

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事事務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ご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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注意ください。

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とは、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。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。

H8S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

HS2000EPI61HJ(B)

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（４）オプションメモリボード

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 H8S 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使用してください。この目的以外に当エミュレータを使用することを堅くお断りします。

使用制限：

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

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

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

製品の変更について：

日立は、当エミュレータのデザイン、機能および性能を絶えず改良する方針をとっています。

したがって、予告なく仕様、デザイン、およびユーザーズマニュアルを変更することがあります。

エミュレータを使用する人は：

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

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

保証の範囲：

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せん。

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。

著作権所有：

このユーザ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、すべての権利は日立に帰属しています。このユーザ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、日立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：

このユーザズマニュアルの一部の図は、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、このユーザズマニュアルと当エミュレータに貼付されている警告がすべてではありません。お客様の責任で、当エミュレータを正しく安全に使用してください。

安全事項

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

シグナル・ワードの定義



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

危険

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

警告

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

注意

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

注意

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

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

警告

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

まえがき

本書について

本書は、H8S 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 H8S シリーズ共通マニュアルです。なお、各製品個別の仕様については製品添付の「補足説明書」に記載しています。

「第 1 章 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。

「第 2 章 セットアップ」は、E6000 エミュレータのセットアップ方法と、HDI に接続する方法について記載します。

「第 3 章 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。

「第 4 章 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徴を紹介します。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。

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

- 補足説明書
- 日立デバッグインタフェースユーザズマニュアル
- 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（本共通ユーザズマニュアルでは、以下のいずれかを指します。）
 - ISAバスインタフェースボード(HS6000EII01HJ)
 - PCIバスインタフェースボード取扱い説明書(HS6000EIC01HJ, HS6000EIC02HJ)
 - PCMCIAインタフェースカード取扱い説明書(HS6000EIP01HJ)
 - LANアダプタ取扱い説明書(HS6000ELN01HJ)
- オプションメモリボード取扱い説明書
 - 1M SIMMメモリボード(HS6000EMS11HJ)
 - 4M SIMMメモリボード(HS6000EMS12HJ)

Windows®, Windows® 95, Windows® 98, Windows NT® 4.0, Windows® 2000 および MS-DOS® は米国マイクロソフト社の商標です。

IBM PC は米国 IBM 社の商標です。

本マニュアルは動作環境を IBM PC 上の英語版 Microsoft® Windows® 98 として記述しています。

目次

第 1 章 はじめに

1.1	デバッグの特長	1-1
1.1.1	ブレイクポイント	1-1
1.1.2	トレース	1-1
1.1.3	実行時間測定	1-2
1.1.4	パフォーマンスアナリシス	1-2
1.1.5	バスモニタ	1-2
1.2	イベント検出システム (CES: Complex Event System)	1-2
1.2.1	イベントチャンネル	1-2
1.2.2	範囲チャンネル	1-3
1.2.3	ブレイク	1-3
1.2.4	イベント間実行時間測定	1-3
1.3	ハードウェアの特長	1-3
1.3.1	メモリ	1-3
1.3.2	エミュレーションクロック	1-4
1.3.3	外部プローブ	1-4
1.3.4	使用環境条件	1-4
1.3.5	外形寸法と質量	1-5

第 2 章 セットアップ

2.1	パッケージ内容	2-1
2.2	Windows® 95、98 での PC インタフェースボードのセットアップ	2-2
2.2.1	PC インタフェースボードのセットアップ	2-2
2.2.2	CONFIG.SYS の変更	2-4
2.2.3	SYSTEM.INI の変更	2-5
2.3	Windows NT® 4.0 での PC インタフェースボードのセットアップ	2-5
2.4	HDI のインストール	2-7
2.5	トラブルシューティング	2-7
2.5.1	接続不良	2-7
2.5.2	通信不良	2-7

第 3 章 ハードウェア

3.1	ユーザシステムへの接続	3-1
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	3-2
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	3-3
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	3-3
3.2	電源供給	3-3
3.2.1	AC 電源アダプタ	3-3
3.2.2	極性	3-3

3.2.3	電源モニタ回路	3-4
3.3	オプションメモリボード	3-4
3.3.1	オプションメモリボードの構成	3-4
3.4	ハードウェアインタフェース	3-4
3.4.1	信号保護	3-4
3.4.2	ユーザインタフェース回路	3-4
3.4.3	クロック発振器	3-6
3.4.4	外部プローブ 1 (EXT1) / トリガ出力	3-7
3.4.5	外部プローブ 2 (EXT2) / トリガ出力	3-7
3.4.6	電源フォロワ回路	3-8
3.5	MCU と E6000 エミュレータの相違点	3-9
3.5.1	A/D コンバータ、D/A コンバータ	3-10
第 4 章 チュートリアル		
4.1	はじめに	4-1
4.2	HDI の起動	4-1
4.2.1	ターゲットプラットフォームの選択	4-2
4.3	E6000 エミュレータのセットアップ	4-4
4.3.1	プラットフォームの構成	4-4
4.3.2	メモリマッピング	4-5
4.4	チュートリアルプログラムのダウンロード	4-7
4.4.1	オブジェクトファイルのダウンロード	4-7
4.4.2	プログラムリストの表示	4-8
4.5	ブレークポイントの使い方	4-10
4.5.1	PC Break の設定	4-10
4.5.2	プログラムの実行	4-11
4.5.3	レジスタ内容の参照	4-12
4.5.4	ブレークポイントの確認	4-13
4.6	メモリと変数の表示	4-14
4.6.1	メモリを表示する	4-14
4.6.2	変数を表示する	4-14
4.7	プログラムのステップ実行	4-16
4.7.1	シングルステップ	4-17
4.7.2	関数全体のステップ実行	4-19
4.7.3	ローカル変数の表示	4-20
4.8	イベント検出システム (Complex Event System) の使用方法	4-22
4.8.1	イベント検出システムによる Event の設定	4-22
4.9	トレースバッファの使い方	4-24
4.9.1	トレースバッファの表示	4-24
4.9.2	トレースフィルタの設定	4-25
4.10	プログラム実行時間効率の計測	4-27
4.10.1	計測条件の選択	4-27
4.10.2	計測結果の表示	4-29

4.11	バスモニタ	4-30
4.12	スタックトレース機能	4-31
4.13	セッションの保存	4-32
4.14	さてつぎは?	4-32
付録 A	コマンダー一覧表	付録-1

図目次

図 2.1	Computer Properties ダイアログボックス (設定前)	2-2
図 2.2	Edit Resource Setting ダイアログボックス	2-3
図 2.3	Computer Properties ダイアログボックス (設定後)	2-4
図 2.4	エラーメッセージ (1)	2-7
図 2.5	エラーメッセージ (2)	2-7
図 3.1	E6000 コネクタの位置	3-1
図 3.2	ユーザシステムインタフェースケーブルの接続	3-2
図 3.3	ネジの締め付け順序	3-2
図 3.4	ユーザシステムインタフェースケーブル外観図	3-3
図 3.5	電源プラグ	3-3
図 3.6	ユーザインタフェース信号回路	3-5
図 3.7	モード端子, WAIT, NMI 等信号回路	3-5
図 3.8	RESET 信号回路	3-5
図 3.9	アナログポート制御信号回路	3-6
図 3.10	IRQ0-IRQ7 信号回路	3-6
図 3.11	外部プローブ 1 コネクタ	3-7
図 3.12	外部プローブ 1 インタフェース回路	3-7
図 3.13	外部プローブ 2 コネクタ	3-7
図 3.14	ユーザシステムと E6000 との Vcc の関係 (Vcc=3.3V の例)	3-9
図 4.1	HDI 起動メニュー	4-2
図 4.2	プラットフォームの選択	4-2
図 4.3	HDI ウィンドウ	4-3
図 4.4	Configuration ダイアログボックス	4-4
図 4.5	Memory Mapping ダイアログボックス	4-5
図 4.6	Edit Memory Mapping ダイアログボックス	4-6
図 4.7	System Status ウィンドウ (Memory シート)	4-7
図 4.8	Open ダイアログボックス (オブジェクトファイルの選択)	4-8
図 4.9	HDI ダイアログボックス	4-8
図 4.10	Open ダイアログボックス (ソースファイルの選択)	4-9
図 4.11	ソースプログラム画面	4-9
図 4.12	ブレークポイント (PC Break) の設定	4-10
図 4.13	ステートメントの強調表示	4-11
図 4.14	System Status ウィンドウ (Platform シート)	4-12
図 4.15	Registers ウィンドウ	4-12
図 4.16	Register ダイアログボックス	4-13
図 4.17	Breakpoints ウィンドウ	4-13
図 4.18	Open Memory Window の設定	4-14
図 4.19	Memory ウィンドウ(Byte)	4-14
図 4.20	Watch ウィンドウ (変数追加後)	4-15
図 4.21	Watch ウィンドウ (シンボル拡張)	4-15
図 4.22	Add Watch ダイアログボックス	4-16
図 4.23	Watch ウィンドウ (変数の追加)	4-16
図 4.24	Reset Go 実行後の Program ウィンドウ	4-17
図 4.25	Step In 実行後の Program ウィンドウ (1)	4-18
図 4.26	Step Out 実行後の Program ウィンドウ	4-18

図 4.27	Step In 実行後の Program ウィンドウ (2)	4-19
図 4.28	Step Over 実行後の Program ウィンドウ	4-20
図 4.29	Locals ウィンドウ	4-20
図 4.30	Locals ウィンドウ (変数"min"内容変更後)	4-21
図 4.31	Locals ウィンドウ (配列変数"a"ソート後)	4-21
図 4.32	ブレークポイントの追加 (アドレス指定)	4-22
図 4.33	ブレークポイントの追加 (回数指定)	4-23
図 4.34	Breakpoints ウィンドウ (追加後)	4-23
図 4.35	Event ブレークポイントによるプログラムの停止	4-24
図 4.36	Trace ウィンドウ	4-24
図 4.37	Trace Filter ダイアログボックス	4-25
図 4.38	Bus / Area の設定	4-26
図 4.39	Trace ウィンドウ (トレースフィルタ指定)	4-26
図 4.40	計測条件の選択	4-27
図 4.41	計測条件の表示	4-28
図 4.42	計測結果の表示 (1)	4-29
図 4.43	計測結果の表示 (2)	4-29
図 4.44	バスモニタの機能選択ダイアログボックス	4-30
図 4.45	RAM モニタのアドレス設定ダイアログボックス	4-30
図 4.46	RAM モニタの表示ウィンドウ	4-31
図 4.47	Stack Trace ウィンドウ	4-31

表目次

表 1.1	メモリタイプの定義.....	1-3
表 1.2	E6000 使用環境条件.....	1-4
表 1.3	外形寸法および質量.....	1-5
表 2.1	PC インタフェースボードのメモリ領域図.....	2-2
表 3.1	MCU と E6000 エミュレータの相違.....	3-9
表 4.1	コンフィグレーションオプションの設定例.....	4-5
表 4.2	メモリタイプの定義.....	4-5
表 4.3	メモリタイプオプション.....	4-6
表 4.4	プログラムステップオプション.....	4-16
表 A.1	コマンド一覧表.....	1

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 H8S シリーズマイクロコントロ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 HDI とともに動作します。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ンネルと範囲チャンネルを使って、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るので、ユーザシステム上の ROM には 1 箇所のみ (On chip Break) 設定できます。

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用し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. はじめに

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, 1 μ s, 2 μ s, 4 μ s, 8 μ s, 16 μ s

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

1.1.4 パフォーマンスアナリシス

E6000 エミュレータは、プログラム実行時間効率の計測機能を備えています。指定した範囲のプログラムの実行効率をヒストグラムまたはパーセントで表示することができます。タイマーの分解能は 20ns、40ns、160ns のいずれかの値に設定できます。また、指定した範囲のプログラムの実行回数（1～65535）を測定することができます。

1.1.5 バスモニタ

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

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イクポイントを使用してその問題を調べ上げるのは、非常に困難です。

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

イベント検出システムは、E6000 エミュレータのトレース、ブレイク、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
- リード、ライトまたは両方とも
- マスク条件指定付きデータ
- MCU アクセスタイプ（DMAC、命令プリフェッチなど）
- MCU アクセス領域（内蔵 ROM、内蔵 RAM など）
- 4 つの外部プローブ信号の値
- イベントの発生回数
- イベントの発生後のディレイサイクル数

また、最大 8 イベントがシーケンスで組み合わせられます。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イ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
- リード、ライトまたは両方とも
- マスク条件指定付きデータ
- MCU アクセスタイプ (DMAC、命令プリフェッチなど)
- MCU アクセス領域 (内蔵 ROM、内蔵 RAM など)
- 4 つの外部プローブ信号の値
- イベントの発生後のディレイサイクル数

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

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 用代替メモリを標準装備しています。ただし、選択したデバイスやモードにおいて内蔵 ROM や内蔵 RAM が存在しない場合、これらのエミュレーションメモリは使用できません。従って、外部アドレス空間にプログラムやデータを配置して E6000 エミュレータ単体でデバッグする場合は、別売で用意しているオプションメモリボードを使用してください。

エミュレーションメモリは、MCU アドレス空間の任意のサイズのメモリブロックに、64 バイト単位で割り付けできます。各メモリブロックは、[**Configure Map...**] 機能を使用して、ユーザシステム上のメモリまたは E6000 エミュレータ上のオプションメモリボードのいずれかに指定でき、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以下に示します。

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
ユーザ	ユーザシステム上のメモリ
エミュレータ	オプションメモリボードのメモリ

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

1. はじめに

ンドウにすぐに反映されます。なお、プログラム実行中のメモリ内容変更に要する時間は以下のようになります。

(1) MCU 内蔵 ROM、RAM およびオプションメモリボードのメモリ

ユーザプログラムはブレイクすることなく、メモリバスを一時エミュレータ側に切替えてメモリの変更を行いません。メモリリード/ライトいずれの場合でも HDI が最大 256 バイト分のメモリ内容をバッファに取り込む操作があるため、メモリバスがエミュレータに占有される時間は最大約 80 μ s (25MHz、内蔵 ROM) です。

(2) MCU 内蔵 I/O、DTCRAM およびユーザシステム上のメモリ

ユーザプログラム実行をブレイクし、エミュレータがメモリの変更を行いません。上記(1)同様、最大 256 バイト分のメモリアクセスがあるため、プログラムが停止する時間は最大約 2ms (25MHz、エミュレーションメモリ) です。

1.3.2 エミュレーションクロック

エミュレーションクロックは E6000 エミュレータ内蔵クロックとターゲットクロックのいずれかの周波数に設定できます。選択できるクロックは H8S シリーズ各々で異なりますので、添付の「補足説明書」を参照してください。

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イクもしくはトレースに使用するために、E6000 エミュレータには外部プローブ 1 (EXT1) および外部プローブ 2 (EXT2) が接続できます。外部プローブ 1 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。また、外部プローブ 2 の信号はバスモニタ機能のトリガ設定 (1~4) の条件が一致した場合にハイレベルを出力しますので、オシロスコープ等のトリガ条件に使用できます。

1.3.4 使用環境条件

表 1.2 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 \pm 5% 電流 : Max. 6A
5	ユーザ Vcc (Uvcc)	電圧 : 2.7 ~ 5.5V の範囲で各 MCU の電源仕様に従う
6	AC 入力電源	電圧 : AC100 ~ 240V 周波数 : 50 / 60Hz 消費電力 : 61 ~ 70VA

