お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ（フラッシュメモリ・SRAM等）を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ（http://www.renesas.com）

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

RENESAS

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しては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (http://www.renesas.com)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任は負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

User's Manual

H8/3864シリーズ、H8/3887シリーズ E6000 エミュレータ

ユーザーズマニュアル

ルネサスマイクロコンピュータ開発環境システム

HS3880EPI60H

ルネサス エレクトロニクス
www.renesas.com

Rev.1.00　1998.09

ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合，または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して，弊社もしくは第三者の特許権，著作権，商標権，その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合，弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので，最終的な設計，ご購入，ご使用に際しましては，事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが，宇宙，航空，原子力，燃焼制御，運輸，交通，各種安全装置，ライフサポート関連の医療機器等のように，特別な品質・信頼性が要求され，その故障や誤動作が直接人命を脅かしたり，人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は，事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては，特に最大定格，動作電源電圧範囲，放熱特性，実装条件及びその他諸条件につきましては，弊社保証範囲内でご使用いただきますようお願い致します。
   保証値を越えてご使用された場合の故障及び事故につきましては，弊社はその責を負いません。
   また保証値内のご使用であっても半導体製品について通常予測される故障発生率，故障モードをご考慮の上，弊社製品の動作が原因でご使用機器が人身事故，火災事故，その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ，ご相談は弊社営業担当迄お願い致します。

重要事項

・当エミュレータをご使用になる前に、
必ずユーザーズマニュアルをよく読んで理解してください。

・ユーザーズマニュアルは、必ず保管し、使用上不明な点がある場合は再読してください。

エミュレータとは：

ここでいうエミュレータとは、株式会社日立製作所（以下、「日立」という。）が製作した次の製品を指します。
（1）E6000 エミュレータ本体、（2）ユーザシステムインタフェースケーブル、（3）PC インタフェースボード　お客様のユーザシステム及びホストコンピュータは含みません。

エミュレータの使用目的：

当エミュレータは、日立マイクロコンピュータ H8/3864 シリーズ、H8/3887 シリーズ、（以下、MCU と略します）を使用したシステムの開発を支援する装置です。ソフトウェアとハードウェアの両面から、システム開発を支援します。
この使用目的に従って、当エミュレータを正しくお使いください。この目的以外に当エミュレータを使用することを堅くお断りします。

使用制限：

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

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

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

製品の変更について：

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

エミュレータを使う人は：

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

特に、当エミュレータを初めて使う人は、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すすめします。

保証の範囲：

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

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

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

その他の重要事項：

１．本資料に記載された情報、製品または回路の使用に起因する損害または特許権その他権利の侵害に関しては、日立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日立の特許権その他権利の実施権を許諾するものではありません。

版権所有：

図について：

このユーザーズマニュアルの図の一部は、実物と異っていることがあります。

予測できる危険の限界：

日立は、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

危険は、**切迫**した危険な状況で**回避しない場合には、**
死亡または**重傷**を負うことになりうることを定義します。
ただし、本製品では該当するものはありません。

警告は、**潜在的**に危険な状況で**回避しない場合には、**
死亡または**重傷**を負うことになりうることがあることを定義します。
これは**機器、装置などが損害を被る可能性があることの警告**にも使用しています。

注意は、**潜在的**に危険な状況で**回避しない場合には、**
軽傷または中程度の傷害を負うことになるおそれがあることを定義します。
それは**人体，機器，および情報の損傷を被る可能性のある行動に対する注意**にも使用しています。

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、日立または日立特約店保守担当にお申し付けください。

2．エミュレータまたはユーザシステムのパワーオン時、全てのケーブル類の抜き差しを行なわないでください。抜き差しを行った場合、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の破壊の可能性があります。

3．エミュレータまたはユーザシステムのパワーオン時、エミュレータとユーザシステムインタフェースケーブルおよびユーザシステムインタフェースケーブルとユーザシステム上の IC ソケットの抜き差しを行わないでください。
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の破壊の可能性があります。

4．ユーザシステムインタフェースケーブルとユーザシステム上の IC ソケットはピン番号を確かめて正しく接続してください。
接続を誤るとエミュレータとユーザシステムの発煙、発火の可能性があります。

5．電源給電については電源仕様に従って供給してください。使用する電源ケーブルは製品に添付のものを使用してください。仕様以外の電源電圧を加えないでください。

まえがき

本書について

本書は、H8/3864シリーズ、H8/3887シリーズマイクロコンピュータ用のE6000エミュレータのセットアップと使用方法を説明します。本書はデバッグプラットフォームのマニュアルです。

「1　はじめに」では、E6000エミュレータの主なエミュレーション機能の概要と、E6000エミュレータの制御ソフトウェアである日立デバッギングインタフェース（以降、HDIと呼びます）の機能を簡単に紹介します。

「3　ハードウェア」は、E6000エミュレータとユーザシステムの接続方法、およびハードウェア詳細について記載します。

「4　チュートリアル」は、簡単なCプログラムのロードとデバッグの方法を示しながら、E6000エミュレータの主な特長を紹介します。チュートリアルプログラムはディスクで提供されます。したがって、チュートリアルプログラムを実行することによって、システムの動作を直接理解できます。

「5　H8/3864シリーズ、H8/3887シリーズ　E6000エミュレータ HDIの機能」は、本E6000エミュレータ専用のHDIの特長を記載します。

想定

本書は、読者にMS-DOSおよびWindowsプログラムの実行および使用の手順に関する知識があるものと想定して話を進めます。

関連マニュアル

・日立デバッギングインタフェースユーザーズマニュアル

・ユーザシステムインタフェースケーブル取扱い説明書

・PCインタフェースボード取扱い説明書

本アニュアルは動作環境をIBM PC上の英語版Microsoft Windows95として記述しています。

目 次

図 目 次

表 目 次

1　はじめに

E6000エミュレータは、日立MCUをサポートする高性能リアルタイムインサーキットエミュレータです。本E6000エミュレータは H8/3864シリーズ、H8/3887シリーズマイクロコントーラ用のプログラムの開発とデバッグができます。

E6000エミュレータは、ソフトウェア開発とデバッグのために単体で、あるいはユーザシステムのデバッグのためにユーザシステムインタフェースケーブルでユーザシステムに接続した状態で使用できます。

E6000エミュレータは、Windowsベースのインタフェースプログラムであるに動作します。HDIは、E6000エミュレータハードウェアを制御し、豊富なコマンドを提供します。

1.1　デバッグの特長

1.1.1　ブレークポイント

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

ハードウェアブレークポイント

イベント検出システムのイベントチャネルと範囲チャネルを使って、最大12箇所のブレークポイントが設定できます。ハードウェアブレークポイントに関しては、「1.2　イベント検出システム(CES)」を参照してください。

プログラムブレークポイント（PCブレークポイント）

最大256のプログラムブレークポイントが設定できます。プログラムブレークポイントは、ユーザ命令をBREAK命令で置き換えることによって設定されます。

1.1.2　トレース

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

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

トレースバッファは、すべてのバスサイクルあるいは選択されたサイクルだけを記憶するように制御されます。イベント検出システムを使って所望のトレース制御を選択します。詳細は、以下の「1.2　イベント検出システム」を参照してください。

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

1.1.3　実行時間測定

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

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

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

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

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

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

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

1.2.1　イベントチャネル

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

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

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

1.2.2　範囲チャネル

範囲チャネルは、以下の項目の組み合わせで定義できます。

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

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

1.2.3　ブレーク

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

1.2.4　イベント間実行時間測定

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

1.3 ハードウェアの特長

1.3.1 メモリ

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

エミュレーションメモリは、MCUアドレス空間に1バイト単位で割り付けできます。各メモリブロックは、［**Memory Mapping...**］コマンドを使って、ユーザシステム上のメモリに指定でき、それぞれの場合で、リードライトアクセス、リードオンリアクセス、またはアクセス禁止を指定できます。

エミュレーションメモリの各々のメモリタイプの定義を表1-1に示します。

表 1-1 メモリタイプの定義

メモリタイプ	説明
オンチップ	MCU内蔵メモリ
エミュレータ	エミュレーションボード上のメモリ

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

1.3.2 エミュレーションクロック

システムクロックとサブシステムクロックは表1-2に示すような周波数に設定できます。

表 1-1 エミュレーションクロック一覧

エミュレータ	ターゲットMCU	周波数設定値
HS3880EP160H	H8/3864シリーズ、 H8/3887シリーズ （システムクロック）	2MHz, 1MHz, 0.5MHz およびターゲットクロック÷2
HS3880EP160H	H8/3864シリーズ、 H8/3887シリーズ （サブシステムクロック）	32.768KHz, 38.4 KHz, 307.2 KHz, および ターゲットサブシステムクロック

1.3.3　外部プローブ

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

1.3.4　使用環境条件

表 1-1 E6000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	周囲ガス	腐食性ガスのないこと
4	DC入力電源	電圧：５V±５% 電流：Max. ５A
5	ユーザVcc (Uvcc)	電圧：１．８～５．０V

1.3.5　外形寸法と重量

表 1-1 外形寸法および重量

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

2 セットアップ

本章は、PCインタフェースボードを使用したE6000エミュレータのセットアップ方法、およびHDIと共に使うためのE6000エミュレータの準備方法を述べます。

本章は以下の方法について説明します。

- PCインタフェースボード（別売）のセットアップ
- E6000エミュレータのセットアップ
- HDIソフトウェアのインストールとそれを使ったシステムの正しい動作のチェック

2.1 パッケージ内容

E6000エミュレータには、以下の構成品が梱包されています。

- E6000エミュレータ
- AC電源アダプタ　5V 5A （ACケーブル付き）
- HDIインストールディスク（HS3880EPI60SF）
- テストプログラムディスク（HS3880EVI60SF）
- 外部プローブ
- E6000 H8/3864シリーズ、H8/3887シリーズ用エミュレータユーザーズマニュアル（本マニュアル）
- 日立デバッギングインタフェースユーザーズマニュアル

セットアップの前に、上記の構成品がすべてそろっていることを確認してください。

ホストコンピュータにはIBM PCまたは、IBM PC互換機が使用できます。Windows3.1またはWindows95も必要です。

2.2 PCインタフェースボードのセットアップ

E6000エミュレータは、PCインタフェースボード（HS6000EII01H）を使ってHDIと通信します。はじめに、PCインタフェースボードをPCに差し込む必要があります。

PCインタフェースボードはメモリマップドボードであり、差し込む前に、PCインタフェースボードが使うメモリ領域を確保しなければなりません。これによって、他のプログラムが不用意にPCインタフェースボードを使ってしまうことを防止できます。

PCインタフェースボードに割り当てたメモリ領域が、他のボードに割り当てた領域と重ならないようにします。もしも重なると、PCインタフェースボードとE6000エミュレータは正しく動作しません。出荷時には、PCインタフェースボードのメモリ領域はH'D0000からH'D3FFFに割り当ててあります。

2.2.1 Windows 95でのPCインタフェースボードのセットアップ

- Windows 95を起動します。
- [**My Computer**] アイコンをマウス右ボタンでクリックし、ポップアップメニューから [**Properties**] を選択します。

System Propertiesダイアログが表示されます。

- Device Managerパネルの [**Computer**] アイコンをダブルクリックし、Computer Propertiesダイアログを開きます。
- View Resourcesパネルの [**Memory**] をクリックし、メモリのリソースを表示します。

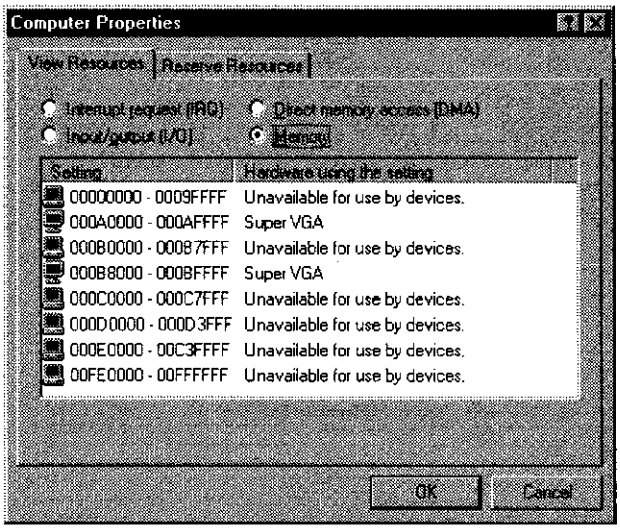


図 2-1 Computer Propertiesダイアログ(設定前)

ここにリストされていないメモリ領域が、PCインタフェースボード用に使用できます。下の表は、PCインタフェースボードのリアパネルのスイッチによって指定されるアドレスを示しています。これらのメモリ領域の中で、Computer Propertiesダイアログでリストされていないメモリ領域を選択してください。たとえば、H'D8000からH'DBFFFの領域を選択すると、対応するスイッチ番号は6になります。

表 2-1 PCインタフェースボードのメモリ領域

メモリ領域	スイッチ
H'C0000～H'C3FFF	0
H'C4000～H'C7FFF	1
H'C8000～H'CBFFF	2
H'CC000～H'CFFFF	3
H'D0000～H'D3FFF（出荷時の設定）	4
H'D4000～H'D7FFF	5
H'D8000～H'DBFFF	6
H'DC000～H'DFFFF	7
H'E0000～H'E3FFF	8
H'E4000～H'E7FFF	9
H'E8000～H'EBFFF	A
H'EC000～H'EFFFF	B

選択したメモリ領域をWindows 95が使用しないよう、以下の手順で登録します。

・ Reserve Resourcesパネルの［**Memory**］をクリックし、［**Add**］をクリックします。

Edit Resource Settingダイアログが表示されます。

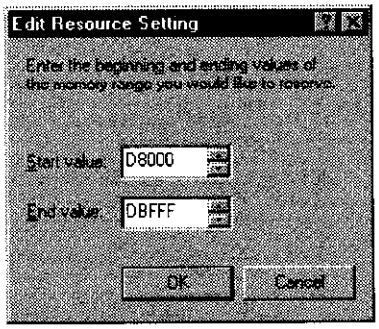


図 2-2 Edit Resource Settingダイアログ

・ 選択したメモリ領域の［**Start value**］［**End value**］を入力してください。

・ PCをリスタートせずシャットダウンし、電源スイッチを切ってください。

・ 小型のマイナスドライバを使って、PCインタフェースボードのリアパネルのスイッチを回し、選択したメモリ領域に対応するスイッチ番号を矢印が差すようにしてください。

・ PCのカバーを取り外し、未使用のISAバススロットにPCインタフェースボードを差し込んでください。

・ PCのカバーを取り付けてください。

・ PCインタフェースボードとE6000エミュレータの "PC IF" コネクタの間にPCインタフェースケーブルを接続してください。各プラグはカチッと音がするまでしっかりと差し込んでください。

・ PCの電源スイッチを入れてください。

・ Computer Propertiesダイアログで選択したメモリ領域が、System Reservedとリストされていることを確認してください。

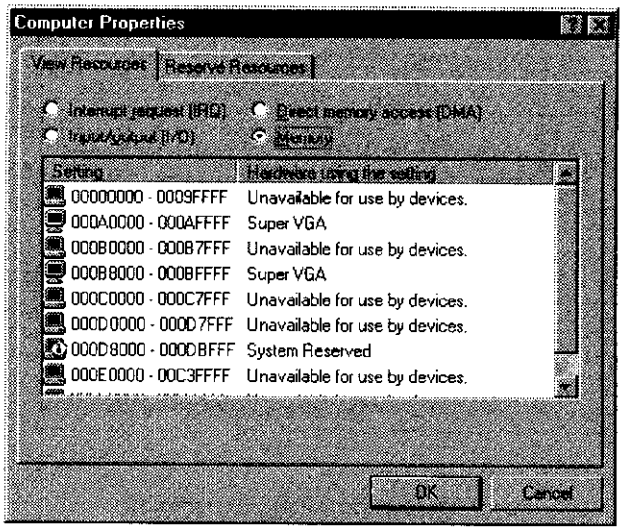


図 2-3 Computer Propertiesダイアログ(設定後)

2.2.2 Windows 3.1でのPCインタフェースボードのセットアップ

・ 「表 2-1 PCインタフェースボードのメモリ領域」を参照し、他のデバイスで使用されていないメモリ領域を選択してください。たとえば、H'D8000からH'DBFFFの領域を選択すると、対応するスイッチ番号は6になります。

・ PCの電源スイッチを切ってください。

・ 小型のマイナスドライバを使って、PCインタフェースボードのリアパネルのスイッチを回し、選択したメモリ領域に対応するスイッチ番号を矢印が差すようにしてください。

・ PCのカバーを取り外し、未使用のISAバススロットにPCインタフェースボードを差し込んでください。

・ PCのカバーを取り付けてください。

・ PCインタフェースボードとE6000エミュレータの "PC IF" コネクタの間にPCインタフェースケーブルを接続してください。各プラグはカチッと音がするまでしっかりと差し込んでください。

・ PCの電源スイッチを入れてください。

2.3 PCインタフェースボードの設定

次のステップは、PCインタフェースボードが使用するメモリ領域を、他のプログラムが使ってしまうことを防止します。

・ [**Start**] メニューから [**Run**] を選択してください。

・ "SYSEDIT" とタイプし、 [**OK**] をクリックしてください。

2.3.1 CONFIG.SYSの変更

CONFIG.SYSファイル中でEMM386.EXEを使用している場合は、以下の変更を行う必要があります。CONFIG.SYSファイルを使用していない場合、またはCONFIG.SYSファイルを使用していても、その中でEMM386.EXEを使用していない場合は、「2.3.2 SYSTEM.INIの変更」に進んでください。

・ CONFIG.SYSファイルの下記の行にラインカーソルを移動してください。

```
DEVICE=C:¥WINDOWS¥EMM386.EXE
```

・ この行を以下のように変更してください。

```
DEVICE=C:¥WINDOWS¥EMM386.EXE X=aaaa-bbbb
```

・ *aaaa* はStart value、*bbbb* はEnd valueのそれぞれ最下位を取った値です。たとえば、メモリ領域 H'D8000～H'DBFFF、スイッチが6に設定されていれば、この行を以下のように設定します。

```
DEVICE=C:¥WINDOWS¥EMM386.EXE X=D800-DBFF
```

・ CONFIG.SYSファイルをセーブしてください。

2.3.2 SYSTEM.INIの変更

- SYSTEM.INIファイル中にある〔386Enh〕セクションに以下の行を追加してください。

  EMMExclude=*aaaa-bbbb*

- *aaaa* はStart value、*bbbb* はEnd valueのそれぞれ最下位を取った値です。たとえば、メモリ領域 H'D8000～H'DBFFF、スイッチが6に設定されていれば、この行を以下のように設定します。

  EMMExclude=D800-DBFF

- SYSTEM.INIファイルをセーブし、SYSEDITを終了させてください。

- PCを再起動してください。

これによって、Windowsはこのメモリ領域を使いません。これでE6000エミュレータを接続し、HDIを実行してE6000エミュレータの通信状態をチェックする準備が整いました。

2.4 HDIのインストール

最初に以下のように、インストールディスクを使ってHDIをPC上にインストールしてください。

- Windowsを起動してください（まだ動作していない場合）。

- 起動中の他のアプリケーションをすべて閉じてください。

- HDIインストールディスク(#1/2)をPCのフロッピーディスクドライバに挿入してください。

- [**Start**] メニューから [**Run**] を選択してください。

- "a:setup" とタイプし、 [**OK**] をクリックしてください。

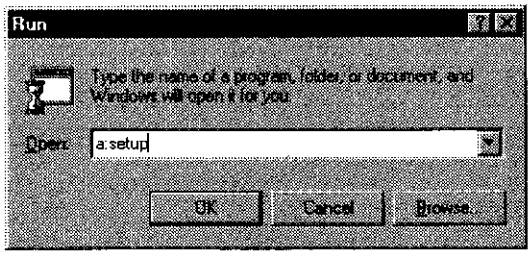


図 2-1 HDIインストールディスクの選択画面

HDIインストーラが動作し、以下の［**Welcome!**］が表示されます。

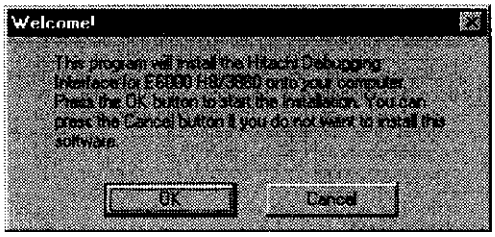


図 2-2 HDIインストーラ起動画面

・ ［**OK**］をクリックし、インストールを続行してください。

以下のダイアログボックスが、インストールしているHDIのバージョンの［**Read Me**］ファイルを表示します。

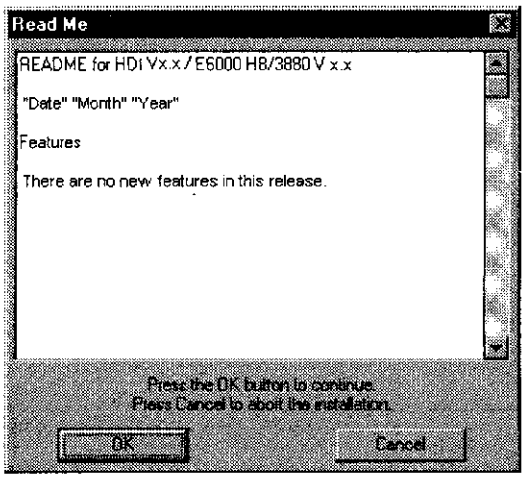


図 2-3 Read Meファイル画面

・ インストールに関する重要な情報として［**Read Me**］ファイルをチェックし、［**OK**］をクリックしてください。

以下のダイアログボックスによって、HDIをインストールするディレクトリを選択できます。

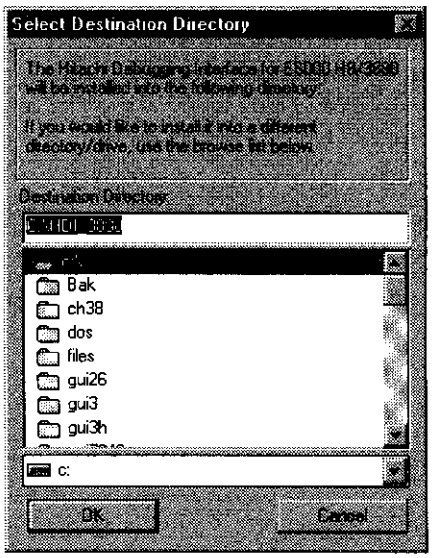


図 2-4 HDIインストールディスクの選択画面

・ [**OK**] をクリックしてデフォルトディレクトリC:¥HDI_3880にインストールするか、あるいは別のディレクトリを指定して [**OK**] をクリックしてください。C:¥HDI_3880以外を指定した場合は、tutorial.absがインストールされません。

以下のダイアログボックスが、インストールによって置き換わるファイルのバックアップを取るかどうかを尋ねます。

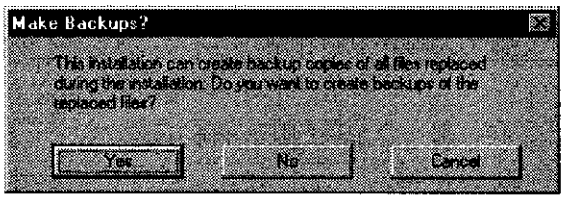


図 2-5 バックアップ指定画面

・ インストールによって置き換わるファイルを保存するには [**Yes**] をクリックしてください。バックアップを取る必要がなければ [**No**] をクリックしてください。

