テキストカーソルの位置に PC を設定します
	ラン...				実行時の PC や PC ブレークポイントの設定が可能な[プログラム実行]ダイアログボックスを表示します。
	PC 位置表示		Shift+Ctrl+Y		現在の PC がある[エディタ]または[逆アセンブリ]ウィンドウをオープンします
	ステップイン		F11		ユーザプログラムの１ブロックを実行して停止します
	ステップオーバ		F10		ユーザプログラムの１ブロックを実行して停止しますが、サブルーチンを呼び出す場合は、サブルーチンには入りません
	ステップアウト		Shift+F11		現在の関数の終わりに到達するまでユーザプログラムを実行します
	ステップ...				ステップ動作の設定が可能な[プログラムステップ]ダイアログボックスを表示します
	ステップモード	自動			[ソース]ウィンドウがアクティブの場合はソースライン一行だけをステップ実行します，[逆アセンブリ]ウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します
		アセンブリ			アセンブリ言語命令単位にステップ実行します
		ソース			ソースライン一行だけをステップ実行します

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考
デバッグ	プログラムの停止	Esc		ユーザプログラムの実行を停止します
	接続			デバッグプラットフォームを接続します
	初期化			デバッグプラットフォームを切断し、再接続します
	接続解除			デバッグプラットフォームを切断します
	ダウンロード			オブジェクトプログラムをロードします
	アンロード			オブジェクトプログラムをアンロードします
メモリ	検索...			指定したメモリ領域から指定した値を検索します
	コピー...			指定したメモリ領域を指定したアドレスにコピーします
	比較...			二つの指定したメモリ領域を比較します
	フィル...			指定したメモリ領域に指定した値を書き込みます
	テスト...			指定したメモリ領域のテストを行います
	最新の情報に更新			全ての[メモリ]ウィンドウの内容を強制的に更新します
	オーバレイの構成...			オーバレイ使用時の優先セクショングループの設定を行います

付録 H　コマンドライン一覧

コマンド一覧を**表** H-1 に示します。

表 H-1　**コマンド一覧**

項番	コマンド名	短縮形	説　明
1	!	-	**コメント**
2	ADD_FILE	AF	**カレントプロジェクトにファイルを追加**
3	ANALYSIS	AN	**性能分析機能の有効化／無効化**
4	ANALYSIS_RANGE	AR	**性能分析範囲の設定，表示**
5	ANALYSIS_RANGE_DELETE	AD	**性能分析範囲の削除**
6	ASSEMBLE	AS	**アセンブルの実行**
7	ASSERT	-	**コンディションのチェック**
8	AUTO_COMPLETE	AC	**オートコンプリート機能の有効化／無効化**
9	BREAKPOINT	BP	**実行命令位置によるブレークポイントの設定**
10	BREAKPOINT_CLEAR	BC	**ブレークポイントの削除**
11	BREAKPOINT_DISPLAY	BD	**ブレークポイント一覧の表示**
12	BREAKPOINT_ENABLE	BE	**ブレークポイントの有効／無効の切換え**
13	BREAKPOINT_SEQUENCE	BS	**実行順序を指定したブレークポイントの設定**
14	BUILD	BU	**カレントプロジェクトのビルド処理を開始**
15	BUILD_ALL	BL	**カレントプロジェクトのすべてのビルド処理を開始**
16	CHANGE_CONFIGURATION	CC	**コンフィギュレーションの設定**
17	CHANGE_PROJECT	CP	**プロジェクトの設定**
18	CHANGE_SESSION	CS	**セッションの設定**
19	CLOCK	CK	**エミュレータの** CPU **のクロック時間の設定**
20	CONFIGURE_PLATFORM	CPF	**エミュレータのデバッグ環境の設定**
21	DEFAULT_OBJECT_FORMAT	DO	**デフォルトオブジェクト（プログラム）フォーマットの設定**
22	DEVICE_TYPE	DE	**デバイスタイプの選択**
23	DISASSEMBLE	DA	**逆アセンブル表示**
24	ERASE	ER	[Command Line]**ウィンドウの内容のクリア**
25	EVALUATE	EV	**式の計算**
26	EXMONITOR_DISPLAY	EXMD	**拡張モニタの内容表示**
27	EXMONITOR_SET	EXMS	**拡張モニタ項目の表示／非表示の切換え**
28	EXMONITOR_SETRATE	EXMSR	**実行中およびブレーク中の拡張モニタ更新時間の設定**
29	FILE_LOAD	FL	**オブジェクト（プログラム）ファイルのロード**
30	FILE_SAVE	FS	**メモリ内容のファイルセーブ**
31	FILE_UNLOAD	FU	**ファイルのアンロード**
32	FILE_VERIFY	FV	**ファイル内容とメモリ内容の比較**
33	GENERATE_MAKE_FILE	GM	HEW **外でビルドするための** make **ファイルを生成**
34	GO	GO	**ユーザプログラムの実行**
35	GO_RESET	GR	**リセットベクタからのユーザプログラムの実行**
36	GO_TILL	GT	**テンポラリブレークポイントまでのユーザプログラムの実行**
37	HALT	HA	**ユーザプログラムの停止**
38	HELP	HE	**コマンドラインのヘルプ表示**
39	INITIALIZE	IN	**デバッグプラットフォームの初期化**
40	LOG	LO	**ロギングファイルの操作**
41	MAP_DISPLAY	MA	**メモリマッピング情報の表示**
42	MAP_SET	MS	**メモリマッピングの設定**

