

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事用の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが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にご使用ください。お客様にかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社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。

H8S ファミリ用 E10A エミュレータ

ユーザーズマニュアル

H8S ファミリ E10A HS0005KCM05HJ

ルネサスマイクロコンピュータ開発環境システム

ご注意

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すては、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、正確を期するため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、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を負いません。
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、その他お気づ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：
ここで言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）エミュレータ、（2）ユーザインタフェースケーブル
お客様のホストコンピュータ及びユーザシステムは含みません。
- 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。この使用目的にしたがって、当エミュレータを正しくお使いください。この目的以外の当エミュレータの使用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。
 - － 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）
 - － 原子力開発機器用
 - － 航空機開発機器用
 - － 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、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ってください。特に、当エミュレータを初めて使う人は、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勧めします。

- 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、有償で交換を行います。

また、日本国内で購入され、かつ、日本国内で使用するものに限りします。

- 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。

- 著作権所有：

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、実物と違っていることがあります。

- デバイス名について：

このユーザーズマニュアルのデバッグ編の1章から6章までは、例として、H8S/xxxxというデバイス名を使用しています。

- 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お使いください。

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ではありません。



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。



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约店保守担当にお申し付けください。
2. ホストコンピュ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わないでください。抜き差しを行った場合、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. ユーザインタフェースケーブルのユーザシステム上側のコネクタとユーザインタフェースケーブル側のコネクタの向き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。

エミュレータ使用時の注意事項

このエミュレータ使用時の注意事項に記載されている事項は、当エミュレータを使用するうえで全ての場合に該当し、例外は存在しません。したがって、エミュレータを使用する前に以下に示されている警告文をよく読み、完全に理解してください。ただし、ここに記載されている事項はエミュレータ使用時における共通の警告のみが記載されており、これがエミュレータを使用するうえでの全ての警告ではありません。

警告

ホストコンピュータまたはユーザシステムのパワーオン時には、全てのケーブル、およびユーザインタフェースの抜き差しを行わないでください。

抜き差しを行った場合、ホストコンピュータとエミュレータおよびユーザシステムの発煙発火、および機器の破壊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の破壊の可能性があります。

注意

ホストコンピュータとユーザシステムの位置関係により、ユーザインタフェース部に大きなストレスが加わり、接点、接触不良等の機械的破損を招く原因となります。また、使用中にホストコンピュータまたはユーザシステムが動いてしまうと、ユーザインタフェース部に思わぬストレスを与える事になります。ホストコンピュータおよびユーザシステムの位置に十分ご注意下さい。

はじめに

High-performance Embedded Workshop（以下、HEW と略します）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、デバッガ編および付録の 3 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
デバッガ編	使用前の準備、E10A エミュレータ機能、デバッグ機能、チュートリアル
付録	トラブルシューティング、正規表現、プレースホルダ、I/O ファイルフォーマット、シンボルファイルフォーマット、ウインドウ機能一覧、コマンドライン機能、HEW の注意事項

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。

HEW は、インストール上、各種言語にカスタマイズされています。このマニュアルでは、HEW アプリケーションの日本語版について説明します。

Microsoft®, 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
<u>"文字列の入力"</u>	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

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	5
1.2.4 ワークスペースウィンドウ	8
1.2.5 エディタウィンドウ	12
1.2.6 アウトプットウィンドウ	13
1.2.7 ステータスバー	13
1.3 ヘルプ機能	14
1.4 HEW を起動する	15
1.5 HEW を終了する	15
1.6 ツールシステム概要	16
2 ビルドの基本	17
2.1 ビルド処理	17
2.2 プロジェクトファイル	18
2.2.1 プロジェクトにファイルを追加する	18
2.2.2 プロジェクトからファイルを削除する	21
2.2.3 ビルドからプロジェクトファイルを除外する	23
2.2.4 ビルドへプロジェクトファイルを入れる	23
2.3 ファイル拡張子とファイルグループ	24
2.4 ファイルのビルド方法を設定する	32
2.5 ビルドのコンフィグレーション	32
2.5.1 ビルドコンフィグレーションを選択する	33
2.5.2 ビルドコンフィグレーションを追加、削除する	34
2.6 プロジェクトをビルド実行する	35
2.6.1 プロジェクトをビルド実行する	35
2.6.2 1つのファイルをビルド実行する	35
2.6.3 ビルド実行を中止する	36
2.6.4 複数のプロジェクトをビルド実行する	37
2.6.5 アウトプットウィンドウ	37
2.6.6 アウトプットウィンドウの内容の制御	38
2.7 ファイル依存関係	39

2.8	ワークスペースウィンドウの構成	40
2.8.1	各ファイルの下に依存を表示する	40
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	41
2.8.3	ファイルのパスを表示する	42
2.9	アクティブプロジェクトを設定する	42
2.10	ワークスペースにプロジェクトを追加する	43
2.11	プロジェクト間の依存関係を指定する	44
2.12	ワークスペースからプロジェクトを削除する	45
2.13	ワークスペースからプロジェクトをロード、アンロードする	45
2.14	ワークスペースの相対プロジェクトパス	46
3	ビルドの応用	47
3.1	ビルド実行の復習	47
3.1.1	ビルドとは?	47
3.2	カスタムビルドフェーズを作成する	49
3.3	ビルドのフェーズ順序	53
3.3.1	ビルドのフェーズ順序	54
3.3.2	ビルドファイルのフェーズ順序	57
3.4	カスタムビルドフェーズのオプション設定	58
3.4.1	オプションタブ	59
3.4.2	Output Files タブ	60
3.4.3	Dependent Files タブ	62
3.5	ファイルのマッピング	64
3.6	ビルドを管理する	66
3.7	ビルドの出力のログを取る	67
3.8	ツールチェーンのバージョンを変更する	68
3.9	外部デバッガを使う	69
3.10	メイクファイルの生成	71
4	エディタの使用	73
4.1	エディタウィンドウ	73
4.2	複数のファイルを使う	74
4.2.1	エディタツールバー	75
4.2.2	エディタツールバーボタン	75
4.2.3	検索ツールバーボタン	76
4.2.4	ブックマークツールバーボタン	77
4.2.5	テンプレートツールバーボタン	77
4.3	標準のファイル操作	78
4.3.1	新規ファイルの作成	78
4.3.2	ファイルの保存	78
4.3.3	全ファイルの保存	78
4.3.4	ファイルを開く	79
4.3.5	ファイルを閉じる	79

4.4	ファイルを編集する	80
4.5	検索とファイル内の移動	81
4.5.1	テキストの検索	81
4.5.2	複数のファイル間でのテキスト検索	82
4.5.3	テキストを置換する	83
4.5.4	指定した行にジャンプする	84
4.6	ブックマーク	84
4.7	ファイルを印刷する	85
4.8	テキストのレイアウト	85
4.8.1	ページ設定	85
4.8.2	タブを変更する	87
4.8.3	自動インデント	87
4.9	ウィンドウを分割する	88
4.10	テキストの表示の変更方法	90
4.10.1	エディタのフォントを変更する	90
4.11	シンタックスを色づけする	91
4.12	テンプレート	93
4.12.1	テンプレートを設定する	93
4.12.2	テンプレートを削除する	94
4.12.3	テンプレートを挿入する	94
4.12.4	かっこの組み合わせ	95
4.13	Editor カラムの管理	96
5	ツール管理	99
5.1	ツールの位置	100
5.2	HEW 登録ファイル (*.HRF)	100
5.3	ツールを登録する	102
5.3.1	ドライブのツール検索	102
5.3.2	ツールを一つ登録する	103
5.4	ツールの登録を取り消す	103
5.5	ツールのプロパティの参照と編集	103
5.6	ツールのアンインストール	106
5.7	テクニカルサポートについて	108
5.8	カスタムプロジェクトタイプ	109
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	114
6.3	ヘルプシステムを構築する	117
6.4	ワークスペースオプションを指定する	118
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	119
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	119

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	120
6.4.5	ワークスペース保存前に確認チェックボックス	120
6.4.6	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
6.4.7	セッション保存前に確認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	121
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	外部デバッグを使う	123
6.8	カスタムプレースホルダを使う	124
6.9	確認ダイアログボックスを使う	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	133
8.1.2	ユーザ定義メニューオプション	134
8.2	バージョン管理コマンドを定義する	135
8.2.1	Executable return code オプション	136
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	139
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	151
9.2	Visual SourceSafe コマンド	152
9.2.1	バージョン管理からファイルを削除する	152
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	152
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	153
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	153
9.2.5	チェックアウト操作を取り消す	153
9.2.6	ファイルの状態を表示する	154
9.2.7	ファイル履歴を表示する	154
9.3	Visual SourceSafe 統合化オプション	155

デバッグ編	157
1. はじめに	159
1.1 使用環境条件	162
1.2 梱包品の確認	163
2. E10A エミュレータ機能	165
2.1 機能概要	165
2.2 トレース機能	167
2.3 ブレーク機能	168
2.4 メモリアクセス機能	169
2.5 スタックトレース機能	170
2.6 オンラインヘルプ	170
3. 使用前の準備	171
3.1 H8S/xxxx E10A エミュレータのインストール	173
3.1.1 Windows® 98 operating system または Windows® Me operating system への インストール	174
3.1.2 Windows NT® 4.0 operating system へのインストール	175
3.1.3 Windows® 2000 operating system または Windows® XP operating system への インストール	176
3.2 ホストコンピュータとカードエミュレータとの接続	177
3.3 カードエミュレータとユーザシステムとの接続	179
3.4 システムチェック	181
3.5 アンインストール	189
4. デバッグの準備をする	193
4.1 ワークスペース、プロジェクト、ファイル	193
4.2 HEW の起動方法	194
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)	195
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)	199
4.2.3 既存のワークスペースを指定する場合	204
4.3 E10A エミュレータ起動時の設定	206
4.4 デバッグセッション	208
4.4.1 セッションを選択する	208
4.4.2 セッションの追加と削除	209
4.4.3 セッション情報を保存する	211
4.5 エミュレータの接続	212
4.6 エミュレータの終了	214
5. デバッグ	215
5.1 エミュレーション環境を設定する	215
5.1.1 [Configuration]ダイアログボックスを開く	215
5.1.2 [General]ページ	215

5.2	プログラムを表示する	217
5.2.1	ソースコードを表示する	217
5.2.2	アセンブリ言語コードを表示する	219
5.2.3	アセンブリ言語コードを修正する	221
5.2.4	特定のアドレスを見る	222
5.2.5	現在のプログラムカウンタアドレスを見る	222
5.3	コマンドラインインタフェースでデバッグする	223
5.3.1	[Command Line]ウィンドウを開く	223
5.3.2	コマンドファイルを設定する	224
5.3.3	コマンドファイルを実行する	224
5.3.4	コマンド実行を中断する	224
5.3.5	ログファイルを設定する	225
5.3.6	ログファイルへの出力を開始/停止する	225
5.3.7	ファイルの full パスを入力する	225
5.3.8	プレースホルダを入力する	225
5.4	レジスタ内容を見る	226
5.4.1	[Register]ウィンドウを開く	226
5.4.2	ビットレジスタを拡張する	226
5.4.3	レジスタの内容を修正する	227
5.4.4	レジスタの内容を使用する	227
5.5	メモリを操作する	228
5.5.1	メモリ領域を見る	228
5.5.2	異なるフォーマットでデータを表示する	229
5.5.3	ウィンドウを分割表示する	229
5.5.4	異なるメモリ領域を見る	230
5.5.5	メモリの内容を修正する	230
5.5.6	メモリ範囲を選択する	230
5.5.7	メモリ内の値を探す	230
5.5.8	メモリ範囲に値をフィルする	232
5.5.9	メモリ領域をコピーする	232
5.5.10	メモリ領域を保存、検証する	233
5.5.11	ウィンドウ内容更新を抑止する	234
5.5.12	ウィンドウ内容を更新する	234
5.5.13	メモリ内容を比較する	234
5.5.14	メモリ領域をファイルからロードする	235
5.6	I/O メモリを見る	236
5.6.1	[IO] ウィンドウを開く	236
5.6.2	I/O Register 表示を拡張する	237
5.6.3	I/O レジスタの内容を修正する	237
5.7	現在の状態を表示する	238
5.8	メモリ内容の日本語表示	239
5.8.1	メモリ内容を UNICODE 形式で表示する	239
5.8.2	メモリ内容を SJIS 形式または EUC 形式で表示する	239
5.9	ラベルを見る	242
5.9.1	ラベルを一覧にする	242

5.9.2	ラベルを追加する	243
5.9.3	ラベルを編集する	243
5.9.4	ラベルを削除する	244
5.9.5	すべてのラベルを削除する	244
5.9.6	ラベルをファイルからロードする	245
5.9.7	ラベルをファイルに保存する	245
5.9.8	ラベルを検索する	246
5.9.9	次を検索する	246
5.9.10	ラベルに対応するソースプログラムを表示する	246
5.10	プログラムを実行する	247
5.10.1	リセットから実行を開始する	247
5.10.2	実行を継続する	247
5.10.3	カーソルまで実行する	247
5.10.4	開始アドレスを指定して実行する	248
5.10.5	シングルステップ	249
5.10.6	複数のステップ	249
5.11	プログラムを停止する	250
5.11.1	Halt による停止	250
5.11.2	標準のブレークポイント(PC ブレークポイント).....	251
5.12	Elf/Dwarf2 のサポート	252
5.12.1	C/C++演算子	253
5.12.2	C/C++の式	253
5.12.3	複数ラベルをサポートする	253
5.12.4	オーバーレイプログラムのデバッグ	254
5.13	変数の表示	257
5.13.1	ツールチップウォッチ	257
5.13.2	インスタントウォッチ	257
5.13.3	[Watch]ウィンドウ	258
5.13.4	[Locals]ウィンドウ	262
5.14	イベントポイントを使用する	263
5.14.1	PC ブレークポイントとは	263
5.14.2	Break condition とは	263
5.14.3	[Event]ウィンドウを開く	263
5.14.4	PC ブレークポイントを設定する	263
5.14.5	追加	264
5.14.6	編集	264
5.14.7	有効	264
5.14.8	無効	264
5.14.9	削除	264
5.14.10	すべてを削除	264
5.14.11	ソースを表示	265
5.14.12	[Set Break]ダイアログボックス	265
5.14.13	ブレークコンディションを設定する	266
5.14.14	編集	267
5.14.15	有効	267
5.14.16	無効	267

5.14.17	削除	267
5.14.18	すべてを削除	267
5.14.19	ソースを表示	267
5.14.20	シーケンシャル設定	268
5.14.21	ブレイクコンディションの編集	268
5.14.22	ブレイクコンディションの設定内容を変更する	268
5.14.23	ブレイクコンディションを有効にする	268
5.14.24	ブレイクコンディションを無効にする	268
5.14.25	ブレイクコンディションを削除する	268
5.14.26	ブレイクコンディションをすべて削除する	268
5.14.27	ブレイクコンディションのソース行を表示する	268
5.14.28	[Break condition 1]ダイアログボックス	269
5.15	トレース情報を見る	271
5.15.1	[Trace Window]ウィンドウを開く	271
5.15.2	トレース情報を取得する	271
5.15.3	トレース情報をクリアする	272
5.15.4	トレース情報をファイルに保存する	272
5.15.5	[Source]ウィンドウを表示する	272
5.15.6	ソース表示を整形する	272
5.16	関数呼び出し履歴を見る	273
5.16.1	[Stack Trace]ウィンドウを開く	273
5.16.2	ソースプログラムを表示する	273
5.16.3	表示形式を設定する	274
5.17	メモリ内容を画像形式で表示する	275
5.17.1	[Image]ウィンドウを開く	275
5.17.2	ウィンドウを自動更新する	277
5.17.3	ウィンドウを更新する	277
5.17.4	ピクセル情報を表示する	277
5.18	メモリ内容を波形形式で表示する	279
5.18.1	[Waveform View]ウィンドウを開く	279
5.18.2	ウィンドウを自動更新する	280
5.18.3	ウィンドウを更新する	280
5.18.4	拡大表示する	280
5.18.5	縮小表示する	280
5.18.6	最初のサイズに戻す	280
5.18.7	拡大/縮小倍率を設定する	281
5.18.8	横軸のサイズを設定する	281
5.18.9	カーソルを非表示にする	281
5.18.10	サンプリング情報を表示する	281
6.	チュートリアル	283
6.1	はじめに	283
6.2	HEW の起動	284
6.3	E10A エミュレータのセットアップ	284
6.4	[Configuration]ダイアログボックスの設定	285
6.5	RAM の動作チェック	287

6.6	チュートリアルプログラムのダウンロード	289
6.6.1	チュートリアルプログラムをダウンロードする	289
6.6.2	ソースプログラムを表示する	290
6.7	PC ブレークポイントの設定	291
6.8	レジスタ内容の変更	292
6.9	プログラムの実行	293
6.10	ブレークポイントの確認	296
6.11	メモリ内容の確認	297
6.12	変数の参照	298
6.13	プログラムのステップ実行	301
6.13.1	ステップインの実行	301
6.13.2	ステップアウトの実行	303
6.13.3	ステップオーバーの実行	304
6.14	プログラムの強制ブレーク	305
6.15	ローカル変数の表示	306
6.16	ブレーク機能	307
6.16.1	PC ブレーク機能	307
6.17	ハードウェアブレーク機能	312
6.18	トレース機能	315
6.18.1	トレースウィンドウの表示方法	315
6.19	スタックトレース機能	316
6.20	さてつぎは？	318
付録 A トラブルシューティング		319
付録 B 正規表現		321
付録 C プレースホルダ		323
C.1	プレースホルダとは？	323
C.2	プレースホルダを挿入する	324
C.3	使用できるプレースホルダ	326
C.4	プレースホルダを使うにあたって	329
付録 D I/O ファイルフォーマット		331
D.1	ファイルフォーマット	331
付録 E シンボルファイルフォーマット		335
付録 F ウィンドウ機能一覧		337
付録 G コマンドライン機能		341
付録 H HEW の注意事項		343

HEW 編

1 概要

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ーのために、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。

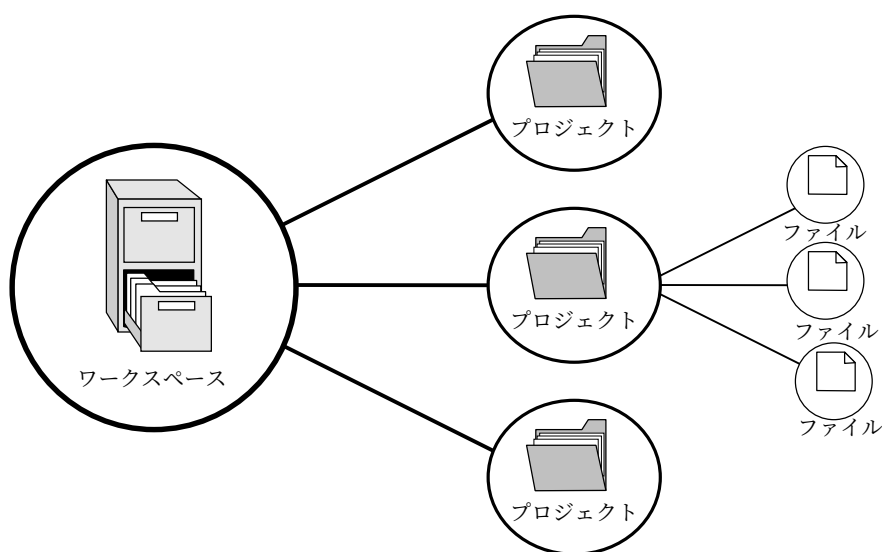


図 1.1: ワークスペース、プロジェクト、ファイル

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 概要

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。

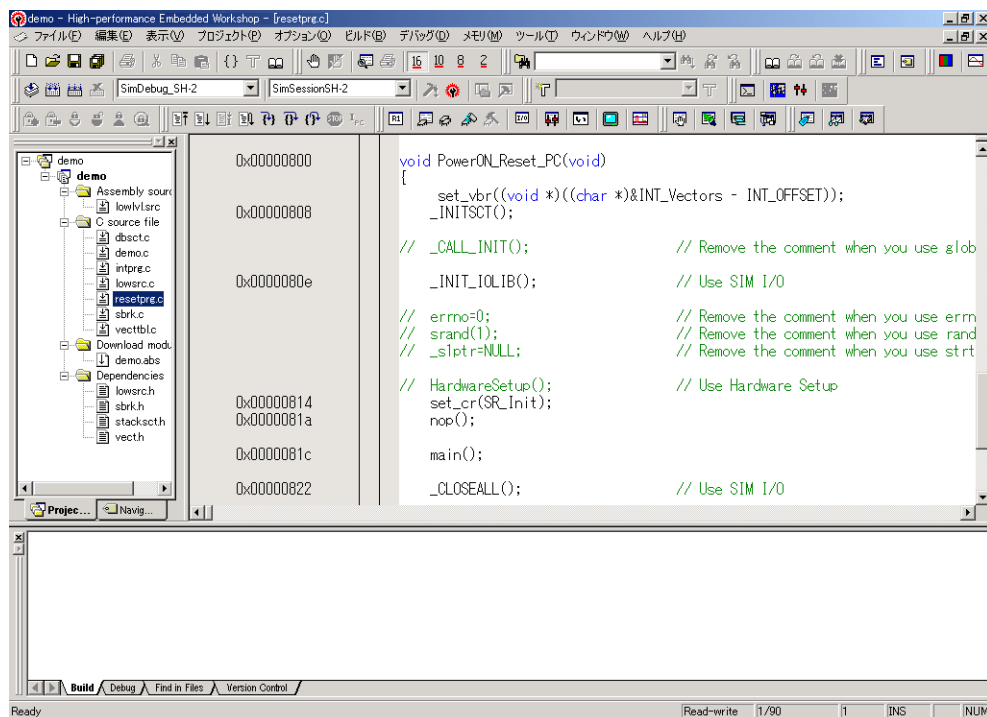


図 1.2: HEW メインウィンドウ

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、開いているワークスペース、プロジェクト、ファイルが表示されます。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 のスタートバー上に 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->編集]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9つ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], [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9つ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。

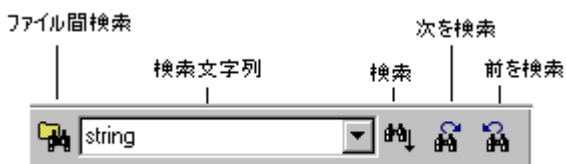


図 1.7: 検索ツールバー

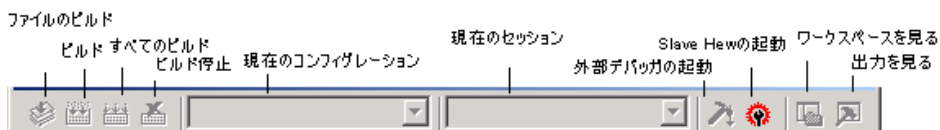


図 1.8: 標準ツールバー

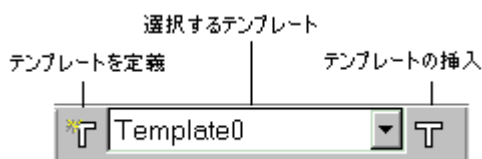


図 1.9: テンプレートツールバー

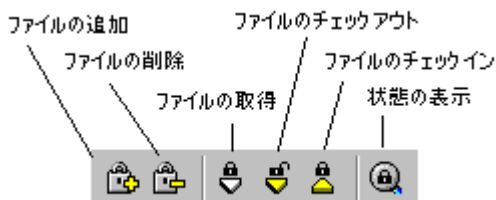


図 1.10: バージョン管理ツールバー

[標準]ツールバーがドッキング（連結）状態のとき、図 1.11.(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 1.11.(i) がドッキング状態、1.11.(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。

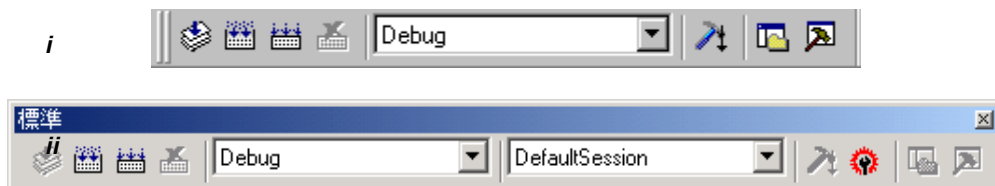


図 1.11: 標準ツールバーのドッキング／フローティング状態

- ➡メニューバーまたは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メニューバーまたは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メニューバーまたは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。

- ➡メニューバーまたはツールバーをフローティング状態にするには
ドッキング状態のメニューバーまたは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、ドッキング状態のメニューバーまたは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

HEW を起動した場合、ワークスペースウィンドウにはタブが1つだけあります。これが [Projects] タブです。ワークスペースを開くと、ワークスペースウィンドウに2つのデフォルトタブを表示します。[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 1.12)。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。

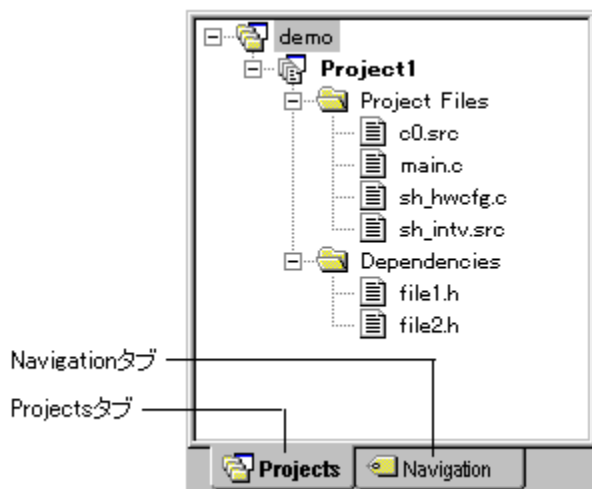


図 1.12: [ワークスペース] ウィンドウ [Projects] タブ

[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 1.13 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。

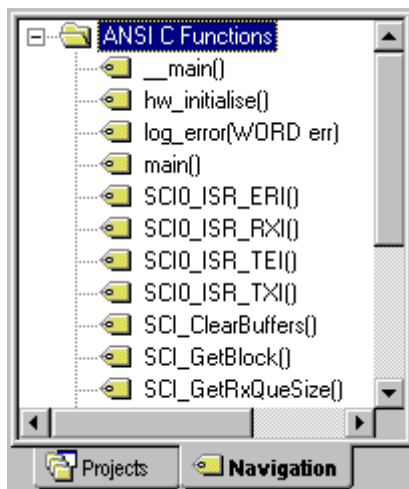


図 1.13: [ワークスペース] ウィンドウ [Navigation] タブ

- ☛ 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、ツールバー、またはメニューバーを 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 1.14.(i) にはドッキング状態のワークスペースウィンドウ、図 1.14.(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。

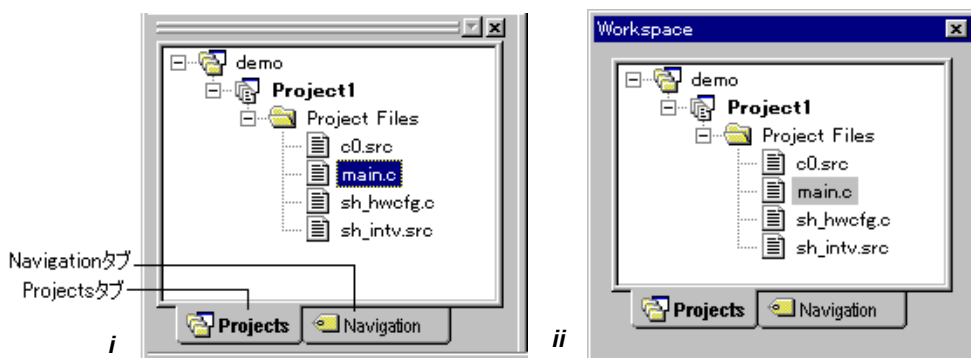


図 1.14: ワークスペースウィンドウのドッキング／フローティング状態

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 1.15 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 1.15: 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进行操作します。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ノート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 1.16)。

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。

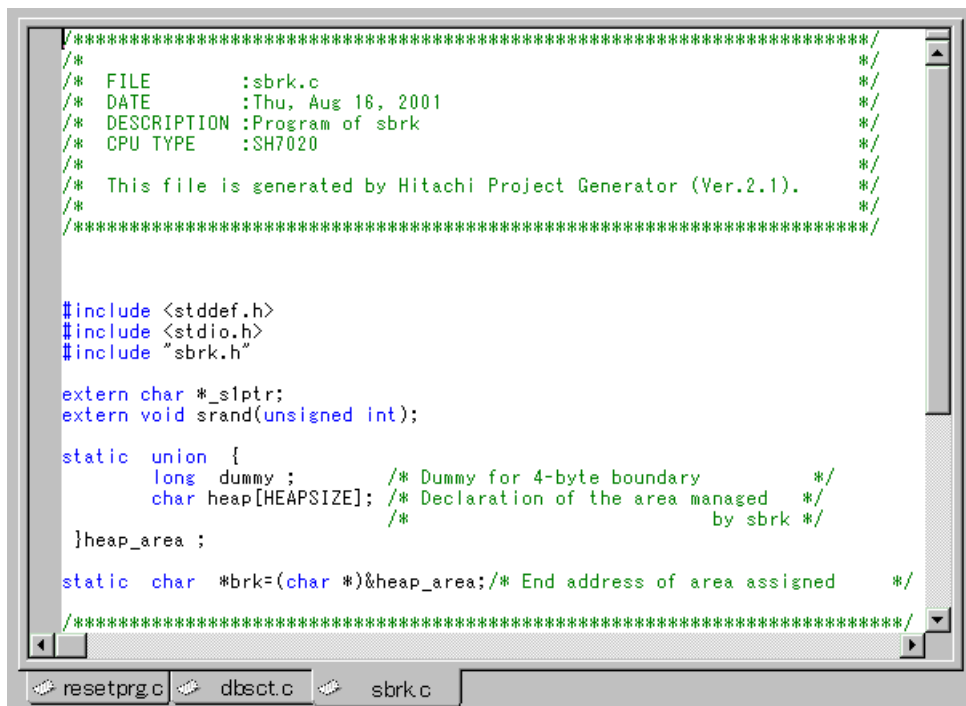


図 1.16: エディタウィンドウ

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、ビューからアウトプットウィンドウをすぐに削除します。

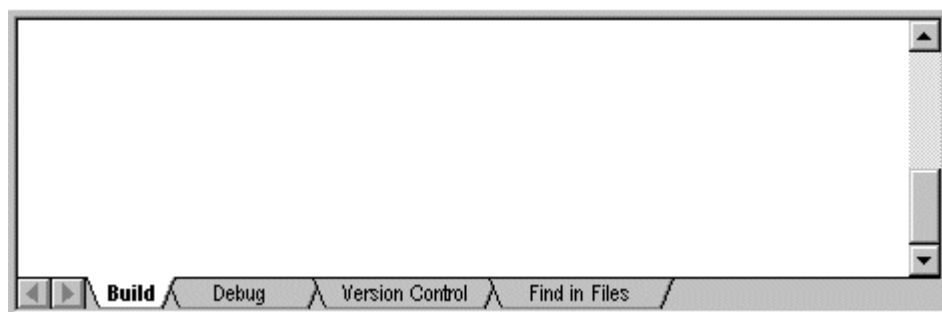


図 1.17: アウトプットウィンドウ

[Debug]タブにはあらゆるデバッグ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7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 1.18 にステータスバーを示します。

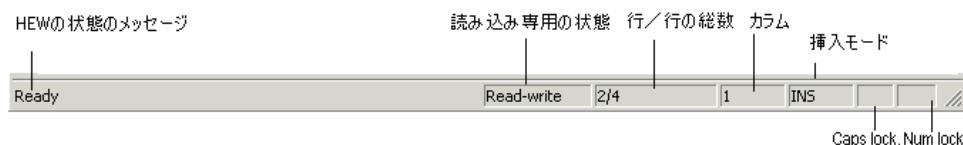


図 1.18:ステータスバー

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 1.19）。

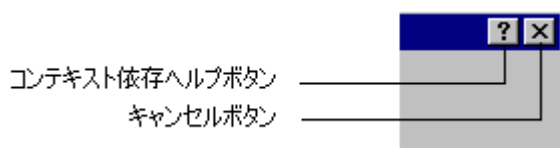


図 1.19: Help ボタン

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、[High-performance Embedded Workshop 2]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 1.20 に示す[ようこそ!]ダイアログボックスが開きます。

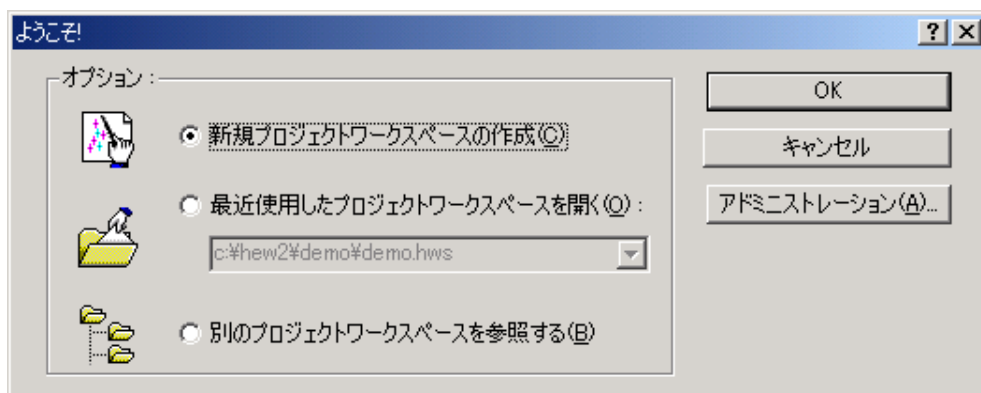


図 1.20: ようこそ! ダイアログボックス

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え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必要があります。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

ビルド処理の一般例を図 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）。いずれにせよ、原則は同じです。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。

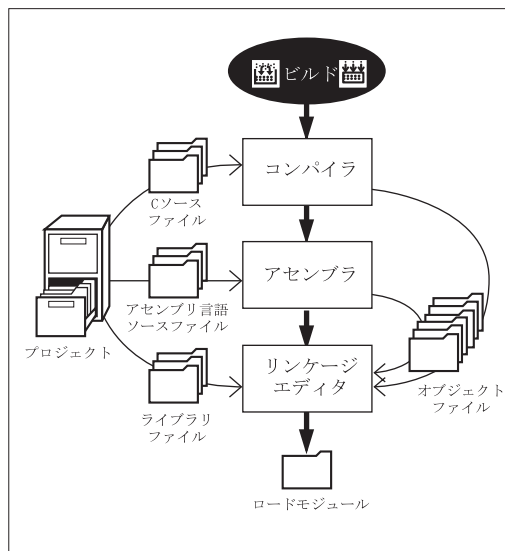


図 2.1: ビルド処理の一般例

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 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)。

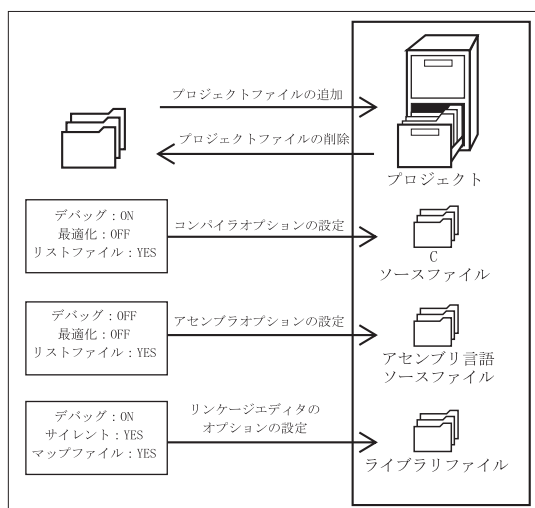


図 2.2: プロジェクトの編集

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。

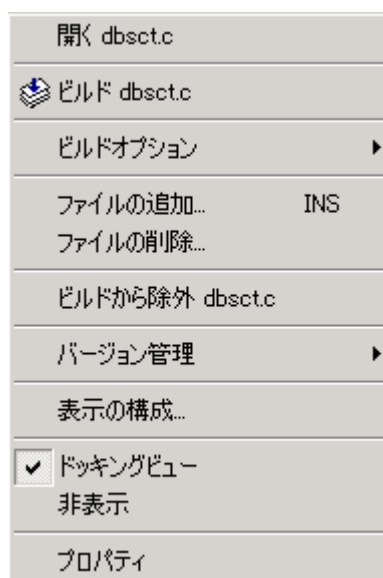


図 2.3: Project ポップアップメニュー

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。

2 ビルドの基本

プロジェクトに新しいファイルを追加するには、数多くの方法があります。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。

ビルドファイル "dbsect.c"	
切り取り (I)	Ctrl+X
コピー (C)	Ctrl+C
貼り付け (P)	Ctrl+V
プロジェクトにファイルの追加 ▶	
検索 (F)...	Ctrl+F
ファイル内から検索 (F)...	F4
置換 (E)...	Ctrl+H
ジャンプ (G)	Ctrl+G
{ } 括弧の呼応 (M)	Shift+Ctrl+M
ブックマーク (Q)	▶
テンプレート (S)	▶
ブレークポイントの挿入/削除 (K)	F9
 ブレークポイントの有効化/無効化 (N)	Ctrl+F9
表示カラムの設定...	
インスタントウォッチ	
カーソル位置まで実行 (U)	
カーソル位置にPCを設定	
逆アセンブリ	

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)。

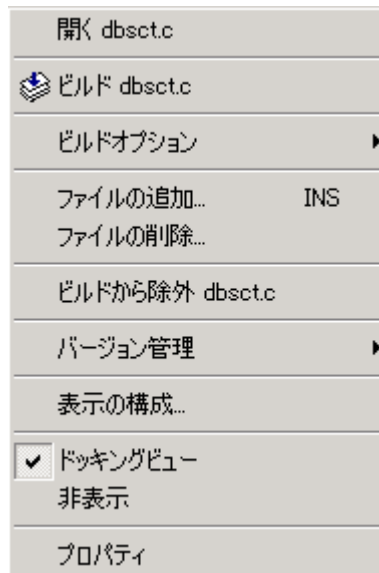


図 2.5: Project タブポップアップメニュー

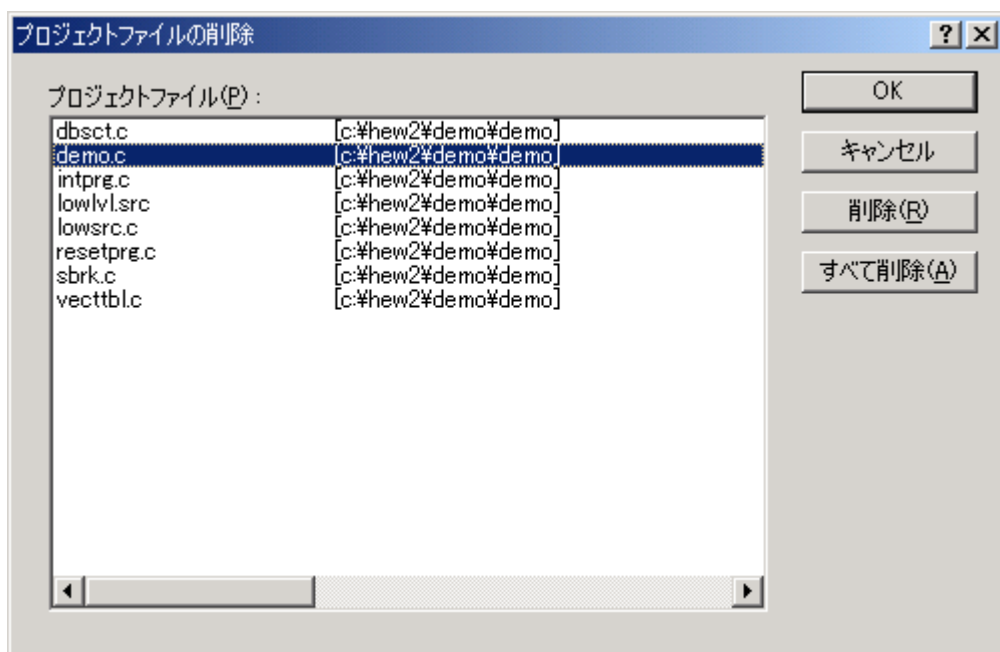


図 2.6: プロジェクトファイルの削除 ダイアログボックス

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[I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 HEW に要求できます。

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。

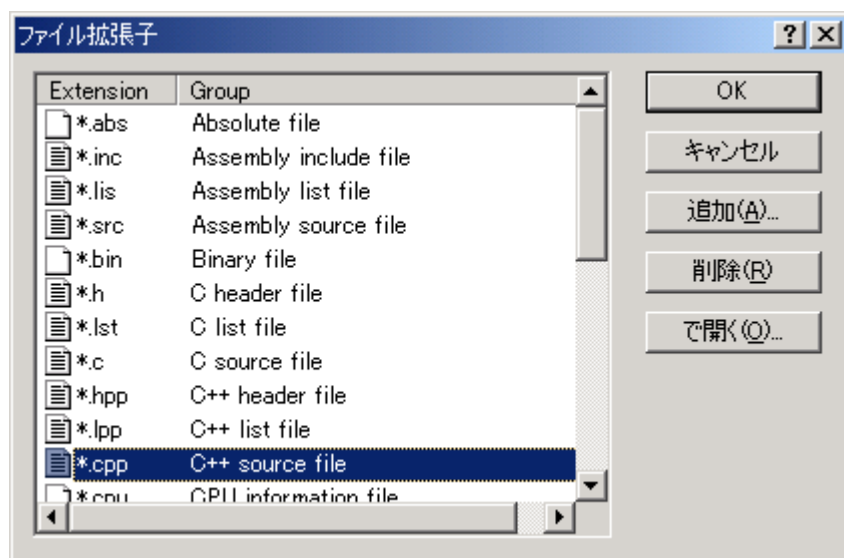


図 2.7: ファイル拡張子 ダイアログボックス

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)。

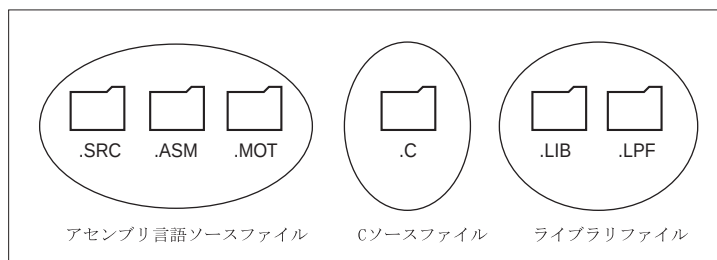


図 2.8: ファイル拡張子とファイルグループ

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。“Open”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。

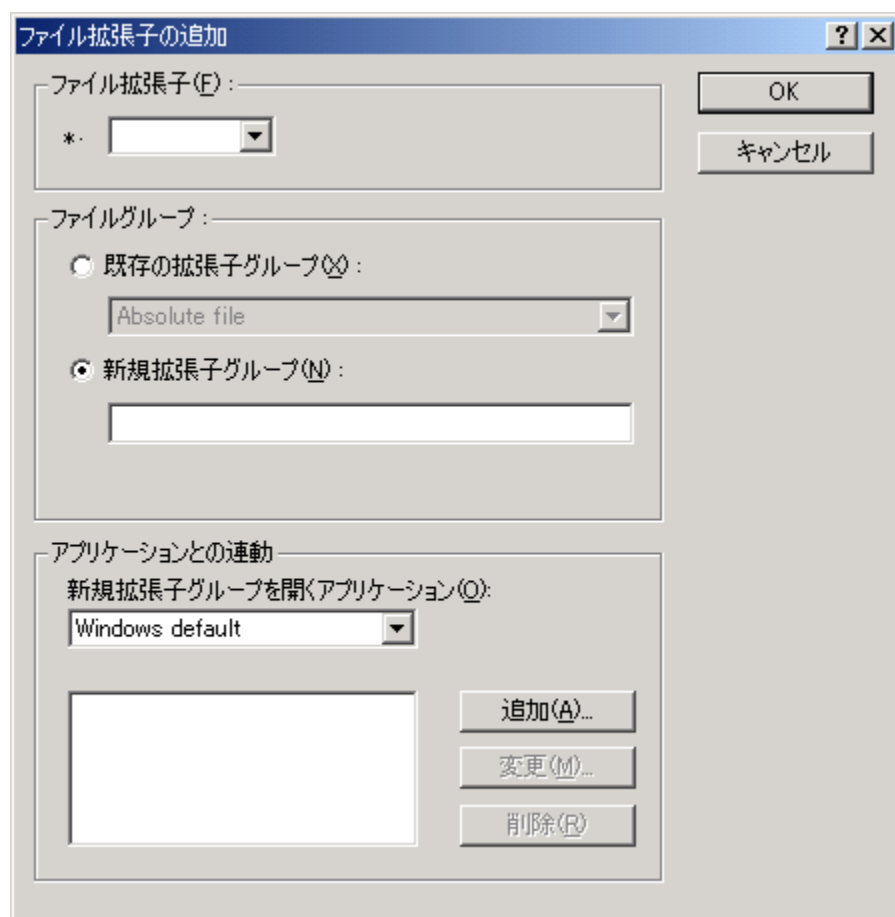


図 2.9: ファイル拡張子の追加ダイアログボックス
(新規グループ)

2 ビルドの基本

ビルダに認識され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。

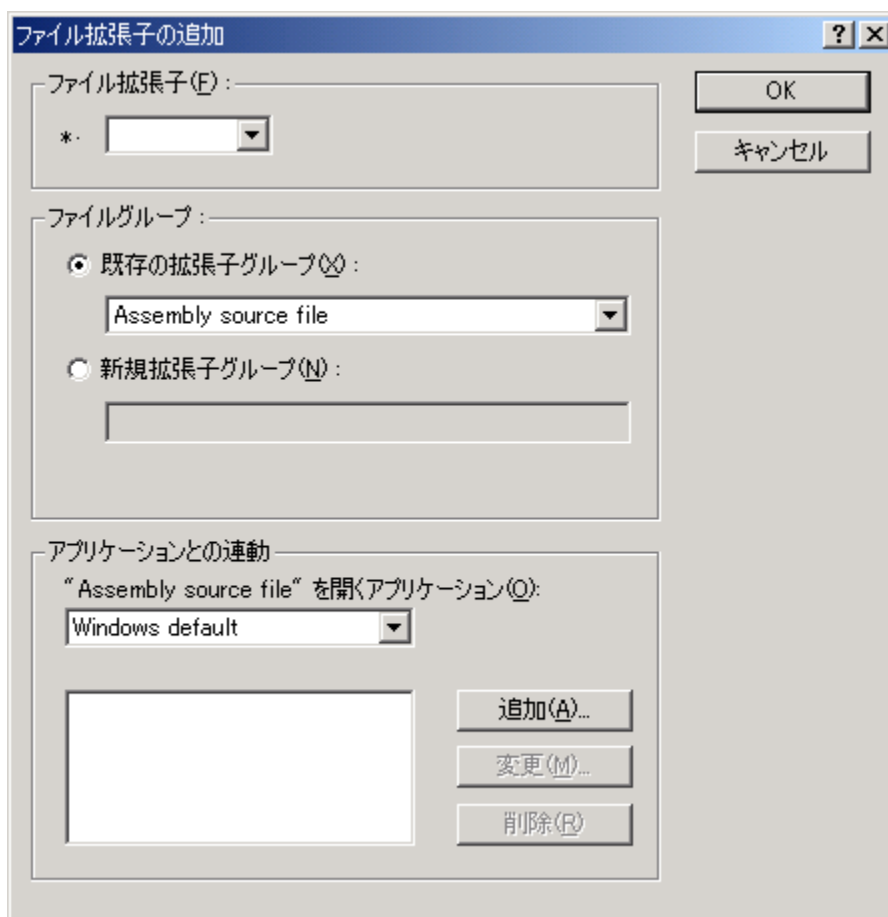


図 2.10: ファイル拡張子の追加ダイアログボックス
(既存グループ)

- [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。

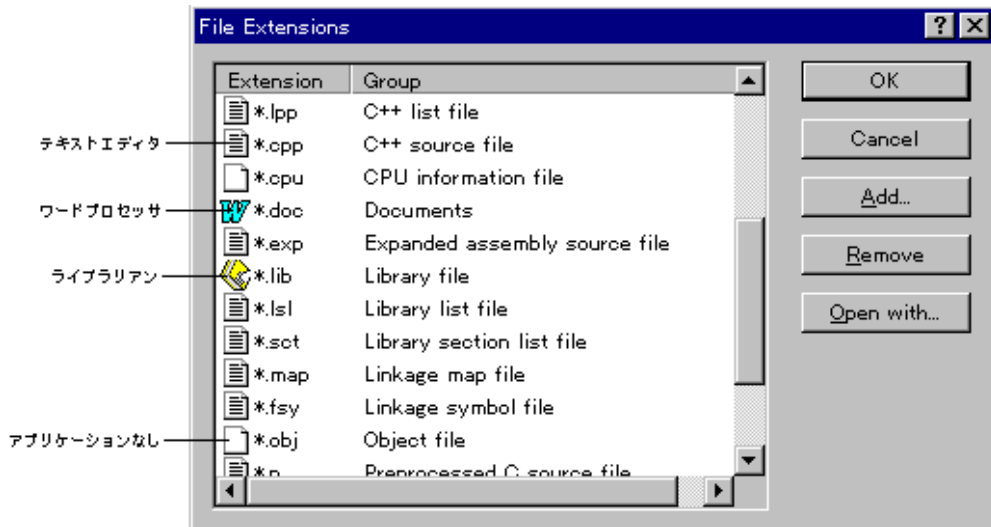


図 2.11: ファイルグループとアプリケーション

2 ビルドの基本

- 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. [で開く...]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)。

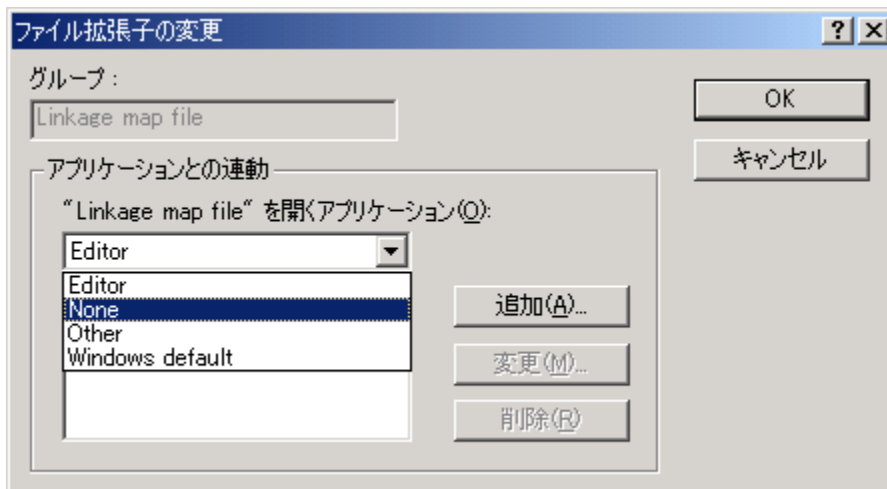


図 2.12: ファイル拡張子の変更 ダイアログボックス

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。
4. 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)。

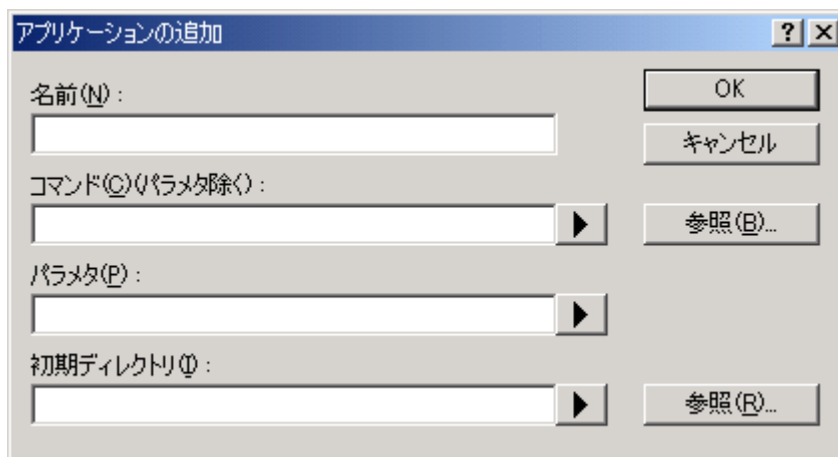


図 2.13: アプリケーションの追加 ダイアログボックス

5. [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。

6. 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。
7. [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、[Options]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。例えば、コンパイラ、アセンブラ、リンカージェディタを使う場合、それぞれに対応する 3 つのメニューオプションがあります。

☛ 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[Options]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歸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グ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「デバッグ編」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。

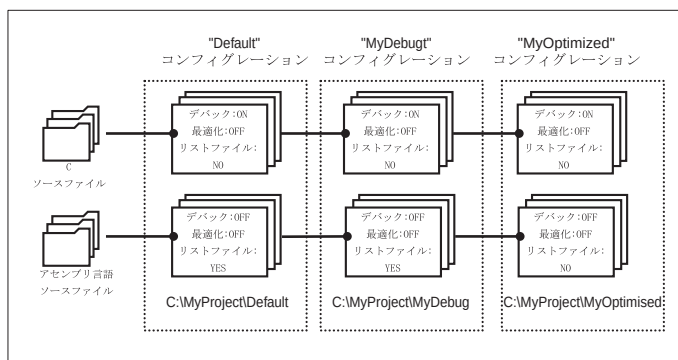


図 2.14: ビルドコンフィグレーションとファイルオプション

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設定する方法には二つあります。

☛ビルドコンフィグレーションを追加するには

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。



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）。

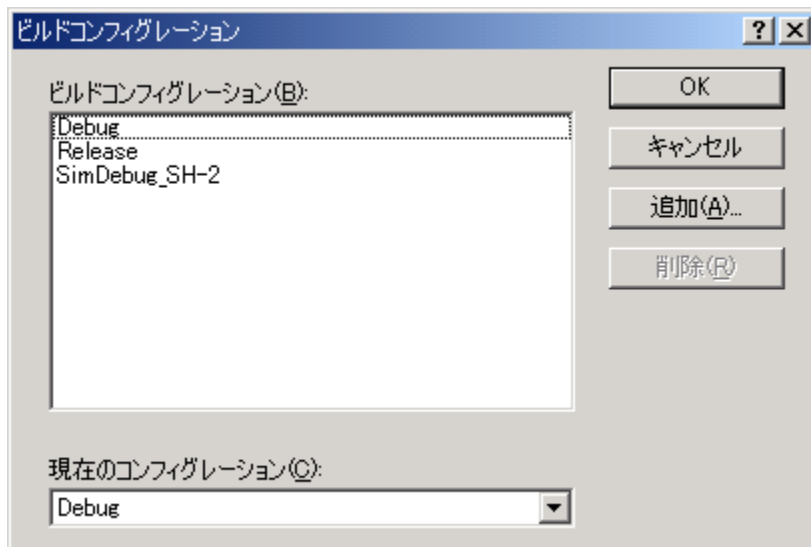


図 2.16: ビルドコンフィグレーション ダイアログボックス

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）。

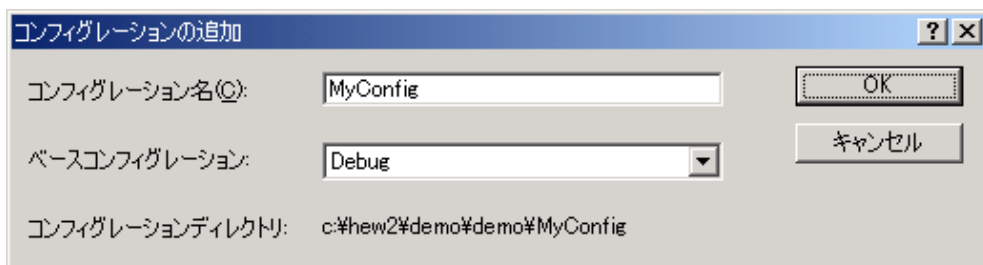


図 2.17: コンフィグレーションの追加 ダイアログボックス

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て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、[Build All]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て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。

プロジェクトファイルのどれかにエラーが起これと、“ビルド”処理も“すべてをビルド”処理も終了します。

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て、ポップアップメニューから [Build <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“Build 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

HEW を使用することで、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。

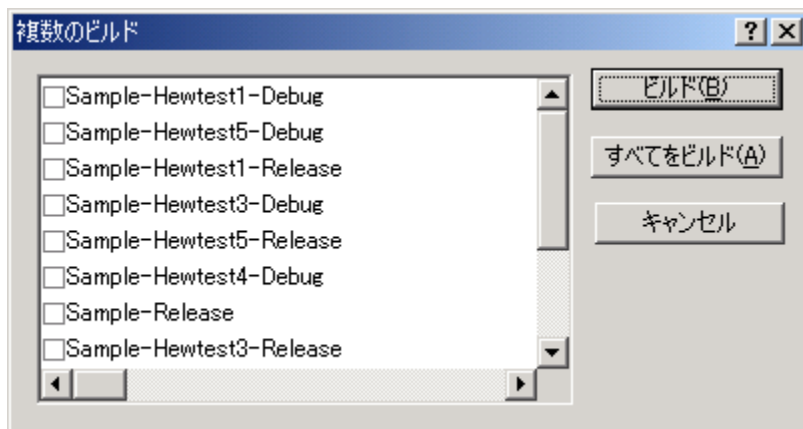


図 2.18 複数のビルドダイアログボックス

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、リンカージェディタなど）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するには、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低レベル情報（ファイルに適用したコマンドラインオプションなど）を表示すると便利です。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。

☞ ビルド実行中に低レベル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。

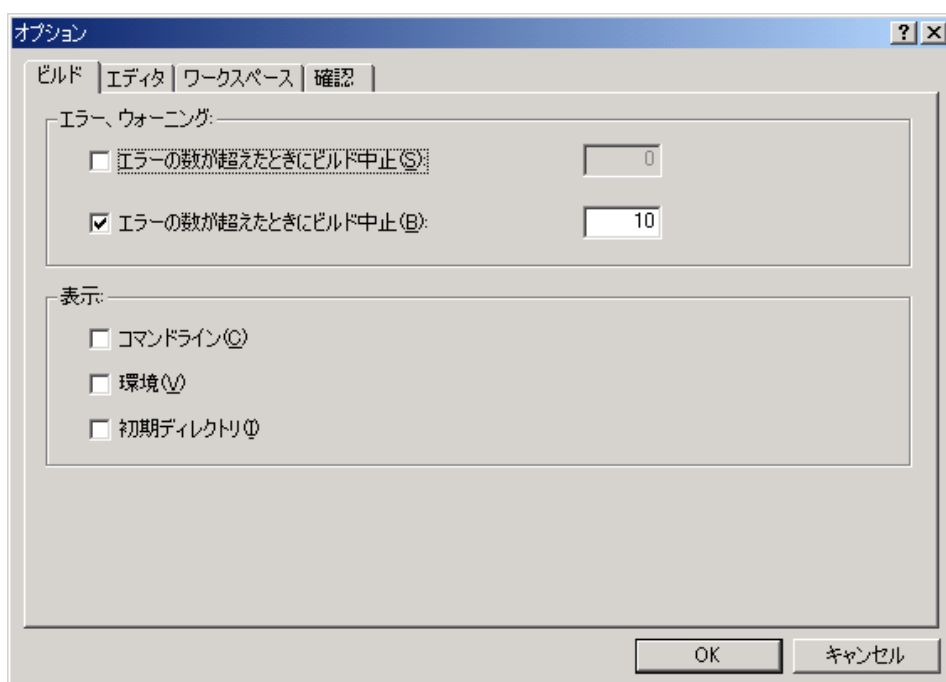


図 2.19: オプションダイアログボックスビルドタブ

2.7 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、管理が難しくなります。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て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、すべてのファイル依存は[Dependencies]フォルダに含まれています（図 2.20.(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、[Configure View]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 2.20.(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ファイル依存を示します（図 2.20.(i)）。

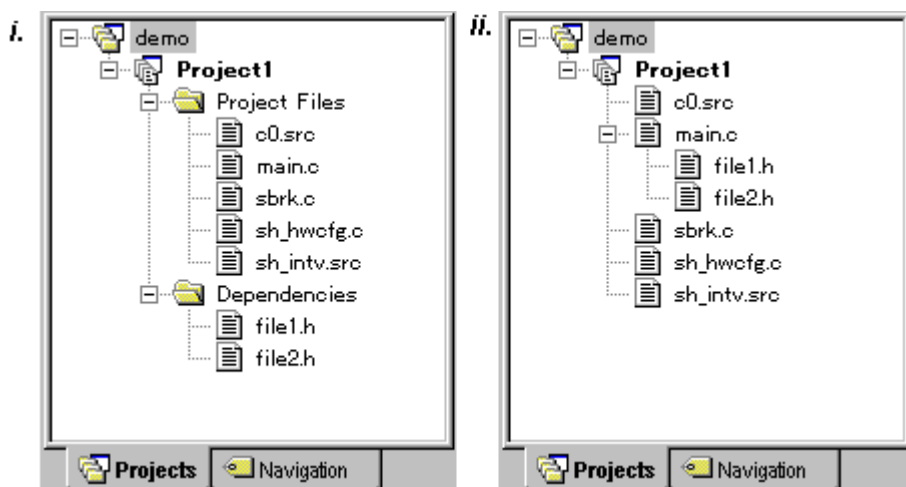


図 2.20: 各ファイルの依存

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 2.21.(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 2.21.(ii))。

