

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

SH7058 E6000H エミュレータ HS7058EPH62H
SH7058S、SH7059 E6000H エミュレータ HS7059EPH62H
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
SuperH™ファミリ／SH7050 シリーズ

HS7058EPH62HJ

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ(<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。
5. 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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およ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ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いいたします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 本資料に関する詳細についてのお問い合わせ、その他お気づきの点等がございましたら弊社営業窓口までご照会ください。

重要事項

- 本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。
ユーザーズマニュアルをよく読まずに、本エミュレータを使用しないでください。

エミュレータとは：

ここで言うエミュレータとは、株式会社ルネサス テクノロジ（以下、「ルネサス」という。）が製作した次の製品を指します。
（1）エミュレータ本体、（2）PC インタフェースボード、（3）ユーザシステムインタフェースボード、（4）ケーブル
お客様のユーザシステムおよ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：

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

この使用目的に従って、本エミュレータを正しく使用してください。目的以外の本エミュレータの使用を堅くお断りします。

使用制限：

本エミュレータは、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

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

製品の変更について：

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

エミュレータを使用する人は：

本エミュレータは、ユーザーズマニュアルをよく読み、理解した人のみをご使用ください。

ユーザーズマニュアルをよく読まずに、本エミュレータを使用しないでください。

特に、本エミュレータを初めて使用する人は、本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強く勧めます。

保証の範囲：

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

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

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

その他の重要事項：

1. 本資料に記載された情報、製品または回路の使用に起因する損害または特許権その他権利の侵害に関しては、ルネサスは一切その責任を負いません。
2. 本資料によって第三者またはルネサスの特許権その他権利の実施権を許諾するものではありません。

著作権所有：

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

図について：

このユーザーズマニュアルの一部の図は、実物と違っていることがあります。

予測できる危険の限界：

ルネサスは、潜在的な危険が存在するおそれのあるすべての起こりうる諸状況や誤使用を予見できません。したがって、このユーザーズマニュアルと本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、本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

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

シグナル・ワードの定義



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



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



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



注意は、回避しないと、軽傷または中程度の傷害を招く可能性がある潜在的に危険な状況を示します。



安全警告記号の付かない注意は、回避しないと財物傷害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。



1. DC電源を内蔵していますので、触れる場所によっては感電する可能性があります。感電、火災等の危険防止および品質保証のために、お客様ご自身による修理や改造は行なわないでください。故障の際のアフターサービスにつきましては、ルネサスまたはルネサス特约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。
3. 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. エミュレータまたはユーザシステムのパワーオン時、エバチップボードまたはユーザシステムインタフェースボードとユーザシステム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の破壊の可能性があります。
5. エバチップボードまたはユーザシステムインタフェースボード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。
6. 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。

エミュレータ使用時の注意事項

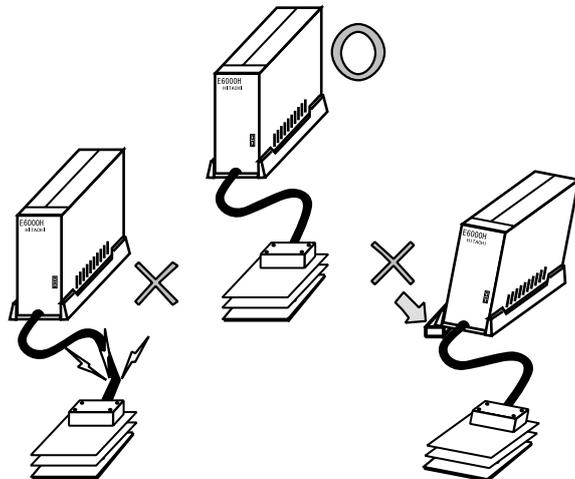
このエミュレータ使用時の注意事項に記載されている事項は、本エミュレータを使用するうえですべての場合に該当し、例外は存在しません。したがって、エミュレータを使用する前に以下に示されている警告文をよく読み、完全に理解してください。ただし、ここに記載されている事項はエミュレータ使用時における共通の警告のみが記載されており、これがエミュレータを使用するうえでのすべての警告ではありません。

警告

エミュレータまたはユーザシステムのパワーオン時には、すべてのケーブル、およびユーザインターフェースの抜き差しを行わないでください。抜き差しを行なった場合、エミュレータとユーザシステムの発煙発火、および機器の破壊の可能性があります。また、デバッグ中のユーザプログラムの破壊の可能性があります。

注意

エミュレータ本体とエバチップボードの位置関係により、トレースケーブルが大きく曲げられることがあります。このような状態で使用すると、ユーザインタフェース部に大きなストレスが加わり、接点、接触不良等の機械的破壊を招く原因となります。また、使用中にエミュレータ本体が動いてしまうと、ユーザインタフェース部に思わぬストレスを与えることになります。エミュレータ本体の設置位置に十分ご注意ください。



はじめに

このたびは、E6000H エミュレータをお買い上げ頂き、誠にありがとうございます。

注意

本エミュレータをご使用になる前に、必ず本マニュアルをよく読み、理解してください。誤った使用方法、接続方法は、本エミュレータ、ユーザプログラム、ユーザシステムの破壊につながります。

E6000H エミュレータは、ルネサスオリジナルマイクロコンピュータを使用したシステムの開発をソフトウェア、ハードウェアの両面からサポートする支援装置です。E6000H 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。

E6000H エミュレータの特徴は以下の通りです。

1. リアルタイムエミュレーション

リアルタイムにターゲットマイコンをエミュレーションすることができます。

2. 充実したデバッグ機能

ブレーク機能の充実、トレースの大容量化（128kサイクル）によりデバッグ効率を向上させます。

3. パラレルアクセス機能

エミュレーション実行中にトレース内容を表示したり、エミュレーションメモリの内容を参照、変更することができます。

4. パフォーマンス測定機能

サブルーチンの実行時間、実行回数測定などユーザプログラムの実行効率を測定することができます。

5. グラフィカルユーザインタフェースを提供

Windows[®]上で動作するHigh-performance Embedded Workshopを使用したグラフィカルユーザインタフェースを提供します。

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

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

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

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

このマニュアルについて

本マニュアルではハードウェア編、およびデバッグ編の2部構成になっています。それぞれ以下の内容を説明しています。

ハードウェア編 使用前の準備、ハードウェア仕様、故障診断解析

デバッグ編 エミュレータ固有の機能、エミュレータ固有のデバッグ機能、チュートリアル、エミュレータソフトウェア仕様
および注意事項

High-performance Embedded Workshop の基本的な使い方に関する情報、High-performance Embedded Workshop 環境のカスタマイズ、High-performance Embedded Workshop のビルド機能、および各 High-performance Embedded Workshop 製品で共通なデバック機能については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

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

Microsoft、MS-DOS、Windows、Windows NT は米国 Microsoft 社の米国およびその他の国における登録商標です。

Visual SourceSafe は Microsoft 社の米国およびその他の国における商標です。

IBM は International Business Machines Corporation の登録商標です。

その他、記載されている製品名は各社の商標または登録商標です。

このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1 記号一覧

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

梱包品の確認

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

旧製品との互換性について

HS7058EPH62H、HS7059EPH62H は、HS7058EPH60H (保守製品)、HS7059EPH60H(保守製品)の後継機です。エバチップボードのユーザ接続用コネクタを変更しているため、保守製品のユーザシステムインタフェースボードは使用することが出来ません。

別途、専用のユーザシステムインタフェースボードをお買い求めいただく必要があります。

目次

ハードウェア編	1
1. 概要	3
1.1 使用前の注意	3
1.2 エミュレータハードウェア構成	4
1.2.1 E6000H 本体の構成（本マニュアルの写真は一部実物と異なります）	5
1.2.2 エバチップボードの構成	7
1.2.3 ユーザシステムインタフェースボードの構成	8
1.3 システム構成	9
1.3.1 PC インタフェースボードによるシステム構成	9
2. 使用前の準備	11
2.1 E6000Hエミュレータ使用までのフローチャート	11
2.2 エミュレータの接続	11
2.2.1 ユーザシステムとの接続	11
2.2.2 ユーザシステムインタフェースボードの接続	12
2.2.3 外部プローブ接続用コネクタ	13
2.2.4 クロックの選択	14
2.2.5 システムグランド系の接続	15
2.2.6 PC インタフェースボードについて	16
3. ハードウェア仕様	17
3.1 使用環境条件	17
3.2 エミュレータの外形寸法と質量	18
3.3 ユーザシステムインタフェース回路	19
3.3.1 ユーザシステムインタフェース回路	19
3.3.2 ユーザシステムインタフェースのディレイ時間	25
3.4 エミュレータとユーザシステムの接続	25
3.4.1 ユーザシステムとの接続方法	25
3.4.2 ユーザシステムインタフェースコネクタのピン配置	32
3.4.3 ユーザシステム接続時の注意事項	42
3.5 ターゲットマイコンのサポート	43
3.5.1 メモリ空間	43
3.5.2 低消費電力状態（スリープ、ソフトウェアスタンバイ、ハードウェアスタンバイ）	44
3.5.3 割り込み	44
3.5.4 コントロール用入力信号（_RES、_BREQ、_WAIT）	44
3.5.5 WDT（ウォッチドッグタイマ）	44
3.5.6 A/D 変換器	45
3.5.7 E6000H エミュレータの状態と内蔵モジュール	45
3.5.8 エミュレータのレジスタ初期値相違点	46
4. 故障解析（故障症状調査書）	47
4.1 故障解析	47
4.1.1 診断プログラムを実行するためのシステムセットアップ	47
4.1.2 診断プログラムのテスト項目	48
4.1.3 診断プログラムによる故障解析	49
4.2 故障症状調査書	54

デバッグ編.....	1
1. 概要	3
2. デバッグの準備をする.....	5
2.1 High-performance Embedded Workshopの起動方法	5
2.1.1 新規にワークスペースを作成する場合（ツールチェイン未使用）	6
2.1.2 新規にワークスペースを作成する場合（ツールチェイン使用）	8
2.1.3 既存のワークスペースを指定する場合.....	12
2.2 エミュレータの接続	13
2.3 エミュレータの再接続	14
2.4 エミュレータの終了	14
3. デバッグ	15
3.1 エミュレーション環境を設定する.....	15
3.1.1 コンフィグレーションプロパティダイアログボックスを開く	15
3.1.2 エミュレーション RAM を使用する.....	18
3.1.3 接続するインタフェースを選択する.....	20
3.2 プログラムをダウンロードする.....	21
3.2.1 プログラムをダウンロードする.....	21
3.2.2 ソースコードを表示する	21
3.2.3 アセンブリ言語コードを表示する.....	23
3.2.4 アセンブリ言語コードを修正する.....	24
3.2.5 特定のアドレスを見る	24
3.2.6 現在のプログラムカウンタアドレスを見る	24
3.3 現在の状態を表示する	25
3.4 エミュレータの情報を定期的に読み出し表示する	26
3.4.1 [拡張モニタ]ウィンドウを開く	26
3.4.2 表示項目を選択する	27
3.5 リアルタイムにメモリ内容を表示する.....	27
3.5.1 モニタウィンドウを開く	28
3.5.2 モニタの設定内容を変更する.....	30
3.5.3 モニタの更新を一時的に停止する.....	30
3.5.4 モニタ設定を削除する	30
3.5.5 変数の内容をモニタする	30
3.5.6 モニタウィンドウを非表示にする.....	30
3.5.7 モニタウィンドウを管理する.....	31
3.6 変数の表示	32
3.6.1 ウォッチウィンドウ	32
3.7 イベントポイントを使用する.....	33
3.7.1 ソフトウェアブレークポイントを設定する	34
3.7.2 オンチップブレークポイントを設定する.....	36
3.7.3 オンエミュレータブレークポイントを設定する	38
3.7.4 イベントポイントの編集	40
3.7.5 イベントポイントの設定内容を変更する.....	40
3.7.6 イベントポイントを有効にする.....	40
3.7.7 イベントポイントを無効にする.....	40
3.7.8 イベントポイントを削除する.....	40
3.7.9 イベントポイントをすべて削除する.....	40
3.7.10 イベントポイントのソース行を表示する.....	40
3.8 トレース情報を見る	41
3.8.1 トレースウィンドウを開く.....	41
3.8.2 トレース情報を取得する	41

3.8.3	トレース情報取得条件を設定する.....	43
3.8.4	Trace レコードを検索する.....	48
3.8.5	トレース情報をクリアする.....	49
3.8.6	トレース情報をファイルに保存する.....	49
3.8.7	[エディタ]ウィンドウを表示する.....	49
3.8.8	ソース表示を整形する.....	49
3.8.9	トレース情報の取得を一時的に停止する.....	50
3.8.10	トレース情報の取得を再開する.....	50
3.8.11	取得したトレース情報から必要なレコードを抽出する.....	50
3.8.12	タイムスタンプの差を計算する.....	52
3.8.13	統計情報を解析する.....	52
3.8.14	取得したトレース情報から関数呼び出し箇所を抽出する.....	54
3.9	パフォーマンスを測定する.....	55
3.9.1	パフォーマンス解析ウィンドウを開く.....	56
3.9.2	実行効率測定条件を設定する.....	57
3.9.3	実行効率測定を開始する.....	62
3.9.4	測定条件を削除する.....	62
3.9.5	すべての測定条件を削除する.....	63
3.10	プロファイル機能を使用する.....	63
3.10.1	プロファイルを有効にする.....	63
3.10.2	測定方法を指定する.....	63
3.10.3	ユーザプログラムを実行し結果を確認する.....	63
3.10.4	List シート.....	63
3.10.5	Tree シート.....	64
3.10.6	ソースファイル表示.....	64
3.10.7	チャート表示.....	65
3.10.8	有効.....	65
3.10.9	関数呼び出しをトレースしない.....	65
3.10.10	検索.....	65
3.10.11	データ検索.....	65
3.10.12	データクリア.....	65
3.10.13	プロファイル情報の保存.....	65
3.10.14	テキスト形式で保存.....	65
3.10.15	表示設定.....	66
3.10.16	プロパティ.....	66
3.11	プロファイル-チャートウィンドウ.....	66
3.11.1	ソースファイル表示.....	67
3.11.2	チャート表示.....	67
3.11.3	有効.....	67
3.11.4	データクリア.....	67
3.11.5	チャートウィンドウを複数開く.....	67
3.11.6	プロファイル情報の保存.....	67
3.11.7	拡大.....	67
3.11.8	縮小.....	67
4.	チュートリアル.....	69
4.1	はじめに.....	69
4.2	High-performance Embedded Workshopの起動.....	70
4.3	チュートリアルプログラムのダウンロード.....	71
4.3.1	チュートリアルプログラムをダウンロードする.....	71
4.3.2	ソースプログラムを表示する.....	72
4.4	ソフトウェアブレークポイントの設定.....	73

4.5	レジスタ内容の変更	74
4.6	プログラムの実行	75
4.7	ブレークポイントの確認	77
4.8	シンボルの参照	78
4.9	メモリ内容の確認	79
4.10	変数の参照	80
4.11	ローカル変数の表示	83
4.12	プログラムのステップ実行.....	83
4.12.1	ステップインコマンドの実行.....	84
4.12.2	ステップアウトコマンドの実行.....	85
4.12.3	ステップオーバコマンドの実行.....	86
4.13	プログラムの強制ブレーク.....	87
4.14	ターゲットマイコンのリセット.....	87
4.15	ブレーク機能	88
4.15.1	ソフトウェアブレーク機能.....	88
4.15.2	オンチップブレーク機能	93
4.16	トレース機能	95
4.16.1	フリートレースによるトレース情報の表示.....	96
4.16.2	トレース停止によるトレース情報の表示.....	97
4.16.3	条件トレースによるトレース情報の表示.....	100
4.16.4	統計	101
4.16.5	関数コール	104
4.17	スタックトレース機能	105
4.18	パフォーマンス測定機能	107
4.18.1	指定範囲内時間測定	107
4.19	プロファイル機能	110
4.20	モニタ機能	115
4.21	さてつぎは?	117
5	本製品固有のソフトウェア仕様と注意事項.....	119
5.1	対応ハードウェアについて.....	119
5.2	デバッグプラットフォームについて.....	119
5.3	メモリ内容の表示と変更方法について.....	119
5.3.1	実行中のメモリ内容の表示と変更.....	119
5.3.2	パラレルアクセス機能停止時間参考値.....	120
5.3.3	モニタ機能	120
5.3.4	メモリアクセスの注意事項.....	120
5.4	プログラムの実行について.....	121
5.4.1	ステップ実行	121
5.4.2	ブレークについて	121
5.5	イベント機能について	121
5.5.1	ソフトウェアブレーク	121
5.5.2	オンチップブレーク	122
5.5.3	オンエミュレータブレーク.....	122
5.6	トレース機能について	123
5.6.1	トレース表示内容	123
5.6.2	トレース取得条件	123
5.6.3	トレース情報の検索	123
5.6.4	トレース情報のフィルタ	123
5.7	モニタ機能について	123
5.8	パフォーマンス測定機能について.....	124
5.8.1	測定誤差について	124
5.8.2	留意事項	124
5.9	プロファイル機能について.....	124
5.10	入力形式について	125

5.10.1	マスク入力	125
5.11	プログラムのダウンロードについて	125
5.12	チュートリアルプログラムについて	126
5.12.1	チュートリアルプログラムのダウンロード	126
5.12.2	チュートリアルプログラムの操作についての注意事項	126
6.	エラーメッセージ	127
6.1	E6000Hのエラーメッセージ	127
6.1.1	起動時のエラーメッセージ	127
6.1.2	実行中のエラーメッセージ	128
付録 A	メニュー一覧	129
付録 B	コマンドライン一覧	133

ハードウェア編

1. 概要

1.1 使用前の注意

注意

本エミュレータをご使用になる前に、以下の注意事項を必ずよく読んで理解してください。誤った使い方は、本エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

- (1) 製品を梱包箱から取り出し、納入品明細書に示されているものが揃っているか確認してください。
- (2) 製品に重量物を上積みするなどして、無理な力を加えないでください。
- (3) 次の条件を考慮して、E6000H本体を設置してください。
 - E6000H本体の背面に内部冷却用ファンがあります。E6000H本体と壁面や他の装置間は20cm以上離してください。
 - 直射日光に当たる場所や暖房機の近く等、高温となる場所に設置しないでください。詳細については、「3.1 使用環境条件」を参照してください。
 - 温度や湿度が極端に変化する場所に設置しないでください。
 - ホコリやチリが多い場所に設置しないでください
 - 振動が多い場所に設置しないでください。「3.1 使用環境条件」を参照してください。
- (4) 製品に過大な物理的衝撃や力を与えないでください。
- (5) E6000H本体に、指定された電圧、電源周波数以外の電源を供給しないでください。
- (6) 設置場所を移動する場合は、本製品に強い振動、衝撃が加わらないように注意してください。
- (7) ケーブルを接続した後は、接続位置が正しいことを再度確認してください。
- (8) 接続方法については、「2 使用前の準備」を参照してください。
- (9) 全てのケーブルを接続し終えてから、E6000Hエミュレータおよび接続した各装置へ電源を投入してください。
- (10) また、電源が入っているときにケーブルの接続および取り外しをしないでください。
- (11) ターゲットマイコンとエミュレータとの相違点など、エミュレーション時の注意事項は「3.5 ターゲットマイコンのサポート」を参照してください。

1.2 エミュレータハードウェア構成

E6000H エミュレータは、E6000H 本体、エバチップボードにより構成されています。ユーザシステムインタフェースボード(オプション)を実装することにより、デバイスと同一パッケージで接続することができます。PC インタフェース(オプション)として、PC インタフェースボード(PCIバス、PC カードバス)、LAN アダプタ(ネットワーク接続)および USB アダプタ(USB インタフェース接続)があります。

これらのインタフェースを用いて、E6000H とホストコンピュータを接続することで、High-performance Embedded Workshop を用いたデバッグが可能になります。

PCIバス、PC カードバス仕様の PC インタフェースボード、LAN アダプタおよび USB アダプタについては、各取扱い説明書をお読みください。

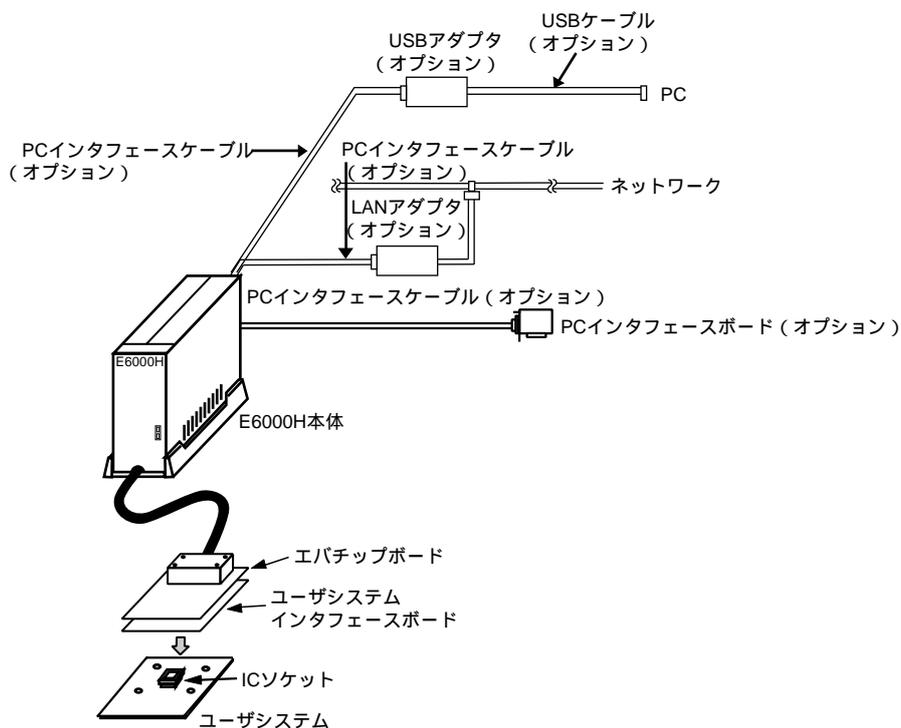


図1.1 エミュレータのハードウェア構成

1.2.1 E6000H 本体の構成（本マニュアルの写真は一部実物と異なります）

E6000H 本体における各部の名称を下記に示します。

(1) 前面パネルの構成



図1.2 E6000H 本体の前面パネル

各部名称

- | | | |
|---------------|---|-----------------------------|
| (a) POWER ランプ | : | E6000H 本体が電源 ON 状態のとき点灯します。 |
| (b) RUN ランプ | : | ユーザプログラムが実行中のとき点灯します。 |

(2) 背面パネルの構成

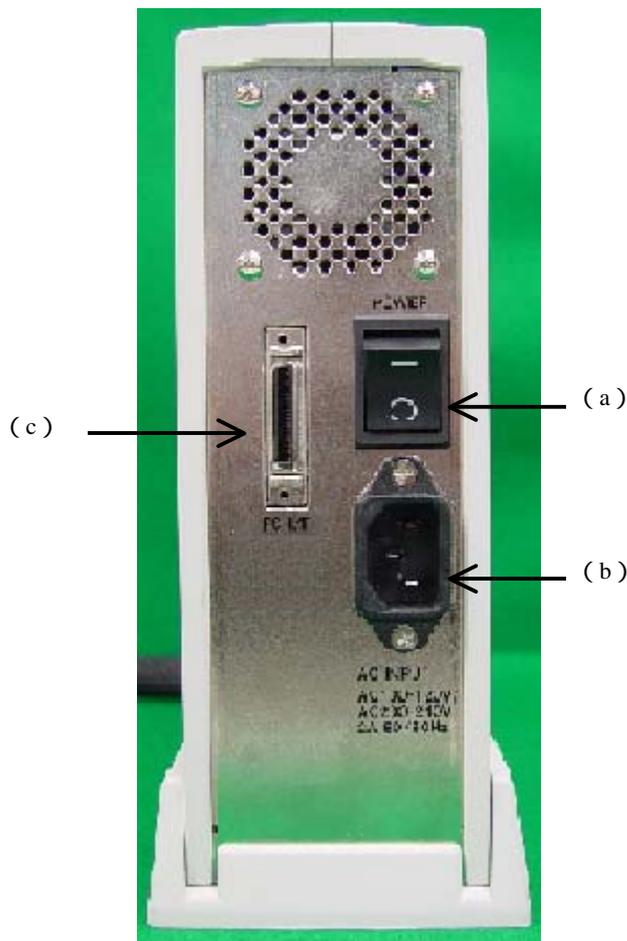


図1.3 E6000H 本体の背面パネル

各部名称

(a) POWER スイッチ

本スイッチを I (input) 側により、エミュレータ (E6000H 本体とエバチップボード) に電源が供給されます。

(b) AC 電源用コネクタ

AC100V 入力用コネクタです。

(c) PC インタフェースケーブル接続用コネクタ

ホストコンピュータと E6000H 本体を接続するインタフェースケーブルのコネクタです。ホスト側 PC I/F ボードとインタフェースを取ります。ここに PC インタフェースボード、PC カードインタフェース、LAN アダプタ、USB アダプタが接続されます。"PC I/F"と表示しています。

1.2.2 エバチップボードの構成

エバチップボードにおける各部の名称を下記に示します。

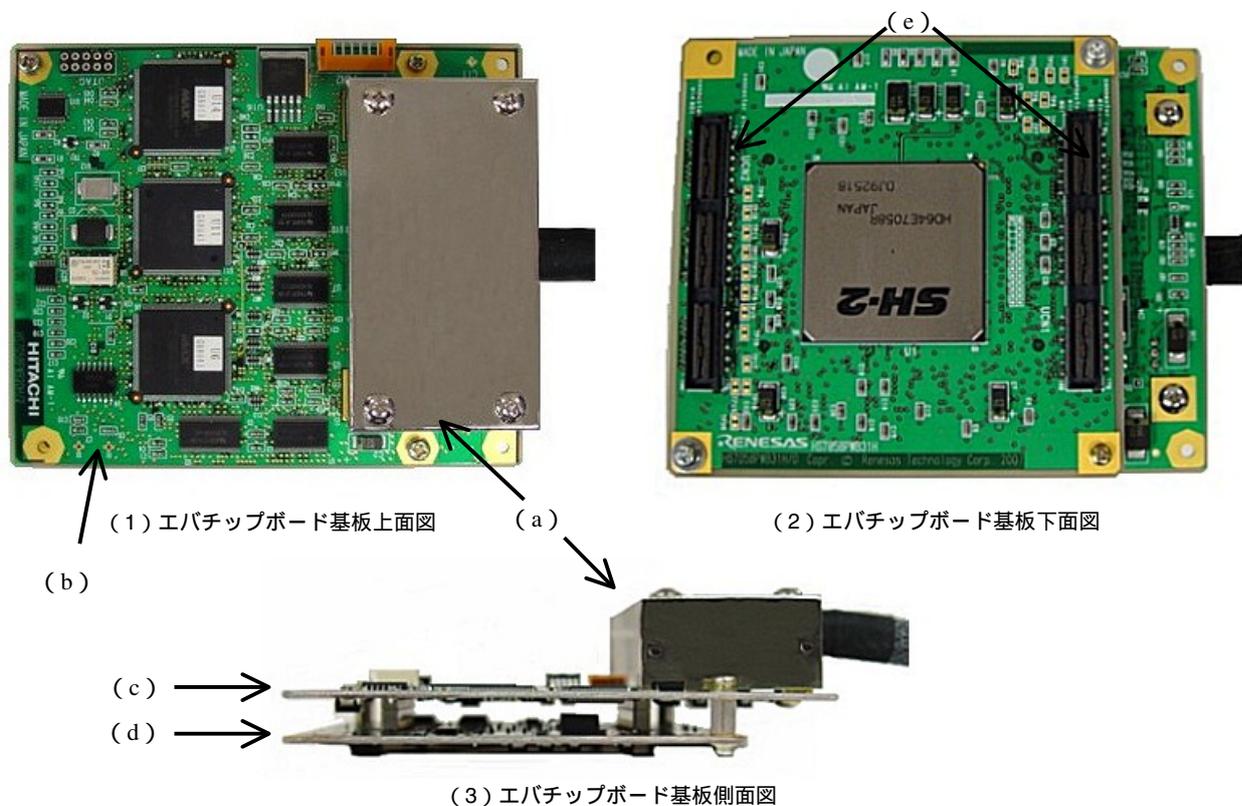


図1.4 エバチップボードの構成

各部名称

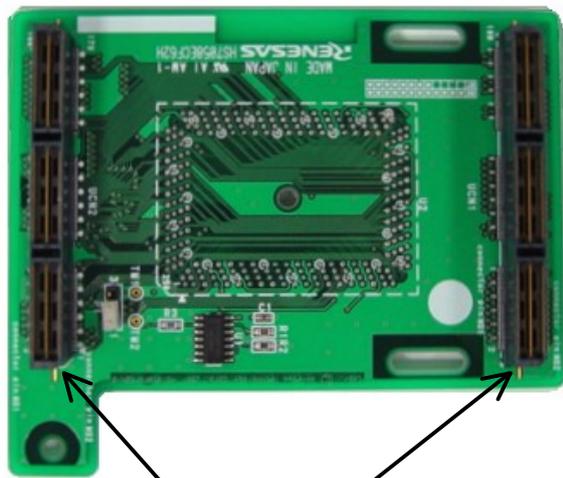
- | | |
|--|--|
| (a) 本体-エバチップボードインタフェース
コネクタカバー | : E6000H 本体とエバチップボードを接続するコネクタ部を保護するカバーです。 |
| (b) 水晶発振子取り付け端子 | : ターゲットマイコンに外部クロックを供給するための水晶発振子を取り付ける端子です。 |
| (c) HS7058PWB20H 基板 または
HS7059PWB20H 基板 | : トレースケールと接続するコネクタを持ちます。
(画像は HS7058PWB20H のものです) |
| (d) HS7058PWB31H 基板 または
HS7059PWB31H 基板 | : エバチップを実装し、ユーザシステムインタフェースボードまたはユーザシステムと
接続するコネクタ(専用コネクタ)を持ちます。
(画像は HS7058PWB31H のものです) |
| (e) ユーザシステムインタフェースボード接続
コネクタ | : ユーザシステムインタフェースボードまたはユーザシステムを接続するためのコネ
クタです。 |

【注】 上記 (a) ~ (e) を総称して、エバチップボードと呼びます。

1.2.3 ユーザシステムインタフェースボードの構成

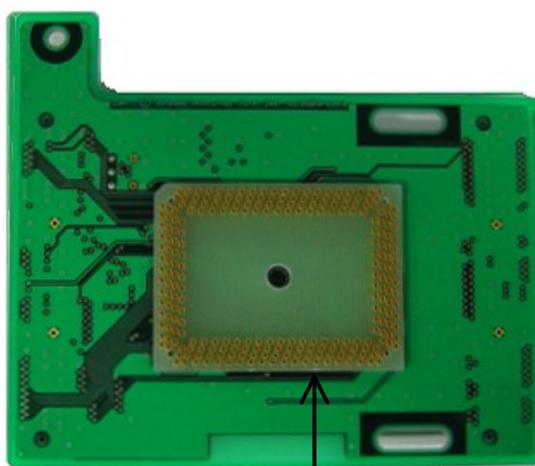
ユーザシステムインタフェースボードにおける各部の名称を下記に示します。

(1) ユーザシステムインタフェースボード基板上面図



(a)

(2) ユーザシステムインタフェースボード基板下面図



(b)

図1.5 エバチップボードの構成

各部名称

- | | | |
|---------------------|---|-------------------------|
| (a) エバチップボード接続用コネクタ | : | エバチップボードを接続するためのコネクタです。 |
| (b) ユーザシステム接続用コネクタ | : | ユーザシステムを接続するためのコネクタです。 |

画像は HS7058ECF62H のものです。HS7058ECB62H とは外観が異なりますが、各部名称は同一です。

1.3 システム構成

E6000H 本体は、ホストコンピュータ（オプションの PC インタフェースボード使用）と接続して使用してください。

1.3.1 PC インタフェースボードによるシステム構成

E6000H 本体は、PC インタフェースボード（オプション：PCI バス、PC カードバス）を介してホストコンピュータと接続することができます。PC インタフェースボードをホストコンピュータの各インタフェースボードに対応する拡張スロットなどに実装し、PC インタフェースボード付属のインタフェースケーブルで E6000H 本体と接続します。また、LAN アダプタは E6000H とネットワークを介してホストコンピュータと接続します。USB アダプタは E6000H とホストコンピュータを USB インタフェースで接続します。PCI バス、PC カードバス仕様の PC インタフェースボード、LAN アダプタおよび USB アダプタについては、各取扱い説明書をお読みください。PC インタフェースボードによるシステム構成を図 1.6 に、LAN アダプタによるシステム構成を図 1.7、USB アダプタによるシステム構成を図 1.8 に示します。

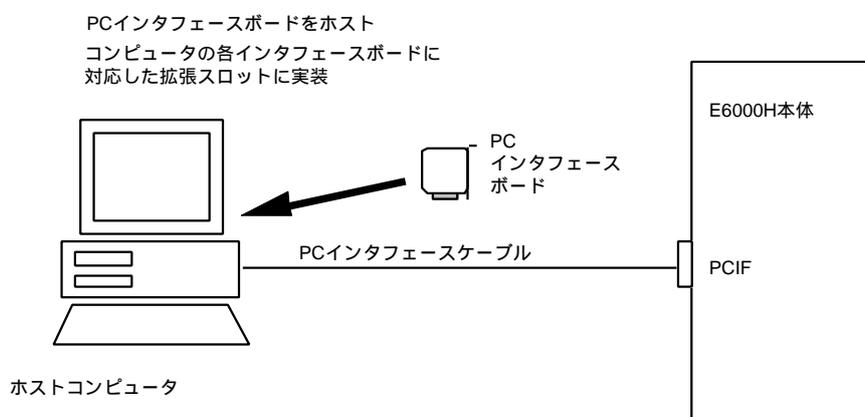


図1.6 PC インタフェースボードによるシステム構成

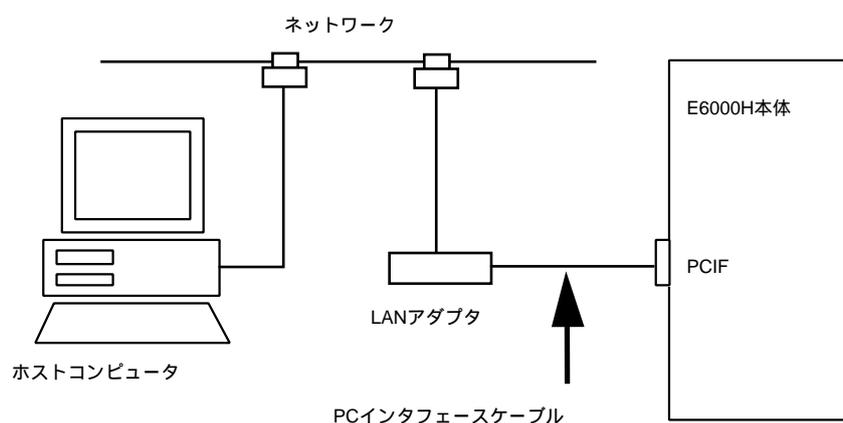


図1.7 LAN アダプタによるシステム構成

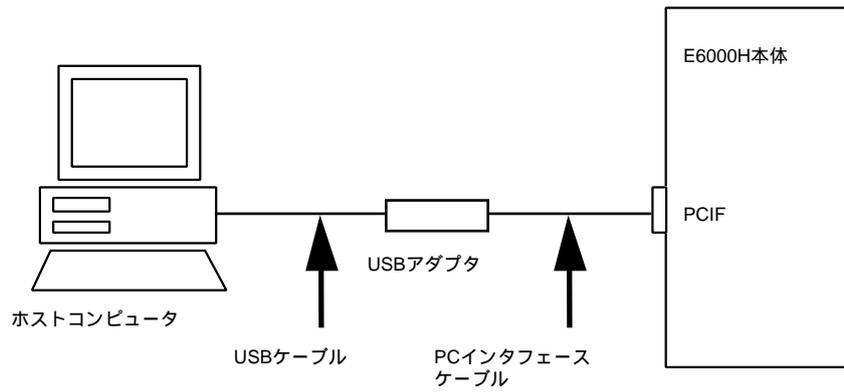


図1.8 USB アダプタによるシステム構成

2. 使用前の準備

2.1 E6000H エミュレータ使用までのフローチャート

本章では、E6000H エミュレータ使用前の準備について説明します。E6000H エミュレータを使用しデバッグを開始するまでのフローチャートを図 2.1 に示します。

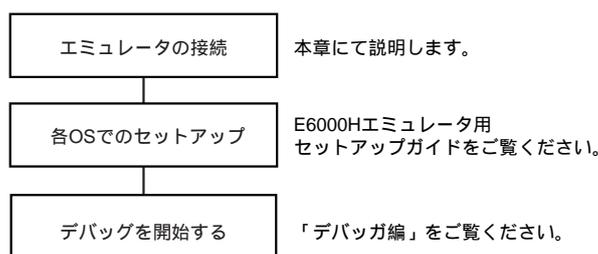


図 2.1 E6000H エミュレータ使用までのフローチャート

注意

準備にあたり、本章の内容を必ずよく読んで理解してください。誤った使い方は、本エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

2.2 エミュレータの接続

2.2.1 ユーザシステムとの接続

警告

エミュレータおよびユーザシステムの電源が入っているとき、全て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を破壊する可能性があります。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。

- (1) E6000H本体の電源がOFFであることを確認してください。E6000H本体前面右側にあるPOWERランプが消灯状態であれば電源はOFF状態です。
- (2) E6000H本体のAC電源ケーブルがAC電源用コンセントに接続されている場合は、AC電源用コンセントから外します。
- (3) E6000Hユーザシステムインタフェースボードの下面に実装しているコネクタにユーザシステム側のコネクタ1pinを合わせて接続します。コネクタ接続の際には、片側が浮いた状態にならないように必ずきちんと奥まで接続してください。

2. 使用前の準備

2.2.2 ユーザシステムインタフェースボードの接続

接続方法は、各 SH7058、SH7058S、SH7059 E6000H シリーズ用ユーザシステムインタフェースボード取扱い説明書を参照してください。

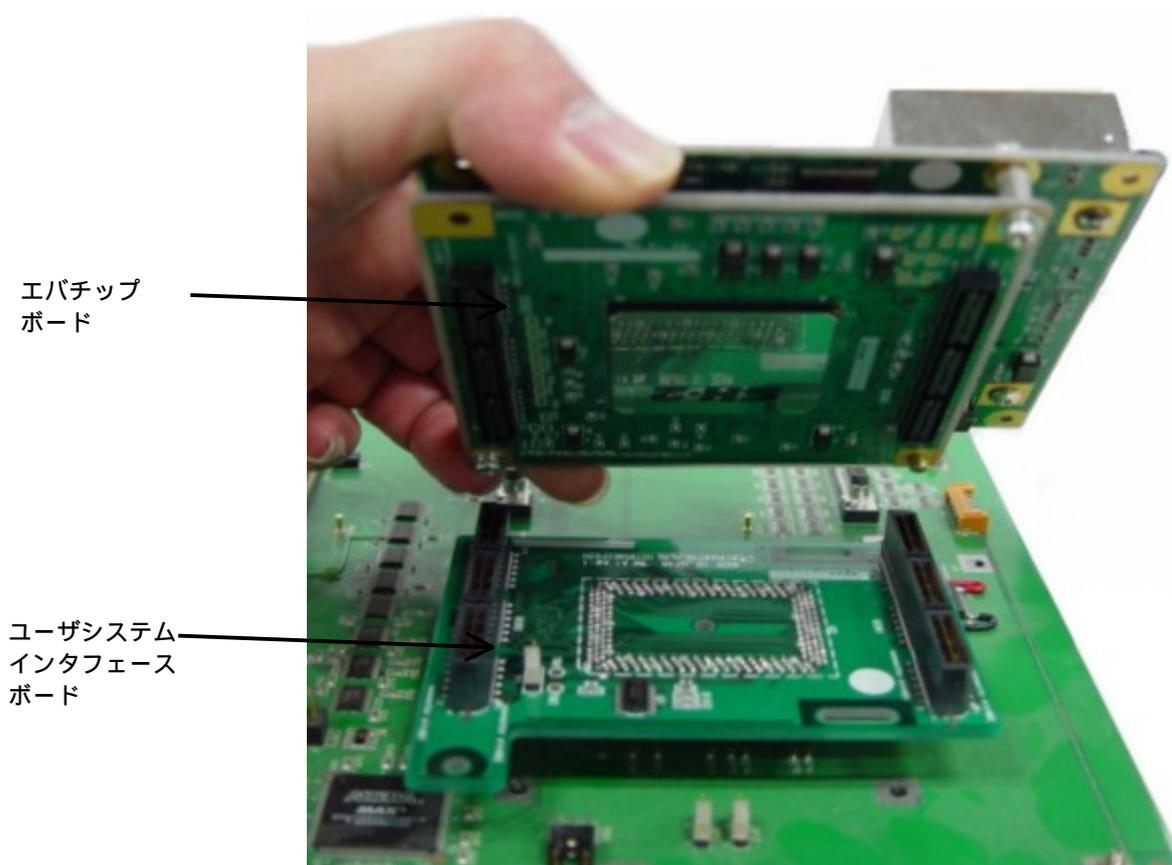


図2.2 ユーザシステムインタフェースボードの接続

警告

ユーザシステムインタフェースボードの接続、取り外しを行う場合は、必ずエミュレータおよびユーザシステムの電源すべてをオフにし、コネクタおよび IC ソケットのピン番号を確かめて作業してください。

電源がオンの状態で作業を行ったり、接続を誤ると、エミュレータ、ユーザシステムインタフェースボードまたは、ユーザシステムの破壊、発煙発火の可能性があります。

2.2.3 外部プローブ接続用コネクタ

注意

外部プローブには方向性があります。よく向きを確認してから挿入してください。誤った方向で挿入しますと、コネクタ部を破壊する原因となります。

E6000H エバチップボードの外部プローブ接続用コネクタの端子を通して、外部信号のトレース、マルチエミュレータのブレーク検出をすることができます。

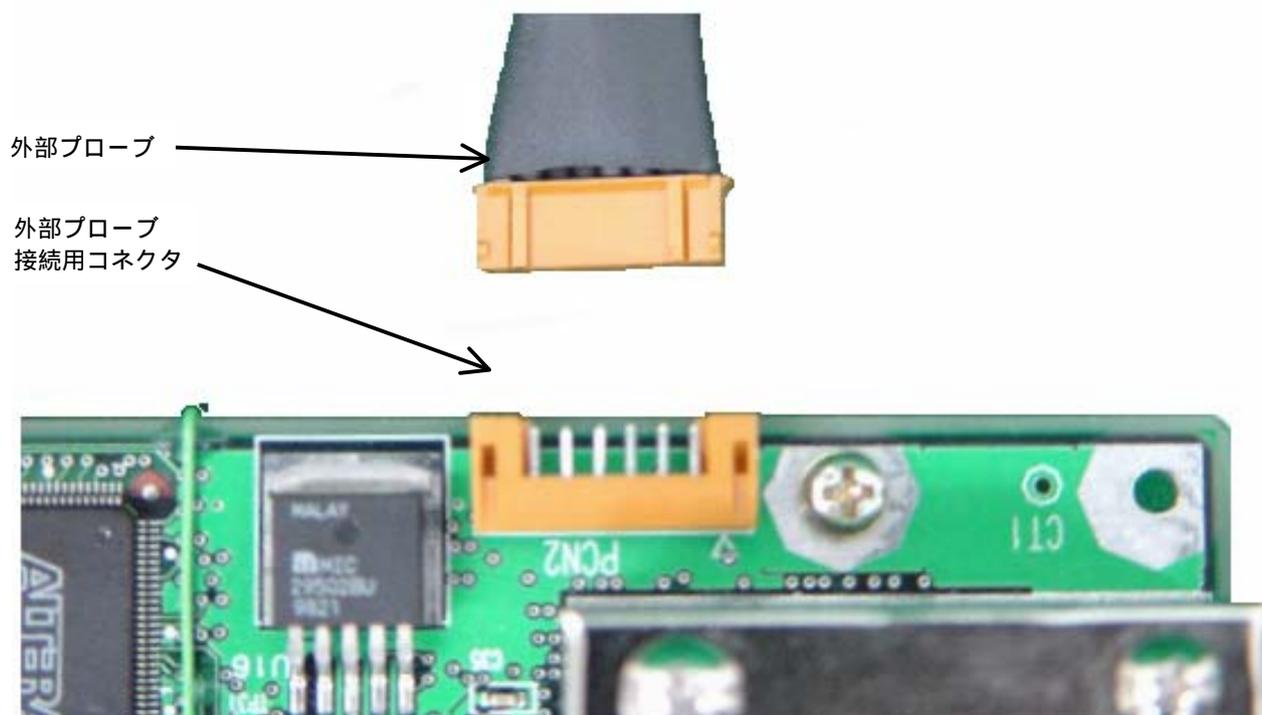


図2.3 外部プローブ接続用コネクタ

ピン番号	プローブ名	信号名	備 考
1	1	プローブ入力 1	同期ブレーク用の入力端子
2	2	プローブ入力 2	
3	3	プローブ入力 3	
4	4	プローブ入力 4	
5	5	GND	GND の接続端子
6	6	トリガ出力	トリガモード時の出力端子
7	7	GND	GND の接続端子
8	8	RUN / ブレーク状態出力	RUNSTEP 時"High"レベル出力

2. 使用前の準備

2.2.4 クロックの選択

エミュレータでは、ターゲットマイコンの入力クロックとして、エバチップボードに実装する水晶発振子、ユーザシステムからの外部クロック入力、およびエミュレータ内部の貸出しクロックの3種類をサポートしています。それぞれのクロックの設定は、[コンフィグレーションプロパティ]ダイアログボックスを使用して行います。また、本エミュレータではターゲットマイコンのシステムクロック(ϕ)として最高動作周波数 80.0MHz (外部クロック 10.0MHz を 8 通倍した値)まで使用することができます。選択できるクロックの選択は下記になります。

Target	外部クロック (ユーザシステムから EXTAL 端子にクロックを供給) 5.0 - 10.0MHz
Xtal	水晶発振子 5.0 - 10.0MHz
10MHz	エミュレータの貸出しクロック
5MHz	エミュレータの貸出しクロック

(1) 水晶発振子を使用する場合

本製品には水晶発振子は添付していませんので、お客様が使用する周波数の水晶発振子を用意してください。ただし、ターゲットマイコンの入力クロックとして水晶発振子を使用する場合の周波数範囲は 5.0 - 10.0MHz です。次に水晶発振子を実装する手順を示します。

注意

エミュレータの電源が入っているとき、水晶発振子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、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。

1. エミュレータの電源がOFFであることを確認してください (POWER LEDが消灯していることを確認してください)。
2. エバチップボードにある水晶発振子取り付け端子に水晶発振子を実装してください。(図2.4参照)
3. ユーザ電源をONし、エミュレータの電源をONすると、自動的に水晶発振子に設定されて立ち上がります。この機能により、ユーザシステムが接続されていない場合でも、ユーザシステムの動作周波数でユーザプログラムを実行することができます。

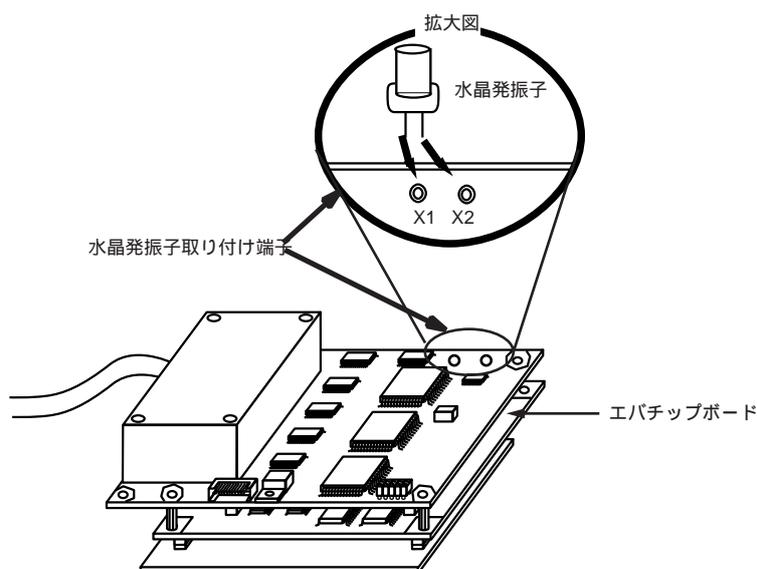


図2.4 水晶発振子の実装方法

(2) 外部クロックを使用する場合

1. エミュレータの電源がOFFであることを確認してください (POWER LEDが消灯していることを確認してください)。
2. ユーザシステムにエバチップボードを接続し、ユーザシステムからEXTAL端子にクロックを供給してください。
3. ユーザシステムの電源をONし、エミュレータの電源をONすると外部クロックに設定されて立ち上がります。

(3) エミュレータの貸出しクロックを使用する場合

[コンフィグレーションプロパティ]ダイアログボックスにてエミュレータの貸出しクロックを指定してください。

[参考] エミュレータのシステムプログラム起動時、次の順序で入力するクロックを自動的に選択します。

1. ユーザシステムから外部クロックが入力されている場合、ユーザシステムのクロックを選択
2. エバチップボードに水晶発振子が実装してある場合、水晶発振子を選択
3. エミュレータの貸出しクロックを選択

2.2.5 システムグランド系の接続

注意

システムグランドは必ずユーザシステム上で、フレームグランドとシグナルグランドを切り離してください。フレームグランドとシグナルグランドを接続した状態でエミュレータを接続すると、誤動作の恐れがあります。

エミュレータのシグナルグランドは、エバチップボードを介して、ユーザシステムのシグナルグランドに接続されます。

エミュレータ本体内部では、フレームグランドとシグナルグランドが接続されています。(図2.5)ユーザシステムでは、フレームグランドとシグナルグランドを接続せず、フレームグランドのみを接続してください。

ユーザシステム内で、フレームグランドとシグナルグランドを切り離すのが難しい場合、図2.6のようにエミュレータ本体の100V用電源のフレームグランドと同一のコンセントでグランドをとり、グランド電位が等しくなるようにしてください。

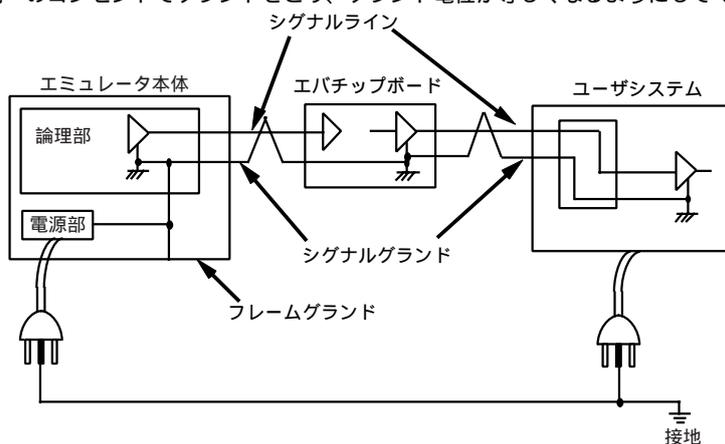


図2.5 システムグランド系の接続

2. 使用前の準備

警告

エミュレータおよびユーザシステムの電源が入っているとき、すべてのケーブル類の抜き差しを行わないでください。抜き差しを行った場合、エミュレータとユーザシステムを破壊する可能性があります。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。

ユーザシステムのシステムグラウンド系は、最適なグラウンド接続にしてノイズやグラウンドループなどの影響を最小にすることが必要です。したがって、エバチップボードとユーザシステムの接続において、エバチップボードの端子とユーザシステムのグラウンドが確実に接続されているか確認してください。

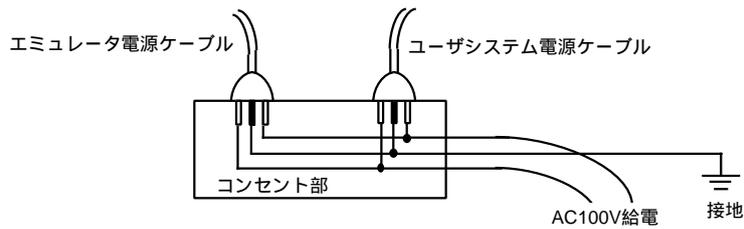


図2.6 フレームグラウンド接続例

2.2.6 PC インタフェースボードについて

PC インタフェースボード、LAN アダプタおよび USB アダプタについては、各製品付属の取扱い説明書を参照してください。

3. ハードウェア仕様

3.1 使用環境条件

注意

E6000H エミュレータを使用する場合、表 3.1 に示す条件を守ってください。この条件を満たさない状態でエミュレータを動作させた場合、本エミュレータ、ユーザプログラムおよびユーザシステムは正常に動作しないおそれがあります。

表3.1 使用環境条件

項番	項目	仕 様
1	温度	動作時 : 10 ~ 35 非動作時 : - 10 ~ 50
2	湿度	動作時 : 35 ~ 80% 結露なし 非動作時 : 35 ~ 80% 結露なし
3	振動	動作時 : 最大 2.45m/s ² 非動作時 : 最大 4.9m/s ² 梱包輸送時 : 最大 14.7m/s ²
4	AC 入力電源	電圧 : AC100V ± 10% 周波数 : 50/60Hz 消費電力 : 75W
5	周囲ガス	腐食性ガスのないこと

3.2 エミュレータの外形寸法と質量

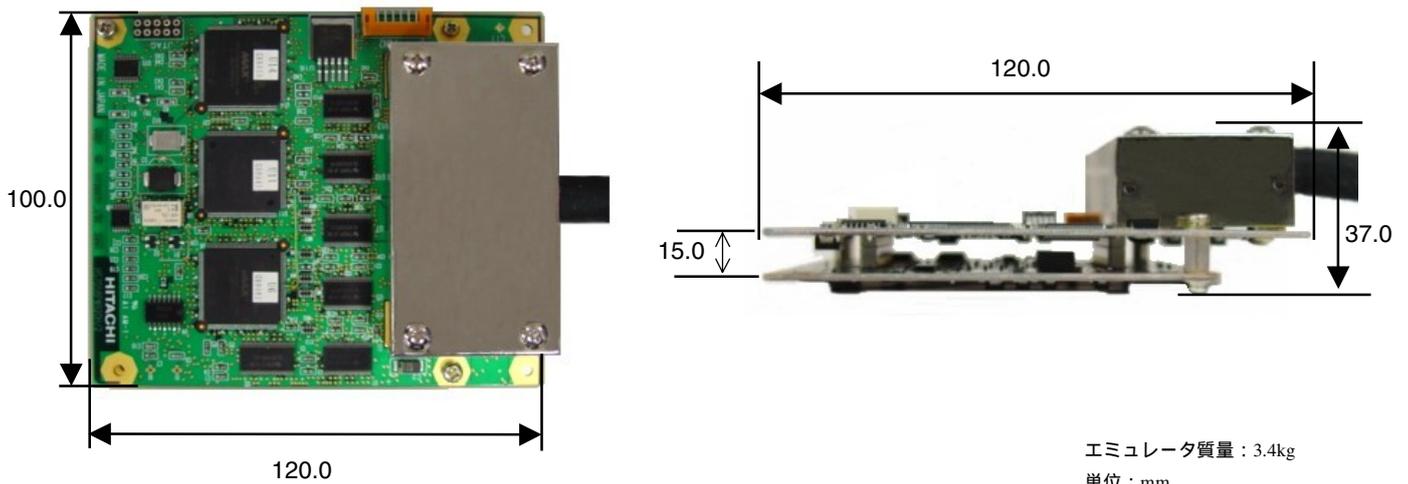
図3.1 に E6000H エミュレータの外形寸法と質量を示します。

(画像は HS7058EPH62H のものですが、寸法は HS7058EPH62H、HS7059EPH62H 共通です)

- E6000Hエミュレータ本体部の外形寸法図



- エバチップボードの外形寸法図



エミュレータ質量：3.4kg

単位：mm

公差：±1.0mm

図 3.1 E6000H エミュレータの外形寸法図

3.3 ユーザシステムインタフェース回路

3.3.1 ユーザシステムインタフェース回路

エミュレータに実装されている MCU とユーザシステム間は、バッファおよび抵抗等が入っている信号があります。したがって、エミュレータとユーザシステムを接続する場合、信号の遅れおよび FANIN、FANOUT に注意してください。

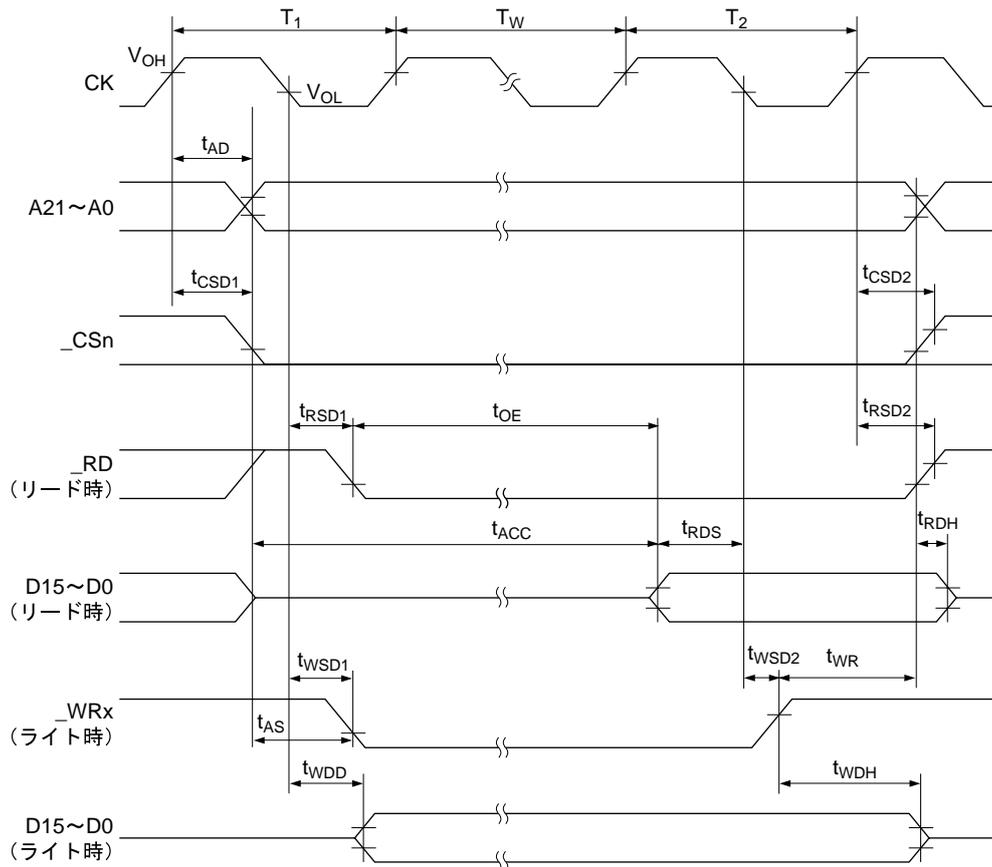
本エミュレータを使用した場合の AC タイミング値を表 3.2 に示します。

【注】 表 3.2 のエミュレータ使用時の値は実測した値で、参考値です。保証値ではありません。

表 3.2 エミュレータ使用時のバスタイミング (外部バスクロック : 20.0MHz)

項目	MCU 仕様 (ns)		エミュレータ使用時 (ns)	
	MIN	MAX	MIN	MAX
tRDS	15		18	
tACC	$tcyc \times (n+1.5) - 39$ (n はウェイト数)		$tcyc \times (n+1.5) - 45$ (n はウェイト数)	

図3.2 に基本バサイクル (ソフトウェアウェイト)、図3.3 ~ 図3.8 にユーザシステムと接続するエミュレータのインタフェース回路を示します。



【注】 t_{RDH} : A21~A0、_CSn、_RDの最も早いネゲートタイミングから規定

図 3.2 基本バサイクル (ソフトウェアウェイト)

3. ハードウェア仕様

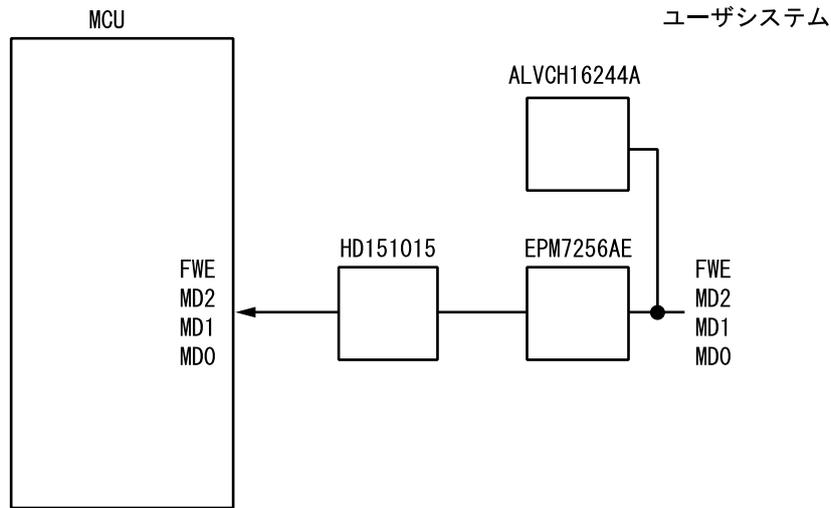
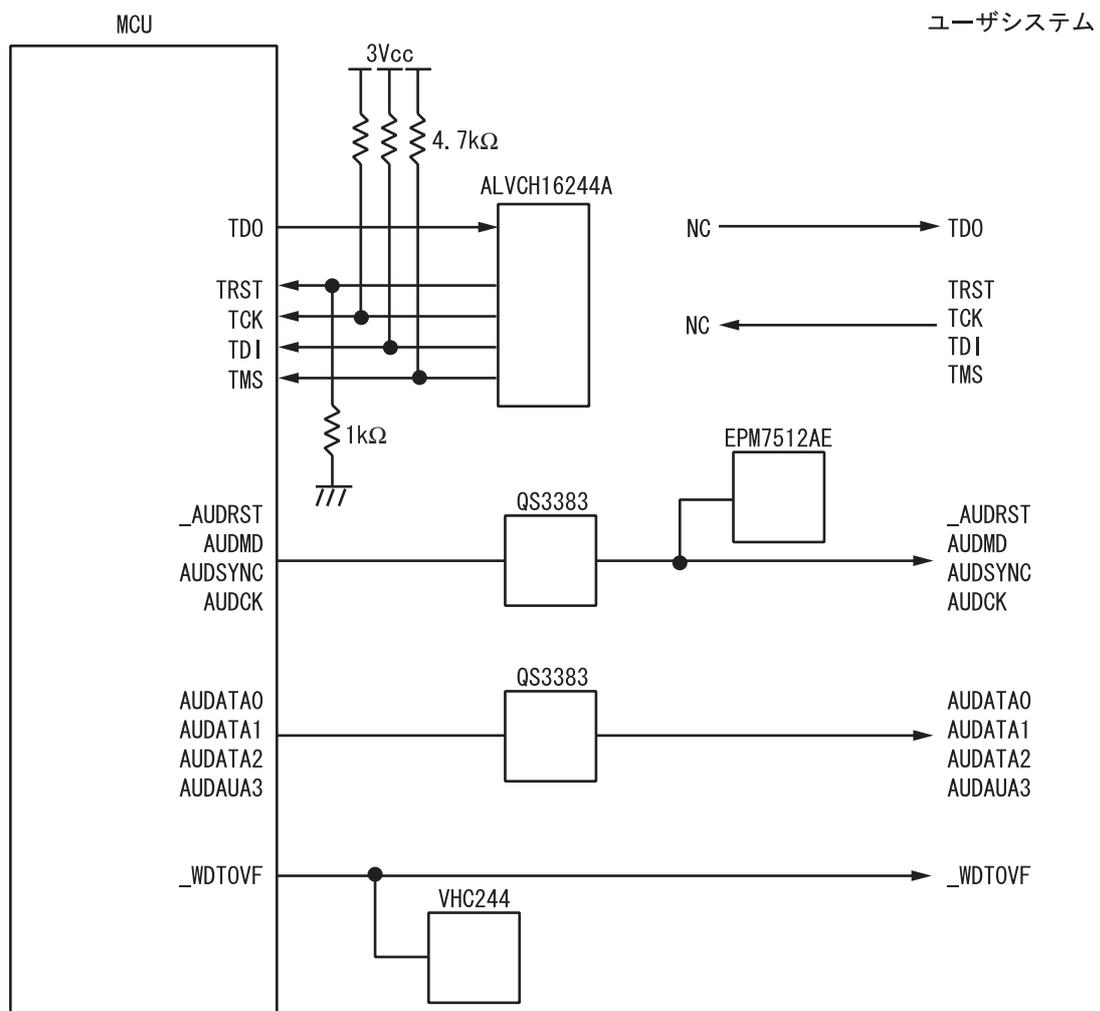