[Yes] を選択すると、以下のダイアログボックスによって、バックアップディレクトリを指定できます。

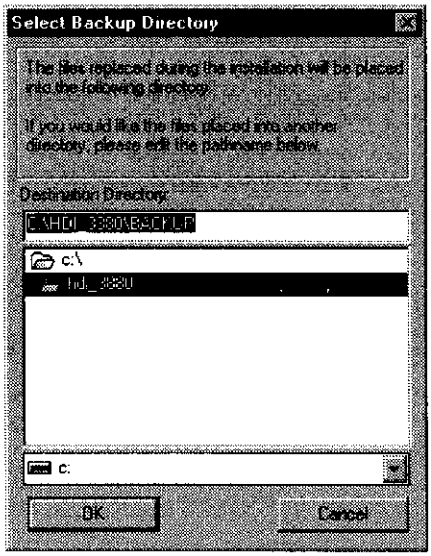


図 2-6 バックアップディレクトリの指定画面

・ 使いたいディレクトリを指定し、 [OK] をクリックしてください。

インストーラは、指定されたディレクトリにHDIファイルをコピーします。

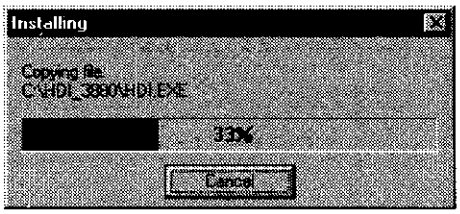


図 2-7 HDIインストール中の画面(1)

注：Errorメッセージボックスが表示された時は、インストールしようとしているランタイムライブラリなどがすでに使用中です。起動中の他のアプリケーションをすべて閉じていることを再度確認して、[**Retry**] をクリックしてください。それでもErrorメッセージボックスが表示される時は、[**Ignore**] をクリックしてインストールを続行してください。

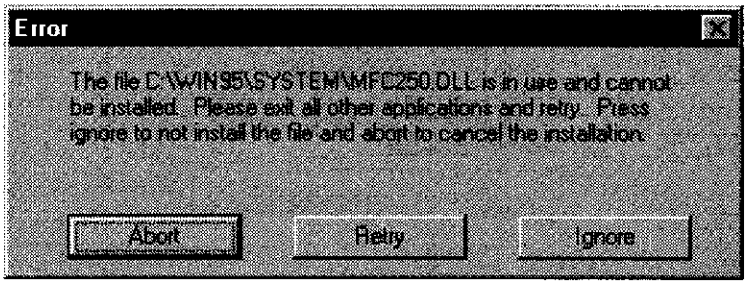


図 2-8 Errorメッセージボックス

1枚目のディスク (#1/2) のインストールが完了すると以下のメッセージが表示されますので、2枚目のディスク (#2/2) をフロッピーディスクドライバに挿入し [**OK**] をクリックしてください。

図 2-9 HDIインストール中の画面(2)

以下のダイアログボックスによって、HDIアイコンのスタートメニューに登録するグループを指定できます。

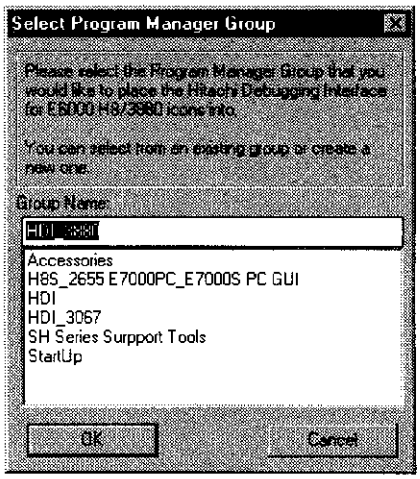


図 2-10 アイコンのプログラムグループの指定画面

・ 表示されているグループから選択するか、または新しいグループ名を入力し、[**OK**] をクリックしてください。

2.4.1 インストールの詳細

インストーラは、指定されたスタートメニューに以下のアイコンを生成します（デフォルト：HDI_3880）。

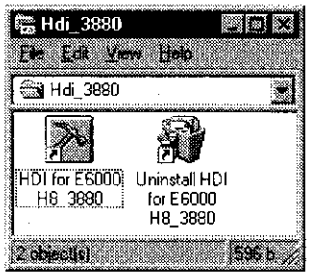


図 2-1 HDIのプログラムグループ

これらのアイコンは以下の機能を備えています。

[**HDI for E6000 H8_3880**] はHDIプログラムです。

[**Uninstall HDI for E6000 H8_3880**] は、HDIのアンインストール時に、HDIとその関連ファイルを削除するのに使います。

2.5 システムのチェック

次に、HDIを実行し、E6000エミュレータが正しく動作することをチェックします。

- E6000エミュレータの電源スイッチを入れ、パワーLEDランプが赤く点灯することを確認してください。
- E6000エミュレータにユーザシステムを接続している場合は、ユーザシステムの電源を必ずオンにしてください。また、ユーザシステムを接続していない場合は、E6000エミュレータからユーザシステムインタフェースケーブルを取り外してください。
- [**HDI**] アイコンをダブルクリックしてください。

全てが正しく設定されると、HDIが表示され、以下のメッセージのシーケンスがウインドウの下にステータスバーで現われます。

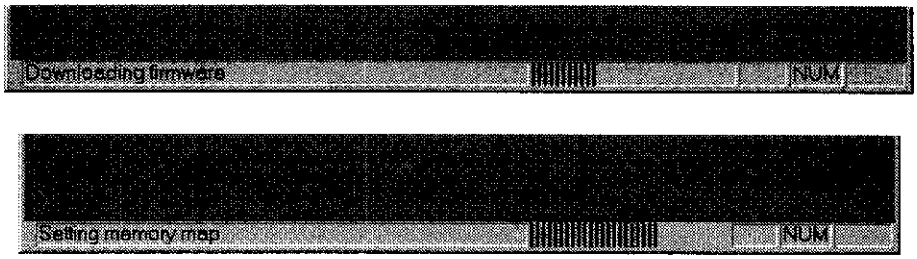


図 2-1 HDI起動中のステータスバー表示

最後に、全てが正しく設定されたことを示すために、ステータスバーが"Link up"を表示し、HDIのウインドウが以下のように表示されます。

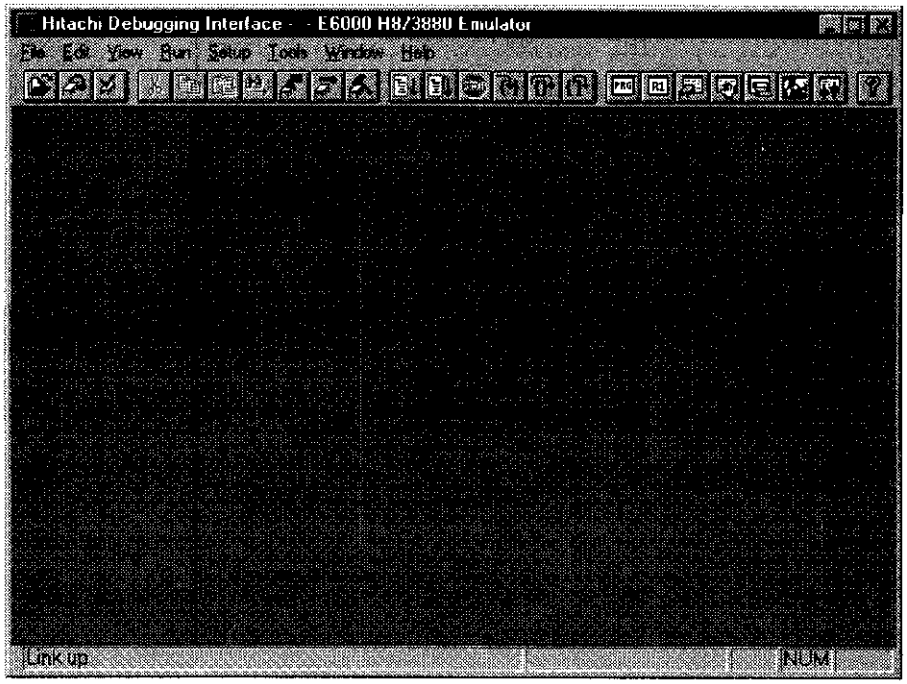


図 2-2 HDIの起動画面

2.6 さてつぎは？

これでE6000エミュレータは正しくセットアップできました。E6000エミュレータの主な機能に慣れるために、「4. チュートリアル」の章にしたがって操作してください。そしてMCUのプログラムの開発とデバッグを行うためにE6000エミュレータの使い方を覚えてください。

2.7 トラブルシューティング

2.7.1 接続不良

イニシャライズ中に以下のメッセージボックスが現われた場合、PCインタフェースボードはE6000エミュレータを認識できていません。

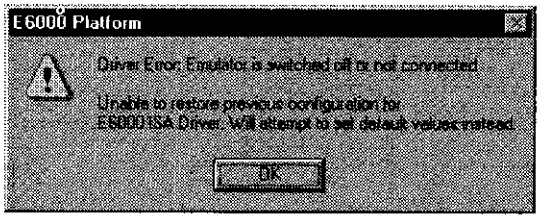


図 2-1 エラーメッセージ（1）

考えられる原因としては以下のようなものがあります。

- 添付のAC電源アダプタがE6000エミュレータに接続されていないか、またはE6000エミュレータの電源スイッチが入っていません。E6000エミュレータのパワーLEDを確認してください。

- PCインタフェースケーブルが、PCインタフェースボードとE6000エミュレータの間で正しく接続されていません。

2.7.2 通信不良

以下のメッセージが表示されると、HDIがE6000エミュレータを正しくセットアップできていません。

図 2-1 エラーメッセージ（2）

考えられる原因としては以下のようなものがあります。

- CONFIG.SYSファイルに確保されたメモリ領域と、PCインタフェースボード上のリヤパネルのスイッチ設定が異なっています。
- 選択されたメモリ領域が別のアプリケーションで使われています。
- 「2.2 PCインタフェースボードのセットアップ」「2.3 PCインタフェースボードの設定」に従って設定を見直してください。

3 ハードウェア

本章は、E6000エミュレータをユーザシステムに接続する方法を説明します。

3.1 ユーザシステムへの接続

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

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

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

以下に、E6000エミュレータのコネクタを示します。

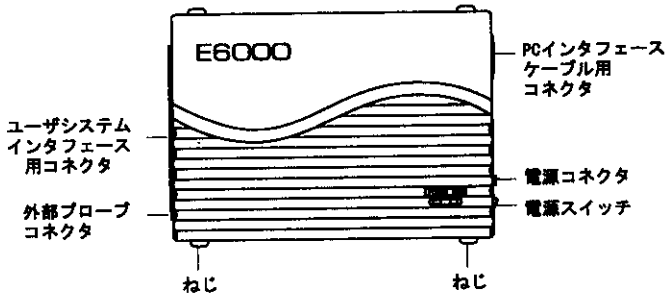


図 3-1 E6000 コネクタの位置

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

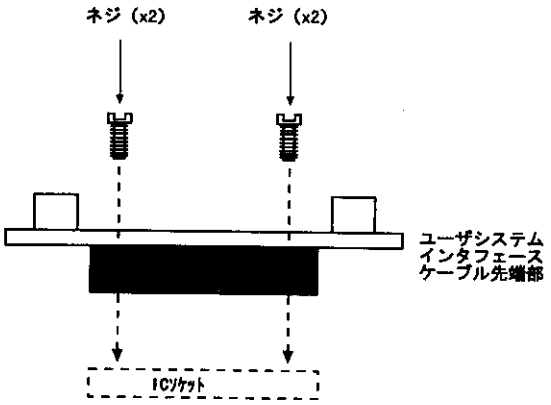


図 3-1 ユーザシステムインタフェースケーブルの接続

・ 接続前に、必ず、E6000エミュレータ、ユーザシステムの電源を切ってください。

・ ユーザシステムインタフェースケーブル先端部をユーザシステム上のソケットに挿入してください。

注：QFPパッケージによっては、ユーザシステムインタフェースケーブル先端部の向きにかかわらず、ソケットに差し込むことができるものがあります。挿入の際には、E6000エミュレータ側とソケットの１ピンの位置を必ず一致させてください。

・ 付属のねじを使って、ユーザシステムインタフェースケーブルにユーザシステムインタフェースケーブル先端部とソケットをねじ留めしてください。以下に示す順番で、ねじを徐々に締め付けてください。

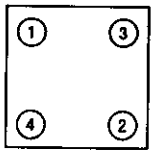


図 3-2 ネジの締め付け順序

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

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

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

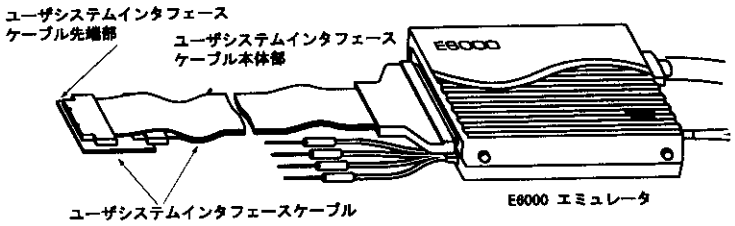


図 3-1 ユーザシステムインタフェースケーブル外観図

3.1.3 ユーザシステムインタフェースケーブル本体部と先端部の接続

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

3.2 電源供給

3.2.1 AC電源アダプタ

E6000エミュレータに付属のAC電源アダプタを常に使用してください。

3.2.2 極性

以下に電源プラグの極性を示します。

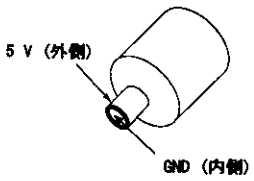


図 3-1 電源プラグ

3.2.3 電源モニタ回路

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

注：必ずE6000エミュレータに付属のAC電源アダプタを使用してください。

3.3 ハードウェアインタフェース

E6000エミュレータのユーザシステムインタフェース信号は、バッファなしに直接エミュレータ上のエバチップに接続されています。

3.3.1 信号保護

ユーザシステムインタフェース信号は、ダイオードによって、過大／過小電圧から保護されています。ただし、AVccとアナログポートには、この保護回路はありません。

アナログポート以外のポートには、プルアップ抵抗が接続されています。

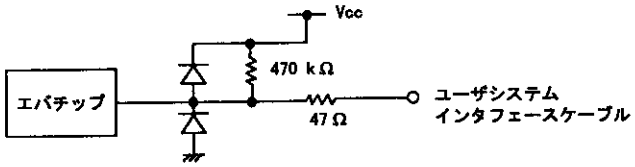
なお、E6000はユーザシステムインタフェースケーブル先端部の信号を監視することにより、ユーザシステムが接続されているかどうかを判断しています。

3.3.2 ユーザシステムインタフェース回路

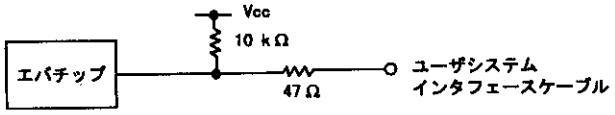
E6000エミュレータのユーザインタフェースには、プルアップ抵抗が入っており信号の遅れが生じます。また、プルアップ抵抗により信号がハイインピーダンス状態でもハイレベルになります。このことを考慮してユーザシステムのハードウェアを調整してください。また、ユーザシステムインタフェースケーブルによる信号の遅れは約3 nsです。

以下にユーザインタフェース信号回路を示します。

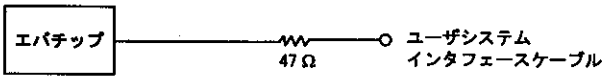
(1) 以下に記述のない信号



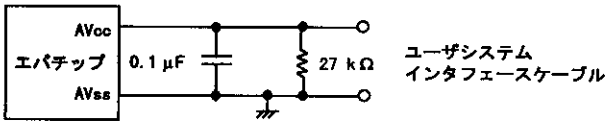
(2) OSC1, X1



(3) PB0/AN0～PB7/AN7, PC0～PC3

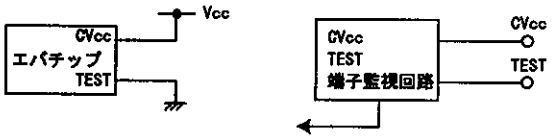


(4) AVcc, AVss



(5) CVcc, TEST

CVccがGNDに接続されている場合、またはTESTがVccレベルに接続されている場合は、HDI起動時に注意のメッセージが出力されます。ユーザシステム上の本端子をもう一度見直してください。



3.3.3 クロック発振器

サブクロックにユーザシステムからの外部クロックを使用する場合、エミュレータのCLOCKコマンドでターゲットクロック（sub t）を選択します。ユーザシステムからエミュレータに外部クロックを供給します。外部サブクロックをユーザシステムから入力する場合は、X1端子から図3-6に示す仕様にて入力してください。外部サブクロックとして水晶振動子を接続し、ユーザケーブルの発振回路でクロック発振させることはできません。

以下にユーザシステムインタフェースケーブル上のシステムクロック発振回路およびサブクロック入力仕様を示します。

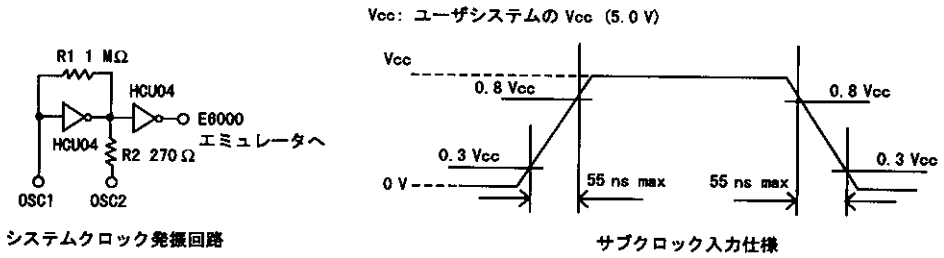


図 3-1 システムクロック発振回路およびサブクロック入力仕様

3.3.4 外部プローブ／トリガ出力

E6000エミュレータ筐体側面にあるEXTのマークが記された８ピンコネクタ（ユーザインタフェースコネクタの横）にE6000エミュレータ付属の外部プローブを接続してください。
外部プローブは入力４本とトリガ出力2本を備えています。
以下にこのコネクタのピン配置を示します。

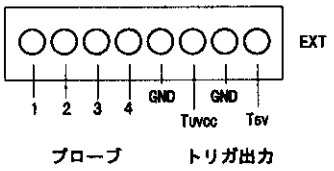


図 3-1 外部プローブコネクタ

以下に外部プローブのインタフェース回路を示します。

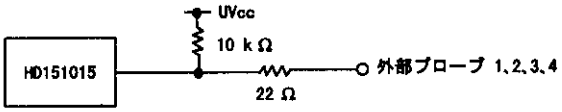


図 3-2 外部プローブインタフェース回路

トリガ出力はイベントチャネル8によって出力されるアクティブロー信号です。トリガ出力はT5V(2.5V～5.0Vの範囲でユーザシステムの電圧レベルに依存しません。)またはTUvcc（ユーザシステム電源電圧）レベルの２つあります。
なお、トリガ出力TUvccについては、ユーザシステム電源電圧1.8Vにての評価は行なえません（2.0V～5.0Vの範囲としてください)。

3.3.5 電源フォロワ回路

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

E6000エミュレータにユーザシステムインタフェースケーブルが接続されていないと、E6000エミュレータ上のMCUは5Vで動作し、ユーザシステムインタフェースケーブルが接続されている場合は、ユーザシステムの電源電圧と同レベルの電圧で動作します。ユーザシステムVccがMCUの動作電圧よりも低い場合であっても、E6000エミュレータは供給電圧をUVccに一致させます。したがって5V以下で動作している場合は、エミュレーションクロックの周波数が各Vccにおける最高動作周波数を超えないように注意してください。

E6000エミュレータコンフィグレーションダイアログボックスを使って、[**User VCC Threshold**]を5Vから0Vの範囲で設定できます。ユーザVccがその値よりも下がった場合、システムステータスウインドウの[**User System Status**]には[**Down**]が表示されます。User Vcc Threshold電源レベルよりも高い場合は[**OK**]が表示されます。

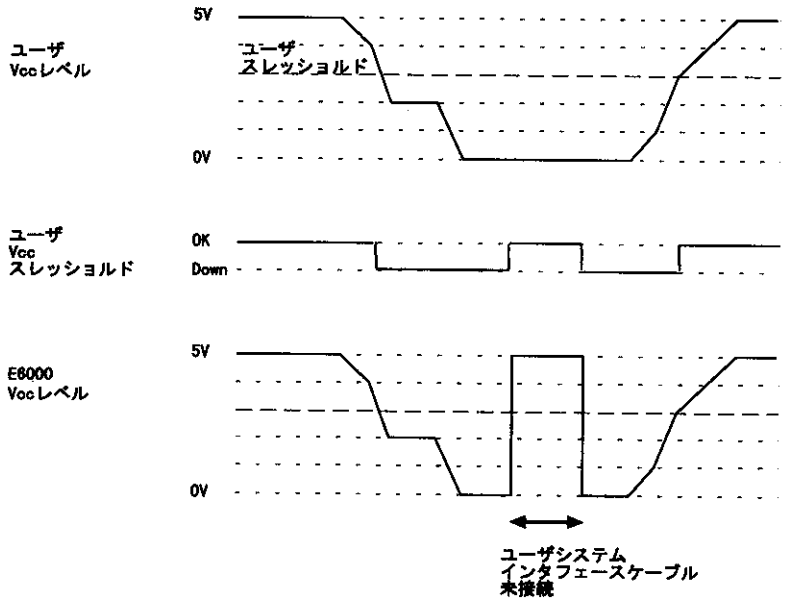


図 3-1 ユーザシステムとE6000とのVccの関係

3.4 MCUとE6000エミュレータの相違点

E6000エミュレータの電源投入後およびコマンドリセット後の、MCUとE6000エミュレータのレジスタの初期値の相違を以下に示します。

表 3-1 MCUとE6000エミュレータの相違

状態	レジスタ	E6000エミュレータ	MCU
電源投入後	PC	不定	リセットベクタ値
	R0 to R6	0000	不定
	R7 (SP)	0010	不定
	CCR	Iマスクは1	Iマスクは1
		その他は不定	その他は不定
リセットコマンド後	PC	リセットベクタ値	リセットベクタ値
	R0 to R6	不定	不定
	R7 (SP)	0010	不定
	CCR	Iマスクは1	Iマスクは1
		その他は不定	その他は不定
		B'1	B'1

E6000エミュレータのI/Oポート上の保護回路の詳細については、「3.3 ハードウェアインタフェース」を参照してください。

3.4.1 A／Dコンバータ

ユーザシステムインタフェースケーブルで接続されているため、A/D変換の精度は、MCUのハードウェアマニュアルに記載の精度より劣下します。

3.4.2 未使用領域のアクセス

未使用領域H'FF80～H'FF8FはエミュレータシステムでMAP設定でエミュレータに割り付けた場合、正常動作できませんので、本領域は使用しないでください。

3.4.3 Go Resetコマンドによるプログラム実行

Go Resetコマンドを用いてプログラムを実行する際、E6000は約500μsのリセット信号をエバチップに入力します。実行時間測定結果には、このリセット信号入力時間が加算されます。

4 チュートリアル

本章では、E6000エミュレータの主な特長をHDIの操作例に従って説明します。
チュートリアルでは、E6000エミュレータ上のエミュレーションメモリを使って実行しますので、E6000エミュレータをユーザシステムに接続する必要はありません。

4.1 はじめに

このチュートリアルは、簡単なCプログラムで作成されています。

本章を読む前に、

- 「2 セットアップ」に従って、E6000エミュレータをHDIで起動してください。このチュートリアルを使うためにE6000エミュレータをユーザシステムに接続する必要はありません。

- MCUのアーキテクチャと命令セットについてよく理解してください。詳しくは、H8/3864シリーズ、H8/3887シリーズのハードウェアマニュアルを参照してください。

4.1.1 概要

チュートリアルは、NAME（アルファベット順）、AGE（昇順）、ID（昇順）を並び換えるプログラムです。ソースプログラム（tutorial.C）およびSysrofフォーマットのオブジェクトファイル（tutorial.abs）は、HDIのインストールディスク中に用意されています。

4.2 チュートリアルプログラムの動作

プログラムの最初の部分は、includeするファイルの宣言です。

```
#include <machine.h>
#include "¥CH38¥INCLUDE¥string.h"
```