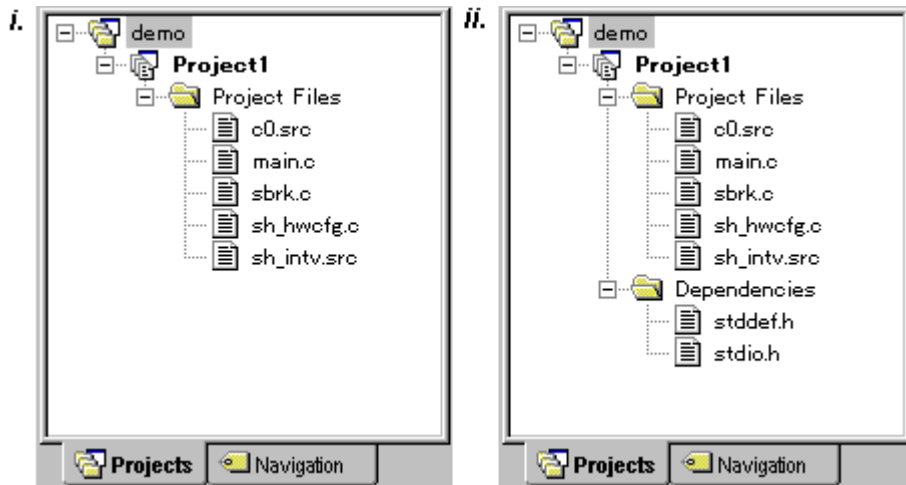


図 2.21: 標準ライブラリファイルのインクルード

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 2.22）。

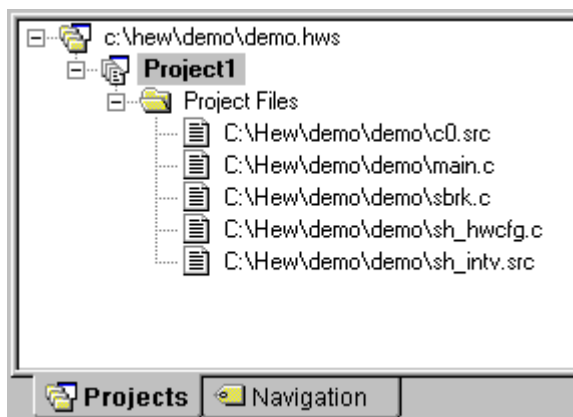


図 2.22: ファイルのパスの表示

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] タブからプロジェクトを選んでください。
2. 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 **[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. 非アクティブのプロジェクトを **[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 2.23)。
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。

注意 新しいプロジェクトを作成するとき、CPU やツールチェーンは指定できません。これらのプロパティはワークスペースにより定義されています。つまり、同じワークスペース内のすべてのプロジェクトは同じ CPU とツールチェーンであることになります。

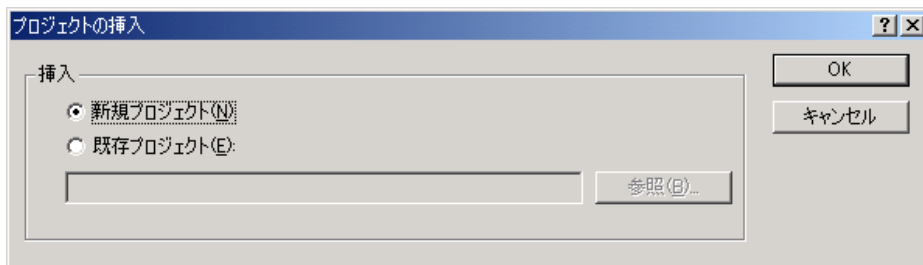


図 2.23: プロジェクトの挿入ダイアログボックス

☞ 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。

注意 既存のプロジェクトをワークスペースに追加するとき、プロジェクトの CPU とツールチェーンがワークスペースのものと一致しなければなりません。一致しない場合、そのプロジェクトはワークスペースに追加できません。

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便利です。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作るには

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 2.24）。
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。

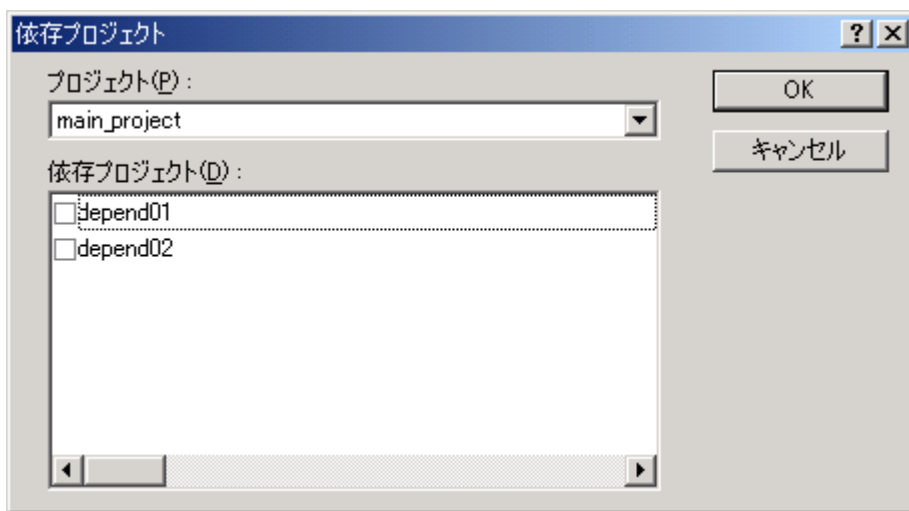


図 2.24: 依存プロジェクトダイアログボックス

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。
または
- 1. ワークスペースウィンドウの [Projects] タブからプロジェクトを選んでください。
- 2. **DEL** キーを押してください。
- 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現在使用中のプロジェクトを削除することはできません。

2.13 ワークスペースからプロジェクトをロード、アンロードする

- ☛ ワークスペースに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。
- ☛ ワークスペースに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。これは、プロジェクトを個別にロードするよりもさらに効率的です。

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2.25）。
4. [OK]ボタンをクリックしてください。

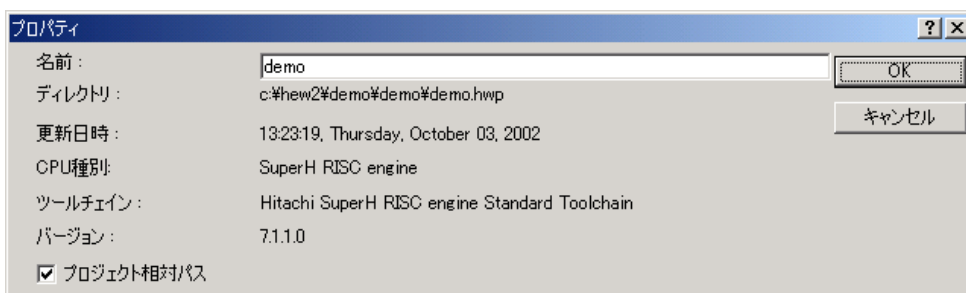


図 2.25: プロパティダイアログボックス

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、リンカージェ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必要があります。

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 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。

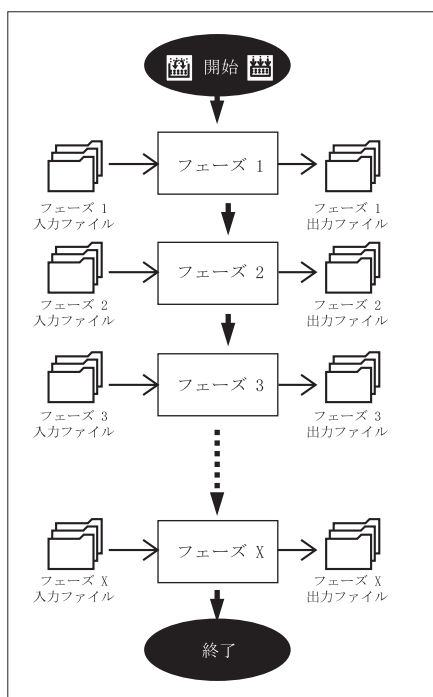


図 3.1: ビルド実行

3 ビルドの応用

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。

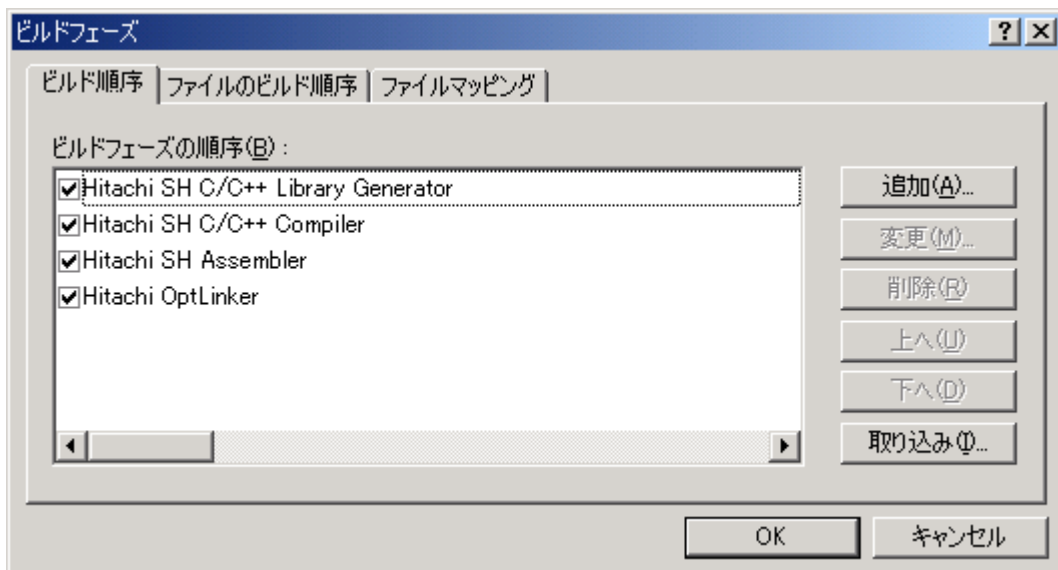


図 3.2: ビルドフェーズ ダイアログボックス

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で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ーン（コンパイラ、アセンブラ、リンカージェディタ、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。

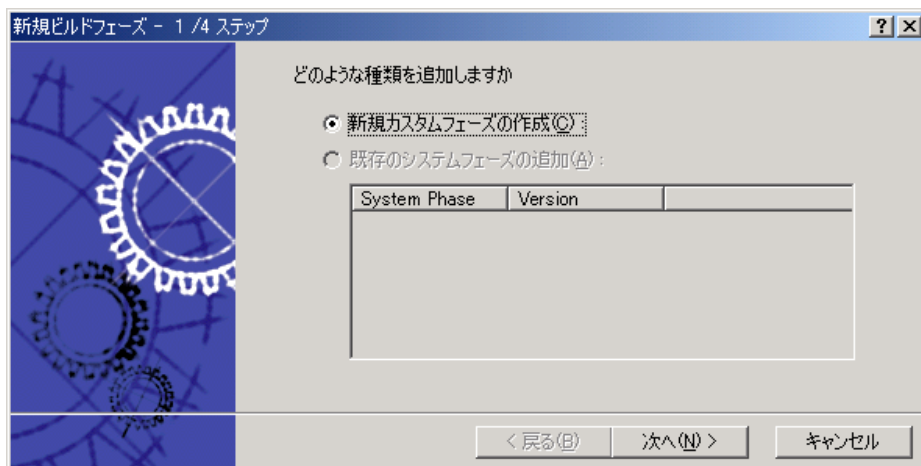


図 3.3a: 新規ビルドフェーズ ダイアログボックス (ステップ 1)

3 ビルドの応用

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。

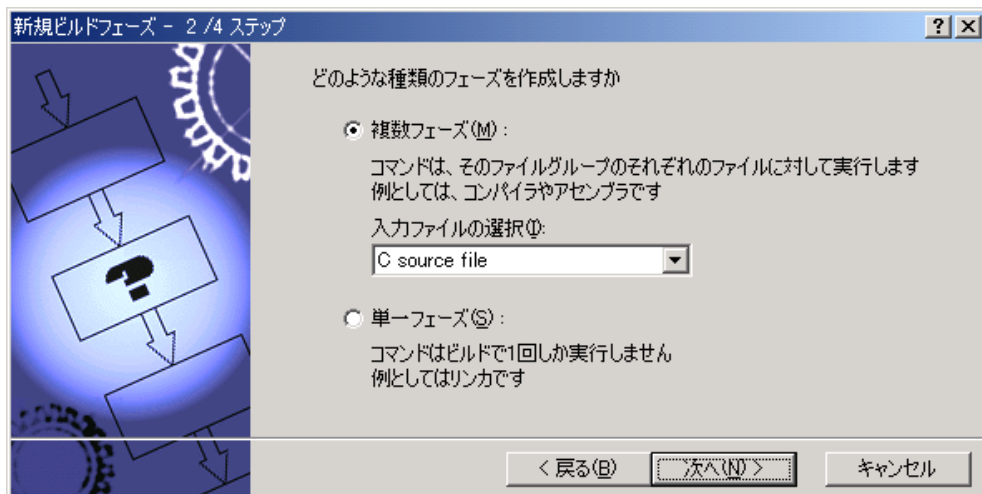


図 3.3b: 新規ビルドフェーズ ダイアログボックス (ステップ 2)

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。

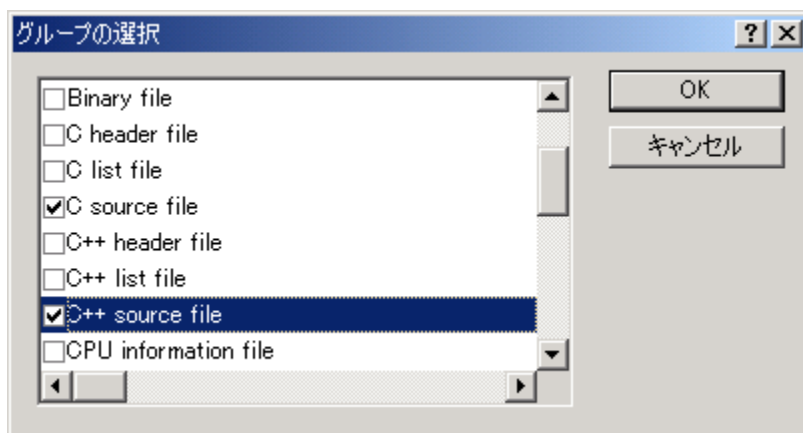


図 3.3c: 複数の入力ファイルグループの変更

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。

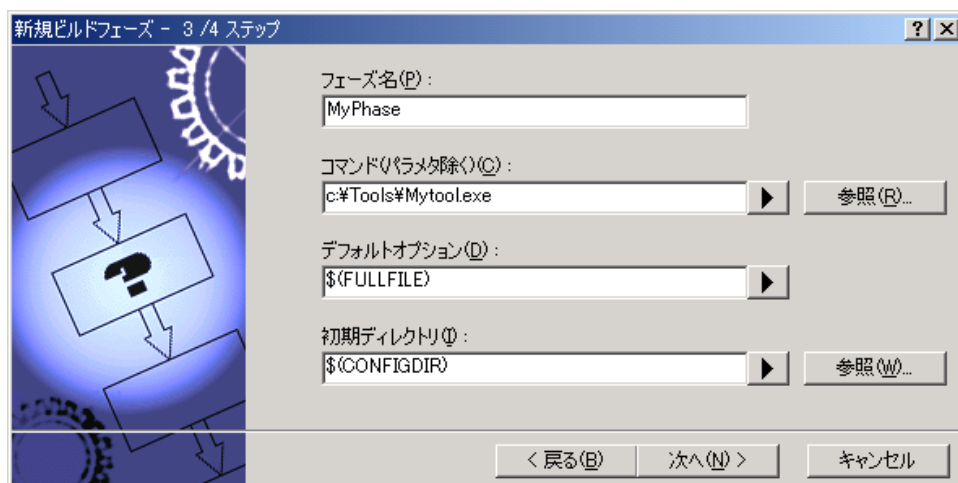


図 3.3d: 新規ビルドフェーズ ダイアログボックス (ステップ 3)

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。

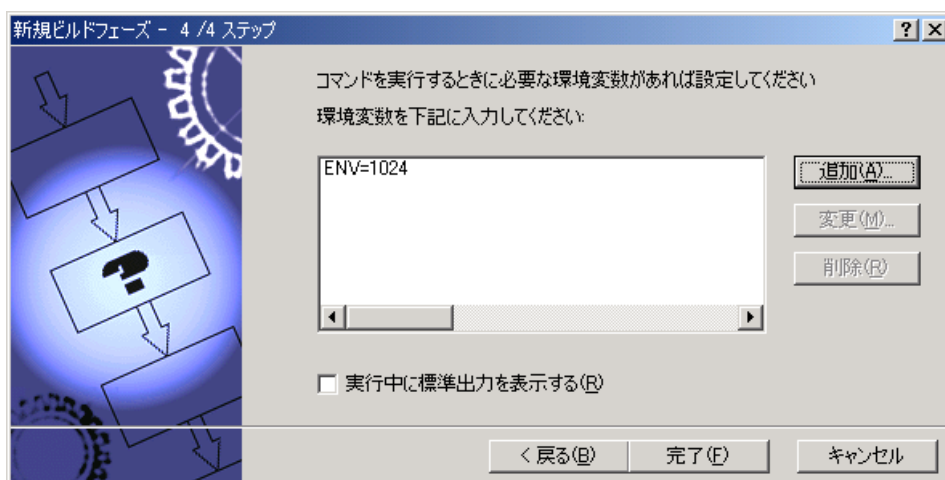


図 3.3e: 新規ビルドフェーズ ダイアログボックス (ステップ 4)

3 ビルドの応用

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。

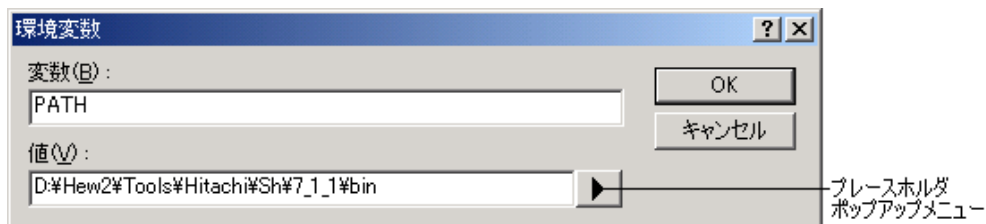


図 3.4: 環境変数 ダイアログボックス

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出力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下さい。

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 の標準的ビルドでは、①コンパイラの前、②アセンブラの前、③リンカージェディタの前、④リンカージェディタの後、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。

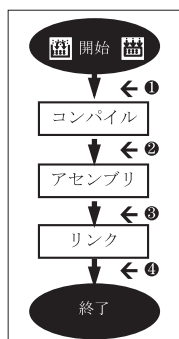


図 3.5: 標準的ビルド処理

[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。

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。

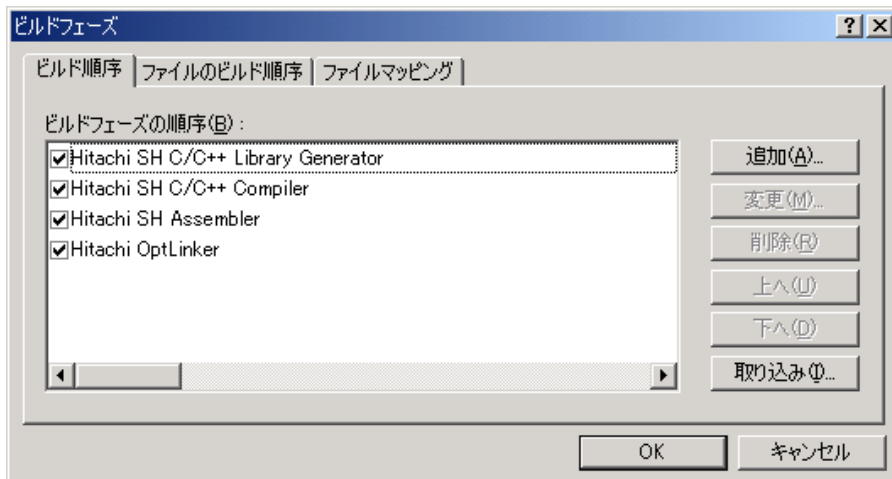


図 3.6: ビルドフェーズ ダイアログボックス ビルド順序タブ

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。
- 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。
- 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. フェーズ名を選択し、プロパティをクリックすると、カスタムフェーズの詳細が表示されます。そのフェーズが必要な機能を果たすかどうか、これを見て決定することができます。
 4. 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。

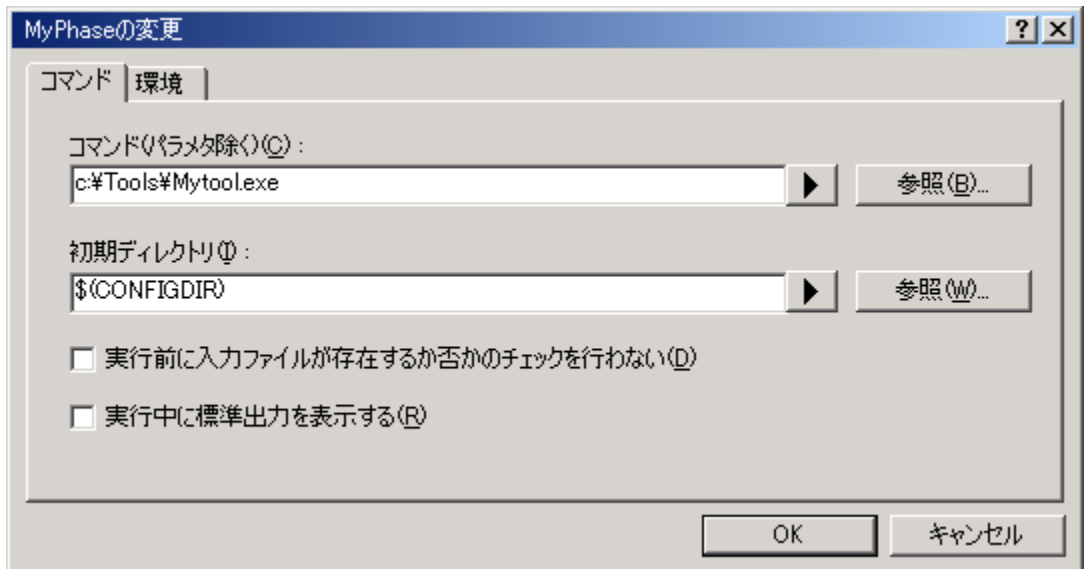


図 3.7: MyPhase の変更 ダイアログボックスコマンド タブ

- 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...] 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。

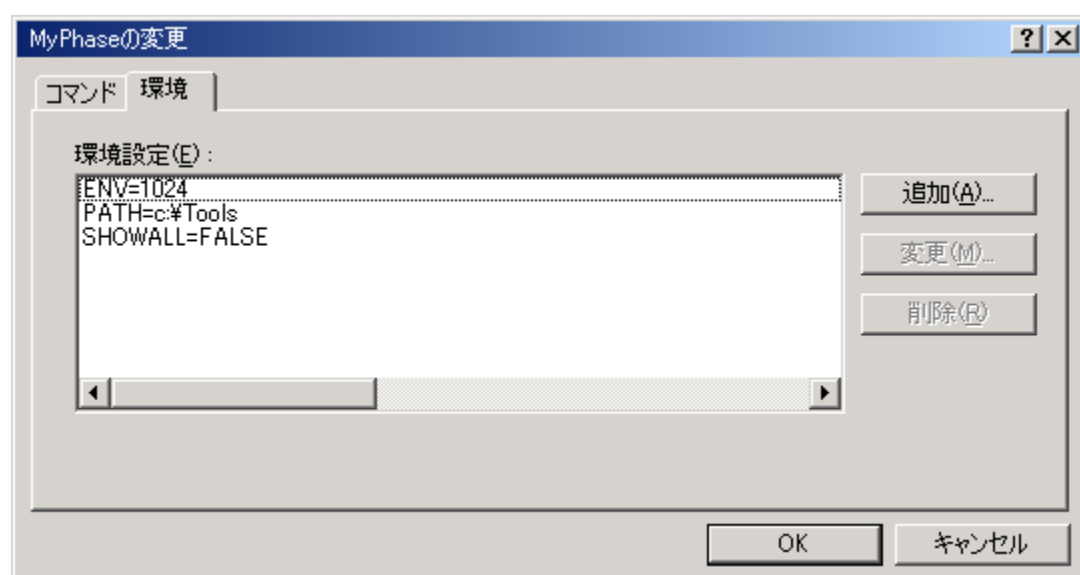


図 3.8: MyPhase の変更 ダイアログボックス環境 タブ

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。

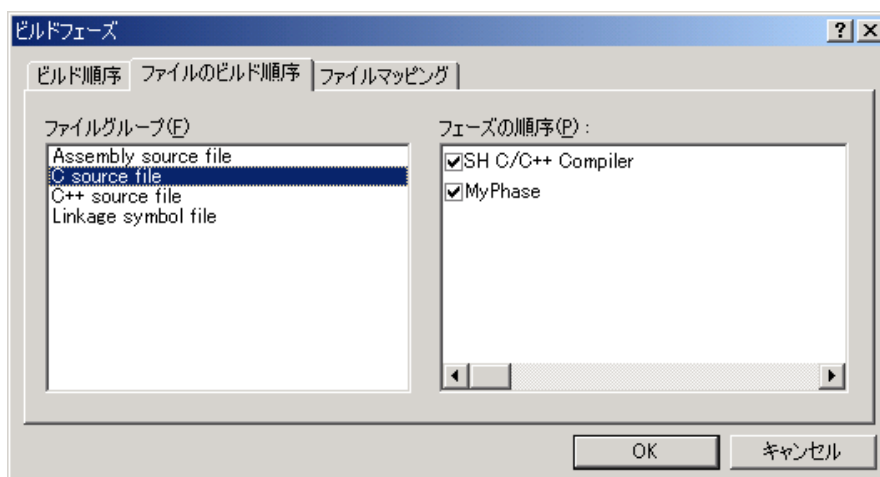


図 3.9: ビルドフェーズ ダイアログボックス ファイルのビルド順序 タブ

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られています。

[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シングルフェーズかマルチプル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）。

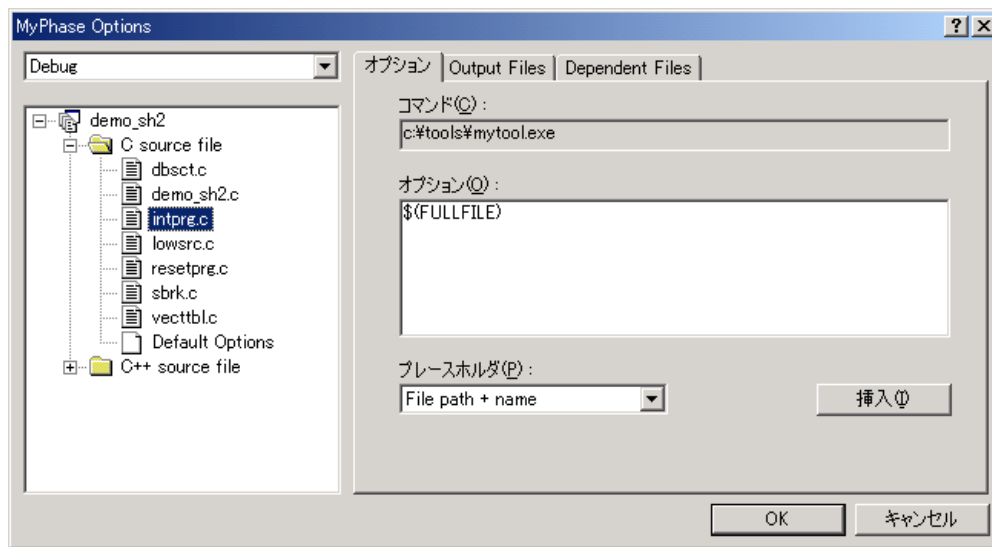


図 3.10: カスタムフェーズのオプションダイアログボックス（複数フェーズ）

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트리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。

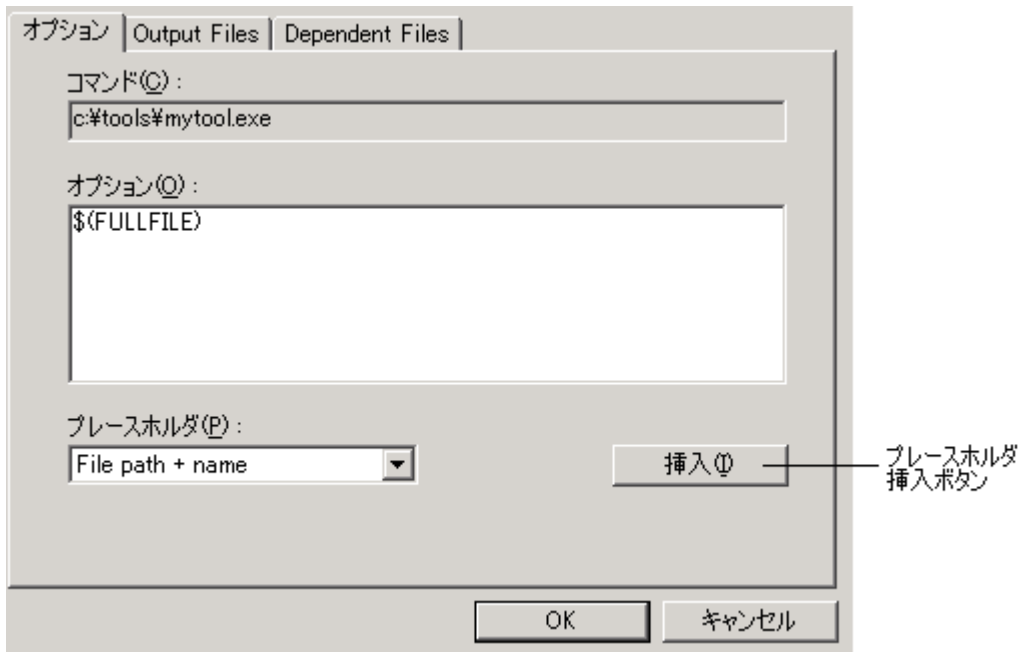


図 3.11: カスタムフェーズのオプション オプション タブ

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。

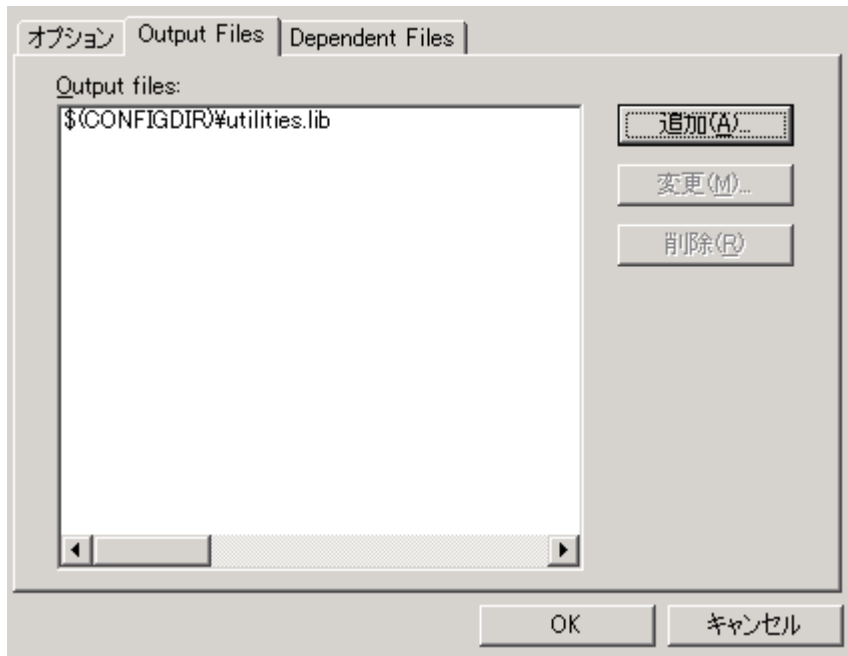


図 3.12: カスタムフェーズのオプション Output Files タブ

- 出力ファイルを追加するには
- 1. [追加...]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...]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。

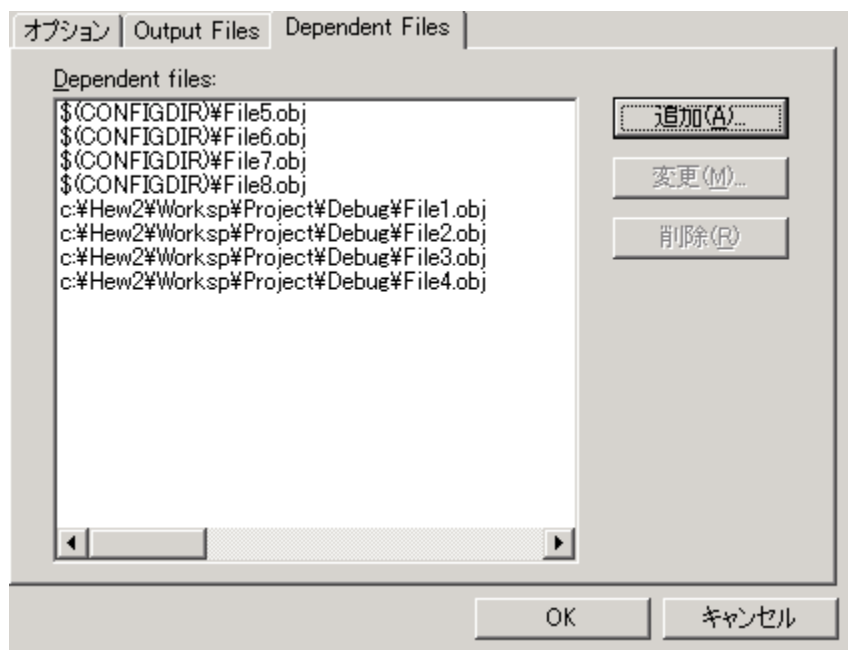


図 3.14: カスタムフェーズのオプション Dependent Files タブ

➡ 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...]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。

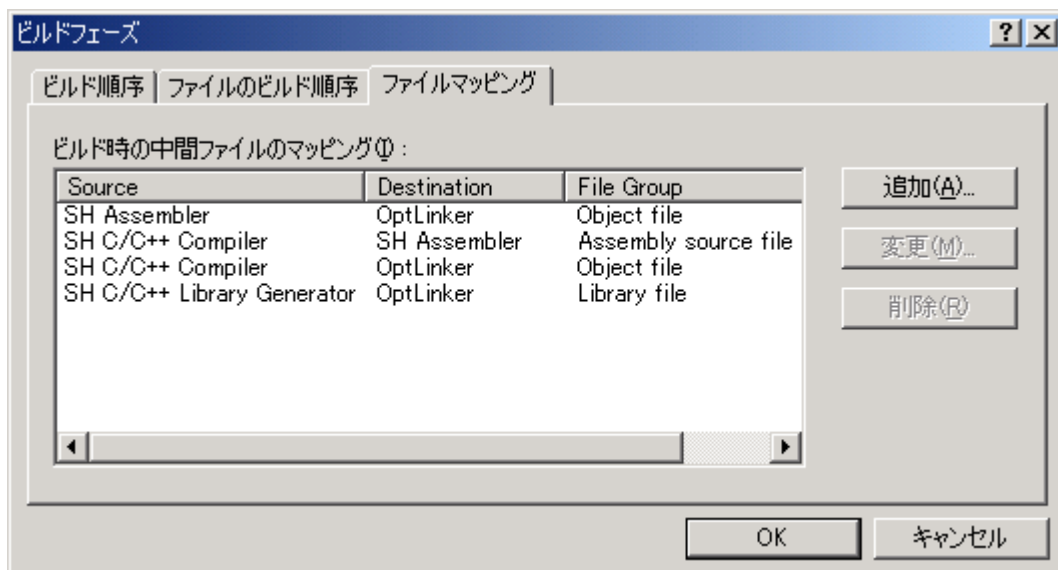


図 3.16: ビルドフェーズ ダイアログボックス ファイルマッピング タブ

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。

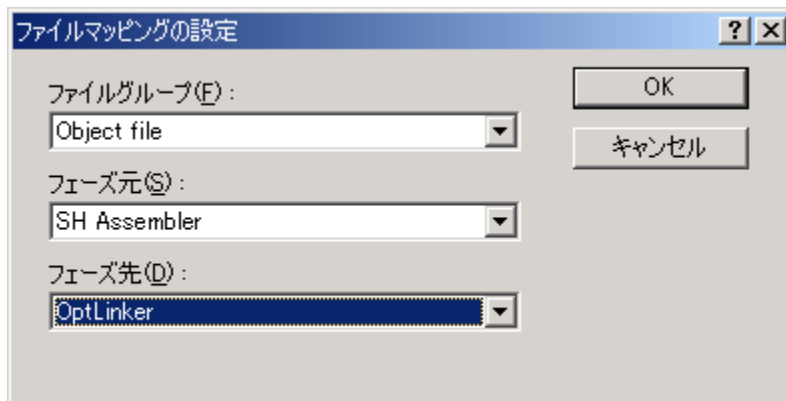


図 3.17: ファイルマッピングの設定 ダイアログボックス

- ☞ 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き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。

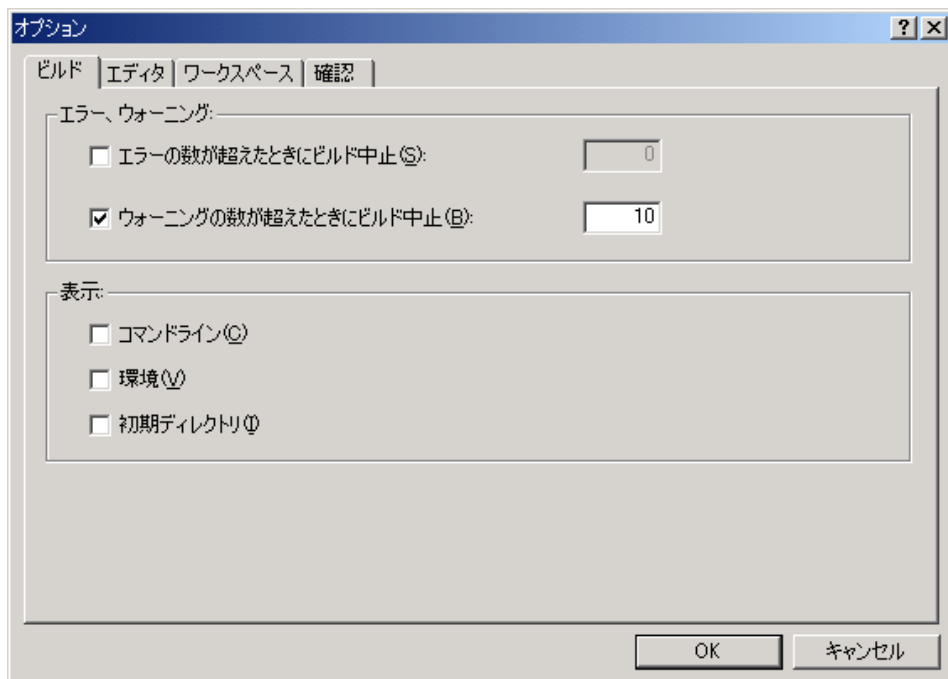


図 3.18: オプションダイアログボックス ビルド タブ

[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。ある一定の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。

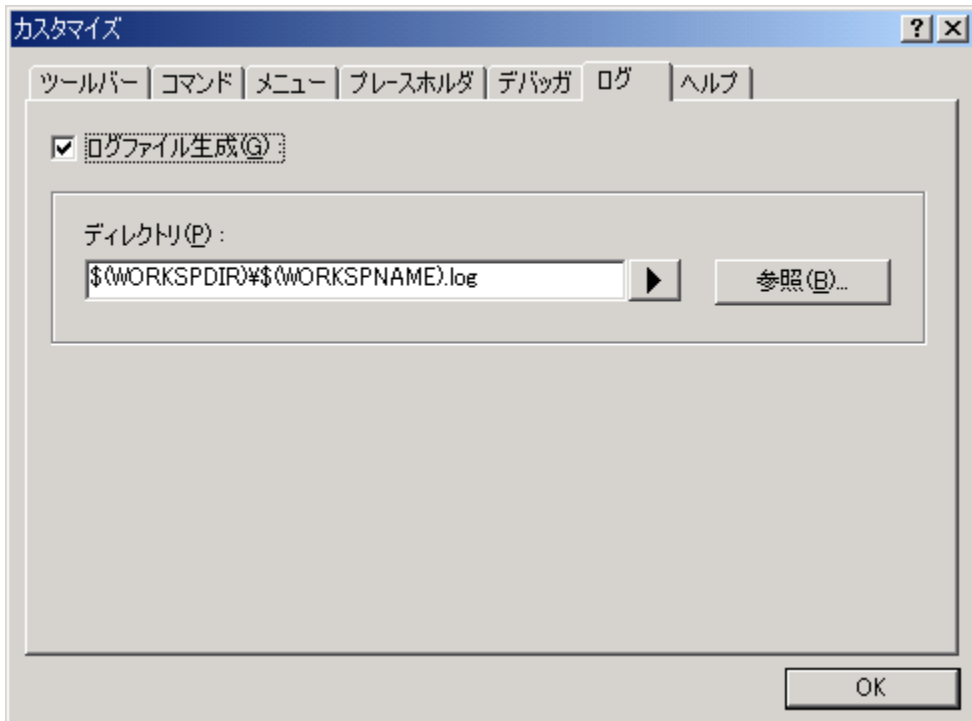


図 3.19: カスタマイズダイアログボックス ログ タブ

3.8 ツールチェーンのバージョンを変更する

同じツールチェーンの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 [ツールチェー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->ツールチェー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[使用可能なバージョン]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。

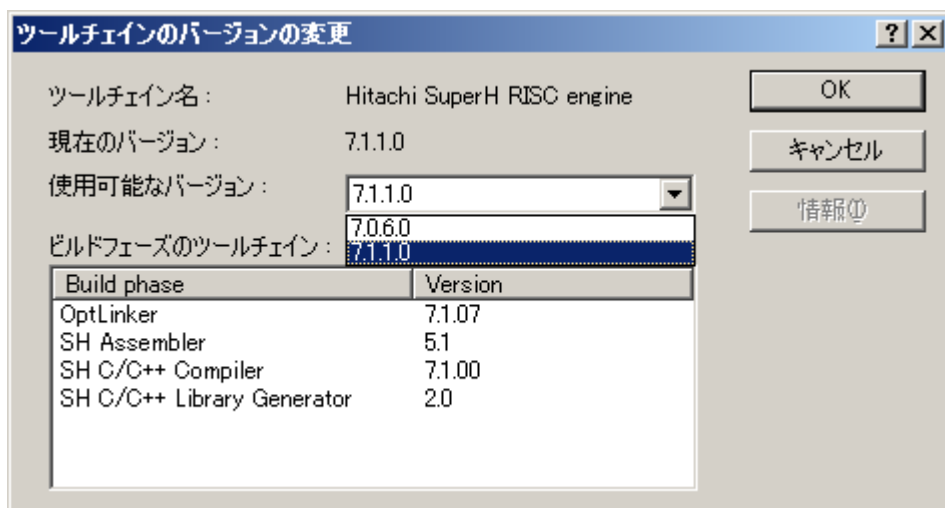


図 3.20: ツールチェーンのバージョンの変更 ダイアログボックス

ツールチェーンを構成するツールの情報を表示するには[ツールチェーンのバージョンの変更]ダイアログボックスの[ビルドフェーズノツールチェーン]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[Close]ボタンをクリックしてください。

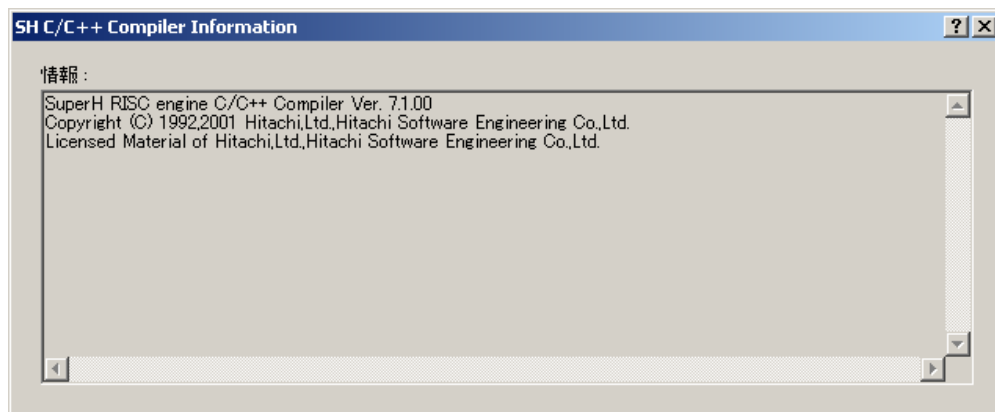


図 3.21: ツール情報 ダイアログボックス

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。もし他のデバッガを使いたい場合は、[ツール]メニューに加えなければいけません。詳細は、6 章「環境のカスタマイズ」を参照してください。

[カスタマイズ] ダイアログボックス (図 3.22) の [デバッガ] タブに HDI に関連する情報を設定します。一部のターゲットが現在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。

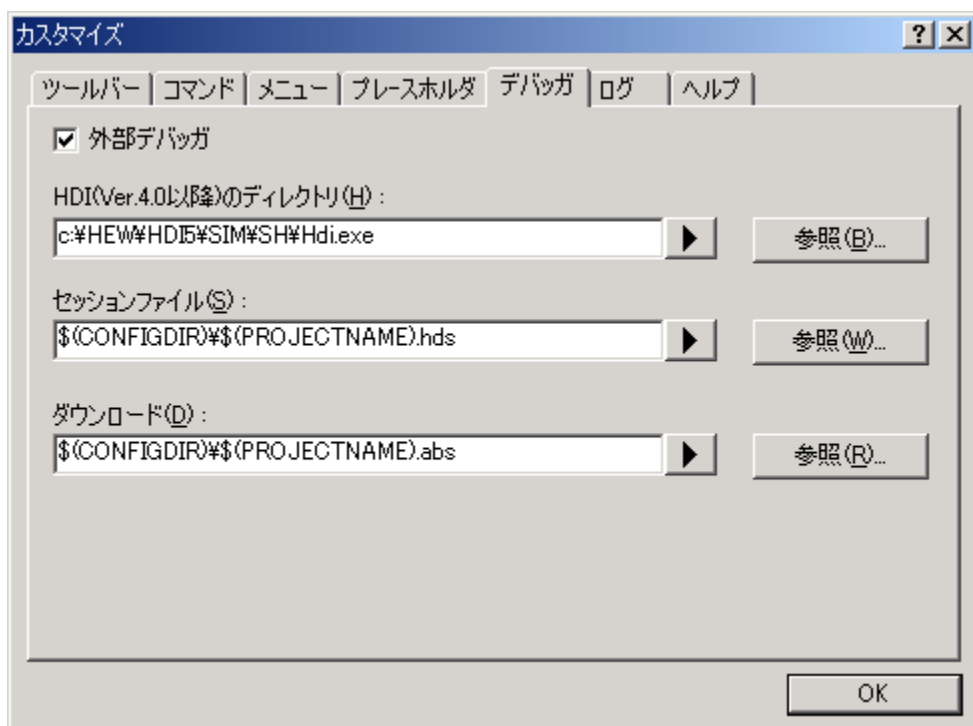


図 3.22: カスタマイズ ダイアログボックス デバッガ タブ

外部デバッガを使用する場合は、[外部デバッガ]をチェックしてください。そして以下の項目を設定してください。1つは HDI の実行ファイルの場所です。HDI はバージョン 4.0 以降のものを指定してください。そうでない場合、動作を保証できません。2つめはセッションファイルの情報です。HDI に切り替わったとき、どのセッションをロードするかを HDI に伝えます。もう 1つはダウンロードモジュールの場所です。これは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 が自動的に HDI に切り替わるために必要です。

指定したセッションファイルで HDI を起動するには、次に示す[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。



3 ビルドの応用

ビルド後、ダウンロードモジュールが更新された場合、HEW から HDI に戻り、すぐデバッグできるようになります。HDI 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 に戻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。

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。

- ☛ 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. このメニューを選ぶと、ダイアログボックスが表示され、ワークスペースのどの部分をメイクファイルに追加するか尋ねます（図3.23参照）。
- 5. [Makeファイル生成前にスキャンを行う]ボックスをチェックするか、チェックを外してください。プロジェクトファイルの依存性が最新のものであることが分かっている場合、特に、ワークスペースに複数のプロジェクトがある場合は、チェックの切り替えが早く、より効率的です。依存性によって切り替わる場合、メイクファイルを含むすべてのプロジェクトがスキャンされます。
- 6. 手持ちのメイクファイルに対応するラジオボタンを選択し、[OK]ボタンをクリックしてください。

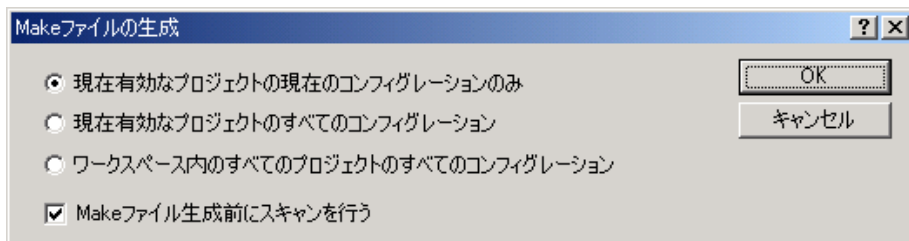


図 3.23: Make ファイルの生成 ダイアログボックス

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。

- ☛ 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”ディレクトリに移行してください。
- 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、失敗する可能性が高いでしょう。一般的に、できるだけブ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。

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“dbstct.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。

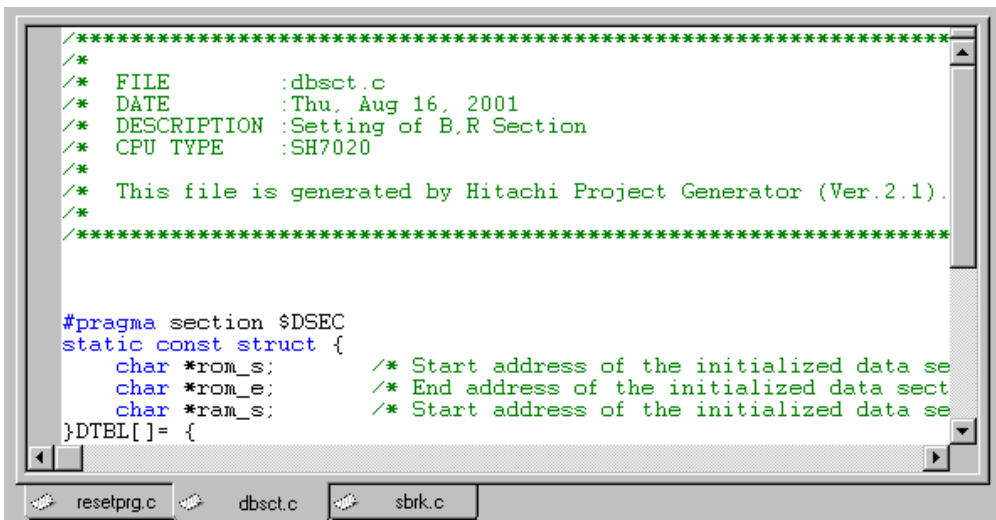


図 4.1: エディタウィンドウ

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

クリックすると、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

クリックすると、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”をクリックします。



ファイルの保存

クリックすると、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

クリックすると、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

クリックすると、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

クリックすると、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

クリックすると、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コピーします。



検索

クリックすると、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。



ファイル間検索

クリックすると、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。



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

クリックすると、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

複数のファイルにおいて文字列を検索します。[Output] ウィンドウの[Find in Files] タブに、検索結果を表示します。詳細は、この章の後半の「検索とファイル内の移動」を参照してください。



検索

クリックすると、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] をクリックしてください。

開いた最新のファイル5つは[File]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かない）	[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）。

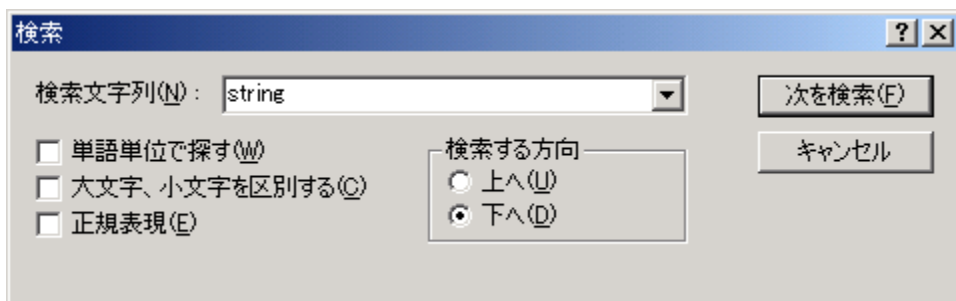


図 4.2: 検索 ダイアログボックス

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[Find]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]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)。

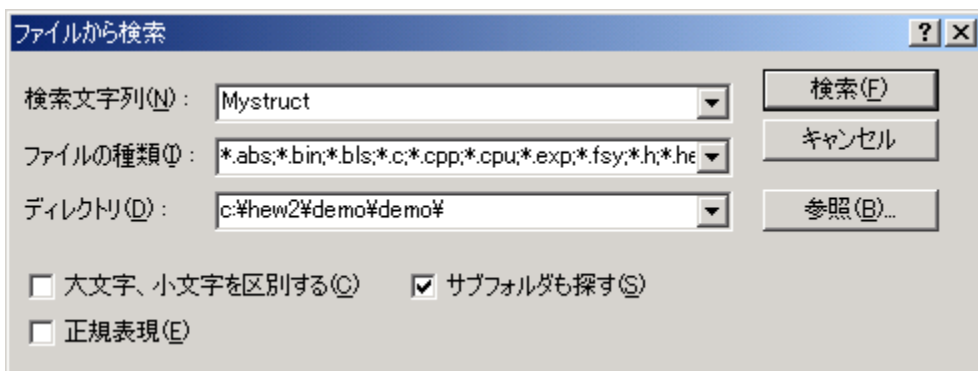


図 4.3: ファイルから検索 ダイアログボックス

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、コンマ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。

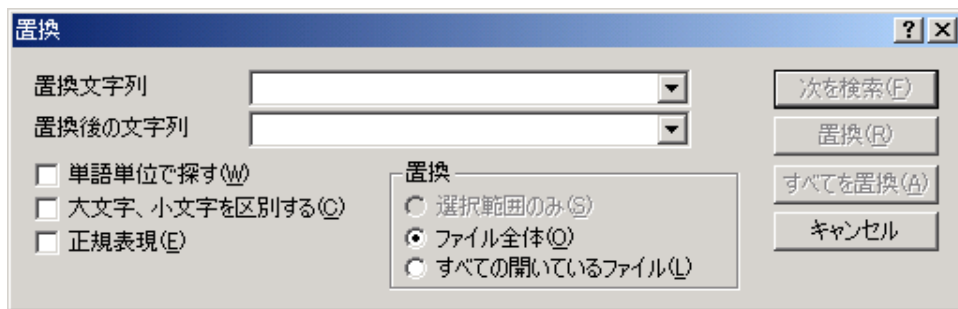


図 4.4: 置換 ダイアログボックス

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。

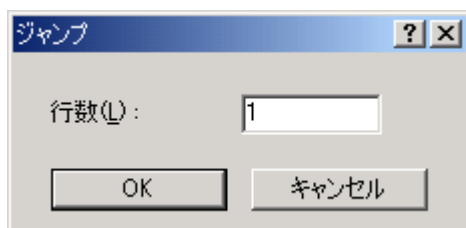


図 4.5: ジャンプダイアログボックス

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、解除するまでか、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。

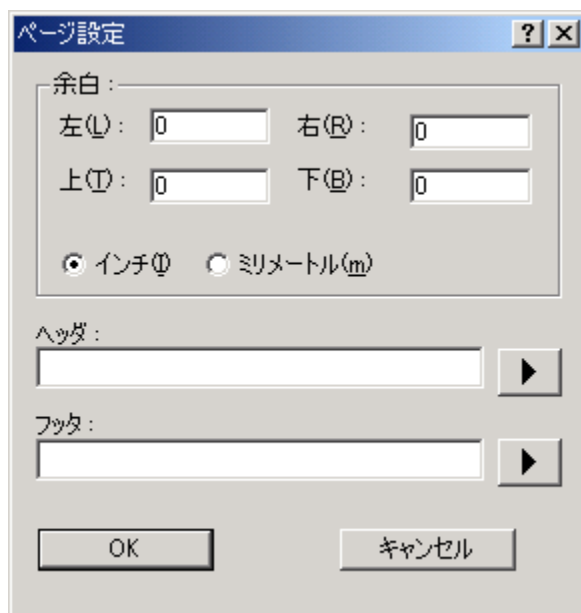


図 4.6: ページ設定 ダイアログボックス

☛ 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。

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。[Options]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。

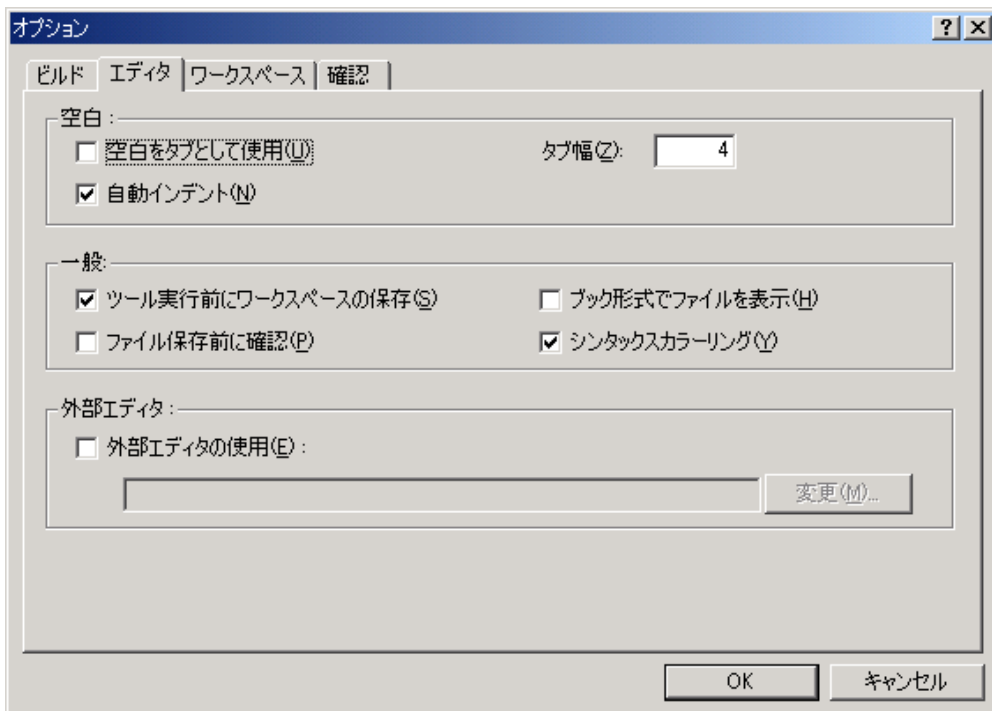


図 4.7: オプション ダイアログボックス エディタ タブ

“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。

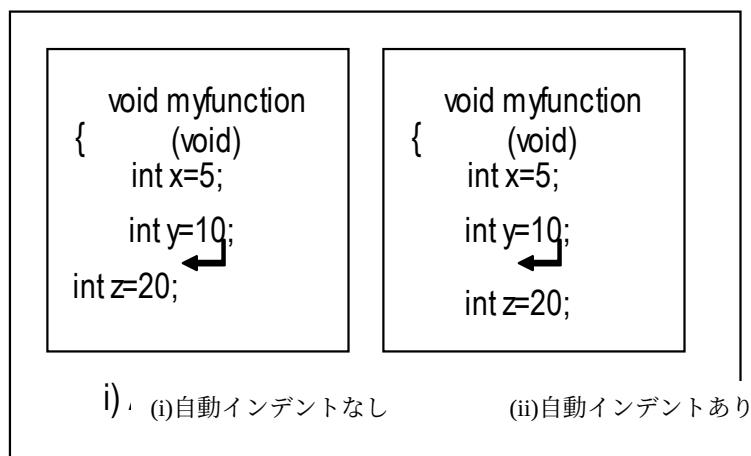


図 4.8: 自動インデント

☛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。

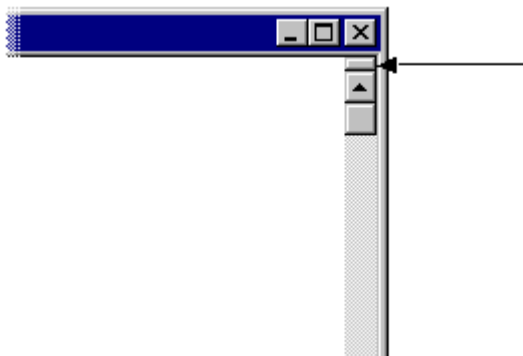


図 4.9: スプリットバー ボタン

- ➡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。

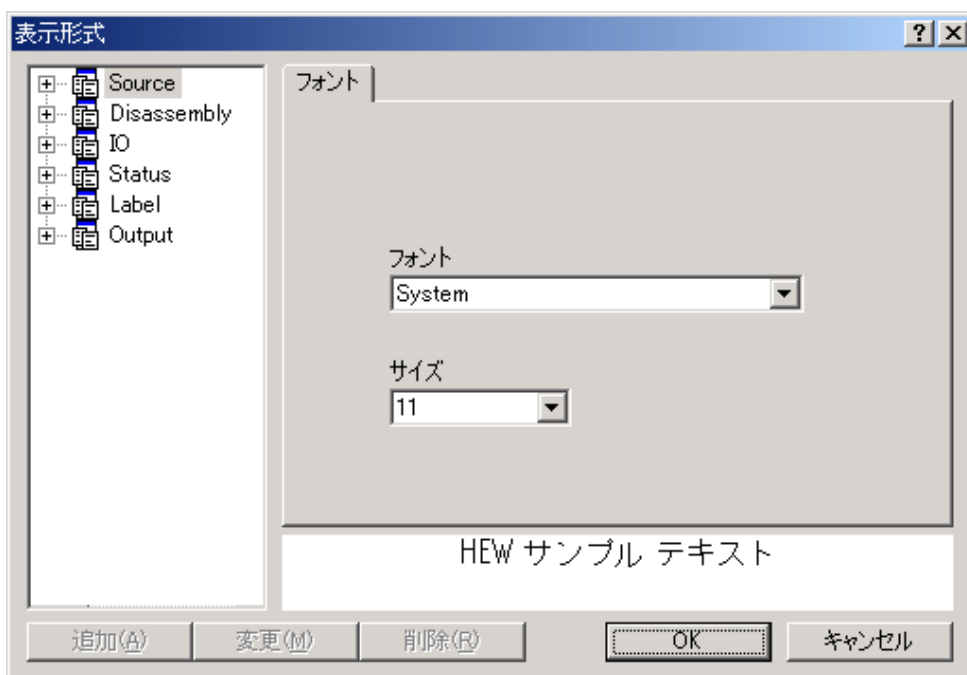


図 4.10: 表示形式 ダイアログボックス フォント タブ

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[Add Category]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: Add Category 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）。