項番	コマンド名	短縮形	説　明
43	MEMORY_COMPARE	MC	メモリ内容の比較
44	MEMORY_DISPLAY	MD	メモリ内容の表示
45	MEMORY_EDIT	ME	メモリ内容の変更
46	MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
47	MEMORY_FIND	MI	メモリ範囲内でデータを検索
48	MEMORY_MOVE	MV	メモリブロックの移動
49	MEMORY_TEST	MT	メモリブロックのテスト
50	MODE	MO	CPU モードの設定，表示
51	MODULES	MU	内蔵周辺機能の設定，表示
52	MONITOR_CLEAR	MOC	モニタポイントの削除
53	MONITOR_DISPLAY	MOD	モニタ内容の表示
54	MONITOR_REFRESH	MOR	モニタ内容の自動更新制御
55	MONITOR_SET	MOS	モニタポイントの設定と表示
56	OPEN_WORKSPACE	OW	ワークスペースのオープン
57	QUIT	QU	HEW の終了
58	RADIX	RA	入力ラディックス（基数）の設定
59	REFRESH	RF	メモリ関連ウィンドウの更新
60	REGISTER_DISPLAY	RD	CPU レジスタ値の表示
61	REGISTER_SET	RS	CPU レジスタ値の設定
62	RESET	RE	CPU のリセット
63	REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
64	SAVE_SESSION	SE	現在のセッションをセーブ
65	SLEEP	-	コマンド実行の遅延
66	STEP	ST	ステップ実行（命令単位またはソース行単位）
67	STEP_MODE	SM	ステップモードの設定
68	STEP_OUT	SP	PC 位置の関数を終了するまでのステップ実行
69	STEP_OVER	SO	ステップオーバー実行
70	STEP_RATE	SR	ステップ実行速度の設定，表示
71	SUBMIT	SU	コマンドファイルの実行
72	SYMBOL_ADD	SA	シンボルの設定
73	SYMBOL_CLEAR	SC	シンボルの削除
74	SYMBOL_LOAD	SL	シンボル情報ファイルのロード
75	SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
76	SYMBOL_VIEW	SV	シンボルの表示
77	TCL	-	TCL の有効／無効の切換え
78	TIMER	TI	実行時間測定タイマの分解能の設定，表示
79	TOOL_INFORMATION	TO	現在登録されているツールの情報をファイルへ出力
80	TRACE	TR	トレース情報の表示
81	TRACE_ACQUISITION	TA	トレース取得条件の設定，表示
82	TRACE_BINARY_COMPARE	TBC	トレースバイナリファイルと現在のトレース情報の比較
83	TRACE_BINARY_SAVE	TBV	トレース情報をバイナリファイルに保存
84	TRACE_FILTER	TF	トレース情報のフィルタ
85	TRACE_STATISTIC	TST	統計情報解析の実行
86	TRIGGER_CLEAR	TGC	EXT.2 のトリガ出力条件の削除
87	TRIGGER_DISPLAY	TGD	EXT.2 のトリガ出力条件の表示
88	TRIGGER_SET	TGS	EXT.2 のトリガ出力条件の設定
89	UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトの依存関係を更新
90	USER_SIGNALS	US	ユーザシグナル情報の有効／無効の切換え