1.3.5 外形寸法と質量

表 1.3 外形寸法および質量

項番	項目	仕様
1	外形寸法	219 × 170 × 54 (mm)
2	質量	1000 (g)

1. はじめに

2. セットアップ

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 HS6000EH01H）のセットアップ
- E6000 エミュレータのセットアップ
- HDI のインストールとシステムの動作チェック

PC Card（PCMCIA）などの他のホストシステムインタフェースボードをご使用の場合は、各製品に添付のマニュアルをご覧ください。

E6000 エミュレータは、PC インタフェースボードを使って 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用する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。

Windows® 95, 98 をご使用の方は、「2.2 Windows® 95, 98 での PC インタフェースボードのセットアップ」を、Windows NT® をご使用の方は、「2.3 Windows NT® 4.0 での PC インタフェースボードのセットアップ」をご覧ください。

【注】 Windows® 2000 では PC インタフェースボードはサポートしていません。

2.1 パッケージ内容

E6000 エミュレータには、以下の構成部品が梱包されています。

- E6000 エミュレータ本体
- AC 電源アダプタ 5V 6A
- CD-R（HDI、マニュアル等）
- 外部プローブ 1
- 外部プローブ 2
- E6000 エミュレータ用 Hitachi Debugging Interface セットアップガイド

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

2.2 Windows® 95、98 での PC インタフェースボードのセットアップ

2.2.1 PC インタフェースボードのセットアップ

- Windows® 95、98 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 ダイアログボックスが表示されます。

- Device Manager パネルの [**Computer**] アイコンをダブルクリックし、Computer Properties ダイアログボックスを開きます。
- 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。

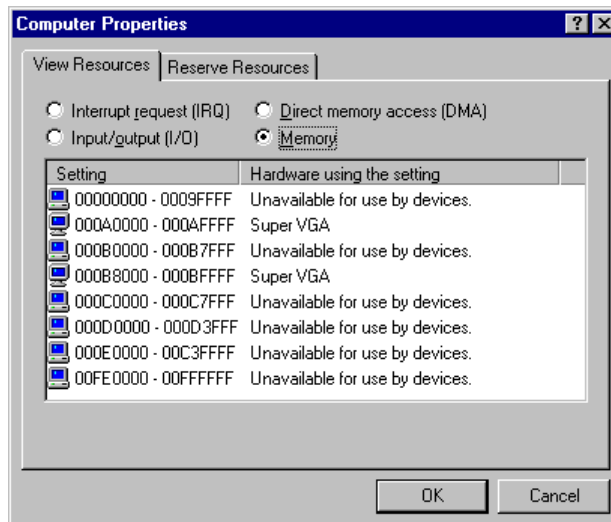


図2.1 Computer Properties ダイアログボックス (設定前)

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 ダイアログボックス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図

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'CEFFF	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

メモリ領域	スイッチ
H'E0000 ~ H'E3FFF	8
H'E4000 ~ H'E7FFF	9
H'E8000 ~ H'EBFFF	A
H'EC000 ~ H'FFFFFF	B

選択したメモリ領域を Windows[®] 95、98 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 ダイアログボックスが表示されます。

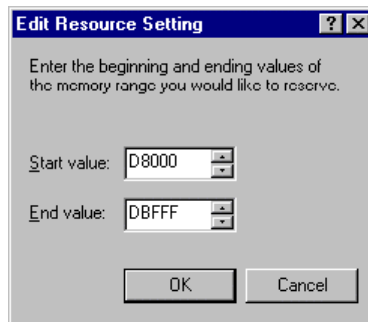


図2.2 Edit Resource Setting ダイアログボックス

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。

小型のマイナスインプラグを使って、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 ダイアログボックス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。

2. セットアップ

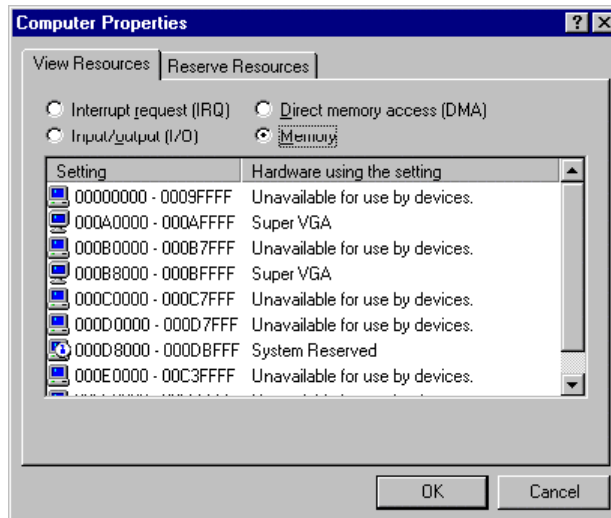


図2.3 Computer Properties ダイアログボックス (設定後)

2.2.2 CONFIG.SYS の変更

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ふことを防止します。

- [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。

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る場合、その中で EMM386.EXE を使用していない場合は、「2.2.3 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。

2.2.3 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しません。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。

2.3 Windows NT[®] 4.0 での PC インタフェースボードのセットアップ

PC インタフェースボードは ISA バススロットを使用しますので、ISA バススロットをサポートしていない PC では使用することはできません。

ISA ボードのインストール方法については、お使いの PC に付属のマニュアルを参照してください。ここでは一般的な方法を述べます。

(1) Windows NT[®]実行

- Start/Programs/Administrative Tools (Common)/Windows NT Diagnostics を実行してください。
- 'Resource' タブの'Memory'ボタンをクリックし、使用されている上位メモリ領域を以下に記録してください。

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

- Windows NT[®]をシャットダウンしてください。

2. セットアップ

(2) PC をセットアップモードで起動

セットアップモードについては、お使いの PC に付属のマニュアルを参照してください。

- 使用されている上位メモリ領域をチェックしてください。

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

(これは、上で得た Windows NT®の値と同じはずです)

- PC インタフェースボードの設定を登録してください。使用するメモリ領域は、他のデバイスにいつも使用されず、かつ PC インタフェースボードのスイッチ位置の一つと同じにする必要があります。

スイッチ位置：

#	Start	End	#	Start	End	#	Start	End
0	C0000	C3FFF	4	D0000	D3FFF	8	E0000	E3FFF
1	C4000	C7FFF	5	D4000	D7FFF	9	E4000	E7FFF
2	C8000	CBFFF	6	D8000	DBFFF	A	E8000	EBFFF
3	CC000	FFFFF	7	DC000	FFFFF	B	EC000	FFFFF

ご使用の PC に Intel P&P BIOS ディスクが付属の場合は以下の手順で設定します。

- Intel P&P BIOS ディスクで PC を起動します。
- 'View/System Resources' で使用している上位メモリ領域をチェックします。
- 'Configure/Add Card/Others...' で 'Unlisted Card' を追加します。
- .CFG ファイルがありませんので、次のダイアログボックスでは No と答えます。
- 'Configure Unlisted Card' ダイアログボックスで、'Memory [hex]' リストボックスに移動します。
- 'Add Memory...' ボタンを使用し、'Specify Memory' ダイアログボックスを表示します。
- 他のデバイスにいつも使用されず、かつ PC インタフェースボードのスイッチ位置の一つと同じメモリ領域を入力します。

- ファイルをセーブします。
- 終了します。
- 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。

2.4 HDI のインストール

HDI のインストールについては、添付の「E6000 用 Hitachi Debugging Interface セットアップガイド」を参照してください。

2.5 トラブルシューティング

2.5.1 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れた場合、PC インタフェースボードは E6000 エミュレータを検出できていません。

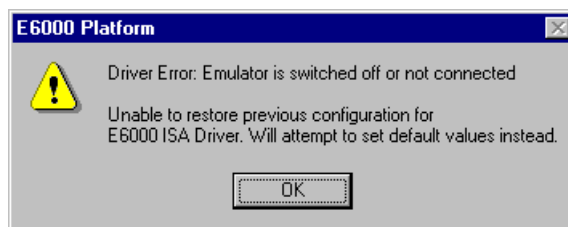


図2.4 エラーメッセージ (1)

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。

2.5.2 通信不良

以下のメッセージボックスが表示されると、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2.5 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エリアと PC インタフェースボード上のリヤパネルスイッチの設定が異なります。
- 選択されたメモリエリアが別のアプリケーションで使用されています。

2. セットアップ

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

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

以下に、E6000 エミュレータのコネクタを示します。

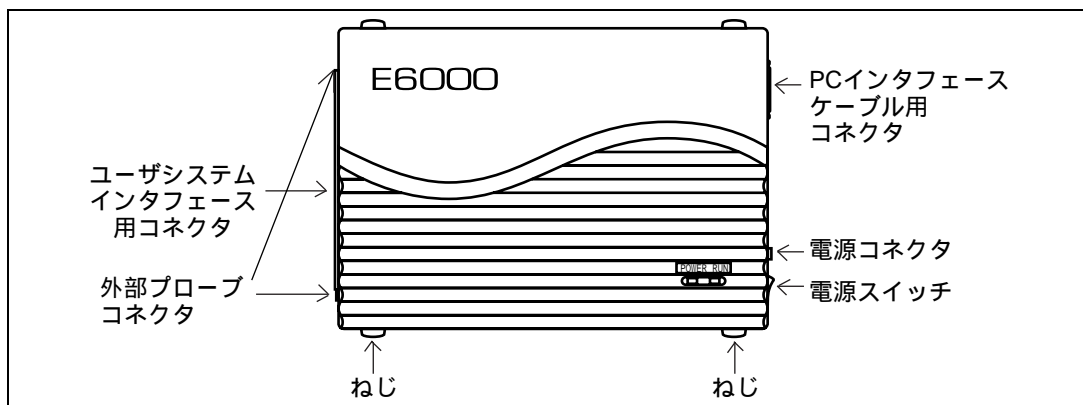


図3.1 E6000 コネクタの位置

3. ハードウェア

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

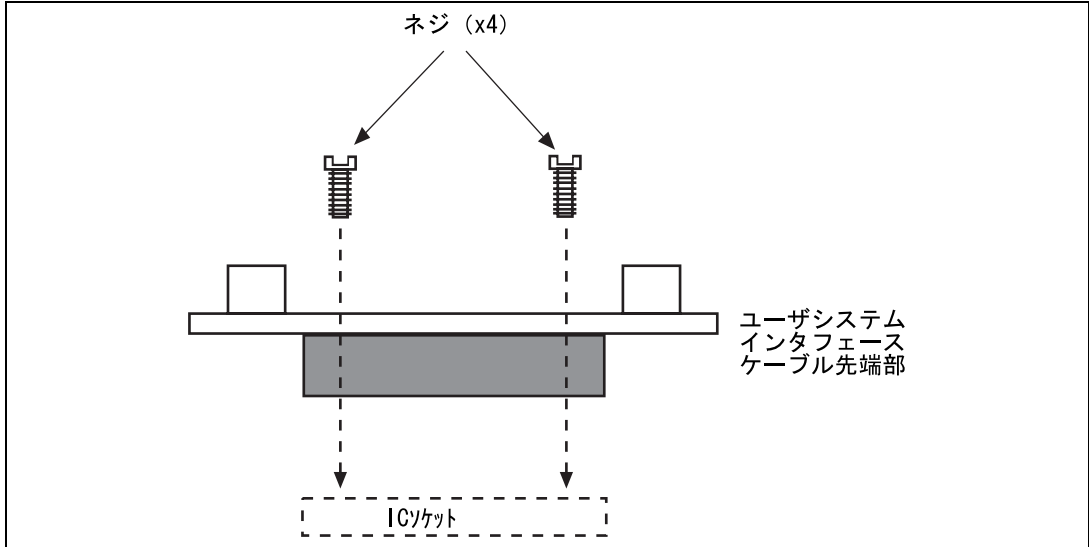


図3.2 ユーザシステムインタフェースケーブルの接続

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

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

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

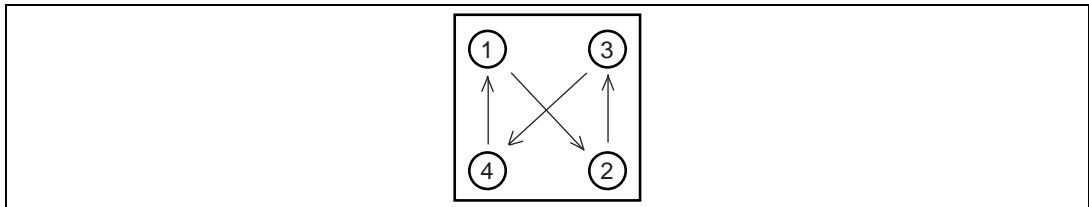


図3.3 ネジの締め付け順序

【注】 ネジ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。

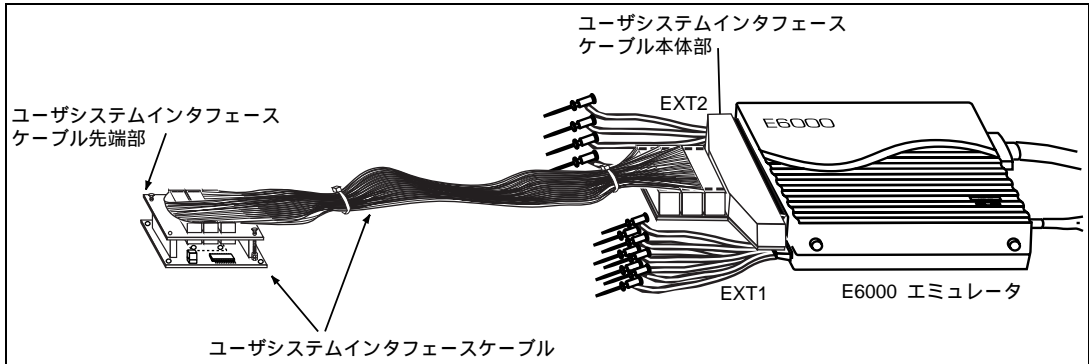


図3.4 ユーザシステムインタフェースケーブル外観図

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。

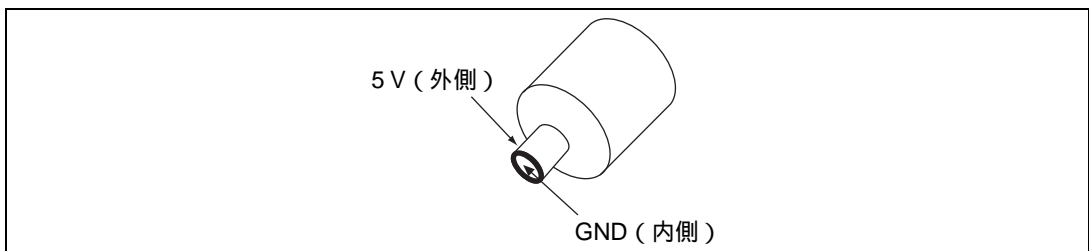


図3.5 電源プラグ

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。電源が 4.75V 未満の場合、E6000 エミュレータに必要な電流が供給されません。

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

3.3 オプションメモリボード

E6000 エミュレータに外部領域のエミュレーションメモリとしてオプションメモリボードを接続できます。オプションメモリボードの詳細はオプションメモリボード添付の取扱い説明書を参照してください。オプションメモリボードは 4 つの等しいサイズのバンクを持っており、これらのバンクをユーザエリアに割り当てることができます。