次の部分は、プログラムの中で使われる定数、構造体および関数の初期値の定義です。

```
#define NAME    (short)0
#define AGE     (short)1
#define ID      (short)2
#define LENGTH  8

struct namelist {
        char    name[LENGTH];
        short   age;
        long    idcode;
};
struct namelist section1[] = {
        "Naoko",  17, 1234,
        "Midori", 22, 8888,
        "Rie",    19, 7777,
        "Eri",    20, 9999,
        "Kyoko",  26, 3333,
        "",       0,    0
};

int count;

void sort();
```

main関数は次のとおりです。

```
main( )
{
        count = 0;
        for ( ; ; ){
                sort(section1, NAME);
                count++;
                sort(section1, AGE);
                count++;
                sort(section1, ID);
                count++;
        }
}
```

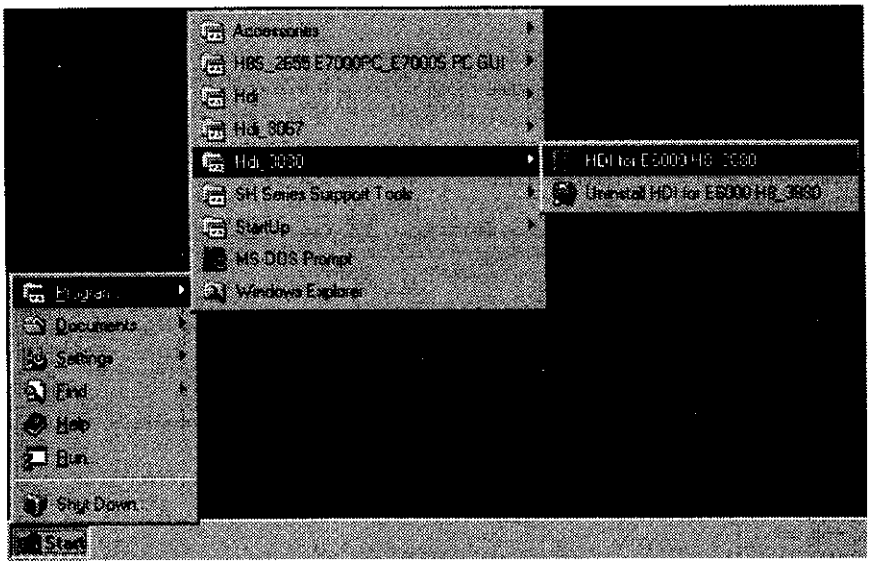
残りの部分はmain関数から呼び出される関数です。

```
void sort(list, key)
struct namelist list[];
short key;
{
      short i,j,k;
      long min;
      char *name;
      struct namelist worklist;

      switch(key){
            case NAME :
                  for (i = 0 ; *list[i].name != 0 ; i++){
                        name = list[i].name;
                        k = i;
                        for (j = i+1 ; *list[j].name != 0 ; j++){
                              if (strcmp(list[j].name , name) < 0){
                                    name = list[j].name;
                                    k = j;
                              }
                        }
                    worklist = list[i];
                        list[i] = list[k];
                        list[k] = worklist;
                  }
                  break;



            case AGE  :
                  for (i = 0 ; list[i].age != 0 ; i++){
                        min = list[i].age;
                        k = i;
                        for (j = i+1 ; list[j].age != 0 ; j++){
                              if (list[j].age < min){
                                    min = list[j].age;
                                    k = j;
                              }
                        }
                        worklist = list[i];
                        list[i] = list[k];
                        list[k] = worklist;
                  }
                  break;



      case ID   :
```

```
            for (i = 0 ; list[i].idcode != 0 ; i++){
                min = list[i].idcode;
                k = i;
                for (j = i+1 ; list[j].idcode != 0 ; j++){
                    if (list[j].idcode < min){
                        min = list[j].idcode;
                        k = j;
                    }
                }
                worklist = list[i];
                list[i] = list[k];
                list[k] = worklist;
            }
            break;
        }

    }
```

4.3 HDIの実行

HDIを実行するには、Start/Programs/Hdi_3880/［**HDI for E6000 H8_3880**］をクリックしてください。



4.3.1 ターゲットプラットフォームの選択

HDIは複数のターゲットプラットフォームをサポートする拡張機能があります。複数のプラットフォーム用にシステムがセットアップされると、まず現在のセッションのプラットフォームを選択する必要があります。

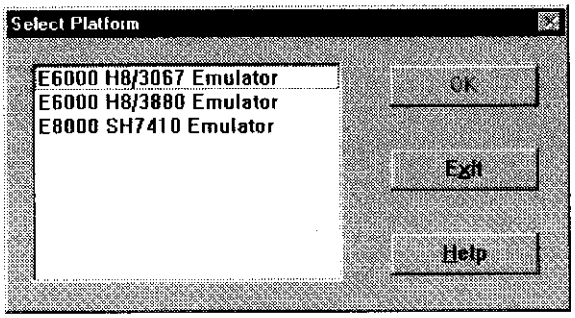


図 4-1 プラットホームの選択

- このチュートリアルでは、E6000 H8/3880 Emulatorを選択し、［**OK**］をクリックしてください。

[**Setup**] メニューから [**Select Platform...**] を選択すれば、いつでもターゲットプラットフォームを変更できます。ただし、プラットフォームが１つしかインストールされていなければ、このメニューオプションは使用できません。

E6000エミュレータが正しくセットアップされていれば、ステータスバーの [**Link up**] メッセージと共に、HDIウインドウが表示されます。以下にウインドウの主な機能を示します。

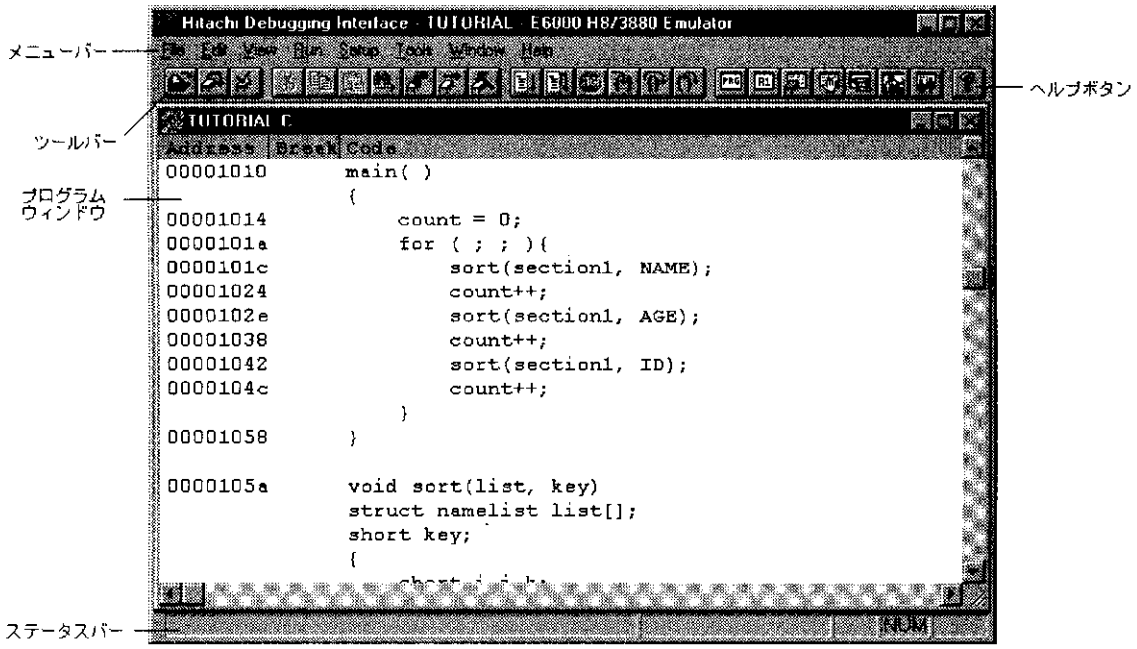


図 4-2 HDIウインドウ画面

HDIの主な機能については次の章で述べます。

4.3.2　メニュー

メニューバーには、E6000エミュレータの環境設定またはHDIのデバッグ機能を使用するためのコマンドがあります。（E6000エミュレータをセットアップしHDIを使うためのHDIコマンドへのアクセスを示します。）

ツールバー

よく使うメニューコマンドのショートカットとして便利なボタンです。

プログラムウインドウ

デバッグしているソースプログラムなどを表示します。

ステータスバー

E6000エミュレータの状態、例えばダウンロードの進捗状況や実行モードにおけるアドレスバスの状態を示します。

ヘルプボタン

HDIの使い方やコマンド構成についてのヘルプ画面を表示します。

4.4 E6000エミュレータのセットアップ

E6000エミュレータにプログラムをダウンロードする前に、E6000に対象MCU条件を設定しなければなりません。以下の項目を設定する必要があります。

- デバイスタイプ
- 動作モード
- 動作クロック
- ユーザ信号
- メモリマップ

以下に、チュートリアルプログラム用にE6000エミュレータを設定する方法について述べます。

4.4.1 プラットフォームの構成

- 選択したプラットフォームに固有の設定をするために、［**Setup**］メニューから［**Configure Platform...**］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。

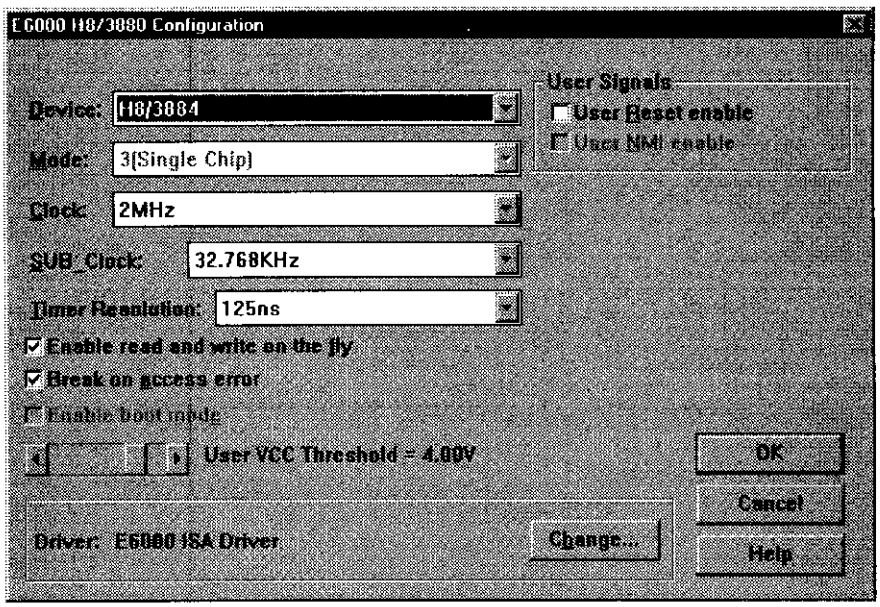


図 4-1 コンフィグレーションダイアログボックス

- オプションを以下のように設定してください。

表 4-1 コンフィグレーションオプションの設定例

オプション	設定値
エバチップ	HD64E3880
デバイス (Device)	H8/3864
モード (Mode)	3 (シングルチップモード)
動作クロック (Clock)	2MHz
タイマ分解能 (Timer Resolution)	125ns
ユーザシステムの電圧レベル (User Vcc Threshold)	4.00V
その他のオプション	イネーブル

・ [OK] をクリックしてターゲットコンフィグレーションを更新してください。

4.4.2 メモリマッピング

コンフィグレーションダイアログボックスでデバイスおよびモードを選択すると、HDIは自動的に選択したデバイスおよびモードに合わせたマップの割付を行います。

・ 現在のメモリマップを表示するには、[View] メニューから [Memory Mapping Window] を選択するか、またはツールバーの [Open Memory Mapping] ボタン をクリックしてください。

[Memory Mapping] ウインドウが以下のように表示されます。

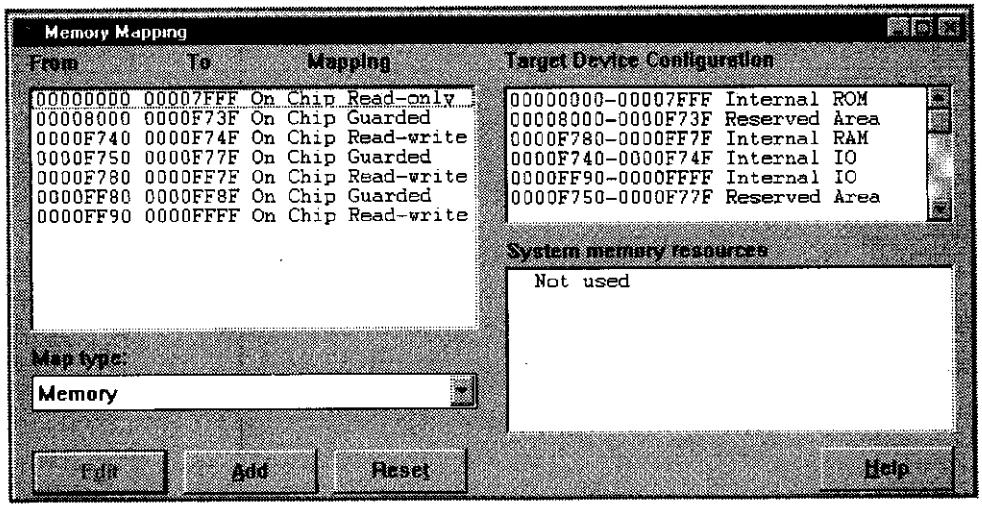


図 4-1 Memory Mapping ウインドウ

E6000エミュレータメモリは、以下のタイプがあります。

表 4-1 メモリタイプの定義

メモリタイプ	説明
オンチップ (On Chip)	MCU内蔵メモリをアクセスします。
エミュレータ (Emulator)	エミュレーションメモリをアクセスします。

また、アクセス制限については以下の３つのタイプがあります。

表 4-2 メモリタイプオプション

アクセスタイプ	説明
Read-Write	RAM
Read-Only	ROM
Guarded	アクセス不可

本チュートリアルでは、デフォルトマッピング使用します。以下のように割り付け状態を見ることもできます。

・ マップ設定を変更する場合は、対象の設定値を選択して [Edit] ボタンをクリックするか、または対象のマップ設定行をダブルクリックしてください。
([**Memory Mapping**] ウインドウのOn Chip Read-onlyの箇所をダブルクリックしてください。)

[**Edit Memory Mapping**] ダイアログボックスが表示されます。

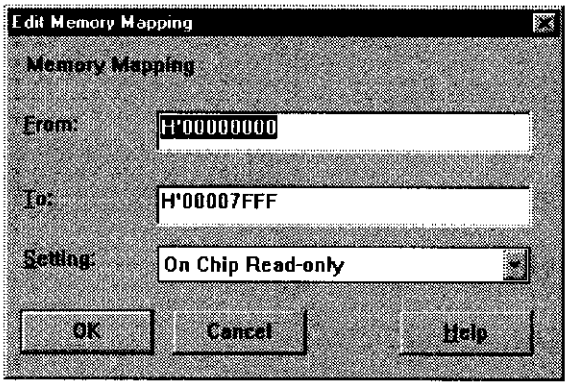


図 4-2 Edit Memory Mapping ダイアログボックス

・ [**OK**] をクリックして、ダイアログボックスを閉じてください。

4.5　チュートリアルプログラムのダウンロード

E6000エミュレータを上記のようにセットアップした後、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します。

4.5.1　　オブジェクトファイルのダウンロード

最初に、以下のようにSysrofフォーマットオブジェクトファイルをロードしてください。

- [**File**] メニューから [**Load Program**] を選択するか、またはツールバーの [**Load Program**]  ボタンをクリックしてください。

- hdi_3880¥tutorialディレクトリの中のtutorial.absファイルを選択し、[**OK**] をクリックしてください。

- tutorial.absファイルはHDIのインストール時にHDIをC:¥HDI_3880ディレクトリに格納したときのみ生成されます。それ以外のディレクトリに格納した場合は、tutorial.batファイルを実行するとtutorial.absファイルが生成されます。ご使用の環境に合わせて、tutorial.bat、turorial.subを修正する必要のある場合があります。

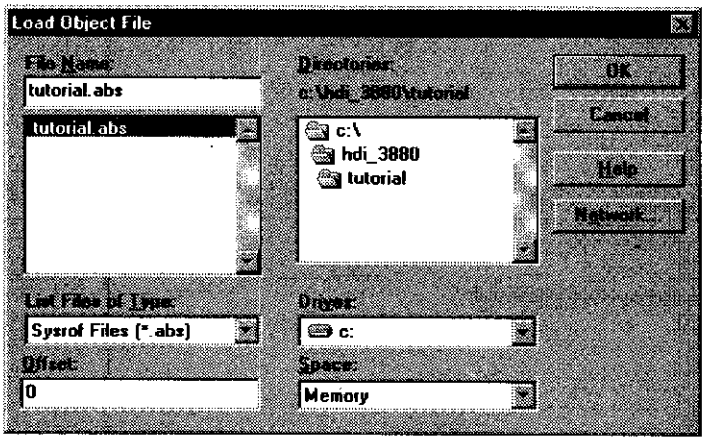


図 4-1 Load Program 画面

ファイルがロードされると、以下のダイアログボックスが、プログラムコードが書き込まれたメモリエリアに関する情報を表示します。

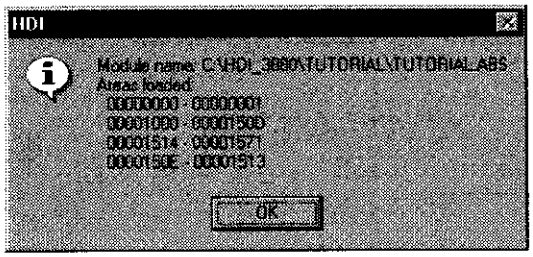


図 4-2 プログラムロードダイアログボックス

・ [OK] をクリックしてください。

プログラムは内蔵ROM領域にロードされます。

4.5.2 プログラムリストの表示

HDIでは、プログラムリストをソースコードやアセンブラニーモニックで表示することができます。

・ [View] メニューから [Program Window...] を選択するか、またはツールバーの [Program Window] ボタン  をクリックしてください。

ロードしたオブジェクトファイルに対応するCソースファイルを選ぶ必要があります。

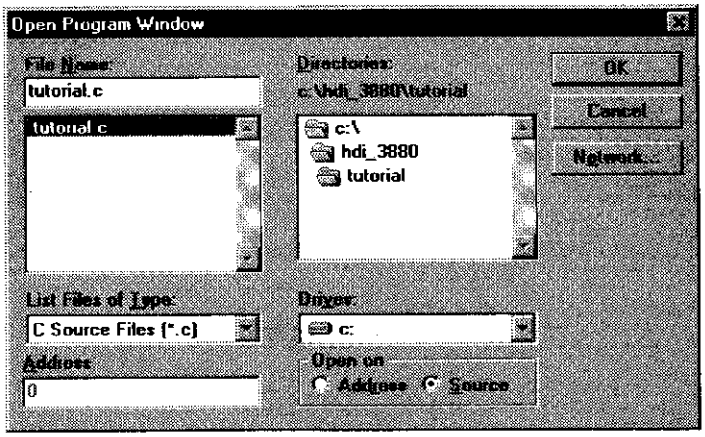


図 4-1 Program Window 画面

・ [tutorial.c] を選択し、[OK] をクリックしてプログラムウインドウを表示してください。

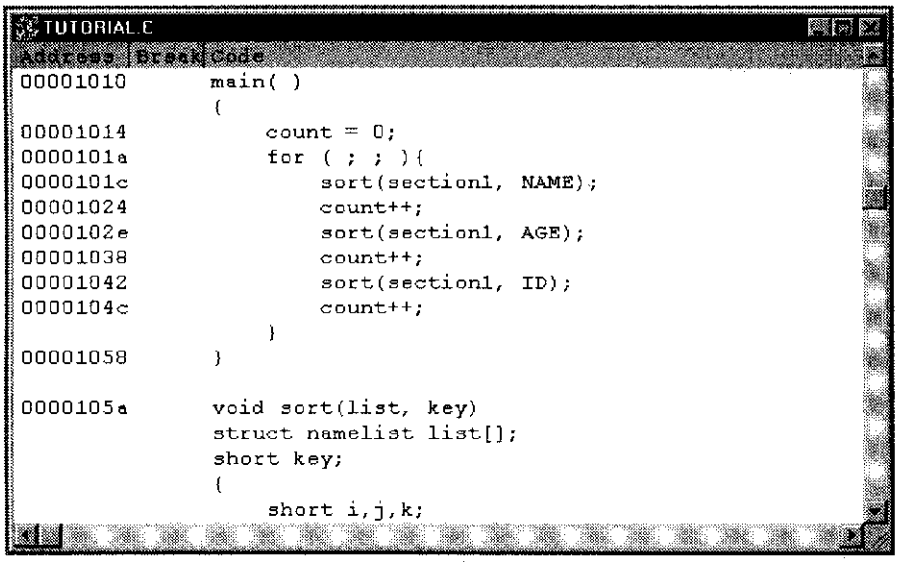


図 4-2 ソースプログラム画面

・ 必要ならば、[**Setup**] メニューの [**Customise**] サブメニューから [**Font**] オプションを選択し、コンピュータに合ったフォントとサイズを選択してください。

プログラムウインドウを、最初に開いたときはメイン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プログラムをスクロールし、定義文等を見ることができます。

4.6 ブレークポイントの使い方

最も簡単なデバッグ機能のひとつにプログラムの特定の箇所に達したときに実行を停止できるプログラムブレークポイントがあります。この機能を使えばプログラムが停止した時のMCUやメモリの状態を調べることができます。

4.6.1 プログラムブレークポイントの設定

プログラムウインドウによって、プログラムのあらゆるポイントにブレークポイントを簡単に設定できます。たとえば、以下のようにしてアドレス1038にブレークポイントを設定します。

・ 1038番地を含むラインの [**Break**] カラムをダブルクリックしてください。

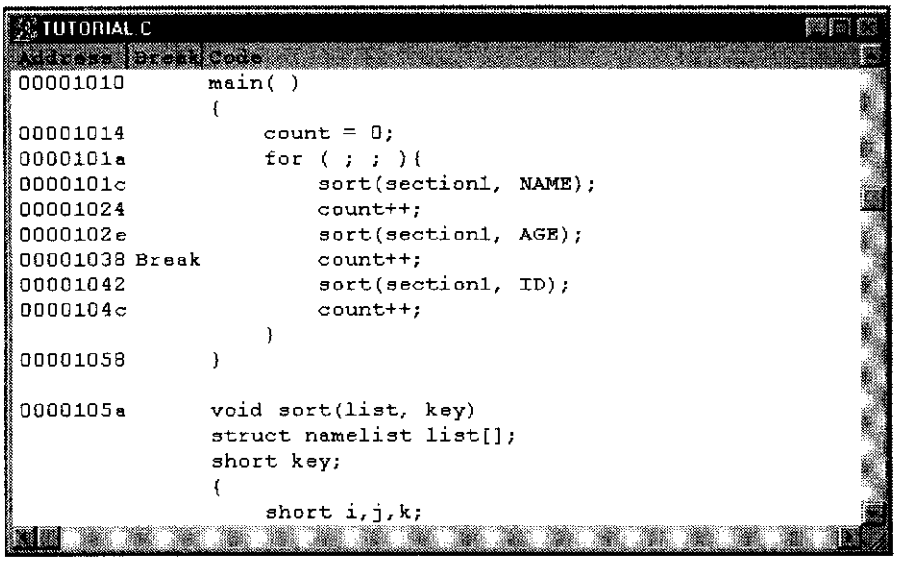


図 4-1 ブレークポイントの設定

その位置に“Break”が表示され、そのアドレスにプログラムブレークポイントが設定されたことを示します。また、本章では実行しませんが、さらにダブルクリックしていくことによりイベント間実行測定のイベント設定（“+Timer”で測定開始、“-Timer”で測定終了）、Point to Pointトレース制御の設定（“+Trace”でトレース開始、“-Trace”でトレース停止）およびトレースストップの設定（“TrStop”でトレースストップ）ができます。これらはダブルクリックすることにより、以下のような順序でサイクリックに設定できます。

“Blank” → “Break” → “+Timer” → “-Timer” → “+Trace” → “-Trace” → “TrStop” → “Blank” →…

4.6.2 プログラムの実行

リセットベクタで指定されているアドレスからプログラムを実行するには、

- [**Run**] メニューから [**Go Reset**] を選択するか、またはツールバーの [**Go Reset**] ボタン  をクリックしてください。