各コマンドのシンタックスについてはオンラインヘルプを参照ください。

付録 I　ハードウェア診断プログラムについて

E6000 エミュレータ用テストプログラムによる故障解析の手順について示します。

I.1　テストプログラムを実行するためのシステムセットアップ

テストプログラムを実行するためには、以下に示す機器が必要です。なお、本テストプログラムの実行時はユーザシステムインタフェースケーブル、ユーザシステム、および、拡張ＩＯボードを接続しないでください。

- E6000 エミュレータ（HS388REPI60H）
- PC
- E6000 PC インタフェースボード（以下のいずれかを指します。PC のインタフェース仕様に合わせて以下のインタフェースボードのいずれか一枚をご用意ください。）
  PCIバスインタフェースボード（HS6000EIC01HまたはHS6000EIC02H）
  PCMCIAインタフェースカード（HS6000EIP01H）

（1） PCにE6000 PCインタフェースボードを挿入し、付属のPCインタフェースケーブルを接続してください。

（2） PCインタフェースケーブルをE6000エミュレータ本体に接続してください。

（3） E6000エミュレータ本体に、付属のACアダプタを接続してください。

（4） PCを起動し、DOSプロンプトのコマンド入力待ち状態にしてください。

（5） E6000エミュレータ本体の電源をオンにしてください。

I.2　テストプログラムによる故障解析

E6000 エミュレータに添付されている CD-R(HS388REPI60SR)を PC の CD-ROM ドライブに挿入し、コマンドプロンプトでカレントディレクトリを <ドライブ>：\Diag フォルダに移動した後、使用している PC インタフェースボードの種類に従い、下記コマンドを入力すると直ちにテストプログラムが起動します。なお、テストプログラムの実行に際しては、HEW がインストールされていることを前提としています。

(1)　PCIバスインタフェースボード (HS6000EIC01HまたはHS6000EIC02H)
　　> TM388R -PCI (RET)

(2)　PCMCIAインタフェースカード (HS6000EIP01H)
　　> TM388R -PCCD (RET)

カレントディレクトリを <ドライブ>：\Diag フォルダに移動しない状態で ><ドライブ>：\Diag\TM388R -PCI (RET)のように他のカレントディレクトリからテストプログラムを起動した場合はテストプログラムが正しく動作しません。必ず <ドライブ>：\Diag フォルダにカレントディレクトリを移動してテストプログラムを実行してください。

なお、> TM388R -PCI -S（RET）のように-S をコマンドラインに追加すると、No.1 から No.11 までのテストを繰返し実行することができます。途中でテストを中断する場合は Q を入力してください。

【注】 1.　**<ドライブ>は** CD-ROM **ドライブのドライブ文字です。**
　　2.　**テストプログラム実行中は** CD-ROM **ドライブから** CD-R **を取り出さないでください。**

テストが実行されているときに表示されるメッセージとテスト内容は次の様になります。テストは No.1 から No.11 までです。

```
E6000 H8/3880 (16MHz) EMULATION BOARD Tests Vx.x
```

テストプログラムのスタートメッセージです。x.x はバージョン番号です。

```
Searching for interface card ...........OK
```

ホスト PC に PC インタフェースボードが正しく接続されていることを示します。

```
Checking emulator is connected ..........OK
```

ホスト PC と E6000 エミュレータが正しく接続されていることを示します。

```
  Emulator Board Information:
    Main Board ID       H'1
```

E6000 エミュレータ(下基板) の ID 番号で、常に 1 を示します。

```
    Emulation Board ID   H' 1a
```

E6000 エミュレータ(上基板)の ID 番号で、常に 1a を示します。

```
    SIMM              No SIMM module inserted
```