3.3.1 オプションメモリボードの構成

オプションメモリボードの構成は、マッピング RAM によって制御されます。System Status ウィンドウの Memory シートを開くと、インストールされたオプションメモリボードをチェックでき、また Memory Mapping ダイアログボックスより 4 つのバンクを必要なアドレスに再配置できます。

3.4 ハードウェアインタフェース

E6000 エミュレータのユーザシステムインタフェース信号は、一部の信号を除いてバッファなしに直接エミュレータ上の MCU に接続されています。MCU に入力する前にエミュレータ制御回路が挿入される信号は各エミュレータにより異なりますので、製品別補足説明書を参照してください。

3.4.1 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 Vref には、この保護回路がありません。

アナログポート以外のポートには、プルアップ抵抗が接続されています。

ユーザシステムインタフェースケーブル先端部の Vcc 端子 (AVcc 端子を除く) は、すべて 1 つに接続されています。E6000 エミュレータは、これを監視して、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。

3.4.2 ユーザインタフェース回路

E6000 エミュレータのユーザインタフェースには、ケーブルにより約 8ns の信号の遅れが生じます。また、プルアップ抵抗により信号がハイインピーダンス状態でもハイレベルになります。このことを考慮してユーザシステムのハードウェアを調整してください。

以下にユーザインタフェース信号回路例を示します。ユーザインタフェースは製品により各々異なりますので、添付の「補足説明書」を参照してください。

(1) 汎用ポート等

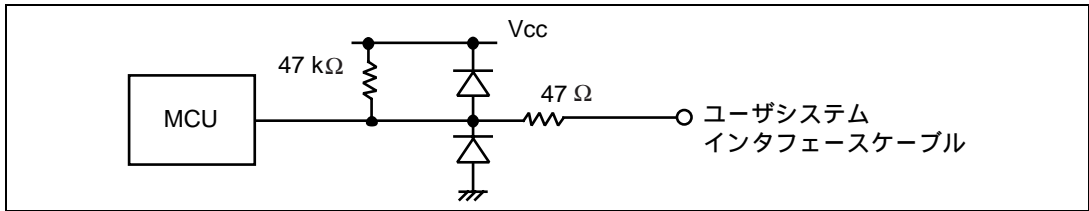


図3.6 ユーザインタフェース信号回路

(2) モード端子 (MD2、MD1、MD0)、WAIT、NMI 等

WAIT 信号および NMI 信号はエミュレータ制御回路を経由して MCU に入力されます。したがって、これらの信号の立ち上がり / 立ち下がり時間は 8ns/V 以下にしてください。STBY 信号およびモード端子はモニタのみ行っています。動作モードは、HDI の設定に従います。

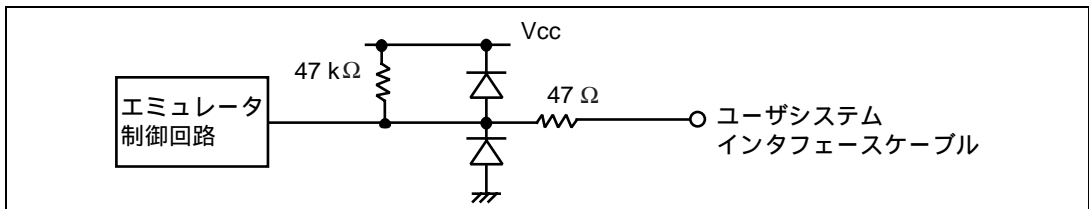


図3.7 モード端子, WAIT, NMI 等信号回路

(3) RESET

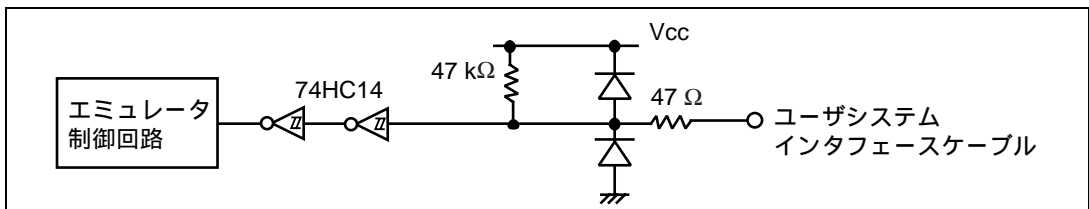


図3.8 RESET 信号回路

3. ハードウェア

(4) アナログポート制御信号

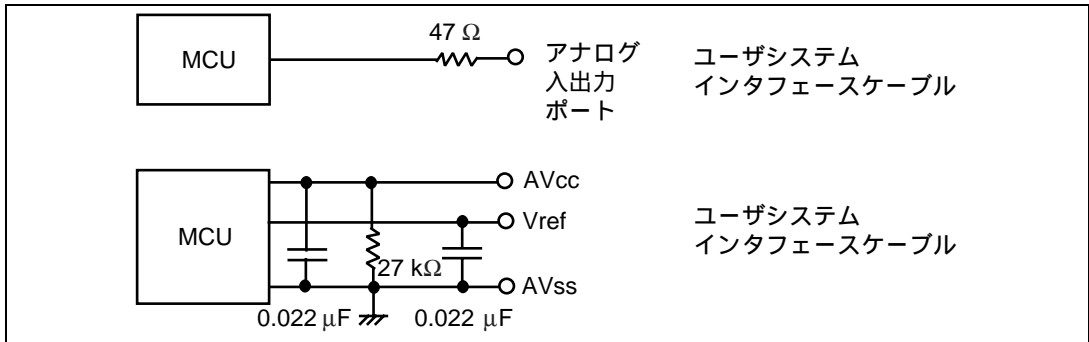


図3.9 アナログポート制御信号回路

(5) IRQ0-IRQ7

IRQ0-IRQ7 信号は MCU への入力と同時にトレース取得用の回路にも入力されます。したがって、これらの信号の立ち上がり / 立ち下がり時間は 8ns/V 以下にしてください。

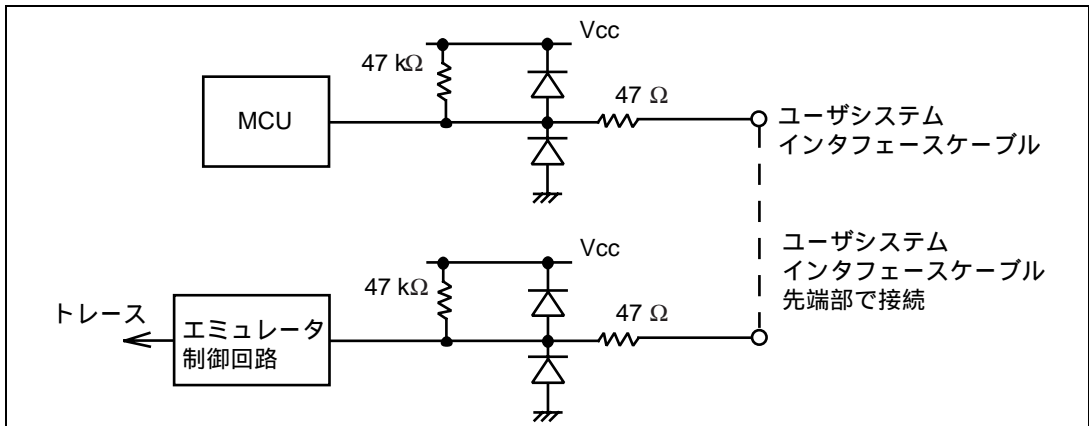


図3.10 IRQ0-IRQ7 信号回路

3.4.3 クロック発振器

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

3.4.4 外部プローブ 1 (EXT1) / トリガ出力

E6000 エミュレータ筐体側面にある EXT1 のマークが記された 8 ピンコネクタ (ユーザインタフェースコネクタの右下) は、外部プローブ入力 4 本とトリガ出力 2 本を備えています。以下にこのコネクタのピン配置を示します。

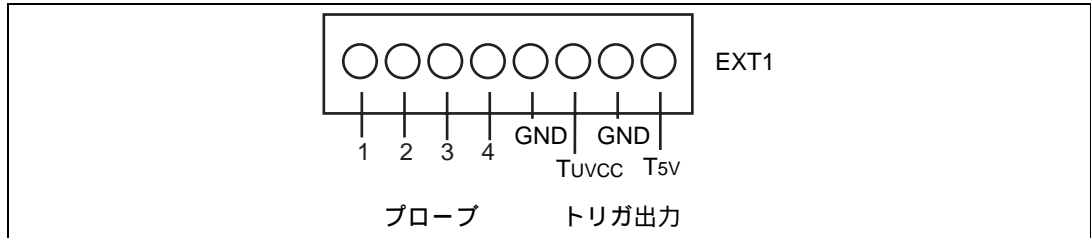


図3.11 外部プローブ 1 コネクタ

以下に外部プローブ 1 のインタフェース回路を示します。

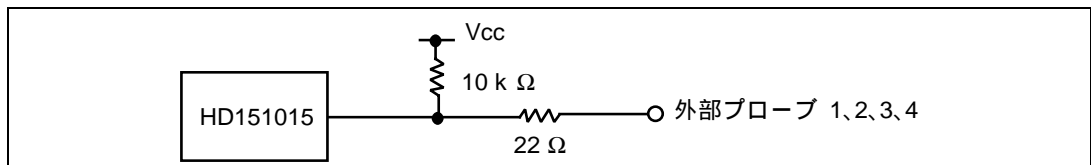


図3.12 外部プローブ 1 インタフェース回路

トリガ出力はイベントチャンネル 8 によって出力されるローレベル信号です。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。

3.4.5 外部プローブ 2 (EXT2) / トリガ出力

E6000 エミュレータ筐体側面にある EXT2 のマークが記された 6 ピンコネクタ (ユーザインタフェースコネクタの右下) は、トリガ出力 4 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。

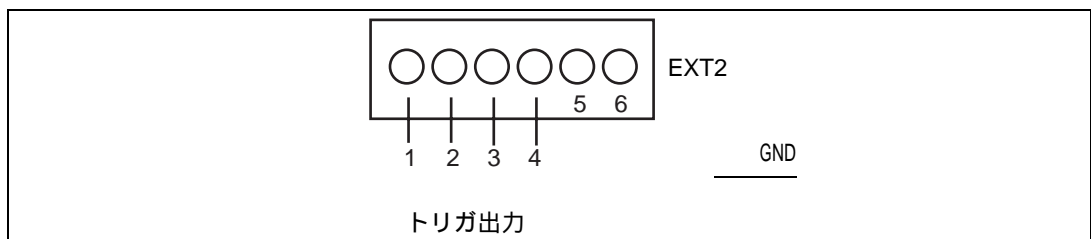


図3.13 外部プローブ 2 コネクタ

トリガ出力はバスモニタ機能のトリガ設定 (1 ~ 4) の条件が一致した場合にリード/ライトの期間だけ出力されるハイレベル信号です。トリガ出力は Vcc (ユーザシステム電源電圧) レベルです。

3.4.6 電源フォロワ回路

注意

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

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

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

Configuration ダイアログボックスを使用して、[User VCC Threshold] を V_{cc} max から 0V の範囲で設定できます。ユーザ V_{cc} がその値よりも下がった場合、System Status ウィンドウの Platform シートにある [User System Voltage] には [Down] が表示されます。User VCC Threshold 電源レベルよりも高い場合は、[OK] が表示されます。

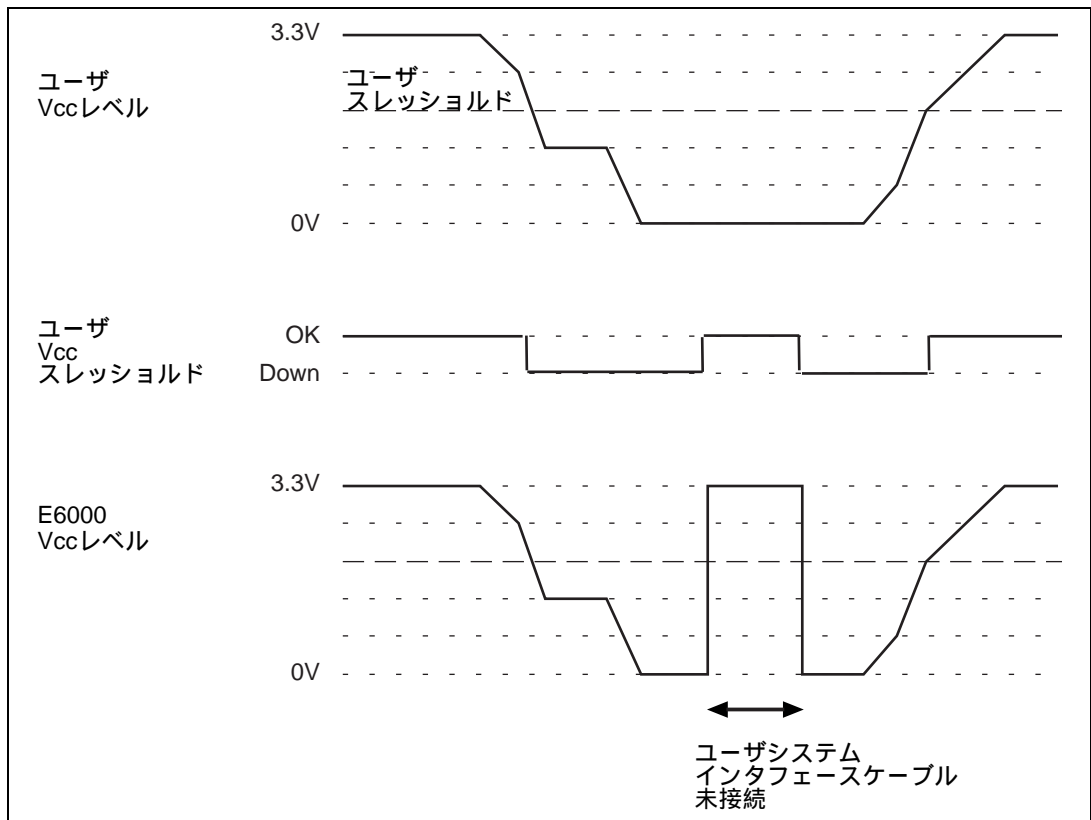


図3.14 ユーザシステムと E6000 との Vcc の関係 (Vcc=3.3V の例)

3.5 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	リセットベクタ値	リセットベクタ値
	ER0 to ER6	不定	不定
	ER7 (SP)	H'10	不定
	CCR	I マスクは 1 その他は不定	I マスクは 1 その他は不定
リセットコマンド後	PC	リセットベクタ値	リセットベクタ値
	ER0 to ER6	不定	不定
	ER7 (SP)	H'10	不定
	CCR	I マスクは 1 その他は不定	I マスクは 1 その他は不定

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、添付の「補足説明書」を参照してください。

3.5.1 A/D コンバータ、D/A 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 変換と D/A 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。

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用して実行しますので、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。

例といたしまして、E6000 H8S/2655 を使用した手順を説明します。他の製品をご使用の方は、ファイル名、ディレクトリ名などをご使用のものに読みかえてください。

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、以下のことを行ってください。

- 「第 2 章 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用するために 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、ご使用になる MCU のハードウェアマニュアルとプログラミングマニュアルを参照してください。

チュートリアルは、10 個のランダムデータを昇順、降順に並び換えるプログラムです。ソースプログラム (Sort.c) および ELF/DWARF2 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 HDI の起動

HDI を起動するには、[Start]メニューの[HDI for E6000 H8S_2655]から[Hitachi Debugging Interface]を選択してください。