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し、プログラムが停止した位置を示すためにプログラムウインドウ中でステートメントが強調表示されます。

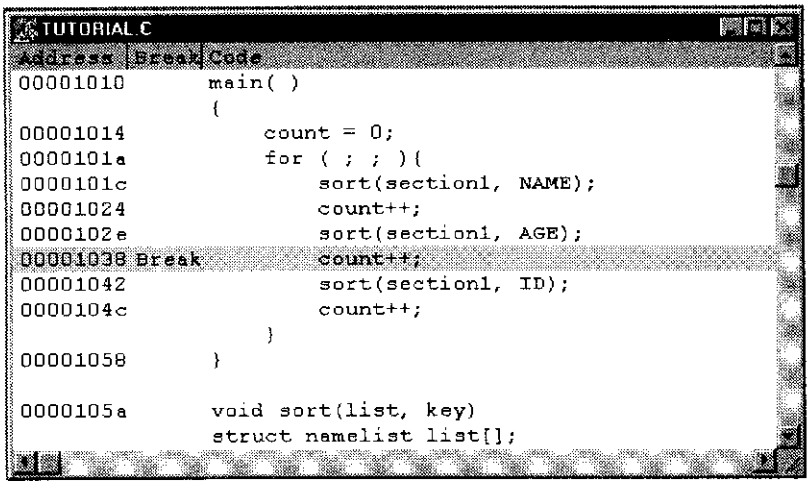


図 4-1 ステートメントの強調表示

[Break=Software Break] メッセージがステータスバーに表示され、ブレークの原因を示します。

[**System Status**] ウインドウで最後のブレークの原因が確認できます。

- [**View**] メニューから [**Status Window**] を選択するか、またはツールバーの [**Status Window**] ボタンをクリックしてください。

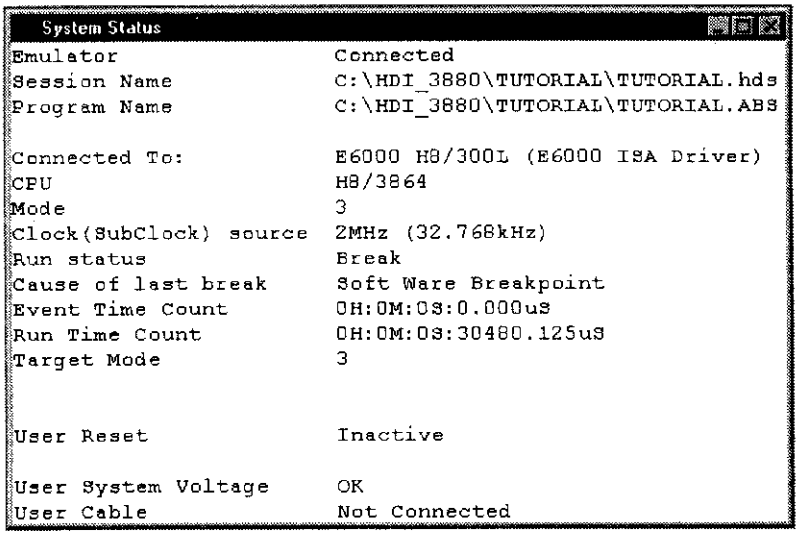


図 4-2 Status Window 画面

[Cause of last break]のラインは、ブレークの原因がプログラムブレークであることを示しています。

4.6.3　レジスタ内容の参照

プログラムが停止している間に、MCUレジスタの内容を参照できます。それらは [**Registers**] ウインドウに表示されます。

・ [**View**] メニューから [**Register Window**] を選択するか、またはツールバーの [**Register Window**] ボタン をクリックしてください。

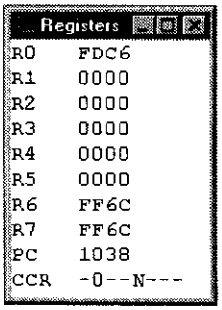


図 4-1 Register Window 画面

プログラムカウンタPCの値は強調表示されたステートメントH'1038になっています。
（注：その他のレジスタの値は上に示すものとは異なることがあります。）
レジスタの値は［**Registers**］ウインドウで変更できます。

・ PCの値を変えるには、［**Registers**］ウインドウで［**PC**］をダブルクリックしてください。

以下のダイアログボックスによって値を編集できます。

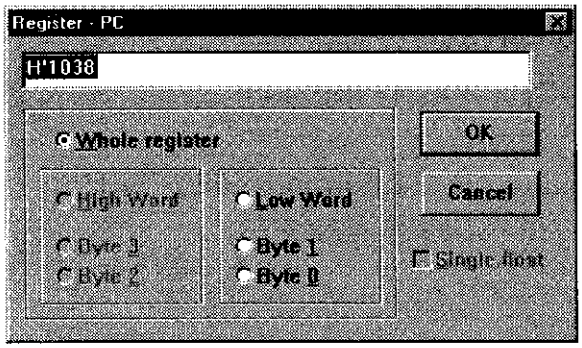


図 4-2 レジスタ値の編集

・ 値をH'102e（前のステートメントのアドレス）に変更し、［**OK**］をクリックしてください。

強調表示されたバーがプログラムウィンドウの前のステートメントに移動し、新しいプログラムカウンタの値を示します。

・ ［**Run**］メニューから［**Go**］を選択し、ブレークポイントまでの実行を再開してください。

4.6.4 ブレークポイントの確認

プログラムに設定した全てのブレークポイントの一覧をブレークポイントウインドウで見ることができます。

・ ［**View**］メニューから［**Breakpoint Window**］を選択するか、またはツールバーの［**Breakpoint Window**］ボタン  をクリックしてください。

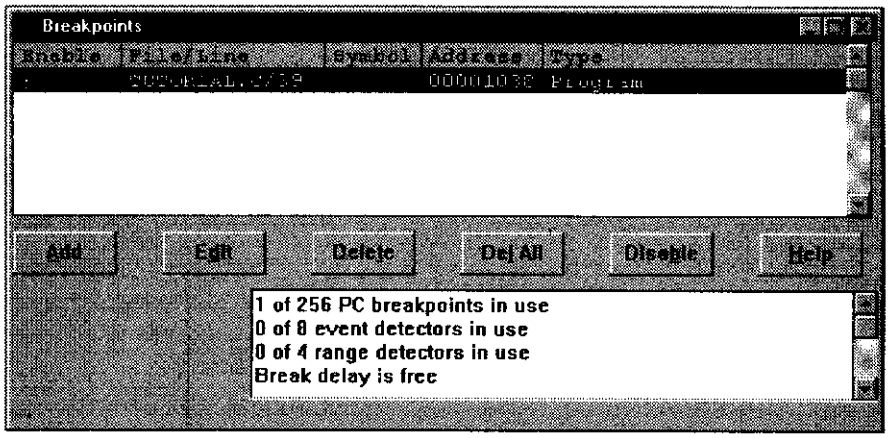


図 4-1 Breakpoint Window 画面

[**Breakpoints**] ウインドウによって、ブレークポイントの許可または禁止、新しいブレークポイントの設定、およびブレークポイントの削除ができます。

次へ進む前に、以下のようにブレークポイントを削除してください。

- [**Breakpoints**] ウインドウのブレークポイントを強調表示し、[**Delete**] をクリックしてください。
- [**Breakpoints**] ウインドウを閉じてください。

4.7 メモリと変数の表示

メモリ領域の内容を参照することにより、またはプログラム中で使われる変数の値を表示することによって、プログラムの動作をモニタできます。

4.7.1 メモリを表示する

メモリブロックの内容を [**Memory**] ウインドウで見ることができます。

たとえば、ASCIIで [section1] 列に対応したメモリを見る場合：

- [**View**] メニューから [**Memory Window...**] を選択するか、またはツールバーの [**Memory window**] ボタン をクリックしてください。

・ [**Address**] フィールドにsection1を入力し、[**Format**] をASCIIに設定してください。

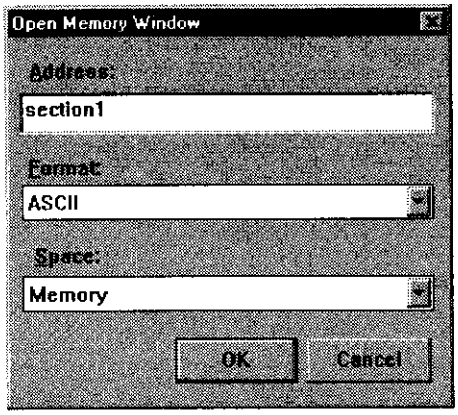


図 4-1 Memory Window の設定

・ [**OK**] をクリックして、指定されたメモリ領域を示す [**Memory**] ウインドウを開いてください。

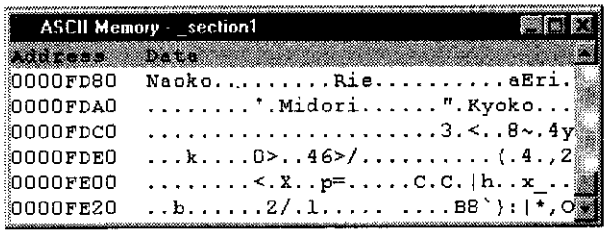


図 4-2 Memory Window 画面 (ASCII)

・ [section1] 列の内容をモニタできるように、[**Memory**] ウインドウを開いたままの状態にしてください。

4.7.2 変数を表示する

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を見ることができ、期待した様にそれらが変化することを確認できます。

たとえば以下の手順で、プログラムの始めに宣言した [struct] 変数 "section1" を見ることができます。

・ sort (section1, ID);のラインが見えるように、プログラムウィンドウをスクロールアップしてください。

・ プログラムウインドウのsection1の左にカーソルを置くようにクリックしてください。

・ 右のマウスボタンでプログラムウインドウをクリックし、ポップアップメニューを表示し、[**Add Watch**] を選択してください。

すると以下のウインドウが表示されます。

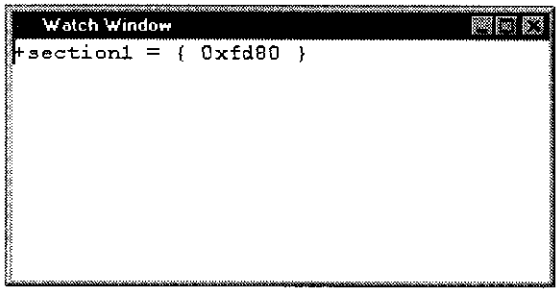


図 4-1 Watch 画面

また、名前を指定することによって、［**Watch**］ウインドウに変数を加えることができます。この方法を使って、以下のように、変数countを加えてください。

・ ［**Watch**］ウインドウで右のマウスボタンを押し、ポップアップメニューから［**Add Watch...**］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。

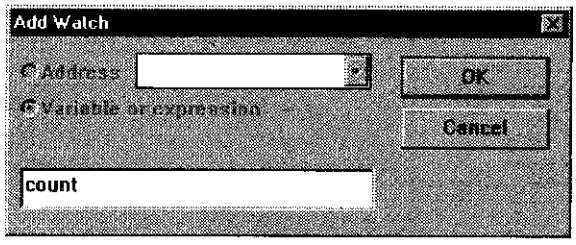


図 4-2 Add Watch ダイアログボックス

・ 変数“count”をタイプし、［**OK**］をクリックしてください。

［**Watch**］ウインドウは、定数文字列“count”を表示します。

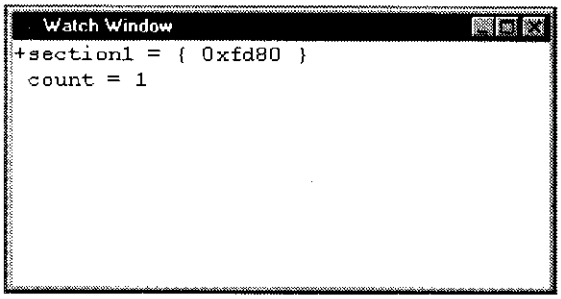


図 4-3 Watch 画面（変数追加後）

［**Watch**］ウインドウのシンボルの左の＋をダルブクリックし、シンボルを拡張して各要素を配列して見ることができます。

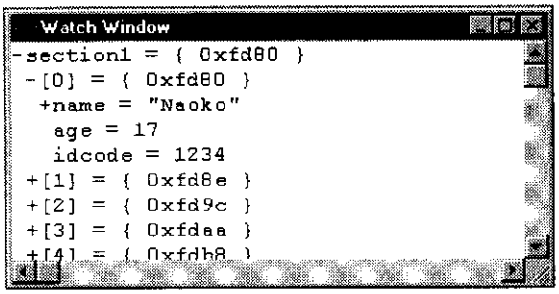


図 4-4 Watch 画面（シンボル拡張）

4.8 プログラムのステップ実行

E6000エミュレータは、プログラムのシングルステップにおけるオプションを備えており、命令やステートメントを一度に実行します。表4.4に示すようなステップオプションがあります。

表 4-1 プログラムステップオプション

コマンド	説明
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）。
Step Over	呼び出された関数の全ステートメントを実行します。
Step out	関数を抜け出し、関数を呼び出したプログラムにおける次のステートメントで停止します。
Step...	指定したステートメント数ステップ実行します。

4.8.1 シングルステップ

- PC=1038にブレークポイントを設定してください。
- さらに [**Step In**] コマンドによって、sortファンクションコールまでプログラムを実行してください。sort(section1, ID)のステートメントが強調表示されます。

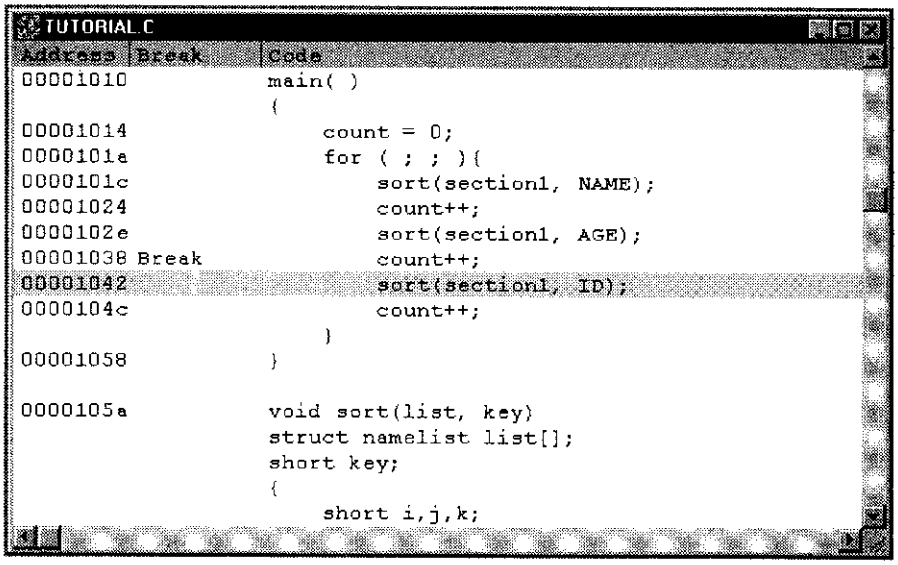


図 4-1 Step In 実行後のプログラムウインドウの表示（1）

- sort中のステートメントをステップ実行するために [**Run**] メニューから [**Step In**] を選択するか、またはツールバーの [**Step In**] ボタン をクリックしてください。

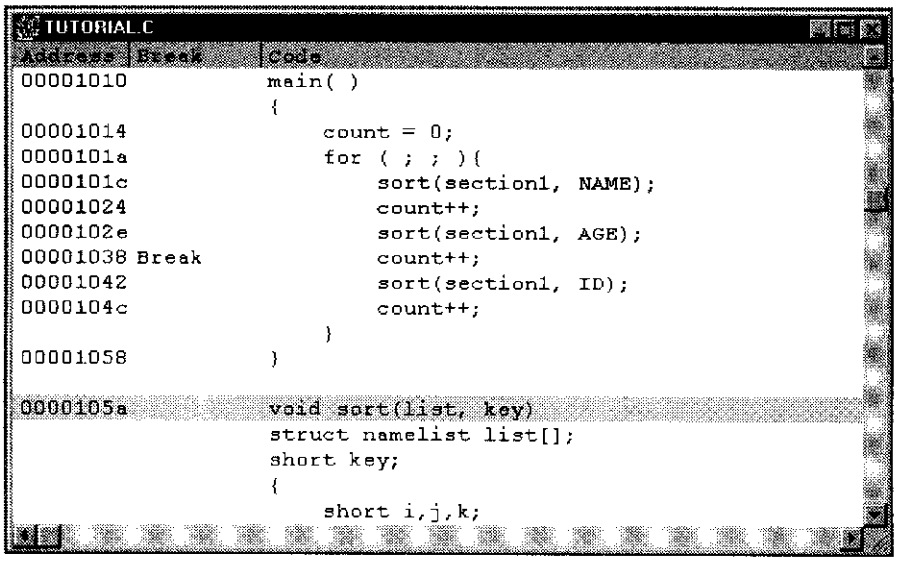


図 4-2 Step In 実行後のプログラムウインドウの表示（2）

・ [**Run**] メニューから [**Step Out**] を選択するか、またはツールバーの [**Step Out**] ボタン をクリックして関数を抜け出し、メイン関数内の次のステートメントに戻ってください。

アドレスH'104Cが強調表示されます。

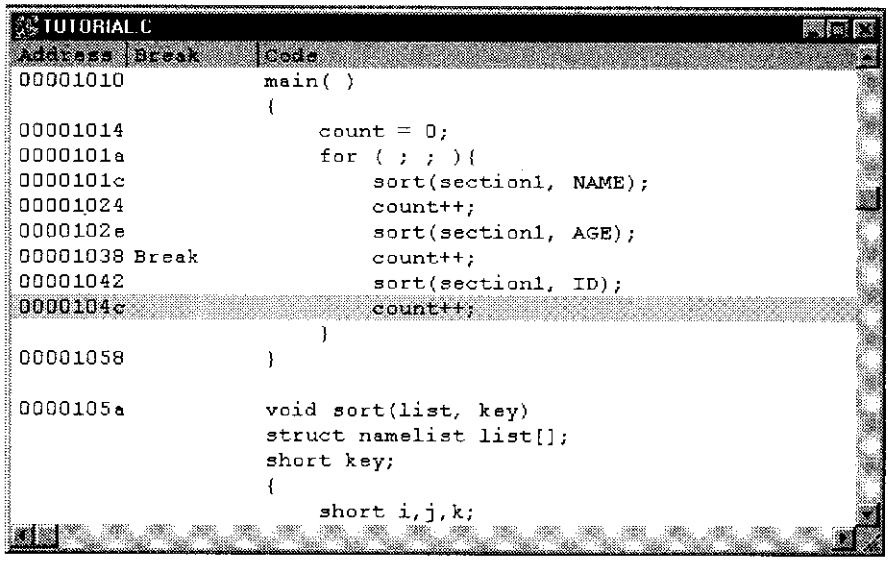


図 4-3 Step Out 実行後のプログラムウインドウ画面

・ さらに2回 [**Step In**] コマンドによってsortファンクションコールまでプログラムを実行してください。

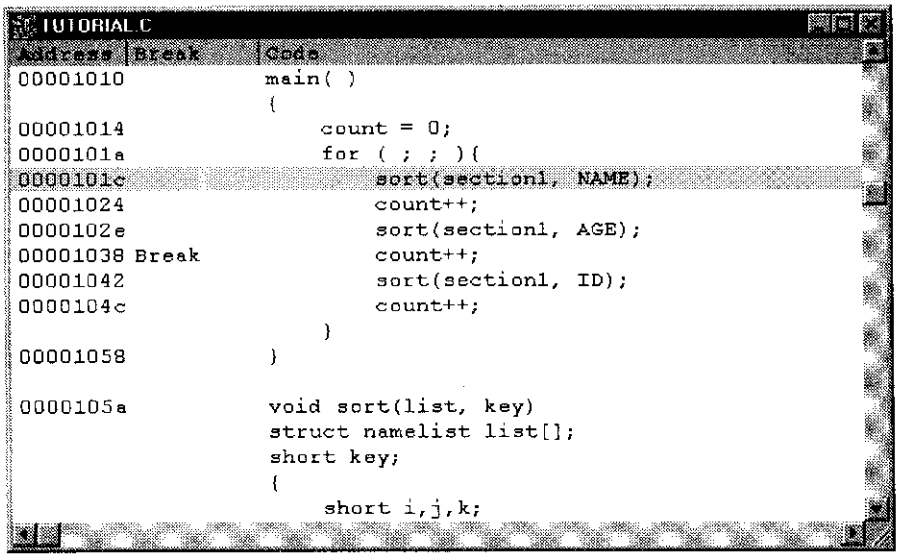


図 4-4 Step In 実行後のプログラムウインドウの画面（3）

4.8.2 関数全体のステップ実行

[**Step Over**] コマンドは、関数本体をシングルステップすることなく実行し、メインプログラムの中の次のステートメントで停止します。

・ [**Run**] メニューから [**Step Over**] を選択するか、またはツールバーの [**Step Over**] ボタン をクリックしてください。

プログラムはsort関数を実行し、次のアドレス H'1024で停止します。

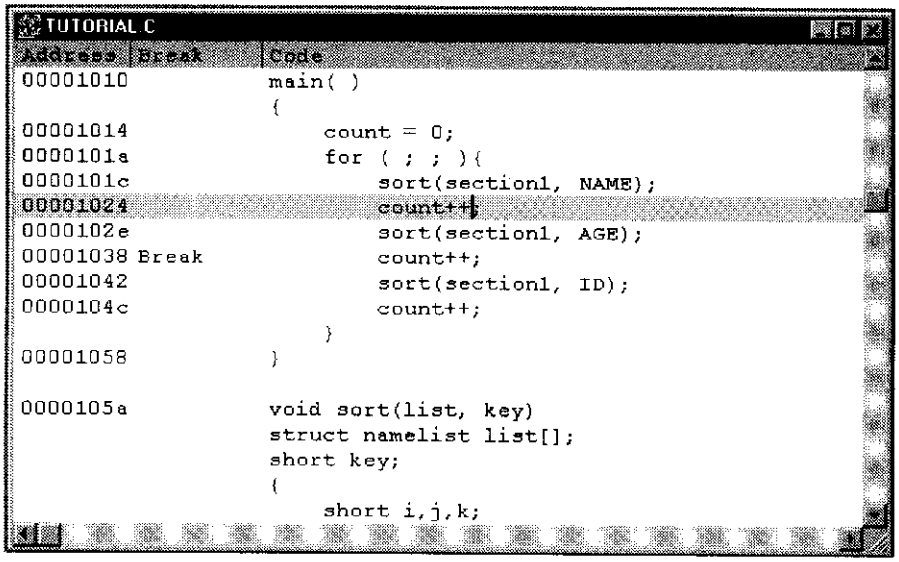


図 4-1 Step Over 実行後のプログラムウインドウの画面

4.8.3 ローカル変数の表示

[**Local Variables**] ウインドウを使って関数専用の変数を表示させることができます。例として、sort関数のローカル変数を調べます。

- [**Run**] メニューから [**Step In**] を選択するか、またはツールバーの [**Step In**] ボタン を2回クリックして、 sort関数の実行を開始してください。