図 3.3 ユーザインタフェース回路 (1)



3VccはE6000Hから供給される3.3V電源です。

図 3.4 ユーザインタフェース回路 (2)

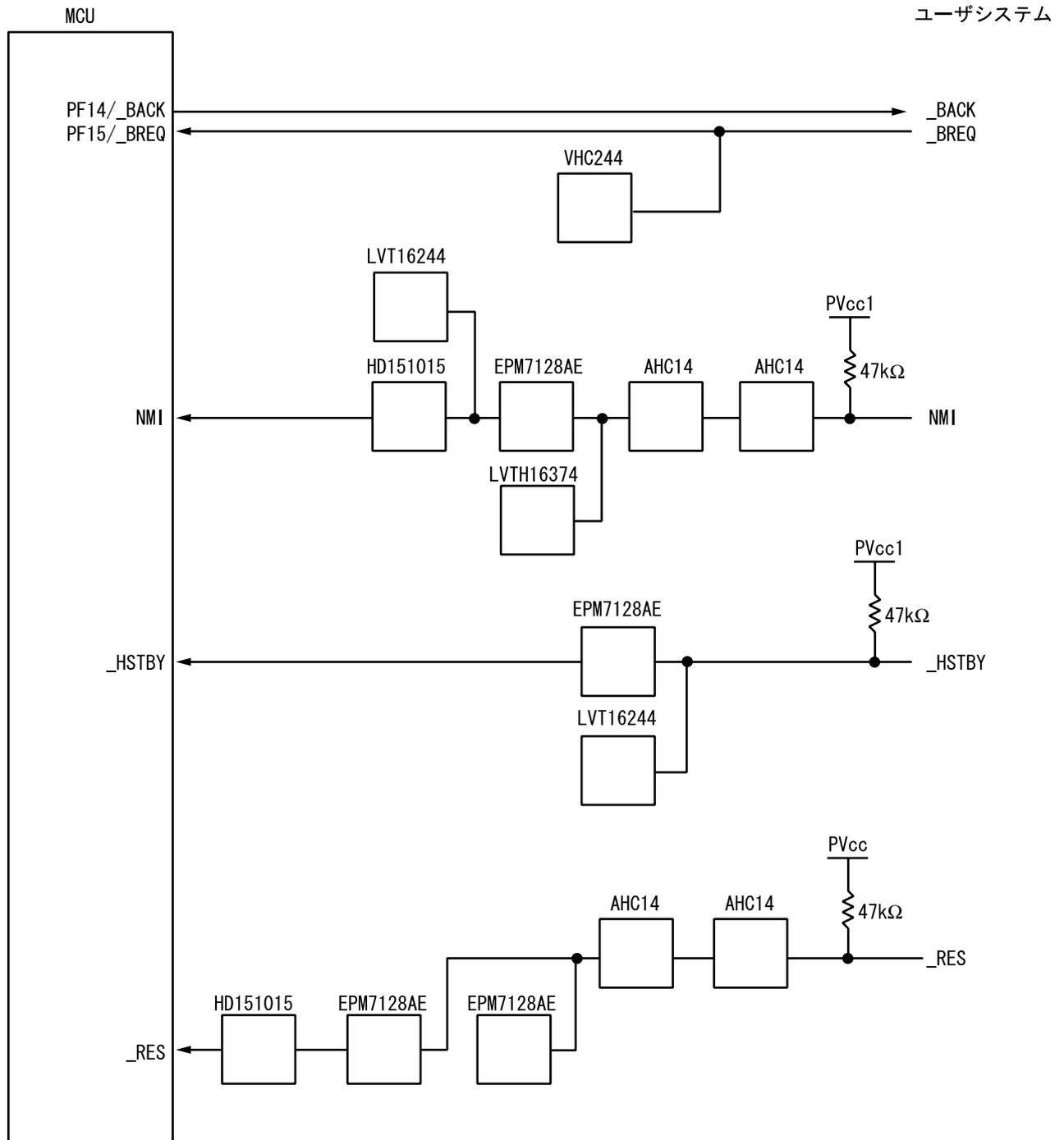
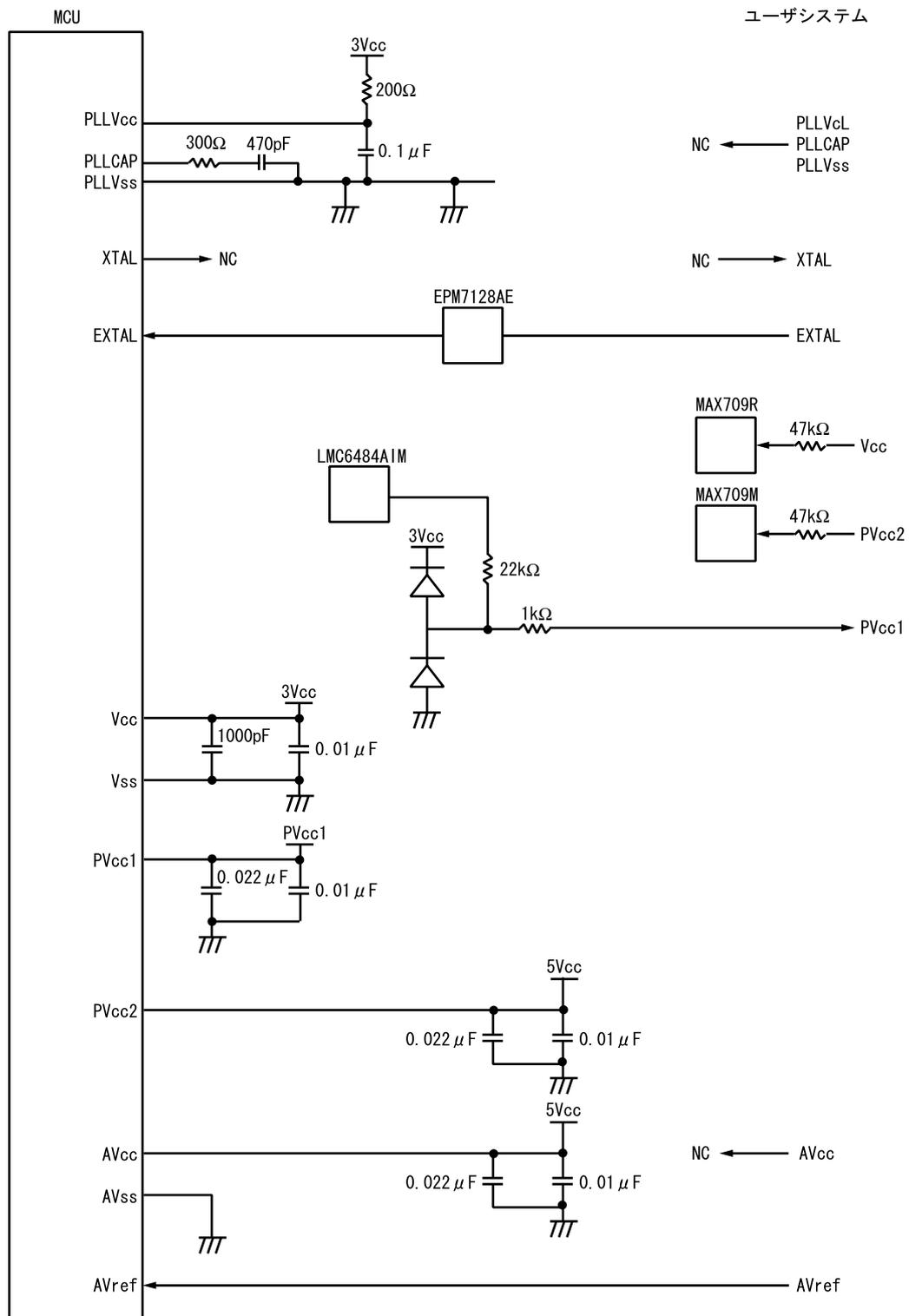


図 3.5 ユーザインタフェース回路 (3)

3. ハードウェア仕様



3VccはE6000Hから供給される3.3V電源です。
5VccはE6000Hから供給される5V電源です。

図 3.6 ユーザインタフェース回路 (4)

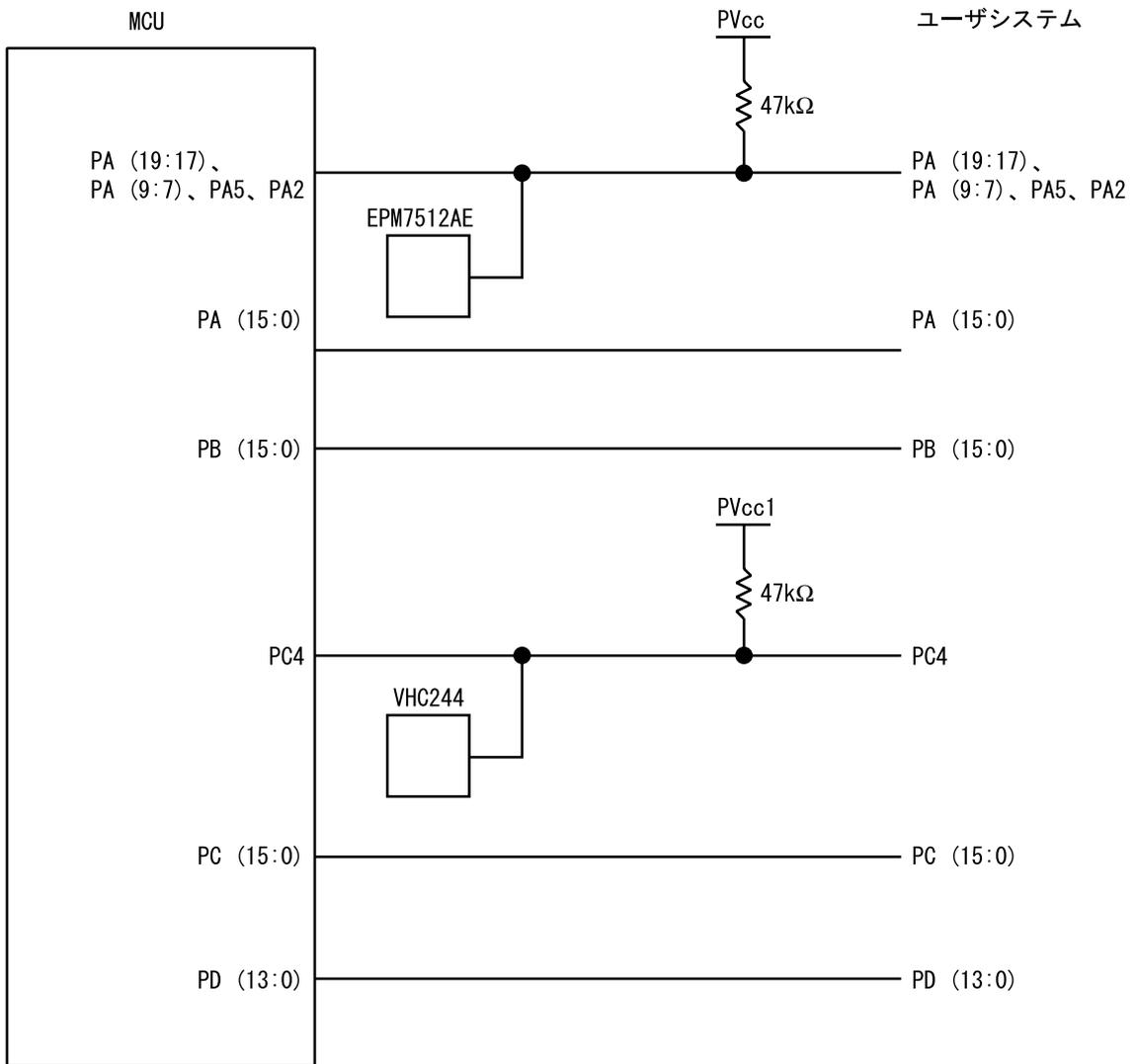


図 3.7 ユーザインタフェース回路 (5)

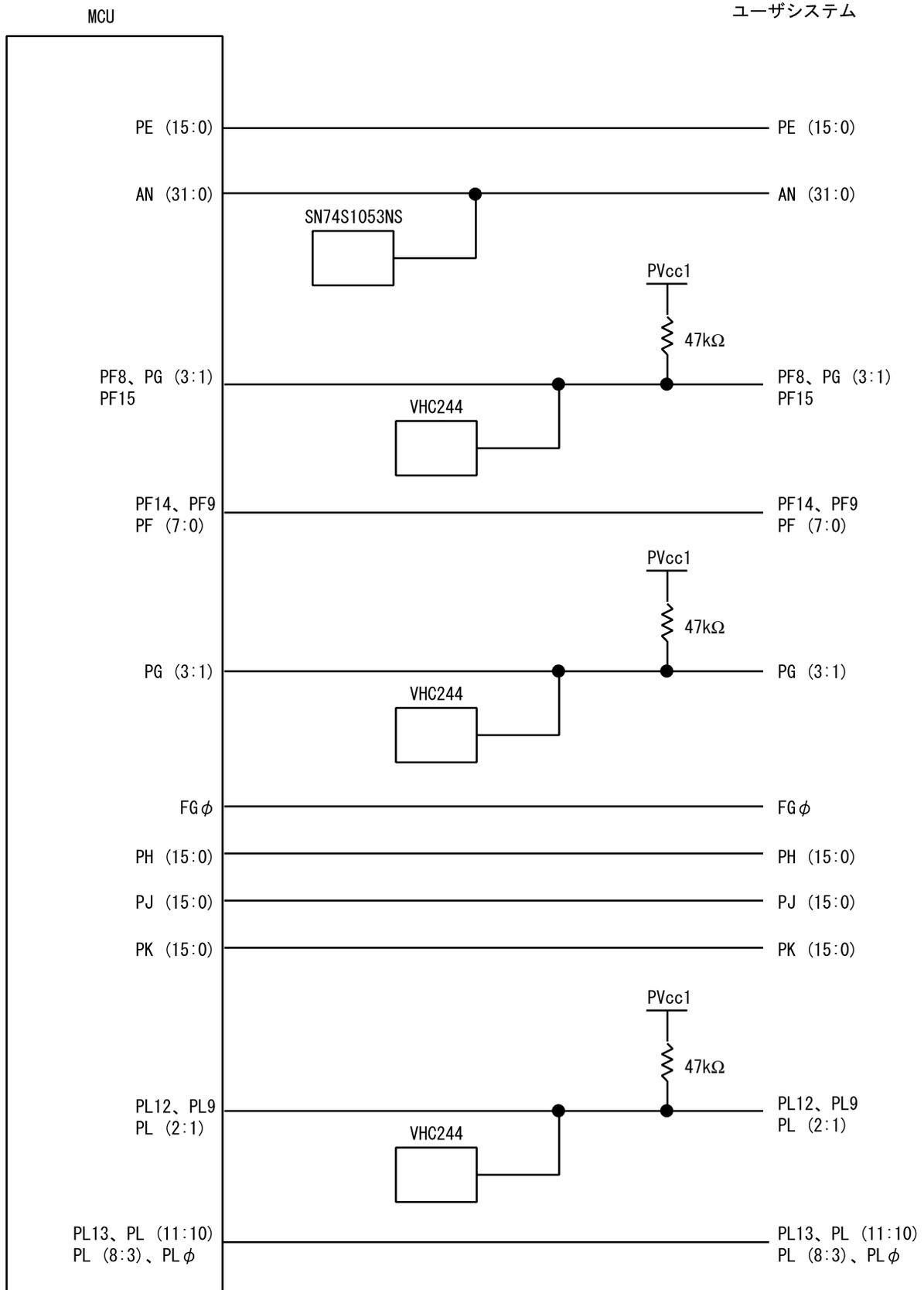


図 3.8 ユーザインタフェース回路 (6)

3.3.2 ユーザシステムインタフェースのディレイ時間

_RES 信号はエバチップボード上の論理を介してユーザシステムに接続されますので、ユーザシステムから MCU に信号が入力されるまで、表 3.3 に示すディレイ時間が生じます。

表 3.3 エバチップボード経由信号のディレイ時間

No	信号名	ディレイ時間 (ns)
1	_RES	15.0

3.4 エミュレータとユーザシステムの接続

3.4.1 ユーザシステムとの接続方法



警告

エミュレータおよびユーザシステムのパワーオン時、すべてのケーブル類、およびソケット類の抜き差しを行ないてください。

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。

エミュレータとユーザシステムとの接続方法として、ユーザシステムインタフェースボードを使用します。

表 3.4 使用するユーザシステムインタフェースボードとユーザインタフェース

ユーザシステムインタフェースボードのタイプ	ユーザインタフェース
HS7058ECF62H	PRQP0256KB-A (TQPACK256RD)
HS7058ECB62H	PRBG0272FA-A (CSPACK256Z2021H01)

【注】 TQPACK,CSPACK シリーズは東京エレテック株式会社製

(1) ICソケットの取り付け

1. ICソケットの実装

各パッケージ用ICソケットをユーザシステムに実装します。1ピンの位置を確認した上、はんだ付け前にICソケットの底面をエポキシ樹脂系の接着剤でユーザシステムに固定してください。

2. ICソケットのはんだ付け

ICソケットの固定の後、はんだ付けします。

はんだ付けされるリードの端面には必ずフィレットが生成されるようにしてください(通常より多めにはんだの量を調整してください)。

3. ハードウェア仕様

(2) HS7058ECF62H を使用した接続



エミュレータおよびユーザシステムのパワーオン時、すべてのケーブル類、およびソケット類の抜き差しを行わないでください。

抜き差しを行った場合、エミュレータとユーザシステムの発煙発火の可能性があります。また、デバッグ中のユーザプログラムの破壊の可能性があります。

- 【注】
1. HS7058ECF62Hの詳しい取扱いについては製品に付属の取扱説明書を参照してください。
 2. 指定した QFP ソケット (TQPACK256RD) 以外の IC ソケットを使用した場合、本ユーザシステムインタフェースボードは使用できません。

エミュレータを接続するユーザシステム上に PRQP0256KB-A 対応 IC ソケット (TQPACK256RD: 東京エレクトック (株) 製) を用意してください。ピン配置は SH7058、SH7058S、SH7059 実チップと全く同じになっていますので、ハードウェアマニュアルのピン配置を参照してください。図 3.9 に HS7058ECF62H を使用した接続図、図 3.10 に HS7058ECF62H を使用した際の部品配置制限、図 3.11 にユーザシステム側 IC ソケット推奨マウントパッド寸法を示します。

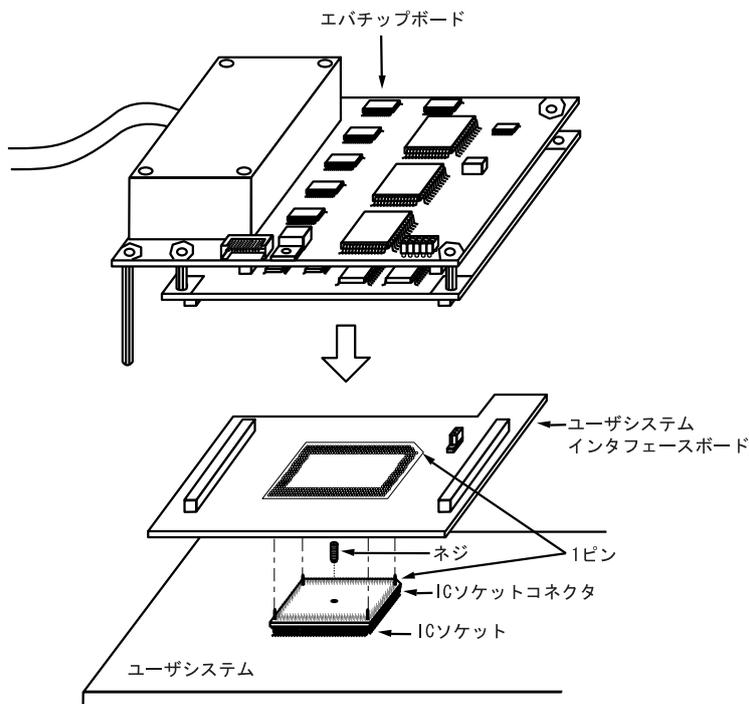


図 3.9 HS7058ECF62H を使用した接続

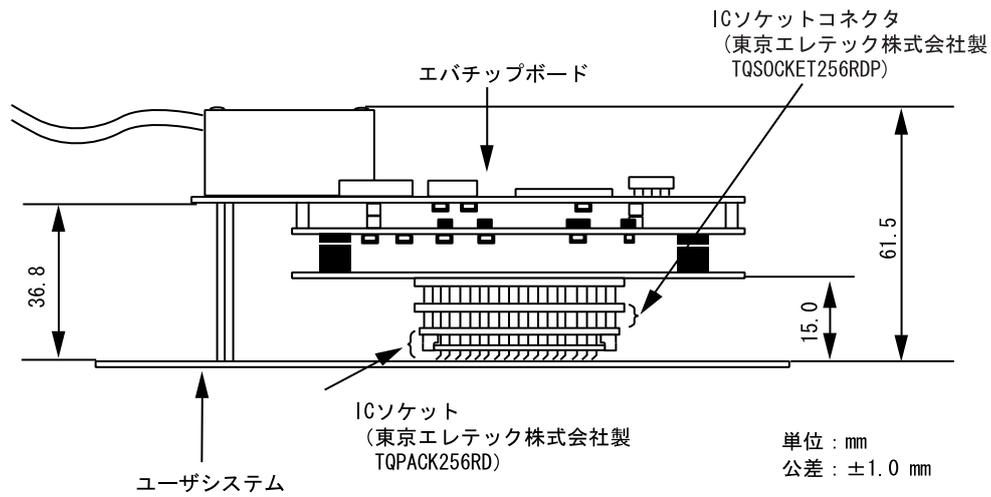


図 3.10 部品配置制限

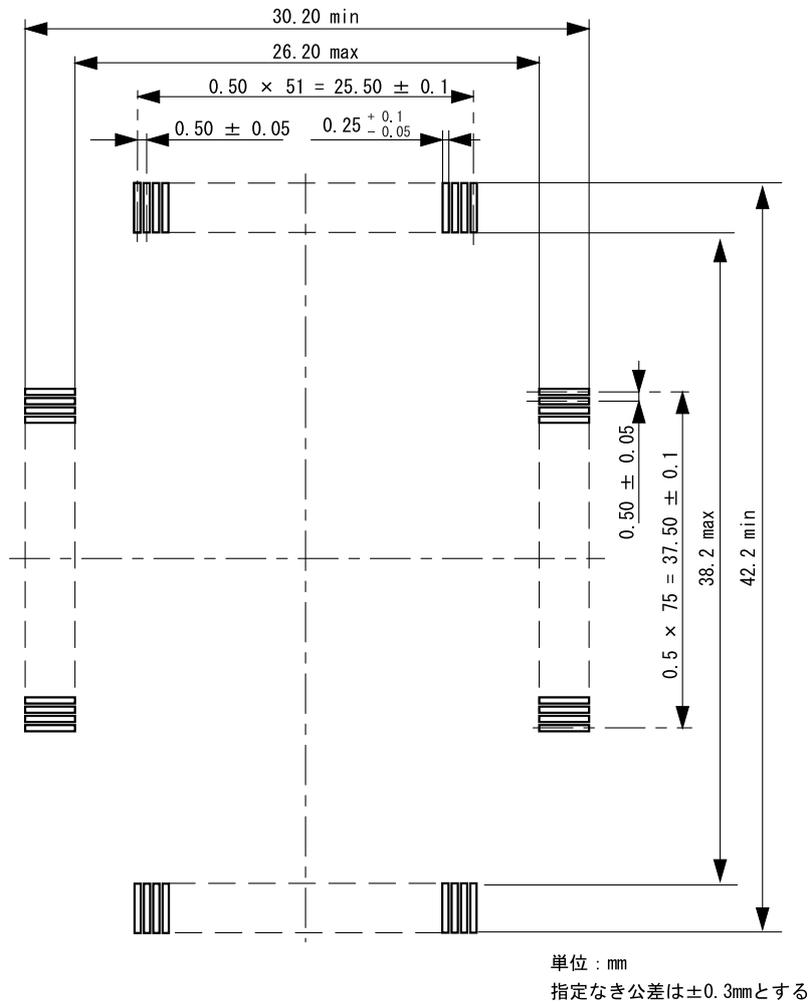


図 3.11 ユーザシステム側 IC ソケット推奨マウントパッド寸法

3. ハードウェア仕様

(3) HS7058ECB62H を使用した接続



警告

エミュレータおよびユーザシステムのパワーオン時、すべてのケーブル類、およびソケット類の抜き差しを行わないでください。

抜き差しを行った場合、エミュレータとユーザシステムの発煙発火の可能性があります。また、デバッグ中のユーザプログラムの破壊の可能性があります。

- 【注】
1. HS7058ECB62Hの詳しい取扱いについては製品に付属の取扱説明書を参照してください。
 2. 指定した BGA ソケット (CSPACK256Z2021H01) 以外の IC ソケットを使用した場合、本ユーザシステムインタフェースボードは使用できません。

エミュレータを接続するユーザシステム上に PRBG0272FA-A 対応 IC ソケット (CSPACK256Z2021H01 : 東京エレクトック (株) 製) を用意してください。ピン配置は SH7058、SH7058S、SH7059 実チップと全く同じになっていますので、ハードウェアマニュアルのピン配置を参照してください。図 3.12 に HS7058ECB62H を使用した接続図、図 3.13 に HS7058ECB62H を使用した際の部品配置制限、図 3.14 にユーザシステム側 IC ソケット推奨マウントパッド寸法を示します。

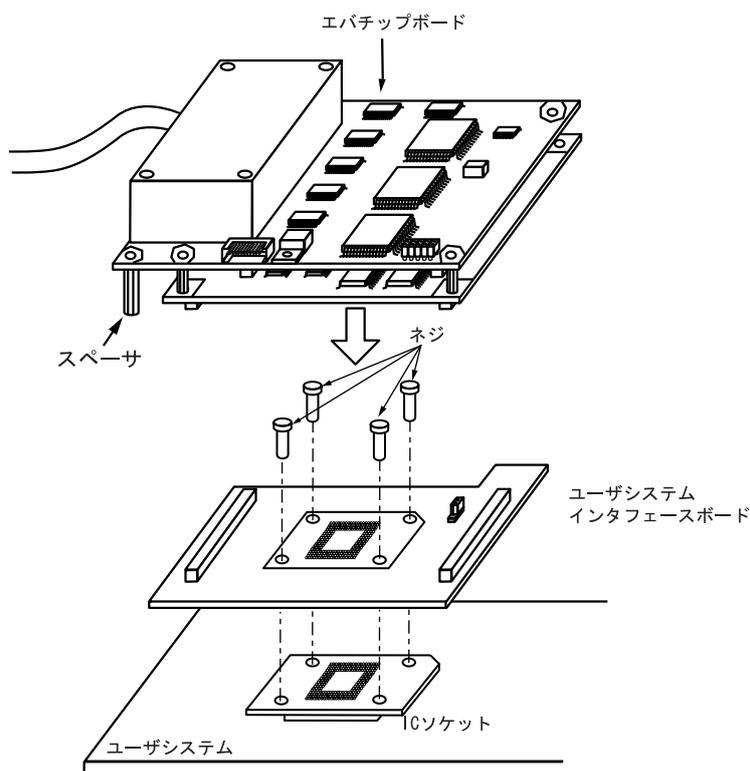


図 3.12 HS7058ECB62H を使用した接続

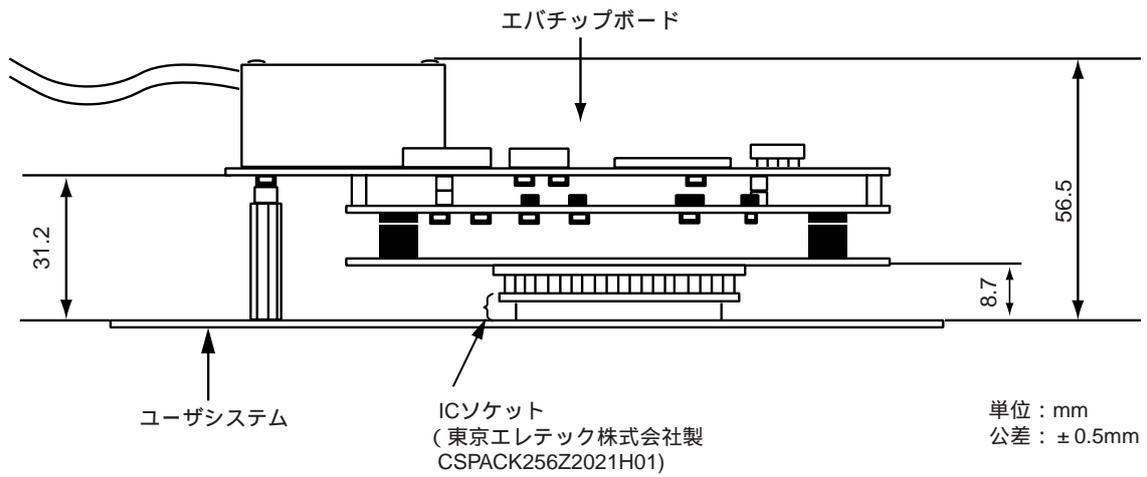


図 3.13 部品配置制限

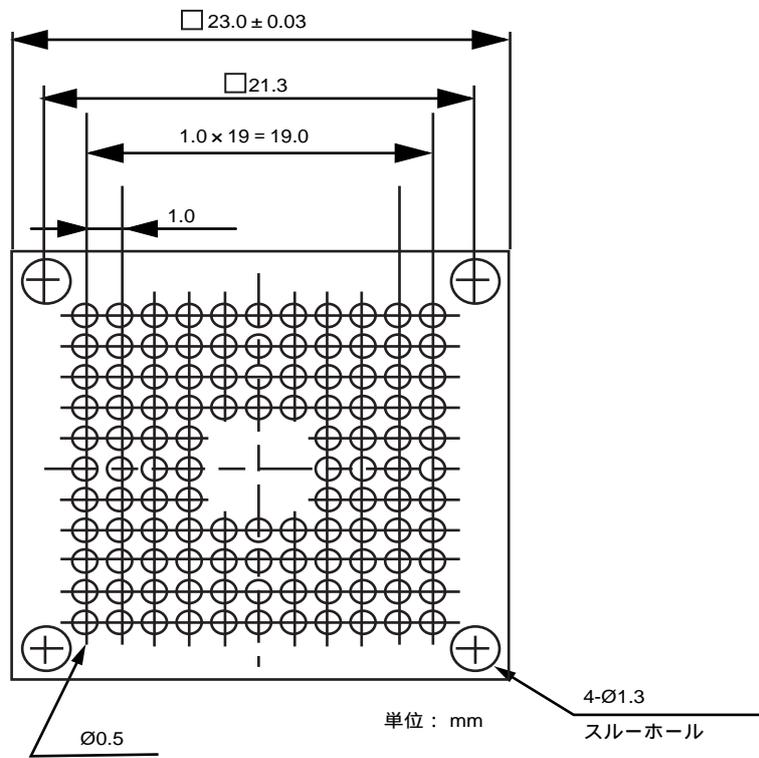


図 3.14 ユーザシステム側 IC ソケット推奨マウントパッド寸法

3. ハードウェア仕様

(4) 専用コネクタを使用した接続



エミュレータおよびユーザシステムのパワーオン時、すべてのケーブル類、およびソケット類の抜き差しを行わないでください。

抜き差しを行った場合、エミュレータとユーザシステムの発煙発火の可能性があります。また、デバッグ中のユーザプログラムの破壊の可能性があります。

【注】 指定した専用コネクタ (QTH-090-0x-L-D-A : SAMTEC 製) × 2 以外のコネクタを使用した場合、本エバチップボードは使用できません。

エミュレータを接続するユーザシステム上に専用コネクタ (QTH-090-0x-L-D-A : SAMTEC 製) を用意してください。

図 3.15 に専用コネクタを使用した接続図、図 3.16 に専用コネクタを使用した際の部品配置制限、図 3.17 にユーザシステム側コネクタ実装位置を示します。

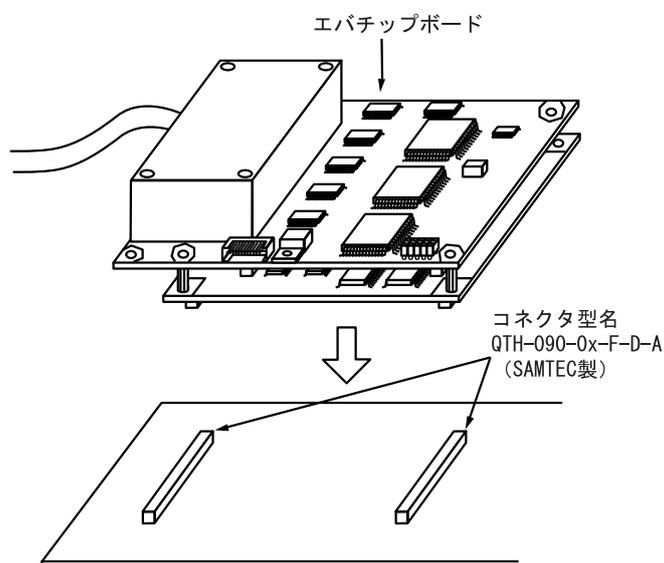
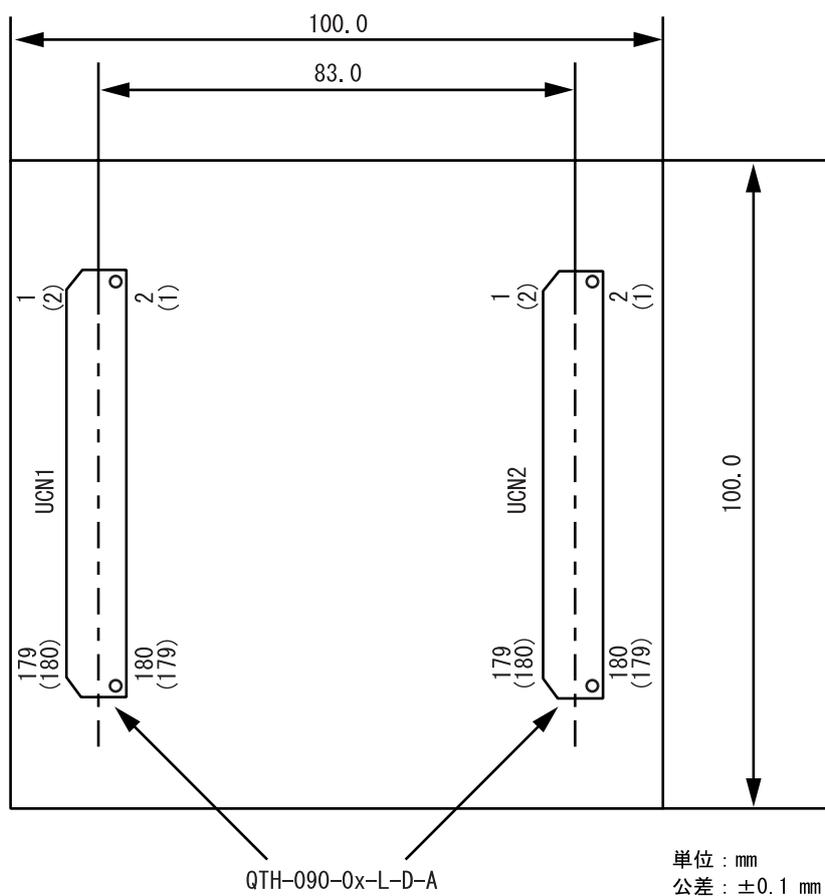


図 3.15 専用コネクタを使用した接続



※ ()内の数値はSamtec社表記のピンNo.

図 3.16 ユーザシステム側コネクタの実装位置

フットパターン寸法については、QTH-090-0x-L-D-Aのカタログを参照して設計してください。

注意

基板表記のピン No.は、Samtec 社指定のピン No.と異なります。

図 3.16 の括弧なしの値が基板表記のピン No.で括弧ありの表記が Samtec 社の指定のピン No.になります。

ユーザシステム側のコネクタ実装及び信号配線の誤りに注意してください。

3. ハードウェア仕様

3.4.2 ユーザシステムインタフェースコネクタのピン配置

HS7058EPH62H / HS7059EPH62H のユーザシステムインタフェースコネクタのピン配置を表 3.5 に示します。

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置

User I/F 1	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN1	1	(2)	GND
	2	(1)	GND
	3	(4)	GND
	4	(3)	GND
	5	(6)	GND(TGBON1)
	6	(5)	GND
	7	(8)	PE15/A15
	8	(7)	PF15/_BREQ
	9	(10)	PE14/A14
	10	(9)	PF14/_BACK
	11	(12)	PE13/A13
	12	(11)	PF13/_CS3
	13	(14)	PE12/A12
	14	(13)	PF12/_CS2
	15	(16)	GND
	16	(15)	GND
	17	(18)	PE11/A11
	18	(17)	PF11/_CS1
	19	(20)	PE10/A10
	20	(19)	PF10/_CS0
	21	(22)	PE9/A9
	22	(21)	PF9/_RD
	23	(24)	PE8/A8
	24	(23)	PF8/_WAIT
	25	(26)	GND
	26	(25)	GND
	27	(28)	PE7/A7
	28	(27)	PF7/_WRH
	29	(30)	PE6/A6
	30	(29)	PF6/_WRL
	31	(32)	PE5/A5
	32	(31)	PF5/A21/_POD
	33	(34)	PE4/A4
	34	(33)	PF4/A20

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 1	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN1	35	(36)	GND
	36	(35)	GND
	37	(38)	PE3/A3
	38	(37)	PF3/A19
	39	(40)	PE2/A2
	40	(39)	PF2/A18
	41	(42)	PE1/A1
	42	(41)	PF1/A17
	43	(44)	PE0/A0
	44	(43)	PF0/A16
	45	(46)	GND
	46	(45)	PD13/PULS6/HRxD0/HTxD1
	47	(48)	GND
	48	(47)	PD12/PULS4
	49	(50)	GND
	50	(49)	GND
	51	(52)	GND
	52	(51)	PD11/PULS3
	53	(54)	GND
	54	(53)	PD10/PULS2
	55	(56)	PD9/PULS1
	56	(55)	GND
	57	(58)	PD8/PULS0
	58	(57)	GND
	59	(60)	GND
	60	(59)	GND
	61	(62)	PD7/TIO1H
	62	(61)	GND
	63	(64)	PD6/TIO1G
	64	(63)	GND
	65	(66)	PD5/TIO1F
	66	(65)	PD4/TIO1E
67	(68)	GND	
68	(67)	GND	
69	(70)	GND	
70	(69)	PD3/TIO1D	

3. ハードウェア仕様

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 1	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN1	71	(72)	GND
	72	(71)	PD2/TIO1C
	73	(74)	GND
	74	(73)	PD1/TIO1B
	75	(76)	GND
	76	(75)	PD0/TIO1A
	77	(78)	AUDCK
	78	(77)	AUDMD
	79	(80)	PL13/_IRQOUT
	80	(79)	_AUDSYNC
	81	(82)	PL12/_IRQ4
	82	(81)	_AUDRST
	83	(84)	GND
	84	(83)	GND
	85	(86)	PL11/HRxD/HRxD1/HRxD0&HRxD1
	86	(85)	AUDATA3
	87	(88)	PL10/HTxD0/HTxD1/HTxD0&HTxD1
	88	(87)	AUDATA2
	89	(90)	PL9/SCK4/_IRQ5
	90	(89)	AUDATA1
	91	(92)	PL8/SCK3
	92	(91)	AUDATA0
	93	(94)	GND
	94	(93)	PL7/SCK2
	95	(96)	3Vcc
	96	(95)	PL6/ADEND
	97	(98)	3Vcc
	98	(97)	PL5/_ADTRG1
	99	(100)	3Vcc
	100	(99)	PL4/_ADTRG0
101	(102)	3Vcc	
102	(101)	GND	
103	(104)	3Vcc	
104	(103)	PL3/TCLKB	
105	(106)	3Vcc	
106	(105)	PL2/TIO11B/_IRQ7	

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 1	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN1	107	(108)	5Vcc
	108	(107)	PL1/TIO11A/_IRQ6
	109	(110)	5Vcc
	110	(109)	PL0/TI10
	111	(112)	5Vcc
	112	(111)	GND
	113	(114)	5Vcc
	114	(113)	PK15/TO8P
	115	(116)	5Vcc
	116	(115)	PK14/TO8O
	117	(118)	5Vcc
	118	(117)	PK13/TO8N
	119	(120)	GND
	120	(119)	PK12/TO8M
	121	(122)	GND
	122	(121)	GND
	123	(124)	PK11/TO8L
	124	(123)	GND
	125	(126)	PK10/TO8K
	126	(125)	GND
	127	(128)	PK9/TO8J
	128	(127)	GND
	129	(130)	PK8/TO8I
	130	(129)	GND
	131	(132)	PK7/TO8H
	132	(131)	PK6/TO8G
	133	(134)	PK5/TO8F
	134	(133)	PK4/TO8E
	135	(136)	PK3/TO8D
	136	(135)	PK2/TO8C
137	(138)	PK1/TO8B	
138	(137)	GND	
139	(140)	PK0/TO8A	
140	(139)	GND	
141	(142)	GND	
142	(141)	GND	
143	(144)	PJ15/TI9F	

3. ハードウェア仕様

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 1	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN1	144	(143)	GND
	145	(146)	PJ14/TI9E
	146	(145)	GND
	147	(148)	PJ13/TI9D
	148	(147)	PJ12/TI9C
	149	(150)	GND
	150	(149)	GND
	151	(152)	GND
	152	(151)	PJ11/TI9B
	153	(154)	GND
	154	(153)	PJ10/TI9A
	155	(156)	GND
	156	(155)	PJ9/TIO5D
	157	(158)	GND
	158	(157)	PJ8TIO5C
	159	(160)	GND
	160	(159)	GND
	161	(162)	PJ7/TIO2H
	162	(161)	GND
	163	(164)	PJ6/TIO2G
	164	(163)	GND
	165	(166)	PJ5/TIO2F
	166	(165)	GND
	167	(168)	PJ4/TIO2E
	168	(167)	PG3/_IRQ3/_ADTRG0
	169	(170)	GND
	170	(169)	PG2/_IRQ2/ADEND
	171	(172)	PJ3/TIO2D
	172	(171)	PG1/_IRQ1
	173	(174)	PJ2/TIO2C
174	(173)	PG0/PULS7/HRxD0/HRxD1	
175	(176)	PJ1/TIO2B	
176	(175)	PJ0/TIO2A	
177	(178)	GND	
178	(177)	GND	
179	(180)	GND	
180	(179)	GND	

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 2	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN2	1	(2)	GND
	2	(1)	GND
	3	(4)	GND
	4	(3)	GND
	5	(6)	GND
	6	(5)	GND
	7	(8)	GND
	8	(7)	GND
	9	(10)	PH0/D0
	10	(9)	FWE
	11	(12)	PH1/D1
	12	(11)	MD2
	13	(14)	PH2/D2
	14	(13)	MD1
	15	(16)	PH3/D3
	16	(15)	MD0
	17	(18)	GND
	18	(17)	PH4/D4
	19	(20)	EXTAL
	20	(19)	PH5/D5
	21	(22)	GND
	22	(21)	PH6/D6
	23	(24)	_RES
	24	(23)	PH7/D7
	25	(26)	GND
	26	(25)	GND
	27	(28)	_HSTBY
	28	(27)	PH8/D8
	29	(30)	GND
	30	(29)	PH9/D9
	31	(32)	CK
	32	(31)	PH10/D10
	33	(34)	GND
	34	(33)	PH11/D11
	35	(36)	GND
	36	(35)	GND
	37	(38)	PH12/D12

3. ハードウェア仕様

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 2	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN2	38	(37)	GND
	39	(40)	PH13/D13
	40	(39)	GND
	41	(42)	PH14/D14
	42	(41)	GND
	43	(44)	PH15/D15
	44	(43)	GND
	45	(46)	Vcc
	46	(45)	GND
	47	(48)	Vcc
	48	(47)	GND
	49	(50)	Vcc
	50	(49)	GND
	51	(52)	PVcc1
	52	(51)	GND
	53	(54)	PVcc1
	54	(53)	GND
	55	(56)	PVcc1
	56	(55)	PVcc2
	57	(58)	GND
	58	(57)	PVcc2
	59	(60)	AN0
	60	(59)	PVcc2
	61	(62)	AN1
	62	(61)	GND
	63	(64)	AN2
	64	(63)	GND
	65	(66)	AN3
	66	(65)	GND
	67	(68)	GND
68	(67)	GND	
69	(70)	AN4	
70	(69)	AN5	
71	(72)	GND	
72	(71)	AN6	
73	(74)	AN7	
74	(73)	NMI	

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 2	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN2	75	(76)	GND
	76	(75)	AN8
	77	(78)	AN9
	78	(77)	AN10
	79	(80)	AN11
	80	(79)	GND
	81	(82)	AN12
	82	(81)	AN13
	83	(84)	AN14
	84	(83)	AN15
	85	(86)	GND
	86	(85)	AVcc
	87	(88)	AN16
	88	(87)	AVcc
	89	(90)	AN17
	90	(89)	AVcc
	91	(92)	AN18
	92	(91)	AVcc
	93	(94)	AN19
	94	(93)	AVref
	95	(96)	GND
	96	(95)	AVref
	97	(98)	AN20
	98	(97)	AVss
	99	(100)	AN21
	100	(99)	AVss
	101	(102)	AN22
	102	(101)	AVss
	103	(104)	AN23
	104	(103)	AVss
105	(106)	GND	
106	(105)	AN24	
107	(108)	GND	
108	(107)	AN25	
109	(110)	_WDTOVF	
110	(109)	AN26	
111	(112)	GND	

3. ハードウェア仕様

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 2	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN2	112	(111)	AN27
	113	(114)	GND
	114	(113)	GND
	115	(116)	GND
	116	(115)	AN28
	117	(118)	GND
	118	(117)	AN29
	119	(120)	AN30
	120	(119)	GND
	121	(122)	AN31
	122	(121)	GND
	123	(124)	GND
	124	(123)	GND
	125	(126)	PA0/TIOA
	126	(125)	GND
	127	(128)	PA1/TIOB
	128	(127)	GND
	129	(130)	PA2/TIO3C
	130	(129)	PA3/TIO3D
	131	(132)	GND
	132	(131)	GND
	133	(134)	GND
	134	(133)	PA4/TIO3A
	135	(136)	GND
	136	(135)	PA5/TIO3B
	137	(138)	PB0/TO6A
	138	(137)	PA6/TIO3C
	139	(140)	PB1/TO6B
140	(139)	PA7/TIO3D	
141	(142)	PB2/TO6C	
142	(141)	GND	
143	(144)	PB3/TO6D	
144	(143)	PA8/TIO4A	
145	(146)	GND	
146	(145)	PA9/TIO4B	
147	(148)	PB4/TO7A/TO8A	
	148	(147)	PA10/TIO4C

表 3.5 HS7058EPH62H / HS7059EPH62H ピン配置 (続き)

User I/F 2	基板表記 Pin No.	Samtec 社指定 Pin No.	Signal Name
UCN2	149	(150)	PB5/TO7B/TO8B
	150	(149)	PA11/TIO4D
	151	(152)	PB6/TO7C/TO8C
	152	(151)	PB7/TO7D/TO8D
	153	(154)	PA12/TIO5A
	154	(153)	GND
	155	(156)	PA13/TIO5B
	156	(155)	PB8/TxD3/TO8E
	157	(158)	PA14/TxD0
	158	(157)	PB9/RxD3/TO8F
	159	(160)	PA15/RxD0
	160	(159)	PB10/TxD4/HTxD0/TO8G
	161	(162)	GND
	162	(161)	PB11/RxD4/HRxD0/TO8H
	163	(164)	PC0/TxD1
	164	(163)	GND
	165	(166)	PC1/RxD1
	166	(165)	PB12/TCLKA/_UBCTRG
	167	(168)	PC2/TxD2
	168	(167)	PB13/SCK0
	169	(170)	PC3/RxD2
	170	(169)	PB14/SCK1/TCLKB/TI10
	171	(172)	PC4/_IRQ0
	172	(171)	PB15/PULS5/SCK2
	173	(174)	GND
	174	(173)	GND (TGBON2)
	175	(176)	GND
	176	(175)	GND
177	(178)	GND	
178	(177)	GND	
179	(180)	GND	
180	(179)	GND	

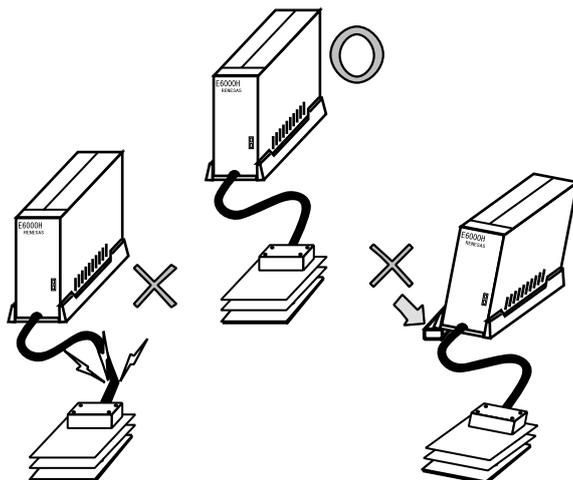
3. ハードウェア仕様

3.4.3 ユーザシステム接続時の注意事項

ユーザシステムとエバチップボードを接続する際には以下の点にご注意ください。

(1) エミュレータ本体の設置

エミュレータ本体とエバチップボードの位置関係により、トレースケーブルが大きく曲げられることがあります。このような状態で使用すると、ユーザインタフェース部に大きなストレスが加わり、接点、接触不良等の機械的破損をまねく原因となります。また、使用中にエミュレータ本体が動いてしまうと、ユーザインタフェース部に思わぬストレスを与えることとなります。エミュレータ本体の設置位置に十分ご注意ください。



(2) 電源の確認

ユーザシステムに接続する際には、エミュレータ、ユーザシステムとも、電源が投入されていないことを確認してから行なってください。

(3) Vcc、PVcc1、PVcc2 の接続

エミュレータはユーザシステムの電源 ON/OFF を以下に示す Vcc 端子にて監視し、検出しています。

(a) HS7058EPH62H / HS7059EPH62H 専用コネクタ接続の場合

Vcc : UCN2-45 ピン、UCN2-47 ピン、UCN2-49 ピン

PVcc1 : UCN2-51 ピン、UCN2-53 ピン、UCN2-55 ピン

PVcc2 : UCN2-56 ピン、UCN2-58 ピン、UCN2-60 ピン

(b) ユーザシステムインタフェースボード接続の場合

各パッケージの Vcc 端子、及び PVcc1、PVcc2 端子

ピン配置は SH7050 シリーズ(F-ZTAT マイコン)のハードウェアマニュアルを参照してください。

ユーザシステムを接続した際にこの端子にユーザ電源が接続されないと、エミュレータからはユーザシステムが接続されていないと認識されます。

ユーザ実機を接続する際は、この端子にユーザシステムの電源が接続されていることを確認してください。

3.5 ターゲットマイコンのサポート

3.5.1 メモリ空間

SH7058、SH7058S、SH7059 は、アーキテクチャ上 4GB のメモリ空間を持っています。

(1) 内蔵フラッシュメモリ領域

1. 内蔵フラッシュメモリ領域へのアクセス。

E6000Hエミュレータは搭載されたMCUの内蔵フラッシュメモリを使用しています。内蔵フラッシュメモリが存在するモードでは、内蔵フラッシュメモリにアクセスします。内蔵フラッシュメモリ領域へのアクセスは、ユーザプログラムの実行によるアクセスとエミュレータのコマンドで次のようになります。

ユーザプログラム実行によるアクセスはリードのみ可能です。内蔵フラッシュメモリ領域へのライトを行ってもブレイクしません。

エミュレータ機能（メモリウィンドウ、ロードなど）によるアクセスは、常にリード/ライト可能です。

内蔵フラッシュメモリ領域へのアクセスは1ステートとなります。

(2) 内蔵 I/O 領域

内蔵 I/O 領域にアクセスするとエミュレータに搭載されている MCU 内の内蔵 I/O にアクセスします。ユーザプログラムを、内蔵 I/O への書き込みやアクセスしたときにブレイクしたい場合はハードウェアブレイクまたは内蔵ブレイクを使用してください。

(3) 外部メモリ領域

MCU の外部メモリ領域には、エミュレータがサポートしている全メモリ属性を割り付けることができます。

(4) エミュレーション RAM 領域

E6000H エミュレータは内蔵フラッシュメモリの RAM エミュレーション用にエミュレーション RAM を搭載しています。

HS7058EPH62H のエミュレーション RAM 領域 16kB × 16 ブロック

HS7059EPH62H のエミュレーション RAM 領域 16kB × 24 ブロック

本エミュレーション RAM (ERAM) は、内蔵 FLASH メモリのアドレスとオーバーラップして使用することができます。

内蔵 FLASH 領域に ERAM をオーバーラップしているアドレスにライトをすると、ERAM にライトされます。

その際、ブレイクは発生しません。

3. ハードウェア仕様

3.5.2 低消費電力状態（スリープ、ソフトウェアスタンバイ、ハードウェアスタンバイ）

SH7058、SH7058S、SH7059 には、低消費電力状態としてスリープモード、ソフトウェアスタンバイモード、ハードウェアスタンバイモードがあります。

（1）ハードウェアスタンバイモード

E6000H エミュレータでは、ユーザシステムからの_HSTBY 信号を MCU に入力していませんので、ハードウェアスタンバイモードはサポートしていません。

（2）スリープモード、ソフトウェアスタンバイモード

1. ブレークについて

スリープモード、ソフトウェアスタンバイモードは、通常の解除要因の他にブレーク条件の一致（強制ブレーク）によって各状態が解除され、ブレークします。ブレーク後のユーザプログラム再開は、SLEEP命令の次の命令になります。

2. トレースについて

スリープモード、ソフトウェアスタンバイモードでのトレース情報は取得しません。

3. エミュレータ機能によるメモリアクセスについて

スリープモード、ソフトウェアスタンバイモードでのメモリ内容の表示と変更については、デバッグ編「5.3 メモリ内容の表示と変更方法について」を参照してください。

3.5.3 割り込み

実行およびステップ実行中の MCU の割り込みは、すべてユーザに開放しています。

エミュレーション停止中（ブレークモード）は、割り込み要因を保持し、エミュレーション復帰直後、割り込み処理に遷移します。

3.5.4 コントロール用入力信号（_RES、_BREQ、_WAIT）

MCU のコントロール用入力信号として _RES、_BREQ、_WAIT 信号があります。

_RES 信号は、実行によるエミュレーション中のみ有効です。（ステップ実行によるエミュレーション中、_RES 信号は無効になります。）

_BREQ、_WAIT 信号は、メモリ内容の表示と更新、実行、ステップ実行によるエミュレーションで有効です。

エミュレータのエミュレーション停止中（ブレーク中）では、_RES、_BREQ、_WAIT 信号は MCU に入力されません。

実行およびステップ実行中の _RES、_BREQ、_WAIT 信号は[コンフィグレーションプロパティ]ダイアログボックスにより入力を禁止することができます。

3.5.5 WDT（ウォッチドッグタイマ）

エミュレータのエミュレーション停止中（ブレーク中）は、WDT のタイマカウンタ（TCNT）のカウントアップを中断し、エミュレーション実行（ユーザモード）に復帰したときにカウントアップを再開します。

ブレークモード中にも、TCNT にクロックを供給するプリスケラは動作し続けます。ブレークモードに遷移する前後で、プリスケラの位相がずれる可能性があるため、オーバフローまでの時間がプリスケラのクロック周期で前後 1 サイクルずれることがあります。

3.5.6 A/D 変換器

A/D 変換器には、アナログ入力端子の他に AVcc、AVss、Avref、_ADTRG 端子がありますが、A/D 変換器は独立した電源で動作するため、AVcc（電源端子）はユーザシステム上で、必ず A/D 電源に接続してください。

- 【注】
1. A/D 変換器を使用しない場合は、AVcc を Vcc に接続してください。
 2. エバチップボードに実装している MCU とユーザシステムの間には、ユーザシステムインタフェースボード、プリント基板配線および保護回路等があります。このため、変換精度は SH7050 シリーズチップよりも劣ります。A/D 変換器を使用したユーザシステムの最終評価の際は SH7050 シリーズチップ（F-ZTAT マイコン）のご使用をお奨めします。

3.5.7 E6000H エミュレータの状態と内蔵モジュール

E6000H エミュレータの状態により、一部の内蔵モジュールは動作しません。以下に、E6000H エミュレータの状態と内蔵モジュールの関係を示します。

表 3.6 E6000H エミュレータの状態と内蔵モジュール

内蔵モジュール	エミュレーション停止中 (ブレイク中)	エミュレーション中 (実行、ステップ実行)
UBC (ユーザブレイクコントローラ)	動作しない	動作する
DMAC (ダイレクトメモリアクセスコントローラ)	動作する	動作する
ATU-II (アドバンスドタイマユニット)	動作する	動作する
APC (アドバンスドパルスコントローラ)	動作する	動作する
WDT (ウォッチドッグタイマ)	動作しない	動作する
CMT (コンペアマッチタイマ)	動作する	動作する
SCI (シリアルコミュニケーションインタフェース)	動作する	動作する
HCAN2	動作する	動作する
A/D 変換器	動作する	動作する
AUD (アドバンスドユーザデバッグ)	動作する ^{*1}	動作する ^{*1}
I/O ポート	動作する	動作する
H-UDI (ユーザデバッグインタフェース)	使用不可 ^{*2}	使用不可 ^{*2}

【注】 *1 E6000H エミュレータでは、ユーザで使用するか、E6000H エミュレータ機能で使用するかの切替えができます。よって E6000H エミュレータで使用している場合、ユーザは使用できません。

*2 E6000H エミュレータではサポートしておりません。

3.5.8 エミュレータのレジスタ初期値相違点

1. エミュレータは、システム起動時およびコマンドでMCUをパワーオンリセットする場合、汎用レジスタおよびコントロールレジスタの一部を初期化していますので注意してください。

表 3.7 MCU とエミュレータのレジスタ初期値相違点

レジスタ名	エミュレータ		MCU (パワーオンリセット)
	起動時	リセット (Reset CPU)	
PC	パワーオンリセットベクタの PC 値	パワーオンリセットベクタの PC 値	パワーオンリセットベクタの PC 値
R0 ~ R14	H'00000000	リセット前の値	不定
R15 (SP)	パワーオンリセットベクタの SP 値	パワーオンリセットベクタの SP 値	パワーオンリセットベクタの SP 値
SR	H'000000F0	H'000000F0	H'00000XFX*1
GBR	H'00000000	リセット前の値	不定
VBR	H'00000000	H'00000000	H'00000000
MACH	H'00000000	リセット前の値	不定
MACL	H'00000000	リセット前の値	不定
PR	H'00000000	リセット前の値	不定

【注】 X は不定の値

4. 故障解析（故障症状調査書）

4.1 故障解析

E6000H エミュレータ用診断プログラムによる故障解析の手順について示します。

4.1.1 診断プログラムを実行するためのシステムセットアップ

診断プログラムを実行するためには、以下に示す機器が必要です。なお、本テストプログラムの実行時はユーザシステムインタフェースボードおよびユーザシステムを接続しないでください。

- E6000Hエミュレータ（HS7058EPH62H または HS7059EPH62H）
- PC
- E6000 PCインタフェースボード（本補足説明書では、以下のいずれかを指します。PCのインタフェース仕様に合わせて以下のインタフェースボードのいずれか一製品をご用意ください。）
 - PCIバスインタフェースボード（HS6000EIC01H,HS6000EIC02H）
 - PCカードインタフェース（HS6000EIP01H）
 - LANアダプタ（HS6000ELN01H）
 - USBアダプタ（HS6000EIU01H,HS6000EIU02H）

1. PCにE6000PCインタフェースボードを挿入し、付属のPCインタフェースケーブルを接続してください。
2. PCインタフェースケーブルをE6000Hエミュレータ本体に接続してください。
3. E6000Hエミュレータ本体に、付属のAC電源ケーブルを接続してください。
4. PCを起動し、DOSプロンプト（Windows[®] 98SE、Windows[®] Me）またはコマンドプロンプト（Windows NT[®]、Windows[®] 2000、Windows[®] XP）のコマンド入力待ち状態にしてください。

その際、開いた状態が全画面表示でなかった場合、[Alt+Enter] キーを押し、全画面表示に切り替えてください。
また、元の状態に戻す場合も [Alt+Enter] キーを押すと、OSに関係なく、画面表示を切り替えることができます。

【注】 MS-DOS プロンプトにおいて、[Alt+Enter]の操作を行っても画面表示が切り替わらなかった場合は、下記のようにプロパティの[その他]ページの[Windows ショートカットキー]にある[Alt+Enter]のチェックボックスをチェックし、[更新]ボタンをクリックして更新してください。

4. 故障解析（故障症状調査書）

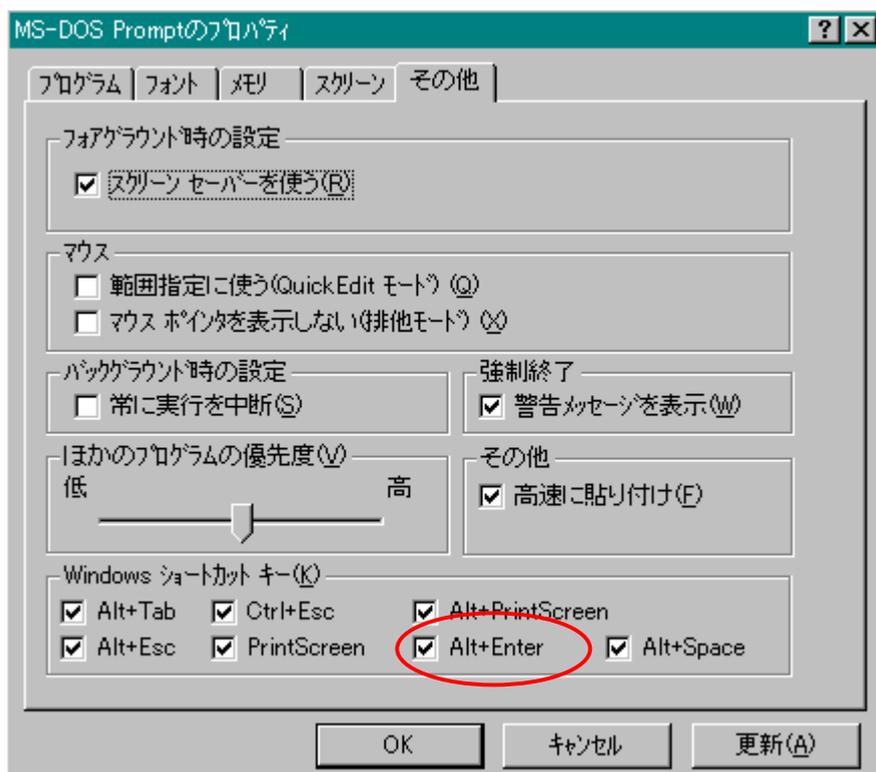


図 4.1 「その他」ページ

5. E6000Hエミュレータ本体の電源をオンにしてください。

【注】 診断プログラムを実行する場合には、必ず、E6000Hエミュレータの電源オンから始めてください。

診断プログラムにおいては、ハードウェアの初期状態のチェックを行っています。

よって、電源オン後、診断プログラムを実行する前に、High-performance Embedded Workshop の起動は行わないでください。

4.1.2 診断プログラムのテスト項目

本診断プログラムのテスト項目を表 4.1 に示します。

表 4.1 診断プログラムのテスト項目

テスト No.	テスト項目	テスト内容
1	Main Board Access	E6000H メインボード内レジスタテスト
2	Emulation Board Access	E6000H エミュレーションボード内レジスタテスト
3	Evaluation Board Access	E6000H エバチップボード内レジスタテスト
4	Basic Function	基本機能のテスト
5	GO to BREAK Time Measurement	実行時間測定機能のテスト
6	Emulation Monitor	エミュレーションモニタのテスト
7	G/A Break Function	G/A ブレーク機能のテスト
8	G/A Performance Analysis Function	G/A パフォーマンス測定機能のテスト
9	G/A Monitor Function	G/A モニタ機能のテスト
10	G/A Parallel RAM Monitor	G/A パラレル RAM モニタ機能のテスト
11	G/A Trace Function	G/A トレース機能のテスト
12	Combination	各機能の組合せテスト
13	Parallel Access	パラレルアクセス機能のテスト

4.1.3 診断プログラムによる故障解析

E6000H エミュレータに添付されている CD-R (HS7058EPH60SR) を PC の CD-ROM ドライブに挿入し、コマンドプロンプトでカレントディレクトリを

HS7058EPH62H をご使用の場合 <ドライブ> : \Diag\HS7058EPH60H

HS7059EPH62H をご使用の場合 <ドライブ> : \Diag\HS7059EPH60H

フォルダに移動した後、使用している PC インタフェースボードの種類に従い、下記コマンドを入力すると直ちにテストプログラムが起動します。なお、診断プログラムの実行に際しては、High-performance Embedded Workshop がインストールされていることを前提としています。

- (1) PCIバスインタフェースボード (HS6000EIC01H, HS6000EIC02H)

HS7058EPH62Hをご使用の場合

>TM7058 -PCI (ENTER)

HS7059EPH62Hをご使用の場合

>TM7059 -PCI (ENTER)

- (2) PCカードインタフェース (HS6000EIP01H)

HS7058EPH62Hをご使用の場合

>TM7058 -PCCD (ENTER)

HS7059EPH62Hをご使用の場合

>TM7059 -PCCD (ENTER)

- (3) LANアダプタ (HS6000ELN01H)

HS7058EPH62Hをご使用の場合

>TM7058 -ELN (ENTER)

HS7059EPH62Hをご使用の場合

>TM7058 -ELN (ENTER)

- (4) USBアダプタ (HS6000EIU01H, HS6000EIU02H)

HS7058EPH62Hをご使用の場合

>TM7058 -USB (ENTER)

HS7059EPH62Hをご使用の場合

>TM7058 -USB (ENTER)

カレントディレクトリを <ドライブ> : \Diag\HS7058EPH60H フォルダに移動しない状態で

<ドライブ> : \Diag\HS7058EPH60H\TM7058 -PCI (ENTER) のように他のカレントディレクトリから診断プログラムを起動した場合は診断プログラムが正しく動作しません。必ず <ドライブ> : \Diag\HS7058EPH60H フォルダにカレントディレクトリを移動して診断プログラムを実行してください。

なお、>TM7058 -PCI -S (ENTER) のように、-S をコマンドラインに追加すると、No.1 から No.13 までのテストを繰り返し実行することができます。途中でテストを中断する場合は Q を入力してください。HS7059EPH62H をご使用の場合も同様です。この場合は 7058 を 7059 に置き換えてください。

- 【注】 1. <ドライブ>は CD-ROM ドライブのドライブ文字です。
2. テストプログラム実行中は CD-ROM ドライブから CD-R を取り出さないでください。

4. 故障解析 (故障症状調査書)

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ようになります。テストは No.1 から No.13 までです (PCI インタフェースボード使用時にテスト時間は約 3 分です)。ここでは HS7058EPH62H を例にとります。

SH7058 E6000H Emulator Tests Vx.x.xx	テストプログラムのスタートメッセージです。 x.x はバージョン番号です。
Loading driverOK (Use PCI)	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。
Initializing driverOK	ホスト PC と E6000H エミュレータが正しく
Searching for interface cardOK	接続されていることを示します。
Checking emulator is connectedOK	
Emulator board information:	
Main board ID: H'0	Emulation board ID: H'001
	E6000H エミュレータの ID 番号を示します。
Normal started at Tue Oct 07 16:52:05 2003	診断プログラムの開始時間を示します。
***** NORMAL TEST - Press 'Q' to stop *****	(COUNT=0001)

1. Main Board Access

01) Registers Initial Value CheckOK
02) Registers Write/VerifyOK
03) DPRAM Address Decode TestSKIP
04) DPRAM Marching TestSKIP
05) Trace Memory Address Decode TestOK
06) Trace Memory Marching TestOK
07) G/A Registers Initial Value CheckOK
08) G/A Registers Write/VerifyOK

2. Emulation Board Access

01) Registers Initial Value CheckOK
02) Registers Write/VerifyOK
03) H-UDI Interface Registers Initial Value CheckOK
04) H-UDI Interface Registers Write/VerifyOK
05) AUD Interface Registers Initial Value CheckOK
06) AUD Interface Registers Write/VerifyOK
07) AUD Interface Registers Address Decode TestOK
08) PCKCR Register Bit TestOK

3. Evaluation Board Access

01) Registers Initial Value CheckOK
02) Registers Write/VerifyOK
03) H-UDI IDCODE CheckOK
04) Firmware BOOTOK
05) Configuration SetOK

4. Basic Function

01) GO to BREAKOK
02) RESET GOOK
03) STEPOK
04) KEYBREAKOK

05) BRKCONT	OK
06) Internal ROM Test	OK
07) Internal RAM Test	OK
5. GO to BREAK Time Measurement	
01) Counter Test Mode	OK
02) EMU 5MHz MPU 20MHz Sampling 20ns (Default)	OK
03) EMU 5MHz MPU 20MHz Sampling 1.6us	OK
04) EMU 5MHz MPU 20MHz Sampling 52us	OK
05) EMU 10MHz MPU 40MHz Sampling MPU	OK
06) EMU 10MHz MPU 40MHz Sampling MPU/2	OK
07) EMU 10MHz MPU 40MHz Sampling MPU/4	OK
08) EMU 5MHz MPU 20MHz Sampling 20ns	OK
09) EMU 10MHz MPU 80MHz Sampling 20ns	OK
6. Emulation Monitor	
01) AUDRES	OK
02) TRES	OK
03) ASEST1 - 0	OK
04) VCC3VNG	OK
05) VCC2-5VNG	OK
06) VCC1-NG	OK
7. G/A Break Function	
01) Address Condition	OK
02) Data Condition	OK
03) Control Signal Condition (ASEDSHH/HL/LH/HL)	OK
04) Function Code Condition (ASEBM1 ASEAS2-0)	OK
05) Control Signal Condition (ASEAS2-1)	OK
06) Control Signal Condition (ASEIF-N)	OK
07) Control Signal Condition (ASEERAM-N)	OK
08) Control Signal Condition (/DMA_AUD ASEBM0)	OK
09) Control Signal Condition (ASEST1-0)	OK
8. G/A Performance Analysis Function	
01) Time Measurement (20ns Sampling)	OK
9. G/A Monitor Function	
01) STEP/RUN	OK
02) VCCDOWN	OK
03) NOCLK	OK
04) TIMEOUT	OK
10. G/A Parallel RAM Monitor	
01) PRAM Monitor (BYTE)	OK
02) PRAM Monitor (WORD)	OK
03) PRAM Monitor (LONG WORD)	OK
11. G/A Trace Function	
01) Free Trace	OK

4. 故障解析 (故障症状調査書)

- 02) Trace StopOK
- 03) Time StampOK
- 04) Trace SuppressOK

12. Combination

- 01) B to A Time Measurement (FPGA counter)OK
- 02) B to A Time Measurement (G/A counter)OK
- 03) D to C Time Measurement (G/A counter)OK

13. Parallel Access

- 01) AUD Write (INROM BYTE)OK
- 02) AUD Write (INROM WORD)OK
- 03) AUD Write (INROM LONGWORD)OK
- 04) AUD Write (INRAM BYTE)OK
- 05) AUD Write (INRAM WORD)OK
- 06) AUD Write (INRAM LONGWORD)OK
- 07) AUD Sample (INROM BYTE)OK
- 08) AUD Sample (INROM WORD)OK
- 09) AUD Sample (INROM LONGWORD)OK
- 10) AUD Sample (INRAM BYTE)OK
- 11) AUD Sample (INRAM WORD)OK
- 12) AUD Sample (INRAM LONGWORD)OK
- 13) AUD Sample 256pointOK

Normal stopped at Tue Oct 07 16:54:23 2003

Tests run for 0h:2min:18s

Summary:

Tests performed 1 time(s).