4. チュートリアル

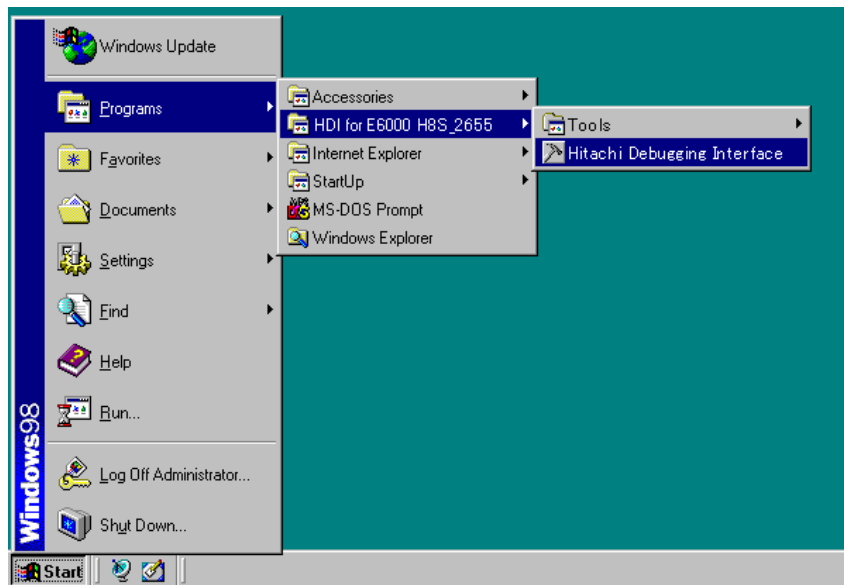


図4.1 HDI 起動メニュー

4.2.1 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、使用するプラットフォームを選択する必要があります。

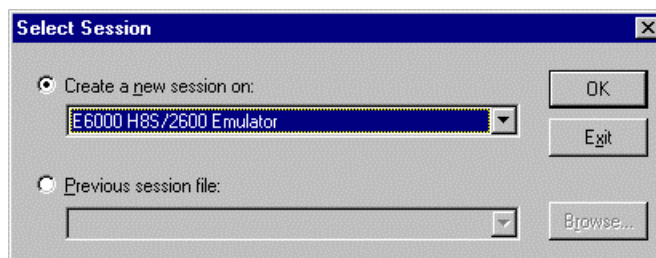


図4.2 プラットホームの選択

- このチュートリアルでは、E6000 H8S/2600 Emulator を選択し、[OK] をクリックしてください。

[File] メニューから [New Session...] 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。

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 ウィンドウが表示されます。以下にウィンドウの主な機能を示します。

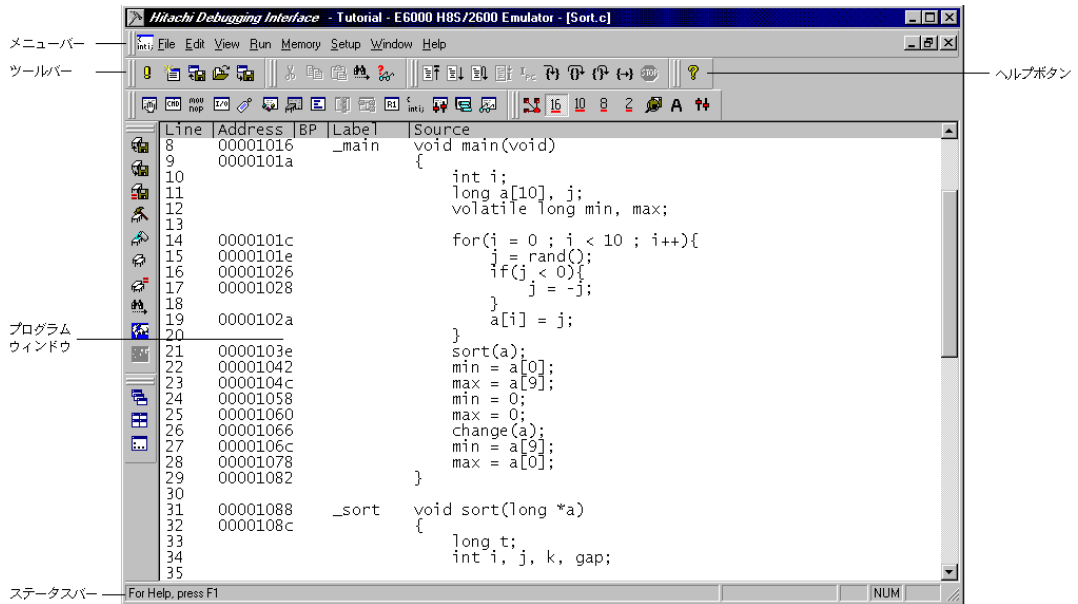


図4.3 HDI ウィンドウ

HDI の主な機能については、「日立デバッグインタフェースユーザズマニュアル」をご覧ください。E6000 エミュレータに特有な機能は、オンラインヘルプを参照してください。

メニューバー

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。

ツールバー

頻繁に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。

プログラムウィンドウ

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。

4.3 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、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。

4.3.1 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。

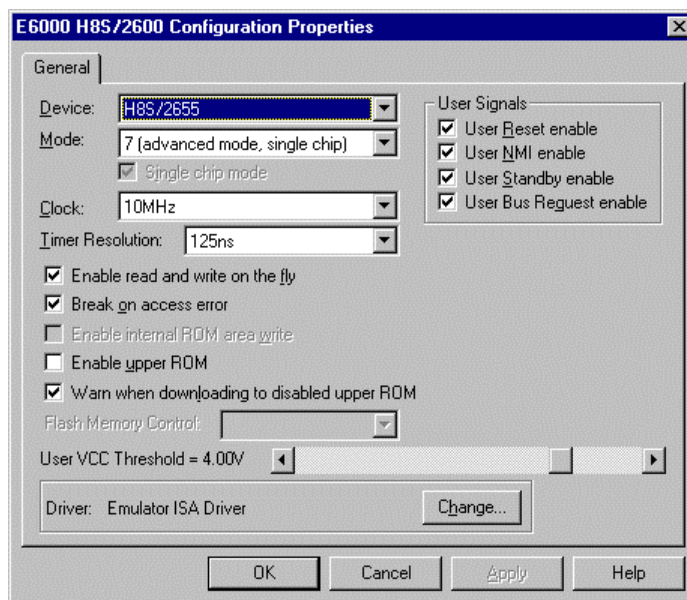


図4.4 Configuration ダイアログボックス

- 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
デバイス (Device)	H8S/2655
モード (Mode)	7 (アドバンスモード、シングルチップ)
動作クロック (Clock)	10MHz
タイマ分解能 (Timer Resolution)	125ns
その他のオプション	デフォルト

- [OK] をクリックして対象 MCU 条件を設定してください。

4.3.2 メモリマッピング

Configuration 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り付けを行います。

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

Memory Mapping ダイアログボックスが以下のように表示されます。

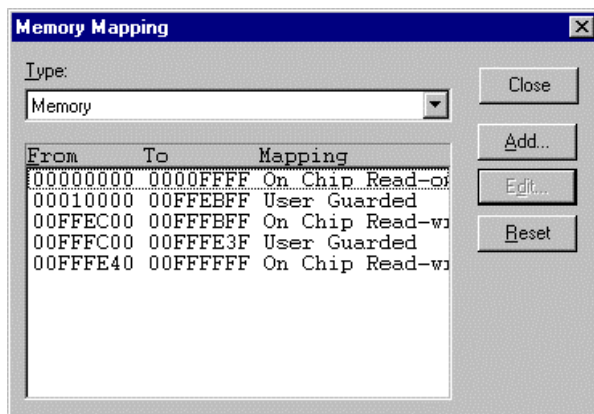


図4.5 Memory Mapping ダイアログボックス

E6000 エミュレータのメモリには、以下の3つのタイプがあります。

表4.2 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。
ユーザ (User)	ユーザシステム上のメモリをアクセスします。
エミュレータ (Emulator)	オプションメモリボードをアクセスします。

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。

4. チュートリアル

表4.3 メモリタイプオプション

アクセスタイプ	説明
リードライト(Read-Write)	RAM
リードオンリ(Read-Only)	ROM
ガード (Guarded)	アクセス不可

本チュートリアルでは、デフォルトのマッピングを使用しますが、以下のようにメモリの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。
- ここでは Memory Mapping ダイアログボックスの 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。

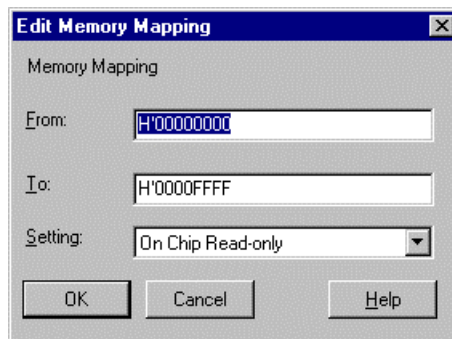



図4.6 Edit Memory Mapping ダイアログボックス

- [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。
- デバイスのマップ情報を表示するには [View] メニューから [Status] を選択するか、またはツールバーの [Status] ボタン  をクリックし System Status ウィンドウを開き、Memory シートを選択してください。デバイスのマップ情報が以下のように表示されます。

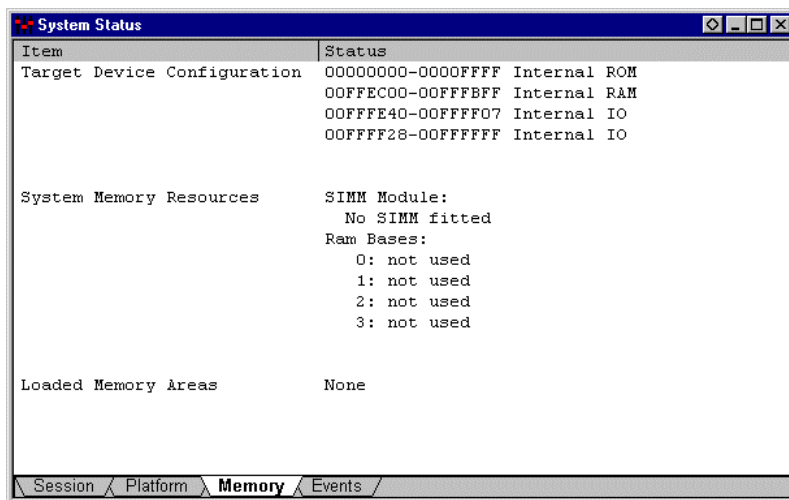


図4.7 System Status ウィンドウ (Memory シート)


【注】 メモリマップは対象 MCU および MCU のモードによって異なります。

4.4 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。

4.4.1 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 ELF/DWARF2 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。

Load Program ダイアログボックスが開きます。

- [Browse...] ボタンをクリックし、Open ダイアログボックスより Tutorial ディレクトリの下
の Tutorial.abs ファイルを選択した後、[Open] ボタンをクリックしてください。

Load Program ダイアログボックスに戻りますので、さらに [Open] ボタンをクリックしファイルのダウンロードを開始してください。

4. チュートリアル

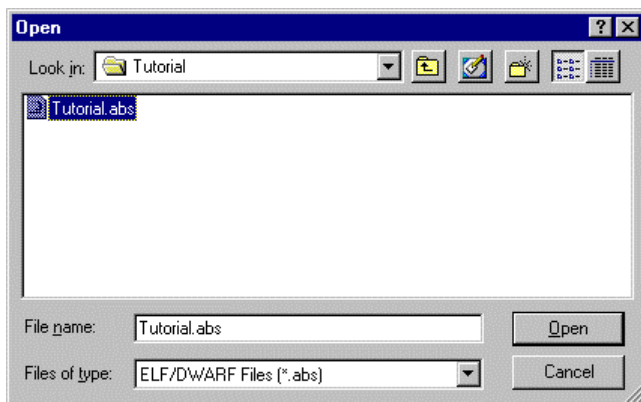


図4.8 Open ダイアログボックス (オブジェクトファイルの選択)

ファイルがロードされると、以下のダイアログボックスに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が表示されます。

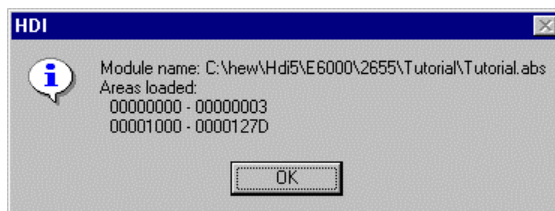



図4.9 HDI ダイアログボックス

- [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した。

4.4.2 プログラムリストの表示

HDI では、ソースレベルでプログラムをデバッグできます。

- [View] メニューから [Source...] を選択するか、またはツールバーの [Program Source] 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。

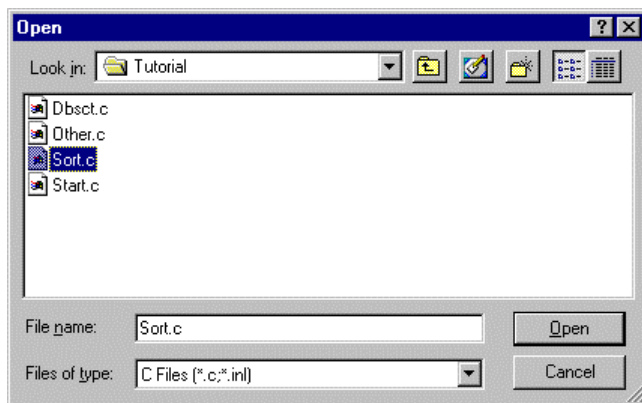


図4.10 Open ダイアログボックス (ソースファイルの選択)

- Sort.c を選択し、[**Open**] をクリックしてください。Program ウィンドウが表示されます。

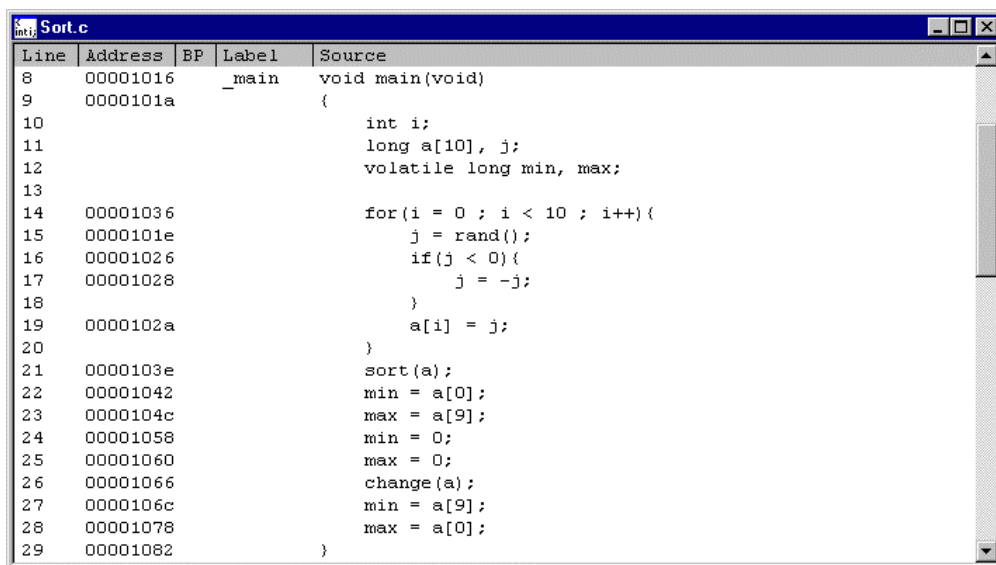


図4.11 ソースプログラム画面

- 必要ならば、[**Setup**] メニューの [**Customize**] サブメニューから [**Font...**] オプションを選択し、ホスト PC に合ったフォントとサイズを選択してください。

Program ウィンドウを、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他の部分を見ることができます。

4.5 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 PC Break があります。この機能を使用することにより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。

4.5.1 PC Break の設定

Program ウィンドウによって、プログラムのあらゆるポイントに PC Break 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 H'103e に PC Break を設定します。

- ・ H'103e 番地を含むラインの [BP] カラムをダブルクリックしてください。

Line	Address	BP	Label	Source
8	00001016		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	00001036			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	00001026			if(j < 0){
17	00001028			j = -j;
18				}
19	0000102a			a[i] = j;
20				}
21	0000103e	● Break		sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}

図4.12 ブレークポイント (PC Break) の設定


その位置に " Break" が表示され、そのアドレスに PC Break 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" ...