オプションのメモリボードの有無を示します。

```
01) Testing Main Board Register :
    IDR Register ....................OK
    BOC control register ............OK
    MODERE register .................OK
    MODERO register .................OK
```

E6000 エミュレータ上のレジスタのチェック結果(正常終了)を示します。

```
02) Testing Dual-Port RAM :
     Decode Test ..............................OK
     Marching Test ............................OK
```

E6000 エミュレータ上の Dual-Port RAM のデコードテスト、マーチングテストチェック結果(正常終了)を示します。

```
03) Testing Firmware RAM :
    Decode Test.  page range H'700 - H'71f .................OK
```

E6000 エミュレータ上の Firm RAM のデコードテストチェック結果(正常終了)を示します。

```
    Marching Test.  page range H'700 - H'71f ...............OK
```

E6000 エミュレータ上の Firm RAM のマーチングテストチェック結果(正常終了)を示します。

```
04) Testing Trace RAM :
    Decode Test.  page range H'000 - H'04f .................OK
```

E6000 エミュレータ上の Trace RAM のデコードテストチェック結果(正常終了)を示します。

```
    Marching Test.  page range H'000 - H'04f ...............OK
```

E6000 エミュレータ上の Trace RAM のマーチングテストチェック結果(正常終了)を示します。

```
05) Testing Mapping RAM :
    Decode Test.  page range H'200 - H'21f .................OK
```

E6000 エミュレータ上の Mapping RAM のデコードテストチェック結果(正常終了)を示します。

```
    Marching Test.  page range H'200 - H'21f ...............OK
```

E6000 エミュレータ上の Mapping RAM のマーチングテストチェック結果(正常終了)を示します。

```
06) Testing Emulation RAM :
    Decode Test ............................OK
    Marching Test ..........................OK
```

エミュレーションメモリのデコードテスト、マーチングテストチェック結果(正常終了)を示します。

```
07) Testing STEP Operation :
    Single Step Operation ..............OK
    Step Into Operation ................OK
08) Testing Key Break :
    Key Break ................................OK
09) Testing Emulation RAM Hardware Break :
    GRD Break ..........................OK
    WPT Break ..........................OK
    IN Rom .............................OK
    EML RAM Write ......................OK
    Access Error Break .................OK
10) Testing Go Reset ..................OK

11) Testing Double Stack, I/O Stack Test :
    Double Stack .......................OK
    I/O Stack ..........................OK
0 total errors
Tests passed, emulator functioning correctly
```

ステップ実行制御回路のチェック結果(正常終了)を示します。

強制ブレーク制御回路のチェック結果(正常終了)を示します。

不当アクセスブレーク制御回路のチェック結果(正常終了) を示します。

Go Reset 制御回路のチェック結果（正常終了）を示します。

スタック制御回路のチェック結果（正常終了）を示します。

エラー発生数の合計を示します。

テストプログラムにより正常動作が確認されたことを示します。

I.3　エラー発生時の処理

E6000 エミュレータをご使用中に動作エラーが発生した場合は、お手数ですが下記故障症状調査書に症状をご記入のうえ、担当営業まで FAX でご連絡いただくようお願い申し上げます。