1. Main Board Access	: 0 Error(s)
2. Emulation Board Access	: 0 Error(s)
3. Evaluation Board Access	: 0 Error(s)
4. Basic Function	: 0 Error(s)
5. GO to BREAK Time Measurement	: 0 Error(s)
6. Emulation Monitor	: 0 Error(s)
7. G/A Break Function	: 0 Error(s)
8. G/A Performance Analysis Function	: 0 Error(s)
9. G/A Monitor Function	: 0 Error(s)
10. G/A Parallel RAM Monitor	: 0 Error(s)
11. G/A Trace Function	: 0 Error(s)
12. Combination	: 0 Error(s)
13. Parallel Access	: 0 Error(s)

診断プログラムの終了時間を示します。

診断プログラムの実行時間を示します。

各テスト項目でのエラー発生数の合計を示します。

4. 故障解析 (故障症状調査書)

4.2 故障症状調査書

貴社益々ご清栄のこととお喜び申し上げます。

この度、E6000H エミュレータ (HS7058EPH62H / HS7059EPH62H) をご購入頂き、厚く御礼申し上げます。

さて、万一故障が発生したときには、お手数ですが次ページの故障症状調査書に症状をご記入の上、担当営業まで御連絡くださいますようお願い申し上げます。

故障症状調査書

ご購入営業担当 行

お客様ご芳名 会社名 _____
 担当者名 _____様
 TEL _____

調査項目	症 状
1 故障発生 年月日、時期	西暦 年 月 日 {システム立ち上げ時、システム動作時} *{ }内の該当時期を で囲んでください。
2 故障発生頻度	() {日、週、月}に()回発生 * ()内に該当数字を記入し、{ }内の該当時期を で囲んでください。
3 エラー発生時の システム構成	* { }内の該当箇所を で囲んでください。 (1) エミュレータ側のシステム構成 • E6000H エミュレータ (HS7058EPH62H, HS7059EPH62H) シリアル No. レビジョン (ケース裏面に表示しています：シリアル No.は数字4桁、レビジョンはそれに続くアルファベットです) • ホストインタフェース PCI インタフェースボード {HS6000EIC01H・HS6000EIC02H} PC カードインタフェース {HS6000EIP01H} LAN アダプタ {HS6000ELN01H} USB アダプタ {HS6000EIU01H・HS6000EIU02H} シリアル No. レビジョン (基板上に捺印表示しています) • 付属 CD-R (HS7058EPH60SR) バージョン V (CD-R に V.x.xx Release xx と表示しています (x: 数字)) • ご使用になっている PC メーカー名 型式 使用 OS {Windows®98SE、Windows® Me、WindowsNT®4.0、Windows®2000、Windows®XP} (2) 実機の接続 {YES、NO} (3) ユーザシステムインタフェースボードの有無 {有・無} 型名
4 エラー発生時の 設定内容	(1) 動作モード：モード (2) ターゲットシステム電圧： V (3) 使用クロック： (貸出しクロック、Xtal 発振子、外部クロック入力 いずれかに) (4) 動作周波数： MHz

調査項目	症 状
5 故障現象	
6 デバッグ時のエラー内容	
7 診断プログラムでのエラー内容	
8 High-performance Embedded Workshop がエミュレータと接続できない	エラーメッセージ内容

上記以外エラーについては、下記に症状を記載いただくようお願いいたします。

--

デバッグ編

1. 概要

デバッグ編では、以下の内容を説明しています。

表1.1 デバッグ編について

章	タイトル	内 容
2	デバッグの準備をする	この章では、ワークスペースの作成から本エミュレータとの接続までを説明しています。
3	デバッグ	High-performance Embedded Workshopで共通な下記機能については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。 <ul style="list-style-type: none">・ デバッグの準備・ プログラムを表示する・ メモリ内容を参照 / 設定する・ メモリ内容を波形形式で表示する・ メモリ内容を画像形式で表示する・ 変数を参照 / 設定する・ I/Oレジスタを参照 / 設定する・ レジスタを参照 / 設定する・ プログラムを実行 / 停止、リセットする・ 関数呼び出し履歴を見る・ コマンドラインインタフェースのデバッグ・ Elf / Dwarf2のサポート・ ラベルを参照 / 設定する
4	チュートリアル	本製品ではチュートリアルプログラムを提供しています。この章ではチュートリアルプログラムを用いて本エミュレータの主な機能の使用方法を説明しています。
5	本製品固有のソフトウェア仕様と注意事項	この章では、本エミュレータに関するソフトウェア仕様および注意事項について説明しています。
6	エラーメッセージ	この章では、本エミュレータ使用時に発生するエラーメッセージの内容と対策を説明しています。

2. デバッグの準備をする

2.1 High-performance Embedded Workshop の起動方法

High-performance Embedded Workshop は以下の手順で起動します。

- (1) ホストコンピュータとE6000Hエミュレータを接続してください。
- (2) ユーザシステムインタフェースケーブルをご使用の場合は、E6000Hエミュレータのコネクタとユーザシステムインタフェースケーブルを接続します。ユーザシステムインタフェースケーブル未使用の場合は本手順は不要です。
E6000Hエミュレータの電源を入れてください。ユーザシステムをご使用の場合は、E6000Hエミュレータの電源を入れる前に、ユーザシステムの電源を入れてください。
- (3) [スタート]メニューの[プログラム]からHigh-performance Embedded Workshopを起動してください。
- (4) [ようこそ!]ダイアログボックスが表示されます。

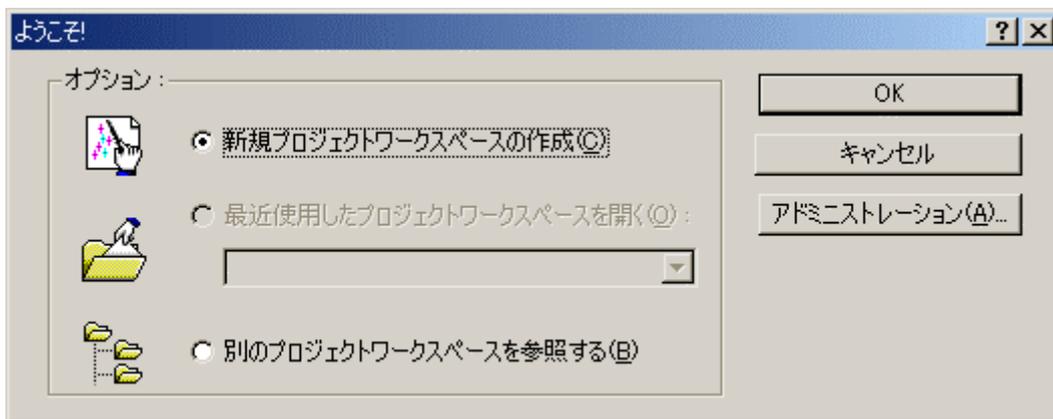


図2.1 ようこそ!ダイアログボックス

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

ここでは、以下の3つの方法を説明します。

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

2 デバッグの準備をする

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

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

2.1.1 新規にワークスペースを作成する場合（ツールチェイン未使用）

- (1) [ようこそ!]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

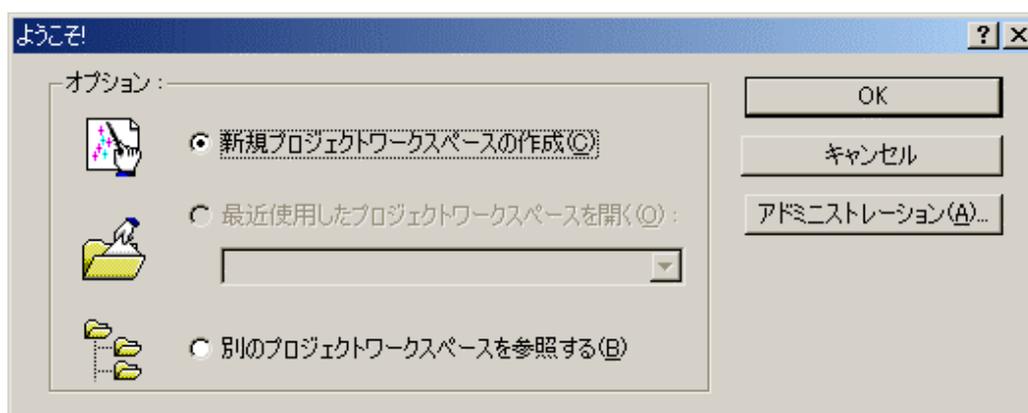


図2.2 ようこそ!ダイアログボックス

- (2) 新規プロジェクトワークスペースの作成を開始します。

以下の画面が開きます。

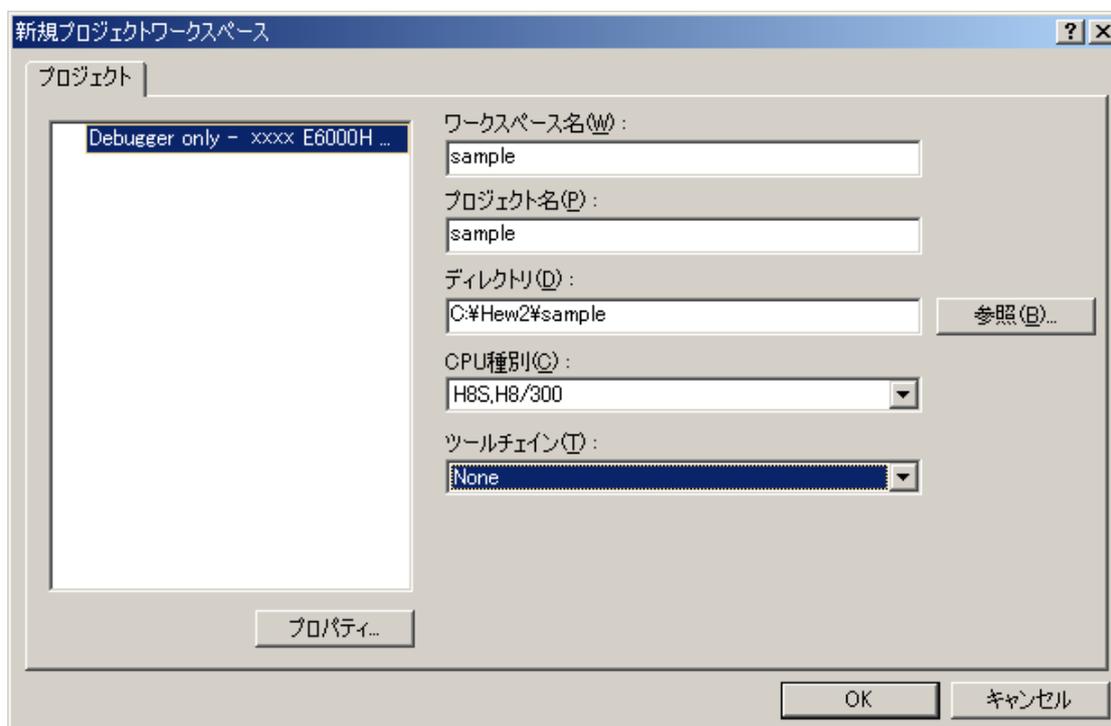


図2.3 新規プロジェクトワークスペースダイアログボックス

- [ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。
- [プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じでよければ、入力する必要はありません。
- [ディレクトリ]エディットボックス
ワークスペースを作成するディレクトリを入力してください。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することもできます。
- [CPU種別] ドロップダウンリストボックス
該当するCPUファミリを選択してください。

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

[OK]ボタンをクリックしてください。

- (3) 次に、セッションファイルのターゲットプラットフォームを選択します。

以下の画面が表示されます。

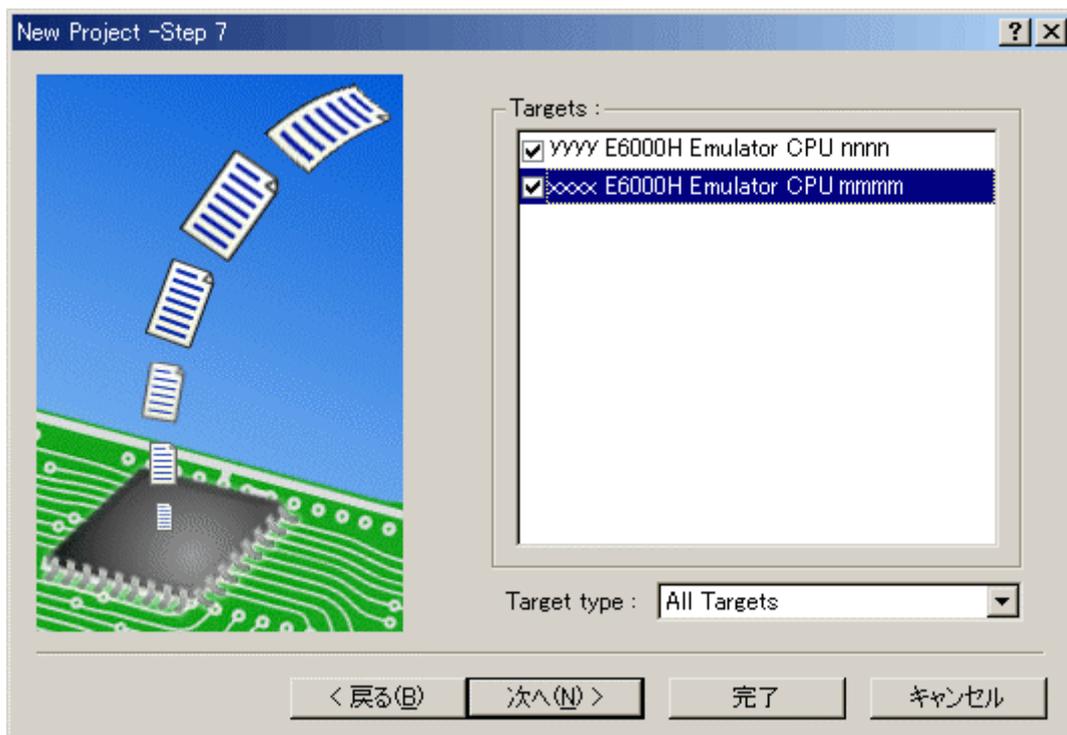


図2.4 デバッガターゲットダイアログボックス

ここでは、セッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンをクリックしてください。セッションファイルについての詳細は High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

2 デバッグの準備をする

(4) 次に、コンフィグレーションファイル名を設定します。

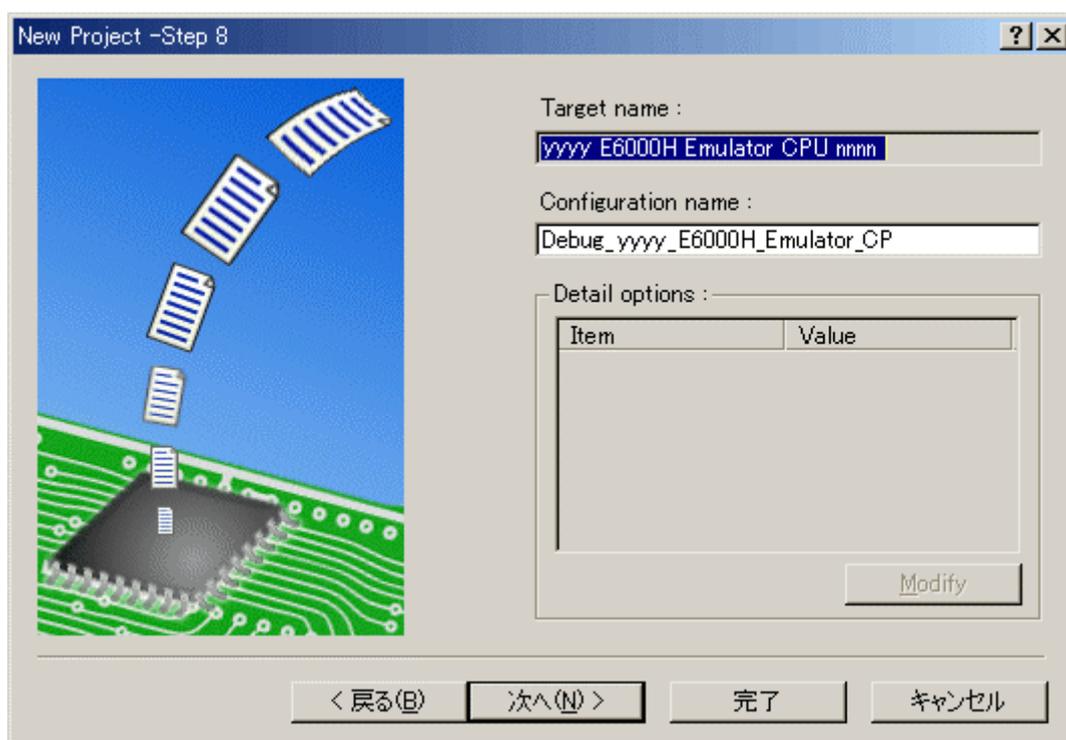


図2.5 デバッガオプションダイアログボックス

[デバッガターゲット]ダイアログボックス(図2.4)で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンをクリックし、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了したら、E6000H エミュレータに関する設定は終了です。

[完了]ボタンをクリックすると[概要]ダイアログボックスが表示されます。[OK]ボタンをクリックすると、High-performance Embedded Workshop が起動します。

(5) 起動後、自動的にE6000Hエミュレータが接続されます。

接続が完了すると、[アウトプット]ウィンドウの[Debug]タブに「Connected」と表示されます。

2.1.2 新規にワークスペースを作成する場合(ツールチェーン使用)

(1) [ようこそ!]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

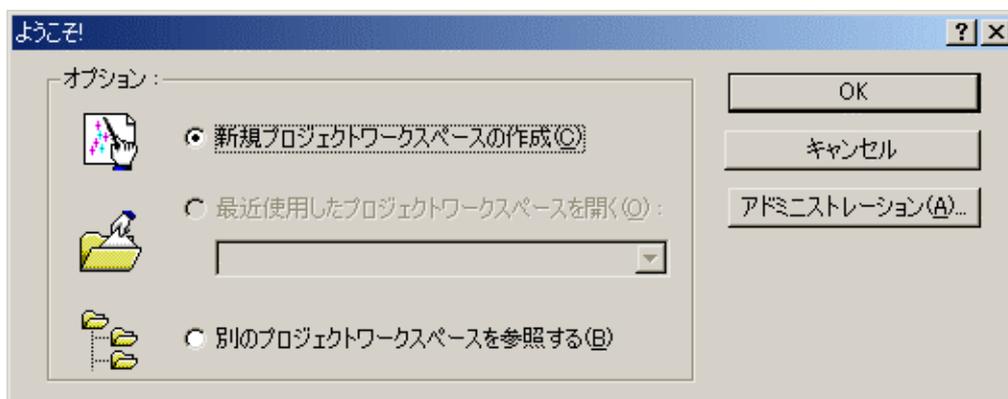


図2.6 ようこそ!ダイアログボックス

(2) 新規プロジェクトワークスペースを作成します。

以下の画面が開きます。

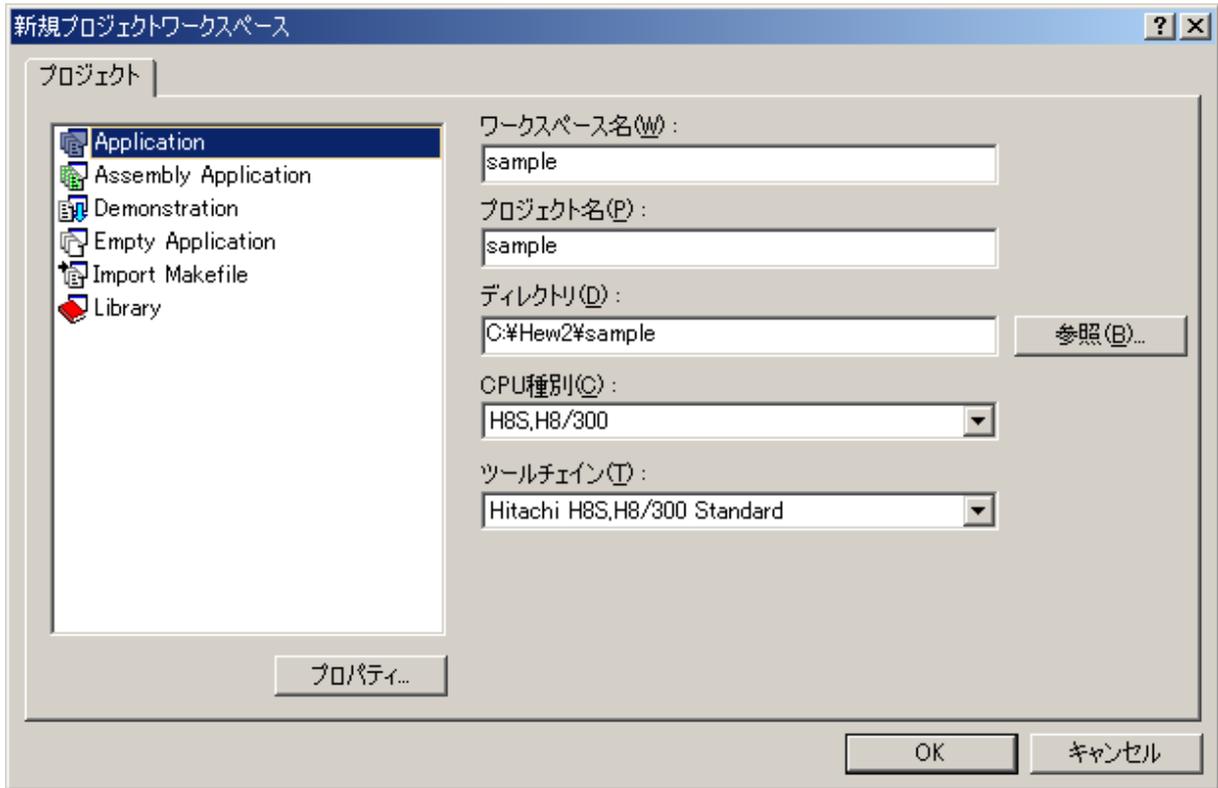


図2.7 新規プロジェクトワークスペースダイアログボックス

- [ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。
- [プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じであれば、入力する必要はありません。
- [ディレクトリ]エディットボックス
ワークスペースを作成するディレクトリを入力してください。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することもできます。
- [CPU種別]ドロップダウンリストボックス
該当するCPUファミリを選択してください。
- [ツールチェーン]ドロップダウンリストボックス
ツールチェーンをご使用になる場合、該当するツールチェーン名を選択してください。
使用しない場合、[None]を選択してください。
- [プロジェクトタイプ]リストボックス
使用したいプロジェクトタイプを選択してください。

2 デバッグの準備をする

【留意事項】

E6000H エミュレータの場合、[Demonstration]を選択した場合に以下の注意事項があります。

[Demonstration]は H8S, H8/300 コンパイラパッケージまたは SuperH RISC engine C/C++コンパイラパッケージ付属のシミュレータ用のプログラムです。生成されたソースファイルを使用する場合、ソースファイル中の"Printf 文"を削除してください。

(3) 次に、ツールチェーンの設定を行いますので、必要な設定を行ってください。

ツールチェーンの設定が終了したら、以下の画面が表示されます。

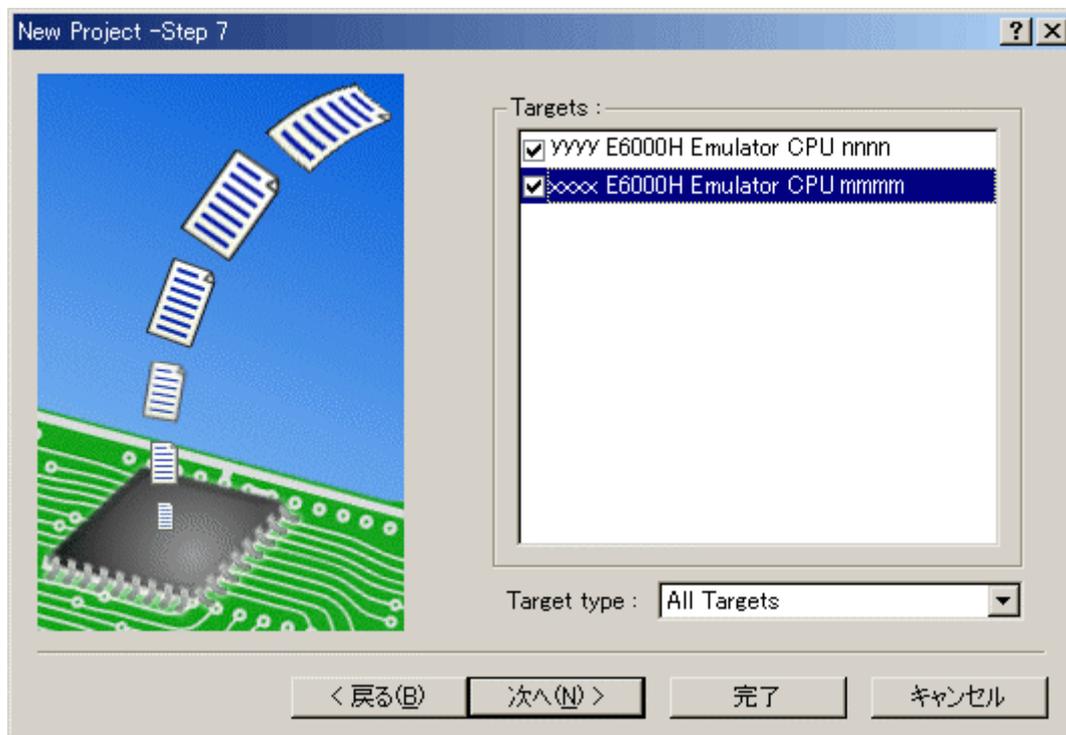


図2.8 新規プロジェクト-7/9-デバッガダイアログボックス

ここでは、セッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンを押してください。

(4) 次に、コンフィグレーションファイル名を設定します。

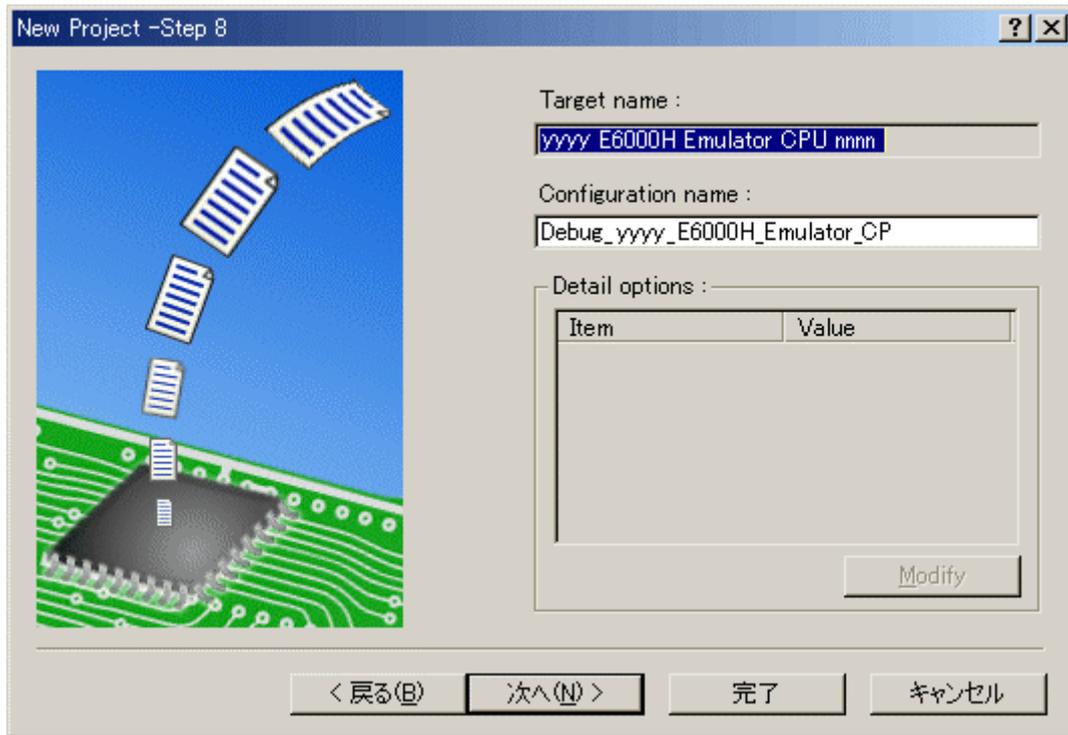


図2.9 新規プロジェクト-8/9-デバッグオプションダイアログボックス

[新規プロジェクト-7/9-デバッグ]ダイアログボックス(図 2.8)で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンをクリックし、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了したら、E6000H エミュレータに関する設定は終了です。

画面の指示に従い、新規ワークスペースの作成を完了してください。High-performance Embedded Workshop が起動します。

(5) 起動後、E6000H エミュレータを接続してください。

E6000H エミュレータは、High-performance Embedded Workshop 起動後すぐに接続する必要はありません。

E6000H エミュレータを接続する場合は、E6000H エミュレータ起動時の設定を行ってから接続する方法と E6000H エミュレータ起動時の設定を行わずに簡単に接続する方法があります。エミュレータの接続についての詳細は「2.2 エミュレータの接続」を参照してください。

2 デバッグの準備をする

2.1.3 既存のワークスペースを指定する場合

- (1) [ようこそ!]ダイアログボックスで、[別のプロジェクトワークスペースを参照する]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

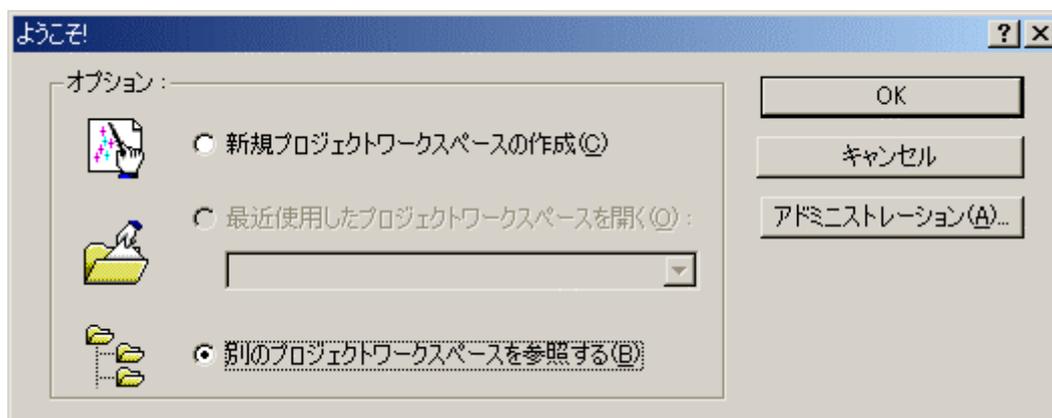


図2.10 ようこそ!ダイアログボックス

- (2) [ワークスペースを開く]ダイアログボックスが開きますので、ワークスペースが作成されているディレクトリを指定してください。

ディレクトリの指定後、ワークスペースファイル (拡張子 .hws) を選択し[開く]ボタンをクリックしてください。

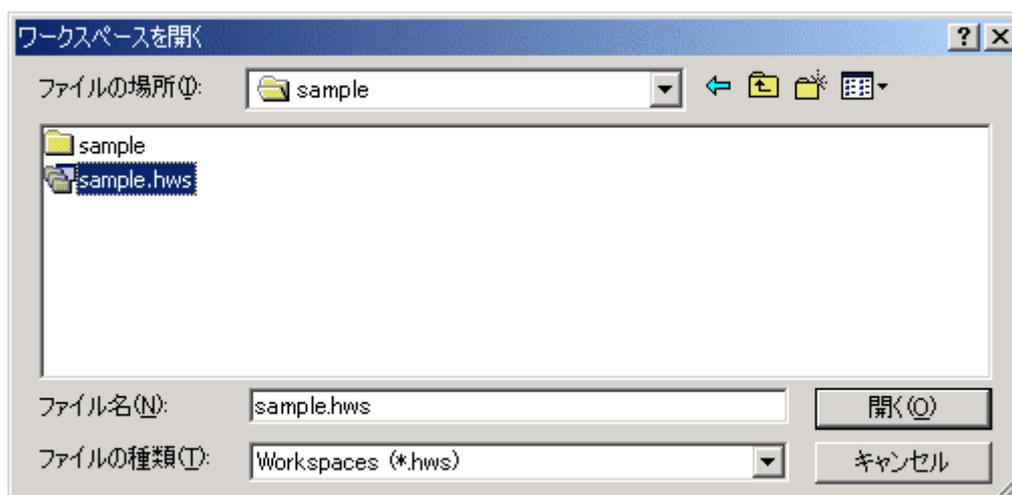


図2.11 ワークスペースを開くダイアログボックス

- (3) High-performance Embedded Workshop が起動され、指定したワークスペースの保存状態が復元されます。

指定したワークスペースがエミュレータに接続された状態を保存していた場合には、エミュレータへの接続が自動で行われます。指定したワークスペースがエミュレータに接続されていない状態を保存していた場合に、エミュレータの接続を行う場合は、「2.2 エミュレータの接続」を参照してください。

2.2 エミュレータの接続

エミュレータの接続には、以下の方法があります。

(1) E6000H エミュレータ起動時の設定を行ってから接続する方法

[デバッグ -> デバッグの設定...]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェーンなどを登録することができます。

[デバッグの設定]ダイアログボックスの設定終了後、ダイアログボックスを閉じると、E6000H エミュレータが接続されます。

(2) E6000H エミュレータ起動時の設定を行わずに簡単に接続する方法

E6000H エミュレータを使用する設定があらかじめ登録されているセッションファイルに切り替えることにより、E6000H エミュレータを簡単に接続できます。

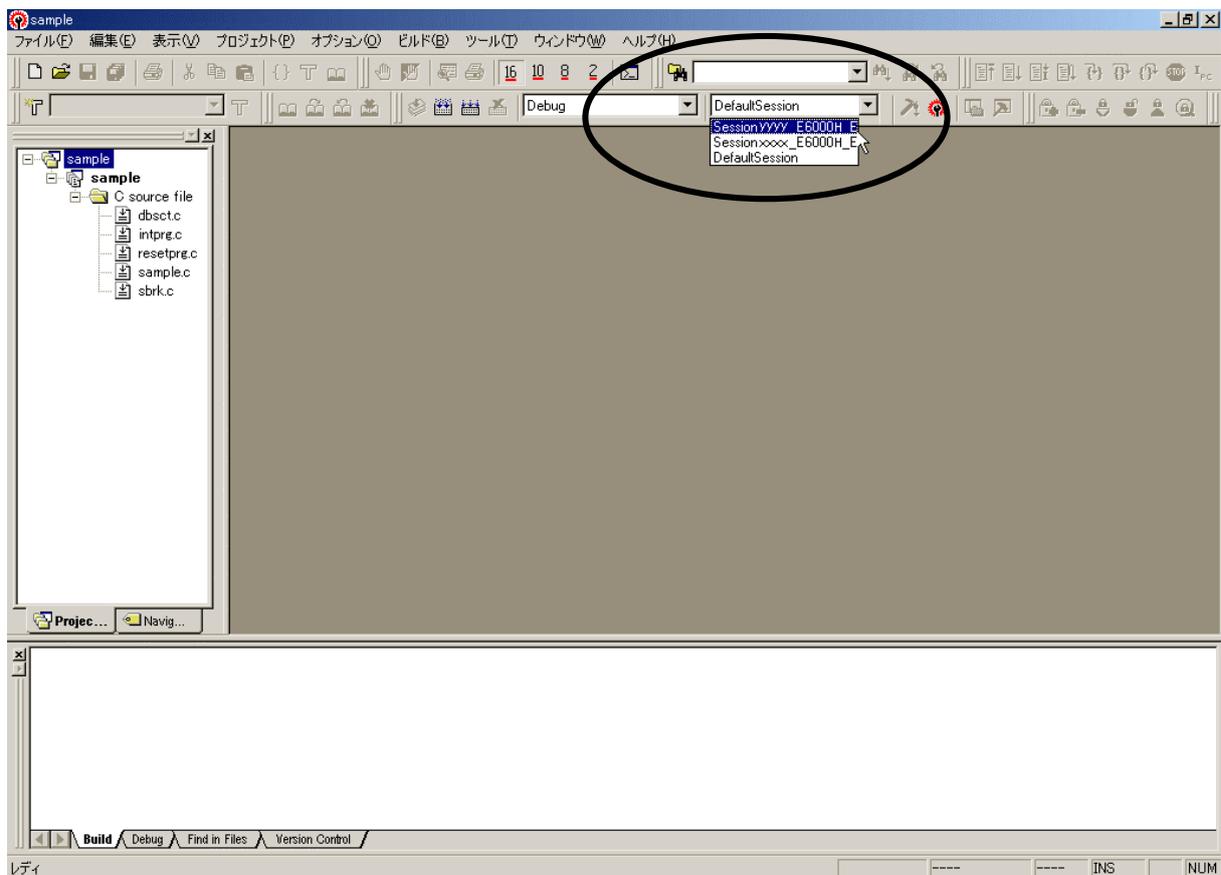


図2.12 セッションファイルの選択

上記図中の、丸印の中にあるリストボックスから、[新規プロジェクト-8/9-デバッガオプション]ダイアログボックス（図 2.9）の[ターゲット名]テキストボックス内で設定されている文字列を含んだセッションファイル名を選択してください。

このセッションファイルには、E6000H エミュレータを使用する設定が登録されています。

[デバッグ->接続]を選択する事で、接続されます。

2 デバッグの準備をする

2.3 エミュレータの再接続

エミュレータ切断状態時に以下の方法で再接続を行うことができます。

[デバッグ -> 接続]を選択するか、接続ツールバーボタン  をクリックしてください。

エミュレータの接続が開始されます。

【注】 エミュレータの再接続を行う場合、あらかじめロードモジュールが High-performance Embedded Workshop に登録されている必要があります。

2.4 エミュレータの終了

エミュレータの終了方法は2通りあります。

- ・ 起動中のエミュレータの接続を解除する方法
- ・ High-performance Embedded Workshop 自体を終了する方法

(1) 起動中のエミュレータの接続を解除する方法

[デバッグ -> 接続解除]を選択するか、接続解除ツールバーボタン  をクリックしてください。

(2) High-performance Embedded Workshop 自体を終了する方法

[ファイル->アプリケーションの終了]を選択してください。

メッセージボックスが表示されます。必要なら、[はい]ボタンをクリックし、セッションをセーブしてください。セーブ後、High-performance Embedded Workshop は終了します。不要なら、[いいえ]ボタンをクリックしてください。High-performance Embedded Workshop は終了します。

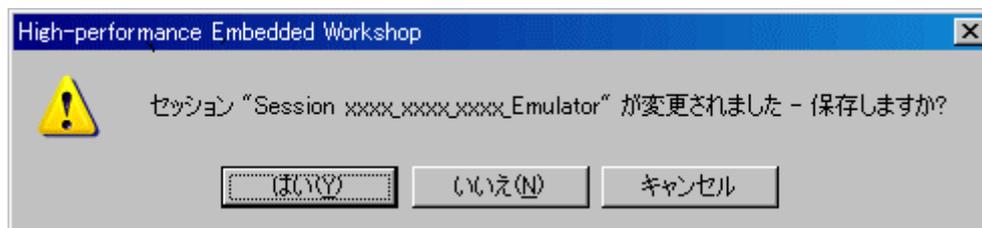


図2.13 セッション保存確認メッセージボックス

3. デバッグ

デバッグ操作と関連するウィンドウおよびダイアログボックスについて説明します。

3.1 エミュレーション環境を設定する

この節では、エミュレーションを行うための環境を設定する方法を説明します。デバッグを開始する前にエミュレーション環境を正しく設定する必要があります。

3.1.1 コンフィグレーションプロパティダイアログボックスを開く

[基本設定 -> エミュレータ -> システム...]を選択するか、[エミュレータシステム]ツールバーボタンをクリックすると、[コンフィグレーションプロパティ]ダイアログボックスが開きます。

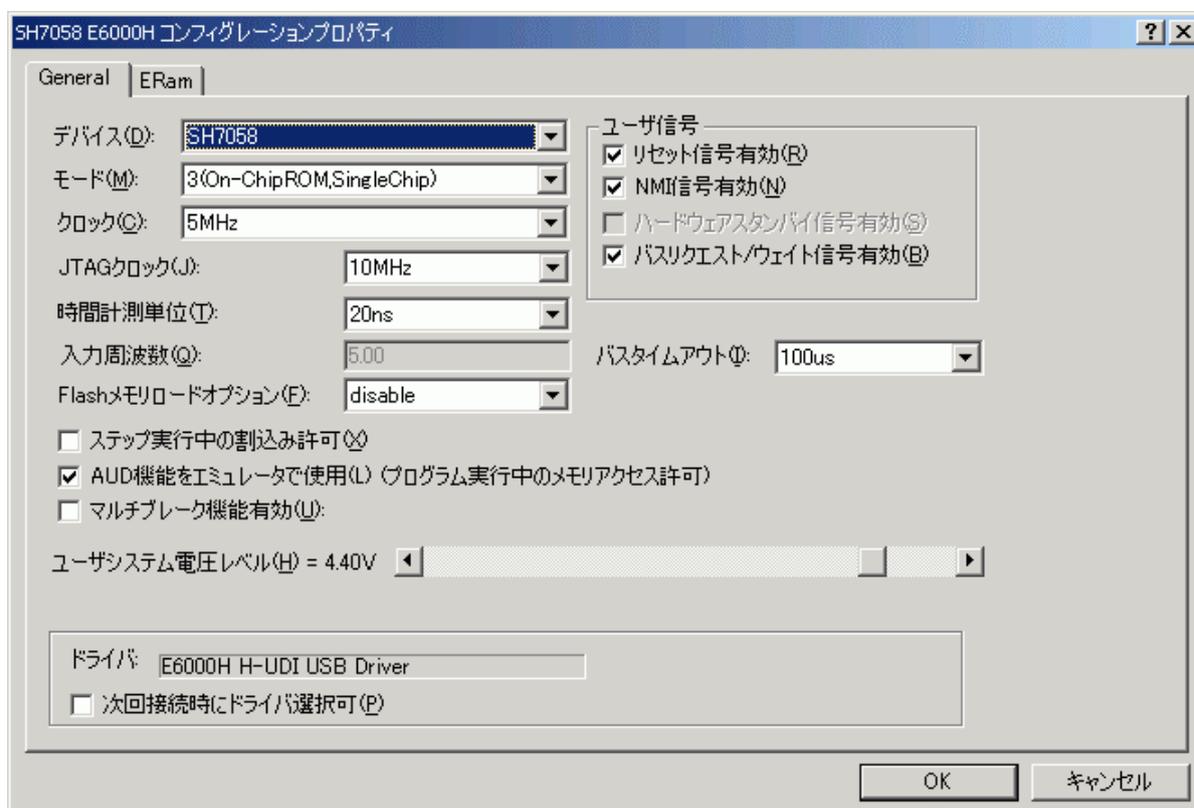


図3.1 コンフィグレーションプロパティダイアログボックス

3 デバッグ

[General]

[デバイス]	エミュレーションするターゲットマイコンを指定します。詳細に関しては、それぞれのハードウェアマニュアルを参照してください。
[モード]	ターゲットマイコンの動作モードを指定します。動作モードには、0 ~ 3、Target の 5 つのモードがあります。 0 : 8 ビットバスモード 1 : 16 ビットバスモード 2 : 内蔵 ROM 有効モード 3 : 内蔵 ROM 有効、シングルチップモード Target ユーザシステム上で設定されているモード
[クロック]	ターゲットマイコンの入力クロックを指定します。 Target : ターゲットクロック Xtal : エバチップボード上の水晶発振子 xMHz : 内蔵のクロック (HS7058EPH62 使用時 : :x = 5,10 HS7059EPH62H 使用時 : :x = 5,6,8,10)
[JTAG クロック]	JTAG クロックを設定します。 HS7058EPH62H 使用時 : 10, 15, 20 (MHz) HS7059EPH62H 使用時 : 5, 6, 8, 10, 12, 12.5, 15, 18, 20 (MHz)
[時間計測単位]	実行時間の測定およびパフォーマンスに使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 実時間測定 52us、1.6us、20ns クロックカウンタ測定 CLOCK、CLOCK/2、CLOCK/4、CLOCK/8
[入力周波数]	ターゲットマイコンの入力クロックを指定します。クロック選択で Target または Xtal を選択時に入力されている周波数を設定してください。5.00MHz から 10.00MHz の範囲で、小数点第 3 位で四捨五入し、小数点第 2 位まで入力してください。 入力周波数は内蔵フラッシュメモリへのメモリロードのために使用します。
[Flash メモリロードオプション]	内蔵フラッシュメモリへのメモリロードの方法を選択してください。 disable 内蔵フラッシュメモリに対し、プログラムのロードを禁止します。プログラムのロードを実行しても内蔵フラッシュメモリへの書き込みは発生しません。 update 内蔵フラッシュメモリにプログラムのロードを行うモードです。このモードでは、ロードするプログラム以外の内蔵フラッシュメモリの内容はロード実行前と同じになります。 erase 内蔵フラッシュメモリにプログラムのロードを行うモードです。このモードでは、ロードを実行する際、該当する内蔵フラッシュメモリのブロックを消去した状態でロードを行います。 all erase 内蔵フラッシュメモリにプログラムのロードを行うモードです。このモードでは、ロードを実行する際、内蔵フラッシュメモリを全て消去して、ロードを行います。
[ステップ実行中の割り込み許可]	このチェックボックスをチェックすると、STEP 実行中の割り込みを受け付けます。
[AUD 機能をエミュレータで使用 (プログラム実行中のメモリアクセス許可)]	このチェックボックスをチェックすると、ターゲットマイコンの AUD 機能をエミュレータで使用し、ユーザプログラム実行中にメモリ内容の表示および変更を行うことができます。
[マルチブレイク機能有効]	このチェックボックスをチェックすると、マルチブレイク機能が有効となります。マルチブレイク機能を有効にすると、トリガ出力とプローブ端子を使用して複数の E6000H を同時にブレイクすることが出来ます。
[ユーザシステム電圧レベル]	ユーザシステム電圧レベルを指定します。 ユーザ VCC が指定値よりも下がった場合、[拡張モニタ]ウィンドウの[User PVCC1]に[Down]が表示されます。 動作モード変更時に電圧レベルの設定は、動作モード毎の初期値に変更されます。
[ユーザ信号]	このグループボックス内のチェックボックスをチェックすると、対応する制御端子の入力が有効になります。
[バスタイムアウト]	バスタイムアウト検出時間を指定します。100us、1.6ms、13ms、210ms が指定できます。

[ドライバ]	現在使用している E6000H ドライバを表示しています。
[次回接続時にドライバ選択可]	このチェックボックスをチェックすると、次回 E6000H 接続時にドライバを選択することができます。

【注】 クロックカウンタ設定の入力は、内部クロック () になります。

【注】 エミュレータは、AUD 機能を使用してパラレルアクセス (ユーザプログラム実行中のメモリのリード/ライト) を実現します。[AUD 機能をエミュレータで使用 (プログラム実行中のメモリアクセス許可)]のチェックを外し、ユーザー機能として AUD を用いる場合、プログラム実行中のメモリアクセスが出来なくなります。

(1) 動作モードの変更について

動作モードの変更を行うと ERAM、ソフトブレイクの設定が解除されます。また、CPU リセットを行います。Target を選択した時、ユーザシステム上のモードが不正な場合、エラーを表示します。正しいモードを選択してください。

(2) ターゲットマイコンの入力クロックについて

クロックを選択する場合、以下の事を確認してください。

Target を選択する場合は、ユーザ電源の有無を確認してください。ユーザ電源が無い場合、OK を押した後、エラーを表示します。

Xtal を選択する場合、パチップボード上の水晶発振子取り付け端子に水晶発振子の有無を確認してください。無い場合、不正な動作になります。

クロックを変更するとターゲットマイコンをリセットします。

(3) JTAG Clock について

E6000H エミュレータは、ターゲットマイコンとシリアル通信を行うことでデバッグ機能を実現しています。JTAG クロックはこのシリアル通信のための入力クロックとして使用します。JTAG クロックの設定をできるだけ大きな値に指定することで、ダウンロード性能およびメモリアクセス性能を向上することができます。ただし、JTAG クロックは、ターゲットマイコンの周辺モジュールクロック (P) より低くなるように設定してください。

3 デバッグ

3.1.2 エミュレーション RAM を使用する

[コンフィグレーションプロパティ]ダイアログボックスの[ERam]ページにて、エミュレーション RAM の設定を行うことができます。エミュレーション RAM は、HS7058EPH62H では 16k バイト単位に 16 ブロックを、HS7059EPH62H では 16k バイト単位に 24 ブロックを設定することができます。

エミュレーション RAM は、内蔵フラッシュメモリのアドレスとオーバーラップして使用することができます。エミュレーション RAM を使用することで、内蔵フラッシュメモリ上のプログラムまたはデータを書き換えずにデバッグを進めることができます。

エミュレーション RAM を E6000H にて使用しない場合は、デバッグ用の内蔵 RAM として使用することができます。

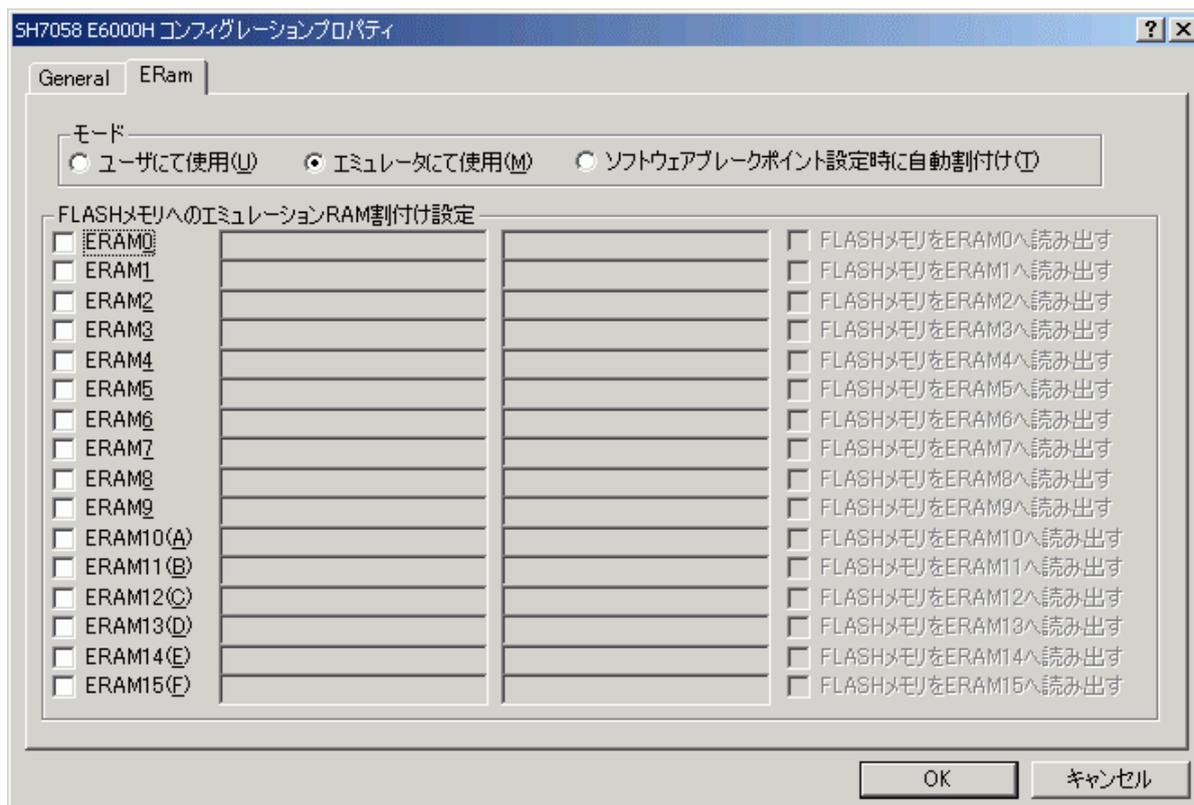


図3-2 コンフィグレーションプロパティダイアログボックス (ERam ページ)

[ERam]

[モード]

[ユーザにて使用]	エミュレーション RAM を E6000H にて使用しないモードです。
[エミュレータにて使用]	エミュレーション RAM を E6000H にて使用するモードです。 [FLASH メモリへのエミュレーション RAM 割付け設定]を設定することで、内蔵フラッシュメモリのアドレスとオーバーラップして使用することができます。
[ソフトウェアブレイクポイント設定時に自動割付け]	エミュレーション RAM を E6000H にて使用しかつ、内蔵フラッシュメモリへソフトウェアブレイクを設定する際、自動的にエミュレーション RAM を割り付けるモードです。
[FLASH メモリへのエミュレーション RAM 割付け設定]	内蔵フラッシュメモリのアドレスとオーバーラップして使用するエミュレーション RAM を設定します。
先頭のチェックボックス	このチェックボックスをチェックすると、右欄で指定されたアドレス上の内蔵フラッシュメモリにエミュレーション RAM を割り付けます。チェックした状態からチェックを外して、OK ボタンをクリックすると、"ERAMn の値を FLASH メモリにコピーします。よろしいですか?"とメッセージが表示されます。この時、"はい(Y)"をクリックすると、エミュレーション RAM の内容を内蔵フラッシュメモリに書き込みます。エミュレーション RAM のメモリ内容は、以下の場合、内蔵フラッシュメモリに書き込みを行いません。 [モード]を[エミュレータにて使用]から[ユーザにて使用]に変更した時。 [モード]を[ソフトウェアブレイクポイント設定時に自動割付け]から[ユーザにて使用]に変更した時。 システム終了時。 また、内蔵フラッシュメモリへの書き込み時、エラーとなった場合は "FLASH メモリアクセスエラー" を表示し、ダイアログを再オープンしますので 設定内容を確認し OK ボタンをクリックしてください。
アドレス入力欄	割り付ける先頭アドレスを設定してください。次の欄に割り付けた範囲を表示します。 アドレス範囲以外の表示は、入力値が不正である事を示します。 入力アドレスは 16k バイト境界に補正されます。
最終のチェックボックス	このチェックボックスをチェックすると、エミュレーション RAM を割り付ける際に該当エリアの内蔵フラッシュメモリの内容を読出し、割付け後内容をエミュレーション RAM に書き込みます。また、このチェックボックスをチェックした状態で内蔵フラッシュメモリへプログラムのロード実行すると、エミュレーション RAM にもプログラムがロードされます。

【注】 エミュレーション RAM 上に設定されたソフトウェアブレイクは、エミュレーション RAM を解除すると機能しなくなります。また、動作モードの変更を行うとエミュレーション RAM の設定は解除されます。

【注】 エミュレーション RAM の設定時には DMAC の動作を停止してください。

(1) エミュレーション RAM をデバッグ用の内蔵 RAM として使用する

エミュレーション RAM をデバッグ用の内蔵 RAM として使用する場合は、ERam ページにて[モード]を[ユーザにて使用]に設定してください。使用できるデバッグ用の内蔵 RAM のアドレスは、H'FFFB 0000 ~ H'FFFE FFFF になります。

エミュレーション RAM は E6000H エミュレータ専用の機能です。製品版ターゲットマイコンでは、この領域はリザーブになっているため使用できません。

3 デバッグ

3.1.3 接続するインタフェースを選択する

[コンフィグレーションプロパティ]ダイアログボックスの[次回接続時にドライバ選択可]項目をチェックすると、次回 E6000H 接続時にドライバを選択することができます。