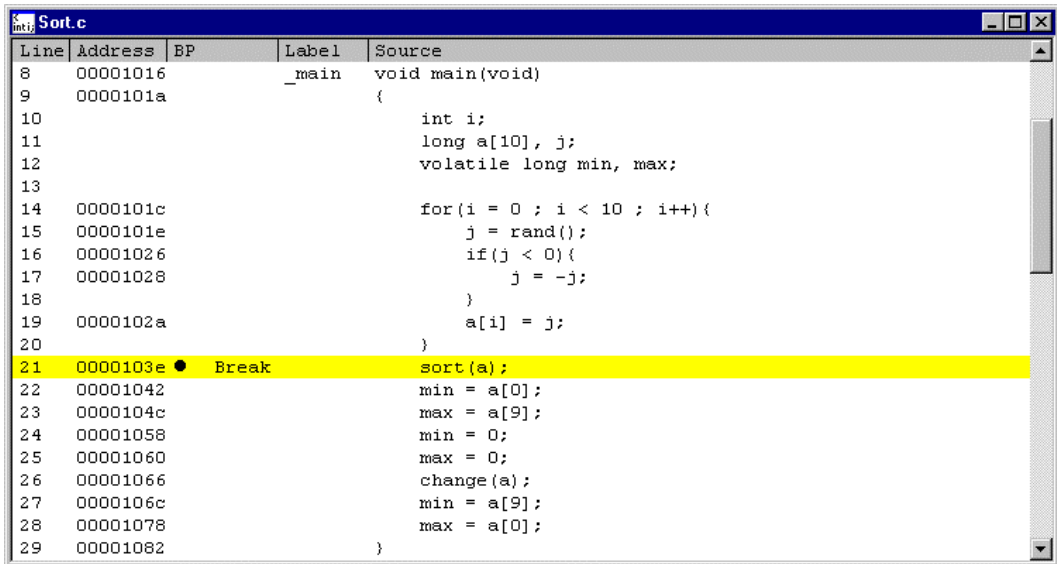
【注】 "-Timer" および "-Trace" は対応する "+Timer" および "+Trace" が設定されているときのみ選択可能となります。

4.5.2 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 [Reset Go]を選択するか、またはツールバーの [Reset Go] ボタン  をクリックしてください。


プログラムは PC Break 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 Program ウィンドウ中でステートメントが強調表示されます。



Line	Address	BP	Label	Source
8	00001016		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	0000101c			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	00001026			if(j < 0){
17	00001028			j = -j;
18				}
19	0000102a			a[i] = j;
20				}
21	0000103e	●	Break	sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}

図4.13 ステートメントの強調表示

[Break = PC 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。また System Status ウィンドウでも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]を選択するか、またはツールバーの [Status] ボタン  をクリックし System Status ウィンドウを開き Platform シートを選択してください。

4. チュートリアル

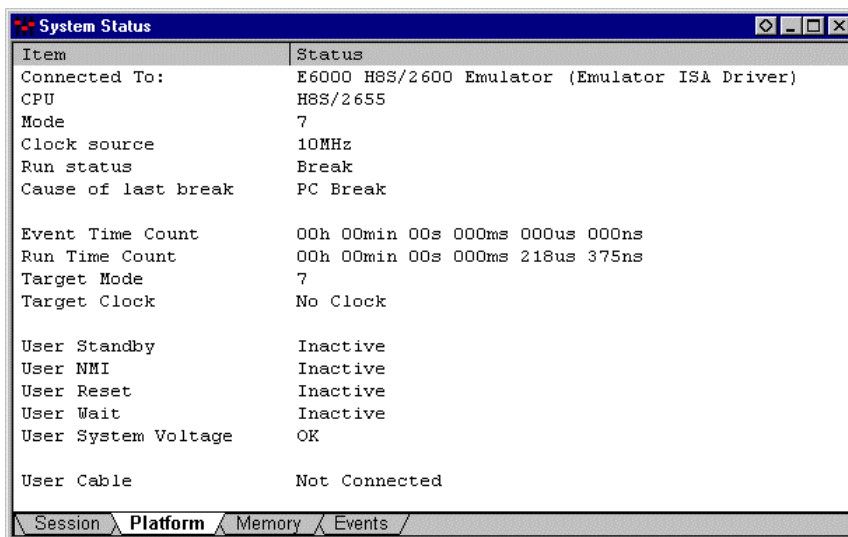



図4.14 System Status ウィンドウ (Platform シート)

[Cause of last break] のラインは、ブレークの原因が PC Break であることを示しています。また [Run Time Count] のラインは、プログラムが実行してから停止するまでの時間 (実行時間) が 218.375 μ s だったことを示しています。イベント間実行時間測定 [Event Time Count] (+Timer, -Timer で設定) および実行時間に使用するタイマの分解能は、Configuration ダイアログボックスの [Timer Resolution] により設定されます。長時間の時間計測に分解能 20ns のような小さい値を使用すると、誤差が大きくなることがあります。計測時間の長さに合わせて分解能を調整してください。

4.5.3 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 ウィンドウに表示されます。

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

Register	Value
ER0	00FFFBF0
ER1	00000FF6
ER2	0FD05770
ER3	00FFFA04
ER4	0000000A
ER5	00FFFBCC
ER6	00000FF6
ER7	00FFFBCC
PC	00103E
+ CCR	I1-U-Z--
+ EXR	-----111
MACH	00000040
MACL	00000000

図4.15 Registers ウィンドウ

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'103e を示しています。(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 ウィンドウで変更できます。

- PC の値を変えるには、Registers ウィンドウで PC に対応する [Value] カラム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。

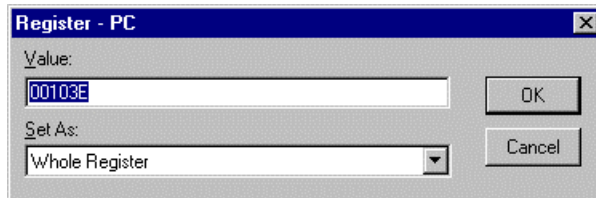



図4.16 Register ダイアログボックス


- 値を H'1016 (メインプログラムの先頭アドレス) 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メインプログラムの先頭に移動し、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] を選択するか、またはツールバーの [Go] ボタン  をクリックし、ブレークポイントまでの実行を再開してください。

4.5.4 ブレークポイントの確認

プログラムに設定したすべてのブレークポイントの一覧を Breakpoints ウィンドウで見ることができます。

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

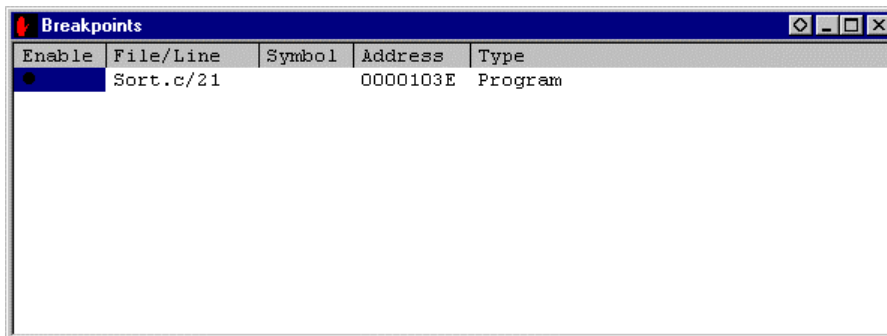


図4.17 Breakpoints ウィンドウ

Breakpoints ウィンドウによって、ブレークポイントの有効または無効、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。


4.6 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。

4.6.1 メモリを表示する

メモリブロックの内容を Memory ウィンドウで見ることができます。

たとえば、Byte で main に対応したメモリを見る場合：

- [View] メニューから [Memory...] を選択するか、またはツールバーの [Memory] ボタン  をクリックしてください。
- [Address] エディットボックスに main を入力し、[Format] ドロップダウンリストボックスを Byte に設定してください。

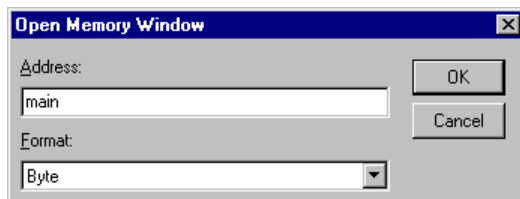


図4.18 Open Memory Window の設定

- [OK] をクリックして、指定されたメモリ領域を示す Memory ウィンドウを開くことにより、メモリブロックの内容が確認できます。

Address	Data	Value
00001016	79 37 00 30 0F F5 19 44	y7.0...D
0000101E	5C 00 01 90 17 F0 0F 86	\.....
00001026	4C 02 17 B6 17 F4 0F C0	L.....
0000102E	10 70 0A D0 01 00 69 86	.p...i.
00001036	0B 54 79 24 00 0A 4D E0	.Ty\$.M.
0000103E	0F D0 55 46 01 00 69 50	..UF..iP
00001046	01 00 6F F0 00 2C 01 00	..o....
0000104E	6F 50 00 24 01 00 6F F0	oP.\$..o.

図4.19 Memory ウィンドウ(Byte)

4.6.2 変数を表示する

プログラムをステップ処理するとき、プログラムで使用される変数の値を見ることができ、期待したよう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 long 型の配列変数"a"を見ることができます。

- Program ウィンドウに表示されている配列変数"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 Program ウィンドウをクリックし、ポップアップメニューより、[Add Watch] を選択してください。

Watch ウィンドウに変数が表示されます。

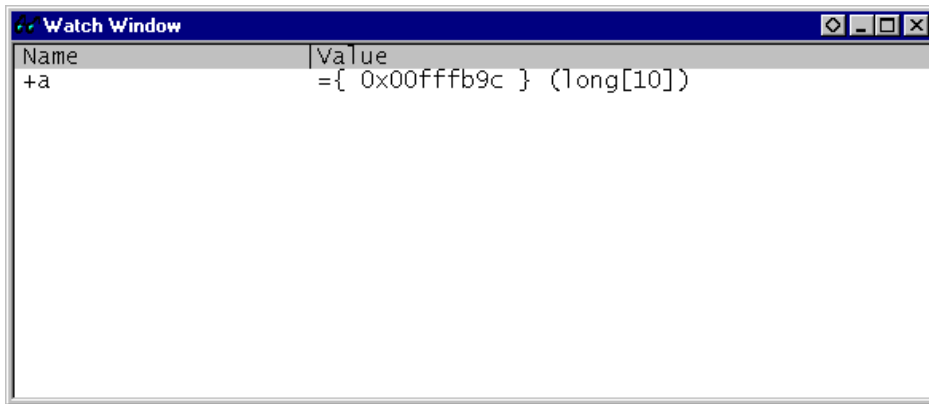



図4.20 Watch ウィンドウ (変数追加後)

Watch ウィンドウのシンボル“a”の左にある“+”をダブルクリックし、シンボルを拡張して各配列の要素を見ることができます。

必要ならば、[Setup] メニューの [Radix] サブメニューから [Decimal] を選択するか、あるいはツールバーの [Radix = Decimal] ボタン  を押し、10 進数表示にしてください。

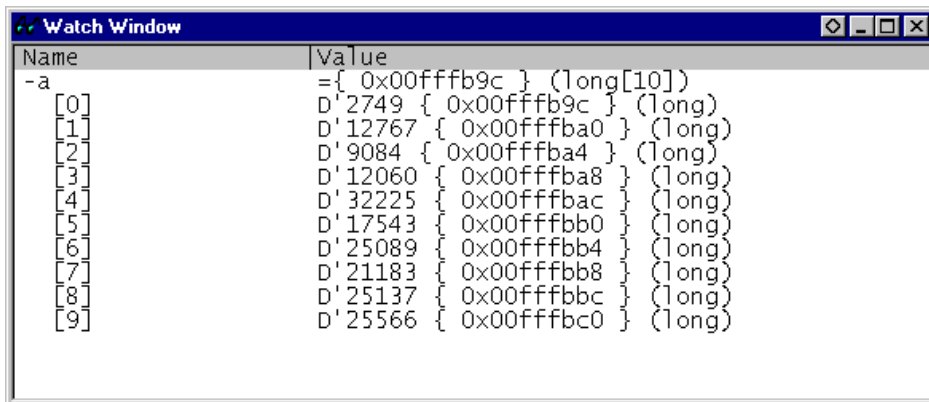


図4.21 Watch ウィンドウ (シンボル拡張)

また、変数名を指定して、Watch ウィンドウに変数を追加することもできます。

- 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 [Add Watch...] を選択してください。
- 変数名“max”を入力し、[OK] ボタンをクリックしてください。

4. チュートリアル

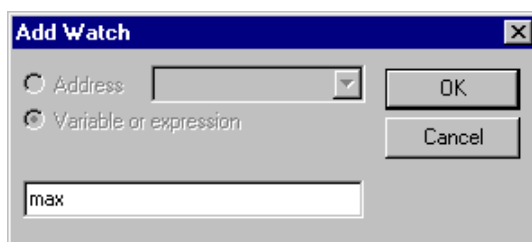


図4.22 Add Watch ダイアログボックス

Watch ウィンドウに volatile long 型の変数"max"が追加されます。

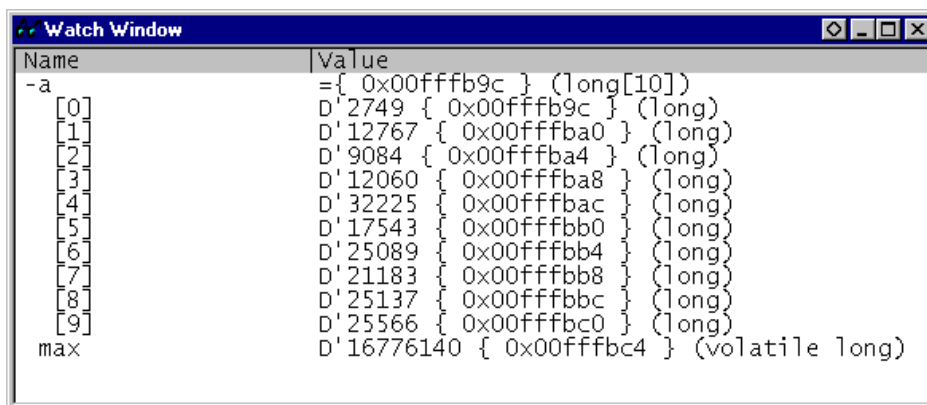


図4.23 Watch ウィンドウ (変数の追加)