故障症状調査書

ご購入営業担当行

お客様ご芳名 会社名＿＿＿＿＿＿＿＿＿＿＿＿

ご担当者名 ＿＿＿＿＿＿＿＿様

TEL＿＿＿＿＿＿＿＿＿＿＿＿

FAX＿＿＿＿＿＿＿＿＿＿＿＿

1) 不具合発生製品型名およびシステム構成

a）E6000エミュレータ（HS3880EPI60H, HS388REPI60H）

シリアルNo.＿＿＿＿、レビジョン＿＿＿＿

（ケース裏面に表示しています： シリアルNo.は数字4桁、レビジョンはそれに続くアルファベットです）

b）PCインタフェースボード

型式 HS＿＿＿＿＿＿＿＿H、　シリアルNo.＿＿＿＿、レビジョン＿＿＿＿

（基板上に捺印表示しています）

c）ユーザシステムインタフェースケーブル

型式 HS＿＿＿＿＿＿＿＿H、　シリアルNo.＿＿＿＿、レビジョン＿＿＿＿

（基板上に捺印表示しています）

d）オプションボード

型式 HS＿＿＿＿＿＿＿＿H、　シリアルNo.＿＿＿＿、レビジョン＿＿＿＿

（基板上に捺印表示しています）

e）HEW（HS388REPI60SR） バージョンV＿＿＿＿

（CD-RにV.x.xx Release xxと表示しています）

f）ご使用になっているPC

メーカ名＿＿＿＿＿＿＿＿、 型式＿＿＿＿＿＿＿＿

使用OS（Windows®98SE, WindowsNT®4.0, Windows®Me, Windows®2000, Windows®XP　←いずれかに○）

2) ターゲットシステムの使用条件

a）デバッグ対象マイコン型名：H8/＿＿＿＿＿＿＿＿

b）動作モード：モード＿＿＿＿＿＿

c）ターゲットシステム電圧：＿＿＿＿＿＿V

d）使用クロック：（貸出しクロック、Xtal発振、外部クロック入力　←いずれかに○）

e）動作周波数：＿＿＿＿＿＿MHz

3) エラー発生状況

a～cのいずれかに○をつけ、内容を記載してください。

a) HEWがエミュレータと接続できない

（エラーメッセージ：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿）

b) テストプログラムでエラーが発生

（テスト番号：＿＿＿＿＿＿エラーメッセージ：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿）

c) デバッグ中にエラーが発生

下記の各項目についてエラー内容を記載してください。

4) メモリのデータ化けは発生していますか？（はい、 いいえ ←いずれかに○）
   a) データ化けを起こしている箇所のMemory Mapping設定
   （ROM, RAM, I/O, Emulator, その他＿＿＿＿ ←いずれかに○）
   b) データ化けを起こしている箇所はメモリウィンドウでリード／ライトできますか？
   （はい、 いいえ ←いずれかに○）

5）ターゲットシステムに対して入出力できない特定の信号はありますか？
   （はい、 いいえ ←いずれかに○）
   a) 信号名：＿＿＿＿、ピン番号：＿＿＿＿
   b) 信号レベル異常：（High固定、Low固定、中間レベル　←いずれかに○）

6）上記以外のエラーについては、下記に症状を記載いただくようお願いいたします。

ルネサスマイクロコンピュータ開発環境システム
ユーザーズマニュアル
H8/388R E6000エミュレータ

発行年月日　2004 年 11 月 17 日　Rev.1.00
発　　行　株式会社ルネサス テクノロジ 営業企画統括部
　　〒100-0004 東京都千代田区大手町 2-6-2
編　　集　株式会社ルネサス小平セミコン 技術ドキュメント部

© 2004. Renesas Technology Corp., All rights reserved. Printed in Japan.

株式会社ルネサス テクノロジ 営業企画統括部　〒100-0004 東京都千代田区大手町2-6-2 日本ビル

営業お問合せ窓口
株式会社ルネサス販売

http://www.renesas.com

本社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
京浜支社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西東京支社	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
札幌支店	〒060-0002	札幌市中央区北二条西4-1 (札幌三井ビル5F)	(011) 210-8717
東北支社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
いわき支店	〒970-8026	いわき市平小太郎町4-9 (損保ジャパンいわき第二ビル3F)	(0246) 22-3222
茨城支店	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新潟支店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松本支社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中部営業本部	〒460-0008	名古屋市中区栄3-13-20 (栄センタービル4F)	(052) 261-3000
浜松支店	〒430-7710	浜松市板屋町111-2（浜松アクトタワー10F）	(053) 451-2131
西部営業本部	〒541-0044	大阪市中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
北陸支社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
広島支店	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
鳥取支店	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
九州支社	〒812-0011	福岡市博多区博多駅前2-17-1 (ヒロカネビル本館5F)	(092) 481-7695
鹿児島支店	〒890-0053	鹿児島市中央町12-2 (明治安田生命鹿児島中央町ビル)	(099) 284-1748

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：カスタマサポートセンタ　E-Mail: csc@renesas.com

Colophon 3.0

H8/388R E6000 エミュレータ
ユーザーズマニュアル
H8/388R E6000　　HS388REPI60HJ-U2

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

RJJ10B0165-0100