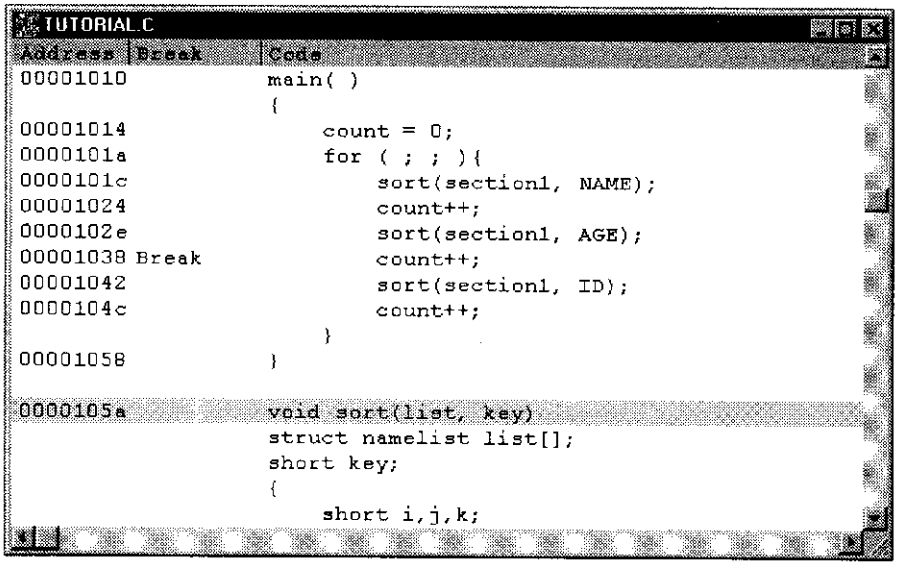


図 4-1 Step In 実行後のプログラムウインドウ画面（4）

・ [View] メニューから [Local Variable Window] を選択することによって、[Locals] ウインドウを開いてください。

最初は、ローカル変数宣言が行われていないため、[Locals] ウインドウは何も表示しません。

・ [Run] メニューから [Step In] を選択するか、またはツールバーの [Step In] ボタン を10回クリックしてステップ実行を行ってください。

[Local Variables] ウインドウは、ローカル変数とその値を表示します。

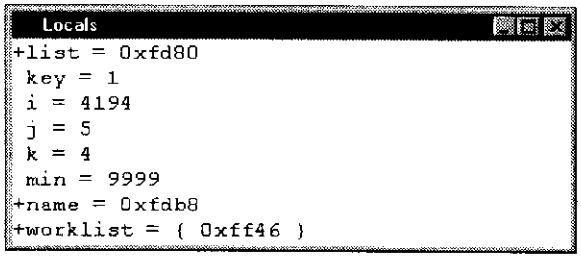


図 4-2 ローカル変数の表示

・ [Locals] ウインドウの変数strの前にあるシンボル + をダブルクリックし、配列worklistの構成要素を表示させてください。

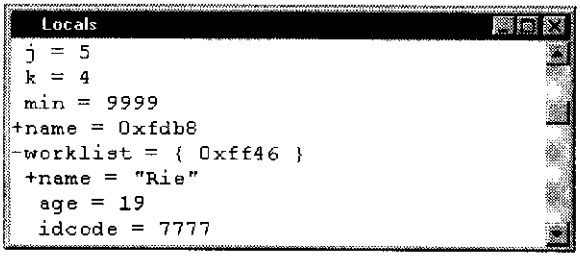


図 4-3 ローカル変数の表示（配列要素の表示）

・ [Run] メニューから [Step Out] を選択するか、またはツールバーの [Step Out] ボタン をクリックして、メインプログラムに戻ってください。

4.9　イベント検出システムの使用方法

本チュートリアルでは、[**Memory**] ウインドウでメモリ領域の内容を見ること、あるいは [**Watch**] および [**Locals**] ウインドウで変数の値を見ることによって、プログラムの動作をモニタしてきました。

しかしプログラムの動作は非常に複雑なため、メモリ領域をモニタしたり、変数を見たりできないことがあります。E6000エミュレータのイベント検出システムを使うことによりプログラムがH'109Eをアクセスした時を検出することができます。

4.9.1　イベント検出システムによるハードウェアブレークポイントの設定

イベント検出システムを使用したハードウェアブレークポイントを設定して、以下のようにプログラムの一部をモニタしてください。

- [**View**] メニューから [**Breakpoint Window**] を選択するか、またはツールバーの [**Breakpoint Window**] ボタンを クリックして、 [**Breakpoints**] ウインドウを表示してください。

- [**Add**] をクリックして、新しいブレークポイントを設定してください。

以下のダイアログボックスが現われ、ブレークポイントの属性を設定できます。

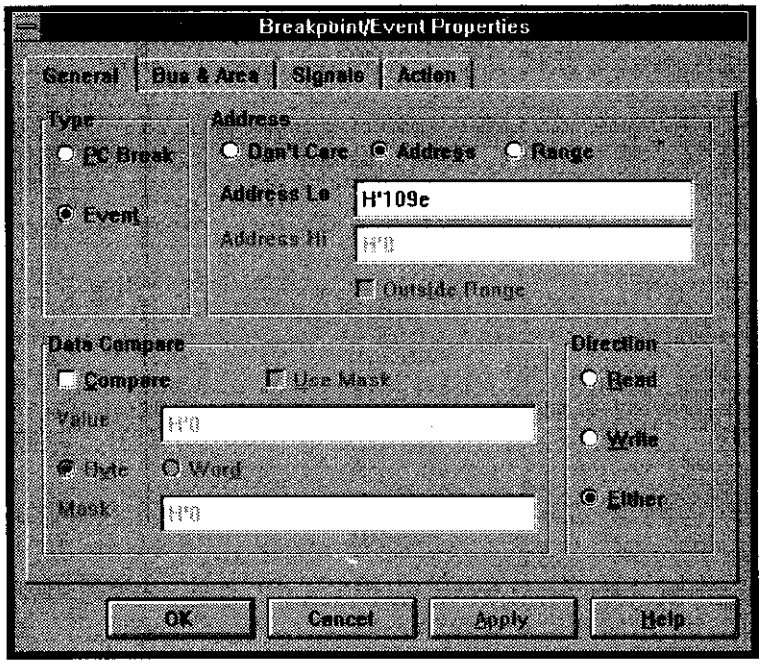


図 4-1 ブレークポイントの追加

・ [Type] 選択を [Event] にし、イベント条件として [Address Lo] ボックスにアドレス H'109e を入力してください。

・ [OK] をクリックしてブレークポイントを設定してください。

これによって、アドレス H'109e がアクセス（読み出しまたは書き込み）されたときにブレークします。

[Breakpoints] ウインドウは、設定された新しいハードウェアブレークポイントを表示します。

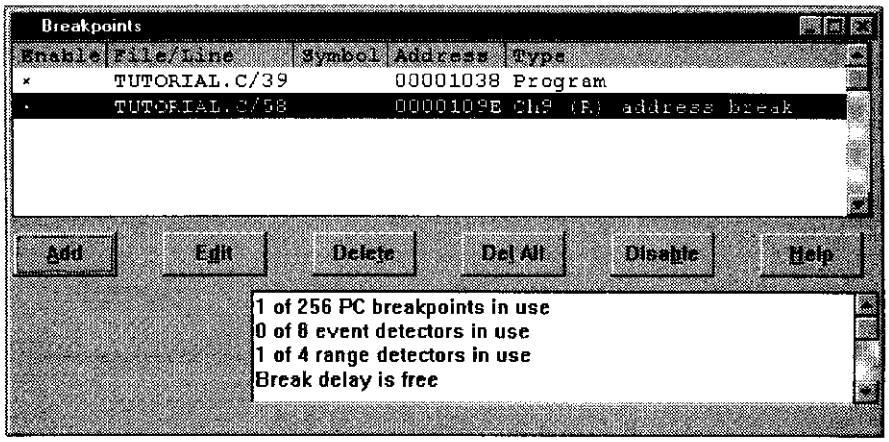


図 4-2 Breakpoint Window 画面（追加後）

・ [Run] メニューから [Go] を選択するか、あるいはツールバーの [Go] ボタン をクリックして、プログラムを現在の位置から実行してください。

アドレスH'109eで実行が停止します。

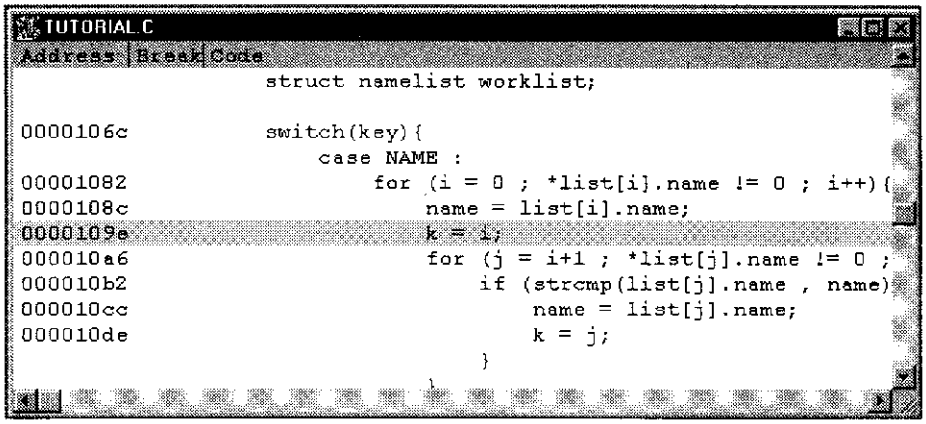


図 4-3 ブレークポイントによるプログラムの停止

ステータスバーが、"BREAK = Event Break" と表示し、イベント条件の一致によってブレークが発生したことを示します。

4.10 トレースバッファの使い方

MCUの動作を確認するため、指定されたイベントの直前までのMCUサイクルはトレースバッファに記録されています。

4.10.1 トレースバッファの表示

プログラムのアクセスアドレスを指定し、トレースバッファ内のMCUサイクルを調べることによって、どのようなアクセスが起こったかを知ることができます。

・ [View] メニューから [Trace Window] を選択するか、あるいはツールバーの [Trace Window] ボタン をクリックして、[Trace] ウインドウを開いてください。

必要ならば、最後の数サイクルが見えるようにウインドウをスクロールダウンしてください。
[Trace] ウインドウが以下のように表示されます。

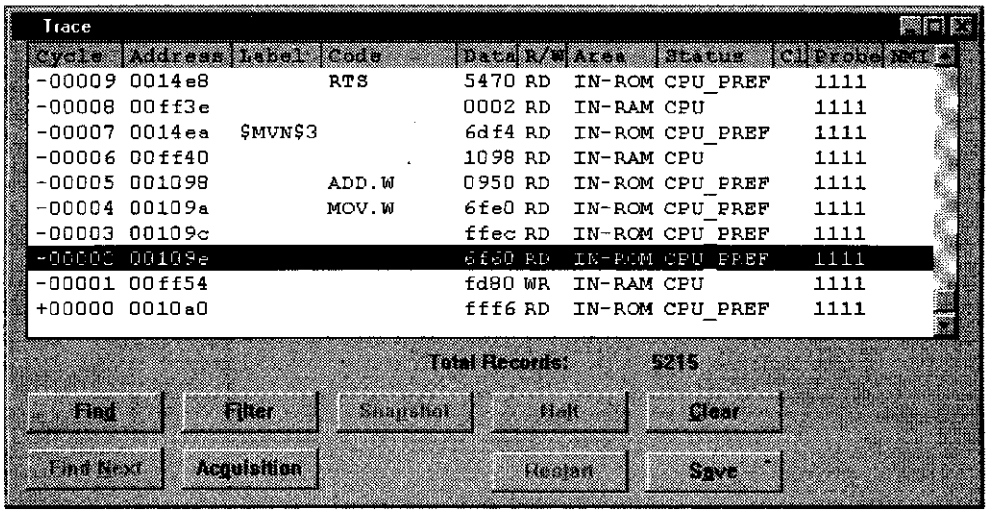


図 4-1 Trace Window 画面

・ 必要ならば、タイトルバーのすぐ下のラベルの横にあるカラムディバイダをドラッグして、カラムの幅を調節してください。

注: H8/3864シリーズ、H8/3887シリーズではclock countには何も表示されません。
ソフトウエアブレークで停止した場合、Codeカラムに“Data”、Dataカラムに“5770”が表示されます。

4.10.2　トレースフィルタの設定

現在［Trace］ウインドウは、すべてのMCUサイクルを表示しています。

・ まず［Trace］ウインドウの［Clear］をクリックして、現在のトレースバッファを削除してください。

・ 次に［Filter］をクリックして、［Trace Filter］ダイアログボックスを表示してください。

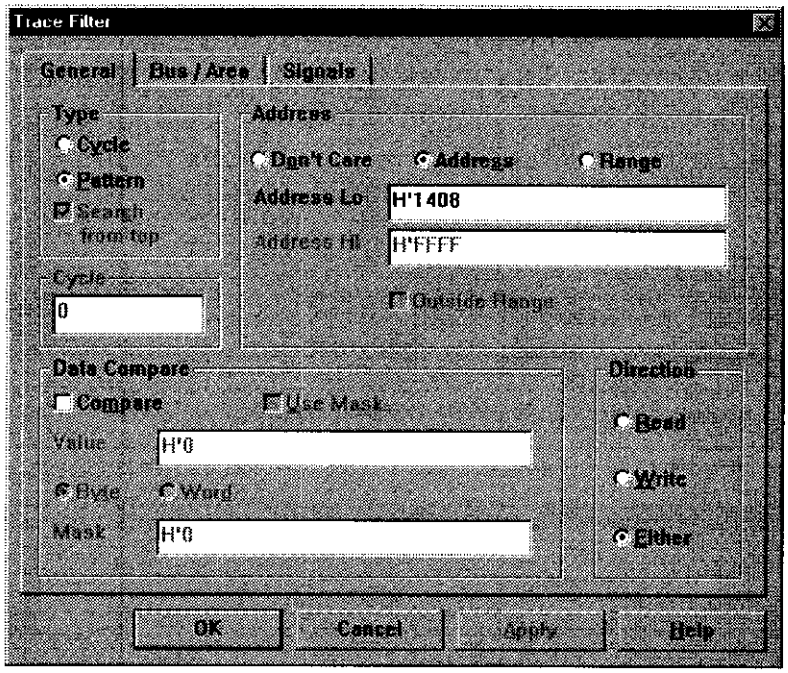


図 4-1 Trace Filter ダイアログボックス

これによって、トレースバッファに表示されるサイクルを限定するためのフィルタ条件を設定できます

・ 必要ならば、［General］をクリックして、［General］パネルを表示してください。

・ ［Pattern］タイプを選択してください。

・ ［Address］セクションで、［Address］をクリックし、［Address Lo］フィールドに H'1408 と入力してください。

・ ［Bus & Area］をクリックし、［Bus & Area］パネルを表示してください。

・ ［Bus State］を［CPU_PREFETCH］に設定してください。

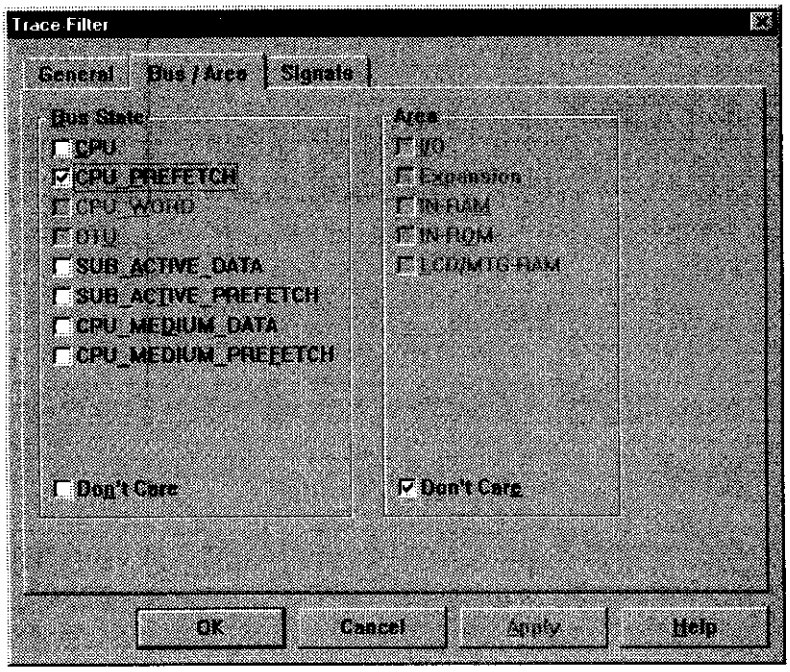


図 4-2 Bus & Area の設定

- [OK] をクリックして、トレースフィルタを保存してください。
- [View] メニューから [Breakpoint Window] を選択して、ブレークポイントウインドウを開き、全てのブレークポイントを削除してください。
- [Run] メニューから [Go] を選択して、プログラムの終わりまで実行してください。
- [Run] メニューから [Halt Program] を選択して、トレースバッファを見ることができるように実行を停止してください。

トレースウインドウには、MCUがH'1408番地をアクセスしたサイクルだけが表示されます。

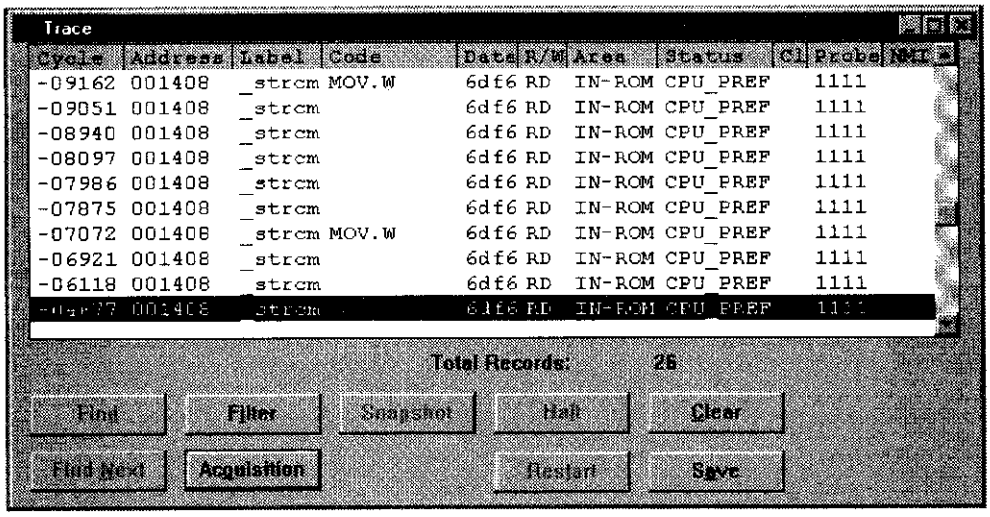


図 4-3 Trace Window 画面（トレースフィルタ指定）

4.11 セッションの保存

終了する前に、次回のデバッグセッションで同じE6000エミュレータとHDIコンフィグレーションを使用して再開できるように、セッションを保存しておくと良いでしょう。

- [**File**] メニューから [**Save Session...**] を選択してください。

- [**File**] メニューから [**Exit**] を選択して、HDIから抜け出してください。

4.11.1 さてつぎは？

このチュートリアルは、E6000エミュレータのいくつかの主な特長と、HDIの使用方法を紹介しました。E6000エミュレータで提供されるエミュレーションツールを組み合わせることによって、非常に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、識別することにより、それらの問題点を効果的に調査することができます。

HDIの使用方法に関する詳細については、別に発行されている「日立デバッギングインタフェースユーザーズマニュアル」を参照してください。

5 H8/3864シリーズ、H8/3887シリーズ用 E6000エミュレータ HDIの機能

本章は、H8/3864シリーズ、H8/3887シリーズ専用のHDIの特徴に関する情報について述べます。あらゆるターゲットに共通するHDIの一般的な特長に関しては、別に発行されている「日立デバッギングインタフェースユーザーズマニュアル」を参照してください。以下にHDIメニューと「日立デバッギングインタフェースユーザーズマニュアル」（HDIマニュアル）および本マニュアルに記載する項目の対応表を示します。

表 5-1 HDIのメニューとマニュアルの対応表

メニューバー	プルダウン　メニュー	HDIマニュアル	本マニュアル
File Menu	Load Program...	○	4.5
	Save Memory...	○	-
	Verify Memory...	○	-
	Save Session	○	4.11
	Load Session...	○	-
	Save Session As...	○	-
	Initialise	○	-
	Exit	○	-
Edit Menu	Cut	○	-
	Copy	○	-
	Paste	○	-
	Find...	○	-
	Set Line...	○	-
	Fill Memory...	○	-
	Move Memory...	○	-
	Test Memory...	○	-
	Update Memory	○	-
View Menu	Toolbar	○	-
	Status Bar	○	-
	Breakpoint Window	○	4.6.4, 4.9.1, 5.2, 5.3
	Command Line Window	○	5.7
	I/O Register Window	○	-
	Local Variable Window	○	4.8.3

表 5-1 HDIのメニューとマニュアルの対応表（つづき）

メニューバー	プルダウン　メニュー	HDIマニュアル	本マニュアル
View Menu	Memory Mapping Window	○	4.4.2, 5.4
	Memory Window...	○	4.7.1
	Performance Analysis Window	○	–
	Program Window...	○	4.5
	Register Window	○	4.6.3
	Status Window	○	4.6.2
	Text Window	○	–
	Trace Window	○	4.10, 5.5
	Watch Window	○	–
	Localized Dump Windows...	○	–
Run Menu	Go	○	4.6.2
	Go Reset	○	4.6.2
	Go to Cursor	○	–
	Run...	○	–
	Step In	○	4.8
	Step Over	○	4.8
	Step Out	○	4.8
	Step...	○	4.8
	Halt Program	○	–
	Set PC To Cursor	○	–
	Reset CPU	○	–
Setup Menu	Options	○	–
	Radix	○	–
	Customise	○	–
	Select Platform...	○	–
	Configure Platform...	○	4.4.1, 5.1
Tools Menu	Symbols...	○	–
	Evaluate...	○	–
Window Menu	Cascade	○	–
	Tile	○	–
	Arrange Icons	○	–
	Close All	○	–
Help Menu	Index	○	–
	Using Help	○	–
	Search for Help on	○	–
	About HDI	○	–

5.1 コンフィグレーションダイアログボックス

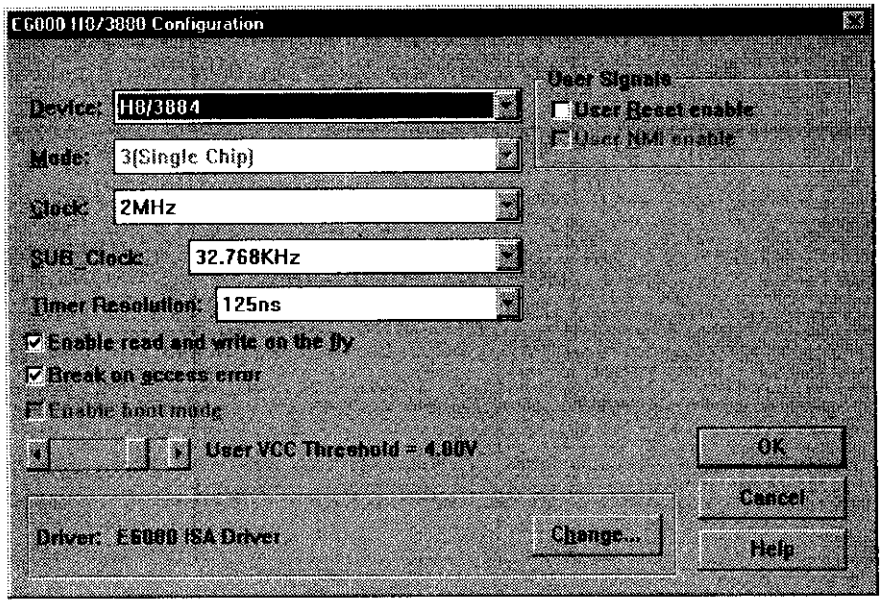


図 5-1 コンフィグレーションダイアログボックス

コンフィグレーションダイアログボックスによって、E6000エミュレータをセットアップします。

コンフィグレーションダイアログボックスを表示するには、[Setup] メニューから [Configure Platform...] を選択してください。

以下の表は、コンフィグレーションダイアログボックスで表わされるオプションについて説明します。

表 5-1 コンフィグレーションダイアログボックス

オプション	説明
デバイス (Device)	エミュレーションするMCUを指定します。
モード (Mode)	H8/3864シリーズ、H8/3887シリーズではモード3に固定されています。
エミュレーションクロック (Clock)	以下のようにエミュレーションクロックを指定します。 2MHz、1MHz、0.5MHz、またはTarget÷2(H8/3864シリーズ、H8/3887シリーズ　システムクロック) 32.768kHz, 38.4kHz, 307.2kHz,またはTarget(H8/3864シリーズ、H8/3887シリーズ　サブクロック)に設定できます。
タイマ分解能 (timer Resolution)	実行時間の測定に使われるタイマの分解能を指定します。 20ns, 125ns, 250ns, 500ns, 1μs, 2μs, 4μs, 8μs, 16μs のいずれかの値に設定できます。
ユーザ信号 (User Signals)	ユーザシステムからのリセット信号を有効または無効にできます。ボックスをチェックすると、信号が有効になります。
プログラム実行中のリードライトの許可 (Enable read and write on the fly)	プログラム実行中でのユーザメモリのリードライトを有効または無効にできます。
ブートモード (Enable boot mode)	MCUのフラッシュメモリへのブートプログラミング動作を有効にします。(H8/3864シリーズ、H8/3887シリーズは使用できません。)
アクセスエラーでのブレーク (Break on access error)	不当なアクセス (ROM領域へのライト等) ブレークを有効または無効にできます。チェックしなければ、ROMへのあらゆる書き込みおよびアクセス禁止された領域へのアクセスが無視されます。
ユーザVccスレッショルド (User VCC Threshold = x.xxV)	ユーザシステム電圧レベルを監視します。スレッショルドによって設定された値より低くなると、ユーザVccが低いことをシステムステータスウインドウによってユーザに知らせます。