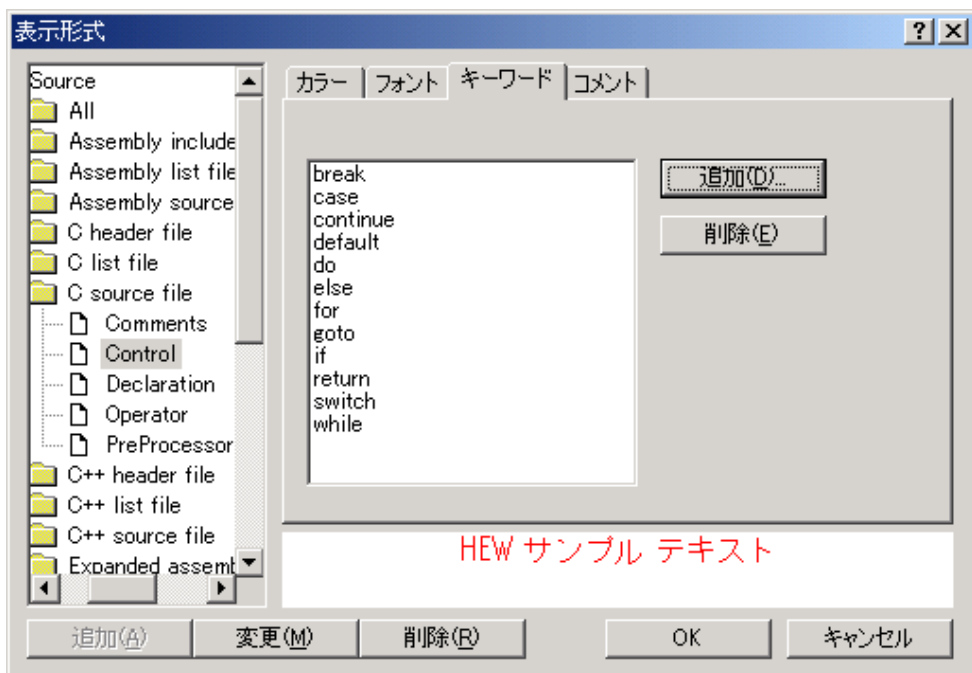


図 4.12: 表示形式ダイアログボックス キーワードタブ

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、“CTRL+T”キーを押下するか、[Define Templates...]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。
2. [追加]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. テンプレートを設定するとき、次のキーワードを使うことができます。

メニューエントリ	ブレースホルダテキスト	説明
時間	\$(TIME)	テンプレートが挿入される時間
日/月/年	\$(DATE DMY)	現在の日/月/年
月/日/年	\$(DATE MDY)	現在の月/日/年
年/月/日	\$(DATE YMD)	現在の年/月/日
テキストの年月日	\$(DATE TEXT)	現在の年月日をテキスト表示
行	\$(LINE)	テンプレートを挿入する最初の行の番号
ユーザ	\$(USER)	現在のユーザ
ファイル	\$(FULLFILE)	ファイルの名称
ファイル名	\$(FILE)	ファイルの名称、フルパス
プロジェクト名	\$(PROJNAME)	現在のプロジェクトの名称
ワークスペース名	\$(WORKSPNAME)	ワークスペースの名称
カーソル位置	\$(^)	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

6. ‘\$(^)’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。‘\$(^)’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）。

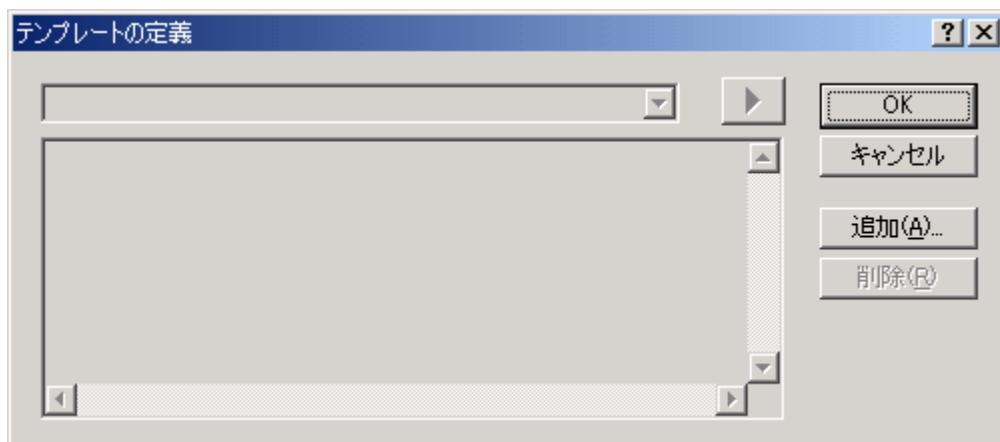


図 4.14: テンプレートの定義 ダイアログボックス

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...]を選び、”CTRL+T”キーを押すか、[Define Templates...]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. [OK]ボタンをクリックすると、テンプレートの変更が保存され、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、“CTRL+B”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。

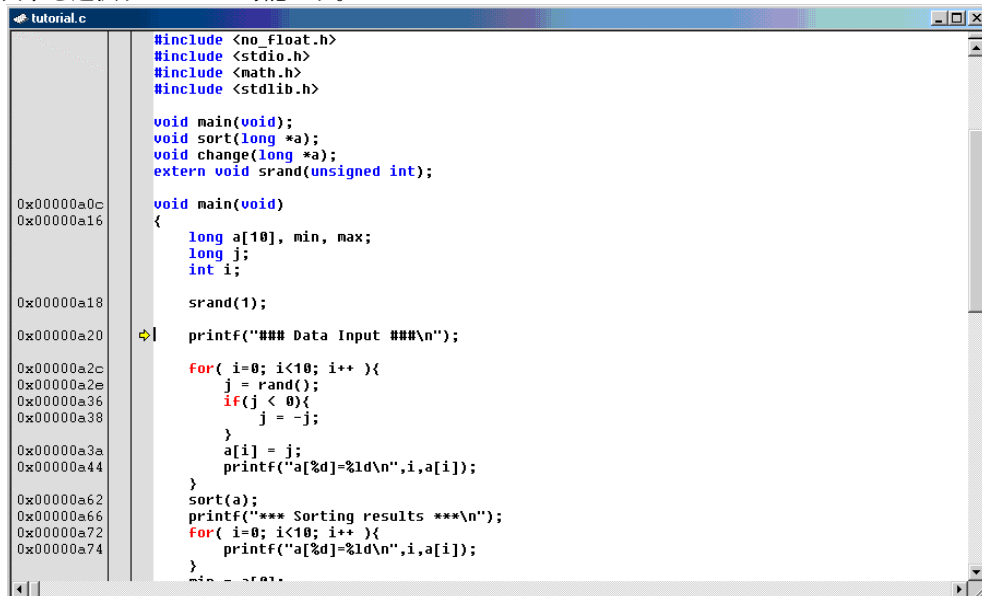


図 4.15: Editor カラム

➡すべてのソースファイルでカラムをオフにするには

1. [Editor]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。
4. [Check status]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[Editor]ウィンドウを右クリックしてください。[Editor]ポップアップメニューが表示されます。
2. [Columns]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。

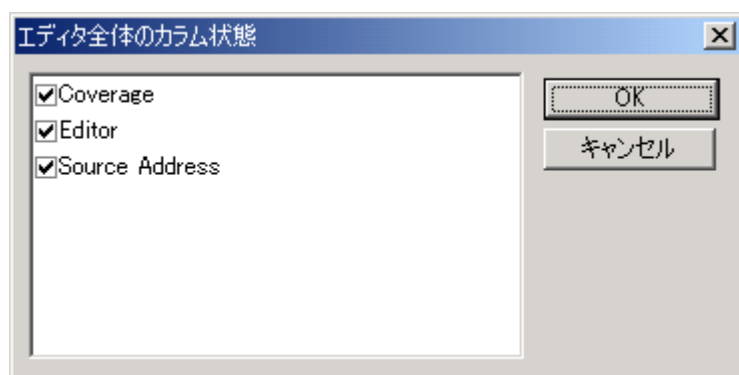


図 4.16: エディタ全体のカラム状態ダイアログボックス

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。

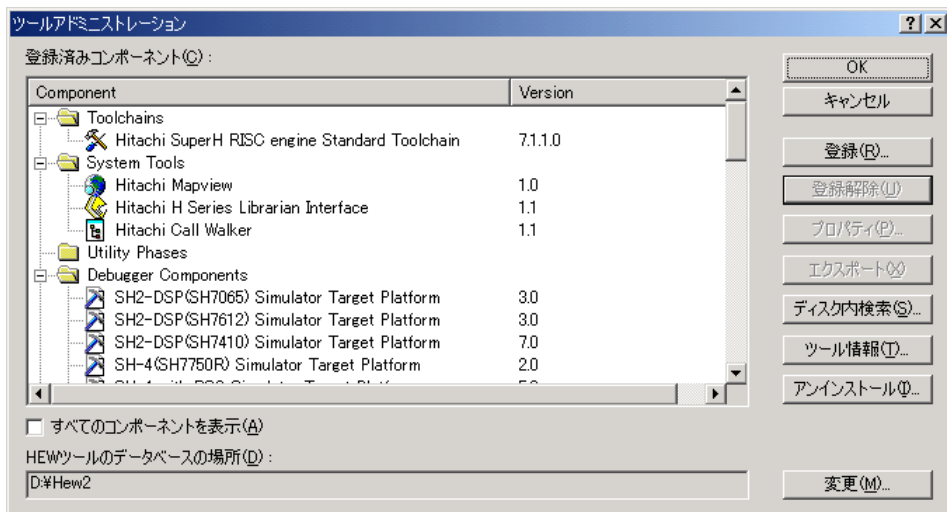


図 5.1: ツールアドミニストレーション ダイアログボックス

5 種類の標準ツールがあります。

Toolchain： 一連のビルドフェーズ（例:コンパイラ、アセンブラ、リンカージェディタ）。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）。ツールチェーンをサポートする追加のアプリケーション（例: Hi-performance Embedded Workshop 2 (HEW2) 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）。特定のツールチェー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（ツールチェーン、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 **“HEW Registration File”**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）。

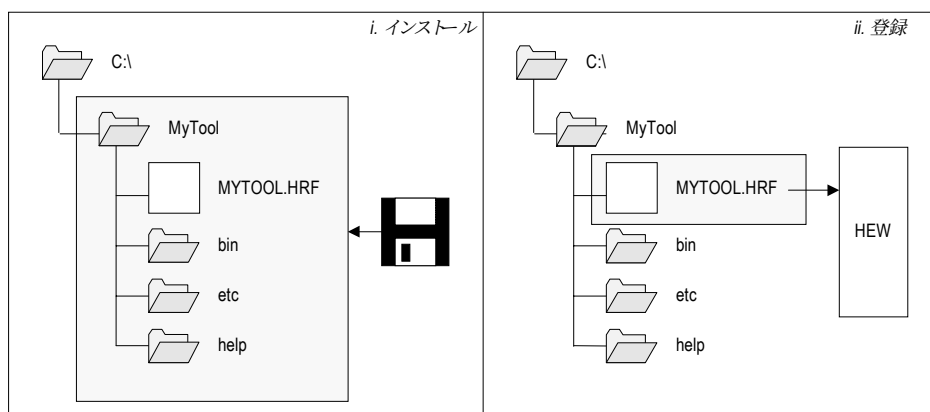


図 5.2: HRF ファイルの位置と登録

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せん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。
この場所にあるかもしれない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)。

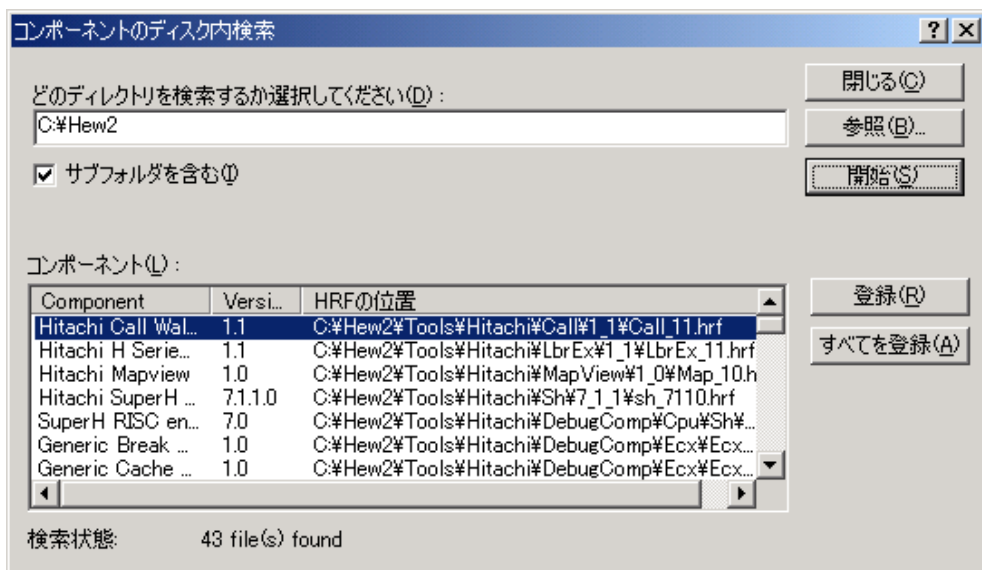


図 5.3: コンポーネントのディスク内検索 ダイアログボックス

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] 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] をクリックしてください。
6. [閉じる] 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、[Select]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]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ったり、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。

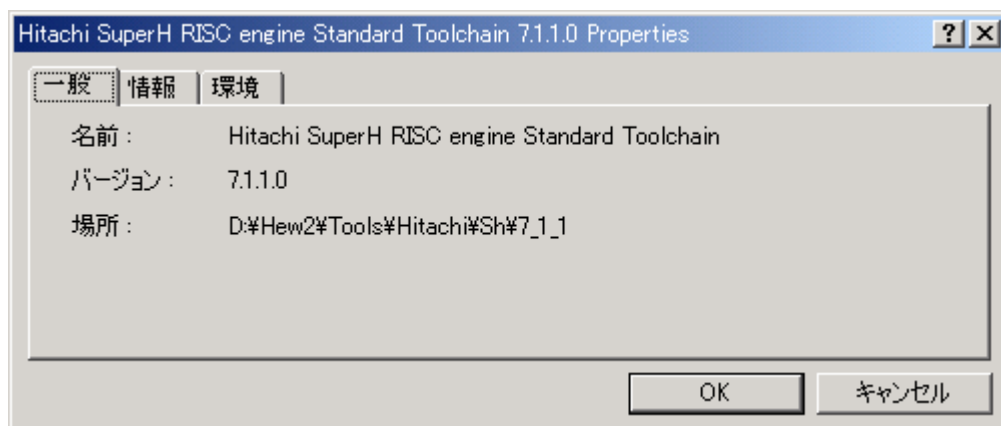


図 5.4: プロパティのダイアログボックス 一般 タブ

5 ツール管理

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。

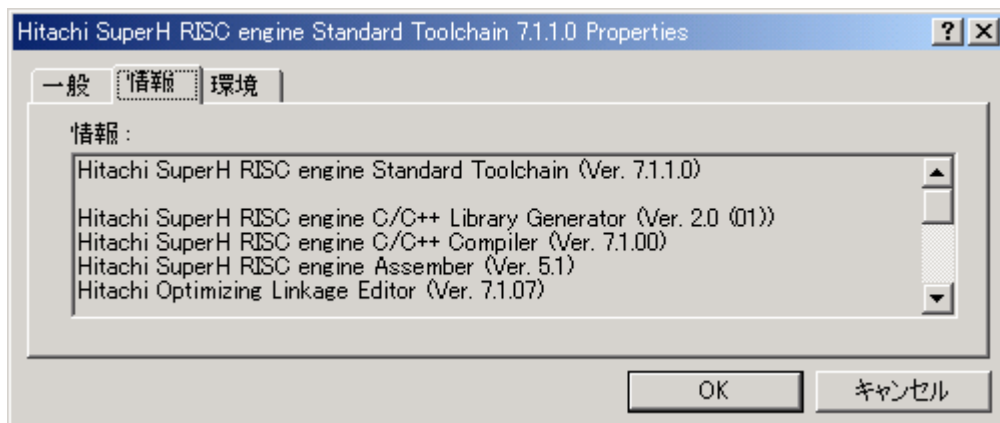


図 5.5: プロパティのダイアログボックス 情報 タブ

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、ツールチェー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。

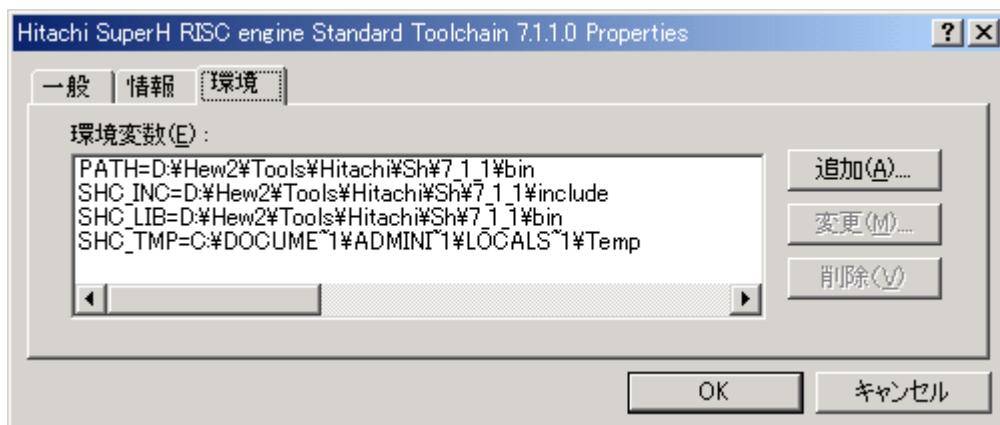


図 5.6: プロパティのダイアログボックス 環境 タブ

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。

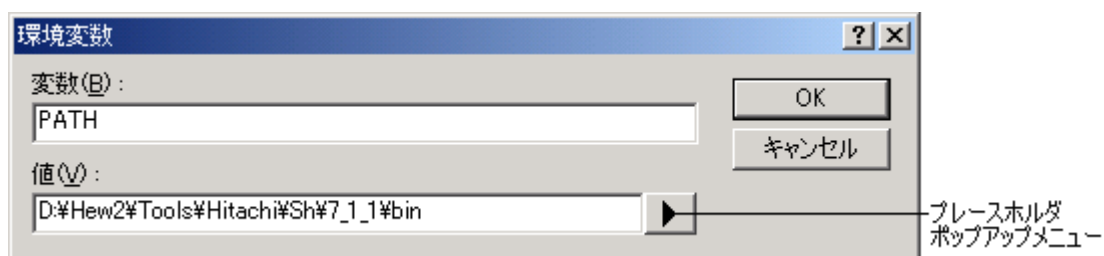


図 5.7: 環境変数 ダイアログボックス

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、内蔵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...]を選択してください。
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)。

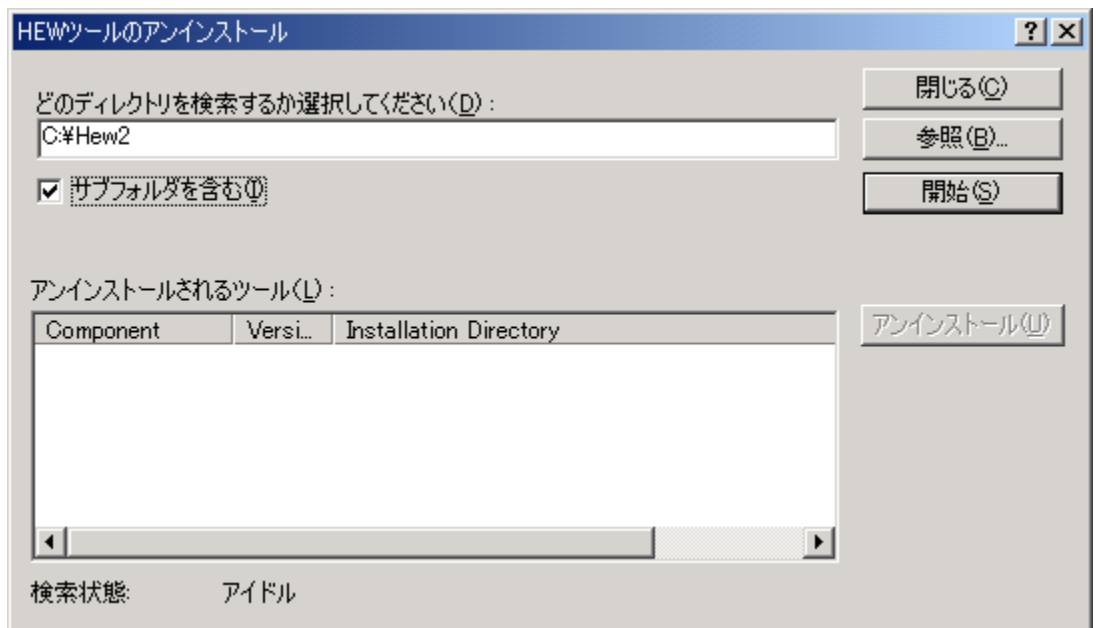


図 5.8: HEW ツールのアンインストールダイアログボックス

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] 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] 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、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。

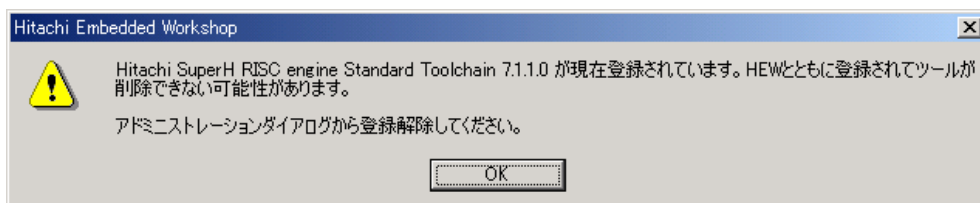


図 5.9: アンインストール不可のダイアログボックス

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。

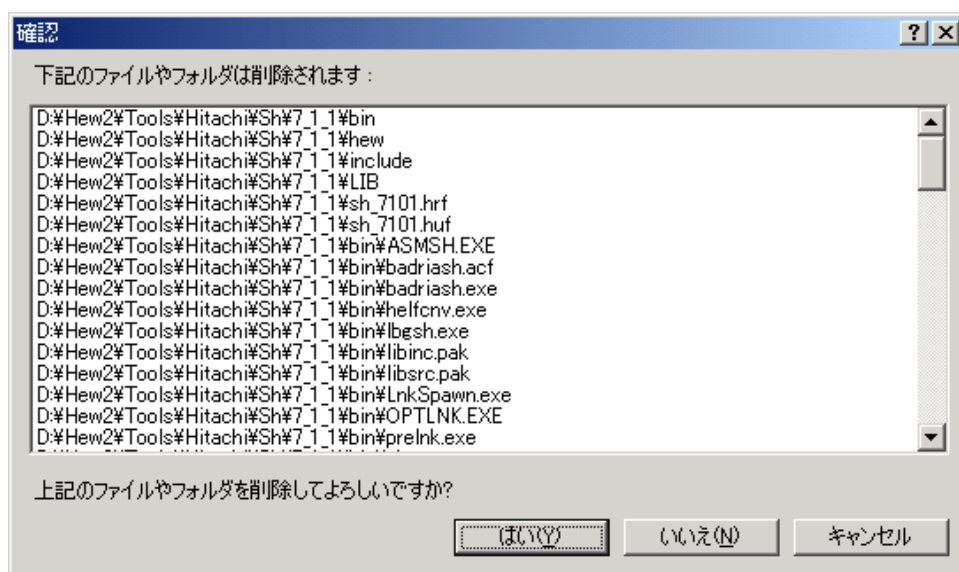


図 5.10: 確認ダイアログボックス

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）。

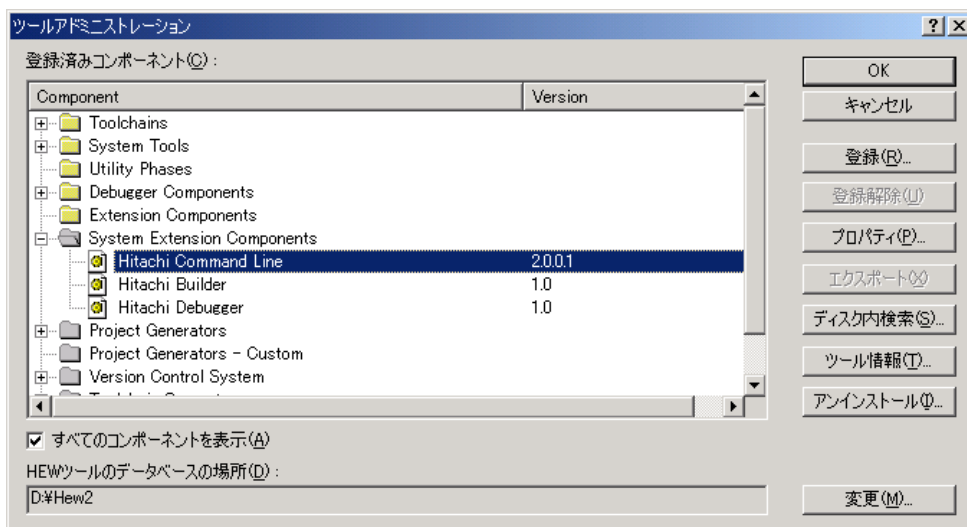


図 5.11: すべてのツールの表示

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様に動作します。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]メニューをクリックしてください。
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. HEW2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。

5.8 カスタムプロジェクトタイプ

HEW の [プロジェクト->テンプレートの作成...] 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。ユーザは、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 HEW 環境で使用するには、[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ユーザのマシン上で実行すれば、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。

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。

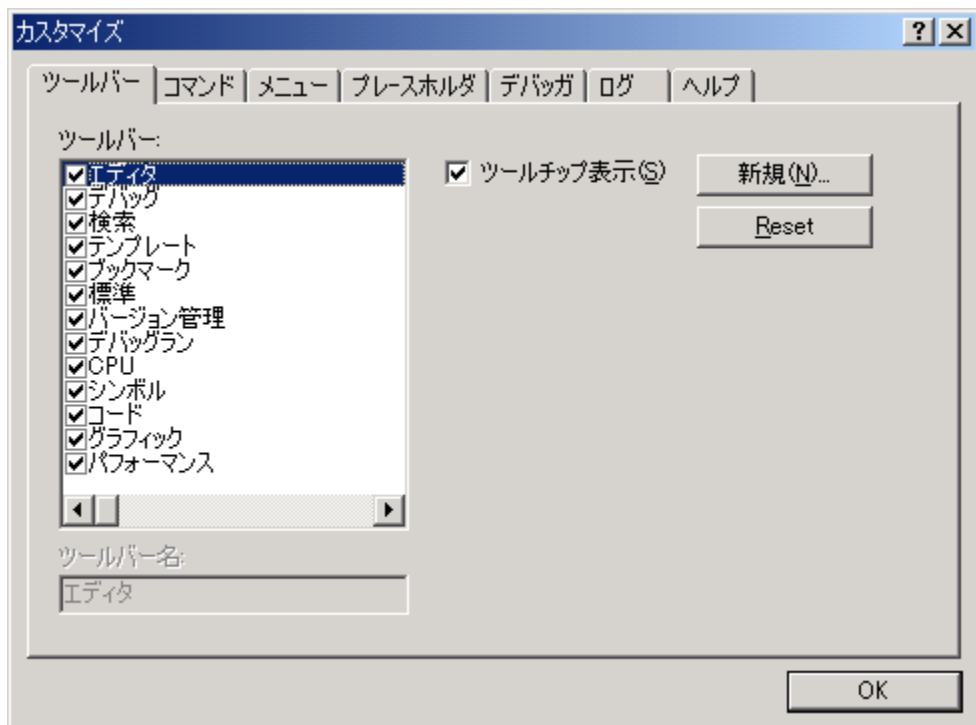


図 6.1: カスタマイズ ダイアログボックス ツールバー タブ

⇒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. [Categories]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。[Buttons]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。

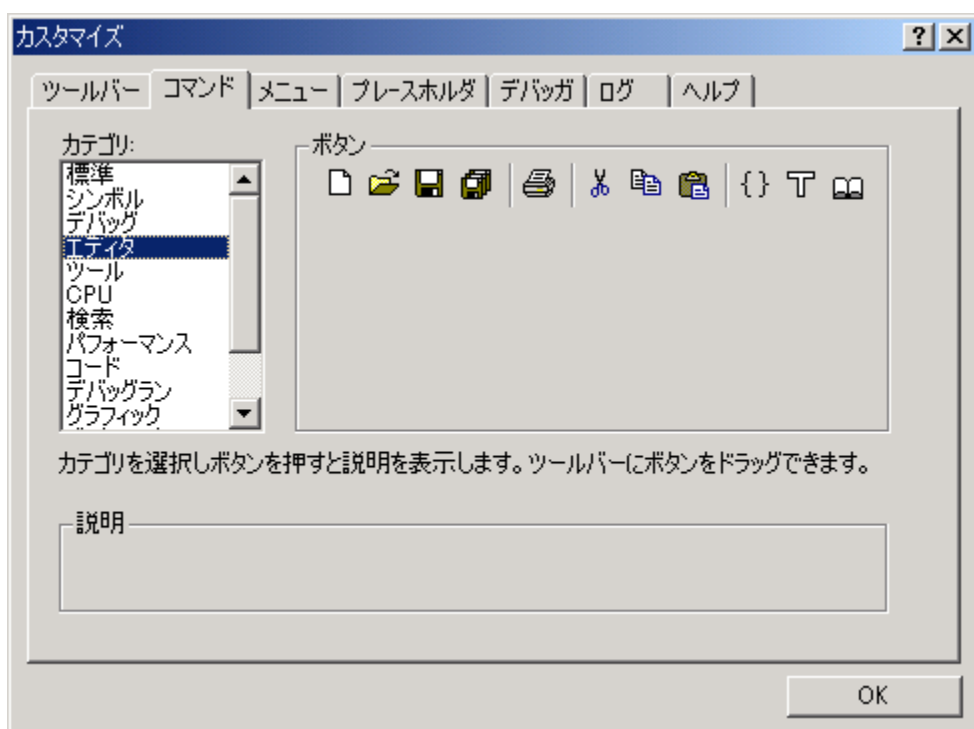


図 6.3: カスタマイズ ダイアログボックス コマンド タブ

☛ 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[Buttons]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べと図6.1の[Reset] ボタンが[Delete] ボタンに変化します。[Delete] 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 [Reset] 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。

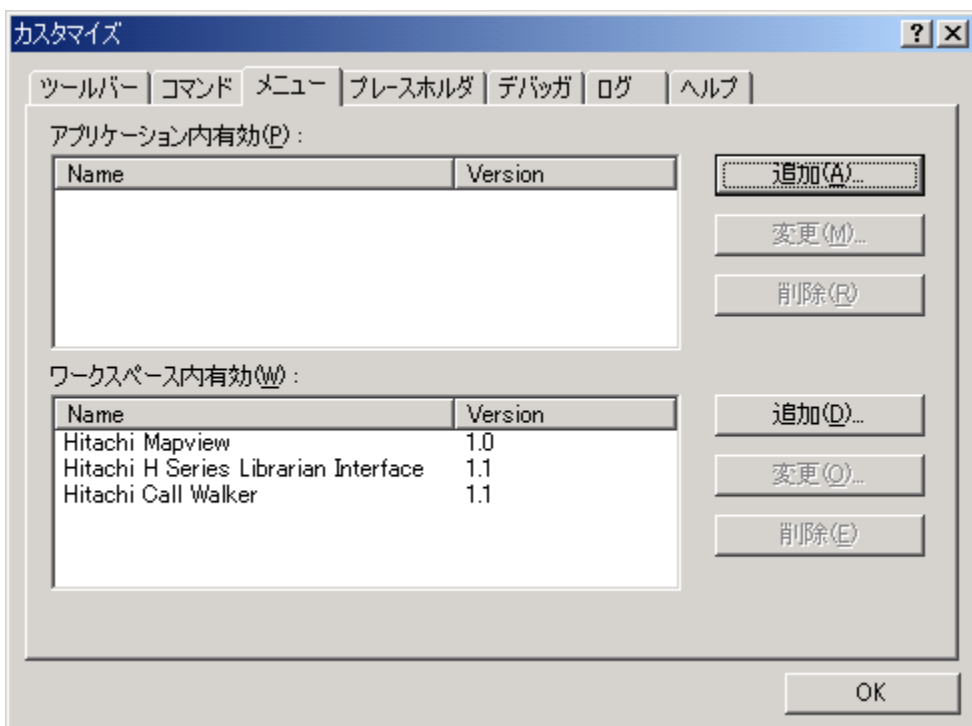


図 6.4: カスタマイズ ダイアログボックス メニュー タブ

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。また、新しいツールを追加するには、以下の手順に従ってください。
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。

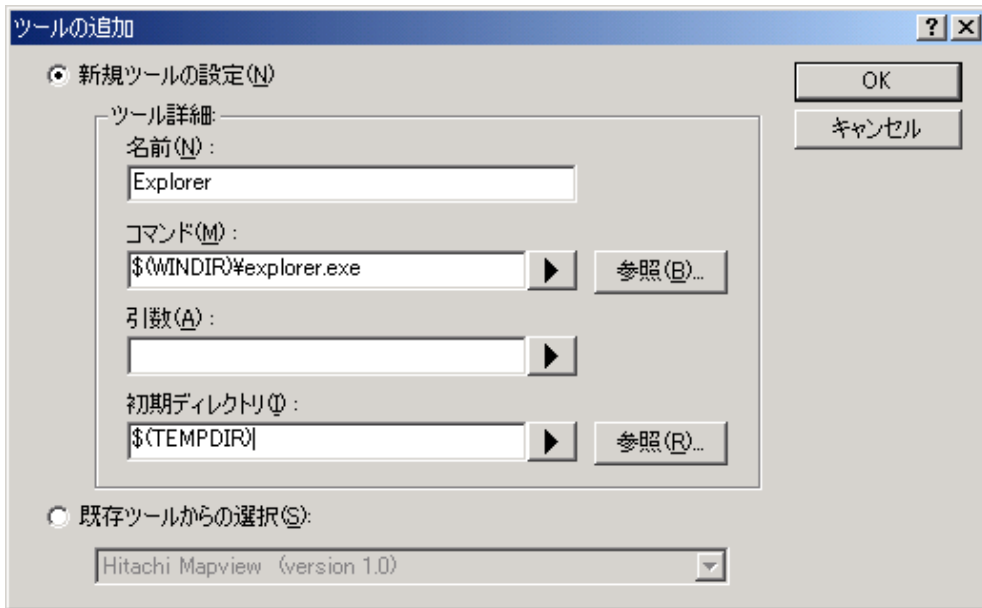


図 6.5: ツールの追加 ダイアログボックス

デフォルトでは、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。

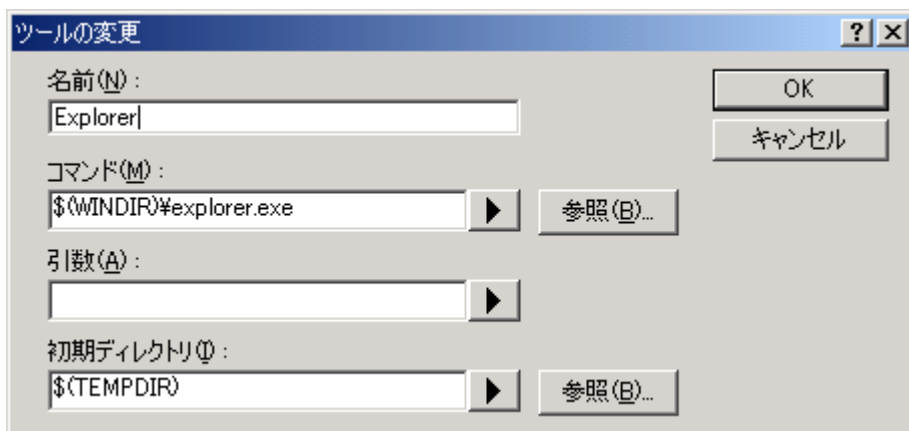


図 6.6: ツールの変更 ダイアログボックス

☛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。

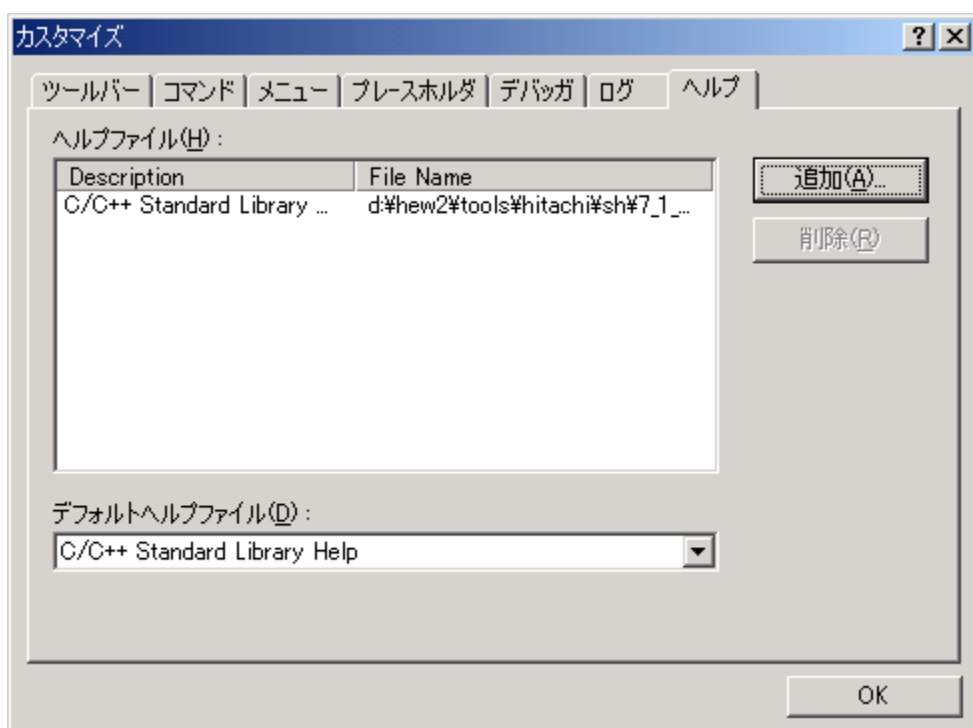


図 6.7: カスタマイズ ダイアログボックス ヘルプ タブ

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。

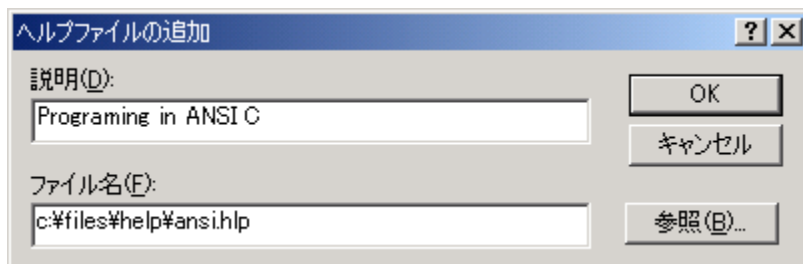


図 6.8: ヘルプファイルの追加 ダイアログボックス

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。

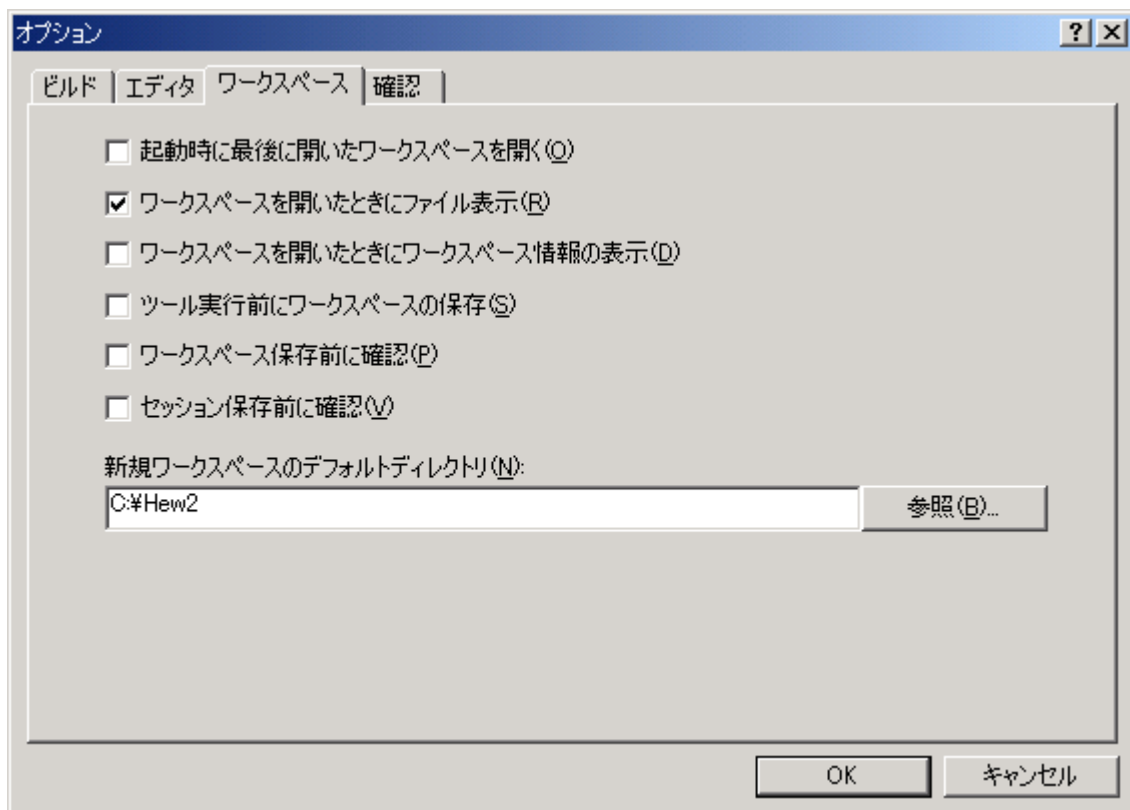


図 6.9: オプション ダイアログボックス ワークスペース タブ

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、それを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. [Workspace] 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。

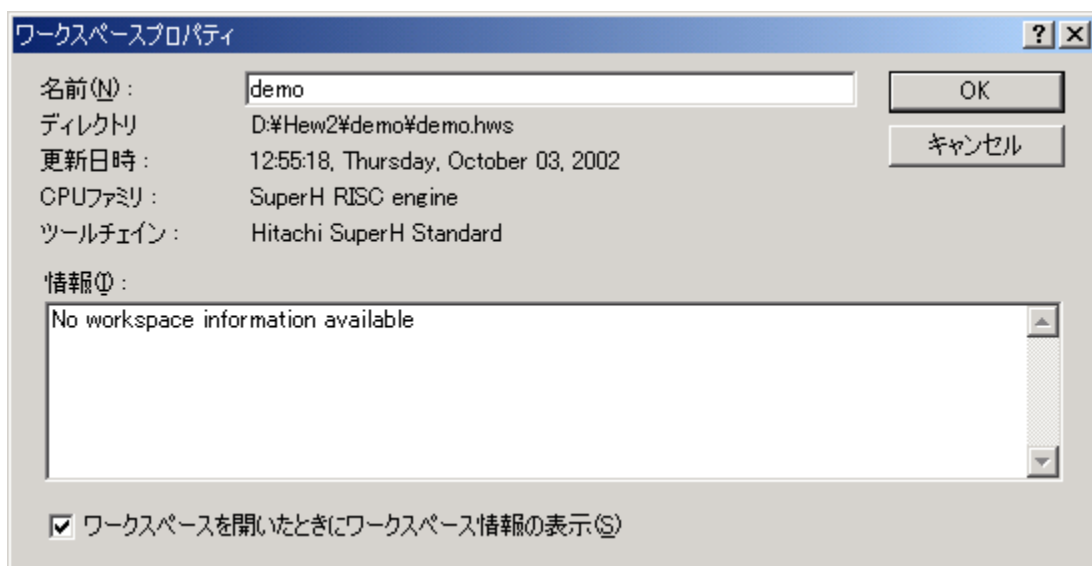


図 6.10: ワークスペースプロパティダイアログボックス

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 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、ルートディレクトリが入力されています。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。

6.4.7 セッション保存前に確認エディットボックス

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。

6.5 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6.11) を選んでください。

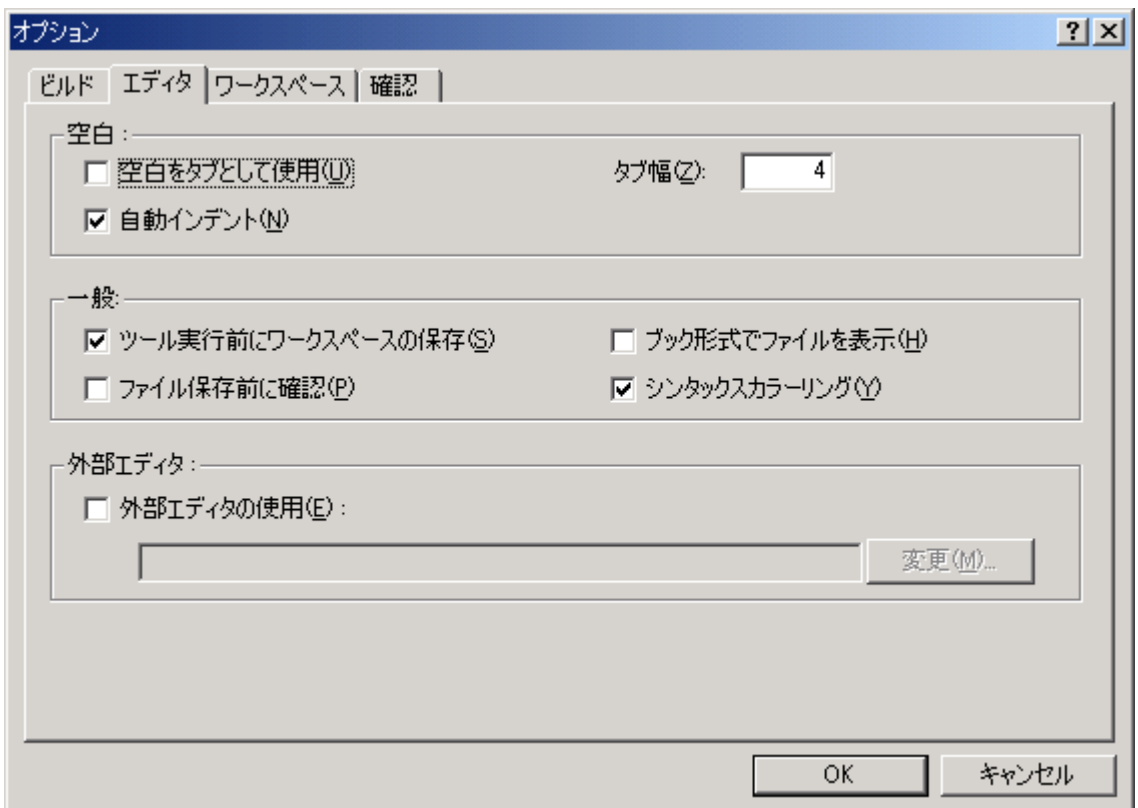


図 6.11: オプション ダイアログボックス エディタ タブ

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6.12)。



図 6.12: 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（引数をのぞいて）を入力してください。
4. [ファイルを開くときの引数]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。

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 6.11）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 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。もし他のデバッガを使いたい場合は、[ツール]メニューに加えなければいけません。

[カスタマイズ] ダイアログボックス (図 6.13) の [デバッガ] タブに HDI に関連する情報を設定します。一部のターゲットが現在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。

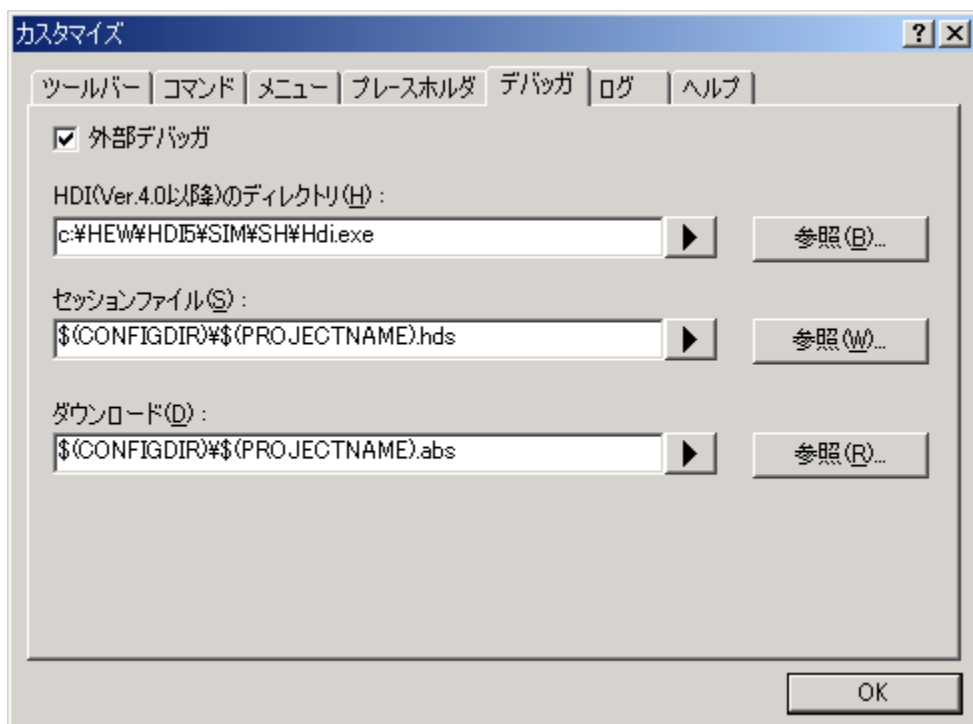


図 6.13: カスタマイズ ダイアログボックス デバッガ タブ

外部デバッガで使う場合は、[外部デバッガ]をチェックし、以下の項目を設定します。指定する項目は3つあります。1つはHDIの実行ファイルの場所です。HDIはバージョン4.0以降のものを指定してください。そうでない場合、動作を保証できません。2つめはセッションファイルの情報です。HDIに切り替わったとき、どのセッションをロードするかをHDIに伝えます。3つめはダウンロードモジュールの場所です。これは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が自動的にHDIに切り替わるために必要です。

指定したセッションファイルでHDI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、ダウンロードモジュールが更新された場合、HEWからHDIに戻り、すぐデバッグできるようになります。HDI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に戻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。

6.8 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。

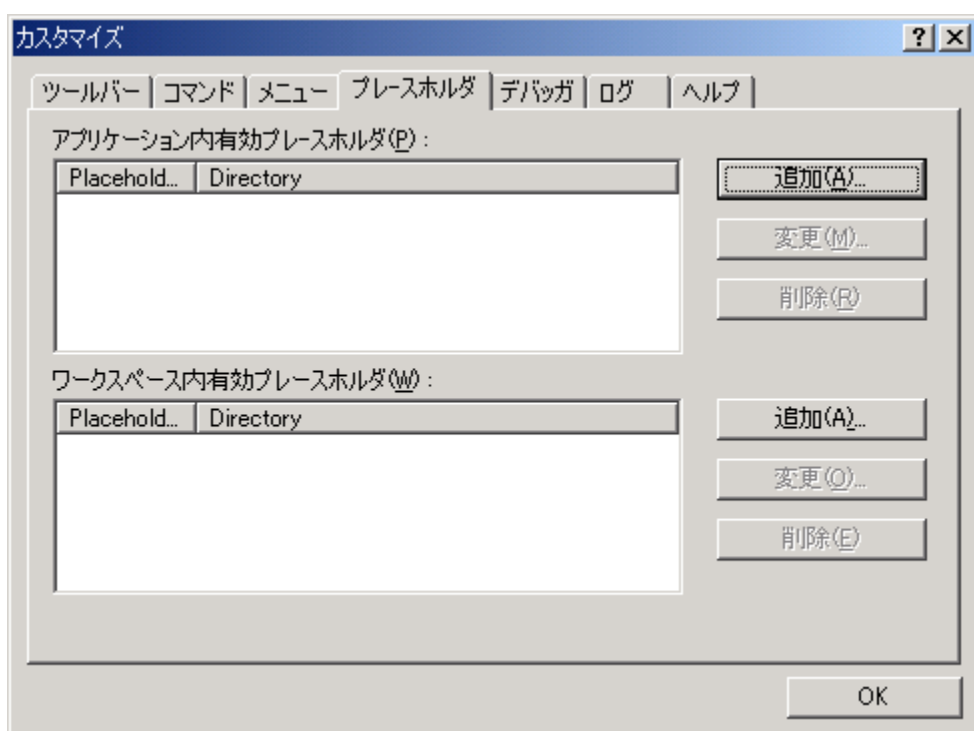


図 6.14: カスタマイズ ダイアログボックス プレースホルダ タブ

➡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プレースホルダ]、あるいは[ワークスペース内有効ホルダ]のどちらを使いたい、を選んでください。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。

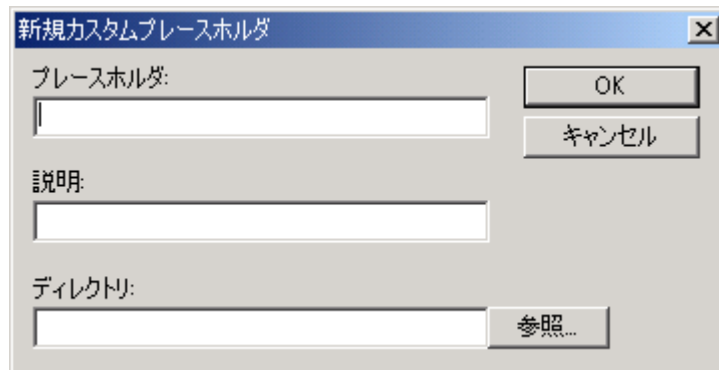


図 6.15: 新規カスタムプレースホルダダイアログボックス

6.9 確認ダイアログボックスを使う

HEW では、ある操作を行うときに、数多くの確認内容を設定することができます。通常、これらはデフォルトで設定されていますが、何度も操作を行う場合は、このダイアログボックスを使うのに疲れるかもしれません。[オプション]ダイアログボックスの[確認]ページは、確認用のダイアログボックスを管理するために使用します。

➡[確認]ダイアログボックスを見るには

1. [ツール->オプション...]から[確認]タブ（図6.16）を選んでください。
2. 確認内容がこのダイアログボックスに一覧表示されます。
3. 確認内容のチェックを切り替えるには、名前の左側にあるチェックボックスを使用してください。

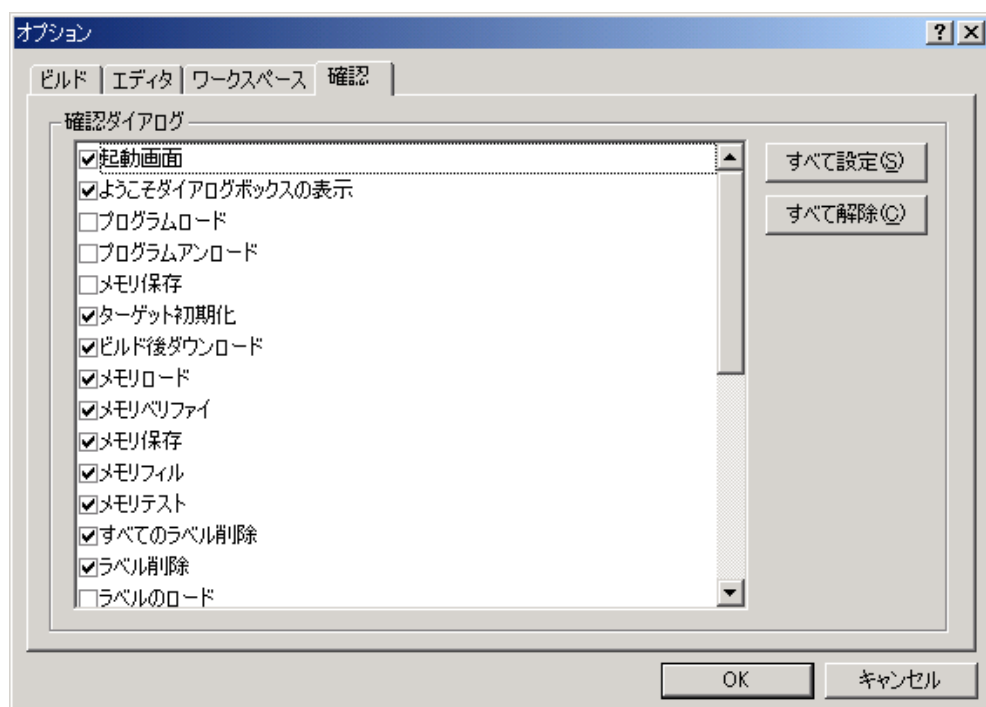


図 6.16: オプションダイアログボックス 確認ページ

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。

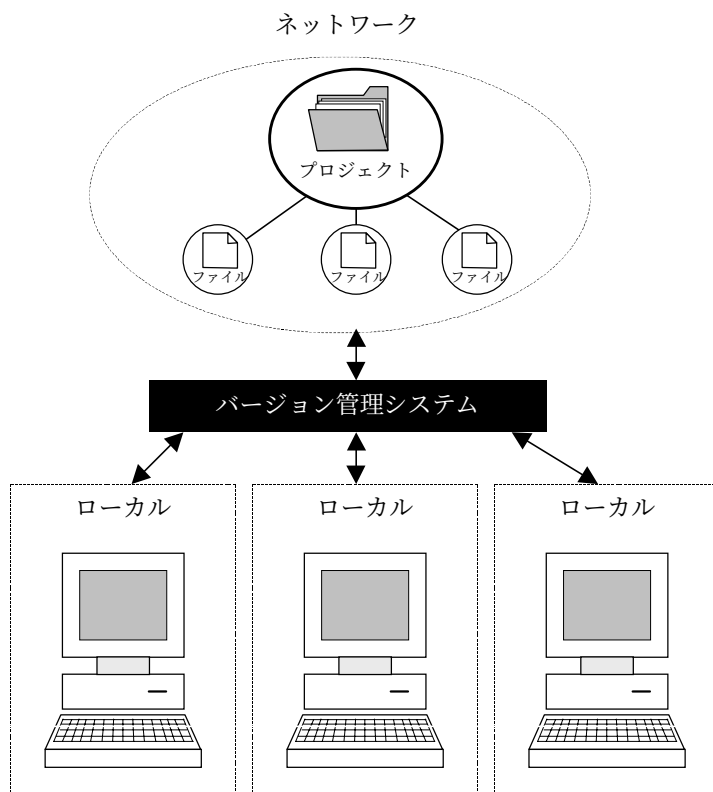


図 7.1: バージョン管理

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。

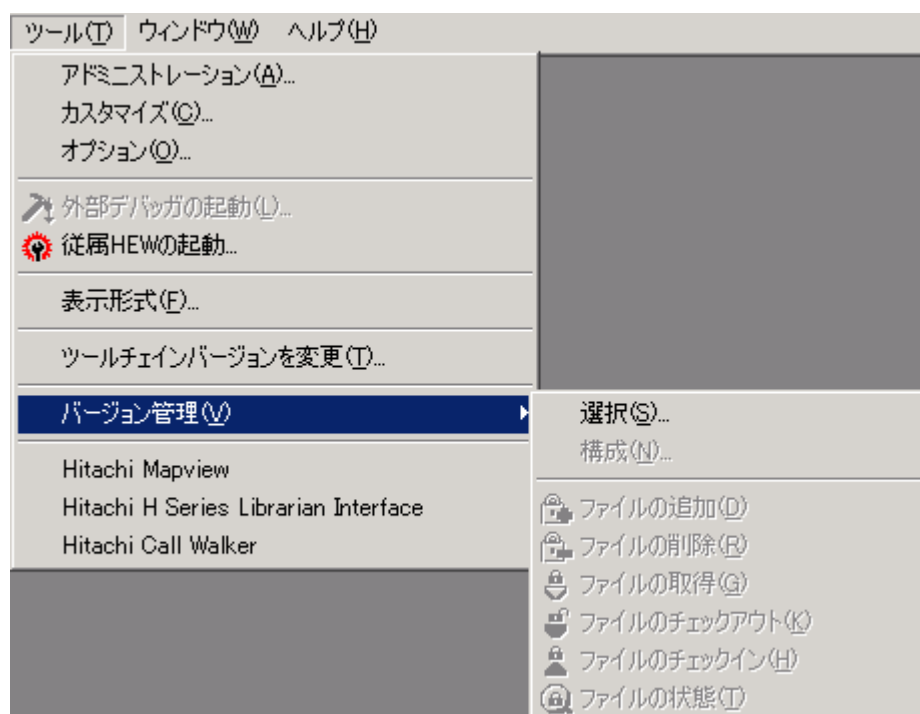


図 7.2: バージョン管理サブメニュー

☛ 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に加えて、カスタムのバージョン管理システムを構築できます。HEWにより、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します。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、関連するコマンドが実行され、出力が[Output]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 [Workspace]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されます。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されます。これには、どのような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フレキシブルに構築できます。[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）。