4.7 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。以下に示すようなステップオプションがあります。

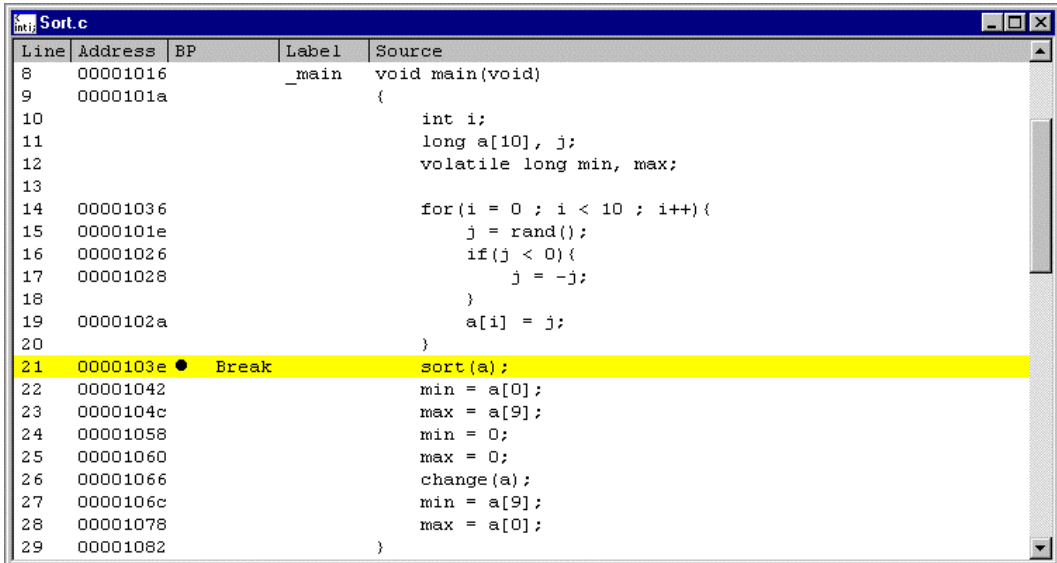
表4.4 プログラムステップオプション

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。

4.7.1 シングルステップ


- H'103e に PC Break が設定されていることを確認してください。
- 次に [Run] メニューから [Reset Go] を選択するか、あるいはツールバーの [Reset Go] ボタン  を押してください。

設定した PC Break により H'103e で停止し、sort(a);のステートメントが強調表示されます。

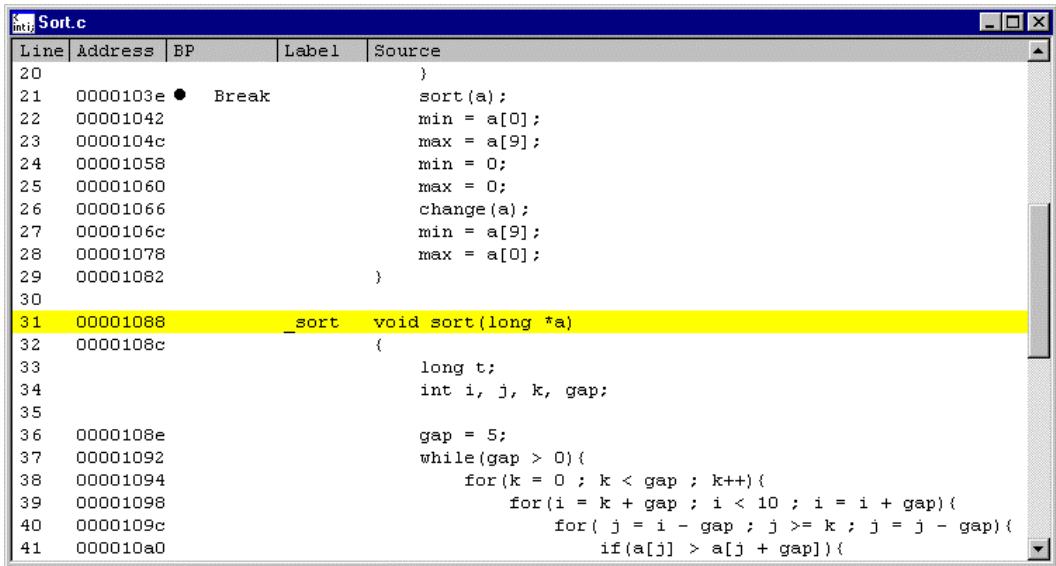


Line	Address	BP	Label	Source
8	00001016		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	00001036			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	00001026			if(j < 0){
17	00001028			j = -j;
18				}
19	0000102a			a[i] = j;
20				}
21	0000103e	●	Break	sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}

図4.24 Reset Go 実行後の Program ウィンドウ


- 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。

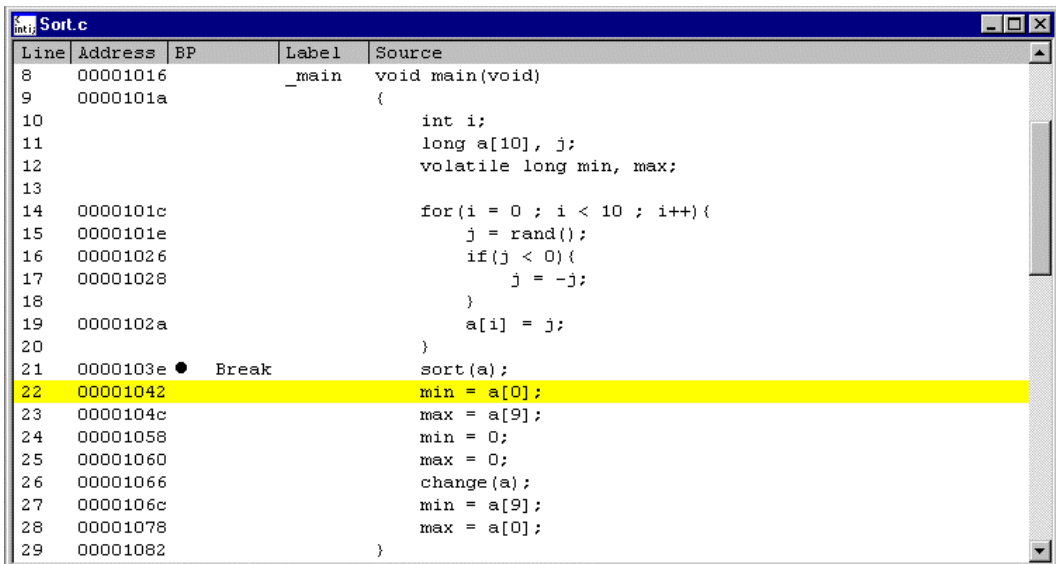
4. チュートリアル



Line	Address	BP	Label	Source
20				}
21	0000103e	● Break		sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}
30				
31	00001088		_sort	void sort(long *a)
32	0000108c			{
33				long t;
34				int i, j, k, gap;
35				
36	0000108e			gap = 5;
37	00001092			while(gap > 0){
38	00001094			for(k = 0 ; k < gap ; k++){
39	00001098			for(i = k + gap ; i < 10 ; i = i + gap){
40	0000109c			for(j = i - gap ; j >= k ; j = j - gap){
41	000010a0			if(a[j] > a[j + gap]){

図4.25 Step In 実行後の Program ウィンドウ (1)

- [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'1042 が強調表示され、関数から抜け出したことが判ります。



Line	Address	BP	Label	Source
8	00001016		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	0000101c			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	00001026			if(j < 0){
17	00001028			j = -j;
18				}
19	0000102a			a[i] = j;
20				}
21	0000103e	● Break		sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}

図4.26 Step Out 実行後の Program ウィンドウ

- さらに [Step In] コマンドによって change 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。


【注】 ステップ実行時 C/C++ライブラリ関数または実行時ルーチン内にステップインした場合、自動的に Disassembly ウィンドウが開きます。この状態でステップ実行を行なった場合、アセンブラレベルでステップ実行が行なわれます。C/C++言語ソースレベルでステップ実行する場合は、[Step Out]コマンドにより C/C++ライブラリ関数または実行時ルーチンを抜けた後、Disassembly ウィンドウを閉じてください。

Line	Address	BP	Label	Source
8	00001016		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	0000101c			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	00001026			if(j < 0){
17	00001028			j = -j;
18				}
19	0000102a			a[i] = j;
20				}
21	0000103e	●	Break	sort(a);
22	00001042			min = a[0];
23	0000104c			max = a[9];
24	00001058			min = 0;
25	00001060			max = 0;
26	00001066			change(a);
27	0000106c			min = a[9];
28	00001078			max = a[0];
29	00001082			}

図4.27 Step In 実行後の Program ウィンドウ (2)

4.7.2 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 change 関数を実行し、次のアドレス H'106c で停止します。

4. チュートリアル

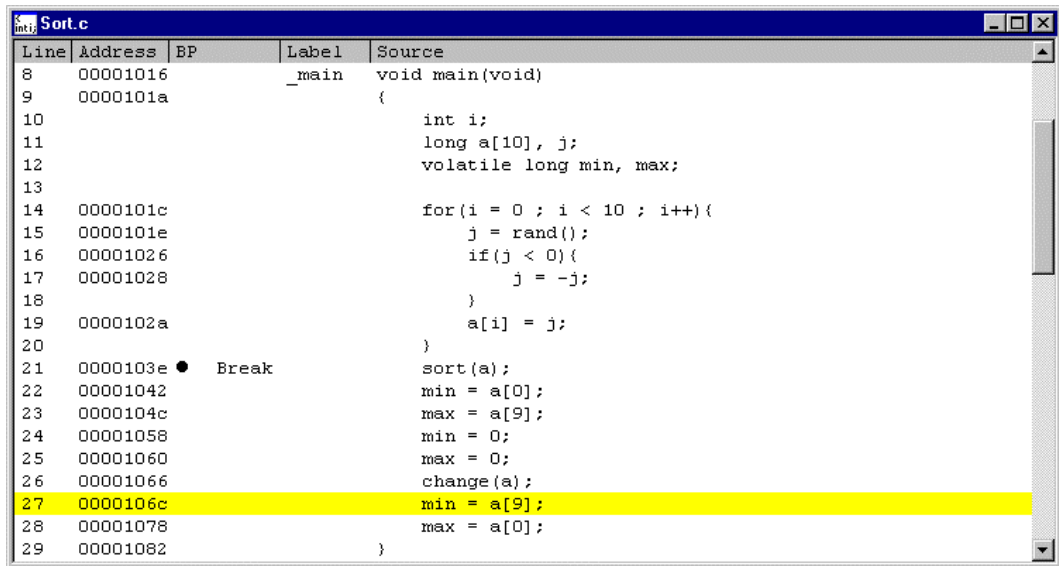



図4.28 Step Over 実行後の Program ウィンドウ

4.7.3 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、5つのローカル変数 i、a、j、min、max を宣言しています。

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

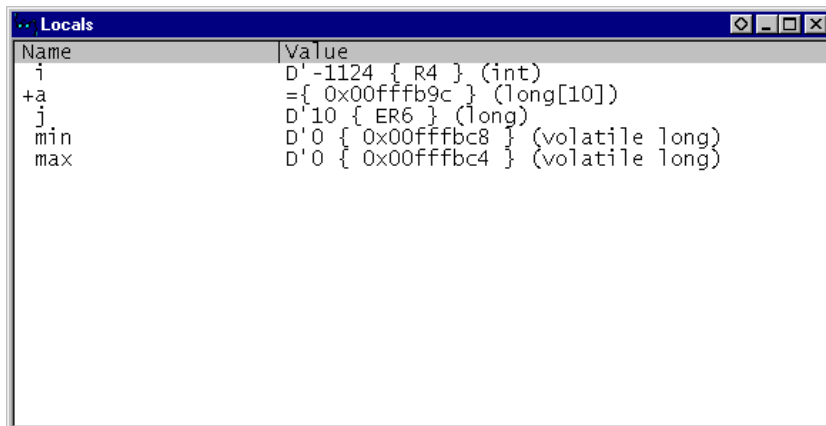



図4.29 Locals ウィンドウ

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

- [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] ボタン  をクリックして、1ステップ実行してください。

変数"min"の内容が変更され、その値が表示されます。

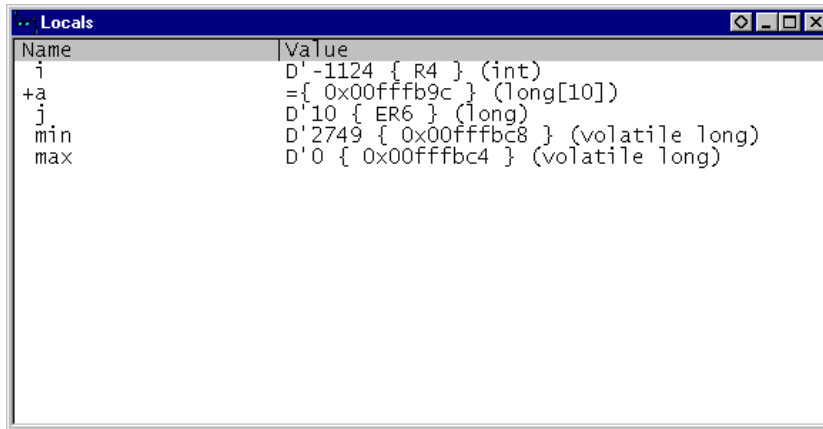


図4.30 Locals ウィンドウ (変数"min"内容変更後)

- Locals ウィンドウのシンボル"a"の左にある"+"をダブルクリックし、シンボルを拡張して各配列の要素を表示させてください。
- sort 関数実行前の配列変数"a"の要素を参照し、ランダムデータが降順にソートされていることを確認してください。

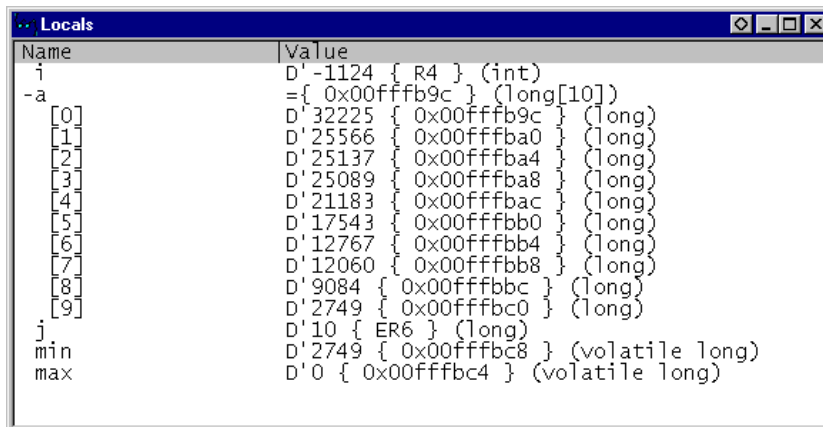


図4.31 Locals ウィンドウ (配列変数"a"ソート後)



4.8 イベント検出システム (Complex Event System) の使用方法

本チュートリアルでは、Memory ウィンドウでメモリ領域の内容を見ること、あるいは Watch ウィンドウおよび Locals ウィ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。そこで、E6000 エミュレータのイベント検出システムを使用することにより、たとえば、プログラムが 5 回 H'1098 をアクセスした時を検出することができます。