5.2 ブレークポイント

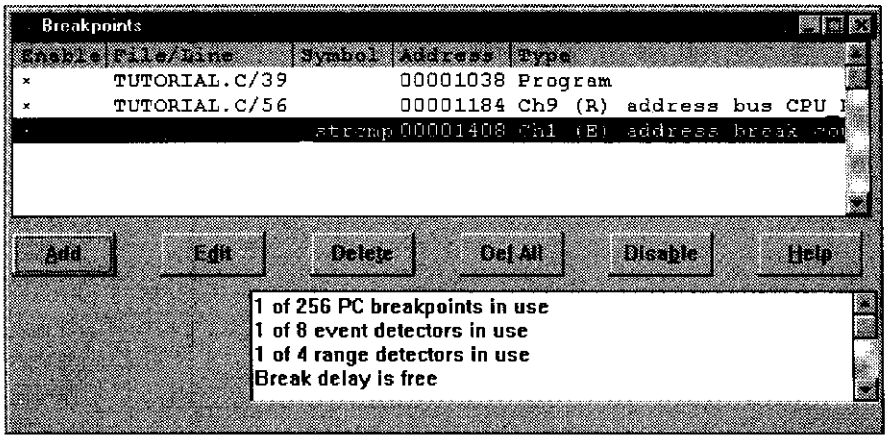


図 5-1 Breakpoints 画面

[**Breakpoints**] ウインドウは設定された全てのブレークポイントの一覧を表示します。

ブレークポイントウインドウを表示するには、[**View**] メニューから [**Breakpoint Window**] を選択してください。

現状のブレークポイントを変更するには、それをダブルクリックするか、あるいは [**Breakpoints**] リストの中から選択して [**Edit**] をクリックしてください。

ブレークポイントを許可または禁止するには、[**Breakpoints**] リストの中から選択し、[**Disable/Enable**] をクリックしてください。ブレークポイントが許可されると、×が [**Enable**] コラムの中に表示されます。

ブレークポイントを削除するには、ブレークポイントリストの中から選択し、[**Delete**] をクリックしてください。全てのブレークポイントを削除するには、[**Del All**] をクリックしてください。

新しいブレークポイントを設定するには、[**Add**] をクリックして、[**Breakpoint/Event Properties**] ダイアログボックスを表示してください。そして、付け加えたいブレークポイントの条件を設定してください。

[**Breakpoints/Event Properties**] ダイアログボックスの詳細に関しては、以下の「5.3 イベント検出システム」を参照してください。

5.2.1 プログラムブレークポイントを設定する

プログラムブレークポイントを設定するには、[**Type**] を [**PC Break**] に設定し、[**Address Lo**] フィールドにブレークポイントのアドレスを入力してください。

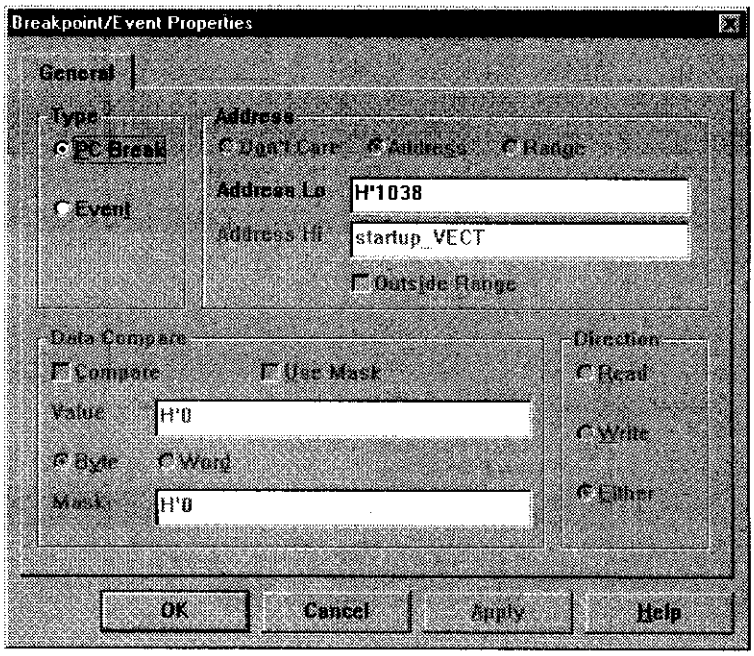


図 5-1 ブレークポイントの設定

あるいは、プログラムウインドウの［**Break**］カラムをダブルクリックしてください。

5.3 イベント検出システム

イベント検出システムに、MCU信号の状態を指定したイベントを設定できます。このイベントはE6000エミュレータのトレース、ブレーク、およびイベント間実行時間測定機能を制御します。

イベント検出システムはイベントチャネルおよび範囲チャネルで構成し、指定されたイベントがいつ発生したかを検出できます。また、最大８イベントによるシーケンス指定も可能です。この場合、それぞれのイベントは、シーケンスにおける前のイベントの発生によって起動、あるいは停止します。

以下の表は、イベントチャネルおよび範囲チャネルで指定できる条件を示します。

表 5-1 イベントチャネルとオプションの設定

オプション	イベントチャネル	範囲チャネル
アドレスまたはアドレス範囲内	○	○
アドレス範囲外	○	
データの値（マスク機能付）	○	○
リードまたはライト	○	○
アクセスタイプ（命令プリフェッチなど）	○	○
アクセス領域（内蔵ROM、内蔵RAMなど）	○	○
外部プローブの値	○	○
イベントの一致回数	○	
シーケンスでの組み合わせが可能	○	

[**Breakpoint/Event Properties**] ダイアログボックスによって、各イベントをブレーク、トレース、もしくはイベント間実行時間測定の内、どれに使用するか設定します。

イベントをブレークに設定するには、[**Type**] を [**Event**] に設定してください。すると、[**Breakpoint/Event Properties**] ダイアログボックスは、４つのオプションパネルを表わし、ブレークポイントで使うイベントの全ての条件（General, Bus & Area, Signals, Action）を設定できます。

5.3.1 [General]

[**General**] 属性パネルによって、イベントチャネルのアドレスとデータアクセス条件を設定できます。

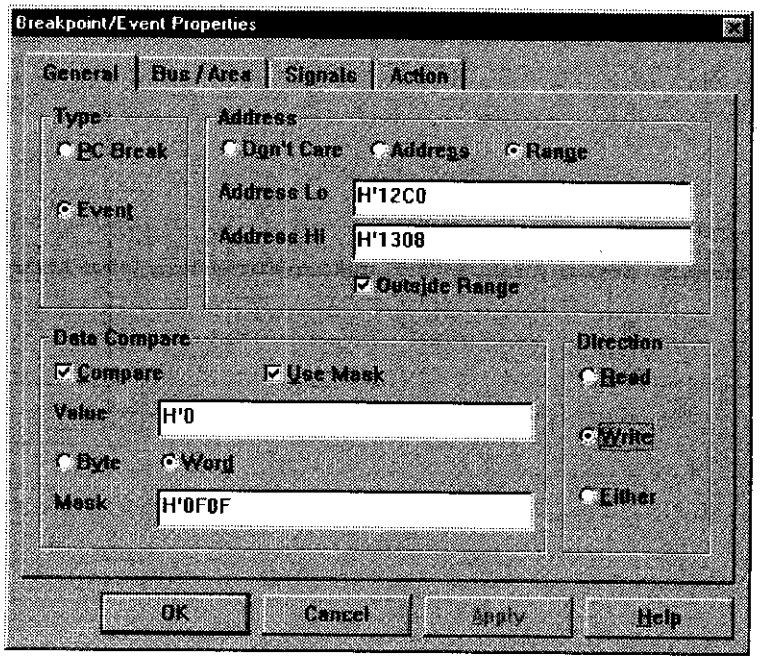


図 5-1 イベントの設定 (General)

アドレス

アドレスまたはアドレスの範囲がアクセスされたときに、チャネルを起動できます。指定された範囲外のアドレスへアクセスするとチャネルを起動するように指定するには、[**Outside Range**] を選択してください。

データコンペア

設定した値とデータバスの値が一致したときにチャネルを起動できます。データの特定のビットをマスクする場合は、[**Use Mask**] を選択してください。

ディレクション

リード、ライト、またはリードライトアクセス時にチャネルが起動できます。

5.3.2 [Bus & Area]

特定のバスステートまたはアクセスされたメモリ領域でチャネルを起動できます。

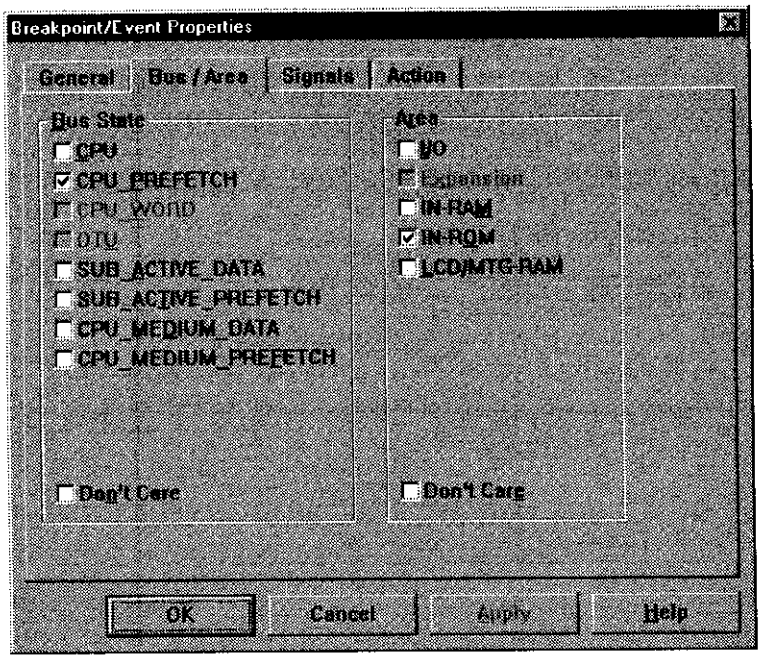


図 5-1 イベントの設定 (Bus & Area)

5.3.3 [Signals]

4つの外部プローブ信号の特定の組み合わせでイベントを起動するように指定します。

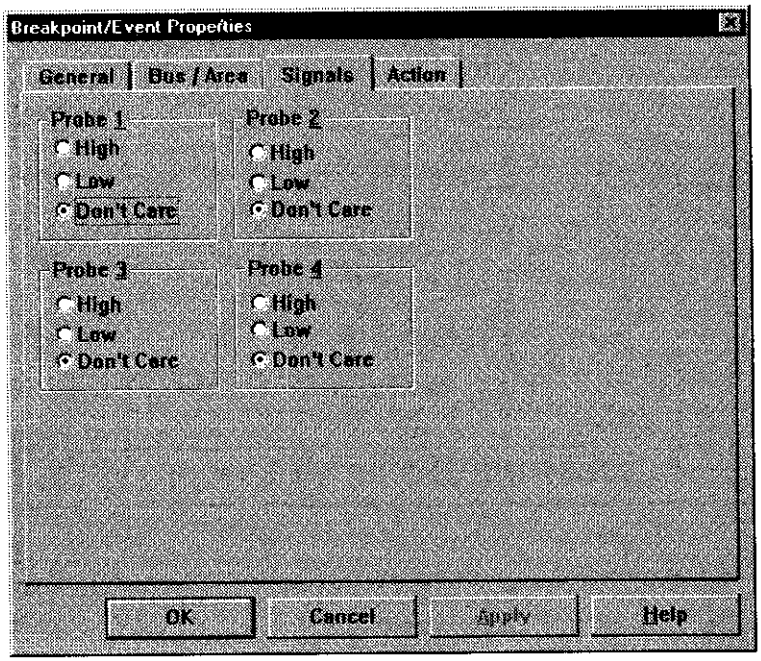


図 5-1 イベントの設定 (Signals)

5.3.4 [Action]

イベントが起動されるときのアクションを指定します。

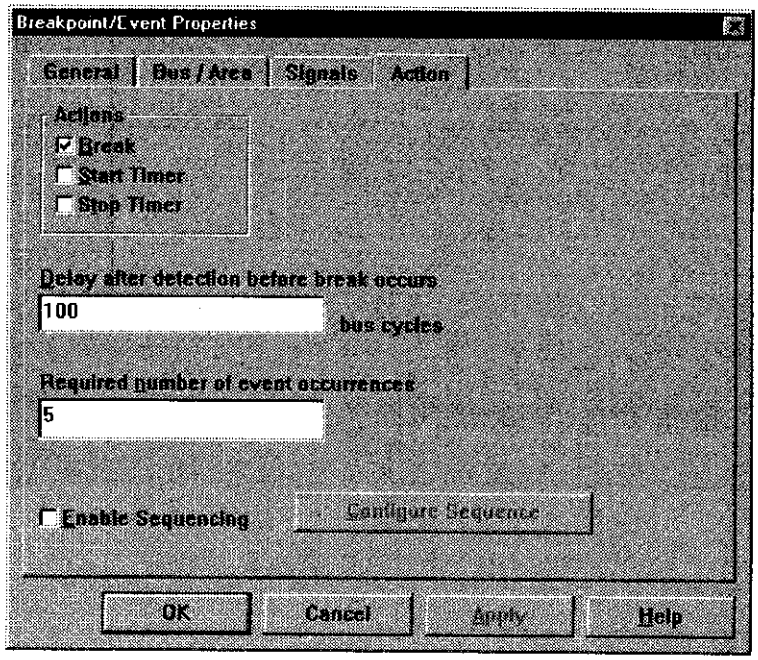


図 5-1 イベントの設定 (Action)

表5-4のアクションが指定できます。

表 5-1 イベントアクション

アクション	説明
ブレーク	プログラムの実行を停止します。
タイマ開始	イベント間実行時間測定タイマを開始します。タイマ分解能については、「5.1 コンフィグレーションダイアログボックス」を参照してください。
タイマ停止	イベント間実行時間測定タイマを停止します。

イベントが起動された後、指定された数のバスサイクルだけチャネルの起動を遅らせるには、[**Delay after detection before break occurs**] フィールドにバスサイクル数を入力してください。

指定された回数だけイベントが起動されるまでチャネルの起動を遅らせるには、[**Required number of event occurrences**] フィールドに必要なイベント発生数を入力してください。

イベントのシーケンスを生成するには、シーケンスの一部を形成する全てのイベントの［**Enable Sequencing**］オプションを選択してください。

5.3.5 イベントシーケンス

シーケンスを構成するには、［**Action**］で［**Configure Sequence**］ボタンをクリックしてください。［**Event Sequencing**］ダイアログボックスが表示されます。

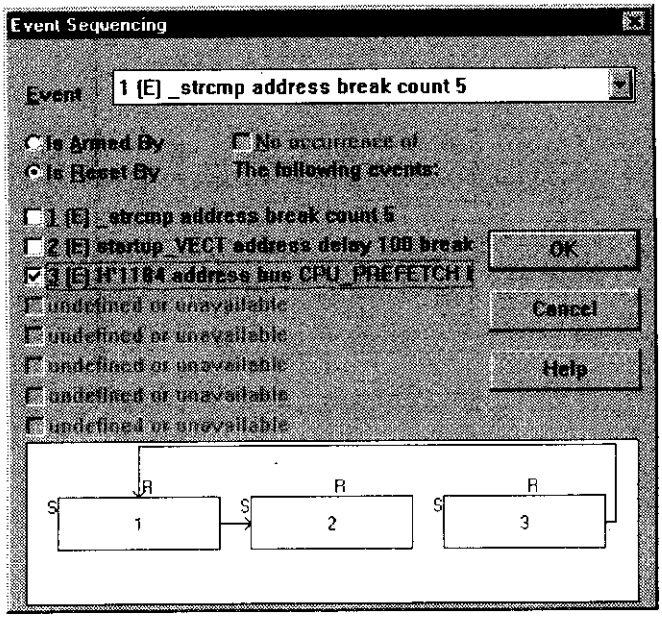


図 5-1 イベントシーケンス画面（1）

このダイアログボックスによって、シーケンスの各イベントにおいて、イベントの前提条件となるイベント、またはリセットするイベントを１つ以上指定できます。

［**Event**］ドロップダウンリストボックスから、構成したいイベントを選択してください。これによって、［**Enable Sequencing**］で指定されたあらゆるイベントが選択できます。

現在選択されているイベントにおいて、［**Is Armed By**］をクリックして、イベントの前提条件となるイベントを設定してください。

同様に、［**Is Reset By**］をクリックして、イベントをリセットするイベントを設定してください。

5.3.6 イベントの前提条件

たとえば、４つのアドレスのシーケンスが発生した時にだけ起動されるイベントシーケンスを定義するには、以下のように設定します。

4 is armed by 3.
3 is armed by 2.
2 is armed by 1.

[**Event Sequencing**] ダイアログボックスは、定義したシーケンスを図式表示します。

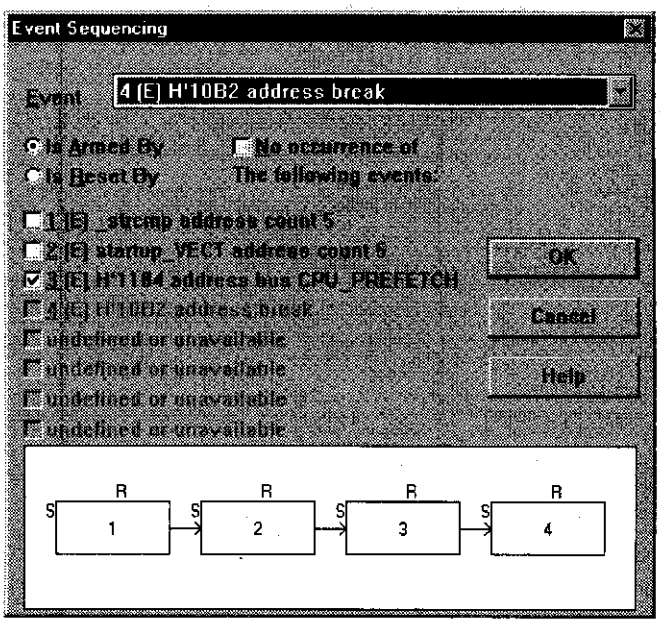


図 5-1 イベントシーケンス画面（2）

シーケンスを定義するときは、シーケンスの最後のイベントだけを［**Break**］として定義してください。

5.3.7 イベントをリセットする

イベントが、シーケンスの中の別のイベントによってリセットされるように、指定することもできます。たとえば、イベント2の後にイベント3が続き、その後にイベント4が続いて、イベント1はその間には発生しない（もし、この間にイベント1が発生した場合は、前提条件のイベント発生がクリアされる場合）という条件でブレークが発生するようにするには、イベントシーケンスを以下のように設定してください。

4 is armed by 3 and reset by 1.
3 is armed by 2 and reset by 1.
2 is reset by 1.
1 is reset by 1.

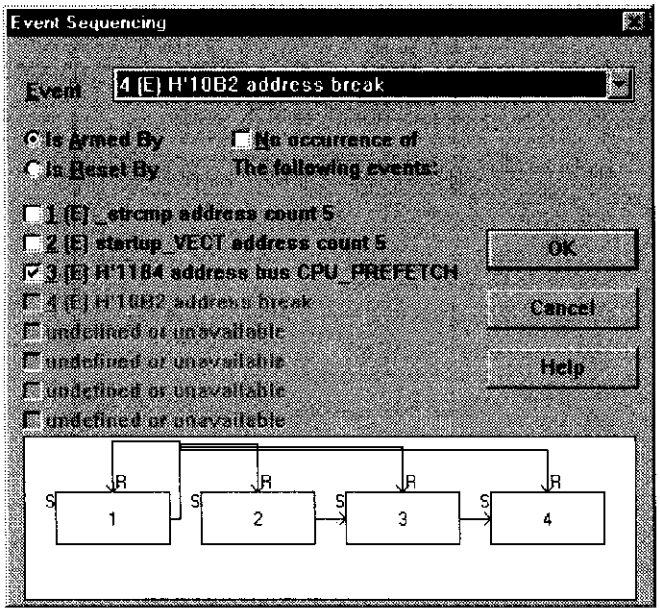


図 5-1 イベントシーケンス画面（3）

5.4 メモリマッピングウインドウ

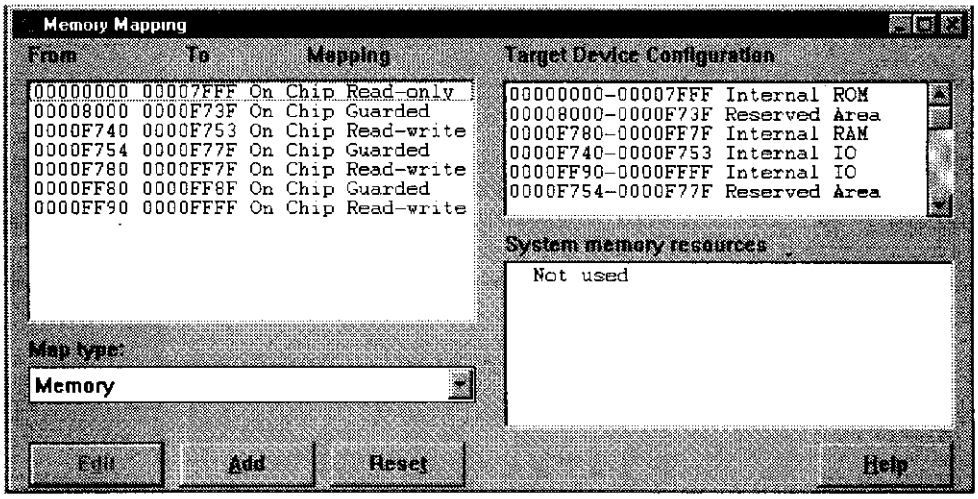


図 5-1 Memory Mapping 画面

メモリマッピングウインドウは、エミュレーションメモリのマッピングの表示および編集ができます。

メモリマッピングウインドウを表示するには、 [View] メニューから [Memory Mapping Window] を選択してください。

メモリマッピングを編集するには、該当メモリブロックをダブルクリックしてください。あるいは、メモリマッピングリストの中から選択し、［**Edit**］をクリックしてください。以下のダイアログボックスがメモリブロックの現在の設定を示します。

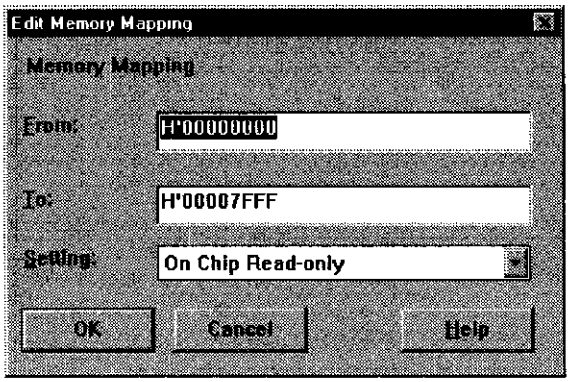


図 5-2 メモリマッピングの変更

［**From**］フィールドと［**To**］フィールドに、メモリブロックのアドレスの範囲を指定してください。そして、［**Setting**］ドロップダウンリストボックスで、メモリのタイプを選択してください。以下のオプションが選択できます。