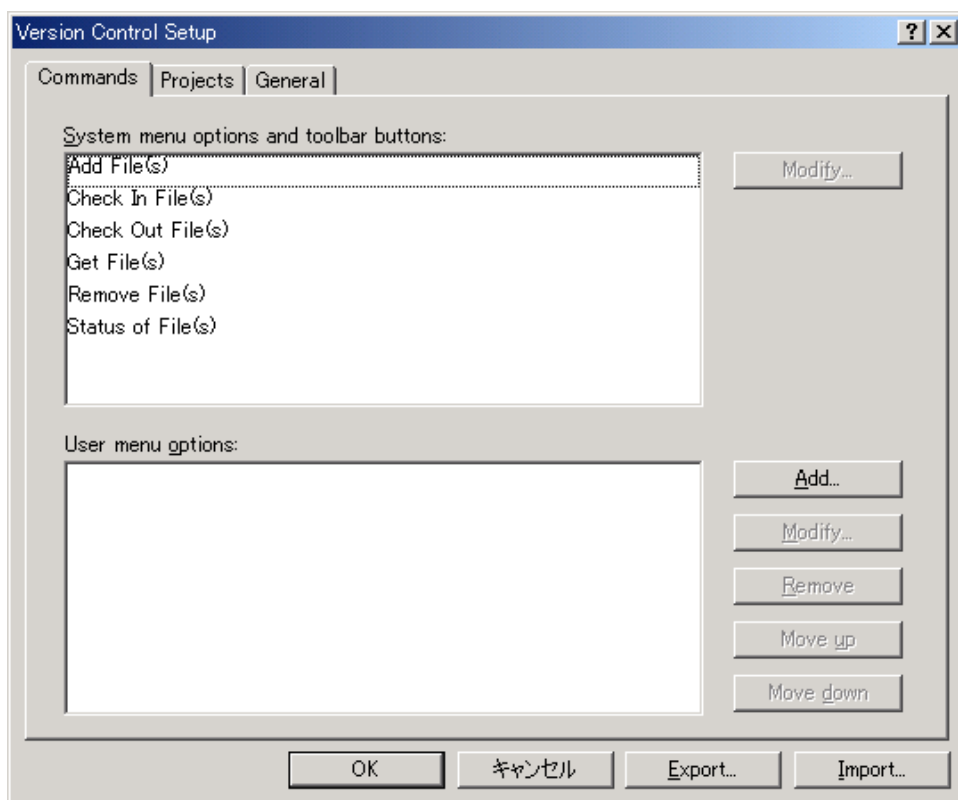


図 8.1: Version Control Setup ダイアログボックス Commands タブ

[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。

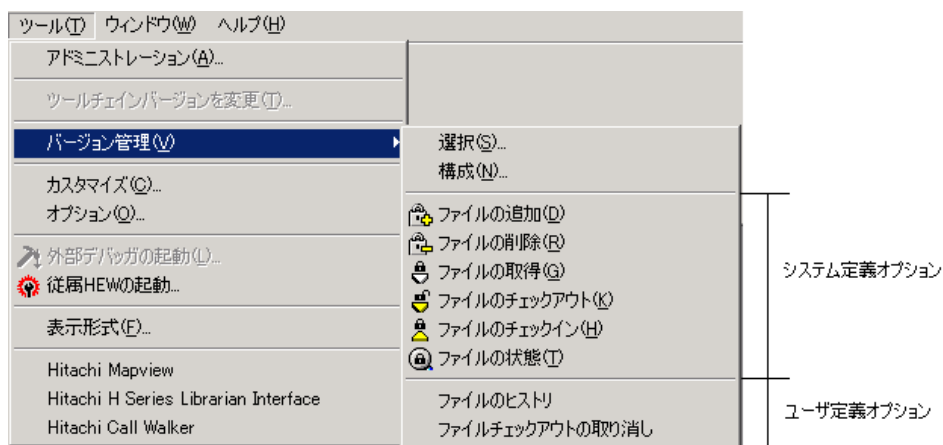


図 8.2: バージョン管理 サブメニュー

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んだファイルを追加する
Remove File(s)	バージョン管理から選んだファイルを削除する
Get File(s)	バージョン管理から選んだ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んだ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んだ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)	選んだ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。

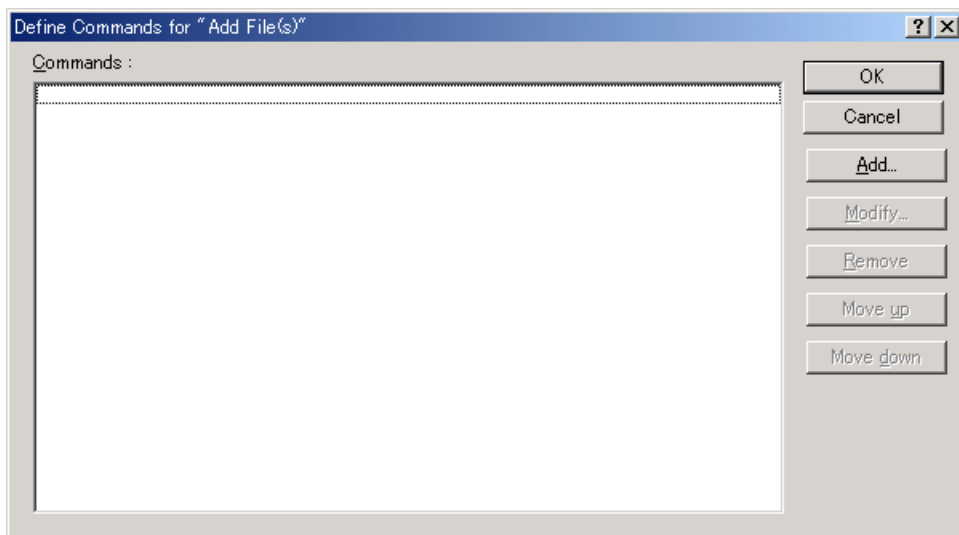


図 8.3: システムメニューオプションの変更 (例)

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、メニューの順序も指定できます。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。

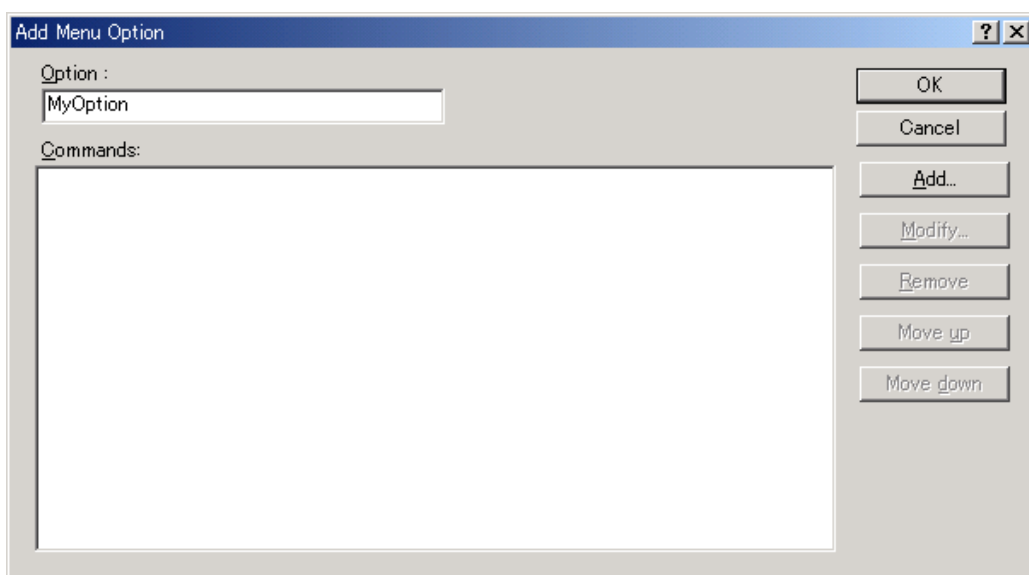


図 8.4: Add Menu Option ダイアログボックス

➡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。

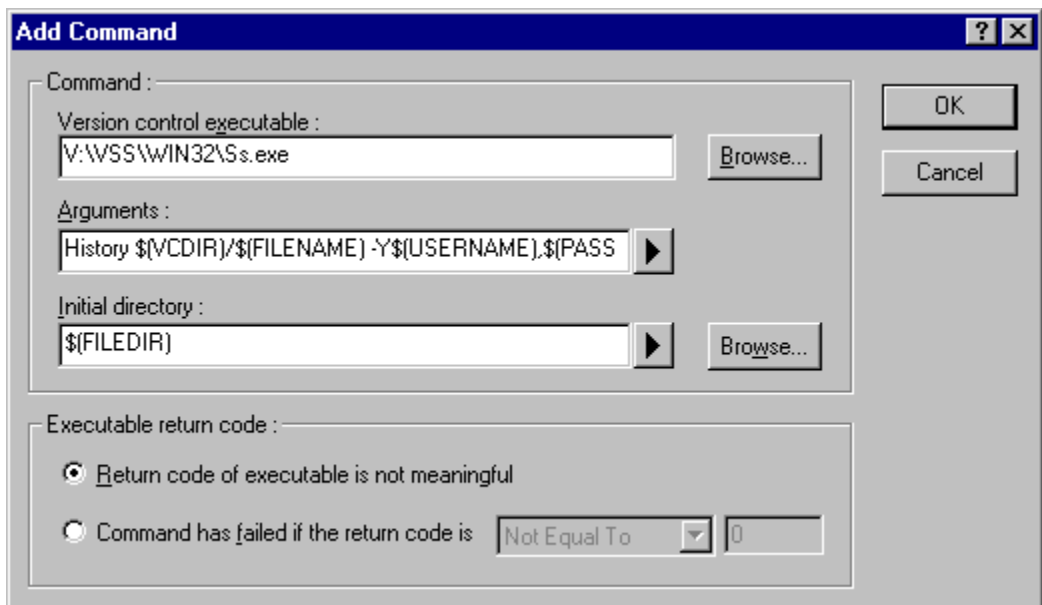


図 8.5: Add/Modify Command ダイアログボックス

☛ 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ことができるならば、[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。

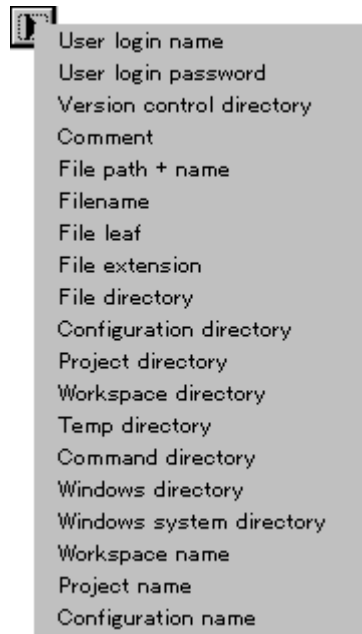


図 8.6: 変数フィールドプレースホルダのポップアップメニュー

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（つまり、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（つまり、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。

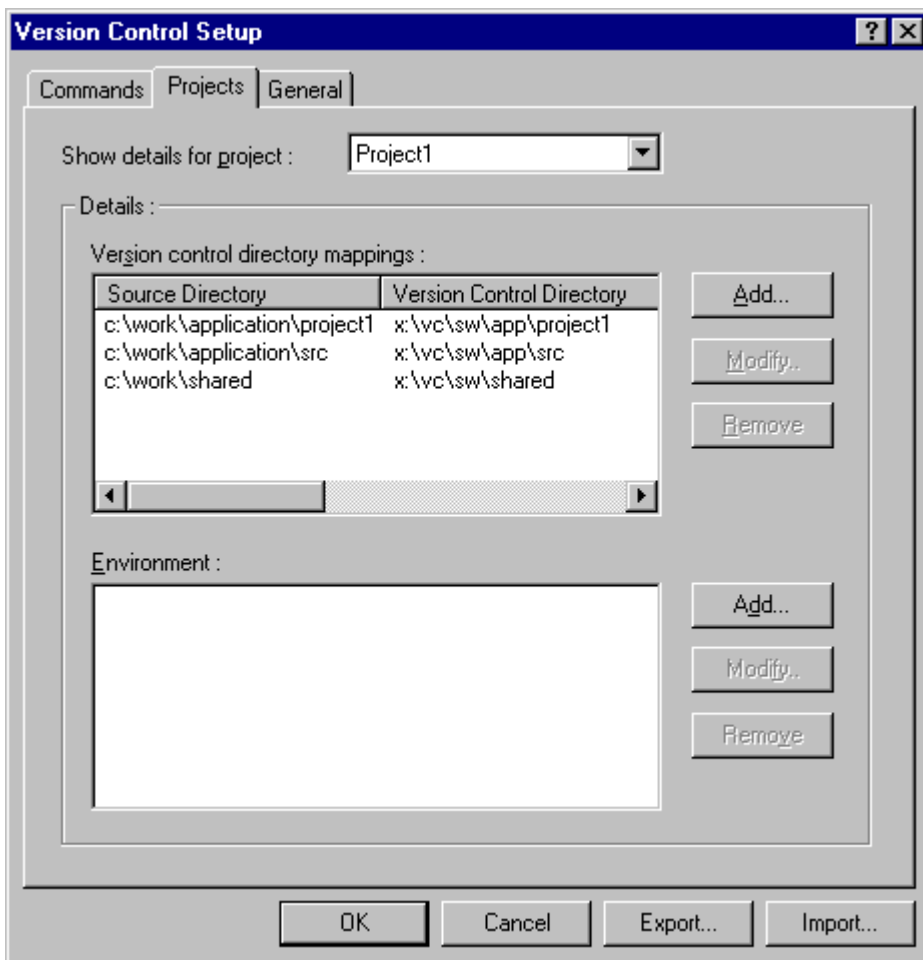


図 8.7: Version Control Setup ダイアログボックス Projects タブ

➡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。

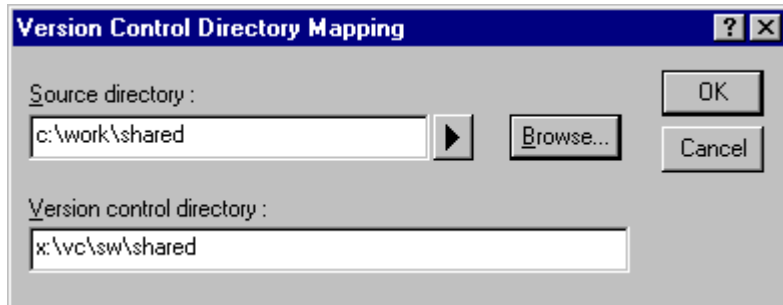


図 8.8: Version Control Directory Mapping ダイアログボックス

➡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。

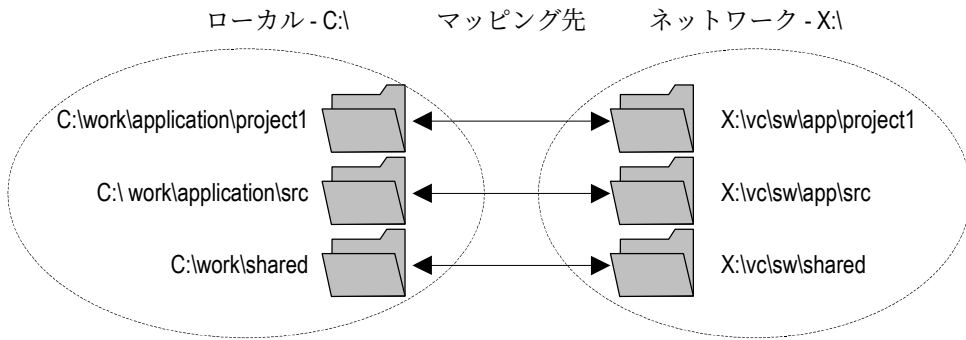


図 8.9: マッピングの例

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

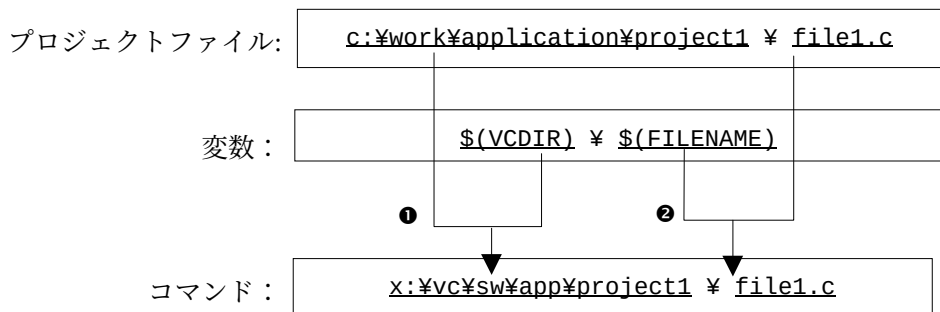


図 8.10: 置き換えの例

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。

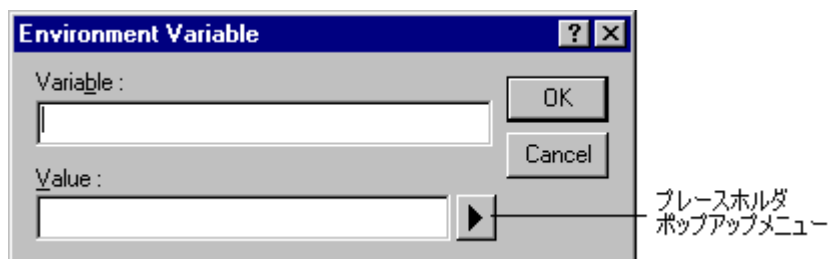


図 8.11: Environment Variable ダイアログボックス

➡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。

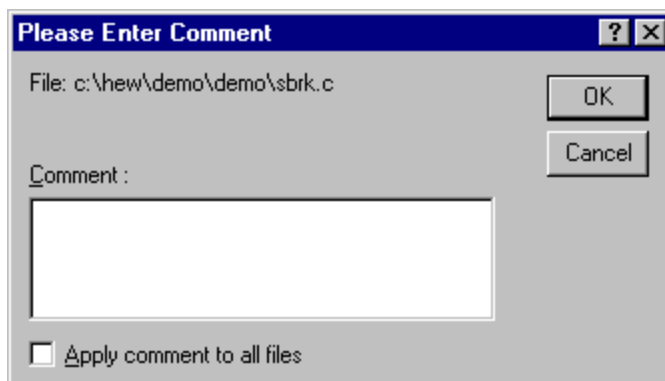


図 8.12: Please Enter Comment ダイアログボックス

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。

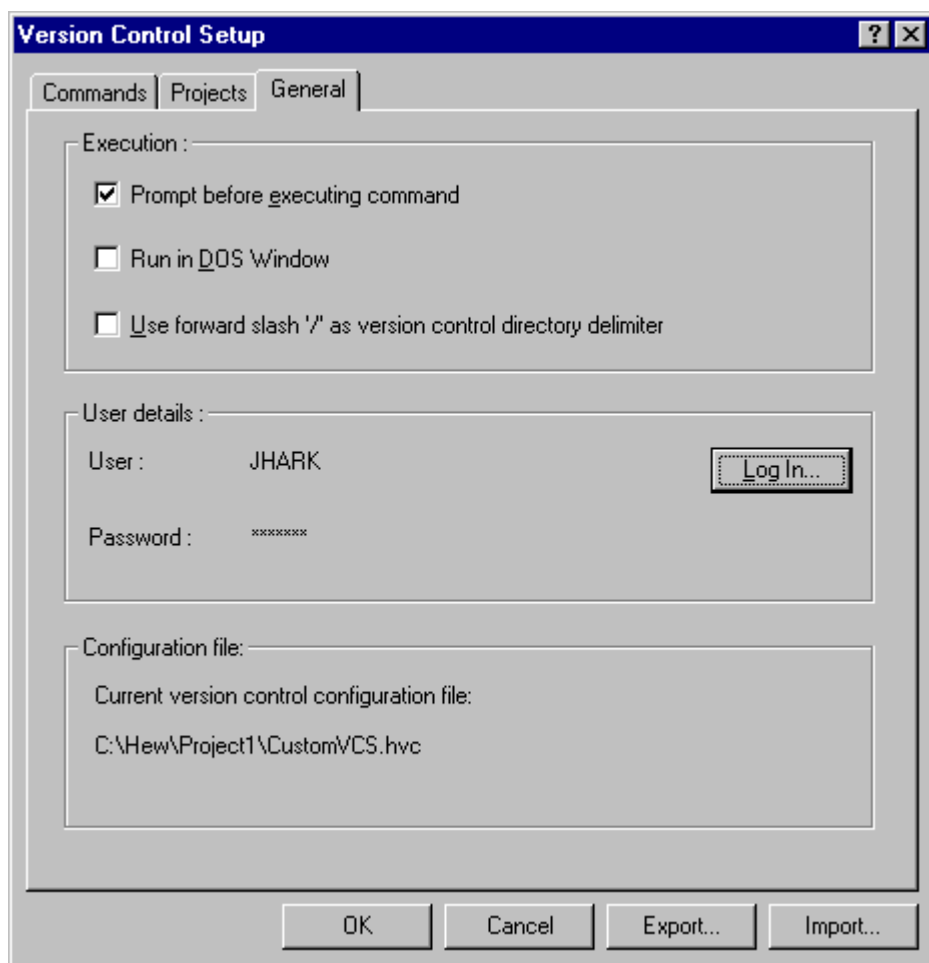


図 8.13: Version Control Setup ダイアログボックス General タブ

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。

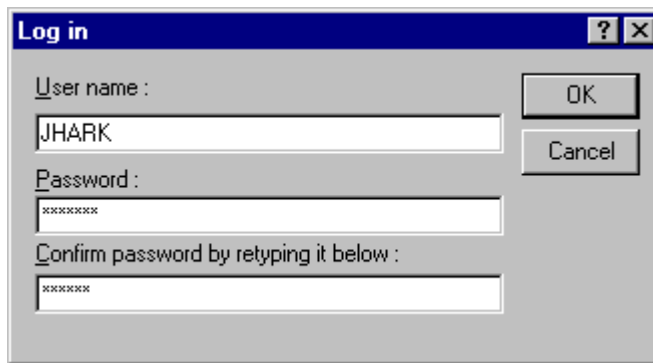


図 8.14: Log in ダイアログボックス

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。[キャンセル] をクリックするとコマンドは実行しません。

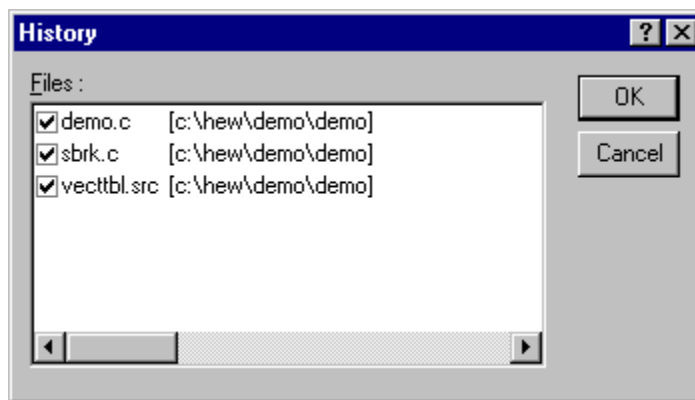


図 8.15: コマンドプロンプト ダイアログボックス (例)

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[Output]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ます。現在、HEW は Visual SourceSafe のバージョン 5 と 6 だけ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（つまり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（つまりフォルダ）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。

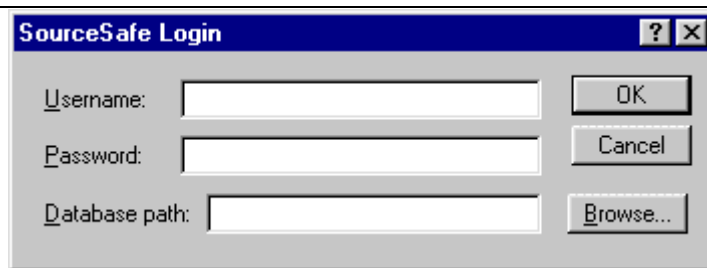


図 9.1: SourceSafe Login ダイアログボックス



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した。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されていません。

9.1.2 Visual SourceSafe にファイルを追加する

前節では、ハードディスクのプロジェクトディレクトリ（つまり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（つまり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最初何もありません。

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、ワーキング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を確認したり、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 [Output]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 ([Workspace] 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。

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 ([Workspace] 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。

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“Output” 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 [Output] 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。

デバッグ編

1. はじめに

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。

本システムは、H8Sシリーズのマイクロコンピュータを使用したシステムの開発をソフトウェア、ハードウェアの両面からサポートする支援装置です。

E10A エミュレータの本体である PCMCIA カードエミュレータ、または PCI カードエミュレータ(以降、カードエミュレータと略す)は、専用デバッグインタフェースを経由して、ユーザシステムに接続します。このため完成した製品に近い形態でデバッグを行うことができます。また、PCMCIA TYPE II スロット、または PCI スロットを搭載しているパーソナルコンピュータ (IBM PC 互換機) をホストコンピュータにして実験室、フィールドと場所を選ばずデバッグを行うことができます。

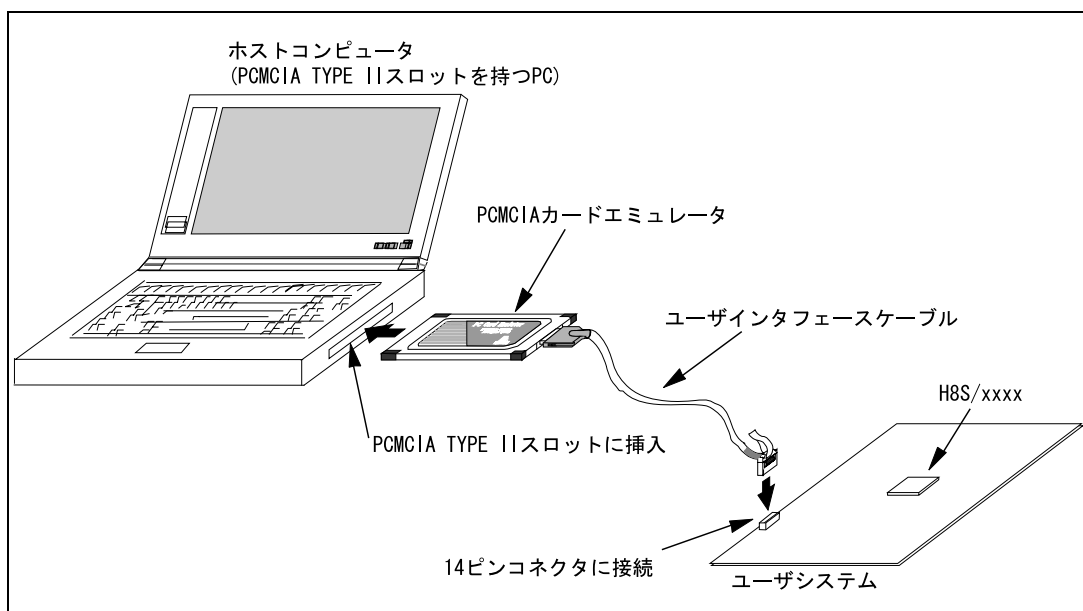


図 1.1 E10A エミュレータを使用したシステム構成外観 (PCMCIA カードエミュレータ使用時)

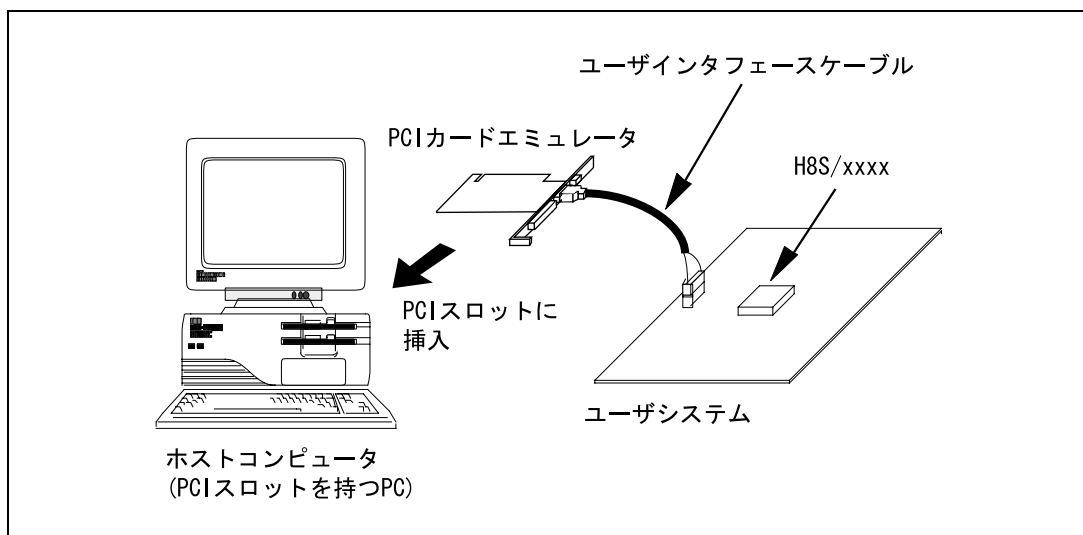


図 1.2 E10A エミュレータを使用したシステム構成外観 (PCI カードエミュレータ使用時)

E10Aエミュレータには2種類の起動モードがあります。

- (1) E10Aエミュレータ用プログラムをターゲットデバイスにダウンロード後、接続するモード
本モードは、ターゲットデバイスのフラッシュメモリ内にE10Aエミュレータ用プログラムが存在しない場合に使用します。
- (2) フラッシュメモリのライターとしてE10Aエミュレータを使用するモード
本モードは、フラッシュメモリのライターとしてE10Aエミュレータを使用する場合に使用します。

E10Aエミュレータの特長は、以下のとおりです。

- (1) コストパフォーマンスに優れたカードエミュレータ
PCMCIAまたはPCIインタフェースにより、小型サイズ、低価格を実現しました。
- (2) リアルタイムエミュレーション
MCUの最高動作周波数でリアルタイムエミュレーションが行えます。
- (3) 優れた操作性を実現
Microsoft® Windows® 98、Microsoft® Windows® Me、Microsoft® Windows NT®、Microsoft® Windows® 2000および、Microsoft® Windows® XP環境下で動作するHEW (High-performance Embedded Workshop) 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。
- (4) 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。
- (5) 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。
- (6) 製品形態でのユーザシステムのデバッグ
ユーザシステム完成時の製品形態に近い状態でユーザシステムのデバッグを行うことができます。
- (7) コンパクトなデバッグ環境
PCMCIAカードエミュレータを使用すると、ノート型パソコンをホストコンピュータとして使用でき、場所を選ばずデバッグ環境を作成することができます。
- (8) マイコン内蔵のフラッシュメモリのライターとしても使用できます。

1.1 使用環境条件

注意

E10A エミュレータを使用する場合、表 1.1、および表 1.2 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 E10A エミュレータを使用した場合、E10A 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。

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

表 1.2 動作環境

項番	項目	動作環境
1	ホストコンピュータ	PentiumIII 以上（推奨 600MHz 以上）を搭載し、PCMCIA TYPE II スロット、または PCI スロットを備えた IBM PC およびその互換機
2	OS	Windows® 98、Windows® Me、Windows NT®, Microsoft® Windows® 2000 および、Microsoft® Windows® XP
3	最小稼動メモリ容量	128MB 以上（推奨ロードモジュールサイズの 2 倍以上）
4	ハードディスク容量	インストールディスク容量 100MB 以上 （スワップ領域を考慮してメモリ容量の 2 倍以上（推奨 4 倍以上）の空き容量をご用意ください。）
5	マウスなどのポインティングデバイス	ホストコンピュータ本体に接続可能で Windows® 98、Windows® Me、Windows NT®, Microsoft® Windows® 2000 および、Microsoft® Windows® XP に対応しているマウスなどのポインティングデバイス
6	電源電圧	5.0±0.25V（エミュレータは PC から供給されます）
7	消費電流	HS0005KCM05H : 55mA (max) HS0005KCI05H : 55mA (max)
8	CD-ROM ドライブ	HEW2 および E10A エミュレータをインストールするため、または E10A エミュレータユーザズマニュアルを参照するために必要

1.2 梱包品の確認

梱包を解いた後、梱包品がそろっているか確認してください。E10A エミュレータの梱包品は、別冊「H8S/xxxx E10A エミュレータ製品固有ガイド」の 1.1 章を参照してください。確認した結果、梱包品に不足がありましたら、ユーザ登録先の弊社メールアドレスまでご連絡くださるか、WEB サイトをご参照ください。

2. E10A エミュレータ機能

本章では、E10A エミュレータの機能を紹介します。

E10A エミュレータがサポートするデバイスにより、多少機能が異なります。
各機能の使用方法は、「6 チュートリアル」をご参照ください。

2.1 機能概要

E10A エミュレータの機能概要を表 2.1 に示します。

製品ごとの機能については、オンラインヘルプを参照してください。

表 2.1 E10A エミュレータの機能

項番	項目	機能
1	ユーザプログラム 実行系機能	・デバイスが保証する範囲の動作周波数による、プログラム実行 ・リセットエミュレーション ・Step 機能 - シングル Step 機能 (1Step: 1 命令) - ソースレベル Step 機能 (1Step: ソース 1 行) - Step Over 機能 (サブルーチン内はブレークしない) - Step Out 機能 (PC 実行中のサブルーチンの呼出し元関数に戻るまで実行)
2	リセット 機能	・ブレーク中、HEW からデバイスへパワーオンリセット発行
3	トレース 取得内容	・デバイス内蔵の分岐トレース機能 (4 分岐: min)
4	ブレーク 条件	・ハードウェアブレーク条件 (2 本: min) ・PC ブレーク条件 (255 箇所)
5	メモリアク セス機能	・RAM へのダウンロード ・フラッシュメモリへのダウンロード ・1 行アセンブル ・逆アセンブル ・メモリリード ・メモリライト ・ユーザプログラム実行中の変数内容の表示自動更新 ・フィル ・サーチ ・ムーブ ・コピー
6	汎用/制御レ ジスタアク セス機能	汎用/制御レジスタのリード/ライト
7	内蔵 I/O レジ スタアクセ ス機能	内蔵 I/O レジスタのリード/ライト
8	ソースレベ ルデバッグ 機能	豊富なソースレベルデバッグ機能。
9	コマンドラ イン機能	コマンド入力をサポートしています。 各コマンドを入力順に羅列したファイルを作成すれば、バッチ処理を行うこともできます。
10	ヘルプ機能	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ます。

次の章から、E10A エミュレータの特徴的な機能についてご説明します。

【留意事項】

ハードウェアブレーク条件の本数は、製品ごとに異なりますのでご了承ください。

2.2 **トレース機能**

分岐元アドレスと、ニーモニック、オペランド、ソース行を表示します。

デバイスに内蔵されているトレースバッファを使用して実現しますので、リアルタイムにトレースを取得することができます。

2.3 ブレーク機能

E10A エミュレータでは、以下の 2 種類のブレーク機能があります。

(1) ハードウェアブレーク機能

デバイス内蔵のブレークコントローラを使用した機能です。

アクセスアドレス条件、命令フェッチアドレス条件、データ条件、バスサイクル条件などが設定できます。

また、ソースウインドウの”EVENT カラム”からも設定できます。設定方法は「5.2 プログラムを表示する」を参照してください。

(2) PCブレーク機能 (BREAKPOINT)

指定アドレスの命令を専用命令に置きかえることでブレークする機能です。

本機能は、[Event]ダイアログボックスの[Breakpoint]ページで設定できます。

また、[Source]ウィンドウや[Disassembly]ウィンドウ上で、設定したい行の[Editor]カラムをダブルクリックすることによっても設定できます。

2.4 メモリアクセス機能

E10A エミュレータには以下のメモリアクセス機能があります。

(1) メモリリード/ライト機能

[Memory]ウィンドウ：メモリ内容をウィンドウ表示します。

[Memory]ウィンドウOPEN時に指定したサイズのみリードします。

また、[Memory]ウィンドウからライトした場合は、ウィンドウの更新のために、[Memory]ウィンドウで表示されている範囲のリードが発生します。

[Memory]ウィンドウを更新したくない場合、ポップアップの[表示固定]メニューで更新しない設定にすることができます。

me コマンド：コマンドライン機能です。

指定アドレスを指定サイズでリード、ライトする機能です。

(2) ユーザプログラムのダウンロード機能

ワークスペース内に登録されたロードモジュールをダウンロードできます。

[デバッグ]メニューの[ダウンロード]で、ダウンロードするモジュールを選択できます。

また、ワークスペース内のロードモジュールを右クリックすることによってポップアップメニューが開きますが、このポップアップメニューからもダウンロードを行うことができます。

ダウンロード先は、RAM またはフラッシュメモリです。

本機能では、デバッグ情報などソースレベルデバッグに必要な情報もダウンロードします。

(3) メモリデータのアップロード機能

指定アドレスから指定サイズ分、ファイルに保存することができます。

ファイル形式はSタイプ形式となります。

(4) メモリデータのダウンロード機能

Sタイプ形式のファイルで保存されているメモリ内容をダウンロードできます。

[Memory]ウィンドウのポップアップメニューから[ロード]を選択してください。

(5) 変数内容表示

ユーザプログラムの指定した変数の内容を表示します。

変数内容表示機能の使用方法については、「5.13 変数の表示」を参照してください。

(6) その他のメモリ操作機能

その他、以下の機能があります。

- メモリフィル機能
- メモリコピー機能
- メモリセーブ機能
- メモリベリファイ機能
- メモリサーチ機能
- 内蔵 I/O 表示機能
- ラベル名、変数名とその内容を表示する機能

【留意事項】

1. ユーザプログラム実行中のメモリアクセス
ユーザプログラム実行中にメモリウィンドウ等からメモリアクセスした場合、E10A エミュレータ内部でユーザプログラムの実行を一旦停止してメモリアクセスし、その後ユーザプログラムを再実行しています。したがって、ユーザプログラムのリアルタイム性はありません。
2. ユーザプログラムブレーク中のメモリアクセス
E10A エミュレータは、フラッシュメモリ領域に対してもダウンロード、BREAKPOINT の設定をすることができます。

2.5 スタックトレース機能

E10A 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、「6.19 スタックトレース機能」を参照してください。

2.6 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。

3. 使用前の準備

E10A 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。



準備を行う前に図 3.1 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、E10A 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。

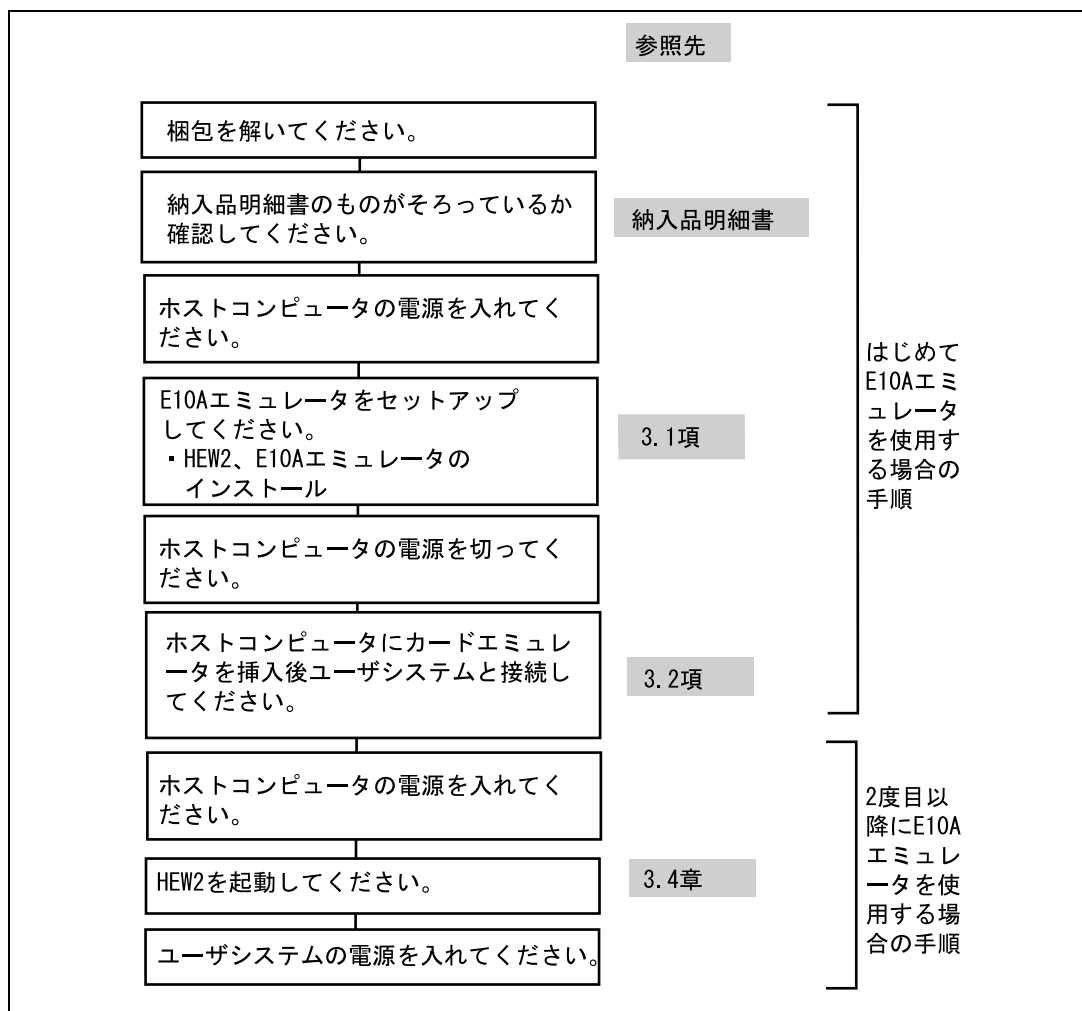


図 3.1 E10A エミュレータ使用フローチャート

3.1 H8S/xxxx E10A エミュレータのインストール

CD-R を CD-ROM ドライブに挿入してください。CD-R のルートディレクトリから Setup.exe を実行、インストールウィザードに従いインストールを行ってください。

また、インストール時にハードウェアの設定も行うため、ご使用のオペレーティングシステムおよびインタフェース（PCI、PCMCIA）によりインストール手順が異なります。ご使用の環境に合わせたインストール手順に従ってインストールを進めてください。

3.1.1 Windows® 98 operating system または Windows® Me operating system へのインストール

(1) PCI カードタイプの E10A エミュレータをご使用の場合

1. H8S/xxxx E10Aエミュレータのインストールを行ってください。
(コンポーネントの選択では[PCI Card Driver]を必ず選択してください。)
2. オペレーティングシステムを終了し、PCの電源を切ってください。
3. E10AエミュレータPCIカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
4. PCを起動してください。ハードウェアが認識され、自動的にドライバのインストールが開始されます。*

(2) PCMCIA タイプの E10A エミュレータをご使用の場合

1. H8S/xxxx E10Aエミュレータのインストールを行ってください。
(コンポーネントの選択では[PC Card Driver (PCMCIA)]を必ず選択してください。)
2. E10AエミュレータPCMCIAカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
3. ハードウェアが認識され、自動的にドライバのインストールが開始されます。*

【注】* 「新しいハードウェアの追加ウィザード」が表示された場合、[使用中のデバイスに最適なドライバを検索する(推奨)]を選択し、検索場所として[検索場所の指定]を選択してください。指定する検索場所は、それぞれ以下としてください。

• PCI カードタイプの E10A をご使用の場合：

「<ドライブ>:¥DRIVERS¥PCI¥98」を指定。

• PCMCIA カードタイプの E10A をご使用の場合：

「<ドライブ>:¥DRIVERS¥PCMCIA¥98」を指定。

(<ドライブ>は CD-ROM ドライブのドライブ名です。)

3.1.2 Windows NT[®] 4.0 operating system へのインストール

(1) PCI カードタイプの E10A エミュレータをご使用の場合

1. オペレーティングシステムを終了し、PCの電源を切ってください。
2. E10AエミュレータPCIカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
3. PCを起動し、Administrator権限でログオンしてください。
4. H8S/xxxx E10Aエミュレータのインストールを行ってください。
 (コンポーネントの選択では[PCI Card Driver]を必ず選択してください。[PCI Card Driver]コンポーネントの下に、各製品の型名を選択するチェックボックスがありますので、正しい型名を選択してください。正しい型名を選択しないと、正しいドライバがインストールされないため、E10Aエミュレータが動作しません。)
5. PCを再起動してください。

(2) PCMCIA タイプの E10A エミュレータをご使用の場合

1. オペレーティングシステムを終了し、PCの電源を切ってください。
2. E10AエミュレータPCMCIAカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
3. PCを起動し、Administrator権限でログオンしてください。
4. H8S/xxxx E10Aエミュレータのインストール中にE10AエミュレータPCMCIAカードが使用するリソースについての問い合わせがあるため、設定値を事前に調べる必要があります。
 [スタート]メニュー → [プログラム] → [管理ツール (共通)] → [Windows NT診断プログラム]を起動し、リソースパネルよりIRQ、I/Oポート、およびメモリの使用状況を確認し、他のデバイスと競合しない設定値を決定してください。(IRQ: 1チャンネル、I/Oポート: H'Fバイト、メモリ: H'4000バイトのリソースを使用します。)
5. H8S/xxxx E10Aエミュレータのインストールを行ってください。
 (コンポーネントの選択では[PC Card Driver (PCMCIA)]を必ず選択してください。[PC Card Driver (PCMCIA)]コンポーネントの下に、各製品の型名を選択するチェックボックスがありますので、正しい型名を選択してください。正しい型名を選択しないと、正しいドライバがインストールされないため、E10Aエミュレータが動作しません。)
6. PCを再起動してください。

【留意事項】

[Drivers]コンポーネントで選択されたドライバは、PC 起動後にすべて開始される設定になっています。したがって、各カードを抜いた状態でPC を起動した場合や、正しいドライバ以外をインストールした場合、ドライバが開始できないため、サービスコントロールマネージャがエラーを通知しますが、問題はありません。

3.1.3 Windows® 2000 operating system または Windows® XP operating system へのインストール

(1) PCI カードタイプの E10A エミュレータをご使用の場合

1. Administrator権限でログオンしてください。
2. H8S/xxxx E10Aエミュレータのインストールを行ってください。
(コンポーネントの選択では[PCI Card Driver]を必ず選択してください。)
3. オペレーティングシステムを終了し、PCの電源を切ってください。
4. E10AエミュレータPCIカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
5. PCを起動し、Administrator権限でログオンしてください。ハードウェアが認識され、自動的にドライバのインストールが開始されます。*

(2) PCMCIA タイプの E10A エミュレータをご使用の場合

1. Administrator権限でログオンしてください。
2. H8S/xxxx E10Aエミュレータのインストールを行ってください。
(コンポーネントの選択では[PC Card Driver (PCMCIA)]を必ず選択してください。)
3. E10AエミュレータPCMCIAカードをPCに装着してください。
「3.2 ホストコンピュータとカードエミュレータとの接続」をご参照ください。
4. ハードウェアが認識され、自動的にドライバのインストールが開始されます。*

【注】 * 「新しいハードウェアの検出ウィザード」が表示された場合、[デバイスに最適なドライバを検索する (推奨)]を選択し、検索場所として[検索場所の指定]を選択してください。指定する検索場所は、それぞれ以下としてください。

- PCI カードタイプの E10A をご使用の場合：
「<ドライブ>:¥DRIVERS¥PCI¥2000」を指定。
- PCMCIA カードタイプの E10A をご使用の場合：
「<ドライブ>:¥DRIVERS¥PCMCIA¥2000」を指定。
(<ドライブ>は CD-ROM ドライブのドライブ名です。)

3.2 ホストコンピュータとカードエミュレータとの接続

カードエミュレータをホストコンピュータの PCMCIA TYPE II スロット、または PCI スロットに挿入してください。

[留意事項] カードエミュレータ装着前に、必ず H8S/xxxx E10A エミュレータのインストールを行ってください。

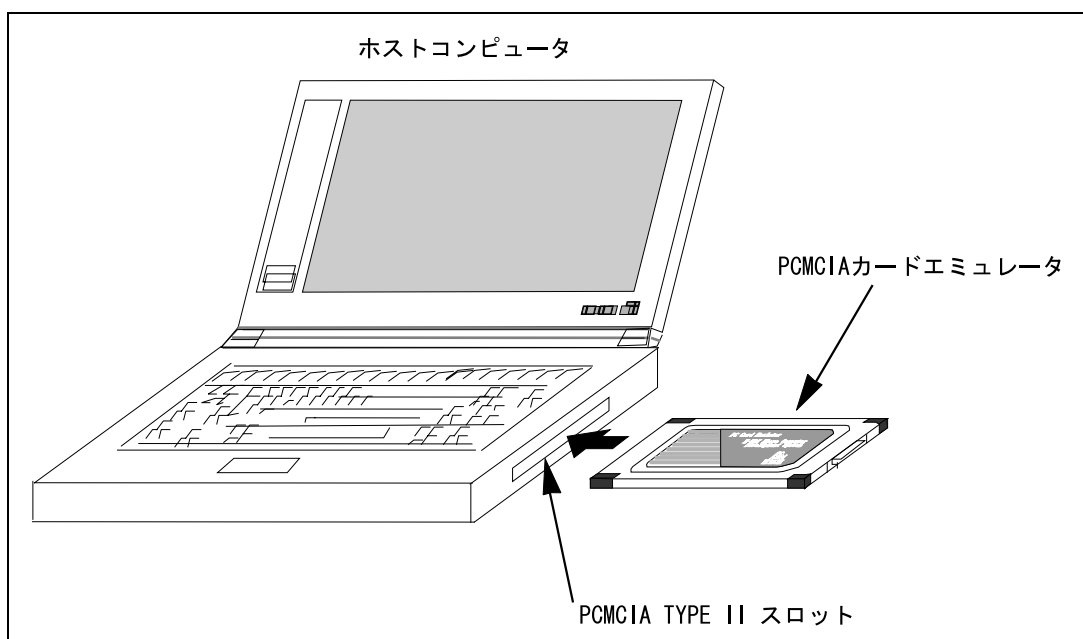


図 3.2 カードエミュレータのホストコンピュータへの挿入 (PCMCIA 使用時)

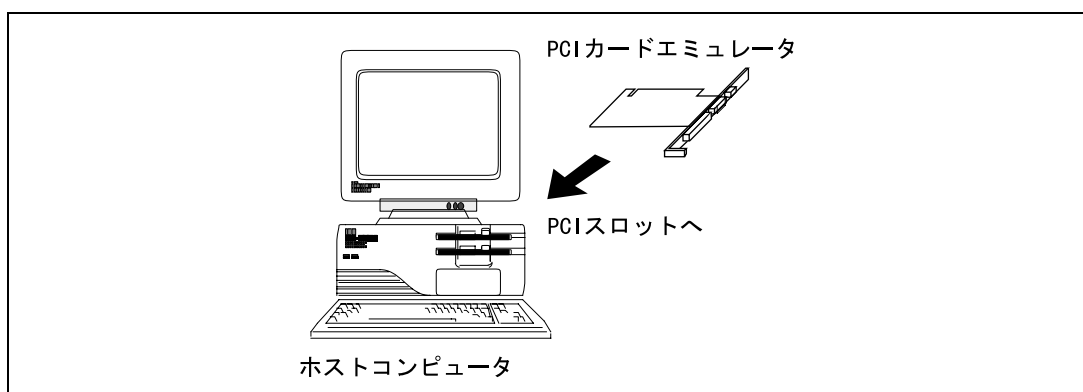


図 3.3 カードエミュレータのホストコンピュータへの挿入 (PCI 使用時)

3 使用前の準備

「3.3 カードエミュレータとユーザシステムとの接続」に示す手順でカードエミュレータとユーザシステムをユーザインタフェースケーブルで接続してください。

また、装置の移動などのためにカードエミュレータとユーザシステムを取り外したり、取り付ける場合も同様の手順で接続してください。

[留意事項]

PCI カードエミュレータをインストールする際、以下の点に注意してください。

1. ホストコンピュータの電源を切ってください。
2. PCI カードエミュレータを PCI スロットに平行に挿入してください。
3. コネクタとケーブル位置を確認し、確実に接続できるようにネジ止めしてください。

3.3 カードエミュレータとユーザシステムとの接続

- (1) コネクタをユーザシステム上に実装してください。カードエミュレータが推奨するコネクタを表3.1に示します。

表 3.1 推奨コネクタ

型 名	メーカ	仕様
7 6 1 4 - 6 0 0 2	住友スリーエム株式会社	14 ピンストレートタイプ

【留意事項】

コネクタ実装時、周囲 3 mm 四方に他の部品を実装しないでください。

- (2) コネクタのピン配置は、別冊の「H8S/xxxx E10A エミュレータ製品固有ガイド」の1.3章に示すように配置されています。
- (3) コネクタ使用時のユーザインタフェースケーブルの接続方法を図3.4に示します。ケーブルのGND線はユーザシステムのGNDに接続してください。 また、GNDにネジ止めする場合は $\varnothing 3$ (mm)のネジを用意してユーザシステムのGNDに接続してください。

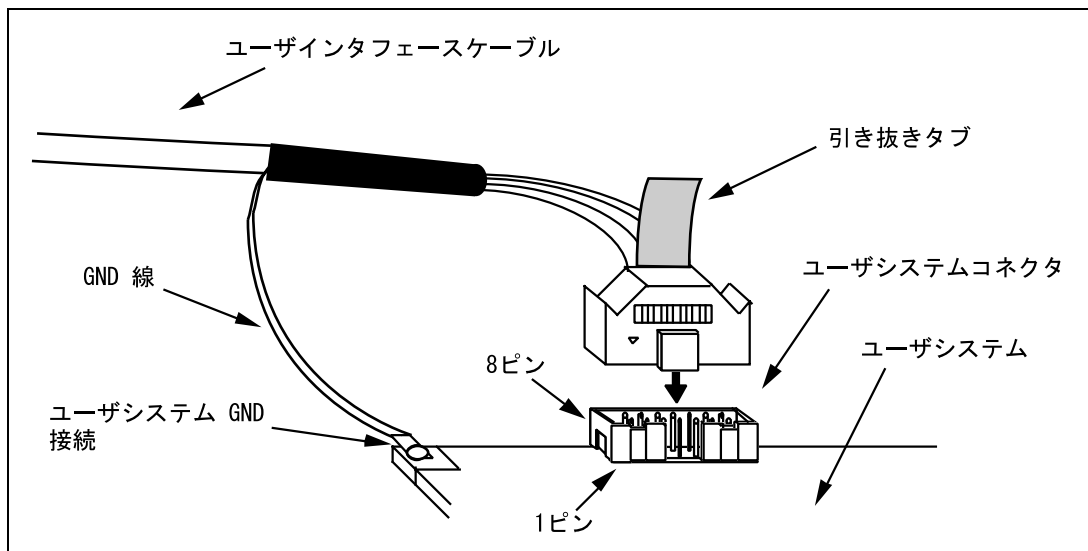


図 3.4 ユーザシステム側のユーザインタフェースケーブル接続方法

【留意事項】

1. コネクタの信号線の接続先は、パッケージによって異なります。MCU のピン配置を参照してください。
2. ユーザシステムからユーザインタフェースケーブルを引き抜く場合、必ず引き抜きタブを持ち、引き抜いてください。
3. エミュレータが動作する通信の範囲は、サポートする MCU によって異なります。
4. ユーザシステムにコネクタを接続する際、信号の配線は、別冊の「H8S/xxxx E10A エミュレータ製品固有ガイド」の 1.4 章に示すように接続してください。

3.4 システムチェック

次に、ソフトウェアを実行し、E10A 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、「4 デバッグの準備をする」を参照してください。

- (1) ホストコンピュータとE10Aエミュレータを接続してください。
- (2) E10A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) ユーザシステム側のコネクタにユーザインタフェースケーブルを接続します。
- (4) [スタート]メニューの[プログラム]から[Renesas High-performance Embedded Workshop 2]→[High-performance Embedded Workshop 2]を選択してください。

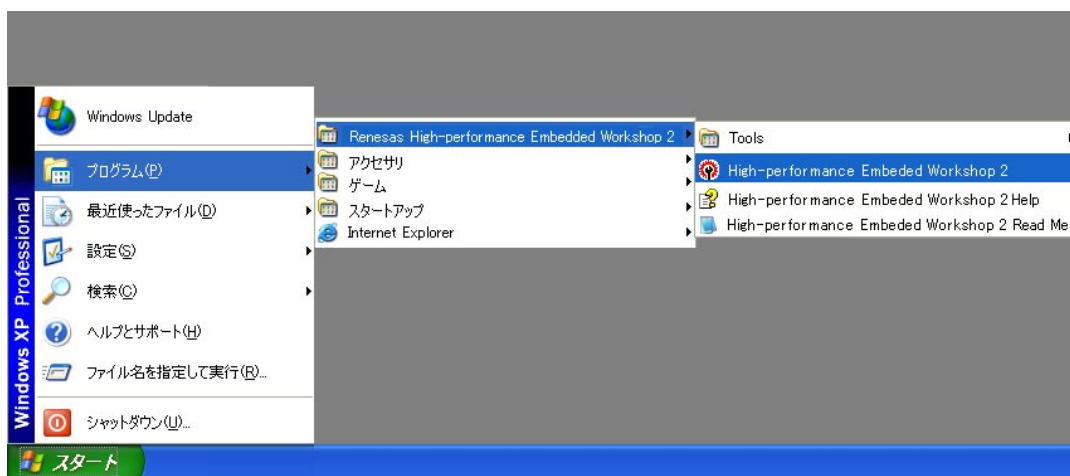


図 3.5 [スタート]メニュー

【留意事項】

[Renesas High-performance Embedded Workshop 2] -> [Tools] は、ご使用の環境によっては表示されません。

(5) [ようこそ!]ダイアログボックスが表示されます。

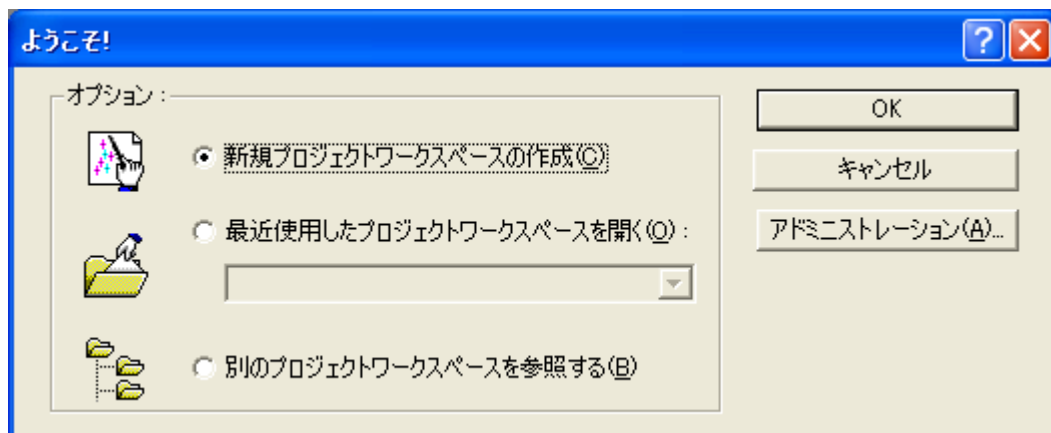


図 3.6 [ようこそ!]ダイアログボックス

- ・ [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、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。

HEW2 インストール先ディレクトリ：¥Tools¥Renesas¥DebugComp¥Platform¥E10A¥xxx¥Tutorial

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。

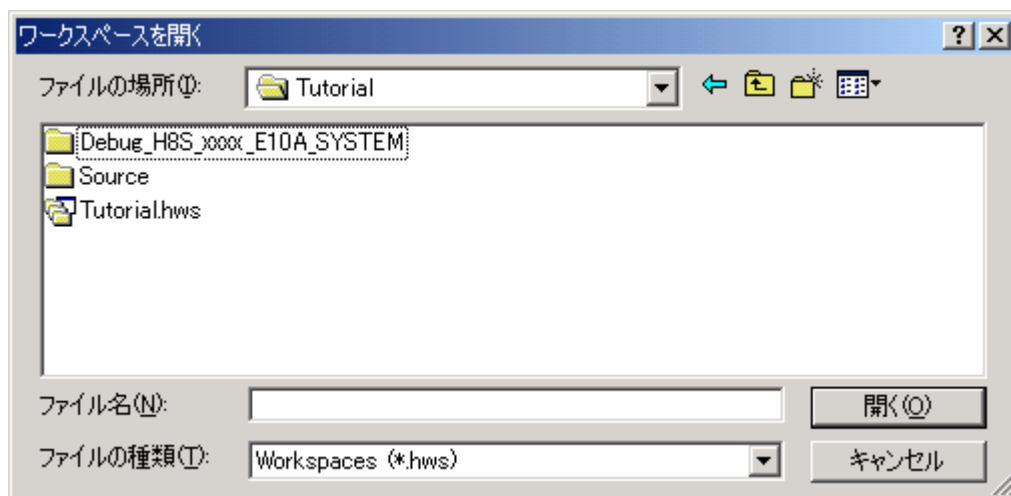


図 3.7 [ワークスペースを開く]ダイアログボックス

- (6) [Select Emulator mode]ダイアログボックスが表示されます。

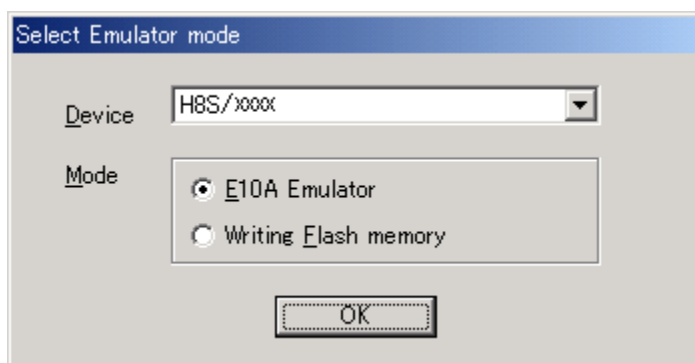


図 3.8 [Select Emulator mode]ダイアログボックス

[Device]ドロップダウンリストボックスでは、ご使用のデバイス名を選択してください。

[Mode]グループボックスで選択する内容は以下です。

- E10A Emulator
指定したデバイスのE10A Emulatorを起動します。
プログラムのデバッグが行えます。
- Writing Flash memory
フラッシュメモリへユーザプログラムを書込むだけのモードです。
プログラムのデバッグはできません。
ダウンロードするロードモジュールをワークスペースに登録し、ダウンロードしてください。