4.8.1 イベント検出システムによる Event の設定

イベント検出システムを使用した Event 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。

- [Setup] メニューの [Radix] サブメニューから [Hexadecimal] を選択するか、あるいはツールバーの [Radix = Hex] ボタン  を押し、16 進数表示にしてください。これにより 16 進数入力時に基数の接頭部 (H') を省略することができます。
- [View] メニューから [Breakpoints] を選択するか、またはツールバーの [Breakpoints] ボタン  をクリックして、Breakpoints ウィンドウを表示してください。
- 新しいブレークポイントを設定するため Breakpoints ウィンドウ内でマウスの右ボタンをクリックし [Add...] を選択してください。

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。

- [Type] 選択を [Event] にし、条件として [Address Lo] ボックスにアドレス H'1098 を入力してください。

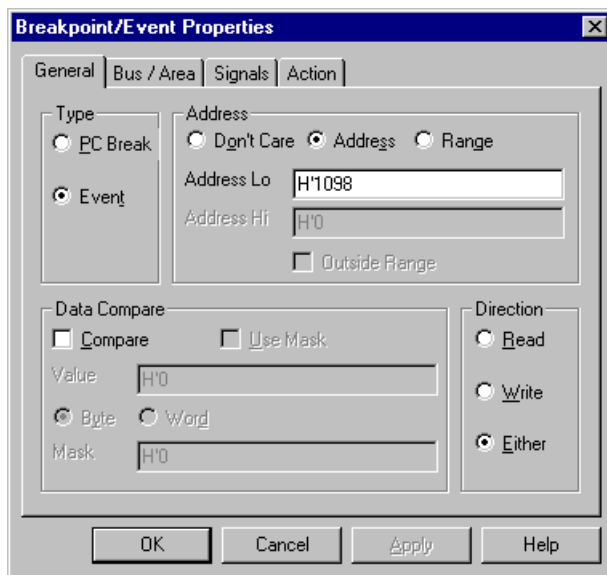


図4.32 ブレークポイントの追加 (アドレス指定)

- [Action] タブをクリックして [Action] パネルを表示してください。

- 5回アクセスした時にブレークさせるため、[Required number of event occurrences]エディットボックスに5を入力してください。

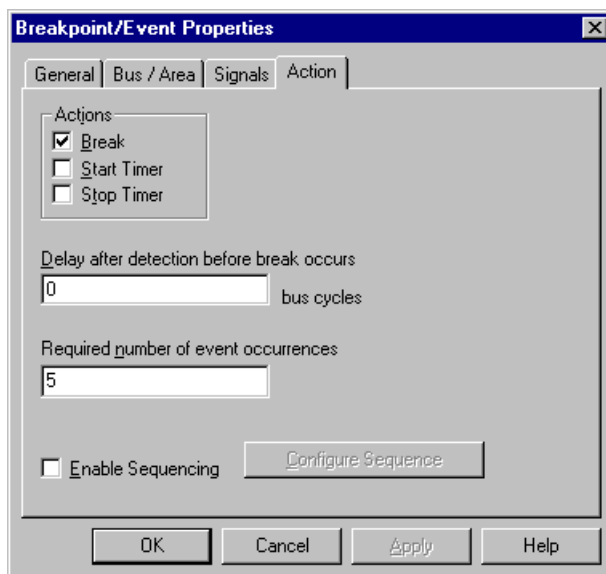


図4.33 ブレークポイントの追加 (回数指定)

- [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'1098 が5回アクセス (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 ウィンドウは、設定された新しい Event を表示します。

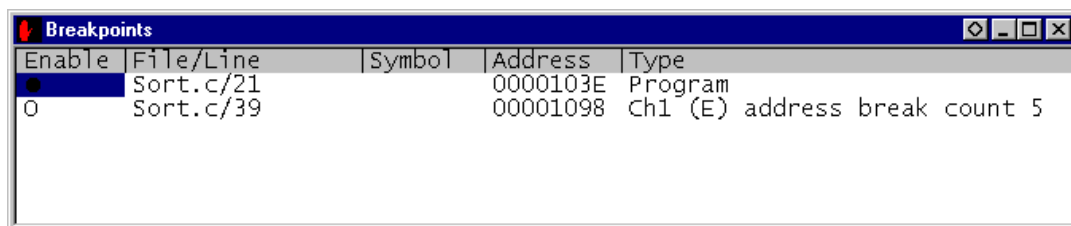




図4.34 Breakpoints ウィンドウ (追加後)

- [Run] メニューから [Reset Go] を選択するか、あるいはツールバーの [Reset Go] ボタン  をクリックしてください。
アドレス H'103e に設定された PC Break で停止します。

- さらに [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'1098 への5回アクセスにより実行が停止します。

4. チュートリアル

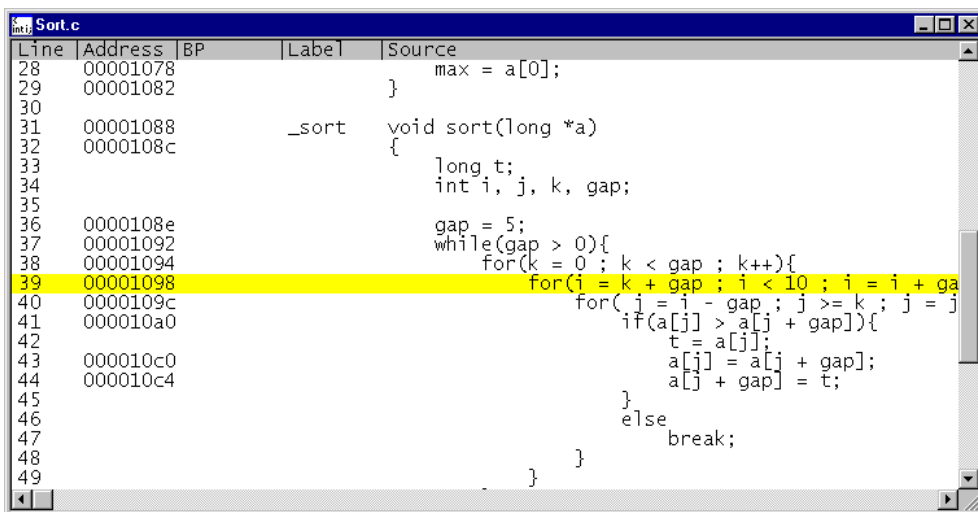


図4.35 Event ブレークポイントによるプログラムの停止

また、ステータスバーには"Break = Complex Event System"と表示し、Event に設定した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。


【注】 イベント検出システムでは条件成立後、実行停止までにディレイが発生します。

4.9 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。

4.9.1 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]を選択するか、あるいはツールバーの [Trace]ボタン  をクリックして、Trace ウィ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ィンドウをスクロールダウンしてください。Trace ウィンドウが以下のように表示されます。

Cycle	Address	Label	Code	Data	R/w	Area	Status	Clock	Probes	IRQ	Source
-00009	00109c			0dbe	RD	ROM	PROG	1	1111		
-00008	0010d8	CMP.w	1d63	RD	ROM	PROG	1	1111			
-00007	0010da	BLT	4dbc	RD	ROM	PROG	1	1111			
-00006	0010dc			0d66	RD	ROM	PROG	1	1111		
-00005	001098	MOV.w	0d3b	RD	ROM	PROG	1	1111			
-00004	00109a	BRA	4032	RD	ROM	PROG	1	1111			
-00003	00109c			0dbe	RD	ROM	PROG	1	1111		
-00002	0010ce	ADD.w	096b	RD	ROM	PROG	1	1111			
-00001	0010d0			792b	RD	ROM	PROG	1	1111		
+00000	0010d2			000a	RD	ROM	PROG	1	1111		

図4.36 Trace ウィンドウ

- 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。
- Cycle カラムの -00005 を見ると、アドレス H'1098 がアクセスされていることを確認できます。

4.9.2 トレースフィルタの設定

現在 Trace ウィンドウは、すべての MCU サイクルを表示しています。

- マウスの右ボタンで Trace ウィンドウをクリックし、ポップアップメニューより [Filter...] を選択して、Trace Filter ダイアログボックスを表示してください。
- 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。
 - [Type] セクションで [Pattern] 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'1098 と入力してください。

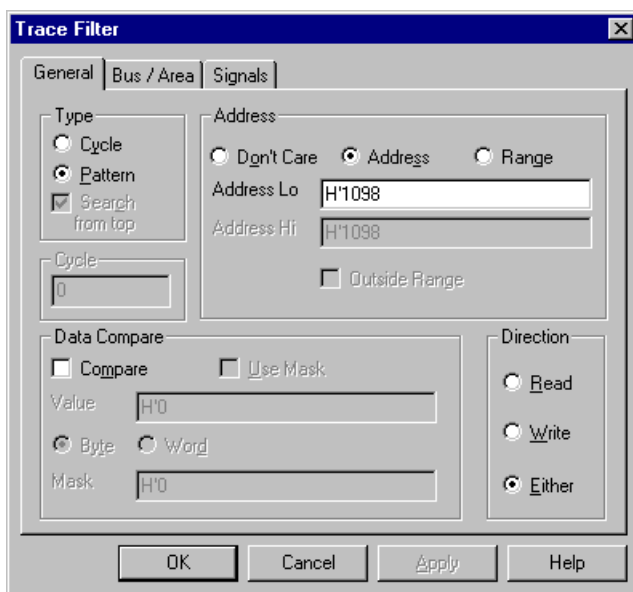


図4.37 Trace Filter ダイアログボックス

- [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。

4. チュートリアル

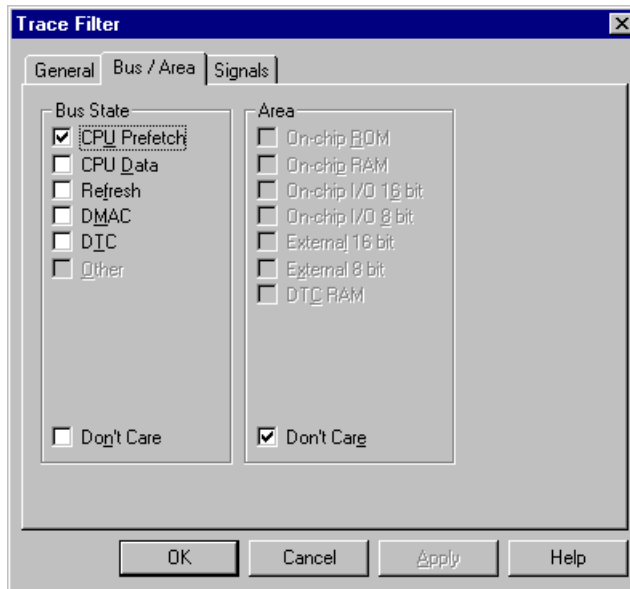


図4.38 Bus / Area の設定

- [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。
Trace ウィンドウには、MCU が H'1098 番地をアクセスしたサイクルだけが表示されます。これにより H'1098 への 5 回のアクセスによりプログラムが停止したことがわかります。

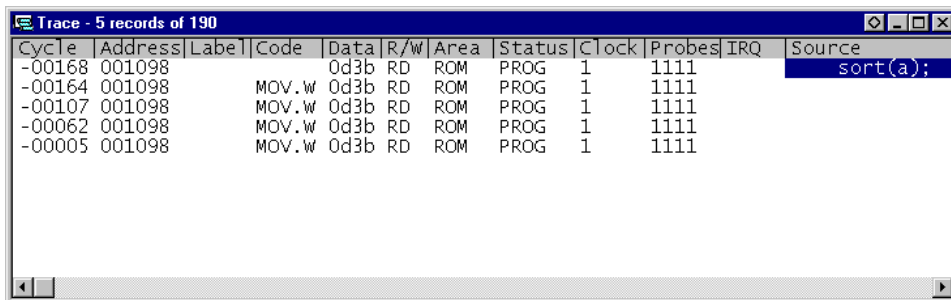



図4.39 Trace ウィンドウ (トレースフィルタ指定)

4.10 プログラム実行時間効率の計測

HDI では Performance Analysis 機能を使用することにより、プログラムが効率よく実行されているかを計測することができます。計測結果はヒストグラムまたはパーセント表示で確認できます。

4.10.1 計測条件の選択

計測を行う為に条件を選択してください。

- [View]メニューから [Breakpoints] を選択して、Breakpoints ウィンドウを開き、すべてのブレークポイントを削除してください。
- [View]メニューから [Performance Analysis] を選択するか、あるいはツールバーの [PA] ボタン  をクリックして、Performance Analysis ダイアログボックスを開いてください。
- [Conditions] ボタンをクリックして、Performance Analysis Conditions ダイアログボックスを開いてください。
- [Performance Analysis Conditions] の "No 1" をクリックした後、[Edit] ボタンをクリックして、Performance Analysis Properties ダイアログボックスを開いてください。

以下のダイアログボックスが現われ、計測条件を設定できます。

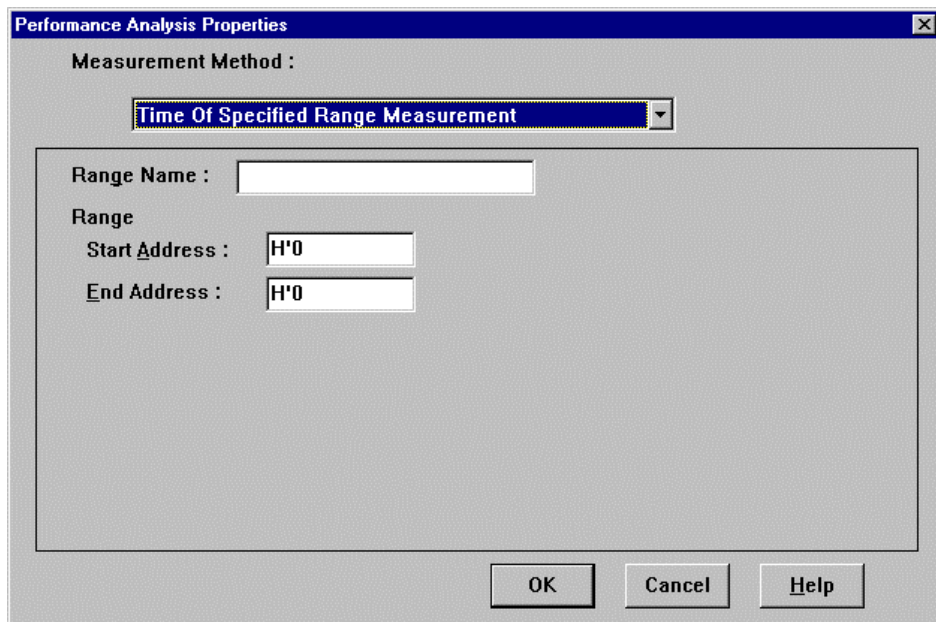


図4.40 計測条件の選択

- 指定範囲内でのプログラム実行時間効率を計測するため [Measurement Method] から [Time Of Specified Range Measurement] を選択してください。
 - [Range Name] に、"Analysis" を入力してください。
 - 指定範囲として [Start Address] にアドレス H'1088、 [End Address] にアドレス H'10ec を入力してください。
 - [OK] をクリックして、計測条件を設定してください。
- これにより、"No 1" に対する計測条件が設定されました。