表 5-1 メモリタイプの定義

メモリ	説明
オンチップ	MCU内蔵メモリをアクセスします。
エミュレータ	エミュレーションメモリをアクセスします。

オプションにおいて、以下の３つのアクセスタイプから１つを指定できます。

表 5-2 アクセスタイプの定義

アクセスタイプ	説明
Read-Write	RAM
Read-Only	ROM
Guarded	アクセス不可

選択したMCUタイプとモードのデフォルトマッピングにメモリマッピングをリセットするには、［**Reset**］をクリックしてください。

5.5 トレースウインドウ

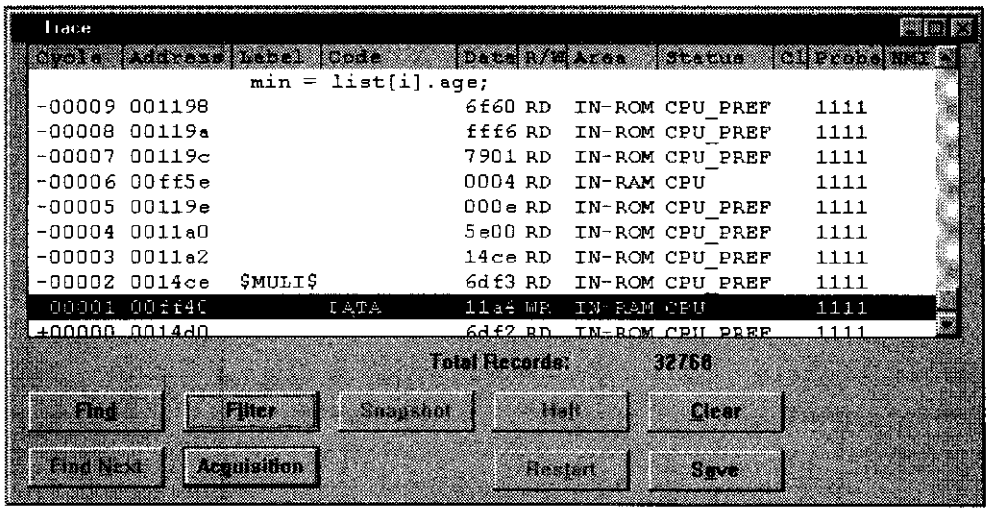


図 5-1 Trace Window 画面

トレースウインドウは、トレースバッファの内容を表示します。

トレースウインドウを表示するには、[**View**] メニューから [**Trace Window**] を選択してください。

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

バスサイクルのクロック数（ϕ）を [**Clock**] カラムに最大７サイクルまで表示します。
E6000 H8/3864シリーズ、H8/3887シリーズ用エミュレータでは、何も表示しません。

４本の外部プローブの値を [**Probes**] カラムに表示します。右から順に外部プローブ番号１、２、３４となり、Highを１、Lowを０と表示します。

[**NMI**] カラムは、E6000 H8/3864シリーズ、H8/3887シリーズ用エミュレータでは、何も表示しません。

[**Clear**] をクリックすることによりトレースバッファを削除できます。また [**Save**] をクリックして、トレースバッファの内容をファイルに保存できます。

デフォルトでは、トレースバッファはあらゆる実行サイクルを収集し、最後の32768サイクルを保存します。また、指定されたサイクルだけを表示するフィルタ機能を設定できます。

5.5.1 [Filter]

フィルタを定義するには、トレースウインドウの [**Filter**] をクリックしてください。

5.5.2 [Find]

トレースバッファの特定のトレースデータを検索するには、 [**Find**] をクリックしてください。見つけたいトレースを指定するため、同じダイアログボックスが現われます。

5.5.3 [Cycle]

フィルタとして特定のサイクルを指定するには、 [**Type**] を [**Cycle**] に設定し、 [**Cycle**] ボックスにサイクル番号を入力してください。

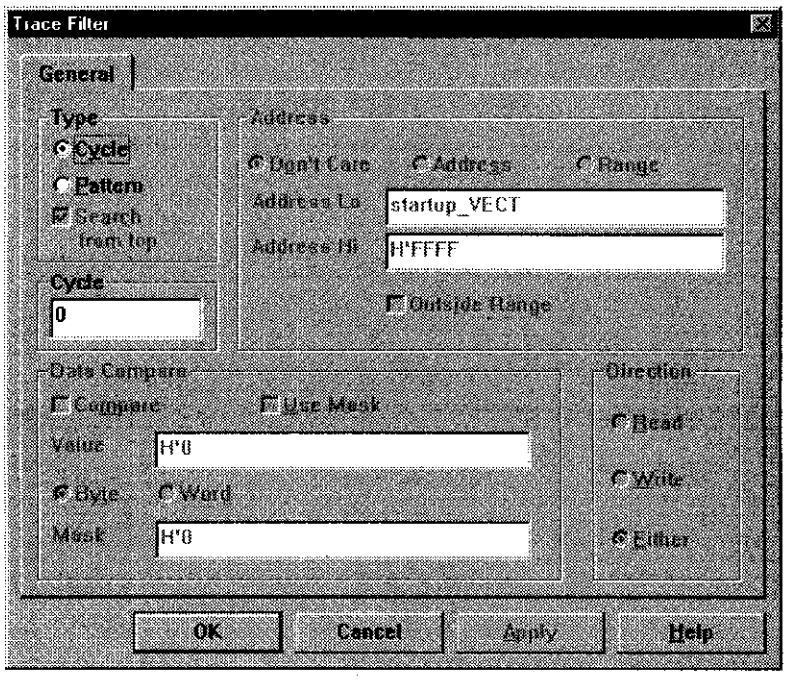


図 5-1 Trace Filter 画面 (General) (1)

5.5.4 [Pattern]

フィルタパターンを入力するには、[**Type**] を [**Pattern**] に設定し、必要な値を指定してください。

[**Trace Filter**] ダイアログボックスが3つのオプションパネルを示しますので、[**General**]、[**Bus & Area**]、および [**Signals**] のうちのどのサイクルを表示するかを指定できます。

5.5.5 [General]

[**General**] パネルによって、表示されるサイクルのアドレスとデータのアクセス条件を設定できます。

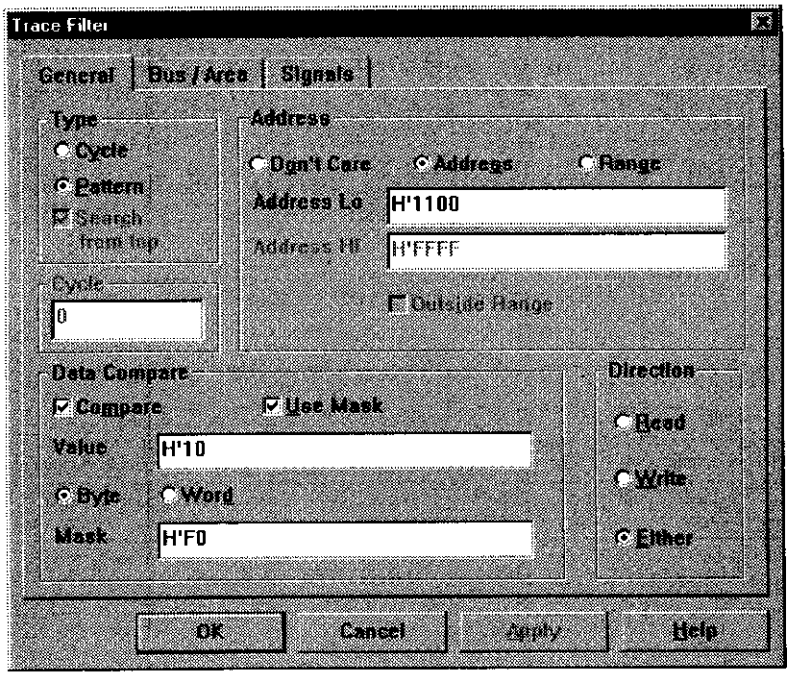


図 5-1 Trace Filter 画面 (General) (2)

5.5.6 [Bus & Area]

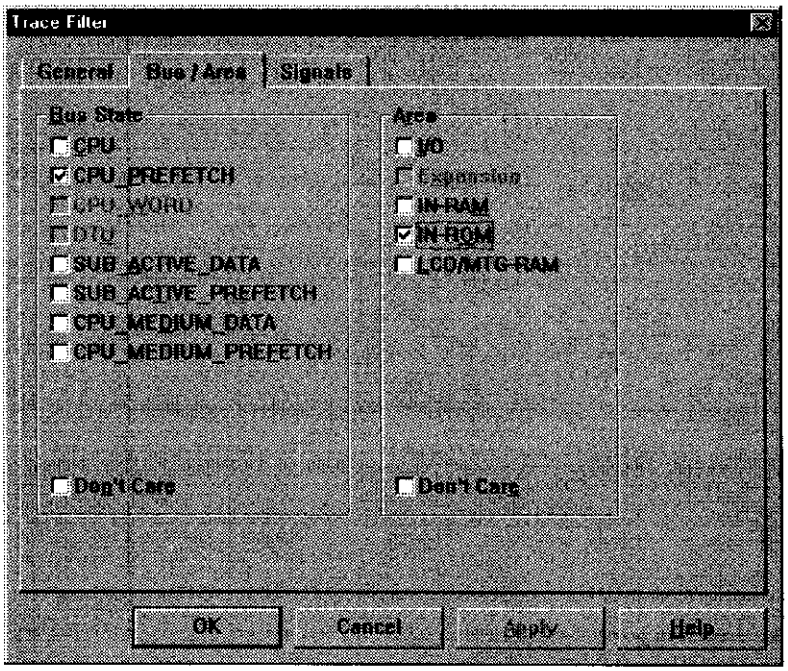


図 5-1 Trace Filter 画面 (Bus & Area)

5.5.7 [Signals]

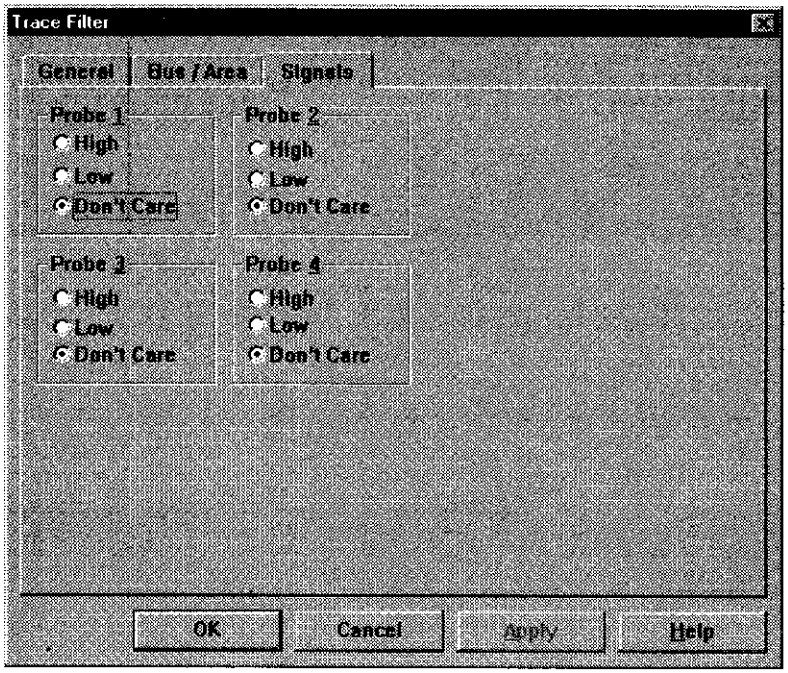


図 5-1 Trace Filter 画面 (Signals)

5.6 トレース制御

バッファは、すべてのバスサイクルあるいは選択されたサイクルだけを記憶するためにセットアップされます。トレース制御を指定するには、トレースウインドウ中の［**Acquisition**］をクリックしてください。

［**Trace Acquisition**］ダイアログボックスは以下のパネルを表示しますので、トレース制御条件を指定できます。

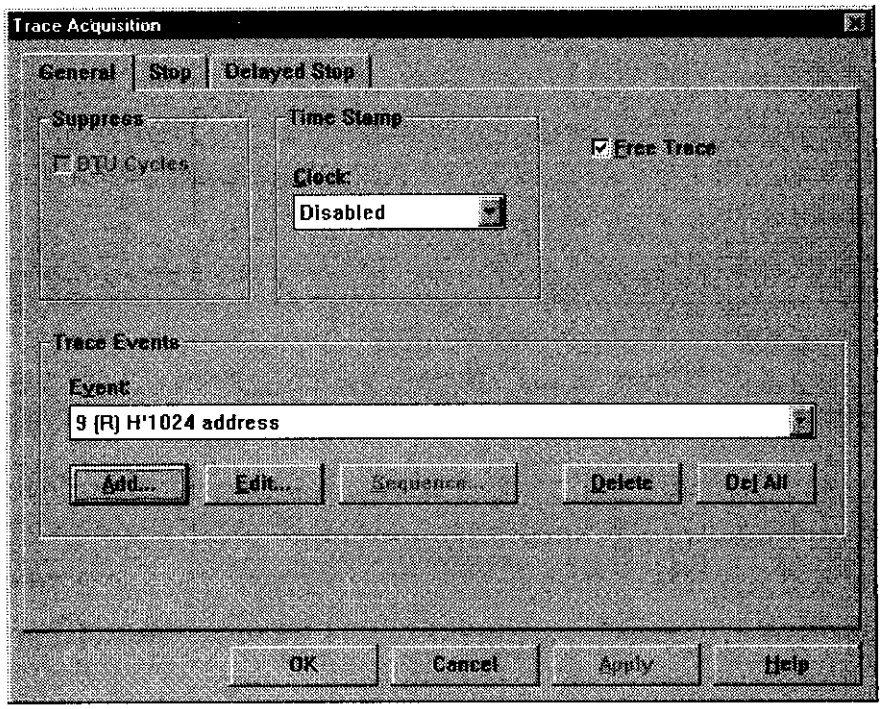


図 5-1 Trace Acquisition 画面 (General)

5.6.1 [General]

[**Suppress**] スペース中の [**DTU Cycles**] ボックスは、無効となります。

[**Time Stamp**] でクロックを指定することで、MCUの動作サイクルに対応した実行時間をトレースバッファに表示させることができます。ただし、実行時間表示を指定した場合は、以下の情報は表示されません。

・Area, ・Status, ・Probes

また、乗除算命令の実行も正しく表示されません。

[**Free Trace**] ボックスをチェックして、すべてのトレース制御条件を禁止してください。これは、条件を削除することなくトレース制御を一時的に禁止します。[**Free Trace**] をチェックすると、トレーススストップ、トレースディレイストップおよび [**Suppress**] セクションで指定された条件を除くすべてのバスサイクルを取得できます。

[**General**] パネルの [**Trace Events**] セクションによって、トレース制御のイベント、およびイベントシーケンスを定義できます。

イベントドロップダウンリストボックスが、現在設定されているすべてのイベントを表示します。

新しいイベントを加えるには、[**Add...**] をクリックし、[**Breakpoint/Event Properties**] ダイアログボックスにイベントの内容を入力してください。利用できるオプションの詳しい情報については、「5.3 イベント検出システム」を参照してください。

イベントを編集するには、[**Event**] リストの中のイベントを選択し、[**Edit...**] をクリックしてください。

イベントのシーケンスを設定するには、[**Sequence...**] をクリックしてください。このオプションは、一つ以上のイベントが設定され、[**Enable Sequencing**] が選択されているときのみ使えます。

イベントを削除するには、[**Event**] リストの中のイベントを選択し、[**Delete**] をクリックしてください。また、すべてのトレースイベントを削除するには、[**Del All**] をクリックしてください。

5.6.2 [Stop]

指定したイベントが発生した時にトレース取得を停止できます。

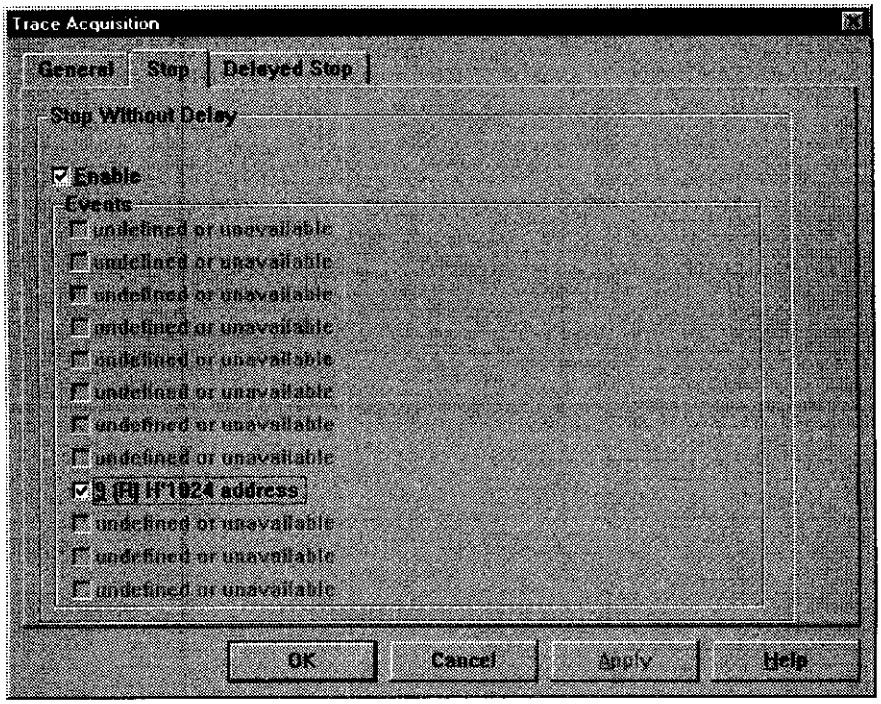


図 5-1 Trace Acquisition 画面 (Stop)

5.6.3 [Delayed stop]

指定したイベントが発生した後、指定されたサイクル数後にトレース制御を停止できます。

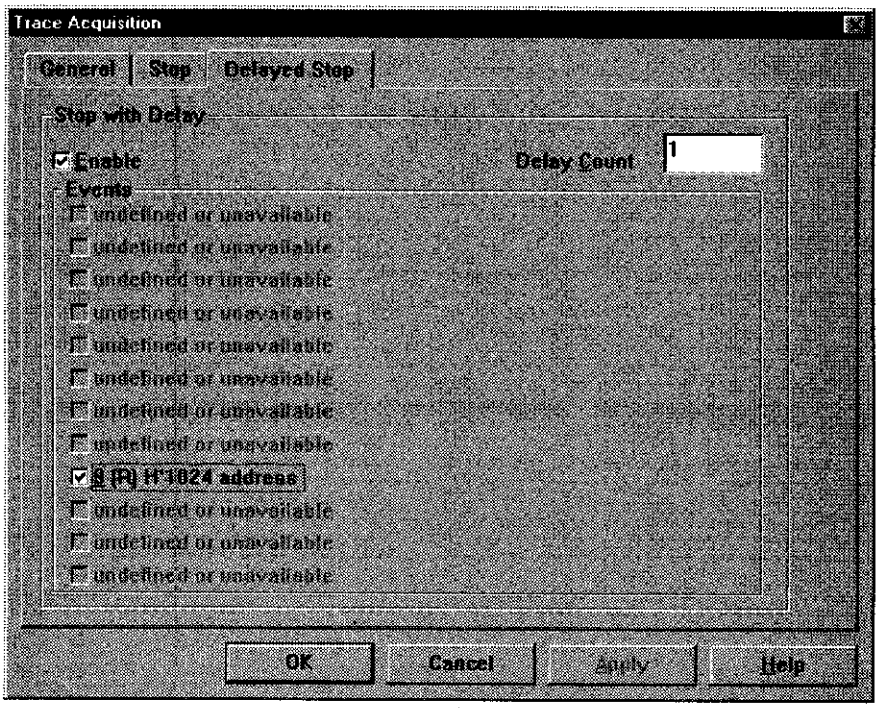


図 5-1 Trace Acquisition 画面 (Delayed Stop)

5.7 コマンドライン

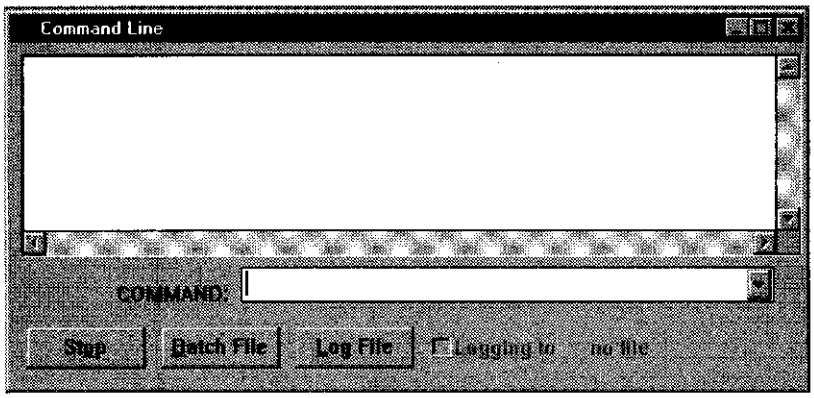


図 5-1 Command Line Window 画面

HDIを使って効率よくデバッグを行うために、コマンドラインウインドウを使ってコマンドの入力と実行ができます。コマンドラインウインドウを表示するには、[**View**] メニューから [**Command Line Window**] を選択してください。

H8/3864シリーズ、H8/3887シリーズ専用のコマンドライン機能の詳細については、「6　コマンドライン機能」を参照してください。

6 コマンドライン機能

本章では、H8/3864シリーズ、H8/3887シリーズ専用のコマンドライン機能について説明します。その他のコマンドライン機能については、「日立デバッギングインタフェースユーザーズマニュアル」を参照してください。以下にコマンドライン機能とHDIマニュアルおよび本マニュアルに記載する項目の対応表を示します。

表 6-1 HDIコマンドライン機能とマニュアルの対応表

コマンド名	短縮形	HDI マニュアル	本 マニュアル	説明
!	–	○	–	コメント
ACCESS	AC	○	–	不当アクセスに対する動作の設定
ANALYSIS_RANGE	AR	○	–	性能分析範囲の設定と表示
ANALYSIS_RANGE_DELETE	AD	○	–	性能分析範囲の解除
ANALYSIS	AN	○	–	性能分析機能の有効化／無効化
ASSEMBLE	AS	○	–	アセンブルの実行
ASSERT	–	○	–	コンディションのチェック
BREAKPOINT / EVENT	BP, EN	–	6.1	ブレークポイント／イベントの設定
BREAKPOINT_CLEAR EVENT_CLEAR	BC EC	–	6.2	ブレークポイント／イベントの解除
BREAKPOINT_DISPLAY EVENT_DISPLAY	BD ED	–	6.3	ブレークポイント／イベントの表示
BREAKPOINT_ENABLE EVENT_ENABLE	BE EE	–	6.4	ブレークポイント／イベントの有効化／無効化
BREAKPOINT_SEQUENCE EVENT_SEQUENCE	BS ES	–	6.5	シーケンスの定義および解除
CLOCK	CK	–	6.6	エミュレータのCPUクロック時間の設定
DEVICE_TYPE	DE	–	6.7	エミュレータのデバイスタイプの選択
DISASSEMBLE	DA	○	–	逆アセンブル表示
ERASE	ER	○	–	コマンドウィンドウの内容のクリア
EVALUATE	EV	○	–	式の計算
FILE_LOAD	FL	○	–	オブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS	○	–	メモリ内容のファイルセーブ
FILE_VERIFY	FV	○	–	ファイル内容とメモリ内容の比較
GO	GO	○	–	ユーザープログラムの実行
GO_RESET	GR	○	–	リセットベクタからのユーザープログラムの実行
GO_TILL	GT	○	–	テンポラリブレークポイントまでのユーザプログラムの実行
HALT	HA	○	–	ユーザープログラムの停止
HELP	HE	○	–	コマンドラインまたはコマンドに対するヘルプ表示