- (7) [Connecting]ダイアログボックスが表示され、エミュレータの接続を開始します。

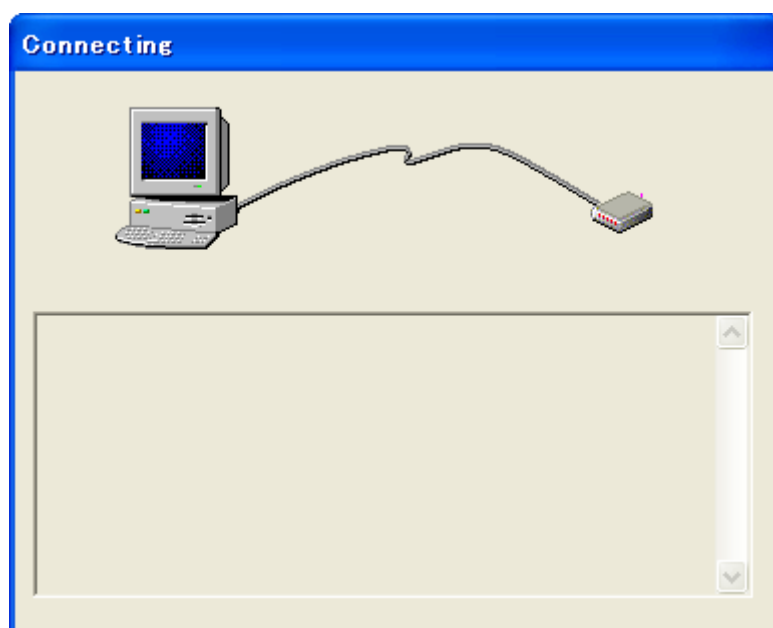


図 3.9 [Connecting]ダイアログボックス

- (8) 図3.10に示すダイアログボックスが表示されます。

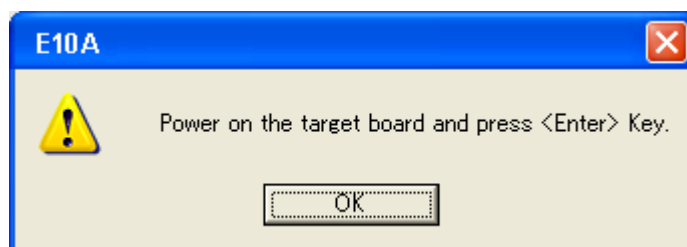


図 3.10 パワーオン要求メッセージのダイアログボックス

- (9) ユーザシステムの電源を入れます。
(10) ユーザシステムからRESET信号を入力し、[OK]ボタンをクリックします。
(11) 以下のダイアログボックスが表示されますので、ご使用のシステムクロック値を入力してください。

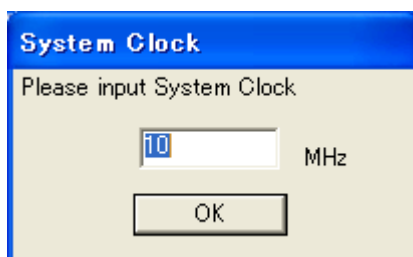


図 3.11 [System Clock] ダイアログボックス

- (12) 以下のダイアログボックスが表示されますので、IDコードを入力してください。
E10A Emulatorモードで起動した場合、設定したいIDコードを入力してください。

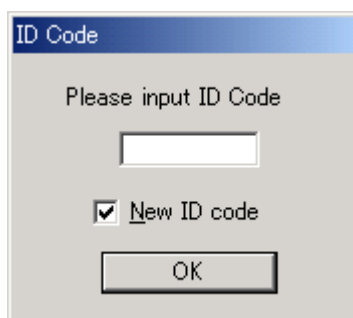


図 3.12 [ID Code] ダイアログボックス

3 使用前の準備

- (13) HEWの[Output]ウィンドウに”Connected”と表示されたら、E10Aエミュレータの起動は完了です。

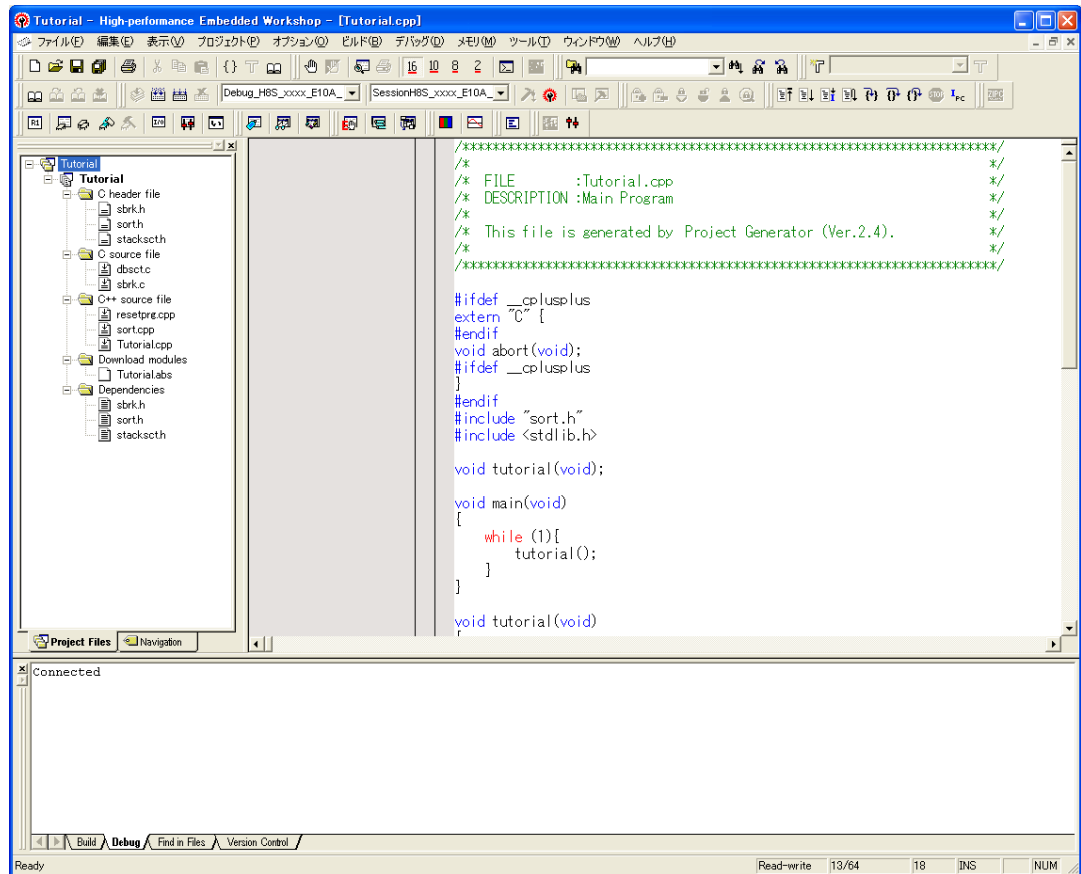


図 3.13 HEW 画面

【注】 ユーザプログラムが既にフラッシュメモリにダウンロードされている場合、E10A エミュレータ起動後にはユーザプログラムのデバッグ情報がないため、そのままではソースレベルデバッグが行えません。ソースレベルデバッグを行うためには、E10A エミュレータ起動後にデバッグ情報ファイルをロードしてください。

【留意事項】

1. ユーザシステムのコネクタ側にユーザインタフェースケーブルが接続されていないと、次のダイアログボックスが表示されます。

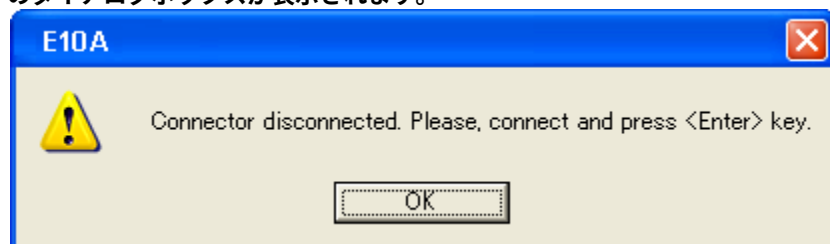


図 3.14 エラーメッセージ

2. E10A エミュレータが起動されない場合、次のダイアログボックスが表示されます。

- (a) 以下のダイアログボックスが表示された場合、フラッシュメモリの消去が行えません。フラッシュメモリの書換え回数が限界を越えていますので、MCU を交換してください。



図 3.15 エラーメッセージ

- (b) 以下のダイアログボックスが表示された場合、フラッシュメモリにライトできません。入力した CPU 動作周波数値が誤っているか、あるいはフラッシュメモリの書換え回数が限界を越えています。



図 3.16 エラーメッセージ

- (c) 以下のダイアログボックスが表示された場合、入力された ID コードが間違っています。この場合、フラッシュメモリのセキュリティのため、フラッシュメモリの内容をすべて消去します。

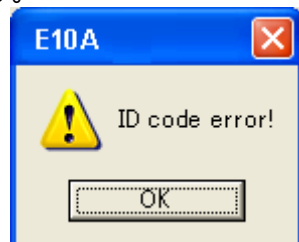


図 3.17 エラーメッセージ

- (d) 以下のダイアログボックスが表示された場合、MCU と E10A エミュレータとの通信がとれません。MCU が正しく動作できる設定になっているかどうか確認してください。

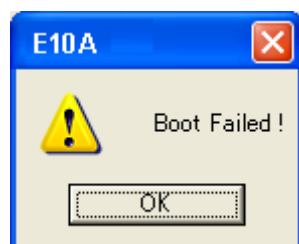


図 3.18 エラーメッセージ

3. ドライバが正しく設定されていない場合、次のダイアログボックスが表示されます。

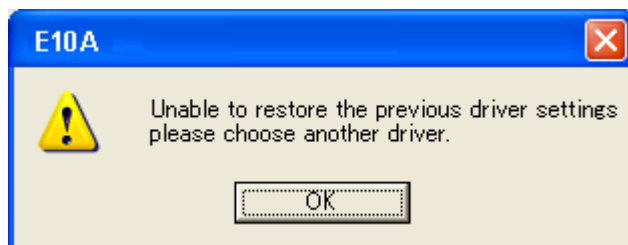


図 3.19 エラーメッセージ

3.5 アンインストール

ご使用の PC からインストールした E10A エミュレータ用ソフトウェアを削除します。

HEW は、インストールした製品を把握していますので、必ず HEW 画面上の操作でアンインストールを行ってください。

1. HEWを起動してください。
2. [ようこそ!]ダイアログボックスの、[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。

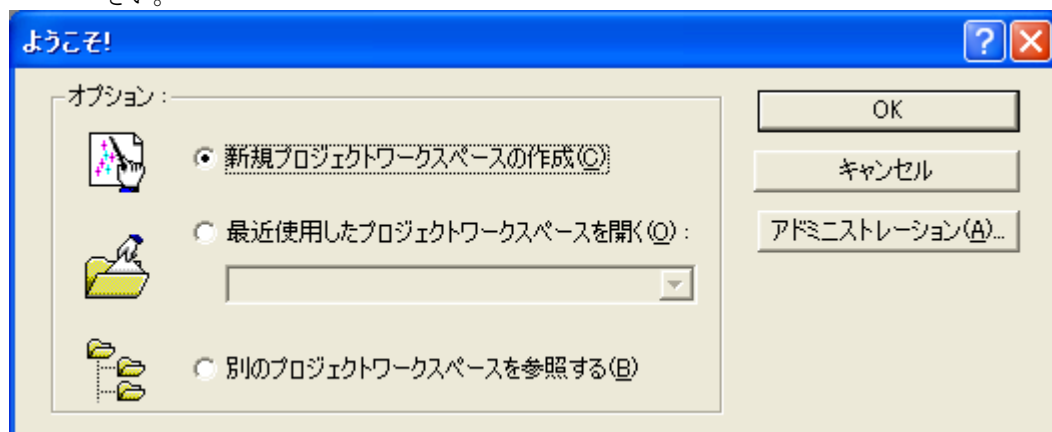


図 3.20 [ようこそ!]ダイアログボックス

3. [ツールアドミニストレーション]ダイアログボックスが開きます。

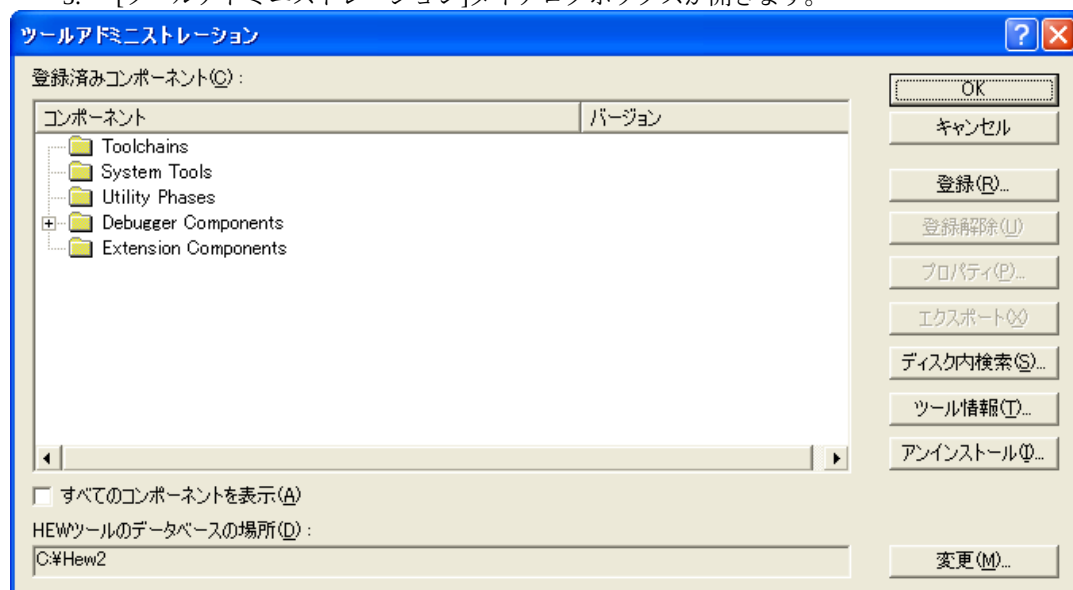


図 3.21 [ツールアドミニストレーション]ダイアログボックス

3 使用前の準備

4. [登録済みコンポーネント]リストボックス内の[Debugger Components]の左側にある[+]マークをクリックしてインストール済みコンポーネントを一覧表示し、アンインストールしたい製品名を強調表示してください。

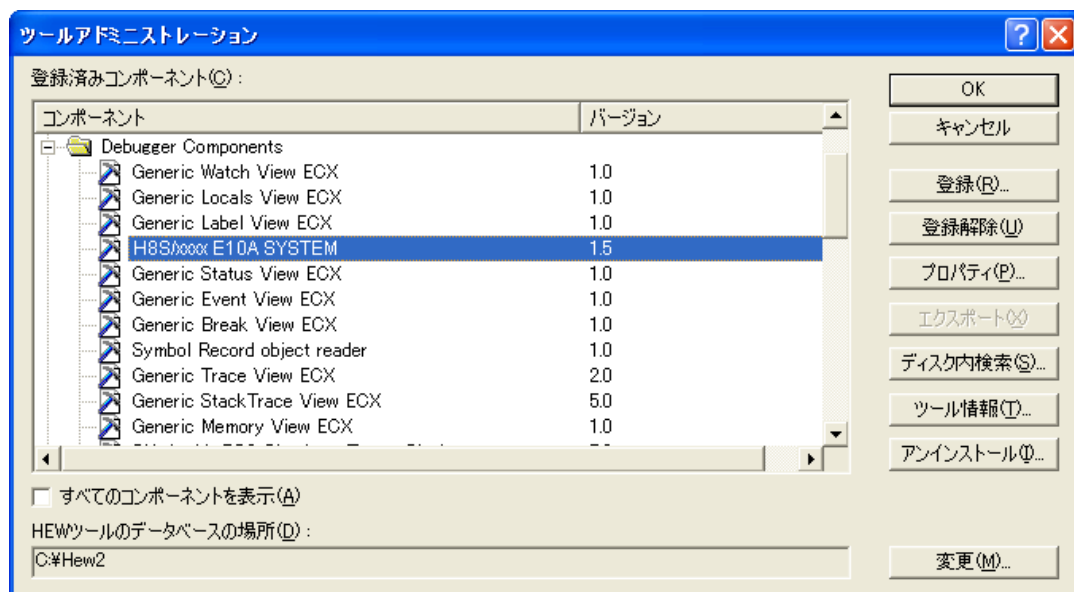


図 3.22 アンインストールする製品の強調表示

5. 次に、[登録解除]ボタンをクリックしてください。
以下のメッセージボックスが表示されますので、[はい]ボタンを押してください。

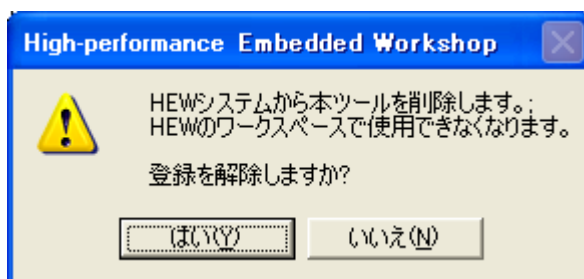


図 3.23 メッセージボックス

これで、HEW への登録解除は完了です。
次に、PC 内から E10A エミュレータ用のファイルを削除します。

6. [ツールアドミニストレーション]ダイアログボックスの[アンインストール]ボタンを押してください。[HEWツールのアンインストール]ダイアログボックスが開きます。

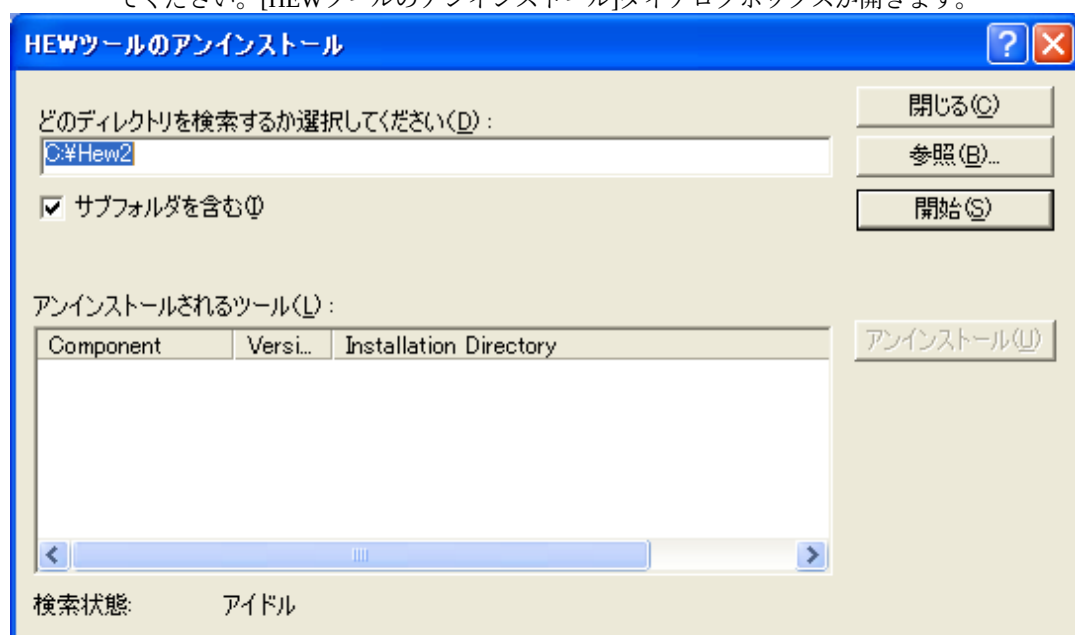


図 3.24 [HEW ツールのアンインストール]ダイアログボックス

7. [開始]ボタンを押してください。インストールされている製品が一覧表示されます。

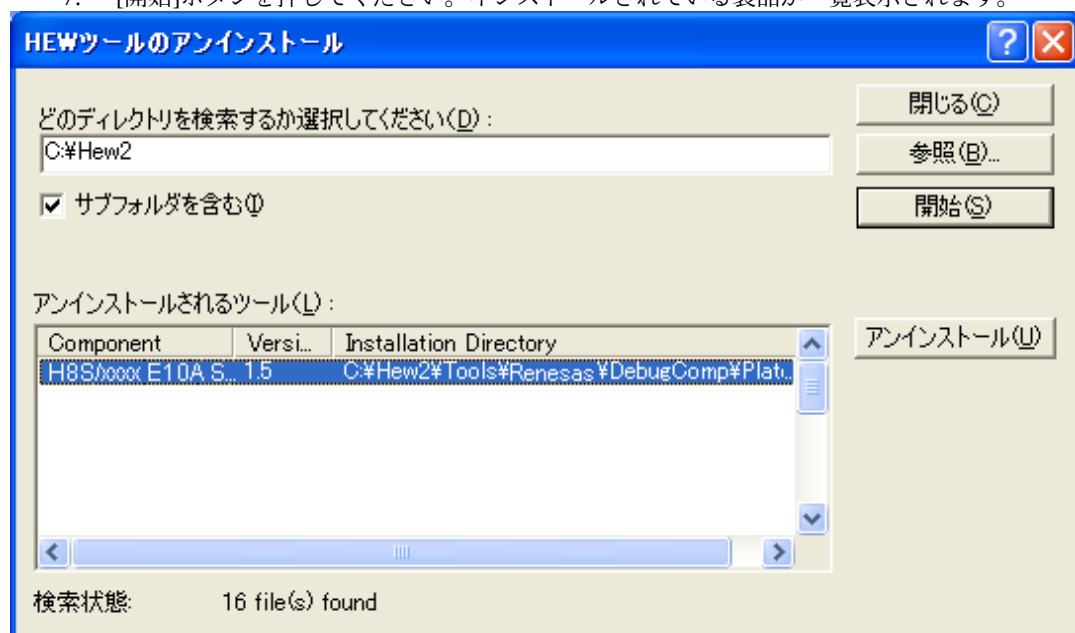


図 3.25 アンインストールする製品の強調表示

8. アンインストールしたい製品名を強調表示し、[アンインストール]ボタンを押してください。

これでアンインストールは完了です。

注意

アンインストール中に共有ファイルの検出が行われる場合があります。
他の製品が共有ファイルを使用する可能性がある場合は共有ファイルの削除を行わないでください。アンインストールにより他の製品が起動しなくなった場合は他の製品を再インストールしてください。

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。

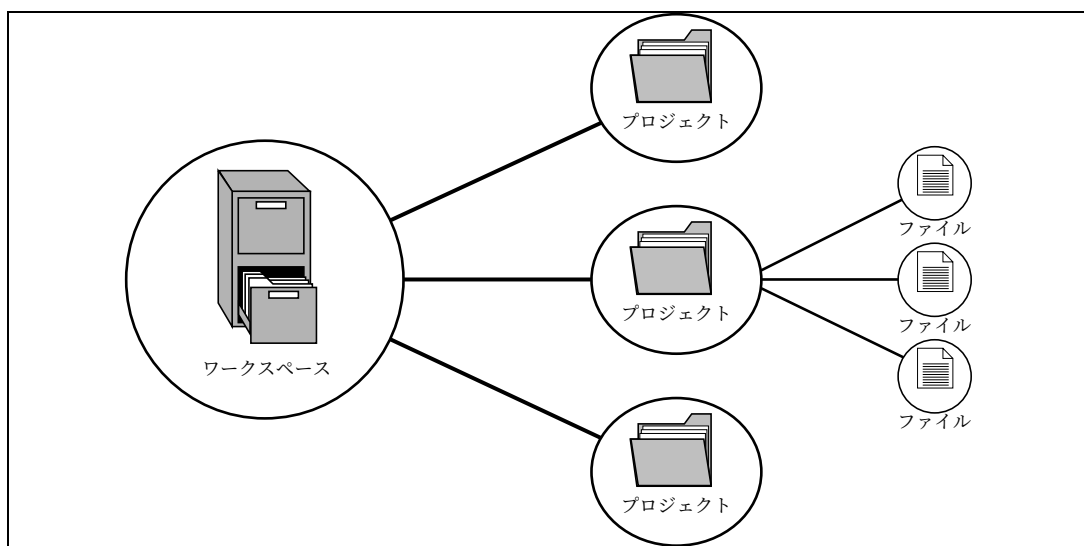


図 4.1 ワークスペース、プロジェクト、ファイルの構成

ワークスペースでは関連したプロジェクトを1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E10Aエミュレータ、ユーザシステムを接続し、ユーザシステムの電源を入れてください。
- (2) [スタート]メニューの[プログラム]から[Renesas High-performance Embedded Workshop 2]→[High-performance Embedded Workshop 2]を選択してください。
- (3) [ようこそ!]ダイアログボックスが表示されます。

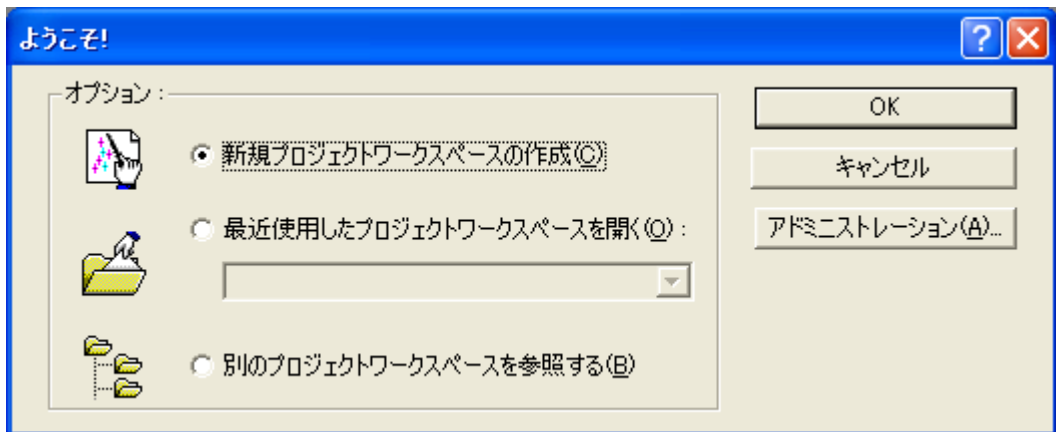


図 4.2 [ようこそ!]ダイアログボックス

- ・[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。

[新規プロジェクトワークスペースの作成]を選択しツールチェインを使用しない場合と、[新規プロジェクトワークスペースの作成]を選択し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]を選択した場合の起動について説明します。[最近使用したプロジェクトワークスペースを開く]は、[別のプロジェクトワークスペースを参照する]を選択した場合のワークスペースファイルの指定が省略された動作となります。

4.2.1 新規にワークスペースを作成する場合(ツールチェーン未使用)

(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。

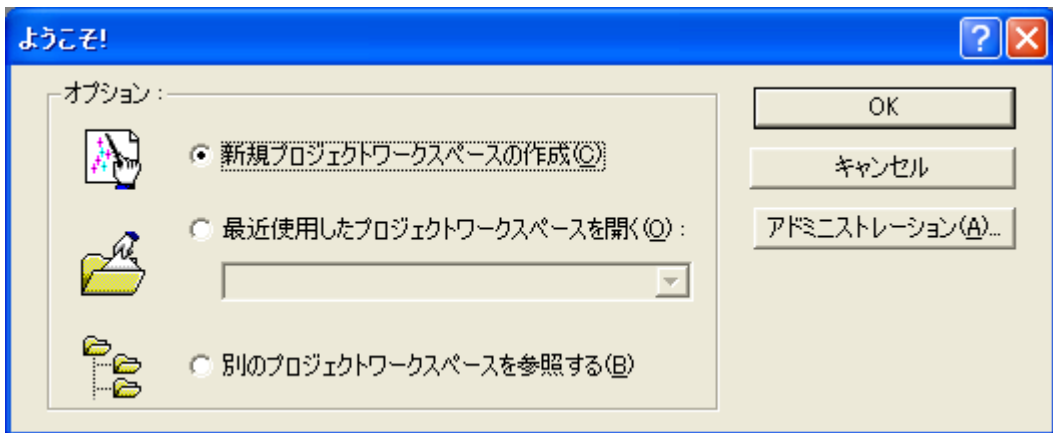


図 4.3 [ようこそ!]ダイアログボックス

4 デバッグの準備をする

(2)Project Generator が開始されます。

ここでは、ツールチェーン用の設定に関する説明は省略します。

ツールチェーンをご購入されていない場合、以下の画面が開きます。

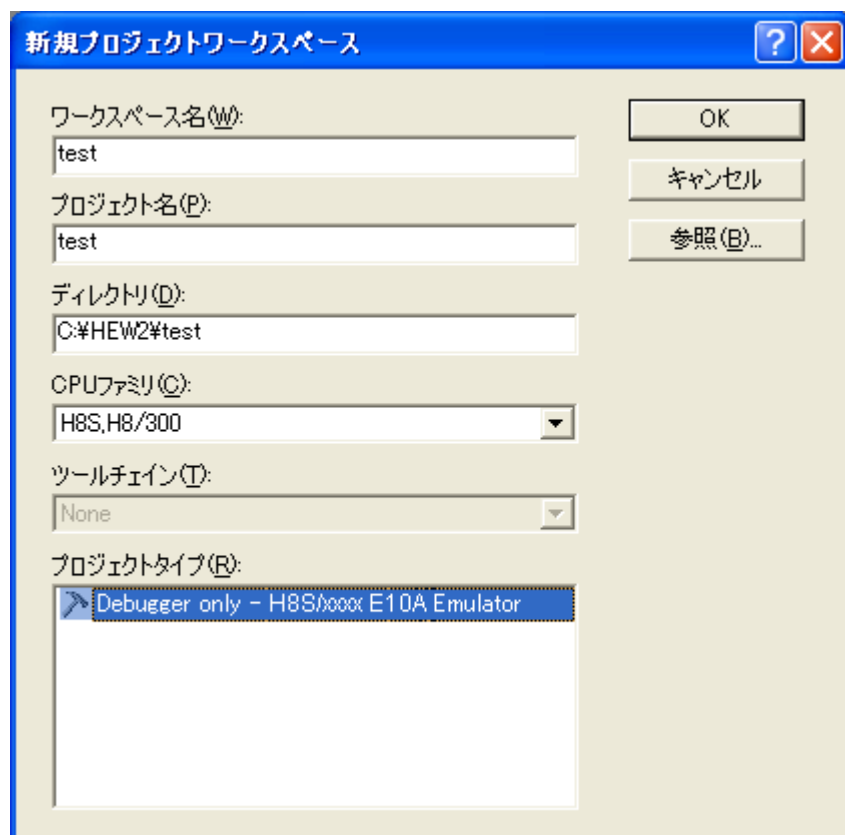


図 4.4 [新規プロジェクトワークスペース]ダイアログボックス

- ・[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。ここでは“test”と入力します。
- ・[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。

その他のリストボックスはツールチェーン設定用ですので、ツールチェー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、以下の画面が表示されます。

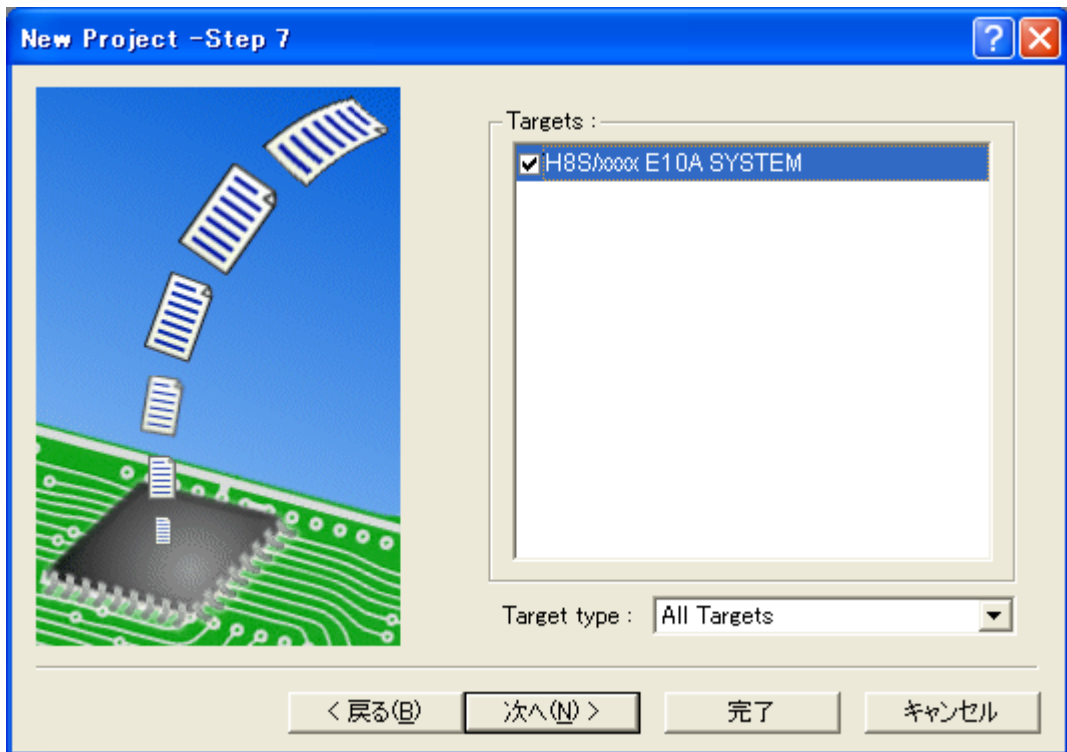


図 4.5 [New Project-Step 7]ダイアログボックス

ここで、H8S/xxx E10A SYSTEM にチェックし、[次へ]ボタンを押してください。

(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。

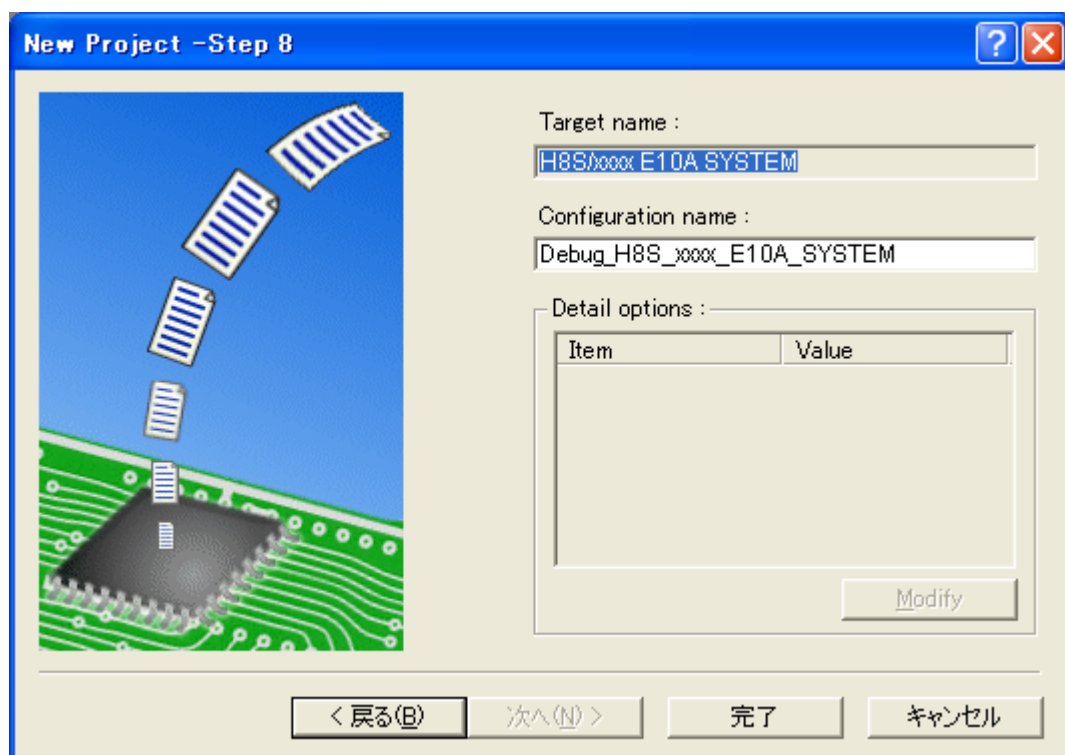


図 4.6 [New Project-Step 8]ダイアログボックス

これで E10A エミュレータに関する設定は終了です。

[完了]ボタンを押し、Project Generator を終了してください。HEW が起動します。

(5) HEW 起動後、自動的に E10A エミュレータが接続されます。

接続中の操作については、「3.4 システムチェック」を参照してください。

4.2.2 新規にワークスペースを作成する場合(ツールチェーン使用)

(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。

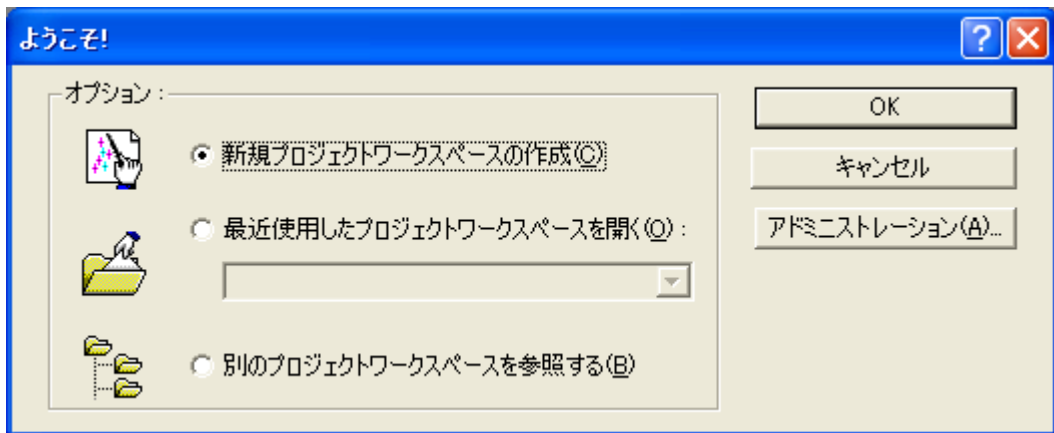


図 4.7 [ようこそ!]ダイアログボックス

(2) Project Generator が開始されます。

Project Generator の詳細については、ツールチェーンご購入時に含まれる、「High-performance Embedded Workshop2 チュートリアル」を参照してください。

ここでは、ツールチェーン用の設定に関する説明は省略します。

ツールチェーンをご購入されている場合、以下の画面が開きます。

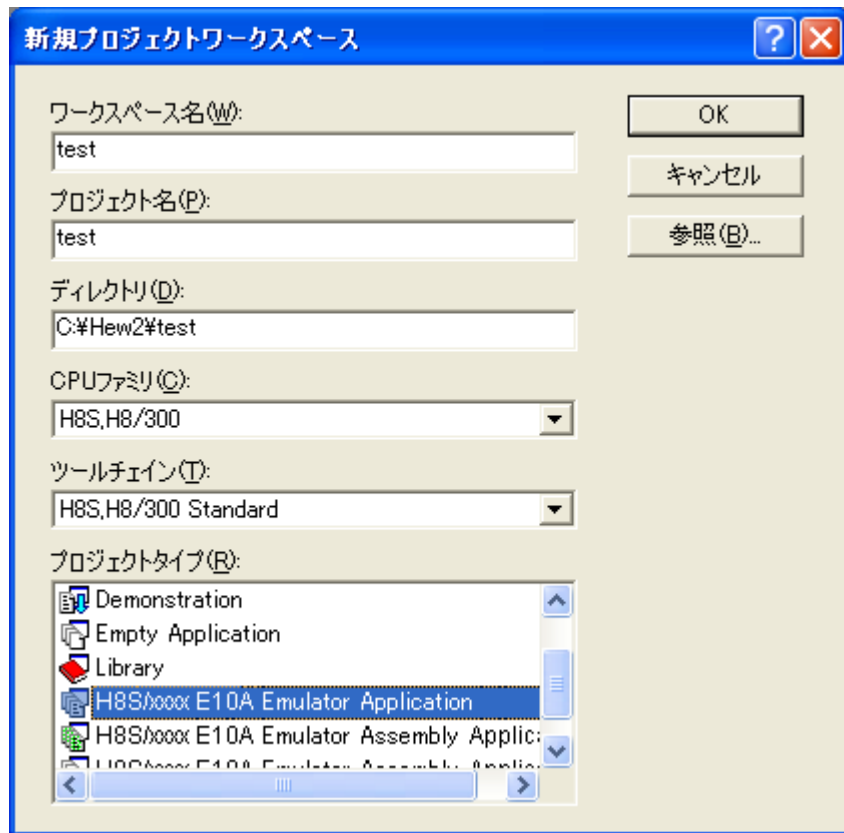


図 4.8 [新規プロジェクトワークスペース]ダイアログボックス

- ・[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。ここでは“test”と入力します。
- ・[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。
- ・[CPU ファミリ]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。
- ・[ツールチェーン]ドロップダウンリストボックス
ツールチェーンをご使用になる場合、該当するツールチェー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. E10A エミュレータの場合、以下のどれかを選択しても同じです。
 - ・ [Application]と[H8S/xxxx E10A Emulator Application]
 - ・ [Assembly Application]と[H8S/xxxx E10A Emulator Assembly Application]
 - ・ [Empty Application]と[H8S/xxxx E10A Emulator Empty Application]
 2. E10A 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 Simulator 用のプログラムです。生成されるプログラムを使用される場合、“Printf 文”を削除してください。
- (3)次に、ツールチェーンの設定を行いますので、必要な設定を行ってください。
ツールチェーンの設定が終了したら、以下の画面が表示されます。

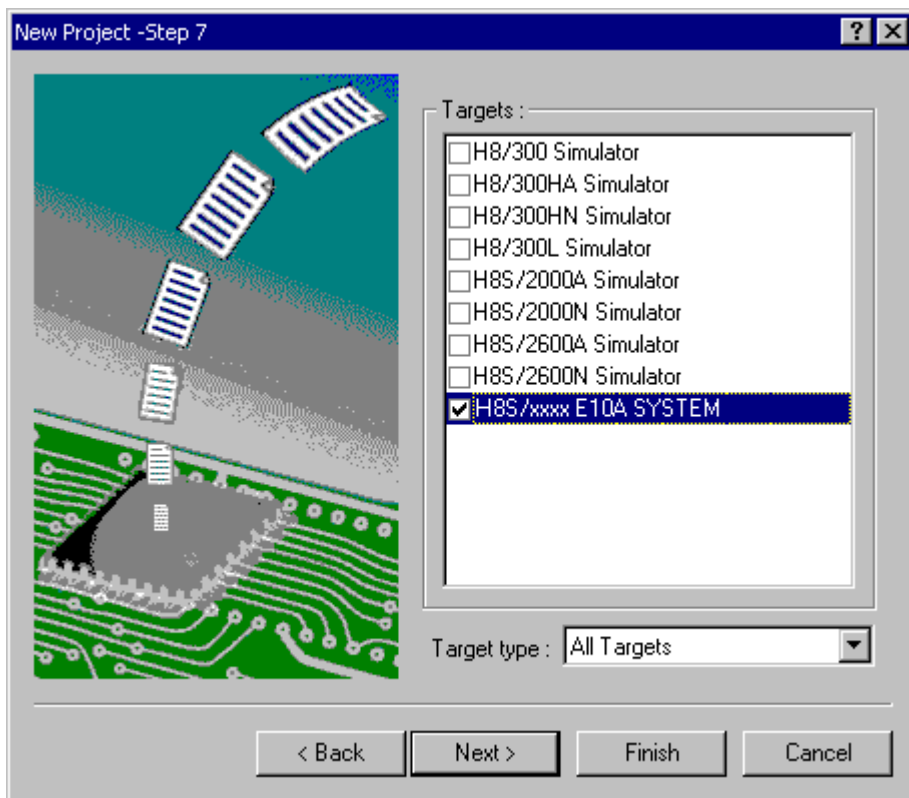


図 4.9 [New Project-Step 7]ダイアログボックス

ここで、H8S/xxxx E10A SYSTEM にチェックし、[Next]ボタンを押してください。
必要であれば、他の製品にもチェックをしてください。

4 デバッグの準備をする

(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。

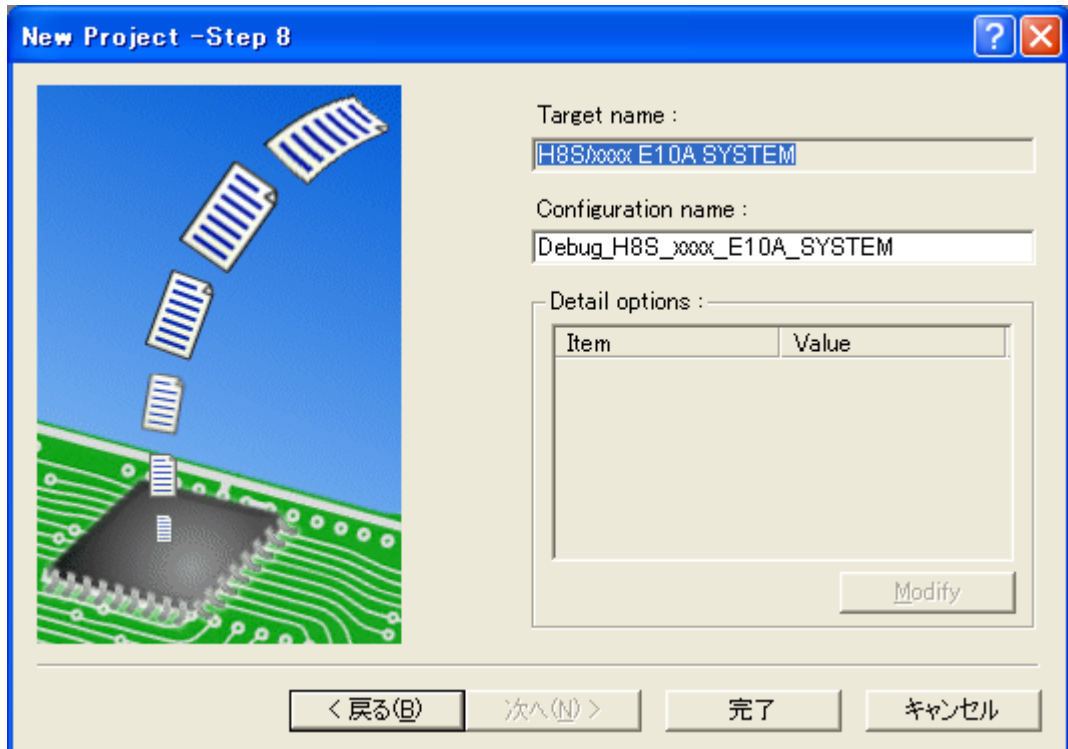


図 4.10 [New Project-Step 8]ダイアログボックス

これで E10A 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、Project Generator を終了してください。HEW が起動します。

(5) HEW 起動後、E10A エミュレータを接続してください。

E10A 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。

E10A エミュレータを接続する場合は、以下のどちらかの操作をしてください。

接続中の操作については、「3.4 システムチェック」を参照してください。

(a)E10A 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]メニューの[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ー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 E10A 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、E10A エミュレータが接続されます。

(b)E10A 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

E10A 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、E10A エミュレータを簡単に接続できます。

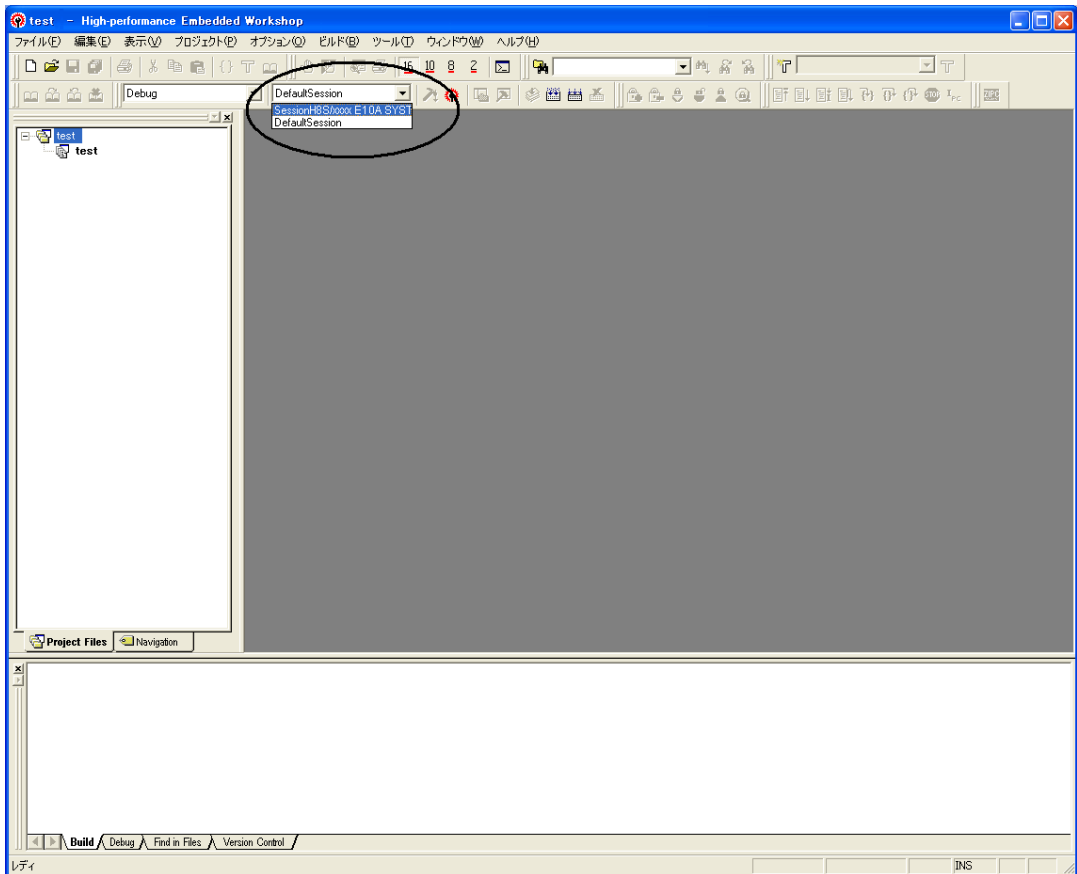


図 4.11 セッションファイルの選択

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、E10A 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、E10A エミュレータが自動的に接続されます。

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。

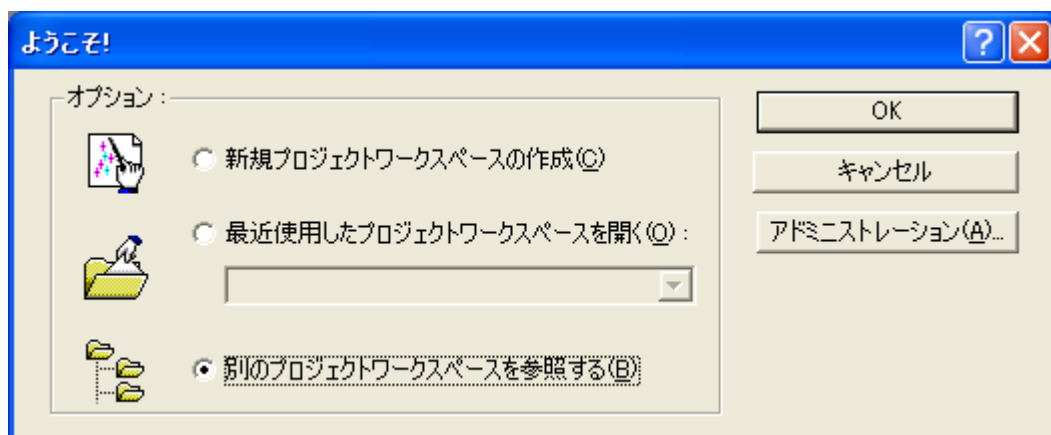


図 4.12 [ようこそ!]ダイアログボックス

- (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。

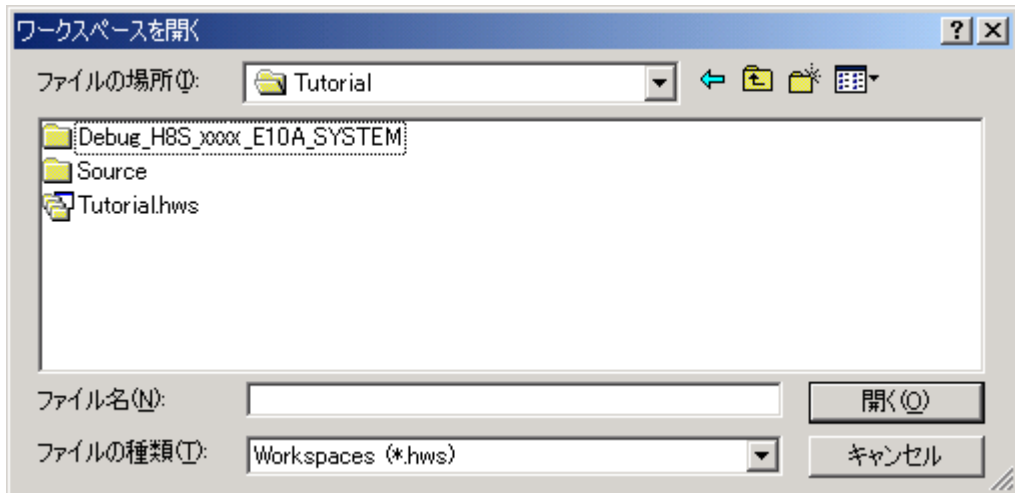


図 4.13 [ワークスペースを開く]ダイアログボックス

- (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 E10A エミュレータ起動時の設定

E10A エミュレータの起動時、コマンドチェー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、ワークスペースウィ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。

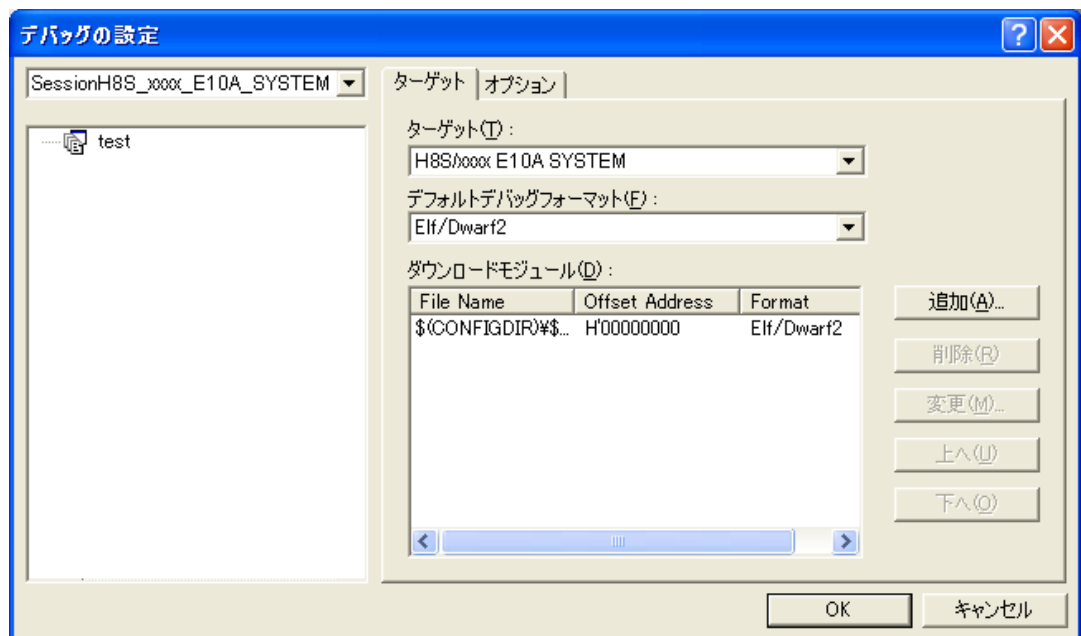


図 4.14 [デバッグの設定]ダイアログボックス ([ターゲット]ページ)

[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。

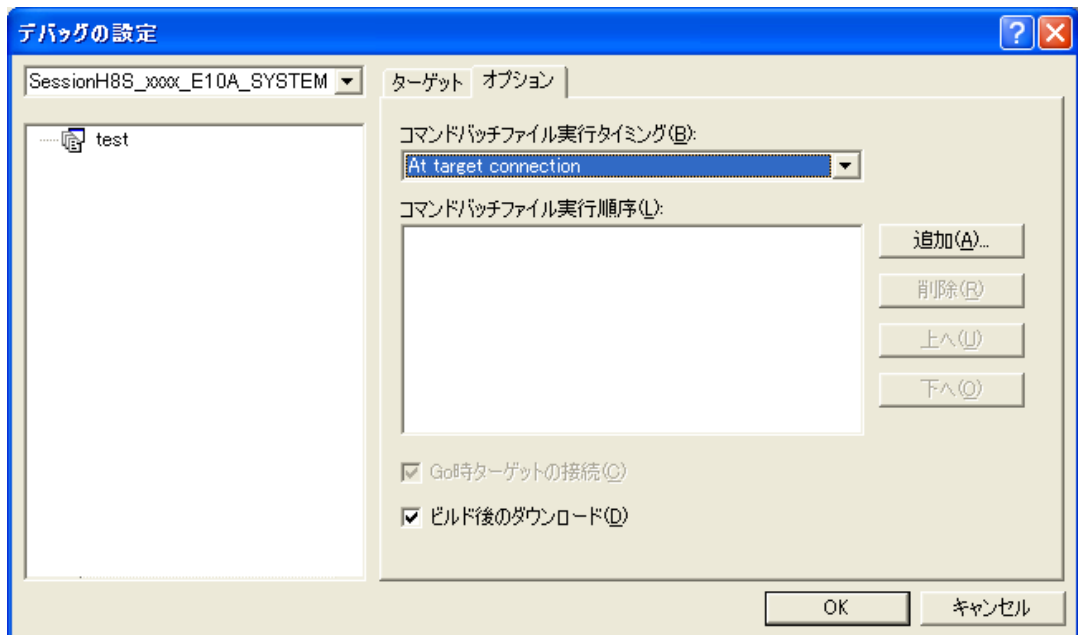


図 4.15 [デバッグの設定]ダイアログボックス ([オプション]ページ)

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ーンを登録します。
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、コマンドチェー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ー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、ビルド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、デバッガ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、デバッグングプラットフォーム、ダウンロードするプログラム、各デバッグ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（図 4.16）からセッションを選んでください。



図 4.16 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 4.17）。

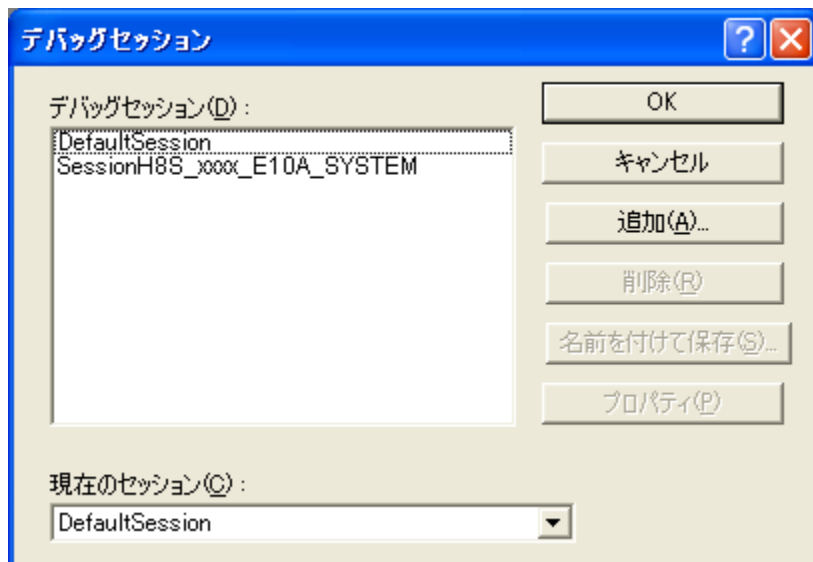


図 4.17 [デバッグセッション]ダイアログボックス

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 4.17)。
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 4.18)。
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。

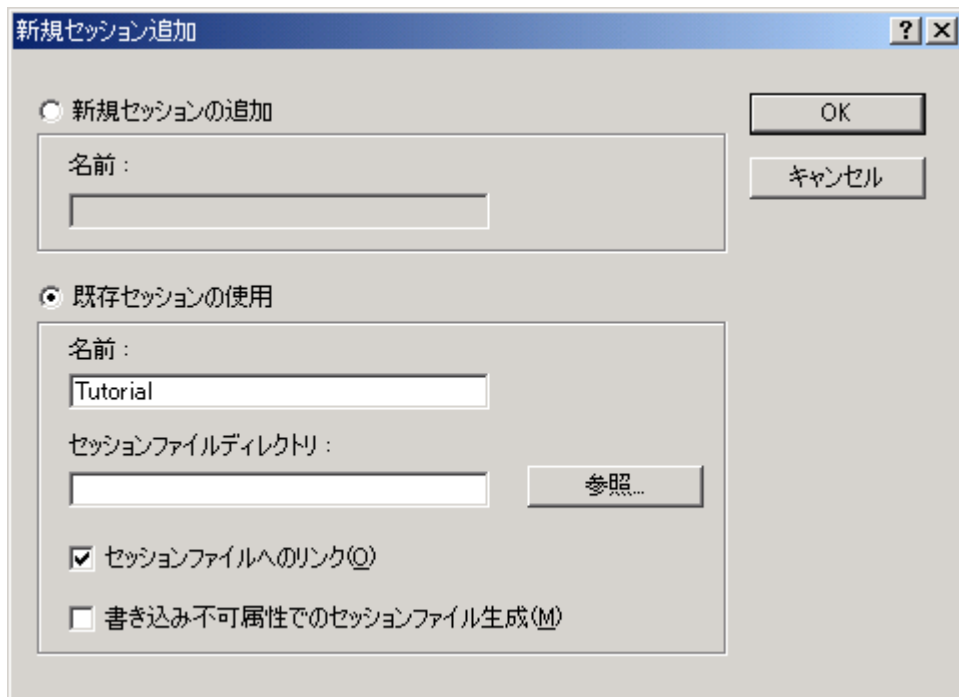


図 4.18 [新規セッション追加]ダイアログボックス

・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 4.17)。
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 4.18)。
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。

4 デバッグの準備をする

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ブラウザしてください。
[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 4.17)。
2. 削除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 4.17)。
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 4.19)。