4. チュートリアル

Performance Analysis Conditions ダイアログボックスは、設定条件を表示しています。

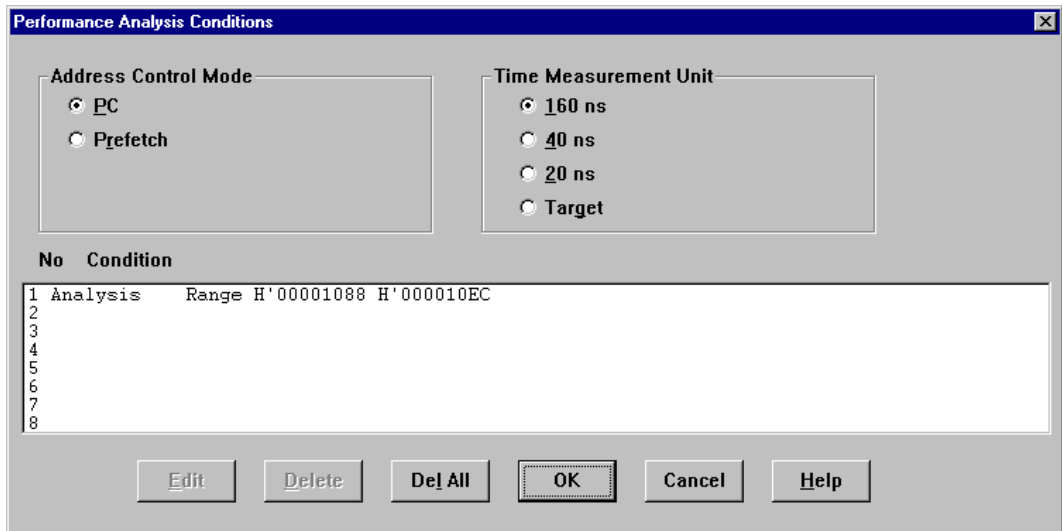




図4.41 計測条件の表示

- [**OK**] をクリックして、計測条件を設定してください。
これにより、プログラムがアドレス H'1088 と H'10ec の範囲を実行した時の実行効率が計測できます。
- [**Close**] をクリックして、Performance Analysis ダイアログボックスを閉じてください。
- Program ウィンドウから H'1082 番地を含むラインの [**BP**] カラムをダブルクリックして PC Break を設定してください。
- [**Run**] メニューから [**Reset Go**] を選択するか、あるいはツールバーの [**Reset Go**] ボタン  をクリックして、プログラムを最初の位置から実行してください。
アドレス H'1082 で実行が停止します。

4.10.2 計測結果の表示

プログラム実行時間効率の計測結果をヒストグラムまたはパーセントで表示します。

- [View]メニューから [Performance Analysis] を選択するか、あるいはツールバーの [PA] ボタン  をクリックして、Performance Analysis ダイアログボックスを開いてください。

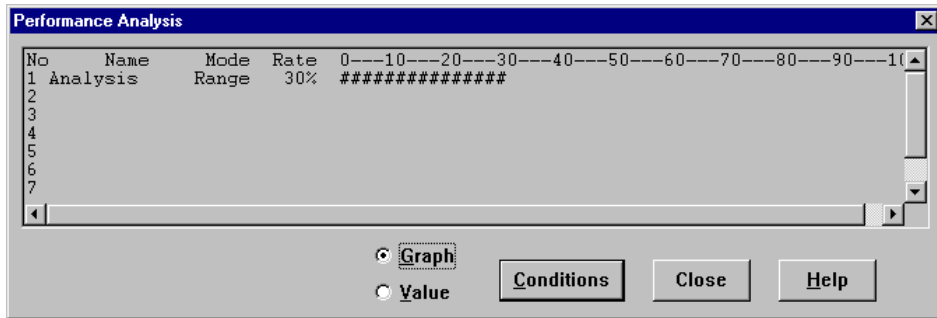


図4.42 計測結果の表示 (1)

プログラム実行時間効率の計測結果をヒストグラムとパーセントで表示します。

- [Value] をクリックしてください。

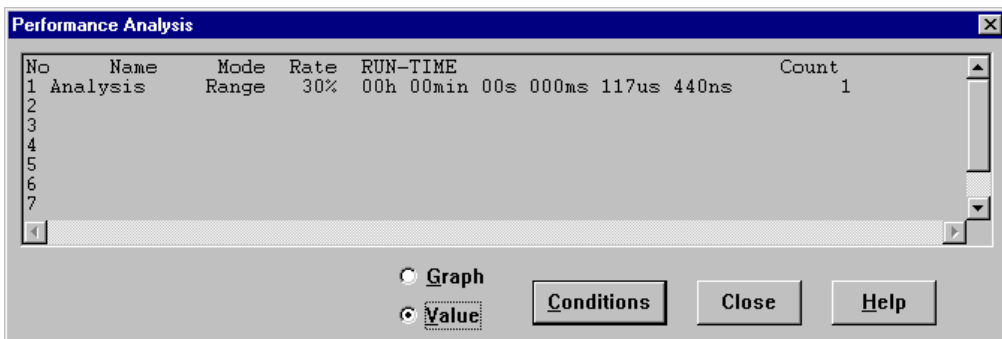


図4.43 計測結果の表示 (2)

プログラム実行時間効率の計測結果をパーセントと計測時間で表示します。

4.11 バスモニタ

バスモニタを使用するとユーザプログラム実行中に、メモリ内容をリアルタイム(ウインドウを更新する最小時間間隔は1秒)で参照することができます。

- [View]メニューから[Bus Monitor Window...]を選択してください。

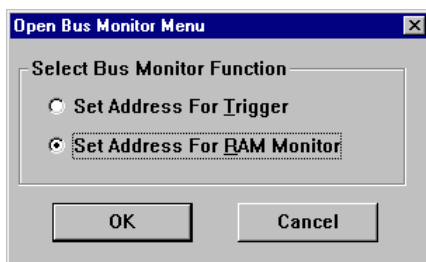


図4.44 バスモニタの機能選択ダイアログボックス

- Select Bus Monitor Function から RAM モニタ機能 (Set Address for RAM Monitor) を選択して [OK]をクリックしてください。
- Monitor1 左のチェックボックスをチェックし、H'fffa00 と入力、Access を選択し、[OK]ボタンをクリックしてください。

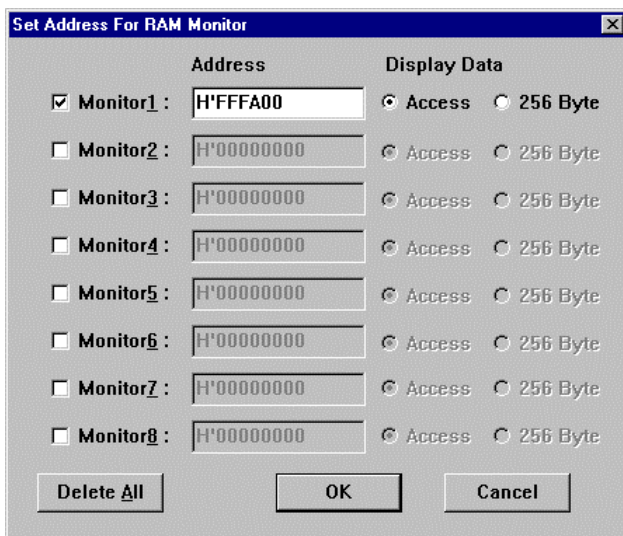



図4.45 RAM モニタのアドレス設定ダイアログボックス

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

以下のウインドウで、メモリ内容が変更される様子をリアルタイム(ウインドウを更新する最小時間間隔は1秒)で見ることができます。

本チュートリアルでは、PC Break でプログラムが停止するため、アクセスがあったアドレスのみ確認できます。

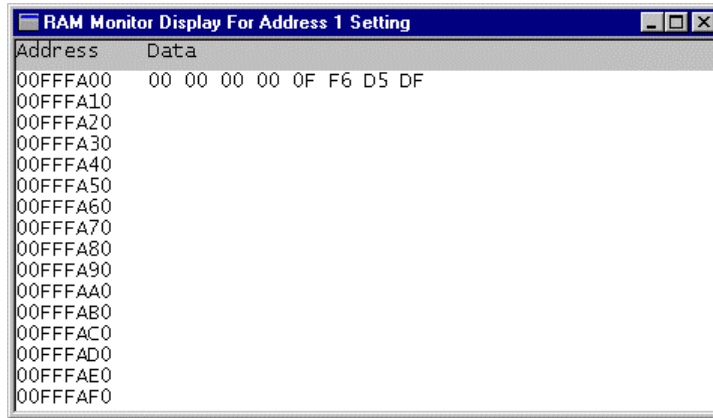



図4.46 RAM モニタの表示ウインドウ

4.12 スタックトレース機能

スタックトレース機能を使用すると、ユーザプログラム停止時に関数の呼び出し履歴を確認できます。

- Program ウィンドウから H'108e 番地を含むラインの [BP] カラムをダブルクリックして PC Break を設定してください。
- [Run] メニューから [Reset Go] を選択するか、あるいはツールバーの [Reset Go] ボタン  をクリックして、プログラムを最初の位置から実行してください。

設定した PC Break によりアドレス H'108e で実行が停止します。

- [View]メニューから[Stack Trace]を選択し[Stack Trace]ウインドウを開いてください。

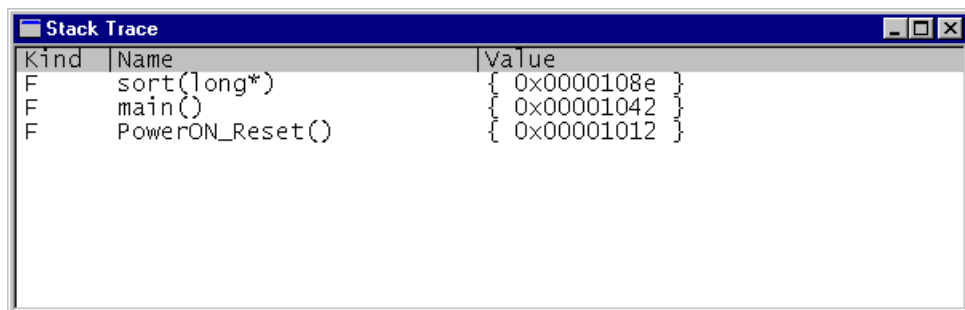


図4.47 Stack Trace ウィンドウ

現在 PC が sort 関数内にあり、main 関数から呼び出されていることがわかります。

【注】 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。

これまで説明いたしました各機能の詳細とその他の機能につきましては、オンラインヘルプをご覧ください。オンラインヘルプは、各ウインドウで [Help] ボタンまたは [F1] キーを押すと表示されます。

4.13 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和良好的でしょう。

- [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 を終了してください。

4.14 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機能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。

付録 A コマンド一覧表

E6000 エミュレータで使用できるコマンドの一覧表を示します。

種別 共通：HDI 共通のコマンド

特有：E6000 エミュレータ特有のコマンド

HDI 共通のコマンドについては、「日立デバッグインタフェースユーザズマニュアル」またはオンラインヘルプをご覧ください。E6000 エミュレータ特有のコマンドについては、オンラインヘルプをご覧ください。オンラインヘルプの使用方法は、Command Line ウィンドウで以下のように入力します。

```
help <command>
```

<command> : コマンド名または短縮形

表 A.1 コマンド一覧表

コマンド名	短縮形	種別	説明
!	-	共通	コメント
ACCESS	AC	共通	不当アクセスに対する動作の設定
ANALYSIS	AN	特有	性能分析機能の有効化 / 無効化
ANALYSIS_RANGE	AR	特有	性能分析範囲の設定と表示
ANALYSIS_RANGE_DELETE	AD	特有	性能分析範囲の解除
ASSEMBLE	AS	共通	アセンブルの実行
ASSERT	-	共通	コンディションのチェック
BREAKPOINT EVENT	BP EN	特有	ブレークポイント / イベントの設定
BREAKPOINT_CLEAR EVENT_CLEAR	BC EC	特有	ブレークポイント / イベントの解除
BREAKPOINT_DISPLAY EVENT_DISPLAY	BD ED	特有	ブレークポイント / イベントの表示
BREAKPOINT_ENABLE EVENT_ENABLE	BE EE	特有	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	特有	シーケンスの定義および解除
CLOCK	CK	特有	エミュレータの MCU クロック時間の設定
DEVICE_TYPE	DE	特有	エミュレータのデバイスタイプの選択
DISASSEMBLE	DA	共通	逆アセンブル表示
ERASE	ER	共通	Command Line ウィンドウの内容のクリア
EVALUATE	EV	共通	式の計算
FILE_LOAD	FL	共通	オブジェクト (プログラム) ファイルのロード
FILE_SAVE	FS	共通	メモリ内容のファイルセーブ
FILE_VERIFY	FV	共通	ファイル内容とメモリ内容の比較
GO	GO	共通	ユーザプログラムの実行

付録 A コマンド一覧表

コマンド名	短縮形	種別	説明
GO_RESET	GR	共通	リセットベクタからのユーザプログラムの実行
GO_TILL	GT	共通	テンポラリブレークポイントまでユーザプログラムの実行
HALT	HA	共通	ユーザプログラムの停止
HELP	HE	共通	コマンドラインまたはコマンドに対するヘルプ表示
INITIALISE	IN	共通	プラットフォームの初期化
LOG	LO	共通	ロギングファイルの操作
MAP_DISPLAY	MA	共通	メモリマッピング情報の表示
MAP_SET	MS	特有	メモリマッピングの設定
MEMORY_DISPLAY	MD	共通	メモリ内容の表示
MEMORY_EDIT	ME	共通	メモリ内容の変更
MEMORY_FILL	MF	共通	指定データによるメモリ内容の一括変更
MEMORY_MOVE	MV	共通	メモリブロックの移動
MEMORY_TEST	MT	共通	メモリブロックのテスト
MODE	MO	特有	MCU モードの設定と表示
MODULES	MU	特有	デバイスの内蔵周辺機能の設定と表示 (サポートしない製品もあります)
QUIT	QU	共通	HDI の終了
RADIX	RA	共通	入カラディックスの設定
REFRESH	RF	特有	メモリ関連ウインドウの更新
REGISTER_DISPLAY	RD	共通	MCU レジスタ値の表示
REGISTER_SET	RS	共通	MCU レジスタ値の設定
RESET	RE	共通	MCU のリセット
SLEEP	-	共通	コマンド実行の遅延
STEP	ST	共通	ステップ実行 (命令単位またはソース行単位)
STEP_OUT	SP	共通	PC 位置の関数を終了するまでのステップ実行
STEP_OVER	SO	共通	ステップオーバー実行
STEP_RATE	SR	共通	ステップ速度の設定
SUBMIT	SU	共通	エミュレータコマンドファイルの実行
SYMBOL_ADD	SA	共通	シンボルの追加
SYMBOL_CLEAR	SC	共通	シンボルの削除
SYMBOL_LOAD	SL	共通	シンボル情報ファイルのロード
SYMBOL_SAVE	SS	共通	シンボル情報のファイルセーブ
SYMBOL_VIEW	SV	共通	シンボルの表示
TEST_EMULATOR	TE	特有	エミュレータハードウェアのテスト
TIMER	TI	特有	実行時間測定タイマ分解能の表示、設定
TRACE	TR	共通	トレース情報の表示
TRACE_ACQUISITION	TA	特有	トレース取得情報の設定と表示
TRACE_COMPARE	TC	特有	トレース情報の比較
TRACE_SAVE	TV	特有	トレース情報の保存
TRACE_SEARCH	TS	特有	トレース情報の検索
USER_SIGNALS	US	特有	ユーザーシグナル情報の有効化 / 無効化

【注】 パスモニタは、コマンドで使用することはできません。

H8S 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-702-269A