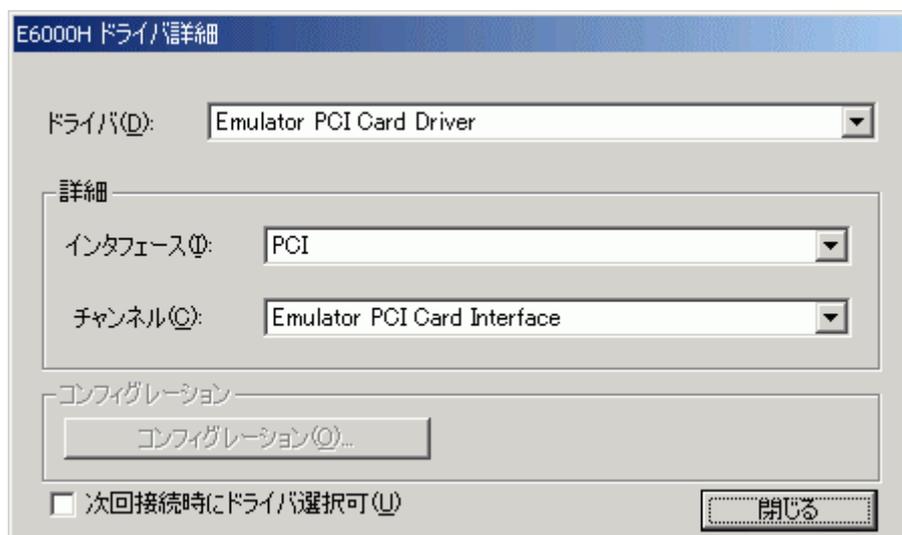


図3.3 E6000H ドライバ詳細ダイアログボックス

[ドライバ]	High-performance Embedded Workshop と E6000H エミュレータを接続するドライバを選択します。
[詳細]	接続するドライバの詳細を設定します。
[インタフェース]	選択したドライバがサポートしているインタフェースを選択します。 本エミュレータでは設定を変更する必要はありません。
[チャンネル]	選択したインタフェースのチャンネルを選択します。 本エミュレータでは設定を変更する必要はありません。
[コンフィグレーション]	ドライバの設定を行います。
[コンフィグレーション...]	ドライバがコンフィグレーションダイアログをサポートしている場合、設定ダイアログボックスを表示します。(本エミュレータでは使用できません。)
[次回接続時にドライバ選択可]	このチェックボックスをチェックすると、次回 E6000H 接続時にドライバを選択することができます。

3.2 プログラムをダウンロードする

プログラムをダウンロードし、ソースコードおよびアセンブリ言語モニターとして見る方法を説明します。

【注】 ユーザプログラムがブレイクした時、[エディタ]ウィンドウは現在の PC がある位置を表示します。ソースファイルを自動的に見つけることができない場合（例えば、ELF/DWARF2 をベースにしたプロジェクトがもともとのパスから移動した場合）、ソースファイルブラウザダイアログボックスが表示されますので、手動でソースファイルを指定してください。

3.2.1 プログラムをダウンロードする

デバッグするロードモジュールをダウンロードします。

プログラムのダウンロードは[デバッグ -> ダウンロード]からロードモジュールを選択するか、[ワークスペース]ウィンドウの[Download modules]のロードモジュールを右クリックすると表示されるポップアップメニューより[ダウンロード]を選択します。

【注】 プログラムをダウンロードする場合、あらかじめロードモジュールとして High-performance Embedded Workshop に登録されている必要があります。

3.2.2 ソースコードを表示する

[ワークスペース]ウィンドウ上のソースファイルをダブルクリックするか、ソースファイル上でマウスの右ボタンをクリックしてポップアップメニューを表示して[開く]を選択すると、[エディタ]ウィンドウが表示されます。

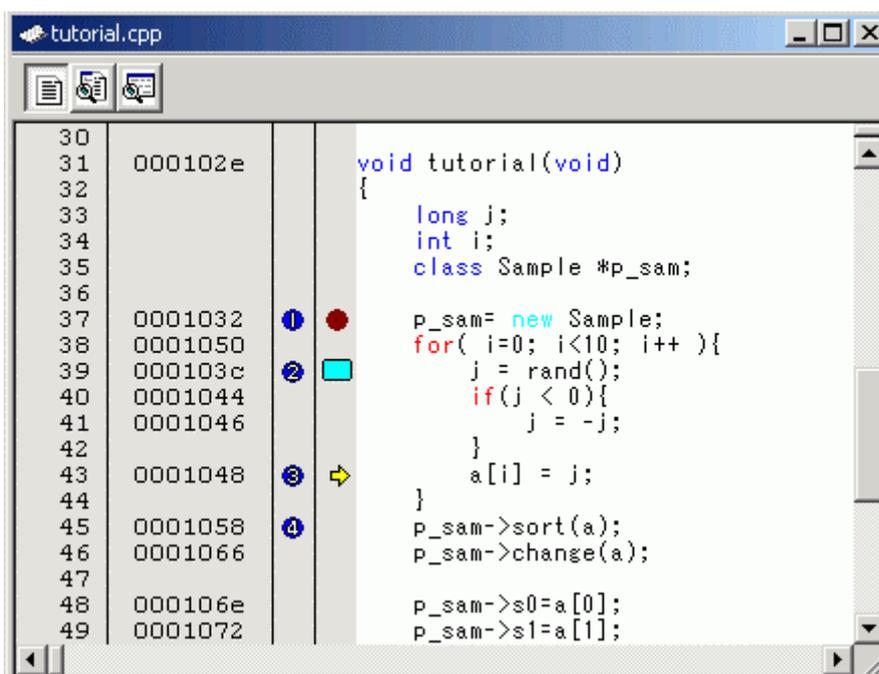


図3.4 [エディタ]ウィンドウ

本ウィンドウでは左端に行情報として下記を表示します。

1 列目	(行番号カラム)	ソースファイルに対応する行番号
2 列目	(ソースアドレスカラム)	ソース行に対応するアドレス情報
3 列目	(オンチップブレイクカラム)	オンチップブレイク
4 列目	(S/W ブレイクポイントカラム)	PC、ブックマーク、ブレイクポイント情報

右側のソース表示画面をテキスト領域と呼びます。

3 デバッグ

行番号カラム

ソースファイルに対応する行番号を表示します。

ソースアドレスカラム

プログラムをダウンロードすると、ソースアドレスカラムに現在のソースファイルに対するアドレスを表示します。本機能は PC 値やブレークポイントをどこに設定するかを決めるときに便利です。

オンチップブレークカラム

オンチップブレークカラムには下記を表示します。

-  オンチップブレークのチャンネル 1 ~ 12
-  リセットポイント

これらの設定は、ポップアップメニューからも可能です。

S/W ブレークポイントカラム

S/W ブレークポイントカラムには下記を表示します。

-  ブックマークを設定している
-  ソフトウェアブレークを設定している
-  PC 位置

⇒すべてのソースファイルでカラムをオフにするには

1. [エディタ]ウィンドウを右クリックしてください。または、[編集]メニューを選択してください。
2. [表示カラムの設定...]メニュー項目をクリックしてください。
3. [エディタ全体のカラム状態]ダイアログボックスを表示します。
4. チェックボックスは、そのカラムが有効が無効かを示します。チェックしている場合は有効です。チェックボックスがグレー表示の場合、一部のファイルではカラムが有効で、別のファイルでは無効であることを意味します。オフにしたいカラムのチェックボックスからチェックを外してください。
5. [OK]ボタンをクリックして、新しいカラム設定を有効にしてください。

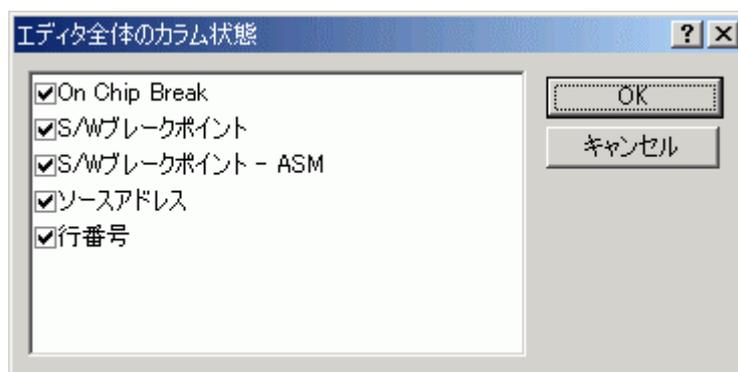


図3.5 エディタ全体のカラム状態ダイアログボックス

3.2.3 アセンブリ言語コードを表示する

ソースファイルが開いている場合、[エディタ]ウィンドウ上でマウスの右ボタンをクリックしてポップアップメニューから[逆アセンブリ表示]を選択すると、[逆アセンブリ]ウィンドウを表示します。[逆アセンブリ]ウィンドウの表示開始アドレスは、[エディタ]ウィンドウのカーソル位置に対応するアドレスとなります。

また、[エディタ]ウィンドウの[逆アセンブリ表示]ボタンを使用して、逆アセンブリコードを表示することもできます。

ソースファイルが存在しない場合、次のいずれかの方法でアセンブリ言語レベルでコードを表示できます。

- [逆アセンブリ]ツールバーボタン  をクリックする
- [表示 -> 逆アセンブリ...]を選択する
- “Ctrl + D” アクセラレータを使用する

この場合、[逆アセンブリ]ウィンドウは現在の PC の位置で開きます。アドレス、コード(オプション)、逆アセンブルされたニーモニック(可能なときはラベル付きで)を表示します。

オプションで、そのアドレスから開始するソースラインすべてを表示させる混合モード表示もサポートします。混合モードの逆アセンブルを表示するには、[混合モードで表示]ボタンをクリックしてください。

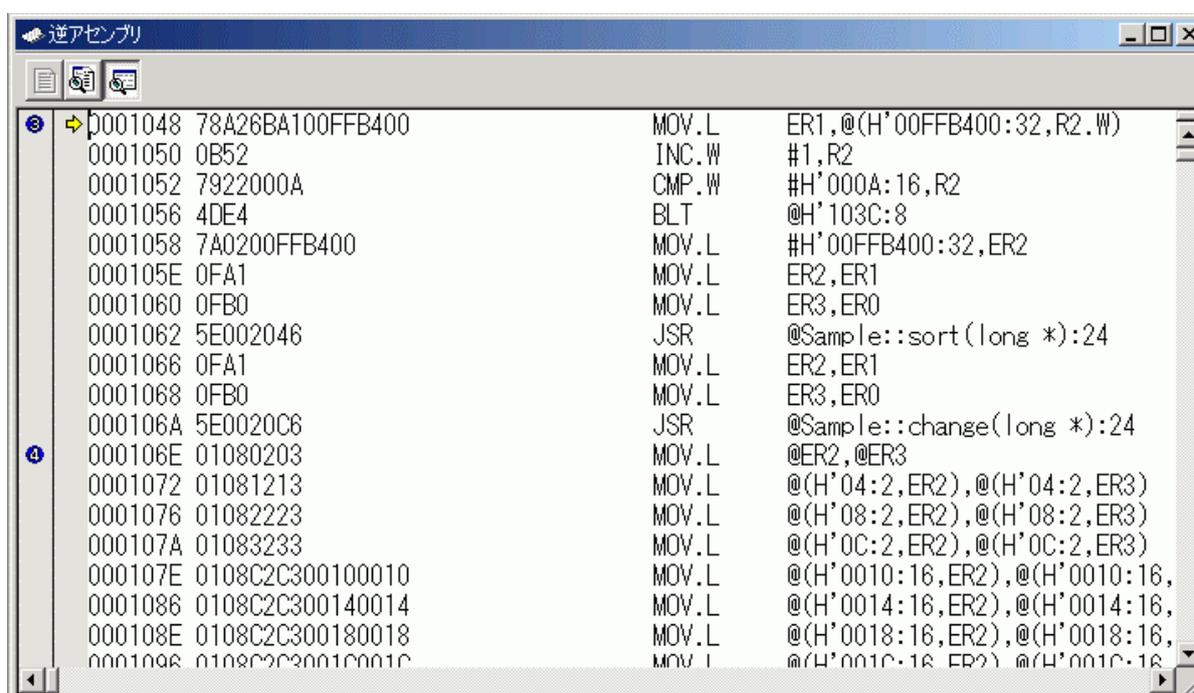


図3.6 逆アセンブリウィンドウ

本ウィンドウでは左はじに行情報として下記を表示します。

- | | | |
|-----|------------------------|---------------|
| 1行目 | (オンチップブレイクカラム) | オンチップブレイク |
| 2行目 | (S/W ブレイクポイント-ASM カラム) | PC、ブレイクポイント情報 |

使用方法はソースコードの表示ウィンドウと同じです。

3 デバッグ

3.2.4 アセンブリ言語コードを修正する

[逆アセンブリ]ウィンドウで修正したい命令をダブルクリックするか、ポップアップメニューから[編集...]を選択すると、[アセンブル]ダイアログボックスが開き、アセンブリ言語コードを修正することができます。



図3-7 アセンブルダイアログボックス

アドレス、命令コード、およびニーモニックを表示します。

[ニーモニック]フィールドに新しい命令を入力（または古い命令を編集）します。"Enter"キーを押すと、メモリ内容を新しい命令コードに書き換えて、次の命令に移ります。

[OK]ボタンをクリックすると、メモリ内容を新しい命令コードに書き換えて、ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックするか"Esc"キーを押すと、メモリ内容を書き換えずに、ダイアログボックスを閉じます。

【注】 アセンブリ言語コードは、現在のメモリ内容から表示してします。メモリの内容を修正すると、[逆アセンブリ]ウィンドウおよび[アセンブル]ダイアログボックスでは、新しいアセンブリ言語コードを表示します。しかし、[エディタ]ウィンドウに表示しているソースファイルは変更しません。これはソースファイルにアセンブラを含む場合も同じです。

3.2.5 特定のアドレスを見る

[逆アセンブリ]ウィンドウを使って作成したプログラムを表示している場合、プログラム内の他の箇所を見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスにジャンプすることができます。

[逆アセンブリ]ウィンドウでアドレスをダブルクリックするか、ポップアップメニューから[表示アドレス設定...]を選択すると、[アドレス指定]ダイアログボックスを表示します。



図3-8 アドレス設定ダイアログボックス

[アドレス]エディットボックスにアドレスを指定し、[OK]ボタンをクリックしてください。アドレスは、ラベル名で入力することも可能です。

[逆アセンブリ]ウィンドウを更新して入力したアドレスコードを表示します。

オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスを開くので、関数を選択してください。

3.2.6 現在のプログラムカウンタアドレスを見る

アドレスまたは値を入力できる場所では、式も入力することができます。先頭に"#"文字をつけたレジスタ名を入力すると、そのレジスタ内容を式の値として使用します。

[アドレス指定]ダイアログボックスで"#PC"という式を入力すると、[エディタ]または[逆アセンブリ]ウィンドウには、現在のPCアドレスを表示します。例えば、"#PC+0x100"といったPCレジスタおよびオフセットの式を入力することにより現在のPCのオフセットを表示することができます。

3.3 現在の状態を表示する

デバッグプラットフォームの現在の状態を知るには[ステイタス]ウィンドウを表示します。

[ステイタス]ウィンドウを開くには、[表示 -> CPU -> ステイタス]を選択するか、[ステイタスの表示]ツールバーボタン  をクリックします。

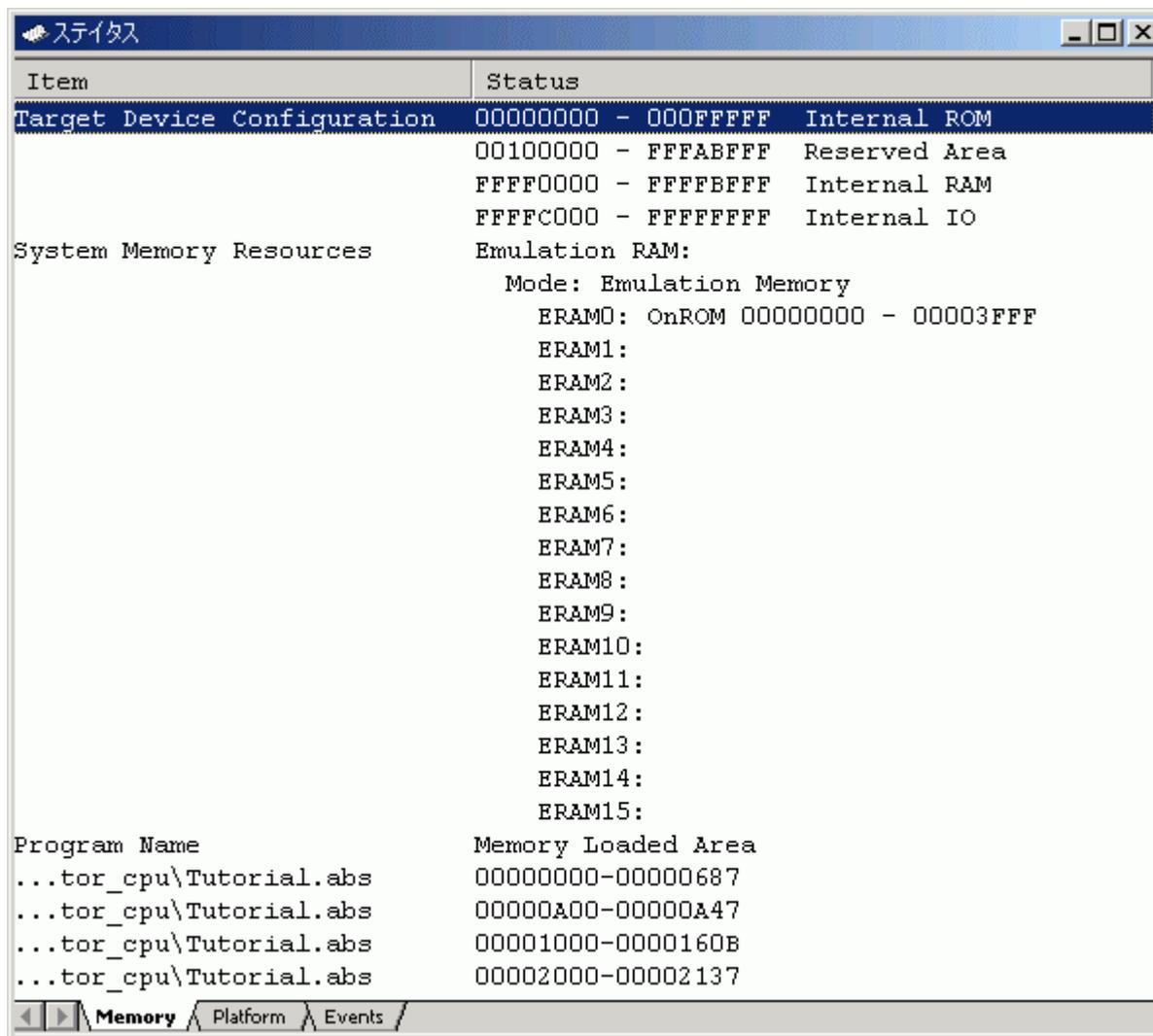


図3.9 ステイタスウィンドウ

[ステイタス]ウィンドウには、以下の3枚のシートがあります。

- [Memory]シート
メモリマッピングおよび現在ロードしたオブジェクトファイルが使用するメモリエリアなど、現在のメモリステータスに関する情報を表示します。
- [Platform]シート
CPU種別および動作モードなど、エミュレーション環境情報を表示します。
- [Events]シート
リソース情報およびブレークポイント等のイベント情報に関する情報を表示します。

【注】 本ウィンドウに表示する項目はご使用のエミュレータにより異なります。

3 デバッグ

詳細につきましては、オンラインヘルプをご参照ください。

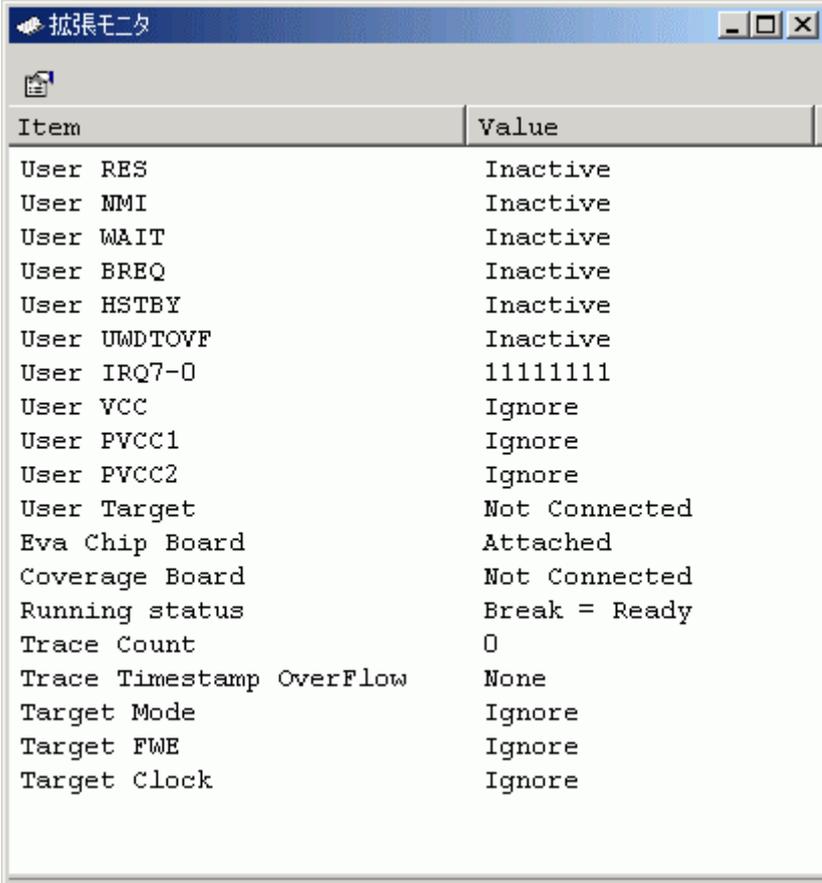
3.4 エミュレータの情報を定期的に取り出し表示する

ユーザプログラム実行中 / 停止中にかかわらず変化するエミュレータの情報を知るには[拡張モニタ]ウィンドウを使用します。

【注】 拡張モニタ機能は E6000H エミュレータのハードウェア回路によりユーザシステムやエミュレータ内部のターゲットマイコンから出力される信号をモニタするため、ユーザプログラムの実行に影響を与えることはありません。

3.4.1 [拡張モニタ]ウィンドウを開く

[拡張モニタ]ウィンドウを開くには、[表示 -> CPU -> 拡張モニタ]を選択するか、[拡張モニタ]ツールバーボタンをクリックします。表示項目の更新間隔は、ユーザプログラム実行中は約 1000ms、ブレーク中は約 5000ms です。



Item	Value
User RES	Inactive
User NMI	Inactive
User WAIT	Inactive
User BREQ	Inactive
User HSTBY	Inactive
User UWDTOVF	Inactive
User IRQ7-0	11111111
User VCC	Ignore
User PVCC1	Ignore
User PVCC2	Ignore
User Target	Not Connected
Eva Chip Board	Attached
Coverage Board	Not Connected
Running status	Break = Ready
Trace Count	0
Trace Timestamp OverFlow	None
Target Mode	Ignore
Target FWE	Ignore
Target Clock	Ignore

図3.10 拡張モニタウィンドウ

3.4.2 表示項目を選択する

ポップアップメニューから[プロパティ...]を選択すると[拡張モニタコンフィギュレーション]ダイアログボックスを表示します。

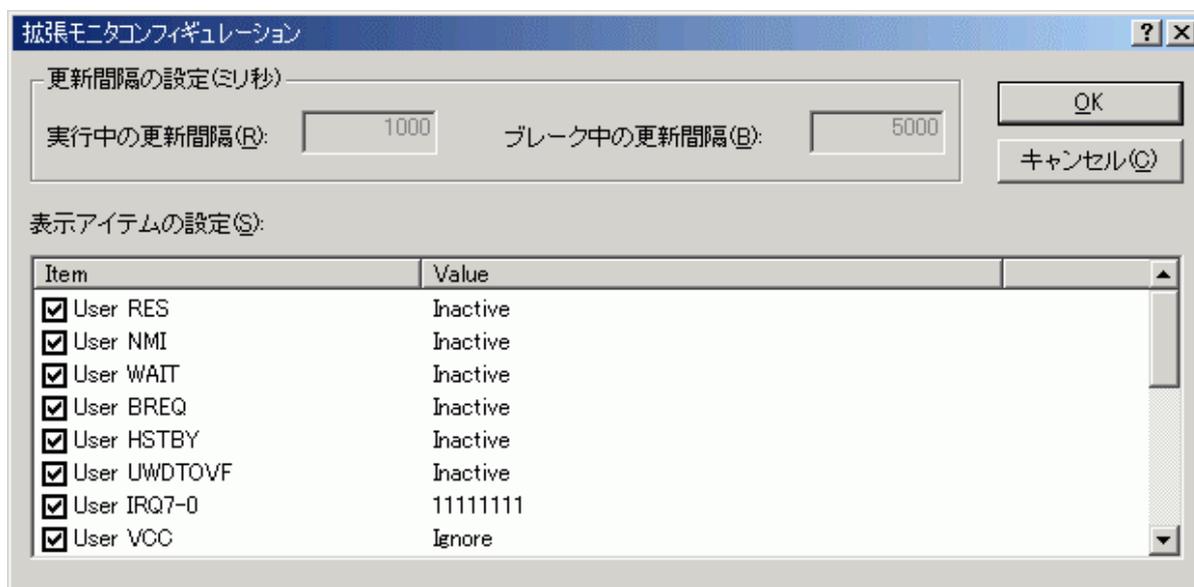


図3.11 拡張モニタコンフィギュレーションダイアログボックス

[拡張モニタ]ウィンドウに表示する各項を設定できます。

【注】 本ウィンドウに表示する項目はご使用のエミュレータにより異なります。
詳細につきましては、オンラインヘルプをご参照ください。

3.5 リアルタイムにメモリ内容を表示する

ユーザプログラム実行中にメモリ内容をモニタするには[モニタ]ウィンドウを使用します。

モニタ機能はE6000Hエミュレータのバスモニタ回路によりターゲットマイコン内部のリード/ライト信号をトリガとしてアドレスバスおよびデータバスの値を保持し、該当するメモリの表示内容を更新するためリアルタイム性は損なわれません。

E6000Hエミュレータのバスモニタ回路に実装されたモニタチャンネル(8チャンネル)を使用し、最大8ポイントまたは合計256バイト設定できます。

各ポイントのモニタ範囲の一部または全部が重複する設定も可能です

【注】

1. ターゲットマイコンに内蔵のタイマカウンタなど、値の更新にターゲットマイコン内部のライト信号が発生しないエリアに対してはモニタできません。
2. メモリ内容の表示および変更方法は製品により異なります。また、ユーザプログラム実行中にメモリ内容の表示を更新するとリアルタイム性がなくなる場合があります。詳しくは「5.3 メモリ内容の表示と変更方法について」を参照してください。

3 デバッグ

3.5.1 モニタウィンドウを開く

[モニタ]ウィンドウを開くには、[表示 -> CPU -> モニタ -> モニタ設定...]を選択するか、[モニタ]ツールバーボタンをクリックして[Monitor Setting]ダイアログボックスを開きます。

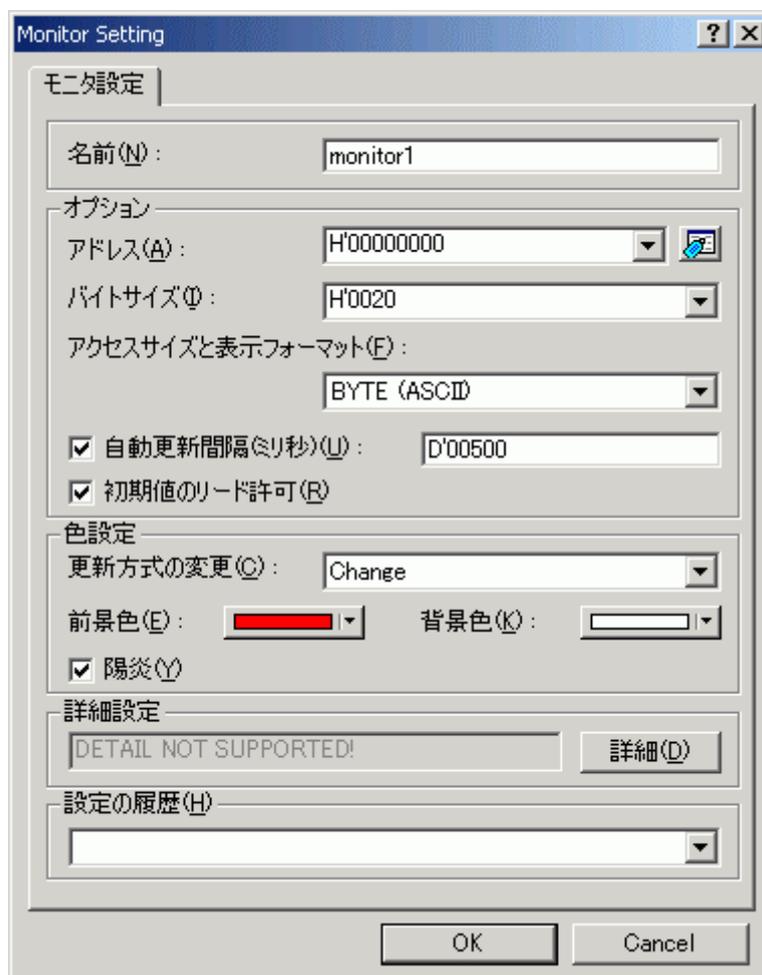


図3.12 Monitor Setting ダイアログボックス

[名前]	モニタウィンドウの名称を設定します。
[オプション]	モニタ条件を設定します。
[アドレス]	モニタを行う先頭アドレスを設定します。
[バイトサイズ]	モニタを行う範囲を設定します。
[アクセスサイズと表示 フォーマット]	モニタウィンドウに表示するアクセスサイズを設定します。
[自動更新間隔(ミリ秒)]	モニタ取得間隔を設定します。(最小値は500ms)
[初期値のリード許可]	モニタウィンドウ OPEN 時に、モニタ表示エリアの値をリードします。
[色設定]	モニタの更新方法および色属性を設定します
[更新方法の変更]	モニタ中に変更があった値をどのように表示するかを設定します。 ([初期値のリード許可]選択時有効)
	No 色の変更は行いません。
	change:
	Change: 色を変更します。 色は前景色オプション、背景色オプション、で設定します。
	Gray: 値の変更のないデータを灰色表示します。
	Appear: 値の変更があると表示します。変更なければ表示しません。
[前景色]	表示文字色を設定します。 ([Change]選択時有効)
[背景色]	背景色を設定します。 ([Change]選択時有効)
[陽炎]	チェックボックスにチェックがある場合、一定間隔更新のないデータの色を背景色オプションで設定した色に戻します。一定間隔とは、モニタ取得間隔の1回分です。 ([Change]、[Gray]、[Appear]選択時有効)
[詳細設定]	エミュレータ固有の項目を設定します。
[設定の履歴]	前回の設定内容呼び出します。

- 【注】**
1. 本エミュレータではモニタを行う先頭アドレスとして奇数アドレスは指定できません。
 2. 前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

設定完了後、[OK]ボタンをクリックすると[モニタ]ウィンドウが開きます。

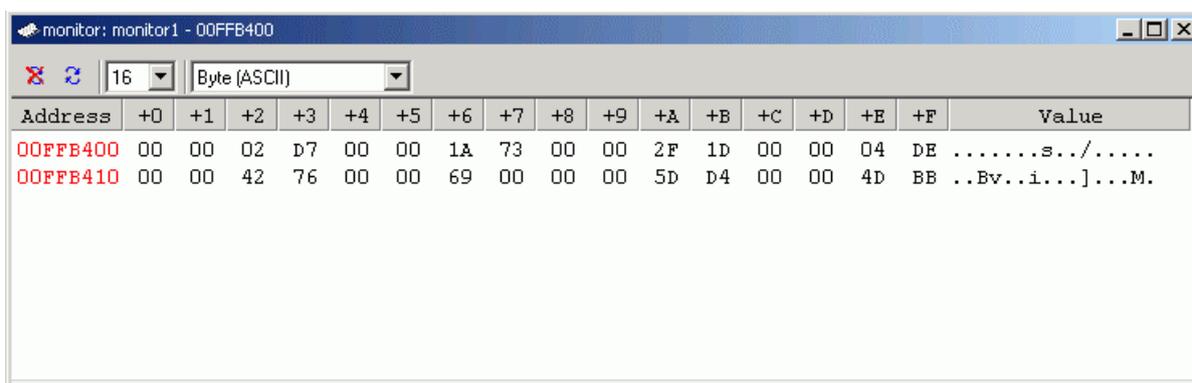


図3.13 モニタウィンドウ

ユーザプログラム実行中、自動更新間隔の設定値に応じて表示を更新します。

- 【注】** アドレス変更時またはメモリ内容変更時、データ内容が正しく表示されない場合は、ポップアップメニューより[最新の情報に更新]を選択してください。

3 デバッグ

3.5.2 モニタの設定内容を変更する

変更したい[モニタ]ウィンドウのポップアップメニューより[モニタ設定...]を選択すると、[Monitor Setting]ダイアログボックスが開き、設定内容を変更することができます。

また、ポップアップメニューの[色設定]メニューおよび[アクセスサイズと表示フォーマット]メニューより簡単に色設定およびアクセスサイズと表示フォーマットを変更できます。

3.5.3 モニタの更新を一時的に停止する

ユーザプログラム実行中、[モニタ]ウィンドウは設定した自動更新間隔にしたがって自動的に表示を更新します。

表示更新を停止させたい[モニタ]ウィンドウのポップアップメニューより[表示固定]を選択してください。

アドレス部の表示文字が黒色となり、表示更新を停止します。

再びポップアップメニューより[表示固定]を選択することにより停止状態は解除できます。

3.5.4 モニタ設定を削除する

削除したい[モニタ]ウィンドウのポップアップメニューより[閉じる]を選択すると、[モニタ]ウィンドウを閉じ、モニタ設定を削除します。

3.5.5 変数の内容をモニタする

任意の変数の値を参照するには、[ウォッチ]ウィンドウを使用します。

[ウォッチ]ウィンドウに登録した変数のアドレスが、モニタ機能で設定したモニタ範囲に存在する場合、該当する変数の値をモニタ機能により更新し表示することができます。

この機能によりリアルタイム性を損なわずに変数の内容を確認できます。

3.5.6 モニタウィンドウを非表示にする

モニタ機能を使用し、[ウォッチ]ウィンドウより変数の値をモニタする場合、[モニタ]ウィンドウを非表示にしておく画面を有効に活用できます。

現在設定しているモニタ情報は[表示 -> CPU -> モニタ]のサブメニューとしてリストされます。

モニタ設定リストは[モニタ]ウィンドウ名およびモニタ開始アドレスで構成されています。

リストの左側にチェックがある場合は該当の[モニタ]ウィンドウが表示されていることを示します。

モニタ設定リストより非表示にしたい[モニタ]ウィンドウ項目を選択すると、該当の[モニタ]ウィンドウが非表示となり、リストの左側にあったチェックマークが消えます。

非表示にした[モニタ]ウィンドウを再び表示するにはモニタ設定リストより非表示にした[モニタ]ウィンドウ項目を選択してください。

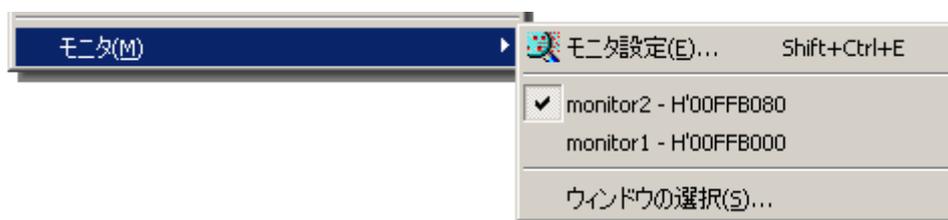


図3.14 モニタ設定リスト

3.5.7 モニタウィンドウを管理する

[表示 -> CPU -> モニタ -> ウィンドウの選択...]を選択すると表示される、[ウィンドウの選択]ダイアログボックスより、現在設定されているモニタ条件の確認、新規モニタ条件の追加、編集、削除などの操作を連続的に行うことができます。

また、現在設定されているモニタ条件を複数選択することにより、更新の一時停止、非表示、削除を一括して操作できます。

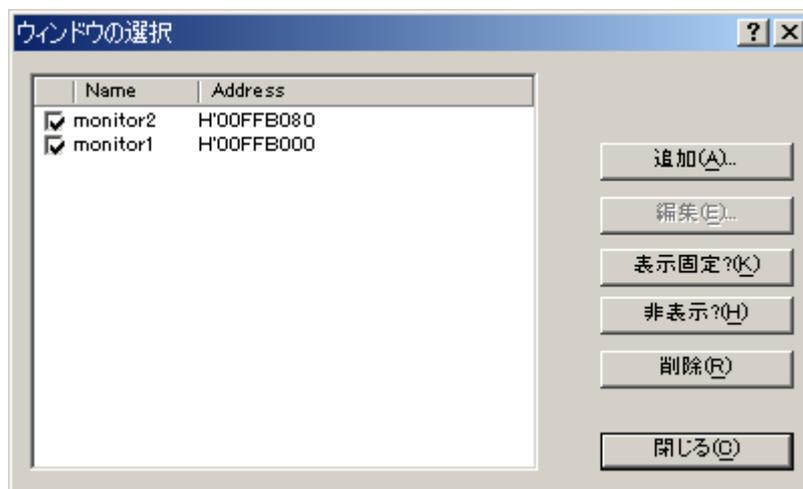


図3.15 ウィンドウの選択ダイアログボックス

[追加]	新規にモニタ条件を追加します。
[編集]	選択している[モニタ]ウィンドウの設定を変更します。 (複数選択時無効)
[表示固定 / 表示固定解除]	選択している[モニタ]ウィンドウの表示を自動更新または更新停止にします。
[表示 / 非表示]	選択している[モニタ]ウィンドウを表示または非表示にします。
[削除]	選択しているモニタ条件を削除します。
[閉じる]	本ダイアログボックスを閉じます。

3.6 変数の表示

本節では、ソースプログラム上の変数の値を表示する方法について説明します。

3.6.1 ウォッチウィンドウ

[ウォッチ]ウィンドウを開くことにより、任意の変数について値を参照することができます。

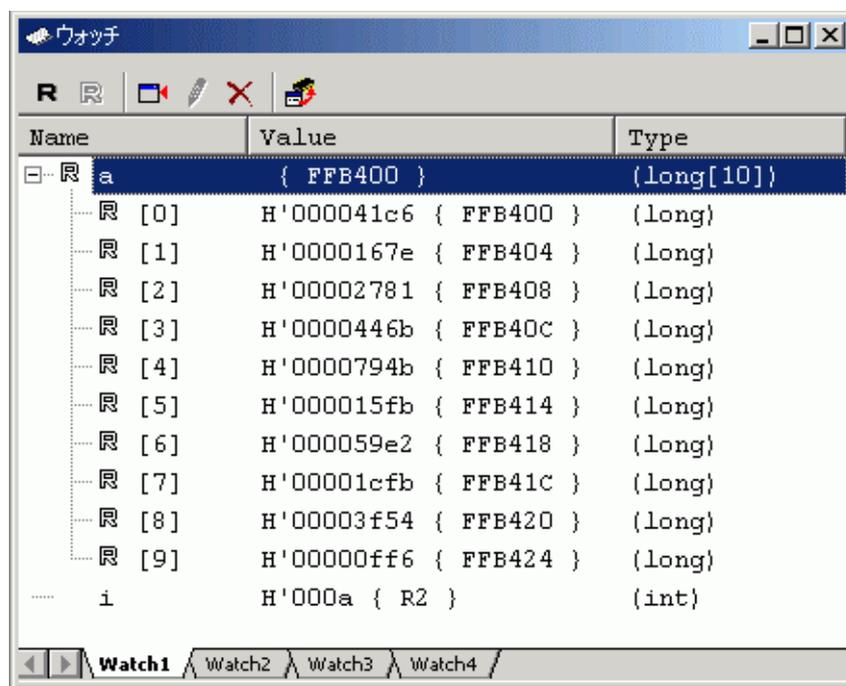


図3.16 ウォッチウィンドウ

R マークのある変数は、ユーザプログラム実行中に更新できることを示します。

R マークの色によって、ユーザプログラム実行中における値更新の実現手段が判別できます。

中抜き青色部	変数のアドレスがモニタ機能で設定されているモニタ範囲内で、モニタ機能を使用したデータリードが可能であることを示します。
青色部	モニタ機能を使用したデータリードによる値の更新を行うことを示します。
中抜き黒色部	変数のアドレスがモニタ機能で設定されているモニタ範囲外で、モニタ機能を使用したデータリードが不可能であることを示します。
黒色部	通常のデータリードにより値の更新を行うことを示します。

- 【注】
1. 本機能は変数ごと、また構造体であれば指定された構造体一括/要素ごとに設定できます。
 2. R マークの色は、モニタ設定を変更したときに変わります。
 3. レジスタに割り付けられている変数には設定することができません。
 4. メモリ内容の表示および変更方法は製品により異なります。また、ユーザプログラム実行中にメモリ内容の表示を更新するとリアルタイム性がなくなる場合があります。詳しくは「5.3 メモリ内容の表示と変更方法について」を参照してください。
 5. SH7059 E6000H エミュレータでは内蔵 RAM のみをリアルタイムにメモリ内容を表示する、256 バイト×12 ポイントのウォッチ機能専用内蔵 RAM モニタを搭載しています。変数登録後に自動更新有効化する事で、ウォッチ機能専用内蔵 RAM モニタを自動設定し、変数の自動更新無効化及び変数の削除でウォッチ機能専用内蔵 RAM モニタを解除します。また、12 ポイントのウォッチ機能専用内蔵 RAM モニタがすべて使われている時に自動更新有効化した場合は、通常のデータリードにより値を更新します。内蔵 RAM 以外をリアルタイムでメモリ内容を表示する場合は通常のモニタをご使用ください。

3.7 イベントポイントを使用する

E6000H エミュレータでは、イベントポイントとして以下の3つのブレークをサポートしています。

ソフトウェアブレーク ポイント	指定アドレスの命令フェッチが行われた場合にユーザプログラムの実行を停止します。最大 255 ポイントまで設定できます。 指定アドレスを専用命令で置き換えるため、ユーザシステム上の ROM またはフラッシュメモリなどの書き換え不可能な領域には設定できません。 [エディタ]ウィンドウおよび[逆アセンブリ]ウィンドウから設定することもできます。
オンチップブレーク ポイント	チップ内蔵のブレーク機能です。 アドレスバス条件、データバス条件、Bus/Area 条件、成立回数条件を設定できます。 ソフトウェアブレークが設定できないユーザシステム上の ROM またはフラッシュメモリなどに設定することもできます。 複数のオンチップブレークポイントを組み合わせさせたシーケンシャルブレークを設定することができます。 [エディタ]ウィンドウおよび[逆アセンブリ]ウィンドウから設定することもできます。
オンエミュレータブレーク ポイント	E6000H エミュレータハードで実現しているブレーク機能です。 アドレスバス条件、データバス条件、Bus/Area 条件、外部プローブ信号条件、外部割り込み信号条件、成立回数条件を設定できます。 E6000H エミュレータハードで実現しているため、条件成立からブレークが発生するまで数サイクルかかることがあります。

ソフトウェアブレークポイント、オンチップブレークポイントおよびオンエミュレータブレークポイントは[イベントポイント]ウィンドウから設定することができます。

[イベントポイント]ウィンドウを開くには、[表示 -> コード -> イベントポイント]を選択するか、[イベントポイント]ツールバーボタンをクリックします。

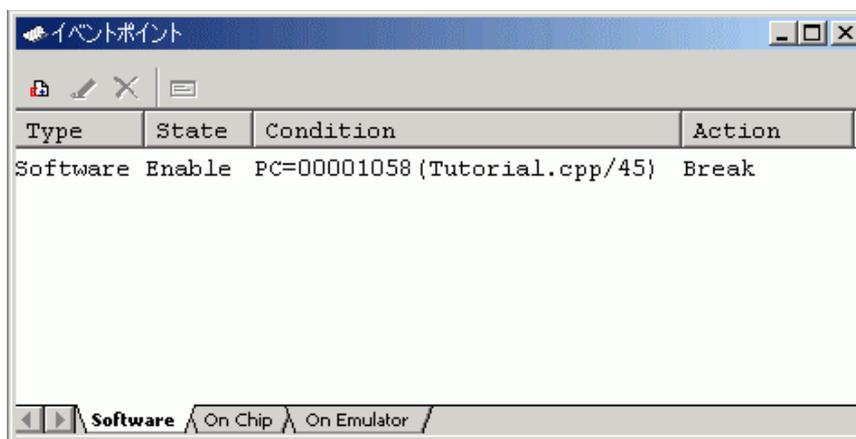


図3.17 イベントポイントウィンドウ

[イベントポイント]ウィンドウには、3枚のシートがあります。

- [Software]シート
ソフトウェアブレークポイントの設定内容を表示します。また、ソフトウェアブレークポイントの設定、変更および解除を行うことができます。
- [On Chip]シート
オンチップブレークポイントを表示、設定します。
- [On Emulator]シート
オンエミュレータブレークポイントを表示、設定します。

3 デバッグ

【注】 イベントポイントについての注意事項は「5.5 イベント機能について」を参照してください。

3.7.1 ソフトウェアブレークポイントを設定する

[イベントポイント]ウィンドウ - [Software]シートではソフトウェアブレークポイントの設定内容の表示、変更および追加ができます。

[Software]シート上のポップアップメニューから[追加...]または[編集...]を選択すると[ブレークポイントプロパティ]ダイアログボックス - [ソフトウェアブレーク]ページを表示します。

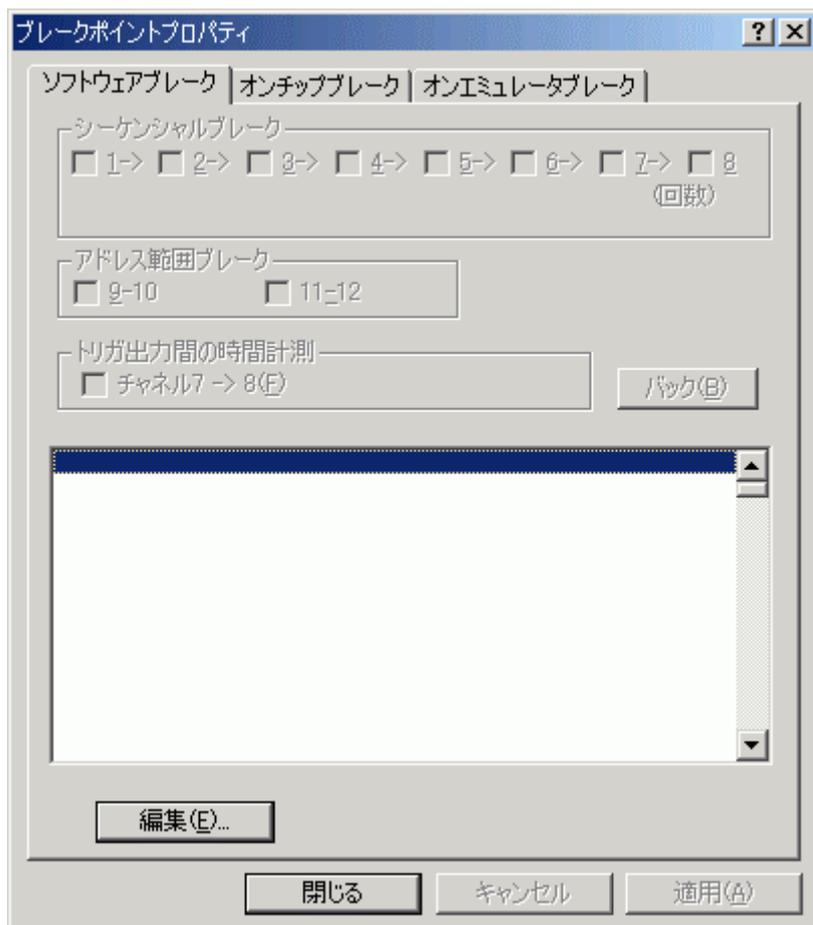


図3.18 ブレークポイントプロパティダイアログボックス (ソフトウェアブレークページ)

新しいソフトウェアブレイクポイントを追加する場合は、[ソフトウェアブレイク]ページのリストボックスから空白行を選択し、[編集...]ボタンをクリックしてください。既存の設定を変更する場合は、変更するソフトウェアブレイクポイントをリストボックスから選択し、[編集...]ボタンをクリックしてください。

ソフトウェアブレイクポイントを設定するための、[ソフトウェアブレイク]ダイアログボックスを表示します。

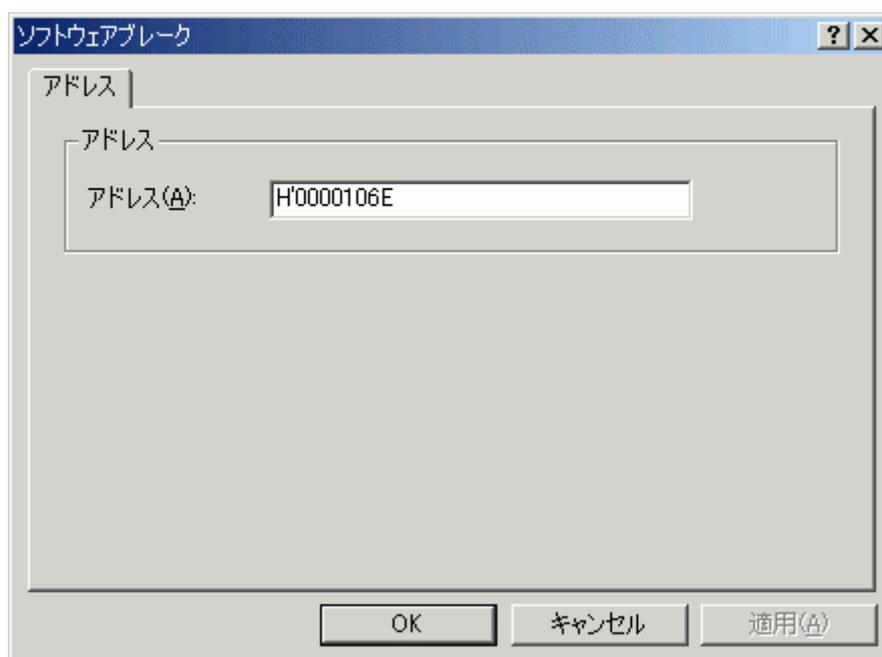


図3.19 ソフトウェアブレイクダイアログボックス

[アドレス]エディットボックスに設定するアドレスを指定し、[OK]ボタンを押してください。

3.7.2 オンチップブレークポイントを設定する

[イベントポイント]ウィンドウ - [On Chip]シートではオンチップブレークポイントの設定内容の表示、変更ができます。

[On Chip]シート上のポップアップメニューから[追加...]または[編集...]を選択すると[ブレークポイントプロパティ]ダイアログボックス - [オンチップブレーク] ページを表示します。



図3.20 ブレークポイントプロパティダイアログボックス (オンチップブレークページ)

[シーケンシャルブレーク]	チャンネル1~8を使用するシーケンシャルブレークの指定をします。チャンネルは、連続した順次で指定されます。チャンネル1~8、5~8以外を使用する場合、未使用の UNUSED で表示されたチャンネルは使用できません。
[アドレス範囲ブレーク]	チャンネルの組合せを指定し、ブレーク(範囲)の停止順を指定します。チャンネルの組合せは、以下を選択し指定します。 9-10 :チャンネル 9,10 を使用しブレークを指定します。 11-12 :チャンネル 11,12 を使用しブレークを指定します。
[トリガ出力間の時間計測]	2点間の時間計測をします。チャンネル7、8を使用し時間計測を行います。チャンネル7の成立後、チャンネル8が成立で時間が測定されます。測定結果は [ステータス] ウィンドウ - [Platform] シートの [RunTime Count] に表示されます。
[バック]	本ボタンをクリックすることにより、本ダイアログで設定した内容を、本ダイアログを表示した時点に戻します。
リストボックス	現在の設定状況をチャンネルごとに表示します。チャンネル番号の R は、シーケンシャルブレークのリセットポイントの設定になります。 設定のないチャンネルはチャンネル番号のみ表示します。 シーケンシャルブレークとして使用している場合は、チャンネル番号の横に S を表示します。
[編集...]	チャンネル指定後、本ボタンをクリックすることにより、指定したチャンネルに対するブレーク条件の設定を行う [オンチップブレーク チャンネル n] ダイアログボックスが表示されます。 (n : チャンネル番号)
[リセット]	本ボタンをクリックすることにより、選択しているチャンネルの設定を解除します。

[すべてリセット] すべてのチャンネルの設定を解除します。

[オンチップブ레이크 チャンネル n]ダイアログボックスでは、[アドレス]ページ、[データ]ページ、[バス/エリア]ページ、[回数]ページおよび[アクション]ページの各条件を組み合わせることで複雑なブ레이크条件を設定することができます。

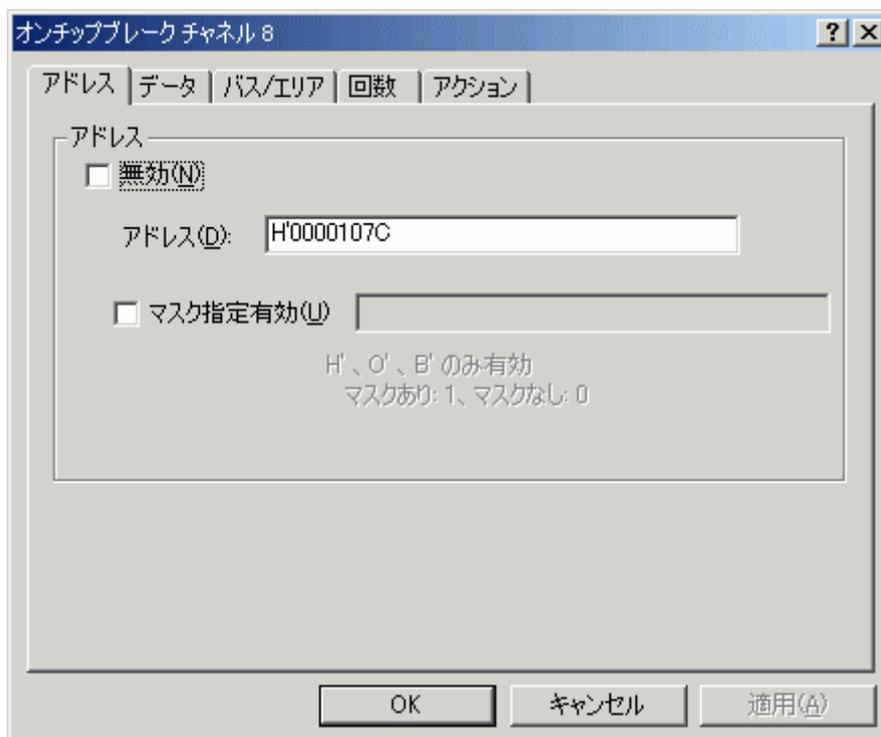


図3.21 [オンチップブ레이크 チャンネル n] ダイアログボックス

[アドレス]	アドレスバス条件を設定します。
[無効]	アドレスバス条件を無効にします。
[アドレス]	アドレスバス値を指定します。
[マスク指定有効]	マスク条件を指定します。[マスク指定有効]を選択した場合に、マスクする値を設定します。マスクを行ったビットは、どんな値でも条件が成立することになります。
[データ]	データバス条件を設定します。
[無効]	データバス条件を無効にします。
[データ値]	データバス条件の値を設定します。
[マスク指定有効]	マスク条件を指定します。[マスク指定有効]を選択した場合に、マスクする値を設定します。マスクを行ったビットは、どんな値でも条件が成立することになります。
[アクセスサイズ]	データアクセスサイズを指定します。
[バス/エリア]	アクセスタイプ、バスステータスおよびリード/ライトサイクル条件を設定します。
[アクセスタイプ]	アクセスタイプ条件を設定します。
[バスステータス]	バスステータス条件を設定します。[無効]を選択した場合は、バスステータス条件を無効にします。
[リード/ライト]	リード/ライト条件を指定します。[無効]を選択した場合は、リード/ライト条件を無効にします。
[回数]	条件成立回数を設定します。[無効]を選択した場合は、条件成立回数は1になります。
[アクション]	
[ブ레이크]	設定条件成立時に実行停止します。
[実行前]	条件が成立したアドレスを実行後に停止します。
[実行後]	条件が成立するアドレスを実行前に停止します。
[トリガ出力]	設定条件成立時にトリガ出力を行います。

3 デバッグ

3.7.3 オンエミュレータブレークポイントを設定する

[イベントポイント]ウィンドウ - [On Emulator]シートではオンエミュレータブレークポイントの設定内容の表示、変更ができます。

[On Emulator]シート上のポップアップメニューから[追加...]または[編集...]を選択すると[ブレークポイントプロパティ]ダイアログボックス - [オンエミュレータブレーク]ページを表示します。

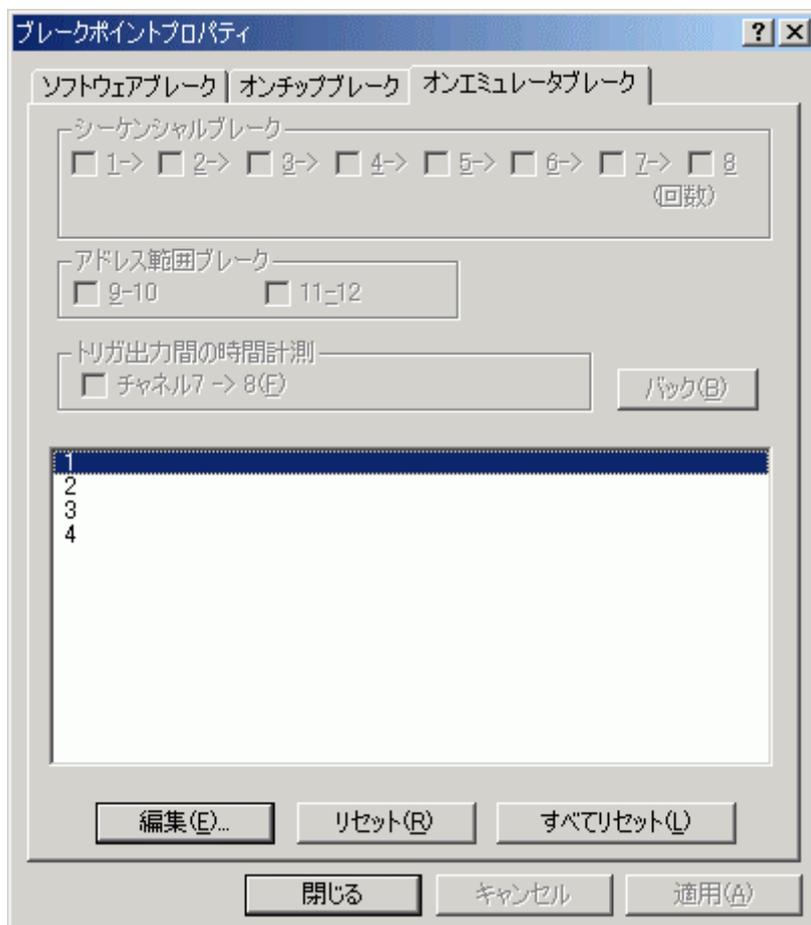


図3.22 ブレークポイントプロパティダイアログボックス (オンエミュレータブレークページ)

リストボックス	現在の設定設定状況をチャンネルごとに表示します。 設定のないチャンネルはチャンネル番号のみ表示します。
[編集...]	チャンネル指定後、本ボタンをクリックすることにより、指定したチャンネルに対するブレーク条件の設定を行う[オンエミュレータブレーク チャンネル n]ダイアログボックスが表示されます。 (n: チャンネル番号)
[リセット]	本ボタンをクリックすることにより、選択しているチャンネルの設定を解除します。
[すべてリセット]	すべてのチャンネルの設定を解除します。

[オンエミュレータブレーク チャンネル n]ダイアログボックスでは、[アドレス]ページ、[データ]ページ、[バス/エリア]ページ、[プローブ]ページ、[割り込み]ページおよび[回数]ページの各条件を組み合わせることで複雑なブレーク条件を設定することができます。

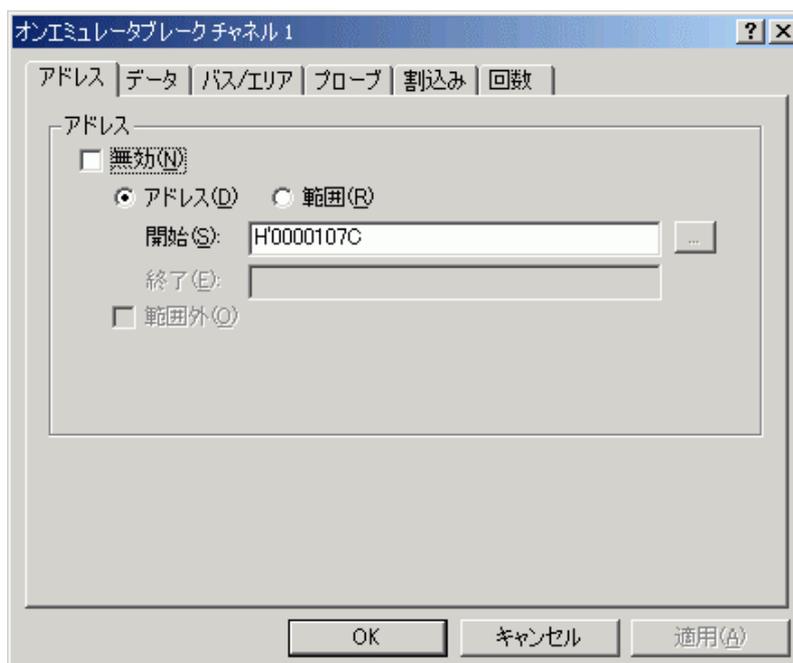


図3.23 [オンエミュレータブレーク チャンネル n] ダイアログボックス

[アドレス]	アドレスバス条件を設定します。
[無効]	アドレスバス条件を無効にします。
[アドレス]	[開始]で指定したアドレスバスの値をブレーク条件にする場合に選択します。
[範囲]	[開始] (開始アドレス) ~ [終了] (終了アドレス) で指定したアドレスバスの値の範囲でブレークします。
[範囲外]	[範囲]で設定した範囲以外のアドレスバスでブレークさせる場合に選択します。
[...]	開数のアドレス範囲を[開始]と[終了]に設定することができます。詳細は「5.10 入力形式について」を参照してください。
[データ]	データバス条件を設定します。
[無効]	データバス条件を無効にします。
[データ値]	データバス条件の値を設定します。
[マスク指定有効]	マスク条件を指定します。 [マスク指定有効] を選択した場合に、マスクする値を設定します。マスクを行ったビットは、どんな値でも条件が成立することになります。
[この値以外]	設定した値以外をデータバス条件とする場合に選択します。
[アクセスサイズ]	データアクセスサイズを指定します。
[位置]	データバスの値を数値で設定します。有効なデータバスの位置を指定します。
[ロングワード]	なし
[ワード]	4n は上位ワード、4n+2 は下位ワードになります。
[バイト]	4n は上位ワードの上位バイト、4n+1 は上位ワードの下位バイト、4n+2 は下位ワードの上位バイト、4n+3 は下位ワードの下位バイトになります。
[バス/エリア]	アクセスタイプ、バスステータスおよびリード/ライトサイクル条件を設定します。
[アクセスタイプ]	アクセスタイプ条件を設定します。 [無効] を選択した場合は、アクセスタイプ条件を無効にします。
[バスステータス]	バスステータス条件を設定します。 [無効] を選択した場合は、バスステータス条件を無効にします。
[リード/ライト]	リード/ライト条件を指定します。 [無効] を選択した場合は、リード/ライト条件を無効にします。
[プローブ]	外部プローブ信号 (PRB1 ~ PRB4) の状態 (HIGH/LOW) を条件にします。 [無効] を指定した場合は、指定した外部プローブ信号の状態をブレーク条件としません。
[割り込み]	IRQ 信号および NMI 信号を条件にします。 [無効] を指定した場合は、各信号の状態をブレーク条件としません。
[回数]	条件成立回数を設定します。 [無効] を選択した場合は、条件成立回数は 1 になります。

3.7.4 イベントポイントの編集

ソフトウェアブレークポイント、オンチップブレークポイント、オンエミュレータポイントに対する設定以外の操作方法はすべて共通となっています。

3.7.5 イベントポイントの設定内容を変更する

変更したいイベントポイントを選択後ポップアップメニューから[編集...]を選択すると、各イベントに対応した設定ダイアログボックスが開き、設定内容を変更することができます。[編集...]メニューはイベントポイントを1個選択しているときのみ有効となります。

3.7.6 イベントポイントを有効にする

イベントポイントを選択後ポップアップメニューから[有効]を選択すると、選択しているイベントポイントを有効にします。

3.7.7 イベントポイントを無効にする

イベントポイントを選択後ポップアップメニューから[無効]を選択すると、選択しているイベントポイントを無効にします。無効にした場合は、イベントポイントはリストには残りますが、指定した条件が一致してもイベントは発生しません。

3.7.8 イベントポイントを削除する

イベントポイントを選択後ポップアップメニューから[削除]を選択すると、選択しているイベントポイントを削除します。イベントポイントを削除しないで、詳細情報は保持したまま、条件が成立してもイベントを発生させないようにするには、[無効]オプションを使用します（「3.7.7 イベントポイントを無効にする」参照）。

3.7.9 イベントポイントをすべて削除する

ポップアップメニューから[すべて削除]を選択すると、すべてのイベントポイントを削除します。

3.7.10 イベントポイントのソース行を表示する

イベントポイントを選択後ポップアップメニューから[ソースを表示]を選択すると、ブレークポイントのある[ソース]または[逆アセンブリ]ウィンドウをオープンします。[ソースを表示]メニューは対応するソースファイルをもつイベントポイントを1個選択しているときのみ有効となります。

3.8 トレース情報を見る

E6000H エミュレータでは命令の実行結果をトレース情報としてトレースバッファに取得し、表示することができます。

トレース情報は[トレース]ウィンドウに表示します。トレース情報の取得条件は、[トレース取得プロパティ]ダイアログボックスで設定します。

トレース機能は E6000H エミュレータのハードウェア回路によりバスサイクル単位にトレース情報を取得し、トレースバッファに保持するためリアルタイム性は損なわれません。

リアルタイムトレースバッファは、128k までのバスサイクルを保持でき、バッファがいっぱいになった場合は取得したトレース情報のうち最も古いデータに新しいデータを上書きします。

【注】 トレース機能についての注意事項は「5.6 トレース機能について」を参照してください。

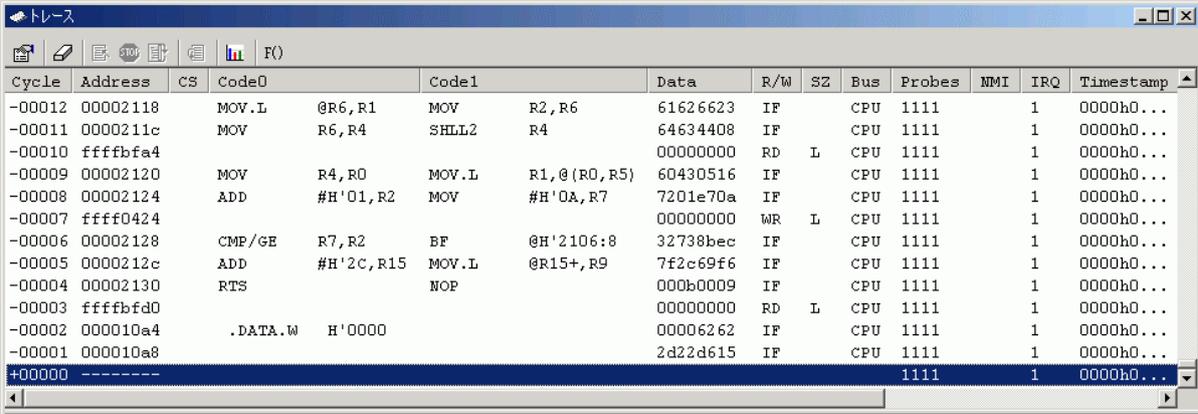
3.8.1 トレースウィンドウを開く

[トレース]ウィンドウを開くには、[表示 -> コード -> トレース]を選択するか、[トレース]ツールバーボタンをクリックします。

3.8.2 トレース情報を取得する

E6000H エミュレータはトレース情報の取得条件を設定しない場合、デフォルトで無条件に全バスサイクルをトレース取得します。(フリートレースモード)

フリートレースモードではユーザプログラムの実行開始と共にトレース取得を開始し、ユーザプログラムの停止によりトレース取得を停止します。取得したトレース情報は[トレース]ウィンドウに表示します。



Cycle	Address	CS	Code0	Code1	Data	R/W	SZ	Bus	Probes	NMI	IRQ	Timestamp
-00012	00002118		MOV.L @R6,R1	MOV R2,R6	61626623	IF		CPU	1111		1	0000h0...
-00011	0000211c		MOV R6,R4	SHLL2 R4	64634408	IF		CPU	1111		1	0000h0...
-00010	ffffbfa4				00000000	RD	L	CPU	1111		1	0000h0...
-00009	00002120		MOV R4,R0	MOV.L R1,@(R0,R5)	60430516	IF		CPU	1111		1	0000h0...
-00008	00002124		ADD #H'01,R2	MOV #H'0A,R7	7201e70a	IF		CPU	1111		1	0000h0...
-00007	ffff0424				00000000	WR	L	CPU	1111		1	0000h0...
-00006	00002128		CMP/GE R7,R2	BF @H'2106:8	32738bec	IF		CPU	1111		1	0000h0...
-00005	0000212c		ADD #H'2C,R15	MOV.L @R15+,R9	7f2c69f6	IF		CPU	1111		1	0000h0...
-00004	00002130		RTS	NOP	000b0009	IF		CPU	1111		1	0000h0...
-00003	ffffbfd0				00000000	RD	L	CPU	1111		1	0000h0...
-00002	000010a4		.DATA.W H'0000		00006262	IF		CPU	1111		1	0000h0...
-00001	000010a8				2d22d615	IF		CPU	1111		1	0000h0...
+00000	-----								1111		1	0000h0...