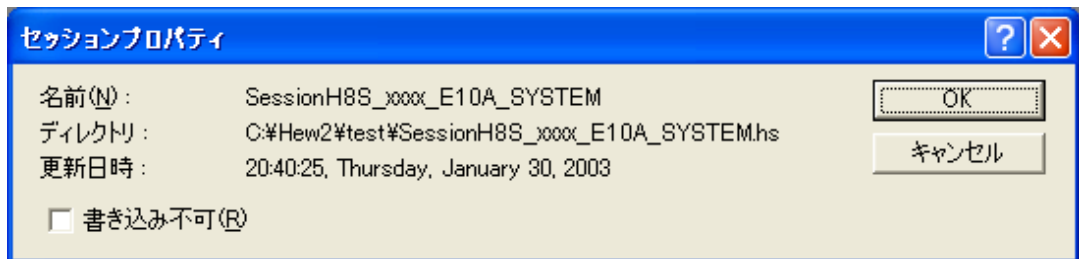


図 4.19 [セッションプロパティ]ダイアログボックス

・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 4.17)。
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 4.19)。
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、デバッグ設定ファイルを共有する場合、およびデータを間違え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 4.17)。
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(図 4.20)。
4. 新しいファイルの場所をブラウズしてください。
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で使いたい場合は、[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. [保存]ボタンをクリックしてください。

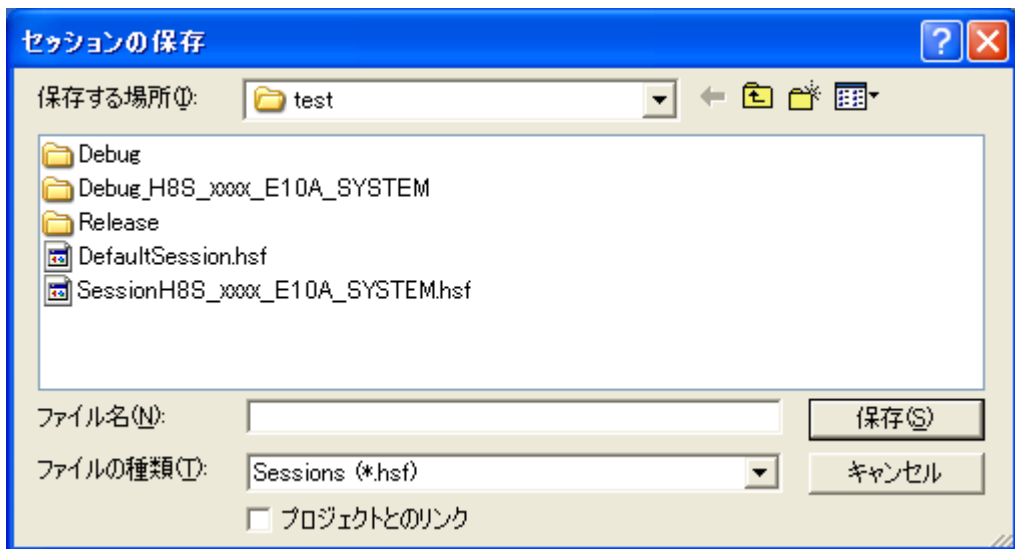


図 4.20 [セッションの保存]ダイアログボックス

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

[ファイル->セッションの保存]を選んでください。

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) E10A 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]メニューの[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ー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 E10A 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、E10A エミュレータが接続されます。

(2) E10A 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

E10A 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、E10A エミュレータを簡単に接続できます。

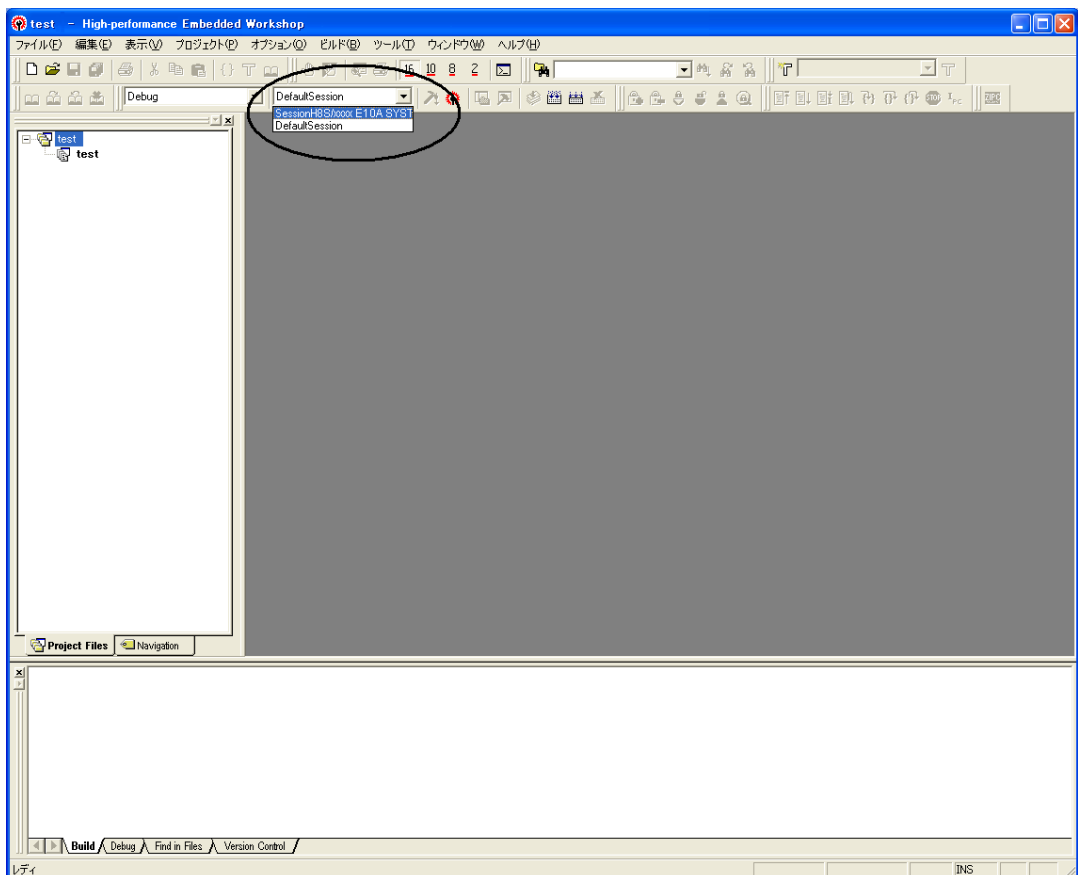


図 4.21 セッションファイルの選択

図 4.21 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、E10A 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、E10A 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 エミュレータの終了

ツールチェーンをご使用の場合、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

[オプション]メニューから[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。(図 4.14 [デバッグの設定]ダイアログボックス ([ターゲット]ページ) 参照)

[ターゲット]ドロップダウンリストボックスで<None>を選択するか、他製品を選択してください。他製品を選択した場合、起動中のエミュレータの接続を解除した後、選択した製品の接続が開始されます。

(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、[Yes]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、[No]ボタンをクリックしてください。HEW は終了します。

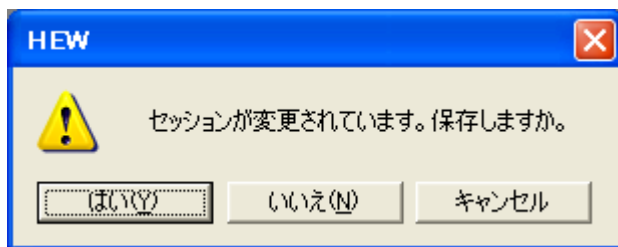


図 4.22 メッセージボックス

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]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]ダイアログボックスが開きます。

5.1.2 [General]ページ

[General]ページでは、E10A エミュレータの基本設定を行います。

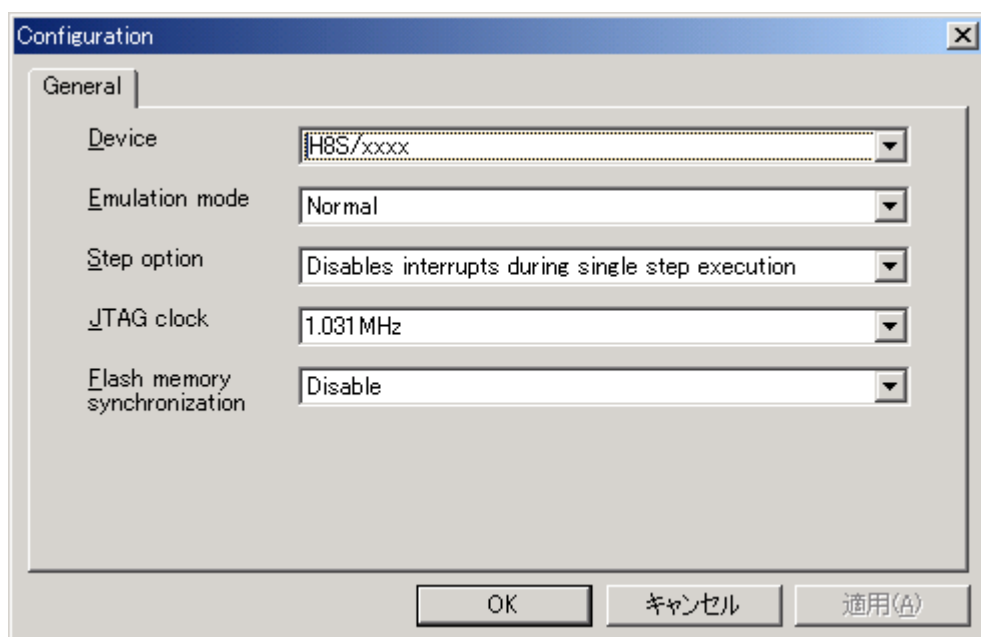


図 5.1 [Configuration]ダイアログボックス ([General]ページ)

【注意事項】

本ダイアログボックスは、製品ごとに異なります。各製品の表示内容については、オンラインヘルプを参照してください。

設定できる項目は以下の通りです。

[Device]コンボボックス	MCU デバイス名を表示します。
[Emulation mode] コンボボックス	ユーザプログラム実行時のエミュレーションモードを選択します。 Normal: 通常の実行を行います。 No break: ソフトウェアブレークポイント、およびハードウェア ブレークポイントを一時的に無効にしてユーザプログラム を実行します。
[Step option]コンボボックス	ステップ中の割込みの開放／マスクを設定します。 Disables interrupts during single step execution: ステップ中の割込みをマスクします。 Enables interrupts during single step execution: ステップ中の割込みを開放します。
[JTAG clock]コンボボックス	JTAG の周波数*を設定します。
[Flash memory synchronization] コンボボックス	ホストコンピュータ、フラッシュメモリ間の同期方法を設定しま す。 ホストコンピュータからフラッシュメモリへの同期を行うと、ユー ザプログラム停止時にフラッシュ書き込みのために待ち時間が生 じますが、表示内容とフラッシュメモリが常に一致します。 フラッシュメモリからホストコンピュータへの同期を行うと、ユー ザプログラム停止時にフラッシュメモリの内容を読み出されるの でユーザプログラムモードにある書き替え内容が反映されます。 Disable : E10A 起動時およびフラッシュメモリ領域の変更時以外は 同期を行いません。 PC to Flash memory : ホストコンピュータからフラッシュメモリへの同期を行 います。 Flash memory to PC : フラッシュメモリからホストコンピュータへの同期を行 います。 PC to Flash memory、Flash memory to PC : ホストコンピュータ、フラッシュメモリ間の同期を行 います。

【注】 JTAG が動作する周波数の範囲は、サポートするデバイスによって異なります。

5.2 プログラムを表示する

プログラムをソースコードおよびアセンブリ言語コード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もともと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 ソースコードを表示する

ソースファイルを選択して[開く]ボタンをクリックすると、HEWは、統合化エディタのファイルを開きます。または、[Workspace]ウィンドウのソースファイルをダブルクリックすることによって表示することができます。

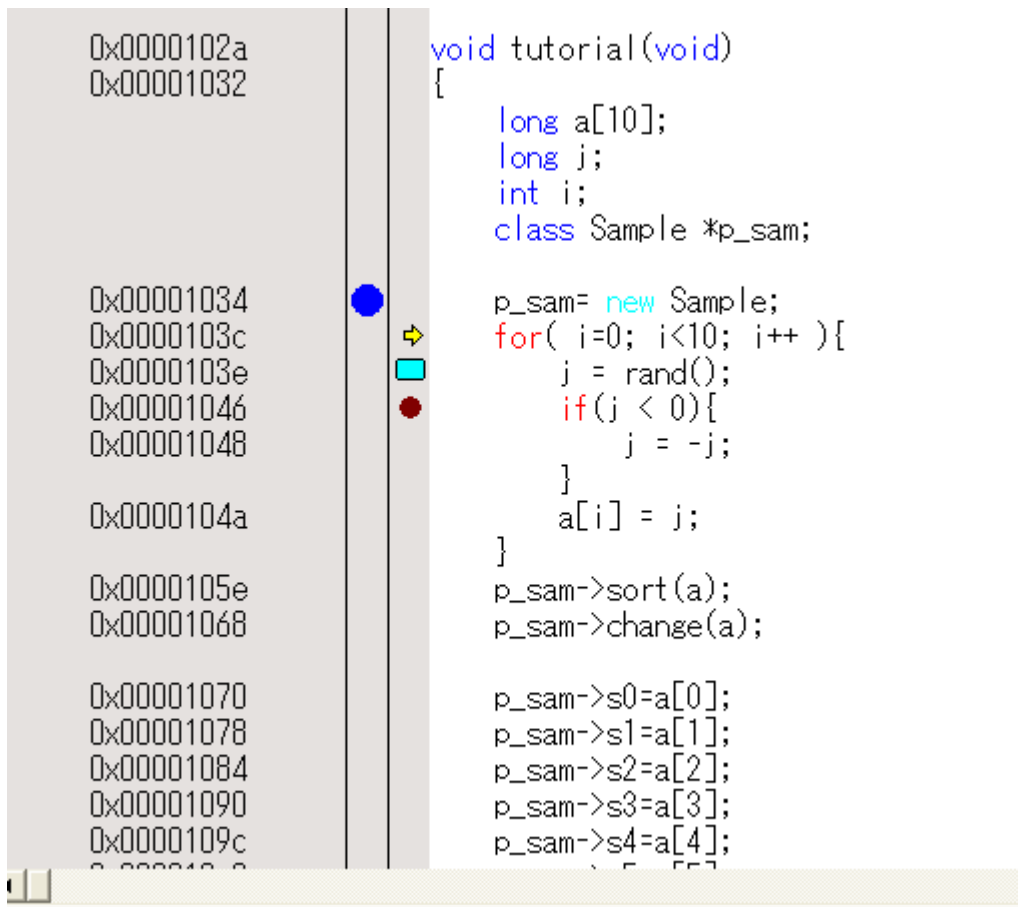


図 5.2 [Source]ウィンドウ

本ウィンドウでは左はじに行情報として下記を表示します。

- | | | |
|------|----------------------|----------------------|
| 1 列目 | (Source address カラム) | アドレス情報 |
| 2 列目 | (Event カラム) | イベント情報(ブレイクコンディション) |
| 3 列目 | (Editor カラム) | PC、ブックマーク、ブレイクポイント情報 |

右側のソース表示画面を[Source]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 値やブレイ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。

- ブレイクコンディションのアドレス条件を設定します。2 箇所のみ設定できます。

この設定は、ポップアップメニューからも可能です。

Eventカラムをダブルクリックすることによって、上記のビットマップが現れます。
この設定は、ポップアップメニューからも可能です。

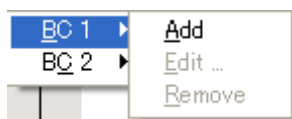


図 5.3 ポップアップメニュー

【留意事項】

1. Edit メニューや[Eventpoint]ウインドウによって、各チャネルの条件にアドレス条件以外を追加した場合、Event カラムの表記は消えます。
2. ブレイクコンディションの設定条件はデバイスによって異なります。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。

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. [Source]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。
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。

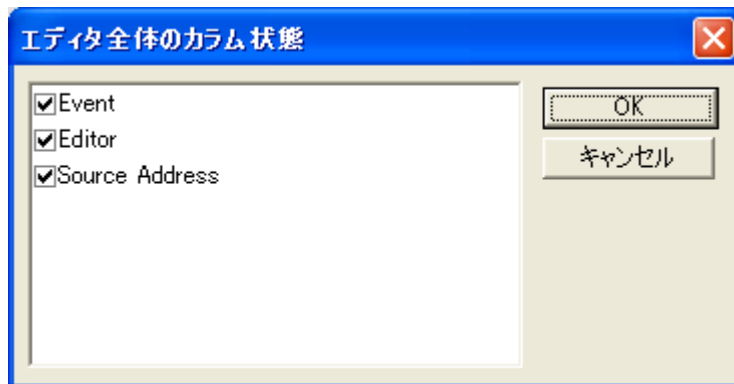


図 5.4 [エディタ全体のカラム状態]ダイアログボックス

⇒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。

5.2.2 アセンブリ言語コードを表示する

ソースファイルが開いているときは、右ボタンをクリックしてポップアップメニューを開き、[逆アセンブリ]を選択してください。開いているソースファイルに対応する[Disassembly]ウィンドウを表示します。

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、[Disassembly]ツールバーボタンをクリックします。

[Disassembly]ウィンドウは現在の PC の場所で開きます。また、ディスアセンブルモニタ(可能なときはラベルも一緒に)を表示する[Address], [Code] (オプション)を表示します。

また、[Disassembly]ウィンドウのポップアップメニューから[混合表示]を選択すると、ソースとコードの両方を表示することができます。以下は[混合表示]を選択した場合の表示例です。

```
22: void main(void)
23: {
24:     while (1){
00001028 40FC          BRA      @_main:8
26:     }
27: }
28:
29: void tutorial(void)
tutorial 5E0010F4      JSR      @$sp_regsv$3:24
0000102E 79370028      SUB.W    #H'0028,R7
30: {
00001032 0D74          MOV.W    R7,R4
31:     long a[10];
32:     long j;
33:     int i;
34:     class Sample *p_sam;
35:
36:     p_sam= new Sample;
00001034 1900          SUB.W    R0,R0
00001036 5E002000      JSR      @Sample::Sample():24
0000103A 0D06          MOV.W    R0,R6
37:     for( i=0; i<10; i++ ){
➔ 0000103C 19EE          SUB.W    E6,E6
```

図 5.5 [Disassembly]ウィンドウ

5.2.3 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。

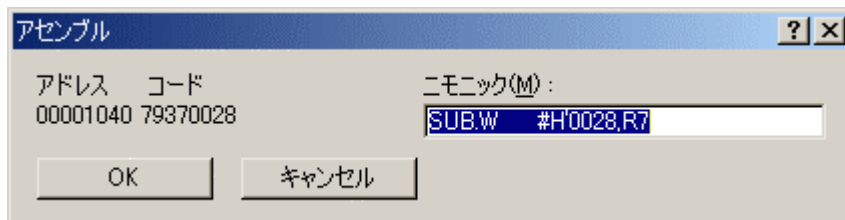


図 5.6 [アセンブル]ダイアログボックス

アドレス、機械語コード、およびディスアセンブル命令を表示します。新しい命令を入力するか、[ニモニック]フィールドの古い命令を編集します。“Enter”キーを押すと、命令をメモリにアセンブルして、次の命令に移ります。[OK]ボタンをクリックすると、命令をメモリにアセンブル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か“Esc”キーを押すと、ダイアログボックスが閉じます。

【注】 アセンブリ言語表示は、デバッグプラットフォームのメモリの実際の機械語コードからディスアセンブルします。メモリの内容を修正すると、ダイアログボックス（および [Disassembly] ウィンドウ）には新しいアセンブリ言語コードを表示します。しかし、[Source] ウィンドウの表示内容は変更しません。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。

5.2.4 特定のアドレスを見る

[Disassembly]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場所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 5.7 [アドレス指定]ダイアログボックス

エディットボックスにアドレスまたはラベル名を入力して、[OK]ボタンをクリックするか“Enter”キーを押します。[Disassembly]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。詳細は「5.12.3 複数ラベルをサポートする」で説明します。

5.2.5 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場所では、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を開いて"#pc"という式を入力すると、[Source]または[Disassembly]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、[Command Line]ウィンドウを使用します。

5.3.1 [Command Line]ウィンドウを開く

[Command Line]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。

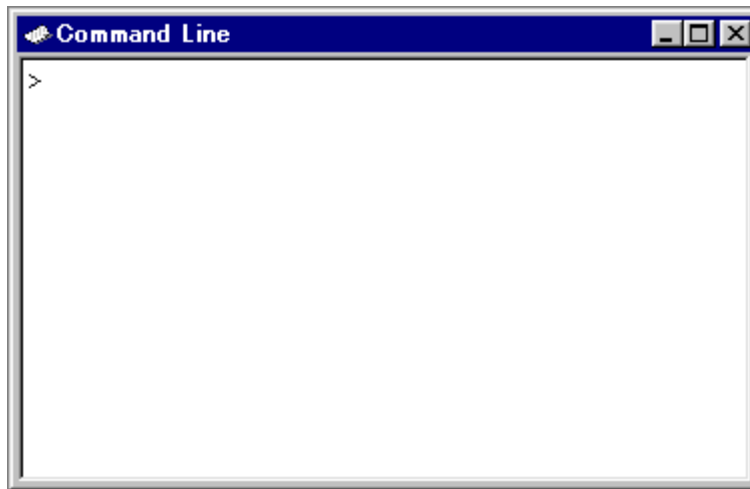


図 5.8 [Command Line]ウィンドウ

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。

[Command Line]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、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。

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。

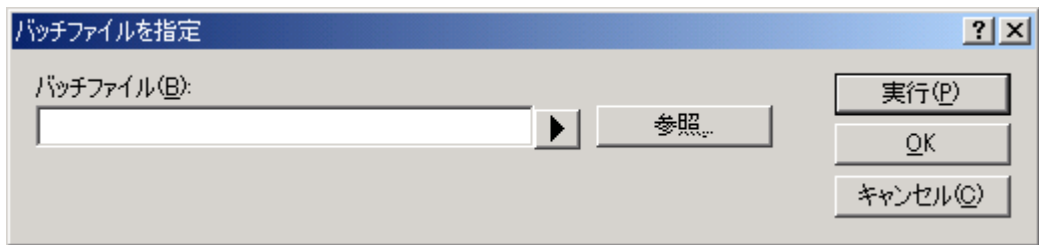


図 5.9 [バッチファイルを指定]ダイアログボックス

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。

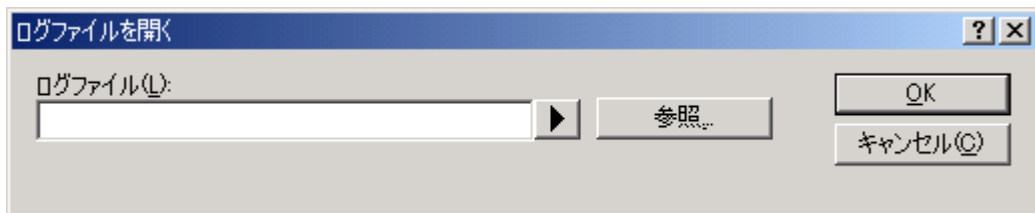


図 5.10 [ログファイルを開く]ダイアログボックス

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、[Command Line]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、ファイルをブラウズ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、[Register]ウィンドウを使用して行います。

5.4.1 [Register]ウィンドウを開く

[Register]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。[Register]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。

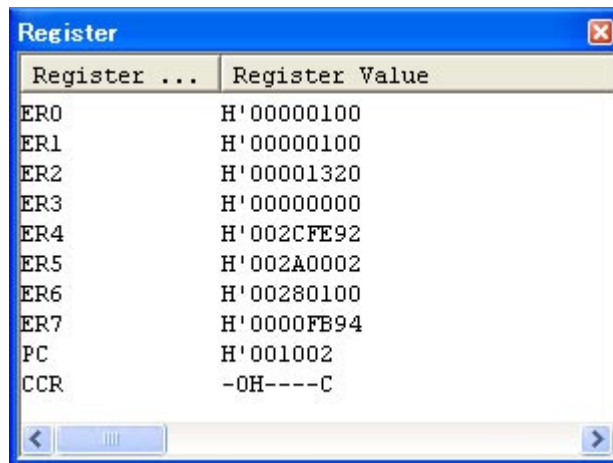


図 5.11 [Register]ウィンドウ

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[Edit Register]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。

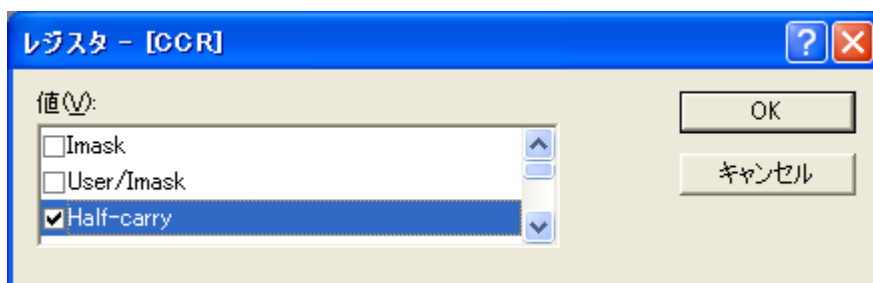


図 5.12 ビットレジスタの拡張

5.4.3 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。

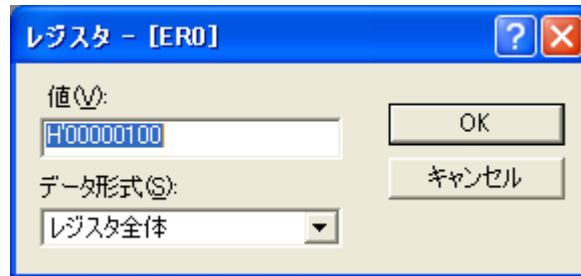


図 5.13 [レジスタ編集]ダイアログボックス

[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。

5.4.4 レジスタの内容を使用する

[Disassembly]または[Memory]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。

5.5 メモリを操作する

この節では、CPU のアドレス空間におけるメモリ領域の見方を説明します。ここでは、1つのメモリ領域を異なるフォーマットで見る方法、メモリブロックをフィルおよび、移動する方法、ならびにディスクファイルを使用してメモリ領域をロードおよび、ベリファイする方法を説明します。

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[表示アドレス]ダイアログボックスを開きます。

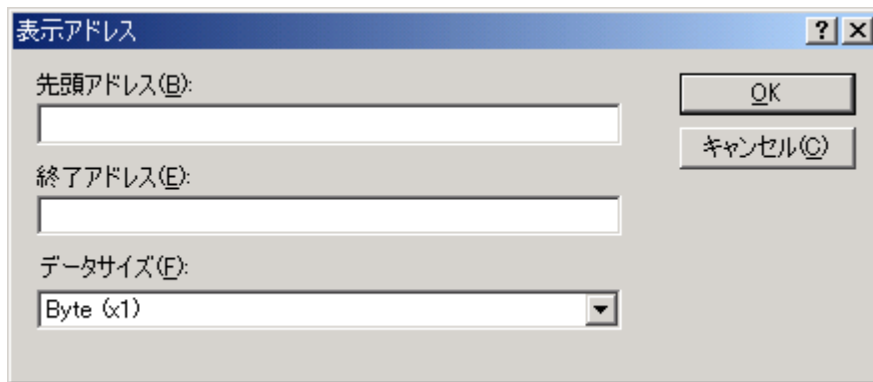


図 5.14 [表示アドレス]ダイアログボックス

[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]ドロップダウンリストから表示するデータサイズを選択します。[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[Memory]ウィンドウが開きます。入力した表示開始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。

Memory				
Address	Data		Value	
0x00000A0C	6DF3	0120	6DF4	?G?
0x00000A12	7937	0028	0D74	?{?
0x00000A18	7900	0001	5E00	?□?
0x00000A1E	0CB6	7900	1F42	???
0x00000A24	6DF0	5E00	0BB4	???
0x00000A2A	0B87	1955	5E00	???
0x00000A30	0C80	17F0	0F86	???
0x00000A36	4C02	17B6	0D53	???
0x00000A3C	1053	0943	0100	??A

図 5.15 [Memory]ウィンドウ

カラムを3つ表示します。

1. [Address] この行の[Data]カラムの最初のアイテムのアドレス
2. [Data] デバッ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

[Memory]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。

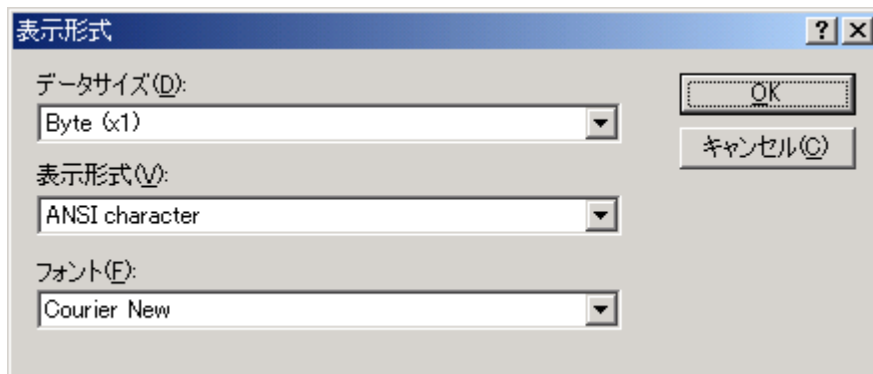


図 5.16 [表示形式]ダイアログボックス

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2つ目の[表示形式]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。

[表示形式]カラムのフォントを、データ表示に使用するフォントと異なるようにすることができます。これは、データを[Word]フォーマットで表示したときに[double]バイト文字値を表示するのに便利です。

5.5.3 ウィンドウを分割表示する

[Memory]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。分割バーをウィンドウの上端または下端に移動すると分割表示を解除します。

5.5.4 異なるメモリ領域を見る

[Memory]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、[表示アドレス]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[アドレス設定]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、[Memory]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([Memory]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。

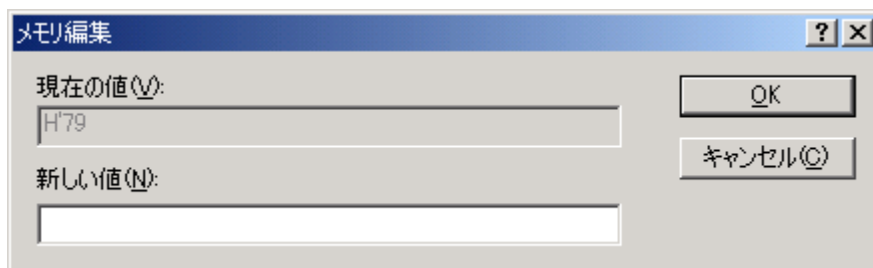


図 5.17 [メモリ編集]ダイアログボックス

[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[Memory]ウィンドウにある場合、最初のメモリユニット ([Memory]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。

メモリアドレス範囲が[Memory]ウィンドウよりも大きい場合、または[Memory]ウィンドウにはない場合、[表示アドレス]ダイアログボックスのそれぞれのフィールドに開始アドレスおよびデータサイズを入力することができます。

5.5.7 メモリ内の値を探す

メモリ内の値を探すには、[Memory]ウィンドウを開き、ポップアップメニューから[検索]を選択します。

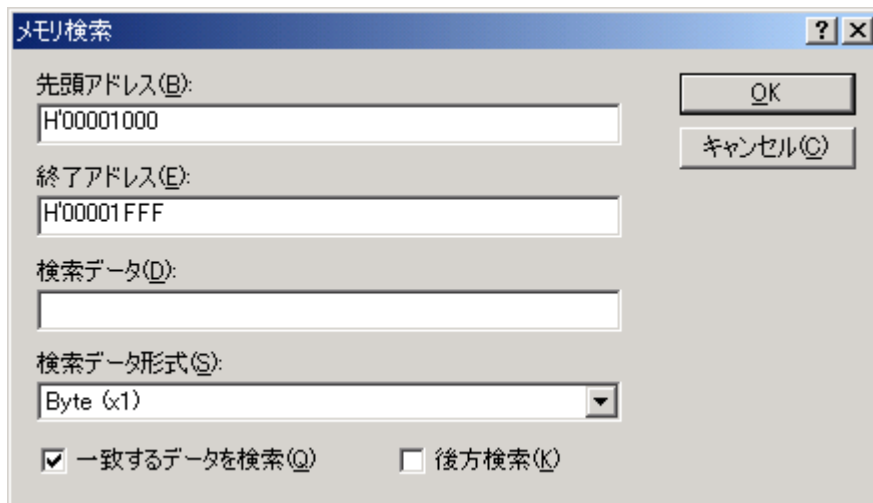


図 5.18 [メモリ検索]ダイアログボックス

検索するアドレス範囲の先頭および終了アドレス([Memory]ウィンドウの中のメモリ領域を選択している場合には、開始および終了アドレス値を自動的にフィルします)、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、データが見つかったアドレスから [Memory] ウィンドウに表示します。

データを見つけることができなかった場合、[Memory] 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、[Memory] 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。

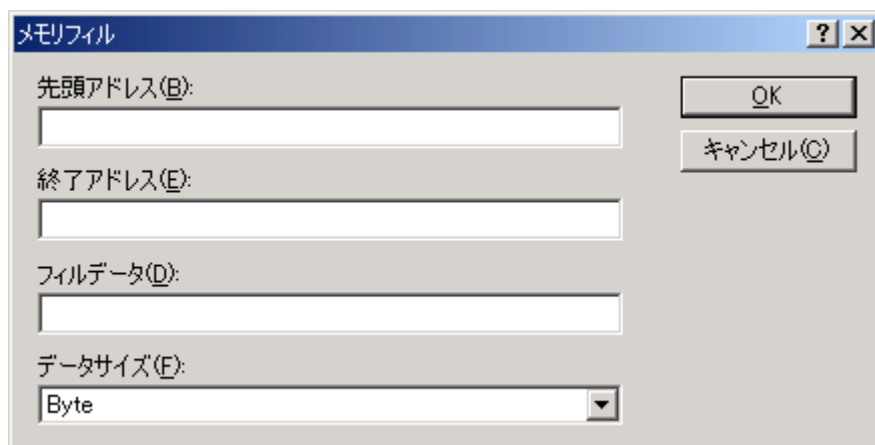


図 5.19 [メモリフィル]ダイアログボックス

[Memory]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...]を選択すると、[メモリコピー]ダイアログボックスを表示します。

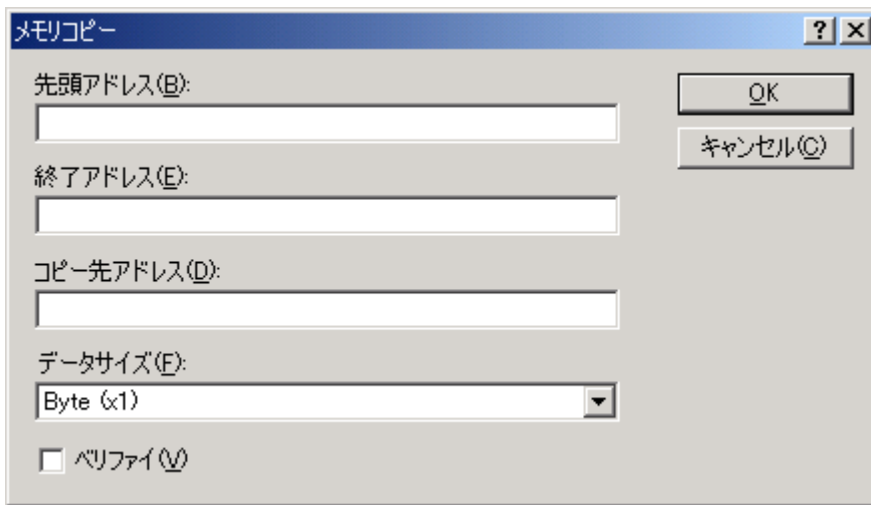


図 5.20 [メモリコピー]ダイアログボックス

[Memory]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。

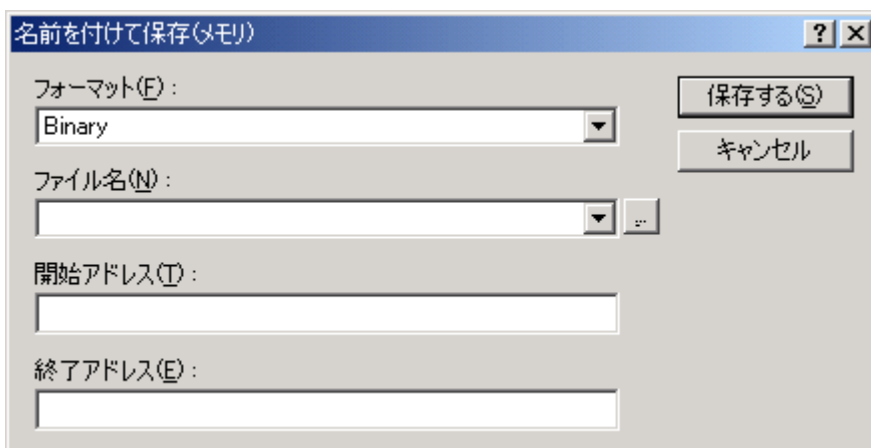


図 5.21 [名前を付けて保存(メモリ)]ダイアログボックス

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[参照...]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。ファイルの保存が完了すると、確認のメッセージボックスを表示します。

メモリバリエーション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->メモリのバリエーション...]を選択して[メモリバリエーション]ダイアログボックスを開きます。

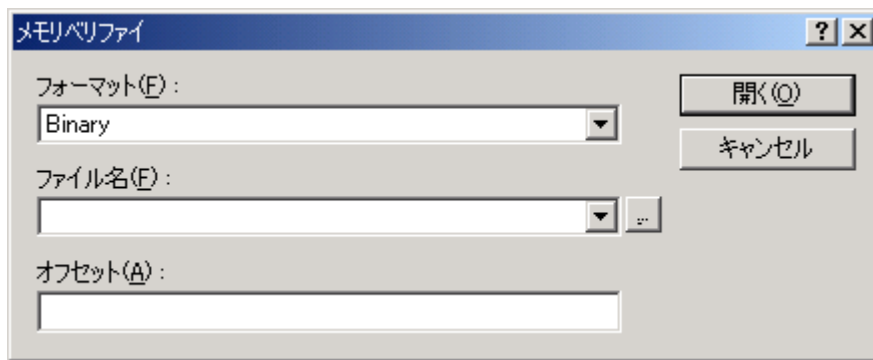


図 5.22 [メモリバリエーション]ダイアログボックス

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[Memory]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。

5.5.12 ウィンドウ内容を更新する

[Memory]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、[Memory]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。

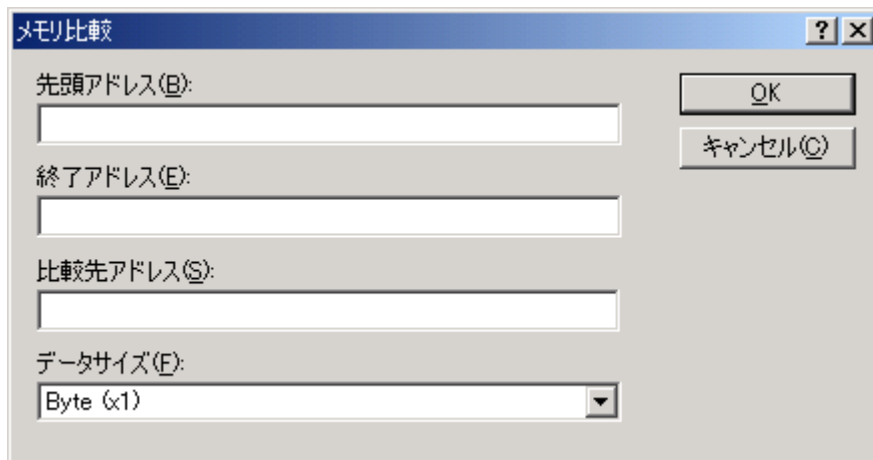


図 5.23 [メモリ比較]ダイアログボックス

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。[Memory]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。[Memory]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。

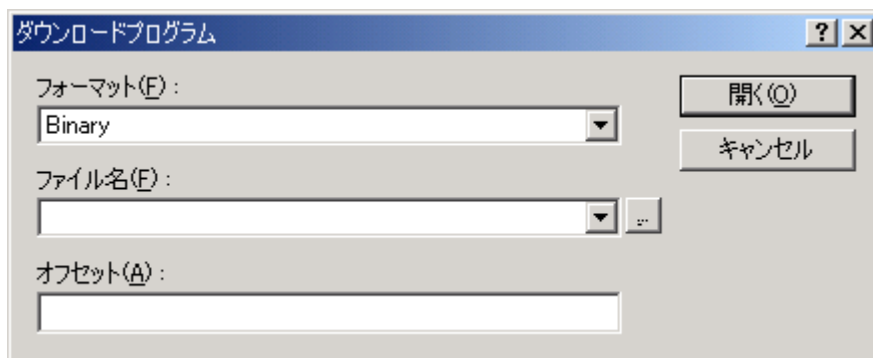


図 5.24 [ダウンロードプログラム]ダイアログボックス

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

マイクロコンピュータ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、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。マイクロコンピュータ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。

[Memory]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きます。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 [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、[I/O 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。

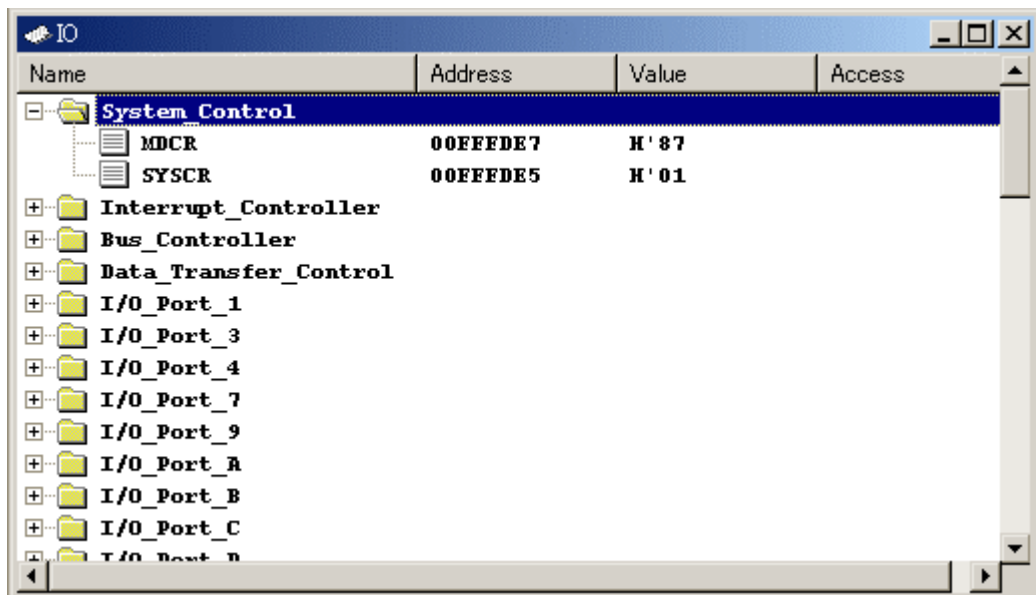


図 5.25 [IO]ウィンドウ

5.6.2 I/O Register 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、モジュール名をダブルクリックするか、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、“Enter”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。モジュール名を再びダブルクリックする（または“Enter”キーを押す）と、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、[Register]ウィンドウと同様のやり方で I/O レジスタを拡張します。

5.6.3 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。

【注】 [I/O]ウィンドウは、“H8Sxxxx.io”に定義されている内容を表示しています。“H8Sxxxx.io”の内容を編集することにより、表示するレジスタを追加／削除することができます。

“H8Sxxxx.io”に記載すべき内容については、「付録 D I/O ファイルフォーマット」を参照してください。

また、“H8Sxxxx.io”は以下ディレクトリ内にあります。

<HEW2 インストール先ディレクトリ>

¥Tools¥Renesas¥DebugComp¥Platform¥E10A¥xxxx¥IOFiles

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[Status]ウィンドウを表示します。

[Status]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。

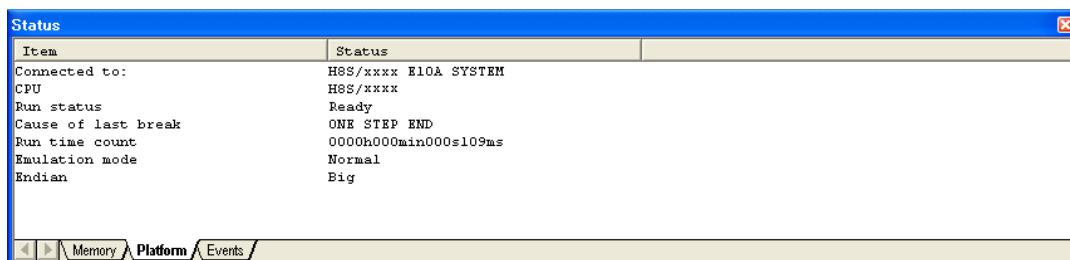


図 5.26 [Status]ウィンドウ

[Status]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、エミュレータのステータス情報、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、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[Memory]ウィンドウを使用します。

[Memory]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 5.16 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、[Localized Dump]ウィンドウを使用します。

(1) [Localized Dump]ウィンドウを開く

[Localized Dump]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。

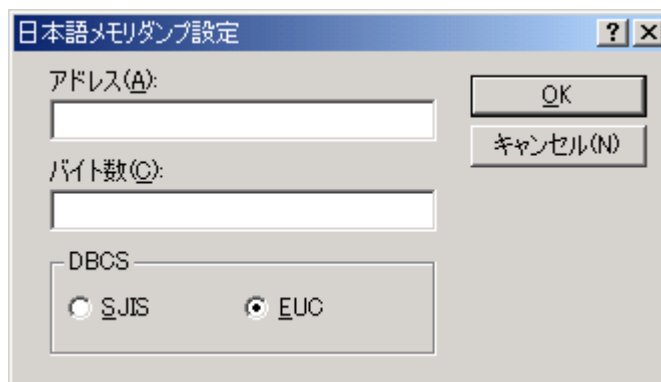


図 5.27 [日本語メモリダンプ設定]ダイアログボックス

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[Localized Dump]ウィンドウが開きます。



図 5.28 [Localized Dump]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を開きます。メモリ内容の日本語データを編集できます。

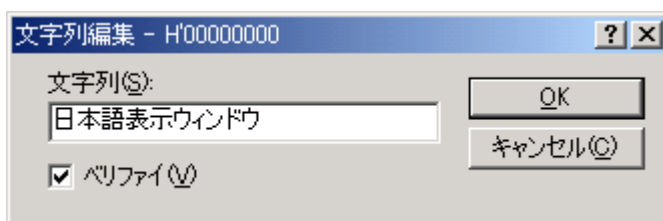


図 5.29 [文字列編集]ダイアログボックス

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。

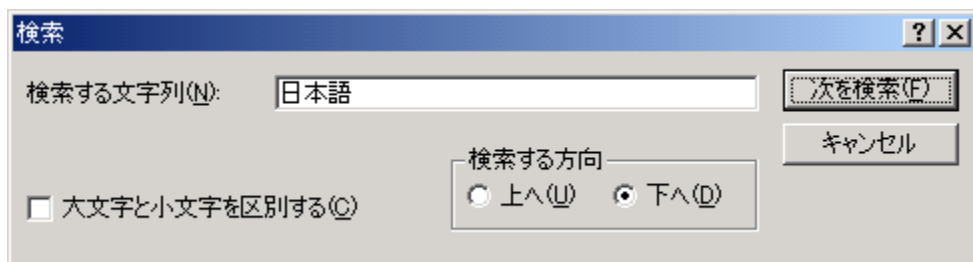


図 5.30 [検索]ダイアログボックス

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) 表示アドレスを変更する

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。

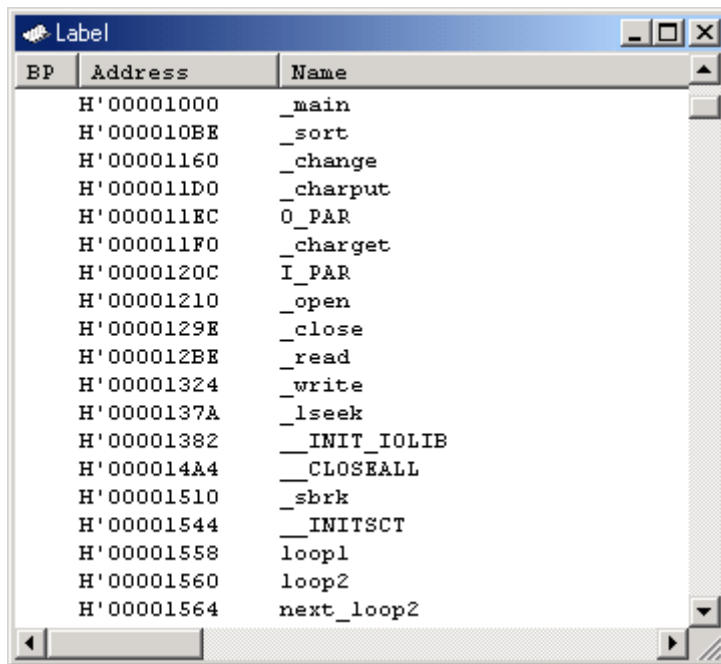
5.9 ラベルを見る

HEW がユーザプログラムソースコードをメモリ上の実際のコードにリンクする際に使用するデバッグ情報のほかに、デバッグオブジェクトファイルにはシンボル情報を含みます。これはテキスト名で、プログラムのアドレスをあらわします。HEW ではラベルと呼ばれています。[Disassembly]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。[エディットコントロール]にアドレスまたは値を入力できる場合には、代わりにラベルを使用することができます。

5.9.1 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。



BP	Address	Name
	H'00001000	_main
	H'000010BE	_sort
	H'00001160	_change
	H'000011D0	_charput
	H'000011EC	O_PAR
	H'000011F0	_charget
	H'0000120C	I_PAR
	H'00001210	_open
	H'0000129E	_close
	H'000012BE	_read
	H'00001324	_write
	H'0000137A	_lseek
	H'00001382	__INIT_IOLIB
	H'000014A4	__CLOSEALL
	H'00001510	_sbrk
	H'00001544	__INIT_SCT
	H'00001558	loop1
	H'00001560	loop2
	H'00001564	next_loop2

図 5.31 [Label]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 PC 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。

5.9.2 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 5.32 [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「5.12.3 複数ラベルをサポートする」を参照してください。

5.9.3 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 5.33 [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「5.12.3 複数ラベルをサポートする」を参照してください。

5.9.4 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 5.34 に示す確認メッセージボックスを表示します。

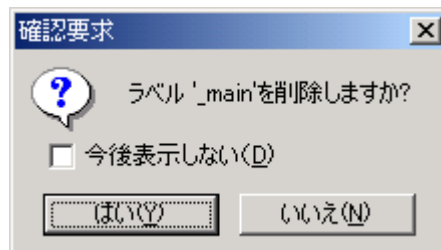


図 5.34 ラベル削除確認メッセージボックス

[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不要のときは、HEW の[オプション]ダイアログボックスの[確認]シートで[ラベル削除]オプションを選択しないでください。

5.9.5 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 5.35 に示す確認メッセージボックスを表示します。

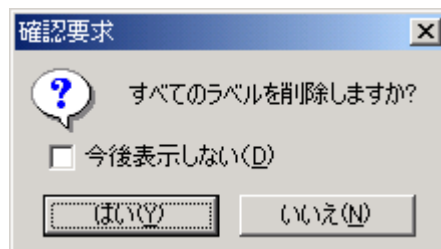


図 5.35 全ラベル削除確認メッセージボックス

[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、HEW の[オプション]ダイアログボックスの[確認]シートで[すべてのラベル削除]オプションを選択しないでください。

5.9.6 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。

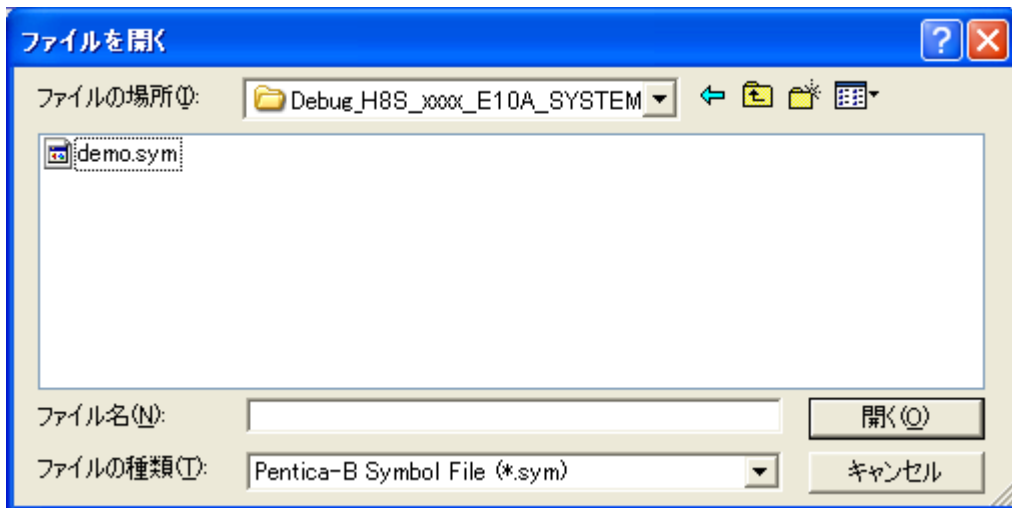


図 5.36 [ファイルを開く]ダイアログボックス

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。シンボルのロードが完了するとロードしたシンボル数を表示したメッセージボックスを表示します。このメッセージボックスは、HEW の[オプション]ダイアログボックスの[確認]シートで非表示にもできます。

5.9.7 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。

フォーマットの詳細は「付録 E シンボルファイルフォーマット」を参照してください。

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。

5.9.8 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。

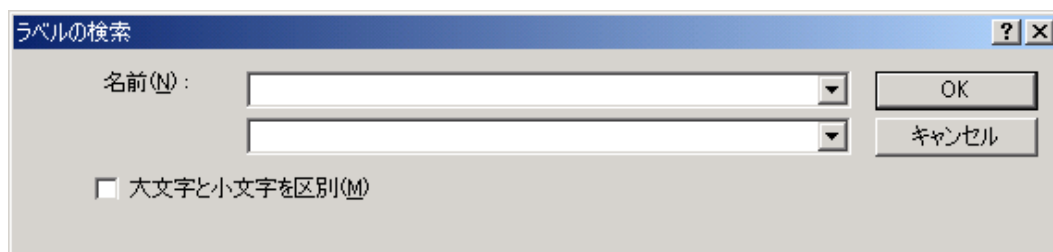


図 5.37 [ラベルの検索]ダイアログボックス

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。

5.9.9 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。

5.9.10 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[Source]または、[Disassembly]ウィンドウをオープンします。

5.10 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。ここでご説明する実行方法は、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することです。

5.10.1 リセットから実行を開始する

ユーザシステム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、“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。

【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。

5.10.2 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[Disassembly]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。

- [Register]ウィンドウ上で変更する。詳しくは、「5.4.3 レジスタの内容を修正する」を参照してください。
- [Source]ウィンドウまたは[Disassembly]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。

5.10.3 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. [Source]または[Disassembly]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、既に 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。

5.10.4 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。

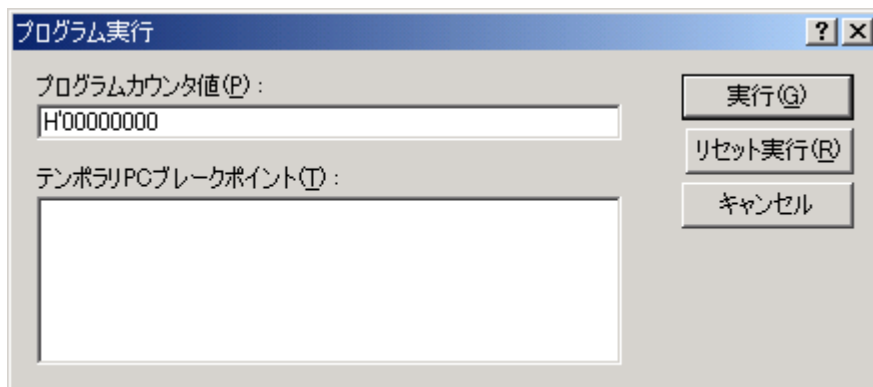


図 5.38 [プログラム実行]ダイアログボックス

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。

5.10.5 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、[Source]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。

[Disassembly]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、デバッグに命令を実行させ、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グ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ー実行する

関数をステップオーバー実行することを選択した場合には、デバッグ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ーするには、[ステップオーバー]ツールバーボタンをクリックするか、[デバッグ ->ステップオーバー]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。

5.10.6 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。

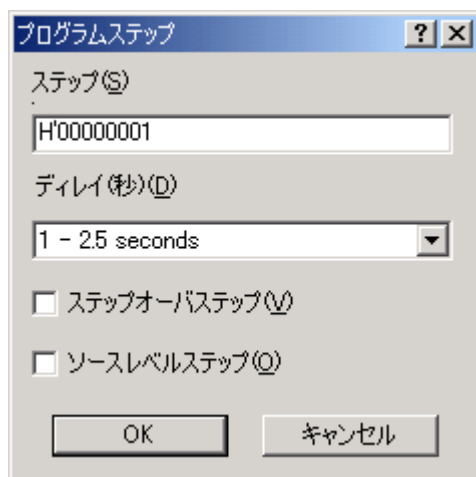


図 5.39 [プログラムステップ]ダイアログボックス

[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 から 6 で 0 が最も遅延が長くなります。
[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。

5.11 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。

5.11.1 Halt 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)。“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“Stop”というメッセージを表示します。

5.11.2 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、[Event]ウィンドウを使用します。[Event]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、「5.14 イベントポイントを使用する」を参照してください。

(1) [Source]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[Disassembly]または[Source]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"BREAK POINT"というメッセージを表示し、実行を停止。[Source]または[Disassembly]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。

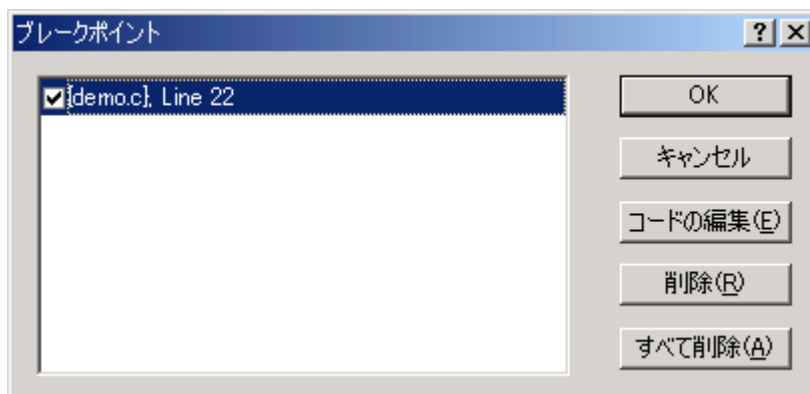


図 5.40 [ブレークポイント]ダイアログボックス

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。

【注】 [Breakpoint]ダイアログボックス設定時の注意事項

1. 指定アドレスが奇数の時は、偶数に切り捨てます。
2. BREAKPOINTは命令を置き換えることにより実現するので、フラッシュメモリ/RAM領域にだけ設定できます。ただし、次に示すアドレスには指定できません。
フラッシュメモリ/RAM 以外の領域
E10A エミュレータ用プログラム占有領域
Break Condition 1 が成立する命令
3. ステップを実行している間は、BREAKPOINTは無効です。
4. BREAKPOINTが設定されている命令を実行する際、実行を開始した直後のみBreak Condition 1は無効です。したがって、実行を開始した直後にBreak Condition 1の条件が成立してもブレークしません。
5. BREAKPOINTで停止後、再度そのアドレスから実行を再開した場合、1度そのアドレスをシングルステップにより実行してから実行を継続するので、リアルタイム性はなくなります。
6. STEP OVER機能を使用する時は、BREAKPOINTの設定とBreak Condition 1の設定は無効となります。

5.12 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。Elf/Dwarf2 オブジェクトファイルフォーマットを使用すると、実行中のアプリケーションに関するシンボルデバッグ情報を強力的にアクセス、および修正できます。

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
- ウォッチ-ローカル、およびユーザ定義
- スタックトレース