表 6-1 HDIコマンドライン機能とマニュアルの対応表（つづき）

コマンド名	短縮形	HDI マニュアル	本 マニュアル	説明
INITIALISE	IN	○	-	プラットフォームの初期化
INTERRUPTS	IR	○	-	プラットフォームの割り込み処理の有効化／無効化（E6000エミュレータではサポートしません）
LOG	LO	○	-	ロギングファイルの操作
MAP_DISPLAY	MA	○	-	メモリマッピング情報の表示
MAP_SET	MS	-	6.8	メモリマッピングの設定
MEMORY_DISPLAY	MD	○	-	メモリ内容の表示
MEMORY_EDIT	ME	○	-	メモリ内容の変更
MEMORY_FILL	MF	○	-	指定データによるメモリ内容の一括変更
MEMORY_MOVE	MV	○	-	メモリブロックの移動
MEMORY_TEST	MT	○	-	メモリブロックのテスト
MODE	MO	-	6.9	CPUモードの設定と表示
QUIT	QU	○	-	HDIの終了
RADIX	RA	○	-	入力ラディックスの設定
REGISTER_DISPLAY	RD	○	-	CPUレジスタ値の表示
REGISTER_SET	RS	○	-	CPUレジスタ値の設定
RESET	RE	○	-	CPUのリセット
SLEEP	-	○	-	コマンド実行の遅延
STEP	ST	○	-	ステップ実行(命令単位またはソース行単位)
STEP_OVER	SO	○	-	ステップオーバー実行
STEP_RATE	SR	○	-	複数ステップ実行
STEP_OUT	SP	○	-	PC位置の関数を終了するまでのステップ実行
SUBMIT	SU	○	-	エミュレータコマンドファイルの実行
SYMBOL_ADD	SA	○	-	シンボルの追加
SYMBOL_CLEAR	SC	○	-	シンボルの削除
SYMBOL_LOAD	SL	○	-	シンボル情報ファイルのロード
SYMBOL_SAVE	SS	○	-	シンボル情報のファイルセーブ
SYMBOL_VIEW	SV	○	-	シンボルの表示
TEST_EMULATOR	TE	-	6.10	エミュレータハードウェアのテスト
TIMER	TI	-	6.11	実行時間測定タイマ分解能の表示、設定
TRACE	TR	○	-	トレース情報の表示
TRACE_ACQUISITION	TA	-	6.12	トレース取得情報の設定と表示
TRACE_COMPARE	TC	-	6.13	トレース情報の比較
TRACE_SAVE	TV	-	6.14	トレース情報の保存
TRACE_SEARCH	TS	-	6.15	トレース情報の検索
USER_SIGNALS	US	-	6.16	ユーザーシグナル情報の有効化／無効化
REFRESH	RF	-	6.17	メモリ関連ウィンドウの更新

6.1 BREAKPOINT / EVEVT

省略形： BP, EN

ブレークポイントを設定します。ブレークポイントには以下の３種類があり、それぞれ別のフォーマットが用意されています。

・ プログラムブレークポイント

・ アクセスブレークポイント

・ 範囲ブレークポイント

6.1.1 プログラムブレークポイント

フォーマット： bp program *address*
　　： bp p *address*

指定したアドレスにプログラムブレークポイントを設定します。

6.1.2 アクセスブレークポイント

フォーマット： bp access *address* [*options*]
　　： bp a *address* [*options*]

オプション：

```
<options>    = [<dataopts>] [read|write] [<signalopts>] [<busopts>] [<areaopts>]
[<actionopts>] [count <countval>] [delay <delayval>] [channel <channelno>]
<dataopts>   = data <data> [mask <mask>] [byte|word]
<signalopts> = signal ((1|2|3|4) (high|low))+
<busopts>    = bus (cpu | cpupre | sadata | sapre| cpumdata | cpumpre)+
<areaopts>   = area (io|iram|irom|lcdram)+
<actionopts> = action (trace | none | break|(timer (start|stop)))+
<channelno>  = 1..12
```

アクセスブレークポイントを設定すると、指定したアドレスを指定した方法でMCUがアクセスしたときに、ブレークが発生します。

6.1.3 範囲ブレークポイント

フォーマット： bp range [outside] <address low> <address hi> [<options>]

オプションは、アクセスブレークポイントのオプションと同じです。

範囲ブレークポイントを設定すると、指定したアドレス範囲または範囲外をMCUがアクセスしたときに、ブレークが発生します。

6.1.4 オプション

data <data> [mask <mask>] [byte | word]

データ比較を行ないます。マスクを指定すると、マスクで0にセットされているビットに対応するデータビットは、比較されません。

例：data h'20 mask h'fff0 word
　データバスの上位12ビットがh'002の場合、イベントが発生します。

このオプションを省略した場合、データ比較は行ないません。

signal ((1 | 2 | 3 | 4) (high | low)) +

指定した外部プローブ信号が指定した状態のとき、イベントが発生します。

例：signal 1 high 3 low
　外部プローブ信号1がハイレベルでかつ外部信号3がローレベルのとき、イベントが発生します。（その他の信号レベルはチェックしません）

このオプションを省略した場合、信号レベルは無視されます。

bus (cpu | cpupre | sadata | sapre| cpumdata | cpumpre) +

MCUのバスが指定した状態のとき、イベントが発生します。

表 6-1 MCUバスステータス

イベント	H8/3864シリーズ、H8/3887シリーズのステータス
cpu	CPUがアクティブモードの時のデータアクセスサイクルを示します。
cpupre	CPUがアクティブモードの時の命令プリフェッチサイクルを示します。
sadata	CPUがサブアクティブモードの時のデータアクセスサイクルを示します。
sapre	CPUがサブアクティブモードの時の命令プリフェッチサイクルを示します。
cpumdata	CPUが中速アクティブモードの時のデータアクセスサイクルを示します。
cpumpre	CPUが中速アクティブモードの時の命令プリフェッチサイクルを示します。

例：bus cpu cpupre
　バス状態が、プリフェッチまたはデータアクセスのとき、イベントが発生します。

このオプションを省略した場合、バス状態は無視されます。

area (io | iram | irom | lcdram)+

busオプションと同様に、指定した領域をアクセスしたとき、イベントが発生します。

例：area irom iram
　内蔵ROMまたは内蔵RAMアクセスのとき、イベントが発生します。

lcdramはH8/3864シリーズ、H8/3887シリーズのLCDRAM領域のアクセスを示します。

このオプションを省略した場合、アクセス領域は無視されます。

action (trace | none | break | (timer (start | stop)))+

イベントが検出されたときの動作を指定します。

デフォルトの動作は、ブレーク発生です。その他の動作として、イベント間実行時間測定の開始と停止を選択することができます。（タイマは１つしかありません）

count <countval>

イベント通過回数を設定します。（10進数）

delay <delayval>

イベントが検出されてから動作が始まるまでのディレイサイクルを、バスサイクル単位で指定します。（10進数）

channel 1..12

イベント検出システムチャネル番号を指定します。イベント発生の順序を指定する場合、このチャネル番号を参照して指定します。（詳細は、[6.5 **event_sequence**] を参照してください。）
チャネル1～8はイベント検出用、チャネル9～12は範囲検出用チャネルです。

使用例

en access 100	アクセスブレークポイントをアドレス100に設定
bp p 110	プログラムブレークポイントをアドレス110に設定
en access 100 data 55 byte	アクセスブレークポイントをアドレス100、データ55に設定
bp range 12 45	範囲ブレークポイントをアドレス12からアドレス45の範囲に設定
bp range outside 60 89	範囲ブレークポイントを、アドレス60からアドレス89以外の範囲がアクセスされたときにブレークするように設定
bp a 200 read	アクセスブレークポイントをアドレス200のリードサイクルに設定
bp a 500 write	アクセスブレークポイントをアドレス500のライトサイクルに設定
bp a 100 read channel 8	アクセスブレークポイントをアドレス100のリードサイクルに設定。チャネル8を指定しているので、イベントが発生した時に外部プローブからトリガ信号が出力されます。

6.2 BREAKPOINT_CLEAR / EVENT_CLEAR

省略形： BC, EC

ユーザが設定したブレークポイントを削除します。

表 6-1 ブレークポイント削除の指定

種類（キーワード）	説明（ブレークポイントの種類）
program <address>	指定したプログラムブレークポイントの削除
access <address> <options>	指定したアクセスブレークポイントの削除
range <address> <options>	指定した範囲ブレークポイントの削除
all	全ブレークポイントの削除
all trace	全トレースイベントの削除
channel 1～12	指定したチャネル番号のイベントの削除

オプションは、BREAKPOINT / EVENTコマンドと同様です。イベントを特定するのに必要最小限のオプションを指定してください。

使用例

bc p 256	アドレス256のプログラムブレークポイントを削除
event_clear chan 5	チャネル番号を指定してイベントを削除
bc all	全ブレークポイントを削除

6.3 BREAKPOINT_DISPLAY / EVENT_DISPLAY

省略形： BD, ED

現在設定されているブレークポイントを表示します。トレースイベントは “trace” と表示します。

使用例

bd　　設定されている全ブレークポイントを表示します。

6.4 BREAKPOINT_ENABLE / EVENT_ENABLE

省略形： BE, EE

指定したブレークポイントまたは全ブレークポイントを有効／無効にします。

表 6-1 ブレークポイント有効／無効

パラメータ	種類（キーワード）	説明
1	true	ブレークポイントを有効にします。
	false	ブレークポイントを無効にします。
2	all	全ブレークポイント
	program <address>	プログラムブレークポイント
	access <address> <options>	アクセスブレークポイント
	range <address1> <address 2> <options>	範囲ブレークポイント
	channel 1..12	指定したチャネル番号のイベントを有効または無効にします。

BREAKPOINT / EVENTコマンドで設定するのと同様に、オプションを指定して、イベントを正確に特定することができます。

使用例

be true all	全ブレークポイントを有効
be false all	全ブレークポイントを無効
be false p 256	アドレス256のプログラムブレークポイントを無効
be true access 12	アドレス12のアクセスブレークポイントを有効
be false chan 1	チャネル1のイベント検出を無効

6.5 BREAKPOINT_SEQUENCE / EVENT_SEQUENCE

省略形：BS, ES

フォーマット

bs <channel> [armed_by [not] <chan1> <chan2> ...]
[armed_by off]
[reset_by <chan1> <chan2> ...]
[reset_by off]

別のイベントの前提条件となるeventまたはリセット条件となるイベントを指定します。

使用例

bs 1 armed_by 2 3

イベント2または3は、イベント1発生の前提条件となります。設定できる番号は、イベント検出用のチャネル1～8です。チャネル番号は、eventコマンドのチャネルオプションで、イベントに割り当てられます。

bs 2 reset_by 4
イベント2は、イベント4が発生したときにリセットされます。

キーワードOffは、他のイベントによる前提／リセット条件を無効にします。これにより、各イベントは、独立して検出・発生します。

6.6 CLOCK

省略形： CK

システムクロック(ϕ)およびサブシステムクロック(ϕw)の指定、表示を行ないます。パラメータを省略すると、現在のシステムクロックを表示します。システムクロックを変更すると、E6000エミュレータはリセットされます。H8/3864シリーズ、H8/3887シリーズのシステムクロック(ϕ)は、(OSC1,OSC2) 入力の、1/2の周波数になります。

表 6-1 クロックパラメータ

パラメータ	種類（キーワード）	説明（エミュレーションクロックϕ）
1	05	0.5 MHz内部クロック
	1	1MHz内部クロック
	2	2MHz内部クロック
	t2	ターゲット/2
2	sub 32k	32.768kHz 内部サブシステムクロック(ϕw)
	sub 38k	38.4kHz 内部サブシステムクロック(ϕw)
	sub 307k	307.2kHz 内部サブシステムクロック(ϕw)
	sub t	ターゲットサブクロック

ユーザシステムのクロックは、ユーザシステムのVccが供給されているときだけ選択可能です。

ck　　現在のエミュレーションクロックを表示

ck 2 sub 32k　　システムクロックとして2MHz内部クロック、サブシステムクロックとして32.768kHz内部クロックを指定

ターゲットMCUであるH8/3864シリーズ、H8/3887シリーズは、内部サブシステムクロック(ϕw)として、307.2kHzを選択することはできません。本エミュレータでサブシステムクロック(32.768kHzおよび38.4kHz)動作中にブレークが検出された場合、そのクロックにて動作、表示反応してしまうため、サブシステムクロック評価時は、38.4kHzの8倍周波数307.2kHzを選択していただき評価することをお勧めします。

6.7 DEVICE_TYPE

省略形：DE

ターゲットMCUを指定、または現在の設定を表示します。

使用例

de	MCUタイプの表示
de h8/3864	H8/3864を指定

6.8 MAP_SET

省略形： MS

エミュレーションメモリマップを設定します。

フォーマット

ms <start> <end> (internal | on-chip) (none | read-only | guarded)

使用例

ms 8000 F73F internal none　　アドレス8000からF73Fに内部のアクセス可能メモリを割り付けます

注：on-chipメモリ（例　内蔵RAM,内蔵ROM,I/Oまたはリザーブエリア）は、デバイス内蔵のメモリです。基本的にはメモリマップの属性変更はできません。
ただし、H'FF80～H'FF8Fを除くリザーブエリアのみInternalを指定することでエミュレーションメモリの設定が可能です。

6.9 MODE

省略形：MO

MCUモードの設定、表示を行ないます。

表 6-1 MCUモードのパラメータ

パラメータ	種類（キーワード）	説明（モード）
1	3	3.（シングルチップモード）

H8/3864シリーズ、H8/3887シリーズではモード3に固定です。

使用例

mode　　現在のモードを表示

mode 3　　モード3に設定、メモリマップを再設定

6.10 TEST_EMULATOR

省略形： TE

E6000エミュレータのハードウェアとエミュレーションメモリ領域のテストを行ないます。本コマンド実行後は、E6000エミュレータシステムを初期化してください。

使用例

te　E6000エミュレータのテスト

6.11 TIMER

省略形： TI

実行時間測定タイマの分解能の表示、変更を行ないます。これにより、実行時間測定およびイベント間実行時間測定のタイマの分解能が設定できます。

表 6-1 タイマコマンド

コマンド	説明
ti	タイマ分解能の表示
ti <timer resolution>	タイマ分解能の設定

タイマ分解能： 20ns、125ns、250ns、500ns、1μs、2μs、4μs、8μs、16μs

使用例

ti 20　　タイマ分解能を20nsに設定

ti 250ns　タイマ分解能を250nsに設定

ti 8　　タイマ分解能を8μsに設定

ti 16us　タイマ分解能を16μsに設定

6.12 TRACE_ACQUISITION

省略形： TA

トレース取得オプションの設定、表示を行ないます。

フォーマット

```
TA [<suppress>] [<freetrace>] [<timestamp>] [<stop>] [<stopdelay>]
   [<range>]

<suppress>  = suppress dtu (true|false)      (H8/3864シリーズ、H8/3887シリーズでは
                                              使用できません。)
<freetrace> = freetrace (true|false)
<timestamp> = timestamp (disable | 125ns | 250ns | 500ns | 1us | 2us |
                                          4us | 8us | 16us | 100us )
<stop>      = stop ( disable | event <1～12> )
<stopdelay> = stopdelay ( disable | event <1～12> [count <count>] )
<range>     = range <1～4> ( disable |
                ptop <startaddr> <stopaddr> [cyclic] |
                range <1～12> |
                event <1～8> <1～8> [cyclic] )
<default>   = default
```

使用例

ta	全トレース制御オプションを表示
ta stop event 1 2	チャネル１もしくは２のどちらかのイベントが発生したときにトレース取得を止める
ta stopdelay event 1 2 count 100	チャネル１もしくは２のどちらかのイベントが発生したときに100バスサイクルトレース取得後にトレース取得を止める
ta timestamp 500ns	トレースのタイムスタンプ機能を有効にし、タイムスタンプ機能の分解能を500nSに設定
ta range 2 event 4 5 cyclic	イベント４でトレース開始し、イベント５でトレース停止する条件をトレース制御２に登録する、また、サイクル指定があるので条件が成立する毎にトレースを取得する

6.13 TRACE_COMPARE

省略形：TC

[**trace_save**] で保存したトレースファイルと現在のトレース結果を比較します。

trace_compare <filename>

6.14 TRACE_SAVE

省略形： TV

トレースした情報を、バイナリ形式のファイルに保存します。保存したデータは、別のトレース結果と［**trace_compare**］コマンドで比較することができます。

trace_save <filename>

6.15 TRACE_SEARCH

省略形： TS

トレース結果を検索します。 [**trace find**] ダイアログボックスと同じ方法でトレース結果を検索できます。

フォーマット

```
TS [<address>][<dataopts>] [<signalopts>] [<busopts>] [<areaopts>]
   [<directionopts>] [<timestampopts>] [<fromopts>]

<address>       = address <address> [to <address>]
<dataopts>      = data <data> [mask <mask>] [byte|word]
<signalopts>    = signal <sig><sig><sig><sig>
      <sig>     = (1|0|x)          1 = high, 0 = low, x = don't care
<busopts>       = bus (cpu | cpupre | sadata | sapre| cpumdata| cpumpre )+
<areaopts>      = area ( io | iram | irom | lcdram )+
<directionopts> = dir (read | write | either)
<timestampopts> = time <start> [ to <stop>]
      <start> and <stop> should be in format 0s:0000.000
<fromopts>      = from <record>
```

使用例

ts address 104 data 55aa w	アドレスが104番地、データが55aaのワードアクセスのトレースサイクルを検索する
ts area irom	ROMエリアアクセスのトレースサイクルを検索する

6.16 USER_SIGNALS

省略形： US

ユーザ信号（Reset）の入力を有効／無効にします。パラメータを省略すると、Reset信号の有効／無効状態を表示します。

表 6-1 ユーザ信号コマンド

種類	説明
us	ユーザ信号状態を表示
us enable reset	指定された信号を有効
us disable reset	指定された信号を無効

6.17 REFRESH

省略形： RF

メモリ関連ウィンドウを更新します。

7 故障解析

本章では、E6000エミュレータ用テストプログラムによる故障解析の手順について示します。

7.1 テストプログラムを実行するためのシステムセットアップ

(1) テストプログラムを実行するためには、以下に示す機器が必要です。なお、本テストプログラムの実行にはユーザシステムインタフェースケーブルおよびユーザシステムは不要です。

・H8/3864シリーズ、H8/3887シリーズ用E6000エミュレータ(HS3880EPI60H)
・E6000 PCインタフェースボード(HS6000EII01H)
・ISAバス仕様ホストコンピュータ(MS-DOS環境)

(2) ホストPCにE6000 PCインタフェースボードを挿入し、付属のPCインタフェースケーブルを接続してください。

(3) PCインタフェースケーブルをH8/3864シリーズ、H8/3887シリーズ用E6000エミュレータ本体に接続してください。

(4) H8/3864シリーズ、H8/3887シリーズ用E6000エミュレータ本体に、付属のACアダプタを接続してください。

(5) ホストPCを起動し、MS-DOSのコマンド入力待ち状態にしてください。

(6) H8/3864シリーズ、H8/3887シリーズ用E6000エミュレータ本体の電源をオンにしてください。

7.2 テストプログラムによる故障解析

H8/3864シリーズ、H8/3887シリーズ用E6000エミュレータに添付されているテストプログラム用フロッピーディスク(HS3880EVI60SF)をPCに挿入し、カレントディレクトリをA:に移動した後、

```
>A:TM3880 (RET)
```

と入力すると直ちにテストプログラムが起動します。

テストが実行されているときに表示されるメッセージとテスト内容は次の様になります。テストはNo.1からNo.11までです。

```
E6000 H8/3880 EMULATION BOARD Tests Vn.m
      Hitachi Ltd (1998)
```

テストプログラムのスタートメッセージです。Vn.mはバージョン番号です。

```
Searching for interface card .......OK,
     card at H'd0000000
```

ホストPCにPCインタフェースボードが正しく接続されていることを示します。また、このときのアドレスを表示します。値はアドレス設定値によって変わります。

```
Checking emulator is connected ......OK
```

ホストPCとE6000エミュレータが正しく接続されていることを示します。

```
  Emulator Board Information:
    Main Board ID          H' 1
```

E6000エミュレータ（下基板）のID番号で、常に1を示します。

```
    Emulation Board ID     H' A
```

E6000エミュレータ（上基板）のID番号で、常にAを示します。

```
    Revision               H' x
```

E6000エミュレータ（上基板）のレビジョン番号をxで示します。

```
Downloading firmware  ........
```

テスト用のプログラムをロードしていることを示します。

```
01) Testing Register :
    IDR Register ....................OK
    BOC control register ............OK
    MODERE register .................OK
    MODERO register .................OK
```

E6000エミュレータ上のレジスタのチェック結果（正常終了）を示します。

```
02) Testing Dual-Port RAM :
    Decode Test .....................OK
    Marching Test ...................OK
```

E6000エミュレータ上のDual-Port RAMのデコードテスト、マーチングテストチェック結果（正常終了）を示します。

```
03) Testing Firmware RAM :
    Decode Test.    page H'7xx of range H'700 - H'71f ..
    Decode Test.    page range H'700 - H'71f ........OK

    Marching Test.  page H'7xx of range H'700 - H'71f ..
    Marching Test.  page range H'700 - H'71f ........OK

Downloading firmware .......

04) Testing Trace RAM :
    Decode Test.    page H'0xx of range H'000 - H'04f ..
    Decode Test.    page range H'000 - H'04f ........OK

    Marching Test. page H'0xx of range H'000 - H'04f ..
    Marching Test. page range H'000 - H'04f ........OK

05) Testing Mapping RAM :
    Decode Test.    page H'2xx of range H'200 - H'21f ..
    Decode Test.    page range H'200 - H'21f ........OK

    Marching Test. page H'2xx of range H'200 - H'21f ..
    Marching Test. page range H'200 - H'21f .........OK

06) Testing Emulation RAM :
    Decode Test ............................OK (注)
    Marching Test ..........................OK
```

E6000エミュレータ上のFirm RAMのデコードテストチェック結果（正常終了）を示します。

E6000エミュレータ上のFirm RAMのマーチングテストチェック結果（正常終了）を示します。

テスト用のプログラムをロードしていることを示します。

E6000エミュレータ上のTrace RAMのデコードテストチェック結果（正常終了）を示します。

E6000エミュレータ上のTrace RAMのマーチングテストチェック結果（正常終了）を示します。

E6000エミュレータ上のMapping RAMのデコードテストチェック結果（正常終了）を示します。

E6000エミュレータ上のMapping RAMのマーチングテストチェック結果（正常終了）を示します。

エミュレーションメモリのデコードテスト、マーチングテストチェック結果（正常終了）を示します。

注: Decode Testのテスト結果表示には時間(CPU 133MHzのホストコンピュータで約2分)を要します。

07) Testing STEP Operation : Single Step Operation .....OK Step Into Operation .....OK	ステップ実行制御回路のチェック結果(正常終了）を示します。
08) Testing Key Break : Key Break .....OK	強制ブレーク制御回路のチェック結果(正常終了）を示します。
09) Testing Emulation RAM Hardware Break : GRD Break .....OK WPT Break .....OK IN Rom .....OK EML RAM Write .....OK	不当アクセスブレーク制御回路のチェック結果（正常終了）を示します。
10) Testing Go Reset .....OK	Go Reset制御回路のチェック結果（正常終了）を示します。
11) Testing Double Stack, I/O Stack Test : Setting up, please wait ...	テスト用のプログラムをロードしていることを示します。
Double Stack .....OK I/O Stack .....OK	スタック制御回路のチェック結果(正常終了）を示します。
0 total errors	エラー発生数の合計を示します。
Tests passed, emulator functioning correctly	テストプログラムにより正常動作が確認されたことを示します。

本テストプログラムは不具合を検出するとERRORを表示してプログラムの実行を中止します。この場合、エミュレータハードウエアの故障が考えられます。発生したエラー内容の詳細を当社の購入営業担当までご連絡ください。

H8/3864 シリーズ、H8/3887 シリーズ E6000 エミュレータ
ユーザーズマニュアル

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

ADJ-702-254