図3.24 トレースウィンドウ

3 デバッグ

Trace ウィンドウの表示内容は製品により異なります。本製品にて表示する項目は以下の通りです。

[PTR]	トレースメモリ内のサイクル番号。 最後に取得されたサイクルの番号を 0 とし、古いサイクルにさかのぼって、順に-1、-2 と番号が小さくなります。 ディレイカウントが設定されている場合は、トレース停止条件が成立したサイクル番号を 0 とし、成立後停止するまでに実行されたサイクル（ディレイ期間中のサイクル）には、最後に取得されたサイクルに向かって順に+1、+2 と番号が大きくなります。
[Address]	プロセッサバス上のアドレス。
[CS]	CS 空間を表示します。（SH7059 E6000H エミュレータには本項目はありません）
[Code0]	上位側 16bit の実行命令コードを表示します。
[Code1]	下位側 16bit の実行命令コードを表示します。
[Data]	バイト、ワード、ロングのデータ。それぞれ 16 進の 2 桁、4 桁、8 桁で表示します。
[R/W]	アクセスサイクルの種類。命令フェッチは IF、読み出しは RD、書き込みは WR と表示します。
[SZ]	アクセスしたサイズを示します。それぞれ B（バイト）、W（ワード）、L（ロング）で表示します。
[Bus]	バスマスタを表示します。
[Probes]	4 本のプローブ信号の状態（2 進数 4 桁）。 左から Probe4、Probe3、Probe2、Probe1 の順で表示します。
[NMI]	NMI 入力の状態を表示します。
[IRQ]	IRQ 入力の状態（0:Low レベル、1:High レベル）を表示します。 左から IRQ7、IRQ6、...IRQ0 の順で表示します。 32 ビットタイムスタンプカウンタを全て有効にした場合は、IRQ7～0 の論理積になります。
[Timestamp]	バスサイクルのタイムスタンプ。 タイムスタンプは、ユーザプログラム実行を開始するたびに 0 からカウントを始めます。測定時間の最小単位は、[トレース取得プロパティ]ダイアログボックスで設定してください。
[Source]	実行命令アドレスに対応するソースコードを表示します。
[Label]	アドレスに対応するラベル（ラベルが設定されている場合のみ表示します）。
[Timestamp-Difference]	前の行とのタイムスタンプの差分時間を表示します。

[トレース]ウィンドウ内の不要なカラムは非表示にすることができます。

カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。

カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

3.8.3 トレース情報取得条件を設定する

トレースバッファは有限であるため、バッファがいっぱいになった場合は最も古いトレースデータから順に新しいデータを上書きします。トレース情報の取得条件を設定することにより、有用なトレース情報のみを取得し、トレースバッファを有効に活用することができます。

トレース情報の取得条件には、以下の種類があります。

フリートレース	ユーザプログラムの実行を開始した時点からブレークするまでのトレース情報を連続的に取得します。(トレース情報の取得条件を設定しない場合です。)
シーケンシャルトレース停止	トレース取得条件の成立順序を指定し、全ての条件が成立した時点でトレース取得を停止します。最大7レベルの通過ポイントと1ポイントのリセット条件を設定することができます。トレース取得は停止しますが、ユーザプログラムはブレークしません。
トレースバッファオーバーフローによるトレース停止	トレース取得時にエミュレータ本体のトレースバッファがオーバーフローした場合に取得を停止します。トレース取得は停止しますが、ユーザプログラムはブレークしません。
トレース停止	指定した条件が成立したときにトレース取得を停止します。このモードでは、ユーザプログラムの実行を停止せずに、トレース取得を停止します。最大12ポイントの独立したトレース停止条件を設定することができます。トレース取得は停止しますが、ユーザプログラムはブレークしません。
アドレス範囲トレース	開始アドレスと終了アドレスで設定されている範囲(サブルーチン)の命令およびオペランドアクセスをトレースに取得します。ただし、指定されたサブルーチンが他のサブルーチンをコールした場合、コールされたサブルーチンはトレースされません。最大12ポイントの独立したアドレス範囲トレースを設定することができます。
条件トレース	指定した条件が成立した箇所のみをトレース取得します。最大12ポイントの独立した条件トレースを設定することができます。
条件アドレス範囲トレース	開始アドレスと終了アドレスで設定されている範囲(サブルーチン)の命令およびオペランドをアクセスし、設定されている条件と一致するバスサイクルのみトレース情報を取得します。アドレス範囲トレースと条件トレースを組み合わせた取得になります。最大6ポイントの独立したトレース停止条件を設定することができます。
2点間トレース	開始条件として指定したアドレス条件成立から終了条件として指定したアドレス条件成立までの2点間のトレースを取得します。
実行時間計測	トレース情報の取得条件を利用した2点間の実行時間計測を行います。
トリガ出力	指定した条件成立時に、トリガ端子からパルスを出力します。

トレース情報の取得条件はポップアップメニューから[設定...]を選択すると表示される[トレース取得プロパティ]ダイアログボックスで設定します。

[トレース取得プロパティ]ダイアログボックスは[条件設定]ページと[オプション設定]ページより構成されています。

3 デバッグ

(1) [条件設定]ページ

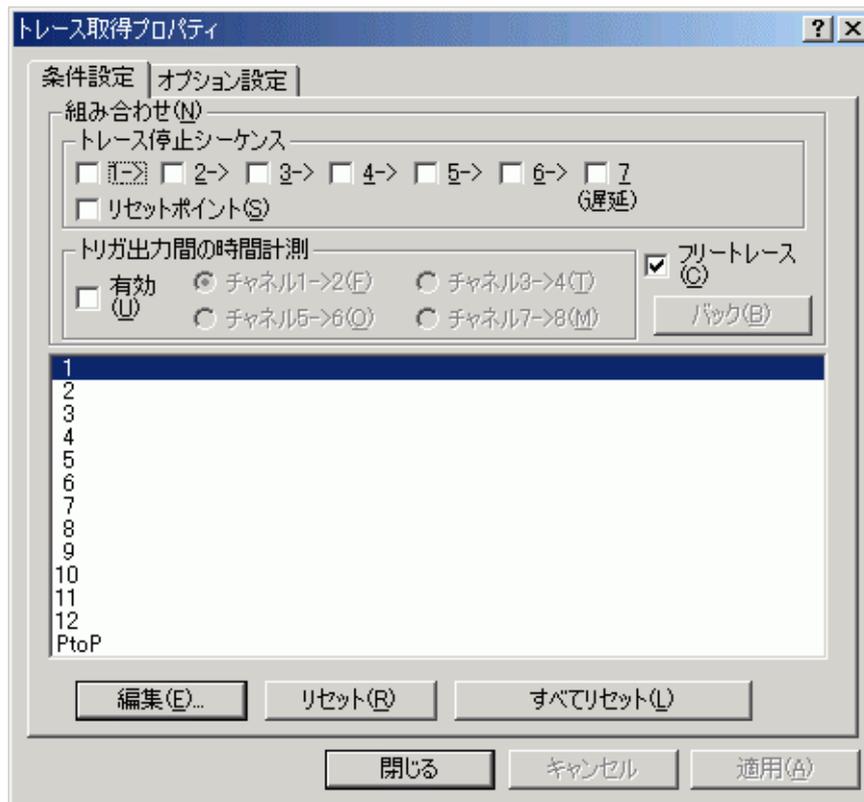


図3.25 トレース取得プロパティダイアログボックス（条件設定ページ）

[トレース停止シーケンス]	<p>チャンネル1から7を使用したシーケンシャルトレース停止を指定する場合に使用します。シーケンシャルトレース停止は複数チャンネルの条件が順番に成立した場合にトレースの取得を停止する機能です。シーケンシャルトレース停止は2ポイントから7ポイントの条件指定と1ポイントのリセット条件を選択できます。成立順序は 1->2->3->4->5->6->7の順番です。シーケンシャルトレース停止を使用する場合は、必要なチャンネルのチェックボックスをチェックしてください。</p> <p>リセット条件を設定する場合は、[リセットポイント]チェックボックスをチェックしてください。リセット条件はチャンネル8を使用します。リセット条件が成立した場合は、それまで成立したシーケンシャルトレース停止条件がすべて破棄され、再度先頭の条件からの成立をチェックします。</p> <p>シーケンシャルトレース停止を有効にした場合、チャンネル1から7のうちシーケンシャルトレース停止に使用しないチャンネルは単独の設定ができなくなります。</p>
[トリガ出力間の時間計測]	<p>実行時間計測を行うチャンネルを指定します。[有効]をクリックすることにより、トレース時間の計測を行います。チャンネルの組み合わせは、測定スタート - 測定エンドの組み合わせで、1 - 2、3 - 4、5 - 6、7 - 8の4通りです。</p>
[フリートレース] [バック]	<p>フリートレースを選択します。[フリートレース]が有効な場合はトレース情報の取得条件設定は無視されます。クリックすることにより、[組み合わせ]グループで設定した内容を、本ダイアログボックスを表示した時点に戻します。</p>
リストボックス	<p>現在の設定状況をチャンネルごとに表示します。設定のないチャンネルはチャンネル番号のみ表示します。シーケンシャルトレース停止として使用している場合はチャンネル番号の横に S を表示します。</p> <p>シーケンシャルトレース停止のリセット条件を有効している場合はチャンネル8の横に R を表示します。</p> <p>PtoP は2点間トレースに使用します。</p> <p>使用できないチャンネルはチャンネル番号の横に UNUSED を表示します。</p>
[編集...]	<p>チャンネル指定後、本ボタンをクリックすることにより、指定したチャンネルの設定を行う [トレース取得条件チャンネル n] ダイアログボックスが表示されます。(n: チャンネル番号、PtoP)</p>
[リセット]	<p>本ボタンをクリックすることにより、選択しているチャンネルの設定を解除します。</p>
[すべてリセット]	<p>すべてのチャンネルの設定を解除します。</p>

(2) [オプション設定]ページ

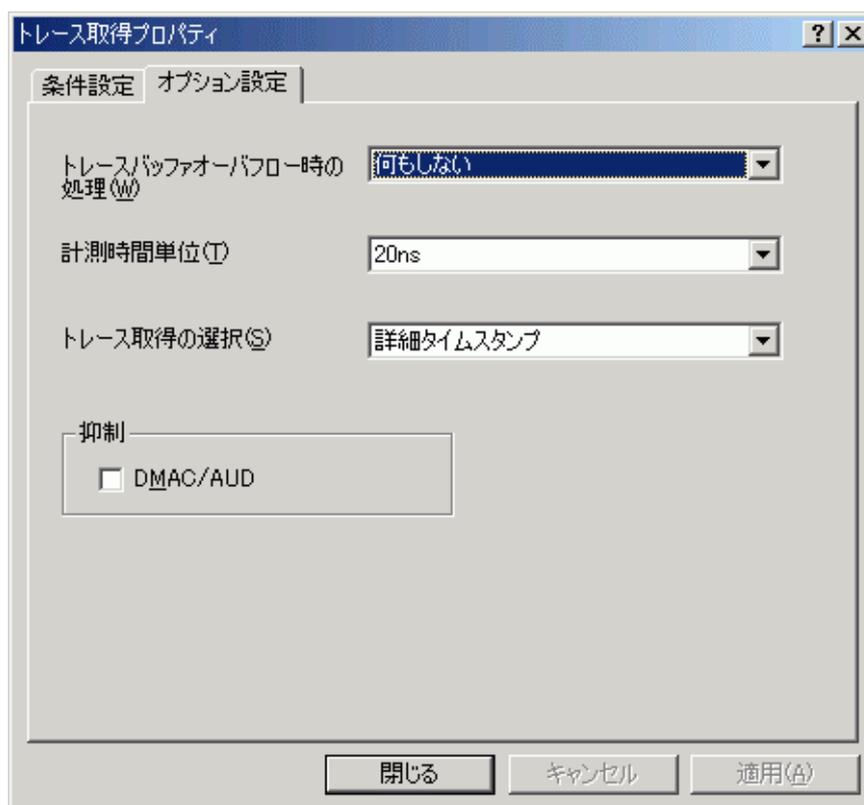


図3.26 トレース取得プロパティダイアログボックス（オプション設定ページ）

[トレースバッファオーバーフロー時の処理]	トレースバッファがいっぱいになった場合の動作を設定します。
[何もしない]	バッファがいっぱいになった場合は最も古いトレース情報から順に上書きします。
[トレース停止]	バッファがいっぱいになった場合はトレース取得を停止します。 トレース取得は停止しますが、ユーザプログラムの実行は停止しません。
[実行停止およびトレース停止]	バッファがいっぱいになった場合はユーザプログラムの実行を停止します。
[計測時間単位]	バストレース情報のタイムスタンプの最小時間を次の中から選択します。
[52us]	最小時間（52us）でタイムスタンプ情報を取得します。
[1.6us]	最小時間（1.6us）でタイムスタンプ情報を取得します。
[20ns]	最小時間（20ns）でタイムスタンプ情報を取得します。
[Clock]	内部クロック（ ）に同期したバスサイクルのクロック数を取得します。
[Clock/2]	内部クロック（ ）の 1/2 に同期したバスサイクルのクロック数を取得します。
[Clock/4]	内部クロック（ ）の 1/4 に同期したバスサイクルのクロック数を取得します。
[Clock/8]	内部クロック（ ）の 1/8 に同期したバスサイクルのクロック数を取得します。
[トレース取得の選択]	タイムスタンプと IRQ7-0 の表示形態を選択します。本 E6000H エミュレータにおいては、タイムスタンプと IRQ のトレースは同じハードリソースを使用し実現しています。このため、32 ビットタイムスタンプカウンタと IRQ7-0 のトレースを同時に行うことはできません。 使用状況にあわせて下記項目を選択してください。
[詳細タイムスタンプ]	タイムスタンプ表示は 32 ビットタイムスタンプカウンタを全て有効にします。 IRQ 表示は IRQ7-0 の論理積となります。
[IRQ7-0 全て表示]	タイムスタンプの下位 16 ビットは 0 固定になります。 IRQ 表示は IRQ7-0 を個別に表示します。
[抑制]	バスサイクルタイプの DMAC/ AUD を取得しない指定をします。

3 デバッグ

(3) [トレース取得条件 チャンネル n]ダイアログボックス

[トレース取得条件 チャンネル n]ダイアログボックスは、シーケンシャルトレース停止の通過ポイントとリセットポイントの設定、トレース停止条件の設定、アドレス範囲トレース条件の設定、条件アドレス範囲トレース条件の設定、条件トレースの設定、2点間トレース条件の設定、実行時間計測およびトリガ出力の条件設定に使用します。

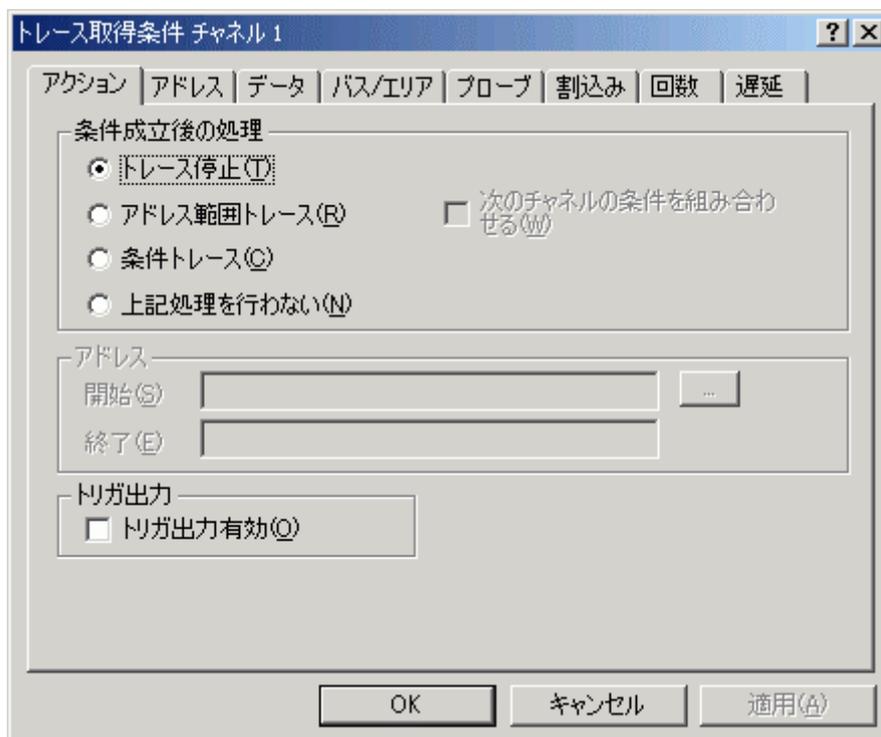


図3.27 [トレース取得条件 チャンネル n]ダイアログボックス

[トレース取得条件 チャンネル n]ダイアログボックスは、[アクション]ページ、[アドレス]ページ、[データ]ページ、[バス/エリア]ページ、[プローブ]ページ、[割り込み]ページ、[回数]ページおよび[遅延]ページから構成されます。

各ページの条件を組み合わせることで複雑な設定を行うことができます。

[アクション]

[条件成立後の処理]	条件成立時の処理を指定します。
[トレース停止]	トレース停止を設定します。
[アドレス範囲トレース]	アドレス範囲トレースを設定します。 [アドレス範囲トレース]を選択し、[次のチャンネルの条件を組み合わせる]をチェックすることで、条件アドレス範囲トレースを設定することができます。
[条件トレース]	条件トレースを設定します。
[上記処理を行わない]	条件が成立しても上記の処理は行わない設定を指定します。トリガ出力や実行時間の計測を行う場合に使用します。
[アドレス]	アドレス範囲トレースと条件アドレス範囲トレースのアドレス範囲の設定および2点間トレース条件の開始アドレスと終了アドレスを設定します。
[開始]	開始アドレスを設定します。
[終了]	終了アドレスを設定します。
[...]	関数のアドレス範囲を[開始]と[終了]に設定することができます。
[トリガ出力有効]	トレース条件成立後、トリガ出力します。
[アドレス]	アドレスバス条件を指定します。
[データ]	データバス条件を指定します。
[バス/エリア]	アクセスタイプ、バスステータスおよびリード/ライトサイクル条件を設定します。
[プローブ]	外部プローブ信号 (PRB1 ~ PRB4) の状態 (HIGH/LOW) を条件にします。
[割り込み]	IRQ 信号および NMI 信号の状態を条件にします。
[回数]	条件成立回数を設定します。
[遅延]	条件成立後処理開始までの遅延サイクル数を指定します。遅延サイクルを利用することで、条件成立前後のトレース情報を確認することができます。[無効]を指定した場合は、遅延を行いません。

- 【注】
1. [アドレス]ページ、[データ]ページ、[バス/エリア]ページ、[プローブ]ページ、[割り込み]ページおよび[回数]ページの設定はオンエミュレータブレークと同様です。オンエミュレータブレークについては、「3.7 イベントポイントを使用する」を参照してください。
 2. アドレス範囲トレースの範囲指定は、開始アドレス 終了アドレスとなるように指定してください。
 3. 条件アドレス範囲トレースはチャンネルを2つ使用します。条件アドレス範囲トレースを設定する場合は、奇数チャンネル (2n+1) にアドレス範囲トレースを設定し、偶数チャンネル (2n+2) に条件トレースを設定したのち[次のチャンネルの条件を組み合わせる]をチェックしてください。

3.8.4 Trace レコードを検索する

トレースレコードを検索するには[Trace Find]ダイアログボックスを使用します。

[Trace Find]ダイアログボックスを開くには、ポップアップメニューの[検索...]を選択します。

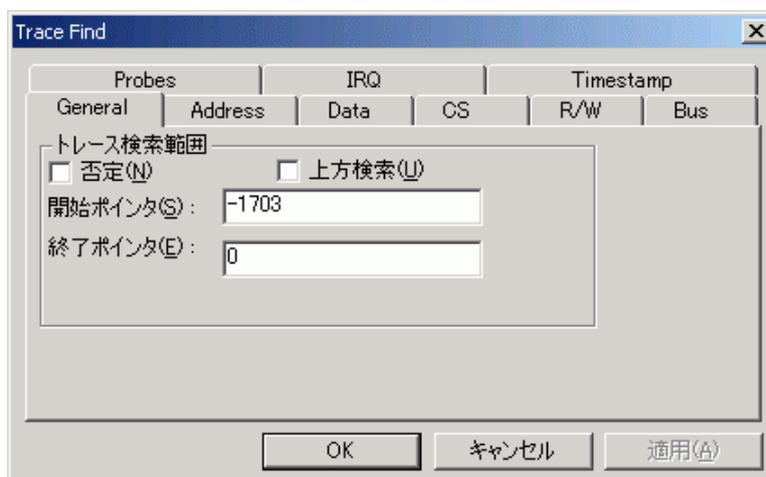


図3.28 Trace Find ダイアログボックス

[Trace Find]ダイアログボックスは下記ページより構成されています。

[General]	検索範囲を指定します。
[否定]	チェックすると他ページで設定した項目の否定条件で検索します。
[上方検索]	チェックすると上方検索を行います。
[開始ポイント]	検索を開始する PTR の値を入力します。
[終了ポイント]	検索を終了する PTR の値を入力します。
[Address]	実行アドレス条件を指定します。
[無効]	チェックすると実行アドレスを検出しません。
[値]	指定した実行アドレスを検出します。実行アドレス値を入力します。
[Date]	データ条件を指定します。
[無効]	チェックするとデータを検出しません。
[値]	指定したデータを検出します。データ値を入力します。
[CS]	CS 条件を指定します。(SH7059 E6000H エミュレータでは CS 条件の指定はできません)
[無効]	チェックすると CS を検出しません。
[値]	指定した CS を検出します。
[R/W]	リード/ライト条件を指定します。
[無効]	チェックするとリード/ライト条件を検出しません。
[設定]	指定したリード/ライト条件を検出します。
	RD: リードサイクル
	WR: ライトサイクル
	IF: 命令フェッチサイクル
[Bus]	バスマスタ条件を指定します。
[無効]	チェックするとバスマスタ条件を検出しません。
[設定]	指定したバスマスタ条件を検出します。
[Probes]	プローブ条件を指定します。
[無効]	チェックするとプローブ条件を検出しません。
[設定]	指定したプローブ条件を検出します。
	Don't care: 選択したプローブ条件を検出しません。
	High: プローブ信号状態が High
	Low: プローブ信号状態が Low

[IRQ]	IRQ 条件を指定します。
[無効]	チェックすると IRQ 条件を検出しません。
[設定]	指定した IRQ 条件を検出します。
	Don't care: 選択した IRQ 条件を検出しません。
	High: 信号状態が High
	Low: 信号状態が Low
[Timestamp]	タイムスタンプ条件を指定します。
[無効]	チェックするとタイムスタンプ条件を検出しません。
[値]	指定したタイムスタンプ値を検出します。タイムスタンプ値を入力します。(全ての桁の入力が必須)

各ページで条件を設定し、[OK]ボタンをクリックすることにより、サーチ条件を設定し、検索を開始します。[キャンセル]ボタンをクリックすると、設定しないでダイアログボックスを閉じます。

検索の結果一致するトレースレコードが見つかった場合は当該レコード行を強調表示します。一致するトレースレコードが見つからなかった場合は、メッセージダイアログボックスを表示します。

トレース情報の検索は各ページで設定した条件がすべて一致するトレース情報のみを検索します。

トレースレコードが検索できた場合は、ポップアップメニューで[次を検索]を選択すると、次のトレースレコードを検索できます。

3.8.5 トレース情報をクリアする

トレース情報をクリアするには、ポップアップメニューから[クリア]を選択します。トレース情報を保持しているトレースバッファを空にします。複数の[トレース]ウィンドウが開いているときは、それらは同じバッファをアクセスしているため、すべての[トレース]ウィンドウをクリアすることになります。

3.8.6 トレース情報をファイルに保存する

トレース情報をファイルに保存するには、ポップアップメニューから[保存...]を選択します。

[名前を付けて保存]ダイアログボックスを表示します。[トレース]ウィンドウに表示しているトレース情報をテキストファイルとして保存します。保存する範囲を、[PTR]の範囲によって指定することができます(すべてのバッファをセーブするには、数分かかることがあります)。このファイルは保存のみ可能で、[トレース]ウィンドウへの読み込みはできないことに注意してください。

【注】 トレース情報をフィルタリングした場合、保存する範囲の指定はできません。フィルタリングした結果[トレース]ウィンドウに表示されたトレース情報すべてを保存します。保存する範囲を指定したい場合は[Trace Filter]ダイアログボックスの[General]ページよりフィルタ範囲を指定してください。
フィルタ機能については「0 取得したトレース情報から必要なレコードを抽出する」を参照してください。

3.8.7 [エディタ]ウィンドウを表示する

トレースレコードに対応する[エディタ]ウィンドウを表示するには2通りの方法があります。

- (1) トレースレコードを選択した状態でポップアップメニューから[ソースファイル表示]を選択する
- (2) トレースレコードをダブルクリックする

上記の操作により、[エディタ]ウィンドウあるいは[逆アセンブル]ウィンドウを開いてソース表示し、選択した行をカーソルで示します。

3.8.8 ソース表示を整形する

ポップアップメニューで[ソーストリム]を選択すると、ソースプログラムの左側の空白を取り除きます。

取り除いた状態だと[ソーストリム]メニューの左にチェックが付きます。チェックありの状態では[ソーストリム]メニューを選択すると取り除いた空白を元に戻します。

3.8.9 トレース情報の取得を一時的に停止する

ユーザプログラム実行中、一時的にトレース情報の取得を停止するにはポップアップメニューから[停止]を選択します。トレース取得を中止し、トレース表示を更新します。ユーザプログラムを停止せずにトレース情報の取得のみ停止し、トレース情報を確認する場合などに使用します。

3.8.10 トレース情報の取得を再開する

ユーザプログラム実行中、一時的にトレース情報の取得を停止した場合、再度トレース情報の取得を再開するにはポップアップメニューから[リスタート]を選択します。

3.8.11 取得したトレース情報から必要なレコードを抽出する

取得したトレース情報から必要なレコードのみを抽出するにはフィルタ機能を使用します。フィルタ機能はハードウェアにより取得したトレース情報をソフトウェアによりフィルタリングします。取得条件を設定してトレース情報を取得する[トレース取得プロパティ]設定と異なり、取得したトレース情報に対し何度もフィルタ設定を変更することで必要な情報が簡単に抽出でき、データの分析に役立ちます。フィルタ機能を使用してもトレースバッファの内容は変更されません。トレースバッファは有限ですので、[トレース取得プロパティ]設定により有用なトレース情報をより多く取得することで、より効果的にデータの分析が可能となります。

フィルタ機能を使用するには[Trace Filter]ダイアログボックスを使用します。フィルタ範囲を指定します。



図3.29 Trace Filter ダイアログボックス

[Trace Filter]ダイアログボックスを開くには、ポップアップメニューの[フィルタ...]を選択します。

[Trace Filter]ダイアログボックスは下記ページより構成されています。

[General]	検索範囲を指定します。
[他のページ設定を無効にする]	チェックするとサイクル番号のみ指定できます。他のオプションはすべて無効になります。
[フィルタを有効にする]	チェックするとフィルタを有効にします。
[否定]	チェックすると他ページで設定した項目の否定条件でフィルタリングします。
[開始ポイント]	フィルタを開始する PTR の値を入力します。
[終了ポイント]	フィルタを終了する PTR の値を入力します。
[Address]	実行アドレス条件を指定します。
[無効]	チェックすると実行アドレスをフィルタしません。
[設定]	指定した実行アドレスを検出します。
[ポイント]	単一実行アドレスを入力します。
[範囲]	実行アドレス範囲を入力します。
[開始]	単一実行アドレスまたは実行アドレス範囲の開始アドレスを入力します。
[終了]	実行アドレス範囲の終了アドレスを入力します。
[Data]	データ条件を指定します。
[無効]	チェックするとデータをフィルタしません。
[設定]	指定したデータを検出します。
[ポイント]	単一データを入力します。
[範囲]	データ範囲を入力します。
[開始]	単一データまたはデータ範囲の最小値を入力します。
[終了]	データ範囲の最大値を入力します。
[CS]	CS 条件を指定します。(SH7059 E6000H エミュレータでは CS 条件の指定はできません)
[無効]	チェックすると CS をフィルタしません。
[設定]	指定した CS を検出します。
[R/W]	リード/ライト条件を指定します。
[無効]	チェックするとリード/ライト条件を検出しません。
[設定]	指定したリード/ライト条件を検出します。
[RD]	チェックするとリードサイクルを検出します。
[WR]	チェックするとライトサイクルを検出します。
[Probes]	プローブ条件を指定します。
[無効]	チェックするとプローブ条件を検出しません。
[設定]	指定したプローブ条件を検出します。
Don't care:	選択したプローブ条件を検出しません。
High:	プローブ信号状態が High
Low:	プローブ信号状態が Low
[IRQ]	IRQ 条件を指定します。
[無効]	チェックすると IRQ 条件を検出しません。
[設定]	指定した IRQ 条件を検出します。
Don't care:	選択した IRQ 条件を検出しません。
High:	IRQ 信号状態が High
Low:	IRQ 信号状態が Low
[Timestamp]	タイムスタンプ条件を指定します。
[無効]	チェックするとタイムスタンプ値を検出しません。
[設定]	指定したタイムスタンプ値を検出します。
[ポイント]	単一タイムスタンプを指定します。
[範囲]	タイムスタンプ範囲を指定します。
[開始]	単一タイムスタンプまたはタイムスタンプ範囲の最小値を入力します。
[終了]	タイムスタンプ範囲の最大値を入力します。

3 デバッグ

各ページでフィルタ条件を設定し、[OK]ボタンをクリックすることにより、フィルタ条件にしたがいフィルタリングを行います。[キャンセル]ボタンをクリックすると、[Trace Filter]ダイアログボックスを開いた時点の設定のままダイアログボックスを閉じます。

フィルタリングは各ページで設定したフィルタ条件が1つ以上一致するトレース情報のみを[トレース]ウィンドウに表示します。

フィルタリングを行ってもトレースバッファの内容は変更されませんので、何度もフィルタ条件を変更しデータの分析ができます。

3.8.12 タイムスタンプの差を計算する

タイムスタンプ情報取得時、トレース結果より指定した2点間の時間差を計算するには、ポップアップメニューから[タイムスタンプ差...]を選択します。

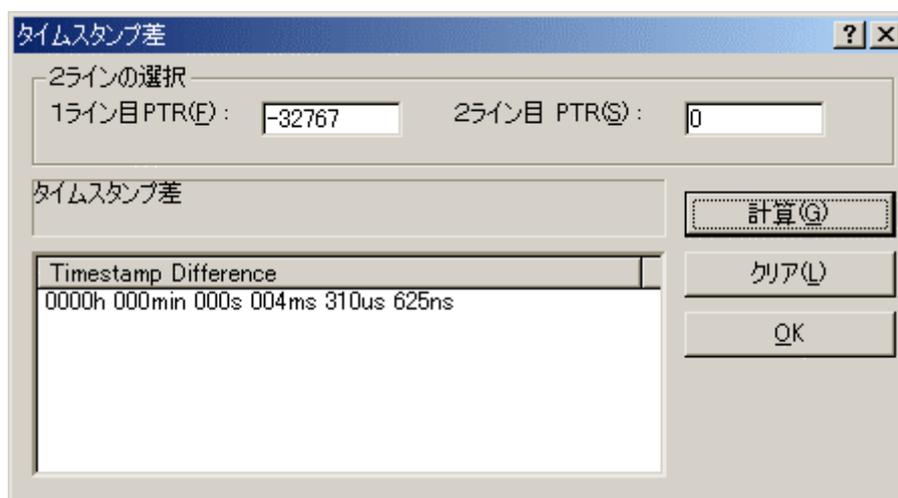


図3.30 タイムスタンプ差ダイアログボックス

[2ラインの選択]	タイムスタンプの差を計算するトレースレコードを指定します。
[1ライン目 PTR]	差分を計測する1点目のポイントを指定します。 デフォルト値は、[トレース]ウィンドウ上で選択されているラインのポイントが表示されます。
[2ライン目 PTR]	差分を計測する2点目のポイントを指定します。
[タイムスタンプ差]	2点間の時間差を計算した結果を表示します。
[計算]	指定した2点間の差分を計算し、結果を[タイムスタンプ差]リストに表示します。
[クリア]	[タイムスタンプ差]リストのすべての結果を消去します。
[OK]	ダイアログボックスを閉じます。 このとき、[タイムスタンプ差]リストのすべての結果は消去されます。

3.8.13 統計情報を解析する

指定された条件で統計情報の解析を実行するには、ポップアップメニューから[統計...]を選択します。
[統計]ダイアログボックスが開きます。

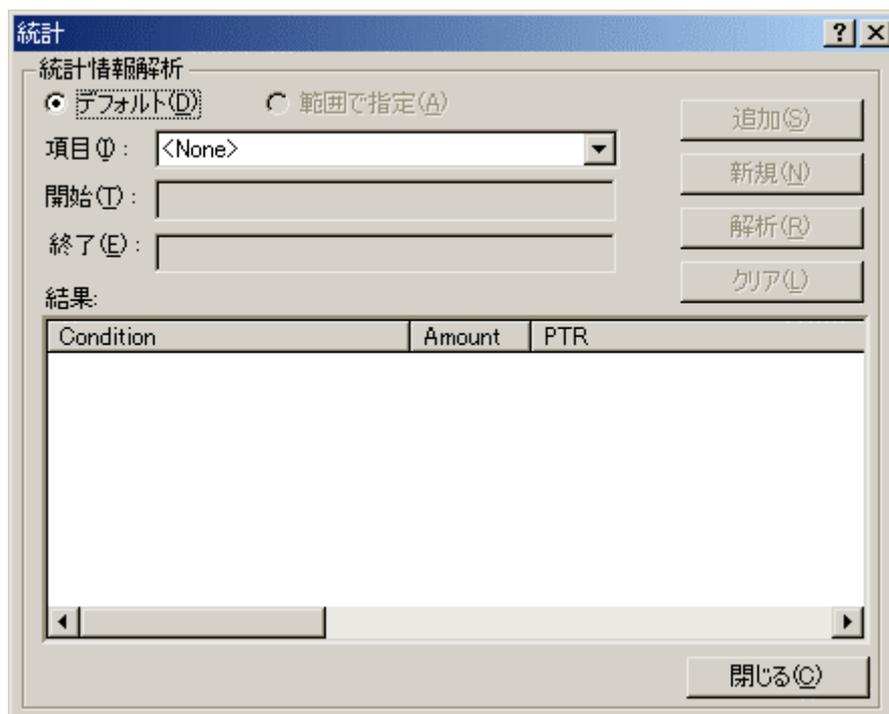


図3.31 統計ダイアログボックス

[統計情報解析]	統計情報を解析するための設定を行います。
[デフォルト]	単一の入力値または文字列を指定します。
[範囲で指定]	入力値または文字列を範囲で指定します。
[項目]	解析対象項目を指定します。
[開始]	入力値または文字列を指定します。 範囲で指定する場合は開始値を設定します。
[終了]	範囲で指定する場合の終了値を設定します。 ([範囲で指定] 選択時有効)
[追加]	現在の条件に追加設定します。
[新規]	新しい条件を指定します。
[解析]	統計情報解析の結果を取得します。
[結果]	すべての条件と統計情報解析結果を削除します。
[閉じる]	ダイアログボックスを閉じます。 このとき、[結果]リストのすべての結果は消去されます。

本ダイアログボックスは、トレース情報の統計情報解析に使用します。[項目]オプションで解析対象項目を指定し、[開始]オプションおよび[終了]オプションで入力値または文字列を指定します。

[新規]ボタンまたは[追加]ボタンにより条件を設定し[解析]ボタンをクリックすると、統計情報を解析し[結果]リストに解析結果を表示します。

【注】 本エミュレータでは[PTR]項目のみ範囲で指定可能です。それ以外の項目は単一の文字列で指定してください。統計情報の解析における文字列の判定は[トレース]ウィンドウに表示される文字列と比較し、完全一致したものをカウントします。ただし、大文字小文字は区別しません。また、空白の数も考慮しません。

3.8.14 取得したトレース情報から関数呼び出し箇所を抽出する

取得したトレース情報から関数呼び出し箇所のみを抽出するには、ポップアップメニューから[関数コール...]を選択します。
[関数コール箇所の表示]ダイアログボックスが開きます。



図3.32 関数コール箇所の表示ダイアログボックス

[設定]	関数呼び出し箇所の抽出を行うかどうか設定します。
[許可]	関数呼び出し箇所の抽出を行います。
[無効]	関数呼び出し箇所の抽出を行いません。

[許可]オプションを選択した場合、取得したトレース情報より関数呼び出しを行っているサイクルのみを抽出し表示します。
関数呼び出し箇所の抽出を行ってもトレースバッファの内容は変更されません。

フリートレースのトレース結果または、関数の呼び出しを含んだトレース情報に対して本機能を使用することにより、関数の呼び出し順序を調べることができます。

3.9 パフォーマンスを測定する

ユーザプログラムの実行効率を測定するにはパフォーマンス解析機能を使用します。

パフォーマンス解析機能はE6000Hエミュレータのハードウェアパフォーマンス測定回路により指定範囲の実行効率を測定するため、リアルタイム性は損なわれません。

実行効率測定の設定は測定用途に応じて下記5つのモードより選択できます。

表3.1 実行効率測定条件で設定可能な測定モード

測定モード	測定内容	測定用途
指定範囲内時間測定	指定した範囲の実行時間および実行回数を測定します。	関数の処理時間のうち関数内から呼び出す子関数の処理時間を除いた処理時間を測定する場合などに使用します。
指定アドレス間時間測定	指定したアドレス間の実行時間および実行回数を測定します。	関数の処理時間を測定する場合などに使用します。
指定アドレス範囲間時間測定	指定範囲から別の指定範囲までの実行時間を測定します。	アセンブリプログラムなど、サブルーチンを連続して配置したプログラムにおいて、連続するサブルーチンのうちのいずれかが呼ばれてから別の連続するサブルーチンのいずれかが呼ばれるまでの実行時間を測定する場合などに使用します。
領域アクセス回数測定	指定した範囲から別の指定した範囲に対するアクセス回数を測定します。	特定の関数からのグローバル変数に対するアクセス回数を測定する場合などに使用します。
指定範囲内コール回数測定	指定した範囲から別の指定した範囲をコールした回数を測定します。	特定の関数からの関数コール回数を測定する場合などに使用します。

実行効率測定の設定はE6000Hハードウェアパフォーマンス測定回路に実装されたパフォーマンスチャンネル(8チャンネル)を使用し、最大8ポイントまで設定できます。

ただし、指定アドレス範囲間時間測定、領域アクセス回数測定、指定範囲内コール回数測定の設定では1条件設定あたり連続した2ポイントを使用するため、これらの測定モードを使用する場合、設定可能ポイント数は最大4ポイントとなります。

表3.2 実行効率測定条件のモード設定

測定モード	ポイント							
	1	2	3	4	5	6	7	8
指定範囲内時間測定								
指定アドレス間時間測定								
指定アドレス範囲間時間測定		x		x		x		x
領域アクセス回数測定		x		x		x		x
指定範囲内コール回数測定		x		x		x		x

: 設定可, x: 設定不可

【注】 指定範囲内時間測定および指定アドレス間時間測定は1ポイントを使用し、指定アドレス範囲間時間測定・領域アクセス回数測定および指定範囲内コール回数測定は連続した2ポイントを使用します。2ポイントを使用するモードから1ポイントを使用するモードに変更したとき、また、1ポイントを使用するモードから2ポイントを使用するモードに変更したとき、設定されていた条件は削除されます。

3 デバッグ

3.9.1 パフォーマンス解析ウィンドウを開く

[パフォーマンス解析]ウィンドウを開くには、[表示 -> パフォーマンス -> パフォーマンス解析]を選択するか、[パフォーマンス解析]ツールバーボタンをクリックして[パフォーマンス解析方式の選択]ダイアログボックスを開きます。



図3.33 パフォーマンス解析方式の選択ダイアログボックス

[E6000H パフォーマンス解析]を選択し[OK]ボタンをクリックすると[パフォーマンス解析]ウィンドウが開きます。

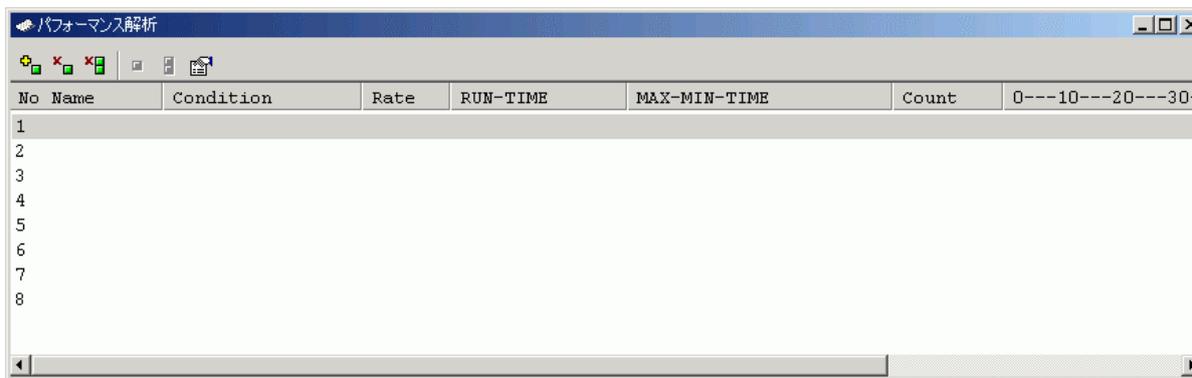


図3.34 パフォーマンス解析ウィンドウ

直前のプログラム実行でユーザが選択したエリアの実行時間比率をパーセント、ヒストグラムおよび数値で表示します。

[パフォーマンス解析]ウィンドウ内の不要なカラムは非表示にすることができます。

カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。

カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

3.9.2 実行効率測定条件を設定する

[パフォーマンス解析]ウィンドウでは測定条件の設定内容の表示および変更ができます。

条件を設定するポイントを選択し、ポップアップメニューから[設定...]を選択すると[パフォーマンス解析プロパティ]ダイアログボックスを表示します。

実行効率測定条件は[測定モード]オプションにより5つのモードから選択可能です。

表3.3 実行効率測定条件

測定モード	
1	指定範囲内時間測定
2	指定アドレス間時間測定
3	指定アドレス範囲間時間測定
4	領域アクセス回数測定
5	指定範囲内コール回数測定

使用するモードに応じて実行効率測定条件を設定します。

各モードにより設定するパラメータが異なります。

[パフォーマンス解析プロパティ]ダイアログボックスにはアドレス範囲指定時、関数名を入力することにより関数のアドレス範囲を自動的に入力する入力補助機能があります。

[パフォーマンス解析プロパティ]ダイアログボックス上にある[...]ボタンをクリックすると表示される[関数範囲入力]ダイアログボックスより、関数名を指定すると自動的に関数のアドレス範囲を入力することができます。

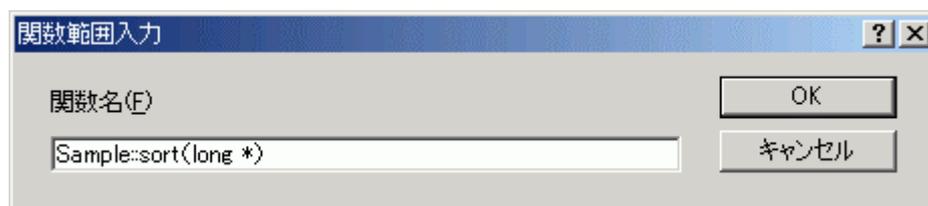


図3.35 関数範囲入力ダイアログボックス

- 【注】
1. オーバード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスが開きますので、関数を選択してください。
 2. 算出されるアドレスは参考値です。場合により関数の終了アドレスが異なる場合があります。[逆アセンブリ]ウィンドウにより関数の最終命令を確認し、[終了アドレス]の設定値を最終命令のアドレスに補正してください。
(一般的に関数の最終命令は RTS 命令となります。)
アドレス入力ではアドレス値以外にラベル名および式の指定も可能です。

3 デバッグ

(1) 指定範囲内時間測定

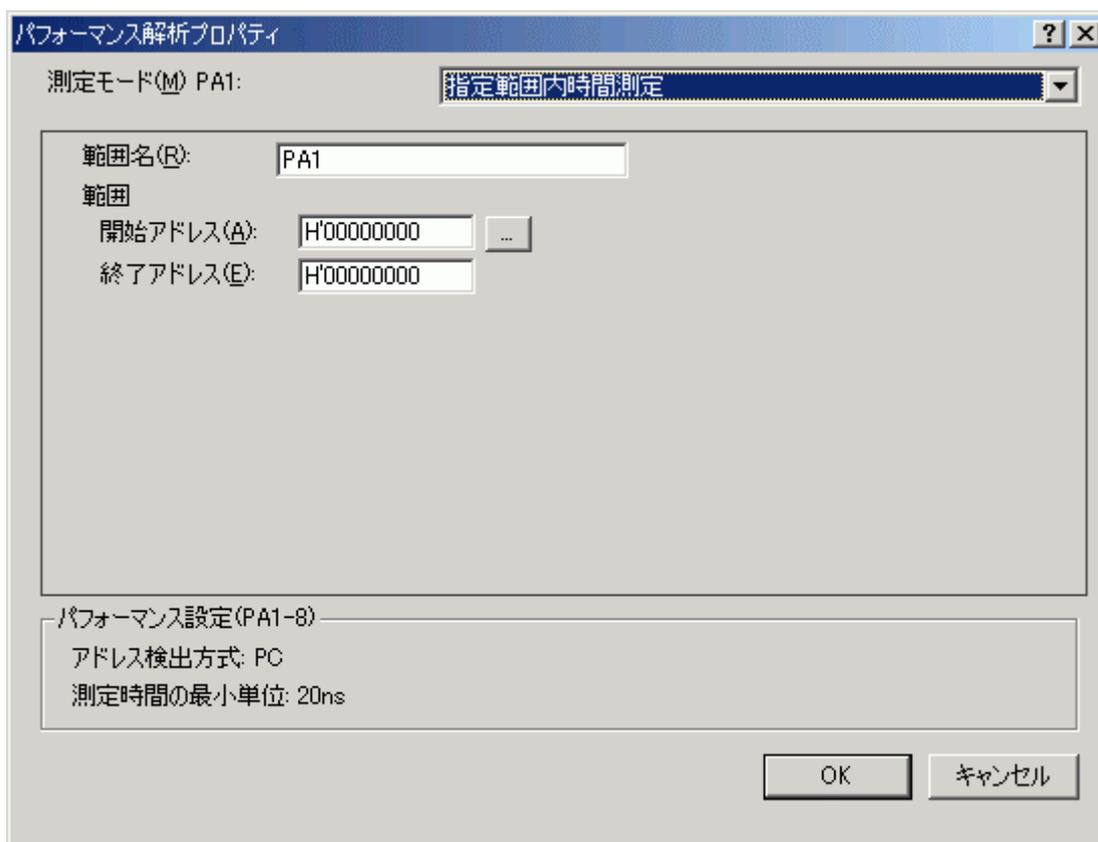


図3.36 指定範囲内時間測定

[範囲名]	範囲の名称を指定します。
[範囲]	指定範囲内時間測定を行う範囲を指定します。
[開始アドレス]	開始アドレスを指定します。
[終了アドレス]	終了アドレスを指定します。

<開始アドレス>と<終了アドレス>で設定された範囲の実行時間および実行回数を測定します。

測定時間は<開始アドレス>と<終了アドレス>の範囲のプログラムプリフェッチで測定を開始し、範囲以外のプログラムプリフェッチで測定を中断します。再度、設定範囲のプログラムプリフェッチで測定を開始します。

実行回数は、設定範囲の<終了アドレス>のプログラムをフェッチするたびにカウントします。

測定結果には、設定範囲内から呼び出された処理の実行時間は含まれません。

(2) 指定アドレス間時間測定

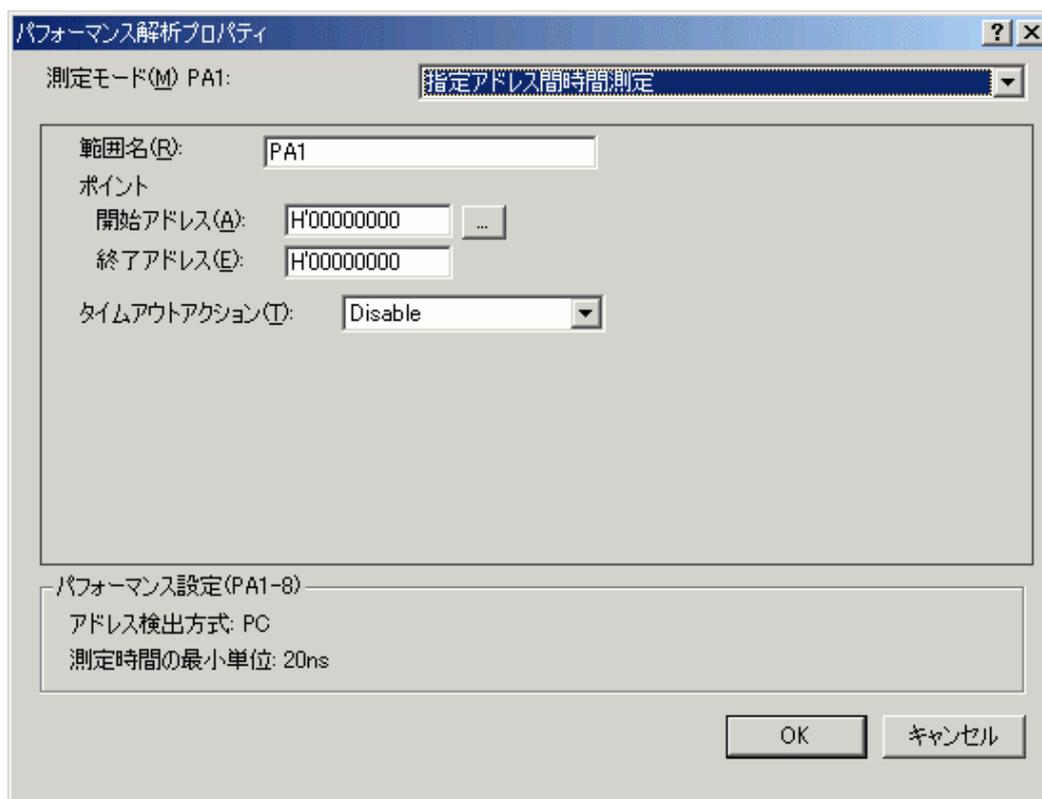


図3.37 指定アドレス間時間測定

[範囲名]	範囲の名称を指定します。
[ポイント]	指定アドレス間時間測定を行う範囲を指定します。
[開始アドレス]	開始アドレスを指定します。
[終了アドレス]	終了アドレスを指定します。
[タイムアウトアクション]	実行時間測定タイムアウトまたはカウントアウトが発生した時の動作を指定します。 無効:実行時間測定タイムアウト値およびカウントアウト値の設定を禁止します。 実行停止:実行時間測定タイムアウトまたはカウントアウト発生時、ユーザプログラムを停止します。 トレース停止:実行時間測定タイムアウトまたはカウントアウト発生時、トレース取得を停止します。 (チャンネル1のみ設定可能)
[タイムアウト値]	実行時間測定タイムアウト値を指定します。 測定時間の最小単位が 160ns、40ns、20ns の指定は、 時 h 分 min 秒 s ミリ秒 ms マイクロ秒 us ナノ秒 ns (例: 1h 2min 3s 123ms 456us 789ns) で入力し、ターゲットクロック指定は、16 進で 10 桁 (例: 123456789A) と入力します。 開始、終了アドレス間の 1 回ごとの計測値がタイムアウト値を超えたときにブレイクします (トータル時間ではありません)。 (チャンネル1のみ設定可能)
[回数]	実行回数測定カウントアップ値を指定します。 実行回数がカウントアップ値を超えたときにブレイクします。 (チャンネル1のみ設定可能)

<開始アドレス>と<終了アドレス>で設定された範囲の実行時間および実行回数を測定します。

測定時間は、<開始アドレス>のプログラムプリフェッチで測定を開始し、<終了アドレス>のプログラムプリフェッチで測定を中断します。

実行回数は、設定範囲の<終了アドレス>のプログラムをプリフェッチするたびにカウントします。

測定結果には、設定範囲内から呼び出された処理の実行時間を含みます。

3 デバッグ

ポイント1~4の場合に、設定された範囲の最大、最小時間を測定します。

(3) 指定アドレス範囲間時間測定

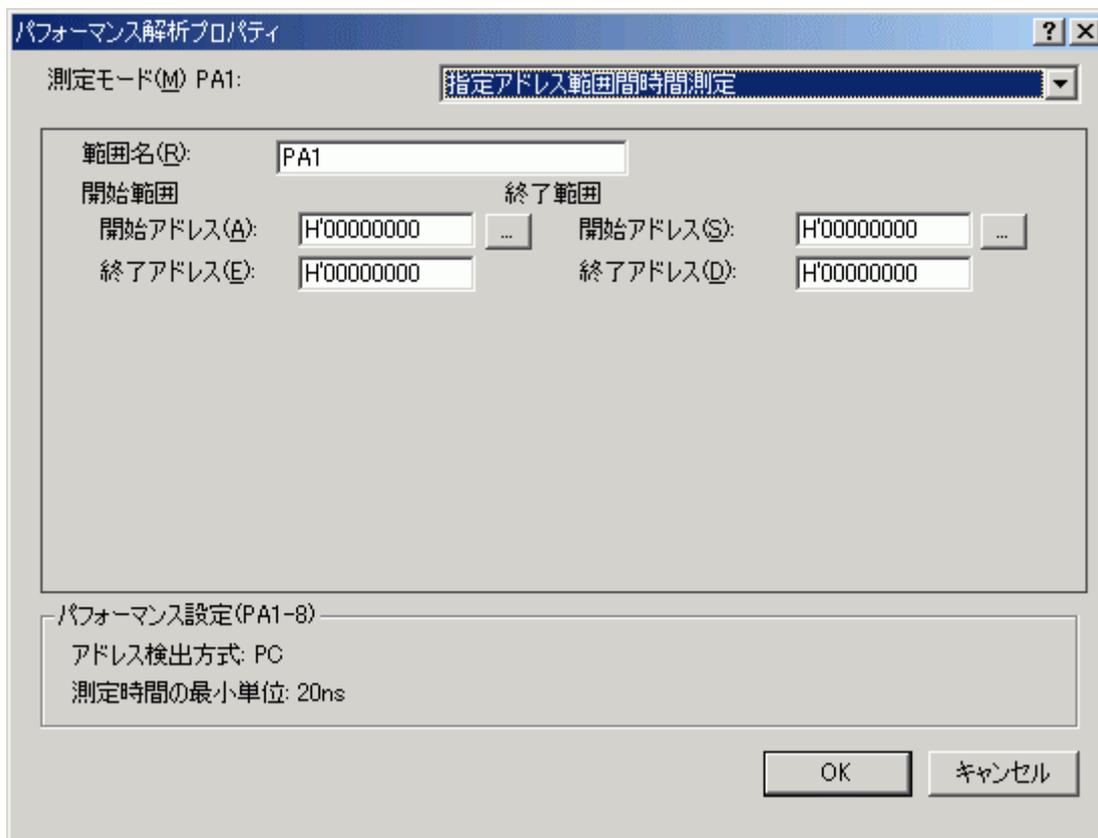


図3.38 指定アドレス範囲間時間測定

[範囲名]	範囲の名称を指定します。
[開始範囲]	指定アドレス範囲間時間測定を行う開始範囲を指定します。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。
[終了範囲]	指定アドレス範囲間時間測定を行う終了範囲を指定します。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。

<開始アドレス範囲>のプリフェッチサイクルで時間測定を開始し、<終了アドレス範囲>のプログラムプリフェッチサイクルで測定を中断します。また、実行回数は、<終了アドレス範囲>を通過するたびにカウントアップします。

(4) 領域アクセス回数測定

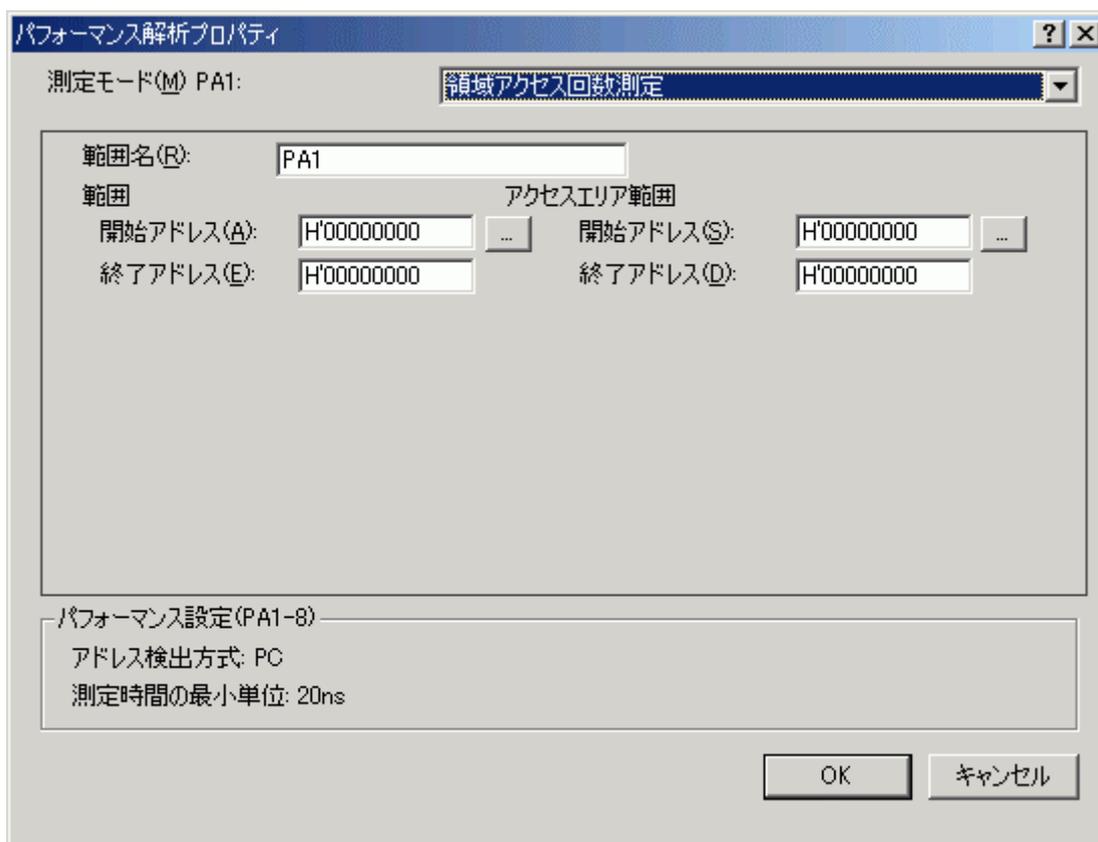


図3.39 領域アクセス回数測定

[範囲名]	範囲の名称を指定します。
[範囲]	領域アクセス回数測定を行う範囲を指定します。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。
[アクセスエリア範囲]	領域アクセス回数測定を行うアクセス領域アドレス範囲を指定します。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。