5.12.1 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)

```

5.12.2 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++)

5.12.3 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 5.41 [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ま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]	[関数名の指定]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[関数名の指定]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

[関数名の指定]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、[関数名の指定]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、[関数名の指定]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。

5.12.4 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。

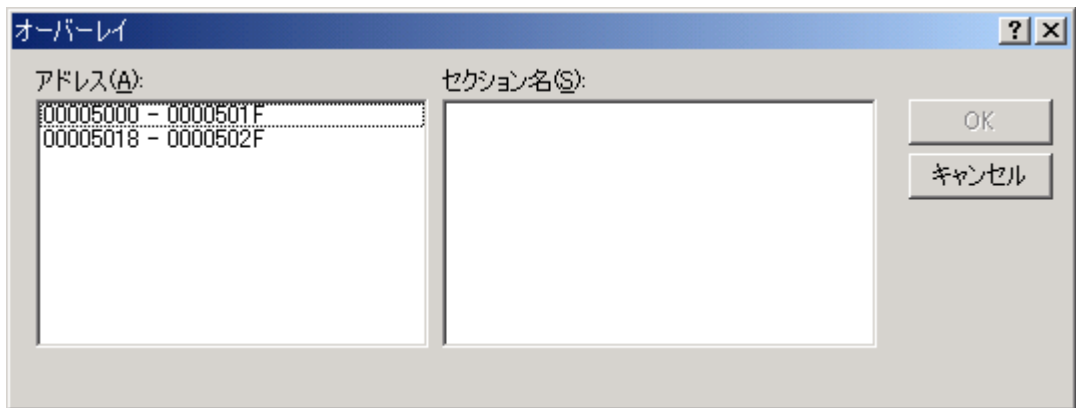


図 5.42 [オーバーレイ]ダイアログボックス (開いたとき)

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、オーバー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。

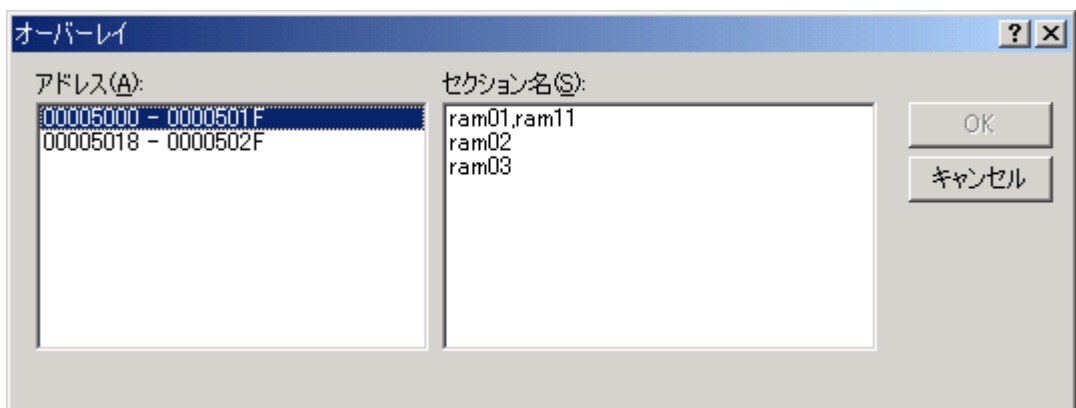


図 5.43 [オーバーレイ]ダイアログボックス (アドレス範囲を選択)

[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

オーバー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[オーバーレイ]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。

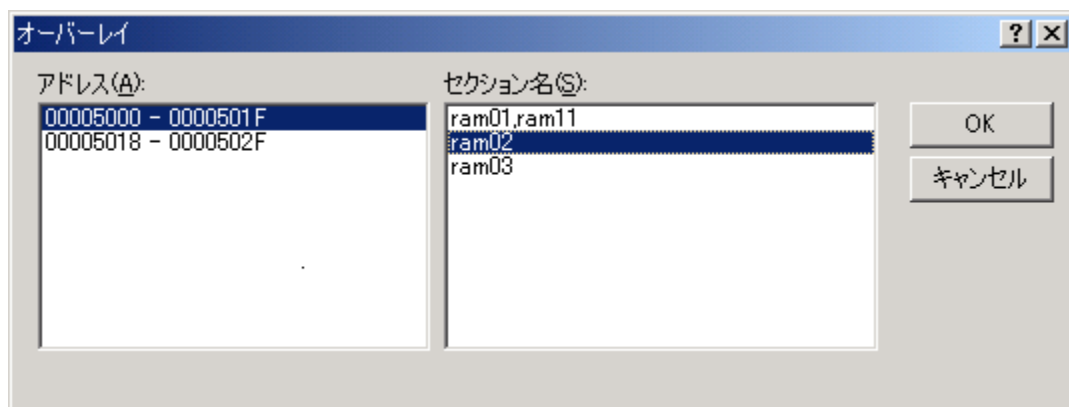


図 5.44 [オーバーレイ]ダイアログボックス (最も優先度の高いセクショングループ選択時)

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[オーバーレイ]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[オーバーレイ]ダイアログボックスで選択しなければなりません。

5.13 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。

5.13.1 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[Source]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。

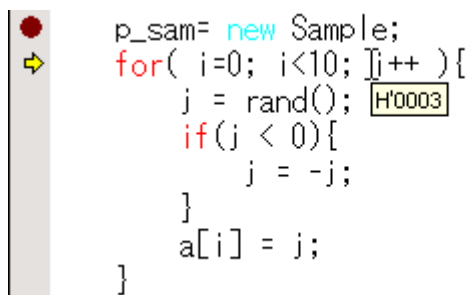


図 5.45 ツールチップウォッチ

5.13.2 インスタントウォッチ

確認したい変数を表示している[Source]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。

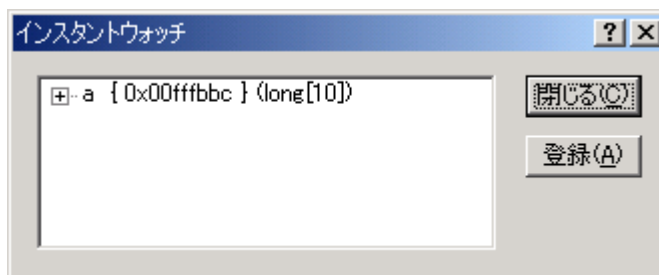


図 5.46 [インスタントウォッチ]ダイアログボックス

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[Watch]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[Watch]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。

5.13.3 [Watch]ウィンドウ

[Watch]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) [Watch]ウィンドウを開く

[Watch]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。[Watch]ウィンドウが開きます。ウィンドウの中は最初空白です。

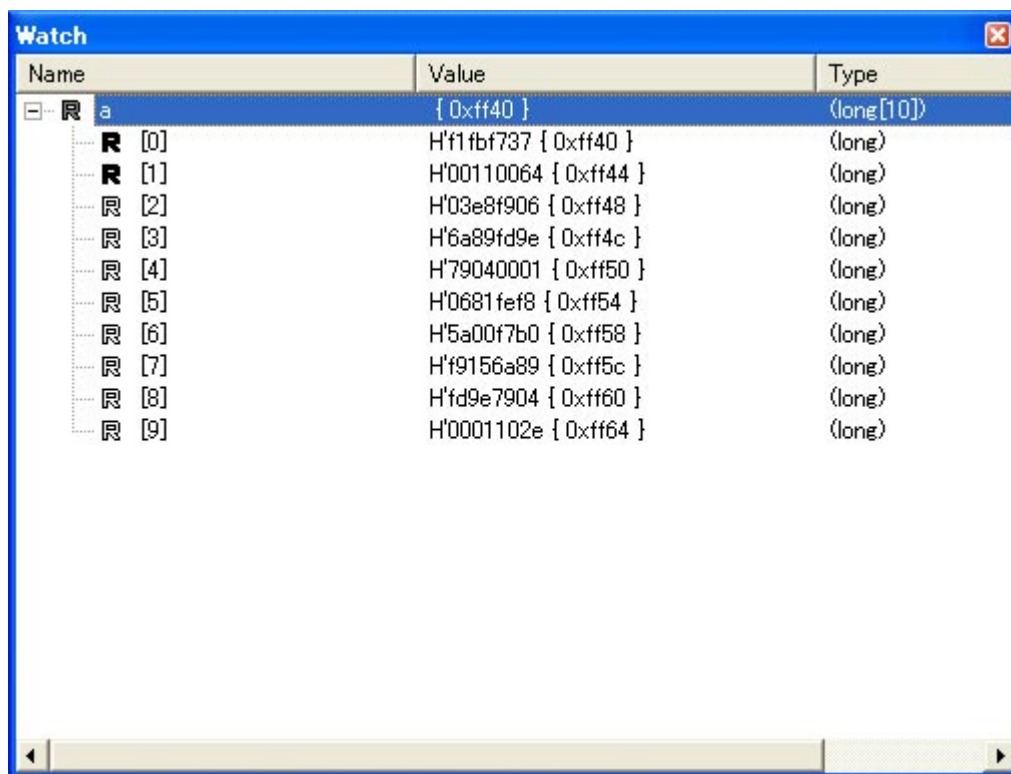


図 5.47 [Watch]ウィンドウ

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が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、割付け位置を表示します 割付け位置は{}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。

E10A エミュレータでは[Watch]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ユーザプログラムを一旦ブレイクし、データをリードします。

【注】 ユーザプログラムを一時的に停止するためユーザプログラムのリアルタイム性は損なわれます。

R マークの色が黒色の場合、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. [Watch]ウィンドウを閉じたりスクロールすると、設定情報が解除されます。
 4. 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) Watch アイテムを追加する

Watch アイテムを[Watch]ウィンドウに追加するには、[Watch]ウィンドウの[シンボル登録]ダイアログボックスを使用します。

- ③ [Watch]ウィンドウからシンボル登録を使用するには
[Watch]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 5.48 [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[Watch]ウィンドウに追加します。

また、[Source]ウィンドウから[Watch]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[Watch]ウィンドウに追加しますが、その値は空白または、クエスチョンマーク'?'を表示します。

(3) Watch アイテムを拡張する

Watch 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、Watch アイテムを拡張できるという意味です。Watch アイテムを拡張するには、プラス記号(+)をクリックします。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。Watch 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。

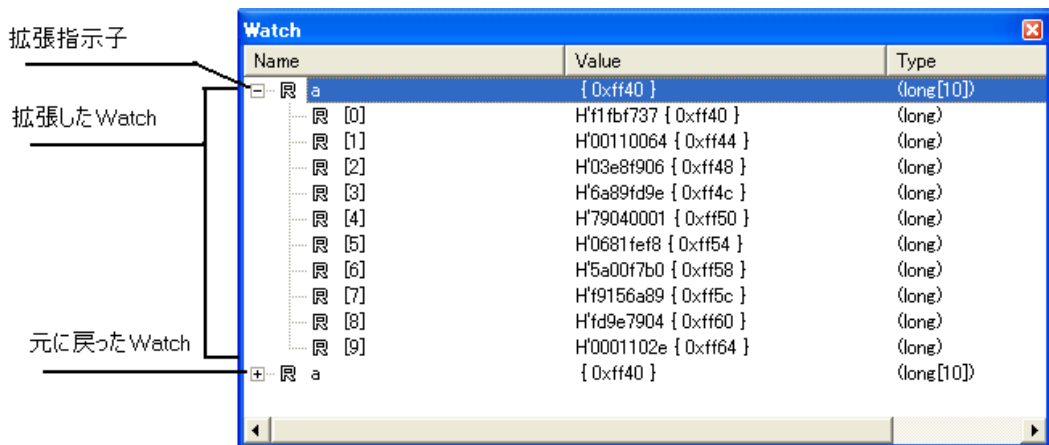


図 5.49 Watch を拡張する

拡張した Watch 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。

(4) Watch 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、Watch 変数の値を変更することができます。Watch 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。

Watch 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。

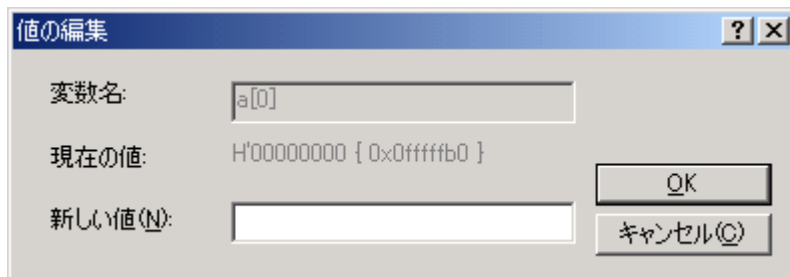


図 5.50 [値の編集]ダイアログボックス

[新しい値]フィールドに新しい値または式を入力して[OK]ボタンをクリックします。[Watch]ウィンドウを更新し、新しい値を表示します。

(5) Watch アイテムを削除する

特定の Watch 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、[Watch]ウィンドウを更新します。

すべての Watch 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、[Watch]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

Watch 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[Watch]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

[Watch]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、[Watch]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) [Memory]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[Memory]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、[表示アドレス]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[Memory]ウィンドウを開きます。

5.13.4 [Locals]ウィンドウ

[Locals]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) [Locals]ウィンドウを開く

[Locals]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。

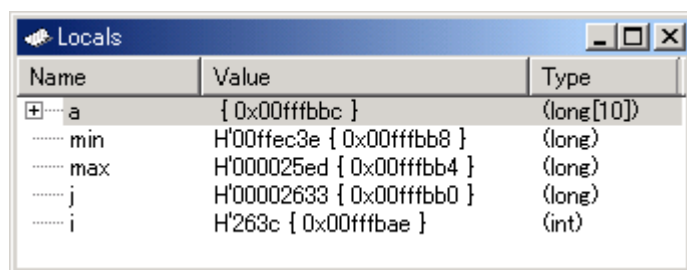


図 5.51 [Locals]ウィンドウ

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、[Watch]ウィンドウと同じ方法で修正することができます。

5.14 イベントポイントを使用する

E10A 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。

5.14.1 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 255 ポイントまで設定できます。

5.14.2 Break condition とは

Break condition 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。

5.14.3 [Event]ウィンドウを開く

[Event]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、[Eventpoints]ツールバーボタンをクリックします。

[Event]ウィンドウには、2 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。

[Breakcondition]シート

ブレークコンディションチャンネルの設定内容を表示、設定します。

5.14.4 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。

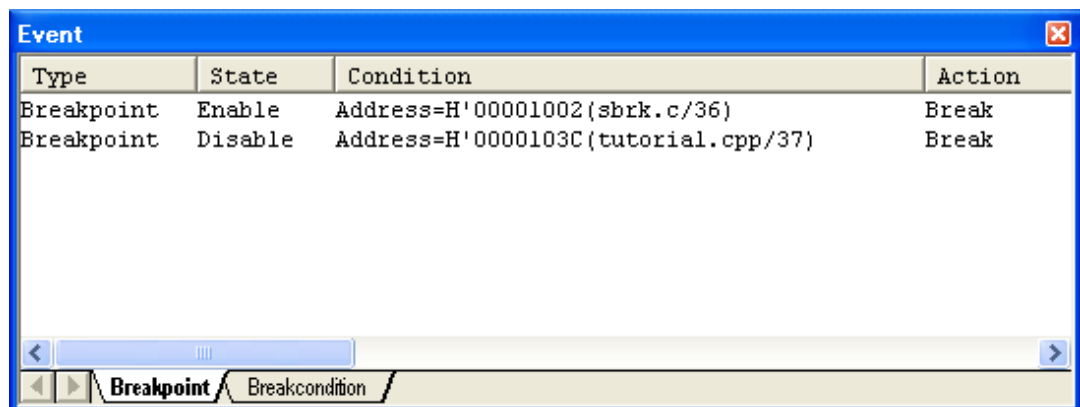


図 5.52 [Event]ウィンドウ ([Breakpoint]シート)

ブレークポイントを表示、設定します。
シート内に表示する項目は以下の通りです。

[Type]	ブレークポイントであることを表示します。
[State]	該当ブレークポイントの有効/無効を示します。 Enable : 有効 Disable : 無効
[Condition]	ブレークポイント設定アドレスを表示します。 Address=プログラムカウンタ (対応するファイル名/行、シンボル名)
[Action]	ブレーク条件成立時の動作を表示します。 Break : 実行停止

本ウィンドウでブレークポイントをダブルクリックすると、[Set Break]ダイアログボックスが開き、ブレーク条件を変更することができます。

ウィンドウ内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を表示します。このメニューは以下のオプションを含みます。

5.14.5 追加

ブレークポイントを設定します。クリックすると、[Set Break]ダイアログボックスが開き、ブレーク条件を設定することができます。

5.14.6 編集

ブレークポイントを1つ選択している場合のみ有効です。変更したいブレークポイントを選択後クリックすると、[Set Break]ダイアログボックスが開き、ブレーク条件を変更することができます。

5.14.7 有効

選択しているブレークポイントを有効にします。

5.14.8 無効

選択しているブレークポイントを無効にします。無効にした場合は、ブレーク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ブレークは成立しません。

5.14.9 削除

選択しているブレークポイントを削除します。ブレークポイントを削除しないで、詳細情報は保持したまま、条件が一致してもブレークを成立させないようにするには、Disable オプションを使用します(「5.14.8 無効」参照)。

5.14.10 すべてを削除

全てのブレークポイントを削除します。

5.14.11 ソースを表示

ブレークポイントを1つ選択している場合のみ有効です。ブレークポイントのあるソースファイルをオープンします。

5.14.12 [Set Break]ダイアログボックス



図 5.53 [Set Break]ダイアログボックス

本ダイアログボックスでは、ブレーク条件を設定します。

設定するブレークポイントアドレスを [Address]エディットボックスで指定します。ブレークポイントは 255 個まで設定できます。

[Address]の設定時に、アドレスに多重定義関数あるいはメンバ関数を含むクラス名を入力した場合、[Select Function]ダイアログボックスが開くので設定する関数を選択します。

指定したブレーク条件は、[OK]ボタンをクリックすることにより設定します。[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。

【注】 [Breakpoint]ダイアログボックス設定時の注意事項

1. 指定アドレスが奇数の時は、偶数に切り捨てます。
2. BREAKPOINTは命令を置き換えることにより実現するので、フラッシュメモリ/RAM領域にだけ設定できます。ただし、次に示すアドレスには指定できません。
 - フラッシュメモリ/RAM 以外の領域
 - E10A エミュレータ用プログラム占有領域
 - Break Condition 1 が成立する命令
3. ステップを実行している間は、BREAKPOINTは無効です。
4. BREAKPOINTが設定されている命令を実行する際、実行を開始した直後のみBreak Condition 1は無効です。したがって、実行を開始した直後にBreak Condition 1の条件が成立してもブレークしません。
5. BREAKPOINTで停止後、再度そのアドレスから実行を再開した場合、1度そのアドレスをシングルステップにより実行してから実行を継続するので、リアルタイム性はなくなります。
6. STEP OVER機能を使用する時は、BREAKPOINTの設定とBreak Condition 1の設定は無効となります。

5.14.13 ブレークコンディションを設定する

[Breakcondition]シートではブレークコンディションの設定内容の表示、変更および追加ができます。

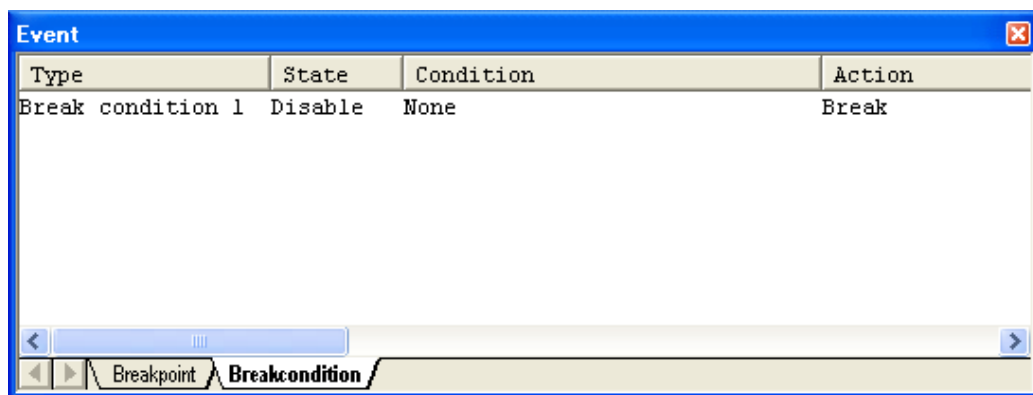


図 5.54 [Event]ウィンドウ ([Breakcondition]シート)

【留意事項】

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、オンラインヘルプを参照してください。

ブレーク条件を表示、設定します。

【注】 Break Condition 条件設定時の注意事項

- (a) Break Condition 1はGo to cursor、Step In、Step Over、Step Out使用時は無効です。
- (b) BREAKPOINTが設定されている命令を実行する際に、Break Condition 1の条件は無効となります。
- (c) STEP OVER機能を使用する時は、BREAKPOINTの設定とBreak Condition 1の設定は無効となります。

シート内に表示する項目は以下の通りです。

[Type]	ブレイクチャネル番号を表示します。
[State]	該当ブレイクポイントの有効/無効を示します。 Enable : 有効 Disable : 無効
[Condition]	Break が成立する条件を表示します。
[Action]	ブレイク条件成立時の動作を表示します。 Break : 実行停止

本ウィンドウでブレイクポイントをダブルクリックすると、[Break condition 1]ダイアログボックスが開き、ブレイク条件を変更することができます。

[Break condition 1]ダイアログボックスの詳細については、各製品のオンラインヘルプを参照してください。

ウィンドウ内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を表示します。このメニューは以下のオプションを含みます。

5.14.14 編集...

ブレイクポイントを1つ選択している場合のみ有効です。変更したいブレイクチャネルを選択後クリックすると、[Break condition 1]ダイアログボックスが開き、ブレイク条件を変更することができます。

5.14.15 有効

選択しているブレイクチャネルを有効にします。
条件が設定されていないブレイクチャネルは Enable にすることができません。

5.14.16 無効

選択しているブレイクチャネルを無効にします。無効にした場合は、指定した条件が一致してもブレイクは成立しません。

5.14.17 削除

選択しているブレイクチャネルの条件を初期化します。ブレイクチャネルを初期化しないで、詳細情報は保持したまま、条件が一致してもブレイクを成立させないようにするには、Disable オプションを使用します(「5.14.16 無効」参照)。

5.14.18 すべてを削除

全てのブレイクチャネルの条件を初期化します。

5.14.19 ソースを表示

ブレイクチャネルを1つ選択している場合のみ有効です。ブレイクチャネルのある[Source]ウィンドウをオープンします。

ブレイクチャネルにアドレス値が設定されていない場合は使用できません。

5.14.20 シーケンシャル設定

ブレイクチャネルのシーケンシャル条件を設定します。

5.14.21 ブレイクコンディションの編集

PC ブレイクポイント、ブレイクコンディション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。

以下ブレイクコンディションを例に設定以外の操作方法について説明します。

5.14.22 ブレイクコンディションの設定内容を変更する

変更したいブレイクコンディション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...]メニューはブレイクコンディションを1個選択している時のみ有効となります。

5.14.23 ブレイクコンディションを有効にする

ブレイクコンディションを選択後ポップアップメニューから[有効]を選択すると、選択しているブレイクコンディションを有効にします。

5.14.24 ブレイクコンディションを無効にする

ブレイクコンディションを選択後ポップアップメニューから[無効]を選択すると、選択しているブレイクコンディションを無効にします。無効にした場合は、ブレイクコンディション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。

5.14.25 ブレイクコンディションを削除する

ブレイクコンディションを選択後ポップアップメニューから[削除]を選択すると、選択しているブレイクコンディションを削除します。ブレイクコンディションを削除しないで、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(「5.14.24 ブレイクコンディションを無効にする」参照)。

5.14.26 ブレイクコンディションをすべて削除する

ポップアップメニューから[すべて削除]を選択すると、すべてのブレイクコンディションを削除します。

5.14.27 ブレイクコンディションのソース行を表示する

ブレイクコンディション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、ブレイクポイントのある[Source]または[Disassembly]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ブレイクコンディションを1個選択している時のみ有効となります。

5.14.28 [Break condition 1]ダイアログボックス

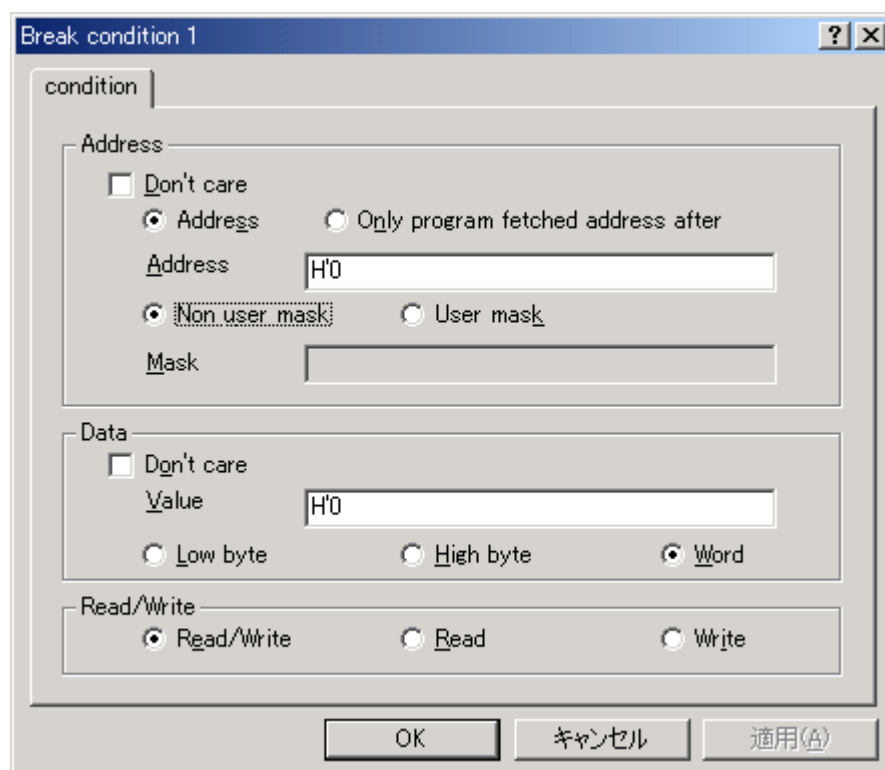


図 5.55 [Break condition 1]ダイアログボックス

アドレスバス、データバスおよび、リード・ライトサイクルに対する条件を設定します。

【留意事項】

本ダイアログボックスは、製品ごとに異なります。各製品の表示内容については、オンラインヘルプを参照してください。

5 デバッグ

アドレスバスに対する条件を[Address]で設定します。

オプション	説明
[Don't care]チェックボックス	アドレス条件を設定しないことを表します。
[Address]ラジオボタン	通常アドレスバスをブレイク条件とします。
[Only program fetched address after] ラジオボタン	プリフェッチアドレス実行後ブレイクをブレイク条件とします。
[Address]エディットボックス	アドレスの値を数値またはシンボルで設定します。
[Mask]エディットボックス	User mask を選択した場合に、マスク値を設定します。マスクを行ったビットは、どんな値でも条件が成立することになります。

ラジオボタンの選択により、選択できるオプションの内容が変わります。

オプション	説明
[Address] ラジオボタン	全ての条件が設定可能です。
[Only program fetched address after] ラジオボタン	[Address]条件のみ設定可能です。

データバスに対する条件を[Data]で設定します。

オプション	説明
[Don't care]チェックボックス	データ条件を設定しないことを表します。
[Value]エディットボックス	データバスの値を数値で設定します。
[Low byte] ラジオボタン	データアクセスサイズを下位 8 ビットアクセスとします。
[High byte] ラジオボタン	データアクセスサイズを上位 8 ビットアクセスとします。
[Word] ラジオボタン	データアクセスサイズをワードアクセスとします。

リード、ライトサイクルの条件を[Read/Write]で設定します。

オプション	説明
[Read/Write]ラジオボタン	リード、ライトサイクル条件をブレイク条件とします。
[Read]ラジオボタン	リードサイクルの場合のみブレイク条件とします。
[Write]ラジオボタン	ライトサイクルの場合のみブレイク条件とします。

5.15 トレース情報を見る

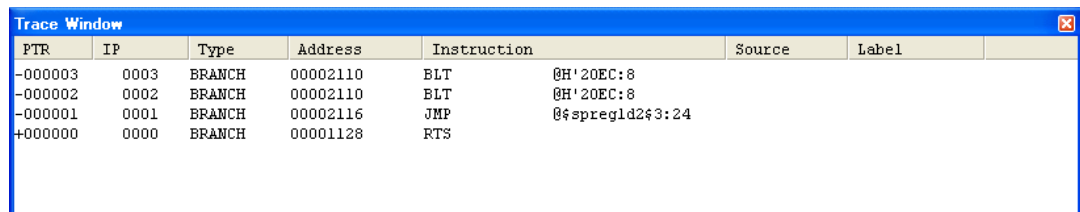
Trace 機能の説明は、「2.2 トレース機能」を参照してください。

5.15.1 [Trace Window]ウィンドウを開く

[Trace Window]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。

5.15.2 トレース情報を取得する

取得したトレース情報は[Trace Window]ウィンドウに表示します。



PTR	IP	Type	Address	Instruction	Source	Label
-000003	0003	BRANCH	00002110	BLT	@H'20EC:8	
-000002	0002	BRANCH	00002110	BLT	@H'20EC:8	
-000001	0001	BRANCH	00002116	JMP	@\$spregld2\$3:24	
+000000	0000	BRANCH	00001128	RTS		

図 5.56 [Trace Window]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ポインタ（最後に実行した命令が+0となります）
[IP]	取得したトレース情報数
[Type]	分岐種別 BRANCH: 分岐元
[Address]	命令アドレス
[Instruction]	命令ニーモニック
[Source]	C/C++またはアセンブラソース
[Label]	ラベル情報

[Trace Window]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。

【留意事項】

トレース取得できる分岐命令の数、トレース表示内容は、製品によって異なります。各製品の仕様については、オンラインヘルプを参照してください。

5.15.3 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。その際にトレース情報を保持しているトレースバッファは空になります。複数の[Trace Window]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 Window]ウィンドウをクリアすることになります。

5.15.4 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 Window]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。このファイルは保存のみ可能で、[Trace Window]ウィンドウへの読み込みはできません。

5.15.5 [Source]ウィンドウを表示する

トレースレコードに対応する[Source]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、[Source]ウィンドウあるいは[Disassembly]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。

5.15.6 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。

5.16 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、[Stack Trace]ウィンドウを使用します。

5.16.1 [Stack Trace]ウィンドウを開く

[Stack Trace]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をクリックします。

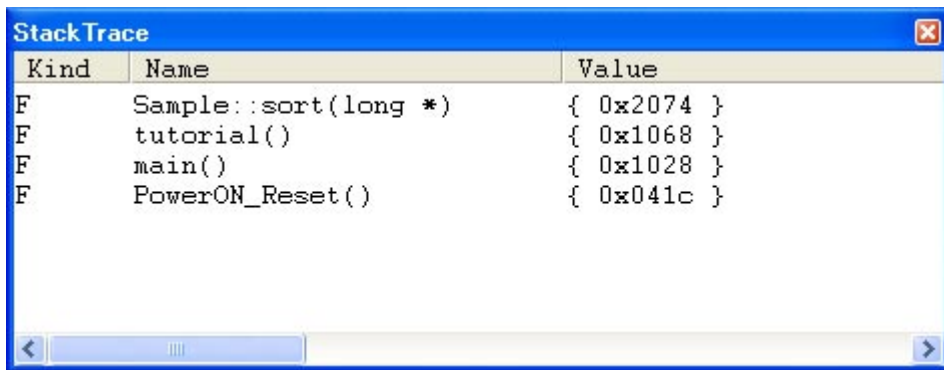


図 5.57 [Stack Trace]ウィンドウ

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。

5.16.2 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[Source]ウィンドウ上に表示します。

5.16.3 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。[Stack Trace]ウィンドウの表示形式を設定します。

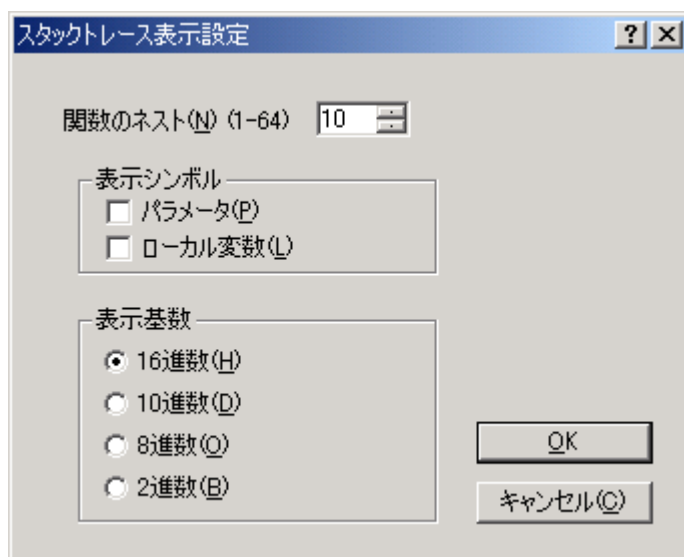


図 5.58 [スタックトレース表示設定]ダイアログボックス

- | | |
|----------|---|
| [関数のネスト] | [Stack Trace]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] | [Stack Trace]ウィンドウの表示基数を指定します。 |

5.17 メモリ内容を画像形式で表示する

[Image]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。

5.17.1 [Image]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。

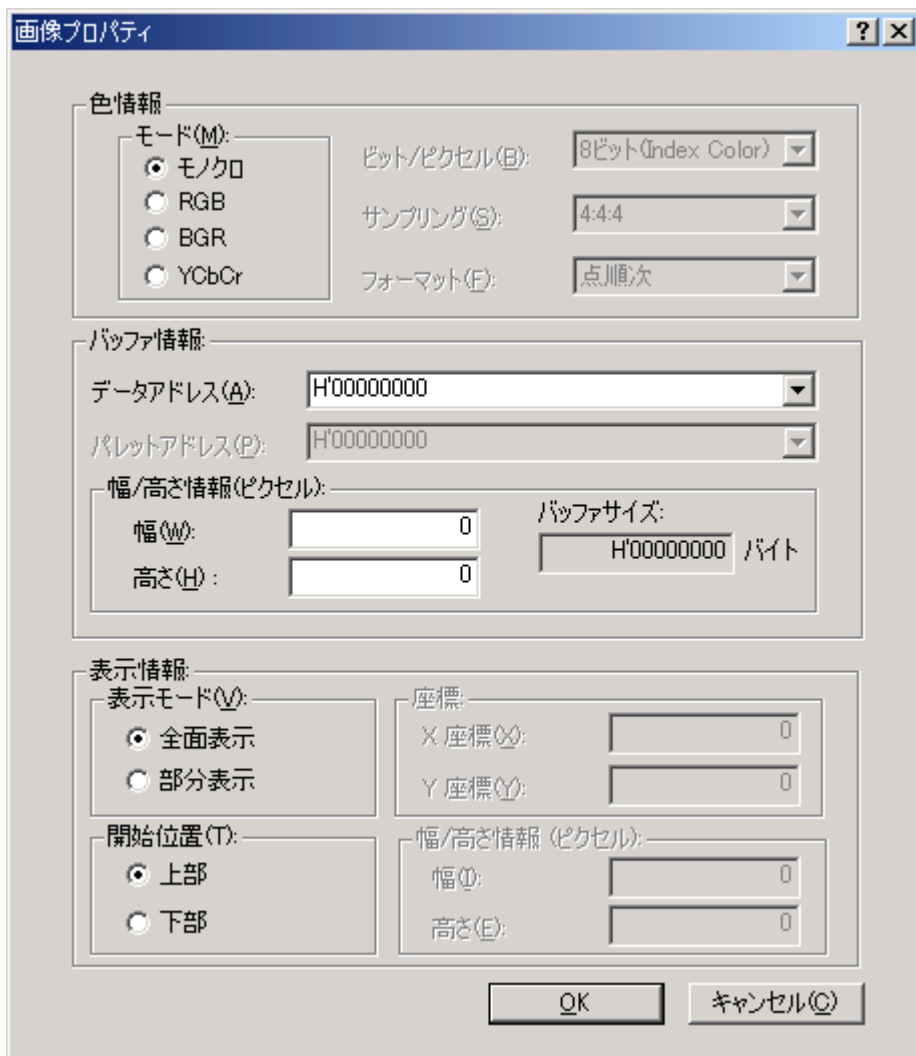


図 5.59 [画像プロパティ]ダイアログボックス

[画像プロパティ]ダイアログボックスでは[Image]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[Image]ウィンドウが開きます。

[Image]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 5.60 [Image]ウィンドウ

メモリの内容を画像で表示します。

5.17.2 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。

5.17.3 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。

5.17.4 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。

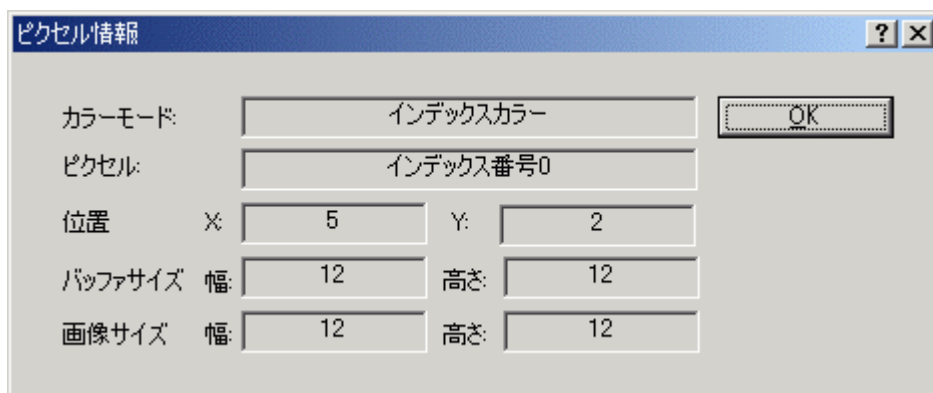


図 5.61 [ピクセル情報]ダイアログボックス

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。

5.18 メモリ内容を波形形式で表示する

[Waveform View]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。

5.18.1 [Waveform View]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。

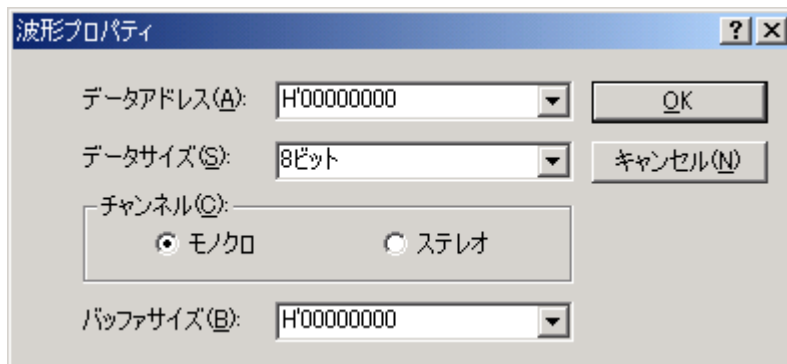


図 5.62 [波形プロパティ]ダイアログボックス

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] モノクロ/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[Waveform View]ウィンドウが開きます。

[Waveform View]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。

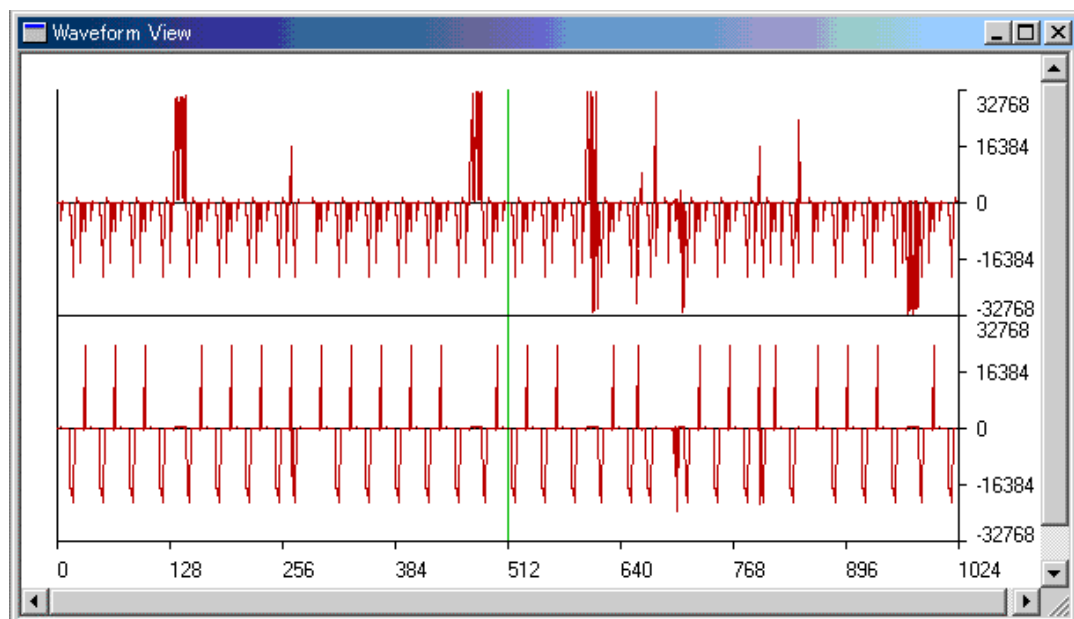


図 5.63 [Waveform View]ウィンドウ

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。

5.18.2 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。

5.18.3 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。

5.18.4 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。

5.18.5 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。

5.18.6 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。

5.18.7 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。

5.18.8 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。

5.18.9 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。

5.18.10 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。

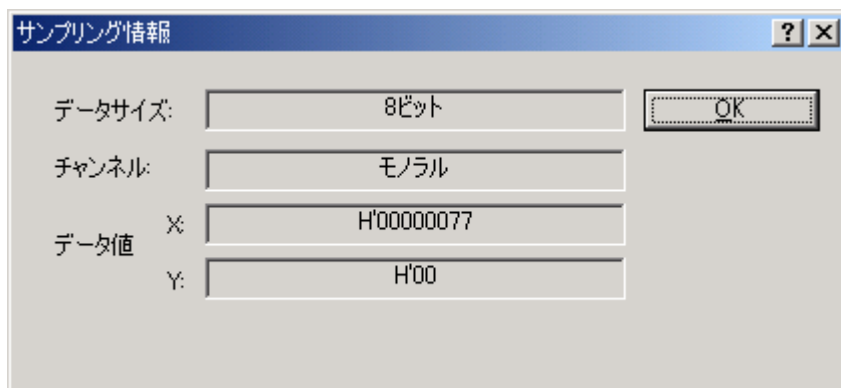


図 5.64 [サンプリング情報]ダイアログボックス

[Waveform View]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)

6. チュートリアル

6.1 はじめに

E10A 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 関数でソートするランダムデータを生成します。

sort 関数では main 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 関数で生成した配列を入力し、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。

【留意事項】

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異なることがあります。

6.2 HEW の起動

「3.4 システムチェック」にしたがって HEW を起動してください。

6.3 E10A エミュレータのセットアップ

E10A エミュレータの基本設定を行います。

6.4 [Configuration]ダイアログボックスの設定

E10A エミュレータの基本設定を行うために、[オプション]メニューから[エミュレータ]を選択し、さらに[システム]を選択してください。[Configuration]ダイアログボックスが表示されます。

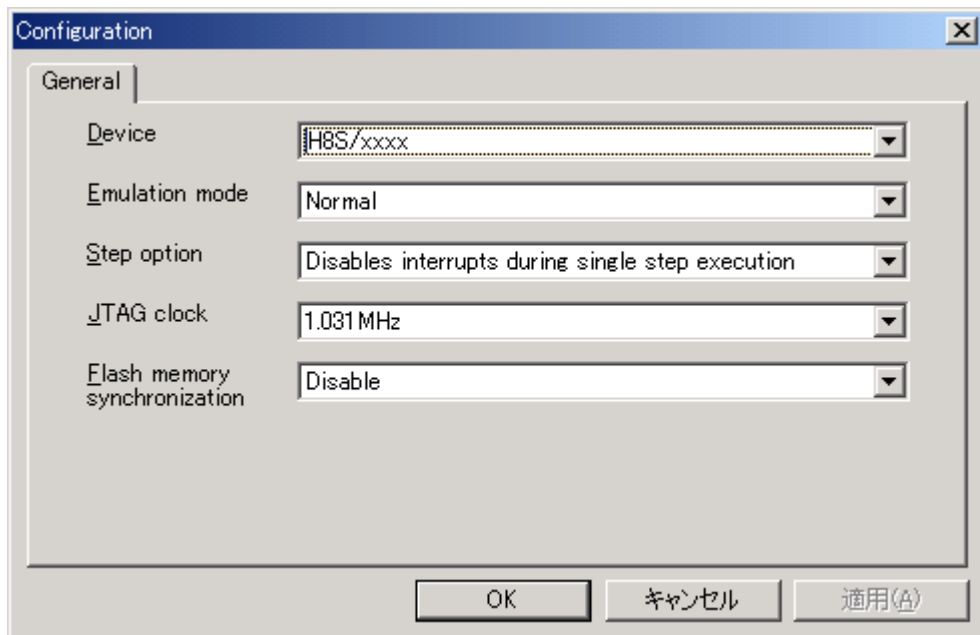


図 6.1 [Configuration]ダイアログボックス

【留意事項】

本ダイアログボックスは、製品ごとに異なります。各製品の表示内容については、オンラインヘルプを参照してください。

オプションを以下のように設定してください。

表 6.1 [General]ページのオプション

オプション	説明
[Device]コンボボックス	MCU デバイス名を表示します。
[Emulation mode]コンボボックス	ユーザプログラム実行時のエミュレーションモードを選択します。 Normal: 通常の実行を行います。 No break: ソフトウェアブレークポイント、およびハードウェアブレークポイントを一時的に無効にしてユーザプログラムを実行します。
[Step option]コンボボックス	ステップ中の割込みの開放／マスクを設定します。 Disables interrupts during single step execution: ステップ中の割込みをマスクします。 Enables interrupts during single step execution: ステップ中の割込みを開放します。
[JTAG clock]コンボボックス	JTAG の周波数*を設定します。
[Flash memory synchronization]コンボボックス	ホストコンピュータ、フラッシュメモリ間の同期方法を設定します。 ホストコンピュータからフラッシュメモリへの同期を行うと、ユーザプログラム停止時にフラッシュ書き込みのために待ち時間が生じますが、表示内容とフラッシュメモリが常に一致します。 フラッシュメモリからホストコンピュータへの同期を行うと、ユーザプログラム停止時にフラッシュメモリの内容を読み出されるのでユーザプログラムモードにある書き替え内容が反映されます。 Disable : E10A 起動時およびフラッシュメモリ領域の変更時以外は同期を行いません。 PC to Flash memory : ホストコンピュータからフラッシュメモリへの同期を行います。 Flash memory to PC : フラッシュメモリからホストコンピュータへの同期を行います。 PC to Flash memory、Flash memory to PC : ホストコンピュータ、フラッシュメモリ間の同期を行います。

【注】 JTAG が動作する周波数の範囲は、サポートするデバイスによって異なります。

[OK]ボタンをクリックして、コンフィグレーションを設定してください。

6.5 RAM の動作チェック

RAM が正常に動作することをチェックします。

[Memory]ウィンドウでメモリ内容を表示、編集し、メモリが正常に動作することを確認します。

【留意事項】

マイコンによってはボード上にメモリをつけることができます。この場合、メモリ動作チェックは上記だけでは不完全な場合があります。メモリチェック用プログラムを作成し、チェックすることをお勧めします。

[表示]メニューの[CPU]サブメニューから[メモリ]を選択し、[先頭アドレス]エディットボックスにRAM のアドレスを入力してください。ここでは”H’FFC000”を入力し、[終了アドレス]エディットボックスに”H’FFEFBF”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。

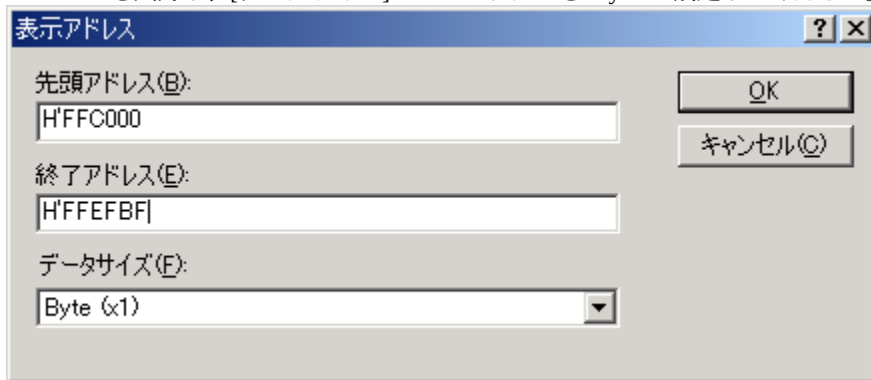


図 6.2 [表示アドレス]ダイアログボックス

【留意事項】

各製品ごとに RAM 領域の設定は異なります。各製品のハードウェアマニュアルを参照してください。

[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[Memory]ウィンドウが表示されます。

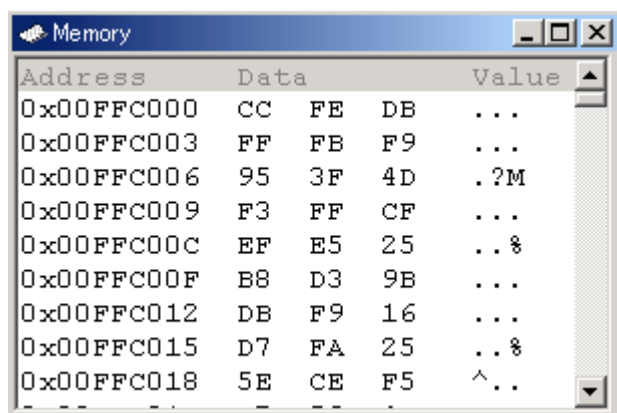


図 6.3 [Memory]ウィンドウ

[Memory]ウィンドウ上のデータ部分をダブルクリックすることにより、値が変更できます。またデータ部分をダブルクリックしなくても、カーソルのある場所のデータ内容を直接編集することができます。

6.6 チュートリアルプログラムのダウンロード

6.6.1 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。

E10A エミュレータでは、内蔵 Flash 領域に対してもダウンロード、PC ブレークポイントを設定することができます。

PC ブレークポイントの設定方法については、「6.16.1 PC ブレーク機能」を参照してください。

[Download modules]の[Tutorial.abs]から[ダウンロード]を選択します。

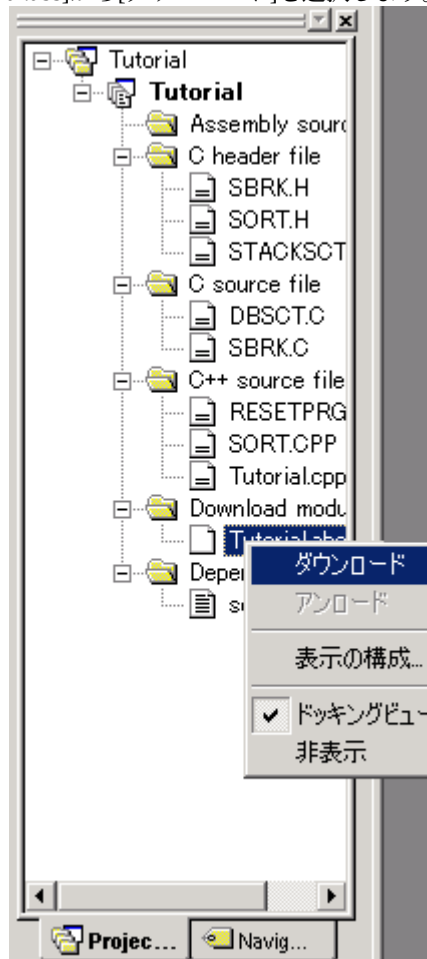


図 6.4 チュートリアルプログラムのダウンロード

6.6.2 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。

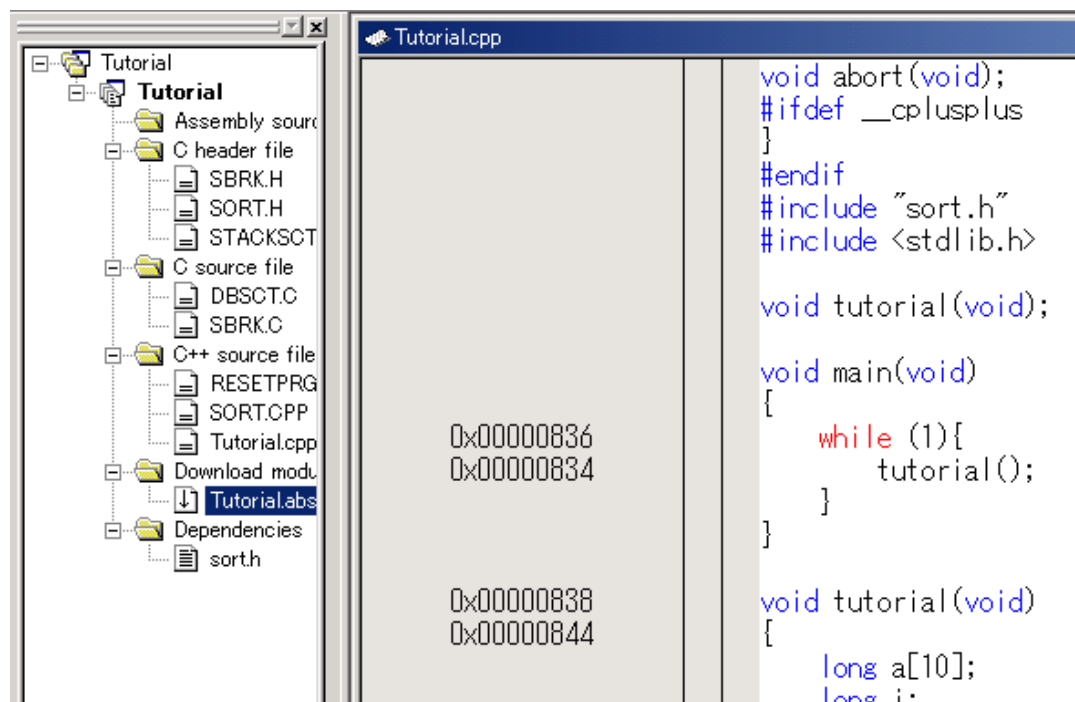


図 6.5 [Source]ウィンドウ (ソースプログラムの表示)

必要であれば、[ツール]メニューから[表示形式]オプションを選択し、見やすいフォントとサイズを選択してください。

[Source]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。

6.7 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。

[Source]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て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。

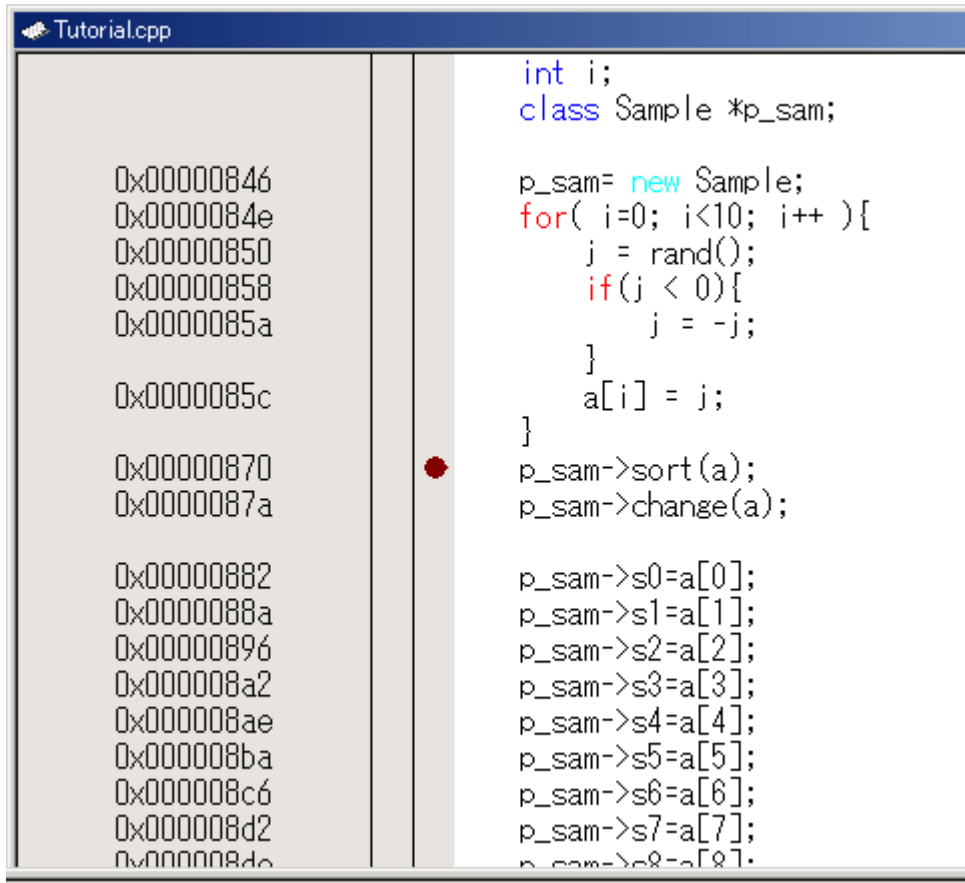


図 6.6 [Source]ウィンドウ (PC ブレークポイントの設定)

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ています。

6.8 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およびスタックポインタの値を設定してください。

[表示]メニューの[CPU]サブメニューから[レジスタ]を選択してください。[Register]ウィンドウが表示されます。

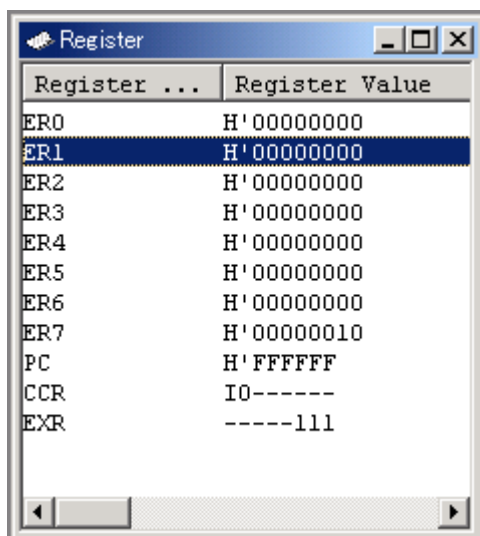


図 6.7 [Register]ウィンドウ

プログラムカウンタ（PC）を変更する場合には、[Register]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'400 を設定し、[OK]ボタンをクリックしてください。

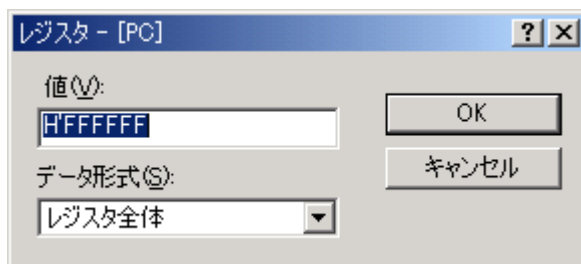


図 6.8 [レジスタ]ダイアログボックス (PC)

6.9 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”*** RUNNING”と表示します。

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[Editor]カラム中に矢印が表示されます。また、[BREAK POINT]メッセージがステータスバーに表示されます。

【留意事項】

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。

HEW2 インストール先ディレクトリ

¥Tools¥Renesas¥DebugComp¥Platform¥E10A¥xxxx¥Tutorial ¥source

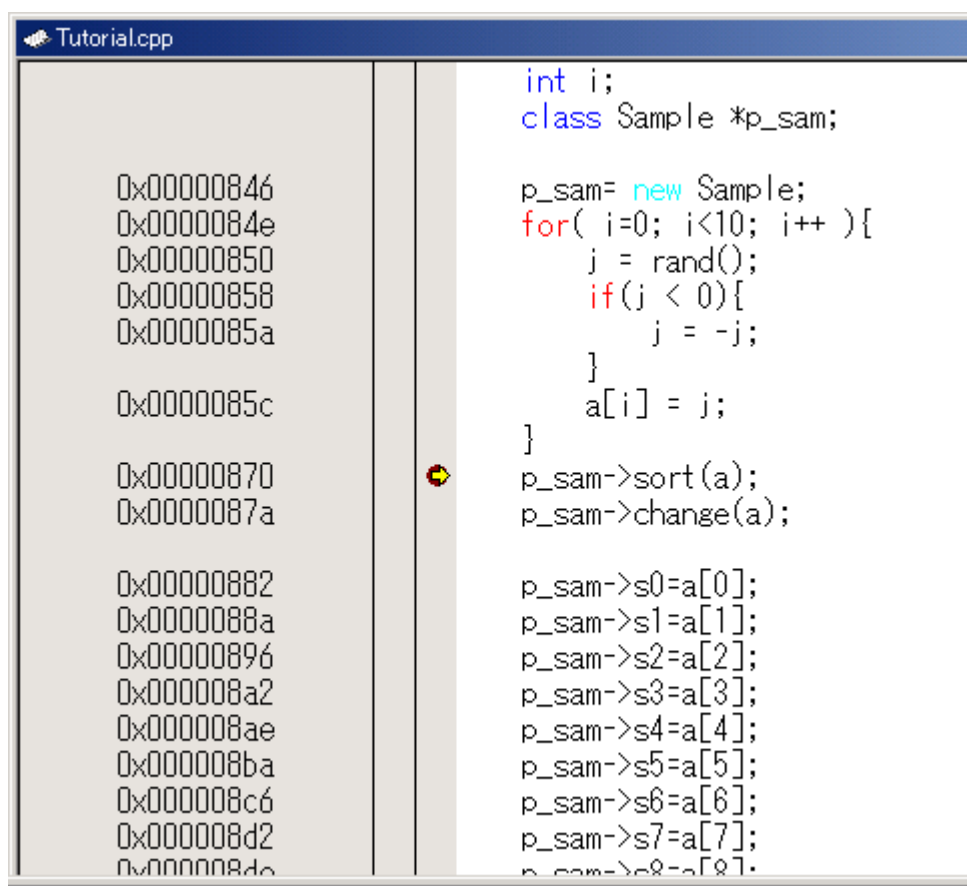


図 6.10 [Source]ウィンドウ (ブレーク状態)

[Status]ウィンドウで最後に発生したブレイクの要因が確認できます。

[表示]メニューの[CPU]サブメニューから[ステイタス]を選択してください。

[Status]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。

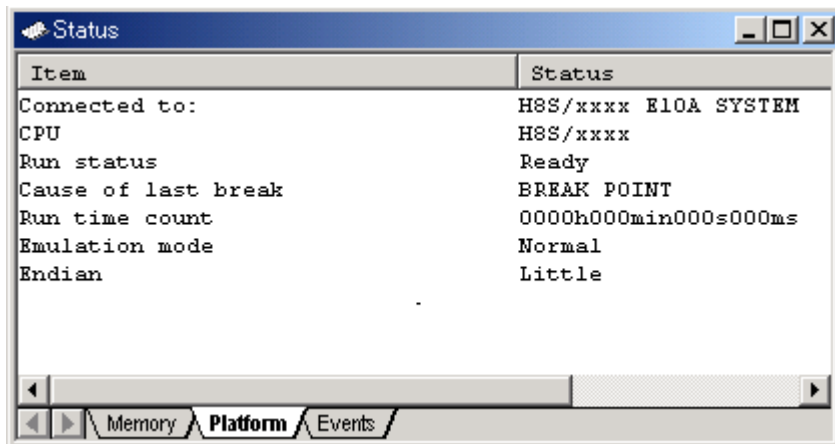


図 6.11 [Status]ウィンドウ

【留意事項】

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、オンラインヘルプを参照してください。

6.10 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、[Event]ウィンドウで確認することができます。

[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[Event]ウィンドウが表示されます。[Breakpoint]シートを開きます。

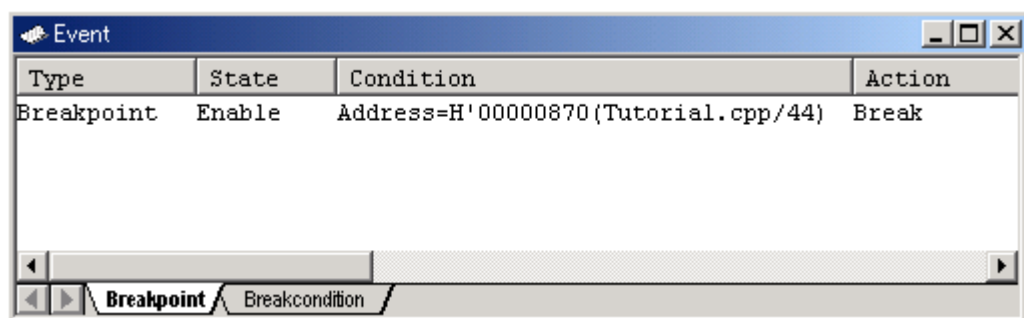


図 6.12 [Event]ウィンドウ

マウスの右ボタンで[Event]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。

6.11 メモリ内容の確認

Label 名を指定することによって、Label が登録されているメモリの内容を[Memory]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、ワードサイズで_main に対応するメモリ内容を確認します。

[表示]メニューの[CPU]サブメニューから[メモリ]を選択し、[先頭アドレス]エディットボックスに”_main”を入力し、[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”Word”に設定してください。

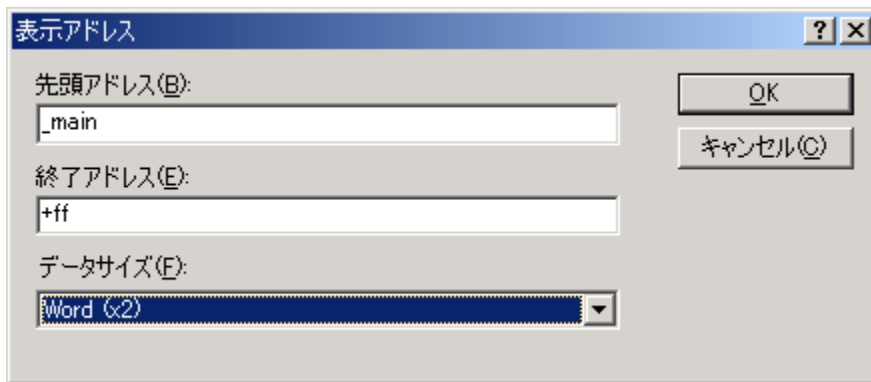


図 6.13 [表示アドレス]ダイアログボックス

[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[Memory]ウィンドウが表示されます。

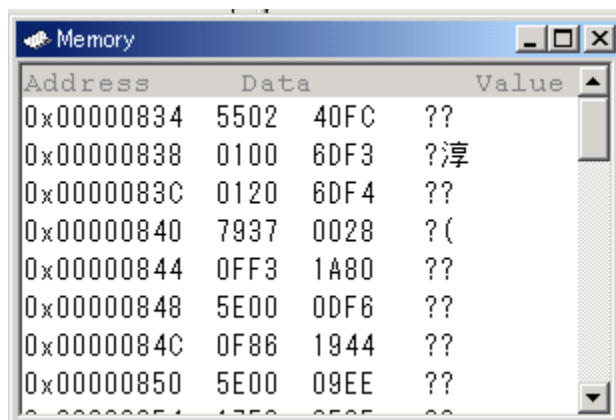


図 6.14 [Memory]ウィンドウ

6.12 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。

[Source]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。

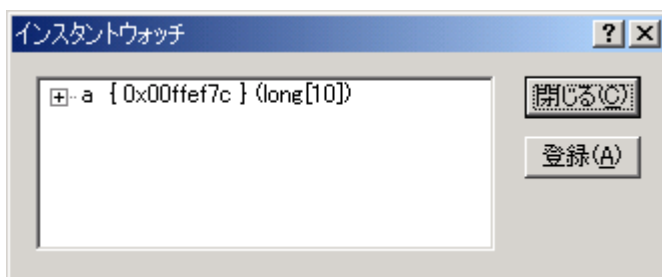


図 6.15 [インスタントウォッチ]ダイアログボックス

[登録]ボタンをクリックして、[Watch]ウィンドウに変数を加えてください。

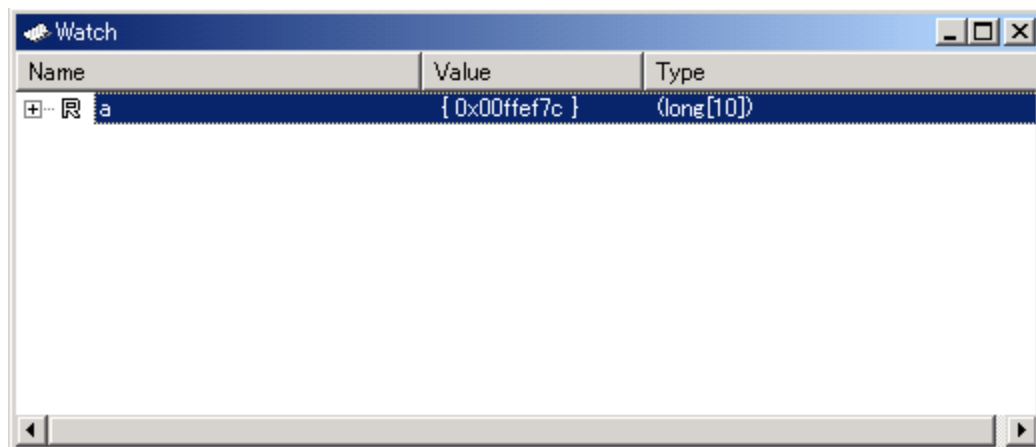


図 6.16 [Watch]ウィンドウ (配列の表示)

また、変数名を指定して、[Watch]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[Watch]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ので、変数 `i` を入力してください。

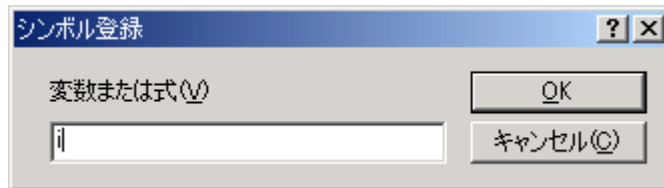


図 6.17 [シンボル登録]ダイアログボックス

[OK]ボタンをクリックします。

[Watch]ウィンドウに、`int` 型の変数 `i` が表示されます。

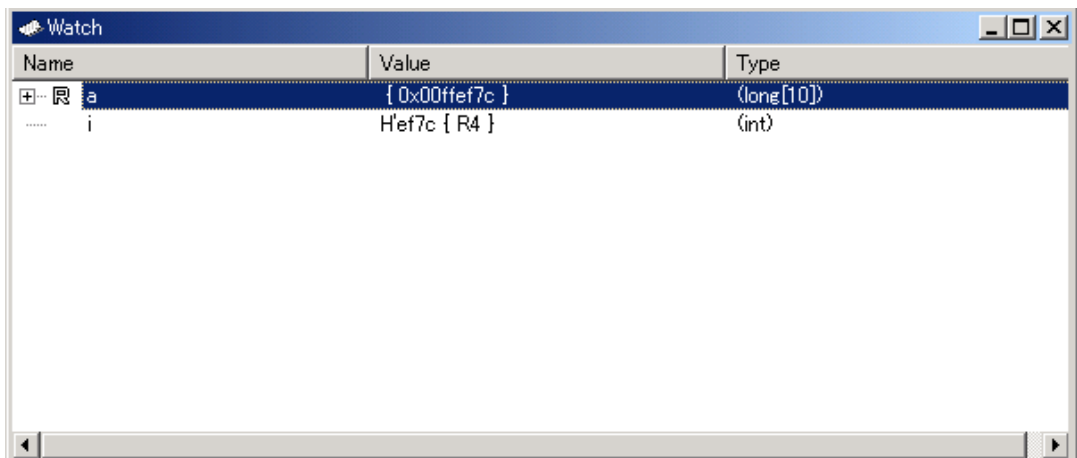


図 6.18 [Watch]ウィンドウ (変数の表示)

[Watch]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。

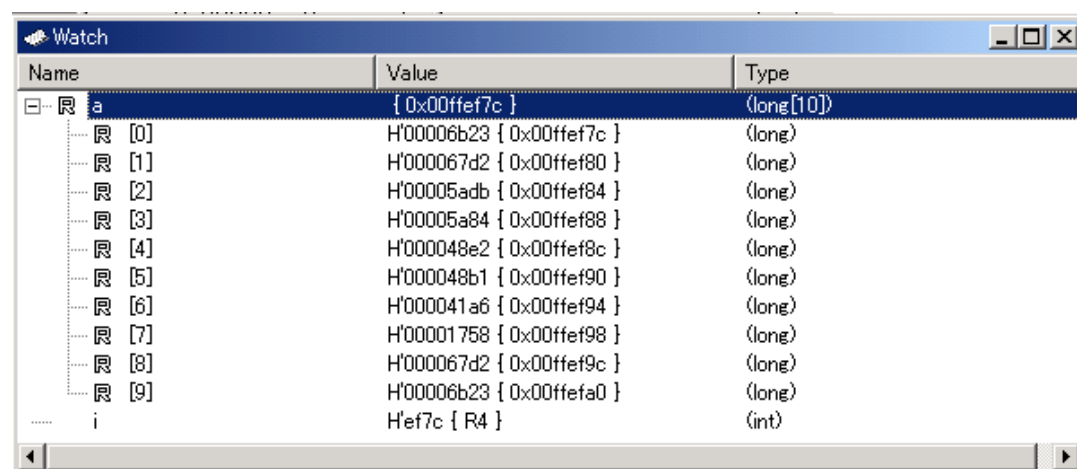


図 6.19 [Watch]ウィンドウ（配列要素の表示）

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	Step In	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	Step Over	関数コールを 1 ステップとして、ステップ実行します。
3	Step Out	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	Step...	指定した速度で指定回数分ステップ実行します。

6.13.1 ステップインの実行

ステップイン機能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 6.20 [ステップイン]ボタン

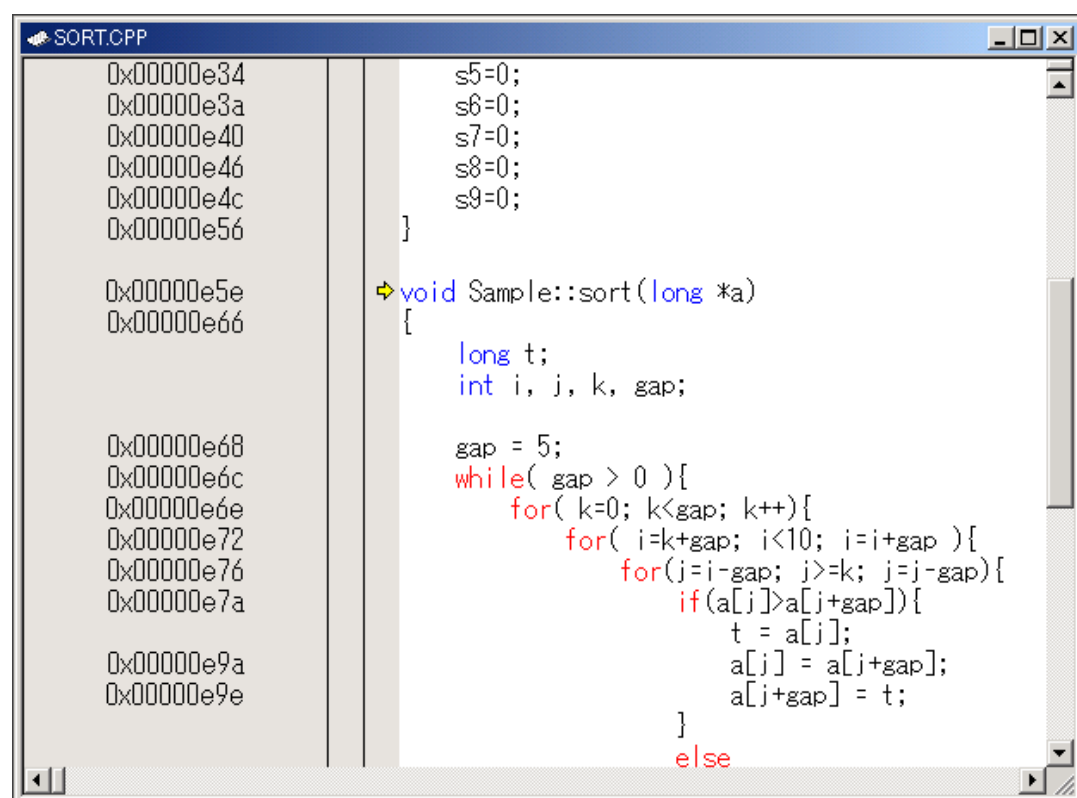


図 6.21 [Source]ウィンドウ (Step In)

[Source]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 ステップアウトの実行

ステップアウト機能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。

【留意事項】

本機能は処理時間がかかります。コール元が分かっている場合は、[カーソル位置まで実行]をご使用ください。



図 6.22 [ステップアウト]ボタン

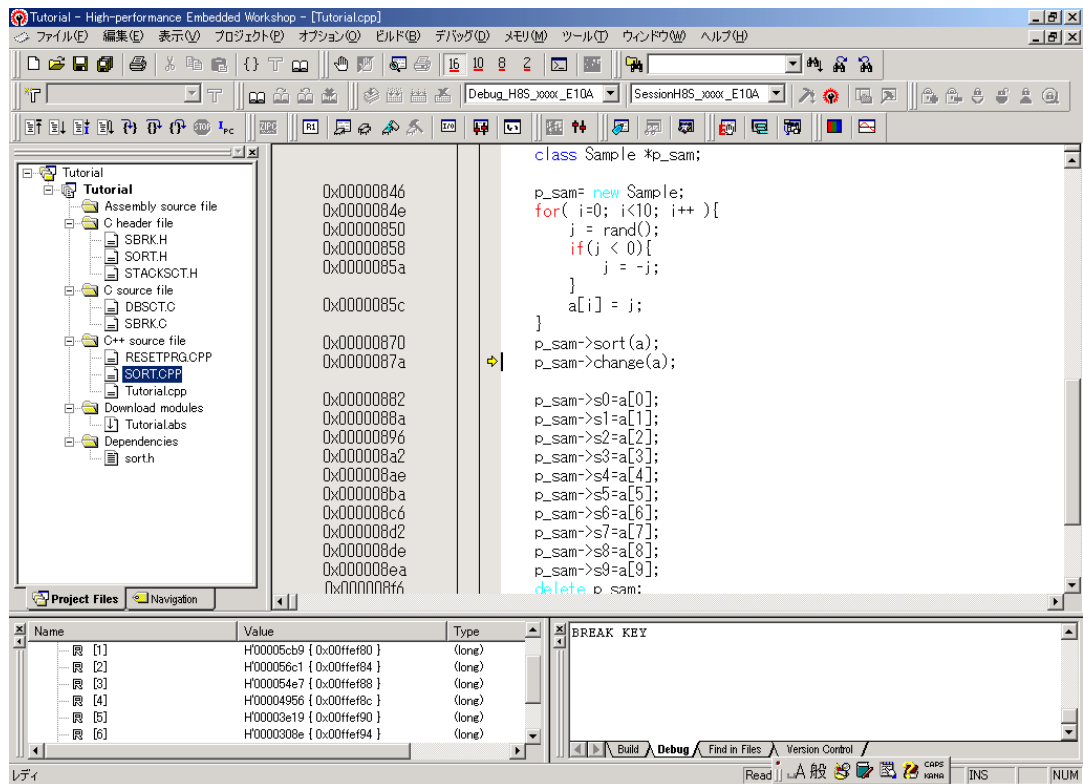


図 6.23 [HEW]ウィンドウ (ステップアウト)

[Watch]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 ステップオーバの実行

ステップオーバ機能は関数コールを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 6.24 [ステップオーバ]ボタン

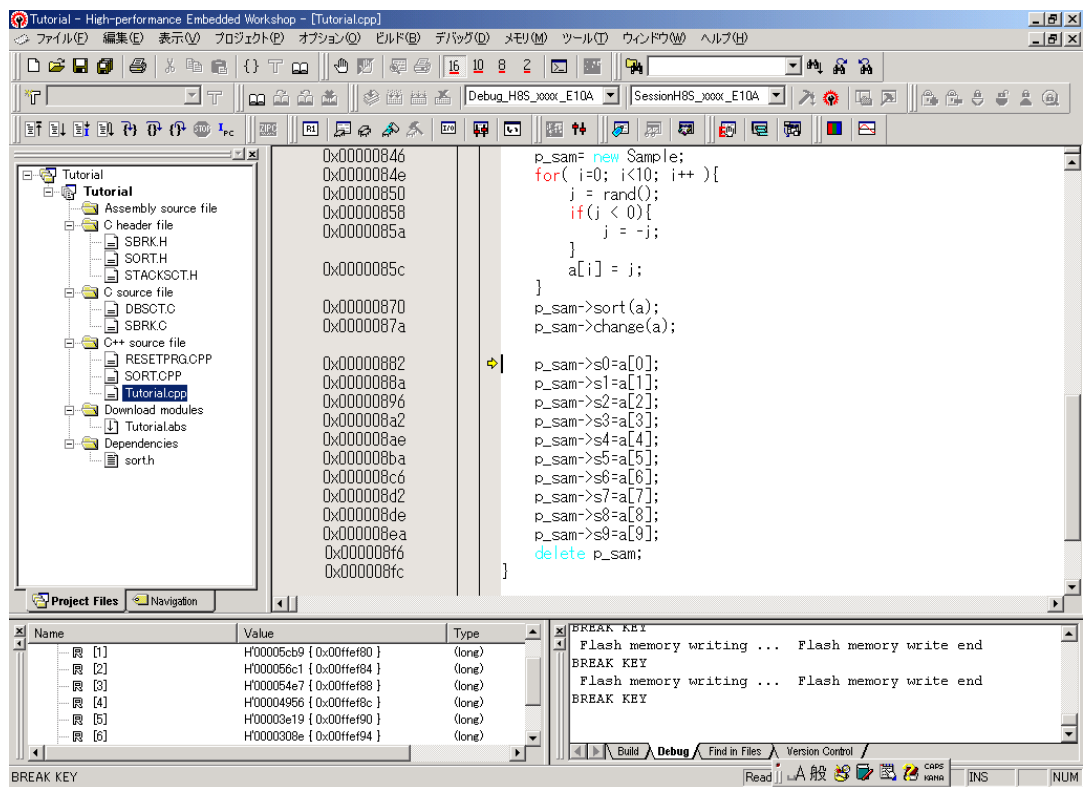


図 6.25 [HIEW]ウィンドウ (ステップオーバ)

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。

ブレークを全て解除してください。

main 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 6.26 [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]ボタンを選択してください。



図 6.27 [停止]ボタン

6.15 ローカル変数の表示

[Locals]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、main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]メニューの[シンボル]サブメニューから[ローカル]を選択してください。[Locals]ウィンドウが表示されます。

ローカル変数が存在しない場合、[Locals]ウィンドウに何も表示されません。

[Locals]ウィンドウには、ローカル変数とその値が表示されます。

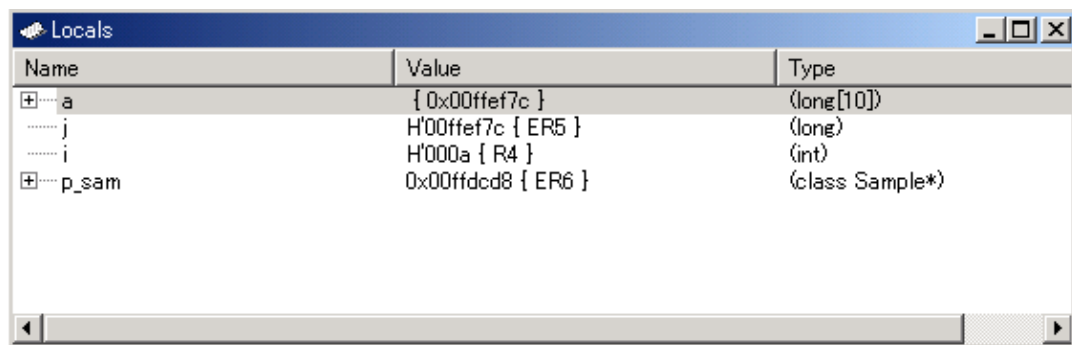


図 6.28 [Locals]ウィンドウ

[Locals]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 の要素を参照すると、ランダムデータが降順にソートされていることがわかります。

6.16 ブレーク機能

E10A エミュレータは、PC ブレーク機能とハードウェアブレーク機能を持っています。

HEW では、PC ブレークポイントの設定を[Event]ウィンドウの[Breakpoint]シートで、また、ハードウェアブレーク条件の設定を[Breakcondition]シート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

E10A エミュレータは、255 ポイントまで PC ブレークを設定することができます。

本章では、6.7 章でご紹介した以外の設定方法を説明します。

[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[Event]ウィンドウが表示されます。

[Breakpoint]シートを開きます。

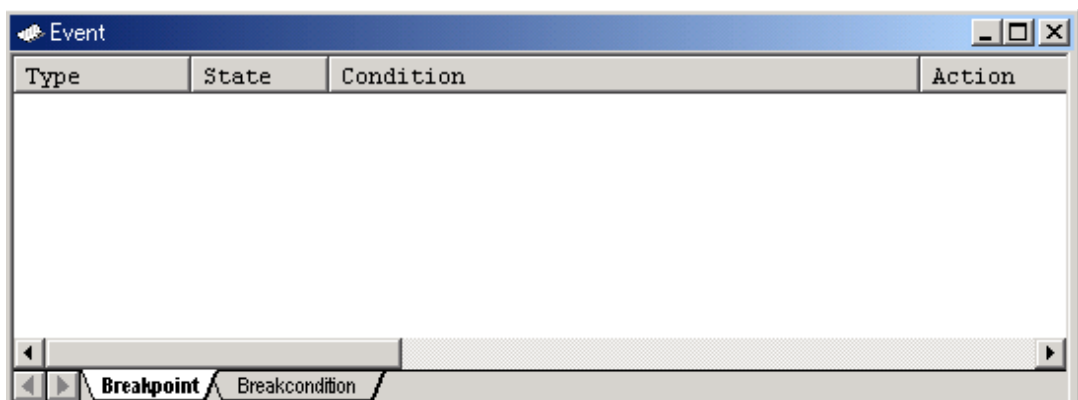


図 6.29 [Event]ウィンドウ (PC ブレーク設定前)

マウスの右ボタンで[Event]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。

[Address]エディットボックスにアドレス H'0000087A を入力してください。

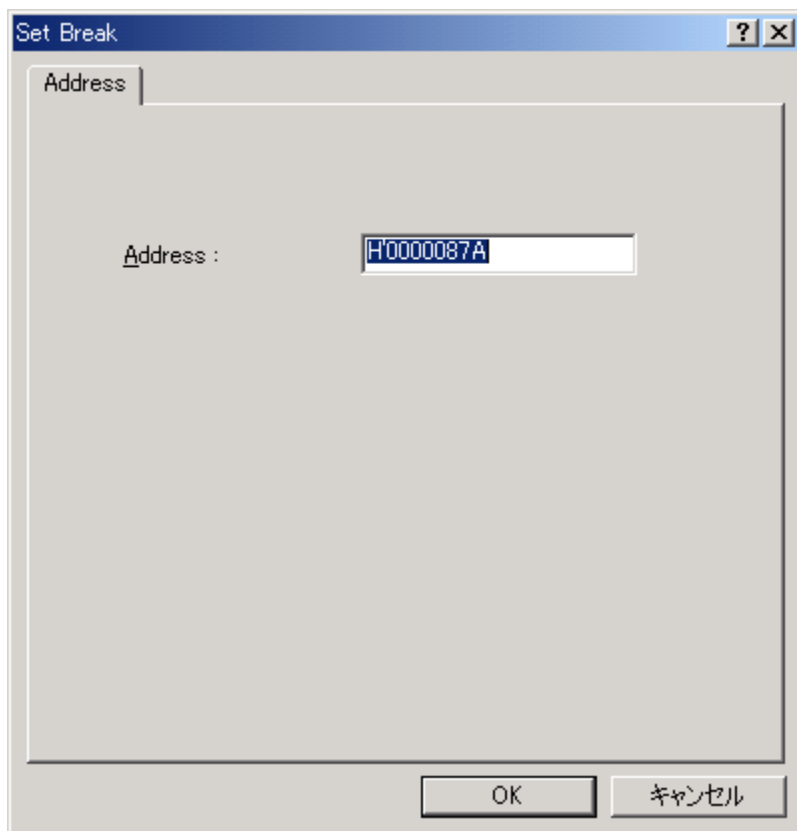


図 6.30 [Set Break]ダイアログボックス

[OK]ボタンをクリックしてください。

[Event]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。

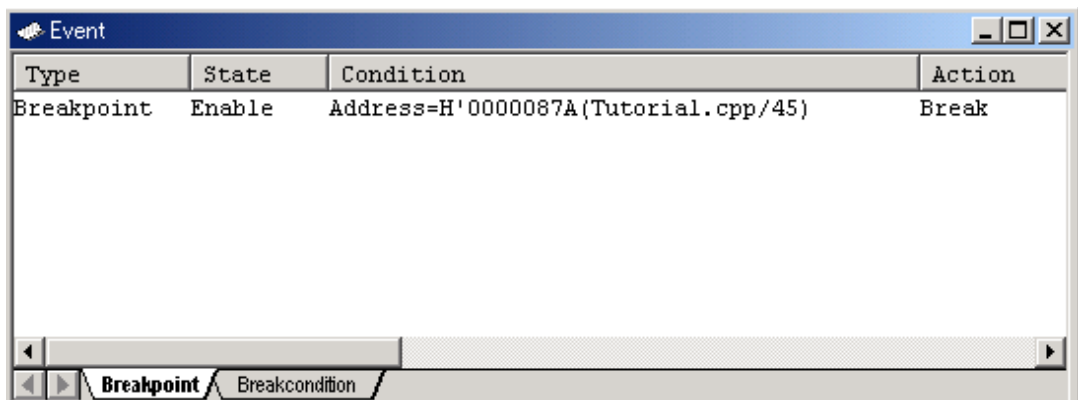


図 6.31 [Event]ウィンドウ (PC ブレーク設定時)

チュートリアルプログラムを PC ブレークポイントで停止させるため、以下の手順を実行してください。

[Event]ウィンドウを閉じてください。

「6.8 レジスタ内容の変更」で設定したプログラムカウンタ (PC=H'400) を[Register]ウィンドウから設定して、[実行]ボタンをクリックしてください。

正常に実行できない場合は、一旦リセットを発行してから上記手順を実行してください。

設定した PC 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。

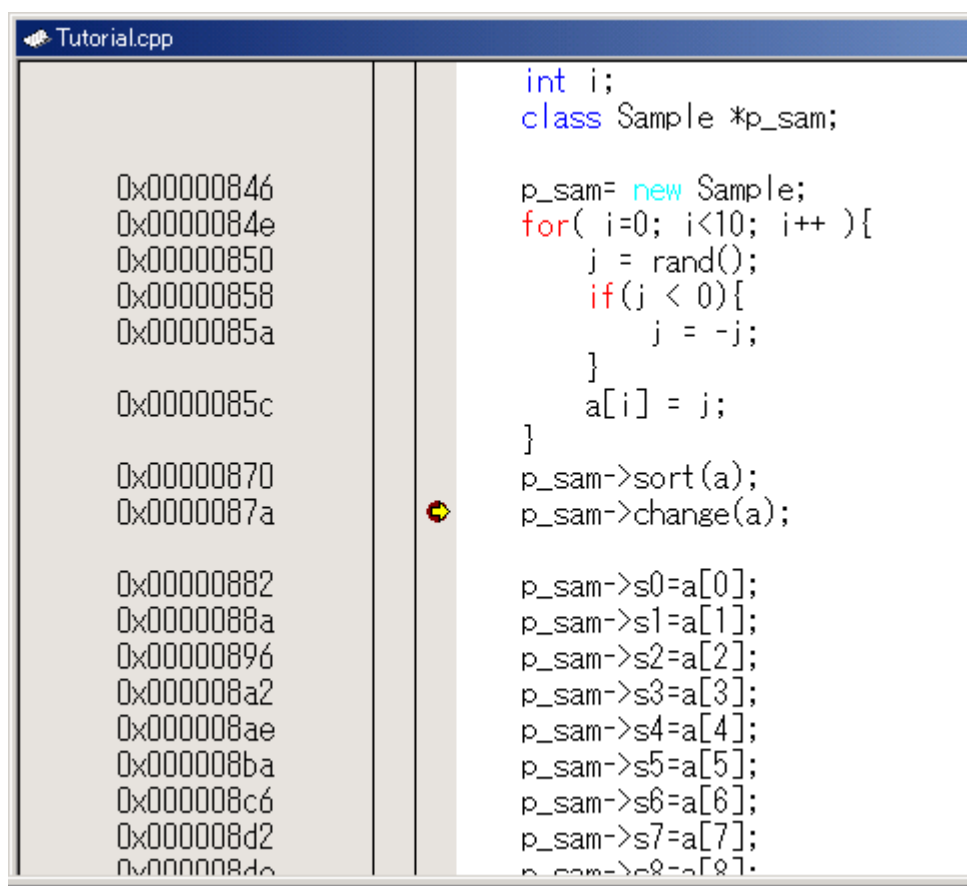
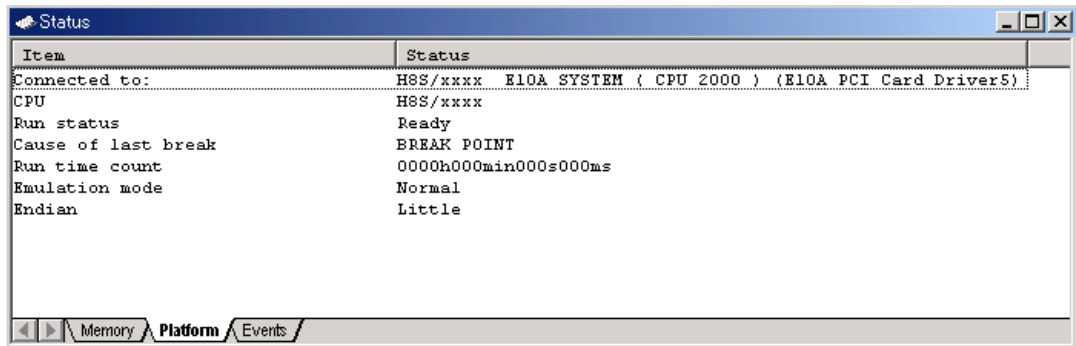


図 6.32 実行停止時の[Source]ウィンドウ (PC ブレーク)

[Status]ウィンドウの表示内容は、以下のようになります。



The screenshot shows a window titled 'Status' with a table of system information. The table has two columns: 'Item' and 'Status'. The items listed are: Connected to:, CPU, Run status, Cause of last break, Run time count, Emulation mode, and Endian. The status values are: H8S/xxxx E10A SYSTEM { CPU 2000 } (E10A PCI Card Driver5), H8S/xxxx, Ready, BREAK POINT, 0000h000min000s000ms, Normal, and Little. At the bottom of the window, there are three tabs: 'Memory', 'Platform' (which is selected), and 'Events'.

Item	Status
Connected to:	H8S/xxxx E10A SYSTEM { CPU 2000 } (E10A PCI Card Driver5)
CPU	H8S/xxxx
Run status	Ready
Cause of last break	BREAK POINT
Run time count	0000h000min000s000ms
Emulation mode	Normal
Endian	Little

図 6.33 [Status]ウィンドウの表示内容 (PC ブレーク)

【留意事項】

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、オンラインヘルプを参照してください。

6.17 ハードウェアブレイク機能

ハードウェアブレイク条件 Break Condition 1 にアドレスバス条件を設定する方法を説明します。

[表示]メニューの[コード]サブメニューから[イベントポイント]を選択してください。[Event]ウィンドウが表示されます。

先ほど設定した PC ブレイクポイントを削除します。マウスの右ボタンで[Event]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 PC ブレイクポイントをすべて解除してください。

次は Break Condition を設定します。

[Breakcondition]タブをクリックしてください。

[Event]ウィンドウ内の Break condition 1 行を選択してください。Break condition 1 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。

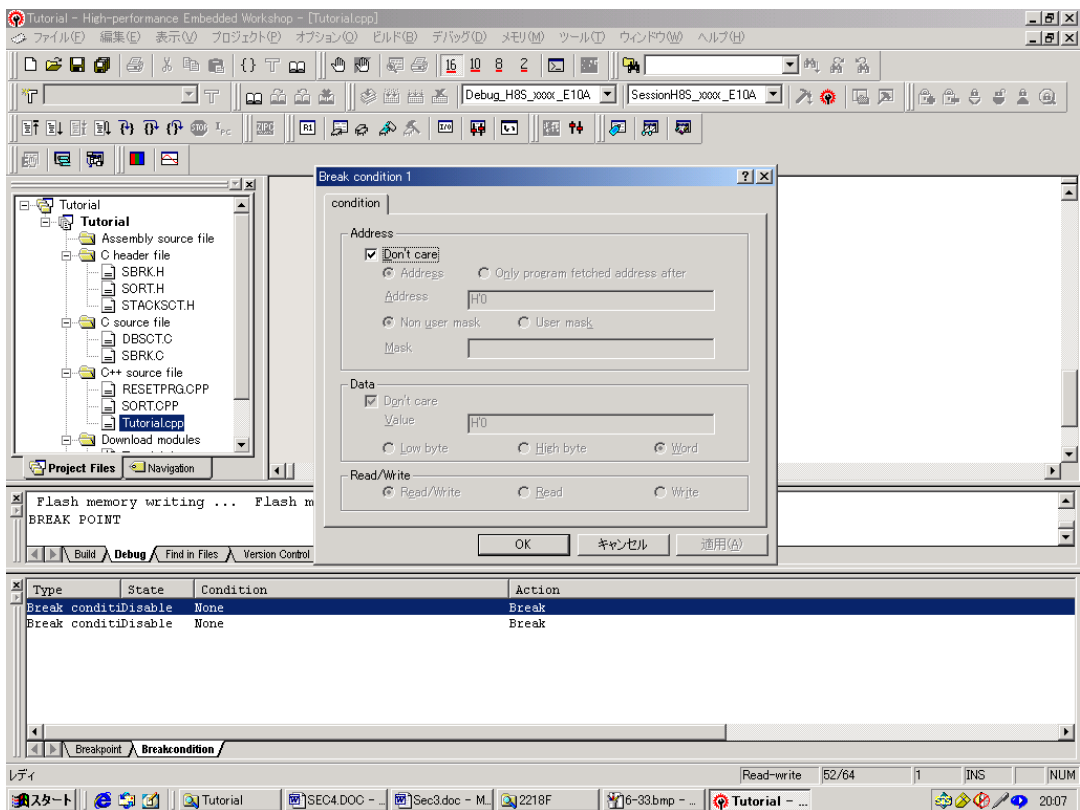


図 6.34 [HEW] ウィンドウ ([Break condition 1])

[Break condition 1]ダイアログボックスが表示されます。

[Address]グループボックスの[Don't care]チェックボックスを無効にします。

[Only program fetched address after]ラジオボタンを選択して、値として[Address]エディットボックスにアドレス H'896 を入力してください。

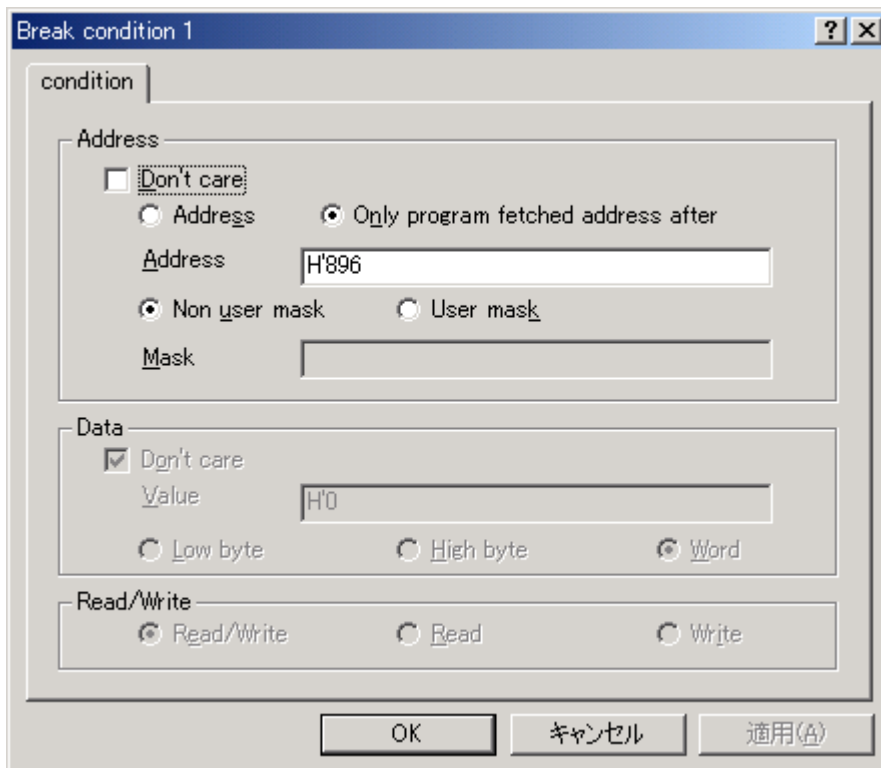


図 6.35 [condition]ページ ([Break condition 1]ダイアログボックス)

[OK]ボタンをクリックしてください。

State 行の 1 ポイント目の表示が”Disable”から”Enable”に変わります。

Condition 行の 1 ポイント目の表示が”None”から”Address=H’00000896(Tutorial.cpp/49) pcafter”に変わります。

「6.8 レジスタ内容の変更」で設定したプログラムカウンタ（PC=H’400）を[Register]ウィンドウに設定して、[実行]ボタンをクリックしてください。

Break Condition1 の条件まで、プログラムを実行して停止します。

【留意事項】

本ページ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、オンラインヘルプを参照してください。

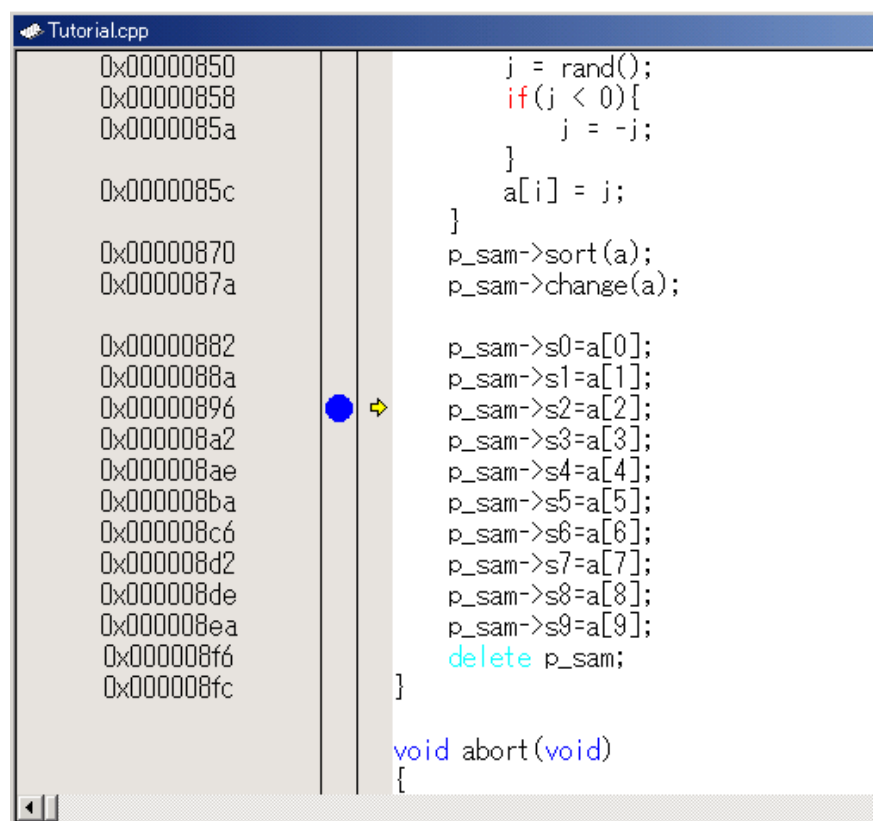


図 6.36 実行停止時の[Source]ウィンドウ (Break Condition 1)

[Status]ウィンドウの表示内容は、以下のようになります。

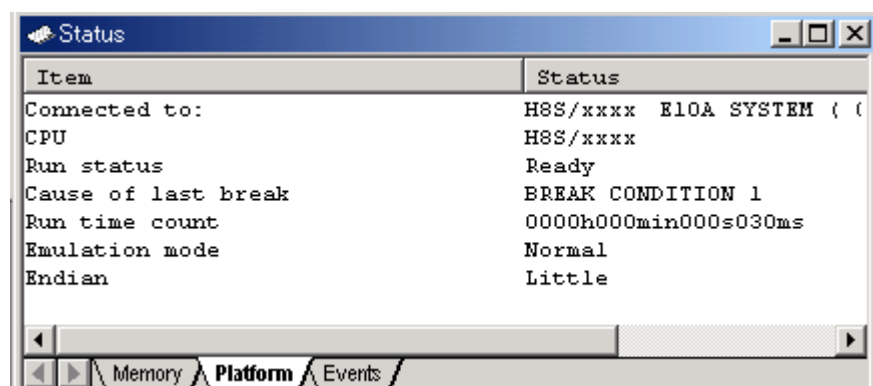


図 6.37 [Status]ウィンドウの表示内容 (Break Condition 1)

【注意事項】

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、オンラインヘルプを参照してください。

6.18 トレース機能

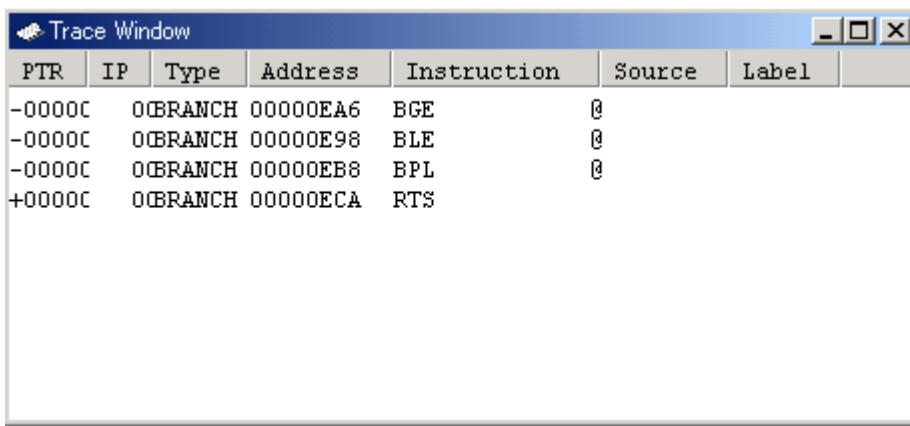
E10A エミュレータには分岐命令トレース機能があります。

4 分岐分の分岐元アドレスと、ニーモニック、オペランド、ソース行、ラベルを表示します。

6.18.1 トレースウィンドウの表示方法

[表示]メニューの[コード]サブメニューから[トレース]を選択してください。

「6.16.1 PC ブレーク機能」の例でプログラムを実行してください。実行停止後に[Trace Window]ウィンドウにトレース結果を表示します。



PTR	IP	Type	Address	Instruction	Source	Label
-0000C	00BRANCH	00000EA6	BGE		@	
-0000C	00BRANCH	00000E98	BLE		@	
-0000C	00BRANCH	00000EB8	BPL		@	
+0000C	00BRANCH	00000ECA	RTS			

図 6.38 [Trace Window]ウィンドウ

必要ならば、タイトルバーの下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。

【留意事項】

トレース取得できる分岐命令の数、トレース表示内容は、製品によって異なります。各製品の仕様については、オンラインヘルプを参照してください。

6.19 スタックトレース機能

E10A 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。

【留意事項】

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。

Dward2 形式のデバッグ情報を持ったロードモジュールは、H8SC/C++コンパイラ V3.0 以降でサポートしています。

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。

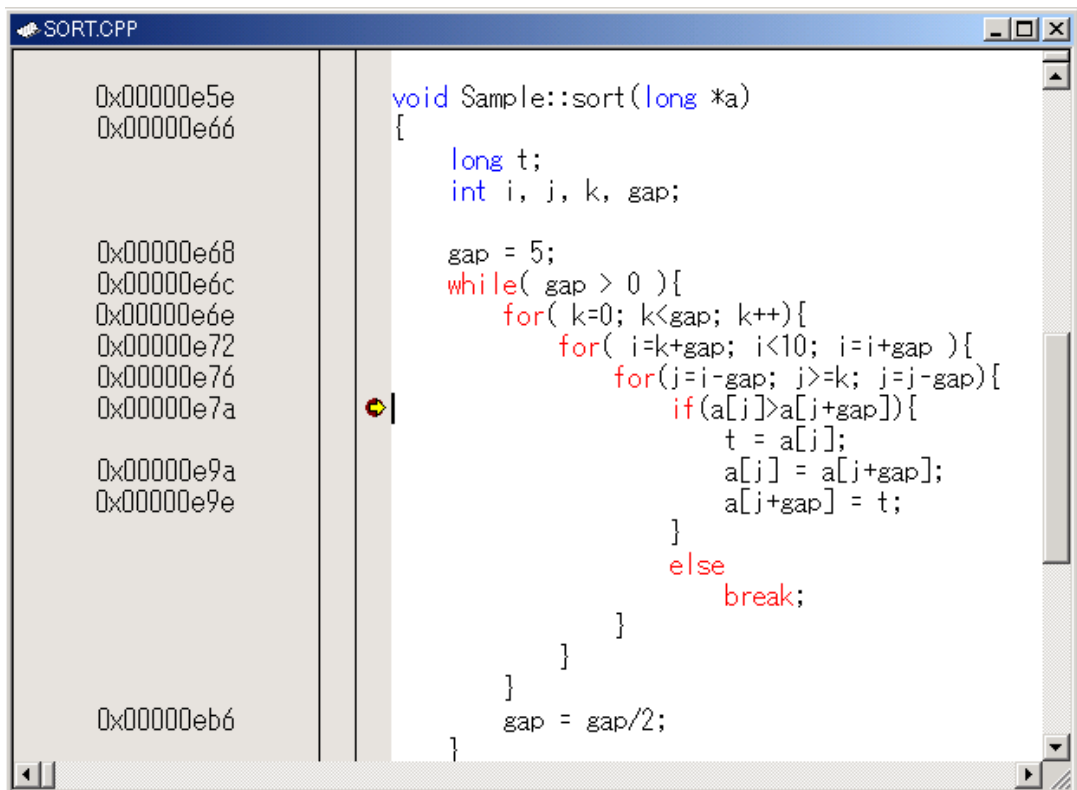


図 6.39 [Source]ウィンドウ (PC ブレークポイントの設定)

「6.8 レジスタ内容の変更」で設定したプログラムカウンタ (PC=H'400) を[Register]ウィンドウに設定して、[実行]ボタンをクリックしてください。

プログラムブレーク後、[表示]メニューの[コード]サブメニューから[スタックトレース]を選択し[Stack Trace]ウィンドウを開いてください。

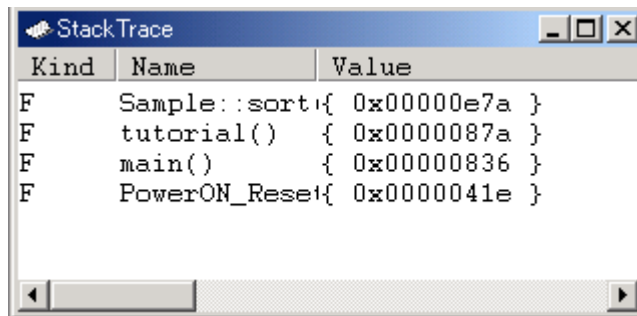


図 6.40 [Stack Trace]ウィンドウ

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。

6.20 さてつぎは？

このチュートリアルでは、E10A 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。

E10A 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。

付録 A トラブルシューティング

1. 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。

2. 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。

3. 日本語版 Windows® 98/Me、Windows NT®, Windows® 2000、Windows® XP 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

[Tools->オプション...] を選んで、“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。

4. 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。

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

特殊文字	機能
?	任意の一文字 (改行文字を除く) に一致
*	任意の文字列 (改行文字を除く) に一致
¥n	改行文字に一致
¥t	タブ文字に一致
[]	かっこ内に列挙した、または、かっこの範囲の任意の一文字に一致
¥	以下に続く正規表現文字を無視

注意 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。

特殊文字	?
意味	改行文字以外の任意の一文字と一致します。
例	t?p は “top”や “tip” と一致するが “trap”とは不一致。
特殊文字	*
意味	改行文字以外の文字列 (0 個以上の文字列) と一致します。二行にまたがって一致することはありません。* 文字は残りのパターンが一致するための最短文字列と一致します。
例	t*o は “too”の“to”, “trowel”の“tro”, “sporty orange” の“ty o”と一致するが、 “smart orange”とは一致しない (* 文字は二行にまたがって一致することはないため)。
特殊文字	¥n
意味	改行文字と一致する。 ¥n は行の終わりや 2 行にわたるパターンを検索するときに使います。
例 1:	¥n 行末のセミコロンと一致する。
例 2:	¥nif セミコロンと改行文字が続き次に if で始まる行が続く場合を検索する。

特殊文字	<code>\t</code>
意味	タブと一致します。
例 1:	<code>\t8</code> タブの次に 8 がある場合を検索する。
例 2:	<code>init\t</code> <code>init</code> の次にタブがある場合を検索する。

特殊文字	<code>[]</code>
意味	かっこ内の文字または文字の範囲の中の任意の一字と一致します。かっこをネストすることはできません。 <code>[~]</code> は文字の範囲を示します。例 <code>[a-z]</code> 、 <code>[0-9]</code> 。範囲の始まりの文字は終わりの文字より小さい ASCII コードでなければなりません。 <code>[~]</code> は <code>[~</code> と <code>]</code> の間の文字以外の一字と一致します。改行文字がこのかっこ内になれば、改行文字とも一致します。
例 1:	<code>[AEIOU]</code> 大文字の母音をすべて検索する。
例 2:	<code>[<>?]</code> 記号 <code><</code> , <code>></code> , <code>?</code> を検索する。
例 3:	<code>[A-Za-z0-9_]</code> 大文字、小文字、数字、下線を検索する。
例 4:	<code>[~0-9]</code> 数字以外の文字を検索する。
例 5:	<code>[\t\n]</code> 空白、タブ、改行文字を検索する。
例 6:	<code>[\t]</code> <code>\t</code> のすぐ後に <code>]</code> を置くと、記号 <code>]</code> を検索する。

特殊文字	<code>\</code>
意味	正規表現を無視する特殊文字です。バックスラッシュの次の文字が正規表現のとき、通常の文字として扱います。バックスラッシュの次が非正規表現のとき、バックスラッシュは無視されます。
例 1:	<code>*</code> アスタリスクをすべて検索します。
例 2:	<code>\\</code> バックスラッシュをすべて検索します。

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 の三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四つあります。

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。

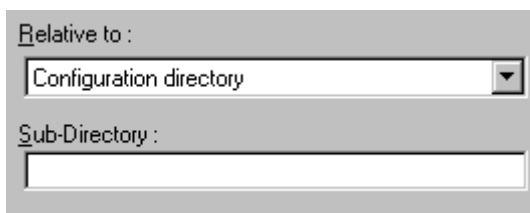


図 C.2: プレースホルダドロップダウンリストと Sub-Directory フィールド

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
\$(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
\$(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
- 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 [Tools->Version Control->Configure...] 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フレキシブルに指定できます。

- 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。

D.1 ファイルフォーマット

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。

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

モジュール