<開始アドレス>と<終了アドレス>で設定されている範囲から<アクセス領域アドレス範囲>で設定されている領域をアクセスした回数を測定します。また、範囲内の実行時間は、指定範囲内時間測定を用いて測定します。

3 デバッグ

(5) 指定範囲内コール回数測定

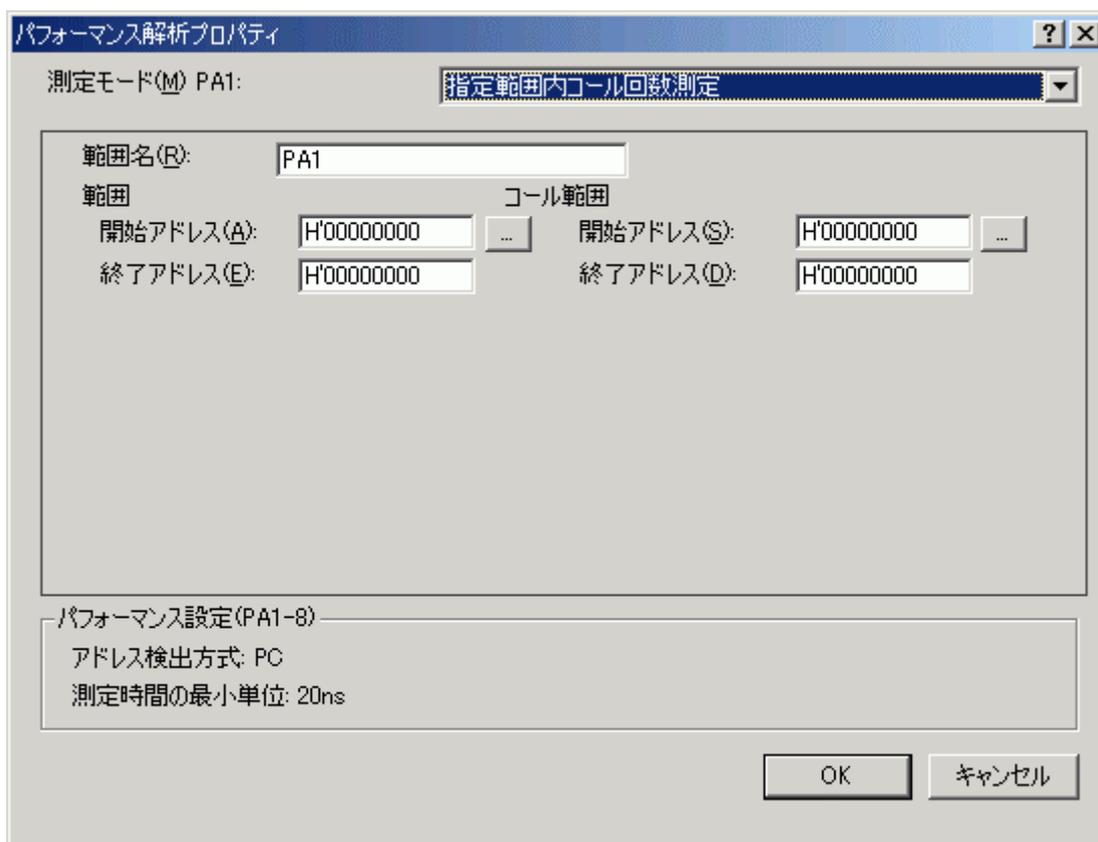


図3.40 指定範囲内コール回数測定

[範囲名]	範囲の名称を指定します。
[範囲]	指定範囲内コール回数測定を行う範囲を指定します。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。
[コール範囲]	指定範囲内コール回数測定を行うコール範囲を指定します。 コール範囲は、指定サブルーチンの開始アドレスと終了アドレスを指定してください。
[開始アドレス]	先頭アドレスを指定します。
[終了アドレス]	最終アドレスを指定します。

<開始アドレス>、<終了アドレス>で設定されている範囲から<コール範囲>で設定されている範囲をコールした回数を測定します。また、範囲内の実行時間は、指定範囲内時間測定を用いて測定します。

<コール範囲>は、指定サブルーチンの開始アドレスと終了アドレスを指定してください。

3.9.3 実行効率測定を開始する

ユーザプログラムを実行すると前回の測定結果をクリアした後、設定した実行効率測定条件にしたがい自動的に実行効率測定を開始します。

ユーザプログラムを停止すると、測定結果を[パフォーマンス解析]ウィンドウに表示します。

3.9.4 測定条件を削除する

測定条件を選択した状態で、ポップアップメニューから[リセット]を選択すると、選択された測定条件を削除します。

3.9.5 すべての測定条件を削除する

ポップアップメニューから[全てリセット]を選択すると、設定している測定条件をすべて削除します。

3.10 プロファイル機能を使用する

3.10.1 プロファイルを有効にする

[表示->パフォーマンス->プロファイル]を選択し、[プロファイル]ウィンドウをオープンします。

[プロファイル]ウィンドウのポップアップメニューで[有効]メニューオプションを選択します（メニューにチェックマークが付きます）。

3.10.2 測定方法を指定する

プロファイルデータの測定時に、関数呼び出しをトレースするかどうかを指定できます。関数呼び出しをトレースすると、ユーザプログラム実行時の関数呼び出し関係をツリー形式で表示できるようになります。関数呼び出しをトレースしないと、関数呼び出し関係を表示できませんが、プロファイルデータの測定時間を短縮することができます。

関数呼び出しをトレースしないようにするためには、[プロファイル]ウィンドウのポップアップメニューで[関数呼び出しをトレースしない]を選択します（メニューにチェックマークが付きます）。

また、OSによるタスクスイッチなど、通常の方法以外で関数を呼び出しているプログラムの場合、関数呼び出しを正しく表示できない場合がありますので、関数呼び出しをトレースせずにプロファイルデータを測定してください。

3.10.3 ユーザプログラムを実行し結果を確認する

ユーザプログラムを実行し、停止すると[プロファイル]ウィンドウに測定結果を表示します。

[プロファイル]ウィンドウには、[List]シートと[Tree]シートがあります。

3.10.4 List シート

Function/Variable	F/V	Address	Size	Times	Cycles
PowerON_Reset()	F	00000400	H'00000024	1	442
Sample::change(long *)	F	000020c6	H'00000054	1	4695
Sample::sort(long *)	F	00002046	H'00000080	1	15632
Sample::Sample()	F	00002000	H'00000046	1	603
00001334	F	00001334	H'00000000	1	544
_malloc	F	0000128E	H'000000A6	1	1203
_malloc	F	0000128A	H'00000004	1	147
_free	F	000011CE	H'000000Bc	1	318
\$_DIVL\$3	F	000011B6	H'00000000	10	2380
_srand	F	000011A8	H'0000000E	1	135
_rand	F	0000117A	H'0000002E	10	4120
_CALL_INIT	F	0000114E	H'0000001E	1	268
operator new(unsigned long)	F	00001110	H'0000003E	1	331
_INIT SCT	F	000010B8	H'00000000	1	88074
tutorial()	F	0000102E	H'00000088	1	3151
_main	F	0000102A	H'00000004	1	63
_sbrk	F	00001000	H'0000002A	1	268

図3.41 プロファイルウィンドウ (List シート)

3 デバッグ

本ウィンドウは、関数とグローバル変数のアドレス、サイズ、関数呼び出し回数および実行サイクル数を表示します。

コラムヘッダをクリックすると、アルファベットまたは数値の昇降順にソートして表示します。

[Function/Variable]列または[Address]列をダブルクリックすると、該当するアドレスに対応したソースプログラムを表示します。ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このポップアップメニューは「3.10.5 Tree シート」を参照してください。

【注】 1. プロファイル機能の注意事項については「5.9 プロファイル機能について」を参照してください。

3.10.5 Tree シート

Function	Address	Size	Stack Size	Times	Cycles
C:\Hew3\Tools\Renesas\DebugCo...					
PowerON_Reset {}	00000400	H'00000024	H'00000000	1	442
__INIT3CT	000010B8	H'00000000	H'00000000	1	88074
_srand	000011A8	H'0000000E	H'00000000	1	135
__CALL_INIT	0000114E	H'0000001E	H'00000000	1	268
main	0000102A	H'00000004	H'00000000	1	63
tutorial {}	0000102E	H'00000088	H'00000000	1	3151
Sample::Sample {}	00002000	H'00000046	H'00000000	1	603
Sample::change (long *)	000020C6	H'00000054	H'00000000	1	4695
Sample::sort (long *)	00002046	H'00000080	H'00000000	1	15632
_rand	0000117A	H'0000002E	H'00000000	10	4120

図3.42 プロファイルウィンドウ (Tree シート)

本ウィンドウは、関数の呼び出し関係をツリー構造で表示します。また、各関数のアドレス、サイズ、スタックサイズ、関数呼び出し回数および実行サイクル数を表示します。スタックサイズ、関数呼び出し回数は、実際の関数を呼び出した経路における値を表示します。

[Tree]シートは、[プロファイル]ウィンドウのポップアップメニューで[関数呼び出しをトレースしない]をチェックしていない時のみ有効です。

[Function]列の関数をダブルクリックすると、ツリー構造を拡張または収縮表示します。また、'+/-'キーでも拡張/収縮表示することができます。[Address]列をダブルクリックすると、該当するアドレスに対応したソースプログラムを表示します。

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このメニューは以下のオプションを含みます。

3.10.6 ソースファイル表示

選択している行の該当アドレスに対応したソースプログラムを表示します。

3.10.7 チャート表示

選択している行の関数に着目した[プロファイル-チャートウィンドウ]を表示します。

3.10.8 有効

プロファイルデータ収集の有効/無効を切り替えます。プロファイルデータ測定が有効のとき、メニューテキストの左にチェックマークを表示します。

3.10.9 関数呼び出しをトレースしない

本メニューをチェックすると、プロファイルデータ測定時に関数呼び出しをトレースしません。たとえば、OS のタスクスイッチのように通常の方法以外で関数が呼び出されるプログラムのデータを測定する場合に使用します。

[プロファイル]ウィンドウの[Tree]シートで関数呼び出し関係を表示するためには、本メニューをチェックせずにプロファイルデータを測定してください。

また、測定結果のプロファイル情報ファイルを使用して、最適化リンケージエディタによる最適化を行う場合も、本メニューをチェックしないでください。

3.10.10 検索...

[Function]列の文字列を検索する[テキスト検索]ダイアログボックスを表示します。検索したい文字列をエディットボックスに入力し、[次を検索]ボタンまたは、ENTER キーを入力すると、検索を開始します。

3.10.11 データ検索...

[データ検索]ダイアログボックスを表示します。

[カラム]コンボボックスで検索カラムを、[検索データ]グループで検索方向を設定し、[次を検索]ボタンまたは、ENTER キーを入力すると、検索を開始します。また、連続して[次を検索]ボタンまたは、ENTER キーを入力すると、次に大きいデータ(最小値の場合は小さいデータ)を検索します。

3.10.12 データクリア

関数呼び出し回数のカウントおよびプロファイルデータをクリアします。[プロファイル]ウィンドウの[List]シートおよび[プロファイルチャート]ウィンドウのデータもクリアします。

3.10.13 プロファイル情報の保存...

[プロファイル情報の保存]ダイアログボックスを表示します。プロファイル結果をプロファイル情報ファイル(拡張子は".pro")に保存します。最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことができます。プロファイル情報を使用した最適化についての詳細は、最適化リンケージエディタのマニュアルを参照してください。

【注】 [関数呼び出しをトレースしない]メニューをチェックして測定した結果のプロファイル情報では、最適化リンケージエディタによる最適化は行えません。

3.10.14 テキスト形式で保存...

[プロファイルデータをテキスト形式で保存]ダイアログボックスを表示します。表示している状態をテキストファイルに保存します。

3 デバッグ

3.10.15 表示設定

このメニューには下記サブメニューがあります。（以下の説明には[List]シートのみメニューも含まれます）

(1) 関数と変数を表示

[Function/Variable]列で、関数およびグローバル変数の両方表示します。

(2) 関数を表示

[Function/Variable]列で、関数のみを表示します。

(3) 変数を表示

[Function/Variable]列で、グローバル変数のみを表示します。

(4) 未実行関数を表示しない

実行した関数のみ表示することができます。最適化リンカージェネレータが出力するスタック使用量情報ファイル(拡張子:sni)がロードモジュールと同一ディレクトリに存在しない場合、このチェックボックスの設定にかかわらず、実行関数のみ表示します。

(5) 子関数の実行結果を含んで表示

表示するプロファイルデータに、関数内で呼び出した子関数のプロファイルデータを含めるかどうかを設定します。

3.10.16 プロパティ...

E6000H エミュレータではサポートしていません。

3.11 プロファイル-チャートウィンドウ

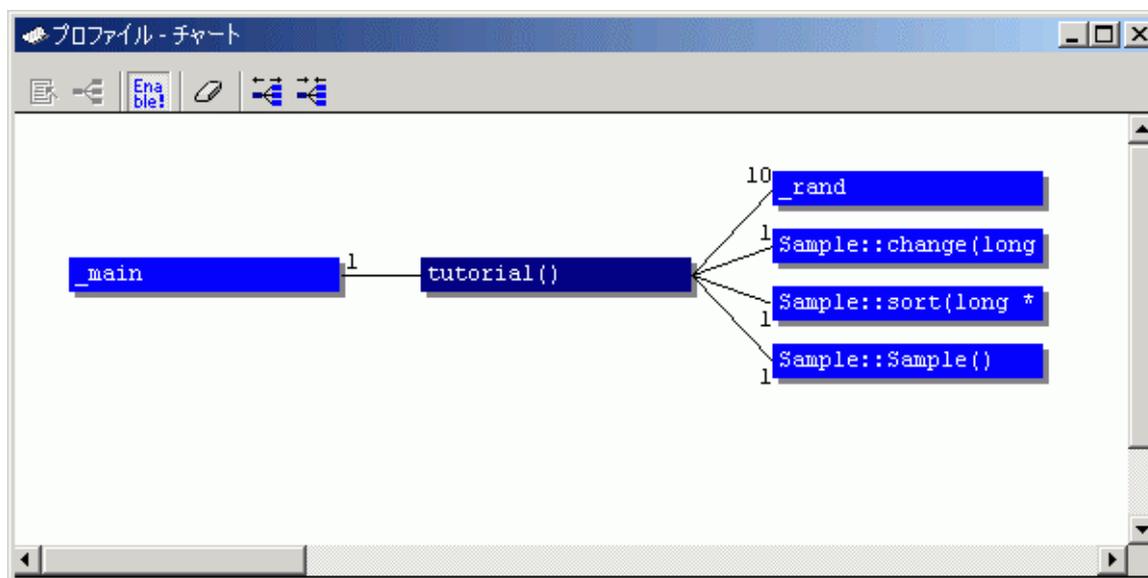


図3.43 プロファイル-チャートウィンドウ

ある関数に着目した呼び出し関係を表示します。[プロファイル-チャート]ウィンドウは、[プロファイル]ウィンドウの[List]シートまたは[Tree]シートから開きます。[プロファイル]ウィンドウの[List]シートまたは[Tree]シートで選択した関数に着目した呼び出し関係を表示します。着目した関数を中心に、左側に呼び出し元関数、右側に呼び出し先関数を表示します。呼び出し元関数および呼び出し先関数横の数値は、呼び出し回数を示します。

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このメニューは以下のオプションを

含みます。

3.11.1 ソースファイル表示

マウスの右ボタンをクリックしたときの位置にある関数の該当アドレスに対応したソースプログラムまたは逆アセンブルを表示します。マウスの右ボタンをクリックしたときの位置が関数ではない場合、このメニューオプションはグレー表示となります。

3.11.2 チャート表示

マウスの右ボタンをクリックしたときの位置にある関数に着目した[プロファイル-チャート]ウィンドウを表示します。マウスの右ボタンをクリックしたときの位置が関数ではない場合、このメニューオプションはグレー表示となります。

3.11.3 有効

プロファイルデータ収集の有効/無効を切り替えます。プロファイルデータ測定が有効のとき、メニューテキストの左にチェックマークを表示します。

3.11.4 データクリア

関数呼び出し回数のカウントをクリアします。[プロファイル]ウィンドウの[List]シートおよび[Tree]シートのデータもクリアします。

3.11.5 チャートウィンドウを複数開く

[プロファイル-チャート]ウィンドウを表示する際、既に Profile-Chart ウィンドウが開いているとき、別のウィンドウを開くか同ウィンドウに表示するかを設定します。メニューテキストの左にチェックマークを表示していれば、別のウィンドウを開きません。

3.11.6 プロファイル情報の保存...

[プロファイル情報の保存]ダイアログボックスを表示します。プロファイル結果をプロファイル情報ファイル(拡張子は".pro")に保存します。最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことができます。プロファイル情報を使用した最適化についての詳細は、最適化リンケージエディタのマニュアルを参照してください。

3.11.7 拡大

各関数の間隔を広げて表示します。また、'+ ' キーでも広げて表示することができます。

3.11.8 縮小

各関数の間隔を縮めて表示します。また、'- ' キーでも縮めて表示することができます。

4. チュートリアル

4.1 はじめに

E6000H エミュレータの主な機能を紹介するために、チュートリアルプログラムを提供しています。このプログラムを用いて説明します。

このチュートリアルプログラムは、C++言語で書かれており、10個のランダムデータを昇順 / 降順にソートします。

チュートリアルプログラムでは、以下の処理を行います。

main 関数ではソート処理を繰り返し実行するため tutorial 関数の呼び出しを繰り返します。

tutorial 関数ではソートするランダムデータを生成し、sort 関数および change 関数を順に呼び出します。

sort 関数では tutorial 関数で生成したランダムデータを格納した配列を入力し、昇順にソートします。

change 関数では sort 関数で昇順にソートした配列を入力し、降順にソートします。

チュートリアルプログラムは、tutorial.cpp ファイルで提供しています。コンパイルされたロードモジュールは、Tutorial.abs ファイルとして Dwarf2 フォーマットで提供しています。

- 【注】
1. 再コンパイルを行った場合、本章で説明しているアドレスとずれることがあります。
 2. 本章は、一般的な E6000H エミュレータの使用例です。各製品の仕様については、「3 デバッグ」またはオンラインヘルプを参照してください。
 3. 各製品に添付される Tutorial.abs の動作アドレスは、製品によって異なります。本章で使用するアドレスについては、対応するソース行より各製品上で該当のアドレスを確認し、適宜読み替えて操作してください。
 4. 本チュートリアルでは SH7058 E6000H エミュレータを例に説明しています。ファイルのパス情報や、図の画面は各製品により異なりますので適宜読み替えて操作してください。

4.2 High-performance Embedded Workshop の起動

「2.1.3 既存のワークスペースを指定する場合」の手順に従ってワークスペースを開きます。

ディレクトリは以下を指定してください。

OS インストールドライブ\Workspace\Tutorial\E6000H\7058

【注】 1. ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その場合は以下のディレクトリを指定してください。

High-performance Embedded Workshop インストール先ディレクトリ\Tools\Renesas\DebugComp\Platform\E6000H\7058\Tutorial

2. バス情報は各製品により異なりますので「\7058」の部分を適宜変更してください。

ファイルは以下を指定してください。

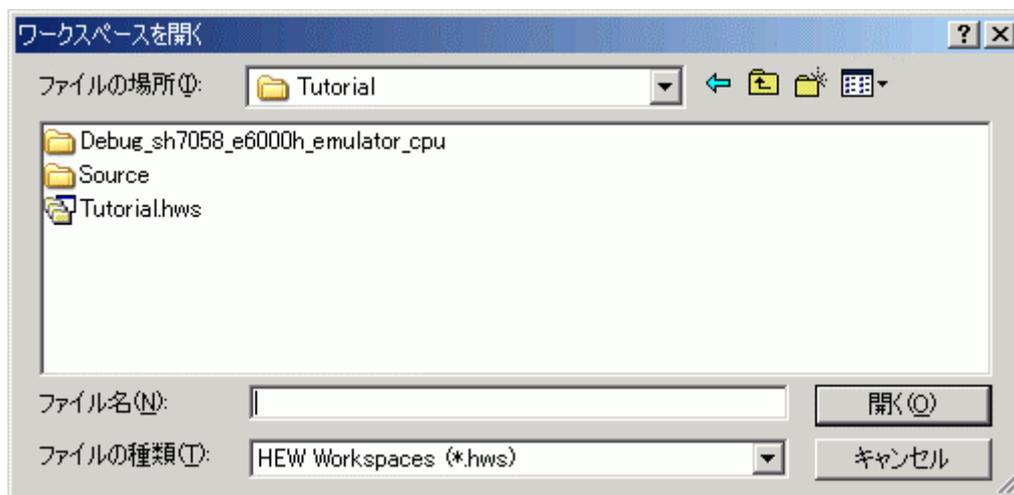


図4.1 ワークスペースを開くダイアログボックス

4.3 チュートリアルプログラムのダウンロード

4.3.1 チュートリアルプログラムをダウンロードする

デバッグしたいオブジェクトプログラムをダウンロードできます。

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

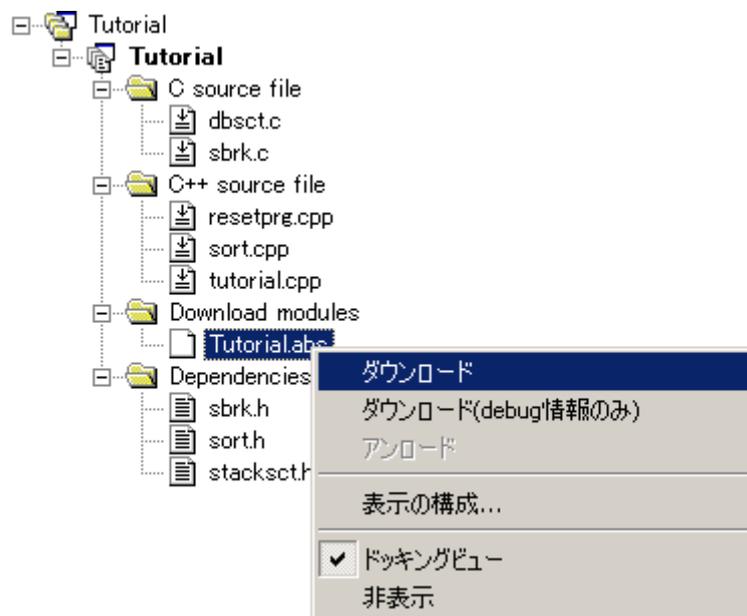


図4.2 チュートリアルプログラムのダウンロード

4 チュートリアル

4.3.2 ソースプログラムを表示する

High-performance Embedded Workshop では、ソースレベルでプログラムをデバッグできます。

[C++ source file]の[tutorial.cpp]をダブルクリックします。

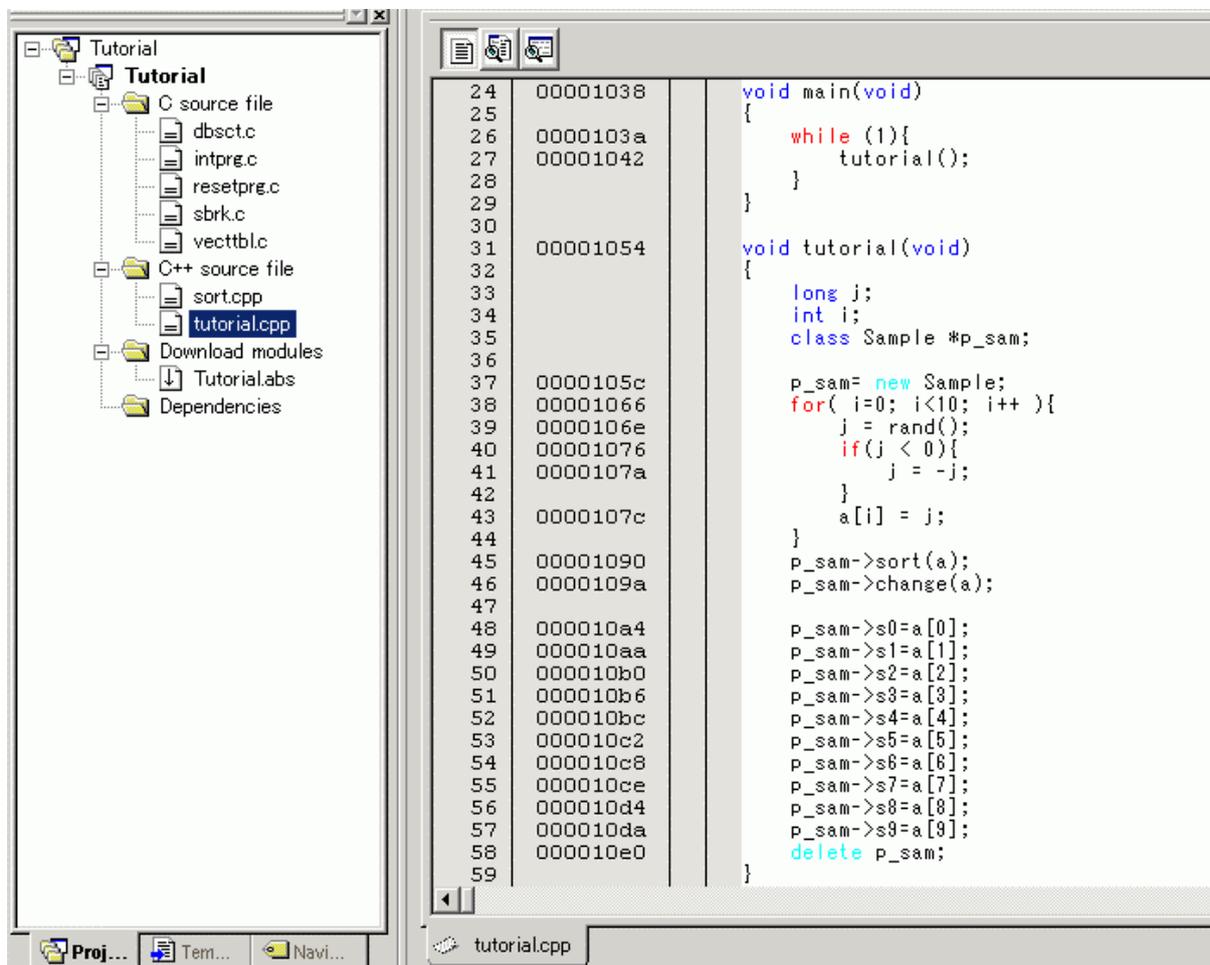


図4.3 [エディタ]ウィンドウ (ソースプログラムの表示)

必要であれば、見やすいフォントとサイズに変更することも可能です。変更方法については High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

[エディタ]ウィンドウは、最初はプログラムの先頭を示しますが、スクロールバーを使って他の部分を見ることができます。

4.4 ソフトウェアブレイクポイントの設定

簡単なデバッグ機能の1つにソフトウェアブレイクポイントがあります。

[エディタ]ウィンドウにおいて、ソフトウェアブレイクポイントを簡単に設定できます。たとえば、`sort` 関数のコール箇所にソフトウェアブレイクポイントを設定します。

`sort` 関数コールを含む行の[S/W ブレイクポイント]カラムをダブルクリックしてください。

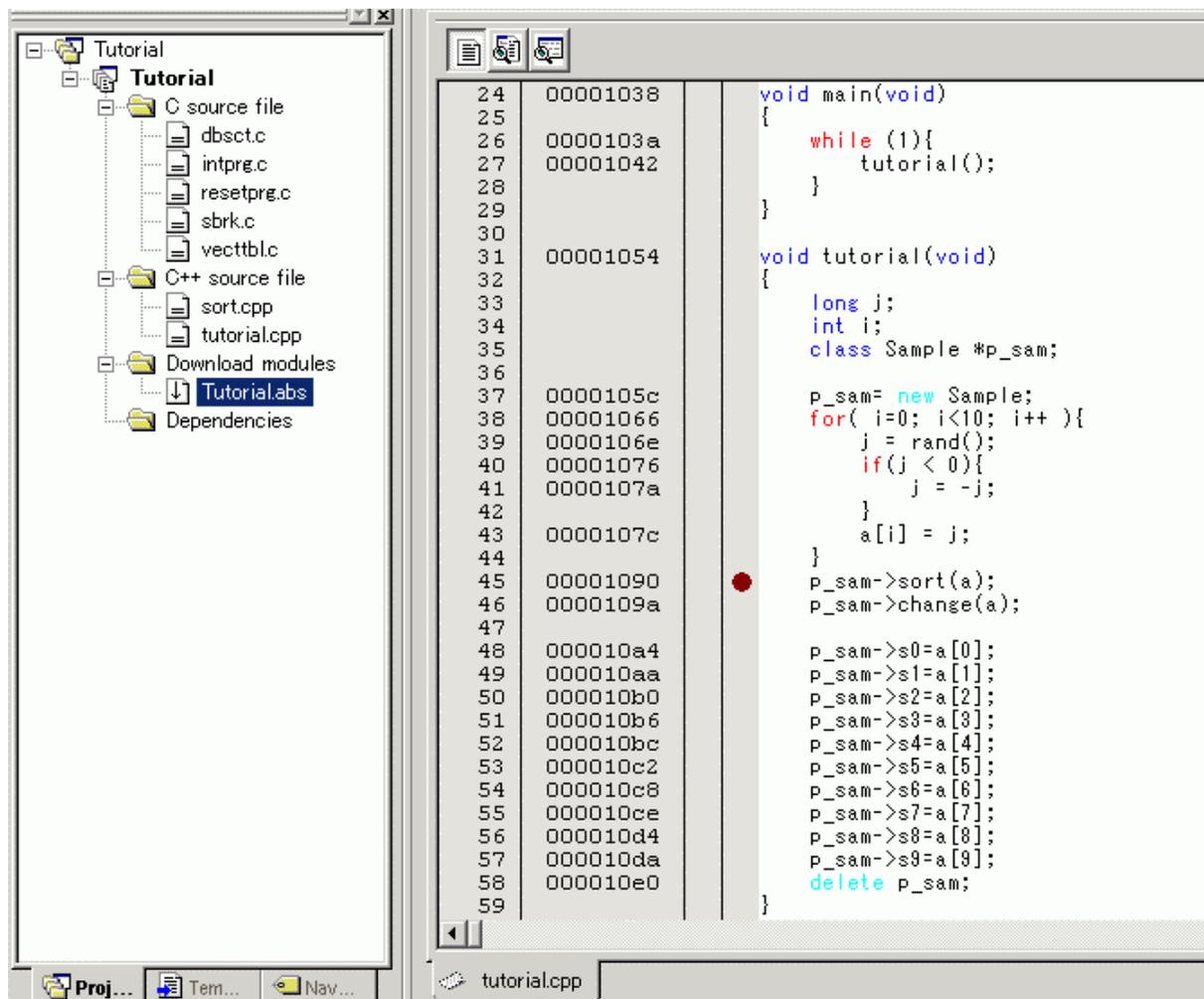


図4.4 [エディタ]ウィンドウ (ソフトウェアブレイクポイントの設定)

`sort` 関数を含む行に"●"と表示されます。この表示によりソフトウェアブレイクポイントが設定されたことを示します。

4.6 プログラムの実行

プログラムの実行方法について説明します。

プログラムを実行する場合は、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。



図4.7 実行ボタン

実行を開始すると、ステータスバーに現在のアドレスバスの値、およびターゲットマイコンの動作状態を表示します。

プログラムはソフトウェアブレークポイントを設定したところまで実行されます。プログラムが停止した位置を示すために[エディタ]ウィンドウの[S/W ブレークポイント]カラムに矢印が表示されます。また、[Break = Software Break]メッセージがステータスバーに表示されます。

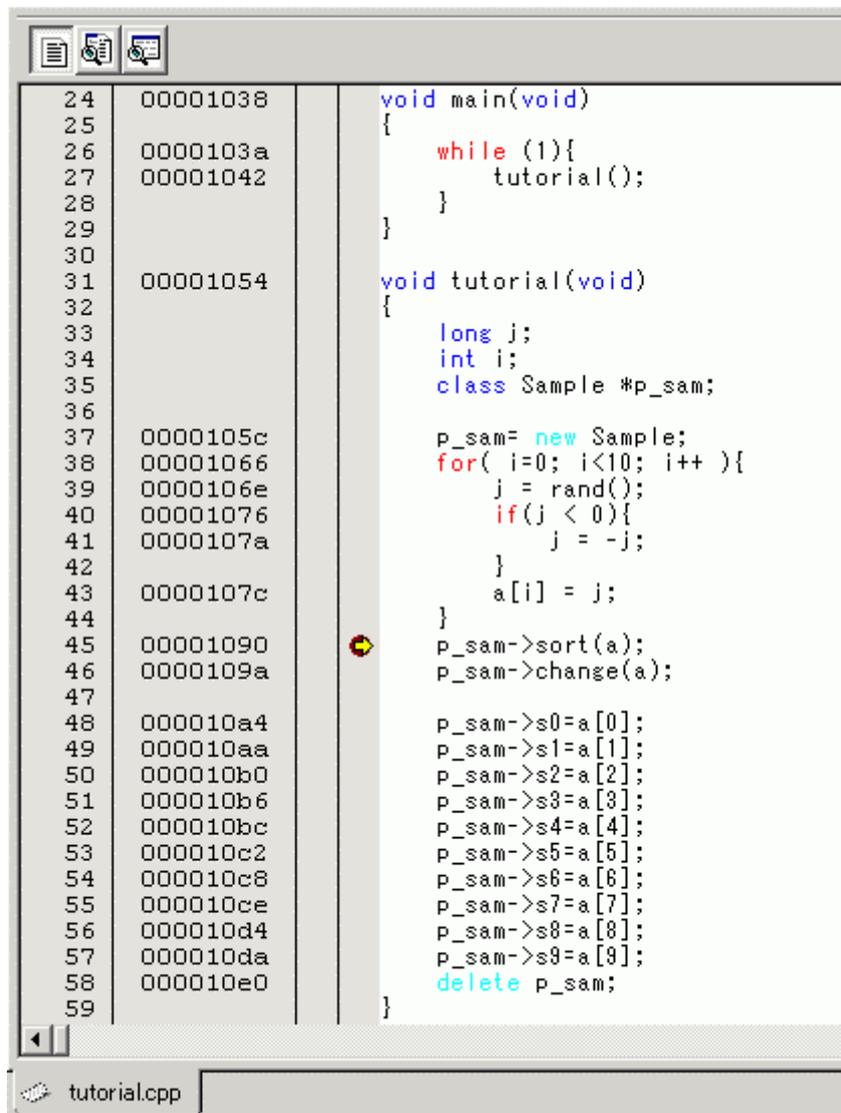
- 【注】** 1. ブレーク後にソースファイルを表示する際に、ソースファイルバスを問い合わせる場合があります。
ソースファイルの場所は以下です。

OS インストールドライブ\Workspace\Tutorial\E6000H\7058\Source

ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その場合は以下のディレクトリを指定してください。

High-performance Embedded Workshop インストール先ディレクトリ
\Tools\Renesas\DebugComp\Platform\E6000H\7058\Tutorial\Source

2. バス情報は各製品により異なりますので「\7058」の部分を適宜変更してください。



```
24 00001038 void main(void)
25
26 0000103a     while (1){
27 00001042         tutorial();
28
29
30
31 00001054 void tutorial(void)
32 {
33     long j;
34     int i;
35     class Sample *p_sam;
36
37 0000105c     p_sam= new Sample;
38 00001066     for( i=0; i<10; i++ ){
39 0000106e         j = rand();
40 00001076         if(j < 0){
41 0000107a             j = -j;
42
43 0000107c             a[i] = j;
44         }
45 00001090     p_sam->sort(a);
46 0000109a     p_sam->change(a);
47
48 000010a4     p_sam->s0=a[0];
49 000010aa     p_sam->s1=a[1];
50 000010b0     p_sam->s2=a[2];
51 000010b6     p_sam->s3=a[3];
52 000010bc     p_sam->s4=a[4];
53 000010c2     p_sam->s5=a[5];
54 000010c8     p_sam->s6=a[6];
55 000010ce     p_sam->s7=a[7];
56 000010d4     p_sam->s8=a[8];
57 000010da     p_sam->s9=a[9];
58 000010e0     delete p_sam;
59 }
```

図4.8 [エディタ]ウィンドウ (ブレーク状態)

[ステータス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示 -> CPU -> ステータス]を選択するか、[ステータスの表示]ツールバーボタン  をクリックしてください。

[ステータス]ウィンドウが表示されますので、[Platform]シートを開いて Cause of last break のステータスを確認してください。

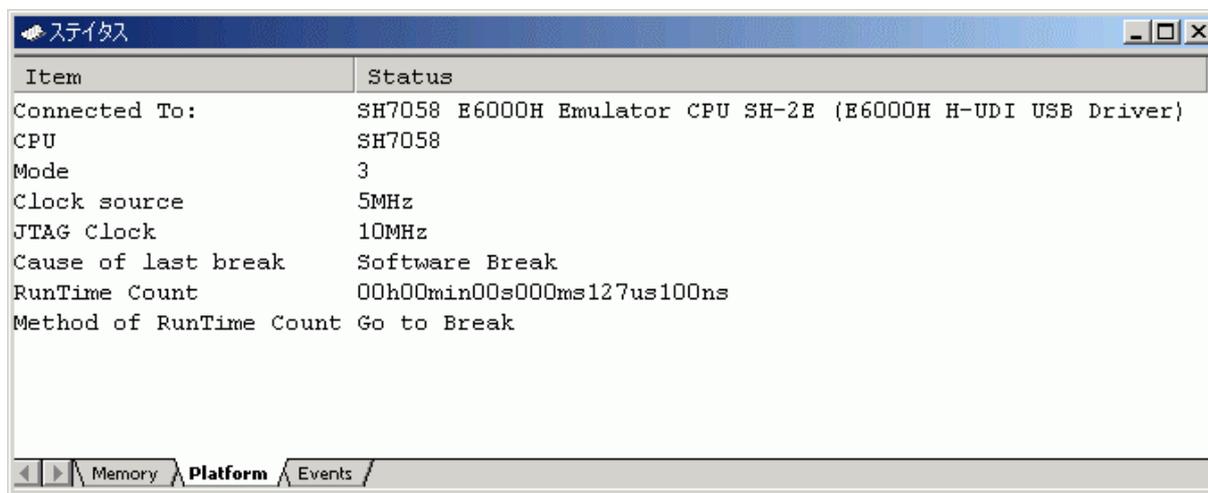


図4.9 ステータスウィンドウ

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

4.7 ブレークポイントの確認

設定したすべてのブレークポイントは、[イベントポイント]ウィンドウで確認することができます。

[表示 -> コード -> イベントポイント]を選択するか、[イベントポイント]ツールバーボタン  をクリックすると、[イベントポイント]ウィンドウが表示されます。[Software]シートを開きます。

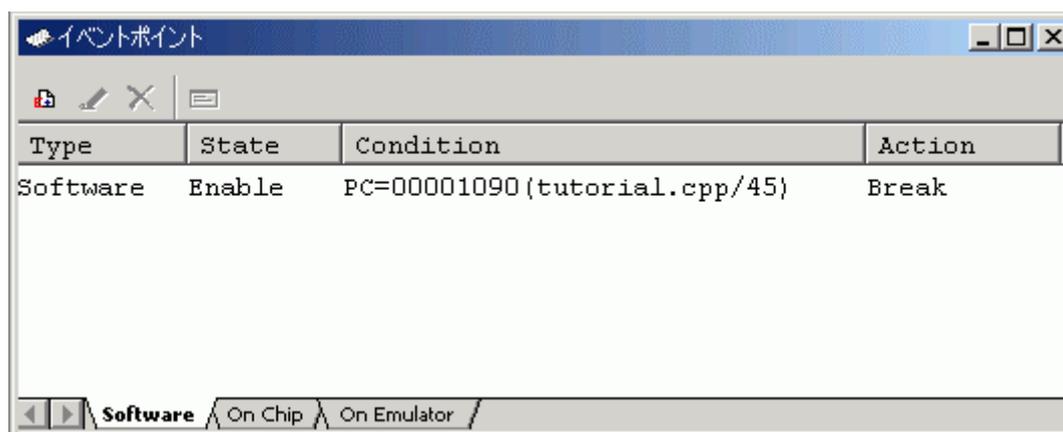


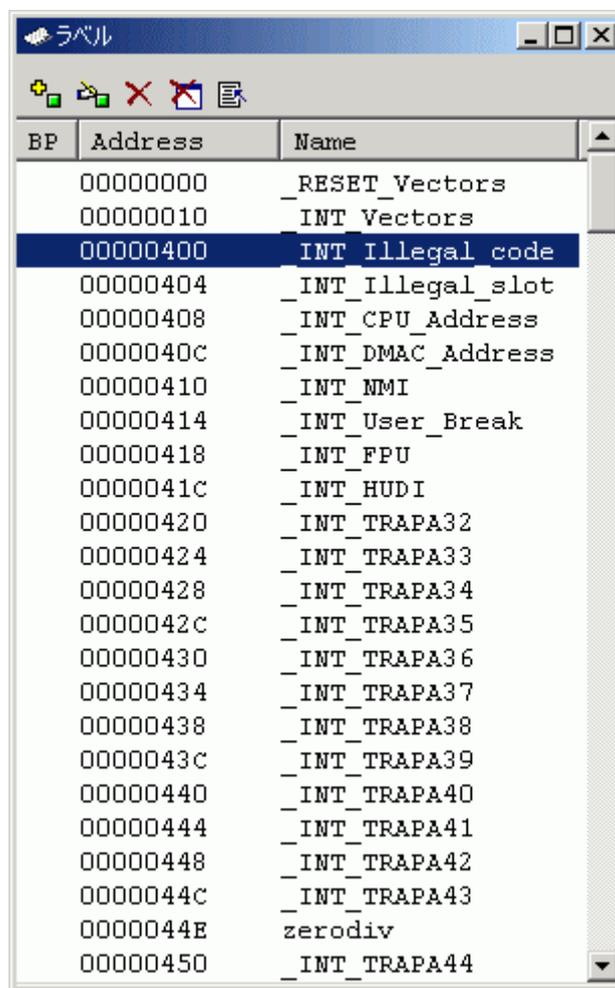
図4.10 イベントポイントウィンドウ

マウスの右ボタンで[イベントポイント]ウィンドウをクリックすると開くポップアップメニューにより、ブレークポイントの設定 / 変更、新しいブレークポイントの定義、およびブレークポイントの削除、有効 / 無効の選択ができます。

4.8 シンボルの参照

[ラベル]ウィンドウを使ってモジュール内のシンボル情報を表示させることができます。

[表示]メニューから[シンボル]サブメニューを選択し、[ラベル]を選択してください。[ラベル]ウィンドウが表示され、モジュール内のシンボル情報が参照できます。



The screenshot shows a window titled 'ラベル' (Labels) with a toolbar and a table of memory addresses and symbols. The table has three columns: BP, Address, and Name. The row for 'INT_Illegal_code' at address '00000400' is highlighted in blue.

BP	Address	Name
	00000000	_RESET_Vectors
	00000010	_INT_Vectors
	00000400	INT_Illegal_code
	00000404	_INT_Illegal_slot
	00000408	_INT_CPU_Address
	0000040C	_INT_DMxAC_Address
	00000410	_INT_NMI
	00000414	_INT_User_Break
	00000418	_INT_FPU
	0000041C	_INT_HUDI
	00000420	_INT_TRAPA32
	00000424	_INT_TRAPA33
	00000428	_INT_TRAPA34
	0000042C	_INT_TRAPA35
	00000430	_INT_TRAPA36
	00000434	_INT_TRAPA37
	00000438	_INT_TRAPA38
	0000043C	_INT_TRAPA39
	00000440	_INT_TRAPA40
	00000444	_INT_TRAPA41
	00000448	_INT_TRAPA42
	0000044C	_INT_TRAPA43
	0000044E	zerodiv
	00000450	_INT_TRAPA44

図4.11 ラベルウィンドウ

4.9 メモリ内容の確認

ラベル名を指定することによって、ラベルが登録されているメモリの内容を[メモリ]ウィンドウで確認することができます。たとえば、以下のように、バイトサイズで_main に対応するメモリ内容を確認します。

[表示 -> CPU -> メモリ...]を選択するか、[メモリ]ツールバーボタンをクリックして、[表示開始アドレス]ダイアログボックスを開いてください。

[表示開始アドレス]エディットボックスに"_main"を入力してください。



図4.12 表示開始アドレスダイアログボックス

[OK]ボタンをクリックしてください。指定されたメモリ領域を示す[メモリ]ウィンドウが表示されます。

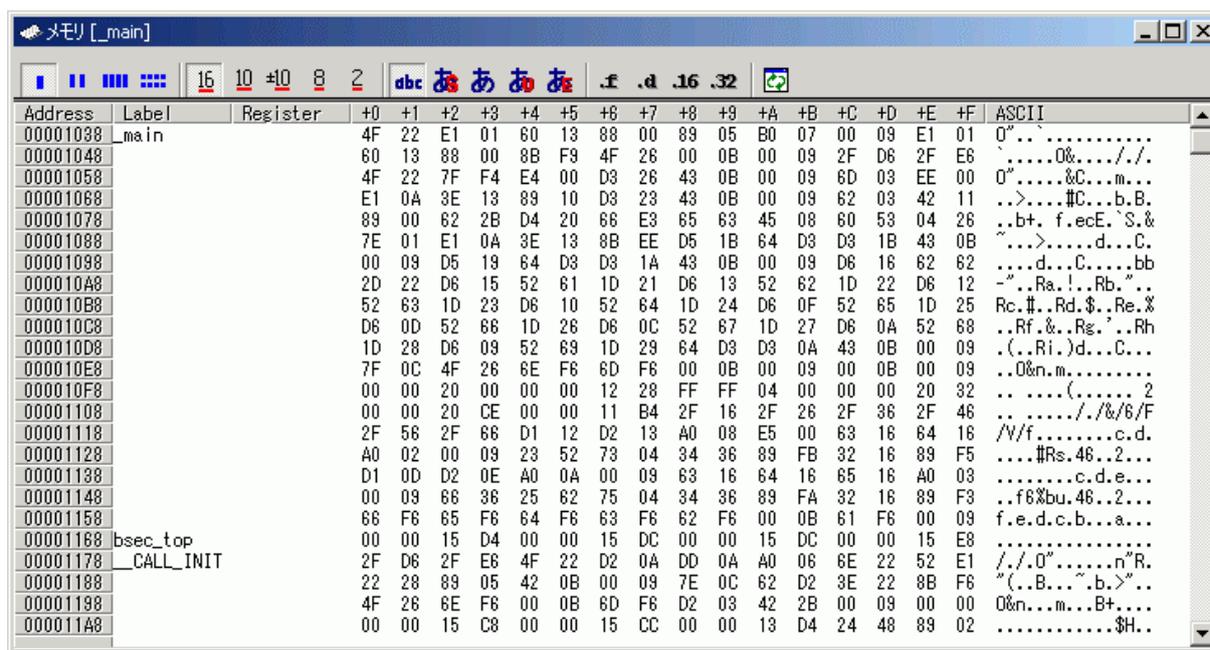


図4.13 メモリウィンドウ

4.10 変数の参照

プログラムをステップ処理するとき、プログラムで使われる変数の値が変化することを確認できます。たとえば、以下の手順で、プログラムの初めに宣言した long 型の配列 a を見ることができます。

[エディタ]ウィンドウに表示されている配列 a の左側をクリックし、カーソルを置いてください。
マウスの右ボタンで[インスタントウォッチ]を選択してください。

以下のダイアログボックスが表示されます。

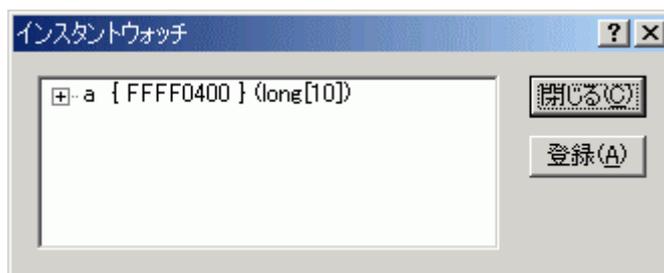


図4.14 インスタントウォッチダイアログボックス

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

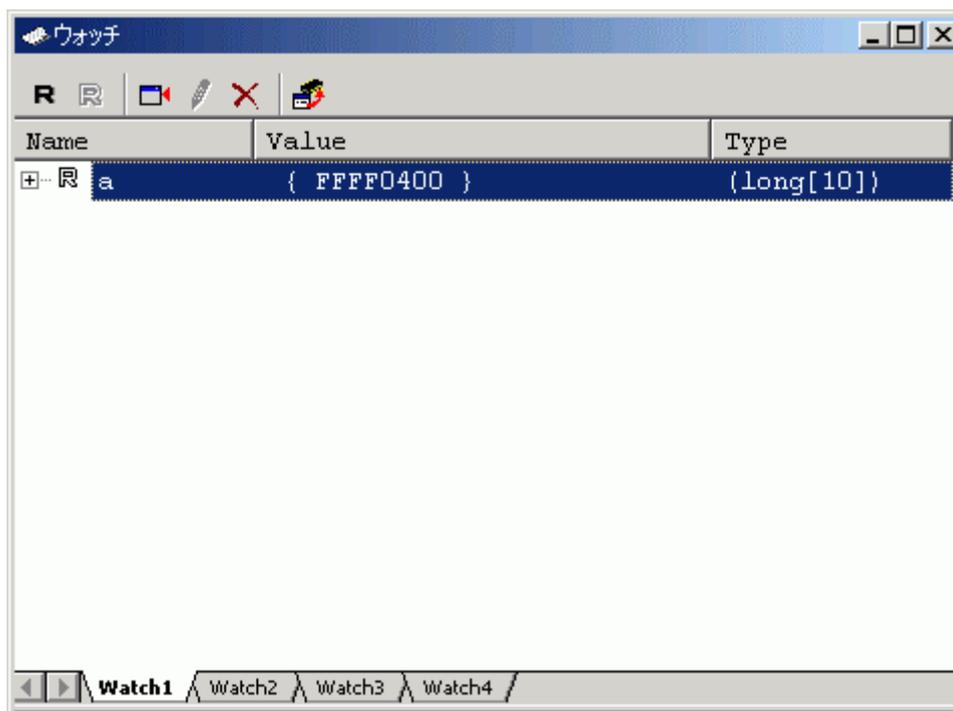


図4.15 ウォッチウィンドウ (配列の表示)

また、変数名を指定して、[ウォッチ]ウィンドウに変数を加えることもできます。

マウスの右ボタンで[ウォッチ]ウィンドウをクリックし、ポップアップメニューから[シンボル登録]を選択してください。

以下のダイアログボックスが表示されます。



図4.16 シンボル登録ダイアログボックス

[変数または式]エディットボックスに変数 `i` を入力し、[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、`int` 型の変数 `i` が表示されます。

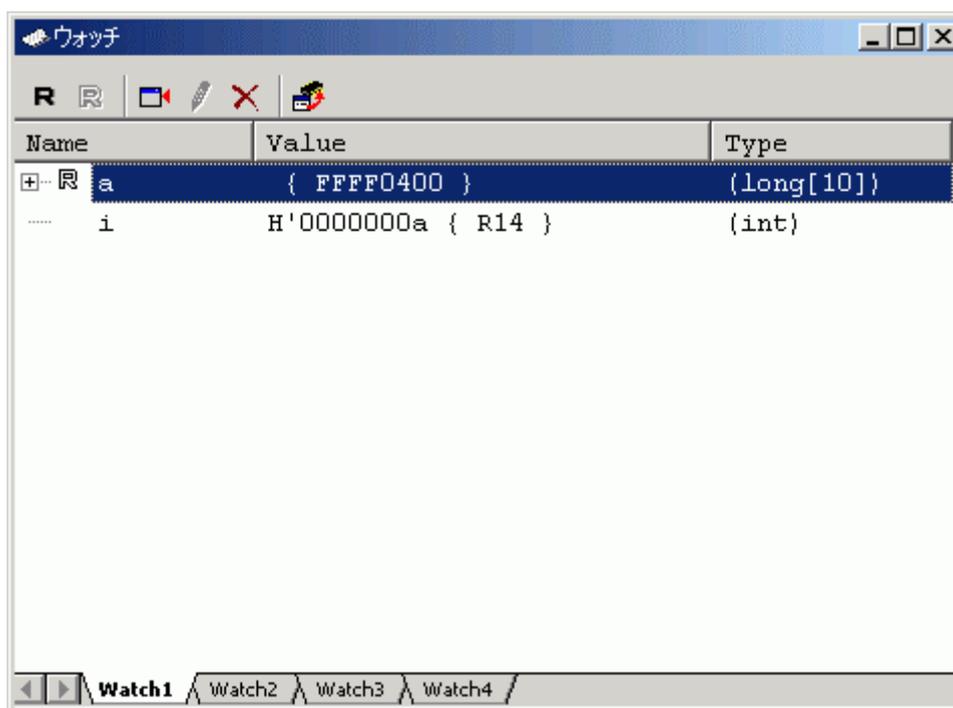


図4.17 ウォッチウィンドウ (変数の表示)

4 チュートリアル

[ウォッチ]ウィンドウの配列 a の左側にある"+"マークをクリックし、配列 a の各要素を参照することができます。

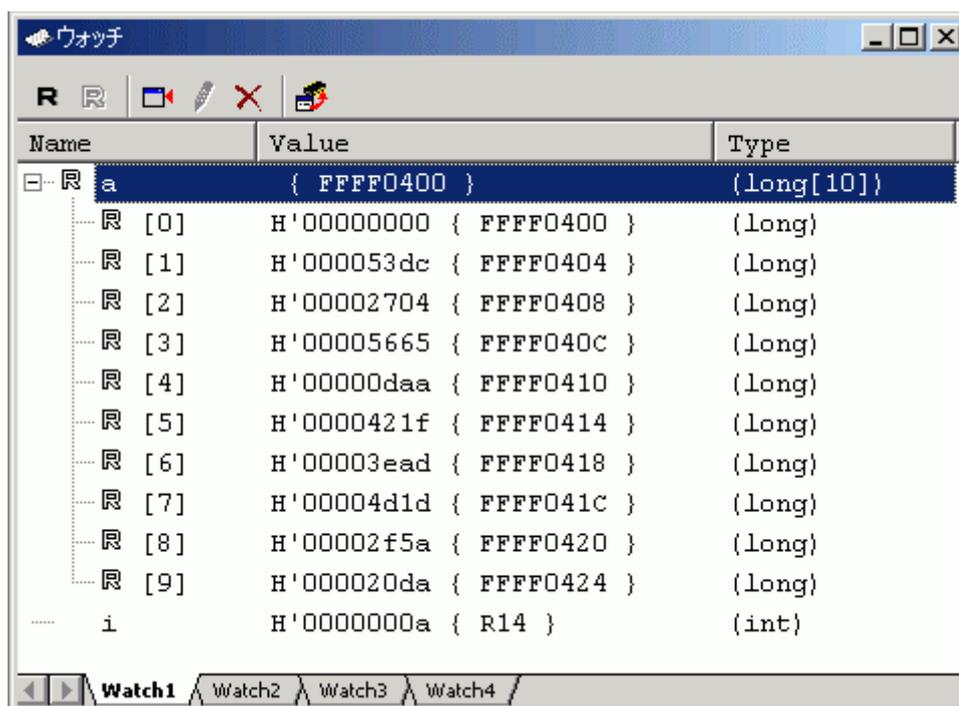


図4.18 ウォッチウィンドウ（配列要素の表示）

4.11 ローカル変数の表示

[ローカル]ウィンドウを使って関数内のローカル変数を表示させることができます。例として、tutorial 関数のローカル変数を調べます。

この関数は、3つのローカル変数 `j`、`i`、`p_sam` を宣言します。

[表示 -> シンボル -> ローカル]を選択するか、[ローカル]ツールバーボタン  をクリックすると、[ローカル]ウィンドウが表示されます。

[ローカル]ウィンドウには、現在のプログラムカウンタ (PC) が指している関数のローカル変数とその値が表示されます。関数内にローカル変数が存在しない場合、[ローカル]ウィンドウに何も表示されません。

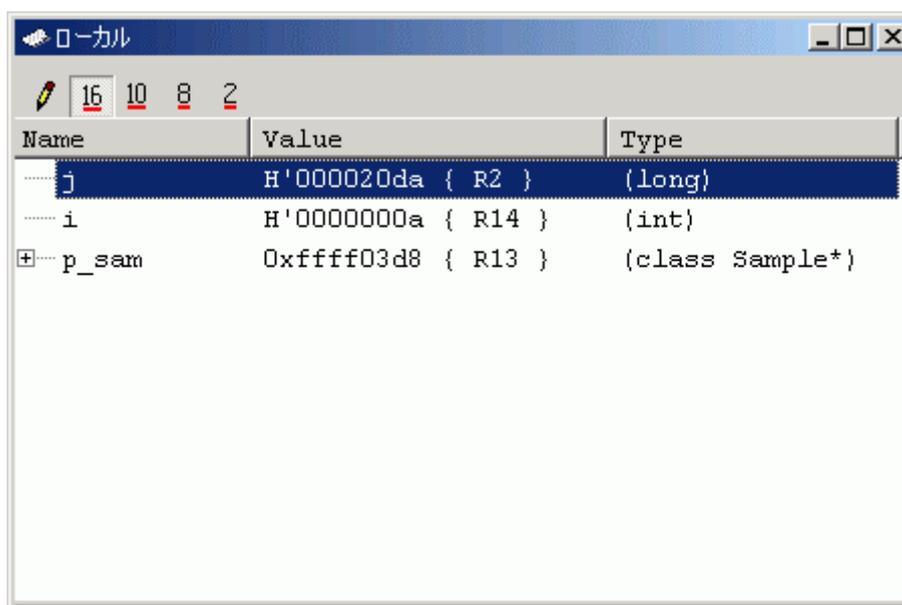


図4.19 ローカルウィンドウ

[ローカル]ウィンドウのクラスインスタンス `p_sam` の左側にある "+" マークをクリックし、クラスインスタンス `p_sam` の構成要素を表示させてください。

`sort` 関数実行前と実行後のクラスインスタンス `p_sam` 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。

4.12 プログラムのステップ実行

High-performance Embedded Workshop は、プログラムのデバッグに有効な各種のステップコマンドを備えています。

表4.1 ステップオプション

項番	コマンド	説明
1	ステップイン	各ステートメントを実行します (関数内のステートメントを含む)。
2	ステップオーバ	関数コールを1ステップとして、ステップ実行します。
3	ステップアウト	関数を抜け出し、関数を呼び出したプログラムの次のステートメントで停止します。
4	ステップ...	指定した速度で指定回数分ステップ実行します。

4.12.1 ステップインコマンドの実行

[ステップイン]コマンドはコール関数の中に入り、コール関数の先頭のステートメントで停止します。

sort 関数の中に入るために、[デバッグ]メニューから[ステップイン]を選択するか、またはツールバーの[ステップイン]ボタンをクリックしてください。



図4.20 ステップインボタン

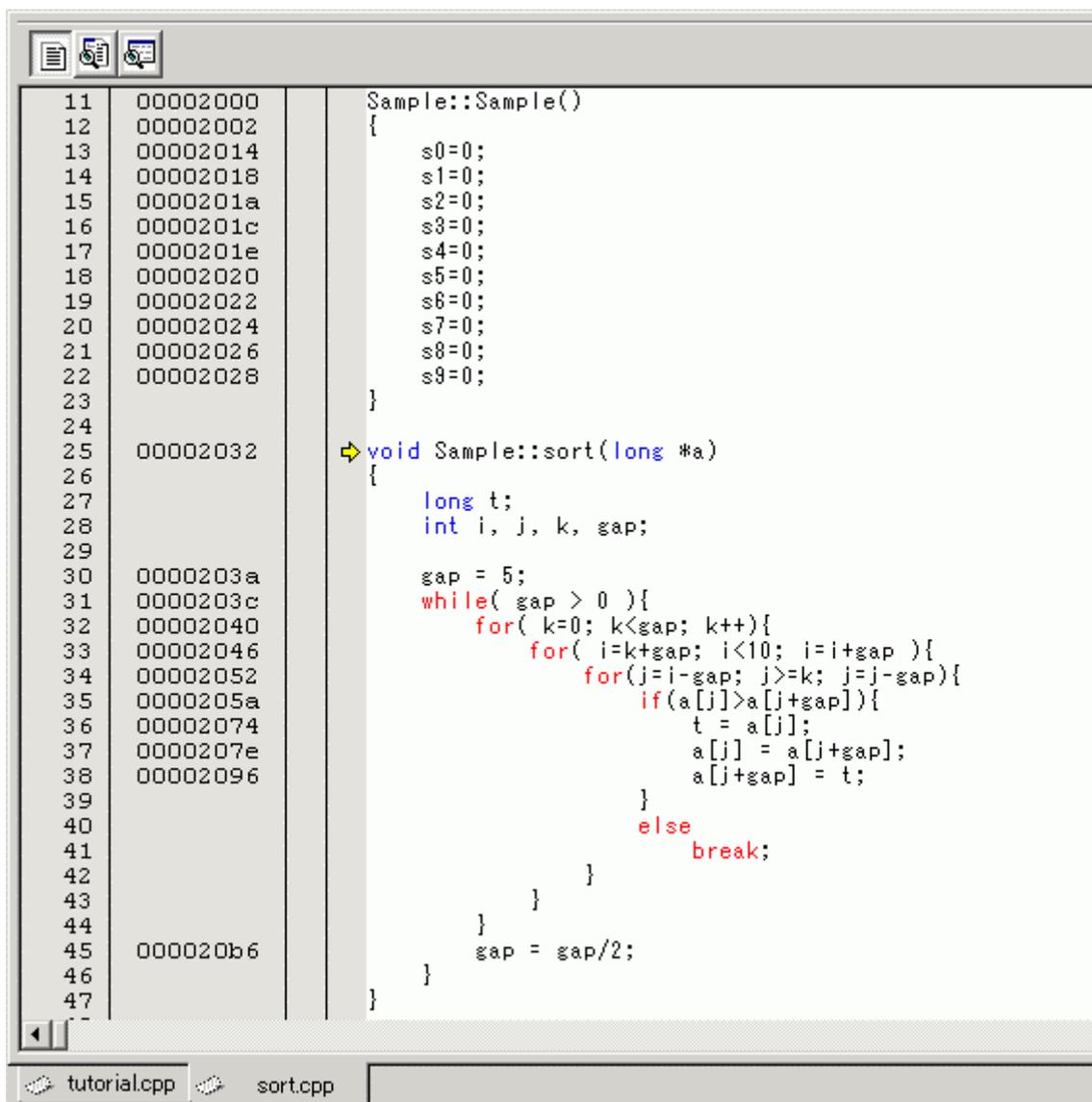


図4.21 [エディタ]ウィンドウ (ステップイン)

[エディタ]ウィンドウの強調表示が、sort 関数の先頭のステートメントに移動します。

4.12.2 ステップアウトコマンドの実行

[ステップアウト]コマンドはコール関数の中から抜け出し、コール元プログラムの次のステートメントで停止します。

sort 関数の中から抜け出すために、[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。



図4.22 ステップアウトボタン

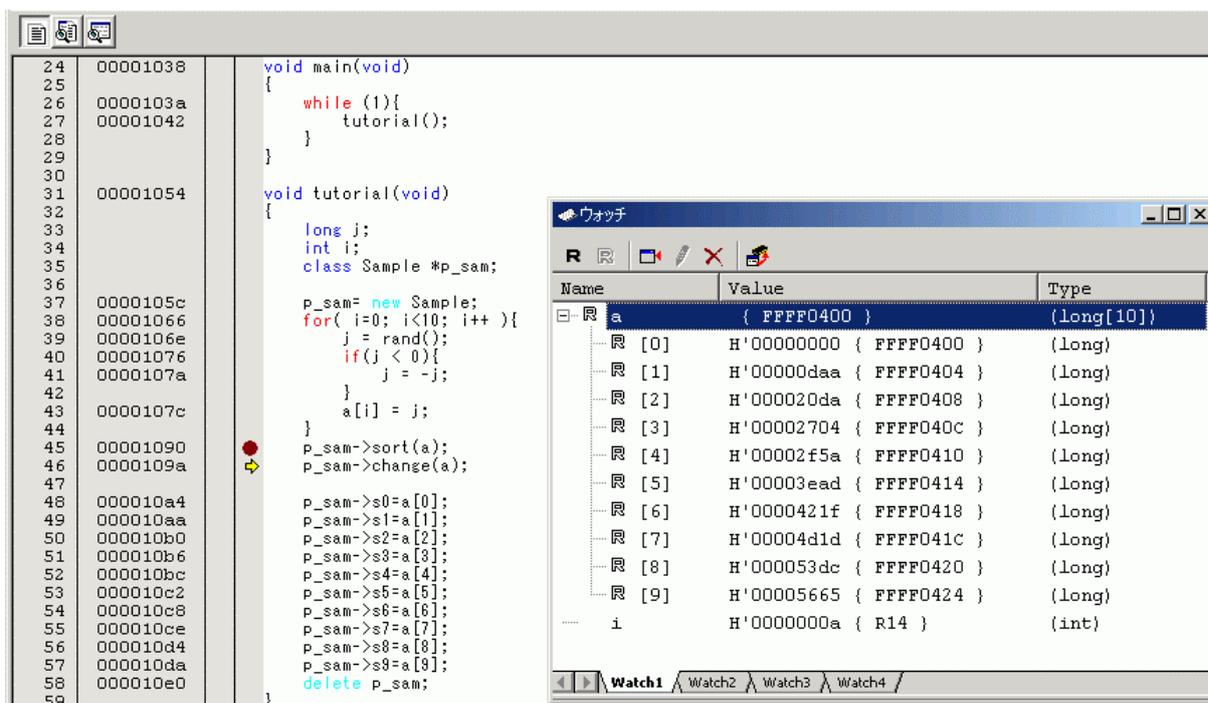


図4.23 High-performance Embedded Workshop ウィンドウ (ステップアウト)

[ウォッチ]ウィンドウに表示された変数 a のデータが昇順にソートされます。

4 チュートリアル

4.12.3 ステップオーバーコマンドの実行

[ステップオーバー]コマンドは関数コールを1ステップとして実行して、メインプログラムの次のステートメントで停止します。

change 関数中のステートメントを一度にステップ実行するために、[デバッグ]メニューから[ステップオーバー]を選択するか、またはツールバーの[ステップオーバー]ボタンをクリックしてください。



図4.24 ステップオーバーボタン

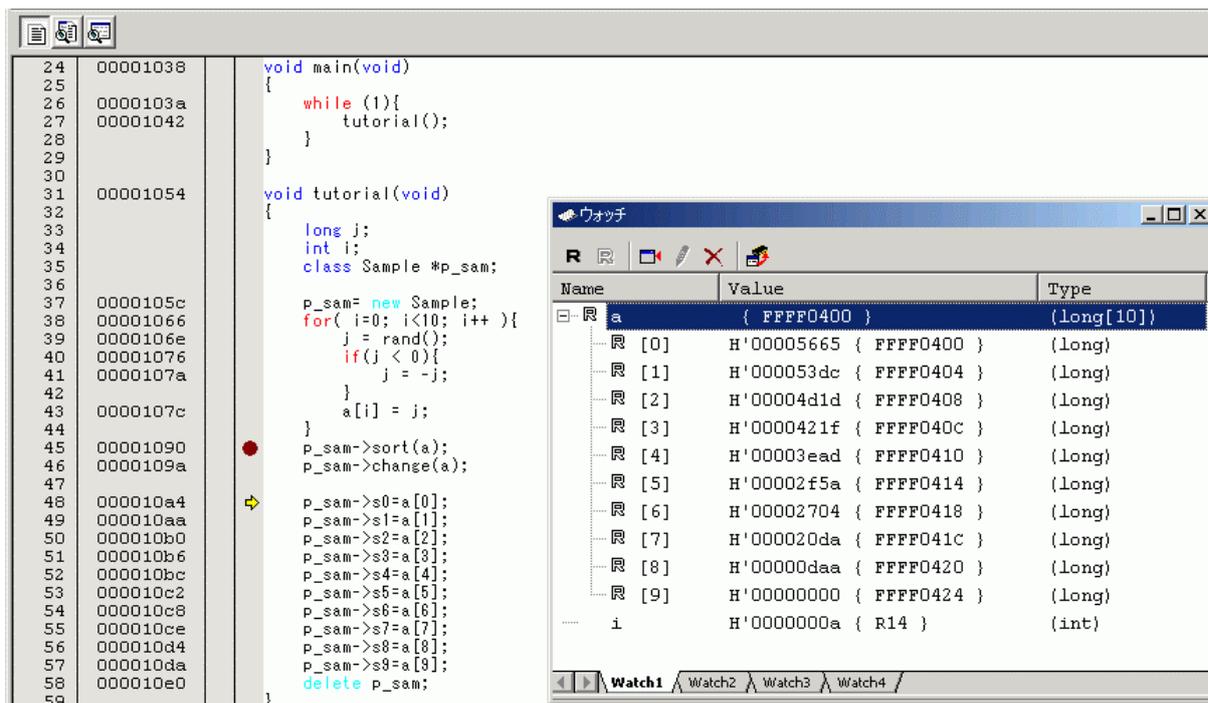


図4.25 High-performance Embedded Workshop ウィンドウ (ステップオーバー)

[ウォッチ]ウィンドウに表示された変数 a のデータが降順にソートされます。

4.13 プログラムの強制ブレーク

High-performance Embedded Workshop は、プログラムを強制的にブレークすることができます。

ブレークポイントをすべて解除してください。

tutorial 関数の残り部分を実行するために、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。



図4.26 実行ボタン

プログラムは無限ループ処理を実行していますので、強制ブレークするために、[デバッグ]メニューから[プログラムの停止]を選択するか、ツールバー上の[停止]ボタンを実行してください。



図4.27 停止ボタン

4.14 ターゲットマイコンのリセット

ターゲットマイコンをリセットすることにより、内蔵 I/O レジスタの初期化、およびリセットベクタに設定されたアドレスを PC に設定します。

ターゲットマイコンのリセットを行うには、[デバッグ]メニューから[CPU のリセット]を選択するか、ツールバー上の[CPU リセット]ボタンを選択してください。



図4.28 CPU リセットボタン

また、リセット直後からプログラムを実行する場合には、[デバッグ]メニューから[リセット後実行]を選択するか、ツールバー上の[リセット後実行]ボタンを選択してください。



図4.29 リセット後実行ボタン

【注】 本チュートリアルプログラムは、リセットベクタからプログラムを実行しても問題なく実行できるように作成しています。

4.15 ブレーク機能

E6000H エミュレータは、ソフトウェアブレーク機能、オンチップブレーク機能およびオンエミュレータブレーク機能をもっています。

ソフトウェアブレークポイントの設定、オンチップブレーク設定およびオンエミュレータブレーク設定は[イベントポイント]ウィンドウでそれぞれ行うことができます。

以下にブレーク機能の概要と設定方法について説明します。

4.15.1 ソフトウェアブレーク機能

E6000H エミュレータは、最大 255 ポイントまでソフトウェアブレークポイントを設定することができます。

[表示 -> コード -> イベントポイント]を選択するか、[イベントポイント]ツールバーボタン  をクリックすると、[イベントポイント]ウィンドウが表示されます。[Software]シートを開きます。

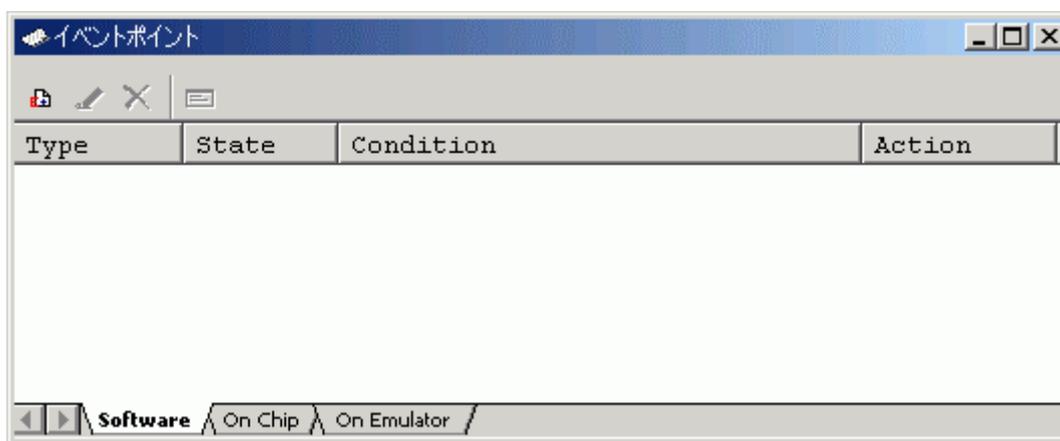


図4.30 イベントポイントウィンドウ（ソフトウェアブレーク設定前）

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加...]を選択してください。[ブレークポイントプロパティ]ダイアログボックス（[ソフトウェアブレーク]ページ）が表示されます。

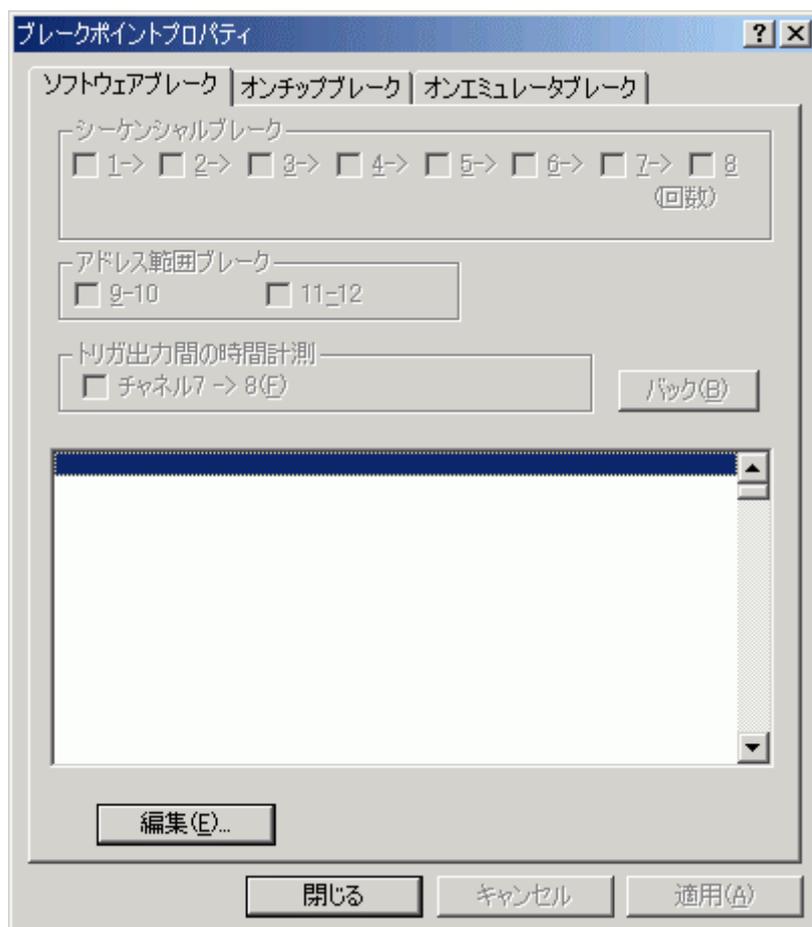


図4.31 ブレークポイントプロパティダイアログボックス

4 チュートリアル

[編集...]ボタンをクリックしてください。[ソフトウェアブレイク]ダイアログボックスが表示されます。



図4.32 ソフトウェアブレイクダイアログボックス

[アドレス]エディットボックスに tutorial 関数内の「p-sam->s0=a[0];」と記述されている行のアドレスを[Source]ウィンドウで参照し入力してください。本例では H'000010A4 を入力します。

【注】 本ダイアログボックスは、製品ごとに異なります。各製品の内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

[OK]ボタンをクリックしてください。[ブレイクポイントプロパティ]ダイアログボックスの[閉じる]ボタンを押してください。

[イベントポイント]ウィンドウには、設定されたソフトウェアブレイクポイントが表示されます。

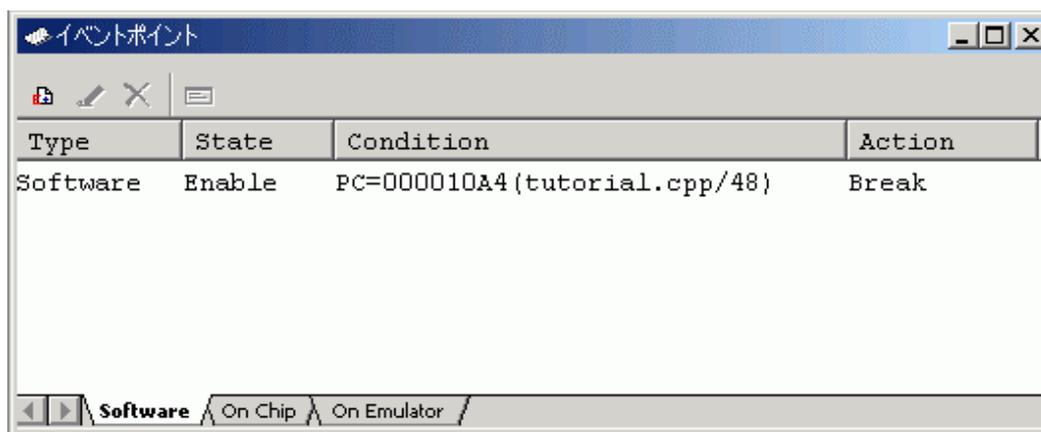


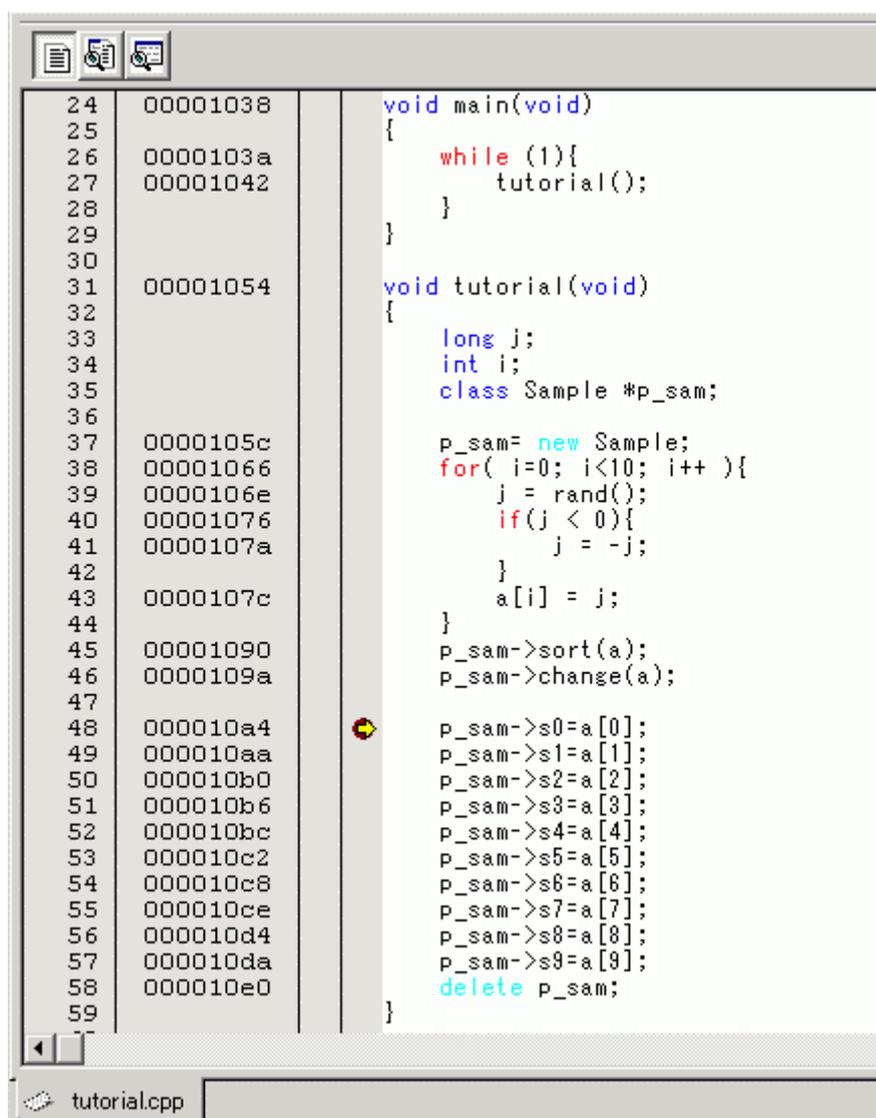
図4.33 イベントポイントウィンドウ (ソフトウェアブレイク設定時)

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

[イベントポイント]ウィンドウを閉じてください。

チュートリアルプログラムをブレークポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。

設定したブレークポイントまで、プログラムを実行して停止します。

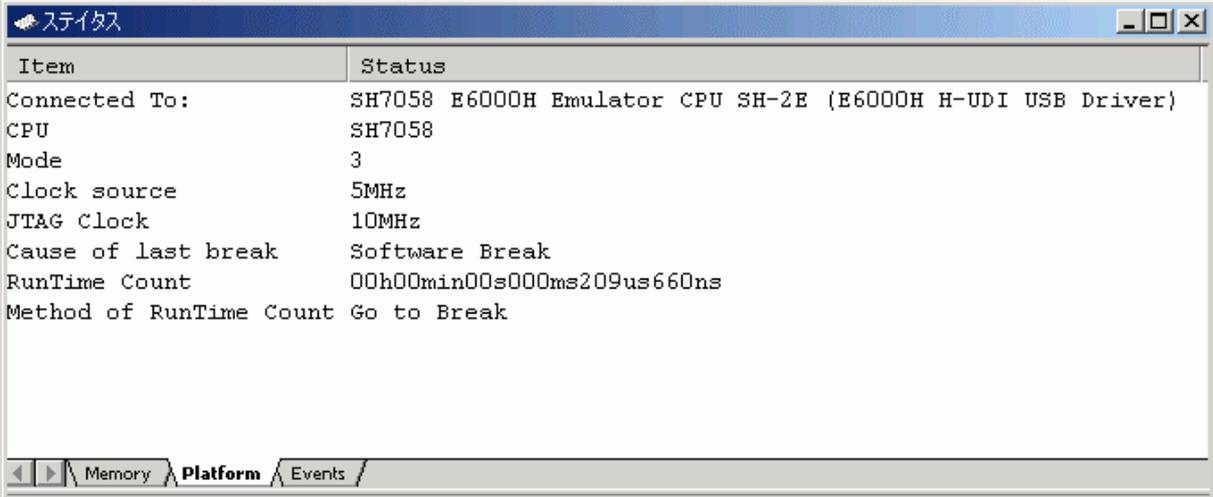


```
24 00001038 void main(void)
25 {
26 0000103a while (1){
27 00001042 tutorial();
28 }
29
30
31 00001054 void tutorial(void)
32 {
33 long j;
34 int i;
35 class Sample *p_sam;
36
37 0000105c p_sam= new Sample;
38 00001066 for( i=0; i<10; i++ ){
39 0000106e j = rand();
40 00001076 if(j < 0){
41 0000107a j = -j;
42 }
43 0000107c a[i] = j;
44 }
45 00001090 p_sam->sort(a);
46 0000109a p_sam->change(a);
47
48 000010a4 p_sam->s0=a[0];
49 000010aa p_sam->s1=a[1];
50 000010b0 p_sam->s2=a[2];
51 000010b6 p_sam->s3=a[3];
52 000010bc p_sam->s4=a[4];
53 000010c2 p_sam->s5=a[5];
54 000010c8 p_sam->s6=a[6];
55 000010ce p_sam->s7=a[7];
56 000010d4 p_sam->s8=a[8];
57 000010da p_sam->s9=a[9];
58 000010e0 delete p_sam;
59 }
```

図4.34 実行停止時の[エディタ]ウィンドウ（ソフトウェアブレーク）

4 チュートリアル

[ステータス]ウィンドウの表示内容は、以下のようになります。



The screenshot shows a window titled 'ステータス' (Status) with a table of emulator parameters. The table has two columns: 'Item' and 'Status'. The data is as follows:

Item	Status
Connected To:	SH7058 E6000H Emulator CPU SH-2E (E6000H H-UDI USB Driver)
CPU	SH7058
Mode	3
Clock source	5MHz
JTAG Clock	10MHz
Cause of last break	Software Break
RunTime Count	00h00min00s000ms209us660ns
Method of RunTime Count	Go to Break

At the bottom of the window, there are three tabs: 'Memory', 'Platform', and 'Events'. The 'Platform' tab is currently selected.

図4.35 ステータスウィンドウの表示内容 (ソフトウェアブレイク)

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

4.15.2 オンチップブ레이크機能

オンチップブ레이크のチャンネル8を用いて、指定した条件が5回成立した場合にブ레이크する方法を説明します。

【注】 成立回数を設定できるチャンネルは、製品ごとに異なります。各製品の内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

[表示 -> コード -> イベントポイント]を選択するか、[イベントポイント]ツールバーボタンをクリックして、[イベントポイント]ウィンドウを表示してください。

先ほど設定したソフトウェアブ레이크ポイントを削除します。マウスの右ボタンで[Software]シートをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているソフトウェアブ레이크ポイントをすべて解除してください。

次はオンチップブ레이크を設定します。[イベントポイント]ウィンドウの[On Chip]タブをクリックしてください。

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加...]を選択してください。[ブ레이크ポイントプロパティ]ダイアログボックス（[オンチップブ레이크]ページ）が表示されます。

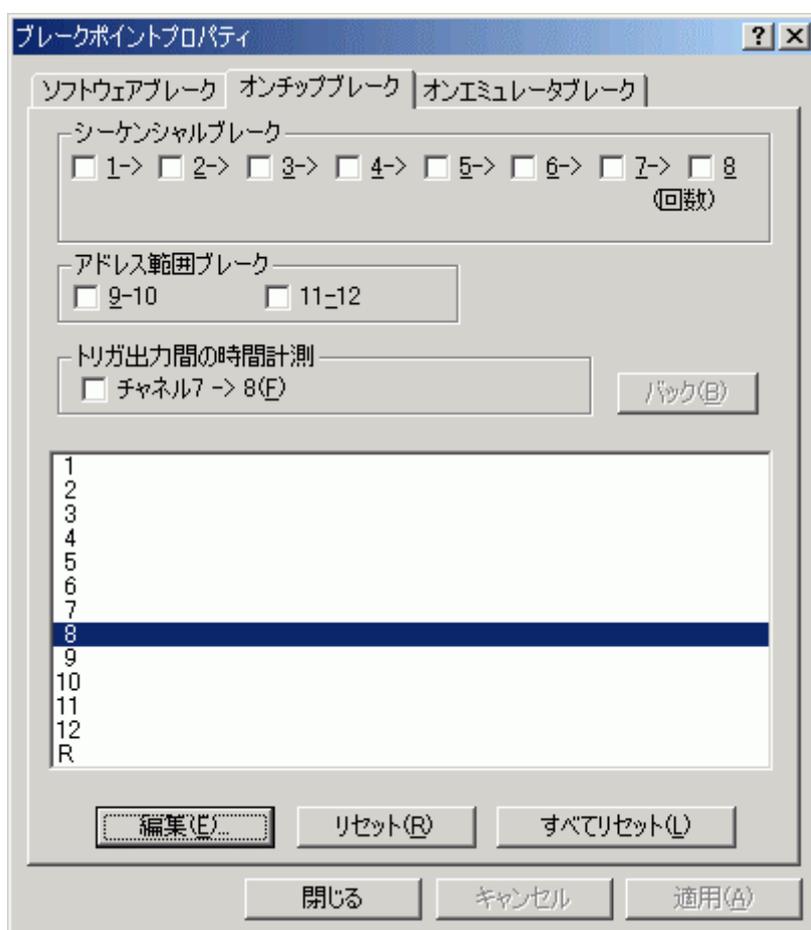


図4.36 ブ레이크ポイントプロパティダイアログボックス（オンチップブ레이크ページ）

リストボックスにて[8]を選択し、[編集...]ボタンをクリックしてください。[オンチップブ레이크 チャンネル 8]ダイアログボックスが表示されます。

4 チュートリアル

[アドレス]ページの各グループボックスを以下のように設定します。

- [無効]チェックボックスのチェックを解除します。
- [アドレス]エディットボックスにtutorial関数内の「a[i]=j;」と記述されている行のアドレスを[エディタ]ウィンドウで参照し入力してください。本例ではH'0000107Cを入力します。

[回数]ページの各グループボックスを以下のように設定します。

- [無効]チェックボックスのチェックを解除します。
- [回数]エディットボックスにD'5を入力します。

【注】 本ダイアログボックスは、製品ごとに異なります。各製品の内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

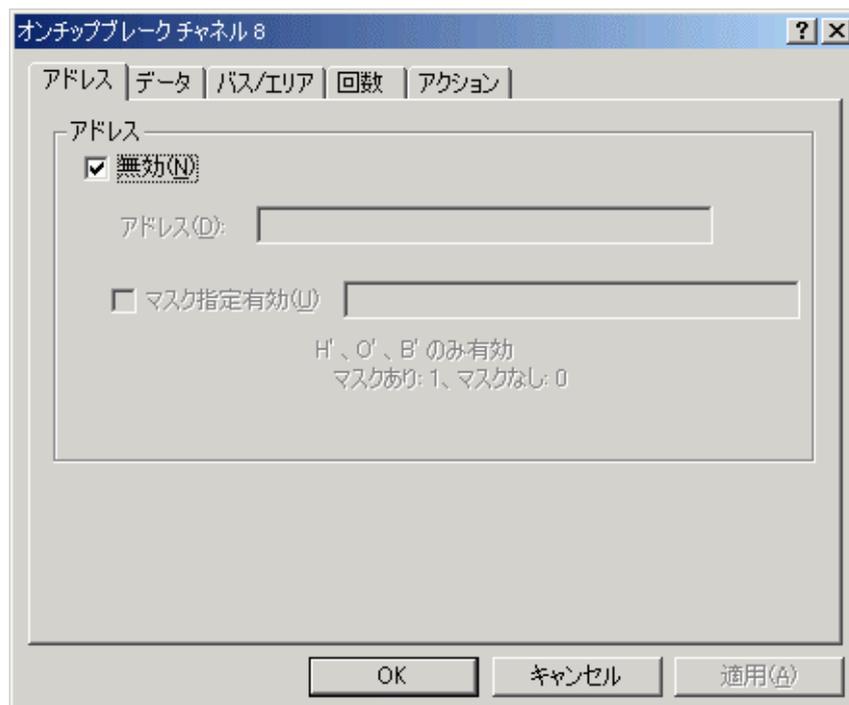


図4.37 オンチップブレイク チャンネル 8 ダイアログボックス

[OK]ボタンをクリックしてください。[ブレイクポイントプロパティ]ダイアログボックスの[閉じる]ボタンをクリックしてください。

[イベントポイント]ウィンドウには、設定されたオンチップブレイクポイントが表示されます。

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「3 デバッグ」またはオンラインヘルプを参照してください。

[イベントポイント]ウィンドウを閉じてください。

チュートリアルプログラムをオンチップブレイクポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。設定したブレイクポイントまで、プログラムを実行して停止します。

停止要因は[ステータス]ウィンドウの[Cause of last break]にて確認することができます。

[ウォッチ]ウィンドウから変数 i の値を参照します。変数 i の値は 4 となっており、イベントが 5 回成立したことによってブレイクが発生したことがわかります。

設定したイベントポイントを解除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除してください。

4.16 トレース機能

E6000H エミュレータのトレース機能もつリアルタイムトレースバッファでは、128k までのバスサイクルを保持でき、実行中は常に更新されます。リアルタイムトレースバッファの内容は[トレース]ウィンドウに表示されます。

[表示 -> コード -> トレース]を選択するか、[トレース]ツールバーボタンをクリックしてください。

[トレース]ウィンドウが表示されます。

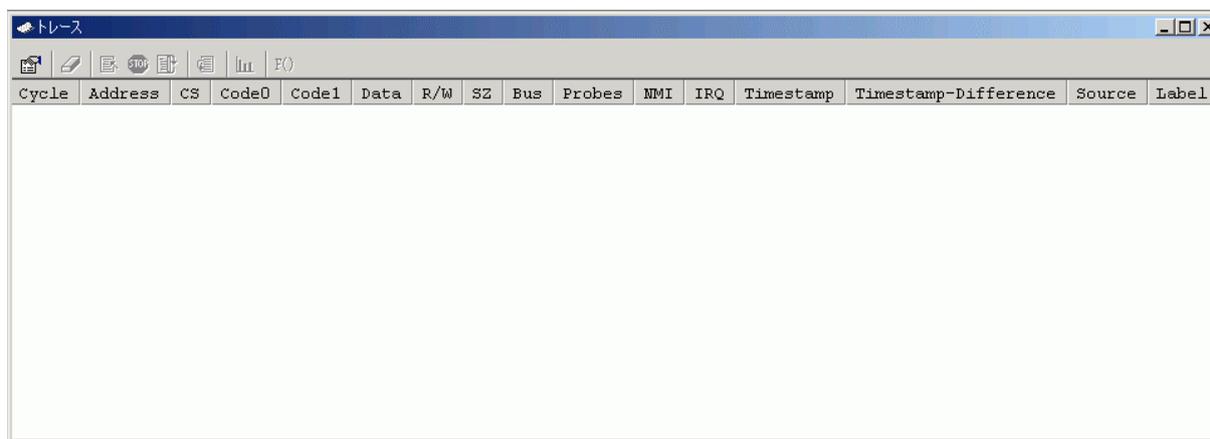


図4.38 トレースウィンドウ

[トレース]ウィンドウにトレース情報が表示されている場合は、マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

以下にトレース機能の概要と設定方法について説明します。

4 チュートリアル

4.16.1 フリートレースによるトレース情報の表示

フリートレース機能は、ユーザプログラムの実行を開始した時点からブレークするまでのトレース情報を連続的に取得します。

- (1) すべてのブレーク条件を解除してください。マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[トレース取得プロパティ]ダイアログボックスが表示されます。[フリートレース]チェックボックスがチェックされていることを確認してください。[閉じる]ボタンをクリックしてください。

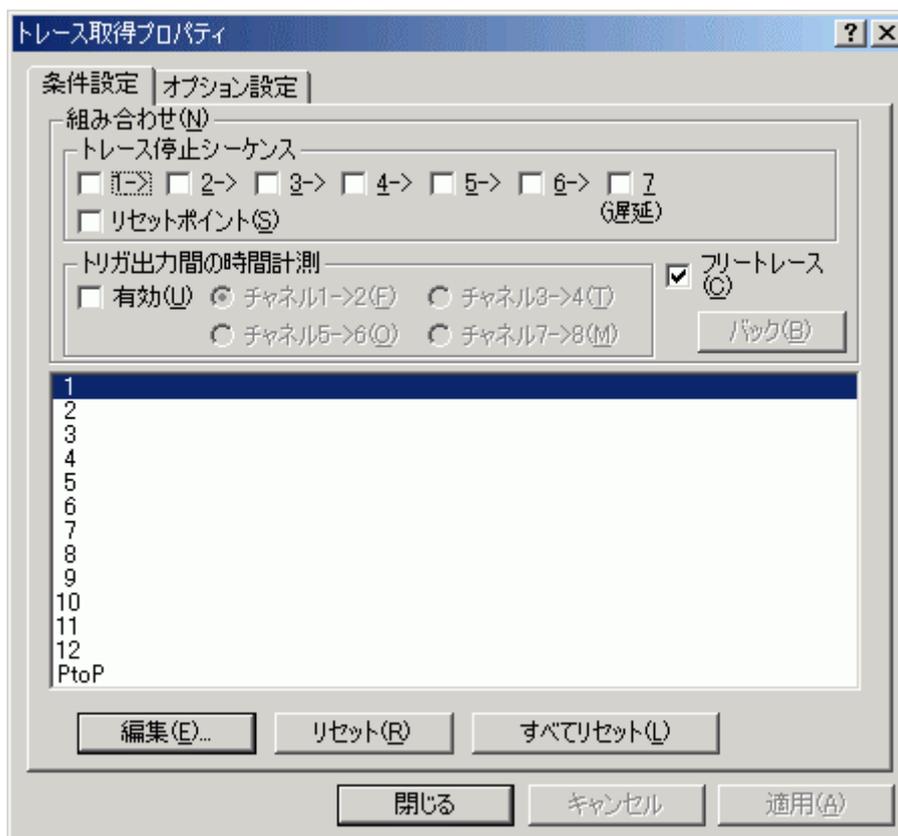


図4.39 トレース取得プロパティダイアログボックス (フリートレース)

- (2) tutorial関数内の「p-sam->s0=a[0];」と記述されている行にソフトウェアブレークを設定してください。(「4.15.1 ソフトウェアブレーク機能」参照)
- (3) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[トレース]ウィンドウにブレークするまでのトレース情報が表示されます。

4 チュートリアル

- (3) 条件のアドレスを設定します。[トレース取得条件 チャンネル 1]ダイアログボックス-[アドレス]ページの[無効]チェックボックスを解除し、[開始]エディットボックスにtutorial関数内の「a[i]=j;」と記述されている行のアドレスを[エディタ]ウィンドウで参照し入力してください。本例ではH'0000107Cを入力します。これでアドレスの指定は終了です。[OK]ボタンをクリックし、[トレース取得条件 チャンネル 1]ダイアログボックスを閉じます。



図4.42 トレース取得条件 チャンネル 1 ダイアログボックス ([アドレス]ページ)

- (4) [トレース取得プロパティ]ダイアログボックス-[条件設定]ページのリストボックスに設定した内容が表示されます。[トレース取得プロパティ]ダイアログボックスの[閉じる]ボタンをクリックしてください。

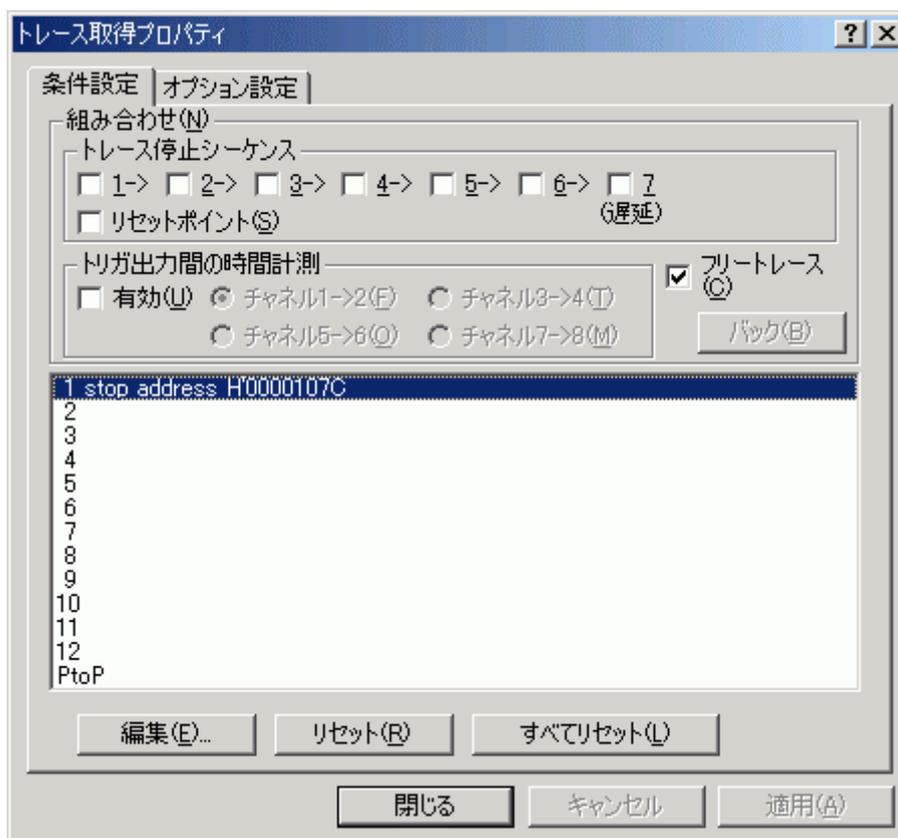


図4.43 トレース取得プロパティダイアログボックス (トレース停止)

- (6) [デバッグ]メニューから[リセット後実行]を選択してください。トレース停止条件が成立し、[トレース]ウィンドウに以下の内容が表示されます。

Cycle	Address	CS	Code0	Code1	Data	R/W	SZ	Bus	Probes	NMI	IRQ	Timestamp
-00012	00001240		MOV R2,R0	RTS	6023000b	IF		CPU	1111		1	0000h0...
-00011	00001244		LDS.L @R15+,...		4f163039	IF		CPU	1111		1	0000h0...
-00010	00001074		MOV R0,R2	CMP/PZ R2	62034211	IF		CPU	1111		1	0000h0...
-00009	ffffbfd0				00000000	RD	L	CPU	1111		1	0000h0...
-00008	00001078		BT @H'107c:8		8900622b	IF		CPU	1111		1	0000h0...
-00007	0000107c				d42066e3	IF		CPU	1111		1	0000h0...
-00006	0000107c		MOV.L @(H'00...	MOV R14,R6	d42066e3	IF		CPU	1111		1	0000h0...
-00005	00001080		MOV R6,R5	SHLL2 R5	65634508	IF		CPU	1111		1	0000h0...
-00004	00001100				ffff0400	RD	L	CPU	1111		1	0000h0...
-00003	00001084		MOV R5,R0	MOV.L R2,@(R0,R4)	60530426	IF		CPU	1111		1	0000h0...
-00002	00001088		ADD #H'01,R14	MOV #H'0A,R1	7e01e10a	IF		CPU	1111		1	0000h0...
-00001	ffff0400				00000000	WR	L	CPU	1111		1	0000h0...
+00000	0000108e				3e138bee	IF		CPU	1111		1	0000h0...

図4.44 トレースウィンドウ (トレースストップ)

4 チュートリアル

4.16.3 条件トレースによるトレース情報の表示

条件トレース機能は、指定した条件が成立した箇所のみをトレース取得します。特定のアドレス(グローバル変数やメモリマップド I/O など)へのリード/ライトに着目したプログラムの解析に役立ちます。

- (1) ユーザプログラムが実行中の場合は、[デバッグ]メニューから[プログラムの停止]を選択し、実行を停止させてください。
- (2) ブレーク条件が設定されている場合はすべて解除してください。[トレース取得プロパティ]ダイアログボックス-[条件設定]ページの[フリートレース]チェックボックスのチェックを解除してください。(解除しない場合はフリートレースになります。)
- (3) [トレース取得プロパティ]ダイアログボックス-[条件設定]ページのリストボックスから[1]を選択し、[編集...]ボタンをクリックしてください。[トレース取得条件 チャンネル 1]ダイアログボックスが表示されます。[アクション]ページの[条件成立後の処理]グループボックスから[条件トレース]ラジオボタンを選択してください。
- (4) 条件のアドレスを設定します。[トレース取得条件 チャンネル 1]ダイアログボックス-[アドレス]ページの[無効]チェックボックスを解除し、[開始]エディットボックスにa[0]のアドレスを[ウォッチ]ウィンドウで参照し入力してください。本例ではH'FFFF0400を入力します。これでアドレスの指定は終了です。[OK]ボタンをクリックし、[トレース取得条件 チャンネル 1]ダイアログボックスを閉じます。
- (5) [トレース取得プロパティ]ダイアログボックス-[条件設定]ページのリストボックスに設定した内容が表示されます。[トレース取得プロパティ]ダイアログボックスの[閉じる]ボタンをクリックしてください。
- (6) tutorial関数内の「delete p_sam;」と記述されている行のアドレス(本例ではH'000010E0)にソフトウェアブレークを設定します。(「4.15.1 ソフトウェアブレーク機能」参照)
- (7) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[トレース]ウィンドウに以下の内容が表示されます。

Cycle	Address	CS	Code0	Code1	Data	R/w	SZ	Bus	Probes	NMI	IRQ	Timestamp
-00012	-----								1111		1	0000h0...
-00011	ffff0400				00000000	RD	L	CPU	1111		1	0000h0...
-00010	-----								1111		1	0000h0...
-00009	ffff0400				00000000	RD	L	CPU	1111		1	0000h0...
-00008	-----								1111		1	0000h0...
-00007	ffff0400				00000000	RD	L	CPU	1111		1	0000h0...
-00006	-----								1111		1	0000h0...
-00005	ffff0400				00000000	RD	L	CPU	1111		1	0000h0...
-00004	-----								1111		1	0000h0...
-00003	-----								1111		1	0000h0...
-00002	ffff0400				00005665	WR	L	CPU	1111		1	0000h0...
-00001	ffff0400				00005665	RD	L	CPU	1111		1	0000h0...
+00000	-----								1111		1	0000h0...

図4.45 トレースウィンドウ(条件トレース)

4.16.4 統計

取得したトレース情報から、内蔵 RAM に対する書き込み回数を収集します。

- (1) ブレーク条件が設定されている場合はすべて解除してください。[トレース取得プロパティ]ダイアログボックス-[条件設定]ページの[すべてリセット]をクリックしトレース条件を解除してください。[トレース取得プロパティ]ダイアログボックス-[条件設定]ページの[フリートレース]チェックボックスのチェックを入れてください。
- (2) tutorial関数内の「p-sam->s0=a[0];」記述されている行のアドレス(本例ではH'000010A4)でブレークするように設定します。(「4.15.1 ソフトウェアブレーク機能」参照)
- (3) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[トレース]ウィンドウにトレース情報が表示されます。
- (4) マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[統計...]を選択してください。トレースデータのロードを示すダイアログが表示された後に、[統計]ダイアログボックスが表示されます。

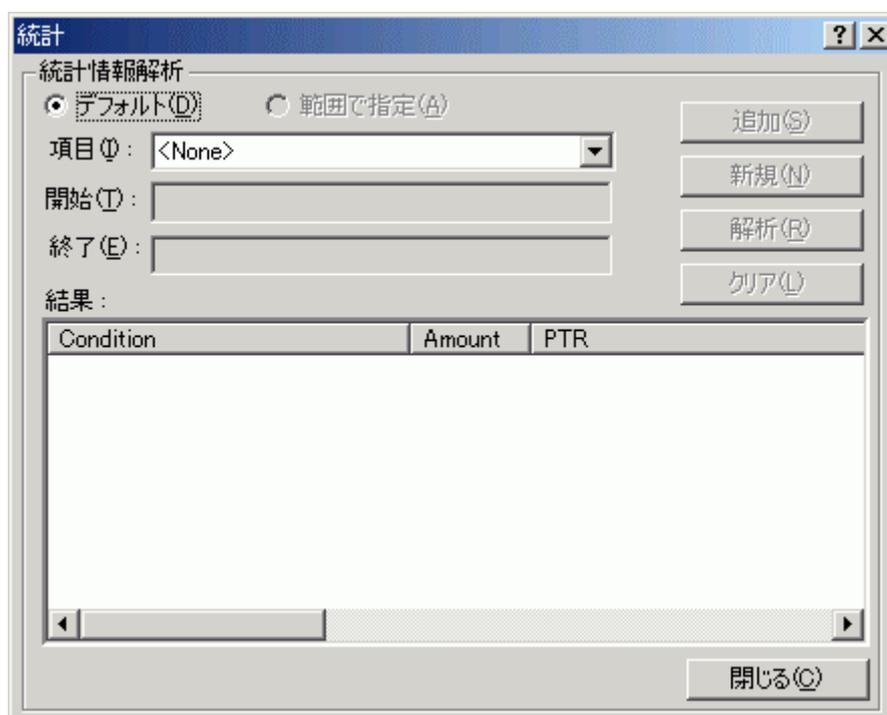


図4.46 統計ダイアログボックス

4 チュートリアル

- (5) [項目]ドロップダウンリストボックスからR/Wを選択し、[開始]エディットボックスにWRを入力します。入力終了後、[新規]ボタンをクリックすると[結果]リストボックスの[Condition]列に"R/W=WR"と表示されます。

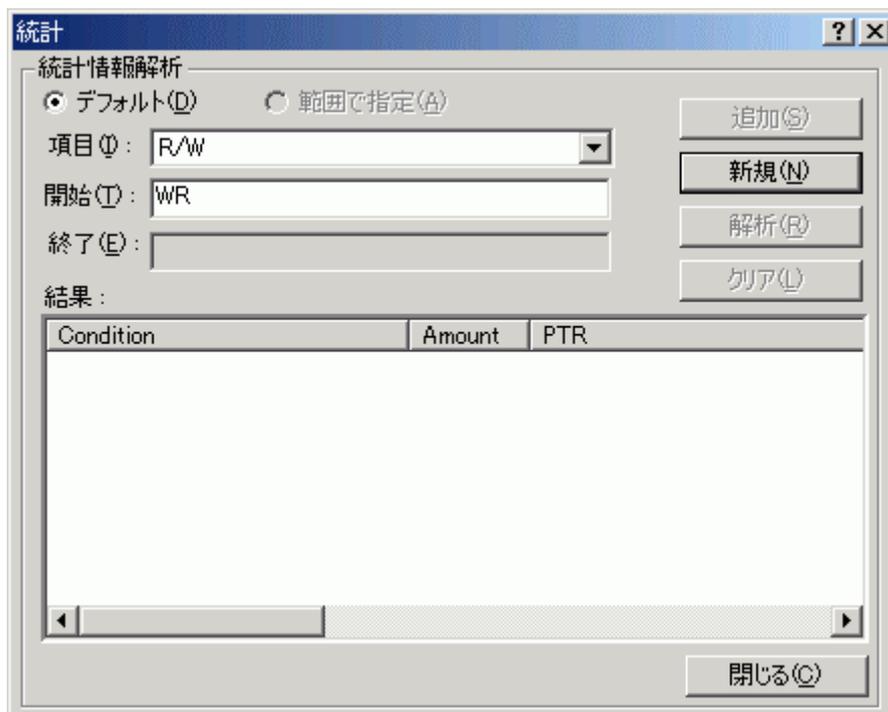


図4.47 統計ダイアログボックス（新規条件）

- (7) 続いて[項目]ドロップダウンリストボックスからSZを選択し、[開始]エディットボックスにLを入力します。入力終了後、[追加]ボタンをクリックすると[結果]リストボックスの[Condition]列に表示されていた"R/W=WR"に条件が追加され、"R/W=WR & SZ=L"と表示されます。これで条件の入力は終了です。

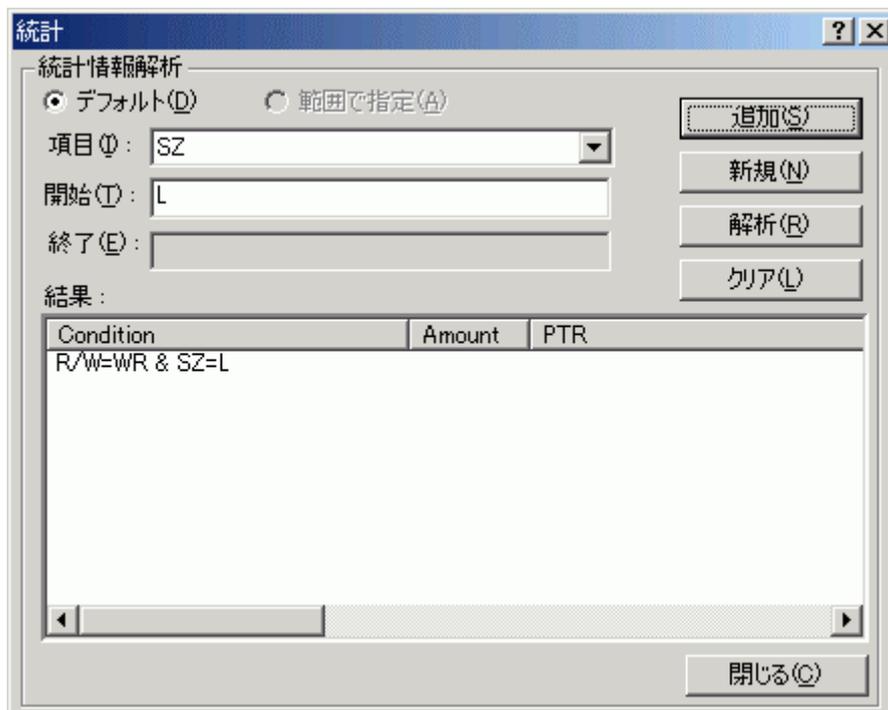


図4.48 統計ダイアログボックス（条件追加）

- (7) 入力した条件による解析を行います。[解析]ボタンをクリックしてください。条件に該当する件数とそのPTRが表示されます。

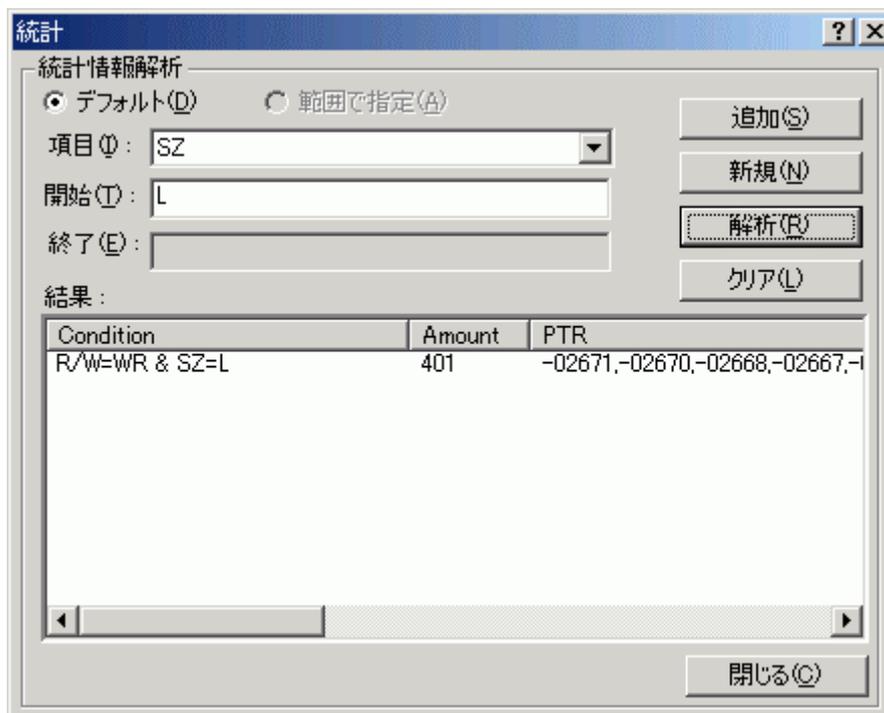


図4.49 統計ダイアログボックス（解析結果）

- (8) [閉じる]ボタンをクリックし、[統計]ダイアログボックスを閉じます。
- (9) 設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

4 チュートリアル

4.16.5 関数コール

取得したトレース情報から、関数をコールしている情報のみを収集します。

- (1) tutorial 関数内の「p-sam->s0=a[0];」と記述されている行のアドレス（本例では H'000010A4）でブレークするように設定します。（「4.15.1 ソフトウェアブレーク機能」参照）
- (2) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[トレース]ウィンドウにトレース情報が表示されます。
- (3) マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[関数コール...]を選択してください。[関数コール箇所の表示]ダイアログボックスが表示されます。



図4.50 関数コール箇所の表示ダイアログボックス

- (4) [許可]ラジオボタンをクリックし、[OK]ボタンをクリックしてください。[トレース]ウィンドウの表示が関数をコールしている情報のみに変更されます。

Cycle	Address	CS	Data	R/W	SZ	Bus	Probes	NMI	IRQ	Source	Label
-02679	00000a00		d20ce610	IF		CPU	1111		1	set_vbr((void *) {(char ...	_PowerON_Reset_PC
-01475	00001038		4f22e101	IF		CPU	1111		1	void main(void)	_main
-01470	00001054		2fd62fe6	IF		CPU	1111		1	void tutorial(void)	tutorial()
-01462	00002000		4f222448	IF		CPU	1111		1	Sample::Sample()	Sample::Sample()
-01389	00001000		7ffcd60b	IF		CPU	1111		1	char *sbrk(unsigned long si...	_sbrk
-00974	00002032		00092fc6	IF		CPU	1111		1	void Sample::sort(long *a)	Sample::sort(long *)
-00268	000020ce		00092f96	IF		CPU	1111		1	void Sample::change(long *a)	Sample::change(long *)

図4.51 トレースウィンドウ（関数コール）

- (5) [トレース]ウィンドウの表示を元の状態に戻します。(3)の手順にて[関数コール箇所の表示]ダイアログボックスを表示し、[無効]ラジオボタンをクリック、その後[OK]ボタンをクリックします。
- (6) 設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[トレース]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

4.17 スタックトレース機能

E6000H エミュレータでは、スタック情報を用いて、関数呼び出し履歴を表示します。

- 【注】
1. 本機能は、Dwarf2 形式のデバッグ情報をもったロードモジュールをロードした場合のみ使用できます。
Dwarf2 形式のデバッグ情報をもったロードモジュールは、H8S,H8/300 C/C++コンパイラ V4.0 以降でサポートしています。
 2. 本機能の詳細については、オンラインヘルプを参照してください。

sort 関数内の行の[S/W ブレークポイント]カラムをダブルクリックして、ソフトウェアブレークポイントを設定してください。

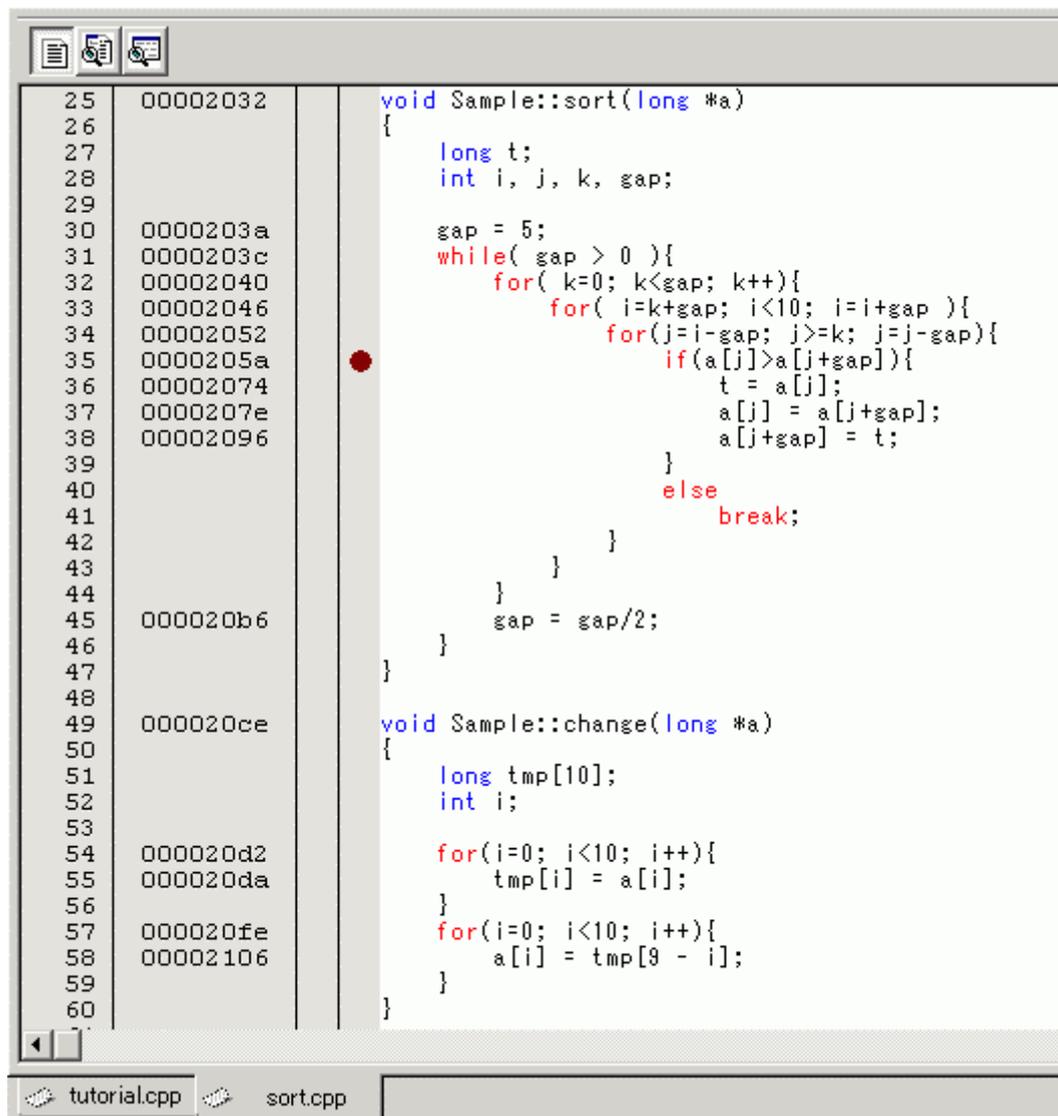
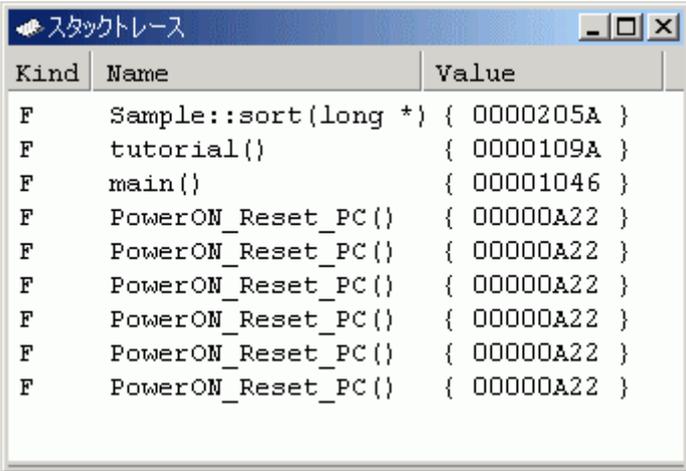


図4.52 [エディタ]ウィンドウ (ソフトウェアブレークポイントの設定)

[デバッグ]メニューから[リセット後実行]を選択してください。

4 チュートリアル

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



Kind	Name	Value
F	Sample::sort(long *)	{ 0000205A }
F	tutorial()	{ 0000109A }
F	main()	{ 00001046 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }
F	PowerON_Reset_PC()	{ 00000A22 }

図4.53 スタックトレースウィンドウ

現在 PC が sort()関数内にあり、sort()関数は、tutorial()関数からコールされていることがわかります。

sort 関数内の行の[S/W ブレークポイント]カラムを再度ダブルクリックして、ソフトウェアブレークポイントを解除します。

4.18 パフォーマンス測定機能

E6000H エミュレータには、チップのパフォーマンスを測定する機能として、以下に示すモードがあります。

- 指定範囲内時間測定
- 指定アドレス間時間測定
- 指定アドレス範囲間時間測定
- 領域アクセス回数測定
- 指定範囲内コール回数測定

本チュートリアルでは「指定範囲内時間測定」の設定方法について説明します。

4.18.1 指定範囲内時間測定

(1) [表示]メニューから[パフォーマンス]サブメニューを選択し、[パフォーマンス解析]を選択します。[パフォーマンス解析方式の選択]ダイアログボックスが表示されます。

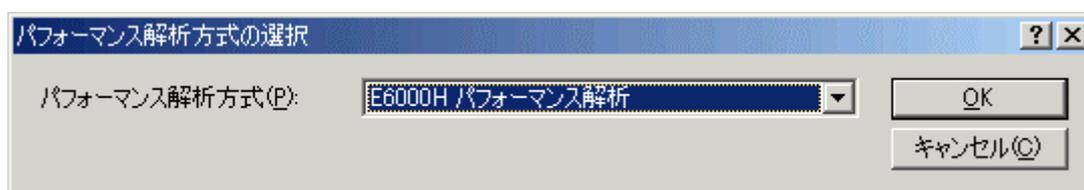


図4.54 パフォーマンス解析方式の選択ダイアログボックス

(2) [パフォーマンス解析方式の選択]ダイアログボックスの[パフォーマンス解析方式]ドロップダウンリストボックスから"E6000H パフォーマンス解析"を選択し、[OK]ボタンをクリックします。[パフォーマンス解析]ウィンドウが表示されます。

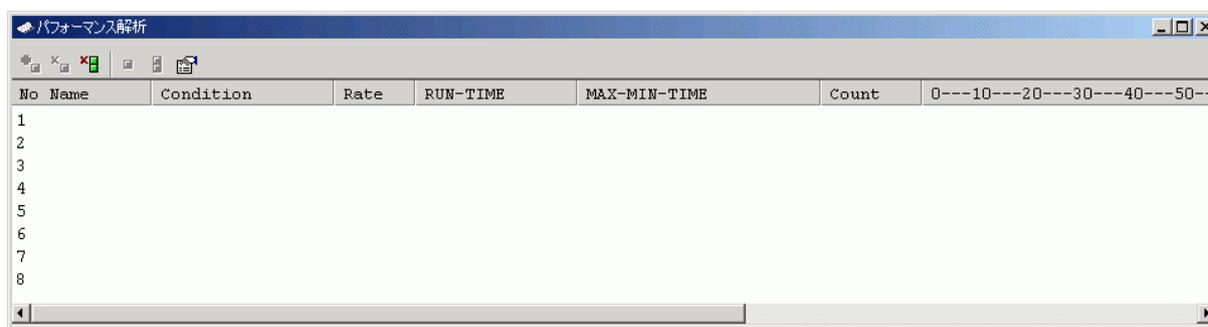


図4.55 パフォーマンス解析ウィンドウ

4 チュートリアル

(3) [パフォーマンス解析]ウィンドウの[No]列が"1"の行を選択し、マウスの右ボタンをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[パフォーマンス解析プロパティ]ダイアログボックスが表示されます。

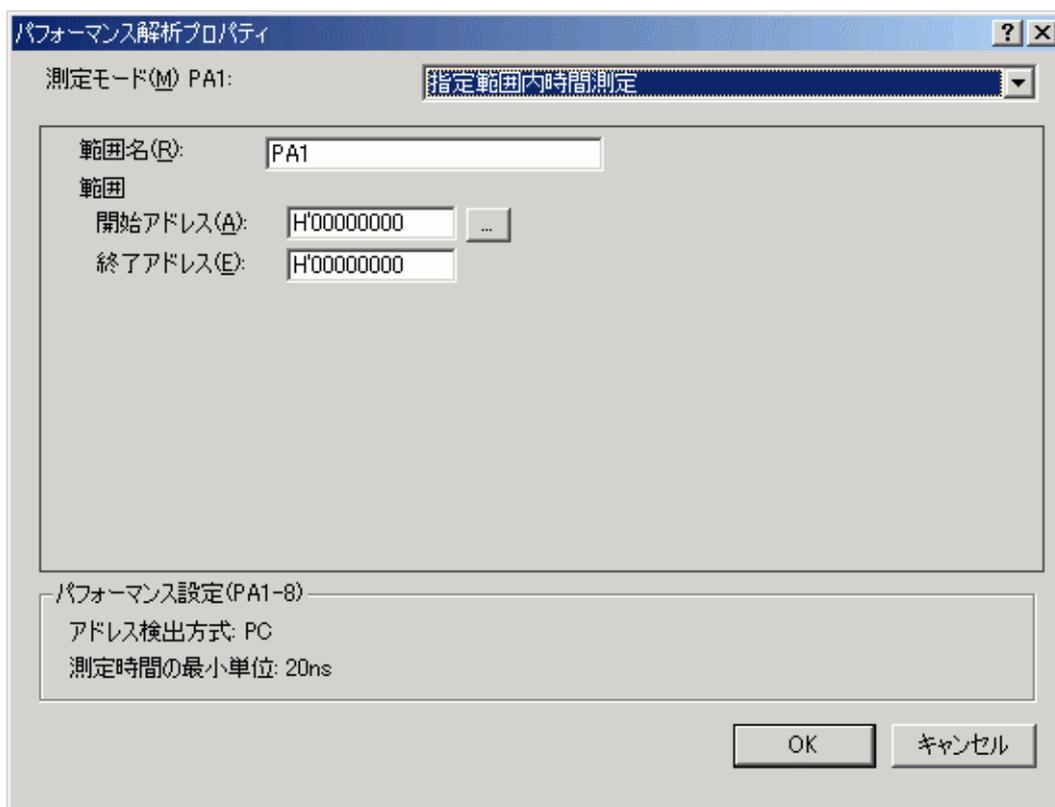


図4.56 パフォーマンス解析プロパティダイアログボックス

(4) [測定モード]ドロップダウンリストボックスから、モードとして指定範囲内時間測定を選択します。

(5) パラメータの設定は以下とします。

- [範囲名]エディットボックスにsortと入力します。
- [開始アドレス]エディットボックスの右側の[...]ボタンをクリックし、[関数範囲入力]ダイアログボックスを表示します。[関数範囲入力]ダイアログボックスの[関数名]エディットボックスに関数名"Sample::sort (long *)"を入力し、[OK]ボタンをクリックします。[開始アドレス]エディットボックスと[終了アドレス]エディットボックスに指定した関数のアドレスが設定されます。