```
[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
...
```

モジュールの定義

```
[DMA_Channel_Common]
reg0=regDMAWER
reg1=regDMATCR
reg2=regDMACR0A
reg3=regDMACR0B
reg4=regDMACR1A
reg5=regDMACR1B
reg6=regDMABCRH
reg7=regDMABCRH
dep= regMSTPCR 7 0
```

レジスタ名
ビット
値

レジスタの定義

```
[regDMAWER]
id=DMAWER 0xffff00 B A H dmawer_bitfields
```

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

ビットフィールドの定義

```
[dmawer_bitfields]
bit3=WE1B
bit2=WE1A
bit1=WE0B
bit0=WE0A
```

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。

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。

ファイルは、ワード[BEGIN]で始めなければなりません。

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。

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 ウィンドウ機能一覧

E10A エミュレータ用 HEW で使用できるウィンドウ機能の一覧表を示します。
ここでは、HEW 共通の機能は省き、エミュレーション用機能のみをご紹介します。

表 F.1 ウィンドウ機能一覧

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		[Disassembly]ウィンドウを表示します
	コマンドライン		Ctrl+L		[Command Line]ウィンドウを表示します
	ワークスペース		Alt+K		[Workspace]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
	CPU	レジスタ	Ctrl+R		[Register]ウィンドウを表示します
		メモリ...	Ctrl+M		[Memory]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		[Status]ウィンドウを表示します
		日本語メモリダンプ...	Shift+Ctrl+D		[Localized Dump]ウィンドウを表示します
	シンボル	ラベル	Shift+Ctrl+A		[Label]ウィンドウを表示します
		ウォッチ	Ctrl+W		[Watch]ウィンドウを表示します
		ローカル	Shift+Ctrl+W		[Locals]ウィンドウを表示します
	コード	イベントポイント	Ctrl+E		[Event]ウィンドウを表示します
		トレース	Ctrl+T		[Trace Window]ウィンドウを表示します
		スタックトレース	Ctrl+K		[Stack Trace]ウィンドウを表示します
グラフィック	画像...	Shift+Ctrl+G		[Image]ウィンドウを表示します	
	波形...	Shift+Ctrl+V		[Waveform View]ウィンドウを表示します	

表 F.1 ウィンドウ機能一覧 (続き)

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] 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			ターゲットマイコンをリセットし、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		ターゲットマイコンをリセットし、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。
	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、サブルーチン呼び出しの場合は、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

表 F.1 ウィンドウ機能一覧 (続き)

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	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			[Source] 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実行します。 [Disassembly] 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
	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
	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 製品によっては使用できません
	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
	最新の情報に更新				全ての[Memory]ウィンドウの内容を強制的に更新します
	オーバーレイの構成...				オーバー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 G コマンドライン機能

E10A エミュレータでは、コマンドラインウィンドウで利用できるコマンドをサポートしています。

コマンドの詳細はオンラインヘルプをご覧ください。

付録 H HEW の注意事項

(1) ロードモジュール作成後のソースファイル位置移動に関する注意事項

ロードモジュール作成後にソースファイルを移動させた場合、作成したロードモジュールのデバッグ中にソースファイルを指定するための[Open]ダイアログボックスが表示されることがあります。対応するソースファイルを選択し、Openボタンを押してください。

(2) ソースレベル実行機能

ソースファイル

ロードモジュールに対応しないソースファイルをプログラムウィンドウに表示しないでください。ロードモジュールに対応するソースファイルと同名のファイルをプログラムウィンドウに表示するとアドレス表示しますが、そのプログラムウィンドウでは操作できません。

Step

標準Cライブラリ等にも移行します。上位関数に戻るにはStep Outを使用してください。また、forおよびwhile文では、1回のステップでは次の行に進みません。進める場合はもう一度ステップしてください。

(3) ファイルアクセス中の操作について

ロードモジュールのダウンロード中、[Verify Memory]、[Save Memory]、[Trace Window]ウィンドウでのセーブ処理中に他の操作を行わないでください。ファイルアクセス処理が正しく実行されない場合があります。

(4) ウォッチ機能

最適化時の局所変数

最適化オプションでコンパイルされたCソースの局所変数表示は、生成されたオブジェクトコードによって、正しく表示できないことがあります。[Disassembly]ウィンドウを表示し、生成されたオブジェクトコードを確認してください。

また、指定した局所変数の割り付け領域がない場合があります。この場合、次のように表示します。

例) 変数名を asc とする。

asc = ? - target error 2010 (xxxx)

変数名の指定

変数名でないシンボル名(関数名)等を指定した場合、内容は表示しません。

例) 関数名を main とする。

main =

付録 H HEW の注意事項

メモリ内容の変更

[Memory]ウィンドウや[Watch]ウィンドウにおいて、メモリ内容を変更する場合、入力するデータに日本語文字列を指定しないでください。日本語文字列を入力する場合は、Localized Dumpを使用してください。

(5) ラインアセンブル機能

入力基数

ラインアセンブル時の入力基数のデフォルトはRadix設定に関係なく、10進数です。16進数で指定する場合は、H'または0xを指定してください。

(6) コマンドラインインタフェース

バッチファイル

バッチファイル実行中に、“Not currently available”が表示される場合は、sleepコマンドを挿入してください。sleepさせる時間は動作環境によって異なりますので、調整してください。

例) memory_fillで、“Not currently available”を表示する場合

```
sleep d'3000
```

```
memory_fill 0 ffff 0
```

コマンドファイルでのファイル指定

コマンドファイルの指定方法によりカレントディレクトリが移動する場合があります。コマンドファイル内のファイル指定は、カレントディレクトリの移動に影響をうけないように絶対パスで記述することをお勧めします。

例) FILE_LOAD

```
C:\¥¥Hew2¥¥Tools¥¥Renesas¥¥DebugComp¥¥Platform¥¥E10A¥¥Tutorial
```

```
¥¥Debug_H8S_xxxx_E10A_SYSTEM¥¥Tutorial.abs
```

(7) ユーザプログラム実行中のメモリセーブ

ユーザプログラムの実行中は、メモリセーブ／バリファイを実行しないでください。

(8) モトローラ S タイプ形式のファイルのロード

HEWでは、レコード末尾が“CRコード”(H'0D)のみのモトローラSタイプ形式ファイルはサポートしていません。モトローラSタイプ形式のファイルをロードする場合は、レコード末尾に“CRコードとLFコード”(H'0D0A)がついている形式のものを使用してください。

(9) [IO]ウィンドウ

表示と変更

アドレスブレイクは、E10Aエミュレータが使用するため、値の変更は行わないでください。

E10Aエミュレータでは、[IO]ウィンドウの無効モジュール表示およびビット情報表示はサポートしていません。

(10) プログラム実行中の[Register]ウィンドウ動作に関する注意事項

プログラム実行中は、[Register]ウィンドウからレジスタ値を変更できません。表示されますが、変更してもレジスタ内容は変更されません。

(11) ブレーク機能

フラッシュメモリ領域にPCブレークポイントを設定すると、ユーザプログラムを実行するたびにフラッシュメモリへのプログラム書き込みを行います。書き換え可能な回数が減少しますのでご注意ください。

BREAKPOINT 解除

BREAKPOINTを設定したアドレスの内容がユーザプログラム実行中に変更されるとユーザプログラム停止後に以下のメッセージが表示されます。

BREAKPOINT IS DELETED A=xxxxxxx

上記メッセージが表示された場合は、[Breakpoints]ウィンドウの[Delete All]ボタンまたは[Disable]ボタンにより、すべてのBREAKPOINT設定を解除してください。

(12) BREAKPOINT の設定数と[Run...]メニューの[Stop At]の設定数

BREAKPOINTの設定数と[Run...]メニューの[Stop At]の設定数の合計は、最大255個です。したがってBREAKPOINTを255個設定した状態では、[Run...]メニューの[Stop At]での指定は無効となります。BREAKPOINTと[Run...]メニューの[Stop At]は、設定数の合計が255個以下で使用してください。

(13) RUN-TIME 表示における注意事項

E10Aエミュレータでは、[ステイタス]ウィンドウにおいてユーザプログラムの実行時間を表示していますが、ホストコンピュータ側のタイマを使用していますので、正確な値ではありません。

(14) Timeout error 表示時の注意事項

Timeout errorが表示された場合、E10Aエミュレータとターゲットマイコンの通信が取れなくなっています。この場合、E10Aエミュレータとユーザシステムの電源を一旦OFFにし、HEWからE10Aエミュレータを再接続してください

(15) Double float 形式のサポート

以下のメモリ操作においては、Double float形式をサポートしていません。

- [Fill Memory]ダイアログボックス
- [Search Memory]ダイアログボックス
- MEMORY_FILL コマンド

また、[Copy Memory]ダイアログボックスの[Format]指定は無視します。メモリコピーはすべてバイト単位に行います。

(16) [Run Program]ダイアログボックスご使用時の注意事項

[デバッグ]メニュー -> [ラン...]を選択して停止アドレスを指定する際に以下の注意事項があります。

Disableに設定しているブレークポイントを停止アドレスと設定した場合、ユーザプログラム停止時にブレークポイントがEnableになりますのでご了承ください。

(17) ユーザプログラム実行中のメモリアクセス

ユーザプログラム実行中にメモリウィンドウ等からメモリアクセスした場合、E10Aエミュレータ内部でユーザプログラムの実行を一旦停止してメモリアクセスし、その後ユーザプログラムを再実行しています。したがって、ユーザプログラムのリアルタイム性はありません。

(18) 内蔵フラッシュメモリ

E10Aエミュレータを接続して使用したMCUは、エミュレーション時にフラッシュメモリの書き込みを繰り返しておりストレスがかかっています。デバッグに使用したMCUをユーザの量産製品には使用しないでください。

デバッグに使用していないMCUであれば、量産用の書き込みツールとしてもお使いいただけます。

(19) メモリのベリファイ

メモリのベリファイを行う場合は、FILE_VERIFYコマンドを使用してください。[ファイル] メニューの [メモリのベリファイ] は使用しないでください。

(20) [IO] ウィンドウからレジスタの値を編集する場合の注意事項

レジスタの値の編集時に、編集前の値と編集後の値が同じとなる場合は、メモリへの書き込みが発生しません。

(21) ファイルの上書き

コマンドラインインタフェースでは同名のファイルが存在しても、ユーザに通知せずに上書きします。

(22) 他製品との共存

HEW2エミュレータソフトウェアをインストールした後で、SuperH RISC engine C/C++コンパイラパッケージまたはH8S、H8/300 Series C/C++コンパイラパッケージをインストールする場合は、コンパイラパッケージのインストール中に [コンポーネントの選択] にて、[High-performance Embedded Workshop] をインストールするコンポーネントに設定してください。

H8Sファミリ用 E10Aエミュレータ ユーザーズマニュアル

発行年月 2003 年 8 月 1 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

H8S ファミリ用 E10A エミュレータ
ユーザズマニュアル
H8S ファミリ E10A HS0005KCM05HJ



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

RJJ10B0014-0100H