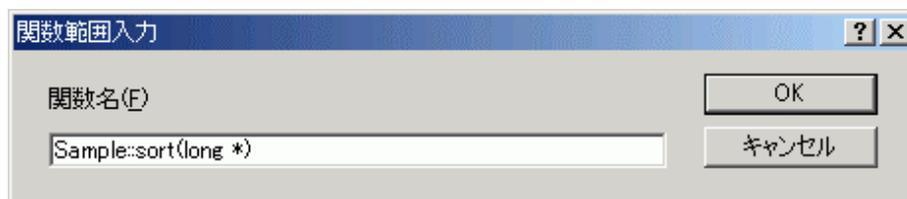


図4.57 関数範囲入力ダイアログボックス

【注】 [関数範囲入力]ダイアログボックスにより算出されるアドレスは参考値です。場合により関数の終了アドレスが異なる場合があります。
[逆アセンブリ]ウィンドウにより関数の最終命令を確認し、[終了アドレス]の設定値を最終命令のアドレスに補正してください。(一般的に関数の最終命令はRTS命令となります。)
アドレス入力ではアドレス値以外にラベル名および式の指定も可能です。

(6) [OK]ボタンをクリックします。[パフォーマンス解析]ウィンドウの[No]列 1 に設定した内容が表示されます。これで指定範囲内時間測定の設定は終了です。

No	Name	Condition	Rate	RUN-TIME	MAX-MIN-TIME	Count
1	PA1	Range H'00002032 H'000020CC	0%	00h 00min 00s 000ms 000us 000ns		0
2						
3						
4						
5						
6						
7						
8						

図4.58 パフォーマンス解析ウィンドウ（設定完了時）

(7) tutorial 関数内の「p-sam->change(a);」と記述されている行のアドレス（本例では H'0000109A）にソフトウェアブレークポイントによるブレークを設定します。（「4.15.1 ソフトウェアブレーク機能」参照）

(8) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[パフォーマンス解析]ウィンドウに以下の内容が表示されます。[Count]列の値が 1 となっており、sort 関数が 1 回実行されたことと、実行時間が確認できます。本チュートリアルでは、最小時間測定幅を 20ns で設定しています。最小時間測定幅の変更は [コンフィグレーションプロパティ] ダイアログボックスにて行います。

No	Name	Condition	Rate	RUN-TIME	MAX-MIN-TIME	Count
1	PA1	Range H'00002032 H'000020CC	32%	00h 00min 00s 000ms 060us 560ns		1 #####
2						
3						
4						
5						
6						
7						
8						

図4.59 パフォーマンス解析ウィンドウ（結果表示）

(9) パフォーマンスの設定をクリアし、イベントポイントを解除します。マウスの右ボタンで[パフォーマンス解析]ウィンドウをクリックすることによって開くポップアップメニューから[全てリセット]を選択し、設定されている設定をすべてクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。

4.19 プロファイル機能

E6000H エミュレータでは、各関数のパフォーマンスを計測できる「プロファイル機能」があります。

(1) [表示]メニューからプロファイルを選択し[プロファイル]ウィンドウを開いてください。

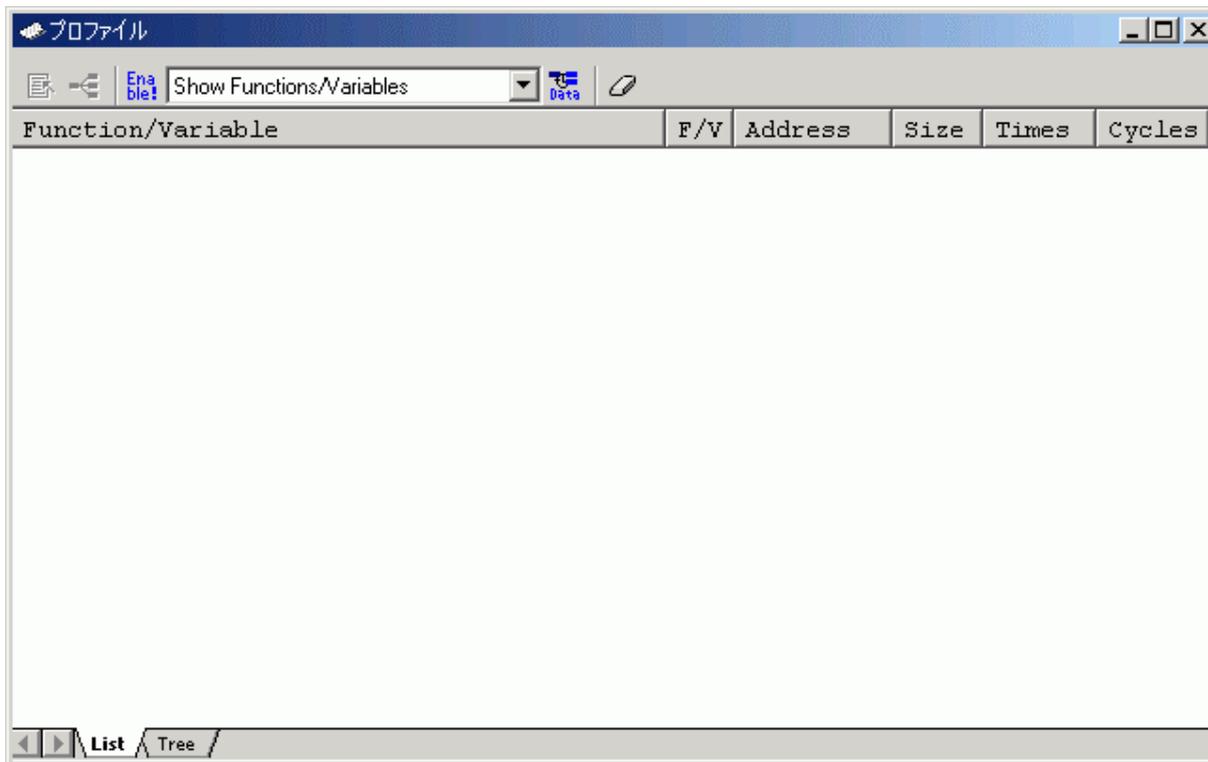


図4.60 プロファイル-List ウィンドウ

(2)次にプロファイル機能を有効にします。マウスの右ボタンで[プロファイル]ウィンドウを右クリックすることによって開くポップアップメニューから[有効]を選択してください。

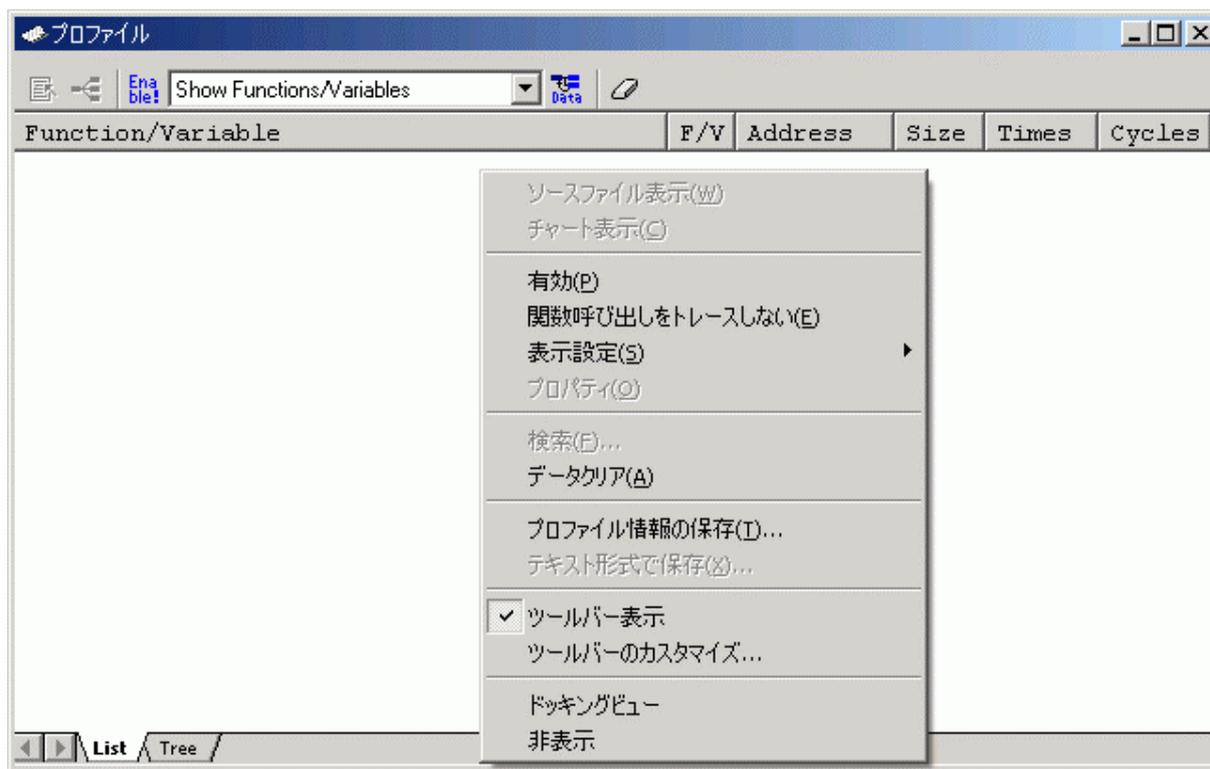


図4.61 有効の選択

4 チュートリアル

- (3) tutorial 関数内の「delete p_sam;」と記述されている行のアドレスにアドレス条件のみのオンチップブレークを設定します。
(「4.15.2 オンチップブレーク機能」参照)

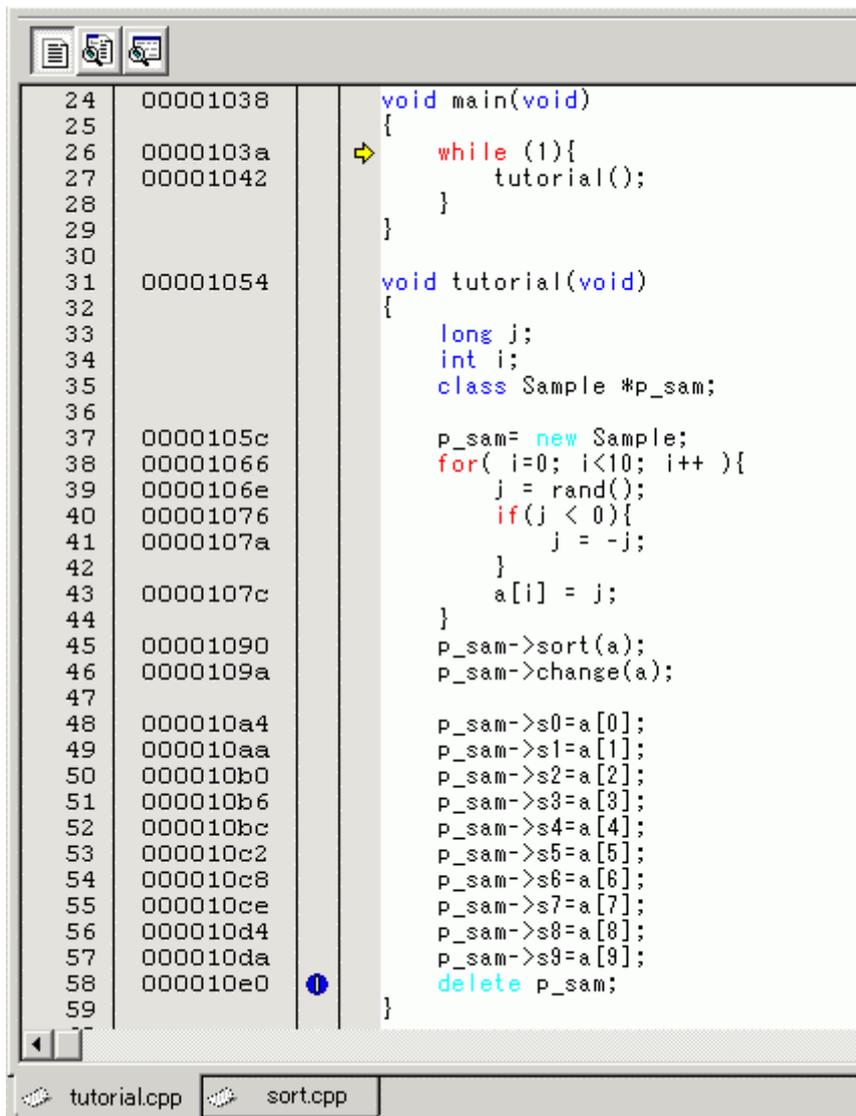


図4.62 [エディタ]ウィンドウ (オンチップブレークの設定)

- (4) プロファイルでの計測を行うため、「デバッグ」メニューから「リセット後実行」を選択してください。

(5) プログラムブレーク後、[プロファイル]ウィンドウに測定結果が表示されます。

- [プロファイル]ウィンドウは以下のようになります。

図4.63 プロファイルウィンドウ (List シート)

- [プロファイル]ウィンドウのTreeタブをクリックすることで[Tree]シートを開きます。

図4.64 プロファイルウィンドウ (Tree シート)

4 チュートリアル

- [プロファイル]ウィンドウを右クリックして、[チャート表示]を選択することで、[プロファイル - チャート]ウィンドウが開きます。

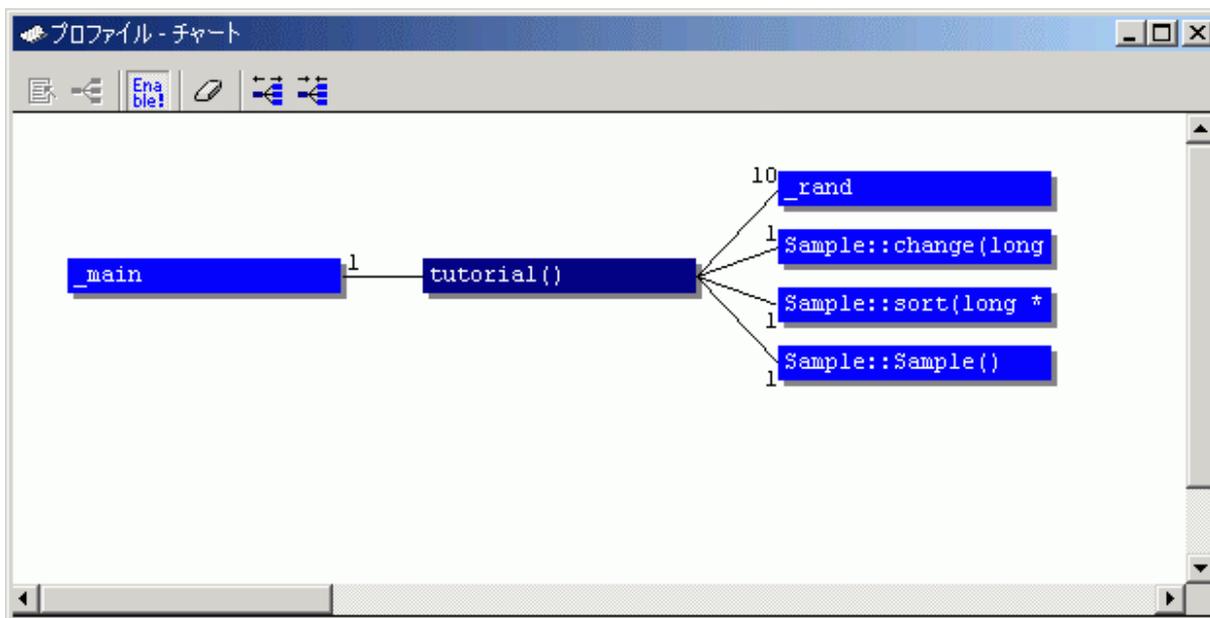


図4.65 プロファイルチャートウィンドウ

(6) プロファイル機能を無効にします。マウスの右ボタンで[プロファイル]ウィンドウを右クリックすることによって開くポップアップメニューで[有効]を解除してください。

ブレイク条件が設定されている場合はすべて解除してください。

4.20 モニタ機能

E6000H エミュレータには、指定したアドレスのメモリ内容を、ユーザプログラム実行中にモニタすることができます。ここでは tutorial 関数の変数 a が割り当てられたアドレス近辺の内容をモニタします。

- (1) [表示]メニューから[CPU]サブメニューを選択、さらに[モニタ]サブメニューを選択し、[モニタ設定...]を選択します。
[Monitor Setting]ダイアログボックスが表示されます。

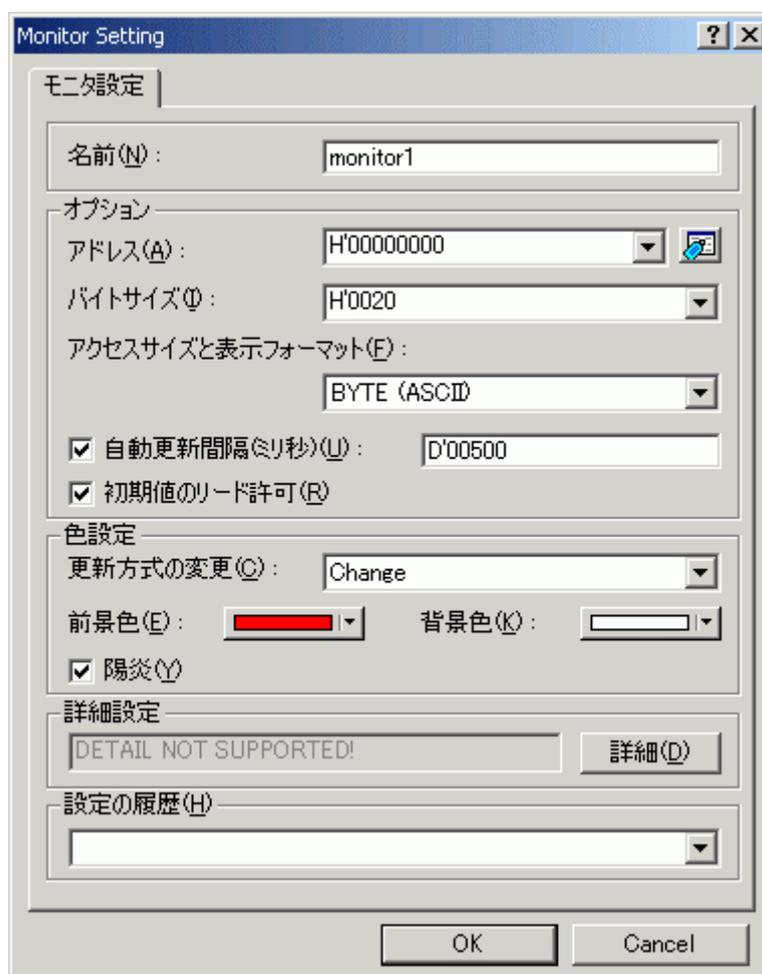


図4.66 Monitor Setting ダイアログボックス

4 チュートリアル

(2) [モニタ設定]ページの内容を以下のように設定します。

- [名前]エディットボックスにmonitor1と入力します。
- [オプション]グループボックスの各パラメータを以下の設定とします。
 - [アドレス]エディットボックスにtutorial関数で定義している変数 a が割りついているアドレスを[ローカル]ウィンドウで参照し入力してください。本例では H'FFFF0400 を入力します。
 - [バイトサイズ]ドロップダウンリストボックスに H'20 と入力します。
 - [アクセスサイズと表示フォーマット]ドロップダウンリストボックスから BYTE (ASCII) を選択します。
 - [自動更新間隔 (ミリ秒)]のチェックボックスをチェックし、エディットボックスに D'00500 と入力します。
 - [初期値のリード許可]チェックボックスをチェックします。
- [色指定]グループボックスの各パラメータを以下の設定とします。
 - [更新方式の変更]ドロップダウンリストボックスから Change を選択します。
 - [前景色]ドロップダウンリストボックスを赤に、[背景色]ドロップダウンリストボックスを白に選択します。
 - [陽炎]チェックボックスをチェックします。

【注】 前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

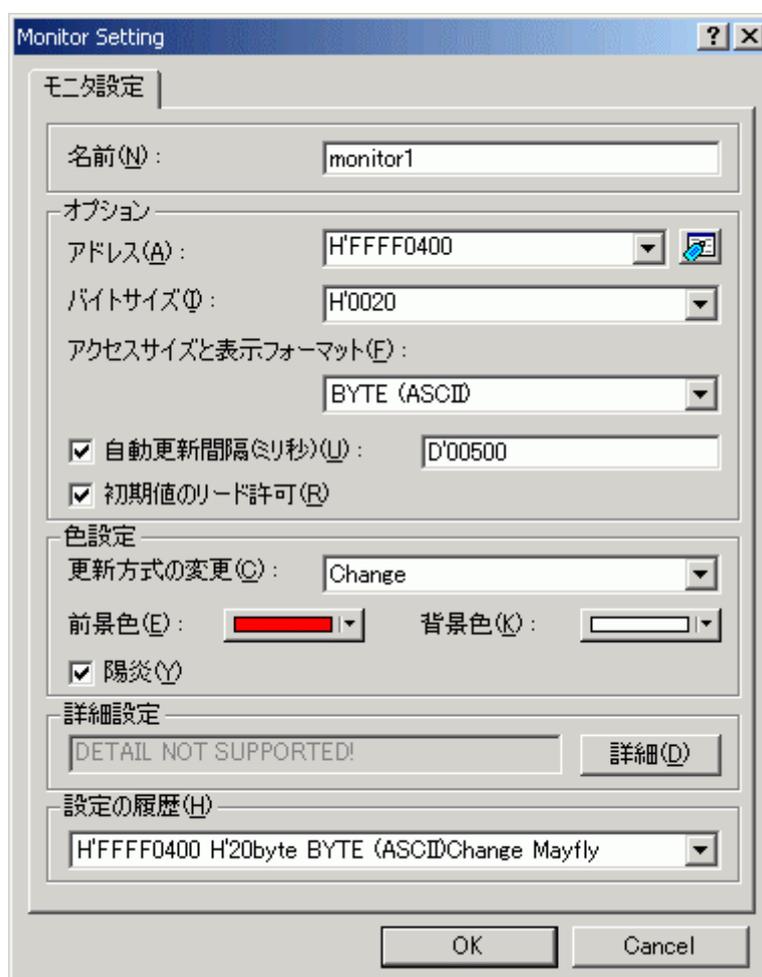


図4.67 Monitor Setting ダイアログボックス (設定完了時)

(3) [OK]ボタンをクリックします。[モニタ]ウィンドウが開きます。

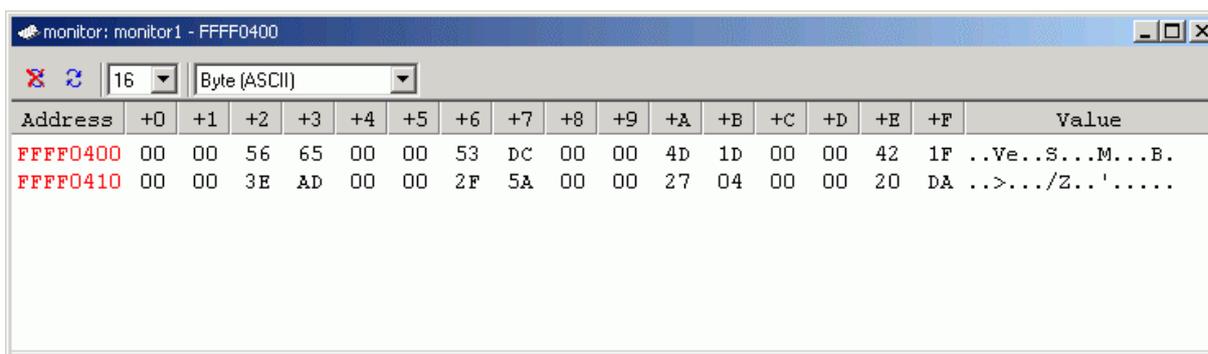


図4.68 モニタウィンドウ（実行前）

(4) [デバッグ]メニューから[リセット後実行]を選択してください。処理が実行されることによりメモリの内容が更新された場合は、更新されたメモリの値が赤（[モニタ設定]ページの[前景色]ドロップダウンリストボックスと[背景色]ドロップダウンリストボックスで設定した色）に変更されます。また、更新が行われないメモリの値および、更新後一定間隔更新が行われなかったメモリの値は黒で表示されます。

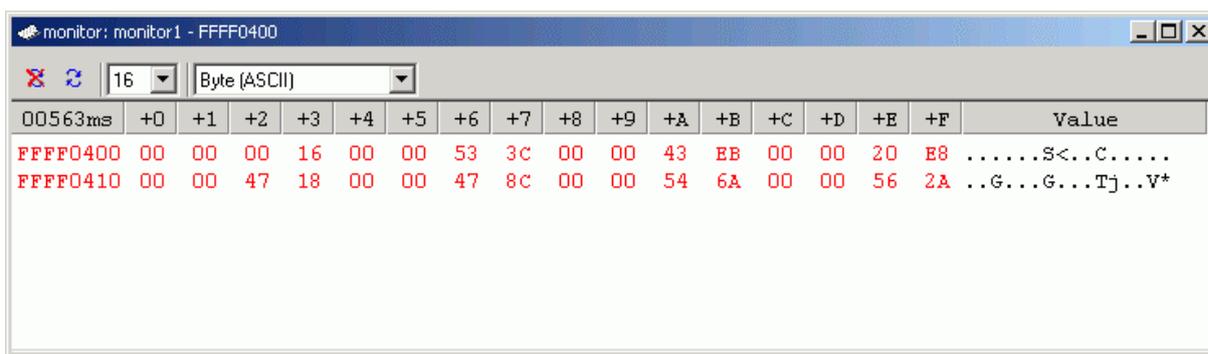


図4.69 モニタウィンドウ（実行中）

(5) [モニタ]ウィンドウの表示状況が確認できましたら、[デバッグ]メニューから[プログラムの停止]を選択し、実行を停止します。

4.21 さてつきは？

このチュートリアルでは、E6000H エミュレータのいくつかの主な特徴と、High-performance Embedded Workshop の使い方を紹介しました。

E6000H エミュレータで提供されるエミュレーション機能を使用することによって、高度なデバッグを行うことができます。それによって、ハードウェアとソフトウェアの問題が発生する条件を正確に分離し、識別すると、それらの問題点を効果的に調査することができます。

5 本製品固有のソフトウェア仕様と注意事項

この章では、E6000H エミュレータに関するソフトウェア仕様と注意事項について説明します。

5.1 対応ハードウェアについて

本エミュレータソフトウェアは SH7058 E6000H エミュレータ (HS7058EPH62H)、および SH7058S、SH7059 E6000H エミュレータ (HS7059EPH62H) に対応したソフトウェアです。

5.2 デバッグングプラットフォームについて

本エミュレータで選択可能なデバッグングプラットフォームは以下の通りです。

選択したデバッグングプラットフォームによりエミュレーション可能なターゲットマイコンが異なります。

表5-1 選択可能ターゲット

デバッグングプラットフォーム	備考	ハードウェア
SH7058 E6000H Emulator CPU SH-2E	以下のターゲットマイコンのエミュレーションを行います。 SH7058	HS7058EPH62H
SH7058S E6000H Emulator CPU SH-2E	以下のターゲットマイコンのエミュレーションを行います。 SH7058S	HS7059EPH62H
SH7059 E6000H Emulator CPU SH-2E	以下のターゲットマイコンのエミュレーションを行います。 SH7059	HS7059EPH62H

5.3 メモリ内容の表示と変更方法について

5.3.1 実行中のメモリ内容の表示と変更

本エミュレータでは、ユーザプログラム実行中のメモリ内容の表示と変更について以下の2つのアクセス方法があります。

表5-2 メモリ内容の表示と変更タイプ

アクセス方法	概要	停止時間	表示	変更
モニタ機能	ユーザプログラムの実行を停止せずにメモリ内容の表示を自動更新します。 指定範囲へのアクセスが発生しないと、更新は行いません。	なし	可	不可
パラレルアクセス機能	内蔵 ROM/RAM/エリアの場合、ユーザプログラムの実行を停止せずにメモリ内容の表示を自動更新します。 更新時は、強制的にメモリアccessを行うため、若干リアルタイム性がなくなります。	小	可	可

また、それぞれのアクセス方法には、以下の特徴があります。

表5-3 メモリ内容の表示と変更タイプ

アクセス方法	対象ウィンドウ / 操作	対象メモリ領域
モニタ機能	[モニタ]ウィンドウ、モニタ機能を使用した[ウォッチ]ウィンドウのリアルタイム更新表示	ユーザプログラムが使用可能な全領域のうち指定した8ポイントまたは合計256バイトまでの領域。
パラレルアクセス機能	[モニタ]ウィンドウ以外のメモリ内容の表示を行うウィンドウ ツールチップウォッチ、インスタントウォッチ メモリ内容を表示または変更するコマンド	ユーザプログラムが使用可能な領域。

【注】 スリープモードおよびスタンバイモード中に、パラレルアクセス機能によるメモリ内容の表示と変更はできません。

5 本製品固有のソフトウェア仕様と注意事項

- 【注】 [コンフィグレーションプロパティ]ダイアログボックスの[AUD 機能をエミュレータで使用(プログラム実行中のメモリアクセス許可)]を無効にすることで、ユーザプログラム実行中のパラレルアクセスによるメモリリードを禁止することができます。ユーザプログラム実行にリアルタイム性が必要な場合は、[AUD 機能をエミュレータで使用(プログラム実行中のメモリアクセス許可)]を無効にしてください。

5.3.2 パラレルアクセス機能停止時間参考値

ユーザプログラム実行中のメモリ内容の表示および変更のためのプログラム停止時間の参考値を示します。

表5-4 停止時間の参考値

アクセス方法	条件		停止時間
パラレルアクセス	リード	内蔵 RAM に対するロングワードアクセスによるリード	320ns
	ライト	内蔵 RAM に対するロングワードアクセスによるライト	320ns
Auto update Memory	ユーザプログラムの実行を停止しません。		

表5-5 測定環境

項目	設定値
システムクロック ()	10.0MHz
クロックモード	クロックモード3
H-UDI クロック	5.0MHz

5.3.3 モニタ機能

- モニタ機能は最大8ポイントまたは合計256バイトまでの指定が可能です。
- 1チャンネル32バイトのチャンネル8本のハード機構によりモニタ機能を実現しています。各チャンネルは32バイト境界で設定する必要があるため、32バイト境界をまたぐ場合は、2本のチャンネルを使用します。このため複数の32バイト境界をまたぐポイント指定した場合は、設定できる合計が256バイト未満になります。
- ユーザプログラム実行中に、モニタ機能の設定または変更を行った場合、ユーザプログラムのリアルタイム性はなくなります。
- ユーザプログラム実行中に、[アクセスサイズと表示フォーマット]の変更を行った場合、ユーザプログラムのリアルタイム性はなくなります。

5.3.4 メモリアクセスの注意事項

シングルチップモード時に H `00200000 ~ H `7FFFFFFF の外部メモリ空間をメモリウィンドウでアクセス(メモリ内容の表示、変更)しないでください。アクセスした場合、アドレスエラーが発生します。

5.4 プログラムの実行について

5.4.1 ステップ実行

- ステップの実行では、ブレークが無効になります。ただしトリガは出力されます。

5.4.2 ブレークについて

ブレーク条件が成立したときに実行している命令によっては、ブレーク処理が保留される場合があります。保留したブレークは、ブレークが受け付けられる状態になったときに発生します。

ブレークが保留される命令を下記に示します。

遅延分岐命令	JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF
FPU 命令	FPU 命令の浮動小数点演算命令および FPU に関する CPU 命令

【注】 ソフトウェアブレークポイントは命令を置き換えることにより実現しているため、保留されることはありません。

ステップ実行機能を実現するため E6000H エミュレータは内部的にブレーク条件を設定しています。このためステップ実行により上記命令を実行した場合は、ブレークが受け付けられる状態になるまでプログラムを実行する場合があります。

5.5 イベント機能について

5.5.1 ソフトウェアブレーク

- ソフトウェアブレークポイントは命令を置き換えることにより実現しますので、RAM (エミュレーションRAMを含む) 領域にのみ設定できます。また、次に示すアドレスには、指定できません。

メモリ内容が H'0000 であるアドレス

- ソフトウェアブレークポイントのアドレスの内容は、ユーザプログラムで変更しないでください。
- ユーザプログラム実行中にソフトウェアブレークポイントを設定したアドレスの内容はブレーク命令に置き換えられます。
- ソフトウェアブレークポイントの設定数と [プログラム実行] ダイアログボックスの [テンポラリPCブレークポイント] の設定数の合計は、最大255個です。したがって、ソフトウェアブレークポイントを255個設定した状態では、[プログラム実行] ダイアログボックスの [テンポラリPCブレークポイント] での指定は無効となります。ソフトウェアブレークポイントと [プログラム実行] ダイアログボックスの [テンポラリPCブレークポイント] は、設定数の合計が255個未満で使用してください。
- 遅延分岐命令の直後 (スロット命令) にはブレークを設定しないでください。ブレークを設定した場合、遅延分岐命令を実行した時点でスロット不当命令割り込みが発生し、ブレークしません。

5.5.2 オンチップブ레이크

- 条件成立回数はチャンネル8のみ指定可能です。
- アドレス条件およびデータ条件は、アドレスバスおよびデータバスの値が一致したバスサイクルを検索します。
 - ロングワードアクセス
1回のバスサイクルでロングワードデータをリード/ライトします。
データ条件はロングワードデータの指定のみ有効となります。アドレス条件は、4の倍数のみ有効となります。アクセスサイズ条件はロングワードの指定のみ有効となります。
 - ワードアクセス
1回のバスサイクルでワードデータをリード/ライトします。アドレス条件は、2の倍数のみ有効となります。アクセスサイズ条件はワードの指定のみ有効となります。
 - バイトアクセス
1回のバスサイクルでバイトデータをリード/ライトします。データ条件はバイトデータの指定のみ有効となります。アドレス条件は偶数と奇数どちらも有効となります。

5.5.3 オンエミュレータブ레이크

- 条件成立からブ레이크が発生するまで数サイクルかかります。
- アドレスバス条件およびデータバス条件は、アドレスバスおよびデータバスの値が一致したバスサイクルを検索します。したがって、次のことを考慮して検索条件を設定してください。
 - ロングワードアクセス
1回のバスサイクルでロングワードデータをリード/ライトします。データ条件はロングワードデータの指定のみ有効となります。アクセスサイズ条件はロングワードの指定のみ有効となります。アドレス条件は、4の倍数のみ有効となります。
 - ワードアクセス
1回のバスサイクルでワードデータをリード/ライトします。アクセスサイズ条件はワードの指定のみ有効となります。アドレス条件は、2の倍数のみ有効となります。
 - バイトアクセス
1回のバスサイクルでバイトデータをリード/ライトします。データ条件はバイトデータの指定のみ有効となります。アドレス条件は偶数と奇数どちらも有効となります。

32ビットデータバスの中で無効なデータはマスク機能を使用してデータ条件から外してください。

5.6 トレース機能について

5.6.1 トレース表示内容

- トレース機能のタイムスタンプとIRQ信号取得は、ハードリソースを共有しています。[トレース取得プロパティ]ダイアログボックスの[トレース取得の選択]の設定によりトレース取得内容が異なります。

[詳細タイムスタンプ]選択時	タイムスタンプを 32 ビットで取得し表示します。IRQ 信号は取得しないため表示しません。
[IRQ7-0 全て表示]選択時	IRQ 信号を取得し、表示します。タイムスタンプは下位 16 ビットを 0 に固定し表示します。

- [トレース取得プロパティ]ダイアログボックスの[計測時間単位]設定をユーザクロック(内部クロック ϕ)にした場合、タイムスタンプ表示は32ビット16進表示になります。

5.6.2 トレース取得条件

- 条件成立からトレース停止が発生するまで数サイクルかかります。
- シーケンシャルトレース停止の各通過ポイントおよびリセット条件の成立間隔は、6バスサイクル以上あけてください。
- [トレース取得プロパティ]ダイアログボックスの[トリガ出力間の時間計測]で[チャンネル1->2]、[チャンネル3->4]、[チャンネル5->6]を選択した場合、測定スタート - 測定エンドを6バスサイクル以上離して設定してください。
- [トレース取得プロパティ]ダイアログボックスの[トリガ出力間の時間計測]で[チャンネル7->8]を選択した場合、測定スタート - 測定エンドを15バスサイクル以上離して設定してください。
- トレース停止モード条件は実行開始から6バスサイクル以上離して成立するように設定してください。
- 設定したアドレス条件を実行開始PCとしてユーザプログラムを実行すると正常にシーケンシャルブレーク/トレースストップしない場合があります。設定したアドレス条件の次の命令を実行開始PCとしてユーザプログラムを実行してください。
- パフォーマンス測定機能のチャンネル1を使用している場合は、2点間トレース測定は設定できません。

5.6.3 トレース情報の検索

- [General]ページにおいて、検索範囲入力時、[開始ポイント]オプションに検索を終了するPTRの値、[終了ポイント]オプションに検索を開始するPTRの値を指定することも可能です。
- [トレース取得プロパティ]ダイアログボックスの[計測時間単位]設定をユーザクロック(内部クロック)にした場合、タイムスタンプの検索はできません。

5.6.4 トレース情報のフィルタ

- トレース情報をフィルタリングした場合、保存する範囲の指定はできません。フィルタリングした結果[トレース]ウィンドウに表示されたトレース情報すべてを保存します。保存する範囲を指定したい場合は[Trace Filter]ダイアログボックスの[General]ページよりフィルタ範囲を指定してください。
- [トレース取得プロパティ]ダイアログボックスの[計測時間単位]設定をユーザクロック(内部クロック)にした場合、タイムスタンプのフィルタはできません。

5.7 モニタ機能について

- 前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

5.8 パフォーマンス測定機能について

5.8.1 測定誤差について

パフォーマンス測定機能では、以下の測定誤差が発生します。

- ±1分解能の誤差（分解能が20nsの場合は、±20nsの誤差が発生します。）
±1分解能の誤差が発生するケースとしては、ユーザプログラム実行開始時、ユーザプログラム実行停止時（ブレーク）、開始条件成立時、および終了条件成立時があります。
- 測定用の水晶発振モジュールの周波数安定度：±0.01%

5.8.2 留意事項

- それぞれの測定モードにおいて、終了条件成立から次の開始条件成立までの時間間隔が、1分解能より大きくなるようにしてください。この間隔が、1分解能未満の場合は、終了条件成立から次の開始条件成立までの時間も計測に含まれます。
- [指定範囲内時間測定]は指定範囲からアドレスがはずれた場合を終了条件成立とします。これに対し、[指定アドレス間時間測定]および、[指定アドレス範囲間時間測定]は、指定条件が成立した時点で終了条件成立とします。このためこれらに同じアドレスを指定した場合、[指定範囲内時間測定]の計測結果は、[指定アドレス間時間測定]および、[指定アドレス範囲間時間測定]より大きくなります。
- 各測定は、プリフェッチサイクルのアドレスバスの値を用いて行っています。分岐命令または遅延スロット命令の次の命令に近隣したアドレスに終了アドレス条件を指定すると正しく測定が行えません。あらかじめバストレースの表示で、分岐命令をプリフェッチしたサイクル以降の動作を確認し、分岐により実行されないプリフェッチサイクルのアドレスを終了アドレスに設定してください。
- 2点間トレースを設定している場合は、パフォーマンス測定機能のチャンネル1は設定できません。
- パフォーマンス測定機能の分解能は[コンフィグレーションプロパティ]ダイアログボックスの[時間計測単位]にて設定します。分解能にクロックカウンタを設定した場合は、[RUN-TIME]および[MAX-MIN-TIME]の値はクロックカウンタ値(16進数表示)になります。
測定用カウンタは24ビットあり、最大測定時間は[時間計測単位]の設定値により下記の時間になります。

[時間計測単位]の設定値	最大測定時間
52us	約 14 分
1.6us	約 26 秒
20ns	約 0.33 秒

- 領域アクセス回数測定および指定範囲内コール回数測定の最大測定回数は65535になります。

5.9 プロファイル機能について

- 最適化リンケージエディタが出力するスタック使用量情報ファイル（拡張子は".sni"）がない場合、プロファイルデータの測定時に実行された関数のみを表示します。スタック使用量情報ファイルについては、最適化リンケージエディタのマニュアルを参照してください。
- スタックサイズは実際の値とは異なります。関数の呼び出し経路における目安としてください。また、最適化リンケージエディタが出力するスタック使用量情報ファイル（拡張子は".sni"）がない場合には、スタックサイズを表示しません。
- プロファイラ機能使用時、イベント機能のソフトウェアブレークとエミュレータブレークは使用できません。
- プロファイラ機能使用時、実行時のパラレルアクセス機能は使用できません。

- 本機能は、プログラム実行途中で内部的にブレークを発生させて実現していますので、リアルタイム性はありません。また、測定には誤差があります。
- 本機能は、内部的にオンチップブレーク9~12を使用しています。このため本機能使用中はオンチップブレーク9~12を使用できません。
- [Cycles]項目の表示は、実行時間の測定用カウンタの値（10進数表示）になります。実行時間の測定用カウンタの分解能は、[コンフィグレーションプロパティ]ダイアログボックスの[時間計測単位]にて設定します。各関数の実行時間は、下記の計算式になります。

$$\text{実行時間} = [\text{Cycles}] \text{項目の値} \times [\text{時間計測単位}] \text{の値}$$

- 本機能を有効にする時はあらかじめ、[コンフィグレーションプロパティ]ダイアログボックスにて[AUD機能をエミュレーターで使用(プログラム実行中のメモリアクセス許可)]を無効にする必要があります。

5.10 入力形式について

5.10.1 マスク入力

アドレスバス条件およびデータバス条件では、マスク入力が指定可能です。アドレスの指定では、1ビット、3ビットまたは4ビット単位でマスク指定を行うことができます。マスクしたビットはどんな値でも条件が成立することになります。

アドレスバス条件でマスクを指定する場合は、[マスク指定有効]エリアにマスク値を指定します。

データ条件でマスクを指定する場合も同様に、[マスク指定有効]エリアにマスク値を指定します。

マスク指定の方法は、値の入力時に無視したい桁と同じビットに1を代入し、指定します。マスク指定の例を示します。

表5-6 アドレスマスク指定

項番	入力値	マスク単位	使用例	マスク位置
1	2進数	1ビット	B'00000111	ビット2~0をマスク
2	8進数	3ビット	O'000017	ビット3~0をマスク
3	16進数	4ビット	H'07FF	ビット10~0をマスク

5.11 プログラムのダウンロードについて

内蔵フラッシュメモリへプログラムをダウンロードする場合は、あらかじめ[コンフィグレーションプロパティ]ダイアログボックスにて入力周波数および内蔵フラッシュメモリへのメモリロードの方法を設定する必要があります。

[コンフィグレーションプロパティ]ダイアログボックスについては、「3.1 エミュレーション環境を設定する」を参照してください。

5.12 チュートリアルプログラムについて

5.12.1 チュートリアルプログラムのダウンロード

本製品に付属しているチュートリアルプログラムのワークスペースでは、チュートリアルプログラムのダウンロードに必要なエリアにエミュレーション RAM の設定がなされています。このため、「4 チュートリアル」の説明に沿ってダウンロードを行った場合、チュートリアルプログラムは内蔵フラッシュメモリにダウンロードされずにエミュレーション RAM にダウンロードされます。

内蔵フラッシュメモリへプログラムをダウンロードする場合は、あらかじめ[コンフィグレーションプロパティ]ダイアログボックスにて入力周波数および内蔵フラッシュメモリへのメモリロードの方法を設定する必要があります。ただし、「4 チュートリアル」の説明に沿って High-performance Embedded Workshop を操作する上では、チュートリアルプログラムを内蔵フラッシュメモリにダウンロードする必要はありません。

エミュレーション RAM の設定および内蔵フラッシュメモリへプログラムをダウンロードする方法については、「3.1 エミュレーション環境を設定する」を参照してください。

5.12.2 チュートリアルプログラムの操作についての注意事項

本製品を使用して「4 チュートリアル」の説明に沿って High-performance Embedded Workshop を操作する場合は、下記の内容にて操作を追加または変更してください。

4.6 プログラムの実行	実行を行う前に、ターゲットマイコンをリセットしてください。ターゲットマイコンのリセット方法は「4.14 ターゲットマイコンのリセット」を参照してください。
4.12.3 ステップオーバーコマンドの実行	本製品ではステップアウト実行後の停止位置が sort 関数の呼び出し行になります。ステップオーバーコマンドを実行する前に、一度ステップインコマンドを実行し停止位置を change 関数の呼び出し行に移動してください。
4.19 プロファイル機能	プロファイル機能を有効にする前に、[コンフィグレーションプロパティ]ダイアログボックスにて[AUD 機能をエミュレータで使用(プログラム実行中のメモリアクセス許可)]を無効にしてください。

6. エラーメッセージ

6.1 E6000H のエラーメッセージ

6.1.1 起動時のエラーメッセージ

エミュレータの起動時にエラーが発生した場合、E6000H からエラー出力されます。High-performance Embedded Workshop では、専用のメッセージダイアログボックスに出力します。

表 5.1 に E6000H エミュレータ起動時のエラーメッセージを示します。

表 6.1 起動時のエラーメッセージ

エラーメッセージ	内容と対策
There is no configuration file.	E6000H エミュレータの起動に必要な、Config file がありません。 High-performance Embedded Workshop を終了し、High-performance Embedded Workshop をインストールし直してください。さらにユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
The contents of the configuration file are incorrect.	E6000H エミュレータの起動に必要な、Config file が不正です。 High-performance Embedded Workshop を終了し、High-performance Embedded Workshop をインストールし直してください。さらにユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
Main Board not Support (XX XX XX) Emulator is switched off or not connected	E6000H エミュレータの電源が入っていないか、ユーザーケーブルが接続されていません。 High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
Emulation Board not Support (XX XX XX) Emulator is switched off or not connected	E6000H エミュレータの電源が入っていないか、ユーザーケーブルが接続されていません。 High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
EVA chip Board not Support (XX XX XX) Emulator is switched off or not connected	E6000H エミュレータの電源が入っていないか、ユーザーケーブルが接続されていません。 High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
Can't initialize G/A registers	E6000H エミュレータ本体の初期化に失敗しました。High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
There is no effective clock source	有効なクロックソースが見つかりません。有効なクロックソースを正しく設定し直してください。
This mode can not specify	ターゲットボードのモード端子状態が異常です。動作モード 4 で起動します。モード端子を正しく設定し直してください。
Can't find firmware file Firmware open Error Firmware Download Error Firmware Name Error	E6000H エミュレータの起動に必要なファイルに異常があります。 High-performance Embedded Workshop を終了し、High-performance Embedded Workshop をインストールし直してください。さらにユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
Failed to receive a firmware initialization command.	E6000H エミュレータのファームウェアの起動に失敗しました。 High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。

6. エラーメッセージ

エラーメッセージ	内容と対策
Target system is Vcc down	Vcc が設定したしきい値以下です。
JTAG Timeout Srval Error JTAG Packet Receive Error	High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。

6.1.2 実行中のエラーメッセージ

エミュレーションの実行中にエラーが発生した場合、E6000H からエラー出力されます。High-performance Embedded Workshop では、専用のメッセージダイアログボックスに出力するほか、ステータスバーにメッセージが出力されます。

表 5.2 にメッセージダイアログボックスに出力される E6000H エミュレーションの実行中のエラーメッセージを示します。

表 6.2 メッセージダイアログボックスに出力される実行中のエラーメッセージ

エラーメッセージ	内容と対策
Communication DLL error. Communication Timeout error.	E6000H エミュレータの電源が落ちた、もしくは通信障害が発生しています。 High-performance Embedded Workshop を終了し、ユーザーケーブルを接続し直して E6000H エミュレータの電源を入れて、High-performance Embedded Workshop を再起動してください。本現象が解決できない場合は、発生状況をご連絡ください。
Parallel Access Error	パラレルアクセスでエラーが発生しました。一度ブレイクするまで、パラレルアクセスは禁止になります。

付録 A メニュー一覧

GUIメニューの一覧を表 A.1に示します。

表 A-1 GUIメニュー一覧

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
表示	コマンドライン	Ctrl+L		[コマンドライン]ウィンドウを表示します	
	ワークスペース	Alt+K		[ワークスペース]ウィンドウを表示します	
	アウトプット	Alt+U		[アウトプット]ウィンドウを表示します	
	逆アセンブリ	Ctrl+D		[逆アセンブリ]ウィンドウを表示します	
	CPU	レジスタ	Ctrl+R		[レジスタ]ウィンドウを表示します
		メモリ...	Ctrl+M		[メモリ]ウィンドウを表示します
		IO	Ctrl+I		[IO]ウィンドウを表示します
		ステータス	Ctrl+U		[ステータス]ウィンドウを表示します
		拡張モニタ			[拡張モニタ]ウィンドウを表示します
		モニタ	設定...	Shift+Ctrl+E	
	ウィンドウの選択...				[モニタ]ウィンドウの一覧表示、および追加 / 編集等を行う[ウィンドウの選択]ダイアログボックスを表示します
	シンボル	ラベル	Shift+Ctrl+A		[ラベル]ウィンドウを表示します
		ウォッチ	Ctrl+W		[ウォッチ]ウィンドウを表示します
		ローカル	Shift+Ctrl+W		[ローカル]ウィンドウを表示します
	コード	イベントポイント	Ctrl+E		[イベントポイント]ウィンドウを表示します
		トレース	Ctrl+T		[トレース]ウィンドウを表示します
		スタックトレース	Ctrl+K		[スタックトレース]ウィンドウを表示します
	グラフィック	画像...	Shift+Ctrl+G		[画像]ウィンドウを表示します
		波形...	Shift+Ctrl+V		[波形]ウィンドウを表示します
	パフォーマンス	パフォーマンス解析	Shift+Ctrl+P		[パフォーマンス解析]ウィンドウを表示します
プロファイル		Shift+Ctrl+F		[プロファイル]ウィンドウを表示します	

付録 A メニュー一覧

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
デバッグ	デバッグセッション...			デバッグセッションの一覧表示、および追加 / 削除等が可能な[デバッグセッション]ダイアログボックスを表示します	
	デバッグの設定...			デバッグ時の条件やダウンロードモジュール等の設定を行う[デバッグの設定]ダイアログボックスを表示します	
	CPU のリセット			ターゲットハードウェアをリセットし、PC をリセットベクタアドレスに設定します	
	実行	F5		現在の PC からユーザプログラムを実行します	
	リセット後実行	Shift+F5		ターゲットハードウェアをリセットし、リセットベクタアドレスからユーザプログラムを実行します	
	カーソル位置まで実行			現在の PC からテキストカーソルの位置までユーザプログラムを実行します	
	カーソル位置を PC 値に設定			テキストカーソルの位置に PC を設定します	
	条件を指定して実行...			実行時の PC や PC ブレークポイントの設定が可能な[条件を指定して実行]ダイアログボックスを表示します。	
	PC 位置表示	Shift+Ctrl+Y		現在の PC がある[エディタ] または[逆アセンブリ] ウィンドウをオープンします	
	ステップイン	F11		ユーザプログラムの 1 ブロックを実行して停止します	
	ステップオーバ	F10		ユーザプログラムの 1 ブロックを実行して停止しますが、サブルーチン呼び出しの場合は、サブルーチンには入りません	
	ステップアウト	Shift+F11		現在の関数の終わりに到達するまでユーザプログラムを実行します	
	ステップ...			ステップ動作の設定が可能な[プログラムステップ]ダイアログボックスを表示します	
	ステップモード	自動			[エディタ]ウィンドウがアクティブの場合はソースライン一行だけをステップ実行します、[逆アセンブリ]ウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します
		アセンブリ			アセンブリ言語命令単位にステップ実行します
		ソース			ソースライン一行だけをステップ実行します
	プログラムの停止	Esc		ユーザプログラムの実行を停止します	
	初期化			本エミュレータを切断し、再接続します	
	接続			本エミュレータを接続します	
	接続解除			本エミュレータを切断します	
メモリの保存			任意のアドレス領域を保存します		
メモリのペリファイ			アドレス領域を検証します		
オーバーレイの構成...			オーバーレイ使用時の優先セクショングループの設定を行います		
ダウンロード			オブジェクトプログラムをロードします		
アンロード			オブジェクトプログラムをアンロードします		

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
基本設定	カスタマイズ...			High-performance Embedded Workshop アプリケーションの設定をカスタマイズします	
	オプション...			High-performance Embedded Workshop アプリケーションのオプションを設定します	
	表示の形式...			ウィンドウの表示色、フォント、キーワードなどを設定します	
	基数	16 進数			数値の表示 / 入力時の基数のデフォルト設定を 16 進数とします
		10 進数			数値の表示 / 入力時の基数のデフォルト設定を 10 進数とします
		8 進数			数値の表示 / 入力時の基数のデフォルト設定を 8 進数とします
2 進数				数値の表示 / 入力時の基数のデフォルト設定を 2 進数とします	
エミュレータ	システム...			本エミュレータの設定を行う[コンフィグレーションプロパティ]ダイアログボックスを表示します	

付録 B コマンドライン一覽

コマンド一覽を表 B.1に示します。

表 B.1 コマンド一覽

項番	コマンド名	短縮形	説明
1	!	-	コメント
2	ADD_FILE	AF	カレントプロジェクトにファイルを追加します
3	ANALYSIS	AN	性能分析機能の有効化 / 無効化
4	ANALYSIS_RANGE	AR	性能分析範囲の設定、表示
5	ANALYSIS_RANGE_DELETE	AD	性能分析範囲の削除
6	ASSEMBLE	AS	アセンブルの実行
7	ASSERT	-	コンディションのチェック
8	AUTO_COMPLETE	AC	オートコンプリート機能の有効化 / 無効化
9	BREAKPOINT_ONCHIP	BC	オンチップブレークポイントの表示、シーケンシャルブレークの設定および PtoP 時間測定の設定
10	BREAKPOINT_ONCHIPn	BCn	オンチップブレークポイントの各チャンネル設定
11	BREAKPOINT_ONCHIP_CLEAR	BCC	オンチップブレークポイントの削除
12	BREAKPOINT_ONCHIP_ENABLE	BCE	オンチップブレークポイントの有効 / 無効の切り換え
13	BREAKPOINT_ONEMULATOR	BE	オンエミュレータブレークポイントの表示
14	BREAKPOINT_ONEMULATORn	BE n	オンエミュレータブレークポイントの各チャンネル設定
15	BREAKPOINT_ONEMULATOR_CLEAR	BEC	オンエミュレータブレークポイントの削除
16	BREAKPOINT_ONEMULATOR_ENABLE	BEE	オンエミュレータブレークポイントの有効 / 無効の切り換え
17	BREAKPOINT_SOFTWARE	BS	ソフトウェアブレークポイントの設定
18	BREAKPOINT_SOFTWARE_CLEAR	BSC	ソフトウェアブレークポイントの削除
19	BREAKPOINT_SOFTWARE_ENABLE	BSE	ソフトウェアブレークポイントの有効 / 無効の切り換え
20	BUILD	BU	カレントプロジェクトのビルド処理を開始します
21	BUILD_ALL	BL	カレントプロジェクトのすべてのビルド処理を開始します
22	CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
23	CHANGE_PROJECT	CP	プロジェクトの設定
24	CHANGE_SESSION	CS	セッションの設定
25	CLOSE_WORKSPACE	CW	ワークスペースを閉じる
26	CONFIGURE_PLATFORM	CPF	プラットフォームの設定
27	DEFAULT_OBJECT_FORMAT	DO	デフォルトオブジェクト (プログラム) フォーマットの設定
28	DEVICE_TYPE	DE	デバイスタイプの選択
29	DISASSEMBLE	DA	逆アセンブル表示
30	EMULATOR_CLOCK	ECK	ターゲットマイコンの入カクロック設定
31	ERAM	ERM	ERAM 設定の表示と設定
32	ERASE	ER	コマンドラインウィンドウの内容クリア
33	EVALUATE	EV	式の計算
34	EXMONITOR_DISPLAY	EXMD	拡張モニタの内容表示
35	EXMONITOR_SET	EXMS	拡張モニタ項目の表示 / 非表示の切り換え
36	EXMONITOR_SETRATE	EXMS R	実行中およびブレーク中の拡張モニタ更新時間の設定
37	FILE_LOAD	FL	オブジェクト (プログラム) ファイルのロード

付録 B コマンドライン一覧

項番	コマンド名	短縮形	説明
38	FILE_SAVE	FS	メモリ内容のファイルセーブ
39	FILE_UNLOAD	FU	ファイルをアンロードします
40	FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
41	FLASH_MEMORY	FLM	内蔵フラッシュメモリの消去
42	GENERATE_MAKE_FILE	GM	High-performance Embedded Workshop 外でビルドするための make ファイルを生成します
43	GO	GO	ユーザプログラムの実行
44	GO_RESET	GR	リセットベクタからのユーザプログラムの実行
45	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
46	HALT	HA	ユーザプログラムの停止
47	HELP	HE	コマンドラインのヘルプ表示
48	INITIALIZE	IN	デバッグプラットフォームの初期化
49	JTAG_CLOCK	JCK	JTAG クロック (TCK) の設定および表示
50	LOG	LO	ロギングファイルの操作
51	MAP_DISPLAY	MA	メモリマッピング情報の表示
52	MEMORY_COMPARE	MC	メモリ内容の比較
53	MEMORY_DISPLAY	MD	メモリ内容の表示
54	MEMORY_EDIT	ME	メモリ内容の変更
55	MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
56	MEMORY_FIND	MI	メモリ範囲内でデータを検索します
57	MEMORY_MOVE	MV	メモリブロックの移動
58	MEMORY_TEST	MT	メモリブロックのテスト
59	MODE	MO	MCU モードを設定または表示
60	MONITOR_CLEAR	MOC	モニタポイントの削除
61	MONITOR_DISPLAY	MOD	モニタ内容の表示
62	MONITOR_REFRESH	MOR	モニタ内容に自動更新制御
63	MONITOR_SET	MOS	モニタポイントの設定と表示
64	OPEN_WORKSPACE	OW	ワークスペースのオープン
65	PROFILE	PR	プロファイルの有効化/無効化
66	PROFILE_DISPLAY	PD	プロファイル情報の表示
67	PROFILE_SAVE	PS	プロファイル情報のファイル出力
68	QUIT	QU	High-performance Embedded Workshop の終了
69	RADIX	RA	入力基数の設定
70	REFRESH	RF	メモリ関連ウィンドウの更新
71	REGISTER_DISPLAY	RD	CPU レジスタ値の表示
72	REGISTER_SET	RS	CPU レジスタ値の設定
73	REMOVE_FILE	REM	指定ファイルをカレントプロジェクトから削除します
74	RESET	RE	CPU のリセット
75	SAVE_SESSION	SE	現在のセッションをセーブ
76	SLEEP	-	コマンド実行の遅延
77	STEP	ST	ステップ実行 (命令単位またはソース行単位)
78	STEP_MODE	SM	ステップモードの選択
79	STEP_OUT	SP	PC 位置の開数を終了するまでのステップ実行
80	STEP_OVER	SO	ステップオーバー実行
81	STEP_RATE	SR	ステップ実行速度の設定、表示
82	SUBMIT	SU	エミュレータコマンドファイルの実行

付録 B コマンドライン一覧

項番	コマンド名	短縮形	説明
83	SYMBOL_ADD	SA	シンボルの設定
84	SYMBOL_CLEAR	SC	シンボルの削除
85	SYMBOL_LOAD	SL	シンボル情報ファイルのロード
86	SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
87	SYMBOL_VIEW	SV	シンボルの表示
88	STATUS	STS	[ステータス] ウィンドウの内容表示
89	SAVE_WORKSPACE	SW	現在のワークスペースの保存
90	TCL	-	TCLの有効/無効の切り換え
91	TEST_EMULATOR	TEM	内蔵フラッシュメモリのテスト
92	TIMER	TI	実行時間測定タイマの分解能の設定、表示
93	TOOL_INFORMATION	TO	現在登録されているツールの情報をファイルへ出力
94	TRACE	TR	トレース情報の表示
95	TRACE_ACQUISITION	TA	トレース取得条件の設定、表示
96	TRACE_ACQUISITIONn	TAn	トレース取得条件の各チャンネルおよび PtoP ポイントの設定
97	TRACE_ACQUISITION_CLEAR	TAC	トレース取得条件の削除
98	TRACE_BINARY_COMPARE	TBC	トレースバイナリファイルと現在のトレース情報の比較
99	TRACE_BINARY_SAVE	TBV	トレース情報をバイナリファイルに保存
100	TRACE_FILTER	TF	トレース情報のフィルタ
101	TRACE_SAVE	TV	トレース情報をテキストファイルに保存
102	TRACE_STATISTIC	TST	統計情報解析の実行
103	UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトの依存関係を更新します
104	USER_SIGNALS	US	ユーザシグナル情報の有効/無効の切り換え
105	WATCH_ADD	WA	Watch アイテムの追加
106	WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
107	WATCH_DELETE	WD	Watch アイテムの削除
108	WATCH_DISPLAY	WI	ウォッチウィンドウの内容の表示
109	WATCH_EDIT	WE	Watch アイテムの値の編集
110	WATCH_EXPAND	WX	Watch アイテムの展開または縮小
111	WATCH_RADIX	WR	Watch アイテムの表示基数の変更
112	WATCH_SAVE	WS	ウォッチウィンドウの表示内容をファイルに保存

各コマンドのシンタックスについてはオンラインヘルプを参照ください。

ルネサスマイクロコンピュータ開発環境システム
SH7058 E6000Hエミュレータ HS7058EPH62H
SH7058S、SH7059 E6000Hエミュレータ HS7059EPH62H
ユーザズマニュアル

発行年月日 2009年6月30日 Rev.1.00

発行 株式会社ルネサス テクノロジ 営業統括部
〒100-0004 東京都千代田区大手町 2-6-2

編集 株式会社ルネサスソリューションズ
グローバルストラテジックコミュニケーション本部
カスタマサポート部

© 2009. Renesas Technology Corp., All rights reserved. Printed in Japan.

株式会社ルネサス テクノロジ 営業統括部 〒100-0004 東京都千代田区大手町2-6-2 日本ビル

営業お問合せ窓口
株式会社ルネサス販売

RENESAS

<http://www.renesas.com>

本			社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
西			社	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル)	(042) 524-8701
東			社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア)	(022) 221-1351
北			支	〒970-8026	いわき市平字田町120 (ラトブ)	(0246) 22-3222
い			店	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田)	(029) 271-9411
茨			店	〒950-0087	新潟市中央区東大通1-4-2 (新潟三井物産ビル)	(025) 241-4361
新			社	〒390-0815	松本市深志1-2-11 (昭和ビル)	(0263) 33-6622
松			社	〒460-0008	名古屋市中区栄4-2-29 (名古屋広小路ブレイス)	(052) 249-3330
中			社	〒541-0044	大阪市中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
関			社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル)	(076) 233-5980
北			支	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング)	(082) 244-2570
広			店	〒812-0011	福岡市博多区博多駅前2-17-1 (博多プレステージ)	(092) 481-7695
九			社			

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：コンタクトセンター E-Mail: csc@renesas.com

SH7058 E6000H エミュレータ HS7058EPH62H
SH7058S、SH7059 E6000H エミュレータ HS7059EPH62H
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

RJJ10J2514-0100